Micro Climate™ Management System – R4.0.0 Installation and Upgrade Guide

PON Manager, PON Controller and Netconf Server

Juniper Networks, Inc. 1133 Innovation Way
Sunnyvale, CA 94089

Phone: 888.JUNIPER (888.586.4737)
Web https://support.juniper.net/support/requesting-support

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

The information in this document is current as of the date on the title page.

Revision History

Revision	Date	Comments
0.1	May 5, 2020	Initial release
1.3	June 2, 2020	TN051 assignment
2.0	October 7, 2020	Updated for Release 2.0.0
2.1	February 16, 2021	Updated for Release 2.1.0
2.2	July 22, 2021	Updated for Release 2.2.0
2.3	December 2, 2021	Updated for Release 2.3.0
3.0	April 11, 2020	Updated for Release R3.0.0
3.1	August 31, 2022	Updated for Release R3.1.0
3.2	December 20, 2022	Updated for Release R3.2.0
3.2.1	August 17, 2023	Updated Caution label
4.0.0	September 5, 2023	Updated for Release R4.0.0

Table of Contents

List of Figures

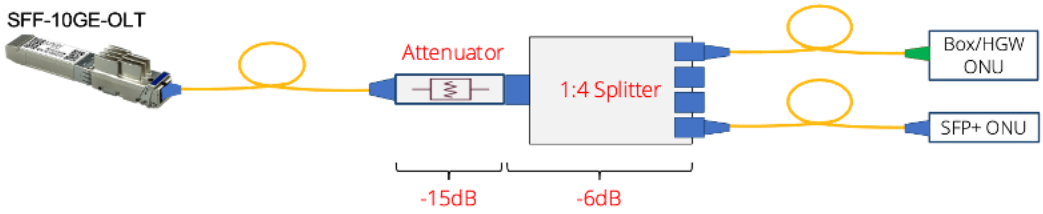
WARNING !

Refer to Install Guide before Installation

Warranty Notice: Device Attenuation Required

Do not connect OLT directly to ONUs without proper attenuation. PON transceivers will be **permanently damaged** unless connected with minimum 16dB attenuation (20dB recommended)
Damage from optical overload will void warranty.

Combination of attenuator and splitters can provide required attenuation -- see example:



Introduction

This document is a guide for installing the PON Domain Manager into a single system running ubuntu 18.04 or 20.04. The system could be running in a virtual machine or directly. For details on building a virtual machine, please reference the Technical Note “Installation Guide for PON Controller, TAPI and GUI” (providing details to create an Ubuntu VM using VirtualBox).

Solution Overview

The MicroClimate™ Management System (MCMS) is the management solution for PON networks. The MCMS architecture is shown in Figure 1 and consists of the MCMS PON Manager graphical user interface, MCMS Netconf Server, and MCMS PON Controller. Together these components provide a complete network management solution for provisioning and monitoring MicroPlug™ OLT devices, as well as the subtended MicroPlug™ ONU devices and third-party ONUs compliant with the 10G-EPON and XGS-PON standards.

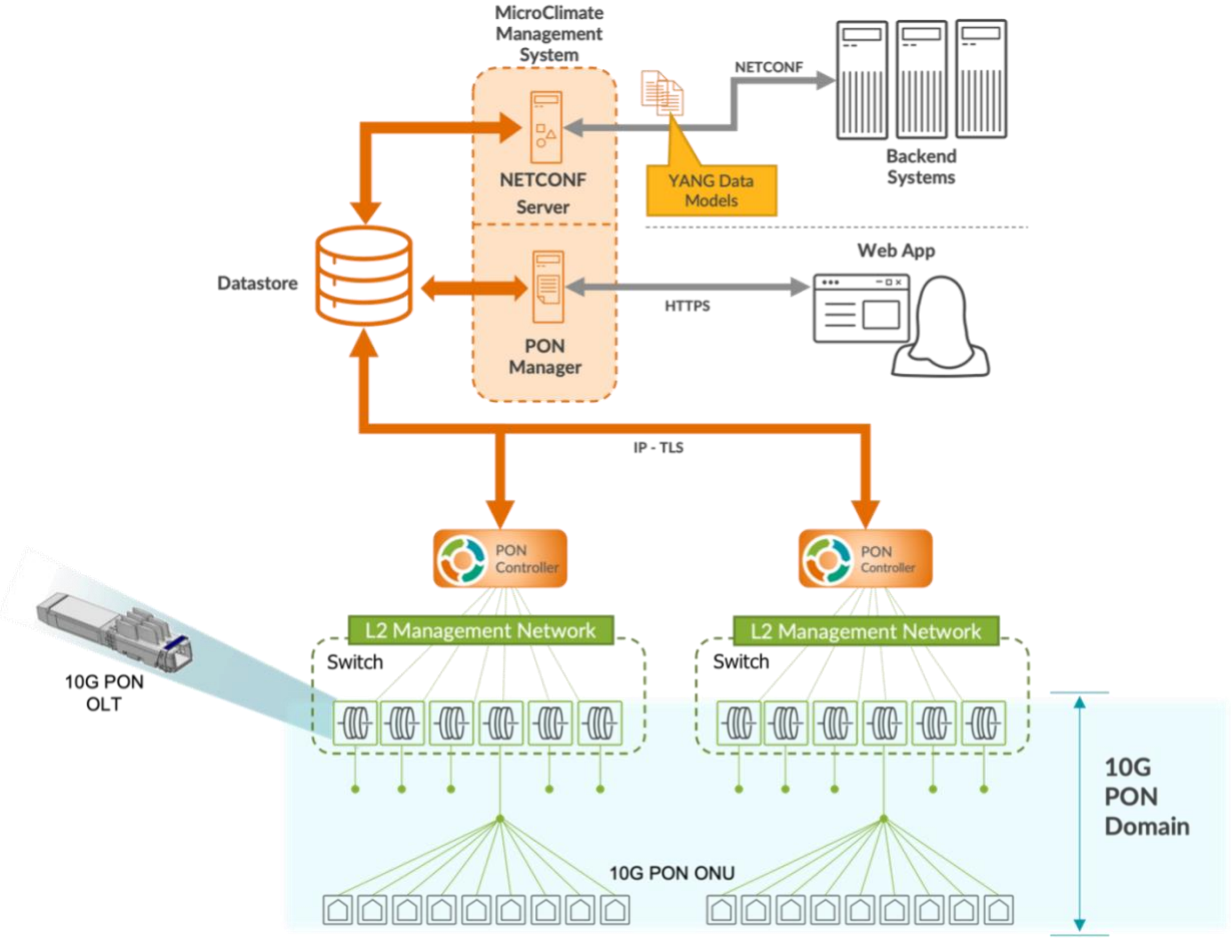


Figure 1 - MicroClimate™ Management System Architecture

MCMS PON Manager

The MCMS PON Manager is a single-page web application (Web App) and an accompanying REST API that provides a graphical user interface for managing the PON Network. The Web App is built on the Angular web application framework, which provides an HTML and JavaScript front-end user interface. The REST API accompanies the Web App for the purposes of providing access to MongoDB for managing MCMS PON Manager users and the PON Network.

The MCMS PON Manager has the following features:

- Alarm management.
- Dashboard view with a summary of PON network conditions.
- Device monitoring and statistics.
- Device provisioning and management.
- Logging for diagnostics and troubleshooting.
- MCMS PON Controller database management.
- MCMS PON Manager user management.
- Polyglot graphical OMCI (and future 10G EPON OAM) service configuration tool.
- Service configuration, including VLANs, Service Level Agreements (SLAs), 802.1X Authentication, and DHCP Relay.

Note that the REST API should only be used with the MCMS PON Manager. At this time, the REST API is not a published interface and will change in future releases. Future versions for MCMS PON Manager may make the REST API available directly to customer applications.

MCMS Netconf Server

The MCMS Netconf Server provides a standard Netconf interface and customer facing API for managing the PON network. The Netconf Server is built on the Netopeer2 and Sysrepo open-source architecture and interfaces with MongoDB. The Netconf solution supports standard Broadband forum (BBF) TR-383 and TR-385 YANG models for configuring subscriber services for the PON network. In addition to standard YANG models, MCMS YANG models provide a complete Netconf management solution for PON Controllers, OLTs, and ONU devices.

MCMS PON Controller

The MCMS PON Controller is a stateless management controller and device driver application for configuring and monitoring the end points in a MicroPlug™ OLT PON network. MongoDB serves as the northbound application programming interface for the PON Controller. The PON Controller applies configuration to OLT and ONU devices from documents stored in MongoDB. The MCMS PON Controller uses IEEE 1904.2 packets (L2) to communicate with the OLT and ONU devices. The PON Controller also collects state information, statistics, and logs from devices and reports the information to higher layer applications through MongoDB. The PON Manager and Netconf interfaces manage the PON Controller through MongoDB.

MongoDB Datastore

The Mongo database (MongoDB) provides the datastore for the MicroClimate™ Management System. The MongoDB datastore contains all the configuration, state, statistics, alarms, and logging data for the devices in the PON network. Northbound interfaces, such as the MCMS PON Manager, MCMS Netconf Server, and customer applications interface with MongoDB to provision and retrieve monitoring information for devices in the PON network. MongoDB serves as the interface between the PON Manager and Netconf and the PON Controller.

Provisioning data generally flows "downstream" through the management network. The PON Manager and Netconf interfaces write device configuration to MongoDB. The PON Controller reads the configuration data from MongoDB and programs the OLT and ONU devices accordingly.

Monitoring data, including device state, statistics, alarms, and logging, is collected and flows "upstream" through the management network. The PON Controller periodically collects state information from devices in the PON network and writes the monitoring data to MongoDB. The PON Manager reads the monitoring data from MongoDB for display in the Web App.

Server Sizing

Lab Testing

It is recommended to run all the MCMS PON Management applications on a single VM (or single server).

- Management Applications
  - Pon Controller
  - Pon Manager
  - MongoDB
  - Netconf server
- Server Hardware Requirements
  - 2 vCPUs
  - 8 GB RAM
  - 20 GB disk space
- PON Size
  - 4 OLTs
  - 32 ONUs

Production Deployments

It is recommended to run each of the MCMS PON Management applications on separate VMs (or servers). The following table provides guidelines for deployments.

Note: The sizing requirements are for the Application process only and does not include the base OS (Ubuntu 18.04 or 20.04).

Application	Subscriber Size	
	3072 Subscriber	**100K Subscribers**
PON Controller [1]	1 Instance: 1 vCPU 1 GB RAM 1 GB disk space	32 x Pon Controller Instances: 32 vCPU 32 GB RAM 32 GB disk space
PON Manager	2 vCPU 8 GB RAM 1 GB disk space	4 vCPU 16 GB RAM 1 GB disk space
Netconf [2]	1 Instance: 2 vCPU 1 GB RAM 50 MB disk space	3 x Netconf Instances: 6 vCPU 3 GB RAM 150 MB disk space
MongoDB	2 vCPU 8 GB RAM 10 GB disk space	16 vCPU 64 GB RAM 100 GB disk space

[1] A single PON Controller instance supports 3072 Subscribers (a single switch with 48 OLTs x 64 ONUs/OLT). This instance supports ~25 ms latency (round trip) between the VM and the OLTs. (Note: 25 ms is the typical latency to Regional Datacenters). For longer latency delays, the scalability (i.e., number of ONUs) could be different. Please contact Juniper PON Support for details.

[2] This is an estimate. Please contact Juniper PON Support for details.

Installation for Lab Testing – Single VM

For initial lab and proof of concept testing, we recommend building an Ubuntu VM or system described in the Technical Note “Installation Guide for Ubuntu VM using VirtualBox on Windows”.

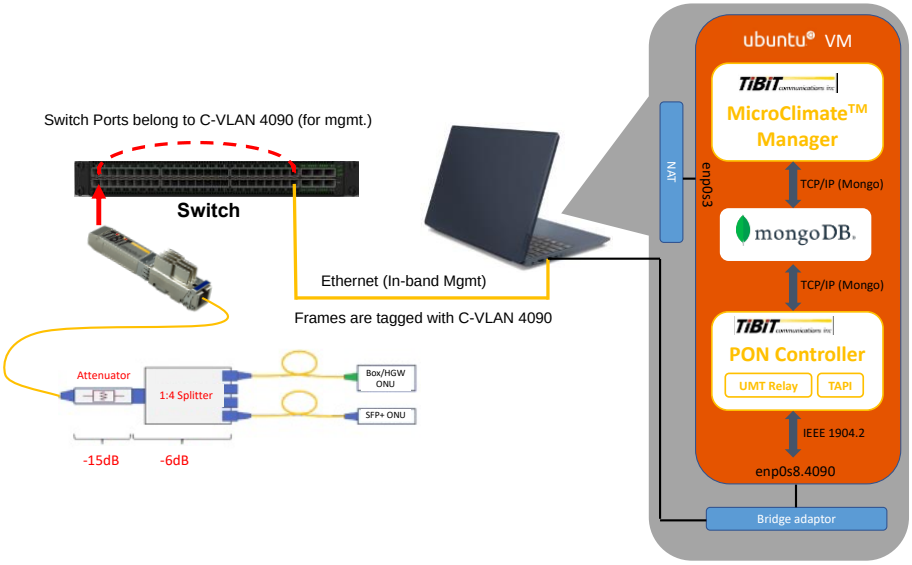


Figure 2 - Single System

After the VM is created, we recommend connecting the PC and OLTs into a switch as illustrated in Figure 2.

System Requirements

Before beginning the installation, ensure the system meets this list of requirements:

- Ubuntu 18.04 or 20.04
  - Verify system is running ubuntu version 18.04 or 20.04

```
user@system:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.6 LTS
Release:        20.04
Codename:       focol
```

- Two Ethernet interfaces
  - Ethernet interface for connectivity to PON 802.1q VLAN 4090
    - Verify the system is configured with the 802.1q VLAN 4090

```
user@system:~$ sudo ip link add link enp0s8 enp0s8.4090 type vlan \
                  proto 802.1q id 4090 egress-qos-map 0:7
user@system:~$ sudo ip link set enp0s8.4090 up
user@system:~$ ifconfig
```

(Note: the commands above are temporary and will have to be re-entered each time Ubuntu is restarted. See Appendix A at the end of this document for instructions on how to create a permanent network config for your system)

    - Interface for Internet connectivity
        - Verify ubuntu system has connectivity to Internet
        - This is required to downloaded required python dependencies and the MongoDB Community Server.
- Firefox version 70.0.0 or higher
- MongoDB version v4.00 or higher is required. With Ubuntu 20.04, Mongo DB version v4.4 or higher is required.
    - Note: This will automatically be installed by the installation scripts

New Installation – Single VM

Note: If you are upgrading an existing installation, please skip to the next section with title "Upgrading Existing Installation" on the next page.

This section describes the installation for a new installation. If you are already running the PON Domain Manager, please skip to the next section, Upgrading Existing Installations.

For a new installation, run the installation script from the ubuntu command line:

```
user@system:<install_dir>/MCMS-R4.0.0-UB2004$ sudo ./install.sh -e <eth_interface>
```

This script will do the following:

- Install MongoDB (downloads from repo.mongo.org)
- Install MCMS PON Manager
- Install MCMS Netconf Server
- Install PON Controller and UMT Relay with Ethernet Interface

The available options for the installation script (install.sh) are:

Required:

- -e <eth_interface> The ethernet vlan interface used by the Pon Controller and UMT relay

Optional:

- -d <database_ip> The database IP for MongoDB (default = 127.0.0.1)
- -n <database_name> The default database used by Pon Mgr (default = tibit_pon_controller)
- -m Install only the Pon Manager/MongoDB/Netconf Server
- -c Install only the Pon Controller

Creating Initial Account in PON Manager

This section describes how to create the initial login account in PON Manager. This account will have administrative privileges and supports the ability to create new accounts for access into the PON Manager.

- Connect to PON Manager using any of the IP addresses configured on the ubuntu system
- Create an Account. Ensure the password meets the Password Requirements listed.

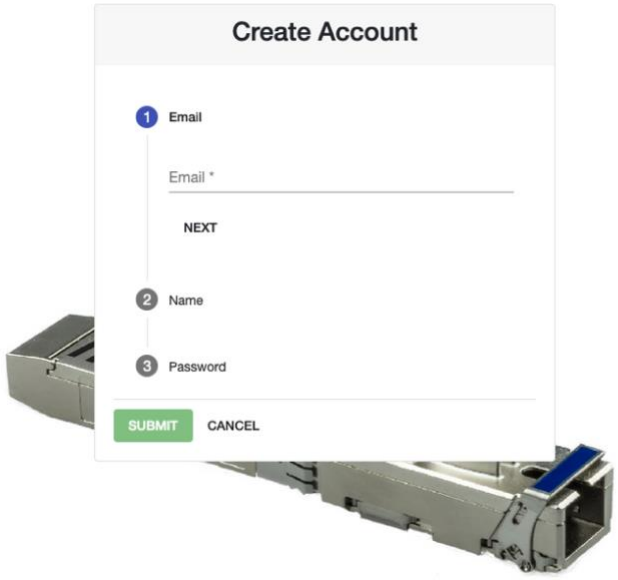


Figure 3 - Create Initial Login Account

Initialize Database

This section describes how to Initialize the Database after the upgrade process completes. Some items (specifically OLT Firmware) will need to be added into the MongoDB.

- Connect to the PON Manager
- Select the Global Config button on the main menu.
- Select Databases
- Then select the **Initialize** tab (see screenshot below)
- Select the Database to initialize (i.e. Default)
- Check (or uncheck) all the boxes for the information wanted to copy into the Database
- Click **Sart** in the **Database to initialize**

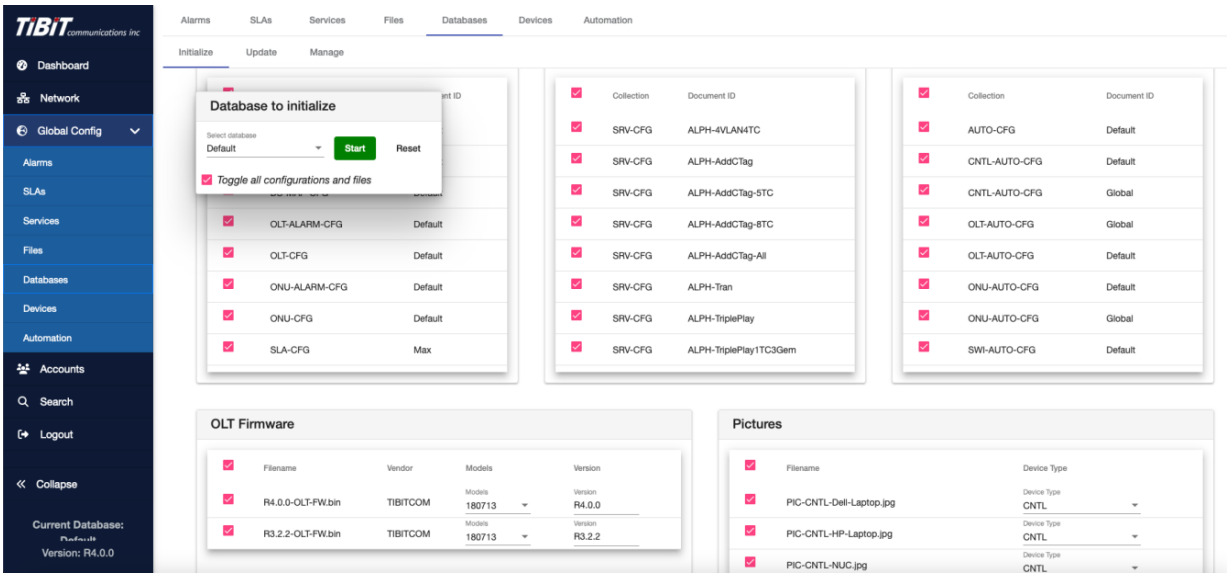


Figure 4: Initialize Database

Upgrading Existing Installations

Upgrade Single VM system with PON Controller, PON Manager and MongoDB

This section describes the installation for an existing installation of the PON Domain Manager. If this is a new installation, please follow the steps in the previous section, New Installation.

For upgrading to R4.0.0 (from any previous version), run the following upgrade script:

```
user@system:<install_dir>/MCMS-R4.0.0-UB2004$ sudo ./upgrade.sh
```

This script will do the following:

- Upgrade to R4.0.0 PON Manager
- Upgrade to R4.0.0 PON Controller
- Upgrade to R4.0.0 Netconf server (if installed)

Post Upgrade tasks

Initialize Database

This section describes how to Initialize the Database after the upgrade process completes. Some items (specifically OLT Firmware) will need to be added into the MongoDB.

- Connect to the PON Manager
- Select the Global Config button on the main menu.
- Select Databases
- Then select the **Initialize** tab (see screenshot below)
- Select the Database to initialize (i.e. Default)
- Check (or uncheck) all the boxes for the information wanted to copy into the Database
- Click **Sart** in the **Database to initialize**

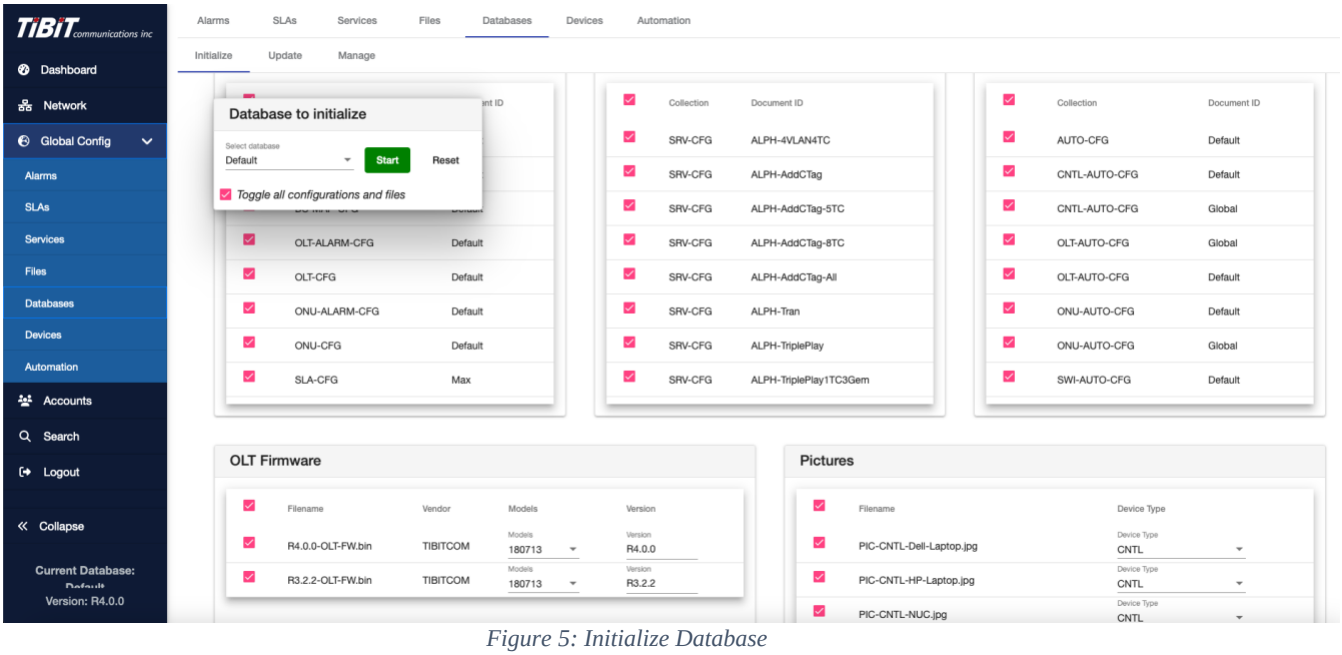


Figure 5: Initialize Database

Database Update

This section describes how to Update the Database after the upgrade process completes. An upgrade re-uses the existing previously installed Mongo DB. Some items will need updating after the upgrade.

- Connect to the PON Manager
- Select the Global Config button on the main menu.
- Select **Databases** and then select the **Update** tab (see screenshot below)
- Under **Select Configuration to Upgrade**, check all the boxes
- Under **From Version** select **R3.2**
- Click **Save** in the **Apply New Fields**

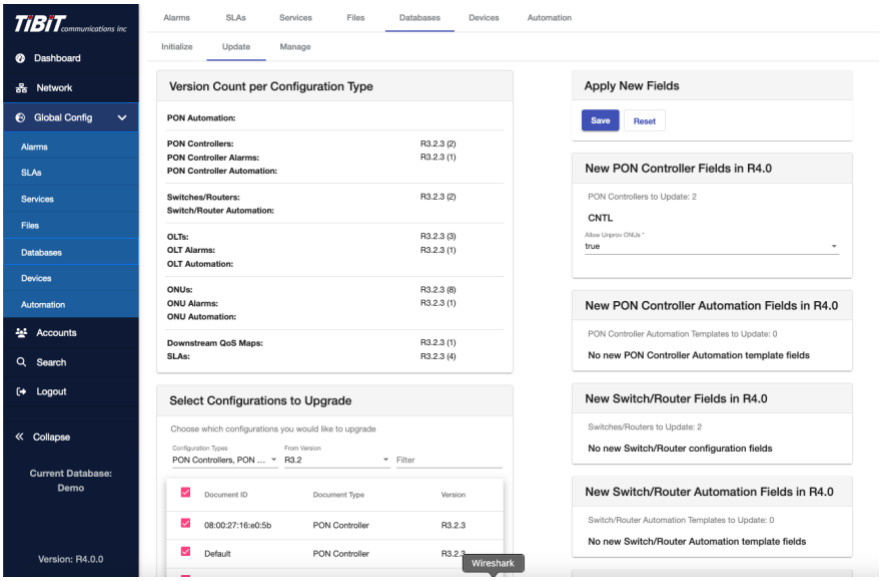


Figure 6 – Apply New Fields

After selecting **Save** , the list of **Configuration Types** should update to R4.0.0

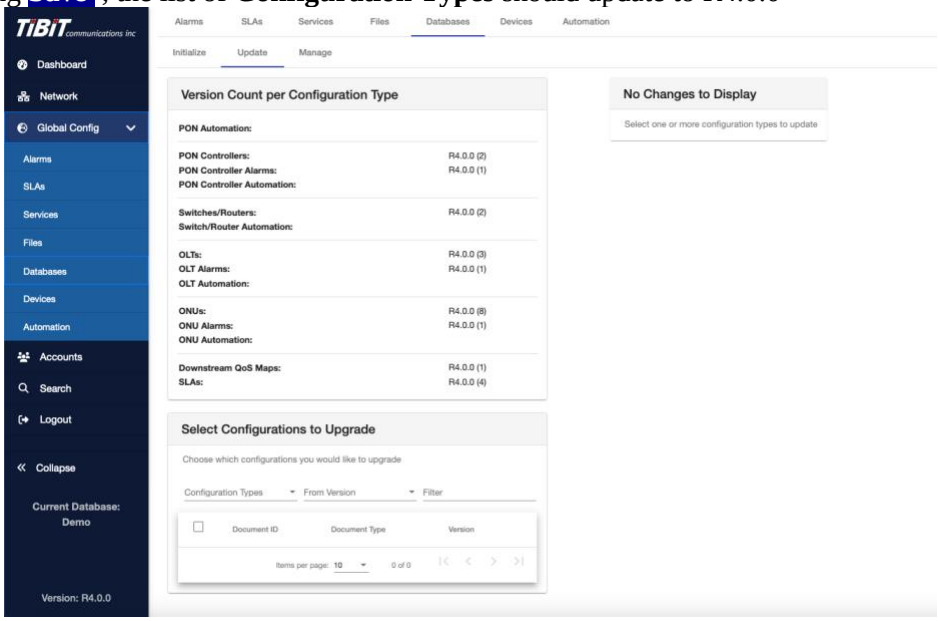


Figure 7 - Database Update

Upgrading firmware on OLT

This section describes how to upgrade the firmware on existing OLTs and the Default OLT configuration.

- Connect to the PON Manager
- Navigate to the **OLT Firmware** tab

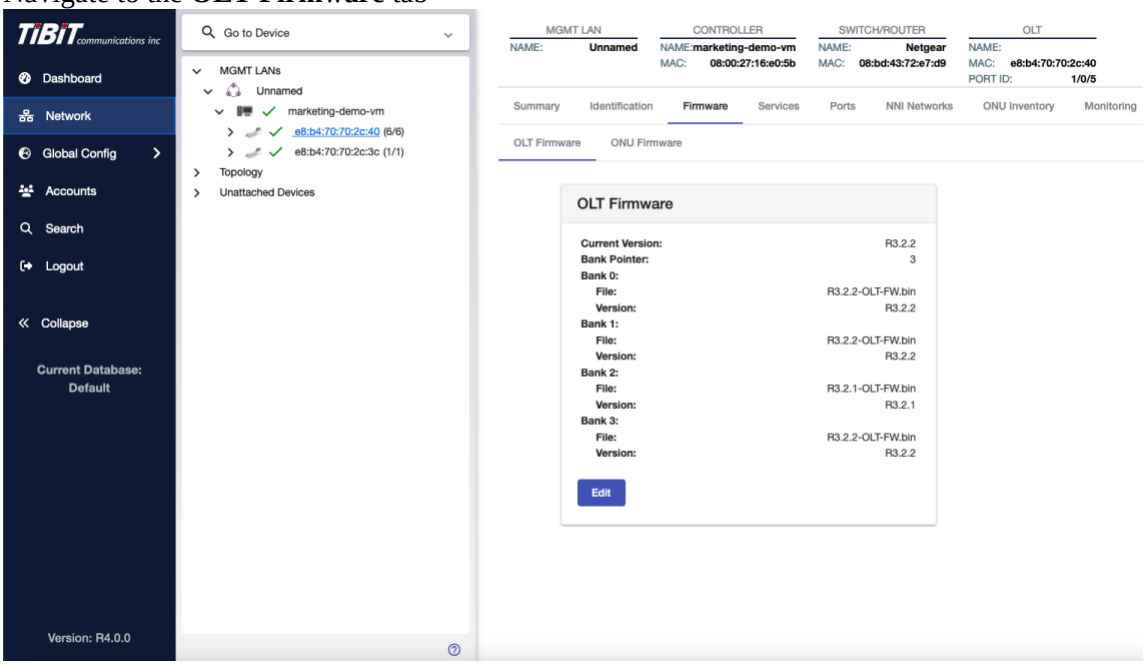


Figure 8 - OLT Firmware

- Click the EDIT button

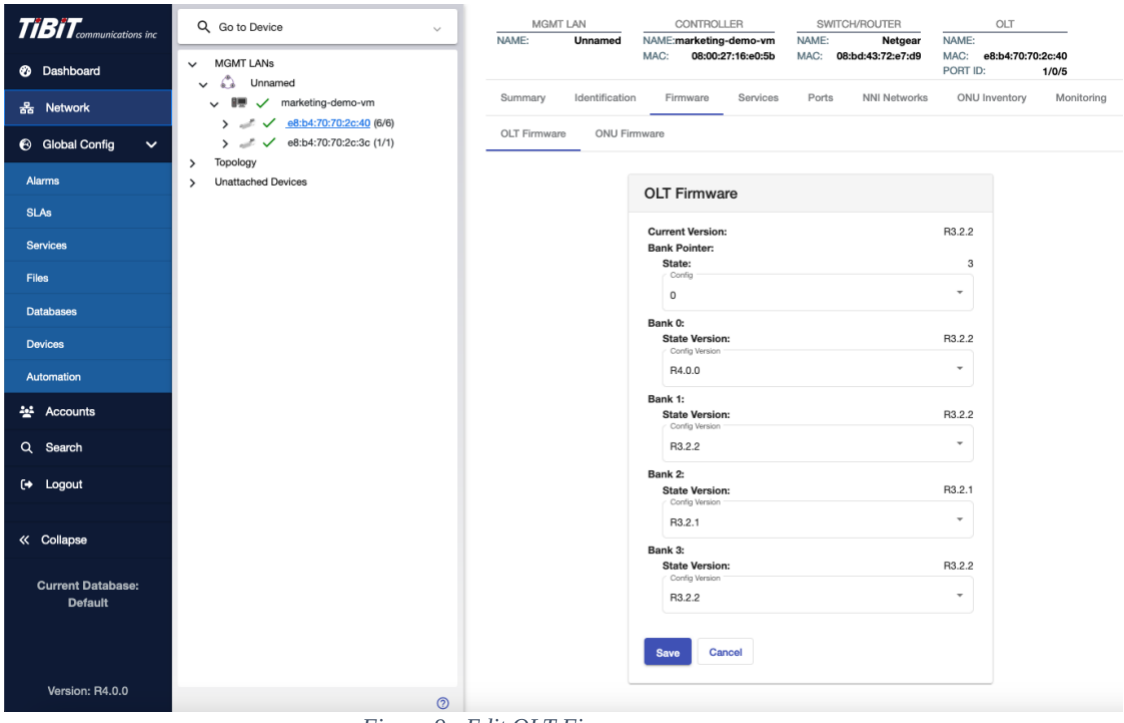


Figure 9 - Edit OLT Firmware

- Select one of the firmware banks and select **R4.0.0**
- Click the SAVE button
- Wait for the PON Controller to transfer the firmware to OLT (~ 1 min.)
    - When the State: R3.2.2 disappears, the firmware is loaded.

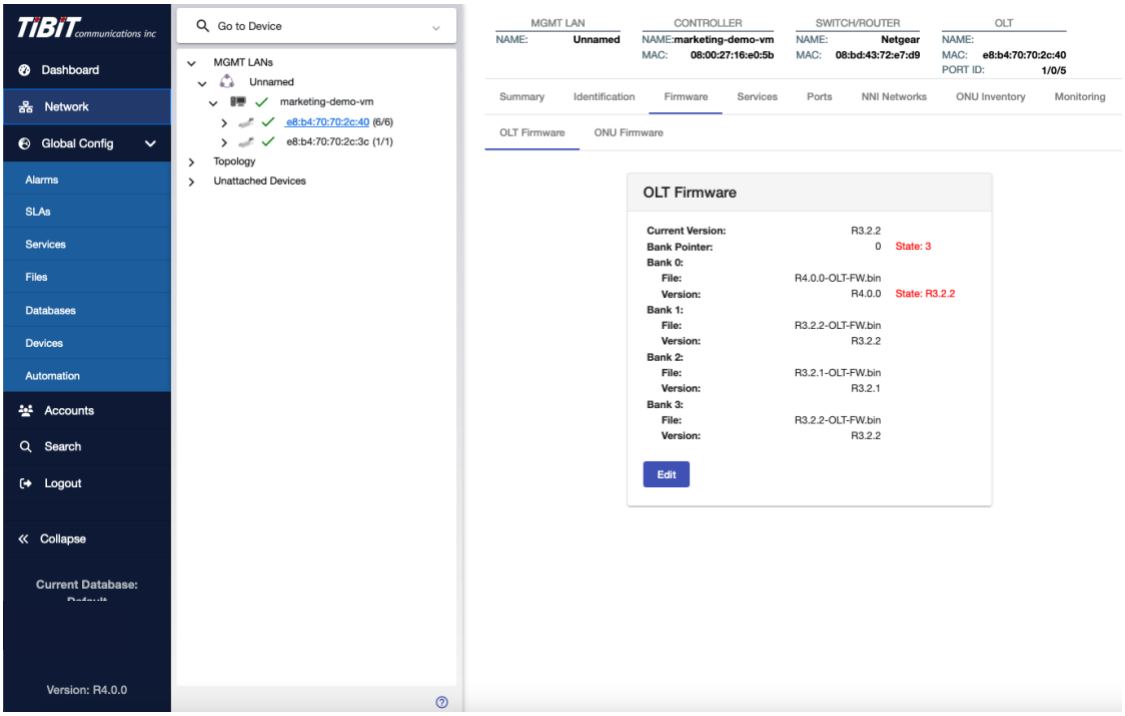


Figure 10 - Upgrade Firmware: Load Firmware to OLT

- Wait for the PON Controller to reset OLT (~ 1 min.)

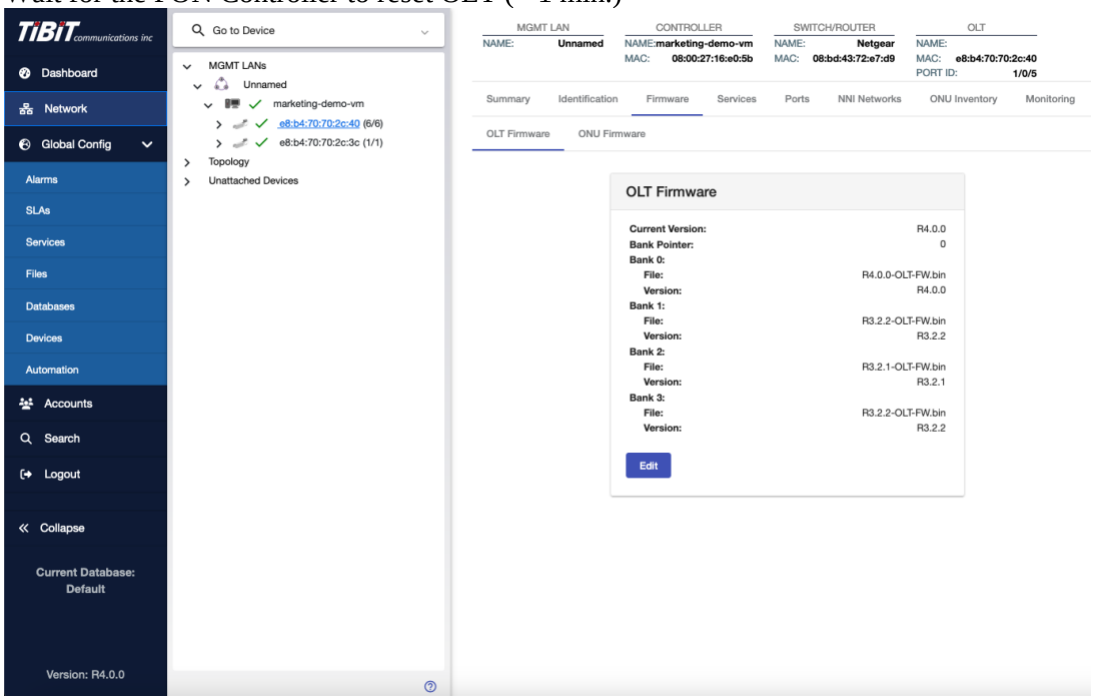


Figure 11 - OLT Firmware (after upgrade)

Post Installation – Verify Processes

This section describes how to verify the PON Manager, PON Controller and MongoDB processes are running.

1. Use the script `status.sh` provided with the release

```
user@system:~/R4.0.0-UB2004$ ./status.sh
MCMS Component Versions:
     PON Manager: R4.0.0
     PON NETCONF: R4.0.0
  PON Controller: R4.0.0

● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-11-30 14:56:17 PST; 4 months 16 days ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 656 (mongod)
   CGroup: /system.slice/mongod.service
           └─656 /usr/bin/mongod --config /etc/mongod.conf

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Fri 2022-04-08 09:56:02 PDT; 1 weeks 2 days ago
  Process: 11716 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
  Process: 29373 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
  Process: 11722 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 11726 (apache2)
    Tasks: 151 (limit: 4915)
   CGroup: /system.slice/apache2.service

● tibit-netopeer2-server.service - Tibit Communications, Inc. Netopeer2 Server
   Loaded: loaded (/lib/systemd/system/tibit-netopeer2-server.service; enabled; vendor: enabled)
   Active: active (running) since Fri 2022-04-08 09:54:42 PDT; 1 weeks 2 days ago
 Main PID: 10819 (netopeer2-serve)
    Tasks: 7 (limit: 4915)
   CGroup: /system.slice/tibit-netopeer2-server.service
           └─10819 /opt/tibit/netconf/bin/netopeer2-server -v 1 -t 55

● tibit-umtrelay.service - Tibit Communications, Inc. UMT Relay
   Loaded: loaded (/lib/systemd/system/tibit-umtrelay.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-04-08 09:55:08 PDT; 1 weeks 2 days ago
 Main PID: 11002 (tibit-umtrelay)
    Tasks: 9 (limit: 4915)
   CGroup: /system.slice/tibit-umtrelay.service
           ├─11002 /opt/tibit/poncntl/bin/tibit-umtrelay -c /etc/tibit/poncntl/PonCntlInit.json
           └─11004 /opt/tibit/poncntl/bin/tibit-umtrelay -c /etc/tibit/poncntl/PonCntlInit.json

● tibit-netconf.service - Tibit Communications, Inc. NetCONF Server
   Loaded: loaded (/lib/systemd/system/tibit-netconf.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-04-08 09:54:42 PDT; 1 weeks 2 days ago
 Main PID: 10812 (tibit-netconf)
    Tasks: 12 (limit: 4915)
   CGroup: /system.slice/tibit-netconf.service
           ├─10812 /opt/tibit/netconf/bin/tibit-netconf
           └─10821 /opt/tibit/netconf/bin/tibit-netconf

● tibit-poncntl.service - Tibit Communications, Inc. PON Controller
   Loaded: loaded (/lib/systemd/system/tibit-poncntl.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-04-08 09:55:06 PDT; 1 weeks 2 days ago
 Main PID: 10987 (tibit-poncntl)
    Tasks: 82 (limit: 4915)
   CGroup: /system.slice/tibit-poncntl.service
           ├─10987 /opt/tibit/poncntl/bin/tibit-poncntl -c /etc/tibit/poncntl/PonCntlInit.json
           └─10990 /opt/tibit/poncntl/bin/tibit-poncntl -c /etc/tibit/poncntl/PonCntlInit.json
```

2. Verify MongoDB is running

```
user@system:~$ sudo systemctl status mongod.service

● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-08-30 11:56:38 PDT; 3 days ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 15035 (mongod)
   CGroup: /system.slice/mongod.service
           └─15035 /usr/bin/mongod --config /etc/mongod.con
```

3. Verify PON Manager Apache Web Server is running

```
user@system:~$ sudo systemctl status apache2.service

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Fri 2019-08-16 15:19:09 PDT; 1 weeks 2 days ago
  Process: 2981 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 Main PID: 8471 (apache2)
```

4. Verify PON Controller is Running

```
usersystem:~/$ sudo systemctl status tibit-poncntl.service

● tibit-poncntl.service - Tibit Communications, Inc. PON Controller
   Loaded: loaded (/lib/systemd/system/tibit-poncntl.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-12-20 13:27:14 PST; 2min 45s ago
 Main PID: 29363 (PonCntl)
    Tasks: 7 (limit: 4915)
   CGroup: /system.slice/tibit-poncntl.service
           ├─29363 /opt/tibit/poncntl/PonCntl
           └─29371 /opt/tibit/poncntl@/PonCntl
```

5. Verify UMT Relay is Running (Optional process for DHCP relay w/ Option 82)

```
usersystem:~/$ sudo systemctl status tibit-umtrelay.service

● tibit-umtrelay.service - Tibit Communications, Inc. UMT Relay
   Loaded: loaded (/lib/systemd/system/tibit-umtrelay.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-12-20 13:27:14 PST; 2min 45s ago
 Main PID: 26707 (tibit-umtrelay)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/tibit-umtrelay.service
           ├─26707 /opt/tibit/poncntl/tibit-umtrelay -c /opt/tibit/poncntl/PonCntlInit.json
           └─26710 /opt/tibit/poncntl/tibit-umtrelay -c /opt/tibit/poncntl/PonCntlInit.json
```

6. Verify Authenticator is Running (Optional process for EAPOL Authentication)

```
usersystem:~/$ sudo systemctl status tibit-authenticator.service

● tibit-authenticator.service - Tibit Communications, Inc. EAPOL Authenticator
   Loaded: loaded (/lib/systemd/system/tibit-authenticator.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-05-20 10:07:32 PDT; 24s ago
 Main PID: 5061 (tibit-authentic)
    Tasks: 7 (limit: 4915)
   CGroup: /system.slice/tibit-authenticator.service
           ├─5061 /opt/tibit/poncntl/bin/tibit-authenticator -c /etc/tibit/poncntl/PonCntlInit.json
           └─5065 /opt/tibit/poncntl/bin/tibit-authenticator -c /etc/tibit/poncntl/PonCntlInit.json
```

Installation for Production Deployments

For production deployments, Tibit recommends installing each of the MCMS PON Management applications on separate VMs (or servers).

MongoDB Installation

MongoDB serves the purpose of being the datastore for the MicroClimate™ Management System. This includes all PON-specific information as well as user models.

For production deployments, it is recommended to configure multiple servers or VMs in a Replica Set using Authentication and TLS. Please reference the Tech Note, Mongo Replication with Authentication and TLS/SSL for configuration of each Mongo instance.

Installation

To install the MongoDB Community Edition on each instance,

1. Access the official MongoDB website.
2. Using the table of contents at the top of the page, select the option, "Uninstall MongoDB Community Edition", and follow the instructions to uninstall MongoDB if installed.
3. Using the same table of contents at the top of the page, select the option; "Install MongoDB Community Edition using .deb Packages", and follow the instructions to install the latest version of MongoDB.
    - Ensure that you are using the "Ubuntu 18.04" or "Ubuntu 20.04" options.

MongoDB Scripts

Alternatively, these scripts can automate installing, starting, and uninstalling MongoDB.

Installation steps:

1. Transfer/download the .zip file to Ubuntu host or VM.
2. From a Linux shell, unpack the .zip file and change to the following directory:

```
unzip MCMS-R4.0.0-UB2004.zip
cd MCMS-R4.0.0-UB2004/R4.0.0-PonManager-UB2004
```

Note: Only the following installation scripts need to be transferred to the Ubuntu host or VM for each Mongo instance.

mongodb_install.sh

This script will uninstall the current version of MongoDB and then install and start the latest version of MongoDB server Community Edition.

- Supported Operating Systems:
    a. Ubuntu 18.04
    b. Ubuntu 20.04
- Dependencies:
    a. systemctl

   b. gpupg
   c. lsb-release
- Run the command:

```
sudo bash ./mongodb_install.sh
```

mongodb_start.sh

This script will start a forked instance of a MongoDB server

- Supported Operating Systems:
   a. Ubuntu 18.04
   b. Ubuntu 20.04
- Dependencies:
   a. systemctl
- Run the command

```
sudo bash ./mongodb_start.sh
```

MCMS PON Manager Installation

The MCMS PON Manager software is composed of a graphical user front-end web application (Web App) and a RESTful interface (REST API) that provides access to the MongoDB datastore.

Prerequisites:

- Ensure that you have an active internet connection.
- Install/upgrade existing python3 package to be equal to or greater than version 3.6.7.
- Install/upgrade existing pip3 package to be equal to or greater than version 9.0.0.
- Ensure that you have your MongoDB Server running that will manage user authentication and serve as your default database.
- Ensure you have the latest ubuntu updates

Installation steps:

1. Transfer/download the .zip file to Ubuntu host or VM.
2. From a Linux shell, unpack the .zip file and change to the following directory:

```
unzip MCMS-R4.0.0-UB2004.zip
cd MCMS-R4.0.0-UB2004/R4.0.0-PonManager-UB8104
```

3. *Optional* | If desired, you can configure the self-signed certificate created during installation by modifying the configuration file within this package directory; 'tibit_ponmgr_selfsigned_cert_req.conf'.
   There are two sections that may be modified:

- [req_distinguished_name]

countryName	Country code. | Ex: 'US' (United States), etc
ST	State or Province
localityName	City or Locality
organizationName	Name of organization
organizationalUnitName	Name of group/unit within organization
CN	Fully Qualified Domain Name of system

- [alt_names]

DNS.*number*	Name to access web server. Configure additional names by adding more 'DNS.x' lines with increasing numbers in place of 'x' in the example above. Ex. DNS.1 = tibitdev-web DNS.2 = tibitdev-api DNS.3 = www.MCMS.com
IP.*number*	IP of machine. Configure additional interfaces by adding more 'IP.x' lines with increasing numbers in place of 'x' in the example above. Ex. IP.1 = 127.0.0.1 IP.2 = 10.1.10.225

3. Use apt-get to install the package.

Note: By default, MCMS PON Manager will deploy over HTTPS utilizing a self-signed certificate created during installation. THIS IS NOT SAFE FOR A PRODUCTION ENVIRONMENT. If installing MCMS PON Manager for any use other than evaluation, you should use legitimate certificates. Step 4 describes how to update the self-signed certificate and key with valid certificate and key.

- Use the following command to install PON Manager with the default self-signed certificate:

```
sudo apt install ./tibit-ponmgr_R4.0.0_all.deb
```

- *Optional* | Use the following command to install PON Manager with the customized self-signed certificate configured in Step 2 above:

```
sudo TIBIT_PONMGR_CERT_REQ=/path/to/file.conf apt \
install ./tibit-ponmgr_R4.0.0_all.deb
```

Configuration steps:

Ensure that you have root/sudo access to modify files.

1. Navigate to the directory: '/etc/tibit/ponmgr/'
   - Within this directory, there is one PON Manager configuration file that will need to be edited with a text editor of your choice.
   - Update the PON Manager configuration files to match the settings for connectivity to MongoDB.
     - user_database.json – Stores configuration for database to be used for user authentication (see Table 1: PON Manager user_database.json file)
     - databases.json – Stores list of all available PON databases in PON Manager (see Table 2: PON Manager databases.json file)
     - recovery_email_configuration.json – Configures SMTP email server for sending user password recovery emails (see Table 3: PON Manager recovery_email_configuration.json file)

2. Restart Pon Controller process (tibit-poncntl):

```
sudo systemctl restart tibit-poncntl.service
```

Table 1: PON Manager user_database.json file

Key	Description
host	Hostname/IP Address of the MongoDB server hosting your user database.
name	Name of your user database. Default user database name is 'tibit_users'. *(If it doesn't exist, it will be created)*
port	MongoDB port number.
auth_enable	Boolean value determining if the MongoDB server at *host:port* is using authentication.
auth_db	Name of your MongoDB authentication database.

Key	Description
	(Used when auth_enabled = true)
username	The username of the MongoDB user to authenticate with. *(Used when auth_enabled = true)*
password	The password of the specified MongoDB user. *(Used when auth_enabled = true)*
tls_enable	Boolean value specifying whether the MongoDB server at *host:port* is using encryption.
ca_cert_path	The local path to the encryption certificate.
compression	Boolean value specifiything whether to enable snapy compression for MongoDB connections.
dns_srv	Boolean value specifying whether the MongoDB server is using a DNS seed list.
db_uri	Raw MongoDB connection URI. All Other fields are ignored if this is used.
replica_set_enable	Boolean value specifying if the MongoDB server is running as a replica set.
replica_set_name	The name of the MongoDB replica set. *(Used when replica_set_enabled = true)*
replica_set_hosts	List of hosts to be used as the MongoDB replica set. *(Used when replica_set_enabled = true)*
django_key	Cryptographic key used by django. The format of the field is a freeform string, however, Django recommends it be minimum 50 characters with minimum 5 unique characters. For more information on the Django Key, please refer to the MCMS PON Manager User Guide. (Note: A randomized key will be generated during installation if the 'django_key' field is not present.)

Table 2: PON Manager databases.json file

Key	Description
host	Hostname/IP Address of your mongoDB server hosting your PON Controller database

Key	Description
name	Name of your PON Controller database. Default PON Controller database name is 'tibit_pon_controller'. *(If it doesn't exist, it will be created)*
port	MongoDB server port number.
auth_enable	Boolean value determining if the MongoDB server at *host:port* is using authentication.
auth_db	Name of your MongoDB authentication database. *(Used when auth_enabled = true)*
username	The username of the MongoDB user to authenticate with. *(Used when auth_enabled = true)*
password	The password of the specified MongoDB user. *(Used when auth_enabled = true)*
tls_enable	Boolean value specifying whether the MongoDB server at *host:port* is using encryption.
ca_cert_path	The local path to the encryption certificate.
dns_srv	Boolean value specifying whether the MongoDB server is using a DNS seed list.
db_uri	Raw MongoDB connection URI. All Other fields are ignored if this is used.
replica_set_enable	Boolean value specifying if the MongoDB server is running as a replica set.
replica_set_name	The name of the MongoDB replica set. *(Used when replica_set_enabled = true)*
replica_set_hosts	List of hosts to be used as the MongoDB replica set. *(Used when replica_set_enabled = true)*

Table 3: PON Manager recovery_email_configuration.json file

Key	Description
host	Host to use for sending email
port	Port to use for the SMTP server defined above *Type: String*

Key	Description
use_tls	Use TLS connection *'true' or 'false'* *Type: String*
user	Username for the SMTP server
password	Password for user

For more information on installing the MCMS PON Manager, please reference the R4.0.0 MCMS PON Manager User Guide (R4.0.0-MCMS-PonManager-UserGuide-TN035.pdf). The section "Installation" on page 21 provides details on requirements and dependencies, package contents and installation methods (including Docker).

MCMS Netconf Server Installation

The MCMS Netconf Server provides an interface for managing the PON Controller, Tibit MicroPlug OLTs, and ONU devices using standard NETCONF protocols and tools.

The NETCONF/YANG interface supports the following IETF standards:

- RFC 4742 NETCONF Protocol over Secure Shell (SSH)
- RFC 6241 Network Configuration Protocol (NETCONF)
- RFC 6020 YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)
- RFC 7950 The YANG 1.1 Data Modeling Language

The NETCONF/YANG interface supports the following Broadband Forum standards:

- TR-383, Issue 1, Amendment 3 Common YANG Modules for Access Networks
- TR-385, Issue 2 ITU-T PON YANG Modules

Prerequisites:

- Ensure that you have an active internet connection.
- Ensure that you have your MongoDB Server running nad configured as a Replica Set.
- Ensure the host or VM is running either Ubuntu 18.04 or 20.04 with the latest ubuntu updates
- Root level access is required to install the software.

Installation steps:

1. Transfer/download the .zip file to Ubuntu host or VM.
2. From a Linux shell, unpack the .zip file and change to the following directory:

```
unzip MCMS-R4.0.0-UB2004.zip
cd MCMS-R4.0.0-UB2004/R4.0.0-Netconf-UB8104
```

3. Use apt-get to install the package.

```
sudo apt-get install ./tibit-netconf_R4.0.0_amd64.deb
```

Configuration steps:

Ensure that you have root/sudo access to modify files.

1. Navigate to the directory: '/etc/tibit/netconf/'
   - Within this directory, there is one Netconf configuration file that will need to be edited with a text editor of your choice.
   - Update the Netconf configuration file to match the settings for connectivity to MongoDB.
   - See Table 4: Netconf Initialization File for details on each configuration parameter.

2. Restart Pon Controller process (tibit-poncntl):

   ```
   sudo systemctl restart tibit-netconf.service
   ```

3. Configure the MCMS Netconf Server's MongoDB connection information. Edit the file '/etc/tibit/netconf/NetconfInit.json' and modify the IP address, port number, and database name as required.

   ```
   cat /etc/tibit/netconf/NetconfInit.json
       {
           "Logging": {
               "Filename" : "/var/log/tibit/netconf.log",
               "FileCount" : 3,
               "FileSize" : 1024000,
               "Netconf" : {
                   "Console" : "INFO",
                   "File" : "INFO",
                   "Syslog" : "INFO"
               }
           },
           "MongoDB": {
               "auth_db": "tibit_users",
               "auth_enable": false,
               "ca_cert_path": "/etc/tibit/ca.pem",
               "host": "127.0.0.1",
               "name": "tibit_pon_controller",
               "password": "",
               "port": "27017",
               "tls_enable": false,
               "username": ""
           }
       }
   ```

4. Restart the MCMS Netconf Server using Linux systemd scripts.

   ```
   sudo systemctl restart tibit-netconf
   ```

Table 4: Netconf Initialization File

	Field	Values	Description
Logging			
	Filename	“directory-path”	Directory path where log file is to be stored
	FileCount	integer	The maximum number of log files to store for each module, before overwriting the oldest.
	FileSize	integer	The maximum size in bytes of log file
	Netconf	true or false	
MongoDB			
	auth_db	“string”	Name of the MongoDB database that will be used for user authentication and authorization.
	auth_enable	true or false	Set to true if MongoDB access control is enabled. Set to false if MongoDB access control is not enabled.
	ca_cert_path	“/directory/filename”	Directory path to the file (in PEM format) containing the Root CA certificate that was used to sign the server certificate.
	db_uri	“string”	URI to use to connect to the Mongo database. If specified, all other Mongo database connection settings are ignored. Refer to the MongoDB website for more information on the Mongo Connection String URI format.
	dns_srv	true or false	Set to true to enable DNS-based Mongo database discovery. If set to true, the “host” field must contain an FQDN that will be used to lookup SRV records for the Mongo database. If set to true, then the “port” field is ignored as the port number will be acquired from the DNS SRV record.
	host	“ip-address”	IP address used to access the Mongo database. Example:”127.0.0.1”. If dns_srv is set to true, then this field must contain the FQDN used to lookup DNS records for the Mongo database.
	name	“database-name”	Name of the database.
	port	“network-port”	Network port number to access the Mongo database. Example:”27017”. If dns_srv is set to true, then this field is ignored.
	replica_set_enable	true or false	Set to true to use Mongo replica sets and the replica_set_name and replica_set_hosts parameters in the configuration file
	replica_set_name	“string”	nly applicable if replica_set_enable is set to true. Contains the name of the replicaSet parameter that is common to the members of the set.

	Field	Values	Description
	replica_set_hosts	List of host:ports	Only applicable if replica_set_enable is set to true and dns_srv is set to false. This field contains a list of hostname:port entries used to connect to the Mongo database.
	tls_enable	true or false	Set to true to enable TLS when connecting to MongoDB. Set to false to disable TLS when connecting to MongoDB.
	username	"string"	The username to use to connect to the MongoDB database.
	password	"string"	The password to use to connect to the MongoDB database.
NETCONF			
	System Name	"string"	The Netconf System Name

For more information on installing the MCMS Netconf Server, please reference the R4.0.0 MCMS Netconf Server User Guide (MCMS-R4.0.0-NetconfServer-UserGuide-TN037.pdf). The section "Installation" on page 14 provides details on requirements and dependencies, package contents and installation methods.

MCMS PON Controller Installation

The MCMS PON Controller is a stateless management controller and device driver application for configuring and monitoring the end points in a MicroPlug™ OLT PON network. The PON Controller applies configuration to OLT and ONU devices from documents stored in MongoDB. The PON Controller also collects state information, statistics, and logs from devices and reports the information to higher layer applications through MongoDB.

Prerequisites:

- Ensure that you have an active internet connection.
- Ensure that you have your MongoDB Server running that will manage user authentication and serve as your default database.
- Ensure the host or VM is running either Ubuntu 18.04 or 20.04 with the latest ubuntu updates

Installation steps:

3. Transfer/download the .zip file to Ubuntu host or VM.
4. From a Linux shell, unpack the .zip file and change to the following directory:

```
unzip MCMS-R4.0.0-UB2004.zip
cd MCMS-R4.0.0-UB2004/R4.0.0-PonController-UB8104
```

Note: The example above is for the Ubuntu 18.04 package with third party images and firmware.

5. Use apt-get to install the package.

```
sudo apt-get install ./tibit-poncntl_R4.0.0_amd64.deb
```

Configuration steps:

Ensure that you have root/sudo access to modify files.

6. Navigate to the directory: '/etc/tibit/poncntl/'
   - Within this directory, there is one PON Controller configuration file that will need to be edited with a text editor of your choice.
   - Update the Pon Controller configuration file to match the settings for connectivity to MongoDB.
   - See Table 5: Pon Controller Initialization File for details on each configuration parameter.

7. Restart Pon Controller process (tibit-poncntl):

```
sudo systemctl restart tibit-poncntl.service
```

Table 5: Pon Controller Initialization File

	Field	Values	Description
	interface	"interface-string"	Selects the interface for the PON Controller to search for OLTs. Example: "enp0s8.4090"
	databaseType	"MongoDB"	Selects the type of Database to be used
CNTL	Auth	true or false	Used to enable or disable the Controllers Authentication Engine
	CFG Version	"R4.0.0"	The PON Controller version for the format of this file
	DHCPv4	true or false	Used to enable or disable the Controller's DHCPv4 Engine
	DHCPv6	true or false	Used to enable or disable the Controller's DHCPv6 Engine
	PPPoE	true or false	(Added in R3.0.0) Used to enable or disable the Controller's PPPoE Engine
	UMT Interface	"string"	Prefix to be used for the virtual interface created between the Authentication engine and Authenticator. For example, "tibitvirt" will result in the following virtual interface pair ("tibitvirteap", "tibitvirtumt")
Local Copy			In MongoDB mode, a local copy of the state file is written to the JSON database. This allows another process to get status information without going to the Mongo database
	CNTL-STATE	true or false	Write the Controller State file to the JSON database as well as MongoDB
	OLT-STATE	true or false	Write the OLT State files to the JSON database as well as MongoDB

	Field	Values	Description
	ONU-STATE	true or false	Write the ONU State files to the JSON database as well as MongoDB
Logging			
	Directory	“directory-path”	Directory path where log files are to be stored
	FileCount	integer	The maximum number of log files to store for each module, before overwriting the oldest. (Modules include the PON Controller, TAPI, UMT Relay and the Authentication and DHCP Engines)
	FileSize	integer	The maximum size in bytes of each individual log file
	Tracebacks	true or false	When set to true, any tracebacks that occur during Controller execution will be logged to a separate file in the Logging Directory
MongoDB			Fields for MongoDB mode
	auth_db	“string”	Name of the MongoDB database that will be used for user authentication and authorization.
	auth_enable	true or false	Set to true if MongoDB access control is enabled. Set to false if MongoDB access control is not enabled.
	ca_cert_path	“/directory/filename”	Directory path to the file (in PEM format) containing the Root CA certificate that was used to sign the server certificate.
	db_uri	“string”	URI to use to connect to the Mongo database. If specified, all other Mongo database connection settings are ignored. Refer to the MongoDB website for more information on the Mongo Connection String URI format.
	dns_srv	true or false	Set to true to enable DNS-based Mongo database discovery. If set to true, the “host” field must contain an FQDN that will be used to lookup SRV records for the Mongo database. If set to true, then the “port” field is ignored as the port number will be acquired from the DNS SRV record.
	host	“ip-address”	IP address used to access the Mongo database. Example:”127.0.0.1”. If dns_srv is set to true, then this field must contain the FQDN used to lookup DNS records for the Mongo database.
	name	“database-name”	Name of the database.
	port	“network-port”	Network port number to access the Mongo database. Example:”27017”. If dns_srv is set to true, then this field is ignored.
	replica_set_enable	true or false	Set to true to use Mongo replica sets and the replica_set_name and replica_set_hosts parameters in the configuration file

	Field	Values	Description
	replica_set_name	“string”	nly applicable if replica_set_enable is set to true. Contains the name of the replicaSet parameter that is common to the members of the set.
	replica_set_hosts	List of host:ports	Only applicable if replica_set_enable is set to true and dns_srv is set to false. This field contains a list of hostname:port entries used to connect to the Mongo database.
	tls_enable	true or false	Set to true to enable TLS when connecting to MongoDB. Set to false to disable TLS when connecting to MongoDB.
	username	“string”	The username to use to connect to the MongoDB database.
	password	“string”	The password to use to connect to the MongoDB database.

Appendix A: Method for creating a permanent Network configuration (Netplan)

In the requirements section on Page 10, it shows you the commands to create temporary interfaces to be used by the MCMS management system. The procedure below creates permanent interfaces.

Verify the ethernet interfaces are configured on the ubuntu system

1) Look for your Ethernet Interfaces: "ifconfig" or "ip a".
   <<< make note of interfaces listed >>>
2) There are multiple ways to configure interfaces on Ubuntu 18.04 and 20.04
3) This is the example for creating interfaces via Netplan
4) Netplan is located in the /etc/netplan directory
5) Located here will be a file similar in name to "01-network-manager-all.yaml"
6) Edit this file with your favorite editor such as "nano" or "vi"

user@system:~$ sudo nano /etc/netplan/<net-plan-name>.yaml

Sample Netplan text <<< Make sure indentation is consistent >>>

```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
      eno1:                                   <<< MCMS IP Interface >>>
        dhcp4: no                             <<< No DHCP >>>
        dhcp6: no                             <<< No DHCP >>>
        addresses: [172.16.41.5/24]           <<< Static IPv4 >>>
        gateway4: 172.16.41.1           <<< IPv4 default Gateway >>>
        nameservers:                          <<< DNS Addresses >>>
            search: [local]                   <<< Search on local machine first >>>
            addresses: [8.8.8.8, 8.8.4.4]     <<< external DNS servers >>>
      eno2:                                   <<< PON Controller Interface >>>
        dhcp4: no                             <<< No DHCP >>>
        dhcp6: no                             <<< No DHCP >>>
  vlans:                                <<< Configured VLANs >>>
      vlan4090:                         <<< "l2EthInterfaceName" >>>
        id: 4090                              <<< VLAN number >>>
        link: eno2                            <<< PON Controller Interface >>>
```

After finished editing, Save and exit, then enter "sudo netplan apply" to enable new configuration.

1) Verify ubuntu system has connectivity to Internet.

2) Run “ip a” and look for the interfaces you’ve created. You should see an interface with the IP interface you configured, and an interface called “vlan4090” shown (or the name you chose to call the "l2EthInterfaceName").