MURAL Software Standard Installation Guide

Version 3.6

Published: 2015-10-13
THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

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

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

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

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

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

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

Copyright © 2015, Cisco Systems, Inc. All rights reserved.
# Table of Contents

**Installation Overview** ................................................................. 7
- Before you begin ............................................................................. 7
- System Components ....................................................................... 8
- Standard Setup ............................................................................... 8
- Installation Package ....................................................................... 10
- Customer Information Questionnaire (CIQ) Sheet ........................... 10

**Installing MURAL Software** .......................................................... 12

**Setting up Master GMS Node** ....................................................... 14
- Manufacture the Master GMS Node ................................................ 14
- Set Up the Master GMS Node .......................................................... 20
- Collect Storage Information and Update the CIQ Sheet .................... 25

**Configuring MURAL Nodes** ............................................................ 27
- Apply Patches on MURAL Nodes .................................................... 27
- Prepare the Final XML with WWIDs and Applications ..................... 27
- Install Appliance on MURAL Nodes ............................................... 34
- Configure Site-Specific Applications .............................................. 37
- Troubleshooting Node Installation .................................................. 37
- Verify the Status of Processes ......................................................... 37

**Generating and Pushing the Information Bases** ............................... 40
- Configuring IBs for EDR ................................................................. 40
- Configuring DCs and Gateways For All IBs ..................................... 41
- Verify Collector Process ................................................................ 49

**Processing the Data** ..................................................................... 51
Setting Up a New User for the ASR in the Collectors ........................................51
Ingesting Data Into the System ...........................................................................52

Validating Data on Nodes ..................................................................................53
Validating Data on the Collector Nodes ...............................................................53
Validating Data on the Compute Blades (Data Nodes) .......................................55
Validating Bulk Stats Data on the Insta Blade ....................................................57
Starting UI Processes and Verifying Data ...........................................................59
Updating Whitelists ..............................................................................................61

Setting Up Offline Reports ..................................................................................62
Uncategorized URL, UA, and TAC Reports ........................................................62
Tethering Databases .............................................................................................62
Tethering Reports ..................................................................................................64
APNTethering Reports ...........................................................................................65

Mandatory Parameters for Incoming ASR Files ...................................................67
Mandatory Attributes for Flow EDRs for MURAL ..............................................67
Mandatory HTTP EDR Attributes for MURAL ....................................................68
ASR-Side Configuration .........................................................................................69

Appendix I: PXE Boot of Blades using the KVM console .................................71
Installation Overview

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

Before you begin

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

- Linux
- Cisco UCS 2.1
- EMC storage devices

Prior to installing the application, we recommend that you:

- Review the MURAL 3.6 Release Notes.
- Complete a training course on MURAL.
- Have an understanding of Cisco UCS 2.1 hardware administration.
- Ensure that MURAL system hardware installation has been completed successfully as specified in the bill of materials (BOM) and the setup is ready to install the system. For more information, refer to the MURAL Hardware Installation Guide.
- Verify that each Fabric Interconnect (FI) is connected to the SAN controllers through fiber cables.
- Verify all UCS B200 M2/M3 blade servers have been installed physically on the UCS 5108 Blade Server Chassis and connected to the UCS 6248 Fabric Interconnect for each of the types of nodes (GMS, Collector, Compute, Insta and Rubix nodes). The number of each type of node is customized for your deployment.
- Update the CIQ sheet completely, excluding 'storageInfo' section with MURAL setup details as required.
System Components

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

![Diagram showing MURAL platform components]

Standard Setup

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

1. General Management System (GMS)—Provides centralized management of the Reflex platform nodes, such as remote manufacturing of blades (installing the MURAL operating system), patch management, monitoring of all nodes and operations, and importing and running node configurations. The GMS node cluster supports high availability.

2. Collector—Collects data streams pushed to the Reflex platform, interprets the exported flows, enriches them with static data, and assembles data sets. The Collector stores the raw data in the Hadoop file system (HDFS) and sends it to the Compute node. The Collector node cluster can have any number of servers, in pairs for master and standby and uses 1+1
redundancy (transparent failover between pairs of active-active nodes).

3. **UI/Caching (Rubix)**—Hosts the Rubix engine and data cache. The Rubix engine queries the Insta nodes constantly and when new data is available, it fetches it to store in the data cache, so that it can respond more quickly to requests from the UI engine. The Caching node is sometimes called the Rubix node. The Caching node uses N+1 redundancy in active-active mode.

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

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

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

**Starter Pack Setup**

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

**Medium Pack Setup**

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

**Hardware**

The MURAL application is hosted on the UCS 5108 Blade Server Chassis that comprises GMS, Collector, Rubix, Compute and Insta nodes. Data storage is hosted on the EMC storage devices.

The data flows that feed the MURAL system are pushed by an ASR 5000 or ASR 5500 platform (hereafter referred to as an ASR).
# Installation Package

The MURAL software installation package contains the following components:

- An ISO image—For example: mfgcd-atlas3.7.rc1.iso
  For the exact image name and the MD5 checksum for the software image, refer to the release notes for your release.

- The CIQ sheet file, which is used to create hardware and final configuration file (XML). This CIQ sheet is used to gather basic configuration information like IP addresses, cluster details, and storage details that will be used to create the configuration file (XML) for installation and configuration of MURAL system.

- Software patches that are available for the release. Refer to the *MURAL Release Notes, 3.6* for a complete list of patches.

  **Note:** Media QoE Analytics is also installed as a patch. For more information, refer to the Media QoE Analytics Release Notes.

- Management information bases (MIBs).

# Customer Information Questionnaire (CIQ) Sheet

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

The CIQ sheet file contains the following worksheets:

- MiscInfo — Specifies configuration file name, site, Hardware Type, Login message, and NTP settings.

- NetworkInfo — Specifies networks, default gateway, DNS, NTP, and SNMP servers. It also specifies UCS Manager access details, which is used to collect the MAC addresses from blades. MAC address can be left blank as the script that runs on CIQ can populate them automatically. If the script gives a warning that it is not able to get the MAC address, then you need to manually provide MAC addresses for all the nodes and run the script again.

- ClusterInfo — Specifies cluster names and type as well as interfaces and VIPs (if any).
MURAL Software Standard Installation Guide

- NodeInfo — Specifies chassis and slot IDs, hostnames, Mac IDs and IP addresses, KVM IPs of blades and Cluster Names for all the blades to group into specific roles.
- StorageInfo — Specifies node hostnames, WWIDs and MURAL components like Collectors, Datanodes, etc. to assign LUNs from SAN storage devices.

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

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

To install MURAL, perform the following steps:

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

1. Obtain the CIQ sheet and verify that the UCS hardware is correctly set up and configured for the MURAL system. For more information, refer to the *MURAL Hardware Installation Guide*.

2. Prepare the Master GMS node and copy CIQ sheet into it. Create the Hardware XML (without WWID) and manufacture all other MURAL nodes using this GMS node as PXE boot server. See "Setting up Master GMS Node" on page 14.

3. Collect WWIDs from all MURAL nodes and update CIQ sheet. Recreate Hardware XML (with WWID) and attach application templates to prepare the final configuration XML. See "Collect WWIDs and Update the CIQ Sheet" section in "Configuring MURAL Nodes" on page 27.

4. Install appliances into all the MURAL blades.

The following table lists the installation types and the order in which the appliances must be installed for each type:

<table>
<thead>
<tr>
<th>Serial Number</th>
<th>Standard Pack</th>
<th>Medium Pack</th>
<th>Starter Pack</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Master GMS Node</td>
<td>Master GCN Node</td>
<td>Master GCU Node</td>
</tr>
<tr>
<td>2</td>
<td>Standby GMS Node</td>
<td>Standby GCN Node</td>
<td>Standby GCU Node</td>
</tr>
<tr>
<td>3</td>
<td>Data Node Cluster</td>
<td>Data Node Cluster</td>
<td>Data Node Cluster</td>
</tr>
<tr>
<td>4</td>
<td>iINSTA Node Cluster</td>
<td>iINSTA Node Cluster</td>
<td>iINSTA Node Cluster</td>
</tr>
<tr>
<td>5</td>
<td>UI Node Cluster</td>
<td>UI Node Cluster</td>
<td>Not applicable</td>
</tr>
<tr>
<td>6</td>
<td>Apps Cluster (RGE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Collector Node Cluster</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See "Install Appliance on MURAL Nodes" section in "Configuring MURAL Nodes" on page 27.

5. Configure environmental parameters specific to your environment. See "Site
specific Applications configuration" section in "Configuring MURAL Nodes" on page 27.


7. Generate and push the information bases. See "Generating and Pushing the Information Bases" on page 40.

8. Process the data. See "Processing the Data" on page 51.

9. Verify that the system is working as expected. See "Validating Data on Nodes" on page 53.
Setting up Master GMS Node

Set up the master GMS node that is used to create configuration files (XML). This node is manufactured by mapping the MURAL release ISO image file as virtual media from UCS blade console. Manufacture all other nodes by installing OS through PXE boot and running the `install appliance` command to prepare these other nodes for the MURAL setup. If PXE boot does not work, manufacture other nodes manually.

Manufacture the Master GMS Node

Prerequisite

Ensure that Serial over LAN (SOL) is configured on all the blades during UCS setup. For information about configuring SOL, refer to the MURAL Hardware Installation Guide.

To manufacture the master GMS node, perform the following steps:

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

   ISO Image: mfgcd-atlas3.7.rc1.iso

   # md5sum mfgcd-atlas3.7.rc1.iso

   7802654d0a1ab8be23d9dada20e192f8  mfgcd-atlas3.7.rc1.iso

   #

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

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

   The login prompt appears when the node has successfully completed the reboot.

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

   The following image illustrates the opening of the kvm.jnlp file.
5. In the KVM Native Library warning box, click **Ok**.

The following image illustrates the KVM Native Library warning box.

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

7. Click **Add Image** and specify the path of the ISO image that you downloaded in Step 1.

The following image illustrates the image that needs to be added.
8. Check the check box in the **Mapped** column next to the ISO image to mount it.

The following image illustrates the selection of the check box.
9. Reboot the blade so that it can boot with the mounted image. Click the **KVM** tab and mouse over the **Ctrl-Alt-Del** from the **Macros** drop-down menu to select the option.

The following image illustrates the **Macro** drop-down list.
**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. Ensure that the same ISO image is not selected for two nodes while manufacturing is in progress for a given node.

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

11. Start the manufacturing process by executing the following manufacture command:

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

12. Follow the screens illustrated below to manufacture the node:
eth0:  DEV 0000:00:0e.0
[ 55.613425] EDAC MCI: Giving out device to 'sbridge_edac.c' 'Sandy Bridge Soc
eth1:  DEV 0000:00:0e.0
[ 55.740329] EDAC sbridge: Driver loaded.
Running startup scripts.
Running /etc/init.d/rcS.d/S10tms_dhcpc
Starting DHCP client on interfaces: eth0 eth1

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

man manufacture.sh -t -u -f /mnt/cdrom/image.img -m 1D -L 1D --cc no --cs no --cl no -a
The system displays the "Manufacture done" message and returns to the # prompt once the manufacturing of a blade is completed.

13. Deselect the ISO image selected in step 7. Type `reboot` to reboot the node with the new ISO image.

**Set Up the Master GMS Node**

To set up the master GMS node, perform the following steps:

1. Log in to the GMS node using the Console.

2. Set the password:

```plaintext
> en
# conf t
(config) # username admin password admin@123
(config) # write memory
(config) # exit
```
3. Assign the IP addresses for the management interface and default gateway:

```
> en
# conf t
(config) # interface <mgmt_interface> ip address <mgmt_IP_of_GMS_server><subnetmask_of_mgmt_network>
(config) # ip default-gateway <mgmt_network_default_gateway_IP>
```

For example:

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

4. Download the patches from the FTP server to the GMS server in the /data/tmp_patch directory and apply all patches applicable for GMS nodes. For more information, refer to the MURAL Release Notes for a complete list of 3.6 release patches and installation instructions.

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

```
# cd /opt/etc/scripts/mmi/CIQ2GMSXML/src/
# python GMSXMLGenerator.py -f <XML-file-name-path> -s 0
# python GMSXMLGenerator.py -f /data/Mural_CIQ_Template.xls -u true -s 0
```

Successful execution creates an XML file under /data/configs/gms/ folder. In case of error, details are available in the log file stored at /data/mmi/ciq2gmsxml.log.
Note: The command executed in this step collects MAC IDs from all the blades and interfaces specified in the CIQ sheet. If UCS Manager is not enabled to fetch MAC IDs, update the CIQ sheet for MAC IDs and run the command above without the \'-u true\' option.

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

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

7. Start gms_server in lite mode and activate the Hardware XML file.

```
# cli -m config
(config)# gms enable lite-version
(config)# pm process gms_server restart
(config)# write memory
(config)# _shell
```

8. Check the status of the GMS server by running the following command:

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

    Current status: running
# ps -eaf | grep tomcat | grep -v grep | awk '{print $(NF-1)}'

org.apache.catalina.startup.Bootstrap
#
```

Before running the next command (to activate XML), wait until the preceding output is displayed.

```
# cli -m config
(config)# gms config Mural_Hardware.xml activate
```
9. Copy (by SCP or FTP) ISO image file into Master GMS node under /data directory.

10. Configure the GMS node as PXE boot server:

   ```
   (config) # image fetch scp://admin@<GMS-IP>:/data/<iso-image-name>
   (config) # image mount <iso-image-name>
   ```

   For example:

   ```
   (config) # image fetch
   scp://admin@192.168.191.103:/data/mfgcd-atlas3.7.rc1.iso
   (config) # image mount mfgcd-atlas3.7.rc1.iso
   ```

   The following sample may resemble the output:

   ```
   (config) # image mount mfgcd-atlas3.7.rc1.iso
   Copying linux...
   Copying rootflop.img...
   Copying image.img...
   (config) # write memory
   (config) # _shell
   ```

11. Verify the image on the GMS node:

   ```
   # ls -l /var/opt/tms/images/
   total 2991084
   -rw-r--r-- 1 root root 3223255040 May 22 17:24 mfgcd-atlas3.7.rc1.iso
   ```

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

   ```
   # cli -m config
   ```

   For all the nodes:
For example:

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

For a single node:

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

For example:

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

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

The following sample may resemble the output:

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

This command triggers blade reboot from network. Once the blade starts booting from the network, GMS pushes the image on the blade using PXE boot. Manufacture process can be started on each blade in parallel.
A blade takes approximately 30 to 45 minutes to manufacture with the new image. Run the following command to check the blade manufacture status.

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

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

```
(config) # gms show manufacturing-status cluster DN-CLUS
UCS-DN-1: OK : Product release: 3.7.1
UCS-DN-2: OK : Product release: 3.7.1
(config) #
```

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

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

**Collect Storage Information and Update the CIQ Sheet**

Once all the blades are manufactured successfully, configure storage LUNs for each blade as described in the *MURAL Hardware Installation Guide*.

To collect storage information:

1. Log in to each node and run the following commands:

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

2. Wait for the node to complete reboot, login and execute the following command to collect the WWID:

```
(config) # tps multipath show
```

Update these WWIDs into storageInfo worksheet of CIQ sheet for the respective node.

3. Repeat the preceding steps for all the blades, including the GMS node.
Configuring MURAL Nodes

After you have installed the MURAL software, perform the following tasks:

- "Apply Patches on MURAL Nodes" below
- "Prepare the Final XML with WWIDs and Applications" below
- "Install Appliance on MURAL Nodes" on page 34
- "Configure Site-Specific Applications" on page 37
- "Troubleshooting Node Installation" on page 37
- "Verify the Status of Processes" on page 37

Apply Patches on MURAL Nodes

Apply all the patches applicable for the respective MURAL nodes.

**Note:** Refer to the MURAL Release Notes, 3.6 for a complete list of patches and installation instructions.

Prepare the Final XML with WWIDs and Applications

**Prerequisite**

Copy (SCP or FTP) the updated CIQ sheet into GMS node under /data directory.

To prepare the final XML:

1. Log in to the GMS node using ssh and run the following commands:

```
> en
# _shell
# cd /opt/etc/scripts/mmi/CIQ2GMSXML/src/
# python GMSXMLGenerator.py -f /data/Mural_CIQ_Template.xls -s 1

File Exists!! Do you want to overwrite file at /data/configs/gms/Mural_Hardware.xml [Y/N]: Y
GMS HARDWARE XML saved as /data/configs/gms/Mural_
Hardware.xml

SUCCESSFULLY COMPLETED GMS HARDWARE XML GENERATION

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

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

2. From the command terminal, launch Mobility Unified Reporting and Analytics (MURAL) Deployment wizard to prepare the final XML:

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

The Mobility Unified Reporting and Analytics (MURAL) Deployment wizard is launched on the terminal to accept various configuration inputs.
General instructions to use the Mobility Unified Reporting and Analytics (MURAL) Deployment wizard:

- Use arrow keys to navigate in the Deployment wizard.
- Use Space-bar to choose the selected item(s).
- Press the Enter key after selecting Next, Previous or Exit to move into next screen, previous screen or exit from the Deployment wizard, respectively.
- Selected items appear with swapped colors for text and background.

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

- HTTP Errors
- Bulkstats and KPI
- Anomaly
- Tethering

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

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>GMS Cluster</th>
<th>Collector Cluster</th>
<th>UI Cluster (RGE)</th>
<th>iNSTA Cluster</th>
<th>DN Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Pack</td>
<td>GMS</td>
<td>Namenode &amp; Collector</td>
<td>Rubix-DPI Other Rubix Apps</td>
<td>iNSTA</td>
<td>Compute</td>
</tr>
</tbody>
</table>

Copyright © 2015, Cisco Systems, Inc.
The following image illustrates the available cluster configurations.

![Cluster Configurations Image]

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

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>GMS Cluster</th>
<th>Collector Cluster</th>
<th>UI Cluster (RGE)</th>
<th>iNSTA Cluster</th>
<th>DN Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium Pack</td>
<td>GC (GMS + Namenode &amp; Collector)</td>
<td>Rubix-DPI Other Rubix Apps</td>
<td></td>
<td>INSTA</td>
<td>Compute</td>
</tr>
<tr>
<td>Starter Pack</td>
<td>GCU (GMS + Namenode &amp; Collector + UI)</td>
<td></td>
<td></td>
<td>INSTA</td>
<td>Compute</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adaptor.bulkStats.numThreads</td>
<td>Keep 2 threads for BS always and other two equal sets to flow and http, totaling 75% of collector cores.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adaptor.edrflow.numThreads:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adaptor.edrhttp.numThreads:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.atlas.cachePersistToDisk</td>
<td>true - enable disk based caching for atlas and BS; false - disable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.bulkstats.cachePersistToDisk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.atlas.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for atlas process (should be &lt;= 50% of Rubix node RAM)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.atlas.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.ruleEngine.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maxima maximum and initial java memory for ruleEngine process (keep it as default)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.ruleEngine.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property Name</strong></td>
<td><strong>Description</strong></td>
<td><strong>Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.bulkstats.rubixInstance.1.tomcat</td>
<td>Maximum and initial java memory for bulkstat process (should be &lt;= 10% of Rubix node RAM)</td>
<td>20g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InstanceMaxSize</td>
<td></td>
<td>20g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.bulkstats.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td>20g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.reportAtlas.rubixInstance.1.tomcat</td>
<td>Maximum and initial java memory for offline report process (keep it as default)</td>
<td>10g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InstanceMaxSize</td>
<td></td>
<td>5g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.reportAtlas.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.rge.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for report engine process (keep it as default)</td>
<td>10g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.rge.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td>10g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.httperror.rubixInstance.1.tomcat</td>
<td>Maximum and initial java memory for http error process (should be &lt;= 10% of Rubix node RAM)</td>
<td>20g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InstanceMaxSize</td>
<td></td>
<td>20g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.httperror.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.launcher.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for launcher process (keep it as default)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.launcher.rubixInstance.1.initialJavaHeapSize</td>
<td>Maximum and initial java memory for launcher process (keep it as default)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timezone:</td>
<td>Sets the timezone for deployment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FQDN:</td>
<td>Sets URL to access the MURAL UI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rgemailSupport:</td>
<td><a href="mailto:dummy-support@cisco.com">dummy-support@cisco.com</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mailSender:</td>
<td>Sets the E-mail ID as sender of offline reports.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mailHost:</td>
<td>Sets SMTP Server host address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mailPort:</td>
<td>Sets SMTP Server Port</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

4. To enable GMS as NTP server, set the value of the **Set_GMS_Server_As_NTP_Server** attribute as true in the file generated in step 3.
Install Appliance on MURAL Nodes

To install appliance on nodes:

1. SSH to the GMS server using the management IP address and start the configuration:

   **Note:** Restart the gms_server process in full-version mode for install appliance.

   ```
   > en
   # conf t
   (config) # pgsql dbroot /data/pgsql
   (config) # pgsql mode external
   (config)# pm process pgsqld restart
   (config) # gms enable full-version
   (config) # pm process gms_server restart
   (config) # write memory
   ```

   Check the status of the GMS server by running the following command:

   ```
   # cli -t "en" "config t" "show pm process gms_server" | grep "Current status"
   Current status: running
   # ps -eaf | grep tomcat | grep -v grep | awk '{print $(NF-1)}'
   org.apache.catalina.startup.Bootstrap
   #
   ```

   Before running the next command (to activate XML), wait until the preceding output is displayed:

   ```
   (config) # gms config Mural_Application.xml activate
   File successfully activated
   (config) #
   ```

   If you make any changes to the XML file, for example during troubleshooting, you must run this command again to activate the changed file.
Run the `install appliance` command for various clusters in sequence as mentioned in the following sections. For successful installation, the `Status` command displays “Node successfully installed” message.

You can view the status of a node by running the following command:

```
(config) # install appliance show installation-status cluster <Cluster Name> node <Hostname>
```

You can view the status of a cluster by running the following command:

```
(config) # install appliance show installation-status cluster <Cluster Name>
```

For the Standard pack, run the commands listed in the following table.

<table>
<thead>
<tr>
<th>Node</th>
<th>Command to Install Appliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master GMS</td>
<td>(config) # install appliance cluster cluster-name &lt;GMS Cluster&gt; node &lt;Hostname of GMS Server&gt; force-format</td>
</tr>
<tr>
<td>Standby GMS</td>
<td>(config) # install appliance cluster cluster-name &lt;GMS Cluster&gt; node &lt;Hostname of GMS Server&gt; force-format</td>
</tr>
<tr>
<td>Data Cluster</td>
<td>(config) # install appliance cluster cluster-name &lt;DN Cluster&gt; force-format</td>
</tr>
<tr>
<td>Collector Cluster</td>
<td>(config) # install appliance cluster cluster-name &lt;Collector Cluster&gt; force-format</td>
</tr>
<tr>
<td>Insta Cluster</td>
<td>(config) # install appliance cluster cluster-name &lt;iNSTA Cluster&gt; force-format</td>
</tr>
<tr>
<td>Collector Cluster</td>
<td>(config) # install appliance cluster cluster-name &lt;Collector Cluster&gt; force-format</td>
</tr>
<tr>
<td>UI Cluster</td>
<td>(config) # install appliance cluster cluster-name &lt;UI Cluster&gt; force-format</td>
</tr>
</tbody>
</table>

For the Medium Pack, run the commands listed in the following table.
### Command to Install Appliance

<table>
<thead>
<tr>
<th>Node</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master GMS Node (GCU cluster)</td>
<td><code>(config) # install appliance cluster cluster-name &lt;GMS Cluster&gt; node &lt;Hostname of GMS Server&gt; force-format</code></td>
</tr>
<tr>
<td>Standby GMS Node (GCU cluster)</td>
<td><code>(config) # install appliance cluster cluster-name &lt;GMS Cluster&gt; node &lt;Hostname of GMS Server&gt; force-format</code></td>
</tr>
<tr>
<td>Data Cluster</td>
<td><code>(config) # install appliance cluster cluster-name &lt;DN Cluster&gt; force-format</code></td>
</tr>
<tr>
<td>Insta Cluster</td>
<td><code>(config) # install appliance cluster cluster-name &lt;iNSTA Cluster&gt; force-format</code></td>
</tr>
<tr>
<td>UI Cluster</td>
<td><code>(config) # install appliance cluster cluster-name &lt;UI Cluster&gt; force-format</code></td>
</tr>
</tbody>
</table>

Run the following command from the master Name node:

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

For the Standard Pack, run the commands listed in the following table.

<table>
<thead>
<tr>
<th>Node</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master GMS Node (GCU cluster)</td>
<td><code>(config) # install appliance cluster cluster-name &lt;GMS Cluster&gt; node &lt;Hostname of GMS Server&gt; force-format</code></td>
</tr>
<tr>
<td>Standby GMS Node (GCU cluster)</td>
<td><code>(config) # install appliance cluster cluster-name &lt;GMS Cluster&gt; node &lt;Hostname of GMS Server&gt; force-format</code></td>
</tr>
<tr>
<td>Data Cluster</td>
<td><code>(config) # install appliance cluster cluster-name &lt;DN Cluster&gt; force-format</code></td>
</tr>
<tr>
<td>Insta Cluster</td>
<td><code>(config) # install appliance cluster cluster-name &lt;iNSTA Cluster&gt; force-format</code></td>
</tr>
</tbody>
</table>

Run the following command from the master Name node:

```
# pmx subshell hadoop_yarn repair hdfs
```
Configure Site-Specific Applications

1. If Anomaly Detection feature is enabled, run the following command on both the master and standby Name nodes:

```
# pmx subshell oozie set dataset anomalyMonthly attribute path /data/output/AnomalyAggDay/%Y/%M/%D/%H
```

Troubleshooting Node Installation

If the installation fails at any point, contact Technical Support.

Logs can be collected on the GMS server from the location /data/gms-server/logs.

Verify the Status of Processes

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

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

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

```
> en
# _shell
# ps -ef | grep java | grep Dproc | grep -v Dproc_dfs | awk '{print $9}'
-Dproc_historyserver
-Dproc_journalnode
-Dproc_namenode
-Dproc_resourcemanager
-Dproc_namenode
-Dproc_datanode
-Dproc_secondarynamenode
```
# hdfs dfsadmin -report 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:** The IP addresses mentioned in the sample code above are internal IP addresses.

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

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

3. Log in to the master Insta node, and run the following commands:

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

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

(config) # insta infinidb get-status-info
```
The output must show all modules in ACTIVE state. It must also list all the instances and Adaptor status as RUNNING.

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

5. Check Postgres process on both the Master and Standby nodes:

   For the Standard Pack setup: GMS, Collector and iNSTA nodes

   For the Medium Pack setup: GMS and iNSTA nodes

```bash
> en
# _shell
# ps -ef | grep postmaster | grep -v greppostgres
   2278   1   0 Feb13 ?      00:22:34
/usr/pgsql-9.3/bin/postmaster -p 5432 -D /data/pgsql/9.3/data
#
```
Generating and Pushing the Information Bases

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

Configuring IBs for EDR

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

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

To configure the IBs for EDR:

1. Log in to the master Collector node and execute following commands:

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

2. Enter the GGSN IPs and GGSN names:

   ```
   pm extension (aggregation center)> edit ib ipGgsn.map add
   GGSN IP: 27.23.157.1
   GGSN: GGSN1
   pm extension (aggregation center)> show ib ipGgsn.map
   1[27.23.157.1][GGSN1]
   pm extension (aggregation center)> 
   ```

3. Enter the SGSN IPs and SGSN name:
pm extension (aggregation center)> **edit ib ipSgsn.map add**
SGSN IP: 2.2.2.1
SGSN: SGSN1
pm extension (aggregation center)> **show ib ipSgsn.map**
1[2.2.2.1][SGSN1]
pm extension (aggregation center)>

4. Enter the APN name and corresponding APN group:

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

To configure the IBs for BulkStats:

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

**Configuring DCs and Gateways For All IBs**

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

The following section lists guidelines for adding new gateways:

- All input directories are created under the /data/collector path. Hence, in the example below, the ASR should send EDR files to data/collector/California/edr1 and send the Bulkstats file to /data/collector/California/bs1.
**Important:** Confirm the actual input directories with the Cisco Technical Support team for your ASR platform.

- Ensure that you provide different input paths for each new gateway being added.

- The `/edr-file-path` and `/bulkstat-file-path` should always start with a forward slash (`/`).

- The ASR should send the gateway name in place of `%DC`, as specified in the file name pattern in the Collector configurations.

- If the filenames will have file extensions of `.gz` or `.txt`, then you must provide `".*"` in the file name format configuration when adding the gateway.

- All incoming files should contain the string as per their type in the file name; that is, flow EDR files should contain the string "flow" delimited by an underscore (`_`) or hyphen (`-`) and similarly HTTP EDR files must contain string "http" delimited by an underscore (`_`) or hyphen (`-`) (so combinations would also work, like "_flow_" or "–http_").

- All collector (internal) IPs can be provided with comma (`,`) separated values like 10.10.10.133,10.10.10.134

- Following are some guidelines and samples to help in configuring filename patterns and collector configurations:
<table>
<thead>
<tr>
<th>Filename pattern</th>
<th>Example</th>
<th>Timestamp</th>
<th>Regex in Wrapper CLI</th>
<th>Regex in Collector Config</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway-name_ (multiple strings separated by underscore or hyphen or both)_ flow_timestamp str4.gz</td>
<td>Gateway-name_ str1_ str2_ str3_ flow_timestamp str4.gz</td>
<td>MMDDYYYY-hhmmss</td>
<td><em>_.</em> <em>.*%MM%DD%YYYY-%hh%mm%ss</em>* .gz</td>
<td>%DC_* _.<em>%MM%DD%YYYY%hh%mm%ss</em>.gz</td>
</tr>
<tr>
<td>Gateway-name_ (multiple strings separated by underscore or hyphen or both)_ flow_timestamp str4.gz</td>
<td>Gateway-name_ str1_ str2_ str3_ flow_timestamp str4.gz</td>
<td>MMDDYYYY-hhmmss</td>
<td><em>_.</em> <em>.*%MM%DD%YYYY-%hh%mm%ss</em>* .gz</td>
<td>%DC_* _.<em>%MM%DD%YYYY%hh%mm%ss</em>.gz</td>
</tr>
<tr>
<td>Filename pattern</td>
<td>Example</td>
<td>Timestamp</td>
<td>Regex in Wrapper CLI</td>
<td>Regex in Collector Config</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>-----------</td>
<td>----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Gateway-name_ (multiple strings separated by underscore or hyphen or both)_ flow_string_timestamp string.gz</td>
<td>Gateway-name_ str1_ str2_ str3_ flow_ str4_ timestamp_ str5.gz</td>
<td>MMDDYYYY-hhmmss</td>
<td><em>__</em>_ <em>__%MM%DD%YYYY-%hh%mm%ss_</em>.*.gz</td>
<td>%DC__*_ <em>__%MM%DD%YYYY%hh%mm%ss</em>.gz</td>
</tr>
<tr>
<td>Filename pattern</td>
<td>Example</td>
<td>Timestamp</td>
<td>Regex in Wrapper CLI</td>
<td>Regex in Collector Config</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Gateway-name_ (multiple strings separated by underscore or hyphen or both)_ flow_string_timestamp_string.gz</td>
<td>Gateway-name_str1_str2_str3_flow_str4_timestamp_str5_str6.gz</td>
<td>MMDDYYYY-hhmmss</td>
<td><em>_</em><em>*</em> <em><em>%MM%DD%YYYY-%hh%mm%ss</em></em>_.gz</td>
<td>%DC_<em>_</em>_ <em>_%MM%DD%YYYY%hh%mm%ss</em>.gz</td>
</tr>
<tr>
<td>Filename</td>
<td>Example</td>
<td>Timestamp</td>
<td>Regex in Wrapper CLI</td>
<td>Regex in Collector Config</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>-----------</td>
<td>----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Gateway-name- (multiple strings separated by underscore or hyphen or both)_ flow_string_timestamp_string.gz</td>
<td>Gateway-name_str1_str2_str3_str4_flow_str4_timestamp_str5_str6.gz</td>
<td>YYYYMMDD-hhmmss</td>
<td><em>.*</em>.<em>_.</em><em>YYYY%MM%DD-%hh%mm%ss</em>*_.gz</td>
<td>%DC_.<em>_.</em>_.<em>_YYYY%MM%DD%hh%mm%ss</em>.gz</td>
</tr>
<tr>
<td>Filename pattern</td>
<td>Example</td>
<td>Timestamp</td>
<td>Regex in Wrapper CLI</td>
<td>Regex in Collector Config</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Gateway-name_ (multiple strings separated by underscore or hyphen or both)<em>flow-string</em> timestamp_string.gz</td>
<td>Gateway-name_str1_str2_str3_flow-str4_timestamp_str5_str6.gz</td>
<td>MMDDYYYY-hhmmss</td>
<td><em>*</em>.<em>-</em><em>.%MM%DD%YYYY-%hh%mm%ss</em><em>.</em>.gz</td>
<td>%DC_<em>_</em>-<em>_.%MM%DD%YYYY%hh%mm%ss</em>.gz</td>
</tr>
</tbody>
</table>

* It is mandatory to send gateway name as the first substring in EDR files, followed by an underscore (_) character.

* We can have 'http' in place of 'flow'.

* If timestamp is in MMDDYYYYhhmmss then set %MM%DD%YYYY%hh%mm%ss in Regex pattern

* If timestamp is in YYYYMMDDhhmmss then set %YYYY%MM%DD%hh%mm%ss in Regex pattern

To configure gateways:

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

```bash
> en
# conf_t
```
(config)# pmx subshell aggregation_center
pm extension (aggregation center)> add gateway name gateway-name region gateway-region location gateway-area schema_version
bulkstat-schema-version ip gateway-IP timezone gateway-timezone
edr-filename-pattern incoming-EDR-fileName-pattern bulkstat-
filename-pattern incoming-BS-fileName-pattern edr-collector-
filename-pattern collector-config-edr-filename-format bulkstat-
collector-filename-pattern collector-config-bs-filename-format
collector-ip collector-IPs type gateway-type edr-file-path
incoming-EDR-files-path-on-collector bulkstat-file-path incoming-
BS-files-path-on-collector
pm extension (aggregation center)>

Tip: For bulkstat-schema-version, specify an integer only, for example use 15, and do not use “15.0”.

NOTE: The supported Bulk Stats schema versions are 15,16 and 17.

# pmx
pm extension> subshell aggregation_center
pm extension (aggregation center)> add gateway name pnst40ems5
region WEST location Pune schema_version 16 ip 10.132.69.121
timezone UTC edr-filename-pattern *_*_*_%MM%DD%YYYY%hh%mm%ss_
*.gz bulkstat-filename-pattern *%YYYY%MM%DD%hh%mm%ss edr-
collector-filename-pattern %DC_*_*_*_%MM%DD%YYYY%hh%mm%ss_*_.gz
bulkstat-collector-filename-pattern *%YYYY%MM%DD%hh%mm%ss collector-ip 10.10.10.101,10.10.10.102 type HA edr-file-path
/GMPLAB1/edr/ bulkstat-file-path /GMPLAB1/bs/

pm extension (aggregation center)>

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

pm extension (aggregation center)> show gateways

3. Verify the IP addresses for all Collectors:
MURAL Software Standard Installation Guide

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

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

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

5. Push the gateway configuration to all the Collectors:

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

6. Generate and push all IBs:

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

For Bulkstats:

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

Run the following commands if Anomaly Detection is enabled:

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

**Verify Collector Process**

To verify the collector process, run the following command on both the master and standby Collector:
cli -t "en" "conf t" "show pm process collector" | grep status

Current status: running
Processing the Data

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

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

Setting Up a New User for the ASR in the Collectors

To set up a new user for the ASR in the Collectors:

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

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

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

2. Repeat step 1 on the standby collector node.

3. Edit `/etc/ssh/sshd_config` to set the following parameters, in case a PAM related authentication error is reported in the master Collector node `/var/log/messages` for the ASR:

```bash
# mount -o remount,rw /
# vi /etc/ssh/sshd_config
UsePAM no
PasswordAuthentication yes
```

4. Set the sshd_config file attribute as immutable.

```bash
# chattr +i /var/opt/tms/output/sshd_config
```

Verify the sshd_config file attribute as below.
5. Run the `sshd restart` command:

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

6. Repeat steps 1 through 3 on the standby Collector node.

**Ingesting Data Into the System**

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

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

**Notes:**

- MURAL processes data that is received after the deployment is completed. MURAL is not configured to process the historical data.

- MURAL recommends ASR files to be of 10MB compressed size for optimum performance.
Validating Data on Nodes

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

Validating Data on the Collector Nodes

1. Log in to the master Namenode and go to _shell.

```bash
> en
# _shell
```

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

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

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

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

Setting the Data Start Time

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

1. Log in to the master Name node and make the / file system writable.

   ```bash
   > en
   # _shell
   # mount -o remount,rw /
   # cd /opt/deployment/Mural_setStartTime/
   # ./setOozieTime --dataStartTime data-start-time --node collector-mgmt-IP --password admin-password
   ```
2. Execute the `setOozieTime` script to set the time at which EDR and BulkStats data starts coming into the Hadoop directories listed in "Validating Data on the Collector Nodes" on the previous page.

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

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

**Note:** Enter minutes as a multiple of 5. For example, "2013-04-01T06:00Z." Ensure that there is a continuous flow of data into the Hadoop without any gaps since the specified time.

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

   **Note:** This script may take up to 30 minutes to complete for one node. Therefore, please wait until it completes and returns to the prompt.

**Starting the Data Processing**

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

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

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

**Note:** It may take approximately 20 minutes to start all the jobs, depending upon what all applications are enabled.
Validating Data on the Compute Blades (Data Nodes)

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

**Caution:** Wait two hours after completing the steps in "Starting the Data Processing" on the previous page. This allows sufficient time for the jobs that process the collector data to start, and the done.txt files to be updated. Not waiting could result in the checks failing.

Validating EDR Data

1. Log in to the master Collector node and go to the _shell.
   
   ```
   $ en
   # _shell
   ```

2. Check the last timestamp for the Core job.
   
   ```
   $ en
   # _shell
   # hadoop dfs -text /data/CoreJob/done.txt 2>/dev/null
   ```

3. Check the last timestamp for the EDR data cubes being generated by the job.
   ```
   # hadoop dfs -text /data/EDR/done.txt 2>/dev/null
   ```

4. Check the last timestamp for CubeExporter data cubes.
   ```
   # hadoop dfs -text /data/CubeExporter/done.txt 2>/dev/null
   ```

5. Check the last timestamp for generated and exported Bulkstats data cubes.
   ```
   # hadoop dfs -text /data/BulkStat/done.txt 2>/dev/null
   # hadoop dfs -text /data/BSAgg15min/done.txt 2>/dev/null
   # hadoop dfs -text /data/BulkStatExporter_15min/done.txt 2>/dev/null
   ```

Validating Insta Data

1. Log in to the master Insta node and check the name of the database configured for EDR:
2. Open the `idbmysql` user interface and specify `database_mural` as the database.

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

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

```
mysql> select * from bin_metatable;
+-----------------+-----------------+-----------------+-----------------+-----------------+
| binclass | aggregationinterval | mints     | maxts     | bintype     |
+-----------------+-----------------+-----------------+-----------------+-----------------+
| 60min | | -1| 1406710800 | 1409295600 | NULL |
| 60min | | 86400| 0 | 0 | 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
```

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

The following example indicates that data was processed between 09:00 on July 30 and 07:00 on August 29.

```
# date -d @1406710800
Wed Jul 30 09:00:00 UTC 2014
```
The following example indicates that data was processed between 11:00 on October 13 and 21:00 on October 18.

Validating Bulk Stats Data on the Insta Blade

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

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

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

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

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

```
mysql> select * from bin_metatable;
+-----------------+-----------------+--------+--------+--------+
| binclass | aggregationinterval | mints | maxts  | binType |
+-----------------+-----------------+--------+--------+--------+
| 5min      | -1              | 0      | 0      | NULL   |
| 5min      | 900             | 1406713500 | 1409301900 | NULL   |
| 5min      | 3600            | 0      | 0      | NULL   |
| 5min      | 86400           | 0      | 0      | NULL   |
```
4. Convert the date format. Run the `date` command with the value of `maxts` (captured from the step above) for the row which shows `aggregationinterval` as 900.

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

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

The following example indicates that data was processed between 07:00 on March 31 and 03:45 on April 30.

```
# date -d @1367293500
Tue Apr 30 03:45:00 UTC 2013
# date -d @1364713200
Sun Mar 31 07:00:00 UTC 2013
```
Starting UI Processes and Verifying Data

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

Starting the Rubix Tomcat Instance on Both UI Nodes

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

1. Log in to the master UI node.

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

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

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

Check the tomcat process status using command:

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

**Note:** Ensure the running status of above service as *Current Status: running* before proceeding to start next process.

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

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

(config)# rubix modify-app ApplicationName enable
(config)# rubix modify-app ApplicationName modify-instance 1 enable
(config)# write memory
```
Where *ApplicationName* is replaced by the following applications in the same order:

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

Check the tomcat process status using command:

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

**Note:** Ensure the running status of above service as Current Status: running before proceeding to start next process. Check the corresponding rubix.log file to ensure that Atlas and HET applications are started properly without any exception.

4. Log in to the standby UI node and repeat Steps 2 and 3.

5. Access the UIs by going to the URL `https://domain-name:21443/` through your browser.

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

<table>
<thead>
<tr>
<th>21443/</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username: admin</td>
</tr>
<tr>
<td>Password: admin123</td>
</tr>
</tbody>
</table>

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

Visit the following ports once and accept the certificates:
https://domainName:20443/
https://domainName:30443/

For example:

https://demo.cisco.com:20443/
https://demo.cisco.com:30443/

If the Anomaly feature is enabled, also run the command on port 50443. For example,

https://demo.sanmateo.com:50443/
https://demo.sanmateo.com:50443/

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

**Updating Whitelists**

After running MURAL system for 2-3 hours, run the following command from master Namenode to generate updated whitelists:

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

Observe the categorization in UI after 2 hours to see the effects of whitelists update.
Setting Up Offline Reports

Uncategorized URL, UA, and TAC Reports

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

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

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

   For example,

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

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

2. Log into the master GCN Name node and navigate to the /data/work subdirectory:

   > en
   # _shell
   # cd /data
   # cd work

3. Create the serverFile_uncatReports file:

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

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

4. Create the same file on the standby GCN Namer node.

Tethering Databases

Create a file called serverFile_tethering with details of the ASR gateways, where the TAC, OS or UA databases, created as a result of tethering processing,
need to be pushed and offline_datatransfer_mapping_file with format
details of input and output files that will be transferred to ASR gateways.

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

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

Where:

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

2. Log in to the master GCN Namenode:

```
> en
# _shell
```

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

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

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

```
# cd work
# vi /data/work/serverFile_tethering

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

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

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

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

This file can have multiple rows of this kind.
The `offline_datatransfer_mapping_file` contains format for original filename and for modified filename that has to be shared with ASR gateway.

The format of entry in this file is as follows:

```
<Input filename regex>:<output filename>
```

Here,

- `<Input filename regex>` is the regex for matching input filename
- `<Output filename>` is the name of the shared file.

Perform the following steps:

1. Go to the data directory:
   ```
   # cd /data
   ``

2. Create `offline_datatransfer_mapping_file` with the following entries:
   ```
   # vi /data/offline_datatransfer_mapping_file
   os-db:new-os-db
   tac-db:new-tac-db
   ua-db:new-ua-db
   ```

   Note: The delimiter in the file must be ":".

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

**Tethering Reports**

Create a file called `serverFile_tethering_subscribers_report` with details of the destination server, where the subscriber reports created as a result of tethering processing, need to be pushed.

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

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

Where:

- `IP` is the destination host IP address
- `username` is the username for logging into the destination server
MURAL Software Standard Installation Guide

- *password* is the corresponding password to the destination server
- *location-to-copy-reports* is the location on the destination machine machine where databases need to be copied

Perform the following steps:

1. Log in to the master GCN Namenode:

   ```
   > en
   # _shell
   ```

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

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

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

   ```
   # cd work
   # vi /data/work/serverFile_tethering_subscribers_reports
   192.168.156.96, admin, admin@123, /data/tethering_subs
   ```

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

4. Create the same file on the standby GCN Namenode as well.

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

   This file can have multiple rows of this kind.

**APNTethering Reports**

Create a file called *serverFile_tethering_ApnDc_report* with details of the destination server, where the subscriber reports created as a result of tethering processing, need to be pushed.

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

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

Where:
IP is the destination host IP address

- **username** is the username for logging into the destination server
- **password** is the corresponding password to the destination server
- **location-to-copy-reports** is the location on the destination machine machine where databases need to be copied

Perform the following steps:

1. Log in to the master GCN Namenode:

   ```
   $ en
   # _shell
   ```

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

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

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

   ```
   # cd work
   # vi /data/work/serverFile_tethering_ApnDc_reports
   192.168.156.96, admin, admin@123, /data/tethering_apn
   ```

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

4. Create the same file on the standby GCN Namenode as well.

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

   This file can have multiple rows of this kind.
Mandatory Parameters for Incoming ASR Files

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

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

Mandatory Attributes for Flow EDRs for MURAL

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

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

**Sample:**

```
```
Mandatory HTTP EDR Attributes for MURAL

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

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

Sample:

```
```
ASR-Side Configuration

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

```plaintext
edr-format
edr-flow-format
attribute sn-start-time format seconds priority 10
attribute sn-end-time format seconds priority 20
attribute radius-calling-station-id priority 30
rule-variable bearer 3gpp imsi priority 35
attribute radius-called-station-id priority 40
attribute sn-volume-amt ip bytes uplink priority 50
attribute sn-volume-amt ip bytes downlink priority 60
attribute sn-volume-amt ip pkts uplink priority 70
attribute sn-volume-amt ip pkts downlink priority 80
rule-variable bearer 3gpp imei priority 90
rule-variable bearer 3gpp rat-type priority 100
rule-variable p2p protocol priority 110
attribute sn-app-protocol priority 120
attribute sn-parent-protocol priority 130
rule-variable ip protocol priority 140
rule-variable traffic-type priority 150
attribute sn-direction priority 160
rule-variable ip server-ip-address priority 170
attribute sn-server-port priority 180
rule-variable ip subscriber-ip-address priority 190
attribute sn-subscriber-port priority 200
rule-variable bearer 3gpp sgsn-address priority 210
rule-variable bearer ggsn-address priority 220
rule-variable bearer 3gpp user-location-information priority 230
rule-variable bearer 3gpp2 bsid priority 240
```
attribute sn-flow-start-time format seconds priority 260
attribute sn-flow-end-time format seconds priority 270
rule-variable tcp os-signature priority 290
rule-variable tethered priority 300
attribute sn-rulebase priority 310
#exit

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

Before you begin, ensure to complete the following:

- Configure Serial over LAN (SOL) on all the blades during EMC setup. For information about configuring SOL, refer to the MURAL Hardware Installation Guide.

- Locate your CIQ, and refer to it for details such as UCS access credentials and KVM SOL IP address.

To reboot the blades:

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

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

   All the blades available on the chassis are displayed.

2. Click the **Launch** button for the first node (Server1).

3. Click **OK** to download and open the `kvm.jnlp` file.

4. Click **OK** in the keyboard access warning message that is displayed.

5. Perform the following steps in the KVM Console that is displayed:

   a. Open the KVM Console of the blade.

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

   c. After the prompt, press **F12** as soon as possible to boot from the network.

      Once the blades start booting from the network, GMS pushes the image on all the blades using PXE boot for the manufacture process to start on each blade in parallel.
A blade takes approximately 30 minutes to manufacture with the new image. Wait until the blade for which PXE boot was issued has been manufactured. A login prompt is displayed once the image has been manufactured on a blade.

Go back to "Configuring MURAL Nodes" on page 27 to continue installing the MURAL system.