

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	MOHAMED EL HASSNAOUI <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

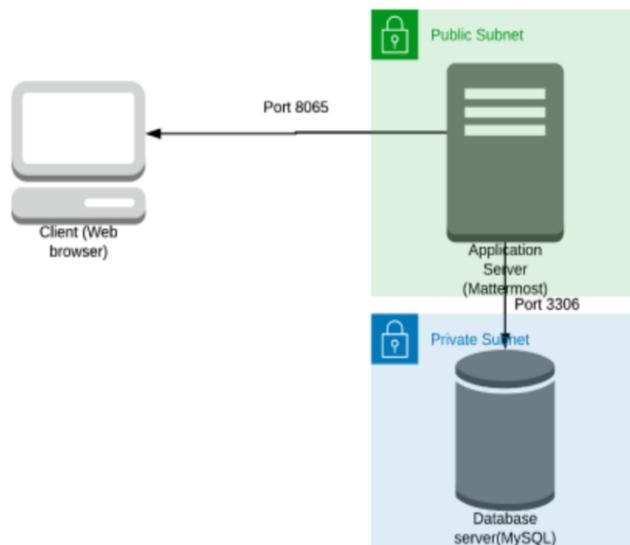
Resource Clean Up
Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.
After completing the lab, make 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 'Virtual private cloud', 'Your VPCs' is selected. In the main content area, the 'Your VPCs (1/2)' section displays a table with one row for 'Project 1 VPC'. The table columns include Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. The 'Project 1 VPC' row has a green checkmark next to it, indicating it is available. Below the table, the details for 'vpc-0dab8c1b934626a83 / Project 1 VPC' are shown. The 'Details' tab is selected, displaying information such as VPC ID, State (Available), Default DHCP option set, Main route table, and Main network ACL.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	ypc-001bb692ac3335f0b	Available	172.31.0.0/16	-
Project 1 VPC	ypc-0dab8c1b934626a83	Available	10.0.0.0/16	-

vpc-0dab8c1b934626a83 / Project 1 VPC				
Details	Resource map	CIDRs	Flow logs	Tags
Details				
VPC ID vpc-0dab8c1b934626a83	State Available	DNS hostnames Enabled	DNS resolution Enabled	
Tenancy Default	DHCP option set dopt-02bd42b2e73d2d203	Main route table rtb-0e10732e130840f76	Main network ACL acl-0aaa62507759a9edc	
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -	
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups Failed to load rule groups	Owner ID 610792625205		

>

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

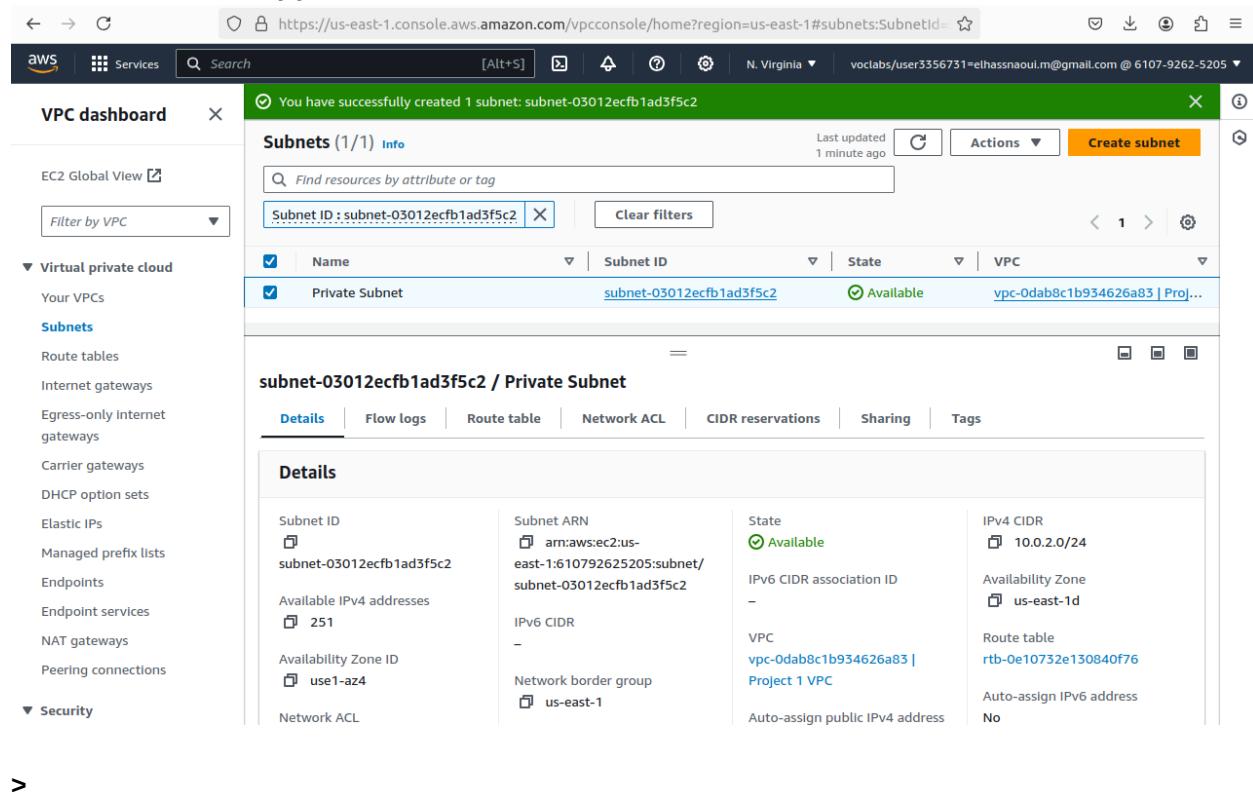
The screenshot shows the AWS VPC Subnets page. In the top navigation bar, the URL is https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#subnets. The left sidebar is titled 'VPC dashboard' and includes sections for EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, Peering connections), and Security. The 'Subnets' section is currently selected. The main content area displays a table titled 'Subnets (1/7)'. The table has columns for Name, Subnet ID, State, and VPC. There are two rows: one for a subnet named '-' with Subnet ID subnet-0ebc872c6a38baccf, State Available, and VPC vpc-001bb692ac3335f0b; and another for a subnet named 'Public Subnet' with Subnet ID subnet-02830fefef003ceea2, State Available, and VPC vpc-0dab8c1b934626a83. Below the table, a modal window is open for the subnet 'subnet-02830fefef003ceea2 / Public Subnet'. The 'Details' tab is selected, showing the following configuration:

Details	Value	Details	Value
Subnet ID	subnet-02830fefef003ceea2	Subnet ARN	arn:aws:ec2:us-east-1:61079265205:subnet/subnet-02830fefef003ceea2
Available IPv4 addresses	251	State	Available
Availability Zone ID	use1-az4	IPv6 CIDR	-
Network ACL	-	VPC	vpc-0dab8c1b934626a83 Project 1 VPC
Auto-assign customer-owned	No	Auto-assign public IPv4 address	Yes
		IPv4 CIDR reservations	

>

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 page. A green success message at the top states: "You have successfully created 1 subnet: subnet-03012ecfb1ad3f5c2". The main table displays one subnet entry:

Name	Subnet ID	State	VPC
Private Subnet	subnet-03012ecfb1ad3f5c2	Available	vpc-0dab8c1b934626a83 Pro...

Below the table, the details for the subnet are shown:

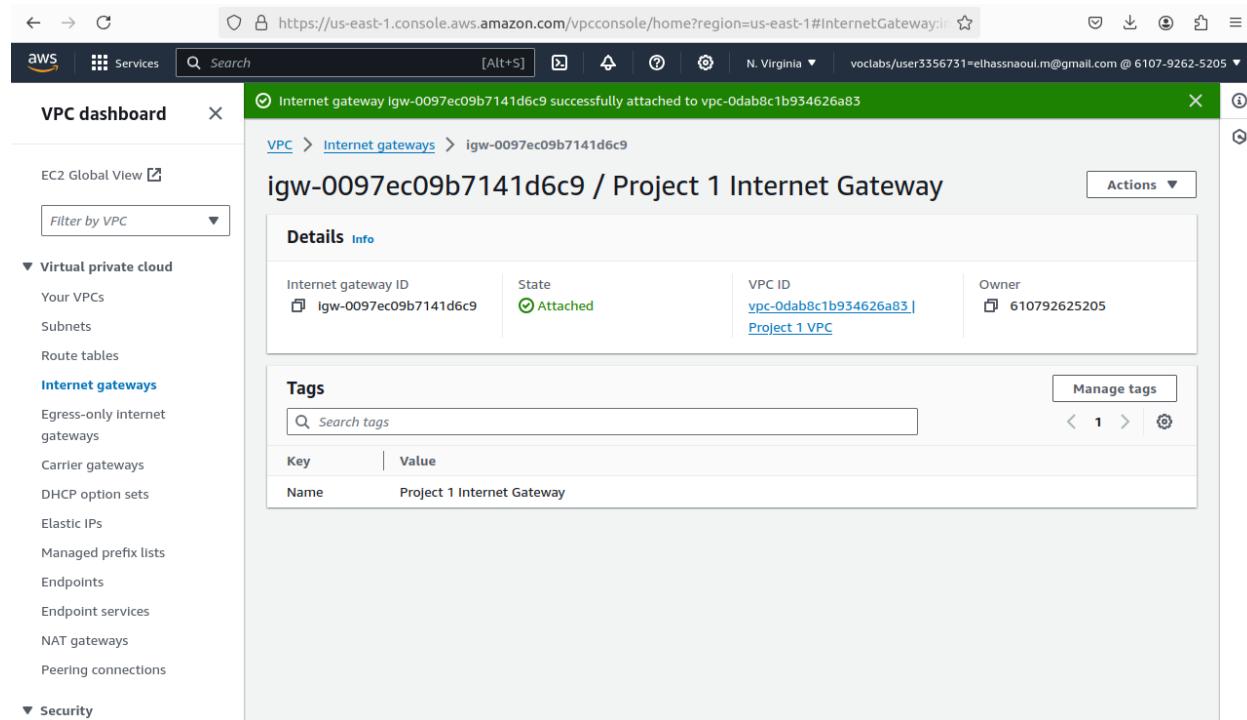
Details	
Subnet ID	subnet-03012ecfb1ad3f5c2
Subnet ARN	arn:aws:ec2:us-east-1:610792625205:subnet/subnet-03012ecfb1ad3f5c2
State	Available
IPv4 CIDR	10.0.2.0/24
Available IPv4 addresses	251
IPv6 CIDR	-
Availability Zone ID	use1-az4
VPC	vpc-0dab8c1b934626a83 Project 1 VPC
Network border group	us-east-1
Network ACL	
Auto-assign public IPv4 address	No

>

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. On the left, there's a sidebar with options like EC2 Global View, Filter by VPC, Virtual private cloud (Your VPCs, Subnets, Route tables), Internet gateways (selected), 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 shows a success message: "Internet gateway igw-0097ec09b7141d6c9 successfully attached to vpc-0dab8c1b934626a83". Below this, the "igw-0097ec09b7141d6c9 / Project 1 Internet Gateway" details are displayed. The "Details" tab is selected, showing the Internet gateway ID (igw-0097ec09b7141d6c9), State (Attached), VPC ID (vpc-0dab8c1b934626a83), and Owner (610792625205). A "Tags" section shows a single tag: Name (Project 1 Internet Gateway).

>

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	1) Route list of the route table
screenshots	2) Subnet Associations of the route table

<Insert Screenshot b(1) here - Route list of the route table

The screenshot shows the AWS VPC Route Table Details page for the route table `rtb-0f1a66b9894362b5b / Public Route Table`. A success message at the top states: "You have successfully updated subnet associations for rtb-0f1a66b9894362b5b / Public Route Table." The page displays the following details:

Route table ID	Main	Explicit subnet associations	Edge associations
<code>rtb-0f1a66b9894362b5b</code>	<input type="checkbox"/> No	<code>subnet-02830fefef003ceea2 / Public Subnet</code>	—
VPC	Owner ID		
<code>vpc-0dab8c1b934626a83</code>	<code>610792625205</code>		
Project 1 VPC			

The **Routes** tab is selected, showing two routes:

Destination	Target	Status	Propagated
<code>0.0.0.0/0</code>	igw-0097ec09b7141d6c9	Active	No
<code>10.0.0.0/16</code>	local	Active	No

>

<Insert Screenshot b(2) here - Subnet Associations of the route table

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A table lists two route tables: 'rtb-0e10732e130840f76' (Public Route Table) and 'rtb-0f1a66b9894362b5b'. The 'Public Route Table' has one explicit subnet association to 'subnet-02830fefef003cee...' with an IPv4 CIDR of '10.0.1.0/24'. It also lists 'Subnets without explicit associations' (Private Subnet) with an IPv4 CIDR of '10.0.2.0/24'.

>

Step number	c
Step name	Creation of NAT gateway
Instructions	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	1) NAT gateway creation details 2) Gateway after creation

<Insert Screenshot c(1) here - NAT gateway creation details

<https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateNatGateway>

Elastic IP address 44.211.11.68 (elalloc-0bec898a5b8a56815) allocated.

VPC > NAT gateways > Create NAT gateway

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the Internet.

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

>

<Insert Screenshot c(2) here - Gateway after creation

<https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#NatGatewayDetail>

NAT gateway nat-0f6eb0cbd8ccda978 | Project_1_Gateway was created successfully.

VPC dashboard

VPC

Details			
NAT gateway ID <input type="text" value="nat-0f6eb0cbd8ccda978"/>	Connectivity type <input type="text" value="Public"/>	State <input type="text" value="Pending"/>	State message <small>Info</small> <input type="text" value="-"/>
NAT gateway ARN <input type="text" value="arn:aws:ec2:us-east-1:610792625205:natgateway/nat-0f6eb0cbd8ccda978"/>	Primary public IPv4 address <input type="text" value="-"/>	Primary private IPv4 address <input type="text" value="-"/>	Primary network interface ID <input type="text" value="-"/>
VPC <input type="text" value="vpc-0dab8c1b934626a83 / Project_1 VPC"/>	Subnet <input type="text" value="subnet-02830fefef003ceea2 / Public Subnet"/>	Created <input type="text" value="Sunday, August 11, 2024 at 01:34:43 EDT"/>	Deleted <input type="text" value="-"/>

Secondary IPv4 addresses

Private IPv4 address	Network interface ID	Status	Failure
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>

>

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

<Insert Screenshot for d(1) here - Route list of the route table

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A new route table named 'Private Route Table' has been created and is highlighted. The 'Routes' tab is selected, showing two routes: one to a NAT gateway and another to a local subnet.

Name	Route table ID	Explicit subnet associations
-	rtb-0686fd0dfebcd6c74	-
-	rtb-0e10732e130840f76	-
Public Route Table	rtb-0f1a66b9894362b5b	subnet-02830fef003ceeaa2 / Public Subnet
Private Route Table	rtb-088912707505ec652	subnet-03012ecfb1ad3f5c2 / Private Subnet

rtb-088912707505ec652 / Private Route Table																			
Details	Routes	Subnet associations	Edge associations																
	Routes (2) <table border="1"> <thead> <tr> <th colspan="4">Routes (2)</th> </tr> <tr> <th>Destination</th> <th>Target</th> <th>Status</th> <th>Propagated</th> </tr> </thead> <tbody> <tr> <td>0.0.0.0/0</td> <td>nat-0f6eb0cbd8ccda978</td> <td>Active</td> <td>No</td> </tr> <tr> <td>10.0.0.0/16</td> <td>local</td> <td>Active</td> <td>No</td> </tr> </tbody> </table>	Routes (2)				Destination	Target	Status	Propagated	0.0.0.0/0	nat-0f6eb0cbd8ccda978	Active	No	10.0.0.0/16	local	Active	No		
Routes (2)																			
Destination	Target	Status	Propagated																
0.0.0.0/0	nat-0f6eb0cbd8ccda978	Active	No																
10.0.0.0/16	local	Active	No																

>

<Insert Screenshot for d(2) here - Subnet association of the route table

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. It displays two route tables: 'Public Route Table' and 'Private Route Table'. The 'Private Route Table' is selected, showing its details. Under the 'Subnet associations' tab, it lists one explicit association to a 'Private Subnet' with CIDR 10.0.2.0/24. There are no subnets without explicit associations.

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 - AMI used

<

<Insert screenshot a(2) here - Instance configuration screen

The screenshot shows the AWS EC2 'Launch Instances' configuration screen. On the left, there's a search bar and a navigation menu. The main area displays an AMI selection card for 'amzn2-ami-kernel-5.10-hvm-2.0.20240719.0-x86_64-gp2'. It includes fields for Name, Description, Image ID, Catalog, Published date, Architecture, Virtualization, Root device type, and ENA Enabled status. Below this is an 'Instance type' section with a dropdown menu showing 't2.micro' selected. To the right is a 'Summary' panel containing instance details like Software Image (AMI), Virtual server type (t2.micro), Firewall (New security group), and Storage (1 volume(s) - 8 GiB). A tooltip for the 'Free tier' is visible.

>

<Insert screenshot a(2) here - Instance configuration screen

This screenshot shows the same configuration screen as the previous one, but with different settings. The AMI selected is 'ami-03972092c42e8c0ca'. The instance type dropdown now shows 't2.micro' with 'All generations' selected. The 'Key pair (login)' section has 'Project1_AppServer_KeyPair' entered. The 'Summary' panel remains the same, displaying the same instance details and a tooltip for the 'Free tier'.

>

<Insert screenshot a(3) here - Security group rules

The screenshot shows the AWS EC2 console for a security group named "launch-wizard-1". The "Inbound Security Group Rules" section contains three rules:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**: Type: ssh, Protocol: TCP, Port range: 22, Source type: Anywhere, Description: e.g. SSH for admin desktop.
- Security group rule 2 (TCP, 80, 0.0.0.0/0)**: Type: HTTP, Protocol: TCP, Port range: 80, Source type: Anywhere, Description: e.g. SSH for admin desktop.
- Security group rule 3 (TCP, 443, 0.0.0.0/0)**: Type: HTTPS, Protocol: TCP, Port range: 443, Source type: Anywhere, Description: e.g. SSH for admin desktop.

The "Summary" panel on the right shows the following details:

- Number of Instances: 1
- Software Image (AMI): Amazon Linux 2 AMI (HVM) - Ker...read more
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A callout box highlights the "Free tier" information: "In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1

The screenshot shows the AWS EC2 console for the same security group "launch-wizard-1". The "Inbound Security Group Rules" section now contains four rules:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**: Type: ssh, Protocol: TCP, Port range: 22, Source type: Anywhere, Description: e.g. SSH for admin desktop.
- Security group rule 2 (TCP, 80, 0.0.0.0/0)**: Type: HTTP, Protocol: TCP, Port range: 80, Source type: Anywhere, Description: e.g. SSH for admin desktop.
- Security group rule 3 (TCP, 443, 0.0.0.0/0)**: Type: HTTPS, Protocol: TCP, Port range: 443, Source type: Anywhere, Description: e.g. SSH for admin desktop.
- Security group rule 4 (TCP, 8065, 0.0.0.0/0)**: Type: Custom TCP, Protocol: TCP, Port range: 8065, Source type: Anywhere, Description: e.g. SSH for admin desktop.

The "Summary" panel on the right shows the following details:

- Number of Instances: 1
- Software Image (AMI): Amazon Linux 2 AMI (HVM) - Ker...read more
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A callout box highlights the "Free tier" information: "In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1

>

<Insert screenshot a(4) here - Instance after creation

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code (Preview), Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area has a title 'Instances (1) Info' with a search bar and filters for Name, Instance ID, Instance state, Instance type, and Status check. A table lists one instance: 'EC2_Project_1_App_Server' with Instance ID 'i-0be7cd5fd53dec1ba', status 'Running', type 't2.micro', and a note 'Initializing'. Below the table is a modal window titled 'Select an instance'.

>

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 - AMI used

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Name: amzn2-ami-kernel-5.10-hvm-2.0.20240719.0-x86_64-gp2 Verified provider

Description: Amazon Linux 2 Kernel 5.10 AMI 2.0.20240719.0 x86_64 HVM gp2

Image ID: ami-03972092c42e8c0ca

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
Community AMIs	2024-07-20T04:09:44.000Z	x86_64	hvm	ebs	Yes

Summary

Number of Instances: 1

Software Image (AMI): amzn2-ami-kernel-5.10-hvm-2.0....read more
ami-03972092c42e8c0ca

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year Includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4

>

<Insert screenshot b(2) here - Instance configuration screen

Instance type: t2.micro Free tier eligible

Key pair (login): Project1_AppServer_KeyPair

Summary

Number of instances: 1

Software Image (AMI): amzn2-ami-kernel-5.10-hvm-2.0....read more
ami-03972092c42e8c0ca

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year Includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4

>

<Insert screenshot b(3) here - Security group rules>

The screenshot shows the AWS EC2 console under the 'Inbound Security Group Rules' section. It displays three security group rules:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**: Type: ssh, Protocol: TCP, Port range: 22. Source type: Anywhere, Source: 0.0.0.0/0.
- Security group rule 2 (TCP, 80, 0.0.0.0/0)**: Type: HTTP, Protocol: TCP, Port range: 80. Source type: Anywhere, Source: 0.0.0.0/0.
- Security group rule 3 (TCP, 443, 0.0.0.0/0)**: Type: HTTPS, Protocol: TCP, Port range: 443. Source type: Anywhere, Source: 0.0.0.0/0.

The right panel shows the summary for the security group, including the number of instances (1), software image (amzn2-ami-kernel-5.10-hvm-2.0...), virtual server type (t2.micro), and storage (1 volume(s) - 8 GiB). A tooltip for the free tier is visible.

The screenshot shows the AWS EC2 console under the 'Inbound Security Group Rules' section. It displays four security group rules:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**: Type: Custom TCP, Protocol: TCP, Port range: 3306. Source type: Anywhere, Source: 0.0.0.0/0.
- Security group rule 2 (TCP, 80, 0.0.0.0/0)**: Type: Custom TCP, Protocol: TCP, Port range: 3306. Source type: Anywhere, Source: 0.0.0.0/0.
- Security group rule 3 (TCP, 443, 0.0.0.0/0)**: Type: HTTPS, Protocol: TCP, Port range: 443. Source type: Anywhere, Source: 0.0.0.0/0.
- Security group rule 4 (TCP, 3306, 0.0.0.0/0)**: Type: Custom TCP, Protocol: TCP, Port range: 3306. Source type: Anywhere, Source: 0.0.0.0/0.

The right panel shows the summary for the security group, including the number of instances (1), software image (amzn2-ami-kernel-5.10-hvm-2.0...), virtual server type (t2.micro), and storage (1 volume(s) - 8 GiB). A tooltip for the free tier is visible.

<Insert screenshot b(4) here -Instance after creation

The screenshot shows the AWS EC2 Instances page. There are two instances listed:

- EC2_Project_1_App_Server**: Instance ID i-0be7cd5fd53dec1ba, State Running, Type t2.micro, Status 2/2 checks passed.
- EC2_Project_1_DB**: Instance ID i-0ecc3edbdcabf3af7, State Running, Type t2.micro, Status 2/2 checks passed.

The details for the EC2_Project_1_DB instance are expanded:

Attribute	Value
Instance ID	i-0ecc3edbdcabf3af7 (EC2_Project_1_DB)
Public IPv4 address	44.206.240.225 open address
Private IPv4 addresses	172.31.91.124
Public IPv4 DNS	ec2-44-206-240-225.compute-1.amazonaws.com open address
Instance state	Running
Hostname type	IP name: ip-172-31-91-124.ec2.internal
Private IP DNS name (IPv4 only)	ip-172-31-91-124.ec2.internal

>

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> <pre>scp -i Project1_AppServer_KeyPair.pem Project1_AppServer_KeyPair.pem ec2-user@ec2-54-xxx-xx-xx.compute-1.amazonaws.com:/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>

- 4) Enter the following commands to install and configure MySQL on the database server

```
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
```

```
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
CrqVQ*=m9+-
 - 3) Executing the provided script

<Insert screenshot a(1) here - Installation of MySQL

```
Activities Aug 11 21:51
ec2-user@ip-10-0-2-126:~
```

```
~~.~. / Amazon Linux 2023, GA and supported until 2028-03-15.
~/m/' https://aws.amazon.com/linux/amazon-linux-2023/
```

```
[ec2-user@ip-10-0-2-126 ~]$ sudo yum localinstall mysql57-community-release-el7-9.noarch.rpm -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Examining mysql57-community-release-el7-9.noarch.rpm: mysql57-community-release-el7-9.noarch
Marking mysql57-community-release-el7-9.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
-->> Package mysql57-community-release.noarch 0:el7-9 will be installed
-->> Finished Dependency Resolution

Dependencies Resolved
```

```
=====
Package           Arch   Version          Repository          Size
=====
mysql57-community-release      noarch  el7-9  /mysql57-community-release-el7-9.noarch  8.6 k
```

```
Transaction Summary
=====
Install 1 Package
```

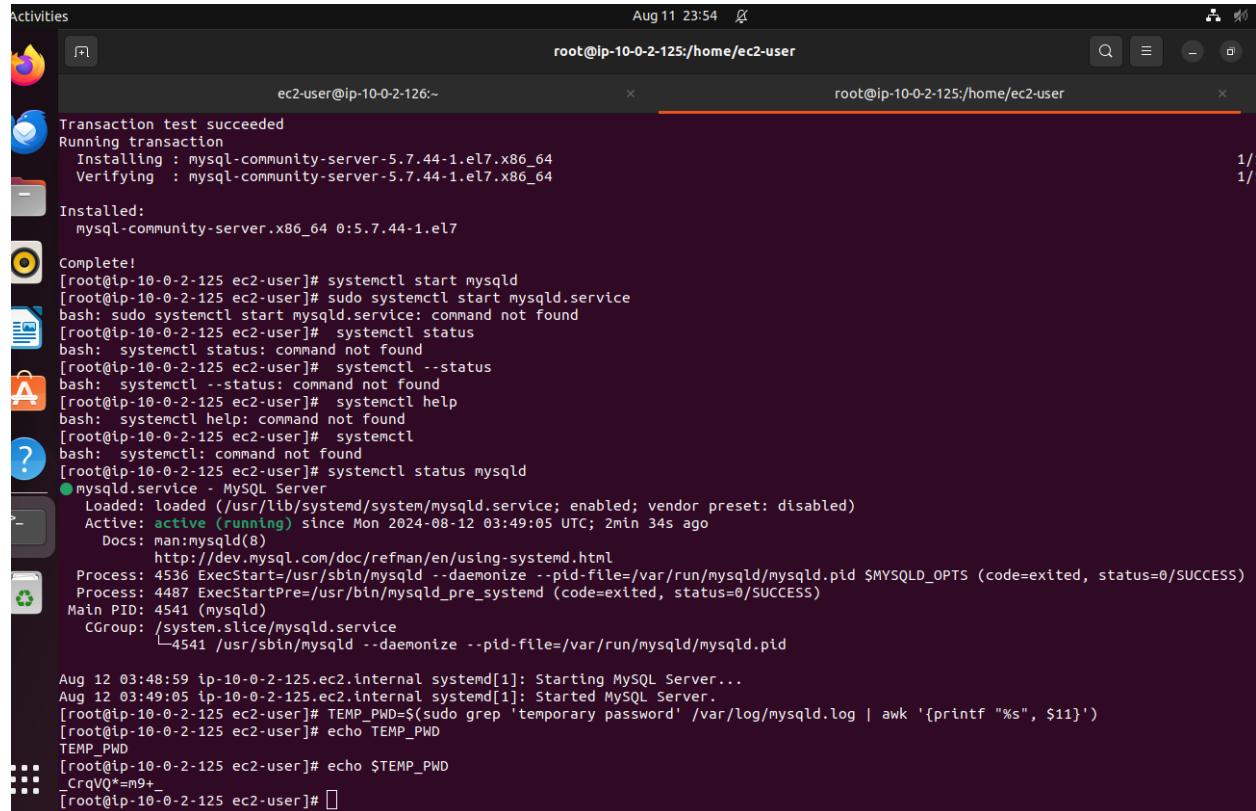
```
Total size: 8.6 k
Installed size: 8.6 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mysql57-community-release-el7-9.noarch          1/1
  Verifying   : mysql57-community-release-el7-9.noarch          1/1

Installed:
  mysql57-community-release.noarch 0:el7-9
```

```
Complete!
[ec2-user@ip-10-0-2-126 ~]$
```

>

<Insert screenshot a(2) here - Retrieving the temporary password



The screenshot shows a terminal window with two tabs. The left tab is titled 'ec2-user@ip-10-0-2-126:-' and the right tab is titled 'root@ip-10-0-2-125:/home/ec2-user'. The terminal output is as follows:

```
Transaction test succeeded
Running transaction
  Installing : mysql-community-server-5.7.44-1.el7.x86_64
  Verifying   : mysql-community-server-5.7.44-1.el7.x86_64

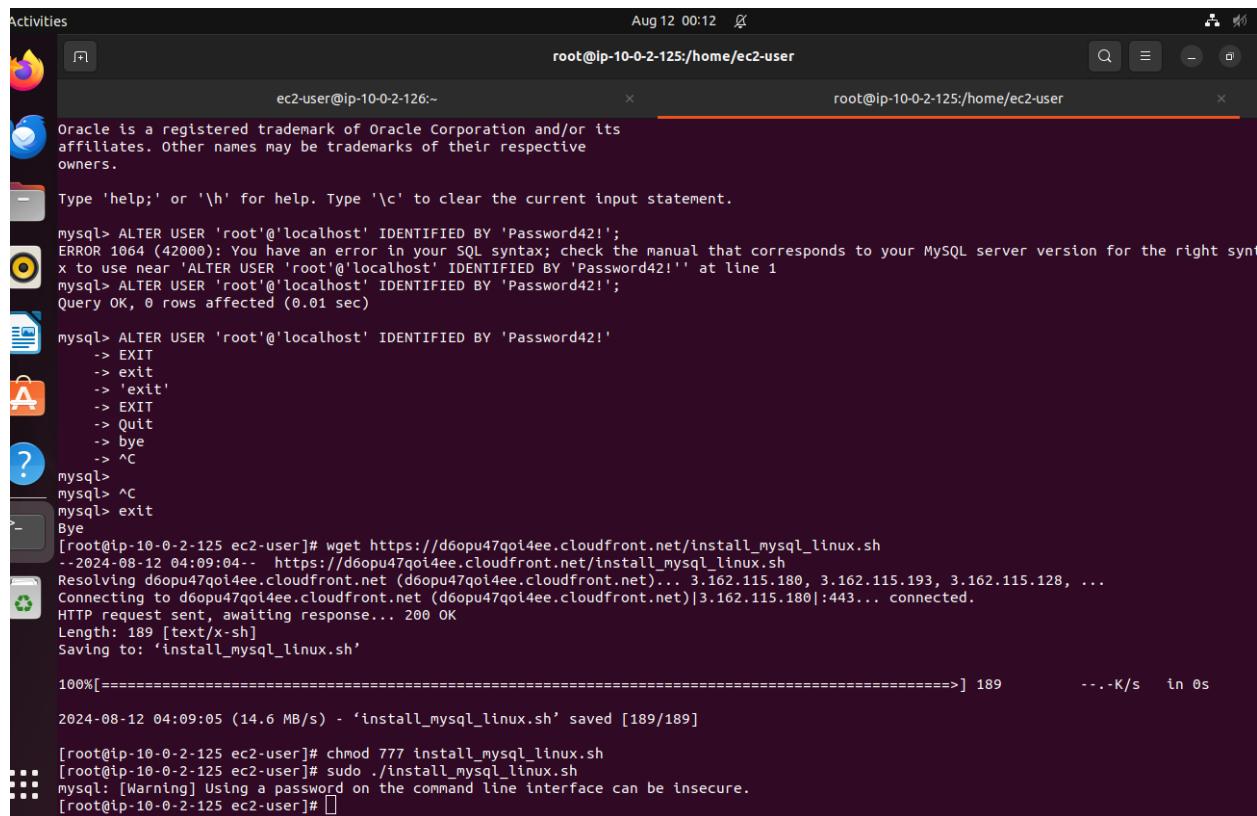
Installed:
  mysql-community-server.x86_64 0:5.7.44-1.el7

Complete!
[root@ip-10-0-2-125 ec2-user]# systemctl start mysqld
[root@ip-10-0-2-125 ec2-user]# sudo systemctl start mysqld.service
bash: sudo systemctl start mysqld.service: command not found
[root@ip-10-0-2-125 ec2-user]# systemctl status
bash: systemctl status: command not found
[root@ip-10-0-2-125 ec2-user]# systemctl --status
bash: systemctl --status: command not found
[root@ip-10-0-2-125 ec2-user]# systemctl help
bash: systemctl help: command not found
[root@ip-10-0-2-125 ec2-user]# systemctl
bash: systemctl: command not found
[root@ip-10-0-2-125 ec2-user]# systemctl status mysqld
● mysqld.service - MySQL Server
  Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
  Active: active (running) since Mon 2024-08-12 03:49:05 UTC; 2min 34s ago
    Docs: man:mysqld(8)
          http://dev.mysql.com/doc/refman/en/using-systemd.html
  Process: 4536 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysql/mysqld.pid $MYSQLD_OPTS (code=exited, status=0/SUCCESS)
  Process: 4487 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
 Main PID: 4541 (mysqld)
    CGroup: /system.slice/mysqld.service
           4541 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysql/mysqld.pid

Aug 12 03:48:59 ip-10-0-2-125.ec2.internal systemd[1]: Starting MySQL Server...
Aug 12 03:49:05 ip-10-0-2-125.ec2.internal systemd[1]: Started MySQL Server.
[root@ip-10-0-2-125 ec2-user]# TEMP_PWD=$(sudo grep 'temporary password' /var/log/mysqld.log | awk '{printf "%s", $11}')
[root@ip-10-0-2-125 ec2-user]# echo TEMP_PWD
TEMP_PWD
[root@ip-10-0-2-125 ec2-user]# echo $TEMP_PWD
_CrqVQ=m9+
[root@ip-10-0-2-125 ec2-user]# 
```

>

<Insert screenshot a(3) here -Executing the provided script



```

Activities Aug 12 00:12
root@ip-10-0-2-125:/home/ec2-user
root@ip-10-0-2-125:/home/ec2-user

ec2-user@ip-10-0-2-126:~ 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!';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'ALTER USER 'root'@'localhost' IDENTIFIED BY 'Password42!'' at line 1
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'Password42!';
Query OK, 0 rows affected (0.01 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'Password42!'
-> EXIT
-> exit
-> 'exit'
-> EXIT
-> Quit
-> bye
-> ^C
mysql>
mysql> ^C
mysql> exit
Bye
[root@ip-10-0-2-125 ec2-user]# wget https://d6opu47qoi4ee.cloudfront.net/install_mysql_linux.sh
--2024-08-12 04:09:04-- https://d6opu47qoi4ee.cloudfront.net/install_mysql_linux.sh
Resolving d6opu47qoi4ee.cloudfront.net (d6opu47qoi4ee.cloudfront.net)... 3.162.115.180, 3.162.115.193, 3.162.115.128, ...
Connecting to d6opu47qoi4ee.cloudfront.net (d6opu47qoi4ee.cloudfront.net)|3.162.115.180|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 189 [text/x-sh]
Saving to: 'install_mysql_linux.sh'

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

2024-08-12 04:09:05 (14.6 MB/s) - 'install_mysql_linux.sh' saved [189/189]

[root@ip-10-0-2-125 ec2-user]# chmod 777 install_mysql_linux.sh
[root@ip-10-0-2-125 ec2-user]# sudo ./install_mysql_linux.sh
mysql: [Warning] Using a password on the command line interface can be insecure.
[root@ip-10-0-2-125 ec2-user]#

```

>

Step number b

Step name Installation and configuration of Mattermost

Instructions 1) Enter the following commands after logging into the application server via SSH to install and configure Mattermost.

```
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
```

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.

`http://YOUR_APP_PUBLIC_IP:8065`

- Expected screenshots
- 1) Executing the script
 - 2) Starting the Mattermost server
 - 3) Accessing the application via web browser

<Insert screenshot b(1) here - Executing the script

```
root@ip-10-0-1-14:/home/ec2-user
simo@simo-QEMU-Virtual-Machine:~$ cd Downloads
simo@simo-QEMU-Virtual-Machine:~/Downloads$ ssh -i "Project1_AppServer_KeyPair.pem" ec2-user@ec2-3-95-201-95.compute-1.amazonaws.com
The authenticity of host 'ec2-3-95-201-95.compute-1.amazonaws.com (3.95.201.95)' can't be established.
ED25519 key fingerprint is SHA256:hE4FBIIIZ2Iy9AE0k/N8uE+9+Hb3XyFw311g5t7PBk.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:6: [hashed name]
  ~/.ssh/known_hosts:7: [hashed name]
  ~/.ssh/known_hosts:8: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-95-201-95.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Mon Aug 12 02:40:36 2024 from c-73-227-205-250.hsd1.ma.comcast.net
,
# ~\_\_ #####      Amazon Linux 2
~~ \_\_\#\#\#\` AL2 End of Life is 2025-06-30.
~~ \#\#\`          V-`'-->
~~ \#/             / A newer version of Amazon Linux is available!
~~ .-.` /`         Amazon Linux 2023, GA and supported until 2028-03-15.
~~ /` /`           https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-10-0-1-14 ~]$ sudo su
[root@ip-10-0-1-14 ec2-user]# wget https://d6opu47qoi4ee.cloudfront.net/install_mattermost_linux.sh
--2024-08-12 14:53:51-- https://d6opu47qoi4ee.cloudfront.net/install_mattermost_linux.sh
Resolving d6opu47qoi4ee.cloudfront.net (d6opu47qoi4ee.cloudfront.net)... 3.162.115.193, 3.162.115.128, 3.162.115.161, ...
Connecting to d6opu47qoi4ee.cloudfront.net (d6opu47qoi4ee.cloudfront.net)|3.162.115.193|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 592 [text/x-sh]
Saving to: 'install_mattermost_linux.sh'

100%[=====] 592      --K/s   in 0s
2024-08-12 14:53:52 (20.9 MB/s) - 'install_mattermost_linux.sh' saved [592/592]
[root@ip-10-0-1-14 ec2-user]# sudo yum install dos2unix -y
```

```
root@ip-10-0-1-14:/home/ec2-user
Saving to: 'install_mattermost_linux.sh'
100%[=====] 592      ---K/s  in 0s
2024-08-12 14:53:52 (20.9 MB/s) - 'install_mattermost_linux.sh' saved [592/592]

[root@ip-10-0-1-14 ec2-user]# sudo yum install dos2unix -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package dos2unix.x86_64 0:6.0.3-7.amzn2.0.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch       Version        Repository      Size
=====
Installing:
dos2unix          x86_64    6.0.3-7.amzn2.0.3   amzn2-core      75

Transaction Summary
=====
Install 1 Package

Total download size: 75 k
Installed size: 190 k
Downloading packages:
dos2unix-6.0.3-7.amzn2.0.3.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : dos2unix-6.0.3-7.amzn2.0.3.x86_64
  Verifying  : dos2unix-6.0.3-7.amzn2.0.3.x86_64
                                                               1/1
                                                               1/1

Installed:
dos2unix.x86_64 0:6.0.3-7.amzn2.0.3

Complete!
[root@ip-10-0-1-14 ec2-user]#
```

```
root@ip-10-0-1-14:/home/ec2-user
Installed:
dos2unix.x86_64 0:6.0.3-7.amzn2.0.3

Complete!
[root@ip-10-0-1-14 ec2-user]# chmod 700 install_mattermost_linux.sh
[root@ip-10-0-1-14 ec2-user]# sudo ./install_mattermost_linux.sh 10.0.2.125
rm: cannot remove '/opt/mattermost': No such file or directory
--2024-08-12 14:59:38-- https://releases.mattermost.com/5.19.0/mattermost-5.19.0-linux-amd64.tar.gz
Resolving releases.mattermost.com (releases.mattermost.com)... 18.165.83.6, 18.165.83.36, 18.165.83.102, ...
Connecting to releases.mattermost.com (releases.mattermost.com)|18.165.83.6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 155314485 (148M) [application/x-gzip]
Saving to: 'mattermost-5.19.0-linux-amd64.tar.gz'

100%[=====] 155,314,485 24.1MB/s   in 6.5s

2024-08-12 14:59:45 (22.8 MB/s) - 'mattermost-5.19.0-linux-amd64.tar.gz' saved [155314485/155314485]

Downloaded Mattermost
mattermost/
mattermost/client/
mattermost/client/18.11f0f217b2217f7cd67.js
mattermost/client/icon_16x16.png
mattermost/client/14.ec1c246b041acc156729.js.map
mattermost/client/32.b198dd14910d6966658c.js
mattermost/client/11.fa060d5c7252f0465e34.js
mattermost/client/manifest.json
mattermost/client/.b91881f719444beb9f76.js.map
mattermost/client/images/
mattermost/client/images/img_trans.gif
mattermost/client/images/favicon/
mattermost/client/images/favicon/favicon-32x32.png
mattermost/client/images/favicon/apple-touch-icon-60x60.png
mattermost/client/images/favicon/96x96.png
mattermost/client/images/favicon/apple-touch-icon-144x144.png
mattermost/client/images/favicon/redfavicon-16x16.png
mattermost/client/images/favicon/apple-touch-icon-76x76.png
mattermost/client/images/favicon/android-chrome-192x192.png
mattermost/client/images/favicon/apple-touch-icon-72x72.png
mattermost/client/images/favicon/16x16.png
mattermost/client/images/favicon/apple-touch-icon-152x152.png
mattermost/client/images/favicon/apple-touch-icon-57x57.png
```

>

<Insert screenshot b(2) here - Starting the Mattermost server

```
root@ip-10-0-1-14:/opt/mattermost
mattermost/prepackaged_plugins/mattermost-plugin-antivirus-v0.1.1.tar.gz
mattermost/prepackaged_plugins/mattermost-plugin-nps-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-github-v0.11.0.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-jira-v2.2.2.tar.gz
mattermost/prepackaged_plugins/mattermost-plugin-gitlab-v1.0.1.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
Created user
[root@ip-10-0-1-14 ec2-user]# sudo chown -R mattermost:mattermost /opt/mattermost
[root@ip-10-0-1-14 ec2-user]# sudo chmod -R g+w /opt/mattermost
[root@ip-10-0-1-14 ec2-user]# cd /opt/mattermost
[root@ip-10-0-1-14 mattermost]# sudo -u mattermost ./bin/mattermost
{"level": "info", "ts": 1723475102.873093, "caller": "utils/i18n.go:83", "msg": "Loaded system translations", "for_locale": "en", "from_locale": "/opt/mattermost/i18n/en.json"}
{"level": "info", "ts": 1723475102.8733528, "caller": "app/server_app_adapters.go:58", "msg": "Server is initializing..."}
{"level": "info", "ts": 1723475102.88085, "caller": "sqlstore/supplier.go:212", "msg": "Pinging SQL", "database": "master"}
{"level": "error", "ts": 1723475103.131265, "caller": "app/server_app_adapters.go:125", "msg": "SiteURL must be set. Some features will operate incorrectly if the SiteURL is not set. See documentation for details: http://about.mattermost.com/default-site-url"}
{"level": "info", "ts": 1723475103.1362007, "caller": "app/license.go:39", "msg": "License key from https://mattermost.com required to unlock enterprise features."}
{"level": "info", "ts": 1723475103.1480303, "caller": "mlog/log.go:166", "msg": "Starting up plugins"}
{"level": "info", "ts": 1723475103.1481943, "caller": "app/plugin.go:213", "msg": "Syncing plugins from the file store"}
{"level": "info", "ts": 1723475106.080228, "caller": "mlog/sugar.go:19", "msg": "Ensuring Surveybot exists", "plugin_id": "com.mattermost.nps"}
{"level": "info", "ts": 1723475106.7264433, "caller": "app/server.go:217", "msg": "Current version is 5.19.0 (5.19.0/Thu Jan 16 18:30:33 UTC 2020 0cf883f84000d6fd0e25308ad14d56e6ed53f05/1268390c0cde16f750b0b6fe025348b2586d595f)"}
{"level": "info", "ts": 1723475106.7266195, "caller": "app/server.go:218", "msg": "Enterprise Enabled: true"}
{"level": "info", "ts": 1723475106.7267084, "caller": "app/server.go:221", "msg": "Printing current working", "directory": "/opt/mattermost"}
{"level": "info", "ts": 1723475106.7267969, "caller": "app/server.go:222", "msg": "Loaded config", "source": "file:///opt/mattermost/config/config.on"}
{"level": "info", "ts": 1723475106.7535732, "caller": "sqlstore/post_store.go:1351", "msg": "Post.Message has size restrictions", "max_characters": 6383, "max_bytes": 65535}
{"level": "info", "ts": 1723475106.8163848, "caller": "jobs/workers.go:68", "msg": "Starting workers"}
{"level": "info", "ts": 1723475106.8247156, "caller": "app/web_hub.go:75", "msg": "Starting websocket hubs", "number_of_hubs": 2}
{"level": "info", "ts": 1723475106.8283193, "caller": "jobs/schedulers.go:74", "msg": "Starting schedulers."}
{"level": "info", "ts": 1723475106.8361568, "caller": "app/server.go:440", "msg": "Starting Server..."}
{"level": "info", "ts": 1723475106.836393, "caller": "app/server.go:506", "msg": "Server is listening on [::]:8065"}
```

>

<Insert screenshot b(3) here - Accessing the application via web browser

The terminal window shows the user is connected to an EC2 instance at ip-10-0-1-14. The user runs the command `ssh -i Project1_App.pem ec2-user@ip-10-0-1-14 /opt/mattermost`. The output of the command shows various logs, including a warning about the end of life for Amazon Linux (2025-06-30), a notice about a newer version of Amazon Linux available, and a message from the Mattermost server indicating it is initializing. The browser window shows the Mattermost application at `54.235.29.133:8065/signup_email`. The page has a header "Mattermost" and subtext "All team communication in one place, searchable and accessible anywhere". It includes fields for "What's your email address?", "Choose your username", and "Choose your password".

>

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

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

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

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

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

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

Port 3306 on the 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?

By Enabling or configuring the following:

Security Groups:

Define inbound and outbound traffic rules for EC2 instances.
Allow traffic only from specific IP addresses or security groups.

Network ACLs (Network Access Control Lists):

Control traffic at the subnet level for additional security.
Deny all traffic by default and allow only necessary traffic.

AWS WAF (Web Application Firewall):

Protect web applications from common web exploits.
Filter traffic based on IP addresses, country, and other criteria.

AWS Shield:

Protect against DDoS attacks.

CloudWatch:

Monitor VPC flow logs to track network traffic.
Set up alarms for suspicious activity.

IP Whitelisting: Only allow access from specific IP addresses or IP ranges.

VPN or Proxy: Require users to access resources through a VPN or proxy for added security.

IAM Roles with Temporary Credentials: Limit the lifetime of credentials for enhanced security (required new credentials every 3 months).

- Q8 How do we achieve elasticity for the Mattermost server?

Auto Scaling Group (ASG)

The ASG automatically scales the number of Mattermost instances based on load.

Load Balancing

The ALB distributes traffic across healthy instances.

Monitoring and Alerting:

- Amazon CloudWatch to monitor Mattermost server performance and set up alarms for critical metrics.
 - Integrate with CloudWatch Events to trigger actions based on alarm states.
- DR plan:** implement a Disaster Recovery plan in case of any disaster

Grade distribution	
MCQs	6 (1 point each)
Subjective questions	10 (5 points each)
Implementation screenshots	24 points (1 point each)
Total	40 points