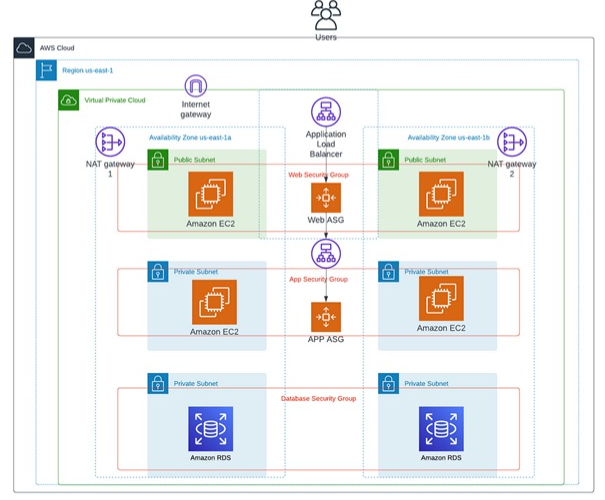
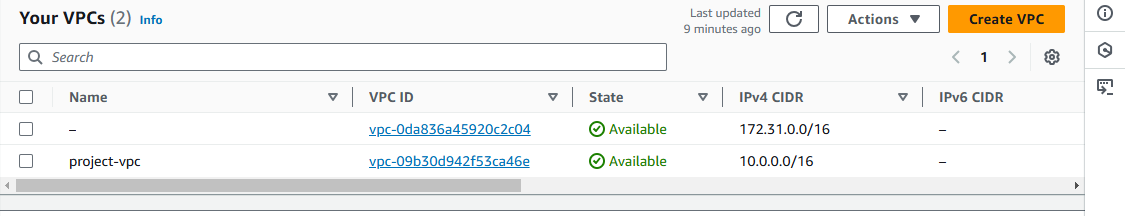
****

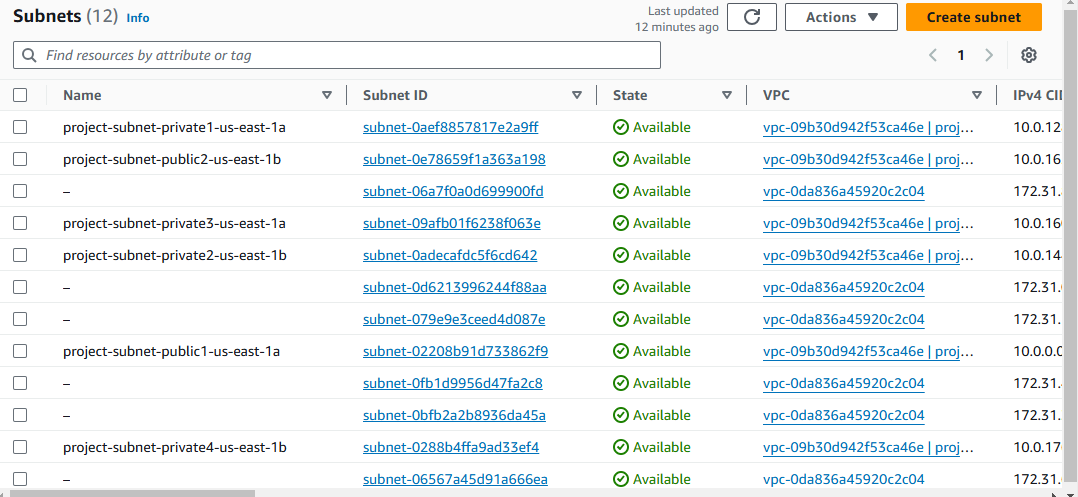
**Project Document**

# Three-tier-architecture :

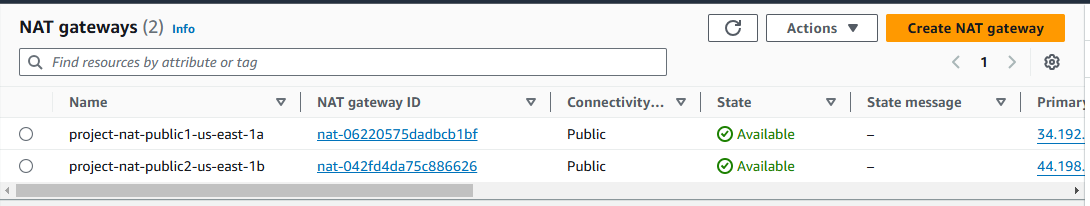
* **Create a vpc in OHIO REGION**
* In the VPC console, let’s create a new VPC. We’ll select the ‘VPC and more’ option and name our project ‘**project-vpc**’ with a CIDR block of **10.0.0.0/16**

****

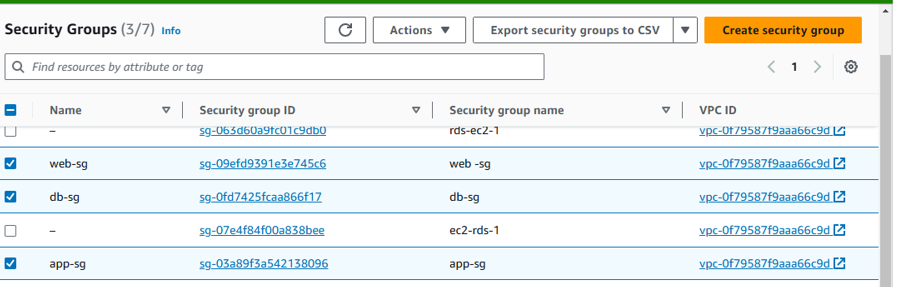
* **Create two public subnets and 4 private subnets**
  + - * One public subnet and two private subnets in one available zone (us-east-2a)
      * One public subnet and two private subnets in one available zone (us-east-2b)

****

* **Now created a subnets as per project requirement shown in above figure**
* **Create two natgateway in two available zones**

****

* **Create 3 security groups for web-tier,app-tier and database tier**
* **Create web-sg with ssh,http**
* **Create app-sg with ssh,http add web-sg in cidr source**
* **Create database-sg with mysql-aurora add app-sg in cidr source**

****

**1. Create a web server launch template**

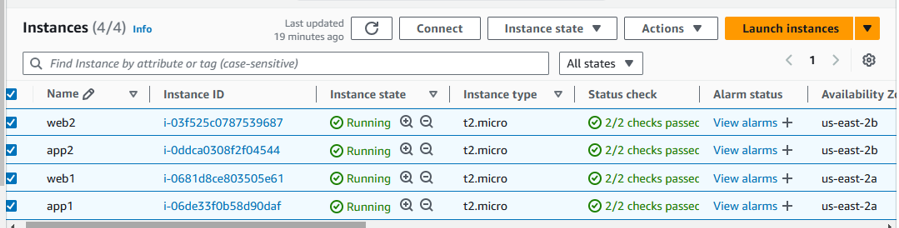
It’s time to create a template that will be used by our ASG to dynamically launch EC2 instances in our public subnets.

In the EC2 console, navigate to ‘*Launch templates*’ under the ‘*Instances*’ sidebar menu. We’re going to create a new template called ‘**Web-template**’ with the following provisions:

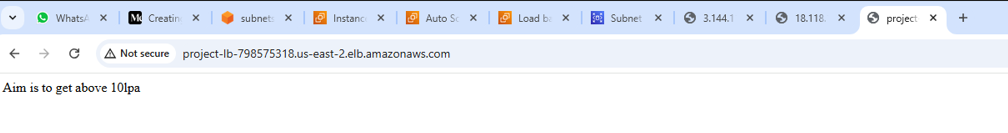
1. AMI: ubuntu
2. Instance type: t2.micro (1GB – Free Tier)
3. A new or existing key pair
4. Network settings: -> Select security groups -> EC2 Webserver Security Group
5. Create launch template

Time to create an Auto Scaling group. Navigate to EC2 -> Auto Scaling groups -> Create Auto Scaling group

* Name: three-tier-asg
* Launch template: web-template
* Network: VPC (project-vpc) & add both public subnets
* Group size: I added 2 for the desired capacity
* Click Next -> Review -> Create Auto Scaling Group
* Now we will get public servers based on our desired capacity in auto sclaing group



* Now Im going to connect gitbash using webser1 and webser2 public ip’s
* Installed apache2 in both servers
* Create target group for application load balancer
* We’ll need an ALB to distribute incoming HTTP traffic to the proper targets (our EC2s). The ALB will be named, ‘**project-alb.’**We want this ALB to be ‘Internet-facing,’ so it can listen for HTTP/S requests.
* Need to copy dns link to check whether out ALB is working or not..Alb is working it shows in below figure



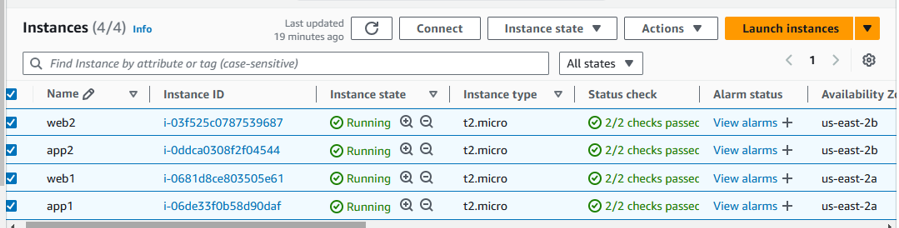
**1. Create a app server launch template**

It’s time to create a template that will be used by our ASG to dynamically launch EC2 instances in our private subnets.In the EC2 console, navigate to ‘*Launch templates*’ under the ‘*Instances*’ sidebar menu. We’re going to create a new template called ‘**app-template**’ with the following provisions:

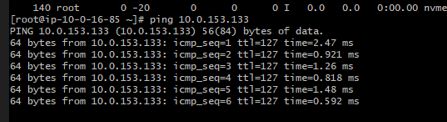
1. AMI: Amazon 2 Linux
2. Instance type: t2.micro (1GB – Free Tier)
3. A new or existing key pair
4. Network settings: -> Select security groups -> EC2 Webserver Security Group
5. Create launch template

Time to create an Auto Scaling group. Navigate to EC2 -> Auto Scaling groups -> Create Auto Scaling group

* Name: three-tier-asg
* Launch template: app-template
* Network: VPC (project-vpc) & add both public subnets
* Group size: I added 2 for the desired capacity
* Click Next -> Review -> Create Auto Scaling Group
* Now we will get private servers based on our desired capacity in auto sclaing group



* Now we need to connect our private server in public server
* To check that use ping private server private IP

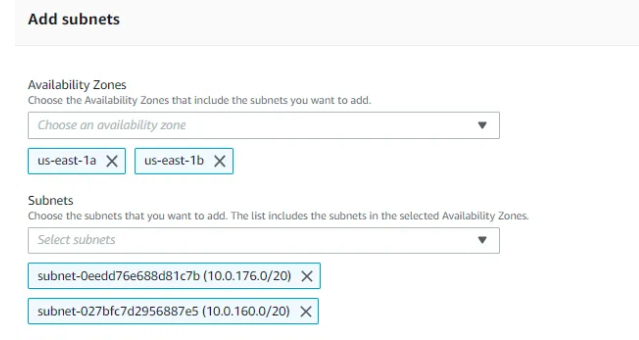


* Successfully established connection between web-tier and app-tier

# Database Tier

Are you ready to dive into the exciting world of database tier design? Buckle up and get ready to create a database tier that will blow your mind!

* Create subnet group
* Navigate to RDS -> Subnet groups -> Create DB subnet group
* Here we need select private 3 and private 4 subnets for subnet group



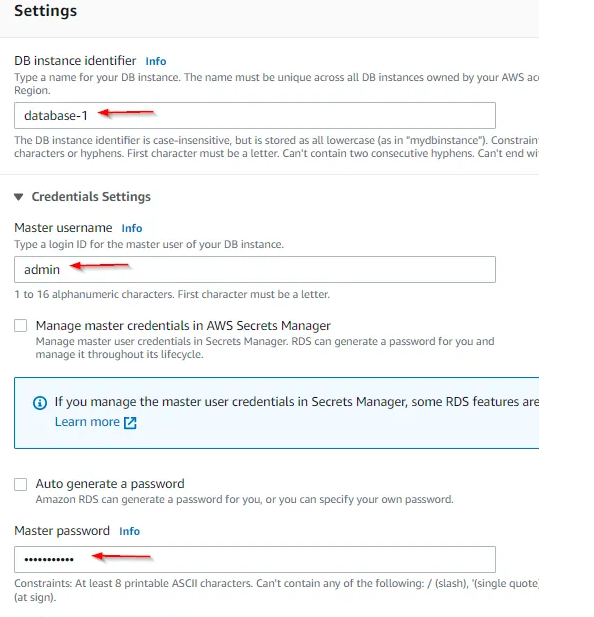
* Once subnet group is created we need to create rds database
* Navigate to Databases -> Create database

**1.Database creation method:**Standard create

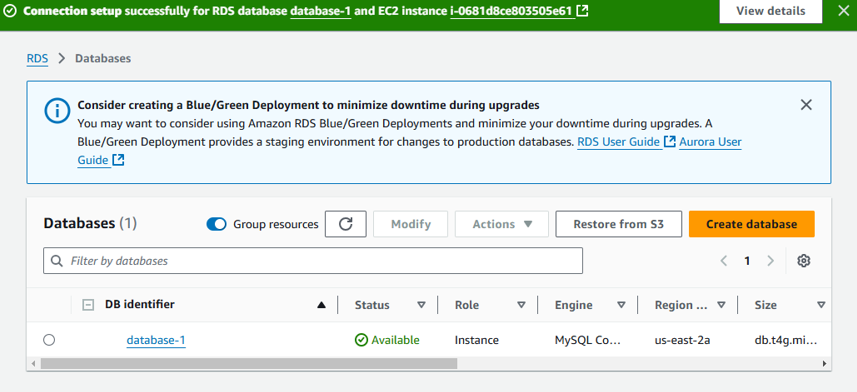
**2.Engine options:**MySql

**3.Templates:** Free tier

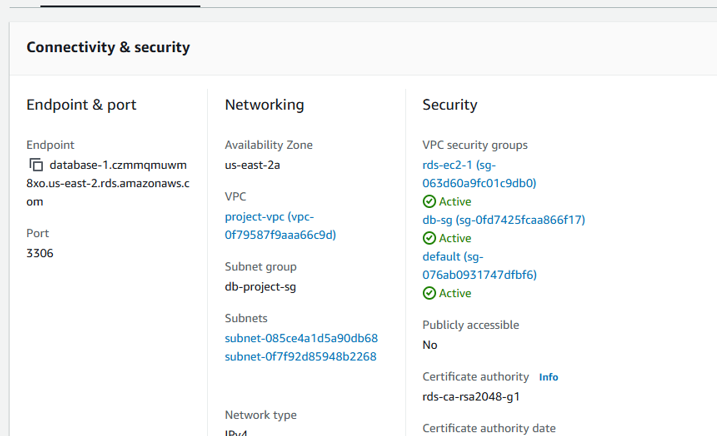
**4.Settings:**



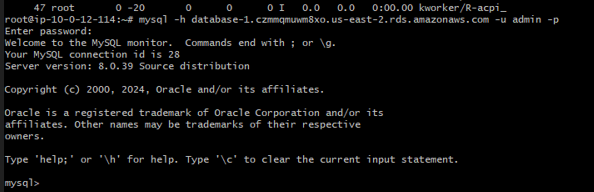
* **Instance configuration:** db.t2.micro
* **Storage**: Default
* **Connectivity**: Add VPC (project-vpc)
* **DB subnet group:**project-SG (created subnet group)
* **Public access:** No
* **VPC security group (firewall):** Create new
* **New VPC security group name:** RDS-securitygroup
* **Database authentication:** Password authentication
* Create database



* Now we need to copy the db endpoint to connect with public server (web-tier)



* Now im gpint to connect database with web-tier using endpoint



* Successfully our database-tier is connected with web-tier