Monitoring AWS EC2 Cloud

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizeable computing capacity-literally, server instances in Amazon's data centers-that you use to build and host your software systems. You can get access to the infrastructure resources that EC2 provides by using APIs, or web tools and utilities.

With EC2, you use and pay for only the capacity that you need. This eliminates the need to make large and expensive hardware purchases, reduces the need to forecast traffic, and enables you to automatically scale your IT resources to deal with changes in requirements or spikes in popularity related to your application or service.

With many mission-critical applications now being delivered via the cloud, end-users have come to expect from the cloud the same quality of service that local service deployments are known to deliver. This means that even the slightest dip in performance levels will not be tolerated!

A sudden non-availability of the cloud, no matter how brief, or a slowdown/failure of any of its regions/availability zones/instances, can make it impossible for cloud providers to build and launch mission-critical services on the cloud and for consumers to access these services for prolonged periods. If you are a (public or private) cloud service provider therefore, your primary concerns would be - can people access my service? Is the self service portal up? Can users see their VMs? Can users connect to their VMs? If not, you need to be able to determine why the problem is happening - is it the web front-end? is it due to the virtualization platform? is it due to the SAN? etc. The action you take depends on what you diagnose as being the root-cause of the problem. Besides problem diagnosis, you are also interested in understanding how you can get more out of your current cloud investments. You want to be able to see how to balance load across your servers to serve a maximum number of users and how you can optimize the capacity of the infrastructure without sacrificing on performance. You need performance management “FOR” the cloud.

eG Enterprise is a unique solution that can provide you performance management FROM the cloud, OF the cloud and FOR the cloud!

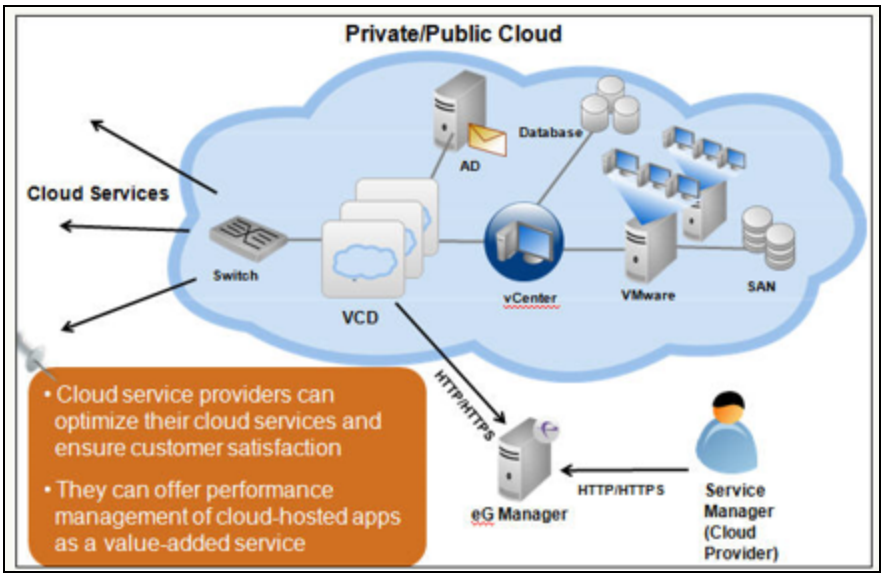


Figure 1.1: How eG monitors the cloud

To deliver performance management FOR the AWS EC2 cloud in particular, the solution offers two specialized monitoring models - the AWS EC2 Cloud model and the AWS EC2 Region model. The AWS EC2 Cloud monitoring model provides you with proactive updates on the overall health and status of the cloud and points you to unavailable regions/availability zones and resource-hungry instances in the cloud. To zoom into the health of specific regions and the instances operating within those regions, use the AWS EC2 Region model.

This document engages in detailed discussions on both the models.

Chapter 2: Pre-Requisites for Monitoring the AWS EC2 Cloud

The following pre-requisites are to be satisfied while you start monitoring the AWS EC2 Cloud models:

- Since the AWS EC2 Cloud can be monitored only in an agentless manner, at least one remote agent should be configured in the environment;
- The system hosting the remote agent should be configured with Internet connectivity;
- You should buy the capability to launch and monitor EC2 instances on the cloud;
- Some tests require the AWS CloudWatch service to be enabled. This is a paid web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. For enabling this service, you need to pay CloudWatch fees. Refer to the AWS web site for the fee details.
- A Secret key and an Access key should be obtained, and the eG remote agent should be configured with these keys

Note that the aforesaid pre-requisites apply to both the AWS EC2 cloud models that eG Enterprise offers.

2.1 Purchasing Monitoring Rights for Amazon EC2 Instances

The Amazon instances are classified into six families namely Standard, Micro, High-Memory, High-CPU, Cluster Compute, and Cluster GPU.

Basically, configuring an instance within an Amazon EC2 Region and monitoring that instance is a **paid** service i.e., you pay charges for both configuring an instance in the Amazon EC2 cloud based on your requirement and collecting metrics from that particular instance. The payment varies according to the instance types and the Region in which the instance is deployed.

For more detailed information regarding the Amazon EC2 purchasing options refer http://aws.amazon.com/ec2/purchasing-options/

For more details regarding the pricing of the Amazon EC2 instances based on the Regions where they are deployed, refer http://aws.amazon.com/ec2/pricing/

2.2 Enable Amazon CloudWatch

Amazon CloudWatch is a web service that enables you to monitor your Amazon EC2 instances, in real-time. Metrics such as CPU utilization, latency, and request counts are provided automatically for these AWS resources. You can also supply your own custom application and system metrics, such as memory usage, transaction volumes, or error rates, and Amazon CloudWatch will monitor these metrics too at a **nominal charge**. With Amazon CloudWatch, you can access up-to-the-minute statistics, view graphs, and set alarms for your metric data. Amazon CloudWatch functionality is accessible via API, command-line tools, the AWS SDK, and the AWS Management Console. Therefore, while configuring the tests for the AWS EC2 Component types, set the **CLOUDWATCH ENABLED** flag to **true**. By default, this flag is set to **true.**

2.3 Obtaining an Access key and Secret key

To monitor the Amazon cloud infrastructure, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account.

For this purpose, you need to follow the following broad steps:

1. Create a special user on the AWS cloud for monitoring purposes.
2. Configure the eG agent with the access key and secret key of the special user.

To create a user on the AWS cloud, do the following:

1. Login to the AWS management console as a root user.
2. Upon successful login, Figure 2.1 will appear. Keep scrolling down Figure 2.1 until you view the IAM option (highlighted by Figure 2.2).

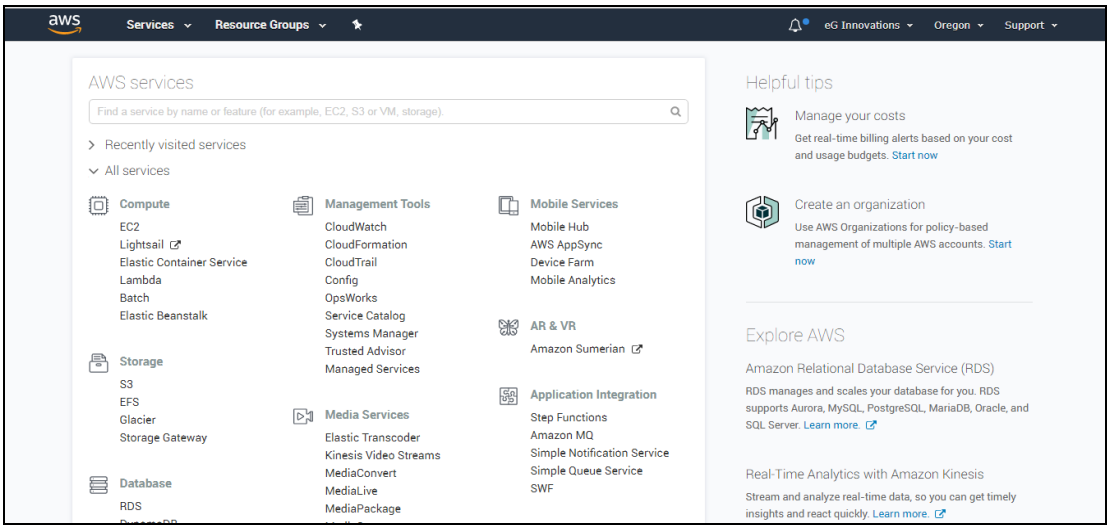


Figure 2.1: The AWS Services page that appears as soon as a root user logs into the AWS management console

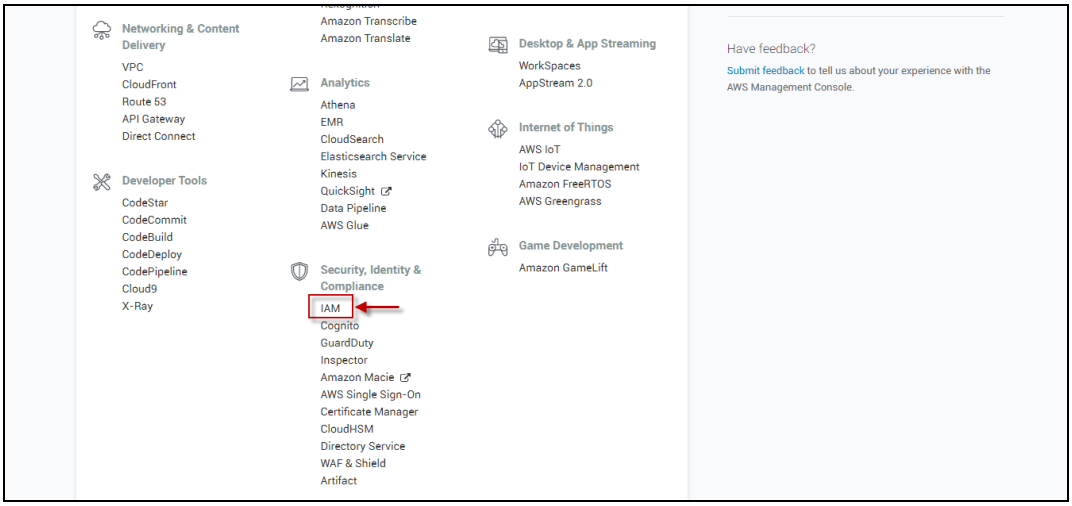


Figure 2.2: Scrolling down the AWS Services page to view the IAM option

3. Figure 2.3 will then appear. The first step to creating a user is to create a policy that defines the rights and privileges of that user. To create a policy, click on the Policies link in the left panel (as indicated by Figure 2.3).

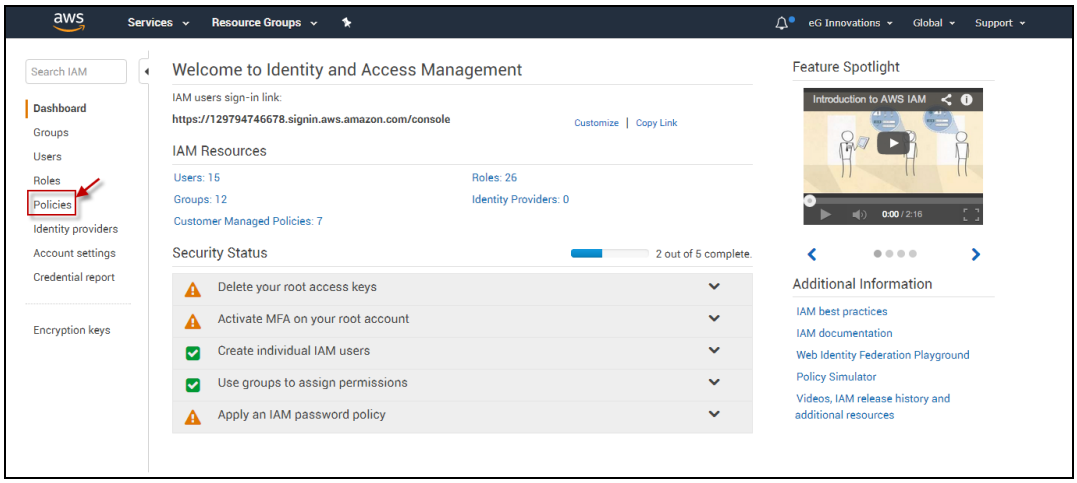


Figure 2.3: Clicking on the Policies link in the left panel

4. Figure 2.4 will then appear listing all the policies that pre-exist. Click on **Create Policy** to create a new policy.

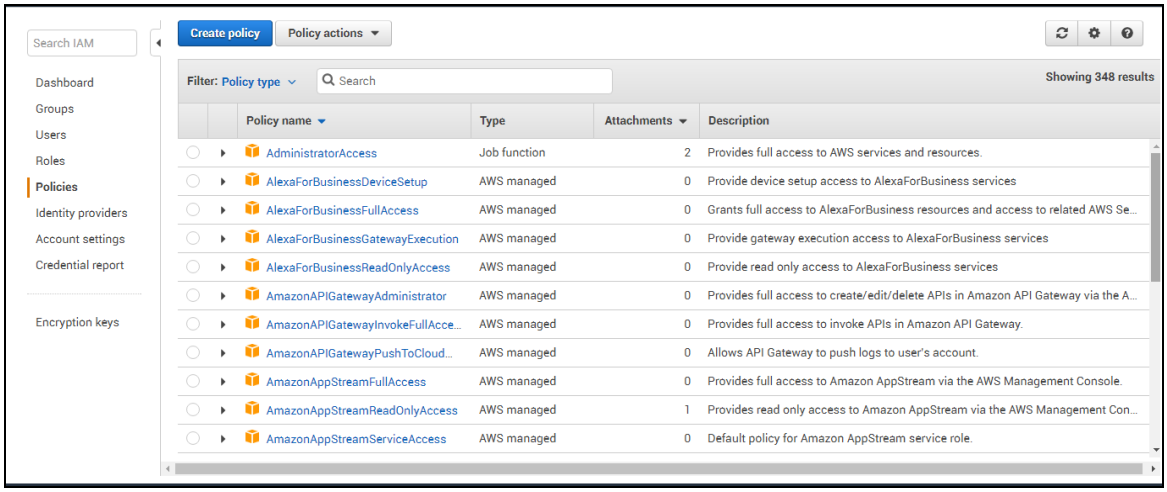


Figure 2.4: Clicking on Create Policy

5. Figure 2.5 will then appear. Click on the **JSON** tab page in Figure 2.5.

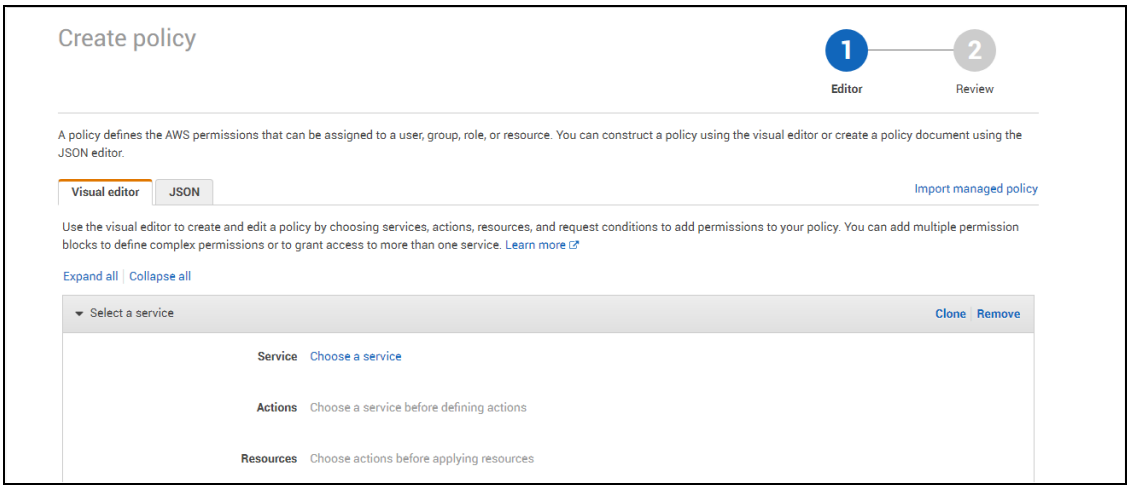


Figure 2.5: Switching to the JSON tab page

6. Figure 2.6 will then appear.

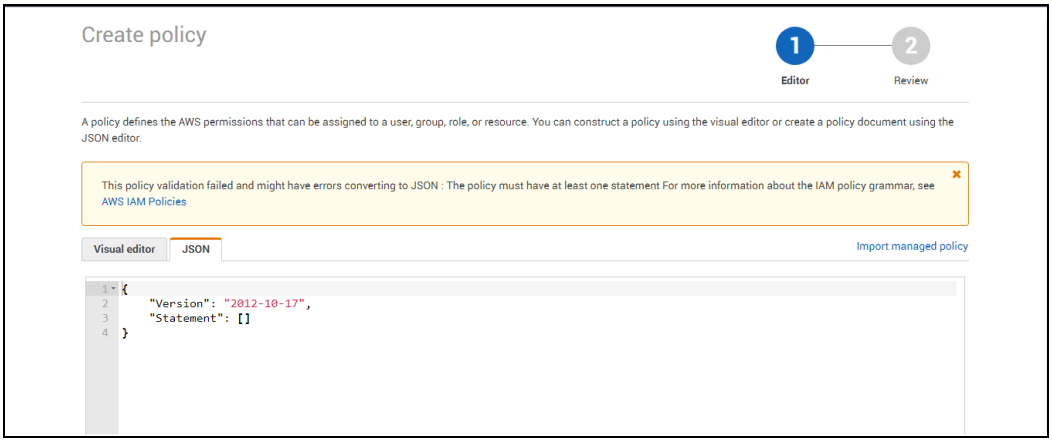


Figure 2.6: The JSON tab page

7. Replace the contents of the **JSON** tab page with the following (see Figure 2.7):

```
{
"Version": "2012-10-17",
"Statement": [
{
"Action": [
"acm:DescribeCertificate",
```

```
"acm:ListCertificates",
"acm:GetCertificate",
"acm:Certificate*",
"autoscaling:Describe*",
"budgets:Describe*",
"cloudfront:List*",
"cloudfront:GetDistributionConfig",
"cloudfront:GetStreamingDistributionConfig",
"cloudsearch:Describe*",
"cloudtrail:DescribeTrails",
"cloudtrail:GetTrailStatus",
"cloudwatch:Describe*",
"cloudwatch:Get*",
"cloudwatch:List*",
"dynamodb:List*",
"dynamodb:Describe*",
"ec2:Describe*",
"ec2:Get*",
"ecs:List*",
"ecs:Describe*",
"elasticache:Describe*",
"elasticache:List*",
"elasticbeanstalk:Describe*",
"elasticbeanstalk:List*",
"elasticfilesystem:Describe*",
"elasticloadbalancing:Describe*",
"elasticmapreduce:Describe*",
"elasticmapreduce:List*",
"iam:Get*",
"iam:List*",
"iam:GenerateCredentialReport",
"iot:Describe*",
"iot:List*",
```

```
"kinesis:List*",
"kinesis:Describe*",
"kinesis:Get*",
"lambda:List*",
"logs:Get*",
"logs:Describe*",
"logs:FilterLogEvents",
"logs:TestMetricFilter",
"logs:OutputLogEvent",
"opsworks:Describe*",
"polly:Describe*",
"polly:GetLexicon",
"polly:ListLexicons",
"rds:Describe*",
"rds:List*",
"redshift:Describe*",
"redshift:ViewQueriesInConsole",
"route53:List*",
"s3:Get*",
"s3:List*",
"s3:S3Object*",
"s3:ObjectListing",
"ses:ListIdentities",
"ses:Get*",
"support:*",
"sns:Get*",
"sns:List*",
"sns:Publish",
"sqs:List*",
"sqs:Get*",
"storagegateway:Describe*",
"storagegateway:List*",
"waf:List*",
```

```
"waf:Get*",
"workspaces:Describe*",
"Organizations:List*",
"Organizations:Describe*"
],
"Effect": "Allow",
"Resource": "*"
}
]
}
```

Note:

If you copy the above code block directly from this document and paste it in the JSON tab page, you will find that the page numbers in the document also get copied on to the tab page inadvertently. Therefore, after copying the code block to the **JSON** tab page, make sure you remove the page numbers from the code block and then proceed.

Figure 2.7: Replacing the contents of the JSON tab page

8. Then, click the **Review policy** button in Figure 2.7 to review the policy that you have defined. This will open Figure 2.8, where you have to provide a name for the new policy and a brief description of the policy.

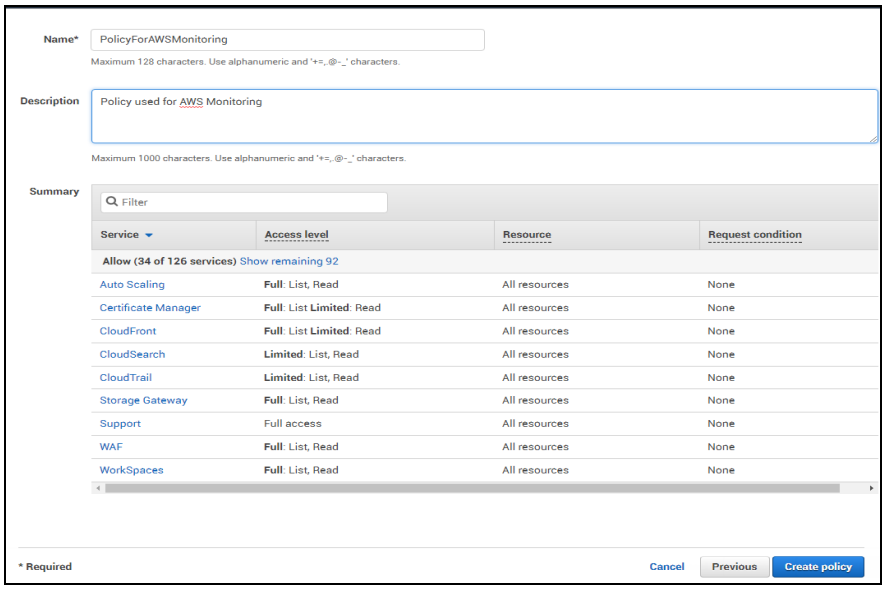


Figure 2.8: Reviewing the policy

9. The **Summary** section of Figure 2.8 lists all the services that this policy allows access to, the level of access (whether Full or Limited), and the resources that can be accessed.

10. Then, click on the **Create Policy** button in Figure 2.8 to create the new policy.

11. Figure 2.9 will then appear displaying the new policy.

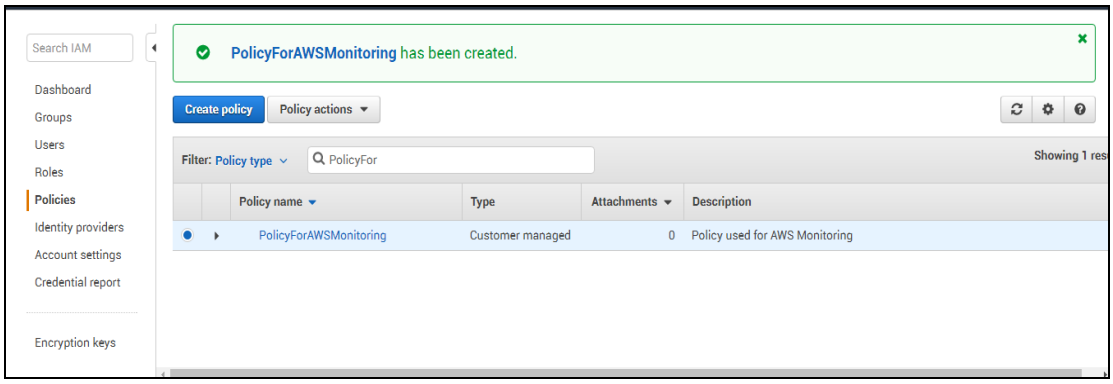


Figure 2.9: Viewing the newly created policy

12. Now, proceed to create a new user. For that, first click the **Users** option in the left panel (as indicated by Figure 2.10).

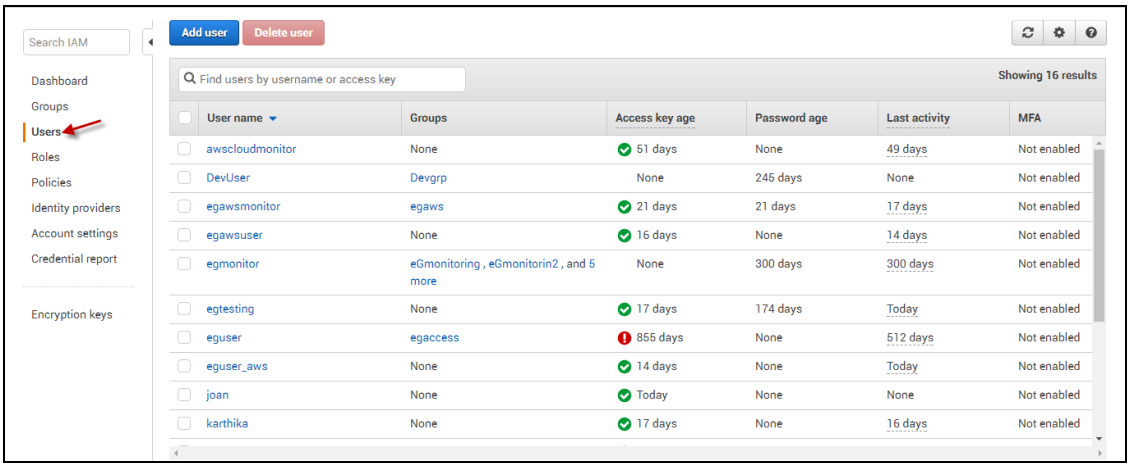


Figure 2.10: Clicking on the Users option

13. This will open Figure 2.11. Here, specify the name of the new user and set **Programmatic access** as the **Access type**. Then, click the **Next: Permissions** button to move to the next step of the user creation.

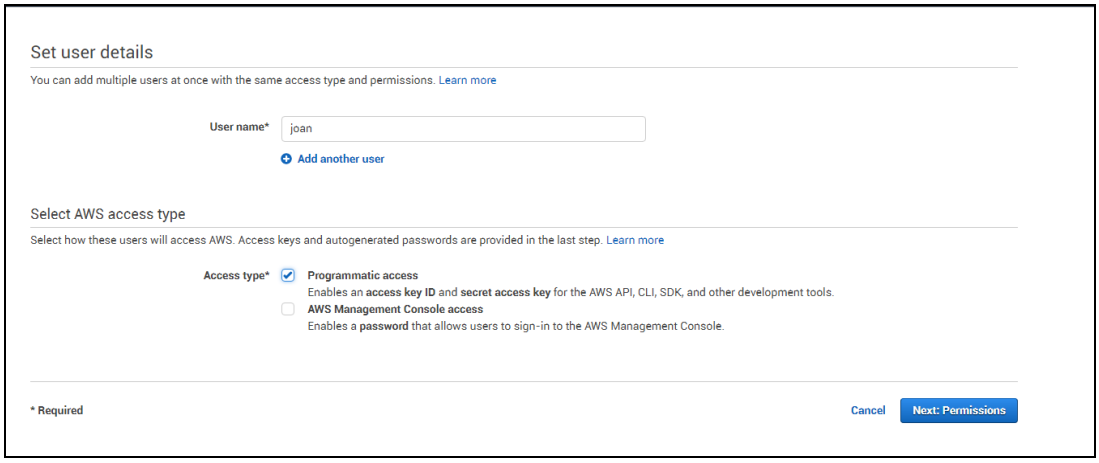


Figure 2.11: Creating a new user

14. Clicking on the **Next: Permissions** button, will invoke Figure 2.12. Click the **Attach Existing Policies Directly** button in Figure 2.12 to associate the newly created policy with the new user. Then, from the list of policies displayed therein, click the check box corresponding to the

policy that you created newly, and then click the **Next: Review** button.

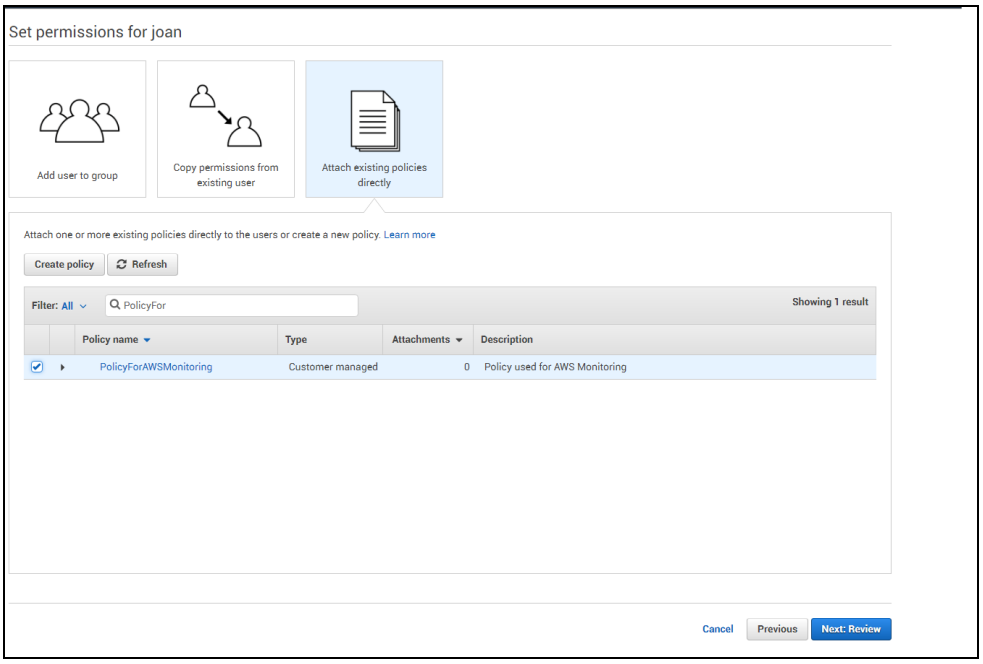


Figure 2.12: Associating the new policy with the new user

15. Figure 2.13 will then appear. Click the **Create User** button in Figure 2.13 to create the user.

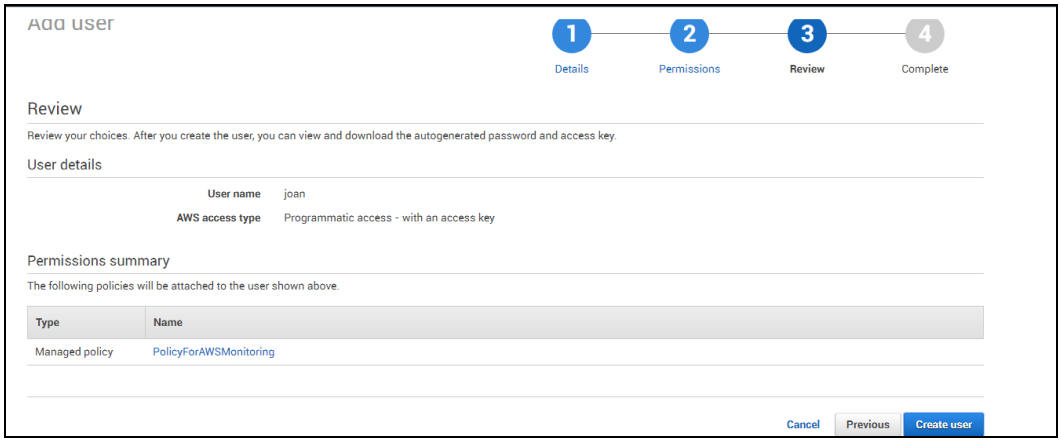


Figure 2.13: Reviewing the new user's details

16. Figure 2.14 will then appear displaying the new user and the access and secret keys assigned to the new user. Click on the **Show link** adjacent to the encrypted secret key to view it. Once you are able to view both keys, make a note of the keys.

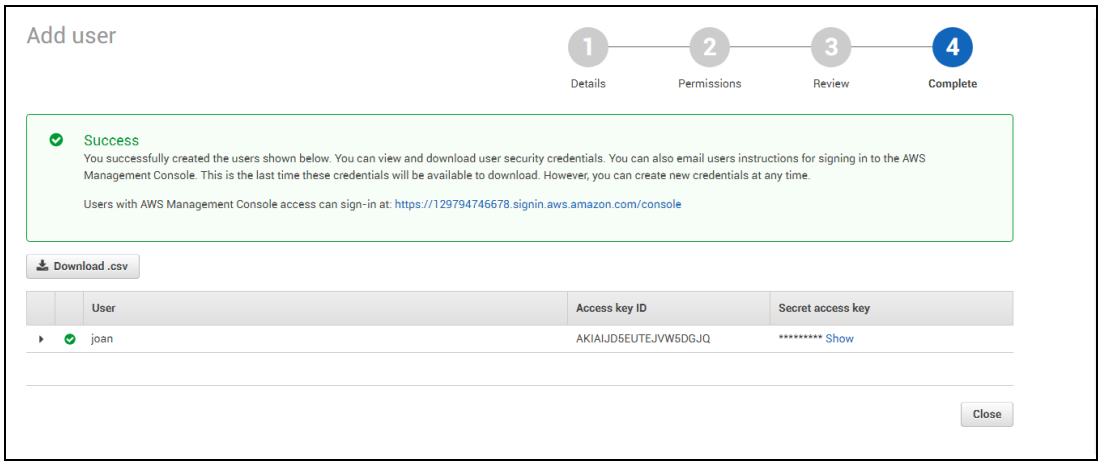


Figure 2.14: Viewing the access and secret key of the new user

Make sure to configure the eG tests with the access key and secret key that you see in Figure 2.14.

Chapter 3: Administering the eG Manager to Monitor the AWS EC2 Cloud

To achieve this, follow the steps given below:

1. Log into the eG administrative interface.
2. eG Enterprise automatically discovers the AWS EC2 Cloud component. If the AWS EC2 Cloud component is already discovered, use the Infrastructure -> Components ->Manage/Unmanage menu to manage it. Otherwise run discovery process as shown in Figure 3.1 from the menu sequence: Infrastructure ->Components -> Discovery. Provide the credentials that you had obtained while creating the AWS account. To know more about the AWS account, refer to Section 2.3 of this document.

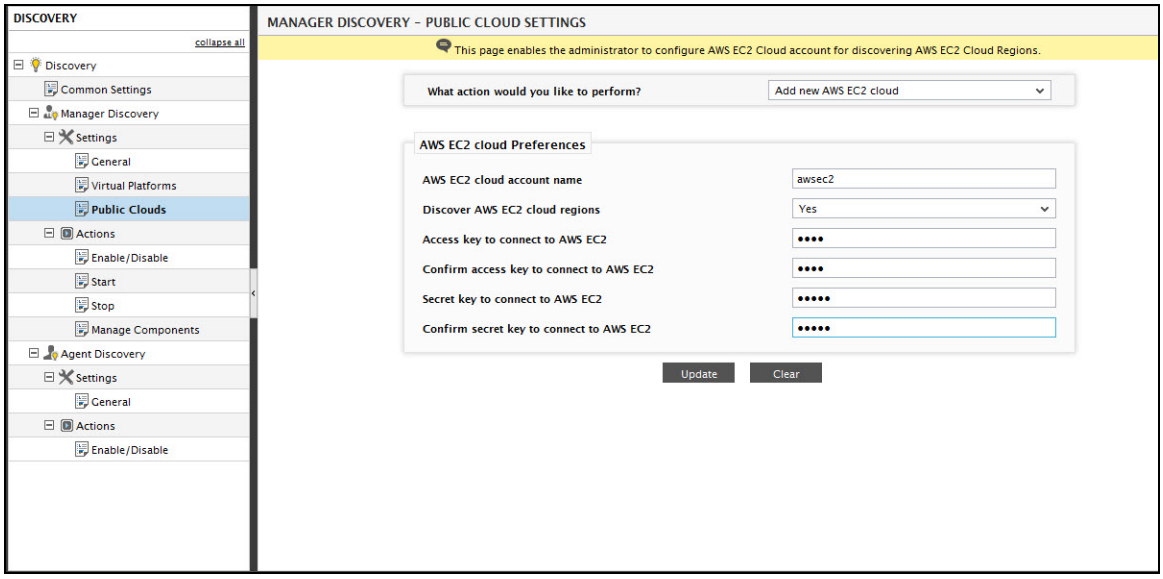


Figure 3.1: Providing the credentials during discovery of the AWS EC2 Cloud component

3. To manage the discovered components, go to the Infrastructure -> Components -> Manage/Unmanage page. The process of managing a component is clearly depicted by Figure 3.2 below.

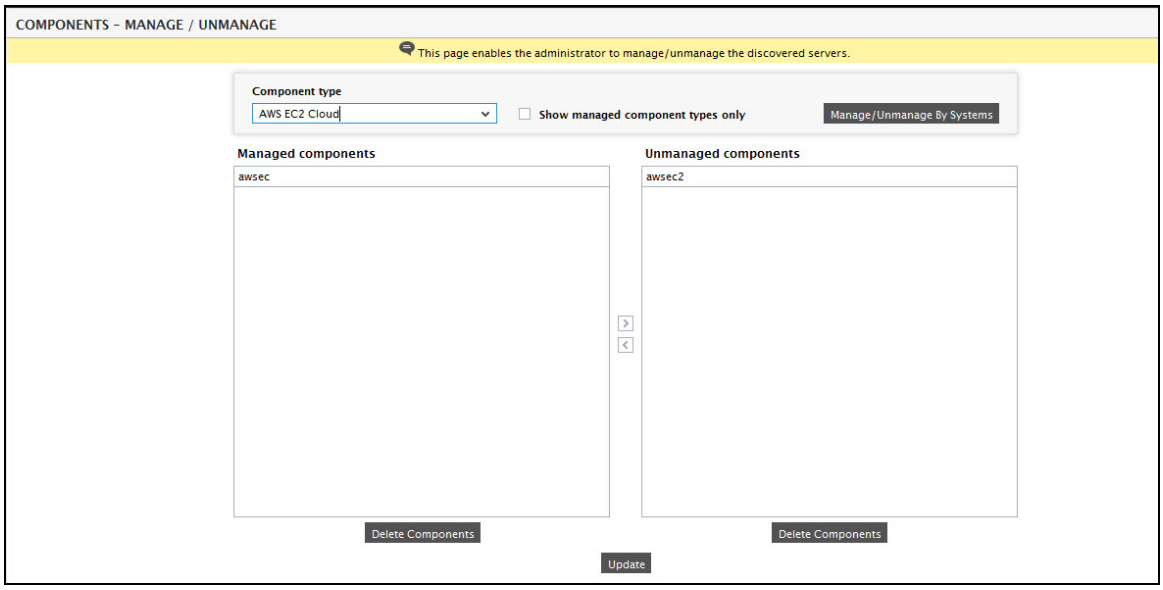


Figure 3.2: Managing the discovered AWS EC2 Cloud components

Note:

For a more detailed procedure for managing components, refer to Configuring and Monitoring Web Servers document.

4. You can also manually add the AWS EC2 Cloud component using Infrastructure -> Components -> Add/Modify. Remember that components manually added are managed automatically. While manually adding the AWS EC2 Cloud component, make sure that you provide a valid name in the HOST text box instead of an IP address. In order to provide a valid name, ensure that the **AllowQualifiedHostnames** flag is set to **Yes** in the **eg_services.ini** file of the **<EG_INSTALL_DIR>\manager\config** directory.

5. Now, when you attempt to sign out of the eG administrative interface, Figure 3.3 appears, listing the tests that require manual configuration.

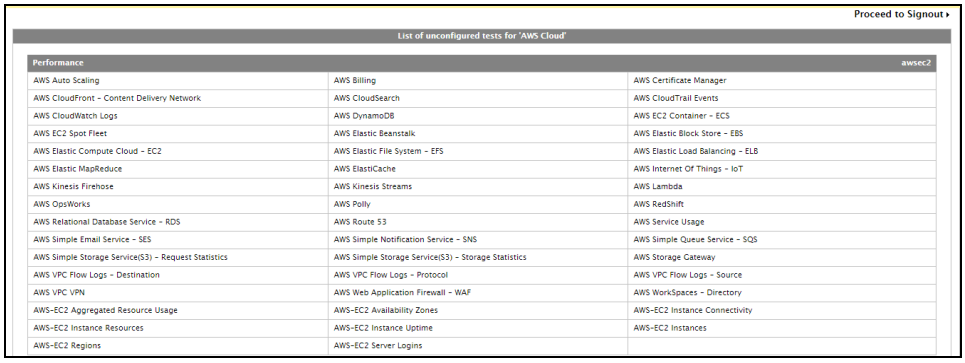


Figure 3.3: The list of unconfigured tests for AWS EC2 Cloud

6. Click on the **AWS – EC2 Regions** test to configure it. This test reports the availability of the default *Region* and enables the administrators to figure out the time taken by the default Region to respond to responses. To know how to configure the test, Click Here.
7. Once the test is configured, signout of the eG Administrative interface.

4

Chapter 4: Monitoring the AWS EC2 Cloud

Figure 4.1 depicts the *AWS EC2 Cloud* monitoring model that eG Enterprise offers out-of-the-box for monitoring the Amazon EC2 cloud.

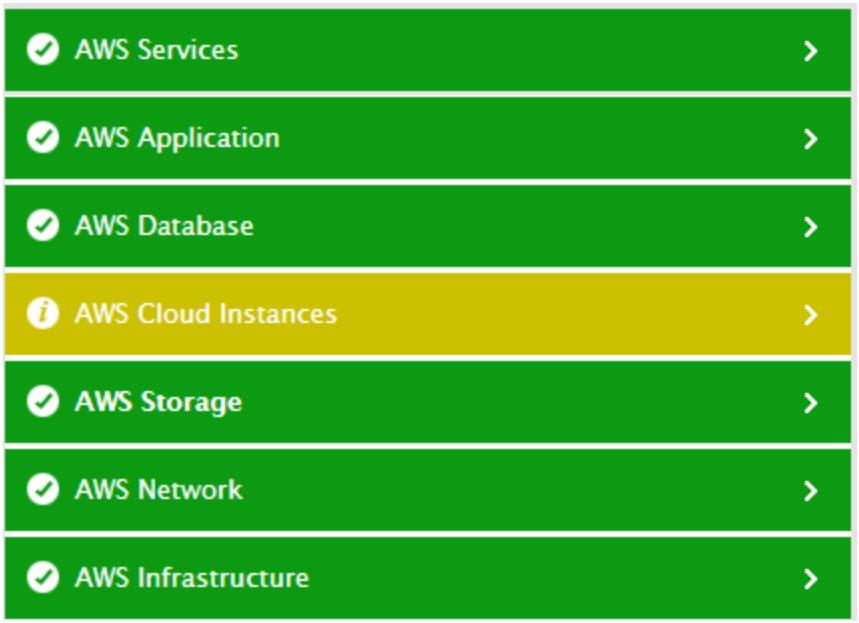


Figure 4.1: Layer model of the AWS EC2 Cloud

Each layer of this model is mapped to tests that reveal the availability of the cloud and whether the regions/availability zones/instances on the cloud are accessible. Using these statistics, cloud administrators can find quick and accurate answers for the following critical performance queries:

- Is web-based (HTTP/HTTPS) access to the cloud available?
- Does it take an unreasonably long time to establish contact with the cloud?
- How many regions does the cloud support? What are they?
- Is any region unavailable?
- Were any connectivity issues experienced while attempting to connect to a region? If so, which region is this?
- How many availability zones exist in each region? What are they?
- Is any availability zone currently unavailable? If so, which one is it?

- Is the default region on the cloud accessible? If so, is it taking too long to connect to the default region?
- Are all instances on the cloud accessible over the network?
- Are any instances powered off currently?
- Were any instances launched/removed recently? If so, which ones are these?
- What type of instances are resource-intensive?
- Is any particular instance consuming too much CPU?
- Is the network traffic to/from any instance unusually high?
- Is the disk I/O of instances optimal?
- Was any instance rebooted recently? If so, which one is it?

Note:

The eG agent reports metrics for only those regions, availability zones, and instances on the cloud that the configured AWS user account is allowed to access.

The sections that will follow discuss each of the layers of Figure 4.1 in great detail.

4.1 The AWS Infrastructure Layer

Using the tests mapped to this layer, you can promptly detect the non-availability of the cloud, inaccessibility of regions and availability zones on the cloud, and connection bottlenecks experienced while connecting to the cloud or its components.

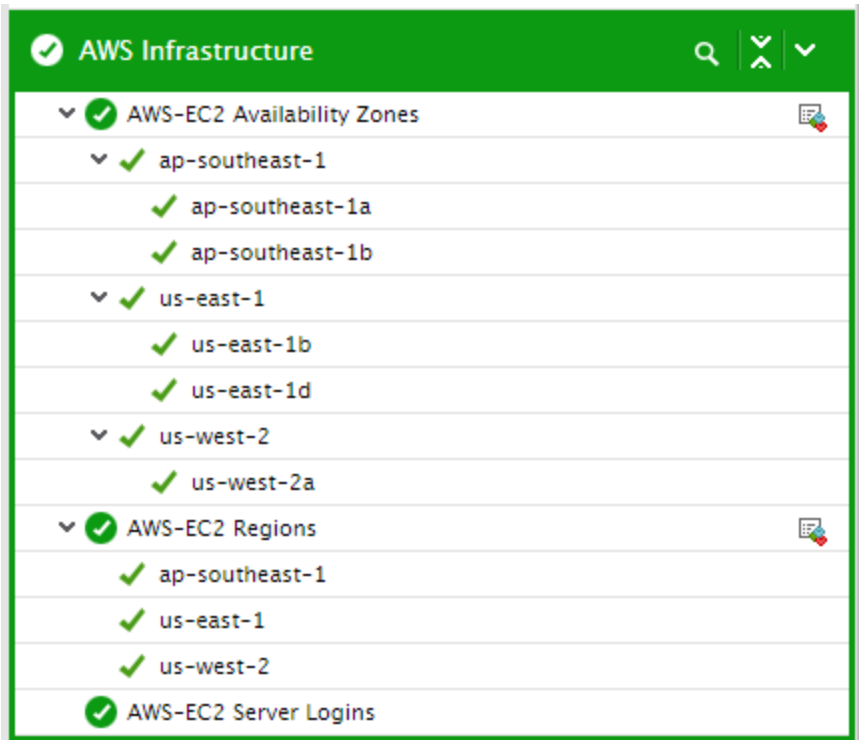


Figure 4.2: The test associated with the AWS Infrastructure layer

4.1.1 AWS-EC2 Web Access Test

This test emulates a user accessing a web page on the cloud via HTTP(S), and reports whether that page is accessible or not. In the process, the test indicates the availability of the cloud over the web, and the time it took for the agent to access the cloud over the web. This way, issues in web-based access to the cloud come to light.

Target of the test : Amazon EC2 Cloud

Agent deploying the test : A remote agent

Output of the test : One set

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured

Parameter	Description
Port	The port to which the specified **HOST** listens
URL	The web page being accessed. While multiple URLs (separated by commas) can be provided, each URL should be of the format URL name:URL value. URL name is a unique name assigned to the URL, and the URL value is the value of the URL. By default, the url parameter is set to *HomePage:http://aws.amazon.com/ec2/*, where *HomePage* is the *URL name*, and *http://aws.amazon.com/ec2* is the *URL value*. You can modify this default setting to configure any URL of your choice - eg., the URL of the login page to your cloud-based infrastructure.
Cookie File	Whether any cookies being returned by the web server need to be saved locally and returned with subsequent requests
Proxy Host and Proxy Port	The host on which a web proxy server is running (in case a proxy server is to be used), and the port at which the web proxy server listens
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
Content	Is a set of instruction:value pairs that are used to validate the content being returned by the test. If the **CONTENT** value is *none:none*, no validation is performed. The number of pairs specified in this text box, must be equal to the number of URLs being monitored. The instruction should be one of *Inc* or *Exc*. Inc tells the test that for the content returned by the test to be valid, the content must include the specified value (a simple string search is done in this case). An instruction of *Exc* instructs the test that the test's output is valid if it does not contain the specified value. In both cases, the content specification can include wild card patterns. For example, an Inc instruction can be *Inc:*Home page**. An *Inc* and an *Exc* instruction can be provided in quick succession in the following format: *Inc:*Home Page*,Exc:*home*.
Credentials	The HttpTest supports HTTP authentication. The **CREDENTIALS** parameter is to be set if a specific user name / password has to be specified to login to a page. Against this parameter, the *URLname* of every configured URL will be displayed; corresponding to each listed *URLname*, a **Username** text box and a **Password** text box will be made available. These parameters will take either of the following values: • valid **Username** and **Password** for every configured *URLname* • *none* in both the **Username** and **Password** text boxes of all configured *URLnames* (the default setting), if no user authorization is required

Parameter	Description
	Where NTLM (Integrated Windows) authentication is supported, valid **CREDENTIALS** are mandatory. In other words, a none specification will not be supported in such cases. Therefore, in this case, against each configured *URLname*, you will have to provide a valid Username in the format: *domainname**username*, followed by a valid **Password**. Please be sure to check if your web site requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many sites use HTTP POST for obtaining the user name and password and validating the user login. In such cases, the username and password have to be provided as part of the POST information and NOT as part of the **CREDENTIALS** specification for the this test.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Timeout	Here, specify the maximum duration (in seconds) for which the test will wait for a response from the server. The default **TIMEOUT** period is 30 seconds.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Availability:	Indicates whether the test was able to access the configured URL or not	Percent	Availability failures could be caused by several factors such as the web server process(es) (hosting the configured web page) being down, the web server being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web server is overloaded. Availability is determined based on the response code returned by the test. A response code between 200 to 300 indicates that the configured web page is available.
Total response time:	Indicates the time taken by the test to access this URL	Secs	Response time being high denotes a problem. Poor response times may be due to an overload. If the URL accessed involves

Measurement	Description	Measurement Unit	Interpretation
			the generation of dynamic content, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.
Tcp connection availability:	Indicates whether the test managed to establish a TCP connection to this URL.	Percent	Failure to establish a TCP connection may imply that either the web server process hosting the web page is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the web page may start functioning properly again.
Tcp connect time:	Quantifies the time for establishing a TCP connection to the configured URL.	Secs	Typically, the TCP connection establishment must be very small (of the order of a few milliseconds).
Server response time:	Indicates the time period between when the connection was established and when the test sent back a HTTP response header to the client.	Secs	While the total response time may depend on several factors, the server response time is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
Response code:	Returned by the test for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
Content length:	The size of the content returned by the test.	Kbytes	Typically the content length returned by the test for a specific URL should be the same across time. Any change in this metric may indicate the need for further investigation.
Content validity:	Validates whether the test was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content may not be valid.

Measurement	Description	Measurement Unit	Interpretation
			This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, "Invalid Login" is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is configured such that the content parameter should exclude the string "Invalid Login," in the above scenario content validity would have a value 0.

4.1.2 AWS-EC2 Availability Zones Test

Amazon has data centers in different areas of the world (e.g., North America, Europe, Asia, etc.). Correspondingly, EC2 is available to use in different *Regions*. Each Region contains multiple distinct locations called *Availability Zones* (illustrated in the following diagram). Each Availability Zone is engineered to be isolated from failures in other Availability zones and to provide inexpensive, low-latency network connectivity to other zones in the same Region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.

If users complaint that their server instances are inaccessible, you may want to know whether it is because of the non-availability of the availability zone within which the instances have been launched. This test auto-discovers the regions and availability zones on the Amazon EC2 Cloud, and reports the availability of each zone.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each availability zone in each region of the AWS EC2 Cloud being monitored

First-level descriptor: AWS Region

Second-level descriptor: Availability zone

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Cloudwatch Enabled	**This parameter only applies to the AWS - EC2 Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource

Parameter	Description
	usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.
Report Instance Datacenter	By default, this test reports the availability of only those availability zones that contain one/more instances. Accordingly, this flag is set to **Yes** by default. If you want the test to report metrics for all availability zones, regardless of whether/not they host instances, set this flag to **No**.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Availability:	Indicates whether/not this availability zone in this region is currently available.	Number	The value *0* indicates that the availability zone is *Not Available* and the value *100* indicates that it is *Available*. If an availability zone fails, then all server instances operating within that zone will also be rendered unavailable. If you host all your Amazon EC2 instances in a single location that is affected by such a failure, your instances will be unavailable, thereby bringing your entire application to a halt. On the other hand, if you have instances distributed across many Availability Zones and one of the instances fails, you can design your application so the instances in the remaining Availability Zones handle any requests.

4.1.3 AWS-EC2 Regions Test

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and Regions. Regions are dispersed and located in separate geographic areas (US, EU, etc.). Each Region is completely independent.

By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements.

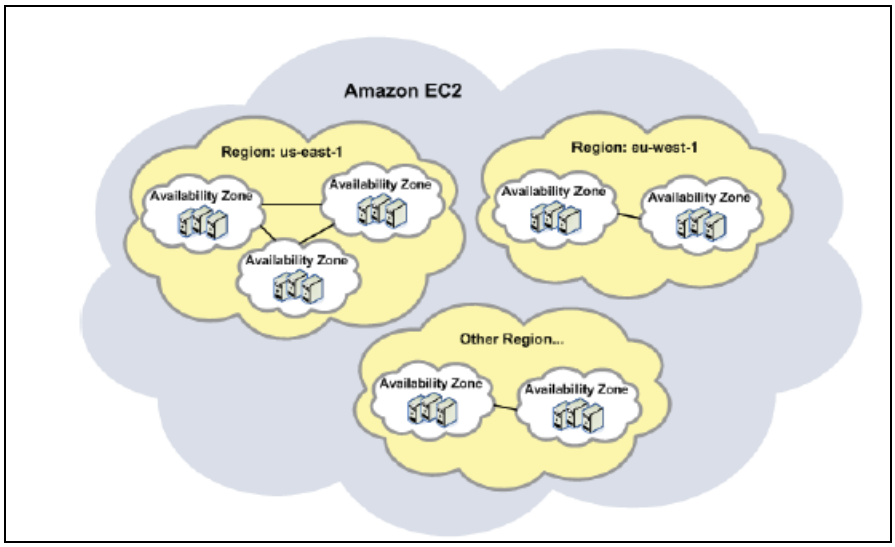


Figure 4.3: Regions and Availability zones

If a region is unavailable, then users to that region will not be able to access the server instances launched in that region. This may, in turn, adversely impact the user experience with the cloud. To avoid such an unpleasant outcome, it is best to periodically monitor the availability of each region, so that unavailable regions can be quickly and accurately identified, and the reasons for their non-availability remedied.

This test performs periodic availability checks on each region on the cloud, and reports the status of the individual regions. In addition, the test also indicates the time taken for connecting to a region so that, regions with connectivity issues can be isolated.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each region of the AWS EC2 Cloud being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Cloudwatch Enabled	**This parameter only applies to the AWS - EC2 Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource

Parameter	Description
	usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.
Report Instance Region	By default, this test reports metrics for only those regions that host at least one instance. This is why, the **REPORT INSTANCE REGION** flag is set to **Yes** by default. If you want, you can configure this test to report metrics for all regions, regardless of whether/not they host any instances. For this, set this flag to **No**.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Availability:	Indicates whether/not this region is currently available.	Number	The value 0 indicates that the region is Not Available and the value 100 indicates that it is Available.
Response time:	Indicates the time taken to connect to this region.	Secs	A low value is typically desired for this measure. A high value or a consistent increase in this value could be indicative of connection bottlenecks. Compare the value of this measure across regions to know which region takes the longest to connect to.

4.1.4 AWS-EC2 Server Logins Test

This test attempts to connect to the default region in the cloud; in the process, the test reports whether the configured AWS user account is able to access the cloud-based infrastructure or not,

and if so, how quickly the connection with the infrastructure was established.

If a user is denied access to a server instance on a clod, or if a user experiences a significant delay in connecting to his/her instances, you can use this test to validate the user credentials and to figure out whether any connectivity issues exist.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for the AWS EC2 Cloud being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name

Parameter	Description
	required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Cloudwatch Enabled	**This parameter only applies to the AWS - EC2 Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Default region availability:	Indicates whether or not the test is able to access the default region on the cloud using the configured AWS user account .	Percent	The value 0 indicates that the region is not accessible, and the value 100 indicates that it is accessible. If the default region is inaccessible, it could be owing to any one of the following reasons: • The cloud is unavailable; • The configured AWS account does not have the access rights to the default

Measurement	Description	Measurement Unit	Interpretation
			region; • The test has been configured with incorrect login credentials.
Response time:	Indicates the time taken by the test to establish a connection with the default region on the cloud.	Secs	A low value is desired for this measure. A high value or a consistent increase in this value could indicate connection bottlenecks.

4.2 The AWS Network Layer

With the help of the tests mapped to this layer, administrators can:

- Proactively detect bottlenecks to successful delivery of web content to users;
- Monitor Route53 health checks, and accurately pinpoint failed health checks;
- Monitor the operational state of and traffic flowing into/out of each VPN that connects a VPC with your network

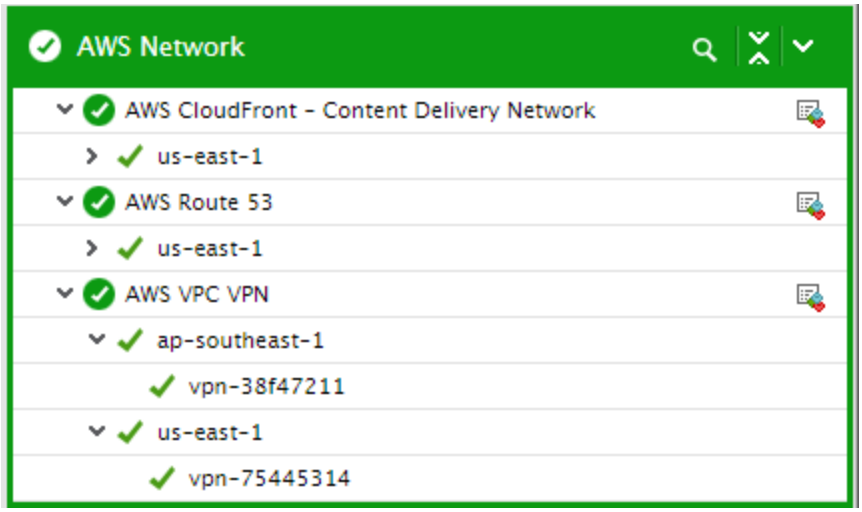


Figure 4.4: The tests mapped to the AWS Network layer

4.2.1 AWS CloudFront - Content Delivery Network Test

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. CloudFront delivers your content through a worldwide network of data centers called edge locations. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance.

In the edge location, CloudFront checks its cache for the requested files. If the files are in the cache, CloudFront returns them to the user. If the files are not in the cache, it does the following:

1. CloudFront compares the request with the specifications in your distribution. A distribution is where you can specify configuration settings such as:

   - Your origin, which is the Amazon S3 bucket or HTTP server from which CloudFront gets the files that it distributes. You can specify any combination of up to 25 Amazon S3 buckets and/or HTTP servers as your origins.
   - Whether you want the files to be available to everyone or you want to restrict access to selected users.
   - Whether you want CloudFront to require users to use HTTPS to access your content.
   - Whether you want CloudFront to forward cookies and/or query strings to your origin.
   - Whether you want CloudFront to prevent users in selected countries from accessing your content.
   - Whether you want CloudFront to create access logs.

   From the distribution, CloudFront determines the origin server that applies to the requested file type and forwards the request to that server.

2. The origin servers then send the files back to the CloudFront edge location.
3. As soon as the first byte arrives from the origin, CloudFront begins to forward the files to the user. CloudFront also adds the files to the cache in the edge location for the next time someone requests those files.

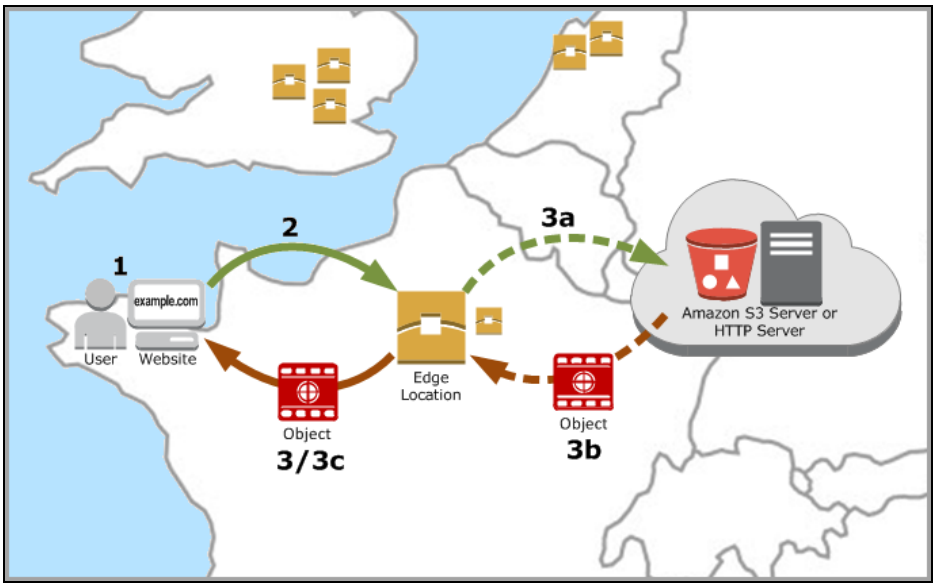


Figure 4.5: How CloudFront delivers content

The success of CloudFront relies on the successful delivery of content to users. If errors in request processing go undetected, it can cause content to not be delivered to the intended audience. This is bound to adversely impact user confidence in CloudFront! To avoid this, administrators should be able to promptly detect errors in request processing, rapidly investigate the reason for the errors, and quickly resolve it. This is where the **AWS CloudFront - Content Delivery Network** test helps.

This test auto-discovers the distributions configured on CloudFront and tracks the requests to and responses of origin servers specified in each distribution. In the process, the test promptly captures HTTP error responses from origin servers, and instantly notifies administrators of the errors. This way, the test pinpoints the distribution that is configured with the origin servers emitting the maximum number of error responses. Administrators can then closely scrutinize such a distribution for any misconfiguration.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each distribution configured on CloudFront

First-level descriptor: AWS Region

Second-level descriptor: DistributionID

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total request	Indicates the number of HTTP and HTTPS	Number	

Measurement	Description	Measurement Unit	Interpretation
	requests (for all HTTP methods) to the origin servers specified in this distribution.		
Data downloaded	Indicates the amount of data downloaded by viewers for GET, HEAD, and OPTIONS requests to the origin server specified in this distribution.	KB	
Data uploaded	Indicates the amount of data uploaded to the origin servers specified in this distribution, using POST and PUT requests.	KB	
Total errors	Indicates what percentage of requests to the origin servers in this distribution returned HTTP error response codes such as 4xx or 5xx.	Percent	Ideally, the value of this measure should be 0. A non-zero value indicates that an HTTP error has occurred. Compare the value of this measure across distributions to know which distribution is configured with origin servers that have returned the maximum HTTP error responses. You may want to take another look at such distributions to find misconfigurations (if any).
HTTP 4xx errors	Indicates what percentage of requests to the origin servers in this distribution returned HTTP error response code 4xx.	Percent	If the value of the *Total error* measure is abnormally high for a distribution, then you can compare the value of these two measures for that distribution to know what type of HTTP errors were common. • **HTTP 4xx:** This class of status code is intended for situations in which the error

Measurement	Description	Measurement Unit	Interpretation
			seems to have been caused by the client. • **HTTP 5xx:** Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request.
HTT 5xx errors	Indicates what percentage of requests to the origin servers in this distribution returned HTTP error response code 5xx.	Percent	

4.2.2 AWS Route53 Test

Using Amazon Route 53, one can get a website or web application up and running. One of the key functions of Route53 is that it sends automated requests over the internet to a resource, such as a web server or an email server, to verify that it's reachable, available, and functional. You also can choose to receive notifications when a resource becomes unavailable and choose to route internet traffic away from unhealthy resources.

Here's an overview of how health checking works:

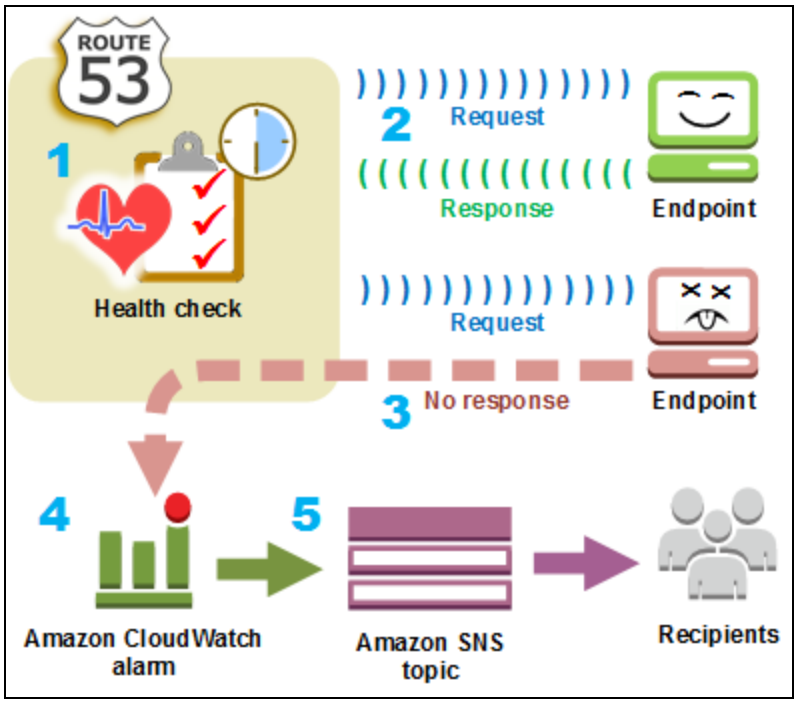


Figure 4.6: How Amazon Route53 health checks work

1. You create a health check and specify values that define how you want the health check to work.
2. Route 53 starts to send requests to the endpoint at the interval that you specified in the health check.
3. If the endpoint responds to the requests, Route 53 considers the endpoint to be healthy and takes no action.
4. If the endpoint does not respond to a request, Route 53 starts to count the number of consecutive requests that the endpoint does not respond to.
5. If the count reaches the value that you specified for the failure threshold, Route 53 considers the endpoint unhealthy.
6. If the endpoint starts to respond again before the count reaches the failure threshold, Route 53 resets the count to 0.

If a health check fails, then administrators should be promptly notified of it, so that they can investigate the reasons for the failure and initiate relevant appropriate remedial measures. This is where the AWS Route53 test will be most useful!

This test discovers all the health checks configured for each AWS region and alerts administrators if any health check fails. In order to help administrators determine how often during a given measure

period a health check reported abnormalities with an endpoint, the test also reports the percentage of time for which each health check reported that its endpoint is healthy. This introduces administrators to problem-prone areas of their infrastructure.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each health check ID in each AWS region

First-level descriptor: AWS Region

Second-level descriptor: Health check ID

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy User Name and Proxy Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region

Parameter	Description
	names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Status of health check	Indicates the status of the endpoint of this health check.		If the health check reports that the endpoint is in an healthy state, then the value of this measure is Healthy. If the health check finds that an endpoint is in an abnormal state, the value of this measure Unhealthy. The numeric values that correspond to these measure values are as follows: **Measure Value** / **Numeric Value** Healthy / 0 Unhealthy / 1 **Note:** By default, the test reports the **Measure Value**s in the table above to indicate the status of a health check. In the graph of this measure however, the status is indicated using numeric equivalents only.
Health check percentage	Indicates the percentage of time this health check reported that an endpoint is healthy.	Percent	A very low value for this measure is indicative of problem-prone endpoints.

4.2.3 AWS VPC VPN Test

Amazon VPC (Virtual Private Cloud) lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address ranges, creation of subnets, and configuration of route tables and network gateways.

By default, instances that you launch into a virtual private cloud (VPC) cannot communicate with your own network. You can enable access to your network from your VPC by attaching a virtual private gateway to the VPC, creating a custom route table, updating your security group rules, and creating an AWS managed VPN connection. A VPN connection refers to the connection between your VPC and your own network. Each VPN connection has two tunnels, with each tunnel using a unique virtual private gateway public IP address. It is important to configure both tunnels for redundancy. When one tunnel becomes unavailable (for example, down for maintenance), network traffic is automatically routed to the available tunnel for that specific VPN connection.

To ensure continuous communication between the VPC and the network therefore, administrators should track the status (up/down) of both tunnels and make sure that at least one tunnel is up and running at all times. Since a VPN tunnel comes up only when traffic is generated from the customer-side of the VPN connection, administrators must keep an eye on the incoming and outgoing traffic on each tunnel to determine whether the absence of traffic is what caused a tunnel to go down. To quickly detect that a tunnel is down and to rapidly diagnose its root-cause, administrators can use the **AWS VPC VPN** Test.

For each VPN tunnel configured for the AWS VPC, this test reports the status of that tunnel and the amount of traffic flowing in the tunnel. This way, the test alerts administrators when a tunnel goes down or is idle.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each VPN tunnel.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.

Parameter	Description
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
VPN Filter Name	By default, this parameter is set to **Vpn Id**. In this case, the test will report metrics for each VPN. If required, you can override this default setting by picking the **Tunnel Ip Address** option from this drop-down. In this case, the test will report metrics for each tunnel in each VPN.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Tunnel state</td><td>By default, this test reports the current state of the tunnels in this VPN.

If the VPN Filter Name is set to Tunnel Ip Address, then this measure will report the state of this tunnel.</td><td></td><td>The values that this measure reports and their corresponding numeric values are listed in the table below:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Up</td><td>1</td></tr><tr><td>Down</td><td>0</td></tr></table>
If both the tunnels in a VPN are down, then this measure will report the value Down for that VPN. If both the tunnels in a VPN are up and running, or if only one of the tunnels is up, then this measure will report the value Up for that VPN. In this case, you can configure the VPN Filter Name parameter of the test to Tunnel Ip Address and determine which tunnel in the VPN is down.

Typically, a VPN tunnel comes up when traffic is generated from your side of the VPN connection. The virtual private gateway is not the initiator; your customer gateway must initiate the tunnels. If your VPN connection experiences a period of idle time (usually 10 seconds, depending on your configuration), the tunnel may go down. To prevent this, you can use a network monitoring tool to generate keepalive pings; for example, by using IP SLA.

Note:

By default, this test uses the</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			Measure Values listed in the table above to indicate the state of a tunnel. In the graph of this measure however, the same is indicated using the numeric equivalents only.
Tunnel incoming data	By default, this test indicates the total amount of data received through both the VPN tunnels in this VPN. If the VPN Filter Name is set to **Tunnel Ip Address**, then this measure will report the amount of data received through this tunnel.	KB	This metric counts the data after decryption. If a VPN tunnel goes down very often, you may want to check if the value of the 'Tunnel incoming data' and 'Tunnel outgoing data' measures is consistently 0 for that tunnel. Lack of traffic on the tunnels is a common reason for VPN tunnels to fail. There is a vendor-specific VPN idle time for policy based VPN connections. If there is no traffic through the VPN tunnel for that duration, the IPsec session can be torn down. For tunnels going down due to idle timeout, be sure there is constant bidirectional traffic between your local network and VPC. Consider setting up a host that sends one ICMP requests every 5 seconds to an instance in the VPC that responds to ICMP. This allows the tunnel to stay up as it continues to respond to the ICMP requests, and makes sure that there are packets being encrypted and decrypted across the tunnel.
Tunnel outgoing data	By default, this test indicates the total amount of data sent through both the VPN tunnels in this VPN. If the VPN Filter Name is	KB	This metric counts the data after encryption. If a VPN tunnel goes down very often, you may want to check if the value of the 'Tunnel incoming data' and 'Tunnel outgoing data' measures is

Measurement	Description	Measurement Unit	Interpretation
	set to **Tunnel Ip Address**, then this measure will report the amount of data sent through this tunnel.		consistently 0 for that tunnel. Lack of traffic on the tunnels is a common reason for VPN tunnels to fail. There is a vendor-specific VPN idle time for policy based VPN connections. If there is no traffic through the VPN tunnel for that duration, the IPsec session can be torn down. For tunnels going down due to idle timeout, be sure there is constant bidirectional traffic between your local network and VPC. Consider setting up a host that sends one ICMP requests every 5 seconds to an instance in the VPC that responds to ICMP. This allows the tunnel to stay up as it continues to respond to the ICMP requests, and makes sure that there are packets being encrypted and decrypted across the tunnel.

4.3 The AWS Storage Layer

The tests mapped to this layer measure the usage and overall efficiency of the different storage services that are available to instances launched on the AWS cloud. The services include:

- AWS Elastic Block Store
- AWS Elastic File System
- AWS Simple Storage Service

Additionally, the layer also monitors the AWS Storage Gateways, using which on-premises software appliances connect with cloud-based storage, and reveals I/O processing bottlenecks and resource contentions experienced by the gateways.

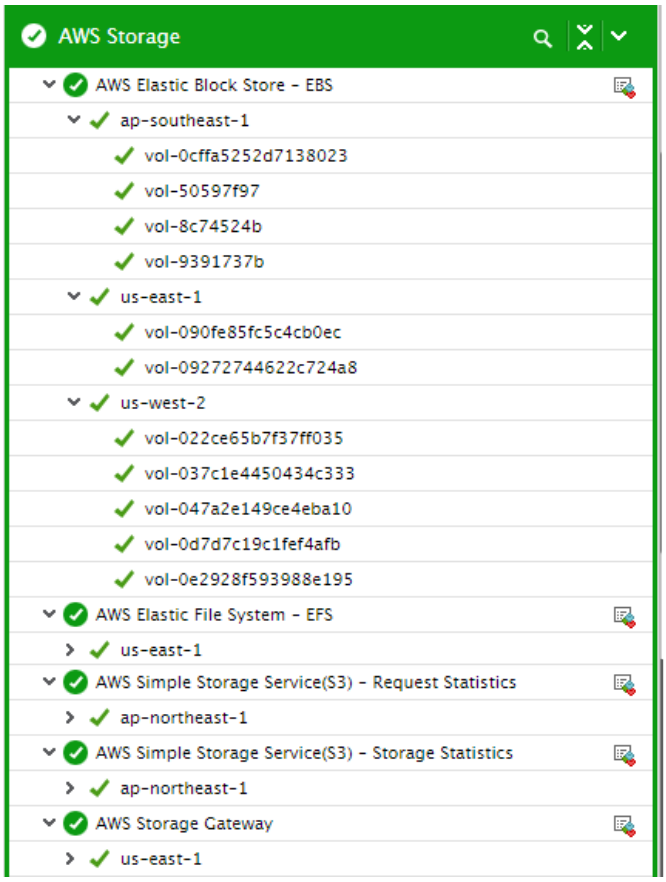


Figure 4.7: The tests mapped to the AWS Storage layer

4.3.1 AWS Elastic Block Store - EBS Test

Amazon Elastic Block Store (Amazon EBS) provides persistent block level storage volumes for use with Amazon EC2 instances in the AWS Cloud. An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as system drive for an instance or storage for a database application. If such an EBS volume suddenly becomes unavailable or impaired, it is bound to adversely impact the operations of the EC2 instance attached to that volume, which in turn will damage the experience of the users of that instance. Administrators need to be promptly alerted to such problem conditions, so that they can instantly initiate remedial action and ensure high instance uptime. Besides volume status, administrators also need to track the I/O load on the EBS volume and continuously measure the ability of the volume to handle that load. This insight will enable administrators to provision the volumes with more or less I/O, so as to optimize I/O processing and maximize volume performance. The AWS Elastic Block Store - EBS test helps administrators in this exercise. The test periodically checks the health and availability status of each volume used by the EC2 instances in every region of the AWS EC2 cloud and notifies administrators if any volume is in

an abnormal state. Similarly, the test also tracks the I/O load on every volume and measures how well each volume processes the load - overloaded volumes and those that are experiencing processing hiccups are highlighted in the process.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each volume in each region of the AWS cloud being monitored

First-level descriptor: AWS EC2 region name

Second-level descriptor: EBS volume name

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.

Parameter	Description
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
State	Indicates the current state of this volume.		The values that this measure can report and their corresponding numeric values are detailed in the table below:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td>Creating</td><td>The volume is being created. The volume will be inaccessible during creation.</td><td>0</td></tr>
<tr><td>Available</td><td>The volume is available</td><td>1</td></tr>
<tr><td>In-use</td><td>The volume is in use</td><td>2</td></tr>
<tr><td>Deleting</td><td>The volume is being deleted</td><td>3</td></tr>
<tr><td>Deleted</td><td>The volume is deleted</td><td>4</td></tr>
<tr><td>Error</td><td>Some error has occurred in the volume</td><td>5</td></tr>
</table>

The detailed diagnosis of this measure will reveal when the volume was created and in which availability zone it resides.

Note:

By default, this measure will report the Measure Values listed in the table above to indicate the current availability state of a volume. In the graph of this measure however, the same will be represented using the numeric equivalents only.

If any EBS volume is found to be in an abnormal state, then you can use the detailed diagnosis of this measure to know the volume type, when that volume was created, and in which availability zone the volume resides.
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Status</td><td>Indicates the current health status of this volume</td><td></td><td>AWS EC2 periodically runs volume status checks to enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume.

Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. The value that this measure reports varies with the status reported by the volume status checks. The table below describes what value this measure reports when , and also lists the numeric values that correspond to the measure values.

<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td>OK</td><td>If all checks pass, the status of the volume is OK.</td><td>0</td></tr>
<tr><td>Impaired</td><td>If a check fails, the status of the volume is impaired</td><td>1</td></tr>
<tr><td>Insufficient-data</td><td>If checks are in progress, then insufficient-data is reported</td><td>2</td></tr>
</table>

Note:

By default, this measure will report the Measure Values listed in the table above to indicate the current status of a volume. In the graph of this measure however, the same will be represented using the numeric equivalents only.</td></tr>
<tr><td>Idle time:</td><td>Indicates the total</td><td>Secs</td><td></td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
	number of seconds during which no read or write operations were submitted to this volume.		
Queue length:	Indicates the number of read and write operation requests waiting to be completed.	Number	A consistent increase in the value of this measure could indicate a I/O processing bottleneck on the volume.
Read operations:	Indicates the rate at which read operations were performed on this volume.	Operations/Sec	Compare the value of this measure across volumes to know which volume is too slow in processing read requests.
Write operations:	Indicates the rate at which write operations were performed on this volume.	Operations/Sec	Compare the value of this measure across volumes to know which volume is too slow in processing write requests.
Reads:	Indicates the rate at which data was read from this volume.	KB/Sec	Compare the value of this measure to identify the volume that is the slowest in responding to read requests.
Writes:	Indicates the rate at which data was written to this volume.	KB/Sec	Compare the value of this measure to identify the volume that is the slowest in responding to write requests.
Total read time:	Indicates the total time taken by all completed read operations.	Secs	A very high value for this measure could indicate that the volume took too long to service one/more read requests.
Total write time:	Indicates the total time taken by all completed write operations.	Secs	A very high value for this measure could indicate that the volume took too long to service one/more write requests.
Provisioned IOPS	Indicates the	Percent	**This measure will be reported for Provisioned IOPS volumes only.**

Measurement	Description	Measurement Unit	Interpretation
(SSD)volume throughput:	percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for this volume.		Provisioned IOPS (SSD) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency in random access I/O throughput. You specify an IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year. A Provisioned IOPS (SSD) volume can range in size from 4 GiB to 16 TiB and you can provision up to 20,000 IOPS per volume. The ratio of IOPS provisioned to the volume size requested can be a maximum of 30; for example, a volume with 3,000 IOPS must be at least 100 GiB. You can stripe multiple volumes together in a RAID configuration for larger size and greater performance. For smaller I/O operations, you may even see an IOPS value that is higher than what you have provisioned - i.e., the value of this measure can be greater than 100%. This could be because the client operating system may be coalescing multiple smaller I/O operations into a smaller number of large chunks. On the other hand, if the value of this measure is consistently lower than the expected IOPS or throughput you have provisioned, then ensure that your EC2 bandwidth is not the limiting factor; your instance should be EBS-optimized (or include 10 Gigabit network connectivity) and your instance type EBS dedicated bandwidth should exceed the I/O throughput you intend to drive. Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes.

Measurement	Description	Measurement Unit	Interpretation
Total IOPS for provisioned IOPS volume:	Indicates the total amount of read and write operations (normalized to 256K capacity units) consumed by this volume in a specified period of time.	Number	**This measure will be reported for Provisioned IOPS volumes only.** I/O operations that are smaller than 256K each count as 1 consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as 4 consumed IOPS.
Size:	Indicates the current size of this volume.	GB	For a General Purpose (SSD) Volume, volume size is what dictates the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. For a Provisioned IOPS (SSD) Volume, the ratio of IOPS provisioned to volume size can be a maximum of 30; for example, a volume with 3,000 IOPS must be at least 100 GiB. Magnetic volumes can range in size from 1 GiB to 1 TiB.
Total IOPS:	Indicates the total number of I/O operations that were performed on this volume per second.	Operations/Sec	IOPS are input/output operations per second. Amazon EBS measures each I/O operation per second (that is 256 KiB or smaller) as one IOPS. I/O operations that are larger than 256 KiB are counted in 256 KiB capacity units. For example, a single 1,024 KiB I/O operation would count as 4 IOPS; however, 1,024 I/O operations at 1 KiB each would count as 1,024 IOPS. When you create a 3,000 IOPS volume, either a 3,000 IOPS Provisioned IOPS (SSD) volume or a 1,000 GiB General Purpose (SSD) volume, and attach it to an EBS-optimized instance that can provide the necessary bandwidth, you can transfer up to 3,000 chunks of data per second (provided that the

Measurement	Description	Measurement Unit	Interpretation
			I/O does not exceed the per volume throughput limit of the volume). If your I/O chunks are very large, then the value of this measure may be lesser than what you provisioned because you are hitting the throughput limit of the volume. For example 1,000 GiB General Purpose (SSD) volume has an IOPS limit of 3,000 and a volume throughput limit of 160 MiB/s. If you are using a 256 KiB I/O size, your volume will reach its throughput limit at 640 IOPS (640 x 256 KiB = 160 MiB). For smaller I/O sizes (such as 16 KiB), this same volume can sustain 3,000 IOPS because the throughput is well below 160 MiB/s. On Provisioned IOPS Volumes, for smaller I/O operations, you may even see that the value of this measure is higher than what you have provisioned. This could be because the client operating system may be coalescing multiple smaller I/O operations into a smaller number of large chunks. On the other hand, if the value of this measure is consistently lower than the expected IOPS or throughput you have provisioned for a Provisioned IOPS volume, then ensure that your EC2 bandwidth is not the limiting factor; your instance should be EBS-optimized (or include 10 Gigabit network connectivity) and your instance type EBS dedicated bandwidth should exceed the I/O throughput you intend to drive. Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes. Magnetic volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS.

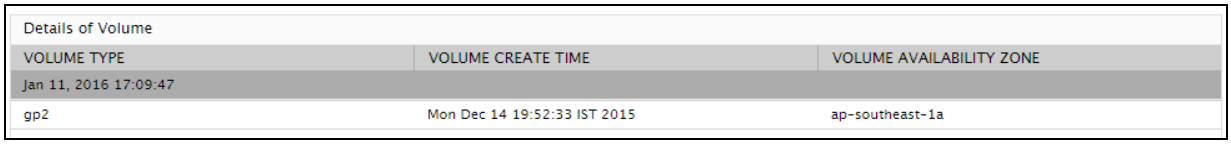
Measurement	Description	Measurement Unit	Interpretation
IOPS limits:	Indicates the IOPS limit of this volume.	Operations/Sec	For Provisioned IOPS volumes, the IOPS limit is specified when creating the volumes. For General Purpose IOPS volumes, the volume size dictates the baseline IOPS limit of that volume and how quickly it accumulates I/O credits.
IOPS utilization:	Indicates the percentage of provisioned IOPS or IOPS limit that is being utilized by this volume.	Percent	This metric can also help you identify over-utilized volumes, which could be impacting application performance. In these cases, you could improve performance by upgrading to a different volume type or provisioning more IOPS.
Throughput:	Indicates the rate of reads and writes processed by this volume.	KB/Second	A consistent drop in this value could indicate a I/O processing bottleneck on the volume. You may want to closely track the variations to this measure, so that you can proactively identify the volume that may soon reach its throughput limit. The maximum throughput of each volume type is indicated below: • General purpose volumes - 160 MiB/sec • Provisioned IOPS volumes - 320 MiB/sec • Magnetic volumes - 40-90 MiB/sec If your I/O chunks are very large, then a volume will reach its throughput limit much before its IOPS limit is reached. If you are not experiencing the throughput you have provisioned, ensure that your EC2 bandwidth is not the limiting factor; your instance should be EBS-optimized (or include 10 Gigabit network connectivity) and your instance type EBS dedicated bandwidth

Measurement	Description	Measurement Unit	Interpretation
			should exceed the I/O throughput you intend to drive.
Burst balance	Indicates the percentage of I/O credits (for gp2) or throughput credits (for st1 and sc1) remaining in the burst bucket for this volume.	Percent	**This measure is applicable to General Purpose SSD (gp2), Throughput Optimized HDD (st1), and Cold HDD (sc1) volumes only.** The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your gp2 volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed. Each gp2 volume receives an initial I/O credit balance of 5.4 million I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes. Volumes earn I/O credits at the baseline performance rate of 3 IOPS per GiB of volume size. When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance. When your volume requires more than the baseline performance I/O level, it draws on I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. This means that for a gp2 volume to burst performance levels above its baseline, a high I/O credit balance is necessary. This implies that the value of this measure should ideally be high for a gp2

Measurement	Description	Measurement Unit	Interpretation
			volume. If your gp2 volume uses all of its I/O credit balance - i.e., if the value of this measure is 0 or very low for a gp2 volume - then the maximum IOPS performance of the volume remains at the baseline IOPS performance level (the rate at which your volume earns credits) and the volume's maximum throughput is reduced to the baseline IOPS multiplied by the maximum I/O size. When I/O demands rise above the baseline performance level of the volume, the volume will be unable to meet with the demand owing to the lack of adequate I/O credits. Like gp2, st1 and sc1 volumes use a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it can drive I/O at the burst level. For peak performance of st1 and sc1 volumes therefore, the value of this measure should be high ideally. After the bucket is depleted, throughput is limited to the baseline rate of 12 MiB/s per TiB.
Attached instances	Indicates the number of instances attached to this volume.	Number	Use the detailed diagnosis of this measure to know which instances are attached to the volume, and which region each instance belongs to.

Detailed Diagnosis:

The detailed diagnosis of the **State** measure of a volume will reveal when the volume was created and in which availability zone it resides.



Details of Volume

VOLUME TYPE	VOLUME CREATE TIME	VOLUME AVAILABILITY ZONE
Jan 11, 2016 17:09:47		
gp2	Mon Dec 14 19:52:33 IST 2015	ap-southeast-1a

Figure 4.8: The detailed diagnosis of the State measure of the AWS Elastic Block Store - EBS Test

4.3.2 AWS Elastic File System - EFS Test

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2.

Using Amazon EFS, you can create a file system, mount the file system on an Amazon EC2 instance, and then read and write data from and to your file system. Since the storage capacity is elastic, EFS can grow and shrink the storage automatically as you add and remove files, so your applications have the storage they need, when they need it. Moreover, as the file systems are distributed across an unconstrained number of storage servers, they allow massively parallel access from Amazon EC2 instances to data.

The elasticity and distributed storage design are reasons why Amazon EFS is widely used to support mission-critical workloads requiring substantial levels of aggregate throughput and I/O processing power. If any file system is unable to meet with the dynamic throughput and I/O demands of such applications, the performance of the file system and the dependent applications will be adversely impacted, causing user experience with EFS to suffer and revenues to drop. To avoid this, administrators should continuously monitor the load on each file system, measure throughput and I/O processing power of every file system, and proactively detect if throughput and IOPS of a file system fall below the established baseline. This is where the AWS Elastic File System - EFS Test helps!

This test automatically discovers the file systems created on AWS, and reports the throughput, number of client connections, and the number of bytes for read, write, and metadata operations on every file system. In the process, the test pinpoints those file systems with a high workload in terms of connections and I/O operations, and those that do not have adequate throughput in reserve to handle its load. File-system configuration can be fine-tuned based on pointers provided by this test.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each file system

First-level descriptor: AWS Region

Second-level descriptor: File system ID

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Burst credit balance	Indicates the average size of burst credits that this file system has during the measure period.	KB	Throughput on Amazon EFS scales as a file system grows. Because file-based workloads are typically spiky—driving high levels of throughput for short periods of time, and low levels of throughput the rest of the time - Amazon EFS is designed to burst to high throughput levels for periods of time. All file systems, regardless of size, can burst to 100 MiB/s of throughput, and those over 1 TiB large can burst to 100 MiB/s per TiB of data stored in the file system. For example, a 10 TiB file system can burst to 1,000 MiB/s of throughput (10 TiB x 100 MiB/s/TiB). The portion of time a file system can burst is determined by its size, and the bursting model is designed so that typical file system workloads will be able to burst virtually any time they need to. Amazon EFS uses a credit system to determine when file systems can burst. Each file system earns credits over time at a baseline rate that is determined by the size of the file system, and uses credits whenever it reads or writes data. The baseline rate is 50 MiB/s per TiB of storage (equivalently, 50 KiB/s per GiB of storage). Accumulated burst credits give the file system permission to drive throughput above its baseline rate. . A file system can drive throughput

Measurement	Description	Measurement Unit	Interpretation
			continuously at its baseline rate, and whenever it's inactive or driving throughput below its baseline rate, the file system accumulates burst credits. For example, a 100 GiB file system can burst (at 100 MiB/s) for 5 percent of the time if it's inactive for the remaining 95 percent. Over a 24-hour period, the file system earns 432,000 MiBs worth of credit, which can be used to burst at 100 MiB/s for 72 minutes. File systems larger than 1 TiB can always burst for up to 50 percent of the time if they are inactive for the remaining 50 percent. A high value is desired for this measure, as it implies that the file system has enough credits for use during periods of high workload. It also means that the file system has been relatively inactive lately. A low value or a consistent drop in the value of this measure implies that the credits have been steadily utilized to service workloads, leaving the file system with very few credits for the future.
Client connections	Indicates the number of client connections to this file system.	Number	This is a good indicator of the workload of a file system.
Data associated with read operations	Indicates the amount of data that was read from this file system, on an average.	KB	These are good indicators of the I/O load on a file system. If the value of one/more of these measures is very high and the value of the Burst credit balance measure is very low, it can imply high workload and excessive usage of burst credits

Measurement	Description	Measurement Unit	Interpretation
			for servicing the workload. In such a circumstance, you can compare the value of these measures for that file system to know what is contributing to the load - read operations? write operations? or metadata operations?
Data associated with write operations	Indicates the average amount of data for this file system's write operations.	KB	
Data associated with metadata operations	Indicates the amount of data for this file system's metadata operations.	KB	
Data associated with all file operations	Indicates the total amount of data for this file system's I/O operations.	KB	
File system to reaching the I/O limits	Indicates how close this file system is to reaching the I/O limit of the General Purpose performance mode.	Percent	To support a wide variety of cloud storage workloads, Amazon EFS offers two performance modes - General Purpose and Max I/O. The General Purpose performance mode is recommended for the majority of Amazon EFS file systems. General Purpose is ideal for latency-sensitive use cases, like web serving environments, content management systems, home directories, and general file serving. If you don't choose a performance mode when you create your file system, Amazon EFS selects the General Purpose mode by default. In General Purpose mode, there is a limit of 7000 file system operations per second. This operations limit is calculated for all clients connected to a single file system.If the value of this measure is close to or equal to 100% for a file system, it means that that file system has reached or is about to reach this limit. In such a case,

Measurement	Description	Measurement Unit	Interpretation
			consider moving your application to a file system using the Max I/O performance mode. File systems in the Max I/O mode can scale to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations. Highly parallelized applications and workloads, such as big data analysis, media processing, and genomics analysis, can benefit from this mode. **Note:** **This measure is reported only for file systems using the General Purpose performance mode.**
Permitted throughput	Indicates the amount of throughput this file system is allowed, given the file system size and the value of the Burst credit balance measure.	KB/Sec	A high value is desired for this measure. A very low value indicates that that file system is under duress owing to a high level of activity and/or small size.

4.3.3 AWS Simple Storage Service(S3) - Request Statistics Test

Amazon Simple Storage Service is storage for the Internet. Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.

To upload data to the cloud, you first create a bucket in one of the AWS Regions. A bucket is a container for data stored in Amazon S3. Once a bucket is created, you can then upload any number of objects to the bucket. Objects are the fundamental entities stored in Amazon S3, and consist of object data and metadata. Every object is contained in a bucket. For example, if the object named photos/puppy.jpg is stored in the johnsmith bucket, then it is addressable using the URL http://johnsmith.s3.amazonaws.com/photos/puppy.jpg

To create objects in a bucket and to manipulate these objects (say, to retrieve objects from a bucket or delete them), administrators often make Amazon S3 REST requests over HTTP - eg., HTTP GET, PUT, LIST, DELETE, etc. By monitoring the HTTP requests to Amazon S3 and their responses, operational issues can be quickly detected . This is exactly what administrators can achieve using the AWS Simple Storage Service(S3) - Request Statistics test!

This test monitors the HTTP requests to each bucket, promptly captures error responses, and brings them to the notice of administrators. In addition, the test also measures the time taken by S3 to service the requests, and in the process, warns administrators of an impending processing slowdown.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each bucket in each AWS region

First-level descriptor: AWS Region

Second-level descriptor: Bucket name

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.

Parameter	Description
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the proxy domain and proxy workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
All requests	Indicates the total number of HTTP requests made to this bucket.	Number	This is a good indicator of the workload of a bucket. Amazon S3 scales to support very high request rates. If your request rate grows steadily, Amazon S3 automatically partitions your buckets as needed to support higher request rates.
Get requests	Indicates the number of HTTP GET requests made for objects in this bucket.	Number	Amazon S3 scales to support very high request rates. If your request rate grows steadily, Amazon S3 automatically partitions your buckets as needed to support higher request rates. However, if you expect a rapid increase in the request rate for a bucket to more than 300 PUT/LIST/DELETE requests per second or more than 800 GET requests per second, we recommend

Measurement	Description	Measurement Unit	Interpretation
Put requests	Indicates the number of HTTP PUT requests made for objects in this bucket.	Number	that you open a support case to prepare for the workload and avoid any temporary limits on your request rate.
Delete requests	Indicates the number of HTTP DELETE requests made for objects in this bucket. This also includes delete multiple objects requests.	Number	
List requests	Indicates the number of HTTP requests this bucket that list the contents of the bucket.	Number	
Head requests	Indicates the number of HTTP HEAD requests made to this bucket.	Number	
Post requests	Indicates the number of HTTP POST requests made to this bucket.	Number	
Data downloaded	Indicates the amount of data downloaded from this bucket.	KB	
Data uploaded	Indicates the amount of data uploaded to this bucket.	KB	
HTTP 4XX client errors	Indicates the number of HTTP requests to this bucket that returned the HTTP 4XX client error status code.	Number	This class of status code is intended for situations in which the error seems to have been caused by the client. Ideally, the value of this measure should be 0.
HTTP 5XX server errors	Indicates the total number of HTTP 5xx server error status code requests made to this bucket.	Number	Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request.

Measurement	Description	Measurement Unit	Interpretation
			Ideally, the value of this measure should be 0.
First byte latency	Indicates the time that elapsed between when this bucket receives a complete request and when it starts returning a response to it .	Seconds	A low value is desired for this measure.
Total request latency	Indicates the time that elapsed from the first byte received to the last byte sent to this bucket.	Secs	This metric includes the time taken to receive the request body and send the response body, which is not included in First byte latency measure. If the value of this measure is very high for a bucket, then you may want to follow the best practices guidelines discussed below to ensure low latency access to and better performance of Amazon S3. These guidelines vary with the type of workload - i.e., workload with mix of request tupes and a GET-intensive workload. • **Workload with a mix of request tyes:** If your requests are typically a mix of GET, PUT, DELETE, or GET Bucket (list objects), choosing appropriate key names for your objects will ensure better performance by providing low-latency access to the Amazon S3 index. It will also ensure scalability regardless of the number of requests you send per second. • **Workloads that are GET-intensive:** If the bulk of your workload consists of GET requests,

Measurement	Description	Measurement Unit	Interpretation
			we recommend using the Amazon CloudFront content delivery service.

4.3.4 AWS Simple Storage Service(S3) - Storage Statistics Test

Amazon Simple Storage Service is storage for the Internet. Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.

To upload data to the cloud, you first create a bucket in one of the AWS Regions. A bucket is a container for data stored in Amazon S3. Once a bucket is created, you can then upload any number of objects to the bucket. Objects are the fundamental entities stored in Amazon S3, and consist of object data and metadata. Every object is contained in a bucket. For example, if the object named photos/puppy.jpg is stored in the johnsmith bucket, then it is addressable using the URL http://johnsmith.s3.amazonaws.com/photos/puppy.jpg

Each object in Amazon S3 has a storage class associated with it. Amazon S3 offers the following storage classes for the objects that you store. You choose one depending on your use case scenario and performance access requirements.

- **STANDARD** – This storage class is ideal for performance-sensitive use cases and frequently accessed data. STANDARD is the default storage class; if you do not specify storage class at the time that you upload an object, Amazon S3 assumes the STANDARD storage class.
- **STANDARD_IA** – This storage class (IA, for infrequent access) is optimized for long-lived and less frequently accessed data. For example - backups and older data where frequency of access has diminished, but the use case still demands high performance. The STANDARD_IA objects are available for real-time access.

  The STANDARD_IA storage class is suitable for larger objects greater than 128 Kilobytes that you want to keep for at least 30 days. For example, bucket lifecycle configuration has minimum object size limit for Amazon S3 to transition objects. For more information, see Supported Transitions and Related Constraints.
- **GLACIER** – The GLACIER storage class is suitable for archiving data where data access is infrequent. Archived objects are not available for real-time access. You must first restore the objects before you can access them. For more information, see Restoring Archived Objects.

You cannot specify GLACIER as the storage class at the time that you create an object. You create GLACIER objects by first uploading objects using STANDARD, RRS, or STANDARD_IA as the storage class. Then, you transition these objects to the GLACIER storage class using lifecycle management.

- **REDUCED_REDUNDANCY** – The Reduced Redundancy Storage (RRS) storage class is designed for noncritical, reproducible data stored at lower levels of redundancy than the STANDARD storage class.

To know how many buckets have been created in each region and how many objects are stored in each storage class by every bucket, use the AWS Simple Storage Service(S3) - Storage Statistics test.

This test automatically discovers the buckets that have been created in every region and reports the total count and size of objects in each bucket. You can then use the detailed diagnosis of this test to know in which storage classes each bucket is currently storing objects, and the total size of objects in each class.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each bucket in each AWS region

First-level descriptor: AWS Region

Second-level descriptor: Bucket name

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and	In some environments, all communication with the AWS EC2 cloud and its regions

Parameter	Description
Proxy Port	could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total objects	Indicates the total number	Number	

Measurement	Description	Measurement Unit	Interpretation
	of objects stored in this bucket for all storage classes except for the GLACIER storage class.		
Bucket size	Indicates the amount of data stored in a bucket in the Standard storage class, Standard - Infrequent Access (Standard_IA) storage class, and/or the Reduced Redundancy Storage (RRS) class.	GB	Use the detailed diagnosis of this measure to know the storage classes in which the bucket stores object, and the total size of objects in each class.

Use the detailed diagnosis of the Bucket size measure to know which storage classes S3 stores the bucket's objects in and the total size of these objects per class.

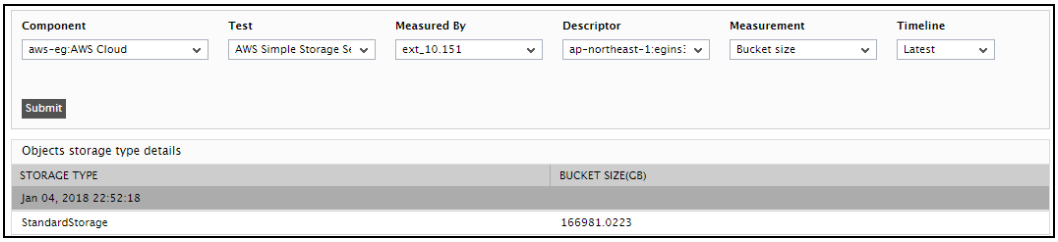


Figure 4.9: The detailed diagnosis of the Bucket size measure

4.3.5 AWS Storage Gateway Test

AWS Storage Gateway connects an on-premises software appliance with cloud-based storage to provide seamless integration with data security features between your on-premises IT environment and the Amazon Web Services (AWS) storage infrastructure.

AWS Storage Gateway offers file-based, volume-based and tape-based storage solutions:

- File Gateway – File gateway is a type of AWS Storage Gateway that supports a file interface into Amazon S3 and that adds to the current block-based volume and VTL storage. File gateway combines a service and virtual software appliance, enabling you to store and retrieve objects in Amazon S3 using industry-standard file protocols such as Network File System (NFS). The software appliance, or gateway, is deployed into your on-premises environment as a virtual machine (VM) running on VMware ESXi. The gateway provides access to objects in S3 as files on

a NFS mount point.

File gateway also provides low-latency access to data through transparent local caching. File gateway manages data transfer to and from AWS, buffers applications from network congestion, optimizes and streams data in parallel, and manages bandwidth consumption.

- Volume Gateway – Volume gateway provides cloud-backed storage volumes that you can mount as Internet Small Computer System Interface (iSCSI) devices from your on-premises application servers. The gateway supports the following volume configurations:
  - Cached Volumes – You store your data in Amazon Simple Storage Service (Amazon S3) and retain a copy of frequently accessed data subsets locally.
  - Stored volumes - If you need low-latency access to your entire data set (and not just the frequently accessed data set), you can configure your on-premises gateway to store all your data locally and then asynchronously back up point-in-time snapshots of this data to Amazon S3.
- Tape Gateway – Tape Gateway provides a virtual tape infrastructure that scales seamlessly with your business needs and eliminates the operational burden of provisioning, scaling, and maintaining a physical tape infrastructure.

In order to ensure the peak performance of their mission-critical applications, administrators must make sure that the storage gateway used by and volumes provisioned for the on-premise applications are able to process I/O requests quickly and are sized commensurate to the current and anticipated load.

The AWS Storage Gateway Test helps administrators with this analysis. This test auto-discovers the storage gateways configured on AWS and reports the I/O throughput, cache usage, and I/O latency of each storage gateway. In the process, the test pinpoints overloaded gateways and those that are experiencing slowness when processing I/O requests. With the help of the test, you can also judge how effectively/otherwise the cache is being used, and determine how the cache can be tweaked to improve performance.

You can also optionally configure the test to report metrics for each volume, instead of each storage gateway. Using the per-volume metrics that the test reports, you can quickly identify volumes under duress, and can rapidly initiate performance improvement measures.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each storage gateway / volume (as the case may be)

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: ***east*,*west***
Gateway Filter Name	By default, this test reports metrics for each storage gateway configured. Accordingly, this flag is set to **GatewayId** by default. In this case, the measures of this test will be aggregated across all storage volumes of a storage gateway. If you want, you can configure this test to report usage metrics per storage volume. For this, set this flag to **VolumeID**.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Read data	If the Gateway Filter Name parameter is set to **GatewayID**, then this measure reports the amount of data that on-premise applications read from this storage gateway for all volumes in the gateway. If the Gateway Filter Name is set to **VolumeID**, then this measure will report the amount of data that was read from this volume by on-premise applications.	KB	If the value of these measures is consistently low for any gateway/volume, it indicates low throughput. Here are some recommended best practices for optimizing gateway performance: • Add high performance disks such as solid-state drives (SSDs) and a NVMe controller. • Attach virtual disks to your VM directly from a storage area network (SAN) instead of the Microsoft Hyper-V NTFS. • Confirm that the virtual processors that are assigned to the gateway VM are backed by an equal number of cores and that you are not oversubscribing the CPUs of the host server. • You can add additional CPUs to the gateway host server. • When you provision disks in a gateway setup, we strongly recommend that you do not provision local disks for the upload buffer and cache storage that use the same underlying physical storage disk.

Measurement	Description	Measurement Unit	Interpretation
			• For volumes gateways, if you find that adding more volumes to a gateway reduces the throughput to the gateway, consider adding the volumes to a separate gateway. In particular, if a volume is used for a high-throughput application, consider creating a separate gateway for the high-throughput application. However, as a general rule, you should not use one gateway for all of your high-throughput applications and another gateway for all of your low-throughput applications.
Write data	If the Gateway Filter Name parameter is set to **GatewayID**, then this measure reports the amount of data that on-premise applications wrote into this storage gateway for all volumes in the gateway. If the Gateway Filter Name is set to **VolumeID**, then this measure will report the amount of data read that was written into this volume by on-premise applications.	KB	
Read time	If the Gateway Filter Name parameter is set to **GatewayID**, then this measure reports the time taken by on-premise applications read from storage volumes in this gateway. If the Gateway Filter Name is set to **VolumeID**, then this measure will report the time taken by on-premise applications to read from this volume.	Secs	An abnormally high value for these measures indicates an I/O processing bottleneck. You may want to investigate the slowdown further and isolate its root-cause. The best practices discussed in the Interpretation of the Read data and Write data measure can be employed to optimize gateway performance and avert such anomalies.
Write time	If the Gateway Filter Name parameter is set to **GatewayID**, then this measure reports the time	Secs	

Measurement	Description	Measurement Unit	Interpretation
	taken by on-premise applications to write into all storage volumes in this gateway. If the Gateway Filter Name is set to **VolumeID**, then this measure will report the time taken by on-premise applications to write into this volume.		
Queued writes	If the Gateway Filter Name parameter is set to **GatewayID**, then this measure reports the amount of data waiting to be written to all volumes of this gateway. If the Gateway Filter Name is set to **VolumeID**, then this measure will report the amount of data waiting to be written to this volume.	KB	A high value of this measure or a steady increase in the value of this measure for a storage gateway/volume could indicate an I/O processing bottleneck.
Compressed data downloaded	Indicates the amount of compressed data that all volumes of this gateway downloaded from AWS.	KB	**This measure is reported only for each storage gateway, and not for each volume.**
Compressed data uploaded	Indicates the amount of compressed data that all volumes of this gateway uploaded to AWS.	KB	**This measure is reported only for each storage gateway, and not for each volume.**
Compressed data read time	Indicates the amount of time taken to read compressed data from gateway.	Secs	**These measures are reported only for each storage gateway, and not for each volume.** A steady increase in the value of this measure could indicate an I/O

Measurement	Description	Measurement Unit	Interpretation
Compressed data write time	Indicates the amount of time taken to write compressed data into gateway.	Secs	processing bottleneck.
Data usage of upload buffer	Indicates the percent usage of this gateway's upload buffer.	Percent	**This measure is reported only for cached volume gateways and tape gateways.** To prepare for upload to Amazon S3, a cached volume gateway and/or a tape gateway stores incoming data in a staging area, referred to as an upload buffer. Your gateway uploads this buffer data over an encrypted Secure Sockets Layer (SSL) connection to AWS, where it is stored encrypted in Amazon S3. A value close to 100% for this measure indicates that the disk used by the storage gateway as the upload buffer is running out of space. This can happen if the gateway is unable to write data to Amazon S3 at the same pace at which it writes to the buffer. This in turn implies a bottleneck when uploading. This can also happen if the disk is not sized right. The minimum disk space recommendation for the working storage upload buffer is 150 GiB and the maximum is 2 TiB.
Data used in upload buffer	Indicates the total number of bytes being used in this gateway's upload buffer.	KB	**This measure is reported only for cached volume gateways and tape gateways.**
Free data in upload buffer	Indicates the total amount of unused space in this gateway's working storage.	KB	**This measure is reported only for cached volume gateways and tape gateways.**

Measurement	Description	Measurement Unit	Interpretation
			A high value is desired for this measure.
Free space in gateway's working storage	Indicates the total amount of unused space in this stored volume gateway's upload buffer.	KB	**This measure is reported only for stored volume gateways.** To prepare for upload to Amazon S3, a stored volume gateway stores incoming data in a staging area, referred to as an upload buffer/working storage. Your gateway uploads this buffer data over an encrypted Secure Sockets Layer (SSL) connection to AWS, where it is stored encrypted in Amazon S3. Adequate free space should be available in the working storage to enable the gateway to store all the incoming data before upload. A high value is hence desired for this measure. The minimum disk space recommendation for the working storage is 150 GiB and the maximum is 2 TiB.
Data usage of gateway's working storage	Indicates the percent usage of this storage volume gateway's working storage.	Percent	**This measure is reported only for stored volume gateways.** A value close to 100% for this measure indicates that the disk used by the storage volume gateway as the working storage is running out of space. This can happen if the gateway is unable to write data to Amazon S3 at the same pace at which it writes to the working storage. This in turn implies a bottleneck when uploading. This can also happen if the disk is not sized right. The minimum disk space recommendation for the working

Measurement	Description	Measurement Unit	Interpretation
			storage upload buffer is 150 GiB and the maximum is 2 TiB.
Data used in gateway's working storage	Indicates the total amount of data being used in the storage volume gateway's upload buffer.	KB	**This measure is reported only for stored volume gateways.**
Application reads served from cache	If the Gateway Filter Name parameter is set to **GatewayID**, then this measure reports the percentage of application reads served from this gateway's cache. If the Gateway Filter Name is set to **VolumeID**, then this measure will report the percentage of read operations from this volume that are served from the cache.	Percent	Ideally, the value of this measure should be above 80%. If not, then it means that many read requests are being serviced by directly accessing the data in AWS. This can increase I/O overheads and adversely impact application performance.
Usage of gateway's cache storage	If the Gateway Filter Name parameter is set to **GatewayID**, then this measure reports the percent usage of this gateway's cache. If the Gateway Filter Name parameter is set to **VolumeID**, then this measure reports what percentage of the gateway's cache storage is used by this volume.	Percent	If the value of this measure grows steadily close to 100%, it denotes the excessive usage of that gateway's cache storage. If the value of this measure is close to 100% for a volume, it implies that a particular volume is taking up too much cache space. If the gateway's cache storage runs out of space, then the cache will no longer be able to hold frequently-accessed objects; this in turn will increase cache misses and related overheads. This is why, the cache storage has to be sized rightly. The recommended minimum cache size is 150 GiB and the maximum is 16

Measurement	Description	Measurement Unit	Interpretation
			TiB.
Gateway's cache has not been persisted to AWS	If the Gateway Filter Name parameter is set to **GatewayID**, then this measure reports the percentage of this gateway's cache that has not been persisted to AWS. If the Gateway Filter Name parameter is set to **VolumeID**, then this measure reports what percentage of the gateway's cache storage has not been persisted to this volume of AWS.	Percent	As your applications write data to the storage volumes in AWS, the gateway initially stores the data on the cache storage before uploading the data to Amazon S3. The value of this measure represents the amount of cached data that is yet to be uploaded to Amazon S3. If this value is very high, it could indicate that the gateway is having trouble uploading data to AWS. You may want to investigate the reasons for the same. In the process, you may also want to configure this test to report metrics and volume, and identify the exact volume on AWS to which maximum data has not been uploaded.
Total cache size	Indicates the amount of data stored in this gateway's cache.	KB	**This measure is reported only for each storage gateway, and not for each volume.**
Time since last available recovery point	Indicates the time since the last available recovery point of this gateway's cache storage.	Secs	**This measure is reported only for each storage gateway, and not for each volume.** A volume recovery point is a point in time at which all data of the volume is consistent. You can clone a volume or create a snapshot of it from its recovery point.
Data used in gateway's cache storage	Indicates the amount of data being used in this gateway's cache storage.	KB	**This measure is reported only for each storage gateway, and not for each volume.**
Free space in gateway's cache storage	Indicates the total amount of unused space in this gateway's cache storage.	KB	**This measure is reported only for each storage gateway, and not for each volume.**

Measurement	Description	Measurement Unit	Interpretation
			Ideally, the value of this measure should be high.

4.4 The AWS Cloud Instances Layer

The tests mapped to this layer take stock of the total number of instances (that are available for the configured AWS user account) on the cloud, and points you to the following:

- The powered-off instances;
- Instances that are unavailable over the network;
- Resource-hungry instances;

- Instances that rebooted recently

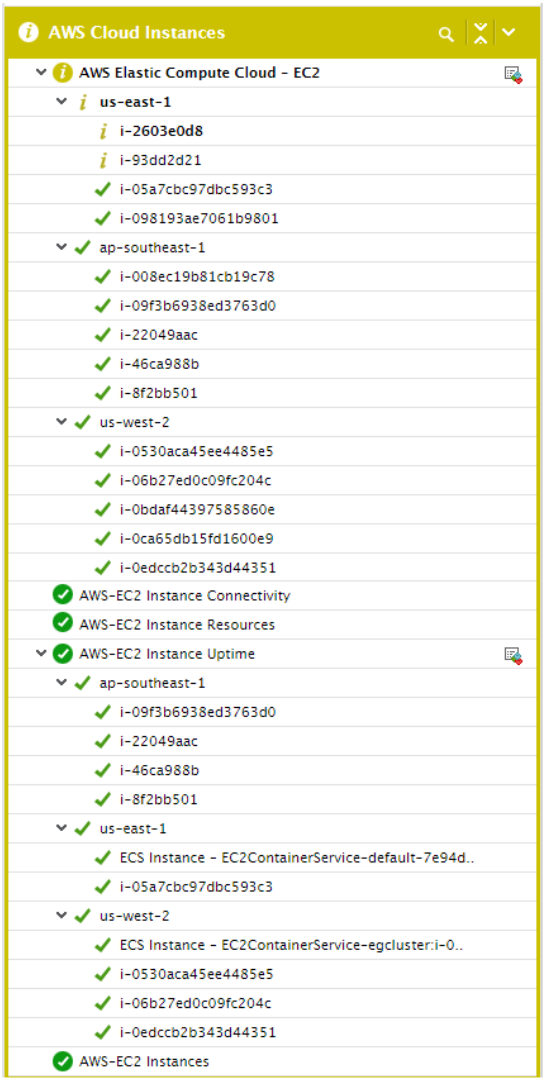


Figure 4.10: The tests mapped to the AWS Cloud Instances layer

4.4.1 AWS-EC2 Instances Test

An Amazon Machine Image (AMI) contains all information necessary to boot instances of your software. For example, an AMI might contain all the software to act as a web server (e.g., Linux, Apache, and your web site) or it might contain all the software to act as a Hadoop node (e.g., Linux, Hadoop, and a custom application). After an AMI is launched, the resulting running system is called an instance. All instances based on the same AMI start out identical and any information on them is lost when the instances are terminated or fail.

Users with valid AWS user accounts can sign into the EC2 cloud to view and use available instances, or purchase and launch new ones. With the help of this test, you can determine the total number of instances that are currently available for the configured AWS user account, the number of instances that were newly purchased/terminated, and the count of powered-off instances.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for the AWS EC2 Cloud being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY**

Parameter	Description
	WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Cloudwatch Enabled	**This parameter only applies to the AWS - EC2 Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Total instances:	Indicates the total number of instances currently available for the configured AWS user account.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances available for use for the configured AWS account, regardless of the current state of the instances.
Instances powered on:	Indicates the total number of instances that are currently powered-on.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the powered-on instances available for use for the configured AWS account.
Instances powered off:	Indicates the total number of instances that are currently powered-off.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the powered-off instances available for the configured AWS account.
Added instances:	Indicates the total number of instances that were newly purchased by the configured AWS user account during the last measurement period.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances that were newly purchased and launched by the configured AWS user account.
Removed instances:	Indicates the total number of instances that were newly terminated by the configured AWS user account during the last measurement period.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances that were newly terminated/removed by the configured AWS user account.

4.4.2 AWS Elastic Compute Cloud - EC2 Test

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. Since users may run mission-critical applications on these EC2 instances, high uptime of the EC2 instances is imperative to the uninterrupted functioning of these applications and to ensure 100% user satisfaction with this

cloud-based service. AWS administrators therefore, should frequently perform health checks on every instance, measure its load and resource usage, and capture potential failures and resource contentions, well before end-users notice and complain. This is exactly where the AWS Elastic Compute Cloud - EC2 test helps!

This test monitors the powered-on state of each EC2 instance and promptly alerts administrators if any instance has been powered-off inadvertently. Additionally, the test also reveals how each instance uses the CPU, disk, and network resources it is configured with, thus providing early pointers to irregularities in instance sizing, and prompting administrators to make necessary amends. This way, the test makes sure that critical applications are always accessible to end-users and perform at peak capacity.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each instance / auto scaling group / instance type / image ID in each region of the AWS cloud being monitored, depending upon the option chosen from the **EC2 FILTER NAME** drop-down

First-level descriptor: AWS EC2 region name

Second-level descriptor: EC2 instance ID / auto scaling group name / instance type / image ID

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address

Parameter	Description
	of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Exclude Instance	**This parameter is applicable only if InstanceId is chosen from the EC2 Filter Name drop-down.** In this case, against **EXCLUDE INSTANCE**, you can provide a comma-separated list of instance IDs you do not want the test to monitor.
EC2 Filter Name	By default, this test reports metrics for each instance in the AWS infrastructure. This is why, the **EC2 FILTER NAME** flag is set to *Instance ID* by default. Alternatively, you can configure this test to aggregate metrics across a chosen collection of instances, and report one set of metrics per collection. For this, you just need to pick an instance collection from the EC2 Filter Name drop-down. The options available are as follows: • *AutoScalingGroupName*: Your EC2 instances can be organized into Auto Scaling Groups so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and, desired number of EC2 instances. If you select the *AutoScalingGroupName* option from the **EC2 FILTER NAME** drop-down, then this test will collect metrics for each instance, aggregate the metrics on the basis of the Auto Scaling Groups to which the instances belong, and report metrics for each group.

Parameter	Description
	• *InstanceType*: Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. If the *InstanceType* option is chosen from the **EC2 FILTER NAME** drop-down, then this test will collect metrics for each instance, aggregate the metrics on the basis of the instance type, and report metrics for each type. • *ImageId*: Instances are created from Amazon Machine Images (AMI). The machine images are like templates that are configured with an operating system and other software, which determine the user's operating environment. If the *ImageId* option is chosen from the **EC2 FILTER NAME** drop-down, then this test will collect metrics for each instance, aggregate the metrics on the basis of the AMI using which the instances were created, and report metrics for each image ID.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Instance power-on state:	Indicates the current powered-on state of this instance.		This measure is reported only if InstanceID is the option from the EC2 Filter Name drop-down of this test. The values that this measure can report and their corresponding numeric values are detailed in the table below:

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
Running	When the instance is ready for you, it enters the running state.	1
Pending	When you launch an instance, it enters the pending state	2
Terminated	When you no longer need an instance, you can terminate it, then it goes to terminated state.	3
Shutting down	While terminate the instance, As soon as the status of an instance changes to	4

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td>shutting-down or terminated</td><td></td></tr>
<tr><td>Stopping</td><td>When you stop your instance, it enters the stopping state</td><td>5</td></tr>
<tr><td>Stopped</td><td>After exiting the stopping state, it enters the stopped state</td><td>0</td></tr>
</table>
Note:
By default, this measure will report the Measure Values listed in the table above to indicate the current powered-on state of an instance. In the graph of this measure however, the same will be represented using the numeric equivalents only.
</td></tr>
<tr><td>EBS volumes</td><td>Indicates the number of EBS volumes attached to this instance.</td><td>Number</td><td>This measure is reported only if the InstanceId option is chosen from the EC2 Filter Name drop-down of this test.
You can attach an EBS volumes to one of your instances that is in the same Availability Zone as the volume.
You can attach multiple volumes to the same instance within the limits specified by your AWS account. Your account has a limit on the number of EBS volumes that you can use, and the total storage available to you.
Using the detailed diagnosis of this measure, you can identify the volumes that are attached to this</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			EC2 instance.
CPU credit usage:	Indicates the number of CPU credits consumed by this T2 instance / all T2 instances / all T2 instances created from this image ID during the last measurement period.	Number	This measure is reported only for individual T2 instances, the T2 instance type, and the image ID using which T2 instances (if any) were created. A CPU Credit provides the performance of a full CPU core for one minute. Traditional Amazon EC2 instance types provide fixed performance, while T2 instances provide a baseline level of CPU performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits. One CPU credit is equal to one vCPU running at 100% utilization for one minute. Other combinations of vCPUs, utilization, and time are also equal to one CPU credit; for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes. Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set rate of CPU credits per hour, depending on instance size. When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting. When your T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization. The more credits your T2 instance has for CPU resources, the more time it can burst beyond its base performance level when more performance is needed. This implies that ideally, the value of the CPU credit usage measure should be low for an instance and the

Measurement	Description	Measurement Unit	Interpretation
			value of the CPU credit balance for that instance should be high, as that way, an instance is assured of more CPU resources when performance demands increase. By comparing the value of this measure across instances, you can precisely identify the instance that has used up a sizeable portion of its CPU credits.
CPU credit balance:	Indicates the number of CPU credits that have been earned by this T2 instance / all T2 instances / all T2 instances created from this image ID	Number	
CPU utilization:	Indicates the percentage of allocated EC2 compute units that are currently in use on this instance.	Percent	A value close to 100% indicates excessive usage of CPU by an instance. If the value of this measure is consistently high for an instance, it could indicate that the application running on that instance requires more processing power. In such a case, you may want to allocate more CPU resources to that instance.
Disk read operations:	Indicates the rate at which read operations were performed on all disks available to this instance.	Operations/Sec	Compare the value of this measure across instances to know which instance is too slow in processing read requests.
Disk write operations:	Indicates the rate at which write operations were performed on all disks available to this instance.	Operations/Sec	Compare the value of this measure across instances to know which instance is too slow in processing write requests.
Disk reads:	Indicates the rate at which data was read from all disks available to this instance.	KB/Sec	Compare the value of this measure to identify the instance that is the slowest in responding to read requests.
Disk writes:	Indicates the rate at which data was written to all disks available to this	KB/Sec	Compare the value of this measure to identify the instance that is the slowest in responding to write requests.

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td>instance.</td><td></td><td></td></tr>
<tr><td>Incoming network traffic:</td><td>Indicates the rate at which data was received by all network interfaces of this instance.</td><td>KB/Sec</td><td rowspan="2">Compare the value of these measures across instances to know which instance is consuming too much bandwidth. Then, compare the value of the Incoming network traffic and Outgoing network traffic measures of that instance to determine where bandwidth consumption was more - when receiving data over the network? or when sending data?</td></tr>
<tr><td>Outgoing network traffic:</td><td>Indicates the rate at which data was sent by all the network interfaces of this instance.</td><td>KB/Sec</td></tr>
<tr><td>EC2 status check:</td><td>Indicates whether a status check (system status check or instance status check) failed for this instance</td><td></td><td>Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. These status checks are of two types: system and instance status checks.

If either of these status checks fails, then this measure will report the value Failed. If none of these status checks fail, then this measure will report the value Passed.

The values that this measure can report and their corresponding numeric values are listed in the table below:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Failed</td><td>1</td></tr><tr><td>Passed</td><td>0</td></tr></table>
Note:

By default, this measure reports the Measure Values above to indicate whether a check passed or failed. In the graph of this measure however, the same is indicated using the numeric equivalents only.</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>EC2 instance status check:</td><td>Indicates whether/not this instance passed the EC2 instance status check in the last minute.</td><td></td><td>Instance status checks monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).

The following are examples of problems that can cause instance status checks to fail:

• Failed system status checks
• Incorrect networking or startup configuration
• Exhausted memory
• Corrupted file system
• Incompatible kernel

The values that this measure can report and their corresponding numeric values are listed in the table below:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Failed</td><td>1</td></tr><tr><td>Passed</td><td>0</td></tr></table>
Note:

By default, this measure reports the Measure Values above to indicate whether a check passed or failed. In the graph of this measure however, the same is indicated using the numeric equivalents only.</td></tr>
<tr><td>EC2 system status check:</td><td>Indicates whether/not this instance passed the EC2 system</td><td>Number</td><td>System status checks monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td>status check in the last minute.</td><td></td><td>to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself (for example, by stopping and starting an instance, or by terminating and replacing an instance).

The following are examples of problems that can cause system status checks to fail:

• Loss of network connectivity
• Loss of system power
• Software issues on the physical host
• Hardware issues on the physical host

The values that this measure can report and their corresponding numeric values are listed in the table below:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Failed</td><td>1</td></tr><tr><td>Passed</td><td>0</td></tr></table>
Note:

By default, this measure reports the Measure Values above to indicate whether a check passed or failed. In the graph of this measure however, the same is indicated using the numeric equivalents only.</td></tr>
<tr><td>Incoming network packets</td><td>Indicates the number of packets received on all network interfaces by this instance.</td><td>Number</td><td>This metric identifies the volume of incoming traffic in terms of the number of packets on a single instance.

By comparing the value of this measure across instances, you can identify that instance which is seeing the maximum incoming traffic.</td></tr>
<tr><td>Outgoing network</td><td>Indicates the</td><td>Number</td><td>This metric identifies the volume of outgoing traffic</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
packets	number of packets sent out on all network interfaces by this instance.		in terms of the number of packets on a single instance. By comparing the value of this measure across instances, you can identify that instance which is seeing the maximum outgoing traffic.
Disk IOPS	Indicates the rate at which read and write operations were performed on all disks available to this instance.	Operations/Sec	Compare the value of this measure across instances to identify the busiest instance in terms of I/O activity. If the value of this measure is abnormally high for an instance, it could hint at a potential I/O overload.
Disk throughput	Indicates the rate at which data was read from and written into all disks available to this instance.	KB/Sec	Compare the value of this measure across instances to identify the instance with the lowest disk throughput.
Network throughput	Indicates the amount of data received and sent on all network interfaces by this instance.	KB/Sec	Compare the value of this measure across instances to identify the instance with the highest network throughput. If any instance is sized with limited bandwidth resources, a high network throughput may choke the instance.
Uptime of instance	Indicates the total time that this instance has been up since its last reboot.	Mins	This measure displays the number of years, months, days, hours, minutes and seconds since the last reboot.

Detailed Diagnosis:

Using the detailed diagnosis of the **EBS volumes** measure, you can identify the volumes that are attached to a particular EC2 instance.

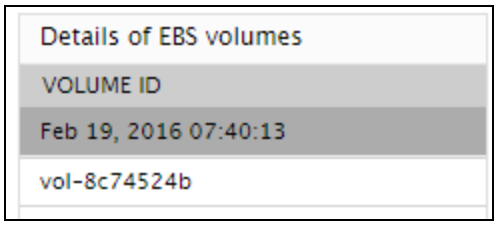


Figure 4.11: The detailed diagnosis of the EBS volumes measure

4.4.3 AWS-EC2 Instance Uptime Test

In cloud-based environments, it is essential to monitor the uptime of server instances launched on the cloud. By tracking the uptime of each of the instances, administrators can determine what percentage of time an instance has been up. Comparing this value with service level targets, administrators can determine the most trouble-prone areas of the infrastructure hosted on the cloud.

In some environments, administrators may schedule periodic reboots of their instances. By knowing that a specific instance has been up for an unusually long time, an administrator may come to know that the scheduled reboot task is not working on an instance.

This test monitors the uptime of each instance available to the configured AWS user account.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each *instancename:instanceID* available for the configured AWS user account

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.

Parameter	Description
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Cloudwatch Enabled	**This parameter only applies to the AWS - EC2 Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of

Parameter	Description
	instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.
Report Manager Time	By default, this flag is set to **Yes**, indicating that, by default, the detailed diagnosis of this test, if enabled, will report the shutdown and reboot times of the cloud in the manager's time zone. If this flag is set to **No**, then the shutdown and reboot times are shown in the time zone of the system where the agent is running (i.e., the system system on which the remote agent is running).
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Has the instance been rebooted?:	Indicates whether this instance has been rebooted during the last measurement period or not.	Boolean	If this measure shows 1, it means that the instance was rebooted during the last measurement period. By checking the time periods when this metric changes from 0 to 1, an administrator can determine the times when this instance was rebooted.
Uptime of the instance during the last measure period:	Indicates the time period that the instance has been up since the last time this test ran.	Secs	If the instance has not been rebooted during the last measurement period and the agent has been running continuously, this value will be equal to the measurement period. If the instance was rebooted during the last measurement period, this value will be less than the measurement period of the test. For example, if the measurement period is 300

Measurement	Description	Measurement Unit	Interpretation
			secs, and if the instance was rebooted 120 secs back, this metric will report a value of 120 seconds. The accuracy of this metric is dependent on the measurement period - the smaller the measurement period, greater the accuracy.
Total uptime of the instance:	Indicates the total time that this instance has been up since its last reboot.	Mins	Administrators may wish to be alerted if an instance has been running without a reboot for a very long period. Setting a threshold for this metric allows administrators to determine such conditions.

4.4.4 AWS-EC2 Instance Resources Test

Tracking the CPU usage, disk and network I/O of every instance launched by a configured AWS user account will provide administrators with valuable insights into how well the instances are utilizing the allocated resources. The **AWS-EC2 Instance Resources** test does just that. This test auto-discovers the instances available for the configured AWS user account, and reports the resource usage of each instance so that, administrators can quickly compare the usage metrics across instances and pinpoint which instance is resource-hungry.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each *instancename:instanceID* available for the configured AWS user account

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The

Parameter	Description
AWS Secret Key	procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Cloudwatch Enabled	**This parameter only applies to the AWS - EC2 Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.

Parameter	Description
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
CPU utilization:	Indicates the percentage of allocated CPU consumed by this instance.	Percent	A high value for this measure indicates that an instance is utilizing CPU excessively - this could be because of one/more resource-intensive processes executing on that instance. Compare the value of this measure across instances to identify the CPU-intensive instances.
Incoming network traffic:	Indicates the rate of incoming network traffic i.e., the rate at which the bytes are received by all the network interfaces connected to this instance.	KB/Sec	Compare the values of these measures across instances to quickly identify the instance that is utilizing the network bandwidth excessively.
Outgoing network traffic:	Indicates the volume of outgoing network traffic i.e., the rate at which the bytes are transferred from all the network interfaces connected to this instance.	KB/Sec	
Disk reads:	Indicates the rate at which data is read from the disks of this instance.	KB/Sec	These measures are good indicators of the level of disk I/O activity on an instance. By comparing the values of these measures across instances, you can accurately determine which instance is performing I/O-

Measurement	Description	Measurement Unit	Interpretation
			intensive operations.
Disk writes:	Indicates the rate at which data is written to the disks of this instance.	KB/Sec	
Disk read operations:	Indicates the rate at which disk read operations are performed on this instance.	Operations/Sec	These measures are good indicators of the level of disk I/O activity on an instance type. By comparing the values of these measures across types, you can accurately determine the type of instances that is performing I/O-intensive operations.
Disk write operations:	Indicates the rate at which disk write operations were performed on this instance.	Operations/Sec	

4.4.5 AWS-EC2 Aggregated Resource Usage Test

When users launch an instance using the AWS management console, they need to specify the instance type. An instance type is a specification that defines the memory, CPU, storage capacity, and hourly cost for an instance. Some instance types are designed for standard applications, whereas others are designed for CPU-intensive applications, or memory-intensive applications, etc. The different instance types offered by the AWS EC2 cloud are as follows:

Type	CPU	Memory	Local Storage	Platform	I/O	Name
Small	1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)	1.7 GB	160 GB instance storage (150 GB plus 10 GB root partition)	32-bit	Moderate	m1.small
Large	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)	7.5 GB	850 GB instance storage (2 x 420 GB plus 10 GB root partition)	64-bit	High	m1.large
Extra Large	8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)	15 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	m1.xlarge
Micro	Up to 2 EC2 Compute Units (for short periodic bursts)	613 MB	None (use Amazon EBS volumes for storage)	32-bit or 64-bit	Low	t1.micro

High-CPU Medium	5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each)	1.7 GB	350 GB instance storage (340 GB plus 10 GB root partition)	32-bit	Moderate	c1.medium
High-CPU Extra Large	20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each)	7 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	c1.xlarge
High-Memory Extra Large	6.5 EC2 Compute Units (2 virtual cores with 3.25 EC2 Compute Units each)	17.1 GB	420 GB instance storage (1 x 420 GB)	64-bit	Moderate	m2.xlarge
High-Memory Double Extra Large	13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each)	34.2 GB	850 GB instance storage (1 x 840 GB plus 10 GB root partition)	64-bit	High	m2.2xlarge
High-Memory Quadruple Extra Large	26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each)	68.4 GB	1690 GB instance storage (2 x 840 GB plus 10 GB root partition)	64-bit	High	m2.4xlarge
Cluster Compute	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture)	23 GB	1690 GB instance 64-bit storage (2 x 840 GB plus 10 GB root partition)	64-bit	Very high (10 Gbps Ethernet)	cc1.4xlarge
Cluster GPU	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture), plus 2 NVIDIA Tesla M2050 "Fermi" GPUs	22 GB (see note after this table)	1690 GB instance 64-bit storage (2 x 840 GB plus 10 GB root partition)	64-bit	Very high (10 Gbps Ethernet)	cg1.4xlarge

By closely monitoring the CPU usage and the network and disk I/O of each instance type, and comparing these metrics across instance types, you can quickly isolate resource-intensive types. Once again, the test will report metrics for only those types of instances that were launched by the AWS user account configured for the test.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each type of instance launched by the configured AWS user account

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Cloudwatch Enabled	**This parameter only applies to the AWS - EC2 Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource

Parameter	Description
	usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
CPU utilization:	Indicates the percentage of allocated CPU consumed by all instances of this type.	Percent	A high value for this measure indicates that one/more instances of a type are utilizing CPU excessively - this could be because of one/more resource-intensive processes executing on the instances. Compare the value of this measure across types to identify the types of instances that are CPU-intensive.
Incoming network traffic:	Indicates the rate of incoming network traffic i.e., the rate at which the bytes are received by all the network interfaces connected to all the instances of this instance type.	KB/Sec	Compare the values of these measures across instance types to quickly identify the types of instances that are utilizing the network bandwidth excessively.
Outgoing network traffic:	Indicates the volume of outgoing network	KB/Sec	

Measurement	Description	Measurement Unit	Interpretation
	traffic i.e., the rate at which the bytes are transferred from all the network interfaces connected to all the instances of a particular instance type.		
Disk reads:	Indicates the rate at which data is read from the disks of all instances of this type.	KB/Sec	These measures are good indicators of the level of disk I/O activity on an instance type. By comparing the values of these measures across types, you can accurately determine the type of instances that is performing I/O-intensive operations.
Disk writes:	Indicates the rate at which data is written to the disks of all instances of this type.	KB/Sec	
Disk read operations:	Indicates the rate at which disk read operations were performed on the disks of all instances of this type.	Operations/Sec	These measures are good indicators of the level of disk I/O activity on an instance type. By comparing the values of these measures across types, you can accurately determine the type of instances that is performing I/O-intensive operations.
Disk write operations:	Indicates the rate at which disk write operations were performed on the disks of all instances of this type.	Operations/Sec	

4.4.6 AWS-EC2 Instance Connectivity Test

Sometimes, an instance could be in a powered-on state, but the failure of the operating system or any fatal error in internal operations of the instance could have rendered the instance inaccessible to users. In order to enable you to promptly detect such 'hidden' anomalies, this test periodically runs a connectivity check on each instance available for the configured AWS user account, and reports whether the instances are accessible over the network or not.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each *instancename:instanceID* available for the configured AWS user account

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions

Parameter	Description
	with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Cloudwatch Enabled	**This parameter only applies to the AWS - EC2 Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Avg network delay:	Indicates the average delay between transmission of packets to this instance and receipt of the response to the packet at the source.	Secs	An increase in network latency could result from misconfiguration of the router(s) along the path, network congestion, retransmissions at the network, etc.
Min network delay:	The minimum time between transmission of a packet and receipt of the response back.	Secs	A significant increase in the minimum round-trip time is often a sure sign of network congestion.
Packet loss:	Indicates the percentage of packets	Percent	Packet loss is often caused by network buffer overflows at a network router or by

Measurement	Description	Measurement Unit	Interpretation
	lost during transmission from source to target and back.		packet corruptions over the network. The detailed diagnosis for this measure provides a listing of routers that are on the path from the external agent to target server, and the delays on each hop. This information can be used to diagnose the hop(s) that could be causing excessive packet loss/delays.
Network availability of Instance:	Indicates whether the network connection to this instance is available or not.	Percent	A value of 100 indicates that the instance is accessible over the network. The value 0 indicates that the instance is inaccessible. Typically, the value 100 corresponds to a Packet loss of 0.

4.5 The AWS Database Layer

Using the tests mapped to this layer, Amazon's database and database-related services, such as Amazon DynamoDB, Amazon ElastiCache, Amazon RedShift, and Amazon Relational Database Service are monitored, and irregularities in the operations, usage, and configuration of these services are brought to light.

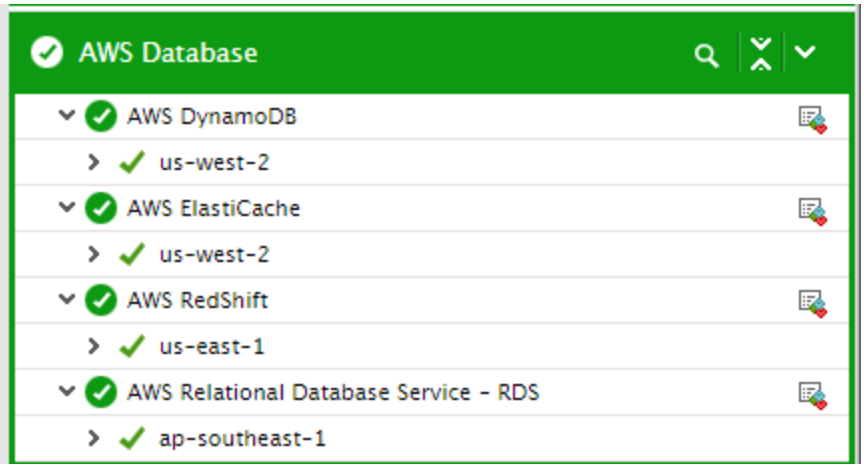


Figure 4.12: The tests mapped to the AWS Database layer

4.5.1 AWS DynamoDB Test

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. You can use Amazon DynamoDB to create a database table

that can store and retrieve any amount of data, and serve any level of request traffic. Amazon DynamoDB automatically spreads the data and traffic for the table over a sufficient number of servers to handle the request capacity specified by the customer and the amount of data stored, while maintaining consistent and fast performance.

If the tables are poorly sized with read and write capacity, then the DynamoDB service will be unable to service the read and/or write requests rapidly and successfully. This can cause critical read/write requests to be throttled, thus negatively impacting user productivity . If this is to be avoided, then administrators should continuously monitor how each table services the read/write requests to it, measure the usage of read/write capacity by every table, and accurately isolate those tables where requests may potentially be throttled owing to inadequate capacity. This is exactly where the AWS DynamoDB test helps!

This test automatically discovers the tables created using the DyanamoDB service. For each table, the test reports the provisioned read/write capacity and capacity consumed by each table, thus turning the spot light on tables that are under-sized - i.e., that do not have adequate resources to process their workload. The test also reports the latency and the number of requests throttled per table, thus enabling administrators to gauge the impact of improper sizing on performance. In the process, administrators are also provided pointers on how table capacity can be fine-tuned to improve performance. Additionally, the test also captures and promptly reports errors and request failures, which also contribute to a sub-par experience with the DynamoDB service.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each table on DynamoDB

First-level descriptor: AWS Region

Second-level descriptor: TableName

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The

Parameter	Description
	procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be

Parameter	Description
	available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Conditional check failed requests	Indicates the number of conditional write attempts to this table that failed.	Number	During a write request like PutItem, UpdateItem or DeleteItem operations, you can define a logical condition that defines whether the item can be modified or not, e.g. the item can be updated only if it's not marked as "protected". This logical condition must return "true" to allow the operation to proceed. If it returns "false", this metric is incremented and a 400 error (Bad request) is returned. Note that a conditional write failure will not increment the value of the User errors measure.
Provisioned read capacity	Indicates the number of read capacity units that has been provisioned for this table.	Number	The value of this measure is the sum of read capacity units provisioned for the table and the global secondary index on that table. You can use the detailed diagnosis of this measure to know how many read capacity units have been provisioned to this table and to the global secondary index on this table. One read capacity unit represents one strongly consistent read per second, or two eventually consistent reads per second, for an item up to 4 KB in size. If you need to read an item that is larger than 4 KB, DynamoDB will need to consume additional read capacity units. The total number of read capacity units required depends on the item size, and whether you want an eventually consistent or

Measurement	Description	Measurement Unit	Interpretation
			strongly consistent read.
Consumed read capacity	Indicates the number of read capacity units consumed by this table.	Number	Tracking changes in read consumed capacity allows administrators to spot abnormal peaks or drops in read activities. In particularly, administrators can make sure that consumption does not exceed the value of the Provisioned read capacity measure. The value of the Consumed read capacity measure is the sum of read capacity units consumed by the table and by the global secondary index on that table. To know how many capacity units have been consumed by the table and by the index, use the detailed diagnosis of this measure. If the value of the Consumed read capacity measure for a table is close to or equal to the value of its Provisioned read capacity measure, or if the value of the Read capacity measure for a table is close to or equal to 100%, it indicates that the read requests to that table have been or will potentially be throttled. Throttling prevents your application from consuming too many capacity units. When a request is throttled, it fails with an HTTP 400 code (Bad Request) and a ProvisionedThroughputExceededException. Administrators might want to set up a first alert before a table consumes its entire capacity —for instance, they can configure an alert to be triggered at a threshold of 80%. This would give them time to scale up capacity before any requests are throttled. If administrators do not want to risk requests getting throttled, they can enable auto scaling. With DynamoDB auto scaling, a table can increase its provisioned read and write capacity to handle sudden increases in

Measurement	Description	Measurement Unit	Interpretation
			traffic, without request throttling. When the workload decreases, DynamoDB auto scaling can decrease the throughput so that administrators do not pay for unused provisioned capacity. Alternatively, administrators can purchase reserved capacity in advance. By reserving read and write capacity units ahead of time, administrators can realize significant cost savings compared to on-demand provisioned throughput settings.
Read capacity	Indicates the percentage of read capacity consumed by this table.	Percent	
Read throttle events	Indicates the number of read events which exceeded provisioned read throughput of this table.	Number	A low value for this measure indicates that the table is well-tuned. If the value of this measure is very high or is increasing consistently, it indicates that many reads are exceeding the provisioned read throughput, and are hence being throttled. If too many read requests to a table are found to get throttled, then use the detailed diagnosis of this measure to determine what type of read requests to that table are getting throttled the most - are they requests to the table? or are they requests to the global secondary index on the table? To avoid read throttle events, administrators should make sure that the table/index is provisioned with adequate read throughput, based on the size of items and the performance level they expect from it. For instance, suppose that administrators want to read 80 items per second from a table containing items of 3 KB in size. Say that they want strongly consistent reads. For this scenario, administrators have to set the table's provisioned read throughput to 80 read capacity units. Alternatively, administrators can enable auto

Measurement	Description	Measurement Unit	Interpretation
			scaling for the table. With DynamoDB auto scaling, a table can increase its provisioned read and write capacity to handle sudden increases in traffic, without request throttling. When the workload decreases, DynamoDB auto scaling can decrease the throughput so that administrators do not pay for unused provisioned capacity. Another option is to purchase reserved capacity in advance. By reserving read and write capacity units ahead of time, administrators can realize significant cost savings compared to on-demand provisioned throughput settings.
Provisioned write capacity	Indicates the number of write capacity units provisioned for this table.	Number	The value of this measure is the sum of write capacity units provisioned for the table and the global secondary index on that table. You can use the detailed diagnosis of this measure to know how many write capacity units have been provisioned to this table and to the global secondary index on this table. One write capacity unit represents one write per second for an item up to 1 KB in size. If you need to write an item that is larger than 1 KB, DynamoDB will need to consume additional write capacity units. The total number of write capacity units required depends on the item size.
Consumed write capacity	Indicates the number of write capacity units consumed by this table.	Number	Tracking changes in write consumed capacity allows administrators to spot abnormal peaks or drops in write activities. In particularly, administrators can make sure that consumption does not exceed the value of the Provisioned write capacity measure. The value of the Consumed write capacity measure for a table is the sum of the write capacity units consumed by that table and by

Measurement	Description	Measurement Unit	Interpretation
			its global secondary index. To know how many write capacity units were consumed by the table and by the index, use the detailed diagnosis of this measure. If the value of the Consumed write capacity measure for a table is close to or equal to the value of the Provisioned write capacity measure, or if the value of the Write capacity measure for a table is close to or equal to 100%, it indicates that the write requests to that table have been or will potentially be throttled. Throttling prevents an application from consuming too many capacity units. When a request is throttled, it fails with an HTTP 400 code (Bad Request) and a ProvisionedThroughputExceededException. Administrators might want to set up a first alert before a table consumes its entire capacity —for instance, they can configure an alert to be triggered at a threshold of 80%. This would give them time to scale up capacity before any requests are throttled. If administrators do not want to risk requests getting throttled, they can enable auto scaling. With DynamoDB auto scaling, a table can increase its provisioned read and write capacity to handle sudden increases in traffic, without request throttling. When the workload decreases, DynamoDB auto scaling can decrease the throughput so that administrators do not pay for unused provisioned capacity. Alternatively, administrators can purchase reserved capacity in advance. By reserving read and write capacity units ahead of time, administrators can realize significant cost savings compared to on-demand provisioned throughput settings.

Measurement	Description	Measurement Unit	Interpretation
Write capacity	Indicates the percentage of provisioned write capacity that has been consumed by this table.	Percent	
Write throttle events	Indicates the number of write events which exceeded the write throughput provisioned for this table.	Number	A low value for this measure indicates that the table is well-tuned. If the value of this measure is very high or is increasing consistently, it indicates that many writes are exceeding the provisioned write throughput, and are hence being throttled. If too many write requests to a table are found to be throttled, then use the detailed diagnosis of this measure to determine what type of write requests to that table are getting throttled the most - are they requests to the table? or are they requests to the global secondary index on the table? Throttling prevents applications from consuming too many capacity units. When a request is throttled, it fails with an HTTP 400 code (Bad Request) and a ProvisionedThroughputExceededException. To avoid write throttle events, administrators should make sure that the table/index is provisioned with adequate write throughput, based on the size of items and the performance level they expect from it. For instance, suppose that administrators want to write 100 items per second to a table, and that the items are 512 bytes in size. For this scenario, administrators have to set the table's provisioned write throughput to 100 write capacity units. Alternatively, administrators can enable auto scaling for the table. With DynamoDB auto scaling, a table can increase its provisioned

Measurement	Description	Measurement Unit	Interpretation
			read and write capacity to handle sudden increases in traffic, without request throttling. When the workload decreases, DynamoDB auto scaling can decrease the throughput so that administrators do not pay for unused provisioned capacity. Another option is to purchase reserved capacity in advance. By reserving read and write capacity units ahead of time, administrators can realize significant cost savings compared to on-demand provisioned throughput settings.
Onlineindex consumed write capacity	Indicates the number of write capacity units consumed when adding a new global secondary index to this table.	Number	This metric should be monitored when a new Global Secondary Index is being added so you can be aware if you did not provision enough capacity. If that's the case, incoming write requests happening during the index building phase might be throttled which will severely slow down its creation and cause upstream delays or problems. You should then adjust the index's write capacity using the UpdateTable operation, which can be done even if the index is still being built. **Note:** This metric does not take into account ordinary write throughput consumed during index creation. To know how much write capacity was consumed ordinarily by the table during index creation, use the detailed diagnosis for this measure; this reveals how many write capacity units were consumed by the table and by the index.
Onlineindex percentage progress	Indicates the percentage of completion when a new global secondary index is being added	Percent	This metrics allows you to follow the progress of the creation of a Global Secondary Index. You should keep an eye on this metric and correlate it with the rest of your DynamoDB metrics to make sure the index creation does

Measurement	Description	Measurement Unit	Interpretation
	to this table.		not impact overall performance. If the index takes too much time to build, it might be due to throttled events so you should check the Onlineindex throttle events metric. DynamoDB must first allocate resources for the new index, and then backfill attributes from the table into the index. For large tables, this process might take a long time. You should monitor this statistic to view the relative progress as DynamoDB builds the index. You can use the detailed diagnosis of this measure to know the percentage of progress of global secondary index creation and table creation. This way, you can assess the impact of index creation on table creation time.
Onlineindex throttle events	Indicates the number of write throttle events that occur when adding a new global secondary index to this table.	Number	Write-throttled events happening when adding a new Global Secondary Index to a table can dramatically slow down its creation. In such a situation, use the detailed diagnosis of this measure to know the count of write throttle events that occurred when adding the index to the table and the count of write throttle events that occurred ordinarily on the table. This will point you to the type of throttle events that are actually delaying table creation. If this metric is not equal to zero, adjust the index's write capacity using UpdateTable. You can prevent throttling by properly monitoring the measure Onlineindex write capacity. You can also use the detailed diagnosis of this measure to know the detailed information on the throttle events. **Note:** The Write throttle Events metric does not

Measurement	Description	Measurement Unit	Interpretation
			count the throttle events happening during index creation.
Returned items	Indicates the number of items returned by a Scan or Query operation to this table.	Number	Use the detailed diagnosis of this measure to view the count of items returned per operation. This will point you to that operation that returned the maximum number of items.
Successful request latency	Indicates the response time (in milliseconds) of successful requests to this table.	Secs	If you see this number increasing above normal levels, you should quickly investigate since it can significantly impact your application's performance. It can be caused by network issues, or requests taking too much time due to your table design. In this case, using Global secondary indexes can help maintain reasonable performance. In the event of high latency, you can also use the detailed diagnosis of this measure to figure out the latency per operation. This will point you to those most latent operations on the table.
System errors	Indicates the number of requests to this table resulting in an HTTP 500 error.	Number	This metric should always be equal to zero. If it is not, you may want to get involved - perhaps restarting portions of the service, temporarily disabling some functionality in your application, or getting in touch with AWS support.
Throttled requests	Indicates the number of user requests to this table, containing at least one event that exceeded your provisioned throughput	Number	Compare the value of this measure across tables to know which table received the maximum requests containing multiple throttle events. If too many requests to a table are getting throttled, use the detailed diagnosis of this measure to identify the precise operations on that table that are being affected by the throttling. The detailed diagnostics list the number of throttled events per operation, thus leading you to that operation that is being throttled the most.

Measurement	Description	Measurement Unit	Interpretation
			When a request is throttled, it fails with an HTTP 400 code (Bad Request) and a ProvisionedThroughputExceededException. To avoid throttle events, administrators should make sure that the table is provisioned with adequate read and throughput, based on the size of items in that table and the performance level they expect from it.
User errors	Indicates the number of requests to this table that are generating an HTTP 400 error.	Number	If your client application is interacting correctly with DynamoDB, this metric should always be equal to zero. It is incremented for any 400 error except for ProvisionedThroughputExceededException, ThrottlingException, and ConditionalCheckFailedException. It is usually due to a client error such as an authentication failure.
Returned data	Indicates the amount of data returned by GetRecords operations (Amazon DynamoDB Streams) performed on this table.	KB	Use the detailed diagnosis of this measure to view the amount of data returned per operation. This will point you to that operation that returned the maximum data.
Returned records	Indicates the number of stream records returned by GetRecords operations (Amazon DynamoDB Streams) performed on this table.	Number	Use the detailed diagnosis of this measure to view the number of records returned per operation. This will point you to that operation that returned the maximum number of records.

4.5.2 AWS ElastiCache Test

Amazon ElastiCache is a web service that improves the performance of web applications by allowing you to retrieve information from a fast, managed, in-memory system, instead of relying entirely on slower disk-based databases. Using Amazon ElastiCache, you can not only improve load

and response times to user actions and queries, but also reduce the cost associated with scaling web applications.

Amazon ElastiCache supports the Memcached and Redis cache engines.

- **Redis** - a fast, open source, in-memory data store and cache. Amazon ElastiCache for Redis is a Redis-compatible in-memory service that delivers the ease-of-use and power of Redis along with the availability, reliability and performance suitable for the most demanding applications.
- **Memcached** - a widely adopted memory object caching system. ElastiCache is protocol compliant with Memcached, so popular tools that you use today with existing Memcached environments will work seamlessly with the service.

Using Amazon ElastiCache, you can create and manage cache clusters. The key components of this cluster are nodes and shards (Redis). A node is the smallest building block of an ElastiCache deployment. Each node is a fixed-size chunk of secure, network-attached RAM. A Redis shard (called a node group in the API and CLI) is a grouping of 1–6 related nodes. A Redis (cluster mode disabled) cluster always has one shard. A Redis (cluster mode enabled) cluster can have from 1–15 shards. A Redis cluster is a logical grouping of one or more ElastiCache Shards (Redis). Data is partitioned across the shards in a Redis (cluster mode enabled) cluster. A Memcached cluster is a logical grouping of one or more ElastiCache Nodes. Data is partitioned across the nodes in a Memcached cluster. Once a cluster is provisioned, Amazon ElastiCache automatically detects and replaces failed nodes, providing a resilient system that mitigates the risk of overloaded databases, which slow website and application load times.

If a cluster is unavailable or is not sized with adequate resources for cache operations, then the cache cluster will not be able to service requests from applications. This can cause the cache hit ratio to fall drastically, thus increasing datastore accesses and related processing overheads. Consequently, request processing will slow down and application performance will suffer. To avoid this, administrators need to track the status, usage, and resource consumption of a cache cluster and its nodes, proactively detect abnormalities, and promptly fix them. This is where the **AWS ElastiCache** test helps!

By default, this test auto-discovers the cache clusters that have been created and launched. For each cluster, the test then reports the following:

- The status of the cluster;
- The load on the cluster, in terms of the number and type of requests received by it;
- How much CPU and memory was consumed by the cluster when servicing the requests;

- How well the cache served the different type of requests (Check and set, Decrement, Delete, Increment, Config get, Config set, Get);

In the process, the test points to unavailable clusters, irregularities in request processing by a cluster, and inadequacies in cache size. The number and type of requests that the cluster was unable to serve are highlighted. Moreover, using the performance results reported by the test, administrators can also receive useful pointers on how to resize the cluster to optimize cache performance.

Optionally, you can configure the test to report metrics for each cluster node, instead of cluster. The node-level analytics will help administrators quickly identify the unavailable nodes and resource-thin nodes in the cluster.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each cluster / cluster node

First-level descriptor: AWS Region

Second-level descriptor: Cluster / cluster node, depending upon the option chosen against the **ElastiCache Filter Name** parameter.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not

Parameter	Description
	configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
ElastiCache Filter Name	By default, this parameter is set to **CacheClusterId**. In this case, this test will report metrics for cache cluster that is created and launched. If required, you can override this default setting by setting the **ElastiCache Filter Name** to **CacheNodeId**. In this case, the test will report metrics for every cluster node.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
CPU utilization	By default, this measure reports the percentage CPU utilization of this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeI,** then this measure reports the percentage CPU utilization of this node.	Percent	Typically, a high value for this measure is a sign of excessive CPU usage by a cluster/node. It could also hint at a potential CPU contention at the cluster / node-level. In case of a cluster, the cache engine used determines how high the CPU usage can be and what its implications are: • **Memcached:** Since Memcached is multi-threaded, this metric can be as high as 90%. If you exceed this

Measurement	Description	Measurement Unit	Interpretation
			threshold, scale your cache cluster up by using a larger cache node type, or scale out by adding more cache nodes. • **Redis:** Since Redis is single-threaded, the threshold is calculated as (90 / number of processor cores). For example, suppose you are using a cache.m1. xlarge node, which has four cores. In this case, the threshold for CPU Utilization would be (90 / 4), or 22.5%. You will need to determine your own threshold, based on the number of cores in the cache node that you are using in redis. f you exceed this threshold, and your main workload is from read requests, scale your cache cluster out by adding read replicas. If the main workload is from write requests, depending on your cluster configuration, we recommend that you: ○ **Redis (cluster mode disabled) clusters:** scale up by using a larger cache instance type. ○ **Redis (cluster mode enabled) clusters:** add more shards to distribute the write workload across more primary nodes.
Freeable memory	By default, this measure reports amount of free memory available to this cluster. If the ElastiCache Filter	MB	A high value is desired for this measure. A steady and significant drop in the value for this measure indicates a memory contention on the cluster/node.

Measurement	Description	Measurement Unit	Interpretation
	Name parameter is set to **CacheNodeId** then this measure reports the amount of free memory on this node.		
Incoming network traffic	By default, this measure reports the rate at which this cluster has read from the network. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the rate at which this node reads from the network.	KB/Sec	
Outgoing network traffic	By default, this measure reports the rate at which this cluster has written to the network. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the rate at which this node has written to the network.	Number	
Swap usage	By default, this measure reports the amount of swap space used by this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the amount of swap space used by this node.	KB	For a memcached cluster, the value of this measure should not exceed 50 MB. If it does, we recommend that you increase the ConnectionOverhead parameter value.
Status	By default, this measure	Number	The values that this measure can report

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td>reports the whether/not this cluster is available.

If the ElastiCache Filter Name parameter is set to CacheNodeId then this measure reports whether/not this node is available.</td><td></td><td>and the states they represent are listed in the table below:

<table><tr><th>Measure Value</th><th>State</th></tr><tr><td>0</td><td>Available</td></tr><tr><td>1</td><td>Creating</td></tr><tr><td>2</td><td>Modifying</td></tr><tr><td>3</td><td>Rebooting</td></tr><tr><td>4</td><td>Cache cluster nodes</td></tr><tr><td>5</td><td>Incompatible - network</td></tr><tr><td>6</td><td>Snapshotting</td></tr><tr><td>7</td><td>Restore-failed</td></tr><tr><td>8</td><td>Deleting</td></tr><tr><td>9</td><td>Deleted</td></tr></table></td></tr>
<tr><td>No of nodes</td><td>Indicates the number of nodes in this cluster.</td><td>Number</td><td>This measure is reported only for memcached clusters.

This measure is not reported for a node - i.e., if the Elastic Filter Name parameter is set to 'CacheNodeId'.</td></tr>
<tr><td>Current connections</td><td>Indicates the current number of connections to this cluster.</td><td>Number</td><td>This measure is reported only for a cluster - i.e., if the ELASTIC FILTER NAME parameter is set to 'CacheClusterId'.

This is a cache engine metric, published for both Memcached and Redis cache clusters. We recommend that you determine your own alarm threshold for this metric based on your</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			application needs. Whether you are running Memcached or Redis, an increasing number of CurrConnections might indicate a problem with your application; you will need to investigate the application behavior to address this issue.
Current items	By default, this measure reports the total number of items currently stored in this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of items in this node.	Number	
Reclaimed	By default, this measure reports the number of expired items this cluster evicted to allow space for new writes. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of expired items that were evicted from this node to allow space for new writes.	Number	In case there are no free chunks, or no free pages in the appropriate slab class, Amazon ElastiCache will look at the LRU in tail to reclaim an item. Basically, it will search the last few items in the “end” and identifies the ones that are already expired, and reclaims it - i.e., makes it free for reuse. A high value for this measure therefore implies that the cache is running out of memory. You may want to check the value of the Freeable memory measure to corroborate this finding. If the value of this measure is very low, while Freeable memory is also low, it means that there are very few expired items in the cache to be reclaimed. This potentially means that very shortly, there may not be any expired items at

Measurement	Description	Measurement Unit	Interpretation
			the end of the LRU to be reused. In such a situation, ElastiCache will evict an item that has not expired. This can result in the loss of frequently-accessed items from the cache. If the situation persists, it will seriously undermine cache performance. To avoid this, you should increase the memory capacity of a memcached cluster by adding more nodes to it or by using a larger node type; for a redis cluster, use a larger node type. Alternatively, you can configure a memcached cluster to send out an error message instead of evicting items (expired or non-expired), whenever it has no more memory to store items. For this, turn on the error_on_memory_ exhausted flag of memacached.
Evictions	Indicates number of non-expired items this cluster evicted to allow space for new writes.	Number	**This measure is reported only for a cache cluster - i.e., this measure is reported only if the Elastic Filter Name is set to 'CacheClusterId'**. Typically, items are evicted from Amazon ElastiCache if they are expired or the slab class is completely out of free chunks and there are no free pages to assign to a slab class. In case there are no free chunks, or no free pages in the appropriate slab class, Amazon ElastiCache will look at the LRU in tail to reclaim an item. Basically, it will search the last few items in the “end” and identifies the ones that are already expired, and makes it free for reuse. If it cannot find an expired item on the end, it will "evict" one which has not yet

Measurement	Description	Measurement Unit	Interpretation
			expired. Actually you could end up with one slab class constantly evicting recently used items, on the other hand another slab having a bunch of old items that just sit around. For example: When we need a 104 byte chunk, it will evict a 104 byte chunk, even though there might be a 280 byte chunk that is even older. This explains the internal workings that "Each slab-class has its own LRU and statistical counter updates, it behaves like a separate cache itself, it is not global LRU, but slab class LRU in short". We recommend that you determine your own alarm threshold for this metric based on your application needs. • **Memcached:** If you exceed your chosen threshold, scale your cluster up by using a larger node type, or scale out by adding more nodes. • **Redis:** If you exceed your chosen threshold, scale your cluster up by using a larger node type.
New connections	By default, this measure reports the number of new connections this cluster has received. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of new	Number	This measure derived from the memcached total_connections statistic by recording the change in total_ connections across a period. This will always be at least 1, due to a connection reserved for an ElastiCache.

Measurement	Description	Measurement Unit	Interpretation
	connections this node has received.		
Data used for cache items	By default, this measure reports the amount of memory in this cluster that has been used to store items. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of new connections this node has received amount of memory used to store items in this node.	KB	If the value of this measure keeps increasing over time, it implies that the items are consuming memory excessively. Keep checking if the cluster/node has sufficient memory to hold new items. If not, cache performance will be adversely impacted, thus degrading application performance as well. To increase the memory capacity of a cluster, add more nodes to a cluster or add nodes of a larger node type. To increase the memory capacity of a memcached node, you may have to increase the max_cache_memory and/or memcached_connections_ overhead parameters of that node. max_cache_memory sets the total amount of memory available on a node. The memcached_connections_ overhead is the memory used for connections and other overheads. The memory available for storing items is the difference between the max_ cache_memory and memcached_ connections_overhead. By increasing the max_cache_memory and/or by reducing the memached_connections_ overhead, you can make more memory available for storing items. To increase the memory capacity of a redis node, you may have to increase the maxmemory and/or reserved-memory parameters of that node. maxmemory sets the total amount of

Measurement	Description	Measurement Unit	Interpretation
			memory available on a node. The reserved-memory is the memory that is set aside for non-data purposes. The memory available for storing items is the difference between the maxmemory and reserved-memory. By increasing the maxmemory and/or by reducing the reserved-memory, you can make more memory available for storing items.
Data read into memcached	By default, this measure reports the amount of data that this cache cluster read from the network. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the amount of data that this node read from the network.	KB	**This measure is reported only for the memcached engine.**
Data written out from memcached	By default, this measure reports the amount of data that this cache cluster wrote to the network. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the amount of data that this node wrote to the network.	KB	**This measure is reported only for the memcached engine.**
Check and set bad val	By default, this measure reports the number of CAS (check and set) requests that this cluster received, where the Cas value did not match with the Cas value stored.	Number	**This measure is reported only for the memcached engine.**

Measurement	Description	Measurement Unit	Interpretation
	If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of CAS (check and set) requests that this node received, where the Cas value did not match with the Cas value stored.		
Check and set hits	By default, this measure reports the number of CAS (check and set) requests that this cluster received, where the requested key was found and the Cas value matched. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of CAS (check and set) requests that this node received, where the requested key was found and the Cas value matched.	Number	**This measure is reported only for the memcached engine.**
Check and set misses	By default, this measure reports the number of CAS (check and set) requests that this cluster received, where the key requested was not found. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of CAS (check	Number	**This measure is reported only for the memcached engine.** In the event of poor cache performance, you can compare the value of this measure with that of the Touch request missed hits, Increment request missed hits, Get request missed hits, Delete request missed hits, and Decrement request missed hits measure to know what type of requests the cache has been unable to serve most of the time.

Measurement	Description	Measurement Unit	Interpretation
	and set) requests that this node received, where the key requested was not found.		
Flush commands	By default, this measure reports the number of flush commands this cluster has received. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of flush commands this node received.	Number	**This measure is reported only for the memcached engine.**
Get commands	By default, this measure reports the number of Get commands this cluster has received. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of Get commands this node received.	Number	**This measure is reported only for the memcached engine.**
Decrement hits	By default, this measure reports the number of decrement requests this cluster has received, where the requested key was found. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of decrement requests this node has	Number	**This measure is reported only for the memcached engine.** A high value is desired for this measure.

Measurement	Description	Measurement Unit	Interpretation
	received, where the requested key was found.		
Decrement misses	By default, this measure reports the number of decrement requests this cluster has received, where the requested key was not found. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of decrement requests this node has received, where the requested key was not found.	Number	**This measure is reported only for the memcached engine.** In the event of poor cache performance, you can compare the value of this measure with that of the Check and set request missed, Touch request missed hits, Increment request missed hits, Get request missed hits, and Delete request missed hits measures to know what type of requests the cache has been unable to serve most of the time.
Delete hits	By default, this measure reports the number of delete requests this cluster has received, where the requested key was found. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of delete requests this node has received, where the requested key was found.	Number	**This measure is reported only for the memcached engine.** A high value is desired for this measure.
Delete misses	By default, this measure reports the number of delete requests this cluster has received, where the requested key was not found.	Number	**This measure is reported only for the memcached engine.** In the event of poor cache performance, you can compare the value of this measure with that of the Check and set

Measurement	Description	Measurement Unit	Interpretation
	If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of delete requests this node has received, where the requested key was not found.		request missed, Touch request missed hits, Increment request missed hits, Get request missed hits, and Decrement request missed hits measures to know what type of requests the cache has been unable to serve most of the time.
Get hits	By default, this measure reports the number of Get requests this cluster has received, where the requested key was found. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of Get requests this node has received, where the requested key was found.	Number	**This measure is reported only for the memcached engine.** A high value is desired for this measure.
Get misses	By default, this measure reports the number of Get requests this cluster has received, where the requested key was not found. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of Get requests this node has received, where the requested key was not found.	Number	**This measure is reported only for the memcached engine.** In the event of poor cache performance, you can compare the value of this measure with that of the Check and set request missed, Touch request missed hits, Increment request missed hits, Decrement request missed hits, and Delete request missed hits measures to know what type of requests the cache has been unable to serve most of the time.
Increment hits	By default, this measure reports the number of	Number	**This measure is reported only for the memcached engine.**

Measurement	Description	Measurement Unit	Interpretation
	increment requests this cluster has received, where the requested key was found. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of increment requests this node has received, where the requested key was found.		A high value is desired for this measure.
Increment misses	By default, this measure reports the number of increment requests this cluster has received, where the requested key was not found. If the ElastiCache Filter Name parameter is set to **CacheNodeId,** then this measure reports the number of increment requests this node has received, where the requested key was not found.	Number	**This measure is reported only for the memcached engine**. In the event of poor cache performance, you can compare the value of this measure with that of the Check and set request missed, Touch request missed hits, Decrement request missed hits, Get request missed hits, and Delete request missed hits measures to know what type of requests the cache has been unable to serve most of the time.
Data used for hash	By default, this measure reports the amount of data in this cluster that is currently used by hash tables. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the amount of data in this	MB	**This measure is reported only for the memcached engine.**

Measurement	Description	Measurement Unit	Interpretation
	node that is currently used by hash tables.		
Command configGet	By default, this measure reports the cumulative number of config 'get' requests to this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of config 'get' requests to this node.	Number	**This measure is reported only for the memcached engine.**
Command configSet	By default, this measure reports the cumulative number of config 'set' requests to this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the cumulative number of config 'set' requests to this node.	Number	**This measure is reported only for the memcached engine.**
Command touch	By default, this measure reports the cumulative number of 'touch' requests to this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the cumulative number of touch requests to this node.	Number	**This measure is reported only for the memcached engine.**
Current configurations	By default, this measure reports the number of	Number	**This measure is reported only for the memcached engine.**

Measurement	Description	Measurement Unit	Interpretation
	configurations currently stored in this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of configurations currently stored in this node.		
Evicted unfetched	By default, this measure reports number of valid items evicted from the least recently used cache (LRU) of this cluster, which were never touched after being set. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of valid items evicted from the least recently used cache (LRU) of this node, which were never touched after being set.	Number	**These measures are reported only for the memcached engine.** If you store an item and it expires, it still sits in the LRU cache at its position. If that item is not fetched by any request, then it falls to the end of the cache and is then picked up for reuse. However, if you fetch an expired item, memcached will find that the item is expired and free its memory for reuse immediately. This means that unfetched items in the LRU take longer to be evicted than the ones fetched.
Expired unfetched	By default, this measure reports the number of expired items reclaimed from the LRU of this cluster, which were never touched after being set. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of expired items	Number	

Measurement	Description	Measurement Unit	Interpretation
	reclaimed from the LRU of this node, which were never touched after being set.		
Slabs moved	By default, this measure reports the total number of slab pages moved in all nodes of this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of slab pages moved in this node	Number	**This measure is reported only for the memcached engine.** Amazon ElastiCache node usually breaks the allocated memory into smaller parts called pages. Each page is usually 1 megabyte in size. Each page is then assigned to a slab class when necessary. Each slab class is in turn divided into chunks of a specific size. The chunks in each slab have the same size. There can be multiple pages assigned for each slab-class, but as soon as a page is assigned to a slab-class, it is permanent.
Touch hits	By default, this measure reports the number of keys in this cluster that were touched and given a new expiration time. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of keys in this node that were touched and given a new expiration time.	Number	**This measure is reported only for the memcached engine.**
Touch misses	By default, this measure reports the number of keys in this cluster that have been touched, but were not found.	Number	**This measure is reported only for the memcached engine.** In the event of poor cache performance, you can compare the value of this measure with that of the Check and set

Measurement	Description	Measurement Unit	Interpretation
	If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of keys in this node that have been touched, but were not found.		request missed, Decrement request missed hits, Get request missed hits, Increment request missed hits, and Delete request missed hits measures to know what type of requests the cache has been unable to serve most of the time.
New items	By default, this measure reports the number of new items stored in this cluster during the last measurement period. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of new items stored in this node during the last measurement period.	Number	**This measure is reported only for the memcached engine.**
Unused memory	By default, this measure represents the amount of memory in this cluster that can be used to store items. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the amount of memory in this node that can be used to store items.	MB	**This measure is reported only for the memcached engine.** A consistent drop in the value of this measure is a cause for concern, as it indicates that there is not enough free memory in the cache to store new items. This can cause the cache hit ratio to drop steeply, which in turn can affect application performance. To avoid this, you may have to increase the memory capacity of the cluster by adding more nodes to it or by adding a large node type. At the node-level, memory leakages can cause a serious memory contention. To avoid memory leakages

Measurement	Description	Measurement Unit	Interpretation
			in a memcached cluster, you need to have a decent understanding of how the memory internals of a node work. A memcached node usually breaks the allocated memory into smaller parts called pages. Each page is usually 1 megabyte in size. Each page is then assigned to a slab class when necessary. Each slab class is in turn divided into chunks of a specific size. The chunks in each slab have the same size. For instance, you can have a page that is assigned to say, slab class 1, which contains 13,107 chunks of 80 bytes each. When you are storing items in Amazon ElastiCache, they are pushed into the slab class of the nearest fit. For instance, in the example above, if an item of size 70 bytes is to be stored in the cache, it will go into slab class 1, causing an overhead loss of 10 bytes per item. If you are running Amazon ElastiCache clusters spanning in Hundreds of GB or TB, you will end up losing lots of allocated memory as overheads this way. This can cause a serious contention for memory resources. To avoid this, it is imperative that the chunk size and growth factor of the chunks is appropriately set. These two factors are governed by chunk_size and chunk_size_growth_factor parameters of a Memcached cluster. chunk_size is the minimum amount, in bytes, of space to allocate for the smallest item's key, value, and flags. By default, this is 48 bytes. chunk_

Measurement	Description	Measurement Unit	Interpretation
			size_growth_factor is the growth factor that controls the size of each successive memcached chunk; each chunk will be chunk_size_growth_ factor times larger than the previous chunk. By default, this is set to 1.25. For best performance, you should keep the chunk sizes closer to your item sizes. This means that if item sizes are big and predictable it is recommended to have bigger chunks and growth factors. If the item sizes are varied, it is better to have smaller initial size and growth factor. This will keep the wastage minimal and increase free memory.
Data used for cache	By default, this measure indicates the amount of memory allocated to this node for cache usage.	MB	**This measure is reported only for the redis engine.** A Redis node will grow until it consumes the maximum memory configured for that node - i.e., the value set against its maxmemory parameter. If this occurs, then node performance will likely suffer due to excessive memory paging. By reserving memory you can set aside some of the available memory for non-Redis purposes to help reduce the amount of paging. Use the reserved-memory parameter for this purpose.
Cache hits	By default, this measure indicates the number of successful key lookups in this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the	Number	**This measure is reported only for the redis engine.** A high value is desired for this measure.

Measurement	Description	Measurement Unit	Interpretation
	number of successful key lookups in this node.		
Cache misses	By default, this measure indicates the number of unsuccessful key lookups in this node. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the number of unsuccessful key lookups in this node.	Number	**This measure is reported only for the redis engine.** Ideally, the value of this measure should be very low. If this value is higher than the value of the Cache request hits measure, it implies poor cache performance.
Hyperloglog based commands	By default, this measure indicates the total number of HyperLogLog commands received by this cluster. If the ElastiCache Filter Name parameter is set to **CacheNodeId**, then this measure reports the total number of HyperLogLog commands received by this node.	Number	**This measure is reported only for the redis engine.** This measure is the sum of all pf type commands (pfadd, pfcount, pfmerge) received by a cluster/node.
Replication lag	Indicates how far behind, in seconds, this replica is in applying changes from the primary cache cluster.	Secs	**This measure is reported only for a redis node running as a read replica.** 'Inconsistency" or lag between a read replica and its primary cache node is common with Redis asynchronous replication. If an existing read replica has fallen too far behind to meet your requirements, you can reboot it. Keep in mind that replica lag may naturally grow and shrink over time, depending on your primary cache node's steady-state usage pattern.

Measurement	Description	Measurement Unit	Interpretation
Replication bytes	Indicates the amount of data that the primary is sending to all its replicas.	KB	**This measure is reported only for a primary in a redis cluster.** This metric is representative of the write load on the replication group. For replicas and standalone primaries, replication is always 0.
Is background save in progress?	Indicates whether/not a background save is in progress on this node.	Number	**This measure is reported only for a redis node.** This measure reports 1 whenever a background save (forked or forkless) is in progress, and 0 otherwise. A background save process is typically used during snapshots and syncs. These operations can cause degraded performance. With the help of this measure, you can diagnose whether or not degraded performance was caused by a background save process.
Get type commands	Indicates the total number of get type of commands received by this node.	Number	**This measure is reported only for a redis node.** This is derived by summing all the get types of commands (get, mget, hget, etc.).
Hash based commands	Indicates the total number of hash-based commands received by this node.	Number	**This measure is reported only for a redis node.** This is derived by summing all the commands that act upon one or more hashes.
Key based commands	Indicates the total number of key-based commands received by this node.	Number	**This measure is reported only for a redis node.** This is derived by summing all the commands that act upon one or more keys.
List based commands	Indicates the total number of list-based commands	Number	**This measure is reported only for a redis node.**

Measurement	Description	Measurement Unit	Interpretation
	received by this node.		This is derived by summing all the commands that act upon one or more lists.
Set based commands	Indicates the total number of set-based commands received by this node.	Number	**This measure is reported only for a redis node.** This is derived by summing all the commands that act upon one or more sets.
Set type commands	Indicates the total number of set type of commands received by this node.	Number	**This measure is reported only for a redis node.** This is derived by summing all the set types of commands (set, hset, etc.).
Sortedset based commands	Indicates the total number of sorted set-based commands received by this node.	Number	**This measure is reported only for a redis node.** This is derived by summing all the commands that act upon one or more sorted sets.
String based commands	Indicates the total number of string-based commands received by this node.	Number	**This measure is reported only for a redis node.** This is derived by summing all the commands that act upon one or more strings.
Is master?	Indicates whether/not this node is the master node in a redis cluster.		**This measure is reported only for a redis node.** The values that this measure can report and their corresponding numeric values are listed below: **Measure Value** / **Numeric Value** Yes / 1 No / 0 **Note:**

Measurement	Description	Measurement Unit	Interpretation
			By default, this measure reports one of the **Measure Value**s listed above to indicate whether/not a redis node is the master. In the graph of this measure however, the same is indicated using the numeric equivalents.

4.5.3 AWS RedShift Test

Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. The first step to create such a data warehouse is to launch an Amazon Redshift cluster. An Amazon Redshift cluster is a collection of computing resources called nodes. Each cluster runs an Amazon Redshift engine and contains one or more databases. Each cluster has a leader node and one or more compute nodes. The leader node receives queries from client applications, parses the queries, and develops query execution plans. The leader node then coordinates the parallel execution of these plans with the compute nodes, aggregates the intermediate results from these nodes, and finally returns the results back to the client applications. Compute nodes execute the query execution plans and transmit data among themselves to serve these queries. The intermediate results are sent back to the leader node for aggregation before being sent back to the client applications.

Where RedShift is in use, query performance, and consequently, the performance of the dependent client applications, depends upon the following factors:

- Cluster availability
- How the cluster and its nodes use the CPU, network, and storage resources of the cluster;
- Responsiveness of the nodes in the cluster to I/O requests from client applications

To be able to accurately assess whether cluster performance is at the desired level or not, an administrator would require real-time insights into each of the factors listed above. The AWS RedShift test provides administrators with these valuable insights. By reporting the current health status of each cluster managed by RedShift, this test brings unavailable clusters to light. The resource usage of the cluster is also reported, so that potential resource contentions can be proactively isolated. Optionally, you can also configure this test to report metrics for individual nodes in the cluster as well. If this is done, then administrators will be able to instantly drill-down from a resource-hungry cluster to the exact node in the cluster that could hogging the resources. At the node-level, the latency and throughput of each node is also revealed. This way, when users complain of degradation in the performance of client applications, you can quickly identify the cluster

and the precise node in the cluster that is slowing down I/O processing and consequently, impacting application performance.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each cluster and/or node in every AWS region on the cloud monitored

First level descriptor: AWS Region

Second level descriptor: Cluster

Third-level descriptor: Node

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.

Parameter	Description
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
RedShift Filter Name	By default, this test reports metrics only for each cluster in each AWS region on the cloud. This is why, this flag is set to **ClusterIdentifier**, by default. If needed, you can configure the test to additionally report metrics for every node in every cluster. For node-level metrics, select the **NodeIdentifer** option from this drop-down. Upon selection, you will be able to view metrics both at the cluster-level and the node-level.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
CPU utilization:	Indicates the percentage of CPU utilized by this cluster/node.	Percent	For a cluster, this measure will report the aggregate CPU usage of all nodes in the cluster. If the value of this measure is consistently above 50% for a cluster, it indicates that a serious resource contention may occur on that cluster, if additional processing power is not provided to it. In such a case, you may want to consider adding more nodes to the cluster, or adding more CPUs to the existing nodes. You can also compare the CPU usage of nodes in the resource-hungry cluster to determine whether one/more nodes are hogging the CPU. If so, you may want to tweak the load-balancing algorithm of your cluster to ensure uniform load distribution.
Database	Indicates the number	Number	**This measure is only reported at the**

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>connections:</td><td>of connections to the databases in this cluster.</td><td></td><td>cluster-level and not the node-level.</td></tr>
<tr><td>Health status:</td><td>Indicates the current health status of this cluster.</td><td>Percent</td><td>This measure is only reported at the cluster-level and not the node-level.

Every minute the cluster connects to its database and performs a simple query. If it is able to perform this operation successfully, then the value of this measure will be Healthy. Otherwise, the value of this measure will be Unhealthy. An Unhealthy status can occur when the cluster database is under extremely heavy load or if there is a configuration problem with a database on the cluster.

The numeric values that correspond to the measure values mentioned above are as follows:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Healthy</td><td>1</td></tr><tr><td>Unhealthy</td><td>0</td></tr></table>
Note:

This measure will report one of the Measure Values listed above to indicate the current state of a cluster. In the graph of this measure however, cluster status will be indicated using the numeric equivalents only.</td></tr>
<tr><td>Is maintenance mode?:</td><td>Indicates whether/not this cluster is in the maintenance mode presently.</td><td></td><td>The values that this measure can report and their corresponding numeric values are listed in the table below:</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Yes</td><td>1</td></tr>
<tr><td>No</td><td>0</td></tr>
</table>
Note:

This measure will report one of the Measure Values listed above to indicate whether/not a cluster is in the maintenance mode. In the graph of this measure however, the same will be indicated using the numeric equivalents only.
This measure is reported only at the cluster-level and not the node-level.
Even though your cluster might be unavailable due to maintenance tasks, the Health status measure of the test will report the value Healthy for that cluster

</td></tr>
<tr><td>Network receive throughput:</td><td>Indicates the rate at which this cluster or node receives data.</td><td>KB/Secs</td><td rowspan="2">For a cluster, a consistent increase in the value of these measures is indicative of excessive usage of network resources by the cluster.
In such a case, compare the value of these measures across the nodes of a cluster to identify the nodes that are over-utilizing network bandwidth.</td></tr>
<tr><td>Network transmit throughput:</td><td>Indicates the rate at which this cluster or node sends data.</td><td>KB/Secs</td></tr>
<tr><td>Disk space used:</td><td>Indicates the percentage of disk space used by this cluster/node.</td><td>Percent</td><td>If the value of this measure is close to 100% for a cluster, it indicates that the cluster is rapidly running out of storage resources. You may want to consider adding more nodes to the cluster to increase the storage space</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			available. Alternatively, you can add fewer nodes and yet significantly increase the cluster resources by opting for node types that are by default large-sized and hence come bundled with considerable storage space. When a cluster's storage resources are rapidly depleting, you may want to compare the space usage of the nodes in cluster, so that you can quickly isolate that node that is eroding the space. Tweaking your load-balancing algorithm could go a long way in eliminating such node overloads.
Read IOPS:	Indicates the average number of disk read operations performed by this node per second.	Reads/Sec	A high value is desired for this measure, as that's the trait of a healthy node. You can compare the value of this measure across nodes to identify the node that is slowest in processing read requests.
Read latency:	Indicates the average amount of time taken by this node for disk read I/O operations.	Reads/Sec	Ideally, the value of this measure should be very low. Its good practice to compare the value of this measure across nodes of a cluster and isolate those nodes in the cluster where the value of this measure is abnormally high. Such nodes slow down I/O processing and adversely affect application performance.
Read throughput:	Indicates the average number of bytes read from disk by this node per second.	KB/Sec	A high throughput signifies faster processing of read I/O requests. A low throughput is indicative of slow read request processing. Compare the value of this measure across nodes of a cluster to isolate those nodes that have registered an abnormally low value for this measure. Such nodes not only affect cluster performance, but also the performance of dependent client applications.
Write IOPS:	Indicates the average	Writes/Sec	A high value is desired for this measure, as

Measurement	Description	Measurement Unit	Interpretation
	number o disk write operations performed by this node per second.		that's the trait of a healthy node. You can compare the value of this measure across nodes to identify the node that is slowest in processing write requests.
Write latency:	Indicates the average amount of time taken by this node for disk write I/O operations.	Secs	Ideally, the value of this measure should be very low. Its good practice to compare the value of this measure across nodes of a cluster and isolate those nodes in the cluster where the value of this measure is abnormally high. Such nodes slow down I/O processing and adversely affect application performance.
Write throughput:	Indicates the average number of bytes written to disk by this node per second.	KB/Sec	A high throughput signifies faster processing of write I/O requests. A low throughput is indicative of slow write request processing. Compare the value of this measure across nodes of a cluster to isolate those nodes that have registered an abnormally low value for this measure. Such nodes not only affect cluster performance, but also the performance of dependent client applications.

4.5.4 AWS Relational Database Service - RDS Test

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizeable capacity for an industry-standard relational database and manages common database administration tasks. It also manages backups, software patching, automatic failure detection, and recovery.

The basic building block of Amazon RDS is the DB instance. A DB instance is an isolated database environment in the cloud. A DB instance can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a stand-alone database instance.

Each DB instance runs a DB engine. Amazon RDS currently supports the MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines.

The computation and memory capacity of a DB instance is determined by its DB instance class. For each DB instance, you can select from 5 GB to 6 TB of associated storage capacity. Each DB instance class has minimum and maximum storage requirements for the DB instances that are created from it. You can select the DB instance that best meets your needs. If your needs change over time, you can change DB instances. For example, initially, you may have launched a standard (previous generation) DB instance, which provides a balance of compute, memory, and network resources for your applications. However later, based on usage, you may have realized that a burst capable - current generation DB instance, with the capability to burst to full CPU usage, is ideal for your needs. In such circumstances, RDS facilitates the switch from one type DB instance to another. But to understand which DB instance class best suits your needs and make timely and accurate adjustments to your DB instance class selection, you will have to constantly track the CPU, memory, network, and space usage of each active DB instance on the cloud and derive usage patterns. Also, to ensure optimal storage performance, you additionally need to keep an eye on the I/O operations performed on the DB instances and identify latent DB instances. This is exactly what the AWS Relational Database Service - RDS enables you to achieve.

This test closely tracks the current status, resource usage, and I/O activity of every active DB instance on each cloud region, and brings the following to light:

- Is any DB instance in an abnormal state presently?
- How are the DB instances using the CPU resources they have been configured with? Is any DB instance consuming high levels of CPU consistently? Should the DB instance class be changed?
- Does the DB instance have enough RAM? Will changing the DB instance class help in reducing the memory pressure on the instance?
- Do any db.t2 instances have a poor CPU credit balance?
- Is the disk I/O queue of any DB instance abnormally high? Which instance is this and when is I/O latency on that instance very high - when reading from or writing to the instance?
- Which DB instance is hungry for network bandwidth?
- Do all DB instances have enough free space? If not, which ones are rapidly running short of space?

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each active DB instance in each region of the AWS EC2 cloud

First-level descriptor: AWS EC2 region name

Second-level descriptor: DB instance identifier / DB instance class / DB engine name, depending upon the option you choose from the **RDS FILTER** drop-down

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Parameter	Description
RDS Filter Name	By default, this test reports metrics for each active DB instance on the cloud. This is why, this flag is set to **DBInstanceIdentifier**, by default. If needed, you can pick either of the following options from this drop-down: • **DatabaseClass:** The computation and memory capacity of a DB instance is determined by its DB instance class. If you select this option, then this test will report metrics for each DB instance class. In other words, eG will aggregate metrics for all databases that belong to a DB intance class, and will present these metrics at the macro class-level. • **EngineName:** Each DB instance runs a DB engine. Amazon RDS currently supports the MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines. Each DB engine has its own supported features, and each version of a DB engine may include specific features. If you select this option, then this test will report metrics for every DB engine. In this case, eG will aggregate metrics for all databases using a particular engine, and will present these metrics at the macro engine-level.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
RDS instance status:	Indicates the current status of this DB instance.		**This measure is reported only for a DB instance descriptor.** The values that this measure reports, the

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>significance of each of these values, and the numeric values that correspond to them are discussed in the table below:
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td>Failed</td><td>The instance has failed and Amazon RDS was unable to recover it. Perform a point-in-time restore to the latest restorable time of the instance to recover the data.</td><td>0</td></tr>
<tr><td>Available</td><td>The instance is healthy and available</td><td>1</td></tr>
<tr><td>Backing-up</td><td>The instance is currently being backed up.</td><td>2</td></tr>
<tr><td>Creating</td><td>The instance is being created. The instance is inaccessible while it is being created.</td><td>3</td></tr>
<tr><td>Inaccessible-encryption-credentials</td><td>The KMS key used to encrypt or decrypt the DB instance could not be accessed.</td><td>4</td></tr>
</table>
</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
Incompatible-credentials	The supplied CloudHSM username or password is incorrect. Please update the CloudHSM credentials for the DB instance.	5
Incompatible-network	Amazon RDS is attempting to perform a recovery action on an instance but is unable to do so because the VPC is in a state that is preventing the action from being completed. This status can occur if, for example, all available IP addresses in a subnet were in use and Amazon RDS was unable to get an IP address for the DB instance.	6
Incompatible-option-group	Amazon RDS attempted to apply an option group change but	7

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
	was unable to do so, and Amazon RDS was unable to roll back to the previous option group state. Consult the Recent Events list for the DB instance for more information. This status can occur if, for example, the option group contains an option such as TDE and the DB instance does not contain encrypted information.	
Incompatible-parameters	Amazon RDS was unable to start up the DB instance because the parameters specified in the instance's DB parameter group were not compatible. Revert the parameter changes or make them compatible with the instance	8

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
	to regain access to your instance. Consult the Recent Events list for the DB instance for more information about the incompatible parameters.	
Incompatible-restore	Amazon RDS is unable to do a point-in-time restore. Common causes for this status include using temp tables or using MyISAM tables.	9
Maintenance	Amazon RDS is applying a maintenance update to the DB instance.	10
Modifying	The instance is being modified because of a customer request to modify the instance.	11
Rebooting	The instance is being rebooted because of a customer request or an Amazon	12

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
	RDS process that requires the rebooting of the instance.	
Renaming	The instance is being renamed because of a customer request to rename it.	13
Resetting-master-credentials	The master credentials for the instance are being reset because of a customer request to reset them.	14
Restore-error	The DB instance encountered an error attempting to restore to a point-in-time or from a snapshot.	15
Upgrading	The database engine version is being upgraded.	16
Storage-full	The instance has reached its storage capacity allocation. This is a critical status and should be remedied immediately; you	17

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td></td><td>should scale up your storage by modifying the DB instance. Set alarms to warn you when storage space is getting low so you don't run into this situation.</td><td></td></tr>
<tr><td>Deleting</td><td>The instance is being deleted.</td><td>18</td></tr>
</table>
Note:

This measure reports the Measure Values listed in the table above to indicate the current status of a DB instance. In the graph of this measure however, the same will be represented using the corresponding numeric equivalents only.

Use the detailed diagnosis of this measure to know the DB name, DB class, engine name, and storage type of a DB instance.
</td></tr>
<tr><td>CPU credit usage:</td><td>Indicates the number of CPU units consumed by this T2 DB instance/ all DB instances that belong to this T2 DB instance class / all T2 DB instances using this DB engine, during the last measurement period.</td><td>Number</td><td>These measures are reported only for individual T2 instances, instances that belong to T2 DB instance classes, and the DB engines used only by T2 instances.

A CPU Credit provides the performance of a full CPU core for one minute. Traditionalinstance types provide fixed performance, while T2 instances provide a baseline level of CPU performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits.

One CPU credit is equal to one vCPU running at</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			100% utilization for one minute. Other combinations of vCPUs, utilization, and time are also equal to one CPU credit; for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes. Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set rate of CPU credits per hour, depending on instance size. When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting. When your T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization. The more credits your T2 instance has for CPU resources, the more time it can burst beyond its base performance level when more performance is needed. This implies that ideally, the value of the CPU credit usage measure should be low for an instance and the value of the CPU credit balance for that instance should be high, as that way, an instance is assured of more CPU resources when performance demands increase. By comparing the value of this measure across instances, you can precisely identify the instance that has used up a sizeable portion of its CPU credits.
CPU credit balance:	Indicates the number of CPU credits that an instance has accumulated.	Number	
CPU utilization:	Indicates the percentage of CPU utilized by this DB instance /	Percent	A value close to 100% for this measure for any DB instance is indicative of excessive CPU usage by that instance. Track the variations to the value of this measure for such

Measurement	Description	Measurement Unit	Interpretation
	DB instance class / DB engine		an instance closely, and figure out whether CPU usage is consistently high and close to 100%. If so , you can conclude that the instance requires more CPU than what's been allocated to it. You may want to change to the DB instance class definition to allot more CPU resources to all instances it governs.
Binlog disk usage:	Indicates the amount of disk space occupied by binary logs on this DB instance / all DB instances of this DB instance class / all DB instances using this DB engine	KB	The binary log on MySQL has two important purposes: • For replication, the binary log on a master replication server provides a record of the data changes to be sent to slave servers. The master server sends the events contained in its binary log to its slaves, which execute those events to make the same data changes that were made on the master. . • Certain data recovery operations require use of the binary log. After a backup has been restored, the events in the binary log that were recorded after the backup was made are re-executed. These events bring databases up to date from the point of the backup. Typically, MySQL uses several logging formats to record information in the binary log. There are three logging formats: • Replication capabilities in MySQL originally were based on propagation of SQL statements from master to slave. This is called statement-based logging. • In row-based logging, the master writes

Measurement	Description	Measurement Unit	Interpretation
			events to the binary log that indicate how individual table rows are affected. • A third option is also available: mixed logging. With mixed logging, statement-based logging is used by default, but the logging mode switches automatically to row-based in certain cases. MySQL on Amazon RDS supports both the row-based and mixed binary logging formats for MySQL version 5.6. The default binary logging format is mixed. For DB instances running MySQL versions 5.1 and 5.5, only mixed binary logging is supported. If the value of this measure grows consistently, it could mean that large binary files are being created. At this juncture, you may want to check the logging format configured for MySQL on Amazon RDS. This is because, very often, row-based binary logging format can result in very large binary log files. If you do not change the logging mode, then such files will continue to be created, thereby reducing the amount of storage space available for a DB instance. This in turn can increase the amount of time to perform a restore operation of a DB instance.
Database connections:	Indicates the number of database connections currently used by this instance / all instances that belong to this DB instance class / all instances using this DB engine	Number	

Measurement	Description	Measurement Unit	Interpretation
Disk queue depth:	Indicates the number of outstanding IOs (read/write requests) waiting to access this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine.	Number	If the value of this measure keeps increasing steadily and significantly for a DB instance, it could indicate that the DB instance is latent, and is unable to process I/O requests quickly. The value of this measure therefore should be low at all times.
Freeable memory:	Indicates the amount of available random access memory for this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine	MB	A high value is desired for this measure to ensure peak performance of a DB instance.
Replica lag time:	Indicates the amount of time a Read Replica DB Instance lags behind this source DB Instance / all source DB instances that belong to this DB instance class / all source DB instances using DB engine	Secs	**This measure applies to MySQL read replicas only.** f your system runs on Amazon Relational Database Service (RDS) you may have opted to configure one or more replicas for your main MySQL database(s). This means you have a master RDS instance and at least one slave RDS instance which receives updates from the master. This process is called replication. Replication ensures that changes made on the master database also happen on the slave after some period of time. For a variety of reasons this period of time can increase. For example, a long- running query or erroneous query can cause replication to slow down or stop entirely. This results in replication lag: changes made on your main database aren't showing up on the slave replica because the replica is lagging

Measurement	Description	Measurement Unit	Interpretation
			behind. If the value of this measure is increasing consistently for a DB instance, it is a cause for concern, as it indicates that the slave is not in sync with the master and will take a long time to catch up. If for any reason the master DB instance fails at this juncture, there is bound to be significant data loss owing to the master-slave non-sync. When there is a replication issue the output of *show slave status*; is quite useful in debugging and resolving it. You need to review the values of: *Slave_SQL_Running* *Last_Error* *Last_SQL_Error* When a particular SQL query failed on the slave it could be that execution of queries in general has stopped. This is indicated by *Slave_SQL_Running* having the value *No*. In that case you'll either need to: • Remedy the error by fixing the issue that caused the SQL query to fail. • Decide to resume replication by letting the slave ignore that error. The former situation can be tricky as it requires you to figure out what data or query is problematic based on the values of *Last_Error* and *Last_SQL_Error*. These fields may provide enough information to determine any incorrect records but this is not always the case. In the latter case you would execute the

Measurement	Description	Measurement Unit	Interpretation
			following command on the slave: *CALL mysql.rds_skip_repl_error;* You should only run this command when you've determined that skipping the SQL query won't lead to inconsistent data or incorrect data on the slave (or, at least, that this is allowed to occur by skipping that particular SQL query).
Swap usage:	Indicates the amount of swap space used on this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine.	KB	
Read IOPS:	Indicates the rate at which disk read I/O operations were performed by this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine	Reads/Sec	Ideally, the value of this measure should be high. A consistent drop in this value could indicate a read latency.
Write IOPS:	Indicates the rate at which disk write I/O operations were performed by this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine	Writes/Sec	Ideally, the value of this measure should be high. A consistent drop in this value could indicate a write latency.
Read latency:	Indicates the average amount of	Secs	Ideally, the value of this measure should be low. A consistent rise in this value could indicate a

Measurement	Description	Measurement Unit	Interpretation
	time this DB instance / all DB instances of this instance class / all DB instances using this engine, took to service read requests.		read latency. Compare the value of this measure across DB instances to know which instance is the slowest in servicing read requests.
Write latency:	Indicates the average amount of time this DB instance / all DB instances of this instance class / all DB instances using this engine, took to service write requests.	Secs	Ideally, the value of this measure should be low. A consistent rise in this value could indicate a write latency. Compare the value of this measure across DB instances to know which instance is the slowest in servicing write requests.
Read throughput:	Indicates the rate at which data was read from the disk by this DB instance / all DB instances of this instance class / all DB instances using this DB engine.	KB/Sec	Ideally, the value of this measure should be high. A steady decrease in this value could indicate a read latency. Compare the value of this measure across DB instances to know which instance is the slowest in servicing read requests.
Write throughput:	Indicates the rate at which data was written to the disk by this DB instance / all DB instances of this instance class / all DB instances using this DB engine.	KB/Sec	Ideally, the value of this measure should be high. A steady decrease in this value could indicate a write latency. Compare the value of this measure across DB instances to know which instance is the slowest in servicing write requests.
Network receive throughput:	Indicates the incoming network	KB/Secs	The value of these measures includes both customer database traffic and Amazon RDS

Measurement	Description	Measurement Unit	Interpretation
	traffic on this DB instance / all DB instances that belong to this instance class / all DB instances using this engine.		traffic used for monitoring and replication. A high value for these measures is indicative of high bandwidth usage by a DB instance. Under such circumstances, compare the value of the Network receive throughput measure with that of th Network transmit throughput measure to determine when the maximum bandwidth was consumed - when sending data or when receiving it?
Network transmit throughput:	Indicates the outgoing network traffic on this DB instance / all DB instances that belong to this instance class / all DB instances using this engine.	KB/Secs	
Total storage space:	Indicates the total amount of storage space allocated to this DB instance / all DB instances that belong to this instance class / all DB instances using this DB engine.	MB	
Used storage space:	Indicates the amount of storage space used by this DB instance / all DB instances that belong to this instance class / all DB instances using this DB engine.	MB	Compare the value of this measre across DB instances to know which instance is consuming storage space excessively.
Free storage space:	Indicates the amount of storage space still unused by this DB instance / all DB	MB	A high value for this measure is ideal. Compare the value of this measure across DB instances to know which instance is left with very little free space.

Measurement	Description	Measurement Unit	Interpretation
	instances that belong to this instance class / all DB instances using this DB engine.		
Free storage space:	Indicates the percentage of storage space allocated to this DB instance / all DB instances that belong to this instance class / all DB instances using this DB engine, which is still available for use.	Percent	A value close to 100% is desired for this measure. If the value of this measure is below 50% consistently, it indicates that the DB instance is not sized with adequate resources. You may want to consider changing the DB instance class of that instance, so that more storage resources are available to it.
Burst balance	Indicates the percentage of General Purpose SSD (gp2) burst-bucket I/O credits available to this DB instance / all DB instances that belong to this instance class / all DB instances using this DB engine.	Percent	The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your gp2 volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed. Each gp2 volume receives an initial I/O credit balance of 5.4 million I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes. Volumes earn I/O credits at the baseline performance rate of 3 IOPS per GiB of volume size. When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are

Measurement	Description	Measurement Unit	Interpretation
			added to the I/O credit balance. When your volume requires more than the baseline performance I/O level, it draws on I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. This means that for a gp2 volume to burst performance levels above its baseline, a high I/O credit balance is necessary. This implies that the value of this measure should ideally be high for a gp2 volume. If your gp2 volume uses all of its I/O credit balance - i.e., if the value of this measure is 0 or very low for a gp2 volume - then the maximum IOPS performance of the volume remains at the baseline IOPS performance level (the rate at which your volume earns credits) and the volume's maximum throughput is reduced to the baseline IOPS multiplied by the maximum I/O size. When I/O demands rise above the baseline performance level of the volume, the volume will be unable to meet with the demand owing to the lack of adequate I/O credits.

4.6 The AWS Application Layer

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services. Using the test mapped to this layer, the overall health, responsiveness, resource usage, and errors encountered by this service can be monitored, and abnormalities promptly captured and reported.

Figure 4.13: The tests mapped to the AWS Application layer

4.6.1 AWS Elastic Beanstalk Test

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services.

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

Elastic Beanstalk supports applications developed in Java, PHP, .NET, Node.js, Python, and Ruby, as well as different container types for each language. A container defines the infrastructure and software stack to be used for a given environment. When you deploy your application, Elastic Beanstalk provisions one or more AWS resources, such as Amazon EC2 instances. The software stack that runs on your Amazon EC2 instances depends on the container type. For example, Elastic Beanstalk supports two container types for Node.js: a 32-bit Amazon Linux image and a 64-bit Amazon Linux image. Each runs a software stack tailored to hosting a Node.js application.

To use Elastic Beanstalk, you create an application, upload an application version in the form of an application source bundle (for example, a Java .war file) to Elastic Beanstalk, and then provide some information about the application. Elastic Beanstalk automatically launches an environment and creates and configures the AWS resources needed to run your code. After your environment is launched, you can then manage your environment and deploy new application versions.

The stability of an application deployed using Elastic Beanstalk relies on the overall health and performance of the underlying environment and the instances in the environment. Unhealthy and resource-starved instances and environments, and those experiencing processing errors/bottlenecks consistently, can adversely impact application performance and impair user productivity. If this is to be averted, then administrators should periodically check on the health of the environments configured for application deployment and instances that underlie applications, promptly detect abnormalities, and rapidly initiate measures to mitigate them. This is where the **AWS Elastic Beanstalk** test helps!

By default, this test automatically discovers the instances that underlie applications and measures the following for each instance:

- Overall health;
- Responsiveness to requests;

- Resource usage;
- HTTP errors;

In the process, the test leads administrators to unhealthy instances.

Optionally, you can configure the test to report metrics for each environment, instead of instance. This environment-level insight enables administrators to identify the precise environments that have been affected by the unhealthy instances.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each instance / environment

First-level descriptor: AWS Region

Second-level descriptor: Instance ID / Environment name, depending upon the option chosen from the **Beanstalk Filter Name** parameter of this test

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password,	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively.

Parameter	Description
and Confirm Password	Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Beanstalk Filter Name	By default, this parameter is set to **InstanceID**. This means that by default, this test will report metrics for each instance. If required, you can override this default setting by setting the **Beanstalk Filter Name** parameter to **EnvironmentName**. In this case, this test will report metrics for every application environment. An environment is a version that is deployed onto AWS resources. Each environment runs only a single application version at a time, however you can run the same version or different versions in many environments at the same time. When you create an environment, Elastic Beanstalk provisions the resources needed to run the application version you specified.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Completed requests	By default, this measure represents the number of requests completed by this instance. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the number of requests that were completed by this environment.	Number	
Completed requests	By default, this measure	Number	This class of status codes indicates

Measurement	Description	Measurement Unit	Interpretation
with 2XX status code	represents the number of requests to this instance that resulted in HTTP 2xx response codes. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the number of requests to this environment that resulted in HTTP 2xx response codes.		the action requested by the client was received, understood and accepted. Ideally therefore, the value of this measure should be high.
Completed requests with 3XX status code	By default, this measure represents the number of requests to this instance that resulted in HTTP 3xx response codes. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the number of requests to this environment that resulted in HTTP 3xx response codes.	Number	This class of status code indicates the client must take additional action to complete the request. Many of these status codes are used in URL redirection.
Completed requests with 4XX status code	By default, this measure represents the number of requests to this instance that resulted in HTTP 4xx response codes. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the number of requests to this environment that resulted in HTTP 4xx	Number	This class of status code is intended for situations in which the error seems to have been caused by the client. A non-zero value is hence desired for this measure.

Measurement	Description	Measurement Unit	Interpretation
	response codes.		
Completed requests with 5XX status code	By default, this measure represents the number of requests to this instance that resulted in HTTP 5xx response codes. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the number of requests to this environment that resulted in HTTP 5xx response codes.	Number	Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request. A non-zero value is hence desired for this measure.
Time to complete 10 percent of requests	By default, this measure represents the average time taken by this instance to complete 10 percent of the fastest requests. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the average time this environment took to complete 10 percent of the fastest requests.	Secs	If the value of this measure is very high, it indicates that the instance/environment has very low processing power.
Time to complete 50 percent of requests	By default, this measure represents the average time taken by this instance to complete 50 percent of the fastest requests. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the average time this	Secs	If the value of this measure is very high, it indicates that the instance/environment has very low processing power.

Measurement	Description	Measurement Unit	Interpretation
	environment took to complete 50 percent of the fastest requests.		
Time to complete 75 percent of requests	By default, this measure represents the average time taken by this instance to complete 75 percent of the fastest requests. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the average time this environment took to complete 75 percent of the fastest requests.	Secs	If the value of this measure is very high, it indicates that the instance/environment has very low processing power.
Time to complete 85 percent of requests	By default, this measure represents the average time taken by this instance to complete 85 percent of the fastest requests. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the average time this environment took to complete 85 percent of the fastest requests.	Secs	If the value of this measure is very high, it indicates that the instance/environment has very low processing power.
Time to complete 90 percent of requests	By default, this measure represents the average time taken by this instance to complete 90 percent of the fastest requests. If the Beanstalk Filter Name is set to **EnvironmentName**, then	Secs	If the value of this measure is very high, it indicates that the instance/environment has very low processing power.

Measurement	Description	Measurement Unit	Interpretation
	this measure represents the average time this environment took to complete 90 percent of the fastest requests.		
Time to complete 95 percent of requests	By default, this measure represents the average time taken by this instance to complete 95 percent of the fastest requests. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the average time this environment took to complete 95 percent of the fastest requests.	Secs	If the value of this measure is very high, it indicates that the instance/environment has very low processing power.
Time to complete 99 percent of requests	By default, this measure represents the average time taken by this instance to complete 99 percent of the fastest requests. If the Beanstalk Filter Name is set to **EnvironmentName**, then this measure represents the average time this environment took to complete 99 percent of the fastest requests.	Secs	If the value of this measure is very high, it indicates that the instance/environment has very low processing power.
Time to complete X percent of requests	By default, this measure represents the average time taken by this instance to complete X percent of the fastest requests.	Secs	If the value of this measure is very high, it indicates that the instance/environment has very low processing power.

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td>If the Beanstalk Filter Name is set to EnvironmentName, then this measure represents the average time this environment took to complete X percent of the fastest requests.</td><td></td><td></td></tr>
<tr><td>Environment status</td><td>Indicates the current health status of this environment</td><td></td><td>This measure is reported only for an Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to EnvironmentName.

The values that this measure can report and their corresponding numeric values are detailed in the table below:

<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>OK</td><td>0</td></tr>
<tr><td>Info</td><td>1</td></tr>
<tr><td>Unknown</td><td>5</td></tr>
<tr><td>No data</td><td>10</td></tr>
<tr><td>Warning</td><td>15</td></tr>
<tr><td>Degraded</td><td>20</td></tr>
<tr><td>Severe</td><td>25</td></tr>
</table>

The table below describes what each of the Measure Values denote for an environment:

<table>
<tr><th>Measure Value</th><th>Description</th></tr>
<tr><td>OK</td><td>Most instances in the environment</td></tr>
</table>
</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description
	are passing health checks and the health agent is not reporting major issues.
Info	An operation is in progress on several instances in an environment.
Unknown	Elastic Beanstalk and the health agent are reporting an insufficient amount of data on an instance.
No data	One/more instances in the environment are not reporting any health status data.
Warning	The health agent is reporting a moderate number of request failures or other issues for an environment. Example: One instance in the environment has a status of Severe.
Degraded	The health agent is reporting a high number of request

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Description</th></tr>
<tr><td></td><td>failures or other issues for an environment. Example: Environment is in the process of scaling up to 5 instances.

Message (Environment): 4 active instances is below Auto Scaling group minimum size 5</td></tr>
<tr><td>Severe</td><td>The health agent is reporting a very high number of request failures or other issues for an environment.

Example: Elastic Beanstalk is unable to contact the load balancer to get instance health.</td></tr>
</table>
Note:

By default, this measure reports the Measure Values listed in the table above to indicate the status of an environment. In the graph of this measure however, the same is indicated using the numeric equivalents only.
</td></tr>
<tr><td>Instances in ok</td><td>Indicates the number of</td><td>Number</td><td>This measure is reported only for an</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
state	instances in this environment with OK health status.		Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to **EnvironmentName**. An instance is said to be in the OK state if the instance is passing health checks and is completing requests normally.
Instances in pending state	Indicates the number of instances in this environment with Pending health status.	Number	This measure is reported only for an Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to **EnvironmentName**. An instance is said to be in the Pending state if an operation is in progress on thatn instance within the command timeout.
Instances in info state	Indicates the number of instances in this environment with Info health status.	Number	This measure is reported only for an Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to **EnvironmentName**. An instance is said to be in the Info state if an operation is in progress on that instance.
Instances in unknown state	Indicates the number of instances in this environment with Unknown health status.	Number	This measure is reported only for an Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to **EnvironmentName**. An instance is said to be in the Unknown state if Elastic Beanstalk and the health agent are reporting an insufficient amount of data on an instance.
Instances in nodata state	Indicates the number of instances in this environment with Warning health status.	Number	This measure is reported only for an Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to **EnvironmentName**.

Measurement	Description	Measurement Unit	Interpretation
			An instance is said to be in the Nodata state if no health status data has been collected from that instance.
Instances in Warning state	Indicates the number of instances in this environment with Info health status.	Number	This measure is reported only for an Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to **EnvironmentName**. An instance is said to be in the Warning state if an operation in progress on that instance and is taking a very long time.
Instances in degraded state	Indicates the number of instances in this environment with Degraded health status.	Number	This measure is reported only for an Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to **EnvironmentName**. An instance is said to be in the Degraded state if the health agent is reporting a high number of request failures or other issues for that instance. Ideally, the value of this measure should be 0.
Instances in severe state	Indicates the number of instances in this environment with Severe health status.	Number	This measure is reported only for an Environment - i.e., only if the 'Beanstalk Filter Name' parameter is set to **EnvironmentName**. An instance is said to be in the Severe state if the health agent is reporting a very high number of request failures or other issues for that instance. Ideally, the value of this measure should be 0.
CPU load over last minute	Indicates the average CPU load of this instance over the last minute.	Number	This measure is reported only for an Instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to **InstanceID**.

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>A high value is indicative of excessive CPU usage by an instance. Compare the value of this measure across instances to know which instance is consuming the maximum CPU. To know where that instance is spending its CPU, take a look at the values reported for the other CPU measures of this test.</td></tr>
<tr><td>Instance status</td><td>Indicates the current status of this instance.</td><td></td><td>This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to InstanceID.

The values that this measure can report and their corresponding numeric values are detailed in the table below:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>OK</td><td>0</td></tr><tr><td>Info</td><td>1</td></tr><tr><td>Unknown</td><td>5</td></tr><tr><td>No data</td><td>10</td></tr><tr><td>Warning</td><td>15</td></tr><tr><td>Degraded</td><td>20</td></tr><tr><td>Severe</td><td>25</td></tr></table>
The table below describes what each of the Measure Values denote for an instance:

<table><tr><th>Measure Value</th><th>Description</th></tr><tr><td>OK</td><td>An instance is</td></tr></table></td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description
	passing health checks and is completing requests normally.
Info	An operation is in progress on an instance.
Unknown	Elastic Beanstalk and the health agent are reporting an insufficient amount of data on an instance.
No data	An instance is not reporting any health status data.
Warning	An operation in progress on an instance and is taking a very long time.
Degraded	The health agent is reporting a high number of request failures or other issues for an instance
Severe	The health agent is reporting a very high number of request failures or other issues for an instance.

Measurement	Description	Measurement Unit	Interpretation
			Note: By default, this measure reports the **Measure Value**s listed in the table above to indicate the status of an instance. In the graph of this measure however, the same is indicated using the numeric equivalents only.
Disk space utilization	Indicates the percentage of disk space utilized by this instance.	Percent	This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to **InstanceID**. A value close to 100% is a cause for concern, as it implies that the instance is running out of disk space.
Interrupt request CPU	Indicates the percentage of time CPU of this instance was in interrupt request state.	Percent	This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to **InstanceID**. In a computer, an interrupt request (or IRQ) is a hardware signal sent to the processor that temporarily stops a running program and allows a special program, an interrupt handler, to run instead. If any instance is using CPU excessively, then compare the value of this measure with that of the User CPU, System CPU, Idle CPU, Waitio CPU, Nice CPU, and Soft interrupt request CPU measures for that instance to know where CPU time has been spent the maximum - in running user processes? system processes? being idle? waiting for I/O? waiting for an interrupt handler? handling a software interrupt? or running nice processes?

Measurement	Description	Measurement Unit	Interpretation
User CPU	Indicates the percentage of CPU time that this instance spent running user programs.	Percent	This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to **InstanceID**. If any instance is using CPU excessively, then compare the value of this measure with that of the Interrupt request CPU, System CPU, Idle CPU, Waitio CPU, Nice CPU, and Soft interrupt request CPU measures for that instance to know where CPU time has been spent the maximum - in running user processes? system processes? being idle? waiting for I/O? waiting for an interrupt handler? handling a software interrupt? or running nice processes?
System CPU	Indicates the percentage of CPU time that this instance spent on system-level processing.	Percent	This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to **InstanceID**. If any instance is using CPU excessively, then compare the value of this measure with that of the Interrupt request CPU, User CPU, Idle CPU, Waitio CPU, Nice CPU, and Soft interrupt request CPU measures for that instance to know where CPU time has been spent the maximum - in running user processes? system processes? being idle? waiting for I/O? waiting for an interrupt handler? handling a software interrupt? or running nice processes?
Idle CPU	Indicates the percentage of CPU time that this instance spent without processing any requests -	Percent	This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to **InstanceID**.

Measurement	Description	Measurement Unit	Interpretation
	i.e., just being idle.		If any instance is using CPU excessively, then compare the value of this measure with that of the Interrupt request CPU, User CPU, System CPU, Waitio CPU, Nice CPU, and Soft interrupt request CPU measures for that instance to know where CPU time has been spent the maximum - in running user processes? system processes? being idle? waiting for I/O? waiting for an interrupt handler? handling a software interrupt? or running nice processes?
Soft interrupt request CPU	Indicates the percentage of CPU time that this instance spent in the soft interrupt request state.	Percent	This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to **InstanceID**. A software interrupt or soft interrupt is caused either by an exceptional condition in the processor itself, or a special instruction in the instruction set which causes an interrupt when it is executed If any instance is using CPU excessively, then compare the value of this measure with that of the Interrupt request CPU, User CPU, System CPU, Idle CPU, Waitio CPU, and Nice CPU measures for that instance to know where CPU time has been spent the maximum - in running user processes? system processes? being idle? waiting for I/O? waiting for an interrupt handler? handling a software interrupt? or running nice processes?
Waitio CPU	Indicates the percentage of CPU time that this instance spent waiting for	Percent	This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to

Measurement	Description	Measurement Unit	Interpretation
	I/O.		**InstanceID**. If any instance is using CPU excessively, then compare the value of this measure with that of the Interrupt request CPU, User CPU, System CPU, Idle CPU, Nice CPU, and Soft interrupt request measures for that instance to know where CPU time has been spent the maximum - in running user processes? system processes? being idle? waiting for I/O? waiting for an interrupt handler? handling a software interrupt? or running nice processes?
Nice CPU	Indicates the percentage of CPU time that this instance spent in running nice processes.	Percent	This measure is reported only for an instance - i.e., only if the 'Beanstalk Filter Name' parameter is set to **InstanceID**. nice is program that is used to invoke a utility or shell script with a particular priority, thus giving the process more or less CPU time than other processes. If any instance is using CPU excessively, then compare the value of this measure with that of the Interrupt request CPU, User CPU, System CPU, Idle CPU, Waitio CPU, and Software interrupt request measures for that instance to know where CPU time has been spent the maximum - in running user processes? system processes? being idle? waiting for I/O? waiting for an interrupt handler? handling a software interrupt? or running nice processes?

4.7 The AWS Services Layer

The tests mapped to this layer measures the efficiency some of the critical services offered by the AWS cloud. These include the following:

- The AWS EC2 Elastic Block Store (EBS) service
- The AWS EC2 Container service (ECS)
- The AWS Relational Database service (RDS)
- The AWS Simple Email service (SES)
- The AWS Billing and Cost Management service.

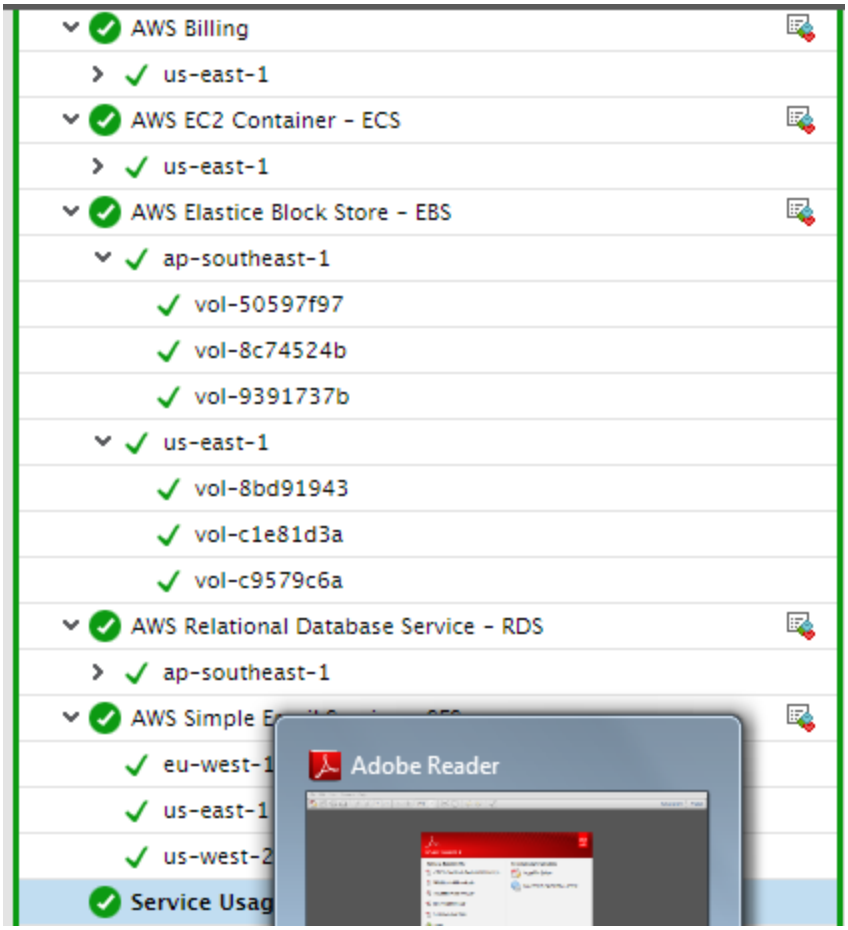


Figure 4.14: The tests mapped to the AWS Services layer

4.7.1 AWS Billing Test

AWS Billing and Cost Management is the service that you use to pay your AWS bill, monitor your usage, and budget your costs.

When budgeting costs, this service also provides forecasts of your estimated costs. Using the AWS Billing test you can configure thresholds for this estimate for each service you subscribe to and also for a roll-up of estimated charges of all services. The test will then proactively alert you if the estimate is about to exceed your budget, and thus enable you to initiate measures for avoiding cost overruns.

Note:

The metrics of this test can be viewed for the 'us-east' region only. However, since this region stores Amazon CloudWatch metrics for worldwide estimated charges, the *Estimated charges* that this region reports for a service will be the consolidated charges for all regions that are using that particular service.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each service subscribed in each AWS region

First-level descriptor: AWS Region

Second-level descriptor: ServiceName

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy	In some environments, all communication with the AWS EC2 cloud and its

Parameter	Description
Port	regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability

Parameter	Description
	• Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Estimated charges:	Indicates the estimated cost of this service across all regions that are using that service.	USD	Compare the value of this measure across services to know which service you will be spending the most on in the future. You can be notified if cost estimations for a service exceed an acceptable limit, by configuring such a limit as a the maximum threshold for this measure for that service. Based on these alarms, you can set out to change how frequently you actually use that service, so as to reduce related overheads. For the Total descriptor, this measure will report the total estimated charges across all services.

4.7.2 AWS Certificate Manager Test

AWS Certificate Manager (ACM) handles the complexity of creating and managing SSL/TLS certificates for your AWS based websites and applications. You can use certificates provided by ACM (ACM Certificates) or certificates that you import into ACM. ACM Certificates can secure multiple domain names and multiple names within a domain. You can also use ACM to create wildcard SSL certificates that can protect an unlimited number of subdomains.

If you are unable to access any web site/web application on the AWS cloud, you may want to check if the certificate attached to that web site/web application has expired, failed, or revoked. This check is made possible by the AWS Certificate Manager test!

This test automatically discovers the certificates managed by the AWS Certificate Manager and reports the current status of each certificate. This way, expired, revoked, failed, and inactive

certificates can be identified. Besides expired certificates, the test also leads you to certificates nearing expiry by reporting the number of days each certificate will remain valid. You can also use the detailed diagnostics of this test to know who issued such a certificate, when it was issued, the resources used by that certificate, and the domains included in it.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for certificate managed by the ACM.

First-level descriptor: AWS Region

Second-level descriptor: Certificate ID

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.

Parameter	Description
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Status	Indicates the current status of this certificate.		The values that this measure can report and their corresponding numeric values are listed in the table below:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Failed</td><td>0</td></tr>
<tr><td>Expired</td><td>1</td></tr>
<tr><td>Validation timed out</td><td>2</td></tr>
<tr><td>Inactive</td><td>3</td></tr>
<tr><td>Pending val-idation</td><td>4</td></tr>
<tr><td>Revoked</td><td>5</td></tr>
<tr><td>Issued</td><td>6</td></tr>
</table>
If the status of a certificate is abnormal, then you can use the detailed diagnosis of this measure to know who issued an expired, when, which resources are managed by that certificate, and which domains are included in it. You can also use the detailed diagnosis to track the expiry of an issued certificate.
Note:
Typically, this measure will report the Measure Values listed in the table above to indicate the status of a certificate. In the graph of this measure however, the same will be indicated using the numeric equivalents only.
</td></tr>
<tr><td>Certificate validity</td><td>Indicates the number of days by which this certificate will expire.</td><td>Days</td><td>A very low value for this measure indicates that the certificate is set to expire shortly. If If this measure reports the value 0, it implies that the certificate has already expired. You can then use the detailed diagnosis of the Status measure of this test to know who issued that certificate,</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			when, and what is the certificate type (whether imported or not) If the certificate that is about to expire is an imported certificate, then ACM will not manage the renewal process of that certificate. In this case, you will have to import a new third-party certificate to replace the expiring one. On the other hand, if the certificate that is about to expire was provided by ACM, then ACM will try to automatically renew that certificate before expiry.

The detailed diagnosis of the *Status* measure reports the certification authority who issued the certificate, the issue date, the certificate type (whether imported or not), the expiry date, the domains included in the certificate, and the resources managed by it. With the help of this information, you can quickly figure out if a certificate is about to expire soon. If this is an imported certificate, then ACM will not manage the renewal process of that certificate. In this case, you will have to import a new third-party certificate to replace the expiring one. On the other hand, if the certificate that is about to expire was provided by ACM, then ACM will try to automatically renew that certificate before expiry.

Details of Certificate

ISSUER	TYPE	CREATED DATE	EXPIRE DATE	DOMAIN INCLUDED IN CERTIFICATE	CERTIFICATE USED BY AWS RESOURCES
Jan 05, 2018 07:11:34					
DigiCert Inc	IMPORTED	-	Wed Oct 31 17:30:00 IST 2018	*.eginnovations.com,eginnovations.c...	arn:aws:cloudfront::129794746678:distribution/E221UUEPNYZH...

Figure 4.15: The detailed diagnosis of the Status measure

4.7.3 AWS Auto Scaling Test

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Auto Scaling ensures that your group has this many instances. If you

specify scaling policies, then Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

To measure the effectiveness of Auto Scaling, administrators should continuously track the size and state of instances in each Auto Scaling group and determine whether/not the groups automatically expand and shrink in size when the configured thresholds are breached. The **AWS Auto Scaling** test makes this evaluation possible!

For each Auto Scaling group, this test reports the minimum and maximum size threshold configured for that group and also the desired capacity specified for the group. Additionally, the test reveals the current number of instances in that group. By comparing the capacity thresholds with the current number of instances, administrators can tell whether/not Auto Scaling is successful in automatically resizing the group based on the demand/configuration. As an Auto Scaling group shrinks and expands automatically, the instances in the group pass through various states - pending state, standby state, terminating state, etc. With the help of this test, administrators can track the state of instances in each group.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each Auto Scaling group

First-level descriptor: AWS Region

Second-level descriptor: Auto Scaling group name

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy	In some environments, all communication with the AWS EC2 cloud and its

Parameter	Description
Port	regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Minimum instances in group	Indicates the minimum size of this group.	Number	
Maximum instances in group	Indicates the maximum size of this group.	Number	
Desired capacity of group	Indicates the number of instances that this group attempts to maintain.	Number	
Running instances in group	Indicates the number of instances in currently running in this group.	Number	This measure represents the number of instances that are currently in the 'InService' state.

Measurement	Description	Measurement Unit	Interpretation
			When Auto Scaling responds to a scale out event, it launches one or more instances. When the instances are fully configured, they are attached to the Auto Scaling group and they enter the InService state Instances remain in the InService state until one of the following occurs: • A scale in event occurs, and Auto Scaling chooses to terminate this instance in order to reduce the size of the Auto Scaling group. • You put the instance into a Standby state. • You detach the instance from the Auto Scaling group. • The instance fails a required number of health checks, so it is removed from the Auto Scaling group, terminated, and replaced. This metric does not include instances that are pending or terminating.
Pending state instances in group	Indicates the number of instances in this group that are in the Pending state.	Number	When a scale out event occurs, the Auto Scaling group launches the required number of EC2 instances, using its assigned launch configuration. These instances start in the Pending state. pending instance is not yet in service. This metric does not include instances that are in service or terminating.
Standby state	Indicates the number of	Number	Instances in the Standby state are still

Measurement	Description	Measurement Unit	Interpretation
instances in group	instances in this group that are in the Standby state.		running but are not actively in service.
Terminating state instances	Indicates the number of instances in this group that are in the Terminating state.	Number	When Auto Scaling responds to a scale in event, it terminates one or more instances. These instances are detached from the Auto Scaling group and enter the Terminating state. This metric does not include instances that are in service or pending.
Total instances in group	Indicates the total number of instances in this group.	Number	This metric identifies the number of instances that are in running, pending, and terminating states.

4.7.4 AWS CloudSearch Test

Amazon CloudSearch is a fully managed service in the cloud that makes it easy to set up, manage, and scale a search solution for your website or application. With Amazon CloudSearch you can search large collections of data such as web pages, document files, forum posts, or product information.

To start searching your data with Amazon CloudSearch, you simply take the following steps:

- Create and configure a search domain
- Configure indexing options for your data
- Upload your data for indexing
- Send search requests to your domain

You create an Amazon CloudSearch search domain for each collection of data that you want to make searchable. A search domain encapsulates your data and the hardware and software resources required to operate a search engine. Each search domain has one or more search instances. A search instance is a server instance that has a finite amount of RAM and CPU resources for indexing data and processing requests. The number of search instances in a domain depends on the documents in your collection and the volume and complexity of your search requests.

As the amounted of data added and the volume of traffic to a domain increases, CloudSearch automatically scales your search domain to maximize search performance. Scaling is performed by automatically adding more search instances in the domain, and by partitioning the index across these instances. If you need more capacity than the additional search instances can offer, you can explicitly increase the number of search instances or instance replicas. To be able to decide whether/not additional capacity is required, you first need to determine the extent of usage of the current capacity. For this, use the **AWS CloudSearch** Test!

This test automatically discovers the search domains that have been configured in a region. For each domain, the test tracks the addition of searchable documents to that domain, and reports how much index capacity these documents consume and how many index partitions have already been created to support this load. From this, administrators can quickly infer whether/not the domain is about to exhaust its current capacity. If so, then the administrators can instantly figure out if the current number of partitions can support the anticipated load on the domain. In the process, administrators can easily compute how many more partitions would be required for maximizing the throughput and minimizing latency of search queries.

Optionally, you can configure this test to report metrics across all domains configured for the AWS account that the test uses.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each domain name / client ID

First-level descriptor: AWS Region

Second-level descriptor: ClientID / DomainName depending upon the option chosen from the **CloudSearch Filter Name** parameter of this test

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key,	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this

Parameter	Description
Confirm AWS Secret Key	has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
CloudSearch Filter Name	By default, this parameter is set to **DomainName**. This means that by default, this test will report metrics for each search domain that is configured. If required, you can override this default setting by setting the CloudSearch Filter Name to **ClientID**. In this case, the test will report metrics for the AWS account that is configured for this test. The measures reported for the ClientID will be aggregated across all search domains configured for that **ClientID**.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Successful search requests	By default, this measure represents the number of search queries/requests	Number	A high value is desired for this measure. Steady drops in the value of this measure is a cause for concern, as it

Measurement	Description	Measurement Unit	Interpretation
	that were successfully processed by this search domain. If the CloudSearch Filter Name is set to **ClientID**, then this measure will report the number of search requests that were successfully processed by all search domains configured for this AWS account.		implies poor search performance. You may want to investigate the reasons for the same.
Searchable documents in domain's search index	By default, this measure represents the number of searchable documents in this domain's search index. If the CloudSearch Filter Name is set to **ClientID**, then this measure will report the number of searchable documents across all search domains configured for this AWS account.	Number	The maximum number of documents a search domain can hold depends upon the following: • **Document size** • **Indexing options:** To index and search movie documents like this one, we configure our search domain with an index field for each document field. We can specify multiple indexing options for each field, such as the type of the field and whether the field is searchable, facet enabled, return enabled, sort enabled, and highlight enabled. These indexing options directly impact how many documents fit onto a search instance. • **Search instance type:** By default, CloudSearch makes the following instance types available:

Measurement	Description	Measurement Unit	Interpretation
			◦ search.m1.small (Small Search Instance) ◦ search.m3.medium (Medium Search Instance) ◦ search.m3.large (Large Search Instance) ◦ search.m3.xlarge (Extra Large Search Instance) ◦ search.m3.2xlarge (Double Extra Large Search Instance).
Search instance's index usage	By default, this measure represents the percentage of this domain's index capacity that has been used. If the CloudSearch Filter Name is set to **ClientID**, then this measure represents the percentage of index capacity used across all search domains configured for this AWS account.	Percent	A value close to 100% indicates that the search domain is about to exhaust its index capacity of its current search instance type. Typically, when the amount of data you add to your domain exceeds the capacity of the initial search instance type, Amazon CloudSearch scales your search domain to a larger search instance type. After a domain exceeds the capacity of the largest search instance type, Amazon CloudSearch partitions the search index across multiple search instances. To know whether the domain has exceeded the capacity of its largest instance type, check the value of the Index partitions measure for that domain. If this measure reports a non-zero value, you can conclude that the largest instance type's capacity has been exceeded.
Index partitions	By default, this measure	Number	If this measure reports a non-zero

Measurement	Description	Measurement Unit	Interpretation
	represents the number of partitions across which the search index of this search domain is distributed. If the CloudSearch Filter Name is set to **ClientID**, then this measure represents the number of partitions across which all the search domains configured for this AWS account have distributed their search index.		value, it indicates that the search domain has exceeded the capacity of its largest instance type. If you anticipate the load on the search domain to increase further, you may have to explicitly increase the number of instances that your index is partitioned across. The maximum number of search instances that can be deployed for a domain is 50 and the maximum number of partitions is 10. To increase these limits, you will have to submit an explicit request to Amazon.

4.7.5 AWS CloudTrail Events Test

AWS CloudTrail is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account. Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail. Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

Administrators can also configure event filters for events that are collected within a CloudTrail trail. This helps AWS customers save time and money by creating trails that contain a subset of overall API operations and account activity. To know which event filters have been created within which CloudTrail, use the AWS CloudTrail Events test!

This test automatically discovers the CloudTrail trails and event filters within each trail. For every event filter, the test reports the total count of events and count of error events captured by that filter. In the process, the test promptly alerts administrators when an error event is captured. Detailed diagnostics provided by this test reveals the complete details of events, thus enabling quick and easy event analysis and troubleshooting.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each event type in every event filter

First-level descriptor: AWS Region

Second-level descriptor: CloudTrail trail

Third-level descriptor: Event filter

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, and Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **CONFIRM** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Show All Events	By default, this flag is set to **Yes**. This implies that by default, this test will also report

Parameter	Description
	metrics for an additional *All* descriptor. Typically, the measures reported by the *All* descriptor will be the aggregate of the measures reported by all the other descriptors of this test - i.e., every measure reported by the *All* descriptor will return the sum of the values that all configured events have registered for that measure. This enables administrators to easily assess the overall performance of events configured for monitoring on a Cloud Trail.
Show All Only	Set this flag to **Yes**, if you wish to view only the consolidated metrics of all the events of this test. In this case therefore, only the *All* descriptor will be listed for this test. By default, this flag is set to **No**.
Show Information DD	In large AWS infrastructures, tens of thousands of CloudTrail events will be generated, even during normal operations. Naturally, the detailed diagnosis of such events will also occupy a considerable amount of database space; with time, this space consumption will grow. To minimize the strain on the eG database, by default, the detailed diagnosis capability is turned off for the *Total events* measure alone. Accordingly, the Show Information DD flag is set to **No** by default. If you want to view detailed metrics for the *Total events* measure, then set this flag to **Yes**.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total events	Indicates the total number of events currently captured by this event filter. For the *All* descriptor, this	Number	If the Show Information DD flag is set to Yes, then you can use the detailed diagnosis of this measure to know the complete details of the

Measurement	Description	Measurement Unit	Interpretation
	measure will report the total number of events captured by this CloudTrail across all its event filters.		events captured by a particular event filter. report Event Name, Event Type, Event Source, Event Time, Source IP Address, Account ID, Region Name, User Name, User Type, User Access key and User Agent.
Error events	Indicates the number of error events currently captured by this event filter. For the *All* descriptor, this measure will report the total number of error events captured by this CloudTrail across all its event filters.	Number	Ideally, the value of this measure should be 0. A non-zero value indicates that an error event has occurred. In such a situation, you can use the detailed diagnosis of this measure to know the details of the error.

The detailed diagnosis of the Total events measure reveals the name, type, and source of the events, the time at which the event occurred, the name of the user, the user access key, and user agent.

Details of CloudTrail Events

EVENT NAME	EVENT TYPE	EVENT SOURCE	EVENT TIME	SOURCE IP ADDRESS	ACCOUNT ID	REGION NAME	USER NAME	USER
Jan 18, 2018 01:40:34								
DescribeAvailabilityZones	AwsApiCall	ec2.amazonaws.com	2017-08-22T11:57:39Z	61.16.173.238	129794746678	ap-northeast-1	-	Roo
DescribeAvailabilityZones	AwsApiCall	ec2.amazonaws.com	2017-08-22T12:01:03Z	61.16.173.238	129794746678	ap-northeast-1	-	Roo
DescribeAvailabilityZones	AwsApiCall	ec2.amazonaws.com	2017-08-22T12:01:21Z	61.16.173.238	129794746678	ap-northeast-1	-	Roo
DescribeFlowLogs	AwsApiCall	ec2.amazonaws.com	2017-08-22T11:59:37Z	ec2-frontend-api.amazonaws.com	129794746678	ap-northeast-1	-	Roo
DescribeFlowLogs	AwsApiCall	ec2.amazonaws.com	2017-08-22T12:10:04Z	ec2-frontend-api.amazonaws.com	129794746678	ap-northeast-1	-	Roo
DescribeInstances	AwsApiCall	ec2.amazonaws.com	2017-08-22T11:58:03Z	52.20.96.17	129794746678	ap-northeast-1	-	Ass
DescribeInstances	AwsApiCall	ec2.amazonaws.com	2017-08-22T11:59:12Z	52.20.96.17	129794746678	ap-northeast-1	-	Ass
DescribeInstances	AwsApiCall	ec2.amazonaws.com	2017-08-22T11:59:07Z	34.204.102.208	129794746678	ap-northeast-1	-	Ass
DescribeInstances	AwsApiCall	ec2.amazonaws.com	2017-08-22T12:00:09Z	54.211.158.86	129794746678	ap-northeast-1	-	Ass
DescribeInstances	AwsApiCall	ec2.amazonaws.com	2017-08-22T12:02:10Z	54.159.133.249	129794746678	ap-northeast-1	-	Ass
DescribeVolumes	AwsApiCall	ec2.amazonaws.com	2017-08-22T12:02:23Z	182.73.75.86	129794746678	ap-northeast-1	-	Roo
DescribeVolumes	AwsApiCall	ec2.amazonaws.com	2017-08-22T11:58:05Z	61.16.173.238	129794746678	ap-northeast-1	-	Roo
DescribeVolumes	AwsApiCall	ec2.amazonaws.com	2017-08-22T12:04:11Z	61.16.173.238	129794746678	ap-northeast-1	-	Roo
DescribeVolumes	AwsApiCall	ec2.amazonaws.com	2017-08-22T12:07:34Z	61.16.173.238	129794746678	ap-northeast-1	-	Roo

Page 1 of 1

Figure 4.16: Detailed diagnosis of the Total events measure

4.7.6 AWS CloudWatch Logs Test

AWS CloudWatch Logs is a highly available, scalable, durable, and secure service to manage your operating system and application log files. It allows you to ingest, store, filter, search, and archive the logs, reducing your operational burden and allowing you to focus on your application and your business.

Administrators typically reach for logs whenever their applications and systems on the cloud encounter issues. This is because, logs greatly help problem diagnosis and troubleshooting. In fact, if administrators need to perform any custom processing on the logs for deep-dive diagnostics or if they want to load these logs on to other systems for deeper analytics, they even configure these logs to be automatically delivered to other services such as Amazon Kinesis stream, Amazon Kinesis Data Firehose stream, or AWS Lambda. In such situations, if delivery errors occur, logs may not be able to reach the destination services in time. This in turn can impede analysis and delay the administrator's troubleshooting efforts. As a result, the downtime of critical applications on the cloud will increase! To avoid this, it is good practice to frequently run the AWS CloudWatch Logs test!

This test automatically discovers the log groups that have been configured on AWS. A log group can have one or multiple log streams in it. Each of these streams will share the same retention policy, monitoring setting or access control permissions. For each log group, the test tracks the log events and log data that is forwarded by each group to AWS service destinations such as Amazon Kinesis stream, Amazon Kinesis Data Firehose stream, or AWS Lambda. In the process, the test promptly captures and reports delivery errors and also brings to light instances where delivery has been throttled. This way, the test reveals bottlenecks in the delivery of logs to AWS services, pinpoints the log groups experiencing the bottlenecks, and thus hastens appropriate corrective action.

Optionally, you can configure this test to report metrics for each log destination or for every subscription filter. This enables administrators to quickly and easily understand if specific destinations / filters are problem-prone.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each log group / destination / subscription filter

First-level descriptor: AWS Region

Second-level descriptor: Log group / destination / subscription filter, depending upon the option chosen from the **Logs Filter Name** parameter of this test

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Logs Filter Name	By default, this parameter is set to **LogGroupName**. This means that by default, this test will report metrics for log group. Log groups define groups of log streams that share the same retention, monitoring, and access control settings. Each log stream has to belong to one log group. A log stream is a sequence of log events that share the same source. For example, a log stream may be associated with an Apache access log on a specific host. If required, you can override this default setting by setting the **Logs Filter Name**

Parameter	Description
	parameter to one of the following: • **FilterName** - If you want this test to report metrics for every subscription filter, select the **FilterName** option from the **Logs Filter Name** drop-down list. Subscriptions are used to get access to a real-time feed of log events from CloudWatch Logs and have it delivered to other services such as an Amazon Kinesis stream, Amazon Kinesis Data Firehose stream, or AWS Lambda for custom processing, analysis, or loading to other systems. To begin subscribing to events, you need to create subscription filters. A subscription filter defines the filter pattern to use for filtering which log events get delivered to your AWS resource, as well as information about where to send matching log events to. • **DestinationType** - If you want this test to report metrics for every destination for your log events (eg., Amazon Kinesis stream, Amazon Kinesis Data Firehose stream, or AWS Lambda), select the **DestinationType** option from the **Logs Filter Name** drop-down list.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Incoming data	Indicates the volume of log events in uncompressed data uploaded to this log group.	KB	This measure is reported only if the 'Logs Filter Name' flag is set to 'LogGroupName'.
Incoming log events	Indicates the number of log events uploaded to this log group.	Number	This measure is reported only if the 'Logs Filter Name' flag is set to 'LogGroupName'.
Forwarded data	By default, this measure represents the volume of log events in uncompressed data that is forwarded from this log group to one/more AWS resource destinations. If the Logs Filter Name is set to **FilterName**, then this measure represents	KB	

Measurement	Description	Measurement Unit	Interpretation
	the amount of log data that is forwarded via this subscription filter to the resource destinations defined within that filter. If the Logs Filter Name is set to **DestinationType**, then this measure represents the amount of log data that is forwarded to this AWS resource destination via one/more subscription filters.		
Forwarded log events	By default, this measure represents the number of log events forwarded from this log group to one/more AWS resource destinations. If the Logs Filter Name is set to **FilterName**, then this measure represents the number of log events forwarded via this subscription filter to the resource destinations defined within that filter. If the Logs Filter Name is set to **DestinationType**, then this measure represents the number of log events forwarded to this AWS resource destination via one/more subscription filters.	Number	
Delivery errors	By default, this measure represents the number of	Number	Ideally, the value of this measure should be 0.

Measurement	Description	Measurement Unit	Interpretation
	log events in this log group that encountered errors when they were being forwarded to one/more AWS resource destinations. If the Logs Filter Name is set to **FilterName**, then this measure represents the number of log events that encountered errors when being forwarded via this subscription filter to the resource destinations defined within that filter. If the Logs Filter Name is set to **DestinationType**, then this measure represents the number of log events that encountered errors when they were being forwarded to this AWS resource destination via one/more subscription filters.		If the count of errors consistently increase for log events in a specific log group, or to a specific destination, or via a specific subscription filter, then that log group / filter / destination should be taken up for closer scrutiny.
Delivery throttling	By default, the number of log events in this log group that were throttled when being forwarded to one/more resource destinations. If the Logs Filter Name is set to **FilterName**, then this measure represents the number of log events that were throttled when being forwarded via this	Number	Ideally, the value of this measure should be low.

Measurement	Description	Measurement Unit	Interpretation
	subscription filter to the resource destinations defined within that filter. If the Logs Filter Name is set to **DestinationType**, then this measure represents the number of log events that were throttled when they were being forwarded to this AWS resource destination via one/more subscription filters.		

4.7.7 AWS EC2 Spot Fleet Test

AWS Spot instances are spare EC2 instances that you can bid on. These spot instances run whenever capacity is available and the maximum price per hour for your request exceeds the Spot price. This means that if EC2's capacity declines or the maximum price drops, spot instances may be terminated. To automatically replenish these instances and maintain your target capacity, you can use EC2 Spot fleets.

A Spot Fleet is a collection, or fleet, of Spot Instances that is launched based on criteria that you specify when placing a Spot Fleet Request. A typical Spot Fleet Request specifies the number of instances required (target capacity) or the performance characteristics (eg., vCPUs, memory, storage, etc.) you require for your application workload, the instance types, the availability zones, and the maximum price you are willing to pay for the instances. The Spot Fleet then attempts to launch the number of Spot Instances that are required to meet the target capacity that you specified in the Spot Fleet request. Spot Instances can also be terminated if any interruptions occur due to an increase in Spot price over maximum price, an increase in demand for Spot instances, or a decrease in supply of Spot instances.

By keeping tabs on each of the Spot Fleets, administrators can quickly identify those Spot Fleets that are unable to maintain the target capacity requested. Such Spot Fleet requests could be candidates for modification. Administrators should also keep an eye out for those Spot Fleets in which too many spot instances are being terminated owing to interruptions; this can reveal if the interruptions could be the reason for the wide gap (if any) between the target and allocated capacity of the Spot Fleet.

The AWS EC2 Spot Fleet Test provides administrators with these useful insights, so that they can change their Spot Fleet requests (if required and possible) to maximize availability of their applications and minimize the cost of their application workload!

This test automatically discovers the Spot Fleet requests and reports the target capacity, fulfilled capacity, and pending capacity of each request. In the process, the test reveals those requests for which EC2 is unable to achieve / maintain the requested capacity. The test also reports the count of instance pools that are included in Spot Fleet requests and the number of pools that are actually eligible for fulfilling these requests. This way, the test points to those Spot Fleets with very few eligible instance pools to launch instances from. The corresponding Spot Fleet request can be scrutinized to understand the reason for the same, and modified (if required). Additionally, the test alerts administrators if many instances launched in response to a Spot Fleet request are terminating, so that they can investigate what is causing the termination.

Optionally, you can configure the test to report metrics for each instance type or Availability Zone included in the Spot Fleet requests. This analysis will help administrators further fine-tune their Spot Fleet request specifications.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each Spot Fleet request / Instance type / Availability Zone

First-level descriptor: AWS Region

Second-level descriptor: Spot Fleet request / Instance type / Availability Zone, depending upon the option chosen from the EC2Spot Filter Name.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.

Parameter	Description
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
EC2Spot Filter Name	By default, this test report metrics for each Spot Fleet Request. Accordingly, the EC2Spot Filter Name parameter is set to **FleetRequestID** by default. If required, you can override this default setting by setting the EC2Spot Filter Name parameter to one of the following: • **InstanceType** - To make sure that this test reports metrics for each instance type included in the Spot Fleet requests, select this option. • **Availability Zone** - To make sure that this test reports metrics for each Availability Zone included in the Spot Fleet requests, select this option.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Available instance pools	By default, this measure reports the number of	Number	

Measurement	Description	Measurement Unit	Interpretation
	instance pools that are included in this Spot Fleet request. If the EC2Spot Filter Name is set to **InstanceType**, then this measure will represent the number of instance pools of this instance type, included across Spot Fleet requests. If the EC2Spot Filter Name is set to **AvailabilityZone**, then this measure will represent the number of instance pools in this Availability zone, included across Spot Fleet requests.		
Bids submitted capacity	By default, this measure reports the number of bids that have been made for the target capacity of this Spot Fleet request. If the EC2Spot Filter Name is set to **InstanceType**, then this measure will represent the number of bids that have been made for instances of this type, across all Spot Fleet requests. If the EC2Spot Filter Name is set to **AvailabilityZone**, then this measure will represent the number of bids that have been made for instances in this Availability Zone, across all	Number	

Measurement	Description	Measurement Unit	Interpretation
	Spot Fleet requests.		
Eligible instance pools	By default, this measure reports the count of instance pools that are eligible for fulfilling this Spot Fleet request. If the EC2Spot Filter Name is set to **InstanceType**, then this measure will represent the number of instance pools with instances of this type that are eligible for fulfilling one/more Spot Fleet requests. If the EC2Spot Filter Name is set to **AvailabilityZone**, then this measure will represent the number of instance pools in this Availability Zone that are eligible for fulfilling one/more Spot Fleet requests.	KB	A pool is ineligible when either (1) The Spot price is higher than the On-Demand price or (2) the maximum price/bid price is lower than the Spot price. If only few instance pools are eligible for a Spot Fleet request, you may want to review that Spot Fleet request to see if the maximum price quoted therein needs to be changed.
Target capacity	By default, this measure reports the target capacity of this Spot Fleet request. If the EC2Spot Filter Name is set to **InstanceType**, then this measure will represent the number of instances of this type that are part of the target capacity across Spot Fleet requests. If the EC2Spot Filter Name	Number	

Measurement	Description	Measurement Unit	Interpretation
	is set to **AvailabilityZone**, then this measure will represent the number of instances in this Availability Zone that are part of target capacity across Spot Fleet requests.		
Fulfilled capacity	By default, this measure reports the number of instances that have been launched in fulfillment of this Spot Fleet request. If the EC2Spot Filter Name is set to **InstanceType**, then this measure will represent the number of instances of this type that have been launched in fulfillment of one/more Spot Fleet requests. If the EC2Spot Filter Name is set to **AvailabilityZone**, then this measure will represent the number of instances that have been launched from this Availability Zone in fulfillment of one/more Spot Fleet requests.	Number	Ideally, the value of this measure should be equal to the value of the Target capacity measure for a request. A very low value indicates that the target capacity of the request is yet to be achieved.
Pending capacity	By default, this measure reports the number of instances that are still to be launched to fulfill the target capacity of the Spot Fleet request. If the EC2Spot Filter Name is set to **InstanceType**,	Number	Ideally, the value of this measure should be low for a request.

Measurement	Description	Measurement Unit	Interpretation
	then this measure will represent the number of instances of this type that are still to be launched against the target capacity across all Spot Fleet requests. If the EC2Spot Filter Name is set to **AvailabilityZone**, then this measure will represent the number of instances that are still to be launched from this Availability Zone against the target capacity across Spot Fleet requests.		
Allocated capacity	By default, this measure reports the percentage of the target capacity of this Spot Fleet request that has been fulfilled. If the EC2Spot Filter Name is set to **InstanceType**, then this measure will represent the percentage of the target capacity across all Spot Fleet requests that has been fulfilled by instances of this type. If the EC2Spot Filter Name is set to **AvailabilityZone**, then this measure will represent the percentage of the target capacity across Spot Fleet requests that has been fulfilled by instances in this Availability Zone.	Percent	Ideally, the value of this measure should be 100% for a request. A very low value indicates non-fulfillment of the target capacity of that request. You may want to take another look at that Spot Fleet request to determine whether/not the specifications of the request need to be changed to ensure that the target capacity is achieved.

Measurement	Description	Measurement Unit	Interpretation
Maximum allocated capacity	By default, this measure reports the maximum value of the *Allocated capacity* measure across all Spot Instance pools specified in this Spot Fleet request. If the EC2Spot Filter Name is set to **InstanceType**, then this measure will represent the maximum value of the *Allocated capacity* measure across Spot Instance pools with instances of this type. If the EC2Spot Filter Name is set to **AvailabilityZone**, then this measure will represent the maximum value of the *Allocated capacity* measure across Spot Instance pools with instances of in this Availability Zone.	Percent	
Terminating capacity	By default, this measure reports the number of instances that have been terminated in this Spot Fleet owing to interruptions. If the EC2Spot Filter Name is set to **InstanceType**, then this measure will represent the number of instances of this type that have been terminated across Spot Fleets, because of interruptions. If the EC2Spot Filter Name	Number	When you use Spot Instances, you must be prepared for interruptions. Amazon EC2 can interrupt your Spot Instance when the Spot price exceeds your maximum price, when the demand for Spot Instances rises, or when the supply of Spot Instances decreases. If the value of this measure is very high for a request, you may want to figure out what caused the interruption. If it is because the max price is lower than spot price, then see if you can modify the maximum price of that request.

Measurement	Description	Measurement Unit	Interpretation
	is set to **AvailabilityZone**, then this measure will represent the number of instances in this Availability Zone that have been terminated across Spot Fleets, owing to interruptions.		

4.7.8 AWS EC2 Container - ECS Tests

AWS users can opt to run instances within Elastic Compute Cloud (EC2) or look into using containers. Amazon EC2 Container Service (ECS) manages Docker containers within AWS, allowing users to easily scale up or down and evaluate and monitor CPU usage. These AWS containers run on a managed cluster of EC2 instances, with ECS automating installation and operation of the cluster infrastructure. The first step to get started with ECS therefore is to create a cluster and launch EC2 instances in it. Then, create task definitions. A task is one or more Docker containers running together for one service or a microservice. When configuring a container in your task definition, you need to define the container name and also indicate how much memory and how many CPU units you want to reserve for each container. Finally, you will have to create a service, so that you can run and maintain a specified number of instances of a task definition simultaneously.

Time and again, administrators will have to check on the resource usage of each cluster, so that they can identify those clusters that have been consistently over-utilizing the CPU and memory resources. Resource usage at the individual service-level should also be monitored, so that administrators can figure out whether the excessive resource consumption by a cluster is because the cluster itself does not have enough resources at its disposal, or because one/more services running on the cluster are depleting the resources. Using the AWS EC2 Container - ECS test, administrators can monitor resource usage both at the cluster and the service-level.

For each AWS region, this test auto-discovers the clusters configured in that region and also the services running on each cluster. CPU and memory usage is then reported for each cluster and service, alongside the CPU and memory reservations (of all tasks) per cluster. These insights help administrators understand where there is a contention for resources - at the cluster-level? or at the service-level? or both? - and accordingly decide what needs to be done to optimize resource usage:

- Should more container instances be added to the cluster to increase the amount of resources at its disposal?
- Should the task definitions of the resource-hungry services be fine-tuned so that the service has more resources to use?

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test:

One set of results for each cluster:service pair in each region of the AWS EC2 cloud

First-level descriptor: AWS EC2 region name

Second-level descriptor: cluster name and/or clustername:servicename

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the

Parameter	Description
	CONFIRM PASSWORD text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
ECS Filter Name	By default, this test reports metrics for each service that is running on a cluster. Accordingly, *ServiceName* is the default selection from the **ECS FILTER** drop-down. If you do not want service-level metrics, then you can configure the test to report resource usage at the cluster-level alone. For this, just select *ClusterName* from the **ECS FILTER** drop-down. If this is done, then the test will only report cluster names as descriptors.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
CPU reservation:	The percentage of CPU units that are reserved by running tasks in this cluster.	Percent	**This measure is reported at the cluster-level only - i.e., for the ClusterName descriptor alone.** This value is computed using the

Measurement	Description	Measurement Unit	Interpretation
			following formula: *Total CPU units reserved by ECS tasks on the cluster / Total CPU units that were registered for all the container instances in the cluster * 100* A value close to 100% indicates that almost all resources available to the cluster are being reserved by running tasks in that cluster. This implies that additional services cannot be configured on that cluster until more resources are made available to the cluster or until the CPU reservation of running tasks is reduced.
CPU utilization:	Indicates the percentage of CPU units used by this cluster or by this service	Percent	For a cluster, this value is computed using the following formula: *Total CPU units currently used by ECS tasks on this cluster / Total CPU units that were registered for all the container instances in this cluster * 100* A value close to 100% for this measure at the cluster-level could either indicate that the cluster is resource-starved or that one/more services running on the cluster are consuming excessive resources. If the reason for high CPU usage is the poor resource configuration of the cluster, then, you may want to add more instances to the cluster to add to its resource base. On the other hand, if the cluster is adequately sized with CPU, then you may want to check the value of this measure for each of the services running on the cluster . For a service, this value is computed

Measurement	Description	Measurement Unit	Interpretation
			using the following formula: *Total CPU units currently used by ECS tasks defined for this service / Total CPU units that are reserved for the tasks defined for this service * 100* Compare the value of this measure across services of a cluster to know which services of that cluster are guilty of over-utilization of CPU. Once the services are identified, check the CPU reservation of the task definitions of those services to determine whether sufficient resources have been allocated to those tasks. If not, increase the reservations to allow optimal resource usage.
Memory reservation:	The percentage of memory that is reserved by running tasks in this cluster.	Percent	**This measure is reported at the cluster-level only - i.e., for the ClusterName descriptor alone.** This value is computed using the following formula: *Total amount of memory reserved by ECS tasks on the cluster / Total amount of memory that was registered for all the container instances in the cluster * 100* A value close to 100% indicates that almost all resources available to the cluster are being reserved by running tasks in that cluster. This implies that additional services cannot be configured on that cluster until more resources are made available to the cluster or until the memory reservation of running tasks is reduced.
Memory utilization:	Indicates the percentage of memory used by this cluster or	Percent	For a cluster, this value is computed using the following formula: *Total memory currently used by*

Measurement	Description	Measurement Unit	Interpretation
	by this service		*ECS tasks on this cluster / Total memory that is registered for all the container instances in this cluster * 100* A value close to 100% for this measure at the cluster-level could either indicate that the cluster is resource-starved or that one/more services running on the cluster are consuming excessive resources. If the reason for high memory usage is the poor resource configuration of the cluster, then, you may want to add more instances to the cluster to add to its resource base. On the other hand, if the cluster is adequately sized with memory, then you may want to check the value of this measure for each of the services running on the cluster . For a service, this value is computed using the following formula: *Total memory currently used by ECS tasks defined for this service / Total memory reserved for the tasks defined for this service * 100* Compare the value of this measure across services of a cluster to know which services of that cluster are guilty of over-utilization of memory. Once the services are identified, check the memory reservation of the task definitions of those services to determine whether sufficient resources have been allocated to those tasks. If not, increase the reservations to allow optimal resource usage.
Is active?	Indicates whether this cluster is active or not.		**This measure is reported only for every cluster - i.e., only when the 'ECS Filter Name' parameter is set to**

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>ClusterName.

The value that this measure can report and its corresponding numeric value are listed in the table below:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Yes</td><td>1</td></tr><tr><td>No</td><td>0</td></tr></table>
Note:

By default, this measure reports one of the Measure Values in the table above to indicate whether/not a cluster is active. In the graph of this measure however, the same is indicated using the numeric equivalents only.</td></tr>
<tr><td>Active services</td><td>Indicates the number of active services in this cluster.</td><td>Number</td><td>This measure is reported only for every cluster - i.e., only when the 'ECS Filter Name' parameter is set to ClusterName.

To know the active services, use the detailed diagnosis of this measure. The details displayed in the detailed diagnosis include:

Service Name: The name of the active service

CPU utilization: Average percentage of CPU units that are used in the service.

Memory utilization: Average percentage of memory that is used in the service

Desired tasks: Number of containers desired per service

Running tasks: Number of containers running per service</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			Pending tasks: Number of containers pending per service
Running task	Indicates the number of tasks that are in the running state in this cluster.	Number	**These measures are reported only for every cluster - i.e., only when the 'ECS Filter Name' parameter is set to ClusterName.** A task definition is required to run Docker containers in Amazon ECS. You can define multiple containers in a task definition. Using a Service, Amazon ECS can run and maintain a specified number of instances (the "desired count") of a task definition simultaneously in an Amazon ECS cluster. Typically, when a task is first pushed into ECS, it is in the PENDING state. Once the task starts running, it switches to the RUNNING state. At any given point in time, the count of running tasks should be equal to the number of desired tasks for that cluster. If a task in a service stops, the task is killed and a new task is launched. This process continues until your service reaches the number of desired running tasks.
Pending task	Indicates the number of tasks in pending state in this cluster.	Number	
Container instances	Indicates the number of container instances that are assigned to this cluster.	Number	**This measure is reported only for every cluster - i.e., only when the 'ECS Filter Name' parameter is set to ClusterName.** To know which container instances are assigned to a cluster, use the detailed diagnosis of this measure. The details displayed as part of detailed diagnosis include the Instance ID, region to which

Measurement	Description	Measurement Unit	Interpretation
			the instance belongs, and the status of the instance. Additionally, the following are also displayed per instance: **Registered CPU:** Number of CPU units registered on the container instance. **Remaining CPU:** Number of CPU units remaining on the container instance. **Registered memory:** Number of Memory units registered on the container instance. **Remaining memory:** Number of Memory units remaining on the container instance. **Running tasks:** Number of running tasks for the container instance **Pending tasks:** Number of pending tasks for the container instance **Is container agent connected:** Indicates whether the container agent is connected to the instance or not. **Docker version:** The version of the Docker container instance From these details, you can quickly isolate those container instances that are running out of CPU and memory resources and those that are disconnected from the container agent.

4.7.9 AWS Elastic Load Balancing - ELB Test

Elastic Load Balancing distributes incoming application traffic across multiple EC2 instances, in multiple Availability Zones.

A load balancer accepts incoming traffic from clients and routes requests to its registered targets (such as EC2 instances) in one or more Availability Zones. The load balancer also monitors the health of its registered targets and ensures that it routes traffic only to healthy targets. When the load

balancer detects an unhealthy target, it stops routing traffic to that target, and then resumes routing traffic to that target when it detects that the target is healthy again.

You can add and remove instances from your load balancer as your needs change, without disrupting the overall flow of requests to your application. Elastic Load Balancing scales your load balancer as traffic to your application changes over time, and can scale to the vast majority of workloads automatically.

This way, Elastic Load Balancing increases the fault tolerance of your applications and improves overall application performance. By keeping an eye out for issues in Elastic Load Balancing, administrators can ensure the prompt detection and swift resolution of issues, and can thus prevent application performance degradation. For this, administrators can take the help of the AWS Elastic Load Balancing - ELB Test.

By default, this test reports metrics for each load balancer that is configured. Flaky connection between a load balancer and its backend instances, latent communication between a load balancer and its instances, and HTTP errors (if any) encountered during load balancing are promptly captured by the test and reported. This enables administrators to be forewarned of issues in load balancing, so that they can initiate measures to avert the issues before they impact application performance.

Optionally, you can configure the test to report metrics per Availability Zone. The zone-level insight will help administrators understand if instances in a particular zone experience more latencies/errors than instances in other zones.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each load balancer / Availability Zone

First-level descriptor: AWS Region

Second-level descriptor: Load balancer / Availability Zone, depending upon the option chosen from the **ELB Filter Name** parameter of this test

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.

Parameter	Description
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
ELB Filter Name	By default, this parameter is set to **LoadBalancerName**. This means that by default, this test will report metrics for each load balancer. If required, you can override this default setting by setting the ELB Filter Name parameter to **Availability Zone**. In this case, this test will report metrics for every Availability Zone in which the instances interacting with the load balancer reside.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Unestablished connections	By default, this measure represents the number of connections that were attempted but failed between this load balancer and a seemingly healthy backend instance. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of connection attempts that were attempted but failed between a load balancer and the seemingly healthy backend instances in this Availability Zone.	Number	Ideally, the value of this measure should be 0. Connection errors between ELB and your servers occur when ELB attempts to connect to a backend, but cannot successfully do so. This type of error is usually due to network issues or backend instances that are not running properly.
Healthy instances	By default, this measure represents the number of healthy instances that are registered with this load balancer. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of healthy instances in this Availability Zone that are registered with a load balancer.	Number	A newly registered instance is considered healthy after it passes the first health check. If cross-zone load balancing is enabled, the number of healthy instances for a load balancer is calculated across all Availability Zones. Otherwise, it is calculated per Availability Zone.
Unhealthy instances	By default, this measure represents the number of unhealthy instances that are registered with this load balancer.	Number	If an instance exceeds the unhealthy threshold defined for the health checks, ELB flags it and stops sending requests to that instance. The most common cause is the health

Measurement	Description	Measurement Unit	Interpretation
	If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of unhealthy instances in this Availability Zone.		check exceeding the load balancer's timeout. Make sure to always have enough healthy backend instances in each availability zone to ensure good performance. You should also correlate this metric with *Latency* and *Pending submission requests* to make sure you have enough instances to support the volume of incoming requests without substantially slowing down the response time.
Latency	By default, this measure represents the time that elapsed from when this load balancer sent the request to a registered instance till when the instance started to send the response headers. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the time that elapsed from when a load balancer sent a request to a registered instance in this Availability Zone, till when the instance started to send the response headers.	Secs	This metric measures your application latency due to request processing by your backend instances, not latency from the load balancer itself. Tracking backend latency gives you good insight on your application performance. If it's high, requests might be dropped due to timeouts, which can lead to frustrated users. High latency can be caused by network issues, overloaded backend hosts, or non-optimized configuration provided by AWS to troubleshoot high latency.
HTTP 2xx response codes	By default, this measure represents the number of HTTP 2xx (success) codes currently returned by the registered backend instances to this load balancer.	Number	This count does not include any response codes generated by the load balancer.

Measurement	Description	Measurement Unit	Interpretation
	If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of HTTP 2xx (success) codes currently returned by the registered backend instances in this Availability Zone.		
HTTP 3xx response codes	By default, this measure represents the number of HTTP 3xx (redirection) codes currently returned by the registered backend instances to this load balancer. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of HTTP 3xx (redirection) codes currently returned by the registered backend instances in this Availability Zone.	Number	This count does not include any response codes generated by the load balancer.
HTTP 4xx response codes	By default, this measure represents the number of HTTP 4xx (client error) codes currently returned by the registered backend instances to this load balancer. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of HTTP 4xx (client error)	Number	This count does not include any response codes generated by the load balancer.

Measurement	Description	Measurement Unit	Interpretation
	codes currently returned by the registered backend instances in this Availability Zone.		
HTTP 5xx response codes	By default, this measure represents the number of HTTP 5xx (server error) codes currently returned by the registered backend instances to this load balancer. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of HTTP 5xx (server error) codes currently returned by the registered backend instances in this Availability Zone.	Number	This count does not include any response codes generated by the load balancer.
Completed requests	By default, this measure represents the number of requests this load balancer received and sent to the registered EC2 instances. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of requests a load balancer received and sent to the registered EC2 instances in this Availability Zone.	Number	This metric measures the amount of traffic your load balancer is handling. Keeping an eye on peaks and drops allows you to alert on drastic changes which might indicate a problem with AWS or upstream issues like DNS. If you are not using Auto Scaling then knowing when your request count changes significantly can also help you know when to adjust the number of instances backing your load balancer.
Rejected requests	By default, this measure represents the number of requests that have been rejected by this load	Number	When the *Pending submission requests* reaches the maximum of 1,024 queued requests, new requests are dropped, the user receives a 503

Measurement	Description	Measurement Unit	Interpretation
	balancer due to a full surge queue. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of requests that have been rejected by the load balancer due to a full surge queue for backend instances in this Availability Zone.		error, and the spillover count metric is incremented. In a healthy system, this metric is always equal to zero.
Pending submission requests	By default, this measure represents the number of inbound requests currently queued by this load balancer waiting to be accepted and processed by a backend instance. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of inbound requests currently queued by a load balancer waiting to be accepted and processed by backend instances in this Availability Zone.	Number	When your backend instances are fully loaded and can't process any more requests, incoming requests are queued, which can increase latency leading to slow user navigation or timeout errors. That's why this metric should remain as low as possible, ideally at zero. Backend instances may refuse new requests for many reasons, but it's often due to too many open connections. In that case you should consider tuning your backend or adding more backend capacity. The "max" statistic is the most relevant view of this metric so that peaks of queued requests are visible. Crucially, make sure the queue length always remains substantially smaller than the maximum queue capacity, currently capped to 1,024 requests, so you can avoid dropped requests.
HTTP 4XX client error	By default, this measure represents the number of HTTP 4xx errors (client error) currently returned by this load balancer.	Number	This is usually not much you can do about 4xx errors, since this metric basically measures the number of erroneous requests sent to ELB (which returns a 4xx code). If you

Measurement	Description	Measurement Unit	Interpretation
	If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of HTTP 4xx errors (client error) currently returned by the load balancer that is routing requests to the backend instances in this Availability Zone.		want to investigate, you can check in the ELB access logs to determine which code has been returned.
HTTP 5XX server error	By default, this measure represents the number of HTTP 5xx errors (server error) currently returned by this load balancer. If the ELB Filter Name is set to **AvailabilityZone**, then this measure represents the number of HTTP 5xx errors (server error) currently returned by the load balancer that is routing requests to the backend instances in this Availability Zone.	Number	The metric is reported if there are no healthy instances registered to the load balancer, or if the request rate exceeds the capacity of the instances (spillover) or the load balancer. This metric counts the number of requests that could not be properly handled. It can have different root causes: • If the error code is 502 (Bad Gateway), the backend instance returned a response, but the load balancer couldn't parse it because the load balancer was not working properly or the response was malformed. • If it's 503 (Service Unavailable), the error comes from your backend instances or the load balancer, which may not have had enough capacity to handle the request. Make sure your instances are healthy and registered with your load balancer.

Measurement	Description	Measurement Unit	Interpretation
			• If a 504 error (Gateway Timeout) is returned, the response time exceeded ELB's idle timeout. You can confirm it by checking if latency (see table below) is high and 5xx errors are returned by ELB. In that case, consider scaling up your backend, tuning it, or increasing the idle timeout to support slow operations such as file uploads. If your instances are closing connections with ELB, you should enable keep-alive with a timeout higher than the ELB idle timeout.

4.7.10 AWS Elastic MapReduce Test

Amazon EMR is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data. Additionally, you can use Amazon EMR to transform and move large amounts of data into and out of other AWS data stores and databases, such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.

The central component of Amazon EMR is the cluster. A cluster is a collection of Amazon Elastic Compute Cloud (Amazon EC2) instances. Each instance in the cluster is called a node. Each node has a role within the cluster, referred to as the node type. Amazon EMR also installs different software components on each node type (master node, core node, task node), giving each node a role in a distributed application like Apache Hadoop.

Amazon EMR service architecture consists of several layers, each of which provides certain capabilities and functionality to the cluster.

- **Storage layer:** The storage layer includes the different file systems that are used with your cluster. There are several different types of storage options such as, Hadoop Distributed File System (HDFS), EMR File System (EMRFS), and the Local File System.
- **Cluster resource management layer:** The resource management layer is responsible for managing cluster resources and scheduling the jobs for processing data.
- **Data processing frameworks layer:** The data processing framework layer is the engine used to process and analyze data. The main processing frameworks available for Amazon EMR are Hadoop MapReduce and Spark.
- **Applications and Programs:** Amazon EMR supports many applications, such as Hive, Pig, and the Spark Streaming library to provide various capabilities.

When a cluster is launched, you choose the frameworks and applications to install for your data processing needs. Once the chosen applications are installed, you should define the work to be done by the cluster. This can be done using Map and Reduce functions, using the EMR console / EMR API / AWS CLI, using the Hadoop API, or with the help of the interface provided by Hive or Pig. Then, the cluster processes data. To enable data processing in the cluster, you can either submit jobs or queries directly to the applications that are installed on your cluster or run steps in the cluster. Once data processing is complete, the cluster automatically shuts down (unless, auto-terminate is disabled).

If data processing is slow in a cluster, then administrators should be able to quickly and accurately tell what could be causing the slowness - map tasks? reduce tasks? storage contention at the cluster? under- utilization of task nodes and core nodes? To proactively detect processing bottlenecks in a cluster and isolate the root- cause of the bottleneck, administrators should periodically run the AWS Elastic MapReduce Test!

This test automatically discovers the clusters on Amazon EMR and tracks the status of each cluster and the progress of cluster activities. In the process, the test turns the spot light on clusters that are processing data slowly, and provides useful pointers to what could be slowing down processing.

Optionally, you can configure this test to report metrics for every job that a cluster runs. This will enable administrators to identify those jobs that could be slowing down data processing and what tasks are performed by the slow jobs - map tasks? or reduce tasks?

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each cluster / job

First-level descriptor: AWS Region

Second-level descriptor: JobFlowID or ClusterID / JobId depending upon the option chosen from the **EMR Filter Name** parameter of this test

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
EMR Filter Name	By default, this parameter is set to **JobFlowId**. This is the same as cluster ID, which is the unique identifier of a cluster in the form j-XXXXXXXXXXXX. In this case, this test

Parameter	Description
	will report metrics for every cluster. If required, you can override this default setting by setting the EMR Filter Name to **JobId**. You can use this to filter the metrics returned from a cluster down to those that apply to a single job within the cluster.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Is cluster idle?	Indicates whether/not this cluster is currently idle.		**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** This measure reports the value *Yes* if the cluster is idle, and *No* if it is not idle. A cluster is said to be idle if it is no longer performing work, but is still alive and accruing charges. The numeric values that correspond to these measure values are listed in the table below: **Measure Value** / **Numeric Value**: Yes / 1; No / 0 **Note:** By default, this measure reports the above-mentioned **Measure Value**s to indicate whether/not a cluster is idle. In the graph of this measure however, the same is indicated using the numeric equivalents.
Failed jobs	Indicates the number of jobs that have failed in this cluster.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter**

Measurement	Description	Measurement Unit	Interpretation
			Name parameter is set to 'JobFlowId'. Ideally, the value of this measure should be 0.
Running jobs	Indicates the number of jobs in this cluster that are currently running.	Numbe	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.**
Running map tasks	By default, this measure reports the total number of map tasks that are currently running in this cluster. If the EMR Filter Name parameter is set to **JobId**, then this measure will report the number of map tasks that are currently running for this job.	Number	Hadoop MapReduce is one of the main data processing frameworks available for Amazon EMR. Hadoop MapReduce is an open-source programming model for distributed computing. It simplifies the process of writing parallel distributed applications by handling all of the logic, while you provide the Map and Reduce functions. The Map function maps data to sets of key-value pairs called intermediate results. To put it simply, the Map procedure performs filtering and sorting (such as sorting students by first name into queues, one queue for each name).
Remaining map tasks	By default, this measure reports the total number of map tasks that are still to be run in this cluster. If the EMR Filter Name parameter is set to **JobId**, then this measure will report the number of map tasks that are still to be run for this job.	Number	A remaining map task is one that is not in any of the following states: Running, Killed, or Completed. A low value is desired for this measure. If the value of this measure is abnormally high for a cluster, you may want to change the EMR Filter Name of this test to JobId and see which job is still to run many map tasks. Such jobs could be slowing down the cluster.
Unused map task capacity	Indicates the unused map task capacity of this cluster.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to**

Measurement	Description	Measurement Unit	Interpretation
			'JobFlowId'. This is calculated as the maximum number of map tasks for a given cluster, less the total number of map tasks currently running in that cluster. A high value indicates that the cluster can take more load than it is currently. A very low value indicates that the cluster is already under duress as it is running too too many map tasks. The cluster may not be able to run any more map tasks until the ones running currently are either killed or completed. This can choke the cluster and slow down processing. Under such circumstances, you may want to increase the maximum number of map tasks the cluster can run to increase its capacity.
Remaining map tasks per slot	Indicates the ratio of the total map tasks remaining to the total map slots available in this cluster.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** A value close to 1 indicates that the available map slots are about to be exhausted. You may want to open more slots by increasing the maximum number of map tasks that a cluster can run.
Running reduce tasks	By default, this measure reports the total number of reduce tasks that are currently running in this cluster. If the EMR Filter Name parameter is set to **JobId**, then this measure	Number	Hadoop MapReduce is one of the main data processing frameworks available for Amazon EMR. Hadoop MapReduce is an open-source programming model for distributed computing. It simplifies the process of writing parallel distributed applications by handling all of the logic, while you provide the Map and Reduce

Measurement	Description	Measurement Unit	Interpretation
	will report the number of map tasks that are currently running for this job.		functions. The Reduce function combines the intermediate results (i.e., the key-value pairs) that are provided by the Map function, applies additional algorithms, and produces the final output. To put it simply, while the Map procedure performs filtering and sorting (such as sorting students by first name into queues, one queue for each name), the Reduce procedure performs a summary operation (such as counting the number of students in each queue, yielding name frequencies).
Remaining reduce tasks	By default, this measure reports the total number of reduce tasks that are still to be run in this cluster. If the EMR Filter Name parameter is set to **JobId**, then this measure will report the number of reduce tasks that are still to be run for this job.	Number	A remaining reduce task is one that is not in any of the following states: Running, Killed, or Completed. A low value is desired for this measure. If the value of this measure is abnormally high for a cluster, you may want to change the EMR Filter Name of this test to JobId and see which job is still to run many reduce tasks. Such jobs could be slowing down the cluster.
Unused reduce task capacity	Indicates the unused reduce task capacity of this cluster.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** This is calculated as the maximum number of reduce tasks for a given cluster, less the total number of reduce tasks currently running in that cluster. A high value indicates that the cluster can take more load than it is currently. A very low value indicates that the cluster is already under duress as it is running too too many reduce tasks. The cluster

Measurement	Description	Measurement Unit	Interpretation
			may not be able to run any more reduce tasks until the ones running currently are either killed or completed. This can choke the cluster and slow down processing. Under such circumstances, you may want to increase the maximum number of reduce tasks the cluster can run to increase its capacity.
Pending core nodes	Indicates the number of core nodes in this cluster waiting to be assigned.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** A core node is slave node with software components that run tasks and store data in the Hadoop Distributed File System (HDFS) on your cluster. All the core nodes requested may not be immediately available. Until a core node is assigned, tasks cannot be run on it. Therefore, a high value for this measure is often indicative of many pending tasks/requests. This can also slow down data processing in a cluster.
Running core nodes	Indicates the number of core nodes in this cluster that are working currently.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.**
Data nodes receive work from Hadoop	Indicates the percentage of data nodes that are receiving work from Hadoop.	Percent	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'**. Core nodes run the Data Node daemon to coordinate data storage as part of the Hadoop Distributed File System (HDFS). . They also run the Task Tracker daemon and perform other

Measurement	Description	Measurement Unit	Interpretation
			parallel computation tasks on data that installed applications require. For example, a core node runs YARN NodeManager daemons, Hadoop MapReduce tasks, and Spark executors. From the value of this measure, you can infer how many of the Running core nodes in the cluster are running Hadoop tasks and how many are engaged in other non-Hadoop tasks.
Pending task nodes	Indicates the number task nodes in this cluster that are currently waiting to be assigned.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** A task node is a slave node with software components that only run tasks. Task nodes are optional. All the task nodes requested may not be immediately available. Until a task node is assigned, tasks cannot be run on it. Therefore, a high value for this measure is often indicative of many pending tasks/requests. This can also slow down data processing in a cluster.
Functional task trackers	Indicates the percentage of task trackers that are functional.	Percent	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.**
Data read from S3	Indicates the amount of data that this cluster read from Amazon S3.	KB	**These measures are reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** Amazon S3 can be used as the file system for a cluster. EMRFS (EMR File System) is an implementation of HDFS

Measurement	Description	Measurement Unit	Interpretation
			used for reading and writing regular files from Amazon EMR directly to Amazon S3.
Data written to S3	Indicates the amount of data written to Amazon S3.	KB	These measures indicate the I/O generated by a cluster when reading from and writing into Amazon S3, using EMRFS. Compare the value of this measure across clusters to know which cluster is generating the maximum I/O.
Data read from HDFS	Indicates the amount of data read from HDFS (Hadoop Distributed File System) by this cluster.	KB	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.**
Data written to HDFS	Indicates the amount of data written into HDFS (Hadoop Distributed File System) by this cluster.	KB	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** HDFS can be used as a file system for an Amazon EMR cluster. HDFS is a distributed, scalable file system for Hadoop. HDFS distributes the data it stores across instances in the cluster, storing multiple copies of data on different instances to ensure that no data is lost if an individual instance fails. HDFS is ephemeral storage that is reclaimed when you terminate a cluster. HDFS is useful for caching intermediate results during MapReduce processing or for workloads that have significant random I/O. These measures indicate the I/O generated by a cluster when reading from and writing into HDFS. Compare the value of this measure across clusters to know which cluster is generating the maximum I/O.

Measurement	Description	Measurement Unit	Interpretation
HDFS utilization	Indicates the percentage of HDFS storage that this cluster is currently utilizing.	Percent	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** A value close to 100% is a cause for concern as it indicates that the cluster is about to run out of storage space.
Blocks in HDFS has not replicas	Indicates the number of blocks in which the HDFS storage used by this cluster has no replicas.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.** Blocks that do not have replicas may be corrupt blocks. The value 0 is therefore desired for this measure.
Concurrent data transfers	Indicates the total number of concurrent data transfers made by this cluster.	Number	**This measure is reported only for a cluster - i.e., only if the EMR Filter Name parameter is set to 'JobFlowId'.**
Has backup failed in HBase?	Indicates whethe/not the last backup failed for this cluster.		**This measure is reported only for HBase clusters.** HBase is an open source, non-relational, distributed database developed as part of the Apache Software Foundation's Hadoop project. With HBase on Amazon EMR, you can also back up your HBase data directly to Amazon Simple Storage Service (Amazon S3). If this backup fails for an HBase cluster, then this measure will report the value *Yes*. If the backup is successful, this measure will report the value *No*. The numeric values that correspond to these measure values are listed in the table below:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Yes</td><td>1</td></tr>
<tr><td>No</td><td>0</td></tr>
</table>
Note:
By default, this measure reports the above-mentioned Measure Values to indicate whether/not an HBase backup failed. In the graph of this measure however, the same is indicated using the numeric equivalents only.
</td></tr>
<tr><td>Time taken for previous backup to complete</td><td>Indicates the amount of time it took the previous backup of this cluster to complete.</td><td>Mins</td><td>This measure is reported only for HBase clusters.
This metric is set regardless of whether the last completed backup succeeded or failed. While the backup is ongoing, this metric returns the number of minutes after the backup started. A very high value for this measure can therefore indicate a backup delay or a failure.</td></tr>
<tr><td>Elapsed time after last successful backup started</td><td>Indicates the number of elapsed minutes after the last successful HBase backup started on this cluster.</td><td>Mins</td><td>This measure is reported only for HBase clusters.</td></tr>
</table>

4.7.11 AWS Simple Email Service - SES Test

Amazon Simple Email Service (Amazon SES) is a cost-effective email service built on the reliable and scalable infrastructure that Amazon.com developed to serve its own customer base. This service allows you to build an email functionality into an application that you are running on AWS. With Amazon SES, you can send transactional email, marketing messages, or any other type of high-quality content to your customers. You can also use Amazon SES to receive messages and

deliver them to an Amazon S3 bucket, call your custom code via an AWS Lambda function, or publish notifications to Amazon SNS.

Amazon SES has a set of sending limits to regulate the number of email messages that you can send and the rate at which you can send them. Depending upon the level of email activity in your environment, you may want to modify these limits, as any violation will result in mails not being sent at all. You may hence have to closely study the email activity in your environment and determine whether/not the sending limits need to be fine-tuned. The **AWS Simple Email Service - SES** test helps with this! By reporting the send quotas configured along with the count of mails sent and the send rate for each AWS region, this test readily provides you with all the information you need to take the right decision with regards to whether/not the quota needs to be reset.

Also, the key measure of the performance of any email service is successful message delivery. If a majority of the delivery attempts made at any given point in time resulted in bounces, rejections, or complaints, it is a problem condition that warrants an investigation. The **AWS Simple Email Service - SES** test proactively alerts you to such abnormalities! For each region, the test reports the count and percentage of emails bounced, mails rejected, and complaints received, and notifies you if these values exceed acceptable limits.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agent

Output of the test: One set of results for each AWS region

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and

Parameter	Description
	collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Sending quota:	Indicates the maximum number of emails that can be sent by this region in a day.	Emails/Day	The sending quota reflects a rolling time period. Every time you try to send an email, Amazon SES checks how many emails you sent in the previous 24 hours. As long as the total number of emails that you have sent is less than your quota, your send request will be accepted and your email will be sent. If you have already sent your full quota, your send request will be rejected with a throttling exception. You will not be able to send more emails until some of the previous

Measurement	Description	Measurement Unit	Interpretation
			sending rolls out of its 24-hour window.
Total sent:	Indicates the total number of emails that this region sent during the last 24 hours.	Number	If the value of this measure keeps growing closer to the value of the value of the Sending quota measure, it implies a high level of email activity in the region. Under such circumstances, it is best to increase the sending quota, so that the quota is not violated, causing SES to stop sending emails.
Current sents:	Indicates the number of emails that this region sent during the last measurement period.	Number	
Sents:	Indicates the percentage of sending quota that this region exhausted in the last 24 hours.	Percent	This measure is computed using the following formulat: (Total sent / Sending quota) * 100 If the value of this measure is consistently higher than 50%, it implies a high level of email activity in the region. Under such circumstances, it is best to increase the sending quota, so that the quota is not violated, causing SES to stop sending emails.
Max send rate:	Indicates the maximum number of emails that this region can send per second.	Emails/Sec	You can exceed this limit for short bursts, but not for a sustained period of time.
Total bounces:	Indicates the total number of emails bounced to this region during the last 24 hours.	Number	An email is hard-bounced when the email is rejected by the recipient's ISP or rejected by Amazon SES because the email address is on the Amazon SES suppression list. This measure reports the count of hard bounces alone.
Current bounces:	Indicates the number of	Number	The value of this measure should be

Measurement	Description	Measurement Unit	Interpretation
	emails that were bounced to this region during the last measurement period.		kept at a minimum, as excessive bounces constitute abuse and can put your AWS account at the risk of termination.
Bounce:	Indicates the percentage of emails that were bounced to this region during the last measure period.	Percent	Ideally, the value of this measure should be very low. A high value constitutes abuse and can put your AWS account at the risk of termination.
Complaints:	Indicates the number of complaints received by this region during the last measure period.	Number	If an email is accepted by the ISP and delivered to the recipient, but the recipient does not want the email and clicks a button such as "Mark as spam.", then SES will send you a complaint notification. The value of this measure should be kept at a minimum, as a large number of complaints constitute abuse and can put your AWS account at the risk of termination.
Complaint:	Indicates the percentage of complaints received by this region during the last measure period.	Percent	Ideally, the value of this measure should be very low. A high value constitutes abuse and can put your AWS account at the risk of termination.
Total rejected:	Indicates the number of emails sent by this region that were rejected during the last 24 hours.	Number	A rejected email is an email that Amazon SES initially accepted, but later rejected because the email contained a virus. Amazon SES notifies you by email and does not send the message. A high value for this measure is a cause for concern as it could indicate that your email system is severely infected.

Measurement	Description	Measurement Unit	Interpretation
Current rejected:	Indicates the number of emails sent by this region that were rejected during the last measurement period.	Number	A consistent rise in the value for this measure is a cause for concern as it could indicate that your email system is severely infected.
Rejected:	Indicates the percentage of emails sent by this region that were rejected during the last measurement period.	Percent	A high value for this measure is a cause for concern as it could indicate that your email system is severely infected.
Total delivery attempts:	Indicates the number of emails that were successfully delivered by this region to the recipient's mail server during the last 24 hours.	Number	A high value is desired for this measure.
Current delivery attempts:	Indicates the number of emails that were successfully delivered by this region to the recipient's mail server during the last measurement period.	Number	A consistent drop in the value of this measure is a cause of concern.

4.7.12 AWS VPC Flow Logs - Destination Test

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC (Virtual Private Cloud).

Using flow logs, you can easily troubleshoot why specific traffic is not reaching an instance, which in turn can help diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your instance, to profile your network traffic, and to look for abnormal traffic behaviors. A common use of these flow log records is to watch for abnormal and unexpected denied outbound connection requests, which could be an indication of a misconfigured or compromised EC2 instance.

To provide administrators with quick and useful insights into network traffic on VPCs , and to enable them to promptly identify and take action against abnormal traffic, the eG agent periodically reads flow logs and reports network traffic metrics. For this, the eG agent runs the following Flow Log tests:

- AWS VPC Flow Logs - Protocol
- AWS VPC Flow Logs - Destination
- AWS VPC Flow Logs - Source

The AWS VPC Flow Logs - Destination test for instance, automatically discovers the network interfaces handling traffic on the VPCs, and reports the following for each discovered interface:

- The destinations to which that interface sent traffic;
- The traffic that was routed to each destination;

In the event of a network congestion on an interface, these destination-wise statistics can help administrators accurately pinpoint which destination is probably contributing to the congestion.

If the detailed diagnostic capability of the test is enabled, then the eG agent will additionally provide deep-dive insights into the traffic by listing the top-10 flows for a destination in terms of the data transferred to it. If the traffic to a destination is abnormally high, then the detailed diagnostics will reveal:

- Has the destination been receiving a large amount of data consistently or is it just a momentary spike in traffic?
- Are transmissions from any particular source significantly higher than the rest? If so, which one?
- How often have network policies/security groups rejected the data sent to this destination?

For the AWS VPC Flow Logs - Destination test to run, the following pre-requisites should be fulfilled:

- You should first create flow logs. To create a flow log, you specify the resource for which you want to create the flow log (VPC, subnet, or network interface), the type of traffic to capture (accepted traffic, rejected traffic, or all traffic), the name of a log group in CloudWatch Logs to which the flow log will be published, and the ARN of an IAM role that has sufficient permission to publish the flow log to the CloudWatch Logs log group.
- After the flow logs are created, the flow data will be collected and published to the CloudWatch logs log group that was specified during flow log creation. To enable the eG agent to read these logs, you need to make sure that the flow logs are exported to Amazon S3.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each destination receiving traffic from every interface of a region

First-level descriptor: AWS Region

Second-level descriptor: Interface name

Third-level descriptor: Destination IP address

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region

Parameter	Description
	names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Top Info Display	By default, this test reports metrics for the top-15 destinations, in terms of the data transferred to them. Accordingly, this parameter is set to 15 by default. You can have this test report metrics for more or less number of destinations by changing the value of this parameter.
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Packet transferred	Indicates the number of packets transferred to this destination.	Number	Compare the value of this measure across destinations to know which destination is receiving the maximum traffic. You can then use the detailed diagnosis of this measure to view the complete details of the top-10 flows

Measurement	Description	Measurement Unit	Interpretation
			for that destination, in terms of the amount of data transferred to it. The details include the Source IP Address, Source Port, Destination IP Address, Destination Port, Packets Transferred, Data transferred, and Log status of each flow. From the Log status, you can quickly figure out whether the traffic handled by the flow was accepted by security groups/network policies or rejected. If many flows to a destination are rejected, you may have to investigate the reasons for the same, so that you do what is necessary to minimize or completely eliminate rejections.
Data transferred	Indicates the amount of data transferred to this destination.	KB	Compare the value of this measure across destinations to know which destination is receiving the maximum traffic.
Traffic data	Indicates what percentage of the total data handled by this interface was sent to this destination.	Percent	A value close to 100% for a destination indicates that almost all of the data handled by the interface was sent to that destination. By comparing the value of this measure across destinations, you can identify which destination is hogging the bandwidth resources.

The detailed diagnosis of the Packets transferred measure reveals the complete details of the top-10 flows for a particular destination, in terms of the amount of data transferred to it. The details include the Source IP Address, Source Port, Destination IP Address, Destination Port, Packets Transferred, Data transferred, and Log status of each flow. From the Log status, you can quickly figure out whether the traffic handled by the flow was accepted by security groups/network policies or rejected. If many flows to a destination are rejected, you may have to investigate the reasons for the same, so that you do what is necessary to minimize or completely eliminate rejections. By studying the flows, you can also identify which flow has transmitted an abnormally high volume of

data to this destination and which source transmitted that data. Such abnormal flows should be taken up for closer scrutiny.

Shows the top 10 flows for a destination

ACCOUNT ID	VERSION	SOURCE IP ADDRESS	SOURCE PORT	DESTINATION IP ADDRESS	DESTINATION PORT	PACKETS TRANSFERRED(NUM	DATA TRANSFERRED(KB)	LOG STATUS
Jan 18, 2018 01:53:17								
129794746678	2	172.30.0.244	3389	104.130.213.111	52639	7	1.8291	ACCEPT
129794746678	2	172.30.0.244	3389	104.130.213.111	52639	7	1.8291	ACCEPT
129794746678	2	172.30.0.244	3389	104.130.213.111	54815	7	1.8291	ACCEPT
129794746678	2	172.30.0.244	3389	104.130.213.111	54815	7	1.8291	ACCEPT
129794746678	2	172.30.0.244	3389	104.130.213.111	53909	7	1.8291	ACCEPT
129794746678	2	172.30.0.244	3389	104.130.213.111	53909	7	1.8291	ACCEPT

Figure 4.17: The detailed diagnosis of the Packets transferred measure of the AWS VPC Flow Logs - Destination test

4.7.13 AWS VPC Flow Logs - Protocol Test

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC (Virtual Private Cloud).

Using flow logs, you can easily troubleshoot why specific traffic is not reaching an instance, which in turn can help diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your instance, to profile your network traffic, and to look for abnormal traffic behaviors. A common use of these flow log records is to watch for abnormal and unexpected denied outbound connection requests, which could be an indication of a misconfigured or compromised EC2 instance.

To provide administrators with quick and useful insights into network traffic on VPCs , and to enable them to promptly identify and take action against abnormal traffic, the eG agent periodically reads flow logs and reports network traffic metrics. For this, the eG agent runs the following Flow Log tests:

- AWS VPC Flow Logs - Protocol
- AWS VPC Flow Logs - Destination
- AWS VPC Flow Logs - Source

The AWS VPC Flow Logs - Protocol test for instance, automatically discovers the network interfaces handling traffic on the VPCs, and reports the following for each discovered interface:

- The network protocols handled by that interface;
- The traffic generated for every protocol;

In the event of a network congestion on an interface, these protocol-wise statistics can help administrators accurately pinpoint the type of traffic (whether TCP, UDP, HTTP, etc.) that is contributing to the congestion.

If the detailed diagnostic capability of the test is enabled, then the eG agent will additionally provide deep-dive insights into the traffic by listing the top-10 flows for a protocol in terms of the data transferred.

If the traffic over a protocol is abnormally high, then the detailed diagnostics will reveal:

- Has the traffic over the protocol been high consistently or is it just a momentary spike?
- Are transmissions from any particular source and/or to any particular destination over this protocol, significantly higher than the rest? If so, which ones?
- How often have network policies/security groups rejected the data transmitted/received over this protocol?

For the AWS VPC Flow Logs - Protocol test to run, the following pre-requisites should be fulfilled:

- You should first create flow logs. To create a flow log, you specify the resource for which you want to create the flow log (VPC, subnet, or network interface), the type of traffic to capture (accepted traffic, rejected traffic, or all traffic), the name of a log group in CloudWatch Logs to which the flow log will be published, and the ARN of an IAM role that has sufficient permission to publish the flow log to the CloudWatch Logs log group.
- After the flow logs are created, the flow data will be collected and published to the CloudWatch logs log group that was specified during flow log creation. To enable the eG agent to read these logs, you need to make sure that the flow logs are exported to Amazon S3.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each protocol handled by every interface of a region

First-level descriptor: AWS Region

Second-level descriptor: Interface name

Third-level descriptor: Protocol name

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Top Info Display	By default, this test reports metrics for the top-15 destinations, in terms of the data transferred to them. Accordingly, this parameter is set to 15 by default. You can have this test report metrics for more or less number of destinations by changing the value of this parameter.
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be

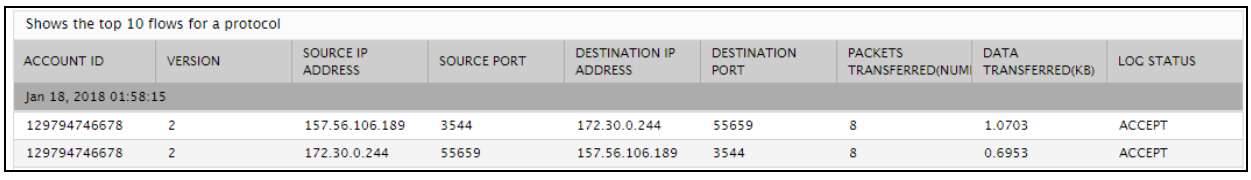
Parameter	Description
	generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Packet transferred	Indicates the number of packets transferred by this interface over this protocol.	Number	Compare the value of this measure across protocols to know what type of data (HTTP, TCP, UDP, etc.) generated the maximum traffic. You can then use the detailed diagnosis of this measure to view the complete details of the top-10 flows for that protocol, in terms of the amount of data transferred. The details include the Source IP Address, Source Port, Destination IP Address, Destination Port, Packets Transferred, Data transferred, and Log status of each flow. From the Log status, you can quickly figure out whether the traffic handled by the flow was accepted by security

Measurement	Description	Measurement Unit	Interpretation
			groups/network policies or rejected. If many flows are rejected for a protocol, you may have to investigate the reasons for the same, so that you do what is necessary to minimize or completely eliminate rejections.
Data transferred	Indicates the amount of data transferred by this interface over this protocol.	KB	Compare the value of this measure across protocols to know what type of data (HTTP, TCP, UDP, etc.) generated the maximum traffic.
Traffic data	Indicates what percentage of the total data transferred by this interface was transferred using this protocol.	Percent	A value close to 100% for a protocol indicates that almost all of the data transfers performed by the interface were over that protocol. By comparing the value of this measure across protocols, you can identify the type of traffic (HTTP, TCP, UDP, etc.) that is hogging the bandwidth resources.

The detailed diagnosis of the Packets transferred measure reveals the complete details of the top-10 flows for a particular protocol, in terms of the amount of data transferred. The details include the Source IP Address, Source Port, Destination IP Address, Destination Port, Packets Transferred, Data transferred, and Log status of each flow. From the Log status, you can quickly figure out whether the traffic handled by the flow was accepted by security groups/network policies or rejected. If many flows are rejected for a protocol, you may have to investigate the reasons for the same, so that you do what is necessary to minimize or completely eliminate rejections. By studying the flows, you can also identify the flow over which the maximum amount of data was transacted, and between which source and destination this traffic flowed. Such abnormal flows should be taken up for closer scrutiny.


Shows the top 10 flows for a protocol

ACCOUNT ID	VERSION	SOURCE IP ADDRESS	SOURCE PORT	DESTINATION IP ADDRESS	DESTINATION PORT	PACKETS TRANSFERRED(NUMI	DATA TRANSFERRED(KB)	LOG STATUS
Jan 18, 2018 01:58:15								
129794746678	2	157.56.106.189	3544	172.30.0.244	55659	8	1.0703	ACCEPT
129794746678	2	172.30.0.244	55659	157.56.106.189	3544	8	0.6953	ACCEPT

Figure 4.18: The detailed diagnosis of the Packets transferred measure of the AWS VPC Flow Logs - Protocol test

4.7.14 AWS VPC Flow Logs - Source Test

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC (Virtual Private Cloud).

Using flow logs, you can easily troubleshoot why specific traffic is not reaching an instance, which in turn can help diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your instance, to profile your network traffic, and to look for abnormal traffic behaviors. A common use of these flow log records is to watch for abnormal and unexpected denied outbound connection requests, which could be an indication of a misconfigured or compromised EC2 instance.

To provide administrators with quick and useful insights into network traffic on VPCs , and to enable them to promptly identify and take action against abnormal traffic, the eG agent periodically reads flow logs and reports network traffic metrics. For this, the eG agent runs the following Flow Log tests:

- AWS VPC Flow Logs - Protocol
- AWS VPC Flow Logs - Destination
- AWS VPC Flow Logs - Source

The AWS VPC Flow Logs - Source test for instance, automatically discovers the network interfaces handling traffic on the VPCs, and reports the following for each discovered interface:

- The sources that sent traffic to the interface;
- The traffic that was generated by each source;

In the event of a network congestion on an interface, these source-wise statistics can help administrators accurately pinpoint which destination is probably contributing to the congestion.

If the detailed diagnostic capability of the test is enabled, then the eG agent will additionally provide deep-dive insights into the traffic by listing the top-10 flows for a source in terms of the data transferred. If the traffic from a source is abnormally high, then the detailed diagnostics will reveal:

- Has the source been transmitting a large amount of data consistently or is it just a momentary spike in traffic?
- Are transmissions to any particular destination significantly higher than the rest? If so, which

one?

- How often have network policies/security groups rejected the data sent by the source?

For the AWS VPC Flow Logs - Source test to run, the following pre-requisites should be fulfilled:

- You should first create flow logs. To create a flow log, you specify the resource for which you want to create the flow log (VPC, subnet, or network interface), the type of traffic to capture (accepted traffic, rejected traffic, or all traffic), the name of a log group in CloudWatch Logs to which the flow log will be published, and the ARN of an IAM role that has sufficient permission to publish the flow log to the CloudWatch Logs log group.
- After the flow logs are created, the flow data will be collected and published to the CloudWatch logs log group that was specified during flow log creation. To enable the eG agent to read these logs, you need to make sure that the flow logs are exported to Amazon S3.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each destination receiving traffic from every interface of a region

First-level descriptor: AWS Region

Second-level descriptor: Interface name

Third-level descriptor: Source IP address

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure

Parameter	Description
	that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Top Info Display	By default, this test reports metrics for the top-15 destinations, in terms of the data transferred to them. Accordingly, this parameter is set to 15 by default. You can have this test report metrics for more or less number of destinations by changing the value of this parameter.
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability

Parameter	Description
	• Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Packet transferred	Indicates the number of packets transferred by this source.	Number	Compare the value of this measure across sources to know which source is generating the maximum traffic. You can then use the detailed diagnosis of this measure to view the complete details of the top-10 flows for that source, in terms of the amount of data transferred. The details include the Source IP Address, Source Port, Destination IP Address, Destination Port, Packets Transferred, Data transferred, and Log status of each flow. From the Log status, you can quickly figure out whether the traffic handled by the flow was accepted by security groups/network policies or rejected. If many flows from a source are rejected, you may have to investigate the reasons for the same, so that you do what is necessary to minimize or completely eliminate rejections.
Data transferred	Indicates the amount of data transferred by this source.	KB	Compare the value of this measure across sources to know which source is generating the maximum traffic.
Traffic data	Indicates what percentage of the total data to this interface was sent by this source.	Percent	A value close to 100% for a source indicates that almost all of the data handled by the interface was sent by that source. By comparing the value of this measure across sources, you can identify which source is hogging the bandwidth resources.

The detailed diagnosis of the Packets transferred measure reveals the complete details of the top-10 flows for a particular source, in terms of the amount of data transferred. The details include the Source IP Address, Source Port, Destination IP Address, Destination Port, Packets Transferred, Data transferred, and Log status of each flow. From the Log status, you can quickly figure out whether the traffic handled by the flow was accepted by security groups/network policies or rejected. If many flows from a source are rejected, you may have to investigate the reasons for the same, so that you do what is necessary to minimize or completely eliminate rejections. By studying the flows, you can also identify which flow has transmitted an abnormally high volume from this source and which destination received that data. Such abnormal flows should be taken up for closer scrutiny.

Shows the top 10 flows for a source

ACCOUNT ID	VERSION	SOURCE IP ADDRESS	SOURCE PORT	DESTINATION IP ADDRESS	DESTINATION PORT	PACKETS TRANSFERRED(NUMI	DATA TRANSFERRED(KB)	LOG STATUS
Jan 18, 2018 01:51:03								
129794746678	2	104.130.213.111	55484	172.30.0.244	3389	8	1.4453	ACCEPT
129794746678	2	104.130.213.111	55484	172.30.0.244	3389	8	1.4453	ACCEPT
129794746678	2	104.130.213.111	51466	172.30.0.244	3389	8	1.4453	ACCEPT
129794746678	2	104.130.213.111	51466	172.30.0.244	3389	8	1.4453	ACCEPT

Figure 4.19: The detailed diagnosis of the Packets transferred measure of the AWS VPC Flow Logs - Source test

4.7.15 AWS Internet of Things - IoT Test

AWS IoT provides secure, bi-directional communication between Internet-connected devices such as sensors, actuators, embedded micro-controllers, or smart appliances and the AWS Cloud. This enables you to collect telemetry data from multiple devices, and store and analyze the data. You can also create applications that enable your users to control these devices from their phones or tablets.

When using AWS IoT, devices are managed as *things*. A thing is a representation of a specific device or logical entity. It can be a physical device or sensor (for example, a light bulb or a switch on a wall). It can also be a logical entity like an instance of an application or physical entity that does not connect to AWS IoT but is related to other devices that do (for example, a car that has engine sensors or a control panel). Information about a thing is stored in the registry as JSON data.

These devices/things report their state by publishing messages, in JSON format, on MQTT topics. Each MQTT topic has a hierarchical name that identifies the device whose state is being updated. When a message is published on an MQTT topic, the message is sent to the AWS IoT MQTT message broker, which is responsible for sending all messages published on an MQTT topic to all clients subscribed to that topic. The message broker supports the use of the MQTT protocol to publish and subscribe and the HTTPS protocol to publish.

You can create rules that define one or more actions to perform based on the data in a message. For example, you can insert, update, or query a DynamoDB table or invoke a Lambda function. Rules use expressions to filter messages. When a rule matches a message, the rules engine invokes the action using the selected properties. Rules also contain an IAM role that grants AWS IoT permission to the AWS resources used to perform the action.

Each device has a shadow that stores and retrieves state information. Each item in the state information has two entries: the state last reported by the device and the desired state requested by an application. An application can request the current state information for a device. The shadow responds to the request by providing a JSON document with the state information (both reported and desired), metadata, and a version number. An application can control a device by requesting a change in its state. The shadow accepts the state change request, updates its state information, and sends a message to indicate the state information has been updated. The device receives the message, changes its state, and then reports its new state.

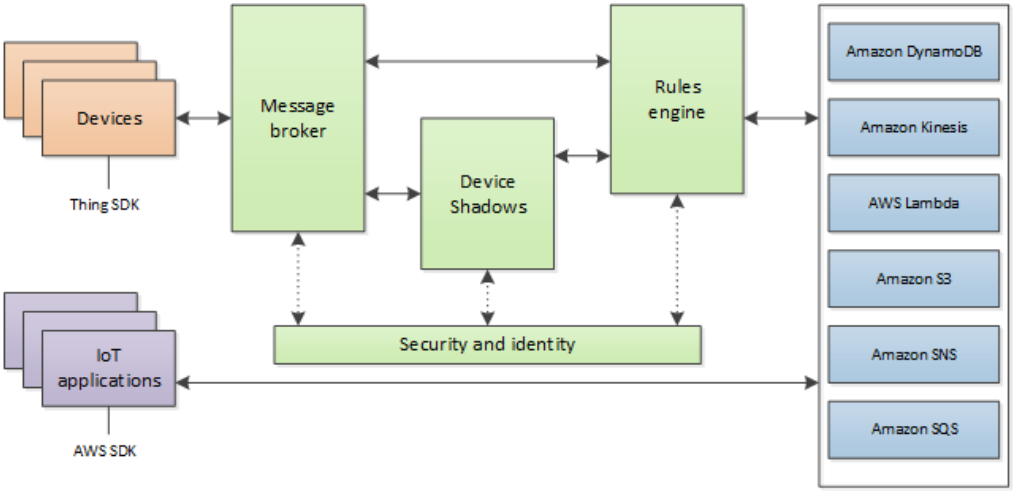


Figure 4.20: How AWS IoT works

Typically, AWS IoT policies are used to enable devices to connect to the message broker and get authorized, so they can perform AWS IoT operations, such as subscribing or publishing to MQTT topics via the broker, or get, update, or delete a device's shadow. If any of these policy actions fail or experience errors, then devices may no longer be able to communicate with the AWS cloud. Also, where messages match configured rules, if the defined rule actions fail, the failure may disrupt communication between the devices and critical AWS services such as AWS Lambda, Amazon S3, Amazon DynamoDB, etc. Moreover, AWS also imposes limits on the AWS IOT service; if these limits are violated, it might adversely impact the overall performance of AWS IOT . To ensure the peak performance of AWS IOT therefore, administrators should continuously track the different types of requests to the message broker, capture errors, failures, and violations promptly, and rapidly initiate measures to fix the faults. This is where the AWS Internet of Things - IOT Test helps!

By default, this test tracks the different types of requests (connection, publish, subscribe, unsubscribe, get/delete/update device shadow, etc.) to the message broker, and automatically discovers the protocols in use. For each protocol, the test then reports the count of requests that were successful, that violated the AWS IOT limits, and that which experienced errors. In the process, administrators can determine whether/not specific protocols are failing or are returning error responses frequently; such protocols could be suspect and may have to be investigated. Optionally, you can configure this test to report metrics for each rule or each action type configured for rules. This provides insights into errors (if any) in messages that match each rule and failed invocations of every action type.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each protocol / rule / action type

First-level descriptor: AWS Region

Second-level descriptor: Protocol / rule / action type, depending upon the option chosen against the **IOT Filter Name** parameter.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.

Parameter	Description
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
IOT Filter Name	By default, this parameter is set to **Protocol**. In this case, this test will report metrics for every protocol. For each protocol, the test reports the count of requests that were successful, that violated the AWS IOT limits, and that which experienced errors. If required, you can override this default setting by setting the IOT Filter Name to one of the following: • **RuleName**: For metrics on each rule that is configured, select this option. In this case, the test will only report the count of topics that match a rule condition and the count of errors (if any) in matching messages. • **ActionType**: For metrics on every action type that is configured for a rule, select this option. In this case, the test will only report the count of successful and failed invocations of each action type.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Connection request authorized errors	Indicates the number of connection requests made using that protocol, which could not be authorized by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** The value 0 is desired for this

Measurement	Description	Measurement Unit	Interpretation
			measure. A high value is a cause for concern as it indicates that many connection requests made using a particular protocol were unauthorized.
Connection request client errors	Indicates the number of connection requests made using that protocol, which were rejected because the MQTT message did not meet the requirements defined in AWS IoT limits.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value indicates that a connection request has been rejected. Typically, AWS requires that a client ID in a connection request should not be more than 128 bytes in size and should consist of only UTF-8 encoded characters. Requests from clients that do not fulfill this requirement will be rejected.
Connection request server errors	Indicates the number of connection requests made using this protocol that failed because an internal error occurred.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value implies that for one/more connection requests made using that protocol, the message broker returned the error code 500.
Connection request success	Indicates the number of connection requests made using this protocol that resulted in successful connections to the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** A high value is desired for this measure.

Measurement	Description	Measurement Unit	Interpretation
Connection request throttled	Indicates the number of connection requests made using this protocol that were throttled because the client exceeded the allowed connect request rate.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** By default, AWS IoT limits an account to a maximum of 300 MQTT CONNECT requests per second. If this request rate is exceeded, then AWS throttles connection requests. You may want to observe the variations to the value of this measure over time and figure out if too many connection requests are throttled. If so, you may want to request for changing the connect request rate limit.
Delete thing shadow requests accepted	Indicates the number of DeleteThingShadow requests made using this protocol, which were processed successfully.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** A DeleteThingShadow request deletes the shadow of a specified thing.
Get thing shadow requests accepted	Indicates the number of GetThingShadow requests made using this protocol, which were processed successfully.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** A GetThingShadow request gets the shadow of a specified thing.
Successful ping messages	Indicates the number of ping messages sent using this protocol, which were successfully received by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.**
Publish in authorized	Indicates the number of	Number	**This measure is reported only for**

Measurement	Description	Measurement Unit	Interpretation
errors	inbound publish requests made using this protocol, which could not be authorized by the message broker.		**each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** The value 0 is desired for this measure. A high value is a cause for concern as it indicates that many publish requests made using a particular protocol were unauthorized. Typically, publish requests that return the HTTP status code 401 are classified as unauthorized requests.
Publish in client errors	Indicates the number of inbound publish requests made using this protocol, which were rejected because the message did not meet the requirements defined in AWS IoT limits.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value indicates that a publish request has been rejected.
Publish in server errors	Indicates the number of inbound publish requests made using this protocol, which the message broker failed to process because an internal error occurred.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value implies that for one/more inbound publish requests made using that protocol, the message broker returned the error code 500.
Publish in success	Indicates the number of inbound publish requests made using this protocol that were successfully processed by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** A high value is desired for this measure.

Measurement	Description	Measurement Unit	Interpretation
Publish in throttled	Indicates the number of inbound publish requests made using this protocol, which were throttled because the client exceeded the allowed inbound request rate.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Typically, AWS IoT limits each client connection to 100 inbound publish requests per second. If this limit is exceeded, then subsequent inbound publish requests are throttled by AWS. You may want to observe variations to this measure over time and figure out if too many inbound publish requests are throttled. If so, you may want to request for changing the inbound publish request rate limit.
Publish out authorized errors	Indicates the number of outbound publish requests made using this protocol, which could not be authorized by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** The value 0 is desired for this measure. A high value is a cause for concern as it indicates that many outbound publish requests made using a particular protocol were unauthorized. Typically, publish requests that return the HTTP status code 401 are classified as unauthorized requests.
Publish out client errors	Indicates the number of outbound publish requests made using this protocol, which were rejected because the message did not meet the requirements defined in AWS IoT limits.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value indicates that an outbound publish

Measurement	Description	Measurement Unit	Interpretation
			request has been rejected.
Publish out success	Indicates the number of outbound publish requests made using this protocol that were successfully processed by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** A high value is desired for this measure.
Rules executed	Indicates the number of rules executed.	Number	
Subscription request authorized errors	Indicates the number of subscription requests made using this protocol, which could not be authorized by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** The value 0 is desired for this measure. A high value is a cause for concern as it indicates that many subscription requests made using a particular protocol were unauthorized. Typically, subscription requests that return the HTTP status code 401 are classified as unauthorized requests.
Subscription request client errors	Indicates the number of subscription requests made using this protocol, which were rejected because the message did not meet the requirements defined in AWS IoT limits.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value indicates that a subscription request has been rejected.
Subscription request server errors	Indicates the number of subscription requests made using this protocol, which the message broker failed to process because an internal error occurred.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value implies

Measurement	Description	Measurement Unit	Interpretation
			that for one/more inbound subscription requests made using that protocol, the message broker returned the error code 500.
Subscription request success	Indicates the number of subscription requests made using this protocol that were successfully processed by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** A high value is desired for this measure.
Subscription request throttled	Indicates the number of subscription requests made using this protocol, which were throttled because the client exceeded the allowed subscription request rate.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Typically, AWS IoT limits each client connection to subscribe to up to 50 subscriptions. A SUBSCRIBE request that pushes the total number of subscriptions past 50 results in the connection being disconnected. You may want to observe variations to this measure over time and figure out if too many subscription requests are throttled. If so, you may want to request for changing the subscription request rate limit.
Unsubscribe request client errors	Indicates the number of UNSUBSCRIBE requests made using this protocol, which were rejected because the UNSUBSCRIBE message did not meet the requirements defined in AWS IoT limits.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value indicates that an UNSUBSCRIBE request has been rejected.
Unsubscribe request	Indicates the number of	Number	**This measure is reported only for**

Measurement	Description	Measurement Unit	Interpretation
server errors	UNSUBSCRIBE requests made using this protocol, which the message broker failed to process because an internal error occurred.		**each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** Ideally, the value of this measure should be 0. A non-zero value implies that for one/more UNSUBSCRIBE requests made using that protocol, the message broker returned the error code 500.
Unsubscribe request success	Indicates the number of UNSUBSCRIBE requests made using this protocol that were successfully processed by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** A high value is desired for this measure.
Unsubscribe request throttled	Indicates the number of UNSUBSCRIBE requests made using this protocol, which were throttled because the client exceeded the allowed unsubscribe request rate.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** You may want to observe variations to this measure over time and figure out if too many UNSUBSCRIBE requests are throttled. If so, you may want to request for changing the request rate limit.
Update thing shadow requests accepted	Indicates the number of a UpdateThingShadow requests made using this protocol, which were received by the message broker.	Number	**This measure is reported only for each protocol - i.e., only if the IOT Filter Name parameter is set to 'Protocol'.** An UpdateThingShadow request updates the shadow of a specified thing.
Topic match	Indicates the number of incoming messages published on a topic on	Number	**This measure is reported only for each rule - i.e., only if the IOT Filter Name parameter is set to**

Measurement	Description	Measurement Unit	Interpretation
	which this rule is listening.		**'RuleName'.**
JSON parse errors	Indicates the number of JSON parse errors that occurred in messages published on a topic on which this rule is listening.	Number	**This measure is reported only for each rule - i.e., only if the IOT Filter Name parameter is set to 'RuleName'.** Ideally, the value of this measure should be 0.
Successful rule action invocations	Indicates the number of successful invocations of this rule action type.	Number	**This measure is reported only for each rule action type - i.e., only if the IOT Filter Name parameter is set to 'ActionType'.** Ideally, the value of this measure should be high.
Failed rule action invocations	Indicates the number of failed invocations of this rule action type.	Number	**This measure is reported only for each rule action type - i.e., only if the IOT Filter Name parameter is set to 'ActionType'.** Ideally, the value of this measure should be 0.

4.7.16 AWS Kinesis Firehose Test

Amazon Kinesis Data Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon Elasticsearch Service (Amazon ES), and Splunk.

With Kinesis Data Firehose, you don't need to write applications or manage resources. You configure your data producers to send data to Kinesis data delivery streams using different sources: a Kinesis stream, the Kinesis Agent, or the Kinesis Firehose API using the AWS SDK. After data is sent to a delivery stream, it is automatically delivered to the destination you choose.

For instance, with Amazon Kinesis Data Firehose, you can capture data continuously from connected devices such as consumer appliances, embedded sensors, and TV set-top boxes. Amazon Kinesis Data Firehose loads the data into Amazon Redshift, enabling you to provide your customers near real-time access to metrics, insights, and dashboards.

If data/records sent by the delivery stream do not reach the intended destination, the analytics derived from that data may be incomplete, inaccurate, or unusuable. To avoid this, it is necessary to track the ingestion and transmission of data and records by the data delivery streams to each of the destinations and promptly capture delivery failures. This is where the AWS Kinesis FireHost test helps!

This test auto-discovers the data delivery streams created using Kinesis FireHose. For each delivery stream, the test tracks the delivery attempts made by that stream to each destination and reports the count of successful deliveries. This way, the test sheds light on delivery failures and the destinations to which delivery failed. Additionally, the test also measures the throughput of each delivery stream by reporting the number of records and amount of data received and processed by that stream. In the process, you can accurately identify those data delivery streams that are experiencing processing bottlenecks. To help you isolate what could be causing a stream to process data/records slowly, the test also reports the time taken by the stream to process different API calls. The API call/method that could be contributing to the slowness can thus be identified. The status of data transformation functions is also checked periodically, so that data transformation failures (if any) are brought to light.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each data delivery stream

First-level descriptor: AWS Region

Second-level descriptor: Data delivery stream

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and	In some environments, all communication with the AWS EC2 cloud and its regions

Parameter	Description
Proxy Port	could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Data delivered to ES	Indicates the amount of data that this data delivery stream indexed to Amazon ES.	KB	
Records delivered to ES	Indicates the number of records that this delivery stream indexed to Amazon ES.	Number	
Total records delivered successfully to ES	Indicates the number of successfully indexed records over the number of records that were attempted.	Number	A value lesser than 1 for this measure is indicative of delivery failures.
Data delivered to RedShift	Indicates the amount of data that this data delivery stream	KB	

Measurement	Description	Measurement Unit	Interpretation
	copied to Amazon RedShift.		
Records delivered to RedShift	Indicates the number of records this data delivery stream copied to Amazon RedShift.	Number	
Total records delivered successfully to RedShift	Indicates the number of successful Amazon Redshift COPY commands over the number of all Amazon Redshift COPY commands that were issued by this data delivery stream.	Number	A value lesser than 1 for this measure is indicative of delivery failures.
Data delivered to S3	Indicates the amount of data that this data delivery stream delivered to S3.	Number	
Records delivered to S3	Indicates the number of records that this data delivery stream delivered to S3.	Number	
Total records delivered successfully to S3	Indicates the number of successful Amazon S3 put commands over the number of all Amazon S3 put commands issued by this data delivery stream.	Number	A value lesser than 1 for this measure is indicative of delivery failures.
Age of the oldest record	Indicates the age (from getting into Kinesis Firehose to now) of the oldest record in this data delivery stream.	Secs	An abnormally high value for this measure could point to a record that is still in the data delivery stream and has not been delivered yet. The reasons for this delivery bottleneck will have to be investigated.
Incoming data	Indicates the amount of data coming into this data delivery stream.	KB	
Incoming records	Indicates the number of records ingested into this	Number	

Measurement	Description	Measurement Unit	Interpretation
	data delivery stream.		
Describedeliverystream latency	Indicates the time taken by Kinesis Firehose to perform the DescribeDeliveryStream operation for this data delivery stream.	Secs	The DescribeDeliveryStream API command describes the specified delivery stream and gets the status. If any slowness or low throughput is noticed in a data delivery stream, then you can compare the value of this measure with the other latency measures that this test reports for that data delivery stream to know which API call is slowing down processing.
Describedeliverystream erequests	Indicates the total number of Describedeliverystream requests for this data delivery stream.	Number	
Listdeliverystream operation latency	Indicates the time taken by Kinesis Firehose to execute the Listdeliverystreams API call and return output for this data delivery stream.	Secs	The ListDeliveryStreams operation Lists your delivery streams. If any slowness or low throughput is noticed in a data delivery stream, then you can compare the value of this measure with the other latency measures that this test reports for that data delivery stream to know which API call is slowing down processing.
Listdeliverystream requests	Indicates the total number of Listdeliverystream requests for this data delivery stream.	Number	
Putrecord latency	Indicates the time taken by Kinesis Firehose to execute	Secs	PutRecord writes a single data record into an Amazon Kinesis

Measurement	Description	Measurement Unit	Interpretation
	the Putrecord API operation to write a record into this data delivery stream.		Firehose delivery stream. If any slowness or low throughput is noticed in a data delivery stream, then you can compare the value of this measure with the other latency measures that this test reports for that data delivery stream to know which API operation is slowing down processing.
Putrecord requests	Indicates the total number of Listdeliverystream requests for this data delivery stream.	Number	
Putrecord data	Indicates the amount of data put into this data delivery stream using the PutRecord API operation.	KB	The PutRecord and PutRecordBatch operations together can put a maximum of 5 MB of data per second into a delivery stream for the US East (N. Virginia), US West (Oregon), and EU (Ireland) regions. The limit is 1 MB/Sec for US East (Ohio), US West (N. California), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), and EU (Frankfurt) regions. You can submit a limit increase request using the Amazon Kinesis Data Firehose Limits form. If the increased limit is much higher than the running traffic, it causes very small delivery batches to destinations, which is inefficient and can result in higher costs at the destination services. Be sure to increase the limit only to match current running traffic, and increase the limit further if traffic increases.

Measurement	Description	Measurement Unit	Interpretation
Putrecordbatch data	Indicates the amount of data put into this data delivery stream using the PutRecordBatch API operation.	KB	The PutRecord and PutRecordBatch operations together can put a maximum of 5 MB of data per second into a delivery stream for the US East (N. Virginia), US West (Oregon), and EU (Ireland) regions. The limit is 1 MB/Sec for US East (Ohio), US West (N. California), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), and EU (Frankfurt) regions. You can submit a limit increase request using the Amazon Kinesis Data Firehose Limits form. If the increased limit is much higher than the running traffic, it causes very small delivery batches to destinations, which is inefficient and can result in higher costs at the destination services. Be sure to increase the limit only to match current running traffic, and increase the limit further if traffic increases.
Putrecordbatch latency	Indicates the time taken to put records into this data delivery stream using the PutRecordBatch API operation.	Secs	If any slowness or low throughput is noticed in a data delivery stream, then you can compare the value of this measure with the other latency measures that this test reports for that data delivery stream to know which API operation is slowing down processing.
Putrecordbatch records	Indicates the number of records that were added to	Number	The PutRecord and

Measurement	Description	Measurement Unit	Interpretation
	this data delivery stream using the PutRecordBatch API operation.		PutRecordBatch operations together can put a maximum of 5000 records per second into a delivery stream for the US East (N. Virginia), US West (Oregon), and EU (Ireland) regions. The limit is 1000 records per second for US East (Ohio), US West (N. California), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), and EU (Frankfurt) regions. You can submit a limit increase request using the Amazon Kinesis Data Firehose Limits form. If the increased limit is much higher than the running traffic, it causes very small delivery batches to destinations, which is inefficient and can result in higher costs at the destination services. Be sure to increase the limit only to match current running traffic, and increase the limit further if traffic increases.
Putrecordbatch request	Indicates the total number of PutRecordBatch requests for this data delivery stream.	Number	
Updatedeliverystream latency	Indicates the time taken to update this data delivery stream using the UpdateDeliveryStream API operation.	Secs	If any slowness or low throughput is noticed in a data delivery stream, then you can compare the value of this measure with the other latency measures that this test reports for that data delivery stream to know which API operation is slowing down processing.

Measurement	Description	Measurement Unit	Interpretation
Updatedeliverystream request	Indicates the number of requests for UpdateDeliveryStream API operations for that data delivery stream.	Number	
Lambda function invocation	Indicates the time taken by Lambda function invocation for this data delivery stream.	Secs	Kinesis Data Firehose can invoke your Lambda function to transform incoming source data and deliver the transformed data to destinations. You can enable Kinesis Data Firehose data transformation when you create your delivery stream. **This measure is reported for only those data delivery streams for which data transformation has been enabled**. If any slowness or low throughput is noticed in a data delivery stream, then you can compare the value of this measure with the other latency measures that this test reports for that data delivery stream to know which API operation is slowing down processing.
Successful Lambda function invocations	Indicates the number of the successful Lambda function invocations over the number of the total Lambda function invocations by this data delivery stream.	Number	**This measure is reported for only those data delivery streams for which data transformation has been enabled**. A value less than 1 for this measure is indicative of one/more Lambda function invocation failures.

Measurement	Description	Measurement Unit	Interpretation
			Common causes for failure of Lambda invocations include a network timeout or reaching the Lambda invocation limit. In the event of such a failure, Kinesis Data Firehose retries the invocation three times by default. If the invocation does not succeed, Kinesis Data Firehose then skips that batch of records. The skipped records are treated as unsuccessfully processed records.
Successfully processed data	Indicates the amount of data that this data delivery stream has successfully processed.	KB	**This measure is reported for only those data delivery streams for which data transformation has been enabled**.
Successfully processed records	Indicates the number of records successfully processed by this data delivery stream.	Number	**This measure is reported for only those data delivery streams for which data transformation has been enabled.** For a data transformation-enabled data delivery stream, the value of this measure should be high ideally. A low value is indicative of data transformation failures. Common causes for failure of Lambda invocations include a network timeout or reaching the Lambda invocation limit. In the event of such a failure, Kinesis Data Firehose retries the invocation three times by default. If the invocation does not succeed, Kinesis Data Firehose then skips that batch of

Measurement	Description	Measurement Unit	Interpretation
			records. The skipped records are treated as unsuccessfully processed records.

4.7.17 AWS Kinesis Streams Test

Amazon Kinesis Data Streams enables you to build custom applications that process or analyze streaming data for specialized needs. Kinesis Data Streams can continuously capture and store terabytes of data per hour from hundreds of thousands of sources such as website clickstreams, financial transactions, social media feeds, IT logs, and location-tracking events. With the Kinesis Client Library (KCL), you can build Kinesis Applications and use streaming data to power real-time dashboards, generate alerts, implement dynamic pricing and advertising, and more. You can also emit data from Kinesis Data Streams to other AWS services such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon EMR, and AWS Lambda.

The following diagram illustrates the high-level architecture of Kinesis Data Streams. A Kinesis data stream is nothing but an ordered sequence of data records. A data record is the unit of data stored in a Kinesis data stream. A shard is a uniquely identified group of data records in a stream. A stream is composed of one or more shards, each of which provides a fixed unit of capacity. The producers continually push data records or shards of data to Kinesis Data Streams and the consumers process the data in real time. Consumers (such as a custom application running on Amazon EC2, or an Amazon Kinesis Data Firehose delivery stream) can store their results using an AWS service such as Amazon DynamoDB, Amazon Redshift, or Amazon S3.

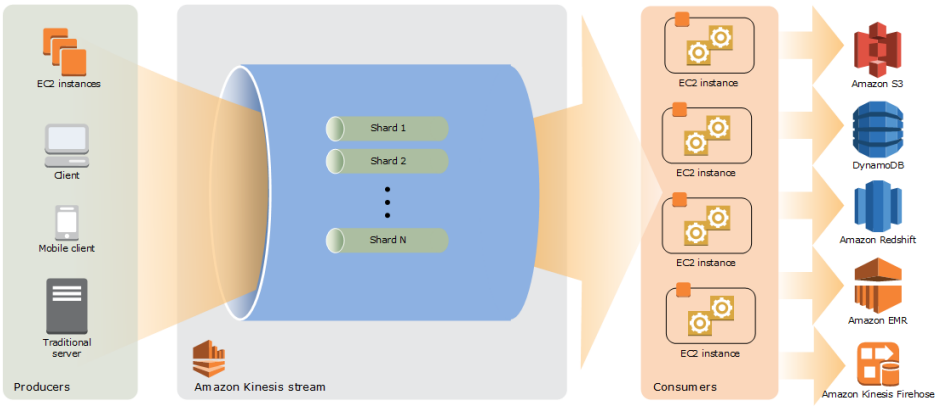


Figure 4.21: High level architecture of Kinesis Data Streams

Typically, you can work with data streams - i.e., put records into a stream, read records from it , etc. - using the Amazon Kinesis Data Streams API. At run time, any delay that a custom application experiences when streaming / analyzing data can be attributed to the delay in execution of these API calls. To capture the Kinesis Data Stream that is experiencing such a slowness, and to pinpoint the source of the slowness, use the AWS Kinesis Data Streams test.

This test automatically discovers the Kinesis Data Streams, and for each stream reports the time taken to put records in and get records from that stream. The test also promptly notifies administrators if any API operation fails, thus enabling administrators to troubleshoot and fix the failure before it causes any serious damage to application performance. The actual and provisioned throughput for each stream is tracked, and any throttling that occurs due to a throughput threshold breach is brought to the attention of administrators, so that the stream capacity/configuration can be changed according to the load.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each Kinesis data stream / shard

First-level descriptor: AWS Region

Second-level descriptor: Kinesis data stream / shard, depending upon the option chosen against the **Kinesis Filter Name** parameter.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and

Parameter	Description
	the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Kinesis Filter Name	By default, this test reports metrics for each Kinesis data stream. Accordingly, this parameter is set to **Streamname** by default. If required, you can override this default setting by configuring the test to report metrics for each Shard, instead. For this, select the **ShardId** option from the Kinesis Filter Name drop-down.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Getrecords operation data	Indicates the amount of data retrieved by the GetRecords API operation from this data stream.	KB	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**.
Getrecords iterator	By default, this measure represents the age of the last record in all GetRecords calls made to all shards in	Secs	Age is the difference between the current time and when the last record of the GetRecords call was written to the stream. A value of zero indicates that the records being read are completely caught up with the stream.

Measurement	Description	Measurement Unit	Interpretation
	this data stream. If the Kinesis Filter Name is set to ShardId, then this measure represents the age of the last record in all GetRecords calls made to this shard.		
Getrecords operation latency	Indicates the time taken per GetRecords operation performed on this data stream.	Secs	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**. Ideally, the value of this measure should be low. A high value indicates that the GetRecords API operation is very slow. Compare the value of this measure across data streams to know for which data stream the GetRecords operation is slowest.
Getrecords operation retrieved	Indicates the number of records retrieved from this data stream by the GetRecords API operation.	Number	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**.
Getrecords operation success	Indicates the number of successful Getrecords API operations for this data stream.	Number	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**. Ideally, the value of this measure should be high.
Incoming data	By default, this measure represents the total amount of data put into all shards in this data stream, by	KB	

Measurement	Description	Measurement Unit	Interpretation
	PutRecord and PutRecords operations. If the Kinesis Filter Name is set to ShardId, then this measure represents the total amount of data put into this shard by PutRecord and PutRecords operations.		
Incoming records	By default, this measure represents the total number of records put into all shards in this data stream, by PutRecord and PutRecords operations. If the Kinesis Filter Name is set to ShardId, then this measure represents the total number of records put into this shard by PutRecord and PutRecords operations.	Number	
Outgoing data	By default, this measure represents the total amount of data retrieved from all shards in this data stream. If the Kinesis Filter Name is set to	KB	

Measurement	Description	Measurement Unit	Interpretation
	ShardId, then this measure represents the total amount of data retrieved from this shard.		
Outgoing records	By default, this measure represents the total number of records retrieved from all shards in this data stream. If the Kinesis Filter Name is set to ShardId, then this measure represents the total number of records retrieved from this shard.	Number	
Putrecord operation data	Indicates the amount of data put into this data stream. by the PutRecord API operation.	KB	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**. Each call to PutRecord operates on a single record. Prefer the PutRecords operation unless your application specifically needs to always send single records per request, or some other reason PutRecords can't be used.
Putrecord operation latency	Indicates the time taken per PutRecord API operation performed on this data stream.	Secs	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**. Ideally, the value of this measure should be low. A high value indicates that the PutRecord API operation is very slow. Compare the value of this measure across

Measurement	Description	Measurement Unit	Interpretation
			data streams to know for which data stream the PutRecord operation is slowest.
Putrecord operation success	Indicates the number of successful PutRecord API operations for this data stream.	Number	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**. Ideally, the value of this measure should be high.
Putrecords operation data	Indicates the amount of data put into this data stream. by the PutRecords API operation.	KB	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**. The PutRecords operation sends multiple records to Kinesis Data Streams in a single request. By using PutRecords, producers can achieve higher throughput when sending data to their Kinesis data stream. Each record in the request can be as large as 1 MB, up to a limit of 5 MB for the entire request
Putrecords operation latency	Indicates the time taken per PutRecords API operation performed on this data stream.	Secs	**This measure is not reported for a shard - i.e., if the Kinesis Filter Name is set to ShardId, then this measure will not be reported**. Ideally, the value of this measure should be low. A high value indicates that the PutRecords API operation is very slow. Compare the value of this measure across data streams to know for which data stream the PutRecords operation is slowest.
Putrecords operation success	Indicates the number of PutRecords API operations, where at least one record succeeded for this data stream.	Number	Ideally, the value of this measure should be high. A low value is indicative of a high failure rate of PutRecords operations. By default, failure of individual records within a request does not stop the processing of subsequent records in a PutRecords request.

Measurement	Description	Measurement Unit	Interpretation
			This means that a response Records array includes both successfully and unsuccessfully processed records. You must detect unsuccessfully processed records and include them in a subsequent call.
Read provision throughput exceeded	By default, this measure represents the number of GetRecords calls throttled for this data stream. If the Kinesis Filter Name is set to ShardId, then this measure represents the number of GetRecords calls throttled for this shard.	Number	The maximum size of data that GetRecords can return is 10 MB. If a call returns this amount of data, subsequent calls made within the next five seconds throw ProvisionedThroughputExceededException. If there is insufficient provisioned throughput on the stream, subsequent calls made within the next one second throw ProvisionedThroughputExceededException. This exception implies that the request rate for the stream is too high, or the requested data is too large for the available throughput. The recommended solution for this problem is to reduce the frequency or size of your requests.
Write provision throughput exceeded	By default, this measure represents the number of PutRecord and PutRecords calls throttled for this data stream. If the Kinesis Filter Name is set to ShardId, then this measure represents the number of PutRecord and PutRecords calls throttled for this shard.	Number	If a PutRecord request cannot be processed because of insufficient provisioned throughput on the shard involved in the request, PutRecord throws ProvisionedThroughputExceededException. This exception implies that the request rate for the stream is too high, or the requested data is too large for the available throughput. The recommended solution for this problem is to reduce the frequency or size of your requests.
Iterator age	Indicates the age of the last record in all	Secs	

Measurement	Description	Measurement Unit	Interpretation
	GetRecords calls made against a shard, measured. By default, this measure represents the age of the last record in all GetRecords calls made to all shards in this data stream.		

4.7.18 AWS Lambda Test

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. In other words, AWS Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging. All you need to do is author your code in a language that AWS Lambda supports (currently Node.js, Java, C#, Go and Python), and upload your application code to AWS Lambda in the form of one or more Lambda functions. Using AWS Lambda, you can even maintain multiple versions of your in-production function code, and can also create aliases for each of your function versions for easy reference.

Typically, AWS Lambda is used to run code in response to events, such as changes to data in an Amazon S3 bucket or an Amazon DynamoDB table; to run your code in response to HTTP requests using Amazon API Gateway; or invoke your code using API calls made using AWS SDKs.

In such scenarios, if the Lambda function code fails or takes too long to execute, it can stall or even completely stop data/request processing by critical AWS services (eg., Amazon S3, Amazon DynamoDB, Amazon API Gateway, etc.) that rely on that code for their operations. To pre-empt the failure/delay of critical AWS services, administrators need to monitor each Lambda function that these services use and promptly capture problems in the function code. This is exactly what the AWS Lambda test does!

This test automatically discovers the Lambda functions, monitors the invocations of each function, and in the process, reports latencies and errors/failures in function execution. This enables

administrators to quickly and accurately identify slow and/or buggy functions, so that they take those functions and their codes up for closer review and fine-tuning.

Optionally, you can configure this test to report metrics for each version of a function or for every alias of a function version. This enables administrators to quickly compare the performance of different versions or aliases of a function, and then decide which version/alias to use in the production environment.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each Lambda function / version / alias in every region

First-level descriptor: AWS Region

Second-level descriptor: Function / Version / Alias, depending upon the option chosen from the **Lambda Filter Name** parameter of this test

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By

Parameter	Description
Password	default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Lambda Filter Name	By default, this parameter is set to **FunctionName**. This means that by default, this test will report metrics for each Lambda function that is in use. If required, you can override this default setting by setting the Lambda Filter Name parameter to one of the following: • **Version** - When you use versioning in AWS Lambda, you can publish one or more versions of your Lambda function. As a result, you can work with different variations of your Lambda function in your development workflow, such as development, beta, and production. Each Lambda function version has a unique Amazon Resource Name (ARN). If you want this test to report metrics for every version of every Lambda function, select the **Version** option. • **Alias** - AWS Lambda also supports creating aliases for each of your Lambda function versions. Conceptually, an AWS Lambda alias is a pointer to a specific Lambda function version. It's also a resource similar to a Lambda function, and each alias has a unique ARN. Each alias maintains an ARN for the function version to which it points. Select the **Alias** option if you want this test to report metrics for each alias that points to a version of a function.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Invocations	By default, this measure represents the number of times this function was	Number	Compare the value of this measure acros functions to know which function is used the maximum. The failure of

Measurement	Description	Measurement Unit	Interpretation
	invoked in response to an event or invocation API call. If the Lambda Filter Name is set to **Version**, then this measure represents the number of times this version of a function was invoked in response to an event or invocation API call.. If the Lambda Filter Name is set to **Alias**, then this measure represents the number of times this alias was invoked in response to an event or invocation API call.		such a function will naturally have a more adverse impact on performance and productivity than other functions.
Invocations failed due to errors	By default, this measure represents the number of invocations of this function failed due to errors (response code 4xx) If the Lambda Filter Name is set to **Version**, then this measure represents the number of number of invocations of this version of a function that failed due to errors. If the Lambda Filter Name is set to **Alias**, then this measure represents the number of invocations of this alias that failed due to errors.	Number	Ideally, the value of this measure should be 0. A non-zero value is indicative of one/more errors in the function code. To know what errors occurred during invocation, check the logs. Typically, each time the code is executed in response to an event, it writes a log entry into the log group associated with a Lambda function, which is /aws/lambda/<function name>. Following are some examples of errors that might show up in the logs: • If you see a stack trace in your log, there is probably an error in your code. Review your code and debug the error that the stack trace refers to. • If you see a permissions denied error

Measurement	Description	Measurement Unit	Interpretation
			in the log, the IAM role you have provided as an execution role may not have the necessary permissions. Check the IAM role and verify that it has all of the necessary permissions to access any AWS resources that your code references. • If you see a timeout exceeded error in the log, your timeout setting exceeds the run time of your function code. This may be because the timeout is too low, or the code is taking too long to execute. • If you see a memory exceeded error in the log, your memory setting is too low. Set it to a higher value. Typically, when creating a function, you need to mention the amount of memory that should be given to that function. Lambda uses this memory size to infer the amount of CPU and memory allocated to your function. Your function use-case determines your CPU and memory requirements. For example, a database operation might need less memory compared to an image processing function. The default value is 128 MB. The value must be a multiple of 64 MB.
Dead letter errors	By default, this measure represents the number of	Number	By default, a failed Lambda function invoked asynchronously is retried twice,

Measurement	Description	Measurement Unit	Interpretation
	times Lambda could not write the failure of this function to the configured dead letter queues. If the Lambda Filter Name is set to **Version**, then this measure represents the number of times Lambda could not write the failure of this version of a function to the configured dead letter queues. If the Lambda Filter Name is set to **Alias**, then this measure represents the number of times Lambda could not write the failure of this alias to the configured dead letter queues.		and then the event is discarded. Using Dead Letter Queues (DLQ), you can indicate to Lambda that unprocessed events should be sent to an Amazon SQS queue or Amazon SNS topic instead, where you can take further action. If the value of this measure keeps increasing, it implies that the event payload is consistently failing to reach the dead letter queue. Probable cause for this are as follows: • Permissions errors • Throttles from downstream services • Misconfigured resources • Timeouts tidings
Time taken to execute function	By default, this measure indicates the average elapsed wall clock time from when this function's code starts executing because of an invocation to when it stops executing. If the Lambda Filter Name is set to **Version**, then this measure represents the average elapsed wall clock time from when this version of a function's code starts executing because of an invocation to when it stops	Secs	Ideally, the value of this measure should be low. A high value indicates that a function/version/alias is taking too long to execute. To determine why there is increased latency in the execution of a Lambda function, do the following: • Test your code with different memory settings: If your code is taking too long to execute, it could be that it does not have enough compute resources to execute its logic. Try increasing the memory allocated to your function and testing the code

Measurement	Description	Measurement Unit	Interpretation
	executing. If the Lambda Filter Name is set to **Alias**, then this measure represents the average elapsed wall clock time from when this alias starts executing because of an invocation to when it stops executing.		again, using the Lambda console's test invoke functionality. You can see the memory used, code execution time, and memory allocated in the function log entries. Changing the memory setting can change how you are charged for duration. • Investigate the source of the execution bottleneck using logs: You can test your code locally, as you would with any other Node.js function, or you can test it within Lambda using the test invoke capability on the Lambda console, or using the asyncInvoke command by using AWS CLI. Each time the code is executed in response to an event, it writes a log entry into the log group associated with a Lambda function, which is named aws/lambda/<function name>. Add logging statements around various parts of your code, such as callouts to other services, to see how much time it takes to execute different parts of your code.
Function invocation that throttled	By default, this measure indicates the number of invocation attempts for this Lambda function that were throttled due to invocation rates	Number	The unit of scale for AWS Lambda is a concurrent execution (see Understanding Scaling Behavior for more details). However, scaling indefinitely is not desirable in all scenarios. For example, you may want

Measurement	Description	Measurement Unit	Interpretation
	exceeding the customer's concurrent limits (error code 429). If the Lambda Filter Name is set to **Version**, then this measure represents the number of invocation attempts that were throttled for this version of the Lambda function due to invocation rates exceeding the customer's concurrent limits (error code 429). If the Lambda Filter Name is set to **Alias**, then this measure represents the number of invocation attempts that were throttled for the version of the function that maps to this alias, due to invocation rates exceeding the customer's concurrent limits (error code 429).		to control your concurrency for cost reasons, or to regulate how long it takes you to process a batch of events, or to simply match it with a downstream resource. To assist with this, Lambda provides a concurrent execution limit control at both the account level and the function level. On reaching the concurrency limit associated with a function, any further invocation requests to that function are throttled, i.e. the invocation doesn't execute your function. Each throttled invocation increases the value of this measure for the corresponding function. AWS Lambda handles throttled invocation requests differently, depending on their source: • **Event sources that aren't stream-based:** Some of these event sources invoke a Lambda function synchronously, and others invoke it asynchronously. Handling is different for each: ◦ **Synchronous invocation:** If the function is invoked synchronously and is throttled, Lambda returns a 429 error and the invoking service is responsible for retries. The ThrottledReason error code explains whether you ran into a function level throttle (if specified) or an account level throttle (see note below). Each service may

Measurement	Description	Measurement Unit	Interpretation
			have its own retry policy. For example, CloudWatch Logs retries the failed batch up to five times with delays between retries. For a list of event sources and their invocation type, see Supported Event Sources. ◦ **Asynchronous invocation:** If your Lambda function is invoked asynchronously and is throttled, AWS Lambda automatically retries the throttled event for up to six hours, with delays between retries. Remember, asynchronous events are queued before they are used to invoke the Lambda function. • **Stream-based event sources:** For stream-based event sources (Kinesis and DynamoDB streams), AWS Lambda polls your stream and invokes your Lambda function. When your Lambda function is throttled, Lambda attempts to process the throttled batch of records until the time the data expires. This time period can be up to seven days for Kinesis. The throttled request is treated as blocking per shard, and Lambda doesn't read any new

Measurement	Description	Measurement Unit	Interpretation
			records from the shard until the throttled batch of records either expires or succeeds. If there is more than one shard in the stream, Lambda continues invoking on the non-throttled shards until one gets through.

4.7.19 AWS OpsWorks Test

Cloud-based computing usually involves groups of AWS resources, such as EC2 instances and Amazon Relational Database Service (RDS) instances. For example, a web application typically requires application servers, database servers, load balancers, and other resources. This group of instances is typically called a stack.

AWS OpsWorks Stacks, the original service, provides a simple and flexible way to create and manage stacks and applications. AWS OpsWorks Stacks lets you deploy and monitor applications in your stacks. You can create stacks that help you manage cloud resources by grouping them.

For example, a stack whose purpose is to serve web applications might look something like the following:

- A set of application server instances, each of which handles a portion of the incoming traffic.
- A load balancer instance, which takes incoming traffic and distributes it across the application servers.
- A database instance, which serves as a back-end data store for the application servers.

A common practice is to have multiple stacks that represent different environments. A typical set of stacks consists of:

- A development stack to be used by developers to add features, fix bugs, and perform other development and maintenance tasks.
- A staging stack to verify updates or fixes before exposing them publicly.
- A production stack, which is the public-facing version that handles incoming requests from users.

The load on a stack will vary according to the environment it represents. For instance, a production stack that front-ends requests from users, may see more traffic than a development stack that is used only by a small set of developers. The optimal performance of a stack therefore relies on whether/not that stack is sized with sufficient resources (CPU and memory) to handle its load. If a stack is not sized commensurate to its load, the performance of that stack and the application it supports will be adversely impacted! To avoid this, administrators can use the AWS OpsWorks test!

Using the **AWS OpsWorks** test, adminstrators can track the load on a stack, measure how much CPU and memory that stack used to process this load, and can thus proactively detect potential resource contentions and/or overload conditions. With the help of the useful pointers provided by this test, administrators can easily pinpoint stacks that are improperly sized in terms of CPU and memory and quickly initiate measures to right-size them.

Optionally, you can configure this test to report the load and resource usage metrics for individual layers or instances that constitute a stack. A layer represents a set of EC2 instances that serve a particular purpose, such as serving applications or hosting a database server. Layers depend on Chef recipes to handle tasks such as installing packages on instances, deploying apps, and running scripts.

Instance-wise insights into performance reveal if there are enough instances in a stack to handle user requests. Administrators can then decide whether/not to add more instances to a stack. Layer-wise insights into performance enable administrators to understand whether resources can be managed better if layer configuration is fine-tuned.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each stack/layer/instance

First-level descriptor: AWS Region

Second-level descriptor: StackID/LayerID/InstanceID, depending upon the option chosen from the **OpsWorks Filter Name** parameter of this test.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.

Parameter	Description
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
OpsWorks Filter Name	By default, this parameter is set to **StackID**. This means that by default, this test will report metrics for each stack. If required, you can override this default setting by setting the **OpsWorks Filter Name** parameter to one of the following: • **LayerID** - Every stack contains one or more layers, each of which represents a stack component, such as a load balancer or a set of application servers. To view load and resource usage metrics per layer, set the OpsWorks Filter Name to **LayerID**. • **InstanceID** - An instance represents a computing resource, such as an Amazon

Parameter	Description
	EC2 instance, which handles the work of serving applications, balancing traffic, and so on. If you want this test to report OpsWorks Filter Name metrics for every instance, set the to **InstanceID**.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Idle CPU	By default, this measure represents the percentage of time for this stack did not use its CPU. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the percentage of time for which the CPU resources of the instances in this layer were idle. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the percentage of time for which the CPU of this instance was idle.	Percent	If the value of this measure is consistently close to 100% for a stack, it could mean that the instances in that stack are probably sized with more CPU than it requires. On the other hand, if the value of this measure is consistently low stack, it could mean that instances in the stack are utilizing their CPU resources excessively. To know which instances are hogging the CPU, you may want to configure this test to report metrics for each instance by setting the OpsWorks Filter Name to **InstanceID**.
Nice CPU	By default, this measure represents the percentage of time that the CPU of this stack is handling processes with a positive nice value. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the percentage of time for which the CPU	Percent	*nice* is a program found on Unix and Unix-like operating systems such as Linux. which is used to invoke a utility or shell script with a particular priority, thus giving the process more or less CPU time than other processes. A niceness of −20 is the highest priority and 19 is the lowest priority. If the value of this measure is constantly close to or equal to 100% for a stack, it implies that most of the time

Measurement	Description	Measurement Unit	Interpretation
	of this layer was handling processes with a positive nice value. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the percentage of time for which the CPU of this instance was handling processes with a positive nice value.		the majority of the instances in this stack are utilizing CPU for processing requests of a lower priority only. On the other hand, if the value of this measure is very low consistently, it means that high-priority programs are hogging the CPU, and not the low-priority programs. In the event of a CPU contention, you can use the value of this measure to determine where is your CPU time being spent - in progressing low-priority programs? or high-priority ones?
Steal CPU	By default, this measure represents the percentage of time that the instances of this stack waited for the hypervisor to allocate physical CPU resources. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the percentage of time the instances in this layer waited for the hypervisor to allocate physical CPU resources. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the number of times this instance waited for the hypervisor to allocate physical CPU resources.	Percent	If the value of this measure is greater than 10% for a stack for over 20 minutes, it means that a majority of the instances in the stack are waiting too long for physical CPU. This can cause the instances to run slower than they should. The probable causes for spikes in CPU steal time are as follows: • The instances are not sized with adequate CPU resources; • The physical server is over-sold and the instances are aggressively competing for resources Therefore, when you notice a consistent increase in the value of this measure, it is good practice to do one of the following: • Shut down the instance and move it to another physical

Measurement	Description	Measurement Unit	Interpretation
			server; • If steal time remains high, increase the CPU resources of the instances; • If steal time remains high even after resizing the instances, contact your hosting provider. Your host may be overselling physical servers.
System CPU	By default, this measure indicates the percentage of time the instances in this stack used CPU for processing system operations. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the percentage of time the instances in this layer used CPU for handling system operations. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the percentage of time this instance used CPU for handling system operations.	Percent	If instances in a stack are experiencing slowness, you may want to compare the value of these measures across instances to know which instance is hogging the CPU and while doing what - when processing system operations? user operations? or just waiting for I/O to complete?
User CPU	By default, this measure indicates the percentage of time the instances in this stack used CPU for processing user	Percent	

Measurement	Description	Measurement Unit	Interpretation
	operations. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the percentage of time the instances in this layer used CPU for handling user operations. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the percentage of time this instance used CPU for handling user operations.		
WaitIO CPU	By default, this measure indicates the percentage of time for which the CPU was ready to run, but could not because it was waiting for input/output operations on the instances of this stack to complete. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure indicates the percentage of time for which the CPU was ready to run, but could not because it was waiting for input/output operations on the instances of this layer to complete. If the **OpsWorks Filter**	Percent	

Measurement	Description	Measurement Unit	Interpretation
	Name is set to **InstanceID**, then this measure indicates the percentage of time for which the CPU was ready to run, but could not because it was waiting for input/output operations on this instance to complete.		
Buffered memory	By default, this measure represents the total amount of memory that is buffered for the instances in this stack. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the total amount of memory that is buffered for the instances in this layer. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the total amount of memory that is buffered for this instance.	KB	
Cached memory	By default, this measure represents the total amount of memory that is cached for the instances in this stack. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the total	KB	

Measurement	Description	Measurement Unit	Interpretation
	amount of memory that is cached for the instances in this layer. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the total amount of memory that is cached for this instance.		
Free memory	By default, this measure represents the total amount of memory that the instances in this stack are still to use. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the total amount of memory that the instances in this layer are yet to use. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the total amount of memory that is still unused by this instance.	KB	Ideally, the value of this measure should be close to the value of the Total memory measure. A consistent drop in the value of this measure is a cause for concern, as it implies that memory is been steadily drained. A very low value for this measure is indicative of excessive memory usage, which can significantly affect the performance of the instances.
Swap memory	By default, this measure represents the total amount of swap memory available for the instances in this stack. If the **OpsWorks Filter Name** is set to **LayerID**,	KB	An unusually high value for the swap usage can indicate a memory bottleneck.

Measurement	Description	Measurement Unit	Interpretation
	then this measure represents the total amount of swap memory available for the instances in this layer. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the total amount of swap memory that is available for this instance.		
Total memory	By default, this measure represents the total memory capacity of this stack across all its instances. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the total memory capacity of this layer across its instances. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the total memory capacity of this instance.	KB	
Used memory	By default, this measure represents the total memory used by all instances in this stack. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure	KB	Ideally, the value of this measure should be low. A consistent increase in the value of this measure is a cause for concern, as it implies that memory is been steadily drained. If the value of this measure is close to or equal to the value of the

Measurement	Description	Measurement Unit	Interpretation
	represents the total memory used by all instances in this layer. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the total memory used by this instance.		Total memory measure, it indicates excessive memory usage by instances. This can significantly affect the performance of the instances. To avoid this, make sure that your instances are sized on the basis of their load.
Load averaged over 1-minute	By default, this measure represents the load on the instances in this stack, averaged over a 1-minute window. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the load on the instances in this layer, averaged over a 1-minute time window. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the load on this instance, averaged over a 1-minute time window.	Percent	Compare the value of these measures across stacks to identify that stack that is consistently handling high traffic. As your incoming traffic varies, your stack may have either too few instances to comfortably handle the load or more instances than necessary. You can save both time and money by using time-based or load-based instances to automatically increase or decrease a layer's instances so that you always have enough instances to adequately handle incoming traffic without paying for unneeded capacity. Automatic scaling is based on two instance types, which adjust a layer's online instances based on different criteria: • **Time-based instances:** They allow a stack to handle loads that follow a predictable pattern by including instances that run only at certain times or on certain days. For example, you could start some instances after 6PM to perform nightly backup tasks or stop some instances on weekends when traffic

Measurement	Description	Measurement Unit	Interpretation
			is lower. • **Load-based instances:** They allow a stack to handle variable loads by starting additional instances when traffic is high and stopping instances when traffic is low, based on any of several load metrics. For example, you can have AWS OpsWorks Stacks start instances when the average CPU utilization exceeds 80% and stop instances when the average CPU load falls below 60%. A common practice is to use all three instance types together, as follows. • A set 24/7 instances to handle the base load. You typically just start these instances and let them run continuously. • A set of time-based instances, which AWS OpsWorks Stacks starts and stops to handle predictable traffic variations. For example, if your traffic is highest during working hours, you would configure the time-based instances to start in the morning and shut down in the evening. • A set of load-based instances, which AWS OpsWorks Stacks starts and stops to handle unpredictable traffic variations. AWS OpsWorks Stacks starts them when the load approaches the capacity of the stacks' 24/7 and time-based instances, and stops them when the traffic returns to

Measurement	Description	Measurement Unit	Interpretation
Load averaged over 5-minute	By default, this measure represents the load on the instances in this stack, averaged over a 5-minute window. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the load on the instances in this layer, averaged over a 5-minute time window. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the load on this instance, averaged over a 5-minute time window.	Percent	
Load averaged over 15-minute	By default, this measure represents the load on the instances in this stack, averaged over a 15-minute window. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the load on the instances in this layer, averaged over a 15-minute time window. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the load on this instance, averaged over a 15-minute	Percent	

Measurement	Description	Measurement Unit	Interpretation
	time window.		
Active processes	By default, this measure represents the number of processes currently active across all instances in this stack. If the **OpsWorks Filter Name** is set to **LayerID**, then this measure represents the number of processes currently active across all instances in this layer. If the **OpsWorks Filter Name** is set to **InstanceID**, then this measure represents the number of processes currently active on this instance.	Number	This is a good indicator of the current workload of a stack / layer / instance.

4.7.20 AWS Polly Test

Amazon Polly is a cloud service that converts text into lifelike speech. You just need to call the SynthesizeSpeech method, provide the text you wish to synthesize, select one of the available Text-to-Speech (TTS) voices, and specify an audio output format. Amazon Polly then synthesizes the provided text into a high-quality speech audio stream.

To manage voices and pronunciations and to convert the text that users type into speech, Amazon Polly internally calls the following API methods:

- **PutLexicon** - Stores a pronunciation lexicon in an AWS Region; Pronunciation lexicons enable you to customize the pronunciation of words.
- **ListLexicons** - Returns a list of pronunciation lexicons stored in an AWS Region;
- **GetLexicon** - Returns the content of the specified pronunciation lexicon stored in an AWS Region;

- **DeleteLexicon** - Deletes the specified pronunciation lexicon stored in an AWS Region;
- **DescribeVoices** - Returns the list of voices that are available for use when requesting speech synthesis;
- **SynthesizeSpeech** - Synthesizes UTF-8 input, plain text or SSML, to a stream of bytes.

A delay or error in any of the aforesaid API operations can adversely impact a user's experience with Amazon Polly. This is why, whenever a cloud user complains of issues in text-to-speech conversion, administrators should be able to quickly identify the precise API operation causing the issue and troubleshoot accordingly. This is exactly where the AWS Polly test helps!

For each API operation (GetLexicon, PutLexicon, DescribeVoices, etc.), this test reports the responsiveness of that operation and the count of HTTP errors encountered by that operation. Whenever users complain of a poor Amazon Polly experience, administrators can use this test to isolate the exact operation that may have failed or may have slowed down causing the overall user experience with Polly to suffer.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each API operation that is called in every region

First-level descriptor: AWS Region

Second-level descriptor: Operation name

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure

Parameter	Description
	that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Characters in request	Indicates the average number of characters in a request for this operation.	Number	This measure reports billable characters only and does not include SSML tags. For the SynthesizeSpeech operation, AWS imposes a limit of 1500 billed characters (3000 total characters) for the input text. If any user complains that he/she encountered the the TextLengthExceededException when performing the SynthesizeSpeech operation, then check the value of this me measure for the SynthesizeSpeech descriptor to determine whether/not it has crossed 1500 characters.
Response latency	Indicates the average	Secs	Ideally, the value of this measure should

Measurement	Description	Measurement Unit	Interpretation
	latency between when the request was made for this operation and the start of the streaming response.		be low. A very high value is indicative of poor responsiveness / slowness of an API operation. In the event of a slowdown, you may want to compare the value of this measure across operations to identify the operation that is the slowest. If the SynthesizeSpeech operation turns out to be the slowest, you may want to check the size of the lexicons. Each lexicon can be upto 4000 characters in size. Larger the size, slower will be the SynthesizeSpeech operation.
HTTP 200 level code returned	Indicates the number of requests for this operation that returned the HTTP 200 level code response.	Number	This class of status codes indicates the action requested by the client was received, understood and accepted. A high value is desired for this measure.
HTTP 400 level code returned	Indicates the number of requests for this operation that returned the HTTP 400 level error code response.	Number	This class of status code is intended for situations in which the error seems to have been caused by the client. Ideally, the value of this measure should be 0.
HTTP 500 level code returned	Indicates the number of requests for this operation that returned the HTTP 500 level error code response.	Number	Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request. Ideally, the value of this measure should be 0.

4.7.21 AWS Simple Notification Service - SNS Test

Amazon Simple Notification Service (Amazon SNS) is a web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients. In Amazon SNS, there are two types of clients - publishers and subscribers—also referred to as producers and consumers.

Publishers communicate asynchronously with subscribers by producing and sending a message to a topic, which is a logical access point and communication channel. Subscribers (i.e., web servers, email addresses, Amazon SQS queues, AWS Lambda functions) consume or receive the message or notification over one of the supported protocols (i.e., Amazon SQS, HTTP/S, email, SMS, Lambda) when they are subscribed to the topic.

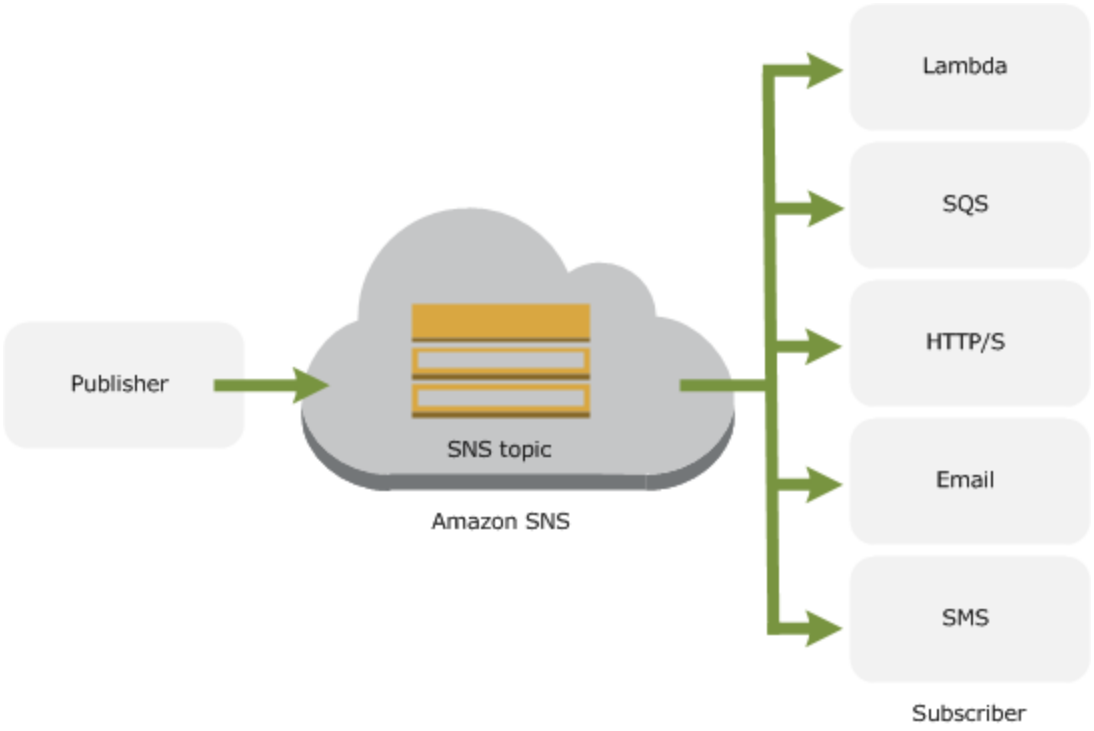


Figure 4.22: How SNS Works

When using Amazon SNS, you (as the owner) create a topic and control access to it by defining policies that determine which publishers and subscribers can communicate with the topic. A publisher sends messages to topics that they have created or to topics they have permission to publish to. Instead of including a specific destination address in each message, a publisher sends a message to the topic. Amazon SNS matches the topic to a list of subscribers who have subscribed to that topic, and delivers the message to each of those subscribers. Each topic has a unique name that identifies the Amazon SNS endpoint for publishers to post messages and subscribers to register for notifications. Subscribers receive all messages published to the topics to which they subscribe, and all subscribers to a topic receive the same messages.

With Amazon SNS, you also have the ability to send push notification messages directly or through topics to apps on mobile devices. Push notification messages sent to a mobile endpoint can appear in the mobile app as message alerts, badge updates, or even sound alerts.

If SNS is unable to deliver messages to an endpoint - for instance, if you install a sports app and enable push notifications, but the app is unable to send you the latest score of your favorite team - it is bound to impact user experience with that endpoint. To avoid this, administrators must continuously track the messages that SNS sends, check whether these messages are delivered to/consumed by the endpoints, and in the process, swiftly detect a delivery failure. This is made possible by the AWS Simple Notification Service - SNS Test.

By default, this test automatically discovers the topics that are created and monitors the messages published to and delivered by each topic. In the process, the test promptly alerts administrators to a message delivery failure. Additionally, the test also reveals topics that are overloaded with messages and those that are handling messages of large sizes.

Optionally, you can configure this test to report metrics for an app on a mobile device or a push notification platform. This way, you can identify the popular apps using SNS and the push notification service that is popular with SNS. This will also lead you to those apps and platforms to which many messages could not be delivered.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each message queue

First-level descriptor: AWS Region

Second-level descriptor: Queue name

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure

Parameter	Description
	that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
SNS Filter Name	By default, this parameter is set to **TopicName**. This means that by default, this test will report metrics for each topic that is created. If required, you can override this default setting by choosing one of the following options: • **Application** - With Amazon SNS, push notification messages can be sent either directly or through topics to apps on mobile devices. For Amazon SNS to send notification messages to mobile endpoints, the mobile apps and the platforms with which the apps are registered should be registered with the AWS. To have this test report metrics for all the mobile apps that are registered with AWS and to which push notification messages are actively sent, select the **Application** option. • **Platform** - With Amazon SNS, push notification messages can be sent either directly or through topics to apps on mobile devices. For Amazon SNS to send notification messages to mobile endpoints, the mobile apps and the platforms with which the apps are registered should be registered with the AWS. To have this test report metrics for all the platforms that are registered with AWS and to which push notification messages are actively sent, select the **Platform** option. The platforms supported by Amazon SNS are as follows: ◦ Amazon Device Messaging (ADM)

Parameter	Description
	◦ Apple Push Notification Service (APNS) for both iOS and Mac OS X ◦ Baidu Cloud Push (Baidu) ◦ Google Cloud Messaging for Android (GCM) ◦ Microsoft Push Notification Service for Windows Phone (MPNS) ◦ Windows Push Notification Services (WNS)

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total published messages	By default, this measure indicates the total number of messages published to this topic during the last measurement period. If the SNS Filter Name chosen is **Application**, then this measure represents the number of messages published to this app during the last measurement period. If the SNS Filter Name chosen is **Platform**, then this measure represents the number of messages published to all apps registered with this platform during the last measurement period.	Number	A high value of this measure is indicative of high messaging activity on a topic/application/platform. A quick comparison across descriptors will point you to the popular topics, applications, or platforms (as the case may be).
Published message size	By default, this measure indicates the average size of all messages published to this topic during the last measurement period.	KB	With the exception of SMS messages, Amazon SNS messages can contain up to 256 KB of text data, including XML, JSON and unformatted text. Each SMS message on the other hand

Measurement	Description	Measurement Unit	Interpretation
	If the SNS Filter Name chosen is **Application**, then this measure represents the average size of messages published to this app during the last measurement period. If the SNS Filter Name chosen is **Platform**, then this measure represents the average size of messages published to all apps registered with this platform during the last measurement period.		can contain up to 140 bytes. If you publish a message that exceeds the size limit, Amazon SNS sends it as multiple messages, each fitting within the size limit. Messages are not cut off in the middle of a word but on whole-word boundaries.
Successfully delivered messages	By default, this measure indicates the total number of messages that were successfully consumed by subscribers to this topic, during the last measurement period. If the SNS Filter Name chosen is **Application**, then this measure represents the number of messages that were successfully delivered to this app either directly or through topics, during the last measurement period. If the SNS Filter Name chosen is **Platform**, then this measure represents the number of messages successfully delivered to	Number	Ideally, the value of this measure should be equal to the value of the 'Total published messages' measure.

Measurement	Description	Measurement Unit	Interpretation
	all apps registered with this platform, during the last measurement period.		
Messages that failed to deliver	By default, this measure indicates the number of messages that subscribers to this topic could not consume, during the last measurement period. If the SNS Filter Name chosen is **Application**, then this measure represents the number of messages that could not be delivered to this app either directly or through topics, during the last measurement period. If the SNS Filter Name chosen is **Platform**, then this measure represents the number of messages that could not be delivered to all apps registered with this platform, during the last measurement period.	Number	Ideally, the value of this measure should be 0. A non-zero value implies that one/more messages could not be delivered to the endpoints. Typically, if Amazon SNS does not receive a successful response from an endpintint, it attempts to deliver the message again. This applies to all messages, including the subscription confirmation message. By default, if the initial delivery of the message fails, Amazon SNS attempts up to three retries with a delay between failed attempts set at 20 seconds. Note that the message request times out at 15 seconds. This means that if the message delivery failure was caused by a timeout, Amazon SNS will retry approximately 35 seconds after the previous delivery attempt. If you do not like the default delivery policy, you can set a different delivery policy on the endpoint. If a delivery to an Amazon SQS, email, SMS, or mobile push endpoint fails, then this measure will disregard all re-delivery attempts that follow. In other words, the value of this measure increases by 1 only if message delivery fails the first time; retries will not impact the value of this measure. On the other hand, if a delivery to an HTTP/HTTPS endpoint fails, then the value of this measure will be incremented by 1 for every subsequent

Measurement	Description	Measurement Unit	Interpretation
			delivery retry as well. To troubleshoot the failure of message deliveries to Application, HTTP, Lambda, and SQS endpoints, you need to enable Amazon SNS delivery status logging. This can be achieved using the AWS Management Console, AWS SDKs, or the AWS CLI.
Successfully delivered SMS	By default, this measure indicates the total number of successful SMS message deliveries made to subscribers of this topic, during the last measurement period. If the SNS Filter Name chosen is **Application**, then this measure represents the number of successful SMS message deliveries made to this app during the last measurement period. If the SNS Filter Name chosen is **Platform**, then this measure represents the number of successful SMS message deliveries made to all apps registered with this platform, during the last measurement period.	Number	Ideally, the value of this measure should be high. A very low value indicates many SMS delivery failures.

4.7.22 AWS Simple Queue Service - SQS Test

Amazon Simple Queue Service (Amazon SQS) offers a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. It moves data between

distributed application components and helps you decouple these components.

The following illustration describes the lifecycle of an Amazon SQS message, from creation to deletion.

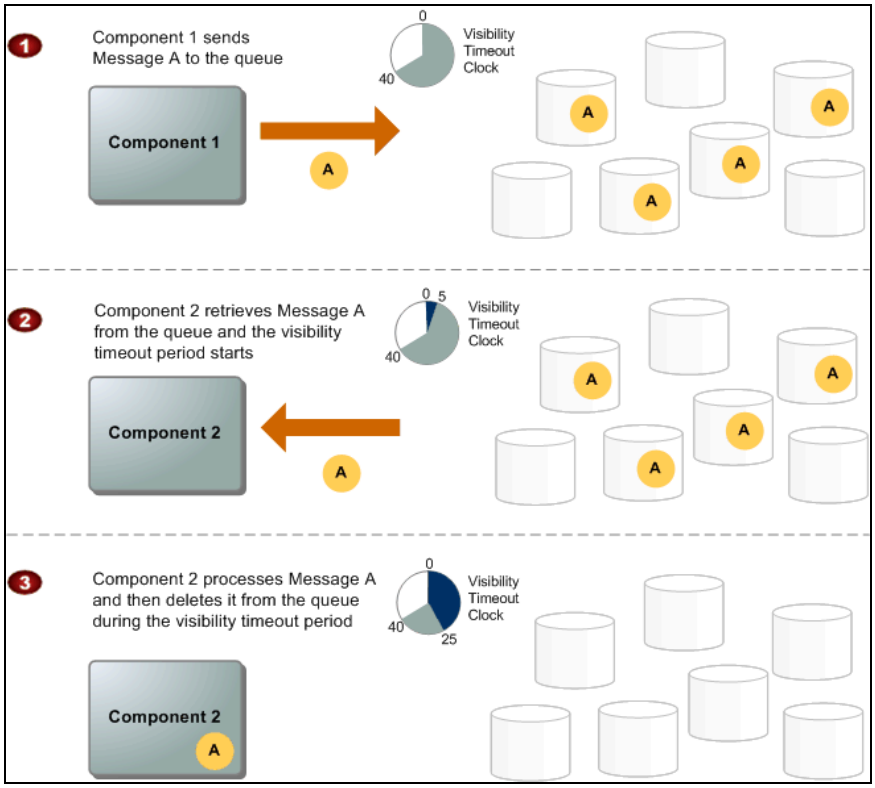


Figure 4.23: A typical message cycle

1. Component 1 sends Message A to a queue, and the message is distributed across the Amazon SQS servers redundantly.
2. When Component 2 is ready to process a message, it consumes messages from the queue, and Message A is returned. While Message A is being processed, it remains in the queue and is not returned to subsequent receive requests for the duration of the visibility timeout.
3. Component 2 deletes Message A from the queue to prevent the message from being received and processed again once the visibility timeout expires.

To ensure the optimal performance of the SQS, administrators should keep an eye on the flow of messages into and out of each queue and proactively capture potential anomalies. For instance, administrators should track the length of each queue, so that they can capture an Overlimit error before it occurs. Likewise, administrators should check whether the size of messages in and the count of empty messages returned by the queues are well-within the prescribed limits. Besides

detecting probable problems, administrators should also able to rapidly diagnose the root-cause of the problems - is it because of an improper configuration? or is it because of an unexpected/unplanned overload? Based on this diagnosis, administrators should then initiate measures to eliminate the root-cause so as to prevent the problems. The AWS Simple Queue Service - SQS Test helps with the quick detection of problems in SQS, accurate diagnosis of their root-cause, and thus facilitates their prompt redressal!

This test auto-discovers the queues that have been created and reports the count of messages sent to, inflight, and received from each queue. In the process, the test points to queues that are about to violate their message limit. By reporting the number of messages delayed, deleted messages, and the age of the oldest undeleted message in each queue, the test provides useful pointers to what can be done to maximize overall queue performance and avoid errors - should the visibility period be changed? should the retention period of messages be altered? should more queues be created to handle the load? The test also monitors the average size of messages in each queue, and turns the spotlight on those queues that contain many messages that violate the prescribed size limit. This way, the test prompts administrators to look for alternative means to send large messages, so that the load on queues can be reduced and message processing is faster. Additionally, the test also pinpoints queues that returned many empty messages, prompting administrators to rethink their choice of polling mechanism (long or short).

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each message queue

First-level descriptor: AWS Region

Second-level descriptor: Queue name

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and

Parameter	Description
Key	secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Messages send	Indicates the number of messages added to this queue	Number	A high value of this measure is indicative of high messaging activity between application components.
Size of messages added to the queue	Indicates the average size of messages in this queue	KB	SQS supports messages with a minimum size of 1 byte and a maximum size of 262,144 bytes (256 KB). If the value of this measure is consistently higher than 256 KB, it means that many messages in the queue are more than 256 KB in size. To reduce the load on the queue, you can

Measurement	Description	Measurement Unit	Interpretation
			send messages larger than 256 KB using the Amazon SQS Extended Client Library for Java. This library allows you to send an Amazon SQS message that contains a reference to a message payload in Amazon S3. The maximum payload size is 2 GB.
Messages received	Indicates the number of messages that this queue returned in response to ReceiveMessage API calls.	Number	
Empty messages received	Indicates the number of ReceiveMessage API calls for which this queue did not return a message.	Number	A low value is desired for this measure. This is because, empty responses and false empty responses increase the cost of using Amazon SQS. If the value of this measure is very high, you may want to consider employing Long Polling. Long polling helps reduce your cost of using Amazon SQS by reducing the number of empty responses (when there are no messages available to return in reply to a ReceiveMessage request sent to an Amazon SQS queue) and eliminating false empty responses (when messages are available in the queue but aren't included in the response): • Long polling reduces the number of empty responses by allowing Amazon SQS to wait until a message is available in the queue before sending a response. Unless the connection times out, the

Measurement	Description	Measurement Unit	Interpretation
			response to the ReceiveMessage request contains at least one of the available messages, up to the maximum number of messages specified in the ReceiveMessage action. • Long polling eliminates false empty responses by querying all (rather than a limited number) of the servers. • Long polling returns messages as soon any message becomes available.
Messages deleted	Indicates the number of messages deleted from this queue.	Number	A high value is desired for this measure. If too few messages are deleted, then it may increase the count of inflight messages in the queue. If the number of inflight messages in standard queues is allowed to grow beyond a prescribed limit, an OverLimit error will occur. To avoid this, you may have to do any of the following: • Change the message retention period: Amazon SQS automatically deletes messages that have been in a queue for more than maximum message retention period. The default message retention period is 4 days. However, if required, you can set the message retention period to a value from 60 seconds to 1,209,600 seconds

Measurement	Description	Measurement Unit	Interpretation
			(14 days) using the SetQueueAttributes action. • Change the visibility timeout: If you do not want to override the default message retention period, you can change the visibility timeout period of the queue or of individual messages. When a consumer receives and processes a message from a queue, the message remains in the queue. To prevent other consumers from processing the message again, Amazon SQS sets a visibility timeout. The visibility timeout begins when Amazon SQS returns a message. During this time, the consumer processes and deletes the message. However, if the consumer fails before deleting the message and your system doesn't call the DeleteMessage action for that message before the visibility timeout expires, the message becomes visible to other consumers and the message is received again. If a message must be received only once, your consumer should delete it within the duration of the visibility timeout. Every Amazon SQS queue has the default visibility timeout setting of 30 seconds. You can change this setting for the entire

Measurement	Description	Measurement Unit	Interpretation
			queue. Typically, you should set the visibility timeout to the average time it takes to process and delete a message from the queue. When receiving messages, you can also set a special visibility timeout for the returned messages without changing the overall queue timeout. If you don't know how long it takes to process a message, specify the initial visibility timeout (for example, 2 minutes) and the period of time after which you can check whether the message is processed (for example, 1 minute). If the message isn't processed, extend the visibility timeout (for example, to 3 minutes).
Messages delayed	Indicates the number of messages in this queue that are delayed and not available for reading immediately.	Number	This can happen when the queue is configured as a delay queue or when a message has been sent with a delay parameter. Delay queues let you postpone the delivery of new messages in a queue for the specified number of seconds. If you create a delay queue, any message that you send to that queue is invisible to consumers for the duration of the delay period. You can use the CreateQueue action to create a delay queue by setting the DelaySeconds attribute to any value between 0 and 900 (15 minutes). You can also change an existing queue into a delay queue using the SetQueueAttributes action to set

Measurement	Description	Measurement Unit	Interpretation
			the queue's DelaySeconds attribute. Instead of an entire queue, you can delay specific messages in a queue. Amazon SQS message timers allow you to specify an initial invisibility period for a message that you add to a queue. For example, if you send a message with the DelaySeconds parameter set to 45, the message is not visible to consumers for the first 45 seconds during which the message stays in the queue. The default value for DelaySeconds is 0. **Note that FIFO queues do not support timers on individual messages.** Messages with message timers and messages in delay queues may take longer to be processed owing to the delay factor.
Messages available for retrieval	Indicates the number of messages available for retrieval from this queue.	Number	
Messages that are inflight	Indicates the number of messages inflight in this queue.	Number	Messages are considered in flight if they have been sent to a client but have not yet been deleted or have not yet reached the end of their visibility window. For standard queues, there can be a maximum of 120,000 inflight messages per queue. If you reach this limit, Amazon SQS returns the OverLimit error message. To avoid reaching the limit, you should delete messages from the queue after they are processed. You can also increase the number of queues you use to process your messages. For FIFO queues, there can be a

Measurement	Description	Measurement Unit	Interpretation
			maximum of 20,000 inflight messages per queue. If you reach this limit, Amazon SQS returns no error messages.
Age of oldest non-deleted messages	Indicates the approximate age of the oldest non-deleted message in this queue.	Secs	A very high value for this measure can indicate that one or more messages are not getting deleted as frequently as desired. To ensure timely processing and deletion of messages, prudently set the visibility timeout period and the message retention period.

4.7.23 AWS Service Usage Test

Use this test to receive an overview of the instances launched and services (EBS and RDS) used by a configured AWS user account in a monitored region. Understanding how many instances of a service are utilized by an account will help you to bill that user accordingly.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for the AWS account configured for monitoring

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy	In some environments, all communication with the AWS EC2 cloud and its

Parameter	Description
Port	regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
EC2 instances:	Indicates the total number of EC2 instances currently available for the configured AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know which are the available instances.
EC2 instances poweredon:	Indicates the total number of instances that are currently powered-on for the configured AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know the names of the powered-on instances.
EBS volumes:	Indicates the total number of EBS volumes currently available for the configured AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know which volumes are available for use presently.
RDS instances:	Indicates the total number of RDS instances that are configured for the AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know the details of all RDS instances configured for the AWS user account.
RDS instances available:	Indicates the number of instances that are currently powered-on and available for the use of the configured AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know the details of all RDS instances that are powered-on and available for the use of the AWS user account.
S3 buckets:	Indicates the total number of S3 buckets available in for of the	Number	Use the detailed diagnosis of this measure to know the bucket names and when they were created.

Measurement	Description	Measurement Unit	Interpretation
	configured AWS user account in the monitored region.		

The detailed diagnosis of the **EC2 instances** measure lists the names of all EC2 instances available for the configured AWS user account.

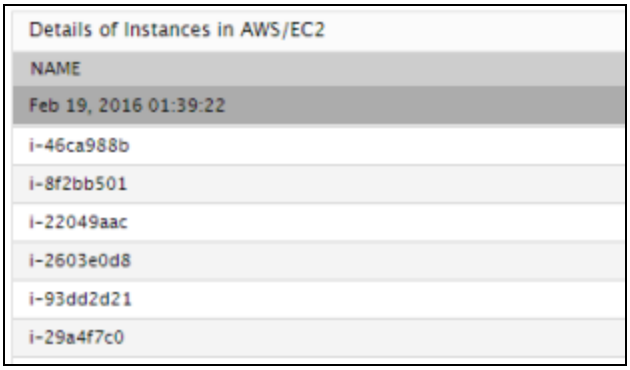


Figure 4.24: The detailed diagnosis of the EC2 instances measure

The detailed diagnosis of the **EC2 instances poweredon** measure lists the names of the powered-on instances alone.

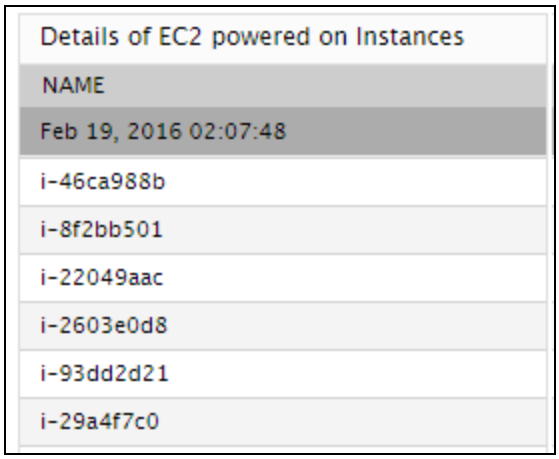


Figure 4.25: The detailed diagnosis of the EC2 instances poweredon measure

The detailed diagnosis of the **EBS volumes** measure displays the names of volumes that are available for use presently.

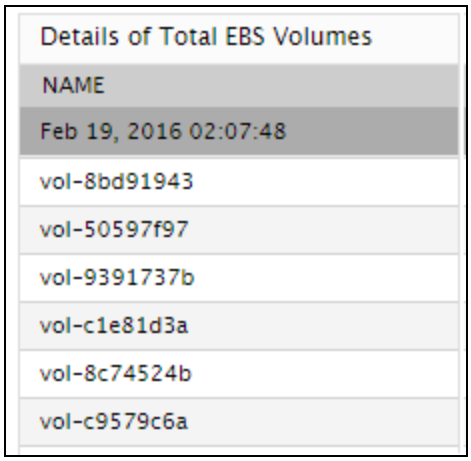


Figure 4.26: The detailed diagnosis of the EBS volumes measure

The detailed diagnosis of the **RDS instances** measure provides the details of all RDS instances configured for the AWS user account.

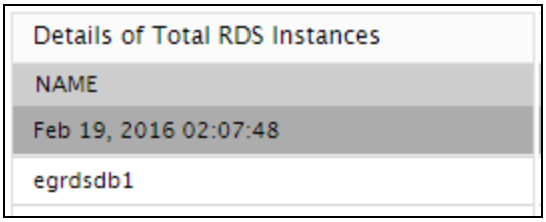


Figure 4.27: The detailed diagnosis of the RDS instances measure

The detailed diagnosis of the **RDS instances available** measure displays the details of the powered-on instances alone.

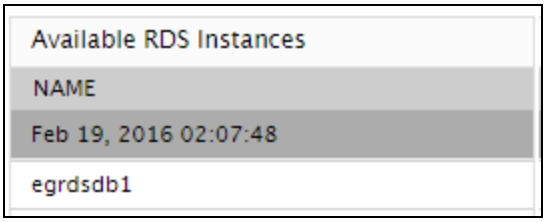


Figure 4.28: The detailed diagnosis of the RDS instances available measure

4.8 AWS Workspaces - Directory Test

An Amazon WorkSpace is a cloud-based virtual desktop that can act as a replacement for a traditional desktop. A WorkSpace is available as a bundle of compute resources, storage space, and software applications that allow a user to perform day-to-day tasks just like using a traditional

desktop. A user can connect to a WorkSpace from any supported device using the free Amazon WorkSpaces client application, or using Chrome or Firefox web browsers. Users will connect using credentials set up by an administrator, or using their existing Active Directory credentials if you've chosen to integrate your Amazon WorkSpaces with an existing Active Directory domain. Once the user is connected to a WorkSpace they can perform all the usual tasks they would do on a desktop computer.

If a user is unable to connect to a Workspace or is experiencing considerable slowness when attempting to do so, that user will not be able to use the cloud resources effectively; this in turn will impact user productivity, hit revenues, and increase support costs and penalties.

To avoid this, administrators should be able to detect and resolve the unavailability/inaccessibility of Workspaces, and latencies in connecting to Workspaces, well before users complain. This is where the AWS Workspaces - Directory test helps!

This test automatically discovers the directories configured on the AWS Cloud for storing and managing information for Workspaces and users. For each directory so discovered, the test then reports the count of Workspaces in that directory in different states of activity - eg., available, unavailable, stopped, etc. This way, the test proactively alerts administrators to the abnormal state of Workspaces in a directory. In addition, the test also tracks connection attempts to the Workspaces in each directory, reports connection failures, and thus brings connection issues to the notice of administrators. The average time taken to launch sessions on the Workspaces in every directory is also reported, so that administrators can identify which directory is managing the most latent Workspaces.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each directory on the AWS EC2 cloud.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key,	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this

Parameter	Description
Confirm AWS Secret Key	has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Available workspaces	Indicates the number of workspaces in this directory that returned a healthy status during the last measurement period.	Number	Amazon WorkSpaces periodically sends status requests to a WorkSpace. A WorkSpace is marked Available when it responds to these requests, and Unhealthy when it fails to respond to these requests. Ideally, the value of this measure should be high.
Unhealthy workspaces	Indicates the number of WorkSpaces that returned	Number	Amazon WorkSpaces periodically sends status requests to a

Measurement	Description	Measurement Unit	Interpretation
	an unhealthy status during the last measure period.		WorkSpace. A WorkSpace is marked Available when it responds to these requests, and Unhealthy when it fails to respond to these requests. Ideally, the value of this measure should be low.
Connection attempt	Indicates the number of connection attempts made to workspaces in this directory during the last measure period.	Number	
Connection success	Indicates the number of connection attempts made to workspaces in this directory that were successful during the last measure period.	Number	Ideally, the value of this measure should be equal to the value of the 'Connection attempt' measure. If the difference between the two measures is very high, it implies that many connections have failed.
Connection failure	Indicates the number of connection attempts to the workspaces in this directory that failed during the last measurement period.	Number	Ideally, the value of this measure should be 0. A value close to or equal to the value of the Connection attempt measure denotes many failed connection attempts, which is a cause for concern.
Session latency	Indicates the average round trip time between the WorkSpaces client and the WorkSpaces in this directory during the measure period.	Secs	This measure is reported after a user has successfully authenticated via the WorkSpaces client and the client then initiates a session. Any delay in authentication and/or session initiation can cause the value of this measure to increase. If you notice abnormal spikes in the value of this measure, then you may want to compare the value of this measure with that of the Session launch time measure to know where the latency is maximum - at the time of authentication? or during session initiation?

Measurement	Description	Measurement Unit	Interpretation
Session launch time	Indicates the average time for initiating sessions with the Workspaces in this directory.	Secs	If you notice abnormal spikes in the value of this measure, then you may want to compare the value of this measure with that of the Session latency measure to know where the latency is maximum - at the time of authentication? or during session initiation?
Session disconnect	Indicates the number of connections with Workspaces in this directory that were closed during the last measure period.	Number	The value of this measure also includes user-initiated and failed connections. Ideally, the value of this measure should be 0.
User connected workspaces	Indicates the number of Workspaces in this directory that had a user connected during the last measurement period.	Number	Amazon WorkSpaces periodically sends connection status requests to a WorkSpace. Users are reported as connected when they are actively using their sessions.
Stopped workspaces	Indicates the number of Workspaces in this directory that stopped during the last measurement period.	Number	
Workspaces in maintenance	Indicates the number of Workspaces in this directory that were under maintenance during the last measurement period.	Number	This metric applies to WorkSpaces that are configured with an AutoStop running mode. With this mode, your WorkSpaces stop after a specified period of inactivity and the state of apps and data is saved. Amazon WorkSpaces schedules maintenance for your WorkSpaces. During the maintenance window, important updates are downloaded and installed. If you enable maintenance mode for your AutoStop WorkSpaces, they are started automatically once a

Measurement	Description	Measurement Unit	Interpretation
			month in order to download and install important service, security, and Windows updates.

4.8.1 AWS Web Application Firewall - WAF Test

AWS WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.

AWS WAF gives you control over which traffic to allow or block to your web applications by defining customizable web security rules. A rule identifies the requests that you want to allow, block, or count. You can add one or more rules to a WebACL, and associate each rule with an action (allow/block/count) - for example, block requests from specified IP addresses or block requests from specified referrers. You also need to specify a default action for a WebACL. You can then associate the WebACL with an Amazon CloudFront distribution or an Application Load Balancer (ALB) - services that AWS customers commonly use to deliver content for their websites and applications. These services receive requests for your web sites and forwards those requests to AWS WAF for inspection against the rules configured in the WebACL. If you add more than one rule to a WebACL, a request needs to match only one of the specifications to be allowed, blocked, or counted. Once a request meets one of the conditions defined in your rules, AWS WAF instructs the underlying service to either block or allow the request based on the action you define.

Periodically, administrators must track the requests allowed and/or blocked to understand whether/not your web aplications/sites are well-protected against malicious attacks. In the process, administrators can isolate ineffective or incorrectly configured rules/WebACLs and the security threats they pose. Administrators can then proceed to fine-tune these rules/WebACLs, so that their mission-critical applications are more secure. This is where, the AWS WAF Test helps!

By default, this test automatically discovers the rules configured in the AWS Web Application Firewall. For each rule, the test reports the count of requests that fulfill at least one of the specifications of that rule and that have been allowed and/or blocked as per that rule. This will enable administrators to figure out how many requests are allowed and/or blocked by each rule, and in the process, identify those rules that may have been configured incorrectly (eg., rules that were defined to block certain requests, but are allowing them), and/or poorly (eg., rules that are blocking less requests than they should). Such rules are candidates for deletion or fine-tuning.

You can optionally configure this test to report metrics for each WebACL. By comparing the measures reported by this test across WebACLs, administrators can rapidly identify WebACLs that may have to be reconfigured.

Target of the test: Amazon EC2 Cloud

Agent deploying the test : A remote agent

Outputs of the test : One set of results for each rule / WebACL, depending upon the option chosen from the **WAF Filter Name** parameter

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the Proxy Host and Proxy Port parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the Proxy User Name and Proxy Password parameters, respectively. Then, confirm the password by retyping it in the Confirm Password text box. By default, these parameters are set to *none*, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the Proxy Domain and Proxy Workstation parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region

Parameter	Description
	names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
WAF Filter Name	By default, this parameter is set to **Rule**. In this case therefore, the test will report metrics for each rule that is configured. To override this default setting, you can pick the **WebACL** option from this drop-down. In this case, this test will report metrics per WebACL.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Allowed web requests	By default, this measure represents the number of web requests that this rule allowed. If you set the WAF Filter Name parameter to **WebACL**, then this measure represents the total number of web requests that all rules in this WebACL allowed.	Number	If a request fulfills at least one specification of a rule/WebACL and is allowed as per that specification, then such a request is counted as an 'Allowed web request'. If a rule/WebACL allows more requests than it should, then you can take that rule/WebACL up for closer scrutiny and make changes to that rule/WebACL (if required).
Blocked web requests	By default, this measure represents the number of web requests that this rule blocked. If you set the WAF Filter Name parameter to **WebACL**, then this measure represents the total number of web requests that all rules in this WebACL blocked.	Number	If a request fulfills at least one specification of a rule/WebACL and is blocked as per that specification, then such a request is counted as a 'Blocked web request'. If a rule/WebACL blocks less/more requests than it should, then you can take that rule/WebACL up for closer scrutiny and make changes to that rule/WebACL (if required).
Counted web requests	By default, this measure represents the number of web requests that fulfill all specifications of this rule.	Number	

Measurement	Description	Measurement Unit	Interpretation
	If you set the WAF Filter Name parameter to **WebACL**, then this measure represents the total number of web requests that fulfill all specifications of all the rules in this WebACL.		

Chapter 5: Administering the eG Manager to Monitor the AWS EC2 Region

To achieve this, follow the steps given below:

1. Log into the eG administrative interface.
2. eG Enterprise automatically discovers the AWS EC2 Regions once you set the ***Discover AWS EC2 cloud regions*** parameter to ***Yes*** while discovering the AWS EC2 Cloud. If the AWS EC2 Region is already discovered, use the Infrastructure -> Components ->Manage/Unmanage menu to manage it.
3. To manage the discovered components, go to the Infrastructure -> Components -> Manage/Unmanage page. The process of managing a component is clearly depicted by Figure 5.1 below.

   Note:

   For a more detailed procedure for managing components, refer to **Configuring and Monitoring Web Servers** document.

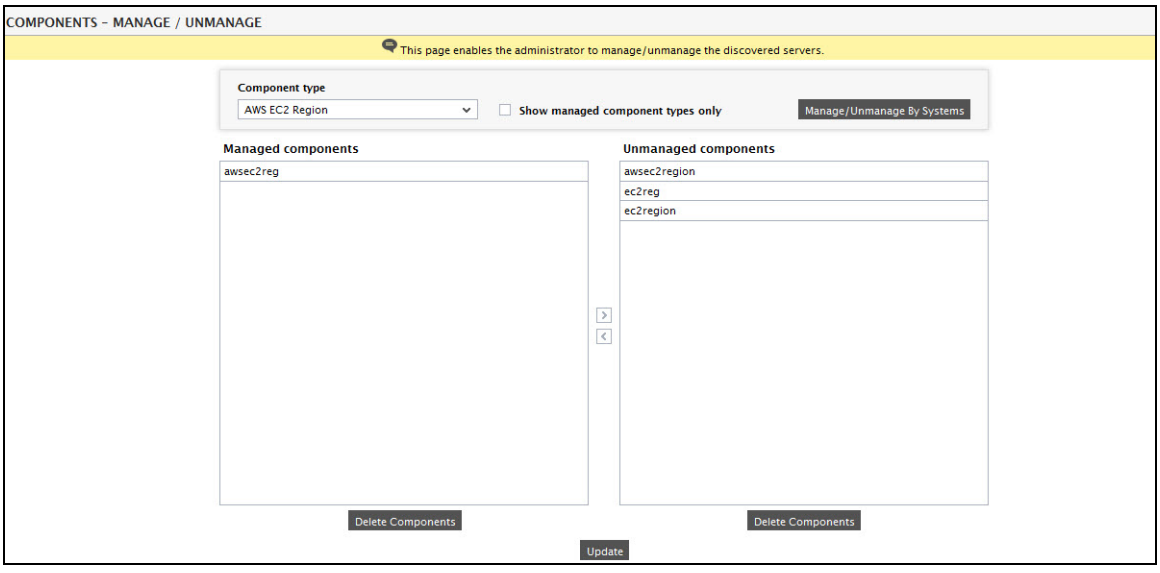


Figure 5.1: Managing an AWS EC2 Region

4. If you wish to manually add the Amazon EC2 Region component using the menu sequence: Infrastructure -> Components -> Add/Modify, ensure that you provide a valid name in the HOST text box instead of an IP address. In order to provide a valid name, ensure that the

AllowQualifiedHostnames flag is set to **yes** in the **eg_services.ini** file of the **<EG_INSTALL_DIR>\manager\config** directory. Remember that components manually added are managed automatically.

Note:

The **HOST** name must be provided in the following format: **AWS EC2 Cloud/Region name**. In this case, the **HOST** name must be given as **awsec2cld/ap-southeast-1**

5. Now, when you attempt to sign out of the eG administrative interface, Figure 5.2 appears, listing the tests requiring manual configuration.

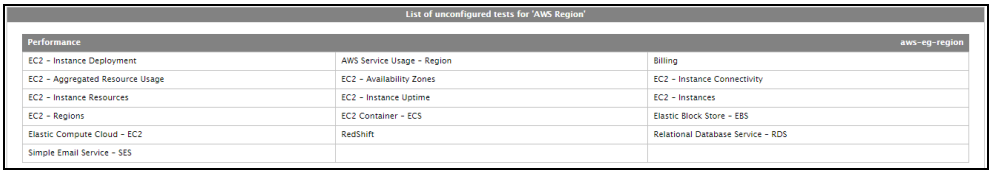


Figure 5.2: The list of unconfigured tests for AWS EC2 Region

6. Click on the **EC2 - Region** test to configure it. This test reports the availability of the Region and enables the administrator to figure out the time taken by the Region to respond to responses. To know how to configure the test, Click Here.
7. Finally, sign out of the eG administrative interface.

Chapter 6: Monitoring the AWS EC2 Region

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and Regions. Regions are dispersed and located in separate geographic areas (US, EU, etc.). Availability Zones are distinct locations within a Region that are engineered to be isolated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same Region. By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements.

The AWS EC2 Region model offered by eG Enterprise monitors a specific region on the cloud and reports the availability and responsiveness of that region.

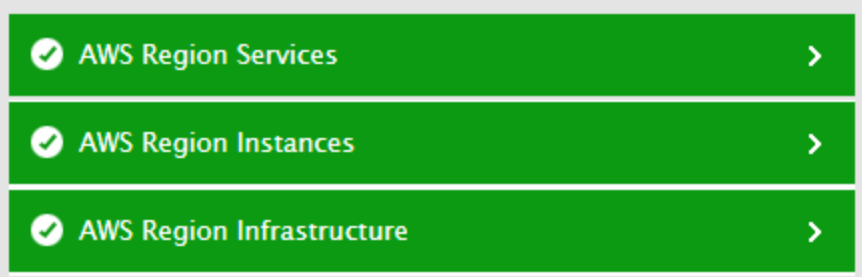


Figure 6.1: The layer model of the AWS EC2 Region

In addition, using a single eG agent installed on a remote Windows host in the environment, the model auto-discovers the IP address and the operating system of the instances launched on the cloud, periodically checks the powered-on status of each of the instances, continously assesses how each instance is utilizing the allocated resources, and thus promptly alerts you to unavailable and resource-hungry instances. As the solution also automatically determines what applications have been deployed on the instances, whenever one of these applications experience slowdowns, administrators can use the eG solution to instantly and accurately diagnose the root-cause of the slowdown - is it owing to the corresponding instance being unavailable or the application being resource-hungry?

Using the metrics so reported, administrators can ascertain the following:

- Is web-based (HTTP/HTTPS) access to the region available?
- Does it take an unreasonably long time to establish contact with the region?
- How many availability zones exist in the monitored region? What are they?

- Is any availability zone currently unavailable? If so, which one is it?
- Are all instances launched in the region accessible over the network?

Are any instances powered off currently?

- Were any instances launched/removed recently? If so, which ones are these?
- What type of instances are resource-intensive?
- Is any particular instance consuming too much CPU?
- Is the network traffic to/from any instance unusually high?
- Is the disk I/O of instances optimal?
- Was any instance rebooted recently? If so, which one is it?

Note:

The eG agent reports metrics for only availability zones and instances in a region that the configured AWS user account is allowed to access.

Some tests require the **AWS CloudWatch** service to be enabled. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. For enabling this service, you need to pay CloudWatch fees. Refer to the AWS web site for the fee details.

The sections that follow will discuss each layer of Figure 6.1 elaborately.

6.1 The AWS Region Infrastructure Layer

Using the tests mapped to this layer, you can promptly detect the non-availability of a target region and the availability zones in that region, and connection bottlenecks experienced while connecting to the cloud or its components.

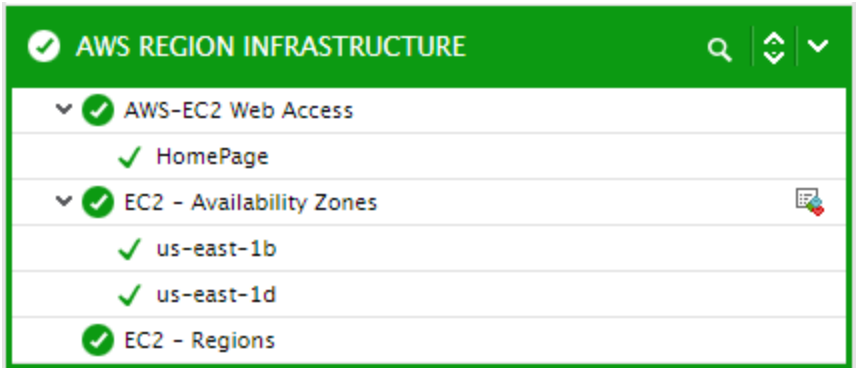


Figure 6.2: The tests mapped to the AWS Region Infrastructure layer

6.1.1 AWS - EC2 Web Access Test

This test emulates a user accessing a web page on the cloud via HTTP(S), and reports whether that page is accessible or not. In the process, the test indicates the availability of the cloud over the web, and the time it took for the agent to access the cloud over the web. This way, issues in web-based access to the cloud come to light.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for every URL configured for monitoring

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured
Port	The port to which the specified **HOST** listens
URL	The web page being accessed. While multiple URLs (separated by commas) can be provided, each URL should be of the format URL name:URL value. URL name is a unique name assigned to the URL, and the URL value is the value of the URL. By default, the url parameter is set to *HomePage:http://aws.amazon.com/ec2/*, where *HomePage* is the *URL name*, and *http://aws.amazon.com/ec2* is the *URL value*. You can modify this default setting to configure any URL of your choice - eg., the URL of the login page to your cloud-based infrastructure.
Cookie File	Whether any cookies being returned by the web server need to be saved locally and returned with subsequent requests
Proxy Host and Proxy Port	The host on which a web proxy server is running (in case a proxy server is to be used), and the port at which the web proxy server listens
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box.
Content	Is a set of instruction:value pairs that are used to validate the content being

Parameter	Description
	returned by the test. If the **CONTENT** value is *none:none*, no validation is performed. The number of pairs specified in this text box, must be equal to the number of URLs being monitored. The instruction should be one of *Inc* or *Exc*. Inc tells the test that for the content returned by the test to be valid, the content must include the specified value (a simple string search is done in this case). An instruction of *Exc* instructs the test that the test's output is valid if it does not contain the specified value. In both cases, the content specification can include wild card patterns. For example, an Inc instruction can be *Inc:*Home page**. An *Inc* and an *Exc* instruction can be provided in quick succession in the following format: *Inc:*Home Page*,Exc:*home*.
Credentials	The HttpTest supports HTTP authentication. The **CREDENTIALS** parameter is to be set if a specific user name / password has to be specified to login to a page. Against this parameter, the *URLname* of every configured URL will be displayed; corresponding to each listed *URLname*, a **Username** text box and a **Password** text box will be made available. These parameters will take either of the following values: • valid **Username** and **Password** for every configured *URLname* • *none* in both the **Username** and **Password** text boxes of all configured *URLnames* (the default setting), if no user authorization is required Where NTLM (Integrated Windows) authentication is supported, valid **CREDENTIALS** are mandatory. In other words, a none specification will not be supported in such cases. Therefore, in this case, against each configured *URLname*, you will have to provide a valid Username in the format: *domainname\username*, followed by a valid **Password**. Please be sure to check if your web site requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many sites use HTTP POST for obtaining the user name and password and validating the user login. In such cases, the username and password have to be provided as part of the POST information and NOT as part of the **CREDENTIALS** specification for the this test.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows

Parameter	Description
	NTLM proxy, set these parameters to *none*.
Timeout	Here, specify the maximum duration (in seconds) for which the test will wait for a response from the server. The default **TIMEOUT** period is 30 seconds.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Availability:	Indicates whether the test was able to access the configured URL or not	Percent	Availability failures could be caused by several factors such as the web server process(es) (hosting the configured web page) being down, the web server being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web server is overloaded. Availability is determined based on the response code returned by the test. A response code between 200 to 300 indicates that the configured web page is available.
Total response time:	Indicates the time taken by the test to access this URL	Secs	Response time being high denotes a problem. Poor response times may be due to an overload. If the URL accessed involves the generation of dynamic content, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.
Tcp connection availability:	Indicates whether the test managed to establish a TCP connection to this URL.	Percent	Failure to establish a TCP connection may imply that either the web server process hosting the web page is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the web page may start functioning properly again.
Tcp connect time:	Quantifies the time for establishing a TCP connection to the configured URL.	Secs	Typically, the TCP connection establishment must be very small (of the order of a few milliseconds).

Measurement	Description	Measurement Unit	Interpretation
Server response time:	Indicates the time period between when the connection was established and when the test sent back a HTTP response header to the client.	Secs	While the total response time may depend on several factors, the server response time is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
Response code:	Returned by the test for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
Content length:	The size of the content returned by the test.	Kbytes	Typically the content length returned by the test for a specific URL should be the same across time. Any change in this metric may indicate the need for further investigation.
Content validity:	Validates whether the test was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content may not be valid. This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, "Invalid Login" is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is configured such that the content parameter should exclude the string "Invalid Login," in the above scenario content validity would have a value 0.

6.1.2 EC2 - Availability Zones Test

Amazon has data centers in different areas of the world (e.g., North America, Europe, Asia, etc.). Correspondingly, EC2 is available to use in different *Regions*. Each Region contains multiple distinct locations called *Availability Zones* (illustrated in the following diagram). Each Availability Zone is

engineered to be isolated from failures in other Availability zones and to provide inexpensive, low-latency network connectivity to other zones in the same Region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.

If users complaint that their server instances are inaccessible, you may want to know whether it is because of the non-availability of the availability zone within which the instances have been launched. This test auto-discovers the availability zones configured within the monitored EC2 region, and reports the availability of each zone.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for each availability zone in the AWS EC2 Region being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none,

Parameter	Description
	indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Cloudwatch Enabled	**This parameter only applies to the EC2 -Aggregated Resource Usage test**. This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.
Report Instance Datacenter	By default, this test reports the availability of only those availability zones that contain one/more instances. Accordingly, this flag is set to **Yes** by default. If you want the test to report metrics for all availability zones, regardless of whether/not they host instances, set this flag to **No**.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Availability:	Indicates whether/not this availability zone is currently available.	Number	The value 0 indicates that the availability zone is Not Available and the value 100 indicates that it is Available. If an availability zone fails, then all server instances operating within that zone will

Measurement	Description	Measurement Unit	Interpretation
			also be rendered unavailable. If you host all your Amazon EC2 instances in a single location that is affected by such a failure, your instances will be unavailable, thereby bringing your entire application to a halt. On the other hand, if you have instances distributed across many Availability Zones and one of the instances fails, you can design your application so the instances in the remaining Availability Zones handle any requests.

6.1.3 EC2 - Regions Test

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and Regions. Regions are dispersed and located in separate geographic areas (US, EU, etc.). Each Region is completely independent.

By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements.

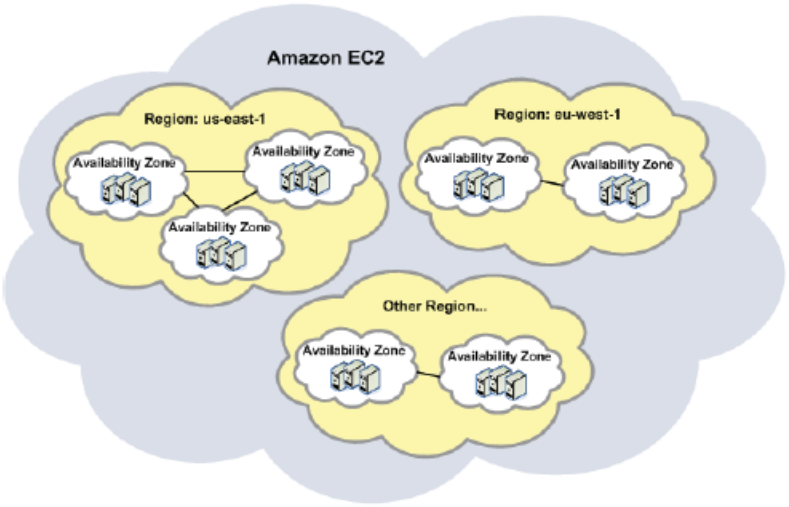


Figure 6.3: Regions and Availability zones

If a region is unavailable, then users to that region will not be able to access the server instances launched in that region. This may, in turn, adversely impact the user experience with the cloud. To avoid such an unpleasant outcome, it is best to periodically monitor the availability of each region, so that unavailable regions can be quickly and accurately identified, and the reasons for their non-availability remedied.

This test performs periodic availability checks on the monitored region, and reports the status of that region. In addition, the test also indicates the time taken for connecting to the region so that, connectivity issues can be isolated.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for the AWS EC2 region being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none,

Parameter	Description
	indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Cloudwatch Enabled	**This parameter is applicable only to the EC2-Aggregated Resource Usage test.** This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Availability:	Indicates whether/not the region is currently available.	Number	The value 0 indicates that the region is Not Available and the value 100 indicates that it is Available.
Response time:	Indicates the time taken to connect to the region.	Secs	A low value is typically desired for this measure. A high value or a consistent increase in this value could be indicative of connection bottlenecks.

6.2 The AWS Region Instances Layer

To determine issues in accessibility server instances launched in a region, and to detect the current state of each instance, use the tests mapped to this layer. The tests also auto-discover the server instances that are available (for the configured AWS user account) in a region, and report the uptime and the resource usage of the individual instances. Resource-hungry instances and those that were recently rebooted can thus be isolated.

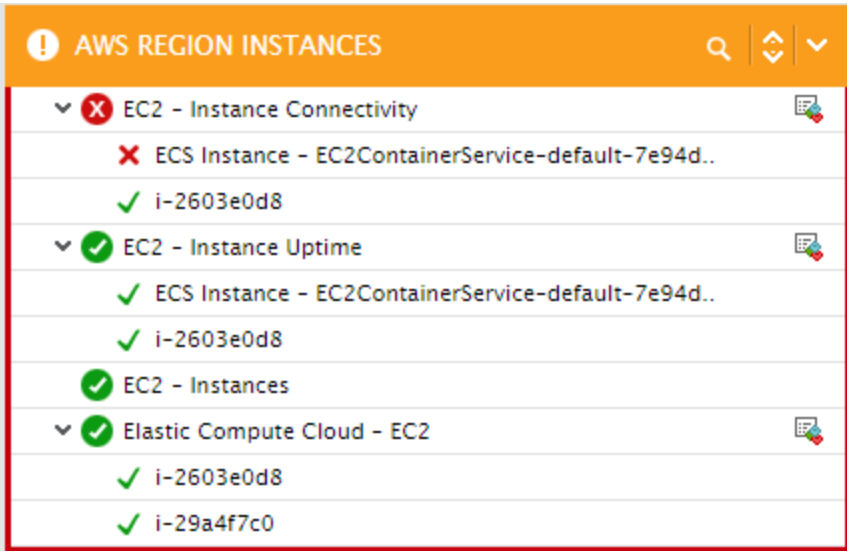


Figure 6.4: The tests mapped to the AWS EC2 Region Instance Status layer

6.2.1 Elastic Compute Cloud - EC2 Test

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. Since users may run mission-critical applications on these EC2 instances, high uptime of the EC2 instances is imperative to the uninterrupted functioning of these applications and to ensure 100% user satisfaction with this cloud-based service. AWS administrators therefore, should frequently perform health checks on every instance, measure its load and resource usage, and capture potential failures and resource contentions, well before end-users notice and complain. This is exactly where the Elastic Compute Cloud - EC2 test helps!

This test monitors the powered-on state of each EC2 instance and promptly alerts administrators if any instance has been powered-off inadvertently. Additionally, the test also reveals how each

instance uses the CPU, disk, and network resources it is configured with, thus providing early pointers to irregularities in instance sizing, and prompting administrators to make necessary amends. This way, the test makes sure that critical applications are always accessible to end-users and perform at peak capacity.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for each instance / auto scaling group / instance type / image ID in the region being monitored, depending upon the option chosen from the **EC2 FILTER NAME** drop-down

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy	If a Windows NTLM proxy is to be configured for use, then additionally, you will

Parameter	Description
Workstation	have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Instance	**This parameter is applicable only if InstanceId is chosen from the EC2 Filter Name drop-down.** In this case, against **EXCLUDE INSTANCE**, you can provide a comma-separated list of instance IDs you do not want the test to monitor.
EC2 Filter Name	By default, this test reports metrics for each instance in the AWS infrastructure. This is why, the **EC2 FILTER NAME** flag is set to *Instance ID* by default. Alternatively, you can configure this test to aggregate metrics across a chosen collection of instances, and report one set of metrics per collection. For this, you just need to pick an instance collection from the EC2 Filter Name drop-down. The options available are as follows: • *AutoScalingGroupName*: Your EC2 instances can be organized into Auto Scaling Groups so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and, desired number of EC2 instances. If you select the *AutoScalingGroupName* option from the **EC2 FILTER NAME** drop-down, then this test will collect metrics for each instance, aggregate the metrics on the basis of the Auto Scaling Groups to which the instances belong, and report metrics for each group. • *InstanceType*: Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. If the *InstanceType* option is chosen from the **EC2 FILTER NAME** drop-down, then this test will collect metrics for each instance, aggregate the metrics on the basis of the instance type, and report metrics for each type. • *ImageId*: Instances are created from Amazon Machine Images (AMI). The machine images are like templates that are configured with an operating system and other software, which determine the user's operating environment. If the *ImageId* option is chosen from the **EC2 FILTER NAME** drop-down, then this test will collect metrics for each instance, aggregate the metrics on the basis of the AMI using which the instances were created, and report metrics for each image ID.

Parameter	Description
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Instance power-on state:</td><td>Indicates the current powered-on state of this instance.</td><td></td><td>This measure is reported only if InstanceID is the option from the EC2 Filter Name drop-down of this test.

The values that this measure can report and their corresponding numeric values are detailed in the table below:

<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td>Running</td><td>When the instance is ready for you, it enters the running state.</td><td>1</td></tr>
<tr><td>Pending</td><td>When you launch an</td><td>2</td></tr>
</table>
</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation		
				instance, it enters the pending state	
			Terminated	When you no longer need an instance, you can terminate it, then it goes to terminated state.	3
			Shutting down	While terminate the instance, As soon as the status of an instance changes to shutting-down or terminated	4
			Stopping	When you stop your instance, it enters the stopping state	5
			Stopped	After exiting	0

Measurement	Description	Measurement Unit	Interpretation
			<table><tr><td></td><td>the stopping state, it enters the stopped state</td><td></td></tr></table> **Note:** By default, this measure will report the **Measure Value**s listed in the table above to indicate the current powered-on state of an instance. In the graph of this measure however, the same will be represented using the numeric equivalents only.
EBS volumes	Indicates the number of EBS volumes attached to this instance.	Number	This measure is reported only if the InstanceId option is chosen from the EC2 Filter Name drop-down of this test. You can attach an EBS volumes to one of your instances that is in the same Availability Zone as the volume. You can attach multiple volumes to the same instance within the limits specified by your AWS account. Your account has a limit on the number of EBS volumes that you can use, and the total storage available to you. Using the detailed diagnosis of this measure, you can identify the volumes that are attached to this EC2 instance.
CPU credit usage:	Indicates the number of CPU credits consumed by this T2 instance / all T2 instances / all T2 instances created from this image ID during the	Number	This measure is reported only for individual T2 instances, the T2 instance type, and the image ID using which T2 instances (if any) were created. A CPU Credit provides the performance of a full CPU core for one minute. Traditional Amazon EC2 instance types provide fixed performance,

Measurement	Description	Measurement Unit	Interpretation
	last measurement period.		while T2 instances provide a baseline level of CPU performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits. One CPU credit is equal to one vCPU running at 100% utilization for one minute. Other combinations of vCPUs, utilization, and time are also equal to one CPU credit; for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes. Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set rate of CPU credits per hour, depending on instance size. When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting. When your T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization. The more credits your T2 instance has for CPU resources, the more time it can burst beyond its base performance level when more performance is needed. This implies that ideally, the value of the CPU credit usage measure should be low for an instance and the value of the CPU credit balance for that instance should be high, as that way, an instance is assured of more CPU resources when performance demands increase. By comparing the value of this measure across

Measurement	Description	Measurement Unit	Interpretation
			instances, you can precisely identify the instance that has used up a sizeable portion of its CPU credits.
CPU credit balance:	Indicates the number of CPU credits that have been earned by this T2 instance / all T2 instances / all T2 instances created from this image ID	Number	
Disk read operations:	Indicates the rate at which read operations were performed on all disks available to this instance.	Operations/Sec	Compare the value of this measure across instances to know which instance is too slow in processing read requests.
Disk write operations:	Indicates the rate at which write operations were performed on all disks available to this instance.	Operations/Sec	Compare the value of this measure across instances to know which instance is too slow in processing write requests.
Disk reads:	Indicates the rate at which data was read from all disks available to this instance.	KB/Sec	Compare the value of this measure to identify the instance that is the slowest in responding to read requests.
Disk writes:	Indicates the rate at which data was written to all disks available to this instance.	KB/Sec	Compare the value of this measure to identify the instance that is the slowest in responding to write requests.
Incoming network traffic:	Indicates the rate at which data was received by all network interfaces of this instance.	KB/Sec	Compare the value of these measures across instances to know which instance is consuming too much bandwidth. Then, compare the value of the Incoming network traffic and Outgoing network traffic measures of that instance to determine where bandwidth consumption was more - when receiving data

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td rowspan="2">over the network? or when sending data?</td></tr>
<tr><td>Outgoing network traffic:</td><td>Indicates the rate at which data was sent by all the network interfaces of this instance.</td><td>KB/Sec</td></tr>
<tr><td>EC2 status check:</td><td>Indicates whether a status check (system status check or instance status check) failed for this instance</td><td></td><td>Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. These status checks are of two types: system and instance status checks.

If either of these status checks fails, then this measure will report the value Failed. If none of these status checks fail, then this measure will report the value Passed.

The values that this measure can report and their corresponding numeric values are listed in the table below:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Failed</td><td>1</td></tr><tr><td>Passed</td><td>0</td></tr></table>
Note:

By default, this measure reports the Measure Values above to indicate whether a check passed or failed. In the graph of this measure however, the same is indicated using the numeric equivalents only.</td></tr>
<tr><td>EC2 instance status check:</td><td>Indicates whether/not this instance passed the EC2 instance status check in the last minute.</td><td></td><td>Instance status checks monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>The following are examples of problems that can cause instance status checks to fail:
Failed system status checks
Incorrect networking or startup configuration
Exhausted memory
Corrupted file system
Incompatible kernel
The values that this measure can report and their corresponding numeric values are listed in the table below:<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Failed</td><td>1</td></tr><tr><td>Passed</td><td>0</td></tr></table>Note:
By default, this measure reports the Measure Values above to indicate whether a check passed or failed. In the graph of this measure however, the same is indicated using the numeric equivalents only.</td></tr>
<tr><td>EC2 system status check:</td><td>Indicates whether/not this instance passed the EC2 system status check in the last minute.</td><td>Number</td><td>System status checks monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself (for example, by stopping and starting an instance, or by terminating and replacing an instance).
The following are examples of problems that</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>can cause system status checks to fail:
Loss of network connectivity
Loss of system power
Software issues on the physical host
Hardware issues on the physical host
The values that this measure can report and their corresponding numeric values are listed in the table below:<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Failed</td><td>1</td></tr><tr><td>Passed</td><td>0</td></tr></table>Note:
By default, this measure reports the Measure Values above to indicate whether a check passed or failed. In the graph of this measure however, the same is indicated using the numeric equivalents only.</td></tr>
</table>

Detailed Diagnosis:

Using the detailed diagnosis of the **EBS volumes** measure, you can identify the volumes that are attached to a particular EC2 instance.

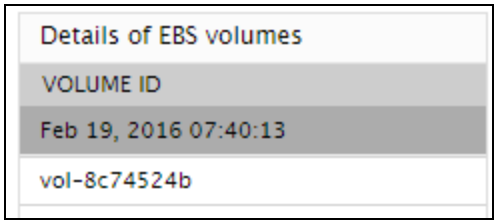


Figure 6.5: The detailed diagnosis of the EBS volumes measure

6.2.2 EC2 - Instance Uptime Test

In cloud-based environments, it is essential to monitor the uptime of server instances launched on the cloud. By tracking the uptime of each of the instances, administrators can determine what percentage of time an instance has been up. Comparing this value with service level targets, administrators can determine the most trouble-prone areas of the infrastructure hosted on the cloud.

In some environments, administrators may schedule periodic reboots of their instances. By knowing that a specific instance has been up for an unusually long time, an administrator may come to know that the scheduled reboot task is not working on an instance.

This test monitors the uptime of each instance available to the configured AWS user account.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for each instance launched by the configured AWS user account in the monitored region

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.

Parameter	Description
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Cloudwatch Enabled	**This parameter only applies to the EC2 -Aggregated Resource Usage test**. This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity, AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.
Report Manager Time	By default, this flag is set to **Yes**, indicating that, by default, the detailed diagnosis of this test, if enabled, will report the shutdown and reboot times of the cloud in the manager's time zone. If this flag is set to **No**, then the shutdown and reboot times are shown in the time zone of the system where the agent is running (i.e., the system system on which the remote agent is running).
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

Parameter	Description
	The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Has the instance been rebooted?:	Indicates whether this instance has been rebooted during the last measurement period or not.	Boolean	If this measure shows 1, it means that the instance was rebooted during the last measurement period. By checking the time periods when this metric changes from 0 to 1, an administrator can determine the times when this instance was rebooted.
Uptime of the instance during the last measure period:	Indicates the time period that the instance has been up since the last time this test ran.	Secs	If the instance has not been rebooted during the last measurement period and the agent has been running continuously, this value will be equal to the measurement period. If the instance was rebooted during the last measurement period, this value will be less than the measurement period of the test. For example, if the measurement period is 300 secs, and if the instance was rebooted 120 secs back, this metric will report a value of 120 seconds. The accuracy of this metric is dependent on the measurement period - the smaller the measurement period, greater the accuracy.
Total uptime of the instance:	Indicates the total time that this intance has been up since its last reboot.	Mins	Administrators may wish to be alerted if an instance has been running without a reboot for a very long period. Setting a threshold for this metric allows administrators to determine such conditions.

Detailed Diagnosis:

The detailed diagnosis of the *Has VM been rebooted?* measure reveals when the instance was last shutdown, when it was rebooted, how long the shutdown lasted, and whether the instance was shutdown as part of a routine maintenance exercise.

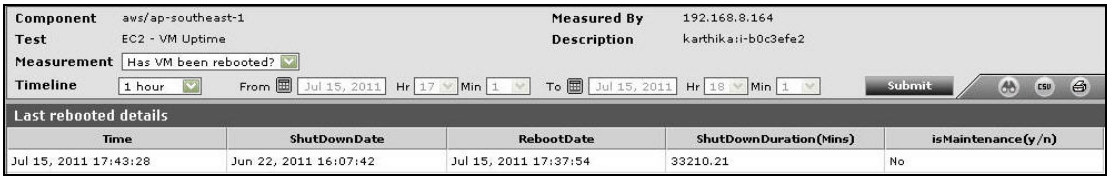


Figure 6.6: The detailed diagnosis of the Has VM been rebooted? measure

6.2.3 EC2 - Instances Test

An Amazon Machine Image (AMI) contains all information necessary to boot instances of your software. For example, an AMI might contain all the software to act as a web server (e.g., Linux, Apache, and your web site) or it might contain all the software to act as a Hadoop node (e.g., Linux, Hadoop, and a custom application). After an AMI is launched, the resulting running system is called an instance. All instances based on the same AMI start out identical and any information on them is lost when the instances are terminated or fail.

Users with valid AWS user accounts can sign into an EC2 region to view and use available instances, or purchase and launch new ones. With the help of this test, you can determine the total number of instances that are currently available for the configured AWS user account in the monitored region, the number of instances that were newly purchased/terminated, and the count of powered-off instances.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for the AWS EC2 Region being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key,	To monitor an Amazon EC2 instance, the eG agent has to be configured with the

Parameter	Description
AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Cloudwatch Enabled	**This parameter only applies to the EC2 -Aggregated Resource Usage test**. This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to AWS-EC2 Instance Connectivity,**

Parameter	Description
	AWS-EC2 Instance Resources , and AWS-EC2 Instance Uptime tests. In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Total instances:	Indicates the total number of instances currently available for the configured AWS user account.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances available for use for the configured AWS account, regardless of the current state of the instances.
Instances powered on:	Indicates the total number of instances that are currently powered-on.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the powered-on instances available for use for the configured AWS account.
Instances powered off:	Indicates the total number of instances that are currently powered-off.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the powered-off instances available for the configured AWS account.
Added instances:	Indicates the total number of instances that were newly purchased by the	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances that were newly purchased and launched by the configured AWS user

Measurement	Description	Measurement Unit	Interpretation
	configured AWS user account during the last measurement period.		account.
Removed instances:	Indicates the total number of instances that were newly terminated by the configured AWS user account during the last measurement period.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances that were newly terminated/removed by the configured AWS user account.

Detailed Diagnosis:

The detailed diagnosis capability of the *Total instances* measure, if enabled, shows the details of all the instances available for use for the configured AWS account in the monitored region, regardless of the current state of the instances.

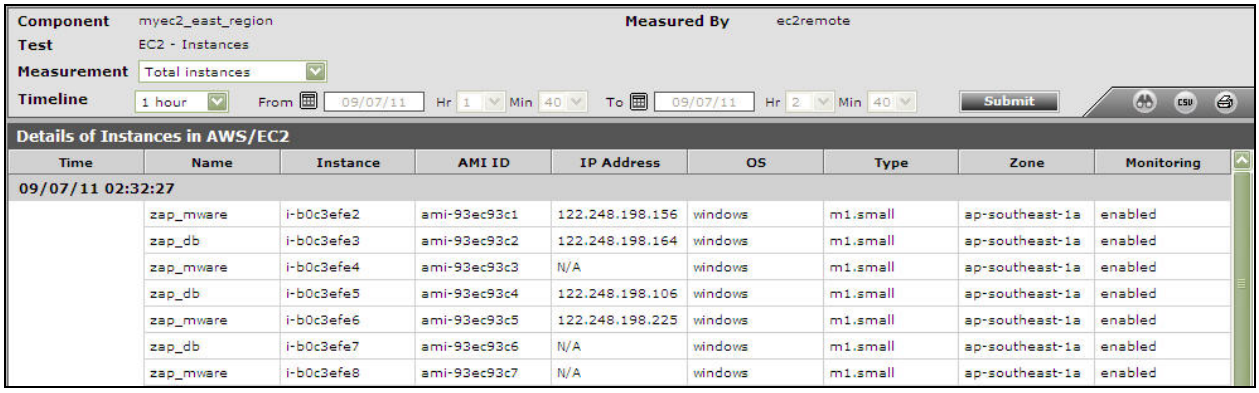


Figure 6.7: The detailed diagnosis of the Total instances measure

The detailed diagnosis capability of the *Instances powered on* measure, if enabled, shows the details of all the powered-on instances available for use for the configured AWS account in the monitored region.

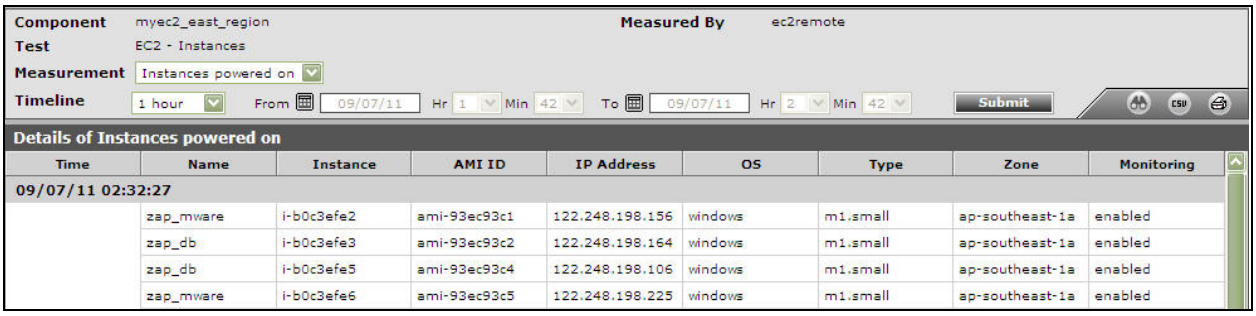


Figure 6.8: The detailed diagnosis of the Instances powered on measure

The detailed diagnosis capability of the *Instances powered off* measure, if enabled, shows the details of all the powered-off instances available for the configured AWS account.

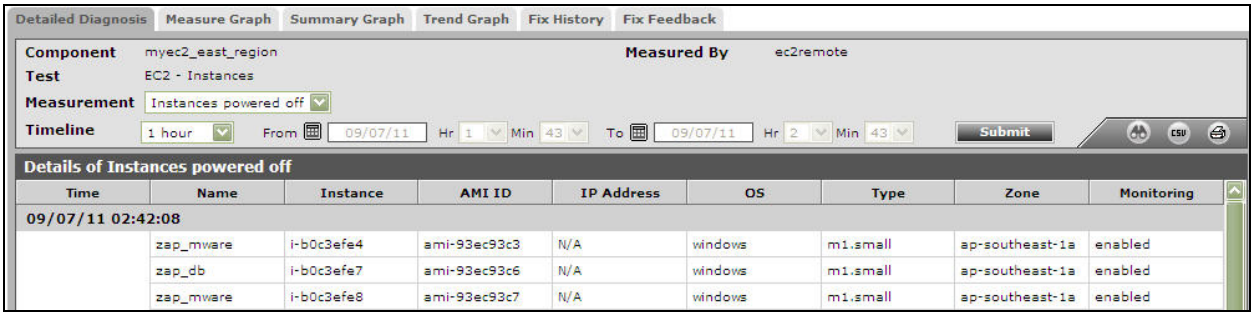


Figure 6.9: The detailed diagnosis of the Instances powered off measure

6.2.4 EC2 - Instance Resources Test

Tracking the CPU usage, disk and network I/O of every instance launched by a configured AWS user account in a region will provide administrators with valuable insights into how well the instances are utilizing the allocated resources. The **EC2 - Instance Resources** test does just that. This test auto-discovers the instances available for the configured AWS user account in a region, and reports the resource usage of each instance so that, administrators can quickly compare the usage metrics across instances and pinpoint which instance is resource-hungry.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for each instance launched by the configured AWS user account in the monitored region

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Cloudwatch Enabled	**This parameter only applies to the EC2 -Aggregated Resource Usage test**. This test reports critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want this test to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable is test to use this service, set the **CLOUDWATCH ENABLED** flag to **Yes**. On the other hand,

Parameter	Description
	to report resource usage metrics less frequently - say, once in 5 minutes or more - this test does not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to **No**. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site**.
Exclude Instance	**This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , and EC2 - Instance Uptime tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: i-b0c3e*,*7dbe56d. By default, this parameter is set to none.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
CPU utilization:	Indicates the percentage of allocated CPU consumed by this instance.	Percent	A high value for this measure indicates that an instance is utilizing CPU excessively - this could be because of one/more resource-intensive processes executing on that instance. Compare the value of this measure across instances to identify the CPU-intensive instances.
Incoming network traffic:	Indicates the rate of incoming network traffic i.e., the rate at which the bytes are received by all the network interfaces connected to this instance.	KB/Sec	Compare the values of these measures across instances to quickly identify the instance that is utilizing the network bandwidth excessively.
Outgoing network traffic:	Indicates the volume of outgoing network traffic i.e., the rate at which the bytes are transferred from all the network interfaces connected to this instance.	KB/Sec	

Measurement	Description	Measurement Unit	Interpretation
Disk reads:	Indicates the rate at which data is read from the disks of this instance.	KB/Sec	These measures are good indicators of the level of disk I/O activity on an instance. By comparing the values of these measures across instances, you can accurately determine which instance is performing I/O-intensive operations.
Disk writes:	Indicates the rate at which data is written to the disks of this instance.	KB/Sec	
Disk read operations:	Indicates the rate at which disk read operations are performed on this instance.	Operations/Sec	These measures are good indicators of the level of disk I/O activity on an instance type. By comparing the values of these measures across types, you can accurately determine the type of instances that is performing I/O-intensive operations.
Disk write operations:	Indicates the rate at which disk write operations were performed on this instance.	Operations/Sec	

6.3 The AWS EC2 Region Services Layer

The tests mapped to this layer auto-discover the server instances that are available (for the configured AWS user account) in a region, and reports the uptime and the resource usage of the individual instances. Resource-hungry instances and those that were recently rebooted can thus be isolated.

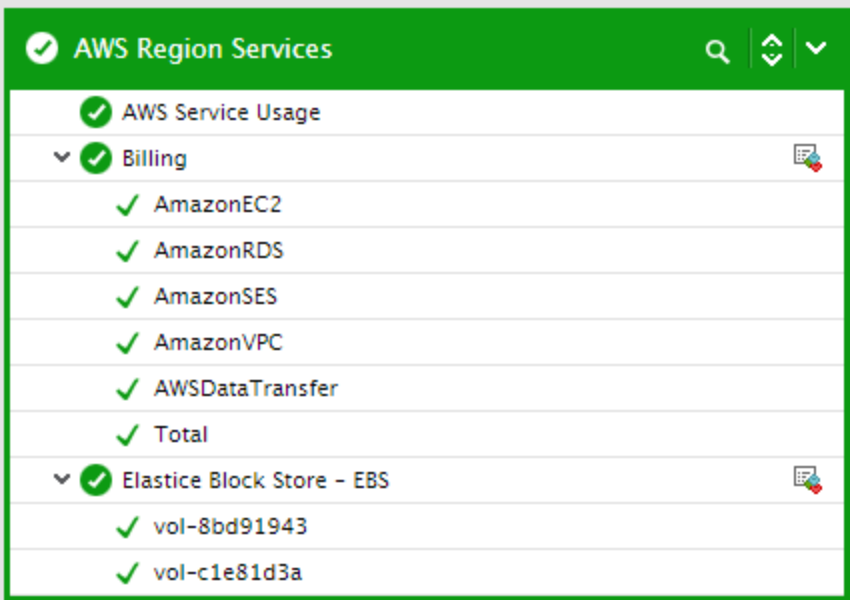


Figure 6.10: The tests mapped to the AWS EC2 Region Instance Details layer

6.3.1 AWS Service Usage Test

Use this test to receive an overview of the instances launched and services (EBS and RDS) used by a configured AWS user account in a monitored region. Understanding how many instances of a service are utilized by an account will help you to bill that user accordingly.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for the AWS account configured for monitoring

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The

Parameter	Description
	procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
EC2 instances:	Indicates the total number of EC2 instances currently available for the configured AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know which are the available instances.
EC2 instances poweredon:	Indicates the total number of instances that are currently powered-on for the configured AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know the names of the powered-on instances.
EBS volumes:	Indicates the total number of EBS volumes currently available for the configured AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know which volumes are available for use presently.
RDS instances:	Indicates the total number of RDS instances that are configured for the AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know the details of all RDS instances configured for the AWS user account.
RDS instances available:	Indicates the number of instances that are currently powered-on and available for the use of the configured AWS user account in the monitored region.	Number	Use the detailed diagnosis of this measure to know the details of all RDS instances that are powered-on and available for the use of the AWS user account.
S3 buckets:	Indicates the total number of S3 buckets	Number	Use the detailed diagnosis of this measure to know the bucket names and when they were

Measurement	Description	Measurement Unit	Interpretation
	available in for of the configured AWS user account in the monitored region.		created.

The detailed diagnosis of the **EC2 instances** measure lists the names of all EC2 instances available for the configured AWS user account.

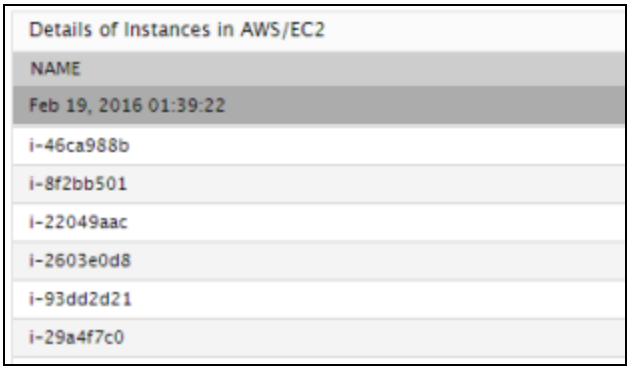


Figure 6.11: The detailed diagnosis of the EC2 instances measure

The detailed diagnosis of the **EC2 instances poweredon** measure lists the names of the powered-on instances alone.

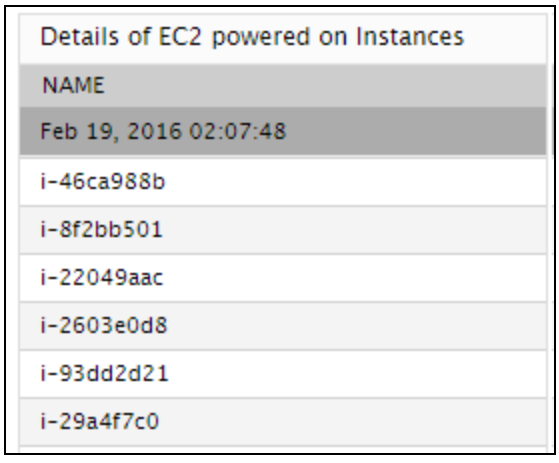


Figure 6.12: The detailed diagnosis of the EC2 instances poweredon measure

The detailed diagnosis of the **EBS volumes** measure displays the names of volumes that are available for use presently.

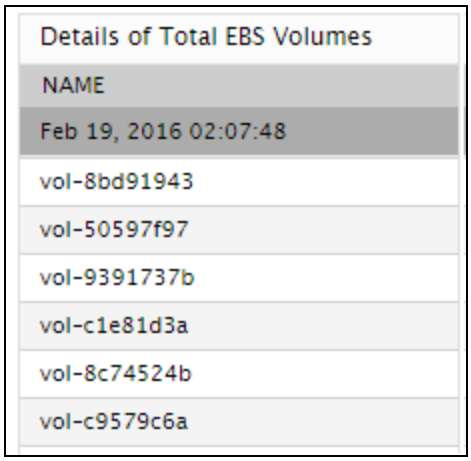


Figure 6.13: The detailed diagnosis of the EBS volumes measure

The detailed diagnosis of the **RDS instances** measure provides the details of all RDS instances configured for the AWS user account.

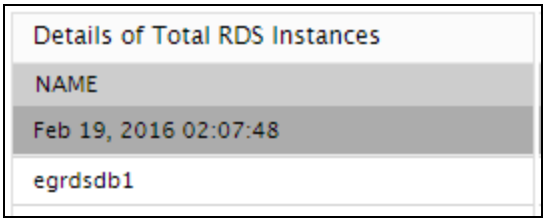


Figure 6.14: The detailed diagnosis of the RDS instances measure

The detailed diagnosis of the **RDS instances available** measure displays the details of the powered-on instances alone.

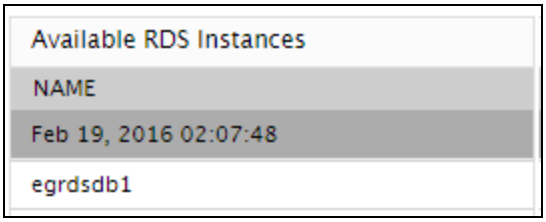


Figure 6.15: The detailed diagnosis of the RDS instances available measure

6.3.2 EC2 Container - ECS Test

AWS users can opt to run instances within Elastic Compute Cloud (EC2) or look into using containers. Amazon EC2 Container Service (ECS) manages Docker containers within AWS, allowing users to easily scale up or down and evaluate and monitor CPU usage. These AWS containers run on a managed cluster of EC2 instances, with ECS automating installation and

operation of the cluster infrastructure. The first step to get started with ECS therefore is to create a cluster and launch EC2 instances in it. Then, create task definitions. A task is one or more Docker containers running together for one service or a microservice. When configuring a container in your task definition, you need to define the container name and also indicate how much memory and how many CPU units you want to reserve for each container. Finally, you will have to create a service, so that you can run and maintain a specified number of instances of a task definition simultaneously.

Time and again, administrators will have to check on the resource usage of each cluster, so that they can identify those clusters that have been consistently over- utilizing the CPU and memory resources. Resource usage at the individual service- level should also be monitored, so that administrators can figure out whether the excessive resource consumption by a cluster is because the cluster itself does not have enough resources at its disposal, or because one/more services running on the cluster are depleting the resources. Using the AWS EC2 Container - ECS test, administrators can monitor resource usage both at the cluster and the service-level.

This test auto-discovers the clusters configured in the region being monitored and also the services running on each cluster. CPU and memory usage is then reported for each cluster and service, alongside the CPU and memory reservations (of all tasks) per cluster. These insights help administrators understand where there is a contention for resources - at the cluster-level? or at the service-level? or both? - and accordingly decide what needs to be done to optimize resource usage:

- Should more container instances be added to the cluster to increase the amount of resources at its disposal?
- Should the task definitions of the resource-hungry services be fine-tuned so that the service has more resources to use?

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for each cluster:service pair in the monitored region

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The

Parameter	Description
AWS Secret Key	procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Exclude Region	Here, you can provide a comma-separated list of region names or patterns of region names that you do not want to monitor. For instance, to exclude regions with names that contain 'east' and 'west' from monitoring, your specification should be: **east*,*west**
ECS Filter Name	By default, this test reports metrics for each cluster and for each service that is running on a cluster. Accordingly, *ServiceName* is the default selection from the **ECS FILTER** drop-down. If you do not want service-level metrics, then you can configure the test to report resource usage at the cluster-level alone. For this, just select *ClusterName* from the **ECS FILTER** drop-down. If this is done, then the test will only report cluster names as descriptors.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
CPU reservation:	The percentage of CPU	Percent	**This measure is reported at the**

Measurement	Description	Measurement Unit	Interpretation
	units that are reserved by running tasks in this cluster.		**cluster-level only - i.e., for the ClusterName descriptor alone.** This value is computed using the following formula: *Total CPU units reserved by ECS tasks on the cluster / Total CPU units that were registered for all the container instances in the cluster * 100* A value close to 100% indicates that almost all resources available to the cluster are being reserved by running tasks in that cluster. This implies that additional services cannot be configured on that cluster until more resources are made available to the cluster or until the CPU reservation of running tasks is reduced.
CPU utilization:	Indicates the percentage of CPU units used by this cluster or by this service	Percent	For a cluster, this value is computed using the following formula: *Total CPU units currently used by ECS tasks on this cluster / Total CPU units that were registered for all the container instances in this cluster * 100* A value close to 100% for this measure at the cluster-level could either indicate that the cluster is resource-starved or that one/more services running on the cluster are consuming excessive resources. If the reason for high CPU usage is the poor resource configuration of the cluster, then, you may want to add more instances to the cluster to add to its resource base. On the other hand, if the cluster is adequately sized with CPU,

Measurement	Description	Measurement Unit	Interpretation
			then you may want to check the value of this measure for each of the services running on the cluster . For a service, this value is computed using the following formula: *Total CPU units currently used by ECS tasks defined for this service / Total CPU units that are reserved for the tasks defined for this service * 100* Compare the value of this measure across services of a cluster to know which services of that cluster are guilty of over-utilization of CPU. Once the services are identified, check the CPU reservation of the task definitions of those services to determine whether sufficient resources have been allocated to those tasks. If not, increase the reservations to allow optimal resource usage.
Memory reservation:	The percentage of memory that is reserved by running tasks in this cluster.	Percent	**This measure is reported at the cluster-level only - i.e., for the ClusterName descriptor alone.** This value is computed using the following formula: *Total amount of memory reserved by ECS tasks on the cluster / Total amount of memory that was registered for all the container instances in the cluster * 100* A value close to 100% indicates that almost all resources available to the cluster are being reserved by running tasks in that cluster. This implies that additional services cannot be configured on that cluster until more resources are made available to the cluster or until the

Measurement	Description	Measurement Unit	Interpretation
			memory reservation of running tasks is reduced.
Memory utilization:	Indicates the percentage of memory used by this cluster or by this service	Percent	For a cluster, this value is computed using the following formula: *Total memory currently used by ECS tasks on this cluster / Total memory that is registered for all the container instances in this cluster * 100* A value close to 100% for this measure at the cluster-level could either indicate that the cluster is resource-starved or that one/more services running on the cluster are consuming excessive resources. If the reason for high memory usage is the poor resource configuration of the cluster, then, you may want to add more instances to the cluster to add to its resource base. On the other hand, if the cluster is adequately sized with memory, then you may want to check the value of this measure for each of the services running on the cluster . For a service, this value is computed using the following formula: *Total memory currently used by ECS tasks defined for this service / Total memory reserved for the tasks defined for this service * 100* Compare the value of this measure across services of a cluster to know which services of that cluster are guilty of over-utilization of memory. Once the services are identified, check the memory reservation of the task definitions of those services to determine

Measurement	Description	Measurement Unit	Interpretation
			whether sufficient resources have been allocated to those tasks. If not, increase the reservations to allow optimal resource usage.

6.3.3 RedShift Test

Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. The first step to create such a data warehouse is to launch an Amazon Redshift cluster. An Amazon Redshift cluster is a collection of computing resources called nodes. Each cluster runs an Amazon Redshift engine and contains one or more databases. Each cluster has a leader node and one or more compute nodes. The leader node receives queries from client applications, parses the queries, and develops query execution plans. The leader node then coordinates the parallel execution of these plans with the compute nodes, aggregates the intermediate results from these nodes, and finally returns the results back to the client applications. Compute nodes execute the query execution plans and transmit data among themselves to serve these queries. The intermediate results are sent back to the leader node for aggregation before being sent back to the client applications.

Where RedShift is in use, query performance, and consequently, the performance of the dependent client applications, depends upon the following factors:

- Cluster availability
- How the cluster and its nodes use the CPU, network, and storage resources of the cluster;
- Responsiveness of the nodes in the cluster to I/O requests from client applications

To be able to accurately assess whether cluster performance is at the desired level or not, an administrator would require real-time insights into each of the factors listed above. The RedShift test provides administrators with these valuable insights. By reporting the current health status of each cluster managed by RedShift, this test brings unavailable clusters to light. The resource usage of the cluster is also reported, so that potential resource contentions can be proactively isolated. Optionally, you can also configure this test to report metrics for individual nodes in the cluster as well. If this is done, then administrators will be able to instantly drill-down from a resource-hungry cluster to the exact node in the cluster that could hogging the resources. At the node-level, the latency and throughput of each node is also revealed. This way, when users complain of degradation in the performance of client applications, you can quickly identify the cluster and the precise node in the cluster that is slowing down I/O processing and consequently, impacting application performance.

Target of the test: Amazon EC2 Cloud

Agent deploying the test: A remote agentx

Output of the test:

One set of results for each cluster and/or node in the monitored AWS region

First level descriptor: Cluster

Second level descriptor: Node

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.

Parameter	Description
RedShift Filter Name	By default, this test reports metrics only for each cluster in each AWS region on the cloud. This is why, this flag is set to **ClusterIdentifier**, by default. If needed, you can configure the test to additionally report metrics for every node in every cluster. For node-level metrics, select the **NodeIdentifer** option from this drop-down. Upon selection, you will be able to view metrics both at the cluster-level and the node-level.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
CPU utilization:	Indicates the percentage of CPU utilized by this cluster/node.	Percent	For a cluster, this measure will report the aggregate CPU usage of all nodes in the cluster. If the value of this measure is consistently above 50% for a cluster, it indicates that a serious resource contention may occur on that cluster, if additional processing power is not provided to it. In such a case, you may want to consider adding more nodes to the cluster, or adding more CPUs to the existing nodes. You can also compare the CPU usage of nodes in the resource-hungry cluster to determine whether one/more nodes are hogging the CPU. If so, you may want to tweak the load-balancing algorithm of your cluster to ensure uniform load distribution.
Database connections:	Indicates the number of connections to the databases in this cluster.	Number	**This measure is only reported at the cluster-level and not the node-level.**
Health status:	Indicates the current health status of this cluster.	Percent	**This measure is only reported at the cluster-level and not the node-level.** Every minute the cluster connects to its database and performs a simple query. If it is able to perform this operation successfully, then the value of this measure will be *Healthy*. Otherwise, the value of this

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>measure will be Unhealthy. An Unhealthy status can occur when the cluster database is under extremely heavy load or if there is a configuration problem with a database on the cluster.

The numeric values that correspond to the measure values mentioned above are as follows:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Healthy</td><td>1</td></tr><tr><td>Unhealthy</td><td>0</td></tr></table>
Note:

This measure will report one of the Measure Values listed above to indicate the current state of a cluster. In the graph of this measure however, cluster status will be indicated using the numeric equivalents only.</td></tr>
<tr><td>Is maintenance mode?:</td><td>Indicates whether/not this cluster is in the maintenance mode presently.</td><td></td><td>The values that this measure can report and their corresponding numeric values are listed in the table below:

<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Yes</td><td>1</td></tr><tr><td>No</td><td>0</td></tr></table>
Note:

• This measure will report one of the Measure Values listed above to indicate whether/not a cluster is in the maintenance mode. In the graph of this measure however, the same will be</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			indicated using the numeric equivalents only. • This measure is reported only at the cluster-level and not the node-level. • Even though your cluster might be unavailable due to maintenance tasks, the *Health status* measure of the test will report the value *Healthy* for that cluster.
Network receive throughput:	Indicates the rate at which this cluster or node receives data.	KB/Secs	For a cluster, a consistent increase in the value of these measures is indicative of excessive usage of network resources by the cluster. In such a case, compare the value of these measures across the nodes of a cluster to identify the nodes that are over-utilizing network bandwidth.
Network transmit throughput:	Indicates the rate at which this cluster or node sends data.	KB/Secs	
Disk space used:	Indicates the percentage of disk space used by this cluster/node.	Percent	If the value of this measure is close to 100% for a cluster, it indicates that the cluster is rapidly running out of storage resources. You may want to consider adding more nodes to the cluster to increase the storage space available. Alternatively, you can add fewer nodes and yet significantly increase the cluster resources by opting for node types that are by default large-sized and hence come bundled with considerable storage space. When a cluster's storage resources are rapidly depleting, you may want to compare the space usage of the nodes in cluster, so that you can quickly isolate that node that is eroding the space. Tweaking your load-balancing algorithm could go a long way in eliminating such node overloads.

Measurement	Description	Measurement Unit	Interpretation
Read IOPS:	Indicates the average number of disk read operations performed by this node per second.	Reads/Sec	A high value is desired for this measure, as that's the trait of a healthy node. You can compare the value of this measure across nodes to identify the node that is slowest in processing read requests.
Read latency:	Indicates the average amount of time taken by this node for disk read I/O operations.	Reads/Sec	Ideally, the value of this measure should be very low. Its good practice to compare the value of this measure across nodes of a cluster and isolate those nodes in the cluster where the value of this measure is abnormally high. Such nodes slow down I/O processing and adversely affect application performance.
Read throughput:	Indicates the average number of bytes read from disk by this node per second.	KB/Sec	A high throughput signifies faster processing of read I/O requests. A low throughput is indicative of slow read request processing. Compare the value of this measure across nodes of a cluster to isolate those nodes that have registered an abnormally low value for this measure. Such nodes not only affect cluster performance, but also the performance of dependent client applications.
Write IOPS:	Indicates the average number o disk write operations performed by this node per second.	Writes/Sec	A high value is desired for this measure, as that's the trait of a healthy node. You can compare the value of this measure across nodes to identify the node that is slowest in processing write requests.
Write latency:	Indicates the average amount of time taken by this node for disk write I/O operations.	Secs	Ideally, the value of this measure should be very low. Its good practice to compare the value of this measure across nodes of a cluster and isolate those nodes in the cluster where the value of this measure is abnormally high. Such nodes slow down I/O processing and adversely affect application performance.
Write throughput:	Indicates the average	KB/Sec	A high throughput signifies faster processing

Measurement	Description	Measurement Unit	Interpretation
	number of bytes written to disk by this node per second.		of write I/O requests. A low throughput is indicative of slow write request processing. Compare the value of this measure across nodes of a cluster to isolate those nodes that have registered an abnormally low value for this measure. Such nodes not only affect cluster performance, but also the performance of dependent client applications.

6.3.4 Elastic Block Store - EBS Test

Amazon Elastic Block Store (Amazon EBS) provides persistent block level storage volumes for use with Amazon EC2 instances in the AWS Cloud. An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as system drive for an instance or storage for a database application. If such an EBS volume suddenly becomes unavailable or impaired, it is bound to adversely impact the operations of the EC2 instance attached to that volume, which in turn will damage the experience of the users of that instance. Administrators need to be promptly alerted to such problem conditions, so that they can instantly initiate remedial action and ensure high instance uptime. Besides volume status, administrators also need to track the I/O load on the EBS volume and continuously measure the ability of the volume to handle that load. This insight will enable administrators to provision the volumes with more or less I/O, so as to optimize I/O processing and maximize volume performance. The AWS Elastic Block Store - EBS test helps administrators in this exercise. The test periodically checks the health and availability status of each volume used by the EC2 instances in the monitored region and notifies administrators if any volume is in an abnormal state. Similarly, the test also tracks the I/O load on every volume and measures how well each volume processes the load - overloaded volumes and those that are experiencing processing hiccups are highlighted in the process.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key,	To monitor an Amazon EC2 instance, the eG agent has to be configured with the

Parameter	Description
AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

Parameter	Description
	• The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>State</td><td>Indicates the current state of this volume.</td><td></td><td>The values that this measure can report and their corresponding numeric values are detailed in the table below:
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td>Creating</td><td>The volume is being created. The volume will be inaccessible during creation.</td><td>0</td></tr>
<tr><td>Available</td><td>The volume is available</td><td>1</td></tr>
<tr><td>In-use</td><td>The volume is in use</td><td>2</td></tr>
<tr><td>Deleting</td><td>The volume is being deleted</td><td>3</td></tr>
<tr><td>Deleted</td><td>The volume is deleted</td><td>4</td></tr>
</table>
</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			<table><tr><td>Error</td><td>Some error has occurred in the volume</td><td>5</td></tr></table> The detailed diagnosis of this measure will reveal when the volume was created and in which availability zone it resides. **Note:** By default, this measure will report the **Measure Value**s listed in the table above to indicate the current availability state of a volume. In the graph of this measure however, the same will be represented using the numeric equivalents only. If any EBS volume is found to be in an abnormal state, then you can use the detailed diagnosis of this measure to know the volume type, when that volume was created, and in which availability zone the volume resides.
Status	Indicates the current health status of this volume		AWS EC2 periodically runs volume status checks to enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. The value that this measure reports varies with the status reported by the volume status checks. The table below describes what value this measure reports when , and also lists the numeric values that correspond to the measure values. <table><tr><th>Measure</th><th>Descriptio</th><th>Numeri</th></tr></table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>n</th><th>c Value</th></tr>
<tr><td>OK</td><td>If all checks pass, the status of the volume is OK.</td><td>0</td></tr>
<tr><td>Impaired</td><td>If a check fails, the status of the volume is impaired</td><td>1</td></tr>
<tr><td>Insufficient-data</td><td>If checks are in progress, then insufficient-data is reported</td><td>2</td></tr>
</table>
Note:
By default, this measure will report the Measure Values listed in the table above to indicate the current status of a volume. In the graph of this measure however, the same will be represented using the numeric equivalents only.
</td></tr>
<tr><td>Idle time:</td><td>Indicates the total number of seconds during which no read or write operations were submitted to this volume.</td><td>Secs</td><td></td></tr>
<tr><td>Queue length:</td><td>Indicates the</td><td>Number</td><td>A consistent increase in the value of this</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
	number of read and write operation requests waiting to be completed.		measure could indicate a I/O processing bottleneck on the volume.
Read operations:	Indicates the rate at which read operations were performed on this volume.	Operations/Sec	Compare the value of this measure across volumes to know which volume is too slow in processing read requests.
Write operations:	Indicates the rate at which write operations were performed on this volume.	Operations/Sec	Compare the value of this measure across volumes to know which volume is too slow in processing write requests.
Reads:	Indicates the rate at which data was read from this volume.	KB/Sec	Compare the value of this measure to identify the volume that is the slowest in responding to read requests.
Writes:	Indicates the rate at which data was written to this volume.	KB/Sec	Compare the value of this measure to identify the volume that is the slowest in responding to write requests.
Total read time:	Indicates the total time taken by all completed read operations.	Secs	A very high value for this measure could indicate that the volume took too long to service one/more read requests.
Total write time:	Indicates the total time taken by all completed write operations.	Secs	A very high value for this measure could indicate that the volume took too long to service one/more write requests.
Provisioned IOPS (SSD)volume throughput:	Indicates the percentage of I/O operations per second (IOPS) delivered of the total IOPS	Percent	**This measure will be reported for Provisioned IOPS volumes only.** Provisioned IOPS (SSD) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and

Measurement	Description	Measurement Unit	Interpretation
	provisioned for this volume.		consistency in random access I/O throughput. You specify an IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year. A Provisioned IOPS (SSD) volume can range in size from 4 GiB to 16 TiB and you can provision up to 20,000 IOPS per volume. The ratio of IOPS provisioned to the volume size requested can be a maximum of 30; for example, a volume with 3,000 IOPS must be at least 100 GiB. You can stripe multiple volumes together in a RAID configuration for larger size and greater performance. For smaller I/O operations, you may even see an IOPS value that is higher than what you have provisioned - i.e., the value of this measure can be greater than 100%. This could be because the client operating system may be coalescing multiple smaller I/O operations into a smaller number of large chunks. On the other hand, if the value of this measure is consistently lower than the expected IOPS or throughput you have provisioned, then ensure that your EC2 bandwidth is not the limiting factor; your instance should be EBS-optimized (or include 10 Gigabit network connectivity) and your instance type EBS dedicated bandwidth should exceed the I/O throughput you intend to drive. Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes.
Size:	Indicates the current size of this volume.	GB	For a General Purpose (SSD) Volume, volume size is what dictates the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have

Measurement	Description	Measurement Unit	Interpretation
			higher baseline performance levels and accumulate I/O credits faster. For a Provisioned IOPS (SSD) Volume, the ratio of IOPS provisioned to volume size can be a maximum of 30; for example, a volume with 3,000 IOPS must be at least 100 GiB. Magnetic volumes can range in size from 1 GiB to 1 TiB.
Total IOPS:	Indicates the total number of I/O operations that were performed on this volume per second.	Operations/Sec	IOPS are input/output operations per second. Amazon EBS measures each I/O operation per second (that is 256 KiB or smaller) as one IOPS. I/O operations that are larger than 256 KiB are counted in 256 KiB capacity units. For example, a single 1,024 KiB I/O operation would count as 4 IOPS; however, 1,024 I/O operations at 1 KiB each would count as 1,024 IOPS. When you create a 3,000 IOPS volume, either a 3,000 IOPS Provisioned IOPS (SSD) volume or a 1,000 GiB General Purpose (SSD) volume, and attach it to an EBS-optimized instance that can provide the necessary bandwidth, you can transfer up to 3,000 chunks of data per second (provided that the I/O does not exceed the per volume throughput limit of the volume). If your I/O chunks are very large, then the value of this measure may be lesser than what you provisioned because you are hitting the throughput limit of the volume. For example 1,000 GiB General Purpose (SSD) volume has an IOPS limit of 3,000 and a volume throughput limit of 160 MiB/s. If you are using a 256 KiB I/O size, your volume will reach its throughput limit at 640 IOPS (640 x 256 KiB = 160 MiB). For smaller I/O sizes (such as 16 KiB), this same volume can sustain 3,000

Measurement	Description	Measurement Unit	Interpretation
			IOPS because the throughput is well below 160 MiB/s. On Provisioned IOPS Volumes, for smaller I/O operations, you may even see that the value of this measure is higher than what you have provisioned. This could be because the client operating system may be coalescing multiple smaller I/O operations into a smaller number of large chunks. On the other hand, if the value of this measure is consistently lower than the expected IOPS or throughput you have provisioned for a Provisioned IOPS volume, then ensure that your EC2 bandwidth is not the limiting factor; your instance should be EBS-optimized (or include 10 Gigabit network connectivity) and your instance type EBS dedicated bandwidth should exceed the I/O throughput you intend to drive. Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes. Magnetic volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS.
IOPS limits:	Indicates the IOPS limit of this volume.	Operations/Sec	For Provisioned IOPS volumes, the IOPS limit is specified when creating the volumes. For General Purpose IOPS volumes, the volume size dictates the baseline IOPS limit of that volume and how quickly it accumulates I/O credits.
IOPS utilization:	Indicates the percentage of provisioned IOPS or IOPS limit that is being utilized by this volume.	Percent	This metric can also help you identify over-utilized volumes, which could be impacting application performance. In these cases, you could improve performance by upgrading to a different volume type or provisioning more IOPS.

Measurement	Description	Measurement Unit	Interpretation
Throughput:	Indicates the rate of reads and writes processed by this volume.	KB/Second	A consistent drop in this value could indicate a I/O processing bottleneck on the volume. You may want to closely track the variations to this measure, so that you can proactively identify the volume that may soon reach its throughput limit. The maximum throughput of each volume type is indicated below: • General purpose volumes - 160 MiB/sec • Provisioned IOPS volumes - 320 MiB/sec • Magnetic volumes - 40-90 MiB/sec If your I/O chunks are very large, then a volume will reach its throughput limit much before its IOPS limit is reached. If you are not experiencing the throughput you have provisioned, ensure that your EC2 bandwidth is not the limiting factor; your instance should be EBS-optimized (or include 10 Gigabit network connectivity) and your instance type EBS dedicated bandwidth should exceed the I/O throughput you intend to drive.

Detailed Diagnosis:

The detailed diagnosis of the **State** measure of a volume will reveal when the volume was created and in which availability zone it resides.

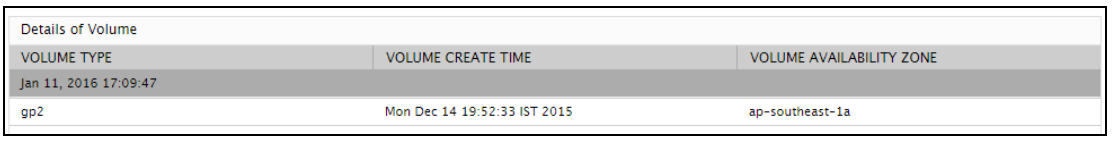


Figure 6.16: The detailed diagnosis of the State measure of the AWS Elastic Block Store - EBS Test

6.3.5 Simple Email Service - SES Test

Amazon Simple Email Service (Amazon SES) is a cost-effective email service built on the reliable and scalable infrastructure that Amazon.com developed to serve its own customer base. This service allows you to build an email functionality into an application that you are running on AWS. With Amazon SES, you can send transactional email, marketing messages, or any other type of high-quality content to your customers. You can also use Amazon SES to receive messages and deliver them to an Amazon S3 bucket, call your custom code via an AWS Lambda function, or publish notifications to Amazon SNS.

Amazon SES has a set of sending limits to regulate the number of email messages that you can send and the rate at which you can send them. Depending upon the level of email activity in your environment, you may want to modify these limits, as any violation will result in mails not being sent at all. You may hence have to closely study the email activity in your environment and determine whether/not the sending limits need to be fine-tuned. The **Simple Email Service - SES** test helps with this! By reporting the send quotas configured along with the count of mails sent and the send rate for the monitored AWS region, this test readily provides you with all the information you need to take the right decision with regards to whether/not the quota needs to be reset.

Also, the key measure of the performance of any email service is successful message delivery. If a majority of the delivery attempts made at any given point in time resulted in bounces, rejections, or complaints, it is a problem condition that warrants an investigation. The **Simple Email Service - SES** test proactively alerts you to such abnormalities! For the monitored region, the test reports the count and percentage of emails bounced, mails rejected, and complaints received, and notifies you if these values exceed acceptable limits.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for the region being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the

Parameter	Description
Key, Confirm AWS Secret Key	access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Sending quota:	Indicates the maximum number of emails that can be sent in a day.	Emails/Day	The sending quota reflects a rolling time period. Every time you try to send an email, Amazon SES checks how many emails you sent in the previous 24 hours. As long as the total number of emails that you have sent is less than your quota, your send request will be accepted and your email will be sent. If you have already sent your full quota, your send request will be

Measurement	Description	Measurement Unit	Interpretation
			rejected with a throttling exception. You will not be able to send more emails until some of the previous sending rolls out of its 24-hour window.
Total sent:	Indicates the total number of emails sent during the last 24 hours.	Number	If the value of this measure keeps growing closer to the value of the value of the Sending quota measure, it implies a high level of email activity in the region. Under such circumstances, it is best to increase the sending quota, so that the quota is not violated, causing SES to stop sending emails.
Sents:	Indicates the percentage of sending quota that was exhausted in the last 24 hours.	Percent	This measure is computed using the following formula: (Total sent / Sending quota) * 100 If the value of this measure is consistently higher than 50%, it implies a high level of email activity in the region. Under such circumstances, it is best to increase the sending quota, so that the quota is not violated, causing SES to stop sending emails.
Max send rate:	Indicates the maximum number of emails that can send per second.	Emails/Sec	You can exceed this limit for short bursts, but not for a sustained period of time.
Current bounces:	Indicates the number of emails that were bounced during the last measurement period.	Number	An email is hard-bounced when the email is rejected by the recipient's ISP or rejected by Amazon SES because the email address is on the Amazon SES suppression list. This measure reports the count of hard bounces alone. The value of this measure should be kept at a minimum, as excessive bounces constitute abuse and can put

Measurement	Description	Measurement Unit	Interpretation
			your AWS account at the risk of termination.
Bounce:	Indicates the percentage of emails that were bounced during the last measure period.	Percent	Ideally, the value of this measure should be very low. A high value constitutes abuse and can put your AWS account at the risk of termination.
Complaints:	Indicates the number of complaints received during the last measure period.	Number	If an email is accepted by the ISP and delivered to the recipient, but the recipient does not want the email and clicks a button such as "Mark as spam.", then SES will send you a complaint notification. The value of this measure should be kept at a minimum, as a large number of complaints constitute abuse and can put your AWS account at the risk of termination.
Complaint:	Indicates the percentage of complaints received by this region during the last measure period.	Percent	Ideally, the value of this measure should be very low. A high value constitutes abuse and can put your AWS account at the risk of termination.
Current rejected:	Indicates the number of emails that were rejected during the last measurement period.	Number	A rejected email is an email that Amazon SES initially accepted, but later rejected because the email contained a virus. Amazon SES notifies you by email and does not send the message. A high value for this measure is a cause for concern as it could indicate that your email system is severely infected.
Rejected:	Indicates the percentage of emails that were rejected	Percent	A high value for this measure is a cause for concern as it could indicate

Measurement	Description	Measurement Unit	Interpretation
	during the last measurement period.		that your email system is severely infected.
Current delivery attempts:	Indicates the number of mails sent during the last measurement period.	Number	

6.3.6 Relational Database Service - RDS Test

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizeable capacity for an industry-standard relational database and manages common database administration tasks. It also manages backups, software patching, automatic failure detection, and recovery.

The basic building block of Amazon RDS is the DB instance. A DB instance is an isolated database environment in the cloud. A DB instance can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a stand-alone database instance.

Each DB instance runs a DB engine. Amazon RDS currently supports the MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines.

The computation and memory capacity of a DB instance is determined by its DB instance class. For each DB instance, you can select from 5 GB to 6 TB of associated storage capacity. Each DB instance class has minimum and maximum storage requirements for the DB instances that are created from it. You can select the DB instance that best meets your needs. If your needs change over time, you can change DB instances. For example, initially, you may have launched a standard (previous generation) DB instance, which provides a balance of compute, memory, and network resources for your applications. However later, based on usage, you may have realized that a burst capable - current generation DB instance, with the capability to burst to full CPU usage, is ideal for your needs. In such circumstances, RDS facilitates the switch from one type DB instance to another. But to understand which DB instance class best suits your needs and make timely and accurate adjustments to your DB instance class selection, you will have to constantly track the CPU, memory, network, and space usage of each active DB instance on the cloud and derive usage patterns. Also, to ensure optimal storage performance, you additionally need to keep an eye on the I/O operations performed on the DB instances and identify latent DB instances. This is exactly what the Relational Database Service - RDS enables you to achieve.

This test closely tracks the current status, resource usage, and I/O activity of every active DB instance on a monitored region, and brings the following to light:

- Is any DB instance in an abnormal state presently?
- How are the DB instances using the CPU resources they have been configured with? Is any DB instance consuming high levels of CPU consistently? Should the DB instance class be changed?
- Does the DB instance have enough RAM? Will changing the DB instance class help in reducing the memory pressure on the instance?
- Do any db.t2 instances have a poor CPU credit balance?
- Is the disk I/O queue of any DB instance abnormally high? Which instance is this and when is I/O latency on that instance very high - when reading from or writing to the instance?
- Which DB instance is hungry for network bandwidth?
- Do all DB instances have enough free space? If not, which ones are rapidly running short of space?

Target of the test : Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for each active DB instance / DB instance class / DB engine name (depending upon the option you choose from the **RDS FILTER** drop-down) in the monitored region

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should

Parameter	Description
	make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
RDS Filter Name	By default, this test reports metrics for each active DB instance on the cloud. This is why, this flag is set to**DBInstanceIdentifier**, by default. If needed, you can pick either of the following options from this drop-down: • **DatabaseClass:** The computation and memory capacity of a DB instance is determined by its DB instance class. If you select this option, then this test will report metrics for each DB instance class. In other words, eG will aggregate metrics for all databases that belong to a DB intance class, and will present these metrics at the macro class-level. • **EngineName:** Each DB instance runs a DB engine. Amazon RDS currently supports the MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines. Each DB engine has its own supported features, and each version of a DB engine may include specific features. If you select this option, then this test will report metrics for every DB engine. In this case, eG will aggregate metrics for all databases using a particular engine, and will present these metrics at the macro engine-level.

Measures reported by the test:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>RDS instance status:</td><td>Indicates the current status of this DB instance.</td><td></td><td>This measure is reported only for a DB instance descriptor.
The values that this measure reports, the significance of each of these values, and the numeric values that correspond to them are discussed in the table below:
<table><tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr><tr><td>Failed</td><td>The instance has failed and Amazon RDS was unable to recover it. Perform a point-in-time restore to the latest restorable time of the instance to recover the data.</td><td>0</td></tr><tr><td>Available</td><td>The instance is healthy and available</td><td>1</td></tr><tr><td>Backing-up</td><td>The instance is currently being backed up.</td><td>2</td></tr><tr><td>Creating</td><td>The instance is being created. The instance is</td><td>3</td></tr></table></td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
	inaccessible while it is being created.	
Inaccessible-encryption-credentials	The KMS key used to encrypt or decrypt the DB instance could not be accessed.	4
Incompatible-credentials	The supplied CloudHSM username or password is incorrect. Please update the CloudHSM credentials for the DB instance.	5
Incompatible-network	Amazon RDS is attempting to perform a recovery action on an instance but is unable to do so because the VPC is in a state that is	6

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
	preventing the action from being completed. This status can occur if, for example, all available IP addresses in a subnet were in use and Amazon RDS was unable to get an IP address for the DB instance.	
Incompatible-option-group	Amazon RDS attempted to apply an option group change but was unable to do so, and Amazon RDS was unable to roll back to the previous option group state. Consult the	7

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
	Recent Events list for the DB instance for more information. This status can occur if, for example, the option group contains an option such as TDE and the DB instance does not contain encrypted information.	
Incompatible-parameters	Amazon RDS was unable to start up the DB instance because the parameters specified in the instance's DB parameter group were not	8

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
	compatible. Revert the parameter changes or make them compatible with the instance to regain access to your instance. Consult the Recent Events list for the DB instance for more information about the incompatible parameters.	
Incompatible-restore	Amazon RDS is unable to do a point-in-time restore. Common causes for this status include using temp tables or using MyISAM tables.	9

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
Maintenance	Amazon RDS is applying a maintenance update to the DB instance.	10
Modifying	The instance is being modified because of a customer request to modify the instance.	11
Rebooting	The instance is being rebooted because of a customer request or an Amazon RDS process that requires the rebooting of the instance.	12
Renaming	The instance is being renamed because of a customer request to rename it.	13

Measurement	Description	Measurement Unit	Interpretation

Measure Value	Description	Numeric Value
Resetting-master-credentials	The master credentials for the instance are being reset because of a customer request to reset them.	14
Restore-error	The DB instance encountered an error attempting to restore to a point-in-time or from a snapshot.	15
Upgrading	The database engine version is being upgraded.	16
Storage-full	The instance has reached its storage capacity allocation. This is a critical status and should be remedied	17

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td></td><td>immediately; you should scale up your storage by modifying the DB instance. Set alarms to warn you when storage space is getting low so you don't run into this situation.</td><td></td></tr>
<tr><td>Deleting</td><td>The instance is being deleted.</td><td>18</td></tr>
</table>
Note:
This measure reports the Measure Values listed in the table above to indicate the current status of a DB instance. In the graph of this measure however, the same will be represented using the corresponding numeric equivalents only.
</td></tr>
<tr><td>CPU credit usage:</td><td>Indicates the number of CPU units consumed by this T2 DB instance/ all DB instances that belong to this T2 DB instance class / all T2 DB instances using this DB engine,</td><td>Number</td><td>These measures are reported only for individual T2 instances, instances that belong to T2 DB instance classes, and the DB engines used only by T2 instances.
A CPU Credit provides the performance of a full CPU core for one minute. Traditionalinstance types provide fixed performance, while T2 instances provide a baseline level of CPU</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
	during the last measurement period.		performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits. One CPU credit is equal to one vCPU running at 100% utilization for one minute. Other combinations of vCPUs, utilization, and time are also equal to one CPU credit; for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes. Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set rate of CPU credits per hour, depending on instance size. When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting. When your T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization. The more credits your T2 instance has for CPU resources, the more time it can burst beyond its base performance level when more performance is needed. This implies that ideally, the value of the CPU credit usage measure should be low for an instance and the value of the CPU credit balance for that instance should be high, as that way, an instance is assured of more CPU resources when performance demands increase. By comparing the value of this measure across instances, you can precisely identify the instance that has used up a sizeable portion of

Measurement	Description	Measurement Unit	Interpretation
			its CPU credits.
CPU credit balance:	Indicates the number of CPU credits that an instance has accumulated.	Number	
CPU utilization:	Indicates the percentage of CPU utilized by this DB instance / DB instance class / DB engine	Percent	A value close to 100% for this measure for any DB instance is indicative of excessive CPU usage by that instance. Track the variations to the value of this measure for such an instance closely, and figure out whether CPU usage is consistently high and close to 100%. If so , you can conclude that the instance requires more CPU than what's been allocated to it. You may want to change to the DB instance class definition to allot more CPU resources to all instances it governs.
Binlog disk usage:	Indicates the amount of disk space occupied by binary logs on this DB instance / all DB instances of this DB instance class / all DB instances using this DB engine	KB	The binary log on MySQL has two important purposes: • For replication, the binary log on a master replication server provides a record of the data changes to be sent to slave servers. The master server sends the events contained in its binary log to its slaves, which execute those events to make the same data changes that were made on the master. . • Certain data recovery operations require use of the binary log. After a backup has been restored, the events in the binary log that were recorded after the backup was made are re-executed. These events bring databases up to date from the point of the backup. Typically, MySQL uses several logging formats

Measurement	Description	Measurement Unit	Interpretation
			to record information in the binary log. There are three logging formats: • Replication capabilities in MySQL originally were based on propagation of SQL statements from master to slave. This is called statement-based logging. • In row-based logging, the master writes events to the binary log that indicate how individual table rows are affected. • A third option is also available: mixed logging. With mixed logging, statement-based logging is used by default, but the logging mode switches automatically to row-based in certain cases. MySQL on Amazon RDS supports both the row-based and mixed binary logging formats for MySQL version 5.6. The default binary logging format is mixed. For DB instances running MySQL versions 5.1 and 5.5, only mixed binary logging is supported. If the value of this measure grows consistently, it could mean that large binary files are being created. At this juncture, you may want to check the logging format configured for MySQL on Amazon RDS. This is because, very often, row-based binary logging format can result in very large binary log files. If you do not change the logging mode, then such files will continue to be created, thereby reducing the amount of storage space available for a DB instance. This in turn can increase the amount of time to perform a restore operation of a DB instance.
Database connections:	Indicates the number	Number	

Measurement	Description	Measurement Unit	Interpretation
	of database connections currently used by this instance / all instances that belong to this DB instance class / all instances using this DB engine		
Disk queue depth:	Indicates the number of outstanding IOs (read/write requests) waiting to access this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine.	Number	If the value of this measure keeps increasing steadily and significantly for a DB instance, it could indicate that the DB instance is latent, and is unable to process I/O requests quickly. The value of this measure therefore should be low at all times.
Freeable memory:	Indicates the amount of available random access memory for this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine	MB	A high value is desired for this measure to ensure peak performance of a DB instance.
Replica lag time:	Indicates the amount of time a Read Replica DB Instance lags behind this source DB Instance / all source DB instances that belong to this DB instance class / all source DB instances using	Secs	**This measure applies to MySQL read replicas only.** f your system runs on Amazon Relational Database Service (RDS) you may have opted to configure one or more replicas for your main MySQL database(s). This means you have a master RDS instance and at least one slave RDS instance which receives updates from the master. This process is called replication. Replication ensures that changes made on the master database also happen on the slave after

Measurement	Description	Measurement Unit	Interpretation
	DB engine		some period of time. For a variety of reasons this period of time can increase. For example, a long- running query or erroneous query can cause replication to slow down or stop entirely. This results in replication lag: changes made on your main database aren't showing up on the slave replica because the replica is lagging behind. If the value of this measure is increasing consistently for a DB instance, it is a cause for concern, as it indicates that the slave is not in sync with the master and will take a long time to catch up. If for any reason the master DB instance fails at this juncture, there is bound to be significant data loss owing to the master-slave non-sync. When there is a replication issue the output of *show slave status*; is quite useful in debugging and resolving it. You need to review the values of: Slave_SQL_Running Last_Error Last_SQL_Error When a particular SQL query failed on the slave it could be that execution of queries in general has stopped. This is indicated by *Slave_SQL_Running* having the value *No*. In that case you'll either need to: • Remedy the error by fixing the issue that caused the SQL query to fail. • Decide to resume replication by letting the slave ignore that error. The former situation can be tricky as it requires

Measurement	Description	Measurement Unit	Interpretation
			you to figure out what data or query is problematic based on the values of *Last_Error* and *Last_SQL_Error*. These fields may provide enough information to determine any incorrect records but this is not always the case. In the latter case you would execute the following command on the slave: *CALL mysql.rds_skip_repl_error;* You should only run this command when you've determined that skipping the SQL query won't lead to inconsistent data or incorrect data on the slave (or, at least, that this is allowed to occur by skipping that particular SQL query).
Swap usage:	Indicates the amount of swap space used on this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine.	KB	
Read IOPS:	Indicates the rate at which disk read I/O operations were performed by this DB instance / all DB instances that belong to this DB instance class / all DB instances using this DB engine	Reads/Sec	Ideally, the value of this measure should be high. A consistent drop in this value could indicate a read latency.
Write IOPS:	Indicates the rate at which disk write I/O operations were performed by this DB instance / all	Writes/Sec	Ideally, the value of this measure should be high. A consistent drop in this value could indicate a write latency.

Measurement	Description	Measurement Unit	Interpretation
	DB instances that belong to this DB instance class / all DB instances using this DB engine		
Read latency:	Indicates the average amount of time this DB instance / all DB instances of this instance class / all DB instances using this engine, took to service read requests.	Secs	Ideally, the value of this measure should be low. A consistent rise in this value could indicate a read latency. Compare the value of this measure across DB instances to know which instance is the slowest in servicing read requests.
Write latency:	Indicates the average amount of time this DB instance / all DB instances of this instance class / all DB instances using this engine, took to service write requests.	Secs	Ideally, the value of this measure should be low. A consistent rise in this value could indicate a write latency. Compare the value of this measure across DB instances to know which instance is the slowest in servicing write requests.
Read throughput:	Indicates the rate at which data was read from the disk by this DB instance / all DB instances of this instance class / all DB instances using this DB engine.	KB/Sec	Ideally, the value of this measure should be high. A steady decrease in this value could indicate a read latency. Compare the value of this measure across DB instances to know which instance is the slowest in servicing read requests.
Write throughput:	Indicates the rate at which data was written to the disk by this DB instance / all DB instances of this instance class / all DB instances	KB/Sec	Ideally, the value of this measure should be high. A steady decrease in this value could indicate a write latency. Compare the value of this measure across DB instances to know which instance is the slowest in servicing write requests.

Measurement	Description	Measurement Unit	Interpretation
	using this DB engine.		
Network receive throughput:	Indicates the incoming network traffic on this DB instance / all DB instances that belong to this instance class / all DB instances using this engine.	KB/Secs	The value of these measures includes both customer database traffic and Amazon RDS traffic used for monitoring and replication. A high value for these measures is indicative of high bandwidth usage by a DB instance. Under such circumstances, compare the value of the Network receive throughput measure with that of th Network transmit throughput measure to determine when the maximum bandwidth was consumed - when sending data or when receiving it?
Network transmit throughput:	Indicates the outgoing network traffic on this DB instance / all DB instances that belong to this instance class / all DB instances using this engine.	KB/Secs	
Total storage space:	Indicates the total amount of storage space allocated to this DB instance / all DB instances that belong to this instance class / all DB instances using this DB engine.	MB	
Used storage space:	Indicates the amount of storage space used by this DB instance / all DB instances that belong to this instance class / all DB instances using this DB engine.	MB	Compare the value of this measre across DB instances to know which instance is consuming storage space excessively.
Free storage	Indicates the amount	MB	A high value for this measure is ideal. Compare

Measurement	Description	Measurement Unit	Interpretation
space:	of storage space still unused by this DB instance / all DB instances that belong to this instance class / all DB instances using this DB engine.		the value of this measure across DB instances to know which instance is left with very little free space.
Free storage space:	Indicates the percentage of storage space allocated to this DB instance / all DB instances that belong to this instance class / all DB instances using this DB engine, which is still available for use.	Percent	A value close to 100% is desired for this measure. If the value of this measure is below 50% consistently, it indicates that the DB instance is not sized with adequate resources. You may want to consider changing the DB instance class of that instance, so that more storage resources are available to it.

6.3.7 Billing Test

AWS Billing and Cost Management is the service that you use to pay your AWS bill, monitor your usage, and budget your costs.

When budgeting costs, this service also provides forecasts of your estimated costs. Using the **Billing test** you can configure thresholds for this estimate for each service you subscribe to and also for a roll-up of estimated charges of all services. The test will then proactively alert you if the estimate is about to exceed your budget, and thus enable you to initiate measures for avoiding cost overruns.

Note:

This test will run for the 'us-east' region only. Since this region stores Amazon CloudWatch metrics for worldwide estimated charges, the *Estimated charges* that this region reports per service will be the consolidated charges for all regions that use that particular service.

Target of the test: Amazon EC2 Region

Agent deploying the test: A remote agent

Output of the test: One set of results for each service subscribed

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
AWS Access Key, AWS Secret Key, Confirm AWS Access Key, Confirm AWS Secret Key	To monitor an Amazon EC2 instance, the eG agent has to be configured with the access key and secret key of a user with a valid AWS account. For this purpose, we recommend that you create a special user on the AWS cloud, obtain the access and secret keys of this user, and configure this test with these keys. The procedure for this has been detailed in the Section **2.3** topic. Make sure you reconfirm the access and secret keys you provide here by retyping it in the corresponding **Confirm** text boxes.
Proxy Host and Proxy Port	In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
Proxy User Name, Proxy Password, and Confirm Password	If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. Then, confirm the password by retyping it in the **CONFIRM PASSWORD** text box. By default, these parameters are set to none, indicating that the proxy sever does not require authentication by default.
Proxy Domain and Proxy Workstation	If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can

Parameter	Description
	be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measures reported by the test:

Measurement	Description	Measurement Unit	Interpretation
Estimated charges:	Indicates the estimated cost of this service for all regions using the service.	USD	Compare the value of this measure across services to know which service you will be spending the most on in the future. You can be notified if cost estimations for a service exceed an acceptable limit, by configuring such a limit as a the maximum threshold for this measure for that service. Based on these alarms, you can set out to change how frequently you actually use that service, so as to reduce related overheads. For the **Total** descriptor, this measure will report the total estimated charges across all services.

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.