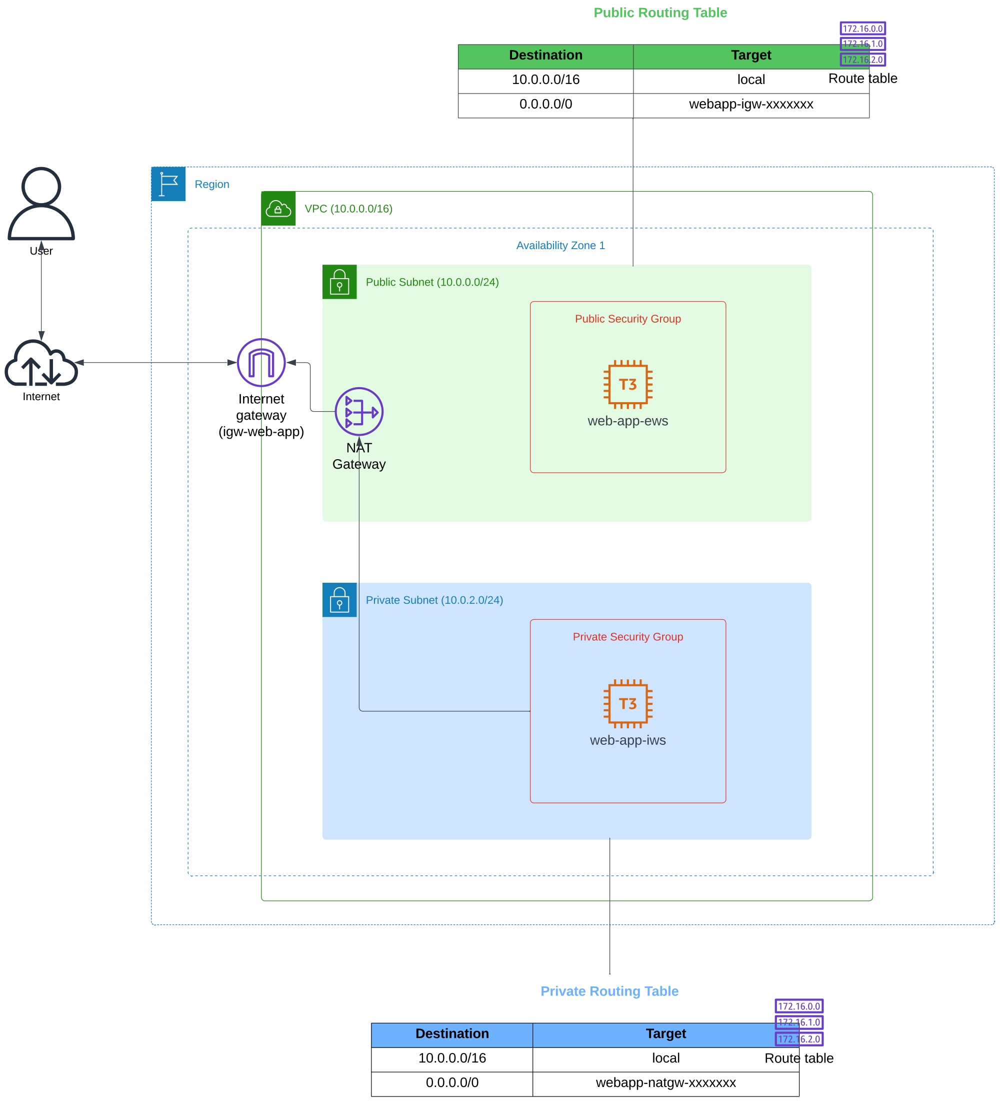

Public Routing Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	webapp-igw-xxxxxxx

Private Routing Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	webapp-natgw-xxxxxxx

Steps : - AWS Infrastructure Setup for 2-Tier Application

● Select Region for infrastructure setup. Add **tags** to everything.

● Create an VPC in Region named **webapp-vpc (10.0.0.0/16)**. From the same page, select **Actions** and **Edit DSN hostnames** and **enable checkbox. Save.** It will assign friendly DNS name.

● Create **Public** subnet named **webapp-pub-sn(10.0.0.0/24).** Configure subnet to automatically assign a public IP for all instances launched under it.

● Create Private subnet named **webapp-pri-sn(10.0.2.0/23)**. It's twice as large as the public subnet as all resources mostly kept private.

● Create an internet gateway named **webapp-igw.** And attache it to the VPC (**webapp-vpc)**.

● Route internet trafic in the public subnet to the internet gateway (by creating the routing table). Create a Public Route Table and add **destincation** 0.0.0.0/0 (means all) traffic to **target** (**webapp-igw-xxxxxx**). And **edit** subnet association to choose **public subnet**. This subnet is now public because it has internet access via internet gateway.

○ Create a Public Security Group (**webapp-pub-sg**) that allows incoming traffic to public instance. Add **Inbound** rule of **type HTTP** to **source Anywhere IPV4**.

○ Launch an EC2 **T3.micro** instance (**webapp-ews**) into Public Subnet and auto assign public IP, select existing SG (**webapp-pub-sg**), add **user data** of (**webapp-ews**) to launch the app and review, tag and launch instance.

○ Connect to public instance via HTTP (Choose Public IPV4 DNS name in browser).

○ Connect to EC2 instance (**webapp-ews**) in the Public Subnet via SessionManager (From EC2 Dashboard, select public instance checkbox, then Connect button, and from SessionManager tab, choose Connect buttton).
On **terminal, run the following commands**:
cd ~
curl -I https://aws.amazon.com/training/
you should get 200 response.

○ Createt a NAT Gateway and configure routing in the private subnet. From the VPC Dashboard, select NAT Gateways. Choose **Create NAT Gateway**. Enter name (**webapp-natgw**), then choose **Subnet** (Public Subnet) and then Choose **Allocate Elastic IP** and Choose **Create NAT Gateway**.

- Now Create a new Routing Table for private subnet that redirects non-local traffic to the NAT Gateway.
- Choose **Create Route Table** from left navigation pane, enter name **Private Route Table** (**webapp-pri-rt**), choose VPC (vpc-webapp) and press **Create Route Table**. The private route table displays.
- Choose **Routes** tab. Add a route to send intenet-bound traffic through the NAT Gateway. Choose **Add Route** and enter **Destination** as **0.0.0.0/0**, and **Target** as **NAT Getway** (**webapp-natgw-xxxxxx**) from the drop down and save changes.
- Choose **Subet Association** tab, choose **Edit Subnet Associations**, select **Private Subnet (webapp-priv-sn)** checkbox and save associations.

○ Create a Security Group for private resources. In the left nativation pane, select **Security Groups**, Choose **Create Security Group**, enter **Name** (webapp-priv-sg), description: **Allows incoming traffic to private instances using Private Security Group**. then select **VPC** (**webapp-vpc**). In the **Inbound Rule** section, add **Rule**, and add configuration
Type: **HTTPS**, Source Type: **Custom**, and Source: **Public Subnet** (**webapp-pub-sn**), then add tags and create rule.

○ Launch an EC2 (**Amazon Linux EMI**) **T3.micro** instance (**webapp-iws**) under VPC (**webapp-vpc**), subnet: Private Subnet (**webapp-priv-sn**), **Auto Assign Public IP: DISABLE** without key/pair, and select existing SG (**webapp-priv-sg**), select **IAM Instance profile:EC2InstanceProfile** role add **user data** of (**webapp-ews**) to launch the app and review, tag and launch instance (check status: **Running**).

○ Connect to EC2 instance (webapp-iws) in the Private Subnet via SessionManager (From EC2 Dashboard, select private instance checkbox, then Connect button, and from SessionManager tab, choose Connect buttton).
On **terminal, run the following commands**:
cd ~
curl -I https://aws.amazon.com/training/
you should get 200 response.

○ Test connectivity from public instance (webapp-ews) to private instance (webapp-iws) using:
ping <private-instance-IPv4-address>
Stop ping after few seconds with CTRL+C. The ping must fail.
To succeed, add **Inbound Rule** in Private Subnet (webapp-priv-sn). Choose EC2 -> Security Groups -> Private Subnet -> Edit Inbound Rule -> Add Rule -> **Type**: Custom ICMP -IPV4, **Source**: Public Security Group (webapp-pub-sg), save changes and try ping again from public instance.

○ Reterive instance metadata with local-link-address of instance
curl http://169.254.169.254/meta-data/