**PROJECT 1**

**Implementing a Team Communication Solution Using Mattermost and AWS**

1. CREATING VPC

* Go to AWS Management Console -> Services->VPC
* Click your VPC->create vpc
* On open form fill name tag
* Fill IPv4 CIDR block block (10.200.0.0/16)
* Click create vpc
* Now vpc created

1. Creating private and public subnets

* From left side menu of open vpc page, click subnets
* Click button create subnet
* On open form select our created vpc
* Give the name of your subnet
* Select Availability Zone
* Fill IPv4 CIDR block (mypri-10.200.1.0\24,mypub-10.200.0.0\24)
* Click create subnet (mypri-private subnet,mypub-public subnet)

1. Creating route table for public and private subnet

* From left side menu of open vpc page, click route tables
* Click button create route table
* Fill name tag in open form
* Select your VPC
* Click create
* For public subnet route table (mypubrt), add route of inter gateway in this route table
* For private subnet route table (myprirt)
* Now select corresponding subnet under subnet associations tab

1. Creating Internet gateway

* From left side menu of open vpc page, click route tables
* Click button create internet gateway
* Attach by selecting vpc to which gateway is created

1. Create Nat gateway

* From left side menu of open vpc page, click route tables
* Click button create nat gateway in public subnet

1. Create two ec2 instance of Ubuntu 18.4

1. Go to AWS Management Console -> Services -> EC2

2. Click on Launch Instance

3. You will be asked to select Amazon Machine Image(AMI)

4. Select Amazon ubunto 16.4 ami

5. Select instance type t2.micro

6. Click next configure instance details

7. Keep everything as default

8. Click next add storage default general purpose

9. Click next add tags

10. In tags click on add tag

11. Add the tags given below • Key • Value

12. Click next Configure security group

13. Click on create a new security group • • Security Group name • Description

15. Now if you see you have 1 rules SSH with source as myip and in other 2 rules ssh with ip range 10.200.0.0\24 and icmp with range 10.200.0.0\24

16. Click on review and launch

17. In the key pair dialog box select Create a new key pair, download the new key pair, select the acknowledgement box, then click launch instances.

18. Now you will get the instance ready scroll down click on view instances

**CONFIGURE INSTANCES FOR INSTALL MYSQL AND MATTEROST**

1. Login to public instance of Ubuntu
2. Save pair key at public Ubuntu instance
3. Give permission Chmod 400 “mypk.pem”
4. Login to private Ubuntu instance using command  ssh -i "priv.pem" [ubuntu@10.200.1.186](mailto:ubuntu@10.200.1.186)
5. After login run follong commands to install mysql

sudo apt update

sudo apt install mysql-server

1. Run following commands
2. Log in to MySQL as root.

sudo mysql

1. Create the Mattermost user ‘mmuser’.

create user 'mmuser'@'%' identified by 'sql123';

1. Create the Mattermost database.

create database mattermost;

1. Grant access privileges to the user ‘mmuser’.

grant all privileges on mattermost.\* to 'mmuser'@'%';

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

1. Log out of MySQL.

exit

**Installing Mattermost**

1. Log in to the server that will host Mattermost Server and open a terminal window.
2. Download mattermost.

wget https://releases.mattermost.com/X.X.X/mattermost-X.X.X-linux-amd64.tar.gz

1. Extract the Mattermost Server files.

tar -xvzf mattermost\*.gz

1. Move the extracted file to the /opt directory.

sudo mv mattermost /opt

1. Create the storage directory for files.

sudo mkdir /opt/mattermost/data

1. Set up a system user and group called mattermost that will run this service, and set the ownership and permissions.
2. Create the Mattermost user and group:

sudo useradd --system --user-group mattermost

1. Set the user and group *mattermost* as the owner of the Mattermost files:

sudo chown -R mattermost:mattermost /opt/mattermost

1. Give write permissions to the *mattermost* group:

sudo chmod -R g+w /opt/mattermost

1. Set up the database driver in the file /opt/mattermost/config/config.json
2. Set "DriverName" to "mysql"
3. Set "DataSource" to the following value, replacing <mmuser-password> and <host-name-or-IP> with the appropriate values. Also make sure that the database name is mattermost instead of mattermost\_test:

"mmuser:<mmuser-password>@tcp(<host-name-or-IP>:3306)/mattermost?charset=utf8mb4,utf8&readTimeout=30s&writeTimeout=30s"

1. Also set "SiteURL" to the full base URL of the site (e.g. "https://mattermost.example.com").
2. Test the Mattermost server to make sure everything works.
   1. Change to the mattermost directory:

cd /opt/mattermost

* 1. Start the Mattermost server as the user mattermost:

sudo -u mattermost ./bin/mattermost

When the server starts, it shows some log information and the text Server is listening on :8065. You can stop the server by pressing CTRL+C in the terminal window.

1. Setup Mattermost to use *systemd* for starting and stopping.
2. Create a *systemd* unit file:

sudo touch /lib/systemd/system/mattermost.service

1. Make systemd load the new unit.

sudo systemctl daemon-reload

1. Check to make sure that the unit was loaded.

sudo systemctl status mattermost.service

1. Start the service.

sudo systemctl start mattermost.service

1. Verify that Mattermost is running.

curl http://localhost:8065

You should see the HTML that’s returned by the Mattermost server.