Virginia Tech

Bradley Department of Electrical and Computer Engineering

ECE 5984 Data Engineering Project
Fall 2023

Lab 0 – Setting up your Environment

In this Lab, you will set up the environment needed for the upcoming assignments (consisting of labs and homework) and the Project.

Please note that all assignments will be submitted in the respective Canvas section, while the Project will be in GitHub.

Step 0: Connect to your console with your AWS IAM user credentials

- You should have received the following AWS login information via email:
  - Account ID (12 digits)
  - IAM user name
  - Password
    - Note that the password will be a one time password and then after logging in the first time you will need to setup a new password that meets the following requirements:
    - Must be at least 32 characters long
    - Must include at least one uppercase letter (A-Z)
    - Must include at least one lowercase letter (a-z)
    - Must include at least one number (0-9)
    - Must include at least one non-alphanumeric character (! @ # $ % ^ & * ( ) _ + - = [ ] { } |')
- If for any reason you have not received this information please contact the TA or the Professor.
- Use the above information to log into you AWS console by going to https://aws.amazon.com/ and pressing sign in on the top right

Step 1: Setting up the coding Environment

The following steps need to be followed to set up the recommended coding environment on your local machine for this class. We will be primarily using Python v3.8 along with PyCharm as our coding IDE. Students are free to use a different IDE (eg. Jupyter Notebooks, Vscode) but expect no help setting up or using said different IDE if you choose to work with it.

Pycharm also has an integration with GitHub steps which are provided. It might be the case that a different IDE does not and if that is the case a GitHub repository of your project has to be maintained separately.

Install Python + IDE (windows)

1. To download and install Python, visit the official Python website and choose your version. https://www.python.org/downloads/
2. Once the download is completed, run the .exe file to install Python. Now click on Install Now and follow the installation instructions.
3. To download PyCharm visit the website https://www.jetbrains.com/pycharm/download/ and Click the "DOWNLOAD" link under the Community Section.
4. Once the download is complete, run the .exe file to install PyCharm. The setup wizard should have started. Click "Next" and follow the installation instructions.

Start a New PyCharm Project

1. After installation, you can create a new project. Choose a location for your project and expand "Python Interpreter: New Virtualenv environment"
2. Select "New environment using" and select Virtualenv from the drop-down menu
3. Set your Base interpreter to the Python version installed previously and click "create". This will create a Python virtual environment for a particular project
4. Next, if you want to add libraries to the particular project just created go to File → Settings → Project:'project name' → Python Interpreter
5. Click the "+" button, search the required library (eg. pandas) and click install package

Share project on GitHub

1. Go to https://github.com/ and sign up for an account
2. In PyCharm open the project you would like to share
3. Go to File → Settings → Version Control → Github
4. Select "Add account" and then login via Github
5. A web browser should open. Log into your Github account using it and press "Apply"
6. Click VCS → Share Project on Github
7. Set the required information like repository name, public/private, etc, and click share

Step 2: Accessing the cloud infrastructure

The following steps are required to access and use the cloud infrastructure

Accessing the EC2 instance

(follow along video: get console access)

1. Log into your AWS console (shown in step 0)
2. On the top search bar search “EC2” and click the top result
3. Click the running instances button
4. Select the EC2 listed there. For this class its is named “ECE5984-F23”
5. Press the Connect button on the top right
6. Go to EC2 instance connect and enter the username “student”
7. Press connect

You should now have terminal access to the EC2 instance

Spin up and exit out of a docker container

(follow along video: spin up and exit out of container)

You will primarily be working by spinning up docker containers on the EC2 instance with the class pipeline loaded inside of it. Following are the steps to deploy a new container and the exiting out of it

1. Access the EC2 instance (shown above)
2. Insert the following command:

```
docker run --rm -it --entrypoint bash -v /home/ubuntu/efs-mount-point/students/<pid>/root:/root -p 8080-8131:8080 --name <pid> <image-name-or-id>
```

   - Replace `<pid>` with your own vt username pid only without “@vt.edu”
   - Replace `<image-name-or-id>` with the docker image name which is `pipeline:latest`
   - For example a student with the pid test would have to run the following command: `docker run --rm -it --entrypoint bash -v /home/ubuntu/efs-mount-point/students/test/root:/root -p 8080-8131:8080 --name test pipeline:latest`

After running this command you are now inside the docker container. In order to exit out of the container and get back to the EC2 just type the command `exit`

Please note the container will destroy itself after you exit the container. You must ensure that you exit the container via this command and the container destroys itself. This can be checked by running the command `docker ps` command on the EC2 terminal and not seeing a container with your pid on it still running

Setup Airflow GUI

(follow along video: setup airflow)

These steps need to be performed once for you to setup your airflow inside the docker container

1. Spin up your docker container (shown previously)
2. Inside the docker container run the command `airflow standalone`
3. After airflow has booted up and you get the message airflow is ready stop it by pressing ctrl-C. wait for it to shut down. If it takes longer than 10 mins press ctrl+C again.
4. Type command `ls` to list the directories in your current working folder
5. Navigate to the created folder named 'airflow' by using the command `cd airflow`
6. Make a folder called 'dags' and 'plugins' within in it using the command

   ```
   mkdir dags plugins
   ```
7. Type the command `ls` again and there should also be a file named standalone_admin_password.txt with your password in it with "user" as "admin"
8. To display this password type command `cat standalone_admin_password.txt`
9. Note down the username and password down for future use on your local machine.

Launching and access the airflow GUI

(follow along video: launching and accessing airflow GUI)

1. Spin up your docker container (shown previously)
2. Inside the docker container run the command `airflow standalone`
3. Open a new browser tab and access the EC2 instance (shown previously)
4. Run the command `docker ps`
5. `Note down the 4-digit port number under the PORTS column to the left of the arrow`
6. `Open another new browser tab and log into your AWS console`
7. `Search for "EC2" and click the top result`
8. Click the running instances button and select the "ECE5984-F23" instance
9. Note down the Public IPv4 DNS address and paste it onto a new tab of your browser
10. Add ":portnumber" at the end of the address where portnumber is the 4 digit port number noted in step 5 and press enter
11. You should be greeted with Apache airflows login screen
12. Login using the admin username and password from step 9 from the setting up airflow section above