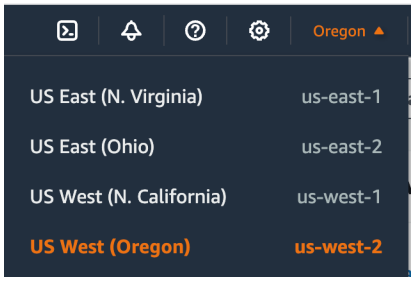
Lab C

Creating a Virtual Network in the Cloud (VPC) with subnets (IP address ranges) and an EC2 with a MySQL RDS database within the VPC

1. Open the Learner Lab.

From our AWS canvas dashboard choose the Learner Lab and open the Modules and then the 'Launch AWS Academy Learner Lab' link. Make sure to click the Start lab link and wait for the AWS link to have the green circle next to it and open the lab.

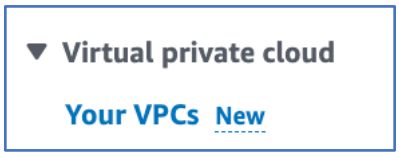
New Region - Oregon



2. Create a Virtual private cloud network

We will set up a virtual network that we have complete control over.

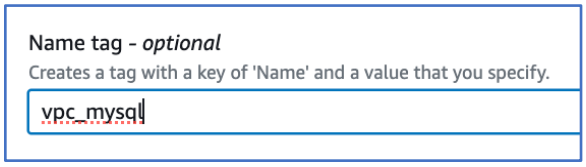
Open the VPC service management console and select 'Your VPCs' in the left side panel.



You are always provided default VPC(s). We will be creating a new one. Click 'Create VPC'.

Leave the defaults of ‘VPC only’,

Give the VPC a Name tag.



Leave the IPv4 CIDER (Classless Inter-domain routing) manual input.

We will be creating a range of 64 IP addresses. So, type in 10.0.0.0/26

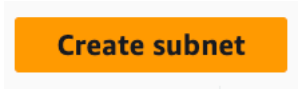
Leave the rest of the defaults and click ‘Create VPC’

Now when you click on ‘Your VPCs’ you will see your new VPC and see that it’s available.

3. Create four subnets (ranges of IP addresses) inside your VPC.

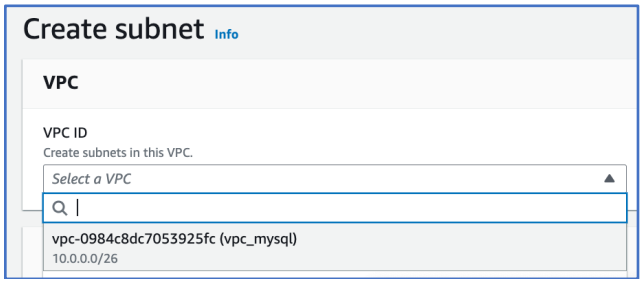
Now click on ‘Subnets’ in the left side panel. You need subnets in your VPC that specify IP address ranges inside AZs to allocate virtual machines and other services.

Notice there are subnets already set up for our default VPC. We need to create subnets in our new VPC. Click ‘Create subnet’

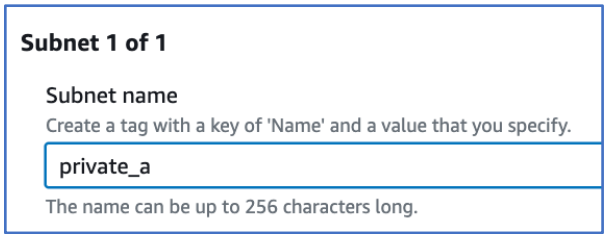


We will set up 4 subnets. 2 public and 2 private. Public subnets have access to the internet whereas private subnets have no access to the internet. Each subnet will get 16 of the 64 IP addresses each.

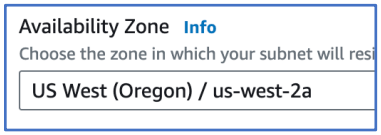
Select the VPC that you will be creating this subnet for. Choose the VPC you just created.



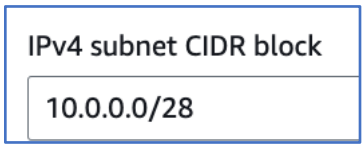
Name this first subnet.



Choose an availability zone.



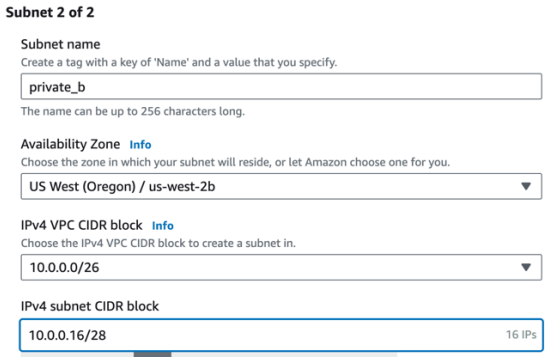
Change the 'IPv4 CIDR block' to 10.0.0.0/28 for the first 16 IP addresses.



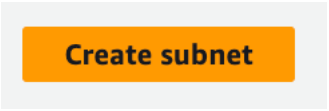
Use the rest of the defaults for the first subnet.

Click the 'Add new subnet' button.

For subnet 2 of 2, give it the name of private_b. Choose a different AZ. Change the 'IPv4 CIDR block' to 10.0.0.16/28 for the next 16 IP addresses.



Leave the rest defaults and click 'Create subnet'.



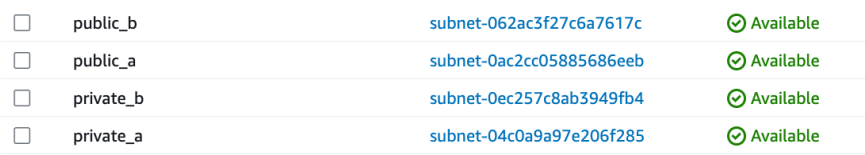
Now we have 2 private subnets in two AZs.

Repeat the process above to create two more subnets. The only difference will be that these 2 subnets will be public.

public_a in us-east-1a with the IP range for the first will be 10.0.0.32/28

public_b in us-east-1b with the IP range for the second will be 10.0.0.48/28

When you click 'Subnets' you should now see, in addition to the default subnets, your 4 new subnets (you might need to clear filters to see them)



Notice under the available IPv4 addresses column it shows 11 in each subnet. That's 11 for each not 16. The last 5 are reserved by AWS for IP networking purposes.

4. Create routing tables

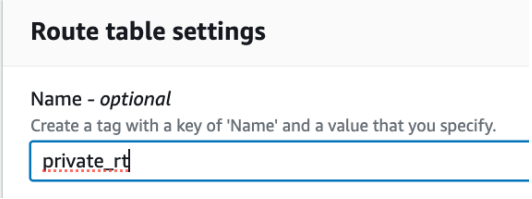
Routing tables determine where network traffic from your subnet or gateway is directed.

Click 'Route tables'

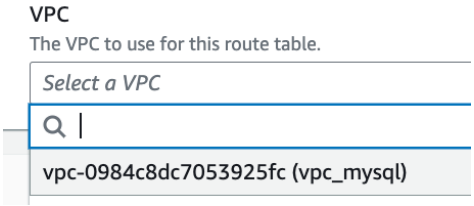
Again, you will see default route tables used with the default VPC. We will create 2 route tables for our VPC, one private and one public. We will associate our subnets into the route tables.

Click 'Create route table'.

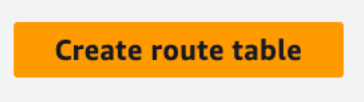
Name the first private route table.



Choose our VPC.



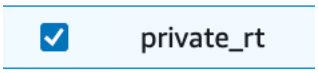
Click 'Create route table'.



Create another public route table the same way, ‘public_rt’ using the same VPC.

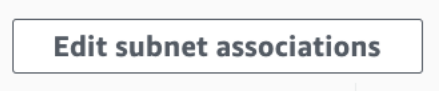
5. Associate your routing tables with the subnets.

Now when you click ‘Route tables’ you will see them listed. Check the private route table.



Click the ‘Subnet associations’ tab below.

Click the ‘Edit subnet associations’.



Check the 2 private subnets to associate with the private routing table.

Click ‘Save associations’

Now check public route table and associate it with the 2 public subnets the same way we did the private ones.

You should now see 2 subnets associated with each route table.

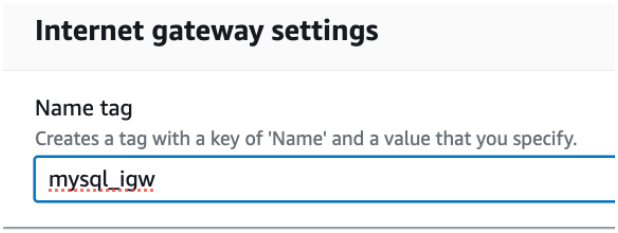
private_rt	rtb-0f2f12d6d1a328ff0	2 subnets
public_rt	rtb-015052355dac3ea79	2 subnets

6. Create internet traffic to our public subnet with an Internet Gateway

Click ‘Internet gateways’.

There are already default gateways. We will create our own. Click ‘Create internet gateway’

Name your gateway.

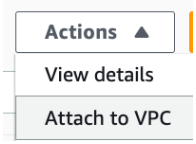


Leave other defaults. Click ‘Create internet gateway’.

7. Attach the internet gateway to our VPC.

When you click ‘Internet gateways’ again, you will see the new gateway but it will have the state of ‘Detached’. We need to attach it to our VPC.

Check the box next to the new internet gateway and under ‘Actions’ click ‘Attach to VPC’.

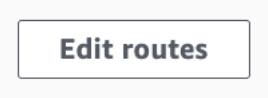


Select our VPC and then click ‘Attach internet gateway’.

8. Add a route to the public routing table to route internet traffic to the internet gateway.

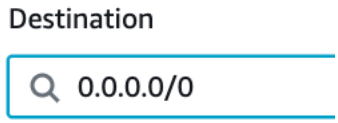
Click the ‘Route tables’ again and check the box next to the public route table. Then choose the ‘Routes’ tab.

Click ‘Edit routes’

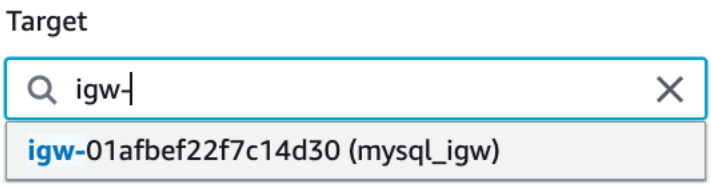


There is a default local destination. We want to add another. Click ‘Add route’.

Add destination of 0.0.0.0/0 to allow all traffic.



And Target ‘Internet Gateway’. Then choose the internet gateway you created earlier.



Click ‘Save changes’

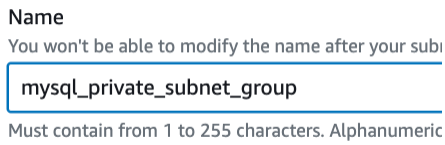
9. Create a MySQL database instance

Find the RDS service.

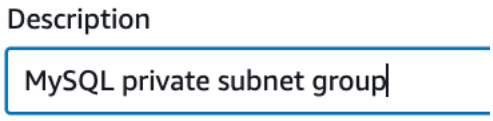
Before we create a database. We need to create a subnet group within RDS. Click ‘Subnet groups’ on the left.

Click ‘Create DB subnet group’

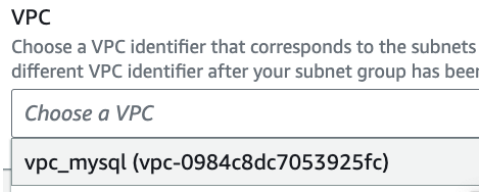
Give it a name.



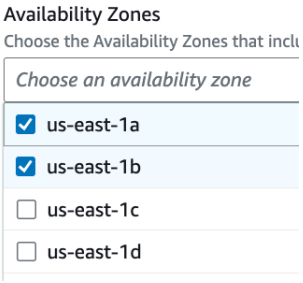
Add a description



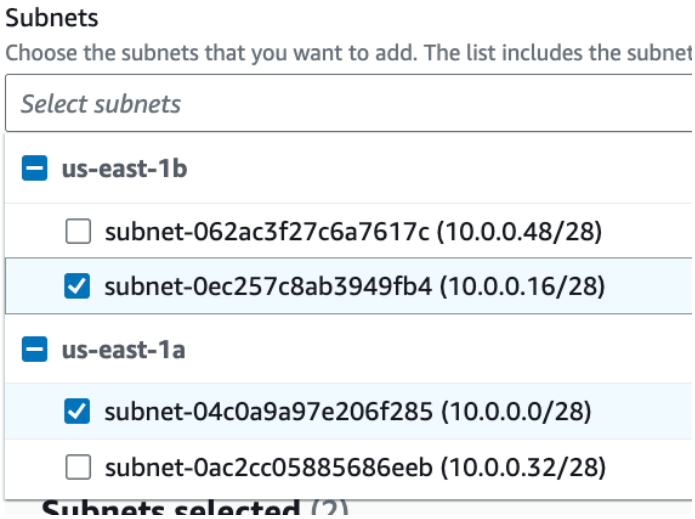
Choose our VPC



Choose 2 AZs by unchecking all but two.



Select a subnet from each AZ.

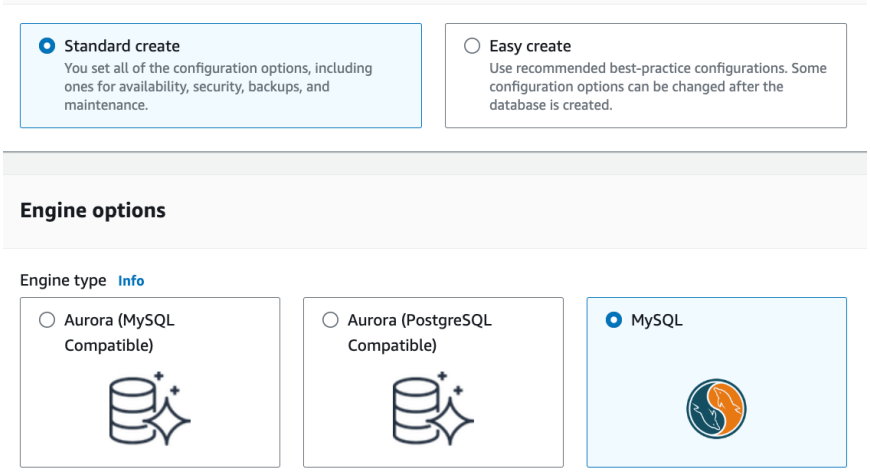


Then click ‘Create’

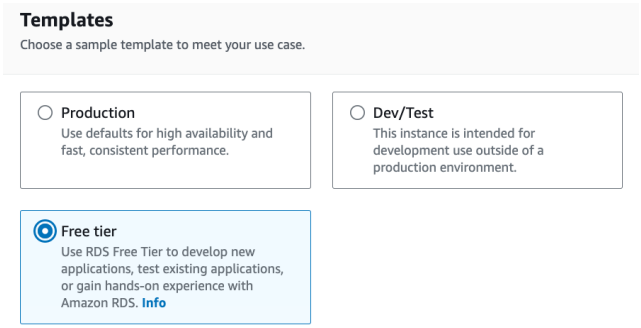
Now we will create the database.

Click ‘Databases’ and ‘Create database’

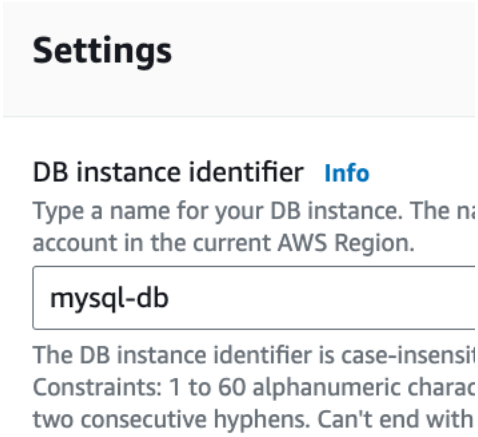
We will choose the ‘Standard create’ and the engine we will use is MySQL.



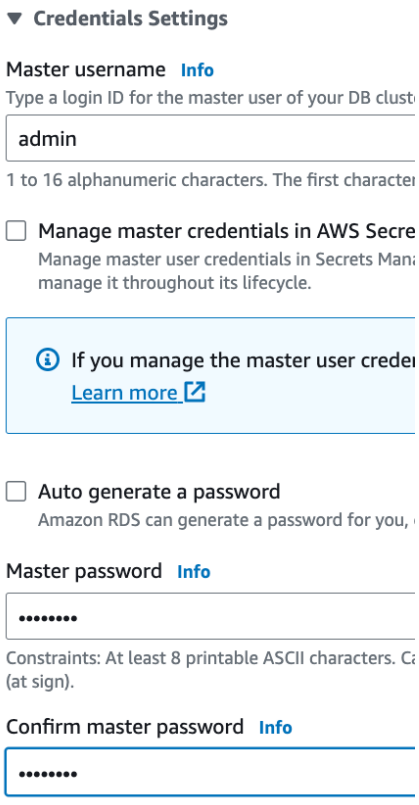
Choose ‘Free tier’ under Templates.



In the ‘Settings’ section. Give it a name of mysql-db and use a dash not an underscore here.

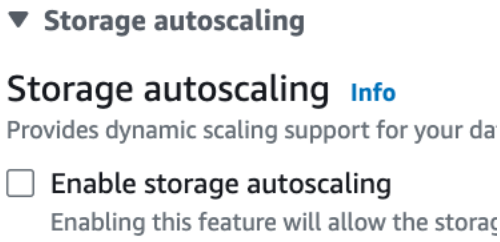


Add login information. I used ‘password’ as the password for this lab and left the admin username.

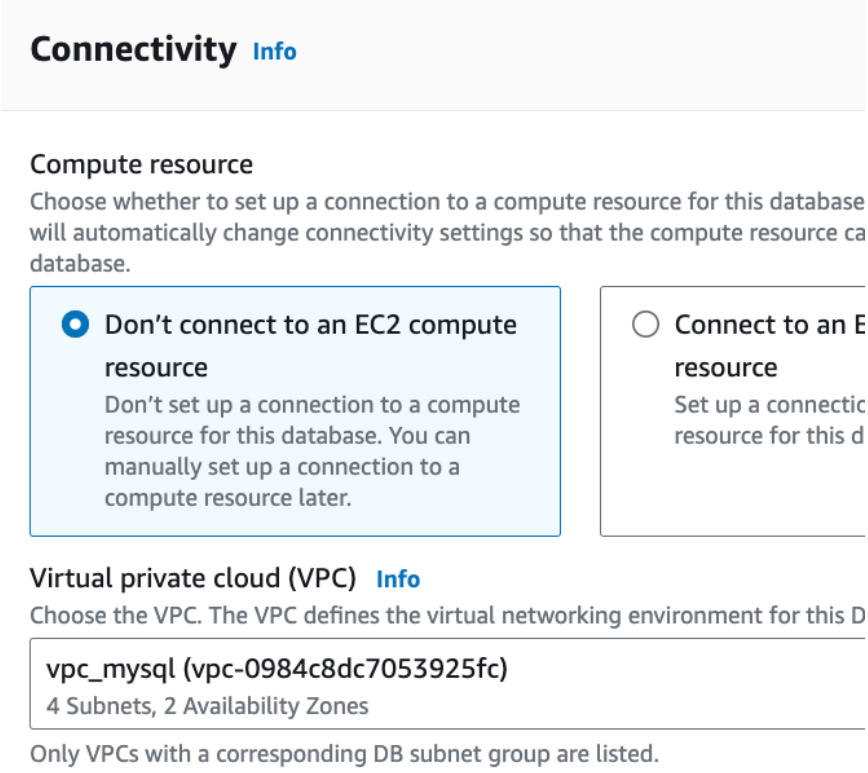


Choose the db.t2.micro.

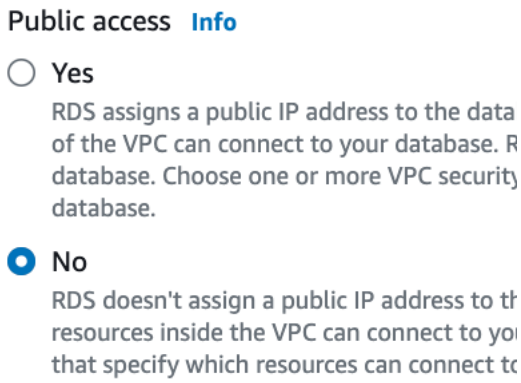
Leave other defaults until ‘Storage autoscaling’ open that section up and uncheck ‘Enable storage autoscaling’



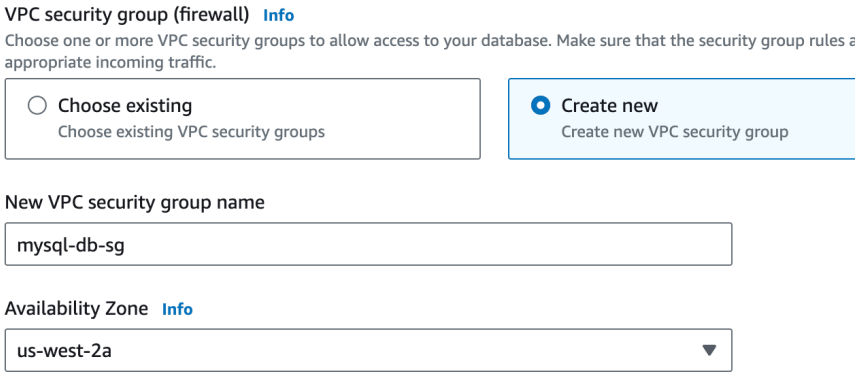
In the ‘Connectivity’ section choose our VPC.



We don’t want Public access, so leave it No.

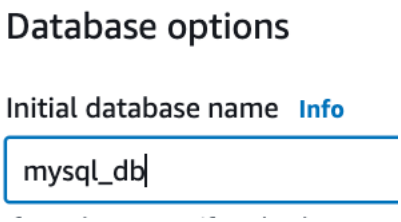


Create a new security group and name it. This is a firewall for our database. You use security groups to control traffic at the instance level and NACL (Network access control lists) to control access at the subnet level. Choose the AZ

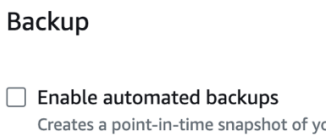


Leave the password authentication.

Under ‘Additional configuration’ give the database a name.



Uncheck enable automated backups



Finally, click ‘Create database’.

If it suggests add-ons, just close that window.

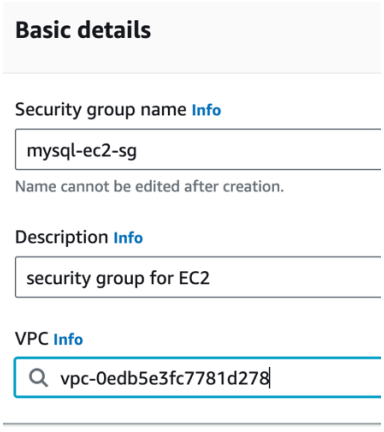
Wait for the database to set up. Do step 10 while you wait.

10. Set up a security group for our EC2.

Go to the EC2 service and click ‘Security groups’ on left side (scroll down a bit)

Click ‘Create security group’

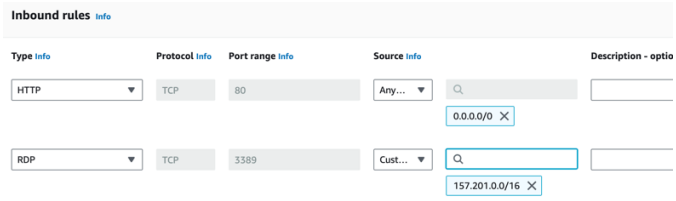
Give it a name, description, and choose our VPC. You might have to delete the default VPC to see our VPC option.



Click ‘Add rule’ under Inbound rules.

Add two rules as follows:

Should be:
Type info – http
Protocol info -leave tcp
Port range leave 80
Source info – custom 0.0.0.0/0
Next rule:
Type info – rdp
Protocol info -leave tcp
Port – leave default
Source info -custom BYUI 157.201.0.0/16

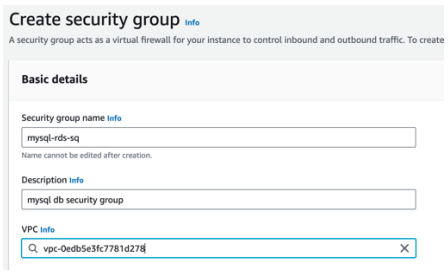


Click ‘Create security group’

11. Edit the inbound rules for the database

Go back to RDS and check that your database finished.
Click on the database name
Scroll down and click on inbound security groups

Create a new security group

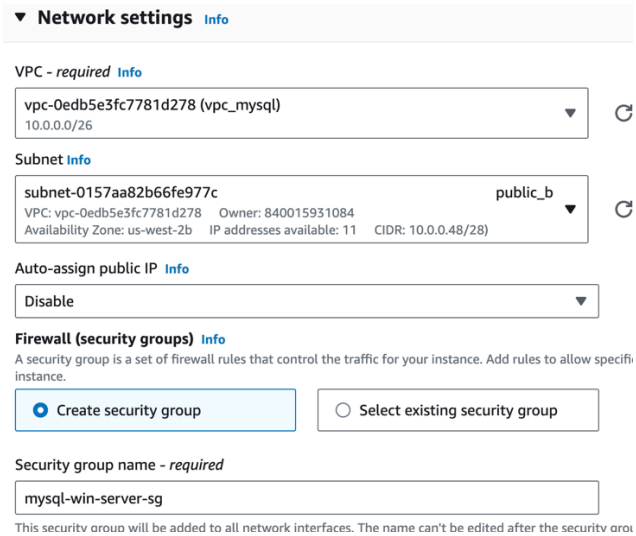


Add and inbound rule

Type info – mysql/aurora
Protocol – tcp - leave
Port range – leave - I had 3306
Source info – custom
Choose the security group you had for your ec2 security group (you might have to close out the default that is there)

Click ‘Create security group’

Create a windows server again like lab B but
Use t3a.large (step 5 in lab b)
and
Edit ‘Network Settings’



Launch the instance. Choose the ‘Connect an RDS database button’

Your instance might take a few minutes to run and have 2/2 checks before you can connect with the next window.

Go to the list of instances. Check the box next to the new instance and connect. Use the RDP tab and download remote desktop file. Double click the .rdp file that downloaded.

Use the ‘Get password’ to get the remote desktop password.