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-20240708T142217

`}`

EMR Studio: Studios

Namd `redfine_data_team`

Setup options

- Custom

Service role to let Studio access your AWS resources

Namd `AmazonEMR-ServiceRole-20240708T142236`

After that EMR Connect to the Jupytenotbook then attach it

Upload the notebook