Watts Lab Basic AWS ParallelCluster Setup

Set up and run an AWS ParallelCluster Cluster for the Watts Lab!

Sources

Generic Tutorial I used (from Hama!): https://aws.amazon.com/blogs/storage/building-an-hpc-cluster-with-aws-parallelcluster-and-amazon-fsx-for-lustre/

Setup

Let's get set up!

Local Setup

Set up a system (desktop, laptop, other AWS system) with Git, Python3, and a virtual environment:

```
mkdir wattslab_01
cd wattlabs_01
python -m venv venv3
source venv3/bin/activate
# or on Windows in Git Bash console:
# source venv3/Scripts/activate
# or on Windows in cmd window:
# venv3/Scripts/activate
pip install -U awscli aws-parallelcluster
```

Set Up AWS CLI

Make sure your AWS CLI is configured, with credentials:

```
aws configure
```

AWS Key Pair

You can copy and use a Key Pair you already have, into `~/.ssh/` directory, or create a new one, and save it there, like:

```
aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text > ~/.ssh/MyKeyPair.pem
```

The ParallelCluster Config File

The create a config file, which contains the details for cluster build:

```
mkdir ~/.parallelcluster
```

Create `config` in that directory:

```
[global]
cluster_template = default
update_check = true
```

```
[aws]
aws_region_name = us-east-1

[cluster default]
key_name = hughmac-key
initial_queue_size = 0
max_queue_size = 16
placement_group = DYNAMIC
#cluster_type = spot
master_instance_type = c5.2xlarge
compute_instance_type = c5.4xlarge
vpc_settings = myvpc
ebs_settings = myebs
fsx_settings = myfsx

[vpc myvpc]
vpc_id = vpc-0ca1592c3d07e52ce
master_subnet_id = subnet-02b98d3f9520e1183

[ebs myebs]
shared_dir = /shared
volume_type = gp2
volume_size = 100

[fsx myfsx]
shared_dir = /fsx
storage_capacity = 1200

[aliases]
ssh = ssh {CFN_USER}@{MASTER_IP} {ARGS}
```

Create and Start the Cluster

Create and start the cluster with:

```
pcluster create wattshpc01
```

Note: if you're running multiple clusters, you can specify different configurations in the `config` file, too, by copying the `[cluster default]` section, and replacing `default` with a cluster name.

Now that you've created the cluster, you *no longer need* to run 'create', unless you 'delete' it. From now on, you can 'start' it, instead.

Once it's up, you can log in, and set that up, too!

Master Setup

Log in with:

```
pcluster ssh wattshpc01
```

Now set up Python 3 and your virtual environment on the Master.

```
sudo su -
pyenv global 3.6.9
exit
cd /fsx
git clone YOURREPO
cd YOURREPO_DIR
```

```
python -m venv venv3
source venv3/bin/activate
echo "source /fsx/wattslab_01/venv3/bin/activate" >> ~/.bashrc
pip install -U pip
pip install -U setuptools wheel
pip install -U requirements.txt  # <- if you have one (recommended!)
```

Data from S3/Dropbox/Box to /fsx

First, the data should be in a Watts Lab S3 Bucket, or Dropbox or Box directory. You can put it there, and get it from there, with `aws s3 sync` if you're using S3, or rclone for S3 *or* Dropbox or Box, You can install on Linux with:

```
curl https://rclone.org/install.sh | sudo bash
```

For other OSes, see the rclone download page.

If you need an S3 bucket:

```
aws s3 mb s3://edu-upenn-wattslab-projectA
```

From wherever the files currently live, copy the files *to* the S3 bucket:

```
aws s3 sync /some/local/path/data s3://edu-upenn-wattslab-projectA/data
```

On the Master, copy the files *from* the S3 bucket/Dropbox/Box with something like:

```
aws s3 sync s3://edu-upenn-wattslab-projectA /fsx/YOURREPO_DIR/data
```

Backups to S3/Dropbox/Box from /fsx

Cron job, or manually, do the reverse:

```
aws s3 sync /fsx/YOURREPO_DIR/data s3://edu-upenn-wattslab-projectA/data
```

Or you can use `rclone` similarly to Dropbox/Box.

Jupyter Lab Setup

If it wasn't installed from the requirements.txt, above, on the Master:

```
pip install -U nodejs jupyterlab
jupyter serverextension enable --py jupyterlab --sys-prefix
jupyter notebook --generate-config
```

To start the Jupyter Lab:

```
$ jupyter lab
```

You'll need to start a port forward in another local Terminal:

```
ssh -i ~/.ssh/PCLUSTER_KEY -N -f ec2-user@PCLUSTER_MASTER_IP -L 8888:localhost:8888
```

PCLUSTER_MASTER_IP can be retreaved with `pcluster status CLUSTER_NAME`

Then in a local browser, copy paste the URL to the Notebook:

```
http://localhost:8888/?token=SOME_TOKEN_HERE
```

Doing Work

Start Master (if stopped)

If you have stopped your Master, make sure to start it again:

```
pcluster instances wattshpc01
MasterServer         i-034be2124b2d0dc9e

aws ec2 start-instances --instance-ids i-034be2124b2d0dc9e
```

This will take a minute or so.

Log In and Do Work

Then ssh in:

```
pcluster ssh wattshpc01 -i ~/.ssh/MyKey.pem
cd /fsx/YOURREPO_DIR
```

To submit a batch job via qsub, create a job script (if you don't have it in your repo already), like:

```
#!/bin/bash
#$ -N myjob     # <- whatever name you like
#$ -cwd         # <- run the job from the current directory
#$ -j y         # <- join output and error files
#$ -o output/   # <- put output in directory (cleaner!)
python myscript.py
```

Make an output directory, particularly if you'll be doing big array jobs:

```
mkdir output
```

Submit like:

```
qsub scriptname.sh
```

Note: you will see a warning when running, like:

```
Unable to run job: warning: ec2-user's job is not allowed to run in any queue
```

This can safely be ignored. It's due to the fact that the compute nodes are not running yet.

Note2: this will take some time if the compute nodes are not started yet, up to 10 minutes.

Monitoring

You can monitor job status with `qstat`, or `qstat -j JOBID`.

Stopping the Master

To save money (running instance cost), you can *manually* stop the master. Make sure only the Master is running:

```
pcluster stop wattshpc01
```

Then get its Instance Id:

```
$ pcluster instances wattshpc01
MasterServer         i-034be2124b2d0dc9e
```

Then stop the instance:

```
$ aws ec2 stop-instances --instance-ids i-034be2124b2d0dc9e
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-034be2124b2d0dc9e",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Note: you are still paying for you /fsx/ (Lustre) file system, which is (at the time of this writing in us-east-1) $0.14/GB/month, and the /shared (EBS) file system ($0.10/GB/month), and a bit more for the master instance volume.

Deleting the Cluster

If you won't need the cluster at all for a number of days, you'll most likely want to delete the cluster. It's easy to set up, so why not tear it down, and build it again when needed, to save the Lustre costs?

Back Up First!

Make sure that you manually run the backup one time before deleting the cluster!

Then you can simply:

```
pcluster delete watthpc01
```