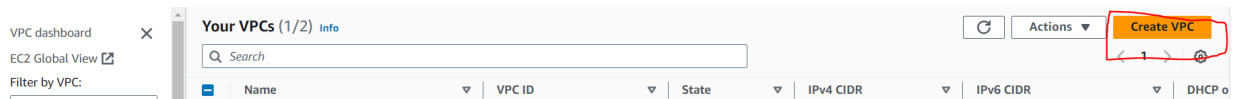Step 1: Create a VPC and Subnets as well as routing and security groups
- Go to "Your VPCs" from the VPC service on the AWS management console and click on the orange "Create VPC" button.

- Only create a vpc here and give it a name. You are free to make your own name or follow along with the one put here
- GIve it a 192.168.0.0/16 CIDR block and leave everything else as default. Click create.

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or create VPC, subnets, etc.

- ● VPC only
- ○ VPC, subnets, etc.

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.

Demo VPC

IPv4 CIDR block Info
- ● IPv4 CIDR manual input
- ○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

192.168.0.0/16

IPv6 CIDR block Info
- ● No IPv6 CIDR block
- ○ IPAM-allocated IPv6 CIDR block
- ○ Amazon-provided IPv6 CIDR block
- ○ IPv6 CIDR owned by me

Tenancy Info

Default ▼

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*	
🔍 Name ✕	🔍 Demo VPC ✕	Remove

Add new tag

You can add 49 more tags.

Cancel   Create VPC

- To create your subnets go to Subnets on the left hand side of the VPC service and click on it
- Add your VPC ID to where it asks

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-03bd2b389d2a4d45e (Demo VPC) ▼

Associated VPC CIDRs

IPv4 CIDRs

192.168.0.0/16

- Assign it a name letting you know what it is your first public subnet
- Put it in any availability zone and give it a CIDR of 192.168.1.0/24

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Public Subnet

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US West (Oregon) / us-west-2a ▼

IPv4 CIDR block Info

🔍 192.168.1.0/24 ✕

▼ Tags - optional

Key | Value - optional
🔍 Name ✕ | 🔍 Public Subnet ✕ | Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

Cancel **Create subnet**

- Add a second subnet and name it Private Subnet 1 or something to let you know it is your first private subnet
- Put it in the same availability zone as the first subnet you made and give it a CIDR of 192.168.2.0/24

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Private Subnet 1

The name can be up to 256 characters long.

Availability Zone **Info**
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US West (Oregon) / us-west-2a ▼

IPv4 CIDR block **Info**

🔍 192.168.2.0/24 ✕

▼ Tags - *optional*

Key

🔍 Name ✕

Value - *optional*

🔍 Private Subnet 1 ✕

Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

- Add a third subnet and assign a name letting you know it is the second private subnet you will be making
- Put it in the same availability zone as your first public subnet and give it a CIDR of 192.168.3.0/24

Subnet 3 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Private Subnet 2

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US West (Oregon) / us-west-2a ▼

IPv4 CIDR block Info

Q 192.168.3.0/24 ✕

▼ Tags - *optional*

Key   Value - *optional*

Q Name   ✕   Q Private Subnet 2   ✕   Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

- Add a fourth and final subnet and give it a name letting you know it is the third private subnet
- Put it in a different availability zone from the rest of your subnets and give it a CIDR of 192.168.4.0/24

Subnet 4 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Private Subnet 3

The name can be up to 256 characters long.

Availability Zone   Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US West (Oregon) / us-west-2b   ▼

IPv4 CIDR block   Info

🔍 192.168.4.0/24   ✕

▼ Tags - *optional*

Key   Value - *optional*

🔍 Name   ✕   🔍 Private Subnet 3   ✕   Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

- Set up for route tables
- Allocate an Elastic IP address by going to Elastic IPs on the left hand side and click "Allocate Elastic IP address"
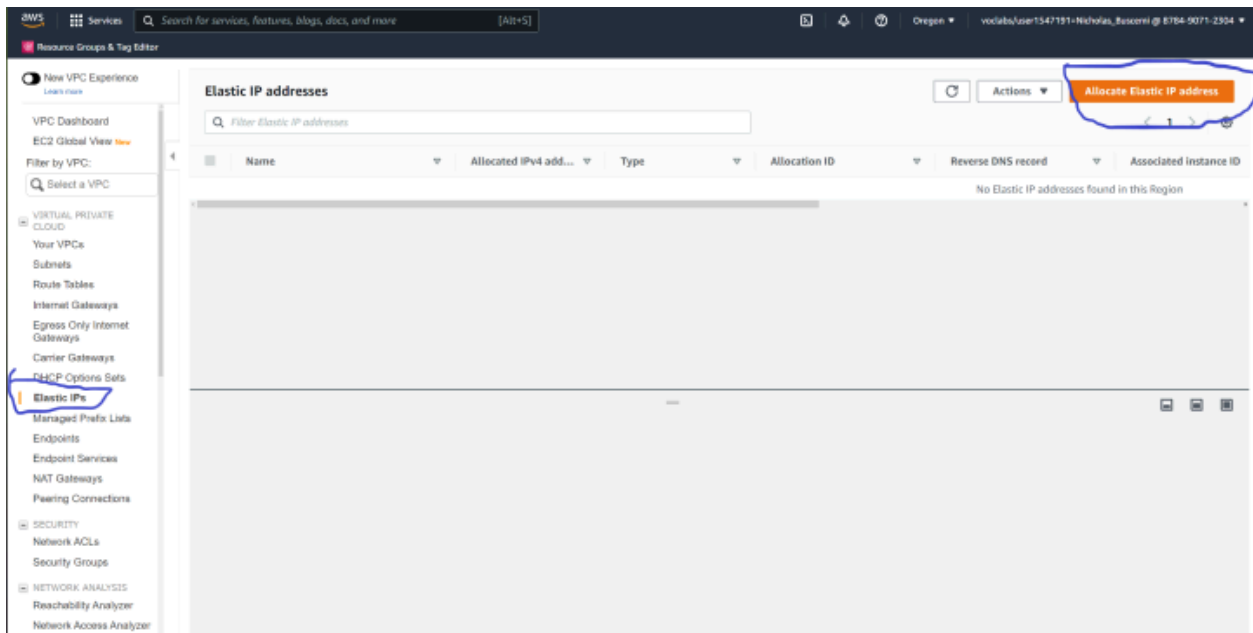
- Everything should be good as default but make sure that you are in the same region you have been creating everything in and then press "Allocate". You can also add a name tag if you wish but it isn't necessary

- Now create an internet gateway and attach it to the VPC by going to Internet Gateways on the left hand side and clicking "Create Internet Gateway"
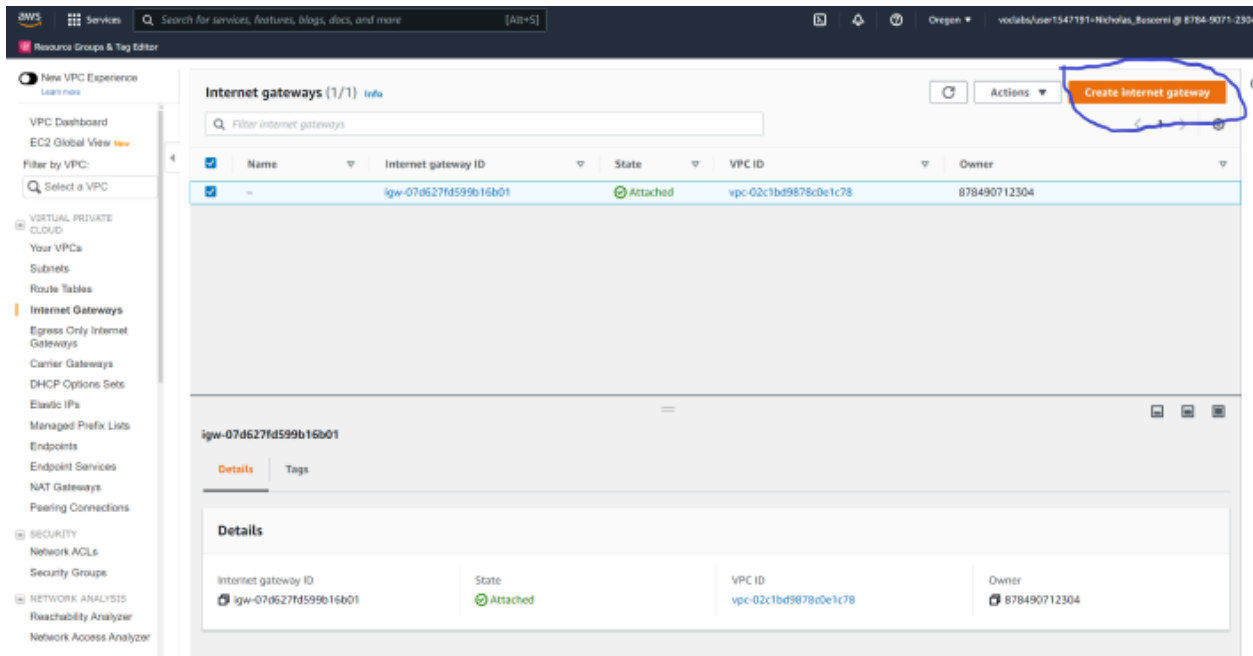
- Name it something similar to what is below and then click "Create Internet Gateway"
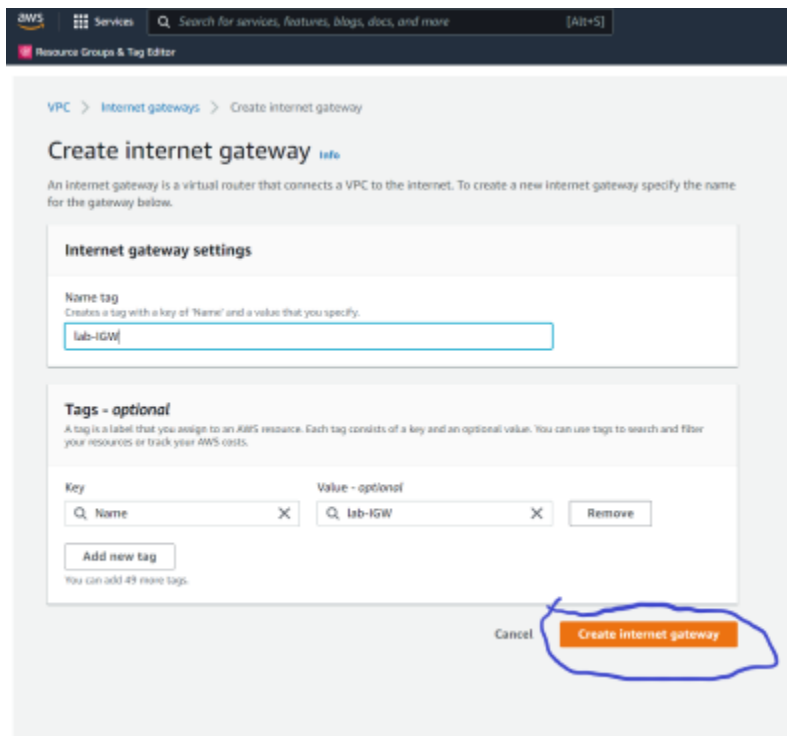
- Once it is created attach it to your VPC by clicking "Attach to a VPC" on the top of the screen
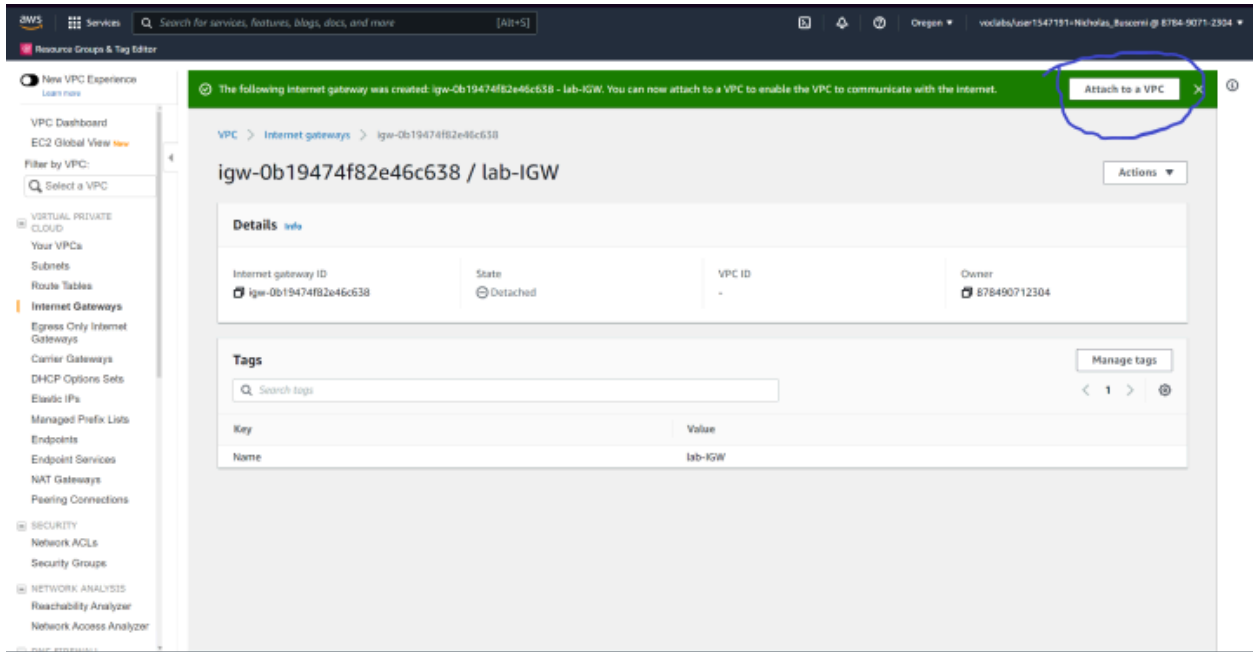
- Click the drop down and select your vpc that you made
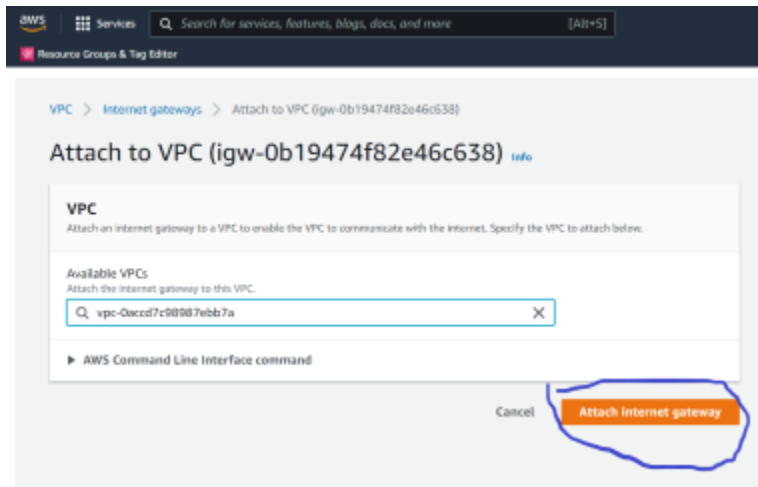
- Create a NAT Gateway by clicking on Nat Gateways on the left hand side and then clicking "Create NAT Gateway"
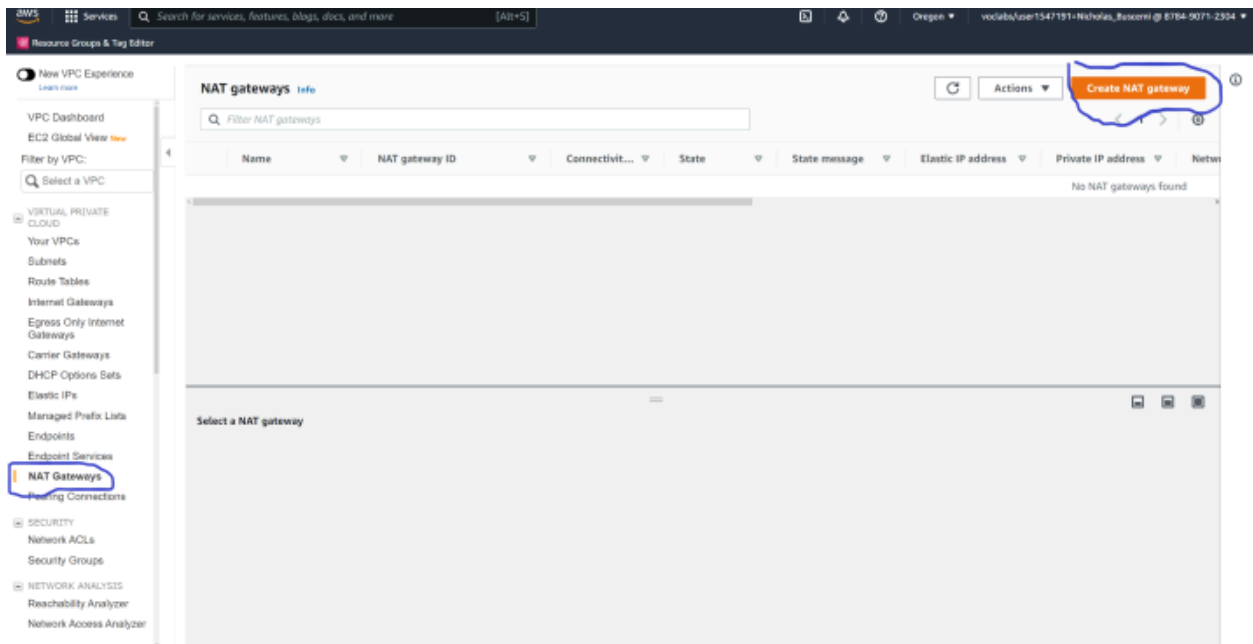
- Give it a name similar to the one below and assign it to a **public subnet**
- Click the drop down for Elastic IPs and click the one you created previously
- Click "Create NAT gateway"

- Create Route Tables by first heading to "Route Tables" on the left hand side
- Click "Create route table"

- Give it a name letting you know this is the public route table for your lab
- Assign your VPC to it and click "Create route table"

- Make a second route table naming it something to let you know that this is the private route table for your lab and assign your VPC to it

- Now associate your subnets with their respective route table
- Click on the public route table and click on "Subnet association" next to "Details"

- Click on "Edit subnet associations"

- Click on your public subnet and then click "Save associations"

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/4)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
☐	Private Subnet 3	subnet-092c5783f28705ad7	192.168.4.0/24	–	Main (rtb-010a67db84e50ef4
☑	Public Subnet	subnet-06bc254b8826ccb04	192.168.1.0/24	–	Main (rtb-010a67db84e50ef4
☐	Private Subnet 1	subnet-0b2f703b31b1dbf78	192.168.2.0/24	–	Main (rtb-010a67db84e50ef4
☐	Private Subnet 2	subnet-00fbb6995bc7ecbc6	192.168.3.0/24	–	Main (rtb-010a67db84e50ef4

Selected subnets

subnet-06bc254b8826ccb04 / Public Subnet ✕

Cancel   **Save associations**

- Now add a route to our public route table to get access to the internet gateway
- Click on "Routes" next to "Details" and click "Edit routes"

- Add a new route having a destination of anywhere and a target of your internet gateway and click "Save changes"

Edit routes

Edit routes

Destination	Target	Status
192.168.0.0/16	🔍 local ✕	⊘ Active

Propagated

No

Edit routes

Destination	Target	Status
🔍 0.0.0.0/0 ✕	🔍 igw-0a2e406efd9302f16 ✕	–

Propagated

No

Remove

Add route

Cancel   Preview   Save changes

- Do the same thing for your private route table by clicking on it and going to its subnet associations and editing them

- Click on all three of your private subnets and save the associations

● Go to edit the routes of the private table

- Add a route to the private table that has a destination of anywhere and a target of your Nat gateway that you made earlier

Edit routes

Edit routes

Destination	Target	Status
192.168.0.0/16	Q local ✕	✓ Active

Propagated

No

Edit routes

Destination	Target	Status
Q 0.0.0.0/0 ✕	Q nat-0dc88ba1b12f5d4bf ✕	–

Propagated

No

Remove

Add route

Cancel   Preview   **Save changes**

Subnets (4)	Route tables (3)	Network connections (2)
Subnets within this VPC	Route network traffic to resources	Connections to other networks
us-west-2a	rtb-0a84ee78be6761821	lab-IGW
Public Subnet-1	lab-public-route	**lab-nat**
Private Subnet-1	lab-private-route	Public NAT gateway
Private Subnet-2		1 ENI with 1 EIP
us-west-2b		
Private Subnet-3		
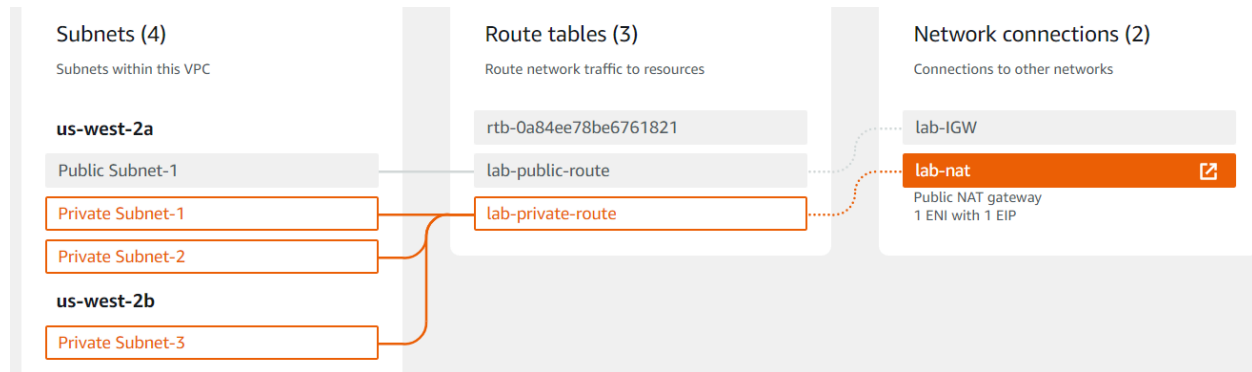
- Now to create our security groups (One for our bastion host, web server, app server, and our database) we will head to Security Groups on the left and click "Create security group"
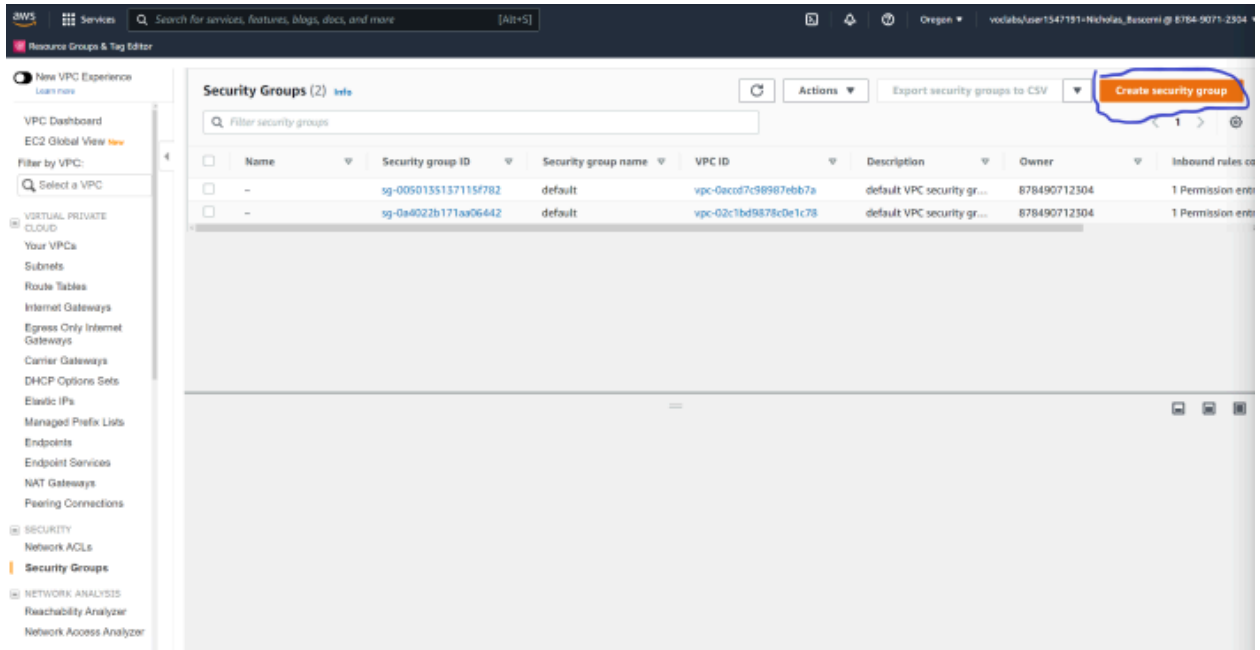
- Give it a name and description letting you know it is for a bastion host
- Assign your VPC to it
- Give it three inbound rules, one for SSH using your IP and one for HTTP using 0.0.0.0/0 as well as https using 0.0.0.0/0.

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name   Info

```
MyBastionHostSG
```

Name cannot be edited after creation.

Description   Info

```
My Bastion Host Security Group
```

VPC   Info

```
🔍  vpc-03bd2b389d2a4d45e                                      ✕
```

Inbound rules   Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
HTTP ▼	TCP	80	Anywh... ▼ 🔍		Del ete
			0.0.0.0/0 ✕		
HTTPS ▼	TCP	443	Anywh... ▼ 🔍		Del ete
			0.0.0.0/0 ✕		
SSH ▼	TCP	22	Anywh... ▼ 🔍		Del ete
			0.0.0.0/0 ✕		

Add rule

- Create another security group
- Give it a name and description letting you know it is for a Web server
- Assign your VPC to it
- Give it the same inbound rules as the Bastion Host security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name   Info

MyWebServerSG

Name cannot be edited after creation.

Description   Info

My Web Server Security Group

VPC   Info

🔍 vpc-03bd2b389d2a4d45e   ✕

Inbound rules   Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
HTTP ▼	TCP	80	Anywh... ▼ 🔍 0.0.0.0/0 ✕		Delete
HTTPS ▼	TCP	443	Anywh... ▼ 🔍 0.0.0.0/0 ✕		Delete
SSH ▼	TCP	22	Anywh... ▼ 🔍 0.0.0.0/0 ✕		Delete

Add rule

- Create another security group
- Give it a name and description letting you know it is for an **app server**
- Assign your VPC to it
- Give it an inbound rule for **All ICMP -IPv4** with **a source of your web server SG** and **another inbound rule for SSH** with a **source** of your **bastion host SG**

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

```
MyAppServerSG
```

Name cannot be edited after creation.

Description Info

```
My App Server Security Group
```

VPC Info

```
Q  vpc-03bd2b389d2a4d45e                          ✕
```

Inbound rules Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
All ICMP - IPv4 ▼	ICMP	All	Custom ▼ Q		Del ete
			sg-094827ea3d 4c27f60 ✕		
SSH ▼	TCP	22	Custom ▼ Q		Del ete
			sg-0418120089 30fcc7b ✕		

Add rule

- Create one final security group
- Give it a name and description letting you know it is for a database server
- Assign your VPC to it
- Give it two inbound rules **both for MYSQL/Aurora** and give one of them a **source of your app server SG and the other one a source of your bastion host SG**

- Go back to your **bastion host inbound rules** and add one more for MYSQL/Aurora and a source of your **database SG**

- Go back to your web server inbound rules and add one more for All ICMP - IPv4 and a source of your app server SG.

- Go back to your app server inbound rules and add one more for MYSQL/Aurora and a source of your database SG and then an HTTP and HTTPS rule both with a source of 0.0.0.0/0

Step 2: Create Servers

- Create Bastion Host

- Select Amazon Linux 2 AMI

Quick Start

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Li
aws	Mac	ubuntu	Microsoft	Red Hat	SUSE

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type   Free tier eligible
ami-0895022f3dac85884 (64-bit (x86)) / ami-09ebdd80c5c138f65 (64-bit (Arm))
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20240223.0 x86_64 HVM gp2

Architecture   AMI ID

64-bit (x86)   ▼   ami-0895022f3dac85884   Verified provider

- Select t2.micro

▼ **Instance type** Info | Get advice

Instance type

t2.micro   Free tier eligible
Family: t2   1 vCPU   1 GiB Memory   Current generation: true
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour   ▼
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour

○ All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

- Select the key pair

▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

vockey   ▼   ↻ Create new key pair

- Put in your VPC and Public Subnet, enable auto assign public IP and Select an existing group and select your Bastion Host SG.

VPC - *required*   Info

vpc-064d8b2be3d4d8d9d (Demo-vpc)
192.168.0.0/16   ▼   ↻

Subnet   Info

subnet-0f1d65ae74aee11cd   Public Subnet-1
VPC: vpc-064d8b2be3d4d8d9d   Owner: 905418371971
Availability Zone: us-west-2a   IP addresses available: 250   CIDR: 192.168.1.0/24   ▼   ↻   Create new subnet ⧉

Auto-assign public IP   Info

Enable   ▼

Firewall (security groups)   Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group   ◉ Select existing security group

Common security groups Info

Select security groups   ▼

Baston Host   sg-0614859e85cf4602a ✕
VPC: vpc-064d8b2be3d4d8d9d

↻   Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▶ Advanced network configuration

- Storage leaves default.

- Follow the same steps to create the Web Server
- Follow along like previously and change your network, subnet, and enable auto assign public ip

- Storage leave default

- Then go to user data and type this into it to set up the web server

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
```

User data - *optional*   **Info**
Upload a file with your user data or enter it in the field.

⤓ Choose file

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
Sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
```

☐ User data has already been base64 encoded

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0895022f3dac85884

Virtual server type (instance type)
t2.micro

Firewall (security group)
Web-Server

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   ✕

Cancel   **Launch instance**

Review commands

- Creating an App server
- Put in your VPC and then choose **Private Subnet 1** for the subnet and leave auto assign public ip disabled

- Then go into Advanced Detail section add the below script in user data column

```
#!/bin/bash
sudo yum update -y
sudo yum install -y mariadb-server
sudo service mariadb start
```

Note:
To check version of mariadb use **mysql -v**

Step 3: Create a Database
- From the management console search RDS.
- In left pane, find **DB subnet group.** Create a DB Subnet group.

- Give it a name and description letting you know what it is and then assign your VPC to it
- Put in the availability zones you used for your subnets
- Select subnets 3 and 4
- Click create

- Go to Databases on the left hand side and click on "Create Database"

aws | Services | Q Search for services, features, blogs, docs, and more [Alt+S]
Oregon ▼ voclabs/user1547191=Nicholas_Buscemi @ 8784-9071-2304
Resource Groups & Tag Editor

Amazon RDS ✕

✅ Successfully created lab-DB-subnet-group. View subnet group

Dashboard
Databases
Query Editor
Performance insights
Snapshots
Automated backups
Reserved instances
Proxies

Subnet groups
Parameter groups
Option groups
Custom engine versions

Events
Event subscriptions

Recommendations ⓪
Certificate update

RDS > Databases

Databases

Group resources | ↻ | Modify | Actions ▼ | Restore from S3 | **Create database**

🔍 Filter by databases

< 1 > ⚙

	DB identifier ▲	Role ▽	Engine ▽	Region & AZ ▽	Size ▽	Status ▽	CPU	Current activity	Maintenance ▽
				No instances found					

- Click on Standard create and MariaDB for the engine type

- Make sure you click on Free tier here

- Give it an identifier you can easily identify it with
- Give it a master username or leave it as default admin. For the purpose of these instructions I will be using **root**
- Give it a password that you write down somewhere else to make sure you have the correct one. For the purpose of these instructions I will be using **Re:Start!9**

Settings

DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

dbinstance

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username Info

Type a login ID for the master user of your DB instance.

root

1 to 16 alphanumeric characters. First character must be a letter.

☐ Auto generate a password
    Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

••••••••••

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm password Info

••••••••••

- Everything between this and the last step is left default
- Assign your vpc
- Make sure your subnet group is listed under the subnet group section
- Public access is no
- Choose existing VPC security groups
- Remove the default security group and add your database security group
- Select your first availability zone as well

Connectivity

Virtual private cloud (VPC) Info
VPC that defines the virtual networking environment for this DB instance.

Demo VPC (vpc-03bd2b389d2a4d45e) ▼

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

Subnet group Info
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

dbsubnetgroup ▼

Public access Info

○ Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

● No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group
Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

● Choose existing
Choose existing VPC security groups

○ Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

MyDatabaseServerSG ✕

Availability Zone Info

us-west-2a ▼

- Scroll down to Additional configuration on the bottom and give it an initial database name and save it in the same spot as your password since it will be used later
- Disable automated backups and encryption since they are not needed (These are normally best practice to leave enabled but the database will spin up faster with those checked off as they are not needed).
- Scroll down all the way to the bottom and create your database

▼ **Additional configuration**
Database options, encryption disabled, backup disabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled.

Database options

Initial database name **Info**

```
mydb
```

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group **Info**

```
default.mariadb10.6                                      ▼
```

Option group **Info**

```
default:mariadb-10-6                                     ▼
```

Backup

☐ Enable automated backups
Creates a point-in-time snapshot of your database

Encryption

☐ Enable encryption
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. **Info**

Monitoring

☐ Enable Enhanced monitoring
Enabling Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Log exports

Select the log types to publish to Amazon CloudWatch Logs
☐ Audit log
☐ Error log

Step 4: Test connections

- SSH into your Bastion Host after downloading both the pem and ppk files from the lab environment
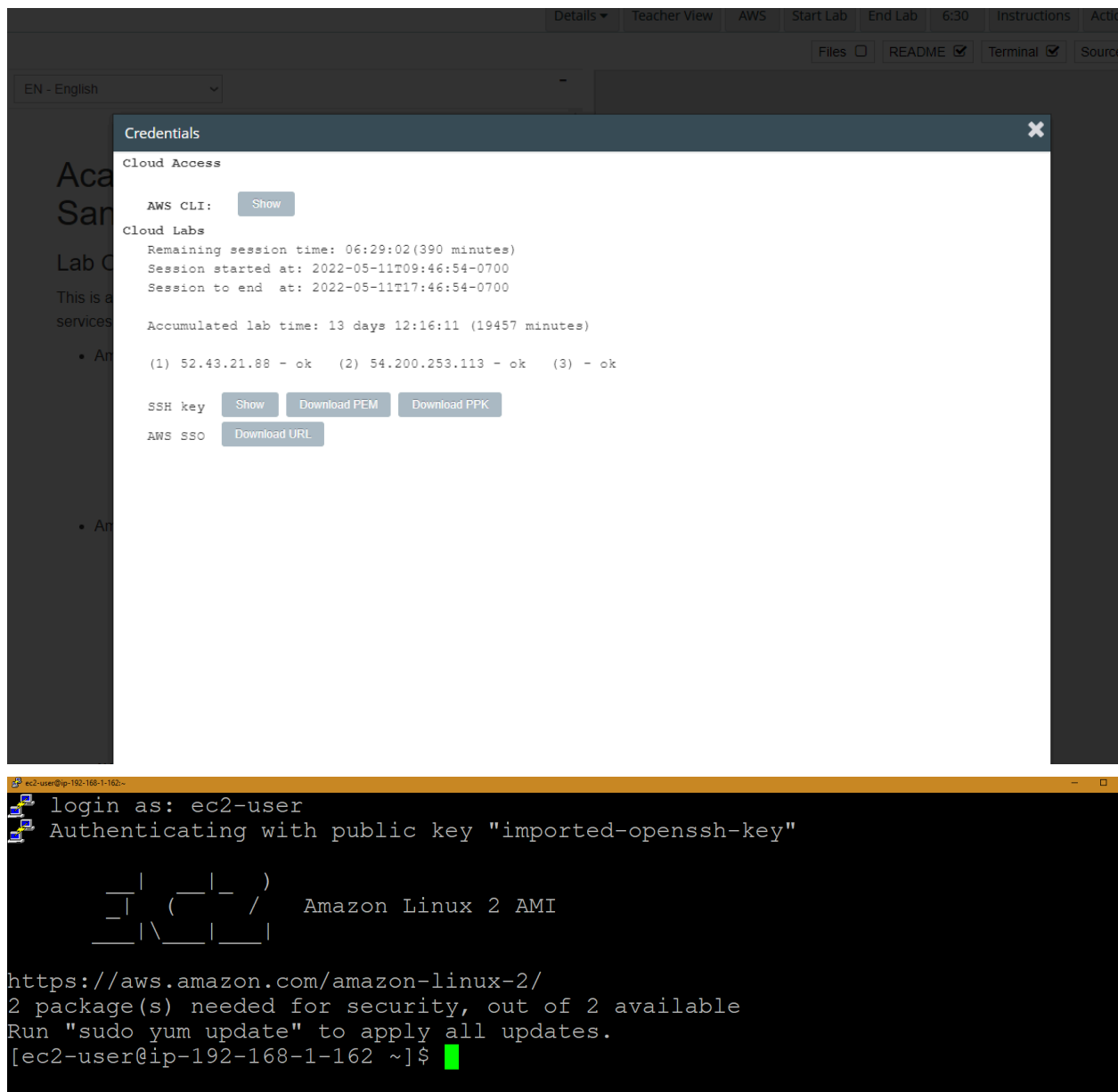
How to access the Host machine from Baston Machine?

1. Download the respective pem file
2. Open Baston machine and copy the contents of pem file
3. Create a new file in baston machine

4. Save all the contents of the pem file in this.

ssh -i vockey.pem ec2-user@192.168.2.138
Yes

 mysql -up <RDS Endpoint> -p <username>
Ex:
 mysql -h
terraform-20240309153133293100000001.c1isoy422nnt.us-west-2.rds.amazonaws.com -u root
-p

Enter the password:

Copy the contents of
- This is for windows only as I only have a windows machine to work on, sorry mac and linux users
- Go into your powershell and type this command out
  - Pscp -scp -P 22 -i '.\Downloads\labsuser.ppk' -| user ec2-user '.\Downloads\labsuser.pem' ec2-user@bastion-host-public-ip:/home/ec2-user
  - Replace bastion-host-public-ip with the public ip address of your bastion host
- Hit enter and it should upload those keys to your bastion host for use on other servers

- Test on your ssh to see if the file is uploaded by using ls. Should return something like this

- Change file permissions for the file we just downloaded to our bastion host by typing
  - chmod 400 labsuser.pem
- Then ssh into our app server by typing
  - ssh -i my-key-pair.pem ec2-user@app-server-private-ip
  - Replace my-key-pair with the name of your key
  - Replace app-server-private-ip with your app server's private ip address
- Type yes when it prompts you to

- Use ls to see that you are now ssh into a different server since there is no more key

```
[ec2-user@ip-192-168-1-162 ~]$ chmod 400 labsuser.pem
[ec2-user@ip-192-168-1-162 ~]$ ssh -i labsuser.pem ec2-user@192.168
.2.172
The authenticity of host '192.168.2.172 (192.168.2.172)' can't be e
stablished.
ECDSA key fingerprint is SHA256:RURnbNWL6+XNSA3+S9k0FtM1Fy0aAHcv4z7
qLtKlm7A.
ECDSA key fingerprint is MD5:ce:f3:58:79:65:eb:ae:de:6a:3c:51:5c:89
:4b:69:88.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.172' (ECDSA) to the list of k
nown hosts.

       __|  __|_  )
       _|  (     /    Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-2-172 ~]$ ls
[ec2-user@ip-192-168-2-172 ~]$ 
```

- Use ping and the private ip address of your web server to ping the web server and see it connect

```
[ec2-user@ip-192-168-2-172 ~]$ ping 192.168.1.252
PING 192.168.1.252 (192.168.1.252) 56(84) bytes of data.
64 bytes from 192.168.1.252: icmp_seq=1 ttl=255 time=0.486 ms
64 bytes from 192.168.1.252: icmp_seq=2 ttl=255 time=0.441 ms
64 bytes from 192.168.1.252: icmp_seq=3 ttl=255 time=0.450 ms
64 bytes from 192.168.1.252: icmp_seq=4 ttl=255 time=0.483 ms
^C
--- 192.168.1.252 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3052ms
rtt min/avg/max/mdev = 0.441/0.465/0.486/0.019 ms
[ec2-user@ip-192-168-2-172 ~]$ 
```

- Test out connecting to the database by typing out mysql –user=root -password='Re:Start!9' –host=database-server-endpoint
- Replace database-server-endpoint with the database server endpoint
- Type show databases; to see your database from the app server

Connection details to your database database ✕

This is the only time you can view this password. Copy and save the password for your reference. If you lose the password, you must modify your database to change it. You can use a SQL client application or utility to connect to your database.

Learn about connecting to your database ↗

Master username

root

Master password

Re:Start!9   Copy

Endpoint

database.c1isoy422nnt.us-west-2.rds.amazonaws.com   Copy

Close

```
[ec2-user@ip-192-168-2-31 ~]$ mysql --user=root --password='Re:Start!9' --host=d
atabase.c1isoy422nnt.us-west-2.rds.amazonaws.com
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 83
Server version: 10.11.6-MariaDB managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mydb               |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.01 sec)

MariaDB [(none)]> █
```

Choose lab database

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

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| lab                |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

MySQL [(none)]> use lab;
Database changed
MySQL [lab]>
```

Create a table RESTART with the following columns

- Student ID (Number),
- Student Name,
- Restart City,
- Graduation Date (Date Time)

```
MySQL [(none)]> use lab;
Database changed
MySQL [lab]> CREATE TABLE RESTART(
    ->      Student_ID     INT NOT NULL,
    ->      Student_Name VARCHAR (20)   NOT NULL,
    ->      Restart_City   CHAR (25) ,
    ->      Graduation_date    datetime,
    ->      PRIMARY KEY (Student_ID)
    -> );
Query OK, 0 rows affected (0.05 sec)

MySQL [lab]> show tables;
+----------------+
| Tables_in_lab  |
+----------------+
| RESTART        |
+----------------+
1 row in set (0.00 sec)

MySQL [lab]>
```

Insert 10 sample rows into this table

```
MySQL [lab]> insert into RESTART values(101,'user1','Hyderabad','1997-04-20 17:37:00');
insert into RESTART values(109,'user9','Ahamedabad','2015-05-11 04:03:00');
insert into RESTART values(110,'user10','Kolkata','2023-12-05 07:03:10');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]> insert into RESTART values(102,'user2','Bangalore','2000-05-05 18:09:00');
Query OK, 1 row affected (0.00 sec)

MySQL [lab]> insert into RESTART values(103,'user3','Chennai','2001-12-15 05:20:30');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]> insert into RESTART values(104,'user4','Thirunanathapuram','2010-01-05 14:00:00');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]> insert into RESTART values(105,'user5','Panaji','2011-05-06 04:30:00');
Query OK, 1 row affected (0.00 sec)

MySQL [lab]> insert into RESTART values(106,'user6','Mumbai','2012-05-09 05:40:00');
Query OK, 1 row affected (0.00 sec)

MySQL [lab]> insert into RESTART values(107,'user7','Bhopal','2013-05-09 02:18:00');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]> insert into RESTART values(108,'user8','Bhubaneswar','2011-10-05 06:00:30');
Query OK, 1 row affected (0.00 sec)

MySQL [lab]> insert into RESTART values(109,'user9','Ahamedabad','2015-05-11 04:03:00');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]> insert into RESTART values(110,'user10','Kolkata','2023-12-05 07:03:10');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]> 
```

Select all rows from this table

```
MySQL [lab]> select * from RESTART;
+------------+--------------+-------------------+---------------------+
| Student_ID | Student_Name | Restart_City      | Graduation_date     |
+------------+--------------+-------------------+---------------------+
|        101 | user1        | Hyderabad         | 1997-04-20 17:37:00 |
|        102 | user2        | Bangalore         | 2000-05-05 18:09:00 |
|        103 | user3        | Chennai           | 2001-12-15 05:20:30 |
|        104 | user4        | Thirunanathapuram | 2010-01-05 14:00:00 |
|        105 | user5        | Panaji            | 2011-05-06 04:30:00 |
|        106 | user6        | Mumbai            | 2012-05-09 05:40:00 |
|        107 | user7        | Bhopal            | 2013-05-09 02:18:00 |
|        108 | user8        | Bhubaneswar       | 2011-10-05 06:00:30 |
|        109 | user9        | Ahamedabad        | 2015-05-11 04:03:00 |
|        110 | user10       | Kolkata           | 2023-12-05 07:03:10 |
+------------+--------------+-------------------+---------------------+
```

14. Create a table CLOUD_PRACTITIONER with the following columns

- Student ID (Number)
- certification date (Date Time)

```
MySQL [lab]> show tables;
+----------------+
| Tables_in_lab  |
+----------------+
| RESTART        |
+----------------+
1 row in set (0.00 sec)

MySQL [lab]> CREATE TABLE CLOUD_PRACTITIONER (
    ->      Student_ID INT,
    ->      Certification_Date DATETIME
    -> );
Query OK, 0 rows affected (0.04 sec)

MySQL [lab]> show tables;
+--------------------+
| Tables_in_lab      |
+--------------------+
| CLOUD_PRACTITIONER |
| RESTART            |
+--------------------+
2 rows in set (0.01 sec)
```

15. Insert 5 sample rows into this table

```
MySQL [lab]> insert into CLOUD_PRACTITIONER values(101,'1997-04-20 17:37:00');
Query OK, 1 row affected (0.00 sec)

MySQL [lab]> insert into CLOUD_PRACTITIONER values(102,'2000-05-05 18:09:00');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]> insert into CLOUD_PRACTITIONER values(103,'2001-12-15 05:20:30');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]> insert into CLOUD_PRACTITIONER values(104,'2010-01-05 14:00:00');
Query OK, 1 row affected (0.00 sec)

MySQL [lab]> insert into CLOUD_PRACTITIONER values(105,'2011-05-06 04:30:00');
Query OK, 1 row affected (0.01 sec)

MySQL [lab]>
```

16. Select all rows from this table

```
MySQL [lab]> SHOW TABLES;
+-------------------+
| Tables_in_lab     |
+-------------------+
| CLOUD_PRACTITIONER |
| RESTART           |
+-------------------+
2 rows in set (0.01 sec)

MySQL [lab]> SELECT * FROM CLOUD_PRACTITIONER;
+------------+---------------------+
| Student_ID | certification_date  |
+------------+---------------------+
|        101 | 1997-04-20 17:37:00 |
|        102 | 2000-05-05 18:09:00 |
|        103 | 2001-12-15 05:20:30 |
|        104 | 2010-01-05 14:00:00 |
|        105 | 2011-05-06 04:30:00 |
+------------+---------------------+
5 rows in set (0.00 sec)
```

17. Inner join

```
MySQL [lab]> SELECT CP.Student_ID, R.Student_Name, CP.certification_date
    -> FROM CLOUD_PRACTITIONER CP
    -> INNER JOIN RESTART R ON CP.Student_ID = R.Student_ID;
+------------+--------------+---------------------+
| Student_ID | Student_Name | certification_date  |
+------------+--------------+---------------------+
|        101 | user1        | 1997-04-20 17:37:00 |
|        102 | user2        | 2000-05-05 18:09:00 |
|        103 | user3        | 2001-12-15 05:20:30 |
|        104 | user4        | 2010-01-05 14:00:00 |
|        105 | user5        | 2011-05-06 04:30:00 |
+------------+--------------+---------------------+
5 rows in set (0.01 sec)
```

DB instance identifier: mysql-lab-demo
Master username   **:**   main
Password   **:**   vaMsi-20
DB name   **:**   lab
Endpoint   **:**   mysql-lab-demo.c9u9cnfyzkt5.us-west-2.rds.amazonaws.com

mysql -u main -p --host mysql-lab-demo.c9u9cnfyzkt5.us-west-2.rds.amazonaws.com

```
CREATE TABLE RESTART(
   Student_ID   INT NOT NULL,
   Student_Name VARCHAR (20)  NOT NULL,
   Restart_City  CHAR (25) ,
   Graduation_date   datetime,
   PRIMARY KEY (Student_ID)
);

insert into RESTART values(101,'user1','Hyderabad','1997-04-20 17:37:00');
insert into RESTART values(102,'user2','Bangalore','2000-05-05 18:09:00');
insert into RESTART values(103,'user3','Chennai','2001-12-15 05:20:30');
insert into RESTART values(104,'user4','Thirunananthapuram','2010-01-05 14:00:00');
insert into RESTART values(105,'user5','Panaji','2011-05-06 04:30:00');
insert into RESTART values(106,'user6','Mumbai','2012-05-09 05:40:00');
insert into RESTART values(107,'user7','Bhopal','2013-05-09 02:18:00');
insert into RESTART values(108,'user8','Bhubaneswar','2011-10-05 06:00:30');
insert into RESTART values(109,'user9','Ahmedabad','2015-05-11 04:03:00');
insert into RESTART values(110,'user10','Kolkata','2023-12-05 07:03:10');


insert into CLOUD_PRACTITIONER values(101,'1997-04-20 17:37:00');
insert into CLOUD_PRACTITIONER values(102,'2000-05-05 18:09:00');
insert into CLOUD_PRACTITIONER values(103,'2001-12-15 05:20:30');
insert into CLOUD_PRACTITIONER values(104,'2010-01-05 14:00:00');
```

```sql
insert into CLOUD_PRACTITIONER values(105,'2011-05-06 04:30:00');

SELECT CP.Student_ID, R.Student_Name, CP.certification_date
FROM CLOUD_PRACTITIONER CP
INNER JOIN RESTART R ON CP.Student_ID = R.Student_ID;
```