
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

CMP305

Best practices for high performance computing in the cloud

Maxime Hugues

Principal HPC Specialist Solutions Architect
AWS

Lowell Wofford

Principal Solutions Architect
AWS

Agenda

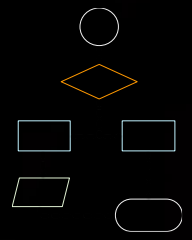
Presentation: Fundamentals of infrastructure for HPC and AWS ParallelCluster

Hands-on: Create a HPC cluster using AWS ParallelCluster

Presentation: Amazon FSx for Lustre and application management using Spack

Hands-on: HPC cluster update to dynamically mount Amazon FSx for Lustre

Compute and orchestration building blocks

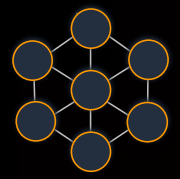


Workflow, notifications, queues

- Workflow management
- Notification and message queues

Workflow management and communication

Abstraction



Serverless & containers

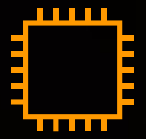
- Event-driven functions (AWS Lambda)
- Batch schedulers; containers orchestrators

Focus on the workload and not infrastructure

Provisioning

- Amazon EC2 Auto Scaling groups: scale up & down
- Instance fleets: capacity at scale across AZs

Compute on events or requests



Instances

- Virtualized
- Bare metal

} Different capabilities (CPU, RAM, SSD, network, and accelerators)

Base compute layer

Base infrastructure

VPC and subnets

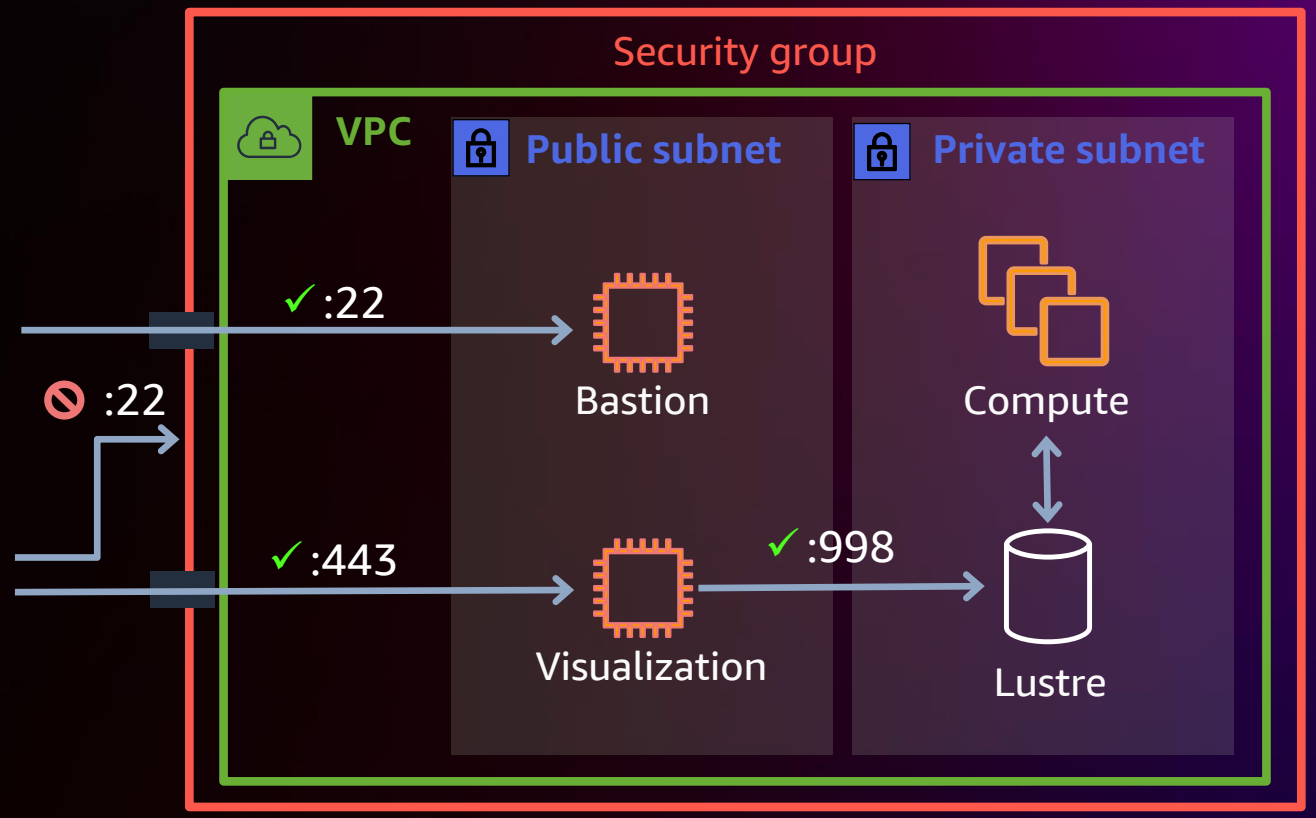
- Virtual private cloud: logical section of the cloud provider infrastructure
- Subnet: logical partition of a network

Security groups

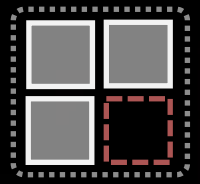
- Virtual firewalls
- VPC & instances

Instances and services

- Instances
- Managed services
- . . .

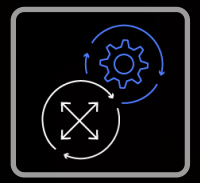


Amazon EC2 Auto Scaling in more detail



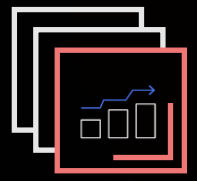
Logical unit

Purpose of scaling or management



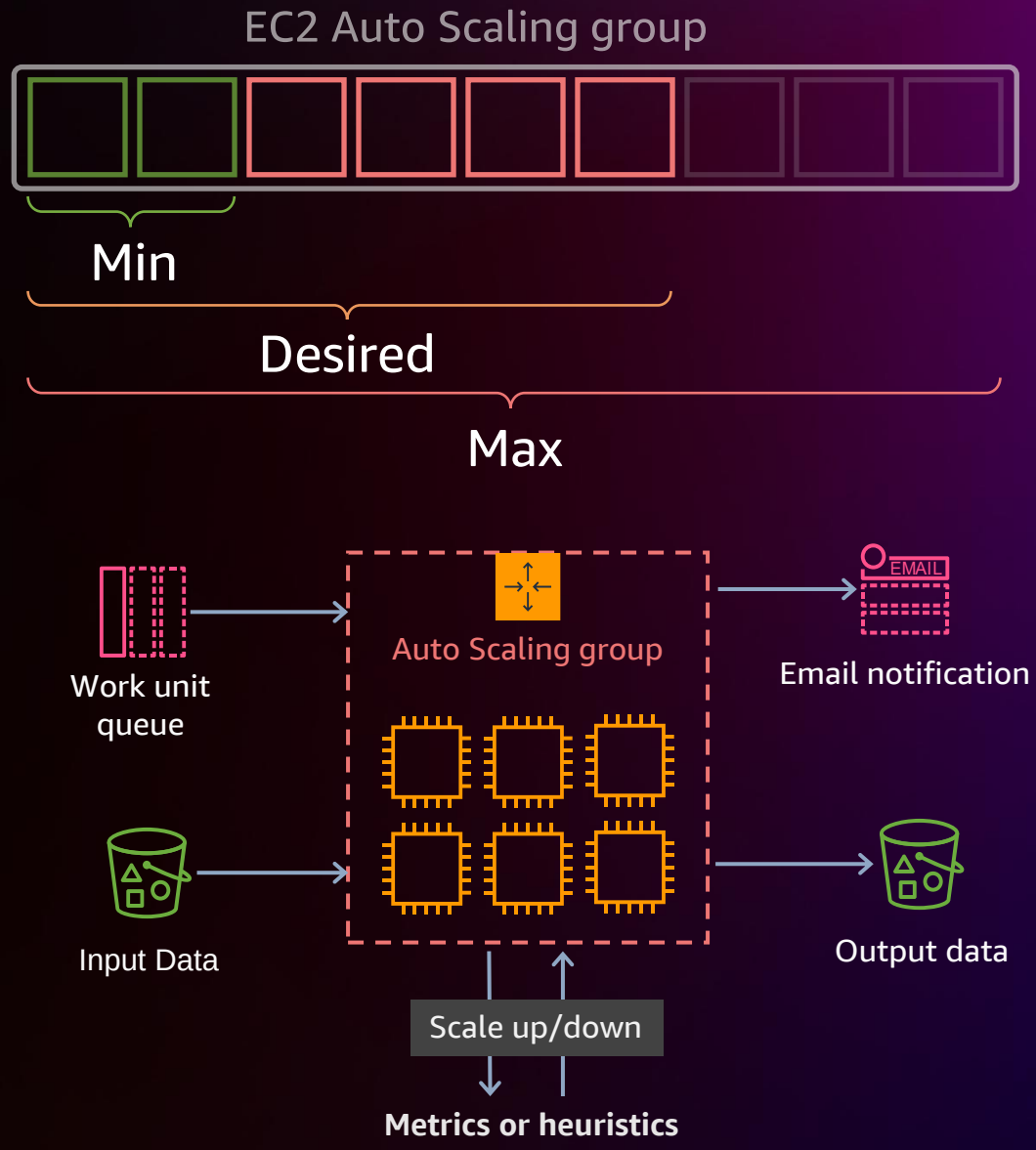
Launch templates

Kind, size, storage, SSH key pair, user data, and security groups



Scaling options

- Manual, schedule, predictive
- Notify on start, stop, terminate . . .



EC2 Auto Scaling compute system?

Imagine that nodes are added when jobs are submitted and removed when they finish

	Start	0 instances in the cluster, 25 max authorized
	Submit	Job 1, request 288 cores, 8 instances
	Provision	8 instances and execute Job 1
	Finalize	Job 1
		Idle 5 min
But after 2 minutes	Submit	Job 2, request 864 cores, 24 instances
	Provision	16 new instances and execute Job 2
	Finalize	Job 2
		Idle 5 min
	Terminate	24 instances
	Stop	0 instances in the cluster

Auto Scaling compute cluster

Min: 0 **Desired: 04** Max: 25

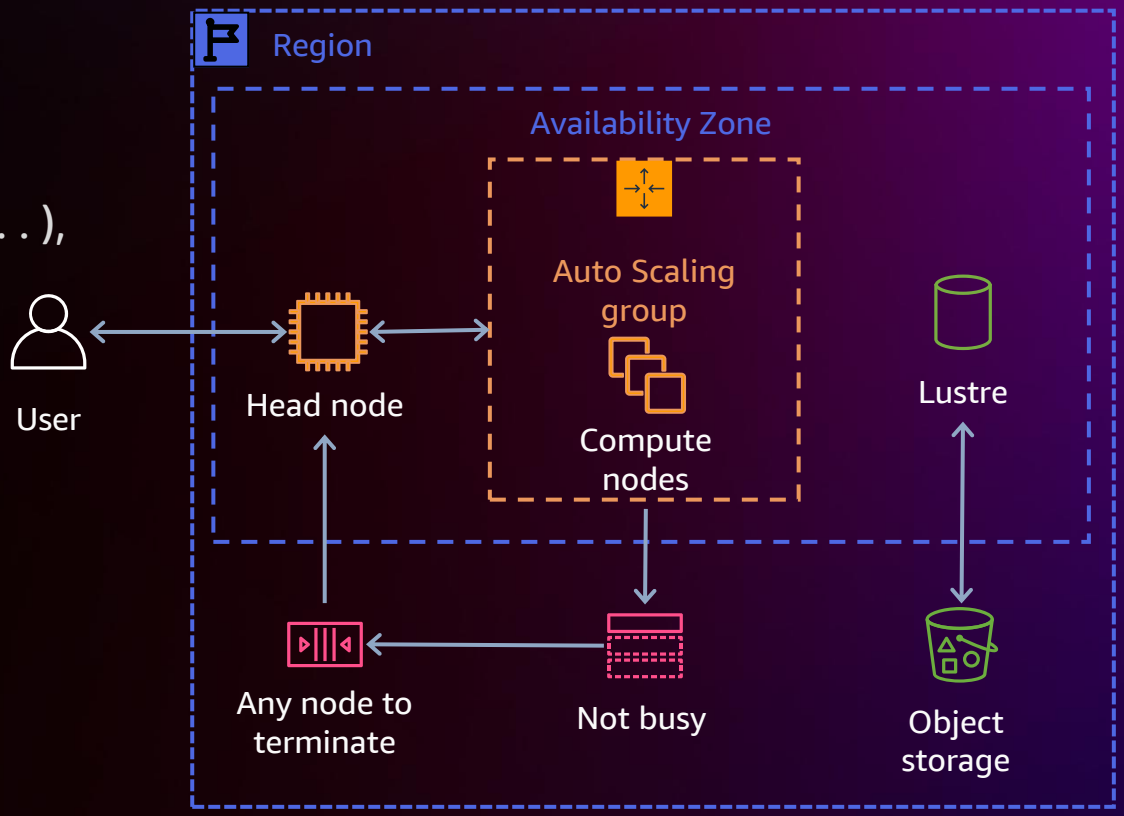
cores / node: 36

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg-scaling-group \
    --desired-capacity 04 \
    --max-size 25 \
    --vpc-zone-identifier "subnet-5ea0c127"
```

Putting it all together

Building an auto-scaling HPC cluster

- Similar to on premises but with auto scaling
- Still a classical HPC system with a scheduler (SLURM, SGE, . . .), Lustre, and placement groups (tightly coupled)
- The same familiar interface with an elastic capacity

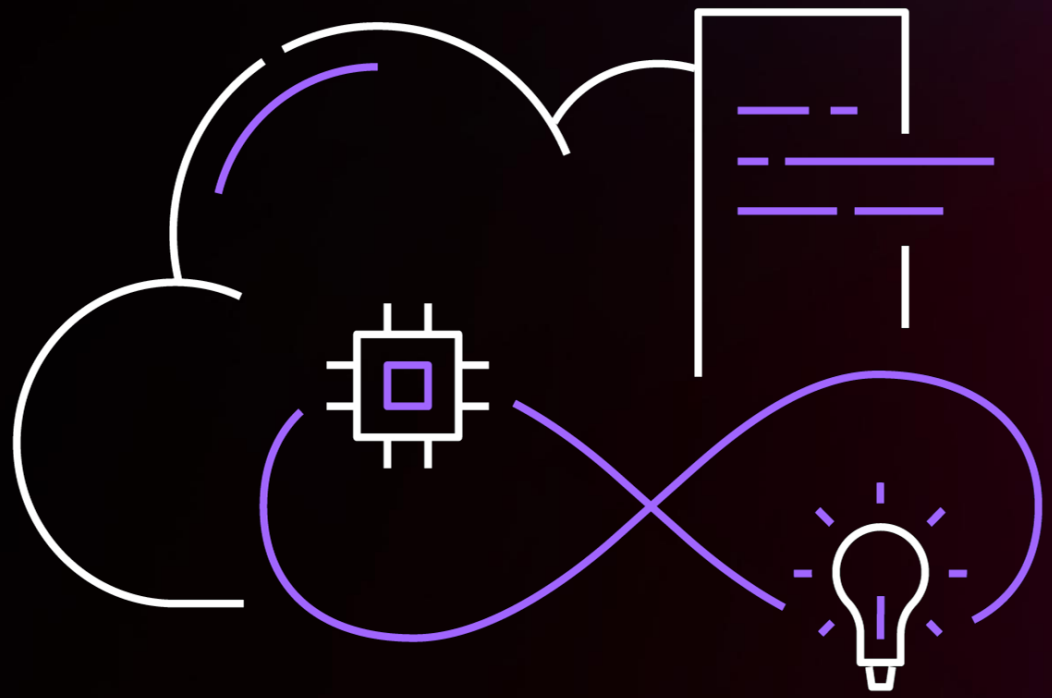


Additional technical considerations

When ready, instances send notifications to a message queuing service – the scheduler watches this queue and adds the compute nodes as they appear

When not busy, they will lock themselves up, check the scheduler queue, send a notification, and terminate

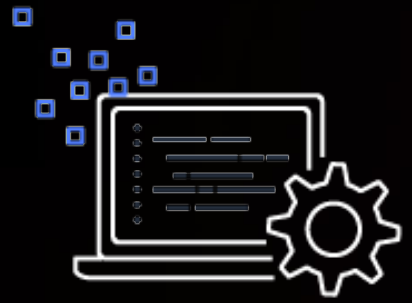
Part of the next hands-on



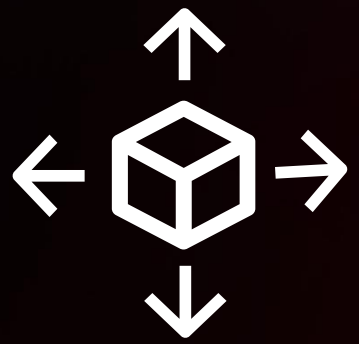
AWS ParallelCluster

Open-source cluster management for HPC on AWS

Why use AWS ParallelCluster?



Easy cluster management



Automatic resource scaling

Seamless migration to the cloud

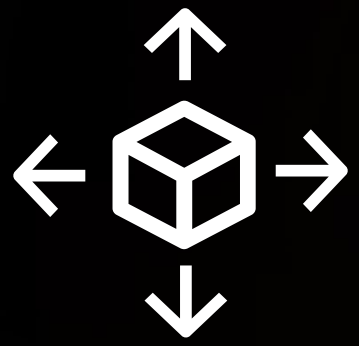
Easy cluster management

“pcluster configure” to set up a cluster in minutes

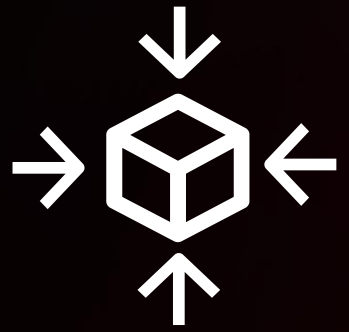
Use config files to define details of replicable clusters

Launch, stop, and restart clusters on demand

Automatic resource scaling



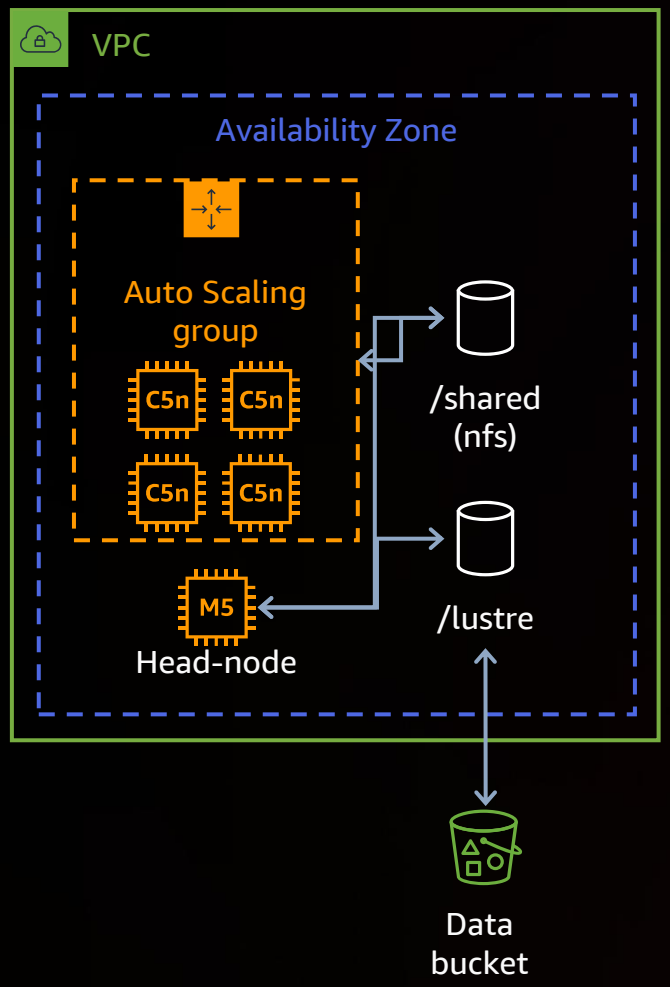
Scale up when jobs are waiting



Scale down when the cluster is idle

Your data storage and file system scale to match your compute

Simple architecture



- Post install configuration
  - Install applications scripts
- Amazon EBS snapshot bootstrap
  - Application installations or static configurations
- Other details
  - Amazon EFS can be shared across clusters
  - Lustre partition can be mounted but per AZ
  - Public/private subnets for head/compute
  - Link to AD for user mapping if required

Example of configuration file

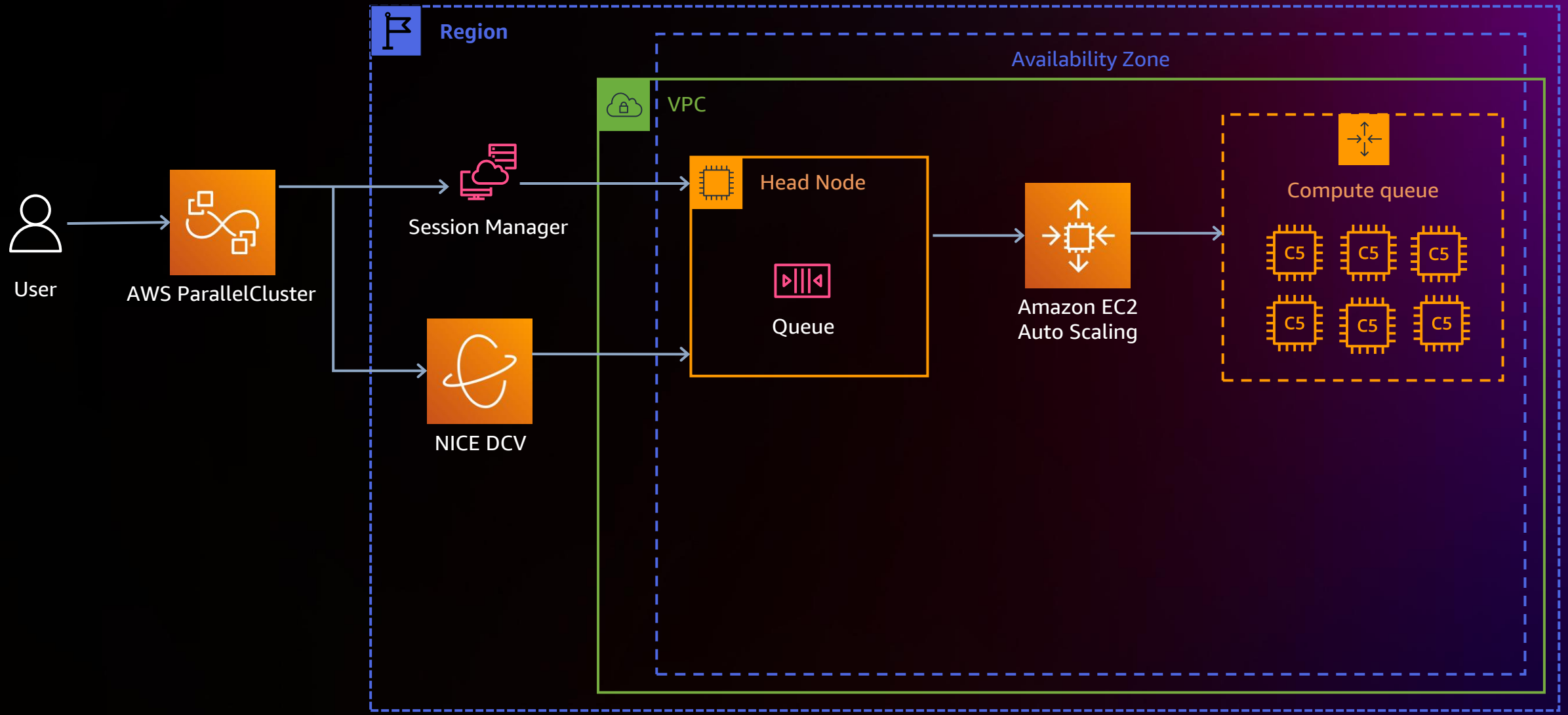
```
Region: [AWS_REGION]
Image:
  Os: alinux2
SharedStorage:
  - Name: Ebs0
    StorageType: Ebs
    MountDir: /shared
    EbsSettings:
      VolumeType: gp3
      DeletionPolicy: Delete
      Size: '100’

HeadNode:
  InstanceType: c5.xlarge
  Ssh:
    KeyName: [KEY_NAME]
  Networking:
    SubnetId: [subnet-12345689]
  Dcv:
    Enabled: true
  LocalStorage:
    RootVolume:
      VolumeType: gp2
```

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: compute-queue
      ComputeResources:
        - Name: hpc6a
          MinCount: 0
          MaxCount: 4
         InstanceType: hpc6a.48xlarge
          Efa:
            Enabled: true
          DisableSimultaneousMultithreading: true
      Networking:
        SubnetIds:
          - [subnet-12345689]
        PlacementGroup:
          Enabled: true
```

Hands-on: Create a HPC cluster using AWS ParallelCluster

Architecture for this workshop

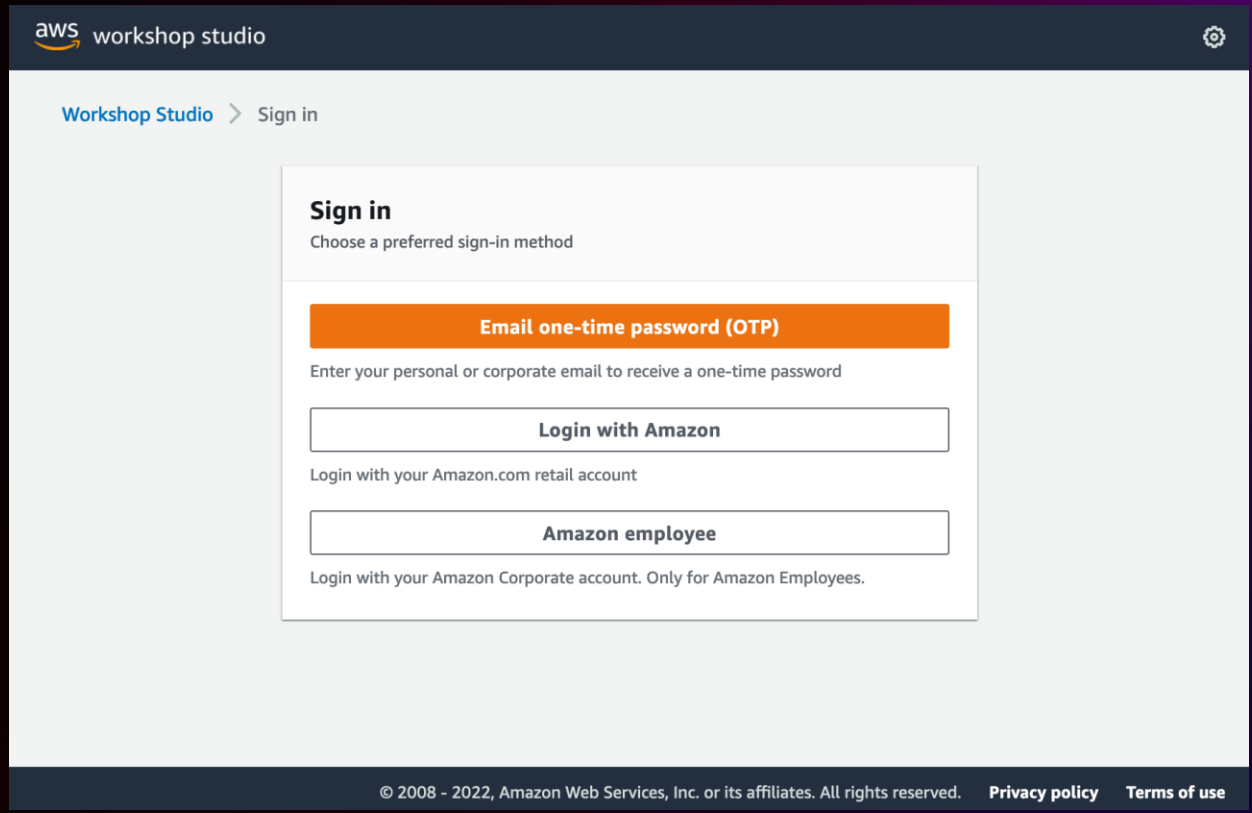


Getting started with this workshop

- As a participant, you will have access to an AWS account with any optional pre-provisioned infrastructure and IAM policies needed to complete this workshop
- The AWS account will only be available for the duration of this workshop; you will lose access to the account thereafter
- The optional pre-provisioned infrastructure will be deployed to a specific region; check your workshop content to determine whether other regions will be used
- Be sure to review the terms and conditions of the event; do not upload any personal or confidential information in the account

Step 1: Sign-in via your preferred method

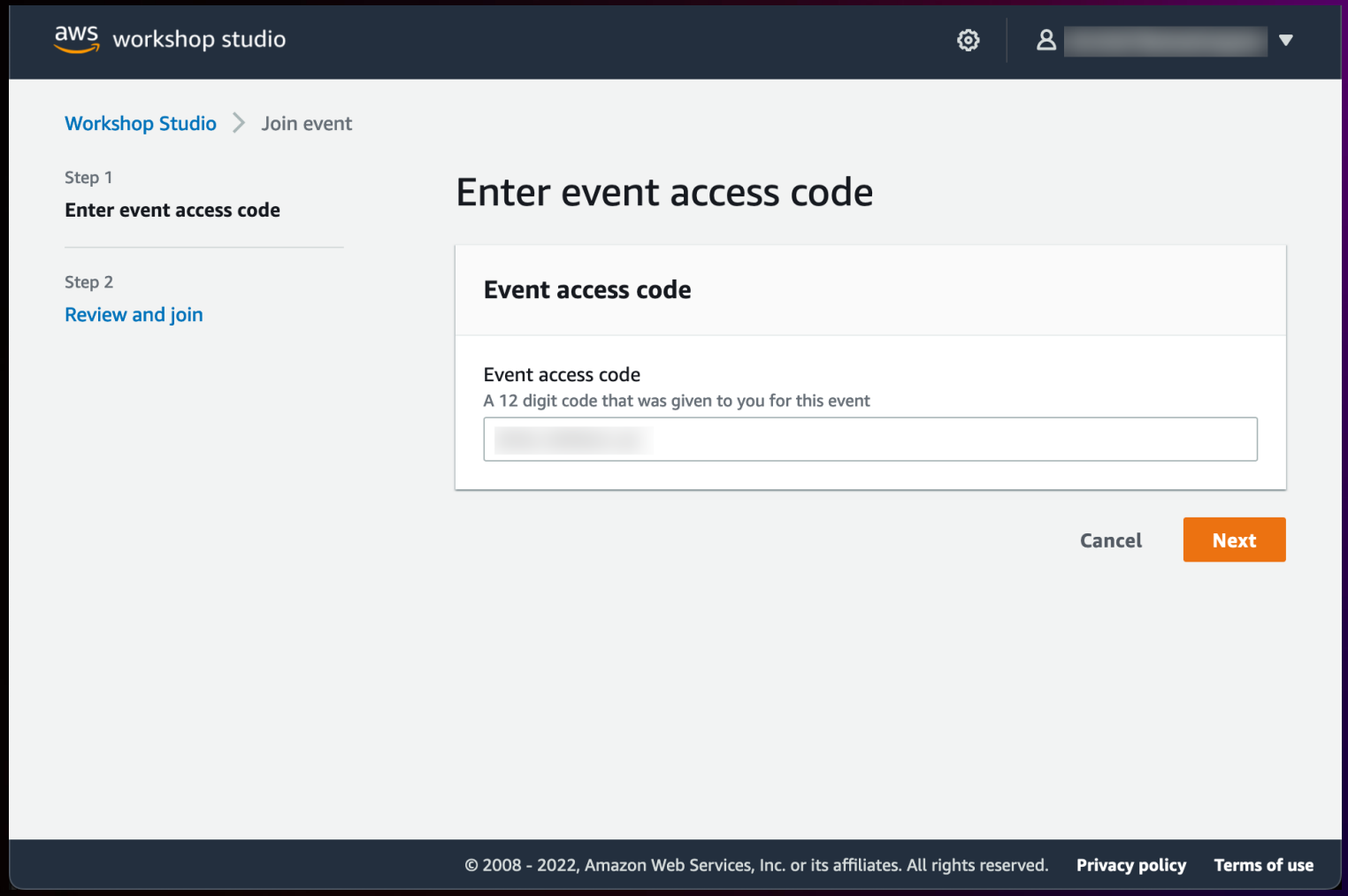
https://catalog.workshops.aws/join



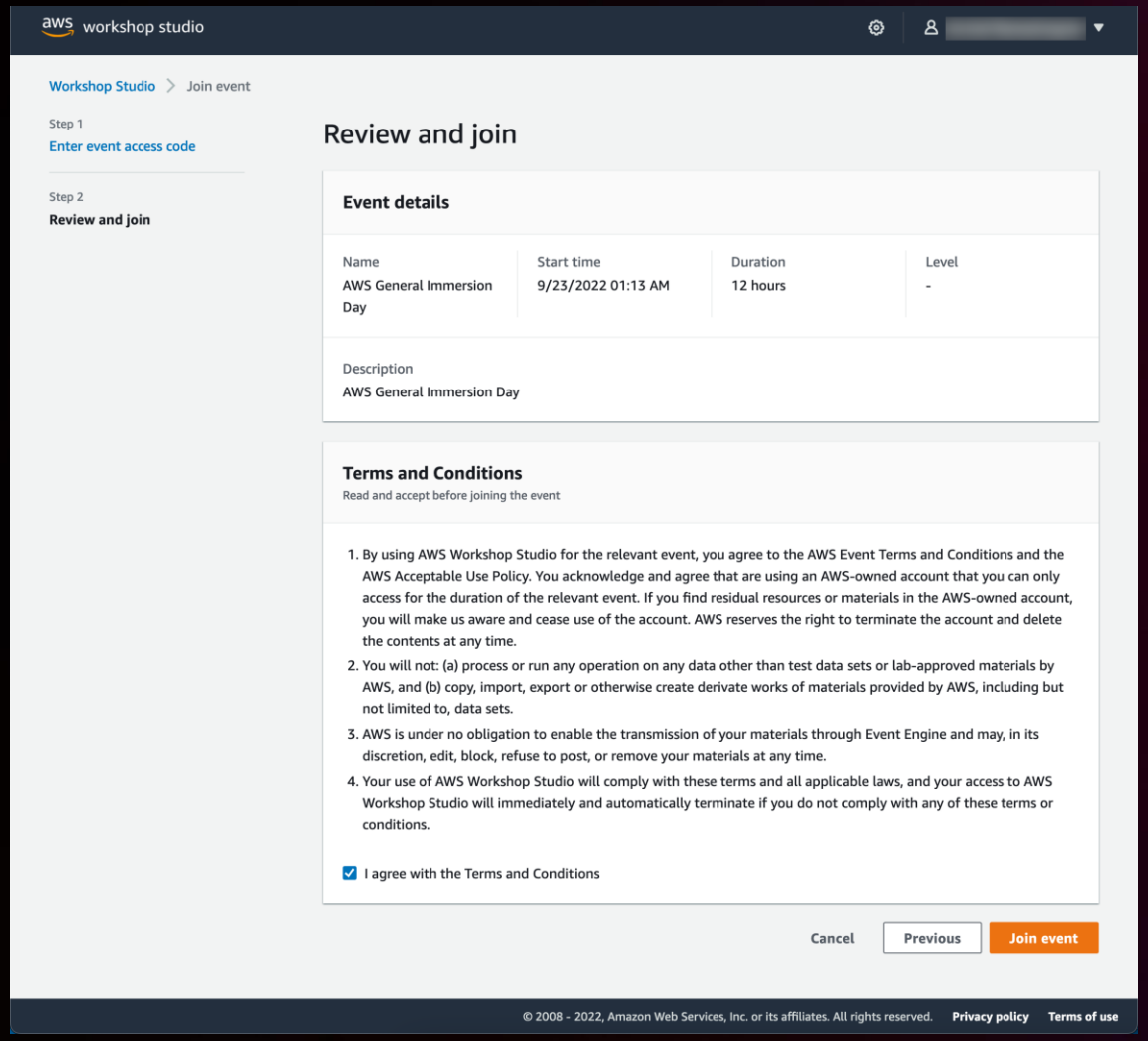
Step 2: Enter event access code

Enter 12-digit event access code

If you were given a one-click join link, you can skip this step

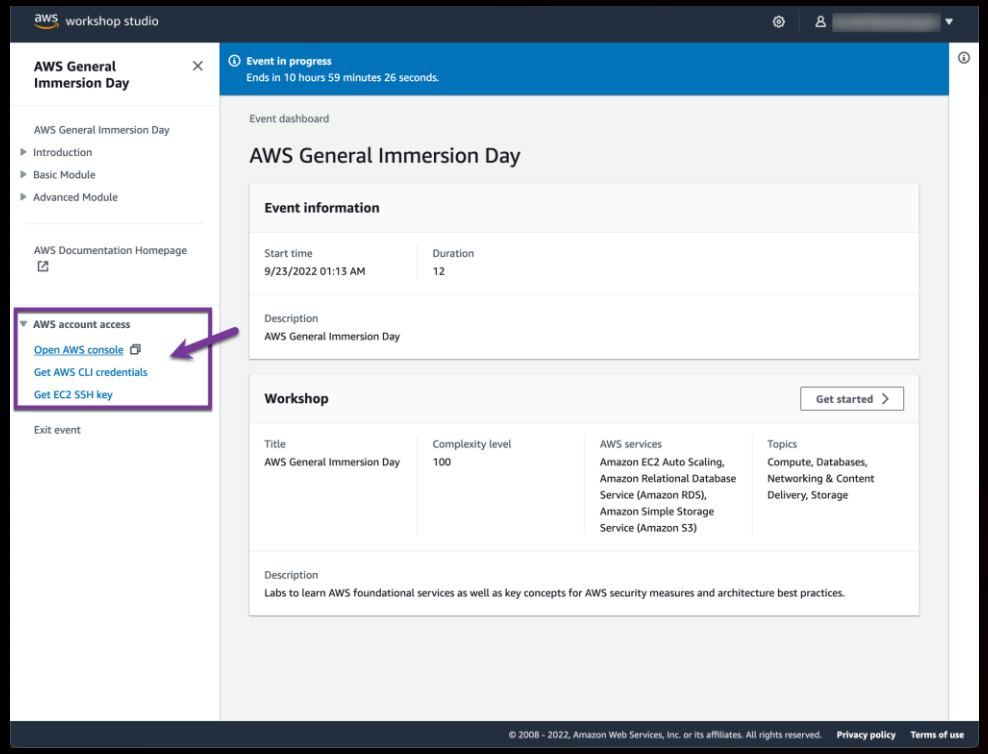


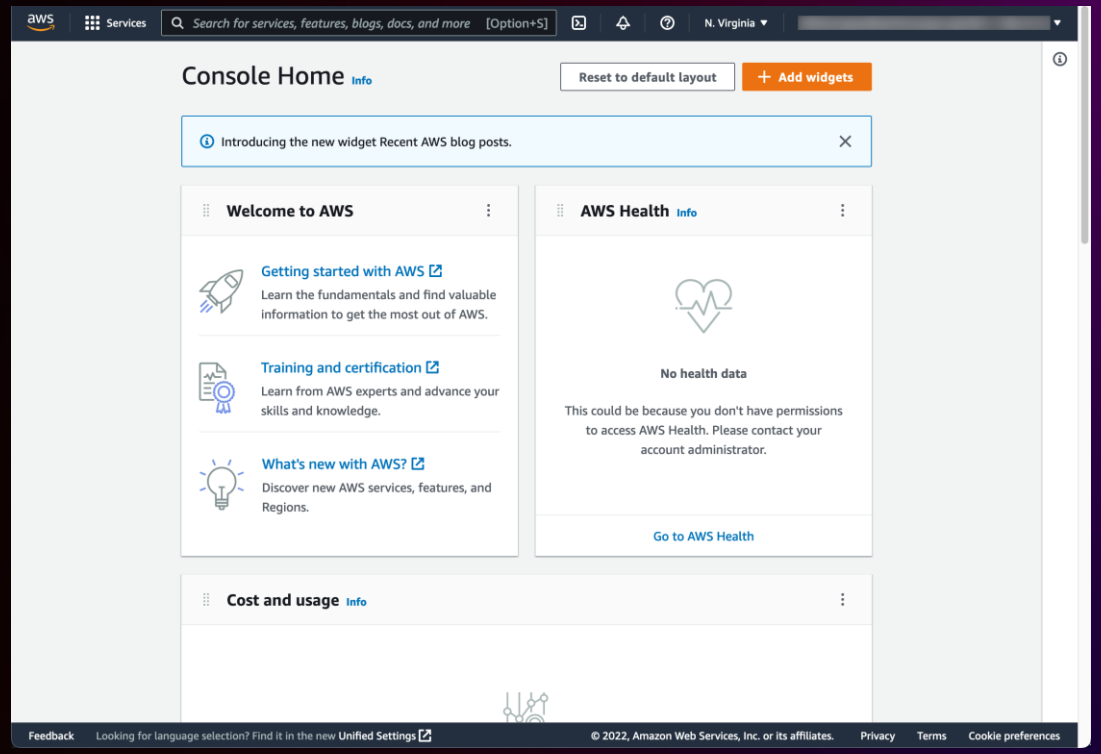
Step 3: Review terms and join event



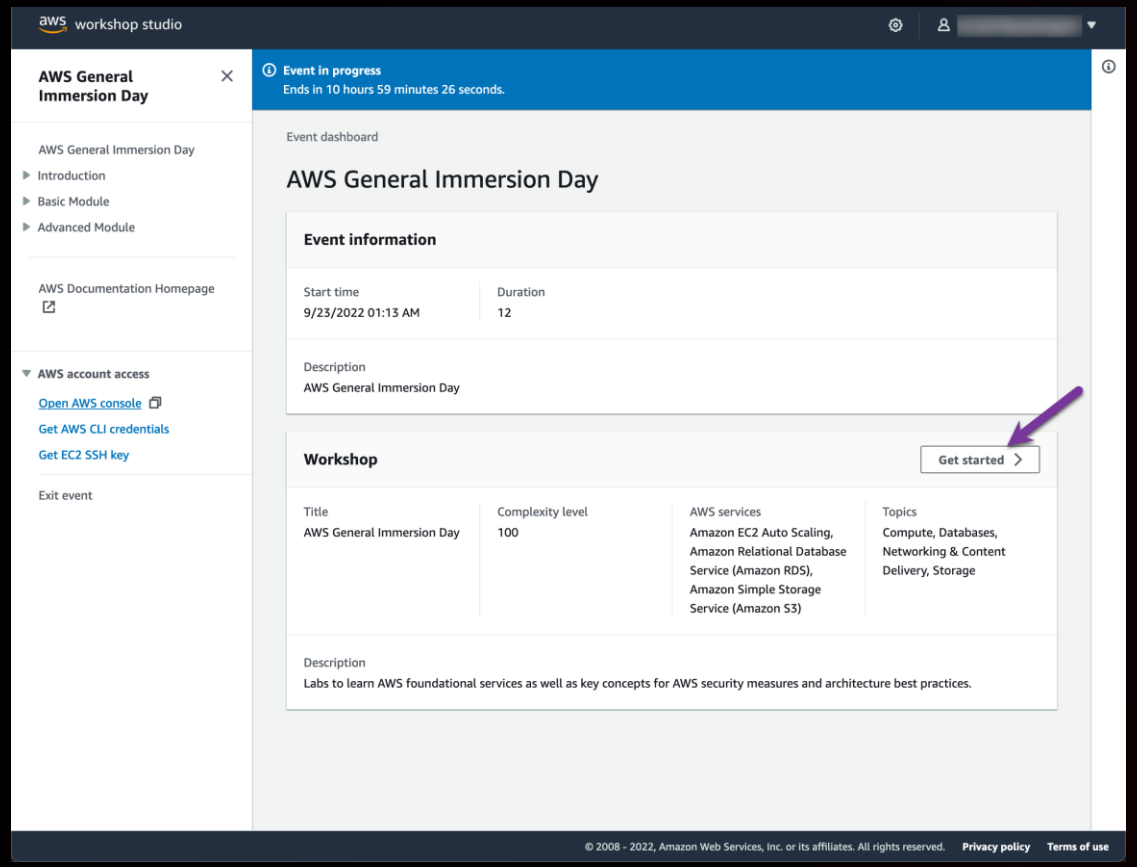
Step 4: Access AWS account

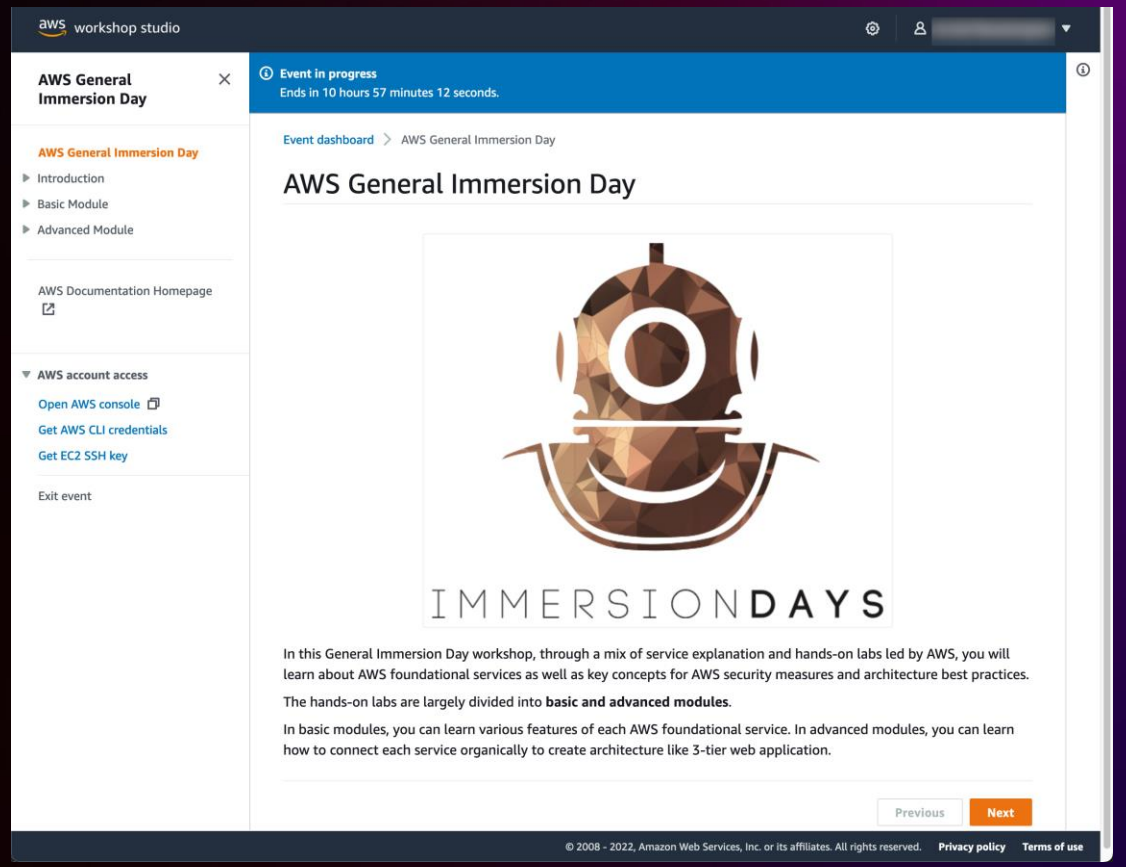
Access the AWS Management Console or generate AWS CLI credentials as needed





Step 5: Get started with the workshop



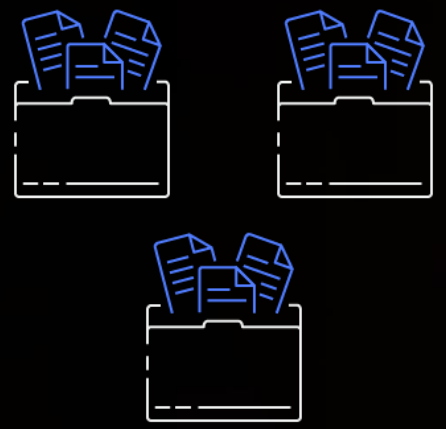


Amazon FSx for Lustre

Storage should not be the bottleneck for compute-intensive workloads

Faster storage reduces time to results and overall TCO by saving compute costs

Vast datasets



Ability to serve data quickly:

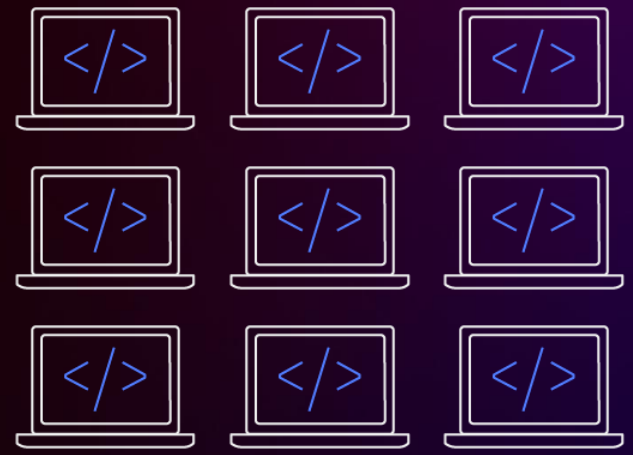
Fast storage

High-speed network

Low latency

Scale out compute capacity

(hundreds to millions of cores)



Amazon FSx for Lustre

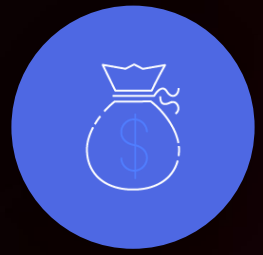
Fully managed Lustre file system for high-performance workloads

Massively scalable performance

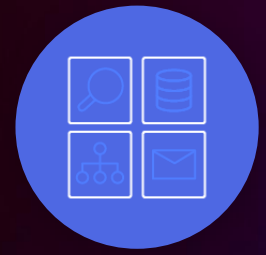
Seamless access to your data repositories

Simple and fully managed

Native file system interface



Cost optimized for compute-intensive workloads

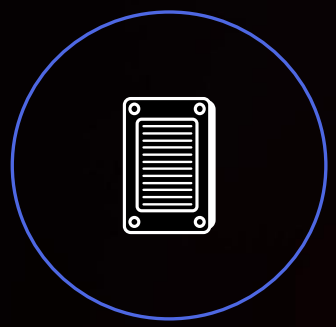


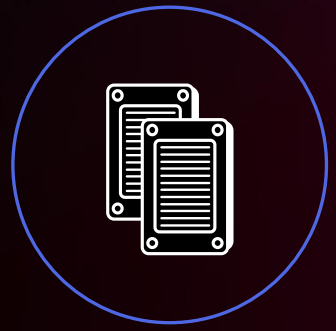
Secure and compliant

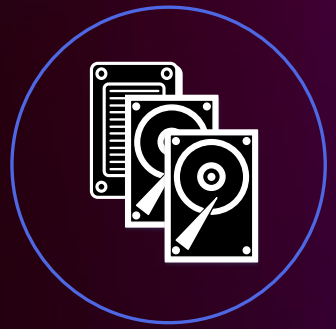
High, scalable performance

High, scalable performance

SSD | SSD or HDD







Scratch

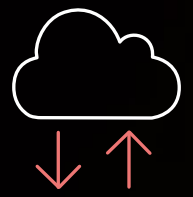
Short-term processing
Spin up → process → spin down
Single copy of data

Persistent

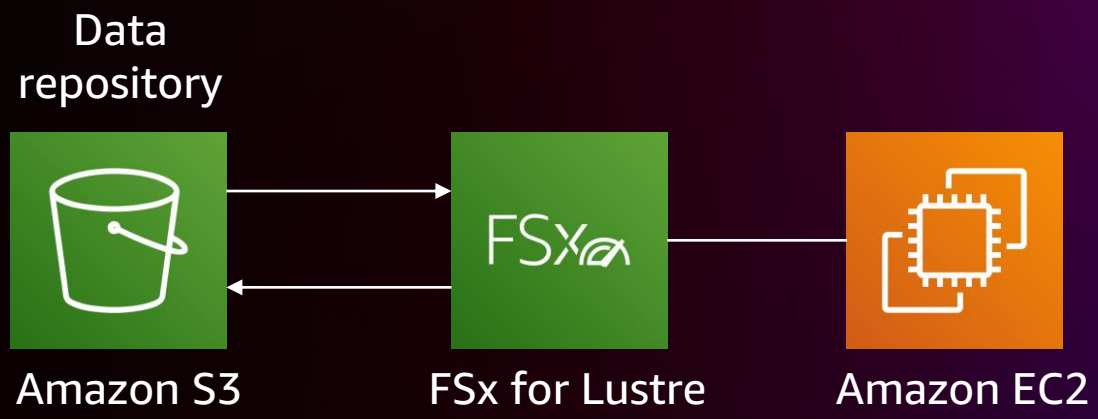
Longer-term processing
Highly available (HA) file servers
Replicated copies of data

Amazon FSx for Lustre control plane (API, management layer, file system control) is designed to be HA for both options

What FSx for Lustre brings to your workloads

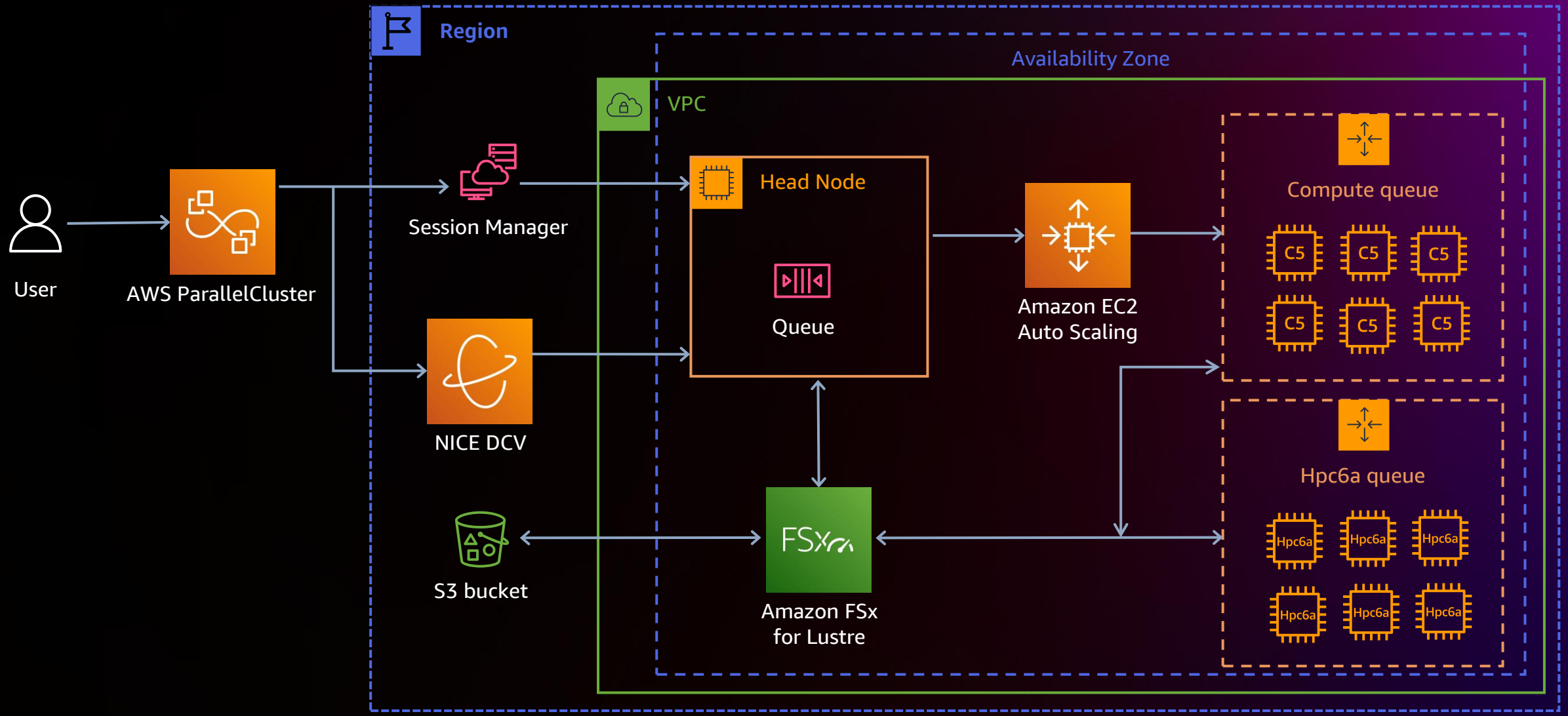


Access to Amazon S3 data through a fast file interface



Hands-on: HPC cluster update to use Amazon FSx for Lustre

Architecture for this workshop



Thank you!

Maxime Hugues
maxhaws@amazon.com

Lowell Wofford

Please complete the session survey in the **mobile app**