Monitoring Atlassian Confluence

eG Innovations Product Documentation

![eG Total Performance Visibility logo]

Table of Contents

Table of Figures

Chapter 1: Introduction

Confluence is a team collaboration software from Atlassian, using which you can create, organize, and discuss work with your team. It enables project collaboration, eases information creation, maintenance, and dissemination, and thus helps your teams work better. For instance, with the help of Confluence, you can create pages using which you can add/import and manage content, email these pages to share content with your team, and even notify team members by email whenever content changes.

If say, such emails are not delivered in time, Project teams will not be able to take key decisions, promptly; this can significantly impact user productivity. Likewise, if users are unable to search a page and locate information they need quickly, user experience with Confluence is bound to suffer. Besides the above, since Confluence is a Java- based software, issues in the underlying JVM can also adversely affect the overall performance of Confluence. To avoid this, administrators should continuously monitor the availability, responsiveness, and operational health of Atlassian Confluence.

eG Enterprise provides a 100% web-based monitoring model for Atlassian Confluence. Each layer of this model is mapped to tests that monitor the email activity of Confluence, its searching and indexing operation, and its interactions with the database. In the process, these tests proactively alert administrators to email delivery bottlenecks, latencies in searching and indexing, and connection failures with the eG database, thus enabling administrators to take quick action against current / potential anomalies. Additionally, since Confluence is a Java-based application, this model also reports critical statistics pertaining to the health of the JVM in which Confluence runs; this way, resource contentions and abnormal thread activity at the JVM-level are brought to light. Moreover, you can, if required, configure eG's Java Business Transaction Monitor (BTM) to track every transaction to Confluence. The eG Java BTM is capable of tracing the path of a transaction, measuring overall transaction responsiveness and the time taken at each point cut, and alerting administrators to slow transactions and the root-cause of the slowness.

Chapter 2: How Does eG Enterprise Monitor Atlassian Confluence?

eG Enterprise is capable of monitoring Atlassian Confluence in an agent-based / agentless manner.

The eG agent, deployed either on the Confluence host or on a remote Windows host in the environment (depending upon the monitoring approach – whether agent-based or agentless), can be configured to connect to the JRE used by Confluence and pull out metrics of interest, using JMX (Java Management Extension). For this, you need to enable JMX support for the JRE of Atlassian Confluence. To know how to enable JMX support for the JRE of Atlassian Confluence, refer to the Enabling JMX Support for JRE topic.

Moreover, to trace the path of each transaction to Atlassian Confluence in real-time and measure its responsiveness, the eG Java Business Transaction Monitor (BTM) can be used. For this, you need to first deploy eG Java BTM on Atlassian Confluence. The steps for this are detailed in the Enabling Java Business Transaction Monitoring for Atlassian Confluence topic.

2.1 Enabling JMX Support for JRE

To enable JMX support for Confluence's JRE, do the following:

1. Open the catalina.bat file in the <<ATLASSIAN_HOME>>\confluence\bin directory (on Windows; on Unix, this will be <<ATLASSIAN_HOME>>/confluence/bin directory).

2. Look for the entry that begins with *set "JAVA_OPTS=*

3. Under that entry, specify the following:

```
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=<<JMX_REMOTE_PORT>> -
Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false "
```

4. Make sure that you replace the text in Bold in the entry above with a valid JMX remote port.

5. Then, save the file.

6. Finally, restart the Confluence server. For this, first go to the command prompt and switch to the <<ATLASSIAN_ CONFLUENCE_ HOME>>\confluence\bin directory. Then, run the following command at the prompt:

   Start-confluence.bat /fg

2.2 Enabling Java Business Transaction Monitoring for Atlassian Confluence

The steps for BTM- enabling Atlassian Confluence will differ based on where the eG agent monitoring Confluence has been deployed - whether on the Confluence server, or on a remote host.

2.2.1 Agent- based Approach to Deploying eG Java BTM on Atlassian Confluence

If Atlassian Confluence is running on Windows, and the eG agent monitoring the server has been deployed on that server itself, then follow the steps below to BTM-enable the Confluence server:

1. Manage Atlassian Confluence using the eG administrative interface. When managing, make a note of the **Nick name** and **Port number** that you provide.

2. If multiple Confluence server instances are operating on a single host, and you want to BTM-enable all the instances, then you will have to manage each instance as a separate Atlassian Confluence server using the eG administrative interface. When doing so, make a note of the **Nick name** and **Port number** using which you managed each instance.

3. In the **<EG_AGENT_INSTALL_DIR>\lib\btm** directory, you will find the following files:

   - **eg_btm.jar**

   - **btmLogging.props**

   - **btmOther.props**

   - **exclude.props**

4. Next, create a new directory under the **<EG_AGENT_INSTALL_DIR>\lib\btm**. Take care to name this directory in the following format: *<Managed_Component_NickName>_<Managed_Component_Port>*. For instance, if you have managed the Confluence server using the nick name *Conflu1* and the port number *8090*, the new directory under the **btm** directory should be named as *Conflu1_8090*.

5. If you have managed multiple Confluence server instances running on a single host, then you will have to create multiple sub- directories under the **btm** directory- one each for every instance. Each of these sub- directories should be named after the **Nick name** and **Port number** using which the corresponding instance has been managed in eG.

6. Once the new directory is created, copy the following files from the **btm** directory to the new directory. If multiple directories have been created as described by step 5 above, then the files

should be copied to each of those directories:

- **btmLogging.props**

- **btmOther.props**

- **exclude.props**

  - Next, edit the **btmOther.props** file. You will find the following lines in the file:

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  Below  property  is  BTM  Server  Socket  Port,  through  which  eG  Agent
Communicates
# Restart is required, if any changes in this property
# Default port is "13931"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#
BTM_Port=13931
#
```

By default, the **BTMPort** parameter is set to 13931. If you want to enable eG Java BTM on a different port, then specify the same here. In this case, when configuring the **Java Business Transactions** test or the **Key Java Business Transactions** test for the Confluence server, make sure you configure the **BTM** port parameter of the test with this port number.

Note:

When BTM-enabling multiple instances on the same server, make sure you configure a different **BTM Port** for each instance.

Also, by default, the **Designated_Agent** parameter will be empty; do not disturb this default setting. In this case therefore, the eG Java BTM will treat the host from which the very first 'measure request' comes in as the **Designated_Agent**.

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Below property is used to specify IP address of eG Agent which collectes BTM
Data.
# Default is None
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
#

Designated_Agent=

#
```

Note:

In case a specific **Designated_Agent** is not provided, and the eG Java BTM treats the host from which the very first 'measure request' comes in as the **Designated_Agent**, then if such a **Designated_Agent** is stopped or uninstalled for any reason, the eG Java BTM will wait for a maximum of 10 measure periods for that 'deemed' **Designated_ Agent** to request for metrics. If no requests come in for 10 consecutive measure periods, then the eG Java BTM will begin responding to 'measure requests' coming in from any other eG agent.

7. Then, you need to configure the Confluence server with the path to the **eg_btm.jar** and **.props** files. For this, first open the **catalina.bat** file in the <<ATLASSIAN_HOME>>\confluence\bin directory .

8. Then, in that file, look for the entry that begins with *set "JAVA_OPTS=*

9. Against that entry, specify the following:

```
set "JAVA_OPTS=%JAVA_OPTS% -javaagent:<<EG_INSTALL_DIR>>\lib\btm\eg_btm.jar -DEG_
PROPS_HOME=<<PATH OF THE LOCAL FOLDER CONTAINING THE .PROPS FILES>>"
```

For instance, if the .props files had been copied to the **<EG_ AGENT_ INSTALL_ DIR>\lib\btm\conflu1_8090** directory, the above specification will be:

```
set "JAVA_OPTS=%JAVA_OPTS% -javaagent:<<EG_INSTALL_DIR>>\lib\btm\eg_btm.jar -DEG_
PROPS_HOME=<EG_AGENT_INSTALL_DIR>\lib\btm\conflu1_8090"
```

10. Next, add the package com.eg in the boot delegation framework. For this, you need to add an entry to this effect in the **catalina.bat** file. Figure 2.1 depicts how this can be done.

Figure 2.1: Adding the com.eg package in the boot delegation framework

11. Then, save the file.

12. Finally, restart the Confluence server. For this, first go to the command prompt and switch to the <<ATLASSIAN_ CONFLUENCE_ HOME>>\confluence\bin directory. Then, run the following command at the prompt:

    Start-confluence.bat /fg

13. Where multiple Confluence server instances on a host are to be monitored, Repeat 7 to 12 for each of the server instances.

If a Confluence server is running on Unix, and the eG agent monitoring the server has been deployed on that server itself, then follow the steps below to BTM-enable that server:

1. Manage the Confluence server using the eG administrative interface. When managing, make a note of the **Nick name** and **Port number** that you provide.

2. If multiple Confluence server instances are operating on a single host, and you want to BTM-enable all the instances, then you will have to manage each instance as a separate Atlassian Confluence server using the eG administrative interface. When doing so, make a note of the **Nick name** and **Port number** using which you managed each instance.

3. In the **/opt/egurkha/lib/btm** directory, you will find the following files:

   - **eg_btm.jar**

   - **btmLogging.props**

   - **btmOther.props**

   - **exclude.props**

4. Next, create a new directory under the **/opt/egurkha/lib/btm**. Take care to name this directory in the following format: *<Managed_Component_NickName>_<Managed_Component_Port>*. For instance, if you have managed the Confluence server using the nick name *Conflu* and the port number *8090*, the new directory under the **btm** directory should be named as *Conflu_ 8090*.

5. If you have managed multiple Confluence server instances running on a single host, then you will have to create multiple sub-directories under the **btm** directory- one each for every instance. Each of these sub-directories should be named after the **Nick name** and **Port number** using which the corresponding instance has been managed in eG.

6. Once the new directory is created, copy the following files from the **btm** directory to the new

directory. If multiple directories have been created as described in step 5 above, then the following files should be copied to all directories:

- **btmLogging.props**

- **btmOther.props**

- **exclude.props**

7. Next, edit the **btmOther.props** file. You will find the following lines in the file:

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Below property is BTM Server Socket Port, through which eG Agent Communicates
# Restart is required, if any changes in this property
# Default port is "13931"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#
BTM_Port=13931
#
```

By default, the **BTMPort** parameter is set to 13931. If you want to enable eG Java BTM on a different port, then specify the same here. In this case, when configuring the **Java Business Transactions** test or the **Key Java Business Transactions** test for that Confluence server, make sure you configure the **BTM** port parameter of the test with this port number.

Note:

When BTM‑enabling multiple instances on the same server, make sure you configure a different **BTM Port** for each instance.

Also, by default, the **Designated_Agent** parameter will be empty; do not disturb this default setting. In this case therefore, the eG Java BTM will treat the host from which the very first 'measure request' comes in as the **Designated_Agent**.

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Below property is used to specify IP address of eG Agent which collectes BTM Data.
# Default is None
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#
Designated_Agent=
```

```
#
```

Note:

In case a specific **Designated_Agent** is not provided, and the eG Java BTM treats the host from which the very first 'measure request' comes in as the **Designated_Agent**, then if such a **Designated_Agent** is stopped or uninstalled for any reason, the eG Java BTM will wait for a maximum of 10 measure periods for that 'deemed' **Designated_Agent** to request for metrics. If no requests come in for 10 consecutive measure periods, then the eG Java BTM will begin responding to 'measure requests' coming in from any other eG agent.

8. Then, you need to configure the Confluence server with the path to the **eg_btm.jar** and **.props** files. For this, first open the **catalina.sh** file in the opt/atlassian/confluence/bin directory.

9. Next, look for the *JAVA_OPTS=* entry in the file. Under this entry, insert the following lines to BTM-enable the server.

```
JAVA_OPTS="$JAVA_OPTS -javaagent:<<Full_path_to_the_eg_btm.jar_file>> -DEG_PROPS_
HOME=<<Path_to_the_.props_file>>
```

For instance, if the .props file had been copied to the **Conflu_8090** folder within the **/opt/egurkha/lib/btm** folder, then your specification will be as follows:

```
JAVA_OPTS="$JAVA_OPTS -javaagent:/opt/egurkha/lib/btm/eg_btm.jar -DEG_PROPS_
HOME=/opt/egurkha/lib/btm/Conflu_8090
```

```
JAVA_OPTS="$JAVA_OPTS $JSSE_OPTS"

# Register custom URL handlers
# Do this here so custom URL handles (specifically 'war:...') can be used in the security policy
JAVA_OPTS="$JAVA_OPTS -Djava.protocol.handler.pkgs=org.apache.catalina.webresources"
JAVA_OPTS="$JAVA_OPTS -javaagent:/opt/egurkha/lib/btm/eg_btm.jar -DEG_PROPS_HOME=/opt/egurkha/lib/btm/Conflu_8090
```

Figure 2.2: Editing the catalina.sh file of a Confluence server on Linux to BTM-enable the server

10. Finally, save the file.

11. Next, add the package com.eg in the boot delegation framework. For this, you need to add an entry to this effect in the **catalina.sh** file. Figure 2.3 depicts how this can be done.

```
250
251  # Register custom URL handlers
252  # Do this here so custom URL handles (specifically 'war:...') can be used in the security policy
253  JAVA_OPTS="$JAVA_OPTS -Djava.protocol.handler.pkgs=org.apache.catalina.webresources"
254  JAVA_OPTS="$JAVA_OPTS -javaagent:/opt/egurkha/lib/btm/eg_btm.jar -DEG_PROPS_HOME=/opt/egurkha/lib/btm/Conflu_8090
     -Datlassian.org.osgi.framework.bootdelegation=META-INF.services,com.yourkit,com.eg.*,com.jprofiler,com.jprofiler.*,org.apache.xerces,
     n.*,sun.*,com.sun.jndi,com.icl.saxon,com.icl.saxon.*,javax.servlet,javax.servlet.*,com.sun.xml.bind.*"
```

Figure 2.3: Adding the com.eg package in the boot delegation framework

12. Then, save the file.

13. Finally, restart the Confluence server. For this, first go to the command prompt and switch to the /opt/atlassian/confluence/bin directory. Then, run the following command at the prompt:

    Start-confluence.sh /fg

14. Where multiple Confluence server instances on a host are to be monitored, Repeat 7 to 13 for each of the server instances.

2.2.2 Agentless Approach to Deploying eG Java BTM on an Atlassian Confluence Server

If Atlassian Confluence is running on Windows, and the eG agent monitoring the server has been deployed on a remote host in the environment, then follow the steps below to BTM-enable that Confluence server:

1. Manage the Atlassian Confluence server as a separate component using the eG administrative interface. When managing, make a note of the **Nick name** and **Port number** that you provide.

2. If multiple Confluence instances are operating on a single node, and you want to monitor each of those instances, then you will have to manage each instance as a separate Atlassian Confluence server using the eG administrative interface. When doing so, make a note of the **Nick name** and **Port number** using which you managed each instance.

3. In the **<EG_AGENT_INSTALL_DIR>\lib\btm** directory on the eG agent host, you will find the following files:

   - **eg_btm.jar**
   - **btmLogging.props**
   - **btmOther.props**
   - **exclude.props**

4. Next, log into the Confluence server that is being monitored.

5. Create a new directory named, say **btm**, in any location on that server.

6. Under this directory, create a sub-directory. Take care to name this directory in the following format: *<Managed_Component_NickName>_<Managed_Component_Port>*. For instance, if you have managed the Confluence server using the nick name *conflu1* and the port number *8090*, the sub-directory should be named as *conflu1_8090*.

7. If you have managed multiple instances of the Confluence server, then you will have to create multiple sub-directories - one each for every instance. Each of these sub-directories should be named after the *Nick name* and *port number* using which the corresponding instance has been managed in eG.

8. Once the new sub-directory is created, copy all the files from the **btm** directory of the remote agent to the sub-directory on the Confluence server. Where multiple sub-directories have been created, you will have to copy the files to each of those directories.

9. Next, edit the **btmOther.props** file. You will find the following lines in the file:

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# Below property is BTM Server Socket Port, through which eG Agent Communicates

# Restart is required, if any changes in this property

# Default port is "13931"

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#

BTM_Port=13931

#
```

10. By default, the **BTM_Port** parameter is set to 13931. If you want to enable eG Java BTM on a different port, then specify the same here. In this case, when configuring the **Java Business Transactions** test or the **Key Java Business Transactions test** for the Confluence server, make sure you configure the **BTM PORT** parameter of the test with this port number.

Note:

When BTM-enabling multiple instances on the same server, make sure you configure a different **BTM Port** for each instance.

Also, by default, the **Designated_Agent** parameter will be empty; do not disturb this default setting. In this case therefore, the eG Java BTM will treat the host from which the very first 'measure request' comes in as the **Designated_Agent**.

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Below property is used to specify IP address of eG Agent which collectes BTM Data.

# Default is None

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#
```

```
Designated_Agent=
```

```
#
```

Note:

In case a specific **Designated_Agent** is not provided, and the eG Java BTM treats the host from which the very first 'measure request' comes in as the **Designated_Agent**, then if such a **Designated_Agent** is stopped or uninstalled for any reason, the eG Java BTM will wait for a maximum of 10 measure periods for that 'deemed' **Designated_Agent** to request for metrics. If no requests come in for 10 consecutive measure periods, then the eG Java BTM will begin responding to 'measure requests' coming in from any other eG agent.

11. Finally, save the **btmOther.props** file.

12. Then, you need to configure the Confluence server with the path to the **eg_btm.jar** and **.props** files. For this, first open the **catalina.bat** file in the <<ATLASSIAN_HOME>>\confluence\bin directory .

13. Then, in that file, look for the entry that begins with *set "JAVA_OPTS=*

14. Against that entry, specify the following:

```
set "JAVA_OPTS=%JAVA_OPTS% -javaagent:<<PATH TO THE EG_BTM.JAR FILE>> -DEG_PROPS_
HOME=<<PATH OF THE LOCAL FOLDER CONTAINING THE .PROPS FILE>>"
```

For instance, if the .props files had been copied to the **E:\btm\conflu1_8090** directory, the above specification will be:

```
set "JAVA_OPTS=%JAVA_OPTS% -javaagent:E:\btm\conflu1_8090\eg_btm.jar -DEG_PROPS_
HOME=E:\btm\conflu1_8090"
```

15. Next, add the package com.eg in the boot delegation framework. For this, you need to add an entry to this effect in the **catalina.bat** file. Figure 2.4 depicts how this can be done.

Figure 2.4: Adding the com.eg package in the boot delegation framework

16. Then, save the file.

17. Finally, restart the Confluence server. For this, first go to the command prompt and switch to the <<ATLASSIAN_ CONFLUENCE_ HOME>>\confluence\bin directory. Then, run the following command at the prompt:

   Start-confluence.bat /fg

18. Where multiple Confluence server instances on a host are to be monitored, Repeat 7 to 17 for each of the server instances.

If an Atlassian Confluence server is running on Unix, and the eG agent monitoring the server has been deployed on a remote host in the environment, then follow the steps below to BTM-enable that Confluence server:

1. Manage the Atlassian Confluence server using the eG administrative interface. When managing, make a note of the **Nick name** and **Port number** that you provide.

2. If multiple Confluence server instances are operating on a single host, and you want to BTM-enable all the instances, then you will have to manage each instance as a separate Atlassian Confluence server using the eG administrative interface. When doing so, make a note of the **Nick name** and **Port number** using which you managed each instance.

3. In the **<EG_AGENT_INSTALL_DIR>\lib\btm** directory on the eG agent host, you will find the following files:

   - **eg_btm.jar**

   - **btmLogging.props**

   - **btmOther.props**

   - **exclude.props**

4. Next, log into the Confluence server that is being monitored.

5. Create a new directory named, say **btm**, in any location on that server.

6. Under this directory, create a sub-directory. Take care to name this directory in the following format: *<Managed_Component_NickName>_<Managed_Component_Port>*. For instance, if you have managed the Confluence server using the nick name *Conflu1* and the port number *8090*, the sub-directory should be named as *Conflu1_8090*.

7. If you have managed multiple instances of the Confluence server, then you will have to create multiple sub-directories - one each for every instance. Each of these sub-directories should be named after the *Nick name* and *port number* using which the corresponding instance has been managed in eG.

8. Once the new sub-directory is created, copy all the files from the **btm** directory of the remote agent to the sub-directory on the Confluence server. Where multiple sub-directories have been created, you will have to copy the files to each of those directories.

9. Next, edit the **btmOther.props** file. You will find the following lines in the file:

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# Below property is BTM Server Socket Port, through which eG Agent Communicates

# Restart is required, if any changes in this property

# Default port is "13931"

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#

BTM_Port=13931

#
```

By default, the **BTM_Port** parameter is set to 13931. If you want to enable eG Java BTM on a different port, then specify the same here. In this case, when configuring the **Java Business Transactions** test or the **Key Java Business Transactions test** for the Confluence server, make sure you configure the **BTM PORT** parameter of the test with this port number.

Note:

When BTM-enabling multiple instances on the same server, make sure you configure a different **BTM Port** for each instance.

10. Also, by default, the **Designated_Agent** parameter will be empty; do not disturb this default setting. In this case therefore, the eG Java BTM will treat the host from which the very first 'measure request' comes in as the **Designated_Agent**.

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# Below property is used to specify IP address of eG Agent which collectes BTM Data.

# Default is None

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
#

Designated_Agent=

#
```

Note:

In case a specific **Designated_Agent** is not provided, and the eG Java BTM treats the host from which the very first 'measure request' comes in as the **Designated_Agent**, then if such a **Designated_Agent** is stopped or uninstalled for any reason, the eG Java BTM will wait for a maximum of 10 measure periods for that 'deemed' **Designated_Agent** to request for metrics. If no requests come in for 10 consecutive measure periods, then the eG Java BTM will begin responding to 'measure requests' coming in from any other eG agent.

11. Finally, save the **btmOther.props** file.

12. Then, proceed to configure the Confluence server with the path to the **eg_btm.jar** and **.props** files. For this, first open the **catalina.sh** file in the **/opt/atlassian/confluence/bin** directory.

13. Look for the *JAVA_OPTS=* entry in the file.

14. Once the entry is found, insert the following lines under that entry, as depicted by 2.2.

```
JAVA_OPTS="$JAVA_OPTS -javaagent:<<PATH TO THE eg_btm.jar>> -DEG_PROPS_HOME=<<PATH
TO THE LOCAL FOLDER CONTAINING THE .PROPS FILE>>"
```

For instance, if the eg_btm.jar and .props files were copied to the **/opt/btm/Conflu1_8090** directory on the Confluence server, then your specification will be as follows (see Figure 2.5):

```
JAVA_OPTS="$JAVA_OPTS -javaagent:/opt/btm/Conflu1_8090/eg_btm.jar -DEG_PROPS_
HOME=/opt/btm/Conflu1_8090"
```

```
# Register custom URL handlers
# Do this here so custom URL handles (specifically 'war:...') can be used in the security policy
JAVA_OPTS="$JAVA_OPTS -Djava.protocol.handler.pkgs=org.apache.catalina.webresources"
JAVA_OPTS="$JAVA_OPTS -javaagent:/opt/egurkha/lib/btm/Conflu1_8090/eg_btm.jar -DEG_PROPS_HOME=/opt/egurkha/lib/btm/Conflu1_8090
```

Figure 2.5: Editing the catalina.sh file to BTM-enable Confluence

15. Next, add the package com.eg in the boot delegation framework. For this, you need to add an entry to this effect in the **catalina.sh** file. Figure 2.6 depicts how this can be done.

```
250
251  # Register custom URL handlers
252  # Do this here so custom URL handles (specifically 'war:...') can be used in the security policy
253  JAVA_OPTS="$JAVA_OPTS -Djava.protocol.handler.pkgs=org.apache.catalina.webresources"
254  JAVA_OPTS="$JAVA_OPTS -javaagent:/opt/egurkha/lib/btm/eg_btm.jar -DEG_PROPS_HOME=/opt/egurkha/lib/btm/Conflu_8090
     -Datlassian.org.osgi.framework.bootdelegation=META-INF.services,com.yourkit,com.eg.*,com.jprofiler,com.jprofiler.*,org.apache.xerces,
     n.*,sun.*,com.sun.jndi,com.icl.saxon,com.icl.saxon.*,javax.servlet,javax.servlet.*,com.sun.xml.bind.*"
```

Figure 2.6: Adding the com.eg package in the boot delegation framework

16. Then, save the file.

17. Finally, restart the Confluence server. For this, first go to the command prompt and switch to the /opt/atlassian/confluence/bin directory. Then, run the following command at the prompt:

    Start-confluence.bat /fg

18. Where multiple Confluence server instances on a host are to be monitored, Repeat 7 to 17 for each of the server instances.

Chapter 3: How to Monitor Atlassian Confluence Using eG Enterprise?

Once the pre-requisites for monitoring Confluence are fulfilled, proceed to manage the target Confluence server using the eG user interface. The steps in this regard are as follows:

1. Login to the eG administrative interface as a user who is assigned the Admin role.

2. Invoke the Admin tile menu. In the Infrastructure tile, follow the Components -> Add/Modify menu sequence.

3. In the page that appears next, select Atlassian Confluence as the Component type and click the Add New Component button.

4. Figure 3.1 will then appear.

Figure 3.1: Managing an Atlassian Confluence server in an agent-based manner

5. In Figure 3.1, provide the IP address / host name of the Confluence server and assign a unique Nick name to it.

6. By default, Confluence listens on port 8090. The same will be displayed against Port number. You can change it if required.

7. To manage Confluence in an agent-based manner, do the following:

- Leave the **Agentless** check box in Figure 3.1 unchecked.

- Select an **External Agent**.

- Click the **Add** button to add the server for monitoring.

8. To manage Confluence in an agentless manner, do the following:

- Select the **Agentless** check box in Figure 3.1. This will throw additional options, as depicted by Figure 3.2.

Figure 3.2: Managing Confluence in an agentless manner

- Next, select the **OS** on which Confluence is running.

- Set **Mode** of monitoring to *Other*.

- Pick a **Remote agent**, which will remotely monitor Confluence.

- Assign an **External agent** to Confluence.

- Finally, click the **Add** button to add the server for monitoring.

9. After adding the component, sign out of the eG admin interface by clicking the **Signout** icon at the top, left corner of the page.

10. Figure 3.3 will then appear listing the unconfigured tests for Confluence.

Figure 3.3: List of unconfigured tests for Atlassian Confluence

11. Click on any Performance test in Figure 3.3 to configure it. For instance, click on the **Indexing Statistics** test.

Figure 3.4: Configuring the Indexing Statistics test

12. To know what this test does and how to configure it, refer to the Indexing Statistics test topic.

13. After configuring the test, click **Update** in Figure 3.5 to register the configuration. Then, try to signout once again.

14. This time, you will be prompted to configure the following tests:

Figure 3.5: Some more tests to be configured for Atlassian Confluence

15. The tests displayed in Figure 3.5 are operating system-level tests. To know what these tests do

and how to configure. Then, configure each test one after another.

16. Finally, sign out of the eG administrative interface.

3.1 The Atlassian Confluence Layer

The tests mapped to the Atlassian Confluence layer capture issues related to content indexing, email delivery, request processing, and database connectivity of Atlassian Confluence.
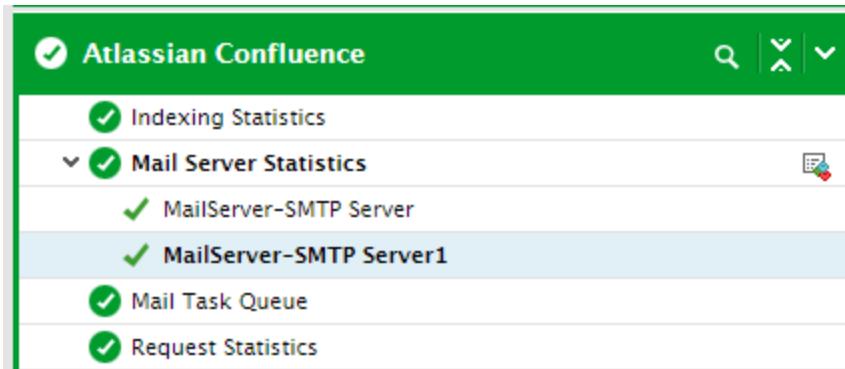

Figure 3.6: The tests mapped to the Atlassian Confluence layer

3.1.1 Indexing Statistics Test

The content index, also called the search index, allows searching of Confluence content. It is also used for a number of related functions such as building email threads in the mail archive, the space activity feature, and lists of recently-updated content.

For reasons of efficiency, Confluence does not immediately add content to the index. New and modified Confluence content is first placed in a queue and the queue is processed once every five seconds (by default). If the queue keeps building up without updating the index, then newly added/modified content will not be picked up by Confluence. Confluence users will hence end up consuming outdated content. This can adversely impact the user experience with Confluence.

To avoid this, administrators need to monitor the indexing activity continuously, track changes to the index queue length, and proactively detect processing bottlenecks well before it impacts user experience. This is where the Indexing Statistics test helps!

This test monitors the index queue and promptly alerts administrators if the count of pending tasks in the queue keeps increasing. This way, the test points administrators to a probable index processing bottleneck, prompts them to investigate the reasons for the same, and enables them to arrive at the appropriate corrective action. Based on these observations, if administrators decide to rebuild the index to clear the bottleneck, they can use the test to figure out if any re-indexing is already in

progress. The test additionally reports the time taken for the last indexing/reindexing, and thus indicates if re-indexing is slow. Detailed diagnostics also reveal the precise task that was last re-indexed and is the probable cause for the indexing slowness.

Target of the test: Atlassian Confluence

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for the Confluence server 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.
Port	The port number at which the specified host listens to
JMX remote port	Here, specify the port at which the JMX listens for requests from remote hosts. Ensure that you specify the same port that you configured in the catalina.bat file in the <<ATLASSIAN_ CONFLUENCE_ INSTALL_ DIR>>\confluence\bin directory.
JNDIname	The JNDIname is a lookup name for connecting to the JMX connector. By default, this is jmxrmi. If you have resgistered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
User, Password, and Confirm password	If JMX requires authentication only (but no security), then ensure that the user and password parameters are configured with the credentials of a user with read-write access to JMX. Confirm the password by retyping it in the Confirm Password text box.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the target server. If there is no response from the target beyond the configured duration, the test will timeout. By default, this is set to 10 seconds.
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

Parameter	Description
	by specifying none against this parameter.
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
Is cache currently flushing	Indicates whether/not the cache is currently flushing.		Sometimes, cached content can be added to the index. This is very often done to make external content - i.e., content from external sources such as SQL databases, Excel, etc. - searchable in Confluence. For this, typically, a Cache Macro is used. The Confluence page displaying the external content should include a cache macro instance with indexing enabled. The cache macro renders contents of the macro into HTML and stores the HTML in the cache. The cache content extractor processes the HTML data from the cache, extracts only the text and attribute fields, and inserts them into the index

Measurement	Description	Measurement Unit	Interpretation
			for that external content. Where a cache macro is in use, you can use this measure to figure out if the HTML data in the cache is being flushed into the index. The values that this measure can report and their corresponding numeric values are listed in the table below: **Measure Value** / **Numeric Value** Yes / 1 No / 0 **Note:** By default, this measure reports the **Measure Value**s listed in the table above to indicate whether/not cache flushing is in progress. In the graph of this measure however, the same is indicated using the numeric equivalents.
Is confluence content currently reindexing	Indicates whether/not Confluence is currently re-indexing.		Typically, when the index is rebuilt, it is reindexed. This measure therefore, indicates whether/not any index rebuild is in progress. The values that this measure can report and their corresponding numeric values are listed in the table below: **Measure Value** / **Numeric Value** Yes / 1 No / 0

Measurement	Description	Measurement Unit	Interpretation
			If the number of items in the index queue increases, it is a sign of an indexing bottleneck. Performing an index rebuild from scratch will resolve this bottleneck. This is why, you may want to check the value of this measure, if you find the value of the *Task queue length* measure increasing consistently. In such a situation, the value *Yes* for this measure could indicate that re-indexing is being performed, probably to resolve an indexing bottleneck. On the other hand, if the value of this measure is *No* in such a situation, it indicates to administrators that an index rebuild is a must, but is yet to begin.
			If the value of this measure is *Yes* for a long time, it could indicate that reindexing is stuck. Probable causes for this are as follows:
			• Intermittent database disruptions might cause problems during the reindex, but resolve quickly enough that Confluence as whole is not impacted Adding a validation query should resolve any intermittent connection problems to the database.
			• Some attachments cause problems when being read. You can disable indexing of attachments temporarily, to give

Measurement	Description	Measurement Unit	Interpretation
			the reindex a chance to complete. The debug logging will help to determine the file that was attempted to be indexed, before the process got into the stuck state.
			• Through analysis of the thread dumps, you might find that the indexer threads are waiting on an external resource; such as a lock on a file or a database connection. This may also occur in other scheduled jobs.
			Note:
			By default, this measure reports the Measure Values listed in the table above to indicate whether/not re-indexing is in progress. In the graph of this measure however, the same is indicated using the numeric equivalents.
Time taken for last indexing	Indicates the time taken during the last re-indexing.	Msecs	Ideally, the value of this measure should be low. A very high value indicates that re-indexing is slow. In such situations, you can use the detailed diagnosis of this measure to determine the task that was last re-indexed.

Typically, the length of time depends on the following factors:

• Number of pages in your Confluence instance.

• Number, type and size of attachments. |

Measurement	Description	Measurement Unit	Interpretation
			• Amount of memory allocated to Confluence. • Disk throughput. It may help to increase the heap memory allocation of Confluence. The process is basically the same for Confluence or Jira applications. If you are running an older version of Confluence and find that the index rebuild is not progressing, you may need to shut down Confluence, and restart it with the following Java system property set: bucket.indexing.threads.fixed=1.
Task queue length	Indicates the number of tasks in the index queue.	Number	If the value of this measure keeps increasing, it counld indicate that indexing is stuck. Usually, re-indexing from scratch resolves the problem, but it is not a permanent solution. You may want to track which artifact (either a page, artifact, or some other problem) is responsible for the stuck state. For this, you can use the detailed diagnosis of the *Time taken for last indexing* measure. This reveals the task that was last re-indexed and has probably contributed to the stuck state. Other causes for the queue being stuck include: • Intermittent database disruptions might cause problems during the reindex, but resolve quickly enough that Confluence as whole is not impacted Adding a validation query

Measurement	Description	Measurement Unit	Interpretation
			should resolve any intermittent connection problems to the database. • Some attachments cause problems when being read. You can disable indexing of attachments temporarily, to give the reindex a chance to complete. The debug logging will help to determine the file that was attempted to be indexed, before the process got into the stuck state. • Through analysis of the thread dumps, you might find that the indexer threads are waiting on an external resource; such as a lock on a file or a database connection. This may also occur in other scheduled jobs

3.1.2 Mail Server Statistics Test

Configuring a Confluence server to send email messages allows Confluence users to:

• Receive emailed notifications and daily reports of updates: You can 'watch' a page, blog post or space. Confluence can be configure to send users a notification by email whenever anyone adds or updates content on a page, blog post, or space that they are watching/following. Users typically receive immediate emails for important notifications (like mentions and new pages). However, when lots of changes are being made at the same time, users will only receive a single email with all the changes within a short window (usually 10 minutes).

• Send a page via email: Users can email the link to a page or blog post that they are viewing to any recipient of their choice.

If users fail to receive email notifications of updates or emails containing links to pages of interest to them, it is bound to adversely impact user productivity and shake user confidence in Confluence. If this is to be avoided, then administrators should track emails sent by Confluence to users, promptly

capture delivery failures, and rapidly troubleshoot the failures before users notice that something is amiss. This is where the Mail Server Statistics test helps!

This test automatically discovers the mail servers that Confluence is configured with, and tracks the emails sent by each mail server. In the process, the test promptly captures email delivery failures, and swiftly alerts administrators to the failures, so that administrators are enabled to quickly investigate and isolate the reason for the failures and fix them.

Target of the test: Atlassian Confluence

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each mail server configured for the target Confluence server

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 number at which the specified host listens to
JMX remote port	Here, specify the port at which the JMX listens for requests from remote hosts. Ensure that you specify the same port that you configured in the catalina.bat file in the <<ATLASSIAN_ CONFLUENCE_ INSTALL_ DIR>>\confluence\bin directory.
JNDIname	The JNDIname is a lookup name for connecting to the JMX connector. By default, this is jmxrmi. If you have resgistered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
User, Password, and Confirm password	If JMX requires authentication only (but no security), then ensure that the user and password parameters are configured with the credentials of a user with read-write access to JMX. Confirm the password by retyping it in the Confirm Password text box.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the target server. If there is no response from the target beyond the configured duration, the test will timeout. By default, this is set to 10 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Emails attempted	Indicates the number of email messages that this mail server has tried to send.	Number	This is a good indicator of the workload of a mail server.
Emails sent	Indicates the number of emails successfully sent by this mail server.	Number	Ideally, the value of this measure should be the same as the value of the *Emails attempted* measure. If this measure value is less than the value of the *Emails attempted* measure, it indicates that one/more attempts have failed
Emails failed	Indicates the number of emails that could not be delivered by this mail server.	Number	Ideally, the value of this measure should be 0. A non-zero value is indicative of email delivery failures.
Success percentage	Indicates the percentage of attempted email deliveries that were successful.	Percent	Ideally, the value of this measure should be 100%. A low value indicates that many emails were not delivered.
Failed percentage	Indicates the percentage of attempted email deliveries that failed.	Percent	Ideally, the value of this measure should be 0. A high value indicates that many emails were not delivered. In such a situation, you may want to do the following: • Ensure that you have properly configured an SMTP Server. Send a Test Mail inside the SMTP Server configuration setup screen. Make a note of any error that is returned from the test.

Measurement	Description	Measurement Unit	Interpretation
			• Check your JIRA application log files and the application server log files for Out of Memory errors. Typically, the log file will show java.lang.OutOfMemoryError: Java heap space. This has been known to cause the service responsible for sending emails out to fail until your applications are restarted. You should further troubleshoot your memory issues. • Check and ensure the Mail Queue Service is installed. Click Admininstration > Services to inspect that the service exists, and is set at a reasonable interval. This interval controls how frequently the mail queue is processed. You can flush the mail queue to send out pending messages immediately to your mail server. • Inspect your Mail Queue under Administration > Mail Queue. See if you are given the option to Bypass currently sending mail. A stuck email or trackback ping can hold up the queue. • Check that your Base URL is set to a domain / IP which your SMTP server will accept. Example: Google apps accounts must have a matching base url to their Google Apps domain. • Enable additional logging in

Measurement	Description	Measurement Unit	Interpretation
			Administration > System > Troubleshooting and Support > Logging and Profiling by setting the following to DEBUG to see more robust logging about services running at the background.

3.1.3 Mail Task Queue Test

Email messages waiting to be sent are queued in a mail queue and periodically flushed from Confluence once a minute. A Confluence administrator can also manually flush messages from the mail queue. If there is an error sending messages, the failed email messages are sent to an error queue from which you can either try to resend them or delete them.

If the number of emails in the queue keeps increasing with time, it could mean that mail delivery has failed or is taking too long. Similarly, if emails are not flushed out of the error queue quickly, it once again hints at issues with mail delivery or mail system configuration. To ensure that the mail system functions smoothly and efficiently, administrators should be able to detect such anomalies instantly, and take action against them promptly. For this purpose, it will be good practice for administrators to run the Mail Task Queue test, periodically.

This test tracks changes to the mail queue and error queue, and proactively alerts administrators if the size of these queues keeps growing consistently. This points administrators to mail delivery failures or slowness in delivery of mails to recipients. The test also alerts administrators if too many mails from the mail queue failed to be delivered and were hence placed in the error queue. This way, the test sheds light on issues with the Confluence mailing system, thereby enabling administrators to promptly resolve them.

Target of the test: Atlassian Confluence

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for the target Confluence server

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.
Port	The port number at which the specified host listens to
JMX remote port	Here, specify the port at which the JMX listens for requests from remote hosts. Ensure that you specify the same port that you configured in the catalina.bat file in the <<ATLASSIAN_ CONFLUENCE_ INSTALL_ DIR>>\confluence\bin directory.
JNDIname	The JNDIname is a lookup name for connecting to the JMX connector. By default, this is jmxrmi. If you have resgistered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
User, Password, and Confirm password	If JMX requires authentication only (but no security), then ensure that the user and password parameters are configured with the credentials of a user with read-write access to JMX. Confirm the password by retyping it in the Confirm Password text box.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the target server. If there is no response from the target beyond the configured duration, the test will timeout. By default, this is set to 10 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Task queue size	Indicates the number of email messages queued for dispatch.	Number	If the value of this measure keeps increasing with time, it indicates that automatic flushing of emails from mail queue is not working. You may want to manually flush the messages from the mail queue in this case. Another reason for a long mail queue could be delay/slowness in delivery of emails to intended recipients. There could be multiple causes for this to happen:

Measurement	Description	Measurement Unit	Interpretation
			• Mail Queue Services are scheduled to be running by default once per minute. But sometimes due to user preference, it can be altered. This change will have a direct impact on the time it will take for an email to reach a user. To resolve this issue, you may want to reset the frequency of operations of the Mail Queue Services to its default or to some other acceptable frequency.
			• There could be times where there is an error message that cannot be flushed out of the queue; Confluence application attempts to flush the mail again and again, but ends up causing other mail items to be delayed. In this case, you may want to attempt a manual flush of the mail queue, in an effort to identify the exact message that fails to be flushed. Then, reach out to https://support.atlassian.com for further assistance.
			• By default, Confluence applications services are running on either 2 or 4 QuartzWorker threads - this depends on the version of Confluence. If there is some obstruction or delay on the threads executing, the next service may not be executed despite having the services scheduled to run. This

Measurement	Description	Measurement Unit	Interpretation
			means that if there are other services that are scheduled to run at the same frequency as the mail service, then the mail service could be delayed if the services scheduled to run before it are delayed. Setting appropriate operational frequencies for the different services will resolve this problem. ● Verify that you are able to reach the mail server by ping and check if the latency is high on delivery. If the latency is high, this will delay the JavaMail service used by Confluence applications to receive the response in a timely manner. Delays can also occur if the SMTP mail server is overloaded, and is therefore unable to process mails quickly. Consult the network administrator or mail server administrator to further troubleshoot on this. ● Every time mail is attempted to be sent, it will perform a reverse DNS lookup for the Confluence server application hostname. If the DNS isn't reachable, Confluence application will have to wait for a timeout which can be a long period of time (20-40 seconds). Make sure your DNS configuration is correct to avoid this issue.

Measurement	Description	Measurement Unit	Interpretation
			• There could be subscriptions being sent to huge user groups or slow queries being run as part of the filters that are subscribed to. Check the filtersubscription table and slow searchrequest table for such anomalous subscriptions/queries.
Error queue size	Indicates the number of error mail messages in the error queue.	Number	Ideally, the value of this measure should be 0. A non-zero value implies that one/more mails in the mail queue could not be delivered and have hence been placed in the error queue. If the value of this measure keeps increasing with time, it indicates that emails in the error queue could not be delivered even after multiple retries. This is a cause for concern and will have to be looked into immediately.
Retry count	Indicates the number of times the delivery of emails in the error queue was retried.	Number	Ideally, the value of this measure should be low.
Is currently flushing	Indicates whether/not the mail queue is currently flushing.		The values that this measure can report and their corresponding numeric values are listed in the table below: **Measure Value** Yes No **Note:**

Measurement	Description	Measurement Unit	Interpretation
			By default, this measure reports the **Measure Value**s listed in the table above to indicate whether/not mail queue flushing is in progress. In the graph of this measure however, the same is indicated using the numeric equivalents.
Error percentage	Indicates what percentage of mails in the mail queue have been placed in the error queue.	Percent	Ideally, the value of this measure should be very less. A value close to 100% is a cause for concern, as it indicates that almost all of the messages in the mail queue could not be delivered, and were hence placed in the error queue. The reason for the delivery failures will have to be investigated and promptly fixed. You may want to do the following in this case: • Ensure that you have properly configured an SMTP Server. Send a Test Mail inside the SMTP Server configuration setup screen. Make a note of any error that is returned from the test. • Check your JIRA application log files and the application server log files for Out of Memory errors. Typically, the log file will show java.lang.OutOfMemoryError: Java heap space. This has been known to cause the service responsible for sending emails out to fail until your applications are restarted. You should further troubleshoot your

Measurement	Description	Measurement Unit	Interpretation
			memory issues. • Check and ensure the Mail Queue Service is installed. Click Admininstration > Services to inspect that the service exists, and is set at a reasonable interval. This interval controls how frequently the mail queue is processed. You can flush the mail queue to send out pending messages immediately to your mail server. • Inspect your Mail Queue under Administration > Mail Queue. See if you are given the option to Bypass currently sending mail. A stuck email or trackback ping can hold up the queue. • Check that your Base URL is set to a domain / IP which your SMTP server will accept. Example: Google apps accounts must have a matching base url to their Google Apps domain. • Enable additional logging in Administration > System > Troubleshooting and Support > Logging and Profiling by setting the following to DEBUG to see more robust logging about services running at the background.

3.1.4 Request Statistics Test

How quickly Confluence responds to user requests determines how happy users are with the application. If Confluence takes too long to respond to requests, it could point to issues in

configuration or tuning that is probably impacting overall server performance. If these issues are not detected promptly and addressed immediately, they may significantly degrade user experience with Confluence.

To avoid this, administrators should run the ConfluRStatsTest at frequent intervals. This test tracks requests to Confluence and reports the time taken by Confluence to service the requests. In the process, the test reveals request processing bottlenecks in Confluence and its overall health. The test also captures and reports errors in processing, thus enabling administrators to promptly intervene and fix the errors. Additionally, the test also runs a sample query on the Confluence database, reports the time taken for the query to run, and thus points to any latency in database connectivity, so that administrators can figure out if that could be impacting the performance of Confluence.

Target of the test: Atlassian Confluence

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for the target Confluence server

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 number at which the specified host listens to
JMX remote port	Here, specify the port at which the JMX listens for requests from remote hosts. Ensure that you specify the same port that you configured in the catalina.bat file in the <<ATLASSIAN_ CONFLUENCE_ INSTALL_ DIR>>\confluence\bin directory.
JNDIname	The JNDIname is a lookup name for connecting to the JMX connector. By default, this is jmxrmi. If you have resgistered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
User, Password, and Confirm password	If JMX requires authentication only (but no security), then ensure that the user and password parameters are configured with the credentials of a user with read-write access to JMX. Confirm the password by retyping it in the Confirm Password text box.

Parameter	Description
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the target server. If there is no response from the target beyond the configured duration, the test will timeout. By default, this is set to 10 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Current number of requests being served	Indicates the number of requests currently being served by Confluence.	Number	This is a good indicator of the current workload of Confluence.
Number of error pages served	Indicates the number of times the Confluence error page was served.	Number	Ideally, the value of this measure should be 0. A non-zero value indicates errors in processing.
Number of requests served	Indicates the number of requests served.	Number	
Average execution time of last 10 requests	Indicates the average execution time for the last ten requests.	Msecs	Ideally, the value of this measure should be low. If this value keeps increasing consistently, it could indicate poor server performance. To avoid this, make sure that the following are in place: • Use the latest versions of your application servers and Java runtime environments. Newer versions are usually better optimized for performance. • Always watch the swapping activity of your server. If there is not enough RAM available, your server may start swapping out some of Confluence's heap data to your hard disk. This will slow down the JVM's garbage

Measurement	Description	Measurement Unit	Interpretation
			collection considerably and affect Confluence's performance. In clustered installations, swapping can lead to a Cluster Panic due to Performance Problems. This is because swapping causes the JVM to pause during Garbage Collection, which in turn can break the inter- node communication required to keep the clustered nodes in sync.
			• Be aware of other systems using the same infrastructure. Because, one application's bug might start affecting other applications. Either make sure applications can't harm each other despite sharing the same infrastructure, or get these systems untangled, for example by moving them to separate instances that can be controlled better.
			• The embedded H2 database is provided for evaluating Confluence, not for production Confluence sites. After the evaluation finishes, you must switch to a supported external database.
			• If load on Confluence is high, you may need more

Measurement	Description	Measurement Unit	Interpretation
			simultaneous connections to the database. If you are using JNDI data-sources, you will do this in your application server's configuration files. If you have configured Confluence to access the database directly, you will need to manually edit the hibernate.c3p0.max_ size property in the confluence.cfg.xml file in your confluence.home directory. After you have changed the URL in this file, restart Confluence.To assess whether you need to tune your database connection pool, take thread dumps during different times (including peak usage). Inspect how many threads have concurrent database connections. ● To reduce the load on the database, and speed up many operations, Confluence keeps its own cache of data. Tuning the size of this cache may speed up Confluence (if the caches are too small), or reduce memory (if the caches are too big). ● Antivirus software greatly decreases the performance of Confluence. Antivirus software that intercepts access to the

Measurement	Description	Measurement Unit	Interpretation
			hard disk is particularly detrimental, and may even cause errors with Confluence. You should configure your antivirus software to ignore the Confluence home directory, its index directory and any database-related directories.
			• If bandwidth is responsible for bottlenecking in your Confluence installation, you should consider enabling HTTP compression. This may also be useful when running an external facing instance to reduce your bandwidth costs. Take note of the known issues with HTTP compression in versions of Confluence prior to 2.8, which may result in high memory consumption.
			• For high- load environments, performance can be improved by using a web server such as Apache in front of the application server. When configuring your new web server, make sure you configure sufficient threads/processes to handle the load. This applies to both the web server and the application server connector, which are typically configured separately. If possible, you should enable

Measurement	Description	Measurement Unit	Interpretation
			connection pooling in your web server connections to the application server.
			• Some external plugins, usually ones that have been written a long time ago and that are not actively maintained anymore, have been reported to consume memory and never return it. Ultimately this can lead to a crash, but first this manifests as reduced performance.
			• If Confluence is running slowly, one of the most likely cause is that there is some kind of bottleneck in (or around) the database. Check the value of the databaseExpLatency measure to determine whether/not you are dealing with a latent database. Then, take some database optimization measures to improve database performance. which in turn will improve the performance of Confluence.
Database example latency	Indicates time it took for the database to execute the sample query.	Msecs	Ideally, the value of this measure should be very less. A high value is indicative of database problems. Typically, the sample query used should return results within 1 or 2 milliseconds. If the value displayed is between 3 and 5 milliseconds,

Measurement	Description	Measurement Unit	Interpretation
			you might already have an issue. If the value is above 10ms, then you definitely need to investigate and improve performance! High latency might stem from all sorts of problems (slow network, slow database, connection- pool contention, etc), so it's up to you to investigate. Don't stop improving until latency is below 2ms on average.

Also, note that you may get zero latency and still have massive database problems, e.g. if your tables are poorly indexed. So don't let a low latency fool you either. |

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.