

# Cluster creation on AWS

AWS provides means for creating HPC-like clusters; via their `ParallelCluster` tool. This includes freedom to customize compute instances themselves (i.e. the head/master and compute/slave nodes), and also their associated filesystem. Neatly, a cluster can also be set up to increase in size as per computational requirements, and have compute nodes that are either `on-demand` or `preemptive`. Available schedulers include: PBS torque, slurm, SGE and AWS batch.

## Installation and configuration:

In a virtual environment:

```
python3 -m pip install --upgrade pip
pip3 install --user --upgrade virtualenv
virtualenv ~/apc-ve
source ~/apc-ve/bin/activate
pip install --upgrade aws-parallelcluster
```

Now, to use it, you will need to set up the configuration options for AWS (like the access key and region name), and also for pcluster (the cluster template, AWS region VPC and keys):

```
aws configure # You will be prompted to enter your AWS Access Key ID, AWS Secret Access Key and Default region
name. You may keep the Default output format as is
pcluster configure # You will be prompted for a lot of options. Note you also need to specify a key pair that
already exists within your EC2 account to log-in to the cluster. Also, pay attention to the VPC and subnet (i.e.
zoneID) you specify for the cluster
```

And, to actually create a custom cluster:

```
pcluster create <-c pcluster.config> <cluster name> #pcluster.config is a file with specific cluster options,
like scheduler type, instances type, storage, ... etc
```

In the above command:

<cluster name>: use a meaningful name, because you will use this name to terminate the cluster when you are done, and you will also see it as part of the AutoScaling logs (found via: `EC2 Console > Autoscaling > parallelcluster-[cluster\_name]`). Additionally, there can be scenarios where you have different clusters and you need to differentiate between them.

< -c pcluster.config>: optional configuration file that you pass to `pcluster` if you like specific options other than the default (which are found in the file `~/parallelcluster/config` essentially specifying an SGE cluster of: 1 `t2.micro` head node, and 2 `t2.micro` compute nodes (`alinux` OS, with username: `ec2-user`) that are instantiated on demand (i.e., more instances can be created with increase in computational load - up to a default value of 10, or terminated when the cluster is idle).

Once the cluster has been created, the prompt shows the IP address to the master/head node, potentially with your username. You can then ssh into the head node and work as you would do in a normal HPC cluster.

## Handy commands

Cluster termination

```
$ pcluster delete <cluster name>
```

Listing available clusters. The example below, shows 1 cluster named `aws-slurm-cluster`, that it has been created, and that the cluster is created via pcluster version 2.4.0

```
$ pcluster list
aws-slurm-cluster CREATE_COMPLETE 2.4.0
```

Listing information about a specific cluster:

```
$ pcluster status aws-slurm-cluster
Status: CREATE_COMPLETE
MasterServer: RUNNING
MasterPublicIP: 3.220.80.129
ClusterUser: ubuntu
MasterPrivateIP: 172.31.27.132
```

## Example configuration options:

Almost all aspects relating to a normal cluster can be specified (machines, storage, network, .. etc). Additionally, AWS specific options should be specified, like the availability zone or subnet (some instance types are not available in all availability zones), or there is a limit on their number in a specific zone, ... etc

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

[cluster default]
key_name = keys_aws_east_1_aea #This key is associated with my account. Can instead be specified as part of the
pcluster config command
vpc_settings = vpc-b6b9b1d0      #My default vpc
base_os = ubuntu1604
master_instance_type = m5a.2xlarge
compute_instance_type = m5a.24xlarge
initial_queue_size = 100          # Here, I like my cluster to be of constant size,
max_queue_size = 100             # so I set the initial and maximum queue size at the same value of
100; and then I maintain this size.
maintain_initial_size = true      # Having different values for these parameters saves money by
allowing the cluster to terminate idle machines
scheduler = slurm                 # Specifying a slurm cluster, instead of the default SGE

[vpc vpc-b6b9b1d0]
vpc_id = vpc-b6b9b1d0
master_subnet_id = subnet-29c01f61

[global]
cluster_template = default
update_check = true
sanity_check = true              # To have pcluster check the sanity of the options specified (and
that they are congruent with each other)

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

## Known errors:

```
$ pcluster create -c pcluster.config aws-slurm-cluster
Beginning cluster creation for cluster: aws-slurm-cluster
ERROR: The configured max size parameter 36 exceeds the On-Demand capacity on AWS.
Insufficient capacity.
```

This happens when attempting to create a cluster beyond the maximum limit of instances in the specified region/availability zone (i.e. the error is saying that the number requested is above AWS capacity of instances in the region specified). Either changing the number of compute instances needed or changing the availability zone (in the `pcluster.config` file) solves this problem assuming another region has the needed capacity. us-east-1 and specifically in the AZs with the following Zoneld (i.e): use1-az2, use1-az4, use1-az5 generally have large capacity.

```
$ pcluster create -c pcluster.config aws-slurm-cluster
Beginning cluster creation for cluster: aws-slurm-cluster
Creating stack named: parallelcluster-aws-slurm-cluster
Status: ComputeFleet - CREATE_IN_PROGRESS
Rate exceeded
```

The rate in this error message is the rate of calling `describe\_stack` function in the CloudFormation API. It is a transient error though (the cluster will continue to be created, but it will be some time between some instances creation). This can be checked via the `Activity History` in the Autoscaling logs (from EC2 Console > Autoscaling > parallelcluster-{cluster\_name})

## Resources:

parallel cluster official page: <https://docs.aws.amazon.com/parallelcluster/latest/ug/what-is-aws-parallelcluster.html>

github issue with the above errors: <https://github.com/aws/aws-parallelcluster/issues/1151>