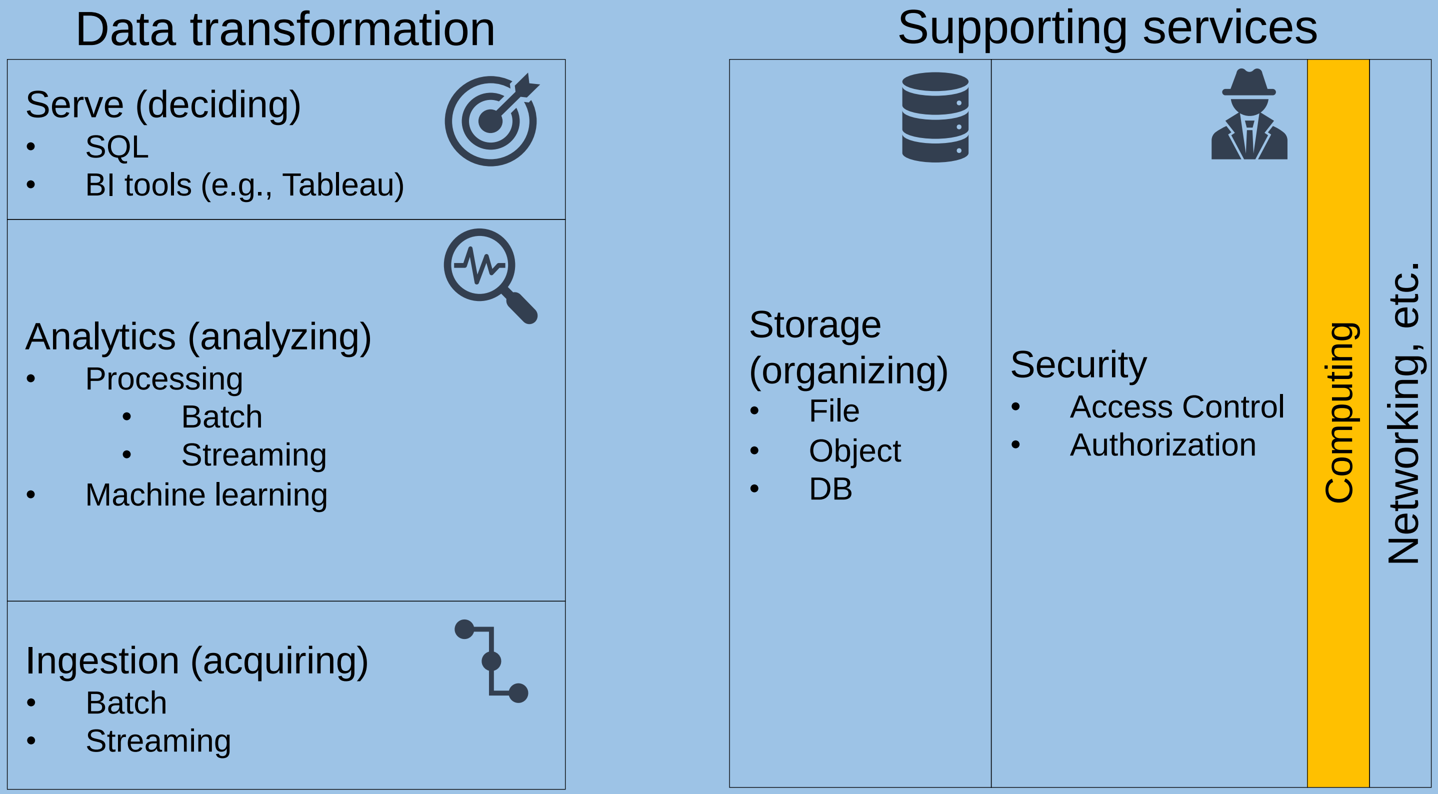
BIG DATA AND CLOUD PLATFORMS

Data pipelines on cloud (Computing)



Data transformation

Serve (deciding)
- SQL
- BI tools (e.g., Tableau)

Analytics (analyzing)
- Processing
  - Batch
  - Streaming
- Machine learning

Ingestion (acquiring)
- Batch
- Streaming

Supporting services

Storage (organizing)
- File
- Object
- DB

Security
- Access Control
- Authorization

Computing

Networking, etc.

Supporting data pipelines

We can choose the XaaS configuration to build our pipelines

IaaS

- Outsource virtual machines to the cloud (AWS EC2)
- (You) Manage technological and business challenges

PaaS

- Outsource the data ecosystem to the cloud (e.g., AWS EMR)
- (You) Manage business challenges

https://aws.amazon.com/emr

Single instance: AWS EC2

Amazon Elastic Compute Cloud

- A web service that provides resizable compute capacity
- Complete control of computing resources
    - Processor, storage, networking, OS, and purchase model

The **instance type** determines the hardware

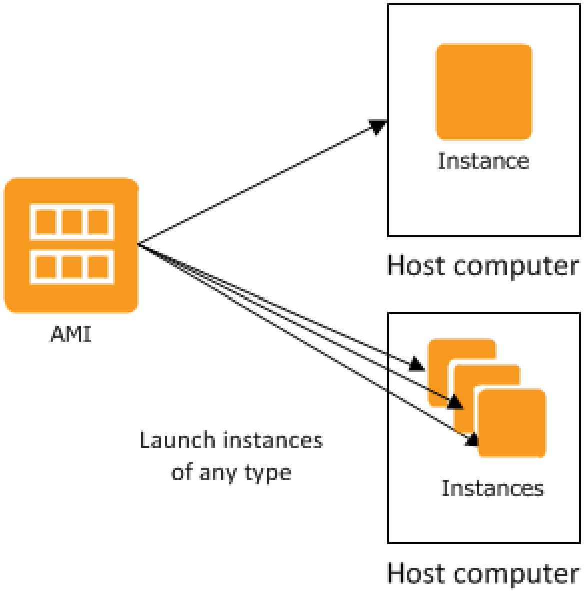
- Different compute and memory capabilities

Amazon Machine Image is a software template

- The EC2 instance is used for creating the virtual server instance
- The AMI is the EC2 virtual machines image

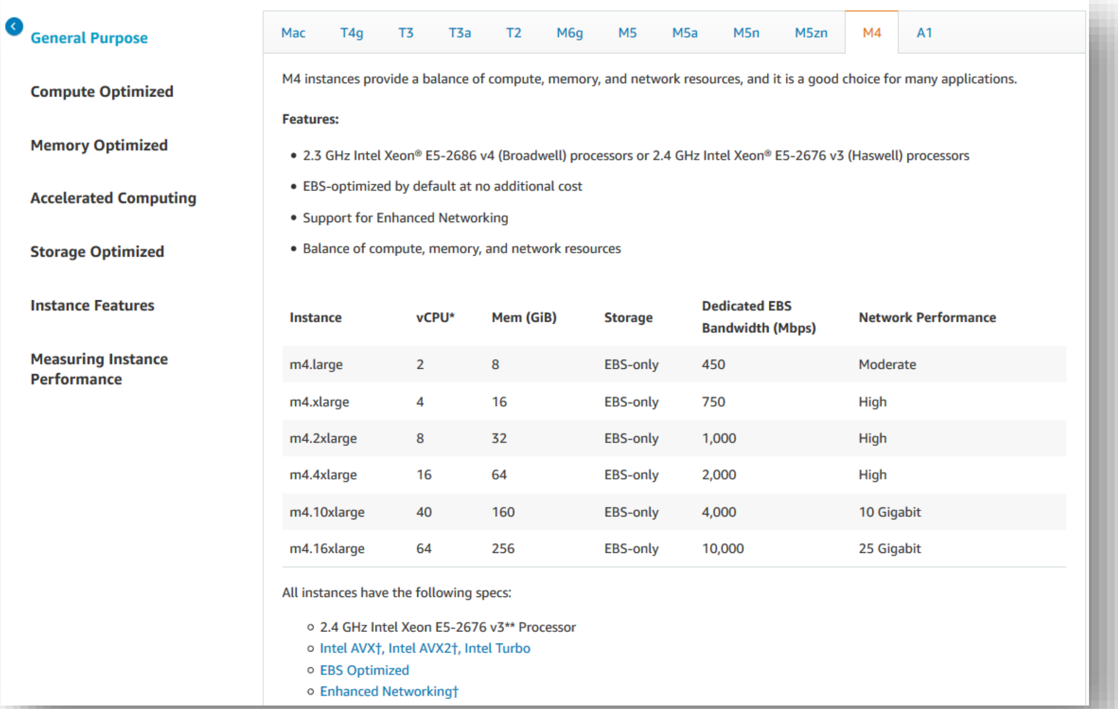
Interact with EC2 instance as with any computer

- You have complete control of your instances



https://aws.amazon.com/ec2/instance-types
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instances-and-amis.html
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/compute-optimized-instances.html

Single instance: AWS EC2



M4 instances provide a balance of compute, memory, and network resources, and it is a good choice for many applications.

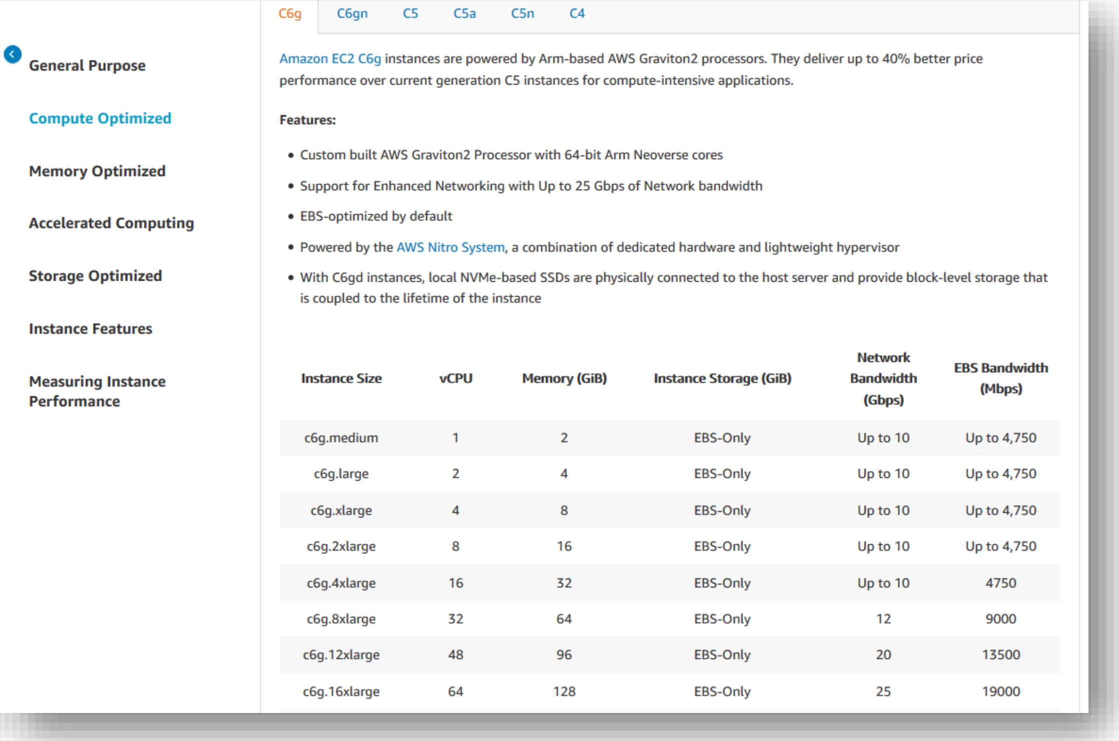
Features:

- 2.3 GHz Intel Xeon® E5-2686 v4 (Broadwell) processors or 2.4 GHz Intel Xeon® E5-2676 v3 (Haswell) processors
- EBS-optimized by default at no additional cost
- Support for Enhanced Networking
- Balance of compute, memory, and network resources

Instance	vCPU*	Mem (GiB)	Storage	Dedicated EBS Bandwidth (Mbps)	Network Performance
m4.large	2	8	EBS-only	450	Moderate
m4.xlarge	4	16	EBS-only	750	High
m4.2xlarge	8	32	EBS-only	1,000	High
m4.4xlarge	16	64	EBS-only	2,000	High
m4.10xlarge	40	160	EBS-only	4,000	10 Gigabit
m4.16xlarge	64	256	EBS-only	10,000	25 Gigabit

All instances have the following specs:

- 2.4 GHz Intel Xeon E5-2676 v3** Processor
- Intel AVX†, Intel AVX2†, Intel Turbo
- EBS Optimized
- Enhanced Networking†



Amazon EC2 C6g instances are powered by Arm-based AWS Graviton2 processors. They deliver up to 40% better price performance over current generation C5 instances for compute-intensive applications.

Features:

- Custom built AWS Graviton2 Processor with 64-bit Arm Neoverse cores
- Support for Enhanced Networking with Up to 25 Gbps of Network bandwidth
- EBS-optimized by default
- Powered by the AWS Nitro System, a combination of dedicated hardware and lightweight hypervisor
- With C6gd instances, local NVMe-based SSDs are physically connected to the host server and provide block-level storage that is coupled to the lifetime of the instance

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
c6g.medium	1	2	EBS-Only	Up to 10	Up to 4,750
c6g.large	2	4	EBS-Only	Up to 10	Up to 4,750
c6g.xlarge	4	8	EBS-Only	Up to 10	Up to 4,750
c6g.2xlarge	8	16	EBS-Only	Up to 10	Up to 4,750
c6g.4xlarge	16	32	EBS-Only	Up to 10	4750
c6g.8xlarge	32	64	EBS-Only	12	9000
c6g.12xlarge	48	96	EBS-Only	20	13500
c6g.16xlarge	64	128	EBS-Only	25	19000

https://aws.amazon.com/ec2/instance-types/

Cluster: AWS EMR

Amazon EMR is a data platform based on the Hadoop stack

- Apache Spark, Apache Hive, Apache HBase, etc.
- You can run workloads on
  - Amazon EC2 instances
  - Amazon Elastic Kubernetes Service (EKS) clusters

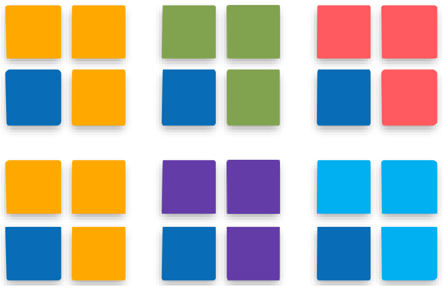
Example of workload

- Upload input data into Amazon S3
- EMR launches EC2 instances that you specified
- EMR begins the execution while pulling the input data from S3 into the launched instances
- Once the cluster is finished, EMR transfers output data to Amazon S3

AWS EMR

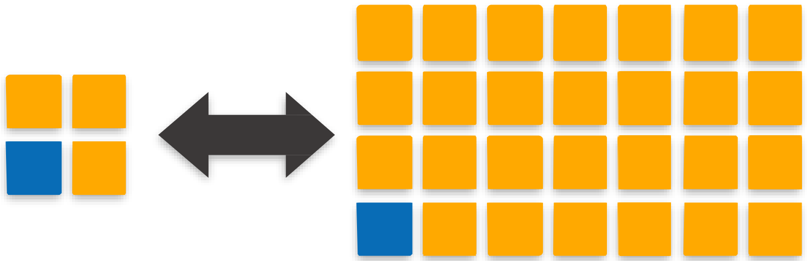
Provision as much capacity as you need

Deploy Multiple Clusters



Add or remove capacity at any time

Resize a Running Cluster



AWS EMR

EMR cluster

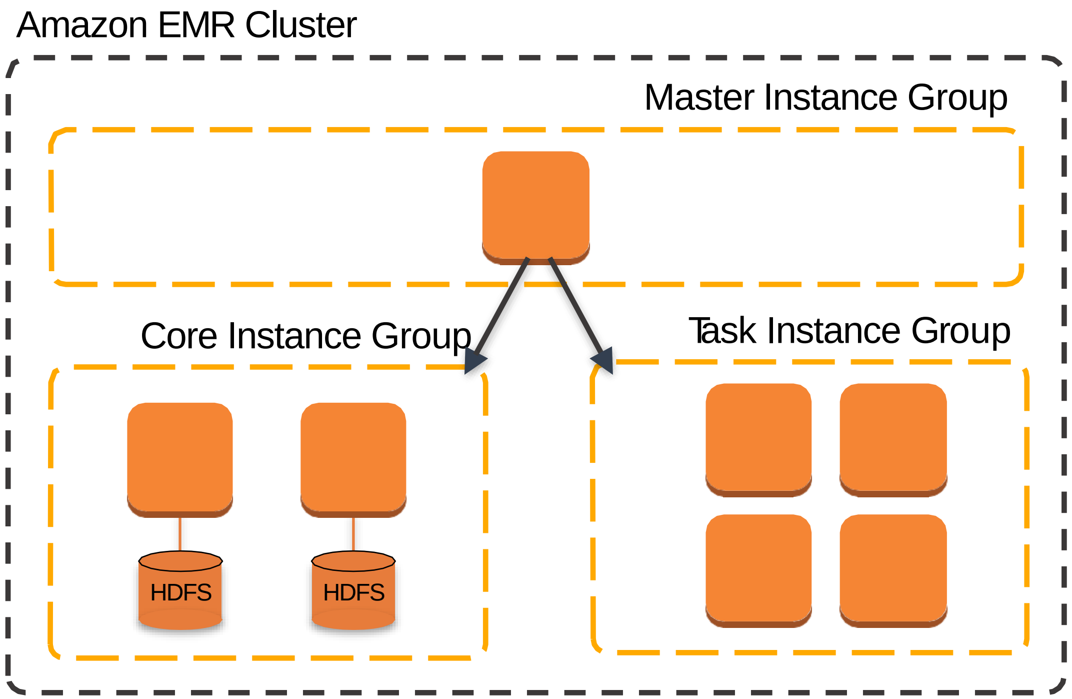
Master group controls the cluster

- Coordinate the work distribution
- Manage the cluster state

Core groups

- Core instances run Data Node daemons

(Optional) Task instances



AWS EMR

The central component of Amazon EMR is the **cluster**

- A collection of **Amazon Elastic Compute Cloud (Amazon EC2)** instances
- Each instance is called a **node**

The **node type** identifies the role within the cluster

- **Master** node coordinates the distribution of data and tasks among other nodes
  - Every cluster has (at least) a master node
  - Always active
- **Core** node runs tasks and store data in the Hadoop Distributed File System (HDFS)
  - Multi-node clusters have at least one core node
  - Always active, contains the data node daemon
- **Task** node only runs tasks
  - Task nodes are optional
  - Decoupling processing and storage, we lose data locality

AWS EMR

On-Demand Instance

- Pay for compute capacity by the hour (minimum of 60 seconds)
- No long-term commitments

Spot Instance

- Unused EC2 instance that is available for less than the on-demand price
- Hourly price is called *spot price*
  - Adjusted based on long-term supply and demand for spot instances
- Run the instance when capacity is available and price is below threshold
  - When data-center resources are low, spot instances are dropped
  - Mainly suitable for batch workloads

https://aws.amazon.com/ec2/pricing/

AWS EMR

Spot Instance cost strategies

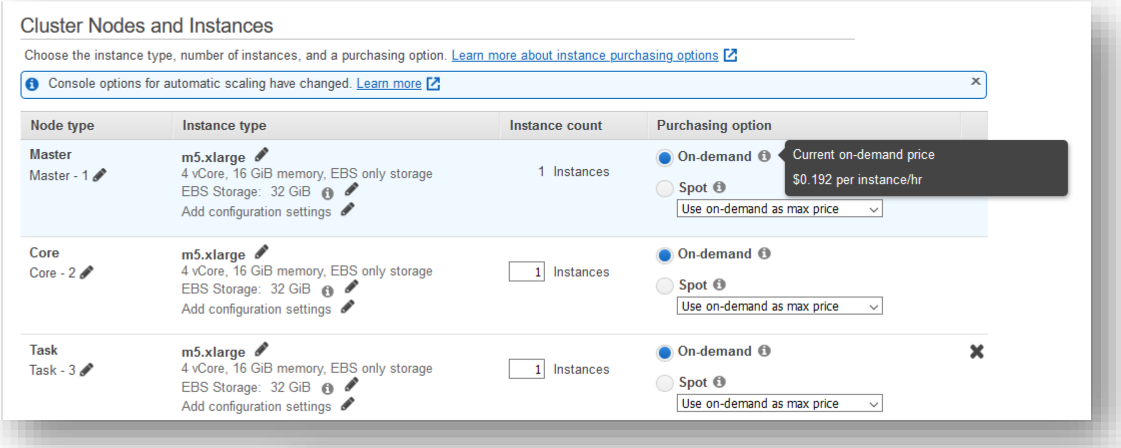
Capacity-optimized strategy

- Allocated instances into the most available pools
- Look at real-time capacity data, predict which are the most available
- Works well for workloads such as big data and analytics
- Works well when we have high cost of interruption

Lowest-price strategy

- Allocates instances in pools with lowest price at time of fulfillment

Creating the cluster



Cluster Nodes and Instances

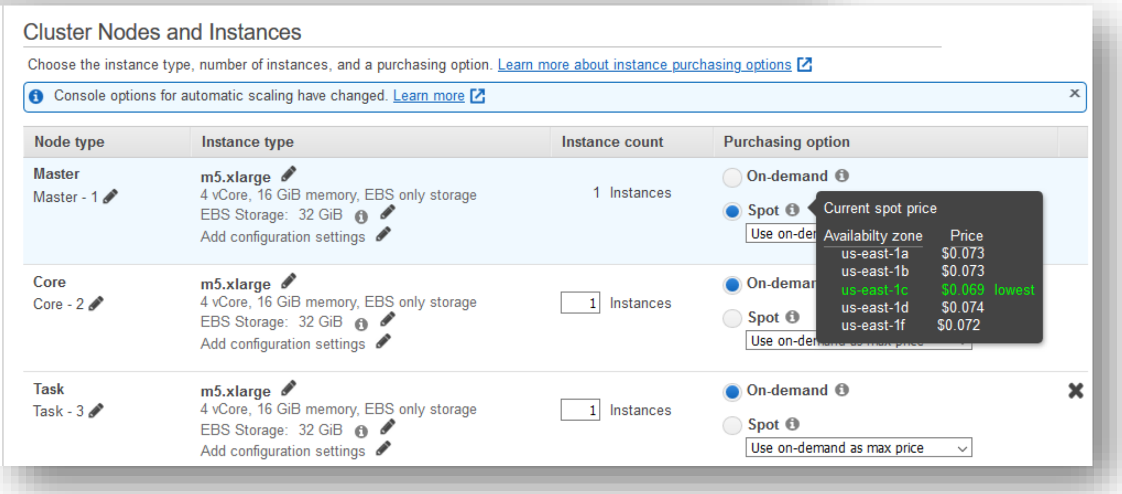
Choose the instance type, number of instances, and a purchasing option. Learn more about instance purchasing options

Console options for automatic scaling have changed. Learn more

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demand ○ Spot Use on-demand as max price
Core Core - 2	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demand ○ Spot Use on-demand as max price
Task Task - 3	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demand ○ Spot Use on-demand as max price

Current on-demand price

$0.192 per instance/hr



Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. Learn more about instance purchasing options

Console options for automatic scaling have changed. Learn more

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	○ On-demand ● Spot Use on-der
Core Core - 2	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demar ○ Spot Use on-de
Task Task - 3	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demand ○ Spot Use on-demand as max price

Current spot price

Availabilty zone	Price	
us-east-1a	$0.073	
us-east-1b	$0.073	
us-east-1c	$0.069	lowest
us-east-1d	$0.074	
us-east-1f	$0.072	

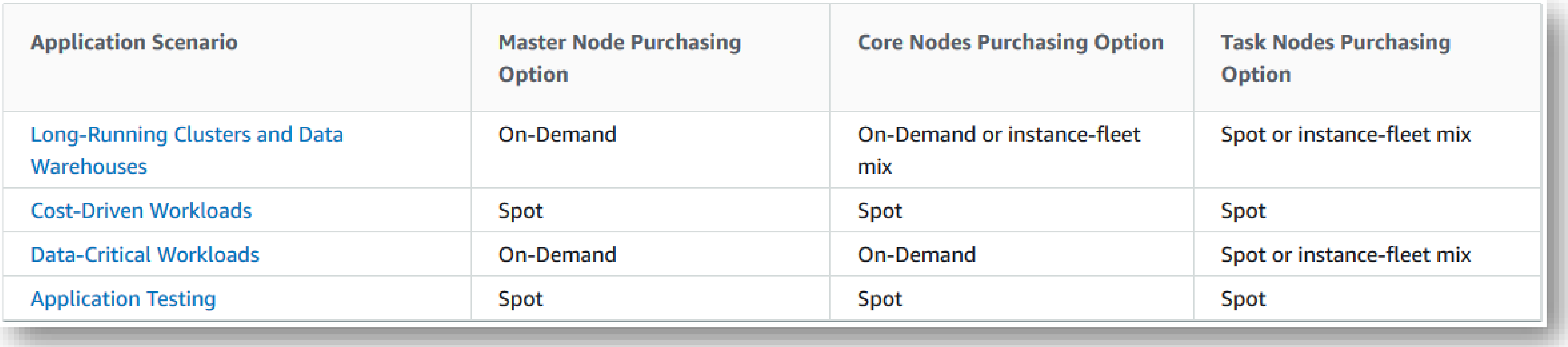
Creating the cluster

Choose to launch **master**, **core**, or **task** on Spot Instances

- The **master** node controls the cluster
  - When terminated, the cluster ends
  - Use *spot instances* if you are running a cluster where sudden termination is acceptable
- **Core** nodes process data and store information using HDFS
  - When terminated, data is lost
  - Use *spot instances* when partial HDFS data loss is tolerable
- **Task** nodes process data but do not hold persistent data in HDFS
  - When terminated, computational capacity is lost
  - The effect of spot instances on the cluster is "minimal"

https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-instances-guidelines.html

Creating the cluster



Application Scenario	Master Node Purchasing Option	Core Nodes Purchasing Option	Task Nodes Purchasing Option
Long-Running Clusters and Data Warehouses	On-Demand	On-Demand or instance-fleet mix	Spot or instance-fleet mix
Cost-Driven Workloads	Spot	Spot	Spot
Data-Critical Workloads	On-Demand	On-Demand	Spot or instance-fleet mix
Application Testing	Spot	Spot	Spot

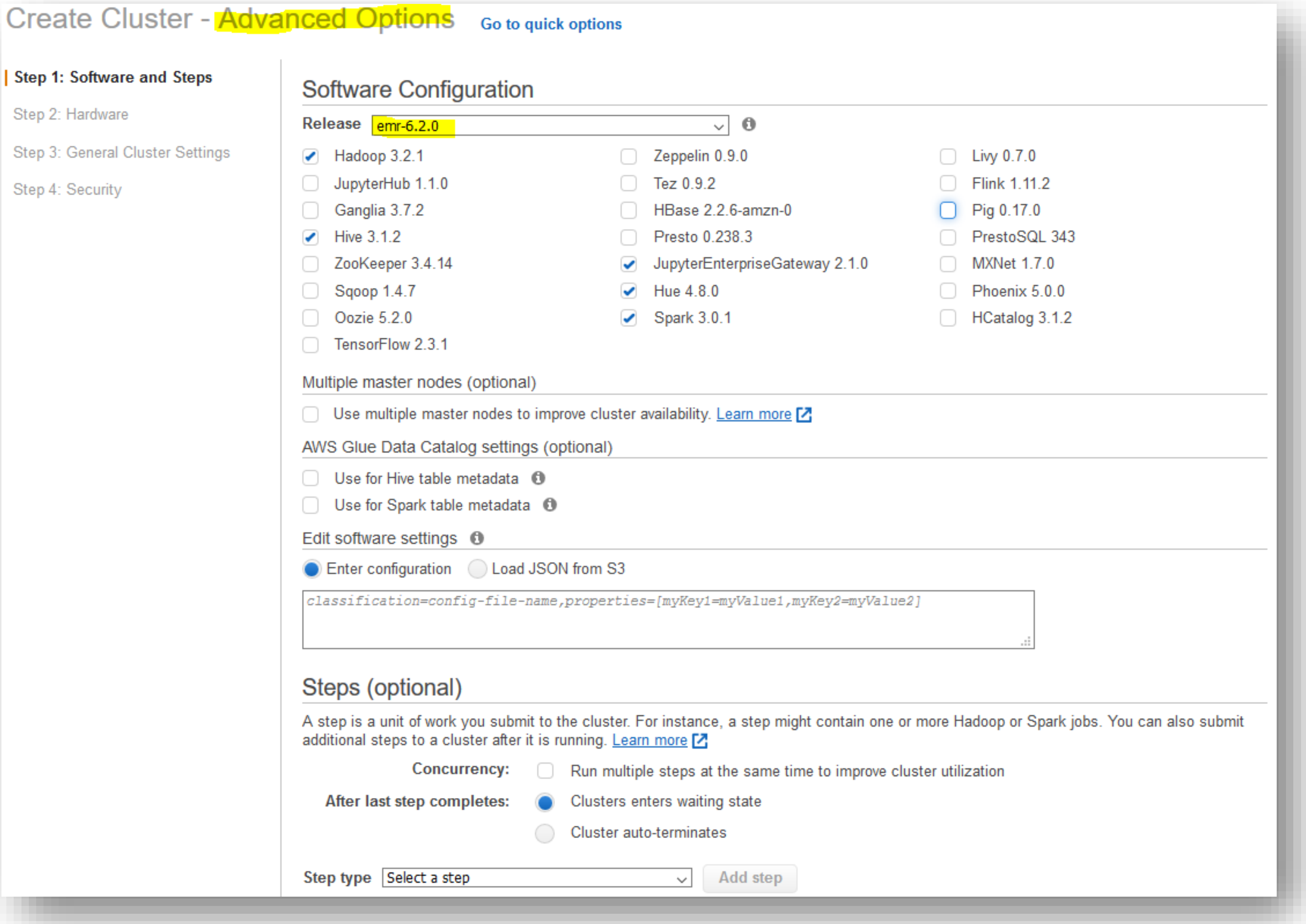
Creating the cluster

Amazon EMR provides two main file systems

- **HDFS** and **EMRFS**, specify which file system to use by the prefix
- `hdfs://path (or just `path`)`
  - HDFS is used by the master and core nodes
  - AWS EBS volume storage is used for HDFS data
  - Is fast, best used for caching the results produced by intermediate job-flow steps, why?
  - It's ephemeral storage which is reclaimed when the cluster ends
- `s3://DOC-EXAMPLE-BUCKET1/path` (EMRFS)
  - An implementation of the Hadoop file system atop Amazon S3
  - We can avoid EBS storage

https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-storage.html

Creating the cluster



Create Cluster - Advanced Options Go to quick options

Software Configuration

Release: emr-6.2.0

- ☑ Hadoop 3.2.1
- ☐ JupyterHub 1.1.0
- ☐ Ganglia 3.7.2
- ☑ Hive 3.1.2
- ☐ ZooKeeper 3.4.14
- ☐ Sqoop 1.4.7
- ☐ Oozie 5.2.0
- ☐ TensorFlow 2.3.1
- ☐ Zeppelin 0.9.0
- ☐ Tez 0.9.2
- ☐ HBase 2.2.6-amzn-0
- ☐ Presto 0.238.3
- ☑ JupyterEnterpriseGateway 2.1.0
- ☑ Hue 4.8.0
- ☑ Spark 3.0.1
- ☐ Livy 0.7.0
- ☐ Flink 1.11.2
- ☐ Pig 0.17.0
- ☐ PrestoSQL 343
- ☐ MXNet 1.7.0
- ☐ Phoenix 5.0.0
- ☐ HCatalog 3.1.2

Multiple master nodes (optional)

☐ Use multiple master nodes to improve cluster availability. Learn more

AWS Glue Data Catalog settings (optional)

☐ Use for Hive table metadata

☐ Use for Spark table metadata

Edit software settings

● Enter configuration ○ Load JSON from S3

```
classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]
```

Steps (optional)

A step is a unit of work you submit to the cluster. For instance, a step might contain one or more Hadoop or Spark jobs. You can also submit additional steps to a cluster after it is running. Learn more

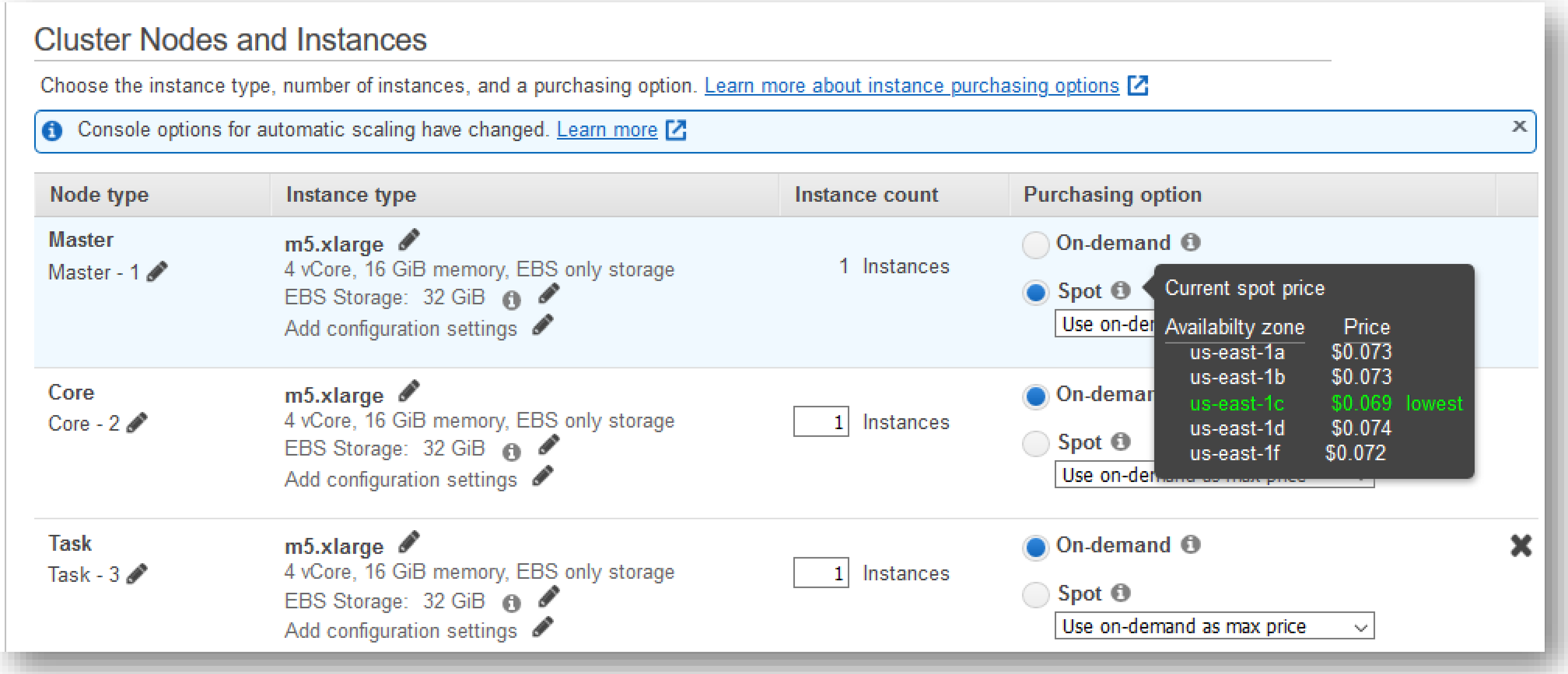
Concurrency: ☐ Run multiple steps at the same time to improve cluster utilization

After last step completes: ● Clusters enters waiting state

○ Cluster auto-terminates

Step type Select a step

Creating the cluster



Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. Learn more about instance purchasing options

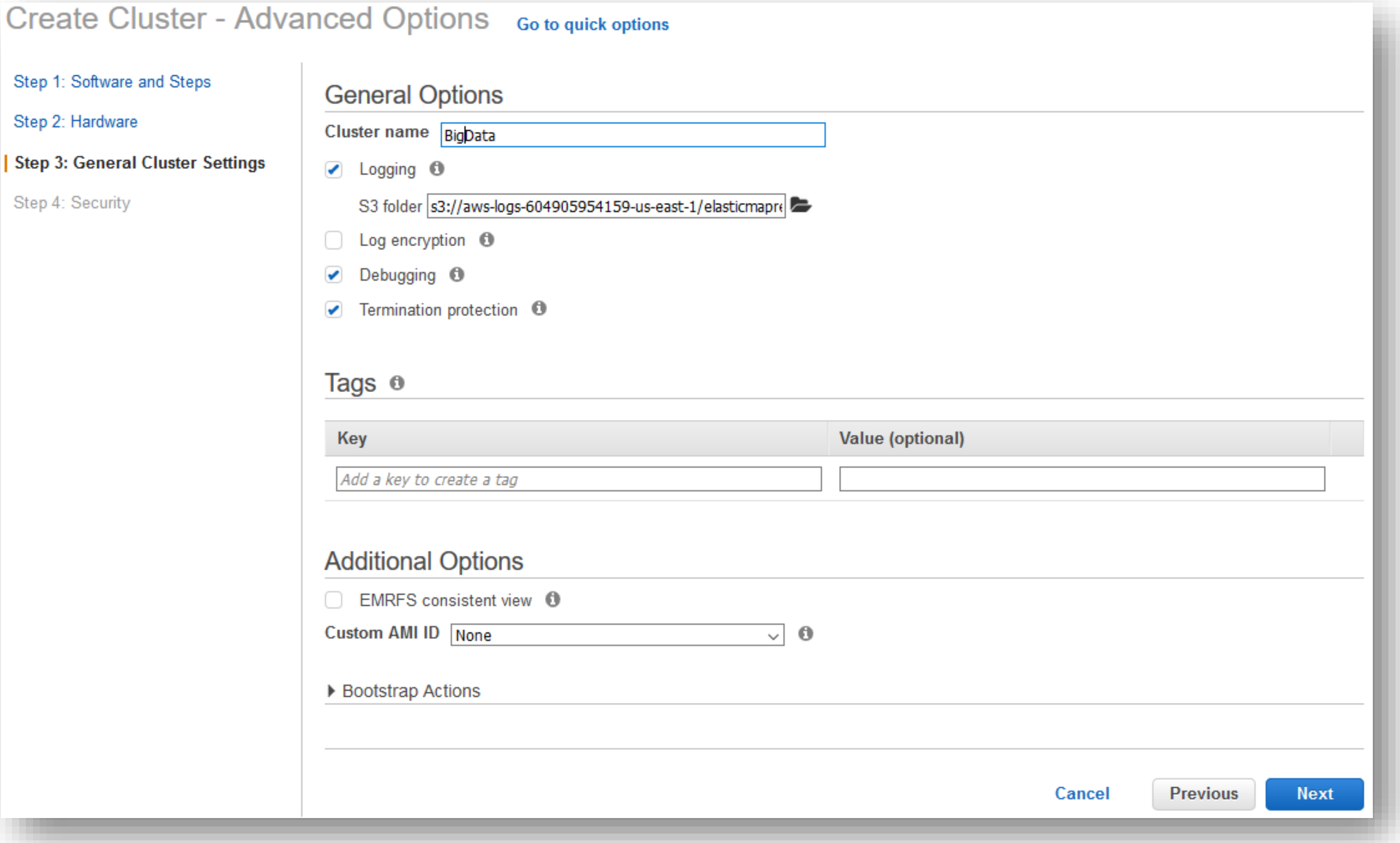
Console options for automatic scaling have changed. Learn more

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	○ On-demand ● Spot Use on-der
Core Core - 2	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demar ○ Spot Use on-de
Task Task - 3	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	● On-demand ○ Spot Use on-demand as max price

Current spot price

Availabilty zone	Price	
us-east-1a	$0.073	
us-east-1b	$0.073	
us-east-1c	$0.069	lowest
us-east-1d	$0.074	
us-east-1f	$0.072	

Creating the cluster



Create Cluster - Advanced Options

General Options

Cluster name: BigData

☑ Logging

S3 folder: s3://aws-logs-604905954159-us-east-1/elasticmapre

□ Log encryption

☑ Debugging

☑ Termination protection

Tags

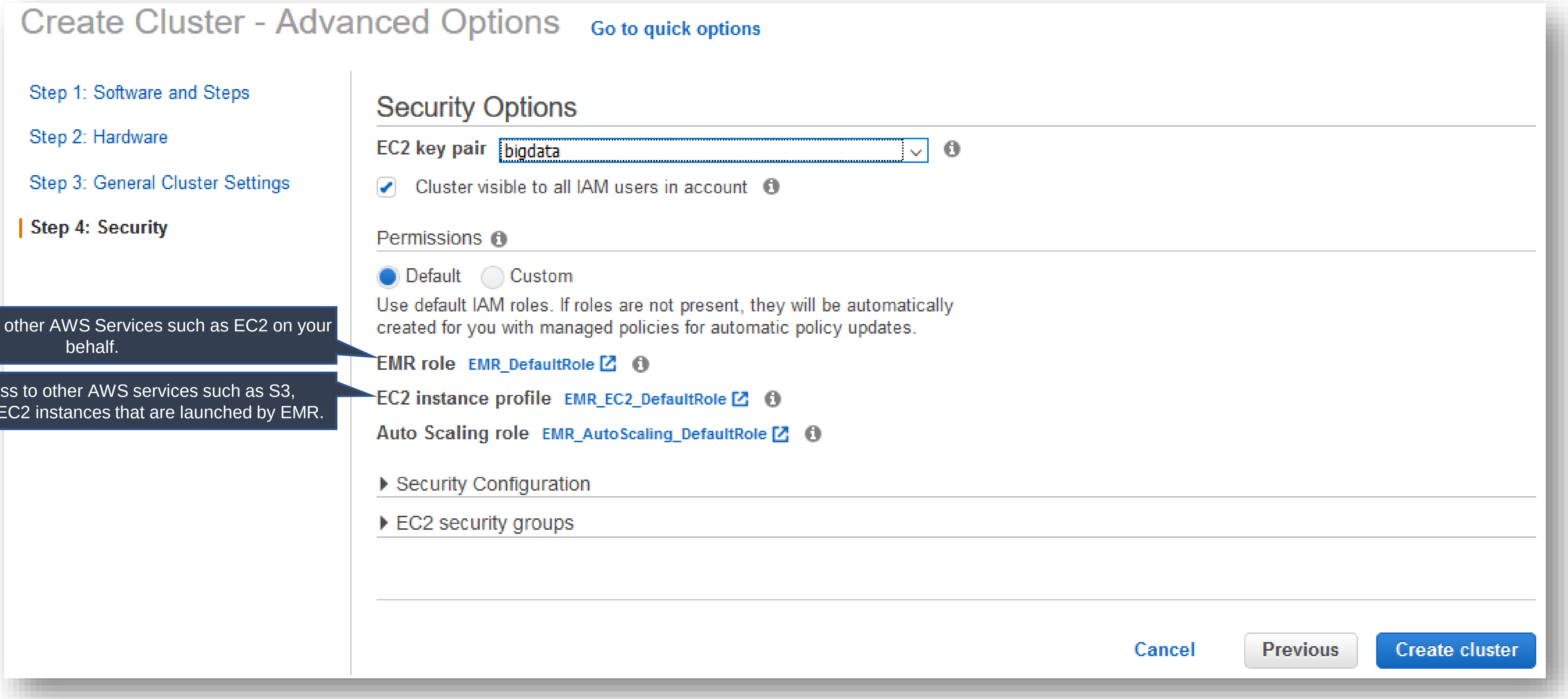
Key	Value (optional)
Add a key to create a tag	

Additional Options

□ EMRFS consistent view

Custom AMI ID: None

Creating the cluster



Create Cluster - Advanced Options

Security Options

EC2 key pair: bigdata

☑ Cluster visible to all IAM users in account

Permissions

● Default ○ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role: EMR_DefaultRole

EC2 instance profile: EMR_EC2_DefaultRole

Auto Scaling role: EMR_AutoScaling_DefaultRole

Allows EMR to call other AWS Services such as EC2 on your behalf.

Provides access to other AWS services such as S3, DynamoDB from EC2 instances that are launched by EMR.

Creating the cluster

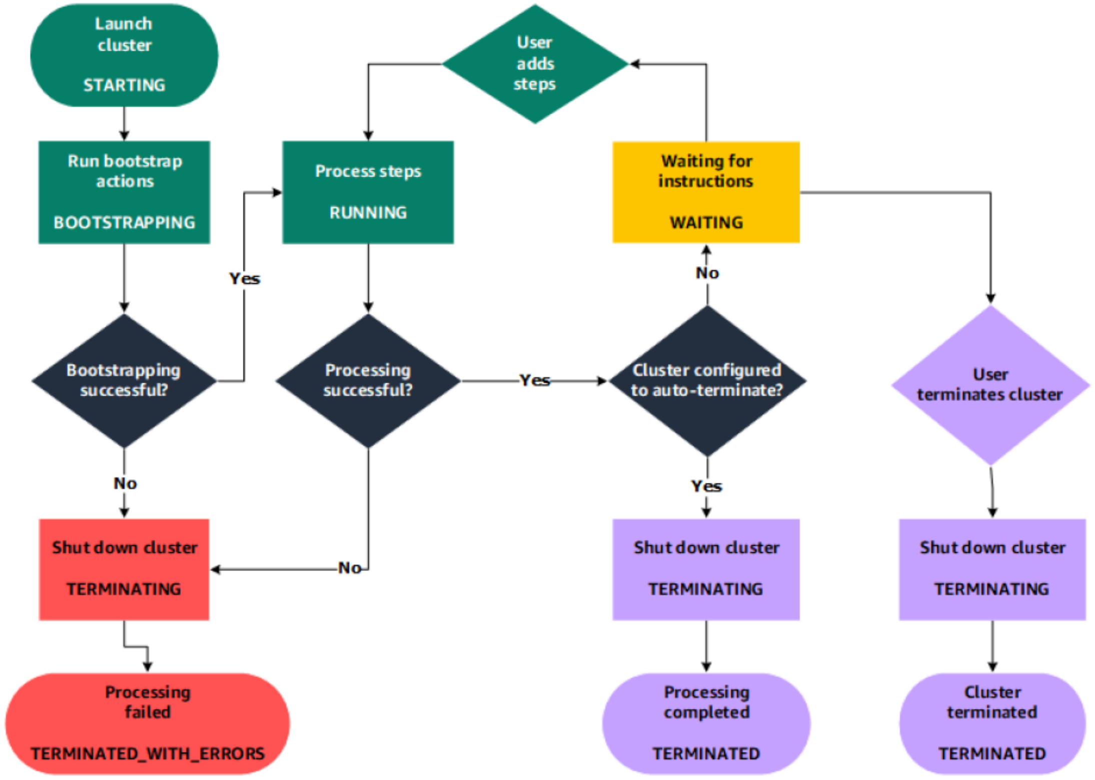
Using CLI (command line interface)

```
aws emr create-cluster --auto-scaling-role EMR_AutoScaling_DefaultRole --termination-protected --
applications Name=Hadoop Name=Hive Name=Hue Name=JupyterEnterpriseGateway Name=Spark --ebs-root-volume-
size 10 --ec2-attributes
'{"KeyName":"bigdata","InstanceProfile":"EMR_EC2_DefaultRole","SubnetId":"subnet-
5fa2f912","EmrManagedSlaveSecurityGroup":"sg-07818b5690a50b3f1","EmrManagedMasterSecurityGroup":"sg-
0e2f5550a2cb98f79"}' --service-role EMR_DefaultRole --enable-debugging --release-label emr-6.2.0 --log-
uri 's3n://aws-logs-604905954159-us-east-1/elasticmapreduce/' --name 'BigData' --instance-groups
'[{"InstanceCount":1,"BidPrice":"OnDemandPrice","EbsConfiguration":{"EbsBlockDeviceConfigs":[{"VolumeSpe
cification":{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"InstanceGroupType":"MASTER","I
nstanceType":"m4.xlarge","Name":"Master -
1"},{"InstanceCount":1,"BidPrice":"OnDemandPrice","EbsConfiguration":{"EbsBlockDeviceConfigs":[{"VolumeS
pecification":{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"InstanceGroupType":"CORE","I
nstanceType":"m4.xlarge","Name":"Core - 2"}]' --scale-down-behavior TERMINATE_AT_TASK_COMPLETION --
region us-east-1
```

Cluster lifecycle

Creating a cluster (it takes ~10 minutes)

- A cluster cannot be stopped
- It can only be terminated



Cluster lifecycle

`STARTING`: EMR provisions EC2 instances for each required instance

`BOOTSTRAPPING`: EMR runs actions that you specify on each instance

- E.g., install custom applications and perform customizations

Amazon EMR installs the native applications

- E.g., Hive, Hadoop, Spark, and so on

`RUNNING`: a step for the cluster is currently being run

- Cluster sequentially runs any steps that you specified when you created the cluster

`WAITING`: after steps run successfully

`TERMINATING`: after manual shut down

- Any data stored on the cluster is deleted

Cluster lifecycle

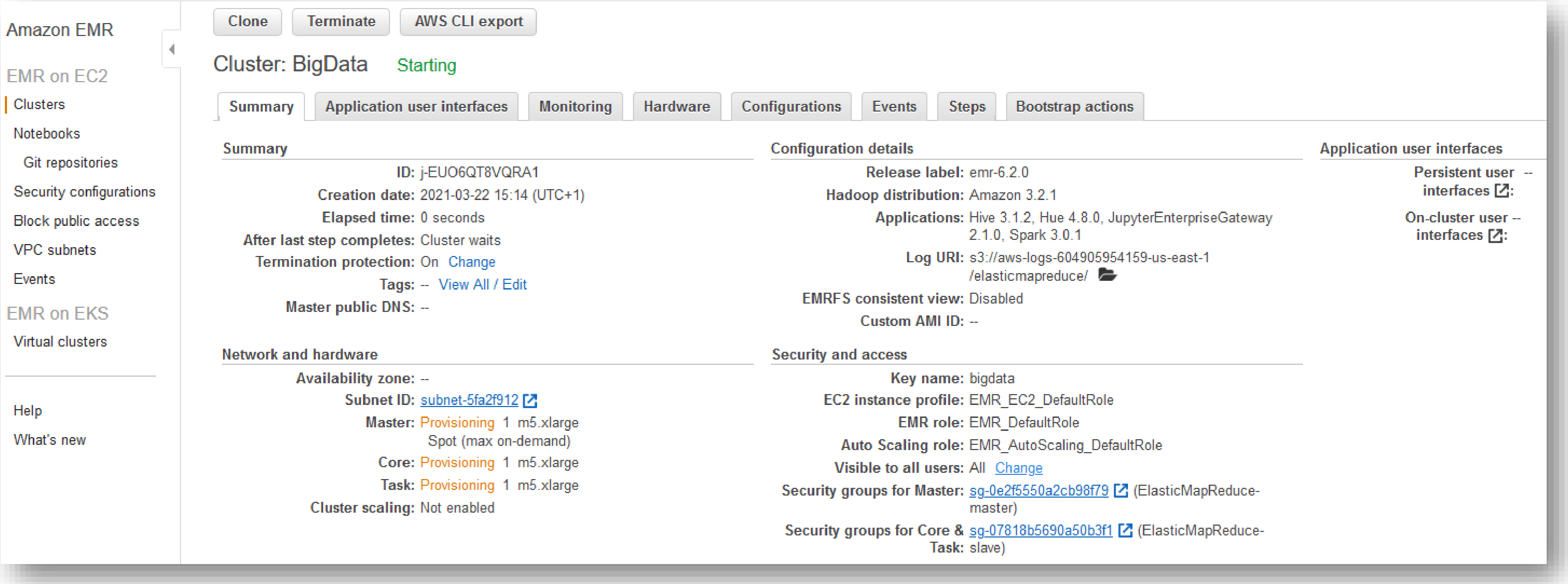
A **step** is a user-defined unit of processing

- E.g., one algorithm that manipulates the data

Step states

- `PENDING:` The step is waiting to be run
- `RUNNING:` The step is currently running
- `COMPLETED:` The step completed successfully
- `CANCELLED:` The step was cancelled before running because an earlier step failed
- `FAILED:` The step failed while running

Running the cluster



Cluster: BigData Starting

Summary

ID: j-EUO6QT8VQRA1
Creation date: 2021-03-22 15:14 (UTC+1)
Elapsed time: 0 seconds
After last step completes: Cluster waits
Termination protection: On Change
Tags: -- View All / Edit
Master public DNS: --

Network and hardware

Availability zone: --
Subnet ID: subnet-5fa2f912
Master: Provisioning 1 m5.xlarge Spot (max on-demand)
Core: Provisioning 1 m5.xlarge
Task: Provisioning 1 m5.xlarge
Cluster scaling: Not enabled

Configuration details

Release label: emr-6.2.0
Hadoop distribution: Amazon 3.2.1
Applications: Hive 3.1.2, Hue 4.8.0, JupyterEnterpriseGateway 2.1.0, Spark 3.0.1
Log URI: s3://aws-logs-604905954159-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

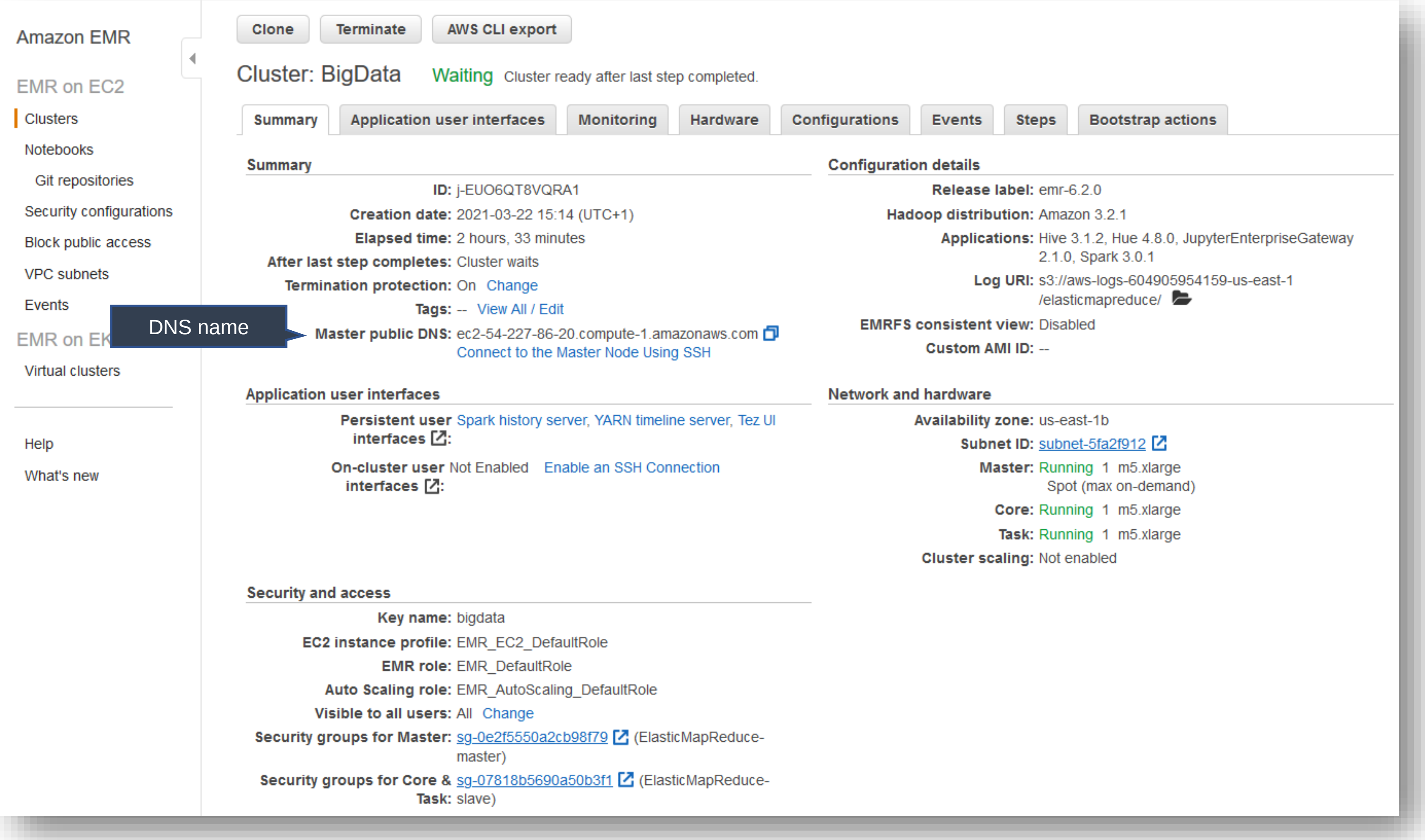
Security and access

Key name: bigdata
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0e2f5550a2cb98f79 (ElasticMapReduce-master)
Security groups for Core & Task: sg-07818b5690a50b3f1 (ElasticMapReduce-slave)

Application user interfaces

Persistent user interfaces: --
On-cluster user interfaces: --

Running the cluster



DNS name

Cluster: BigData Waiting Cluster ready after last step completed.

Summary

ID: j-EUO6QT8VQRA1
Creation date: 2021-03-22 15:14 (UTC+1)
Elapsed time: 2 hours, 33 minutes
After last step completes: Cluster waits
Termination protection: On Change
Tags: -- View All / Edit
Master public DNS: ec2-54-227-86-20.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Application user interfaces

Persistent user interfaces: Spark history server, YARN timeline server, Tez UI
On-cluster user interfaces: Not Enabled Enable an SSH Connection

Security and access

Key name: bigdata
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0e2f5550a2cb98f79 (ElasticMapReduce-master)
Security groups for Core & Task: sg-07818b5690a50b3f1 (ElasticMapReduce-slave)

Configuration details

Release label: emr-6.2.0
Hadoop distribution: Amazon 3.2.1
Applications: Hive 3.1.2, Hue 4.8.0, JupyterEnterpriseGateway 2.1.0, Spark 3.0.1
Log URI: s3://aws-logs-604905954159-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Network and hardware

Availability zone: us-east-1b
Subnet ID: subnet-5fa2f912
Master: Running 1 m5.xlarge
Spot (max on-demand)
Core: Running 1 m5.xlarge
Task: Running 1 m5.xlarge
Cluster scaling: Not enabled