

MobileIron Cloud and Common Platform Service

aruba

a Hewlett Packard
Enterprise company

ClearPass

Change Log

Version	Date	Modified By	Comments
0.1 / 0.2	Dec / Mar 2017	Danny Jump	Draft TechNote
1.0	Nov 2018	Danny Jump	First Published Version

Copyright

© Copyright 2018 Hewlett Packard Enterprise Development LP.

Open Source Code

This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett-Packard Company. To obtain such source code, send a check or money order in the amount of US \$10.00 to:

Hewlett-Packard Company
Attn: General Counsel
3000 Hanover Street
Palo Alto, CA 94304
USA

Please specify the product and version for which you are requesting source code. You may also request a copy of this source code free of charge at HPE-Aruba-gplquery@hpe.com.

Contents

- Introduction.....5
- Software Requirements6
- Access to the Extension store7
- Installation and Deployment Guide7
- Pictorial view of the Integration8
- New Extension support in ClearPass 6.78
 - Extensions and IP address configuration support.....8
 - Extensions and web proxy support8
- MobileIron Extension installation using GUI available in 6.7+ 10
- Configuring the MobileIron Extension 13
 - GUI configuration for the Extension 13
- MobileIron Configuration – Common Platform Services [CPS]..... 14
 - Account Creation..... 14
 - Enabling CPS framework..... 16
 - Manually triggering an event..... 18
- ClearPass Policy Manager Configuration..... 22
- Appendix A – Additional Diagnostics & Support 24
 - The Extensions Service..... 24
 - Extension logs and debugging 24
 - Accessing Extension logs within ClearPass ‘Collect Logs’ 25
- Appendix B – MI Cloud Ingestion Performance Observations..... 27



a Hewlett Packard
Enterprise company

www.arubanetworks.com
3333 Scott Blvd
Santa Clara, CA 95054

Phone: 1-800-WIFI-LAN (+800-943-4526)
Fax 408.227.4550

Figures

- Figure 1: ClearPass MobileIron Extension feature matrix5
- Figure 2: Entering HP Passport credentials7
- Figure 3: Pictorial view of ClearPass Policy Manager integration with MobileIron Common Platform Service..8
- Figure 4: Extension Framework GUI9
- Figure 5: Defining the base IP SUBNET and LOCALHOST for the Extensions Framework.....9
- Figure 6: Extensions Framework GUI 10
- Figure 7: GUI Extension Installation 10
- Figure 8: GUI Extension Search..... 11
- Figure 9: GUI Extension Configuration at Install time 11
- Figure 10: GUI Reviewing and Setting the Extension configuration 12
- Figure 11: Adding a MobileIron account..... 14
- Figure 12: Checking the user has the correct roles assigned..... 14
- Figure 13: Adding a role to a user..... 15
- Figure 14: Adding the “Common Platform Services” role to the user 15
- Figure 15: Enable CPS Notifications framework [enableMqtt]..... 16
- Figure 16: Add a new user for CPS Events [mqttUserName & mqttPassword] 16
- Figure 17: Assign a role to this new user - part1..... 17
- Figure 18: Assign the CPS role to this new user - part2 17
- Figure 19: Assign the CPS role to this new user - part3 18
- Figure 20: Creating a Custom Compliance Policy – part1 18
- Figure 21: Creating a Custom Compliance Policy – part2 19
- Figure 22: Creating a Custom Compliance Policy – part3..... 19
- Figure 23: Creating a Custom Compliance Policy – part4..... 20
- Figure 24: Change an endpoint attribute to trigger an event notification – part1 20
- Figure 25: Change an endpoint attribute to trigger an event notification – part2 21
- Figure 26: Simple Enforcement Policies based upon endpoint attributes 22
- Figure 27: Device status set to RETIRED..... 23
- Figure 28: Checking on the Extensions service and how to start/stop the service 24
- Figure 29: Using the GUI to change the DEBUG level..... 24
- Figure 30: : Extension logs location in 'Collect Logs' diagnostic GZ file..... 26

Introduction

This TechNote covers the setup, configuration, and monitoring of the ClearPass Extension for MobileIron Cloud and MobileIron Common Platform Services (CPS) and Event Notification Extension. This Extension provides two key features.

The first is adding support for MobileIron Cloud. MobileIron Cloud has not been previously supported, this Extension adds support for Cloud version R56 and above.

The second feature supported in this Extension, is the Common Platform Services Event Notifications. This provides for a near-real-time notification feed as discussed below, to allow ClearPass to maintain an up-to-date view of the managed devices, without the need to constantly poll. In this release we are supporting ClearPass in MobileIron Cloud starting in R56.

Note that ClearPass has supported MobileIron Core for several years, our support for this does not change. At this time this Extension could complement an existing MobileIron Core deployment but we have not verified interoperability with Core, we will complete this soon.

Figure 1: ClearPass MobileIron Extension feature matrix

Product \ API	MI API's supported	Native ClearPass Polling	Native ClearPass Polling + Extension CPS API [Hybrid deployment]	MobileIron Extension CPS API
Pre-Core 9.5	V1	Yes	Yes*	No
Core 9.5 +	V1 + CPS	Yes	Yes	Yes
Cloud R56 +	CPS	No	No	Yes

* For the Pre-Core 9.5 in Hybrid mode, only the Native ClearPass Polling API V1 are supported, the CPS API's are not available in the pre 9.5 Core so adding the Extension to add real-time updates is not supported.

As discussed, the Extension has the ability to ingest endpoint attributes (Core 9.5+ & Cloud R56+) and to receive a near real-time [*in testing about 5 seconds*] feed of endpoint changes within the customer's MobileIron tenant. ClearPass subscribes and receives real time event notifications for 5 distinct use-cases.

1. A New Device Added
2. A Device Retired/Deleted
3. A Device changes state to "out of Compliance"
4. A Device changes state to "in Compliance"
5. A Device is Wiped

When one of the above events occurs, the MobileIron EMM places the event notification into a message queue. All active ClearPass nodes that subscribe to that notification event queue, then can receive that message.

If no ClearPass nodes are active, then the message queue server retains that event notification for **maximum duration of up to 3 hours**, later when an active ClearPass node connects, it's able to consume this stored event message. Messages that exceed the maximum duration are purged.

In comparison, the legacy approach [still available] is to poll the tenant every hour and ingest all of the endpoint data, then update the delta changes into the EndpointDB. The obvious issues with the legacy approach is that if a device goes out of compliance, ClearPass won't know of the state change until the next poll. Similarly if a new device is added, typically the access-policy is that when a SmartDevice accesses the network, ClearPass checks to ensure it's a known managed device. In the legacy approach access would be denied until the next poll had completed. Utilizing the full polling capabilities in conjunction with the event notifications allows a near real-time local view of all of the managed Endpoints.

By comparison, the legacy approach [still available] requires polling the tenant every hour, ingesting all of the endpoint data, computing the delta changes for each endpoint and updating the EndpointDB. One issue with the legacy approach is that if a device goes "out of Compliance", ClearPass can't determine the state change until the next poll cycle. Another problem is that when a new device is added to the enterprise, the SmartDevice access-policy requires ClearPass to ensure that it's a known managed device, which can't be determined until the next poll cycle completes, thus the new device access is denied until then. By utilizing both polling with event notification, ClearPass is afforded a new real-time local view of all managed Endpoints. (latency is kept to a minimum).

Below, we cover installation and configuration of the Extension, configuration within MobileIron, and finally ClearPass configuration. Additionally, we document a solution which allows a device to be 'tagged' as in or out of compliance. This creates an event notification and allows for testing of the end-to-end workflow.

Installation of the MobileIron ClearPass Extension is performed either via the REST API interface, or the simplified GUI introduced in ClearPass Policy Manager v6.7, this is the preferred method. Access to the APIs is through the following URL https://<ClearPass_IP>/api-docs.

Software Requirements

The minimum software version required for CPPM is 6.7.2. At the time of writing, version 6.7.6 is available and the recommended release. CPPM runs on hardware appliances with pre-installed software or as a Virtual Machine under the following hypervisors. Hypervisors that run on a client computer such as VMware Player are not supported.

- VMware ESXi 5.5, 6.0, 6.5 or higher
- Microsoft Hyper-V Server 2012 R2 or 2016 R2
- Hyper-V on Microsoft Windows Server 2012 R2 or 2016 R2
- KVM on CentOS 6.6, 6.7, or 6.8.

The versions of MobileIron supported with this Extension are

- MobileIron Cloud R56 or later
- MobileIron Core 9.5.0 or later

This is the first version which enables the installation, configuration and operation of the Extensions via the GUI. To use the simplified GUI Extension installation, details start on Page 11, else to use the legacy REST API approach, details start on Page 14.

Access to the Extension store

Access to the Extension Store to download Extensions is simplified in ClearPass 6.7. The ability to download Extensions from the store and to validate support entitlement for access to the Software Updates Portal (e.g. Posture & Profile Data Updates, Software Updates, & Skins) now uses the HPE Passport account credentials that are associated with the customers' ClearPass licenses. This is configured where previously the subscription-id was defined, under **Administration -> Agents and Software Updates -> Software Updates** as shown below. Ensure you enter your HPE Passport credentials to enable Extension download capabilities.

Figure 2: Entering HP Passport credentials

The screenshot shows the 'ClearPass Policy Manager' interface. The left sidebar has 'Administration' selected, with 'Agents and Software Updates' and 'Software Updates' highlighted. The main content area is titled 'Administration » Agents and Software Updates » Software Updates'. A red box highlights the 'HPE Passport Credentials' section, which includes 'Username:' and 'Password:' input fields. Below this is a table for 'Posture & Profile Data Updates'.

Update Type	Data Version	Data Created	Last Update	Last Updated	Update Status
Posture Signature Updates*	-	-	-	-	Needs Update
Windows Hotfixes Updates*	1.2173	2017/10/23 04:21:15	File	2017/11/21 11:12:44	Updated 4 days ago
Endpoint Profile Fingerprints*	2.545	2017/10/23 22:45:29	File	2017/11/21 11:12:45	Updated 4 days ago

* Automatic download and install is disabled
To manually import Posture & Profile Data Updates, refer to Help for this page.

Installation and Deployment Guide

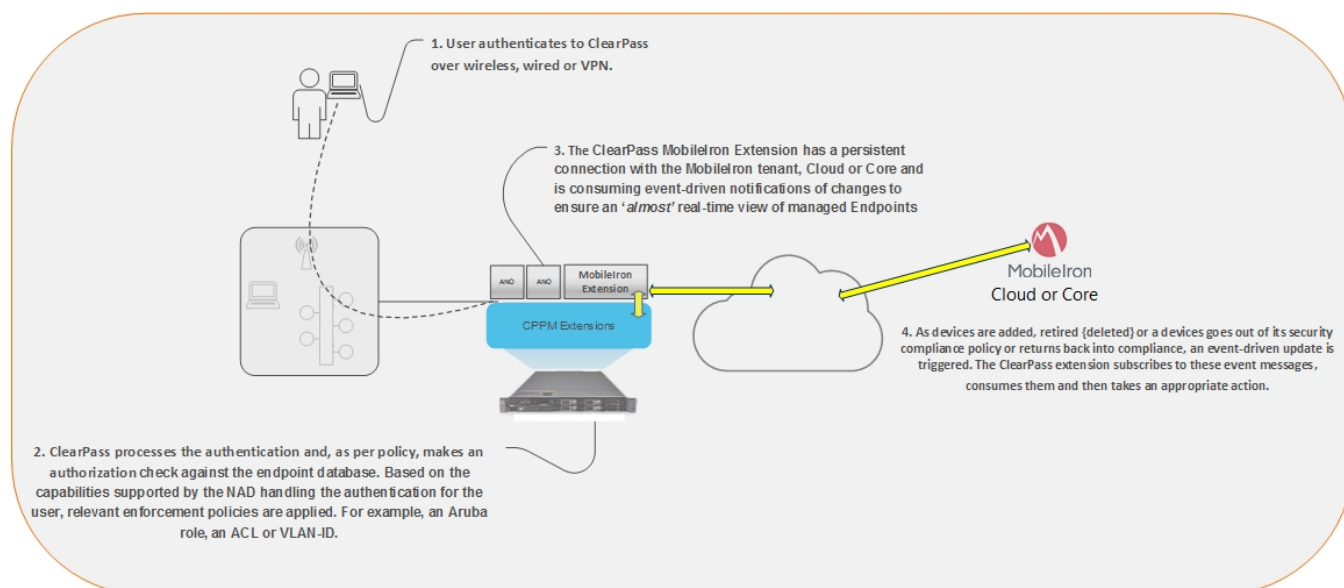
The generic ClearPass installation and deployment guide is located here:

https://www.arubanetworks.com/techdocs/ClearPass/6.7/Aruba_DeployGd_HTML/Default.htm#About%20ClearPass/Intro_ClearPass.htm

Pictorial view of the Integration

The diagram below shows a pictorial overview of the components and how they interact with each other.

Figure 3: Pictorial view of ClearPass Policy Manager integration with MobileIron Common Platform Service



New Extension support in ClearPass 6.7

With the release of 6.7, several new features enhance the functionality of the Extension framework. Previously, all Extension installation and operation tasks required use of the API Explorer to interoperate with the Extension and the underlying framework. This functionality has been exposed with a new GUI. The GUI is accessed from within the Guest UI and is shown below, **Administration -> Extensions**.

Extensions and IP address configuration support

The 6.7 release provides the ability to define the extension framework base IP network and to define the static IP address of the individual extensions. Use the latter when deploying extensions into a cluster and for the ability to set a fixed IP address for the same extension across the cluster regardless of which ClearPass node(s) it was installed on.

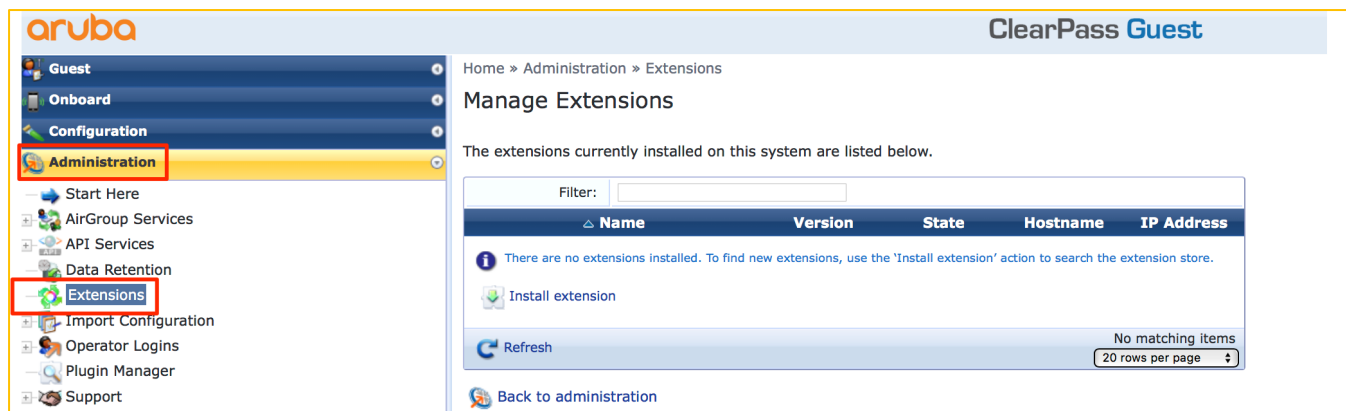
Extensions and web proxy support

Prior to 6.7 support for web proxy was limited to the installation of the Extensions. Starting in ClearPass 6.7, Extensions now support communications with 3rd parties via a web proxy. If a web proxy is defined in ClearPass Policy Manager, then an Extension use that configuration.



Note that the Policy Manger web proxy configuration is ONLY read at by the Extension at installation time. If the web proxy configuration is changed in Policy Manager, then the Extension must be re-installed, so the new settings are re-read and bonded to the Extension.

Figure 4: Extension Framework GUI

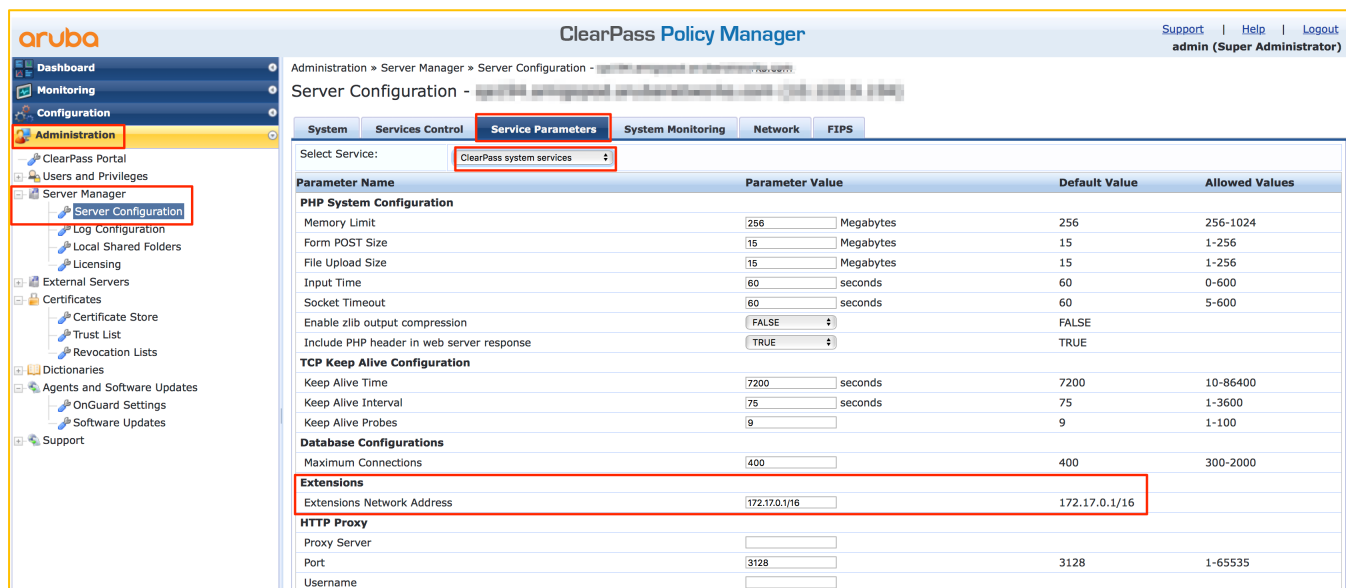


Configuring the base Extension IP subnet, is defined within Policy Manager as shown below under **Administration -> Server Manager -> Server Configuration [chose your node] Service Parameters [ClearPass system service]**. The default address 172.17.0.1/16, is the non-routed address of the ClearPass node itself. The IP addresses range for the extensions depends upon the network prefix used.



Note that the subnet defined here for the Extension framework must fall within the following subnet range 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 as defined by RFC1918.

Figure 5: Defining the base IP SUBNET and LOCALHOST for the Extensions Framework



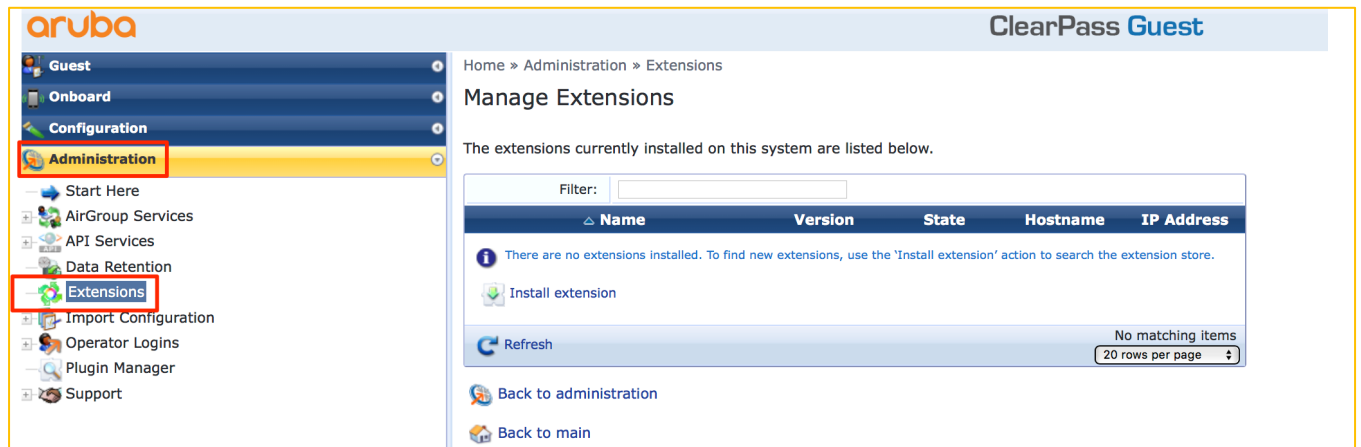
Note when changing the Extension base IP address requires the Extension service to be restarted.

Changing the “Extensions Network Address” range becomes necessary when either the MGMT or DATA interface uses an address in the Extension default range of 172.17.x.x/12. Set the new network address range as needed then restart the Extension service for this change to take effect.

MobileIron Extension installation using GUI available in 6.7+

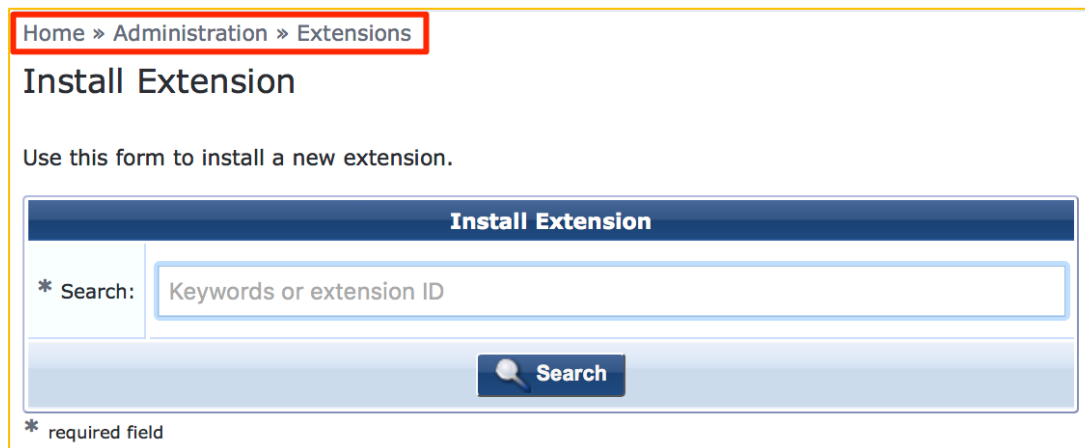
Starting in ClearPass 6.7, a Graphical User Interface (GUI) was introduced to make the process of interacting with the Extension framework easier. To access the Extension GUI, from the **Guest System**, under **Administration** find the **Extension** User Interface as shown below.

Figure 6: Extensions Framework GUI



From here, click on '**Install Extension**', and the search box below appears.

Figure 7: GUI Extension Installation



Enter either the Store-ID, or enter the name or partial name of the Extension, and click on 'Search'. See the example below:

Figure 8: GUI Extension Search

Home > Administration > Extensions

Install Extension

Use this form to install a new extension.

Install Extension								
* Search:	<input type="text" value="Mobile"/>							
Results:	<table border="1"><thead><tr><th>Name</th><th>Version</th><th>State</th></tr></thead><tbody><tr><td> MobileIron MobileIron MQTT and MDM integration.</td><td>1.0.0</td><td> Stopped</td></tr></tbody></table>		Name	Version	State	MobileIron MobileIron MQTT and MDM integration.	1.0.0	Stopped
Name	Version	State						
MobileIron MobileIron MQTT and MDM integration.	1.0.0	Stopped						
<input type="button" value="Search"/>								

* required field

Click on the Extension and then the “Install” option, and if necessary, set the IP address. Note it can be set later if required, e.g. you want to set a permanent static address for the extension.

Figure 9: GUI Extension Configuration at Install time

Home > Administration > Extensions



Install Extension







Use this form to install a new extension.

Install Extension	
Extension:	MobileIron MobileIron MQTT and MDM integration.
Extension Settings	
Start:	<input type="checkbox"/> Start the extension after installation
IP Address:	<input type="text" value="17.17.0.55"/> Enter IPV4 address to allocate to this extension, from the network 172.17.0.1/16. Leave blank to automatically assign an IP address.
<input type="button" value="Install"/>	

After the Extension has been installed, if the option to automatically start was not selected, review the Extension configuration and adjust as needed. Notice the options to Start, Delete, Reinstall or Show Logs and the option to review and set the Extension configuration.

Figure 10: GUI Reviewing and Setting the Extension configuration

Name	Version	State	Hostname	IP Address
 MobileIron MobileIron MQTT and MDM integration.	1.0.0	 Stopped	a6614a550331	

 Show Details  Start  Delete  Reinstall  Show Logs  Configuration


Extension Configuration

* Configuration:


```
{
  "logLevel": "INFO",
  "verifySSLCerts": true,
  "cppmUser": "admin.user",
  "cppmPassword": "admin.password",
  "mobileIronUri": "https://MI_URL",
  "mobileIronUserName": "user.name",
  "mobileIronPassword": "user.password",
  "enableMqtt": false,
  "mqttUri": "ssl://MI_MQTT_URL:8883",
  "mqttUserName": "user.name",
  "mqttPassword": "user.password",
  "enableFullUpdate": false,
  "fullUpdateIntervalMinutes": 10080
}
```


Provide JSON configuration parameters for the extension.

Restart: Restart extension after updating configuration



* required field

 Refresh1Showing 1 - 1 of 1

20 rows per page 

A copy of the default MobileIron Extension is shown above, this will need to be modified for your deployment and the extension started/re-started as appropriate.

Configuring the MobileIron Extension

GUI configuration for the Extension

Regardless of whether the Extension was deployed with the GUI or with the legacy REST API's, a set of mandatory parameters must be collected to allow the default configuration {shown below} to be updated.

```
{
  "logLevel": "INFO",
  "verifySSLCerts": true,
  "cppmUser": "admin.user",
  "cppmPassword": "admin.password",
  "mobileIronUrl": "https://MI_URL",
  "mobileIronUserName": "user.name",
  "mobileIronPassword": "user.password",
  "enableMqtt": true,
  "mqttUrl": "ssl://MI_MQTT_URL:8883",
  "mqttUserName": "user.name",
  "mqttPassword": "user.password",
  "enableFullUpdate": false,
  "fullUpdateIntervalMinutes": 10080
}
```

The default configuration needs to be changed to match your environment. Each configuration option is described below.

- **logLevel:** Defines the logging level of the Extension. ("DEBUG", "INFO", "WARN", "ERROR")
- **verifySSLCerts:** Defines if the Extension verifies the presented certificate by MobileIron.
- **cppmUser:** A CPPM Admin account used to allow internal communications.
- **cppmPassword:** The password associated with the above Admin account.
- **mobileIronUrl:** The URL of the MobileIron tenant hostname/IP address.
- **mobileIronUserName:** The MobileIron username.
- **mobileIronPassword:** The MobileIron username's password.
- **enableMqtt:** Enables/Disables the Event Notification real-time update framework in the Extension.
- **mqttUrl:** The MQTT URL that the extension connects to receives event notifications.
- **mqttUserName:** The MQTT username.
- **mqttPassword:** The MQTT username's password.
- **enableFullUpdate:** If set to true, the Extension will ingest all of the endpoint information and populate the ClearPass EndpointDB. Default = false.
- **fullUpdateIntervalMinutes:** The frequency the full ingest service runs, in minutes. Default = 7 days.



Unless instructed by Aruba TAC, leave the logLevel at the default value.

MobileIron Configuration – Common Platform Services [CPS]

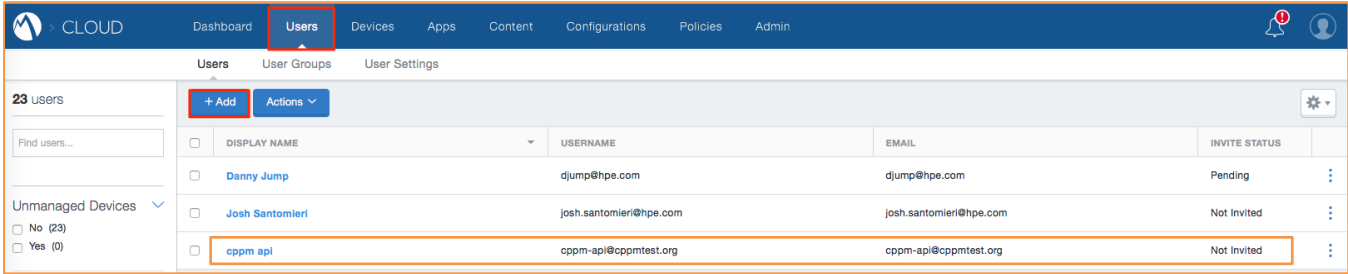
Below we cover the configuration required in the MobileIron environment. To properly configure the CPPM extension, first collect a number of items from the MobileIron tenant. Within the configuration, three username/password combinations required.

Account Creation

CPPM-Credentials. The first pair [cppmUser/cppmPassword] is used by the Extension to call ClearPass API's that allow the creation/deletion/updating of endpoint data. You can use an existing CPPM Local Admin account or better, create a new dedicated read-only Admin account for this function.

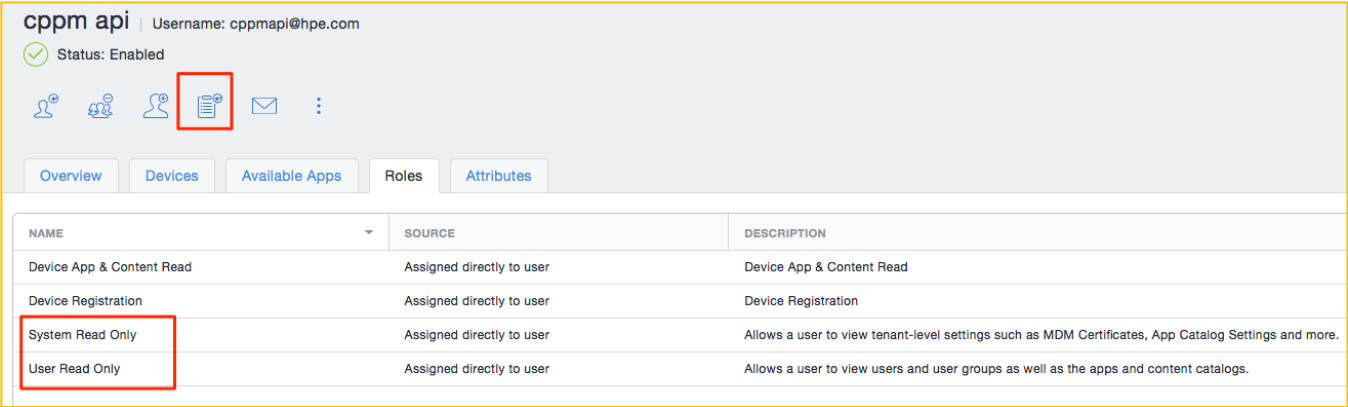
MobileIron-Tenant-Credentials. The second pair [mobileIronUserName/mobileIronPassword] is used by the Extension to communicate with the MobileIron instance when calling MobileIron API's to retrieve all of the endpoint data which is then populated into the ClearPass EndpointDB. For the **MobileIron-Tenant-Credentials**, it is recommended that an account be created in MobileIron dedicated for this function. Although these credentials can be an Administrator account best practice recommends that a new account with the roles shown below in Figure12 be used. To create the account. **Users -> +Add**

Figure 11: Adding a MobileIron account



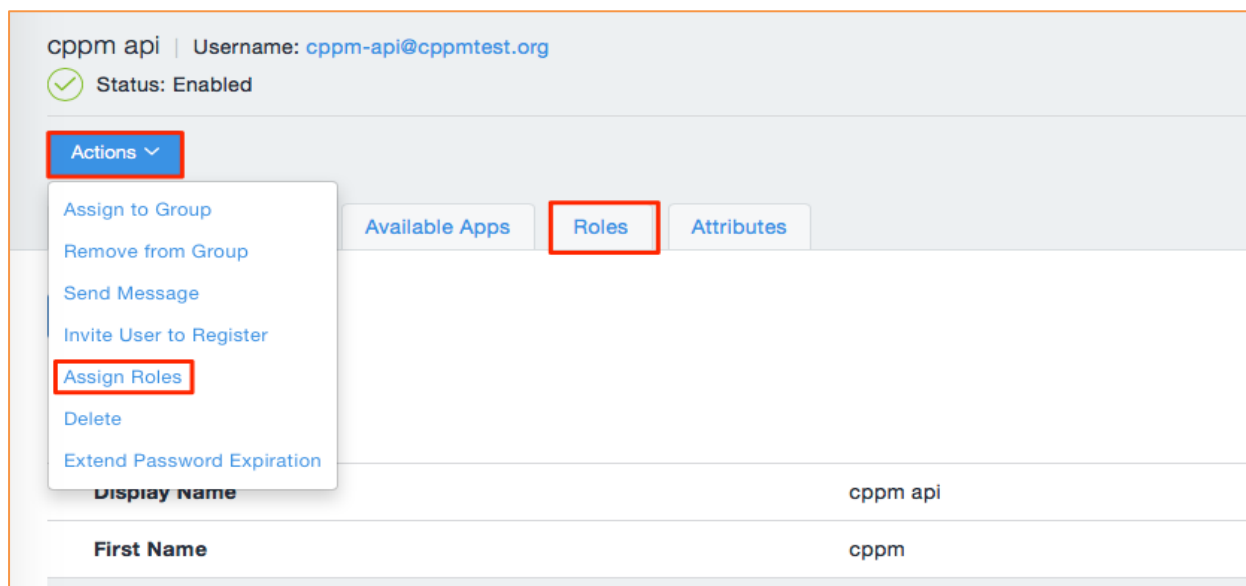
Then select the user and ensure that the user has 'System Read Only' & 'User Read Only' roles assigned.

Figure 12: Checking the user has the correct roles assigned.



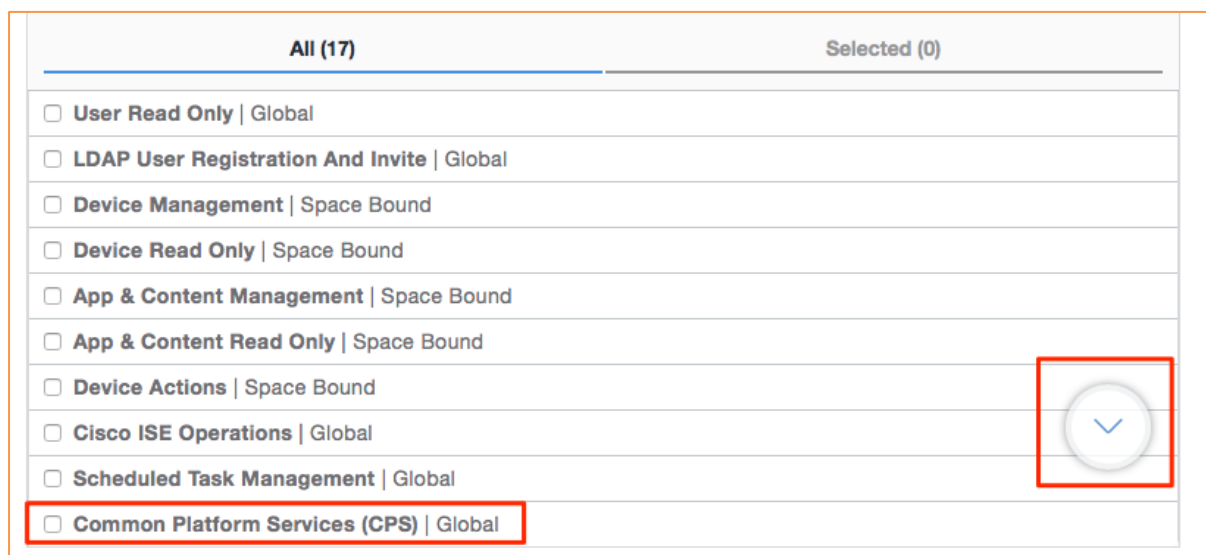
MobileIron-MQTT-Credentials. The third pair [mqttUserName/mqttPassword] is used by the Extension to communicate with the MobileIron Event Notification Services. This service sends the real-time-notifications. It's possible to use the same account as above or a separate account. If using the same account, ensure that the **Common Platform Services** role has been added to the user account. In the Roles tab, click on **Actions -> Assign Roles**

Figure 13: Adding a role to a user



Next add the CPS role to that user. Note, to add the role **Common Platform Services** scroll down as highlighted below to locate the role. Select the role, and confirm.

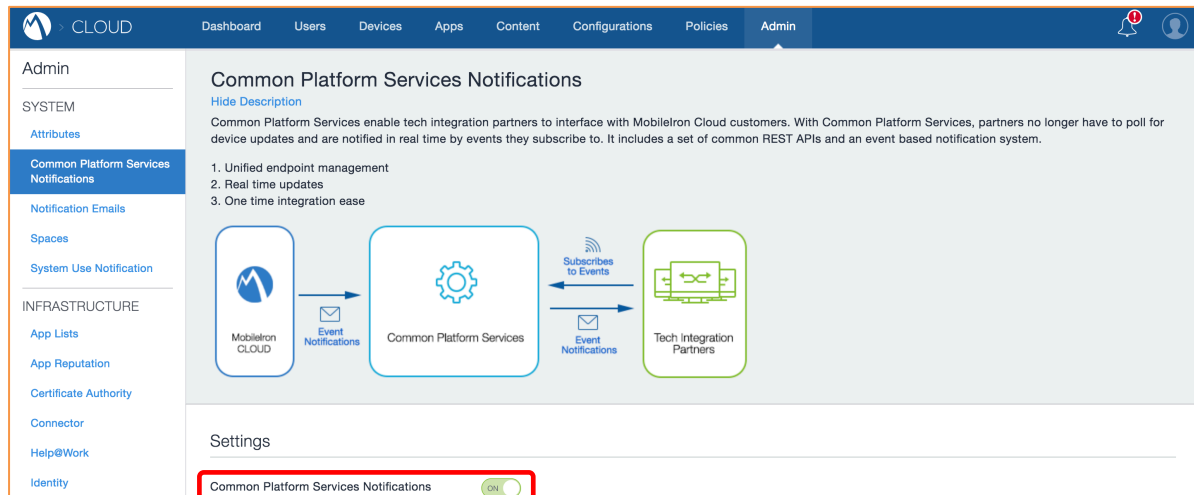
Figure 14: Adding the "Common Platform Services" role to the user



Enabling CPS framework

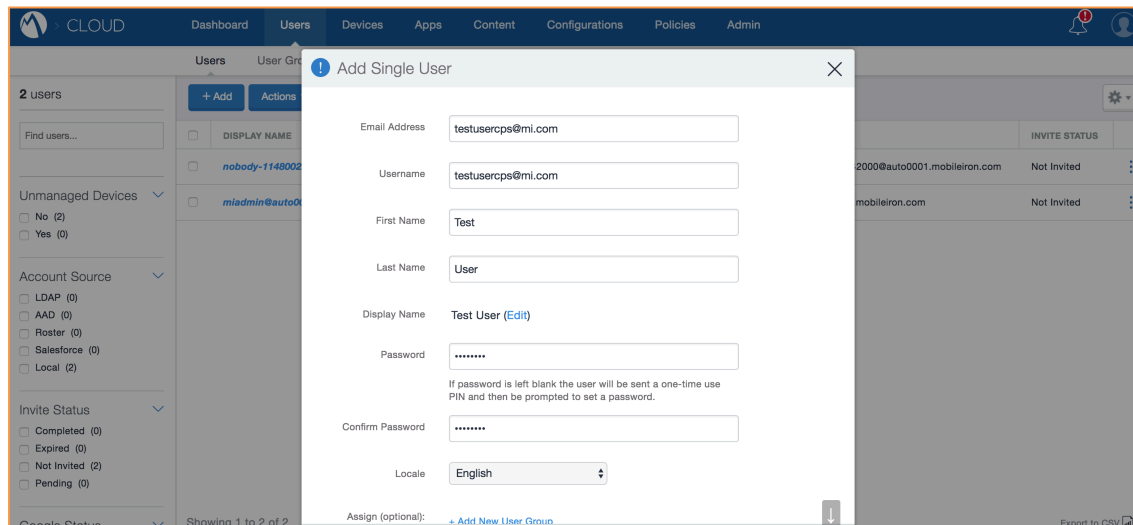
After creating the accounts, to use the *near-real-time* event notification service, additional configuration steps are required. Navigate to the **Admin** tab on MI Cloud portal, click on **CPS Notifications** sub-tab and select enable the service. (see below)

Figure 15: Enable CPS Notifications framework [enableMqtt]



If the CPS role has not been added to an existing user, then create a new CPS user: Create a user by navigating to **Users** tab on the admin portal.

Figure 16: Add a new user for CPS Events [mqttUserName & mqttPassword]



Select the user created and assign "**Common Platform Services**" role to the user.

Figure 17: Assign a role to this new user - part1

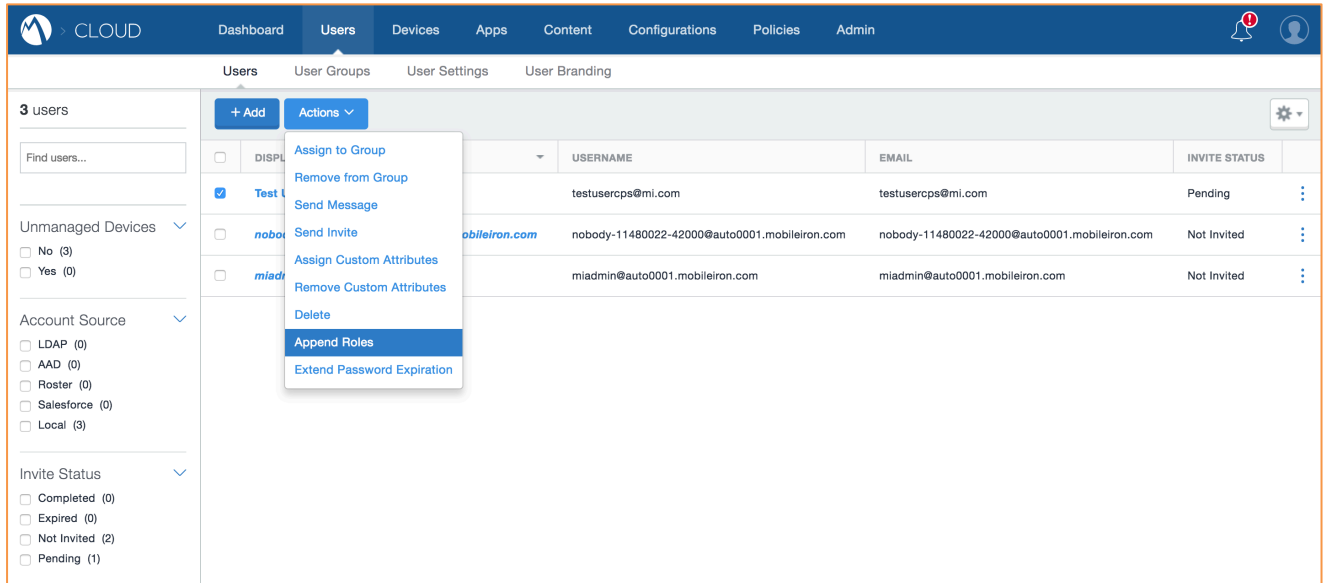


Figure 18: Assign the CPS role to this new user - part2

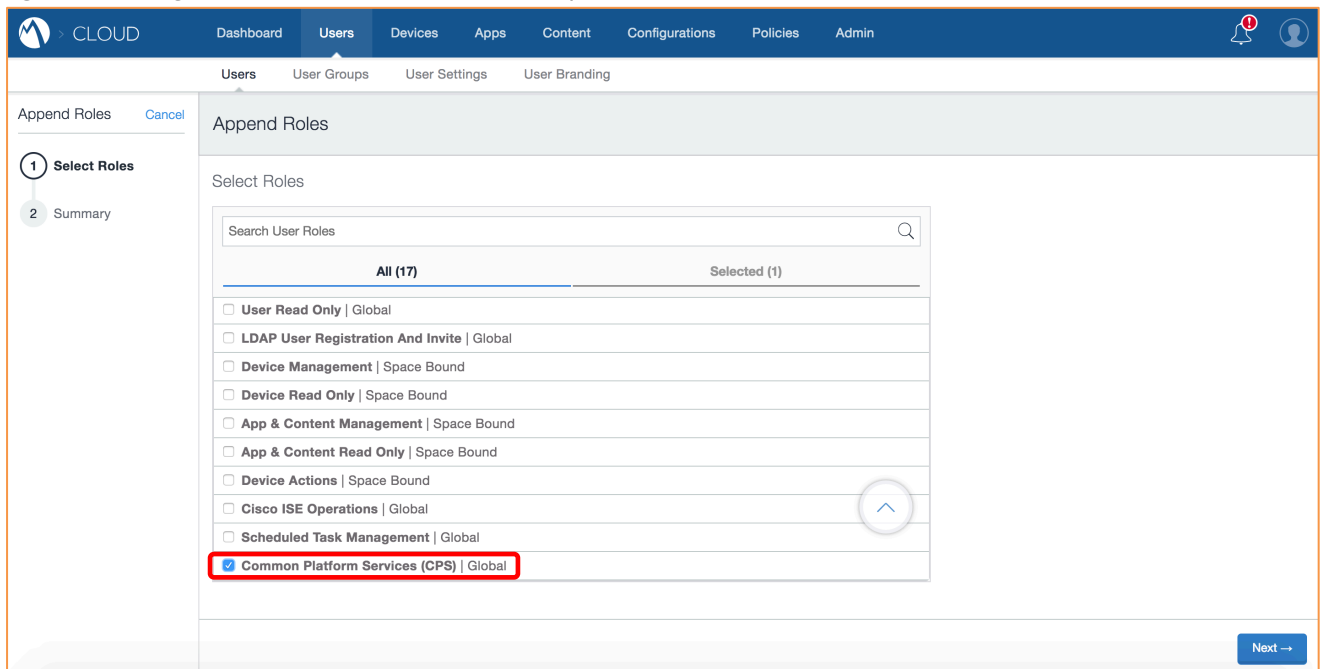
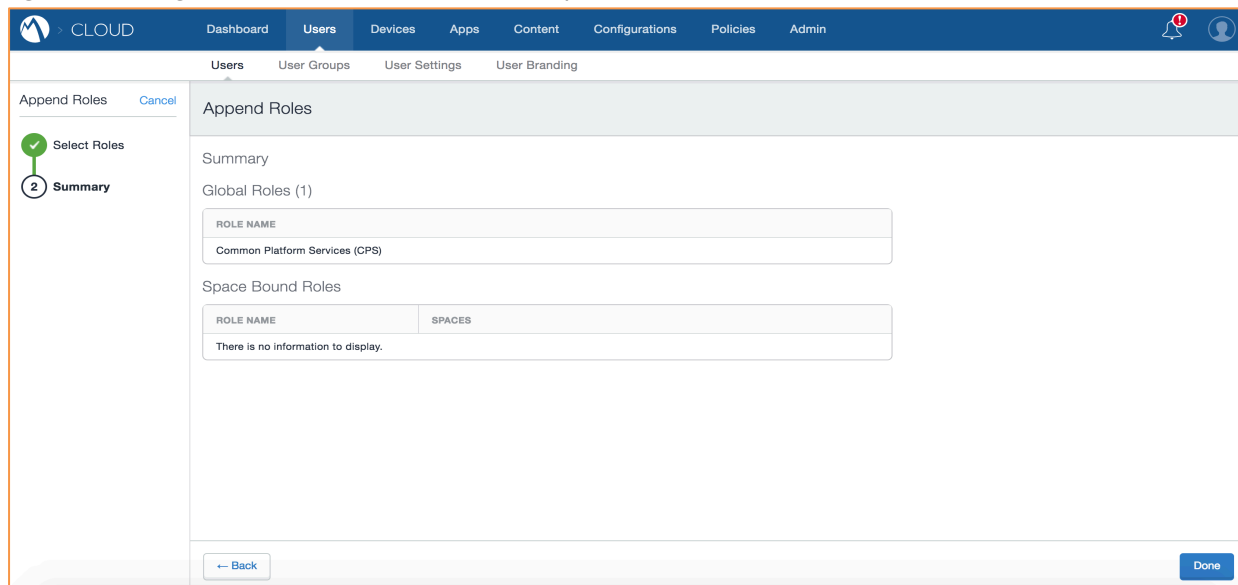


Figure 19: Assign the CPS role to this new user - part3



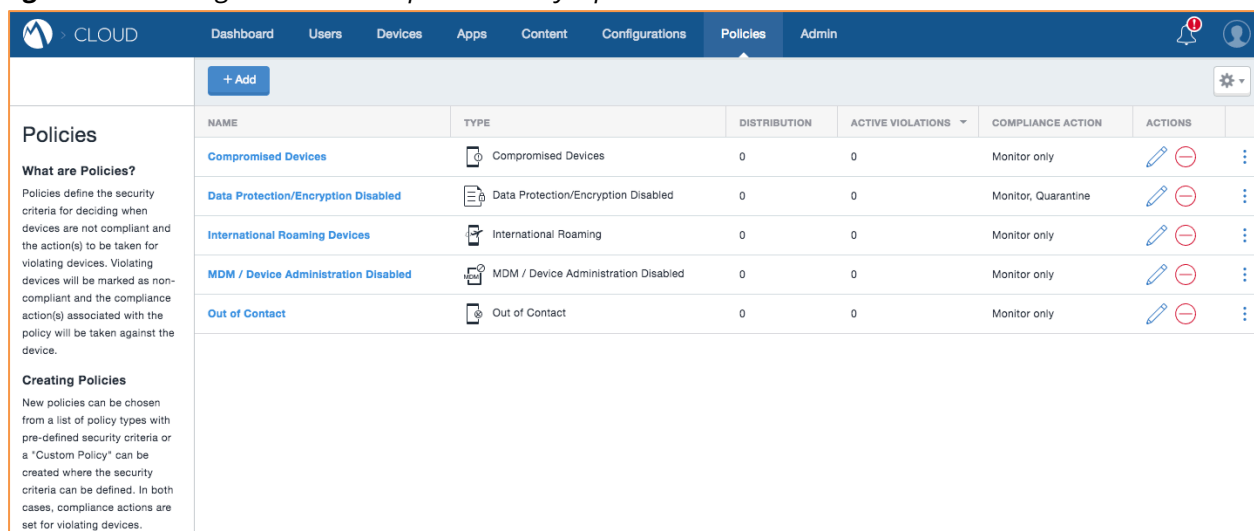
Manually triggering an event

As an example of how you could use custom attributes to simulate a compliance policy to force devices in and out of non-compliant state: This section allows for the creation of an event to test the end-to-end workflow of the system.

A good use case would be to toggle a value of a custom attribute (say, `nacCompliant`) for devices which have moved out of compliance from **true** to **false** and then, use the attribute to take actions on the device. There are multiple ways to force compliance actions on the device to render it non-compliant. Please refer the below steps:

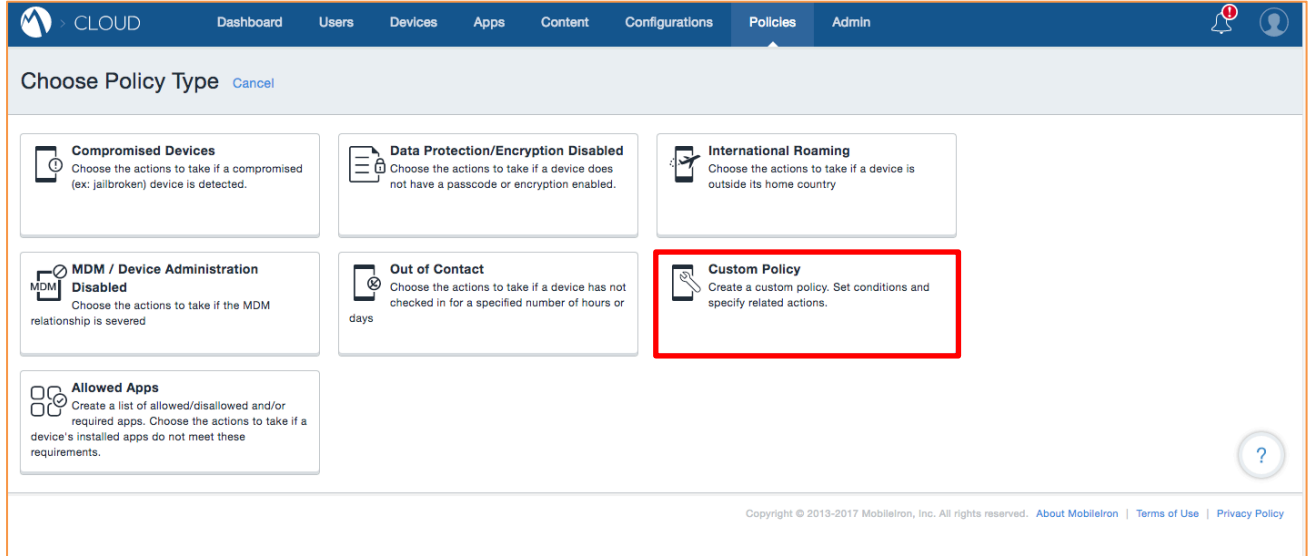
#1: Create a compliance action policy: Navigate to Policies on the admin portal menu bar and click on **Add**

Figure 20: Creating a Custom Compliance Policy – part1



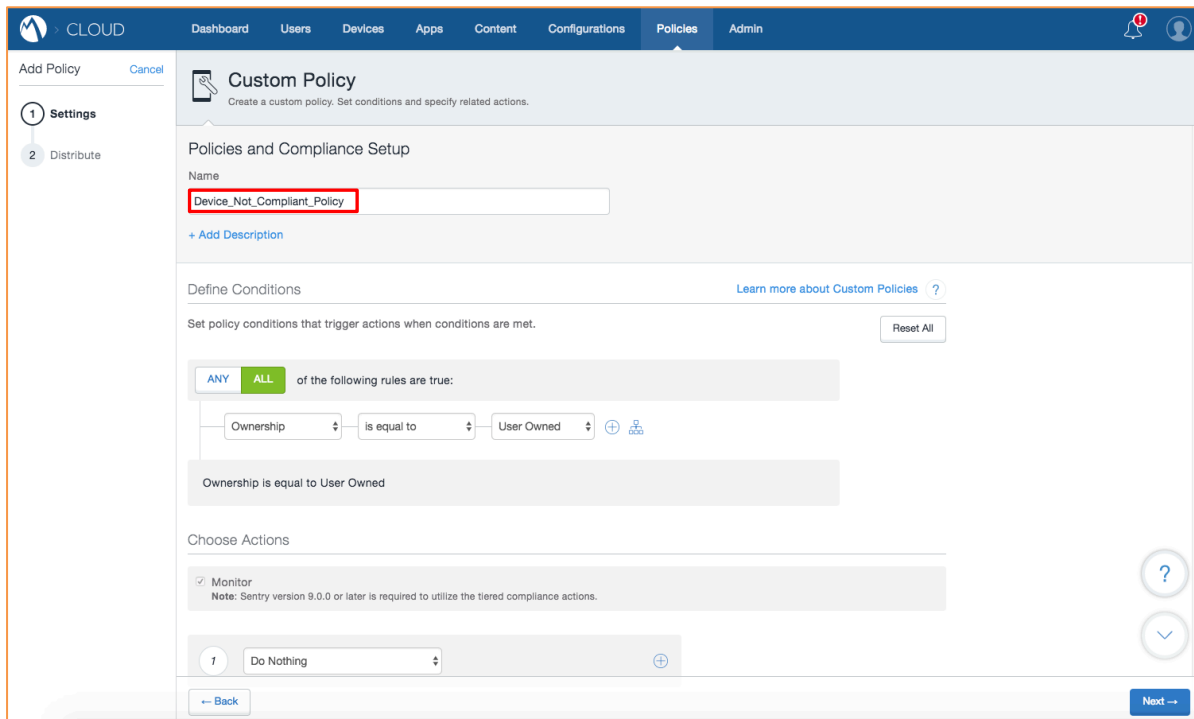
#2: Add policy rule/definition to determine the criteria for a device going non-compliant: Click on “**Custom Policy**” option to create a custom compliance action policy.

Figure 21: Creating a Custom Compliance Policy – part2



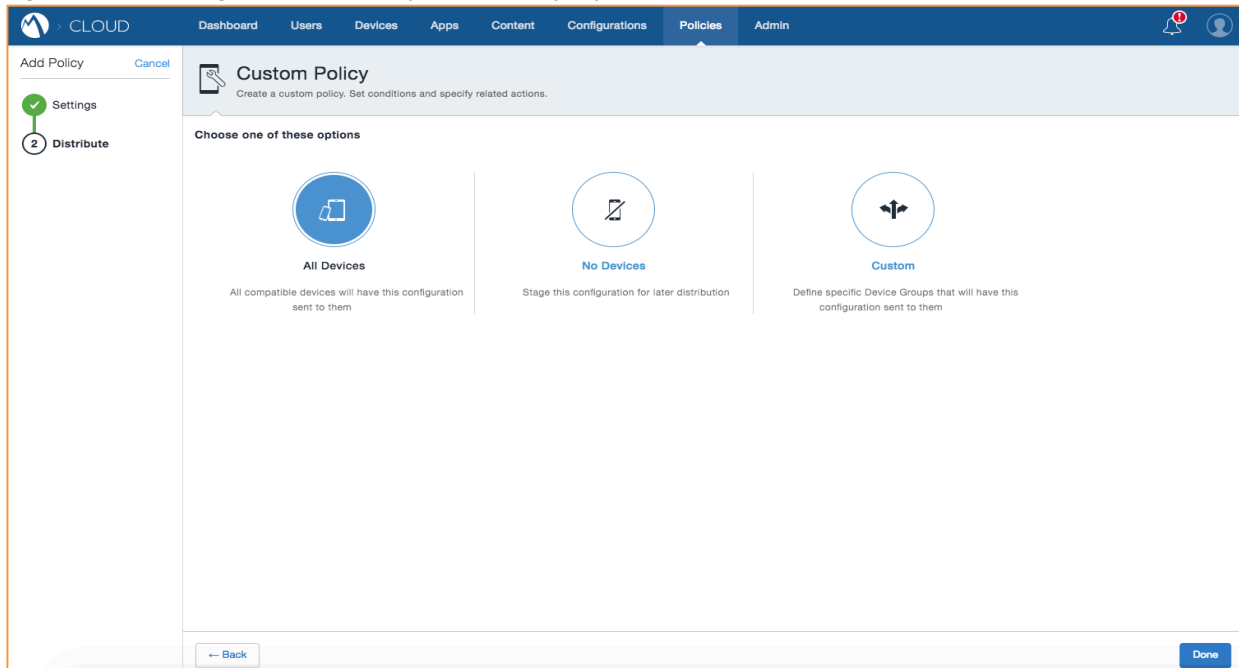
Choose a custom policy rule: Enter a **policy name** and create a criteria query to specify policy rule, E.g. If the device ownership is 'User Owned' or the device OS is type 'Android' mark it as non-compliant and click **Next**.

Figure 22: Creating a Custom Compliance Policy – part3



#3: Distribute the policy to all devices and click **Done**

Figure 23: Creating a Custom Compliance Policy – part4



#4: Perform state changes on the device to match the criteria, this would mark the device non-compliant, in which case, **device_not_compliant** events would be triggered e.g. Changing device ownership to 'User Owned' and initiate force check-in and device sync.

Figure 24: Change an endpoint attribute to trigger an event notification – part1

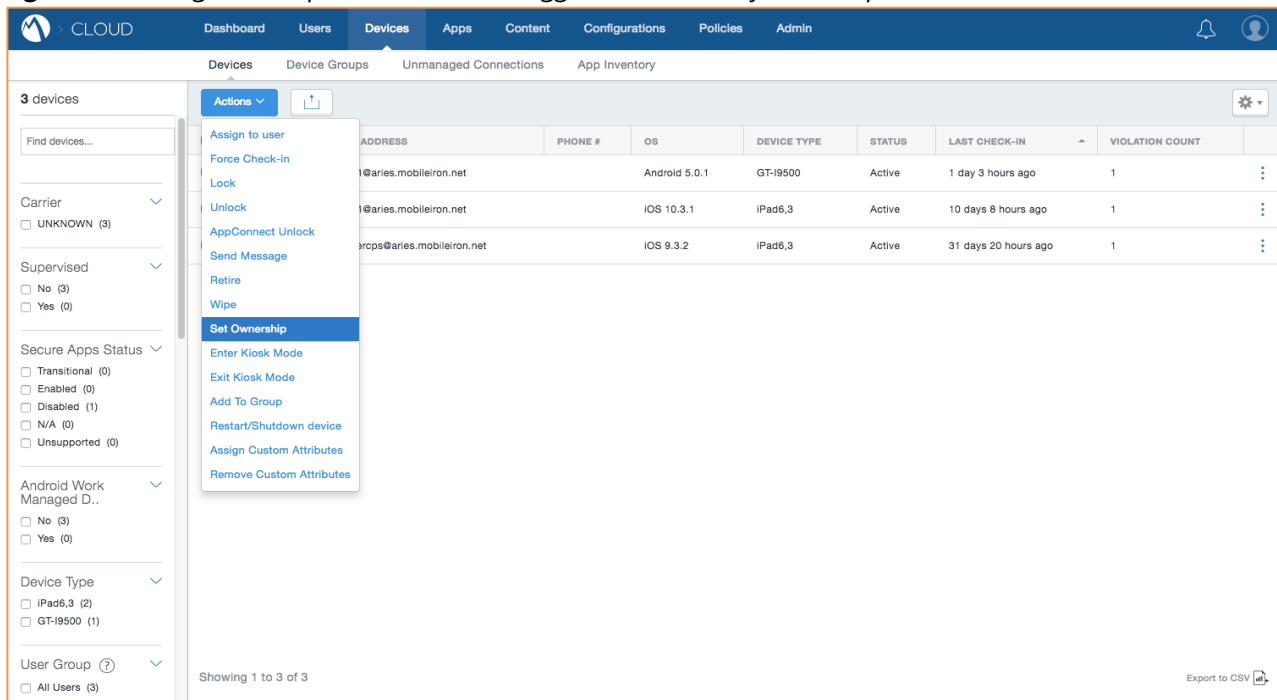
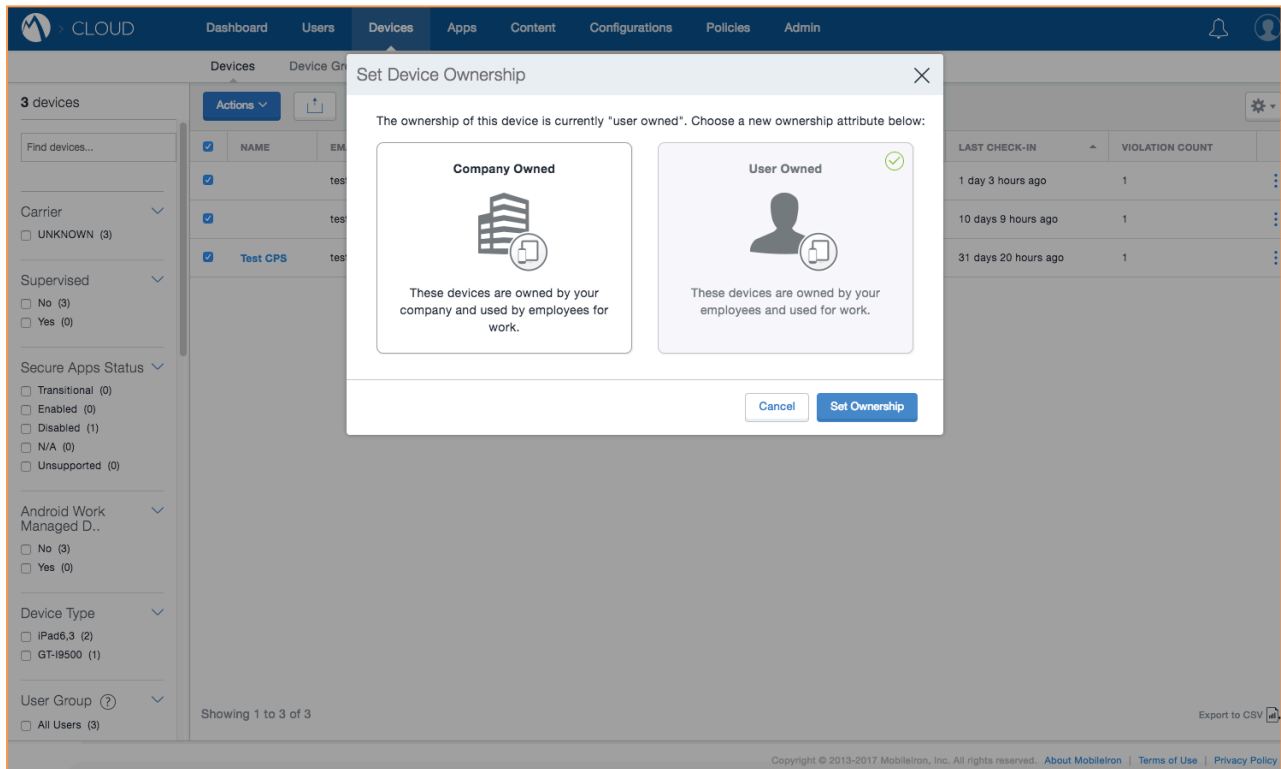


Figure 25: Change an endpoint attribute to trigger an event notification – part2



Certain policy rules can affect a large set of devices at one time; these aren't generally recommended.

- OS is iOS
- Last check-in is 10 hrs ago
- Ownership Type is 'User Owned'

#5: To bring the device back to compliance, either perform reverse device state changes or delete the compliance action policy and initiate force-sync on devices.

ClearPass Policy Manager Configuration

The final part is the configuration on ClearPass Policy Manager. Depending on how you use the integration will ultimately define how you configure the interaction between MobileIron Cloud/Core and the ClearPass Extension and Policy Manager.

If you plan on using the Extension to interface with MobileIron Cloud then configure the Extension and its associated polling. Following this, configuring ClearPass Policy Manager configuration is no different in how you'd authenticate and authorize any other device, it's really about how you use the endpoint database attributes in your authorization policy checks for role-mapping or your enforcement policy.

If you plan on using the Extension to complement the existing MobileIron polling, then overall this is a hybrid deployment. Using the built-in polling to ingest the endpoint details once per day in addition to using the Extension to 'trickle-feed' changes into the endpoint-database as they happen. This hybrid deployment can remove the need for the lengthy and regular polling, carefully consider your polling strategy and how often you poll.

Regardless of which deployment you configure, as noted above the power of the integration is how you use the endpoint data base attributes. As an example here are a few simple examples.

Figure 26: Simple Enforcement Policies based upon endpoint attributes

Configuration » Enforcement » Policies » Edit - MobileIron Enforcement Policy

Enforcement Policies - MobileIron Enforcement Policy

Summary Enforcement Rules

Enforcement:

Name: MobileIron Enforcement Policy

Description:

Enforcement Type: RADIUS

Default Profile: [Allow Access Profile]

Rules:

Rules Evaluation Algorithm: First applicable

Conditions	Actions
1. (Endpoint:Status EQUALS RETIRED)	[Deny Access Profile]
2. (Endpoint:Compliant EQUALS true)	Quarantine Role, Create SNOW incident tickets, Send Quarantined Device Notification (SMS)
3. (Endpoint:OS EQUALS IOS) AND (Endpoint:OS Version NOT_CONTAINS 11)	Old-OS-ArubaRole, BlackBerry endpoint out_of_Compliance, Create SNOW incident tickets
4. (Endpoint:Compromised EQUALS true)	Quarantine Role, Send Quarantined Device Notification (SMS)
5. (Endpoint:Quarantined EQUALS true)	Email Security Response Team, Create SNOW incident tickets, [Deny Access Profile]

To add, a little more clarity, if a device is retired from within MobileIron then the endpoint status flag is set accordingly. Within your enforcement policy you need to add a rule {#1} as shown above, where, when the **status** of the endpoint is set as **RETIRED**, the enforcement action would be to Deny Access. This can be adjusted to fit your own needs, as an example if you detected a device trying to access the network which has been deleted/retired from the system, you may want to have a work flow that drops the device into a captive portal role which directs the user to contact the helpdesk for assistance.

Figure 27: Device status set to *RETIRED*

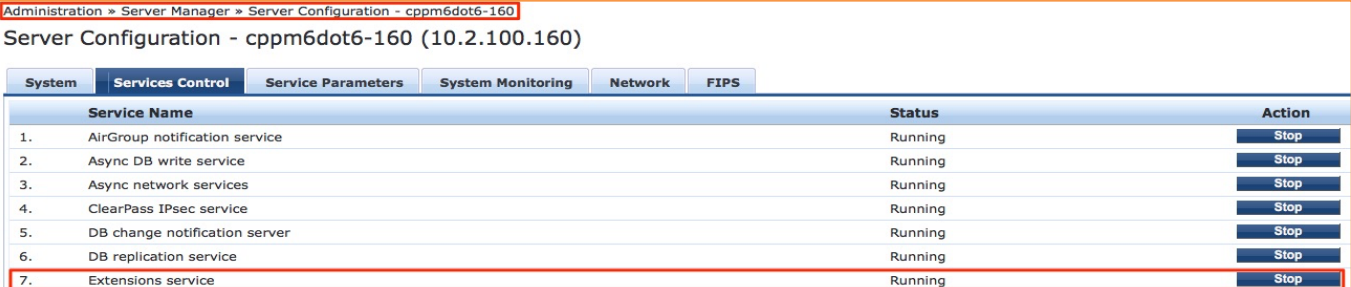
	Attribute	Value
1.	Compliant	= true
2.	Compromised	= false
3.	Last Check In	= 2018-02-14 23:10:13
4.	Manufacturer	= Apple Inc.
5.	OS	= IOS
6.	OS Version	= 9.3
7.	Quarantined	= false
8.	Registration Date	= 2018-02-09 15:54:47
9.	Serial Number	= ee5c69f0bf31
10.	Source	= MobileIron
11.	Status	= RETIRED
12.	UDID	= 6xLyq8QzRgiKuFADeUTKqWgrsrwNrzVBpyL7jIQP
13.	User ID	= oscarjimenez@auto0001.mobileiron.com

Appendix A – Additional Diagnostics & Support

The Extensions Service

The ClearPass Extension is supported by a new system service that was initially added in 6.6. This service should be running. Note that restarting this service will affect **all** deployed and running Extensions. To check on the state and to restart the service, go to **Administration > Server Manager > Server Configuration [select a cppm node] > Service Control**. From here start/stop the Extension service. By default, this service is automatically started.

Figure 28: Checking on the Extensions service and how to start/stop the service

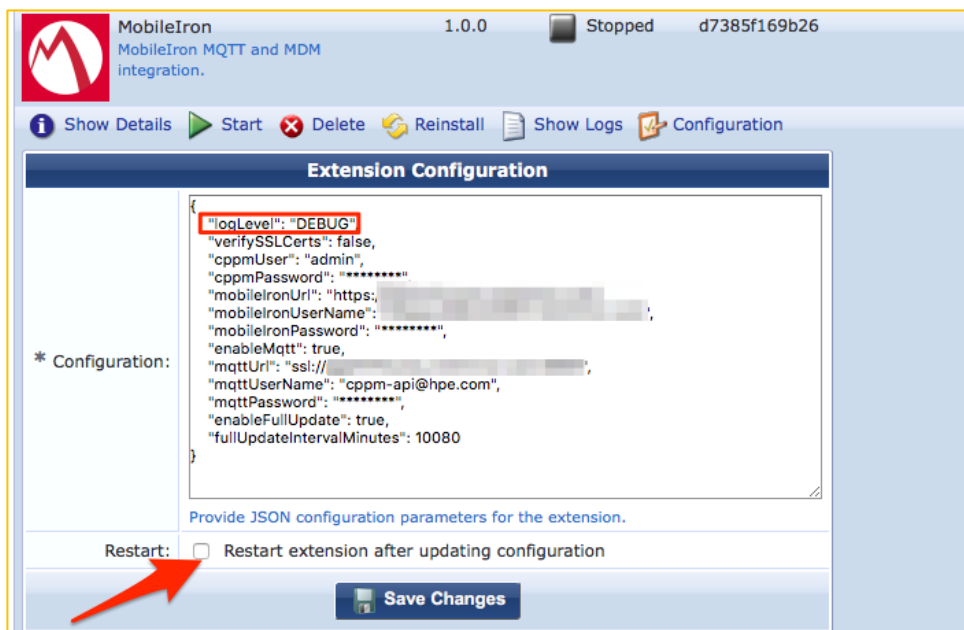


System	Services Control	Service Parameters	System Monitoring	Network	FIPS
	Service Name				
1.	AirGroup notification service				
2.	Async DB write service				
3.	Async network services				
4.	ClearPass IPsec service				
5.	DB change notification server				
6.	DB replication service				
7.	Extensions service				
	Status				
	Running				
	Running				
	Running				
	Running				
	Running				
	Running				
	Running				
	Running				
	Action				
	Stop				
	Stop				
	Stop				
	Stop				
	Stop				
	Stop				
	Stop				

Extension logs and debugging

If there is a need to access the logs from inside the Extension, turn on log collection from the API Explorer. Referencing the configuration previously used, adjust the **"logLevel"** to **"DEBUG"**. In the new 6.7 GUI change the config and restart the Extension as shown below. Logs can then be viewed from the **'Show Logs'**.

Figure 29: Using the GUI to change the DEBUG level



Here are a few examples of 'normal' logs

```
[2017-12-05T10:44:37.651] [INFO] MobileIron - Client reconnecting to
ssl://ppp1234.auto.mobileiron.com:8883.
[2017-12-05T10:44:37.783] [DEBUG] MobileIron - Got a MQTT packet.
[2017-12-05T10:44:37.784] [INFO] MobileIron - Connected to server ssl://ppp1234.auto.mo-
bileiron.com:8883.
[2017-12-05T10:44:37.784] [INFO] MobileIron - Querying for MQTT topics...
[2017-12-05T10:44:37.814] [INFO] MobileIron - Subscribing to da5d1822-5cda-41c0-9507-
dda52597a312/device/compliant, da5d1822-5cda-41c0-9507-dda52597a312/device/wiped,
da5d1822-5cda-41c0-9507-dda52597a312/device/not_compliant, da5d1822-5cda-41c0-9507-
dda52597a312/device/enrolled, da5d1822-5cda-41c0-9507-dda52597a312/device/retired
topic(s).
```

Here are a few logs showing failures.

```
[2017-12-05T10:46:37.897] [DEBUG] MobileIron - Got a MQTT packet.
[2017-12-05T10:47:37.949] [DEBUG] MobileIron - Got a MQTT packet.
[2017-12-05T10:38:00.866] [WARN] MobileIron - MQTT Connection closed.
[2017-12-05T10:38:01.867] [INFO] MobileIron - Client reconnecting to
ssl://ppp1234.auto.mobileiron.com:8883.
[2017-12-05T10:38:01.999] [DEBUG] MobileIron - Got a MQTT packet.
[2017-12-05T10:38:02.000] [ERROR] MobileIron - Connection error!
[2017-12-05T10:38:02.001] [ERROR] MobileIron - { Error: Connection refused: Not author-
ized
```

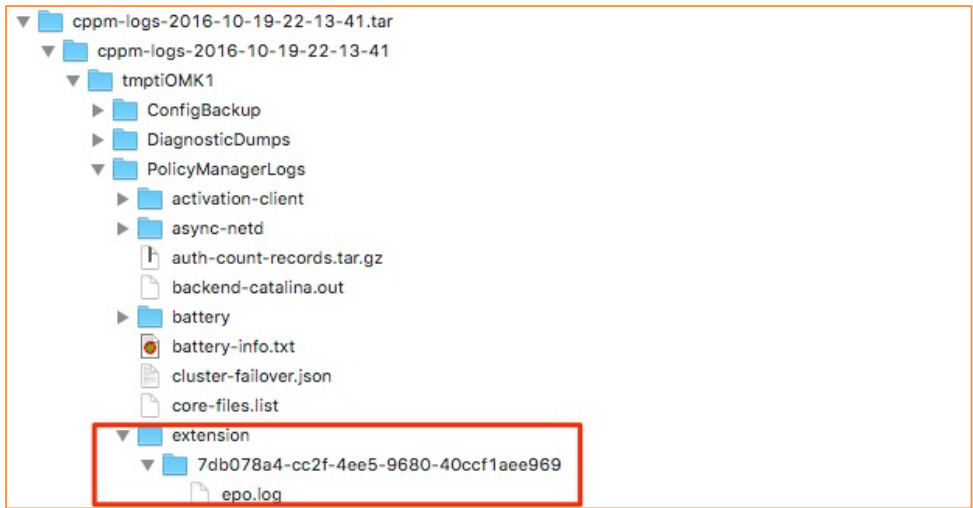
Accessing Extension logs within ClearPass 'Collect Logs'

In addition to the logging of messages that be examined in the Extension as shown above, it's possible to configure the Extension to log messages so that they can be collected and examined via the Policy Manager **'Collect Logs'** system function. This is extremely useful for Aruba TAC.

If there is a requirement for Aruba TAC to investigate a system issue, one of the items they regularly ask for is the system logs to aid with their diagnostic investigation. The ClearPass Extension can write its logs such that they are available and can be collected with all other system diagnostics information when the **'Collect Logs'** function is run. Remember that by default, the logLevel is set to INFO but TRACE, DEBUG, INFO, WARN, ERROR, FATAL can also be set. Any of the levels will display the information for the selected state and lower. For example, if INFO is selected, it will show messages for INFO, WARN, ERROR, FATAL.

After the Logs have been collected and exported from the system, expand the GZ file and locate the Extension logs in the following location **'PolicyManagerLogs->Extension'** as shown below.

Figure 30: : Extension logs location in 'Collect Logs' diagnostic GZ file



Appendix B – MI Cloud Ingestion Performance Observations

During our testing of the integration we performed extensive testing, as part of this process we recorded timing related to the performance of ingesting endpoints from a Cloud tenant. In our testing we had circ 5,000 endpoints.

Your performance for ingesting your tenant data will be dependent on a number of factors which are beyond the scope of this document, but we wanted to provide our experience and observations.

During a new ingest, i.e. a first time sync we recorded over a number of iterations a rate of approximately 1,000 endpoints per 10 minutes. When running a compare, i.e. get all endpoints but update only the ClearPass EndpointDB with new endpoints, changed attributes i.e. very minimal changes the ingest rate was 1,000 endpoints processed in approximately 6 minutes.