



# MURAL Software Installation Guide

*Version: 5.0.2.rc1*

**Last Updated:** 2019-04-04

**Copyright:** Copyright @ 2019, Cisco, Inc.

## **Americas Headquarters**

Cisco Systems, Inc.

170 West Tasman Drive



San Jose, CA 95134-1706 USA

<http://www.cisco.com>

Tel: 408526-4000

800553-NETS (6387)

Fax: 408527-0883

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

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

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

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

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE

USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

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

MURAL Software Installation Guide

Copyright © 2019, Cisco, Inc. All rights reserved.

## Table of Contents

---

<b>1. Introducing MURAL</b> .....	<b>1</b>
<b>2. Checklist to Installing MURAL</b> .....	<b>3</b>
<b>3. System Requirements</b> .....	<b>5</b>
<b>4. Extracting and Installing the Base Packages</b> .....	<b>9</b>
<b>5. Generating the Inventory File</b> .....	<b>13</b>
<b>6. Installing the Platform</b> .....	<b>23</b>
6.1 Prerequisites .....	24
6.2 Running the Platform Installer .....	27
6.3 Installing Platform Components for MURAL UI .....	30
6.4 Verifying the Platform Installation .....	33
<b>7. Installing the Solution</b> .....	<b>35</b>
7.1 Prerequisites .....	36
7.2 Refreshing the Inventory File .....	41
7.3 Running the Solution Installer .....	43
7.4 Verifying the Solution Installation .....	44
7.5 Scheduling Azkaban Jobs .....	46
7.6 Launching the User Interface .....	50
<b>8. Appendix A: Accessing User Interfaces on Multiple Interfaces</b> ..	<b>53</b>
<b>9. Appendix B: Updating Feature Variables</b> .....	<b>55</b>
<b>10. Appendix C: Mandatory Parameters for Incoming ASR Files</b> .....	<b>57</b>
10.1 Prerequisites .....	57
10.2 Mandatory Attributes for Flow EDRs for MURAL .....	57
10.3 Mandatory HTTP EDR Attributes MURAL .....	59

<b>11. Appendix D: Running the SFTP Downloader Installation .....</b>	<b>61</b>
<b>12. Appendix E: Understanding the Inventory File .....</b>	<b>65</b>
12.1 Updating the hosts file .....	65
<b>13. Appendix F: Changing Logo and Password on Azkaban UI .....</b>	<b>73</b>

## 1. Introducing MURAL

---

Cisco Mobility Unified Reporting and Analysis MURAL is a next-gen analytics solution, tailored for the Telecom industry. This next-gen solution is multidimensional and has been enhanced to provide visibility into each subscriber's behavior and usage. Hence; Network Operators can now discover customer intelligence for Marketing and Sales along with near-real-time insight into traffic patterns on their networks. It is intended to deliver valuable insights to the CSP organization using purposed built APIs, Connectors and MURAL User Interface.

This document is designed to guide you through the installation process, from installing the platform to installing the solution on your system. It covers the minimum system requirements, prerequisites that must be met before beginning with the installation. This also describes the verification, post-installation steps, and launching through the UI.



## 2. Checklist to Installing MURAL

MURAL installation is a seamless process that consists of two parts; platform and solution installation. We recommend that you follow the instructions in the same order as mentioned in the following table for a successful software installation and desired results.

Section	Task	Objective	Reference
Chapter 3	System Requirements	Detailed information about the software and hardware requirements.	"System Requirements" on page 5
Chapter 4	Extracting and Installing the Base Packages	Step-by-step instructions for installing and extracting the base packages	"Extracting and Installing the Base Packages" on page 9
Chapter 5	Generating the Inventory File	Step-by-step instructions for generating an inventory file on your system for saving packages.	"Generating the Inventory File" on page 13
<b>Installing Platform</b>			
Chapter 6.1	Prerequisites	Detailed information about the hosts file and network manager.	"Prerequisites" on page 24
Chapter 6.2	Running the Platform Installer	Step-by-step instructions for installing platform packages.	"Running the Platform Installer" on page 27
Chapter 6.3	Installing MURAL Components	Step-by-step instructions for installing the platform components for successful platform installation.	"Installing Platform Components for MURAL UI" on page 30
Chapter 6.4	Verifying the Platform Installation	Step-by-step instructions for verifying the status of the platform installation.	"Verifying the Platform Installation" on page 33
<b>Installing Solution</b>			

Section	Task	Objective	Reference
Chapter 7.1	Prerequisites	Detailed information about the minimum requirements or installing the solution.	"Prerequisites" on page 36
Chapter 7.2	Refreshing the Inventory File	Step-by-step instructions for refreshing the inventory files on your system.	"Refreshing the Inventory File" on page 41
Chapter 7.3	Running the Solution Installer	Step-by-step instructions for installing solution packages.	"Running the Solution Installer" on page 43
Chapter 7.4	Verifying the Solution Installation	Step-by-step instructions for verifying the status of the solution installation.	"Verifying the Solution Installation" on page 44
Chapter 7.5	Scheduling Azkaban Jobs	Step-by-step instructions for scheduling jobs on Azkaban UI.	"Scheduling Azkaban Jobs" on page 46
Chapter 7.6	Launching the UI	Step-by-step instructions for launching the user interface.	"Launching the User Interface" on page 50

### 3. System Requirements

---

Before installing the platform and solution on your system, refer to this section to maintain and meet the minimum system requirements.

In addition to minimum system requirement, review the following list to prepare for a successful installation:

You must ensure the following:

- Possess a working knowledge of Linux.
- Review MURAL v5.0.2.rc1 Release Notes.
- Installation package components are available. Refer to *Release Notes* provided by Cisco.
- 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.
- Inventory file with your network details is updated. For more information, see "Appendix E: Understanding the Inventory File" on page 65

Ensure your system meets the following basic and minimum requirements:

- **Network and Security**

MURAL software requires IPv4 networking.

**Note:** IPv6 is not supported. IPv6 must be disabled if already configured.

- **Disk Space**

The following table describes the primary purpose and minimal memory spaces required for each corresponding node:

Node Type	Mount Point	Size in GB	Primary Purpose
All Nodes	/	50	for OS
	/boot	2	Boot Space
	/var	150	for Containers
	/var/log	100	for Log Messages
Management Node	/opt/repos	200	for Storing RPM/ Packages

**Note:** Additional data disks may be required depending on the sizing of your solution.

Run the following command to check the disc space available on you system:

```
# df -h
```

The output may resemble as follows:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       50G   13G   38G   26% /
/dev/vda2       2048M 32M   2015M  2%  /boot
devtmpfs        21G    0    21G    0%  /dev
tmpfs           21G   54M   21G    1%  /dev/shm
tmpfs           21G   2.1G  19G   10%  /run
tmpfs           21G    0    21G    0%  /sys/fs/cgroup
/dev/vda3       150G   11G   140G    7%  /var
/dev/vda4       150G   11G   140G    7%  /var/logs
tmpfs           4.2G    0    4.2G    0%  /run/user/0
/dev/vdb        250G   78G   173G   31%  /opt/repos
tmpfs           4.2G    0    4.2G    0%  /run/user/26
```

- **Operating System**

Ensure your system is configured with CentOS 7.3 (minimal).

- **Configure Root User**

Ensure that root user is configured on all the nodes and bash shell is available.

- **Configure Host Network**

Based on your hardware selection, perform the following steps on all the nodes:

1. Configure the host name with domain. For example, `hostnamectl set-hostname <hostname.domain>`

**Note:** Host names must not contain uppercase.

2. Verify that correct host name is configured on the setup nodes. All hosts must have a fully working name resolution system. The hostnames must not be in uppercase or contain special characters. Duplicate entries in the hosts file are not supported. Aliasing in hostnames is not supported. Ensure the following:

The `hostname -f` command must return a fully qualified name.

The output may resemble as follows:

```
mural-mgt-01.gvs.ggn
```

3. Mount external storage (if any) for Hadoop and Kafka cluster. Refer to the following table for more information:

Node Type	Mount Point	Size in GB	Primary Purpose
All Data Nodes	/opt/data01...n	<Refer Sizing Sheet>	for Hadoop Cluster
	/opt/data02...n	<Refer Sizing Sheet>	for Kafka Cluster

4. Configure IP address and Gateway for all nodes as per CIQ sheet.

**Note:** Ensure that you configure the same network interface names on all the nodes.

- **MURAL Cluster Management**

The MURAL software requires a minimum of 8 separate nodes (physical machines or virtual machines) distributed as follows:

Node Type	Minimum Number	Minimum Recommendation
Management Node	1	4 cores and 36 GB RAM per node
Name Node	2	4 cores and 36 GB RAM per node
Loadbalancer Node	2	1 core and 3.5 GB RAM per node
Data Node	3	4 cores and 36 GB RAM per node

An inventory file must be filled in with details of each node under the corresponding host group. For more information, see "Appendix E: Understanding the Inventory File" on page 65

## 4. Extracting and Installing the Base Packages

---

Cisco provides the artefacts and provisioner archive that contains the base packages for installing the platform. After procuring the archive, extract the tar file, set up the repository, and install the base packages. Follow the procedure as mentioned in the following section to extract and install the base package components.

Perform the following steps:

1. Log onto the management node as a root user.
2. Run the following commands to create and change the directory to download and extract the software packages:

```
# mkdir /opt/repos
# mkdir /opt/mural_artefacts
# cd /opt/mural_artefacts
```

3. Download the platform artefacts as mentioned in Release Notes provided by Cisco. The archive contains the following:
  - **Artefacts**- The following compressed files contain all the platform artefacts:
    - repos-reflex-platform-5.4.1.tgz
    - mrx-rpms-5.6.2.rc1.tar.gz
    - mrx-docker-release-5.6.2.rc1-193.tar.gz
    - sftp-downloader-4.3.0-29.noarch.rpm
  - **Provisioner**- The following RPMs contains the platform provisioner:
    - reflex-provisioner-cisco-5.6.2-36.x86\_64.rpm
    - reflex-solution-provisioner-5.6.2.rc1-193.el7.centos.x86\_64.rpm
    - sftp-downloader-provisioner-4.3.0-30.tgz

You can either download artefacts and provisioner either from the Cisco ftp server on the management node or it will be provided by Cisco using a USB drive. Upload the software on the management node.

**Note:** Contact Cisco Support for FTP credentials.

4. Verify the MD5 of the files downloaded in preceding step. Run the following command:

```
md5sum <file-name>
```

For example:

```
# md5sum mrx-rpms-5.6.2.rc1.tar.gz
```

The output may resemble as follows:

```
5c6ebf2b8f62b62a50bb211e74e56060  mrx-docker-release-  
5.6.2.rc1-193.tar.gz  
01e8b4c2d9b9600c3aa25f4c1dbcd2eb  mrx-rpms-5.6.2.rc1.tar.gz  
df8e3343e934085cc3c3f330e6c0a837  reflex-provisioner-5.6.2-  
36.x86_64.rpm  
2a90cc7d4541bec73eab9997b4c3df48  reflex-solution-provisioner-  
5.6.2.rc1-203.el7.centos.x86_64.rpm  
8d03935bae471e5ef54e91998dff8e8e  repos-reflex-platform-  
5.4.1.tgz
```

5. Run the following command to install the reflex provisioner using yum installer:

```
yum install reflex-provisioner-cisco-5.6.2-36.x86_64.rpm
```

6. Extract the base repository package tar file in /opt/repos/ folder:

```
tar -zxvf repos-reflex-platform-5.4.1.tgz -C /opt/repos
```

7. Extract the solution repository package tar file in /opt/repos/ folder:

```
tar -zxvf mrx-rpms-5.6.2.rc1.tar.gz -C /opt/repos/
```

8. Run the following command to create a directory:

```
mkdir -p /opt/repos/mrx/5.6/mrx-docker-5.6.2.rc1
```

9. Extract the docker files in the created directory:

```
tar -zxvf mrx-docker-release-5.6.2.rc1-193.tar.gz -C
/opt/repos/mrx/5.6/mrx-docker-5.6.2.rc1/
```

10. Copy `sftp-downloader-4.3.0-29.noarch.rpm` file in the `/opt/repos/gvs-platform-prod/release/5.3.0/` directory:

```
cp sftp-downloader-4.3.0-29.noarch.rpm /opt/repos/gvs-
platform-prod/release/5.3.0/
```

11. Run the following commands to extract the SFTP provisioner packages:

```
tar -zxvf sftp-downloader-provisioner-4.3.0-30.tgz
mv reflex-provisioner /etc/reflex-provisioner/work_dir/sftp-
downloader-provisioner
```

12. Run the following command to execute the Bootstrap Provisioner script in `/etc/reflex-provisioner/` file:

```
cd /etc/reflex-provisioner/
# sh ./bootstrap.sh prod
```

The output may resemble as follows:

```
Command "python setup.py egg_info" failed with error code 1 in
/tmp/pip-build-SrmXcF/pynacl/
```

```
=====
REFLEX-PROVISIONER BOOTSTRAP COMPLETE
```

```
|
```

```
=====
```

The preceding command may fail due to online RPM package installation.

Run the following command manually to install the following two packages:

```
pip install /etc/reflex-provisioner/packages/pip/pycparser-
2.18-py2.py3-none-any.whl
```

```
pip install /etc/reflex-provisioner/packages/pip/cffi-  
1.11.5.tar.gz
```

Run the following commands to re-execute the Bootstrap Provisioner script in `/etc/reflex-provisioner/` file:

```
# sh ./bootstrap.sh prod  
# pip install /opt/repos/pypi-server/packages/srx-pypi-repo-  
5.3.0.prod/jmespath-0.9.3-py2.py3-none-any.whl
```

13. Run the following command to install git on your system:

```
# yum install git
```

14. Run the following script to create json inventory files on your system:

```
# sh ./scripts/composition/create.sh -i mural -c cisco -s prod  
-m cisco
```

15. Switch to `work_dir/reflex-configuration-module/conf/generate_inventory/conf_inventory/prod/mural/` directory and run the following command to create a `extra_vars` directory:

```
# cd work_dir/reflex-configuration-module/conf/generate_  
inventory/conf_inventory/prod/mural/  
# mkdir extra_vars
```

## 5. Generating the Inventory File

---

Set up an inventory to include details about all the nodes and their roles in your setup. For more information, refer to the sample file in "Appendix E: Understanding the Inventory File" on page 65

Perform the following steps:

1. Update the `hosts` file that reflects your setup at `/etc/reflex-provisioner/work_dir/reflex-configuration-module/conf/generate_inventory/conf_inventory/prod/mural/`

**Notes:**

- It is mandatory to update `hosts` with domain for all the available `hosts` groups in the `hosts` file.
- Host names in the `hosts` file must be same as those configured on the setup nodes.
- The `app nodes` entries must be removed from `[platform-nodes]` and `[app nodes]` host group and replaced with master nodes in `[platform-node]` host group.

Update the following entry for every host group in the `hosts` file:

```
[Host group]
<hostname.domain of the node> ansible_ssh_host=<IP address of
the node>
```

**Notes:**

- Ensure that you configure the same host names in the `hosts` file as the host names configured on the setup nodes.
  - Ensure to keep only the default `hosts` file at the default inventory directory. There must be no backup file in it.
2. Update `generate_inventory.json` file to set the following site-specific properties:

```
/etc/reflex-provisioner/work_dir/reflex-configuration-
module/conf/generate_inventory/conf_
inventory/prod/mural/generate_inventory.json
```

Here,

- `data_mountpoint` are the mount points for Hadoop cluster of the external block devices mounted as file system into data nodes.
- `domain_name` is the domain name. It should be same as hostnames.
- `dns_server` is the DNS server. It should be same as nameserver in `/etc/resolv/conf`.
- `floating_ip_addr` is the floating VIP from private network.
- `floating_ip_cidr` is the network prefix of the floating VIP from private network.
- `floating_ip_fqdn` is the host name of floating VIP.
- `floating_ip_inet` is the interface of the floating VIP from private network.
- `nb_app` is the number of application nodes. Set it as 0.
- `ntp_disable` is the disable ntp server.
- `ntp_server` is the ntp server IP.
- `support_email` is the email id for report delivery.
- `smtp_server` is the customer or site mail server IP.

The output may resemble as follows:

```
{
  "--connectivity": "offline",
  "--customer_name": "<customer_name>",
  "--data_mountpoint": [
    "/opt/hdfs01"
  ],
  "--deployment_manifest": "<deployment_manifest>",
  "--deployment_type": "HA",
  "--dns_server": [
    "172.30.12.101"
  ],
  "--domain_name": "us.guavus.com",
```

## MURAL Software Installation Guide

```
"--extra_vars_file": null,  
"--floating_ip_addr": "10.30.3.73",  
"--floating_ip_cidr": "8",  
"--floating_ip_fqdn": "lv-vip-fqdn-vip.us.guavus.com",  
"--floating_ip_inet": "<interface_name>",  
"--floating_ip_is_internal": "yes",  
"--inventory_name": "mural",  
"--nb_app": "0",  
"--nb_etc": null,  
"--nb_hdfs_jrnl": null,  
"--nb_hdfs_snn": null,  
"--nb_imp_slv": null,  
"--nb_imp_slv_pool1": "3",  
"--nb_imp_slv_pool2": "2",  
"--nb_imp_slv_pool3": "2",  
"--nb_imp_slv_pool4": "2",  
"--nb_imp_slv_pool5": "3",  
"--nb_kfk_brk": null,  
"--nb_redis": null,  
"--nb_slv": "3",  
"--nb_zk": null,  
"--ntp_disable": false,  
"--ntp_server": [  
"172.30.12.100"  
],  
"--repository_host_fqdn": "{{ groups['management-nodes'] [0]  
}}",  
"--smtp_disable": false,  
"--smtp_server": [  
"172.30.12.102"  
],  
"--stage_type": "prod",  
"--support_email": [  
"mtl-is-devops@guavus.mtl"
```

```
]
}
```

3. Create `platform.yml` file at `/etc/reflex-provisioner/work_dir/reflex-configuration-module/conf/generate_inventory/conf_inventory/prod/mural/extra_vars/` directory and add the following lines in the file:

- **for single interface**

```
#-----
-----
## set yarn resource allocation parameters
##-----
-----
platform_hadoop_yarn_scheduler_max_mem: "205619"
platform_hadoop_yarn_nodemanager_resource_mem: "205619"
##-----
-----
## set separate disk mount points for kafka
##-----
-----
platform_kafka_disk_mount_points:
- '/opt/kafka01'
platform_hadoop_yarn_nodemanager_resource_cpu_vcores:
'25'
platform_hadoop_yarn_nodemanager_resource_cpu_vcores_
percent: '1'
```

- **for multiple interfaces**

```
#-----
-----
## set yarn resource allocation parameters
##-----
```

```
-----
platform_hadoop_yarn_scheduler_max_mem: "205619"
platform_hadoop_yarn_nodemanager_resource_mem: "205619"
##-----
-----
## set separate disk mount points for kafka
##-----
-----
platform_kafka_disk_mount_points:
- '/opt/kafka01'
platform_hadoop_yarn_nodemanager_resource_cpu_vcores:
'25'
platform_hadoop_yarn_nodemanager_resource_cpu_vcores_
percent: '1'
azkaban_haproxy_listen__vip: "{{ platform_loadbalancer_
vip_02 }}"
azkaban_haproxy_listen__servers_interface: "{{ interface_
name_public }}"
```

Here,

- `platform_hadoop_yarn_scheduler_max_mem` and `platform_hadoop_yarn_nodemanager_resource_mem` requires eighty percentile RAM (size in MB) on one of the data nodes.
  - `platform_hadoop_yarn_nodemanager_resource_cpu_vcores` requires eighty percentile of VCores (integer value) on one of the data nodes.
  - `platform_kafka_disk_mount_points` is the mount point for Kafka cluster of the external block device mounted as file system into data nodes.
4. Create `solution.yml` file at `/etc/reflex-provisioner/work_dir/reflex-configuration-module/conf/generate_inventory/conf_inventory/prod/mural/extra_vars/` directory.

The output may resemble as follows:

- **for single interface**

```
# Customer Variable
vendor: <customer>
source_file_mask: "*.gz"
# # Features
flexi_past_open_bin: 48
flexi_future_open_bins: 12
heavyusers_enabled: true
dly_topsubscriberscount: 1000
writeUncat: true
percentageOfUncatEmit: 10
odsClassification: 'false'
#### KFPS setting ####
input_kfps: 5
http_kfps_percentage: 5
nonhttp_kfps_percentage: 95
ingestion_kfps_per_core: 10
ddj_kfps_per_core: 3
dme_kfps_per_core: 3
agg_5min_kfps_per_core: 6
agg_1hour_kfps_per_core: 6
agg_1day_kfps_per_core: 10
agg_1month_kfps_per_core: 10
ingestion_ram_per_core: 2
ddj_ram_per_core: 2
dme_ram_per_core: 2
agg_5min_ram_per_core: 2
agg_1hour_ram_per_core: 2
agg_1day_ram_per_core: 2
agg_1month_ram_per_core: 2
cores_per_executor: 2
driver_memory: 2g
overhead_memory: 512
```

- **for multiple interfaces**

```
# Customer Variable
vendor: <customer>
source_file_mask: "*.gz"
deploy_multipleInterface: true
platform_loadbalancer_vip_02: "172.30.3.73"
interface_name_public: "eno1"
broadcast_mask_public: 17
# # Features
flexi_past_open_bin: 48
flexi_future_open_bins: 12
heavyusers_enabled: true
dly_topsubscriberscount: 1000
writeUncat: true
percentageOfUncatEmit: 10
odsClassification: 'false'
#### KFPS setting ####
input_kfps: 5
http_kfps_percentage: 5
nonhttp_kfps_percentage: 95
ingestion_kfps_per_core: 10
ddj_kfps_per_core: 3
dme_kfps_per_core: 3
agg_5min_kfps_per_core: 6
agg_1hour_kfps_per_core: 6
agg_1day_kfps_per_core: 10
agg_1month_kfps_per_core: 10
ingestion_ram_per_core: 2
ddj_ram_per_core: 2
dme_ram_per_core: 2
agg_5min_ram_per_core: 2
agg_1hour_ram_per_core: 2
agg_1day_ram_per_core: 2
agg_1month_ram_per_core: 2
```

```
cores_per_executor: 2
driver_memory: 2g
overhead_memory: 512
```

Here,

- `platform_loadbalancer_vip_02` is the external VIP for loadbalancer nodes.
- `interface_name_public` is the network port used for external IP for loadbalancer nodes.
- `broadcast_mask_public` is the network prefix (CIDR) used for external IP for loadbalancer nodes.

**Note:** The default setting is for processing upto 5kfps input data load (`input_kfps: 5`). Change the values in case of higher KFPS rate at production setups. Contact Support Team for appropriate value settings.

Refer "Appendix B: Updating Feature Variables" on page 55 to modify the value as required.

5. Edit `/etc/reflex-provisioner/work_dir/sftp-downloader-provisioner/playbooks/mrx_sftp-downloader.yml` file and edit the `inventory-name`

Replace `mural001` with `mural` to view the following output:

```
vars_files:
- /etc/reflex-
provisioner/inventory/generated/prod/mural/group_
vars/all/mrx/downloader.yml
```

6. Edit `/etc/reflex-provisioner/work_dir/sftp-downloader-provisioner/inventory/templates/group_vars/-global/all/mrx/downloader.yml` file to set the GGSN IP and EDR source path.
7. Update `etc/reflex-provisioner/work_dir/reflex-configuration-module/conf/generate_inventory/conf_`

inventory/prod/mural/manifest.yml file. Add the following lines at the end of the file:

```
name: sftp-downloader-provisioner
```

8. Run the following command to refresh the inventory in /etc/reflex-provisioner file:

```
# export GIT_PYTHON_REFRESH=quiet  
# ./scripts/composition/refresh.sh -i mural -s prod
```



## **6. Installing the Platform**

---

MURAL platform is a highly scalable and feature-rich distributed platform that is useful for developing and deploying applications conveniently. This section provides the information about the prerequisites and steps for installing the platform on your system.

## 6.1 Prerequisites

Complete all the tasks as mentioned in this section prior to installing the platform.

### 6.1.1 Update the Hosts file

Ensure that you update `/etc/hosts` file on the management nodes by adding 'IP address and hostname' mapping for all the nodes.

Run the following command to update the `/etc/hosts` file:

```
# vi /etc/hosts
```

**Note:** Use internal IPs `/etc/hosts` file in case of multiple interface.

The following sample illustrates the mapping for all the nodes in the setup:

```
127.0.0.1 localhost localhost.localdomain
::1 localhost6 localhost6.localdomain
172.16.137.2 mural-mgt-01.gvs.ggn mural-mgt-01
172.16.137.3 mural-mst-01.gvs.ggn mural-mst-01
172.16.137.4 mural-mst-02.gvs.ggn mural-mst-02
172.16.137.5 mural-slv-01.gvs.ggn mural-slv-01
172.16.137.6 mural-slv-02.gvs.ggn mural-slv-02
172.16.137.7 mural-slv-03.gvs.ggn mural-slv-03
172.16.137.8 mural-lb-01.gvs.ggn mural-lb-01
172.16.137.9 mural-lb-02.gvs.ggn mural-lb-02
172.16.137.10 mural-lb-vip.gvs.ggn mural-lb-vip
```

Run the following command to view the list of nodes on your system:

```
# cat /etc/hosts
```

**Note:** Refer CIQ sheet for IP and hostname details.

### 6.1.2 Disable NetworkManager

If NetworkManager is disabled on your setup, the repository will keep updating and will not provide a consistent environment for installation. Therefore, run the following commands from the console for all the nodes to disable NetworkManager:

```
# systemctl stop NetworkManager
# systemctl disable NetworkManager
```

### 6.1.3 Cleaning Old Repositories

Ensure that you delete existing repositories from all the nodes to avoid version compatibility mismatch.

Run the following command to delete existing repository files from all the nodes to delete all the repository files present in `/etc/yum.repos.d/` file:

```
# rm -f *.repo
```

### 6.1.4 Editing the Bash File

Ensure that you perform the following steps on all the nodes prior to platform installation:

1. Log onto the management node as a root user.
2. Add the following lines in the end of the `~/.bashrc` file to display correct UTF-8 and handle non-ASCII characters in programs:

```
export LANG=en_US.UTF-8
export LANGUAGE=en_US.UTF-8
export LC_ALL=en_US.UTF-8
```

3. Run the following command to export all the variables mentioned in `.bashrc` file:

```
source ~/.bashrc
```

4. Run the following command to verify locale settings on your system:

---

```
# locale
```

---

The output may resemble as follows:

```
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=en_US.UTF-8
```

## 6.2 Running the Platform Installer

After downloading the base packages and provisioner on your system, you can now run the Ansible playbook to install the platform.

Perform the following steps:

1. Log onto the management node as a root user.
2. Update the maximum days parameter to 90 in `/etc/reflex-provisioner/roles/grafana/templates/conf/grafana.ini.j2` file.

The output resembles as follows:

```
max_days = 90
```

3. Set the following grafana with graphite variable properties in `/etc/reflex-provisioner/roles/grafana/defaults/main.yml` file as follows:

```
grafana_with_graphite: 'yes'
```

4. Run the following command to switch to `/etc/reflex-provisioner` directory and run the Ansible playbook:

```
# cd /etc/reflex-provisioner
# ansible-playbook -i inventory/generated/prod/mural/hosts
playbooks/bootstrap/main.yml -k
# ansible-playbook -i inventory/generated/prod/mural/hosts
playbooks/platform/main.yml -k
```

Please note, depending on your network, the system may take up to two hours to complete the installation.

Upon successful execution of playbook, the output may resemble as follows:

```
PLAY RECAP
```

```

*****
*****
localhost                : ok=57   changed=0
unreachable=0   failed=0
mural-lb-01.gvs.ggn      : ok=257  changed=125  unreachable=0
  failed=0
mural-lb-02.gvs.ggn      : ok=258  changed=125  unreachable=0
  failed=0
mural-mgt-01.gvs.ggn     : ok=689  changed=272  unreachable=0
  failed=0
mural-mst-01.gvs.ggn     : ok=327  changed=178  unreachable=0
  failed=0
mural-mst-02.gvs.ggn     : ok=277  changed=138  unreachable=0
  failed=0
mural-slv-01.gvs.ggn     : ok=293  changed=154  unreachable=0
  failed=0
mural-slv-02.gvs.ggn     : ok=294  changed=152  unreachable=0
  failed=0
mural-slv-03.gvs.ggn     : ok=293  changed=152  unreachable=0
  failed=0

```

**Note:** Ensure the value is `failed=0` in every case for a successful platform installation.

- Run the following commands to install the Kubernetes after restarting the docker:

```

# ansible master-nodes -i /etc/reflex-
provisioner/inventory/generated/prod/mural/hosts -m shell -a
"systemctl restart docker" -k
# ansible-playbook -i inventory/generated/prod/mural/hosts
playbooks/kubernetes/main.yml -k

```

- Enter the root password to set for all the nodes.

The output may resemble as follows:

## MURAL Software Installation Guide

```
PLAY RECAP
*****
*****
localhost                : ok=57   changed=0
unreachable=0   failed=0
mural-lb-01.gvs.ggn      : ok=257  changed=125  unreachable=0
failed=0
mural-lb-02.gvs.ggn      : ok=258  changed=125  unreachable=0
failed=0
mural-mgt-01.gvs.ggn     : ok=689  changed=272  unreachable=0
failed=0
mural-mst-01.gvs.ggn     : ok=327  changed=178  unreachable=0
failed=0
mural-mst-02.gvs.ggn     : ok=277  changed=138  unreachable=0
failed=0
mural-slv-01.gvs.ggn     : ok=293  changed=154  unreachable=0
failed=0
mural-slv-02.gvs.ggn     : ok=294  changed=152  unreachable=0
failed=0
mural-slv-03.gvs.ggn     : ok=293  changed=152  unreachable=0
failed=0
```

**Note:** In case the platform installer fails and shows the value as `failed=1`, repeat the step 5. The value must show as `failed=0` in every case of successful kubernetes installation.

## 6.3 Installing Platform Components for MURAL UI

After installing the platform successfully, you are required to install the following platform components to enable UI access:

- IB Framework
- OAuth
- User Management

This section describes the procedure for installing the MURAL components.

### 6.3.1 Installing IB Framework

This feature maintains the databases and information base (IB) files for the solution. It is mandatory to enable ansible playbook to have these files on your system.

Perform the following steps to install IB Framework:

1. Log onto the management node as a root user.
2. Switch to `/etc/reflex-provisioner` directory and run ansible playbook:

```
cd /etc/reflex-provisioner
ansible-playbook -i inventory/generated/prod/mural/hosts
playbooks/ibmanagement/main.yml -k
```

### 6.3.2 Installing OAuth

This feature creates the database and configurations files for authentication of users and services. It is mandatory to enable ansible playbook for a successful OAuth installation on your system.

Perform the following steps to install OAuth:

1. Log onto the management node as a root user.
2. Switch to `/etc/reflex-provisioner` directory and run ansible playbook:

```
cd /etc/reflex-provisioner
ansible-playbook -i inventory/generated/prod/mural/hosts
playbooks/oauth/main.yml -k
```

### 6.3.3 Installing User Management

This enables you to view and access the User Management features and functions on the MURAL UI. It is mandatory to enable ansible playbook for a successful User Management installation on your system.

Perform the following steps to install User Management:

1. Log onto the management node as a root user.
2. Switch to `/etc/reflex-provisioner` directory and run ansible playbook:

```
cd /etc/reflex-provisioner
ansible-playbook -i inventory/generated/prod/mural/hosts
playbooks/usermanagement/main.yml -k
```

The output may resemble as follows for all the three components:

```
PLAY RECAP
*****
**
localhost                : ok=57   changed=0   unreachable=0
failed=0
mrx005-lb-01.gvs.ggn     : ok=8   changed=0   unreachable=0
failed=0
mrx005-lb-02.gvs.ggn     : ok=8   changed=0   unreachable=0
failed=0
mrx005-mgt-01.gvs.ggn    : ok=32  changed=8   unreachable=0
failed=0
mrx005-mst-01.gvs.ggn    : ok=9   changed=5   unreachable=0
failed=0
mrx005-mst-02.gvs.ggn    : ok=9   changed=4   unreachable=0
failed=0
mrx005-slv-01.gvs.ggn    : ok=3   changed=0   unreachable=0
```

```
failed=0
mrx005-slv-02.gvs.ggn      : ok=3    changed=0    unreachable=0
failed=0
mrx005-slv-03.gvs.ggn      : ok=3    changed=0    unreachable=0
failed=0
```

## 6.4 Verifying the Platform Installation

This section provides instructions to verify the successful installation of MURAL components on your system.

Perform the following steps:

1. Log onto the management node as a root user.
2. Run the following command to view the status of the containers through all the nodes:

```
kubectl get pods | grep -E
'azkaban|editservice|oauth|usermanagement'
```

**Note:** You may need to wait for few minutes to view the status of the containers.

The output may resemble as follows:

NAME	READY	STATUS
azkaban-platform-1060947321-glxvs	1/1	
Running 0 5d		
editservice-3088048493-90nm9	2/2	
Running 0 10m		
oauth-deployment-3981525391-f8tn4	2/2	
Running 0 5m		
usermanagement-deployment-1078286343-95ggz	3/3	
Running 0 2m		

3. Switch to `cd /etc/reflex-provisioner` directory and run the following command to check the status of different services running on all the nodes:

```
cd /etc/reflex-provisioner
ansible-playbook -i inventory/generated/prod/mural/hosts
playbooks/platform/service_checks/all.yml -k
```

**Note:** Ensure that playbook is executed successfully with `failed=0` counters for all the nodes.

**The platform installation is completed successfully.**

## **7. Installing the Solution**

---

MURAL solution provides a greater understanding of subscribers' interests and near-real-time insight into traffic patterns on their networks. With its rich and interactive user interface, you can solve various business use-cases, and increase upsell and cross-sell opportunities. This section provides information about the prerequisites and steps for installing the solution on your system.

## 7.1 Prerequisites

Complete all the tasks as mentioned in this section prior to installing the solution.

### 7.1.1 Verify Platform and Other Nodes Status

Ensure the following:

- MURAL Platform 5.4.1 is successfully installed on your system.
- All the nodes and processes are up and running.
- All repository files must be available on the management node and system admin must disconnect from internet.
- YARN resources are available as per the production setup configuration with the minimum availability of 75 percentile.

Refer "Verifying the Platform Installation" on page 33 to know the status of platform and other services on your system.

### 7.1.2 Verify Yum Repository Status

Before installing the solution on your system, ensure that Yum repository is up and functional from management node.

Run the following command to check the status of the Yum repository on your system:

```
# yum repolist all
```

The output may resemble as follows:

```
[root@mural-1 yum.repos.d]# yum repolist all
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
repo id                repo name                status
!base                  CentOS-7 - Base         enabled:
```

## MURAL Software Installation Guide

```
9,363
!cdh                                Cdh repository for Centos7 - x86_64
                                     enabled:

147
!centos-7-security-updates Packages with security updates not
backported into official repos
enabled:          9
!confluent        Confluent repository for Centos7 -
x86_64
enabled:          76
!docker           Docker repository for Centos7 - x86_64
                                     enabled:

65
!epel             Extra Packages for Enterprise Linux 7
- x86_64          enabled:
11,221
!mrx              MRX Solution Repo
                                     enabled:

33
!platform        Gvs Platform - x86_64
                                     enabled:

10
!postgres        Postgresql repository for Centos7 -
x86_64
enabled:          10
!scl              CentOS-7 - SCL
                                     enabled:

5,545
!sp-base/x86_64  CentOS-7 - Base
                                     enabled:

9,363
!sp-epel         Extra Packages for Enterprise Linux 7
- x86_64         enabled:
11,221
```

```

!sp-postgres          Postgresql repository for Centos7 -
x86_64
enabled:      373
!sp-updates/x86_64    CentOS-7 - Updates
                                enabled:
1,851
!updates             CentOS-7 - Updates
                                enabled:
1,851
!zabbix              Zabbix repository for Centos7 - x86_64
                                enabled:
79
repolist: 51,217
[root@mural-1 yum.repos.d]#

```

### 7.1.3 Verify Postgres Status

Before installing the solution on your system, ensure that Postgres is up and running on all the three nodes; master namenode, standby namenode and management namenode.

Run the following command to check the status of the Postgres on your system:

```
crm_mon -Afr1
```

The output may resemble as follows:

```

[root@mural-01 ~]# crm_mon -Afr1
Stack: corosync
Current DC: mural-mst-02.guavus.com (version 1.1.15-11.e17_3.4-
e174ec8) - partition with quorum
Last updated: Tue Oct 9 10:46:22 2018           Last change: Mon
Oct 1 10:54:46 2018 by root via crm_attribute on mural-mst
01.guavus.com
5 nodes and 7 resources configured
Online: [ mural001-lb-01.guavus.com mural001-lb-02.guavus.com

```

## MURAL Software Installation Guide

```
mural001-mgt-01.guavus.com mural001-mst-01.guavus.com mural001
mst-02.guavus.com ]
Full list of resources:
Clone Set: haproxy-clone [haproxy]
Started: [ mural001-lb-01.guavus.com mural001-lb-02.guavus.com ]
vip-haproxy (ocf::heartbeat:IPAddr2): Started mural001-lb-
01.guavus.com
Resource Group: pgsq-ports-group
pgsq-master-port (ocf::guavus:iptables): Started mural001-mst-
01.guavus.com
Master/Slave Set: pgsq-master-slave [pgsq]
Masters: [ mural001-mst-01.guavus.com ]
Slaves: [ mural001-mgt-01.guavus.com mural001-mst-02.guavus.com ]
Node Attributes:
* Node mural001-lb-01.guavus.com:
* Node mural001-lb-02.guavus.com:
* Node mural001-mgt-01.guavus.com:
+ master-pgsq : 10
+ pgsq-data-status : STREAMING|POTENTIAL
+ pgsq-receiver-status : normal
+ pgsq-status : HS:potential
+ pgsq-xlog-loc : 000000008DC0B508
* Node mural001-mst-01.guavus.com:
+ master-pgsq : 1000
+ pgsq-data-status : LATEST
+ pgsq-master-baseline : 0000000005000060
+ pgsq-receiver-status : normal (master)
+ pgsq-status : PRI
+ pgsq-xlog-loc : 000000008DC0B508
* Node mural001-mst-02.guavus.com:
+ master-pgsq : 100
+ pgsq-data-status : STREAMING|SYNC
+ pgsq-receiver-status : normal
+ pgsq-status : HS:sync
```

```
+ pgsq1-xlog-loc : 000000008DC0B508
```

```
Migration Summary:
```

```
* Node mural001-mgt-01.guavus.com:
```

```
* Node mural001-mst-02.guavus.com:
```

```
* Node mural001-mst-01.guavus.com:
```

```
* Node mural001-lb-01.guavus.com:
```

```
* Node mural001-lb-02.guavus.com:
```

```
[root@mural001-mgt-01 ~]#
```

**Note:** Ensure that no action fails.

## 7.2 Refreshing the Inventory File

This section states the steps to refresh the inventory for a successful solution installation.

For more information on how to generate an inventory file, refer to "Generating the Inventory File" on page 13

Perform the following steps in the same sequence as mentioned to run the solution ansible script successfully:

1. Log onto the management node as a root user.
2. Run the following command to install yum:

```
yum install /opt/repos/mrx/5.6/5.6.2.rc1/reflex-solution-provisioner-5.6.2.rc1-193.el7.centos.x86_64.rpm
```

3. Update `/etc/reflex-provisioner/work_dir/reflex-configuration-module/conf/generate_inventory/conf_inventory/prod/mural/manifest.yml` file. Add the following lines at the end to update the file:

```
- name:
  reflex-solution-provisioner
  uri:
  Guavus/reflex-solution-provisioner.git
  revision:
  mrx_solution_improvements
```

4. Run the following command to refresh the inventory in `/etc/reflex-provisioner` file:

```
export GIT_PYTHON_REFRESH=quiet
./scripts/composition/refresh.sh -i mural -s prod
```

The output may resemble as follows:

```
-i mural was triggered!
-s prod was triggered!
```

Refreshing init inventory

Refreshing mural inventory

For more information on how to update an inventory file, refer to "Appendix E: Understanding the Inventory File" on page 65

## 7.3 Running the Solution Installer

This section states the steps to run the solution installer on your system.

Perform the following steps:

1. Log onto the management node as a root user.
2. Run the following command to switch to `/etc/reflex-provisioner/` directory:

```
cd /etc/reflex-provisioner/
```

3. Run the following command to install the solution through ansible playbook:

```
ansible-playbook -i inventory/generated/prod/mural/hosts  
playbooks/mrx/deploy.yml -k
```

A message will be shown during the ansible playbook run. The following section illustrates the sample output:

```
Enter initialization Key of size 32 bytes(use this default  
value if key not available ->  
12345678901234567890123456789012) or press ENTER if encryption  
not required:  
Enter Vector Key of size 16 bytes(use this default value if  
key not available -> 1234567890123456) or press ENTER if  
encryption not required:
```

**Note:** If PII data encryption is required, set the customer provided key and vector ID or default values as described in the preceding sample output.

4. Run the following command to configure HAProxy through haproxy ansible playbook:

```
ansible-playbook -i inventory/generated/prod/mural/hosts  
playbooks/mrx/haproxy/main.yml -k
```

## 7.4 Verifying the Solution Installation

This section states the procedure to verify the successful installation of MURAL solution on your system.

Perform the following steps:

1. Log onto the management node as a root user.
2. Run the following command to check the status of MURAL UI containers running on all the nodes:

```
# kubectl get pods | grep -E 'NAME|tomcat'
```

The output may resemble as follows:

NAME	READY	STATUS	RESTARTS
AGE			
tomcat-mrxui-1223023554-d3zhp	2/2	Running	0
17m			
tomcat-mrxui-1223023554-wvxcl	2/2	Running	0
17m			

3. Add the external *<LB-VIP>* and *<LB-VIP-Hostname.Domainname>* mapping into `/etc/hosts` file either into DNS or `/etc/hosts` file of your machine used to access the UI applications. Please refer CIQ sheet for *<LB-VIP>* and *<LB-VIP-Hostanem.Domainname>* values.

For example:

```
172.30.3.73 lb-vip.us.guavus.com lb-vip
```

4. Enter the following URL in web browser to know the version of the OAuth:

```
http://<LB-URL>/oauth/api/v1/version
```

The output may resemble as follows:

```
{"versions":{"USER_MANAGEMENT":
{"v":"1","buildNumber":"385","buildTime":"2018-12-
```

## MURAL Software Installation Guide

```
17T17:00:30Z", "buildBy": "root", "buildJdk": "1.8.0_131", "applicationVersion": "3.0.0-SNAPSHOT"}, "AUTHORIZE": {"v": "1", "buildNumber": "451", "buildTime": "2018-12-17T18:07:00Z", "buildBy": "root", "buildJdk": "1.8.0_131", "applicationVersion": "2.0.0-SNAPSHOT"}}}
```

5. Enter the following URL in web browser to verify the login screen of the User Management UI:

```
http://<LB-URL>/usermanagement/
```

6. Enter the following URL in web browser to verify the login screen of the Grafana UI:

```
http://<management-ip>:5080/
```

## 7.5 Scheduling Azkaban Jobs

You are required to schedule jobs after successful installation of the solution on your system. This section describes the procedure for scheduling of Azkaban jobs.

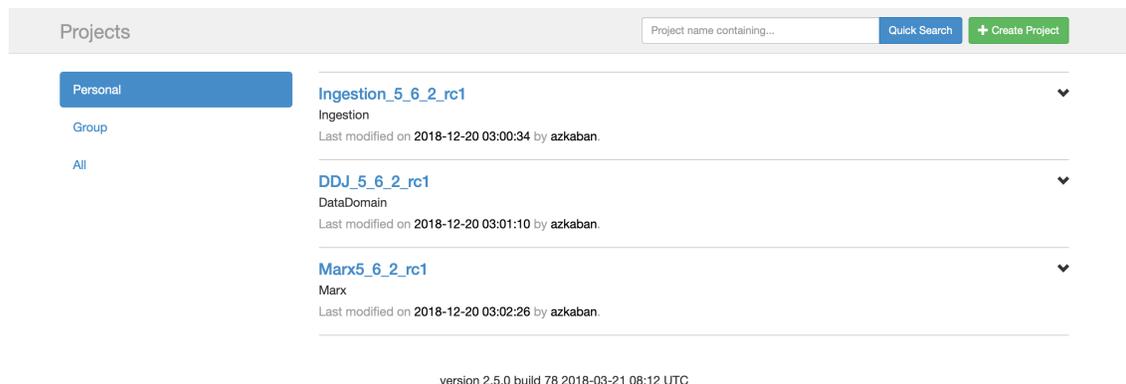
Before launching the Azkaban UI, ensure you update the following configuration files on both the namenodes and all the data nodes:

- in `/opt/mrx/ingestion/etc/extract.conf` file:  
 Replace `source_file_mask = *.gz` with `source_file_mask = *http*.gz`
- in `/opt/mrx/ingestion/etc2/extract.conf` file:
  - Replace `source_file_mask = *.gz` with `source_file_mask = *flow*.gz`
  - Replace `root_dir = /user/mrx/ingestion2/` with `root_dir = /user/mrx/ingestion/`

Perform the following steps to launch Azkaban UI:

1. Enter URL: `http://lb-vip:8507/` in your web browser.
2. Log into the Azkaban UI as username: **azkaban** and password: **!4zk4b4n\$**

The Azkaban UI is launched. The following image illustrates the landing page for Azkaban UI:



### 7.5.1 Scheduling Jobs

Before scheduling the jobs, you need to change the logo and password on Azkaban UI. For more information, refer "Appendix F: Changing Logo and Password on Azkaban UI" on page 73

This section states the steps to schedule the jobs on Azkaban UI.

#### Prerequisites

Before you schedule the jobs, you require to update a command for `agg_1day-dimensionTable`.

Read the following section to know how to update the job's command:

1. Log into the Azkaban UI.
2. Click **Marx-5\_6\_2\_rc1** from the list of jobs appearing on the screen.
3. Scroll down and click the drop-down arrow left to **agg\_1day-dimensionTable** job.
4. Click **agg\_1day-dimensionTable** job.
5. Click **Edit** button on the right side of the table on the screen.
6. In the **command** section, search for `queue jobs.hourly` and replace it with `queue default`.
7. Click **Set/Change Job Description** to save the changes.

Perform the following steps to schedule a job via Azkaban UI.

1. Log into the Azkaban UI from your web browser.
2. Click **Projects** tab from top-navigation bar.
3. Click **Ingestion-5\_6\_2\_rc1** from the list of jobs displayed on the screen.
4. Click **Execute Flow** button corresponding to the job.
5. Click **Schedule** on the dialogue box which appears next.
6. Enter the values in the **Schedule Flow Options** dialogue box as desired.

- Click **Schedule** button to schedule the job or **Cancel** to discard the function.

The job is scheduled.

Similarly, schedule in the following jobs in the same sequence as mentioned.

- Ingestion-5\_6\_2\_rc1 job
- Marx-5\_6\_2\_rc1 job
- Master job
- agg\_5min job
- agg\_1hour job
- agg\_1day-dimensionTable job
- agg\_1month job

The following image illustrates the landing page once all the jobs are scheduled:

Filters

ID	Schedule Status	Project	Flow	Schedule name	First Scheduled	Next Execution	Next in	Repeats Every	Action
51	Enabled	Ingestion_5_6_2_rc1	talendviaspark_http	talend_http	2019-01-18 05:43:00	2019-01-18 05:50:00	55s	1 minute(s)	Disable Remove
52	Enabled	Ingestion_5_6_2_rc1	talendviaspark_nonhttp	talend_nonhttp	2019-01-18 05:44:00	2019-01-18 05:50:00	55s	1 minute(s)	Disable Remove
58	Enabled	Marx5_6_2_rc1	agg_1day-dimensionTable	agg_1day	2019-01-18 20:00:00	2019-01-18 20:00:00	14h, 10m, 55s	1 day(s)	Disable Remove
55	Enabled	Marx5_6_2_rc1	agg_1hour	agg_1hour	2019-01-18 06:10:00	2019-01-18 06:10:00	20m, 55s	1 hour(s)	Disable Remove
57	Enabled	Marx5_6_2_rc1	agg_1month	agg_1month	2019-01-31 21:00:00	2019-01-31 21:00:00	13d, 15h, 10m, 55s	1 minute(s)	Disable Remove
54	Enabled	Marx5_6_2_rc1	agg_5min	agg_5min	2019-01-18 05:51:00	2019-01-18 05:51:00	1m, 55s	5 minute(s)	Disable Remove
53	Enabled	Marx5_6_2_rc1	master	master	2019-01-18 05:46:00	2019-01-18 05:51:00	1m, 55s	5 minute(s)	Disable Remove

version 2.5.0 build 78 2018-03-21 08:12 UTC

Refer the following table to know the jobs that will schedule and run on Azkaban UI

Jobs	Frequency	Start Time
talendviaspark_http	1-min	Current Time
talendviaspark_nonhttp	1-min	Current Time
master	5-min	Current Time

Jobs	Frequency	Start Time
agg_5min	5-min	Current Time
agg_1hour	1-hour	10th Minute of Next Hour
agg_1day-dimensionTable	1-day	01:00 Hour of Next Day
agg_1month	1-month	02:00 Hour of 1st Day of Next Month

After the successful completion of monthly agg job, run the following command to remove the `/data/streaming/monthly-aggregation-ts` file from HDFS

```
ssh <any-namenode>
hdfs dfs -rm -skipTrash /data/streaming/monthly-aggregation-ts
```

Refer to "Appendix B: Updating Feature Variables" on page 55 to know how to update the feature value.

## 7.6 Launching the User Interface

After successful installation of the platform and solution, you can now launch MURAL User Interface and access all the available features and functions.

Ensure that you have successfully executed the following jobs:

- **agg\_1hour job:** to launch the MURAL UI.
- **agg\_1day job:** to access the features and dimensions on UI.

Perform the following steps on both the namenodes to launch the UI on your system:

1. Run the following command to identify the MURAL tomcat docker ID:

```
docker ps | grep mrxui | grep mrxtomcat | awk '{print $1}'
```

For example:

```
# docker ps | grep mrxui | grep mrxtomcat | awk '{print $1}'  
d652e6e4aa35
```

Here, `d652e6e4aa35` is docker Id.

2. Run the following command to login into mrx-tomcat docker using the preceding docker ID:

```
docker exec -it <docker-Id> bash
```

3. Run the following command to move the docker jar in `/opt/apache-tomcat-8.5.11` directory:

```
cd /opt/apache-tomcat-8.5.11  
mv webapps/mrx-web/WEB-INF/lib/xercesImpl-2.6.2.jar logs/  
touch conf/web.xml
```

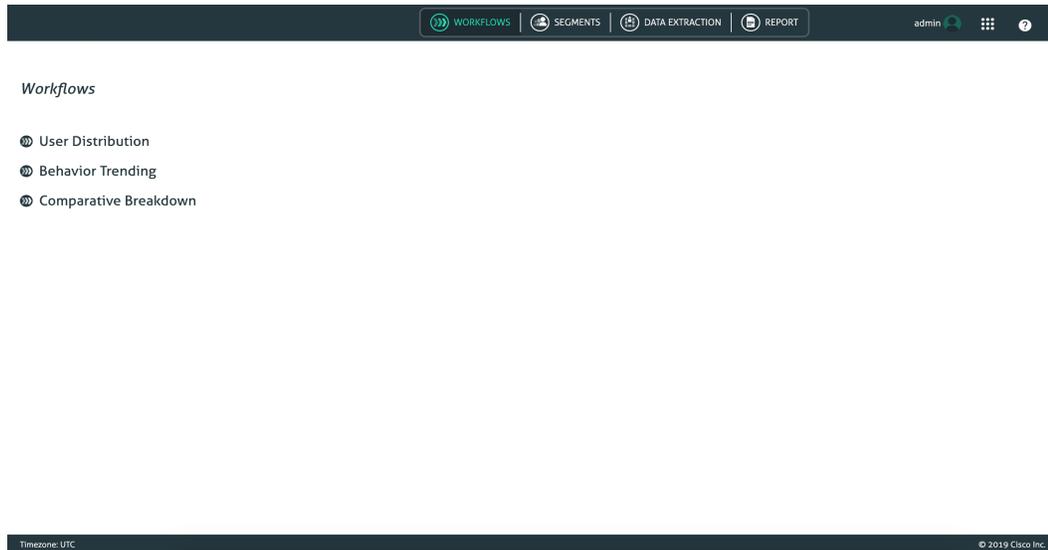
4. Run the following command to escape the file:

```
exit
```

5. Enter the URL: `http://<load-balancer-url/>` in your web browser.

6. Log into the user interface using credentials as **admin** and **abc125** as default username and password respectively.

The MURAL UI is launched. The following image illustrates the landing page:





## 8. Appendix A: Accessing User Interfaces on Multiple Interfaces

---

This section explains the steps to access YARN and Hadoop UI on multiple interfaces on based setup on external network.

Once the platform installation is complete, perform the following step to open YARN and Hadoop resource manager on user interface both the master namens:

```
# iptables -t nat -A PREROUTING -i <publicIPIntfName> -p tcp -m tcp
--dport 8088 -j
DNAT --to-destination <mstnode privateIP>:8088
# iptables -t nat -A PREROUTING -i <publicIPIntfName> -p tcp -m tcp
--dport 50070 -j
DNAT --to-destination <mstnode privateIP>:50070
```

**Note:** Refer to the CIQ sheet to know the VIP and other local IPs to be used for configuration.



## 9. Appendix B: Updating Feature Variables

---

MURAL provides default feature values to run the jobs. You can modify these values by overwriting the variables and adding them to `/extra_vars/solution.yml` file. This section lists the basic feature variables which can be modified as required.

Refer to the following table to know about the features and their default values.

Feature	Property	Default value
Uncategorized Report for urlCat	writeUncat	True
Uncat Percentange	percentageOfUncatEmit	100
Enablement of Data Quality Feature (DQM)	writeUncatGDSODS	True
Data Quality Management GDS Whitelist Enablement	dqm_gds_whitelist_enabled	False
Data Quality Management ODS Whitelist Enablement	dqm_ods_whitelist_enabled	False
Enablement of ODS Classification Feature	odsClassification	False
Enablement of Heavy Users Identification Logic in Base Version	heavyusers_enabled	True
Count of Top Heavy Users Stored in Hive Table	topHeavyHitterCount	1000



## 10. Appendix C: Mandatory Parameters for Incoming ASR Files

---

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

### 10.1 Prerequisites

Ensure that the ASR files have the following:

- The values are separated with a comma.
- They must be gzip compressed files.
- The filename convention should be as mentioned:
  - **flow:** <Node\_name>\_flow\_\*<Timestamp>\_\*.gz
  - **http:** <Node\_name>\_http\_\*<Timestamp>\_\*.gz
- There should be no header and footer in the input file.

**Note:** It is recommended that ASR files must be 10MB compressed size for optimum performance.

### 10.2 Mandatory Attributes for Flow EDRs for MURAL

Flow EDR data sent to the MURAL software must contain the following attributes in the same position as mentioned in the table:

Input Field Position	Field Name
1	sn-flow-start-time
2	sn-flow-end-time
3	radius-calling-station-id
4	ip-subscriber-ip-address
5	bearer-3gpp imsi
6	bearer-3gpp sgsn-address
7	bearer-3gpp user-location-information

Input Field Position	Field Name
8	bearer-ggsn-address
9	sn-charge-volume-ip-bytes-uplink
10	sn-charge-volume-ip-bytes-downlink
11	radius-called-station-id
12	ip-server-ip-address
13	bearer-3gpp imei
14	bearer-3gpp rat-type
15	sn-server-port
16	sn-app-protocol
17	p2p-protocol
18	ip-protocol
19	sn-subscriber-port
20	sn-flow-id
21	sn-duration
22	traffic-type
23	sn-direction
24	p2p-tls-sni
25	sn-volume-amt-ip-pkts-uplink
26	sn-volume-amt-ip-pkts-downlink
27	sn-ruledef
28	sn-service-id
29	sn-rulebase

For example:

```
1541980878,1541980969,447467996231,10.59.16.217,234159503537330,
125.18.124.130,234-15-26-34571,212.183.147.97,0,0,
apn.operator.com,10.204.128.1,35645709-017529-82,
1,53,14,,17,34235,236:1324:28300,0,,
FromMobile,,0,0,RC_DNS_UDP,,wap_adult
```

### 10.3 Mandatory HTTP EDR Attributes MURAL

HTTP EDR data sent to the MURAL system must contain the following attributes in the same position as mentioned in the table:

Input Field Position	Field Name
1	sn-start-time
2	sn-end-time
3	radius-calling-station-id
4	ip-subscriber-ip-address
5	bearer-3gpp imsi
6	bearer-3gpp sgsn-address
7	bearer-3gpp user-location-information
8	bearer-ggsn-address
9	transaction-charge-downlink-bytes
10	transaction-charge-uplink-bytes
11	radius-called-station-id
12	ip-server-ip-address
13	bearer-3gpp imei
14	bearer-3gpp rat-type
15	sn-server-port
16	sn-app-protocol
17	p2p-protocol
18	ip-protocol
19	sn-subscriber-port
20	sn-flow-id
21	sn-duration
22	http-user-agent
23	http-host
24	http-content type
25	http-referer
26	http-reply code
27	http-request method

Input Field Position	Field Name
28	http-url
29	transaction-charge-downlink-packets
30	transaction-charge-uplink-packets

For example:

```
1542055087,1542055088,447467996038,10.59.16.130,234159503414459,
213.223.92.0,234-15-26-34571,212.183.147.97,422,919,
apn.operator.com,85.205.221.240,35462609-855003-88,1,80,5,
http,6,33190,189:403:14020,1,"Mozilla/5.0
(Linux; Android 8.0.0; SM-G950F Build/R16NW; wv)
AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0
Chrome/66.0.3359.126 Mobile Safari/537.36",
start.operator.com,,,304,GET,
http://start.operator.com/ssp/api/services/abc?,1,1
```

## 11. Appendix D: Running the SFTP Downloader Installation

---

The data ingested in the software is fetched by SFTP downloader. This section states the steps to install the SFTP downloader.

Perform the following steps:

1. Log onto the management node as a root user.
2. Run the following command to create and update the repository in `/opt/repos/gvs-platform-prod/release/5.3.0` file:

```
cd /opt/repos/gvs-platform-prod/release/5.3.0
createrepo --update .
```

3. Run the following command to clean the yum repository cache:

```
yum clean all
```

4. Run the following command to install SFTP downloader package on your system:

```
yum install sftp-downloader
```

5. Run the following command to enable SFTP downloader playbook in `/etc/reflex-provisioner` file:

```
cd /etc/reflex-provisioner
ansible-playbook -i ./inventory/generated/prod/mural/hosts
playbooks/mrx_sftp-downloader.yml -k
```

**Note:** Ensure that playbook is executed successfully with `failed=0` counters for all nodes.

6. Run the following command to create MURAL user on both the master namenodes and GGSN source node:

```
useradd -d /home/mural -m mural
```

7. Run the following command to set the common password for MURAL user

on both the GGSN source node and all master namenodes:

```
passwd mural
```

8. Login as MURAL user and run the following commands to generate RSA key on either of the master namenodes:

```
ssh-keygen -t rsa
ssh-copy-id mural@<GGSN Node IP>
scp -r ~/.ssh mural@<Other Master Node IP>:.
```

9. Run the following command to verify keyless access from master namenodes to GGSN source node:

```
ssh mural@<GGSN IP>
```

You have successfully logged into GGSN/EDR source node with MURAL user.

10. Log into GGSN/EDR source node as either root or admin user and run the following commands to create EDR source directories defined in `/etc/reflex-provisioner/work_dir/sftp-downloader-provisioner/inventory/templates/group_vars/global/all/mrx/downloader.yml` file:

```
mrx_di_sftp_sources:
mural-1:
source:
sftp:
port: 22
user: "mural"
password: "<password>"
key: "/home/mural/.ssh/id_rsa"
readyDir: "/data/collector/mural-1/edr/"
consumeDir: "/data/collector/mural-1/consume/"
retentionDir: "/data/collector/mural-1/retention/"
errorDir: "/data/collector/mural-1/error/"
file_pattern: ".{{ source_file_mask }}"
target:
```

Run the following command to create a directory from the user account:

```
mkdir -p <readyDir> <consumeDir> <consumeDir> <retentionDir>
<errorDir>
```

**Note:** Here EDR will be pushed from ASR/SPGW/GGSN into <readyDir> directory.

Run the following command as a MURAL user to copy `id_rsa` file:

```
cd .ssh
cp authorised_keys id_rsa
```

11. Add the following entry in Cron of EDR SOURCE (GGSN) node as a root user to delete processed EDRs older than a day:

```
0 0 * * * /usr/bin/find <retentionDir> -type f -mtime +1 -exec
rm -f {} \;
```

The output may resemble as follows:

```
0 0 * * * /usr/bin/find /data/collector/mural-1/retention/ -
type f -mtime +1 -exec rm -f {} \;
```

12. Run the following command to restart the Crond service:

```
systemctl restart crond
```

13. Run the following command to check the status of Crond service:

```
systemctl status crond
```



## 12. Appendix E: Understanding the Inventory File

---

The platform installation package contains an inventory, the `hosts` file that must be updated with details about all the nodes and their roles in your setup. This `hosts` file is available at `/etc/reflex-provisioner/work_dir/reflex-configuration-module/conf/generate_inventory/conf_inventory/prod/mural`

At least an eight-node setup is required to install the platform. The nodes distribution is as follows. All these nodes are assigned specific roles and these roles are defined under host group in the Inventory file.

Node Type	Minimum Number
Management Node	1
Loadbalancer Node	2
Name Node	2
Data Node	3

However, based on your requirement you can increase the number of Data nodes in your setup. If you increase the number of Data nodes in your setup, ensure that equal number of nodes are assigned appropriate roles in the Inventory file. This ensures maximum computational power for a successful functioning of the platform post installation.

### 12.1 Updating the `hosts` file

Update the `hosts` file for every host group.

Read the following table to know the count of zookeeper, journal and etcd nodes:

Count of Data Node Cluster	Number of Nodes Required
Less than or equal to 10	3
In between 10 to 20	5
More than 20	7

The following sample illustrates the sample `hosts` file:

```
[local]
localhost ansible_connection=local
```

[platform-nodes]

mural-mgmt-1.us.guavus.com platform\_node\_id=1 platform\_rack\_id=1  
mural-lb-1.us.guavus.com platform\_node\_id=2 platform\_rack\_id=2  
mural-lb-2.us.guavus.com platform\_node\_id=3 platform\_rack\_id=3  
mural-mst-1.us.guavus.com platform\_node\_id=4 platform\_rack\_id=4  
mural-mst-2.us.guavus.com platform\_node\_id=5 platform\_rack\_id=5  
mural-slv-1.us.guavus.com platform\_node\_id=6 platform\_rack\_id=6  
mural-slv-2.us.guavus.com platform\_node\_id=7 platform\_rack\_id=7  
mural-slv-3.us.guavus.com platform\_node\_id=8 platform\_rack\_id=8

[management-nodes]

mural-mgmt-1.us.guavus.com

[loadbalancer-nodes]

mural-lb-1.us.guavus.com  
mural-lb-2.us.guavus.com

[master-nodes]

mural-mst-1.us.guavus.com  
mural-mst-2.us.guavus.com

[slave-nodes]

mural-slv-1.us.guavus.com  
mural-slv-2.us.guavus.com  
mural-slv-3.us.guavus.com

[app-nodes]

[zookeeper-nodes]

mural-slv-1.us.guavus.com  
mural-slv-2.us.guavus.com  
mural-slv-3.us.guavus.com

[hdfs-journal-nodes]

## MURAL Software Installation Guide

mural-slv-1.us.guavus.com  
mural-slv-2.us.guavus.com  
mural-slv-3.us.guavus.com

[hdfs-secondary-namenode-nodes]

[kafka-broker-nodes]

mural-slv-1.us.guavus.com  
mural-slv-2.us.guavus.com  
mural-slv-3.us.guavus.com

[schema-registry-nodes]

mural-mst-1.us.guavus.com  
mural-mst-2.us.guavus.com

[hbase-master-nodes]

mural-mst-1.us.guavus.com  
mural-mst-2.us.guavus.com

[hbase-regionserver-nodes]

mural-slv-1.us.guavus.com  
mural-slv-2.us.guavus.com  
mural-slv-3.us.guavus.com

[public-kafka-nodes]

[redis-nodes]

mural-slv-1.us.guavus.com  
mural-slv-2.us.guavus.com  
mural-slv-3.us.guavus.com

[hive-metastore-nodes]

mural-mst-1.us.guavus.com  
mural-mst-2.us.guavus.com

```
[hive-server-nodes]
mural-slv-1.us.guavus.com
mural-slv-2.us.guavus.com
mural-slv-3.us.guavus.com
```

```
[sentry-store-nodes]
mural-mst-1.us.guavus.com
mural-mst-2.us.guavus.com
```

```
[etcd-nodes]
mural-slv-1.us.guavus.com
mural-slv-2.us.guavus.com
mural-slv-3.us.guavus.com
```

```
[impala-state-store-nodes]
mural-mst-1.us.guavus.com
mural-mst-2.us.guavus.com
```

```
[impala-catalog-nodes]
mural-mst-1.us.guavus.com
mural-mst-2.us.guavus.com
```

```
[impala-slave-nodes]
mural-slv-1.us.guavus.com
mural-slv-2.us.guavus.com
mural-slv-3.us.guavus.com
```

```
[impala-slave-nodes-pool1]
mural-slv-1.us.guavus.com
mural-slv-2.us.guavus.com
mural-slv-3.us.guavus.com
```

```
[impala-slave-nodes-pool2]
```

## MURAL Software Installation Guide

```
mural-slv-1.us.guavus.com  
mural-slv-2.us.guavus.com
```

```
[impala-slave-nodes-pool3]  
mural-slv-3.us.guavus.com  
mural-slv-1.us.guavus.com
```

```
[impala-slave-nodes-pool4]  
mural-slv-2.us.guavus.com  
mural-slv-3.us.guavus.com
```

```
[impala-slave-nodes-pool5]  
mural-slv-1.us.guavus.com  
mural-slv-2.us.guavus.com  
mural-slv-3.us.guavus.com
```

```
[yarn-resourcemanager:children]  
master-nodes
```

```
[yarn-nodemanager:children]  
slave-nodes
```

```
[yarn-nodes:children]  
yarn-resourcemanager  
yarn-nodemanager  
app-nodes  
management-nodes
```

```
[srx-ui-impala-slave-nodes:children]  
impala-slave-nodes-pool1
```

```
[srx-alert-impala-slave-nodes:children]  
impala-slave-nodes-pool2
```

```
[srx-di-impala-slave-nodes:children]  
impala-slave-nodes-pool4
```

```
[srx-data-impala-slave-nodes:children]  
impala-slave-nodes-pool3
```

```
[srx-data-read-only-impala-slave-nodes:children]  
impala-slave-nodes-pool5
```

```
[platform-azkaban-nodes]  
mural-mst-1.us.guavus.com  
mural-mst-2.us.guavus.com
```

```
[kube-master:children]  
master-nodes
```

```
[kube-node:children]  
loadbalancer-nodes  
management-nodes  
app-nodes  
slave-nodes  
master-nodes
```

```
[k8s-cluster:children]  
kube-master  
kube-node
```

```
[postgres-nodes:children]  
management-nodes  
master-nodes
```

```
[corosync-nodes:children]  
postgres-nodes  
loadbalancer-nodes
```

## MURAL Software Installation Guide

```
[graphite-node:children]  
management-nodes
```

```
[grafana-node:children]  
management-nodes
```

```
[zabbix-server-nodes:children]  
management-nodes
```



## 13. Appendix F: Changing Logo and Password on Azkaban UI

---

You can change the UI banner after installing the MURAL software on your system. This section describes the procedure of changing the UI logo as required and updating the password on Azkaban UI.

Perform the following steps:

1. Log onto the management node as root user.
2. Switch to `/opt/guavus/carereflex/platform/default/deployment/azkaban/platform` directory.  

```
cd
/opt/guavus/carereflex/platform/default/deployment/azkaban/pla
tform
```
3. Update the password value in `azkaban-config.yml` file under `<azkaban-users>` section:

The output may resemble as follows:

```
<user username="azkaban" password="<new-password>"
roles="admin" groups="azkaban" />
```

4. Update `azkaban-config.yml` file to modify `azkaban.name` as *Cisco*
5. Update the password value in `azkaban-config.yml` file under `azkacli.cfg` section:

The output may resemble as follows:

```
[api]
host = localhost
port = 8507
user = azkaban
password = <new-password>
```

6. Run the following command to update the Azkaban configuration manager:

```
kubectl edit cm azkaban-platform
```

Execute the steps 3, 4 and 5 to change the logo and password in the container.

7. Run the following command to delete the already running pod:

```
kubectl delete pod <azkaban-platform-name> --force --grace-period=0
```

8. Run the following command to verify the status of created azkaban pod:

```
kubectl get pods
```

**Note:** Ensure that all the pods are in running state after executing the mentioned commands after 5 minutes.