Your First HPC Cluster on AWS

AWS ParallelCluster

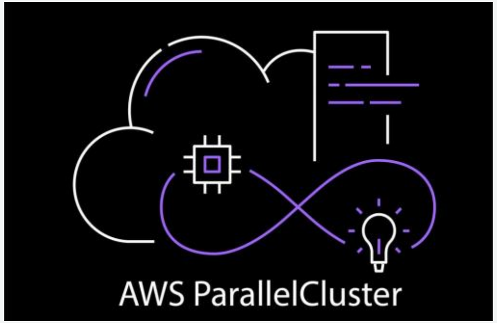
Francesco Ruffino
Sr. HPC Specialist Solution Architect
fruffino@amazon.com

Agenda

- What is AWS ParallelCluster
- Architecture
- Installation
- Configuration
- Your first MPI job
- Advanced configuration
- Q&A

AWS ParallelCluster

AWS **ParallelCluster** is an AWS supported Open Source cluster management tool that makes it easy for you to deploy and manage High Performance Computing (HPC) clusters in the AWS cloud



Built on the Open Source **CfnCluster** project, AWS ParallelCluster enables you to quickly build an HPC compute environment in AWS

https://github.com/aws/aws-parallelcluster

Features

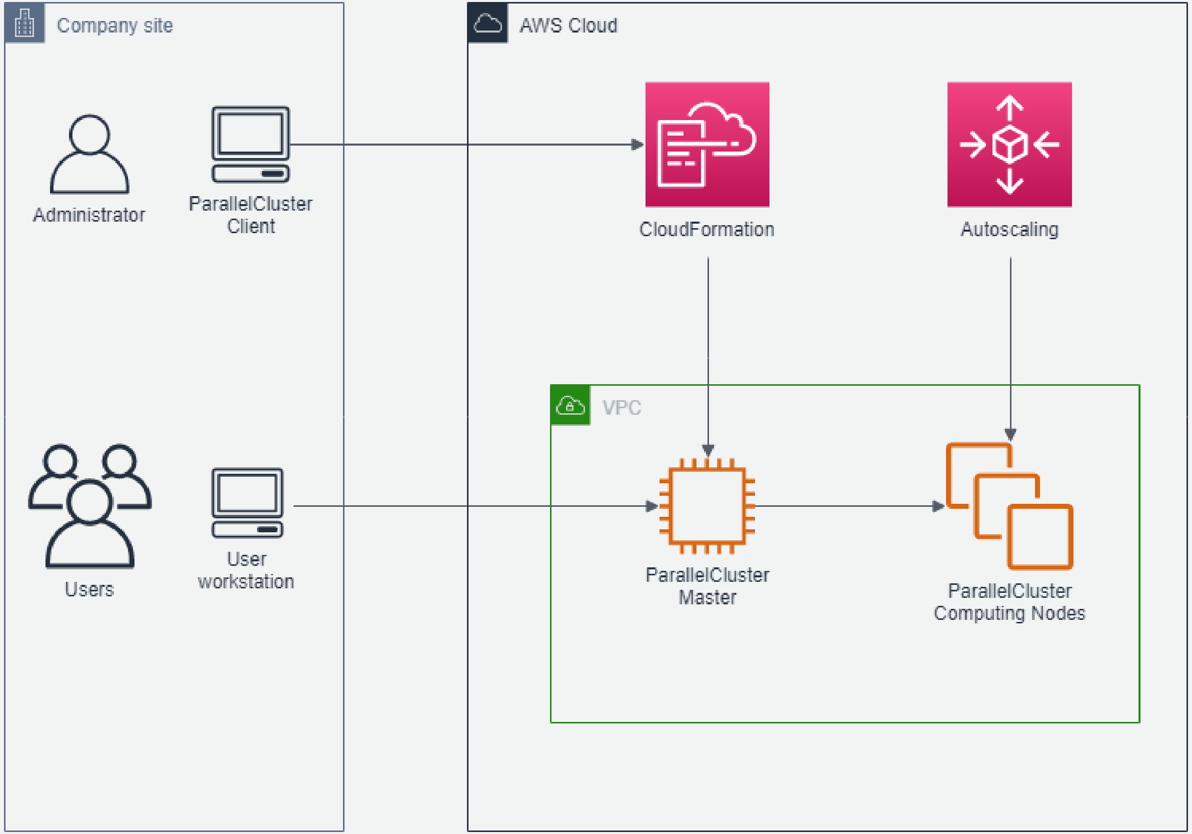
It automatically sets up the required compute resources and a shared filesystem and offers a variety of batch schedulers such as:

- AWS Batch,
- SGE,
- Torque, and
- Slurm
- (and many more in the future...)

AWS ParallelCluster facilitates both quick start proof of concepts (POCs) and production deployments

You can build higher level workflows, such as a Genomics portal that automates the entire DNA sequencing workflow, on top of AWS ParallelCluster

Architecture



How to deploy

- Install ParallelCluster Client
- Configure the Client
- Use the Client commands to deploy your first HPC cluster

AWS ParallelCluster Client

You can run the PC Client on-premises or on AWS

Installation requirements:

- OS:
    - Linux and MacOS are supported
    - Windows is experimental
- Python
- AWS CLI
- Pip
- Virtualenv (recommended)

Installations commands

```
$ sudo apt update
$ sudo apt install python-pip
$ sudo -H pip install awscli
$ aws configure
$ sudo –H pip install virtualenv
$ virtualenv pcluster
$ source pcluster/bin/activate
$ sudo pip install aws-parallelcluster
```

1) Let’s install!

ParallelCluster commands

pcluster [command]

- create Creates a new cluster
- update Updates a running cluster using the values in the config file or in a TEMPLATE_URL provided
- delete Deletes a cluster
- start Starts the compute fleet for a cluster that has been stopped
- stop Stops the compute fleet, leaving the master server running
- status Pulls the current status of the cluster
- list Displays a list of stacks associated with AWS ParallelCluster
- instances Displays a list of all instances in a cluster
- ssh Connects to the master instance using SSH
- createami (Linux/macOS) Creates a custom AMI to use with AWS ParallelCluster
- configure Start the AWS ParallelCluster configuration
- version Displays the version of AWS ParallelCluster

optional arguments:
-h, --help show this help message and exit

For command specific flags, please run: "pcluster [command] --help"

Configure

Collect information:

- ssh keys
- Region
- VPC
- Subnet
- Instance Type for the Master Node
- Instance Type for the Computing Nodes

```
$ pcluster configure
```

2) Configure

EC2 Instance Types

Instance Types	M5	M5d	R5	R5d	C4	C5	C5d	NEW C5n	Z1d	P3	NEW P3dn	G3	F1
Example use case	FEA Implicit				CFD, FEA Explicit				EDA, CFD	ML/AI CUDA		Remote Visualization	Genomics, Finance
Max CPU (GHz)	3.1		3.1		2.9	3.5			4.0	2.7	2.7	2.7	2.7
Max RAM (GB)	384		768		60	144		192	384	488	768	488	976
Max vCPUs	96		96		36	72			48	64	96	64	64
Max cores (*)	48		48		18	36			24	32	48	32	32
RAM/vCPUs	4		8		1.6	2		2.6	8	7,6	8	7,6	15.25
RAM/cores (*)	8		16		3.3	4		5.3	16	15,2	16	15,2	30.5
Max NVMe SSD (TB)	NA	1.8	NA	3.6	NA	NA	1.8	NA	1.8	NA	1.8	NA	3.7
Max Network Bandwidth (Gbps)	25	25	25	25	10	25		100	25	25	100	25	25
Network Adapter	ENA	ENA	ENA	ENA	ENA	ENA		NEW EFA	ENA	ENA	NEW EFA	ENA	ENA
Accelerated Computing	---									Up to 8 Nvidia Volta V100		Up to 4 Nvidia Tesla M60	Up to 8 Xilinx FPGA s

EFA

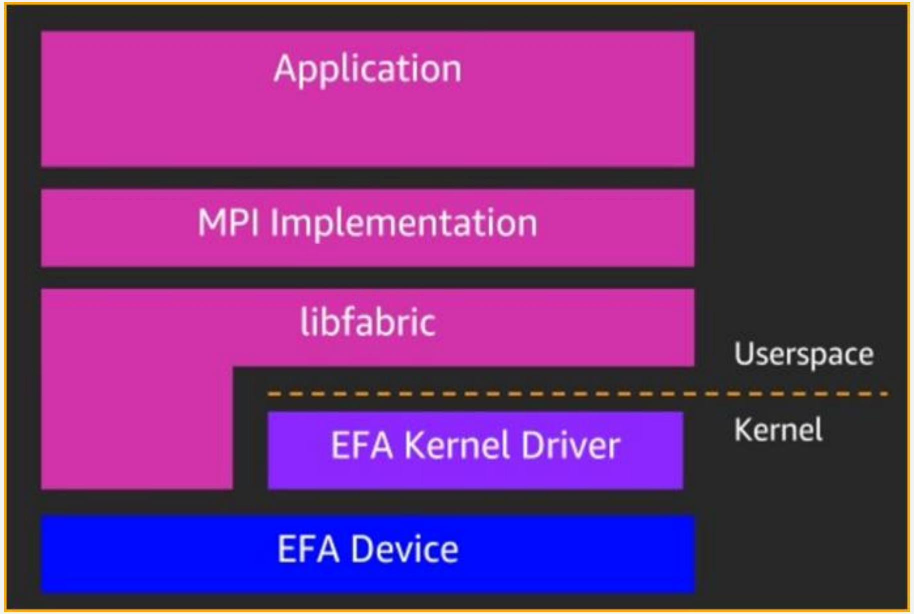
An Elastic Fabric Adapter is an AWS Elastic Network Adapter (ENA) with added capabilities.

An EFA can still handle IP traffic, but also supports an important access model commonly called **OS bypass**.

This model allows the application access the network interface without having to get the kernel involved

Doing so reduces overhead and allows the application to run more efficiently

EFA can provide **one-way MPI latency of 15.5** microseconds.



More info: https://aws.amazon.com/blogs/aws/now-available-elastic-fabric-adapter-efa-for-tightly-coupled-hpc-workloads/

Configure (2)

```
$ vi .parallelcluster/config
```

```
[cluster default]
key_name = fruffino-hpcdemo
vpc_settings = public
compute_instance_type = c5.18xlarge
master_instance_type = m5.xlarge
maintain_initial_size = false
initial_queue_size = 0
max_queue_size = 10
placement_group = DYNAMIC
placement = cluster
scaling_settings = custom
tags = {"name" : "HPCWebinar"}
base_os = centos7
```

```
[scaling custom]
scaledown_idletime = 1

[vpc public]
vpc_id = vpc-xxxxx
master_subnet_id = subnet-xxxx
```

Deploy

```
$ pcluster create c5
$ pcluster ssh c5 –i mykey.pem
```

AWS ParallelCluster mounts an ebs volume as an nfs filesystem as configured in the [ebs] section of the config. This defaults to /shared.

3) Finalize the configuration and deploy

Your first job

Create a file helloworld.sh

```
#!/bin/bash
#$ -cwd
#$ -j y
#$ -pe mpi 144
#$ -S /bin/bash
module load mpi/openmpi-x86_64
mpirun -np 144 hostname
```

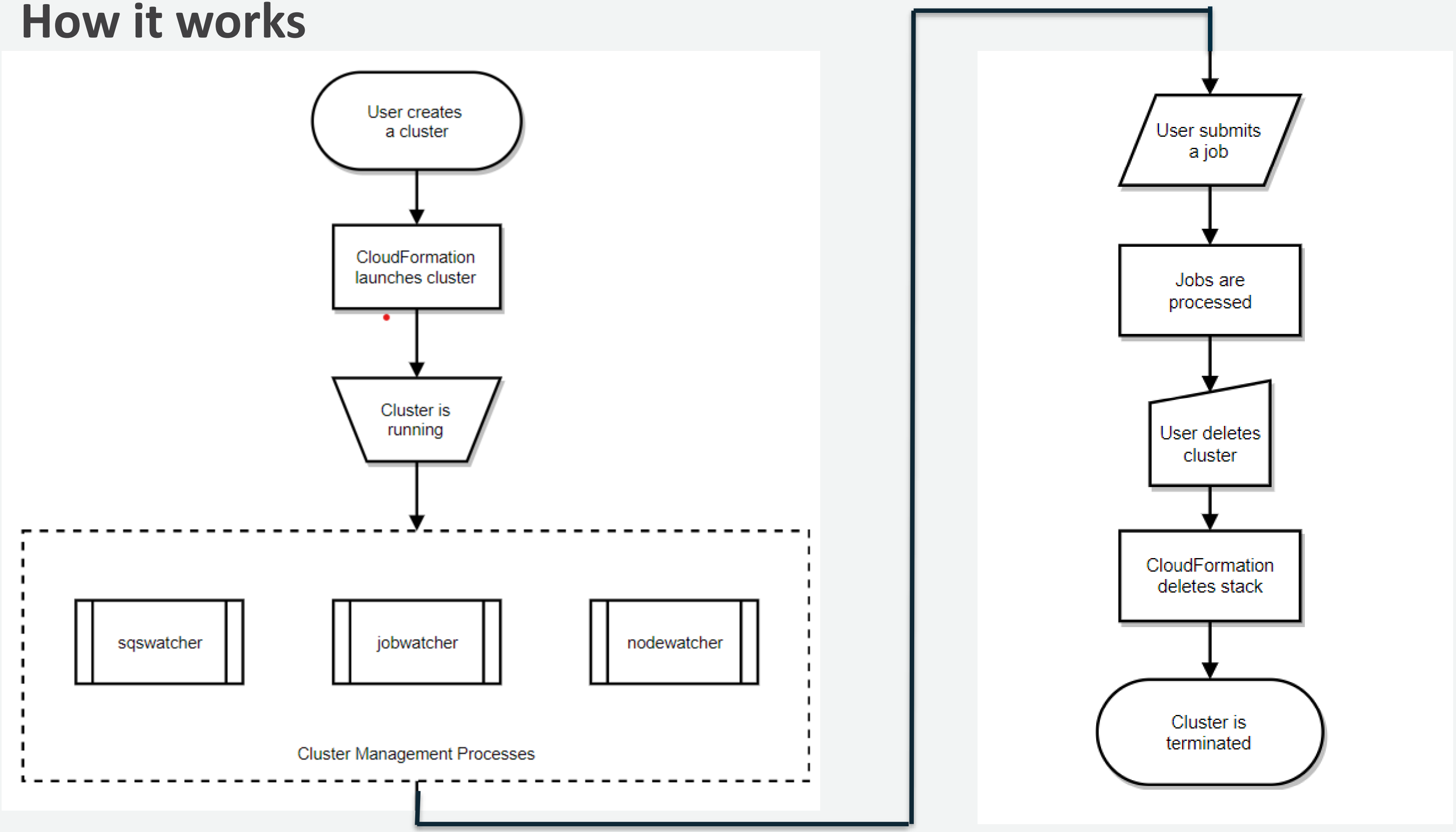
Submit the job

```
$ qsub ~/helloworld.sh
```

Read the results

```
$ cat helloworld.sh.o1
```

How it works


User creates a cluster
CloudFormation launches cluster
Cluster is running
sqswatcher
jobwatcher
nodewatcher
Cluster Management Processes
User submits a job
Jobs are processed
User deletes cluster
CloudFormation deletes stack
Cluster is terminated

How it works (2)

Jobwatcher

Once a cluster is running, a process owned by the root user will monitor the configured scheduler (SGE, Torque, Slurm, etc) and each minute, it’ll evaluate the queue in order to decide when to scale up

SQSwatcher

The sqswatcher process monitors for SQS messages emitted by Auto Scaling which notifies of state changes within the cluster. When an instance comes online, it will submit an “instance ready” message to SQS, which is picked up by sqs_watcher running on the master server. These messages are used to notify the queue manager when new instances come online or are terminated, so they can be added or removed from the queue accordingly

Nodewatcher

The nodewatcher process runs on each node in the compute fleet. After the user defined `scaledown_idletime` period, the instance is terminated.

4) Your first job

Run an AWS Batch job

```
[global]
sanity_check = true

[aws]
aws_region_name = us-east-1

[cluster awsbatch]
base_os = alinux
# Replace with the name of the key you
intend to use.
key_name = key-#######
vpc_settings = my-vpc
scheduler = awsbatch
compute_instance_type = optimal
min_vcpus = 2
desired_vcpus = 2
max_vcpus = 24
```

```
[vpc my-vpc]
# Replace with the id of the vpc
you intend to use.
vpc_id = vpc-#######
# Replace with id of the subnet for
the Master node.
master_subnet_id = subnet-#######
# Replace with id of the subnet for
the Compute nodes.
# A NAT Gateway is required for
MNP.
```

Run an AWS Batch job

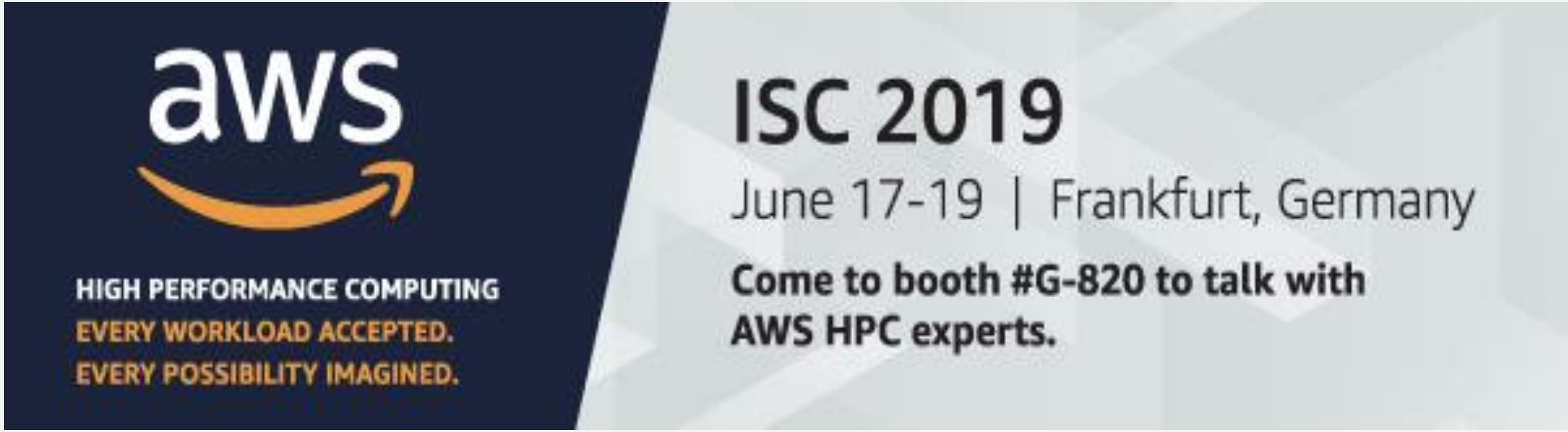
Advanced configuration

More parameters

```
#MORE OPTIONS
cluster_type = spot
spot_price = 1.00
pre_install =
http://hostname/path/to/disable
HT.sh
scheduler = sge
base_os = centos7
fsx_settings = fs
efs_settings = customfs
```

```
[fsx fs]
shared_dir = /fsx
storage_capacity = 3600
import_path = s3://bucket
imported_file_chunk_size = 1024
export_path = s3://bucket/folder
weekly_maintenance_start_time =
1:00:00
[efs customfs]
shared_dir = efs
encrypted = false
performance_mode = generalPurpose
|| maxIO
efs_fs_id = fs-12345
```

Thank you!



aws

HIGH PERFORMANCE COMPUTING

EVERY WORKLOAD ACCEPTED.

EVERY POSSIBILITY IMAGINED.

ISC 2019

June 17-19 | Frankfurt, Germany

Come to booth #G-820 to talk with AWS HPC experts.

Learn more

Home page:
https://aws.amazon.com/hpc/resources

Docs:

- Whitepaper: What a TCO Analysis Won't Tell You
- Reference Architecture : HPC Lens - Well Architected Framework

Webinar:

- High Performance Computing on AWS - AWS Online Tech Talks

Blog e Web Pages:

- AWS ParallelCluster
- AWS Batch
- EFA
- FSx for Lustre

Q&A

Francesco Ruffino

Sr. Specialized HPC Solution Architect

fruffino@amazon.com