Example of Using UCSD DSMLP: Spark Cluster

Yuhao Zhang (yuz870@eng.ucsd.edu)

Fall 2021

1 Getting started

1.1 How to read and copy commands in this section

1. In this section, we have three different hosts where you can type commands: your computer (`local`), the login node (`dsmlp-login`), and Spark master node (`spark-master`). All shell commands will be given you in the format of:

   ```
   @<host>: <commands>
   ```

   For instance, if we would like you to list the directory on your computer, the command would be:

   ```
   @local: ls
   ```

   In this scenario, what you need to do is open a terminal (Linux and OS X users) or a PowerShell (Windows users), copy-paste `ls`, and execute it.

2. On the other hand, if you are given a command like:

   ```
   @dsmlp-login: ls
   ```

   This means the command `ls` needs to be executed on the login node. You need to first SSH into it and then execute the command. We will show you how to do the SSH.

3. Sometimes you may encounter angular brackets `<XXX>`; in this situation, you will need to substitute it with the desired value. Do **not** leave the brackets. For example, the following command

   ```
   @local: echo <pid>
   ```

   You need to put your pid in the command, and the command you run would become (assuming your pid is a10000000):

   ```
   @local: echo a10000000
   ```

4. Be extra cautious when copy-pasting commands. Quite often copying from pdf will lead to missing/extra characters. If anything does not run, please first ensure your pasted command is exactly the same as listed in this document.

1.2 SSH into the login node

First, use your ETS account and password to sign into the login node via SSH from your machine:

```
@local: ssh <ETS account>@dsmlp-login.ucsd.edu
```

Your ETS account name is usually the same as your UCSD email name. If you have trouble finding it or if you forget the password, use ETS Account Lookup[1].

[1] `<https://sdacs.ucsd.edu/~icc/index.php>`

1.3 Prepare the dev-kit

You only need to do this once. In the login node's shell, clone the repo prepared to you by:

```
@dsmlp-login:
    git clone https://github.com/makemebitter/dsmlp-dev-tool.git
```

This should create a folder named `dsmlp-dev-tool` in your home directory.

1.4 Launch the cluster

1. In the login node's shell, go to the home directory of dev-kit:

   ```
   @dsmlp-login: cd ~/dsmlp-dev-tool
   ```

2. Create the cluster via:

   ```
   @dsmlp-login: ./cluster-manager.sh create
   ```

   Wait until the cluster is up and it will output instructions similar to below:

   ```
   => Successfully initiated the Spark cluster
   => Next create a SSH tunnel from your personal computer using the following command:
   ssh -N -L 127.0.0.1:8888:127.0.0.1:XXX -L 127.0.0.1:8080:127.0.0.1:XXX -L
       127.0.0.1:4040:127.0.0.1:XXX XXX@dsmlp-login.ucsd.edu
   => Link to PySpark/Jupyter UI: http://127.0.0.1:8888?token=XXXXXXXXXXX
   ```

3. Copy paste the port-forwarding command printed to your console to a new shell on **your computer** and provide your UCSD password when prompted. You will get a new command every time you create a Spark cluster.

   ```
   @local: ssh -N -L 127.0.0.1:8888: ... ... ... ... @dsmlp-login.ucsd.edu
   ```

   The ssh command will continue to run in the foreground without retuning back to the command prompt. This is in place to avoid port clashes when you create new Spark clusters later. If you decide to send the command to background by putting a **-f** flag, you will have to manually find the process and kill it before you run a new SSH tunnel command.

4. (Optional) In your browser, connect to the following.
   - Jupyter notebook:.

     ```
     http://127.0.0.1:8888/notebooks/?token=<token>
     ```

   - Spark cluster manager UI: Where you will see the available resources in your Spark cluster.

     ```
     http://127.0.0.1:8080
     ```

   - Spark job UI: Where you will the stages of your currently running Spark job. This will be active only when your have created a Spark session in PySpark (i.e., after you run the imports and initialization code blocks in the provided Jupyter notebook).

     ```
     http://127.0.0.1:4040
     ```

5. Your login node's home directory is mounted on your cluster nodes. So only your modifications to `$HOME` is kept, and no files are stored in the cluster.

1.5 SSH into one of the nodes

1. From the shell on the login node, query the node's pod name:

   ```
   @dsmlp-login: kubectl get pods
   ```

   The name would be in the format of `spark-XXX-XXX-XXX`.

2. SSH into the node via

   ```
   @dsmlp-login: kubectl exec -it <spark-XXX-XXX-XXX> bash
   ```

1.6 Delete the cluster

Don't forget to delete the cluster. You should do this whenever you are not using it. ITS will also have a monitoring system to delete your cluster if it has been running for more than 3 hours. You need to manually re-launch should it be terminated.

1. Go to the dev-kit directory from your login node's shell

```
@dsmlp-login: cd ~/dsmlp-dev-tool
```

2. Delete the cluster via

```
@dsmlp-login: ./cluster-manager.sh delete
```