# Research Computing on AWS Made Easy

July 12th 2021, CyberColombia - Summer School 2021

Nina Vogl, AWS PS HPC Specialist SA

# Agenda

- A few “Cloud Principals”
- How this DOES benefit HPC and Research Computing?
- We  CAN EASILY arrange our “Bag of Lego Blocks”!
- Let me show you!!
- Do it yourself: Read the Spanish Blog!!!

# A few “Cloud Principals”

- Infrastructure as Code
- Ephemeral Clusters
- Re-usable Setups
- Only pay for what you use, when you use it
- VAST Capacity of Resources
- Large Variety of Architectures

# How this benefits HPC and Research Computing

First, lets state some goals:

- Have your very own Supercomputer at your disposal when you need it!
- Only pay when your cluster is actually running.
- Have your HPC Applications ready installed and data easily accessible

AWS Lambda

Amazon Linux

template

Auto Scaling

Python

# Why use AWS for HPC?

instance with Amazon CloudWatch

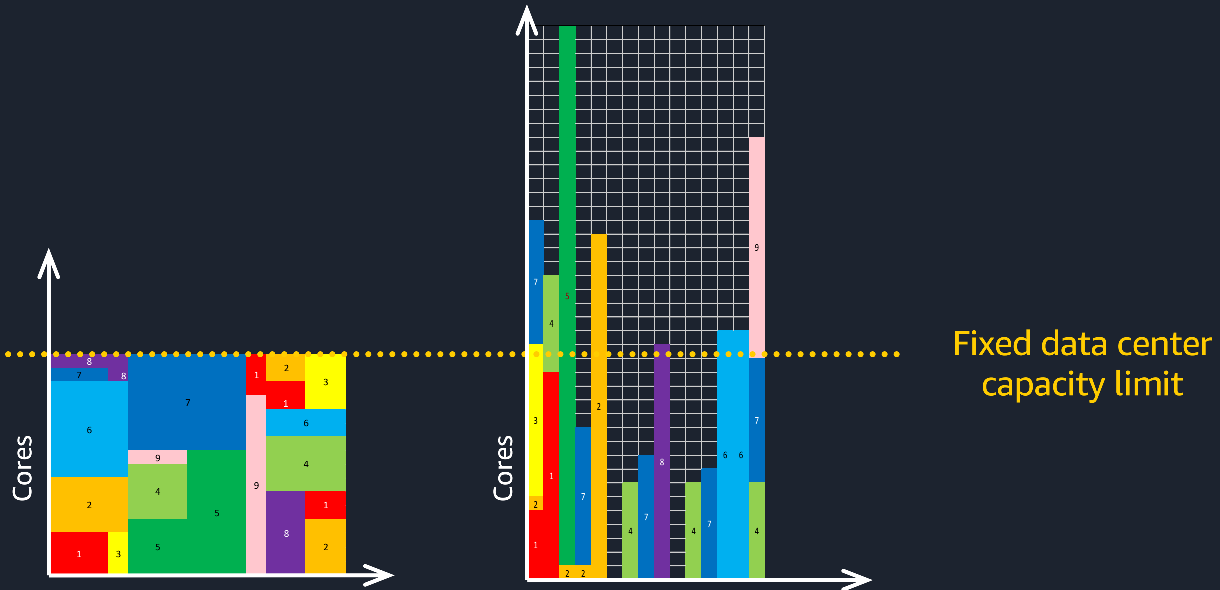
- Amazing Amount of Compute Capacity – Economy of Scale
- A virtually unlimited number of architecture options
  - Instance Types, OS, Traditional Cluster, Auto Scaling Clusters, Serverless, GPUs
- Extensive deployment options – “Infrastructure as Code”
  - Console, Configuration Control, Automated, SDK, Bash/CLI, AWS CloudFormation
- Lots of useful services
  - Amazon DynamoDB, Amazon CloudWatch, Amazon Glacier, and much more!

Ruby

Amazon DynamoDB

# No Queue Wait Times
# Always Get Access to the Resources You Need




Finite capacity
Queue Wait times
Limited Selection of resources

Elasticity: Massive capacity when needed
Flexibility: Time to results speed up
Agility: Resource availability when experimentation is needed

# Cloud Components that enable HPC on AWS

AWS ParallelCluster

Amazon EC2

Elastic Fabric Adapter (EFA)

Amazon FSx for Lustre

Amazon Storage

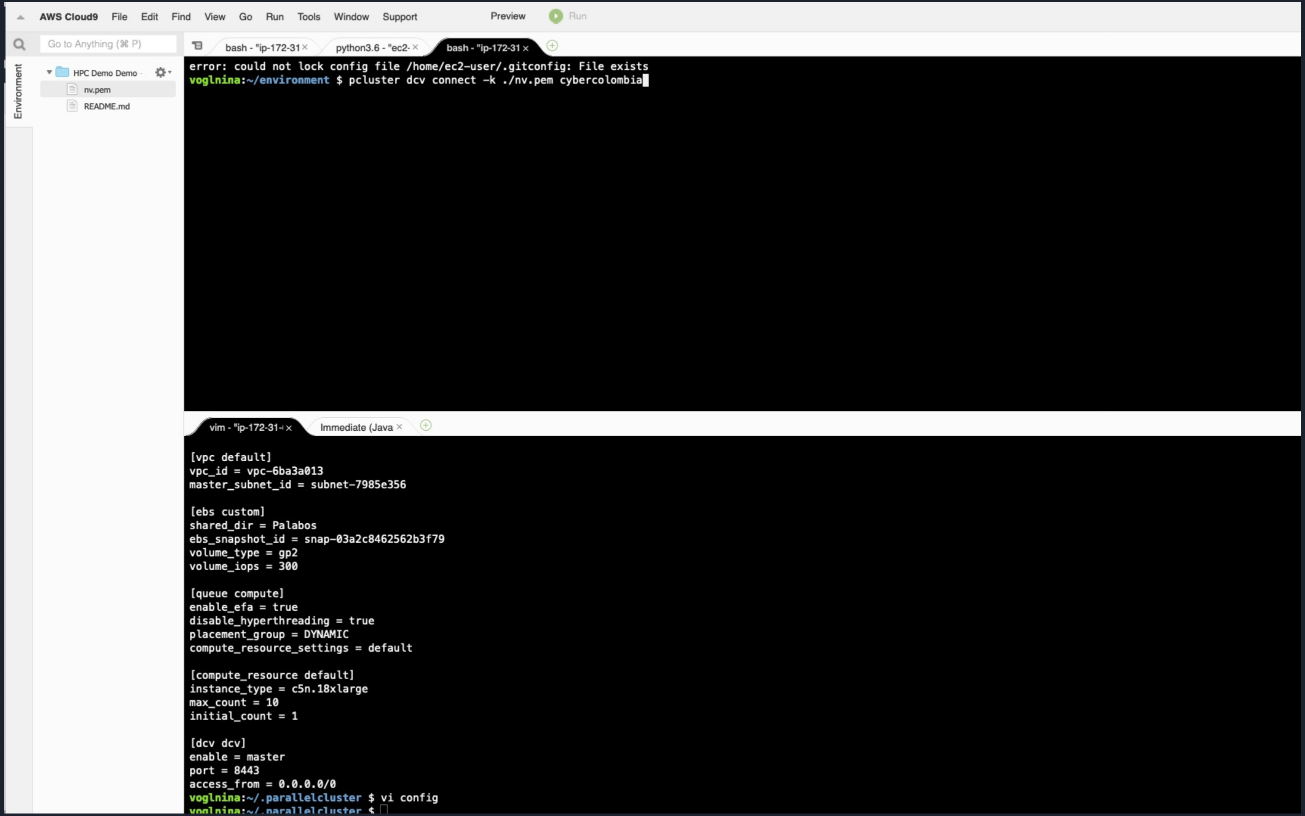
# We'll arrange our "Bag of Lego Blocks" Easily
# Tools That Help Us Do So

- AWS Cloud9
- AWS Parallel Cluster
- The Configuration File
- EBS Snapshot ID
- Nice DCV to Visualize the Results

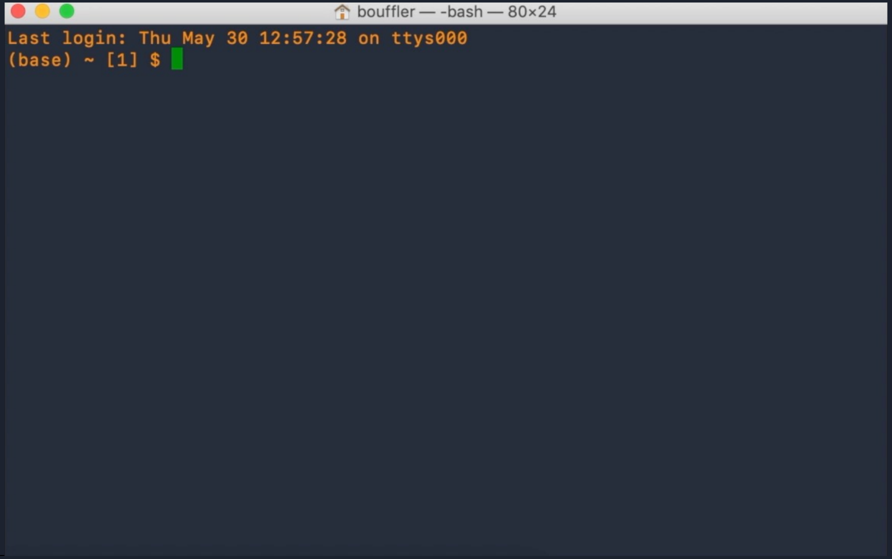
# AWS Cloud9 – A Linux Terminal in Your Cloud

# A Terminal in YOUR Cloud



# AWS Parallel Cluster Installation

```
$ pip3 install aws-parallelcluster --upgrade --user
```

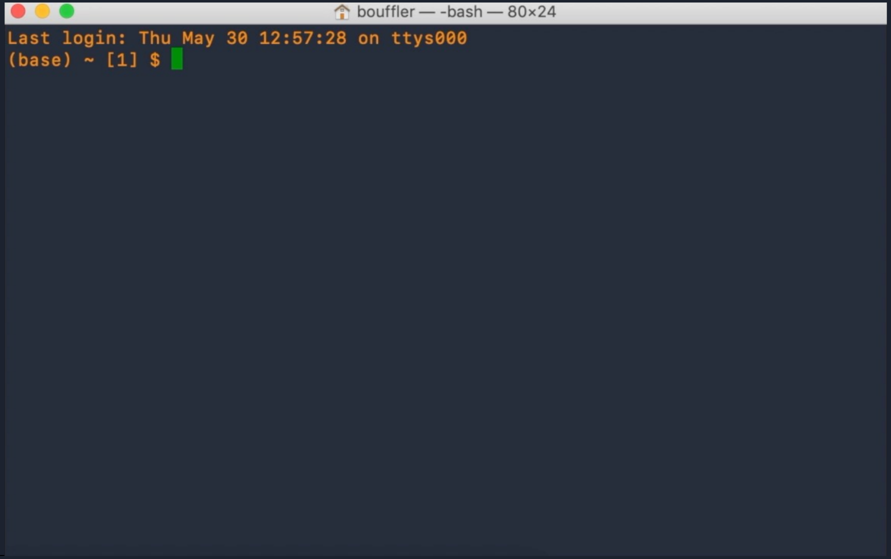


# A Human Readable Configuration File Helps Design your Cluster

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

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

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

[cluster default]
key_name = KeyPairHPCluster
base_os = alinux2
scheduler = slurm
s3_read_resource = arn:aws:s3:::myhpcblogs3bucket/*
post_install = s3://myhpcblogs3bucket/postInstall.sh
post_install_args = 'R curl wget'
master_instance_type = g3.4xlarge
vpc_settings = default
queue_settings = compute
ebs_settings = custom
dcv_settings = dcv

[vpc default]
vpc_id = vpc-xxx
master_subnet_id = subnet-xxx

[ebs custom]
shared_dir = MyApps
ebs_snapshot_id = snap-xxx
volume_type = gp2

[queue compute]
enable_efa = true
disable_hyperthreading = true
placement_group = DYNAMIC
compute_resource_settings = default

[compute_resource default]
instance_type = c5n.18xlarge
max_count = 10
initial_count = 1
min_count = 0

[dcv dcv]
enable = master
port = 8443
access_from = 0.0.0.0/0
```

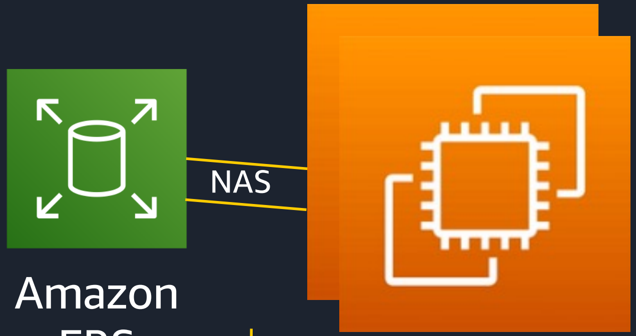
# AWS Parallel Cluster Installation

```
$ pcluster configure
$ pcluster create myhpcluster
$ pcluster ssh myhpcluster -i ~/KeyPairHPCluster.pem
```



# “Pre-Install” Your Application by using an EBS Snapshot
# Do this ONCE

/dev/sdb1
/dev/nvme1n1




/MyApps/Palabos
/MyApps/ParaView
/MyApps/MyScripts

Amazon
EBS

Use the AWS Console to
Create a Snapshot
Get Snapshot ID




aws

# A Human Readable Configuration File Helps Design your Cluster

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

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

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

[cluster default]
key_name = KeyPairHPCluster
base_os = alinux2
scheduler = slurm
s3_read_resource = arn:aws:s3:::myhpcblogs3bucket/*
post_install = s3://myhpcblogs3bucket/postInstall.sh
post_install_args = 'R curl wget'
master_instance_type = g3.4xlarge
vpc_settings = default
queue_settings = compute
ebs_settings = custom
dcv_settings = dcv

[vpc default]
vpc_id = vpc-xxx
master_subnet_id = subnet-xxx

[ebs custom]
shared_dir = MyApps
ebs_snapshot_id = snap-xxx
volume_type = gp2

[queue compute]
enable_efa = true
disable_hyperthreading = true
placement_group = DYNAMIC
compute_resource_settings = default

[compute_resource default]
instance_type = c5n.18xlarge
max_count = 10
initial_count = 1
min_count = 0

[dcv dcv]
enable = master
port = 8443
access_from = 0.0.0.0/0
```

# "Back in the HPC World"
# Use A Scheduler of your choice to Submit Your HPC Job

```
$ pcluster ssh myhpcluster -i ~/KeyPairHPCluster.pem
```

```
#!/bin/bash
#SBATCH --job-name=PALABOS
#SBATCH --output=PALABOS_%j.out
#SBATCH --nodes=10 # Number of nodes
#SBATCH --ntasks=360 # Number of MPI ranks
#SBATCH --ntasks-per-node=36 # Number of MPI ranks per node
#SBATCH --cpus-per-task=1 # Number of OpenMP threads for each MPI process/rank
date;hostname;pwd
#export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
cd /Palabos/palabos-v2.2.1/examples/showCases/boussinesqThermal3d
mpirun -np 360 ./rayleighBenard3D 1000
date
```

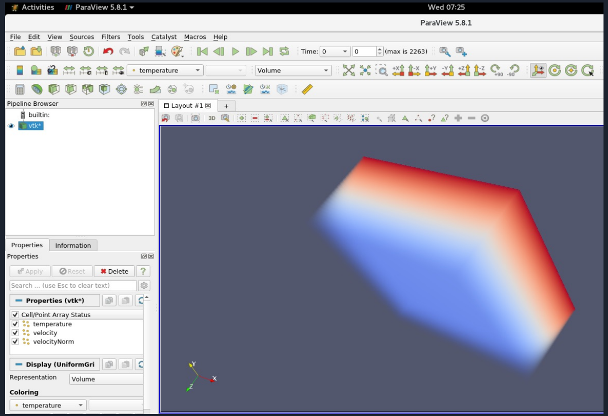
```
$ sbatch submit.sh
```

# Visualize Your Results Directly on the Cluster

- NICE DCV
- In Your “Terminal in the Cloud”, The Cloud9 Environment, Type:

```
$ pcluster dcv connect -k ~/environment/KeyPairHPCluster.pem myhpcluster
```



# Let Me Show You

- Cloud9 – A Terminal/Shell "in your cloud"
- AWS Parallel Cluster Install
- AWS Parallel Cluster Configuration
- Starting a new cluster
- Leverage a Snapshot with Palabos
- Using DCV on the master node for visualization of the results
- Demonstrate Scaling the Compute Nodes With "mpi_hello_world"
    - Default ScaleDownIdle Time is 10 Minutes

# DIY - Do It Yourself

- PreRequisits
  - AWS Account
  - Potentially Request Limit Increase
  - Limited Linux CLI Knowledge
  - AWS Folks You Can Reach out to
- "Costing"
  - Need More Time for Details AS there are LOTS of Choices and Things to Consider
  - Remember: You only pay for what you use
  - Sample Prices JUST to Give an Idea – See Next Slide
- Documentation To Help You:
  - AWS Parallel Cluster Documentation https://docs.aws.amazon.com/parallelcluster/index.html
  - Spanish Blog https://aws.amazon.com/es/blogs/aws-spanish/como-poner-una-supercomputadora-en-manos-de-todo-cientifico/

# Just To Give An Idea – Sample Prices To Run Palabos BoussinesqThermal3d

**Viewing 4 of 352 available instances**

Search: C5n

| Instance name | On-Demand hourly rate | vCPU | Memory | Storage | Network performance |
|---|---|---|---|---|---|
| c5n.18xlarge | $3.888 | 72 | 192 GiB | EBS Only | 100 Gigabit |

**Viewing 25 of 352 available instances**

Search: g3

| Instance name | On-Demand hourly rate | vCPU | Memory | Storage | Network performance |
|---|---|---|---|---|---|
| g3.4xlarge | $1.14 | 16 | 122 GiB | EBS Only | Up to 10 Gigabit |

**Viewing 352 of 352 available instances**

Search: t3

| Instance name | On-Demand hourly rate | vCPU | Memory | Storage | Network performance |
|---|---|---|---|---|---|
| t3.nano | $0.0052 | 2 | 0.5 GiB | EBS Only | Up to 5 Gigabit |
| t3.micro | $0.0104 | 2 | 1 GiB | EBS Only | Up to 5 Gigabit |

This Example is Using "OnDemand" Pricing
There ARE Other Choices
There is "Free Tier"

- One Head Node + 10 Compute Nodes (360 Cores) for Approximately 1.5 Hrs:
  - \$58.32 + \$1.14 = \$59.46 OR \$58.32 + \$0.0104 = \$58.3304
  - Plus Storage: EBS: ~ 8-10 Cents per GB Per MONTH: ~\$30 OR \$1 Per Day
  - Plus "Incidentals":
    - EBS Snapshot: \$0.05 Per GB Per Month: \$26.50 Per Months OR \$0.88 Per Day




aws

# How this benefits HPC and Research Computing

First, lets state some goals:

- Have your very own Supercomputer at your disposal when you need it!
- Only pay when your cluster is actually running.
- Have your HPC Applications ready installed and data easily accessible

# Thank you!

voglnina@amazon.com

 | 