

Reference Architecture ML Ops

Python 1 Use case – XGBoost

1. Create ML Ops tenant with AD
2. Assign AD user to the ML Ops tenant as a tenant admin
3. Log in as AD user
4. Go into ML Ops tenant and set up Project Repository connection to NFS
 - a. Note: this will automatically create a data, misc, docs, models, and code directory
5. Create the following directories:
 - a. /data/UCI_Income
 - i. Storage for raw, cleaned, and encoded data
 - b. /code/XGB
 - i. Storage for Jupyter notebook and scoring script
 - c. /models/XGB_Income
 - i. Storage for all the models
6. Upload the following files in the corresponding directories
 - a. /data/UCI_Income
 - i. adult_data.csv
 - ii. adult_test.csv
 - b. /code/XGB
 - i. XGB_Income.ipynb
 - ii. XGB_Scoring.py
7. Create a training cluster named “pythonmldl” using the Python ML and DL Toolkit image

The screenshot shows the 'Create Training Cluster' interface in the BlueData ML Ops console. The left sidebar contains navigation links: Dashboard, Users, Project Repository, Source Control, Model Registry, Training, Deployments, Data Sources, Flavors, App Store, and Notebooks. The main panel is titled 'Create Training Cluster' and contains several sections:

- Cluster Detail:** Includes 'Name*' (pythonmldl), 'Description', and 'RunTime Image*' (circled in red, set to 'Python ML and DL Toolkit').
- Node Roles:** Includes 'NotebookServer' and 'TrainingEngine', both set to 'Small - 4 VCPU, 8.00 GB RAM' with a count of 1.
- Add-Ons:** Includes 'Endpoint Wrapper' (checked), 'RESTServer', and 'LoadBalancer', all set to 'Small - 4 VCPU, 8.00 GB RAM' with a count of 1.
- Upload Scaling Policy:** A field to 'Select a valid JSON file' with a 'Browse' button.
- Advanced Settings:** A section at the bottom with a 'Submit' button.

- a.
8. Create Jupyter notebook and attach the notebook to the “pythonmldl” training cluster
 - a. The training cluster must be in “ready” state before you can attach the notebook to the training cluster

BlueData ML Ops AIML / qa1

Dashboard
Users 1
Project Repository
Source Control
Model Registry
Training 3
Deployments 2
Data Sources 2
Flavors
App Store
Notebooks 2

Create Notebook Cluster

Cluster Detail
Name* Jupyter
Description
RunTime Image* Jupyter Notebook with JupyterHub

Node Roles
NotebookServer Small - 4 VCPU, 8.00 GB RAM 1

Associate with Training Environments pythonmldl

Advanced Settings

Submit

- b.
- Open Jupyter notebook and log in with AD credentials
 - Obtain XGB_Income.ipynb file (2 methods)
 - Method 1: Directly upload file once logged into Jupyter
 - Method 2: Open terminal in Jupyter and copy from project repo file path
 - Open XGB_Income.ipynb file and run each cell individually
 - Notebook contains detailed comments and explanations
 - Generates cleaned csv files, encoded file, and model files
 - Navigate back to EPIC and go into Model Registry to register the newly created model
 - Click on register new model and fill in information to match screenshot below

BlueData ML Ops AIML / qa1

Dashboard
Users 1
Project Repository
Source Control
Model Registry
Training 2
Deployments 2
Data Sources 2
Flavors
App Store
Notebooks 2

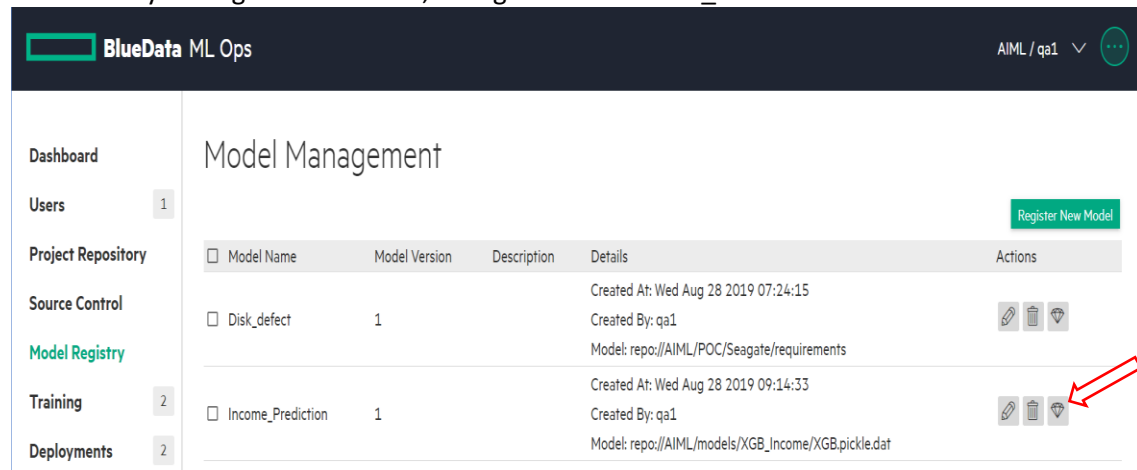
Register/Update Model

Label
Name* Income_Prediction
Description
Model Version* 1
Path to Model Repo* repo://AIML/models/XGB_Income/XGB.pickle.dat Browse
Path to Scoring Script* repo://AIML/code/XGB/XGB_Scoring.py Browse
Trained on Environment

Submit

b.

- c. Click submit
13. Deploy the newly registered model (2 methods)
 - a. Method 1: Click the diamond icon located on the far right side of the model registry page, which shows your registered models, to register the Income_Prediction model



- i.
- b. Method 2: Navigate to Deployments on the left side nav bar and create a new deployment
- c. Give name as "Income_Prediction" and select the model to deploy. (The model will be preselected if you went through method 1)
- d. Use "Python ML/DL Toolkit" as the run time image

The screenshot shows the 'Create Deployment Cluster' interface in BlueData ML Ops. On the left is a navigation sidebar with options: Dashboard, Users (1), Project Repository, Source Control, Model Registry, Training (2), Deployments (2), Data Sources (2), Flavors, App Store, and Notebooks (2). The main area displays a form for creating a new deployment cluster:

Cluster Detail

- Name*: Income_Prediction
- Description*:
- Select Model*: Income_Prediction, 1
- RunTime Image*: Python ML/DL Toolkit

Node Roles

- InferenceEngine: Small - 4 VCPU, 8.00 GB RAM, 1 instance

Add-Ons

- Endpoint Wrapper: [checked]
- RESTServer: Small - 4 VCPU, 8.00 GB RAM, 1 instance
- LoadBalancer: Small - 4 VCPU, 8.00 GB RAM, 1 instance

Upload Scaling Policy: Select a valid JSON file [Browse]

Advanced Settings

[Submit]

- e.
- f. Click submit

14. Generate prediction request using Postman

- Download Postman here: <https://www.getpostman.com/products>
- Once the deployment cluster is in “ready” state, click into the cluster
- Copy the link and auth token given in the LoadBalancer role to Postman
 - Clicking the auth token text will automatically copy the auth token

BlueData ML Ops AIML / qa1

Income_Prediction ● ready
[AIML/Deployment]

Node(s) Info | ActionScript(s) | ServiceStatus | Cluster Histories

Public Endpoints | Actions

Name	Distribution	Role	Instance IP	Services
bluedata-23.bdlocal	Endpoint Wrapper	RETSer	172.18.0.28	API Server : http://mip-bd-vm16.mip.storage.hpecorp.net:10041 [Auth Token] SSH : mip-bd-vm16.mip.storage.hpecorp.net -p 10042
bluedata-22.bdlocal	Python ML/DL Toolkit	InferenceEngine	172.18.0.26	sshd : mip-bd-vm16.mip.storage.hpecorp.net -p 10040
bluedata-24.bdlocal	Endpoint Wrapper	LoadBalancer	172.18.0.27	Model serving request balancer stats API Server : http://mip-bd-vm16.mip.storage.hpecorp.net:10044 [Auth Token] Model Serving LoadBalancer : http://mip-bd-vm16.mip.storage.hpecorp.net:10045/<<model_name>>/<<model_version>>/predict [Auth Token]

- Example of Postman
 - Substitute <<model_name>> with “Income_Prediction”
 - Substitute <<model_version>> with “1”
 - Fill in the following in the “Headers” tab to match screenshot below

Postman File Edit View Help

New Import Runner My Works

POST http://gateway.terry.lab.blued... POST http://mip-bd-vm16.mip.stora...

Untitled Request

POST http://mip-bd-vm16.mip.storage.hpecorp.net:10045/Income_Prediction/1/predict

Params Authorization Headers (10) Body Pre-request Script Tests

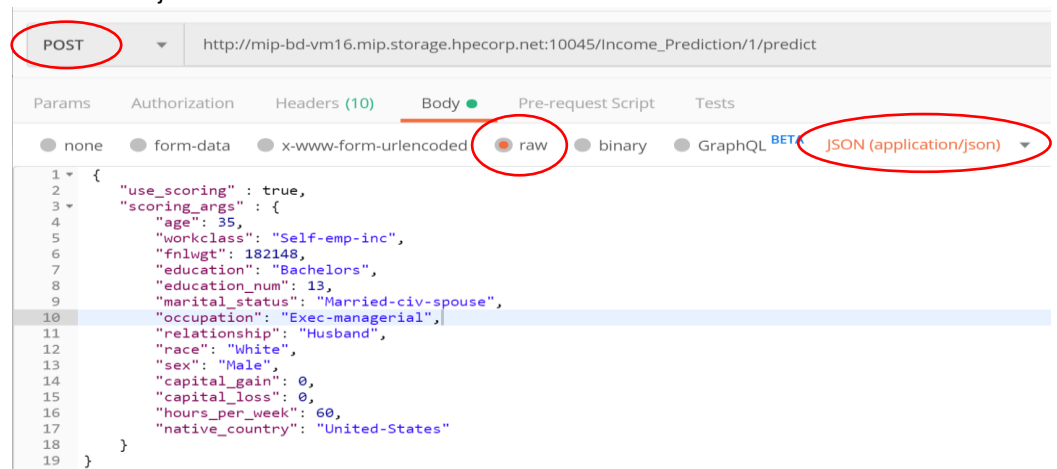
▼ Headers (2)

KEY	VALUE
X-Auth-Token	VBCakf7eC5
Content-Type	application/json
Key	Value

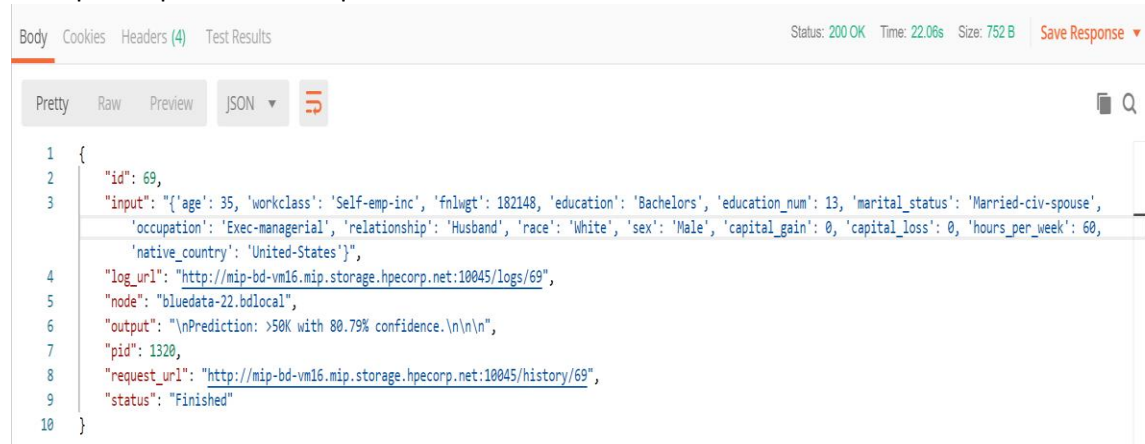
- Fill in the following in the “Body” tab
 - Make sure the request type is POST
 - Body text should be “raw” and application/json

3. Copy text below into Postman

```
a. {
  "use_scoring" : true,
  "scoring_args" : {
    "age": 35,
    "workclass": "Self-emp-inc",
    "fnlwgt": 182148,
    "education": "Bachelors",
    "education_num": 13,
    "marital_status": "Married-civ-spouse",
    "occupation": "Exec-managerial",
    "relationship": "Husband",
    "race": "White",
    "sex": "Male",
    "capital_gain": 0,
    "capital_loss": 0,
    "hours_per_week": 60,
    "native_country": "United-States"
  }
}
```



v. Example output of POST request at the bottom of Postman window



Images Used

1. Python ML and DL Toolkit
 - Image Version: 1.2
 - Distro ID: bluedata/scikittraining_gpu
 - Category: AIML/Training
 - Used for: creating training cluster
2. Jupyter Notebook with JupyterHub
 - Image Version: 1.0
 - Distro ID: bluedata/notebook
 - Category: AIML/Notebook
 - Used for: creating notebook server
3. Python ML/DL Toolkit
 - Image Version: 1.0
 - Distro ID: bluedata/scikittraining
 - Category: AIML/Deployment
 - Used for: deploying model
4. Endpoint Wrapper
 - Image Version: 1.4
 - Distro ID: bluedata/flask_serving
 - Categories: AIML/Deployment, AIML/Training
 - Used for: creating REST endpoint for prediction