Article

Service Assurance Suite

Release number 9.4

Troubleshooting the Deployment of SAM, EMC M&R and the SolutionPack for EMC Smarts

January, 2015

- Overview 2
- Log files 2
- Status checks on services 3
- Unable to access Smarts EDAA 6
- Notification cache is empty 6
- Stale alerts and messages appear in Notification cache 7
- RabbitMQ does not start 8
- Tomcat does not start 9
- Tomcat starts but then stops 9
- Changing the password of the EDAA account 9
- How do I change the RabbitMQ passwords 10
- UI cannot display alerts or Smarts topology when CAS is enabled 13
- Wrong CAS host name in runcmd_env.sh 13
- Services remain registered and running after an uninstall 13
- Retrieving the current elasticsearch PID 15
- Ports that must be open between SAM and EMC M&R 15

EMC²

Overview

Use this article to help you find solutions to issues you may have with SAM, the EMC M&R platform, and the SolutionPack for Smarts.

Log files

Log files help you diagnose issues that you may have with SAM, the EMC M&R platform, and the SolutionPack for EMC Smarts.

SAM log files

The SAM log file names may differ depending on the domain name. You can find these log files in `SAM/smarts/local/logs`.

brstart_en_US_UTF-8.log
: Created by the Smarts broker.

emc-alerts-es.log
: Created by the elasticsearch service. The Notifications cache uses elasticsearch to store and index its notifications to be served by the Alerts EDAA.

INCHARGE-SA-PRES.audit
: The SAM audit log.

INCHARGE-SA-PRES_en_US_UTF-8.log
: The SAM server log. In this case, the log's name represents the INCHARGE-SA-PRES domain.

rabbit@‹*HOSTNAME*›.log
: Created by the RabbitMQ service. The Notification cache uses the RabbitMQ service to transfer notifications from SAM to the Notification cache. This log file name depends on the host name of the SAM installation.

setup_elasticsearch.log
: Created the first time you run the elasticsearch service. This service must be primed the first time it is run, and this log records the setup results.

setup_rabbitmq.log
: Created the first time that you run the RabbitMQ service. This service must be primed the first time it is run, and this log records the setup results.

sm_serviced_en_US_UTF-8.log
: Created by the sm_serviced daemon process that is responsible for running the registered Smarts services.

Tomcat log files

These log files are generated by Tomcat and any web applications running within Tomcat. You can find these log files in `SAM/smarts/local/logs/tomcat`.

alerts-edaa.log
: Created by the Alerts EDAA web application. This web application exposes notifications in the Notification cache.

alerts-ingestion.log
: Created by the alerts-ingestion web application. This web application populates notifications in to the Notification cache that come from SAM via RabbitMQ.

catalina.‹*date*›.log
: The Tomcat engine writes log messages pertaining to Tomcat.

catalina.out
Logs the code output running under Tomcat. This output is also written to System.out or System.err.

localhost.‹*date*›.log
Logs information about the virtual host within Tomcat. Always check this log if you cannot find an error in catalina.out or if something is seriously wrong.

localhost_access_log.‹*date*›.txt
Logs the calls being made to the Tomcat server.

smarts-edaa.log
Created by the smarts-edaa web application. This web application exposes the topology from one or more domain managers in a RESTful feed.

Status checks on services

You can verify that the services are running and performing their required tasks.

Perform status checks on services by following these procedures:

- Checking elasticsearch on page 3
- Checking the notifications exchange on page 4
- Checking the RabbitMQ configuration on page 4
- Checking Alerts EDAA on page 5
- Checking Smarts EDAA on page 5

Checking elasticsearch

The elasticsearch service, a component of the Notifications cache, is only accessible from the local machine. Verify this service is running and serving requests.

Procedure

1. Verify elasticsearch is running by typing **`$ smarts/bin/sm_service show`**.

   The expected result is: RUNNING smarts-elasticsearch

2. Verify elasticsearch is serving requests from a browser on the same machine on Windows or using a tool like curl on Linux.

   Browser example:

   `http://localhost:9200/_status`

   A page specific to the browser appears.

   Linux examples:

   `curl -XGET http://localhost:9200/_status`
   `curl http://localhost:9200/smarts/alert/_count`

   A JSON page appears indicating success and showing Smarts as one of the indices. For example:

   ```
   "_shards":{"total":10,"successful":5,"failed":0}
   .
   .
   .
   "indices":{"smarts":...
   ```

3. If the Smarts index does not exist, the setup script has failed. Check the `setup_elasticsearch.log` file. Note that after you first start elasticsearch, it may take a few minutes before the setup script begins running.

Checking the notifications exchange

The RabbitMQ service exchanges notifications from SAM to the Notification cache. This service is running by default and is a password-protected web-based management interface running on port 15672.

Procedure

1. Verify RabbitMQ is running: **`http://<HOSTNAME>:15672/`**
2. Type the credentials for the RabbitMQ interface. The default values are **`admin`** for the user and **`changeme`** for the password.

   The main page shows various statistics and status information for RabbitMQ.

Checking the RabbitMQ configuration

Follow these steps to check if the RabbitMQ service was properly configured.

When you start the `sm_rabbitmq` script, it verifies the RabbitMQ configuration and creates the required exchanges, queues, users, and priming message in `qnameq` if they do not exist. If this script fails, check the `local/logs/setup_rabbitmq.log` file.

Procedure

1. Log in to the RabbitMQ interface: **`http://<HOSTNAME>:15672/`**
2. Type the credentials for the RabbitMQ interface. The default values are **`admin`** for the user and **`changeme`** for the password.
3. Verify these exchanges are under the **Exchanges** tab:
   - `emc.smarts.notifications` (of type topic)
   - `emc.smarts.notifications.ingestion` (of type x-consistent-hash)
4. Verify these queues are under the **Queues** tab:
   - `emc.smarts.notifications.q1`, the main queue for holding notifications for ingestion
   - `emc.smarts.notifications.qnameq`

   If a high number of messages appear under `emc.smarts.notifications.q1`, ingestion is either backlogged or has stopped.

   One message should appear under `emc.smarts.notifications.qnameq`.
5. Verify these users are defined under the **Admin** tab, and have access to the emc.smarts.notifications vhost:
   - admin (with the administrator tag)
   - ingestion
   - publisher

Checking Tomcat

Verify Tomcat is running and serving requests.

Procedure

1. Verify Tomcat is running by typing **`$ bin/sm_service show`**.

   The expected result is: RUNNING smarts-tomcat

2. Verify Tomcat is serving requests from a browser: **`http://<HOSTNAME>:8080/`**

   A welcome page for Tomcat Version 8.x appears.

Checking Alerts EDAA

Verify the Alerts EDAA, a component of the Notification cache, is running and connected to elasticsearch.

Procedure

1. Verify Tomcat is running, as described in Checking Tomcat on page 5.
2. Verify `alerts_edaa` is running.

   For example:

   `http://itops-qa-192.lss.emc.com:8080/alerts-edaa/msa?alt=json&pretty=true`

3. Make sure there is an entry for alerts on the page.
4. Check that the cache is connected to the elasticsearch service and has the expected notifications.

   For example:

   `http://itops-qa-192.lss.emc.com:8080/alerts-edaa/msa/alerts/instances?alt=json&pretty=true`

Checking Smarts EDAA

Verify the Smarts EDAA is running, connected to the broker and is able to retrieve information from your presentation SAM.

Procedure

1. Verify Tomcat is running, as described in Checking Tomcat on page 5.
2. Verify `smarts-edaa` is running.

   For example:

   `http://itops-qa-192.lss.emc.com:8080/smarts-edaa/msa?alt=json&pretty=true`

3. Make sure there are entries for the dmbroker and the presentation SAM.
4. Verify `smarts-edaa` is connected to the broker.

   For example:

   `http://itops-qa-192.lss.emc.com:8080/smarts-edaa/msa/dmbroker/types/DomainManager/instances?alt=json&pretty=true`

5. Make sure there are entries for the various domains registered with the broker.
6. Verify `smarts-edaa` can retrieve information from your presentation SAM.

For example:

```
http://itops-qa-192.lss.emc.com:8080/smarts-edaa/msa/
INCHARGE-SA-PRES/instances/SM_System::SM-System?
alt=json&pretty=true
```

7. Make sure you can see the properties of `SM_System::SM-System`.

Unable to access Smarts EDAA

If your system cannot access Smarts EDAA, follow these steps.

Procedure

1. Verify the SAM server is not using a broker from a pre-9.4 Smarts installation.
2. Verify the SAM broker is configured for Smarts EDAA access:
   <*SAM_base_directory*>/SAM/smarts/bin> dmctl -s localhost:426/dmbroker getI EDAAService

   The expected result is: `SMARTS-EDAA`
3. If you made configuration changes to the broker or SAM using the `runcmd_env.sh` file, do the following in this order:
   a. Shutdown SAM.
   b. Shutdown Tomcat.
   c. Shutdown the broker.
   d. Restart the broker.
   e. Restart Tomcat.
   f. Restart SAM.

Notification cache is empty

Follow these steps when notifications are not appearing in the Notification cache.

Before you begin

Review this before you start the troubleshooting steps.

- If alerts show up in the Alerts EDAA but not in EMC M&R, the issue is in EMC M&R.
- If you see a backup of messages in RabbitMQ, the SAM publisher is working but the ingestion service or elasticsearch service may be failing.
- If no messages are passing through into RabbitMQ, the problem is on the SAM publisher side.

Procedure

1. Verify all services are running, as described in Status checks on services on page 3.
2. Check that SAM has started to publish to RabbitMQ and started with `--bootstrap=bootstrap-amqp.conf`.
3. Verify SAM is exposed via the EDAA and started with `--edaa=sam-presentation/2.0`.
4. Under the **Connections** tab in the RabbitMQ web-based management interface, do the following:

a. Verify that there are two connections: one from RabbitMQ / Java and the other from rabbitmq-c.

b. If RabbitMQ / Java is not present, check if Tomcat is running and view the `<SAM_base_directory>/SAM/smarts/local/logs/tomcat/alert-ingestion`.

c. If rabbitmq-c is not present and if SAM is running, check the presentation SAM log to see if it is publishing to RabbitMQ.

5. From the host running the presentation SAM server, check if you can access the elasticsearch database and verify the count from a browser on Windows or using a tool like curl on Linux.

   Browser example:

   `http://localhost:9200/smarts/alert/_count`

   Linux example:

   `curl http://localhost:9200/smarts/alert/_count`

   A result similar to this appears:

   `{"count":50002,"_shards":{"total":5,"successful":5,"failed":0}}`

6. If the count is zero, check if there are any notifications in SAM.

7. From a browser, check if you can access Tomcat in the SAM server host.

   For example:

   `http://hostname:8080`

8. Verify elasticsearch is running by checking `<SAM_base_directory>/SAM/smarts/local/logs/tomcat/alert-ingestion.log`

   If the service was running and is now shut down or unable to process notifications, a message similar to this may appear in the `alerts-ingestion.log` file:

```
{date} {time},{pid} INFO  'ALERTS-IN' EventConsumerService
EventConsumerPool-1 watchdog attemps to reconnect
 {date} {time},{pid} INFO  'ALERTS-IN' plugins EventConsumerPool-1
[La Lunatica] loaded [], sites [] {date} {time}
 {date} {time},{pid} ERROR 'ALERTS-IN'
ElasticsearchNodeConfigurationService EventConsumerPool-1
 ElasticSearchNodeConfigurationService::verifyIndexing:
ElasticSearch :

SERVICE_UNAVAILABLE :org.elasticsearch.client.transport.NoNodeAvail
ableException: No node available
 {date} {time},{pid} INFO  'ALERTS-IN' transport elasticsearch[Ent]
[generic][T#8] [Ent] failed to get node info for [#transport#-1]
[inet
 [localhost/127.0.0.1:9300]], disconnecting...
 org.elasticsearch.transport.NodeDisconnectedException: []
[inet[localhost/127.0.0.1:9300]][cluster/nodes/info] disconnected
```

Stale alerts and messages appear in Notification cache

If the Notification cache contains stale alerts and messages, you can synchronize SAM and the Notifications cache. This happens when you restart SAM with a different RPS file.

You have two options to synchronize SAM and the Notification cache:

- Stopping the SAM publisher to clear the Notification cache on page 8. This is the recommended option because it has low impact on the system.

- Stopping SAM to clear the Notification cache on page 8.

Stopping the SAM publisher to clear the Notification cache

This method of clearing the Notification cache has less impact on the system because you are only stopping the SAM publisher.

Procedure

1. Stop the publisher: **`$ bin/dmctl -s INCHARGE-SA-PRES invoke SM_AMQPNotificationPublisher::SM-AMQPNotificationPublisher stop`**
2. Clear the Notifications cache by typing these commands:
   a. **`$ bin/sm_service stop smarts-elasticsearch`**
   b. **`$ rm -rf local/repos/emc-alerts-es`**
   c. **`$ bin/sm_service start smarts-elasticsearch`**
3. Restart the publisher: **`$ bin/dmctl -s INCHARGE-SA-PRES invoke SM_AMQPNotificationPublisher::SM-AMQPNotificationPublisher start`**

Stopping SAM to clear the Notification cache

This method of clearing the Notification cache has a substantial impact on the system.

Procedure

1. Stop SAM: **`$ bin/sm_service stop ic-sam-pres`**
2. Clear the Notifications cache by typing these commands:
   a. **`$ bin/sm_service stop smarts-elasticsearch`**
   b. **`$ rm -rf local/repos/emc-alerts-es`**
   c. **`$ bin/sm_service start smarts-elasticsearch`**
3. Restart SAM: **`$ bin/sm_service start ic-sam-pres`**

RabbitMQ does not start

Follow these steps to determine why the RabbitMQ service does not start.

Procedure

1. Verify RabbitMQ is NOT already running by typing **`sm_service show`**.
2. Verify the `Erlang` process is NOT running by typing **`ps -ef | grep erl`**.
3. If the Erlang process is running, kill it before restarting RabbitMQ.

   The Erlang process may be running from a prior installation, or was started outside the `sm_service` mechanism.
4. Verify the `qpidd` process is NOT running by typing **`ps -ef | grep qpidd`**.
5. If the `qpidd` process is running, kill it before restarting the RabbitMQ service.

   The `qpidd` process runs on some Linux distributions and can conflict with RabbitMQ by using the same port.
6. Check that no other RabbitMQ installation exists by typing **`ps -ef | grep rabbit`**.

Tomcat does not start

Follow these steps to determine why Tomcat does not start.

Procedure

1. Verify that Tomcat is NOT already running by typing **`sm_service show`**.

   Tomcat may be running from a prior install, or was started outside of the `sm_service` mechanism. Also, another instance of Tomcat or another web server may be running on the system that prevents this Tomcat instance from binding to port 8080, causing it to shut down.

2. Verify the underlying application is not already running Tomcat by typing **`ps -ef | grep java`**.

   If you see that the `/opt/InCharge/SAM/smarts/tomcat/bin/bootstrap.jar` process is running, then Tomcat is likely running as well.

3. Review the messages in `SAM/smarts/local/logs/tomcat/catalina.out` and `SAM/smarts/local/logs/tomcat/catalina.<date>.log`.

Tomcat starts but then stops

Procedure

1. Follow the steps in Tomcat does not start on page 9
2. Check if another process is running on port 8080.

   This can cause Tomcat to start for a few minutes and then return to a NOT RUNNING state.

Changing the password of the EDAA account

The Smarts EDAA logs into domains using the EDAA account `_edaa`. This account has the same access privileges as the admin account as well as the ability to perform actions on behalf of other users. To secure this account, change its default password, which is changeme.

Procedure

1. In the install directory of each domain server accessed by Smarts EDAA, type `sm_edit conf/serverConnect.conf`

   By default, only presentation SAM is exposed.

2. In the `serverConnect.conf` file, change the encrypted default password for the `_edaa` account.

   For example, you can change this password:

   ```
   *:_edaa:<E-1.0>94C355DA7E0EB7AADE32F80F201F70C5CCA88288443EE23CB524
   9ACD10C773E1:Impersonate
   ```

   to the new password:

   ```
   *:_edaa:<new_password>:Impersonate
   ```

When you save the `serverConnect.conf` file, `sm_edit` encrypts the new password.

3. In the install directory of the SAM server hosting the Smarts EDAA, type `sm_edit conf/edaaClientConnect.conf`
4. In the `edaaClientConnect` file, change the encrypted default password for the `_edaa` account.

   For example, you can change this password:

   ```
   *:*:_edaa:<E-1.0>94C355DA7E0EB7AADE32F80F201F70C5CCA88288443EE23CB5
   249ACD10C773E1
   ```

   to the new password:

   ```
   *:*:_edaa:<new_password>
   ```

   When you save the `edaaClientConnect` file, `sm_edit` encrypts the new password.
5. Stop the Smarts Tomcat hosting the Smarts EDAA by typing `bin/sm_service stop smarts-tomcat`.
6. Restart the Smarts Tomcat hosting the Smarts EDAA by typing `bin/sm_service start smarts-tomcat`.
7. Verify that the Smarts EDAA can access each domain.

How do I change the RabbitMQ passwords

There are three default accounts in the RabbitMQ installation: an admin account, an account used by the SAM publisher to deliver messages into RabbitMQ, and an account used by the ingestion process to pull messages from RabbitMQ. You can change the password for each one of these accounts.

Follow these procedures to change the passwords of the RabbitMQ accounts:

- Changing the RabbitMQ admin password on page 10
- Changing the RabbitMQ publisher password on page 11
- Changing the RabbitMQ ingestion password on page 12

Changing the RabbitMQ admin password

The RabbitMQ admin account is used for configuring RabbitMQ and accessing RabbitMQ's web-based management interface. It is not used by any other component of the system except for the initial RabbitMQ setup process.

Procedure

1. Make sure the smarts-rabbitmq service was started.
2. To access the RabbitMQ web-based management interface, connect with a browser to port 15672 on the host where RabbitMQ is installed.

   For example:

   `http://my-pres-sam-host.example.com:15672/`
3. Log in as **`admin`** using the default password of **`changeme`**.
4. If this is the first time RabbitMQ was started, make sure the first-time setup has completed before continuing, as explained Checking the RabbitMQ configuration on

page 4. The setup process uses the default `admin/changeme` login credentials to make changes to RabbitMQ.

5. Click the **Admin** tab and select **admin** from the list of RabbitMQ login users.
6. Click **Update this user**.
7. Type the new password in both password fields. Do not change the **Tags** field.
8. Click **Update User**.

   A message appears indicating the login failed. This is because you changed the password for the user that is currently logged on, making the password invalid.
9. Click **Log Out**.

   A new login form appears.
10. Type **`admin`** and the new password to confirm the password was successfully changed.

Changing the RabbitMQ publisher password

The publisher is a component in the presentation SAM domain that sends notifications into RabbitMQ. The RabbitMQ login for this process is `publisher` with the default password of `changeme`. To change this password, you must update both RabbitMQ and the SAM configuration with a new matching password.

Procedure

1. Log in to the RabbitMQ web-based management interface as **`admin`**.
2. Click the **Admin** tab and select **publisher** from the list of RabbitMQ login users.
3. Click **Update this user**.
4. Enter the new password in both password fields.
5. Click **Update User**.
6. To change the password in the SAM configuration:
   a. Log in to the host where the presentation SAM is installed.
   b. Type **`sm_edit conf/clientConnect.conf`**.
   c. In the `clientConnect.conf` file, look for a line beginning with `AMQP/*`

      It is similar to this:

      ```
      *:AMQP/
      *:publisher:<E-1.0>94FF41842043EDA1DE6E67FB08015B3D456752865F8B5
      D7DC63F259270B36092
      ```

   d. In the `AMQP/*"` line, replace the password field with the same password that you entered for the RabbitMQ publisher account.

      In this example, the password was set to `california`:

      ```
      *:AMQP/*:publisher:california
      ```

   e. Save `clientConnect.conf` and exit `sm_edit`.
7. Restart the publisher by connecting to your presentation SAM using `dmctl`.

   For example:

   a. **`$ dmctl -s INCHARGE-SA-PRES`**

b. **dmctl> invoke SM_AMQPNotificationPublisher::SM-AMQPNotificationPublisher stop**

Note

If the `stop` command hangs, verify that you entered the same password for the RabbitMQ publisher and SAM.

---

c. **dmctl> invoke SM_AMQPNotificationPublisher::SM-AMQPNotificationPublisher start**

8. In the RabbitMQ interface, click the **Connections** tab and verify the publisher is connected.

Changing the RabbitMQ ingestion password

Alert-ingestion is a component running in Tomcat that pulls notifications from RabbitMQ and inserts them in to the Notification cache. The RabbitMQ login for this service is `ingestion` with the default password of `changeme`. To change this password, you must update both RabbitMQ and the alerts-ingestion configuration with a new matching password.

Procedure

1. Log in to the RabbitMQ web-based management interface as **admin**.
2. Click the **Admin** tab and select **ingestion** from the list of RabbitMQ login users.
3. Click **Update this user**.
4. Enter the new password in both password fields.
5. Click **Update User**.
6. To change the password in alerts-ingestion:
   a. Stop the Smarts Tomcat service by typing **sm_service stop smarts-tomcat**.
   b. Using a text editor, open the file `smarts/tomcat/webapps/alerts-ingestion/WEB-INF/classes/rabbitmq-config.xml`.
   c. Find the password line that looks like this:

   ```
   <property name="password" value="changeme" />
   ```

   d. Replace **changeme** with the new password.

   In this example, the password was changed to `california`:

   ```
   <property name="password" value="california" />
   ```

   e. Restart the Smarts Tomcat service by typing **sm_service start smarts-tomcat**.
7. In the RabbitMQ interface, click the **Connections** tab and verify alerts-ingestion is connected. It may take a few minutes for Tomcat to start and for alerts-ingestion to connect.

UI cannot display alerts or Smarts topology when CAS is enabled

Follow these steps when the Central Authentication Service is enabled and the UI is unable to display information about alerts and objects in the Smarts topology.

Procedure

1. Verify the status of the Alerts EDAA, as described in Checking Alerts EDAA on page 5.
2. Verify the status of the Smarts EDAA, as described in Checking Smarts EDAA on page 5.
3. If Tomcat is running but queries to the EDAA result in the browser hanging, type this command on the SAM server:

   ```
   cd /opt/InCharge/SAM/smarts/bin
   ./runcmd_env.sh
   ```

4. In the `runcmd_env.sh` output:
   a. If `SM_CAS_SERVER` exists, verify this entry includes the protocol, fully-qualified domain name of the EMC M&R host, and the port number. Make sure there is nothing after the port number (not even a "/").
   b. If the https protocol is used, make sure the port is correct for https.
   c. Verify `SM_TOMCAT_SERVER` has an URL for Tomcat that is part of the SAM installation.
   d. Make sure localhost and 127.0.0.1 are NOT in `SM_CAS_SERVER` or `SM_TOMCAT_SERVER` entries. These environment variables must contain URLs that reach services from outside the host machine.

Wrong CAS host name in runcmd_env.sh

The UI informs you when you entered an incorrect Central Authentication Service host name in the runcmd_env.sh script.

This error message appears throughout the UI when you specify an incorrect host name for CAS in `runcmd_env.sh`:

```
java.util.concurrent.ExecutionException:com.emc.asd.mashupengine.utils
.JSONException:A JSONObject text must begin with '{' at 1 [character
2 line 1]
```

Procedure

- Specify the correct CAS host name in the `runcmd_env.sh` script.

Services remain registered and running after an uninstall

If you did not stop all services before performing an uninstall, the services are still running and using ports. If you do not want to reboot your system, you must manually stop these services to allow other services to use these ports.

An uninstall only removes those services that were configured during the installation of the software being uninstalled. If there are services configured by a second installation on the same host or by manually invoking `sm_service install`, those services are not removed.

Follow these procedures to stop services:

- Stopping services on Linux on page 14
- Stopping services on Windows on page 15

Stopping services on Linux

Follow this procedure if you did not stop all services before performing an uninstall. This procedure shows you examples of how to find the process IDs of the services that you need to stop on Linux systems.

Procedure

1. Find the process ID for elasticsearch, a Java process running `org.elasticsearch.bootstrap.ElasticSearch`

   For example:

   ```
   root      7223 10165  0 Jun05 ?        00:11:36 /opt/InCharge/SAM/
   smarts/jre/bin/java -cp /opt/InCharge/SAM/smarts/toolbox/
   elasticsearch/lib/elasticsearch-0.90.7.jar:/opt/InCharge/SAM/
   smarts/toolbox/elasticsearch/lib/*:/opt/InCharge/SAM/smarts/
   toolbox/elasticsearch/lib/sigar/* -Xms1g -Xmx1g -Xss256k -
   Djava.awt.headless=true -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -
   XX:CMSInitiatingOccupancyFraction=75 -XX:
   +UseCMSInitiatingOccupancyOnly -XX:+HeapDumpOnOutOfMemoryError -
   Delasticsearch -Des.pidfile=/opt/InCharge/SAM/smarts/local/logs/
   elasticsearch.pid -Des.foreground=yes -Des.path.home=/opt/
   InCharge/SAM/smarts/toolbox/elasticsearch -Des.path.data=/opt/
   InCharge/SAM/smarts/local/repos -Des.path.logs=/opt/InCharge/SAM/
   smarts/local/logs -Des.path.conf=/opt/InCharge/SAM/smarts/conf/
   elasticsearch org.elasticsearch.bootstrap.ElasticSearch
   ```

2. Find the process ID for Tomcat, a Java process running `org.apache.catalina.startup.Bootstrap`

   For example:

   ```
   root     12495 10165  0 Jun05 ?        00:31:17 /opt/InCharge/SAM/
   smarts/jre/bin/java -Xrs -Xms1g -Xmx3g -XX:PermSize=512m -XX:
   +HeapDumpOnOutOfMemoryError -classpath /opt/InCharge/SAM/smarts/
   tomcat/bin/bootstrap.jar:/opt/InCharge/SAM/smarts/tomcat/bin/
   tomcat-juli.jar:/opt/InCharge/SAM/smarts/local/l10n/spool:/opt/
   InCharge/SAM/smarts/local/l10n:/opt/InCharge/SAM/smarts/l10n -
   Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
   Djava.util.logging.config.file=/opt/InCharge/SAM/smarts/tomcat/
   conf/logging.properties -Djava.endorsed.dirs=/opt/InCharge/SAM/
   smarts/tomcat/endorsed -Dsmarts.remote.broker=localhost:426 -
   Dcatalina.logdir=/opt/InCharge/SAM/smarts/local/logs/tomcat -
   Dcatalina.base=/opt/InCharge/SAM/smarts/tomcat -
   Dcatalina.home=/opt/InCharge/SAM/smarts/tomcat -
   Djava.io.tmpdir=/opt/InCharge/SAM/smarts/tomcat/temp -
   Djava.awt.headless=true -Dcom.smarts.disable_ci_jni=true -
   Djava.security.egd=file:/dev/./urandom -
   Dcom.emc.msa.dmt.clientconnect=/opt/InCharge/SAM/smarts/conf/
   edaaClientConnect.conf -Dsmarts.cas.server=https://itops-
   dev-121.lss.emc.com:58443 -Dcom.smarts.incoming_protocol=1,0 -
   Dcom.smarts.outgoing_protocol=1,0 -Dcom.smarts.fips_mode=FALSE -
   Dsmarts.tomcat.server=https://itops-dev-120.lss.emc.com:8443
   org.apache.catalina.startup.Bootstrap --ignoreme start
   ```

3. Find the process ID for RabbitMQ, an Erlang beam process running with `-s rabbit`

   For example:

   ```
   root      7221 10165  0 Jun05 ?        00:24:03 /opt/InCharge/SAM/
   smarts/toolbox/OTP/erts-5.9.3.1/bin/beam.smp -W w -K true -A30 -P
   ```

```
1048576 -- -root /opt/InCharge/SAM/smarts/toolbox/OTP -progname
erl -- -home /root -- -pa /opt/InCharge/SAM/smarts/toolbox/
rabbitmq/ebin -noshell -noinput -s rabbit boot -sname rabbit@itops-
dev-120 -boot start_sasl -config /opt/InCharge/SAM/smarts/conf/rmq/
rabbitmq.config -kernel inet_default_connect_options
[{nodelay,true}] -sasl errlog_type error -sasl sasl_error_logger
false -rabbit error_logger {file,"/opt/InCharge/SAM/smarts/local/
logs/rabbit@itops-dev-120.log"} -rabbit sasl_error_logger
{file,"/opt/InCharge/SAM/smarts/local/logs/rabbit@itops-dev-120-
sasl.log"} -rabbit enabled_plugins_file "/opt/InCharge/SAM/smarts/
conf/rmq/enabled_plugins" -rabbit plugins_dir "/opt/InCharge/SAM/
smarts/toolbox/rabbitmq/plugins" -rabbit plugins_expand_dir "/opt/
InCharge/SAM/smarts/local/repos/rmq/rabbit@itops-dev-120-plugins-
expand" -os_mon start_cpu_sup false -os_mon start_disksup false -
os_mon start_memsup false -mnesia dir "/opt/InCharge/SAM/smarts/
local/repos/rmq/rabbit@itops-dev-120" --ignoreme
```

4. Use the process IDs to kill these three processes: **`$ kill -9 7223 12495 7221`**

Stopping services on Windows

Follow this procedure if you did not stop all services before performing an uninstall.

Procedure

1. Access the **Services Management Console** to see which services are still registered and running.
2. Stop the services from the **Services Management Console**.
3. Delete the SAM directory from `\InCharge\SAM`.
4. Manually remove the services using the command prompt tool name `sc`. For example:
   a. **`sc delete ic-broker`**
   b. **`sc delete smarts-tomcat`**
   c. **`sc delete ic-sam-server-pres`**

Retrieving the current elasticsearch PID

You can monitor the resource consumption of elasticsearch using the Linux `top -p <pid>` command. This procedure explains where to retrieve the current process ID of elasticsearch.

Procedure

- To determine the current process ID of elasticsearch, view `<SMARTSHOME>/local/logs/elasticsearch.pid`.

Ports that must be open between SAM and EMC M&R

Ensure that the firewall between the server hosting SAM and the servers hosting EMC M&R (Watch4net) are open. EMC M&R collects data from the Smarts domain managers. Configure the firewall settings to allow access across ports.

Ports in use with HTTP (no SSL encryption)

- Tomcat server in Service Assurance Manager (SAM): 8080
- Tomcat server in EMC M&R (Watch4net): 58080
- Smarts Broker: 426 (also used for SSL encryption)

- Smarts License Server: 1744 (also used for SSL encryption)
- Elasticsearch: 9200
- RabbitMQ: 15672

Ports in use with HTTPS enabled (SSL encryption)

- Tomcat server in SAM: 8443
- Tomcat server in EMC M&R (Watch4net): 58443
- Smarts Broker: 426
- Smarts License Server: 1744

Copyright © 2014-2015 EMC Corporation. All rights reserved. Published in USA.

Published January, 2015

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (https://support.emc.com).