Hackathon 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

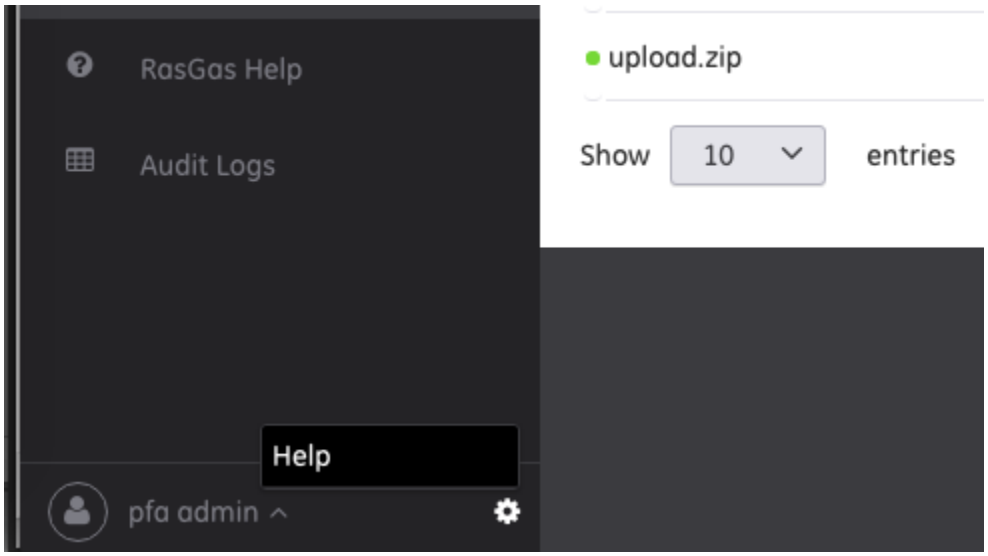
Step 1: Login to APM using your tenant URL

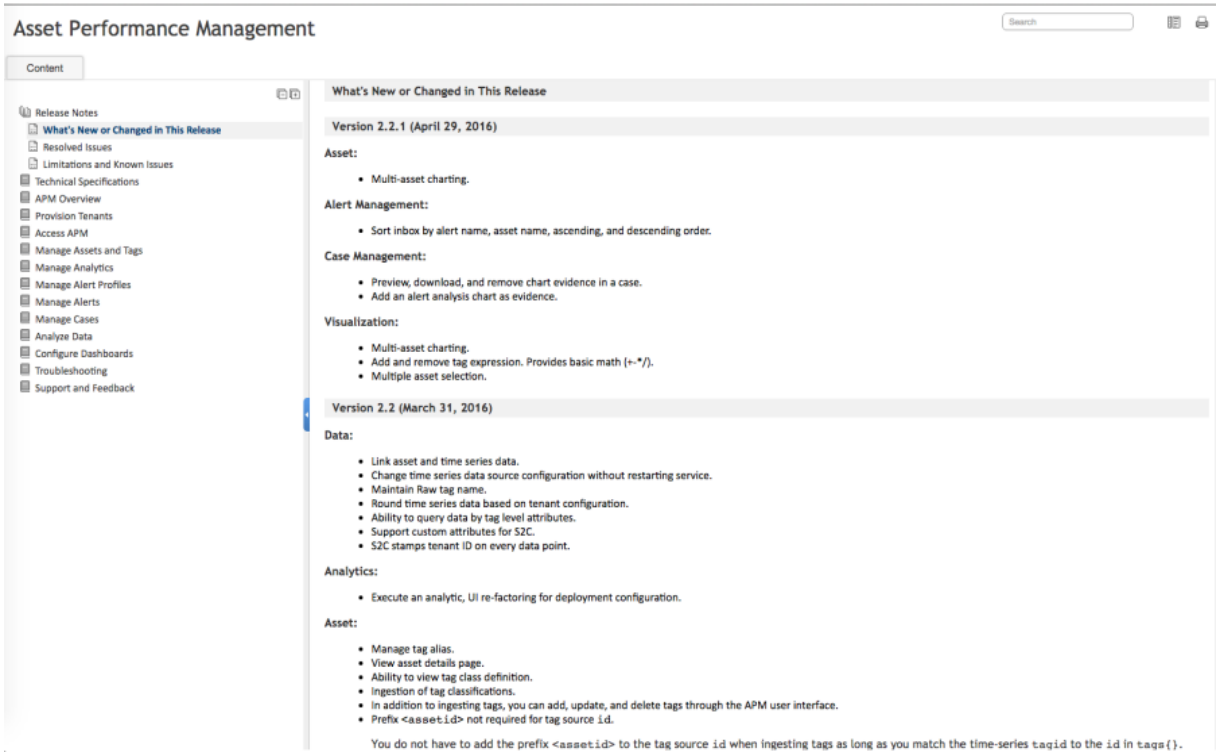
Tenant URL will be provided on a piece of paper
https://apm-tubs-hackapm-basic.run.aws-usw02-pr.ice.predix.io/<tenant_id>

Step 2: View API documentation

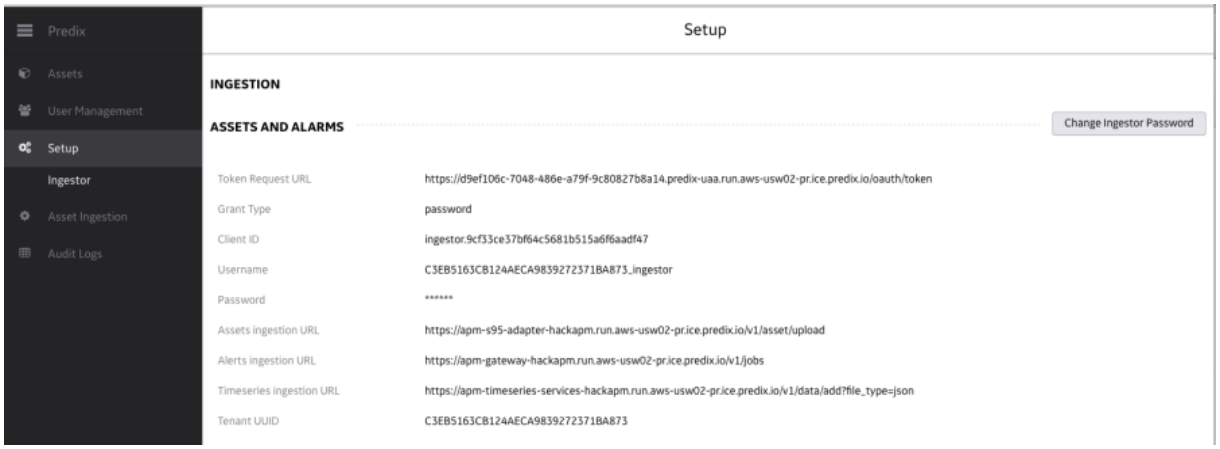
https://apm-apidocs-hackapm.run.aws-usw02-pr.ice.predix.io/

Step 3: View APM User documentation





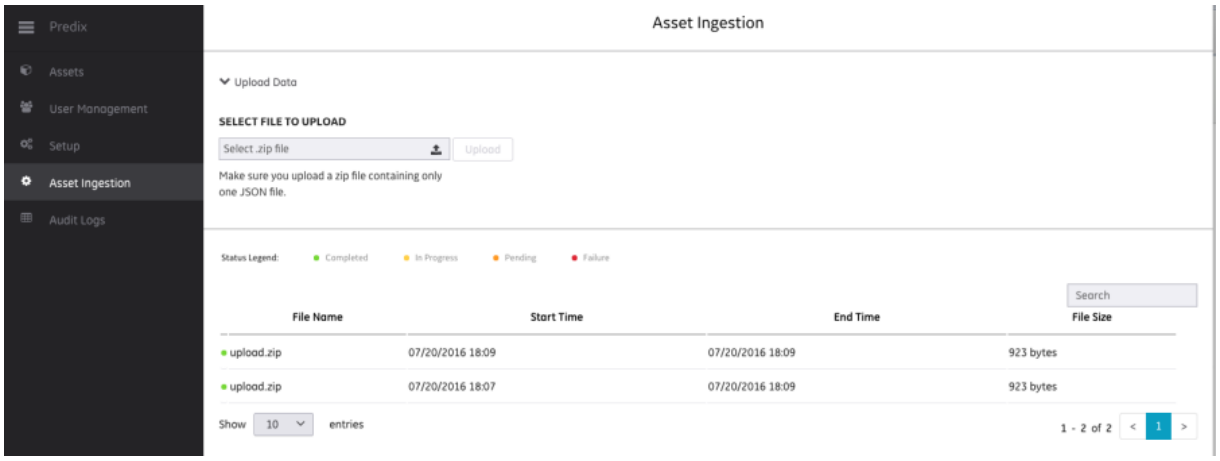
Step 4: You will be logged in as Tenant Admin.



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

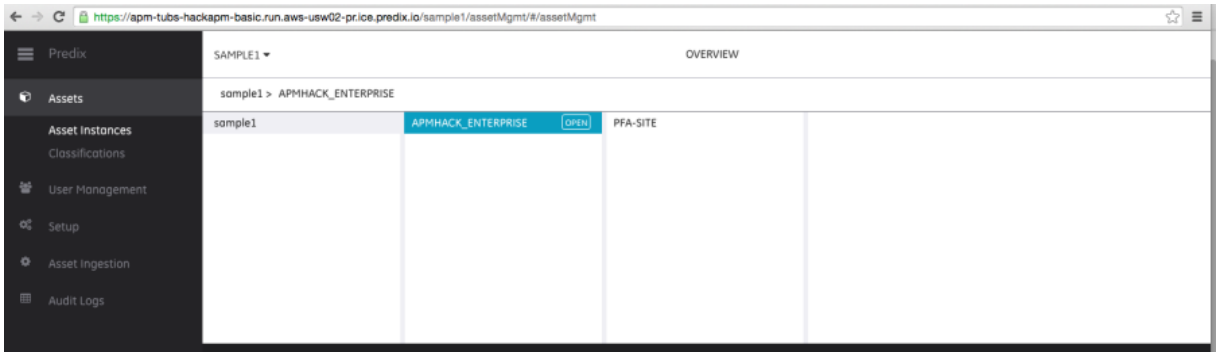
Go to tools/ingestion_data/intel_board_asset and look for pt_sample_assets.zip

Click on "Asset Ingestion"



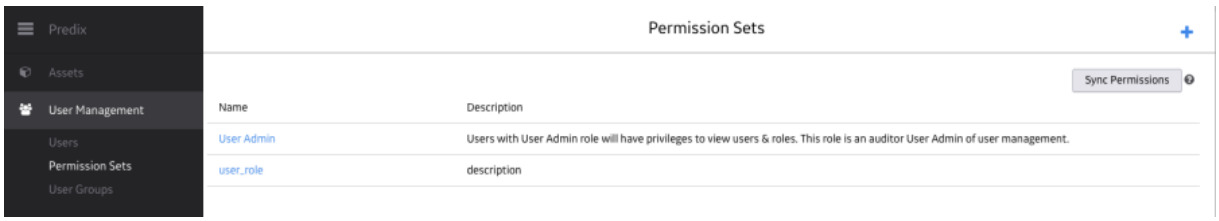
Verify asset ingestion is complete using ingestion logs.

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

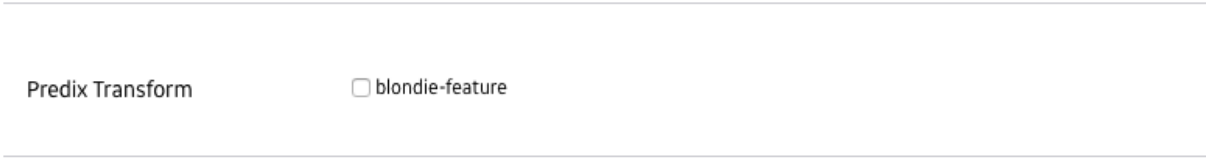


Step 6: Create permission set for non-admin user

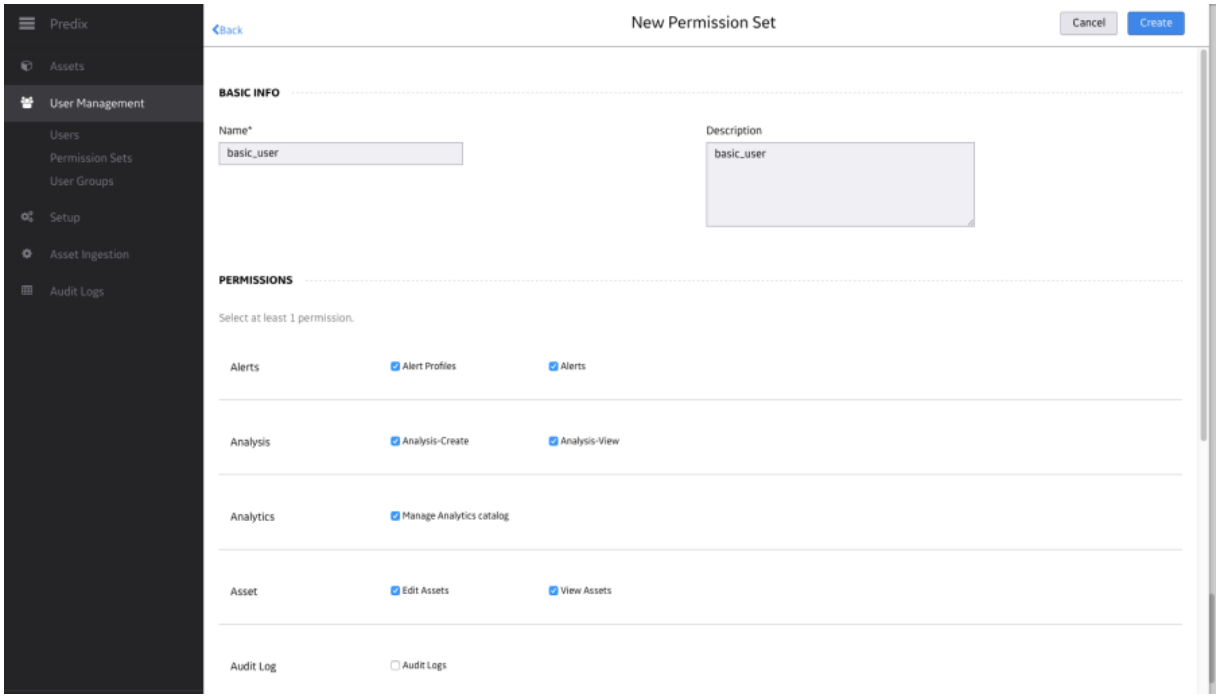
Navigate to "User Management" -> "Permission Sets" Click on +



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

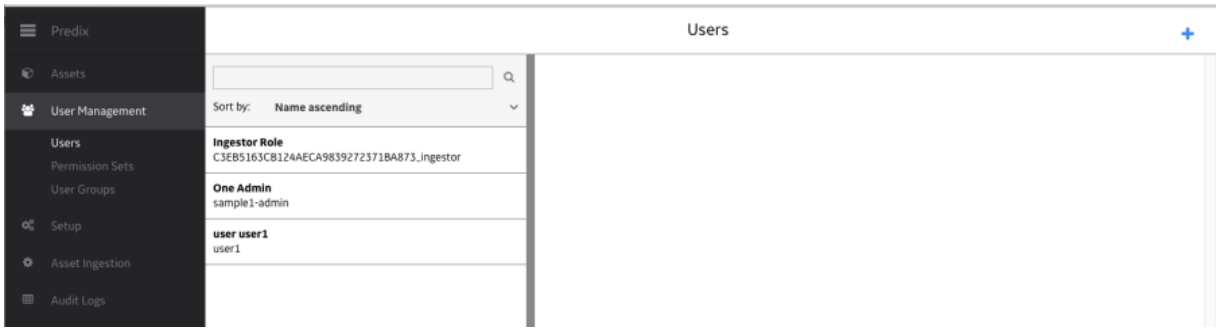


Pick services that you want and click "Create"



Step 7: Create non-admin user

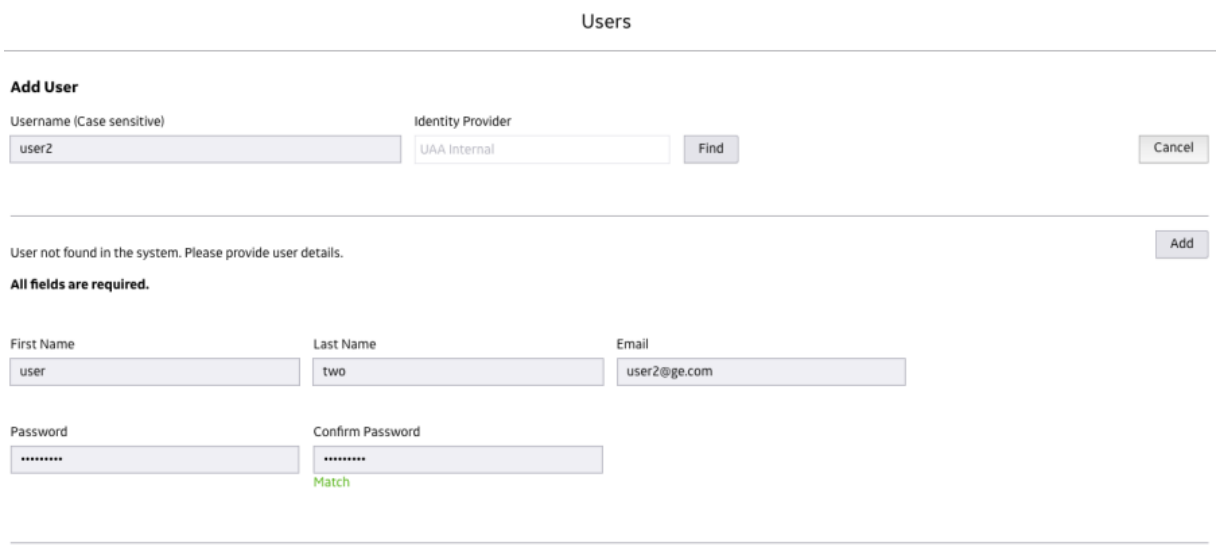
Navigate to "User Management" Click on +



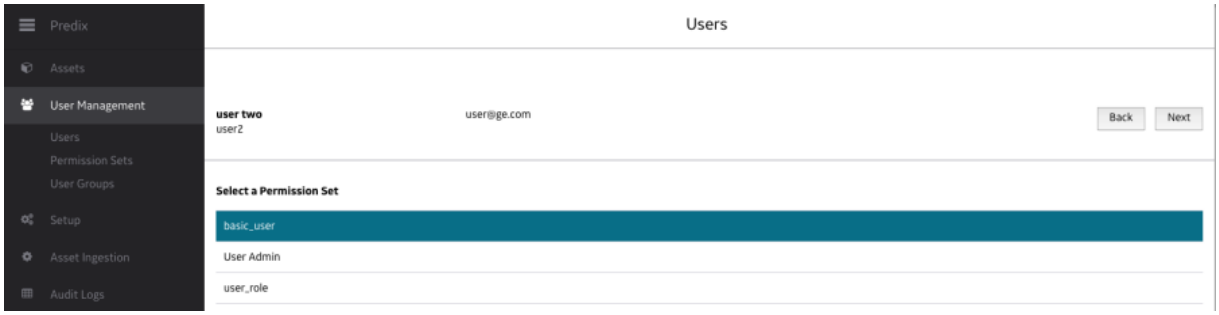
Navigate to "User Management" Click on +

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

Create a user by filling up the form.

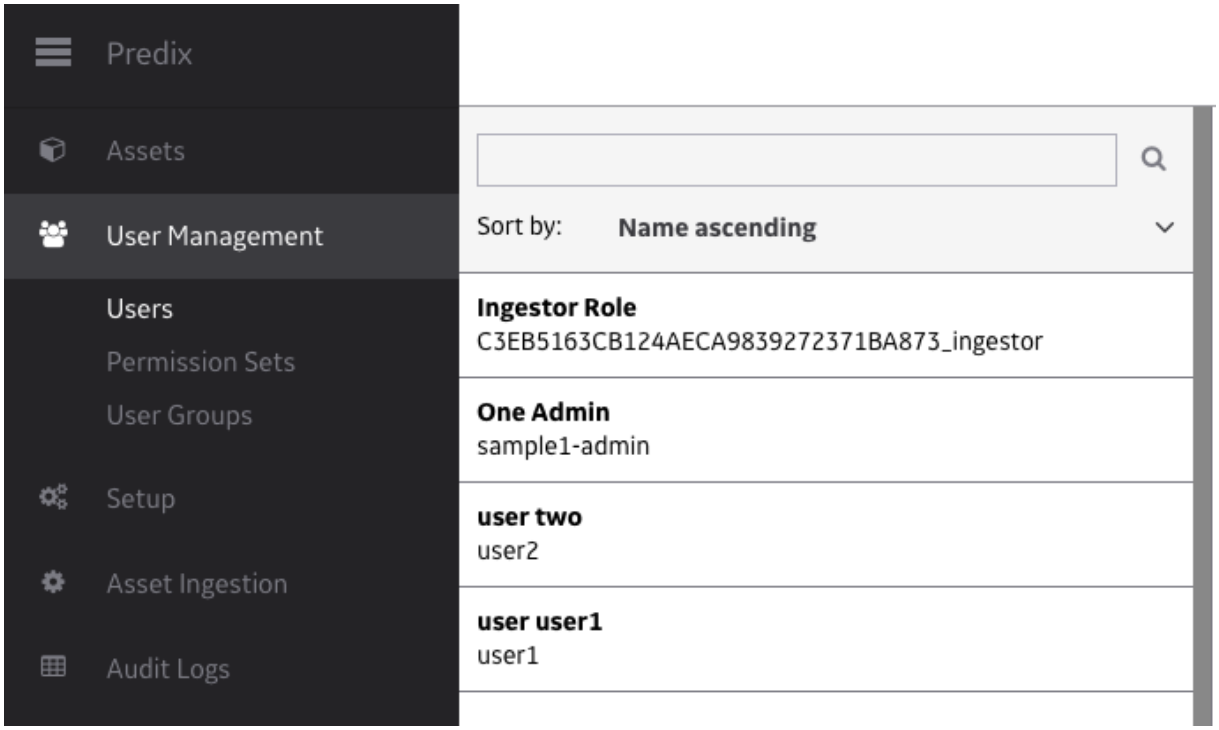


Select permission set that you just created and click "Next"



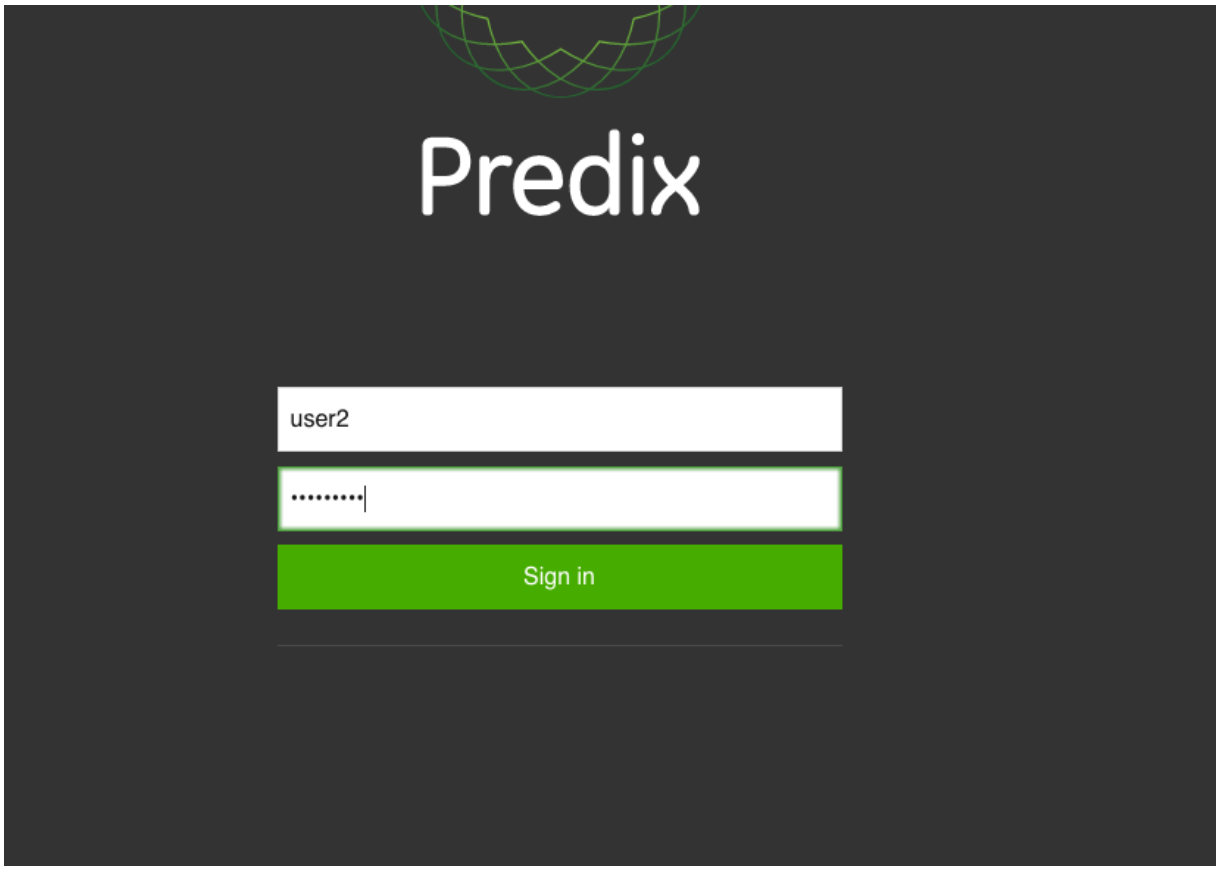
Select an Asset that this user can access. You can pick any number of assets at any level.

Click Finish.

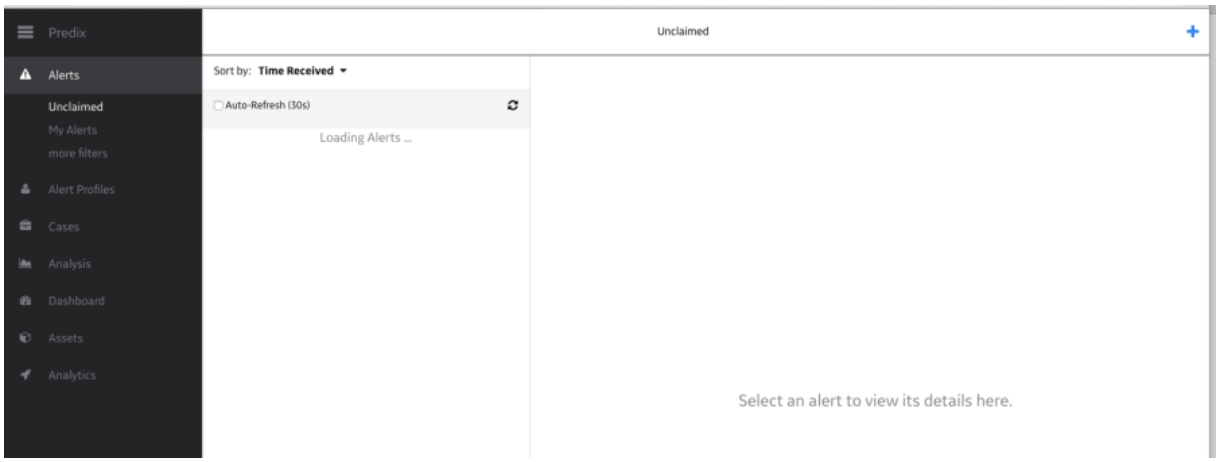


Step 8: Login to APM as non-admin user

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



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

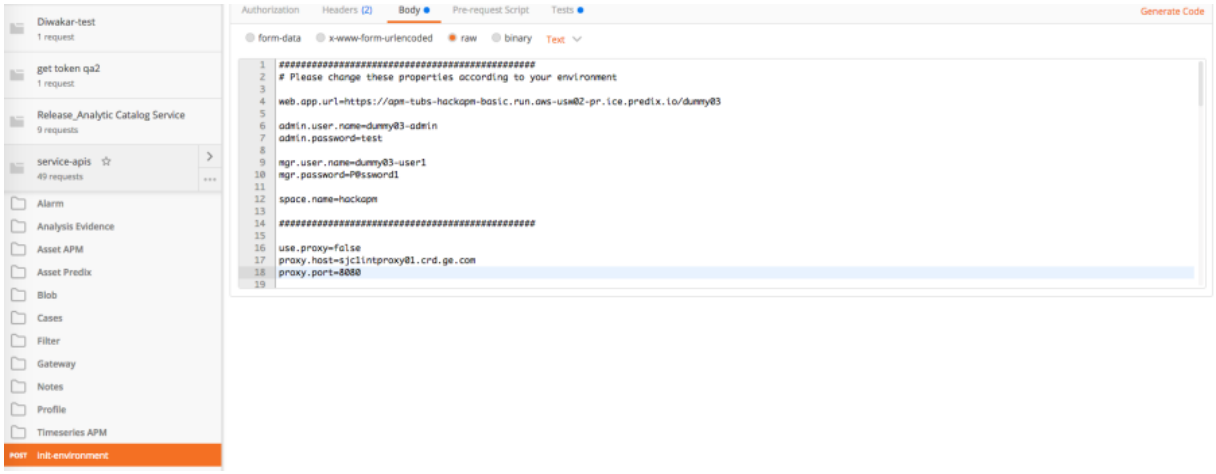
Get file service-apis.postman_collection.json and import into postman.

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

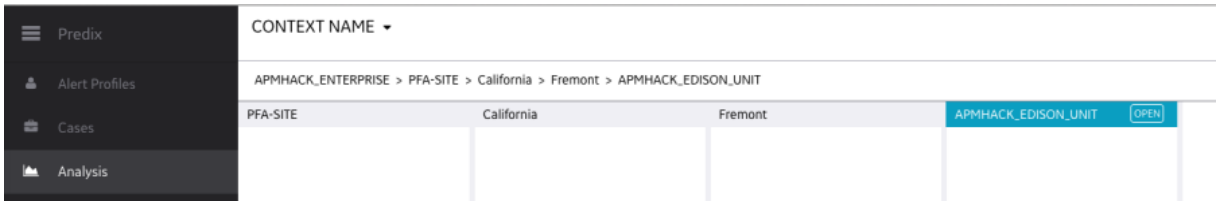
$ cd ~/pfa-container-edison/machine/bin/predix

$ ./predixmachine clean

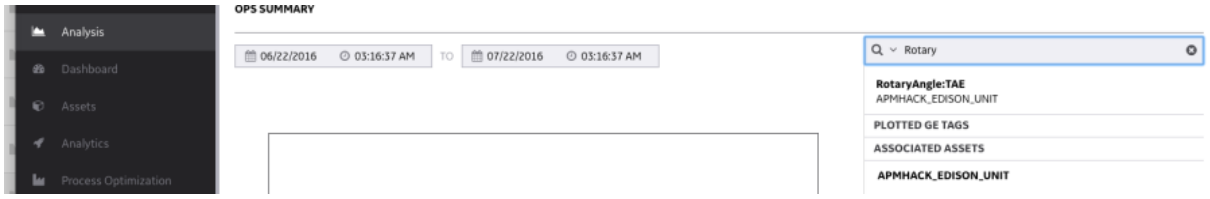
With this step data is now streaming into Timeseries

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"



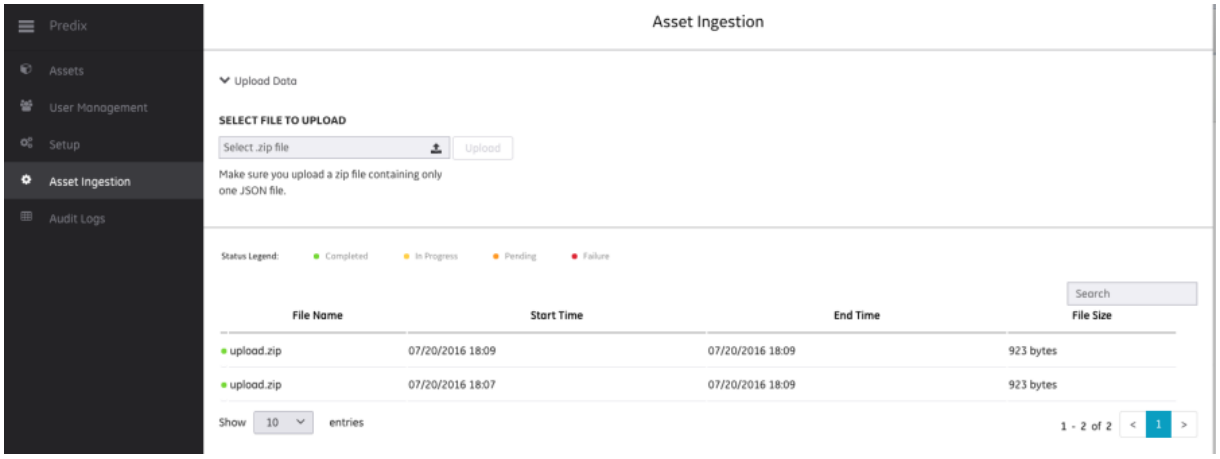
Search tags that are being stored in Timeseries
TemperatureTAE
LightTAE
RotaryAngleTAE
ButtonTAE



\

Step 10: Ingest Sample Asset model (Optional)

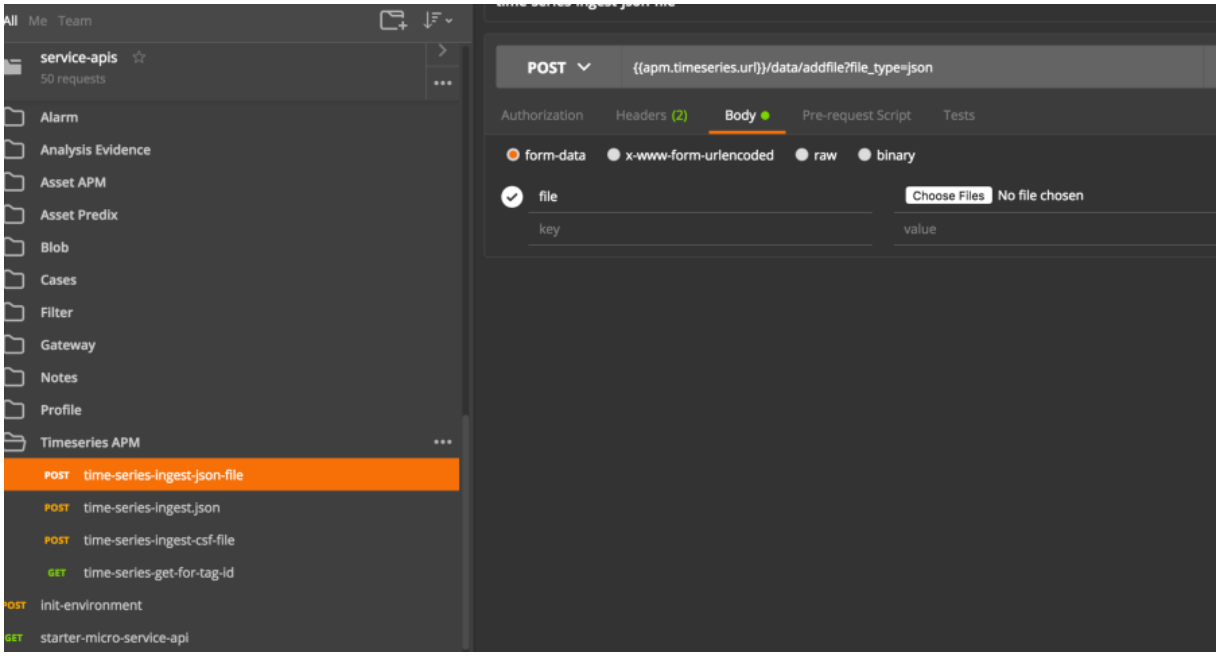
Ingest asset model from the UI by clicking on "Asset Ingestion" in the left hand pane of the UI.
You can get a sample_assets.zip from
https://github.com/apmdev/tools/tree/master/ingestion_data



Step 11: Ingest sample TS data (Optional)

To ingest timseries 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

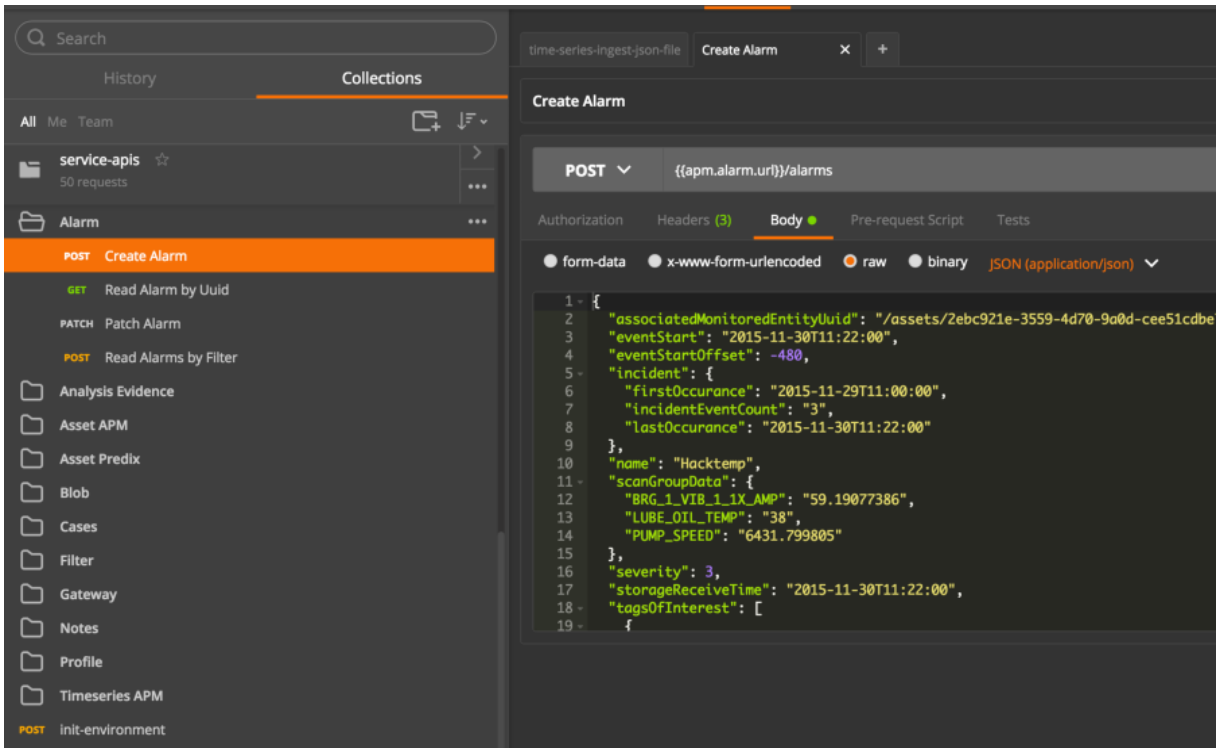


Step 12: Ingest Alerts & cases (Optional)

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

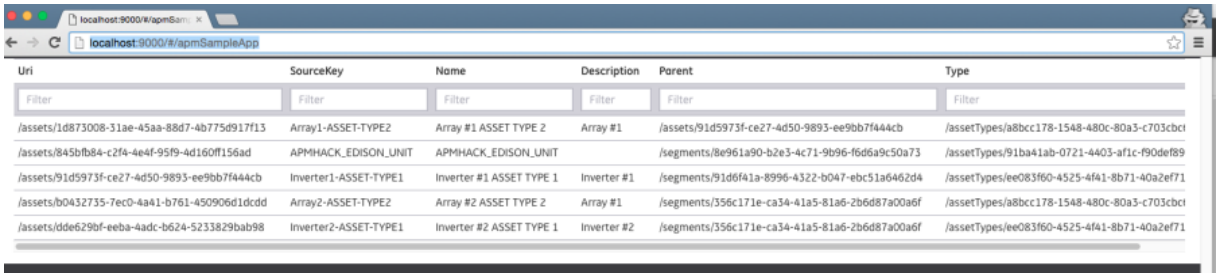
Step 3: git clone sample micro-app

$ 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



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

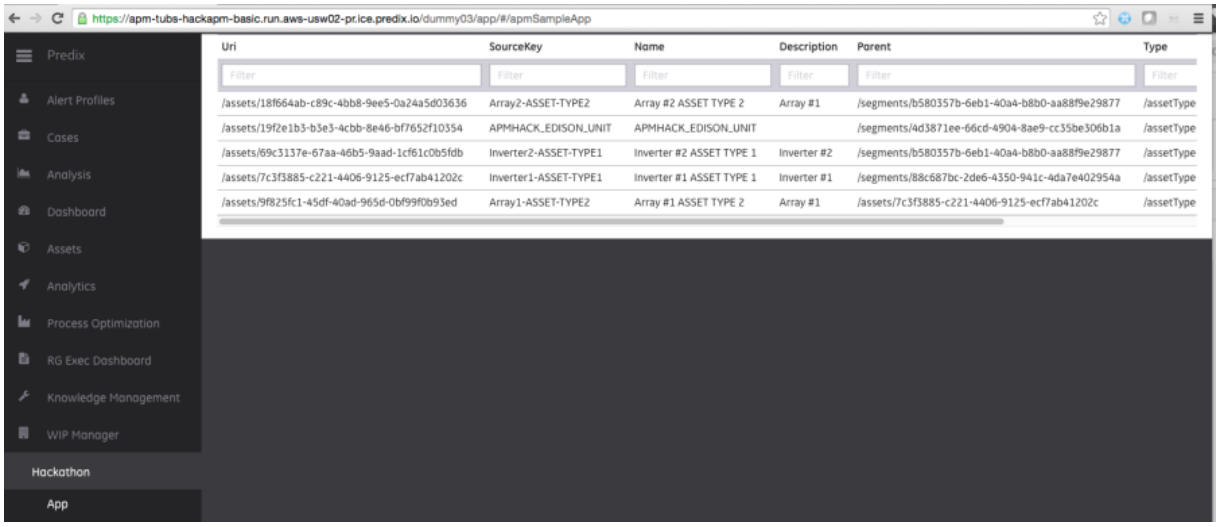
services:

- apm-ext-microservice**-<spacename>**

$ cf push

Step 7: Verify microapp in APM

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



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