Instructions to deploy applications to your spark cluster

by jfigueroa@pro-pair.com

Install zsh

```
sudo yum install zsh
```

The .pem File

Before you can provision a new spark cluster from the command line, you need to have your `.pem` file.

Pick one `.pem` file you have available and working with AWS and use it. Alternatively go to AWS Console in the EC2 section, create a brand new `.pem` file.

Set your AWS keys before you deploy applications in your cluster

In your AWS account, locate AWS Secrets Manager, create a new Secret, inside your newly created Secret, add the following 2 parameters

```
AWS_ACCESS_KEY_ID=<your-access-key-here>
AWS_SECRET_ACCESS_KEY=<your-secret-key-here>
```

Accept all defaults except the secret name which should be your user like this <your-user>/`aws_credentials`, then follow all steps until your create the Secret.

Modify you config.yaml file

Before you proceed with anything, you should edit/modify your `config.yaml` file. Today the following parameters are passed on to the script:

```
env:
db_name:
db_endpoint:
db_user:
bucket:
aws_region:
table_name:
user:
```

You definitively want to set the `user` value to reflect your user in the `secrets mngr`, otherwise your code will fail at getting the `AWS credentials` for the program to access AWS resources like: s3, sns, etc.

Modify your bootstrap.sh if needed

If you change/add a library in your python script
you should add those libraries to the bootstrap.sh before you create your spark cluster.

The Automation Scripts

You have the following scripts which automates much of the job fo you:

Script Name	Description	Help
spark-cluster.sh	Creates and/or Terminates Clusters	`./spark-cluster.sh --help yes`
spark-deploy.sh	Deploys applications in either `Client or Cluster` modes	`./spark-deploy.sh --help yes`

Spark Cluster Options

To create cluster run

```
./spark-cluster.sh --create-cluster yes --key-name <the-name-of-your-pem-file-here>
```

After *spark-cluster.sh* completed will yield a cluster Id just like this:

```
...{
    cluster_id="j-1ZE9CULSCCZ7N"
   }
...
```

Copy that `cluster_id`, you'll need it for the other scripts.

To delete a cluster just type in the console this:

```
./spark-cluster.sh --delete-cluster yes --cluster-id "j-1ZE9CULSCCZ7N"
```

Deploy applications to your Spark cluster

Applications may be submitted to your Spark cluster in 2 modes: Client mode and Cluster mode.

Client mode is the practice of using your `master` as standalone cluster. This is useful for development/debugging purposes.

Cluster mode is the way you submit applications to run in the entire cluster, including the worker nodes.

In `client` mode the spark cluster expects all code, libraries and config files to be available at your root directory where you're invoking `spak-submit`. Environment variables my be set from within the `master` node in the cluster.

In `cluster` mode the spark clusters expects the `spark-submit` command to include the path to the program, libraries/dependencies, config files and environment variables.

SSH into your spark cluster's master

It's useful to always access your `master` node in a separate terminal console window or tab. To ssh into your `master` node use this:

```
cluster_id="j-2GTE70PYHSUJL"
aws emr ssh --cluster-id $cluster_id \
            --key-pair-file "~/.ssh/propair-etl.pem"
```

Alternatively you may check the `log.log` file to follow execution of applications, like this:

```
tail -f log.log
```

Submit your applications to your Spark cluster

Use *spark-deploy.sh* script to submit your applications to your cluster.

While your program is running, you may switch to your `master` node and take a look at your `log.log` file.