

# Watson Machine Learning + DevOps

## Introduction

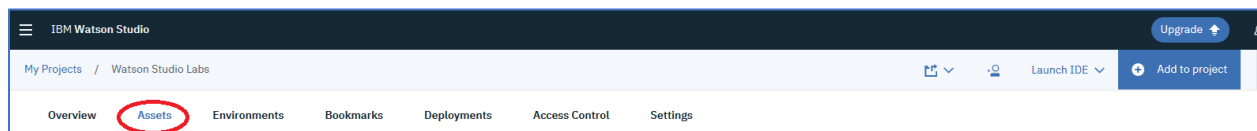
This lab introduces the Watson Machine Learning (WML) capability using the Female Human Trafficking datasets. It also covers the DevOps toolchain in the IBM Cloud. The lab consists of the following steps:


- Adding a data asset to the Watson Studio Labs project
- Creating a model to predict the trafficking risk for a traveler using the Model Builder interface. Saving the model to the model repository.
- Deploying and testing the model using the WML UI.
- Deploying a simple web front-end application and connecting it to the deployed model using an IBM Cloud toolchain.

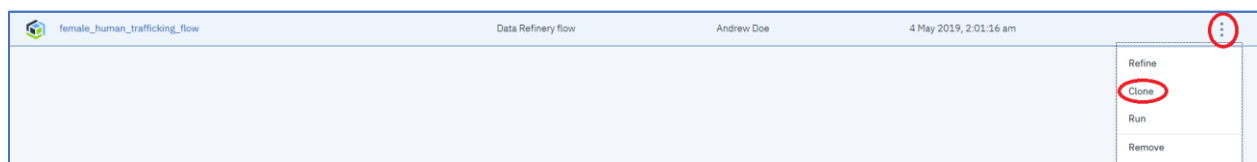
## Adding a Data Asset to the project


Our input into the model creation consists of the VETTING\_LEVEL, AGE, Category, COUNTRIES\_VISITED\_COUNT, and the PASSPORT\_COUNTRY fields. The VETTING\_LEVEL is the target field. The others are input fields. We will modify the Data Refinery flow created in a previous lab to create this data asset and add it to the project.

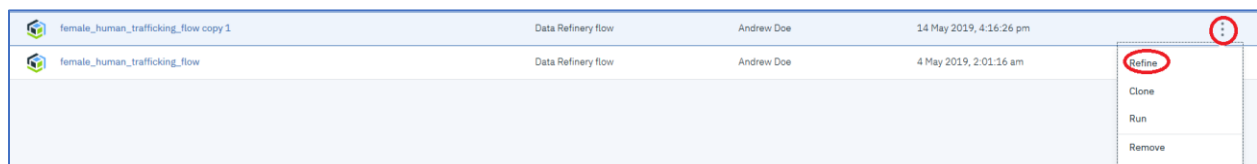
1. Click on the **Assets** tab in the Project.



2. Click on the vertical ellipse  on the right side of the **female\_human\_trafficking\_flow** and click on **Clone**.



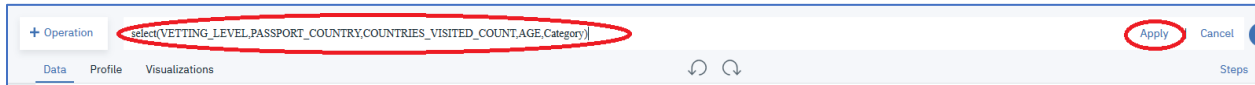
3. Click on the vertical ellipse  on the right side of the **female\_human\_trafficking\_flow copy 1** and click on **Refine**.



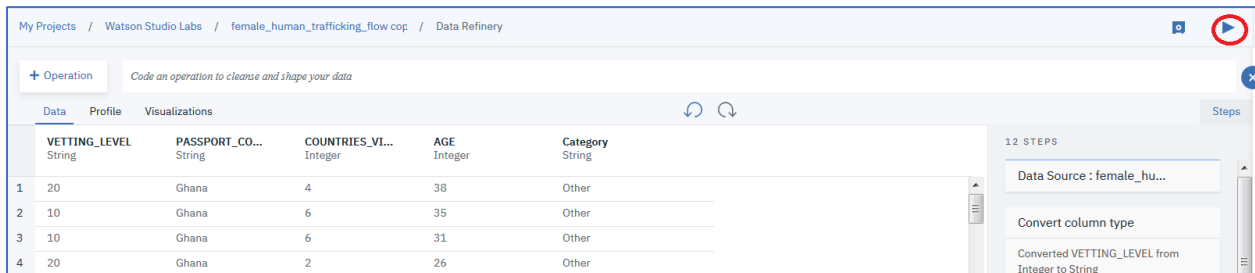
4. In the Operation entry field type the following:

```
select(VETTING_LEVEL,  
PASSPORT_COUNTRY,COUNTRIES_VISITED_COUNT,AGE,Category)
```

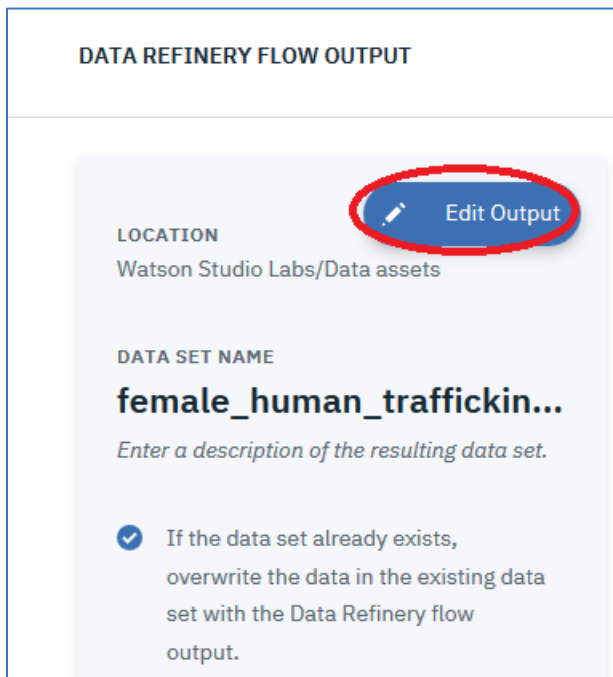
then click **Apply**.






5. Click on the **Run** icon.



6. Click on the **pencil icon** to modify the **DATA SET NAME**.



7. Change the DATA SET NAME to **female\_human\_trafficking\_output** and click on the check icon 

**Edit output**  


**LOCATION \***  
Watson Studio Labs/Data assets  
[Change Location](#)

**DATA SET NAME \***  
female\_human\_trafficking\_output 69


**DESCRIPTION**  

*Enter a description of the resulting data set.*

 300

**FILE FORMAT**  
CSV 

8. Click on **Save and Run flow**

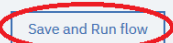
**LOCATION**  
Watson Studio Labs/Data assets 

**DATA SET NAME**  
**female\_human\_traffickin...**  
*Enter a description of the resulting data set.*

☒ If the data set already exists, overwrite the data in the existing data set with the Data Refinery flow output.

File format: CSV

ery flow output details before running the Data Refinery flow.

Close 

9. Click on **View Flow**.

# What's next?

Your Data Refinery flow is currently running. You can view its progress on the Summary and Runs page. When the flow completes, you can view its output from there too.

[Continue Working](#)

[View Flow](#)

10. The flow has completed successfully, and 269 rows have been written to the `female_human_trafficking_output` data asset.

Summary

Source

Data Refinery flow

Output

female\_human\_trafficking

12

Steps

female\_human\_trafficking\_output

Run Environment: Data Refinery Default

Runs

History

Schedule

TIMESTAMP	STATUS	DURATION	ROWS READ / WRITTEN	SIZE	INITIATED BY
14 May 2019 - 10:40 pm	<div><div></div>Completed</div>	18 sec	1592 / 269	0.00475 MB	Andrew Doe

11. Click on **Watson Studio** to return to the **Assets** page.

My Projects


/ Watson Studio Labs


female\_human\_trafficking\_flow.cop

Refine

Summary

Source





female\_human\_trafficking

12

Steps

female\_human\_trafficking\_output

Run Environment: Data Refinery Default

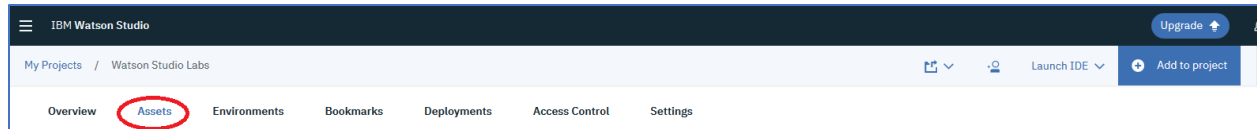
Runs

12. The `female_human_trafficking_output` asset has been added to the **Data Assets**.

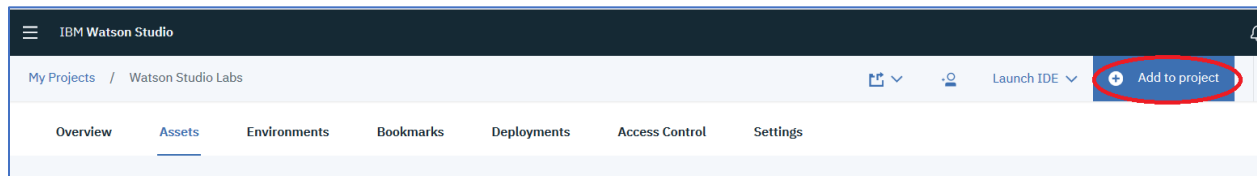
Data assets					
0 asset selected.					
<input type="checkbox"/>	NAME	TYPE	CREATED BY	LAST MODIFIED	ACTIONS
<input type="checkbox"/>	<code>female_human_trafficking_output</code>	Data Asset	Andrew Doe	14 May 2019, 10:41:12 pm	
<input type="checkbox"/>	<code>female_human_trafficking_shaped.csv</code>	Data Asset	Andrew Doe	4 May 2019, 2:01:59 am	
<input type="checkbox"/>	<code>Categories</code>	Data Asset	Andrew Doe	30 Apr 2019, 11:47:30 pm	

# Create a Model to predict VETTING\_LEVEL

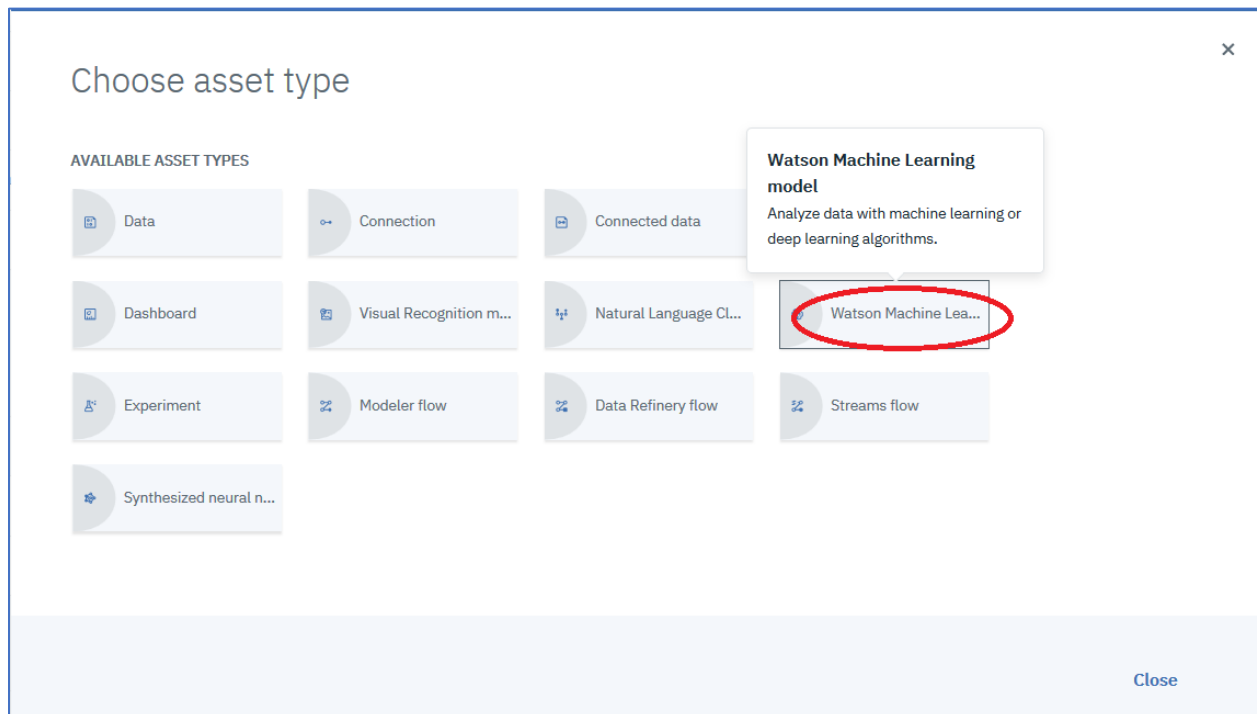
1. If not on the **Assets** page, click on the **Assets** Tab



2. Click on **Add to project**.



3. Click on **Watson Machine Learning**.



4. Enter a **Model Name** (eg FHT\_WML\_Model), optionally a **Description**, leave the default for the **Machine Learning Service** (should be the one created in the pre-reqs), leave **Model builder** selected, select the **Spark Scala Service**, select **Manual**, and click on **Create**.

New model

Define model details

Name

FHT\_WML\_Model

Description

Model description

Machine Learning Service

Machine Learning

Select model type

☒ Model builder ☐ From file ☐ From sample

Select runtime

Only Spark environments supporting Scala kernels can be used for model builder creation.

Default Spark Scala 2.11

The selected runtime uses one driver with 1 vCPU and 4 GB RAM, and 2 executors each with 1 vCPU and 4 GB RAM. This runtime consumes 1.5 capacity units per hour.

⚠ Your Spark runtime will be automatically stopped when you save your model, or after 3 hours of inactivity. To avoid consuming extra capacity unit hours delete your model builder instance or [stop your runtime](#) when you are finished with it.

[Learn more](#) about capacity unit hours and Watson Studio pricing plans.

**Automatic**

Prepare my data and create a model automatically

**Manual**

Let me prepare my data and select which models to train

Need something more flexible? Create a [notebook](#) or design a [Modeler flow](#).

Cancel Create

5. Click on the **female\_human\_trafficking\_output** and click on **Next**

Select data asset

The model builder currently supports CSV files and IBM Db2 Warehouse on Cloud data assets.

Q What asset are you looking for?

Name	Type	Service
<input type="radio"/> Categories.csv	Data Asset	Project
<input type="radio"/> Occupation.csv	Data Asset	Project
<input type="radio"/> female_human_trafficking.csv	Data Asset	Project
<input checked="" type="radio"/> female_human_trafficking_output	Data Asset	Project
<input type="radio"/> female_human_trafficking_shaped.csv	Data Asset	Project

Close Next

6. For **Column value to predict (Label Col)** select **VETTING\_LEVEL**. For **Feature columns** select the following features (**AGE**, **Category**, **COUNTRIES\_VISITED\_COUNT**, and **PASSPORT\_COUNTRY**). Click on the **Multiclass Classification** Box. Adjust the **Validation Split** to be 70%, 15%, 15%. Click on **Add Estimators** to add the specific models to use.

**Select a technique**

Column value to predict (Label Col)  
**VETTING\_LEVEL (Integer)**

Feature columns  
**AGE (Integer), Category (String), COUNTRIES\_VISITED\_COUNT (Integer), PASSPORT\_COUNTRY (String)**

**Binary Classification**  
 Classify new data into defined categories based on existing data. Choose if your label column contains two distinct categories.

**Multiclass Classification**  
 Classify new data into defined categories based on existing data. Choose if your label column contains a discrete number of categories.

**Regression**  
 Predict values from a continuous set of values. Choose if your label column contains a large number of values.

**Validation Split**  
 Train: **70** % Test: **15** % Holdout: **15** %

**Add Estimators**

Configured estimators

7. Select **Random Forest**. You can select more if you wish to see the results of multiple models. Select **Add**.

**Select estimator(s)**

Q What type of estimator are you looking for?

**Decision Tree Classifier**  
 Maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in...

**Random Forest Classifier**  
 Constructs multiple decision trees to produce the label that is a mode of each decision tree. It constructs multiple decision trees to produce the label that is a mode of each tree. It supports both binary and multiclass labels, as well as both categorical features.

**Naive Bayes**  
 Classifies features based on Bayes'

Cancel **Add**

8. Note you can adjust the algorithm's hyperparameters by clicking on **Random Forest**.

Column value to predict (Label Col)  
VETTING\_LEVEL (Integer)

Feature columns  
AGE (Integer), Category (String), COUNTRIES\_VISITED\_COUNT (Integer), P

Configured estimators

Random Forest Classifier  
Not Yet Trained

Binary Multiclass Regression

9. For now, we will leave the hyperparameters unchanged, so click **Cancel**.

## Configure Random Forest Classifier

Feature subset strategy (string)  
auto

Checkpoint interval (integer)  
10

Impurity (string)  
gini

Maximum number of bins (integer)  
--

Cancel Save

10. Select the **Next** button.



Column value to predict (Label Col)

VETTING\_LEVEL (Integer)

Feature columns

AGE (Integer), Category (String), COUNTRIES\_VISITED\_COUNT (Integer), P

**Binary Classification**

Classify new data into defined categories based on existing data. Choose if your label column contains two distinct categories.

**Multiclass Classification**

Classify new data into defined categories based on existing data. Choose if your label column contains a discrete number of categories.

**Regression**

Predict values from a continuous set of values. Choose if your label column contains a large number of values.

Validation Split

Train: 70 % Test: 15 % Holdout: 15 %

Close

Previous

Next

11. The system trains and evaluates each model. If more than one model was selected, the models would be listed in descending order of quality with the best result at the top.  
 Note: if a model fails to run (rare, but happens), select Previous, delete the estimator, and re-add it. Then run again. Click on **Random Forest** (if it is the best) and then click **Save**.

My Projects / Watson Studio Labs / FHT\_WML\_Model

Select Data

Train

Evaluate

Select model

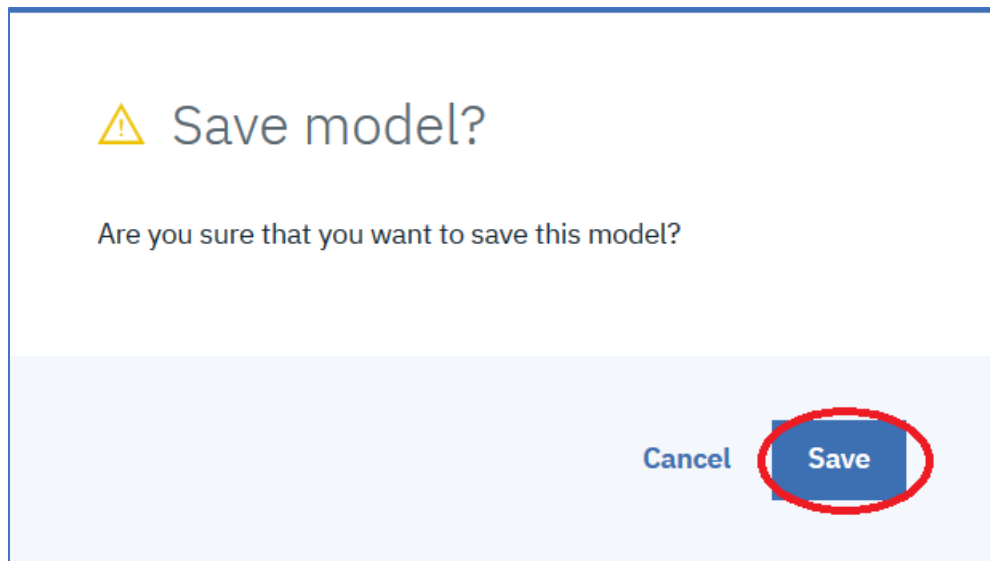
Estimator Type	Status	Performance	Weighted True Positive Rate	Weighted False Positive Rate	Weighted Precision	Weighted F Measure	Weighted Recall	Last Evaluation	Actions
RandomForestClassifier	Trained & Evaluated	Fair	0.72414	0.20095	0.72107	0.70823	0.72414	15 May 2019, 3:44 PM	

Close

Previous

Save

12. Click **Save** again on the next screen. The model is saved in the Model Repository.



13. The system displays the model training summary. Click on **Evaluation**.

My Projects / Watson Studio Labs / FHT\_WML\_Model

MODEL

FHT\_WML\_Model

Overview Evaluation Deployments Lineage

Summary

Machine learning service	Machine Learning
Model Type	wml-1.2
Runtime environment	spark-2.3
Training date	15 May 2019, 3:47 PM
Label column	VETTING_LEVEL
Latest version	12f075bf-954e-4aee-8065-7cf5764d269b

14. The system displays the recorded evaluation statistics for the run. You can also set up Continuous Learning (Performance Monitoring) on this screen. We will not do this now. Select **Deployment**.

My Projects / Watson Studio Labs / FHT\_WML\_Model

FHT\_WML\_Model

Overview Evaluation **Deployments** Lineage

**Last Evaluation Result**

Version	12f075bf-954e-4aee-8065-7cf5764d269b
Phase	setup
accuracy	0.724
weightedPrecision	0.721
weightedRecall	0.724
weightedFMeasure	0.708
weightedFalsePositiveRate	0.201
weightedTruePositiveRate	0.724

## Deploying the Model

We can deploy the model from the model repository to enable applications to invoke it via an API call. This is called a Web Service deployment or Online deployment.

### 1. Click on **Add Deployment**

My Projects / Watson Studio Labs / FHT\_WML\_Model

MODEL

FHT\_WML\_Model

Overview Evaluation **Deployments** Lineage

**Add Deployment**

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Your model is not deployed.			

### 2. Enter **FHT\_WML\_Model\_Deployed** for **Name**, optionally a **Description**, select the **Web Service** radio button, and click on **Save**.

### Define deployment details

**Name**

**Description**

Deployment description

**Deployment type**

☒ Web service
 ☐ Batch prediction
 ☐ Realtime streaming prediction

Cancel
Save

- The system responds with an acknowledgement that the model was successfully deployed. Click on **FHT\_WML\_Model\_Deployed Titanic\_Deployment** to test the deployed API.

MODEL

FHT\_WML\_Model

Overview Evaluation **Deployments** Lineage

Add Deployment

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
FHT_WML_Model_Deployed	DEPLOY_SUCCESS	Web Service	

- The system displays information about the deployed service. Click on **Test** to test out the API.

FHT\_WML\_Model\_Deployed

Overview Implementation **Test**

**Deployment**

Name	FHT_WML_Model_Deployed
Type	Web Service
Deployment ID	30ac498d-1da5-4b97-a96a-66b98a0422ac
Status	DEPLOY_SUCCESS
Asset type	model
Asset name	FHT_WML_Model
Machine learning service	Machine Learning
Created	15 May 2019 04:35pm
Last modified	15 May 2019 08:59pm

- Enter values for the following fields:  
**COUNTRIES\_VISITED\_COUNT: 4**

**AGE:** 24

**PASSPORT\_COUNTRY:** Ghana

Scroll down

**Category:** Information Technology

and then click on **Predict**.

My Projects / Watson Studio Labs / FHT\_WML\_Model / FHT\_WML\_Model\_Deployed

## FHT\_WML\_Model\_Deployed

Overview Implementation **Test**

Enter input data

COUNTRIES\_VISITED\_COUNT

4

AGE

24

PASSPORT\_COUNTRY

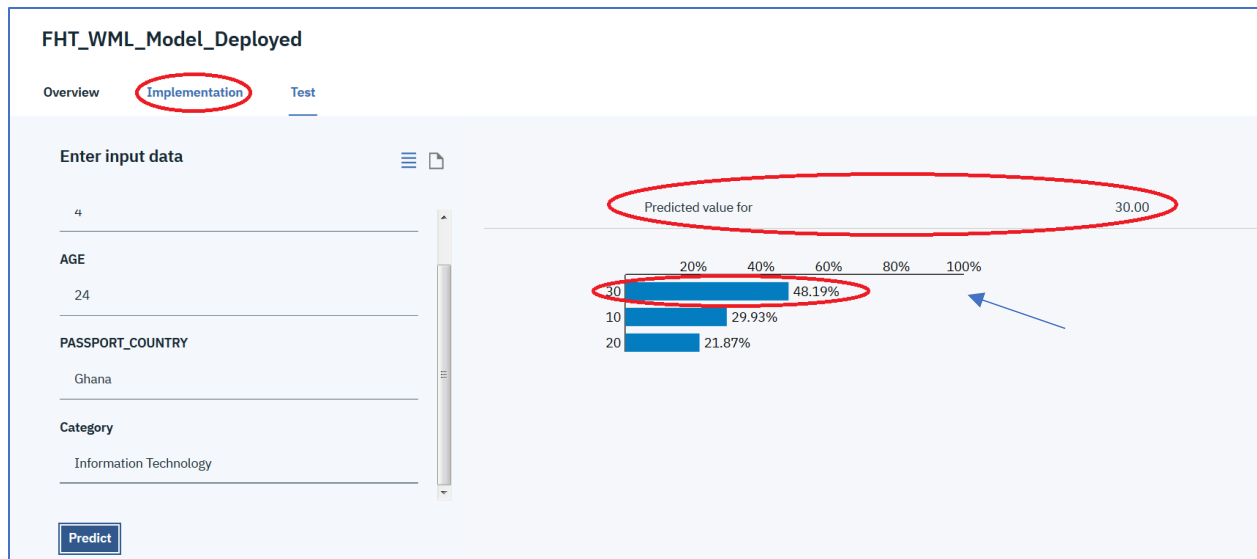
Ghana

Category

Information Technology

**Predict**

7. The predicted result should be returned which indicates that the model has been successfully deployed. Now click on the **Implementation** tab.



8. The Implementation panel provides information for the application developers to invoke the deployed model. It includes sample code in various programming languages and the scoring endpoint to be used when invoking the web service. Open Windows Notepad to copy and paste the scoring endpoint (just leave this panel available to cut and paste the scoring endpoint). We will need the scoring endpoint in the next section.

The screenshot shows the 'Implementation' tab of the 'FHT\_WML\_Model\_Deployed' interface. It includes a 'View API Specification' link. The 'Scoring End-point' is highlighted with a red box: `https://us-south.ml.cloud.ibm.com/v3/wml_instances/02eafab5-7bd2-4ac4-abb1-334f0cc6232c/deployments/30ac498d-1da5-4b97-a96a-66b98a0422ac/online`. Below this, there are fields for 'Authorization: Bearer <token>' and 'Content-type: application/json'. The 'Code Snippets' section shows tabs for cURL, Java, JavaScript, Python, and Scala. The cURL tab is selected, showing a command to retrieve service credentials.

```
# retrieve your $WML_SERVICE_CREDENTIALS_USERNAME, $WML_SERVICE_CREDENTIALS_PASSWORD, and $WML_SERVICE_CREDENTIALS_URL from the
# Service credentials associated with your IBM Cloud Watson Machine Learning Service instance.

curl --basic --user $WML_SERVICE_CREDENTIALS_USERNAME:$WML_SERVICE_CREDENTIALS_PASSWORD $WML_SERVICE_CREDENTIALS_URL/v3/identity/token

# the above CURL request will return an auth token that you will use as $WML_AUTH_TOKEN in the scoring request below
```

## Deploy a simple web front-end to invoke the Watson Machine Learning service

This section provides an example of a simple Python Flask web front-end application that invokes the Female Human Trafficking risk prediction deployed model, demonstrating embedding machine learning in a web app. You will click on a link below that will deploy the

sample Python web application into your IBM Cloud account. A toolchain will be set up for continuous delivery of the application. The application code will be cloned from a public Git repository into a private Git repo in your account that will be set up as part of the toolchain. Each time you commit changes to the repo, the app will be built and deployed.

The toolchain uses tools that are part of the Continuous Delivery service. If an instance of that service isn't already in your account, when you click **Deploy**, it is automatically added with the free [Lite](#) plan selected.

The steps below guide you in configuring the application to connect to your Watson Machine Learning service, and to update the application with the deployed model's scoring endpoint.

1. Click on the **Deploy to IBM Cloud** link below to deploy a sample Python Flash web application into your IBM Cloud account. Note you may get this message – “*An IBM Cloud account is required. To get started, click Log In or Sign Up at the top of this page*”. If you get this message, click on **Log In**.

[Deploy to IBM Cloud](#)

2. Scroll down to the bottom. Click on the **Create+** button to create an IBM Cloud API key.

The Delivery Pipeline automates continuous deployment.

**App name:** ⓘ

FHT-20190516021126700

**IBM Cloud API Key:** ⓘ

IBM Cloud API Key

The value is required.

**Create +**

Region	Organization	Space
The IBM Cloud CF region	The IBM Cloud CF org	The IBM Cloud CF space
The value is required.	The value is required.	The value is required.

3. Click on the **Create** button.

×

Create API key

The following API key will be created to deploy your application from the Delivery Pipeline.

API Key for FHT-20190516021126700

API keys can be managed from **Manage > Security > Platform API keys**.

Cancel

Create

4. Please wait until the Region, Organization, and Space are filled in. **Note that these must match the corresponding Region, Organization, and Space where the FHT\_WML\_Model\_Deployed is deployed.** Switch the **Region** to Dallas (Production), the Organization should display the userid, and the space should be filled in as well. If this is not the case, try a different **Region**. Click on **Deploy**.

Git Repos and Issue Tracking

Eclipse Orion Web IDE

Delivery Pipeline

The Delivery Pipeline automates continuous deployment.

App name:

FHT-20190516021126700

ⓘ

IBM Cloud API Key:

.....

ⓘ

Create +

Region

Dallas (Production) ▾

Organization

wsuser19000@gmail.c... ▾

Space

dev ▾

Deploy



5. Your app is being created! To watch the pipeline deploy your app, click **Delivery Pipeline**.

The screenshot shows the Azure DevOps interface for an application named **FHT-20190516022609200**. The interface is divided into three main sections: **THINK**, **CODE**, and **DELIVER**. Each section contains a card representing a different tool or service. The **DELIVER** section is highlighted with a red circle, indicating the **Delivery Pipeline** option. Below the stages, there are four cards: **Issues**, **Git**, **Delivery Pipeline**, and **Eclipse Orion Web IDE**. Each card has a 'Configured' status.

Resource Group: Default    Location: Dallas    [Add tags](#)

✓ Your app is being created! Quick start: To watch the pipeline deploy your app, click **Delivery Pipeline**. After the app is deployed, you can view the running app.

**THINK**    **CODE**    **DELIVER**

**Issues**  
FHT-2019051602260...  
✓ Configured

**Git**  
FHT-2019051602260...  
✓ Configured

**Delivery Pipeline**  
FHT-2019051602260...  
✓ Configured

**Eclipse Orion Web IDE**  
✓ Configured

6. After the app is deployed successfully (should say Deploy Passed in the Deploy stage-may take about 2 minutes), view the running app by clicking on **View Console**

The screenshot shows two side-by-side panels for a CI/CD pipeline. The left panel is titled 'Build Stage' and shows a 'STAGE PASSED' status. It includes a 'LAST INPUT' section with a commit by 'bleonardb3' and a 'JOBS' section showing a 'Build' job that passed 3 minutes ago. The right panel is titled 'Deploy Stage' and also shows a 'STAGE PASSED' status. It includes a 'LAST INPUT' section showing 'Build 2' and a 'JOBS' section showing a 'Deploy' job that passed 'now'. A red circle highlights the 'Deploy Passed now' status. Below the jobs, the 'LAST EXECUTION RESULT' section shows 'FHT-20190516022609200' with a 'view console' link circled in red.

7. Click on **Visit App URL**

The screenshot shows a resource page for an application. At the top, it says 'Resource list /'. Below this, there is a card for the application 'FHT-20190516022609200'. The card includes a '.py' icon, the application name, a status indicator 'This app is awake.', and a 'Visit App URL' link circled in red. At the bottom of the card, it shows 'Org: wsuser19000@gmail.com', 'Location: Dallas', and 'Space: dev'.

8. The web form collecting the FHT data should appear. Note that the application is not functional until we connect it to the Watson Machine Learning service so if you Submit you will get an error! Close the FHT Prediction browser tab.

## FHT Prediction

To determine the trafficking risk prediction, please enter the following:

### Categories

Sports/Travel

Age:

Number of countries visited:

### Passport Country

- ☐ Ghana
- ☐ Brazil
- ☐ Pakistan
- ☐ Bangladesh
- ☐ Haiti
- ☐ India

Submit

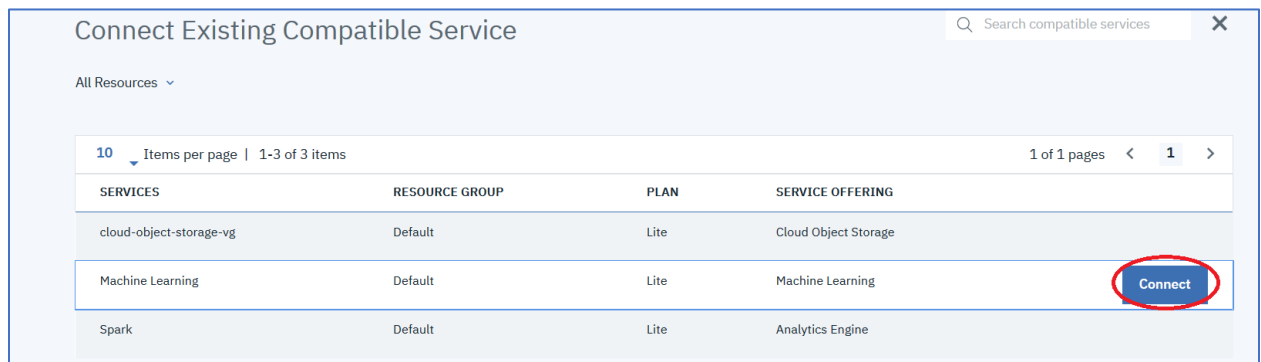
9. We are now going to connect the application to the Watson Machine Learning service that was created earlier. Scroll down until you see the Connections panel. Click on **Create Connection**.

### Connections

No services are connected to this app  
You can bind a service:

Create connection

10. You should see at least 3 services listed, a Cloud Object Storage service, a Spark service, and a Watson Machine Learning service. Point the cursor on the **Machine Learning** service for your application, and then click on **Connect**.



Connect Existing Compatible Service

Search compatible services

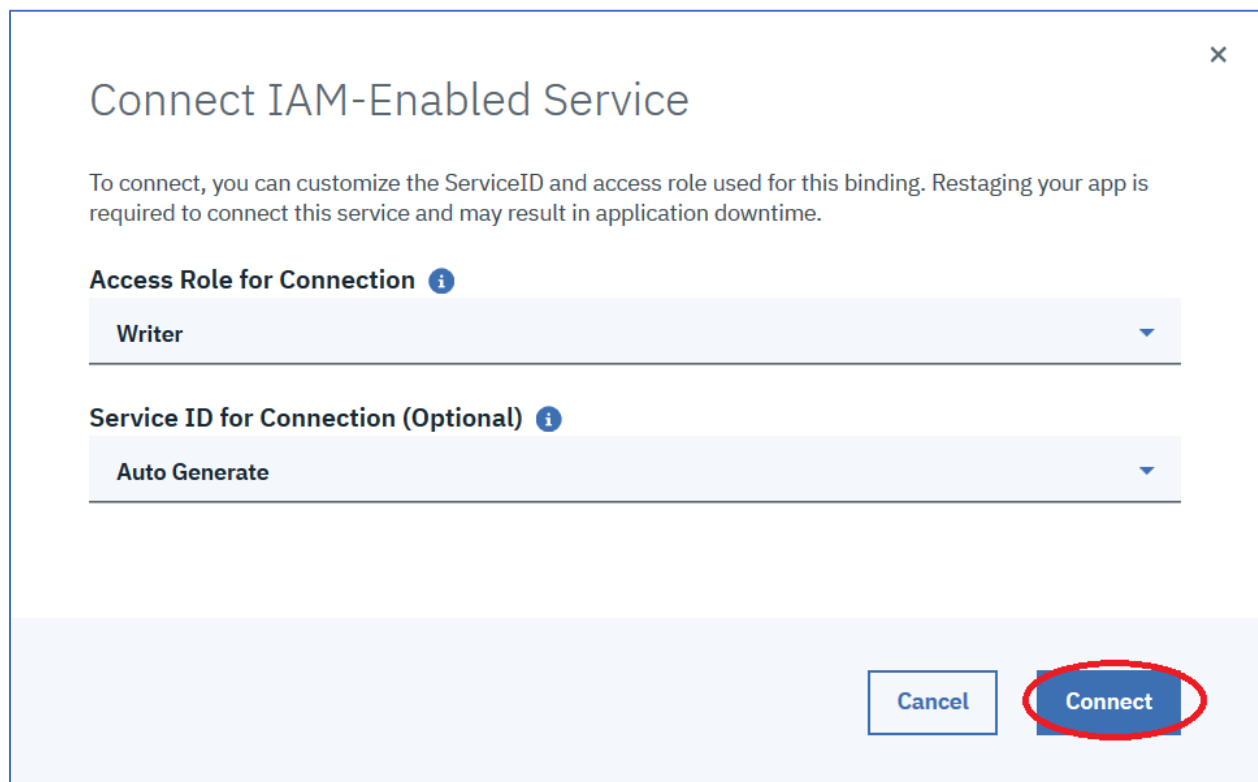
All Resources

10 Items per page | 1-3 of 3 items 1 of 1 pages < 1 >

SERVICES	RESOURCE GROUP	PLAN	SERVICE OFFERING
cloud-object-storage-vg	Default	Lite	Cloud Object Storage
Machine Learning	Default	Lite	Machine Learning
Spark	Default	Lite	Analytics Engine

Connect

11. A **Connect IAM-enabled service** pop up will appear. Click **Connect**.



Connect IAM-Enabled Service

To connect, you can customize the ServiceID and access role used for this binding. Restaging your app is required to connect this service and may result in application downtime.

Access Role for Connection ⓘ

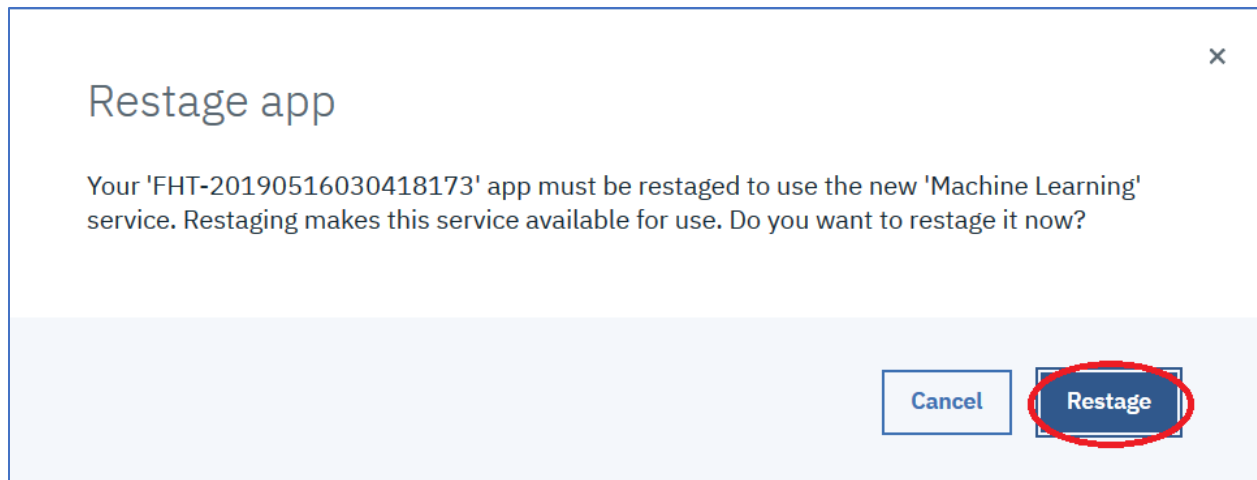
Writer


Service ID for Connection (Optional) ⓘ

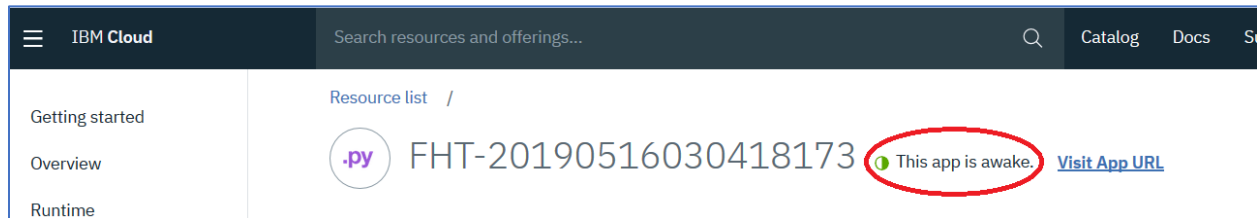
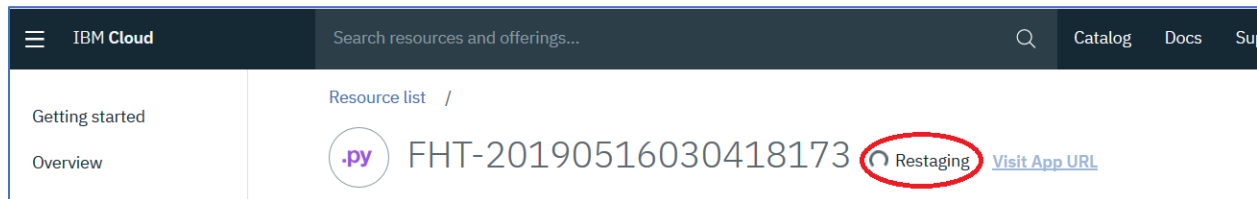
Auto Generate


Cancel Connect

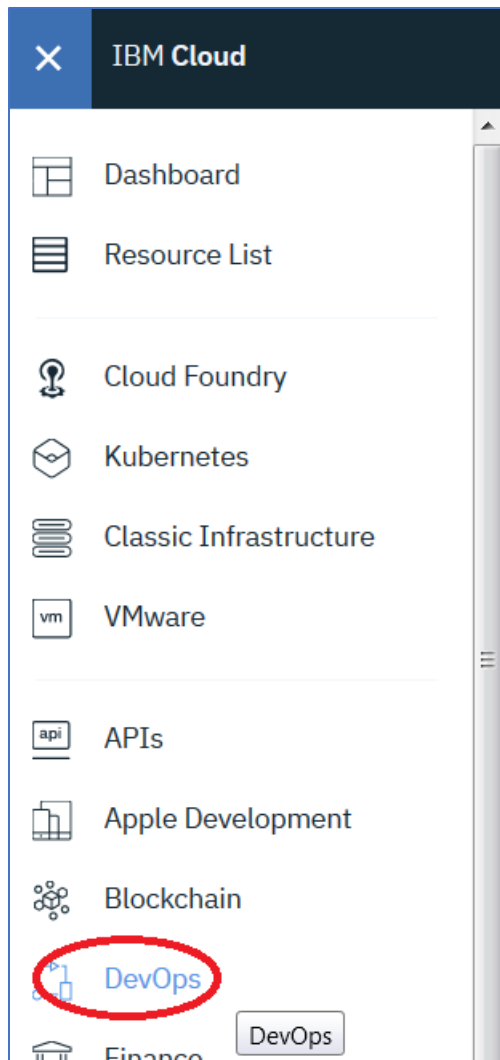
12. A **Restage app** pop up will appear. Click on **Restage**.



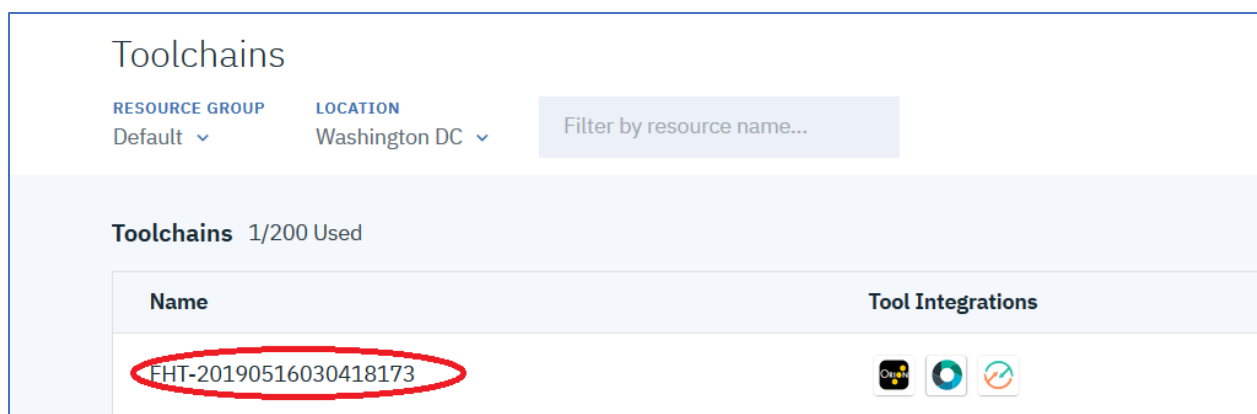
13. Wait for the application status to change from **Restaging** to  This app is awake , or something similar.



14. We now have tied the web application to the Watson Machine Learning service. Note that the Watson Machine Learning service could have more than one deployed model available to select and then embed in the web application. We now need to copy the scoring endpoint, which we previously copied and pasted into Notepad, and paste it in the web application code. Click on the  icon and click on DevOps in the pulldown to navigate to the Toolchain.




15. We are now going to paste the scoring endpoint into the application code. Click on the Toolchain (FHT-2019xxxxx below).




16. Click on the Eclipse Orion Web IDE. The IDE will enable the editing of the source code to update the scoring endpoint url.


Toolchains /

 FHT-20190516030418173 [Visit App URL](#)

Resource Group: Default Location: Washington DC [Add tags](#)


 **Your app is being created! Quick start:** To watch the pipeline deploy your app, click **Delivery Pipeline**. After the app is deployed, you app.

THINK




**Issues**  
FHT-2019051603041...  
✓ Configured

CODE




**Git**  
FHT-2019051603041...  
✓ Configured

DELIVER



**Delivery Pipeline**  
FHT-2019051603041...  
✓ Configured

 **Eclipse Orion Web IDE**

17. Click on the FHT.py file. This is a python source file.

FHT-20190516030418173


▼ README.md

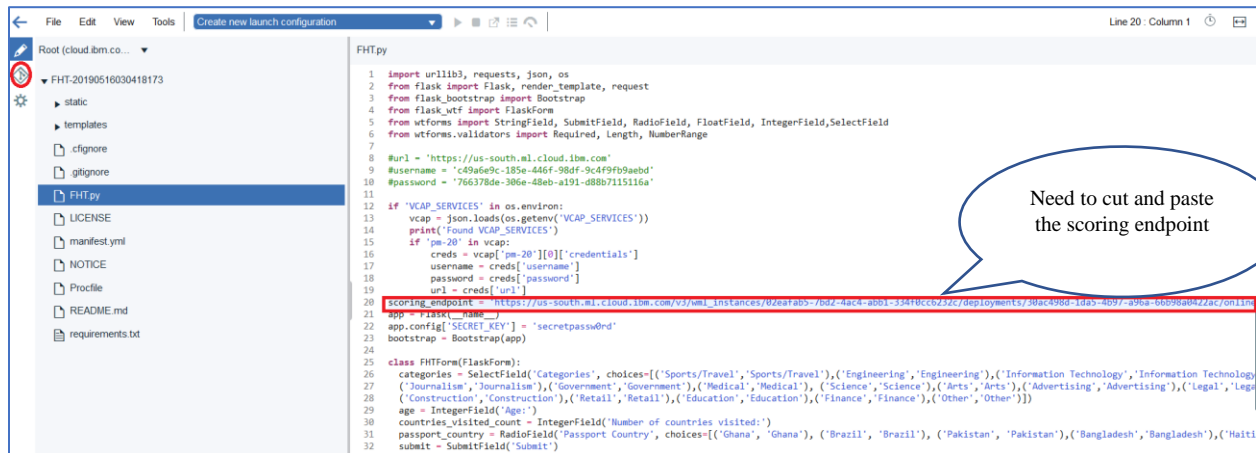
**This repository contains a Python Flask program that invokes a Watson Machine Learning service to predict trafficking risk based on travel itinerary data**

▼ FHT-20190516030418173

Name	Date Modified	Size
static	5/15/2019, 11:37:15 PM	--
templates	5/15/2019, 11:37:15 PM	--
.cignore	5/15/2019, 11:37:16 PM	1 KB
.gitignore	5/15/2019, 11:37:16 PM	1 KB
<b>FHT.py</b>	5/15/2019, 11:37:15 PM	4 KB
LICENSE	5/15/2019, 11:37:15 PM	12 KB
manifest.yml	5/15/2019, 11:37:15 PM	1 KB
NOTICE	5/15/2019, 11:37:15 PM	1 KB
Profile	5/15/2019, 11:37:15 PM	1 KB
README.md	5/15/2019, 11:37:15 PM	1 KB
requirements.txt	5/15/2019, 11:37:15 PM	1 KB

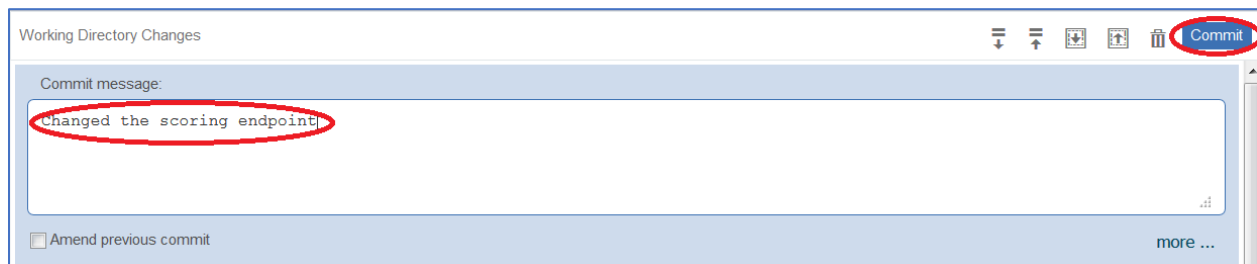
18. Go back to the Notepad file and copy the scoring endpoint to the clipboard. Look around line 20 in the FHT.py file for the “scoring endpoint =”. Select the scoring endpoint value in line 20 (starting with https:// may want to use Shift-End to get to the end of the

line, and then back up one space to not select the endpoint quote – if you do just make sure to put it back in). Enter Ctrl-V to paste the new scoring endpoint from your Notepad file. Enter Ctrl-S or File > Save to save the file. Then click on the  icon on the top left.



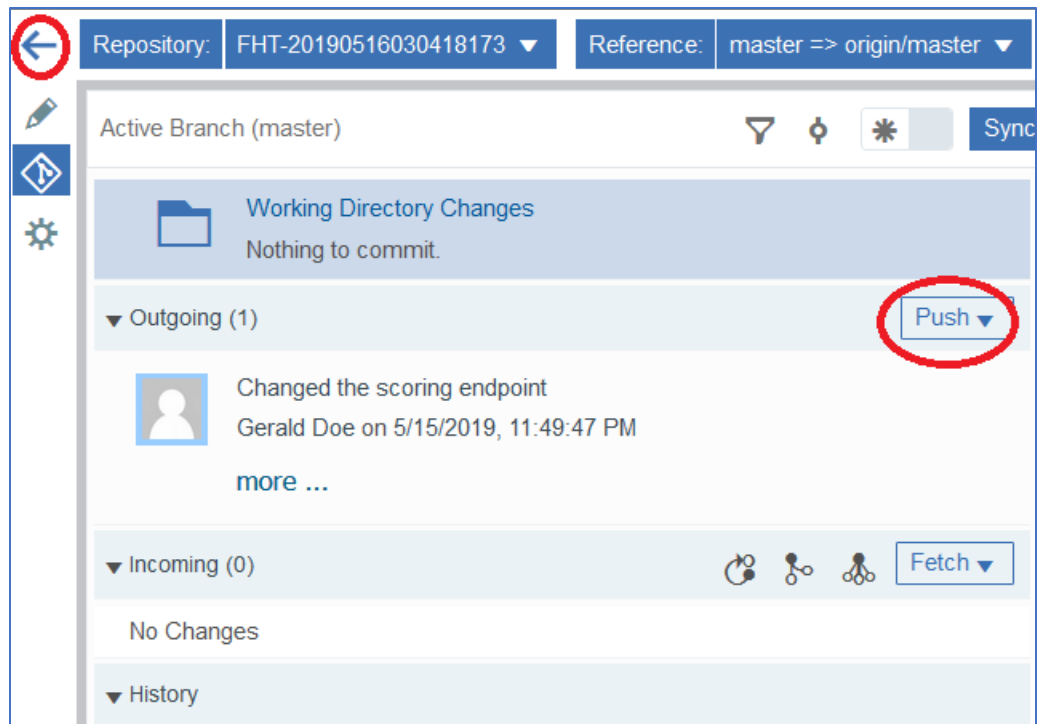
```
1 import urllib3, requests, json, os
2 from flask import Flask, render_template, request
3 from flask_bootstrap import Bootstrap
4 from flask_wtf import FlaskForm
5 from wtforms import StringField, SubmitField, RadioField, FloatField, IntegerField, SelectField
6 from wtforms.validators import Required, Length, NumberRange
7
8 #url = 'https://us-south-ml.cloud.ibm.com'
9 #username = 'c09a6e9c-185e-446f-9c4f-9b9aebd'
10 #password = '766378de-306e-48eb-a191-d88b7115116a'
11
12 if 'VCAP_SERVICES' in os.environ:
13     vcap = json.loads(os.getenv('VCAP_SERVICES'))
14     print('Found VCAP_SERVICES')
15     if 'pe-20' in vcap:
16         creds = vcap['pe-20'][0]['credentials']
17         username = creds['username']
18         password = creds['password']
19         url = creds['url']
20
21 scoring_endpoint = 'https://us-south-ml.cloud.ibm.com/v3/ml_instances/62eafab5-b02-4ac8-a081-394f8c6252c/deployments/30ac398d-1da5-4097-a96a-b0898a8422ac/online'
22
23 app = Flask(__name__)
24 app.config['SECRET_KEY'] = 'secretpasswd'
25 bootstrap = Bootstrap(app)
26
27 class FHTForm(FlaskForm):
28     categories = SelectField('Categories', choices=[('Sports/Travel', 'Sports/Travel'), ('Engineering', 'Engineering'), ('Information Technology', 'Information Technology'), ('Journalism', 'Journalism'), ('Government', 'Government'), ('Medical', 'Medical'), ('Science', 'Science'), ('Arts', 'Arts'), ('Advertising', 'Advertising'), ('Legal', 'Legal'), ('Construction', 'Construction'), ('Retail', 'Retail'), ('Education', 'Education'), ('Finance', 'Finance'), ('Other', 'Other')])
29     age = IntegerField('Age:')
30     countries_visited_count = IntegerField('Number of countries visited:')
31     passport_country = RadioField('Passport Country', choices=[('Ghana', 'Ghana'), ('Brazil', 'Brazil'), ('Pakistan', 'Pakistan'), ('Bangladesh', 'Bangladesh'), ('Haiti', 'Haiti')])
32     submit = SubmitField('Submit')
```

19. The next step is to commit the change to the git repository. Enter “Changed the Scoring Endpoint” in the Enter Commit Message field, and then click on **Commit**.




20. Then click on **Push** to push the changes to the central Git repo which will start the build and deploy of the application. Click on the left arrow to return to the Toolchain.







21. Click on the **Delivery Pipeline** to view status of the deployment as before.

 FHT-20190516030418173 [Visit App URL](#)


Resource Group: Default      Location: Washington DC      [Add tags](#)

 **Your app is being created! Quick start:** To watch the pipeline deploy your app, click **Delivery Pipeline** app.


THINK


  
**Issues**  
FHT-2019051603041...  
✓ Configured

CODE

  
**Git**  
FHT-2019051603041...  
✓ Configured

DELIVER

  
**Delivery Pipeline**  
FHT-2019051603041...  
✓ Configured

  
**Eclipse Orion Web IDE**  
✓ Configured

22. Once the Deployment status shows **Deploy passed now** it shouldn't take longer than 2 minutes (reload the browser in case the UI didn't update after 2 minutes). Click on **View Console**.

The screenshot displays two stages of a Jenkins pipeline: 'Build Stage' and 'Deploy Stage'. Both stages are marked as 'STAGE PASSED' with a green bar at the top. The 'Build Stage' shows a 'LAST INPUT' from 'Gerald Doe' and a 'JOBS' section with 'Build' passed 4m ago. The 'Deploy Stage' shows a 'LAST INPUT' from 'Stage: Build Stage / Job: B...' and a 'JOBS' section with 'Deploy' passed 2m ago. In the 'Deploy Stage', the 'Deploy' job is circled in red. Below the jobs, the 'LAST EXECUTION RESULT' for 'Build 2' is shown, with a 'View console' link circled in red.

23. Click on **Visit App URL**.

The screenshot shows the Jenkins application resource page. At the top, it says 'Resource list /'. Below this, there is a large ID 'FHT-20190516030418173' next to a green dot and the text 'This app is awake.' To the right of this text is a 'Visit App URL' link circled in red. Below the ID, there is a section with 'Org: wsuser19000@gmail.com', 'Location: Dallas', and 'Space: dev'.

24. The web form should appear. Enter data in all the fields and click on the **Submit** button.

To determine the trafficking risk prediction, please enter the following:

**Categories**

Information Technology

Age: 24

Number of countries visited: 4

**Passport Country**

☒ Ghana

☐ Brazil

☐ Pakistan

☐ Bangladesh

☐ Haiti

☐ India

Submit

25. You should see something similar to the following depending on the values of the input fields that you entered. Click on the **Try Again!**, if you want to experiment with different inputs.

FHT Prediction

prediction: low risk  
probability: 0.481944116123

[Try Again](#)