EC2 setup

Kirtikumar Shinde

Exported on 07/21/2020

Table of Contents

If the client is handover the docker, then following steps would be needed to setup conda environment and spark on the client environment.

1 Conda Setup

- Run the commands from the below script to download and setup anaconda environment.
- This would setup the conda environment for the particular user.
- Run the commands one by one.
- After the command is complete, run the `conda --version` to make sure anaconda is installed.

```
#!/bin/bash
set -e

## Configurations
# Anaconda installtion prefix
PREFIX="$HOME/anaconda3"
# Path for conda binary
CONDA_BIN="$PREFIX/bin/conda"
# Path for python binary
CONDA_PY="$PREFIX/bin/python"


## Install Anaconda
# Downoad the latest version of Anaconda installation file
echo -e "\nDownloading..."
curl -f# -o "$HOME/install_anaconda.sh" https://repo.anaconda.com/archive/$(curl -sf https://
repo.anaconda.com/archive/ | grep -Eo "Anaconda3-20(1[8-9]|2[0-9])\.[0-9]{1,2}-Linux-x86_64\.sh" | head -1)

# Install Anaconda
echo -e "\nInstalling..."
bash "$HOME/install_anaconda.sh" -b -f -p $PREFIX
rm -f "$HOME/install_anaconda.sh"


## Config Anaconda
# Update conda to the latest version
$CONDA_BIN update -y conda

# Initialize conda for shell interaction
$CONDA_BIN init bash

# Set conda base env Python version to 3.7 then create the kernel for Jupyter
echo -e "\nSetting up Conda environment and Jupyter kernel..."
$CONDA_BIN install -y python=3.6
$CONDA_PY -m ipykernel install --user --name python3 --display-name "Python 3 (conda)"

echo -e "\nInstall Succeeded!"
```

2 Spark Installation

- Run the below commands to download the Spark. If you are looking for different version of Spark and Hadoop - change them accordingly.
- Make sure commands are run one by one.

```
##Spark Setup:
# make sure you are logged in as root to run below commands

export SPARK_VERSION=2.4.0
export HADOOP_VERSION=3.2.1

apt-get update -y
apt-get install default-jre -y
apt-get install default-jdk -y


mkdir /usr/local/spark/
chmod -R 755 /usr/local/spark/

cd /usr/local/spark/

curl https://archive.apache.org/dist/spark/spark-$SPARK_VERSION/spark-$SPARK_VERSION-bin-without-hadoop.tgz
--output spark-$SPARK_VERSION-bin-without-hadoop.tgz
tar -xvzf spark-$SPARK_VERSION-bin-without-hadoop.tgz

curl https://archive.apache.org/dist/hadoop/common/hadoop-$HADOOP_VERSION/hadoop-$HADOOP_VERSION.tar.gz --
output hadoop-$HADOOP_VERSION.tar.gz
tar -xvf  hadoop-$HADOOP_VERSION.tar.gz

rm -rf spark-$SPARK_VERSION-bin-without-hadoop.tgz
rm -rf hadoop-$HADOOP_VERSION.tar.gz


ln -s /usr/local/spark/ /opt/spark
```

3 Spark configuration

- Edit the `spark-default.conf` and append the contents as below in the file.
- If the file does not exist, create this file from the template file available in the `/usr/local/spark/spark-2.4.0-bin-without-hadoop/conf/` directory.
- Make sure you only add the following lines at the end of the file. Please note - depending on the size of memory on EC2 you would have change the values for:
    - spark.driver.memory - should be set to around 70% of total RAM

```
spark.hadoop.fs.s3a.impl org.apache.hadoop.fs.s3a.S3AFileSystem
spark.hadoop.fs.s3a.connection.maximum 100
spark.jars.packages org.apache.hadoop:hadoop-aws:3.1.1
spark.driver.memory 180g
spark.driver.maxResultSize 10g
spark.driver.memoryOverhead 5g
spark.debug.maxToStringFields 10000
```

4 User profile setup

- Edit the user's .bashrc file using the command `vi ~/.bashrc' and add the following lines. Make sure these lines are added the end of the file without impacting any existing commands in the file.

```
export SPARK_VERSION=2.4.0
export HADOOP_VERSION=3.1.1
export PATH="$PATH:/opt/spark/spark-$SPARK_VERSION-bin-without-hadoop/bin"
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
export SPARK_HOME=/opt/spark/spark-$SPARK_VERSION-bin-without-hadoop
export SPARK_DIST_CLASSPATH=/opt/spark/hadoop-$HADOOP_VERSION/etc/hadoop:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/common/lib/*:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/common/*:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/hdfs:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/hdfs/lib/*:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/hdfs/*:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/mapreduce/lib/*:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/mapreduce/*:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/yarn:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/yarn/lib/*:/opt/spark/hadoop-$HADOOP_VERSION/share/hadoop/yarn/*
```

5 QB python packages

- On the home directory, create a folder as 'qb_python_packages'.
- Copy the provided QB python packages there. The directory should have following two packages:
    - PAI

6 Python libraries

- First activate the conda environment which you have created.
- Download the GIT repository for the project and go to that directory. Alternatively you can also download the list from page 1.2 Python package list(see page 3) and add it to 'requirements.txt'.
- Before install libraries make sure the QB packages are copied manually to the home directory of user in the folder 'qb_python_packages'.
- *These steps would report some warnings/error about package versions. Please ignore these errors or warnings.*
- Assumptions:
    - It is 'base' conda environment.
    - The PAI package path is in the '/home/unix_user/qb_python_packages'.
    - The git repo is copied under the folder '/home/unix_user/repo/project_baldur'

```
# Activate conda environment
conda activate base

# Go to project directory
cd /home/unix_user/repo/project_baldur

# Run the python package installation command
pip install -r src/requirements.txt --find-links /home/unix_user/qb_python_packages
```

7 Jupyter kernel setup

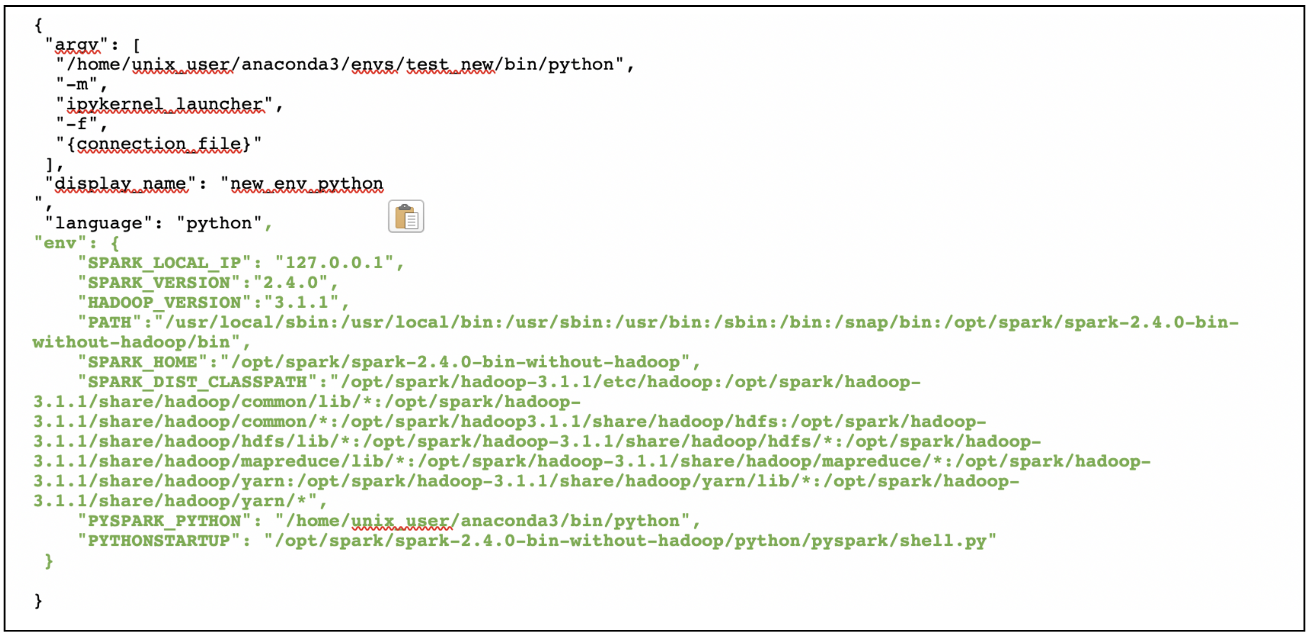
- First find out the Jupyter kernel file name using the command 'jupyter kernelspec list'.
- Then edit that file to add following variables in it.
- *Note:*
    - *Only add lines from 'env' onwards.*
    - *Code below show the final file.*
    - *Image below shows* ***only the text*** *which you have to add including the* ***comma****.*
- Assumptions:
    - Kernel filename output from the command is '/home/unix_user/.local/share/jupyter/kernels/python3/kernel.json'

7.1 Code to be added:

```
{
"argv": [
"/home/unix_user/anaconda3/envs/test_new/bin/python",
"-m",
"ipykernel_launcher",
"-f",
"{connection_file}"
],
"display_name": "new_env_python",
 "language": "python",
"env": {
 "SPARK_LOCAL_IP": "127.0.0.1",
 "SPARK_VERSION":"2.4.0",
 "HADOOP_VERSION":"3.1.1",
 "PATH":"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/opt/spark/spark-2.4.0-bin-without-hadoop/bin",
 "SPARK_HOME":"/opt/spark/spark-2.4.0-bin-without-hadoop",
 "SPARK_DIST_CLASSPATH":"/opt/spark/hadoop-3.1.1/etc/hadoop:/opt/spark/hadoop-3.1.1/share/hadoop/common/lib/*:/opt/spark/hadoop-3.1.1/share/hadoop/common/*:/opt/spark/hadoop3.1.1/share/hadoop/hdfs:/opt/spark/hadoop-3.1.1/share/hadoop/hdfs/lib/*:/opt/spark/hadoop-3.1.1/share/hadoop/hdfs/*:/opt/spark/hadoop-3.1.1/share/hadoop/mapreduce/lib/*:/opt/spark/hadoop-3.1.1/share/hadoop/mapreduce/*:/opt/spark/hadoop-3.1.1/share/hadoop/yarn:/opt/spark/hadoop-3.1.1/share/hadoop/yarn/lib/*:/opt/spark/hadoop-3.1.1/share/hadoop/yarn/*",
 "PYSPARK_PYTHON": "/home/unix_user/anaconda3/bin/python",
 "PYTHONSTARTUP": "/opt/spark/spark-2.4.0-bin-without-hadoop/python/pyspark/shell.py"
 }

}
```

7.2 Image of the changes:



```
{
 "argv": [
  "/home/unix_user/anaconda3/envs/test_new/bin/python",
  "-m",
  "ipykernel_launcher",
  "-f",
  "{connection_file}"
 ],
 "display_name": "new_env_python
",
 "language": "python",
"env": {
    "SPARK_LOCAL_IP": "127.0.0.1",
    "SPARK_VERSION":"2.4.0",
    "HADOOP_VERSION":"3.1.1",
    "PATH":"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/opt/spark/spark-2.4.0-bin-
without-hadoop/bin",
    "SPARK_HOME":"/opt/spark/spark-2.4.0-bin-without-hadoop",
    "SPARK_DIST_CLASSPATH":"/opt/spark/hadoop-3.1.1/etc/hadoop:/opt/spark/hadoop-
3.1.1/share/hadoop/common/lib/*:/opt/spark/hadoop-
3.1.1/share/hadoop/common/*:/opt/spark/hadoop3.1.1/share/hadoop/hdfs:/opt/spark/hadoop-
3.1.1/share/hadoop/hdfs/lib/*:/opt/spark/hadoop-3.1.1/share/hadoop/hdfs/*:/opt/spark/hadoop-
3.1.1/share/hadoop/mapreduce/lib/*:/opt/spark/hadoop-3.1.1/share/hadoop/mapreduce/*:/opt/spark/hadoop-
3.1.1/share/hadoop/yarn:/opt/spark/hadoop-3.1.1/share/hadoop/yarn/lib/*:/opt/spark/hadoop-
3.1.1/share/hadoop/yarn/*",
    "PYSPARK_PYTHON": "/home/unix_user/anaconda3/bin/python",
    "PYTHONSTARTUP": "/opt/spark/spark-2.4.0-bin-without-hadoop/python/pyspark/shell.py"
 }

}
```

8 Environment Test

- Test the environment to make sure everything is working.
- To do this open jupyter notebook. Make sure the same user for which above setting is done is logging to notebook.
- Make sure kernel is selected as the one created in step "Conda setup".
- In Jupyter, select 'shutdown kernel'. Give about 30 seconds and then select 'restart kernel'.
- This will make sure spark variables above are picked up by kernel.
- Then in the notebook write the below code. Make sure commands are written in different cells as marked in the code below, so each cell is run sequentially before other.
- The output from reading the data should show some data to confirm that environment is working fine.
- Assumptions:
    - S3 path of the data is: s3://rwe-study-data/data/data/a_raw/lu_procedure/*
    - Kernel in jupyter is selected - "Python 3 (conda)"

```
# Cell 1 - init spark
import findspark
findspark.init()

# Cell 2 - Create spark session
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()

# Cell 3 - Check spark session is set
spark

# Cell 4 - import kedro
import pyspark.sql
from kedro.io import DataCatalog
from kedro.contrib.io.pyspark import SparkDataSet

# Cell 5 - Initialize file using kedro
spark_proc_ds = SparkDataSet(
    filepath="s3a://rwe-study-data/data/data/a_raw/lu_procedure/*",
    file_format="csv",
    load_args={"header": False, "inferSchema": False, "sep":"|",
      'schema':
        '''   category_dtl_cd          STRING,
   category_dtl_code_desc   STRING,
   category_genl_cd         STRING,
   category_genl_code_desc  STRING,
   proc_cd                  STRING,
   proc_desc                STRING,
   proc_end_datex           STRING,
   proc_typ_cd              STRING,
   proc_end_date            LONG'''},
    save_args={"sep": ",", "header": True}
)
catalog = DataCatalog({'raw_lu_procedure' : spark_proc_ds})


# Cell 6 - read the data
df_proc = catalog.load('raw_lu_procedure')

# Cell 7 - Display the
df_proc.show()

# Cell 8 - count rows
df_proc.count()
```

9 Attachments

pai-0.18.0-py3-none-any.whl