

Declaration	
Questions in this exercise are intentionally complex and could be convoluted or confusing. This is by design and to simulate real life situations where customers seldom give crystal clear requirements and ask unambiguous questions.	
I have read the above statement and agree to these conditions	
I AGREE	JULIEI SONA <Enter your name above this line to indicate that you are in agreement>

Instructions
Every screenshot requested in this workbook is compulsory and carries 0.5 points.
Your AWS account ID must be clearly visible in every screenshot using the AWS console; missing id or using someone else's id is not permitted. Such cases will be considered as plagiarism and severe penalty will be imposed.
All screenshots must be in the order mentioned under "Expected Screenshots" for every step
DO NOT WAIT UNTIL THE LAST MINUTE. The program office will not extend the project submission deadline under any circumstances.
The file should be renamed in the format BATCH_FIRSTNAME_LASTNAME_PROJECT1. For example: PGPCCMAY18_VIJAY_DWIVEDI_PROJECT1.pdf

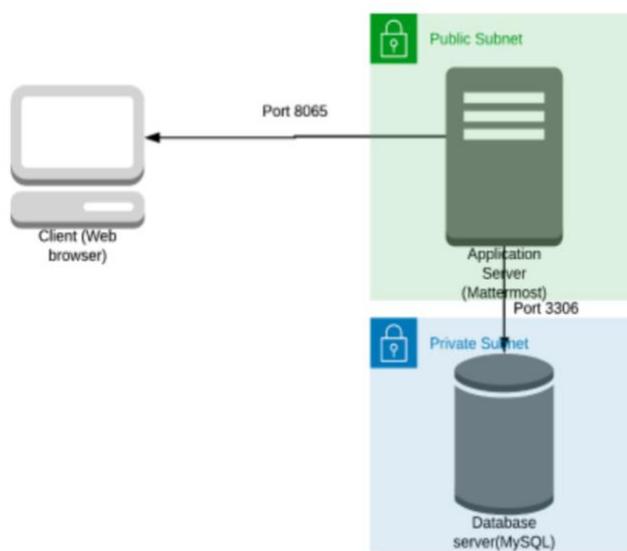
Resource Clean Up
Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.
After completing the lab, make sure to delete each resource created in reverse chronological order.

Scenario

Team communication and instant messaging solutions are an integral part of any business environment today. As of 2020, the total number of users of Slack and Microsoft Teams exceeded 20 million.

Some organizations might have compliance policies in place which do not allow them to use services managed by third parties. They will prefer solutions that can be managed and hosted on servers controlled by them. The same will extend to communication solutions as well.

Architecture diagram



Architecture Implementation	
1	Implement 2 different subnets (one public and the other private) in a custom VPC
2	Install and configure MySQL on an Amazon Linux 2 instance on the private subnet using the instructions provided. (Hint: Use a bastion host and a NAT gateway)
3	Install and configure Mattermost on an Amazon Linux 2 instance on the public subnet using the provided instructions.
4	Configure the security groups to allow the ports as shown in the architecture.
5	Test the installation by accessing the IP of the public instance in a browser via the port 8065.

Step 1: VPC and Subnet Creation

Step number	a
Step name	Creation of VPC
Instructions	<p>1) Navigate to VPC using the Services button at the top of the screen</p> <p>2) Select "Your VPCs" on the left side of the screen</p> <p>3) Click on "Create VPC"</p> <p>4) Enter the following fields :</p> <p>Name: Project 1 VPC</p> <p>IPv4 CIDR Block : 10.0.0.0/16</p> <p>The rest of the options can be ignored</p> <p>5) Select "Create VPC"</p> <p>6) Select the VPC and click on Actions->Edit DNS hostnames</p> <p>7) Enable DNS hostnames and click on Save</p>
Expected screenshots	Created VPC with properties visible

<Insert Screenshot a(1) here>

The screenshot shows the AWS VPC dashboard. On the left sidebar, under 'Your VPCs', there is a list of resources including Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The main content area displays 'Your VPCs (1/1) Info'. It shows a table with one row for 'Project 1 VPC'. The table columns include Name (Project 1 VPC), VPC ID (vpc-02f5761d5c2e05807), State (Available), IPv4 CIDR (10.0.0.0/16), and IPv6 CIDR (None). Below the table, the specific details for 'vpc-02f5761d5c2e05807 / Project 1 VPC' are shown. The 'Details' tab is selected, displaying information such as VPC ID, State, Tenancy (Default), Default VPC (No), Network Address Usage metrics (Disabled), and various network settings like Main route table, IPv6 pool, and Owner ID.

Step number	B
Step name	Creation of public subnet
Instructions	<p>1) Navigate to VPC->Subnets</p> <p>2) Click on "Create Subnet"</p> <p>3) Enter the following fields</p> <p>Name tag : Public Subnet</p> <p>VPC : Select the Project 1 VPC</p> <p>IPv4 CIDR block : 10.0.1.0/24</p> <p>The other options can be ignored</p> <p>4) Click on Create</p> <p>5) Once the subnet has been created, select the subnet and click on Actions->Modify Auto-assign IP settings</p> <p>6) Enable the option "Auto assign IPv4" and select Save</p>
Expected screenshots	Subnet Creation screen

<Insert Screenshot b(1) here>

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

- Subnets (1/1) Info:** A success message says "YOU HAVE SUCCESSFULLY CHANGED SUBNET SETTINGS." with a note "Enable auto-assign public IPv4 address".
- Subnet Details:**

Name	Subnet ID	State	VPC	IPv4 CIDR
Public Subnet	subnet-0ea0d916989b78fad	Available	vpc-02f5761d5c2e05807 Proj...	10.0.1.0/24
- Configuration Table:**

subnet-0ea0d916989b78fad	arn:aws:ec2:us-east-1:47112771580:subnet/subnet-0ea0d916989b78fad	Available	10.0.1.0/24
Available IPv4 addresses	Oea0d916989b78fad	Availability Zone	Availability Zone ID
251	us-east-1	Route table	use1-az3
Network border group	-	Network ACL	-
us-east-1	VPC	Auto-assign IPv6 address	Auto-assign customer-owned IPv4 address
Default subnet	vpc-02f5761d5c2e05807 Project 1 VPC	No	No
No	Customer-owned IPv4 pool	IPv4 CIDR reservations	IPv6 CIDR reservations
-	Auto-assign public IPv4 address	-	-
Yes	IPv4 CIDR	Resource name DNS A record	Resource name DNS AAAA record
IPv6-only	Outpost ID	Disabled	Disabled
No	-		
DNS564	Hostname type		
Disabled	IP name		
Owner	47112771580		

Step number	C
Step name	Creation of private subnet
Instructions	<p>1) Navigate to VPC->Subnets</p> <p>2) Click on "Create Subnet"</p> <p>3) Enter the following fields</p> <p>Name tag : Private Subnet</p> <p>VPC : Select the Project 1 VPC</p> <p>IPv4 CIDR block : 10.0.2.0/24</p> <p>The other options can be ignored</p> <p>4) Click on Create</p>
Expected screenshots	Subnet Creation screen

<Insert Screenshot c(1) here>

The screenshot shows the AWS VPC Subnets console. In the top navigation bar, there are three tabs: 'Your VPCs | VPC Management' (highlighted), 'subnets | VPC Console', and another 'subnets | VPC Console'. The URL in the address bar is 'https://aws.amazon.com/vpcconsole/home?region=us-east-1#subnets:'.

The main content area displays a table titled 'Subnets (1/2) Info'. The table has columns for Name, Subnet ID, State, VPC, and IPv4 CIDR. It lists two subnets: 'Public Subnet' (subnet-0ea0d916989b78fad) and 'Private Subnet' (subnet-010bde8c6fb953326). The 'Private Subnet' row is selected, indicated by a blue background and a checked checkbox.

Below the table, a message box says 'You have successfully created 1 subnet: subnet-010bde8c6fb953326'. The subnet details are shown in a modal window:

- Details** tab is active.
- Subnet ID:** subnet-010bde8c6fb953326
- Subnet ARN:** arn:aws:ec2:us-east-1:47111271580:subnet/subnet-010bde8c6fb953326
- State:** Available
- IPv4 CIDR:** 10.0.2.0/24
- Availability Zone:** us-east-1e
- Availability Zone ID:** use1-az3
- Route table:** rtb-01d487231929c3f4b
- Network ACL:** acl-0d781aae8d8010713
- Auto-assign IPv6 address:** No
- Auto-assign customer-owned IPv4 address:** No
- IPv4 CIDR reservations:** -
- IPv6 CIDR reservations:** -
- Resource name DNS A record:** -

Step 2 : Internet Gateway and VPC

Step number	A
Step name	Creation and Configuration of Internet Gateway
Instructions	<ol style="list-style-type: none"> 1) Navigate to VPCs->Internet Gateway 2) Click on "Create Internet Gateway" 3) Enter the name tag "Project 1 Internet Gateway" and click on "Create Internet Gateway" 4) After the gateway is created, select it and click on Actions->Attach to VPC 5) Select the Project 1 VPC and click on "Attach Internet Gateway"
Expected screenshots	Creation of Internet Gateway

<Insert Screenshot a(1) here >

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A single internet gateway, 'Project 1 Internet Gateway' (ID: igw-01316670eb4106660), is listed and marked as 'Attached' to the 'vpc-02f5761d5c2e05807 | Project 1 VPC'. The gateway's details are shown in a modal window, including its ID, state (Attached), VPC ID, and owner.

Name	Internet gateway ID	State	VPC ID	Owner
Project 1 Internet Gateway	igw-01316670eb4106660	Attached	vpc-02f5761d5c2e05807 Project 1 VPC	471112771580

Step number	B
Step name	Creation of public route table
Instructions	<p>1) Navigate to VPC -> Route Tables and click on Create Route table</p> <p>2) Enter the name tag "Public Route Table", select the Project 1 VPC from the dropdown and click on Create</p> <p>3) Once the route table is created, select it and select the Routes tab below the list of route tables</p> <p>4) Click in Edit Routes and add the following route (Don't edit the existing one)</p> <ul style="list-style-type: none"> - Destination : 0.0.0.0/0 - Target : Select Internet Gateway and the select the Project 1 Internet Gateway <p>Click on Save Routes</p> <p>5) Select the Subnet Associations tab and click on Edit Subnet Associations</p> <p>6) Select the Public Subnet from the list and click on Save</p>
Expected screenshots	<p>1) Route list of the route table</p> <p>2) Subnet Associations of the route table</p>

<Insert Screenshot b(1) here>

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A success message at the top right states: "You have successfully updated subnet associations for rtb-03c9994a1452a2cce / Public Route Table." The main content area displays the details for the route table 'rtb-03c9994a1452a2cce / Public Route Table'. The 'Routes' tab is active, showing two routes:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-01316670eb4106660	Active	No
10.0.0.0/16	local	Active	No

<Insert Screenshot b(2) here>

You have successfully updated subnet associations for rtb-03c9994a1452a2cce / Public Route Table.

rtb-03c9994a1452a2cce / Public Route Table

Details		Info																	
Route table ID	rtb-03c9994a1452a2cce	Main	No																
VPC	vpc-02f5761d5c2e05807 Project 1	Owner ID	471112771580																
Explicit subnet associations (1) Edit subnet associations <table border="1"> <thead> <tr> <th>Name</th> <th>Subnet ID</th> <th>IPv4 CIDR</th> <th>IPv6 CIDR</th> </tr> </thead> <tbody> <tr> <td>Public Subnet</td> <td>subnet-0ea0d916989b7...</td> <td>10.0.1.0/24</td> <td>-</td> </tr> </tbody> </table> Subnets without explicit associations (1) Edit subnet associations <table border="1"> <thead> <tr> <th>Name</th> <th>Subnet ID</th> <th>IPv4 CIDR</th> <th>IPv6 CIDR</th> </tr> </thead> <tbody> <tr> <td>Private Subnet</td> <td>subnet-010bde8c6fb95...</td> <td>10.0.2.0/24</td> <td>-</td> </tr> </tbody> </table>				Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Public Subnet	subnet-0ea0d916989b7...	10.0.1.0/24	-	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Private Subnet	subnet-010bde8c6fb95...	10.0.2.0/24	-
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR																
Public Subnet	subnet-0ea0d916989b7...	10.0.1.0/24	-																
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR																
Private Subnet	subnet-010bde8c6fb95...	10.0.2.0/24	-																

Routes Subnet associations Edge associations Route propagation Tags

Step number	C
Step name	Creation of NAT gateway
Instructions	<ol style="list-style-type: none"> 1) Navigate to VPC using the Services button at the top of the screen 2) Select NAT Gateway at the left side of the screen 3) Click on Create NAT Gateway - Deploy it in the public subnet - Connectivity type: Public - Allocate an elastic IP by clicking on “Allocate Elastic IP” 4) Click on “Create NAT Gateway” to create the gateway
Expected screenshots	<ol style="list-style-type: none"> 1) NAT gateway creation details 2) Gateway after creation

<Insert Screenshot c(1) here>

Your VPCs | VPC Manager | subnets | VPC Console | subnets | VPC Console | igws | VPC Console | RouteTableDetails | VPC C | CreateNatGateway | VPC C | + | us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateNatGateway: N. Virginia | Julieisona | Services | Search | [Option+S] | Elastic IP address 34.206.94.116 (eipalloc-08764298f67118920) allocated.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
 The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

Connectivity type
Select a connectivity type for the NAT gateway.
 Public
 Private

Elastic IP allocation ID [Info](#)
Assign an Elastic IP address to the NAT gateway.

[► Additional settings \[Info\]\(#\)](#)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="NAT Gateway"/> <input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>	

You can add 49 more tags.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

<Insert Screenshot c(2) here>

Your VPCs | VPC Manager | subnets | VPC Console | subnets | VPC Console | igws | VPC Console | RouteTableDetails | VPC C | NatGateways | VPC Console | + | us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#NatGateways: N. Virginia | Julieisona | Services | Search | [Option+S] | Actions | Create NAT gateway | nat-07f1cbfb7a88f6002 / NAT Gateway | Find resources by attribute or tag | < 1 > | [Actions](#) | [Create NAT gateway](#)

VPC dashboard

- EC2 Global View
- Filter by VPC
- 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
- Security
 - Network ACLs
 - Security groups
- DNS firewall

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

NAT gateways (1/1) Info					
Find resources by attribute or tag Actions Create NAT gateway					
Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...
<input checked="" type="radio"/> NAT Gateway	nat-07f1cbfb7a88f6002	Public	Available	-	34.206.94.116 10.0.1

nat-07f1cbfb7a88f6002 / NAT Gateway

Details			
NAT gateway ID	Connectivity type	State	State message
nat-07f1cbfb7a88f6002	Public	Available	-
NAT gateway ARN	Primary public IPv4 address	Primary private IPv4 address	Primary network interface ID
arn:aws:ec2:us-east-1:471112771580:natgateway/nat-07f1cbfb7a88f6002	34.206.94.116	10.0.1.103	eni-0e091911cb4b4f6cd
VPC	Subnet	Created	Deleted
vpc-02f5761d5c2e05807 / Project 1	subnet-0ea0d916989b78fad / Public Subnet	Monday, July 29, 2024 at 16:11:11 CDT	-

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step number	D
Step name	Creation of private route tables
Instructions	<p>1) Navigate to VPC -> Route Tables and click on Create Route table</p> <p>2) Enter the name tag "Private Route Table", select the Project 1 VPC from the dropdown and click on Create</p> <p>3) Once the route table is created, select it and select the Routes tab below the list of route tables</p> <p>4) Click in Edit Routes and add the following route (Don't edit the existing one)</p> <ul style="list-style-type: none"> - Destination : 0.0.0.0/0 - Target: Select NAT Gateway and select the NAT Gateway created in the previous step <p>Click on Save Routes</p> <p>5) Select the Subnet Associations tab and click on Edit Subnet Associations</p> <p>6) Select the private Subnet from the list and click on Save</p>
Expected screenshots	<p>1) Route list of the route table</p> <p>2) Subnet association of the route table</p>

<Insert Screenshot for d(1) here >

The screenshot shows the AWS VPC Route Tables console. On the left, there's a navigation sidebar with options like EC2 Global View, Filter by VPC, Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), and DNS firewall. The main area displays a table of Route tables (1/3) with columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. A 'Private Route Table' is selected, showing its details. The 'Routes' tab is active, displaying two routes: one to '0.0.0.0/0' targeting 'nat-07f1cbfb7a88f6002' with an 'Active' status, and another to '10.0.0.0/16' targeting 'local' with an 'Active' status.

<Insert Screenshot for d(2) here>

The screenshot shows the AWS VPC Route Tables console. On the left, there's a navigation sidebar with sections like EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables), Security, and DNS firewall. The main area displays a table of route tables with columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. One route table is selected: "Private Route Table" (rtb-08ffd2652c369ef85). Below it, the details for this specific route table are shown, including its subnet associations (one entry: Private Subnet, subnet-010bde8c6fb953..., 10.0.2.0/24) and a note that no other subnets are explicitly associated.

Step 3 : Creation of database and application servers

Step number	A
Step name	Creation of application server
Instructions	<ol style="list-style-type: none"> 1) Navigate to EC2 using the Services button at the top of the screen 2) Select Instances at the left side of the screen 3) Click on Launch Instance <ul style="list-style-type: none"> - Select the AMI Amazon 2 Linux - Select the instance type t2.micro - Select Network as "Project 1 VPC" and subnet as "Public Subnet" - For the security group, open the ports 80,443, 22 and 8065 for source set to "Anywhere" 4) Launch the instance after creating a new pem file and downloading it
Expected screenshots	<ol style="list-style-type: none"> 1) AMI used 2) Instance configuration screen 3) Security group rules 4) Instance after creation

<Insert screenshot a(1) here>

The screenshot shows the AWS CloudFormation console with a stack named "Application Server". The "Template" tab is selected, displaying the CloudFormation template. The template includes resources such as an AWS Lambda function, an API Gateway endpoint, and an Amazon RDS database. The "Outputs" section shows the ARN of the Lambda function and the ARN of the RDS endpoint.

<Insert screenshot a(2) here>

The screenshot shows the AWS EC2 Instances page for the instance "Application Server". The instance is listed as "Running" with a Public IPv4 address of 35.168.3.68. The instance type is t2.micro, and it is associated with a VPC ID of vpc-0f5761d5c2e05807. The instance has a Subnet ID of subnet-0ea0d916989b78fad and an Instance ARN of arn:aws:ec2:us-east-1:471112771580:instance/i-087396aaa77819b9e.

<Insert screenshot a(3) here>

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pull
Application Se...	i-087396aaa77819b9e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1e	ec2

i-087396aaa77819b9e (Application Server)

Mon Jul 29 2024 16:29:31 GMT-0500 (Central Daylight Time)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-0f93bcef2a2d1e326	8065	TCP	0.0.0.0/0	vpc1-sg
-	sgr-0b7da178d46004fc4	443	TCP	0.0.0.0/0	vpc1-sg
-	sgr-08ac750011fe2420f	80	TCP	0.0.0.0/0	vpc1-sg
-	sgr-003767ff2d4df05b	22	TCP	0.0.0.0/0	vpc1-sg

Outbound rules

<Insert screenshot a(4) here>

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pull
Application Se...	i-087396aaa77819b9e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1e	ec2

i-087396aaa77819b9e (Application Server)

Networking

Public IPv4 address	Private IPv4 addresses	VPC ID
35.168.3.68 open address	10.0.1.111	vpc-02f5761d5c2e05807 (Project 1 VPC)
Public IPv4 DNS	Private IP DNS name (IPv4 only)	
ec2-35-168-3-68.compute-1.amazonaws.com open address	ip-10-0-1-111.ec2.internal	

Networking details

Subnet ID	IPV6 addresses	Secondary private IPv4 addresses
subnet-0ea0d916989b78fad (Public Subnet)	-	-
Availability zone	Carrier IP addresses (ephemeral)	Outpost ID
us-east-1e	-	-

Network Interfaces (1) Info

Use RBN as guest OS hostname	Answer RBN DNS hostname IPv4
Disabled	Disabled

Step number B

Step name Creation of database server

Instructions

- 1) Navigate to EC2 using the Services button at the top of the screen
- 2) Select Instances at the left side of the screen
- 3) Click on Launch Instance
- Select the AMI Amazon 2 Linux
- Select the instance type t2.micro
- Select Network as "Project 1 VPC" and subnet as "Private Subnet"
- For the security group, open the ports 80, 443,22 and 3306 for source set to "Anywhere"
- 4) Launch the instance by selecting the same pem file created in the previous step

Expected screenshots

- 1) AMI used
- 2) Instance configuration screen
- 3) Security group rules
- 4) Instance after creation

<Insert screenshot b(1) here>

The screenshot shows the AWS Lambda function configuration page. The function name is 'HelloWorld'. The 'Handler' dropdown is set to 'lambda_function.lambda_handler'. The 'Runtime' dropdown is set to 'Python 3.9'. The 'Memory' dropdown is set to '128 MB'. The 'Timeout' dropdown is set to '3 seconds'. The 'Environment' section shows the environment variables: 'AWS_LAMBDA_FUNCTION_NAME' (value: 'HelloWorld'), 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE' (value: '128'), 'AWS_LAMBDA_FUNCTION_TIMEOUT' (value: '3'), and 'AWS_LAMBDA_HANDLER' (value: 'lambda_function.lambda_handler'). The 'Role' dropdown is set to 'HelloWorld - Lambda execution role'. The 'Code' section shows the code in the 'File' tab, which is a simple Python script. The 'Logs' section shows the log stream 'HelloWorld.log'. The 'Test' section shows a successful test result with the message 'Hello World!'. The 'Version' dropdown is set to '1'. The 'Publish' button is highlighted in orange.

<Insert screenshot b(2) here>

Screenshot b(3) showing the AWS EC2 Instances page. The left sidebar shows navigation for VPCs, Global View, Events, and Instances. Under Instances, 'Instances' is selected. The main pane displays two instances: 'Application Server' (running, t2.micro) and 'Database server' (pending, t2.micro). A detailed view of the 'Database server' instance is open, showing its ID (i-0466e6b7bb1eb5236), creation time (Mon Jul 29 2024 16:39:26 GMT-0500), security group (sg-0e2dee0a09ff0a051 (vpc2-sg)), and inbound rules (port 443 TCP, port 22 TCP, port 80 TCP, port 3306 TCP).

<Insert screenshot b(3) here>

Screenshot b(4) showing the AWS EC2 Instance Details page for the 'Database server' instance (i-0466e6b7bb1eb5236). The left sidebar is identical to screenshot b(3). The main pane displays the 'Instance summary' for the database server. Key details include:

- Instance ID:** i-0466e6b7bb1eb5236 (Database server)
- Public IPv4 address:** 52.3.220.143 | [open address](#)
- Private IPv4 address:** 10.0.2.109
- Public IPv4 DNS:** ec2-52-3-220-143.compute-1.amazonaws.com | [open address](#)
- InstanceState:** Running
- Hostname type:** IP name: ip-10-0-2-109.ec2.internal
- Private IP DNS name (IPv4 only):** ip-10-0-2-109.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-02f5761d5c2e05807 (Project 1 VPC)
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)
- Auto Scaling Group name:** -
- Subnet ID:** subnet-010bde8c6fb953326 (Private Subnet)
- Instance ARN:** arn:aws:ec2:us-east-1:471112771580:instance/i-0466e6b7bb1eb5236

The bottom navigation bar includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

<Insert screenshot b(4) here>

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various EC2-related options like Dashboard, Global View, Events, and Instances. The Instances section is expanded, showing sub-options like Instances, Instance Types, Launch Templates, etc. The main pane displays a table of instances. Two instances are listed: 'Application Server' (i-087396aaa77819b9e) and 'Database server' (i-0466e6b7bb1eb5236). Both instances are in the 'Running' state and are t2.micro type. The 'Database server' instance is selected. A detailed view for this instance is shown on the right, including its instance ID, public and private IP addresses, instance state, and VPC ID.

Step 4: Application and Database Installation and Testing

Step number	A
Step name	Installation and configuration of MySQL
Instructions	<p>1) Copy the database pem file into the application server using the below command</p> <pre>scp -i YOUR_APP.pem YOUR_DB.pem ec2-user@YOUR_APP_PUBLIC_IP:/home/ec2-user</pre> <p>2) Log into the application server using SSH/Putty</p> <p>3) From the application server, log into the database server using the pem file copied in step 1 and the private IP address of the database server with the following command</p> <pre>ssh -i YOUR_DB.pem ec2-user@YOUR_DB_PRIVATE_IP</pre> <p>4) Enter the following commands to install and configure MySQL on the database server</p> <pre>sudo yum update wget http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm sudo yum localinstall mysql57-community-release-el7-9.noarch.rpm -y sudo yum install mysql-community-server -y --nogpgcheck</pre>

```
sudo systemctl start mysqld.service
```

Run the below command to retrieve a temporary password for MySQL

```
TEMP_PWD=$(sudo grep 'temporary password' /var/log/mysqld.log | awk '{printf "%s", $11}')
```

Log in to MySQL with the below command

```
mysql -u root --password=$TEMP_PWD
```

Enter the below command after you login to MySQL. Do not change the password set in the below command.

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'Password42!';
```

Type 'exit' into the MySQL prompt and press Enter to exit out of the MySQL environment. Enter the below commands to complete the setup. Ignore any warning messages you receive.

```
wget https://d6opu47qoi4ee.cloudfront.net/install_mysql_linux.sh
chmod 777 install_mysql_linux.sh
sudo ./install_mysql_linux.sh
```

5) Type `exit` to exit the database server and go back to the application server

- Expected screenshots
- 1) Installation of MySQL
 - 2) Retrieving the temporary password
 - 3) Executing the provided script

<Insert screenshot a(1) here>

```
aws Your VPCs Subnets RouteTables NatGateways EC2 us-east-1 Instances EC2 Instance + N. Virginia Julieisona

← → G ↻ us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-087396aaa77819b9e&osUser=ec2-user&sshPort=22#/

Resolving Dependencies
--> Running transaction check
--> Package mysql57-community-release.noarch 0:e17-9 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

-----  

  Package           Arch      Version            Repository          Size  

-----  

Installing:    mysql57-community-release        noarch    e17-9            /mysql57-community-release-e17-9.noarch     8.6 k  

Transaction Summary
-----  

Install 1 Package

Total size: 8.6 k
Installed size: 8.6 k
Downloading packages:
Running transaction check
Running transaction test
Pre-trans: 0 succeeded
Running transaction
  Installing : mysql57-community-release-e17-9.noarch [#####
  Installing : mysql57-community-release-e17-9.noarch [#####
  Installing : mysql57-community-release-e17-9.noarch [#####
  Verifying   : mysql57-community-release-e17-9.noarch [#####
  Installed: mysql57-community-release.noarch 0:e17-9  

Complete!: i-087396aaa77819b9e (Application Server)
Public IPs: 35.168.3.68 Private IPs: 10.0.1.111
```

<Insert screenshot a(2) here>

```
Your VPCs | V < subnets | VPC < igws | VPC < RouteTables | X NatGateways | X EC2 | us-east | Instances | EC2 Instance X + us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-087396aaa77819b9e&osUser=ec2-user&sshPort=22# Services Search [Option+S] N. Virginia Julieisona Erasing : l:mariadb-libs-5.5.68-1.amzn2.0.1.x86_64 7/7 Verifying : mysql-community-libs-5.7.44-1.el7.x86_64 1/7 Verifying : mysql-community-libs-compat-5.7.44-1.el7.x86_64 2/7 Verifying : mysql-community-client-5.7.44-1.el7.x86_64 2/7 Verifying : ncurses-compat-libs-6.0-8.20170212.amzn2.1.8.x86_64 3/7 Verifying : mysql-community-common-5.7.44-1.el7.x86_64 4/7 Verifying : mysql-community-server-5.7.44-1.el7.x86_64 5/7 Verifying : mariadb-libs-5.5.68-1.amzn2.0.1.x86_64 6/7 Verifying : mariadb-libs-5.5.68-1.amzn2.0.1.x86_64 7/7 Installed: mysql-community-libs.x86_64 0:5.7.44-1.el7      mysql-community-libs-compat.x86_64 0:5.7.44-1.el7      mysql-community-server.x86_64 0:5.7.44-1.el7 Dependency Installed: mysql-community-client.x86_64 0:5.7.44-1.el7      mysql-community-common.x86_64 0:5.7.44-1.el7      ncurses-compat-libs.x86_64 0:6.0-8.20170212.amzn2.1.8 Replaced: mariadb-libs.x86_64 1:5.5.68-1.amzn2.0.1 Complete! [root@ip-10-0-1-111 ec2-user]# sudo systemctl start mysqld.service [root@ip-10-0-1-111 ec2-user]# TEMP_PWD=$(sudo grep 'temporary password' /var/log/mysqld.log | awk '{printf "%s", $11}') [root@ip-10-0-1-111 ec2-user]# mysql -u root --password=$TEMP_PWD mysql: [Warning] Using a password on the command line interface can be insecure. Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 2 Server version: 5.7.44 Copyright (c) 2000, 2023, Oracle and/or its affiliates. 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> i-087396aaa77819b9e (Application Server) PublicIPs: 35.168.3.68 PrivateIPs: 10.0.1.111
```

```

Verifying : mysql-community-client-5.7.44-1.el7.x86_64
Verifying : ncurses-compat-libs-6.0-8.20170212.amzn2.1.8.x86_64
Verifying : mysql-community-common-5.7.44-1.el7.x86_64
Verifying : mysql-community-server-5.7.44-1.el7.x86_64
Verifying : mariadb-libs-5.5.68-1.amzn2.0.1.x86_64

Installed:
  mysql-community-libs.x86_64 0:5.7.44-1.el7          mysql-community-libs-compat.x86_64 0:5.7.44-1.el7      mysql-community-server.x86_64 0:5.7.44-1.el7
Dependency Installed:
  mysql-community-client.x86_64 0:5.7.44-1.el7        mysql-community-common.x86_64 0:5.7.44-1.el7      ncurses-compat-libs.x86_64 0:6.0-8.20170212.amzn2.1.8
Replaced:
  mariadb-libs.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
[root@ip-10-0-1-111 ec2-user]# sudo systemctl start mysqld.service
[root@ip-10-0-1-111 ec2-user]# TEMP_PWD=$(sudo grep 'temporary password' /var/log/mysqld.log | awk '{printf "%s", $11}')
[root@ip-10-0-1-111 ec2-user]# mysql -u root --password=$TEMP_PWD
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.44

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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> ALTER USER 'root'@\localhost' IDENTIFIED BY 'Password42!';
Query OK, 0 rows affected (0.00 sec)

mysql> 
```

i-087396aaa77819b9e (Application Server)
PublicIPs: 35.168.3.68 PrivateIPs: 10.0.1.111

<Insert screenshot a(3) here>

```

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.44

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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> ALTER USER 'root'@\localhost' IDENTIFIED BY 'Password42!';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
[root@ip-10-0-1-111 ec2-user]# wget https://d6opu47goi4ee.cloudfront.net/install_mysql_linux.sh
--2024-07-29 22:22:57-- https://d6opu47goi4ee.cloudfront.net/install_mysql_linux.sh
Resolving d6opu47goi4ee.cloudfront.net (d6opu47goi4ee.cloudfront.net)... 3.162.115.128, 3.162.115.161, 3.162.115.180, ...
Connecting to d6opu47goi4ee.cloudfront.net (d6opu47goi4ee.cloudfront.net)|3.162.115.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 189 [text/x-sh]
Saving to: 'install_mysql_linux.sh'

0% [=====] 189          0K/s    in 0s          100%[=====]

2024-07-29 22:22:58 (24.8 MB/s) - 'install_mysql_linux.sh' saved [189/189]

[root@ip-10-0-1-111 ec2-user]# chmod 777 install_mysql_linux.sh
[root@ip-10-0-1-111 ec2-user]# sudo ./install_mysql_linux.sh
mysql: [Warning] Using a password on the command line interface can be insecure.
[root@ip-10-0-1-111 ec2-user]# exit
exit
[root@ip-10-0-1-111 ec2-user]# 
```

i-087396aaa77819b9e (Application Server)
PublicIPs: 35.168.3.68 PrivateIPs: 10.0.1.111

Step number	B
Step name	Installation and configuration of Mattermost
Instructions	<p>1) Enter the following commands after logging into the application server via SSH to install and configure Mattermost.</p> <pre>wget https://d6opu47qoi4ee.cloudfront.net/install_mattermost_linux.sh sudo yum install dos2unix -y sudo dos2unix install_mattermost_linux.sh chmod 700 install_mattermost_linux.sh sudo ./install_mattermost_linux.sh YOUR_DB_PRIVATE_IP sudo chown -R mattermost:mattermost /opt/mattermost sudo chmod -R g+w /opt/mattermost cd /opt/mattermost sudo -u mattermost ./bin/mattermost</pre> <p>2) Check whether the server has been successfully deployed by navigating to the following URL in your web browser. The web page might take a couple of minutes to load.</p> <p>http://YOUR_APP_PUBLIC_IP:8065</p>
Expected screenshots	<ol style="list-style-type: none">1) Executing the script2) Starting the Mattermost server3) Accessing the application via web browser

<Insert screenshot b(1) here>

<Insert screenshot b(2) here>

```
Your VPCs | V subnets | VPC subnets | VPC igws | VPC RouteTables | NatGateways | EC2 | us-east-1 Instances | EC2 Instance +  
aws Services Search [Option+S] N. Virginia Julieisona  
mattermost/fonts/  
mattermost/fonts/OFL.txt  
mattermost/fonts/nunito-bold.ttf  
mattermost/i18n  
mattermost/i18n/ko.json  
mattermost/i18n/zh-CN.json  
mattermost/i18n/en.json  
mattermost/i18n/de.json  
mattermost/i18n/fr.json  
mattermost/i18n/pl.json  
mattermost/i18n/ru.json  
mattermost/i18n/ta.json  
mattermost/i18n/zh-TW.json  
mattermost/i18n/it.json  
mattermost/i18n/ja.json  
mattermost/i18n/nl.json  
mattermost/i18n/pt-BR.json  
mattermost/i18n/ro.json  
mattermost/i18n/es.json  
mattermost/logs  
mattermost/prepackaged_plugins/  
mattermost/prepackaged_plugins/mattermost-plugin-antivirus-v0.1.1.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-npm-v1.0.3.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-autolink-v1.1.1.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-aws-SNS-v1.0.2.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-aws-SQS-v1.0.2.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-welcomebot-v1.1.1.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-jenkins-v1.0.0.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-zipkin-v2.2.1.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-github-v0.1.1.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-custom-attributes-v1.0.2.tar.gz  
mattermost/prepackaged_plugins/mattermost-plugin-zoom-v1.1.2.tar.gz  
Extracted mattermost  
    user  
[root@ip-10-0-1-111 ec2-user]# i-087396aaa77819b9e (Application Server)  
PublicIPs: 55.168.5.68 PrivateIPs: 10.0.1.111
```

<Insert screenshot b(3) here>

Your VPCs | V subnets | VPC subnets | VPC igws | VPC RouteTables | V NatGateways | EC2 us-east-1 Instances | EC2 Instance +

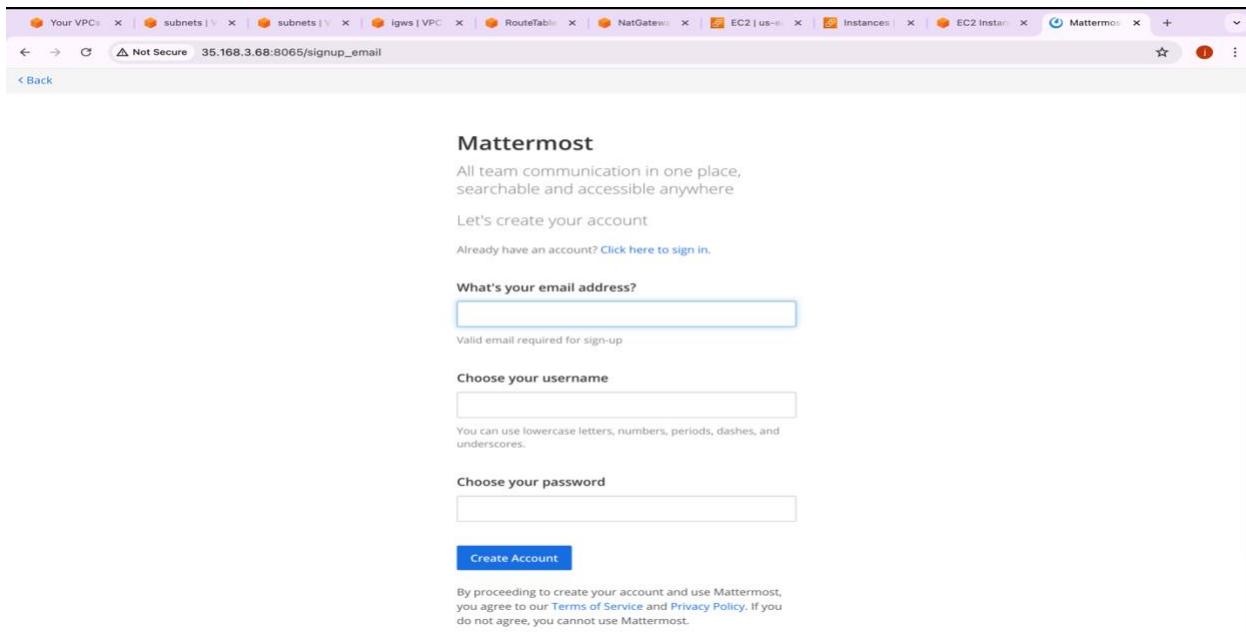
← → G 22 us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-087396aa77819b9e&osUser=ec2-user&sshPort=22# N. Virginia Juliesona

aws Services Search [Option+S]

```
5535}) {"level": "info", "ts": 1722292287.640277, "caller": "app/migrations.go:102", "msg": "Migrating emojis config to database."} {"level": "info", "ts": 1722292287.873299, "caller": "mlog/log.go:166", "msg": "Starting up plugins"} {"level": "info", "ts": 1722292287.873441, "caller": "app/plugin.go:213", "msg": "Syncing plugins from the file store"} {"level": "info", "ts": 1722292290.7565773, "caller": "mlog/sugar.go:19", "msg": "Ensuring Surveybot exists", "plugin_id": "com.mattermost.nps"} {"level": "info", "ts": 1722292290.799402, "caller": "mlog/sugar.go:19", "msg": "Surveybot created", "plugin_id": "com.mattermost.nps"} {"level": "info", "ts": 1722292290.8157032, "caller": "mlog/sugar.go:19", "msg": "Upgrade detected. Checking if a survey should be scheduled.", "plugin_id": "com.mattermost.nps"} {"level": "info", "ts": 1722292290.9102018, "caller": "mlog/sugar.go:19", "msg": "Scheduling next survey for Aug 19, 2024", "plugin_id": "com.mattermost.nps"} {"level": "info", "ts": 1722292291.43515, "caller": "app/server.go:217", "msg": "Current version is 5.19.0 (5.19.0/Thu Jan 16 18:30:33 UTC 2020/90cf883f84000d6fdb025308ad14d56e6ed5f05/1268390c0cdce16f750b0b6fe2534b82586d595f)" } {"level": "info", "ts": 1722292291.4353182, "caller": "app/server.go:218", "msg": "Enterprise Enabled: true"} {"level": "info", "ts": 1722292291.4353707, "caller": "app/server.go:221", "msg": "Printing current working", "directory": "/opt/mattermost"} {"level": "info", "ts": 1722292291.4354467, "caller": "app/server.go:222", "msg": "Loaded config", "source": "file:///opt/mattermost/config/config.json"} {"level": "error", "ts": 1722292291.4545849, "caller": "mlog/log.go:174", "msg": "RPC call OnConfigurationChange to plugin failed.", "plugin_id": "com.mattermost.nps"} {"level": "error", "ts": 1722292291.5126047, "caller": "mlog/log.go:174", "msg": "RPC call OnConfigurationChange to plugin failed.", "plugin_id": "com.mattermost.nps"} {"level": "error", "ts": 1722292291.534654, "caller": "jobs/workers.go:68", "msg": "Starting workers"} {"level": "info", "ts": 1722292291.5409856, "caller": "jobs/schedulers.go:74", "msg": "Starting schedulers."} {"level": "info", "ts": 1722292291.5547864, "caller": "app/web_hub.go:75", "msg": "Starting websocket hubs", "number_of_hubs": 2} {"level": "info", "ts": 1722292291.6287475, "caller": "app/server.go:440", "msg": "Starting Server..."} {"level": "info", "ts": 1722292291.629105, "caller": "app/server.go:506", "msg": "Server is listening on [::]:8065"} {"level": "error", "ts": 1722292321.4421978, "caller": "plugin/health_check.go:90", "msg": "Health check failed for plugin", "id": "com.mattermost.nps", "error": "Plugin RC connection is not responding"} {"level": "warn", "ts": 1722292321.4423158, "caller": "plugin/hclog_adapter.go:51", "msg": "error closing client during Kill", "plugin_id": "com.mattermost.nps", "wrapped_extras": "errconnection is shut down"} {"level": "warn", "ts": 1722292321.4423475, "caller": "plugin/hclog_adapter.go:53", "msg": "plugin failed to exit gracefully", "plugin_id": "com.mattermost.nps"} {"level": "info", "ts": 1722292321.4600015, "caller": "mlog/sugar.go:19", "msg": "Ensuring Surveybot exists", "plugin_id": "com.mattermost.nps"} {"level": "info", "ts": 1722292352.882818, "caller": "migrations/worker.go:109", "msg": "Worker: Job is complete", "worker": "Migrations", "job_id": "7ui6z41ad3gubxn87jtn6ohy5a"}
```

i-087396aa77819b9e (Application Server)

Public IPs: 35.168.3.68 Private IPs: 10.0.1.111



Step 5: Answer the following questions

Q1 What is the default setting for DNS hostnames when a new VPC is created?

- a) Enabled
- b) Disabled
- c) Can be set during VPC creation
- d) Depends on the region used

Enter your answer here

b. Disabled

Q2 What is the term used for the machine when we use it to log into the database server?

- a) Bastion Host
- b) NAT Gateway
- c) Tunnel Interface
- d) SSH Gateway

Enter your answer here

a. Bastion Host

Q3 The database server security group in this exercise has to keep port 3306 open. Which protocol uses this port to communicate?

- a) HTTPS
- b) RDP
- c) TCP
- d) SCP

Enter your answer here

c. TCP

Q4 Which port is being used by Mattermost to communicate with the client application

- a) 8080
- b) 80
- c) 443
- d) 8065

Enter your answer here

d. 8065

Q5 Which of the following is a reason why we cannot set the CIDR block for the public subnet to 10.0.2.0/16, assuming the values for the other CIDR blocks are the same as mentioned in the instructions?

- a) CIDR block overlaps with existing block
- b) CIDR block is not a valid CIDR
- c) CIDR block does not fall within the VPC
- d) There is no reason, this is a perfectly valid CIDR

Enter your answer here

a.CIDR block overlaps with existing block

Q6 Assume that you have been asked to create 3 EC2 instances - application server, the database server and NAT instance. Each of these instances have their own security groups with a set of ports to be kept open. One of these ports is entirely unnecessary for the given architecture to function. Which of the following could it be?

- a) Port 22 on the NAT instances
- b) Port 3306 on the database server
- c) Port 3306 on the application server
- d) Port 22 on the application server

Enter your answer here

c. port 3306 on application server

Q7 How are we going to increase the security of the Mattermost server to ensure the users are from a specific organization and the traffic is originating from a known IP address?

- 1.By enabling IP Whitelisting allowing specific IP addresses to access Mattermost Server
2. By using OAuth Authentication to authenticate users and ensuring they belong to specific organisations
- 3.by Configuring Strict firewall rules
- 4.by Enabling 2factor Authentication
- 5.By implementing Role based access controls
- 6.by Regularly updating and Patching
- 7.by Monitoring and logging server logs and network traffic to detect suspicious activities
- 8.by Setting up Reverse proxy
- 9.By Enabling SSL/TLS Encryption to secure traffic between clients and mattermost server
- 10.Isolating the mattermost server in separate network server

Q8 How do we achieve elasticity for the Mattermost server?

- 1.By using Horizontal scaling
- 2.by using Load Balancer to distribute traffic across instances
- 3.By using Auto Scaling groups to add or remove instances on demand
- 4.by Using Containerization
5. By using Monitoring and Alerting tools
- 6.Ensuring database(MySQL, PostgreSQL) can scale independently
- 7.By Implementing caching mechanisms
- 8.Using a Queue based architecture
9. by regularly performing performance testing to handle traffic better.
10. By using managed cloud services such as AWS Elastic beanstalk.

Grade distribution

MCQs	6 (1 point each)
Subjective questions	110 (5 points each)
Implementation screenshots	24 points (1 point each)
Total	40 points