Step:1

you can create an IAM user named `redfine_user` with administrative permissions in both the AWS Management Console (UI) and programmatically (using AWS CLI)

Step:2

To create a key pair in AWS,named `redfine_keypair_amazon_linux`

Step:3

Login to IAM User

Step:4

To create S3 Bucket

Named {

Redfine-data-zone-423701805547-us-east-1

Testing_423701805547-us-east-1

scripts-423701805547-us-east-1/

redfine_transform_zone-423701805547-us-east-1

Aws-emr-studio-423701805547-us-east-1

Aws-logs-423701805547-us-east-1

}

Step:5

To create VPC

Option: VPC and more

Namd `redfine_emr_vpc`

To Configuretion

{

Zone:2

Public_Subnet:2

Private_subnet:0

Net Getway:Non

Vpc Gatway endpoint:S3 Getway

Other option Bydefault

}

Step:6

To create EMR

Namd `redfine_emr_cluster`

Application bundle

{

Spark 3.5.0

Hadoop 3.3.6

Hive 3.1.3

JupyterEnterpriseGateway 2.6.0

JupyterHub 1.5.0

Zeppelin 0.10.1

}

Cluster configuration:

- Uniform instance groups

Networking - required:

My VPC and Subnet

Amazon EMR service role:

Namd `AmazonEMR-ServiceRole-20240708T142236`

Permissions policie {

AdministratorAccess
AmazonEC2FullAccess
AmazonEMRFullAccessPolicy_v2
AmazonEMRServicePolicy_v2
AmazonS3FullAccess
IAMFullAccess
}

EC2 instance profile for Amazon EMR:

Namd `AmazonEMR-InstanceProfile-20240708T142217`

Permissions policie {

AmazonEMR-InstanceProfile-Policy-20240708
T142217

}

Case:1

Step:9

To create cloud9 IDE

Named: `EMR_Masterclass_Cloud9`

Environment type: New EC2 instance

New EC2 instance:

Instance type:t2.micro (1 GiB RAM + 1 vCPU)

Platform:Amazon Linux 2

Timeout :30

Connection:Secure Shell (SSH***)***

Amazon Virtual Private Cloud (VPC) my vpc

<u>Step:10</u>

Upload keypair on Cloud9 IDE

<u>Step:11</u>

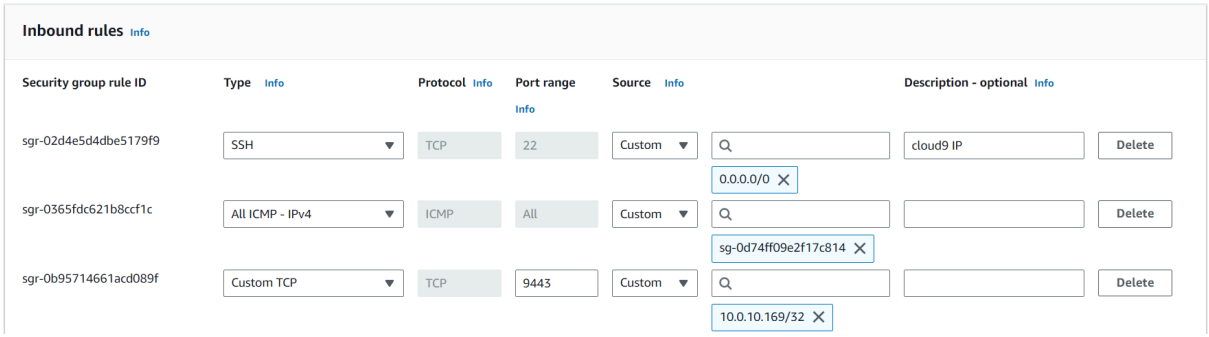
chmod 400: This command sets file permissions so that only the owner can read the file, ensuring sensitive data remains protected**.**

chmod 400 path/to/your/private-key.pem

<u>Step:12</u>

Go To the instance page and select to the EC2 Cloud9 and pick up the private IP
Then Go the master node of the EMR and Security Tab and add the connection

1. SSH =>privater IP/32
2. Customer TCP =>9443 ->any IPV4



commands.py
connection
ssh -i [key-pair] hadoop@[emr-master-public-dns-address]

nano spark-etl.py
Write the custom code

Spark-submit commands:
spark-submit spark-etl.py s3://<YOUR-BUCKET>/input/ s3://<YOUR-BUCKET>/output/spark

spark-submit s3://<bucketname>/files/spark-etl.py s3://<bucketname>/input s3://<bucketname>/output

Case:2

Step:9

To create some additional folder

Named {
Input_raw_data
Output_transformed_data
Script_file
}
Then upload the python code file in script_file folder

Step:10

Go To the EMR Cluster and add the steps

Type: Custom Jar

Named: `Masterclass_step_spark`

Location: command-runner.jar

Argument:

{
Spark-submit s3://<bucketname>/files/spark-etl.py s3://<bucketname>/input s3://<bucketname>/output
}

like:
{
spark-submit s3://redfine-data-zone-7266/emr_masterclss_code_with_yu/file/spark-etl.py s3://redfine-data-zone-7266/emr_masterclss_code_with_yu/input/tripdata.csv s3://redfine-data-zone-7266/emr_masterclss_code_with_yu/output/steps/spark/
}

Jar location	Permissions	Main class
command-runner.jar	-	-
Action on failure	Argument	
Continue	spark-submit s3://redfine-data-zone-7266/emr_masterclss_code_with_yu/file/spark-etl.py s3://redfine-data-zone-7266/emr_masterclss_code_with_yu/input/tripdata.csv s3://redfine-data-zone-7266/emr_masterclss_code_with_yu/output/steps/spark/	