## SkillXchange – three tire project

**Step-by-Step Guide to Create VPC and Route Tables for SkillXchange**

**Step 1: Log into AWS Console**

1. Visit the [AWS Management Console](https://aws.amazon.com/console/).
2. Log in using your AWS account credentials.

**Step 2: Select the Mumbai Region**

1. In the **top-right corner** of the console, click the region dropdown.
2. Select **Asia Pacific (Mumbai)** or **ap-south-1**.

**Step 3: Navigate to VPC Dashboard**

1. In the search bar at the top of the AWS console, type **VPC**.
2. Click on **VPC** to open the **VPC Dashboard**.

**Step 4: Create a VPC (VPC Only Option)**

1. On the left sidebar of the VPC dashboard, click **Create VPC**.
2. Select **VPC only** to create a simple VPC setup.
3. Complete the following details:
   * **Name tag**: skillxchnage-vpc
   * **IPv4 CIDR block**: 10.0.0.0/16 (allows 65,536 private IPs).
   * **IPv6 CIDR block**: Choose **No IPv6 CIDR block** (unless you want IPv6 support).
   * **Tenancy**: Choose **Default** (unless dedicated hardware is needed).
4. Click **Create** to create the VPC.

**Step 5: Create Subnets (2 Public and 4 Private)**

You will now create **2 public subnets** and **4 private subnets**. These subnets will be spread across **2 Availability Zones**: **ap-south-1a** and **ap-south-1b**.

**Create Public Subnets (2 Subnets):**

1. Go to the **Subnets** section in the VPC dashboard.
2. Click **Create subnet**.
   * **Public Subnet 1**:
     + **Name**: skillxchnage-web-sb-ap-south-1a
     + **VPC**: Select skillxchnage-vpc.
     + **Availability Zone**: Select **ap-south-1a**.
     + **IPv4 CIDR block**: 10.0.1.0/24.
   * **Public Subnet 2**:
     + **Name**: skillxchnage-web-sb-ap-south-1b
     + **VPC**: Select skillxchnage-vpc.
     + **Availability Zone**: Select **ap-south-1b**.
     + **IPv4 CIDR block**: 10.0.2.0/24.
3. Click **Create** to create the subnets.

**Create Private Subnets (4 Subnets):**

1. For each private subnet, click **Create subnet** again.
   * **Private Subnet for App 1**:
     + **Name**: skillxchnage-app-sb-ap-south-1a
     + **VPC**: Select skillxchnage-vpc.
     + **Availability Zone**: Select **ap-south-1a**.
     + **IPv4 CIDR block**: 10.0.3.0/24.
   * **Private Subnet for App 2**:
     + **Name**: skillxchnage-app-sb-ap-south-1b
     + **VPC**: Select skillxchnage-vpc.
     + **Availability Zone**: Select **ap-south-1b**.
     + **IPv4 CIDR block**: 10.0.4.0/24.
   * **Private Subnet for DB 1**:
     + **Name**: skillxchnage-db-sb-ap-south-1a
     + **VPC**: Select skillxchnage-vpc.
     + **Availability Zone**: Select **ap-south-1a**.
     + **IPv4 CIDR block**: 10.0.5.0/24.
   * **Private Subnet for DB 2**:
     + **Name**: skillxchnage-db-sb-ap-south-1b
     + **VPC**: Select skillxchnage-vpc.
     + **Availability Zone**: Select **ap-south-1b**.
     + **IPv4 CIDR block**: 10.0.6.0/24.
2. Click **Create** to create the private subnets.

**Step 6: Create and Attach an Internet Gateway (for Public Access)**

1. In the VPC dashboard, go to the **Internet Gateways** section.
2. Click **Create internet gateway**.
3. Name it skillxchnage-IGW, then click **Create**.
4. After the internet gateway is created, select it and click **Actions** > **Attach to VPC**.
5. Choose the VPC you created (skillxchnage-vpc) and click **Attach**.

**Step 7: Create and Configure Route Tables**

**Create Route Table for Public Subnets (skillxchnage-pb-rt):**

1. In the **Route Tables** section of the VPC dashboard, click **Create route table**.
2. **Name**: skillxchnage-pb-rt (Public Route Table).
3. **VPC**: Select your VPC (skillxchnage-vpc).
4. Click **Create** to create the route table.
   * **Edit Routes**:
     1. Select the **Public Route Table** (skillxchnage-pb-rt).
     2. Click **Edit routes** > **Add route**.
     3. **Destination**: 0.0.0.0/0 (for all internet-bound traffic).
     4. **Target**: Select the **Internet Gateway** (skillxchnage-IGW).
     5. Click **Save routes**.
   * **Associate Route Table with Public Subnets**:
     1. Go to the **Subnet Associations** tab.
     2. Click **Edit subnet associations**.
     3. Select **skillxchnage-web-sb-ap-south-1a** and **skillxchnage-web-sb-ap-south-1b**.
     4. Click **Save** to associate the public route table with the web subnets.

**Create Route Table for Private Subnets (skillxchnage-pr-rt):**

1. In the **Route Tables** section, click **Create route table** again.
2. **Name**: skillxchnage-pr-rt (Private Route Table).
3. **VPC**: Select your VPC (skillxchnage-vpc).
4. Click **Create** to create the route table.
   * **Edit Routes**:
     1. Select the **Private Route Table** (skillxchnage-pr-rt).
     2. **Add Routes** for private subnets (if you have a NAT Gateway or NAT Instance):
        + **Destination**: 0.0.0.0/0
        + **Target**: **NAT Gateway** or **NAT Instance** (for outbound internet access from private subnets).
     3. Click **Save routes**.
   * **Associate Route Table with Private Subnets**:
     1. Go to the **Subnet Associations** tab.
     2. Click **Edit subnet associations**.
     3. Select **skillxchnage-app-sb-ap-south-1a**, **skillxchnage-app-sb-ap-south-1b**, **skillxchnage-db-sb-ap-south-1a**, and **skillxchnage-db-sb-ap-south-1b**.
     4. Click **Save** to associate the private route table with the private subnets.

**Recap of Your Setup**

**Subnets:**

* **Public Subnets**:
  + skillxchnage-web-sb-ap-south-1a (10.0.1.0/24) in **ap-south-1a**
  + skillxchnage-web-sb-ap-south-1b (10.0.2.0/24) in **ap-south-1b**
* **Private Subnets**:
  + **App Subnets**:
    - skillxchnage-app-sb-ap-south-1a (10.0.3.0/24) in **ap-south-1a**
    - skillxchnage-app-sb-ap-south-1b (10.0.4.0/24) in **ap-south-1b**
  + **DB Subnets**:
    - skillxchnage-db-sb-ap-south-1a (10.0.5.0/24) in **ap-south-1a**
    - skillxchnage-db-sb-ap-south-1b (10.0.6.0/24) in **ap-south-1b**

**Route Tables:**

* **Public Route Table (skillxchnage-pb-rt)**:
  + Used for the public subnets.
  + Routes all internet-bound traffic to the **Internet Gateway** (skillxchnage-IGW).
* **Private Route Table (skillxchnage-pr-rt)**:
  + Used for the private subnets.
  + Routes outbound traffic to a **NAT Gateway** (or **NAT Instance**) for internet access.

**2. Create an Internet Gateway**

1. In the **Internet Gateways** section, click the **Create internet gateway** button.
2. **Name tag**: Give it a name (e.g., skillxchnage-IGW).
3. Click **Create** to create the internet gateway.

**3. Attach the Internet Gateway to Your VPC**

1. After the Internet Gateway is created, it will appear in the list.
2. Select the newly created Internet Gateway (skillxchnage-IGW).
3. In the **Actions** dropdown, click on **Attach to VPC**.
4. **VPC**: Select your VPC (skillxchnage-vpc).
5. Click **Attach**.

**Steps to Route Traffic to the Internet Gateway via the Public Route Table**

**1. Go to the Route Tables Section**

1. In the **VPC Dashboard**, navigate to the **Route Tables** section on the left sidebar.
2. You should see the list of all route tables in your VPC.

**2. Select the Public Route Table (skillxchnage-pb-rt)**

1. Find and select the **Public Route Table** you created earlier (named skillxchnage-pb-rt).
2. Click on the **Route Table Name** (e.g., skillxchnage-pb-rt) to open the details of the route table.

**3. Edit Routes to Add a Default Route to the Internet Gateway**

1. In the **Routes** tab of the public route table, click on **Edit routes**.
2. Click on **Add route**.
   * **Destination**: 0.0.0.0/0 (this represents all traffic destined for the internet).
   * **Target**: Select **Internet Gateway** and then choose the **Internet Gateway** you created earlier (skillxchnage-IGW).
3. Click **Save routes** to apply the changes.

**Create Security Groups**

**Create Web Tier Security Group**

1. In the **VPC Dashboard**, go to **Security Groups** on the left sidebar.
2. Click **Create security group**.
3. **Name**: skillxchnage-web-sg.
4. **Description**: Web tier security group for SkillXchange.
5. **VPC**: Select skillxchnage-vpc.
6. Click **Create**.

**5.2. Configure Inbound Rules**

1. Select the **skillxchnage-web-sg** and go to the **Inbound rules** tab.
2. Click **Edit inbound rules** → **Add Rule**:
   * **SSH** (port 22) from your specific IP or a trusted range.
   * **HTTP** (port 80) from 0.0.0.0/0 (anywhere).
   * **HTTPS** (port 443) from 0.0.0.0/0 (anywhere).
3. Click **Save rules**.

**5.3. Configure Outbound Rules**

1. Leave the **Outbound rules** as **Allow all traffic** (0.0.0.0/0).

**## Steps to Launch an EC2 Instance in the Public Subnet**

Launch a free-tier **Ubuntu EC2 instance** in the **public subnet** of the **SkillXchange VPC**.

**2. Navigate to EC2 Dashboard**

1. In the **AWS Console**, search for **EC2** in the search bar and select **EC2** to open the EC2 Dashboard.

**3. Launch a New EC2 Instance**

1. In the EC2 Dashboard, click on **Launch Instance**.

**4. Choose an Amazon Machine Image (AMI)**

1. Select **Ubuntu Server 20.04 LTS** (Free Tier eligible) under the **Quick Start** tab.

**5. Choose Instance Type**

1. Choose **t2.micro** (Free Tier eligible).
2. Click **Next: Configure Instance Details**.

**6. Configure Instance**

1. **Network**: Select your **VPC** (skillxchnage-vpc).
2. **Subnet**: Choose one of the **public subnets** (skillxchnage-web-sb-ap-south-1a or skillxchnage-web-sb-ap-south-1b).
3. **Auto-assign Public IP**: Select **Enable** to assign a public IP address.
4. Click **Next: Add Storage**.

**7. Add Storage**

1. Use the default **8 GB** of storage.
2. Click **Next: Add Tags**.

**8. Add Tags (Optional)**

1. Add a tag with:
   * **Key**: Name
   * **Value**: Ubuntu-Instance
2. Click **Next: Configure Security Group**.

**9. Configure Security Group**

1. Select **Select an existing security group**.
2. Choose **skillxchnage-web-sg** (Security Group for web servers).
3. Ensure the security group allows **SSH (port 22)**, **HTTP (port 80)**, and **HTTPS (port 443)**.
4. Click **Review and Launch**.

**10. Review and Launch**

1. Review your instance configuration.
2. Click **Launch**.

#############################################

**Step 2: Setup Application on EC2**

* SSH into EC2 instance
* Install Node.js and Nginx (or serve the React app with Nginx)
* Copy React build files to /var/www/html or configured directory
* Start the Nginx server or app server (e.g., using serve)

**Step 3: Configure Load Balancer**

* Type: **Application Load Balancer**
* Scheme: Internet-facing
* Listeners:
  + HTTP (80)
  + HTTPS (443) with ACM SSL Certificate
* Target Group:
  + Type: instance or IP
  + Protocol: HTTP
  + Target: Your EC2 instance
* Health Checks: Ensure they pass (use / or your app’s base path)

**Step 4: Set Up ACM (SSL Certificate)**

* Go to ACM → Request a public certificate
* Add domain (e.g., app.skillxchange.in)
* Validate using **DNS validation**
* Once issued, attach it to the HTTPS listener of your load balancer

**Step 5: Configure Route 53**

* Hosted Zone: Create or use an existing one for skillxchange.in
* Create a **Record Set (A or CNAME)**:
  + **Name**: app.skillxchange.in
  + **Type**: A (alias to ALB) or CNAME (to ALB DNS name)
  + **Alias**: Yes (if using A record)
  + **Value**: Your Load Balancer DNS name (e.g., my-alb-123456.ap-south-1.elb.amazonaws.com)

**✅ Final Check**

* Go to your browser and open:  
  **https://app.skillxchange.in**
* You should see your **React app with HTTPS** (SSL secured)

**Dependency go to /etc/nginx/sites-avilable/default**

**Vim default : check the changes add url for running app**

server {

listen 80 default\_server;

listen [::]:80 default\_server;

root /var/www/html;

index index.html index.htm index.nginx-debian.html;

server\_name \_;

location / {

proxy\_pass http://localhost:3000; # Forward requests to Node.js app on port 3000

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

:wq

**🚀 Launch EC2 Instance in Private Subnet + NAT Gateway (Temporary Internet Access) for application and connect with DB**

**🔧 Step 1: Launch EC2 in Private Subnet**

1. **Go to EC2 Dashboard** → Click **Launch Instance**
2. **Choose AMI**: Ubuntu Server 20.04 LTS
3. **Instance Type**: t2.micro (Free Tier)
4. **Configure Instance:**
   * **Network**: skillxchnage-vpc
   * **Subnet**: Choose one of your private subnets:
     + skillxchnage-app-sb-ap-south-1a or
     + skillxchnage-app-sb-ap-south-1b
   * **Auto-assign Public IP**: **Disabled**
5. **Add Storage**: Leave default (8 GB)
6. **Add Tags**: (Optional)
   * Name = skillxchnage-app-instance
7. **Configure Security Group**:
   * Create or use one that allows:
     + **SSH** from your **bastion or jump box**
     + **Inbound traffic** from load balancer / web tier
8. **Review and Launch**

**🌐 Step 2: Create a NAT Gateway (in Public Subnet)**

A NAT Gateway allows private instances to access the internet temporarily for updates or external calls.

1. **Go to VPC Dashboard → NAT Gateways**
2. **Click Create NAT Gateway**
   * **Name**: skillxchnage-nat-gw
   * **Subnet**: Select a **Public Subnet** (e.g., skillxchnage-web-sb-ap-south-1a)
   * **Elastic IP**: Allocate a new Elastic IP or choose existing
3. Click **Create NAT Gateway**

**🔁 Step 3: Update Route Table for Private Subnets**

1. Go to **Route Tables** in VPC Dashboard
2. Select skillxchnage-pr-rt (Private Route Table)
3. Go to **Routes** → Click **Edit routes**
4. Click **Add route**:
   * **Destination**: 0.0.0.0/0
   * **Target**: Select your **NAT Gateway** (skillxchnage-nat-gw)
5. Click **Save routes**

**🛠️ Connect EC2 App Instance (Private Subnet) to RDS MySQL Database**

**✅ Step 1: Ensure Your RDS Instance is Ready**

If you haven’t already:

1. Go to **RDS Dashboard**
2. Click **Create database**
   * Engine: **MySQL**
   * DB Instance Identifier: skillxchnage-db
   * Credentials: Save the **username** and **password**
   * VPC: **skillxchnage-vpc**
   * Subnet Group: Make sure it includes **private DB subnets**:
     + skillxchnage-db-sb-ap-south-1a
     + skillxchnage-db-sb-ap-south-1b
   * Public access: **No**
   * VPC security group: Create or select one that **allows MySQL (port 3306)** from the **app security group**
3. Click **Create database**

**🔐 Step 2: Setup Security Groups for Access**

**DB Security Group (skillxchnage-db-sg)**

* Inbound Rule:
  + Type: **MySQL/Aurora**
  + Port: **3306**
  + Source: **Security group of your app instance** (skillxchnage-app-sg)  
    *(Not CIDR like 10.0.3.0/24 — use SG-to-SG reference)*

**App Security Group (skillxchnage-app-sg)**

* No changes needed unless you want to allow SSH or traffic from other layers (like LB or bastion)

**🔄 Step 3: Connect from EC2 to RDS**

**SSH or SSM into the App EC2 Instance**

You can either connect through a **bastion host** in the public subnet or use **AWS Systems Manager (SSM)** if the instance has SSM agent and IAM permissions.

**Install MySQL Client (if not installed)**

bash

CopyEdit

sudo apt update

sudo apt install mysql-client -y

**Connect to the RDS DB**

bash

CopyEdit

mysql -h <your-rds-endpoint> -u <username> -p

* Replace <your-rds-endpoint> with something like skillxchnage-db.abc123xyz.ap-south-1.rds.amazonaws.com
* It will prompt for your password

**🔄 Step-by-Step: Create Internal & External Load Balancers and Route 53 Integration**

**✅ 1. Create External Application Load Balancer (Web Tier)**

**📍 Navigate to EC2 > Load Balancers**

1. Go to the **EC2 Dashboard**.
2. On the left menu, click **Load Balancers** → **Create Load Balancer**.

**⚙️ Configuration**

* **Choose**: Application Load Balancer (ALB)
* **Name**: skillxchnage-external-alb
* **Scheme**: Internet-facing
* **IP address type**: IPv4

**🌐 Network Mapping**

* **VPC**: Select skillxchnage-vpc
* **Availability Zones**:
  + Select ap-south-1a and ap-south-1b
  + Choose public subnets:
    - skillxchnage-web-sb-ap-south-1a
    - skillxchnage-web-sb-ap-south-1b

**🔒 Security Groups**

* Select the security group: skillxchnage-web-sg

**📥 Listeners**

* Add a **listener**:
  + **Protocol**: HTTP (Port 80)
  + (You’ll attach HTTPS with ACM certificate later.)

**📌 Target Group (for Web Tier EC2s)**

* **Create a new target group**
  + **Name**: web-tier-tg
  + **Target type**: Instance
  + **Protocol**: HTTP
  + **Port**: 80
* **Register EC2 web tier instances**

**🧪 Health Check Settings**

* Path: / or your app health check path

✅ Click **Create Load Balancer**

**🔄 2. Create Internal Application Load Balancer (App Tier)**

**📍 Navigate back to EC2 > Load Balancers → Create Load Balancer**

**⚙️ Configuration**

* **Name**: skillxchnage-internal-alb
* **Scheme**: Internal
* **IP address type**: IPv4

**🌐 Network Mapping**

* **VPC**: skillxchnage-vpc
* **Subnets**: Select private app subnets:
  + skillxchnage-app-sb-ap-south-1a
  + skillxchnage-app-sb-ap-south-1b

**🔒 Security Group**

* Create or select skillxchnage-app-sg (Allow traffic from Web SG on port 4000 or 3000 depending on your app)

**📥 Listener**

* Protocol: HTTP, Port: 4000 (or whatever your Node app uses)

**📌 Target Group**

* **Name**: app-tier-tg
* **Target type**: Instance
* **Port**: 4000
* Register your **App EC2 instances**

✅ Click **Create Load Balancer**

**🔐 3. Set Up ACM (SSL Certificate)**

**📍 Navigate to ACM (AWS Certificate Manager)**

1. Click **Request a Certificate**
2. Select: **Public Certificate**
3. Add domain:  
   Example: app.skillxchange.in
4. **Validation**: DNS Validation (preferred for Route 53)
5. ACM will give a **CNAME record** → Copy this.

✅ Click **Next** and Request Certificate.

**🌐 4. Configure Route 53 (DNS)**

**📍 Open Route 53 > Hosted Zones**

1. Select hosted zone for your domain: skillxchange.in
2. Click **Create Record**:
   * **Record name**: app
   * **Type**: A or CNAME
   * **Alias**: Yes (if A record)
   * **Target**: Select your **external ALB DNS** (e.g., skillxchange-alb-123.ap-south-1.elb.amazonaws.com)

✅ Save record.

**🔗 5. Attach ACM Certificate to HTTPS Listener**

**📍 Go back to EC2 > Load Balancers**

1. Select skillxchnage-external-alb
2. Under **Listeners**, click **Add Listener**
   * Protocol: HTTPS
   * Port: 443
3. **SSL Certificate**: Choose your validated ACM certificate
4. **Target Group**: Select web-tier-tg

✅ Click **Save**

**🔄 Bonus: Ensure Traffic Only Goes Through Load Balancer**

**✅ Modify Security Groups:**

**Web Tier SG:**

* Inbound: Only allow from **Internet** on 80, 443
* Outbound: Allow to App SG (internal ALB)

**App Tier SG:**

* Inbound: Only allow from **Web SG**
* Outbound: Allow DB SG or internet via NAT if needed

**🧪 Final Test**

1. Go to browser → <https://app.skillxchange.in>
2. You should see your app via Load Balancer over HTTPS