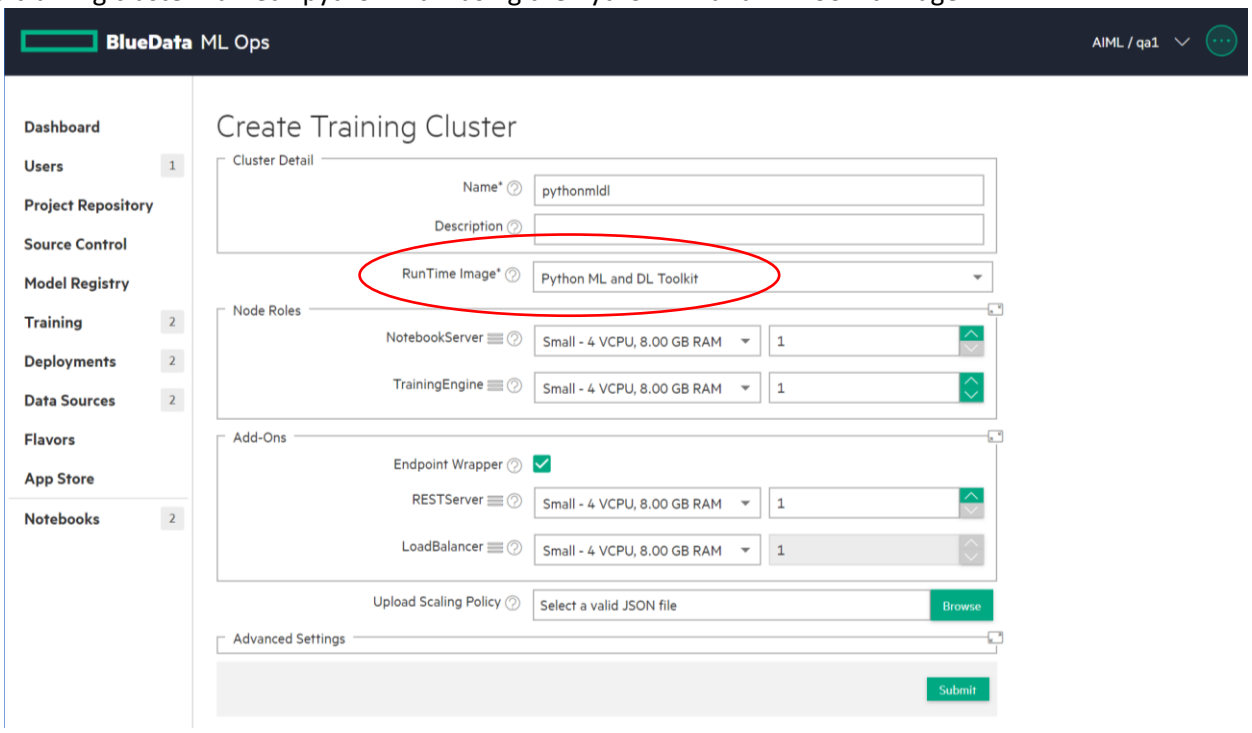
Reference Architecture ML Ops

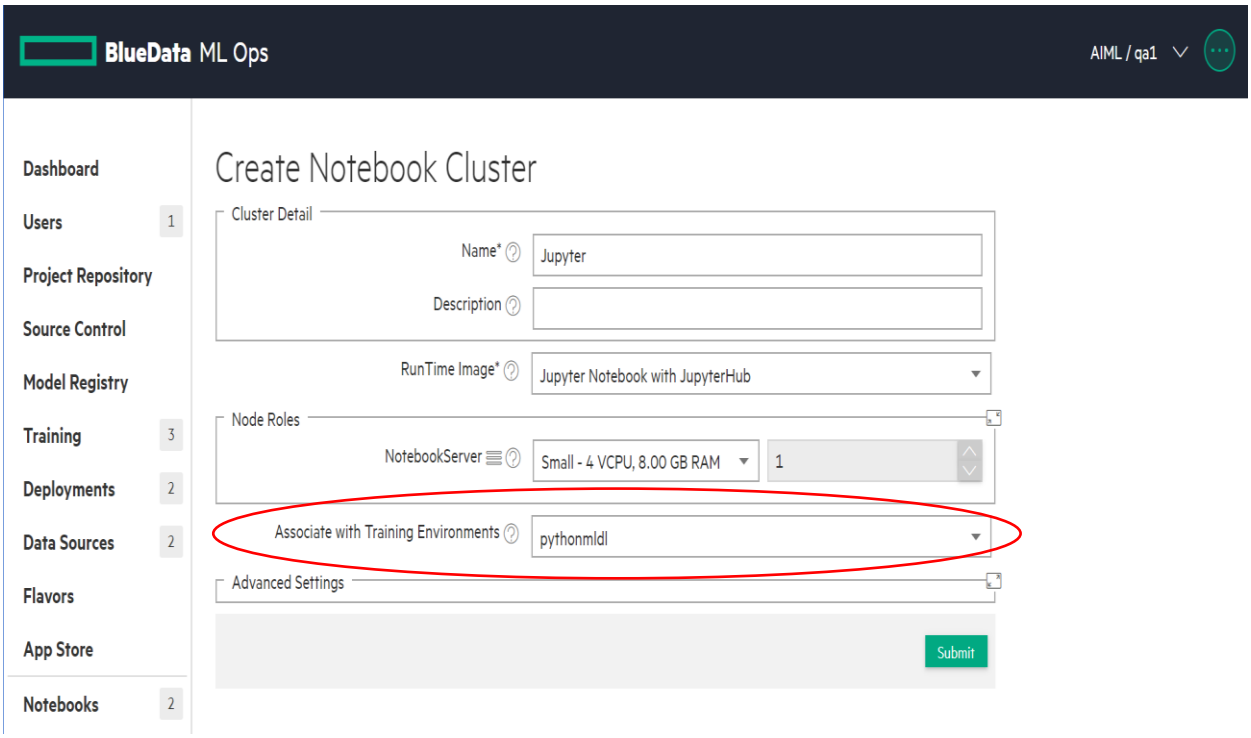
Python 1 Use case – XGBoost

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
    a.



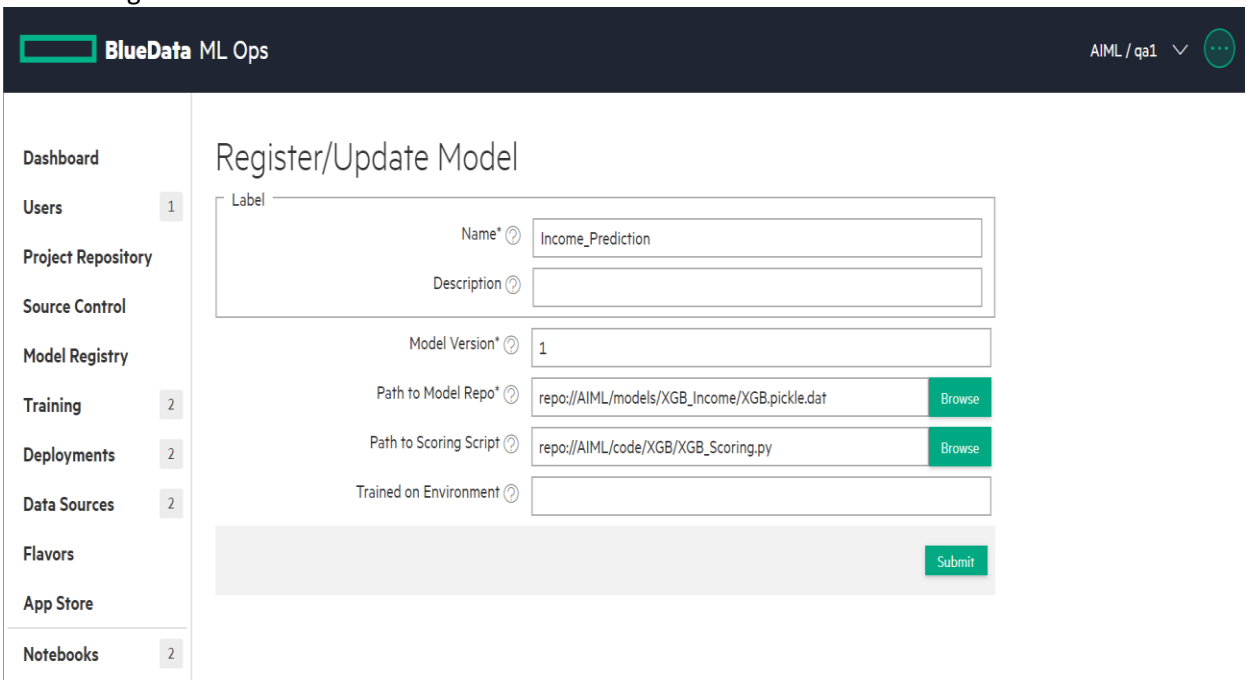
8. Create Jupyter notebook and attach the notebook to the “pythonmldl” training cluster
    a. The training cluster must be in “ready” state before you can attach the notebook to the training cluster

b.



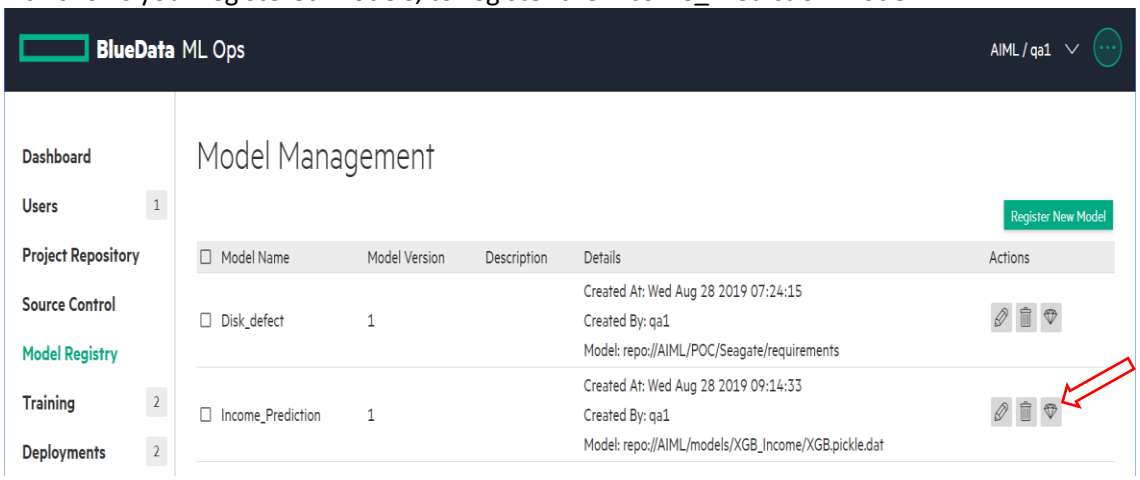
9. Open Jupyter notebook and log in with AD credentials
10. Obtain XGB_Income.ipynb file (2 methods)
    a. Method 1: Directly upload file once logged into Jupyter
    b. Method 2: Open terminal in Jupyter and copy from project repo file path
11. Open XGB_Income.ipynb file and run each cell individually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

b.

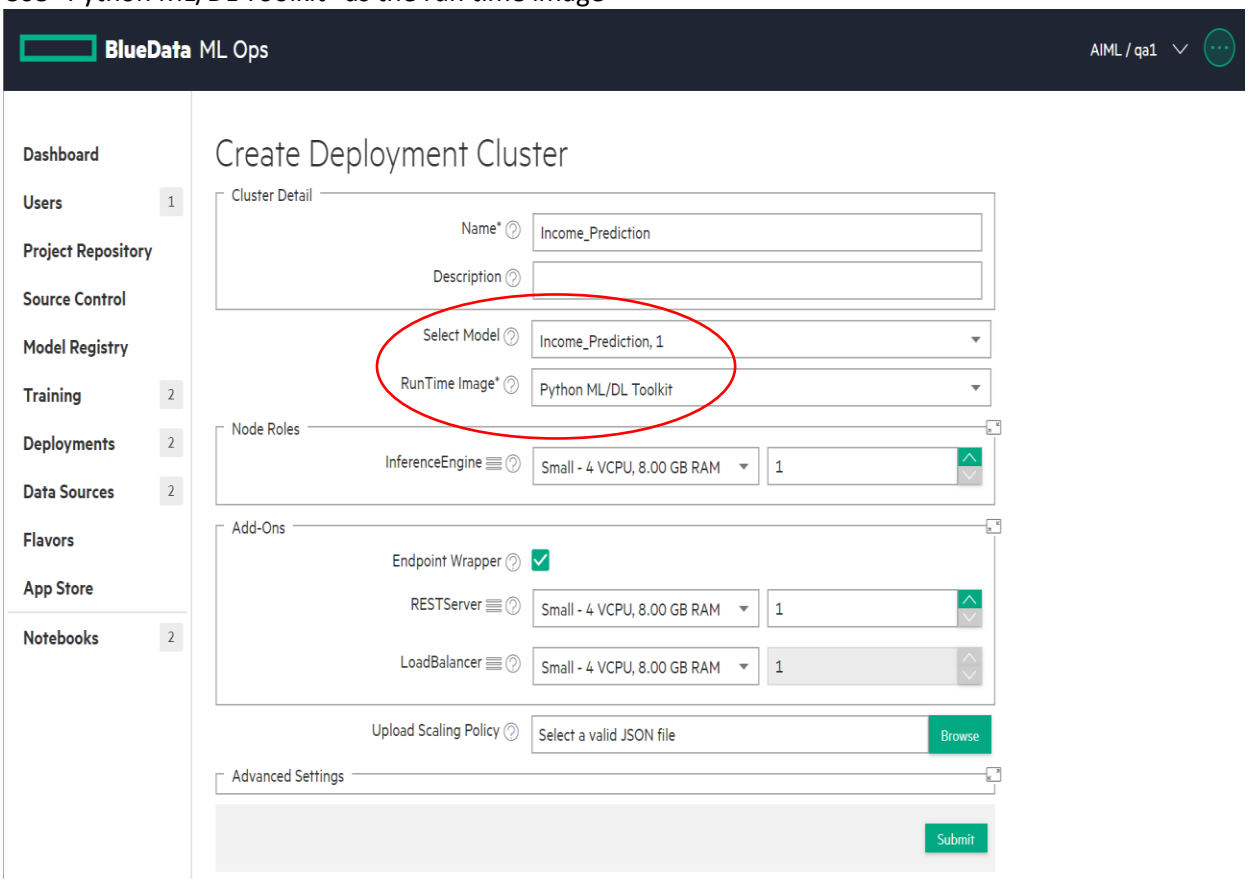


c. Click submit

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.

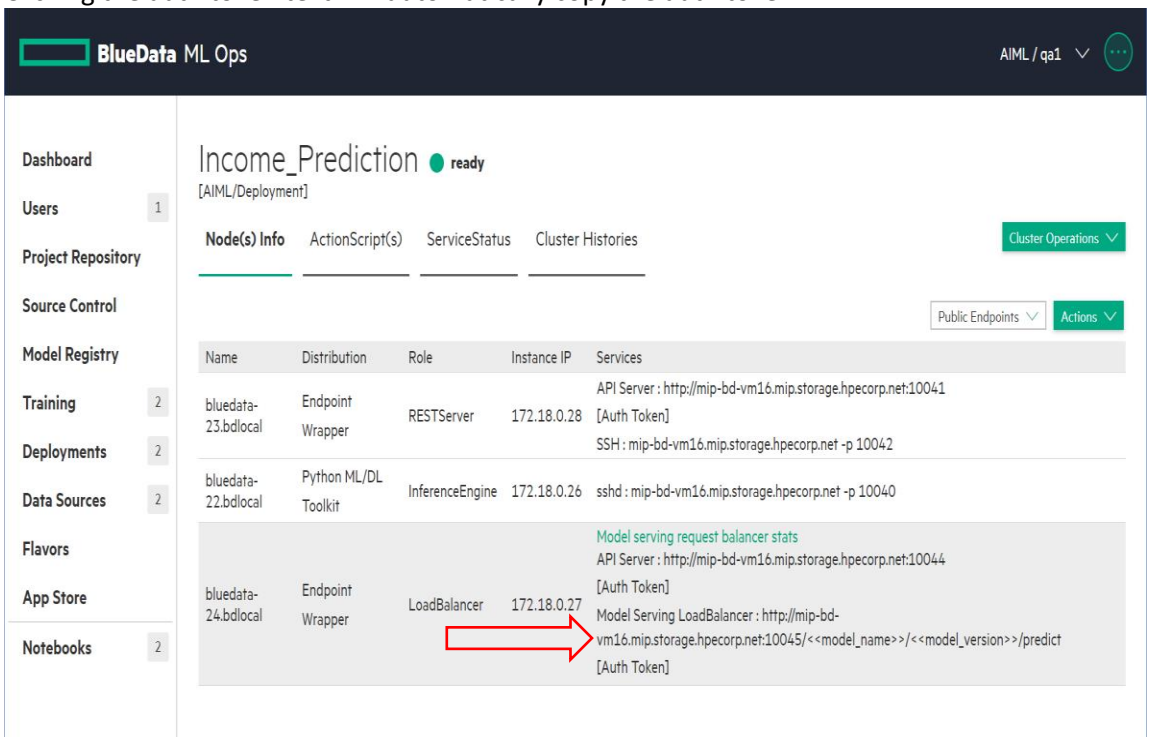


    b. Method 2: Navigate to Deployments on the left side nav bar and create a new deployment
    c. Give name as “Income_Prediction” and select the model to deploy. (The model will be preselected if you went through method 1)
    d. Use “Python ML/DL Toolkit” as the run time image

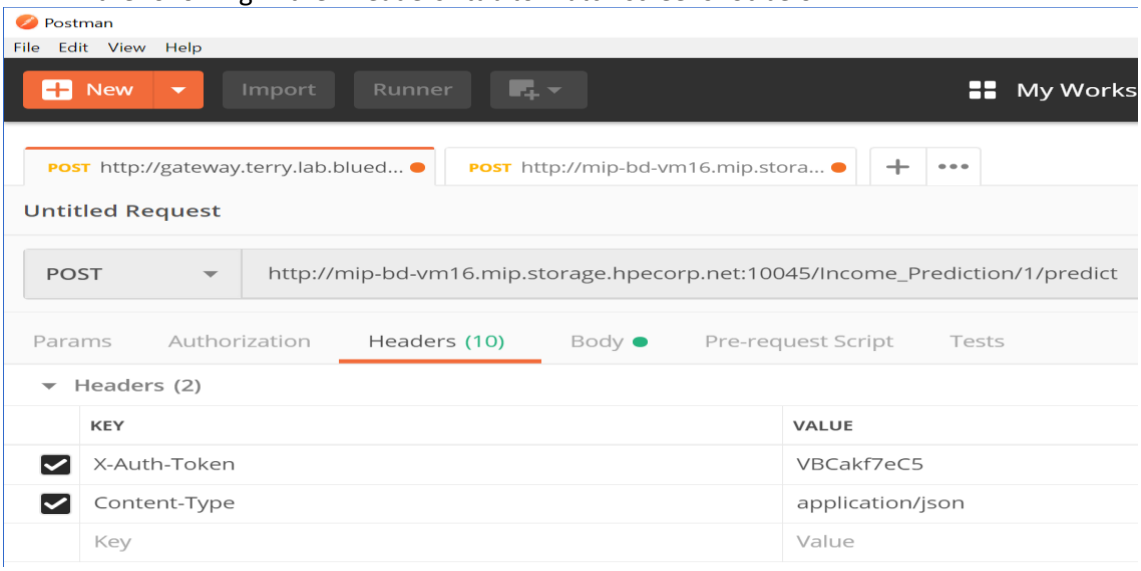


    e.
    f. Click submit

14. Generate prediction request using Postman
    a. Download Postman here: https://www.getpostman.com/products
    b. Once the deployment cluster is in "ready" state, click into the cluster
    c. Copy the link and auth token given in the LoadBalancer role to Postman
        i. Clicking the auth token text will automatically copy the auth token
        ii.

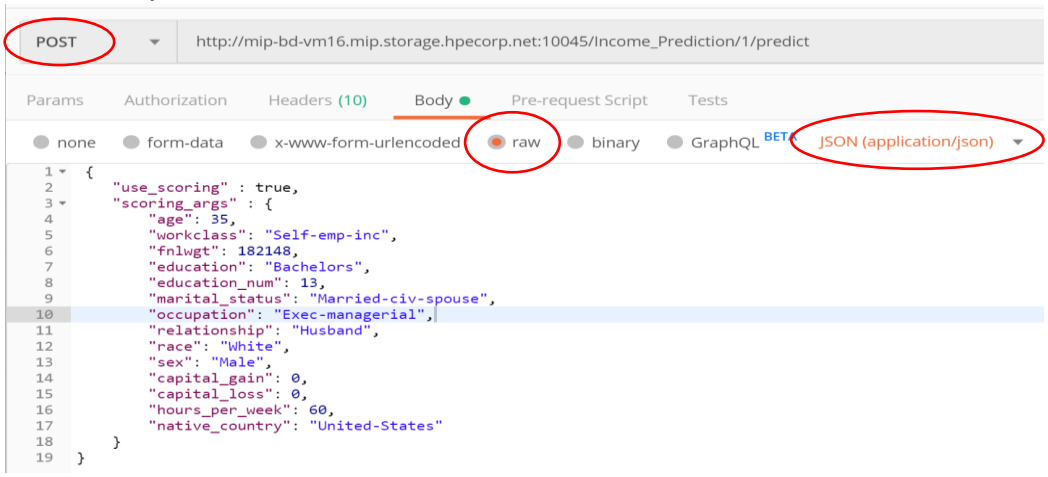


    d. Example of Postman
        i. Substitute <<model_name>> with "Income_Prediction"
        ii. Substitute <<model_version>> with "1"
        iii. Fill in the following in the "Headers" tab to match screenshot below

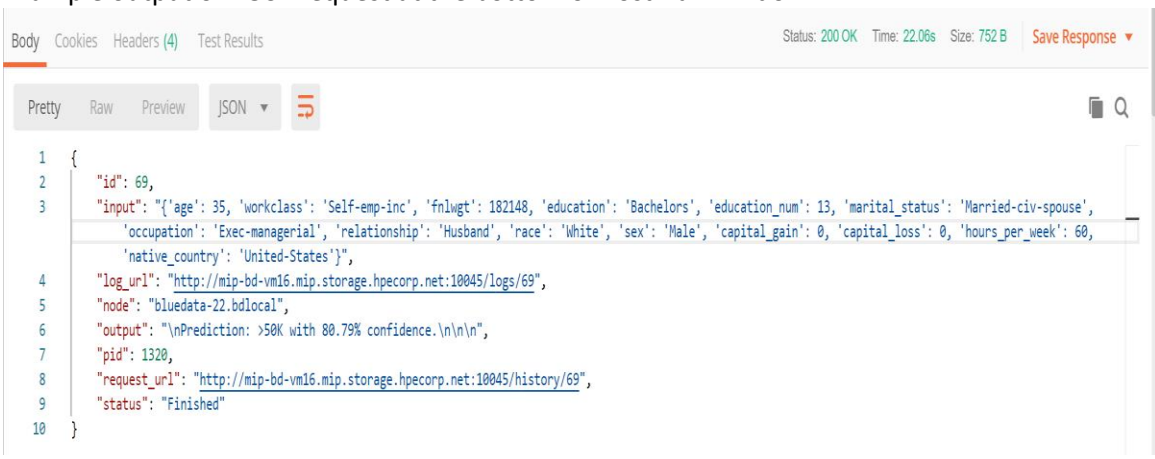


        iv. Fill in the following in the "Body" tab
            1. Make sure the request type is POST
            2. Body text should be "raw" and application/json

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