

PGPCC
Project Implementation
Team Communication Solution using
Mattermost and AWS

--Mahesh Jasti

Contents

Section 1: Problem Statement.....	Pg.3
Section 2: Scope of the Project.....	Pg.3
Section 3: Implementation Architecture.....	Pg.4
Section 4: High level steps to implement the project...	Pg.4
Section 5: Screenshots from AWS console.....	Pg.5
Section 6: Lessons learnt / Observations.....	Pg.40

Section 1: Problem Statement

Mattermost is an open-source, self-hostable online chat service with file sharing, search, and has other integration options. It is designed as an internal chat platform for organizations and companies, and mostly markets itself as an open-source alternative to Slack. It uses a 3 tier architecture that can be hosted using an IaaS provider or on-premises servers.

The purpose of this project is to deploy trial version of Mattermost application along with MySql server in AWS.

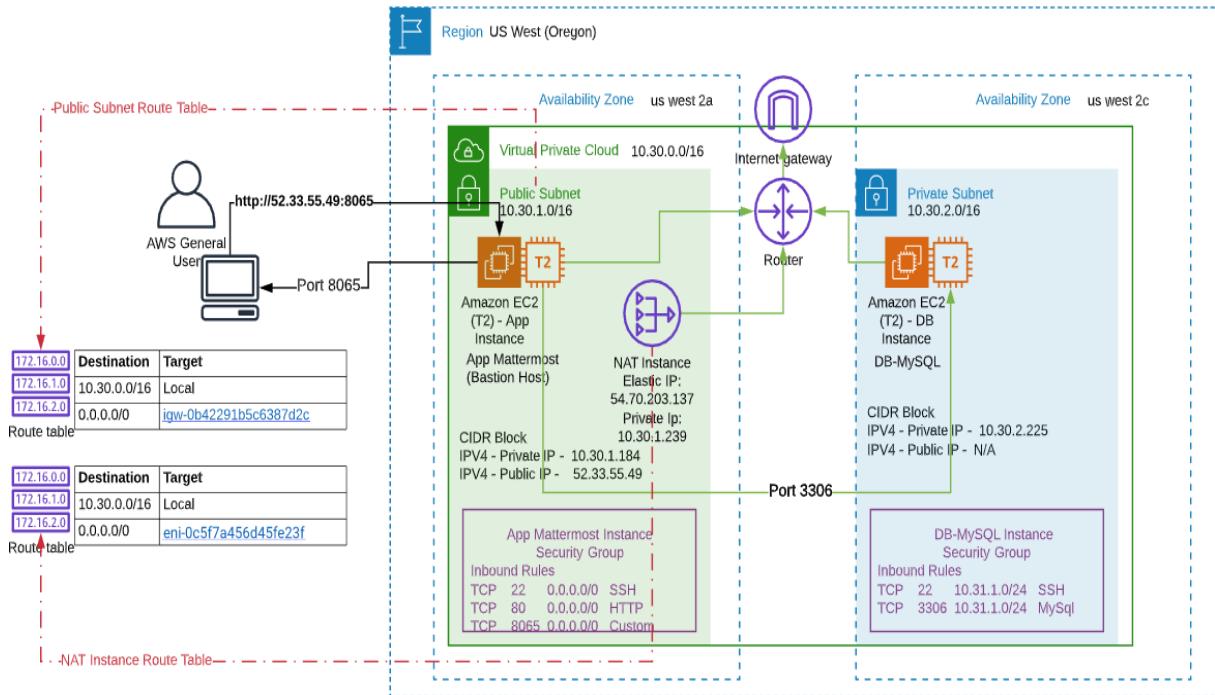
Section 2: Scope of the Project

The scope of the project implementation is as follows.

1. Architecture
2. Implementation
 - a. Create VPC with 2 Subnets.
 - b. Install and configure MySQL on an Ubuntu 18.04 in Private Subnet.
 - c. Install and configure Mattermost on an Ubuntu 18.04 instance on the public subnet.
 - d. Configure appropriate security groups.
 - e. Validate the application by accessing the IP of the public instance thru port 8065.

Section 3: Implementation Architecture

The following picture outlines the implementation architecture for the project. I have placed all the information related to the IP addresses(Public or Private), Route table configuration, Security Groups rules etc. This will give complete picture on how I have implemented the project



Section 4: High level Steps to implement the project

1. VPC, Subnets, Internet Gateway and Route Tables setup
2. Ubuntu EC2 and NAT Instances setup
3. Route table setup for PrivateSubnet setup
4. EC2 instances – Access check b/n Public and Private
5. Installation of Mysql & Mattermost softwares and browse the public IP– Setup

Section 5: Screenshots from AWS console

*****VPC, Subnets, IGWY and Route Tables setup *****

Step#1

Login to AWS Console and search for VPC

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with various VPC-related options like Subnets, Route Tables, Internet Gateways, and Security Groups. The main area displays a table of VPCs. One VPC is selected, showing its details: Name (vpc-71dead09), VPC ID (vpc-71dead09), State (available), IPv4 CIDR (172.31.0.0/16), IPv6 CIDR (Network Border Group), DHCP options set (dopt-31cca449), and Main Route table (rtb-1288d469 | Default Route T...). Below this, a detailed view of the selected VPC ('vpc-71dead09') is shown with tabs for Description, CIDR Blocks, Flow Logs, and Tags. The Description tab lists various configuration parameters such as VPC ID, State, IPv4 CIDR, IPv6 CIDR (Network Border Group), DNS resolution, DNS hostnames, ClassicLink DNS Support, and Owner.

Step#2

Click on Create VPC button and enter the respective details

Nametag: ProjectMattermost ; CIDR Block: 10.30.0.0/16

The screenshot shows the 'Create VPC' wizard. It's the first step, titled 'Create VPC'. It asks for a 'Name tag' (ProjectMattermost), an 'IPv4 CIDR block' (10.30.0.0/16), and an 'IPv6 CIDR block' (with options: No IPv6 CIDR Block, Amazon provided IPv6 CIDR block, or IPv6 CIDR owned by me). It also has a 'Tenancy' dropdown set to 'Default'. At the bottom, there's a note about VPC requirements and a 'Create' button.

PGPCC – Project Implementation Task

Step#3

Once click on create, it creates a VPC ID in the main window

The screenshot shows the 'Create VPC' wizard in the AWS Management Console. A green success message box contains the text: 'The following VPC was created:' followed by the VPC ID 'vpc-0c7865baf88ba1857'. Below the message is a 'Close' button.

The screenshot shows the 'VPC Dashboard' in the AWS Management Console. On the left, the 'Your VPCs' section lists the newly created VPC 'ProjectMattermost' and another VPC 'vpc-71dead09'. The main pane displays detailed information for the VPC 'vpc-71dead09', including its configuration like CIDR blocks and network ACLs. The 'Description' tab is selected.

Step#4

Now after creating the VPC, click on the Subnets option on the left menu..

The screenshot shows the 'VPC Dashboard' in the AWS Management Console. The 'Subnets' section is selected on the left sidebar. The main pane displays a table of subnets for the VPC 'vpc-71dead09', including details like subnet IDs, IP ranges, and availability zones. The 'CIDR Blocks' tab is selected.

Step#5

Click on “Create subnet” blue button and create a Subnets as shown below.

Subnet#1 – PublicSubnet with CIDR addr as 10.30.1.0/24 under us-west-2a

Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag	PublicSubnet	<small>i</small>	
VPC*	vpc-0c7865baf88ba1857	<small>i</small>	
Availability Zone	us-west-2a	<small>i</small>	
VPC CIDRs	CIDR	Status	Status Reason
	10.30.0.0/16	associated	
IPv4 CIDR block*	10.30.1.0/24	<small>i</small>	

* Required

Create

Subnets > Create subnet

Create subnet

The following Subnet was created:

Subnet ID subnet-0d0cfea42bf27567a

Close

New VPC Experience
Tell us what you think

VPC Dashboard New

Filter by VPC:
Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
PublicSubnet	subnet-0d0cfea42bf27567a	available	vpc-0c7865baf88ba1857 ...	10.30.1.0/24	251	-
Default-West-2d	subnet-3f481e14	available	vpc-71dead09	172.31.48.0/20	4091	-
Default-West-2C	subnet-4c914f11	available	vpc-71dead09	172.31.0.0/20	4091	-
Default-West-2b	subnet-79e30a33	available	vpc-71dead09	172.31.32.0/20	4091	-
Default-West-2a	subnet-e622ee9e	available	vpc-71dead09	172.31.16.0/20	4091	-

Step#6

Subnet#2 – PrivateSubnet

CIDR: 10.30.2.0/24 under us-west-2c

Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag	PrivateSubnet		
VPC*	vpc-0c7865baf88ba1857		
Availability Zone	us-west-2c		
VPC CIDRs	CIDR 10.30.0.0/16	Status associated	Status Reason
IPv4 CIDR block*	10.30.2.0/24		

* Required

Cancel Create

Subnets > Create subnet

Create subnet

The following Subnet was created:

Subnet ID subnet-04196b43c3129275e

Close

New VPC Experience [Tell us what you think](#)

VPC Dashboard [New](#)

Filter by VPC: [Select a VPC](#)

VIRTUAL PRIVATE CLOUD

- Your VPCs
- Subnets**
- Route Tables
- Internet Gateways [New](#)

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
PrivateSubnet	subnet-04196b43c3129275e	available	vpc-0c7865baf88ba1857 [...]	10.30.2.0/24	251	-	us-west-2c
PublicSubnet	subnet-0d0cfea42bf27567a	available	vpc-0c7865baf88ba1857 [...]	10.30.1.0/24	251	-	us-west-2a
Default-West-2d	subnet-3f481e14	available	vpc-71dead09	172.31.48.0/20	4091	-	us-west-2d
Default-West-2C	subnet-4c914f11	available	vpc-71dead09	172.31.0.0/20	4091	-	us-west-2c
Default-West-2b	subnet-79e30a33	available	vpc-71dead09	172.31.32.0/20	4091	-	us-west-2b
Default-West-2a	subnet-e622ee9e	available	vpc-71dead09	172.31.16.0/20	4091	-	us-west-2a

Both the Subnets has been created with 251 Available IPV4 addresses.

Step#7

Now select the “PublicSubnet” and go to actions and select the Modify auto-assign IP settings.

Enable Auto Assign public IPV4 Address for the “PublicSubnet” and click on Save. This will help to create the IP address automatically when EC2 instance created.

The screenshot shows the AWS Subnets interface. The URL in the address bar is [Subnets > Modify auto-assign IP settings](#). The main title is "Modify auto-assign IP settings". Below it, a note says: "Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched in this subnet. You can override the auto-assign IP settings for an instance at launch time." A "Subnet ID" field contains "subnet-0d0cfea42bf27567a". A checkbox labeled "Auto-assign IPv4" is checked. At the bottom right are "Cancel" and "Save" buttons. A small note at the bottom left says "* Required".

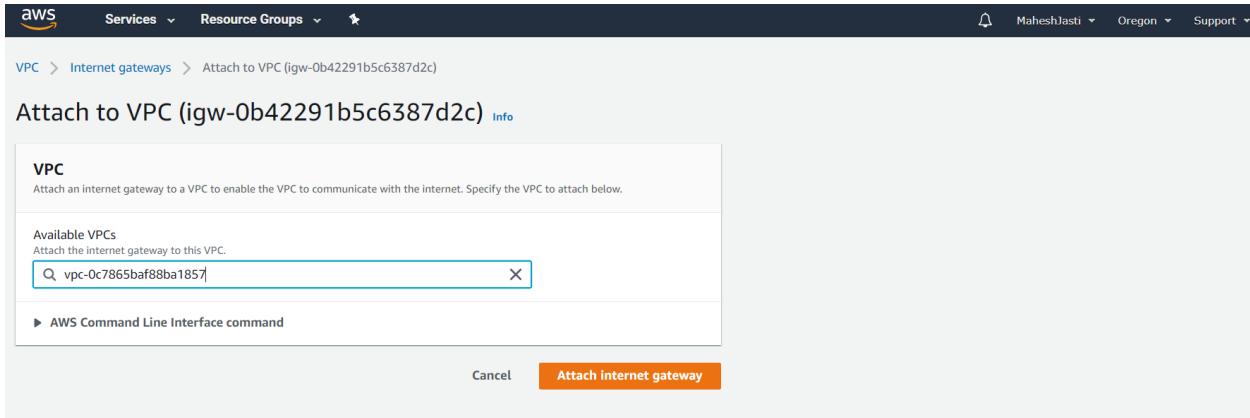
Step#8

Create an Internet Gateway. Click on the Internet Gateways option on the left menu.

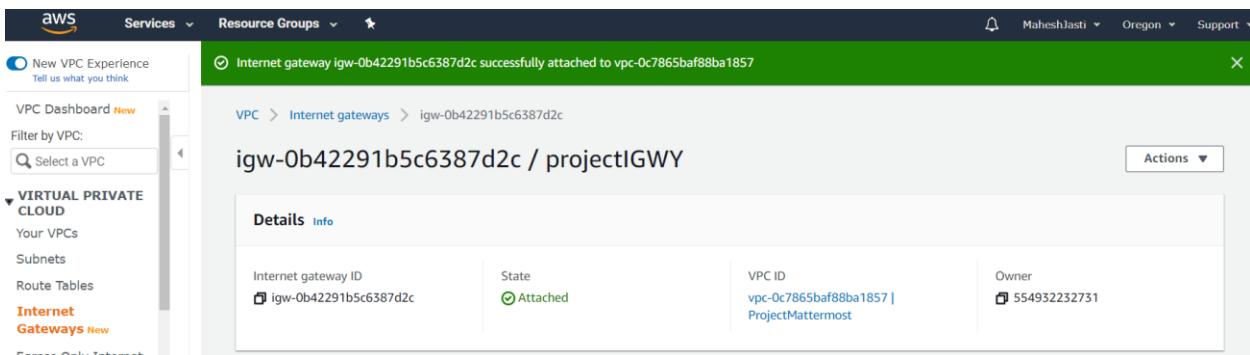
Now click on “Create Internet Gateway” at top right side of the screen. Enter the name tag as “projectIGWY” and click on Create Internet Gateway button.

The screenshot shows the AWS VPC Dashboard. The left sidebar has a "VIRTUAL PRIVATE CLOUD" section with "Internet Gateways" selected. The main area shows a message: "The following internet gateway was created: igw-0b42291b5c6387d2c. You can now attach to a VPC to enable the VPC to communicate with the internet." Below this is a breadcrumb trail: "VPC > Internet gateways > igw-0b42291b5c6387d2c". The gateway details are shown in a table: Internet gateway ID is "igw-0b42291b5c6387d2c", State is "Detached", VPC ID is "-", and Owner is "554932232731". Below the table is a "Tags" section with a table: Key "Name" and Value "projectIGWY". At the top right of the main area is a "Actions" dropdown and a "Manage tags" button.

Now attach to the ProjectMattermost VPC by clicking at the button “Attach to a VPC”

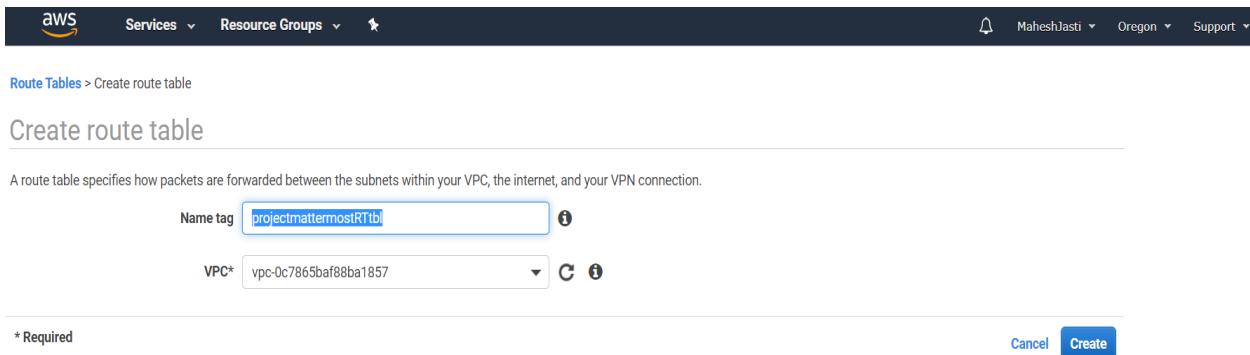


After attaching the IGWY we would see the following screen.



Step#9

Now create a new Route table (“projectmattermostRTtbl”) for the VPC by clicking on the Route Table option.



The route table is created and the “Main” attribute should show “No”

The screenshot shows the AWS VPC Route Tables page. A new route table named 'ProjectmattermostRTtbl' has been created. In the 'Main' column, it is listed as 'No', indicating it is not the primary route table for the VPC.

Step#10

Go to Routes tab and click on “Edit Routes” and add the Internet Gateway and click on Save routes button.

The screenshot shows the 'Edit routes' dialog box. A new route is being added with the destination '0.0.0.0/0' and the target 'igw-0b42291b5c6387d2c'. The 'Save routes' button is visible at the bottom right.

And should show as below after pressing the Save routes button.

The screenshot shows the AWS VPC Route Tables page again. The 'Edit routes' tab is selected for the 'ProjectmattermostRTtbl' route table. It lists two routes: one to '10.30.0.0/16' (target 'local') and another to '0.0.0.0/0' (target 'igw-0b42291b5c6387d2c'). Both routes are marked as 'active'.

Step#11

Go to “Subnet Associations” tab and click on “Edit subnet associations” button and associate the PublicSubnet.

The screenshot shows the AWS VPC Route Tables page. On the left sidebar, under the 'Route Tables' section, the 'Route Tables' link is highlighted. The main content area displays a table of route tables. One row is selected, showing the following details:

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
ProjectmattermostRTtbl	rtb-090a961af978c6a8f	subnet-0d0cfea42bf27567a	-	No	vpc-0c7865ba88ba1857 ...

Below the table, there is a section titled "The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:" which lists one subnet:

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0d0cfea42bf27567a PublicSubnet	10.30.1.0/24	-

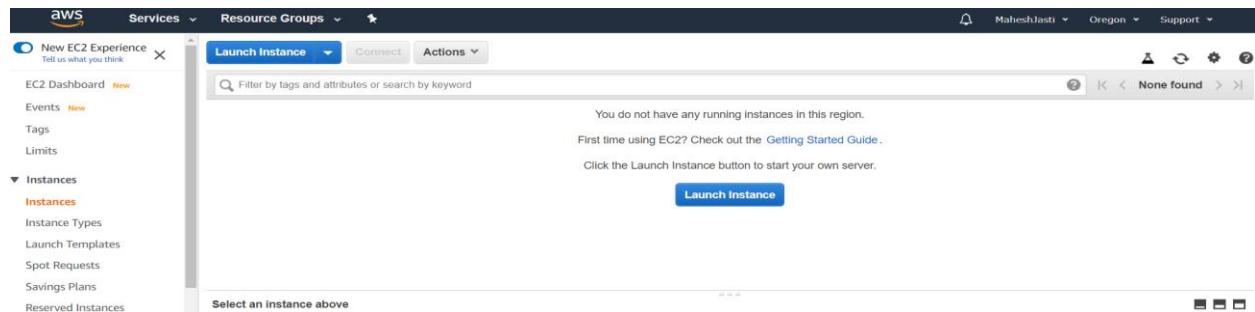
*****VPC, Subnets, IGWY and Route Tables setup is completed*****

*****Ubuntu EC2 and NAT Instances setup *****

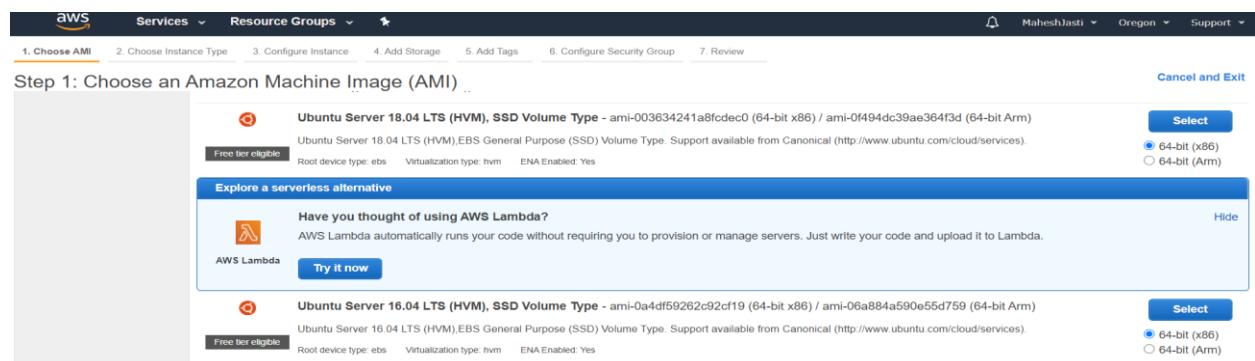
Create 2 EC2 instances – (Ubuntu 18.04 LTS)

Step#12

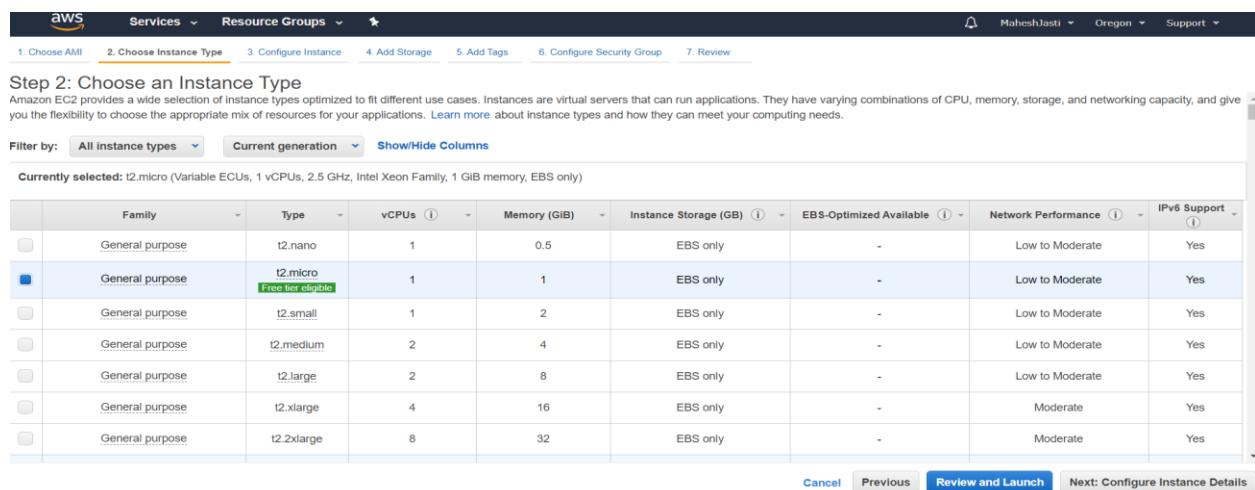
Go to EC2 window, Click on Launch Instance



Select Ubuntu 18.04 LTS AMI



And select Ubuntu Server 18.04 LTS (FreeTier)



Step#13

Select the ProjectMattermost VPC and Public Subnet and click on Add Storage button below..

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-0c7865baf88ba1857 ProjectMattermost	<input type="button" value="Create new VPC"/>
Subnet	subnet-0d0cfea42bf27567a PublicSubnet us-west-2	<input type="button" value="Create new subnet"/> 251 IP Addresses available
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	<input type="button" value="Create new Capacity Reservation"/>
IAM role	None	<input type="button" value="Create new IAM role"/>
Shutdown behavior	Stop	
Stop - Hibernate behavior	<input type="checkbox"/> Enable hibernation as an additional stop behavior	

Cancel Previous **Review and Launch** Next: Add Sto

Step#14

No changes to Storage screen

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0e54a519c999adbbd	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Step#15

Add the name tag as “App-Mattermost”

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	App-Mattermost	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Step#16

Create a security group as below. Allow SSH and HTTP (as to install the software and access the application through outside). Click on “Review and Launch” button

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security group name: SSH & HTTP

Description: Allows SSH-22 and HTTP-80

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

⚠ Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

Step#17

And click on Launch button

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, SSH & HTTP, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-003634241a8fcdec0
Free tier eligible
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root Device Type: ebs Virtualization type: hvm

Edit AMI

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Edit instance type

Security Groups

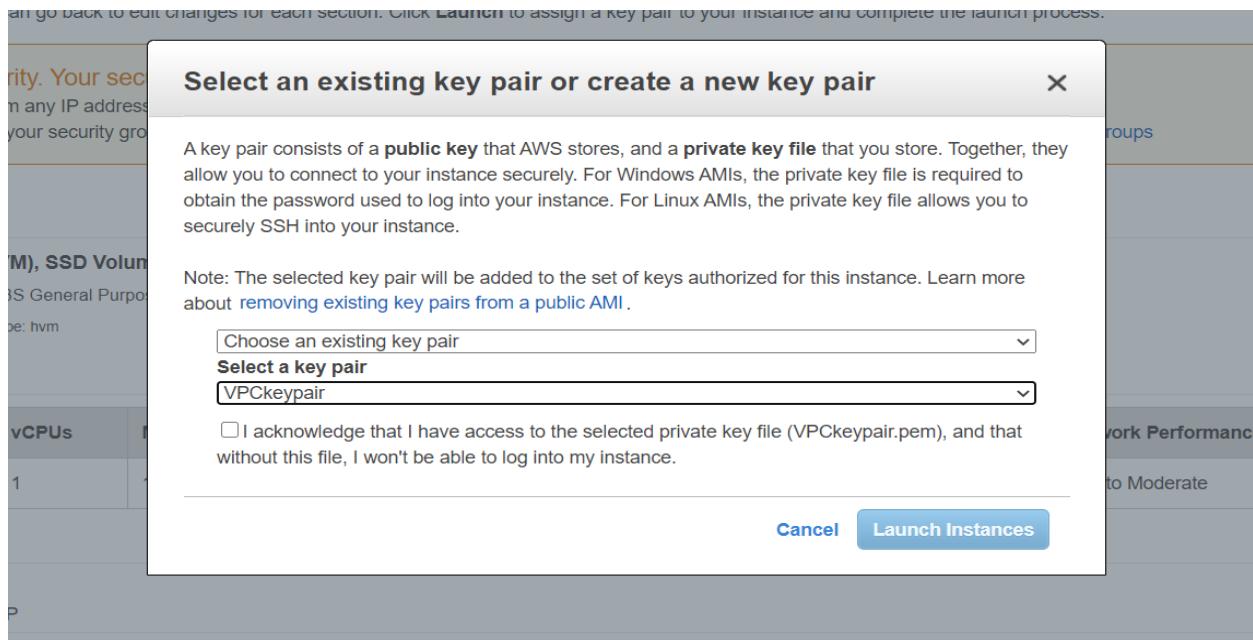
Security group name: SSH & HTTP

Edit security groups

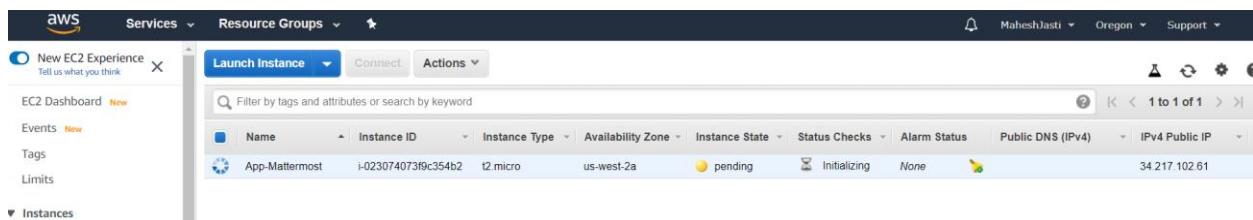
Cancel Previous Launch

Step#18

Now Choose an existing Key Pair (VPCkeypair)... Acknowledge and Click on “Launch Instances”



It will show like below.



Step#19

The Public IP addr – 52.33.55.49 and the Private IP - 10.30.1.184 has been assigned.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like 'EC2 Dashboard', 'Instances', 'Images', and 'Elastic Block Store'. The main area displays a table with one row for the instance 'App-Mattermost'. The columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Public IP. The instance details are also shown in a modal window below the table.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
App-Mattermost	i-076246c3dca56e417	t2.micro	us-west-2a	Pending	Initializing	None	-	52.33.55.49

Instance: i-076246c3dca56e417 (App-Mattermost) Public IP: 52.33.55.49

- Description
- Status Checks
- Monitoring
- Tags

Instance ID: i-076246c3dca56e417	Public DNS (IPv4): -
Instance state: pending	IPv4 Public IP: 52.33.55.49
Instance type: t2.micro	IPv6 IPs: -
Finding: Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs:
Private DNS: ip-10-30-1-184.us-west-2.compute.internal	Availability zone: us-west-2a
Private IPs: 10.30.1.184	Security groups: SSH & HTTP, view inbound rules, view outbound rules

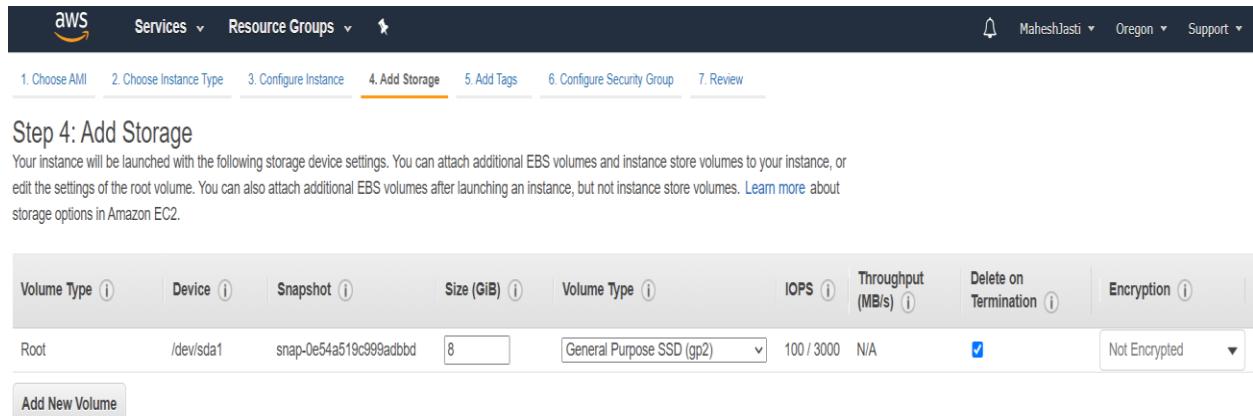
Step#20

Now launch the another Ubuntu EC2 instance and select the subnet “PrivateSubnet” (us-west-2c)

The screenshot shows the 'Configure Instance Details' step of the AWS Launch Instance wizard. The steps are numbered 1 through 7 at the top. The 'Number of instances' is set to 1. Under 'Purchasing option', there's a checkbox for 'Request Spot instances'. The 'Network' dropdown shows 'vpc-0c7865baf88ba1857 | ProjectMattermost' and a 'Create new VPC' button. The 'Subnet' dropdown shows 'subnet-04196b43c3129275e | PrivateSubnet | us-west-2c' and a 'Create new subnet' button. The 'Auto-assign Public IP' dropdown is set to 'Use subnet setting (Disable)'.

Step#21

No changes to the Add Storage screen



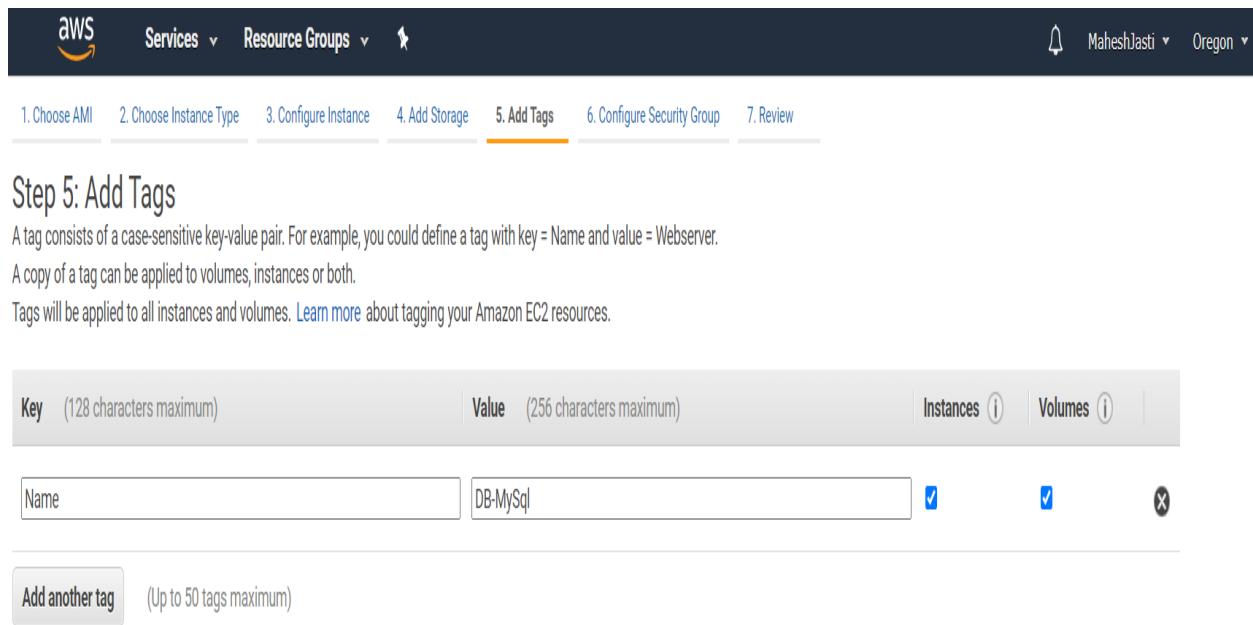
The screenshot shows the 'Add Storage' step of the AWS EC2 instance creation wizard. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information 'MaheshJasti Oregon Support'. Below the navigation is a progress bar with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (highlighted in orange), 5. Add Tags, 6. Configure Security Group, and 7. Review. The main content area is titled 'Step 4: Add Storage' with the sub-instruction: 'Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.' A table displays the current storage configuration:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0e54a51c999adbbd	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

A 'Add New Volume' button is located at the bottom left of the table.

Step#22

and enter the name tag as “DB-MySQL”



The screenshot shows the 'Add Tags' step of the AWS EC2 instance creation wizard. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information 'MaheshJasti Oregon'. Below the navigation is a progress bar with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (highlighted in orange), 6. Configure Security Group, and 7. Review. The main content area is titled 'Step 5: Add Tags' with the sub-instruction: 'A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.' A table displays the current tag configuration:

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	DB-MySQL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

A 'Add another tag' button is located at the bottom left of the table, with the note '(Up to 50 tags maximum)'.

Step#23

Select existing Security group – mysqlprivate

We want to restrict the access to the private subnet, Hence we are restricting the access to only public CIDR's

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom <input type="text" value="10.30.1.0/24"/>	e.g. SSH for Admin Desktop
MySQL/Aurora	TCP	3306	Custom <input type="text" value="10.30.1.0/24"/>	e.g. SSH for Admin Desktop

[Add Rule](#)

Step#24

Click on Launch

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details [Edit AMI](#)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-003634241a8fcdec0

Free tier eligible Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
i2.micro	Variable	1	1	EBS only	-	Low to Moderate

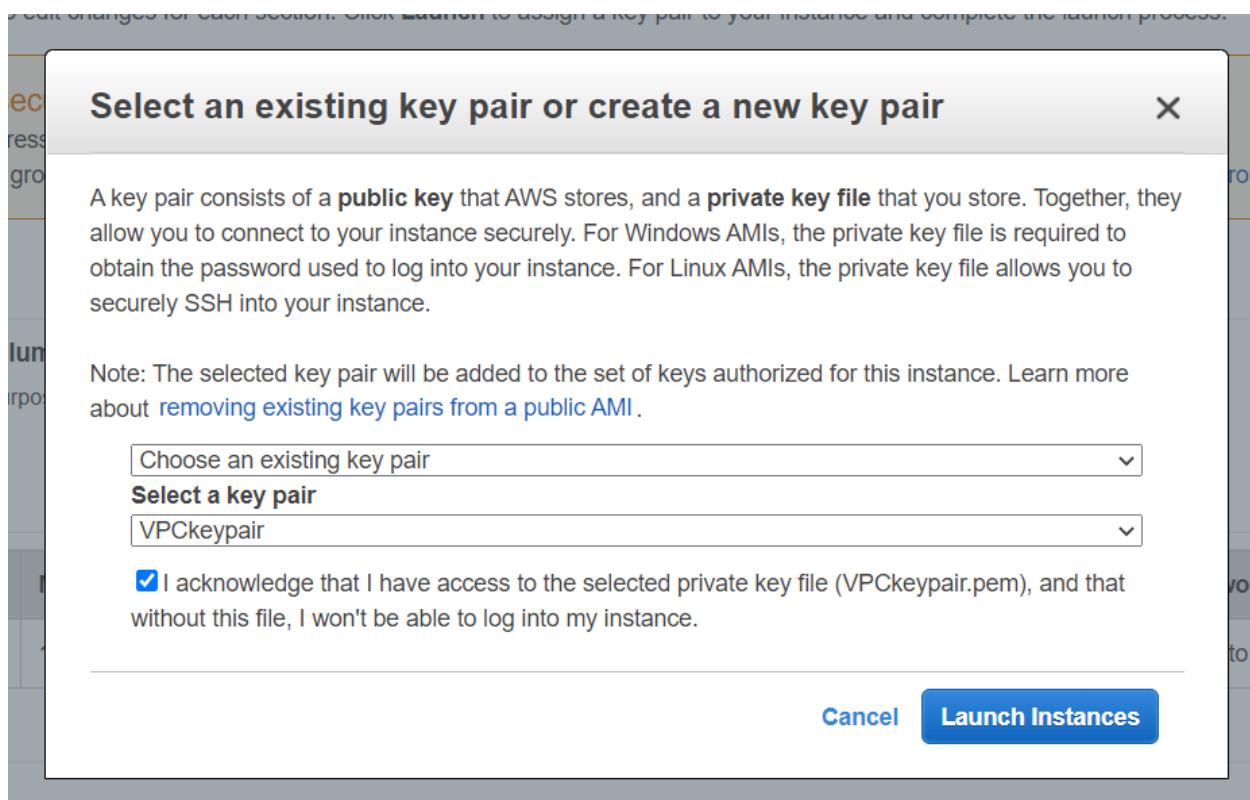
Security Groups [Edit security groups](#)

Security Group ID	Name	Description
	mysqlprivate	

[Cancel](#) [Previous](#) [Launch](#)

Step#25

Now choose an existing key pair and click on Launch Instances.



Step#26

The DB-MySQL instance didn't get the public IPV4 address as we haven't enabled the Auto Assign IPV4 Address for the Private Subnet. (Private IP: 10.30.2.225)

PGPCC – Project Implementation Task

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances, Images, AMIs, and Elastic Block Store. The main area displays a table of instances. One instance, 'App-Mattermost', is listed with a status of 'running'. Another instance, 'DB-MySQL', is listed with a status of 'Initializing'. The table includes columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Public IP.

Step#27

Create a NAT Instance in the PublicSubnet and link to the Private Subnet in the Route table.

The screenshot shows the 'Launch Instance' wizard, Step 1: Choose an Amazon Machine Image (AMI). The top navigation bar includes Services, Resource Groups, and a progress bar with steps 1. Choose AMI through 7. Review. A search bar at the top right contains the text 'NAT'. Below it, a table lists AMIs under 'Community AMIs (480)'. One entry is highlighted: 'amzn-ami-vpc-nat-hvm-2017.09.1-test|longids.20180307-x86_64-ebs - ami-0032ea5ae08aa27a2'. This entry has a 'Select' button to its right. Other columns in the table include 'Name', 'Description', 'Root device type', 'Virtualization type', and 'ENI Enabled'.

Step#28

Select default T2 Micro and select the VPC ProjectMattermost and PublicSubnet.

PGPCC – Project Implementation Task

The screenshot shows the 'Configure Instance Details' step of the AWS EC2 wizard. The user has selected '1' instance. Under 'Purchasing option', there is a checkbox for 'Request Spot instances'. The 'Network' section shows 'vpc-0c7865baf88ba1857 | ProjectMattermost' selected. Under 'Subnet', 'subnet-0d0cfea42bf27567a | PublicSubnet | us-west-2' is selected, with a note that 250 IP Addresses are available. The 'Auto-assign Public IP' dropdown is set to 'Use subnet setting (Enable)'. In the 'Placement group' section, there is a checkbox for 'Add instance to placement group'. The 'Capacity Reservation' dropdown is set to 'Open'. Under 'IAM role', 'None' is selected. The 'Shutdown behavior' dropdown is set to 'Stop'. The 'Stop - Hibernate behavior' dropdown has a checkbox for 'Enable hibernation as an additional stop behavior'. At the bottom right, there are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Add Storage'.

Step#29

No changes to Add Storage screen..

The screenshot shows the 'Add Tags' step of the AWS EC2 wizard. A single tag is being added: 'Name' with a value of 'NAT-Instance'. The 'Instances' checkbox is checked, while 'Volumes' is unchecked. There is a button to 'Add another tag'.

Step#30

Select the existing Security Group (SSH & HTTP) and launch the instance.

Note: Since it is a NAT instance, we won't login, access or install any software. No need to think about the Security Group and their rules.

PGPCC – Project Implementation Task

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
sg-06090d82ff79f3c43	default	default VPC security group	Copy to new
sg-059451afdf95e0efbf	mysqlprivate	mysqlprivate Security Group - Project Mattermost	Copy to new
sg-04f9c7fc35539a6c	SSH & HTTP	Allow SSH-22 and HTTP-80	Copy to new

Inbound rules for sg-04f9c7fc35539a6c (Selected security groups: sg-04f9c7fc35539a6c)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	0.0.0.0/0	

[Cancel](#) [Previous](#) [Review and Launch](#)

Step#31

Since we won't be logging in or installing into NAT, no need to select any key pair.

And select the drop down values as "Proceed without a key pair", Acknowledge and click on Launch Instances.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.

[Cancel](#) [Launch Instances](#)

Now NAT instance has been created (Public IP 54.70.203.137 ; Private IP: 10.30.1.239)

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Limits, Instances, Images, and Elastic Block Store. The main area displays a table of instances. One instance is selected: "NATInstance" (Instance ID: i-08546b17f2eff66d4). Below the table, detailed information about the instance is shown, including its Public DNS (54.188.78.184), Instance ID (i-08546b17f2eff66d4), Instance state (running), Instance type (t2.micro), and VPC ID (vpc-03502f64cf88c8384). The "Description" tab is selected.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
App-Mattermost	i-023074073f9c354b2	t2.micro	us-west-2a	running	2/2 checks ...	None	-
DB-MySQL	i-091179bcfc83ff55	t2.micro	us-west-2c	running	2/2 checks ...	None	-
NATInstance	i-08546b17f2eff66d4	t2.micro	us-west-2a	running	2/2 checks ...	None	-

Step#32

Now Disable the Change Source/Dest. Check for the NAT Instance..

The screenshot shows the AWS EC2 Instances page with the same setup as before. The "NATInstance" is selected. A context menu is open over the instance row, with the "Actions" dropdown expanded. The "Change Source/Dest. Check" option is highlighted with a yellow arrow. Other options in the menu include Connect, Get Windows Password, Create Template From Instance, Launch More Like This, Instance State, Instance Settings, Image, Networking, CloudWatch Monitoring, Change Security Groups, Attach Network Interface, Detach Network Interface, Disassociate Elastic IP Address, Change Source/Dest. Check, and Manage IP Addresses.

Enable Source/Destination Check

X

Are you sure that you would like to disable Source/Destination Check for the instance with the following details:

Instance: i-05a383e183aecd607 (NAT-Instance)
Network Interface: eni-0c5f7a456d45fe23f
Status Enabled

Cancel

Yes, Disable

*****Ubuntu EC2 and NAT Instances setup done*****

*****Route table setup for PrivateSubnet*****

Step#33

Go to the Route Tables and create a new route table Nainstance-RT Table-Private and map to the PrivateSubnet.

Route Tables > Create route table

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag	Nainstance-RT Table-Private	i
VPC*	vpc-0c7865baf88ba1857	C i

* Required

[Cancel](#) [Create](#)

New VPC Experience Tell us what you think

VPC Dashboard [New](#)

Filter by VPC: [Select a VPC](#)

VIRTUAL PRIVATE CLOUD

- Your VPCs
- Subnets
- Route Tables**
- Internet Gateways [New](#)
- Egress Only Internet Gateways [New](#)
- DHCP Options Sets [New](#)
- Elastic IPs [New](#)
- Managed Prefix Lists [New](#)
- Endpoints
- Endpoint Services
- NAT Gateways

Create route table Actions ▾

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
Nainstance-RT Table-Private	rtb-01b48d6c7877fb5c9	-	-	No	vpc-0c7865baf88ba1857 ...
VPC Default Route table	rtb-05690431412472d35	-	-	Yes	vpc-0c7865baf88ba1857 ...
ProjectmattermostRTtbl	rtb-090a961af978c6a8f	subnet-0d0cfea42bf27567a	-	No	vpc-0c7865baf88ba1857 ...
Default Route Table	rtb-128d469	-	-	Yes	vpc-71deadd9

Route Table: rtb-01b48d6c7877fb5c9

Summary **Routes** Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

Destination	Target	Status	Propagated
10.30.0.0/16	local	active	No

Step#34

Click on Edit routes and Add route as below and map the target as NAT Instance and click on Save routes.

Destination	Target	Status	Propagated
10.30.0.0/16	local	active	No
0.0.0.0/0	i-05a383e183aec607	active	No

Add route

* Required

Cancel Save routes

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
NAtinstance-RT Table-Private	rtb-01b48d6c7877fb5c9	-	-	No	vpc-0c7865ba88ba1857 ...
VPC Default Route table	rtb-05690431412472d35	-	-	Yes	vpc-0c7865ba88ba1857 ...
ProjectmattermostRTtbl	rtb-090a961af978c6a8f	subnet-0d0cfea42bf27567a	-	No	vpc-0c7865ba88ba1857 ...
Default Route Table	rtb-1288d469	-	-	Yes	vpc-71dead09

Route Table: rtb-01b48d6c7877fb5c9

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

Destination	Target	Status	Propagated
10.30.0.0/16	local	active	No
0.0.0.0/0	eni-0c5f7a456d45fe23f	active	No

Step#35

Associate the PrivateSubnet in Edit subnet associations tab and click on Save button

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-04196b43c3129275e PrivateSubnet	10.30.2.0/24	-	Main
subnet-0d0cfea42bf27567a PublicSubnet	10.30.1.0/24	-	rtb-090a961af978c6a8f

* Required Cancel Save

Subnet ID associated after clicking on the Save button

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
Natinstance-RT Table-Private	rtb-01b48d6c7877fb5c9	subnet-04196b43c3129275e	-	No	vpc-0c7865ba88ba1857 ...
VPC Default Route table	rtb-05690431412472d35	-	-	Yes	vpc-0c7865ba88ba1857 ...
ProjectmattermostRTtbl	rtb-090a961af978c6a8f	subnet-0d0cfea42bf27567a	-	No	vpc-0c7865ba88ba1857 ...
Default Route Table	rtb-1288d469	-	-	Yes	vpc-71dead09

*****Route table setup for PrivateSubnet setup done*****

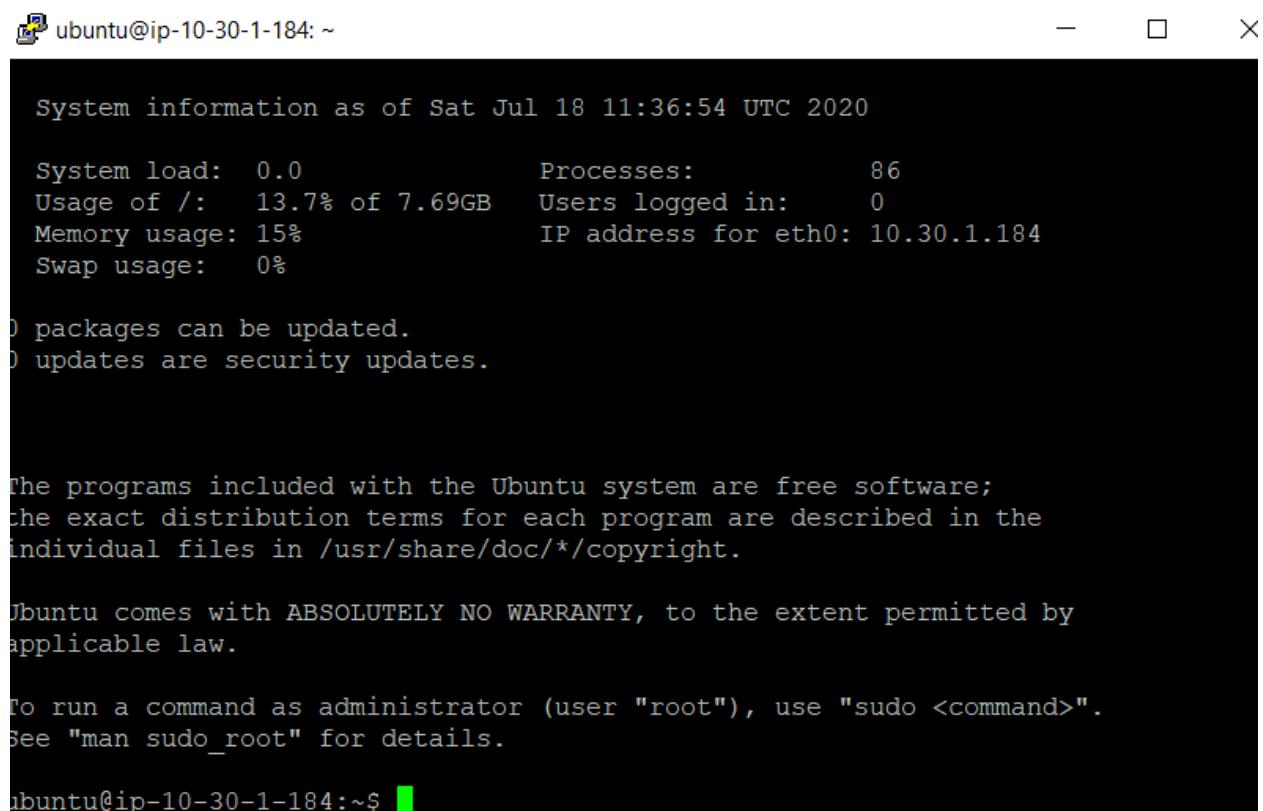
*****EC2 instances – Access check *****

Step#36

Until this we have setup IGWY to Public Subnet and NAT Instance to the Private Subnet.

Now login to the Public Instance thru Putty – ubuntu@52.33.55.49

10.30.1.184



ubuntu@ip-10-30-1-184: ~

```
System information as of Sat Jul 18 11:36:54 UTC 2020

System load:  0.0          Processes:      86
Usage of /:   13.7% of 7.69GB  Users logged in:  0
Memory usage: 15%
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

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.

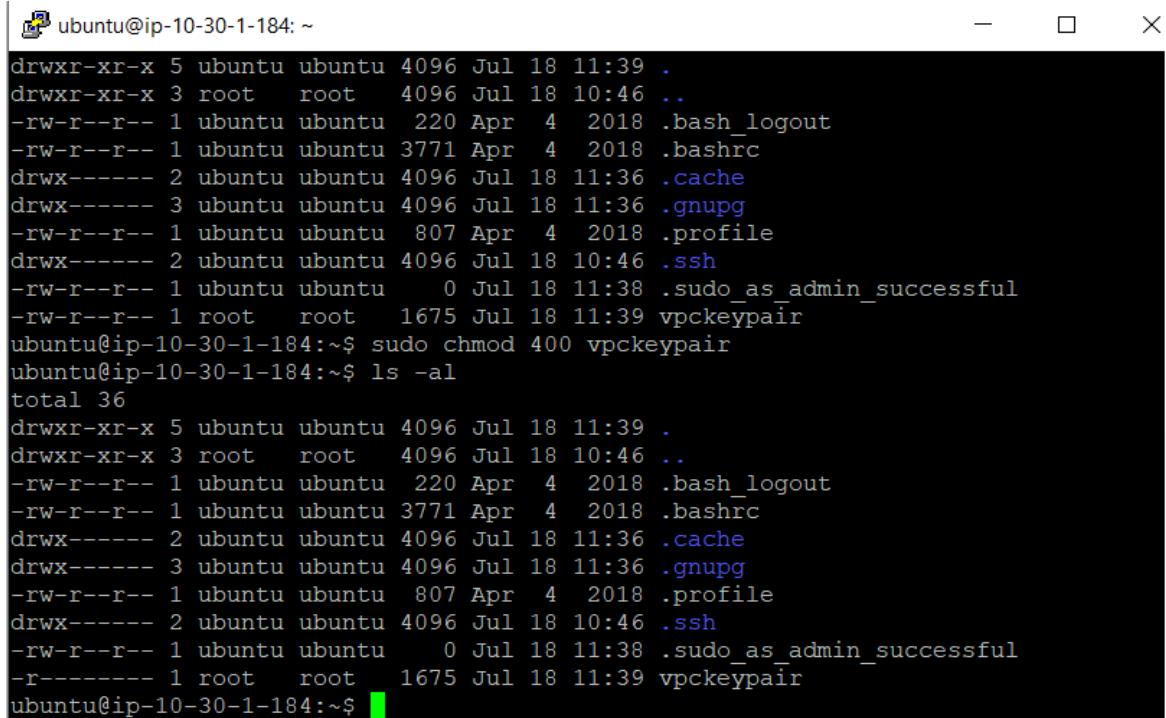
ubuntu@ip-10-30-1-184:~$
```

Step#37

We need to login to the 10.30.2.225 Private instance to install the MySQL Server. So we need to SSH from 10.30.1.184. We need the key to login. So create a key file(vpckeypair) and change the permissions of the file to Read only and execute the following commands

sudo nano vpckeypair

sudo chmod 400 vpckeypair



```

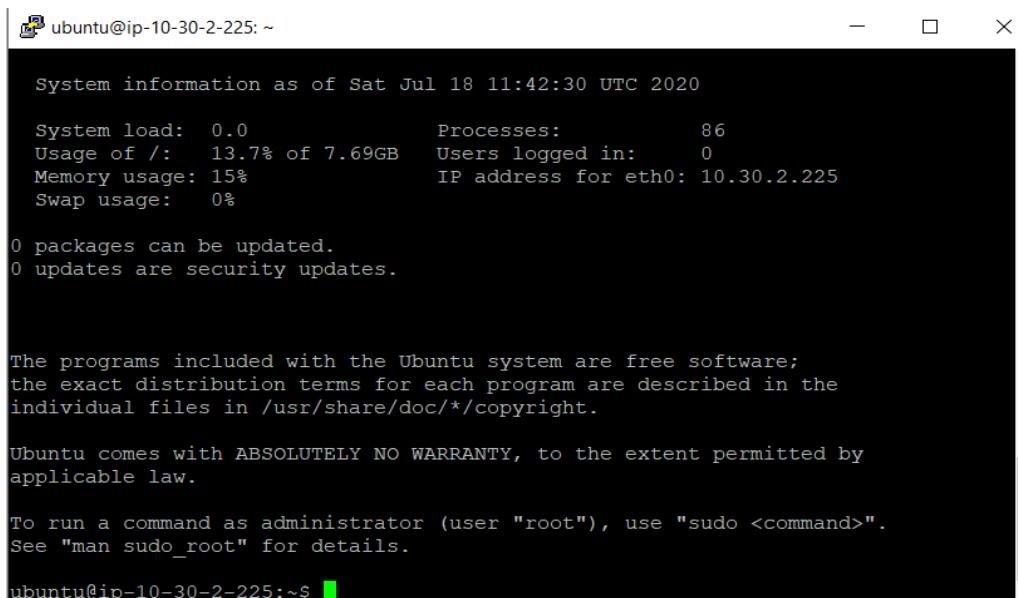
ubuntu@ip-10-30-1-184: ~
drwxr-xr-x 5 ubuntu ubuntu 4096 Jul 18 11:39 .
drwxr-xr-x 3 root root 4096 Jul 18 10:46 ..
-rw-r--r-- 1 ubuntu ubuntu 220 Apr 4 2018 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Apr 4 2018 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Jul 18 11:36 .cache
drwx----- 3 ubuntu ubuntu 4096 Jul 18 11:36 .gnupg
-rw-r--r-- 1 ubuntu ubuntu 807 Apr 4 2018 .profile
drwx----- 2 ubuntu ubuntu 4096 Jul 18 10:46 .ssh
-rw-r--r-- 1 ubuntu ubuntu 0 Jul 18 11:38 .sudo_as_admin_successful
-rw-r--r-- 1 root root 1675 Jul 18 11:39 vpckeypair
ubuntu@ip-10-30-1-184:~$ sudo chmod 400 vpckeypair
ubuntu@ip-10-30-1-184:~$ ls -al
total 36
drwxr-xr-x 5 ubuntu ubuntu 4096 Jul 18 11:39 .
drwxr-xr-x 3 root root 4096 Jul 18 10:46 ..
-rw-r--r-- 1 ubuntu ubuntu 220 Apr 4 2018 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Apr 4 2018 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Jul 18 11:36 .cache
drwx----- 3 ubuntu ubuntu 4096 Jul 18 11:36 .gnupg
-rw-r--r-- 1 ubuntu ubuntu 807 Apr 4 2018 .profile
drwx----- 2 ubuntu ubuntu 4096 Jul 18 10:46 .ssh
-rw-r--r-- 1 ubuntu ubuntu 0 Jul 18 11:38 .sudo_as_admin_successful
-rw-r--r-- 1 root root 1675 Jul 18 11:39 vpckeypair
ubuntu@ip-10-30-1-184:~$ 

```

Step#38

Execute the following command to move to Private instance

`sudo ssh -i vpckeypair ubuntu@10.30.2.225`



```

ubuntu@ip-10-30-2-225: ~
System information as of Sat Jul 18 11:42:30 UTC 2020

System load: 0.0          Processes: 86
Usage of /: 13.7% of 7.69GB  Users logged in: 0
Memory usage: 15%          IP address for eth0: 10.30.2.225
Swap usage: 0%

0 packages can be updated.
0 updates are security updates.

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.

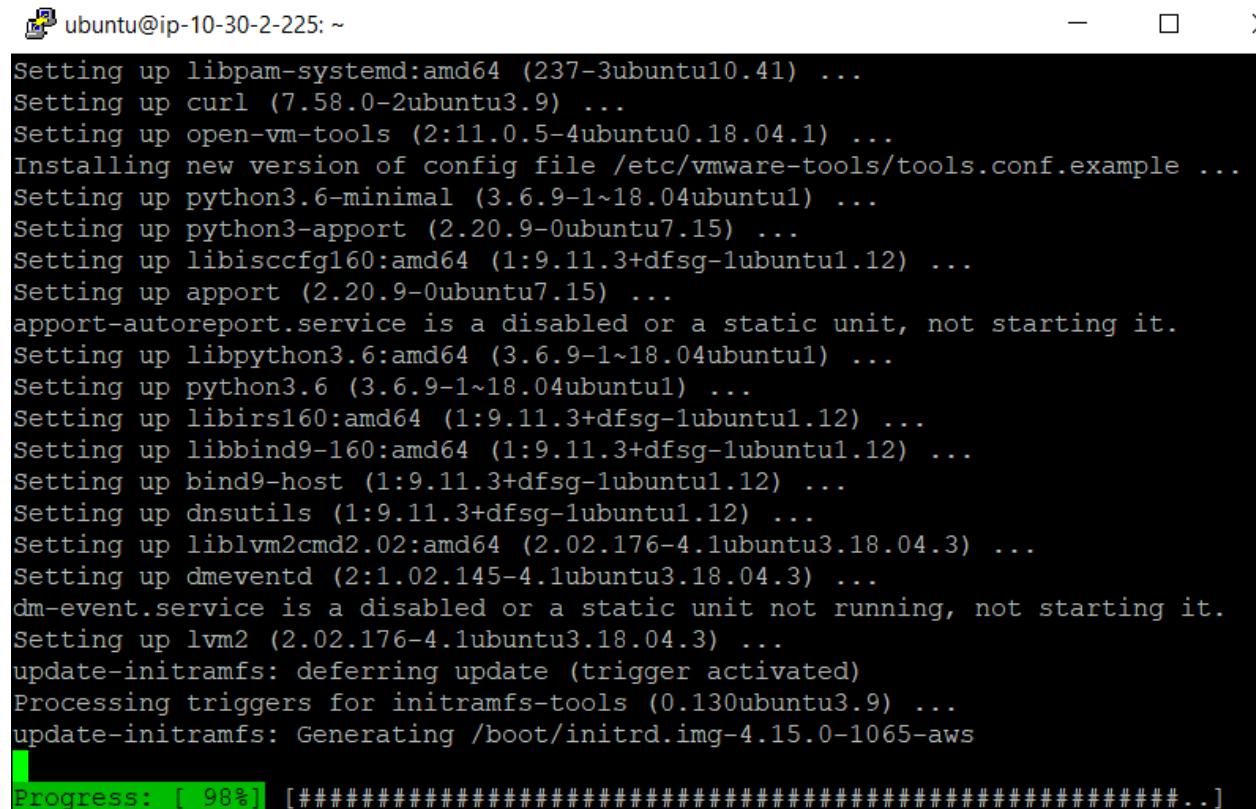
ubuntu@ip-10-30-2-225:~$ 

```

Step#39

You are into Ubuntu machine (Private Subnet) and execute the following 2 below commands..

sudo apt update and sudo apt upgrade (**if you don't upgrade, you will have some problems in installing MySql-Server**)



```
ubuntu@ip-10-30-2-225: ~
Setting up libpam-systemd:amd64 (237-3ubuntu10.41) ...
Setting up curl (7.58.0-2ubuntu3.9) ...
Setting up open-vm-tools (2:11.0.5-4ubuntu0.18.04.1) ...
Installing new version of config file /etc/vmware-tools/tools.conf.example ...
Setting up python3.6-minimal (3.6.9-1~18.04ubuntul) ...
Setting up python3-apport (2.20.9-0ubuntu7.15) ...
Setting up libiscfg160:amd64 (1:9.11.3+dfsg-1ubuntu1.12) ...
Setting up apport (2.20.9-0ubuntu7.15) ...
apport-autoreport.service is a disabled or a static unit, not starting it.
Setting up libpython3.6:amd64 (3.6.9-1~18.04ubuntul) ...
Setting up python3.6 (3.6.9-1~18.04ubuntul) ...
Setting up libirs160:amd64 (1:9.11.3+dfsg-1ubuntu1.12) ...
Setting up libbind9-160:amd64 (1:9.11.3+dfsg-1ubuntu1.12) ...
Setting up bind9-host (1:9.11.3+dfsg-1ubuntu1.12) ...
Setting up dnsutils (1:9.11.3+dfsg-1ubuntu1.12) ...
Setting up liblvm2cmd2.02:amd64 (2.02.176-4.1ubuntu3.18.04.3) ...
Setting up dmeventd (2:1.02.145-4.1ubuntu3.18.04.3) ...
dm-event.service is a disabled or a static unit not running, not starting it.
Setting up lvm2 (2.02.176-4.1ubuntu3.18.04.3) ...
update-initramfs: deferring update (trigger activated)
Processing triggers for initramfs-tools (0.130ubuntu3.9) ...
update-initramfs: Generating /boot/initrd.img-4.15.0-1065-aws
[Progress: [ 98%] [##########################################...]
```

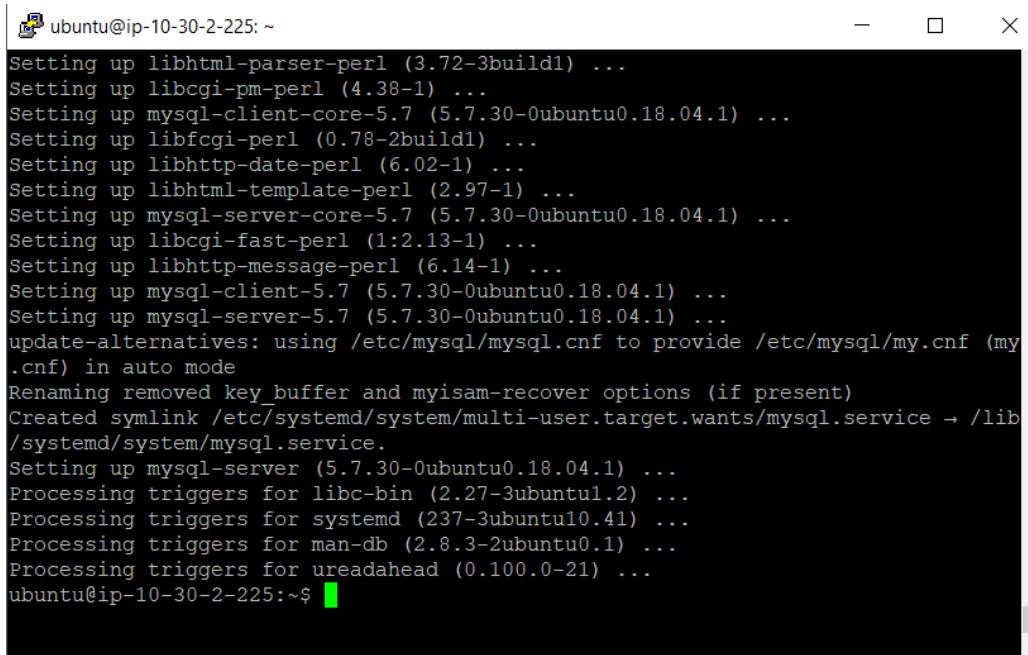
*****EC2 instances – Access check - Completed *****

*****Installation of Mysql & Mattermost softwares – Setup *****

Step#40

Install MySQL Server. Use the following command to install.

```
sudo apt install mysql-server
```



A terminal window showing the output of the command 'sudo apt install mysql-server'. The window has a title bar with a logo and the text 'ubuntu@ip-10-30-2-225: ~'. The main area contains the log output of the package installation process, including dependency resolution and configuration steps. The window has standard window controls (minimize, maximize, close) at the top right.

```
ubuntu@ip-10-30-2-225: ~
Setting up libhtml-parser-perl (3.72-3build1) ...
Setting up libcgi-pm-perl (4.38-1) ...
Setting up mysql-client-core-5.7 (5.7.30-0ubuntu0.18.04.1) ...
Setting up libfcgi-perl (0.78-2build1) ...
Setting up libhttp-date-perl (6.02-1) ...
Setting up libhtml-template-perl (2.97-1) ...
Setting up mysql-server-core-5.7 (5.7.30-0ubuntu0.18.04.1) ...
Setting up libcgi-fast-perl (1:2.13-1) ...
Setting up libhttp-message-perl (6.14-1) ...
Setting up mysql-client-5.7 (5.7.30-0ubuntu0.18.04.1) ...
Setting up mysql-server-5.7 (5.7.30-0ubuntu0.18.04.1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up mysql-server (5.7.30-0ubuntu0.18.04.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Processing triggers for systemd (237-3ubuntu10.41) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
ubuntu@ip-10-30-2-225:~$
```

Step#41

Running mysql_secure_installation – To check the password settings and setup root password and admin settings.

```
sudo mysql_secure_installation (sudo – should be in small letters)
```

```
ubuntu@ip-10-30-2-225: ~
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No)
: Y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-10-30-2-225:~$
```

Step#42

sudo mysql

Created an User and the password and also set only the public subnet which has the mattermost app to access the MySQL which is in private subnet.

```
create user 'mmuser'@'%' identified by 'maheshjast';
```

```
create database mattermost;
```

```
GRANT ALTER, CREATE, DELETE, DROP, INDEX, INSERT, SELECT,
UPDATE ON mattermost.* TO 'mmuser'@'%';
```

```

ubuntu@ip-10-30-2-225:~$ sudo mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.30-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user 'mmuser'@'%' identified by 'maheshjast';
Query OK, 0 rows affected (0.00 sec)

mysql> create database mattermost;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALTER, CREATE, DELETE, DROP, INDEX, INSERT, SELECT, UPDATE ON matte
rmost.* TO 'mmuser'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql>

```

Step#43

Exit mysql

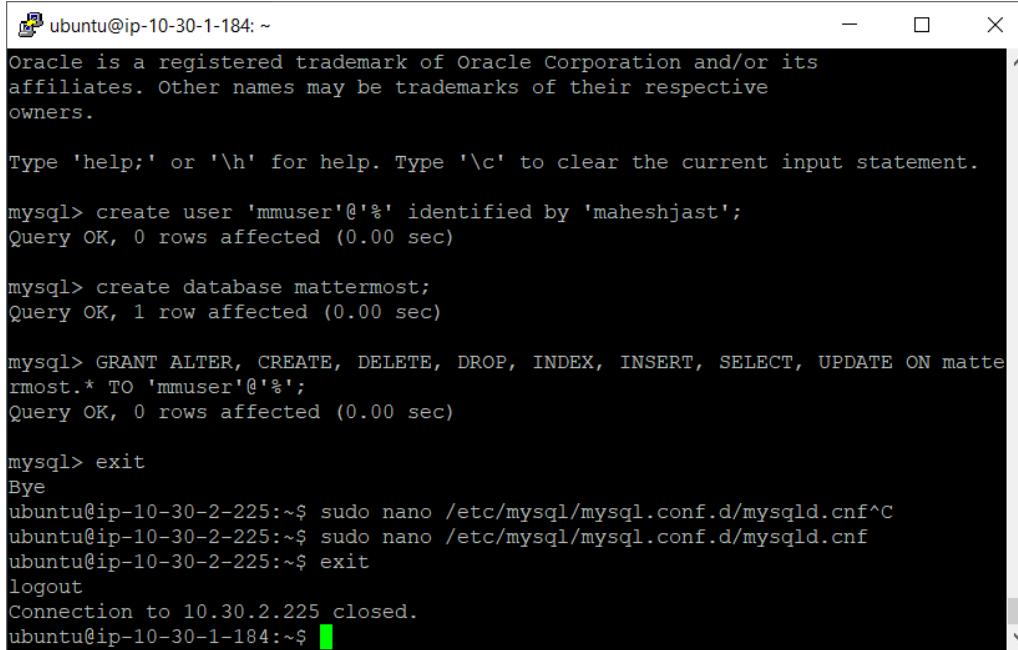
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

Comment the bind-address 127.0.0.1

```

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
# bind-address        = 127.0.0.1
#
# Fine Tuning
#
key_buffer_size      = 16M
max_allowed_packet   = 16M
thread_stack         = 192K

```



```

ubuntu@ip-10-30-1-184: ~
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user 'mmuser'@'%' identified by 'maheshjast';
Query OK, 0 rows affected (0.00 sec)

mysql> create database mattermost;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALTER, CREATE, DELETE, DROP, INDEX, INSERT, SELECT, UPDATE ON matte
rmost.* TO 'mmuser'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
ubuntu@ip-10-30-2-225:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf^C
ubuntu@ip-10-30-2-225:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
ubuntu@ip-10-30-2-225:~$ exit
logout
Connection to 10.30.2.225 closed.
ubuntu@ip-10-30-1-184:~$ 

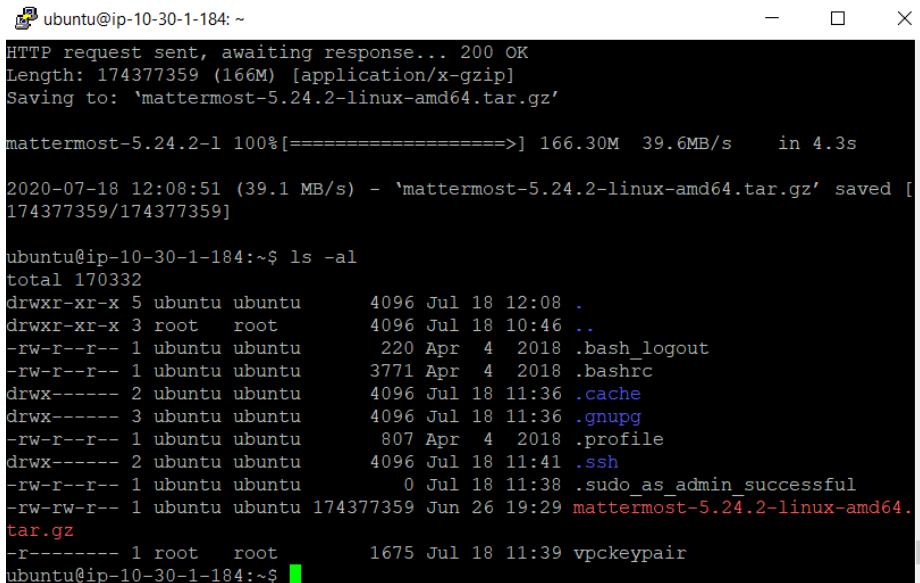
```

Step#44

Exit the private instance and install the mattermost in the public instance by using the following command

sudo apt update and sudo apt upgrade

wget <https://releases.mattermost.com/5.24.2/mattermost-5.24.2-linux-amd64.tar.gz>



```

ubuntu@ip-10-30-1-184: ~
HTTP request sent, awaiting response... 200 OK
Length: 174377359 (166M) [application/x-gzip]
Saving to: 'mattermost-5.24.2-linux-amd64.tar.gz'

mattermost-5.24.2-1 100%[=====] 166.30M 39.6MB/s    in 4.3s

2020-07-18 12:08:51 (39.1 MB/s) - 'mattermost-5.24.2-linux-amd64.tar.gz' saved [174377359/174377359]

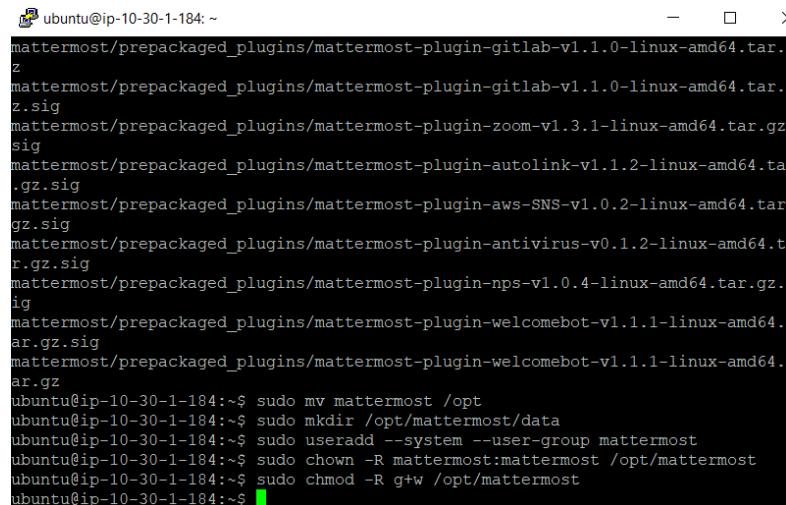
ubuntu@ip-10-30-1-184:~$ ls -al
total 170332
drwxr-xr-x 5 ubuntu ubuntu    4096 Jul 18 12:08 .
drwxr-xr-x 3 root   root     4096 Jul 18 10:46 ..
-rw-r--r-- 1 ubuntu ubuntu    220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu   3771 Apr  4  2018 .bashrc
drwx----- 2 ubuntu ubuntu   4096 Jul 18 11:36 .cache
drwx----- 3 ubuntu ubuntu   4096 Jul 18 11:36 .gnupg
-rw-r--r-- 1 ubuntu ubuntu    807 Apr  4  2018 .profile
drwx----- 2 ubuntu ubuntu   4096 Jul 18 11:41 .ssh
-rw-r--r-- 1 ubuntu ubuntu      0 Jul 18 11:38 .sudo_as_admin_successful
-rw-rw-r-- 1 ubuntu ubuntu 174377359 Jun 26 19:29 mattermost-5.24.2-linux-amd64.
tar.gz
-r----- 1 root   root     1675 Jul 18 11:39 vpckeypair
ubuntu@ip-10-30-1-184:~$ 

```

Step#45

And follow the commands to configure the Mattermost application....

```
tar -xvzf mattermost*.gz  
sudo mv mattermost /opt  
sudo mkdir /opt/mattermost/data  
sudo useradd --system --user-group mattermost  
sudo chown -R mattermost:mattermost /opt/mattermost  
sudo chmod -R g+w /opt/mattermost
```



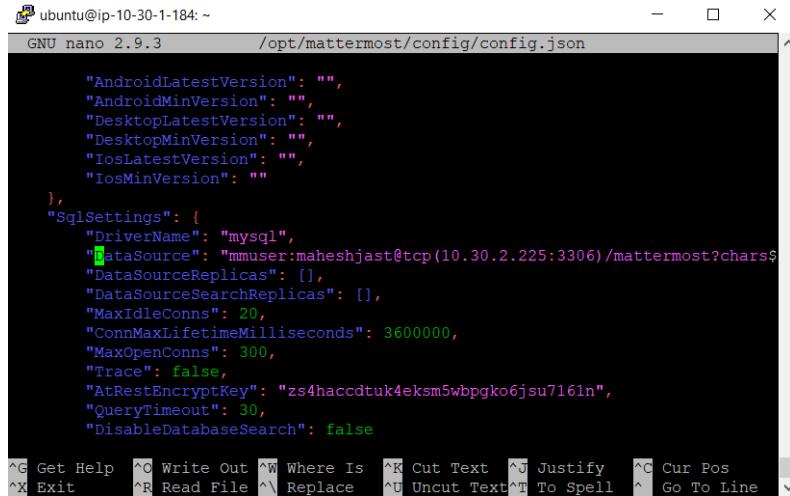
The screenshot shows a terminal window on an Ubuntu system (version 18.04) with the title bar "ubuntu@ip-10-30-1-184: ~". The terminal displays the following command history:

```
mattermost/prepackaged_plugins/mattermost-plugin-gitlab-v1.1.0-linux-amd64.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-gitlab-v1.1.0-linux-amd64.tar.gz.sig  
mattermost/prepackaged_plugins/mattermost-plugin-zoom-v1.3.1-linux-amd64.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-zoom-v1.3.1-linux-amd64.tar.gz.sig  
mattermost/prepackaged_plugins/mattermost-plugin-autolink-v1.1.2-linux-amd64.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-autolink-v1.1.2-linux-amd64.tar.gz.sig  
mattermost/prepackaged_plugins/mattermost-plugin-aws-SNS-v1.0.2-linux-amd64.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-aws-SNS-v1.0.2-linux-amd64.tar.gz.sig  
mattermost/prepackaged_plugins/mattermost-plugin-antivirus-v0.1.2-linux-amd64.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-antivirus-v0.1.2-linux-amd64.tar.gz.sig  
mattermost/prepackaged_plugins/mattermost-plugin-nps-v1.0.4-linux-amd64.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-welcomebot-v1.1.1-linux-amd64.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-welcomebot-v1.1.1-linux-amd64.tar.gz.sig  
ubuntu@ip-10-30-1-184:~$ sudo mv mattermost /opt  
ubuntu@ip-10-30-1-184:~$ sudo mkdir /opt/mattermost/data  
ubuntu@ip-10-30-1-184:~$ sudo useradd --system --user-group mattermost  
ubuntu@ip-10-30-1-184:~$ sudo chown -R mattermost:mattermost /opt/mattermost  
ubuntu@ip-10-30-1-184:~$ sudo chmod -R g+w /opt/mattermost  
ubuntu@ip-10-30-1-184:~$
```

Step#46

Setup the database drive and update the datasource details with the user details that has been created for mysql

```
sudo nano /opt/mattermost/config/config.json
```



```
ubuntu@ip-10-30-1-184: ~
GNU nano 2.9.3          /opt/mattermost/config/config.json

    "AndroidLatestVersion": "",
    "AndroidMinVersion": "",
    "DesktopLatestVersion": "",
    "DesktopMinVersion": "",
    "IosLatestVersion": "",
    "IosMinVersion": ""

},
"SqlSettings": {
    "DriverName": "mysql",
    "DataSource": "mmuser:maheshjast@tcp(10.30.2.225:3306)/mattermost?charset=utf8&parseTime=True&loc=Local",
    "DataSourceReplicas": [],
    "DataSourceSearchReplicas": [],
    "MaxIdleConns": 20,
    "ConnMaxLifetimeMilliseconds": 3600000,
    "MaxOpenConns": 300,
    "Trace": false,
    "AtRestEncryptKey": "zs4haccdtuk4eksm5wbpgko6jsu7161n",
    "QueryTimeout": 30,
    "DisableDatabaseSearch": false
}

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^ Go To Line
```

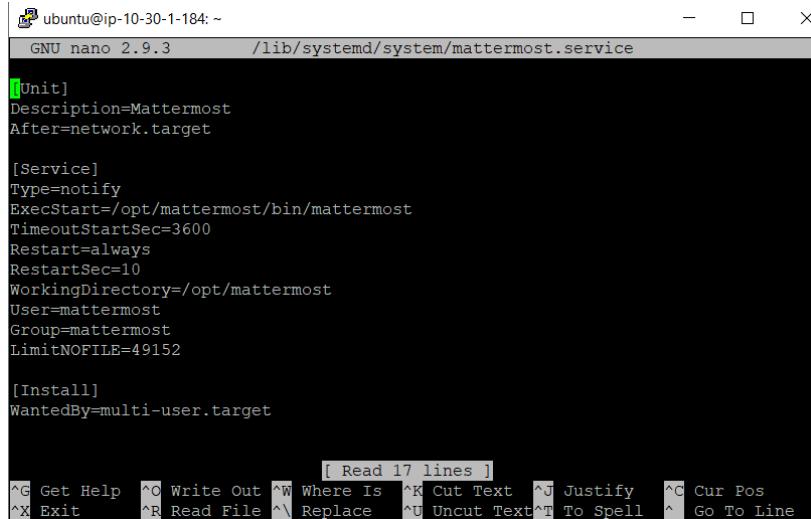
Step#47

Execute the following commands to setup the mattermost service

cd /opt/mattermost

sudo -u mattermost ./bin/mattermost

sudo nano /lib/systemd/system/mattermost.service (update the file as per the comments)



```
ubuntu@ip-10-30-1-184: ~
GNU nano 2.9.3          /lib/systemd/system/mattermost.service

[Unit]
Description=Mattermost
After=network.target

[Service]
Type=notify
ExecStart=/opt/mattermost/bin/mattermost
TimeoutStartSec=3600
Restart=always
RestartSec=10
WorkingDirectory=/opt/mattermost
User=mattermost
Group=mattermost
LimitNOFILE=49152

[Install]
WantedBy=multi-user.target

[ Read 17 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^ Go To Line
```

```
sudo systemctl daemon-reload
sudo systemctl status mattermost.service
sudo systemctl start mattermost.service
sudo systemctl enable mattermost.service
```

Step#47

Now update the security group for App-mattermost EC2 instance(Bastion host) with the following

Open the TCP for the port 8065. This is very important step.

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	::/0	-
SSH	TCP	22	0.0.0.0/0	-
Custom TCP	TCP	8065	0.0.0.0/0	-

Step#48

Now open a browser and go to <http://52.33.55.49:8065/>

The screenshot shows a web browser window with the Mattermost sign-up page. The URL in the address bar is http://52.33.55.49:8065/signup_email. The page has a light gray header with various links like 'Basic: Exerci...', 'Home - Tableau On...', 'Mathematics for M...', 'AWS Certified Deve...', 'CloudBees University', 'Installing Mattermo...', 'AWS CLI Command', 'Terraform', and 'Vi'. The main content area has a heading 'Mattermost' and a sub-headline 'All team communication in one place, searchable and accessible anywhere'. It says 'Let's create your account' and provides a link 'Click here to sign in.' Below this, there are three input fields: 'What's your email address?' (with a placeholder and validation message 'Valid email required for sign-up'), 'Choose your username' (with a placeholder and validation message 'You can use lowercase letters, numbers, periods, dashes, and underscores.'), and 'Choose your password' (with a placeholder). A blue 'Create Account' button is at the bottom. A small note at the bottom of the form says 'By proceeding to create your account and use Mattermost.'

*****Installation of Mysql & Mattermost softwares – Setup completed*****

Section 6: Lessons learnt / Observations

- Security Groups setup is very important to allow / deny the traffic. Due to 1 small wrong setting in SG, I literally googling for almost 10hours on mattermost community.
- Changes in Security Group(s) will take immediate effect to the instances available in the respective VPCs
- **Security Group operates at instance level and supports allow rules only.**
- **Instances associated with Security Group can't talk to each other unless you add rules allowing in the Security Group**
- VPC and Subnets setup can remain available instead of deleting but where as EC2 instances should be stopped.
- VPC automatically comes with a main route table that can modify
- A subnet can be associated with only one network ACL at a time. If you associate a new one, then the previous subnet association will be removed.
- **Multiple subnets can be associated with network ACLs.**
- **Can associate multiple subnets with the same route table.**
- **A subnet can only be associated with one route table.**
- Since we have to install the softwares in Ubuntu machines, I searched for Ubuntu NAT instances (which has name as natt) and created a setup but it didn't work. As per AWS, only LINUX NAT instances should be used. (**AWS recommends the instances with the name amzn-ami-vpc-nat as NAT instances only**)
- As we know NAT instance doesn't require elastic IP unless the auto assign IP address option is not checked. If the auto assign IP option is checked and instance initiated then if we create the elastic IP, AWS assign the same public IP which is there for EC2 NAT instance.

Public IPv4 address	Allocation ID	Associated instance ID	Private IP address	Association ID
35.167.86.116	eipalloc-0b3e695a271e56666	i-0423ab81f65369139	10.40.1.184	eipassoc-03d46840cda6ce7dl
InstanceNAT	i-0423ab81f65369139	t2.micro	us-west-2a	● running ✓ 2/2 checks ... None ? 35.167.86.116

- Since we are installing mattermost and mysql in public and private EC2 machines respectively, we need to make sure we point the correct IP address where mysql is installed in the config.json file
- Proper naming convention should be used for EC2 names, Security Groups, Subnet names. Since we would be creating multiple of these, I sometimes confused especially with the SG names.
- Create the flow steps based on the architecture which helps in the setup creation fastly.

- **Can connect to the private instance (mysql server) in VPC2 and access the mattermost application from public instance which is in VPC1 (10.40.1.0 – VPC1 <> 10.60.1.0 – VPC2) by using VPC peering connection.**
- Route tables should be updated with respective to the VPC peering ID for both VPCs