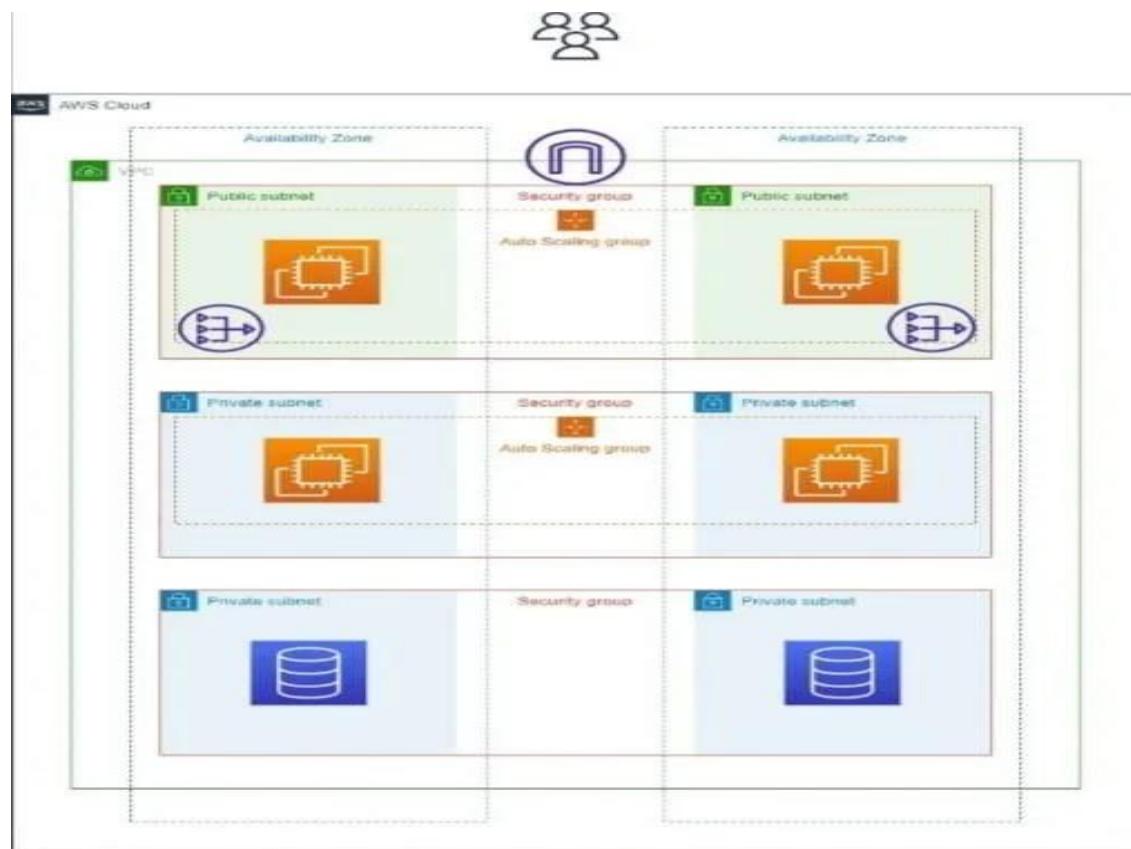


Name: Chekka Rajesh
Mail-ID: rajesh.nani555@gmail.com
Topic: AWS-Load balancer, Autoscaling, VPC, RDS
Date of submission: 08th-March-2024

Batch no: 115
Course: DevOps
Trainer Name: Mr. Madhukar sir
Project No: 04 (AWS-3-Tier)

AWS-3-Tier-Project

Implementation of below diagram



- **US East (N. Virginia)us-east-1**

Step-1: Create VPC

AWSConsole-vpc-create vpc-Name(VPC-1)-IPv4 CIDR(11.0.0.0/16)-Create vpc.

Now VPC created

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with navigation options like 'Your VPCs', 'Subnets', 'Route tables', etc. The main area displays a success message: 'You successfully created vpc-0c633671d8f910d0f / VPC-1'. Below this, the VPC details are shown in a table:

VPC ID	vpc-0c633671d8f910d0f	State	Available
Tenancy	Default	DHCP option set	dopt-098ada3df315b3664d
Default VPC	No	IPv4 CIDR	11.0.0.0/16
Network Address Usage metrics	Disabled	Route 53 Resolver DNS Firewall rule groups	-
		Owner ID	926074783450

Below the table, there are tabs for 'Resource map', 'CIDRs', 'Flow logs', 'Tags', and 'Integrations'. At the bottom, there are buttons for 'VPC Show details', 'Subnets (0)', 'Route tables (1)', and 'Netw'. The footer includes copyright information and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Step-2: Create & Attach Internet Gateway to the VPC-1

VPC-Internet Gateway-Create Internet Gateway-Name(My-IGW)-Create Internet Gateway. Actions-attach to VPC-Select a VPC(vpc-0c633671d8f910d0f(VPC-1))-Attach Internet Gateway.

Step-2: Create 2-Public subnets in different Availability zones

1. public-SN-1(us-east-1a)
2. public-SN-2 (us-east-1b)

VPC-Subnet-Create Subnet-Select VPC ID (vpc0c633671d8f910d0f VPC-1)-name(**public-SN-1**)-AZ (**us-east-1a**)-subnet cidr block- **11.0.1.0/24**-create subnet.

VPC-Subnet-Create Subnet-Select VPC ID (vpc0c633671d8f910d0f VPC-1)-name(**public-SN-2**)-AZ (**us-east-1b**)-subnet cidr block- **11.0.2.0/24**-create subnet.

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-0908055b0d0e64bdc	Available	vpc-03c2f0128f050fd80	172.31.80.0/20
-	subnet-04c4fcfb7014a345e	Available	vpc-03c2f0128f050fd80	172.31.32.0/20
-	subnet-090f1460dd94ee939	Available	vpc-03c2f0128f050fd80	172.31.48.0/20
-	subnet-07a3d90edaef9ea0	Available	vpc-03c2f0128f050fd80	172.31.16.0/20
-	subnet-01b3528b01bf15a22	Available	vpc-03c2f0128f050fd80	172.31.64.0/20
-	subnet-0b467eac06d01e160	Available	vpc-03c2f0128f050fd80	172.31.0.0/20
public-SN-1	subnet-073cb3263cb803b74	Available	vpc-0c633671d8f910d0f VPC-1	11.0.1.0/24
public-SN-2	subnet-08b536df86f440208	Available	vpc-0c633671d8f910d0f VPC-1	11.0.2.0/24

Step-3: Create Route table one end connected to the IGW & Another end connected to the public-SN-1.

VPC-Route tables-Create Route table-Name(My-RT-1)-Select VPC(vpc0c633671d8f910d0f (VPC-1))-Create route table.

Select My-RT-1-Edit routes-destination (0.0.0.0/0)-IGW- My-IGW-Save changes-edit subnet associations-select public-SN-1-Save associations.

Step-4: Create Route table one end connected to the IGW & Another end connected to the public-SN-2.

VPC-Route Tables-Create Route Table-Name (My-RT-2)-Select VPC (vpc0c633671d8f910d0f (VPC-1))-Create route table.

Select My-RT-2-Edit routes-destination (0.0.0.0/0)-IGW- My-IGW-Save changes-edit subnet associations-select public-SN-2-Save associations.

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with navigation links for VPC dashboard, EC2 Global View, Filter by VPC (with a dropdown menu), Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, Peering connections), and Security (Network ACLs). The main area is titled "Subnets (8) Info" and contains a table with 8 rows. The columns are: Name, Subnet ID, State, VPC, IPv4 CIDR, IPv6 CIDR, Available IP range, Availability zone, Network interface count, and Route table. The subnets are grouped under two route tables: "rtb-055d8b58dd9403c0c | My-RT-2" and "rtb-09724b1dc75c5c054 | My-RT-1".

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IP range	Availability zone	Network interface count	Route table
public-SN-2	subnet-0c6...	Available	vpc-0c6...	11.0.2.0/24	-	251	us-east-1b	use1-az4	-
public-SN-1	subnet-0c6...	Available	vpc-0c6...	11.0.1.0/24	-	251	us-east-1a	use1-az2	-
	subnet-0c6...	Available	vpc-03c...	172.31.8...	-	4091	us-east-1a	use1-az2	-
	subnet-0c6...	Available	vpc-03c...	172.31.3...	-	4091	us-east-1c	use1-az6	-
	subnet-0c6...	Available	vpc-03c...	172.31.4...	-	4091	us-east-1e	use1-az3	-
	subnet-0c6...	Available	vpc-03c...	172.31.1...	-	4091	us-east-1b	use1-az4	-
	subnet-0c6...	Available	vpc-03c...	172.31.6...	-	4091	us-east-1f	use1-az5	-
	subnet-0c6...	Available	vpc-03c...	172.31.0...	-	4091	us-east-1d	use1-az1	-

Step-5: Create two Security group name SG-1 & SG-2 & allow SSH & HTTP Ports.

EC2-Network and security-security groups-Create security group-Name (**SG-1**)-Description (Security ssh-http)-Select VPC (VPC-1)-edit inbound rules-add rules-ssh&http ports-create security group.

EC2-Network and security-security groups-Create security group-Name (**SG-2**)-Description (Security ssh-http)-Select VPC (VPC-1)-edit inbound rules-add rules-ssh&http ports-create security group.

The screenshot shows the AWS Security Groups page. On the left, there's a sidebar with navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code (Preview), Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main area is titled "sg-05eadf0260fa29c3b - SG-1" and contains a "Details" section with fields: Security group name (SG-1), Security group ID (sg-05eadf0260fa29c3b), Description (Security ssh-http), and VPC ID (vpc-0c633671d8f910d0f). Below this is an "Inbound rules" section with a table showing two entries: one for port 80 (HTTP) and one for port 22 (SSH).

Inbound rules (2)						
	Name	Security group rule...	IP version	Type	Protocol	Port range
	-	sgr-0bd1097f2418f6b99	IPv4	HTTP	TCP	80
	-	sar-04278ec830d36cac4	IPv4	SSH	TCP	22

Step-6: Create 2-EC2-Instances in different Availability zones

- 1.** public-Ins-1(us-east-1a)
- 2.** public-Ins-2(us-east-1b)

EC2-Launch instance-AMI (ubuntu)-Instance type(t2.micro)-Create new keypair-edit network settings-vpc(vpc-1)-subnet(**public-SN-1**)-Auto assign public ip(Enable)-select existing security group-select(**SG-1**)-Launch instance.

EC2-Launch instance-AMI (ubuntu)-Instance type(t2.micro)-Create new keypair-edit network settings-vpc(vpc-1)-subnet(**public-SN-2**)-Auto assign public ip(Enable)-select existing security group-select(**SG-2**)-Launch instance.

Now connect both instances install apache2 and run web page as a public-Ins-1 & public-Ins-2.

Sudo -i

apt-get update -y

apt-get install apache2 -y

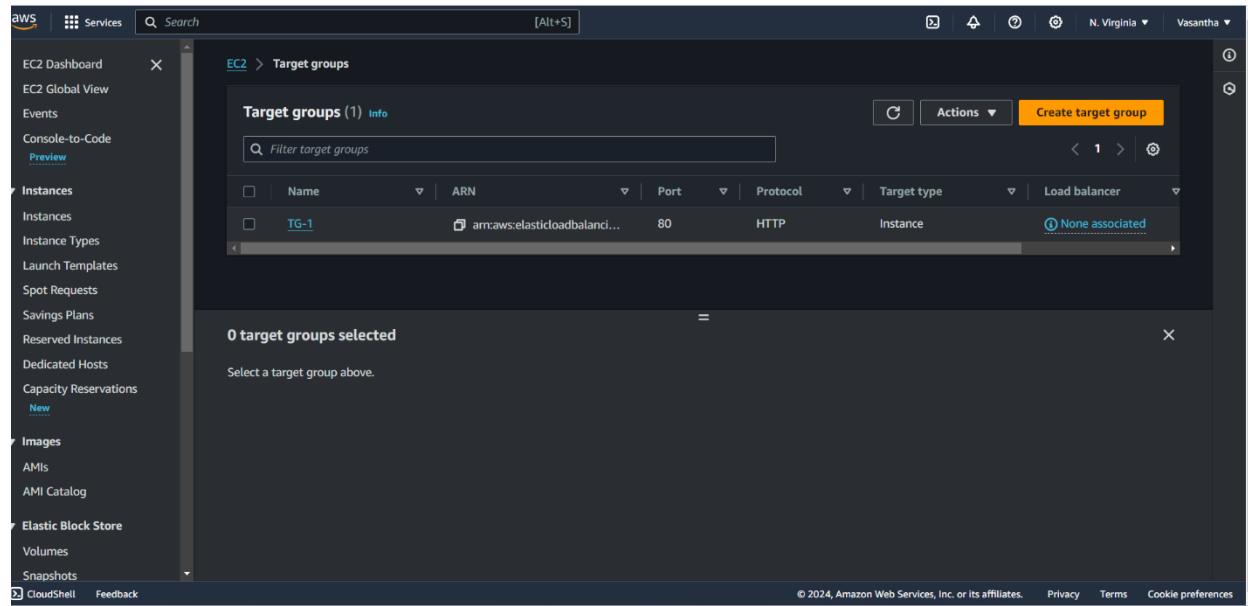
echo "public-Ins-1"> /var/www/html/index.html **(or) another instance echo "public-Ins-2"> /var/www/html/index.html**

The screenshot shows the AWS EC2 Instances page. The left sidebar has a tree view with 'Instances' selected, under which 'Instances' is also selected. The main content area displays a table titled 'Instances (2) Info'. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. There are two rows: 'public-Ins-1' (Instance ID: i-0a57e05a66d994b41, State: Running, Type: t2.micro, Checks: 2/2 passed, Zone: us-east-1a, IP: -) and 'public-Ins-2' (Instance ID: i-0b8cdf3b60d855d3, State: Running, Type: t2.micro, Checks: 2/2 passed, Zone: us-east-1b, IP: -). Below the table, a modal window titled 'Select an instance' is open, showing the same two instances. The bottom right corner of the modal has a close button (X).

Step-7: Create auto scaling group for public-Ins-1 & public-Ins-2.

Create target groups:

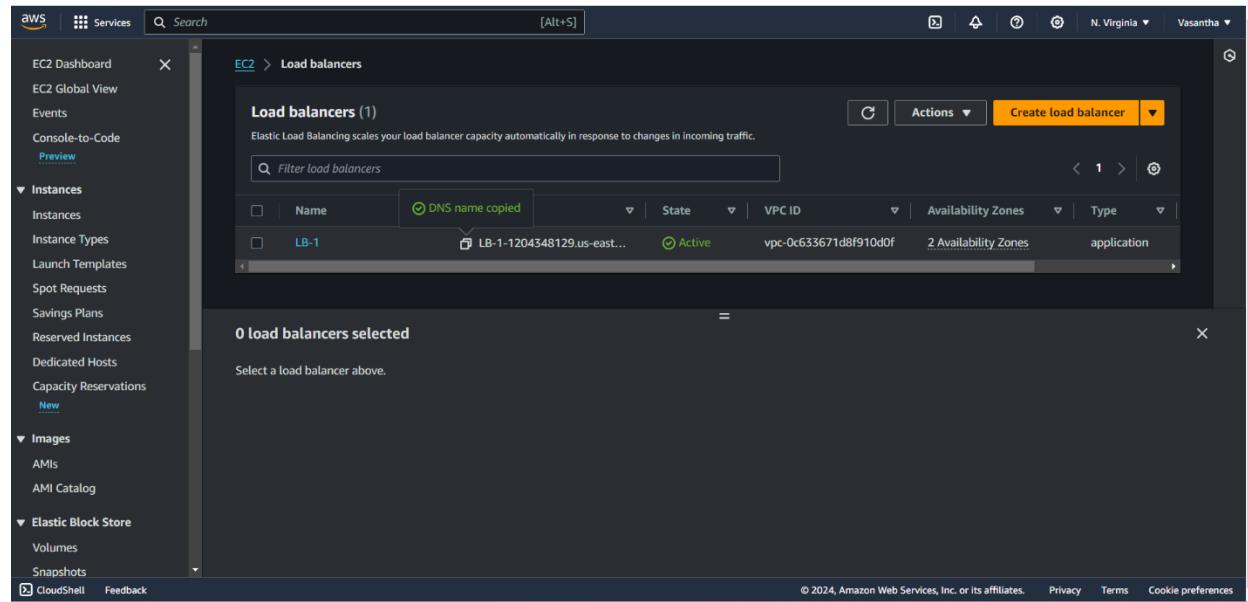
EC2-Load balancing-Target group-create targert group-instances-Name (TG-1)-select VPC(VPC-1)-next-select two instances (public-Ins-1 & public-Ins-2)-Including as pending below-create target group.



The screenshot shows the AWS EC2 Target Groups page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images, and Elastic Block Store. The main content area shows a table for 'Target groups (1)'. The table has columns for Name, ARN, Port, Protocol, Target type, and Load balancer. One row is listed: TG-1, arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/TG-1, 80, HTTP, Instance, and None associated. Below the table, a modal window titled '0 target groups selected' displays the message 'Select a target group above.'

Create Load balancer:

EC2-Load balancer-create load balancer-application load balancer-create-name (LB-1)-Select VPC(VPC-1)-mappings-us-east-1a&us-east-1b-select security group-SG-1-Select target group (TG-1)-create load balancer.



The screenshot shows the AWS EC2 Load Balancers page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images, and Elastic Block Store. The main content area shows a table for 'Load balancers (1)'. The table has columns for Name, State, VPC ID, Availability Zones, and Type. One row is listed: LB-1, Active, vpc-0c633671d8f910d0f, 2 Availability Zones, application. A status message 'DNS name copied' is shown next to the Name column. Below the table, a modal window titled '0 load balancers selected' displays the message 'Select a load balancer above.'

Now copy the load balancer DNS Name and access through google. The output is **public-Ins-1** And then refresh again the output is **public-Ins-2**.

Create Image:

Select public-Ins-1-Instance-Actions-Image and templates-create image-name (Image-1)-description(image)-create image.

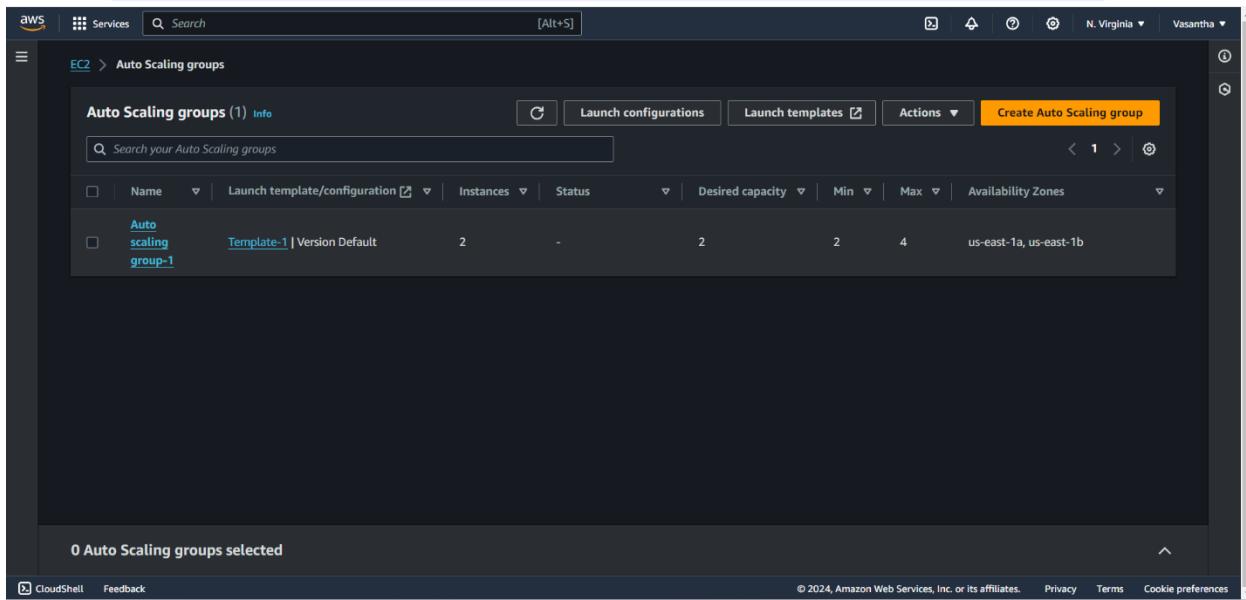
The screenshot shows the AWS Management Console with the left sidebar expanded. Under the 'Images' section, 'AMIs' is selected. In the main content area, the 'Amazon Machine Images (AMIs)' table shows one entry: 'Image-1' with AMI ID 'ami-02f221c7889ee050'. Below the table, a modal window titled 'Select an AMI' is open, displaying the same information. The bottom right corner of the modal shows the copyright notice '© 2024, Amazon Web Services, Inc. or its affiliates.'

Launch template: -EC2-Launch template-Create launch template-name (Template-1)-description (LT)-ami-my ami-owned by me-select ami(image-1)-Instance type(t2.micro)-key pair name (select which you want)-select existing security group (SG-1)-Create launch template.

The screenshot shows the AWS Management Console with the left sidebar expanded. Under the 'Instances' section, 'Launch Templates' is selected. In the main content area, the 'Launch Templates' table shows one entry: 'Template-1' with Launch Template ID 'lt-0ae2af2ec9fe9ce01'. Below the table, a modal window titled 'Select a launch template' is open, displaying the same information. The bottom right corner of the modal shows the copyright notice '© 2024, Amazon Web Services, Inc. or its affiliates.'

Create Autoscaling Groups: -

EC2-Autoscaling-Autoscaling Groups-create autoscaling groups-Autoscaling group-1-select launch template(Template-1)-Next-select vpc(VPC-1)-Availability zones and subnets(us-east-1a,us-east-2,public-SN-1,public-SN-2)-next-attach to an existing load balancer-Choose from your load balancer target groups-existing load balancer target groups(TG-1)-Health check grace period(100sec)-next-desired capacity:2-minimum desired capacity:2-Maximum desired capacity:4-Automatic scaling-target tracking scaling policy-name(target tracking policy)-Metric typ(Average cpu utilization)-Target value(50)-instance warmup(100sec)-Next-Next-Next-create autoscaling group.



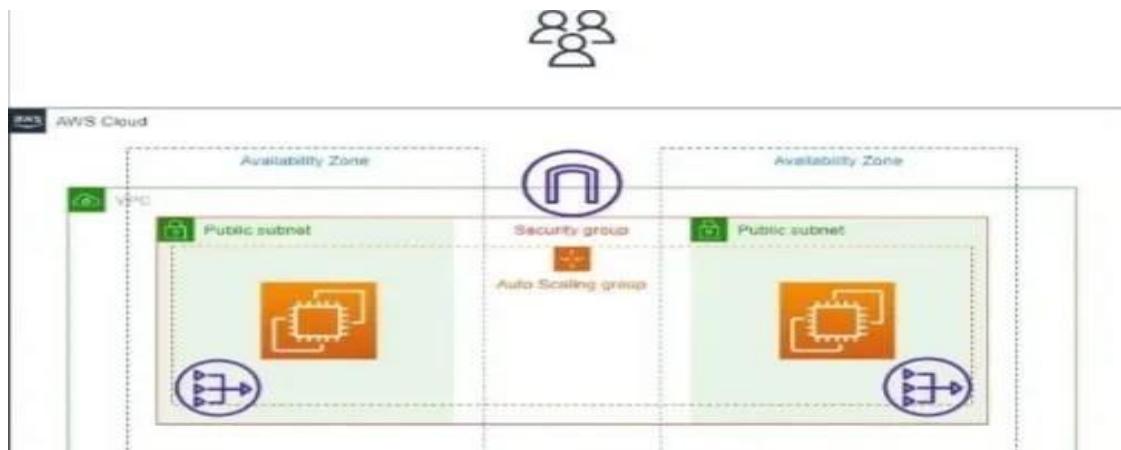
The screenshot shows the AWS Auto Scaling Groups page. At the top, there are tabs for 'Launch configurations', 'Launch templates', and 'Actions'. A prominent yellow button labeled 'Create Auto Scaling group' is visible. Below the buttons is a search bar and a table header with columns: Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones. The table contains one row for 'Auto scaling group-1', which is associated with 'Template-1 | Version Default', has 2 instances, and is set to a minimum of 2 and a maximum of 4 across the availability zones 'us-east-1a, us-east-1b'. At the bottom of the table, it says '0 Auto Scaling groups selected'.

Now click on instances there is 2-New EC2-Instances creating because of maximum desired capacity is 4.

Instances (4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	i-06b52d53d58b16d57	Running	t2.micro	Initializing	View alarms +	us-east-1b	-
	i-003ed384d04d8e505	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
	i-0f057146c0faa7c20	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-
	i-088471e2c3623b1f0	Running	t2.micro	Initializing	View alarms +	us-east-1a	-

Select an instance



Now this part completed. (NAT Not created because of the purpose of NAT is to connect Private-Ins-1&2 from Public-Ins-1&2)

Step-8: Create 2-Private subnets in different Availability zones

- 1.** private-SN-1(us-east-1a)
- 2.** private-SN-2(us-east-1b)

VPC-Subnet-Create subnet-Select VPC ID (vpc0c633671d8f910d0f
VPC-1)-name(**private**

-SN-1-AZ(us-east-1a)-subnet cidr block- 11.0.3.0/24-create subnet.

VPC-Subnet-Create subnet-Select VPC ID (vpc0c633671d8f910d0f)
VPC-1)-name(**private-SN-2**)-AZ(**us-east-1b**)-subnet cidr block- **11.0.4.0/24**-create subnet.

Name	Subnet	State	VPC	IPv4 Range	IPv6 Range	Available AZs	Available AZs	Net. Range	Route table
private-SN-1	subnet-...	Available	vpc-0c6...	11.0.3.0/24	-	251	us-east-1a	use1-az2	-
private-SN-2	subnet-...	Available	vpc-0c6...	11.0.4.0/24	-	251	us-east-1b	use1-az4	-
public-SN-2	subnet-...	Available	vpc-0c6...	11.0.2.0/24	-	248	us-east-1b	use1-az4	rtb-055d8b58dd94c
public-SN-1	subnet-...	Available	vpc-0c6...	11.0.1.0/24	-	248	us-east-1a	use1-az2	rtb-09724b1dc75c3
-	subnet-...	Available	vpc-05c...	172.31.8.0/24	-	4091	us-east-1a	use1-az2	-
-	subnet-...	Available	vpc-05c...	172.31.3.0/24	-	4091	us-east-1c	use1-az6	-
-	subnet-...	Available	vpc-05c...	172.31.4.0/24	-	4091	us-east-1e	use1-az3	-
-	subnet-...	Available	vpc-05c...	172.31.1.0/24	-	4091	us-east-1b	use1-az4	-
-	subnet-...	Available	vpc-05c...	172.31.6.0/24	-	4091	us-east-1f	use1-az5	-
-	subnet-...	Available	vpc-05c...	172.31.0.0/24	-	4091	us-east-1d	use1-az1	-

Step-9: Create Route tables My-RT-3 and attach private subnets(private-SN-1)

VPC-Route tables-Create Route table-Name(My-RT-3)-Select VPC(vpc0c633671d8f910d0f (VPC-1))-Create route table.

Select My-RT-3-Edit routes-Actions-edit subnet associations-select private-SN-1-Save associations.

Step-10: Create Route tables My-RT-4 and attach private subnets(private-SN-2)

VPC-Route tables-Create Route table-Name(My-RT-4)-Select VPC(vpc0c633671d8f910d0f (VPC-1))-Create route table.

Select My-RT-4-Edit Routes-Actions-edit subnet associations-select private-SN-2-Save associations.

Step-11: Create two Security group name SG-3 & SG-4 & allow SSH & HTTP Ports.

EC2-Network and security-security groups-Create security group-Name(**SG-3**)-Description(Security ssh-http)-Select VPC (VPC-1)-edit inbound rules-add rules-ssh&http ports-create security group.

EC2-Network and security-security groups-Create security group-Name(**SG-4**)-Description(Security ssh-http)-Select VPC (VPC-1)-edit inbound rules-add rules-ssh&http ports-create security group.

Security Groups (15) <small>Info</small>					
	Name	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	-	sg-0030c1bd930298ba5	SG-4	vpc-0c633671d8f910d0f	sg
<input type="checkbox"/>	-	sg-013c797dcccdead32	SG-2	vpc-0c633671d8f910d0f	sg
<input type="checkbox"/>	-	sg-07d8f8a6edb0f5b91	SG-3	vpc-0c633671d8f910d0f	sg
<input type="checkbox"/>	-	sg-05eadff0260fa29c3b	SG-1	vpc-0c633671d8f910d0f	Security ss
<input type="checkbox"/>	-	sg-095889691dfed491a	default	vpc-0c633671d8f910d0f	default VP
<input type="checkbox"/>	-	sg-02df6fe438337d084	launch-wizard-5	vpc-03c2f0128f050fd80	launch-wiz
<input type="checkbox"/>	-	sg-0e6929616531f0sec	launch-wizard-6	vpc-03c2f0128f050fd80	launch-wiz
<input type="checkbox"/>	-	sg-0a4b30e8311031ea0	launch-wizard-2	vpc-03c2f0128f050fd80	launch-wiz
<input type="checkbox"/>	-	sg-0dc7df37bb7c211c	default	vpc-03c2f0128f050fd80	default VP
<input type="checkbox"/>	-	sg-07ef1ac5b912163e0	launch-wizard-1	vpc-03c2f0128f050fd80	launch-wiz
<input type="checkbox"/>	-	sg-09df588521fffc9ec8	ec2-rds-1	vpc-03c2f0128f050fd80	Security gr

Step-12: Now launch two EC2-Instances in private-SN-1 & private-SN-2 with different availability zones Then we have to check these two private instances connected form public-ins-1 & public-ins-2.

1. private-Ins-1(us-east-1a)
2. private-Ins-2(us-east-1b)

EC2-Launch instance-AMI(ubuntu)-Instance type(t2.micro)-Create new keypair-edit network settings-vpc(vpc-1)-subnet(**private-SN-1**)-Auto assign public ip(Enable)-select existing security group-select(**SG-3**)-Launch instance.

EC2-Launch instance-AMI(ubuntu)-Instance type(t2.micro)-Create new keypair-edit network settings-vpc(vpc-1)-subnet(**private-SN-2**)-Auto assign public ip(Enable)-select existing security group-select(**SG-4**)-Launch instance.

Now connect both instances install apache2 and run web page as a private-Ins-1 & private-Ins-2.

Now Connect EC2-Instances, But couldn't connect. Because it is Private subnet.

For reference

Public-Ins-1

Private ip: 11.0.1.98

Public ip: 100.27.18.73

Public-Ins-2

Private ip: 11.0.2.160

Public ip: 44.22318.22

Private-Ins-1

Private ip: 11.0.3.51

Public ip: 18.212.192.180

Private-Ins-2

Private ip: 11.0.4.134

Public ip: 54.226.126.160

Now connect public-Ins-1 instance-(Connect through git bash)- copy-**chmod 400 "123456.pem"** & **ssh -i "123456.pem" ubuntu@44.214.204.88**

Sudo -i

Vi 98745.pem (Keypair name of the Private ec2 instance (private-Ins-1))

Insert

Copy the 98745.pem file and then paste here

:wq

Now click on private-Ins-1(private ec2 instance)-Connect-ssh-copy- chmod 400 "98745.pem"

Now copy the private-Ins-1(private ec2 instance of private ip)- **11.0.3.51**

Now copy the private-Ins-1 ssh-id (ssh -i "98745.pem" [ubuntu@18.212.192.180](#))

This ip address replaced with private-Ins-1(private ec2 instance of private ip)- **11.0.3.51**

Now the ssh-id is ssh -i "98745.pem" [ubuntu@11.0.3.51](#)

```
root@ip-11-0-1-98:~ Expanded security maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-11-0-3-51:~$ exit  
logout  
Connection to 11.0.3.51 closed.  
root@ip-11-0-1-98:~#
```

Now I am root@ip-11-0-1-98:~# means public-Ins-1 private ip
After given this command ssh -i "98745.pem" ubuntu@11.0.3.51 we are entered
into the private-Ins-1 (private ec2 instance).

ssh -i "98745.pem" ubuntu@11.0.3.51

```
ubuntu@ip-11-0-3-51:~$ exit  
logout  
connection to 11.0.3.51 closed.  
root@ip-11-0-1-98:~# https://changelogs.ubuntu.com/meta-release-lts. check your Internet connection or proxy settings  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Fri Mar  8 16:39:52 UTC 2024  
  
System load: 0.0      Processes: 98  
Usage of /: 20.6% of 7.57GB  Users logged in: 0  
Memory usage: 21%      IPv4 address for eth0: 11.0.3.51  
Swap usage: 0%  
  
Expanded security maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. check your Internet connection or proxy settings  
  
Last login: Fri Mar  8 16:35:06 2024 from 11.0.1.98  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-11-0-3-51:~$
```

- Now access the private-Ins-1 through public-Ins-1. This is also called as Bastion server.

Now clone the repository from [Venna12/dockerjenkin](https://github.com/Venna12/dockerjenkin).

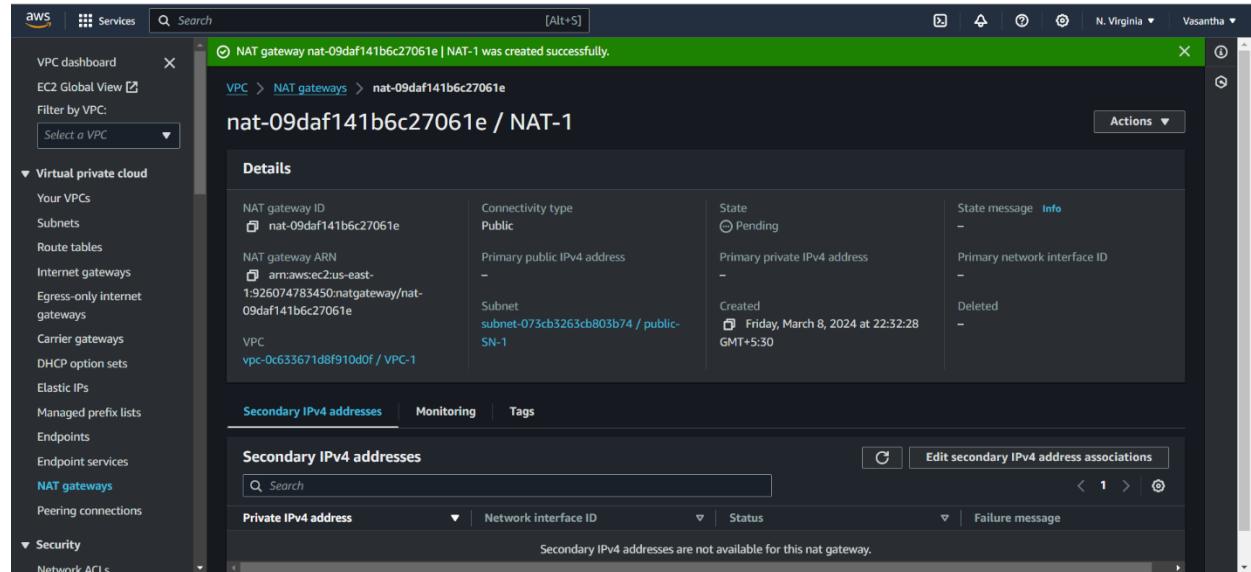
git clone <https://github.com/Venna12/dockerjenkin.git>

```
ubuntu@ip-11-0-3-51:~$ git clone https://github.com/Venna12/dockerjenkin.git
Cloning into 'dockerjenkin'...
^C
ubuntu@ip-11-0-3-51:~$
```

But not cloning the repository. So we need to create NAT Gateway and add NAT Gateway to my-RT-3.

Create NAT Gateway: -

AWS Console-VPC-NAT Gateways-create NAT Gateway-name(NAT-1)-Select subnet(**public-SN-1**)-Elastic IP allocation ID-Allocate elastic IP-create NAT Gateway.



VPC-Elastic ip-Allocate elastic ip address-network border group(us-east-1)-Allocate.

Elastic ip address: **18.205.55.185**

Select elastic ip-Associate elastic ip address-Choose instance(public-Ins-1)-public-Ins-1 private ip address(11.0.1.98)-Associate

The screenshot shows the AWS VPC console with the following details:

- Elastic IP address associated successfully.**
- Elastic IP address:** 18.205.55.185
- Associated instance ID:** i-0f057146c0faa7c20
- Type:** Public IP
- Scope:** VPC
- Allocation ID:** eipalloc-0fbeb82601b507140
- Associated instance ID:** i-0f057146c0faa7c20
- Public DNS:** -
- Private IP address:** 11.0.1.98
- Network interface ID:** eni-03ae2f8c18421f922
- Network interface owner account ID:** 926074783450
- Address pool:** Amazon
- Network border group:** us-east-1

Now select RT-3-Edit routes-add routes-0.0.0.0/0-NAT Gateway-NAT

The screenshot shows the AWS VPC Route Tables page with the following details:

- Route table ID:** rtb-009fecf0a9e2f38d3
- Main:** No
- VPC:** vpc-0c633671d8f910d0f | VPC-1
- Explicit subnet associations:** subnet-0d33f582b87c35ba3 / private-SN-1
- Edge associations:** -

Routes:

Destination	Target	Status	Propagated
0.0.0.0/0	nat-09daf141b6c27061e	Active	No
11.0.0.16	local	Active	No

Now open git bash here-copy the ssh-command of public ec2-instance(public-Ins-1)

ssh -i "123456.pem" [ubuntu@18.205.55.185](#)

sudo -i

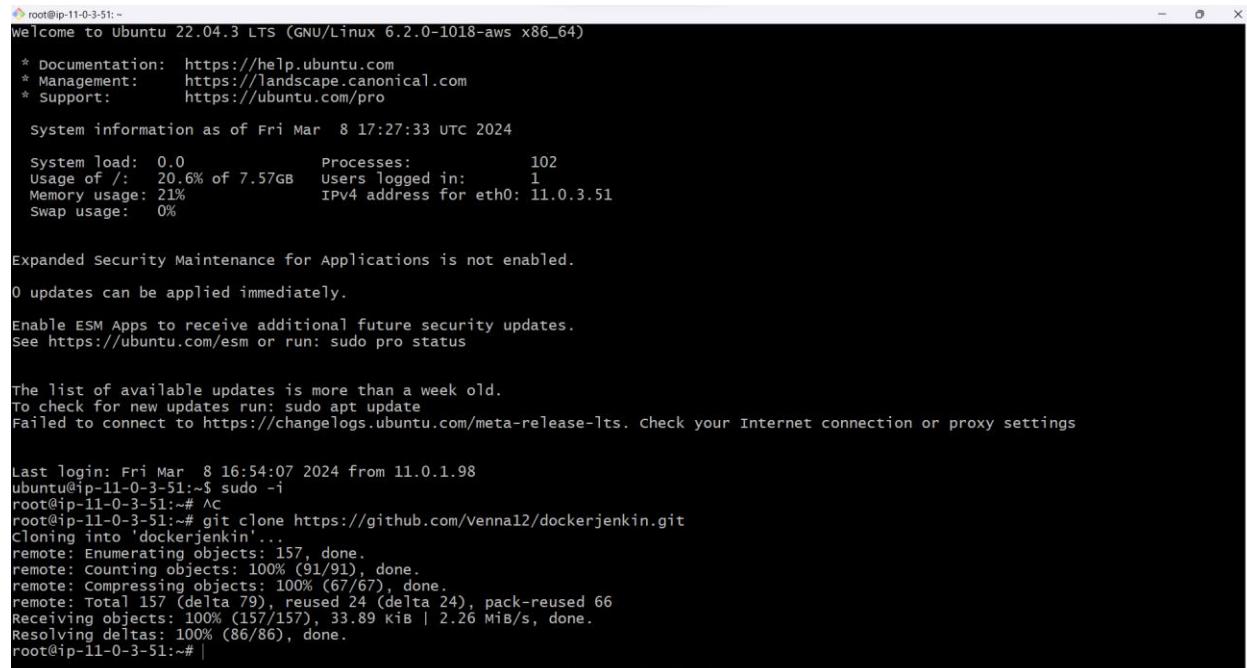
```
root@ip-11-0-1-98:~#  
ssh -i "98745.pem" ubuntu@11.0.3.51
```

```
ubuntu@ip-11-0-3-51:~$
```

```
sudo -i
```

Now clone the repository from [Venna12/dockerjenkin](#).

```
git clone https://github.com/Venna12/dockerjenkin.git
```



The screenshot shows a terminal window with the following output:

```
root@ip-11-0-3-51:~  
welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-aws x86_64)  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Fri Mar  8 17:27:33 UTC 2024  
  
System load: 0.0      Processes: 102  
Usage of '/': 20.6% of 7.57GB  Users logged in: 1  
Memory usage: 21%          IPv4 address for eth0: 11.0.3.51  
Swap usage: 0%  
  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings  
  
Last login: Fri Mar  8 16:54:07 2024 from 11.0.1.98  
ubuntu@ip-11-0-3-51:~$ sudo -i  
root@ip-11-0-3-51:~# ^C  
root@ip-11-0-3-51:~# git clone https://github.com/Venna12/dockerjenkin.git  
Cloning into 'dockerjenkin'...  
remote: Enumerating objects: 157, done.  
remote: Counting objects: 100% (91/91), done.  
remote: Compressing objects: 100% (67/67), done.  
remote: Total 157 (delta 79), reused 24 (delta 24), pack-reused 66  
Receiving objects: 100% (157/157), 33.89 KiB | 2.26 MiB/s, done.  
Resolving deltas: 100% (86/86), done.  
root@ip-11-0-3-51:~# |
```

Now cloning done.

Now connect public-Ins-2 instance-(Connect through git bash)- copy- chmod 400 "98745.pem" & ssh -i "98745.pem" ubuntu@54.226.126.160

Sudo -i

Vi 98745.pem (Keypair name of the Private ec2 instance (private-Ins-2))

Insert

Copy the 98745.pem file and then paste here

:wq

Now click on private-Ins-2(private ec2 instance)-Connect-ssh-copy- chmod 400 "98745.pem"

Now copy the private-Ins-2(private ec2 instance of private ip)- **11.0.4.134**

Now copy the private-Ins-2 ssh-id (ssh -i "98745.pem" ubuntu@54.226.126.160)

This ip address replaced with private-Ins-1(private ec2 instance of private ip)- **11.0.4.134**

Now the ssh-id is ssh -i "98745.pem" ubuntu@11.0.4.134

```
Last login: Fri Mar  8 12:42:56 2024 from 18.206.107.29
ubuntu@ip-11-0-2-160:~$ vi 98745.pem
Vi: command not found
root@ip-11-0-2-160:~$ ; chmod 400 98745.pem
root@ip-11-0-2-160:~$ ssh -i "98745.pem" ubuntu@11.0.4.134
Warning: Permanently added '11.0.4.134' (ED25519) to the list of known hosts.
The authenticity of host '11.0.4.134 (11.0.4.134)' can't be established.
ED25519 key fingerprint is SHA256:OlnARlxn8GtCslqfutv33ct09zGw137MgknsG/gs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added '11.0.4.134' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Mar  8 17:49:18 UTC 2024

System load: 0.080078125 Processes: 94
Usage of /: 20.3% of 7.57GB Users logged in: 0
Memory usage: 20% IPv4 address for eth0: 11.0.4.134
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-11-0-4-134:~$ sudo -i
root@ip-11-0-4-134:~#
```

Now clone the repository from [Venna12/dockerjenkin](https://github.com/Venna12/dockerjenkin).

git clone <https://github.com/Venna12/dockerjenkin.git>

```

root@ip-11-0-4-134:~#
Last Login: Fri Mar  8 12:42:56 2024 from 18.206.107.29
ubuntu@ip-11-0-2-160:~$ sudo -i
root@ip-11-0-2-160:~$ vi 98745.pem
Vi: /home/ubuntu not found
root@ip-11-0-2-160:~# chmod 400 "98745.pem"
root@ip-11-0-2-160:~# ssh -i "98745.pem" ubuntu@11.0.4.134
The authenticity of host '11.0.4.134 (11.0.4.134)' can't be established.
ED25519 key fingerprint is SHA256:0InARlxpxGhc6UqxfHG'53Et09zGw137MggknsG/gs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[Fingerprint])? yes
warning: Permanently added '11.0.4.134' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Mar  8 17:49:18 UTC 2024

System load: 0.080078125 Processes: 94
Usage of /: 20.3% of 7.57GB Users logged in: 0
Memory usage: 20% IPv4 address for eth0: 11.0.4.134
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See man sudo_root for details.

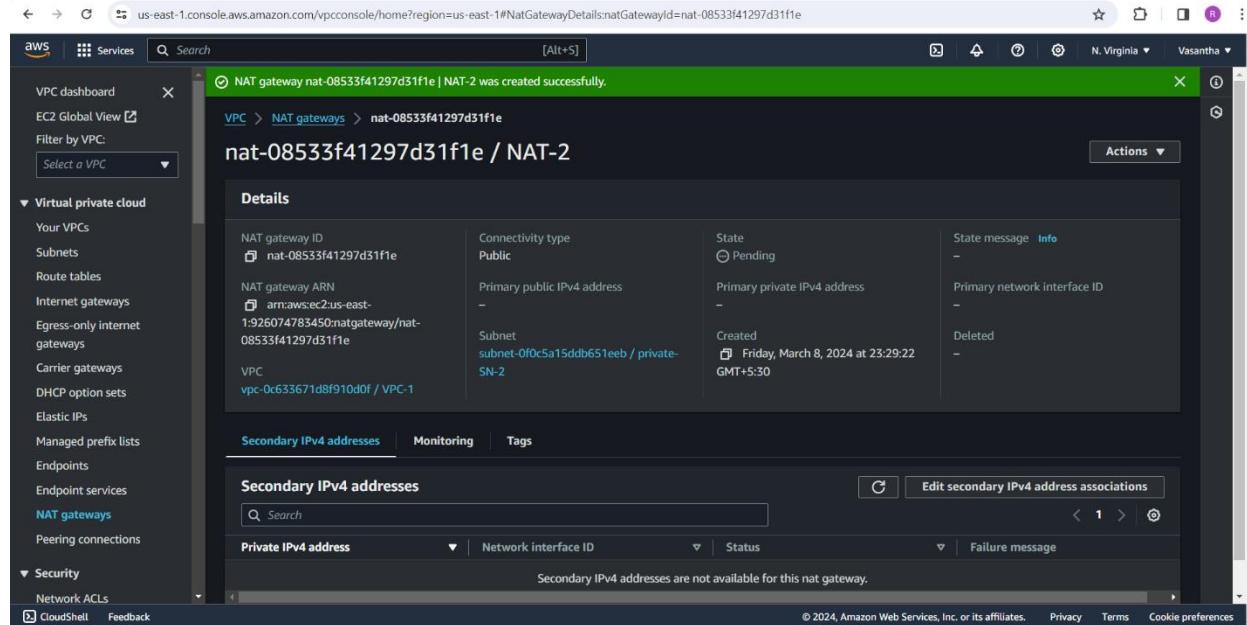
ubuntu@ip-11-0-4-134:~$ sudo -i
root@ip-11-0-4-134:~# git clone https://github.com/venna12/dockerjenkin.git
Cloning into 'dockerjenkin'...

```

But not cloning. So we need to create NAT Gateway and add NAT to my-Rt-4.

Create NAT Gateway: -

AWS Console-VPC-NAT Gateways-create NAT Gateway-name(NAT-2)-Select subnet(**public-SN-2**)-Elastic IP allocation ID-Allocate elastic IP-create NAT Gateway.



VPC-Elastic ip-Allocate elastic ip address-network border group(us-east-1)-Allocate.

Elastic ip address: 18.204.114.249

Select elastic ip-Associate elastic ip address-Choose instance(public-Ins-2)-public-Ins-1 private ip address(11.0.2.160)-Associate.

The screenshot shows the AWS VPC dashboard with the 'Elastic IP addresses' section selected. A success message at the top states: "Elastic IP address associated successfully. Elastic IP address 18.204.114.249 has been associated with instance i-003ed384d04d8e505". Below this, the details for the IP address 18.204.114.249 are listed in a table. The table includes columns for Allocated IPv4 address, Type, Allocation ID, Reverse DNS record, Association ID, Scope, Associated instance ID, Private IP address, Network interface ID, Network interface owner account ID, Public DNS, NAT Gateway ID, Address pool, and Network border group. The data shows the IP is a Public IP associated with instance i-003ed384d04d8e505, which is part of the us-east-1 network border group. There are no tags present.

Now select My-RT-4-Edit routes-add routes-0.0.0.0/0-NAT Gateway-NAT-2.

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A success message at the top states: "Updated routes for rtb-0503456af268de290 / My-RT-4 successfully". Below this, the details for the route table rtb-0503456af268de290 are shown. The table includes columns for Route table ID, Main, Explicit subnet associations, Edge associations, VPC, Owner ID, and Subnet associations. Under the 'Routes' tab, two routes are listed: one to 0.0.0.0/0 targeting the NAT gateway nat-08533f41297d31f1e, and another local route to 11.0.0.0/16. Both routes are active and not propagated.

Now open git bash here-copy the ssh-command of public ec2-instance(public-Ins-2)

```
ssh -i "123456.pem" ubuntu@18.204.114.249
```

```
sudo -i
```

```
root@ip-11-0-2-160:~#  
ssh -i "98745.pem" ubuntu@11.0.3.51
```

```
root@ip-11-0-4-134:~#  
sudo -i
```

Now clone the repository from [Venna12/dockerjenkin](https://github.com/Venna12/dockerjenkin).

```
git clone https://github.com/Venna12/dockerjenkin.git
```

```
root@ip-11-0-4-134:~#  
Enable ESM Apps to receive additional future security updates.  
see https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Fri Mar  8 18:25:21 2024 from 49.43.234.254  
ubuntu@ip-11-0-2-160:~$ sudo -i  
root@ip-11-0-2-160:~# ^C  
root@ip-11-0-2-160:~# ssh -i "98745.pem" ubuntu@11.0.4.134  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Fri Mar  8 18:46:17 UTC 2024  
  
System load: 0.0      Processes:          105  
Usage of /: 20.6% of 7.57GB  Users logged in: 1  
Memory usage: 22%           IPv4 address for eth0: 11.0.4.134  
Swap usage: 0%  
  
Expanded security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings  
  
Last login: Fri Mar  8 18:29:56 2024 from 11.0.2.160  
ubuntu@ip-11-0-4-134:~$ sudo -i  
root@ip-11-0-4-134:~#
```

```

root@ip-11-0-2-160:~# ssh -i "98745.pem" ubuntu@11.0.4.134
welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Mar  8 18:46:17 UTC 2024

System load: 0.0          Processes:           105
Usage of /: 20.6% of 7.57GB Users logged in: 1
Memory usage: 22%          IPv4 address for eth0: 11.0.4.134
Swap usage: 0%             Swap usage: 0% 

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. check your Internet connection or proxy settings

Last login: Fri Mar  8 18:29:56 2024 from 11.0.2.160
ubuntu@ip-11-0-4-134:~$ sudo -i
root@ip-11-0-4-134:~# git clone https://github.com/venna12/dockerjenkin.git
cloning into 'dockerjenkin' ...
remote: Enumerating objects: 157, done.
remote: Counting objects: 100% (91/91), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 157 (delta 79), reused 24 (delta 24), pack-reused 66
Receiving objects: 100% (157/157), 33.89 KiB | 1.03 MiB/s, done.
Resolving deltas: 100% (86/86), done.
root@ip-11-0-4-134:~#

```

Now cloning done.

Step-13: Create auto scaling group for private-Ins-1 & private-Ins-2.

Create target groups:

EC2-Load balancing-Target group-create target group-instances-Name(TG-2)-select VPC(VPC-1)-next-select two instances (private-Ins-1 & private-Ins-2)-Including as pending below-create target group-create target group.

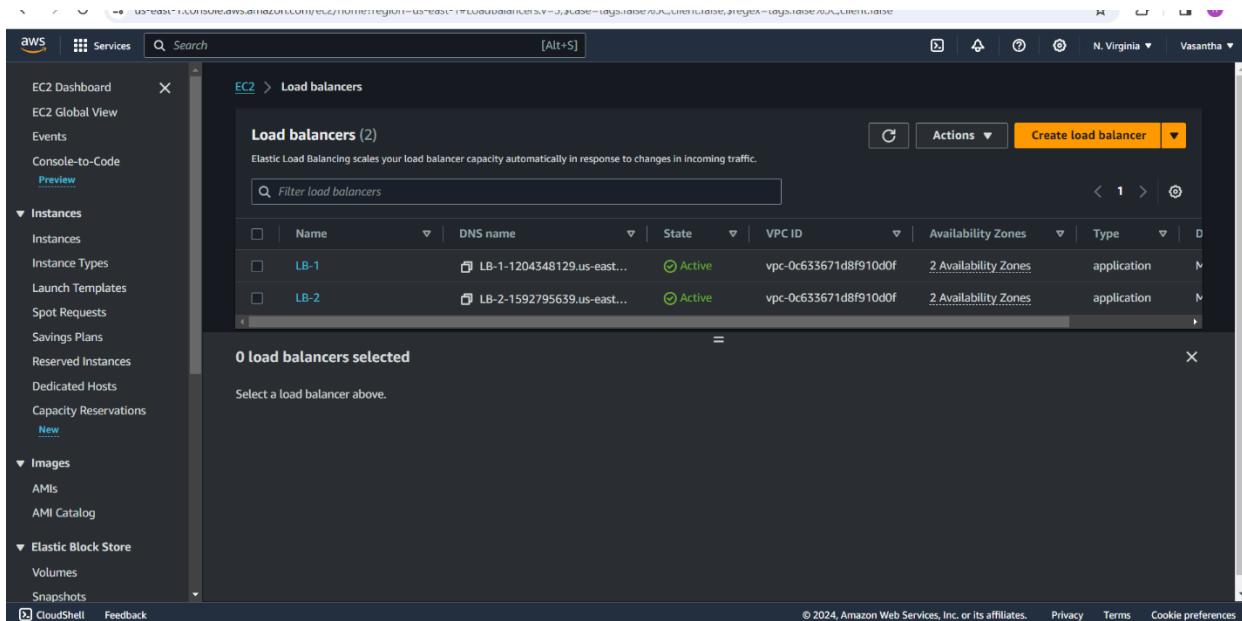
The screenshot shows the AWS EC2 Target Groups page. The left sidebar lists various EC2 services like Instances, Images, and Elastic Block Store. The main pane displays a table of target groups with the following data:

	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input type="checkbox"/>	TG-1	arn:aws:elasticloadbalancing:us-east-1:0633671d8f910d0f	80	HTTP	Instance	LB-1	vpc-0c633671d8f910d0f
<input type="checkbox"/>	TG-2	arn:aws:elasticloadbalancing:us-east-1:0633671d8f910d0f	80	HTTP	Instance	None associated	vpc-0c633671d8f910d0f

Below the table, a message says "0 target groups selected" and "Select a target group above."

Create Load balancer:

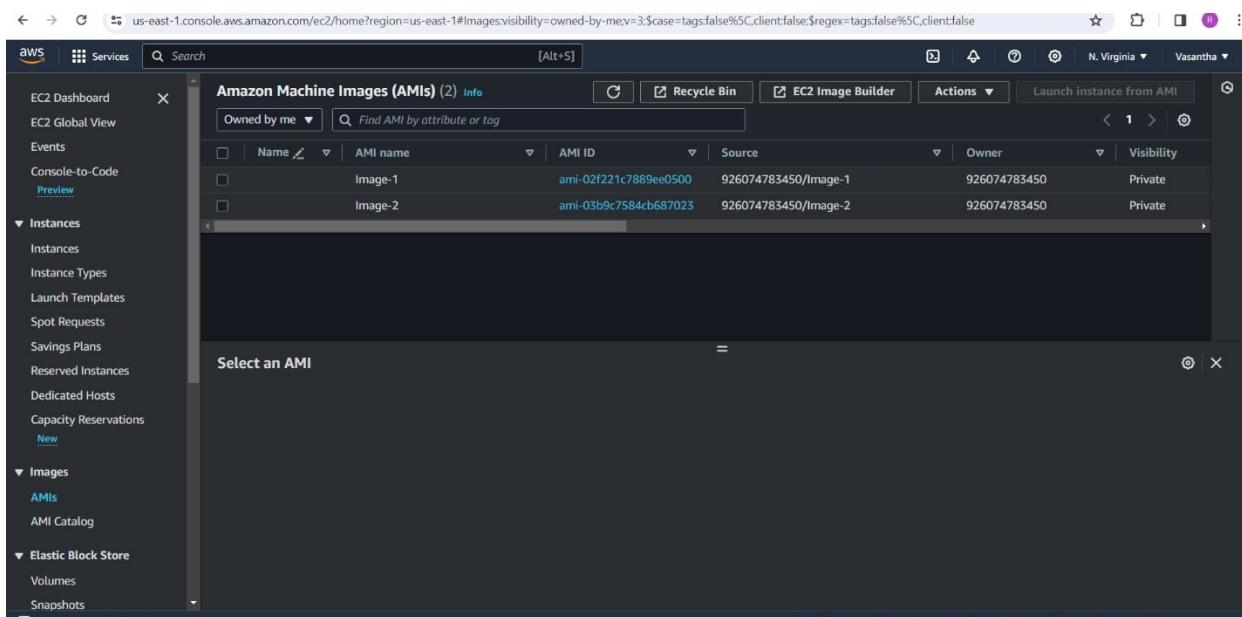
EC2-Load balancer-create load balancer-application load balancer-create-name(LB-2)-Select VPC(VPC-1)-mappings-us-east-1a&us-east-1b-select security group-SG-3>Select target group(TG-2)-create load balancer.



The screenshot shows the AWS EC2 Load Balancers page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images, and Elastic Block Store. The main area is titled "Load balancers (2)". It displays two entries: LB-1 and LB-2. Both are listed as "Active" with "vpc-0c633671d8f910d0f" as the VPC ID and "2 Availability Zones". The Type is listed as "application". A "Create load balancer" button is visible at the top right. Below the table, it says "0 load balancers selected" and "Select a load balancer above."

Create Image:

Select private-Ins-1-Instance-Actions-Image and templates-create image-name(Image-2)-description(image)-create image.



The screenshot shows the AWS EC2 Amazon Machine Images (AMIs) page. The left sidebar includes options for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images (with AMIs selected), and Elastic Block Store. The main content area is titled "Amazon Machine Images (AMIs) (2)" and lists "Image-1" and "Image-2". Both AMIs are marked as "Private". Below the table, a modal window titled "Select an AMI" is open, prompting the user to choose an AMI to launch.

Launch template: -

EC2-Launch template-Create launch template-name(Template-2)-description(LT)-ami-my ami-owned by me-select ami(image-2)-Instance type(t2.micro)-key pair name(select which you want)-select existing security group(SG-3)-Create launch template.

The screenshot shows the AWS EC2 Launch Templates page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances, Instance Types, Launch Templates (which is selected), Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, CloudShell, and Feedback. The main content area has a title "Launch Templates (2) Info" with a search bar and a table. The table has columns: Launch Template ID, Launch Template Name, Default Vers..., Latest Ver..., Create Time, and Created By. It lists two entries: "lt-0ae2af2ec9fe9ce01" (Template-1) and "lt-081a0c1c2e6f936bd" (Template-2). Below the table is a modal window titled "Select a launch template" with a single option: "Template-1". At the bottom right of the main page, there are links for "Privacy", "Terms", and "Cookie preferences".

Create Autoscaling Groups: -

EC2-Autoscaling-Autoscaling Groups-create autoscaling groups-Autoscaling group2-select launch template(Template-2)-Next-instance type requirements-specify instance attributes-vCPUs-minimum:2-maximum:2-Memory GiB-minimum:2-maximum:4-select vpc(VPC-1)-Availability zones and subnets(us-east-1a,us-east-2,private-SN-1,private-SN-2)-next-attach to an existing load balancer-Choose from your load balancer target groups-existing load balancer target groups(TG-2)-Health check grace period(100sec)-desired capacity type: CPU-desired capacity-2-scaling- Min desired capacity:2- Max desired capacity:4- Automatic scaling-target tracking scaling policy-name(target tracking policy)-Metric typ(Average cpu utilization)-Target value(50)-instance warmup(100sec)-Next-Next-Next-create autoscaling group.

The screenshot shows the AWS EC2 Auto Scaling groups page. At the top, there are tabs for Launch configurations, Launch templates, Actions, and Create Auto Scaling group. Below the tabs is a search bar and a table listing two Auto Scaling groups:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability zone
ACG-2	Template-2 Version Default	1	-	2	2	4	us-east...
Auto scaling group-1	Template-1 Version Default	2	-	2	2	4	us-east...

Below the table, a message says "0 Auto Scaling groups selected".

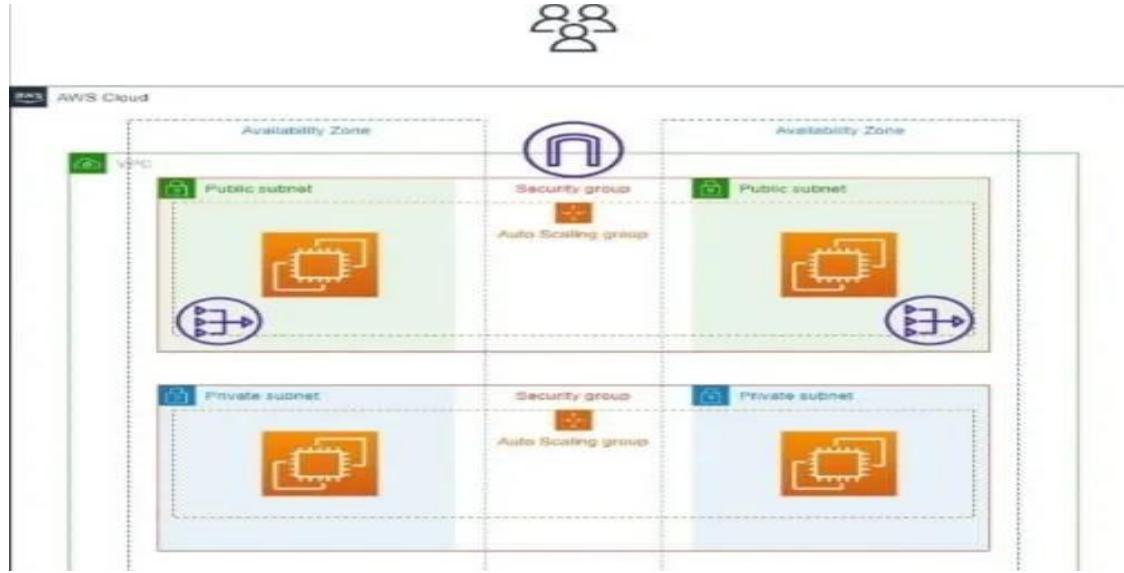
Now click on instances there is

The screenshot shows the AWS EC2 Instances page. On the left, there is a navigation sidebar with various EC2-related options like Dashboard, Global View, Events, and Instances. The Instances section is expanded, showing sub-options like Instances Types, Instance Templates, and Launch Templates.

The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
I-06b59d53d58b16d57	I-06b59d53d58b16d57	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
public-Ins-2	I-003ed384d04d8e505	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
private-Ins-1	I-0f7dd47345e1219e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-
private-Ins-2	I-0f4e7469f83ea13f9	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
	I-0a8267665abca031a	Terminated	c1.medium	-	View alarms +	us-east-1b	-
	I-0f057146c0faa7c20	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-
	I-088471e2c3625b1f0	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-
	I-0381163d1d6c5c6cc	Running	c6a.large	2/2 checks passed	View alarms +	us-east-1a	-

Below the table, a message says "Select an instance".



This Part Done

Create RDS Instances

Step14: - Create 2-Private subnets in different Availability zones

- 3.** private-SN-3(us-east-1a)
- 4.** private-SN-4(us-east-1b)

VPC-Subnet-Create subnet-Select VPC ID (vpc0c633671d8f910d0f)
VPC-1)-name(**private-SN-3**)-AZ(**us-east-1a**)-subnet cidr block- **11.0.5.0/24**-create subnet.

VPC-Subnet-Create subnet-Select VPC ID (vpc0c633671d8f910d0f)
VPC-1)-name(**private-SN-4**)-AZ(**us-east-1b**)-subnet cidr block- **11.0.6.0/24**-create subnet.

Subnets (12) <small>Info</small>													
	Name	Subnet	State	VPC	IPv4 CIDR	IPv6 ...	Available	Available	Available	Netw...	Route table		
	-	subnet...	Available	vpc-03c...	172.31.80.0/20	-	4091	us-east-1a	use1-az2	-	-		
	-	subnet...	Available	vpc-03c...	172.31.32.0/20	-	4091	us-east-1c	use1-az6	-	-		
	-	subnet...	Available	vpc-03c...	172.31.48.0/20	-	4091	us-east-1e	use1-az3	-	-		
	-	subnet...	Available	vpc-03c...	172.31.16.0/20	-	4091	us-east-1b	use1-az4	-	-		
	-	subnet...	Available	vpc-03c...	172.31.64.0/20	-	4091	us-east-1f	use1-az5	-	-		
	-	subnet...	Available	vpc-03c...	172.31.0.0/20	-	4091	us-east-1d	use1-az1	-	-		
	private-SN-1	subnet...	Available	vpc-0c6...	11.0.3.0/24	-	248	us-east-1a	use1-az2	-	rtb-009fecf0a5		
	public-SN-2	subnet...	Available	vpc-0c6...	11.0.2.0/24	-	246	us-east-1b	use1-az4	-	rtb-055d8b58c		
	public-SN-1	subnet...	Available	vpc-0c6...	11.0.1.0/24	-	247	us-east-1a	use1-az2	-	rtb-09724b1dc		
	private-SN-2	subnet...	Available	vpc-0c6...	11.0.4.0/24	-	250	us-east-1b	use1-az4	-	rtb-0503456af		
	private-SN-3	subnet...	Available	vpc-0c6...	11.0.5.0/24	-	251	us-east-1a	use1-az2	-	-		
	private-SN-4	subnet...	Available	vpc-0c6...	11.0.6.0/24	-	251	us-east-1b	use1-az4	-	-		

Step-15:- Create two Security group name SG-5 & SG-6 & allow SSH,HTTP & mysql/Aurora.

EC2-Network and security-security groups-CREATE security group-Name(**SG-5**)-Description(Security ssh-http)-Select VPC (VPC-1)-edit inbound rules-add rules-ssh,http& mysql/Aurora ports-create security group.

EC2-Network and security-security groups-CREATE security group-Name(**SG-6**)-Description(Security ssh-http)-Select VPC (VPC-1)-edit inbound rules-add rules-ssh,http& mysql/Aurora -create security group.

Step-16:- Create two RDS Instances with same VPC and different availability zones.

EC2-Launch instance-name(RDS-1)-ubuntu-t2.micro-create new key pair-edit network settings-vpc(VPC-1)-Subnet(private-SN-3)- Autoassign ip(enable)- select existing security group(SG-5)-Launch instance.

EC2-Launch instance-name(RDS-2)-ubuntu-t2.micro-create new key pair-edit network settings-vpc(VPC-1)-Subnet(private-SN-4)- Autoassign ip(enable)- select existing security group(SG-6)-Launch instance.

Step-17:- Create Two RDS Databases (Database-1 & Database-2)

Database-1: AWS Console-RDS-Create database-easy to create-MYSQL-Free tier-master user name: admin-master password: Rajesh#0987-confirm password: Rajesh#0987-setup ec2 connection-connect to an ec2 compute resource-choose an ec2-Instance-RDS-1-create database.

Database-2: AWS Console-RDS-Create database-easy to create-MYSQL-Free tier-master user name: admin-master password: Rajesh#0987-confirm password: Rajesh#0987-setup ec2 connection-connect to an ec2 compute resource-choose an ec2-Instance-RDS-2-create database.

The screenshot shows the Amazon RDS console with the 'Databases' page selected. A modal dialog at the top right provides information about creating a Blue/Green deployment. The main table lists two database instances:

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU
database-1	Available	Instance	MySQL Community	us-east-1a	db.r6g.large		
database-2	Available	Instance	MySQL Community	us-east-1b	db.r6g.large		

Now go to instances then click on RDS-1 Instance-Connect-but will not connect because of this instance in private subnet.

For reference

Public-Ins-1

Private ip: 11.0.1.98

Public ip: 100.27.18.73

Public-Ins-2

Private ip: 11.0.2.160

Public ip: 44.22318.22

RDS-1 Instance

Private ip: 11.0.5.250

Public ip: 54.89.226.79

RDS-2 Instance

Private ip: 11.0.6.185

Public ip: 54.242.218.189

Now connect public-Ins-1 instance-

```
chmod 400 "123456.pem" & ssh -i "123456.pem" ubuntu@18.205.55.185
```

```
sudo -i
```

```
ls
```

```
98745.pem (This keypair is for private-Ins-1)
```

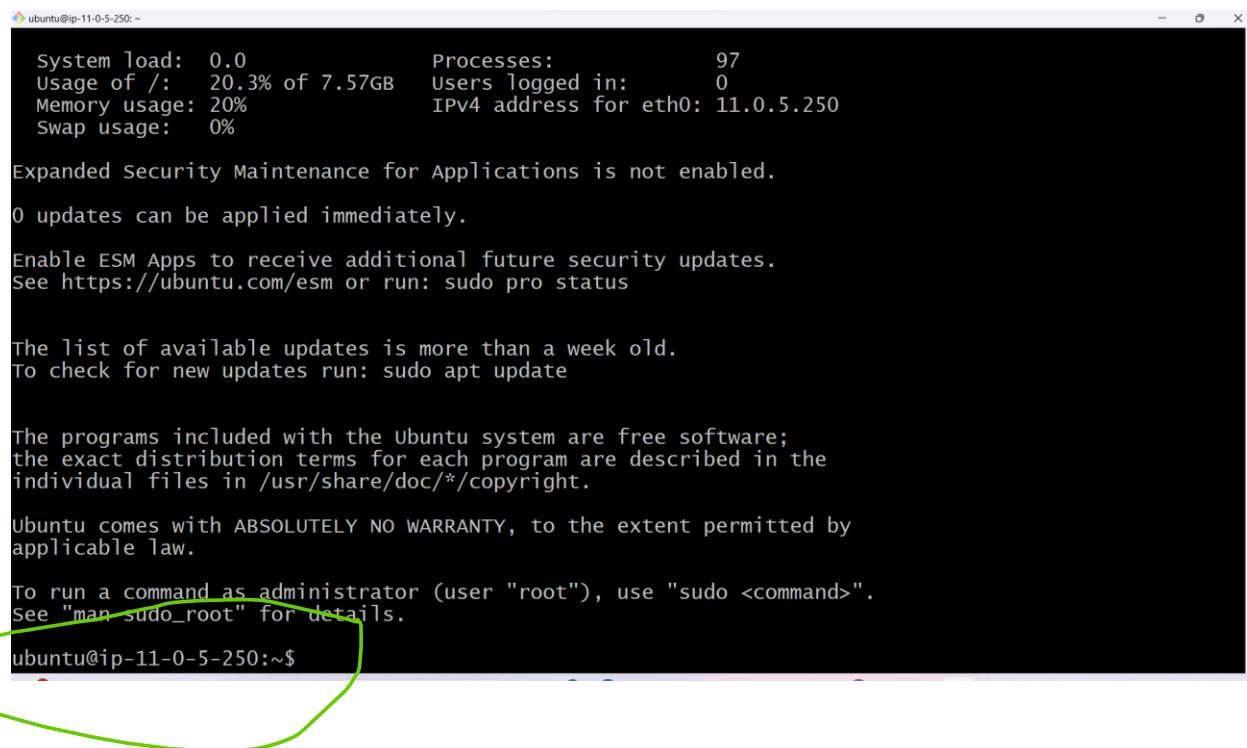
```
Vi 123.pem (Keypair name of the Private ec2 instance (RDS-1))
```

```
Insert the 123.pem details :wq
```

Now open the RDS-1 Instance

```
chmod 400 "123.pem" & ssh -i "123.pem" ubuntu@54.89.226.79 (replace this ip  
with RDS-1 Instance private ip: 11.0.5.250.
```

then the command is **ssh -i "123.pem" ubuntu@11.0.5.250**



```
ubuntu@ip-11-0-5-250: ~
System load: 0.0      Processes: 97
Usage of /: 20.3% of 7.57GB  Users logged in: 0
Memory usage: 20%          IPv4 address for eth0: 11.0.5.250
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-11-0-5-250:~$
```

Now connect public-Ins-2 instance-

```
chmod 400 "123456.pem" & ssh -i "123456.pem" ubuntu@18.204.114.249
```

```
sudo -i
```

```
98745.pem (This keypair is for private-Ins-2)
```

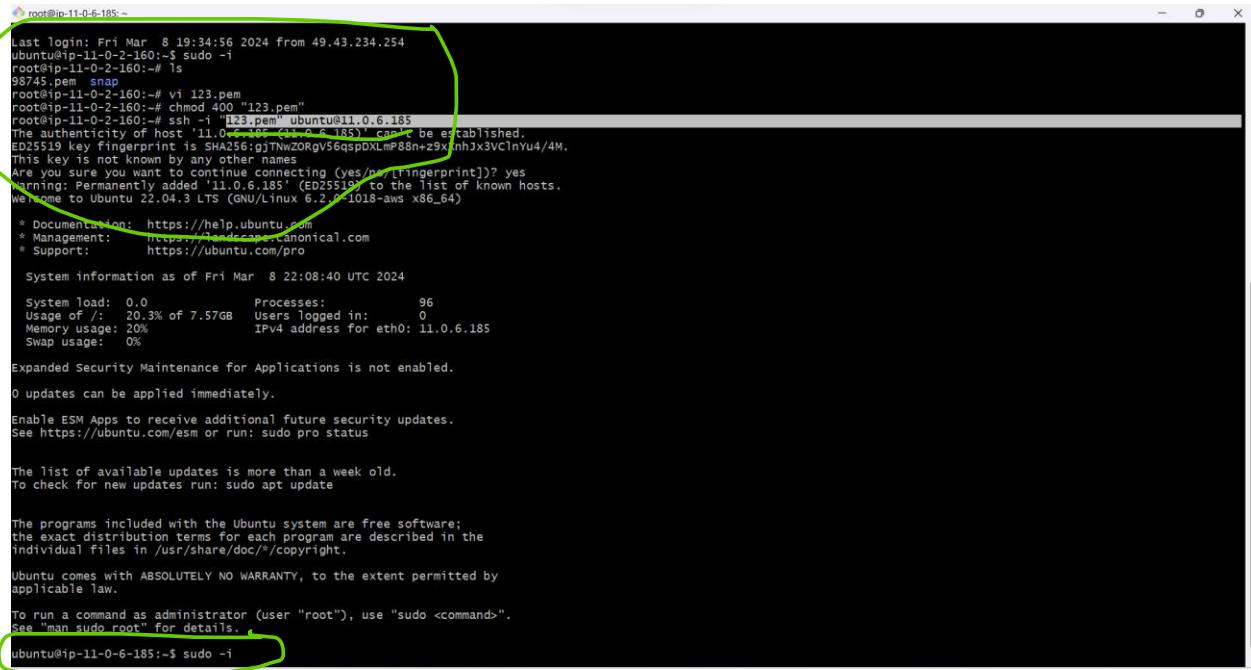
```
Vi 123.pem (Keypair name of the Private ec2 instance (RDS-2))
```

Insert the 123.pem details :wq

Now open the RDS-2 Instance

chmod 400 "123.pem" & ssh -i "123.pem" ubuntu@54.242.218.189 (replace this ip with RDS-2 Instance private ip:11.0.6.185)

then the command is ssh -i "123.pem" ubuntu@11.0.6.185



```
Last login: Fri Mar  8 19:34:56 2024 from 49.43.234.254
ubuntu@ip-11-0-2-160:~$ sudo -i
root@ip-11-0-2-160:~# ls
98/45.pem snap
root@ip-11-0-2-160:~# chmod 400 "123.pem"
root@ip-11-0-2-160:~# ssh -i "123.pem" ubuntu@11.0.6.185
The authenticity of host '11.0.6.185 (11.0.6.185)' can't be established.
ED25519 key fingerprint is SHA256:giTNwZORgV5GqspdXLMp88n+z9XnhJx3VClnYu4/4M.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '11.0.6.185' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Mar  8 22:08:40 UTC 2024

System load: 0.0          Processes:           96
Usage of /: 20.3% of 7.57GB   Users logged in:      0
Memory usage: 20%          IPv4 address for eth0: 11.0.6.185
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-11-0-6-185:$ sudo -i
```

Now our RDS (mysql-database) accessed by public-ins-1 & public-Ins-2.

