

## Launch an EMR cluster- Steps 1- 4

### Create service linked role for Spot using Cloud9

aws iam create-service-linked-role --aws-service-name [spot.amazonaws.com](https://spot.amazonaws.com)

### Upload spark script and dataset to S3 bucket in AWS Console

- Upload [Spark Script](#) to S3 application bucket : spark-app-<account-id>
- Create a new S3 folder: spark-app-<account-id>/input
- Upload [Dataset](#) to Input folder

### Cluster Configuration

- Select **Instance Fleets**
- For **Primary**, click on **Add instance type** and add additional instances: m4.xlarge, m5a.xlarge, m5d.xlarge
- For **Core**, click on **Add instance type** and add additional instances recommended in previous section using ec2-instance-select tool. **Note:** change the default instance type already populated (m5.xlarge, which only has 4vCPU and 16GB memory)

## Core

Choose one or more EC2 instance type

Choose EC2 instance type

<b>r5.xlarge</b> 4 vCore 32 GiB memory EBS only storage On-Demand price: \$0.252 per instance/hour Lowest Spot price: \$0.075 (us-east-1f)	Actions ▼	Remove
<b>r4.xlarge</b> 4 vCore 30.5 GiB memory EBS only storage On-Demand price: \$0.266 per instance/hour Lowest Spot price: \$0.108 (us-east-1d)	Actions ▼	Remove
<b>r5.2xlarge</b> 8 vCore 64 GiB memory EBS only storage On-Demand price: \$0.504 per instance/hour Lowest Spot price: \$0.156 (us-east-1f)	Actions ▼	Remove
<b>r4.2xlarge</b> 8 vCore 61 GiB memory EBS only storage On-Demand price: \$0.532 per instance/hour Lowest Spot price: \$0.244 (us-east-1a)	Actions ▼	Remove
<b>r4.4xlarge</b> 16 vCore 122 GiB memory EBS only storage On-Demand price: \$1.064 per instance/hour Lowest Spot price: \$0.470 (us-east-1d)	Actions ▼	Remove

- For **Task**, click on **Add Instance Type**. Add up to 15 instance types based on instances recommended in previous section. **Note**: change the default instance type already populated (m5.xlarge, which only has 4vCPU and 16GB memory)

Cluster scaling and provisioning option as shown in Figure below.

### Scaling configuration

- Select Use **EMR-managed scaling**
- Set **minimum cluster size** to 4 units
- Set **maximum cluster size** to 68 units
- Set **maximum core nodes** to 4 units
- Set **maximum On-demand instances** in the cluster to 4 units
- Click on the checkbox **Apply allocation strategy**

## Cluster scaling and provisioning option [Info](#)

Set up scaling and provisioning configurations for the core and task node groups for your cluster.

### Choose an option



#### Set cluster size manually

Use this option if you know your workload patterns in advance.



#### Use EMR-managed scaling

Monitor key workload metrics so that EMR can optimize the cluster size and resource utilization.



#### Use custom automatic scaling

To programmatically scale core and task nodes, create custom automatic scaling policies.

### Scaling configuration

#### Minimum cluster size

unit(s)

#### Maximum cluster size

unit(s)

#### Maximum core nodes in the cluster

Limit the number of core nodes in your cluster.

unit(s)

#### Maximum On-Demand instances in the cluster

To provision the primary node to use On-Demand pricing and other nodes in the cluster to use Spot pricing, set this value to 1. To provision the entire cluster to use On-Demand pricing, use the same value as your maximum cluster size.

unit(s)

### Provisioning configuration

- Set Core On-Demand size to 4 units. Leave Core spot size at 0 units
- Set Task Spot size to 32 units.
- Leave timeout configuration settings default.

### Provisioning configuration

Set the size of your core and task instance fleets. Amazon EMR attempts to provision this capacity when you launch your cluster.

#### Core

##### Spot size

unit(s)

##### On-Demand size

unit(s)

#### Task

##### Spot size

unit(s)

##### On-Demand size

unit(s)

### Allocation Strategy

- Spot Strategy: Choose Capacity optimized to minimize risk of interruption

## Allocation strategy [Info](#)

### ☒ Apply allocation strategy (recommended)

The allocation strategy determines which of your available pools to request Spot Instances from. Amazon EMR always provisions On-Demand capacity with the lowest-price strategy.

### On-Demand strategy

Lowest price

### Spot strategy

☐ **Price-capacity optimized (recommended)**  
Request the lowest priced Spot Instances from your most available pools. This is the best strategy to balance instance price and the risk of interruption.

☒ **Capacity optimized**  
Request Spot Instances from your most available pools. This strategy has the lowest risk of interruption.

☐ **Lowest price**  
Request Spot Instances from the lowest priced pools based on your instance type requirements. This strategy has the highest risk of interruption.

☐ **Diversified across all pools**  
Request Spot Instances evenly across all your available pools.

## Networking

**Networking** [Info](#)

**Virtual private cloud (VPC)** [Info](#)

[Browse](#)

[Create VPC](#)

**Subnets** [Info](#)

For high availability, choose at least 2 subnets in different Availability Zones. You can't choose both public and private subnets.

[Refresh](#)

[Create subnet](#)

subnet-00f64ed43bcefd8f5 | - | us-east-1a

subnet-010ea179faa564347 | - | us-east-1e

subnet-0255b3c968b185800 | - | us-east-1f

subnet-02de1f4bb27070b54 | - | us-east-1c

subnet-072de9f1f352ef450 | - | us-east-1d

subnet-0ab0f7726f2b95811 | - | us-east-1b

Minimum: 1 subnet.

[EC2 security groups \(firewall\)](#)

### Steps - Add

- Configure Steps as show in figure below
- **Arguments:** spark-submit --executor-memory 18G --executor-cores 4 s3://spark-app-574488550952/script.py s3://spark-app-574488550952/input s3://spark-app-574488550952/results

Edit step

×

Type

☒ Custom JAR  
Adds a step that enables you to write a custom script to process your data using the Java programming language.

☐ Spark application  
Adds a step that submits work to the Spark framework on the cluster.

☐ Shell script  
Troubleshoot your cluster.

Name

Spark Application

JAR location

The JAR location may be a path into S3 or a fully qualified java class in the classpath.

command-runner.jar

Arguments - optional

These are passed to the main function in the JAR. If the JAR does not specify a main class in its manifest file, you can specify another class name as the first argument. [Learn more](#)

```
spark-submit --deploy-mode client --executor-memory 18G --executor-cores 4 s3://spark-app-574488550952/script.py s3://spark-app-574488550952/results/
```

Action if step fails

The action to take when the step fails.

☒ Continue  
Continues to the next step in the queue.

☐ Cancel and wait  
Cancels any pending steps and returns the cluster to the waiting state.

☐ Terminate cluster  
Shuts down the cluster.

Cancel

Save step

Skip the following sections: **Cluster Termination**, **Bootstrap actions**, **Cluster Logs**

**For Tags - optional**, click on **Add new tag**. For **Key**, enter Name. For **Value**, enter EMRTransientCluster1

Leave **Software settings** default.

#### Security configuration and EC2 key pair

- On the **EC2 key pair** drop-down, select emr-workshop-key-pair

#### Identity and Access Management (IAM) roles

- For **Amazon EMR service role**, select **Create a service role**
- For **Security group**, select default

## Amazon EMR service role [Info](#)

The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.

### ☐ Choose an existing service role

Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.

### ☒ Create a service role

Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.

## Networking resources

We've already added the resources that you configured in the [Networking](#) section. Choose the VPC, subnet, and security groups that the service role can access.

### Virtual Private Cloud (VPC)

Choose one or more VPCs

-  
vpc-0290aa3308980ad71

### Subnet

Choose one or more subnets

-  
subnet-0255b3c968b185800

-  
subnet-02de1f4bb27070b54

-  
subnet-00f64ed43bcefd8f5

-  
subnet-010ea179faa564347

[+ Show more chosen options \(+2\)](#)

### Security group

Choose one or more security groups

default  
sg-07a63d757dcd5a674

## EC2 instance profile for Amazon EMR

- Select **Create an instance profile**
- For **S3 bucket access**, select **All S3 buckets in this account with read and write access**