Spark on AWS EC2 Tutorial for Dummies

Litong "Leighton" Dong, Chi-Liang (Daniel) Kuo, and Steven Rea

---

1 Preparation

1.1 In your browser

1.1.1 Access Amazon Web Service

- Go to: `http://aws.amazon.com`
- Click on "Sign In to the Console" in the upper right hand corner.
- Sign up (if necessary) and sign in

1.1.2 Obtain Security Credentials

Notes: Your AWS account is like your computer at a different location, thus requires "username" and "password" to login just like your own computer

- Click on your username in the upper right hand corner
- Click "Security Credential" from the drop-down
- Click on "Continue to Security Credentials" in the pop-up window (if there is any)
- Click "Access Keys (Access Key ID and Secret Access Key)"
- Click "Create New Access Key"
- A download should start automatically; the file would be a .csv (named "rootkey.csv" most likely)
- Open the downloaded file for later use

1.1.3 Obtain a Security Key for EC2

- Click the orange box in the top left corner of web page
- Click on "EC2 (Virtual Servers in the Cloud)" in the upper left corner of the page
- In the navigation window on the left, click "NETWORK & SECURITY"
- Click "Key Pairs"
- Click "Create Key Pair"

- Insert the name you like (for this tutorial we'll call it "Spark_EC2_key_pair") and click "Create"
- A download should start automatically; the file will be a .pem named whatever you named in the previous step. It will be "Spark_EC2_key_pair.pem" in this tutorial.
- Remember where the file is (which will be the "Downloads" folder in this tutorial)

1.1.4 Downloading Spark

- Go to the Spark downloads page: `http://spark.apache.org/downloads.html`
- Select the most current release and remember the version you selected. It will be "1.2.1 (Feb 09 2015)" in this tutorial.
- Select "Source Code [can build several Hadoop versions]" as package type
- Select "Direct Download" as download type
- Click on the .tgz file ("spark-1.2.1.tgz" in this tutorial)
- Save the file to wherever you like (it will be the home directory in this tutorial)

1.2 In your terminal

1.2.1 Setting up with your security credentials

- Go to the folder where you store the .pem file

```
cd ~/Downloads
```

- Move your private key (.pem file) into your .ssh directory

```
mv Spark_EC2_key_pair.pem ~/.ssh
```

- Modify Credentials for private key

```
chmod 400 ~/.ssh/Spark_EC2_key_pair.pem
```

2 Launch and Login

2.1 In your terminal

2.1.1 Setting up for Launch

- Go to the folder where your downloaded spark is

```
cd ~
```

- Unzip the file

  ```
  tar -xvzf spark-1.2.1.tgz
  ```

- Go to the sub folder where your spark ec2 is

  ```
  cd spark-1.2.1/ec2
  ```

- Define variables in your terminal

  CLUSTER_SIZE = The number of slave nodes you want (only one in this tutorial)
  SPARK_VERSION = Your downloaded spark version (1.2.1 in this tutorial)
  REGION = Whichever region of AWS you would like (US West (N. California) in this tutorial)
  `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html`
  SSH_KEY_NAME = Your key pair name ("Spark_EC2_key_pair" in this tutorial, refer to section 1.1.3)
  SSH_KEY_FILENAME = Your key pair file name ("Spark_EC2_key_pair.pem" in this tutorial, refer to section 1.1.3)
  YOUR_NAME = The name of your cluster (ML_Project_Spark_EC2 in this tutorial)
  AWS_ACCESS_KEY_ID = The username for your remote computer (in the file you downloaded (rootkey.csv), refer to section 1.12)
  AWS_SECRET_ACCESS_KEY = The password for your remote computer (in the file you downloaded (rootkey.csv), refer to section 1.12)
  WAIT_TIME = The wait time that the Spark script uses to allow the Amazon instances to start up (300 in this tutorial)

  ```
  export CLUSTER_SIZE=1
  export SPARK_VERSION=1.2.1
  export REGION=us-west-1
  export SSH_KEY_NAME=Spark_EC2_key_pair
  export SSH_KEY_FILENAME=Spark_EC2_key_pair.pem
  export YOUR_NAME=ML_Project_Spark_EC2
  export AWS_ACCESS_KEY_ID=aaGPSODJAasldmvapoALSKF
  export AWS_SECRET_ACCESS_KEY=AncaapANafaAFIVNSOAFAaklasf
  export WAIT_TIME=300
  ```

2.1.2 Cluster Operations

- Launch your cluster

```
./spark-ec2 -k ${SSH_KEY_NAME} -i ~/.ssh/${SSH_KEY_FILENAME}
--region=${REGION} -s ${CLUSTER_SIZE} -w ${WAIT_TIME}
 -v ${SPARK_VERSION} launch sparkvm-${YOUR_NAME}
```

You should be able to see an overwhelming list of messages. If everything works as planned, you would see the last line of the list being "Done!".

- Login to your cluster

```
./spark-ec2 -k ${SSH_KEY_NAME} -i ~/.ssh/${SSH_KEY_FILENAME}
 --region=${REGION} start sparkvm-${YOUR_NAME}
```

- Destroy your cluster

```
./spark-ec2 --region=us-west-1 --delete-groups destroy sparkvm-${YOUR_NAME}
```

- (Optional): stop your cluster
  Notes: Stopping you cluster means that the cluster is created but inactive. You will still be charged for a stopped cluster

```
./spark-ec2 --region=${REGION} stop sparkvm-${YOUR_NAME}
```

- (Optional): start your cluster after you stop it

```
./spark-ec2 -k ${SSH_KEY_NAME} -i ~/.ssh/${SSH_KEY_FILENAME}
 --region=${REGION} start sparkvm-${YOUR_NAME}
```

3 Using Cluster

3.1 Basic Info and Tips

- After you log in, you will be at the root directory of a remote machine. Enter the following:

```
screen -x -RR
```

Here's the best description we found on why we should do this:
"If your wireless connection on your local machine is interrupted, your connection to the Spark cluster can be affected and you might have to start over again. This will put you in a shell session that you'll be able to go back to even if you lose your connection. If you lose the connection then when you connect back to the master node, type "screen -x -RR" again and you should be back where you were."

- Obtain the url for your master node

  `cat ephemeral-hdfs/conf/masters`

- Obtain the url for your slave nodes

  `cat ephemeral-hdfs/conf/slaves`

- You can learn about your remote machines in your *browser* using the url you obtained above

  `http://<master_url>:8080` – general information about the Spark cluster
  `http://<master_url>:4040` – information about the Spark tasks being executed
  `http://<master_url>:50070` – information about the HDFS

3.2 Uploading your python code to AWS S3 and your cluster

3.2.1 In your browser

- Go to: `http://aws.amazon.com` and login to your account
- Click the orange box in the top left corner of web page
- Click "S3 (Scalable Storage in the Cloud)"
- Click "Create Bucket"
- Insert a bucket name and choose the same region as you used for your cluster ("sparkec2" and US West (N. California) in this tutorial)
- Click "Upload" and use default options to upload your .py file ("Spark_EC2.py in this tutorial)
- After successfully upload the file, select the file and click "Properties" in the top right corner
- Copy the "Link" on the third line ("https://s3-us-west-1.amazonaws.com/sparkec2/Spark_EC2.py" in this tutorial)
- Click "Permissions"
- Click "Add more permissions"
- Select "Everyone" and check "Open/Download" box
- Click "Save"

3.2.2 In your terminal

- Go to the directory you want to store the file

  ```
  cd ~
  ```

- Copy the file from S3 with the link you get from section 3.2.1

  ```
  wget https://s3-us-west-1.amazonaws.com/sparkec2/Spark_EC2.py
  ```

  (After copying, don't forget to change the file permission in your S3 permission page)

4 Running your code

4.1 In your terminal

- Go to your root directory

  ```
  cd ~
  ```

- Run code using pyspark

  ```
  spark/bin/pyspark Spark_EC2.py
  ```

5 NOTE

Always remember to destroy your cluster after you are done!