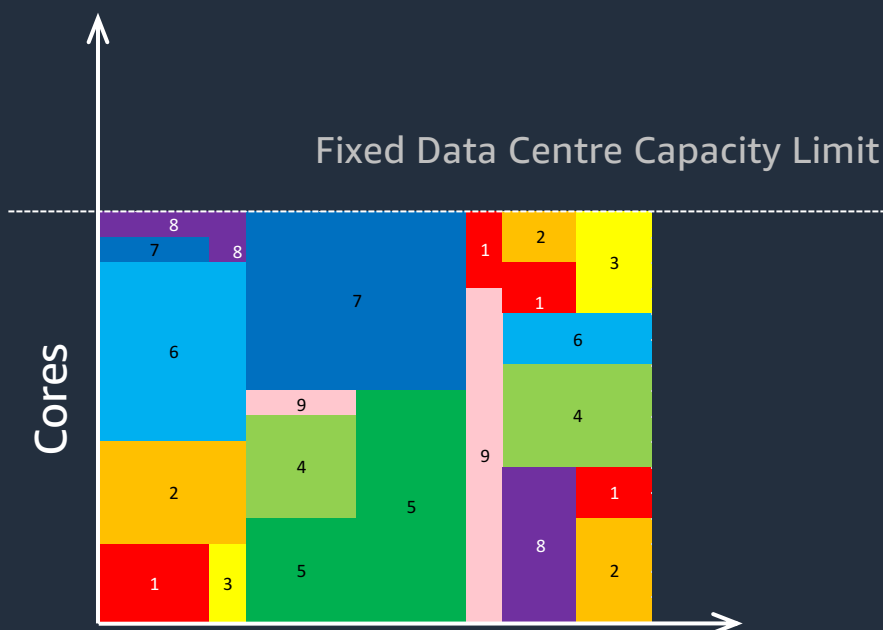




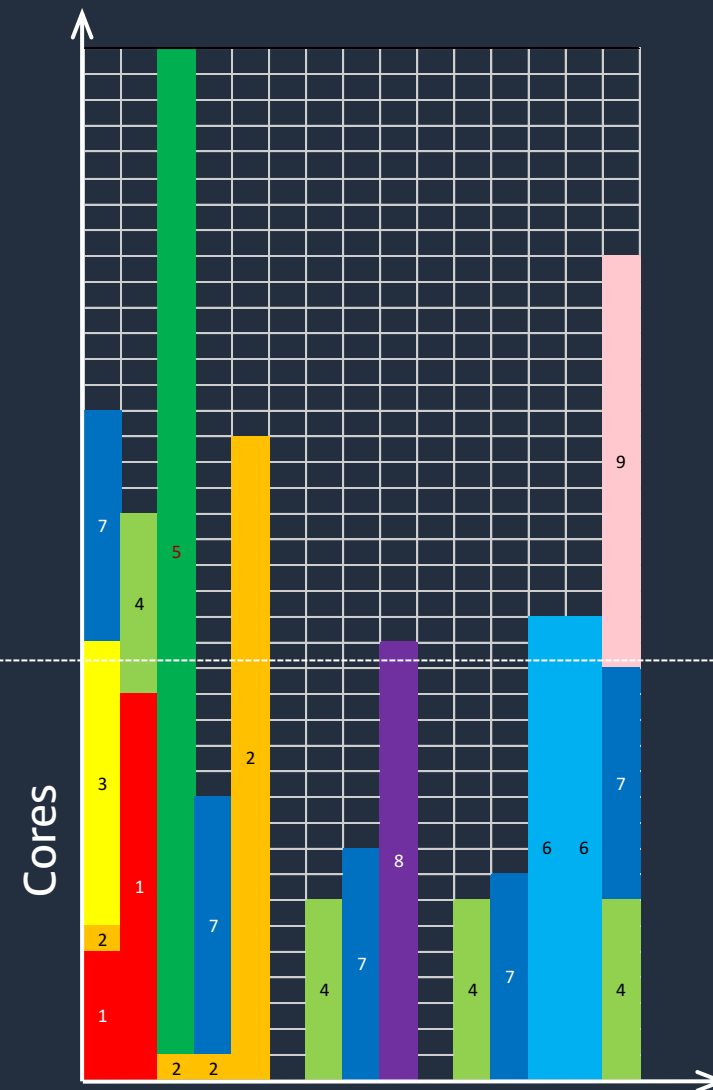
Standing up an HPC cluster on AWS using AWS ParallelCluster

Angel Pizarro, HPC @ AWS
MUG '21

The metric for success should be time-to-results



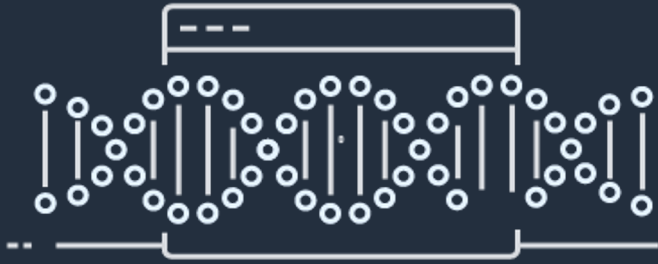
Finite capacity, usually with long queues to wait in.



Massive capacity when needed to speed up time to results, and agile environment when additional hardware and software experimentation is needed.

"For every \$1 spent on HPC, businesses see \$463 in incremental revenues and \$44 in incremental profit."

HPC workloads across industries



Life Sciences



Financial Services



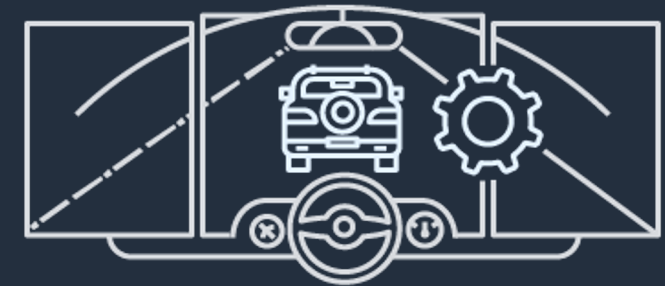
Oil & Gas



Design & Engineering



Climate & Geosciences



Autonomous Vehicles

HPC workloads with different compute and throughput characteristics



Tightly-coupled workloads



Loosely-coupled workloads



Accelerated computing



Visualization



AI/ML



High volume data analytics

What do HPC customers need

High Performance

Reliability

Low Latency/Overhead

Consistency & Jitter free network

Fairness

Scalable Reliable Datagram (SRD)

A reliable high-performance lower-latency network transport

Guaranteed delivery

- Does not consume any resources on EC2 instances

Network aware multipath routing

- Optimally utilizes all network paths (ECMP), no hot-spots

Orders of magnitude lower tail latency & jitter

- Fast recovery from network events

No ordering guarantees

- No head-of-line blocking

<https://ieeexplore.ieee.org/document/9167399>

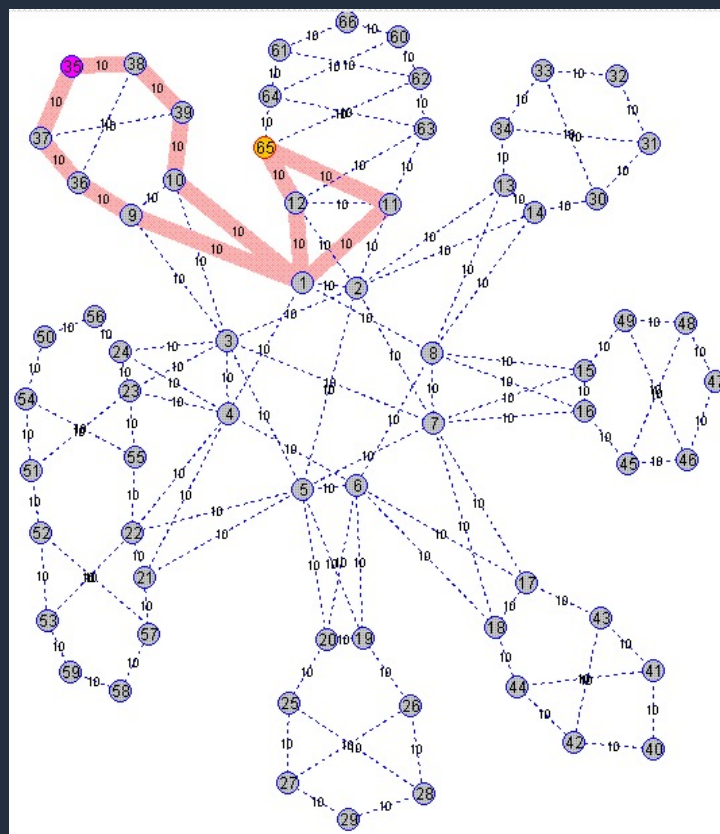


Elastic Fabric Adapter – Networks built to scale



Up to 400 Gbps
networking
bandwidth

- ✓ OS bypass
- ✓ GPUdirect and RDMA
- ✓ Libfabric core supports wide array of MPIs and NCCL

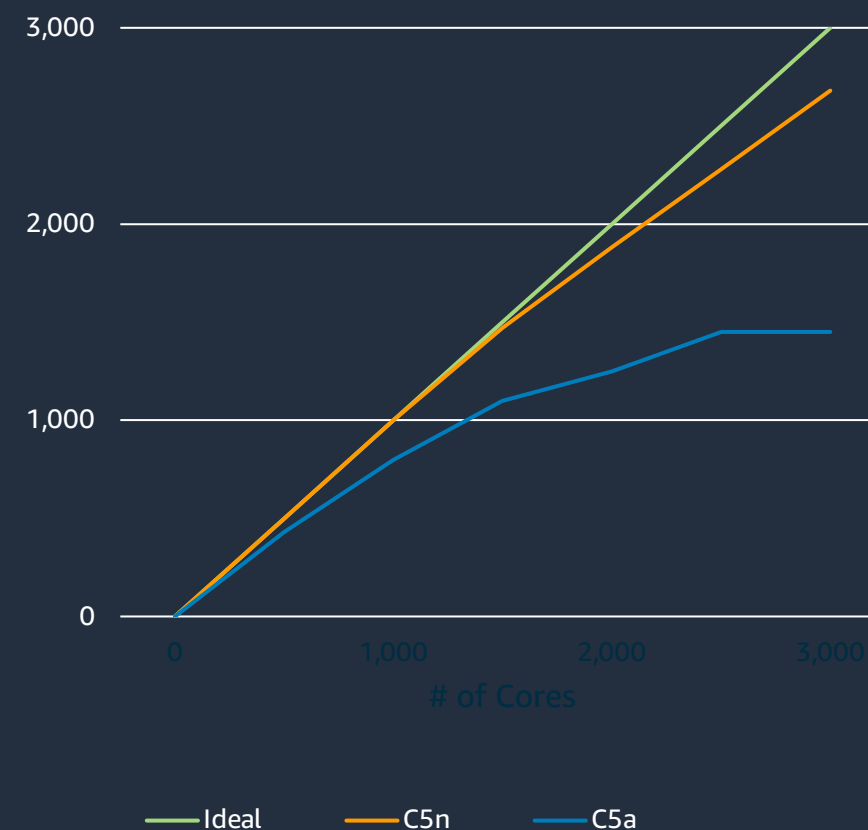


OCMP-enabled packet spraying and
cloud-scale congestion control

ANSYS Fluent

External flow over F1 race car (140M cell mesh)

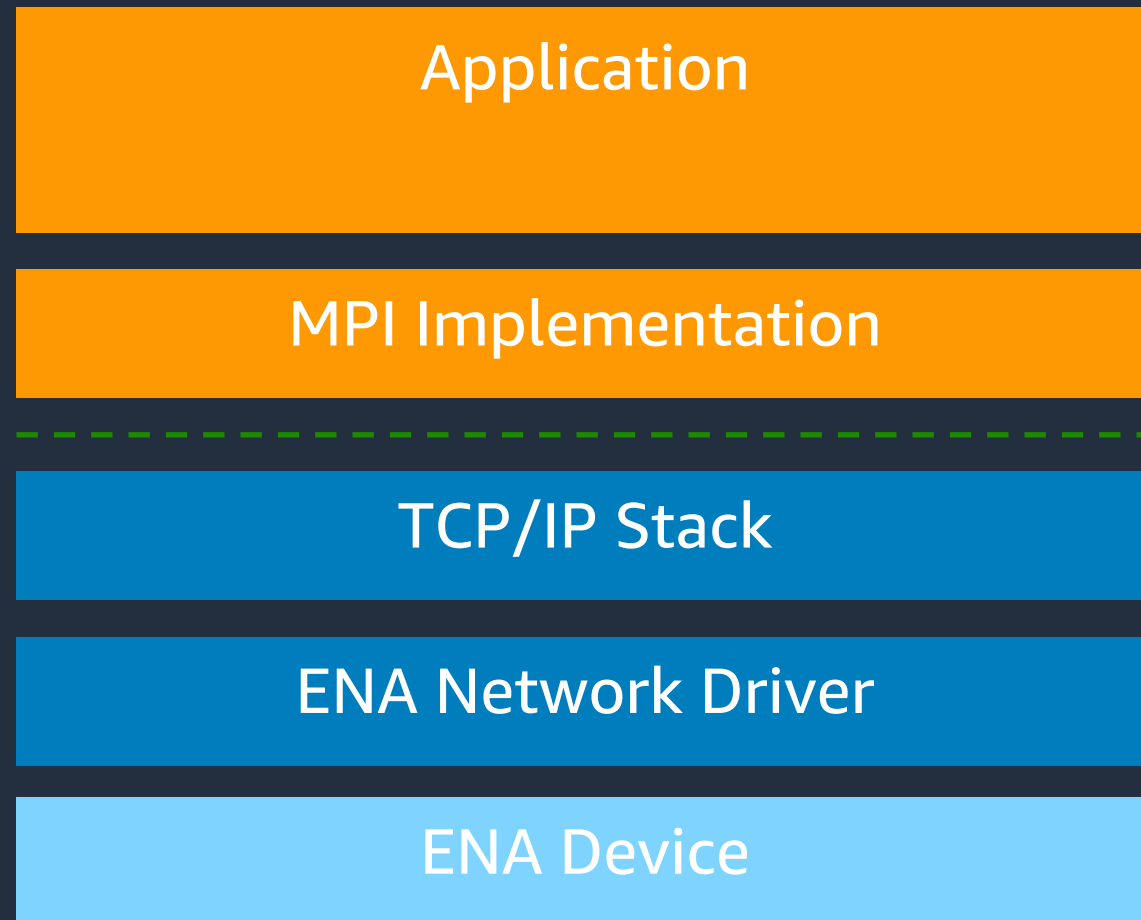
Scaling



At ~3,000 cores (~83 nodes), C5n+EFA shows ~89% scaling efficiency vs ~48% using C5 w/o EFA

HPC software stack on Amazon EC2

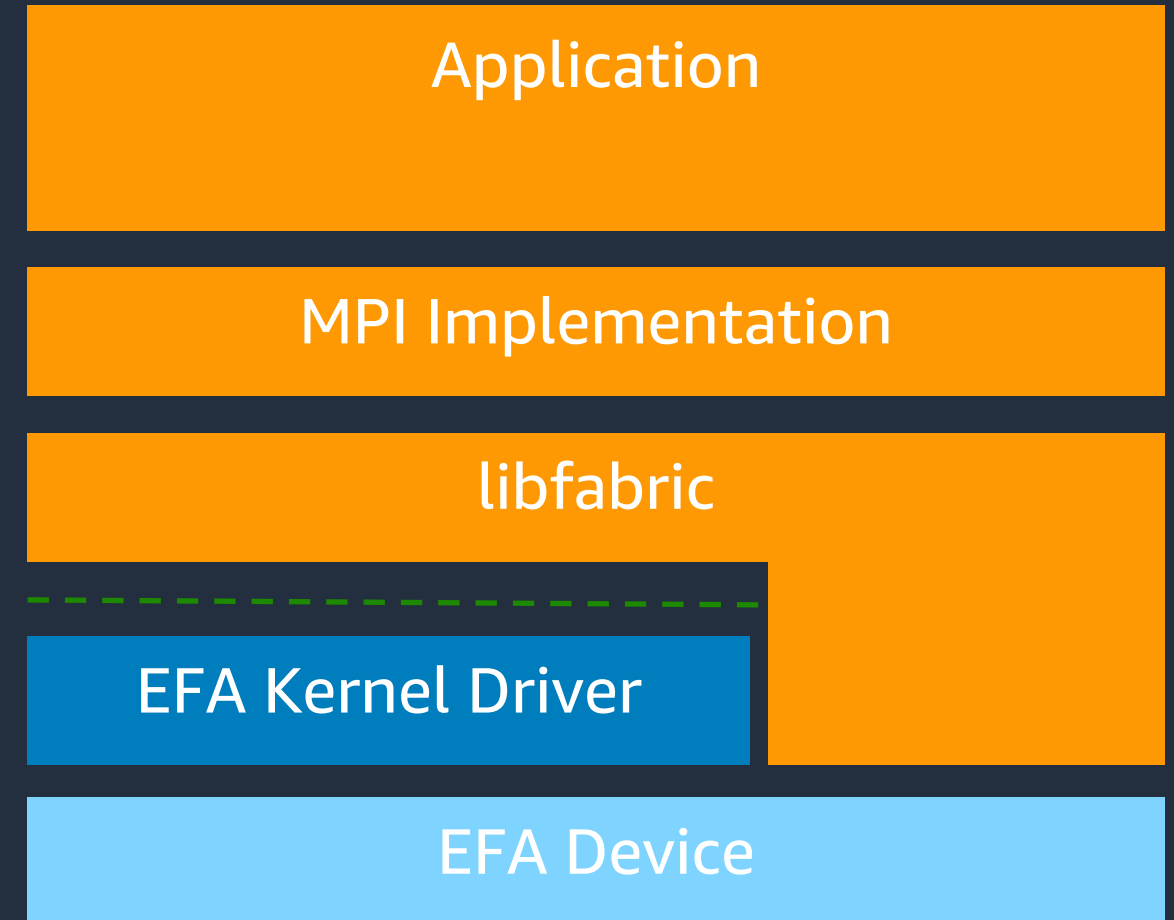
Without EFA



Userspace

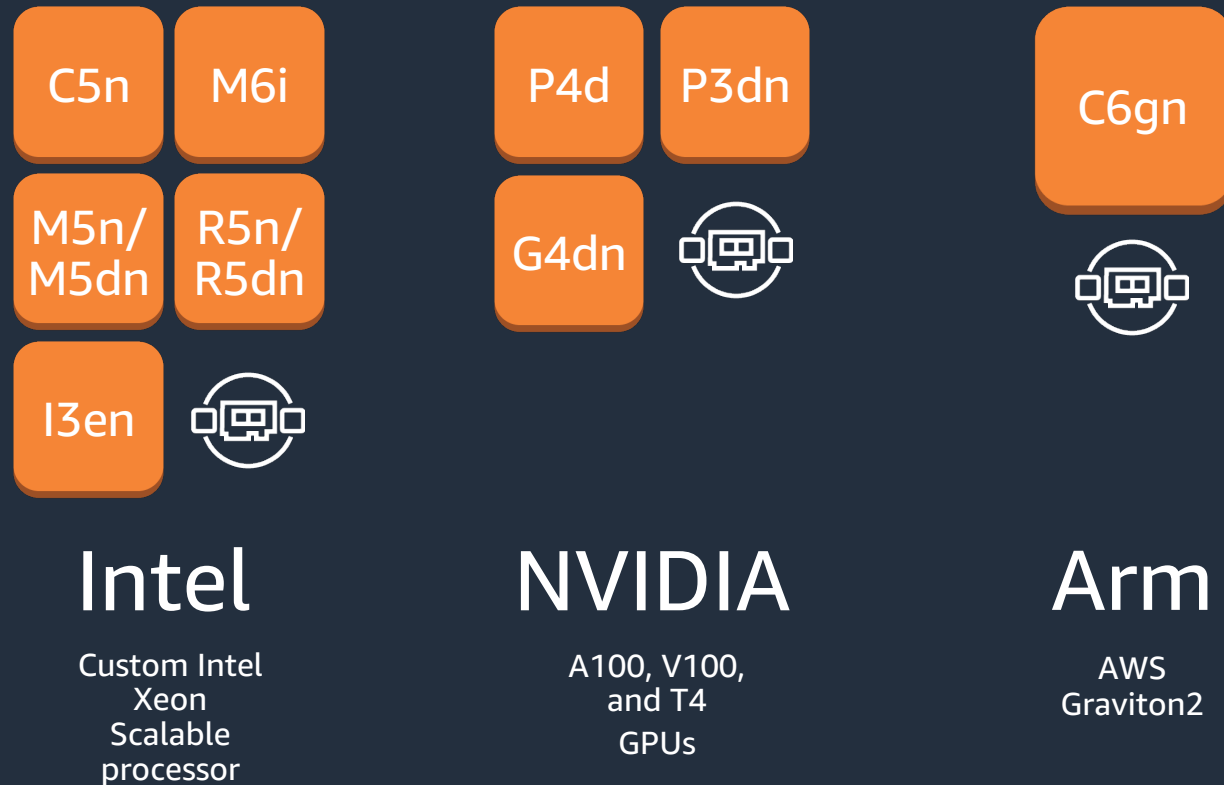
Kernel

With EFA



Elastic Fabric Adapter (EFA)

Scale **tightly coupled** HPC applications on AWS



EFA

AWS HPC/ML Network Interface

- Instance flexibility
- Infrastructure elasticity
- High data throughput
- Low-latency message passing
- Faster application time-to-completion

High Performance Computing (HPC) on AWS

On AWS, secure and well-optimized HPC clusters can be automatically created, operated, and torn down in just minutes



Machine learning and analytics

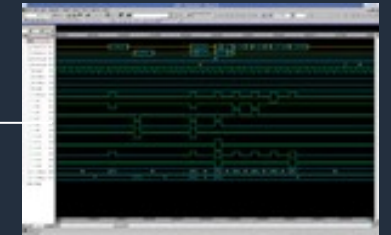
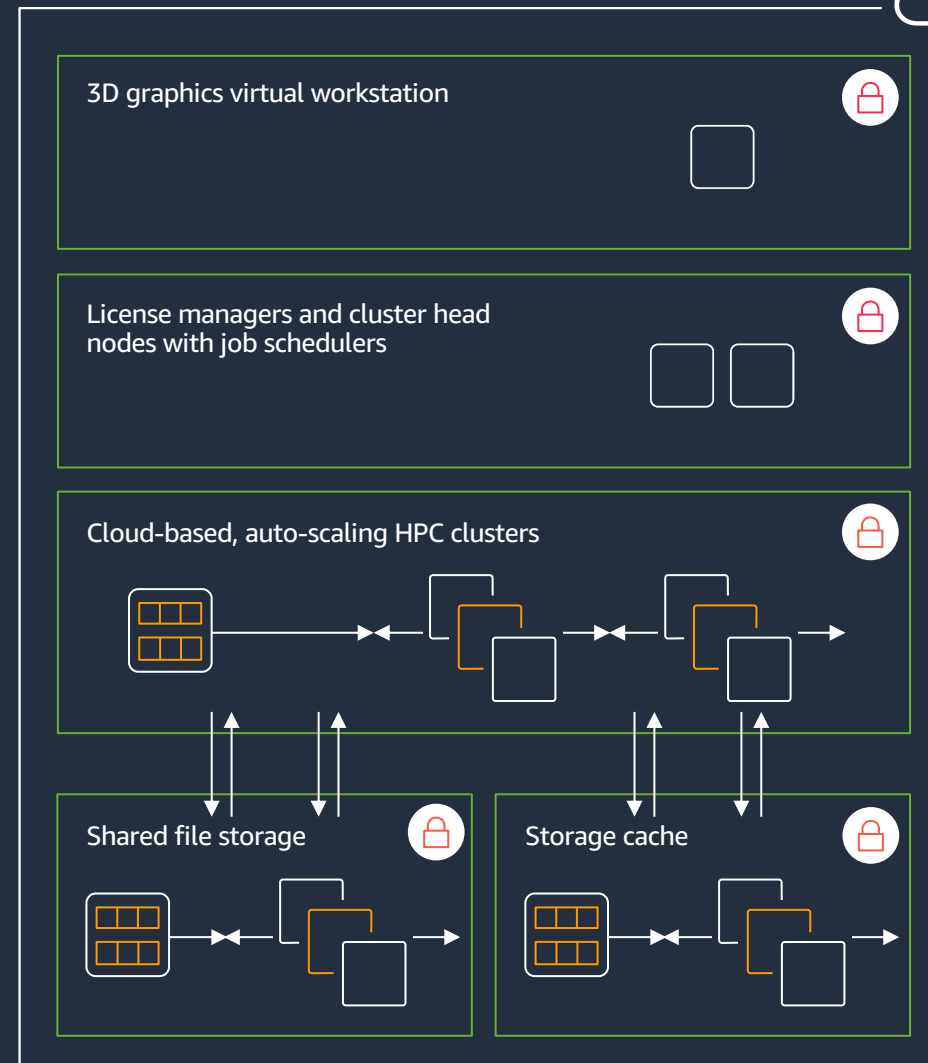


Amazon S3 and Amazon Glacier



Third-party IP providers and collaborators

Virtual Private Cloud on AWS



Thin or zero client—no local data

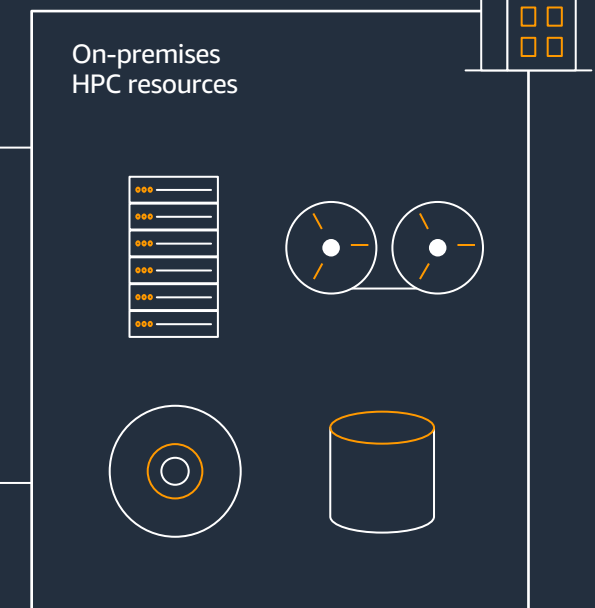
Corporate datacenter

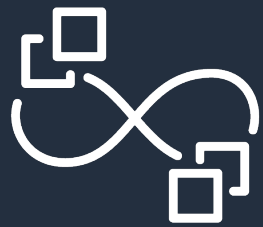


AWS Snowball



AWS Direct Connect





AWS ParallelCluster

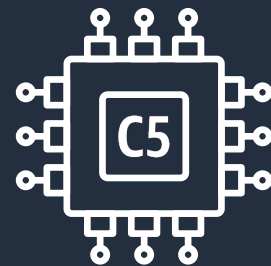


One-stop shop to
set up your HPC
cluster

Integrated with AWS services you need



Amazon FSx
for Lustre



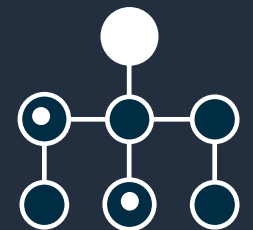
Amazon EC2
instances



EFA



NICE DCV

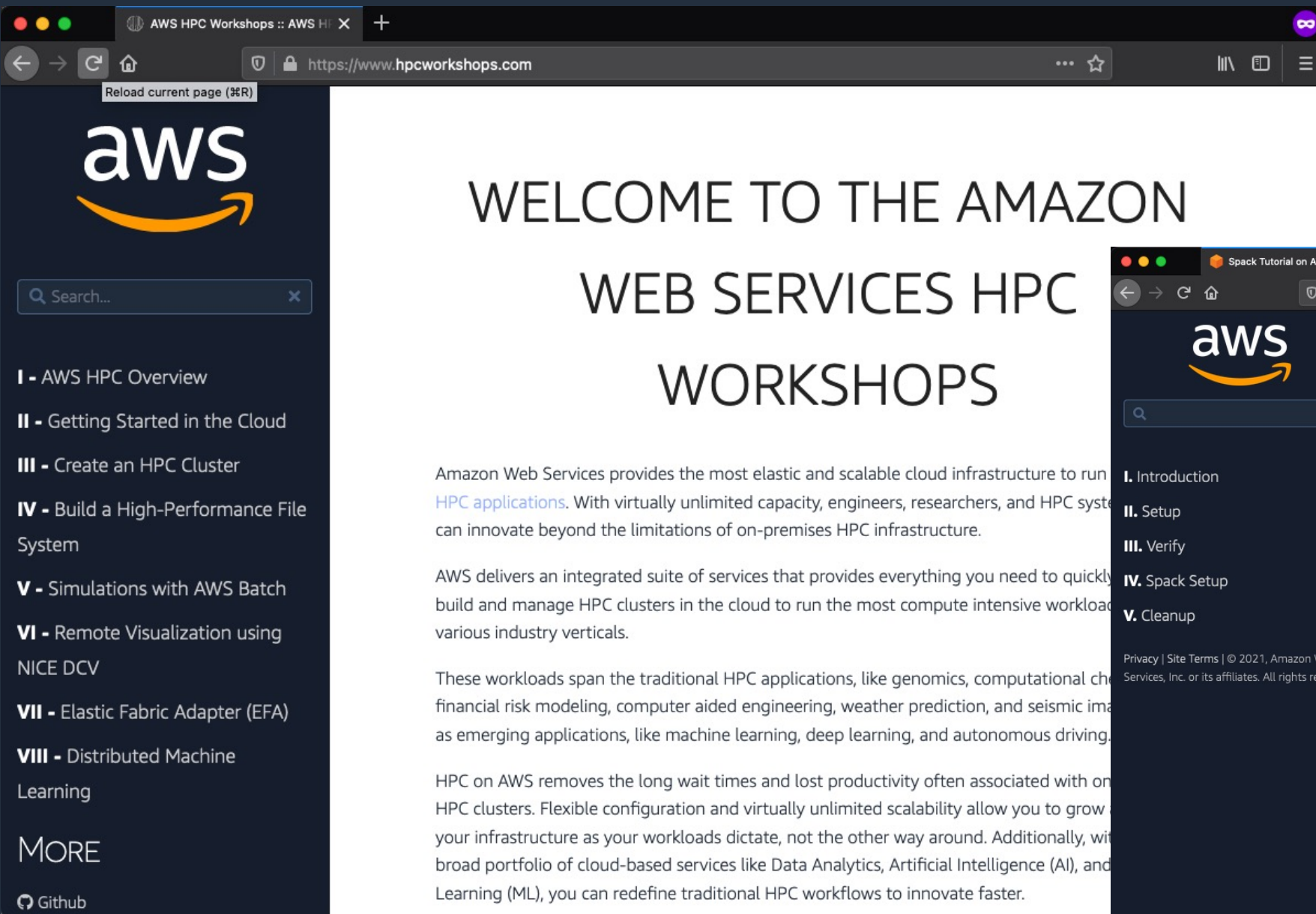


AWS Batch

Running EFA using AWS ParallelCluster

Subtitle

Getting started using Spack with AWS ParallelCluster



The screenshot shows the AWS HPC Workshops website. The header features the AWS logo and the text "WELCOME TO THE AMAZON WEB SERVICES HPC WORKSHOPS". Below the header, there is a navigation menu on the left with links to various workshops, including "AWS HPC Overview", "Getting Started in the Cloud", "Create an HPC Cluster", "Build a High-Performance File System", "Simulations with AWS Batch", "Remote Visualization using NICE DCV", "Elastic Fabric Adapter (EFA)", and "Distributed Machine Learning". The main content area contains text about AWS HPC services and a list of workshops.

aws

WELCOME TO THE AMAZON
WEB SERVICES HPC
WORKSHOPS

Amazon Web Services provides the most elastic and scalable cloud infrastructure to run [HPC applications](#). With virtually unlimited capacity, engineers, researchers, and HPC systems can innovate beyond the limitations of on-premises HPC infrastructure.

AWS delivers an integrated suite of services that provides everything you need to quickly build and manage HPC clusters in the cloud to run the most compute intensive workloads across various industry verticals.

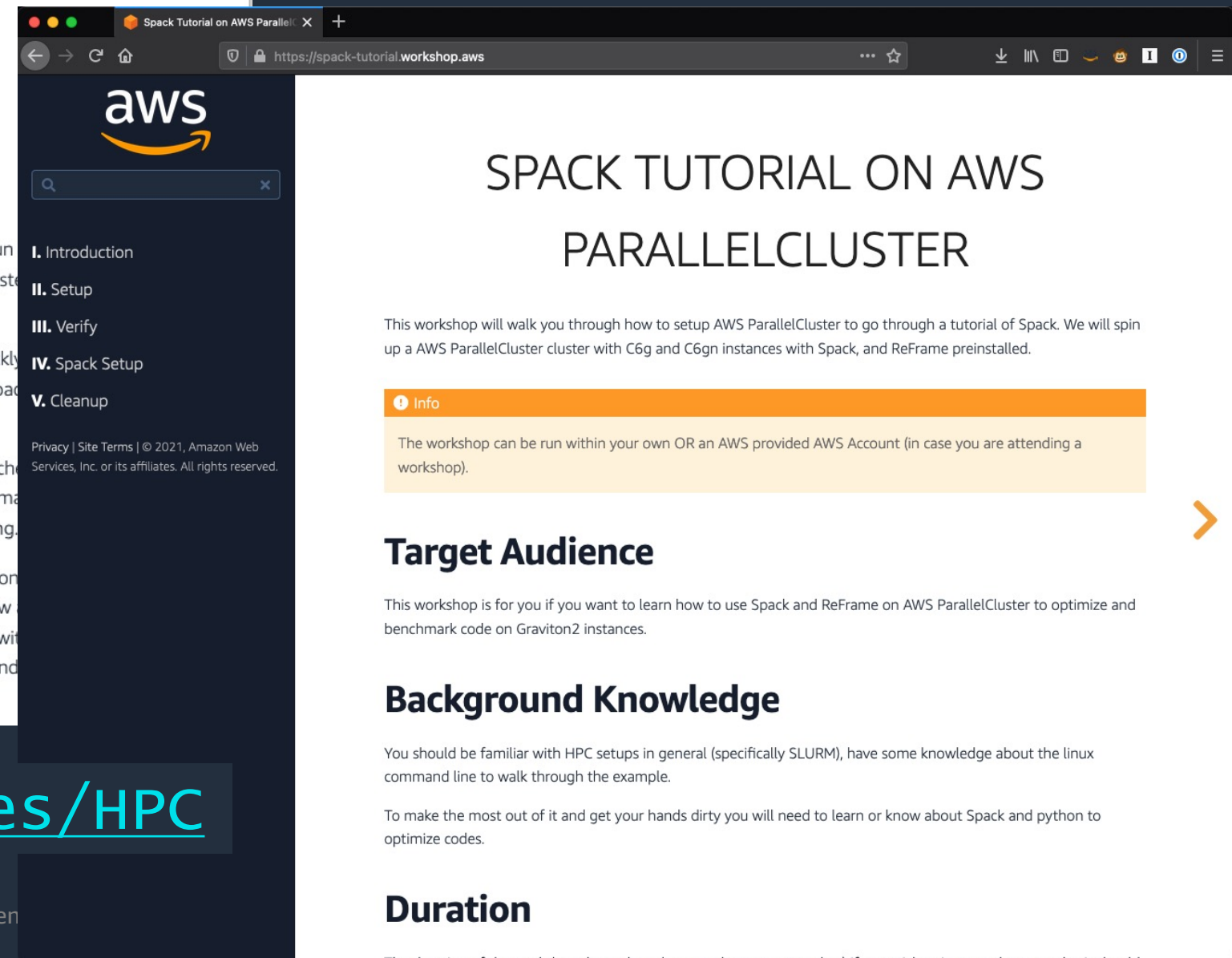
These workloads span the traditional HPC applications, like genomics, computational chemistry, financial risk modeling, computer aided engineering, weather prediction, and seismic imaging, as emerging applications, like machine learning, deep learning, and autonomous driving.

HPC on AWS removes the long wait times and lost productivity often associated with on-premises HPC clusters. Flexible configuration and virtually unlimited scalability allow you to grow your infrastructure as your workloads dictate, not the other way around. Additionally, with our broad portfolio of cloud-based services like Data Analytics, Artificial Intelligence (AI), and Machine Learning (ML), you can redefine traditional HPC workflows to innovate faster.

I - AWS HPC Overview
II - Getting Started in the Cloud
III - Create an HPC Cluster
IV - Build a High-Performance File System
V - Simulations with AWS Batch
VI - Remote Visualization using NICE DCV
VII - Elastic Fabric Adapter (EFA)
VIII - Distributed Machine Learning

MORE

Github



The screenshot shows the Spack Tutorial on AWS ParallelCluster website. The header features the AWS logo and the text "SPACK TUTORIAL ON AWS PARALLELCLUSTER". Below the header, there is a navigation menu on the left with links to various sections, including "Introduction", "Setup", "Verify", "Spack Setup", and "Cleanup". The main content area contains text about the workshop and a list of sections.

aws

SPACK TUTORIAL ON AWS
PARALLELCLUSTER

This workshop will walk you through how to setup AWS ParallelCluster to go through a tutorial of Spack. We will spin up a AWS ParallelCluster cluster with C6g and C6gn instances with Spack, and ReFrame preinstalled.

Info

The workshop can be run within your own OR an AWS provided AWS Account (in case you are attending a workshop).

Target Audience

This workshop is for you if you want to learn how to use Spack and ReFrame on AWS ParallelCluster to optimize and benchmark code on Graviton2 instances.

Background Knowledge

You should be familiar with HPC setups in general (specifically SLURM), have some knowledge about the linux command line to walk through the example.

To make the most out of it and get your hands dirty you will need to learn or know about Spack and python to optimize codes.

Duration

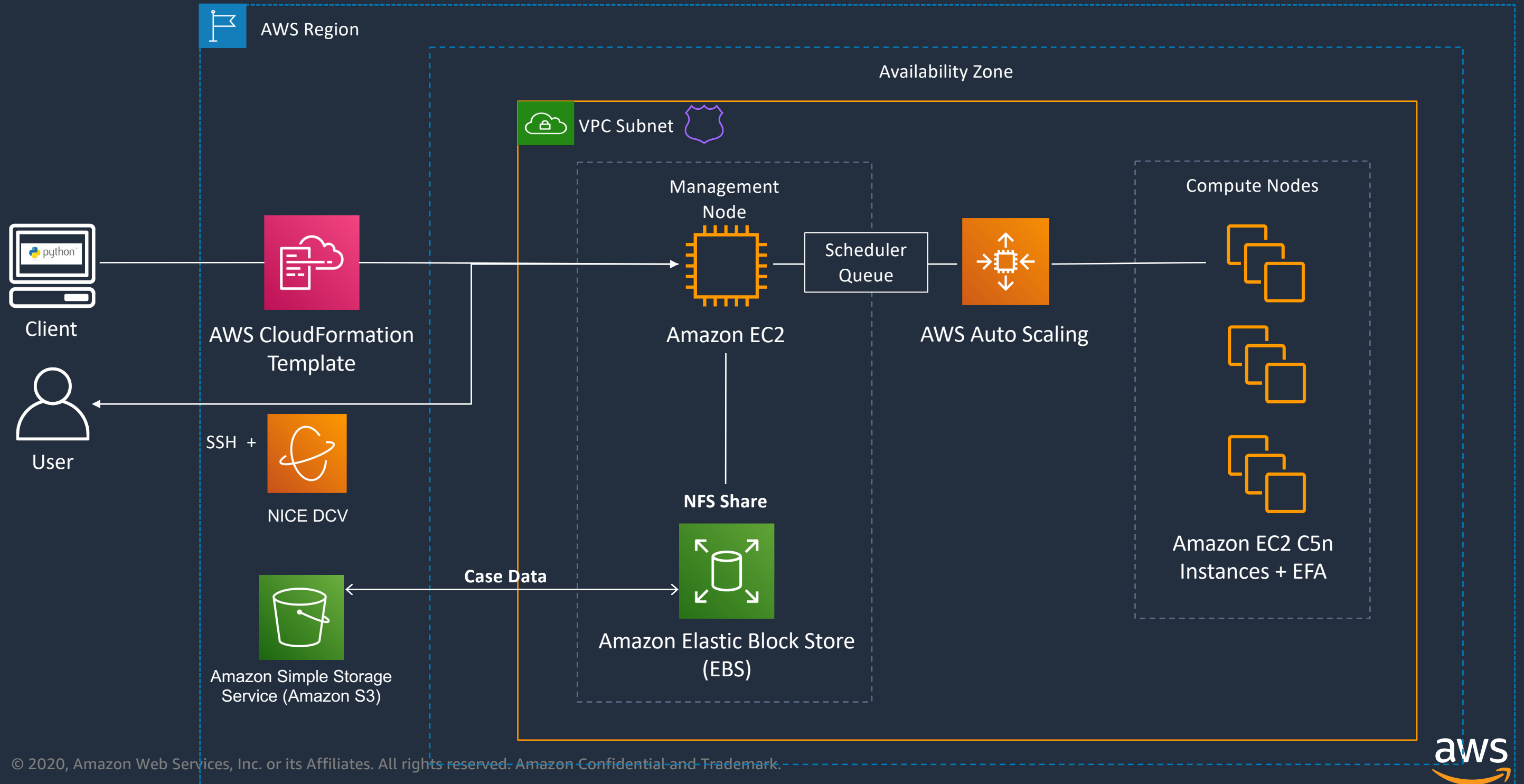
The duration of the workshop depends on how much you want to do. If you stick to just run the examples it should

I. Introduction
II. Setup
III. Verify
IV. Spack Setup
V. Cleanup

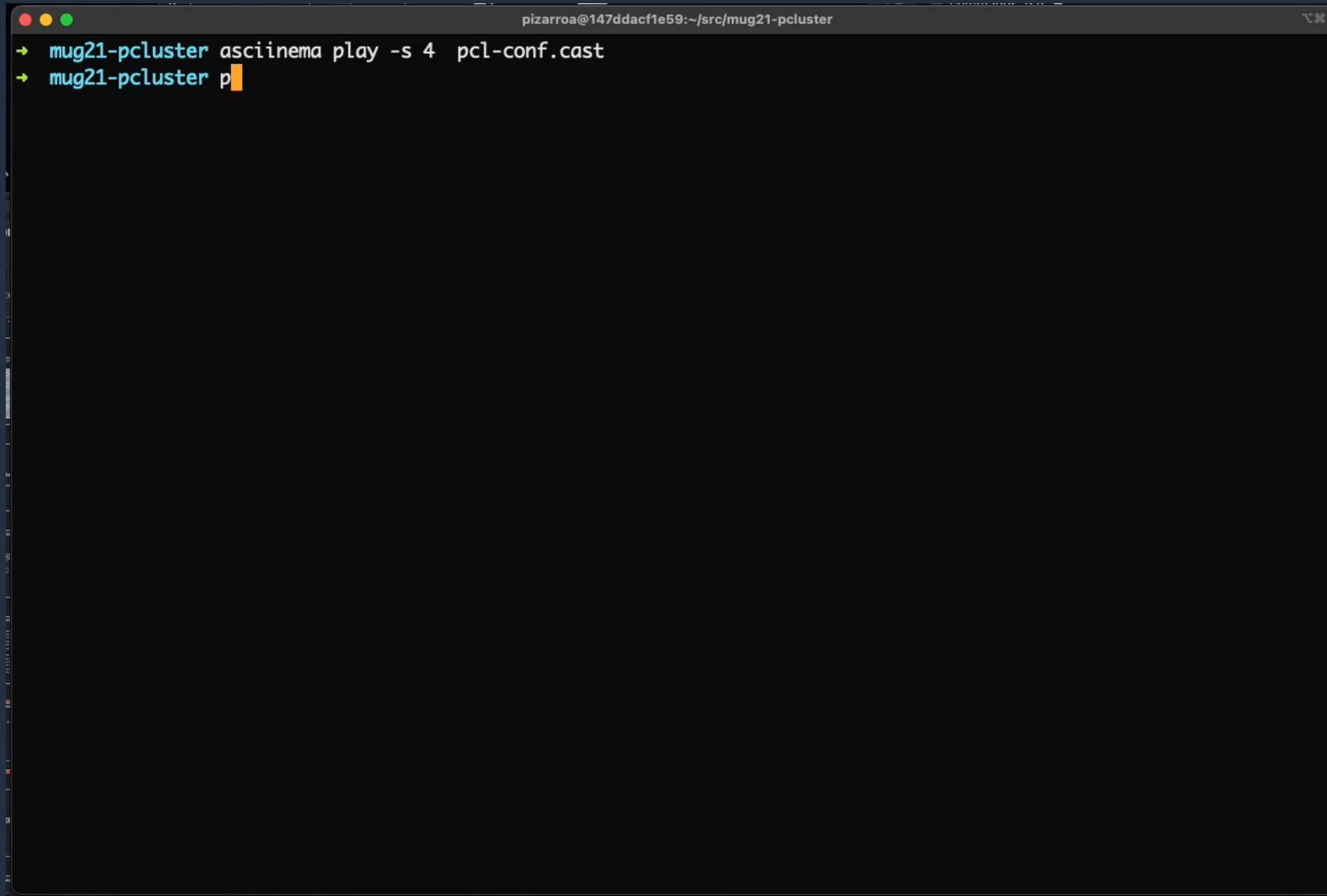
Privacy | Site Terms | © 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

<https://workshops.aws/categories/HPC>

AWS ParallelCluster architecture



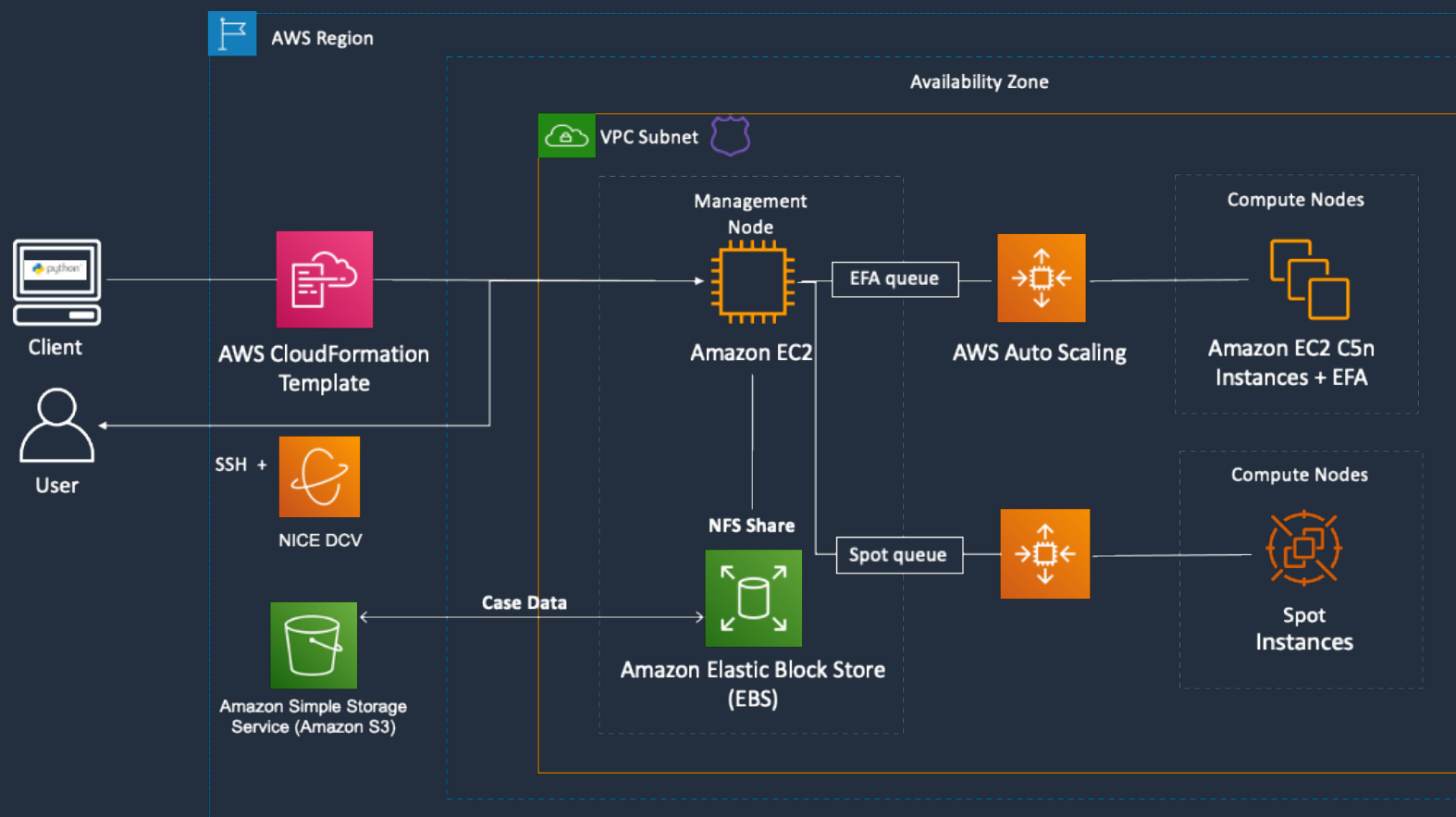
DEMO TIME (sort of 😄💧)



A terminal window with a dark background and light green text. The window title bar shows the user 'pizarroa' and the directory '~/src/mug21-pcluster'. Two commands are entered, each preceded by a green arrow prompt. The first command is 'mug21-pcluster asciinema play -s 4 pcl-conf.cast'. The second command is 'mug21-pcluster p' followed by a cursor.

```
pizarroa@147ddacf1e59:~/src/mug21-pcluster
→ mug21-pcluster asciinema play -s 4 pcl-conf.cast
→ mug21-pcluster p
```

Feature: Multiple Slurm Queues

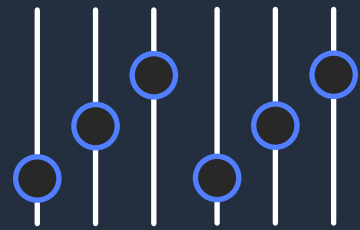


```
12 [scaling demo]
13 scaledown_idletime = 5           # optional, defaults to 10 minutes
14
15 [cluster multi-queue]
16 key_name = pc-key-mug21
17 base_os = alinux2
18 scheduler = slurm
19 master_instance_type = c5.2xlarge
20 vpc_settings = default
21 scaling_settings = demo
22 queue_settings = spot,ondemand
23
24 [queue spot]
25 compute_resource_settings = spot_i1,spot_i2
26 compute_type = spot             # optional, defaults to ondemand
27
28 [compute_resource spot_i1]
29 instance_type = c5.xlarge
30 min_count = 0                   # optional, defaults to 0
31 max_count = 10                  # optional, defaults to 10
32
33 [compute_resource spot_i2]
34 instance_type = t3.medium
35 min_count = 1
36 initial_count = 2
37
38 [queue ondemand]
39 compute_resource_settings = ondemand_i1
40 disable_hyperthreading = true   # optional, defaults to false
```

<https://docs.aws.amazon.com/parallelcluster/latest/ug/tutorial-mqm.html>

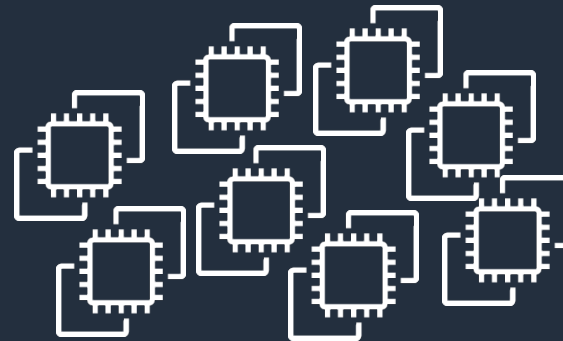
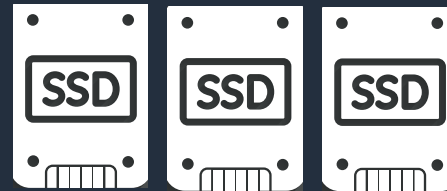
Feature: Amazon FSx for Lustre

Parallel file system



100+ GiB/s throughput
Millions of IOPS
Consistent sub-millisecond latencies

SSD-based

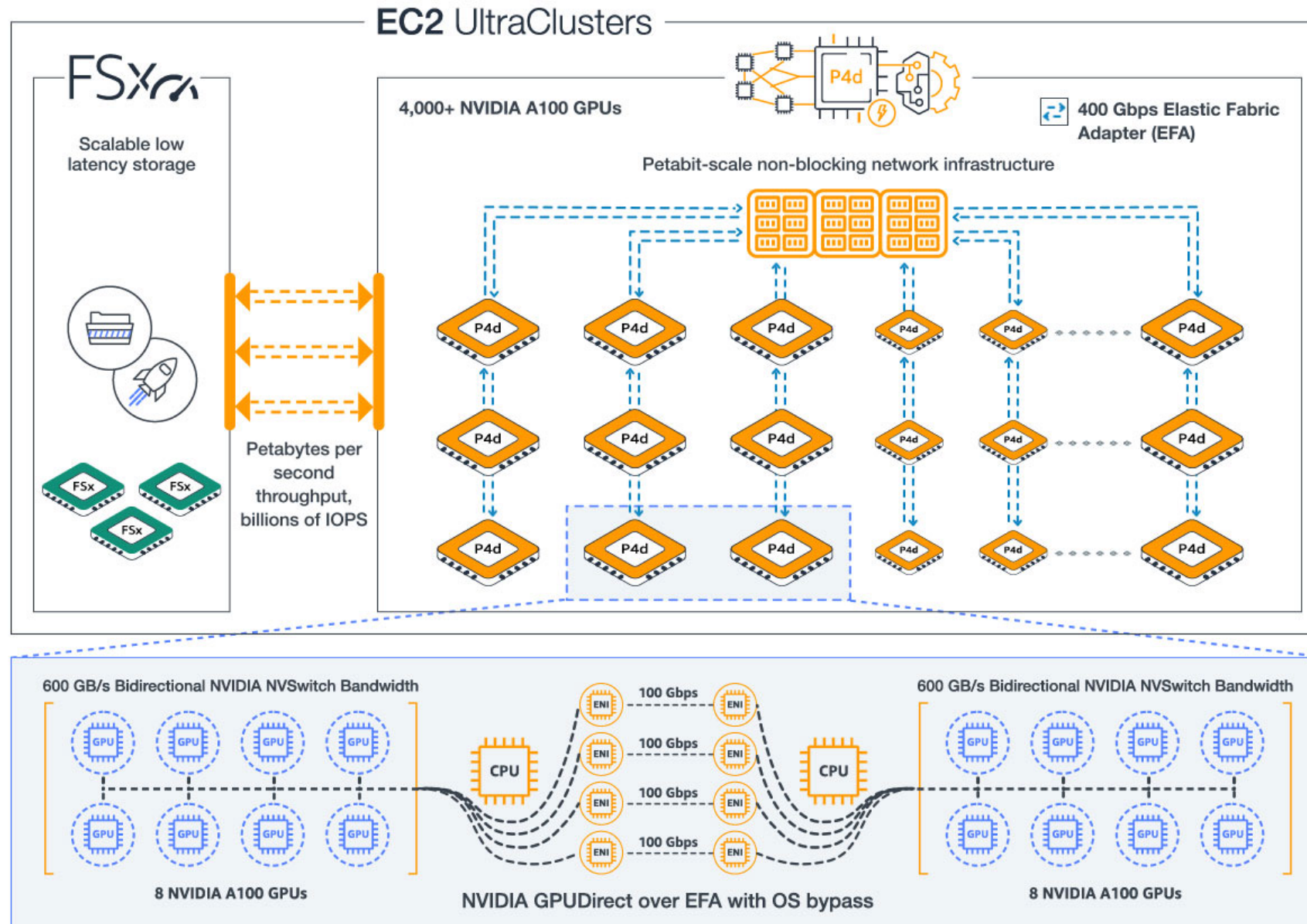


Supports hundreds of
thousands of cores

```
15 [cluster default]
16 key_name = pc-key-mug21
17 scheduler = slurm
18 master_instance_type = c5.2xlarge
19 base_os = alinux2
20 vpc_settings = default
21 queue_settings = compute
22 scaling_settings = demo
23 fsx_settings = fs-mug21
24
25 [fsx fs-mug21]
26 shared_dir = /fsx
27 storage_capacity = 3600
28 imported_file_chunk_size = 1024
29 export_path = s3://bucket/folder
30 import_path = s3://bucket
31 weekly_maintenance_start_time = 1:00:00
32
```

EC2 UltraClusters of P4d

Supercomputing-class performance for deep learning workflows



- Based on NVIDIA A100 GPUs
- Availability to scale-out to large clusters for distributed training
- 2.5x better deep learning performance and 60% lower cost to train
- EC2 UltraClusters with EFA enables 400 Gbps and allows you to scale to over 4,000 GPUs

Thank you!

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

<https://workshops.aws/categories/HPC>

<https://spack-tutorial.workshop.aws/>

<https://www.hpcworkshops.com/>

<https://mvapich.cse.ohio-state.edu/userguide/mv2x-aws/>

https://mvapich.cse.ohio-state.edu/userguide/userguide_spack/