

Hackathon Steps

- Setting up APM
 - Step 0: Get sample files
 - Step 1: Login to APM using your tenant URL
 - Step 2: View API documentation
 - Step 3: View APM User documentation
 - Step 4: You will be logged in as Tenant Admin.
 - Step 5: Build an asset model & Ingest or use an existing asset model
 - Step 6: Create permission set for non-admin user
 - Step 7: Create non-admin user
 - Step 8: Login to APM as non-admin user
- Instructions to setup Intel Edison Board
 - Step 1: Set up a new Intel Edison board
 - Step 2: Setup sensors
 - Step 3: Move pfa-container to Edison board
 - Step 4: Untar on Edison board
 - Step 5: Generate keytool
 - Step 6: Download Postman collection & get environment variables
 - Step 7: Change configs (WSS endpoint)
 - Configure Web Socket River
 - Step 8: Run Predix Machine
 - Step 9: Visualize TS data
 - Step 10: Ingest Sample Asset model (Optional)
 - Step 11: Ingest sample TS data (Optional)
 - Step 12: Ingest Alerts & cases (Optional)
 - Step 13: Ingest sample Aviation flight data (Optional)
- Start building an app
 - Step 1: Login to Cloud Foundry using your credentials
 - Step 2: git clone sample micro-service
 - Step 3: git clone sample micro-app
 - Step 4: Visualize Assets
 - Step 5: Push microservice to cloud foundry
 - Step 6: Push microapp to cloud foundry
 - Step 7: Verify microapp in APM
- What just happened?
- Next steps

Pre-requisites:

Software	Version	Notes
Cloud Foundry CLI	Latest	
Git	Latest	
Java SE Development Kit (JDK)	8	
Maven	Latest	Install JDK before installing Maven

Setting up APM

Step 0: Get sample files

```
$ git clone https://github.com/apmdev/tools
```

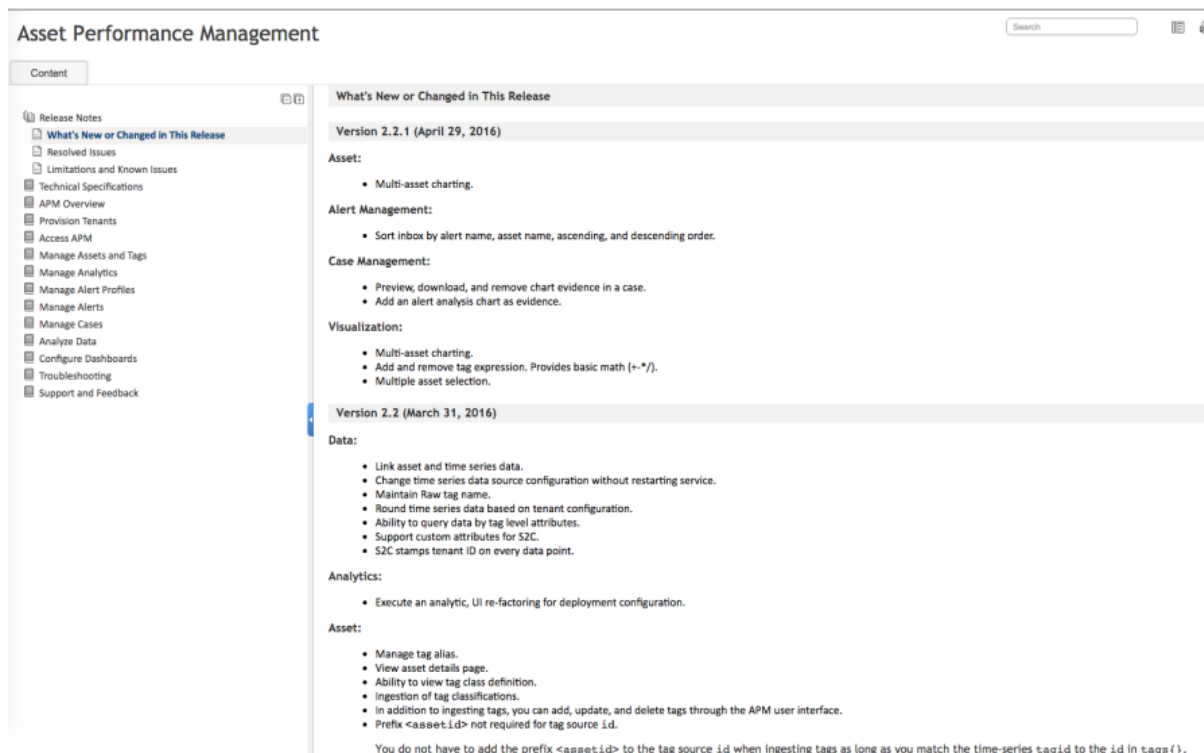
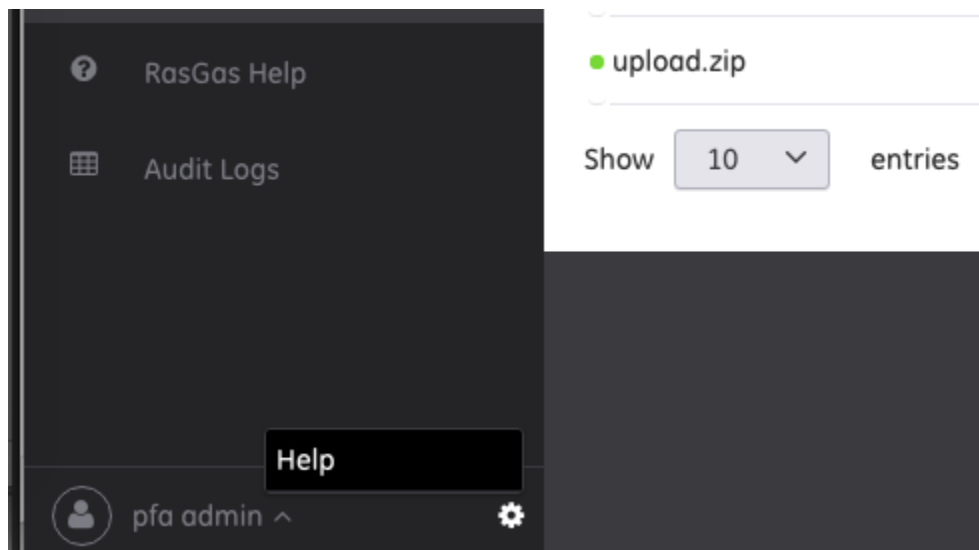
Step 1: Login to APM using your tenant URL

Tenant URL will be provided on a piece of paper
https://apm-tubs-hackapm-basic.run.aws-usw02-pr.ice.predix.io/<tenant_id>

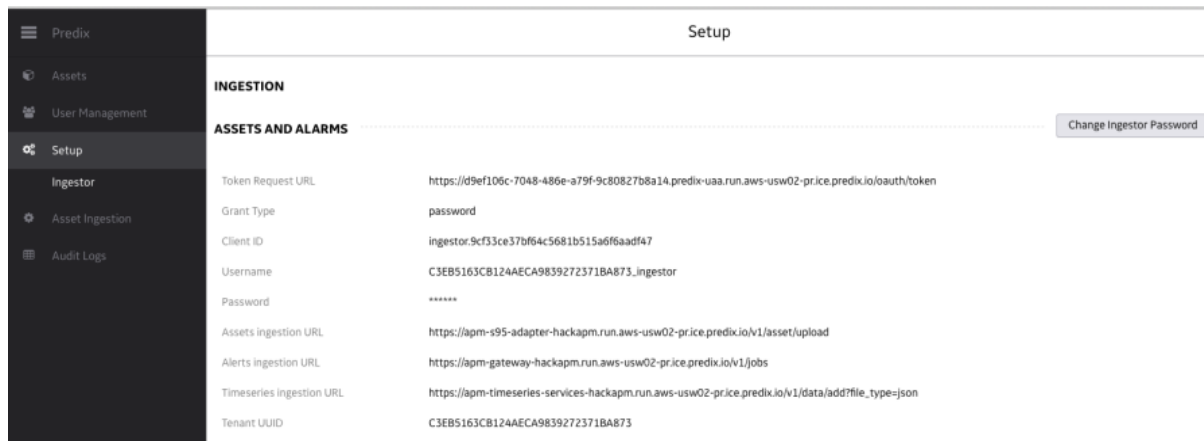
Step 2: View API documentation

<https://apm-apidocs-hackapm.run.aws-usw02-pr.ice.predix.io/>

Step 3: View APM User documentation



Step 4: You will be logged in as Tenant Admin.



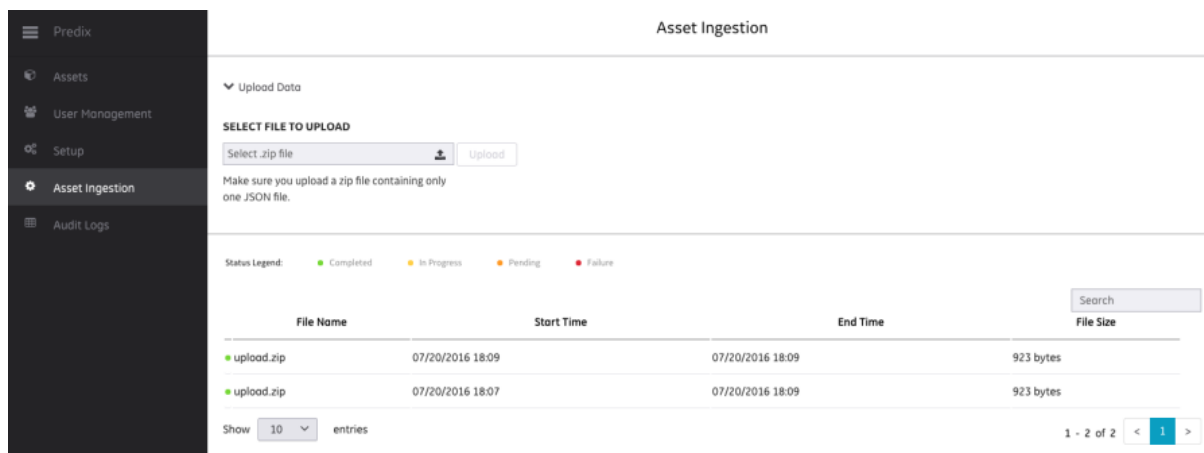
The screenshot shows the 'Setup' page in the Predix interface. The left sidebar has 'Setup' selected, with 'Ingestor' highlighted. The main content area is titled 'INGESTION' and 'ASSETS AND ALARMS'. It displays configuration details for the ingestor, including Token Request URL, Grant Type, Client ID, Username, Password, Assets ingestion URL, Alerts ingestion URL, Timeseries ingestion URL, and Tenant UUID. A 'Change Ingestor Password' button is visible in the top right.

Property	Value
Token Request URL	https://d9ef106c-7048-486e-a79f-9c80827b8a14.predix-uaa.run.aws-usw02-pr.ice.predix.io/oauth/token
Grant Type	password
Client ID	ingestor.9cf33ce37bf64c5681b515a6f6aadf47
Username	C3EB5163CB124AECA9839272371BA873_ingestor
Password	*****
Assets ingestion URL	https://apm-s95-adapter-hackapm.run.aws-usw02-pr.ice.predix.io/v1/asset/upload
Alerts ingestion URL	https://apm-gateway-hackapm.run.aws-usw02-pr.ice.predix.io/v1/jobs
Timeseries ingestion URL	https://apm-timeseries-services-hackapm.run.aws-usw02-pr.ice.predix.io/v1/data/add?file_type=json
Tenant UUID	C3EB5163CB124AECA9839272371BA873

Step 5: Build an asset model & Ingest or use an existing asset model

Go to tools/ingestion_data/intel_board_asset and look for pt_sample_assets.zip

Click on "Asset Ingestion"

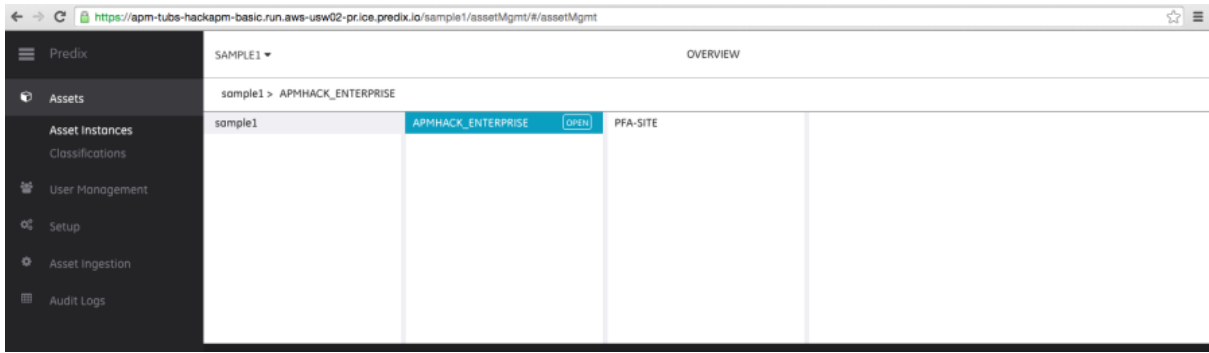


The screenshot shows the 'Asset Ingestion' page. It includes a 'SELECT FILE TO UPLOAD' section with a file selection button and an 'Upload' button. Below this, a status legend indicates 'Completed' (green), 'In Progress' (yellow), 'Pending' (orange), and 'Failure' (red). A table lists uploaded files with columns for File Name, Start Time, End Time, and File Size. Two files named 'upload.zip' are shown, both completed at 07/20/2016 18:09 with a size of 923 bytes. A search bar and pagination controls are also present.

File Name	Start Time	End Time	File Size
upload.zip	07/20/2016 18:09	07/20/2016 18:09	923 bytes
upload.zip	07/20/2016 18:07	07/20/2016 18:09	923 bytes

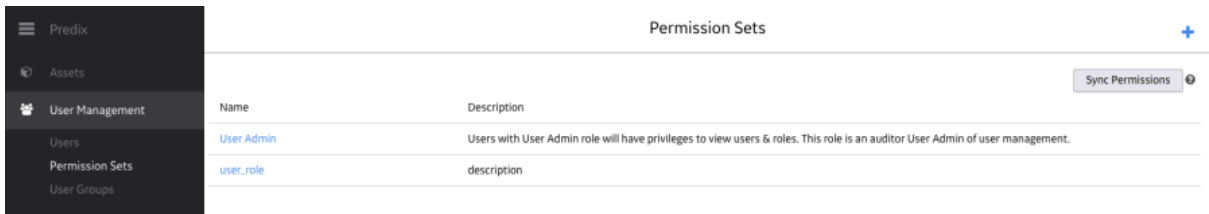
Verify asset ingestion is complete using ingestion logs.

Note: You need to manually refresh the page to get latest status.
Once Asset ingestion is complete you can validate using APM



Step 6: Create permission set for non-admin user

Navigate to "User Management" -> "Permission Sets" Click on +



Check all services for Alert, Analysis, Asset, Time series and Dashboard.
Also pick your tenant specific feature



Pick services that you want and click "Create"

New Permission Set

[Back](#) [Cancel](#) [Create](#)

BASIC INFO

Name*

Description

PERMISSIONS

Select at least 1 permission.

Alerts	<input checked="" type="checkbox"/> Alert Profiles	<input checked="" type="checkbox"/> Alerts
Analysis	<input checked="" type="checkbox"/> Analysis-Create	<input checked="" type="checkbox"/> Analysis-View
Analytics	<input checked="" type="checkbox"/> Manage Analytics catalog	
Asset	<input checked="" type="checkbox"/> Edit Assets	<input checked="" type="checkbox"/> View Assets
Audit Log	<input type="checkbox"/> Audit Logs	

Permission Sets [+](#) [Sync Permissions](#) ⓘ

Name	Description
basic_user	basic_user
User Admin	Users with User Admin role will have privileges to view users & roles. This role is an auditor User Admin of user management.
user_role	description

Step 7: Create non-admin user

Navigate to "User Management" Click on +

Users [+](#)

Search

Sort by: **Name ascending** ▼

Ingestor Role C3EB5163CB124AECA9839272371BA873_ingestor
One Admin sample1-admin
user user1 user1

Navigate to "User Management" Click on +

HINT: Use following format to create username <Tenant_name>_user1

Create a user by filling up the form.

Users

Add User

Username (Case sensitive)	Identity Provider		
<input type="text" value="user2"/>	<input type="text" value="UAA Internal"/>	<input type="button" value="Find"/>	<input type="button" value="Cancel"/>

User not found in the system. Please provide user details.

All fields are required.

First Name	Last Name	Email
<input type="text" value="user"/>	<input type="text" value="two"/>	<input type="text" value="user2@ge.com"/>
Password	Confirm Password	
<input type="password" value="*****"/>	<input type="password" value="*****"/>	
	Match	

Select permission set that you just created and click "Next"

<div><div>Predix</div><div>Assets</div><div>User Management</div><div>Users</div><div>Permission Sets</div><div>User Groups</div><div>Setup</div><div>Asset Ingestion</div><div>Audit Logs</div></div>	<h2>Users</h2> <table><tr><td>user two</td><td>user@ge.com</td><td></td></tr><tr><td>user2</td><td></td><td><input type="button" value="Back"/> <input type="button" value="Next"/></td></tr></table> <p>Select a Permission Set</p> <div><div>basic_user</div><div>User Admin</div><div>user_role</div></div>	user two	user@ge.com		user2		<input type="button" value="Back"/> <input type="button" value="Next"/>
user two	user@ge.com						
user2		<input type="button" value="Back"/> <input type="button" value="Next"/>					

Select an Asset that this user can access. You can pick any number of assets at any level.

<div><div>Predix</div><div>Assets</div><div>User Management</div><div>Users</div><div>Permission Sets</div><div>User Groups</div><div>Setup</div><div>Asset Ingestion</div><div>Audit Logs</div></div>	<h2>Users</h2> <table><tr><td>user two</td><td>user@ge.com</td><td></td></tr><tr><td>user2</td><td></td><td><input type="button" value="Back"/> <input type="button" value="Finish"/></td></tr></table> <p>Permission Set</p> <div><div>basic_user</div><div>basic_user</div></div> <p>SELECT AN ASSET ▾</p> <div><div>APMHACK_ENTERPRISE</div><div>APMHACK_ENTERPRISE</div></div>	user two	user@ge.com		user2		<input type="button" value="Back"/> <input type="button" value="Finish"/>
user two	user@ge.com						
user2		<input type="button" value="Back"/> <input type="button" value="Finish"/>					

Click Finish.

☰

Predix

📦

Assets

👤

User Management

🔧

Setup

⚙️

Asset Ingestion

📋

Audit Logs

👤

Users

📄

Permission Sets

👥

User Groups

🔍

Sort by:


Name ascending

▼

Ingestor Role
C3EB5163CB124AECA9839272371BA873_ingestor
One Admin
sample1-admin
user two
user2
user user1
user1

Step 8: Login to APM as non-admin user

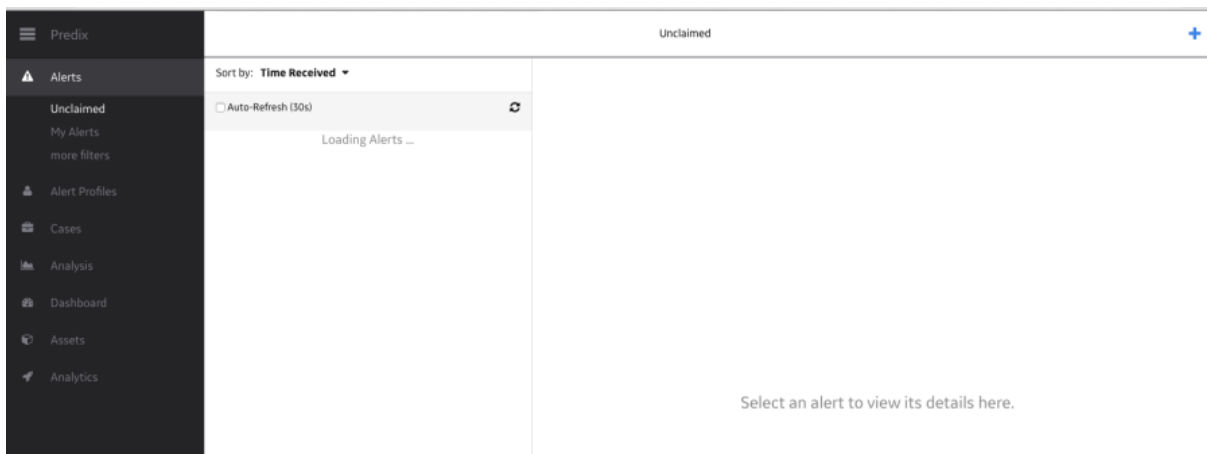
https://apm-tubs-hackapm-basic.run.aws-usw02-pr.ice.predix.io/<tenant_id>



Predix

Sign in

You will see all APM services



You have completed APM setup !!!

Instructions to setup Intel Edison Board

Step 1: Set up a new Intel Edison board

https://www.predix.io/resources/tutorials/tutorial-details.html?tutorial_id=1739&tag=1752&journey=Setup%20Intel%20Edison%20for%20Predix&resources=1785,1739,1742,1743

Step 2: Setup sensors

A0 – Temperature sensors
A1 – Light sensor
A2 – Rotary
A3 - Button

Step 3: Move pfa-container to Edison board

Request Hackathon volunteer to provide you pfa-container-edison.tar file.

```
$ scp -r pfa-container-edison.tar root@<Edison IP>:/home/root
```

Step 4: Untar on Edison board

```
$ tar -xvf pfa-container-edison.tar
```

Step 5: Generate keytool

\$ keytool

If keytool not found, then create shortlink:

\$ ln -sf \$(dirname `readlink -f \$(which java)`)/keytool /usr/bin/keytool

Reference: [Check here](#)

Step 6: Download Postman collection & get environment variables

Navigate to tools directory

Get file service-apis.postman_collection.json and import into postman.

Click on Init-Environment and change

Web.app.url – only change the tenant name for your environment

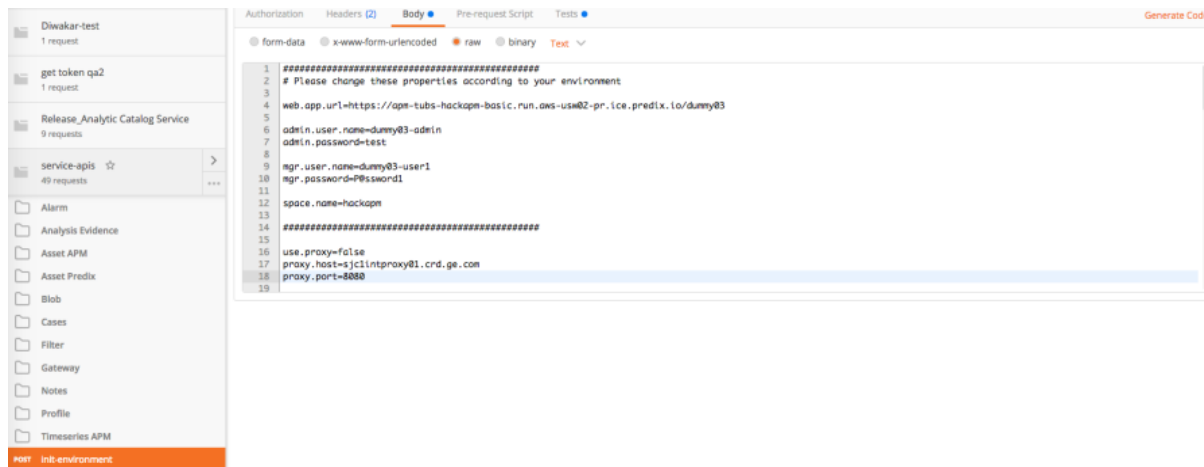
admin.user.name

admin.password

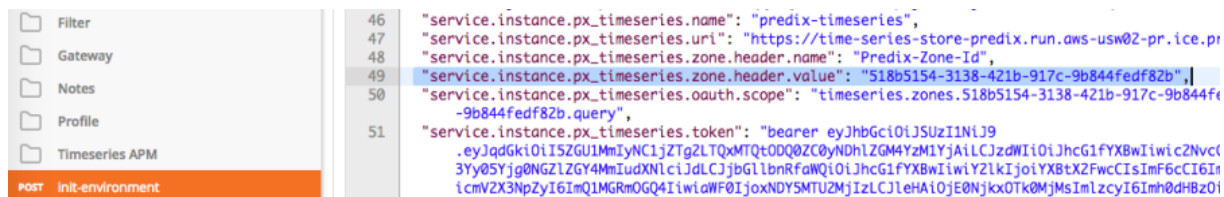
mgr.user.name

mgr.password

Hit Send.



Search for property and get the value for "service.instance.px_timeseries.zone.header.value"



Step 7: Change configs (WSS endpoint)

On Intel Board

\$ cd pfa-container-edison/configuration/machine

Configure Web Socket River

```
$ vi com.ge.dspmicro.websocketriver.send-0.config
```

Example:

[Required] The zone ID for the TimeSeries service instance

```
com.ge.dspmicro.websocketriver.send.header.zone.value="55dc40b2-18ef-4615-b55d-5ace39fd5250"
```

Replace the value with the value of service.instance.px_timeseries.zone.header.value from previous step

```
# [Required] The zone ID for the TimeSeries service instance
com.ge.dspmicro.websocketriver.send.header.zone.value="518b5154-3138-421b-917c-9b844fedf82b"
```

Step 8: Run Predix Machine

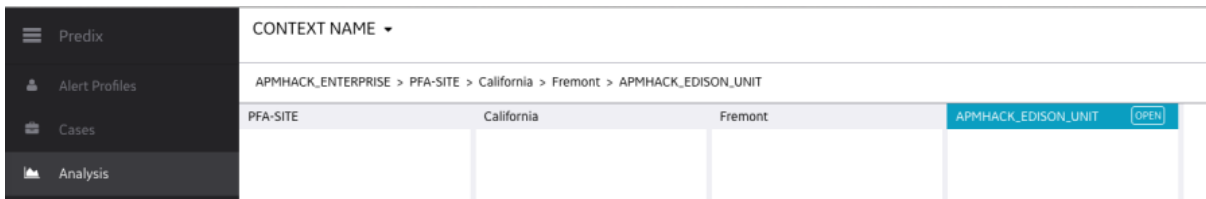
```
$ cd ~/pfa-container-edison/machine/bin/predix
```

```
$ ./predixmachine clean
```

With this step data is now streaming into Timeseries

Step 9: Visualize TS data

Login to APM using non-admin user that you created and click on "Analysis"
Select the asset and click "Open"



Search tags that are being stored in Timeseries

TemperatureTAE
LightTAE
RotaryAngleTAE
ButtonTAE





Step 10: Ingest Sample Asset model (Optional)

Ingest asset model from the UI by clicking on "Asset Ingestion" in the left hand pane of the UI. You can get a sample_assets.zip from https://github.com/apmdev/tools/tree/master/ingestion_data

The screenshot shows the 'Asset Ingestion' UI. The left sidebar menu includes: Predix, Assets, User Management, Setup, Asset Ingestion, and Audit Logs. The main area has a section 'Upload Data' with a 'SELECT FILE TO UPLOAD' button and an 'Upload' button. Below this, a message states: 'Make sure you upload a zip file containing only one JSON file.' A status legend shows: Completed (green), In Progress (yellow), Pending (orange), and Failure (red). A table lists ingested files:

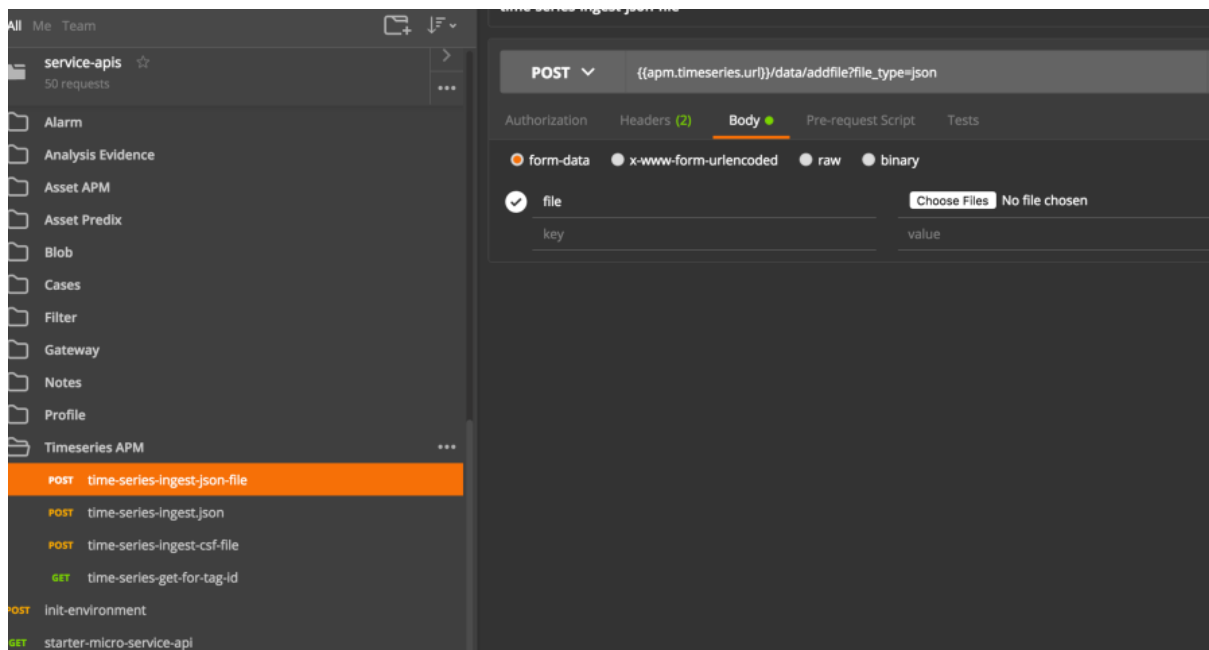
File Name	Start Time	End Time	File Size
upload.zip	07/20/2016 18:09	07/20/2016 18:09	923 bytes
upload.zip	07/20/2016 18:07	07/20/2016 18:09	923 bytes

At the bottom, there is a 'Show 10 entries' dropdown and a pagination control '1 - 2 of 2' with a '1' button.

Step 11: Ingest sample TS data (Optional)

To ingest timeseries files, it is assumed that you have downloaded postman collection by following the instructions described in the previous section "Download Postman collection & get environment variables". Use the APIs under "service-apis/Timeseries API" as shown in the picture below to ingest timeseries data files.

Sample timeseries data files are available at https://github.com/apmdev/tools/tree/master/ingestion_data

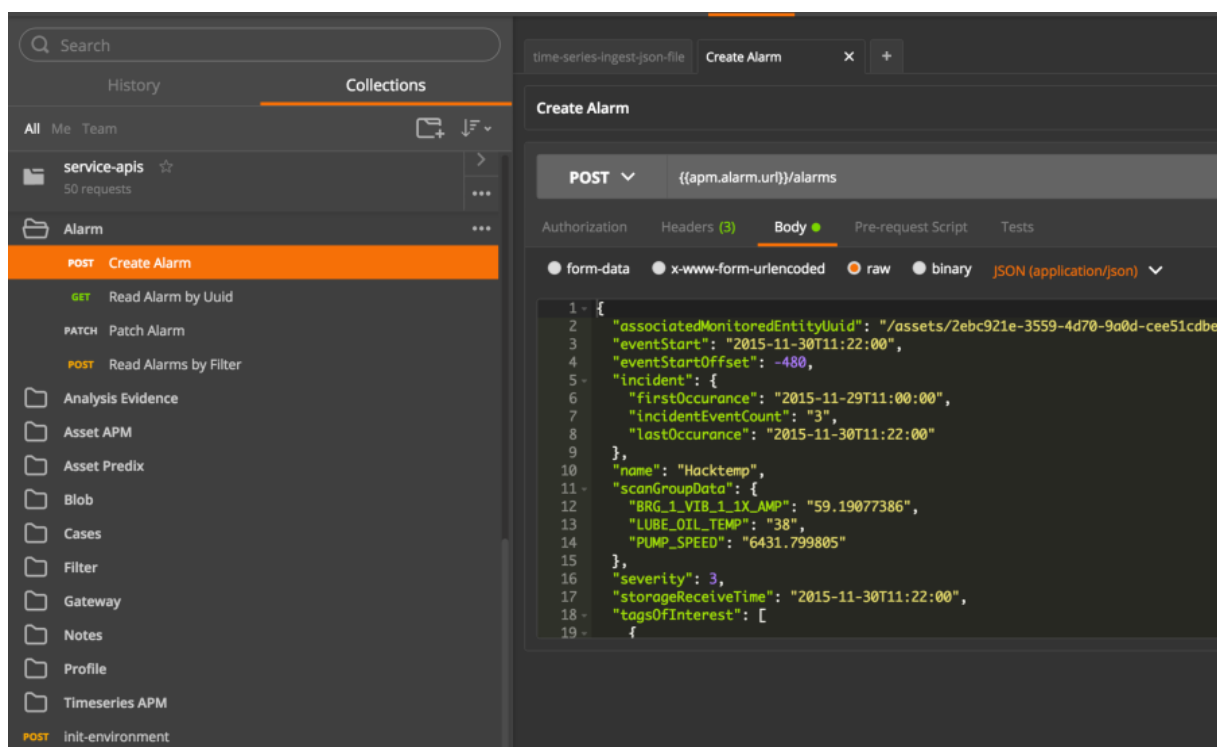


Step 12: Ingest Alerts & cases (Optional)

To ingest alerts files, it is assumed that you have downloaded postman collection by following the instructions described in the previous section "Download Postman collection & get environment variables".

Use the APIs under "service-apis/Alarm" and "service-apis/Cases" as shown in the picture below to ingest alarms and cases data files.

Sample alarms and cases data files are available at https://github.com/apmdev/tools/tree/master/ingestion_data



Step 13: Ingest sample Aviation flight data (Optional)

We have prepared an Aviation sample flight data set, which contains data for 20 flights and 300+ tags per flight. If you are interested, please follow the instructions at

https://github.com/apmdev/tools/tree/master/ingestion_data/sample-aviation-data

Start building an app

Step 1: Login to Cloud Foundry using your credentials

```
$ cf l -a https://api.system.aws-usw02-pr.ice.predix.io
```

Step 2: git clone sample micro-service

```
$ git clone https://github.com/apmdev/apm-ext-microservice
```

Follow readme on <https://github.com/apmdev/apm-ext-microservice>

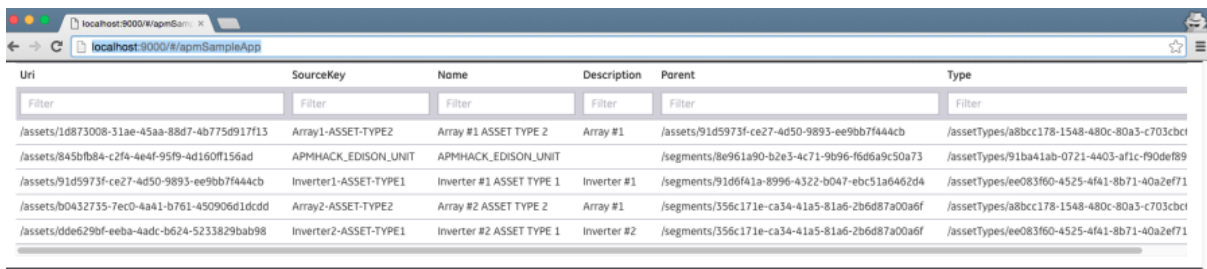
Step 3: git clone sample micro-app

```
$ git clone https://github.com/apmdev/apm-ext-microapp
```

d
Follow [readme](#)

Step 4: Visualize Assets

After this you should be able to see the following



Uri	SourceKey	Name	Description	Parent	Type
/assets/1d873008-31ae-45aa-88d7-4b775d917f13	Array1-ASSET-TYPE2	Array #1 ASSET TYPE 2	Array #1	/assets/91d5973f-ce27-4d50-9893-ee9bb7f444cb	/assetTypes/a8bcc178-1548-480c-80a3-c703cbci
/assets/845bfb84-c2f4-4e4f-95f9-4d160ff156ad	APMHACK_EDISON_UNIT	APMHACK_EDISON_UNIT		/segments/8e961a90-b2e3-4c71-9b96-f6d6a9c50a73	/assetTypes/91ba41ab-0721-4403-af1c-f90de89
/assets/91d5973f-ce27-4d50-9893-ee9bb7f444cb	Inverter1-ASSET-TYPE1	Inverter #1 ASSET TYPE 1	Inverter #1	/segments/91d6f41a-8996-4322-b047-ebc51a6462d4	/assetTypes/ee083f60-4525-4f41-8b71-40a2ef71
/assets/b0432735-7ec0-4a41-b761-450906d1dcd	Array2-ASSET-TYPE2	Array #2 ASSET TYPE 2	Array #1	/segments/356c171e-ca34-41a5-81a6-2b6d87a00a6f	/assetTypes/a8bcc178-1548-480c-80a3-c703cbci
/assets/dde629bf-eeba-4adc-b624-5233829bab98	Inverter2-ASSET-TYPE1	Inverter #2 ASSET TYPE 1	Inverter #2	/segments/356c171e-ca34-41a5-81a6-2b6d87a00a6f	/assetTypes/ee083f60-4525-4f41-8b71-40a2ef71

Step 5: Push microservice to cloud foundry

```
$ vi manifest.yml
```

Replace <spacename> with your space name

applications:

- name: apm-ext-microservice-<spacename>

```
$ cf push
```

Once the app is pushed then create a User Provided service using following command

```
$ cf cups apm-ext-microservice-<spacename> -p '{"uri": "https://apm-ext-microservice<space>.run.aws-usw02-pr.ice.predix.io"}'
```

Step 6: Push microapp to cloud foundry

```
$ vi app.js
```

Update assetPath variable to the name of the microservice

```
assetPath = "https://apm-ext-microservice-<spacename>.run.aws-usw02-pr.ice.predix.io/v1";
```

```
$ vi manifest.yml
```

Change the following:

applications:

- name: {tenant_name}-hackapm

services:

- apm-ext-microservice-<spacename>

```
$ cf push
```

Step 7: Verify microapp in APM

Login to APM as non-admin user and navigate to "Hackathon"

Uri	SourceKey	Name	Description	Parent	Type
/assets/18f64ab-c89c-4bb8-9ee5-0a24a5d03636	Array2-ASSET-TYPE2	Array #2 ASSET TYPE 2	Array #1	/segments/b580357b-6eb1-40a4-b8b0-aa889e29877	/assetType
/assets/19f2e1b3-b3e3-4cbb-8e46-bf7652f10354	APMHACK_EDISON_UNIT	APMHACK_EDISON_UNIT		/segments/4d3871ee-66cd-4904-8ae9-cc35be306b1a	/assetType
/assets/69c3137e-67aa-46b5-9aad-1cf61c0b5fdb	Inverter2-ASSET-TYPE1	Inverter #2 ASSET TYPE 1	Inverter #2	/segments/b580357b-6eb1-40a4-b8b0-aa889e29877	/assetType
/assets/7c3f3885-c221-4406-9125-ec7ab41202c	Inverter1-ASSET-TYPE1	Inverter #1 ASSET TYPE 1	Inverter #1	/segments/88c687bc-2de6-4350-941c-4da7e402954a	/assetType
/assets/9f825fc1-45df-40ad-965d-0b99f0b93ed	Array1-ASSET-TYPE2	Array #1 ASSET TYPE 2	Array #1	/assets/7c3f3885-c221-4406-9125-ec7ab41202c	/assetType

Success!!!

What just happened?

You were able to call APM asset service to visualize data from your own microservice & micro app.

Next steps

- Review sample micro service and microapp code.
- Create new applications and visualization using sample micro service & microapp
- Ingest your own data by creating instances of Predix data services