ORACLE

Feature DataStore Lab

Caixabank Developer Session

June 2021

Oracle Cloud Environment for Feature DataStore

Users	Jupyter URL	login
holuser01 to holuser10	https://129.213.113.81/hopsworks/	holuser01@oracle.com to holuser10@oracle.com
holuser11 to holuser20	https://150.136.14.68/hopsworks/	holuser11@oracle.com to holuser20@oracle.com

Instances *in* DataScienceHOL *Compartment*

The Compute service helps you provision VMs and bare metal instances to meet your compute and application requirements. An instance is a compute host. Choose between virtual machines (VMs) and bare metal instances. The image that you use to launch an instance determines its operating system and other software.

Create Instance

Name	State	Public IP	Shape	OCPU Count	Memory (GB)	Availability Domain	Fault Domain	Created
featureds11	Running	150.136.14.68	VM.Standard.E4.Flex	8	128	AD-2	FD-1	Sun, Jun 20, 2021, 19:50:42 UT
featureds11	Running	129.213.113.81	VM.Standard.E4.Flex	8	128	AD-1	FD-2	Sun, Jun 20, 2021, 19:32:16 UT

Environment to download Documentation / Scripts

GitHub Repository: https://github.com/operard/featuredslab

Documentation for Workshop: https://github.com/operard/featuredslab/tree/main/doc

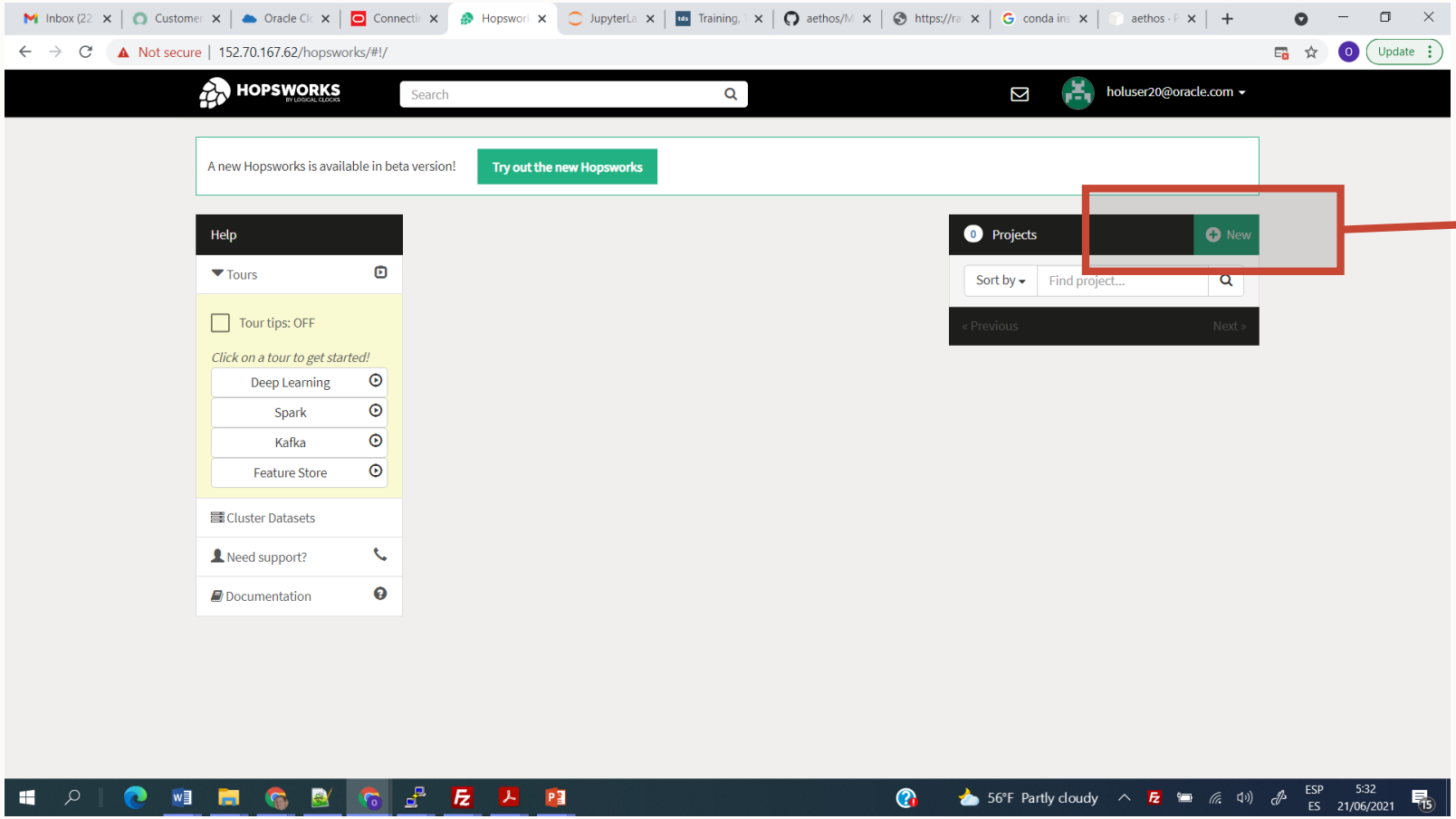
Documentation of Hopsworks:

https://github.com/logicalclocks/hops-examples

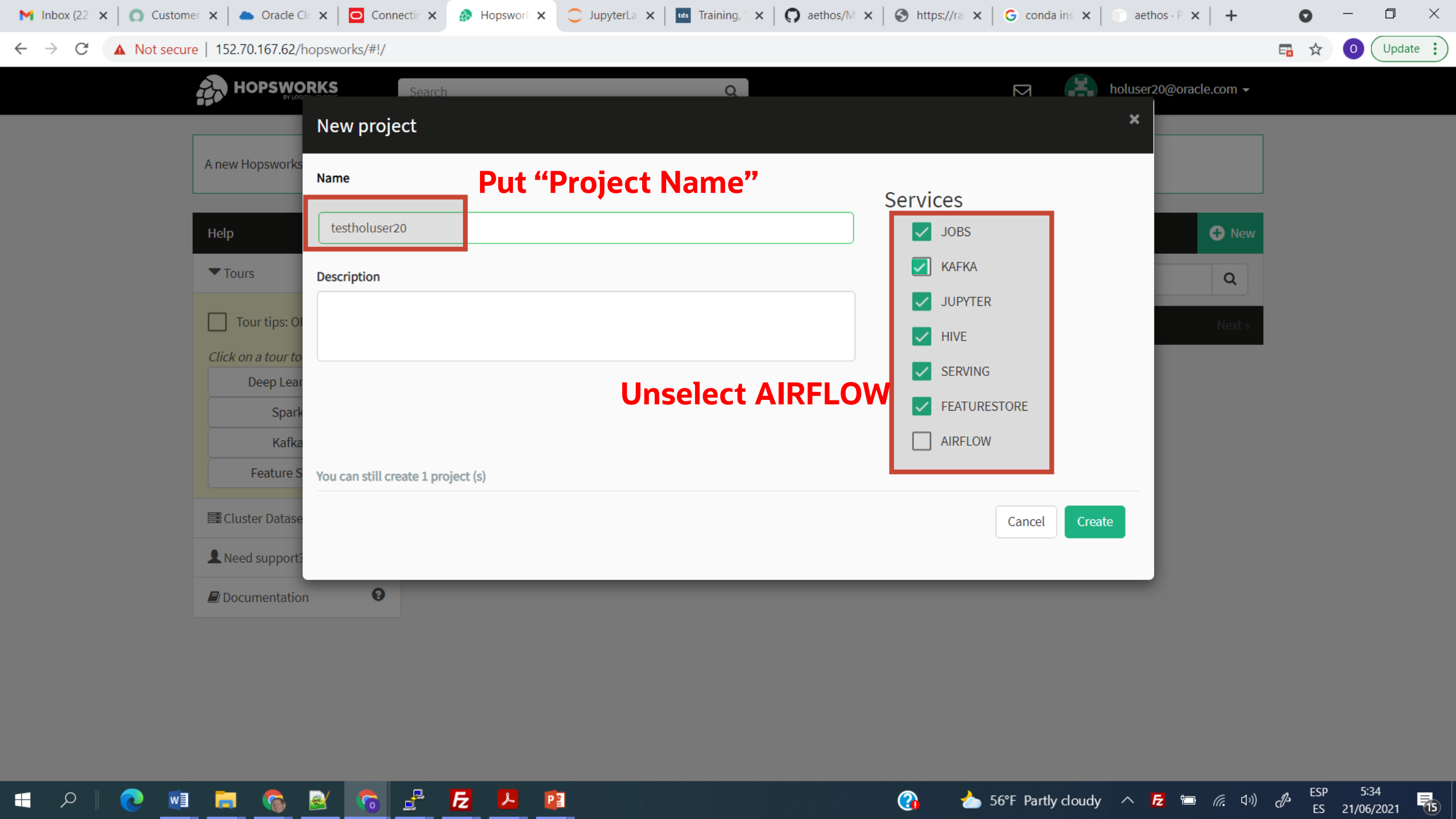
https://examples.hopsworks.ai/

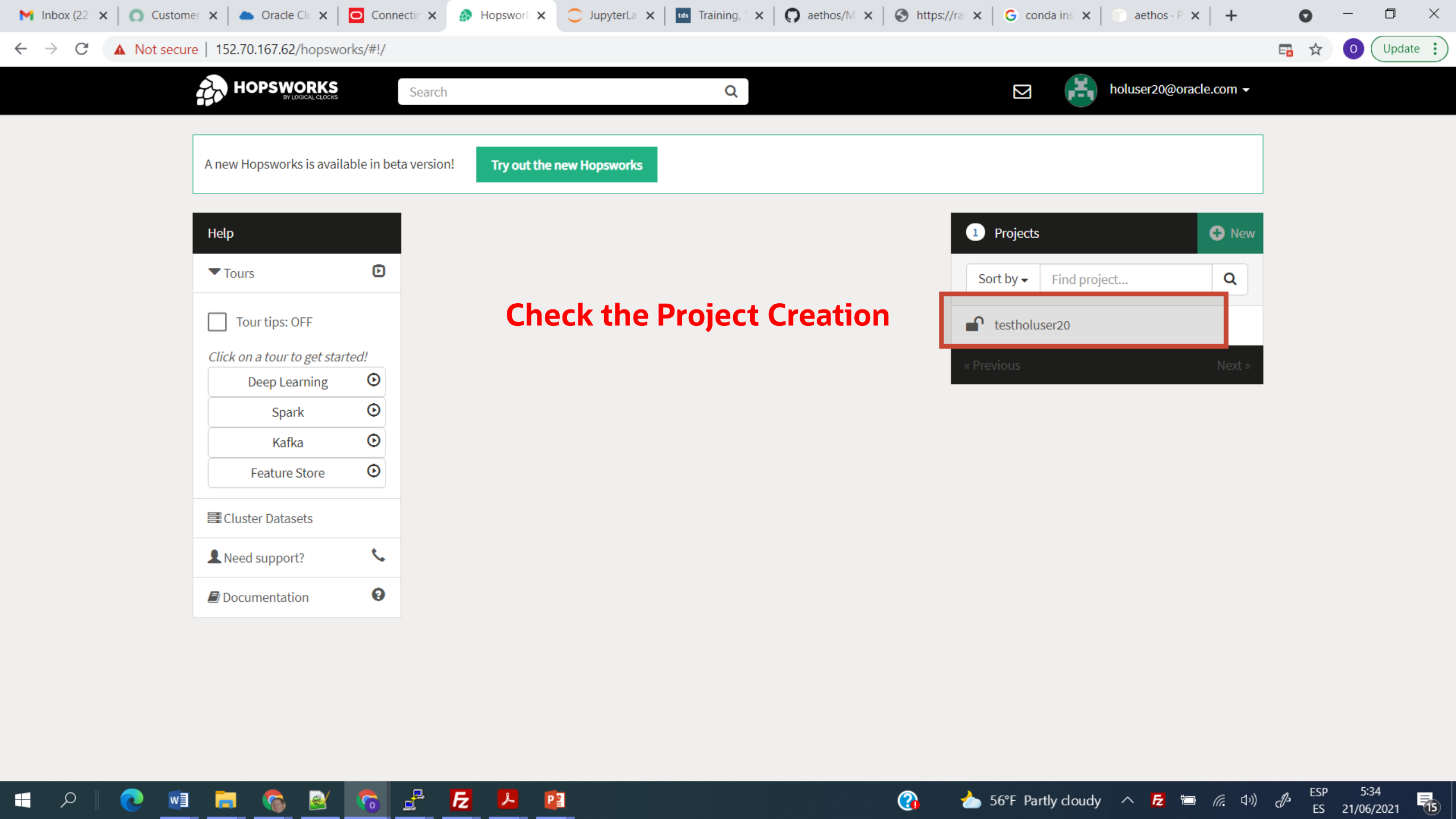
STEP1: Create a project and upload datasets

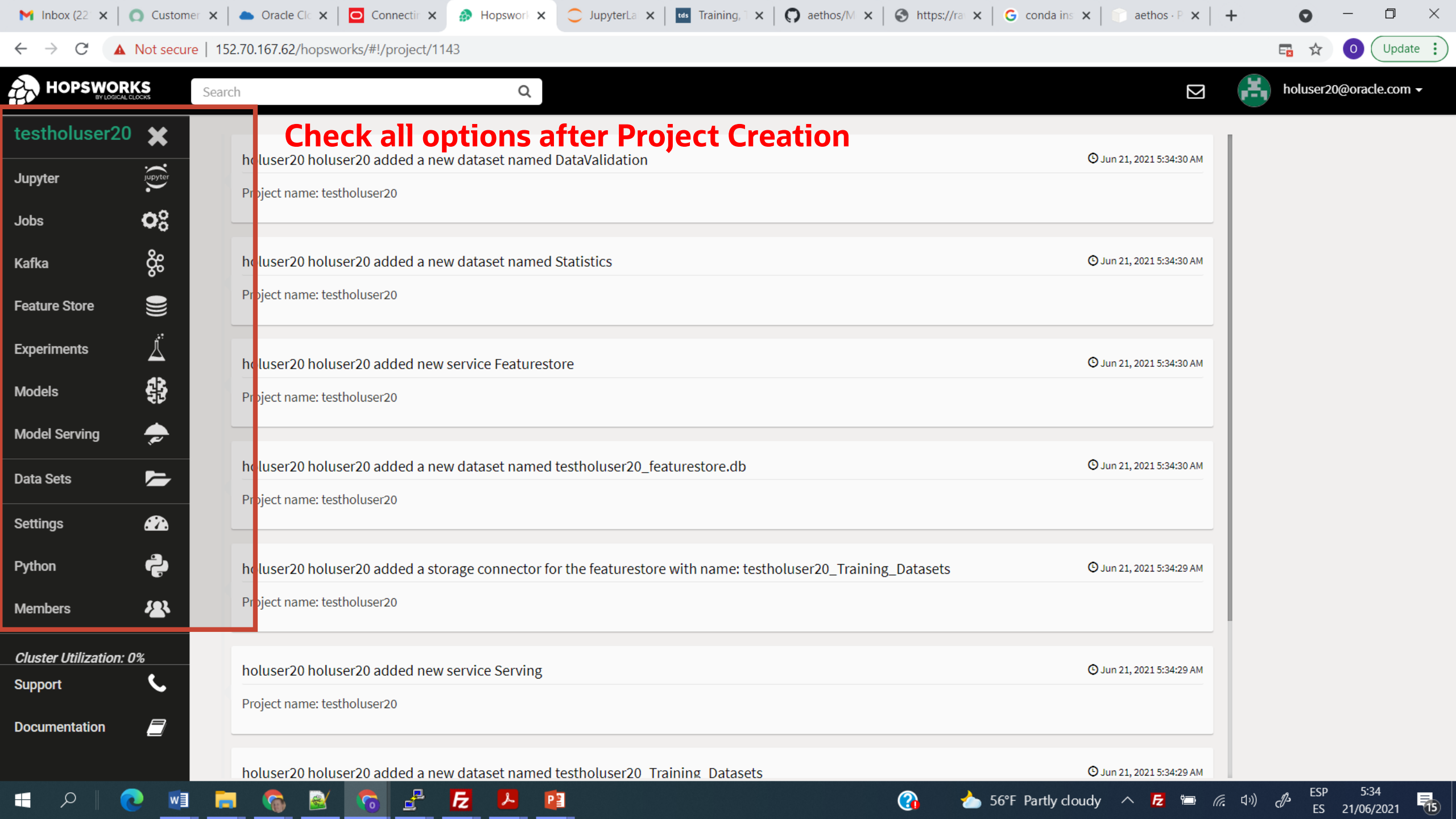
Create a project

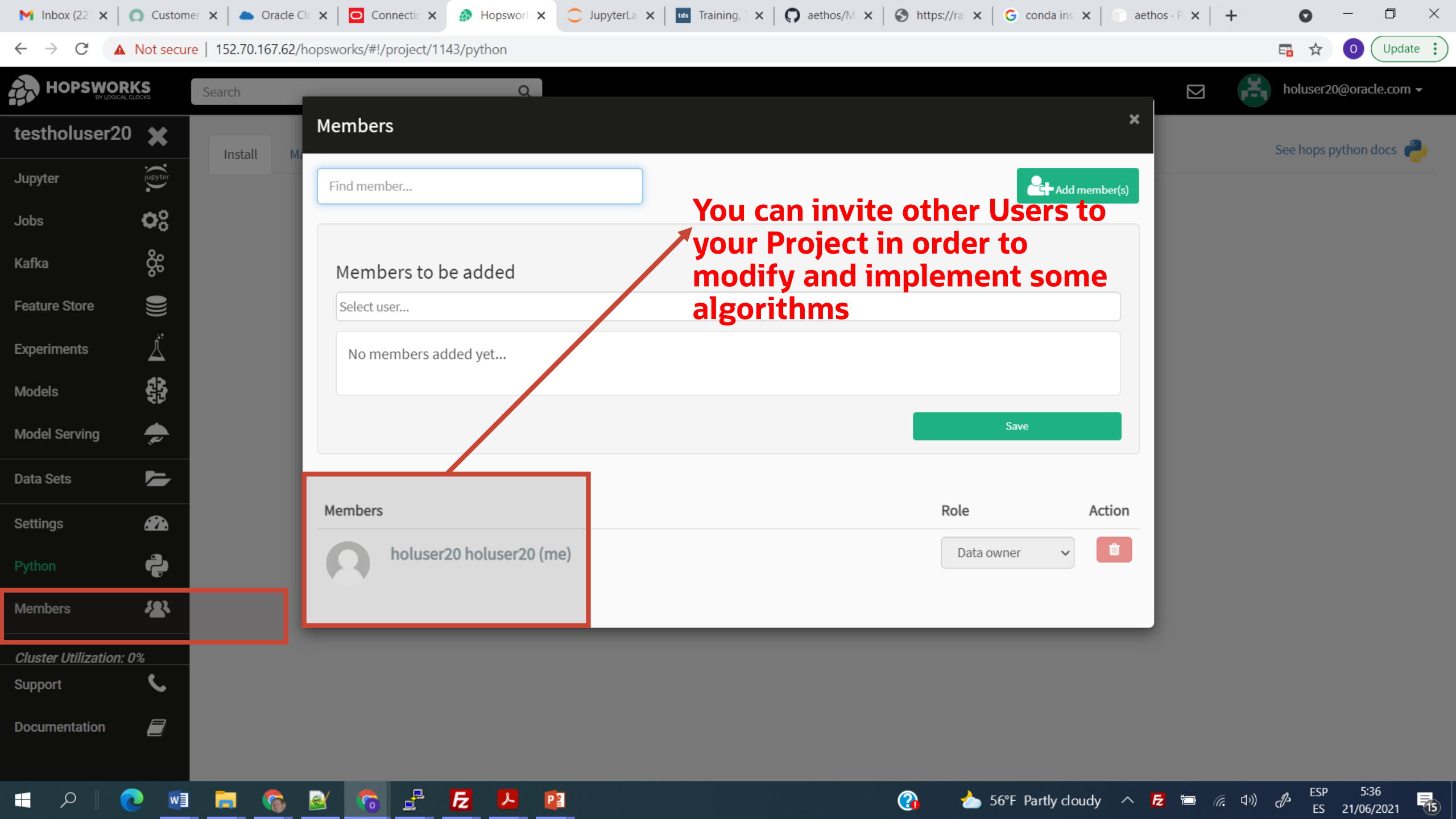


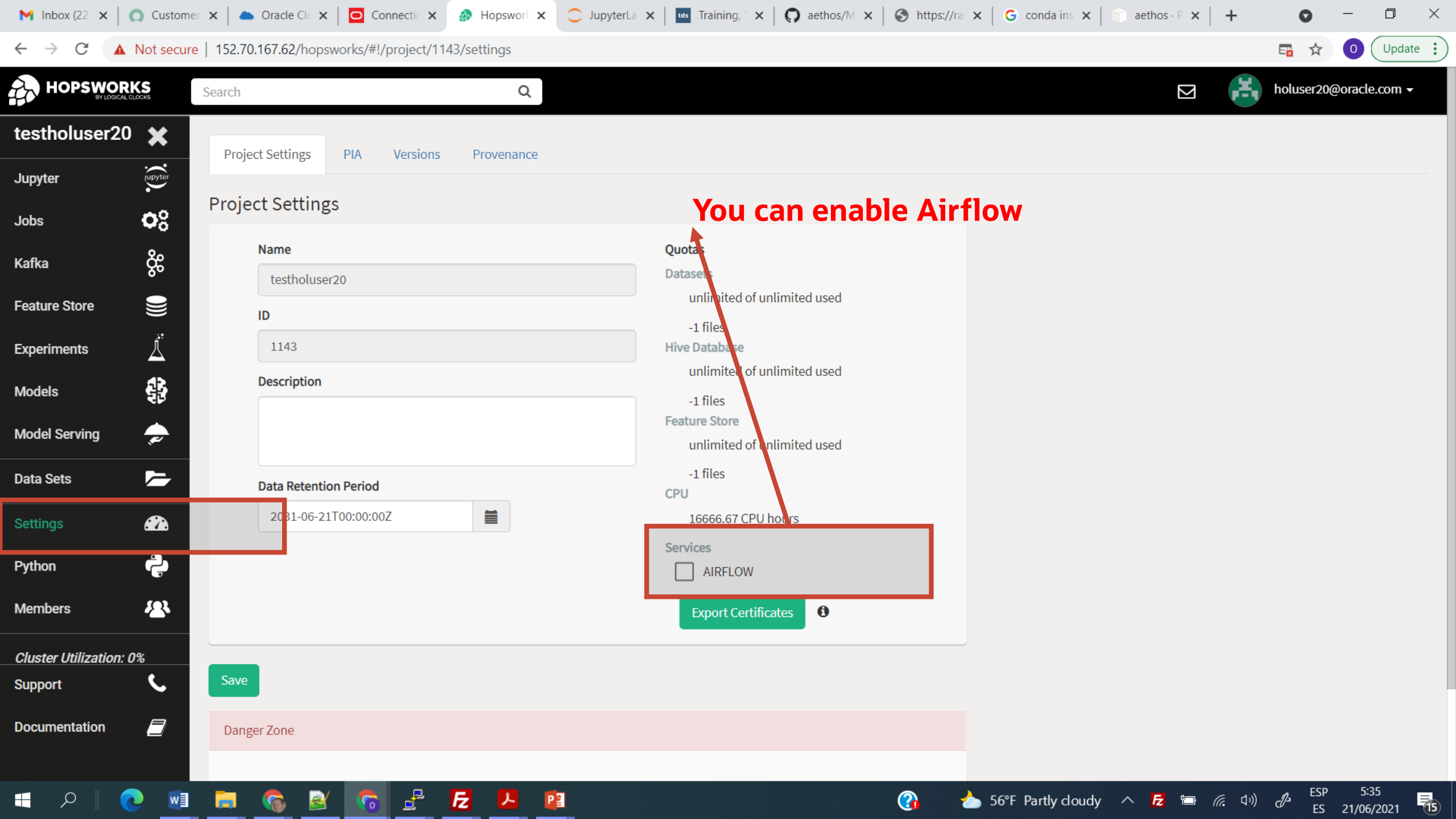
Click NEW button

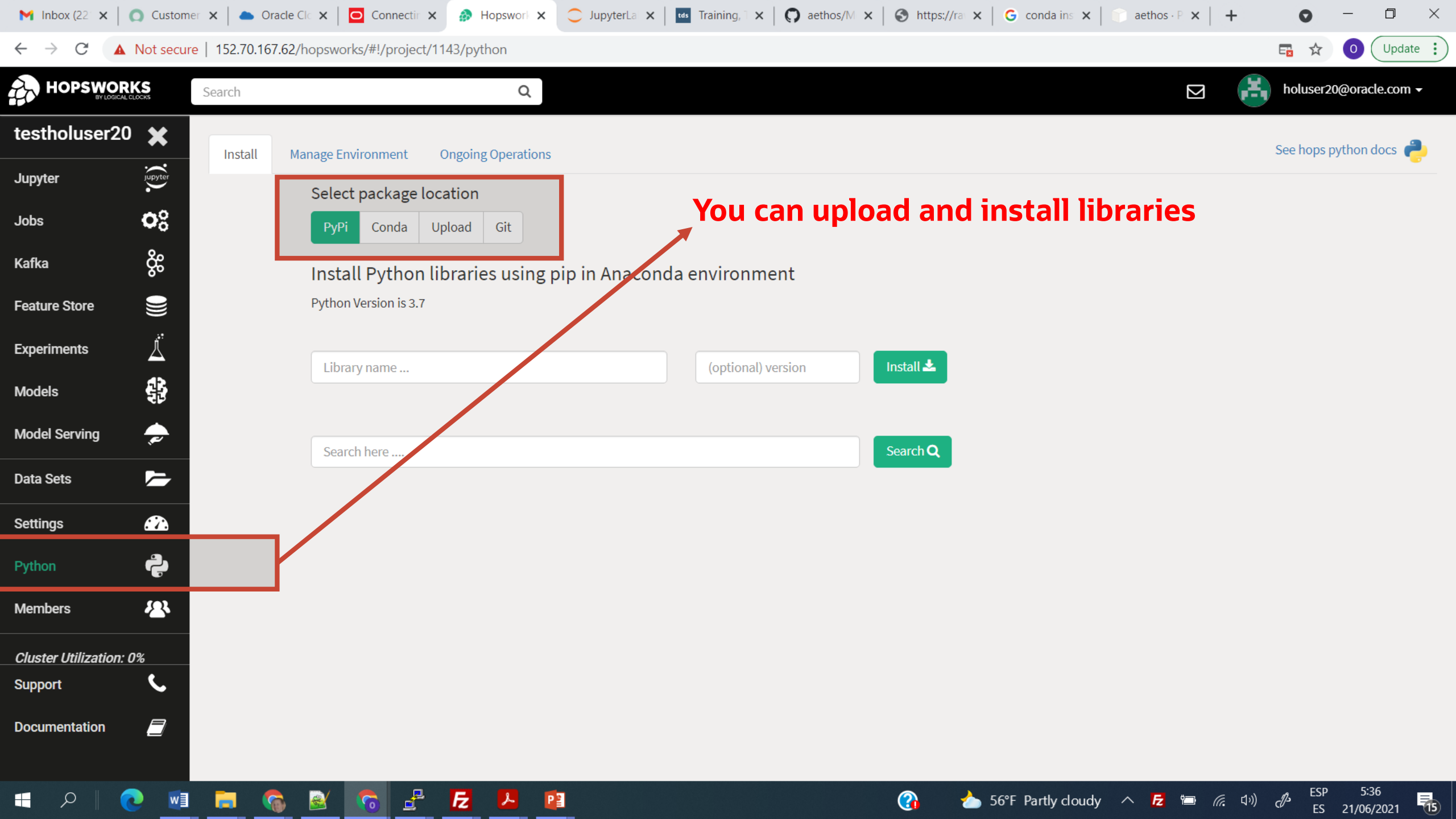

Not secure | 152.70.167.62/hopsworks/#!/
HOPSWORKS
holuser20@oracle.com
New project
Name
Put “Project Name”
testholuser20
Description
Unselect AIRFLOW
Services
JOBS
KAFKA
JUPYTER
HIVE
SERVING
FEATURESTORE
AIRFLOW
You can still create 1 project (s)
Cancel
Create
Help
Tours
New
Cluster Datase
Need support?
Documentation
56°F Partly cloudy
ESP
ES
5:34
21/06/2021

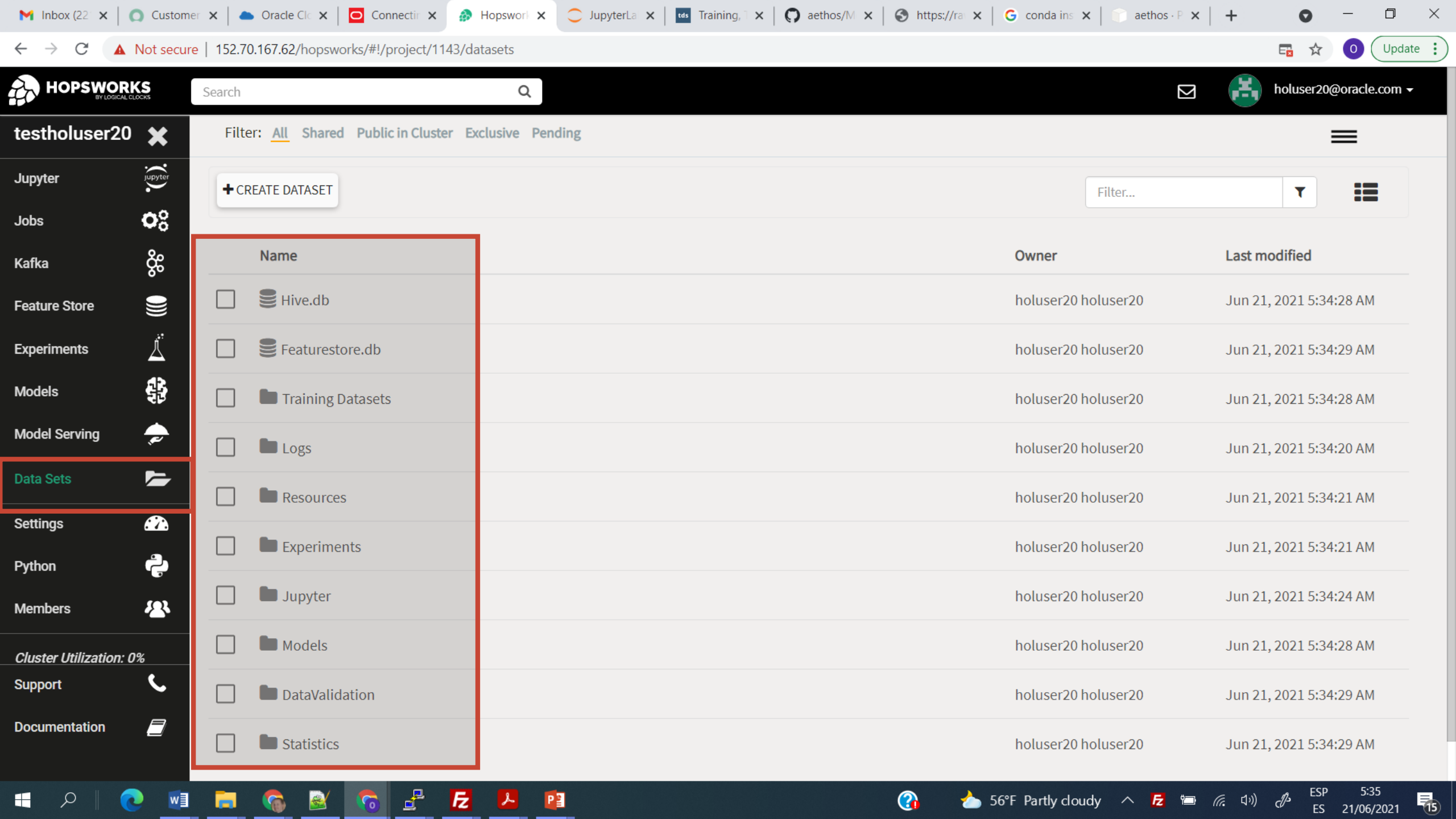

Not secure | 152.70.167.62/hopsworks/#!/
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
A new Hopsworks is available in beta version!
Try out the new Hopsworks
Help
Tours
Tour tips: OFF
Click on a tour to get started!
Deep Learning
Spark
Kafka
Feature Store
Cluster Datasets
Need support?
Documentation
Check the Project Creation
1 Projects
New
Sort by
Find project...
testholuser20
« Previous
Next »
56°F Partly cloudy
ESP
ES
5:34
21/06/2021

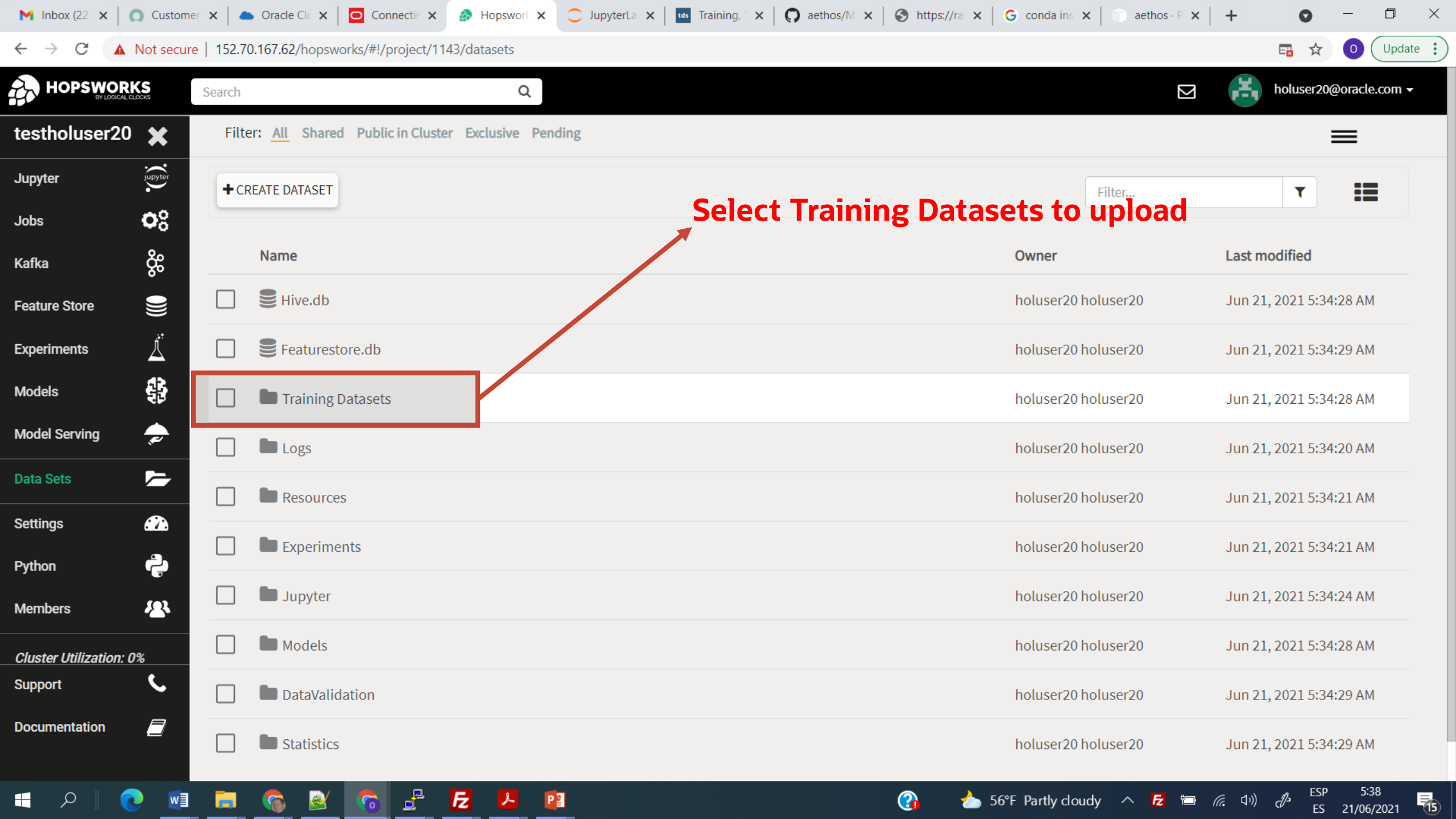

Not secure | 152.70.167.62/hopsworks/#!/project/1143
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
testholuser20
Check all options after Project Creation
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
holuser20 holuser20 added a new dataset named DataValidation
Jun 21, 2021 5:34:30 AM
Project name: testholuser20
holuser20 holuser20 added a new dataset named Statistics
Jun 21, 2021 5:34:30 AM
Project name: testholuser20
holuser20 holuser20 added new service Featurestore
Jun 21, 2021 5:34:30 AM
Project name: testholuser20
holuser20 holuser20 added a new dataset named testholuser20_featurestore.db
Jun 21, 2021 5:34:30 AM
Project name: testholuser20
holuser20 holuser20 added a storage connector for the featurestore with name: testholuser20_Training_Datasets
Jun 21, 2021 5:34:29 AM
Project name: testholuser20
holuser20 holuser20 added new service Serving
Jun 21, 2021 5:34:29 AM
Project name: testholuser20
holuser20 holuser20 added a new dataset named testholuser20_Training_Datasets
Jun 21, 2021 5:34:29 AM
56°F Partly cloudy
ESP
ES
5:34
21/06/2021

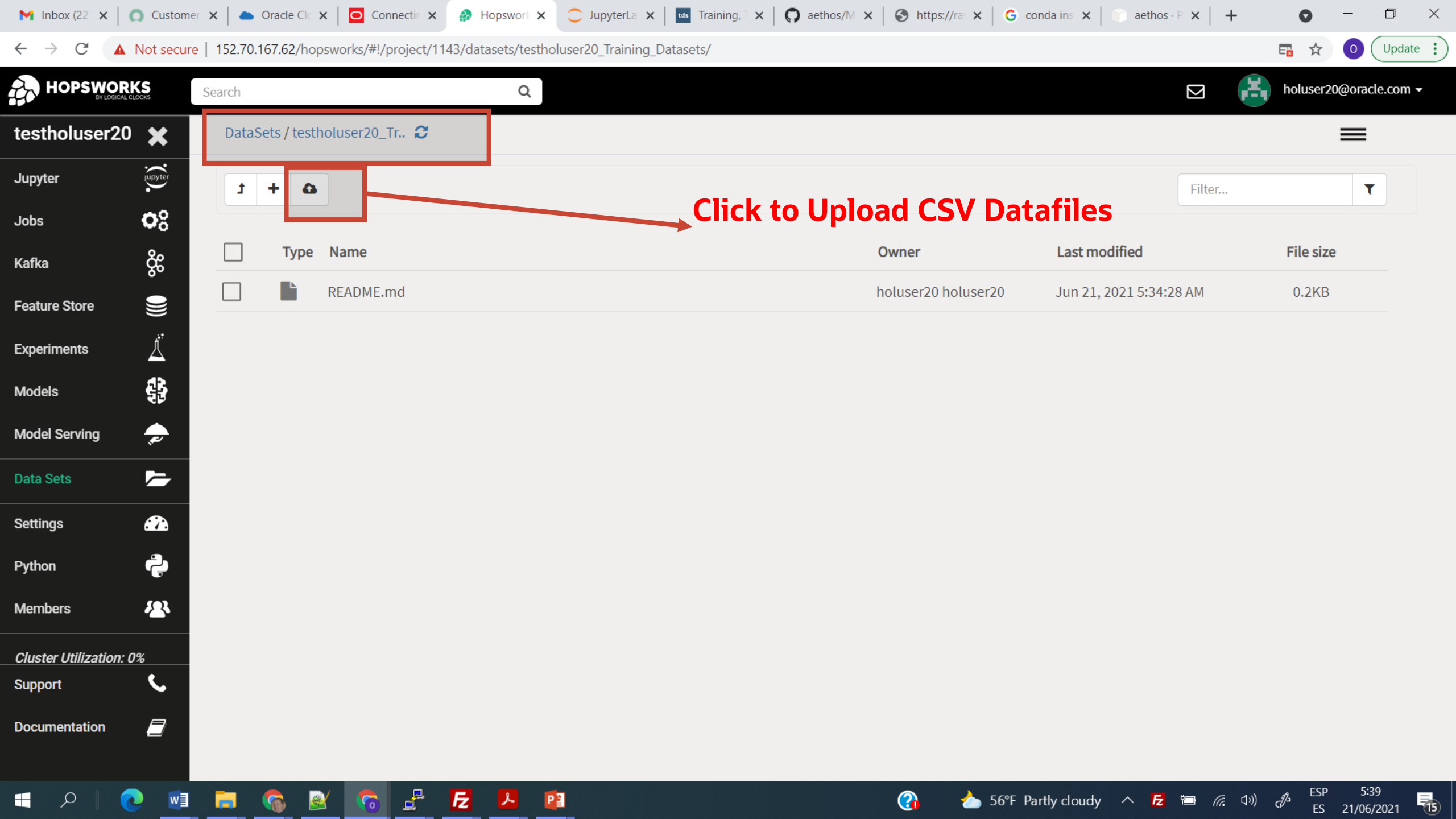

Not secure | 152.70.167.62/hopsworks/#!/project/1143/python
HOPSWORKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
Install
See hops python docs
Members
Find member...
Add member(s)
You can invite other Users to your Project in order to modify and implement some algorithms
Members to be added
Select user...
No members added yet...
Save
Members
Role
Action
holuser20 holuser20 (me)
Data owner
56°F Partly cloudy
ESP
ES
5:36
21/06/2021

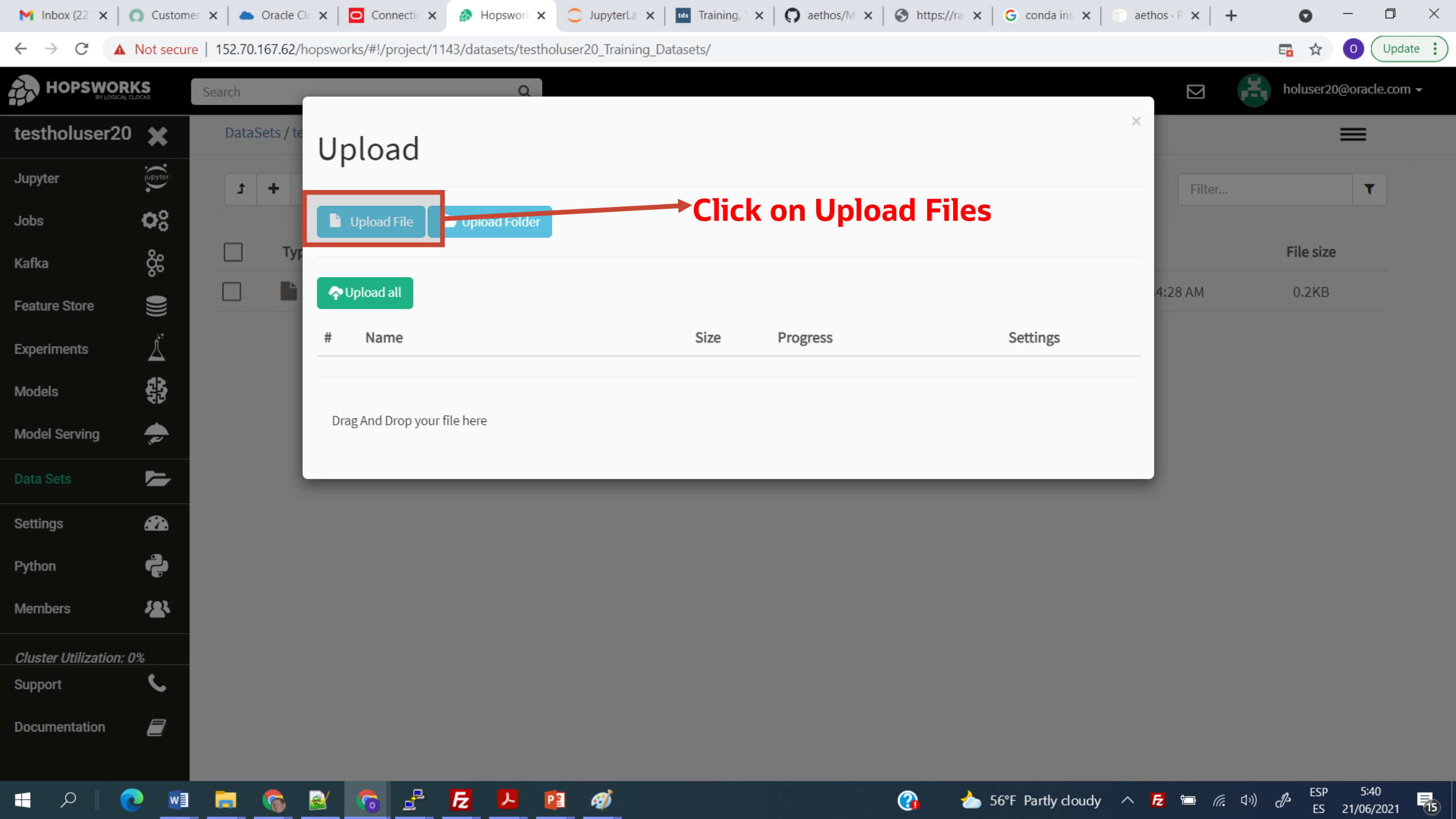

Not secure | 152.70.167.62/hopsworks/#!/project/1143/settings
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
Project Settings
PIA
Versions
Provenance
Project Settings
You can enable Airflow
Name
testholuser20
ID
1143
Description
Data Retention Period
2031-06-21T00:00:00Z
Quotas
Datasets
unlimited of unlimited used
-1 files
Hive Database
unlimited of unlimited used
-1 files
Feature Store
unlimited of unlimited used
-1 files
CPU
16666.67 CPU hours
Services
AIRFLOW
Export Certificates
Save
Danger Zone
56°F Partly cloudy
ESP
ES
5:35
21/06/2021


Not secure | 152.70.167.62/hopsworks/#!/project/1143/python
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
Install
Manage Environment
Ongoing Operations
See hops python docs
Select package location
PyPi
Conda
Upload
Git
You can upload and install libraries
Install Python libraries using pip in Anaconda environment
Python Version is 3.7
Library name ...
(optional) version
Install
Search here ....
Search
56°F Partly cloudy
ESP
ES
5:36
21/06/2021

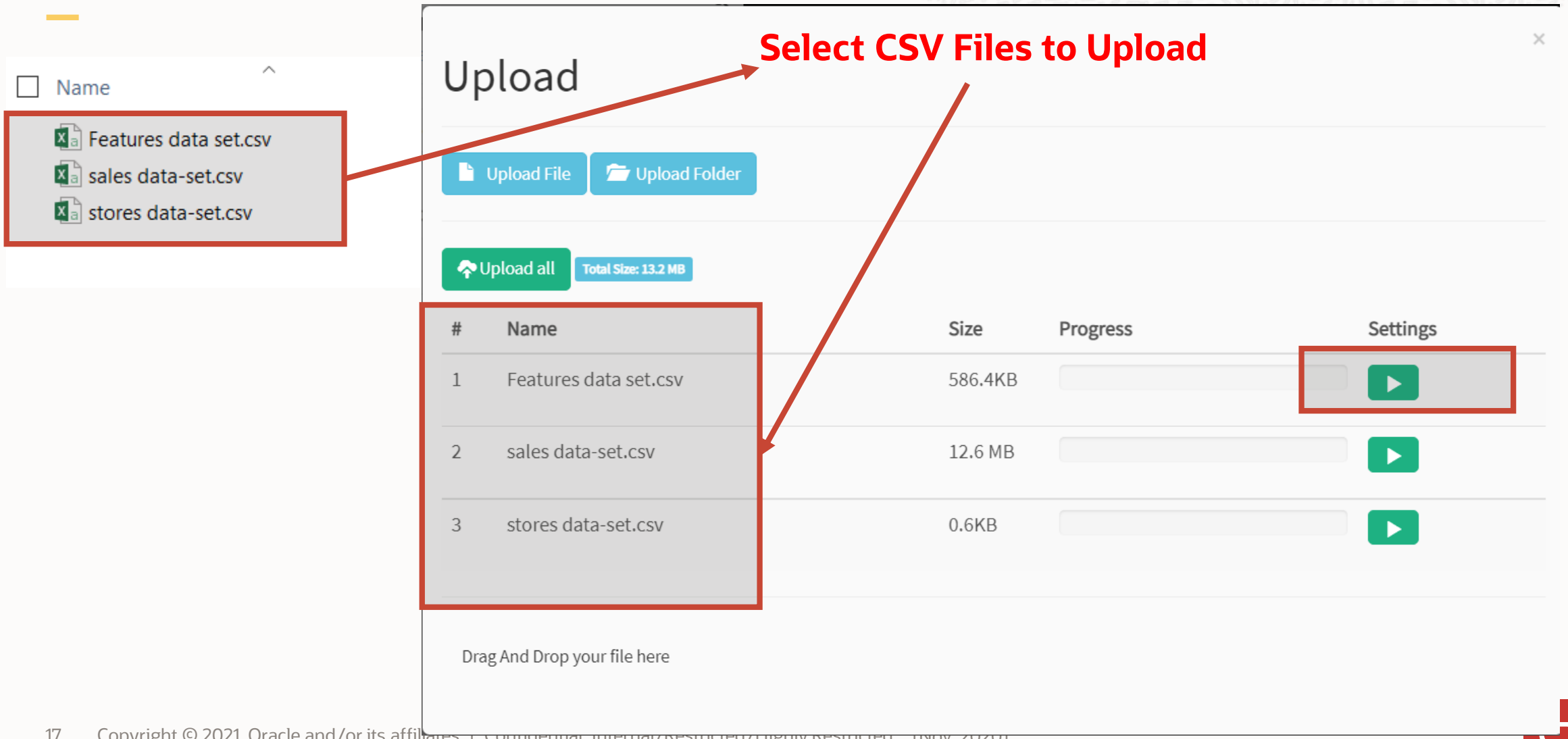

Not secure | 152.70.167.62/hopsworks/#!/project/1143/datasets
HOPSWORKS
Search
holuser20@oracle.com
testholuser20
Filter: All Shared Public in Cluster Exclusive Pending
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
+ CREATE DATASET
Filter...
Name
Owner
Last modified
Hive.db holuser20 holuser20 Jun 21, 2021 5:34:28 AM
Featurestore.db holuser20 holuser20 Jun 21, 2021 5:34:29 AM
Training Datasets holuser20 holuser20 Jun 21, 2021 5:34:28 AM
Logs holuser20 holuser20 Jun 21, 2021 5:34:20 AM
Resources holuser20 holuser20 Jun 21, 2021 5:34:21 AM
Experiments holuser20 holuser20 Jun 21, 2021 5:34:21 AM
Jupyter holuser20 holuser20 Jun 21, 2021 5:34:24 AM
Models holuser20 holuser20 Jun 21, 2021 5:34:28 AM
DataValidation holuser20 holuser20 Jun 21, 2021 5:34:29 AM
Statistics holuser20 holuser20 Jun 21, 2021 5:34:29 AM
56°F Partly cloudy
ESP ES
5:35 21/06/2021

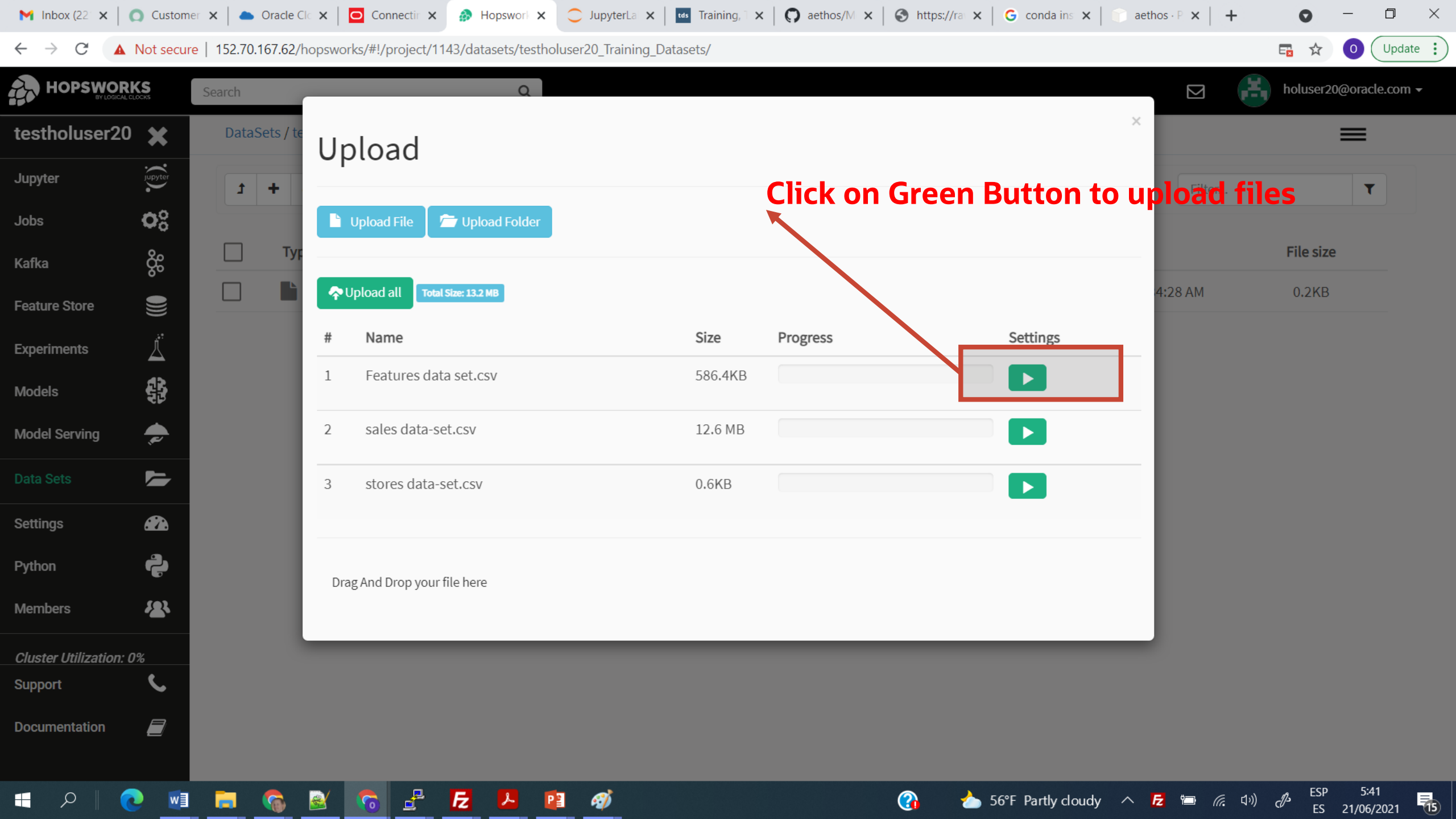

Not secure | 152.70.167.62/hopsworks/#!/project/1143/datasets
HOPSWORKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
Filter: All Shared Public in Cluster Exclusive Pending
+ CREATE DATASET
Filter...
Select Training Datasets to upload
Name
Owner
Last modified
Hive.db holuser20 holuser20 Jun 21, 2021 5:34:28 AM
Featurestore.db holuser20 holuser20 Jun 21, 2021 5:34:29 AM
Training Datasets holuser20 holuser20 Jun 21, 2021 5:34:28 AM
Logs holuser20 holuser20 Jun 21, 2021 5:34:20 AM
Resources holuser20 holuser20 Jun 21, 2021 5:34:21 AM
Experiments holuser20 holuser20 Jun 21, 2021 5:34:21 AM
Jupyter holuser20 holuser20 Jun 21, 2021 5:34:24 AM
Models holuser20 holuser20 Jun 21, 2021 5:34:28 AM
DataValidation holuser20 holuser20 Jun 21, 2021 5:34:29 AM
Statistics holuser20 holuser20 Jun 21, 2021 5:34:29 AM
56°F Partly cloudy
ESP ES
5:38 21/06/2021

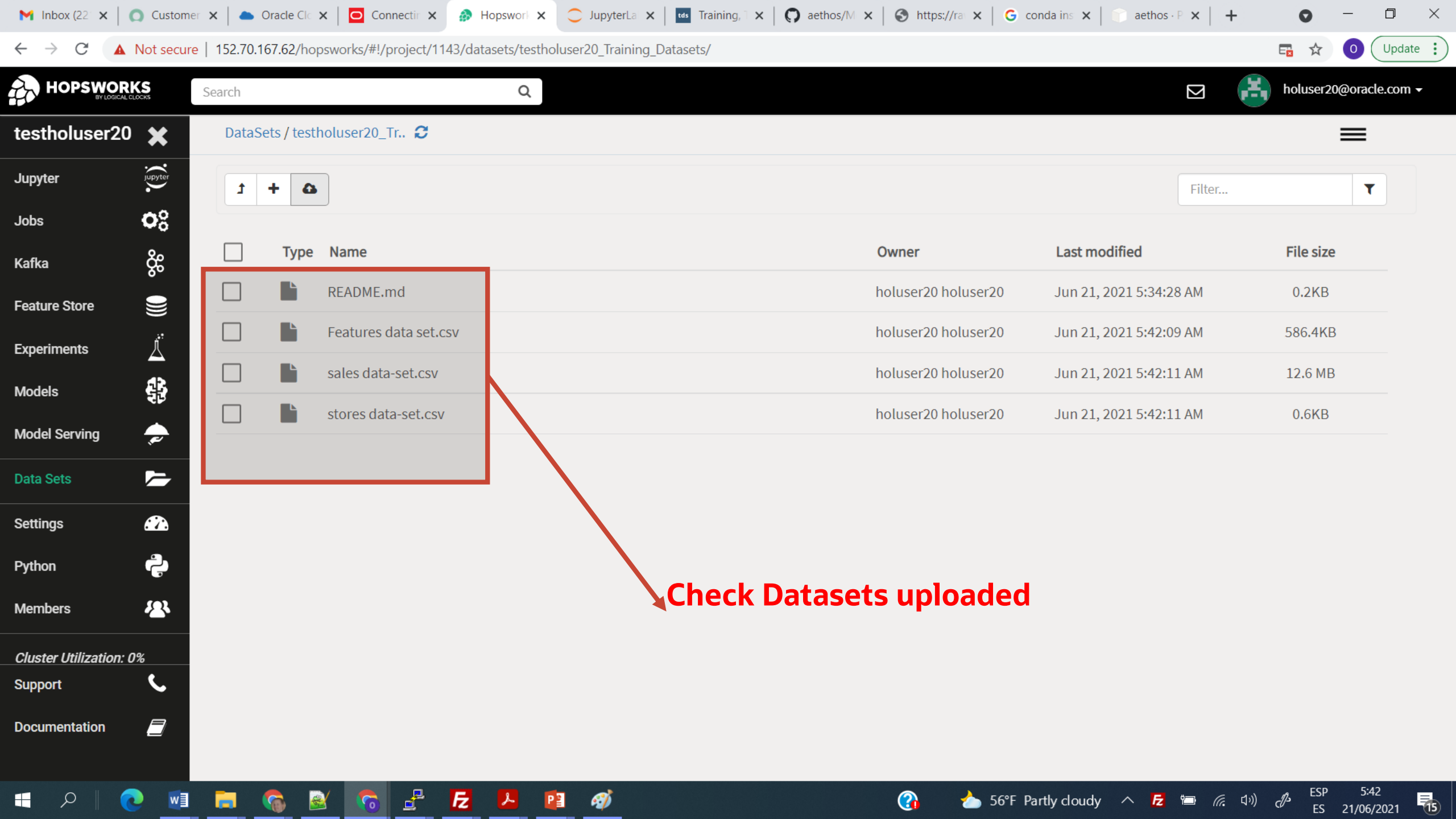

Not secure | 152.70.167.62/hopsworks/#!/project/1143/datasets/testholuser20_Training_Datasets/
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
testholuser20
DataSets / testholuser20_Tr..
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
Filter...
Click to Upload CSV Datafiles
Type
Name
Owner
Last modified
File size
README.md
holuser20 holuser20
Jun 21, 2021 5:34:28 AM
0.2KB
56°F Partly cloudy
ESP
ES
5:39
21/06/2021

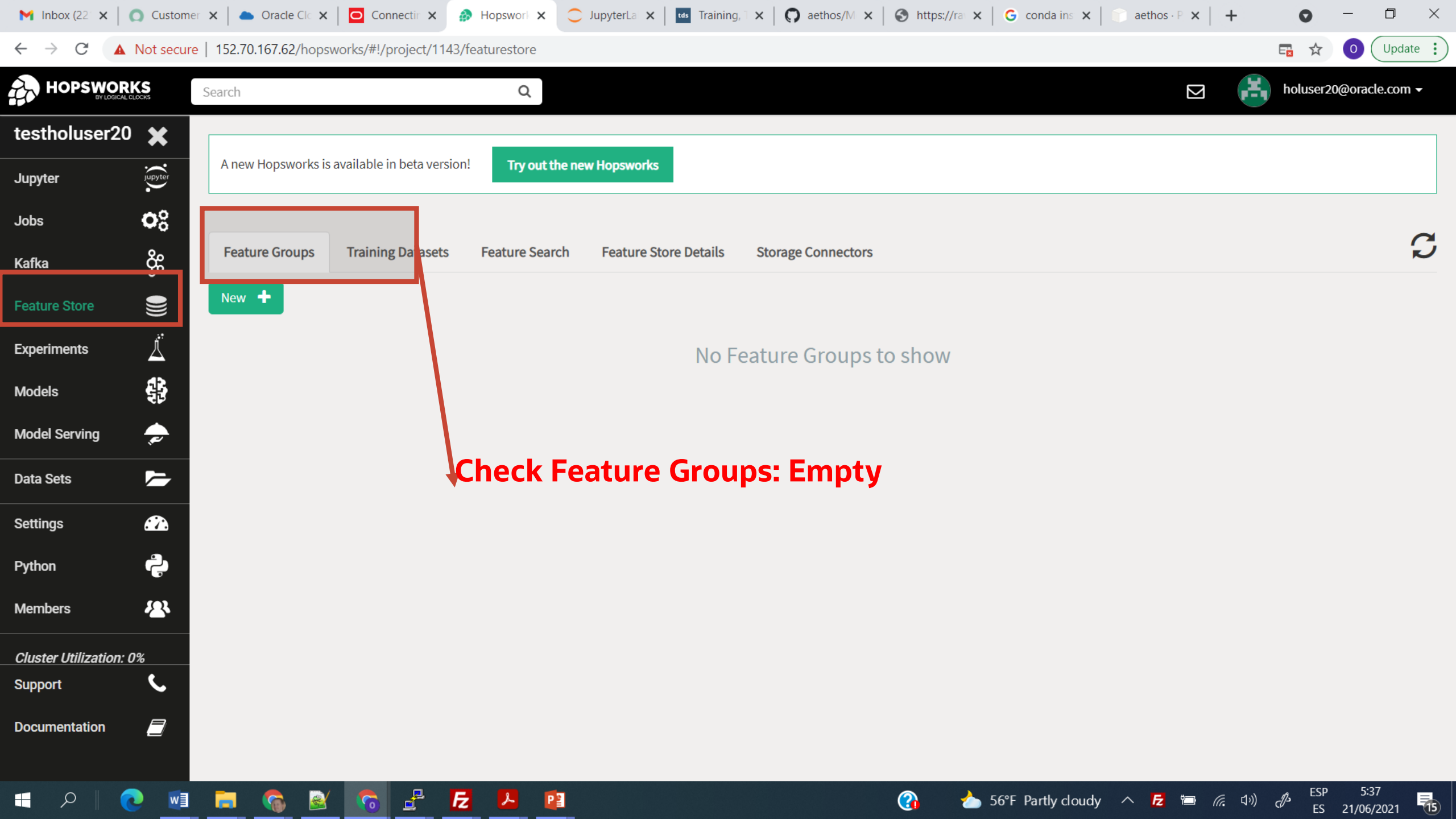

Not secure | 152.70.167.62/hopsworks/#!/project/1143/datasets/testholuser20_Training_Datasets/
HOPSWORKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
DataSets / te
Filter...
File size
4:28 AM
0.2KB
Upload
Upload File
Upload Folder
Click on Upload Files
Upload all
Name Size Progress Settings
Drag And Drop your file here
56°F Partly cloudy
ESP
ES
5:40
21/06/2021

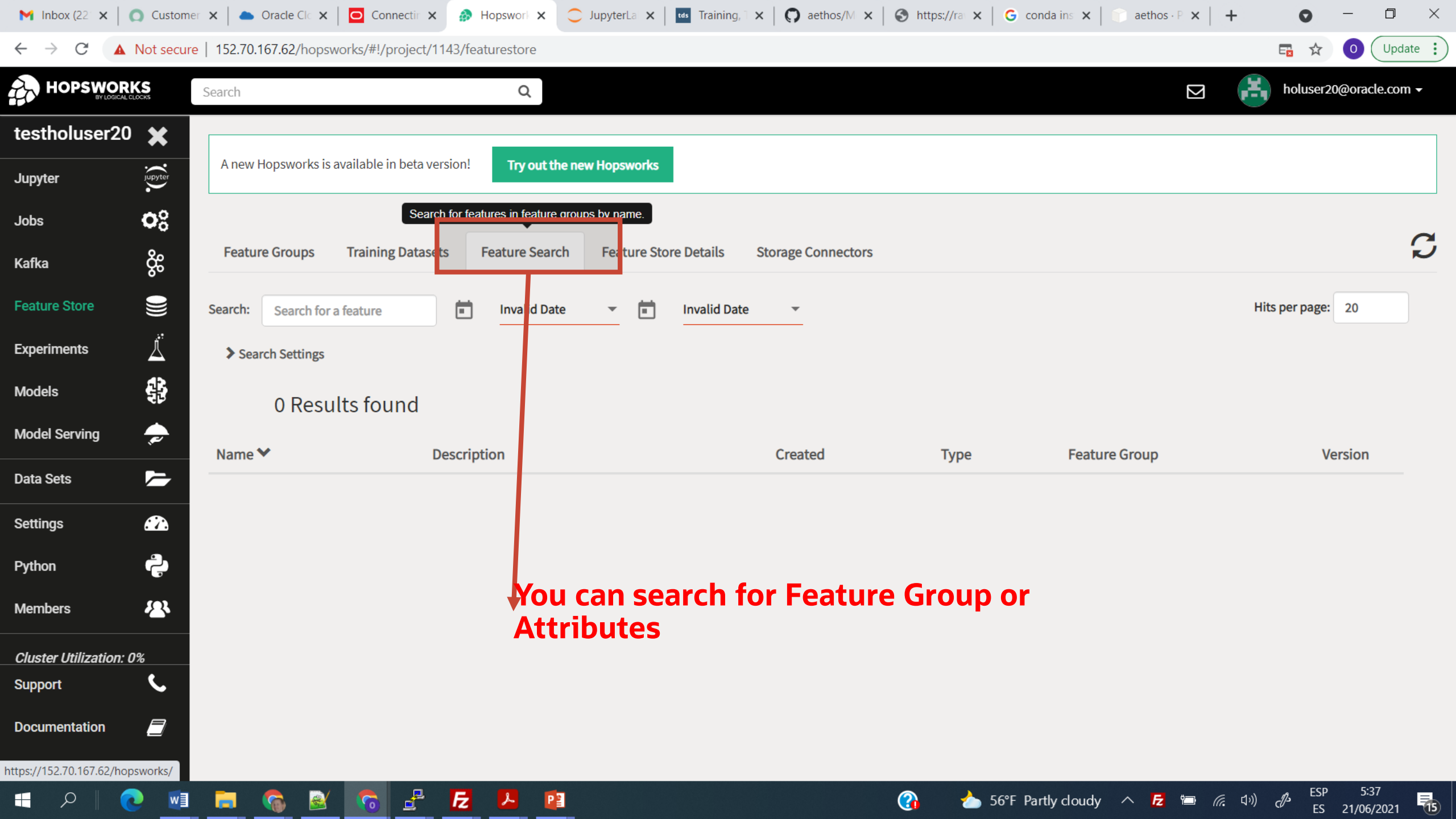
Download all files from Github Projects to Upload to Feature DataStore




Not secure | 152.70.167.62/hopsworks/#!/project/1143/datasets/testholuser20_Training_Datasets/
HOPSWORKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
DataSets / te
File size
4:28 AM
0.2KB
Upload
Upload File
Upload Folder
Upload all
Total Size: 13.2 MB
Name Size Progress Settings
1 Features data set.csv 586.4KB
2 sales data-set.csv 12.6 MB
3 stores data-set.csv 0.6KB
Drag And Drop your file here
Click on Green Button to upload files
56°F Partly cloudy
ESP ES
5:41 21/06/2021

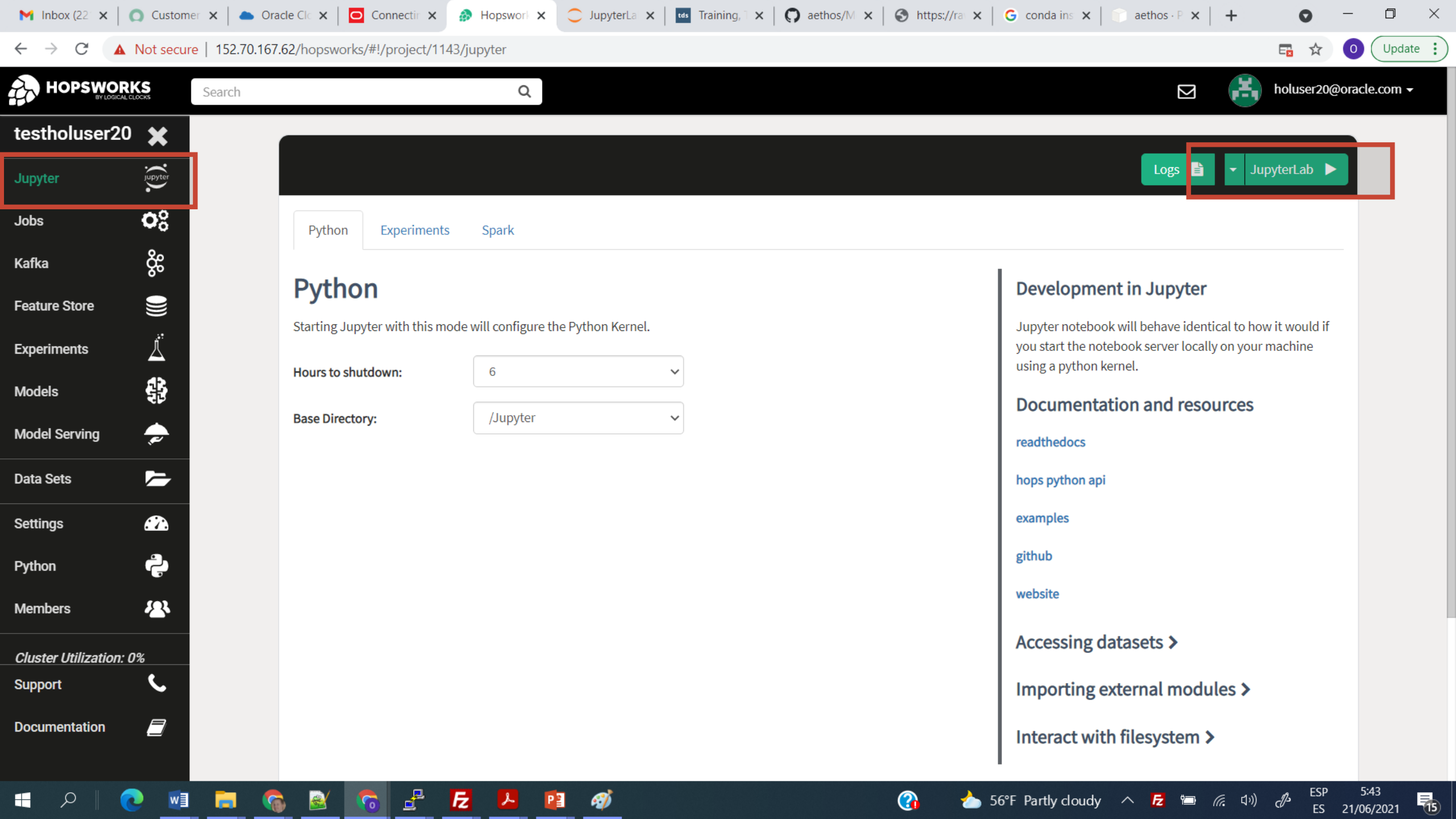

Not secure | 152.70.167.62/hopsworks/#!/project/1143/datasets/testholuser20_Training_Datasets/
HOPSWORKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
DataSets / testholuser20_Tr..
Filter...
Type
Name
Owner
Last modified
File size
README.md
holuser20 holuser20
Jun 21, 2021 5:34:28 AM
0.2KB
Features data set.csv
holuser20 holuser20
Jun 21, 2021 5:42:09 AM
586.4KB
sales data-set.csv
holuser20 holuser20
Jun 21, 2021 5:42:11 AM
12.6 MB
stores data-set.csv
holuser20 holuser20
Jun 21, 2021 5:42:11 AM
0.6KB
Check Datasets uploaded
56°F Partly cloudy
ESP
ES
5:42
21/06/2021

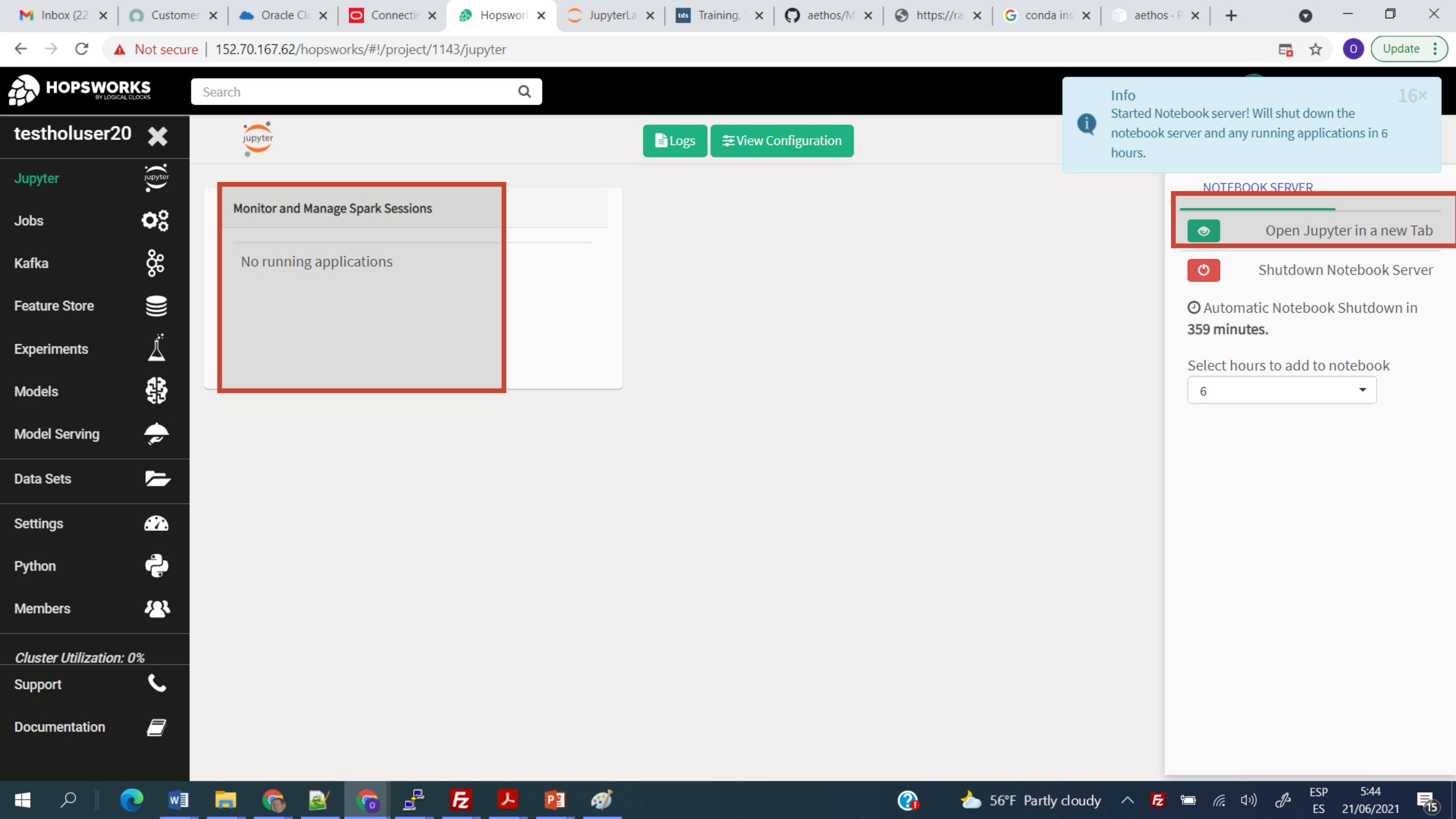

Not secure | 152.70.167.62/hopsworks/#!/project/1143/featurestore
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
A new Hopsworks is available in beta version!
Try out the new Hopsworks
Feature Groups
Training Datasets
Feature Search
Feature Store Details
Storage Connectors
New
No Feature Groups to show
Check Feature Groups: Empty
56°F Partly cloudy
ESP
ES
5:37
21/06/2021

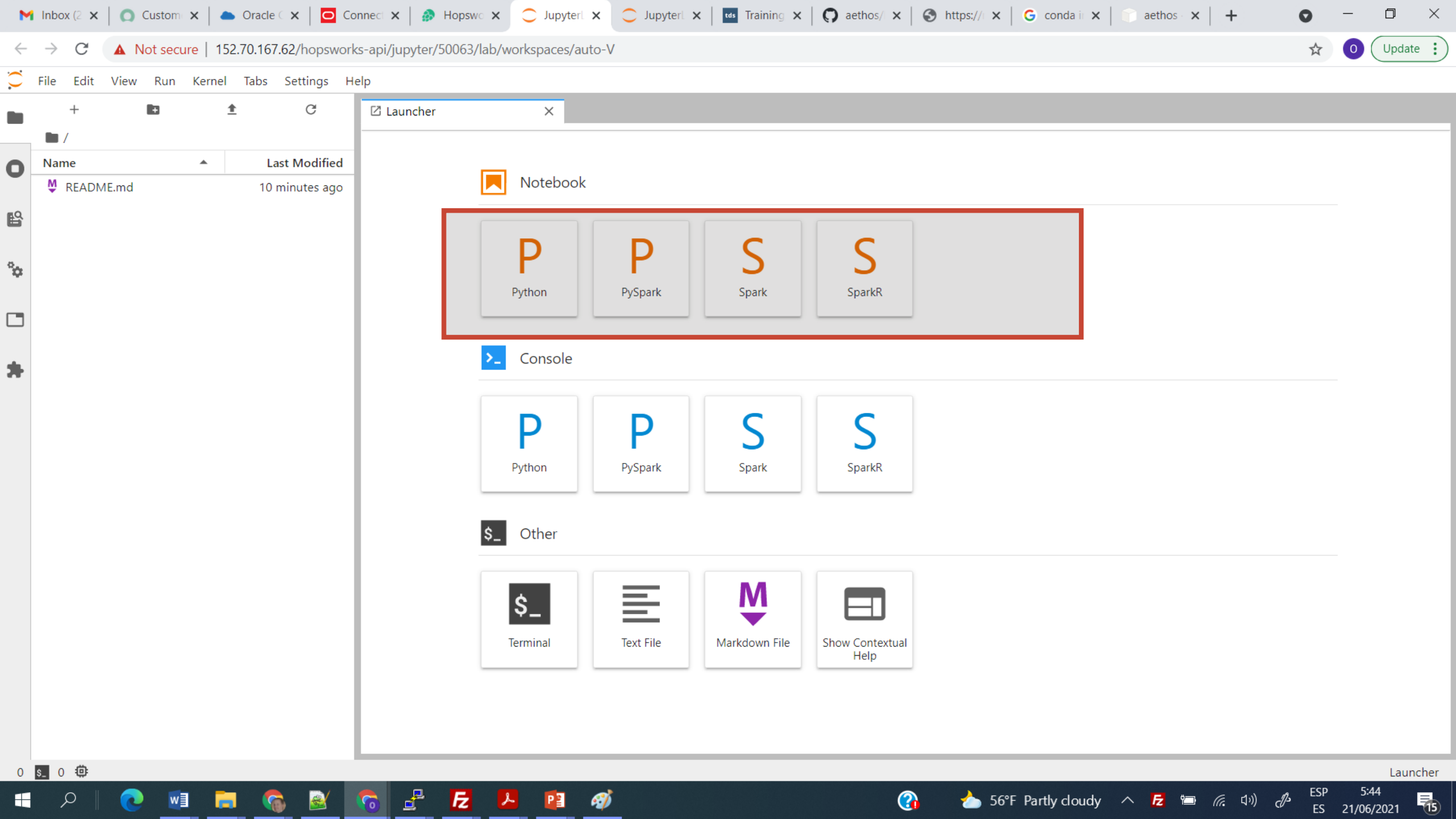

Not secure | 152.70.167.62/hopsworks/#!/project/1143/featurestore
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
A new Hopsworks is available in beta version!
Try out the new Hopsworks
Search for features in feature groups by name.
Feature Groups
Training Datasets
Feature Search
Feature Store Details
Storage Connectors
Search:
Search for a feature
Invalid Date
Invalid Date
Hits per page:
20
Search Settings
0 Results found
Name
Description
Created
Type
Feature Group
Version
You can search for Feature Group or Attributes
https://152.70.167.62/hopsworks/
56°F Partly cloudy
ESP
ES
5:37
21/06/2021

STEP2: Execute Labs.

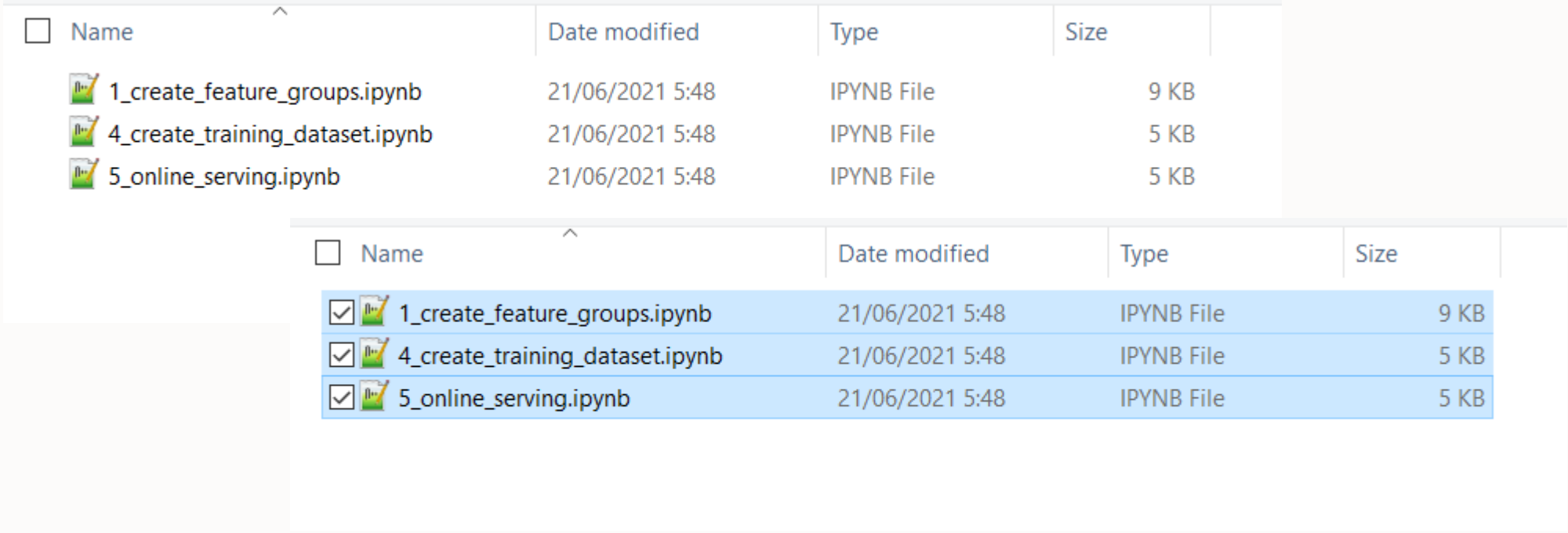
Lab 1: How to create Feature Group and Features Attributes.


Not secure | 152.70.167.62/hopsworks/#!/project/1143/jupyter
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
Logs
JupyterLab
Python
Experiments
Spark
Python
Starting Jupyter with this mode will configure the Python Kernel.
Hours to shutdown:
6
Base Directory:
/Jupyter
Development in Jupyter
Jupyter notebook will behave identical to how it would if you start the notebook server locally on your machine using a python kernel.
Documentation and resources
readthedocs
hops python api
examples
github
website
Accessing datasets >
Importing external modules >
Interact with filesystem >
56°F Partly cloudy
ESP
ES
5:43
21/06/2021


Not secure | 152.70.167.62/hopsworks/#!/project/1143/jupyter
Update
HOPSWORKS
BY LOGICAL CLOCKS
Search
testholuser20
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Cluster Utilization: 0%
Support
Documentation
Logs
View Configuration
Info
Started Notebook server! Will shut down the notebook server and any running applications in 6 hours.
Monitor and Manage Spark Sessions
No running applications
NOTEBOOK SERVER
Open Jupyter in a new Tab
Shutdown Notebook Server
Automatic Notebook Shutdown in 359 minutes.
Select hours to add to notebook
6
56°F Partly cloudy
ESP
ES
5:44
21/06/2021


Not secure | 152.70.167.62/hopsworks-api/jupyter/50063/lab/workspaces/auto-V
Update
File Edit View Run Kernel Tabs Settings Help
Launcher
/
Name
Last Modified
README.md
10 minutes ago
Notebook
Python
PySpark
Spark
SparkR
Console
Python
PySpark
Spark
SparkR
Other
Terminal
Text File
Markdown File
Show Contextual Help
Launcher
56°F Partly cloudy
ESP ES
5:44
21/06/2021

Download all notebooks from Github Projects to Feature DataStore JupyterLab



Not secure | 152.70.167.62/hopsworks-api/jupyter/50063/lab/workspaces/auto-V

File Edit View Run Kernel Tabs Settings Help

/

Name	Last Modified
1_create_feature_groups.ip...	seconds ago
4_create_training_dataset.ip...	seconds ago
5_online_serving.ipynb	seconds ago
README.md	17 minutes ago

Launcher | 1_create_feature_groups.ipyr

Markdown | PySpark

---

title: "Create empty feature groups for Online Feature Store" date: 2021-04-25 type: technical_note draft: false

---

[1]:
```
import json
from pyspark.sql.types import StructField, StructType, StringType, DoubleType, TimestampType, LongType, IntegerType
```

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log
1	application_1619309085643_0002	pyspark	idle	Link	Link

```
SparkSession available as 'spark'.
```

Create empty feature groups

In this demo example we are expecting to recieve data from Kafka topic, read using spark streaming, do streamig aggregations and ingest aggregated data to feature groups. Thus we will create empy feature groups where we will ingest streaming data.

Define schema for feature groups

[2]:
```
card_schema = StructType([StructField('tid', StringType(), True),
                          StructField('datetime', StringType(), True),
                          StructField('cc_num', LongType(), True),
                          StructField('amount', DoubleType(), True)])

schema_10m = StructType([StructField('cc_num', LongType(), True),
                         StructField('num_trans_per_10m', LongType(), True),
                         StructField('avg_amt_per_10m', DoubleType(), True),
                         StructField('stdev_amt_per_10m', DoubleType(), True)])

schema_1h = StructType([StructField('cc_num', LongType(), True),
                        StructField('num_trans_per_1h', LongType(), True)
```

0 1 PySpark | Connecting

Mode: Command Ln 1, Col 1 1_create_feature_groups.ipynb

56°F Partly cloudy ESP ES 5:51 21/06/2021

testholuser20

Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members

Cluster Utilization: 16%

Support
Documentation

DataSets / testholuser20_Tr..

Filter...

Type	Name	Owner	Last modified	File size
	README.md	holuser20 holuser20	Jun 21, 2021 5:34:28 AM	0.2KB
	Features data set.csv	holuser20 holuser20	Jun 21, 2021 5:42:09 AM	586.4KB
	sales data-set.csv	holuser20 holuser20	Jun 21, 2021 5:42:11 AM	12.6 MB
	stores data-set.csv	holuser20 holuser20	Jun 21, 2021 5:42:11 AM	0.6KB

testholuser20

Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members

Cluster Utilization: 11%

Support
Documentation

DataSets / testholuser20_Tr..

hdfs:///Projects/testholuse

Filter...

	Type	Name	Owner	Last modified	File size
☐		README.md	holuser20 holuser20	Jun 21, 2021 5:34:28 AM	0.2KB
☐		Features data set.csv	holuser20 holuser20	Jun 21, 2021 5:42:09 AM	586.4KB
☐		sales data-set.csv	holuser20 holuser20	Jun 21, 2021 5:42:11 AM	12.6 MB
☑		stores data-set.csv	holuser20 holuser20	Jun 21, 2021 5:42:11 AM	0.6KB

DETAILS

Size	0.6KB
Type	File
Last changed	Jun 21, 2021 5:42:11 AM
Owner	holuser20 holuser20
Permission	rwxrwx---

testholuser20

Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members

Cluster Utilization: 11%

Support
Documentation

hdfs:///Projects/testholuse

Copy hdfs path

Filter...

	Type	Name	Owner	Last modified	File size
☐		README.md	holuser20 holuser20	Jun 21, 2021 5:34:28 AM	0.2KB
☐		Features data set.csv	holuser20 holuser20	Jun 21, 2021 5:42:09 AM	586.4KB
☐		sales data-set.csv	holuser20 holuser20	Jun 21, 2021 5:42:11 AM	12.6 MB
☑		stores data-set.csv	holuser20 holuser20	Jun 21, 2021 5:42:11 AM	0.6KB

DETAILS

Size	0.6KB
Type	File
Last changed	Jun 21, 2021 5:42:11 AM
Owner	holuser20 holuser20
Permission	rwxrwx---

Inbox (2 | Custom | Oracle | Connec | Hopswo | Jupyter | Jupyter | Training | aethos/ | https:// | conda i | aethos

Not secure | 152.70.167.62/hopsworks-api/jupyter/50063/lab/workspaces/auto-V

Update

File Edit View Run Kernel Tabs Settings Help

/

Name	Last Modified
1_create_feature_groups.ip...	3 minutes ago
4_create_training_dataset.ip...	5 minutes ago
5_online_serving.ipynb	5 minutes ago
README.md	22 minutes ago
Untitled.ipynb	a minute ago

Untitled.ipynb | 1_create_feature_groups.ipyr

Code PySpark

Ingestion to Online Feature Store

[1]:
```
import hsfs
connection = hsfs.connection()
fs = connection.get_feature_store()
```

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log
10	application_1624183360724_0002	pyspark	idle	Link	Link

```
SparkSession available as 'spark'.
Connected. Call `.close()` to terminate connection gracefully.
```

[9]:
```
from hops import hdfs
from pyspark.sql import functions as F

# hdfs:///Projects/testholuser20/testholuser20_Training_Datasets/stores data-set.csv
stores_csv = spark.read\
             .option("inferSchema", "true")\
             .option("header", "true")\
             .format("csv")\
             .load("hdfs:///Projects/{}/testholuser20_Training_Datasets/stores data-set.csv".format(hdfs.project_name()))
```

[ ]:
```
online_store_fg_meta = fs.create_feature_group(name="online_store_fg",
                                               version=1,
                                               primary_key=['store'],
                                               description="Store related features",
                                               online_enabled=True,
                                               time_travel_format=None,
                                               statistics_config={"enabled": True, "histograms": True, "correlations": True})
```

[ ]:
```
online_store_fg_meta.save(stores_csv)
```

0 4 PySpark | Idle Saving completed Mode: Edit Ln 9, Col 130 Untitled.ipynb

55°F Mostly clear ESP ES 5:56 21/06/2021

Not secure | 152.70.167.62/hopsworks-api/jupyter/50063/lab/workspaces/auto-V

Update

File Edit View Run Kernel Tabs Settings Help

1_create_feature_store.ipynb | 1_create_feature_groups.ipyr

Code | PySpark

/

Name	Last Modified
1_create_feature_groups.ip...	25 minutes ago
1_create_feature_store.ipynb	seconds ago
5_online_serving.ip	
README.md	

Name: 1_create_feature_store.ipynb
Created: 2021-06-21 06:18:16
Modified: 2021-06-21 06:18:16
Writable: true
Kernel: PySpark

10 application_1624183360724_0002 pyspark idle Link Link

```
SparkSession available as 'spark'.
Connected. Call `.close()` to terminate connection gracefully.
```

[9]:
```
from hops import hdfs
from pyspark.sql import functions as F

# hdfs:///Projects/testholuser20/testholuser20_Training_Datasets/stores data-set.csv
stores_csv = spark.read\
             .option("inferSchema", "true")\
             .option("header", "true")\
             .format("csv")\
             .load("hdfs:///Projects/{}/testholuser20_Training_Datasets/stores data-set.csv".format(hdfs.project_name()))
```

[10]:
```
online_store_fg_meta = fs.create_feature_group(name="online_store_fg",
                                               version=1,
                                               primary_key=['store'],
                                               description="Store related features",
                                               online_enabled=True,
                                               time_travel_format=None,
                                               statistics_config={"enabled": True, "histograms": True, "correlations": True})
```

[11]:
```
online_store_fg_meta.save(stores_csv)
```

```
<hsfs.feature_group.FeatureGroup object at 0x7f3f595d84d0>
```

[ ]:

0 4 PySpark | Idle Saving completed Mode: Command Ln 1, Col 1 1_create_feature_store.ipynb

stores data-set.csv

Show all

55°F Mostly clear ESP ES 6:18 21/06/2021

testholuser20

Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Support
Documentation

A new Hopsworks is available in beta version! Try out the new Hopsworks

Feature Groups | Training Datasets | Feature Search | Feature Store Details | Storage Connectors

New +

Search: 6/20/2021 6/22/2021 Hits per page: 20

Name	Description	Created	Type	Online	Version
online_store_fg	Store related features	Jun 21, 2021 6:16:52 AM	CACHED	Yes	1

testholuser20

Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Support
Documentation

A new Hopsworks is available in beta version! Try out the new Hopsworks

Feature Groups | Training Datasets | Feature Search | Feature Store Details | Storage Connectors

Search: Search for a feature 6/20/2021 6/22/2021 Hits per page: 20

Search Settings

3 Results found

Name	Description	Created	Type	Feature Group	Version
size		Jun 21, 2021 6:16:52 AM	int	online_store_fg	1
store		Jun 21, 2021 6:16:52 AM	int	online_store_fg	1
type		Jun 21, 2021 6:16:52 AM	string	online_store_fg	1

testholuser20

Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Support
Documentation

A new Hopsworks is available in beta version! **Try out the new Hopsworks**

View Details of the Feature Store

Feature Groups | Training Datasets | Feature Search | Feature Store Details | Storage Connectors

FEATURES
3

FEATURE GROUPS
1

TRAINING DATASETS
0

Cross-Project Features

Feature Stores can be shared across projects. Select a feature store below.

Number of available feature stores: 1

testholuser20_featurestore

Details

Id: 1091
Offline Featurestore Name: testholuser20_featurestore
Offline Featurestore HDFS Path: hopsfs://namenode.service.consul:8020/apps/hive/warehouse/testholuser20_featurestore.db
Hive Endpoint: 10.0.0.216:9085
Offline Featurestore Size: 0.9KB
Online Featurestore Enabled: true
Online Featurestore Name: testholuser20
Online Featurestore Size: 0 MB
MySQL Server Endpoint: jdbc:mysql://10.0.0.216:3306/?useSSL=false&allowPublicKeyRetrieval=true

Quota
100%

unlimited of unlimited used (Offline Feature Store quota).

stores data-set.csv

Show all

55°F Mostly clear

ESP ES 6:20 21/06/2021

Lab 2: Credit Card Fraud Model

Developer Journey Map

Create Kafka Topic

Create Feature Group

Create Producer Card Transactions

Create Spark Stream Ingestion

Create Training Dataset

Model Training

Online Serving

Inference Analysis

0_create_kafka_topic.ipynb

1_create_feature_groups.ipynb

2_produce_transactions.ipynb

3_stream-ingestion.ipynb

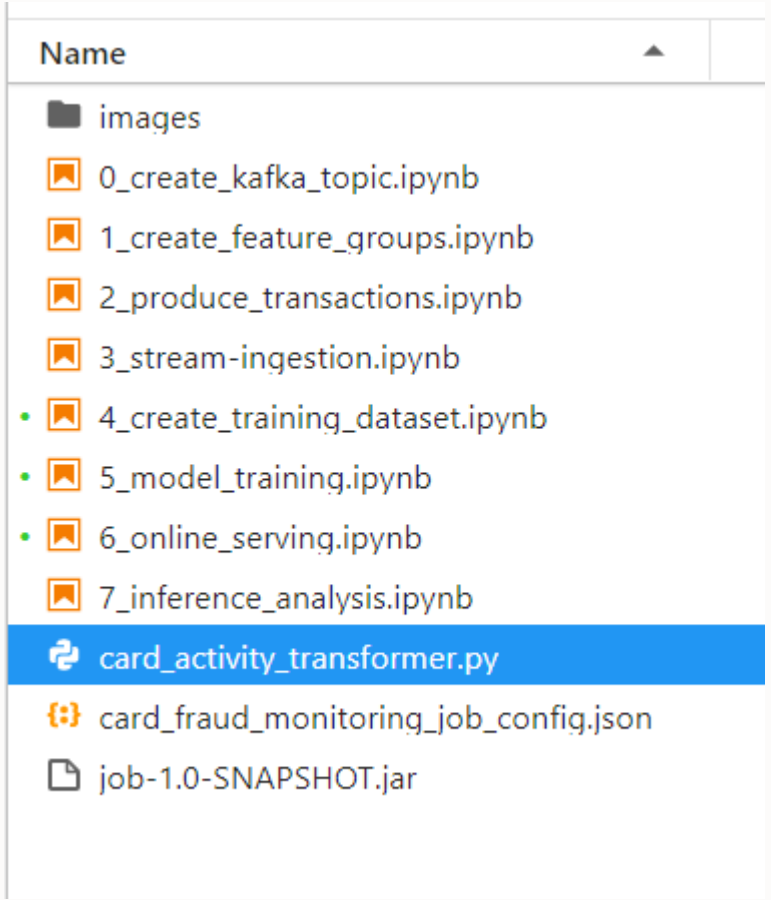
4_create_training_dataset.ipynb

5_model_trainng.ipynb

6_online_serving.ipynb

7_inference_analysis.ipynb

Upload all Notebooks to your project



Download the code from :
https://github.com/operard/featuredslab

Upload Code from Lab2.

0_create_kafka_topic: To store events

Create Kafka topics

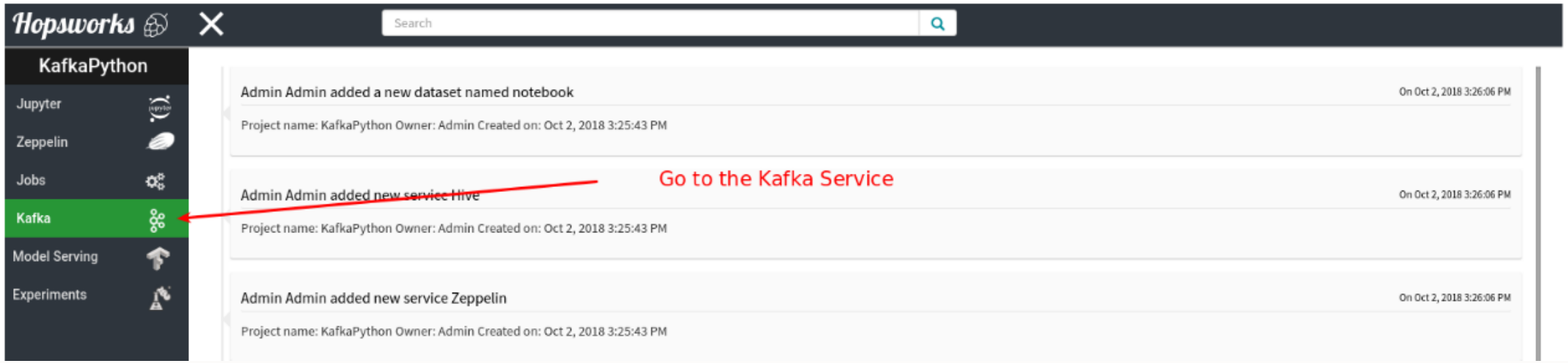
Create Kafka Topic: credit_card_transactions_<holuserXX>
Type: empty schema

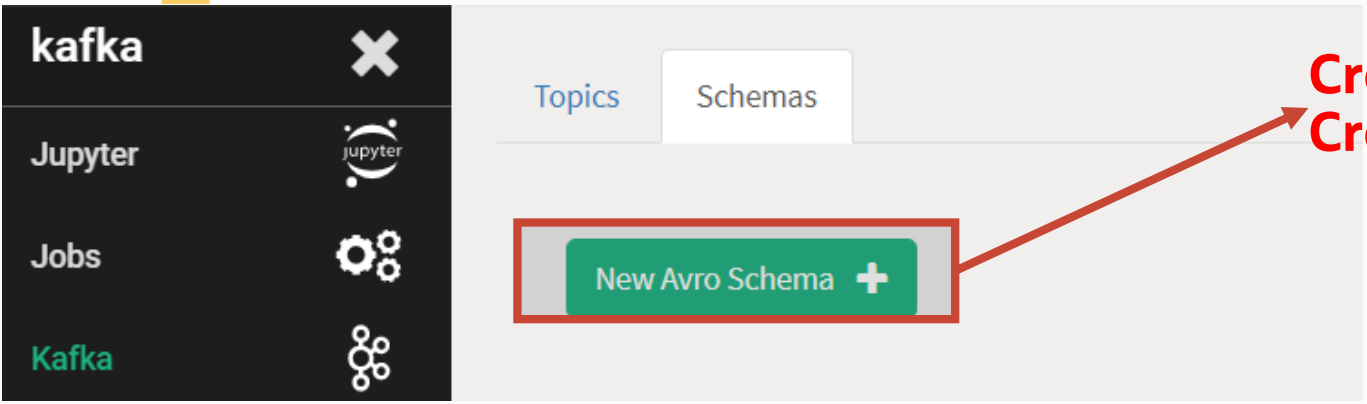
The first step for this demo is to create the following Kafka topics:

- **'credit_card_transactions'**: used in `2_produce-transactions.ipynb` and `3_stream-ingestion.ipynb` for logging the financial transactions and ingesting aggregations on this data into the Online Feature Store. To create this topic, first we need to create and **emtpy schema** `credit_card_transactions_schema`. An empty schema is represented with the value `[]`.
- **'credit_card_prediction_logs'**: used in `6_online-serving.ipynb` and `7_inference-analysis.ipynb` for storing the prediction logs. For this schema we can choose the built-in schema with name `inferenceschema`

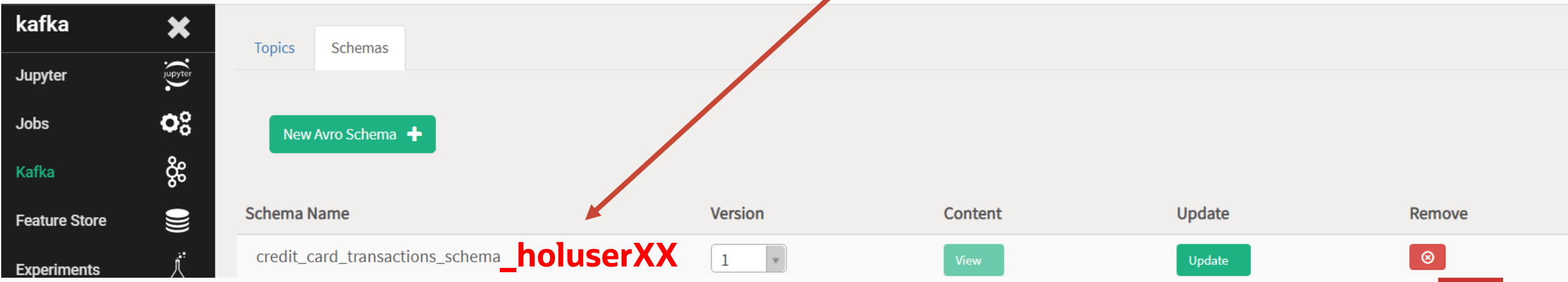
Create Kafka Topic: credit_card_prediction_logs_<holuserXX>
Type: inferenceschema

Step-by-step example

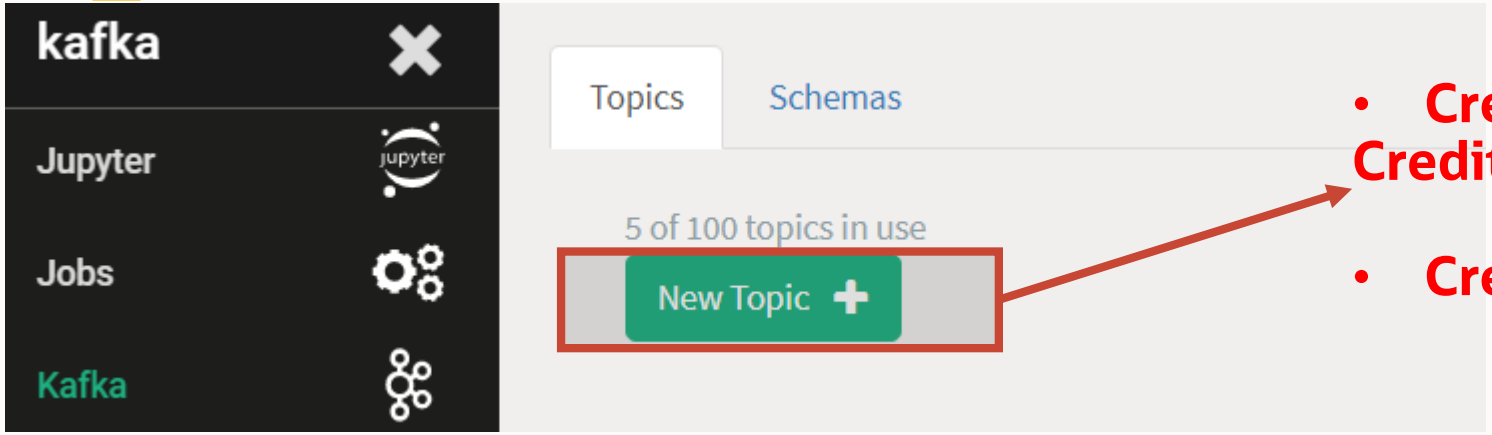




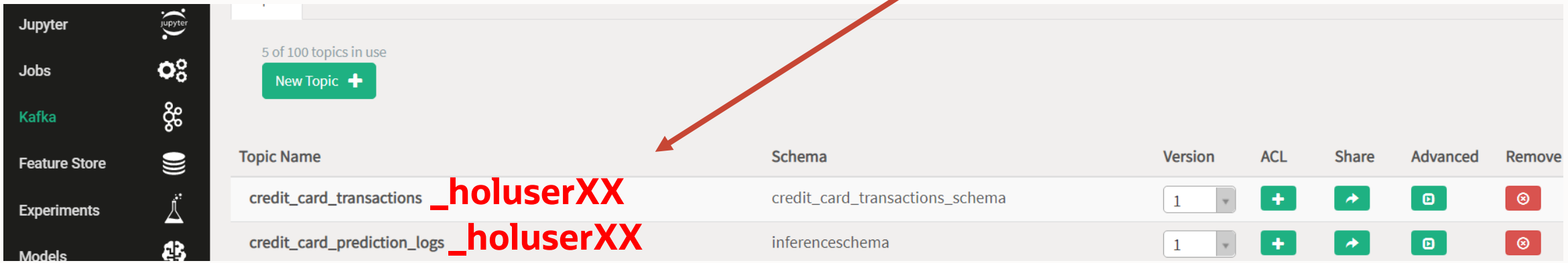
Create New Schema for Credit Card:
Credit_card_transactions_schema_holuserXX



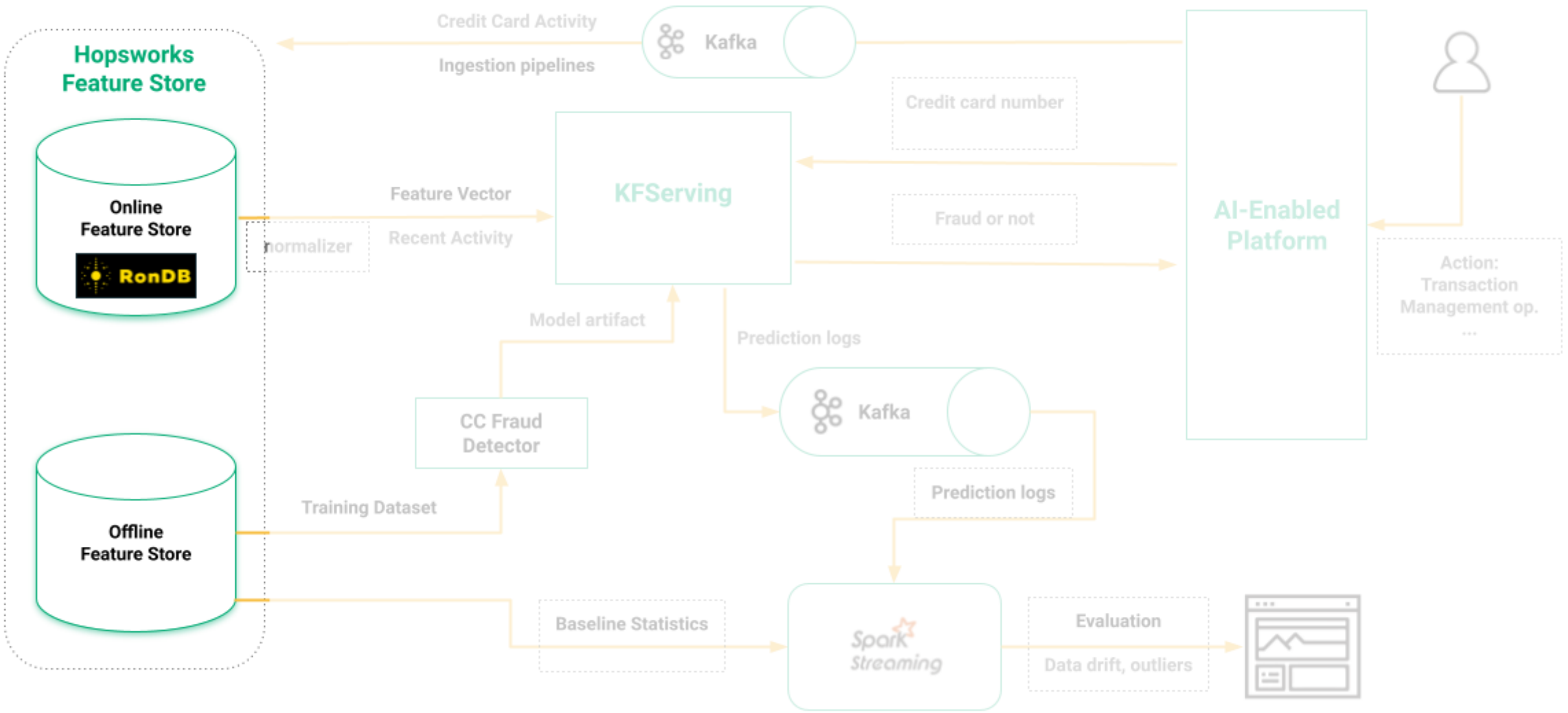
Topics Creation Results



- **Create New Topic for Credit Card: Credit_card_transactions_holuserXX**
- **Create Other topic for Predictions**



1_create_feature_groups



feature groups. Thus we will create empy feature groups where we will ingest streaming data.

Define schema for feature groups

```
[2]: card_schema = StructType([StructField('tid', StringType(), True),
                               StructField('datetime', StringType(), True),
                               StructField('cc_num', LongType(), True),
                               StructField('amount', DoubleType(), True)])

schema_10m = StructType([StructField('cc_num', LongType(), True),
                         StructField('num_trans_per_10m', LongType(), True),
                         StructField('avg_amt_per_10m', DoubleType(), True),
                         StructField('stdev_amt_per_10m', DoubleType(), True)])

schema_1h = StructType([StructField('cc_num', LongType(), True),
                        StructField('num_trans_per_1h', LongType(), True),
                        StructField('avg_amt_per_1h', DoubleType(), True),
                        StructField('stdev_amt_per_1h', DoubleType(), True)])

schema_12h = StructType([StructField('cc_num', LongType(), True),
                         StructField('num_trans_per_12h', LongType(), True),
                         StructField('avg_amt_per_12h', DoubleType(), True),
                         StructField('stdev_amt_per_12h', DoubleType(), True)])
```

Empty Schema for feature Group

Create empty spark dataframes

```
[3]: empty_card_df = sqlContext.createDataFrame(sc.emptyRDD(), card_schema)
empty_10m_agg_df = sqlContext.createDataFrame(sc.emptyRDD(), schema_10m)
empty_1h_agg_df = sqlContext.createDataFrame(sc.emptyRDD(), schema_1h)
```

```
card_transactions_10m_agg = fs.create_feature_group("card_transactions_10m_agg_n",
                                                    version = 1,
                                                    online_enabled=True,
                                                    statistics_config=False,
                                                    primary_key=["cc_num"])

card_transactions_10m_agg.save(empty_10m_agg_df)
```

```
<hsfs.feature_group.FeatureGroup object at 0x7f34624f8a50>
```

```
card_transactions_1h_agg = fs.create_feature_group("card_transactions_1h_agg_n",
                                                   version = 1,
                                                   online_enabled=True,
                                                   statistics_config=False,
                                                   primary_key=["cc_num"])

card_transactions_1h_agg.save(empty_1h_agg_df)
```

```
<hsfs.feature_group.FeatureGroup object at 0x7f3462538ad0>
```

```
card_transactions_12h_agg = fs.create_feature_group("card_transactions_12h_agg_n",
                                                    version = 1,
                                                    online_enabled=True,
                                                    statistics_config=False,
                                                    primary_key=["cc_num"])

card_transactions_12h_agg.save(empty_12h_agg_df)
```

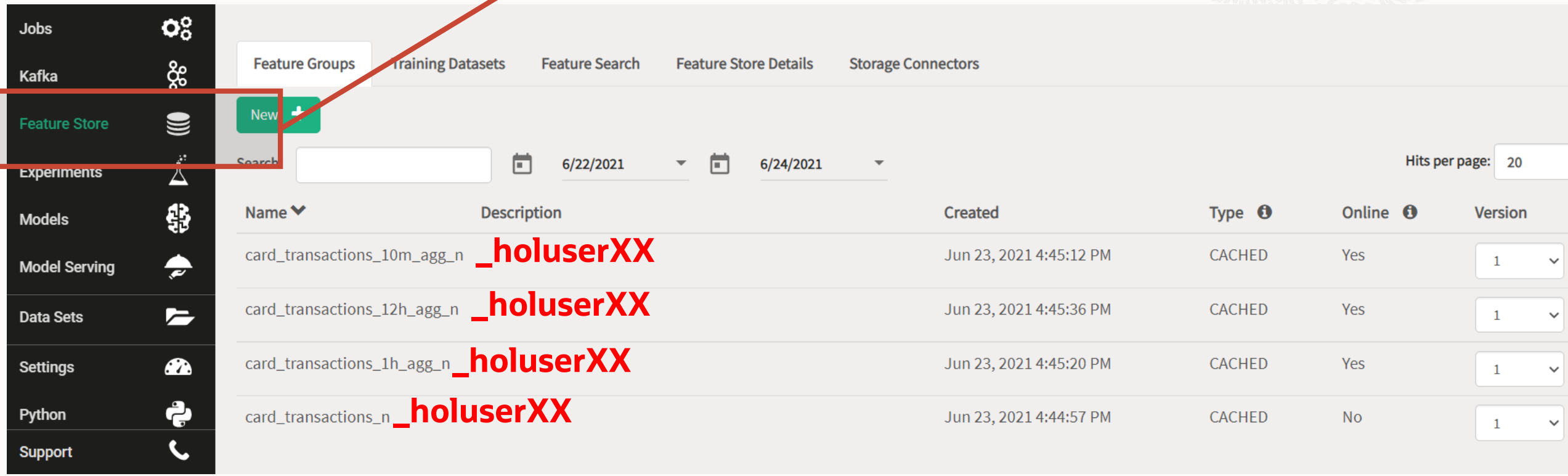
```
<hsfs.feature_group.FeatureGroup object at 0x7f346251eb90>
```

_holuserXX

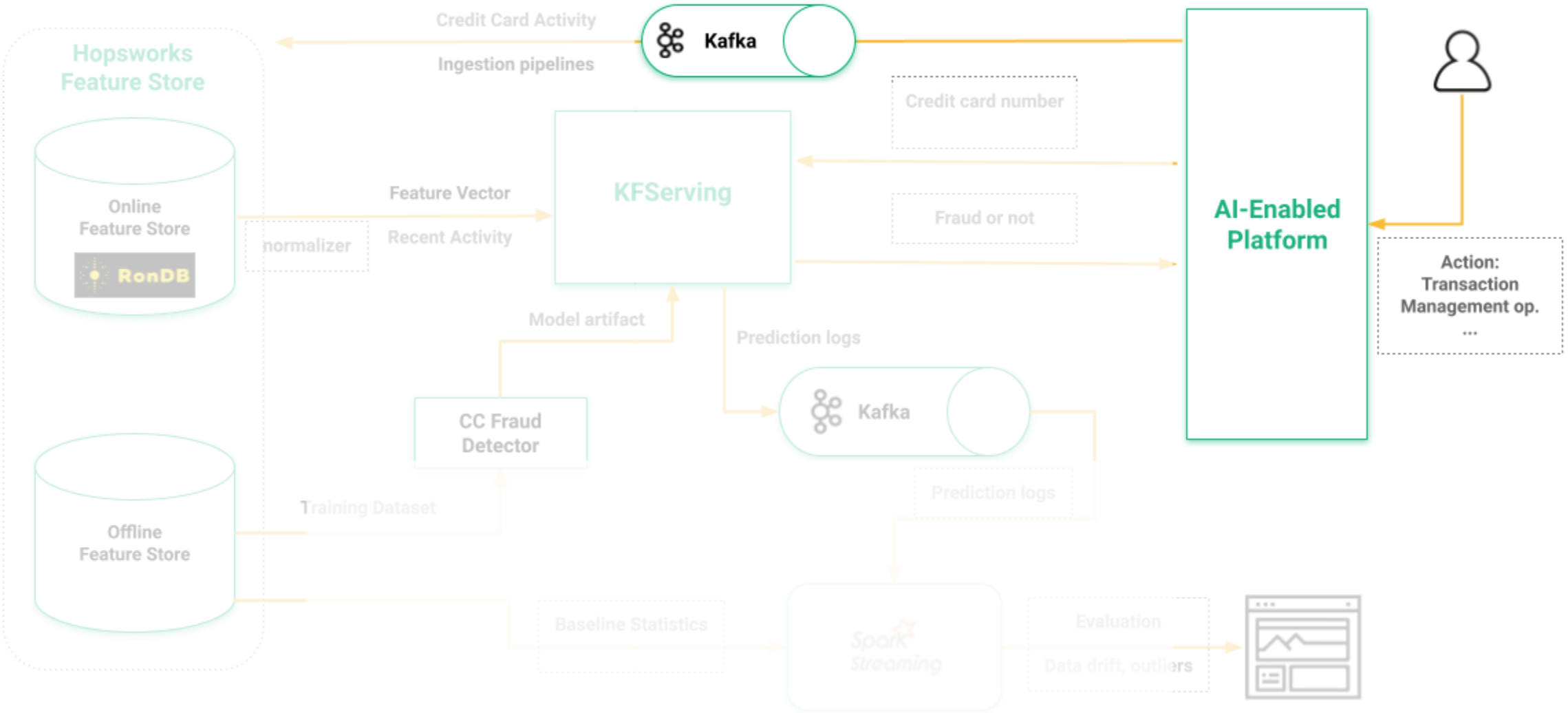
Create feature Group for Aggregated Data

10m, 1h, 12h

Check Results in Feature Store



2_produce_transactions



Hopswork | JupyterLa | python ch | reflection | introspec | hopswork | Training D | hopswork | hadoop r | Hadoop F | Read and

Not secure | 152.70.167.62/hopsworks-api/jupyter/44726/lab

Apps | free computer ebo... | microsoft | openkm | bigdata | olivier | dev | amazon | mcentric | Bookmarks | perso | oracle | teradata | gartner | Other bookmarks | Reading list

File Edit View Run Kernel Tabs Settings Help

/

Name	Last Modified
images	a day ago
0_create_kafka_topic.ipynb	a day ago
1_create_feature_groups.ip...	a day ago
2_produce_transactions.ipy...	5 hours ago
3_stream-ingestion.ipynb	5 hours ago
4_create_training_dataset.ip...	4 hours ago
5_online_serving.ipynb	4 hours ago
images.zip	20 minutes ago
README.md	a day ago

Terminal 1 | 1_create_feature_groups. | 2_produce_transactions.i | 4_create_training_datase | 5_online_serving.ipynb | 3_stream-ingestion.ipynb

Markdown | Python

title: "Generate credit card transactions data and send to kafka topic" date: 2021-04-25 type: technical_note draft: false

Generate credit card transactions data and send to kafka topic.

Inspiration of this example was taken from here.

Prerequisites

[3]:
```
#!pip install Faker
```

Imports

[4]:
```
from collections import defaultdict
from faker import Faker
import pandas as pd
import numpy as np
import datetime
import hashlib
import random
import math
import os

from hops import hdfs
from hops import pandas_helper as pandas
```

Use Faker library to generate Credit Card Operations

1 | 5 | Python | Idle

Mode: Command | Ln 1, Col 1 | 2_produce_transactions.ipynb

images.zip | 5_online_serving.ipynb | 4_create_trainin....ipynb | 3_stream-ingesti....ipynb | 2_produce_trans....ipynb | 1_create_feature....ipynb | Show all

66°F Mostly cloudy | ESP ES | 15:57 22/06/2021

Put the configuration to Connect to Kafka

```
# Change this according to your settings
KAFKA_BROKER_ADDRES = "broker.kafka.service.consul:9091"
KAFKA_TOPIC_NAME = "credit_card_transactions"
```

Generate Transactions

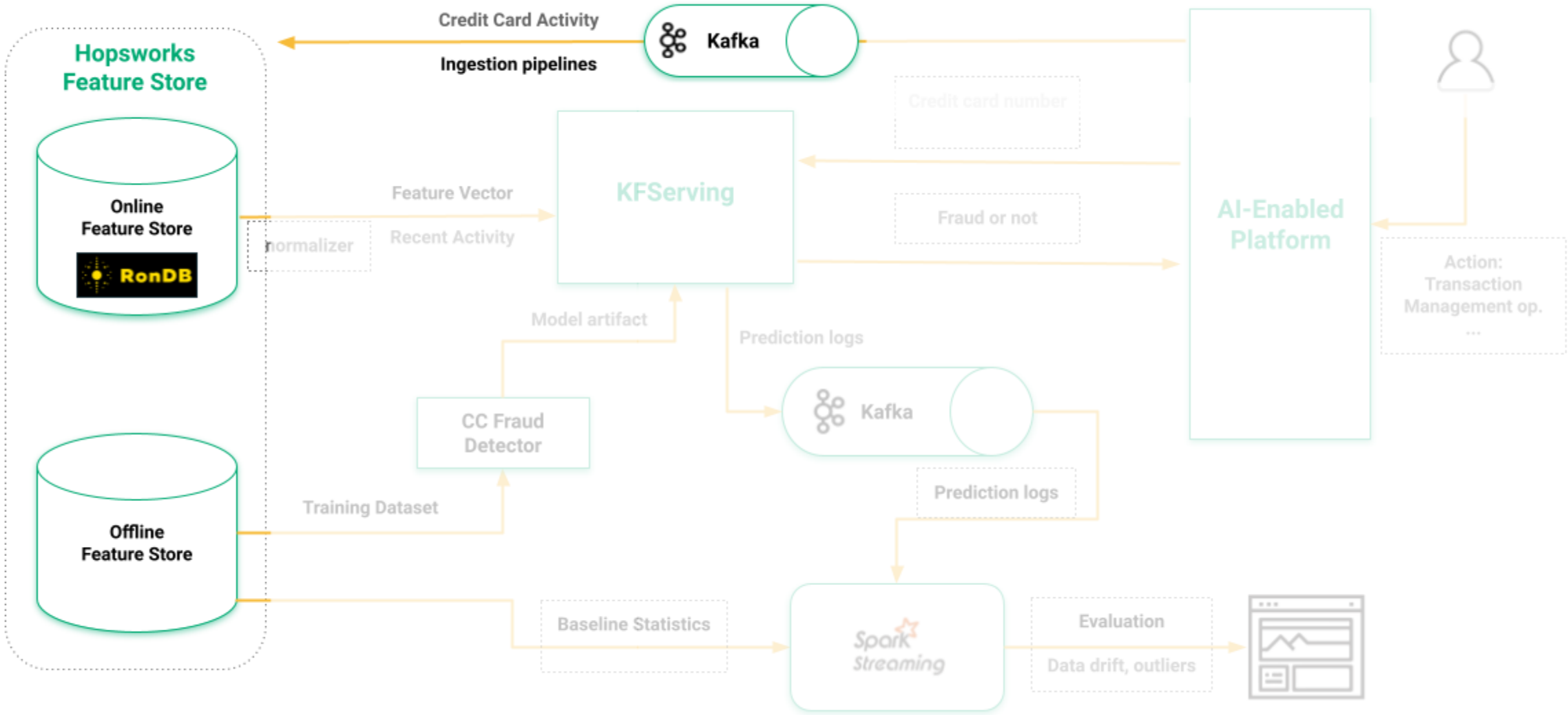
Generate Transactions to send to Kafka

Generate Unique Credit Card Numbers

Credit card numbers are uniquely assigned to users. Since, there are 10K users, we would want to generate 10K unique card numbers.

```
def generate_unique_credit_card_numbers(n: int) -> list:
    cc_ids = set()
    for _ in range(n):
        cc_id = faker.credit_card_number(card_type='visa')
        cc_ids.add(cc_id)
    return list(cc_ids)
```

3_stream-ingestion



Create a stream from the kafka topic

```
df_read = spark \
  .readStream \
  .format("kafka") \
  .option("kafka.bootstrap.servers", kafka.get_broker_endpoints()) \
  .option("kafka.security.protocol",kafka.get_security_protocol()) \
  .option("kafka.ssl.truststore.location", tls.get_trust_store()) \
  .option("kafka.ssl.truststore.password", tls.get_key_store_pwd()) \
  .option("kafka.ssl.keystore.location", tls.get_key_store()) \
  .option("kafka.ssl.keystore.password", tls.get_key_store_pwd()) \
  .option("kafka.ssl.key.password", tls.get_trust_store_pwd()) \
  .option("kafka.ssl.endpoint.identification.algorithm", "") \
  .option("startingOffsets", "earliest")\
  .option("subscribe", KAFKA_TOPIC_NAME) \
  .load()
```

Create a Stream for Kafka Topic

```
# Define schema to read from kafka topic
parse_schema = StructType([StructField('tid', StringType(), True),
                           StructField('datetime', StringType(), True),
                           StructField('cc_num', StringType(), True),
                           StructField('amount', StringType(), True)])
```

```
# Deserialise data from and create streaming query
df_deser = df_read.selectExpr("CAST(value AS STRING)")\
                  .select(from_json("value", parse_schema).alias("value"))\
                  .select("value.tid", "value.datetime", "value.cc_num", "value.amount")\
                  .selectExpr("CAST(tid as string)", "CAST(datetime as string)", "CAST(cc_num as long)", "CAST(amount as double)")
```

```
df_deser.isStreaming
```

Create windowing aggregations over different time windows using spark streaming.

```
# 10 minute window
windowed10mSignalDF =df_deser \
    .selectExpr("CAST(tid as string)", "CAST(datetime as timestamp)", "CAST(cc_num as long)", "CAST(amount as double)")\
    .withWatermark("datetime", "60 minutes") \
    .groupBy(window("datetime", "10 minutes"), "cc_num") \
    .agg(avg("amount").alias("avg_amt_per_10m"), stddev("amount").alias("stdev_amt_per_10m"), count("cc_num").alias("num_trans_per_10m"))\
    .select("cc_num", "num_trans_per_10m", "avg_amt_per_10m", "stdev_amt_per_10m")
```

```
windowed10mSignalDF.isStreaming
```

```
True
```

Create Aggregated Streams to send to Kafka

```
windowed10mSignalDF.printSchema()
```

```
root
 |-- cc_num: long (nullable = true)
 |-- num_trans_per_10m: long (nullable = false)
 |-- avg_amt_per_10m: double (nullable = true)
 |-- stdev_amt_per_10m: double (nullable = true)
```

```
# 1 hour window
windowed1hSignalDF = \
  df_deser \
    .selectExpr("CAST(tid as string)", "CAST(datetime as timestamp)", "CAST(cc_num as long)", "CAST(amount as double)")\
    .withWatermark("datetime", "60 minutes") \
    .groupBy(window("datetime", "60 minutes"), "cc_num") \
    .agg(avg("amount").alias("avg_amt_per_1h"), stddev("amount").alias("stdev_amt_per_1h"), count("cc_num").alias("num_trans_per_1h"))\
    .select("cc_num", "num_trans_per_1h", "avg_amt_per_1h", "stdev_amt_per_1h")
```

Get feature groups from hopsworks feature store.

Create Feature Groups

```
card_transactions = fs.get_feature_group("card_transactions_n", version = 1)
card_transactions_10m_agg = fs.get_feature_group("card_transactions_10m_agg_n", version = 1)
card_transactions_1h_agg = fs.get_feature_group("card_transactions_1h_agg_n", version = 1)
card_transactions_12h_agg = fs.get_feature_group("card_transactions_12h_agg_n", version = 1)
```

Insert streaming dataframes to the online feature group

Insert Aggregated Streams

Now we are ready to write this streaming dataframe as a long living application to the online storage of the other feature group.

```
query_10m = card_transactions_10m_agg.insert_stream(windowed10mSignalDF)
```

```
StatisticsWarning: Stream ingestion for feature group `card_transactions_10m_agg_n`, with version `1` will not compute statistics.
```

```
query_1h = card_transactions_1h_agg.insert_stream(windowed1hSignalDF)
```

```
StatisticsWarning: Stream ingestion for feature group `card_transactions_1h_agg_n`, with version `1` will not compute statistics.
```

```
query_12h = card_transactions_12h_agg.insert_stream(windowed12hSignalDF)
```

```
StatisticsWarning: Stream ingestion for feature group `card_transactions_12h_agg_n`, with version `1` will not compute statistics.
```

Check if spark streaming query is active

```
print("IsActive:\n query_10m: {}\n query_1h: {}\n query_12h: {}".format(query_10m.isActive, query_1h.isActive, query_12h.isActive))
```

```
IsActive:
```

Simulate fraudulent transactions

Execute Simulation Fraudulent Transactions to Topic "credit_card_transactions"

> NOTE: Before polluting the `credit_card_transactions` topic with fraudulent transactions, you can run the notebooks `3_stream-ingestion`, `4_create_training_dataset` and `5_model_training` to create a Training Dataset with the original transactions and train an autoencoder that learns these patterns. Once the model is served, you can create fraudulent transactions by running the code below.

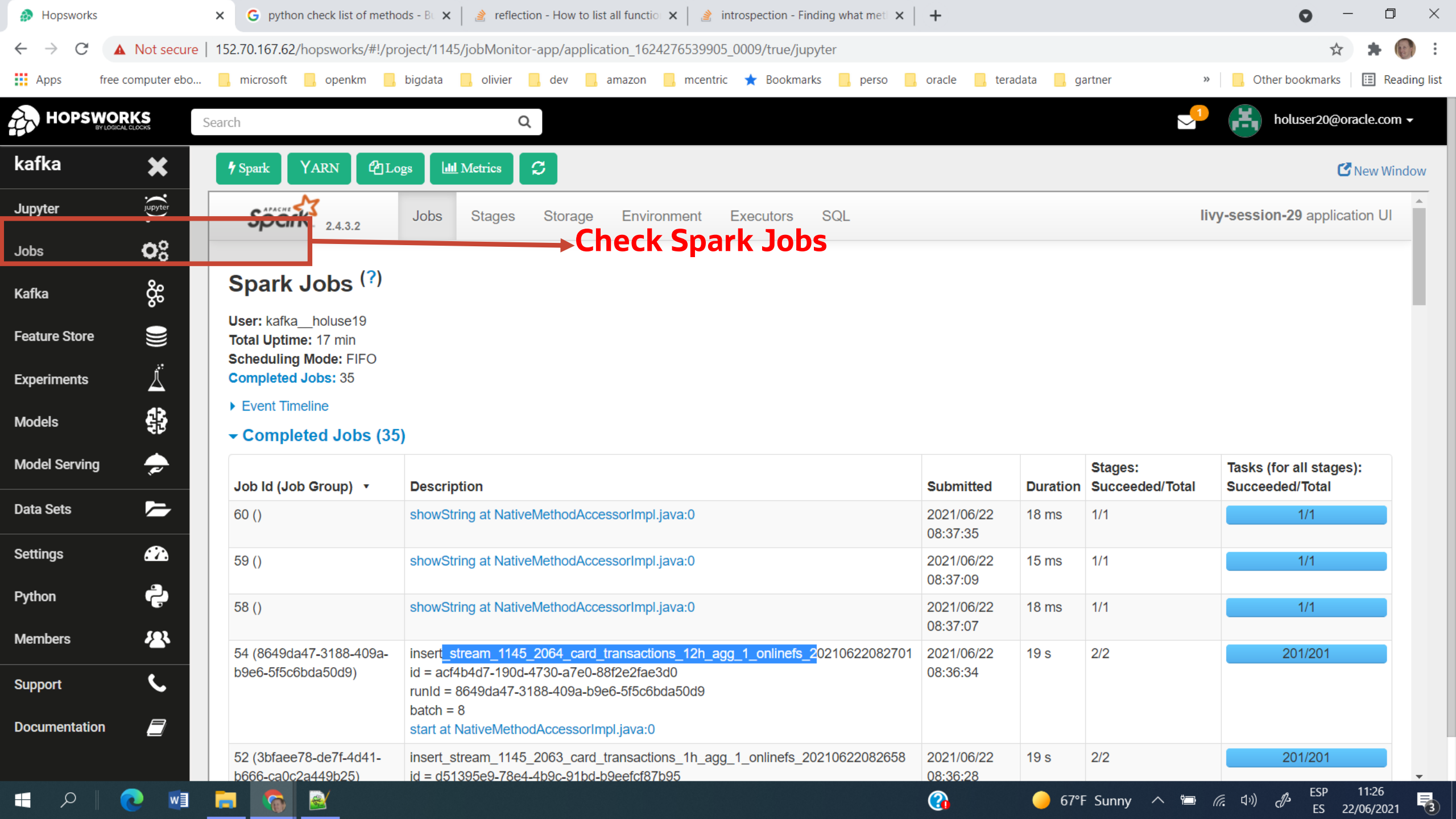
Create Attack Transaction Chains

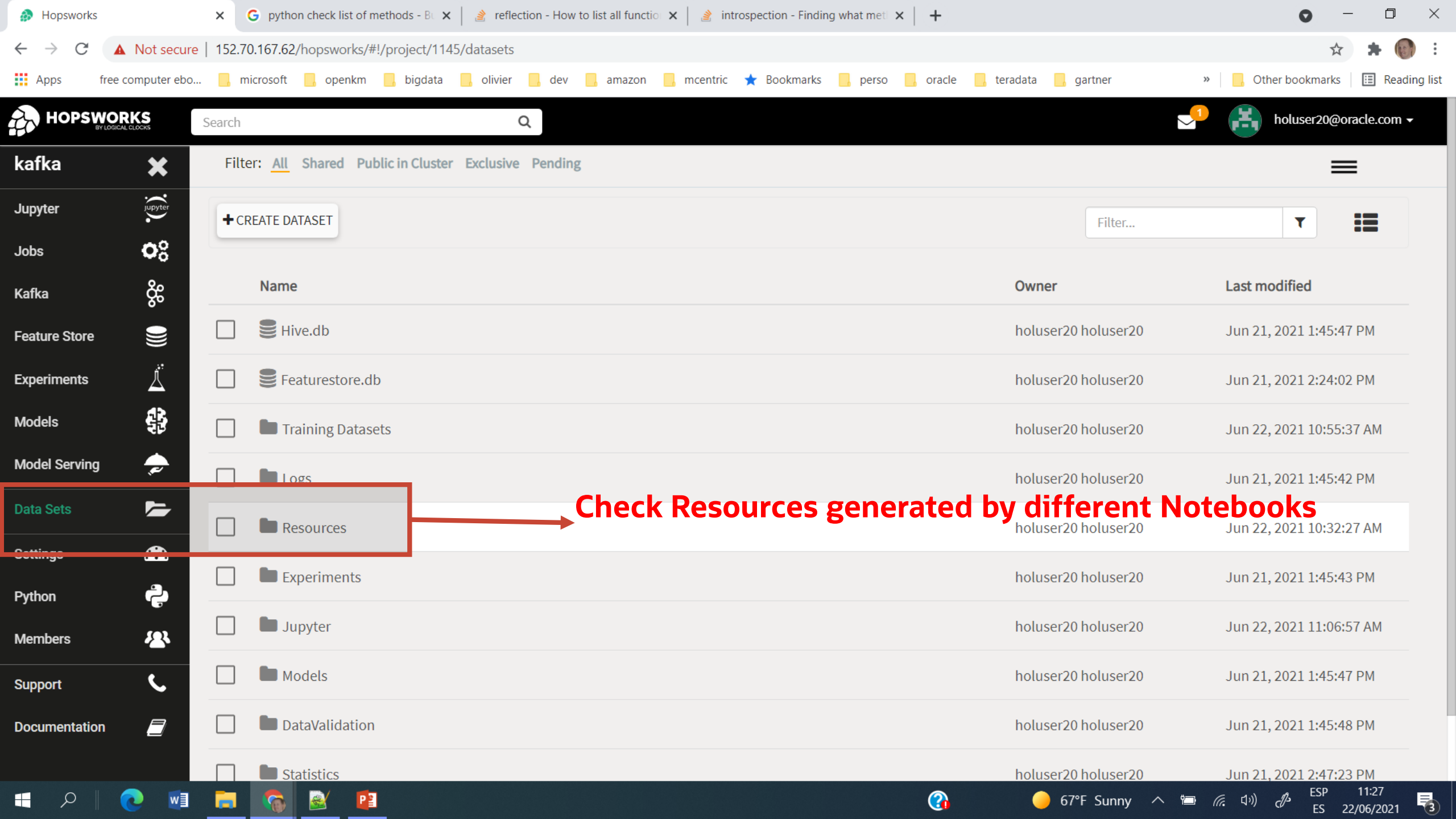
```
FRAUD_RATIO = 0.0025 # percentage of transactions that are fraudulent
NUMBER_OF_FRAUDULENT_TRANSACTIONS = int(FRAUD_RATIO * TOTAL_UNIQUE_TRANSACTIONS)
ATTACK_CHAIN_LENGTHS = [3, 4, 5, 6, 7, 8, 9, 10]
```

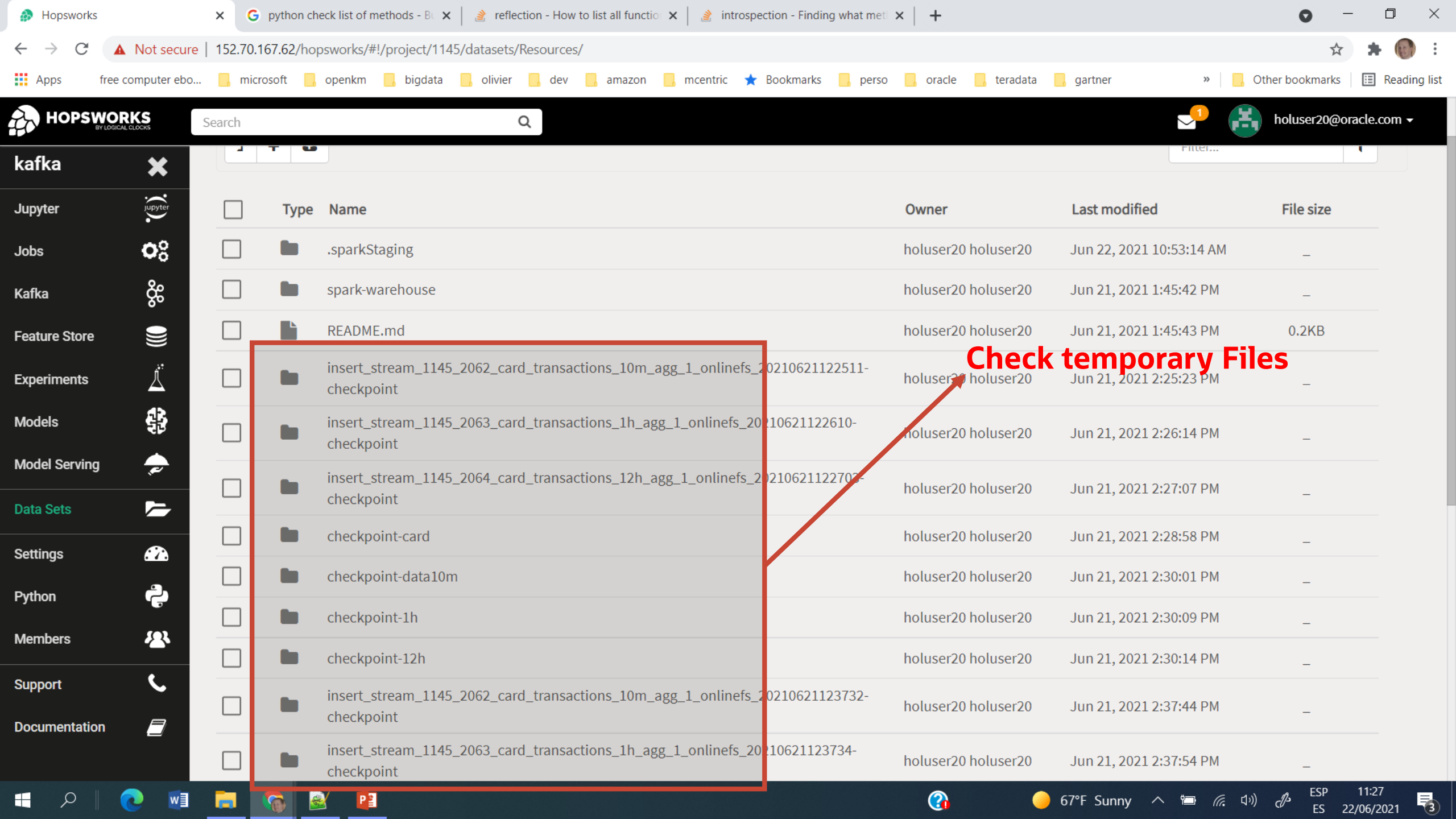
```
visited = set()
chains = defaultdict(list)
```

```
def size(chains: dict) -> int:
    counts = {key: len(values)+1 for (key, values) in chains.items()}
    return sum(counts.values())
```

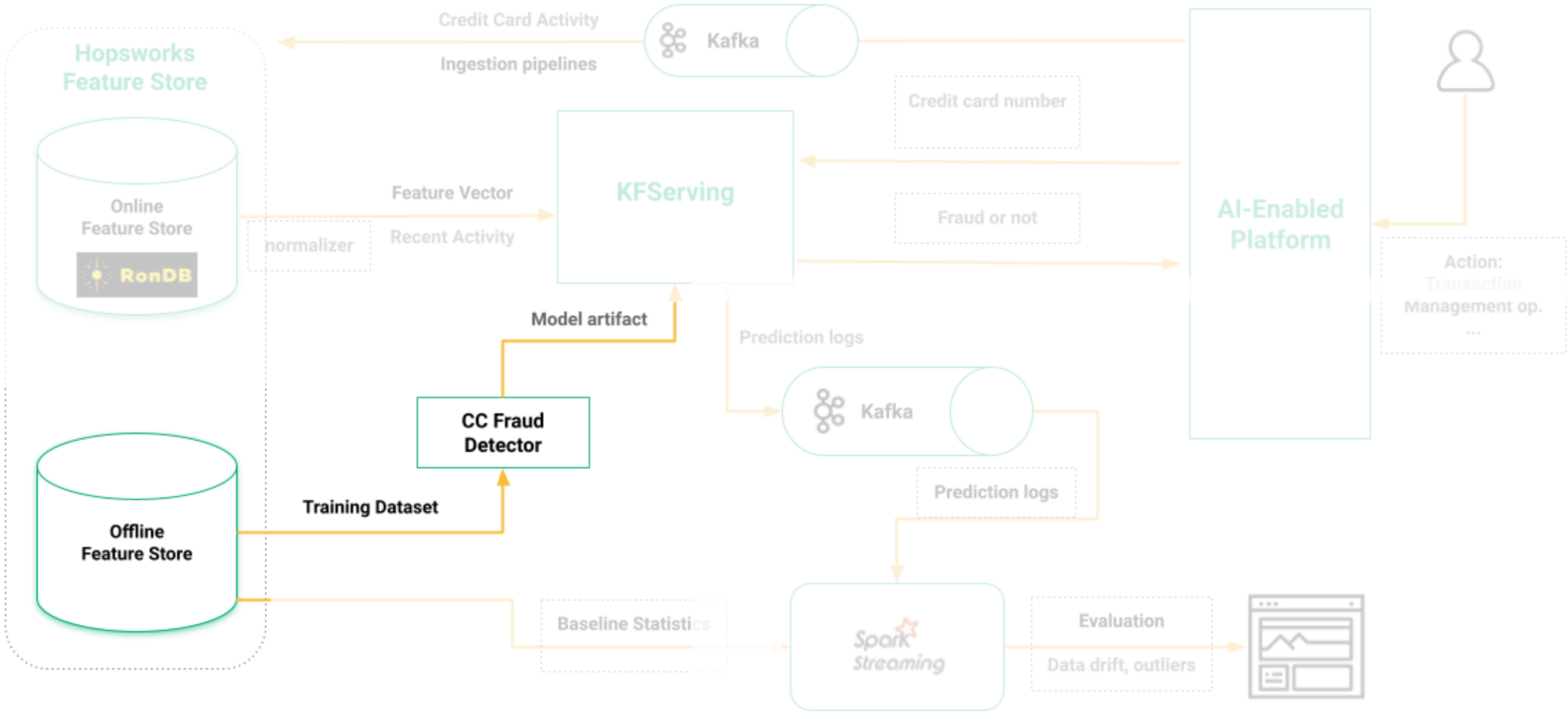
```
def create_attack_chain(i: int):
    chain length = random.choice(ATTACK CHAIN LENGTHS)
```


Hopsworks
python check list of methods - B
reflection - How to list all functio
introspection - Finding what met
Not secure | 152.70.167.62/hopsworks/#!/project/1145/jobMonitor-app/application_1624276539905_0009/true/jupyter
Apps
free computer ebo...
microsoft
openkm
bigdata
olivier
dev
amazon
mcentric
Bookmarks
perso
oracle
teradata
gartner
Other bookmarks
Reading list
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
kafka
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Support
Documentation
Spark
YARN
Logs
Metrics
New Window
2.4.3.2
Jobs
Stages
Storage
Environment
Executors
SQL
livy-session-29 application UI
Check Spark Jobs
Spark Jobs (?)
User: kafka__holuse19
Total Uptime: 17 min
Scheduling Mode: FIFO
Completed Jobs: 35
Event Timeline
Completed Jobs (35)
Job Id (Job Group)
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
60 ()
showString at NativeMethodAccessorImpl.java:0
2021/06/22 08:37:35
18 ms
1/1
1/1
59 ()
showString at NativeMethodAccessorImpl.java:0
2021/06/22 08:37:09
15 ms
1/1
1/1
58 ()
showString at NativeMethodAccessorImpl.java:0
2021/06/22 08:37:07
18 ms
1/1
1/1
54 (8649da47-3188-409a-b9e6-5f5c6bda50d9)
insert_stream_1145_2064_card_transactions_12h_agg_1_onlinefs_20210622082701
id = acf4b4d7-190d-4730-a7e0-88f2e2fae3d0
runId = 8649da47-3188-409a-b9e6-5f5c6bda50d9
batch = 8
start at NativeMethodAccessorImpl.java:0
2021/06/22 08:36:34
19 s
2/2
201/201
52 (3bfaee78-de7f-4d41-b666-ca0c2a449b25)
insert_stream_1145_2063_card_transactions_1h_agg_1_onlinefs_20210622082658
id = d51395e9-78e4-4b9c-91bd-b9eefcf87b95
2021/06/22 08:36:28
19 s
2/2
201/201
67°F Sunny
ESP
ES
11:26
22/06/2021


Hopsworks
python check list of methods - B
reflection - How to list all functio
introspection - Finding what met
Not secure | 152.70.167.62/hopsworks/#!/project/1145/datasets
Apps
free computer ebo...
microsoft
openkm
bigdata
olivier
dev
amazon
mcentric
Bookmarks
perso
oracle
teradata
gartner
Other bookmarks
Reading list
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
kafka
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Support
Documentation
Filter: All Shared Public in Cluster Exclusive Pending
+ CREATE DATASET
Filter...
Name
Owner
Last modified
Hive.db
holuser20 holuser20
Jun 21, 2021 1:45:47 PM
Featurestore.db
holuser20 holuser20
Jun 21, 2021 2:24:02 PM
Training Datasets
holuser20 holuser20
Jun 22, 2021 10:55:37 AM
Logs
holuser20 holuser20
Jun 21, 2021 1:45:42 PM
Resources
holuser20 holuser20
Jun 22, 2021 10:32:27 AM
Experiments
holuser20 holuser20
Jun 21, 2021 1:45:43 PM
Jupyter
holuser20 holuser20
Jun 22, 2021 11:06:57 AM
Models
holuser20 holuser20
Jun 21, 2021 1:45:47 PM
DataValidation
holuser20 holuser20
Jun 21, 2021 1:45:48 PM
Statistics
holuser20 holuser20
Jun 21, 2021 2:47:23 PM
Check Resources generated by different Notebooks
67°F Sunny
ESP
ES
11:27
22/06/2021


Hopsworks
python check list of methods - B
reflection - How to list all functio
introspection - Finding what met
Not secure | 152.70.167.62/hopsworks/#!/project/1145/datasets/Resources/
Apps
free computer ebo...
microsoft
openkm
bigdata
olivier
dev
amazon
mcentric
Bookmarks
perso
oracle
teradata
gartner
Other bookmarks
Reading list
HOPSWORKS
BY LOGICAL CLOCKS
Search
holuser20@oracle.com
kafka
Jupyter
Jobs
Kafka
Feature Store
Experiments
Models
Model Serving
Data Sets
Settings
Python
Members
Support
Documentation
Type
Name
Owner
Last modified
File size
.sparkStaging
holuser20 holuser20
Jun 22, 2021 10:53:14 AM
spark-warehouse
holuser20 holuser20
Jun 21, 2021 1:45:42 PM
README.md
holuser20 holuser20
Jun 21, 2021 1:45:43 PM
0.2KB
insert_stream_1145_2062_card_transactions_10m_agg_1_onlinefs_20210621122511-checkpoint
holuser20 holuser20
Jun 21, 2021 2:25:23 PM
insert_stream_1145_2063_card_transactions_1h_agg_1_onlinefs_20210621122610-checkpoint
holuser20 holuser20
Jun 21, 2021 2:26:14 PM
insert_stream_1145_2064_card_transactions_12h_agg_1_onlinefs_20210621122703-checkpoint
holuser20 holuser20
Jun 21, 2021 2:27:07 PM
checkpoint-card
holuser20 holuser20
Jun 21, 2021 2:28:58 PM
checkpoint-data10m
holuser20 holuser20
Jun 21, 2021 2:30:01 PM
checkpoint-1h
holuser20 holuser20
Jun 21, 2021 2:30:09 PM
checkpoint-12h
holuser20 holuser20
Jun 21, 2021 2:30:14 PM
insert_stream_1145_2062_card_transactions_10m_agg_1_onlinefs_20210621123732-checkpoint
holuser20 holuser20
Jun 21, 2021 2:37:44 PM
insert_stream_1145_2063_card_transactions_1h_agg_1_onlinefs_20210621123734-checkpoint
holuser20 holuser20
Jun 21, 2021 2:37:54 PM
Check temporary Files
67°F Sunny
ESP
ES
11:27
22/06/2021

4_create_training_dataset



Install GIT Transformer

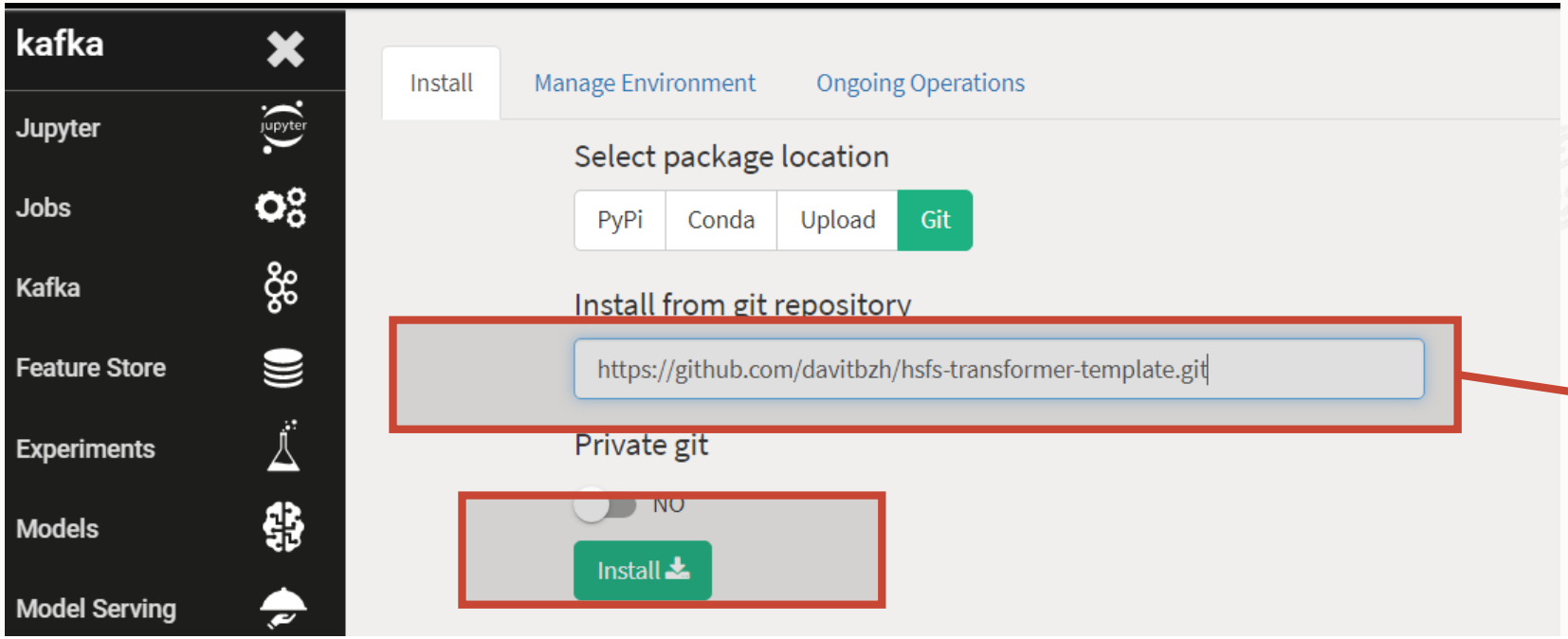
Create transformation function

Transformation functions are python functions that receive a feature value as input and returns the result of applying a specific transformation on it. It's possible to defined your own python functions to transform feature values. These functions are created at the feature store level and can be used to generate training datasets by attaching them to specific features composing the dataset.

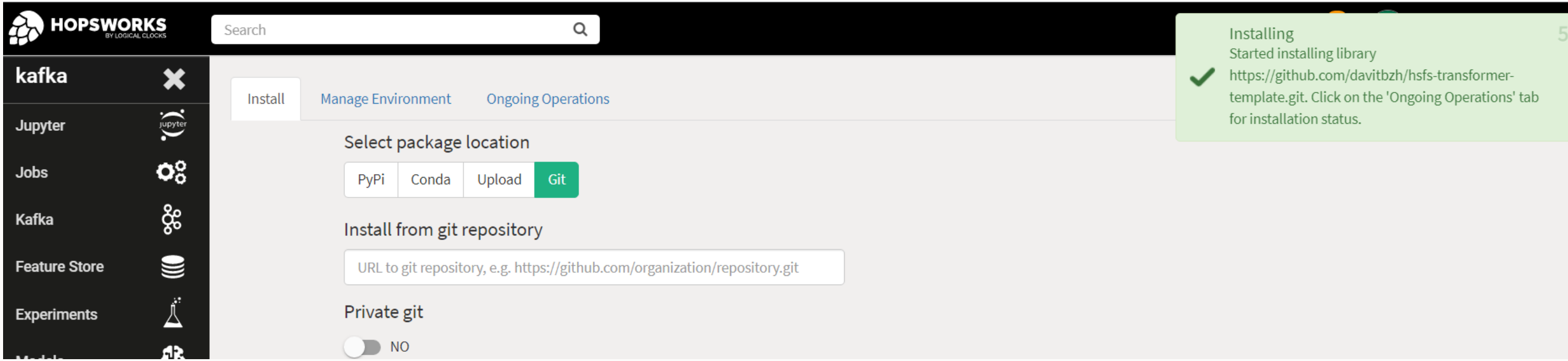
For the sake of the example, in this demo we define a `min_max` transformation function and attach it to all the features in the training dataset. You can install the module with the demo transformation function using the Hopsworks UI by following the steps below:

> Hopsworks UI -> Python -> Install with Git -> https://github.com/davitbzh/hsfs-transformer-template.git

```
from hsfs_transformers import scalers
min_max_normalizer = fs.create_transformation_function(transformation_function=scalers.min_max,
                                                       output_type=float,
                                                       version=1)
min_max_normalizer.save()
```



Install GIT Transformer



Create Transformation Functions

Create transformation function

Transformation functions are python functions that receive a feature value as input and returns the result of applying a specific transformation on it. It's possible to defined your own python functions to transform feature values. These functions are created at the feature store level and can be used to generate training datasets by attaching them to specific features composing the dataset.

For the sake of the example, in this demo we define a `min_max` transformation function and attach it to all the feature in the training dataset. You can install the module with the demo transformation function using the Hopsworks UI by following the steps below:

> Hopsworks UI -> Python -> Install with Git -> https://github.com/davitbzh/hsfs-transformer-template.git

```
from hsfs_transformers import scalers
min_max_normalizer = fs.create_transformation_function(transformation_function=scalers.min_max,
                                                       output_type=float,
                                                       version=1)

min_max_normalizer.save()
```

Get feature groups

Get Feature Groups

```
card_transactions_10m_agg = fs.get_feature_group("card_transactions_10m_agg_n", version = 1)
card_transactions_1h_agg = fs.get_feature_group("card_transactions_1h_agg_n", version = 1)
card_transactions_12h_agg = fs.get_feature_group("card_transactions_12h_agg_n", version = 1)
```

Create training dataset

```
query = card_transactions_10m_agg.select(["stdev_amt_per_10m", "avg_amt_per_10m", "num_trans_per_10m"])\
                                 .join(card_transactions_1h_agg.select(["stdev_amt_per_1h", "avg_amt_per_1h", "num_trans_per_1h"]))\
                                 .join(card_transactions_12h_agg.select(["stdev_amt_per_12h", "avg_amt_per_12h", "num_trans_per_12h"]))
```

Create Training Dataset for Card Fraud Model

```
td_meta = fs.create_training_dataset(name="card_fraud_model",
                                     description="Training dataset to train card fraud model",
                                     data_format="tfrecord",
                                     transformation_functions={"stdev_amt_per_10m":min_max,
                                                               "avg_amt_per_10m":min_max,
                                                               "num_trans_per_10m":min_max,
                                                               "stdev_amt_per_1h":min_max,
                                                               "avg_amt_per_1h":min_max,
                                                               "num_trans_per_1h":min_max,
                                                               "stdev_amt_per_12h":min_max,
                                                               "avg_amt_per_12h":min_max,
                                                               "num_trans_per_12h":min_max},
                                     statistics_config={"enabled": True, "histograms": True, "correlations": False},
                                     version=1)
```

Check MetaData for Training Dataset

```
td_meta.read().show()
```

```
+---------------+---------------+---------------+-------------+---------------+----------------+----------------+----------------+---------------+
|avg_amt_per_10m|avg_amt_per_12h|stdev_amt_per_1h|avg_amt_per_1h|num_trans_per_1h|num_trans_per_12h|stdev_amt_per_12h|num_trans_per_10m|stdev_amt_per_10m|
+---------------+---------------+---------------+-------------+---------------+----------------+----------------+----------------+---------------+
|       1.005665|        1.03796|      1.0266877|    1.0247675|         1.0015|          1.0065|       1.0697409|           1.001|         1.0005|
|        1.03907|       1.139612|         1.0005|      1.02391|          1.001|           1.003|       1.1538386|           1.001|         1.0005|
|        1.04755|       1.318325|         1.0005|      5.18205|          1.001|           1.009|       1.9687029|           1.001|         1.0005|
|       1.043745|      1.0632564|         1.0005|      1.02109|          1.001|           1.006|       1.0999271|           1.001|         1.0005|
|       1.006885|       1.101762|         1.0005|     1.045685|          1.001|          1.0055|       1.1411338|           1.001|         1.0005|
|         1.0099|      1.1035554|         1.0005|     1.046075|          1.001|           1.006|       1.1777302|           1.001|         1.0005|
|        1.75089|      1.0588467|         1.0005|       1.2385|          1.001|          1.0035|       1.0908353|           1.001|         1.0005|
|       1.001085|      1.0586371|      3.8128898|      2.65805|          1.002|           1.004|       1.0761116|           1.001|         1.0005|
|       1.000665|       1.358466|      1.2214001|     1.206005|         1.0015|          1.0075|       1.8494561|           1.001|         1.0005|
|        1.00264|      1.1111832|         1.0005|      1.35051|          1.001|          1.0045|       1.1633795|           1.001|         1.0005|
|        1.05099|      1.1732512|      1.0785328|     1.1408975|         1.0015|           1.005|       1.1760855|           1.001|         1.0005|
|        1.30523|      1.6181775|      2.3883066|     2.0242176|         1.0015|          1.0065|       2.3270547|           1.001|         1.0005|
```

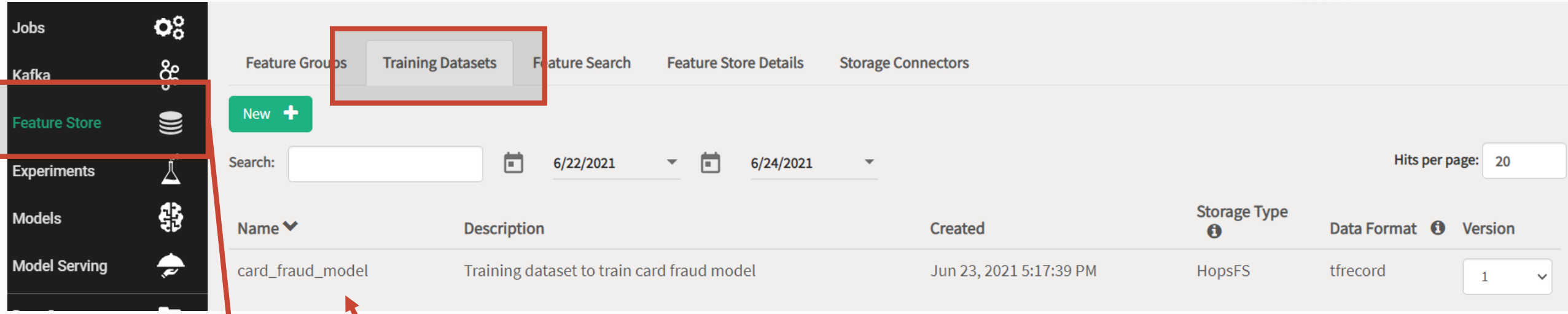
Check descriptive statistics

```
td_meta = fs.get_training_dataset("card_fraud_model", 1)
statistics = td_meta.get_statistics()

for feat_list in statistics.content.items():
    for stats in feat_list[1]:
        print("Feature: " + str(stats['column']))
        print(stats)
```

Get Training Dataset "card_fraud_model" for Statistics

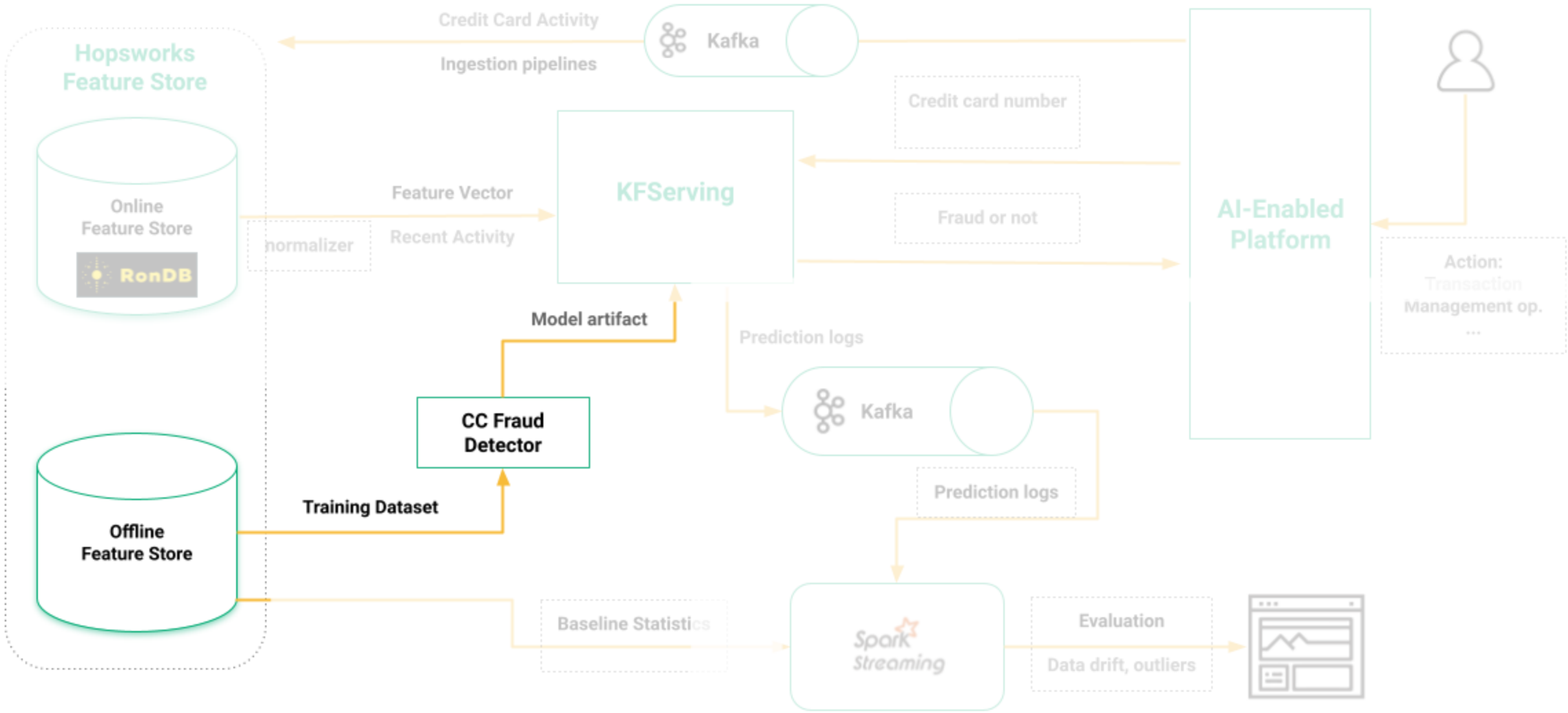
```
Feature: num_trans_per_1h
{'dataType': 'Fractional', 'column': 'num_trans_per_1h', 'sum': 100.1280027627945, 'completeness': 1, 'histogram': [{'count': 10, 'value': '1.0
02', 'ratio': 0.1}, {'count': 1, 'value': '1.0025', 'ratio': 0.01}, {'count': 33, 'value': '1.0015', 'ratio': 0.33}, {'count': 56, 'value': '1.
001', 'ratio': 0.56}], 'distinctness': 0.04, 'entropy': 0.9668672345930647, 'approximateNumDistinctValues': 4, 'isDataTypeInferred': 'false',
'uniqueness': 0.01, 'mean': 1.001280027627945, 'maximum': 1.002500057220459, 'stdDev': 0.0003557872363590212, 'minimum': 1.0010000467300415, 'a
pproxPercentiles': []}
Feature: stdev_amt_per_10m
{'dataType': 'Fractional', 'column': 'stdev_amt_per_10m', 'sum': 103.46918630599976, 'completeness': 1, 'histogram': [{'count': 1, 'value': '1.
0158194', 'ratio': 0.01}, {'count': 1, 'value': '1.0237107', 'ratio': 0.01}, {'count': 1, 'value': '4.2807717', 'ratio': 0.01}, {'count': 96,
'value': '1.0005', 'ratio': 0.96}, {'count': 1, 'value': '1.1008879', 'ratio': 0.01}], 'distinctness': 0.05, 'entropy': 0.22339592217896861, 'a
pproximateNumDistinctValues': 5, 'isDataTypeInferred': 'false', 'uniqueness': 0.04, 'mean': 1.0346918630599975, 'maximum': 4.280771732330322,
'stdDev': 0.32640657836797243, 'minimum': 1.000499963760376, 'approxPercentiles': []}
Feature: stdev_amt_per_1h
{'dataType': 'Fractional', 'column': 'stdev_amt_per_1h', 'sum': 120.96687138080597, 'completeness': 1, 'histogram': [{'count': 1, 'value': '1.0
512278', 'ratio': 0.01}, {'count': 1, 'value': '1.2470504', 'ratio': 0.01}, {'count': 1, 'value': '1.1666249', 'ratio': 0.01}, {'count': 1, 'va
lue': '1.0686651', 'ratio': 0.01}, {'count': 1, 'value': '1.0186267', 'ratio': 0.01}, {'count': 1, 'value': '1.0060296', 'ratio': 0.01}, {'coun
t': 1, 'value': '2.3883066', 'ratio': 0.01}, {'count': 1, 'value': '1.0785328', 'ratio': 0.01}, {'count': 1, 'value': '1.105774', 'ratio': 0.0
1}, {'count': 1, 'value': '1.1941308', 'ratio': 0.01}, {'count': 1, 'value': '1.0181073', 'ratio': 0.01}, {'count': 1, 'value': '1.3772634', 'r
atio': 0.01}, {'count': 1, 'value': '1.1348182', 'ratio': 0.01}, {'count': 1, 'value': '1.1406308', 'ratio': 0.01}, {'count': 1, 'value': '1.01
69014', 'ratio': 0.01}, {'count': 1, 'value': '3.8128898', 'ratio': 0.01}, {'count': 1, 'value': '1.2591444', 'ratio': 0.01}, {'count': 1, 'val
ue': '1.1521285', 'ratio': 0.01}, {'count': 1, 'value': '2.1266055', 'ratio': 0.01}, {'count': 1, 'value': '1.1044518', 'ratio': 0.01}, {'coun
t': 1, 'value': '1.1864266', 'ratio': 0.01}, {'count': 1, 'value': '1.0185984', 'ratio': 0.01}, {'count': 56, 'value': '1.0005', 'ratio': 0.5
```

Check Training Dataset generated



- **Click on Feature Store**
- **Click on Training Datasets**
- **Check Card Fraud Model**

5_model_training



Define the autoencoder

Create Fraud Detector

```
class CCFraudDetector(tf.keras.Model):
  def __init__(self, input_dim):
    super(CCFraudDetector, self).__init__()
    self.encoder = tf.keras.Sequential([
        tf.keras.layers.Dense(16, activation='selu', input_shape=(input_dim,)),
        tf.keras.layers.Dense(8, activation='selu'),
        tf.keras.layers.Dense(4, activation='linear', name="bottleneck")])

    self.decoder = tf.keras.Sequential([
        tf.keras.layers.Dense(8, activation='selu'),
        tf.keras.layers.Dense(16, activation='selu'),
        tf.keras.layers.Dense(input_dim, activation='selu')])

  def call(self, x):
    encoded = self.encoder(x)
    decoded = self.decoder(encoded)
    return decoded
```

Extend the autoencoder to return the reconstruction loss

```
class CCFraudDetectorModule(tf.Module):
    def __init__(self, detector):
        self.detector = detector

    @tf.function()
    def reconstruct(self, instances):
```

Create experiment

Create Experiment

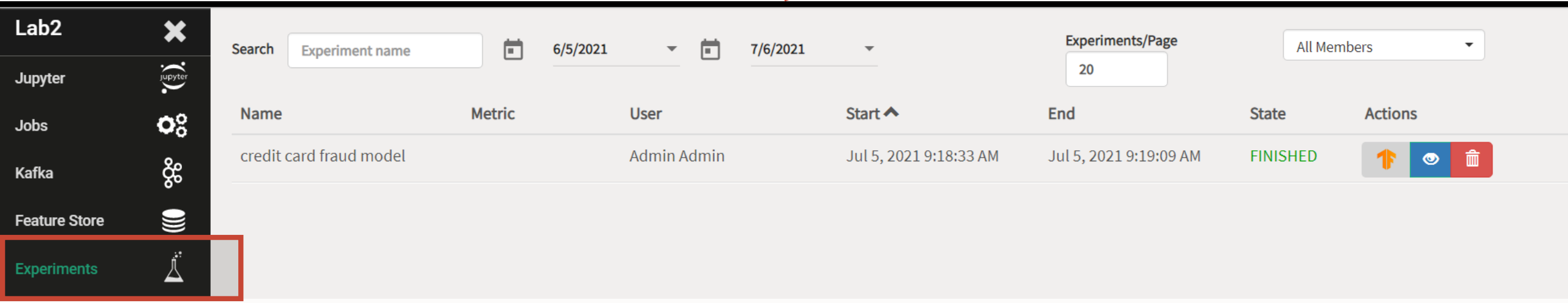
```
def experiment_wrapper():

    import os
    import sys
    import uuid
    import random

    import tensorflow as tf
    from tensorflow.keras.callbacks import TensorBoard
    from hops import tensorboard

    from hops import model as hops_model
    from hops import hdfs
    import hsfs

    # Create a connection
    connection = hsfs.connection(engine='training')
    # Get the feature store handle for the project's feature store
    fs = connection.get_feature_store()
    # Get training dataset
    td_meta = fs.get_training_dataset("card_fraud_model", 1)

    input_dim = 9
    BATCH_SIZE = 32
    EPOCHS = 5

    # Training data
    train_input = td_meta.tf_data(target_name=None, is_training=True)
```

Launch experiment

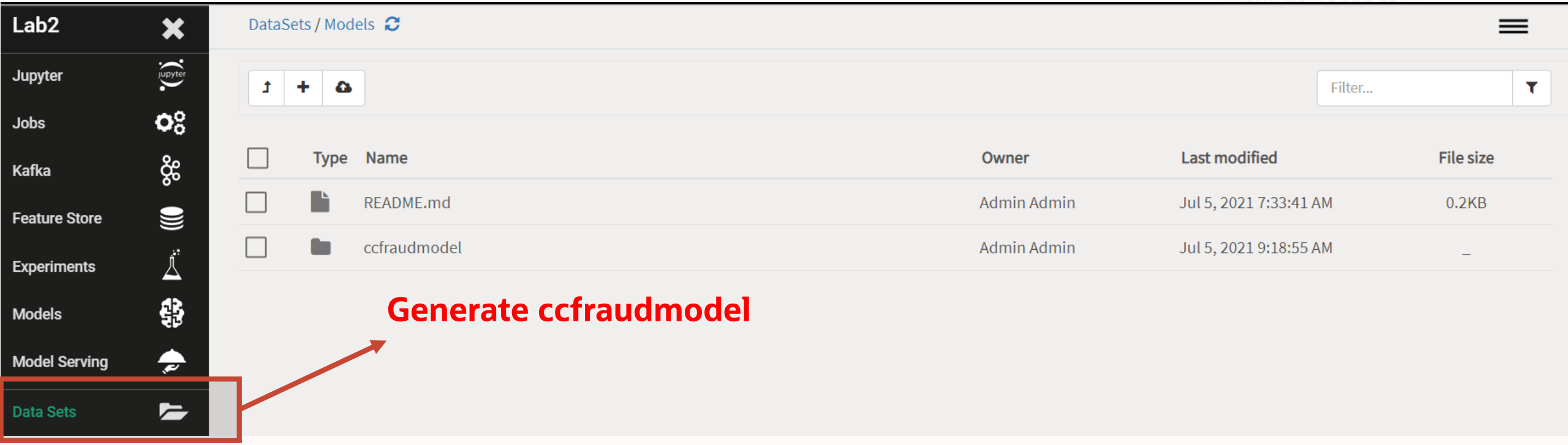
Launch Experiment

```
from hops import experiment
from hops import hdfs

experiment.launch(experiment_wrapper, name='credit card fraud model', local_logdir=True, metric_key='loss')
```

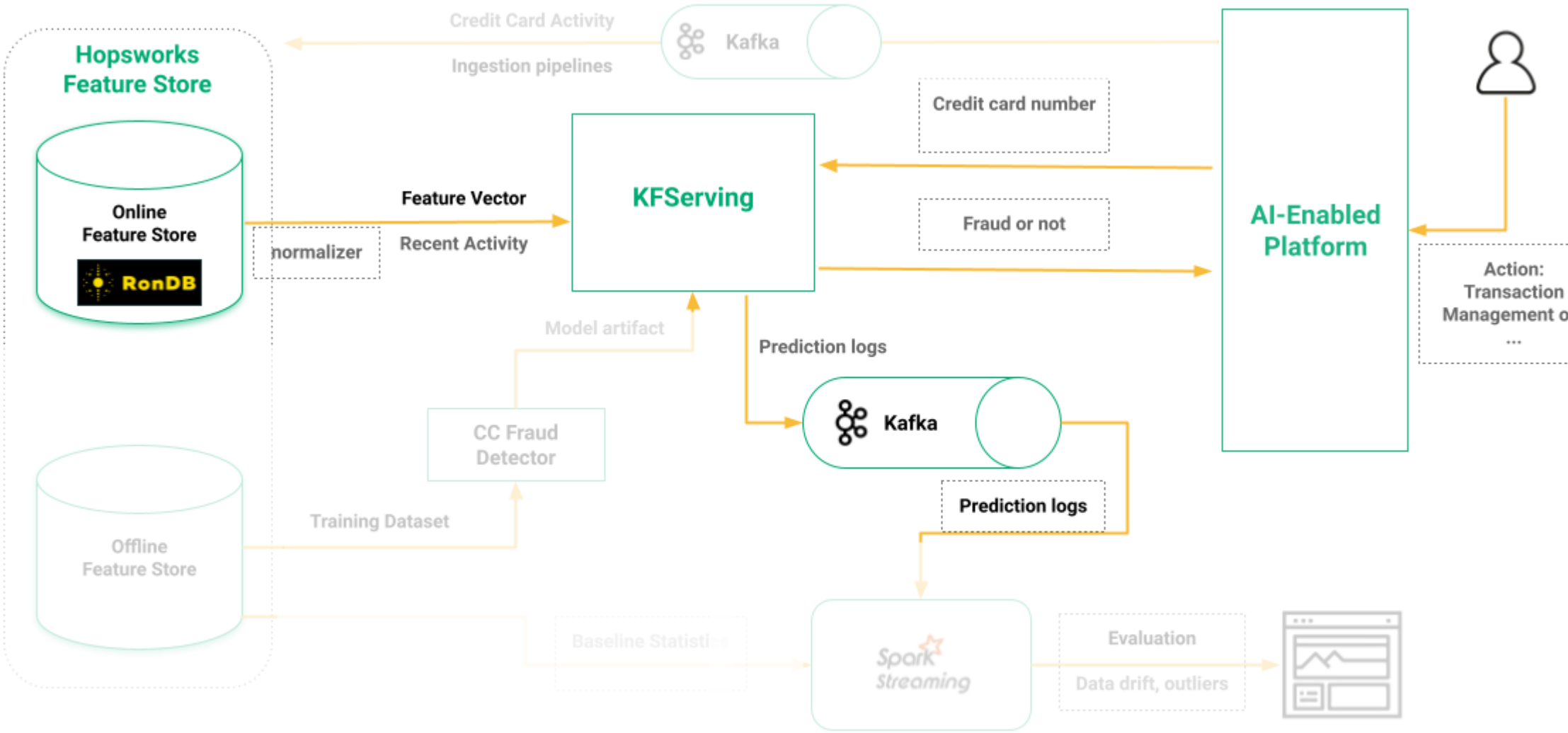
```
Finished Experiment

('hdfs://rpc.namenode.service.consul:8020/Projects/card_fraud_detection/Experiments/application_1623853832952_0045_1', {'loss': 1.7422299385070
8, 'log': 'Experiments/application_1623853832952_0045_1/output.log'})
```



Generate Credit Card Fraud Model



6_online_serving



Query Model Repository for best mnist Model

```
from hops import model
from hops.model import Metric
MODEL_NAME="ccfraudmodel"
EVALUATION_METRIC="loss"
```

```
best_model = model.get_best_model(MODEL_NAME, EVALUATION_METRIC, Metric.MIN)
```

```
print('Model name: ' + best_model['name'])
print('Model version: ' + str(best_model['version']))
print(best_model['metrics'])
```

```
Model name: ccfraudmodel
Model version: 1
{'loss': '1.74222993850708'}
```

Get Best Model for Fraud Detector

Create Model Serving of Exported Model

```
from hops import serving
from hops import hdfs
```

```
TOPIC_NAME = "credit_card_prediction_logs"

SERVING_NAME = MODEL_NAME
MODEL_PATH="/Models/" + best_model['name']
TRANSFORMER_PATH = "/Projects/" + hdfs.project_name() + "/Jupyter/card_activity_transformer.py"

response = serving.create_or_update(SERVING_NAME, MODEL_PATH, model_version=best_model['version'], artifact_version="CREATE",
                                    kfserving=True, transformer=TRANSFORMER_PATH,
                                    topic_name=TOPIC_NAME, inference_logging="TRANSFORMED_AND_PREDICTIONS",
                                    instances=1, transformerInstances=1)
```

Create Model Serving to be used

```
Inferring model server from artifact files: TENSORFLOW_SERVING
Creating serving ccfraudmodel for artifact /Projects/card_fraud_detection//Models/ccfraudmodel ...
Serving ccfraudmodel successfully created
```

```
# List all available servings in the project
for s in serving.get_all():
    print(s.name)
```

```
ccfraudmodel
```

```
# Get serving status
serving.get_status(SERVING_NAME)
```

```
'Stopped'
```

Start Model Serving Server

```
if serving.get_status(SERVING_NAME) == 'Stopped':
    serving.start(SERVING_NAME)
```

```
Starting serving with name: ccfraudmodel...
Serving with name: ccfraudmodel successfully started
```

```
import time
while serving.get_status(SERVING_NAME) != "Running":
    time.sleep(5) # Let the serving startup correctly
time.sleep(5)
```

Starting Serving Server for Model Serving

Sample credit card numbers

```
import hsfs

connection = hsfs.connection()
fs = connection.get_feature_store()
```

```
Connected. Call `.close()` to terminate connection gracefully.
```

```
td_meta = fs.get_training_dataset("card_fraud_model", 1)
```

Test using Training Dataset

Get serving vector and send to Prediction Requests to the Served Model using Hopsworks REST API

```
import numpy as np
TOPIC_NAME = serving.get_kafka_topic(SERVING_NAME)
print("Topic: " + TOPIC_NAME)
```

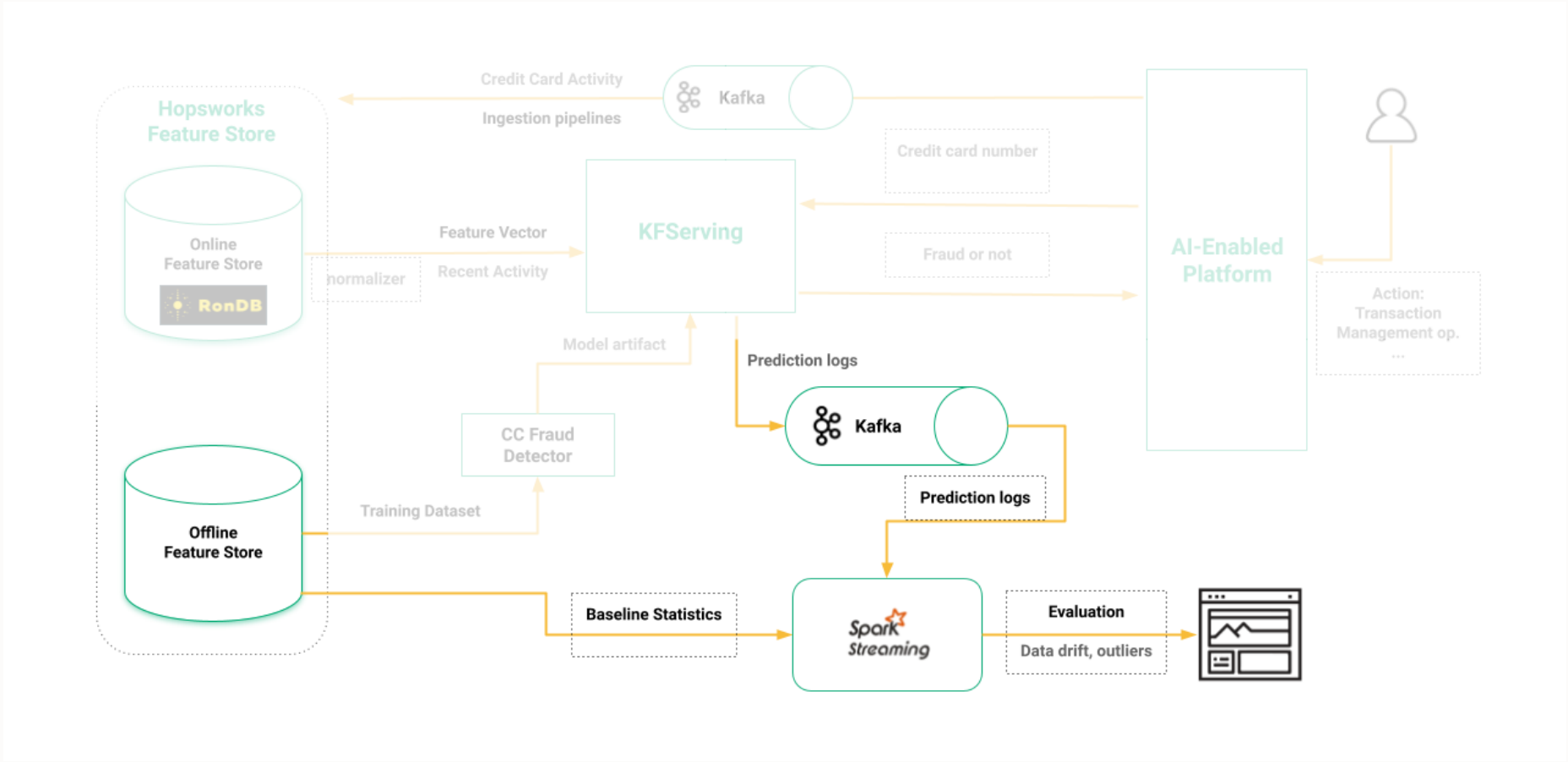
```
Topic: credit_card_prediction_logs
```

```
i=0
for cc_num in cc_nums_inputs['fg0_cc_num']:
    data = { "signature_name": "serving_default", "instances": [{'cc_num': int(cc_num)}] }
    response = serving.make_inference_request(SERVING_NAME, data)
    if i % 2000 == 0:
        print(response)
    i+=1
```

```
{'predictions': [2.20478344]}
{'predictions': [0.955848038]}
{'predictions': [2.77835608]}
{'predictions': [1.0319711]}
{'predictions': [1.04669809]}
```

Using Model Serving and check inference

7_inference_analysis



Start the Monitoring Job

To achieve this, we need to create a streaming job using the jar file `job-1.0-SNAPSHOT.jar` located together with the demo notebooks and the following job configuration:

Start the Monitoring Job

- **Main class name:** `io.hops.ml.monitoring.job.Monitor`
- **Default arguments:** `--conf card_fraud_monitoring_job_config.json`

Then, in advance configuration add the json file with name `card_fraud_monitoring_job_config.json` stored together with the demo notebooks. You can customize the monitoring job by modifying this configuration file. Among other things, you can define which statistics to compute, the algorithms for detecting data drift or where to store the resulting analysis.

Once the monitoring job is running and the previous notebook has already made some predictions, we can access the statistics, outliers and drift detection that are continuously computed.

```
from hops import hdfs
import pyarrow.parquet as pq
from hops import kafka
from hops import tls
from confluent_kafka import Producer, Consumer
import json

import pandas as pd
pd.options.display.max_columns = None
pd.options.display.max_rows = None
pd.set_option('display.max_colwidth', None)
```

Inference Statistics

Read inference statistics from parquet files

Check Monitoring in Parquet Files

```
MONITORING_DIR = "hdfs:///Projects/" + hdfs.project_name() + "/Resources/CardFraudDetection/Monitoring/"
LOGS_STATS_DIR =  MONITORING_DIR + "credit_card_activity_stats-parquet/"
hdfs.mkdir(LOGS_STATS_DIR)
```

```
credit_card_activity_stats = spark.read.parquet(LOGS_STATS_DIR + "*.parquet")
```

```
credit_card_activity_stats.createOrReplaceTempView("credit_card_activity_stats")
```

```
desc_stats_df = spark.sql("SELECT window, feature, min, max, mean, stddev FROM credit_card_activity_stats ORDER BY window")
distr_stats_df = spark.sql("SELECT feature, distr FROM credit_card_activity_stats ORDER BY window")
corr_stats_df = spark.sql("SELECT window, feature, corr FROM credit_card_activity_stats ORDER BY window")
cov_stats_df = spark.sql("SELECT feature, cov FROM credit_card_activity_stats ORDER BY window")
```

Descriptive statistics

```
print(desc_stats_df.show(6, truncate=False))
```

```
+------------------------------------------+-----------------+------------------+------------------+----+------+
|window                                    |feature          |min               |max               |mean|stddev|
+------------------------------------------+-----------------+------------------+------------------+----+------+
|{2021-06-22 13:00:44, 2021-06-22 13:00:50}|num_trans_per_1h |1.0005            |2.4701648761537442|0.05|0.27  |
|{2021-06-22 13:00:44, 2021-06-22 13:00:50}|avg_amt_per_12h  |1.0005            |1.3102870163717324|0.01|0.1   |
|{2021-06-22 13:00:44, 2021-06-22 13:00:50}|avg_amt_per_1h   |1.00099           |5.791895          |0.17|1.35  |
|{2021-06-22 13:00:44, 2021-06-22 13:00:50}|avg_amt_per_10m  |1.001             |1.0015            |0.0 |0.05  |
|{2021-06-22 13:00:44, 2021-06-22 13:00:50}|stdev_amt_per_12h|1.0105183333333334|2.2248025         |0.04|0.33  |
|{2021-06-22 13:00:44, 2021-06-22 13:00:50}|stdev_amt_per_1h |1.000745          |4.4910049999999995|0.12|0.85  |
```

Distributions

Distributions, Correlations

```
print(distr_stats_df.show(6, truncate=False))
```

```
+----------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------+
|feature         |distr
|
+----------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------+
|num_trans_per_1h |{1.0016000509262085 -> 0.0, 1.0010000467300415 -> 0.0, 1.001900053024292 -> 0.0, 1.0022000551223755 -> 0.0, 1.00130004882812
5 -> 0.0}                                  |
|avg_amt_per_12h  |{1.5285710096359253 -> 0.0, 2.5619930028915405 -> 0.0, 3.0787039995193481 -> 0.0, 1.0118600130081177 -> 2.0, 2.0452820062637
329 -> 0.0}                                |
|avg_amt_per_1h   |{1.0005899667739868 -> 24.0, 5.1820502281188965 -> 1.0, 4.3457581758499 1456 -> 1.0, 2.6731740713119 5068 -> 0.0, 3.5094661235
8093262 -> 2.0, 1.83688201904296874 -> 1.0}|
|avg_amt_per_10m  |{2.0005550384521484 -> 0.0, 3.0005550384521484 -> 0.0, 4.0005550384521484 -> 0.0, 5.0005550384521484 -> 0.0, 1.0005550384521
484 -> 29.0}                               |
|stdev_amt_per_12h|{3.16779012680053708 -> 0.0, 1.0057396888 73291 -> 24.0, 2.62727751731872556 -> 0.0, 1.54625229835510252 -> 3.0, 2.0867649078
3691404 -> 2.0}                            |
|stdev_amt_per_1h |{2.19480900764465332 -> 0.0, 3.38911805152893064 -> 1.0, 2.79196352958679198 -> 1.0, 1.000499963760376 -> 26.0, 1.5976544857
0251466 -> 0.0}                            |
+----------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------+
only showing top 6 rows

None
```

Correlations

```
print(corr_stats_df.show(6, truncate=False))
```

Outliers and Data Drift Detection (kafka)

```
def get_consumer(topic):
    config = kafka.get_kafka_default_config()
    config['default.topic.config'] = {'auto.offset.reset': 'latest'}
    consumer = Consumer(config)
    consumer.subscribe([topic])
    return consumer
```

```
def poll(consumer, n=2):
    df = pd.DataFrame([])
    for i in range(0, n):
        msg = consumer.poll(timeout=5.0)
        if msg is not None:
            value = msg.value()
            try:
                d = json.loads(value.decode('utf-8'))
                df_msg = pd.DataFrame(d.items()).transpose()
                df_msg.columns = df_msg.iloc[0]
                df = df.append(df_msg.drop(df_msg.index[[0]]))
            except Exception as e:
                print("A message was read but there was an error parsing it")
                print(e)
    return df
```

Outliers detected

Detect Outliers

```
outliers_consumer = get_consumer("credit_card_activity_outliers")
```

Data drift detected

Detect Data Drift

```
drift_consumer = get_consumer("credit_card_activity_drift")
```

```
drift = poll(drift_consumer, 10)
```

```
drift.head(5)
```

```
0                                                                  window  \
1  {'start': '2021-06-17T14:13:36.000Z', 'end': '2021-06-17T14:13:42.000Z'}
1  {'start': '2021-06-17T14:13:36.000Z', 'end': '2021-06-17T14:13:42.000Z'}
1  {'start': '2021-06-17T14:13:36.000Z', 'end': '2021-06-17T14:13:42.000Z'}
1  {'start': '2021-06-17T14:13:36.000Z', 'end': '2021-06-17T14:13:42.000Z'}
1  {'start': '2021-06-17T14:13:36.000Z', 'end': '2021-06-17T14:13:42.000Z'}

0            feature            drift     value             detectionTime
1   num_trans_per_1h      wasserstein  0.733333  2021-06-17T14:15:54.584Z
1   num_trans_per_1h  kullbackLeibler  0.972924  2021-06-17T14:15:54.584Z
1   num_trans_per_1h    jensenShannon  0.282642  2021-06-17T14:15:54.584Z
1  num_trans_per_12h      wasserstein         2  2021-06-17T14:15:54.584Z
1  num_trans_per_12h  kullbackLeibler   1.60944  2021-06-17T14:15:54.585Z
```

Thank You

DevRel Team

ORACLE