(ON CONSOLE)

Q 01.

1. Create Security Group:

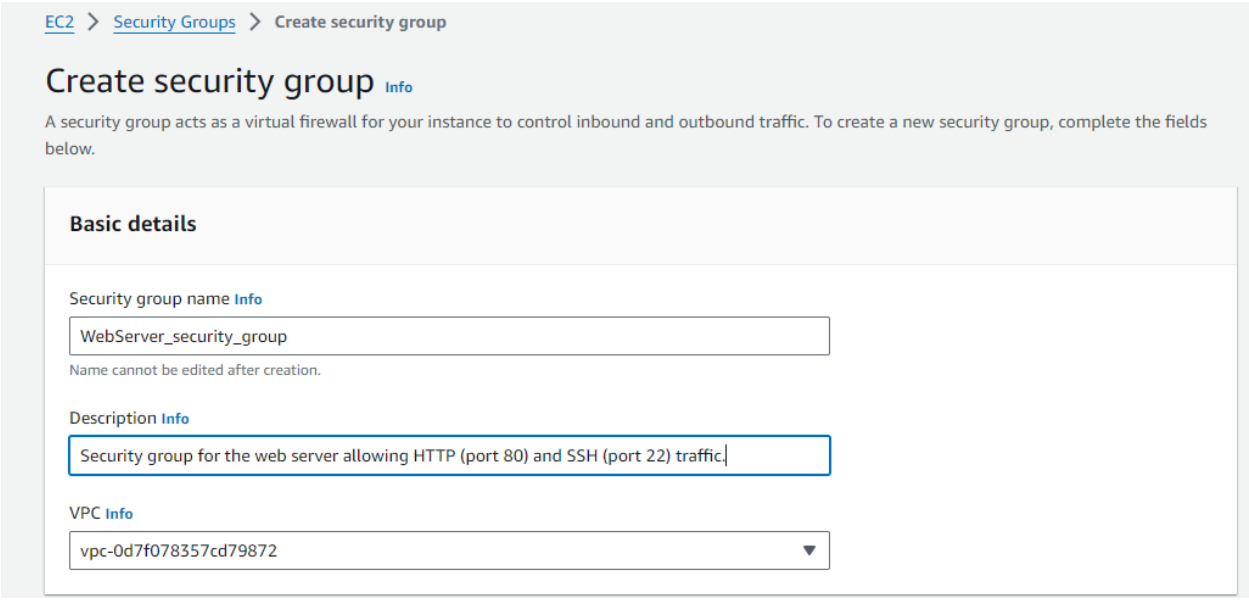
- Create one security group for the web server.
- Configure inbound rules for the web server security group to allow HTTP traffic (port 80) and SSH traffic (port 22) from any source.

SOLUTION:-

A) In AWS console, go to the EC2 dashboard.

B) Find ‘security groups’ and click on ‘create security group’.

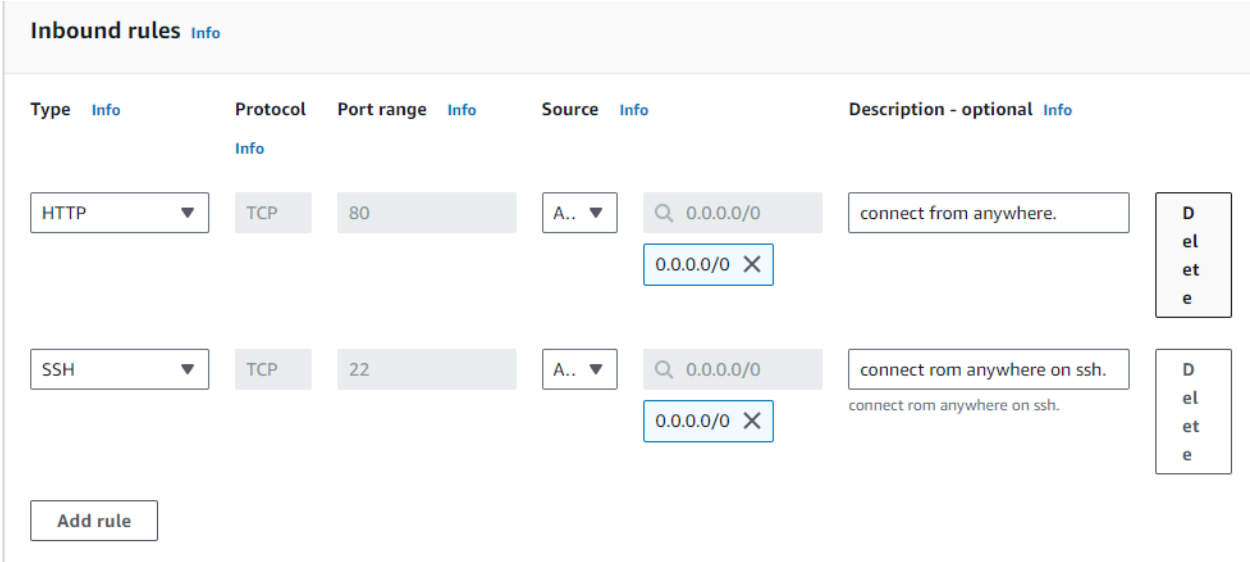
C) Then provide the name and description for your security group.
As shown below:-



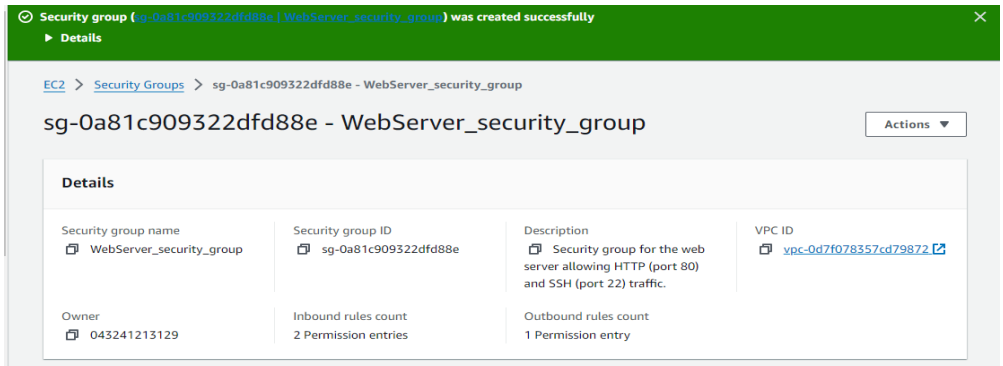
NOTE- the VPC section is chosen i.e. “by default” and description is mandatory.

D) For configuring inboud rules: go to inbound rules and click on add rules.

E) to allow HTTP traffic (port 80) and SSH traffic from any source:



F) Now the security group is configured to allow HTTP (80) and SSH (port 22) traffic from any source to your web-server.



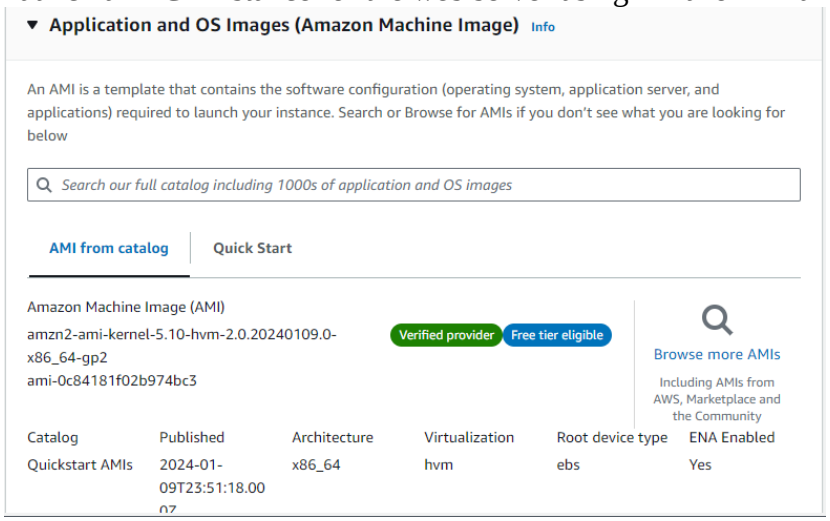
==========

Q**2. Launch EC2 Instance:**

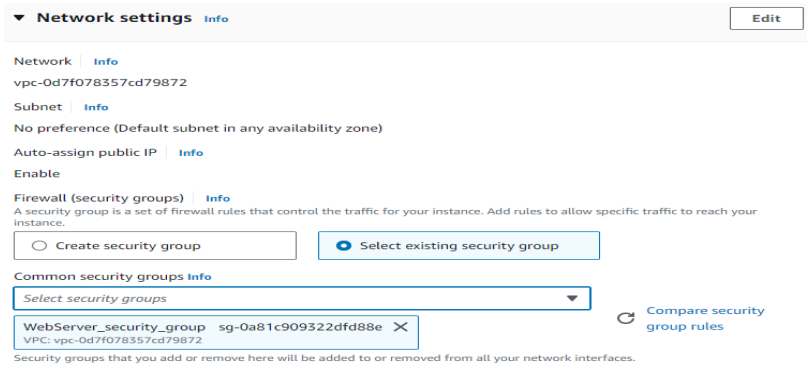
- Launch an EC2 instance for the web server using Amazon Linux 2 AMI.
- Associate the web server security group created earlier with this instance.
- Use an appropriate instance type for a web server.
- Ensure the instance has a public IP address.

SOLUTION:-

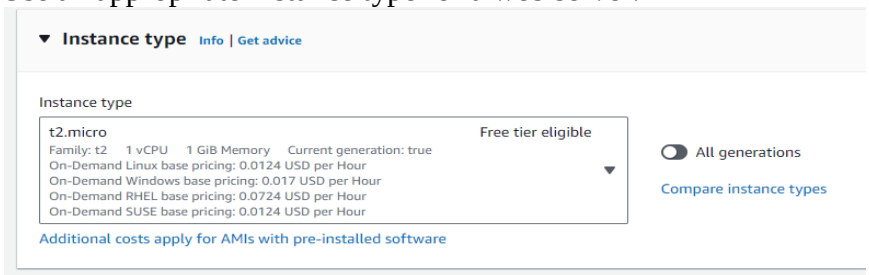
Launch an EC2 instance for the web server using Amazon Linux 2 AMI.



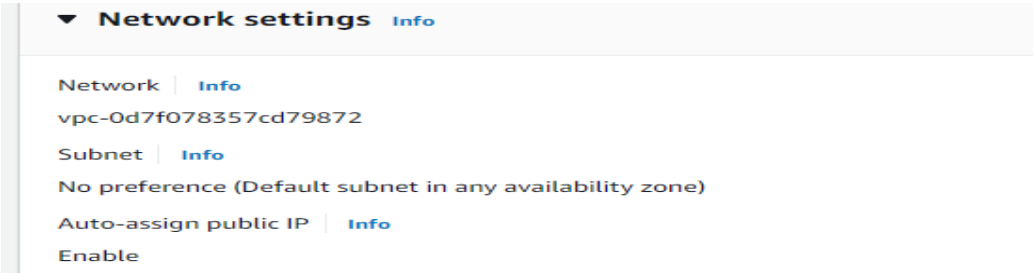
- Associate the web server security group created earlier with this instance:



Use an appropriate instance type for a web server.



Ensure the instance has a public IP address:-
Auto assign public IP is enabled.



==========

3. SSH Access:

- Generate an SSH key pair for secure access to the instances.
- Configure the web server instance to accept SSH connections using the generated key pair.
- Attempt to SSH into the web server instance to verify successful access.

SOLUTION:-

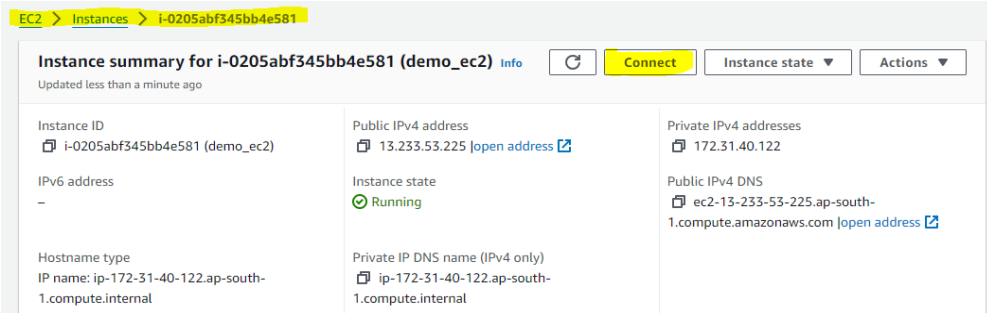
- Generate an SSH key pair for secure access to the instances.

<ssh-keygen>

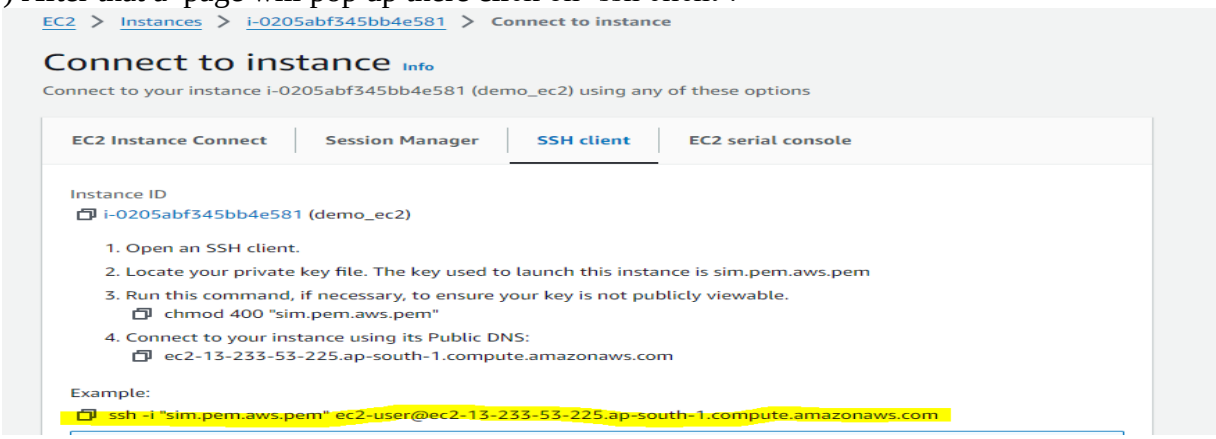
And you will find the the pub key in < /home/simpal/.ssh> (simpal is root)

```
root@DESKTOP-NJSOG33:simpal# cd .ssh

root@DESKTOP-NJSOG33:.ssh# ll

total 24

drwx------ 2 simpal simpal 4096 Jan  7 16:20 ./

drwxr-x--- 4 simpal simpal 4096 Jan 10 12:27 ../

-rw------- 1 simpal simpal 2610 Jan  7 16:16 id_rsa

-rw-r--r-- 1 simpal simpal  576 Jan  7 16:16 id_rsa.pub

-rw------- 1 simpal simpal  978 Jan  7 16:20 known_hosts

-rw-r--r-- 1 simpal simpal  142 Jan  7 16:06 known_hosts.old
```

- Configure the web server instance to accept SSH connections using the generated key pair.



A) Now, in here click on connect,
B) After that a page will pop up there click on 'ssh client'.



C) Now copy the command from this pop-up box to SSH.
D)Then go to your terminal locate the key ".pem" file on your terminal.

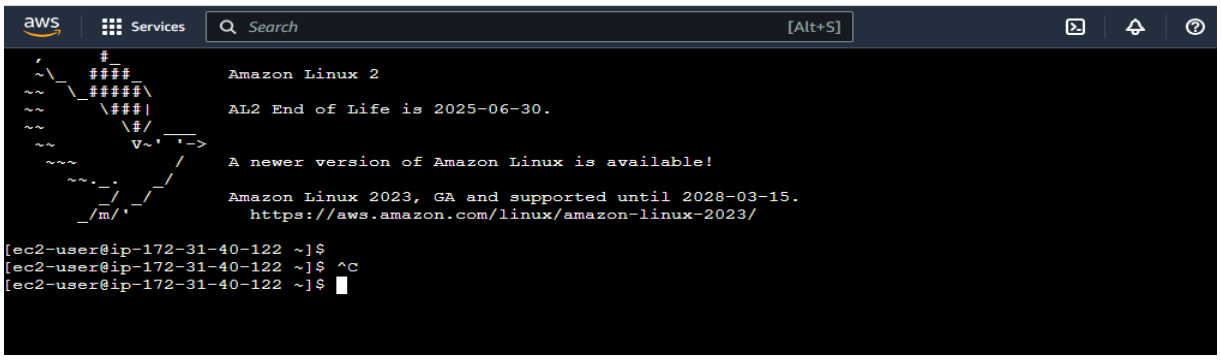
```
-rwxrwxrwx 1 simpal simpal       99 Jan 12 11:44  rootkey.csv*
-rwxrwxrwx 1 simpal simpal     1674 Jan 15 17:01  sim.pem.aws.pem*
-rwxrwxrwx 1 simpal simpal      125 Jan 11 11:15 'simpal_credentials (1).csv'*
-rwxrwxrwx 1 simpal simpal      162 Dec 21  2022 '~$kesh_Resume.docx'*
root@DESKTOP-NJSOG33:Downloads# pwd
/mnt/c/Users/kharg/Downloads
```

E) After that paste the entire command and try to access the ec2-user .

```
root@DESKTOP-NJSOG33:Downloads# ssh -i "sim.pem.aws.pem" ec2-user@ec2-13-233-53-225.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-233-53-225.ap-south-1.compute.amazonaws.com (13.233.53.225)' can't be established.
ED25519 key fingerprint is SHA256:B1DKGYOS7ppZUBpy47fY1EBdhRZ1RqPEYS/NugB/AXc.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:6: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-233-53-225.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Tue Jan 16 04:41:47 2024 from ec2-13-233-177-5.ap-south-1.compute.amazonaws.com
       #_
       ~\_  ####_        Amazon Linux 2
      ~~  \_#####\
      ~~     \###|       AL2 End of Life is 2025-06-30.
      ~~       \#/ ___
       ~~       V~' '->
        ~~~         /    A newer version of Amazon Linux is available!
          ~~._.   _/
             _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
           _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-40-122 ~]$
```

OR----------

A) After process no. ‘B’ we can click on ‘EC2-instance connect’ and then click on connect and there the below page will appear.



B) And after that we can manually add the local terminals ‘pub-key’ to the instance’s authorized keys and try to access from local terminal.

==========

or

4. Web Application Setup:

- Install a web server (e.g., Apache or Nginx) on the web server instance.
- Create a simple HTML page to confirm the web server is working.
- Test accessing the web server's public IP address in a web browser.

SOLUTION:-

A) Install a web server (e.g., Apache or Nginx) on the web server instance.

a) switching to root user run the commands:-

<yum update –y>

<yum install httpd –y>

<service httpd start>N

b) Now, enable the apache to start on boot.
<chkconfig httpd on>

c) After that we will create the simple HTML file, for eg (index.html):-
and we will create this file under the path </var/www/html/index.html> (in the context of a web server, this is a common location for serving static HTML files and it is also a default document root for apache on many linux distributions.
<vim /var/www/html/index.html >

d) And under this file write the below mentioned txt in as it is form.
< echo "<html><head><title>Web server Test</title></head><body><h1>web Server is working!</h1></body></html>"
And save the file.

- Test accessing the web server's public IP address in a web browser
a) First copy the public IP from instance.

b) Now go to your web browser and and on URL section write
http://your_instance_public.ip

The below output shows that the access to the browser is successful.

5. Documentation:
- Provide clear documentation outlining the steps you took to complete each task.

- Include relevant screenshots or command outputs to demonstrate the successful implementation of security groups, instance launches, and SSH access.

==========

(ON CLI)

Q 02.

1. Create Security Group for Web Server Using AWS CLI:

- Use the AWS CLI to create a security group for the web server.

- Configure inbound rules to allow HTTP traffic (port 80) and SSH traffic (port 22) from any source.

SOLUTION:-

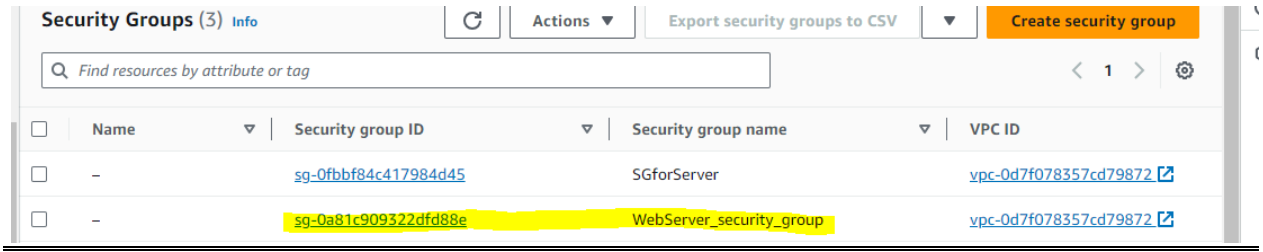
Use the AWS CLI to create a security group for the web server.

```
root@DESKTOP-NJSOG33:AWS# aws ec2 create-security-group --group-name SGforServer --description "security group for the web server htt
p and ssh traffic"
{
    "GroupId": "sg-0303c85e6ff9f9a42"
}
root@DESKTOP-NJSOG33:AWS#
```

- Configure inbound rules to allow HTTP traffic (port 80) and SSH traffic (port 22) from any source.

```
root@DESKTOP-NJSOG33:AWS# aws ec2 authorize-security-group-ingress --group-id  "sg-0303c85e6ff9f9a42" --protocol tcp --port 80 --cidr
 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0e46ba738eb7e6790",
            "GroupId": "sg-0303c85e6ff9f9a42",
            "GroupOwnerId": "043241213129",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
root@DESKTOP-NJSOG33:AWS#
```

```
    ]
}
root@DESKTOP-NJSOG33:AWS# aws ec2 authorize-security-group-ingress --group-id  "sg-0303c85e6ff9f9a42" --protocol tcp --port 22 --cidr
 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-07058f09aa135d98c",
            "GroupId": "sg-0303c85e6ff9f9a42",
            "GroupOwnerId": "043241213129",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
root@DESKTOP-NJSOG33:AWS#
```



==========

2. Launch EC2 Instance for Web Server Using AWS CLI:

- Use the AWS CLI to launch an EC2 instance for the web server using Amazon Linux 2 AMI.
  - Associate the security group created earlier with this instance.
  - Use an appropriate instance type for a web server.
  - Ensure the instance has a public IP address.

SOLUTION :-

.

root@DESKTOP-NJSOG33:AWS# aws ec2 run-instances --image-id ami-0d3f444bc76de0a79 --key-name sim.pem.aws --instance-type t2.micro --security-group-ids sg-0fbbf84c417984d45 --associate-public-ip-address --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Ec2

_Instance}]'

{

"Groups": [],

"Instances": [

{
    "AmiLaunchIndex": 0,
    "ImageId": "ami-0d3f444bc76de0a79",
    "InstanceId": "i-0db322a9016876d5c",
    "InstanceType": "t2.micro",
    "KeyName": "sim.pem.aws",
    "LaunchTime": "2024-01-16T18:24:07.000Z",
    "Monitoring": {
        "State": "disabled"
    },
    "Placement": {
        "AvailabilityZone": "ap-south-1a",
        "GroupName": "",
        "Tenancy": "default"
    },
    "PrivateDnsName": "ip-172-31-47-217.ap-south-1.compute.internal",
    "PrivateIpAddress": "172.31.47.217",
    "ProductCodes": [],
    "PublicDnsName": "",
    "State": {
        "Code": 0,
        "Name": "pending"
    },

"StateTransitionReason": "",
"SubnetId": "subnet-0b62104472025c636",
"VpcId": "vpc-0d7f078357cd79872",
"Architecture": "x86_64",
"BlockDeviceMappings": [],
"ClientToken": "8c7c8d12-2fdb-434b-84ff-2eaa30434bd9",
"EbsOptimized": false,
"EnaSupport": true,
"Hypervisor": "xen",
"NetworkInterfaces": [
  {
    "Attachment": {
      "AttachTime": "2024-01-16T18:24:07.000Z",
      "AttachmentId": "eni-attach-000571f1585794bdf",
      "DeleteOnTermination": true,
      "DeviceIndex": 0,
      "Status": "attaching",
      "NetworkCardIndex": 0
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "SGforServer",

```
            "GroupId": "sg-0fbbf84c417984d45"
        }
    ],
    "Ipv6Addresses": [],
    "MacAddress": "02:87:92:7f:68:ff",
    "NetworkInterfaceId": "eni-051e9b474e65f854e",
    "OwnerId": "043241213129",
    "PrivateDnsName": "ip-172-31-47-217.ap-south-1.compute.internal",
    "PrivateIpAddress": "172.31.47.217",
    "PrivateIpAddresses": [
        {
            "Primary": true,
            "PrivateDnsName": "ip-172-31-47-217.ap-south-1.compute.internal",
            "PrivateIpAddress": "172.31.47.217"
        }
    ],
    "SourceDestCheck": true,
    "Status": "in-use",
    "SubnetId": "subnet-0b62104472025c636",
    "VpcId": "vpc-0d7f078357cd79872",
    "InterfaceType": "interface"
  }
],
```

"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupName": "SGforServer",
        "GroupId": "sg-0fbbf84c417984d45"
    }
],
"SourceDestCheck": true,
"StateReason": {
    "Code": "pending",
    "Message": "pending"
},
"Tags": [
    {
        "Key": "Name",
        "Value": "Ec2\n_Instance"
    }
],
"VirtualizationType": "hvm",
"CpuOptions": {
    "CoreCount": 1,
    "ThreadsPerCore": 1

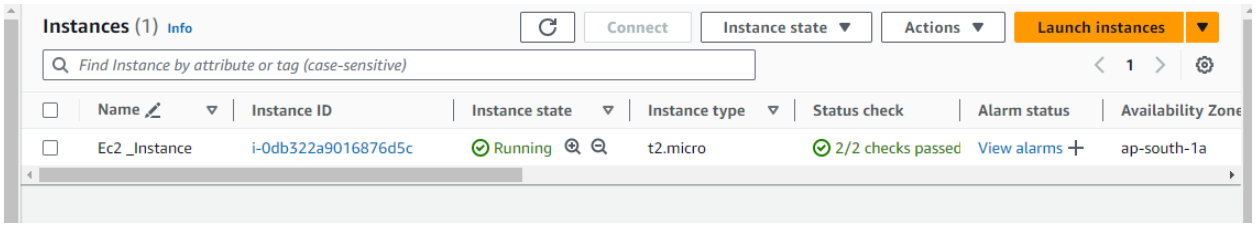
```
        },
        "CapacityReservationSpecification": {
            "CapacityReservationPreference": "open"
        },
        "MetadataOptions": {
            "State": "pending",
            "HttpTokens": "required",
            "HttpPutResponseHopLimit": 2,
            "HttpEndpoint": "enabled",
            "HttpProtocolIpv6": "disabled",
            "InstanceMetadataTags": "disabled"
        },
        "EnclaveOptions": {
            "Enabled": false
        },
        "BootMode": "uefi-preferred",
        "PrivateDnsNameOptions": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        }
    }
],
```

"OwnerId": "043241213129",

"ReservationId": "r-05f49d62357faf709"

}

root@DESKTOP-NJSOG33:AWS#



==========

3. SSH Access Using AWS CLI:

- Use the AWS CLI to generate an SSH key pair for secure access to the web server instance.
- Configure the web server instance to accept SSH connections using the generated key pair.
- Use the AWS CLI to attempt to SSH into the web server instance to verify successful access.

root@DESKTOP-NJSOG33:AWS# aws ec2 help

root@DESKTOP-NJSOG33:AWS# aws ec2 create-key-pair helo

To see help text, you can run:

aws help

aws <command> help

aws <command> <subcommand> help

usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]

aws: error: the following arguments are required: --key-name

root@DESKTOP-NJSOG33:AWS# aws ec2 create-key-pair help

```
root@DESKTOP-NJSOG33:AWS# aws ec2  create-key-pair  --key-name sim.pem.aws



An error occurred (InvalidKeyPair.Duplicate) when calling the CreateKeyPair operation: The
keypair already exists

root@DESKTOP-NJSOG33:AWS# aws ec2  create-key-pair  --key-name sim.new.key

{

    "KeyFingerprint": "94:8e:61:1f:d4:8f:cd:61:67:ce:8e:de:5e:37:ad:53:c5:18:f4:88",

    "KeyMaterial": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAi8C1ywiAURen4p3ndjefPGnKllPqTm14R5CHZNzljfploEb+\nXC1j
WTBTZxMKoYBgZ3OkeZgHeXEgkkSvOmXee6zE/tLAzf2eDfFOPGvCrSoq9fUj\n5Phb2rau/
P0wUd7DBjAlMu7UDURRjL0jFM9WNj4levzAVR7IwJi/RcEiqy6vxOqR\nItriKAdR9uXfkSB
O3WzWoFbEKvlDv02baAl4pmx/owHJN8cPz2aMvbL0s9AdCLc4\n58uWz0zdfgREcu/C/0vpZ
NDavwafMMmr6WRqcfjZEvk5vIff9RVCWAFygab1G7rE\ny/sJX0++gTqfP8ozcf+2FVjOfQ33
922/Tu1PIwIDAQABAoIBABaHcYcSjsUD4D+r\nHvYyz0vI7iy/yGTuRtaamQkMh0EVHa7x4
u1vL7XgEYHrUupoLKJgxSII/SN5tjt+\nMRVj+LLNTlCaHxTWDtXGkcsxwAd0ZcqSwz+VPh
GLI7iLBHAeTihAOK72S178JX1f\nvDkNl6NU72vdNRBDzy8lMKiJExKvTLFeFROVsMFov
GjRc3FEQSJn1BGN/9h5RC5b\nOxQhsCuYXllG0E2LO4Qq5rZTdwru7W0wfMG9uPToDPfX
nQ+oigKC/l0vgYFQT6ZR\nOD29PSljvVm299lixvXa26vs8r5rnSDwZMZHqTiRJfV3LLiUCK
ozfBJUivnpB2uH\nEOZrzFkCgYEA6b9khCoZnHWCncVeWsXq7WiRmkNMfDmyRxzvVEEI
CejCEWpFFT52\n5ULun/TShxqcFZ0I/bNmpbO12hePMqwQDPOia9+wWrCed3e5UIyvVhwRl
7anD+GI\n7M9GLpNPwdUoRdSxSiWmVGCROgLvl70esg7HsDwtrWKCS45gV8o0onUCgY
EAmQ6X\nadtkiDutiZUANO93Ij8EIEtyZEf/RYe0SYrd837SY4UoWNtZyNDKJLp6rGU6g6A
q\nuy/gMBGbR+2OE//f7+6b7Iq6OpcruVwkMEGbAmOxcdXz3gEF9+fnOQQ8o29NlFfw\nBP
GU6CpTpXTvy5fgwlhV5JuhlkYKP3qPP2eHyDcCgYBzGlxU+KZD9Vmsd1RPPsbA\nwY1xe
VJgmcjAW+8+fgeHWaa3DK2YGHpTyvHWsqg4/1F9EycqRv10+1nBW3iYa0I8\nHn0MwcoF
3pMqITqP/7cXoBrJqpf5qgXTFv5oUQIlYOHEAUiMSp3tTuA0wN1ayzYi\nWETc88VKbAfdT
O8ES/4/QQKBgCq+z3yp4A7IE/Qrn84o3q26ya03RPBKxzkk6C3t\n0YTSc3GF27nNMOsLnJjb
V61T7B6cmbKDcWobULGq35YoTONWUUejitUYKQbwe87a\nU688Jm9zYFMtF/yfUI006La
APDtkv8yxp3Obdpbr7JiAhy3fu8VHzb4JO4107hK4\nGEzLAoGBALchVDSJGx99f00YVI0EE
XBGfh35BDqoaL26TbJ69fgyFEX7YC6fhRmp\nkP/L3Rick6FquFR57FuKDIuezZz7jtkS81eMk
B6W1RPG+GJORT4bQRAZJECbK0eT\nf49SwOJqLoEIqIA8tPYfSob0NNAFwr+fdc+vcTqyj
Pt84gGfm/nH\n-----END RSA PRIVATE KEY-----",

    "KeyName": "sim.new.key",

    "KeyPairId": "key-018ae4132332581ac"

}
```

```
root@DESKTOP-NJSOG33:AWS# aws ec2 describe-instances --instance-ids i-
0db322a9016876d5c --query 'Reservations[0].Instances[0].[InstanceId,PublicIpAddress]' --
output text

i-0db322a9016876d5c     3.110.182.200



root@DESKTOP-NJSOG33:~# cd  /mnt/c/kharg/Downloads/

root@DESKTOP-NJSOG33:Downloads# ll

total 12520

drwxrwxrwx 1 simpal simpal     4096 Jan 15 17:14  ./

drwxrwxrwx 1 simpal simpal     4096 Jan 15 13:31  ../

-rwxrwxrwx 1 simpal simpal 10542392 Jan 15 16:09 '12 January 2024 aws session.pdf'*

-rwxrwxrwx 1 simpal simpal  2174868 Jan 10 12:39  DOCKER_NOTES_PDF.pdf*

-rwxrwxrwx 1 simpal simpal    95524 Jan 15 17:14  Invoice_1543754689.pdf*

-rwxrwxrwx 1 simpal simpal      282 Apr 19  2022  desktop.ini*

-rwxrwxrwx 1 simpal simpal       99 Jan 12 11:44  rootkey.csv*

-rwxrwxrwx 1 simpal simpal     1674 Jan 15 17:01  sim.pem.aws.pem*

-rwxrwxrwx 1 simpal simpal      125 Jan 11 11:15 'simpal_credentials (1).csv'*

-rwxrwxrwx 1 simpal simpal      162 Dec 21  2022 '~$kesh_Resume.docx'*

root@DESKTOP-NJSOG33:Downloads# chmod 400 sim.pem.aws.pem



root@DESKTOP-NJSOG33:Downloads# ssh -i sim.pem.aws.pem  ec2-user@3.110.182.200

The authenticity of host '3.110.182.200 (3.110.182.200)' can't be established.

ED25519 key fingerprint is
SHA256:rK3n9BkKKzy1t9kg0R6Mk6oCScwrTZSNnVWarWGSiYY.

This key is not known by any other names
```

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added '3.110.182.200' (ED25519) to the list of known hosts.

  ,     #_

  ~\_  ####_        Amazon Linux 2023

 ~~  \_#####\

 ~~     \###|

 ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023

  ~~       V~' '->

   ~~~         /

     ~~._.   _/

        _/ _/

      _/m/'

[ec2-user@ip-172-31-47-217 ~]$ pwd

/home/ec2-user

[ec2-user@ip-172-31-47-217 ~]$
```

---

4. Web Application Setup Using AWS CLI:

- Use the AWS CLI to install a web server (e.g., Apache or Nginx) on the web server instance.
- Create a simple HTML page using the AWS CLI to confirm the web server is working.
- Use the AWS CLI to test accessing the web server's public IP address in a web browser.

SOLUTION:-

As we have already ssh the ec2-instance.

After that,

```
[root@ip-172-31-47-217 ec2-user]# yum update -y

Last metadata expiration check: 0:30:59 ago on Tue Jan 16 18:24:58 2024.

Dependencies resolved.
```

```
Nothing to do.

Complete!

[root@ip-172-31-47-217 ec2-user]# yum install -y nginx

Last metadata expiration check: 0:31:32 ago on Tue Jan 16 18:24:58 2024.

Dependencies resolved.

================================================================================================

 Package                    Architecture       Version                        Repository            Size

================================================================================================

Installing:

 nginx                      x86_64             1:1.24.0-1.amzn2023.0.2        amazonlinux           32
k

Installing dependencies:

 generic-logos-httpd        noarch             18.0.0-12.amzn2023.0.3         amazonlinux
19 k

 gperftools-libs            x86_64             2.9.1-1.amzn2023.0.3           amazonlinux
308 k

 libunwind                  x86_64             1.4.0-5.amzn2023.0.2           amazonlinux
66 k

 nginx-core                 x86_64             1:1.24.0-1.amzn2023.0.2        amazonlinux
586 k

 nginx-filesystem           noarch             1:1.24.0-1.amzn2023.0.2        amazonlinux
9.1 k

 nginx-mimetypes            noarch             2.1.49-3.amzn2023.0.3          amazonlinux
21 k


Transaction Summary
```

```
================================================================================
==================================================

Install  7 Packages


Total download size: 1.0 M

Installed size: 3.4 M

Downloading Packages:

(1/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                 902 kB/s |  66 kB
00:00

(2/7): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm                    349 kB/s |  32 kB
00:00

(3/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm       1.4 MB/s |  19
kB     00:00

(4/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.noarch.rpm          520 kB/s | 9.1 kB
00:00

(5/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm            5.2 MB/s | 308 kB
00:00

(6/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm            1.4 MB/s |  21
kB     00:00

(7/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm                3.7 MB/s | 586 kB
00:00

--------------------------------------------------------------------------------
---

Total                                           4.9 MB/s | 1.0 MB     00:00

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.
```

Complete!

[root@ip-172-31-47-217 ec2-user]# service nginx start

Redirecting to /bin/systemctl start nginx.service

[root@ip-172-31-47-217 ec2-user]# chkconfig nginx on

Note: Forwarding request to 'systemctl enable nginx.service'.

Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.

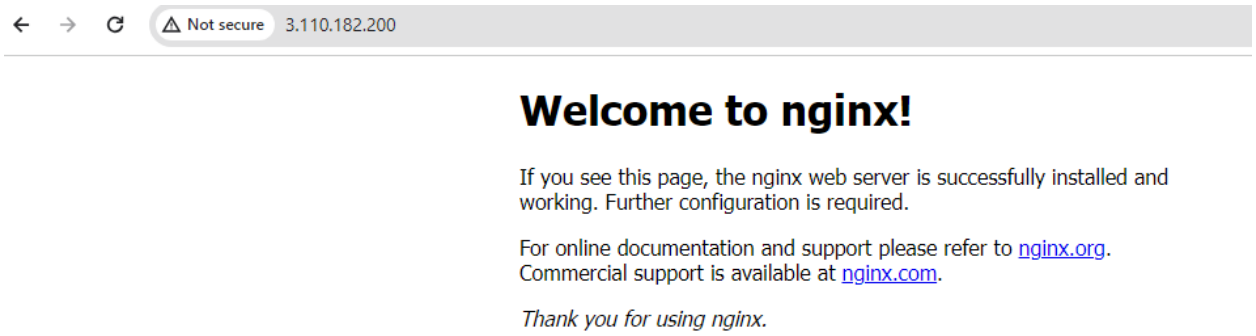
[root@ip-172-31-47-217 ec2-user]# systemctl enable nginx.service

[root@ip-172-31-47-217 ec2-user]# vim /usr/share/nginx/html/index.html

Under vim file write :-

```
echo "<html><body><h1>Welcome to nginx!</h1></body></html>"
```

And after going to any web server
Give the public IP on URL section.
The output:-



5. Documentation:

- Provide clear documentation in a text file outlining the AWS CLI commands used for each task along with their outputs.
- Include any relevant information such as IP addresses, instance IDs, etc.