MURAL Software Installation Guide

Version 4.1

Published: 2017-09-29

Americas Headquarters

Cisco Systems, Inc.

170 West Tasman Drive

San Jose, CA 95134-1706 USA

http://www.cisco.com

Tel: 408 526-4000

800 553-NETS (6387)

Fax: 408 527-0883

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 Installation Guide

Table of Contents

1. Installation Overview

This document describes how to install the MURAL application. MURAL provides Web-based reporting and analytics abilities for deep packet inspection (DPI) data emerging from your network.

1.1 Before you begin

Before installing MURAL, you should possess a working knowledge of the following:

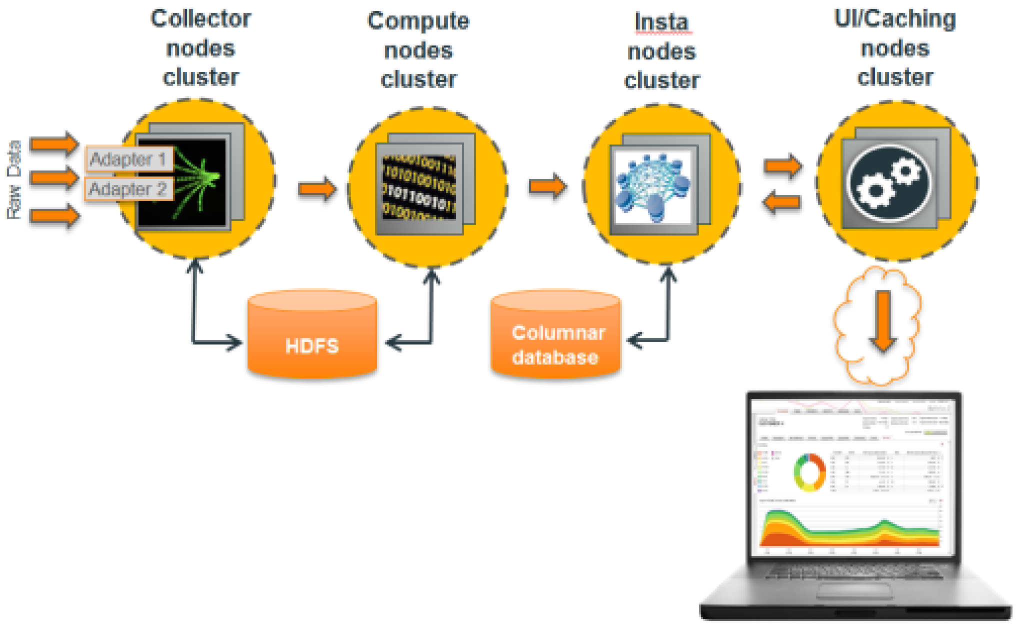
- Linux
- Cisco UCS

Prior to installing the application, we recommend that you:

- Review the *MURAL 4.1 Release Notes*.
- Complete a training course on MURAL.
- Ensure that installation package components are available. For more information see "Installation Package " on page 10.
- Have an understanding of Cisco UCS hardware administration.
- Ensure that MURAL system hardware installation has been completed successfully as specified in the bill of materials (BOM) and the setup is ready to install the system.
- The number of each type of node is customized for your deployment.
- Update the CIQ sheet completely, excluding ‘storageInfo’ section with MURAL setup details as required. For more information, see "Understanding the CIQ Sheet" on page 106.

1.2 System Components

The following figure illustrates the components of the MURAL platform, focusing on how the data flows through the system:



1.3 Standard Setup

The MURAL platform consists of the following nodes, each hosted on blades in the UCS Chassis:

1. General Management System (GMS)—Provides centralized management of the Reflex platform nodes, such as remote manufacturing of blades (installing the MURAL operating system), patch management, monitoring of all nodes and operations, and importing and running node configurations. The GMS node cluster supports high availability.
2. Collector—Collects data streams pushed to the Reflex platform, interprets the exported flows, enriches them with static data, and assembles data sets. The Collector stores the raw data in the Hadoop file system (HDFS) and sends it to the Compute node. The Collector node cluster can have any number of servers, in pairs for master and standby and uses 1+1 redundancy (transparent failover between pairs of active-active nodes).
3. UI/Caching (Rubix)—Hosts the Rubix engine and data cache. The Rubix engine queries the Insta nodes constantly and when new data is available, it fetches it to store in the data cache, so that it can respond more quickly

to requests from the UI engine. The Caching node is sometimes called the Rubix node. The Caching node uses N+1 redundancy in active-active mode.

4. Compute node—Analyzes and aggregates the data, creating data cubes. The Compute node cluster can have any number of servers, depending on your deployment, and uses N+1 redundancy.

5. Insta node—Stores and manages the processed data in a columnar database. It also manages the Insta database, which stores processed data cubes. The Insta node cluster has two servers with 1+1 redundancy.

Apart from standard installations, MURAL system can be prepared with fewer blades by accommodating two or three application components into a single cluster, as described in the following sections.

1.3.1 Starter Pack Setup

In the Starter Pack setup, GMS, Collector (with Name-node) and UI components are hosted together on the same cluster.

1.3.2 Medium Pack Setup

In the Medium Pack setup, GMS and Collector (with Name node) components are hosted together on the same cluster.

1.3.3 Hardware

The MURAL application is hosted on the UCS Blade Server Chassis that comprises GMS, Collector, Rubix, Compute and Insta nodes. Data storage is hosted on the EMC storage devices.

The data flows that feed the MURAL system are pushed by an ASR.

1.4 Installation Package

The MURAL software installation package contains the following components:

- An ISO image: mfgcd-atlas<**release version**>.iso, where **release version** changes with each release, for example 4.1.rc1. **mfgcd-atlas4.1.rc1.iso** is used as an example of ISO file.

Note:For the exact image name and the MD5 checksum for the software image, refer to the release notes for your release.

- The CIQ sheet file, which is used to create hardware and final configuration file (XML). This CIQ sheet is used to gather basic configuration information like IP addresses, cluster details, and storage details that will be used to create the configuration file (XML) for installation and configuration of MURAL system. For details, see "Customer Information Questionnaire (CIQ) Sheet" below
- Software patches that are available for the release. Refer to the release notes for your release for more information about patches.
- Management information bases (MIBs).

1.5 Customer Information Questionnaire (CIQ) Sheet

The Customer Information Questionnaire (CIQ) sheet is a spreadsheet containing hardware inventory information in a prescribed manner that needs to be collected before starting the installation process.

The CIQ sheet file contains the following worksheets:

- MiscInfo — Specifies configuration file name, site, hardware type, login message, and NTP settings.
- NetworkInfo — Specifies networks, default gateway, DNS, NTP, and SNMP servers. It also specifies UCS Manager access details, which is used to collect the MAC addresses from blades. MAC address can be left blank as the script that runs on CIQ can populate them automatically. If the script gives a warning that it is not able to get the MAC address, then you need to manually provide MAC addresses for all the nodes and run the script again.
- ClusterInfo — Specifies cluster names and type as well as interfaces and VIPs (if any).
- NodeInfo — Specifies chassis and slot IDs, hostnames, Mac IDs and IP addresses, KVM IPs of blades and Cluster Names for all the blades to group into specific roles.

- StorageInfo — Specifies node hostnames, WWIDs and MURAL components like Collectors, Datanodes, and so on to assign LUNs from SAN storage devices.

Apart from the CIQ sheet, following information is also required from site during the installation:

- Connectivity— Specifies the details for ports and connections
- Firewall— Identifies the firewall changes required for connectivity
- Static Configuration— Specifies locations for various information bases (IBs) that are required by the application

For detailed information about CIQ sheet, refer "Understanding the CIQ Sheet" on page 106

2. Installation Phases

This topic provides information about the various phases involved in setting up the system. It involves installing the software and all the configurations to make the system live.

Prerequisites:

- Obtain the CIQ sheet and verify that the UCS hardware is correctly configured for the MURAL system. For more information, see "Understanding the CIQ Sheet" on page 106
- Obtain the sizing sheet from the customer.

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

2.1 Installing Operating System

This phase starts with uploading of the ISO file and manufacturing of all the nodes. Once the nodes are manufactured, the clusters are defined as per the CIQ sheet and the required software are installed. For details, see "Installing Operating System" on page 15

2.2 Configuring MURAL Nodes

Once the required software are installed on each node, the nodes are configured and roles are assigned for each node. The configuration are based on the site requirements. For details, see "Configuring Mural Nodes" on page 31

2.3 Configuring ASR and IBs

Once all the nodes are installed, the ASR feeds and IBs list must be configured to ensure that system receives the required data. Correct configuration also ensures that system is able to process the data as per the requirements. For more details, see "Configuring ASR and IBs" on page 43

2.4 Processing the Data

The successfully installed system is required to be configured with data processing parameters. These configurations are required to ensure that the system performs as per the requirements. For more details, see "Processing the Data " on page 54

3. Installing Operating System

This topic describes the process of uploading of the ISO image file to be used for creating the master GMS node and configuring the CIQ sheet to create the hardware xml file. After the master GMS node is manufactured, all the related patches must be installed on the master GMS node.

3.1 Manufacturing the Initial Node

Set up a node that is used to create configuration files (XML). This node is manufactured by mapping the ISO image file of MURAL software as virtual media from UCS blade console. Manufacture all other nodes by installing OS through PXE boot. If PXE boot does not work, manufacture other nodes manually.

Prerequisite

Ensure that Serial over LAN (SOL) is configured on all the blades during UCS setup.

To manufacture the initial node, perform the following steps:

1. Download the ISO image to the machine from which you will access the Cisco UCS blades.

2. Verify the MD5 of the downloaded file. Execute the following command:

   ```
   ISO Image: mfgcd-atlas4.1.rc1.iso
   # md5sum mfgcd-atlas4.1.rc1.iso
   ```

   The output must resemble as follows:

   ```
   b3753743b06299910dfe5f8295f5f961  mfgcd-atlas4.1.rc1.iso
   ```

   Verify the value of generated MD5 of ISO file with the actual MD5 of the ISO file.

   Note: For the exact name of the ISO file and the related MD5 value, refer to the Release Notes. The name of ISO image, **mfgcd-atlas4.1.rc1.iso** is used as an example in the document.

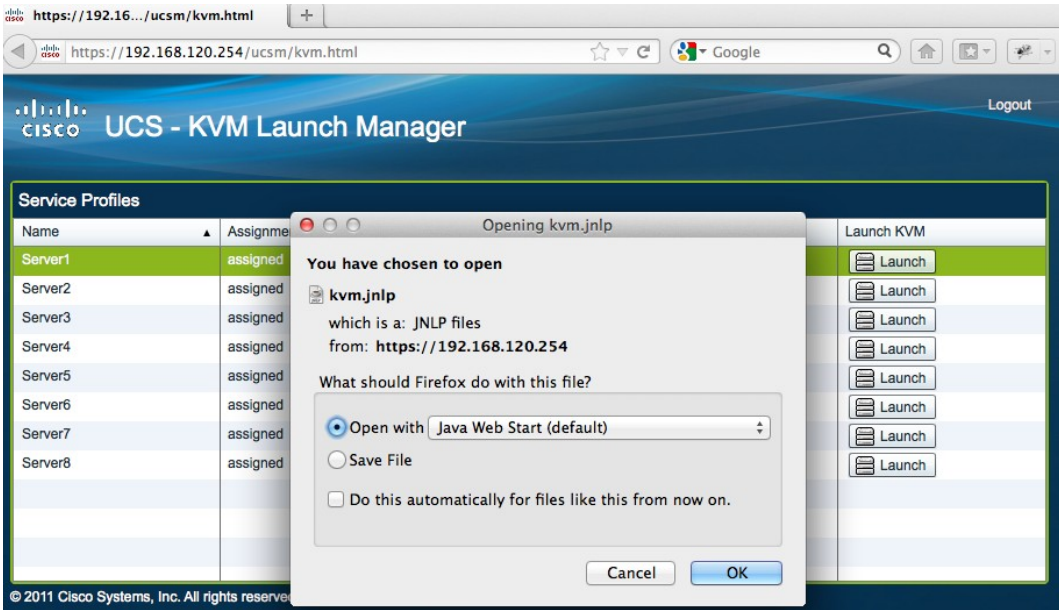
3. Open the Cisco KVM Login page of the GMS node in a Web browser.

4. Click the **Launch KVM Manager** button to open the KVM login page.

5. Log in to the KVM Manager using your credentials. All the blades available on the chassis are displayed.

6. Click the **Launch** button for the first node. Click **OK** to download and open the `kvm.jnlp` file.

   The following image illustrates the opening of the `kvm.jnlp` file.



https://192.16.../ucsm/kvm.html
https://192.168.120.254/ucsm/kvm.html
Google
UCS - KVM Launch Manager
Logout
Service Profiles
© 2011 Cisco Systems, Inc. All rights reserve

Name	Assignme	Launch KVM
Server1	assigned	Launch
Server2	assigned	Launch
Server3	assigned	Launch
Server4	assigned	Launch
Server5	assigned	Launch
Server6	assigned	Launch
Server7	assigned	Launch
Server8	assigned	Launch

Opening kvm.jnlp

You have chosen to open

kvm.jnlp

which is a: JNLP files

from: https://192.168.120.254

What should Firefox do with this file?

● Open with Java Web Start (default)

○ Save File

□ Do this automatically for files like this from now on.

Cancel OK

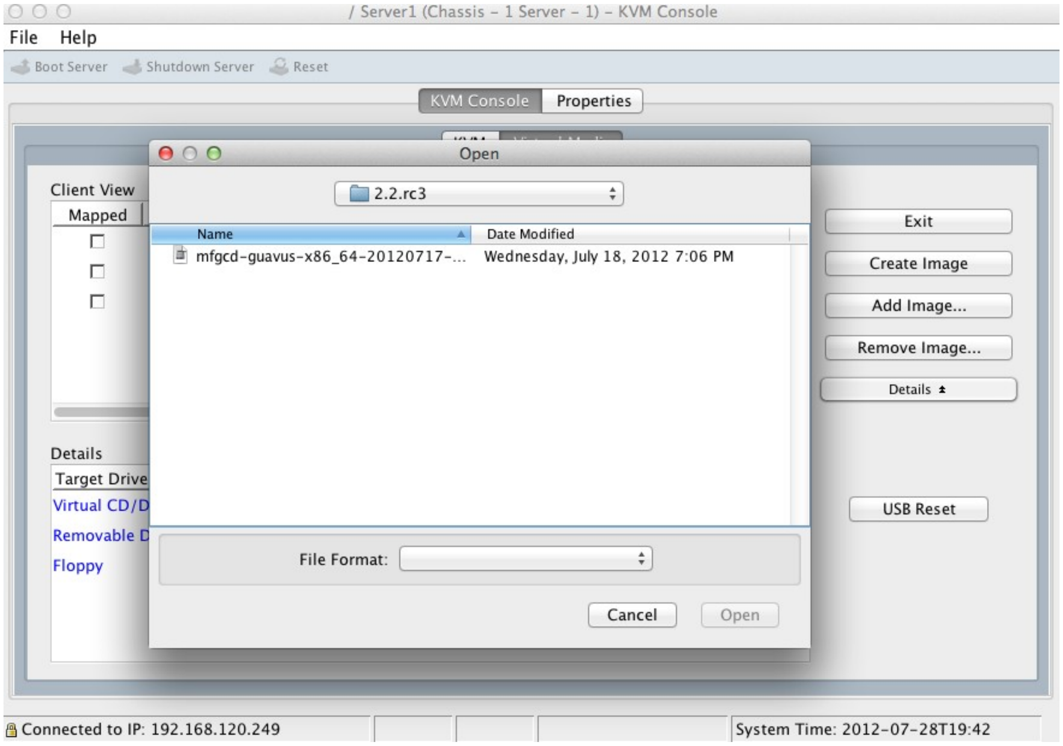
7. In the KVM Native Library warning box, click **OK**.

8. The console for the port is displayed. Click the **Virtual Media** tab.

9. Click **Activate Virtual Devices**.

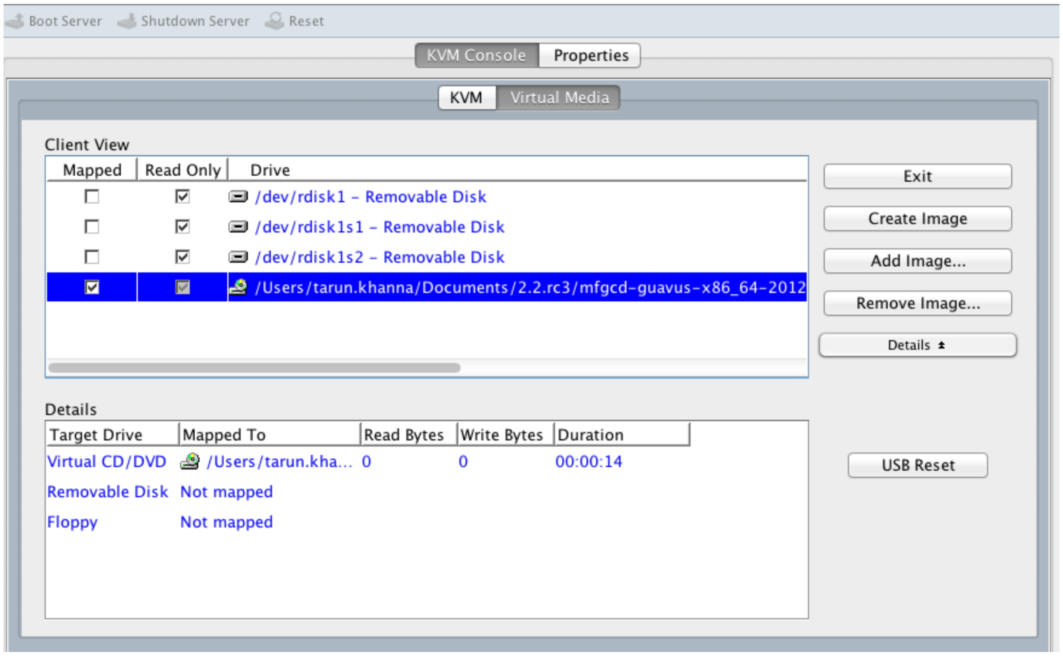
10. Click **Map CD/DVD** and specify the path of the ISO image that you downloaded in Step 1.

    The following image illustrates the image that needs to be added.



11. Check the check box in the **Mapped** column next to the ISO image to mount it.

    The following image illustrates the selection of the check box.



Client View

Mapped	Read Only	Drive
□	☑	/dev/rdisk1 – Removable Disk
□	☑	/dev/rdisk1s1 – Removable Disk
□	☑	/dev/rdisk1s2 – Removable Disk
☑	☑	/Users/tarun.khanna/Documents/2.2.rc3/mfgcd-guavus-x86_64-2012

Details

Target Drive	Mapped To	Read Bytes	Write Bytes	Duration
Virtual CD/DVD	/Users/tarun.kha...	0	0	00:00:14
Removable Disk	Not mapped			
Floppy	Not mapped			

12. Reboot the blade so that it can boot with the mounted image. Click the **KVM** tab and mouse over the **Ctrl-Alt-Del** from the **Macros** > **Static Macros** drop-down menu to select the option.

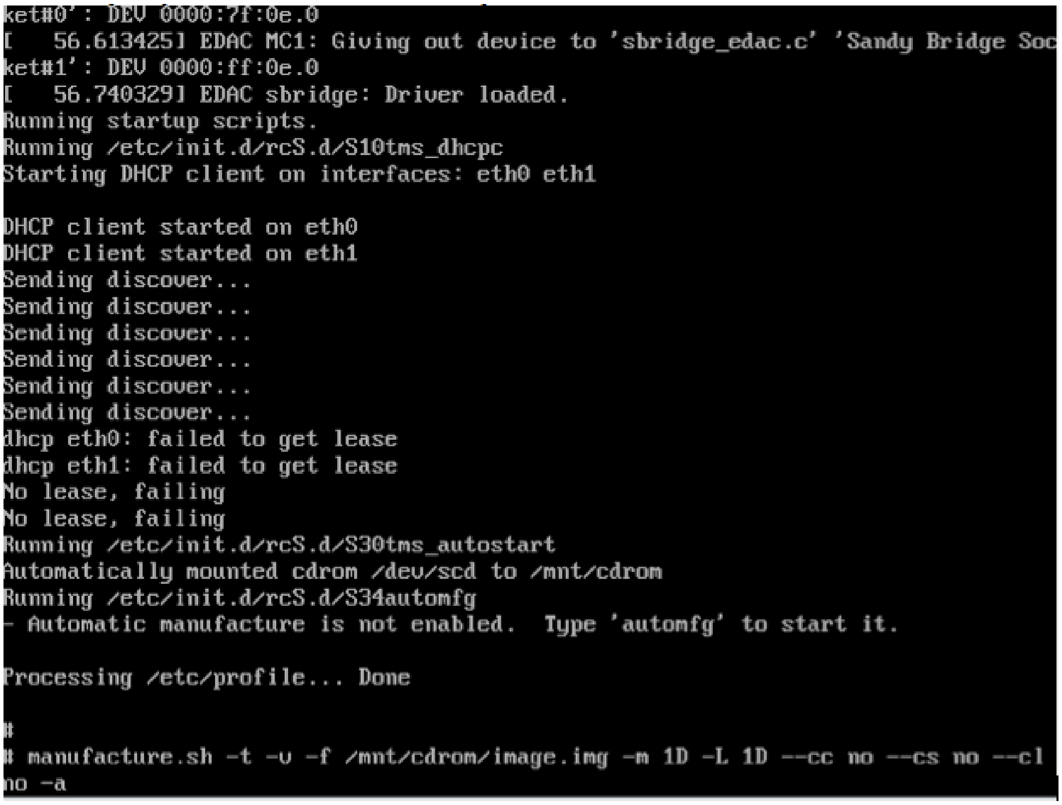
13. When the boot order screen appears, press **F6** to select the boot order.

    Note: Wait for 10 to 15 minutes before executing the following command.

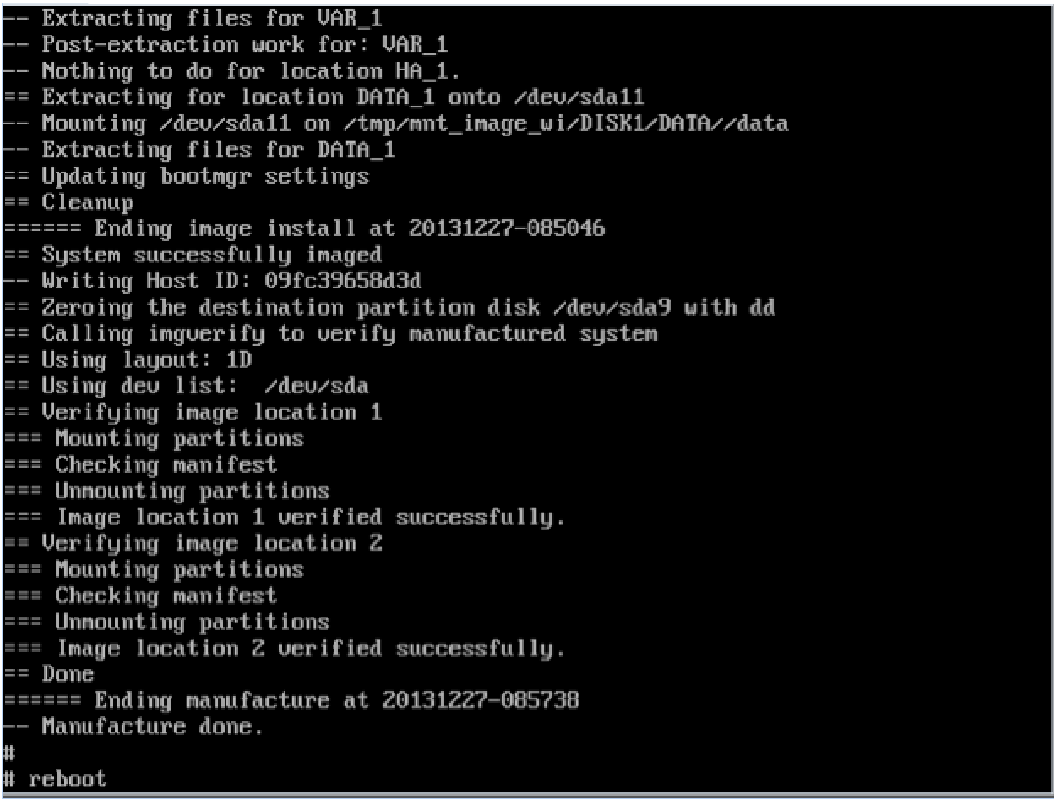
14. Execute the following command to start the manufacture process:

```
# manufacture.sh -v -t -f /mnt/cdrom/image.img -m 1D -L 1D
--cc no --cs no --cl no -a
```

The following screens illustrate manufacturing of the node.



```
ket#0': DEV 0000:7f:0e.0
[   56.613425] EDAC MC1: Giving out device to 'sbridge_edac.c' 'Sandy Bridge Soc
ket#1': DEV 0000:ff:0e.0
[   56.740329] EDAC sbridge: Driver loaded.
Running startup scripts.
Running /etc/init.d/rcS.d/S10tms_dhcpc
Starting DHCP client on interfaces: eth0 eth1

DHCP client started on eth0
DHCP client started on eth1
Sending discover...
Sending discover...
Sending discover...
Sending discover...
Sending discover...
Sending discover...
dhcp eth0: failed to get lease
dhcp eth1: failed to get lease
No lease, failing
No lease, failing
Running /etc/init.d/rcS.d/S30tms_autostart
Automatically mounted cdrom /dev/scd to /mnt/cdrom
Running /etc/init.d/rcS.d/S34automfg
- Automatic manufacture is not enabled.  Type 'automfg' to start it.

Processing /etc/profile... Done

#
# manufacture.sh -t -v -f /mnt/cdrom/image.img -m 1D -L 1D --cc no --cs no --cl
no -a
```



```
-- Extracting files for VAR_1
-- Post-extraction work for: VAR_1
-- Nothing to do for location HA_1.
== Extracting for location DATA_1 onto /dev/sda11
-- Mounting /dev/sda11 on /tmp/mnt_image_wi/DISK1/DATA//data
-- Extracting files for DATA_1
== Updating bootmgr settings
== Cleanup
====== Ending image install at 20131227-085046
== System successfully imaged
-- Writing Host ID: 09fc39658d3d
== Zeroing the destination partition disk /dev/sda9 with dd
== Calling imgverify to verify manufactured system
== Using layout: 1D
== Using dev list:  /dev/sda
== Verifying image location 1
=== Mounting partitions
=== Checking manifest
=== Unmounting partitions
=== Image location 1 verified successfully.
== Verifying image location 2
=== Mounting partitions
=== Checking manifest
=== Unmounting partitions
=== Image location 2 verified successfully.
== Done
====== Ending manufacture at 20131227-085738
-- Manufacture done.
#
# reboot
```

The system displays the "Manufacture done" message and returns to the # prompt once the manufacturing of a blade is completed.

15. Uncheck the ISO image selected in step 11. Type `reboot` to reboot the node with the new ISO image.

3.2 Configuring Node as Master GMS Node

To configure the initial node as the master GMS node, perform the following steps:

1. Log in to the initial node using the Console.
2. Execute the following commands to set the password:

```
> en
# conf t
(config) # username root password <rootpassword>
(config) # username admin password <adminpassword>
```

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

3. Assign the IP addresses for the management interface and default gateway:

```
> en
# conf t
(config) # interface <mgmt_interface> ip address <mgmt_IP_
of_GMS_ server><subnetmask_of_mgmt_network>
(config) # ip default-gateway <mgmt_network_default_
gateway_IP>
```

   For example:

```
(config) # interface eth0 ip address 192.168.103.78 /24
(config) # ip default-gateway 192.168.103.1
(config) # write memory
```

4. Download the patches from the FTP server to the GMS server in the `/data/tmp_patch` directory and apply all patches applicable for GMS nodes. For more information, refer to the *MURAL Release Notes* for a complete list of 4.1 release patches and installation instructions.

5. Update the CIQ sheet with MAC IDs. For details, see "Fetching MAC ID from UCS Manager" on page 77.

6. Copy the CIQ sheet into GMS node under the `/data` folder and run the `CIQ2XML` script (without WWID mode) to generate the Hardware XML file.

```
# mount -o remount,rw /
# cd /opt/etc/scripts/mmi/CIQ2GMSXML/src/
# python GMSXMLGenerator.py -f /data/Mural_CIQ_Template.xls
-u true -s 0

GMS HARDWARE XML saved as /data/configs/gms/Mural_
Hardware.xml
```

```
**************************************************
SUCCESSFULLY COMPLETED GMS HARDWARE XML GENERATION
**************************************************
```

Successful execution creates an XML file under `/data/configs/gms/` folder. In case of error, details are available in the log file stored at `/data/mmi/ciq2gmsxml.log`.

Note:

- The command executed in this step collects MAC IDs from all the blades and interfaces specified in the CIQ sheet. If UCS Manager is not enabled to fetch MAC IDs, update the CIQ sheet for MAC IDs and run the command above without the '`-u true`' option.
- If the command is unable to fetch the MAC IDs, automatically, refer to "Fetching MAC ID from UCS Manager" on page 77

6. Ensure that the hardware XML file has been created and available under `/data/configs/gms` folder.

```
# ls -l /data/configs/gms/Mural_Hardware.xml
-rw-r--r-- 1 admin root 27352 Feb 23 11:16
/data/configs/gms/Mural_Hardware.xml
#
```

3.3 Manufacturing Other Nodes using the GMS Node

To manufacture all the other nodes for MURAL, perform the following steps:

1. Copy (by SCP or FTP) ISO image file into master GMS node under `/data` directory. This configures GMS node as the PXE boot server that is used to boot other nodes using the ISO image of MURAL.
2. Execute the `prePxeGmsFlow.py` script to mount the ISO image for all the nodes. Provide the information as mentioned in the following table.

Parameter Name	Description	Example
Hardware xml	Name and full path of the hardware XML file, name is same as the value of XML File Name from the MiscInfo tab in CIQ sheet.	Mural_Hardware.xml
Management Network	Name of the management network, same as value of Network Name in the NetworkInfo tab in CIQ sheet. hardware xml name.xml is a dynamic parameter that changes as per the hardware xml name provided, for example Mural_Hardware_StandardPack.xml	MgmtNW
ISO file name	Filename with full path of ISO image used to manufacture nodes.	/data/mfgcd-atlas4.1.rc1.iso

The following sample may resemble the output.

```
> en
# _shell
# cd /opt/etc/scripts/GMS_FlowAutomation
# python prePxeGmsFlow.py
#Logging started in log file : /data/mmi/install_gms_
20161006.log
Available XML's :
/data/configs/gms/Mural_Hardware_StandardPack.xml
Enter the Hardware xml generated [default = Mural_Hardware_
StandardPack.xml]:
Enter the Management Network name in Mural_Hardware_
StandardPack.xml Hardware xml : MgmtNW
Enter the ISO path : /data/mfgcd-atlas4.1.rc1.iso
Initializing the GMS Node Class ...
Cluster = rge-clu :
rge-1 => 192.168.188.53
```

```
rge-2 => 192.168.188.55
Cluster = ui-clu :
ui-1 => 192.168.188.57
ui-2 => 192.168.188.62
Cluster = col-clu :
col-1 => 192.168.188.58
col-2 => 192.168.188.59
Cluster = gms-clu :
gms-1 => 192.168.188.66
gms-2 => 192.168.188.56
Cluster = cmp-clu :
dn-1 => 192.168.188.60
dn-2 => 192.168.188.61
Cluster = ins-clu :
ins-2 => 192.168.188.54
ins-1 => 192.168.188.52

Stopping PM Process: pgsqld
Current state of pgsqld : stopped
Stopping PM Process: gms_server
Current state of gms_server : stopped
GMS Version Mode set: lite-version
Starting gms_server process
Launched GMS in lite-version
Waiting web services to come up
Activating GMS HW XML: /data/configs/gms/Mural_Hardware_
StandardPack.xml
Gms Hw Xml Activated : True
Fetching ISO: /data/mfgcd-atlas4.1.rc1.iso on GMS Node
ISO Fetch complete Done
Mounting ISO: mfgcd-atlas4.1.rc1.iso
Image mount successfully completed
```

```
Please manually execute the gms pxeBoot enable command(s) on
gms master

Execution Complete !!!

Reboot all other blades to boot via PXE to initiate
manufaturing
via cli on GMS node, execute below command till all blades are
manufactured:
gms show manufacturing all
```

3. Launch PXE boot using the KVM console for all the MURAL nodes, one by one to manufacture the nodes with ISO image:

```
# cli -m config
```

For all the nodes:

```
(config) # gms pxeBoot cluster all node all ucsManagerIP <IP
of UCS Manager> loginName <User Name> loginPassword <User
Password>
```

For example:

```
(config) # gms pxeBoot cluster all node all ucsManagerIP
192.168.125.4 loginName Gurgaon\uttam.meena loginPassword *
```

For a single node:

```
(config) # gms pxeBoot cluster <cluster-name> node <node-name>
ucsManagerIP <IP of UCS Manager> loginName <User Name>
loginPassword <User Password>
```

For example:

```
(config) # gms pxeBoot cluster DN-CLUS node UCS-DN-1
ucsManagerIP 192.168.125.4 loginName Gurgaon\uttam.meena
loginPassword *
```

Note: User ID used for loginName must have admin privilege on UCS Manager for all the MURAL nodes.

The following sample may resemble the output:

```
Total number of nodes to be processed : 1
Total number of nodes processed :1 .  .  .  .  .  .  .  .  .
.
All nodes proccessed
Cluster : DN-CLUS      Node :  UCS-DN-1 Sucessfully Pxebooted
(config) #
```

This command triggers blade reboot from network. Once the blade starts booting from the network, GMS pushes the image on the blade using PXE boot. Manufacture process can be started on each blade in parallel.

A blade takes approximately 30 to 45 minutes to manufacture with the new image. Run the following command to check the blade manufacture status.

```
(config) # gms show manufacturing-status cluster <Cluster
Name>
```

The following example illustrates the PXE boot status for nodes under specific cluster:

```
(config) # gms show manufacturing-status cluster DN-CLUS
UCS-DN-1: OK : Product release: 4.1.rc1
UCS-DN-2: OK : Product release: 4.1.rc1
(config) #
```

The following example illustrates the PXE boot status for all the nodes:

```
(config) # gms show manufacturing-status all
```

Note: If the command based on PXE boot from GMS is not possible as mentioned in the procedure above, see "PXE Boot of Blades using the KVM console" on page 78 to launch blade PXE boot using KVM console.

3.4 Preparing the Final XML with WWIDs and Applications

Prerequisite

Once all blades are ready with MURAL image, ask hardware team to assign the required LUNs to respective nodes as specified in sizing sheet.

To collect storage information, perform the following:

1. Log into Master GMS node and run the following commands to reboot all the nodes after LUN allocation is done by hardware team. Provide the name of the Application XML and Management network name. If the name of the Application XML is same as the value displayed for default, press **Enter**.

```
> en
# _shell
# mount -o remount,rw /
# cd /opt/etc/scripts/GMS_FlowAutomation
# python reload_all_nodes.py
```

   The output may resemble to the following sample.

```
Logging started in log file : /data/mmi/install_gms_
20161006.log
Available XML's :
/data/configs/gms/Mural_Hardware_StandardPack.xml
Enter the Hardware xml generated [default = Mural_Hardware_
StandardPack.xml]:
Enter the Management Network name in Mural_Hardware_
StandardPack.xml Hardware xml : MgmtNW
GMS Version Mode set: lite-version
Cluster = rge-clu :
rge-1 => 192.168.188.53
rge-2 => 192.168.188.55Cluster = ui-clu :
ui-1 => 192.168.188.57
ui-2 => 192.168.188.62
Cluster = col-clu :
```

```
col-1 => 192.168.188.58
col-2 => 192.168.188.59
Cluster = gms-clu :
gms-1 => 192.168.188.66
gms-2 => 192.168.188.56
Cluster = cmp-clu :
dn-1 => 192.168.188.60
dn-2 => 192.168.188.61
Cluster = ins-clu :
ins-2 => 192.168.188.54
ins-1 => 192.168.188.52
Reloading nodes of Cluster : rge-clu
Restarting Node: 192.168.188.53
Restarting Node: 192.168.188.55
--------------------------------------------------------------
------------------
Reloading nodes of Cluster : ui-clu
Restarting Node: 192.168.188.57
Restarting Node: 192.168.188.62
--------------------------------------------------------------
------------------
Reloading nodes of Cluster : col-clu
Restarting Node: 192.168.188.58
Restarting Node: 192.168.188.59
--------------------------------------------------------------
------------------Reloading nodes of Cluster : gms-clu
ALERT : This node [self] will also be restarted at the end:
192.168.188.66
Restarting Node: 192.168.188.56
--------------------------------------------------------------
------------------
Reloading nodes of Cluster : cmp-clu
Restarting Node: 192.168.188.60
Restarting Node: 192.168.188.61
--------------------------------------------------------------
------------------
Reloading nodes of Cluster : ins-clu
Restarting Node: 192.168.188.54
Restarting Node: 192.168.188.52
--------------------------------------------------------------
------------------
SSH Public Key saved on node : 192.168.188.66
Restarting Node [self]: 192.168.188.66
Execution complete !!
#
```

See "Getting the WWIDs of LUNs" on page 80 to collect WWIDs and update CIQ sheet.

2. Copy (SCP or FTP) the updated CIQ sheet into GMS node under `/data` directory.
3. Log into the GMS node using ssh and run the following commands to prepare the final XML:

```
> en
# _shell
# mount -o remount,rw /
# cd /opt/etc/scripts/mmi/CIQ2GMSXML/src/
# python GMSXMLGenerator.py -f /data/Mural_CIQ_Template.xls -s
1
File Exists!! Do you want to overwrite file at
/data/configs/gms/Mural_Hardware.xml [Y/N]: Y
GMS HARDWARE XML saved as /data/configs/gms/Mural_Hardware.xml
**************************************************
SUCCESSFULLY COMPLETED GMS HARDWARE XML GENERATION
**************************************************
#
```

Successful execution creates an XML file under `/data/configs/gms/` folder. In case of error, details are available in the log file stored at `/data/mmi/ciq2gmsxml.log`.

```
# ls -l /data/configs/gms/Mural_Hardware.xml
-rw-r--r-- 1 admin root 32349 Feb 23 11:26
/data/configs/gms/Mural_Hardware.xml
```

3.5 Applying Patches on MURAL Nodes

Apply all the patches applicable for the respective MURAL nodes.

Note: Refer to the MURAL Release Notes for a complete list of patches and installation instructions.

4. Configuring Mural Nodes

After you have installed the MURAL software on GMS, perform the following tasks to configure the software:

Note: It is recommended to create screen and use the created screen for installation to ensure that the connection is not lost between the terminal and servers during installation process.

4.1 Creating GMS XML using Deployment Wizard

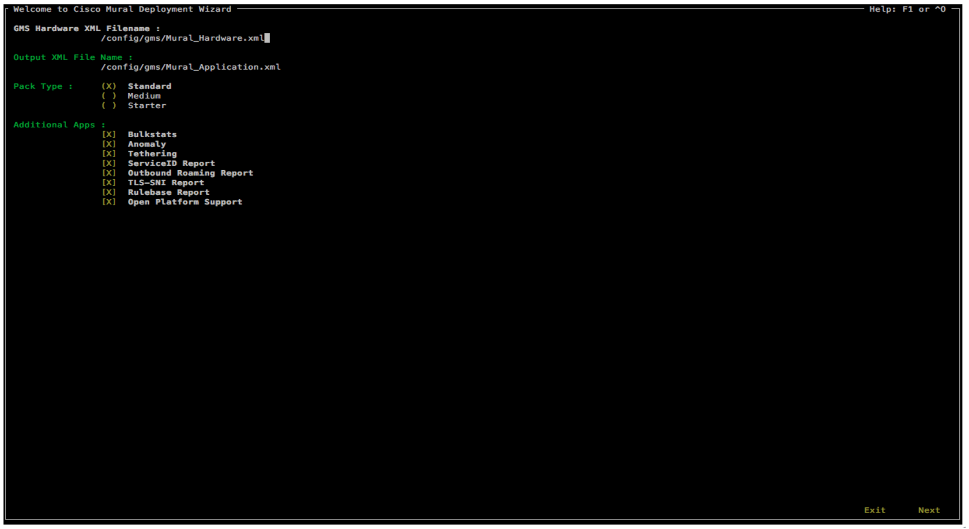
The GMS XML is created by using the deployment wizard. The XML is further used for configuring the nodes by scripts.

To create the GMS XML, perform the following steps:

1. From the command terminal, execute the following commands to launch MURAL Deployment wizard to prepare the final hardware XML.

   ```
   # sh /opt/etc/scripts/mmi/cisco_mural_deployment_wizard.sh
   ```

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

   ```
   Welcome to Cisco Mural Deployment Wizard                              Help: F1 or ^O

   GMS Hardware XML Filename :
                   /config/gms/Mural_Hardware.xml

   Output XML File Name :
                   /config/gms/Mural_Application.xml

   Pack Type :     (X)  Standard
                   ( )  Medium
                   ( )  Starter

   Additional Apps :
                   [X]  Bulkstats
                   [X]  Anomaly
                   [X]  Tethering
                   [X]  ServiceID Report
                   [X]  Outbound Roaming Report
                   [X]  TLS-SNI Report
                   [X]  Rulebase Report
                   [X]  Open Platform Support
   ```

   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.

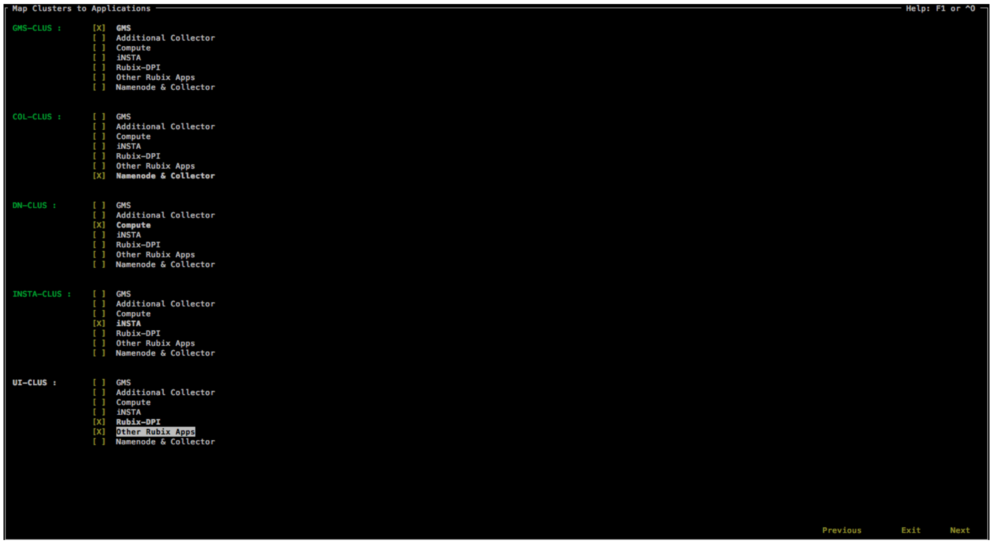
By default, MURAL installs Content Analytics (CA) application and following additional application can be selected if purchased. These additional applications have impact on resources required for deployment and should be enabled only if the deployment was sized for them:

- Bulkstats and KPI
- Anomaly
- Tethering

MURAL nodes are further classified and installed together in a particular cluster to host specific application component. The following table lists cluster configurations for various installation types and applications to be selected for these clusters.

Installation Type	**GMS Cluster**	**Collector Cluster**	**UI Cluster (RGE)**	**iNSTA Cluster**	**DN Cluster**
Standard Pack	GMS	Namenode & Collector	Rubix-DPI Other Rubix Apps	iNSTA	Compute
Medium Pack	GC (GMS + Namenode & Collector)		Rubix-DPI Other Rubix Apps	iNSTA	Compute
Starter Pack	GCU (GMS + Namenode & Collector + UI)			iNSTA	Compute

The following image illustrates the available cluster configurations.



```
Map Clusters to Applications

GMS-CLUS :        [X]  GMS
                  [ ]  Additional Collector
                  [ ]  Compute
                  [ ]  iNSTA
                  [ ]  Rubix-DPI
                  [ ]  Other Rubix Apps
                  [ ]  Namenode & Collector

COL-CLUS :        [ ]  GMS
                  [ ]  Additional Collector
                  [ ]  Compute
                  [ ]  iNSTA
                  [ ]  Rubix-DPI
                  [ ]  Other Rubix Apps
                  [X]  Namenode & Collector

DN-CLUS :         [ ]  GMS
                  [ ]  Additional Collector
                  [X]  Compute
                  [ ]  iNSTA
                  [ ]  Rubix-DPI
                  [ ]  Other Rubix Apps
                  [ ]  Namenode & Collector

INSTA-CLUS :      [ ]  GMS
                  [ ]  Additional Collector
                  [ ]  Compute
                  [X]  iNSTA
                  [ ]  Rubix-DPI
                  [ ]  Other Rubix Apps
                  [ ]  Namenode & Collector

UI-CLUS :         [ ]  GMS
                  [ ]  Additional Collector
                  [ ]  Compute
                  [ ]  iNSTA
                  [X]  Rubix-DPI
                  [X]  Other Rubix Apps
                  [ ]  Namenode & Collector
```

The following table lists the global variables for Application Configuration Parameters for site-specific requirements.

Property Name	Description	Value
adaptor.bulkStats.numThreads	Keep 2 threads for BS always and other two equal sets to flow and http, totaling 75% of collector cores.	2
adaptor.edrflow.numThreads:		8
adaptor.edrhttp.numThreads:		8

Property Name	**Description**	**Value**
adaptor.edrflow.profile: adaptor.edrhttp.profile:	For Anonymisation, replace `EdrFlowParquetReaderNoAnoConfig.xml` file with `EdrFlowParquetReaderConfig.xml` and `EdrHttpParquetReaderNoAnoConfig.` file With `EdrHttpParquetReaderConfig.xml`	No Anonymisation
application.atlas.cachePersistToDisk application.bulkstats.cachePersistToDisk	true - enable disk based caching for atlas and BS; false - disable	true true
application.atlas.rubixInstance.1.tomcatInstanceMaxSize application.atlas.rubixInstance.1.initialJavaHeapSize:	Maximum and initial java memory for atlas process (should be <= 50% of Rubix node RAM)	90g 90g

Property Name	Description	Value
application.ruleEngine.rubixInstance.1.tomcatInstanceMaxSize application.ruleEngine.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for ruleEngine process (keep it as default)	5g 5g
application.bulkstats.rubixInstance.1.tomcatInstanceMaxSize application.bulkstats.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for bulkstat process (should be <= 10% of Rubix node RAM)	20g 20g
application.reportAtlas.rubixInstance.1.tomcatInstanceMaxSize application.reportAtlas.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for offline report process (keep it as default)	10g 5g
application.rge.rubixInstance.1.tomcatInstanceMaxSize application.rge.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for report engine process (keep it as default)	10g 10g

Property Name	Description	Value
Timezone:	Sets the timezone for deployment	UTC
FQDN:	Sets URL to access the MURAL UI	ucsd.-cisco.com
rgemailSupport:	dummy-support@cisco.com	As Default
mailSender:	Sets the E-mail ID as sender of offline reports.	admin@-cisco.com
mailHost:	Sets SMTP Server host address	mx1.cisco.-com
mailPort:	Sets SMTP Server Port	25

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.

4.2 Assigning Role to MURAL Nodes

After all the nodes are manufactured and GMS XML is manufactured, the nodes are assigned roles to be performed.

1. Log into the GMS server using the management IP address and start the configuration

2. Execute the `postPxeGmsFlow.py` script on the GMS server:

```
>en
# _shell
# cd /opt/etc/scripts/GMS_FlowAutomation
# python postPxeGmsFlow.py
```

3. The script installs appliance on all the nodes.

Provide the name of the Application XML and Management network name. If the name of the Application XML is same as the value displayed for default, press Enter.

The following sample may resemble the output:

```
Logging started in log file : install_gms_20160711.log
Enter the Application Installation xml generated [default =
Mural_Application.xml]:
Enter the Management Network name in Mural_Application.xml
Application xml : MgmtNW
Initializing the GMS Node Class ...
Cluster = rge-clu :
rge-1 => 192.168.188.53
rge-2 => 192.168.188.55
Cluster = gc-clu :
gcn-2 => 192.168.188.59
gcn-1 => 192.168.188.58
Cluster = ins-clu :
ins-2 => 192.168.188.54
ins-1 => 192.168.188.52
Cluster = cmp-clu :
dn-1 => 192.168.188.60
dn-2 => 192.168.188.61
Cluster = ui-clu :
ui-1 => 192.168.188.57
ui-2 => 192.168.188.62

Stopping PM Process: pgsqld
Current state of pgsqld : stopped
Stopping PM Process: gms_server
Current state of gms_server : stopped
GMS Version Mode set: full-version
Starting pgsql process
Starting gms_server process
Launched GMS in full-version
Waiting web services to come up
Current process state Pgsql : running
Current process state GMS : running
Activating GMS Install XML: /data/configs/gms/Mural_
Application.xml
GMS XML Activation Done : True
Conjugate cluster node [standby 192.168.188.56]
Installation started on Cluster: [gms-clu] Node [gms-1]
Installation started on Cluster: [gms-clu] Node [gms-2]
Installation started on Cluster: [col-clu]
Installation started on Cluster: [cmp-clu]
Installation started on Cluster: [ins-clu]
Installation started on Cluster: [ui-clu]
Installation started on Cluster: [rge-clu]

Node installation commands executed.
**************************************************
dn-1 : Node successfully installed. DateTime: 2016-10-06
13:53:06.788538 Config-Version 1
dn-2 : Node successfully installed. DateTime: 2016-10-06
13:53:21.014069 Config-Version 1
col-1 : Node successfully installed. DateTime: 2016-10-06
13:16:11.987805 Config-Version 1
col-2 : Node successfully installed. DateTime: 2016-10-06
13:12:54.077793 Config-Version 1
gms-1 : Node successfully installed. DateTime: 2016-10-06
12:04:30.191348 Config-Version 1
gms-2 : Node successfully installed. DateTime: 2016-10-06
12:13:03.730950 Config-Version 1
ins-1 : Node successfully installed. DateTime: 2016-10-06
14:29:20.633255 Config-Version 1
```

```
ins-2 : Node successfully installed. DateTime: 2016-10-06
14:05:30.575501 Config-Version 1
ui-1 : Node successfully installed. DateTime: 2016-10-06
15:12:41.938352 Config-Version 1
ui-2 : Node successfully installed. DateTime: 2016-10-06
15:10:37.865661 Config-Version 1
rge-1 : Node successfully installed. DateTime: 2016-10-06
15:25:32.565343 Config-Version 1
rge-2 : Node successfully installed. DateTime: 2016-10-06
15:25:37.715186 Config-Version 1
**************************************************

Execution Complete !!!
```

From output messages of `postPxeGmsFlow.py` script, make sure that all the nodes have been successfully installed, before moving to next step.

4. Run the following command on pmx subshell of the master Name node:

```
# pmx subshell hadoop_yarn repair hdfs
```

5. Run the following command on the master Name node:

```
(config) # pm process tps restart
```

4.3 Configuring Site-Specific Applications

If anonymization feature is enabled, run the following commands on both the master and standby Name nodes:

```
# cli -m config
(config) # collector anonymization key <key> iv <iv_key> encode
(config) # write memory
(config) # quit
```

For example:

```
(config) # collector anonymization key
01234567890123456789012345678901
 iv 0123456789012345 encode
```

`<key>` and `<iv_key>` string shall be provided by customer during installation.

4.4 Verifying the Setup

After making any performance related modifications, verify that the processes are running.

Note: After you make any modifications, wait for at least 15 minutes to ensure processes have had a chance to restart before running the following commands.

4.4.1 Verifying Processes

Perform the following steps to verify that the required processes are running:

1. Log in to the master Name node and execute the following commands to verify that all Compute nodes have joined the HDFS cluster. The output shown in the following sample command sequence indicates correct configuration.

```
> en
# _shell
# ps -ef | grep java | grep Dproc | grep -v Dproc_dfs | awk '
{print $9}'
-Dproc_historyserver
-Dproc_journalnode
-Dproc_namenode
-Dproc_resourcemanager
-Dproc_namenode
-Dproc_datanode
-Dproc_secondarynamenode
#
# hdfs dfsadmin -report | egrep "available|Name|Status"
```

```
Datanodes available: 3 (3 total, 0 dead)  Name:
10.10.2.13:50010
Decommission Status : Normal
Name: 10.10.2.14:50010
Decommission Status : Normal
Name: 10.10.2.17:50010
Decommission Status : Normal
```

Note: The IP addresses mentioned in the sample code above are internal IP addresses.

2. Run the following commands on the standby Name node:

```
> en
(config) # _shell
# ps -ef | grep java | grep Dproc | grep -v Dproc_dfs | awk '
{print $9}'
-Dproc_namenode
-Dproc_namenode
-Dproc_datanode
-Dproc_secondarynamenode
-Dproc_journalnode
```

3. Log into the master Insta node, and execute the following commands:

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

If the process is running, the command output is `Current status: running`.

Execute the following command to verify the status of the Insta node.

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

The output must show all modules in ACTIVE state. It must also list all the instances and Adaptor status as RUNNING.

4. Log into the standby Insta node and repeat the same commands shown in step 3.

5. Execute the following commands to check status of Postgres process on both the Master and Standby nodes.

   For the Standard Pack setup, check the status of Postgress on GMS, Collector and iNSTA nodes.

   For the Medium Pack setup, check the status of Postgress on GMS and iNSTA nodes.

   ```
   > en
   # _shell
   # ps -ef | grep postmaster | grep -v grep
   2278     1  0 Feb13 ?        00:22:34
   /usr/pgsql-9.3/bin/postmaster -p 5432 -D /data/pgsql/9.3/data
   ```

4.4.2 Verifying Installation with Dummy Data

You can verify the installation and identify configuration issues by running the custom utility. For more information, see "Verifying Installation with Dummy Data" on page 82.

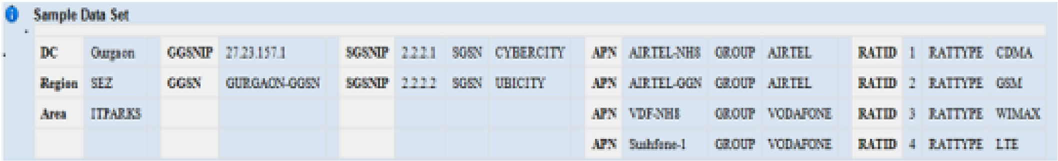
5. Configuring ASR and IBs

This topic provides information about creating and configuring the ASR and IBs. It also provides information about verification of the collector processes.

To configure your system for the data specific for your environment, you must update the information bases (IBs) and fetch them. In some cases, you might need to also manually modify the IBs to add or reclassify entries.

5.1 Updating IBs

The following table shows a sample data set for setting up the IBs.



Sample Data Set

DC	Gurgaon	GGSNIP	27.23.157.1	SGSNIP	2.2.2.1	SGSN	CYBERCITY	APN	AIRTEL-NH8	GROUP	AIRTEL	RATID	1	RATTYPE	CDMA
Region	SEZ	GGSN	GURGAON-GGSN	SGSNIP	2.2.2.2	SGSN	UBICITY	APN	AIRTEL-GGN	GROUP	AIRTEL	RATID	2	RATTYPE	GSM
Area	ITPARKS							APN	VDF-NH8	GROUP	VODAFONE	RATID	3	RATTYPE	WIMAX
								APN	Sushfone-1	GROUP	VODAFONE	RATID	4	RATTYPE	LTE

Note: Refer to the preceding table as a sample. Use the data that matches your environment. For example, for GGSN, you may want to use GGSN, PGW, or HA. In this case, GGSNIP is the management IP address. For SGSN, you may want to use SGSN, SGW, HSGW, or PDSN. In this case, SSGNIP is the service IP address.

To configure the IBs :

1. Log into the master collector node and execute the following commands:

```
[admin@collector-1 ~]# pmx
Welcome to pmx configuration environment.
pm extension> subshell aggregation_center
pm extension (aggregation center)> update all ibs from image
```

2. Enter the GGSN IPs and GGSN names:

```
pm extension (aggregation center)> edit ib ipGgsn.map add
GGSN IP: 27.23.157.1
GGSN: GGSN1
pm extension (aggregation center)> show ib ipGgsn.map
```

```
1[27.23.157.1][GGSN1]
pm extension (aggregation center)>
```

3. Enter the SGSN IPs and SGSN name:

```
pm extension (aggregation center)> edit ib ipSgsn.map add
SGSN IP: 2.2.2.1
SGSN: SGSN1
pm extension (aggregation center)> show ib ipSgsn.map
1[2.2.2.1][SGSN1]
pm extension (aggregation center)>
```

4. Enter the APN name and corresponding APN group:

```
pm extension (aggregation center)> edit ib apnGroup.map add
APN: Sushfone-1
APN Group: Sushfone-1
pm extension (aggregation center)> show ib apnGroup.map
1[Sushfone-1][Sushfone-1]
2[Sushfone-2][Sushfone-2]
3[Sushfone-3][Sushfone-3]
pm extension (aggregation center)> quit
```

For performing the cell sector annotation see Modifying Behaviour for Cell Sector Annotation.

Execute the following commands to configure the IBs for BulkStats:

```
pm extension> subshell bulkstats
pm extension (bulk stats)> update all ibs from image
pm extension (bulk stats)> quit
```

5.2 Adding ASR Gateways

Add all new DC or gateways to the configuration for the system. DC, ASR, and Gateway are synonymous terms and can be used interchangeably. Gateway/DC name is a unique key for a gateway configuration. In the BulkStat hierarchy, the gateway name and DC name are the same.

The following section lists guidelines for adding new gateways:

- All input directories are created under the /data/collector path. Hence, in the example below, the ASR should send EDR files to `data/collector/California/edr1` and send the Bulkstats file to `/data/collector/California/bs1`.

  Important: Confirm the actual input directories with the Cisco Technical Support team for your ASR platform.

- Log into master Name-node and create a CSV file (under /data dir) for adding the gateways into system as per below prescribed format and example:

  ```
  gateway,dc,region,location,schema-version,timezone,edr-filename-
  pattern,bulkstat-filename-pattern,edr-collector-filename-
  pattern,bulkstat-collector-filename-pattern,collector-ip,edr-
  file-path,bulkstat-file-path
  ```

  Note: This header must be the first line in the CSV file and the gateway details must be added after these.

  Example of EDR with Bulkstat.

  ```
  DC1,DC1,REGION1,AREA1,19,UTC,*_MURAL-edr-
  *_%MM%DD%YYYY%hh%mm%ss*,*_%YYYY%MM%DD%hh%mm%ss,*_*_
  *_%MM%DD%YYYY%hh%mm%ss*.gz,*_%YYYY%MM%DD%hh%mm%ss,192.168.192.130
  |192.168.192.131,/DC1/edr,/DC1/bs
  ```

  Example of EDR with without Bulkstat.

  ```
  DC1,DC1,REGION1,AREA1,,UTC,*_MURAL-edr-
  *_%MM%DD%YYYY%hh%mm%ss*,,*_*_
  *_%MM%DD%YYYY%hh%mm%ss*.gz,,192.168.192.130|192.168.192.131,/DC1/
  edr,
  ```

  Note: Use internal physical IPs for collector IPs and separate them using vertical bar i.e. “|” for multiple entries.

  For adding gateway without Bulkstat feed, keep the fields related to Bulkstat empty.

For EDR filename patterns, see "File Pattern Regular Expression" on page 96

To configure gateways:

1. Add gateway information for each gateway based on the guidelines provided in the preceding table:

```
> en
# conf t
(config)# pmx subshell aggregation_center
pm extension (aggregation center)> Welcome to pmx configuration
environment.
pm extension (aggregation center)> add bulk gateways
Please enter bulk file path: /data/gw.csv
```

2. Verify that the new gateway has been added successfully:

```
pm extension (aggregation center)> show gateways
```

3. Verify the IP addresses for all Collectors:

```
pm extension (aggregation center)> show collector IPs
 192.168.103.110, 192.168.103.113
pm extension (aggregation center)>
```

4. Verify that the BulkStats timezone is UTC in gateway.conf for every gateway. The reason for this is that the ASR internally changes the time zone to GMT for the BulkStats file. Edit gateway.conf for every BulkStats source at the path:

```
/data/configs/gateway/gateway.conf "timezone": "UTC"
```

5. Push the gateway configuration to all the Collectors:

```
pm extension (aggregation center)> push gateway configuration
```

6. Generate and push all IBs:

```
pm extension (aggregation center)> generate urlcat ib from image
pm extension (aggregation center)> show urlcat version
pm extension (aggregation center)> generate all ibs

pm extension (aggregation center)> push all ibs
pm extension (aggregation center)> quit
```

For Bulkstats:

```
pm extension> subshell bulkstats
pm extension (bulk stats)> generate all ibs
pm extension (bulk stats)> push all ibs
```

Run the following commands if Anomaly Detection is enabled:

```
[admin@collector-1 ~]# pmx
Welcome to pmx configuration environment.
pm extension> subshell anomaly
pm extension (anomaly)> update all ibs
pm extension (anomaly)> quit
```

5.3 Modifying Behaviour for Cell Sector Annotation

You can categorize the Cell Sectors, which are not present in the IB configuration into Other cells, and 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
   ```

2. 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
        "groupGGSNAsMisc":"false"
        "groupSGSNAsMisc":"false"
        "emitUsingMCCMNC":"false"
```

3. Execute the following command to set the value of groupGGSNAsMisc as true:

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

4. Execute the following command to set the value of groupSGSNAsMisc as true:

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

5. Execute the following command to set the value of emitUsingMCCMNC as true:

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

6. 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
        "groupGGSNAsMisc":"true"
        "groupSGSNAsMisc":"true"
        "emitUsingMCCMNC":"true"
```

5.3.1 Adding the Cell Sectors, GGSN and SGSN in IBs

If you want to retain the values of Cell Sectors, GGSN and SGSN, add their values in cellLocationInfo.list, ipGGSN.map and ipSGSN.map respectively.

To add the values, perform the perform the following steps:

1. To add the values of Cell Sectors, GGSN and SGSN to be retained in cellLocationInfo.list, ipGGSN.map and ipSGSN.map respectively, execute the following commands:

```
> en
# conf t
(config) # pmx subshell aggregation_center
pm extension (aggregation center)> edit ib
cellLocationInfo.list add
MCC,MNC: 101,01
pm extension (aggregation center)> edit ib ipGgsn.map add
GGSN IP: 27.23.157.1
GGSN: GGSN1
pm extension (aggregation center)> edit ib ipSgsn.map add
SGSN IP: 2.2.2.1
SGSN: SGSN1
```

   NOTE: If the IB is empty, all the cell sectors will be emitted as "Other Cells".

2. To update the IBs, execute the following commands:

```
> en
# conf t
(config) # pmx subshell aggregation_center
pm extension (aggregation center)> generate all ibs
pm extension (aggregation center)> push all ibs
```

5.4 Enabling CSV Report

This section explains how to enable reports selected under **Additional Apps** on the deployment wizard in "Updating IBs " on page 43.

- 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 **Outbound Roaming** report:

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

   ```
   >en
   # _shell
   # sed -i "s/:SP:/SP:/g"
   /opt/etc/oozie/Reports/config.json

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

- 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 .

5.5 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
```

5.6 Verifying Collector Process

To verify the collector process, run the following command on both the master and standby Collector:

```
# cli -t "en" "conf t" "show pm process collector" | grep status
Current status:  running
```

6. Processing the Data

This section includes information for setting up an ASR user in the Collector nodes and sending the EDR and BulkStats data feeds to the MURAL platform, setting the data start time, and running the data processing commands.

6.1 Verifying Incoming Data

This section includes information for setting up an ASR user in the Collector nodes and sending the EDR and BulkStats data feeds to the MURAL platform, setting the data start time, and running the data processing commands.

Use one of the ASR data feed methods to send data to the MURAL platform.

6.1.1 Creating a New User for the ASR in the Collectors

Perform the following step to set up a new user for the ASR in the Collectors.

1. Log into the master Collector node and execute the following commands create the user:

   ```
   > en
   # conf t
   (config)# username <username> password <password>
   (config)# write memory
   (config)# _shell
   ```

   Note: The username and password must be same as configured on the ASR for EDR and BulkStats file transfers.

6.1.2 Configuring the SSH Session Limit

Perform the following steps to configure the SSH session limits:

1. 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
```

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

6.1.3 Verifying the Input Data Feed

Start sending the EDR and BulkStats data feeds to the MURAL platform. If the ASR is used as an input node, the start time from the filename is created in the incoming EDR flow directory.

The file name has the timestamp, which can be used for job scheduling in the following process.

Notes:

- MURAL processes data that is received after the deployment is completed. MURAL is not configured to process the historical data.
- MURAL recommends ASR files to be of 10MB compressed size for optimum performance.

Validating Data on the Collector Nodes

1. Log into the master Namenode and go to `_shell`.

   ```
   > en
   # _shell
   ```

2. Run the indicated **hdfs** commands for the *mm* variable (minutes) in the timestamp), specify a multiple of 5 (**05**, **10**, **15**, and so on) up to **55**.

   Note: Specify the year, month day, hour, and minute for which data is being sent to the MURAL system.

   If the Collector node is receiving data in the expected format, it retains the data in HDFS. These directories and files are updated continuously as the data keeps coming in.

6.2 Configuring Oozie Jobs

This topic provides information about setting the start time of the oozie jobs and starting the oozie jobs after configuring the start time.

6.2.1 Setting the Start Time of Oozie jobs

To set the data start time in the configuration, perform the following steps:

1. Log into the master node.

```
>en
# _shell
# mount -o remount,rw /
# cd /opt/deployment/Mural_setStartTime/
# ./setOozieTime --dataStartTime data-start-time --node  --password
admin-password
```

2. Execute the **setOozieTime** script to set the time at which and data starts coming into the directories listed in Configuring Oozie Jobs.

   For example, if and data starts coming into the system from April 1, 2015, 06:00 onwards, run the following scripts with the `start_time` value as "2015-04-01T06:00Z":

```
# ./setOozieTime --dataStartTime  2015-04-01T06:00Z --node
192.168.147.11 --password admin@123 --verbose
```

   Note:

   - Enter minutes as a multiple of 5. For example, "2013-04-01T06:00Z."
   - If the starttime of the setoozie script is set as the last day of the month, update the job timing of **AnomalyHiveMonthly** manually after executing the script. The new job starttime to be updated should be last hour of the next month **+/- offset** of based on timezone. For example, if data starttime for the system is 31st march 5:00 and timezone is europe/budapest, then set the job **AnomalyHiveMonthly** starttime as 30 april 22:00.

     Execute the following commands to update the start time:

```
>en
# _shell
# pmx subshell oozie
pm extension (oozie)> set job AnomalyHiveMonthly
attribute jobStart 2016-04-30T22:00
```

3. Execute the following commands to set cleanupOffset for dataset generic_repost_output_hourly.

```
>en
# conf t
# pm extension (oozie)> pmx subshell oozie set dataset
generic_report_output_hourly attribute cleanupOffset 3
```

4. Execute the following commands to change jobstart time of NNModelD to ensure that it runs on 00, 06, 12, 18 hours of the day.

   For example, for data coming into the system from April 1, 2017,

   03:00 or 04:00 onwards, run the following scripts with the `jobStart` value as "2017-04-01T06:00z" as shown in the example below.

```
>en
# _shell
# pmx subshell oozie
# pm extension (oozie)> set job NNModelD attribute jobStart
2017-04-01T06:00z
```

5. Execute the Set Job Time Script for both the master and standby Namenodes.

   Note: This script may take upto minutes to complete for one node. Therefore, wait until it completes and returns to the prompt.

6.2.2 Starting the Oozie Jobs

Log into the master Name node and run the data processing commands from the Oozie subshell:

```
> en
# conf t
(config)# pmx
Welcome to pmx configuration environment.
pm extension> subshell oozie
pm extension (oozie)> run job all
```

The command output shows all the jobs that were initiated and if the jobs started successfully or not.

Note: It may take approximately 20 minutes to start all the jobs, depending upon what all applications are enabled.

6.3 Validating Data on Nodes

This section includes instructions for validating data after completing the system installations.

6.3.1 Validating Data on the Compute Blades (Data Nodes)

This section includes the steps required to validate data on the Compute blades (Data nodes) following the installation process.

Caution: Wait for two hours after completing the steps in "Validating Data on Nodes" above. This allows sufficient time for the jobs that process the collector data to start, and the `done.txt` files to be updated. If you do not wait, the validation may fail.

6.3.2 Validating EDR Data

1. Log into the master node and go to the `_shell`.

   ```
   > en
   # _shell
   ```

2. Check the last timestamp for the Core job.

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

3. Check the last timestamp for the EDR data cubes being generated by the

job.

```
# hdfs dfs -text /data/EDR/done.txt
```

4. Check the last timestamp for CubeExporter data cubes.

```
# hdfs dfs -text /data/CubeExporter/done.txt
```

5. Check the last timestamp for generated and exported Bulkstats data cubes.

```
# hdfs dfs -text /data/BulkStat/done.txt
# hdfs dfs -text /data/BSAgg15min/done.txt
# hdfs dfs -text /data/BulkStatExporter_15min/done.txt
```

6.3.3 Validating Generic Report Job Hourly

Caution: Before executing this procedure, wait for at least 30 minutes after verifying the successful running of CoreJob.

If the Outbound Roaming Report feature is enabled, execute the following command to verify the last timestamp of the Generic Report Job Hourly job after the first instance of CoreJob completes:

```
> en
# _shell
# hdfs dfs -text /data/GenericReportJobHourly/done.txt
```

The output may resemble to the following sample:

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

The following reports are generated:

- OutboundRoamerContent_0728201600.csv (from FLOW)

  "#DC, #RAT, #Country, #Network, #Device, #TT App, #TT App Category, #Down Bytes, #Up Bytes, #Flow Count, #Flow Duration, #Peak Flow Duration, #Subscriber Count"

- OutboundRoamerSp_0728201600.csv (from FLOW and HTTP)

  "#DC, #RAT, #Country, #Network, #SP, #Is HTTPS, #Down Bytes, #Up Bytes, #Hit Count, #Hit Duration, #Peak Hit Duration, #Subscriber Count"

6.3.4 Validating Insta Data

1. Log into the master Insta node and check the name of the database configured for DPI:

```
> en
# _shell
# cli -t "en" "conf t" "show runn full" |
grep "insta instance 0 cubes-database" | awk -F ' ' '{print $5}'
database_mural
```

2. Open the **idbmysql** user interface and specify **database_mural** as the database.

```
# idbmysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
...
mysql> use database_mural;
Database changed
```

3. Display the values in the **mints** and **maxts** columns for the 60-minute bin class and -1 aggregation level (shown in the first row of the following example).

```
mysql> select * from bin_metatable ;
+----------+---------------------+------------+------------+---------+-
------------+
| binclass | aggregationinterval | mints      | maxts      | binType |
maxexportts |
+----------+---------------------+------------+------------+---------+-
------------+
| 60min    |                  -1 | 1483228800 | 1483527600 | NULL    |
1483527600 |
| 60min    |               86400 |          0 |          0 | NULL    |
         0 |
| 60min    |              604800 |          0 |          0 | NULL    |
         0 |
| 60min    |             2419200 |          0 |          0 | NULL    |
         0 |
+----------+---------------------+------------+------------+---------+-
```

```
------------+
4 rows in set (0.02 sec)
```

4. Run the **date** command to convert the values from the **mints** and **maxts** columns to human-readable format.

   The following example indicates that data was processed between 00:00 on January 1 and 11:00 on January 4.

   ```
   date -d@1483228800
   Sun Jan  1 00:00:00 UTC 2017
   date -d@1483527600
   Wed Jan  4 11:00:00 UTC 2017
   ```

6.3.5 Validating Bulk Stats Data on the Insta Blade

1. Use SSH to log into the master Insta node and check the name of the database configured for EDR:

   ```
   > en
   # _shell
   # cli -t "en" "conf t" "show runn full" |
   grep "insta instance 1 cubes-database" | awk -F ' ' '{print $5}'
   bulkstats
   ```

2. Open the **idbmysql** user interface and select **bulkStats** as the database.

   ```
   # idbmysql
   Welcome to the MySQL monitor.  Commands end with ; or \g.
   ...
   mysql> use bulkStats;
   Database changed
   ```

3. Display the values in the **mints** and **maxts** columns for the 900 aggregation interval (shown in the second row in the example).

   ```
   mysql> select * from bin_metatable;
   +----------+---------------------+------------+------------+---------+
   | binclass | aggregationinterval | mints      | maxts      | binType |
   +----------+---------------------+------------+------------+---------+
   | 5min     |                  -1 |          0 |          0 | NULL    |
   ```

```
| 5min     |                900 | 1364713200 | 1367293500 | NULL    |
| 5min     |               3600 | 1364713200 | 1365004800 | NULL    |
| 5min     |              86400 | 1364688000 | 1364860800 | NULL    |
| 5min     |             604800 |          0 |          0 | NULL    |
| 5min     |            2419200 |          0 |          0 | NULL    |
+----------+--------------------+------------+------------+---------+
6 rows in set (12.18 sec)
mysql> quit
```

4. Convert the date format. Run the **date** command with the value of **maxts** (captured from the step above) for the row which shows **aggregationinterval** as 900.

   The following example indicates that data was processed between 09:45 on July 30 and 08:45 on August 29.

```
# date -d @1406713500
Wed Jul 30 09:45:00 UTC 2014
# date -d @1409301900
Fri Aug 29 08:45:00 UTC 2014
```

6.3.6 Configuring 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.
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.

5. Execute the following commands:

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

6.4 Starting UI Processes and Verifying Data

Start the UI processes and verify UI data. Ensure that the URL is set up in the DNS for the production system.

Log into UI nodes and execute the following steps to enable reverse proxy on your setup.
Note: This section is optional, if you do not require to use reverse proxy in your setup, you can skip these steps.

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-atlas4.1/config/config.json
   - /opt/tms/rubix-atlas4.1/config/app-switcher-config.json
   - /opt/tms/rubix-bulkstats-atlas4.1/config/config.json
   - /opt/tms/rubix-bulkstats-atlas4.1/config/app-switcher-config.json
   - /opt/tms/rubix-bulkstats-atlas4.1/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.**

```
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. 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` = **5443**,

```
sed -i "s/:443/:5443/g" /var/opt/tms/output/httpd.conf
```

5. 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
   ```

   This creates a directory, `/data/certificate/`.

6. Execute the following commands to 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. 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. 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 **5443**,

```
(config) # web https port 5443
```

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-atlas4.1/config/config_proxy.json
/opt/tms/rubix-atlas4.1/config/config.json
# cp /opt/tms/rubix-atlas4.1/config/app-switcher-config_
proxy.json /opt/tms/rubix-atlas4.1/config/app-switcher-
config.json
# cp /opt/tms/rubix-bulkstats-atlas4.1/config/config_
proxy.json /opt/tms/rubix-bulkstats-
atlas4.1/config/config.json
# cp /opt/tms/rubix-bulkstats-atlas4.1/config/app-switcher-
config_proxy.json /opt/tms/rubix-bulkstats-
atlas4.1/config/app-switcher-config.json
# cp /opt/tms/rubix-bulkstats-atlas4.1/index_proxy.html
/opt/tms/rubix-bulkstats-atlas4.1/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. 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. 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 5443 protocol
tcp
```

Starting the Rubix Tomcat Instance on Both Nodes

Note: You should only start UI Tomcat instances after at least two hours of data has been pushed into the Insta node.

1. Log into the master UI node.

   ```
   > en
   # conf t
   ```

   If RGE and DPI UI are running on same node, execute the following command:

   ```
   (config)# rubix modify-app rge set adv-attribute
   rubixReportQueueMax value 1
   (config)# write memory
   ```

2. Run the following commands to start the EDR process.

   ```
   (config)# pm process rubix restart
   (config)# rubix modify-app atlas enable
   (config)# rubix modify-app atlas modify-instance 1 enable
   (config)# write memory
   ```

   Check the tomcat process status using command:

   ```
   (config)# rubix status application atlas
   ```

Note: Ensure the running status of above service as `Current Status: running` before starting the next process.

3. Run the following commands to start the other processes:

   ```
   > en
   # conf t
   (config)# pm process rubix restart
   (config)# rubix modify-app ApplicationName enable
   (config)# rubix modify-app ApplicationName modify-instance 1 enable
   (config)# write memory
   ```

   Where *ApplicationName* is replaced by the following applications in the same order:

   - bulkstats
   - reportAtlas
   - rge
   - ruleEngine (if Anomaly or BulkStat is enabled)

   Check the tomcat process status using command:

   ```
   (config)# rubix status ApplicationName atlas
   ```

   Note: Ensure the running status of above service as `Current Status: running` before starting the next process. Check the corresponding rubix.log file to ensure that Atlas application is started properly without any exception.

4. Log into the standby node and repeat Steps 2 and 3.

5. Access the UIs by going to the URL **https://*domain-name*:6443/** through your browser.

   The domain name to be used is the one which was provided at the time of initial configuration through GMS for the UI nodes configuration details. For example:

```
URL: https://demo.cisco.com:6443/
Username: admin
Password: admin123
```

Note: Since the common certificate installation procedure is not finalized, you must click the same URL for the RGE ports once before opening the actual URL.

Visit the following ports once and accept the certificates:

```
https://domainName-report:20443/
https://domainName-report:30443/
```

For example:

```
https://demo-report.cisco.com:20443/
https://demo-report.cisco.com:30443/
```

6. If Reverse Proxy feature is enabled, visit the following port and accept the certificates:

```
https://domain-name:5443/
```

Use the domain name, which was provided at the time of initial configuration through GMS.

For example:

```
URL: https://demo.cisco.com:5443/
Username: admin
Password: admin123
```

Note: Once the installation is completed, ensure to back up the configurations. Refer to the *Operations and Troubleshooting Guide* for more information.

6.5 Updating Whitelists

After running MURAL system for two-three hours, run the following command from master Namenode to generate updated whitelists:

```
# pmx subshell aggregation_center
pm extension (aggregation center)> generate whitelist
```

```
Triggered Whitelist Generation; Check status via 'show whitelist
status' command
pm extension (aggregation center)> quit
pm extension> quit
#
```

Observe the categorization in UI after two hours to see the effects of whitelists update.

6.6 Setting Up Offline Reports

This topic provides information related to the configuration required for offline reports.

6.6.1 Uncategorized URL, UA, and TAC Reports

Create a file named `serverFile_uncatReports` on the master Namenode containing the destination information, to which the uncategorized URL, UA, and TAC reports would be copied.

1. The `serverFile_uncatReports` file contains the entries for the data transfer destination location. This file has the following format:

   ```
   IP, username, password, location-to-copy-reports
   ```

   For example,

   ```
   192.168.156.96, admin, admin@123, /data/offline_uncat_reports
   ```

   Note: The delimiter in this file must be ", " (comma followed by a space).

2. Log into the master Namenode and navigate to the `/data/work` subdirectory:

   ```
   > en
   # _shell
   # cd /data
   # cd work
   ```

3. Create the `serverFile_uncatReports` file:

```
# vi /data/work/serverFile_uncatReports
192.168.156.96, admin, admin@123, /data/offline_uncat_reports
```

Note: For the SCP protocol, the destination path should be the destination server. The destination path is not required for SFTP.

4. Create the same file on the standby Namenode.

6.6.2 Tethering Reports

Create two files called `serverFile_tethering` and `serverFile_tethering_subscribers_report` with details of the ASR gateways, where the TAC, OS or UA databases, and subscriber reports created as a result of tethering processing, need to be pushed.

1. The `serverFile_tethering` file contains the entries for the data transfer destination location. This file has the following format:

   ```
   Gateway-IP, gateway-username, gateway-password, location-to-copy-reports
   ```

   Where:

   - *Gateway-IP* is the ASR5K gateway IP address
   - *gateway-username* is the username for logging into ASR5K Gateway
   - *gateway-password* is the corresponding password to the username
   - *location-to-copy-reports* is the location on the ASR5K Gateway machine where databases need to be copied

2. Log in to the master Namenode:

   ```
   > en
   # _shell
   ```

3. Go to the data directory and create a sub-directory named work:

   ```
   # cd /data
   # mkdir work
   ```

4. Go to the work subdirectory and create the files:

```
# cd work
# vi /data/work/serverFile_tethering
192.168.156.96, admin, admin@123, /data/tethering_ibs
# vi  /data/work/serverFile_tethering_subscribers_report
192.168.156.96, admin, admin@123, /data/tethering_subs
```

Note: The delimiter in this file must be ", " (comma followed by a space).

5. Create the same file on the standby Namenode as well.

   For the SCP protocol, the destination path should be present at the destination server. This is not required for SFTP.

   This file can have multiple rows of this kind.

A. Fetching MAC ID from UCS Manager

This topic provides information on getting the MAC ID from the UCS manager.

Before you execute the script, get UCS IP, and admin user password details.

1. Upload the **adaptors.sh** script available at `/opt/etc/scripts/GMS_FlowAutomation` on the master GMS node and execute the script from Master GMS Node.

   ```
   # cd /opt/etc/scripts/GMS_FlowAutomation
   # sh ./adaptors.sh <UCS IP> <adminPassword>
   ```

   For example:

   ```
   # cd /opt/etc/scripts/GMS_FlowAutomation
   # sh ./adaptors.sh 172.30.4.9 *
   1/1
   eth0 00:A0:D7:42:08:7E
   eth1 00:A0:D7:42:08:6E
   1/2
   eth0 00:A0:D7:42:08:9D
   eth1 00:A0:D7:42:08:8D
   1/3
   eth0 00:A0:D7:42:08:5D
   eth1 00:A0:D7:42:08:4D
   ```

   Note: The script lists MAC IDs of all NICs for each blade that is installed into chassis/slot.

2. Update these MAC IDs into NodeInfo tab of the CIQ sheet for their respective blades, listed against chassis/slots.

B. PXE Boot of Blades using the KVM console

Before you begin, ensure to complete the following:

- Configure Serial over LAN (SOL) on all the blades during EMC setup.
- Locate your CIQ, and refer to it for details such as UCS access credentials.

To reboot the blades:

1. Open the Cisco UCS - KVM Launch Manager in a browser and enter your credentials.

   Note: Firefox with Java version 6, or later, is the most suitable browser to access the UCS.

   All the blades available on the chassis are displayed.

2. Click the **Launch** button for the first node.
3. Click **OK** to download and open the `kvm.jnlp` file.
4. Click **OK** in the keyboard access warning message that is displayed.
5. Perform the following steps in the KVM Console that is displayed:

   a. Open the KVM Console of the blade.

   b. Press **CTRL-ALT-DEL** to reboot the blade. Alternatively, click **Reset** at the top of the KVM console.

   c. After the prompt, press **F12** immediately to boot from the network.

      Once the blades start booting from the network, GMS pushes the image on all the blades using PXE boot for the manufacture process to start on each blade in parallel.

      A blade takes approximately 30 minutes to manufacture with the new image. Wait until the blade for which PXE boot was issued has been manufactured. A login prompt is displayed once the image has been manufactured on a blade.

Go back to "Installing Operating System" on page 15 to continue installing the MURAL system.

C. Getting the WWIDs of LUNs

This topic describes the process of getting the WWIDs of the LUNs assigned to various nodes.

Execute the `listWWIDs.py` script to fetch WWIDs from all the blades, using GMS hardware XML and Name of Management Network specified in the NetworkInfo tab of the CIQ sheet

Execute the **listWWIDs.py** script on the master GMS Node to generate the list of WWIDs.

```
# python listWWIDs.py -f <GMS XML File> -n <Network Name of
Management IPs>
```

For example:

```
# python listWWIDs.py -f /config/gms/active-
configuration/active.xml -n InternalNW
192.168.188.53, rge-1, [('36006016085003600f2264d58c633e611',
'250G')]
192.168.188.55, rge-2, [('36006016085003600ce7bc22ac633e611',
'250G')]
192.168.188.57, ui-1, [('36006016085003600e41beec8c833e611',
'1.0T')]
192.168.188.62, ui-2, [('360060160850036002c59f07d94e2e311',
'1.0T')]
192.168.188.58, gcn-1, [('360060160850036008680eec7c733e611',
'250G'), ('36006016085003600a8f5513694e2e311', '1.0T')]
192.168.188.59, gcn-2, [('36006016085003600e408d47dc833e611',
'250G'), ('360060160850036006e3e2b4494e2e311', '1.0T')]
192.168.188.66, reflex-42087e,
[('36006016085003600ead331eb56f7e311', '250G')]
192.168.188.60, dn-1, [('3600601608500360036be5c5294e2e311',
'1.0T')]
192.168.188.61, dn-2, [('360060160850036004e8dd75e94e2e311',
'1.0T')]
```

```
192.168.188.52, ins-1, [('36006016085003600c480bc55c433e611',
'250G'), ('36006016085003600fe1700b9ec52e511', '500G'),
('3600601608500360006d16fa3ef52e511', '500G')]
192.168.188.54, ins-2, [('36006016085003600b0a00d76c433e611',
'250G'), ('36006016085003600fe1700b9ec52e511', '500G'),
('3600601608500360006d16fa3ef52e511', '500G')]
```

Update the WWIDs into **StorageInfo** tab of the CIQ Sheet against respective node. Refer sizing sheet for appropriate LUN allocation to each node.

D. Verifying Installation with Dummy Data

If real input feed is not available due to any reason, you can verify the installation and identify configuration issues by running the custom utility.

This custom utility performs the following functions:

- Data Generator
    - Generates EDR and Bulkstats feeds with the configurable parameters including input header and input file name.
    - Sets up configuration required for collector.
    - Sets up IB specific configuration including Bulkstats.
    - Creates new iNSTA database.
- Job Configurations
    - Stops all jobs and removes HDFS .done file.
    - Configures the job with parameters defined in the config.txt file.
    - Configures Rubix by enabling atlas application on Rubix servers.
    - Runs and monitors the jobs.
    - Checks disk cache size.
- Cleanup
    - Deletes the content from the `/data/ib/` folder on all the servers.
    - Deletes the content from the `/data/atlasData/` folder on Rubix servers.
    - Removes data from the `/data/collector/edrflow_backup`, `/data/collector/edrhttp_backup`, and `/data/collector/bulkstats_files_backup/test/` folders on all the namenodes.
    - Removes the HDFS-related contents from the `/IB/`, `/local/IB/`,`/data/ib/`, and `/data/output/` folders.

- Removes the HDFS-related content from the /data/EDR_EDRSubcrBytesAgg_lastSuccess, /data/EDR_SubscriberDeviceMPH_lastSuccess, /data/EDR_EDRSubcrDev_lastSuccess, and /data/EDR_EDRTopN_lastSuccess folders.
- Removes HDFS-related content from the /data/TEST/edrhttp/, and /data/TEST/edrflow/ folders.
- Removes disk cache (/data/diskstore/UI-EDR) from all the Rubix servers.
- Removes Bulkstats cache (/data/diskstore/bulkstats-<hostname>) from all the RGE nodes (applicable only if bulkstat is enabled).
- Reverts the system to its original configurations (settings made before executing the utility).

D.1 Prerequisite

The following table lists the parameters and their default values that are required to be filled in the config.txt file available at /opt/etc/scripts/DummyData.

Group	Parameter Name	Default Value	Description
COMMON_CONF	jobName	CoreJob, EDR, and CubeExporter	Jobs that are configured, initiated, and monitored by the utility.
	edrFlowPath	/data/TEST/edrflow/%Y/%M/%D/%H/%mi	Path where collector puts the dummy data output for edrflow.
	edrHttpPath	/data/TEST/edrhttp/%Y/%M/%D/%H/%mi	Path where collector puts the dummy data output for edrhttp.

Group	Parameter Name	Default Value	Description
	isBulkstatEnabled	true	Flag indicating if bulkstats need to be configured by the utility to initiate bulkstats pipeline. **Note**: The version of the Bulkstats file is 19.
	bulkstatInputPath	/data/TEST/bulkStats/%Y/%M/%D/%H/%mi	Path where collector puts the dummy data output for bulkstats.
	gmsVip	None	Virtual IP address of the GMS node for parsing the active XML.

Group	Parameter Name	Default Value	Description
	edrFilePattern	None Note: Ensure that this value is same as the value set for collector. To obtain the collector value, execute the command, `echo show running-config full | cli -m config | grep edr | grep filename-format | awk '{print $NF}' | sort -nu.`	edr file pattern in which input data is required.
GENERATOR_CONF	edrflow_header	<default header>	Input edrflow file header required for data generation. Ensure to update it based on business requirement.
	edrhttp_header	<default header>	Input edrhttp file header required for data generation. Ensure to update it based on business requirement.

D.2 Assumptions

Refer to the following assumptions before using this utility:

- The Bulkstats generator version 19 is used.
- The anomaly jobs are not configured.
- The setup is freshly manufactured with no data in collector input/output paths and empty disk cache.
- After successful execution of the full pipe, separate script is run to clean the data from the system for all modules as follows:
  a. Collector— input local path and output hdfs path
  b. Jobs output and intermediate path (for example, last successful path for EDR actions)
  c. iNSTA— database used for dummy execution
  d. Rubix— diskcache and /data/atlasData ibs
  e. IBs that are updated or generated as part of the process
- The database used in iNSTA is hardcoded as TEST_MURAL and TEST_BULKSTATS. The default database is hardcoded as DATABASE_MURAL and BULKSTATS.
- The time range and data sizes are hardcoded.
- The cleanup IB procedure reverts any changes done in the IB manually.
- The setOozieTime script is executed after the cleanup action.
- The config.txt is correctly filled. Refer to the table in the "Prerequisite" on page 83.
- If the tool is stopped in between due to any reason, you must run the dummy script with option cleanup first before proceeding further.
- The whole pipeline is executed in the live mode.

D.3 Run the Complete Pipe

1. Log into the Master Namenode.

2. Update the config.txt file available at /opt/etc/scripts/DummyData as described in the "Prerequisite" on page 83.

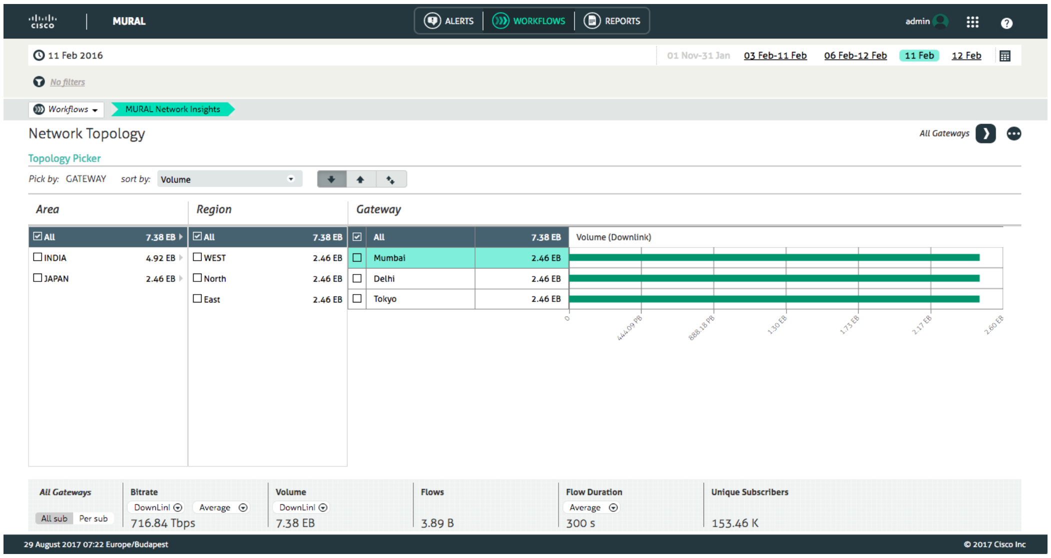
3. Execute the script as follows:

```
[admin@VM-MUR-GCN1 DummyData]# cd /opt/etc/scripts/DummyData
[admin@VM-MUR-GCN1 DummyData]# python DummyData.py
************************************
Welcome to the Dummy Data tool :
 This tool gives you the option of creating dummy data and run
multiple jobs on it so that UI can be seen with dummy data
 Once dummy data is populated, this tool also gives the option
of cleaning up the dummy data and revert the system to its
original state
 Input required : /opt/etc/scripts/DummyData/config.txt
 Logs Path: /opt/etc/scripts/DummyData/log.txt
 ************************************
Enter valid input from below options :
 1. Run complete pipe
 2. Cleanup the system
 3. Exit
1
Have you entered the proper values in config.txt file . Enter
yes or no
yes
Please check /opt/etc/scripts/DummyData/log.txt for logs



Welcome to Mural 4.1
Execution finished . Now check UI for data
```

Refer to the following images that illustrate data in the MURAL UI after running the complete pipeline.

Note: Actual data in the UI may vary by two percent.

Data on the MURAL Network Insights Workflow



11 Feb 2016

Network Topology

Pick by: GATEWAY sort by: Volume

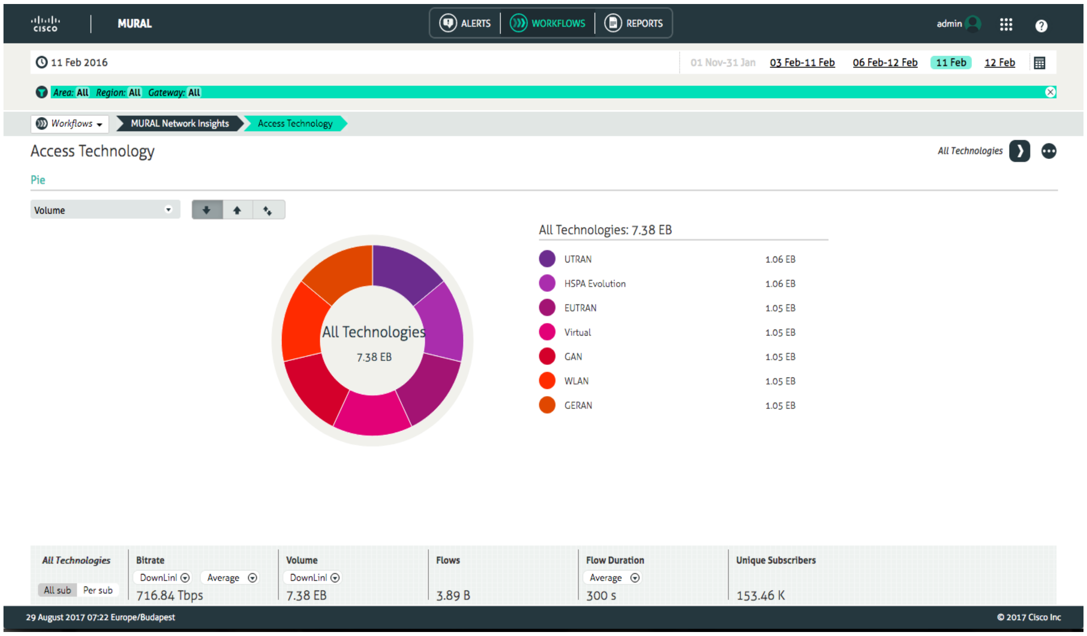
Area		Region		Gateway	
☑ All	7.38 EB	☑ All	7.38 EB	☑ All	7.38 EB
□ INDIA	4.92 EB	□ WEST	2.46 EB	□ Mumbai	2.46 EB
□ JAPAN	2.46 EB	□ North	2.46 EB	□ Delhi	2.46 EB
		□ East	2.46 EB	□ Tokyo	2.46 EB

All Gateways	Bitrate	Volume	Flows	Flow Duration	Unique Subscribers
	DownLinl Average	DownLinl		Average	
	716.84 Tbps	7.38 EB	3.89 B	300 s	153.46 K

29 August 2017 07:22 Europe/Budapest

© 2017 Cisco Inc

Data on the Access Technology Workflow



11 Feb 2016

Area: All Region: All Gateway: All

Access Technology

Volume

All Technologies: 7.38 EB

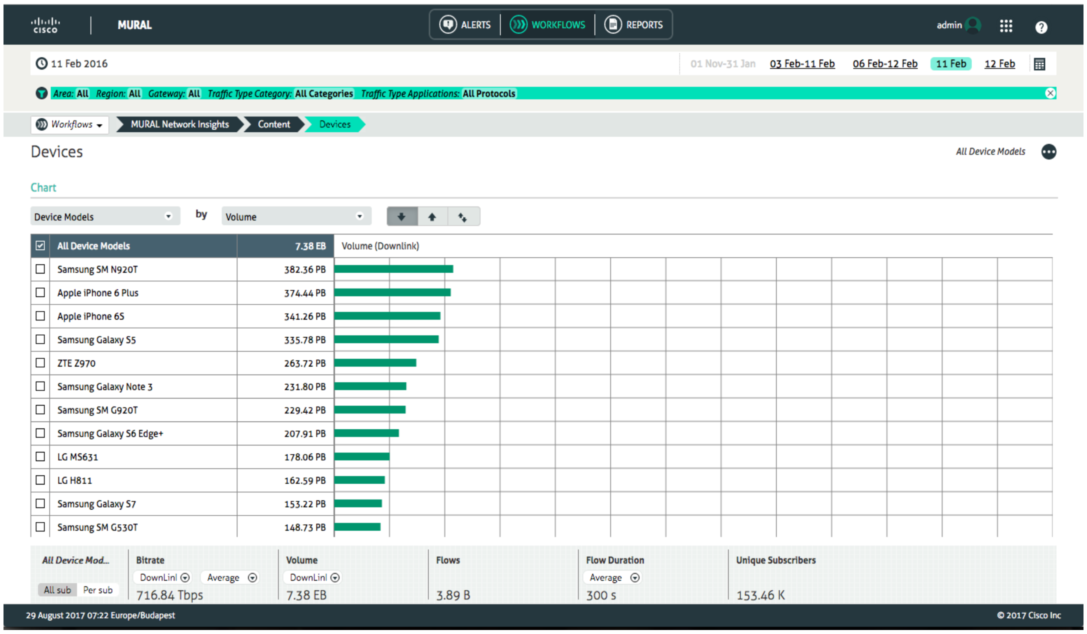
UTRAN	1.06 EB
HSPA Evolution	1.06 EB
EUTRAN	1.05 EB
Virtual	1.05 EB
GAN	1.05 EB
WLAN	1.05 EB
GERAN	1.05 EB

All Technologies	Bitrate	Volume	Flows	Flow Duration	Unique Subscribers
	DownLinl Average	DownLinl		Average	
	716.84 Tbps	7.38 EB	3.89 B	300 s	153.46 K

29 August 2017 07:22 Europe/Budapest

© 2017 Cisco Inc

Data on the Devices Workflow



11 Feb 2016

01 Nov-31 Jan | 03 Feb-11 Feb | 06 Feb-12 Feb | 11 Feb | 12 Feb

Area: All Region: All Gateway: All Traffic Type Category: All Categories Traffic Type Applications: All Protocols

Devices — All Device Models

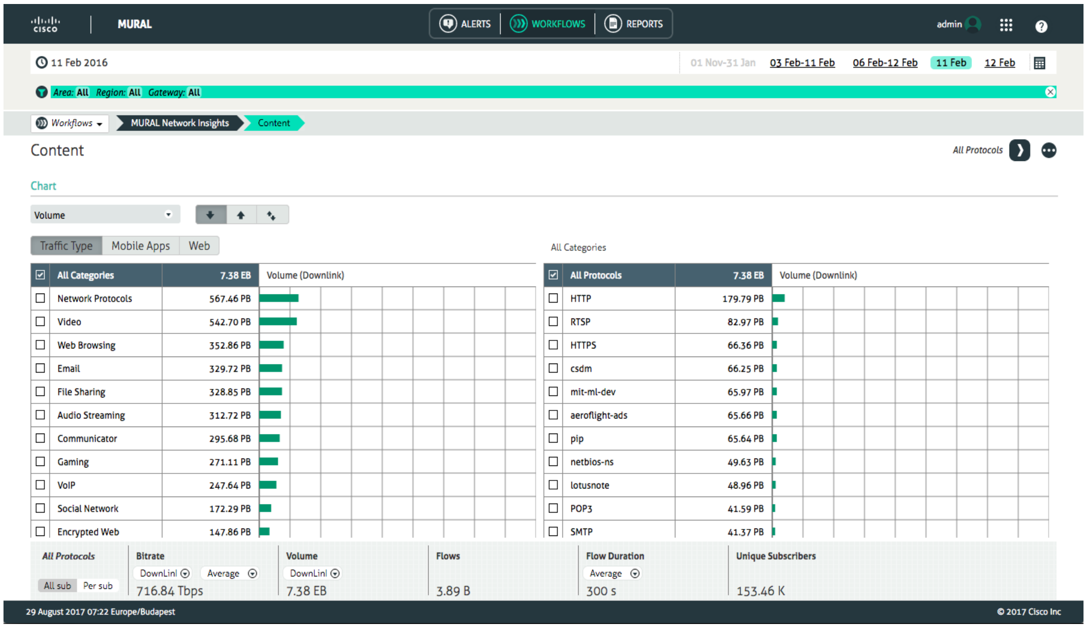
Device Models by Volume

☑ All Device Models	7.38 EB	Volume (Downlink)
□ Samsung SM N920T	382.36 PB	🟩
□ Apple iPhone 6 Plus	374.44 PB	🟩
□ Apple iPhone 6S	341.26 PB	🟩
□ Samsung Galaxy S5	335.78 PB	🟩
□ ZTE Z970	263.72 PB	🟩
□ Samsung Galaxy Note 3	231.80 PB	🟩
□ Samsung SM G920T	229.42 PB	🟩
□ Samsung Galaxy S6 Edge+	207.91 PB	🟩
□ LG MS631	178.06 PB	🟩
□ LG H811	162.59 PB	🟩
□ Samsung Galaxy S7	153.22 PB	🟩
□ Samsung SM G530T	148.73 PB	🟩

All Device Mod...	Bitrate	Volume	Flows	Flow Duration	Unique Subscribers
All sub Per sub	DownLinl Average 716.84 Tbps	DownLinl 7.38 EB	3.89 B	Average 300 s	153.46 K

29 August 2017 07:22 Europe/Budapest © 2017 Cisco Inc

Data on the Content Workflow



11 Feb 2016

01 Nov-31 Jan | 03 Feb-11 Feb | 06 Feb-12 Feb | 11 Feb | 12 Feb

Area: All Region: All Gateway: All

Content — All Protocols

Volume

☑ All Categories	7.38 EB	Volume (Downlink)
□ Network Protocols	567.46 PB	🟩
□ Video	542.70 PB	🟩
□ Web Browsing	352.86 PB	🟩
□ Email	329.72 PB	🟩
□ File Sharing	328.85 PB	🟩
□ Audio Streaming	312.72 PB	🟩
□ Communicator	295.68 PB	🟩
□ Gaming	271.11 PB	🟩
□ VoIP	247.64 PB	🟩
□ Social Network	172.29 PB	🟩
□ Encrypted Web	147.86 PB	🟩

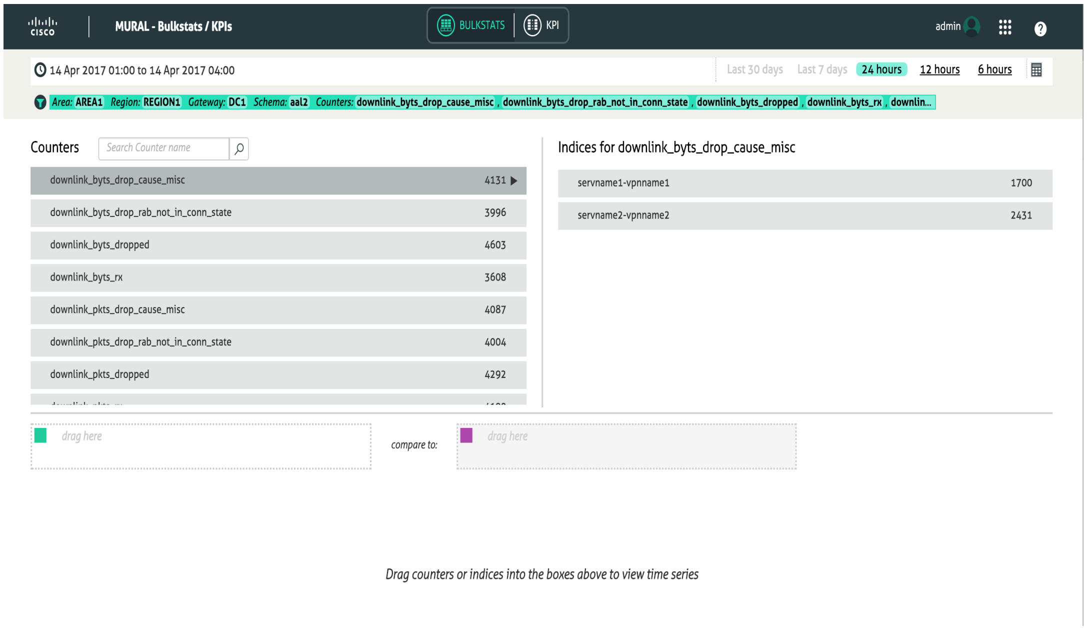
All Categories

☑ All Protocols	7.38 EB	Volume (Downlink)
□ HTTP	179.79 PB	🟩
□ RTSP	82.97 PB	🟩
□ HTTPS	66.36 PB	🟩
□ csdm	66.25 PB	🟩
□ mit-ml-dev	65.97 PB	🟩
□ aeroflight-ads	65.66 PB	🟩
□ pip	65.64 PB	🟩
□ netbios-ns	49.63 PB	🟩
□ lotusnote	48.96 PB	🟩
□ POP3	41.59 PB	🟩
□ SMTP	41.37 PB	🟩

All Protocols	Bitrate	Volume	Flows	Flow Duration	Unique Subscribers
All sub Per sub	DownLinl Average 716.84 Tbps	DownLinl 7.38 EB	3.89 B	Average 300 s	153.46 K

29 August 2017 07:22 Europe/Budapest © 2017 Cisco Inc

Data on the Bulkstats Tab



MURAL - Bulkstats / KPIs

BULKSTATS | KPI

admin

14 Apr 2017 01:00 to 14 Apr 2017 04:00

Last 30 days | Last 7 days | 24 hours | 12 hours | 6 hours

Area: AREA1 *Region:* REGION1 *Gateway:* DC1 *Schema:* aal2 *Counters:* downlink_byts_drop_cause_misc , downlink_byts_drop_rab_not_in_conn_state , downlink_byts_dropped , downlink_byts_rx , downlin...

Counters — Search Counter name

Counters	
downlink_byts_drop_cause_misc	4131 ▶
downlink_byts_drop_rab_not_in_conn_state	3996
downlink_byts_dropped	4603
downlink_byts_rx	3608
downlink_pkts_drop_cause_misc	4087
downlink_pkts_drop_rab_not_in_conn_state	4004
downlink_pkts_dropped	4292

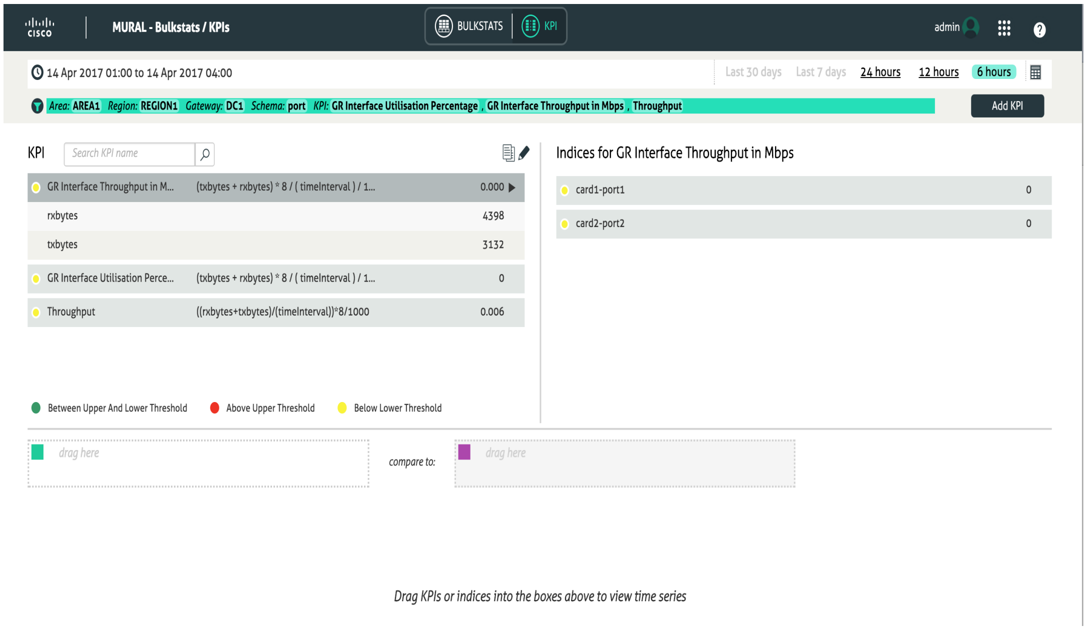
Indices for downlink_byts_drop_cause_misc

Indices	
servname1-vpnname1	1700
servname2-vpnname2	2431

🟩 *drag here* *compare to:* 🟪 *drag here*

Drag counters or indices into the boxes above to view time series

Data on the KPI Tab



MURAL - Bulkstats / KPIs

BULKSTATS | KPI

admin

14 Apr 2017 01:00 to 14 Apr 2017 04:00

Last 30 days | Last 7 days | 24 hours | 12 hours | 6 hours

Area: AREA1 *Region:* REGION1 *Gateway:* DC1 *Schema:* port *KPI:* GR Interface Utilisation Percentage , GR Interface Throughput in Mbps , Throughput

Add KPI

KPI — Search KPI name

KPI		
● GR Interface Throughput in M...	(txbytes + rxbytes) * 8 / (timeInterval) / 1...	0.000 ▶
rxbytes		4398
txbytes		3132
● GR Interface Utilisation Perce...	(txbytes + rxbytes) * 8 / (timeInterval) / 1...	0
● Throughput	((rxbytes+txbytes)/(timeInterval))*8/1000	0.006

● Between Upper And Lower Threshold ● Above Upper Threshold ● Below Lower Threshold

Indices for GR Interface Throughput in Mbps

Indices	
● card1-port1	0
● card2-port2	0

🟩 *drag here* *compare to:* 🟪 *drag here*

Drag KPIs or indices into the boxes above to view time series

D.4 Clean up the Pipe

Execute the script as follows:

```
[admin@VM-MUR-GCN1 DummyData]# cd /opt/etc/scripts/DummyData
[admin@VM-MUR-GCN1 DummyData]# python DummyData.py
************************************
Welcome to the Dummy Data tool :
 This tool gives you the option of creating dummy data and run
multiple jobs on it so that UI can be seen with dummy data
 Once dummy data is populated, this tool also gives the option of
cleaning up the dummy data and revert the system to its original
state
 Input required : /opt/etc/scripts/DummyData/config.txt
 Logs Path: /opt/etc/scripts/DummyData/log.txt
 ************************************
Enter valid input from below options :
 1. Run complete pipe
 2. Cleanup the system
 3. Exit
2
Have you entered the proper values in config.txt file . Enter yes
or no
yes
The Cleanup will do the following things :
=======================================================
 1. It will delete all folder except /data/CoreJob/config from
directory /data/CoreJob on local file system from all namenodes
  2. It will delete the /data/ib/ contents on local file system
from all servers
 3. It will delete the contents of /data/atlasData/ on local file
system from rubix servers
 4. It will remove data from /data/collector1/bulkstats_files_
backup/GMPLAB4/, /data/collector/bulkstats_files_backup/DC1/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector/bulkstats_files_backup/DC1/,
/data/collector/bulkstats_files_backup/GMPLAB1/,
/data/collector/bulkstats_files_backup/delhi/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector/bulkstats_files_backup/DC1/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector/bulkstats_files_backup/DC1/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector1/edrflow_backup, /data/collector1/edrhttp_backup on
local file system from all namenodes
 5. It will remove the hdfs contents from the path : /IB/,
/local/IB/,/data/ib/, /data/output/
 6. It will remove the hdfs content from the path:
/data/CoreJob,/data/EDR_EDRSubcrBytesAgg_lastSuccess,/data/EDR_
SubscriberDeviceMPH_lastSuccess,/data/EDR_EDRSubcrDev_
lastSuccess,/data/EDR_EDRTopN_lastSuccess
 7. It will remove hdfs content from the path
/data/TEST/edrhttp/,/data/TEST/edrflow/ and /data/TEST/bulkStats/
(if bulkstat is enabled)
 8. It will remove diskcache (/data/diskstore/UI-EDR) from all
rubix servers
 9. It will remove bulkstats cache (/data/diskstore/bulkstats-
<hostname>) from all rge nodes (applicable only if bulkstat is
enabled)
=======================================================

Do you want to proceed : yes or no yes



Welcome to Mural 4.1
resolved_active.xml                                100% 1618KB
1.6MB/s   1.6MB/s   00:00
```

```
Please check /opt/etc/scripts/DummyData/log.txt for logs
Cleanup finished Successfully
```

E. File Pattern Regular Expression

This topic provides information about the regular expression for the EDR and collector files.

The following list provides detailed description for the using the regular expression for EDR and Collector file names:

- Ensure to provide different input paths for each new gateway being added.
- The /edr-file-path and /bulkstat-file-path should always start with a forward slash (/).
- The ASR should send the gateway name in place of %DC, as specified in the file name pattern in the Collector configurations.
- If the filenames will have file extensions of .gz or .txt, then you must provide ".*" in the file name format configuration when adding the gateway.
- All incoming files should contain the string as per their type in the file name; that is, flow EDR files should contain the string "flow" delimited by an underscore (_) or hyphen (-) and similarly HTTP EDR files must contain string "http" delimited by an underscore (_) or hyphen (-) (so combinations would also work , like "_flow-" or "–http_").
- All collector (internal) IPs can be provided with comma (,) separated values like 10.10.10.133,10.10.10.134

Following are the guidelines and samples to help in configuring filename patterns and collector configurations:

FileName pattern	Example	TimeStamp	Regex for EDR filename	Regex for collector Filename
Gatewayname_(multiple strings separated by underscore or hyphen or both)_flow_timestamp_str4.gz	Gatewayname_str1_str2_str3_flow_timestamp_str4.gz	MMDDYYYY-hhmmss	*_*_*_%MM%DD%YYYY%hh%mm%ss_*.gz	%DC_*_*_%MM%DD%YYYY%hh%mm%ss*.gz
Gatewayname_(multiple strings separated by underscore or hyphen or both)_flow_timestamp_str4_str5.gz	Gatewayname_str1_str2_str3_flow_timestamp_str4_str5.gz	MMDDYYYY-hhmmss	*_*_*_%MM%DD%YYYY%hh%mm%ss_*_*.gz	%DC_*_*_%MM%DD%YYYY%hh%mm%ss*.gz

FileName pattern	Example	TimeStamp	Regex for EDR filename	Regex for collector Filename
Gateway-name_(multiple strings separated by underscore or hyphen or both)_flow_string_timestamp_string.gz	Gateway-name_str1_str2_str3_flow_str4_timestamp_str5.gz	MMDDYYYYhhmmss	*_*_*_*_%MM%DD%YYYY%hh%mm%ss_*.gz	%DC_*_*_*_%MM%DD%YYYY%hh%mm%ss*.gz

FileName pattern	Example	TimeStamp	Regex for EDR filename	Regex for collector Filename
Gateway-name_(multiple strings separated by underscore or hyphen or both)_flow_string_timestamp_string_string.gz	Gateway-name_str1_str2_str3_flow_str4_timestamp_str5_str6.gz	MMDDYYYYhhmmss	*_*_*_*_%MM%DD%YYYY%hh%mm%ss_*_*.gz	%DC_*_*_*_%MM%DD%YYYY%hh%mm%ss*.gz

FileName pattern	Example	TimeStamp	Regex for EDR filename	Regex for collector Filename
Gateway-name_(multiple strings separated by underscore or hyphen or both)_flow_string_timestamp_string_string.gz	Gateway-name_str1_str2_str3_flow_str4_timestamp_str5_str6.gz	YYYYMMDDhhmmss	*_*_*_*_%YYYY%MM%DD%hh%mm%ss_*_*.gz	%DC_*_*_*_%YYYY%MM%DD%hh%mm%ss*.gz

FileName pattern	Example	TimeStamp	Regex for EDR filename	Regex for collector Filename
Gatewayname_(multiple strings separated by underscore or hyphen or both)_flowstring_timestamp_string_string.gz	Gatewayname_str1_str2_str3_flowstr4_timestamp_str5_str6.gz	MMDDYYYYhhmmss	*_*_*-*_%MM%DD%YYYY%hh%mm%ss_*_*.gz	%DC_*_*-*_%MM%DD%YYYY%hh%mm%ss*.gz

- It is mandatory to send gateway name as the first substring in EDR files, followed by an underscore (_) character.
- 'flow' can be replaced with ‘http'.
- If timestamp is in MMDDYYYYhhmmss then set %MM%DD%YYYY%hh%mm%ss in Regex pattern.
- If timestamp is in YYYYMMDDhhmmss then set %YYYY%MM%DD%hh%mm%ss in Regex pattern.

F. Mandatory Parameters for Incoming ASR Files

The following is the list of mandatory headers that need to be present in files coming from the ASR so that the MURAL system can deduce meaningful information.

Note: MURAL recommends ASR files to be of 10MB compressed size for optimum performance.

F.1 Mandatory Attributes for Flow EDRs for MURAL

Flow EDR data sent by the ASR platform to the MURAL system must contain the following attributes:

- flow-end-time
- flow-start-time
- radius-calling-station-id
- sn-app-protocol
- p2p-protocol
- sn-server-port
- sn-volume-amt-ip-bytes-downlink
- sn-volume-amt-ip-pkts-uplink
- sn-volume-amt-ip-pkts-downlink
- sn-volume-amt-ip-bytes-uplink
- tcp-os-signature
- bearere-3gpp imei

Sample:

```
#sn-start-time,sn-end-time,radius-calling-station-id,sn-server-
port,sn-app-protocol,sn-volume-amt-ip-bytes-uplink,sn-volume-amt-
ip-bytes-downlink,sn-volume-amt-ip-pkts-uplink,sn-volume-amt-ip-
pkts-downlink,p2p-protocol,ip-server-ip-address,bearer-3gpp rat-
type,voip-duration,sn-direction,traffic-type,bearer-3gpp
```

```
imei,bearer-3gpp sgsn-address,bearer-ggsn-address,sn-flow-end-
time,sn-flow-start-time,radius-called-station-id,bearer-3gpp user-
location-information,sn-subscriber-port,ip-protocol,sn-
rulebase,tcp-os-signature,bearer-3gpp charging-id

1381518310,1381518337,1000000018,70000,29,20000,20000,182,36,iax,27
.9.126.155,2,1,FromMobile,,,2.2.2.1,27.23.157.2,1381518337,13815183
10,Sushfone-2,231-10-1073-10065,43769,1985,rb31,,2
```

F.2 Mandatory HTTP EDR Attributes for MURAL

HTTP EDR data sent to the MURAL system must contain the following attributes:

- sn-start-time
- sn-end-time
- transaction-downlink-packets
- transaction-uplink-packets
- transaction-downlink-bytes
- transaction-uplink-bytes
- http-content type
- radius-calling-station-id
- http-User-Agent
- http-URL
- http-host
- http-reply code
- tcp-os-signature
- bearer-3gpp imei

Sample:

```
#sn-start-time,sn-end-time,radius-calling-station-id,transaction-
uplink-bytes,transaction-downlink-bytes,ip-subscriber-ip-
address,ip-server-ip-address,http-host,http-content type,http-
url,voip-duration,traffic-type,transaction-downlink-
packets,transaction-uplink-packets,bearer-3gpp rat-type,radius-
called-station-id,tcp-os-signature,bearer-3gpp imei,http-request
method,http-reply code,http-user-agent

1381518310,1381518338,1000000019,15000,15000,1.1.1.1,27.2.248.155,i
mages.craigslist.org,image/png,images.craigslist.org,11,,60,1,1,Sus
hfone-1,,,GET,506 Variant Also Negotiates,"Dalvik/1.6.0 (Linux; U;
Android 4.0.3; Galaxy Nexus Build/ICL53F)"
```

F.3 ASR-Side Configuration

The corresponding configuration on the side of the ASR platform is as follows:

```
    edr-format edr-flow-format
attribute sn-start-time format seconds priority 10
attribute sn-end-time format seconds priority 20
attribute radius-calling-station-id priority 30
rule-variable bearer 3gpp imsi priority 35
attribute radius-called-station-id priority 40
attribute sn-volume-amt ip bytes uplink priority 50
attribute sn-volume-amt ip bytes downlink priority 60
attribute sn-volume-amt ip pkts uplink priority 70
attribute sn-volume-amt ip pkts downlink priority 80
rule-variable bearer 3gpp imei priority 90
rule-variable bearer 3gpp rat-type priority 100
rule-variable p2p protocol priority 110
attribute sn-app-protocol priority 120
attribute sn-parent-protocol priority 130
rule-variable ip protocol priority 140
rule-variable traffic-type priority 150
attribute sn-direction priority 160
```

```
rule-variable ip server-ip-address priority 170
attribute sn-server-port priority 180
rule-variable ip subscriber-ip-address priority 190
attribute sn-subscriber-port priority 200
rule-variable bearer 3gpp sgsn-address priority 210
rule-variable bearer ggsn-address priority 220
rule-variable bearer 3gpp user-location-information priority 230
rule-variable bearer 3gpp2 bsid priority 240
attribute sn-flow-start-time format seconds priority 260
attribute sn-flow-end-time format seconds priority 270
rule-variable tcp os-signature priority 290
rule-variable tethered priority 300
attribute sn-rulebase priority 310
#exit

edr-format edr-http-format
attribute  sn-start-time format seconds priority 10
attribute  sn-end-time format seconds priority 20
attribute radius-calling-station-id priority 30
attribute radius-called-station-id priority 40
rule-variable http host  priority 70
rule-variable http content type  priority 80
attribute transaction-downlink-bytes priority 90
attribute transaction-uplink-bytes priority 100
attribute transaction-downlink-packets priority 110
attribute transaction-uplink-packets priority 120
rule-variable bearer 3gpp imei  priority 130
rule-variable bearer 3gpp rat-type  priority 140
rule-variable http reply code priority 150
rule-variable http url priority 160
rule-variable http referer priority 170
rule-variable http user-agent priority 180
#exit
```

G. Understanding the CIQ Sheet

Customer Information Questionnaire (CIQ) sheet provides key information related to configuration of the network and storage. CIQ sheet serves as a reference document for the various values related to network, clusters, nodes and storage. The CIQ sheet is has following tabs:

- "MiscInfo" below
- "NetworkInfo" on the next page
- "ClusterInfo" on page 108
- "NodeInfo" on page 108
- "StorageInfo" on page 109

G.1 MiscInfo

The MiscInfo tab provides miscellaneous information, for example site name. The information provided in this tab is used for providing the names of various XMLs generated during the installation process.

The following table describes fields in the MiscInfo tab.

Field Name	Description	Example
XML File Name	Name of the XML file to be created by system after uploading the CIQ sheet.	Mural_Hardware_StandardPack.xml
Site Name	Name of the site.	InterNap
HW Type	The type of hardware used at the site.	Cisco
Banner	The welcome text to be displayed on the console	Welcome to Mural Standard Pack
Set GSM as NTP	Whether to use GSM server as NTP server.	FALSE
CIQ Template Version	The current version of the CIQ sheet.	1.0

G.2 NetworkInfo

The NetworkInfo tab provides information related to various network parameters that are required to be configured. The network information is categorized into VLAN, UCS Management Interfaces and Misc Network.

The following table describes fields in the NetworkInfo tab.

Field Name	Description	Example
VLAN		
Network Name	Name of network, whether public or management.	MgmtNW
Interface	The name of the interface.	eth0
Network Type	The type of network, whether normal or management.	Mgmt/Data
Network Prefix	The network prefix for the IP.	192.168.188.0
Subnet	The subnet or the value of subnet.	24
Vlan ID	The Vlan ID.	Not in use
No of Addresses	Number of IP addresses to be used.	Not in use
Remark	The cell is provided to be used for providing and additional information.	Not in use
UCS Management Interfaces		
Fabric A IP	Fabric IP of interface A.	Not in use
Fabric B IP	Fabric IP of interface B.	Not in use
Cluster VIP	UCS manager IP.	17.10.10.9
Domain	domain name used for login to UCS.	native
Username	username to login to the UCS manager.	username
Password	Password to login to the UCS manager.	password
Misc Network		
Default Gateway	The default gateway IP.	172.30.0.1
DNS	The DNS IP.	103.14.2.35
NTP IP	NTP server IP.	103.14.2.37
NTP Version	The version of the NTP.	4

Field Name	Description	Example
SNMP Community Name	Name of the SNMP used.	cisco
SNMP Receiver IP	The IP of the SNMP server.	172.30.17.75
SNMP Port	The port of the SNMP server.	162

G.3 ClusterInfo

The ClusterInfo tab provides information related to various clusters to be created on the hardware. It provides information such as Clustername, Cluster interface, cluster VIP and so on.

The following table describes fields in the ClusterInfo tab.

Field Name	Description	Example
Cluster Name	Name of the cluster.	cmp-clu
Cluster Type	Type of the cluster setup.	1+1 HA
Cluster Interface	Name of the interface connection for the cluster.	eth0
Eth0 VIP	VIP of eth0 interface.	192.168.188.75
Eth1 VIP	VIP of eth1 interface.	172.30.17.75
Eth2 VIP	VIP of the eth2 interface.	172.30.17.67

G.4 NodeInfo

The NodeInfo provides detailed information related to each node in the system. It provides information like chassis number, slot number, hostname, cluster name etc. These information are required to uniquely identify each node present in the system.

The following table describes fields in the NodeInfo tab.

Field Name	Description	Example
Chassis Number	The chassis number in which serverin installed.	c1
Slot Number	The slot number where server is installed.	1
Hostname	The host name of the sever.	gms-1
Eth0 MAC	The MAC address of the eth0 interface.	00:A0:D7:42:08:7E
Eth0 IP	The MAC IP of the eth0 interface.	192.168.188.66
Eth1 MAC	The MAC address of the eth1 interface.	00:A0:D7:42:08:6E
Eth1 IP	The MAC IP of the eth1 interface.	172.30.17.66
Eth2 MAC	The MAC address of the eth2 interface.	00:A0:D7:42:08:7E
Eth2 IP	The MAC IP of the eth2 interface.	172.30.17.98
KVM IP	The IP of the KVM assiciated with the server.	17.18.10.0
Cluster Name	The name of the cluster.	gms-clu

G.5 StorageInfo

The StorageInfo provides information related to various parameters of the storage. The storage information provided are classified into two storage information, which provides general storage information and EMC interfaces, which provides information related to the EMC interfaces for the storage.

The following table describes fields in the StorgaeInfo tab.

Field Name	Description	Example
Storage Information		
Hostname	Host name of the server.	gms-1
WWID	The WWID of the storage associated with the server.	36006016085003600ead331eb56f7e311
Component	The name of the components using the assigned storage.	PGSQL

Field Name	Description	Example
LUN name	The name of LUN associated storage.	only for information
LUN size (GB)	The LUN size of associated storage	only for information
RAID type	The RAID type used for storage.	only for information
EMC Interfaces		
EMC a	EMC interface a.	not in use
EMC b	EMC interface b.	not in use
username	Username to connect with EMC interface.	not in use
password	Password corresponding to the username to connect with the EMC interface.	not in use