



MURAL Software Upgrade Guide

Version: 4.2

Published: 2018-03-28

Copyright © 2018, Cisco Systems, Inc.

Americas Headquarters

Cisco Systems, Inc.

170 West Tasman Drive

San Jose, CA 95134-1706 USA

MURAL Software Upgrade Guide

<http://www.cisco.com>

Tel: 408 526-4000

800 553-NETS (6387)

Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

MURAL Software Upgrade Guide

Copyright © 2018, Cisco Systems, Inc. All rights reserved.

Table of Contents

1. Upgrade Overview	7
1.1 Before You Begin	7
1.2 Upgrade Package	7
2. Pre-Upgrade Tasks	9
2.1 Copying and Installing Image on Nodes	9
2.2 Taking Backups	12
2.3 Stopping Processes	16
3. Upgrade Tasks	19
3.1 Upgrade Image on the Nodes	19
3.2 Restart Processes on GSM Nodes	21
3.3 Apply Patches on MURAL Nodes	21
3.4 Upgrade MURAL and Bulkstats Databases on Insta Cluster	22
4. Post-Upgrade Tasks	25
4.1 Activate the New XML from the Master GSM Node	25
4.2 Apply Configurations to Nodes	33
4.3 Update the IBs	37
4.4 Modifying Behavior for Cell Sector Annotation	38
4.5 Enabling CSV Report	40
4.6 Adding Home Network for Outbound Roaming Report	42
4.7 Configure System Monitoring Interface	44
4.8 Restart Processes	47
4.9 Enable the Collector in the Live Mode	56

1. Upgrade Overview

MURAL provides Web-based reporting and analytics for deep packet inspection (DPI) data emerging from your network.

This document describes how to upgrade the MURAL application from release 4.1 to release 4.2.

1.1 Before You Begin

This document assumes that you have a working knowledge of Linux operating system. Before you begin the upgrade, we recommend that you review the Release Notes for 4.2.

1.2 Upgrade Package

The MURAL software upgrade package contains the following components:

- **An IMG image file.** For the image name and associated MD5 checksum, refer to the Release Notes.
- **ImageUpgrade.sh** script. The script is used to automatically upgrade image on all the nodes from master GMS node.
- **configBackup.sh** script. The script is used to take backup of configuration dumps from all the MURAL nodes.
- The XML file which is used by the master GMS node. This configuration file provides configuration settings for your setup based on the network topology.
- Any software patches that apply to the release. A complete list is provided in the Release Notes.
- Management information bases (MIBs).

Warning: Skipping a phase or performing the tasks out of sequence may cause an incorrect configuration and result in installation failure.

If upgrade step fails at any point or does not show output as expected in the guide, please contact Technical Support.

2. Pre-Upgrade Tasks

Before you begin the upgrade, perform the following tasks:

- "Copying and Installing Image on Nodes" below
- "Taking Backups" on page 12
- "Stopping Processes" on page 16

2.1 Copying and Installing Image on Nodes

Before starting the upgrade process, copy and install image on all the nodes using master GMS node.

1. Log into the master GMS node and execute the following commands to create a directory.

```
> en
# _shell
# mkdir /data/Mural4.2/
```

This creates a directory, where the image will be copied and installed.

2. From the master GMS node, copy the IMG file to /data/MURAL4.2 directory, and imageUpgrade.sh and configBackup.sh scripts to /data/ directory from location /opt/etc/scripts/gms/.

```
> en
# _shell
# cp /opt/etc/scripts/gms/imageUpgrade.sh /data
# cp /opt/etc/scripts/gms/configBackup.sh /data
```

3. Execute the following command on master GMS node to copy and install image on all the nodes:

```
# sh /data/imageUpgrade.sh <GMS XML File> <Image File>
```

For Example,

```
[admin@gms-1 ~]# sh /data/imageUpgrade.sh /config/gms/active-
```

```
configuration/resolved_active.xml /data/Mural4.2/image-  
atlas4.2.rc1.img
```

The output may resemble as follows:

```
imageUpgrade.sh:msg:IMGNAME == image-atlas4.2.rc1.img  
MD5SUM is: 5a522971ebbd5cf720998898def29fea  
 /data/Mural4.2/image-atlas4.2.rc1.img  
-----  
imageUpgrade.sh:msg: Network Name is : MgmtNW  
-----  
imageUpgrade.sh:msg: Node : 192.168.188.56 have  
sufficient empty space in /var FS - Required :  
4187688 KB, Available : 7302120 KB  
-----  
imageUpgrade.sh:msg: Node : 192.168.188.66 have  
sufficient empty space in /var FS - Required :  
4187688 KB, Available : 7611028 KB  
-----  
imageUpgrade.sh:msg: Copying Image file into Node :  
192.168.188.56 ...  
imageUpgrade.sh:msg: Image file successfully copied into Node  
:  
      192.168.188.56  
-----  
imageUpgrade.sh:msg: Copying Image file into Node :  
192.168.188.66 ...  
imageUpgrade.sh:msg: Image file successfully copied into Node  
:  
      192.168.188.66  
-----
```

```
Copying Remote Image upgrade script to 192.168.188.56 node ...

imageUpgrade.sh:msg:Image upgrade command has been submitted
on 192.168.188.56 ...

-----

Copying Remote Image upgrade script to 192.168.188.66 node ...

imageUpgrade.sh:msg:Image upgrade command has been submitted
on 192.168.188.66 ...

-----

.
.
.
imageUpgrade.sh:msg: Please check /data/image_upgrade.log file
and monitor remote_img_upgrade.sh process on target nodes
[admin@gms-1 ~]#
```

After the successful execution of the above commands, image installation process starts on all the nodes.

4. Execute the following command on other target nodes, except master GMS node, to verify the successful execution of the installation command.

```
[admin@gms-2 data]# ps -eaf | grep remote_img_upgrade.sh |
grep -v grep
admin 19791 1 0 11:01 ? 00:00:00 sh /data/remote_img_
upgrade.sh
[admin@gms-2 data]#
```

The preceding output shows that the installation command is running on this node. Run the same command after 15-20 minutes to verify that the image is successfully installed.

```
[admin@gms-2 data]# ps -eaf | grep remote_img_upgrade.sh |  
grep -v grep  
[admin@gms-2 data]#
```

The preceding output shows that the image installation is complete for this node. You can run the command on other target nodes to verify the successful installation.

2.2 Taking Backups

It is recommended to take back ups before proceeding with the upgrade.

Log in to each node and execute the following commands:

```
> en  
# _shell  
# mkdir /data/Backup_files
```

This creates a folder, where all the backed-up files will be saved.

2.2.1 Take a Backup of all IBs

Once you have created the folder to save the back up files on all the nodes.

Perform the following steps on the master name node:

1. Execute the following commands to take backup of all the IBs:

```
# cp -r /data/ib/inbox /data/Backup_files/  
# cp -r /data/ib/work /data/Backup_files/  
# hdfs dfs -get /IB /data/Backup_files/
```

2.2.2 Take a Backup of PGSQL Database

1. Execute the following commands on the master GMS node:

```
>en
# config terminal
(config)# pgsq1 file-path /data/Backup_files/pgsq1DB4.1_
gms.txt dbname gms_configuration backup
```

2. Execute the following commands on the master iNSTA node:

```
> en
# config terminal
(config)# pgsq1 file-path /data/Backup_files/pgsq1DB4.1_
rubix.txt dbname rubixdb backup
```

3. Execute the following commands on the master Name node:

```
> en
# config terminal
(config)# pgsq1 file-path /data/Backup_files/pgsq1DB4.1_
hive.txt dbname hivedb backup
```

2.2.3 Take a Backup of Active Configurations

Execute the following commands from the master GMS Node to take configuration backup of all the MURAL nodes.

```
[admin@gms-1 ~]# sh /data/configBackup.sh /config/gms/active-
configuration/resolved_active.xml
```

The output may resemble as follows:

```
configBackup.sh:msg: Network Name is : MgmtNW
-----
configBackup.sh:msg: Collecting Running Config from Nodes ...
-----
configBackup.sh:msg:Backup config for 192.168.188.52 node ....
configBackup.sh:msg:Backup config for 192.168.188.53 node ....
```

```

.
.
.
configBackup.sh:msg:Backup config for 192.168.188.62 node ....
configBackup.sh:msg:Backup config for 192.168.188.66 node ....
-----
configBackup.sh:msg: Collecting Running Config from Nodes : Done.
-----
configBackup.sh:msg: Creating tar ball of Config Backup files
collected from all the nodes ...
runningConf_col-1_201707191018.txt runningConf_col-2_
201707191018.txt
.
.
.
runningConf_ui-1_201707191018.txt
runningConf_ui-2_201707191018.txt
-----
configBackup.sh:msg: Config Backup is available as :
/data/configDumpDir/RunningConf_201707191018.tgz
-----

[admin@gms-1 ~]# ls -l
/data/configDumpDir/RunningConf_
201707191018.tgz
-rw-r--r-- 1 admin root 258441 Jul 19 10:19
/data/configDumpDir/RunningConf_201707191018.tgz
[admin@gms-1 ~]#

```

2.2.4 Take a Backup of Gateway Configurations

- Execute the following command from all the collector nodes to take a backup of gateway configurations for bulkstats:

```

# cli -t en 'show running' >
/data/Backup_files/`hostname`_configBackup.txt

```

```
# grep "bulkStatsFile" /data/Backup_files/`hostname`_
configBackup.txt >
/data/Backup_files/coll_bsGW_config.txt
# echo "write memory" >>
/data/Backup_files/coll_bsGW_config.txt
```

2.2.5 Take a Backup of Manually Modified Files

Perform the following procedure:

1. Execute the following commands on the master name node to create a folder for saving backup:

```
> en
# _shell
# mkdir -p /data/Backup_files/Patch_backup
```

2. Execute the following command on the master Name node:

```
# cp /data/CoreJob/config/solutionConfig.json /data/Backup_
files/
# cp /opt/catalogue/atlas/dynamicWhiteListConfig.json
/data/Backup_files/image_dynamicWhiteListConfig.json
# cp /data/ib/inbox/dynamicWhiteListConfig.json /data/Backup_
files/
# cp -r /data/patchspace/patch/4.1.rc1 /data/Backup_
files/Patch_backup/
# cp /opt/samples/yarn_conf/capacity-scheduler.xml.template
/data/Backup_files/
```

Execute the following command from all the UI node to take a backup of rubix app properties:

```
# cli -t en 'show running' >
/data/Backup_files/`hostname`_configBackup.txt
```

2.3 Stopping Processes

You must stop processes as explained in the following sections before proceeding with the upgrade.

2.3.1 Stop Oozie Jobs

- Log into the master Name node and execute the following commands:

```
> en
# _shell
# pmx subshell oozie stop jobname all
```

Note: Wait until all the jobs are stopped. To verify if all the jobs are stopped, execute the following command and ensure that no job is running:

```
# pmx subshell oozie show coordinator RUNNING jobs
```

For example:

```
[admin@MUR-COLL-2 ~]# pmx subshell oozie show coordinator
RUNNING jobs
No Jobs match your criteria!
[admin@MUR-COLL-2 ~]#
```

2.3.2 Stop Rubix Processes

- To disable all applications, execute the following commands on the corresponding UI nodes:

```
> en
# conf t
(config)# rubix modify-app <app-name> modify-instance 1
disable
(config)# rubix modify-app <app-name> disable
(config)# no pm process rubix launch auto
(config)# pm process rubix terminate
(config)# write memory
```

Where, <app-name> is as follows. Stop the applications in the same order as mentioned below.

For all the UI nodes:

- atlas

For both the RGE nodes:

- bulkstats
- reportAtlas
- rge
- ruleEngine

2.3.3 Stop Insta Processes

- Execute the following command on both the Insta nodes:

```
> en
# conf t
(config)# pm process insta terminate
```

2.3.4 Stop GMS Processes

- Execute the following commands on both the GMS nodes:

```
> en
# conf t
(config)# no pm process gms_server launch auto
(config)# no pm process pgsqlld launch auto
(config)# wr mem
(config)# pm process gms_server terminate
(config)# pm process pgsqlld terminate
```

2.3.5 Stop the Collector

- Execute the following commands on all the collector nodes:

```
> en
# conf t
```

```
(config)# no pm process collector launch auto  
(config)# pm process collector terminate  
(config)# write memory
```

3. Upgrade Tasks

Perform the following steps to upgrade MURAL from release 4.1 to release 4.2:

- "Upgrade Image on the Nodes" below
- "Restart Processes on GMS Nodes" on page 21
- "Upgrade MURAL and Bulkstats Databases on Insta Cluster" on page 22

3.1 Upgrade Image on the Nodes

Upgrade the image on the following nodes:

- Master and Standby NameNode
- Master and Standby node of the additional Collector (if any)
- Master and standby GMS nodes
- All the Rubix nodes including the RGE node
- Master and standby Insta nodes
- All Compute nodes, chassis-wise

Note: Ensure to upgrade image on Compute nodes chassis-wise. After upgrading the compute nodes on the first chassis, verify that the nodes have come up with new image and joined the YARN cluster. Only after this verification, proceed with the upgrade on the next chassis.

Perform the following steps on all the nodes in the same order as listed in the preceding section:

1. SSH login to each node as an admin user.
2. Execute the following commands to identify role of the individual nodes in HA cluster:

Note: Before executing these commands, note the role of each node to check if any master or standby nodes have interchanged their roles.

```
>en
```

```
# conf t
(config) # show cluster global brief
```

Run the following command to reboot the node:

```
(config)# write memory
(config)# reload
```

Note: Before executing this command on any other node(s) of the same cluster, wait for at least 10 to 20 minutes for the node to come up after reboot.

3. Run the following commands on all the nodes:

```
>en
# _shell
# mount -o remount,rw /
```

4. Upgrade the platform version on all the nodes:

```
>en
# _shell
# mount -o remount,rw /
# sed -i "s/Platform_4.2.4.rc2/platform_4.2.4.rc1/g"
/etc/solution_version
```

5. To verify if the new image is available on a node, execute the following commands:

```
host [master: standby] >en
host [master: standby] # conf t
host [master: standby] (config)# sho ver
```

The output may resemble to the following:

```
Product name:   Content Analytics
Product release:      4.2.rc1
Build   ID:       #1-dev
Build   date:     2017-12-10 17:24:54
```

Note: While upgrading the image of a node in a cluster, execute the following command on the master node of the nodes whose roles have changed. This is to ensure that the role of the master node is the same as it was before the upgrade.

```
host [master: standby] (config)# cluster master self
```

3.2 Restart Processes on GMS Nodes

1. Execute the following commands on both the GMS nodes:

```
>en
# conf t
(config)# pm process gms_server launch auto
(config)# pm process pgsqld launch auto
(config)# wr mem
(config)# pm process pgsqld restart
(config)# pm process gms_server restart
```

3.3 Apply Patches on MURAL Nodes

Perform the following steps to apply and verify patches on all the MURAL nodes:

1. Log into the master GMS node and download all the patches, which are released for 4.2.rc1, available in the /data directory.
2. Execute the following commands from the master GMS node to apply patches on all the MURAL nodes:

```
> en
# _shell
# mount -o remount,rw /
# pmx
Welcome to pmx configuration environment.
pm extension> subshell patch_management
pm extension (patch management)> set gms-file
```

```

/config/gms/active-configuration/resolved_active.xml
pm extension (patch management)> fetch /data/4.2.rc1.p1.tgz
Patch download successful
pm extension (patch management)> install patch 4.2.rc1.p1 at
all

```

3. Execute the following command to verify the successful application of patches on all the nodes:

```
pm extension (patch management)> show patch_history at all
```

The following sample may resemble to the output:

```

192.168.194.105:
File system read/write mode enabled.
(+) 4.2.rc1.p1 2018-01-02 00:16:09 +0000

```

3.4 Upgrade MURAL and Bulkstats Databases on Insta Cluster

After the image is installed successfully, wait for 15 minutes before upgrading the MURAL and bulkstats databases on the Insta cluster.

1. Perform the following steps to upgrade the schema for new XML for bulkstats cubes from the master iNSTA node:
 - a. Execute the following commands:

```

>en
# _shell
# cli -m config
(config)# insta infinidb upgrade
(config)# quit

```

- b. Wait for 15 minutes, and then execute the following commands:

```

#echo "insta infinidb get-status-info" | cli -m config |
grep -A3 "instance 0 service status :"

```

Ensure that the output contains the following:

```
Insta instance 0 service status : RUNNING
Insta instance 1 service status : RUNNING
Infinidb Adaptor status : Adaptor Running
```

- c. Execute the following command to restart iNSTA process:

```
# cli -m config
(config)# pm process insta restart
(config)# quit
```

2. Perform the following steps to restart iNSTA on the standby node:

- a. Execute the following commands:

```
>en
# _shell
# cli -m config
(config)# pm process insta restart
```

- b. Wait for a few minutes and check the status of infinidb by executing the following command:

```
(config)# insta infinidb get-status-info
(config)# quit
```

The output may resemble as follows:

```
Infinidb Install status : INSTALLED
Insta instance 0 service status : RUNNING
Insta instance 1 service status : RUNNING
```

3. Execute the following command on the master iNSTA node to check the installation status:

```
# cli -t "en" "conf t" "insta infinidb get-status-info" | grep
"Insta instance 0 service status"
```

The output may resemble as follows:

```
Insta instance 0 service status : RUNNING
```

Execute the following command:

```
# cli -t "en" "conf t" "insta infinidb get-status-info" | grep
"Infinidb Adaptor status"
```

The output may resemble as follows:

```
Infinidb Adaptor status : Adaptor Running
```

Execute the following command:

```
# cli -t "en" "conf t" "insta infinidb get-status-info" | grep
"Active insta nodes"
```

The output may resemble as follows:

```
Active insta nodes : <comma separated list of insta IPs>
```

Note the status as RUNNING and list of active iNSTA nodes in the output.

- Execute the following command on both the iNSTA nodes (master and standby) to ensure that 55555 port is in listen mode:

```
# netstat -napto | grep -i 55555
```

The output may resemble as follows:

```
tcp      0      0 :::55555      :::*
LISTEN          2730/insta      off (0.00/0/0)
```

4. Post-Upgrade Tasks

Read the following section to know about post- upgrade tasks.

4.1 Activate the New XML from the Master GMS Node

1. Perform the following steps to run MMI for generating new XML:
 - a. The following properties, which are configured through mmi, can be updated during mmi run:
 - `adaptor.<>.numthreads-` can be obtained by searching for the string `"num_threads"` in the config backup file from Master Namenode.

Example:

```
[admin@col-1 ~]# grep num_threads /data/Backup
files/'hostname'_configBackup.txt
internal set modify -
/nr/collector/instance/1/adaptor/bulkStats/num_
threads value int16 2
internal set modify -
/nr/collector/instance/1/adaptor/edrDNS/num_threads
value int16 8
internal set modify -
/nr/collector/instance/1/adaptor/edrflow/num_threads
value int16 8
internal set modify -
/nr/collector/instance/1/adaptor/edrhttp/num_threads
value int16 8
[admin@col-1 ~] #
```

- values for `tomcastInstanceMaxSize` and `initialJavaHeapSize` properties can be searched in config backup file from Master UI nodes (DPI as well as RGE):

Example:

```

[admin@ui-1 Backup_files]# grep -e
tomcatInstanceMaxSize -e initialJavaHeapSize
/data/Backup_files/`hostname`_configBackup.txt
rubix modify-app atlas modify-instance 1 set adv-
attribute initialJavaHeapSize value 150g
rubix modify-app atlas modify-instance 1 set adv-
attribute tomcatInstanceMaxSize value 150g
[admin@ui-1 Backup_files]#
[admin@rge-1 ~]# grep -e tomcatInstanceMaxSize -e
initialJavaHeapSize /data/Backup_files/`hostname`_
configBackup.txt
rubix modify-app bulkstats modify-instance 1 set
adv-attribute initialJavaHeapSize value 80g
rubix modify-app bulkstats modify-instance 1 set
adv-attribute tomcatInstanceMaxSize value 80g
rubix modify-app reportAtlas modify-instance 1 set
adv-attribute initialJavaHeapSize value 10g
rubix modify-app reportAtlas modify-instance 1 set
adv-attribute tomcatInstanceMaxSize value 50g
rubix modify-app rge modify-instance 1 set adv-
attribute initialJavaHeapSize value 10g
rubix modify-app rge modify-instance 1 set adv-
attribute tomcatInstanceMaxSize value 50g
rubix modify-app ruleEngine modify-instance 1 set
adv-attribute initialJavaHeapSize value 5g
rubix modify-app ruleEngine modify-instance 1 set
adv-attribute tomcatInstanceMaxSize value 5g
[admin@rge-1 ~]#

```

- Values for rubixFQDN and TimeZone properties can be searched in config backup file from Master DPI node.

Example:

```
[admin@ui-1 Backup_files]# grep "rubix modify-app
atlas " /data/Backup_files/`hostname`_
configBackup.txt | grep -e rubixFQDN -e timeZone
rubix modify-app atlas set adv-attribute rubixFQDN
value demo.cisco.com
rubix modify-app atlas set adv-attribute timeZone
value UTC
[admin@ui-1 Backup_files]#
```

- Values for mailHost, mailPort, mailNotificationSender and rge-mailSupport (mailNotificationRecipient) properties can be searched in config backup file from Master RGE node.

Example:

```
[admin@rge-1 ~]# grep -e mailNotificationHost -e
mailNotificationPort -e mailNotificationRecipient -e
mailNotificationSender /data/Backup_
files/`hostname`_configBackup.txt
rubix modify-app rge set adv-attribute
mailNotificationHost value mx1.guavus.com
rubix modify-app rge set adv-attribute
mailNotificationPort value 25
rubix modify-app rge set adv-attribute
mailNotificationRecipient value deploy@guavus.com
rubix modify-app rge set adv-attribute
mailNotificationSender value rubix-admin@guavus.com
[admin@rge-1 ~]
```

- Execute the following command:

```
# sh /opt/etc/scripts/mmi/cisco_mural_deployment_
wizard.sh
```

Use the existing active.xml as input XML to retain the hardware configuration. Note the name of the generated XML provided on the first page.

The MURAL Deployment wizard, as illustrated in the following section, is launched on the terminal to accept various configuration inputs.

```

Welcome to Cisco Mural Deployment Wizard
GNS Hardware XML Filename :
/config/gns/Mural_Hardware.xml
Output XML File Name :
/config/gns/Mural_Application.xml
Pack Type : (X) Standard
            ( ) Medium
            ( ) Starter
Additional Apps :
(X) Bulkstats
(X) Anomaly
(X) Telemetry
(X) ServiceID Report
(X) Outbound Roaming Report
(X) TLA-SMI Report
(X) Rulebase Report
(X) Ruledef Report
(X) Open Platform Support
Exit Next

```

General instructions to use the MURAL Deployment wizard:

- Use arrow keys to navigate in the Deployment wizard.
- Use Space-bar to choose the selected item(s).



- Press **Enter** after selecting **Next**, **Previous** or **Exit** to move into next screen, previous screen or exit from the Deployment wizard, respectively.
- Selected items appear with swapped colors for text and background.

```

Map Clusters to Applications
GMS-CLUS : [X] GMS
           [ ] Additional Collector
           [ ] Compute
           [ ] INSTA
           [ ] Rubix-OP1
           [ ] Other Rubix Apps
           [ ] NameNode & Collector

COL-CLUS : [ ] GMS
           [ ] Additional Collector
           [ ] Compute
           [ ] INSTA
           [ ] Rubix-OP1
           [ ] Other Rubix Apps
           [X] NameNode & Collector

HW-CLUS : [ ] GMS
           [ ] Additional Collector
           [X] Compute
           [ ] INSTA
           [ ] Rubix-OP1
           [ ] Other Rubix Apps
           [ ] NameNode & Collector

INSTA-CLUS : [ ] GMS
             [ ] Additional Collector
             [ ] Compute
             [X] INSTA
             [ ] Rubix-OP1
             [ ] Other Rubix Apps
             [ ] NameNode & Collector

UI-CLUS : [ ] GMS
          [ ] Additional Collector
          [ ] Compute
          [ ] INSTA
          [X] Rubix-OP1
          [X] Other Rubix Apps
          [ ] NameNode & Collector
    
```

- Set Global variables based upon values collected above from config backup files of Master Name-node and UI nodes.

The following image illustrates the screen:

```

Global Variables Configuration
col-clu -> Collector (0)
adaptor.bulkStats.numThreads:
    2
adaptor.bulkStats.input.FileConfig.backupDeleteTimeout:
    24
adaptor.edfFlow.profile:
    /opt/deployment/GMS_Templates/collector/EdfFlowParquetReaderConfig.xml
adaptor.edfFlow.numThreads:
    8
adaptor.edmhttp.profile:
    /opt/deployment/GMS_Templates/collector/EdmhttpParquetReaderConfig.xml
adaptor.edmhttp.numThreads:
    8
adaptor.edmsn.profile:
    /opt/deployment/GMS_Templates/collector/DnsParquetReaderConfig.xml
adaptor.edmsn.numThreads:
    8
cmp-clu -> Compute (0)
compute.dfs.datanode.data.dir:
    /data/hadoop-admin
el-clu -> Rubix (0)
application.atlas.anonymizedSubID:
    false
application.atlas.cachePersistToDisk:
    true
application.atlas.rubixInstance.1.tomcatInstanceMaxSize:
    150g
application.atlas.rubixInstance.1.initialJavaHeapSize:
    150g
rge-clu -> Rubix (2)
application.ruleEngine.rubixInstance.1.tomcatInstanceMaxSize:
    5g
application.ruleEngine.rubixInstance.1.initialJavaHeapSize:
    5g
application.bulkStats.cachePersistToDisk:
    true
    
```

```

Global Variables Configuration
application.bulkstats.rubixInstance.1.tomcatInstanceMaxSize:
80g
application.bulkstats.rubixInstance.1.initialJavaHeapSize:
80g
application.report4tlos.anonymizeSubID:
false
application.report4tlos.rubixInstance.1.tomcatInstanceMaxSize:
50g
application.report4tlos.rubixInstance.1.initialJavaHeapSize:
10g
application.rge.rubixInstance.1.tomcatInstanceMaxSize:
50g
application.rge.rubixInstance.1.initialJavaHeapSize:
10g
ins-cis -> Solution (0)
Timezone: UTC
FQDN: demo.cisco.com
rgmailSupport:
utlon.merod@guavis.com
mailSender: mural-admin@guavis.com
mailHost: 172.30.12.102
mailPort: 25
Previous Exit Generate & Exit Page 2 of 2

```

- Select **Generate & Exit**, and press the **Enter** key to create the final XML.

After successful execution, final XML (/config/gms/Mural_Application.xml) is generated.

2. Manually update the site specific collector Property value using vi editor in the Mural_Application.xml, generated in the preceding step.
 - a. Log into master NameNode as admin.
 - b. Execute the following commands:

Note the output values of the following commands.

```

> en
# _shell
# confFile=/data/Backup_files/`hostname`_configBackup.txt
# grep "edr.* filename-format " $confFile | awk '{print
$(NF-1)"\t"$NF}' | sort -u
filename-format %DC_ *_*_MM%DD%YYYY%hh%mm%ss_*.gz
# grep "edr.* transfer-filename-fmt " $confFile | awk '
{print $(NF-1)"\t"$NF}' | sort -u
transfer-filename-fmt %DC_ *_*_MM%DD%YYYY%hh%mm%ss_

```

```

*.gz.tmp
# grep "edr.*output-directory" $confFile | awk '{print
$(NF-1)"\t"$NF}' | sort -u
output-directory
/data/collector/1/output/edrdns/%y/%m/%d/%h/%mi/
gurgaon.EDRDNS.
output-directory
/data/collector/1/output/edrflow/%y/%m/%d/%h/%mi/
gurgaon.EDRFLOW.
output-directory
/data/collector/1/output/edrhttp/%y/%m/%d/%h/%mi/
gurgaon.EDRHTTP.
# grep "edr.*-expiry-" $confFile | awk '{print $(NF-
1)"\t"$NF}' | sort -u
backup-file-expiry-period 24
    
```

3. Update GMS XML using the vi editor from the master GMS node.

For example, `/config/gms/Mural Application.xml`.

Search for all the occurrences of properties in the GMS XML as mentioned in the following table and specify corresponding values displayed in the preceding output (step 3).

The following table lists the properties to be searched for in the GMS XML and their mapping to the properties and values displayed in the preceding output (step 3):

Properties in the GMS XML	Properties in the Out-put	Value Displayed in the Out-put
File.fileNameFormat	filename-format	%DC_*_*_ %MM%DD%YYYY%h- h%m- m%ss_*.gz

Properties in the GMS XML	Properties in the Output	Value Displayed in the Output
File.transferFileNameFormat	transfer-file-name-fmt	%DC_*_*_%MM%DD%YYYY%h- h%m-m%ss_*.gz.tmp
edr.*.output.directory	output-directory	/data/collector/1/output/edrdns/%y/%m/%d/%h/%mi/ gurgaon .EDRDNS.
		/data/collector/1/output/edrflow/%y/%m/%d/%h/%mi/ gurgaon .EDRFLOW.
		/data/collector/1/output/edrhttp/%y/%m/%d/%h/%mi/ gurgaon .EDRHTTP.
edr.*.- backupUpDeleteTimeOut	backupfile-expiry-period	24
autoBinSlide	disable	

4. Execute the following commands to copy data from standby collector to master node:

```
> en
# _shell
# echo show running-config full | cli -m config | grep
"flowFile input-directory" | awk '{print $9}'
# echo show running-config full | cli -m config | grep
"httpFile input-directory" | awk '{print $9}'
# echo show running-config full | cli -m config | grep
"bulkStatsFile" |grep "input-directory" | awk '{print $9}'
# scp <dir>/* admin@<collector-vip>:<dir>
```

Here, `dir` contains the output of these commands.

5. From the master GMS node, activate the XML that has only collector related changes in templates:

```
> en
# _shell
# cli -m config
(config)# gms config <New-XML> activate
```

Note: Allow the runtime activation to complete so that it applies the new template changes. This process may take 5 to 7 minutes as the activation process applies the new changes as per the new activated XML and will only stop and start Collector Process on all 4 Collector nodes including the NameNodes. Execute the following commands to verify the status:

```
> en
# _shell
# cli -m config
(config)# gms config-runtime show status all
```

If any error is encountered, correct the error and execute the following command:

```
(config)# gms config-runtime cluster cluster-name
<clustername> all
```

Here, <clustername> refers to the cluster to which the erroneous nodes belong. If the erroneous nodes belong to different clusters, execute the preceding command on all those clusters.

4.2 Apply Configurations to Nodes

- Perform the following steps on both the name nodes (as specified):
 - a. Execute the following commands to verify the difference between the current and earlier configuration:

```
> en
# _shell
# cli -t "en" "conf t" "show running full" >
```

```
/data/Backup_files/`hostname`_configCurrent.txt
```

- b. Verify the difference between the current (`/data/Backup_files/*_con-figCurrent.txt`) and previous (`((/data/Backup_files/*_configBackup.txt)) configuration files.`

Execute the commands that are found.

Note: Following are the expected commands to be executed on oozie subshell due to feature updates. Take a note of default values corresponding to the properties as these commands will overwrite the values.

Execute the following command to set jobStart time of RuledefDataTransfer job:

```
# hdfs dfs -cat /data/CoreJob/done.txt
```

The output may resemble as follows:

```
2016-07-28T08:00Z
```

NOTE: if RulebaseReport was not configured earlier then set jobStart time of GenericReportJobDaily job as well.

Note the timestamp in the preceding output and execute the following commands to ensure that RuledefDataTransfer job (and GenericReportJobDaily) always starts from 00:00 hour of the next day till CoreJob is executed.

Execute the following commands to set the start time for generation of CSV reports.

Note: Ensure that the start time of the job in the following command is set to an hour later to that of the previous output.

For example, in the preceding output, the time stamp is set to `2016-07-28T08:00Z`, and the timestamp in the following output is `2016-07-28T09:00Z`. This is done to ensure correct aggregation of data in jobs.

```
# pmx subshell oozie
pm extension (oozie)>
set job RuledefDataTransfer attribute jobStart 2016-07-29T00:00Z
pm extension (oozie)> set job GenericReportJobDaily attribute jobStart 2016-07-29T00:00Z
pm extension (oozie)>set dataset aggr_daily_report attribute startTime 2016-07-28T09:00Z
```

Note the timestamp in the preceding output and execute the following commands to ensure that OfflineEngine_OutboundReport job always starts from next hour till CoreJob is executed.

```
pm extension (oozie)> set job OfflineEngine_OutboundReport attribute jobStart 2016-07-28T01:00Z
```

Add dataset to cleanupOutput job:

```
corejob_report_output
pm extension (oozie)> set job CleanupOutput action CleanupDatasetAction attribute cleanupDatasets corejob_report_output
pm extension (oozie)> set dataset corejob_report_output attribute cleanupOffset 3
pm extension (oozie)> set dataset corejob_report_output attribute leaveDonefile false
pm extension (oozie)> quit
#
```

Execute the following commands to set the Jobstart time of CleanupOutput job

```
# hdfs dfs -cat /data/CleanupOutput/done.txt
```

The output may resemble as follows:

```
2016-07-28T00:00Z
```

Note the timestamp in the preceding output, and execute the following commands:

```
# cli -m config
(config)# internal set modify -
/tps/process/oozie/jobs/CleanupOutput/attribute/jobS
tart /value value string 2016-07-28T00:00Z
```

Execute the following command to set starttime for dataset ruledef_report_output,outbound_report_output:

```
# pmx subshell oozie set dataset ruledef_report_
output attribute startTime 2016-07-28T00:00Z
# pmx subshell oozie set dataset outbound_report_
output attribute startTime 2016-07-28T00:00Z
```

NOTE: if RulebaseReport was not configured

Run the following as well:

```
# pmx subshell oozie set
dataset generic_report_output_daily attribute
startTime 2016-07-28T00:00Z
```

- c. Drop the EdrFlow Hive db to update the hive to accept the new sn-ruledef-name field in edrflow data

Perform the following steps from Master name node servers:

Steps for Hive:

```
# hive
hive> drop table edrflow;
hive> quit;
```

- d. Execute the following command on the master name node:

```
# cli -m config
(config)# pm process tps restart
```

```
(config)# quit
#
```

Wait for 10 minutes and execute these commands on standby namenode.

Run the following commands to verify Hadoop status:

```
# hdfs dfsadmin -report | head -12
```

The output may resemble to the following sample:

```
Configured Capacity: 5624751722496 (5.12 TB)
Present Capacity: 5326934437888 (4.84 TB)
DFS Remaining: 1634512154624 (1.49 TB)
DFS Used: 3692422283264 (3.36 TB)
DFS Used%: 69.32%
Under replicated blocks: 852
Blocks with corrupt replicas: 0
Missing blocks: 0
-----
Datanodes available: 4 (4 total, 0 dead)
```

Here, 4 denotes the number of Compute nodes available in the setup.

4.3 Update the IBs

Perform the following steps on Master Namenode:

1. Execute the following commands to update the DPI IB:

```
> en
# _shell
# pmx subshell aggregation_center
(aggregation_center)# update all ibs from image
(aggregation_center)# generate urlcat ib from image
(aggregation_center)# show urlcat version
```

```
(aggregation_center)# generate all ibs
(aggregation_center)# push all ibs
(aggregation_center)# quit
```

2. Execute the following commands to update the Bulkstats IB:

```
# pmx subshell bulkstats
(bulkstats)# update all ibs from image
(bulkstats)# generate all ibs
(bulkstats)# push all ibs
(bulkstats)# quit
```

3. Execute the following commands to update the Anomaly IB:

```
# pmx subshell anomaly
(anomaly)# update all ibs
(anomaly)# quit
```

4.4 Modifying Behavior for Cell Sector Annotation

You can categorize the Cell Sectors, which are not present in the IB configuration into Other cells, and GGSN and SGSN which are not present in the IB configuration into Miscellaneous.

The values of Cell Sectors, GGSN and SGSN to be retained in data are specified in the following IBs:

- cellLocationInfo.list for Cell Sectors
- ipGGSN.map for GGSN
- ipSGSN.map for SGSN

Perform the following steps for categorizing the Cell Sectors, GGSN and SGSN:

1. Execute the following command to create the backup of Annotations.json:

```
# mount -o remount,rw /
# cp -p /opt/etc/oozie/EDR/app/Annotations.json
/opt/etc/oozie/EDR/app/Annotations.json.org
```

- Execute the following command to check the current values assigned to groupSGSNAsMisc, groupGGSNAsMisc and emitUsingMCCMNC:

```
# grep -e groupSGSNAsMisc -e groupGGSNAsMisc -e  
emitUsingMCCMNC /opt/etc/oozie/EDR/app/Annotations.json
```

The output may resemble as follows:

```
"groupGGSNAsMisc":"false"  
"groupSGSNAsMisc":"false"  
"emitUsingMCCMNC":"false"
```

- Execute the following command to set the value of emitUsingMCCMNC, groupGGSNAsMisc, and groupSGSNAsMisc as true:

```
# sed -  
i's/"emitUsingMCCMNC":\s*"false"/"emitUsingMCCMNC":"true"/'  
/opt/etc/oozie/EDR/app/Annotations.json  
# sed -i  
's/"groupGGSNAsMisc":\s*"false"/"groupGGSNAsMisc":"true"/'  
/opt/etc/oozie/EDR/app/Annotations.json  
# sed -i  
's/"groupSGSNAsMisc":\s*"false"/"groupSGSNAsMisc":"true"/'  
/opt/etc/oozie/EDR/app/Annotations.json
```

- Execute the following command to validate the value of groupSGSNAsMisc, groupGGSNAsMisc and emitUsingMCCMNC:

```
# grep -e groupSGSNAsMisc -e groupGGSNAsMisc -e  
emitUsingMCCMNC /opt/etc/oozie/EDR/app/Annotations.json
```

Ensure that the output shows these values as "true":

```
"groupSGSNAsMisc":"true"  
"emitUsingMCCMNC":"true"  
"groupGGSNAsMisc":"true"
```

4.5 Enabling CSV Report

This section explains how to enable reports selected under **Additional Apps** on the deployment wizard in "Activate the New XML from the Master GMS Node" on page 25

- For **ServiceID** report:
 1. Execute the following commands on master and all standby namenodes:

```
> en
# _shell
# python
/opt/etc/oozie/SolutionConfigs/EnableServiceId.py yes
```

2. Edit or create the file `/data/work/serverfile_ServiceId` on master namenode with destination details in the format shown below.

```
<ip_address>,admin,admin@123,/data/serviceIdReports/
```

For Example, `192.168.193.205, admin, admin@123, /data/serviceIdReports/`

3. Copy the file, `serverfile_ServiceId` at `/data/work/` on all the standby namenodes.

- For **Rulebase** report:
 1. Execute the following commands on master and all standby namenodes:

```
> en
# _shell
# python /opt/etc/oozie/SolutionConfigs/EnableRulebase.py
yes
```

2. Edit or Create the file `/data/work/serverfile_Rulebase` on master namenode with destination details in the format shown below.

```
<ip_address>,admin,admin@123,/data/rulebaseReports/
```

For example, 192.168.193.205, admin, admin@123, /data/rulebaseReports/.

3. Copy the file, `serverfile_Rulebase` at `/data/work/` on all the standby namenodes.

- For **RuleDef** report:

1. Execute the following commands on master and all standby namenodes:

```
> en
# _shell
# python /opt/etc/oozie/SolutionConfigs/EnableRuledef.py
yes
```

2. Edit or Create the file `/data/work/serverfile_Ruledef` on master namenode with destination details in the format shown below.

`<ip_address>,admin,admin@123,/data/ruledefReports/`

For example, 192.168.193.205, admin, admin@123, /data/ruledefReports/.

3. Copy the file, `serverfile_Ruledef` at `/data/work/` on all the standby namenodes.

- For **Outbound Roaming** report:

1. Execute the following commands on master and all standby name nodes:

```
> en
# _shell
# python
/opt/etc/oozie/SolutionConfigs/EnableOutboundRoaming.py
yes
```

2. Edit or Create the file `/data/work/serverfile_OutboundReport` on

master namenode with destination details in format shown below.

```
<ip_address>,admin,admin@123,/data/OutboundReport/
```

For example,

```
192.168.193.205, admin, admin@123, /data/OutboundReport/
```

3. Copy the file, `/data/work/serverfile_OutboundReport` at `/data/work/` on all the standby namenodes.

- For **TLS-SNI** report:

1. Execute the following commands on master and all standby namenodes:

```
> en
# _shell
# python /opt/etc/oozie/SolutionConfigs/EnableTlsSni.py
yes
```

2. Edit or Create the file `/data/work/serverFile_SNIReports` on master namenode with destination details in format shown below.

```
<ip_address>,admin,admin@123,/data/SNIReports/
```

For example,

```
192.168.193.205, admin, admin@123, /data/SNIReports/
```

3. Copy the file, `serverFile_SNIReports` at `/data/work/` on all the standby namenodes .

4.6 Adding Home Network for Outbound Roaming Report

You can generate a report to understand the usage of the subscribers who are roaming (outbound roamers). MURAL provides device, content and Service Provider usage reports per roaming partner based on the feature.

For the outbound roaming report feature, you must configure the mobile country code (MCC)/mobile network code (MNC) of the required outbound roamers.

Perform the following steps:

1. Execute the following commands on both the Name nodes to verify if the outbound roaming report feature is enabled:

```
> en
# _shell
# grep -e emitCubes /opt/etc/oozie/EDR/app/Annotations.json
"emitCubes":"true",
```

Note: The value of "emitCubes" as "true" signifies that the feature is enabled.

2. Execute the following commands to add the MCC/MNC of the outbound roamers:

```
> en
# conf t
(config)# pmx subshell aggregation_center
pm extension (aggregation center)> edit ib homeNetwork.list
add
mcc: <mobile country code of roamer>
mnc: <mobile network code of roamer>
```

Add all the possible MCC/MNC combinations. To view all the added values:

```
pm extension (aggregation center)> show ib homeNetwork.list
1 [440][001]
```

3. Execute the following commands to add the IB to the system:

```
pm extension (aggregation center)> push ib homeNetwork.list
```

Note: Execute the code, `opt/etc/scripts/validateHomeNwList.py` before running the core job to check any overlapping MNC prefixes for a given MCC in the home network list.

For example,

```
# python /opt/etc/scripts/validateHomeNwList.py
```

The output may resemble to the following sample:

```

Fetching information from /data/ib/inbox/homeNetwork.list
Fetching information from /data/ib/inbox/globalNetwork.list
For 440:002 also exits 440:00
For 310:02 also exits 310:020

```

4.7 Configure System Monitoring Interface

System Monitoring Interface (SMI) is a dashboard that can be used to display MURAL's system-related health information for the rolling 24-hours period.

Benefits of SMI:

- The system administrator can check the health of the MURAL system and contact Technical Support if a condition is not being met.
- The system administrator can use the dashboard to provide assurance to customer that MURAL services are working fine as per the benchmarking.

Note: You can install SMI only in a Standard pack or a Medium pack.

To configure SMI, perform the following steps:

1. Log into the master RGE node. Use local IP address of Master RGE node instead of VIP for SSH connection.
2. Execute the following command to copy the active GMS file from the GMS node to the RGE master node:

```

> en
# _shell
# cd /var/home/root
# scp admin@<GMS Master IP/Hostname>:/config/gms/active-
configuration/resolved_active.xml .

```

3. Execute the following commands to install SMI on the RGE master node:

```

# cd /var/home/root
# tar -xvzf pkg_dashboard.tgz
# ./install_dashboard.sh

```

The output may resemble as follows:

```
-----
install_dashboard.sh:msg: Usage.....
-----
install_dashboard.sh:input: Enter TAR Filename (with full path)
: /var/home/root/dashboard.tgz
install_dashboard.sh:input: Enter GMS XML Filename (with full
path) : /var/home/root/resolved_active.xml
install_dashboard.sh:input: Enter Management Network Name used
in above XML [MgmtNW]: MgmtNW
install_dashboard.sh:msg: Network Name is : MgmtNW
-----
install_dashboard.sh:msg: Extracting tar ball into node ...
.
.
.
Stopping pm: [ OK ]
Starting pm: [ OK ]
install_dashboard.sh:msg: Dashboard configuration is done
successfully!!!!
install_dashboard.sh:msg: Start Dashboard from UI by
http://<This node IP>:15443
#
```

Note: Provide the value for "Management Network Name" attribute as specified in the resolved_active.xml file.

4. Execute the following commands to check if SMI processes have started successfully:

To check if grafana is up:

```
# ps -ef | grep grafana
```

The output may resemble as follows:

```
admin  4300 2653 0 14:43 ?          00:00:00
/opt/grafana/bin/grafana-server -homepath=/opt/grafana
```

```
admin  6669 31484 0 14:44 pts/1      00:00:00 grep grafana
```

- To check if graphite and carbon-cache are up:

```
# ps -ef|grep graph
```

The output may resemble as follows:

```
admin  3313    2653    0 14:43 ?          00:00:00
/usr/local/bin/python /opt/graphite/bin/carbon-cache.py -
-nodaemon --logdir=/var/log --pidfile=/var/run/carbon-
cache.pid start
admin  5546    2653    1 14:43 ?          00:00:00
/usr/local/bin/python /usr/local/bin/gunicorn_django --
bind=127.0.0.1:12040 --log-file=/var/log/gunicorn-
graphite/gunicorn.log --preload --
pythonpath=/opt/graphite/webapp/graphite --
settings=settings --workers=3 --pid=/var/run/gunicorn-
graphite/gunicorn-graphite.pid
admin  5753    5546    0 14:43 ?          00:00:00
/usr/local/bin/python /usr/local/bin/gunicorn_django --
bind=127.0.0.1:12040 --log-file=/var/log/gunicorn-
graphite/gunicorn.log --preload --
pythonpath=/opt/graphite/webapp/graphite --
settings=settings --workers=3 --pid=/var/run/gunicorn-
graphite/gunicorn-graphite.pid
admin  5773    5546    0 14:43 ?          00:00:00
/usr/local/bin/python /usr/local/bin/gunicorn_django --
bind=127.0.0.1:12040 --log-file=/var/log/gunicorn-
graphite/gunicorn.log --preload --
pythonpath=/opt/graphite/webapp/graphite --
settings=settings --workers=3 --pid=/var/run/gunicorn-
```

```

graphite/gunicorn-graphite.pid
admin  5779    5546    0 14:43 ?        00:00:00
/usr/local/bin/python /usr/local/bin/gunicorn_django --
bind=127.0.0.1:12040 --log-file=/var/log/gunicorn-
graphite/gunicorn.log --preload --
pythonpath=/opt/graphite/webapp/graphite --
settings=settings --workers=3 --pid=/var/run/gunicorn-
graphite/gunicorn-graphite.pid
admin  6702 31484    0 14:44 pts/1    00:00:00 grep
graph

```

Note the running statuses of the graphite and carbon-cache processes as highlighted in the preceding sample output.

- Execute the following commands:

```

# cli -m config
(config)# cluster master self
(config)# quit

```

4.8 Restart Processes

- Run the following commands on all the collector nodes to restore the collector bulkstats configuration from configuration backup:

```

> en
# _shell
# echo -e "\nen\n\nconf t\n\n" > /data/Backup_files/coll_
bsConfigRestore.cli
# grep "add-file-if bulkStatsFile" /data/Backup_
files/`hostname`_configCurrent.txt | sed 's/add/delete/' >>
/data/Backup_files/coll_bsConfigRestore.cli
# cat /data/Backup_files/coll_bsGW_config.txt >> /data/Backup_
files/coll_bsConfigRestore.cli
# cli -x -h /data/Backup_files/coll_bsConfigRestore.cli

```

For more information about gateway configurations, see [Take a Backup of Gateway Configurations](#)

2. Perform the following steps to configure SSH session limit:
 - a. Execute the following commands to set the concurrent SSH limit to 50:

```
> en
# conf t
(config)# ssh server max-startups 50
(config)# ssh server max-sessions 50
(config)# write memory
(config)# pm process sshd restart
```

- b. Repeat steps 1 through 2 on the standby Collector node.

3. Log into Collector nodes and restart collector:

```
> en
# conf t
# pm process collector restart
```

4. Log into corresponding UI nodes, and execute the following steps to enable reverse proxy on your setup.

This section is optional, if you do not require to use reverse proxy in your setup, you can skip these steps.

Note: You must perform all the steps if you are enabling reverse proxy on your setup for the first time. You can skip steps marked as **(Optional)** if reverse proxy was already enabled on your setup before the upgrade.

However, performing optional steps in either case will not cause any issue in the upgrade.

1. Execute the following commands on all the UI and RGE nodes to take backup of the files:

- /opt/tms/lib/web/conf.d/httpd.conf
- /var/opt/tms/output/httpd.conf
- /opt/tms/rubix-atlas4.2/config/config.json
- /opt/tms/rubix-atlas4.2/config/app-switcher-config.json
- /opt/tms/rubix-bulkstats-atlas4.2/config/config.json
- /opt/tms/rubix-bulkstats-atlas4.2/config/app-switcher-config.json
- /opt/tms/rubix-bulkstats-atlas4.2/index.html
- /opt/tms/apache-tomcat/apache-tomcat-7.0.41/conf/web.xml

For example,

```
> en
# _shell
# mkdir /data/backup_reverse
# cp /opt/tms/lib/web/conf.d/httpd.conf /data/backup_reverse
```

2. Execute the following command to replace the file, /opt/tms/lib/web/conf.d/httpd.conf with the new file on all UI and RGE nodes.

```
# cp /opt/tms/lib/web/conf.d/httpd.conf_proxy
/opt/tms/lib/web/conf.d/httpd.conf
```

3. Execute the following commands to update the url in the httpd file on all the UI and RGE nodes.

Note: Before updating the URL in the httpd file, you must have complete information about the the rubix applications configured for the setup.

```
> en
# _shell
# sed -i 's/\$/g' /opt/tms/lib/web/conf.d/httpd.conf
```

```
# sed -i "s#CLI_REPLACE_ANOMALYSOLUTIONURL#<RGE-NODE
URL>:50443#g" /opt/tms/lib/web/conf.d/httpd.conf
# sed -i "s#CLI_REPLACE_BULKSTATSURL#<RGE-NODE
URL>:20443#g" /opt/tms/lib/web/conf.d/httpd.conf
# sed -i "s#CLI_REPLACE_RGESOLUTIONURL#<RGE-NODE
URL>:30443#g" /opt/tms/lib/web/conf.d/httpd.conf
# sed -i "s#CLI_REPLACE_MURALDPIURL#<UI-NODE
URL>:6443/#g" /opt/tms/lib/web/conf.d/httpd.conf
```

Note: There is an extra '/' at the end of <UI URL>:6443, where UI URL is **ucsd.cisco.com** and RGE URL is **ucsd-report.cisco.com**.

```
sed -i "s#CLI_REPLACE_ANOMALYSOLUTIONURL#https://ucsd-
report.cisco.com:50443#g"
/opt/tms/lib/web/conf.d/httpd.conf
sed -i "s#CLI_REPLACE_BULKSTATSURL#https://ucsd-
report.cisco.com:20443#g"
/opt/tms/lib/web/conf.d/httpd.conf
sed -i "s#CLI_REPLACE_RGESOLUTIONURL#https://ucsd-
report.cisco.com:30443#g"
/opt/tms/lib/web/conf.d/httpd.conf
sed -i "s#CLI_REPLACE_
MURALDPIURL#https://ucsd.cisco.com:6443/#g"
/opt/tms/lib/web/conf.d/httpd.conf
```

4. **(Optional)** Execute the following commands to update the file, /var/opt/tms/output/httpd.conf with value of common port, port_value, that will be used to open UI and RGE nodes.

```
> en
# shell
```

```
# sed -i "s/:443/:<port_value>/g"
/var/opt/tms/output/httpd.conf
```

For Example, if the port_value = **21443**,

```
sed -i "s/:443/:21443/g" /var/opt/tms/output/httpd.conf
```

5. **(Optional)** Log into a UI node (master or standby), and execute the following commands to generate the certificate:

Note: Enter the password as admin123 when prompted for password while executing the following commands.

```
> en
# _shell
# mkdir /data/certificate/
# cd /data/certificate/
# /usr/java/jre1.7.0_45/bin/keytool -genkey -keystore
httpdkeystore -alias httpd -storepass admin123 -keypass
admin123 -dname "CN=mural-scrum, OU=MURAL, O=GUA, L=SM,
S= CA, C=US" -validity 365
# /usr/java/latest/bin/keytool -importkeystore -
srckeystore httpdkeystore -destkeystore intermediate.p12
-deststoretype PKCS12
# openssl pkcs12 -in intermediate.p12 -out mural_cert.pem
-nodes
```

6. **(Optional)** Execute the following commands to create directory, /data/certificate/, and copy the file generated in the preceding step, mural_cert.pem, on all the other UI and RGE nodes.

```
> en
# _shell
# scp /data/certificate/mural_cert.pem admin@<other
UI/RGE node ip>:/data/certificate
```

7. **(Optional)** Run the following commands on all the RGE nodes:

```
> en
# conf t
(config)# rubix modify-app rge set adv-attribute rubixUrl
value <RGE Node URL>
```

For example,

```
rubix modify-app rge set adv-attribute rubixUrl value
https://ucsd-report.cisco.com:40443
```

8. **(Optional)** Execute the commands to update the cli httpd property on all the UI and RGE nodes:

```
> en
# conf t
(config)# web https port <port_value>
```

For example, if the the port_value is **21443**,

```
(config) # web https port 21443
```

9. Execute the following commands to restart httpd process on all the UI and RGE nodes:

```
> en
# conf t
(config)# pm process httpd restart
//To verify that https is running.
(config)# show pm process httpd
```

10. Execute the following commands to update new files on all UI and RGE nodes.

```
> en
# _shell
# cp /opt/tms/rubix-atlas4.2/config/config_proxy.json
/opt/tms/rubix-atlas4.2/config/config.json
# cp /opt/tms/rubix-atlas4.2/config/app-switcher-config_
```

```

proxy.json /opt/tms/rubix-atlas4.2/config/app-switcher-
config.json
# cp /opt/tms/rubix-bulkstats-atlas4.2/config/config_
proxy.json /opt/tms/rubix-bulkstats-
atlas4.2/config/config.json
# cp /opt/tms/rubix-bulkstats-atlas4.2/config/app-
switcher-config_proxy.json /opt/tms/rubix-bulkstats-
atlas4.2/config/app-switcher-config.json
# cp /opt/tms/rubix-bulkstats-atlas4.2/index_proxy.html
/opt/tms/rubix-bulkstats-atlas4.2/index.html

```

11. Execute the following commands to update the web.xml file on all the UI and RGE nodes.

```

# sed -i
"s#>application/json<#>application/javascript<#g"
/opt/tms/apache-tomcat/apache-tomcat-7.0.41/conf/web.xml

```

12. **(Optional)** Execute the following commands to add the hostname mapping on all the UI and RGE node:

```

> en
# conf t
(config)# ip host <RGE node URL> <RGE node vip>
(config)# ip host <UI node URL> <UI node vip>

```

For example, vip of UI cluster is 192.168.193.245 and vip of rge cluster is 192.168.193.246.

```

(config)# ip host ucsd.cisco.com 192.168.193.245
(config)# ip host ucsd-report.cisco.com 192.168.193.246

```

13. **(Optional)** Execute the following commands to enable ip filter setting that allows the new port on all the UI and RGE nodes:

```

> en
# conf t
(config)# ip filter chain INPUT rule append tail target

```

```
ACCEPT source-addr <ip of the server> /32 dest-port
<port_value> protocol tcp
(config)# configuration write
```

For example,

```
(config)# ip filter chain INPUT rule append tail target
ACCEPT source-addr 192.168.117.67 /32 dest-port 21443
protocol tcp
```

- To update the *About Us* information, run the following commands on all the UI and RGE nodes:

```
> en
# _shell
# mount -o remount,rw /
# for i in rubix-atlas4.2 rubix-bulkstats-atlas4.2 ; do cd
/opt/tms/$i/config ; cp aboutUs.json aboutUs.json_bkp;
pwd ; sed -i 's/4.1.rc1/4.2.rc1/g' aboutUs.json ; sed -i
's/7th August, 2017/10th December, 2017/g' aboutUs.json; done
```

The output may resemble as follows:

```
/opt/tms/rubix-atlas4.2/config
/opt/tms/rubix-bulkstats-atlas4.2/config
```

- Log into corresponding UI nodes, and execute the following commands:

```
> en
# _shell
# /bin/rm -f /data/instances/*/1/bin/blazed*
# cli -m config
(config)# no pm process rubix launch auto
(config)# pm process rubix restart
(config)# rubix modify-app <app-name> modify-instance 1
enable
(config)# rubix modify-app <app-name> enable
```

```
(config)# write memory
```

Where, <app-name> is as follows. Start the applications in the same order as mentioned below.

For all the UI nodes:

- atlas

For both the RGE nodes:

- bulkstats
- reportAtlas
- rge
- ruleEngine

7. Perform the following steps to update the `/etc/hosts` mapping:

- Map the UI node to domain. For example, <VIP of UI> domain.
- Map the RGE and other rubix application nodes to domain-report. For example, <VIP of RGE cluster> domain-report.

8. Hit the following URLs through your browser:

- `https://domain-name:21443/`, if reverse proxy configurations are applied.
For example, `https://demo.cisco.com:21443/`.
- `https://domain-name:6443/`, if reverse proxy configuration not applied.
For example, `https://demo.cisco.com:6443/`.

Use the domain name which was provided to you at the time of initial configuration for the UI nodes.

Note: For browsers, except Chrome, you need to perform an additional step for the UI to run properly. In the browser address bar, hit the following URLs to accept certificates in order to access different UI modules:

- <https://demo-report.cisco.com:20443>
- <https://demo-report.cisco.com:30443>
- <https://demo-report.cisco.com:50443>

9. Perform the following steps to access SMI user interface:

- Log into MURAL with administrator credentials.
- Click optional modules,  on the top-right corner.
- Click **System Monitoring Interface** icon, the SMI user interface opens in a new tab.
- Log into the application by entering username as **mural** and password as **mural123**.
- Log into master NameNode and run all jobs:

```
> en
# _shell
# pmx subshell oozie run job all
```

4.9 Enable the Collector in the Live Mode

- Execute the following commands on all collector nodes to enable the collector again in the live mode:

```
> en
# _shell
# cli -m config
(config)# collector modify-instance 1 modify-adaptor
bulkStatsauto-bin-slide enable
(config)# collector modify-instance 1 modify-adaptor edrflow
auto-bin-slide enable
(config)# collector modify-instance 1 modify-adaptor edrhttp
auto-bin-slide enable
(config) # pm process collector launch auto
(config)# write memory
```

```
(config)# pm process collector restart
(config)# quit
#
```

- Run the following command to verify collector process status after few minutes:

```
# cli -t en 'show pm process collector' | grep status
```

The output may resemble as follows:

```
Current status:  running
#
```

Note: It should be in running mode.

