# SAP Leonardo Machine Learning Foundation

## Prerequesites

Install Cloud Foundry (CLI)

Go to Cloud Foundry CLI download:

<https://github.com/cloudfoundry/cli#>[downloads](https://github.com/cloudfoundry/cli)

Download and run the installer specific for your platform.

Online Help :

<https://developers.sap.com/tutorials/cp-cf-download-cli.html>

## Prerequesites

Install Chrome Postman application

<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop>

Install Chrome Postman Interceptor application

1. Navigate to <https://chrome.google.com/webstore/search/postman%20interceptor> and select the Postman Interceptor tool

2. Click Add to Chrome

# SAP Cloud Cockpit

Login to CF CL

*cf login -a api.cf.us10.hana.ondemand.com -u* [*ml-train+us-112@sap.com*](mailto:ml-train+us-112@sap.com)

Listing all Services

Cf services

## Re-train Image Classification with sapml

*cf login -a api.cf.us10.hana.ondemand.com -u* [*ml-train+us-112@sap.com*](mailto:ml-train+us-112@sap.com)

*cf service-key ml\_instance\_2806 ml\_instance\_2806\_key*

Download the SAPML CF plugin from:

[https://tools.hana.ondemand.com/#mlfoundation](https://tools.hana.ondemand.com/)

*cf install-plugin -f <your Folder>\sapmlcli*

Setting/Getting sapml config

*cf sapml config get*

*cf sapml config set auth\_server* [*https://ml-train-us-112.authentication.us10.hana.ondemand.com*](https://ml-train-us-112.authentication.us10.hana.ondemand.com/)

*cf sapml config set job\_api* [*https://training.prod.us-east-1.aws.ml.hana.ondemand.com*](https://training.prod.us-east-1.aws.ml.hana.ondemand.com)

*cf sapml config set retraining\_image\_api* [*https://mlfproduction-retrain-image-api.cfapps.us10.hana.ondemand.com/api/v2/image/retraining*](https://mlfproduction-retrain-image-api.cfapps.us10.hana.ondemand.com/api/v2/image/retraining)

Init and list sapml fs (file system)

*cf sapml fs init*

*cf sapml fs list Brands/training/*

(Optional) Uploading images before retraining 1

* <https://help.sap.com/viewer/a86b12f81f424c45a0e7c83c1d5025c4/1904B/en-US/65df4d78cb6b485aab2c9fb11ae89ac1.html>

Install Minio

*brew install minio/stable/minio*

Install minio/mc

*brew install minio/stable/mc*

*Cf sapml fs config // to retrieve credentials and end point*

Recursively upload the local directory jewelry\_local to remote directory “jewelry” add your credentials once:

*./mc config host add saps3 https://<endpoint> <access\_key> <secret\_key>.*

You can then upload the file:

*./mc cp ~/.../jewelry\_local saps3/data/jewelry -- recursive*

Lancer Minio Webserver

https://<end point>

## Submit a retraining job

Edit : retrain.json

Launching the retraining

*cf sapml retraining job\_submit retrain.json –m image*

Checking the status :

*cf sapml retraining jobs –m image*

Check Job status with Swagger UI from Retraining URL

Retrieving the logs

List all the jobs :

*cf sapml fs list jobs/*

## Get a log file

*cf sapml fs get <job\_name>/retraining.log <local\_name.log>*

# Deploying the model

Model is stored in a repository and needs to be deployed

Listing all the models that can be deployed:

*cf sapml retraining models –m image*

Deploy the model

*cf sapml retraining model\_deploy <model\_name> <version> -m model*

*cf sapml retraining model\_deploy Brands1 1 –m image*

To get the list of all the models currently deployed

*cf sapml retraining model\_deployments –m image*

Listing all deployed models

*cf sapml modelserver list*

Getting info on a deployed model

*cf sapml modelserver get <model\_id>*

Undeploying a model

*cf sapml modelserver delete <model\_id>*

# Bring Your Own Model

Only TensorFlow is supported 1.3, 1.7, 1.8

Edit file ~.mc/config.json to add previously retrieved

* + - secretkey
    - accessKey

Launch Minio Webserver

<https://play.min.io:9000/minio/login>

# Re-train Image Classification with sapml

Enter the command:

*cf service-key <instance name> <service key>*

to display your service key and make sure that everything is available.

*cf sapml modelserver delete <model\_id>*

## BYOM

To access the model sample

[https://github.com/saphanaacademy/MLF](https://www.youtube.com/redirect?q=https%3A%2F%2Fgithub.com%2Fsaphanaacademy%2FMLF&redir_token=5D9DEOSe6vNyXg9877J2iInFnFt8MTU1ODQyMTk3OEAxNTU4MzM1NTc4&event=video_description&v=GZZ9Qn0uKTo) File: mnist.zip

To list the models in the repo:

*cf sapml model list*

Create a model :

*cf sapml model create minst -f minst.zip*

* will upload the model and create it on the server

We need to serve it (i.e. deploy it)

*cf sapml modelserver*

Pay attention to the fact that to deploy a new model you need to associate to a resource plan

List all the resource plans :

*cf sapml modelserver resourceplans*

Deploy a model in the resource plan cpu-small

*cf sapml modelserver create mnist -r cpu-small -u <version>*

Getting info on the deployed model

*cf sapml modelserver get <modelserverid>*

# Building an Inference Application

Application BYOM

App.py

Edit the manifest.yml

*---*

*applications:*

*- name: byom*

*host: ml-train-us112-byom*

*memory: 256M*

*disk\_quota: 1028M*

*timeout: 60*

*buildpack: python\_buildpack*

*services:*

*- ml\_instance\_2806*

*env:*

*MLF\_SERVICE: ml-foundation-trial-beta*

Application BYOM:

Create unique name for application :

host : <subaccount-name>-byom

Services : // can be find with: *cf s*

Env: MLF\_SERVICE: // can be find with command: *cf sapml config get*

After editing the manifest.yml type cf push

This will deploy the application in the cloud.

To check details about the application being deployed:

Cf app byom

Launch postman and do a POST with:

<https://ml-train-us112-byom.cfapps.us10.hana.ondemand.com/mnist>