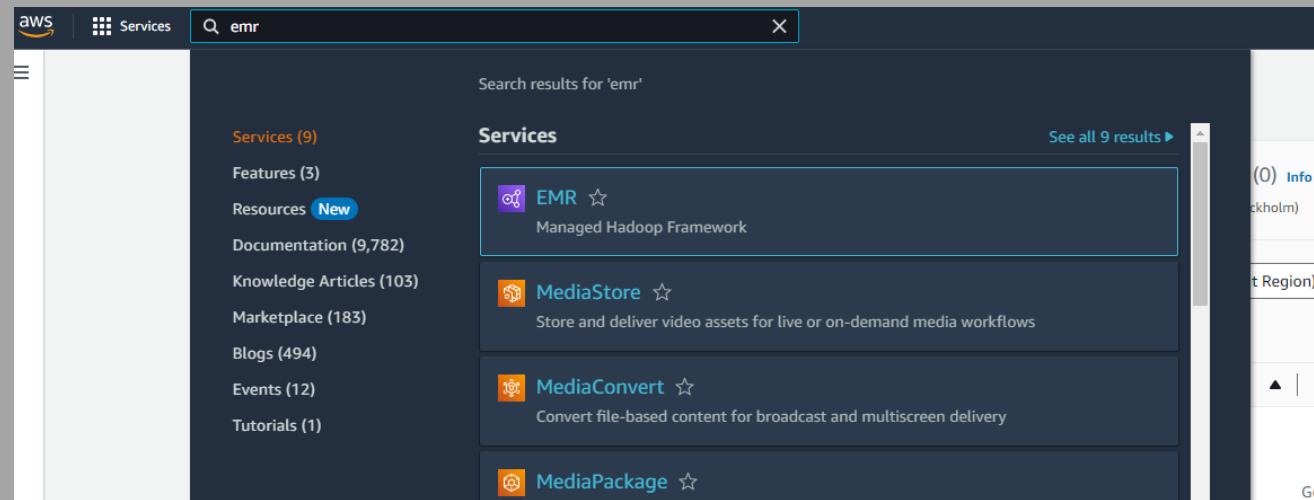


AWS ElasticMapReduce Data Processing

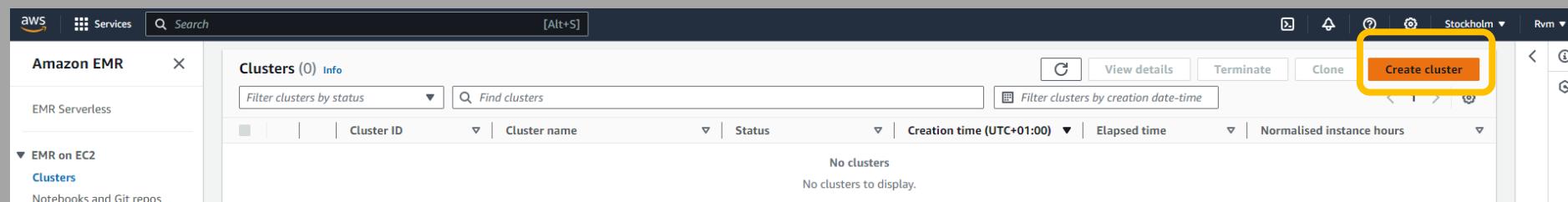


Create an EMR cluster

1. Find and Access to EMR service



2. Create the EMR Cluster



Guide workflow



January 2024

2 amazon EMR

- 2.1. Name and applications
- 2.2. Cluster configuration
- 2.3. Task nodes and EBS root volume
- 2.4. Cluster scaling and provisioning
- 2.5. Networking
 - 2.5.a VPC
 - 2.5.b FIREWALL
- 2.6. Steps - Optional
- 2.7. Cluster termination
- 2.8. Bootstrap actions
- 2.9. Cluster logs
- 2.10. Tags -optional
- 2.11. Edit software settings
- 2.12. Security config & EC2 key pair
 - 2.12.a. Create a Key pair
- 2.12. EC2 key pair[Continued]
- 2.13. IAM roles
 - 2.13.a. Service Role
 - 2.13.b. Instance Role S3
- 2.14. Created EMR Cluster

3 AWS Cloud9

- 3. Aws Cloud9 IDE
 - 3. 1. Create Environment
 - 3. 2. New EC2 Instance & Network settings
 - 3. 3. Create the EC2 instance
 - 3. 4. Inside Cloud9
 - 3.4.a Security groups | IP
 - 3.4.a Get SSH command
 - 3.5. ssh EMR cluster connection
 - 3.6. Submitting Spark job command
 - 3.6.1 Submitting Spark job command
 - 3.6.2. Check Spark job output

4 Spark & Hadoop

- 4. Spark & Hadoop workout results
 - 4.a.Apache Spark History Server web UI
 - 4.b. Hadoop's Resource Manager UI

This is an “interactive” guide workflow.
By clicking each step you will be redirected there. You can come back and forth here clicking on the aws logo on the upper right corner in every slide

Create an EMR cluster

2.1. Name and applications

The screenshot shows the 'Create cluster' wizard in the AWS Management Console. The current step is 'Name and applications'. The 'Name' field contains 'EMR RafaelVera'. The 'Amazon EMR release' dropdown is set to 'emr-7.0.0'. Under 'Application bundle', there are several pre-defined bundles: Spark Interactive, Core Hadoop, Flink, HBase, Presto, Trino, and a 'Custom' option. Below these are individual application checkboxes. The checked applications are: Hadoop 3.3.6, JupyterEnterpriseGateway 2.6.0, Hive 3.1.3, JupyterHub 1.5.0, Pig 0.17.0, Sqoop 1.4.7, and Trino 426. Other available applications include AmazonCloudWatchAgent 1.300031.1, HCatalog 3.1.3, Hue 4.11.0, Livy 0.7.1, Phoenix 5.1.3, Spark 3.5.0, Tez 0.10.2, and ZooKeeper 3.5.10. At the bottom, under 'AWS Glue Data Catalogue settings', there are two checked options: 'Use for Hive table metadata' and 'Use for Spark table metadata'.

EMR Release:

emr-7.0.0 - A managed cluster platform that simplifies running big data frameworks.

Custom Applications:

Enables a personalized selection of applications and versions for project-specific needs.

AWS Glue Data Catalogue:

Utilized for storing Hive and Spark table metadata, serving as a central metadata repository.

Application Bundle:

Custom package including Hadoop 3.3.6, Hive 3.1.3, Hue 4.11.0, and Jupyter interfaces.

Create an EMR cluster

2.2. Cluster configuration

The screenshot shows the 'Cluster configuration' section of the AWS EMR console. It starts with a choice between 'Instance groups' (selected) and 'Instance fleets'. Under 'Instance groups', there's a 'Primary' section where 'm5.xlarge' is chosen as the EC2 instance type. This section includes options for 'Use high availability' and a link to find out more. Below this is a 'Core' section with the same 'm5.xlarge' configuration. At the bottom, there's a note about optional node configuration.

This section allows to choose the configuration method for the primary, core, and task node groups of the cluster.

Instance Groups:

Where one selects a single EC2 instance type for the primary and core node groups.

Primary:

m5.xlarge. Use high availability, which means launching a more resilient cluster with three primary nodes on On-Demand Instances.

Core:

Same config as Primary group.

Create an EMR cluster

2.3. Task nodes and EBS root volume

Task 1 of 1

Name

Task - 1

Remove instance group

Choose EC2 instance type

m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-demand price: USD 0.204 per instance/hour
Lowest spot price: \$0.065 (eu-north-1c)

Actions ▾

▶ Node configuration - optional

Add task instance group

You can add up to 47 more task instance groups.

EBS root volume

EBS root volume applies to the operating systems and applications that you install on the cluster. [EBS root volume ratio constraints](#)

Size (GiB)	IOPS	Throughput (MiB/s)
15	3000	125

15 - 100 GiB per volume
General purpose SSD (gp3)
3000 - 16000 IOPS per volume.
Choose a maximum ratio of 500:1 between IOPS and volume size.
125 - 1000 MiB/s per volume.
Choose a maximum ratio of 0.25:1 between throughput and IOPS.

Task nodes:

are utilized to boost computational power for parallel data processing without holding persistent data.

They offer scalability to handle fluctuating workloads, cost efficiency via cheaper spot instances, flexibility in instance selection, and have a non-disruptive nature as their removal doesn't affect the cluster's stored data, making them suitable for temporary spot usage.

Elastic Block Store (EBS) root volume:
act as the primary storage for the operating system and applications on the cluster nodes.

Size: indicating the storage capacity of the root volume.

IOPS: Input/Output Operations Per Second, a measure of the volume's performance. How many read/write operations the volume can handle.

Throughput: the rate at which data can be read from or written to the storage volume.

Create an EMR cluster

2.4. Cluster scaling and provisioning

Cluster scaling and provisioning [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 optimise the cluster size and optimise its resource utilisation.
- Use custom automatic scaling
To programmatically scale core and task nodes, create custom automatic scaling policies.

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

Name	Instance type	Instance(s) size	Use spot purchasing option
Core	m5.xlarge	<input type="text" value="1"/>	<input type="checkbox"/>
Task - 1	m5.xlarge	<input type="text" value="1"/>	<input type="checkbox"/>

Scaling option:

Set cluster size manually:

when you know your workload patterns in advance and want to specify the cluster size yourself.

Provisioning configuration:

allows you to set the size of your core and task instance groups, which Amazon EMR will attempt to provision when launching your cluster.

Instance Type:

Both the 'Core' and 'Task - 1' groups are set to use m5.xlarge instances.

Instance(s) Size: Indicates the number of instances for each group.

Use spot purchasing option:

is not checked in the image, allows the option to purchase spot instances for cost savings.

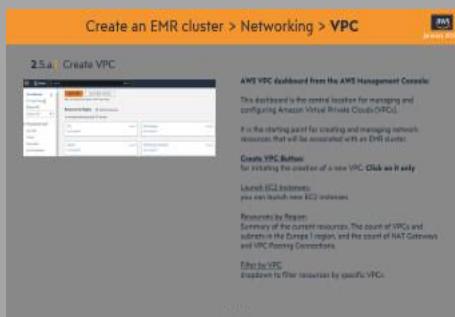
Create an EMR cluster

2.5. Networking

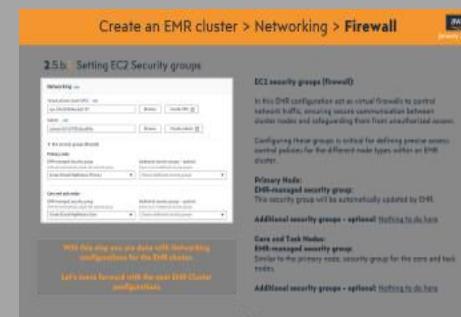
The screenshot shows the AWS Management Console Networking page. It displays a list of resources:

- VPC:** A VPC with ID `vpc-0f16318a4593502a1`. There is a "Create VPC" button and a "Create VPC(Opens in a new tab)" link.
- Subnet:** A subnet with ID `subnet-02079acbd9625e283`. There is a "Create subnet" button and a "Create subnet(Opens in a new tab)" link.
- EC2 security groups (firewall):** A section indicated by a right-pointing arrow.

1st. Create VPC (a)



2nd. Create firewall (b)



Virtual private cloud (VPC)

is a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

VPC ID is a unique identifier for a specific VPC.

Subnet:

is a range of IP addresses in the VPC.

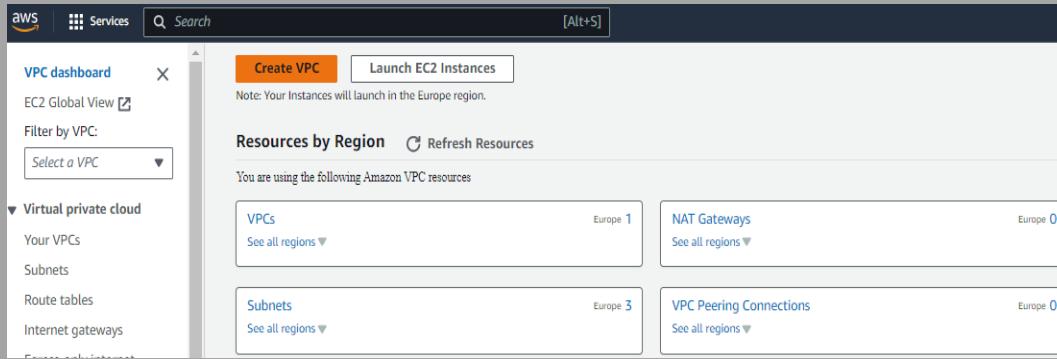
Setting the VPC and subnet for an EMR cluster is crucial as it determines the virtual network isolation and dictates the IP range and availability zone, ensuring secure and optimized network traffic for the cluster's computational resources within the AWS ecosystem.

EC2 Security Groups (firewall):

configure security groups for the service.

Security groups in AWS act as a virtual firewall for instances to control inbound and outbound traffic.

2.5.a.1 Create VPC



AWS VPC dashboard from the AWS Management Console:

This dashboard is the central location for managing and configuring Amazon Virtual Private Clouds (VPCs).

It is the starting point for creating and managing network resources that will be associated with an EMR cluster.

Create VPC Button:

for initiating the creation of a new VPC. **Click on it only**

Launch EC2 Instances:

you can launch new EC2 instances.

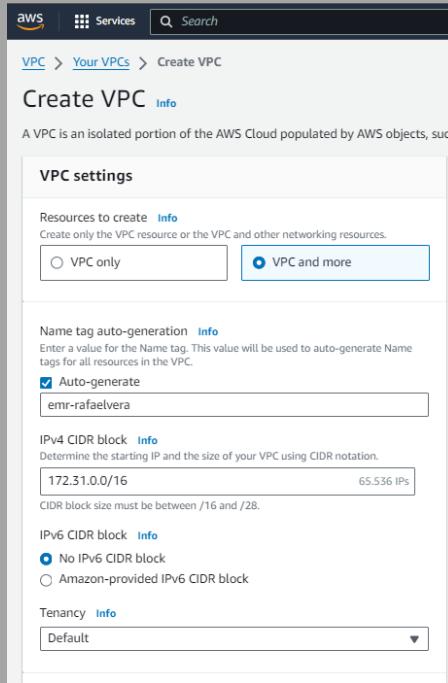
Resources by Region:

Summary of the current resources. The count of VPCs and subnets in the Europe 1 region, and the count of NAT Gateways and VPC Peering Connections.

Filter by VPC:

dropdown to filter resources by specific VPCs.

2.5.a.2 New VPC Settings



VPC settings

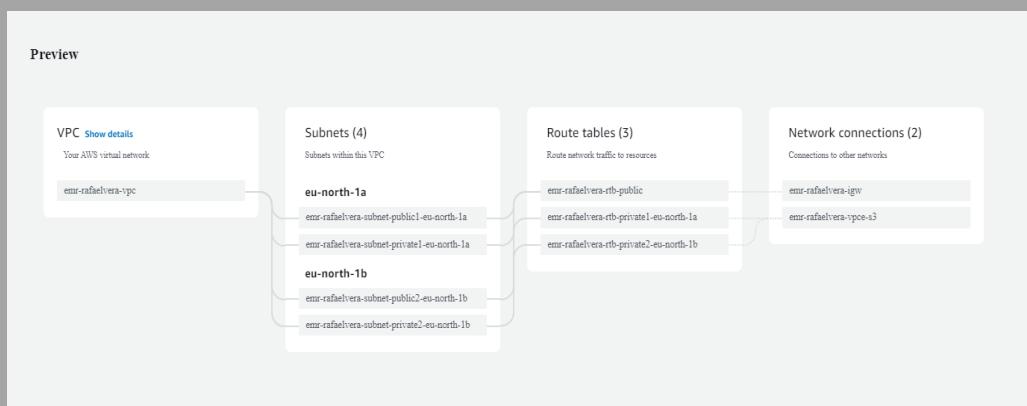
Resources to create: VPC only VPC and more

Name tag auto-generation: Auto-generate
Name tag: emr-rafaelvera

IPv4 CIDR block: 172.31.0.0/16 65,536 IPs

IPv6 CIDR block: No IPv6 CIDR block Amazon-provided IPv6 CIDR block

Tenancy: Default



Name tag:

Enter a name tag for the new VPC. It helps in identifying the VPC within the AWS environment.

IPv4 CIDR block:

Defines the IP address range of the VPC. **Fill it with the IP on the image.**

IPv6 CIDR block:

choose an Amazon-provided IPv6 CIDR block or not assign one.

Pv4 and IPv6 CIDR blocks specify the IP address ranges for networks, with IPv4 being mandatory for current networks and IPv6 addressing the growing need for more IP addresses.

Tenancy:

Choose tenancy of instances launched in the VPC.
"Default" tenancy means instances run on shared hardware.

Preview Pane:

displays the impending creation of a VPC, subnets split between public and private across availability zones, route tables for network routing, and network connections for potential internet or VPC peering. **(this will change in the next step)**

2.5.a.3 New VPC Settings [Continued]

Number of Availability Zones (AZs) Info
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.
1 **2** **3**

Number of public subnets Info
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.
0 **1**

Number of private subnets Info
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.
0 **1** **2**

NAT gateways (\$)
Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.
None **In 1 AZ** **1 per AZ**

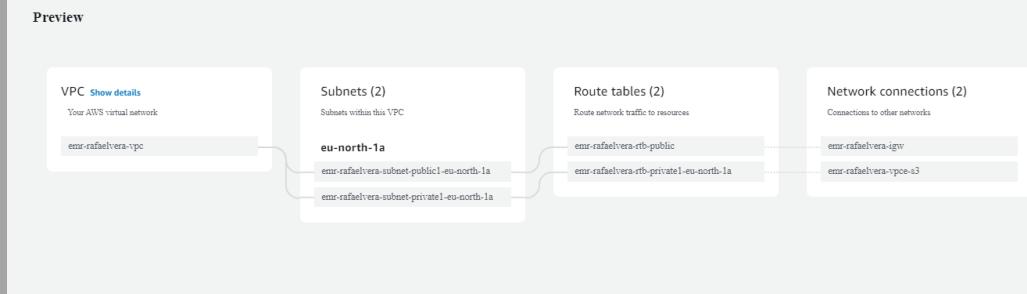
VPC endpoints
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.
None **S3 Gateway**

DNS options Info
 Enable DNS hostnames
 Enable DNS resolution

Additional tags

Cancel **Create VPC**

When finished click here



Number of Availability Zones (AZs) :

Selection of how many Availability Zones to use within the VPC for high availability. (**1**)

Customize AZs: Nothing to do here.

Number of public subnets (1) and Number of private subnets (1)

Customize Subnets CIDR Blocks: Nothing to do here.

NAT Gateway(s):

to be created along with the desired AZ for deployment. **NONE**

VPC Endpoints creation:

Allow private connections to AWS services. **S3 Gateway**

DNS Options:

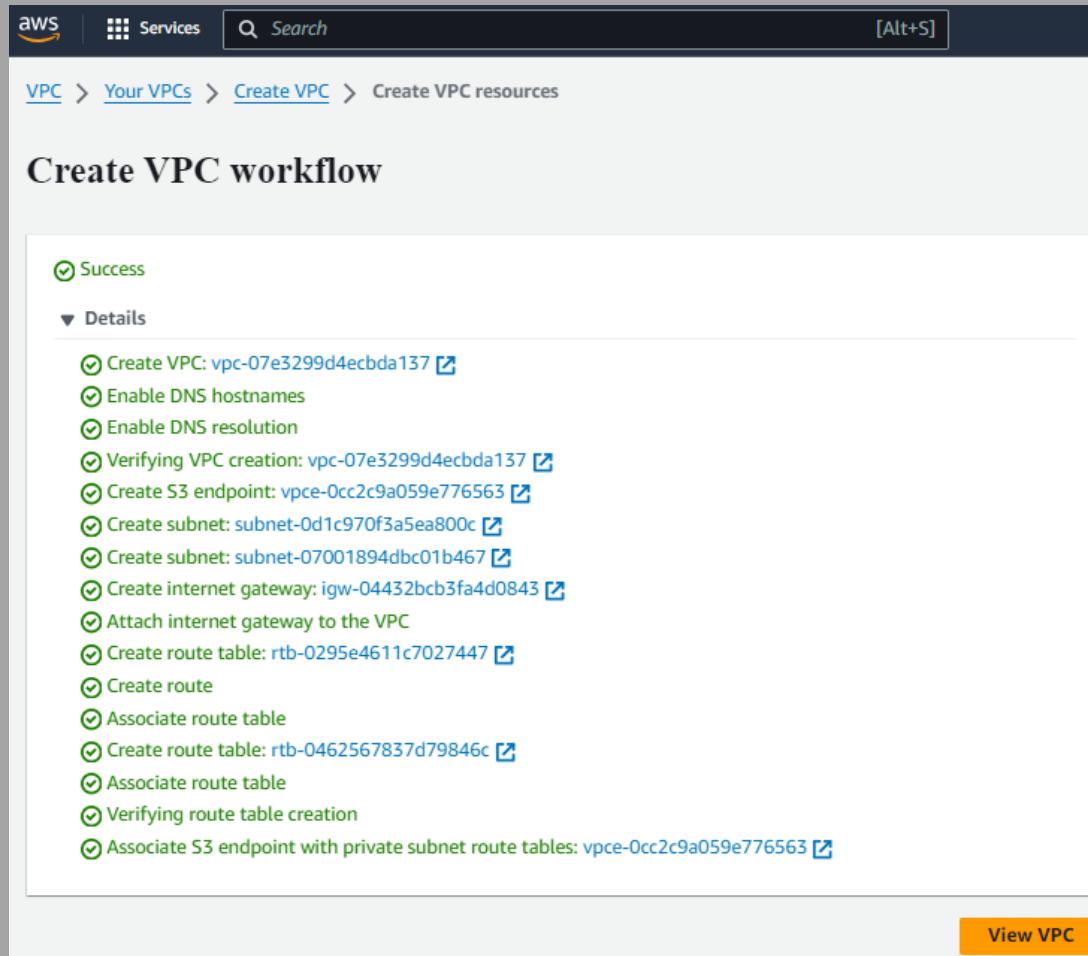
enable DNS hostnames and DNS resolution within the VPC.

Additional Tags: Nothing to do here.

Preview Pane:

two subnets designated as public and private across different availability zones, two route tables for network routing, and two network connections (external/internal)

2.5.a.4 VPC Create workflow



The screenshot shows the AWS VPC Create workflow summary page. At the top, there's a navigation bar with 'aws', 'Services', a search bar, and a keyboard shortcut '[Alt+S]'. Below the navigation, the breadcrumb trail reads 'VPC > Your VPCs > Create VPC > Create VPC resources'. The main title is 'Create VPC workflow'. On the left, there's a 'Success' status indicator and a 'Details' section. The 'Details' section lists 17 completed steps, each with a green checkmark and a blue link. At the bottom right of the list is a yellow 'View VPC' button.

Success

▼ Details

- >Create VPC: [vpc-07e3299d4ecbda137](#)
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: [vpc-07e3299d4ecbda137](#)
- Create S3 endpoint: [vpce-0cc2c9a059e776563](#)
- Create subnet: [subnet-0d1c970f3a5ea800c](#)
- Create subnet: [subnet-07001894dbc01b467](#)
- Create internet gateway: [igw-04432bcb3fa4d0843](#)
- Attach internet gateway to the VPC
- Create route table: [rtb-0295e4611c7027447](#)
- Create route
- Associate route table
- Create route table: [rtb-0462567837d79846c](#)
- Associate route table
- Verifying route table creation
- Associate S3 endpoint with private subnet route tables: [vpce-0cc2c9a059e776563](#)

View VPC

Create VPC workflow" summary:

The list confirms the steps taken to set up a VPC with necessary components for network connectivity, access to AWS services via endpoints, and proper routing.

At the bottom, there's a "**View VPC**" button, to inspect the newly created VPC configuration.

Create VPC: A VPC has been created with a specific ID.

Enable DNS hostnames: has been enabled.

Enable DNS resolution: has been activated.

Verifying VPC creation: The creation of the VPC has been verified.

Create S3 endpoint: has been established within the VPC.

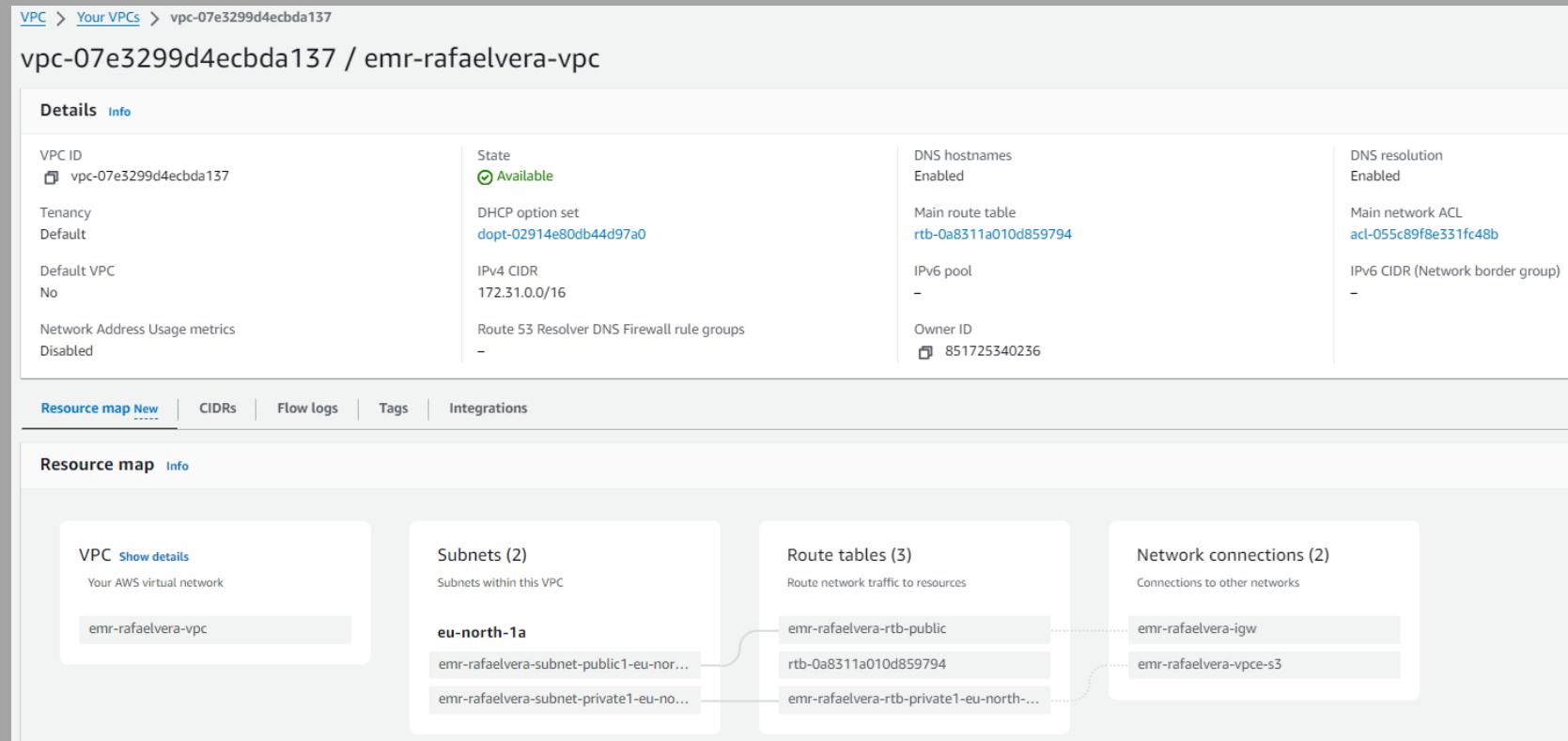
Create subnets: Two subnets have been created.

Create internet gateway: has been created and attached to the VPC.

Create route table: have been created and associated with the VPC.

Associate S3 endpoint with private subnet route tables: has been associated.

2.5.a.5 Created VPC Overview



This overview allows to verify the components of the VPC are set up as intended, and to navigate to further configuration details for each component.

Great! The VPC is now ready to be used with our EMR cluster within the defined network.
Come back to the [EMR Cluster Settings > Networking](#) to continue configuring it

Create an EMR cluster

2.5. Networking (VPC created) [Continued]

The screenshot shows the configuration steps for an EMR cluster's networking:

- Networking Info:** Shows the recently created VPC ID: `vpc-0f16318a4593502a1` and a public subnet ID: `subnet-02079acbd9625e283`.
- Choose VPC:** A modal window lists two VPCs:

Name	VPC ID	State	IPv4 CIDR
-	<code>vpc-0f16318a4593502a1</code>	Available	172.31.0.0/16
emr-rafaelvera-vpc	<code>vpc-07e5299d4ecbda137</code>	Available	172.31.0.0/16

A hand cursor points to the first VPC entry, labeled "This one".
- Choose subnet:** A modal window lists two subnets:

Name	Subnet ID	Type	IPv4 CIDR	AZ
emr-rafaelvera-subnet-public1-eu-north-1a	<code>subnet-0d1c970f5a5ea00c</code>	Public	172.31.0.0/20	eu-north-1a
emr-rafaelvera-subnet-private1-eu-north-1a	<code>subnet-07001894dbc01b467</code>	Private	172.31.128.0/20	eu-north-1a

A hand cursor points to the first public subnet entry.

Browse VPC:

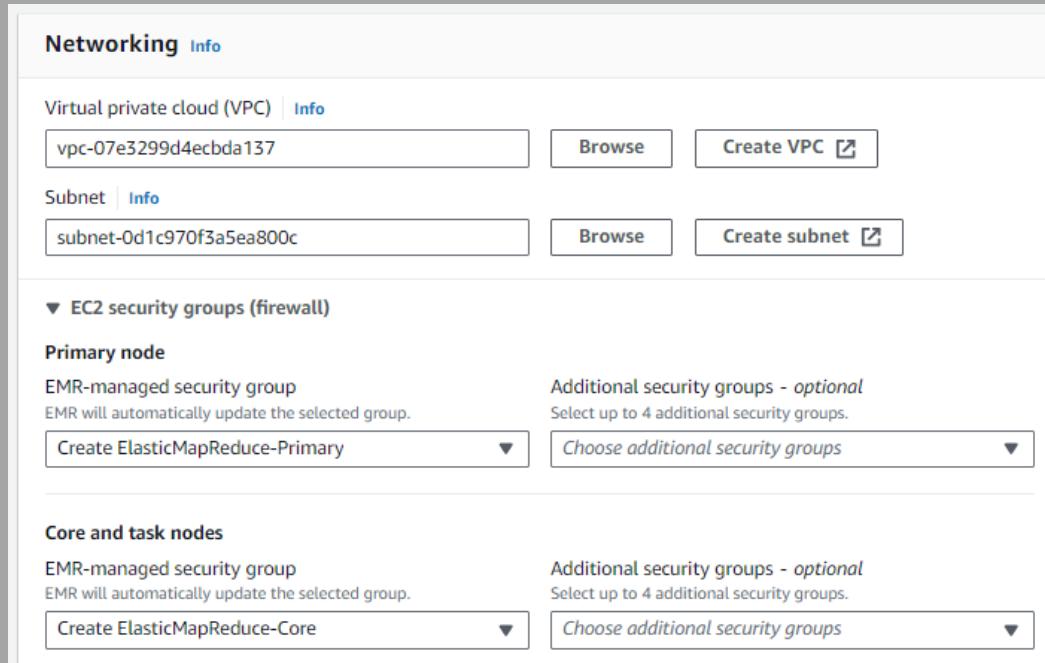
1. Select the VPC recently created.
2. Select the public subnet.

2nd part of Networking configuration

Rafael Vera



2.5.b.1 Setting EC2 Security groups



The screenshot shows the 'Networking' section of the EMR cluster creation wizard. It displays the selected VPC (vpc-07e3299d4ecbda137) and Subnet (subnet-0d1c970f3a5ea800c). Under 'EC2 security groups (firewall)', the 'Primary node' section shows an 'EMR-managed security group' dropdown set to 'Create ElasticMapReduce-Primary'. The 'Additional security groups - optional' dropdown is empty. The 'Core and task nodes' section also shows an 'EMR-managed security group' dropdown set to 'Create ElasticMapReduce-Core' and an 'Additional security groups - optional' dropdown which is empty.

With this step you are done with Networking configurations for the EMR cluster.

Let's move forward with the next EMR Cluster configurations

EC2 security groups (firewall):

In this EMR configuration act as virtual firewalls to control network traffic, ensuring secure communication between cluster nodes and safeguarding them from unauthorized access.

Configuring these groups is critical for defining precise access control policies for the different node types within an EMR cluster.

Primary Node:

EMR-managed security group:

This security group will be automatically updated by EMR.

Additional security groups - optional: Nothing to do here

Core and Task Nodes:

EMR-managed security group:

Similar to the primary node, security group for the core and task nodes.

Additional security groups - optional: Nothing to do here

Create an EMR cluster

2.6. Steps - Optional

▼ Steps - optional (0) [Info](#)

Use commands and scripts to tell your cluster where to find and how to process your data. Steps run consecutively unless you enable the Concurrency option.

Filter steps by status [▼](#) Find steps [🔍](#) [1](#) [2](#) [3](#) [...](#)

Name	Status	Type	Arguments	Script location
No steps				
You don't have any steps added.				

[Add](#)

Concurrency [Info](#)

Run multiple steps in parallel to improve cluster utilisation

Nothing to do here

This section is critical for automating data processing tasks within an EMR cluster, allowing for the scheduling of specific data processing activities that the cluster should perform.

Steps are used for defining commands and scripts to direct the cluster on where to find data and how to process it

Steps List:

List of steps available.

Add Step:

There is an "Add" button, suggesting that the user can add new steps to the cluster.

Concurrency:

By default, the steps defined for processing data in an EMR cluster are executed one after another, in the order they were added.

"Run multiple steps in parallel to improve cluster utilisation," if checked, would allow multiple steps to be executed at the same time, which can lead to more efficient use of the cluster's resources and potentially reduce the overall time taken for data processing.

2.7. Cluster termination

Cluster termination Info

Manually terminate cluster
 Automatically terminate the cluster after the last step ends
 Automatically terminate cluster after idle time (recommended)

Idle time
Enter the time until your cluster terminates.
0 days
Choose a time that is greater than 1 minute (00:01:00) and less than 7 days. The time is in hh:mm:ss (24-hour) format.

Use termination protection
Protect your EC2 instances from accidental termination.

These settings are designed to help manage costs and prevent accidental termination of the cluster, which can be particularly important in a pay-as-you-go cloud environment where running clusters incur charges.

Automatically Terminate After Idle Time (Recommended):
The recommended option where the cluster will automatically terminate after a specified period of inactivity.

Idle Time:

You can specify the duration of inactivity after which the cluster should be terminated.

Use Termination Protection:

Is a feature that safeguards EC2 instances within an AWS EMR cluster from being unintentionally shut down, ensuring that your cluster continues to run until you decide to terminate it explicitly.

Create an EMR cluster

2.8. Bootstrap actions - Optional

The screenshot shows a web-based interface for managing bootstrap actions. At the top, there's a header with a dropdown menu, a search bar, and several navigation links. Below this is a section titled "Bootstrap actions - optional (0)" with a "Info" link. A note says "Use bootstrap actions to install software or customise your instance configuration." There are three buttons: "Remove", "Edit", and "Add". A table follows, with columns for "Name", "Amazon S3 location", and "Arguments". The table shows one row with the status "No bootstrap actions" and the message "You don't have any bootstrap actions to display.". At the bottom is a large "Add" button.

Nothing to do here

Bootstrap actions are used to install additional software or to customize the configuration of cluster instances during their initialization phase. They are a powerful feature for users who need to run custom setup scripts or install additional software packages that are not included in the EMR base image.

2.9. Cluster logs - Optional

▼ Cluster logs – optional [Info](#)

We automatically archive your log files to Amazon S3. You can specify your own S3 location, or use the default S3 location for Amazon EMR. The default log location is pre-populated in the [Amazon S3 location](#) field.

Publish cluster-specific logs to Amazon S3

Amazon S3 location

s3://aws-logs-851725340236-eu-north-1/elasticmapreduce X View Browse S3

Format: Use s3://bucket/prefix

Encrypt cluster-specific logs

These settings are important for monitoring and debugging EMR clusters, as logs provide valuable information about the execution of cluster jobs and the performance of the cluster itself. The ability to encrypt logs adds an additional layer of security for sensitive data.

Automatic Archiving:

User can specify archiving the log files into their own S3 bucket or use the default S3 location pre-populated for Amazon EMR.

Publish Cluster-specific Logs:

Logs specific to the cluster will be published to Amazon S3.

Amazon S3 Location:

Shows the S3 bucket where logs will be stored.

Create an EMR cluster

2.10. Tags - Optional

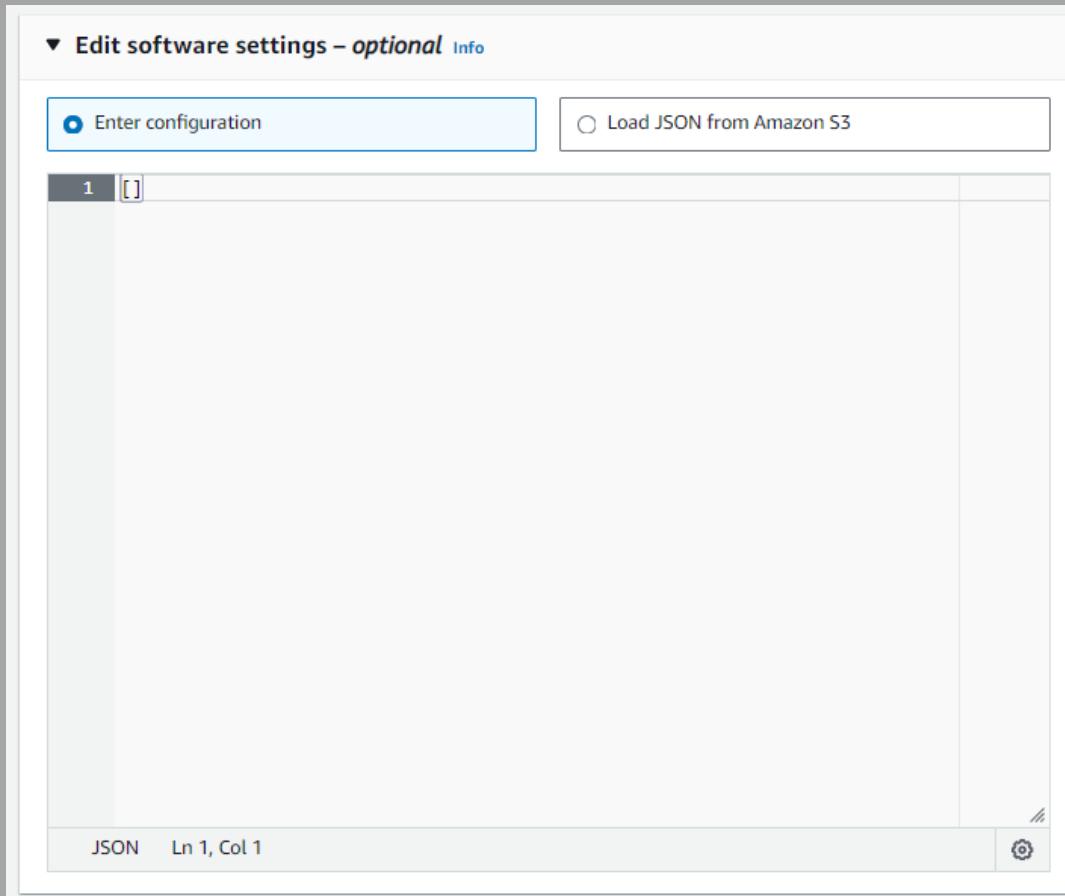
The screenshot shows a section titled "Tags – optional" with a "Info" link. A descriptive text explains that tags are labels assigned to AWS resources. Below this, it states "No tags associated with the resource." There is a button labeled "Add new tag" and a note indicating "You can add 50 more tags."

Nothing to do here

Tags are useful for organizing and managing AWS resources, especially for cost allocation and reporting in larger environments with multiple resources and services.

A **TAG** is a label assigned to an AWS resource consisting of a key and an optional value. Tags are used for searching and filtering resources or tracking AWS costs.

2.11. Edit software settings - Optional



Nothing to do here

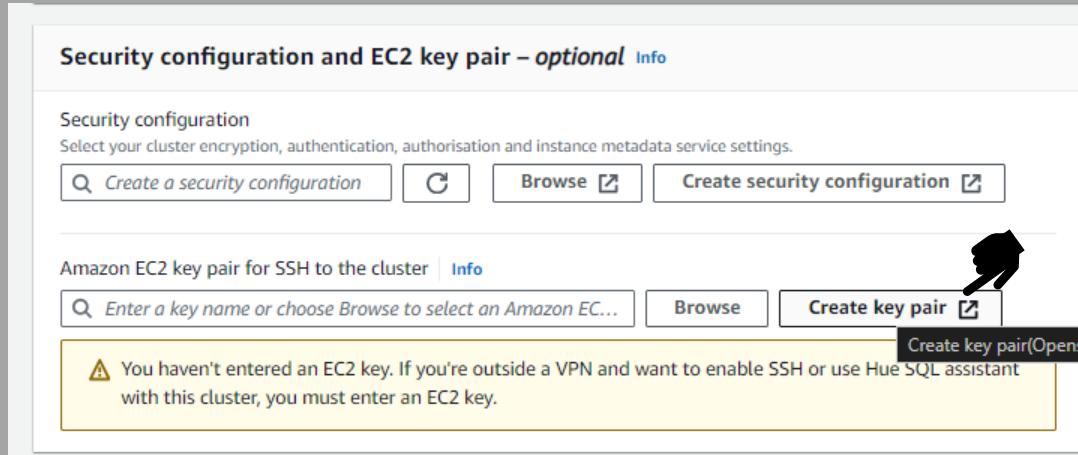
This interface allows users to specify detailed configurations for software on their EMR cluster, such as setting properties for Hadoop, Spark, and other applications within the cluster.

Users can fine-tune the behavior of their EMR cluster software using these configurations.

You can either enter the software configuration settings manually or you can load it from a JSON file stored in an Amazon S3 bucket.

Create an EMR cluster

2.12. Security config and EC2 key pair- Optional



Security configuration and EC2 key pair - *optional* [Info](#)

Security configuration

Select your cluster encryption, authentication, authorisation and instance metadata service settings.

Amazon EC2 key pair for SSH to the cluster [Info](#)

⚠ You haven't entered an EC2 key. If you're outside a VPN and want to enable SSH or use Hue SQL assistant with this cluster, you must enter an EC2 key.

EC2 key pair:

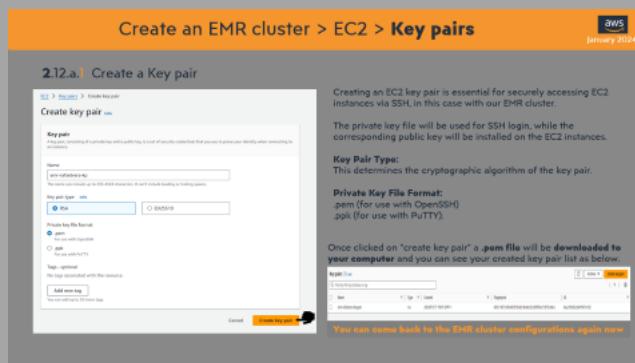
is crucial for securely accessing cluster instances via SSH, which is important for administration tasks, such as troubleshooting or managing the software on the instances. Without this, remote access to the cluster's nodes would not be possible.

Security Configuration: Nothing to do here

Determines the cluster's encryption, authentication, authorization, and instance metadata service settings.

Amazon EC2 Key Pair for SSH to the Cluster:
Hit on create key pair and start creating it.

Create Key pairs (a)



2.12.a] Create a Key pair

Creating an EC2 key pair is essential for securely accessing EC2 instances via SSH, in this case with our EMR cluster.

The private key file will be used for SSH login, while the corresponding public key will be installed on the EC2 instances.

Key Pair Type:
This determines the cryptographic algorithm of the key pair.

Private Key File Format:
pem (for use with OpenSSH)
ssh (for use with PuTTY).

Once clicked on "create key pair" a .pem file will be downloaded to your computer and you can see your created key pair list as below.

[Create key pair](#)

2.12.a.1 Create a Key pair

EC2 > Key pairs > Create key pair

Create key pair Info

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name: emr-rafaelvera-kp
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type: RSA ED25519

Private key file format: .pem For use with OpenSSH .ppk For use with PuTTY

Tags - optional:
No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Cancel **Create key pair** 

Creating an EC2 key pair is essential for securely accessing EC2 instances via SSH, in this case with our EMR cluster.

The private key file will be used for SSH login, while the corresponding public key will be installed on the EC2 instances.

Key Pair Type:

This determines the cryptographic algorithm of the key pair.

Private Key File Format:

.pem (for use with OpenSSH)
.ppk (for use with PuTTY).

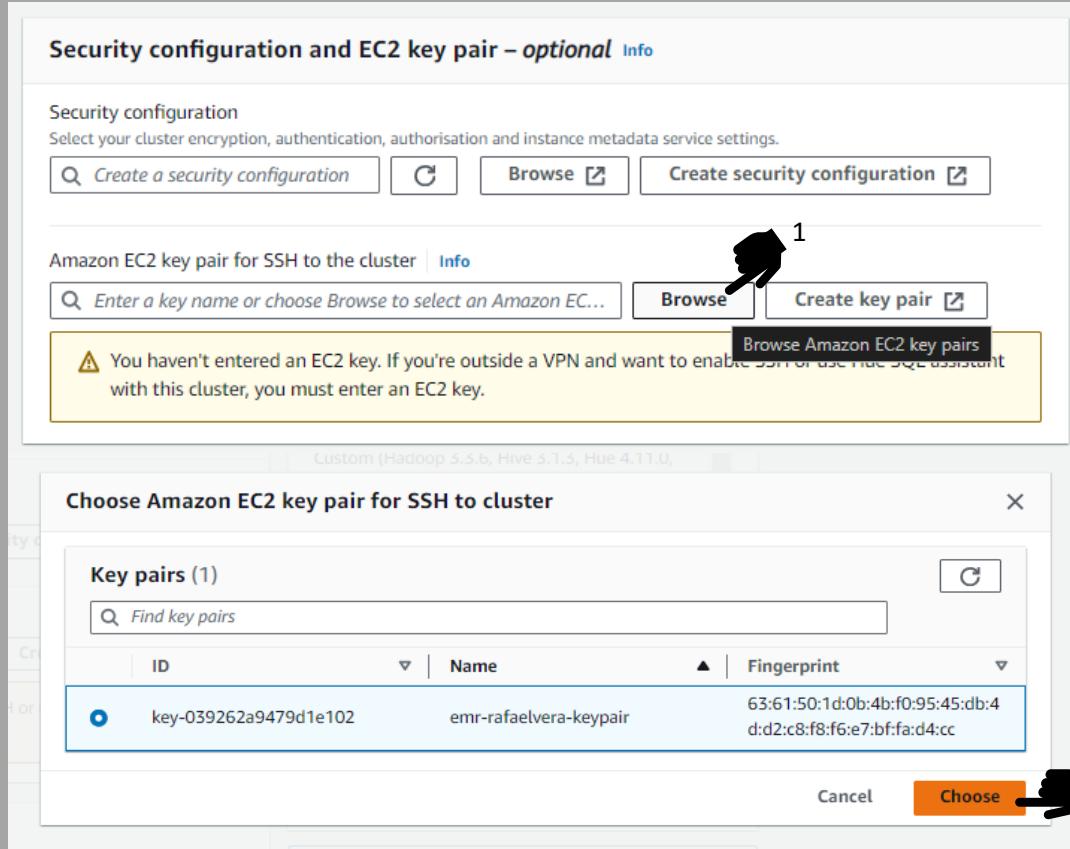
Once clicked on “create key pair” a **.pem file** will be **downloaded to your computer** and you can see your created key pair list as below.

Key pairs (1) <small>Info</small>				
<input type="text"/> Find Key Pair by attribute or tag				
	Name	Type	Created	Fingerprint
	emr-rafaelvera-keypair	rsa	2024/01/17 19:07 GMT+1	63:61:50:1d:0b:f0:95:45:db:4d:2c:8:f8:f6:e7:bf:fa:d4:cc

You can come back to the EMR cluster configurations again now

Create an EMR cluster

2.12. EC2 key pair- Optional [Continued]



Amazon EC2 Key Pair for SSH to the Cluster:

1. Hit on browse key pair and select the recently created Key Pair.
2. Click on **Choose**.

Create an EMR cluster

2.13. IAM roles

Identity and Access Management (IAM) roles Info

Choose or create a service role and instance profile for the EC2 instances in your cluster.

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.

Service role

Choose an IAM role ▾ C

Q |

No IAM roles were found.

EC2 instance profile for Amazon EMR

The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

Choose an existing instance profile
Select a default role or a customised instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a customised set of resources for it to access in Amazon S3.

Instance profile

Choose an IAM role ▾ C

Amazon EMR Service Role:

IAM roles and instance profiles are critical for securing EMR clusters and controlling access to AWS resources. They ensure that the cluster has only the necessary permissions to operate effectively and securely, following the principle of least privilege.

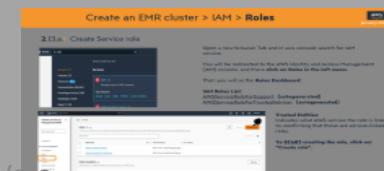
EC2 Instance Profile for Amazon EMR:

assigns a role to EC2 instances in the cluster that allows them to access AWS resources necessary for cluster steps and bootstrap actions

The first time you see this configuration will show as the image on the left and you aren't able to select any existing Service role, so you must “Create a service role” first.

To start with it you need to follow the next steps across the **aws IAM service** to create both a Service role and Instance profile.

Create Service role (a)

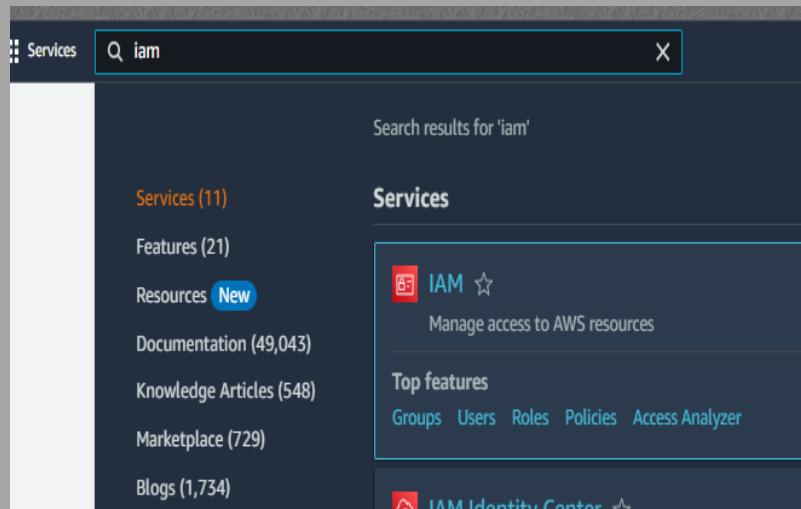


Rafael Velez

Create Instance Profile role (b)



2.13.a.1 Create Service role



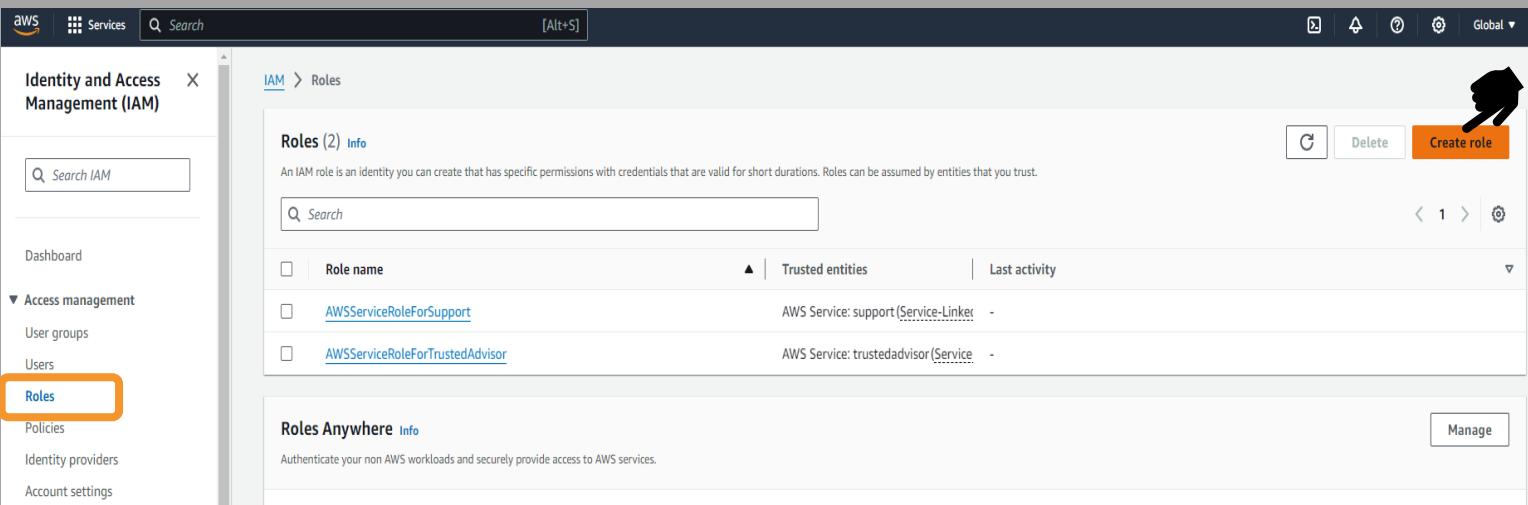
Open a new browser Tab and in aws console search for IAM service.

You will be redirected to the AWS Identity and Access Management (IAM) console, and there **click on Roles in the left menu**.

Then you will see the **Roles Dashboard**:

IAM Roles List:

AWSServiceRoleForSupport: (autogenerated)
AWSServiceRoleForTrustedAdvisor: (autogenerated)



A screenshot of the AWS IAM Roles dashboard. The left sidebar is expanded, showing 'Identity and Access Management (IAM)' and 'Access management' with 'Roles' selected. The main content area shows a table of roles:

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-

At the top right of the table, there is a 'Create role' button with a hand cursor icon pointing to it.

Trusted Entities:

Indicates what AWS service the role is linked to, confirming that these are service-linked roles.

To **START creating the role, click on: "Create role".**

2.13.a.2 Create Service role

Select trusted entity [Info](#)

Trusted entity type

AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EMR

Choose a use case for the specified service.

Use case

EMR
Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.

EMR Role for EC2
Allows EC2 instances in an Elastic MapReduce cluster to call AWS services such as S3 on your behalf.

EMR - Cleanup
Allows EMR to terminate instances and delete resources from EC2 on your behalf.

 Click here to proceed

Cancel Next

This interface defines the permissions and scope of an IAM role, ensuring that it is tailored to the needs of the EMR service

Trusted Entity Type:

AWS service:

indicating that the role is intended for AWS services such as EC2 to perform actions within the user's AWS account.

AWS account: For allowing entities in other AWS accounts to perform actions in the user's account.

Web identity: For use with identity providers that are federated through web identity (like Google, Facebook, or Amazon).

SAML 2.0 federation: For federation with SAML 2.0 from a corporate directory.

Custom trust policy: For creating a custom trust policy.

Use case:

EMR: Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.

EMR Role for EC2: Allows EC2 instances in an Elastic MapReduce cluster to call AWS services on your behalf.

EMR - Cleanup: Allows EMR to terminate instances and delete resources from EC2 on your behalf.

2.13.a.3 Create Service role

IAM > Roles > Create role

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Add permissions [Info](#)

Permissions policies (1) [Info](#)
The type of role that you selected requires the following policy.

Policy name Type
[AmazonElasticMapReduceRole](#) AWS managed

Set permissions boundary - optional

Cancel Previous Next

IAM > Roles > Create role

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
emr-rafaelvera-role

Description
Add a short explanation for this role.
Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.

Maximum 1000 characters. Use alphanumeric and '+,-_,@-' characters.

Step 1: Select Trusted entity

Made on the slide before.

Step 2: Add permissions

Involves attaching policies that grant the IAM role the necessary permissions to function correctly within AWS EMR.

[AmazonElasticMapReduceRole](#) is an AWS managed policy designed to grant the necessary permissions for an Amazon EMR role.

Step 3: Name, review and create

Role name: is used to identify the role within AWS.

The description: context for what the role is intended to do.

Add tags: prompts to add metadata tags to the IAM role for identification, organization, or search purposes.

IAM > Roles > Create role

Step 1 Select trusted entities

Trust policy

```
1+ "Version": "2012-10-17",
2- "Statement": [
3-   {
4-     "Effect": "Allow",
5-     "Action": [
6-       "sts:AssumeRole"
7-     ],
8-     "Principal": [
9-       "elasticmapreduce.amazonaws.com"
10-     ]
11-   }
12- ]
13- ]
14- ]
15- ]
16- ]
```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonElasticMapReduceRole	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional [Info](#)
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Cancel Previous Create role

Rafael Vera

Review of role creation showing 3 main steps:
Just click Create role

2.13.a.4 Create Service role - Created

Role emr-rafaelvera-role created.

IAM > Roles

Roles (3) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-
emr-rafaelvera-role	AWS Service: elasticmapreduce	-

emr-rafaelvera-role Info

Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.

Summary

Creation date: January 17, 2024, 19:21 (UTC+01:00)
Last activity: -
ARN: arn:aws:iam:851725340236:role/emr-rafaelvera-role
Maximum session duration: 1 hour

Permissions

Permissions policies (1) Info

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonElasticMapReduceRole	AWS managed	1

Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

Generate policy

No requests to generate a policy in the past 7 days.

The **role** is already **created** and appears **listed** on our roles console.

We need **add an additional Permission policy to this role** in order to having access to the S3 Buckets service, which will be needed afterwards.

- 1st. Click on the **role name** as shown on the image.
- 2nd. On the Role configuration screen select **attach policies**.
- 3rd. Search for **AmazonS3 full access**.
- 4th. Add the new permission policy.
- 5th. See the two inlisted Permission policies.
- 6th. Go back to the **EMR Cluster configurations**

Policy was successfully attached to role.

emr-rafaelvera-role Info

Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.

Summary

Creation date: January 17, 2024, 19:21 (UTC+01:00)
Last activity: -
ARN: arn:aws:iam:851725340236:role/emr-rafaelvera-role
Maximum session duration: 1 hour

Permissions

Permissions policies (2) Info

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonElasticMapReduceRole	AWS managed	1
AmazonS3FullAccess	AWS managed	1

Create an EMR cluster

2.13. IAM roles [continued]

Identity and Access Management (IAM) roles [Info](#)
Choose or create a service role and instance profile for the EC2 instances in your cluster.

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.

Service role
emr-rafaelvera-role

EC2 instance profile for Amazon EMR
The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

Choose an existing instance profile
Select a default role or a customised instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a customised set of resources for it to access in Amazon S3.

Instance profile
Choose an IAM role

No IAM roles were found.

Custom automatic scaling role - optional
When a custom automatic scaling rule triggers, Amazon EMR assumes this role to add and terminate EC2 instances. [Find out more](#)

Custom automatic scaling role
Choose an IAM role

[Create an IAM role](#)

Amazon EMR Service Role:

Back to here,

1st. Select the Service role already created from the dropdown list.

EC2 Instance Profile for Amazon EMR:

Defines access scope of the EMR cluster's EC2 instances to S3 resources (data storage and operations performed by the cluster).

2nd. Select “Create an instance profile”

3rd. Click on Browse S3 and we haven't S3 buckets instances.

4th. Follow the next guide steps -> Create instance profile (b)

EC2 instance profile for Amazon EMR
The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

Choose an existing instance profile
Select a default role or a customised instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a customised set of resources for it to access in Amazon S3.

S3 bucket access [Info](#)
 Specific S3 buckets or prefixes in your account [Info](#)
Choose the buckets or prefixes that you want this instance profile to access.
 All S3 buckets in this account with read and write access
Grant the instance profile access to all buckets that have read and write access enabled in your account.

S3 buckets
We've already added the resources that you configured in the Cluster logs section. Choose the S3 buckets and bucket prefixes where you store logs and data for your cluster, bootstrap actions and steps.

S3 URI
s3://bucket/prefix/object

S3 bucket aws-logs-85172534...
Prefix elasticmapreduce
Permission Read and write

[Edit](#)

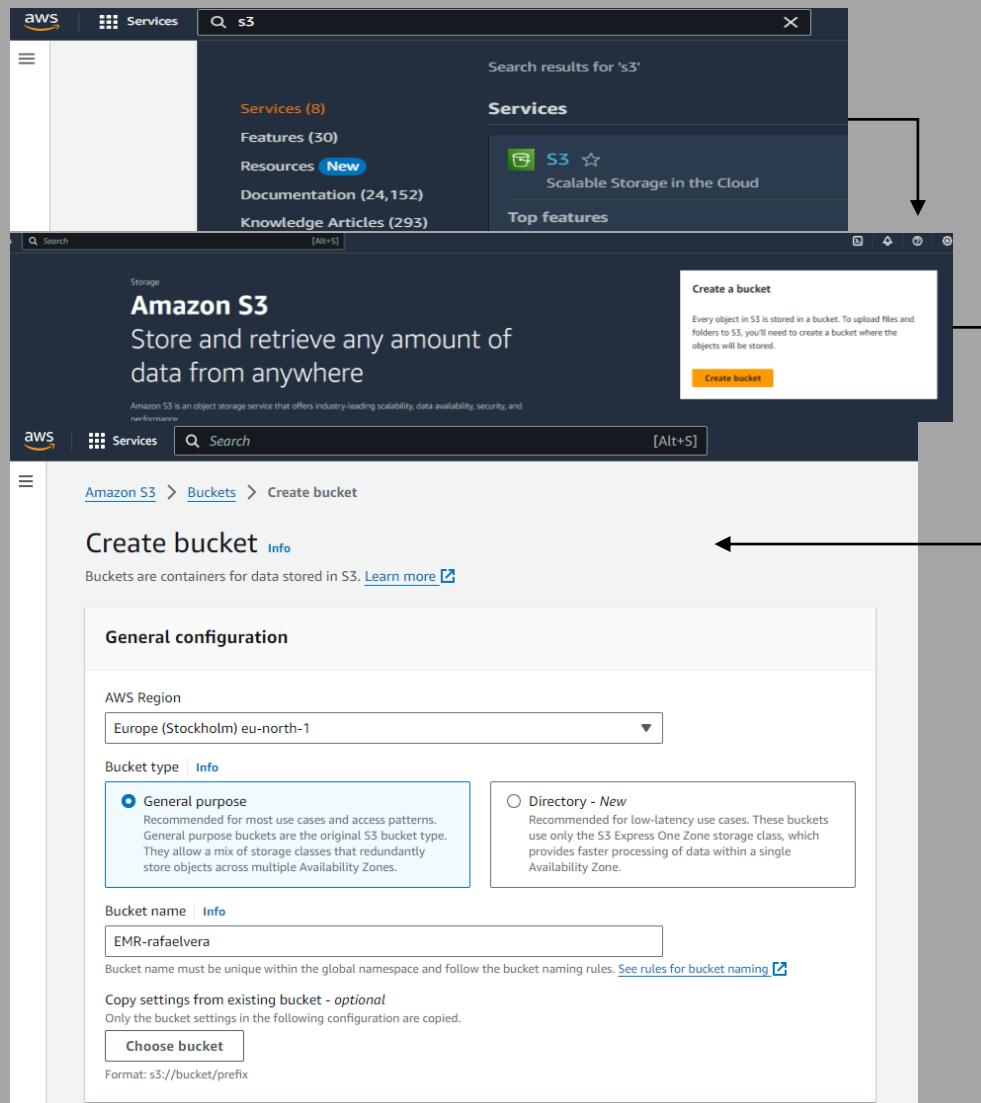
Create Instance Profile (b)

Create an EMR cluster > IAM > Instance Profile (\$3Bucket)

2.13.b. Create Instance Profile role (\$3 Bucket)
To open before we need to create a S3 Bucket before continue the EMR Cluster configuration.
1#. Open a new tab and search in aws console for S3.
2#. Creating the Bucket. Only the Bucket Name is going to be given.
the rest remains stay as default.
3#. Go to the Bucket and click on Properties.
4#. Once created the S3 Bucket, go back to EMR Cluster configurations > IAM Roles -> EC2 Instance Profile for Amazon EMR.

General Configuration:
AWS Region: Europe (Stockholm) eu-north-1 region.
Bucket Type: General purpose. Recommended for most use cases and allows for storing objects across multiple Availability Zones.
Bucket Name: the intended name for the new bucket.
Copy Settings from Existing Bucket - optional. Nothing to do here.

2.13.b.1 Create Instance Profile role (S3 Bucket)



As seen before we need to create a S3 Bucket before continue the EMR Cluster configuration.

- 1st. Open a new tab and search in aws console for S3.
 - 2nd. Create a Bucket.
 - 3rd. Creating the Bucket. **Only the Bucket Name is going to be given, the rest options stay by default.**
- Let's see it step by step in the next slides.

- 4th. Once created the S3 Bucket, go back to EMR Cluster configurations > IAM roles > **EC2 Instance Profile for Amazon EMR**.

General Configuration:

AWS Region: "Europe (Stockholm) eu-north-1" region.

Bucket Type: General purpose. Recommended for most use cases and allows for storing objects across multiple Availability Zones.

Bucket Name: the intended name for the new bucket.

Copy Settings from Existing Bucket - optional: Nothing to do here

Rafael Vera

2.13.b.2 Create Instance Profile role (S3 Bucket)

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

These settings maintain the security and privacy of the data stored in the S3 bucket. By enforcing bucket ownership and blocking public access, the bucket owner can prevent unauthorized access and data leaks.

ACLs disabled (recommended): Indicates that access control lists (ACLs) are disabled, and all objects in the bucket are owned by the current account. Access to the bucket and its objects is managed through policies.

The "Bucket owner enforced" setting is highlighted meaning that the bucket owner will retain ownership of all uploaded objects.

2.13.b.3 Create Instance Profile role (S3 Bucket)

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning
 Disable
 Enable

Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.
[Add tag](#)

Default encryption [Info](#)
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)
 Server-side encryption with Amazon S3 managed keys (SSE-S3)
 Server-side encryption with AWS Key Management Service keys (SSE-KMS)
 Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)
 Disable
 Enable

These settings maintain data integrity and security within an S3 bucket. Versioning helps with data recovery, tags provide organization and cost management, and encryption ensures data confidentiality and security.

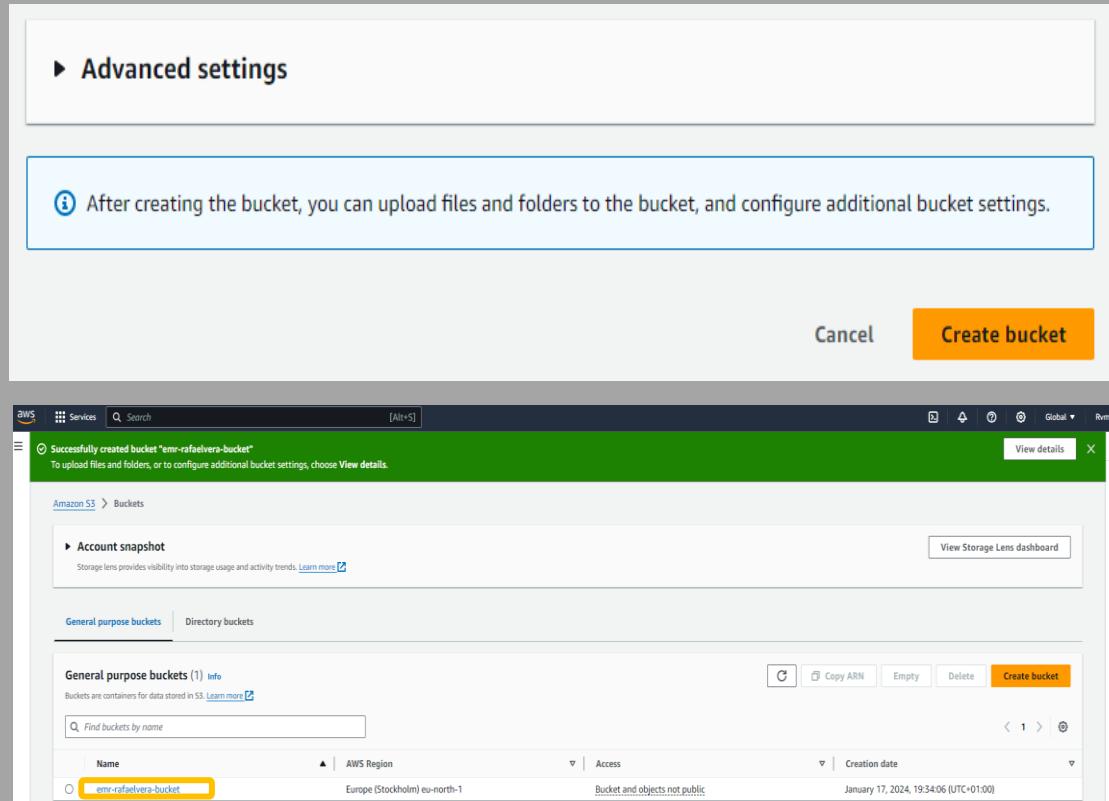
Tags – Optional: Nothing to do here

Default Encryption:

Describes the server-side encryption automatically applied to new objects stored in this bucket.

"Server-side encryption with Amazon S3 managed keys (SSE-S3)": This option means objects will be encrypted with keys managed by S3.

2.13.b.4 Create Instance Profile role (S3 Bucket)



Advanced settings: Nothing to do here

Click on “CREATE BUCKET”

Amazon S3 console within the AWS Management Console: showing an overview of S3 buckets.

Bucket Information:

Bucket name: "emr-rafaelvera-bucket".

AWS Region: "Europe (Stockholm) eu-north-1".

Creation Date: January 17, 2024, 19:34:06 UTC+00:00".

We are ready to come back to the EMR Configuration again and select this bucket in there.

Create an EMR cluster

2.13. IAM roles [continued]

EC2 instance profile for Amazon EMR
The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

Choose an existing instance profile
Select a default role or a customised instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a customised set of resources for it to access in Amazon S3.

S3 bucket access | Info
 Specific S3 buckets or prefixes in your account | Info
Choose the buckets or prefixes that you want this instance profile to access.
 All S3 buckets in this account with read and write access
Grant the instance profile access to all buckets that have read and write access enabled in your account.

S3 buckets
We've already added the resources that you configured in the Cluster logs section. Choose the S3 buckets and bucket prefixes where you store logs and data for your cluster, bootstrap actions and steps.

S3 URI
s3://emr-rafaelvera-bucket X View Edit | Browse S3 Add | Edit

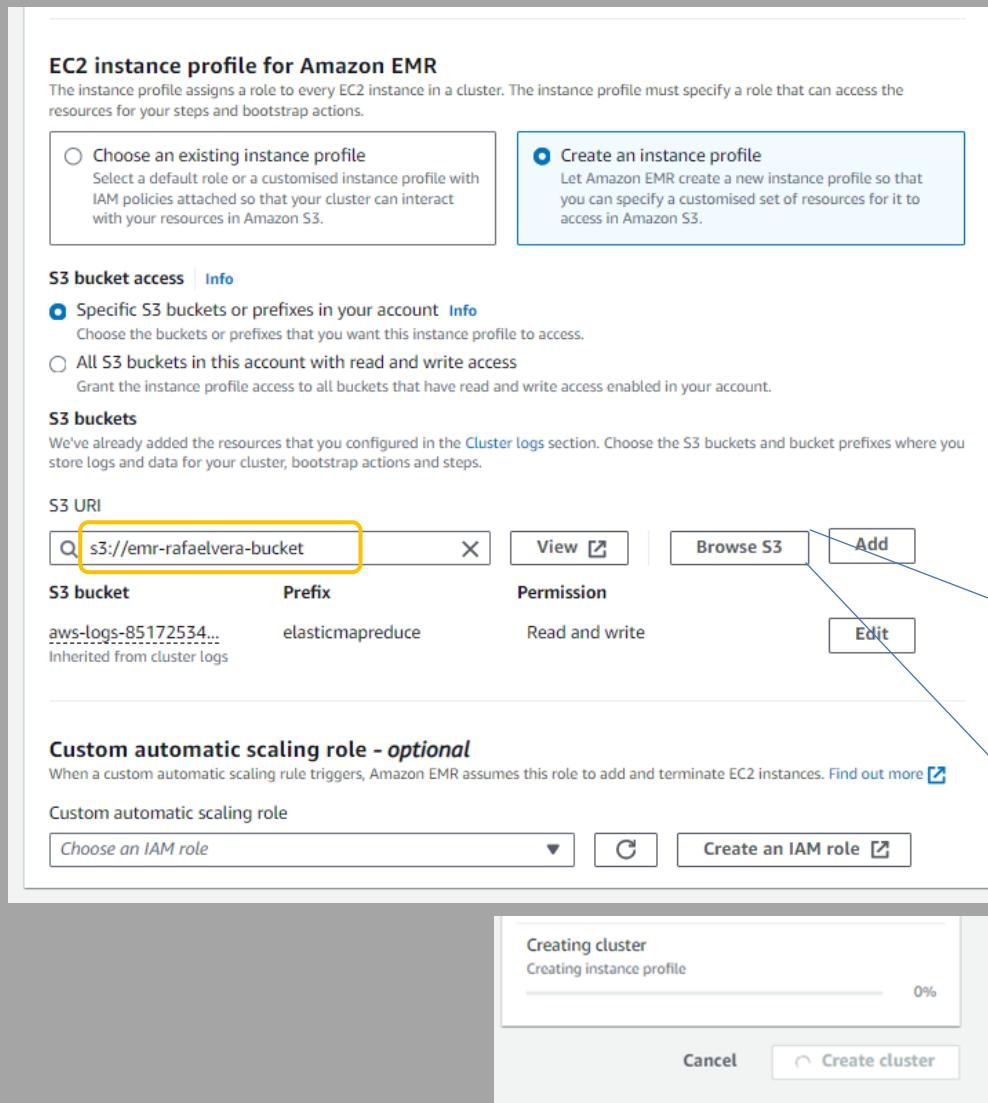
S3 bucket	Prefix	Permission
aws-logs-85172534...	elasticmapreduce	Read and write

Inherited from cluster logs

Custom automatic scaling role - optional
When a custom automatic scaling rule triggers, Amazon EMR assumes this role to add and terminate EC2 instances. [Find out more](#)

Custom automatic scaling role
Choose an IAM role C Create an IAM role Create

Creating cluster
Creating instance profile 0%
Cancel Create cluster



EC2 Instance Profile for Amazon EMR > S3 Bucket

This configuration settings define how the EMR cluster will interact with S3 for data storage and scaling policies for efficient resource management.

Just click on “BROWSE S3” to find the recently created S3Bucket out, and “CHOOSE” it.

After that, we are done.

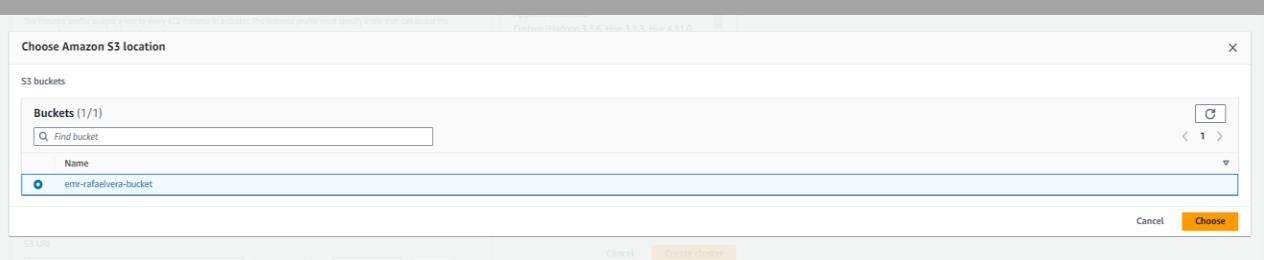
Hit the button “CREATE CLUSTER” to start creating the EMR Cluster instance and wait for while until being redirected to next screen.

Choose Amazon S3 location

S3 buckets

Buckets (1/1)
Q Find bucket
Name emr-rafaelvera-bucket

Cancel Choose



Create an EMR cluster

aws
January 2024

2.14. Created EMR Cluster

The screenshot shows the AWS Management Console for the Amazon EMR (Elastic MapReduce) cluster 'EMR RafaelVera'. The status is 'Starting'. A yellow box highlights the 'Status and time' section, which includes the status, creation time (17 January 2024 19:37 UTC+01:00), and elapsed time (-2 seconds). The 'Cluster management' section shows log destination in Amazon S3 (aws-log-851725340236-eu-north-1/elasticmapreduce) and primary node public DNS. The 'Applications' section lists Amazon EMR version (emr-7.0.0) and installed applications (Hadoop 3.3.6, Hive 3.1.3, Hue 4.11.0, JupyterEnterpriseGateway 2.6.0, JupyterHub 1.5.0, Livy 0.7.1, Pig 0.17.0, Spark 3.5.0, Tez 0.10.2). The 'Cluster logs' section shows archive log files to Amazon S3 and termination info. The 'Network and security' section shows network configuration (Virtual Private Cloud (VPC) vpc-07e329964edcda137, Subnet ID and Availability Zone (AZ) subnet-0d1c970f5a5ea800 eu-north-1a) and security configuration (Security configuration none, EC2 key pair emr-rafaelvera-keypair). The 'Properties' tab is selected.

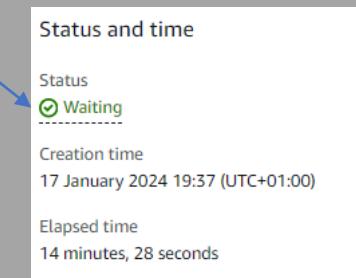
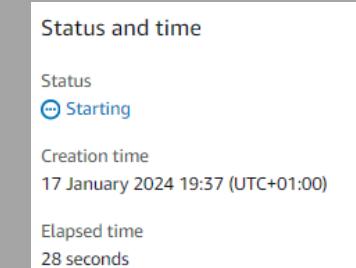
The screenshot shows the AWS Management Console for the Amazon EMR (Elastic MapReduce) cluster 'EMR RafaelVera'. The status is 'Waiting'. A yellow box highlights the 'Status and time' section, which includes the status, creation time (17 January 2024 19:37 UTC+01:00), and elapsed time (14 minutes, 58 seconds). The 'Cluster management' section shows log destination in Amazon S3 (aws-log-851725340236-eu-north-1/elasticmapreduce) and primary node public DNS (ec2-13-53-170-251.eu-north-1.compute.amazonaws.com). The 'Applications' section lists Amazon EMR version (emr-7.0.0) and installed applications (Hadoop 3.3.6, Hive 3.1.3, Hue 4.11.0, JupyterEnterpriseGateway 2.6.0, JupyterHub 1.5.0, Uvy 0.7.1, Pig 0.17.0, Spark 3.5.0, Tez 0.10.2). The 'Cluster logs' section shows archive log files to Amazon S3 and termination info. The 'Network and security' section shows network configuration (Virtual Private Cloud (VPC) vpc-07e329964edcda137, Subnet ID and Availability Zone (AZ) subnet-0d1c970f5a5ea800 eu-north-1a) and security configuration (Security configuration none, EC2 key pair emr-rafaelvera-keypair). The 'Properties' tab is selected.

AWS Management Console view of the summary page for the Amazon EMR (Elastic MapReduce) cluster:

This summary page provides a comprehensive view of the cluster's setup, including configuration, network and security settings, software applications, and management settings. Allows therefore an easy monitoring and management of the EMR cluster.

Just wait for Status and Time switch from **Starting** to **Waiting** (aprox 15 min)

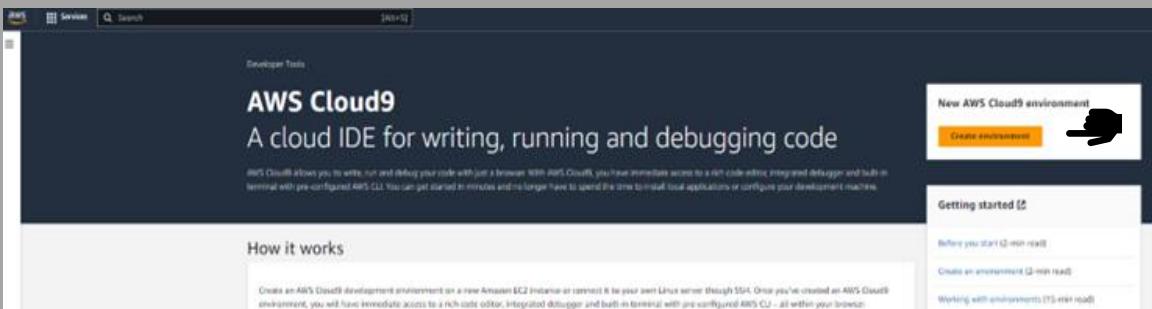
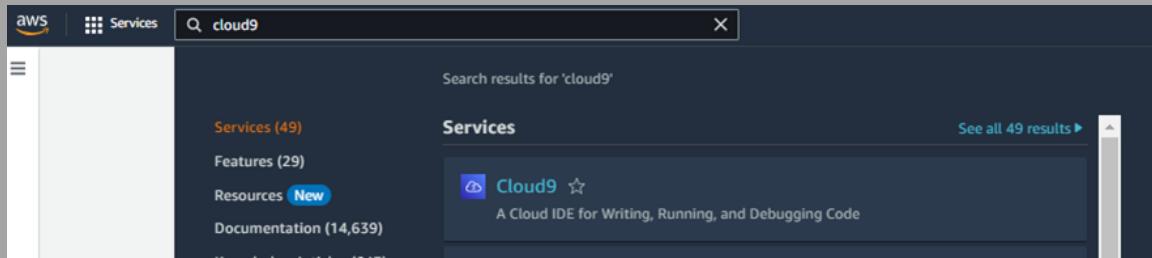
Grab a cup of coffee, you are doing it great!



EMR Cluster ready to use.

Move forward to next guide section

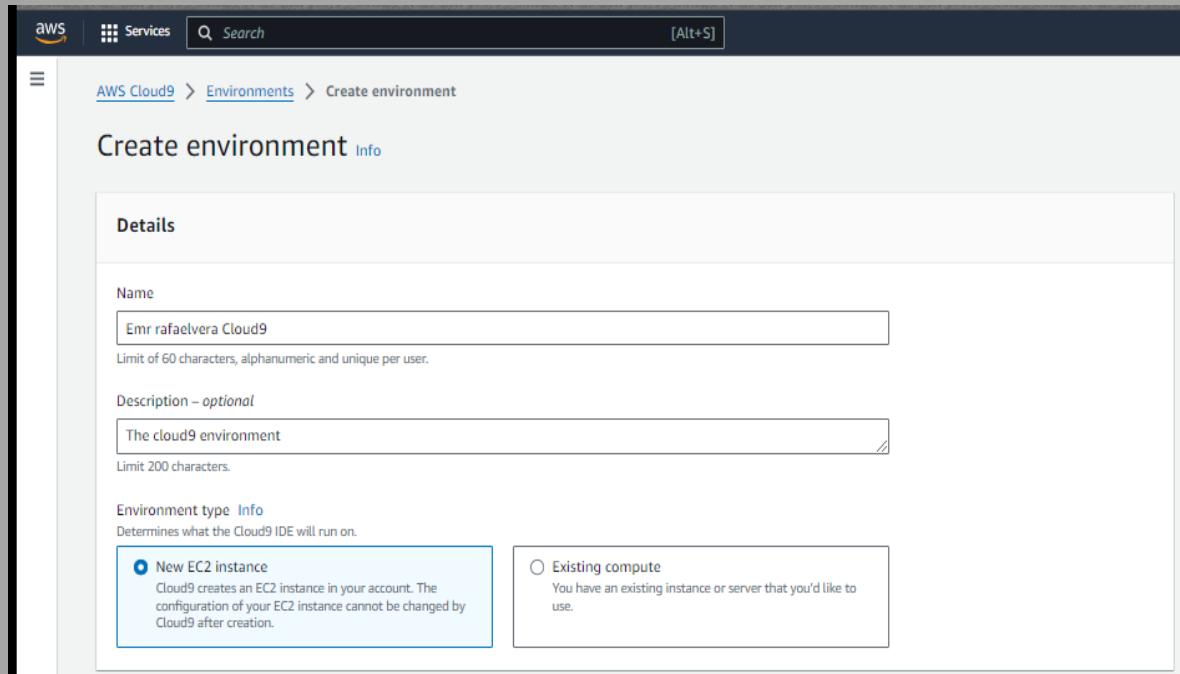
3. Aws Cloud9 IDE



We are instantiating the AWS Cloud9 service and use it as our IDE to run the Data processing script (python) within our EMR cluster.

1. Open a new Browser tab for aws services
2. Search for cloud 9 and click on it
3. You will be landed on aws Cloud9 service homepage
4. Click on create environment
5. Follow the next steps to configure the cloud9 instance environment

3. 1. Create Environment



The screenshot shows the 'Create environment' page in the AWS Cloud9 interface. The 'Name' field contains 'Emr rafaelvera Cloud9'. The 'Description' field contains 'The cloud9 environment'. Under 'Environment type', the 'New EC2 instance' option is selected, with a note explaining it creates a new EC2 instance in the account. The 'Existing compute' option is also present but not selected.

AWS Cloud9 > Environments > Create Environment

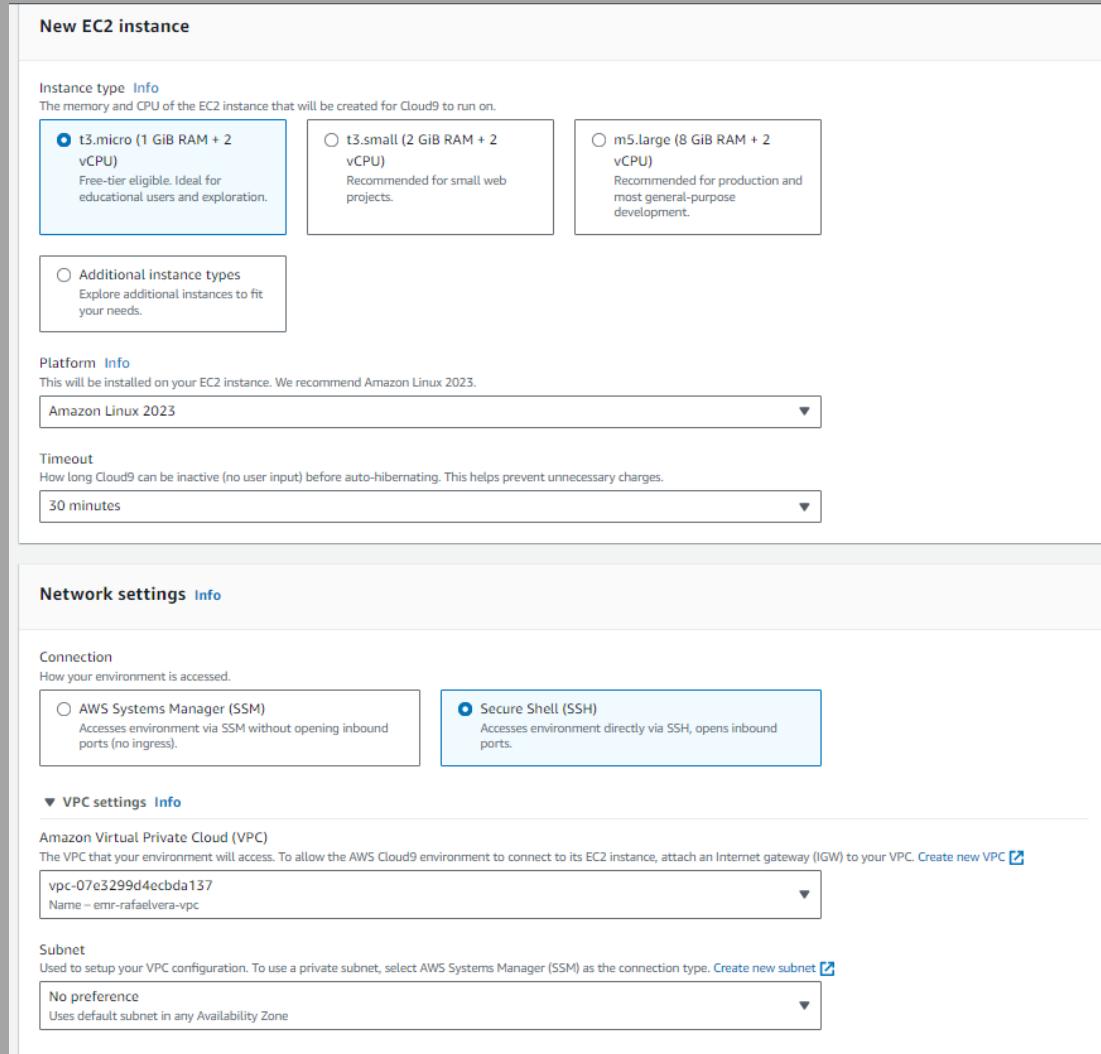
This interface guides us through the setup of the new Cloud9 IDE environment, with options to configure the underlying compute resources. It simplifies setting up a development workspace in the cloud, integrated with AWS services.

Environment Type:

New EC2 Instance: Selected by default. Cloud9 will create a new EC2 instance for the environment.

Existing Compute: If selected, this would allow the user to connect the Cloud9 environment to an existing instance or server.

3. 2. New EC2 Instance & Network settings



New EC2 instance

Instance type Info
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

- t3.micro (1 GiB RAM + 2 vCPU)
Free-tier eligible. Ideal for educational users and exploration.
- t3.small (2 GiB RAM + 2 vCPU)
Recommended for small web projects.
- m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.

Additional instance types
Explore additional instances to fit your needs.

Platform Info
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023

Timeout
How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes

Network settings Info

Connection
How your environment is accessed.

- AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).
- Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.

VPC settings Info

Amazon Virtual Private Cloud (VPC)
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an Internet gateway (IGW) to your VPC. Create new VPC

vpc-07e3299d4ecbda137
Name - emr-rafaelvera-vpc

Subnet
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. Create new subnet

No preference
Uses default subnet in any Availability Zone

New EC2 instance:
will host the AWS Cloud9 environment.

Instance Type:
t3.micro (1 GiB RAM + 2 vCPU): free-tier eligible.

Platform:
set to "Amazon Linux 2023", indicates the operating system that will be installed on the EC2 instance.

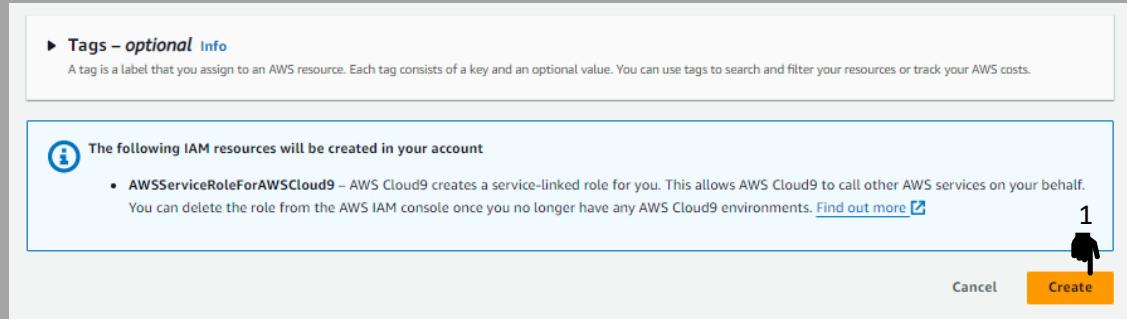
Timeout:
Set the duration for how long the Cloud9 environment can be inactive before auto-hibernating, to prevent unnecessary charges.

Network Settings:
Secure Shell (SSH): Direct access to coud9 environment via SSH, which opens inbound ports.

VPC Settings:
Select the VPC created in steps before.
In my case emr-rafaelvera-vpc.

Subnet:
no specific preference for a subnet, and the default subnet in any Availability Zone will be used.

3. 3. Create the EC2 instance



Tags – optional: Nothing to do here

1. Click on **CREATE** and you will be redirected to **Aws cloud9 console**

This Aws cloud9 console lists all the created environments. In this case shows main information from the recently environment instance created.

2. Click on **OPEN** to see how Cloud9 looks like

The screenshot shows the AWS Cloud9 'Environments' page. A green header bar displays a success message: 'Successfully created Emr rafaelvera Cloud9. To get the most out of your environment, see Best practices for using AWS Cloud9'. The main table lists one environment:

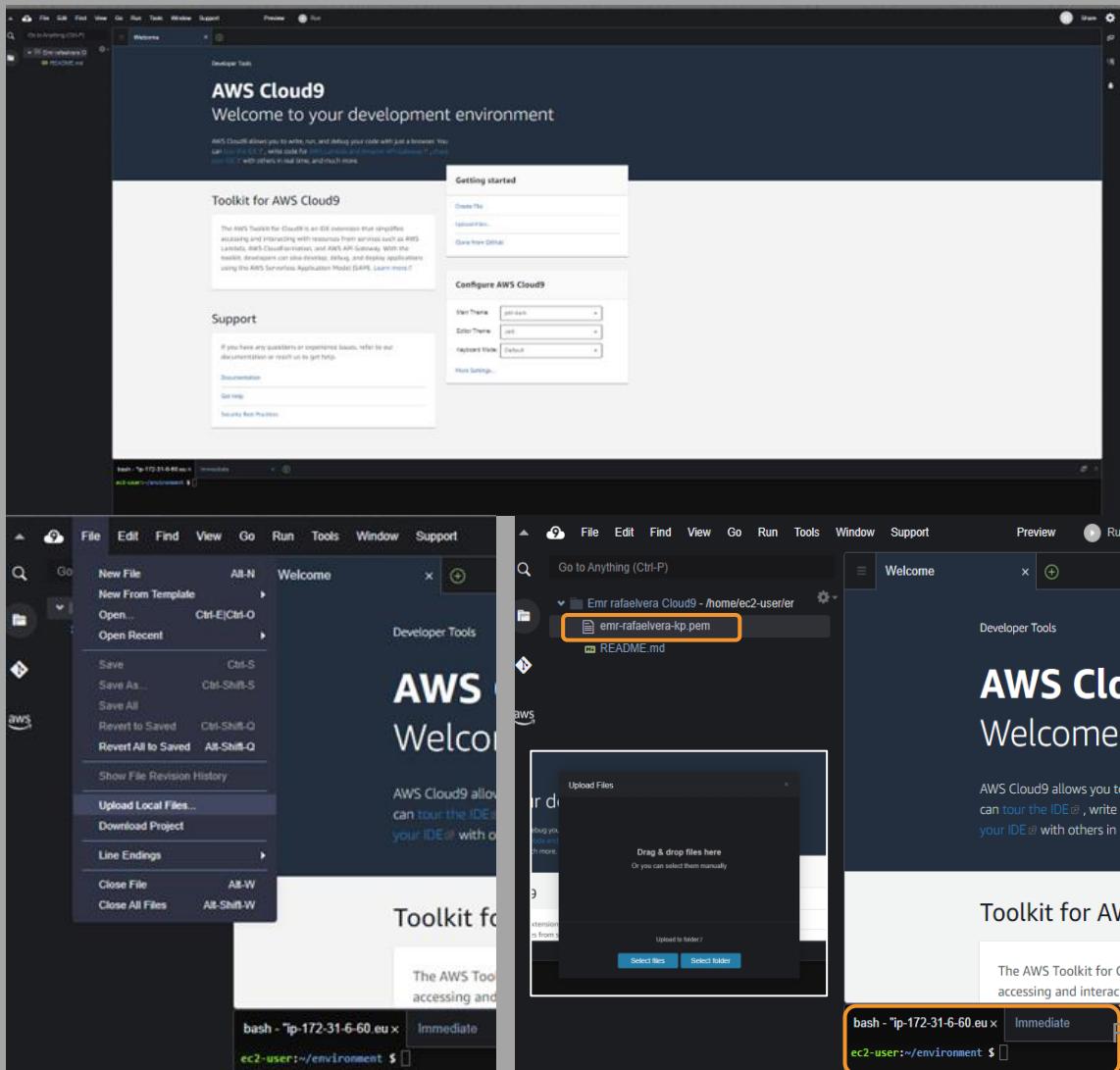
Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Emr rafaelvera Cloud9	Open 2	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:iam::851725340236:root

Aws Cloud9



January 2024

3. 4. Inside Cloud9



AWS Cloud9 integrated development environment (IDE): provides users with a powerful cloud-based environment for software development, including access to a pre-configured editor, terminal, and essential AWS tools and resources.

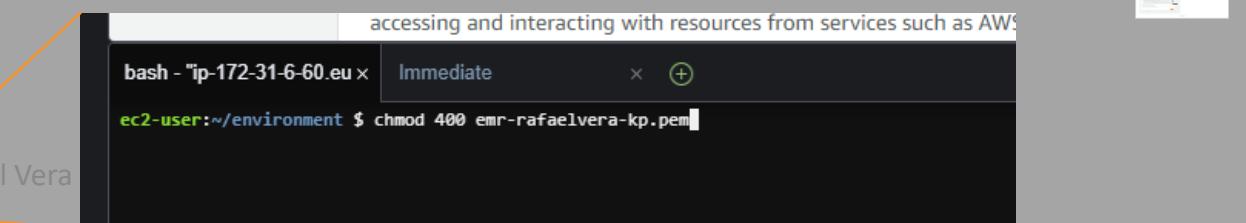
- Upload the Keypairs** generated during EMR Cluster. This .pem file was automatically downloaded locally as well. [Reminder here](#)
- Run this command:** change the file name.
`chmod 400 emr-rafaelvera-kp.pem`

This command changes the permissions to be read-only by the file's owner.

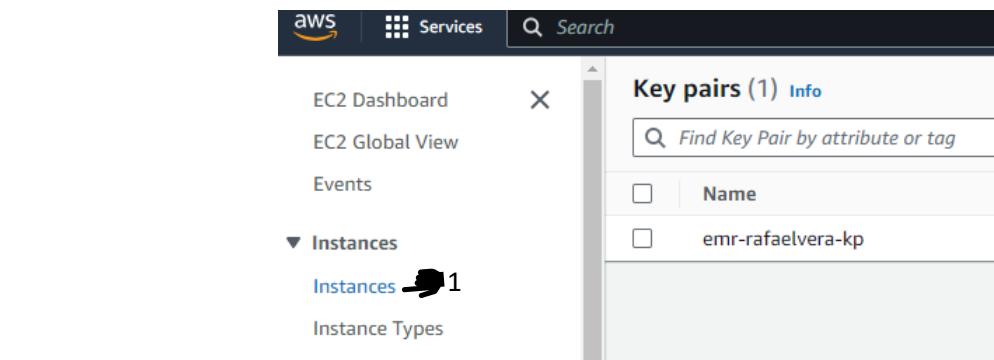
The 400 permission setting is used for SSH private key files ensuring they are kept secure and aren't accessible by other users.

Before any further, we need to:

- Create inbound Rules in the EMR Instance allowing cloud9 connection
- Get the SSH command from EMR to be able to connect with it.



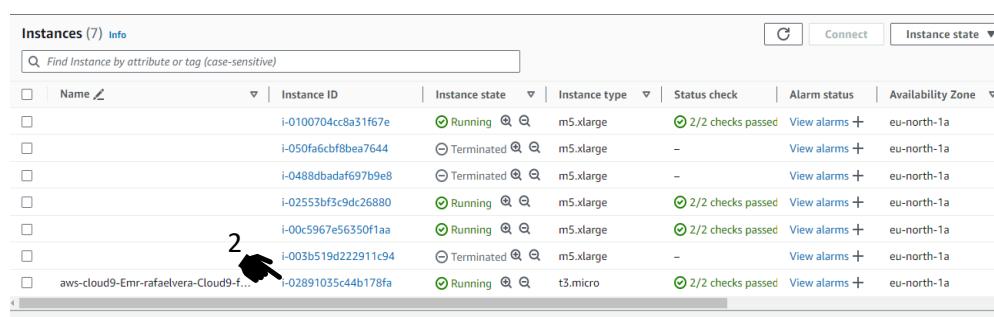
3.4.a.1. Get Aws Cloud9 instance IP



Key pairs (1) Info

Find Key Pair by attribute or tag

Name
emr-rafaelvera-kp

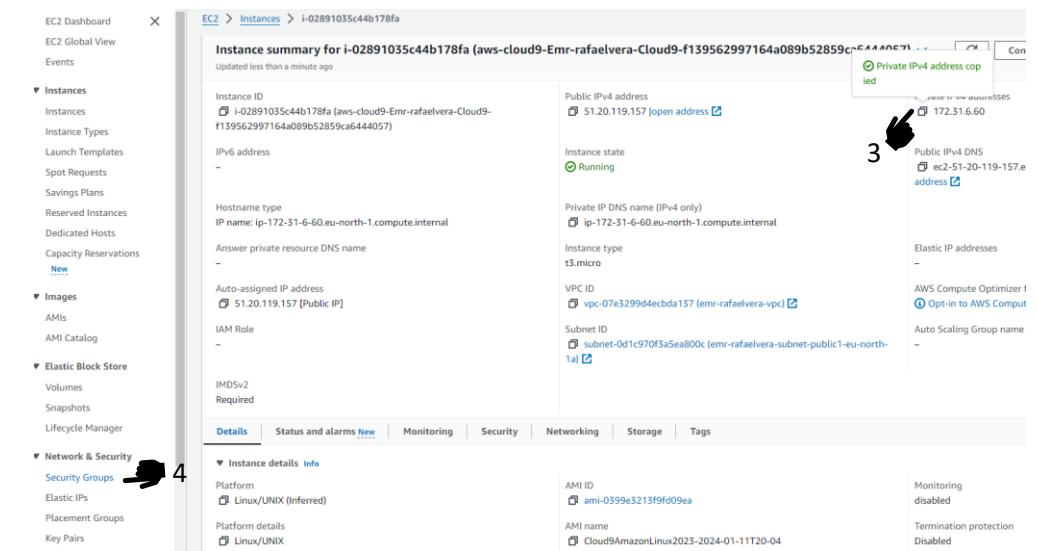


Instances (7) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
i-0100704cc8a31f67e	i-0100704cc8a31f67e	Running	m5.xlarge	2/2 checks passed	View alarms	eu-north-1a	ec2-16-16-56-82.eu-no...	16.16.56.82
i-050fa6cbf8bea7644	i-050fa6cbf8bea7644	Terminated	m5.xlarge	-	View alarms	eu-north-1a	-	-
i-0488badaf697b9e8	i-0488badaf697b9e8	Terminated	m5.xlarge	-	View alarms	eu-north-1a	-	-
i-02553bf3c9dc6880	i-02553bf3c9dc6880	Running	m5.xlarge	2/2 checks passed	View alarms	eu-north-1a	ec2-16-170-234-123.eu...	16.170.234.123
i-00c5967e56350f1aa	i-00c5967e56350f1aa	Running	m5.xlarge	2/2 checks passed	View alarms	eu-north-1a	ec2-51-21-130-58.eu-n...	51.21.130.58
i-003b519d222911c94	i-003b519d222911c94	Terminated	m5.xlarge	-	View alarms	eu-north-1a	-	-
aws-cloud9-Emr-rafaelvera-Cloud9-i...	i-02891035c44b178fa	Running	t3.micro	2/2 checks passed	View alarms	eu-north-1a	ec2-51-20-119-157.eu...	51.20.119.157

We need to get the **AWS Cloud9 instance IP** for having access from Cloud9 IDE to the EMR Cluster through an inbound rule.

1. Select any of the already opened tabs and click on “**instances**”.
2. Search in the list the aswcloud9 instance and click on its “**instance ID**”. You will be redirected to this instance console.
3. Copy the instance **IPv4**.
4. Click on “**Security groups**”



EC2 > Instances > i-02891035c44b178fa

Instance summary for i-02891035c44b178fa (aws-cloud9-Emr-rafaelvera-Cloud9-f139562997164a089b52859c...)

Updated less than a minute ago

Private IPv4 address
51.20.119.157 [open address]

Instance state Running

Public IPv4 DNS ip-172-31-6-60.eu-north-1.compute.internal

Public IPv4 address 51.20.119.157

IPv6 address -

Hostname type IPv4 only

Auto-assigned IP address 51.20.119.157 [Public IP]

Private IP DNS name (IPv4 only) ip-172-31-6-60.eu-north-1.compute.internal

Private IP address 172.31.6.60

Instance type t3.micro

VPC ID vpc-07e3299d4ecbda137 (emr-rafaelvera-vpc)

Subnet ID subnet-0d1c970f3a5ea800c (emr-rafaelvera-subnet-public1-eu-north-1a)

Elastic IP addresses -

AWS Compute Optimizer I Opt-in to AWS Comput

Auto Scaling Group name -

IMDSv2 Required

Details Status and alarms New Monitoring Security Networking Storage Tags

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Platform Linux/UNIX (Inferred)

Platform details Linux/UNIX

AMI ID ami-0399e3213f9fd09ea

AMI name Cloud9AmazonLinux2023-2024-01-11T20-04

Monitoring disabled

Termination protection Disabled

3.4.a.2. Security groups and Inbound rules

Security Groups (5) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-08be052ff0e3fb52e	default	vpc-0f16318a4593502a1	default VPC security group	851725340236
-	sg-04e37236b3c0875f0	default	vpc-07e3299d4ecbda137	default VPC security group	851725340236
aws-cloud9-Emr-ra...	sg-011fbe72462b4dc16	aws-cloud9-Emr-rafaelvera-Cloud9-f1...	vpc-07e3299d4ecbda137	Security group for AWS Cloud9 enviro...	851725340236
-	sg-00927a0d0accb25a3	ElasticMapReduce-master	vpc-07e3299d4ecbda137	Master group for Elastic MapReduce cr...	851725340236
-	sg-0c57b898391bbb558	ElasticMapReduce-slave	vpc-07e3299d4ecbda137	Slave group for Elastic MapReduce cre...	851725340236

BWS Services Search [Alt+S]

Operating system info

- Amazon Linux release 2023.3.20231211.4

Cluster logs info

- Archive log files to Amazon S3 Turned on
- Amazon S3 location s3://aws-logs-851725340236-eu-north-1/elasticsearch/
- Turn on encryption for logs Turned off

Cluster termination info

- Termination option Automatically terminate the cluster after idle time
- Idle time 1 hour
- Termination protection Turned on

Network and security info

Network

- Virtual Private Cloud (VPC) vpc-07e3299d4ecbda137
- Subnet ID and Availability Zone (AZ) subnet-0dc1c970f5a5ea800c eu-north-1a
- EC2 security groups (firewall)

 - Primary node EMR-managed security group sg-00927a0d0accb25a3
 - Core and task nodes EMR-managed security group sg-0c57b898391bbb558

Security configuration

- Security configuration none
- EC2 key pair emr-rafaelvera-kp

Permissions

- Service role for Amazon EMR emr-rafaelvera-role
- EC2 instance profile AmazonEMR-InstanceProfile-20240117T202223
- Auto-scaling role Not configured

Rafael Ve

Security Groups:

This console lists all the security groups active in the account. And can be matched with those implemented on the EMR cluster.

1. Go to the EMR Cluster console browser tab
2. Check the security groups implemented
3. Click on EMR Manager security group (**sg-00927a0d0accb25a3 – ElasticMapReduce-master**)
4. Click on “Edit inbound rules”

sg-00927a0d0accb25a3 - ElasticMapReduce-master

Details

Security group name	Security group ID	Description	VPC ID
ElasticMapReduce-master	sg-00927a0d0accb25a3	Master group for Elastic MapReduce created on 2024-01-05T00:27:59.899Z	vpc-07e3299d4ecbda137
Owner	851725340236	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry

Inbound rules (7)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-03f2236d3e953efbc	-	All TCP	TCP	0 - 65535	sg-0c57b898391bbb5...	-
-	sgr-0de0e3dcbb28200...	-	All UDP	UDP	0 - 65535	sg-0c57b898391bbb5...	-
-	sgr-091817e27e09a53f2	-	All TCP	TCP	0 - 65535	sg-00927a0d0accb25a...	-
-	sgr-0fb7b70014bec81670	-	All ICMP - IPv4	ICMP	All	sg-0c57b898391bbb5...	-
-	sgr-01e5bf93dbcffa663	-	All UDP	UDP	0 - 65535	sg-00927a0d0accb25a...	-
-	sgr-0ccdd393677e0d97	-	Custom TCP	TCP	8443	pl-dfra84d06...	-

4

3.4.a.3 Inbound rules

[Edit inbound rules](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-03f2236d3e953efbc	All TCP	TCP	0 - 65535	Custom	sg-0c57b898391bbb558 X
sgr-0de0e3dcbb2820061	All UDP	UDP	0 - 65535	Custom	sg-0c57b898391bbb558 X
sgr-091817e27e09a53f2	All TCP	TCP	0 - 65535	Custom	sg-00927a0d0accb25a3 X
sgr-0f7b70014bec81670	All ICMP - IPv4	ICMP	All	Custom	sg-0c57b898391bbb558 X
sgr-01e5b93dbcffa663	All UDP	UDP	0 - 65535	Custom	sg-00927a0d0accb25a3 X
sgr-0cccd3936777e0d97	Custom TCP	TCP	8443	Custom	pl-dfa84db6 X
sgr-09cf86e8c4dfdee45	All ICMP - IPv4	ICMP	All	Custom	cloud9 ip address X
-	SSH	TCP	22	Custom	172.31.6.60/32 X
-	Custom TCP	TCP	9443	Anywhere...	0.0.0.0/0 X

Add rule

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

5

[Cancel](#) [Preview changes](#) [Save rules](#)

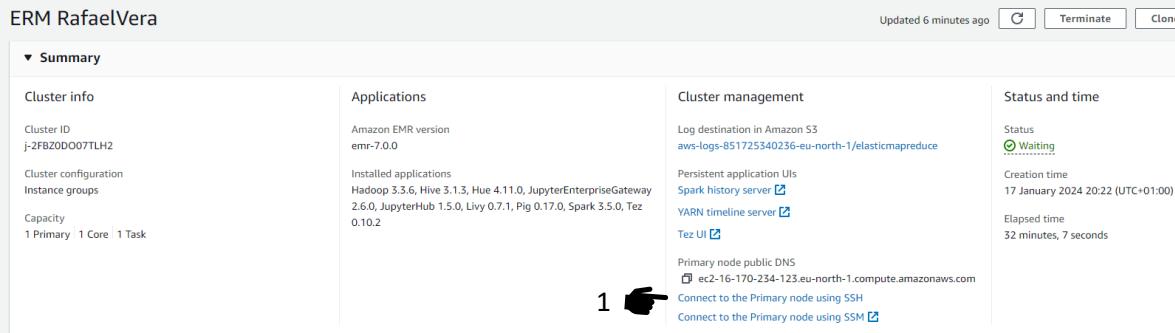
Edit inbound rules section of the AWS EC2 Security Group of sg-00927a0d0accb25a3 – ElasticMapReduce-master

This configuration defines the network access policies for the instances in this Security Group, specifying what kind of traffic is allowed to reach them.

1. Hit “**Add rule**” button
2. Add **SSH** rule as in the image. The IP is the **IPv4 copied before + /32** from the Cloud9 instance.
3. Hit “**Add rule**” button
4. Add Custom TCP rule as in the image. Allowing all addresses
5. Click on “**Save rules**”
6. Once done, lets continue with **section b** – get the ssh command from the EMR cluster to be pasted on Cloud9 IDE.



3.4.b.1 Get the SSH command



ERM RafaelVera

Summary

Cluster info

- Cluster ID: j-2FBZ0D007TLH2
- Cluster configuration: Instance groups
- Capacity: 1 Primary, 1 Core, 1 Task

Applications

- Amazon EMR version: emr-7.0.0
- Installed applications: Hadoop 3.3.6, Hive 3.1.3, Hue 4.11.0, JupyterEnterpriseGateway 2.6.0, JupyterHub 1.5.0, Livy 0.7.1, Pig 0.17.0, Spark 3.5.0, Tez 0.10.2

Cluster management

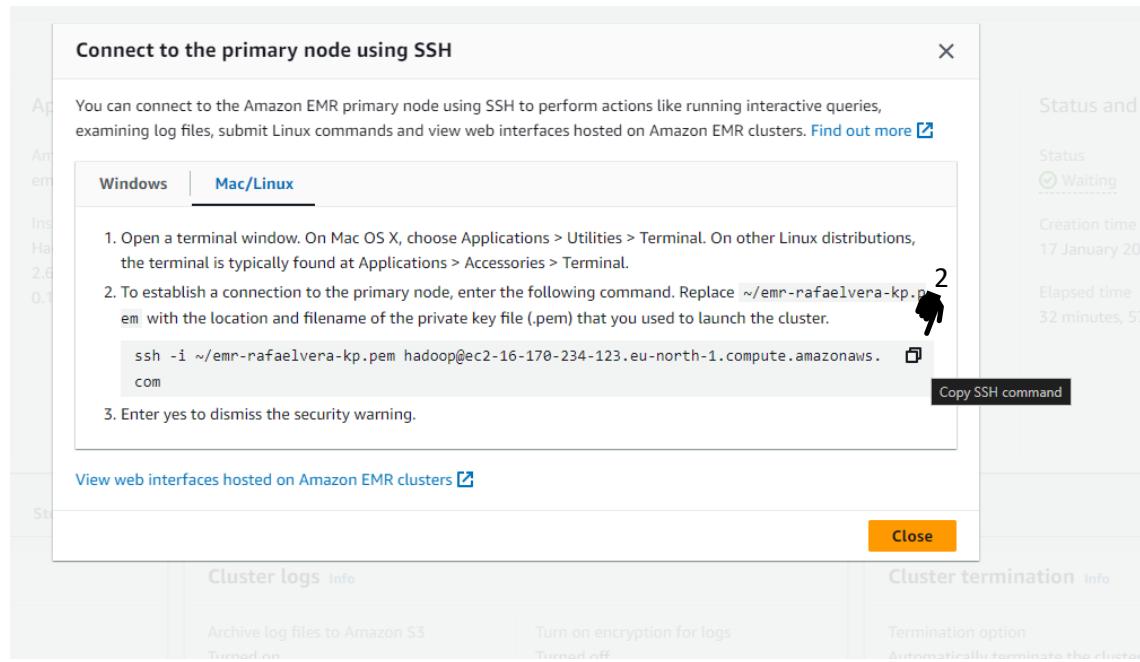
- Log destination in Amazon S3: aws-logs-851725340236-eu-north-1/elasticmapreduce
- Persistent application UIs: Spark history server, YARN timeline server, Tez UI

Status and time

- Status: Waiting
- Creation time: 17 January 2024 20:22 (UTC+01:00)
- Elapsed time: 32 minutes, 7 seconds

Primary node public DNS: ec2-16-170-234-123.eu-north-1.compute.amazonaws.com

Buttons: Connect to the Primary node using SSH, Connect to the Primary node using SSM



Connect to the primary node using SSH

You can connect to the Amazon EMR primary node using SSH to perform actions like running interactive queries, examining log files, submit Linux commands and view web interfaces hosted on Amazon EMR clusters. [Find out more](#)

Windows | Mac/Linux

- Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, the terminal is typically found at Applications > Accessories > Terminal.
- To establish a connection to the primary node, enter the following command. Replace `~/emr-rafaelvera-kp.pem` with the location and filename of the private key file (.pem) that you used to launch the cluster.
`ssh -i ~/emr-rafaelvera-kp.pem hadoop@ec2-16-170-234-123.eu-north-1.compute.amazonaws.com`
- Enter yes to dismiss the security warning.

View web interfaces hosted on Amazon EMR clusters

Cluster logs | Cluster termination

Archive log files to Amazon S3 | Turn on encryption for logs | Termination option

BACK to the EMR Cluster Summary Console:

- Click on “**Connect to the primary node using SSH**” and a popups will be shown
- Copy the ssh command**

With this ssh command copied, now **you must go to AWS Cloud9 IDE again**

3.5. ssh EMR cluster connection

```
bash - "ip-172-31-6-60.eu" x Immediate x +  
ec2-user:~/environment $ chmod 400 emr-rafaelvera-kp.pem  
ec2-user:~/environment $ ssh -i emr-rafaelvera-kp.pem hadoop@ec2-16-170-234-123.eu-north-1.compute.amazonaws.com
```

```
hadoop@ip-172-31-13-14: ~ Immediate x +  
ec2-user:~/environment $ ssh -i emr-rafaelvera-kp.pem hadoop@ec2-16-170-234-123.eu-north-1.compute.amazonaws.com  
The authenticity of host 'ec2-16-170-234-123.eu-north-1.compute.amazonaws.com (172.31.13.142)' can't be established.  
ED25519 key fingerprint is SHA256:vnc3rpvBd3opbYeigxjl1WvdGj50LAL4+5n/4F5x5TPg.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-16-170-234-123.eu-north-1.compute.amazonaws.com' (ED25519) to the list of known hosts.  
  
A newer release of "Amazon Linux" is available.  
Version 2023.3.20231218:  
Version 2023.3.20240108:  
Run "/usr/bin/dnf check-release-update" for full release and version update info  
, #  
~\###_ Amazon Linux 2023  
~ \####|  
~~ \##|  
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023  
~~ V-' '-->  
~~ /  
~~ /  
~~ /  
~~ /  
~~ /m'  
Last login: Wed Jan 17 19:49:16 2024  
  
EEEEEEEEE EEEEEE M:----M M:----M R:----RRRRRRRRRRRRRR  
E:-----: E:----M:----M M:----M R:----RRRRRRRRRRRRRR  
EE:-----: E:----M:----M M:----M R:----RRRRRRRRRRRRRR  
E:----E: EEEE M:----M M:----M R:----RRRRRRRRRRRRRR  
E:----E: M:----M:----M M:----M R:----RRRRRRRRRRRRRR  
E:----E: EEEEEE M:----M M:----M R:----RRRRRRRRRRRRRR  
E:----E: M:----M M:----M M:----M R:----RRRRRRRRRRRRRR  
E:----E: M:----M M:----M M:----M R:----RRRRRRRRRRRRRR  
E:----E: M:----M M:----M R:----RRRRRRRRRRRRRRRRRRRRRRRR  
E:----E: EEEE M:----M M:----M R:----RRRRRRRRRRRRRRRRRRRRRR  
E:----E: EEEEEE M:----M M:----M R:----RRRRRRRRRRRRRRRRRRRRRR  
E:----E: EEEEEE M:----M M:----M R:----RRRRRRRRRRRRRRRRRRRRRR  
E:----E: EEEEEE M:----M M:----M R:----RRRRRRRRRRRRRRRRRRRRRR  
EEEEEEEEE EEEEEE M:----M M:----M R:----RRRRRRRRRRRRRRRRRRRRRR  
[hadoop@ip-172-31-13-14 ~]$
```

AWS Cloud9 integrated development environment (IDE):

Back here again we continue with the next command which ables to connect cloud9 IDE with the EMR cluster instance.

1. Paste de copied ssh command
2. Watch how the connection success

We continue next giving the spark job command to trigger the distributed processing. But we are not yet done, first we need to complete a couple steps.

3.6. Submitting Spark job command

```

  \#! /usr/bin/python
  https://aws.amazon.com/linux/amazon-linux-2023

  /m/
  Last login: Wed Jan 17 19:49:16 2024

  EEEEEEEEEE M::::::M      M::::::M RRRRRRRRRRRRRR
  E:::::::::::E M:::::M     M:::::M R:::::::::::R
  EE:::::EEEEE:::E M:::::M     M:::::M R:::::RRRRRR:::R
  E:::E     EEEE M:::::M     M:::::M RR:::::R     R:::::R
  E::::E     M:::::M:::M     M:::M::::M R:::R     R:::::R
  E:::::EEEEE EEE M:::::M M:::M:::M M:::::M R:::::RRRRRR:::R
  E:::::::::::E M:::::M M:::M:::M M:::::M R:::::::::::RR
  E:::::EEEEE EEE M:::::M M:::::M M:::::M R:::::RRRRRR:::R
  E::::E     M:::::M M:::M M:::::M R:::R     R:::::R
  E::::E     EEEE M:::::M     M:::::M R:::R     R:::::R
  EE:::::EEEEE:::E M:::::M     M:::::M R:::R     R:::::R
  E:::::::::::E M:::::M     M:::::M RR:::::R     R:::::R
  EEEEEEEEEE M::::::M      M::::::M RRRRRRRR     RRRRRR

[hadoop@ip-172-31-13-142 ~]$ nano spark-etl.scala
[hadoop@ip-172-31-13-142 ~]$ nano spark-etl.py
[hadoop@ip-172-31-13-142 ~]$ nano spark-etl.py
[hadoop@ip-172-31-13-142 ~]$ spark-submit spark-etl.py

```

Now we need to submit the spark job command with this structure:

spark-submit {Script.py} {S3 input folder / data} {S3 output folder/}

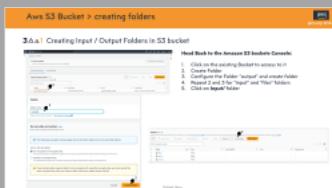
As you can see, we need this 3 components. So go ahead:

1. Run **nano spark-etl.py** to create the file within the EMR Instance
2. Paste the python code inside the nano window. (you can find it in my Github repository)
3. Save and close nano.
4. We have one of the three command components:

spark-submit spark-etl.py {S3 input folder / data} {S3 output folder/}

We need to create the folders for our S3 Bucket instance needed to submit spark job and grant EMR access to the S3 bucket.

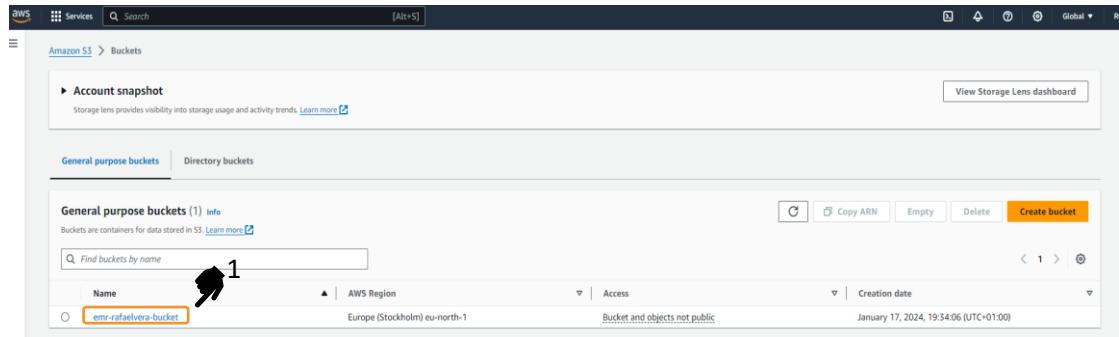
5. Create folders in S3 bucket (a)



6. Grant EMR access to S3 bucket (b)



3.6.a.1 Creating Input / Output Folders in S3 bucket



Amazon S3 > Buckets

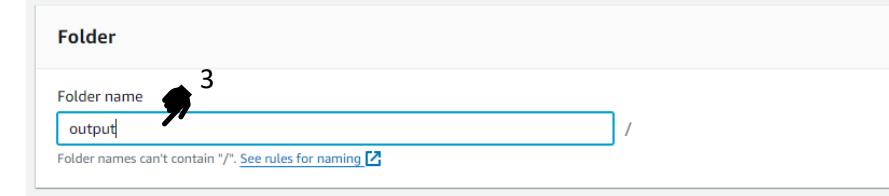
General purpose buckets (1) Info

Buckets are containers for data stored in S3. Learn more [\[?\]](#)

Find buckets by name 1

Name	AWS Region	Access	Creation date
emr-rafaelvera-bucket	Europe (Stockholm) eu-north-1	Bucket and objects not public	January 17, 2024, 19:34:06 (UTC+01:00)

Create bucket



Folder

Folder name 3
output

Folder names can't contain "/". See rules for naming [\[?\]](#)



Server-side encryption [Info](#)

Server-side encryption protects data at rest.

The following encryption settings apply only to the folder object and not to sub-folder objects.

Do not specify an encryption key
The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.

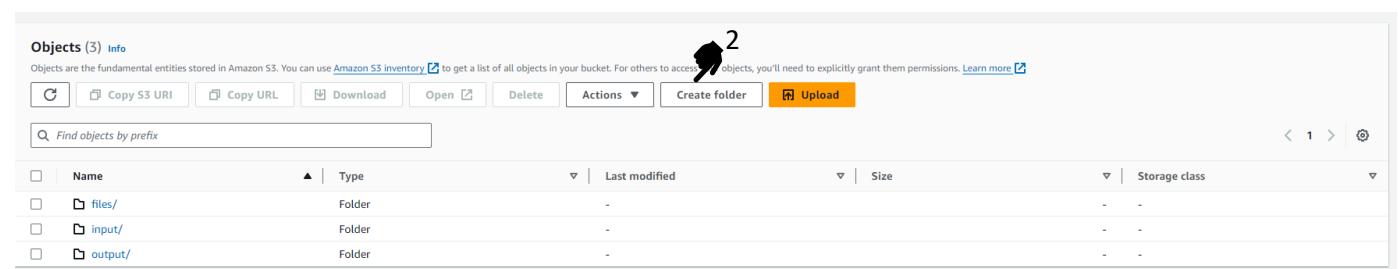
Specify an encryption key
The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.

If your bucket policy requires objects to be encrypted with a specific encryption key, you must specify the same encryption key when you create a folder. Otherwise, folder creation will fail. 3

Create folder

Head Back to the Amazon S3 buckets Console:

1. Click on the existing Bucket to access to it
2. Create Folder
3. Configure the Folder “output” and create folder
4. Repeat 2 and 3 for “input” and “files” folders
5. Click on **input/** folder



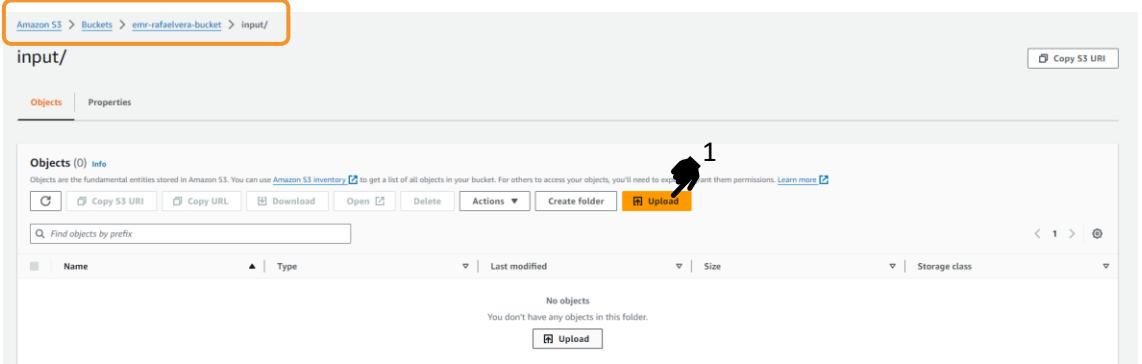
Objects (3) Info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access objects, you'll need to explicitly grant them permissions. Learn more [\[?\]](#)

Name	Type	Last modified	Size	Storage class
files/	Folder	-	-	-
input/	Folder	-	-	-
output/	Folder	-	-	-

Actions [Upload](#)

3.6.a.2 Uploading dataset to input folder



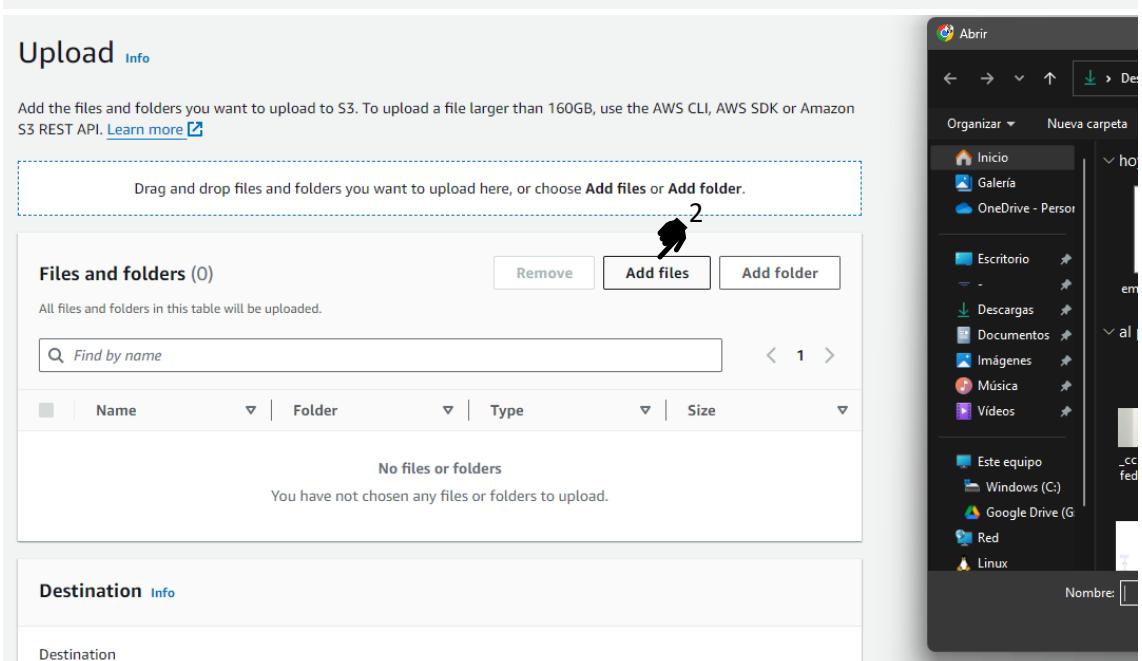
Amazon S3 > Buckets > emr-rafaelvera-bucket > input/

input/

Objects (0) info

No objects

Upload



Upload info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files or Add folder.

Files and folders (0)

All files and folders in this table will be uploaded.

Add files

Add folder

Find by name

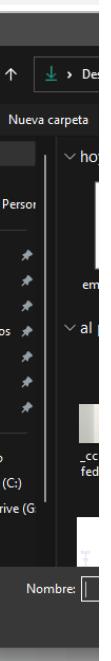
Name Folder Type Size

No files or folders

You have not chosen any files or folders to upload.

Destination info

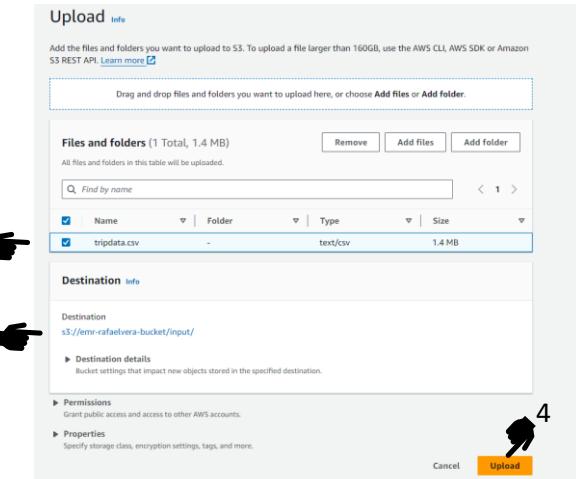
Destination



Amazon S3 > Buckets > emr-rafaelvera-bucket > input/

In this bucket folder we need to upload the dataset (.csv) to be processed

1. Click on **Upload**
2. Click on **Add files** and select the .csv from your PC. (.csv available in my Github repository)
3. See the file to be uploaded in the list and destination are correct.
4. Click on **Upload**



Upload info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files or Add folder.

Files and folders (1 Total, 1.4 MB)

All files and folders in this table will be uploaded.

Find by name

Name Folder Type Size

tripdata.csv - text/csv 1.4 MB

Destination info

Destination s3://emr-rafaelvera-bucket/input/

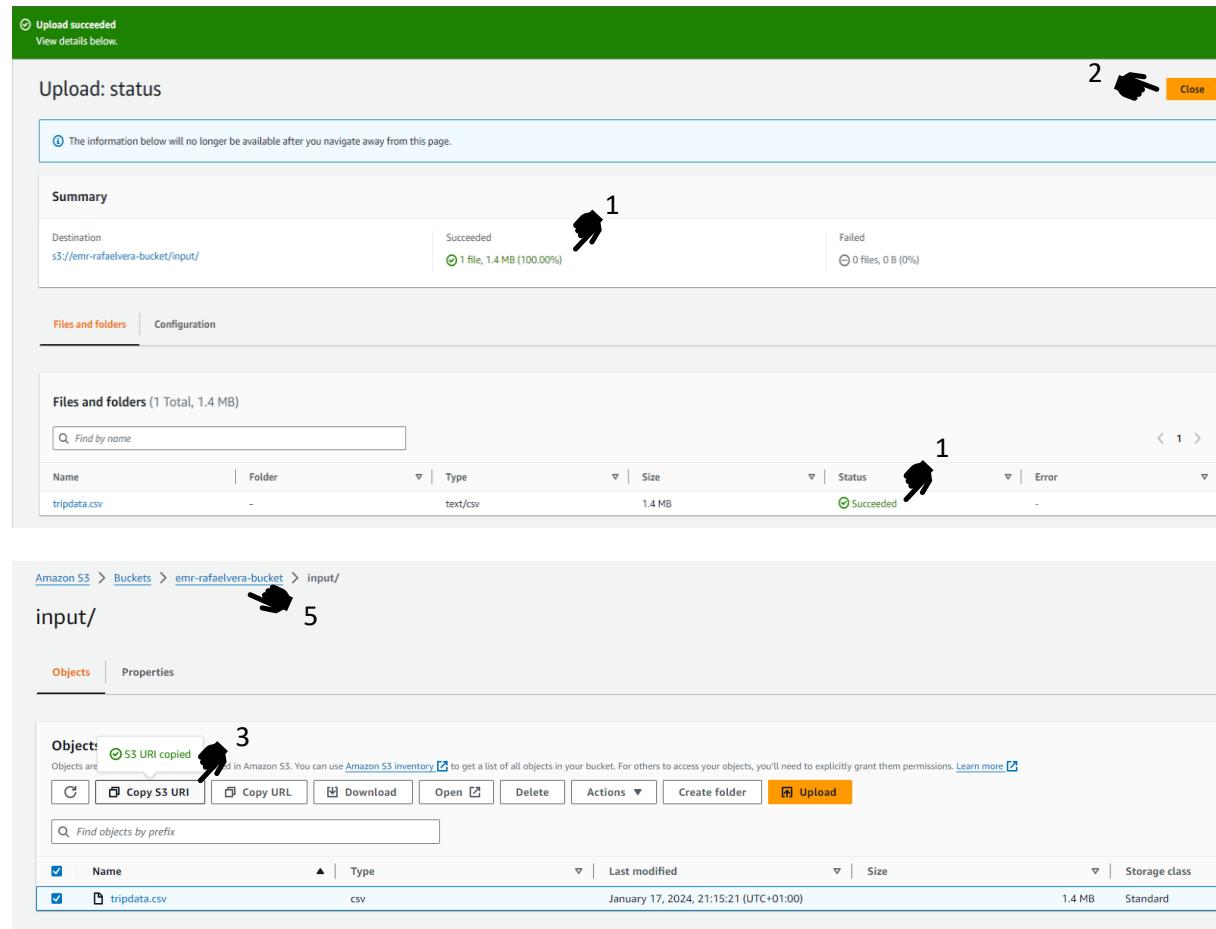
Destination details Bucket settings that impact new objects stored in the specified destination.

Permissions Grant public access and access to other AWS accounts.

Properties Specify storage class, encryption settings, tags, and more.

Cancel Upload

3.6.a.3 Successfully uploaded dataset to input folder



The screenshot shows the AWS S3 console interface. At the top, a green banner indicates "Upload succeeded". Below it, the "Upload: status" section shows a summary of the upload: "Destination s3://emr-rafaelvera-bucket/input/" with "Succeeded" status and "1 file, 1.4 MB (100.00%)". A "Close" button is highlighted with a hand icon labeled "2". The "Files and folders" tab is selected, showing a table with one row: "tripdata.csv" (text/csv, 1.4 MB, Status: Succeeded). The URL "s3://emr-rafaelvera-bucket/input/tripdata.csv" is copied to the clipboard. The "Objects" tab is also visible at the bottom.

Amazon S3 > Buckets > emr-rafaelvera-bucket > input/

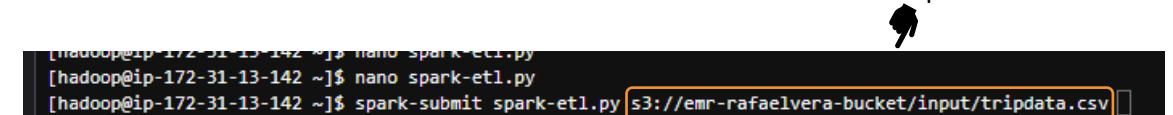
1. Confirm the tripdata.csv was successfully uploaded
2. Hit on “Close”
3. Copy S3 URI

Remember that we are here to load the dataset to be processed by the python script launched with the spark job.
And we still building the spark job command.

4. Go shortly to Cloud9 IDE and paste the S3 URI as the second command component.

```
spark-submit spark-etl.py s3://emr-rafaelvera-
bucket/input/tripdata.csv {S3 output folder/}
```

5. Go back to this Bucket main console



```
[hadoop@ip-172-31-13-142 ~]$ nano spark-etl.py
[hadoop@ip-172-31-13-142 ~]$ nano spark-etl.py
[hadoop@ip-172-31-13-142 ~]$ spark-submit spark-etl.py s3://emr-rafaelvera-bucket/input/tripdata.csv
```

3.6.a.4 Getting S3 URI from Output folder

Amazon S3 > Buckets > emr-rafaelvera-bucket

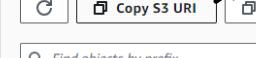
emr-rafaelvera-bucket [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects: [S3 URI copied](#)        

[Find objects by prefix](#)

Name	Type	Last modified	Size
files/	Folder	-	-
input/	Folder	-	-
output/	Folder	-	-

 **1**

 **2**

```
[hadoop@ip-172-31-13-142 ~]$ nano spark-etl.py
[hadoop@ip-172-31-13-142 ~]$ spark-submit spark-etl.py s3://emr-rafaelvera-bucket/input/tripdata.csv s3://emr-rafaelvera-bucket/output/
```

 **3**

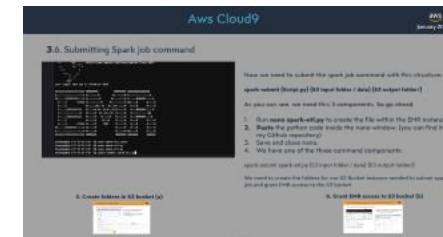
Amazon S3 > Buckets > emr-rafaelvera-bucket

1. **Select output/ folder**
2. **Copy S3 URI**
3. **Go shortly to Cloud9 IDE and paste the S3 URI as the third command component.**

Remember that we are here to load the dataset to be processed by the python script launched with the spark job.
And we still building the spark job command.

```
spark-submit spark-etl.py s3://emr-rafaelvera-
bucket/input/tripdata.csv s3://emr-rafaelvera-bucket/output/
```

4. We have the **spark submit command complete. Not run it yet**
5. **Copy the command handy. We'll add something to it later**
6. **Before running this command**, we need to grant EMR access to this S3 Bucket folders. **Follow Section b** 



3.6.b.1 Granting EMR Access to S3 Bucket folders

1

```
[hadoop@ip-172-31-13-142 ~]$ aws s3 ls s3://emr-rafaelvera-bucket
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied
[hadoop@ip-172-31-13-142 ~]$
```

2

Instances (1/4) Info								
Find Instance by attribute or tag (case-sensitive) Connect Instance state Actions Launch instances								
status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
aws	+ eu-north-1a	ec2-16-16-56-82.eu-no...	16.16.56.82	-	-	disabled	ElasticMapReduce-slave	emr-rafaelver...
aws	+ eu-north-1a	ec2-16-170-234-123.eu...	16.170.234.123	-	-	disabled	ElasticMapReduce-master	emr-rafaelver...
aws	+ eu-north-1a	ec2-51-21-130-58.eu-n...	51.21.130.58	-	-	disabled	ElasticMapReduce-slave	emr-rafaelver...
aws	+ eu-north-1a	ec2-51-20-119-157.eu...	51.20.119.157	-	-	disabled	aws-cloud9-Emr-rafaelv...	-

Instances (1/4) Info	
Find Instance by attribute or tag (case-sensitive)	
Name	Instance ID
<input type="checkbox"/>	i-0100704cc8a31f67e
<input checked="" type="checkbox"/>	i-02553bf3c9dc26880
<input type="checkbox"/>	i-00c5967e56350f1aa
<input type="checkbox"/>	aws-cloud9-Emr-rafaelvera-Cloud9-f... i-02891035c44b178fa

1. Go shortly to Cloud9 IDE and try to list the content of the S3 Bucket: (EMR is not yet granted to access S3 Bucket)

`aws s3 ls s3://emr-rafaelvera-bucket`

2. Select your Browser tab in which is the **EC2 Instances console**.
3. Open the “ElasticMapReduce-master” instance by clicking on the Instance ID

4. Inside the instance: click on **IAM Role**

3

4

Instance summary for i-02553bf3c9dc26880 Info			
Updated less than a minute ago			
Instance ID	i-02553bf3c9dc26880	Public IPv4 address	16.170.234.123 open address
IPv6 address	-	Private IPv4 addresses	172.31.13.142
Instance state	Running	Public IPv4 DNS	ec2-16-170-234-123.eu-north-1.compute.amazonaws.com open address
Hostname type	IP name: ip-172-31-13-142.eu-north-1.compute.internal	Private IP DNS name (IPv4 only)	ip-172-31-13-142.eu-north-1.compute.internal
Answer private resource DNS name	-	Instance type	m5.xlarge
Auto-assigned IP address	16.170.234.123 [Public IP]	VPC ID	vpc-07e3299d4ecbda157 (emr-rafaelvera-vpc)
IAM Role	AmazonEMR-InstanceProfile-20240117T202223	Subnet ID	subnet-0d1c970f3a5ea800c (emr-rafaelvera-subnet-public1-eu-north-1a)
IMDsv2	Required	Auto Scaling Group name	-

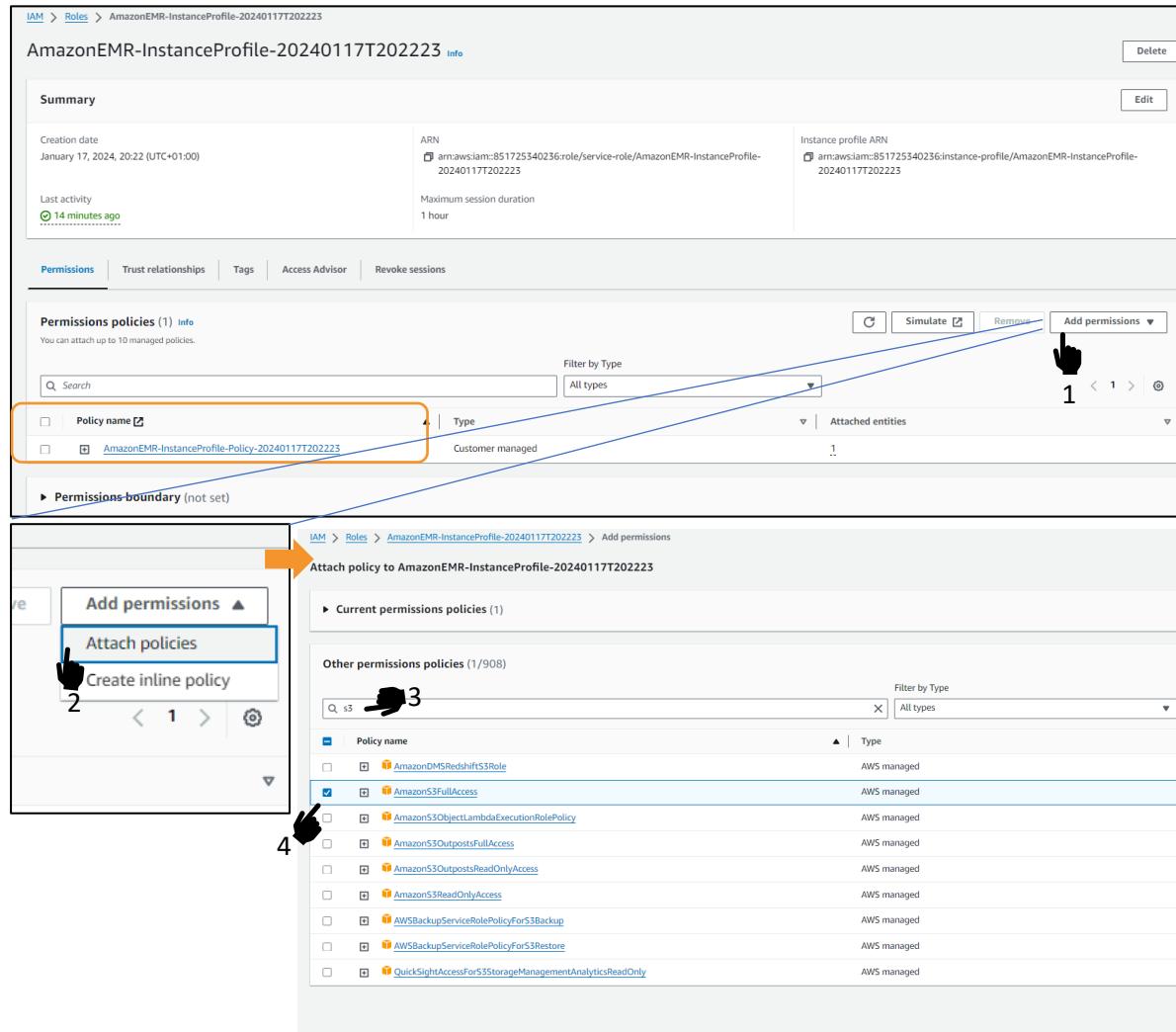
[Details](#) [Status and alarms New](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

[Instance details info](#)

Platform: Linux/UNIX (Inferred)
Platform details: Linux/UNIX

AMI ID: ami-023d6ff51a47013a
AMI name: emr-7.0-x86_64-2023_3_20231211_4-Hadoop_Hive_Pig_Spark-2023-Enabled

3.6.b.2 Granting EMR Access to S3 Bucket folders



IAM > Roles > Amazon EMR-instanceProfile-20240117T202223

We are in the summary page of IAM role for our EMR instance.

Under the **Permissions tab**, there is one customer-managed policy attached to this role, with the policy name being **AmazonEMR-InstanceProfile-Policy-20240117T202223**.

1. Click on “Add Permissions”
2. Click on “Attach policies”
3. Search S3
4. Select AmazonS3FullAccess (*not a great practice)
5. Hit “Add permissions”

6. Go to Cloud9 and run the S3 content listing command:

```
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: ACCESS DENIED
[hadoop@ip-172-31-13-142 ~]$ aws s3 ls s3://emr-rafaelvera-bucket
PRE files/
PRE input/
PRE output/
[hadoop@ip-172-31-13-142 ~]$
```

3.6.1 Submitting Spark job command

```

PRE output/
[hadoop@ip-172-31-13-142 ~]$ spark-submit spark-etl.py s3://emr-rafaelvera-bucket/input/tripdata.csv s3://emr-rafaelvera-bucket/output/
Jan 17, 2024 8:34:53 PM org.apache.spark.launcher.Log4jHotPatchOption static/avalongentOption
WARNING: spark.log4jHotPatch.enabled is set to true, but /usr/share/log4j-cve-2021-44228-hotpatch/jdk17/Log4jHotPatch.jar does not exist at the configured location

3
24/01/17 20:34:56 INFO SparkContext: Running Spark version 3.5.0-amzn-0
24/01/17 20:34:56 INFO SparkContext: OS info Linux, 6.1.66-91.160.amzn2023.x86_64, amd64
24/01/17 20:34:56 INFO SparkContext: Java version 17.0.9
24/01/17 20:34:56 INFO ResourceUtils: ****
24/01/17 20:34:56 INFO ResourceUtils: No custom resources configured for spark.driver.
24/01/17 20:34:56 INFO ResourceUtils: ****
24/01/17 20:34:56 INFO SparkContext: Submitted application: SparkETL
24/01/17 20:34:56 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 4, script: , vendor: , memory -> name: memory, amount: 9486, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpu -> name: cpus, amount: 1.0)
24/01/17 20:34:56 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
24/01/17 20:34:56 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/01/17 20:34:56 INFO SecurityManager: Changing view acls to: hadoop
24/01/17 20:34:56 INFO SecurityManager: Changing modify acls to: hadoop
24/01/17 20:34:56 INFO SecurityManager: Changing view acls groups to:
24/01/17 20:34:56 INFO SecurityManager: SecurityManager: authentication disabled; vi acls disabled; users with view permissions: hadoop; groups with view permissions: EMPTY; users with modify permissions: hadoop; groups with modify permissions: EMPTY
24/01/17 20:34:56 INFO Utils: Successfully started service 'sparkDriver' on port 35321.
24/01/17 20:34:56 INFO SparkEnv: Registering MapOutputTracker
24/01/17 20:34:56 INFO SparkEnv: Registering BlockManagerMaster
24/01/17 20:34:56 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/01/17 20:34:56 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/01/17 20:34:56 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/01/17 20:34:56 INFO DiskBlockManager: Created local directory at /mnt/tmp/blockmgr-559dc7a-61ef-41f1-b9cf-c59899bb8e1c
24/01/17 20:34:56 INFO MemoryStore: MemoryStore started with capacity 1048.8 MiB

24/01/17 20:35:32 INFO YarnScheduler: Removed TaskSet 2.0, whose tasks have all completed, from pool
24/01/17 20:35:32 INFO DAGScheduler: ResultStage 2 (showString at NativeMethodAccessorImpl.java:0) finished in 0.377 s
24/01/17 20:35:32 INFO DAGScheduler: Job 2 is finished. Cancelling potential speculative or zombie tasks for this job
24/01/17 20:35:32 INFO YarnScheduler: Killing all running tasks in stage 2: Stage finished
24/01/17 20:35:32 INFO DAGScheduler: Job 2 finished: showString at NativeMethodAccessorImpl.java:0, took 0.381467 s
24/01/17 20:35:32 INFO CodeGenerator: Code generated in 39.39558 ms

+-----+
| vendorID|lpep_pickup_datetime|lpep_dropoff_datetime|store_and_fwd_flag|RatecodeID|PUlocationID|DOLocationID|passenger_count|trip_distance|fare_amount|extra|mta_tax|tip_amount|tolls_amount|ehail_fee|improvement_surcharge|total_amount|payment_type|trip_type|current_date|
+-----+
| 2| 1/1/17 0:01| 1/1/17 0:11| N| 1| 42| 166| 1| 1.71| 9.0| 0.0| 0.5| 0.0| 0.0| NULL| 0.3| | |
| 9.8| 2| 1| 2024-01-17 20:35:...| 1/1/17 0:09| N| 1| 75| 74| 1| 1.44| 6.5| 0.5| 0.5| 0.0| 0.0| NULL| 0.3|
| 2| 1/1/17 0:08| 1| 2024-01-17 20:35:...| 1/1/17 0:12| N| 1| 82| 70| 5| 3.45| 12.0| 0.5| 0.5| 2.66| 0.0| NULL| 0.3|
| 15.96| 2| 1| 2024-01-17 20:35:...| 1/1/17 0:14| N| 1| 255| 232| 1| 2.11| 10.5| 0.5| 0.5| 0.0| 0.0| NULL| 0.3|
| 11.8| 2| 1| 2024-01-17 20:35:...| 1/1/17 0:15| N| 1| 166| 239| 1| 2.76| 11.5| 0.5| 0.5| 0.0| 0.0| NULL| 0.3|
| 12.8| 2| 1| 2024-01-17 20:35:...| 1/1/17 0:13| N| 1| 178| 226| 1| 4.41| 15.0| 0.5| 0.5| 0.0| 0.0| NULL| 0.3|
+-----+
24/01/17 20:35:33 INFO YarnScheduler: Killing all running tasks in stage 5: Stage finished
24/01/17 20:35:33 INFO DAGScheduler: Job 4 finished: count at NativeMethodAccessorImpl.java:0, took 0.140432 s
Total number of records: 20000
24/01/17 20:35:33 INFO FileSourceStrategy: Pushed Filters:
24/01/17 20:35:33 INFO FileSourceStrategy: Post-Scan Filters:
Traceback (most recent call last):
  File ".../home/hadoop/spark-etl.py", line 29, in <module>

```

1

2

3

Create folders in S3 bucket (a) > DONE

Now EMR is able to execute the job pointing to the S3 bucket folders passed into the spark-job command.

Grant EMR access to S3 bucket (b) > DONE

EMR is able to read and write files within the S3 Buckets

1. Run the spark job. Add a subfolder into “output” to avoid overwriting issues (I'd mistaken by the first try)

spark-submit spark-etl.py s3://emr-rafaelvera-bucket/input/tripdata.csv s3://emr-rafaelvera-bucket/output/spark

2. The distributed data processing started.

3. The processed dataset is printed

4. Dataset schema showing the new added column

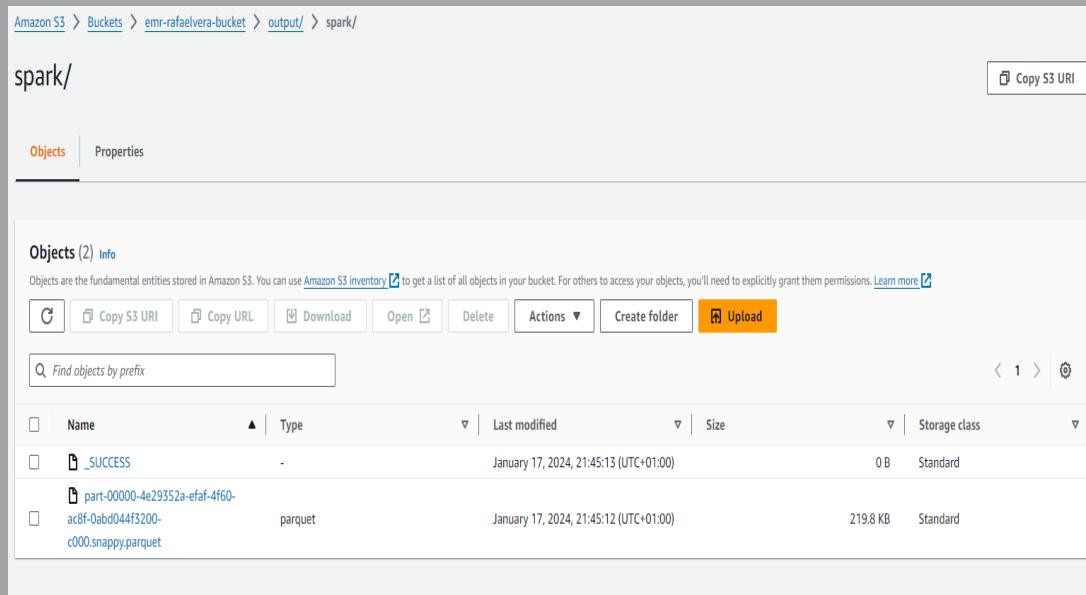
Our spark-etl.py is a simple example of transformation, consisting in adding a new column “Current_date” (timestamp)

```

24/01/17 20:35:31 INFO DAGScheduler: Job 1 finished: csv at Nat
root
|-- VendorID: integer (nullable = true)
|-- lpep_pickup_datetime: string (nullable = true)
|-- lpep_dropoff_datetime: string (nullable = true)
|-- store_and_fwd_flag: string (nullable = true)
|-- RatecodeID: integer (nullable = true)
|-- PUlocationID: integer (nullable = true)
|-- DOLocationID: integer (nullable = true)
|-- passenger_count: integer (nullable = true)
|-- trip_distance: double (nullable = true)
|-- fare_amount: double (nullable = true)
|-- extra: double (nullable = true)
|-- mta_tax: double (nullable = true)
|-- tip_amount: double (nullable = true)
|-- tolls_amount: double (nullable = true)
|-- ehail_fee: string (nullable = true)
|-- improvement_surcharge: double (nullable = true)
|-- total_amount: double (nullable = true)
|-- payment_type: integer (nullable = true)
|-- trip_type: integer (nullable = true)
-- current_date: timestamp (nullable = false)

```

3.6.2 Check Submitted Spark job output



Name	Type	Last modified	Size	Storage class
_SUCCESS	-	January 17, 2024, 21:45:13 (UTC+01:00)	0 B	Standard
part-00000-4e29352a-efaf-4f60-a8cf-00bad443f200-c000.snappy.parquet	parquet	January 17, 2024, 21:45:12 (UTC+01:00)	219.8 KB	Standard

Amazon S3 > Buckets > emr-rafaelvera-bucket > output/ > spark/

1. Go to the S3 Bucket folder “spark2” and see the outputted files after running the Spark Job

Two objects listed:

SUCCESS - Marker file that indicates that the Spark job has been completed successfully. The file has a size of 0 bytes, which is typical for a _SUCCESS file, as it is empty and serves only as an indicator.

part-00000-4e29352a-efaf-4f60-a8cf-00bad443f200-c000.snappy.parquet - Data file in Parquet format that has been compressed using the Snappy algorithm.

Parquet is a columnar storage file format that is commonly used in the Hadoop ecosystem, and **Snappy** is a compression and decompression library that aims for very high speeds and reasonable compression. The file size is 219.8 KB.

Spark & Hadoop workout results

4. Check Submitted Spark job output

The screenshot shows the Amazon EMR cluster summary page for 'EMR RafaelVera'. It includes sections for Cluster info, Applications, Cluster management, and Status and time. Applications listed include Amazon EMR version emr-7.0.0, Installed applications (Hadoop 3.3.6, Hive 3.1.3, Hue 4.11.0, JupyterEnterpriseGateway 2.6.0, JupyterHub 1.5.0, Livy 0.7.1, Pig 0.17.0, Spark 3.5.0, Tez 0.10.2), and Primary node public DNS (ec2-16-170-234-123.eu-north-1.compute.amazonaws.com). The status is 'Waiting'.

2. Check results in Spark console (a)

The screenshot shows the Apache Spark History Server web UI. It displays two completed Spark applications. A note states: "The first application did not save the output correctly, as I mention before, because I run the Spark job command ended on /output/ instead of /spark. This is due to since the job is run, the spark process generates the output folder passed within the command by default. So, if the folder is created in S3 Bucket manually, the process runs correctly but is not able to overwrite the folder. Such overwrite permission must be set explicit to this (I didn't do it) or simply create a subfolder from the job command." Another note says: "The second submission did save the output correctly as shown before."

Amazon EMR > EMR on EC2 cluster > EMR RafaelVera

1. Come back again to the EMR cluster console

3. Check results in Hadoop console (b)

The screenshot shows the Hadoop's Resource Manager UI. It displays two applications: 'Application ID: application_1705199945285_0002' and 'Application ID: application_1705199945285_0001'. Both applications have completed successfully with a 'SUCCEEDED' final status. A note at the bottom right says: "Both applications have been processed by the Hadoop cluster, have completed successfully, and have a 'SUCCEEDED' final status."

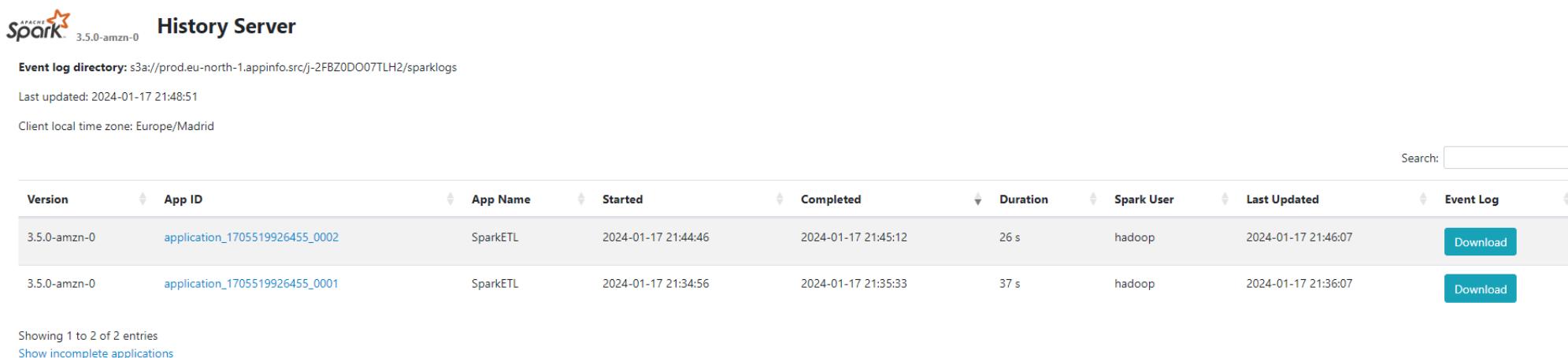
4.a.1 Apache Spark History Server web UI

This History Server UI is used to review and diagnose the performance of Spark applications after they have run

Lists two completed Spark applications :

the first application did not save the output correctly, as I mention before, because I run the Spark Job command ended on /output/ instead of /spark. This is due to since the job is run, the spark process generates the output folder passed within the command by default. So, if the folder is created in S3 Bucket manually, the process runs correctly but is not able to overwrite the folder. Such overwrite permission must be set explicit to this (I didn't do it) or simply create a subfolder from the job command.

The second submission did save the output correctly as shown before.



The screenshot shows the Apache Spark History Server web UI. At the top, it displays the version 3.5.0-amzn-0 and the event log directory s3a://prod.eu-north-1.appinfo/src/j-2FBZ0DO07TLH2/sparklogs. It also shows the last update time as 2024-01-17 21:48:51 and the client local time zone as Europe/Madrid. A search bar is present at the top right. Below this, a table lists two completed applications:

Version	App ID	App Name	Started	Completed	Duration	Spark User	Last Updated	Event Log
3.5.0-amzn-0	application_1705519926455_0002	SparkETL	2024-01-17 21:44:46	2024-01-17 21:45:12	26 s	hadoop	2024-01-17 21:46:07	Download
3.5.0-amzn-0	application_1705519926455_0001	SparkETL	2024-01-17 21:34:56	2024-01-17 21:35:33	37 s	hadoop	2024-01-17 21:36:07	Download

Showing 1 to 2 of 2 entries
[Show incomplete applications](#)

4.a.2 Event timeline for a Spark job

The timeline provides a graphical representation of events that occurred during the execution of the job.

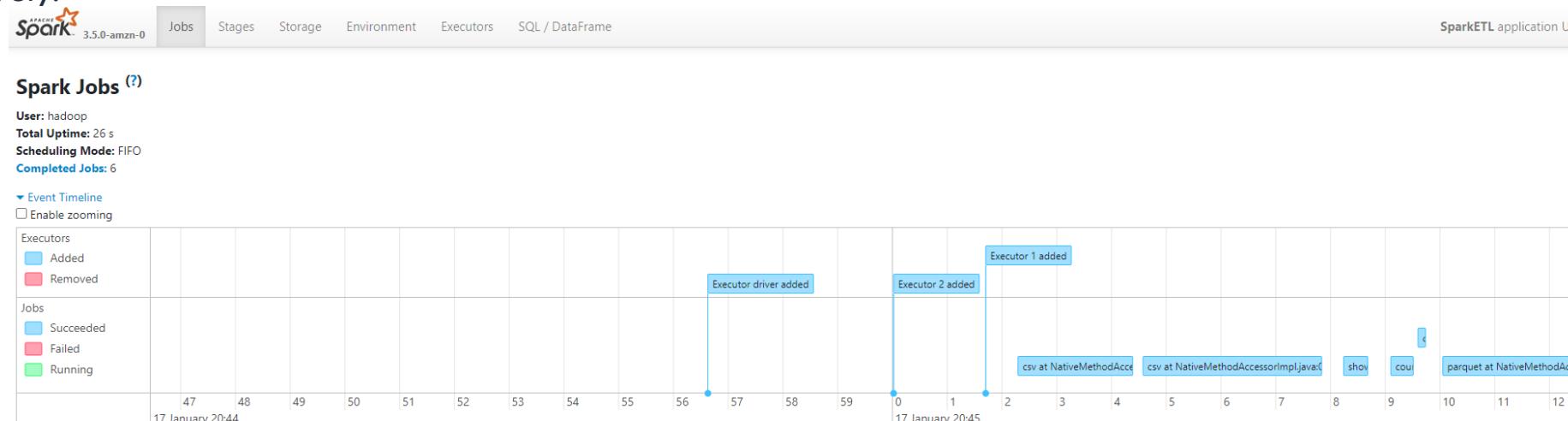
key points:

user running the job: Hadoop. The total runtime of the job: 26 seconds. The scheduling mode: FIFO (First In, First Out)

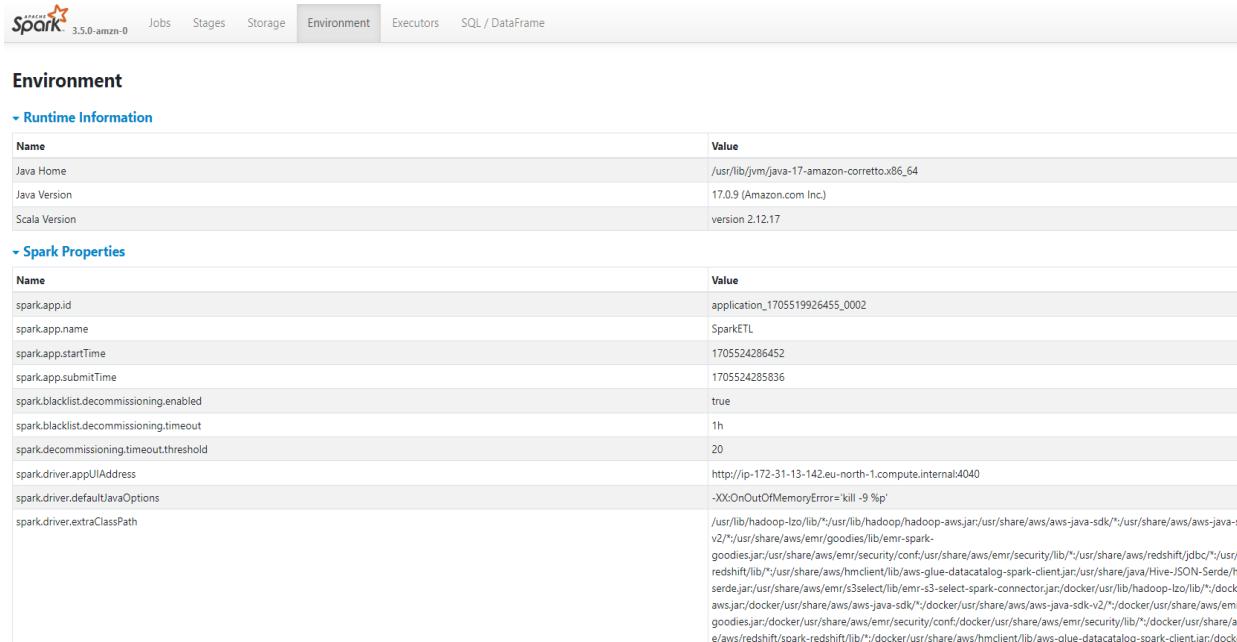
Timeline itself:

17 January 2024 at seconds mark 52 and 54, two events labeled "Executor added" occur, (resources were allocated for the job). Following that, there are several stages marked as "stage 0," "stage 1," etc., which represent different stages of the Spark job. These are likely to be stages of computation that Spark has broken the job into for execution.

On the far right, there are two blue circles, one labeled "csv at NativeMethodAccessorImpl.java" and another labeled "parquet at NativeMethodAccessorImpl.java." These likely indicate the completion of tasks that involve reading or writing data in CSV and Parquet formats, respectively.



4.a.4 Environment



The screenshot shows the Apache Spark History Server web UI with the 'Environment' tab selected. The page is titled 'Environment'. It contains two sections: 'Runtime Information' and 'Spark Properties'.

Runtime Information:

Name	Value
Java Home	/usr/lib/jvm/java-17-amazon-corretto.x86_64
Java Version	17.0.9 (Amazon.com Inc.)
Scala Version	version 2.12.17

Spark Properties:

Name	Value
spark.app.id	application_1705519926455_0002
spark.app.name	SparkETL
spark.app.startTime	1705524286452
spark.app.submitTime	1705524285836
spark.blacklist.decommissioning.enabled	true
spark.blacklist.decommissioning.timeout	1h
spark.decommissioning.timeout.threshold	20
spark.driver.appUIAddress	http://ip-172-31-13-142.eu-north-1.compute.internal:4040
spark.driver.defaultJavaOptions	-XX:OnOutOfMemoryError=kill -9 %p
spark.driver.extraClassPath	/usr/lib/hadoop-lzo/*;/usr/lib/hadoop-hadoop-aws.jar;/usr/share/aws/aws-java-sdk/*;/usr/share/aws/aws-java-sdk-v2/*;/usr/share/aws/emr/goodies/lib/emr-spark-goodies.jar;/usr/share/aws/emr/security/conf;/usr/share/aws/emr/security/lib/*;/usr/share/aws/redshift/jdbc/*;/usr/share/aws/redshift/lib/*;/usr/share/aws/hmclient/lib/aws-glue-datacatalog-spark-client.jar;/usr/share/java/Hive-JSON-Serde/hive-serde.jar;/usr/share/aws/emr-s3select/lib/emr-s3-select-spark-connector.jar;/docker/usr/lib/hadoop-lzo/lib/*;/docker aws.jar;/docker/usr/share/aws/aws-java-sdk/*;/docker/usr/share/aws/aws-java-sdk-v2/*;/docker/usr/share/aws/emr/goodies.jar;/docker/usr/share/aws/emr/security/conf;/docker/usr/share/aws/emr/security/lib/*;/docker/usr/share/aws/redshift-soar-redshift/lib/*;/docker/usr/share/aws/hmclient/lib/aws-glue-datacatalog-spark-client.jar;/docker

Shows various runtime information and Spark properties set for the application.

Runtime Information:

Java Home: /usr/lib/jvm/java-17-amazon-corretto.x86_64,
(Amazon Corretto 17 as the Java environment)

Java Version: Version 17.0.9

Scala Version: 2.12.17.

Spark Properties:

spark.app.id and spark.app.name are shown, with name SparkETL.
spark.executorEnv.JAVA_HOME and
spark.yarn.appMasterEnv.JAVA_HOME Amazon Corretto Java 17 path.

Other properties such as spark.driver.memory, spark.executor.memory, spark.executor.cores, spark.dynamicAllocation.enabled, spark.eventLog.enabled, spark.master, spark.sql.warehouse.dir, spark.blacklist.decommissioning.enabled, spark.shuffle.service.enabled, and spark.ui.filters are listed, each influencing various aspects of Spark application performance, resource allocation, and user interface configuration.

Amazon Corretto is a performance-optimized, long-term support OpenJDK distribution by Amazon, suitable for Java applications such as Apache Spark.

It is open-source and multi-platform compatible. Proper configuration of Spark properties, including application naming, memory allocation, and dynamic resource management, is crucial for optimal performance and stability. Corretto's use in Spark can notably impact performance.

4.a.5 Spark properties extended

spark.driver.memory (2048M) and spark.executor.memory (9486M): These properties define the amount of memory allocated to the Spark driver and executors, respectively. They are crucial for the performance and stability of Spark applications, especially when processing large volumes of data.

spark.executor.cores (4) and spark.emr.default.executor.cores (4): These configure the number of CPU cores to be allocated to each executor. They are important for application parallelism and performance.

spark.dynamicAllocation.enabled (true): This property, when enabled, allows Spark to automatically adjust the number of executors used by the application based on the workload. It is useful for optimizing resource usage.

spark.eventLog.enabled (true) and spark.eventLog.dir (hdfs://var/log/spark/apps): These properties enable and define the location for Spark event logs, which are essential for debugging and performance analysis of the application.

spark.master (yarn): Indicates the cluster manager used for running the application. In this case, it is YARN, which is common in Hadoop environments.

spark.sql.warehouse.dir (hdfs://user/spark/warehouse): This is the default location for storing table data when working with Spark SQL and DataFrames.

spark.blacklist.decommissioning.enabled (true) and spark.blacklist.decommissioning.timeout (1h): These properties are related to the management of unreliable or unstable resources in the cluster.

spark.shuffle.service.enabled (true): This setting is important for the performance of shuffle operations in Spark, especially in large-scale applications.

Apache Spark History Server web UI | Job results



January 2024

4.a.6 Executors

The screenshot shows the Apache Spark History Server web UI with the 'Executors' tab selected. The top navigation bar includes links for Jobs, Stages, Storage, Environment, Executors (which is highlighted), and SQL / DataFrame. The title bar indicates the application is 'SparkETL application UI'. The main content area is titled 'Executors' and contains two sections: 'Summary' and 'Executors'.

Summary: A table showing the status of executors. It has three rows: Active(3), Dead(0), and Total(3). Key metrics include Storage Memory (0.0 B / 10.6 GiB), Disk Used (0.0 B), Cores (8), Active Tasks (0), Failed Tasks (0), Complete Tasks (6), Total Tasks (6), Task Time (GC Time) (35 s (0.3 s)), Input (4.3 MiB), Shuffle Read (50 B), Shuffle Write (50 B), and Excluded (0).

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(3)	0	0.0 B / 10.6 GiB	0.0 B	8	0	0	6	6	35 s (0.3 s)	4.3 MiB	50 B	50 B	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(3)	0	0.0 B / 10.6 GiB	0.0 B	8	0	0	6	6	35 s (0.3 s)	4.3 MiB	50 B	50 B	0

Executors: A table listing active executors. It includes columns for Executor ID, Address, Status, RDD Blocks, Storage Memory, Disk Used, Cores, Active Tasks, Failed Tasks, Complete Tasks, Total Tasks, Task Time (GC Time), Input, Shuffle Read, Shuffle Write, Logs, Add Time, and Remove Time. The table shows three executors: driver (ip-172-31-13-142.eu-north-1.compute.internal:33045), 1 (ip-172-31-9-211.eu-north-1.compute.internal:45695), and 2 (ip-172-31-5-140.eu-north-1.compute.internal:35341). The driver is active with 0 tasks. Executors 1 and 2 have 2 and 4 tasks respectively, all completed.

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs	Add Time	Remove Time
driver	ip-172-31-13-142.eu-north-1.compute.internal:33045	Active	0	0.0 B / 1 GiB	0.0 B	0	0	0	0	0	26 s (0.0 ms)	0.0 B	0.0 B	0.0 B		2024-01-17 21:44:56	-
1	ip-172-31-9-211.eu-north-1.compute.internal:45695	Active	0	0.0 B / 4.8 GiB	0.0 B	4	0	0	2	2	2 s (72.0 ms)	23.5 KiB	0.0 B	0.0 B	stderr stdout	2024-01-17 21:45:01	-
2	ip-172-31-5-140.eu-north-1.compute.internal:35341	Active	0	0.0 B / 4.8 GiB	0.0 B	4	0	0	4	4	6 s (0.2 s)	4.3 MiB	50 B	50 B	stderr stdout	2024-01-17 21:45:00	-

Showing 1 to 3 of 3 entries

Previous 1 Next

Miscellaneous Process: A table showing processes. It includes columns for Process ID and Address. The table shows one process: driver (ip-172-31-13-142.eu-north-1.compute.internal:33045).

Process ID	Address
driver	ip-172-31-13-142.eu-north-1.compute.internal:33045

Show 20 entries

Search:

Displays the metrics and status of Spark executors for the running application.

Executors Summary: Shows an overview with one active driver and two active executors, indicating no failed tasks.

Memory Usage: The driver has 1 GB of storage memory, and each executor has 4.8 GB allocated, with no disk used by any.

Tasks: The driver has not completed any tasks, which is typical, as the driver coordinates the executors. Executors have completed 6 tasks.

Task Time: The total task time is 35 s, with individual task times listed for each executor.

Shuffle Read/Write: minimal shuffle activity, indicating the tasks did not involve significant data shuffling or it's an initial stage of processing.

Logs: Each executor has a stdout and stderr log available for review.

Apache Spark History Server web UI | Job results



January 2024

4.a.7 Stages

Completed Stages (6):

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
6	parquet at NativeMethodAccessorImpl.java:0	+details 2024/01/17 20:45:10	2 s	1/1	1461.9 kB	219.8 kB		
5	count at NativeMethodAccessorImpl.java:0	+details 2024/01/17 20:45:09	0.1 s	1/1		50.0 B		
3	count at NativeMethodAccessorImpl.java:0	+details 2024/01/17 20:45:09	0.4 s	1/1	1461.9 kB		50.0 B	
2	showString at NativeMethodAccessorImpl.java:0	+details 2024/01/17 20:45:08	0.4 s	1/1	16.0 kB			
1	csv at NativeMethodAccessorImpl.java:0	+details 2024/01/17 20:45:04	3 s	1/1	1461.9 kB			
0	csv at NativeMethodAccessorImpl.java:0	+details 2024/01/17 20:45:02	2 s	1/1	7.5 kB			

Skipped Stages (1):

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
4	count at NativeMethodAccessorImpl.java:0	+details Unknown	Unknown	0/1				

The Spark application has processed several stages involving reading from and writing to CSV and Parquet files, and performing count operations

Completed Stages (6):

Stage 6: Involved an operation with Parquet files, took 2.5 seconds, processed 163.9 kB of input and produced 219.8 kB of output.

Stage 5: Involved a count operation, took 0.1 seconds, processed 163.9 kB of input, and no output is listed.

Stage 3: Another count operation, took 0.4 seconds, same input size as Stage 5 and no output.

Stage 2: Show operation, took 0.4 seconds, with 160 kB input and 7.5 kB output.

Stage 1: CSV operation, took 3.5 seconds, with 163.9 kB input and no output listed.

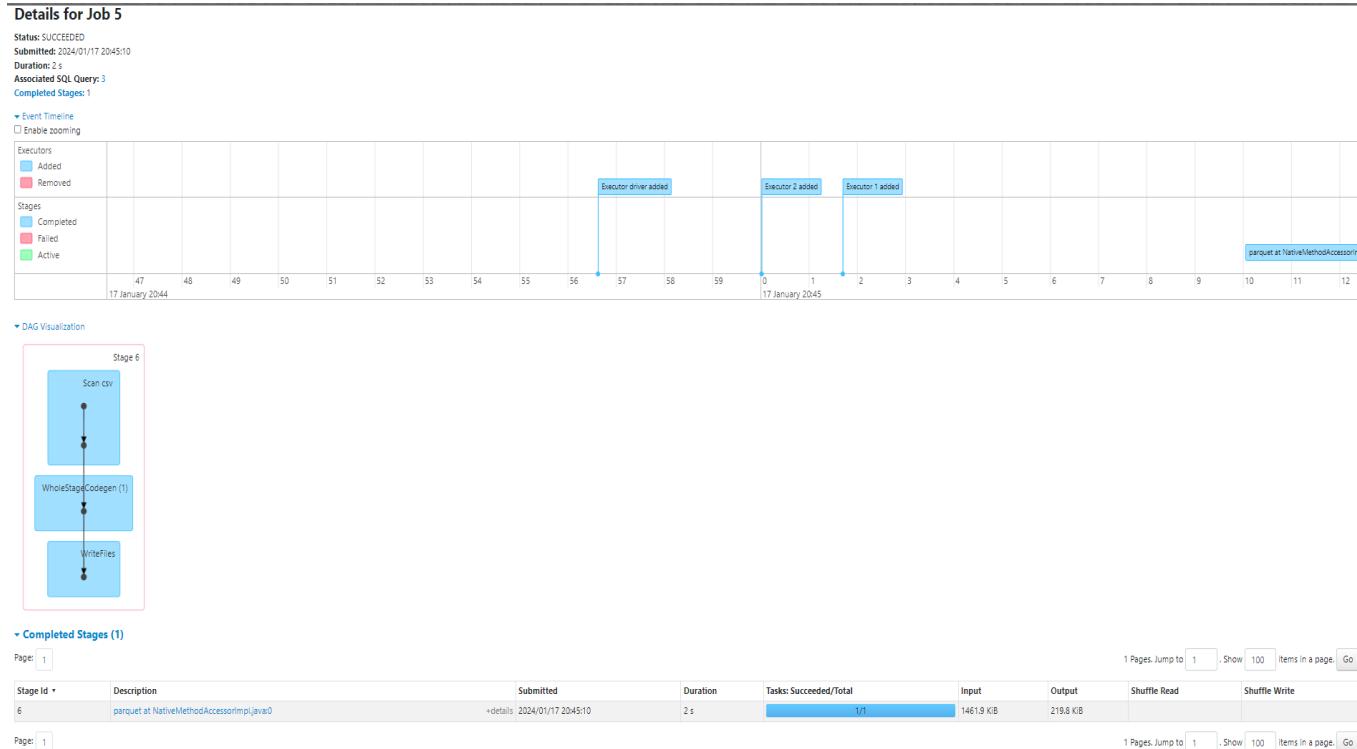
Stage 0: CSV operation, took 2 seconds, with 7.5 kB input and no output listed.

Each stage had one task that succeeded.

Skipped Stages (1):

Stage 4: A count operation that was skipped, therefore no duration, task, input, or output information is available.

4.a.8 Stages | Details for Job 5



Job ID: 5

Status: SUCCEEDED

Submission Time: 2024/01/17 20:45:10

Duration: The job took a total of 2.5 seconds to run.

Stages: 1 stage is completed, and 0 stages are skipped.

Active Stages: 0, Completed Stages: 1

Skipped Stages: 0, Failed Stages: 0

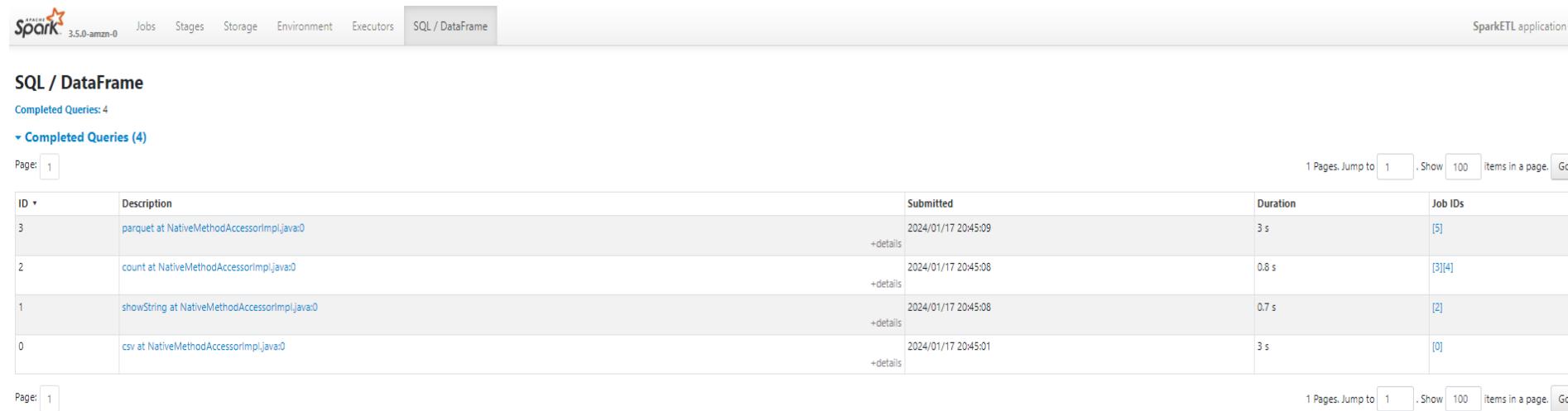
In the Event Timeline, the following events are marked:
 Executors are added at three different points in time.
 A stage is marked as completed.

The DAG Visualization (Directed Acyclic Graph) of the job's stages, with one stage, labeled "Stage 1", indicating the flow of tasks within the job.

The Completed Stages section lists:

Stage ID 6: Described as "parquet at NativeMethodAccessormpl.java", submitted on January 17, 2024, and took 2.5 seconds to complete. The input size was 163.9 KB, and the output size was 219.8 KB. There was 1 task for this stage, and it succeeded.

4.a.9 SQL / DataFrame



ID	Description	Submitted	Duration	Job IDs
3	parquet at NativeMethodAccessorImpl.java:0	2024/01/17 20:45:09 +details	3 s	[5]
2	count at NativeMethodAccessorImpl.java:0	2024/01/17 20:45:08 +details	0.8 s	[3][4]
1	showString at NativeMethodAccessorImpl.java:0	2024/01/17 20:45:08 +details	0.7 s	[2]
0	csv at NativeMethodAccessorImpl.java:0	2024/01/17 20:45:01 +details	3 s	[0]

List of completed queries for a Spark application

This tab is useful for performance tuning and debugging of Spark SQL queries, as it provides a clear mapping from high-level DataFrame operations to the underlying jobs executed by Spark.

There are four completed queries listed: ID (0 to 3).

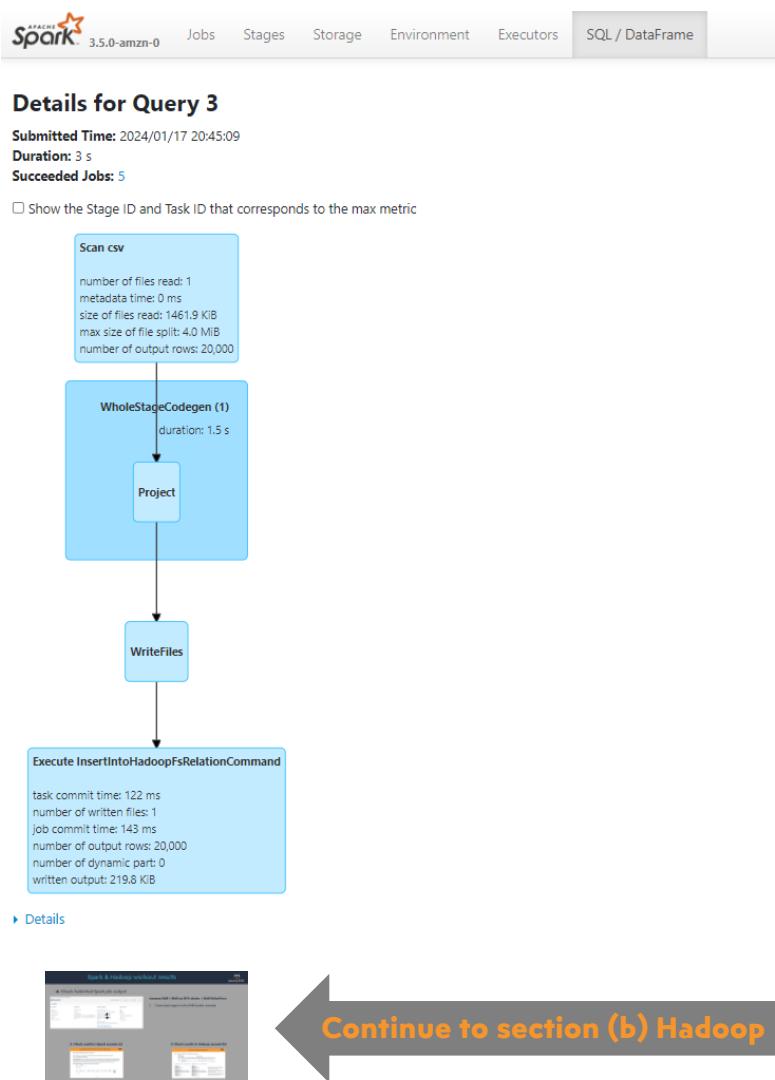
Each query has a description that includes a method invocation, likely indicating a DataFrame operation. These descriptions involve operations such as 'save' or 'count', and they reference a Java method, likely where the DataFrame operation is triggered.

Submitted: submission times for all queries.

Duration: how long each query took to complete.

Job ID: links each query to the corresponding Spark job, providing a way to trace the SQL/DataFrame operations back to the broader context of the Spark job they are part of.

4.a.10 SQL / DataFrame | Details for Query 3



Query execution plan is visualized as a DAG:

Scan csv: read 1 file with metadata time of 0 ms, and the size of files read was 1461.9 KB. The maximum size of file split was 40 MB, and the number of output rows was 20,000.

WholeStageCodegen (1): A performance optimization feature in Spark SQL that compiles parts of the query to bytecode. This stage had a duration of 1.5 seconds.

Project: A transformation operation that is likely manipulating or transforming the data in some way. In our case, adding a new column to the dataframe with current date (datetime)

WriteFiles: This operation is responsible for writing the output of the transformation to storage.

Execute InsertIntoHadoopFsRelationCommand: This command indicates an action to insert data into a Hadoop FileSystem (HDFS) relation (table or file), and it involved:

Task commit time: 122 ms, Number of written files: 1

Job commit time: 143 ms, Number of output rows: 20,000

Number of dynamic partitions: 0, Written output: 219.8 KB

The diagram visually represents the execution plan of the query, illustrating the flow from reading a CSV file, processing the data (projecting), and then writing the results to the filesystem. The metrics provide insights into the efficiency of the operation, including the speed of reading, processing, and writing the data.

Hadoop's Resource Manager UI

4.b.1 Hadoop application submitted overview

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	Finish Time	State	FinalStatus	Progress	Tracking UI
application_1705519926455_0002	hadoop	SparkETL	SPARK		default	0	Wed Jan 17 21:44:51 +0100 2024	Wed Jan 17 21:44:51 +0100 2024	Wed Jan 17 21:45:12 +0100 2024	FINISHED	SUCCEEDED	<div style="width: 100%;">100%</div>	History
application_1705519926455_0001	hadoop	SparkETL	SPARK		default	0	Wed Jan 17 21:35:11 +0100 2024	Wed Jan 17 21:35:12 +0100 2024	Wed Jan 17 21:35:33 +0100 2024	FINISHED	SUCCEEDED	<div style="width: 100%;">100%</div>	History

Application ID: application_1705199945285_0002

User: hadoop
Name: SparkETL
Application Type: SPARK
Tags: SPARK
Queue: default
Application Priority: 0
Start Time: Wed Jan 17 20:44:41 +0100 2024
Launch Time: Wed Jan 17 20:44:51 +0100 2024
Finish Time: Wed Jan 17 20:45:12 +0100 2024
State: FINISHED
Final Status: SUCCEEDED
Progress: 100%
Tracking UI: History

Application ID: application_1705199945285_0001

User: hadoop
Name: SparkETL
Application Type: SPARK
Tags: SPARK
Queue: default
Application Priority: 0
Start Time: Wed Jan 17 20:34:36 +0100 2024
Launch Time: Wed Jan 17 20:34:52 +0100 2024
Finish Time: Wed Jan 17 20:35:33 +0100 2024
State: FINISHED
Final Status: SUCCEEDED
Progress: 100%
Tracking UI: History

Both applications have been processed by the Hadoop cluster, have completed successfully, and have a "SUCCEEDED" final status.

The "History" link under the Tracking UI column likely allows users to view detailed logs and metrics for each application's run.

This interface is typically used for monitoring and managing the performance and resource usage of applications running on a Hadoop cluster.

Hadoop's Resource Manager UI



January 2024

4.b.2 Hadoop processed Spark application details

The screenshot shows the Hadoop Resource Manager UI. On the left, there is a sidebar with a yellow elephant icon and the word "hadoop". The sidebar menu includes "Application History", "About Applications" (with sub-options "FINISHED", "FAILED", "KILLED"), and "Tools". The main content area is titled "Application application_1705519926455_0002". It displays various application details:

Application Overview	
User:	hadoop
Name:	SparkETL
Application Type:	SPARK
Application Tags:	
Application Priority:	0 (Higher integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Wed Jan 17 20:44:51 +0000 2024
Launched:	Wed Jan 17 20:44:51 +0000 2024
Finished:	Wed Jan 17 20:45:12 +0000 2024
Elapsed:	21sec
Tracking URL:	History
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

Below this, there is a table with columns "Attempt ID", "Started", "Node", and "Logs". One entry is listed:

Attempt ID	Started	Node	Logs
appattempt_1705519926455_0002_000001	Wed Jan 17 21:44:51 +0100 2024	http://ip-172-31-5-140.eu-north-1.compute.internal:8042	Logs

At the bottom, it says "Showing 1 to 1 of 1 entries" and has navigation buttons for First, Previous, Next, and Last.

Below is a table with the following columns: Attempt ID, Started, Node, and Logs.

Only one attempt is listed:

Attempt ID: application_1705519926455_0002_000001

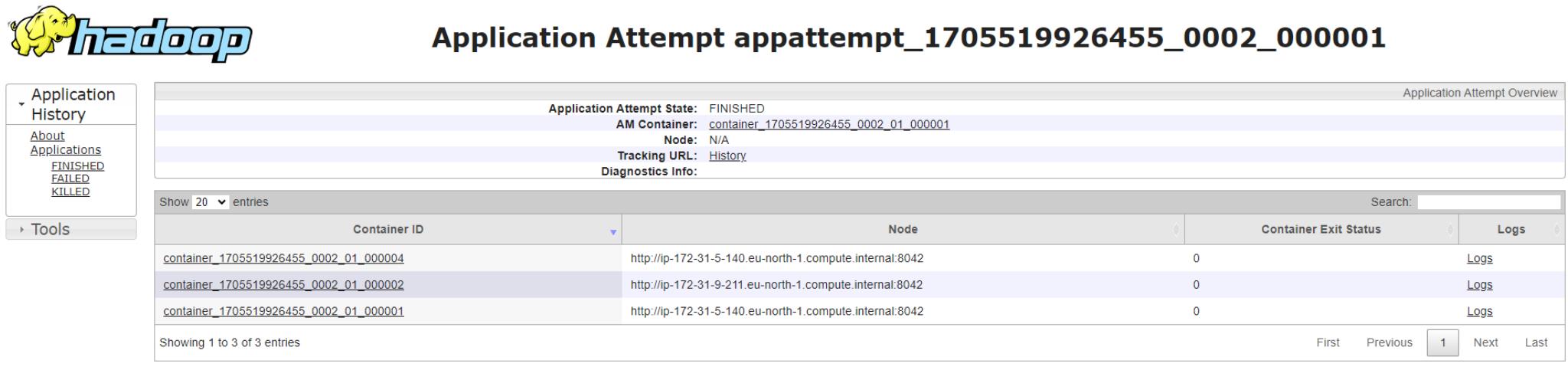
Started: Wed Jan 17 20:44:51 +0100 2024

Node: (a URL indicating the node's address within the cluster)

Logs: There's a link to view the logs, which would provide detailed information about the application's execution.

The application named SparkETL was submitted by the user "hadoop" and successfully completed within 21 seconds. The application ran in the default queue with the lowest priority and did not have any specific node labeling for resource allocation. The "Tracking UI" points to the "History" which means detailed historical data about the application's execution can be accessed for further analysis.

4.b.3 Application attempt within the Hadoop ecosystem



The screenshot shows the Hadoop Resource Manager UI. On the left, there is a sidebar with a yellow elephant icon and the word "hadoop". The main area has a title "Application Attempt appattempt_1705519926455_0002_000001". Below the title, it says "Application Attempt State: FINISHED", "AM Container: container_1705519926455_0002_01_000001", "Node: N/A", and "Tracking URL: History". A "Diagnostics Info:" section follows. A table titled "Application Attempt Overview" lists three containers. The columns are "Container ID", "Node", "Container Exit Status", and "Logs". The rows show:

Container ID	Node	Container Exit Status	Logs
container_1705519926455_0002_01_000004	http://ip-172-31-5-140.eu-north-1.compute.internal:8042	0	Logs
container_1705519926455_0002_01_000002	http://ip-172-31-9-211.eu-north-1.compute.internal:8042	0	Logs
container_1705519926455_0002_01_000001	http://ip-172-31-5-140.eu-north-1.compute.internal:8042	0	Logs

At the bottom, it says "Showing 1 to 3 of 3 entries" and has navigation buttons: First, Previous, 1 (highlighted), Next, Last.

Below there is a table with the following columns:

Container ID: Three different container IDs are listed (container_1705519926455_0002_01_000001, container_1705519926455_0002_01_000002, and container_1705519926455_0002_01_000003).

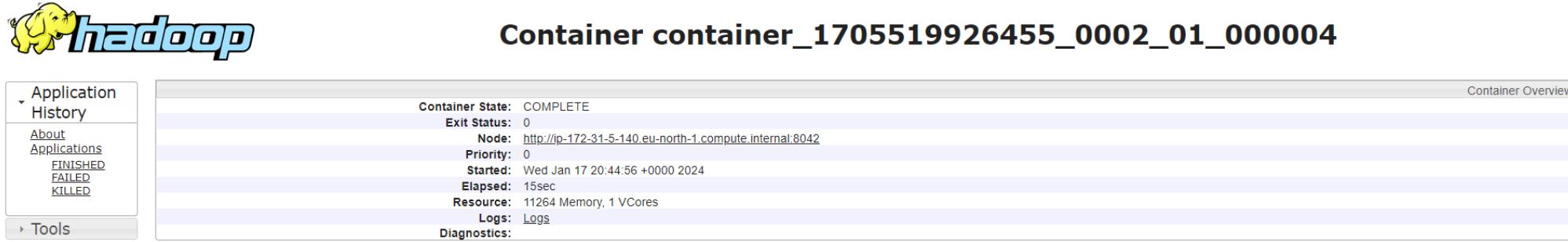
Node: Lists the HTTP addresses of the nodes where these containers were located, which are internal addresses within the Hadoop cluster.

Container Exit Status: All three containers have an exit status of '0', which conventionally means that they exited without error.

Logs: Each container has a link to its respective logs, which would provide detailed runtime information for each container.

This information is typically used to monitor the status and health of specific application attempts and to debug if there were any issues during the execution of these application attempts. The successful exit status for all containers and the "FINISHED" state of the application attempt indicate that this particular run of the application was successful.

4.b.4 Details for a specific container associated with a Hadoop application



The screenshot shows the Hadoop Resource Manager UI. On the left, there's a sidebar with a yellow elephant icon and the word "hadoop". The main area has a title "Container container_1705519926455_0002_01_000004". Below the title is a table with the following data:

Container Overview	
Container State:	COMPLETE
Exit Status:	0
Node:	http://ip-172-31-5-140.eu-north-1.compute.internal:8042
Priority:	0
Started:	Wed Jan 17 20:44:56 +0000 2024
Elapsed:	15sec
Resource:	11264 Memory, 1 VCores
Logs:	Logs
Diagnostics:	

Node: The URL provided is the address where the container was running within the Hadoop cluster.

Elapsed: The time the container was running, which is 15 seconds.

Resources: The resources allocated to this container are 1,024 MB of memory and 1 vCore (virtual CPU).

This information is used to audit and troubleshoot the performance of specific containers within Hadoop applications, ensuring that they are executing as expected and utilizing the allocated resources effectively. The successful exit status and the "COMPLETE" state indicate that this container ran its task successfully.