Cisco MURAL Software Installation Guide

Version 3.2

July 8, 2013

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

Text Part Number: OL-27417-02

MURAL Software Installation Guide

CONTENTS

CHAPTER 1

Introduction

This document describes the process of installing the Mobility Unified Reporting and Analytics (MURAL) application. It is intended for deployment engineers who are installing the MURAL application with Cisco ASR 5000 platforms.

MURAL provides Web-based reporting and analytics abilities for Deep Packet Inspection (DPI) data emerging from your network.

MURAL provides dashboard capabilities for network traffic analysis, content distribution analysis, 5 device traffic distribution, subscriber traffic patterns, reporting for bulk stats and key performance indicators (KPIs) generated from different components of the ASR 5000.

MURAL also offers scheduled offline reports, tethering detection, reporting enhancements and OAM enhancements.

It is assumed the reader has a working knowledge of the following:

- Linux
- Cisco UCS
- EMC SAN
- It is also recommended that the reader complete a training course in MURAL prior to installing the application.

Installation Package Components

The MURAL installation package contains the following components:

- ISO image
- Patches
- Sample XML to be used for GMS
- MIBS

Prerequisites

This section lists some things you should do before beginning the MURAL installation process.

Customer Information Questionnaire

The Customer Information Questionnaire (CIQ) is a site survey that you should complete before beginning the installation process. The survey is in an Excel spreadsheet. Please fill in all details in the spreadsheet that pertain to your installation, so that you can provide the required input when it is requested by the install scripts.

The Excel file contains the following worksheets:

- **Contacts** - Identify site personnel and their responsibilities.
- **Space_Power Req** - Supply space and power requirements for the Cisco UCS Server chassis and the UCS 5108 Server chassis.
- **IP Survey** - Supply specifics for physical network connections, VLANs, various interfaces, SNMP/SMTP settings, and so forth.
- **Network Diagrams** - Shows the system components and how they are connected.
- **Connectivity** - Supply the details for ports and connections.
- **Firewall** - Identifies the firewall changes required for connectivity
- **Alarms** - Lists and describes all SNMP traps supported by the application
- **ASR5K** - Supply locations for various ASR 5000 information bases (IBs) required by the application.

System Components

The MURAL platform consists of the following components:

- **General Management Server (GMS)**—A node that enables a centralized installation, rather than requiring manual installation and configuration of software on each blade.
- **Collector node**—Consists of a cluster of Collectors that collect data from the ASR 5000. The Collector is optimized for low-latency, high-throughput transactions, and it assembles and understands the exported flows. The Collector node distributes data to the local compute node.
- **Compute node**—Also called the Data node. Analyzes and aggregates data. It is connected by cable to the Collector nodes and consists of multiple clusters. The Compute node sends the data to the storage array and makes it available to applications.
- **Insta node**—Also called the Caching Compute node. Generates and manages caches of processed data. The processed data (cubes) are stored in the Insta database (infinidb). The data is generally hosted on a separate SAN device, though sometimes hosted on the Insta node itself.
- **UI node**—Includes both the Cube engine (Rubix) and the RG (report generation) engine. The Cube engine forwards requests from the UI engine to the Insta node. It also prefetches data and locally caches it so that if the requested data is in the local cache, it can return the response directly to the UI node without querying the Insta node. The RG engine serves as the HTTP request server.

The Installation Process

The following is a brief description of the MURAL installation process. Detailed instructions are provided in Chapter 2 of this guide.

Step 1 Set up the hardware.

Note The hardware components should match the details listed in the bill of materials (BoM).

Step 2 Configure UCS

Step 3 Configure EMC

Step 4 Manufacture all the blades.

Step 5 Configure GMS using the xml file provided in the installation package.

CHAPTER 2

Installation

This chapter explains how to build the nodes in a Collection Data Center. Some tasks have multiple commands that must all be executed in sequence. Examples are provided for commands that generate output.

Warning **Skipping a task or performing the tasks out of sequence may cause a misconfiguration resulting in system failure.**

Prerequisites

Before you begin the installation process, verify the following:

- All 10 UCS B200 M2/M3 blade servers (2 Collector nodes, 4 Compute nodes, 2 Caching Compute nodes, and 2 UI nodes) have been installed physically on the UCS 5108 Chassis and connected to the UCS 6248 Fabric Interconnect.

Note Hardware should be set up as specified in the bill of materials (BOM).

- Each Fabric Interconnect is connected to the SAN disk through a Fibre Channel providing multipath configurations.
- You have completed the Customer Information Questionnaire (CIQ). The survey is in an Excel spreadsheet. Please fill in all details in the spreadsheet that pertain to your installation, so that you can provide the required input when it is requested by the install scripts.

Configuring UCS on MURAL

This section explains the initial configuration and advanced setup procedures for the UCS on the MURAL platform.

Before you begin, verify that you have all items listed in the BOM.

The following is a brief summary of the configuration process:

Step 1 Hardware Configuration

a. Blade locations

b. Fabric interconnects

c. SAN wiring

d. Network up-links

Step 2 Configure UCS

a. Base Configurations

b. CLI script for advanced configuration

c. GUI verification

Prepare for Initial Configuration

Before you begin configuring the fabrics, verify the following physical connections:

- A console port on the first fabric interconnect is physically connected to a computer terminal or console server
- The management Ethernet port (mgmt0) is connected to an external hub, switch, or router
- The L1 ports on both fabric interconnects are directly connected to each other
- The L2 ports on both fabric interconnects are directly connected to each other

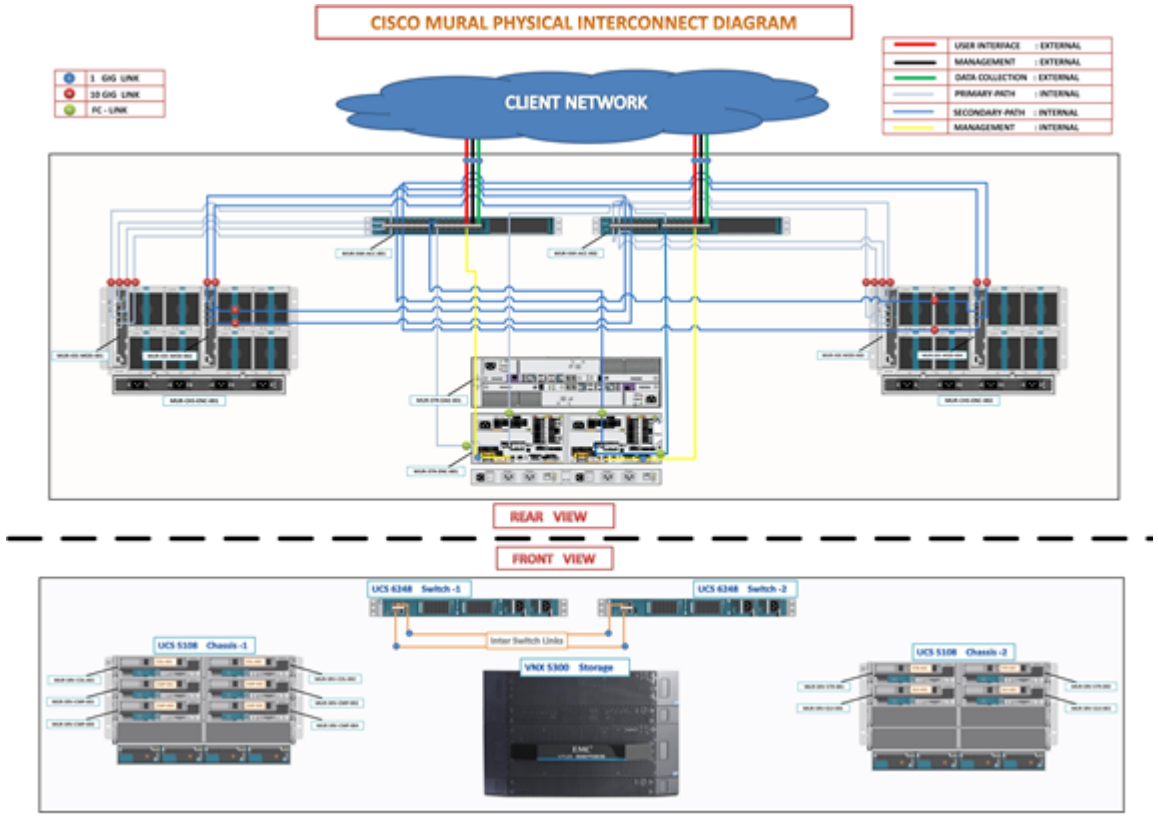
Verify that the console port parameters on the computer terminal (or console server) attached to the console port are as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

Deployment Topology

The standard MURAL deployment has been tested and validated for the topology shown in Figure 2-1.

Figure 2-1 Deployment Topology



Hardware Configuration

Configure the blades using the diagram shown. Figure 2-2 shows the configuration in each chassis. All blades are identical, with the exception of the Rubix / UI blade, which has two to three times more RAM to support the MURAL application.

Figure 2-2 UCS Blade Hardware Configuration

.

Install the Blades

Install the blades in Chassis 1 and Chassis 2 as shown in Table 2-1 and Table 2-2. All blades are identical except the Rubix/UI blade, which has more RAM. Slots are numbered 1-8 from top to bottom, left to right.

Table 2-1 Chassis 1 (Fabric A) slot assignments

Chassis	Slot	Function
Chassis 1	Slot 1	Collector 1
Chassis 1	Slot 2	Compute 1
Chassis 1	Slot 3	Compute 2
Chassis 1	Slot 4	Insta 1
Chassis 1	Slot 5	Rubix/UI 1
Chassis 1	Slot 6	GMS 1

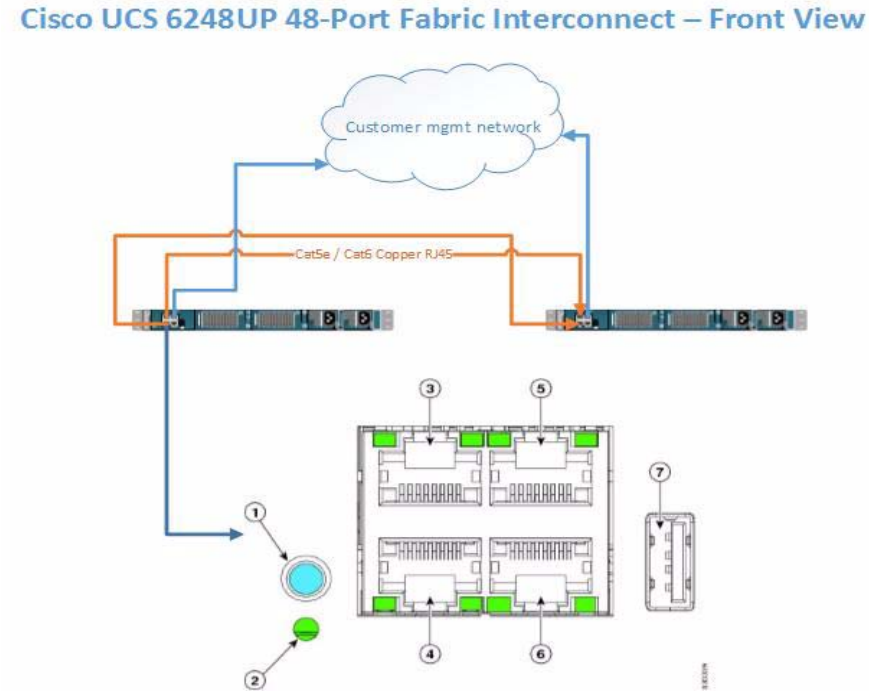
Table 2-2 Chassis 2 (Fabric B) slot assignments

Chassis	Slot	Blade
Chassis 2	Slot 1	Collector 2
Chassis 2	Slot 2	Compute 3
Chassis 2	Slot 3	Compute 4
Chassis 2	Slot 4	Insta 2
Chassis 2	Slot 5	Rubix/UI 2
Chassis 2	Slot 6	GMS 2 (optional)

Install the Fabric Interconnect

Figure 2-3 illustrates the hardware configuration as it relates to the MURAL components.

Figure 2-3 Cisco UCS 6248UP fabric interconnect--front view



3 UCS cross connect Port L1

4 UCS cross connect Port L2

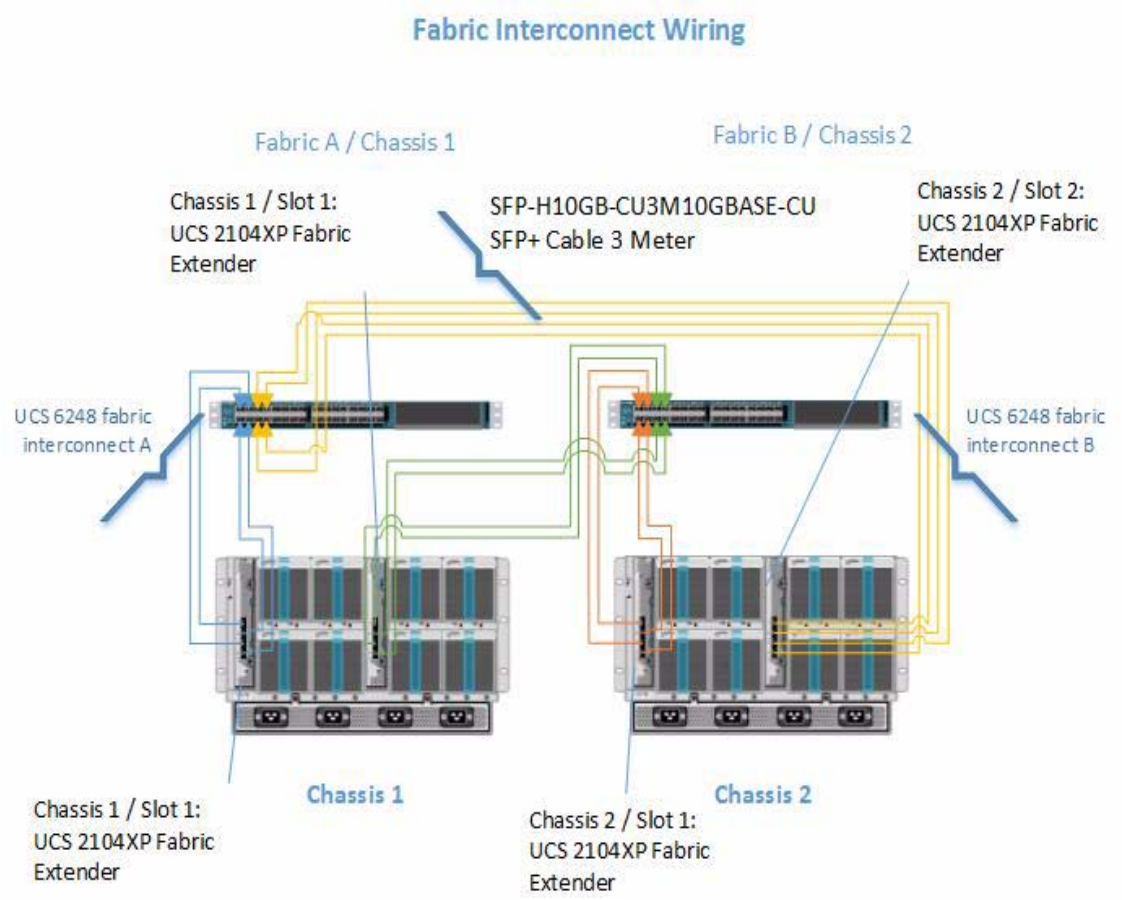
5 Network Management Port

6 Console Port

Connect the Fabrics

Make the fabric interconnections as shown in Figure 2-4:

Figure 2-4 Fabric Interconnect Wiring



Make chassis 1 connections to Fabric interconnects as shown in Table 2-3:

Table 2-3

UCS 2104XP Fabric Extender Slot 1	**UCS 6248 Fabric Interconnect A**
Chassis 1 : Extender 1 : Port 1	Interconnect A : Port 1
Chassis 1 : Extender 1 : Port 2	Interconnect A : Port 2
Chassis 1 : Extender 1 : Port 3	Interconnect A : Port 3
Chassis 1 : Extender 1 : Port 4	Interconnect A : Port 4
UCS 2104XP Fabric Extender Slot 2	**UCS 6248 Fabric Interconnect B**
Chassis 1 : Extender 2: Port 1	Interconnect B: Port 5
Chassis 1 : Extender 2: Port 2	Interconnect B: Port 6
Chassis 1 : Extender 2: Port 3	Interconnect B : Port 7
Chassis 1 : Extender 2: Port 4	Interconnect B: Port 8

Make chassis 2 connections to Fabric interconnects as shown in Table 2-4:

Table 2-4

UCS 2104XP Fabric Extender Slot 1	UCS 6248 Fabric Interconnect B
Chassis 2: Extender 1 : Port 1	Interconnect B : Port 1
Chassis 2: Extender 1 : Port 2	Interconnect B: Port 2
Chassis 2 : Extender 1 : Port 3	Interconnect B : Port 3
Chassis 2 : Extender 1 : Port 4	Interconnect B : Port 4
UCS 2104XP Fabric Extender Slot 2	**UCS 6248 Fabric Interconnect A**
Chassis 2 : Extender 2: Port 1	Interconnect A: Port 5
Chassis 2 : Extender 2: Port 2	Interconnect A: Port 6
Chassis 2 : Extender 2: Port 3	Interconnect A : Port 7
Chassis 2 : Extender 2: Port 4	Interconnect A: Port 8

UCS SAN Up-links

Connect the UCS SAN up-links as shown in Figure 2-5 and Table 2-5:

Figure 2-5 UCS SAN Uplinks

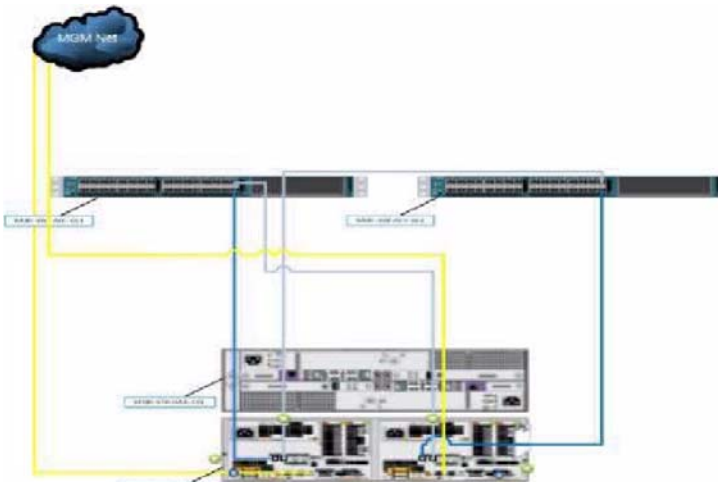


Table 2-5

SP-A management port	Customer management switch
SP-B management port	Customer management switch
SP-A FC-A	Fabric A--Port 31
SP-A FC-B	Fabric B--Port 31
SP-B FC-A	Fabric A--Port 32
SP-A FC-B	Fabric B--Port 32

UCS Network Up-links

Connect the UCS network up-links as shown in Figure 2-6 and Table 2-6:

Figure 2-6 UCS Network Uplinks

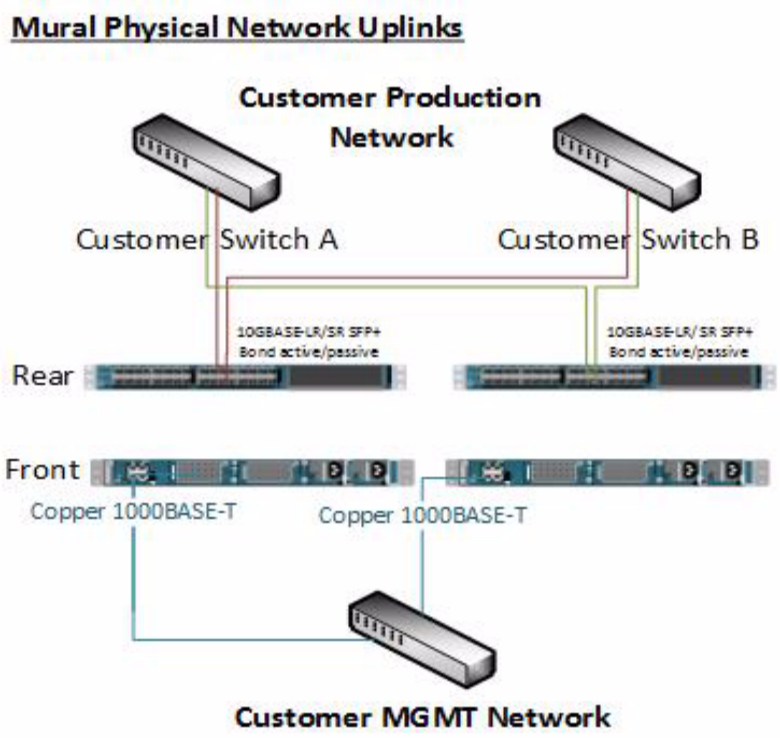


Table 2-6

Fabric A--port 17	Customer production network
Fabric B--port 17	Customer production network
Fabric A--port 18	Customer secondary production switch (optional)
Fabric B--port 18	Customer secondary production switch (optional)

Base Configuration for UCS System

You use the CLI configuration procedure to bring up the UCS fabrics. Once the fabrics are set up with their initial configuration, which includes setting up the management port IP address, proceed to use the CLI and config script for the Advanced UCS setup.

Set up management interfaces to UCS by using the console port on the front of each fabric switch and use the following setting; 9600 baud, 8 data bits, 1 stop bit, no parity.

Follow this procedure to bring up a fresh UCS installation:

Step 1 Connect to the fabric's A console, and perform the console system configuration process with the following parameters:

Configuration method: **console**

Setup mode: **setup**

New fabric interconnect: **Y**

Enforce strong password: **Y**

Admin password: <admin password>

Is this Fabric Interconnect part of a cluster: **Y**

Switch fabric: **A**

System Name: <UCS name, without a trailing -A or –B>

Mgmt0 IP address: <Fab A management port IP address>

Mgmt0 Netmask: <management port IP netmask>

IPv4 default gateway: <gateway address in the management subnet>

Cluster IPv4 address: <Virtual IP used by the active node, usually belonging to the management subnet>

DNS server address and the unit's domain name can be configured, but are not mandatory

Step 2 Connect to the fabric's B console and make sure the redundancy cables between the two fabrics are connected. Perform the initial configuration with the following parameters:

Configuration method: **console**

This fabric interconnect will be added to the cluster: **Y**

Admin password of interconnect: <admin password used in the configuration of Fab A>

Mgmt0 IP address: <Fab B management port IP address>

At this point, after the nodes reboot, it should be possible to log on to the management UI. You can access the management UI from a web browser at http://<ip address of the cluster>

Advanced UCS Configuration

The configuration for UCS can be programmed using the CLI and a simple script.

Files Needed

ucs-config-<*version-number*>.txt (where <*version-number*> is the most recent version available).

ucs_config.exp

Note These files can be obtained from either Cisco Advanced Services, or Technical Support. You will need to adapt **ucs_config.txt** for your local setup.

Requirements

The script needs to run from either Cygwin, Linux, or a Mac terminal.

Step 1 Start by editing the **ucs-config-<*version-number*>.txt** file and modifying every value marked with a CIQ label.

Step 2 Save and re-name the completed **ucs.txt** file into the same <dir> as the **ucs_config.exp** script.

Step 3 Verify you can ping the UCS management IP before continuing.

Step 4 Usage: **/ucs.exp ucs_mgmt_ip ucs_password**

Step 5 Run the script and watch for any errors or issues.

Note If you encounter an error, the easiest way to recover is to reset the UCS to **defaults.ssh** to the UCS manager; you need to do this for both A and B sides.

Setting Up the EMC

This section describes how to set up EMC Unisphere to manage the storage environment.

Prerequisites

Before beginning, verify that you have completed the following:

- Installed the VNX storage chassis and SPS (Standby Power Supply) chassis in the rack following the instructions provided in the installation guide (EMC P/N 300-012-924) included with the hardware.
- Connect SPS to SP (Storage Processor) management ports (using cables provided with the product) following the instructions provided in the installation guide.
- Connect the power cords from SPS A / SPS B to SP A and SP B and from SPS A and SPS B to PDUs following the instructions provided in the installation guide.
- The Fiber Channel SFP + included with the hardware is installed in Ports 4 and 5 of both SP A and SP B.
- Obtained IP addresses for the two SP management ports and other items listed in the following table:

Item	Value
SP A management port IP	
SP B management port IP	
Subnet mask and gateway for above	
Admin name/password	
Storage system serial number	
Scope	
DNS server address	Optional
Time server address	
In-bound email address	

Note Addresses in the 128.121.1.248-56 and 192.168.1.1 and 2 cannot be used.

Note Leave the storage system uncabled to the server array until after initialization is complete.

Configure the Base IP for Service Processors

The default IP's for the system are 1.1.1.1 and 1.1.1.2. By configuring your laptop to a similar range, you can connect to 1.1.1.1 using a web browser and reconfigure the IP address information.

Step 1 After configuring your laptop's IP to 1.1.1.4/24, connect a cable to Service Processor A.

Step 2 Open a web browser to http://1.1.1.1/setup

Step 3 Reconfigure the IP's to the desired range.

Note If you need to manually restart EMC during the set up procedures described in this section, open a web browser to **http://1.1.1.1/setup**. Log in as admin, and select the restart option.

Step 1: Register All the Nodes

Using the WWNN and WWPN you setup on the UCS side, you need to register each WWPN with a label indicating which machine it is from.

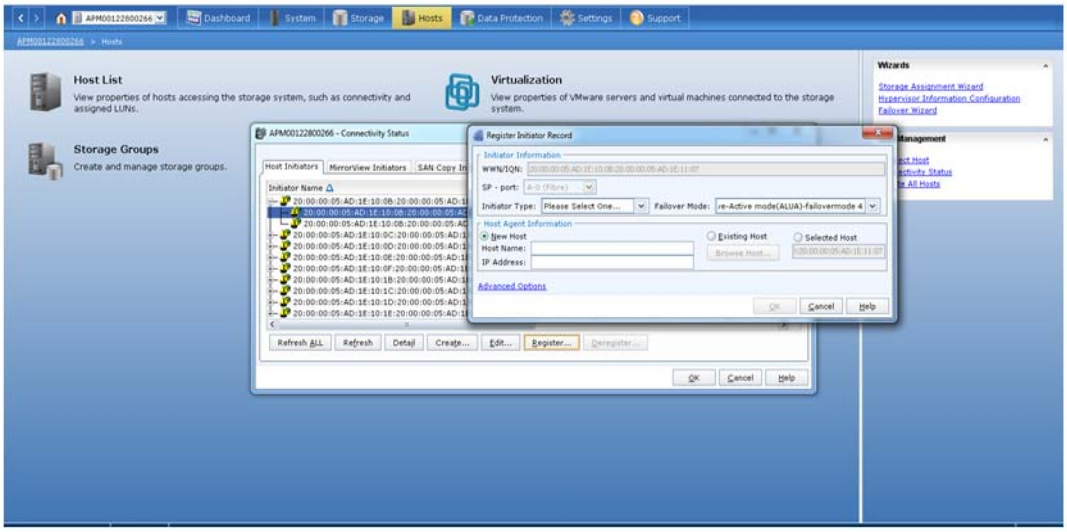
Note The WWPN for each blade can be displayed from the UCS manager; typically the last two digits are enough to identify it on the EMC interface.

Nodes:

Node – WWNN	WWPN – 1	WWPN - 2
Collector1	B4	B5
Collector2	B6	B7
Compute1		
Compute2		
Compute3		
Compute4		
Insta1		
Insta2		
UI1		
UI2		

a. To register the nodes, mouse over (not click) on <Hosts>, and select **Connectivity Status**.

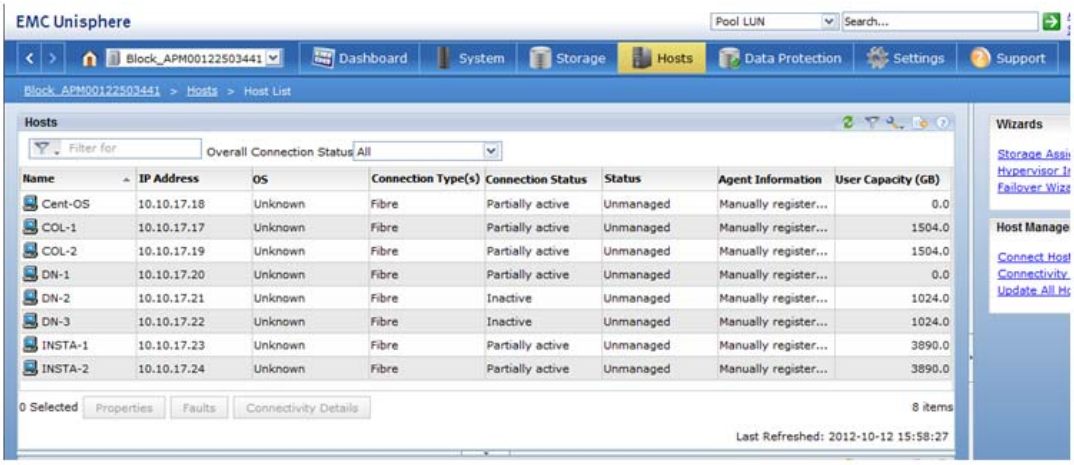
b. When the following screen displays, select **Register** to enter the registration information.



For each FC port, you will need the FC port World Wide Name (WWN)— a unique 16-digit hexadecimal number, such as 21-00-00-30-D9-00-12-34. This identifier is hard-coded into every FC host bus adapter (HBA). These WWNs (similar in concept to the MAC address in Ethernet cards) can be obtained from the HBA BIOS utility or HBA management software.

Note The Initiator type is SGI from the pull down list. The host is the host highlighted. Host Name: ideally, should be the same as on the UCS. IP Address is the Management IP. Failover is set to mode 4 ALUA (default).

c. Go to the UniSphere management agent, and view the hosts via <Dashboard>, <Host List>. You will see a list of hosts similar to the following:



Note Always check for alerts. When you start this process, there will be alerts saying that nodes are not registered (Alert 0x721c). These are normal until provisioning is complete.

Step 2: Create RAID Groups and Assign Them to Available Disks

Create the following RAID groups:

RAID group 10

Storage pool id: 10

Disk: 10,11,12,13

RAID type: 10

RAID group 5

Storage pool id: 5

Disk: 4,5,6,7,8

RAID type: 5

RAID group 100

Storage pool id: 100

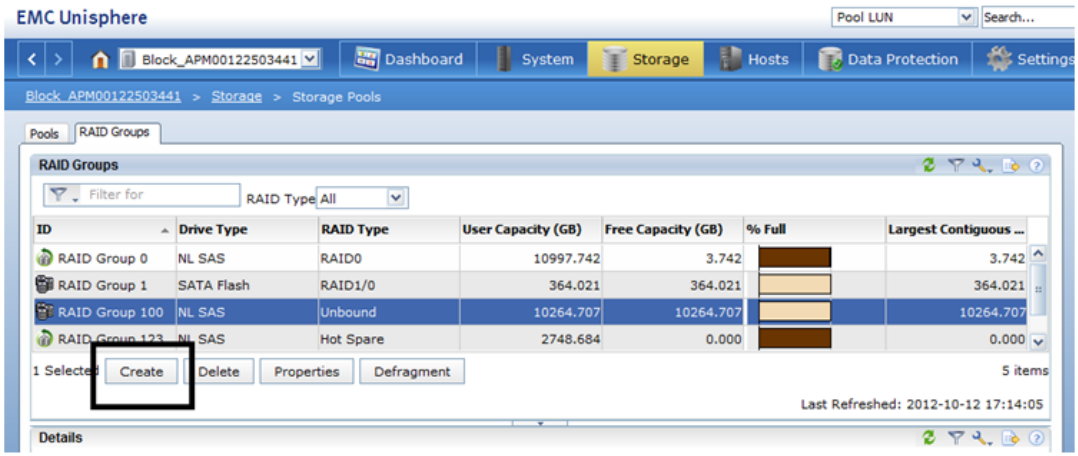
RAID type: unbound

Disk: 0,1,2,3

To Create Storage Pools:

a. Go to <Storage><Storage Pools>,<RAID Group> and click on “Create” as highlighted below.

b. Use the settings for the storage pools shown on the following screen.

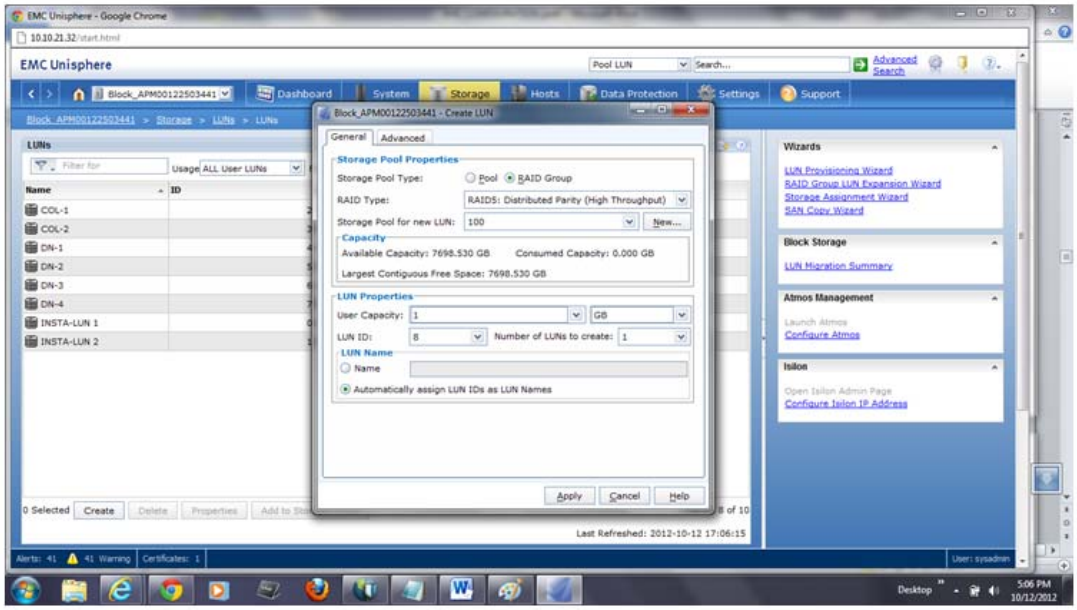


Creating LUNS

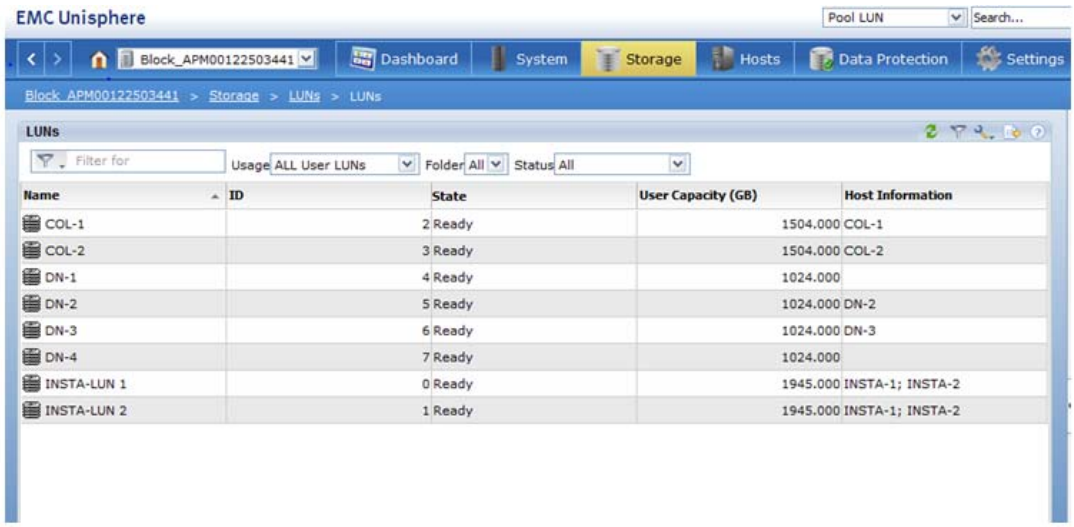
Refer to the following table for LUN creation information.

RAID	RAID Group Name	LUN Name	LUN ID	Disk Size (GB)	Controller	Storage Pool	HOST - MAP
RAID 10	RAID Group 10	INSTA-1	0	1945.6	FAB-A	INSTA-STR-1	INSTA Node-1
RAID 10	RAID Group 10	INSTA-2	1	1945.6	FAB-B	INSTA-STR-2	INSTA Node-2
RAID 5	RAID Group 5	COL-1	2	1024	FAB-A	COL-STR-1	COL Node-1
RAID 5	RAID Group 5	COL-2	3	1024	FAB-B	COL-STR-2	COLNode-2
RAID 5	RAID Group 5	DN-1	4	1024	FAB-A	DN-STR-1	DN -1
RAID 5	RAID Group 5	DN-2	5	1024	FAB-B	DN-STR-2	DN -2
RAID 5	RAID Group 5	DN-3	6	1024	FAB-A	DN-STR-3	DN-3
RAID 5	RAID Group 5	DN-4	7	1024	FAB-B	DN-STR-4	DN-4
RAID 5	RAID Group 5	UI1	8	1024	FAB-A	UI-STR-1	UI-1
RAID 5	RAID Group 5	UI2	9	1024	FAB-B	UI-STR-2	UI-2

To Create a LUN, go to <Storage><LUNS><LUNS> and click on “create.” LUN ID’s auto-increment. Disks that are already assigned are not available. Use the Names shown in the screen to create the LUNs shown



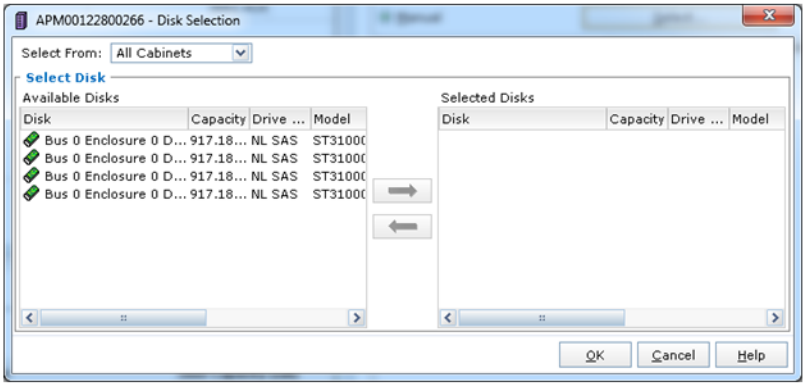
When you are done, go to <Storage><LUNS> and you will see a screen similar to the following.



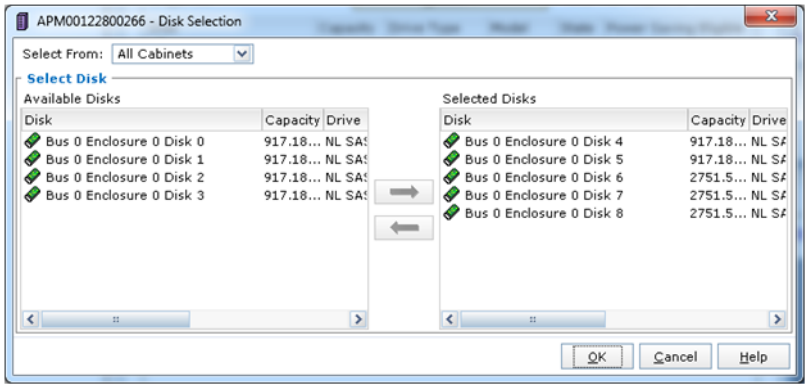
Note the IDs of the LUNs.

Note The sample screen does not show an LUN for the UI. However, it is required.

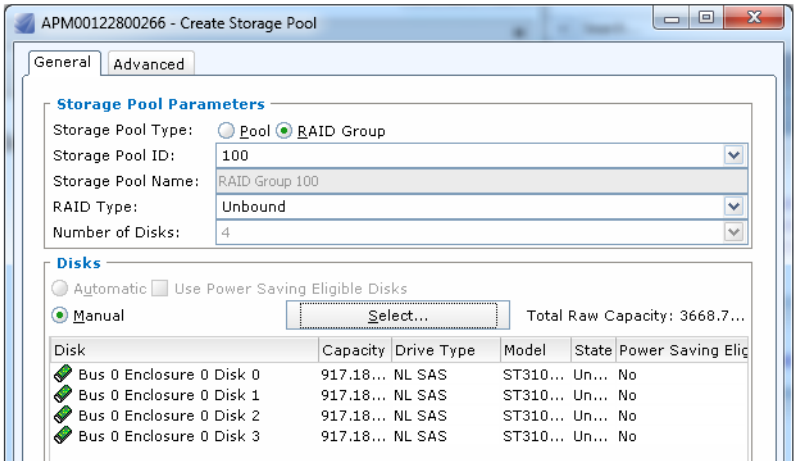
The following window is displayed in the Web UI when the **Disks** option is selected:

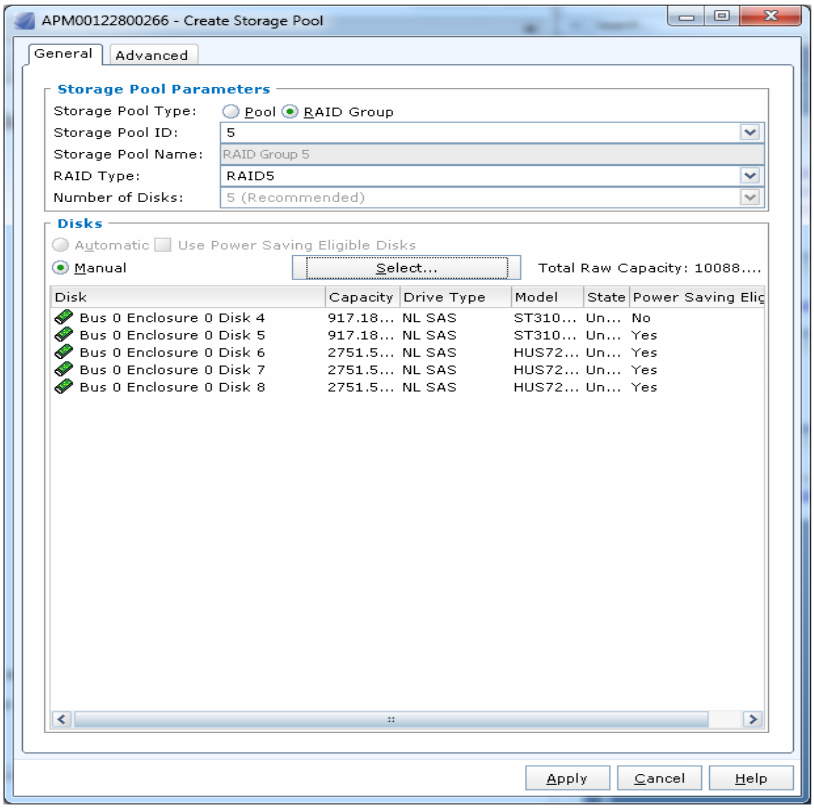


Move disks from **Available Disks** to **Selected Disks**.



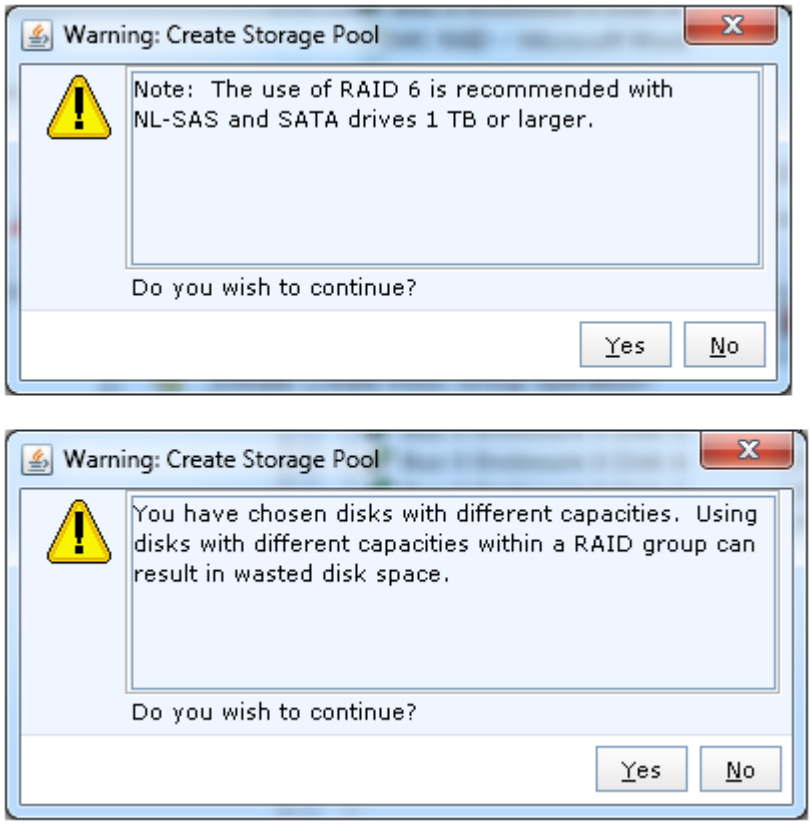
Set up the RAID Group in the Storage Pool, as shown in the following two examples.





Selecting **Apply** brings up a screen requesting confirmation. Select **Yes** to initiate the Create RAID Group operation.

On the next screen, select **Yes** to continue, as shown in the following two examples.

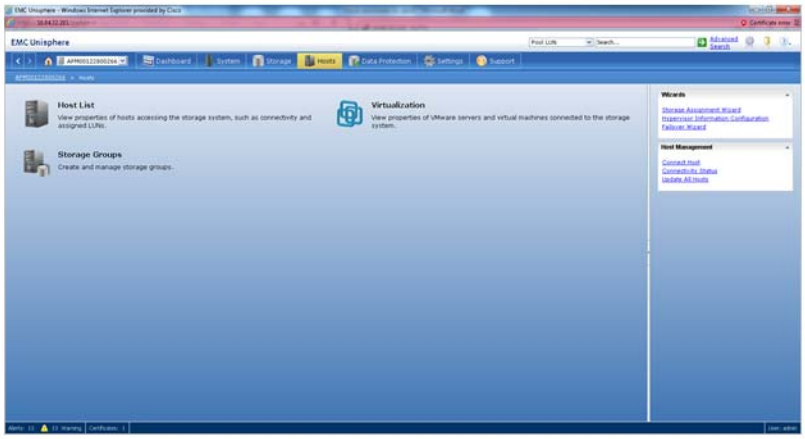


Finally, select **OK** on the next screen to complete the creation process.

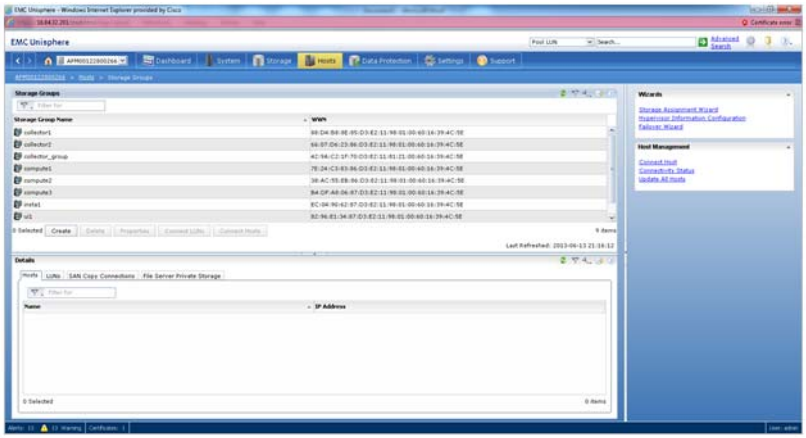
Step 3: Create Storage Groups and Assign Nodes to Them

You will create seven storage groups. Each data node will appear in its own storage group. The Insta Group is separate and contains two nodes.

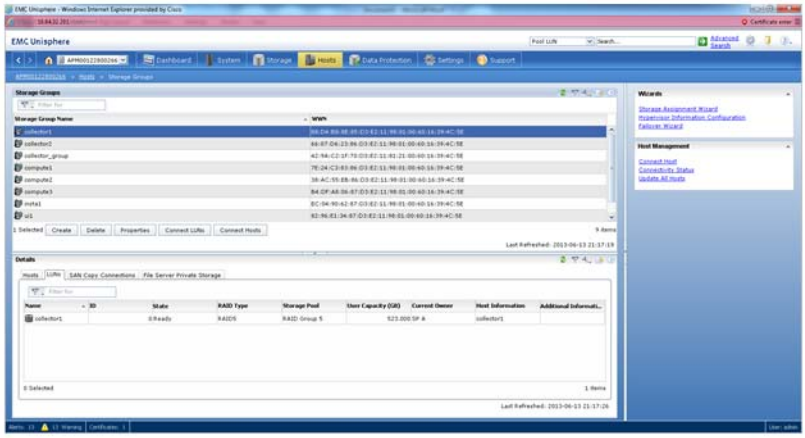
a. Select Host List, as shown on the following screen:



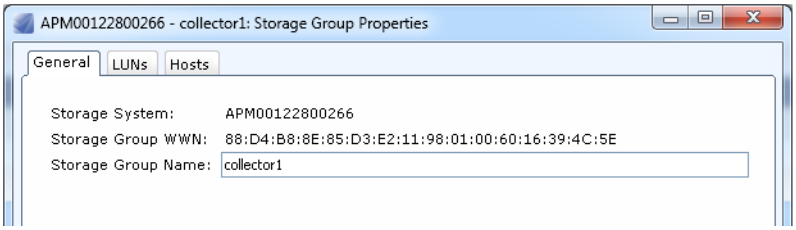
Storage groups are listed in the following screen:



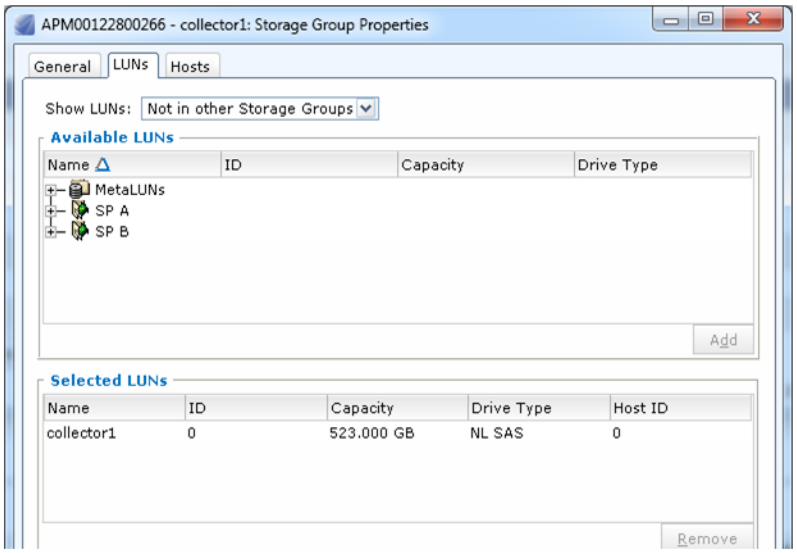
b. Highlight and select a node; in the following example, Collector1 has been selected.



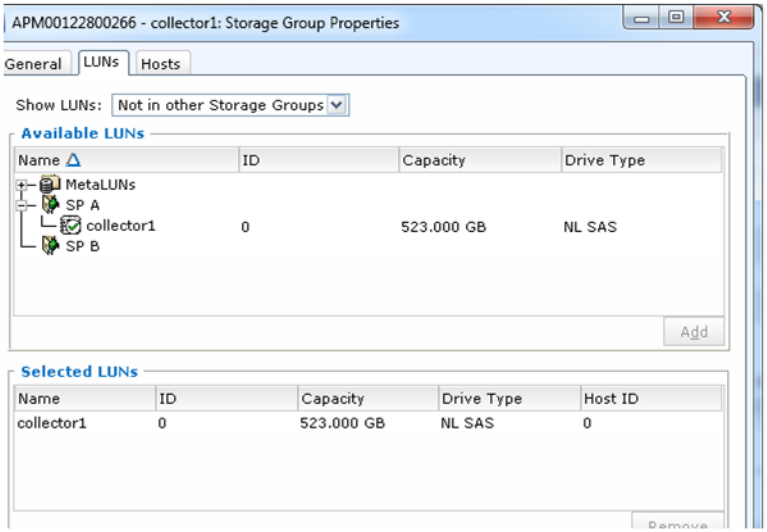
c. Select Properties for Collector1, then select the LUN tab.



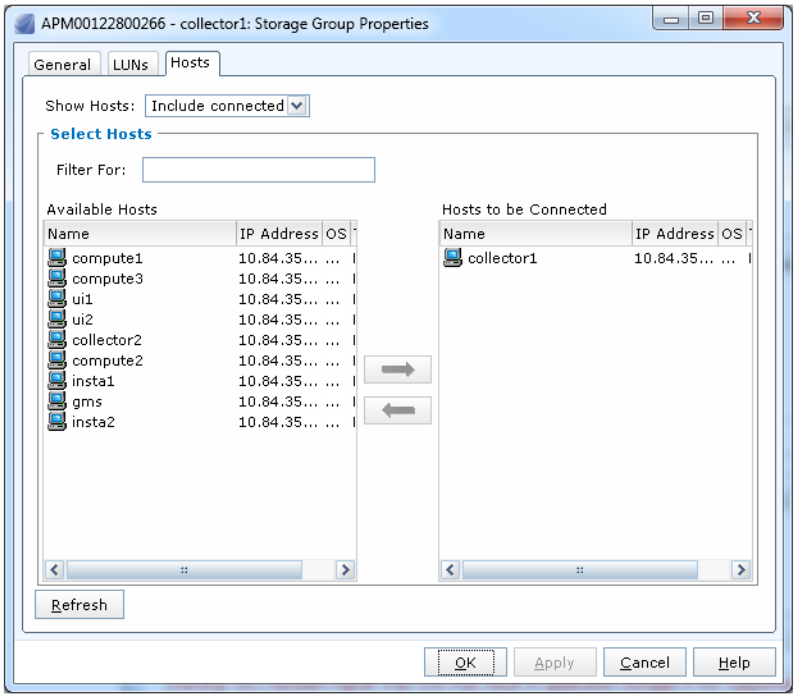
d. On the following screen, expand SP A for the Collector1 Storage group properties.



e. Select Collector 1 in the SP A.

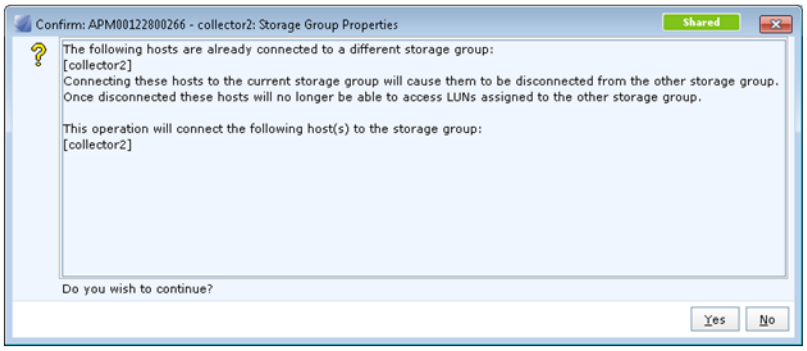


f. Select the Hosts tab.

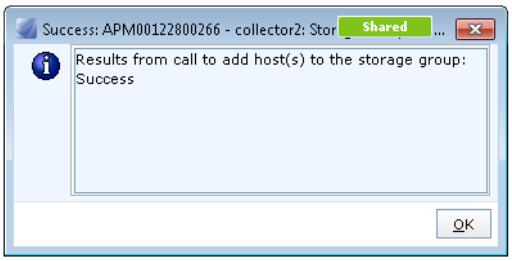


g. In the above screen, move Collector 1 from **Available Hosts** to **Hosts to be Connected**.

The storage group for Collector1 is completed on successful acceptance of actions in the subsequent windows, as in the example for Collector2.

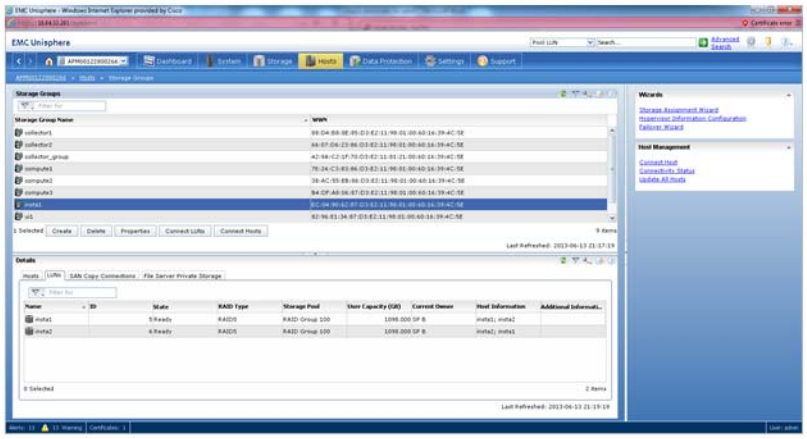


h. Select **Yes**, to continue.

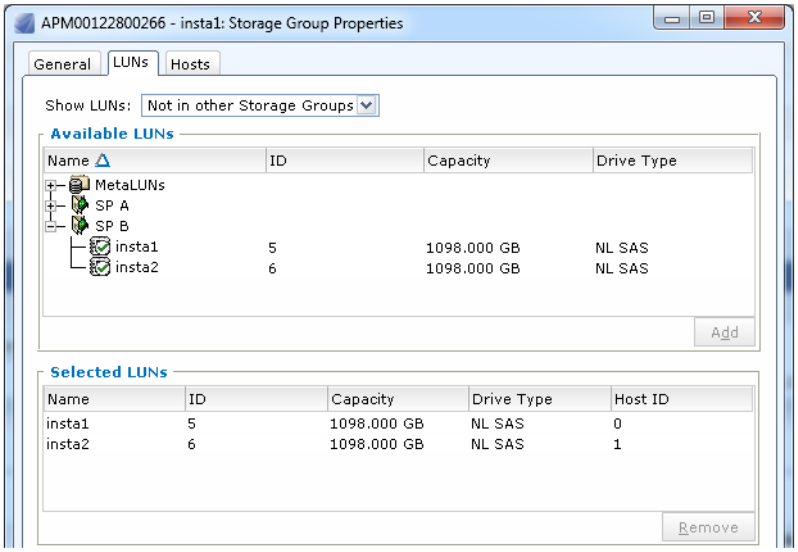


i. Click **OK** to close the window.

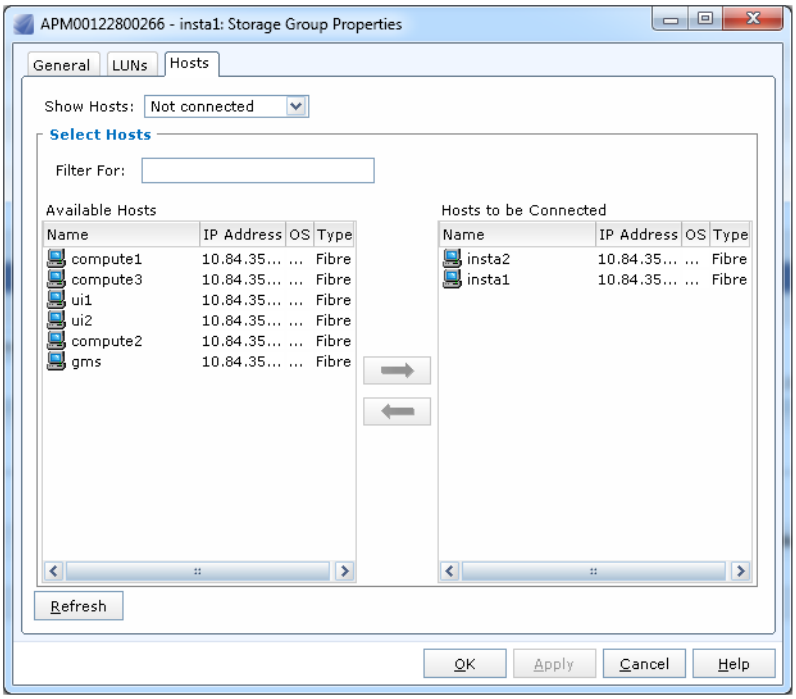
j. Set up storage groups for all other collectors, Compute and UI nodes, as per steps for Collector1. However, you should set up a different storage group for each node. All Insta nodes need to be in the same storage groups. Click one of the Insta nodes to select it, as in the following example for Insta1.



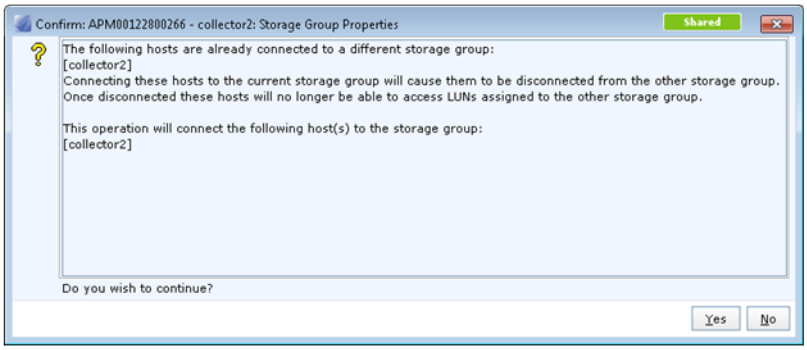
k. Set up the nodes in the LUNs, as shown in this example for Insta1. Note that both Insta 1 and Insta 2 in this example belong to the same storage group.



l. Select all Insta nodes and move them to **Hosts to be Connected**, as in the following example, where Insta1 and Insta2 are made available.

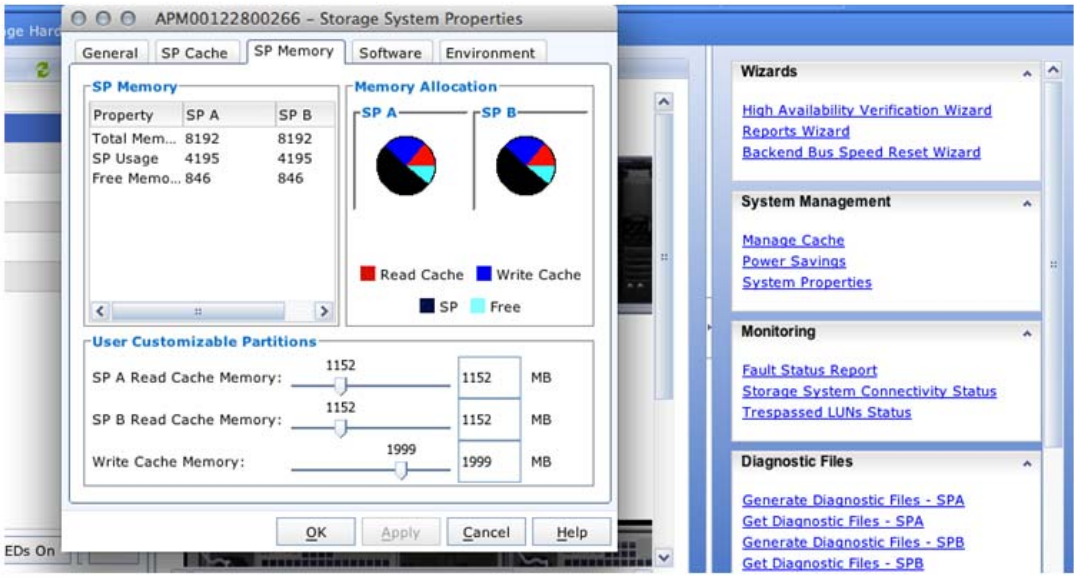


Insta 1 and insta 2 will be set up upon successful acceptance in subsequent windows, as in the following example for setting up Collector2.

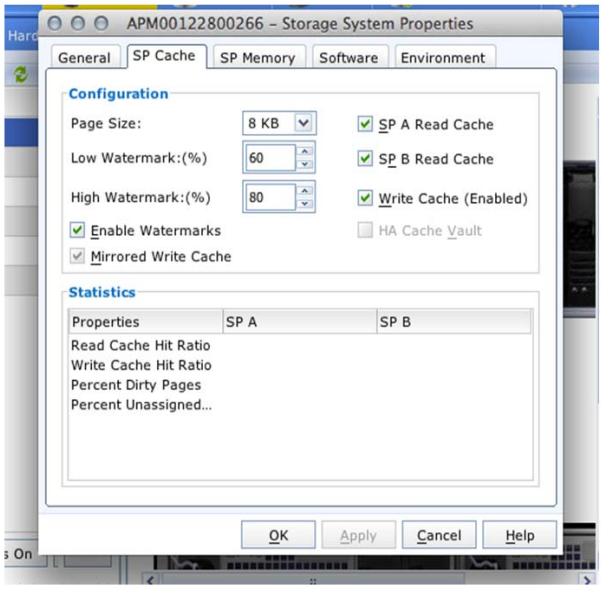


Step 4: Turn on Caching

Select **System Management / Manage Cache**. Disable all caching under **SP Cache**. Then, adjust the SP memory as shown in the example screen below: maximum to write and read set to 1152. Select **Apply** after making individual changes.



After completing the changes, re-enable all cache check boxes under the **SP Cache** tab.



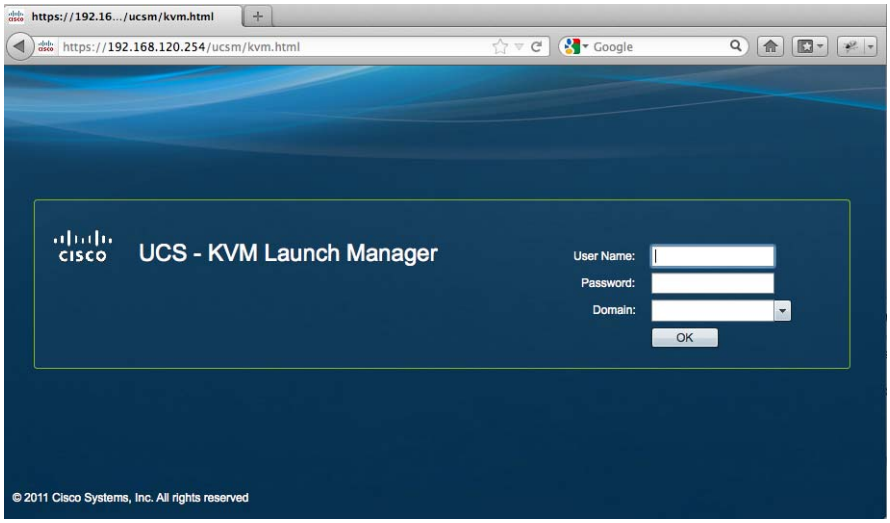
Manufacturing the Blades

Follow these steps to manufacture (install the software release on) all the blades in the setup.

Note At this point, Serial over LAN (SOL) should be configured on all the blades during EMC setup.

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

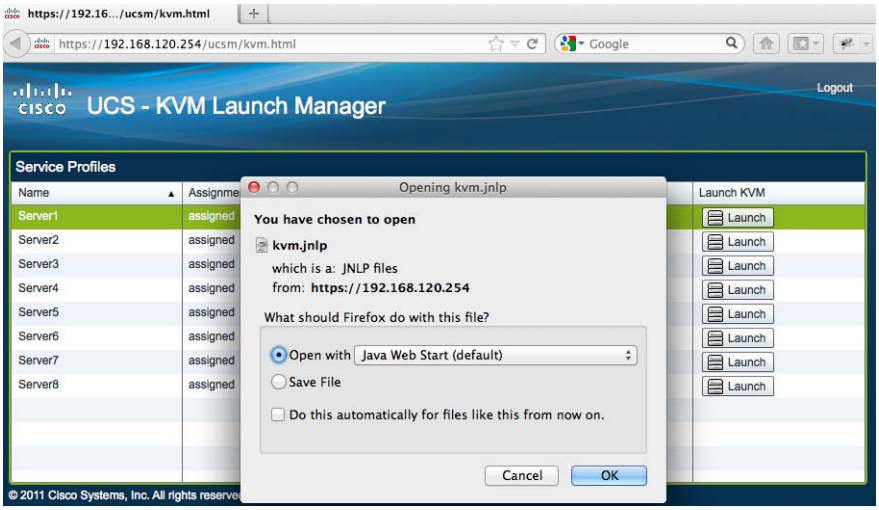
Step 2 Open the Cisco KVM login page in a browser.

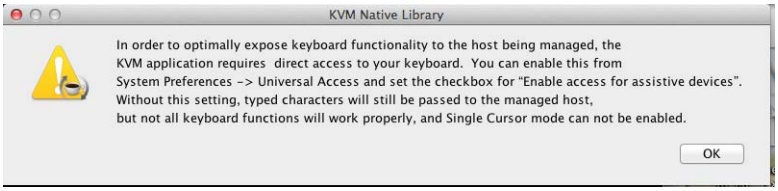


Step 3 Log in to the KVM Manager. You will see all the blades available on the chassis.

The login prompt is presented once the node has successfully completed reboot.

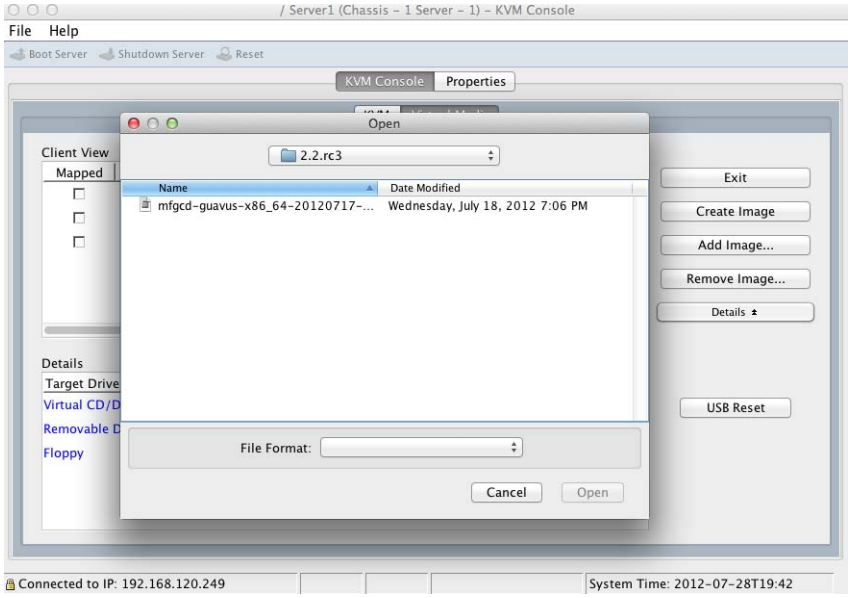
Step 4 Click the **Launch** button for the first node. Click **OK** to download and open a **kvm.jnlp** file. In the keyboard access warning message, click **OK**.



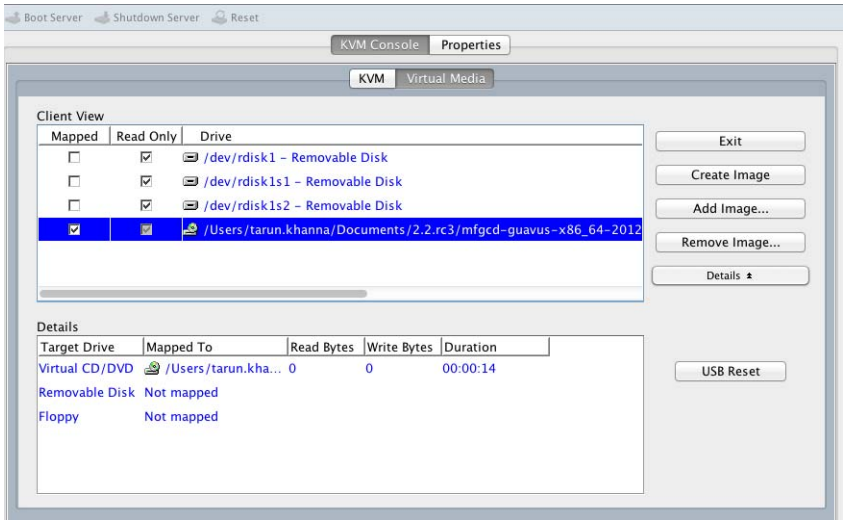


The console for the port opens.

Step 5 Click the **Virtual Media** tab. Click **Add Image** and specify the path of the ISO image that you downloaded in Step 1.

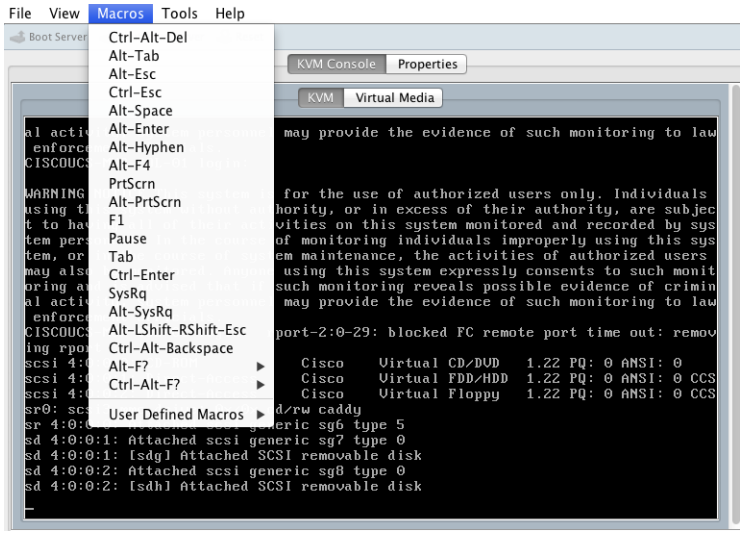


Step 6 Mount the ISO image by clicking the check box next to the added image.

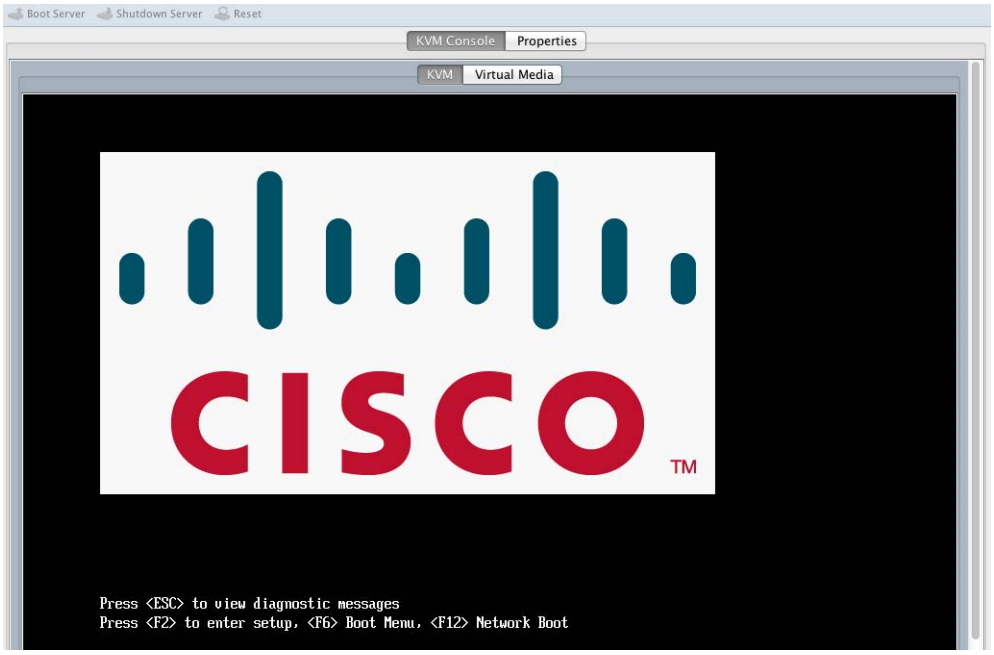


Step 7 Reboot the blade so that it can boot with the mounted image. Click the **KVM** tab and select **Ctrl-Alt-Del** from the Macros drop-down menu.

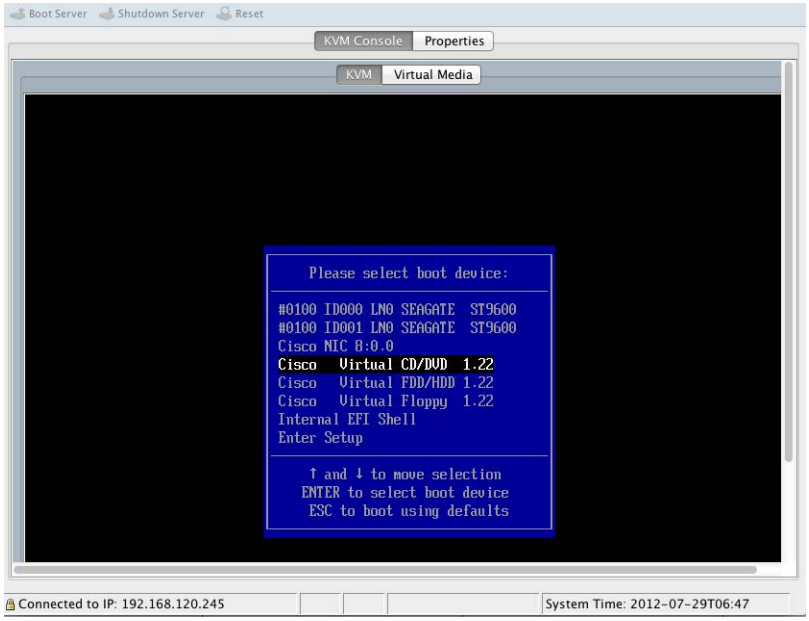
Note Copy the ISO image to multiple directories if you need to manufacture more than one blade at the same time. Select the ISO image from these copied directories for each individual node. Be sure that the same ISO image is not selected for two nodes while manufacturing is in progress for a given node.



Step 8 When the boot order screen appears, press **F6** to select the boot order.



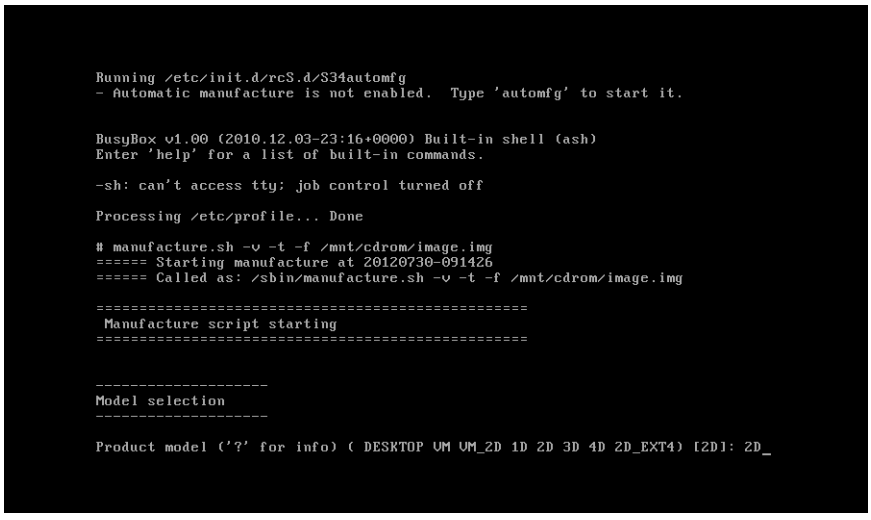
Step 9 Select **Virtual CD/DVD** so the blade boots with the mounted ISO image.

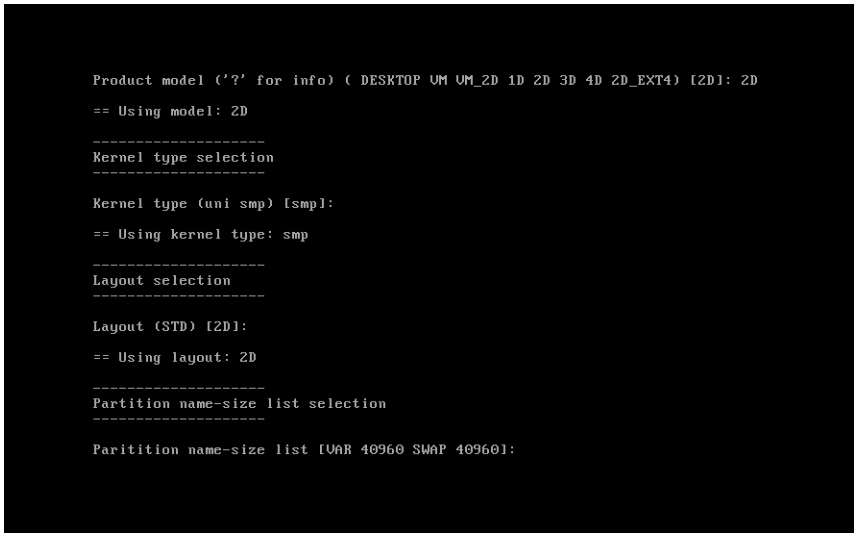


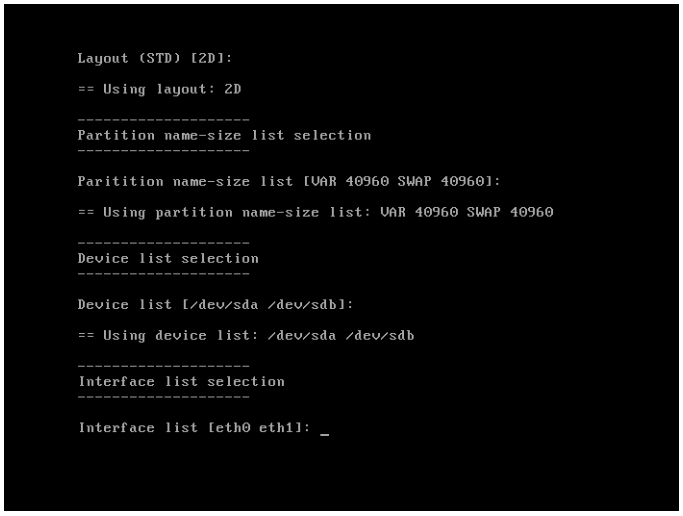
Step 10 At the # prompt, start the manufacture process by entering the manufacture command:

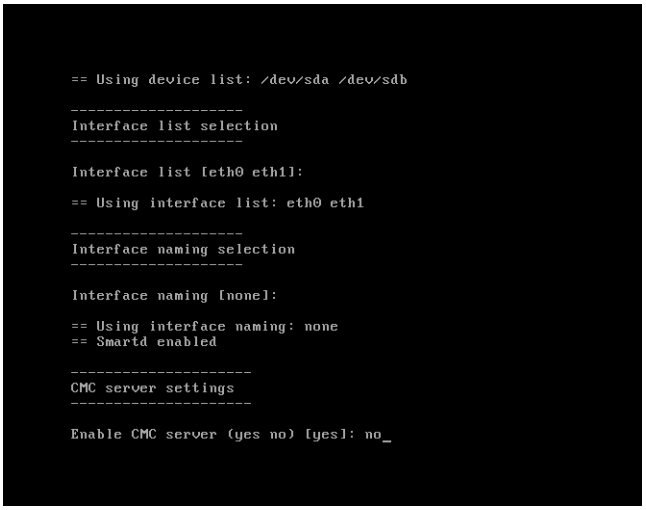
```
# manufacture.sh -v -t -f /mnt/cdrom/image.img --cc no --cs no --cl no -L2D
```

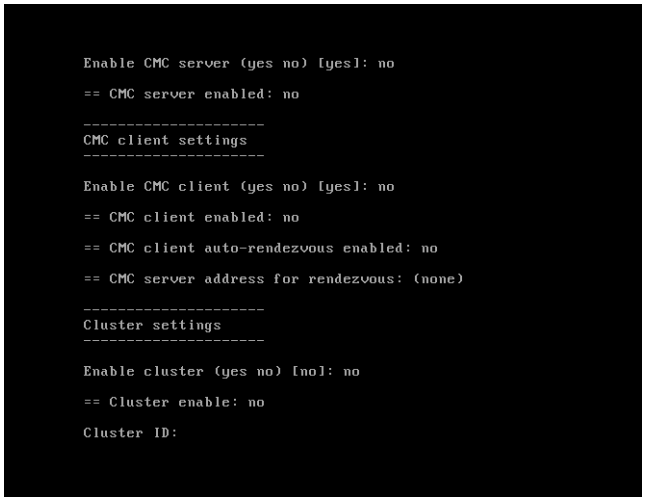
Follow the screens to manufacture the node.

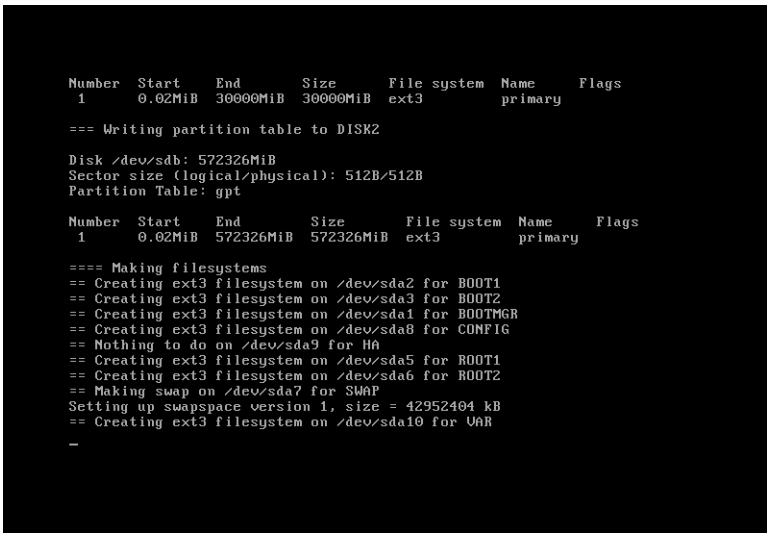


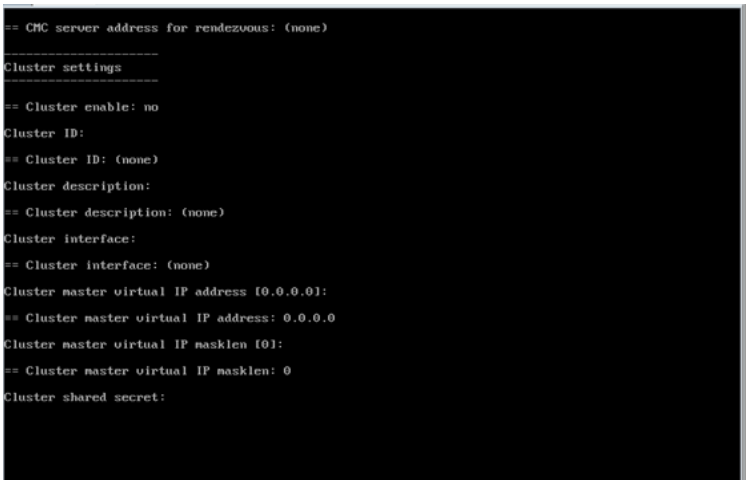

Product model ('?' for info) (DESKTOP VM VM_2D 1D 2D 3D 4D 2D_EXT4) [2D]: 2D
== Using model: 2D
Kernel type selection
Kernel type (uni smp) [smp]:
== Using kernel type: smp
Layout selection
Layout (STD) [2D]:
== Using layout: 2D
Partition name-size list selection
Paritition name-size list [VAR 40960 SWAP 40960]:


Layout (STD) [2D]:
== Using layout: 2D
Partition name-size list selection
Paritition name-size list [VAR 40960 SWAP 40960]:
== Using partition name-size list: VAR 40960 SWAP 40960
Device list selection
Device list [/dev/sda /dev/sdb]:
== Using device list: /dev/sda /dev/sdb
Interface list selection
Interface list [eth0 eth1]: _


== Using device list: /dev/sda /dev/sdb
Interface list selection
Interface list [eth0 eth1]:
== Using interface list: eth0 eth1
Interface naming selection
Interface naming [none]:
== Using interface naming: none
== Smartd enabled
CMC server settings
Enable CMC server (yes no) [yes]: no_


Enable CMC server (yes no) [yes]: no
== CMC server enabled: no
CMC client settings
Enable CMC client (yes no) [yes]: no
== CMC client enabled: no
== CMC client auto-rendezvous enabled: no
== CMC server address for rendezvous: (none)
Cluster settings
Enable cluster (yes no) [no]: no
== Cluster enable: no
Cluster ID:


Number Start End Size File system Name Flags
1 0.02MiB 30000MiB 30000MiB ext3 primary
=== Writing partition table to DISK2
Disk /dev/sdb: 572326MiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Number Start End Size File system Name Flags
1 0.02MiB 572326MiB 572326MiB ext3 primary
==== Making filesystems
== Creating ext3 filesystem on /dev/sda2 for BOOT1
== Creating ext3 filesystem on /dev/sda3 for BOOT2
== Creating ext3 filesystem on /dev/sda1 for BOOTMGR
== Creating ext3 filesystem on /dev/sda8 for CONFIG
== Nothing to do on /dev/sda9 for HA
== Creating ext3 filesystem on /dev/sda5 for ROOT1
== Creating ext3 filesystem on /dev/sda6 for ROOT2
== Making swap on /dev/sda7 for SWAP
Setting up swapspace version 1, size = 42952404 kB
== Creating ext3 filesystem on /dev/sda10 for VAR


== CMC server address for rendezvous: (none)
Cluster settings
== Cluster enable: no
Cluster ID:
== Cluster ID: (none)
Cluster description:
== Cluster description: (none)
Cluster interface:
== Cluster interface: (none)
Cluster master virtual IP address [0.0.0.0]:
== Cluster master virtual IP address: 0.0.0.0
Cluster master virtual IP masklen [0]:
== Cluster master virtual IP masklen: 0
Cluster shared secret:

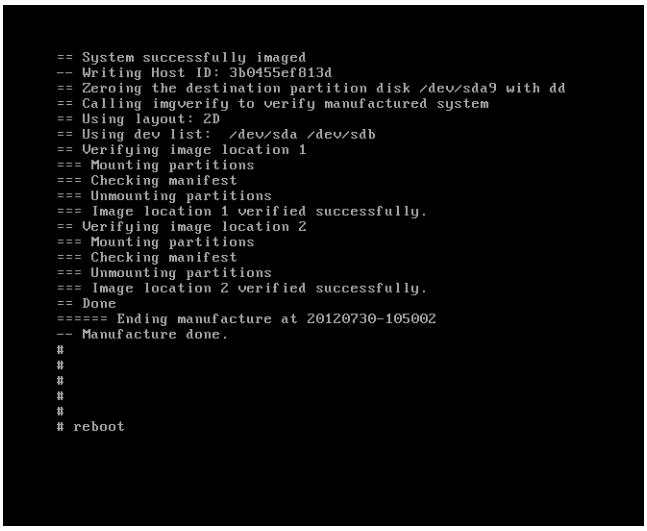
Enter **Return** when each of the following prompts display:

Cluster shared secret:

Cluster expected no. of nodes:

After the manufacture process is completed, return to the **#** prompt by entering **manufacture.sh**.

Step 11 Deselect the ISO image selected in Step 5 of this procedure. Type **reboot** to reboot the node with the new ISO image.



Step 12 Repeat Step 4 through Step 11 for each node.

Setting Up the MURAL Nodes

Step 1 Log in to each node using the Console (KVM Manager).

Step 2 Enter **NO** at the Configuration Wizard:

```
Guavus NetReflex 2.0 configuration wizard
Do you want to use the wizard for initial configuration? no
To return to the wizard from the CLI, enter the "configuration jump-start" command
from the cli.
```

Step 3 Set the password:

```
> en
# conf t
(config) # username admin password admin@123
(config) # write mem
(config) # exit
```

Step 4 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>
(config) # license install LK2-RESTRICTED_CMDS-88A4-FNLG-XCAU-U
(config) # write memory
(config) # _shell
```

Step 5 Reboot the node from configuration mode.

```
> en
# conf t
(config) # reload
```

When all nodes (except GMS nodes) reboot, a login prompt is displayed. Log in to the server, go to a shell. Verify the LUNs exist using **fdisk -l** . Check multipath using **multipath -ll**.

Output of the latter command is similar to the following example (not applicable to the GMS nodes because a LUN is not required for GMS):

```
mpath0 (13600508b1001c2a093828914) dm-2 IET,VIRTUAL-DISK
[size=512M][features=0][hwhandler=0][rw]
\_ round-robin 0 [prio=1][active]
\_ 10:0:0:1 sdc 8:32  [active][ready]
\_ round-robin 0 [prio=1][enabled]
\_ 10:0:0:2 sdd 8:48  [active][ready]
```

An Insta node will have multipaths for all Insta nodes. In the following example with two Insta nodes, one Insta node shows two multipaths follows (multipath output will show names as mpathx, rather than dbroot1 and dbroot2, until GMS is run in later sections of this guide).

```
[admin@guavus-00000f ~]# multipath -ll
dbroot2 (36006016096c03000556d6de218cfe111) dm-1 SGI,RAID 10
[size=7.6T][features=0][hwhandler=0][rw]
\_ round-robin 0 [prio=1][active]
\_ 1:0:0:1 sdd 8:48 [active][ready]
\_ round-robin 0 [prio=1][enabled]
\_ 1:0:1:1 sdf 8:80 [active][ready]
\_ round-robin 0 [prio=1][enabled]
\_ 1:0:2:1 sdh 8:112 [active][ready]
\_ round-robin 0 [prio=1][enabled]
\_ 1:0:3:1 sdj 8:144 [active][ready]
dbroot1 (36006016096c03000546d6de218cfe111) dm-0 SGI,RAID 10
[size=7.6T][features=0][hwhandler=0][rw]
\_ round-robin 0 [prio=1][active]
\_ 1:0:0:0 sdc 8:32 [active][ready]
\_ round-robin 0 [prio=1][enabled]
\_ 1:0:1:0 sde 8:64 [active][ready]
\_ round-robin 0 [prio=1][enabled]
\_ 1:0:2:0 sdg 8:96 [active][ready]
\_ round-robin 0 [prio=1][enabled]
\_ 1:0:3:0 sdi 8:128 [active][ready]
[admin@guavus-00000f ~]#
```

Step 6 Repeat steps 1 through 5 for all other nodes.

Step 7 Download the patches from the FTP server to the GMS server in the /data directory (for example, **atlas3.2.rc3.p1.tgz**). Apply all patches applicable for GMS nodes.

Note See the ***Release Notes for Cisco MURAL Software Version 3.2***, "Downloading and Applying Patches to the GMS Server," for a complete list of patches and installation instructions.

Step 8 Start the GMS Server:

```
> en
# conf t
(config) # pm process gms_server restart
```

Step 9 Check the status of the GMS Server by running the following command:

```
> en
# _shell
# cli -t "en" "config t" "show pm process gms_server" | grep "Current status"
      Current status:  running
```

Configuring the General Management System

This section describes the process of configuring the General Management System (GMS). The GMS is a node that enables a centralized installation, rather than requiring manual installation and configuration of software on each blade.

Step 1 Open the following GMS link in a browser from a machine that can access the GMS: **http://<Mgmt_IP_GMS_SERVER>/applet**.

The initial GMS Configuration screen displays, prompting you to "Load XML."

Step 2 Download the sample file, **mural.xml** to the local machine from which the GMS GUI is being accessed.

Step 3 Enter the path on the local machine for the **mural.xml** file next to the **Load XML** button.

Step 4 Click **Load XML** to load the applet and the configuration into the GMS UI.

Configuration files in GMS can be used in two ways:

- To create a fresh set up using Excel sheets
- To create a set up using an existing data sheet

Step 5 Add, modify, or delete entries on the following tabs as required.

- **Server Details -** The RAC_IP column lists the SOL IP addresses for each node.

  To look up SOL IP for each node, run the following commands on the UCS shell terminal:

  ```
  > scope ip-pool ext-mgmt
  > show detail
  ```

  Example Output:

  ```
  Id: 10.84.35.207
  Subnet: 255.255.255.224
  Def Gw: 10.84.35.193
  Assigned: Yes
  Assigned To: sys/chassis-2/blade-3/mgmt/ipv4-pooled-addr
  Poolable Dn: ip/10.84.35.207/pool-36668
  Prev Assigned To Dn: sys/chassis-2/blade-3/mgmt/ipv4-pooled-addr
  ```

  The exmple above shows SOL IP 10.84.35.207 assigned to a node located on Chassis 2 and node 3.

  Look up the WWID (shown on the EMC Web UI as 'unique ID') of your destination LUN by clicking **LUNs** from the **Storage** tab of the EMC, highlighting the destination LUN, and clicking **Properties**. For all nodes remove the separator ': ' from the unique ID and prefix the complete unique ID with 3, as shown in this example of the WWID to be used in the GMS configuration:

```
360060160c7102f004887f4015d49e211
```

The Collector node, Compute node, and UI node are to have one WWID assigned. However, Insta (being clustered) will have two WWIDs assigned. The GMS configuration will have a WWID for both Insta 1 and Insta 2 assigned to each other. Thus, Insta 1 and Insta 2 will both have 2 WWIDs. Ensure that the same WWID is assigned to **dbroot1** for both insta l and Insta 2 in the GMS configuration. Similarly, the same WWID is to be assigned to **dbroot2** for both Insta l and Insta 2 in the GMS configuration.

- **Networks** - The labels under the **Network_Name** column are defined as follows:
  - **Internal** - The network that the nodes will use to communicate among themselves. It is recommended that a control IP address of the format 192.x.x.x be used for the **eth0** interface.
  - **External** - The management network that will be used to SSH to the blades. It is recommended that a physical or management IP address of the format 10.x.x.x be used for the **eth1** interface. The management IP address can be used for the eth0 interface when an internal network is not available.
  - **Collection** - The network where the Collector feeds will come in so the Collector can process the EDRs
- **Global Settings**
- **Nodes**
- **Clusters**

Note The Collector and Hadoop run on the same system; only the Collector needs to be selected. The **multipath -ll** command on a node can be used to find the WWID. GMS configuration requires the WWID for each node.

a. Click **Validate** and check for configuration errors. If no errors are returned, click **Save**.

b. Enter the full path and file name to be saved on the user's system. Click **Save**.

c. Enter the name of the file to be saved on the GMS server (the recommended file name is **mural.xml**). Click **Save**.

Installing MURAL on the UCS Nodes

Step 1 SSH to the GMS server using the management IP and start the installation on all UCS nodes:

```
> en
# conf t
(config) # install appliance mural.xml all
```

Note The **mural.xml** file is the same file used during GMS Configuration screens.

Step 2 Enter the following command to monitor the installation status on all UCS blades:

```
(config) # gms appliance mural.xml show installation-status all
```

Step 3 Enter the following command to display the running logs for the node while GMS is doing the configuration on the nodes:

```
(config) # gms appliance mural.xml show installation-progress node <node_name>
```

where <node_name> is the name of the node as specified in the GMS screens and the **mural.xml** file.

The above command will show you the running logs for a specific node while GMS is doing the configuration on the nodes. (Press **Ctrl+c** to stop the logs.)

Note The installation process takes approximately one hour to complete, although it may vary by installation site.

The above command shows the percentage of the installation status per blade server. When the installation on all nodes is complete, the following messages are displayed:

```
Collector-GMS-1 : Node successfully installed
Collector-GMS-2 : Node successfully installed
Compute-GMS-1 : Node successfully installed
Compute-GMS-2 : Node successfully installed
CachingCompute-GMS-1 : Node successfully installed
CachingCompute-GMS-2 : Node successfully installed
Rubix-GMS-1 : Node successfully installed
Rubix-GMS-2 : Node successfully installed
```

Troubleshooting Node Installation

GMS can format the storage only if the LUNs have been freshly assigned by EMC. If the LUNs are previously formatted by **mkfs** before running GMS, the installation will fail on that particular node, and can be seen in the logs as shown in the example below.

If you are using previously used LUNs, check the logs for any formatting related errors, as follows.

Step 1 Log in to the GMS server:

```
# cd /data/gms/logs
# cat <NODE_NAME>_cmc.log |  grep "ERROR: Partition does not exist: /dev/mapper/"
```

For example:

```
# cat Collector-GMS-1_cmc.log | grep "ERROR: Partition does not exist: /dev/mapper/"
# cat Collector-GMS-2_cmc.log | grep "ERROR: Partition does not exist: /dev/mapper/"
# cat Compute-GMS-1_cmc.log | grep "ERROR: Partition does not exist: /dev/mapper/"
# cat CachingCompute-1_cmc.log | grep "ERROR: Partition does not exist: /dev/mapper/"
# cat Rubix-GMS-1_cmc.log | grep "ERROR: Partition does not exist: /dev/mapper/"
```

Step 2 Run the following commands on the node on which the above error is found during GMS installation:

```
# dd if=/dev/zero of=/dev/mapper/<partition_name_as_printed_in_Logs> bs=4096
count=262144
# multipath -f <partition_name_as_printed_in_Logs>
# multipath
```

For example:

```
# dd if=/dev/zero of=/dev/mapper/dbroot1 bs=4096 count=262144
# multipath -f dbroot1
# multipath
```

Or

```
# dd if=/dev/zero of=/dev/mapper/mpathd bs=4096 count=262144
# multipath -f mpathd
# multipath
```

Note that /dev/mapper/dbroot1 is an example. Change it to the name that you want to use.

Step 3 Re-run the GMS install command for that particular module:

```
(config t)#install appliance mural.xml cluster cluster-name <module_cluster_name> node
<node_name>
```

For example:

```
install appliance cluster cluster-name Compute-Cluster node Compute-1
```

This command can be run for a complete cluster as follows:

```
install appliance mural.xml cluster cluster-name <module_cluster_name>
```

Note The cluster name and node name appear on the CLI when you click the tab.

Blacklist the Local Disk

Blacklist the local disk (sdb) in multipath.conf on the master Insta node, as described below:

Step 1 Log in to the master Insta node and enter shell mode:

a. mount -o remount,rw /

vi /etc/multipath.conf

The blacklist section should appear as follows:

```
# Blacklist all devices by default. Remove this to enable multipathing
# on the default devices.
blacklist {
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^(sda|sdb)$"
}
```

b. vi /opt/tms/lib/md/templates/multipath.conf

The blacklist section should appear as follows:

```
# Blacklist all devices by default. Remove this to enable multipathing
# on the default devices.
blacklist {
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^(sda|sdb)$"
}
```

Step 2 Modify the same files on the standby Insta node.

Step 3 Reboot both the master Insta node and standby Insta node, one at a time.

Applying Patches

Before beginning the next section of this installation guide, patches and configuration changes must be applied on the nodes. To apply the patches, follow the instructions in the ***Release Notes for Cisco MURAL Software Version 3.2***, in the order shown:

- Copy relevant patches from the GMS to the nodes.
- Apply patches to the master and standby Collector nodes.
- Apply patches to the master and standby Rubix nodes.
- Apply patches to the master and standby Insta nodes.
- Apply configuration changes to the master and standby Insta nodes.

Configuration for the Various Nodes

Before You Begin

The following information is included to facilitate the process of modifying the configuration information on the nodes. We recommend that you read this section before proceeding, and that you refer to it for guidance, as needed.

ASR.X.NAME: Provide the name of the ASR chassis (gateway) that is sending data to the system.

- The same ASR name (case sensitive) must be provided while editing the **gateway.map** IB in the generate and push IB steps of Bulk Stats.
- This is required by the system to correctly annotate the ASR name in processed bulk stats data and display it on the UI node.

ASR.X.EDR_INPUT_DIR: Provide the path where the ASR chassis (gateway) is going to copy the EDR files on the collector.

ASR.X.BULKSTATS_INPUT_DIR: Provide the path where the ASR chassis (gateway) is going to copy the Bulk Stats files on the collector.

The two values described above must be kept different for different ASR chassis (gateways) that are sending data to the collector.

ASR.X.TIMEZONE: Provide the time zone for the incoming filenames sent by the ASR chassis (gateway).

The standard time zone names can be found at the following link:

https://github.com/themattharris/json-dsttime/blob/master/pytz/zoneinfo/zone.tab

It is also available in the following location:

/usr/lib64/python2.6/site-packages/pytz/zoneinfo/zone.tab

Note If you want to configure more ASRs at the time of installation and configuration, you can add another ASR.X.yyyyy block for each ASR and fill in the details, as described above.

COLLECTOR.EDR.FILENAMEFORMAT: Provide the file name format in which the ASR is sending the EDR files.

- The filename pattern provided here should be able to match file names of both incoming http and flow EDR files.
- 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 (_). http EDR files should contain the string "http" delimited by an underscore (_).

Example:

```
NETHERLAND_MURAL-edr_flow_01252013120000.gz
NETHERLAND_MURAL-edr_http_01252013120000.gz
```

The ASR must send the distribution center name as part of the file name. The way to tell the collector which part of the file name holds the distribution center name is to use **%DC** delimited by underscores at that place in the file name format.

For example, if the DC name (ASR chassis name) is **NETHERLAND** and it is entered as follows in the file name NETHERLAND_MURAL-edr_flow_01252013120000.gz, give the file name format as %DC_MURAL-edr_*_%MM%DD%YYYY%hh%mm%ss.gz.

You must provide the same DC name while editing the **dcRegionArea.map** IB in the generate and push IB steps of the EDR. The DC name is case sensitive and it refers to the ASR chassis name.

This is required by the system to correctly annotate the distribution center name in the processed EDR data and display it the on UI node.

If the file names are going to come with **.gz** at the end, it is necessary to provide **.gz** in the file name formt configuration.

COLLECTOR.BULKSTATS.FILENAMEFORMAT: Provide the file name format in which the ASR is sending the BulkStats files.

The BulkStats file name format should be kept as *_<timeAndDateFormat>.

```
*_%MM%DD%YYYY%hh%mm%ss
```

All incoming files should contain the string as per their type in the file name; that is, BulkStats files should contain the string "bulkstats" including an underscore (_) in the incoming file name.

Example:

```
bulkstats_01252013120000
```

Modifying the Configuration on the Nodes

To modify the configuration inputs for the various nodes, log in to the GMS server and modify **input.txt** (configuration inputs) as per the setup details.

```
> en
# conf t
(config) # _shell
# mount -o remount,rw /
```

```
# mkdir -p /data/platform_conf
# cp /opt/var/tps/input.txt /data/platform_conf
# vi /data/platform_conf/input.txt


@collector
################################################################################
#### Please fill in the values as per your network/setup without the quotes ####
################################################################################

SNMP.COMMUNITY=cisco
#SNMP.TRAPRECEIVER=<snmp server ip>
SNMP.TRAPRECEIVER=10.19.2.1
#### Provide the timeout value (in seconds) after which a trap would be generated if no
FLOW/HTTP EDRs were received for that amount of time ####
COLLECTOR.EDR.TIMEOUT=600
# Do not change
# Following are the collector defaults, DO NOT edit unless required.
# Provide in the compression type for the input files fed to the collectors, values can be
either "gzip" or "none"
COLLECTOR.EDR.FILECOMPRESSION=gzip

# Provide in the Drop Alarm Threshold ; if collector data drop exceeds this much
percentage in a 5min interval , then this alarm would be raised
COLLECTOR.EDR.DROPTHRESHOLD=10

# Provide in the Drop Alarm Raise Interval ;
## if collector data drop is greater than configured threshold for N conecutive 5min
intervals, only then this alarm will be raised ; provide value of N
COLLECTOR.EDR.RAISEINTERVAL=2

# Provide in the Drop Alarm Clear Interval ;
## clears the Collector Drop threshold crossed alarm if collector data drop is less than
configured threshold for N consecutive 5min interval ; provide the value of N
COLLECTOR.EDR.DROPCLEARINTERVAL=2

#### Provide the repeat trap timer value (in seconds) after which the trap would be
re-generated if system continues to see no FLOW/HTTP EDRs for that amount of time ####
COLLECTOR.EDR.NODATAALARMREPEAT=900

####CIQ-Provide the input http and flow files name formats ####
## <file-name format of incoming files. Date, time and other keywords must be configured
as below: >
## %YYYY or %YY    - Year as numeric (eg. 2012 or 12)
## %Mmm or %MM     - Month as case-insensitive string (Jan, Feb, Mar, Apr, May, Jun, Jul,
Aug, Sep, Oct, Nov, Dec)
##           or MM as numeric   (01-12)
## %DD             - Day as numeric     (01-31)
## %hh             - Hour as numeric    (00-23)
## %mm             - Minutes as numeric (00-59)
## %ss             - Seconds as numeric (00-59)
## %DC             - Distribution Center Name as string     (max 127 chars) - ASR5K
chassis name
## NOTE: ASR MUST send files with Distribution Center Name string at the place where '%DC'
has been specified in the format here ####
## If the file names are going to come with ".gz" at the end, then it is necessary to
provide ".gz" in the following filename format configuration ####
## Flow :  %DC_<ASR5K rulebase-edrformat name>_%MM%DD%YYYY%hh%mm%ss_*.gz
## Http :  %DC_<ASR5K rulebase-edrformat name>_%MM%DD%YYYY%hh%mm%ss_*.gz
COLLECTOR.EDR.FILENAMEFORMAT=%DC_MURAL-edr_*_%MM%DD%YYYY%hh%mm%ss_*.gz

#### Provide the input http and flow files name formats of the file while it is being
copied ; after copy is complete ASR must rename the file
 to the above format ####
```

```
COLLECTOR.EDR.TRANSFERFILENAME=%DC_MURAL-edr-*_%MM%DD%YYYY%hh%mm%ss_*.gz.tmp

#### Provide the input bulkstats file name formats of the file while it is being copied
####
COLLECTOR.BULKSTATS.FILENAMEFORMAT=*_%YYYY%MM%DD%hh%mm%ss

###################### Provide the information about ASR gateways below
###############################
## ASR.x.NAME=<ASR name>
## ASR.x.EDR_INPUT_DIR=<Local directory on collector where ASR will drop EDR record files>
## ASR.x.BULKSTATS_INPUT_DIR=<Local directory on collector where ASR will drop bulkstats
record files>
## ASR.x.TIMEZONE=<Timezone where ASR is physically located>
## Note  : The tzselect linux utility comes handy while determining the timezone.
#################################################################################################
#############
ASR.1.NAME=GMPLAB1
ASR.1.EDR_INPUT_DIR=/data/collector/edr1
ASR.1.BULKSTATS_INPUT_DIR=/data/collector/bs/GMPLAB1/
ASR.1.TIMEZONE=America/Jamaica

ASR.2.NAME=GMPLAB2
ASR.2.EDR_INPUT_DIR=/data/collector/edr2
ASR.2.BULKSTATS_INPUT_DIR=/data/collector/bs/GMPLAB2/
ASR.2.TIMEZONE=Asia/Kolkata

#############################################################################
## Protocol used for transferring Uncategorized URL/UA/TAC Reports for Offline
Categorization
#############################################################################
COLLECTOR.UNCAT_REPORTS.XFER_PROTOCOL=SFTP

#################### JOBS Config ################################
## JOBS.TETHERING_TRANSFER_PROTOCOL=<Transfer protocol used to send tethering databases to
ASR Gateway>
JOBS.TETHERING_TRANSFER_PROTOCOL=SFTP

## JOBS.TETHERING_THRESHOLD=<Threshold for deciding whether a conflicting UAs/OS-Sign
(appear for both Smart and non-Smart device groups) need to be considered for addition to
tethering database. This represents the difference in % between smart and non-smart
flows.> # (Between 0 - 100) ##
JOBS.TETHERING_THRESHOLD=50

## JOBS.TETHERING_CONFIDENCE_LEVEL=<Tethering Confidence Level is used for filtering the
conflicting UAs/OS-Signs which have crossed the tethering threshold. Essentially keeps a
check on how many UAs/OS-Signs can be added to the tethering database.> ##
JOBS.TETHERING_CONFIDENCE_LEVEL=50
#############################################################


@insta
#########################################################################
#This is a file of CONSTANTS/variables for an Insta file. Base file does not have any
extension. It is assumed that the base file and var file
 are in same folder.
#Please note, only unique values are allowed in config file . Duplicate key
#### Provide a name for the database that would be created to persist the processed data
so that it can be rendered on the UI ####
EDR_DB_NAME: 'ucsdb_UI_31rc4'
#### Provide a name for the database that would be created to persist the processed data
so that it can be rendered on the UI ####
BULKSTATS_DB_NAME:'bulkstats_UI_31rc4'
#### Provide the SNMP location be set which be returned in SNMP get/walk query ####
SETUP_LOCATION_INFO: 'Gurgaon'
```

```
### Provide the SNMP community name to be set which be used in SNMP get/walk queries ####
SNMP_COMMUNITY_NAME: 'cisco'
#### Provide the IP of the server where the SNMP traps are to be sent for monitoring
health of the system ####
SNMP_TRAP_RECEIVER_IP: '192.168.124.93'
#######################################################################


@ui
#############################################################################################
#### This is a file of CONSTANTS/variables for an UI file. Base file does not have any
extension. ####
#### It is assumed that the base file and var file are in same folder.####
#############################################################################################

###### Provide the TIME ZONE for UI Nodes ####
TIME_ZONE:'GMT'

###### Provide the COUNT of TOMCAT SERVERS for EDR, the value needs to 2 for this script
####
TOMCAT_SERVER_COUNT: '2'

###### Provide the memory in Gigabytes which will be assigned to each EDR tomcat in the
script ####
EDR_RAM: '90'

###### Provide the memory in Gigabytes which will be assigned to BulkStats tomcat in the
script ####
BulkStats_RAM: '80'

###### Provide the memory in Gigabytes which will be assigned to RGE tomcat in the script
####
RGE_RAM: '10'

###### Provide the DOMAIN NAME for the Script ####
DOMAIN_NAME: 'mural.com'

###### Provide the complete URL NAME for the Script #####
URL_NAME: 'ucsd.mural.com'

###### Provide the SMTP SERVER  ADDRESS ####
SMTP_SERVER_ADDRESS: 'mx1.guavus.com'

###### Provide the IP ADDRESS OF SMTP SERVER  ####
SMTP_SERVER_IP: '10.10.10.10'

###### Provide the PORT Number which is used for SMTP SERVER COMMUNICATION ####
SMTP_SERVER_PORT: '25'

###### Provide the RECIPIENT EMAIL ADDRESS ####
RECIPIENT_EMAIL: 'qa@guavus.com'

###### Provide the SENDER EMAIL ADDRESS ####
SENDER_EMAIL: 'admin@guavus.com'

##### Provide the IP Address of SNMP Reciever #####
SNMP_TRAP_RECEIVER_IP: '192.168.153.220'

##### Provide the Setup Location ####
SETUP_LOCATION_INFO: 'gurgaon'

###### Provide the SNMP community string used for sending SNMP traps ####
SNMP_COMMUNITY_NAME: 'cisco'
```

Note You can configure the different nodes seperately, as described in the next three sections, or you can configure multiple nodes at one time.

Configuring the Collector and Compute Nodes

Log in to the GMS server and apply the configuration for the Collector and Compute nodes:

```
> en
# _shell
# pmx
pm extension> subshell platform
pm extension (platform)> set xml file mural.xml
pm extension (platform)> generate config for collector
pm extension (platform)> apply config for collector
```

The above command, **apply config for collector** opens an SSH connection to the Collector and Compute nodes and configures them.

Wait for the **pm extension (platform)>** prompt which indicates the Collector and Compute nodes have been successfully configured.

Note No prompt is returned when the Collector and Compute nodes have been successfully configured, but if the configuration of either node fails, an error is returned.

Configuring the Insta Nodes

Log in to the GMS server and apply the configuration for the Insta node:

```
> en
# _shell
# pmx
pm extension> subshell platform
pm extension (platform)> set xml file mural.xml
pm extension (platform)> generate config for insta
pm extension (platform)> apply config for insta
```

The above command, **`apply config for insta`** opens an SSH connection to both Insta nodes and configures them.

Wait for the message, "Caching Compute (Insta) module installation is complete; please proceed to the next step in the IOG." The message indicates that installation for the Insta nodes is complete.

Note Configuring of the Insta nodes may take up to one hour to complete.

Configuring the UI Nodes

Log in to the GMS server and apply the configuration for the UI nodes:

```
> en
# _shell
```

```
# pmx
pm extension> subshell platform
pm extension (platform)> set xml file mural.xml
pm extension (platform)> generate config for ui
pm extension (platform)> apply config for ui
```

The above command "apply config for ui" will prompt with a message asking for a fresh INSTALL or to UPGRADE the existing UI nodes. Type INSTALL, as this is a fresh installation.

```
pm extension (platform)> apply config for ui
JOB ID = JOB20130331_112653477898
/data/platform_conf/ui.conf
/usr/gap/Scripts/batchUpdate/Cisco2_UI/config.cfg
```

Type **Install** if it is a manufactured UI node (INSTALL). Otherwise, type **Upgrade** if you are upgrading from a previous release (UPGRADE)? INSTALL

Wait for the following message indicating that the installation for the Rubix nodes is complete:

"Script for Configuring Distributed Rubix is COMPLETE. PLEASE MOVE TO NEXT STEP IN IOG and start all the TOMCATS"

Generate and Push the Information Bases

Tethering

The Tethering feature generates an information base of devices and UA/OS signatures that aids the ASR 5000 in detecting tethered traffic.

The following is the list of valid device groups returned from the traffic cat engine, depending on its Type Allocation Code (TAC).

- 3G Watch
- Feature Phone
- M2M
- Mobile Router
- Netbook
- PC Card
- SmartPhone
- Tablet
- UNKNOWN

Manual Modifications

In addition to the necessary configurations done by the scripts, some manual modifications may be required to support tethering. These are described below.

Step 1 Go to the CLI configure terminal of master Collector node and run the following command to add any new smart device groups. Choose from the list of valid device groups mentioned in the previous section to add them to the smart device group list, as specified below:

```
> en
# _shell
# cli -m config
(config) # pmx
pm extension> subshell aggregation_center
pm extension (aggregation center)> show ib smartDG.list

pm extension (aggregation center)>
pm extension (aggregation center)> edit ib smartDG.list add
Device Group: SmartPhone
pm extension (aggregation center)> edit ib smartDG.list add
Device Group: Feature Phone
pm extension (aggregation center)> show ib smartDG.list
     1 SmartPhone
     2 Feature Phone
pm extension (aggregation center)>
```

Step 2 Create a file with details of the ASR 5000 gateways, where TAC, OS, or UA databases need to be pushed.

```
File : /data/work/serverFile_tethering
```

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

<<IP Address of Gateway>>, <<username for logging into ASR5K Gateway>>, <<password for logging into ASR5K Gateway>>, <<Location on the ASR5K Gateway machine where databases need to be copied>>

Log in to the master Collector node:

```
> en
# _shell
# cd /data
# mkdir work
# cd work
# vi /data/serverFile_tethering
192.168.156.96, admin, admin, /data/TTYT
```

Example:

```
192.168.1.1, admin, password, /data
```

Step 3 Create the same file on the standby Collector node as well.

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

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

This file can have multiple such rows.

Event Data Record (EDR)

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 Use the above table for example purposes only. You should use the data that matches your environment. For example, for GGSN, you might use GGSN, PGW, or HA. In this case, GGSNIP is the management IP address. For SGSN, you might use SGSN, SGW, HSGW, or PDSN. In this case, SSGNIP is the service IP address.

Step 1 To generate the EDR information bases in the correct database on the Insta node, change the **sm config** file on the master Collector node.

Step 2 Go to the shell and then run the CLI, as shown:

```
# _shell
# cli -m config
(config) # sm service-info modify ps-server-1 port 11111
(config) # write memory
(config) # _shell
```

Verify port change:

```
cli -t "en" "config t" "show running-config full" | grep "ps-server-1 port"
sm service-info modify ps-server-1 port 11111
```

Step 3 Go to the CLI configure terminal of the master Collector node, and run the following commands:

```
# en
> conf t
pmx
pm extension> subshell aggregation_center
pm extension (aggregation center)> fetch all ibs from image
pm extension (aggregation center)> add ib_destination <IP
address>
```

Add Control or Internal IP addresses of the master and standby Collector and UI nodes.

Step 4 Enter Region, Area and DC name. For example, Region = US, Area = NE, and DC = GMPLAB1 (chassis name). DC name must be the same name you provided in the collector configuration output directory path name:

"/data/collector/1/output/edrflow/%y/%m/%d/%h/%mi/Cisco.EDRFLOW"

Cisco is the name used in the above example.

```
pm extension (aggregation center)> edit ib dcRegionArea.map add
```

Step 5 Enter the GGSN IP and GGSN name:

```
pm extension (aggregation center)> edit ib ipGgsn.map add
```

Step 6 Enter the SGSN IP and SGSN name:

```
pm extension (aggregation center)> edit ib ipSgsn.map add
```

Step 7 Enter the APN name and APN group:

```
pm extension (aggregation center)> edit ib apnGroup.map add
```

Step 8 Enter the RAT ID and RAT TYPE:

```
pm extension (aggregation center)> edit ib ratidtype.map add
```

Step 9 Change the default segments to have only **Low**, **Regular**, and **High** segments in the configuration, as shown in the following example:

```
pm extension> subshell aggregation_center
pm extension (aggregation center)> show ib segment.map
     1  [629145600][Extreme]
     2  [314572800][High]
     3  [157286400][Active]
     4  [62914560] [Regular]
     5  [31457280] [Mild]
     6  [0]        [Low]
pm extension (aggregation center)>
pm extension (aggregation center)> edit ib segment.map delete record 5
IB [segment.map] updated.
pm extension (aggregation center)> show ib segment.map
     1  [629145600][Extreme]
     2  [314572800][High]
     3  [157286400][Active]
     4  [62914560] [Regular]
     5  [0]        [Low]
pm extension (aggregation center)> edit ib segment.map delete record 3
IB [segment.map] updated.
pm extension (aggregation center)> show ib segment.map
     1  [629145600][Extreme]
     2  [314572800][High]
     3  [62914560] [Regular]
     4  [0]        [Low]
pm extension (aggregation center)> edit ib segment.map delete record 1
IB [segment.map] updated.
pm extension (aggregation center)> show ib segment.map
     1  [314572800][High]
     2  [62914560] [Regular]
     3  [0]        [Low]

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

Step 10 After the above push command completes, execute the following command on the standby Collector node from the CLI configure terminal:

```
pmx
pm extension> subshell aggregation_center
pm extension (aggregation center)> fetch all ibs from inbox
Copying IB mobileappname.list             [OK]
Copying IB segment.map                    [OK]
Copying IB apnGroup.map                   [OK]
Copying IB segment.txt                    [OK]
Copying IB dcNameClli.map                 [OK]
Copying IB ReservedNames.json             [Failed]
Copying IB collection_center.list         [OK]
Copying IB ratidtype.map                  [OK]
```

```
Copying IB wngib.id.map                    [OK]
Copying IB pcsarange.list                  [OK]
Copying IB dcRegionArea.map                [OK]
Copying IB mobileappcategory.list          [OK]
Copying IB model.list                      [OK]
Copying IB apnNetworkIB.map                [OK]
Copying IB UnresolvedTraps                 [Failed]
Copying IB ipGgsn.map                      [OK]
Copying IB keyword.list                    [OK]
Copying IB sp.list                         [OK]
Copying IB url.list                        [OK]
Copying IB ipSgsn.map                      [OK]
Copying IB mobilenhappcategory.list        [OK]
Copying IB SNMPRecordConverterConfig       [Failed]
Copying IB mobileapptype.list              [OK]
Copying IB ServiceGateway.list             [Failed]
Copying IB mime.list                       [OK]
Copying IB topsubscriberib.id.map          [OK]
Copying IB blacklist                       [Failed]
Copying IB mobilenhappname.list            [OK]
Copying IB subdevice.id.map                [OK]
Copying IB subseg.id.map                   [OK]
Copying IB manufacturer.list               [OK]
Copying IB IBStore.tab                     [OK]
Copying IB category.list                   [OK]
Copying IB subscriberib.id.map             [OK]
pm extension (aggregation center)>
pm extension (aggregation center)> quit
pm extension> quit
COL-02 [COL-VIP: standby] (config) # write mem
COL-02 [COL-VIP: standby] (config) #
```

Initial Configuration of BulkStats

Step 1 To generate the bulk stats IB in the correct database on insta, change the **sm config** file on the master Collector node. Go to the shell, then run the CLI, as shown:

```
# _shell
# cli -m config
(config) # sm service-info modify ps-server-1 port 22222
(config) # write memory
(config) # _shell
```

Verify the port change:

```
# cli -t "en" "config t" "show running-config full" | grep "ps-server-1 port" sm
service-info modify ps-server-1 port 22222

# cli -m config

coll-100-10 [coll-100-73: master] (config) # pmx
pm extension> subshell bulkstats
pm extension (bulk stats)> fetch all ibs from image
Copying IB gatewayIDVersion.map            [OK]
Copying IB key.map                         [OK]
Copying IB kpi.threshold.ib                [OK]
Copying IB gateway.map                     [OK]
Copying IB schema.map                      [OK]
Copying IB subschema_definition.list       [OK]
Copying IB metric.map                      [OK]
```

```
Copying IB cube_names.list                [OK]
pm extension (bulk stats)> edit ib gateway.map  add
Gateway: ASR5K
Version: 14
DC: Gurgaon
Region: SEZ
Area: ITPARKS
```

Step 2 Provide the gateway name (ASR chassis name) here. Use the same name provided in the **input.txt** file during the initial Collector configuration. This name will be displayed on the UI.

```
pm extension (bulk stats)> show ib gateway.map
     1  [ASR5K][14][Gurgaon][SEZ][ITPARKS]
pm extension (bulk stats)>
pm extension (bulk stats)> generate all ibs
[key.map]:
    generated key.id.map
[gateway.map]:
    generated gateway.id
    generated version.id
    generated dc.id
    generated region.id
    generated area.id
    generated gatewayRegionArea.id.map
[schema.map]:
    generated schema.id.map
[metric.map]:
    generated metric.id.map
[gatewayIDVersion.map]:
Summary:
========
Successful IBs : 5 out of 5
Failed IBs : No id generation failures.

pm extension (bulk stats)>push all ibs
```

On standby:

```
coll-bk-100-207 [coll-100-73: standby] (config) # pmx
pm extension> subshell  bulkstats
pm extension (bulk stats)> fetch all ibs  from inbox
Copying IB gatewayIDVersion.map           [OK]
Copying IB key.map                        [OK]
Copying IB kpi.threshold.ib               [OK]
Copying IB gateway.map                    [OK]
Copying IB schema.map                     [OK]
Copying IB subschema_definition.list      [OK]
Copying IB metric.map                     [OK]
Copying IB cube_names.list                [OK]
Copying IB key.id.map                     [OK]
Copying IB gateway.id                     [OK]
Copying IB version.id                     [OK]
Copying IB dc.id                          [OK]
Copying IB region.id                      [OK]
Copying IB area.id                        [OK]
Copying IB gatewayRegionArea.id.map       [OK]
Copying IB schema.id.map                  [OK]
Copying IB metric.id.map                  [OK]
pm extension (bulk stats)>
```

Step 3 Revert the port to the default on the master Collector. Once the bulk stats IB generate and push operations are done, change the port value back to 11111. Exit the current CLI shell, log in again to the CLI, and set the port to 11111.

```
(config) # _shell
#cli -m config
(config) # sm service-info modify ps-server1 port 11111
(config) # write memory
```

Uncategorized URL, UA, and TAC Reports

Step 1 Create a file to contain the destination information for the uncategorized URL, UA, and TAC reports, as shown:

File : /data/work/serverFile_uncatReports

"serverFile_uncatReports" contains the entry for data transfer destination. This file has the following format:

<<IP>>, <<username>>, <<password>>, <<location where db's need to be copied>>

Example:

```
192.168.156.96, admin, password, /data/offline_uncat_reports
```

Note IP address is the external IP.

Step 2 Log in to the master Collector node:

```
> en
# _shell
# cd /data# mkdir work
# cd work
# vi /data/work/serverFile_uncatReports192.168.156.96, admin, password, /data/TTYT
```

Note IP address is the external IP.

Step 3 Create the same file on the standby Collector node.

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

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

Single Certificate Installation to Access EDR, BulkStats and RGE

This section provides instructions for generating and installing a common certificate for all Tomcat processes on the distributed Rubix setup. The example shown is based on generating and using a certificate with the Microsoft CA. However, when deploying the MURAL application, you can use any third party CA (Verisign, for example).

Backing Up and Generating the Keystore Files

Step 1 Log in to the master Rubix node and make a backup of the original keystore.

```
mv /data/apache-tomcat/apache-tomcat-7.0.27/keystore
/data/apache-tomcat/apache-tomcat-7.0.27/keystore_orig
```

Step 2 Generate the new keystore file following the steps below:

```
cd /usr/java/latest/bin/
./keytool -keysize 2048 -genkey -alias tomcat -keyalg RSA -keystore keystore
```

Enter **password** -- rubix123

first and last name -- The fully-qualified domain name, or URL, you are securing. If you are requesting a wildcard certificate, add an asterisk (*) to the left of the common name where you want the wildcard; for example, ***.cisco.com.**

Organizational Unit -- Optional

Organization -- The full legal name of your organization (for example, Cisco)

City/Locality -- Name of the city in which your organization is registered/located — do not abbreviate

State/Province -- Name of state or province where your organization is located — do not abbreviate

Country Code -- The two-letter International Organization for Standardization (ISO) format country code for where your organization is legally registered.

```
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  *.cisco.com
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:  Cisco
What is the name of your City or Locality?
  [Unknown]: Tewksbury
What is the name of your State or Province?
  [Unknown]: Massachusetts
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=*.cisco.com, OU=Unknown, O=Guavus, L=Gurgaon, ST=Haryana, C=IN correct?
  [no]:  yes
Enter key password for <tomcat>
    (RETURN if same as keystore password):
[admin@CISCOMUR-UI-147-17 bin]#
```

Step 3 Enter the following command into keytool to create a CSR:

```
[admin@CISCOMUR-UI-147-17 bin]# ./keytool -certreq -keyalg RSA -alias tomcat -file
csr.csr -keystore keystore
```

Step 4 Enter keystore password.

Note This is the same password provided above.

A csr.csr file is generated.

Step 5 Open the CSR file, and copy all of the text

```
[admin@CISCOMUR-UI-147-17 bin]# vi csr.csr
```

Step 6 Paste all of the text into the online request form as mentioned in the steps below and complete your application.

Downloading the Signed Certificate

Step 1 Log in to **https://mx1.guavus.com/certsrv**.

Step 2 Enter your username and password.

Step 3 From the list of tasks, select **Request a certificate**.

Step 4 Select **Advanced Certificate Request** from the next screen.

Step 5 On the next screen, **Submit a Certificate Request or Renewal Request**, enter the CSR generated above (in the **csr.csr** file) and select **Web Server** from the Certificate Template.

Step 6 Download the CA certificate using the link - Download certificate, provided on submitting of the CSR.

The certificate you requested is issued.

Step 7 Rename the downloaded file to **tomcat.cer**.

Step 8 Place the file **tomcat.cer** on the same path on the server where keystore is generated: **/usr/java/latest/bin/**

```
scp tomcat.cer admin@63.118.245.63:/usr/java/latest/bin
```

Downloading the CA Certificate

Step 1 Log in to **https://mx1.guavus.com/certsrv**.

Step 2 Enter your username and password.

Step 3 From the list of tasks, select the task **Download a CA certificate, certificate chain, or CRL**

Step 4 On the next screen, select **Download CA certificate**.

Step 5 Rename the downloaded file to **root.cer** and place the file on the same path on the server where the keystore is generated: /usr/java/latest/bin/

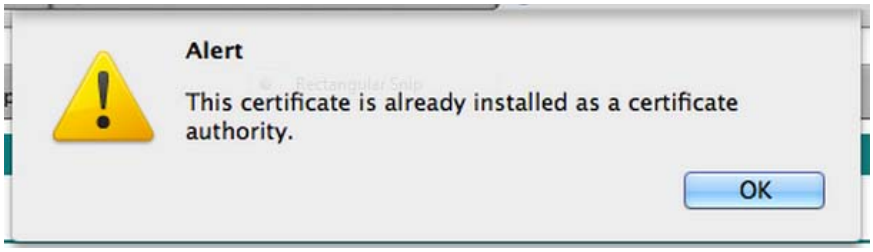
```
scp root.cer admin@63.118.245.63:/usr/java/latest/bin
```

Step 6 Click on the **Install CA certificate** link.

Note This option should be used on all client machines to install the CA certificate on the client, which will be able to identify the certificates signed using GUAVUS-MX1-CA.

Step 7 Click all the check boxes and then click OK.

Note If the CA certificate is already installed, the following message is displayed:



Installing the Signed Certificate in Keystore

For the CA GUAVUS-MX1-CA, the root (CA) certificate and signed certificates (downloaded using the steps given above) need to be added to keystore.

Step 1 Using keytool, enter the following commands to install the certificates.

Step 2 Install the Root certificate.

Step 3 Enter the keystore password and type **yes** to accept the certificate.

```
[admin@CISCOMUR-UI-147-17 bin]# ./keytool -import -alias root -keystore keystore
-trustcacerts -file root.cer
Enter keystore password:
Owner: CN=Guavus-MX1-CA, DC=guavus, DC=com
Issuer: CN=Guavus-MX1-CA, DC=guavus, DC=com
Serial number: 7570961243691e8641088832d6c218a5
Valid from: Sun Feb 14 15:36:50 GMT 2010 until: Fri Feb 14 15:46:49 GMT 2020
Certificate fingerprints:
     MD5:  27:BE:E3:EA:DB:16:A1:79:8B:00:96:A4:54:D8:16:6D
     SHA1: 8B:6B:44:2B:08:FE:3F:C5:8B:B5:AD:FA:EE:DF:E2:A7:07:59:D0:49
     Signature algorithm name: SHA1withRSA
     Version: 3
Extensions:
#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]
#2: ObjectId: 2.5.29.15 Criticality=false
KeyUsage [
  DigitalSignature
  Key_CertSign
  Crl_Sign
]
#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D4 DA C9 FC FB AF 5E 32   BF C0 A3 F1 6F 03 39 C3  ......^2....o.9.
0010: 29 36 84 7D                                        )6..
]
]
```

```
#4: ObjectId: 1.3.6.1.4.1.311.21.1 Criticality=false
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

Step 4 Install the signed certificate.

```
[admin@CISCOMUR-UI-147-17 bin]# ./keytool -import -alias tomcat -keystore keystore
-trustcacerts -file tomcat.cer
```

Step 5 Enter keystore password.

The certificate reply was installed in keystore.

Step 6 Copy the new keystore to apache-tomcat-7.0.27 directory for all EDRs, Bulkstats and RGE on both master and standby machines.

```
[admin@CISCOMUR-UI-147-17 bin]# cp keystore /data/apache-tomcat/apache-tomcat-7.0.27/
[admin@CISCOMUR-UI-147-17 apache-tomcat-7.0.27]# cp keystore
/data/apache-tomcat-bulkstats/apache-tomcat-7.0.27/
[admin@CISCOMUR-UI-147-17 apache-tomcat-7.0.27]# cp keystore
/data/rge/apache-tomcat/apache-tomcat-7.0.27/
[admin@CISCOUCS-2-19 apache-tomcat-7.0.27]# scp keystore
admin@192.168.147.20:/data/apache-tomcat/apache-tomcat-7.0.27/
[admin@CISCOUCS-2-19 apache-tomcat-7.0.27]# scp keystore
admin@192.168.147.20:/data/apache-tomcat2/apache-tomcat-7.0.27/
```

Note 192.168.147.20 is the management IP address of the standby Rubix node.

Step 7 Verify that the keystorePass in **server.xml** of all Tomcat processes is the same as the keystore password given above.

```
# vi /data/apache-tomcat/apache-tomcat-7.0.27/conf/server.xml
# vi /data/apache-tomcat-bulkstats/apache-tomcat-7.0.27/conf/server.xml
# vi /data/rge/apache-tomcat/apache-tomcat-7.0.27/conf/server.xml

 <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               keystoreFile="keystore" keystorePass="rubix123"
               clientAuth="false" ciphers="SSL_RSA_WITH_RC4_128_MD5,
SSL_RSA_WITH_RC4_128_SHA, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA,
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA, SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA" sslProtocol="TLS" />
```

Make Performance Related Modifications

Step 1 Log in to the master Collector node and run the following commands:

```
> en
# configure terminal

(config)# internal set create -
/tps/process/hadoop/attribute/mapred.reduce.tasks.speculative.execution/value value
string false
(config)# internal set create -
/tps/process/hadoop/attribute/io.file.buffer.size/value value string 131072
```

```
(config)# internal set create -
/tps/process/hadoop/attribute/io.sort.record.percent/value value string 0.11
(config)# internal set create - /tps/process/hadoop/attribute/io.sort.factor/value
value string 100
(config)# internal set create -
/tps/process/hadoop/attribute/io.sort.spill.percent/value value string 0.5
(config)# internal set create -
/tps/process/hadoop/attribute/mapred.child.java.opts/value value string -Xmx3584m
(config)# internal set create -
/tps/process/hadoop/attribute/mapred.min.split.size/value value string 1288490189
(config)# internal set create -
/tps/process/hadoop/attribute/mapred.tasktracker.map.tasks.maximum/value value string
16
(config)# internal set create -
/tps/process/hadoop/attribute/mapred.tasktracker.reduce.tasks.maximum/value value
string 4
(config)# internal set create -
/tps/process/hadoop/attribute/heartbeat.recheck.interval/value value string 130000
(config)# write memory
```

Step 2 Log in to the standby Collector node and repeat the above steps.

Start the Collector Process

Step 1 Log in to the master Collector node and run the following commands:

```
> en
# conf t
(config) # write memory
(config) # pm process tps restart
```

Step 2 Verify whether all Compute nodes have joined the hdfs cluster or not. All should be shown with status as Normal. The output should list six processes, as shown in the following example.

```
(config) # _shell
# ps -ef | grep hadoop | awk {'print $NF'}
org.apache.hadoop.hdfs.server.datanode.DataNode
org.apache.hadoop.hdfs.server.namenode.NameNode
hadoop
org.apache.hadoop.hdfs.server.namenode.NameNode
org.apache.hadoop.hdfs.server.namenode.SecondaryNameNode
org.apache.hadoop.mapred.JobTracker

hadoop dfsadmin -report 2>/dev/null | 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 If all six processes are not listed on the first attempt, reissue the command 10 minutes after starting the TPS process from Step 1.

After the hdfs related processes are up, you can start the collector processes.

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

Step 3 Perform the following steps on the standby Collector node:

```
> en
# conf t
(config) # write memory
(config) # pm process tps restart
(config) # _shell
# ps -ef | grep hadoop | awk {'print $NF'}
org.apache.hadoop.hdfs.server.datanode.DataNode
org.apache.hadoop.hdfs.server.namenode.NameNode

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

Note If these processes are not listed on the first attempt, then reissue the command 10 minutes after starting the TPS process.

Processing the Data

This section includes information for setting up a user for ASR in the Collectors, sending the EDR and Bulk stats data feeds to the MURAL Platform, setting the data start time, and running the data processing commands.

Use one of the ASR 5000 data feed methods to send data to the MURAL platform.

Setting Up a New User for ASR in the Collectors

The following are the steps required to set up a new user for ASR in the Collectors.

Step 1 Log on to the master Collector node and create the user:

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

Note The username and password should be the same ones configured for EDR and Blukstats files transfer on the ASR5000.

Step 2 Edit **/etc/ssh/sshd_config** to set the following parameters, as indicated:

```
UsePAM yes
PasswordAuthentication no
```

Step 3 Run the **sshd restart** command:

```
# en
> conf
(config) pm process sshd restart
(conifg) _shell
```

Step 4 Repeat steps 1, 2, and 3 on the standby Collector node.

ASR 5000 Data Feed

Start sending the EDR and bulk stats data feeds to the MURAL Platform. If the ASR5000 is used as an input node, the start time from the filename is created in the collector folder - /data/collector/edrflow.

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

Note It is assumed that the timestamp on the data that is pushed to the platform is >= current and not an old timestamp.

Set the Data Start Time

This section provides instructions for setting the data start time in the configuration.

Step 1 Log in to the GMS server:

```
> en
# _shell
# mount -o remount,rw /
# cd /opt/var/MURAL_Install_1_v1.5/bin
# ./setOozieTimeMUR31Atlas31_DC --profile 10.MURAtlas30 --node <IP_ADDRESS>
--dataStartTime 2013-04-01T06:00Z --verbose
```

Step 2 Execute the Set Job Time Script for both Master and Standby Collector nodes:

Step 3 Execute the script to set the data start times to the time from which EDR and BulkStats data starts coming into the system.

For example, if EDR and Bulk Stats data starts coming into the system from 1st April, 2013, 06:00 onwards, run the following scripts with the start_time value as "2013-04-01T06:00Z":

Note Enter minutes as a multiple of 5. For example, "2013-06-21T05:25Z".

```
" ./setOozieTimeMUR31Atlas31_DC --profile 10.MURAtlas30 --node <Collector_Node_Mgmt_IP>
--dataStartTime <start_time>  --verbose"2.
```

Step 4 Set the data start time in the configuration based upon the timestamp set during the data feed process, described above.

Start the Data Processing

Log in to the Master Collector/NameNode and from oozie subshell run the data processing commands:

```
> 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.

Validating the System Installations

Data Validation on the Collector Nodes

Step 1 Log on to the master Collector node:

```
> en
# _shell
```

Step 2 Execute the following two commands:

```
# hadoop dfs -ls /data/collector/1/output/edrflow/YYYY/MM/DD/HH/mm/* 2>/dev/null
# hadoop dfs -ls /data/collector/1/output/edrhttp/YYYY/MM/DD/HH/mm/* 2>/dev/nul
```

If the collector is receiving data in the expected format, it will persist it in hdfs.

Note Specify the year, month day, hour, and minute for which data is being sent to the MURAL system; minutes are always in multiples of 5 - 00,05,10,....55.

These directories and files are updated continuously as the data keeps coming in.

Step 3 Execute the following command:

```
# hadoop dfs -ls /data/collector/1/output/bulkStats/YYYY/MM/DD/HH/mm/* 2>/dev/null
```

If the collector is receiving data in the expected format, it will persist it in hdfs.

Note Specify the year, month, day, hour, and minute for which data is being sent to the MURAL system; minutes are always in multiples of 5 - 00,05,10,....55.

Data Validation on Compute Blades (Data Nodes)

EDR Data

All commands in this section are to be executed on the master Collector node.

Step 1 Check the last timestamp for EDR Cubes being generated by the EDR job from the master Collector node.

```
> en
# _shell
[admin@collector-1 0000041-121020165932574-oozie-admi-W]# hadoop dfs -text
/data/EDR/done.txt 2>/dev/null
```

Step 2 Check the last timestamp for CubeExporter Cubes being exported.

```
[admin@collector-1 0000041-121020165932574-oozie-admi-W]# hadoop dfs -text
/data/CubeExporter/done.txt 2>/dev/null
```

Step 3 Check the last timestamp for BulkStat Cubes being generated by the BulkStat Job:

```
[admin@CISCO-COL1-147-11 ~]# hadoop dfs -text /data/BulkStat/done.txt 2>/dev/null
# hadoop dfs -text /data/BSAgg15min/done.txt 2>/dev/null
```

Step 4 Check last timestamp for BulkStat Cubes being exported.

```
# hadoop dfs -text /data/BulkStatExporter_15min/done.txt 2>/dev/null
```

Data Validation on Insta Blades

Step 1 SSH to the Insta node and check the name of the database configured for EDR:

```
CISCOUCS-MUR-INSTA-01 [MUR-INSTA-CLUST: master] > en
CISCOUCS-MUR-INSTA-01 [MUR-INSTA-CLUST: master] # _shell
[admin@CISCOUCS-MUR-INSTA-01 ~]# cli -t "en" "conf t" "show runn full" | grep "insta
instance 0 cubes-database" | awk -F ' ' '{print $5}'
ucsdb_UI_31rc4
```

Step 2 Connect to idbmysql and select the database:

```
[admin@CISCOUCS-MUR-INSTA-01 ~]# idbmysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 151
Server version: 5.1.39 MySQL Embedded / Calpont InfiniDB Enterprise 2.2.9.1-1 GA
(Commercial)
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
mysql> use ucsdb_31rc4;
Database changed
```

Step 3 Get the value for the **mints** and **maxts** field for -1 aggregation level and 60 minute bin class:

```
mysql> select * from bin_metatable;


+----------+--------------------+------------+------------+---------+
| binclass | aggregationinterval | mints      | maxts      | bintype |
+----------+--------------------+------------+------------+---------+
| 60min    |                 -1 | 1350126000 | 1350594000 | NULL    |
| 60min    |              86400 | 1350086400 | 1350432000 | NULL    |
| 60min    |             604800 |          0 |          0 | NULL    |
| 60min    |            2419200 |          0 |          0 | NULL    |
+----------+--------------------+------------+------------+---------+
4 rows in set (1.14 sec)

Press Ctrl+D to exit
mysql> Bye
```

Step 4 Convert the date format: Run the date command with the value of **maxts** captured from the step above. The example shows the user has processed data from Oct 13 11:00AM to Oct 18 21:00 PM.

```
[admin@CISCOUCS-MUR-INSTA-01 ~]# date -d @1350126000 Sat Oct 13 11:00:00 UTC 2012
[admin@CISCOUCS-MUR-INSTA-01 ~]# date -d @1350594000 Thu Oct 18 21:00:00 UTC 2012
```

Validate Bulk Stats Data on the Caching Compute Blade

Step 1 SSH to the Insta node and check the name of the database configured for EDR:

```
CISCOUCS-MUR-INSTA-01 [MUR-INSTA-CLUST: master] > en
CISCOUCS-MUR-INSTA-01 [MUR-INSTA-CLUST: master] # _shell
[admin@CISCOUCS-MUR-INSTA-01 ~]# cli -t "en" "conf t" "show runn full" | grep "insta
instance 1 cubes-database" | awk -F ' ' '{print $5}'
bsdb_UI_31rc4
```

Step 2 Connect to idbmysql and select the database:

```
[admin@CISCOUCS-MUR-INSTA-01 ~]# idbmysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 151
Server version: 5.1.39 MySQL Embedded / Calpont InfiniDB Enterprise 2.2.9.1-1 GA
(Commercial)
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
mysql> use bsdb_UI_31rc4;
Database changed
```

Step 3 Select Data from table. Run the following command to get the value for mints and maxts field for 900 aggregation interval:

```
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
```

Step 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. This shows we have processed data from Sun Mar 31 07:00:00 UTC 2013 to Tue Apr 30 03:45:00 UTC 2013.

```
[admin@CachingCompute-GMS-2 ~]# date -d@1367293500
Tue Apr 30 03:45:00 UTC 2013
[admin@CachingCompute-GMS-2 ~]# date -d@1364713200
Sun Mar 31 07:00:00 UTC 2013
```

Start UI Processes and Verify Data

Start the Rubix Tomcat Instance on Both UI Nodes

Note You should only start UI Tomcat instances after at least 2 hours of data has been pushed into the Insta node.

Step 1 Log in to the master UI node:

```
> en
# _shell
# cd /data/apache-tomcat/apache-tomcat-7.0.27/bin/
# ./startup.sh
```

Step 2 Log in to the standby UI node:

```
> en
# _shell
# cd /data/apache-tomcat/apache-tomcat-7.0.27/bin/
# ./startup.sh
```

Step 3 Start the second tomcat instance on the standby UI node:

```
# cd /data/apache-tomcat2/apache-tomcat-7.0.27/bin/
# ./startup.sh
```

Step 4 Start the RGE Tomcat instance on the master UI node.

```
# cd /data/rge/apache-tomcat/apache-tomcat-7.0.27/bin/
# ./startup.sh
```

Step 5 Start the Tomcat instance for BulkStats on the master UI node.

Allow a space of five minutes between starting rubix tomcat and starting bulk stats tomcat:

```
# cd /data/apache-tomcat-bulkstats/apache-tomcat-7.0.27/bin
# ./startup.sh
```

Step 6 Access the UI's by going to the URL https://<domainName>:8443/ through your browser.

For example :

```
https://demo.sanmateo.com:8443/
Username: admin
Password: admin123
```

Note Once the installation is completed, be sure to back up the configurations. Refer to the *Cisco MURAL Platform Monitoring and Troubleshooting Guide* for more information.

GLOSSARY

A

Aggregate Flow Count / Subscribe — The number of flows per subscriber.

B

Big data — A collection of data that is too large to manage, store, or analyze by traditional methods such as relational database management systems and desktop applications. The size at which a data collection is considered “big data” depends on the industry and the common software tools available to capture and process the data.

C

Concurrent Flows (Count) — Number of flows which are active at the same time.

Cluster — The cluster of nodes including the Collector node(s), Compute node(s), Master node(s), Scheduler node(s) and Cube Storage node.

Collector Cluster — The cluster of nodes consisting of the Collector nodes in active/standby High Availability clustering.

Collector node — Consists of a cluster of Collectors that collect data from the ASR 5000. The Collector is optimized for low-latency, high-throughput transactions, and it assembles and understands the exported flows. The Collector node distributes data to the local compute node.

Compute Cluster — The cluster consisting of the master and standby Compute nodes.

Compute node — Also called the Data node. Analyzes and aggregates data. It is connected by cable to the Collector nodes and consists of multiple clusters. The Compute node sends the data to the storage array and makes it available to applications.

Concurrent Sessions Count — Number of sessions which are active.

Cube engine — The Cube engine (Rubix) and the RG (report generation) engine are hosted on the UI node. The Cube engine forwards requests from the UI engine to the Insta node. It also prefetches data and locally caches it so that if the requested data is in the local cache, it can return the response directly to the UI node without querying the Insta node. The RG engine serves as the HTTP request server.

D

Downlink Rate The average bytes received by the mobile device from the Internet during a selected interval.

Downlink Tonnage The total amount of data received by the mobile device from the Internet.

F

Flows (fps) AVG number of session flows in a second (Count of flows per Second –fps)

Flow Duration (seconds) Duration of flows which are active during selected time interval.

H

Hadoop Open-source software that supports running applications on large clusters of hardware. See http://hadoop.apache.org/

I

Image Operating system plus the application bundle.

Insta node Also called the Caching Compute node. Generates and manages caches of processed data. The processed data (cubes) are stored in the Insta database (infinidb). The data is generally hosted on a separate SAN device, though sometimes hosted on the Insta node itself.

I

N

Number of Subscribers The unique number of subscribers active during the sample period.

R

Rate (bps) Amount of data sent and received per second between the device and the Internet. (Bytes per Second – bps)

RG engine Report Generation engine, which serves as the HTTP request server. In Cisco MURAL, it is hosted on the same blade as the UI engine.

Rubix engine See Cube engine.

S

Session Count / Subscriber The number of sessions per subscriber which are active.

T

Tonnage (MB) Total volume amount of data sent and received. (MegaBytes –MB)

U

Uplink Tonnage The total amount of data sent from the mobile device out to the Internet.

Uplink Rate The average bytes sent from the mobile device out to the Internet during a selected interval.

UI node Includes both the Cube engine (Rubix) and the RG (report generation) engine. The Cube engine forwards requests from the UI engine to the Insta node. It also prefetches data and locally caches it so that if the requested data is in the local cache, it can return the response directly to the UI node without querying the Insta node. The RG engine serves as the HTTP request server.