Days:

day1

-AWS console and introduction

-IAM

-IAM is a Global service mainly for Authentication and Authorization

-Route53 is also a global service

Cloud formation: It is mainy for infrastructure as a code

Type of instance:

1. On demand

2. Reserved

3. Spot instance

GLobal infrastructure:

Region

Availability zone

edge location

CDN(CONTENT DELIVERY NETWORK): origin of the data is somewhere else but

Types of machine type

1.General purpose type

2.storage type

3.memory type

4.Network optimized

5.Compute optimized

NACL-Network access control list

IAM is a Global service mainly for Authentication and Authorization

JSON-javascript object notation

IAM policy

-Identity based policy

-Resource based policy-eg:sns,s3

Compute services

Ec2

-on demand

-Reserved

-spot instance

You can connect Aws through console,Cli,sdk/cdk,Iac

6 pillars of Aws

-operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability

Three types of storage

-object storage -s3

-block storage -EBS

-File storage -EFS

-Compute service

-IAM

-Storage

-container

-Integration service (SNS SQS)

-Serverless service(api,dynamo db,lambda)

-Database service-Rds

-Monitoring service-cloud watch

-Management services and Govt services

-Network services(vpc,loadbalancer)

DEVOPS PROFESSIONAL:

-code build

-code pipeline

-system manager

-Difference between dedicated instance and dedicated host

-Difference between SAAS,PAS,IAS

-Purchase options

-What is placement group

**DAY-2**

IAM user

Group

Roles

DAY-3(Launching ec2 isntance,launching vm in windows,ssh connection to windows ,hosting static website in windows with global access enabled and restricted)

Instance:

1:Name of the instance

2.AMI(Amazon machine images)-OS+pre installed softwares

3.Instance type

**-General purpose-Tseries**

**-Compute optimized- High perfomance-cseries**

**-Memory optimized-Datasets -Rseries**

**-storage optimized-Sequential read and write**

4.Key-pair is required for authentication purpose -pem file

If key is lost we can get back using session manager

Ssh connection:

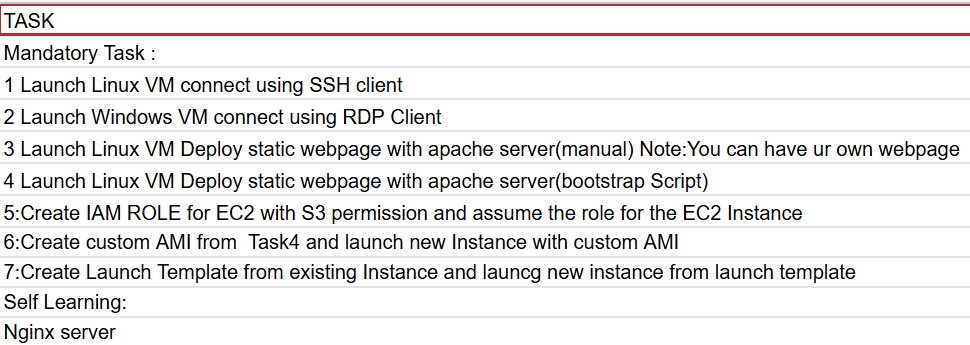
ssh -i “ust1.pem” username@publicip of machine

yum update -y -Linux machine update

yum install httpd -y

systemctl status httpd

systemctl restart httpd



Task1:

Placement group

Spot

Partition

**DAY-3**

-Code commit-source code management in aws

--Iam—user—assign mfa device—

Aws s3 ls

Aws configure

Aws –version

Aws s3 mb s3://my-new-bucket

Task1:

**Create IAM user and attach the policy and validate:**

-Go to IAM → Click Users → Click Add user

- **User details:**

* Username: user1 (or any name)
* Select **Access key - Programmatic access** (for CLI/API)

 **Permissions:**

* Choose **Attach policies directly**
* Select a policy (e.g., AmazonS3FullAccess or AdministratorAccess as needed)
* Then create bucker in user1 account you can create it but now go to root account and change the permission

Then check with keeping policy as s3readonlyaccess you cannot create bucket in the user1 account

**Task2**

**Create IAM groups and add various users and validate**

-Go to AWS Console > IAM > User groups

-Click Create group ->Group name:anything

-**Attach policies (optional)**:

* Example: AmazonS3ReadOnlyAccess for read-only users

-Click **Create group**

**Creating users**

IAM > Users > Add users

Add usernames (e.g., user1, user2)

IAM > User groups > Select your group

Go to **Users tab** → Click **Add users**

Select the users (e.g., user1, user2) → Add

IAM > User groups > [Select your group] > **Permissions** tab

Click **Add permissions**

Choose **Attach policies directly**

Add a policy (e.g., AmazonEC2ReadOnlyAccess, AmazonS3FullAccess)

Click **Add permissions**

**Task3:**

Go to **IAM > Roles > Create Role**

Choose **AWS service**

**Use case**: Choose **EC2**, click **Next**

Attach a policy like AmazonS3ReadOnlyAccess or any policy you want.

Name the role: e.g., EC2\_S3\_ReadOnly\_Role

Create the role.

**Attach IAM Role to EC2 Instance**

Go to **EC2 > Instances**

Select your instance

Choose **Actions > Security > Modify IAM Role**

Attach the role you created (EC2\_S3\_ReadOnly\_Role)

Click **Update IAM role**

**SSH into your EC2 instance:**

aws s3 ls(listing all the available buckets)

aws s3 mb s3://my-buck-name

**Task4:**

Task 6: enable MFA and validate

Security credentials tab, scroll to “Multi-factor authentication (MFA)”

Click “Assign MFA device”

Scan the **QR code** using your app

Enter **two consecutive MFA codes** generated by the app

Click **Assign MFA**

**Task7:**

**DAY-4**

**Class A – Network bit -8**

Host bits -24

Default subnet for class A -255.0.0.0

Cidr -/8

Class B – 16 net bits 16 hostbits

Cidr-/16

Ip-255.255.0.0

-Class c

Default subnet mask -255.255.255.0

Cidr-/24

Nw bits=24 hs bit=8

No of hosts=2^8-2 for aws 2^8-5

**Step-1**

-Create vpc -192.168.3.0/24

Name:NCPL-VPC-A

**Step-2: create two subnet name it public subnt and private subnet**

-2a-Create public subnet -192.168.3.0/25 under VPC-A

NAME:VPC-A-Pubsub

-2b-Create public subnet -192.168.3.128/25 under VPC-A

NAME:VPC-A-PriSub

Step 3:Create two route table ,one for each subnet

3a-route table1

NAME:A-PubSub-RT

3b-route-Table1

NAME:A-PriSub-RT

Step-4:subnet Association

Ie:A-PubSub-RT must be associated with VPC-A-PubSub

Ie:A-PriSub-RT must be associated with VPC-A-PriSub

Step5: Create Internet Gateway and attach it wih VPC-A

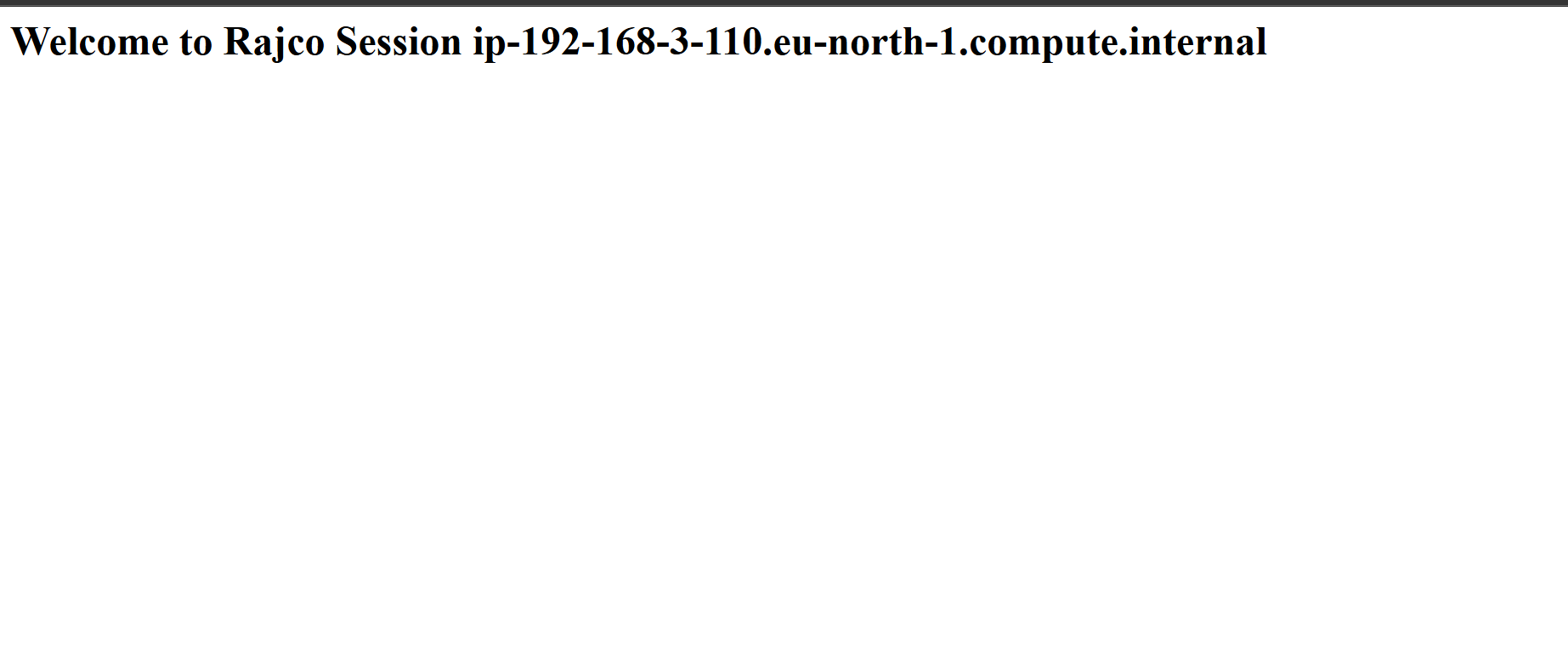
NAME:UST-ITGW

Step6:To make a subnet VPC-A-PubSubas public -go to route table of A-PubSub-RT of the subnet add the route to destination 0.0.0.0./0 through target internetgateway

Step7:Create security group under UST-VPC-A with needed inbounds rule(ssh,http)

Step8:launch ec2 instance under custom UST-VPC-A under public subnet A-PubSub with auto assign public ip enabled

Output:



Doing the above task using Aws command line:

**step1:**Configure Aws: Aws configure

export AWS\_ACCESS\_KEY\_ID=

AWS\_SECRET\_ACCESS\_KEY=

AWS\_SESSION\_TOKEN=

**Step-1**

**Create a vpc**

aws ec2 create-vpc --cidr-block 192.168.3.0/24 --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=NCPL-VPC-A}]'

**Step-2**

**Create Subnets**

**Public subnet:**

aws ec2 create-subnet --vpc-id <VPC-ID> --cidr-block 192.168.3.0/25 --availability-zone eu-north-1a --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=VPC-A-PubSub}]'

**Private subnet:**

aws ec2 create-subnet --vpc-id <VPC-ID> --cidr-block 192.168.3.128/25 --availability-zone eu-north-1a --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=VPC-A-PriSub}]'

**Create Route Tables**

**Public Route Table**

aws ec2 create-route-table --vpc-id <VPC-ID> --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=A-PubSub-RT}]'

**Private Route Table**

aws ec2 create-route-table --vpc-id <VPC-ID> --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=A-PriSub-RT}]'

**DAY-5**

**Amazon rds –**

**In a vpc if you**

**NAT gateway- it enables internet access to private instance in private subnet**

**Bastion host: to enable**