Monitoring Ansible Tower

eG Innovations Product Documentation

www.eginnovations.com

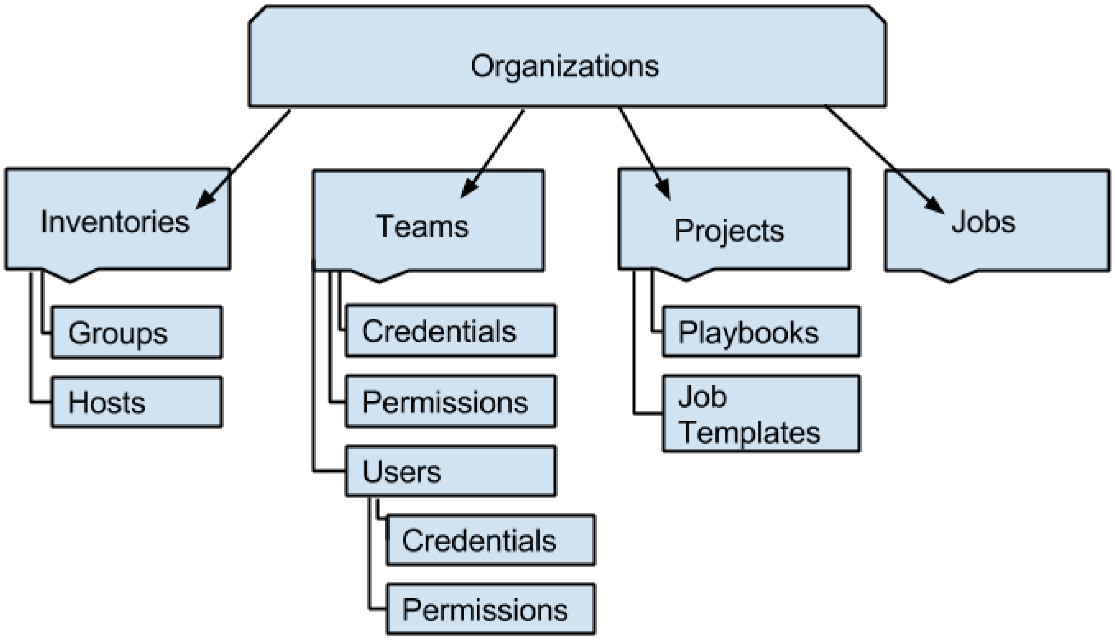
Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

Ansible Tower is Ansible at a more enterprise level, and is designed to be the hub for all of your automation tasks. It is a web-based solution for managing your organization with a very easy user interface that provides a dashboard with all of the state summaries of all the hosts, allows quick deployments, and monitors all configurations. Ansible Tower offers multilevel access delegation and role-based access control to an organization. The organization is nothing but a logical collection of Users, Teams, Projects, and Inventories. It is the highest level in the Ansible Tower object hierarchy. To support multiple clients from one umbrella, you could create an organization for each client and manage multiple teams under that. The following figure gives you a pictorial representation of the organization hierarchy in the Ansible Tower.



a. **Inventories** - Ansible inventories are lists of hosts (nodes) along with their IP addresses, servers, databases etc. which needs to be managed.

   ◦ **Groups** - A group consists of several hosts assigned to a pool that can be conveniently targeted together, as well as given variables that they share in common.

   ◦ **Hosts** - A system managed by Tower, which may include a physical, virtual, cloud-based server, or other device. Typically an operating system instance. Hosts are contained in Inventory. Sometimes referred to as a “node”.

b. **Teams** - A team is a sub-division of an organization with associated users, projects, credentials, and permissions. Teams provide a means to implement role-based access control schemes and delegate responsibilities across the organizations.

  ○ **Credentials** - Authentication details that may be utilized by Ansible Tower to launch jobs against machines, to synchronize with inventory sources, and to import project content from a version control system.

  ○ **Permissions** - The set of privileges assigned to Users and Teams that provide the ability to read, modify, and administer Projects, Inventories, and other Tower objects.

  ○ **Users** - A user is someone with access to the Tower with associated roles and credentials.

c. **Projects** - Tower projects are a logical collection of Ansible Playbooks that are set up with each other based on what they might be doing or which hosts they might interact with.

  ○ **Playbooks** - Playbooks are the language by which Ansible orchestrates, configures, administers, or deploys systems.

  ○ **Job Templates** - The combination of an Ansible playbook and the set of parameters required to launch it.

d. **Jobs** - One of many background tasks launched by Tower, this is usually the instantiation of a Job Template; the launch of an Ansible playbook. Other types of jobs include inventory imports, project synchronizations from source control, or administrative cleanup actions.

With the above-mentioned distributed architecture and a friendly graphical framework for your IT orchestration needs, the Ansible Tower makes your automation life easier in applications/service delivery environments. Enterprises that make money delivering applications/services via the web find that Ansible Tower excels at zeroing downtime during rolling updates, removing IT bottlenecks, automating repetitive tasks, and accelerating the delivery of applications/services to market. The Ansible Tower also helps administrators describe how all of systems in the infrastructure should inter-communicate, organize the teams and manage the operations across the teams to ensure high efficiency in developing and delivering applications/services.

Synchronized and failure-free functioning of the Ansible Tower is therefore a key requirement for the continuous delivery of critical applications/services. To ensure that the Tower is always accessible and is performing to peak capacity, it is necessary to constantly watch over the performance of the Tower, and detect and resolve failure conditions before they cause serious impact. This can be easily achieved using the specialized Ansible Tower monitoring model offered by eG Enterprise.

Chapter 2: How to Monitor Ansible Tower Using eG Enterprise ?

eG Enterprise can monitor Ansible Tower in an agent-based or an agentless manner. In case of the agentless approach, the remote agent used to monitor the Ansible Tower should be deployed on a remote Windows host in the environment.

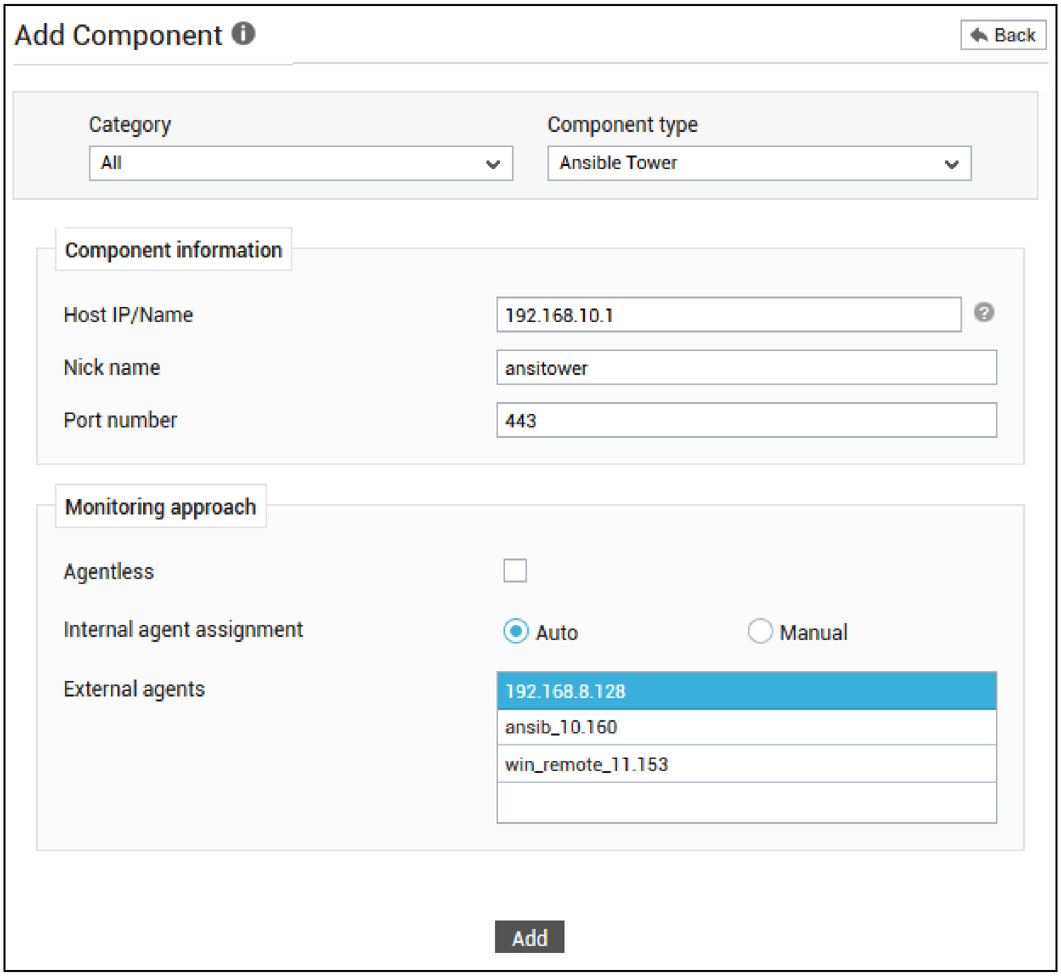
Regardless of the approach (agent-based or agentless), the eG agent makes RESTful API calls to the Ansible Tower to pull the performance metrics of interest. To enable this connection, each test that the eG agent runs on the Ansible Tower should be configured with the credentials of a user who has administrator privileges on the Ansible Tower.

To start monitoring the Ansible Tower, first manage the *Ansible Tower* component using eG admin interface. The steps for achieving it are explained in the sections below.

2.1 Managing Ansible Tower

eG Enterprise cannot automatically discover the Ansible Tower. This implies that you need to manually add the component for monitoring. To manage an Ansible Tower component, do the following:

1. Log into the eG admin interface.
2. Follow the Components -> Add/Modify menu sequence in the **Infrastructure** tile of the **Admin** menu.
3. In the **Components** page that appears next, select *Ansible Tower* as the **Component type**. Then, click the **Add New Component** button. This will invoke Figure 2.1.



Add Component

Category: All

Component type: Ansible Tower

Component information

Host IP/Name: 192.168.10.1

Nick name: ansitower

Port number: 443

Monitoring approach

Agentless: □

Internal agent assignment: ● Auto ○ Manual

External agents: 192.168.8.128
ansib_10.160
win_remote_11.153

Figure 2.1: Adding Ansible Tower in an agent-based manner

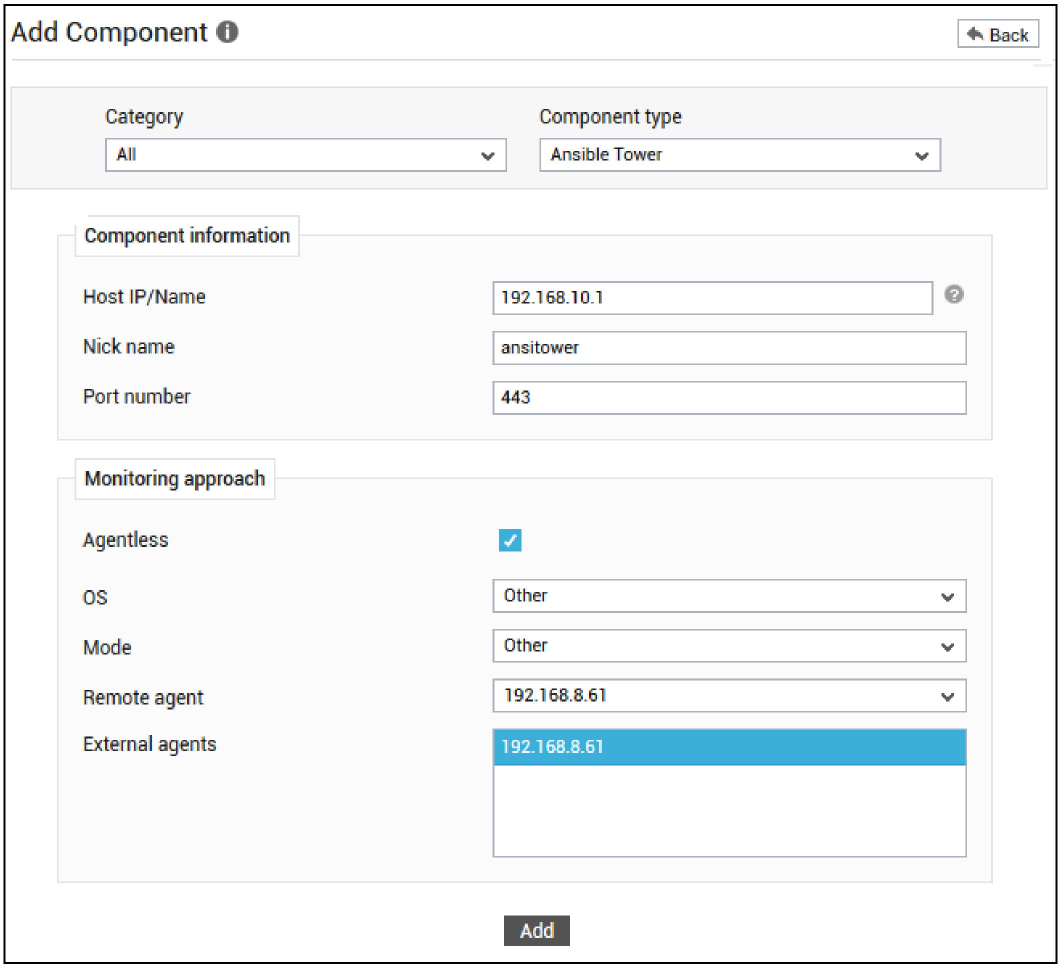
4. Specify the **Host Name** and **Nick name** for the Ansible Tower component.

5. The **Port number** will be set as 443 by default. If the Ansible Tower is listening on a different port in your environment, then override this default setting.

6. In case you are monitoring a Ansible Tower in an agent-based manner, just pick an external agent from an the **External agents** list box and click the **Add** button to add the component for monitoring.

7. On the other hand, if you want to monitor an Ansible Tower in an agentless manner, then do the following:

   - Select the **Agentless** check box in Figure 2.1. This will list the additional options, as shown in Figure 2.2.



Add Component

Category: All

Component type: Ansible Tower

Component information

Host IP/Name: 192.168.10.1

Nick name: ansitower

Port number: 443

Monitoring approach

Agentless: ☑

OS: Other

Mode: Other

Remote agent: 192.168.8.61

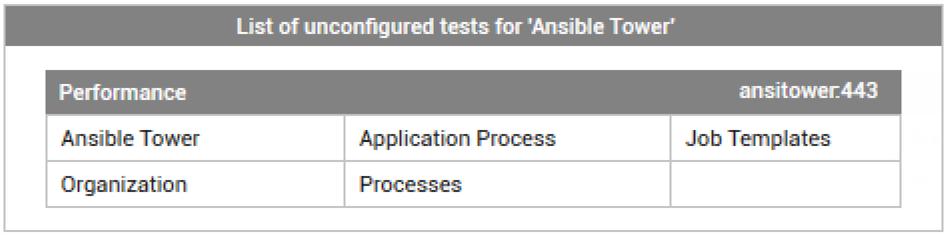
External agents: 192.168.8.61

Figure 2.2: Managing Ansible Tower in an agentless manner

- Next, select **Other** as the **OS**.
- Set the **Mode** to **Other**.
- Select the **Remote agent** that will be monitoring the Ansible Tower. Note that the remote agent you choose should run on a Windows host.
- Choose an external agent for the Ansible Tower by picking an option from the **External agents** list box.
- Finally, click the **Add** button to add the Ansible Tower for monitoring.

2.2 Configuring Tests

After adding the component, try to sign out of the eG admin interface, a **LIST OF UNCONFIGURED TESTS** page will appear, revealing the list of tests mapped to the *Ansible Tower* that require manual configuration:



List of unconfigured tests for 'Ansible Tower'

Performance		ansitower:443
Ansible Tower	Application Process	Job Templates
Organization	Processes	

Figure 2.3: The list of unconfigured tests requiring manual configuration

To know how to configure the **Ansible Tower**, **Job Templates** and **Organizations** tests, refer to **Monitoring Ansible Tower** chapter. Refer to *Monitoring Unix and Windows Servers* document to know how to configure the **Processes** and **Application Process** tests.

Chapter 3: Monitoring Ansible Tower

eG Enterprise offers a dedicated monitoring model for Ansible Tower that periodically monitors the objects such as organizations, inventories, teams, groups, hosts, credentials, etc., in the Tower and reveals the overview. This specialized monitoring model promptly reports the job failures and host failures and issues in the synchronization among the inventories/projects.

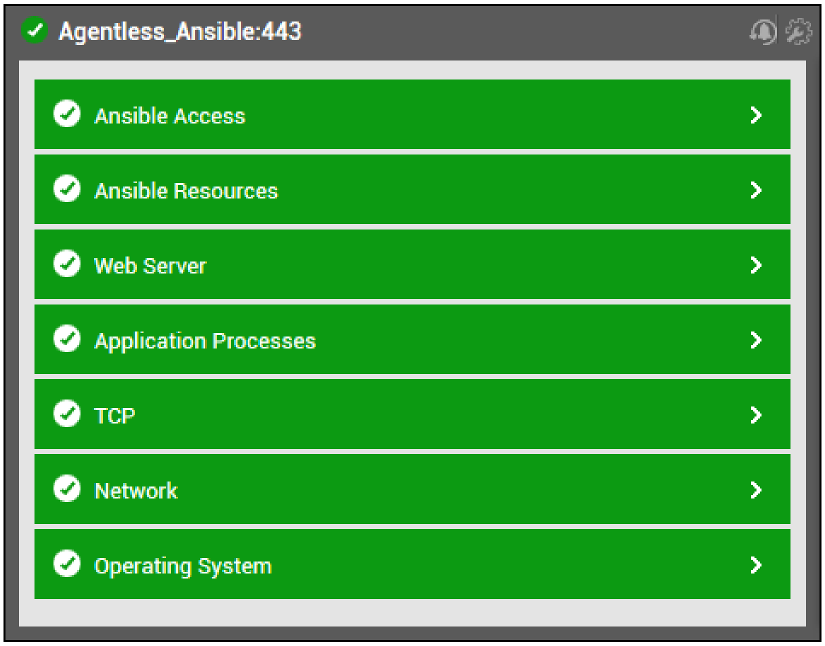


Figure 3.1: Layer model of the Ansible Tower

Each layer of Figure 3.1 is mapped to tests that reveal the details of the inventories, projects, job health, etc. in the Tower environment. Using the metrics reported by these tests, administrators can find quick and accurate answers to certain persistent performance queries, such as the following:

- How many projects and inventories are in the Tower?
- Did any project/inventory syncing process fail in the Tower?
- Is the last ran job on the Tower successfully completed?
- Did the job that ran last in the Tower take more time to complete?
- How many hosts/groups are associated with each job template?
- Did any host/group failure occur while running any job template, if so, which job template?

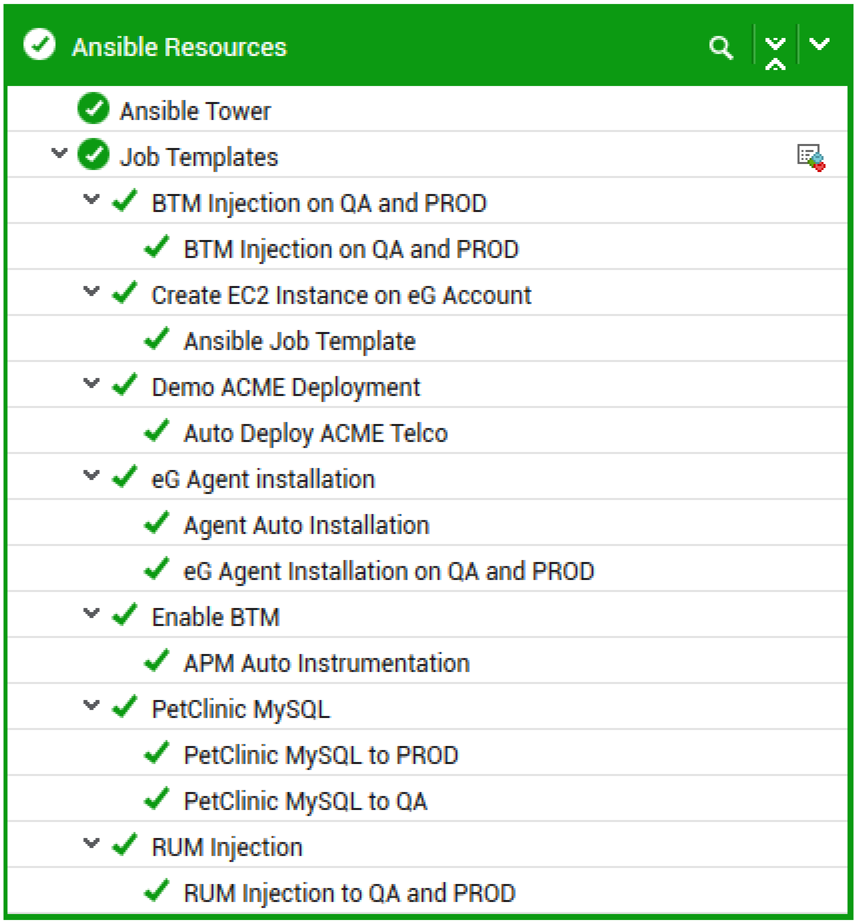
- How many forks are running in parallel when a job is executed?
- How many hosts in the Tower are successfully performing the jobs ?

The four layers at the bottom of Figure 3.1 have been dealt with extensively in the *Monitoring Unix and Windows Servers* document. For the details on the **Web Server** layer, refer to the *Monitoring Microsoft IIS Web Server* document. The section that follows will discuss the **Ansible Access** and **Ansible Access** layers alone.

3.1 Ansible Resource Layer

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

- The number of hosts that are performing the jobs successfully;
- The number of projects and inventories in the Tower;
- The number of project sync failures;
- The number of inventory sync failures;
- The number of jobs that are completed successfully;
- The number of hosts and groups associated with each job template;
- The time taken by the job that was executed during the last measurement period;
- The number of hosts and groups failed during launching the jobs using each job template;



Ansible Resources

- Ansible Tower
- Job Templates
  - BTM Injection on QA and PROD
    - BTM Injection on QA and PROD
  - Create EC2 Instance on eG Account
    - Ansible Job Template
  - Demo ACME Deployment
    - Auto Deploy ACME Telco
  - eG Agent installation
    - Agent Auto Installation
    - eG Agent Installation on QA and PROD
  - Enable BTM
    - APM Auto Instrumentation
  - PetClinic MySQL
    - PetClinic MySQL to PROD
    - PetClinic MySQL to QA
  - RUM Injection
    - RUM Injection to QA and PROD

Figure 3.2: The Ansible Resource Layer

3.1.1 Ansible Tower Test

Red Hat® Ansible® Tower helps you scale IT automation, manage complex deployments and speed productivity. Ansible Tower also helps you keep the inventories and projects in sync. The Tower is centered around the idea of organizing Projects (which run your playbooks via Jobs) and Inventories (which describe the servers on which your playbooks should be run) inside of Organizations.

In Tower automated environments, the jobs play a vital role in performing various operations on the Tower such as updating and synchronizing the inventories and projects, performing system upgrades, updating applications delivered via Tower, etc. Therefore, to ensure peak performance of the Tower, it is important for administrators to continuously track job health of the Tower. If, for any reason, job health deteriorates on the Tower, then the overall performance, data synchronization, reliability and data integrity of the Tower will also deteriorate. To avoid such anomalies, administrators can use the **Ansible Tower** test to proactively detect the job health of the Tower before anything untoward happens.

This test continuously monitors the Ansible tower, and proactively reveals the number of hosts that failed while executing the jobs, and the number of jobs that failed. These statistics help administrators to find out how well/badly the jobs are performed on the Tower. In addition, this test also reports the synchronization failures among the inventories/projects while launching the jobs. This helps administrators to detect the issues in the synchronization, if any, and take remedial actions immediately.

Target of the test : Ansible Tower

Agent deploying the test : A remote agent

Outputs of the test : One set of the results for the Ansible Tower being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed. By default, this is set to 5 minutes.
Host	The IP address of the host for which this test is to be configured.
Port	The port at which the specified host listens. By default, this is *443*
Username and Password	The eG agent makes REST API calls for pulling out performance metrics from the Ansible Tower. For this purpose, the eG agent should be allowed to connect to Ansible Tower's REST API. To enable this connection, administrators need to configure the valid credentials of a user who has administrator privileges on the Ansible Tower against these parameters.
Confirm Password	Confirm the password by retyping it here.
SSL	By default, the Ansible Tower is SSL-enabled. Accordingly, the SSL flag is set to **True** by default. This indicates that the eG agent will communicate with the Ansible Tower via HTTPS, by default.
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

Parameter	Description
	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 hosts	Indicates the total number of hosts in the Tower.	Number	
Successful hosts	Indicates the number of hosts that are successfully performing the jobs.	Number	
Failed hosts	Indicates the number of hosts that failed to perform the jobs.	Number	
Jobs health	Indicates the percentage of jobs that is successfully performed on the Tower.	Percent	Ideally, the value of this measure is should be high.
Total jobs	Indicates the total number of jobs launched on the Tower.	Number	This measure is good indicators of the workload on the Tower.
Successful jobs	Indicates the number of jobs that are completed successfully on the Tower.	Number	A high value is desired for this measure.
Failed jobs	Indicates the number of jobs that failed on the	Number	The value of this measure should be zero. A non-zero value indicates that

Measurement	Description	Measurement Unit	Interpretation
	Tower.		one/more jobs failed; this is a cause for concern and requires investigation.
Total projects	Indicates the total number of projects in the Tower.	Number	
Project sync failures	Indicates the number of times the project syncing process failed during job execution.	Number	Ideally, the value of this measure should be zero. A non-zero value for this measure may lead to serious synchronization issues, and is a cause for concern.
Credentials	Indicates the number of credentials available on the Tower.	Number	
Inventories	Indicates the total number of inventories in the Tower.	Number	
Inventory sync failures	Indicates the number of times the inventory syncing process failed during job execution.	Number	Ideally, the value of this measure should be zero. A non-zero value for this measure is a cause for concern.

3.1.2 Job Templates Test

A job template is a definition and set of parameters for running an Ansible job. Job templates are useful to execute the same job many times without having to define the parameters repeatedly. The job templates also encourage the reuse of Ansible playbook content and collaboration between teams. The job templates can be associated with multiple hosts and groups in the Ansible Tower. Each job template enables administrators to run important jobs such as playbook runs, cloud inventory updates, source control updates, etc. across the hosts and groups simultaneously. The predefined job templates save time and reduce workload of administrators. Periodically, administrators should check whether the jobs executed using each template are completed successfully, identify failures and delays (if any), investigate the reasons for the failure and fix them, so that such job failures do not adversely impact the performance of the Tower. The **Jobs Templates** test helps administrators rapidly capture job failures and promptly initiate remedial actions.

This test auto-discovers the job templates in each project in the Ansible Tower, and reports the total number of jobs that was launched using each job template in a particular project. In the process, this test reports the status of a job that ran last in each project and time taken for each job to complete. This enables administrators to identify a job template that causes more job failures and find out if any of the jobs took longer than usual to complete. In addition, this test alerts administrators to the number of hosts and groups that failed during jobs execution using each job template. These statistics help administrators to find out where the real problem (i.e. is it with job template or with host/group?) lies and resolve the same quickly.

Target of the test : Ansible Tower

Agent deploying the test : A remote agent

Outputs of the test : One set of the results for each *project:job template* pair in the Ansible Tower being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed. By default, this is set to 5 minutes.
Host	The IP address of the host for which this test is to be configured.
Port	The port at which the specified host listens. By default, this is *NULL*.
Username and Password	The eG agent makes REST API calls for pulling out performance metrics from the Ansible Tower. For this purpose, the eG agent should be allowed to connect to Ansible Tower's REST API. To enable this connection, administrators need to configure the valid credentials of a user who has administrator privileges on the Ansible Tower against the Username and Password parameters.
Confirm Password	Confirm the password of the specified user by retyping it here.
Excluded Projects	Specify the comma-separated list of projects that you want the eG agent to exclude from monitoring against this parameter. By default, this is set to "none" indicating that this test will monitor all the projects in the Ansible tower.
Excluded Job Templates	Specify the comma-separated list of Job Templates that you want the eG agent to exclude from monitoring in this text box. By default, this is set to "none" indicating that this test will monitor all the Job Templates in the Ansible tower.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Total jobs</td><td>Indicates the total number of jobs launched using this job template.</td><td>Number</td><td></td></tr>
<tr><td>Last job status</td><td>Indicates whether the job that was last executed using this job template is successful or not.</td><td></td><td>The values that this measure can report and its corresponding numeric equivalents are listed in the table below:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Failed</td><td>0</td></tr>
<tr><td>Success</td><td>1</td></tr>
</table>
<b>Note:</b>
By default, this measure reports the <b>Measure Value</b>s listed in the table above to indicate the job is successful or not. In the graph of this measure however, the same is represented using the corresponding numeric equivalents only.
Comparing the output reported for this measure across the job templates will reveal the job template that is prone to more failures.</td></tr>
<tr><td>Last job elapsed time</td><td>Indicates the time taken by the job that was last executed using this job template.</td><td>Seconds</td><td></td></tr>
<tr><td>Forks</td><td>Indicates the number of Forks i.e., hosts that are running in parallel while launching the jobs using this job template.</td><td>Number</td><td>Ansible talks to remote hosts in parallel, and the level of parallelism can be set either by passing -forks, or editing the default in a configuration file. The default is a very conservative 5 forks, though if you have a lot of RAM, you can easily set this to a value like 50</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			for increased parallelism.
Total hosts	Indicates the total number of hosts to which this job template has been associated.	Number	
Host active failures	Indicates the number of hosts that failed while running the jobs using this job template.	Number	Host failures may occur while running jobs due to missing of the host details such as new/changed IP addresses, variables, updates, etc., in the playbooks of the job template. Ideally, the value of this measure should be zero or very low. A high value for this measure is a cause for concern. Compare the value of this measure across the job templates to find out which job template is causing more host failures. This way, administrators are accurately pinpointed to the faulty job template and enabled to take corrective measures quickly.
Total groups	Indicates the total number of groups to which this job template has been associated.	Number	A group consists of several hosts assigned to a pool that can be conveniently targeted together, and also given variables that they share in common.
Groups active failures	Indicates the number of groups that failed while running the jobs using this job template.	Number	Group failures during job execution occur due to many reasons. For instance, lets say, if the details about the new group or any host in the group are not updated in the playbooks of a job template that is to be executed on the inventory, then the job will fail. This in turn will cause the group failures during job launches. Ideally, the value of this measure should be zero or very low. Compare the value of this measure across the

Measurement	Description	Measurement Unit	Interpretation
			job templates to find out which job template is causing more failures.

3.2 Ansible Access Layer

The test mapped to this layer reveals the total number of users, admins, teams, job templates, inventories and projects in each organization.

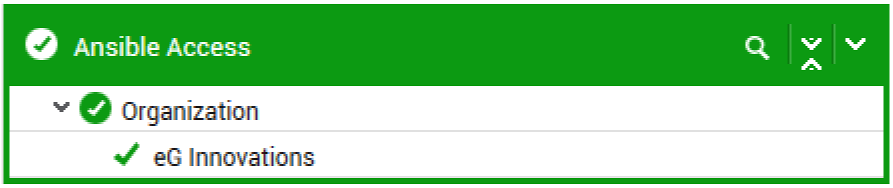


Figure 3.3: The Ansible Access layer

3.2.1 Organizations Test

This test auto-discovers the organizations in the Ansible Tower, and reports the total number of users, admin users, teams, inventories, job templates and projects in each organization. Use the detailed diagnosis provided by this test to find out which resource of the organization is currently utilized by more number of users and which user consumes more resources.

Target of the test : Ansible Tower

Agent deploying the test : A remote agent

Outputs of the test : One set of the results for every organization in the Ansible Tower that is being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed. By default, this is set to 5 minutes.
Host	The IP address of the host for which this test is to be configured.
Port	The port at which the specified host listens. By default, this is *NULL*
Username and Password	The eG agent makes REST API calls for pulling out performance metrics from the Ansible Tower. For this purpose, the eG agent should be allowed to connect to Ansible

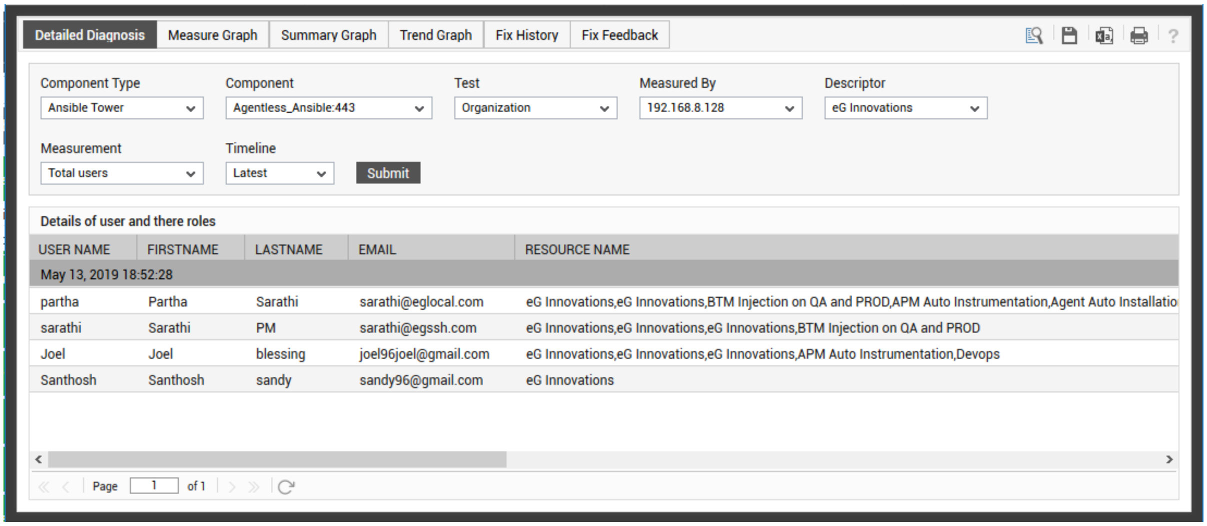
Parameter	Description
	Tower's REST API. To enable this connection, administrators need to configure the valid credentials of a user who has administrator privileges on the Ansible Tower against the Username and Password parameters.
Confirm Password	Confirm the password by retyping it here.
DD Frequency	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
Total users	Indicates the total number of users in this organization.	Number	Compare the value of this measure across the organizations to find out an organization that is widely used by more number of users. The detailed diagnosis of this measure reveals the details of the users such as user names, first and last names and mail IDs of the users and the names of resources that the users accessed.

Measurement	Description	Measurement Unit	Interpretation
Total teams	Indicates the total number of teams in this organization.	Number	The detailed diagnosis of this measure lists the name and description of each team, the name and type of each resource used by each team, and the role assigned to each team.
Total admins	Indicates the total number of admin users in this organization.	Number	An admin user is a Tower user with the rights to modify the Organization's membership and settings, including creating new users and projects within that organization. An organization admin can also grant permissions to other users within the organization. The detailed diagnosis of this measure reveals the details of each admin user such as user name, first and last names and mail ID and the name of resource that each admin user accessed.
Total inventories	Indicates the total number of inventories in this Organization.	Number	
Total projects	Indicates the total number of projects in this Organization.	Number	
Total job templates	Indicates the total number of job templates created in this Organization.	Number	

The detailed diagnosis of the *Total users* measure reveals the user names, first and last names and mail IDs of the users and the names of resources that the users accessed.



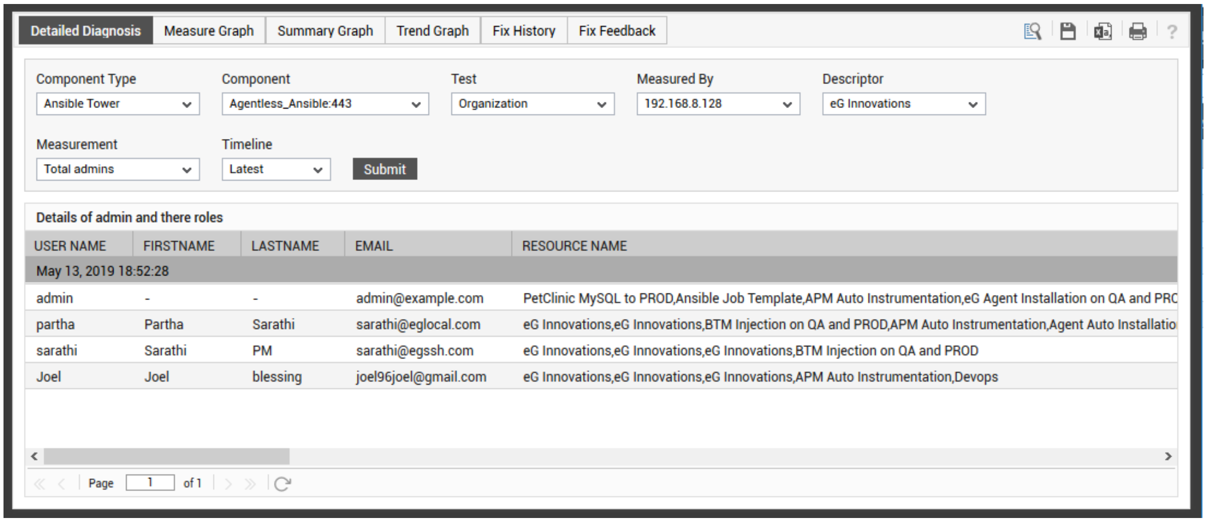
Component Type: Ansible Tower
Component: Agentless_Ansible:443
Test: Organization
Measured By: 192.168.8.128
Descriptor: eG Innovations
Measurement: Total users
Timeline: Latest

Details of user and there roles

USER NAME	FIRSTNAME	LASTNAME	EMAIL	RESOURCE NAME
May 13, 2019 18:52:28				
partha	Partha	Sarathi	sarathi@eglocal.com	eG Innovations,eG Innovations,BTM Injection on QA and PROD,APM Auto Instrumentation,Agent Auto Installatio
sarathi	Sarathi	PM	sarathi@egssh.com	eG Innovations,eG Innovations,eG Innovations,BTM Injection on QA and PROD
Joel	Joel	blessing	joel96joel@gmail.com	eG Innovations,eG Innovations,eG Innovations,APM Auto Instrumentation,Devops
Santhosh	Santhosh	sandy	sandy96@gmail.com	eG Innovations

Figure 3.4: The detailed diagnosis of the Total users measure

The detailed diagnosis of the *Total admins* measure reveals the user name, first and last names and mail ID of each admin user and the resources that the admin users accessed.



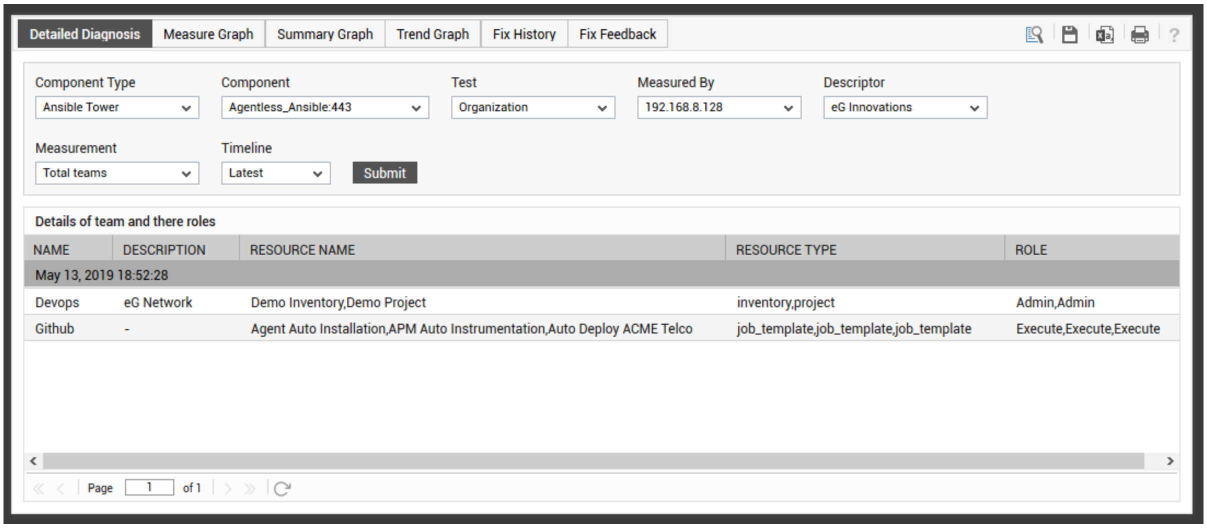
Component Type: Ansible Tower
Component: Agentless_Ansible:443
Test: Organization
Measured By: 192.168.8.128
Descriptor: eG Innovations
Measurement: Total admins
Timeline: Latest

Details of admin and there roles

USER NAME	FIRSTNAME	LASTNAME	EMAIL	RESOURCE NAME
May 13, 2019 18:52:28				
admin	-	-	admin@example.com	PetClinic MySQL to PROD,Ansible Job Template,APM Auto Instrumentation,eG Agent Installation on QA and PRC
partha	Partha	Sarathi	sarathi@eglocal.com	eG Innovations,eG Innovations,BTM Injection on QA and PROD,APM Auto Instrumentation,Agent Auto Installatio
sarathi	Sarathi	PM	sarathi@egssh.com	eG Innovations,eG Innovations,eG Innovations,BTM Injection on QA and PROD
Joel	Joel	blessing	joel96joel@gmail.com	eG Innovations,eG Innovations,eG Innovations,APM Auto Instrumentation,Devops

Figure 3.5: The detailed diagnosis of the Total admins measure

The detailed diagnosis of the *Total teams* measure lists the name and description of each team, the name and type of each resource used by each team, and the role assigned to each team.



Component Type: Ansible Tower

Component: Agentless_Ansible:443

Test: Organization

Measured By: 192.168.8.128

Descriptor: eG Innovations

Measurement: Total teams

Timeline: Latest

Details of team and there roles

NAME	DESCRIPTION	RESOURCE NAME	RESOURCE TYPE	ROLE
May 13, 2019 18:52:28				
Devops	eG Network	Demo Inventory,Demo Project	inventory,project	Admin,Admin
Github	-	Agent Auto Installation,APM Auto Instrumentation,Auto Deploy ACME Telco	job_template,job_template,job_template	Execute,Execute,Execute

Figure 3.6: The detailed diagnosis of the Total teams measure

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.