**1) Create VPC with 2 priavate and 2 public subnets.**









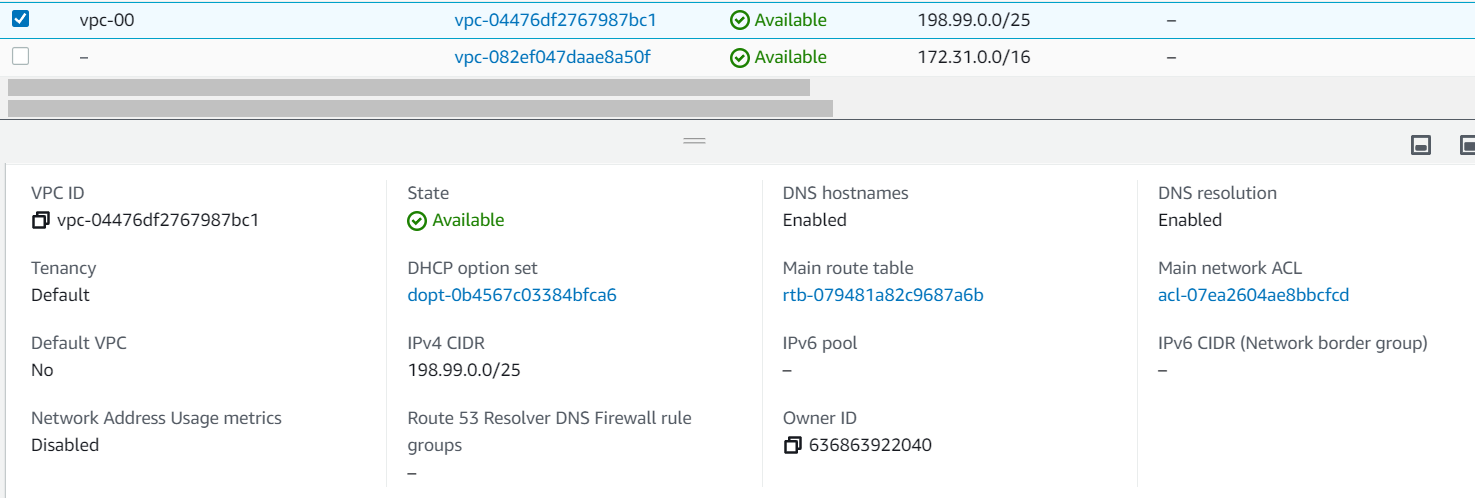
**2) Enable DNS Hostname in VPC**

During VPC Creation:

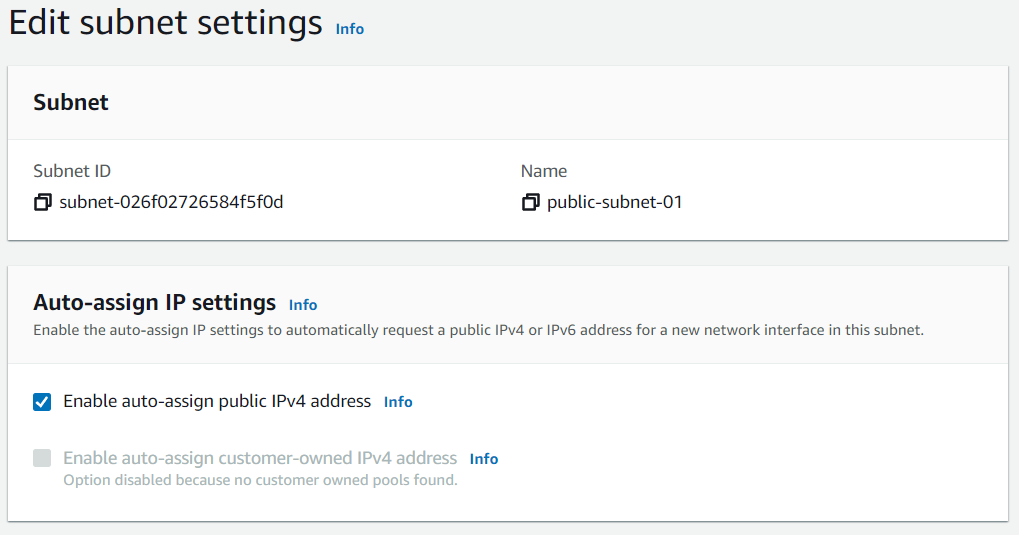
* When creating a new VPC, in the Create VPC wizard, there is an option to enable DNS support and DNS hostnames.
  + DNS Hostnames: Set this to Yes to enable DNS hostnames.
  + DNS Support: Set this to Yes to enable DNS resolution.

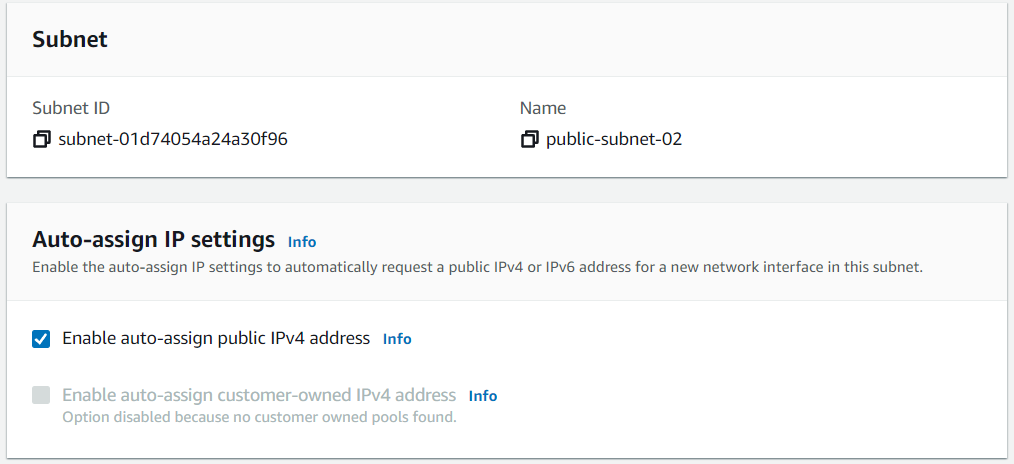
After VPC Creation:

1. Go to the VPC Dashboard.
2. In the left-hand sidebar, click on Your VPCs.
3. Select the VPC for which you want to enable DNS hostnames.
4. In the Details section, click on the Actions button and choose Edit DNS Hostnames.
5. Toggle the Enable DNS Hostnames setting to Yes.
6. Click Save.

****

**3)** **Enable Auto Assign Public ip in 2 public subnets**

****

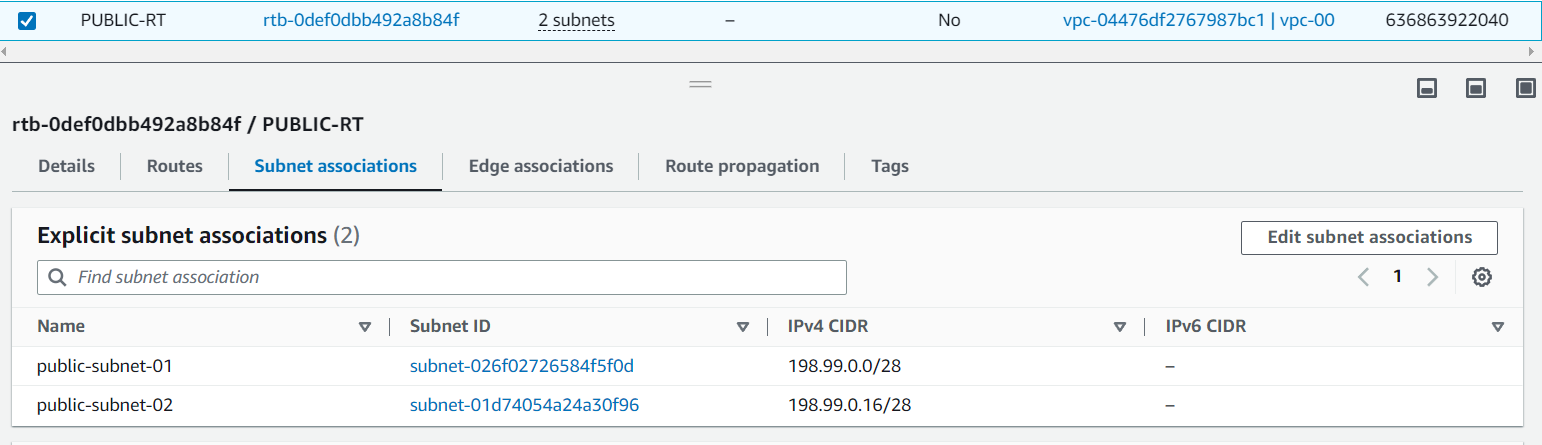
****

**4)** **Add 2 PUBLIC subnets in public route table**

### Steps to Add 2 Public Subnets to the Public Route Table

#### 1. Using the AWS Management Console

1. Navigate to the Route Tables:
   * In the VPC Dashboard on the AWS Console, click on Route Tables on the left-hand side.
2. Identify the Public Route Table:
   * Find the route table that is associated with your public subnets (usually the route table that has a route to the Internet Gateway).
   * If you don't already have a route table set up, create a new one with a route to the Internet Gateway:
     + Click on Create Route Table, name it (e.g., Public-Route-Table), and associate it with your VPC.
     + In the Routes tab, click Edit Routes, then click Add Route and enter 0.0.0.0/0 as the destination and select your Internet Gateway as the target.
3. Add Subnets to the Public Route Table:
   * Once you've identified or created the correct route table, click on the Subnet Associations tab.
   * Click on Edit subnet associations.
   * Select the two public subnets (e.g., Public-Subnet-A and Public-Subnet-B).
   * Click Save to associate these subnets with the public route table.

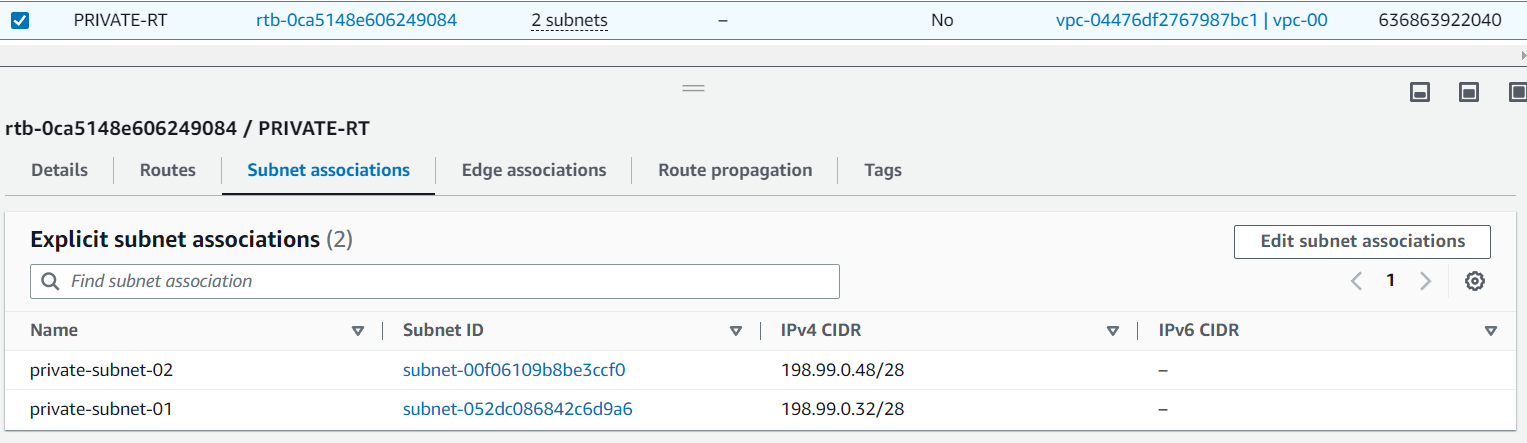
****

**5) Add 2 private subnets in private route table**

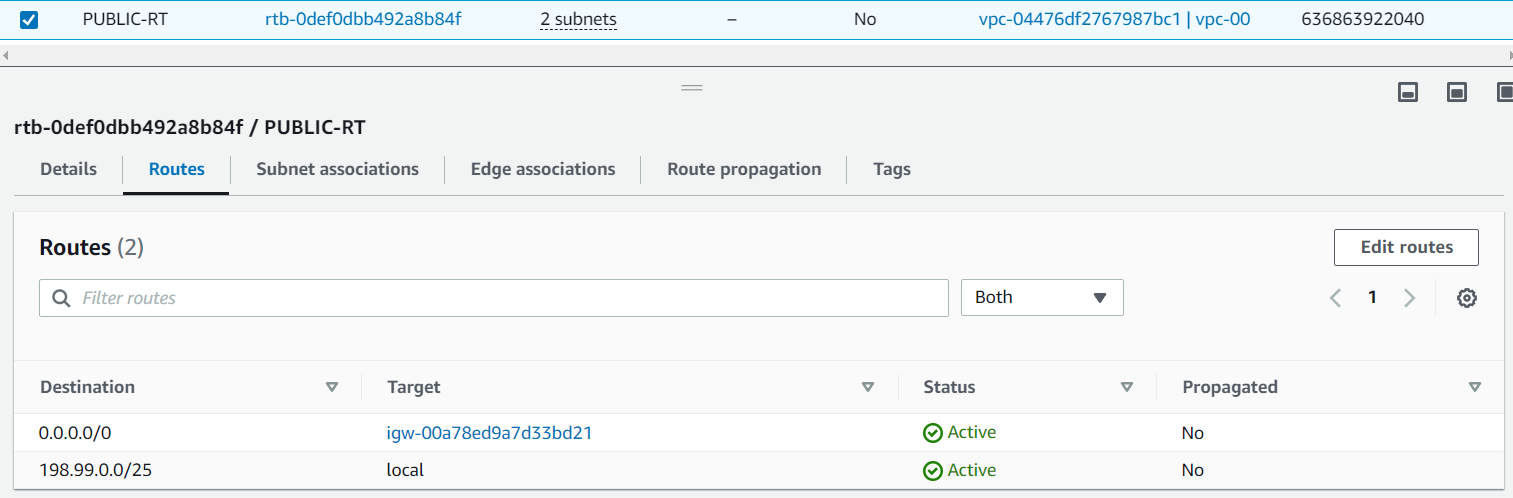
1. Using the AWS Management Console

1. Navigate to the Route Tables:
   * Go to the VPC Dashboard on the AWS Management Console.
   * In the left-hand menu, click on Route Tables.
2. Identify the Private Route Table:
   * If you already have a private route table (a route table without an Internet Gateway route), find it in the list of route tables.
   * If you don't have one yet, create a new route table for your private subnets.
3. Add the Subnets to the Private Route Table:
   * Select the private route table (usually one that does not have a route to the Internet Gateway).
   * Go to the Subnet Associations tab.
   * Click Edit subnet associations.
   * Select your two private subnets (e.g., Private-Subnet-A and Private-Subnet-B).
   * Click Save to associate these subnets with the private route table.
4. Configure Routes (if needed):
   * If you plan to use a NAT Gateway for internet access from private subnets, make sure that the private route table has a route like 0.0.0.0/0 that points to the NAT Gateway (or NAT instance).
   * If you don't need outbound internet access from the private subnets, no further routes are needed.

NAT == > **Network Address Translation (NAT)**



**6) Public route table will have the routes to internet and local**

****

**7)** **Create Ec2 in public subnet with t2micro and install php**

### Steps to Launch EC2 in a Public Subnet and Install PHP

#### Prerequisites:

* You already have a VPC with public subnets and Internet Gateway configured.
* You should have a key pair to SSH into the EC2 instance.
* Security group allowing HTTP (port 80) and SSH (port 22) traffic.

### 1. Launch EC2 Instance in the Public Subnet

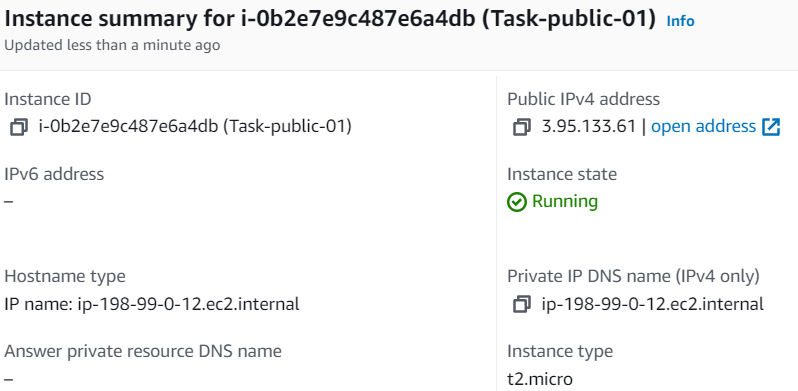
#### Using the AWS Management Console:

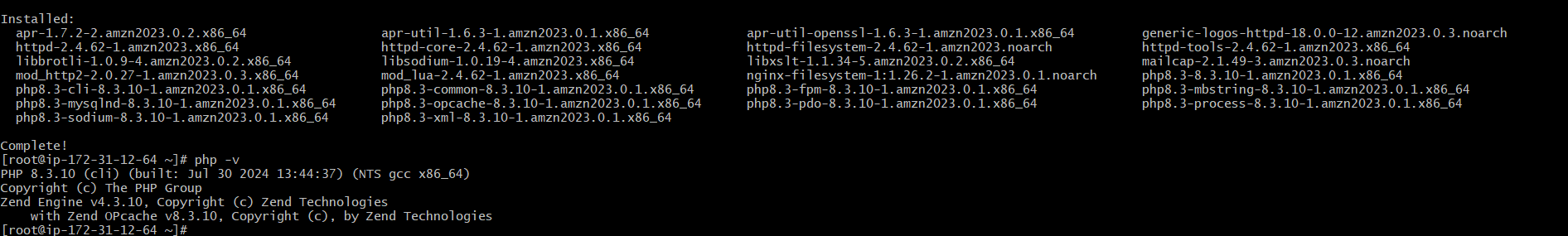
1. Go to EC2 Dashboard:
   * In the AWS Management Console, navigate to the EC2 service.
2. Launch an Instance:
   * Click on the Launch Instance button.
3. Select Amazon Machine Image (AMI):
   * For a simple setup, you can choose the Amazon Linux 2 AMI or an Ubuntu AMI (depending on your preference for PHP installation). The Amazon Linux 2 AMI comes with PHP pre-installed in some cases, but you can install or update it later.
4. Choose an Instance Type:
   * Select t2.micro (eligible for the AWS Free Tier).
5. Configure Instance Details:
   * Under Network, choose the VPC where your public subnet is located.
   * Under Subnet, select one of your public subnets.
   * Enable Auto-assign Public IP to ensure the instance gets a public IP (for internet access).
6. Add Storage:
   * The default 8 GB of General Purpose SSD should be sufficient for a basic PHP setup. You can increase the size if needed.
7. Configure Security Group:
   * Select or create a new Security Group. Make sure it has the following inbound rules:
     + SSH (port 22): Allow from your IP address (for SSH access).
     + HTTP (port 80): Allow from 0.0.0.0/0 (for web traffic).

sudo amazon-linux-extras enable php7.4

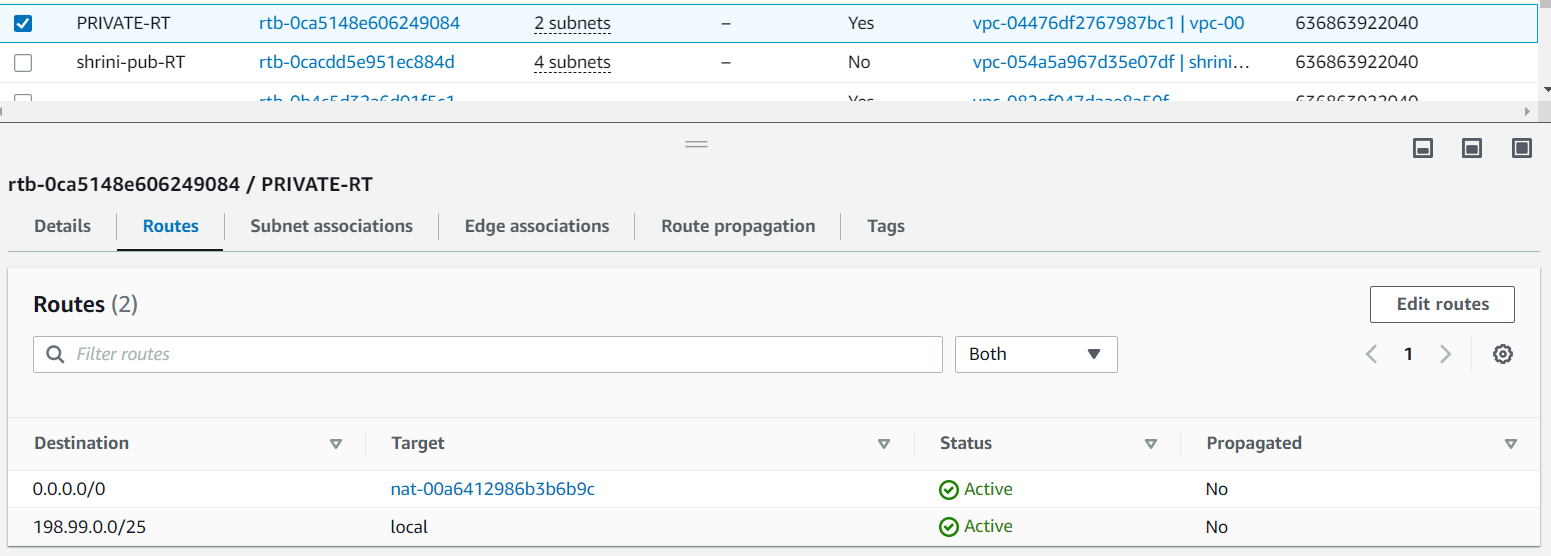
sudo yum install -y php php-cli php-fpm php-mysqlnd php-pdo php-mbstring

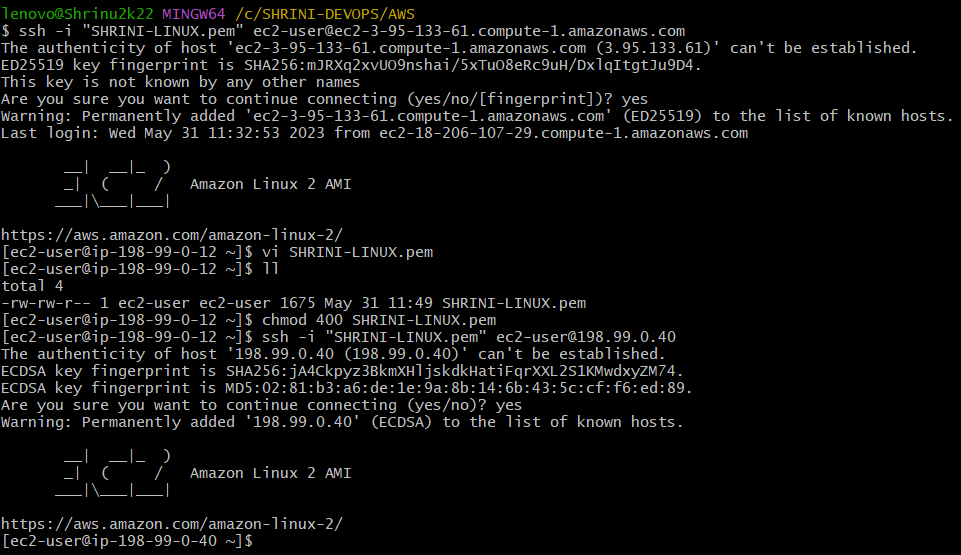
php -v

****

****

**8) Configure Nat gateway in public subnet and connect to private Instance**

****

****

**9. Install Apache Tomcat in private ec2 and deploy a sample app.**

