# Step by step guide

Phase 1

## Create a VPC

Go to: **VPC Dashboard → Your VPCs → Create VPC**

| **Field** | **Value** |
| --- | --- |
| **Name tag** | ThreeTier-VPC |
| **IPv4 CIDR block** | 10.0.0.0/16 |
| **Tenancy** | Default |
| IPv6 CIDR block | No IPv6 for now |
| **Enable DNS Hostnames** | Yes (important for public DNS resolution) |

A screenshot of a computer

AI-generated content may be incorrect.

## 2. Create Subnets

1. Go to **VPC Dashboard → Subnets → Create subnet**
2. Select your VPC: ThreeTier-VPC
3. Create **one at a time**:
   * **Name Tag**: Use names from table above (e.g., Web-Pub-1)
   * **Availability Zone**: Choose us-east-1a, us-east-1b, etc.
   * **IPv4 CIDR**: e.g., 10.0.1.0/24

| **Tier** | **AZ** | **Subnet Name** | **CIDR Block** | **Type** |
| --- | --- | --- | --- | --- |
| Web Tier | AZ1 | Web-Pub-1 | 10.0.1.0/24 | Public |
| Web Tier | AZ2 | Web-Pub-2 | 10.0.2.0/24 | Public |
| App Tier | AZ1 | App-Priv-1 | 10.0.3.0/24 | Private |
| App Tier | AZ2 | App-Priv-2 | 10.0.4.0/24 | Private |
| DB Tier | AZ1 | DB-Priv-1 | 10.0.5.0/24 | Private |
| DB Tier | AZ2 | DB-Priv-2 | 10.0.6.0/24 | Private |

A screenshot of a computer

AI-generated content may be incorrect.

## 3. Create and Attach IGW

1. Go to **VPC Dashboard → Internet Gateways → Create Internet Gateway**
2. **Name**: ThreeTier-IGW
3. Click **Create Internet Gateway**
4. After creation, **select it → Actions → Attach to VPC**
5. Choose your VPC: ThreeTier-VPC

A screenshot of a computer

AI-generated content may be incorrect.

## Public Route Table

1. **VPC Dashboard → Route Tables → Create Route Table**
2. **Name**: Public-RT
3. **VPC**: Select ThreeTier-VPC
4. Click **Create**
5. Select Public-RT → **Routes tab → Edit routes**
6. Click **Add route**:
   * **Destination**: 0.0.0.0/0
   * **Target**: Choose the Internet Gateway (ThreeTier-IGW)
7. Click **Save changes**
8. Now go to **Subnet Associations → Edit subnet associations**
   * Select: Web-Pub-1 and Web-Pub-2
   * Click **Save**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Private Route Table

1. Back in **Route Tables → Create Route Table**
2. **Name**: Private-RT
3. **VPC**: Select ThreeTier-VPC
4. Click **Create**
5. **No need to add route to IGW**
6. Go to **Subnet Associations → Edit subnet associations**
   * Select: App-Priv-1, App-Priv-2, DB-Priv-1, DB-Priv-2
   * Click **Save**

A screenshot of a computer

AI-generated content may be incorrect.

**[ Web-Pub-1, Web-Pub-2 ] --> Public-RT --> IGW --> Internet**

**[ App-Priv-1..DB-Priv-2 ] --> Private-RT --> (internal only)**

## Create a NAT Gateway

1. Go to **VPC Dashboard → NAT Gateways → Create NAT Gateway**
2. **Name**: NAT-GW-AZ1
3. **Subnet**: Select Web-Pub-1 (public subnet)
4. **Elastic IP**: Click “Allocate Elastic IP” → choose it
5. Click **Create NAT Gateway**

Wait for status to show as Available.

A screenshot of a computer

AI-generated content may be incorrect.

## Update Private Route Table to Use NAT

1. Go to **Route Tables → Select Private-RT**
2. Click **Routes tab → Edit routes**
3. **Add route**:
   * **Destination**: 0.0.0.0/0
   * **Target**: Your newly created **NAT Gateway**

Click **Save Changes**

A screenshot of a computer

AI-generated content may be incorrect.

Phase 2

| **SG Name** | **Attached To** | **Inbound Rules From** | **Outbound** |
| --- | --- | --- | --- |
| SG-Web | Web EC2 / ALB | 0.0.0.0/0: HTTP (80), SSH (22) | All traffic |
| SG-App | App EC2 | SG-Web: TCP 3000 (custom app port) | All traffic |
| SG-DB | RDS DB | SG-App: TCP 3306 (MySQL/Aurora) | All traffic |

## SG Design for 3-Tier Architecture

## Create Security Groups

**SG-Web (Web Tier)**

1. Go to **EC2 → Security Groups → Create Security Group**
2. **Name**: SG-Web
3. **VPC**: ThreeTier-VPC
4. **Inbound Rules**:
   * Type: HTTP | Port: 80 | Source: 0.0.0.0/0
   * Type: SSH | Port: 22 | Source: your IP (My IP) *(or restrict it later)*
5. **Outbound**: Allow all (default)

A screenshot of a computer

AI-generated content may be incorrect.

**SG-App (App Tier)**

1. **Name**: SG-App
2. Inbound Rules:
   * Type: Custom TCP | Port: 3000 *(or your backend port)* | Source: SG-Web
3. Outbound: Allow all

A screenshot of a computer

AI-generated content may be incorrect.

**SG-DB (Database Tier)**

1. **Name**: SG-DB
2. Inbound Rules:
   * Type: MySQL/Aurora | Port: 3306 | Source: SG-App
3. Outbound: Allow all

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Public NACL

1. Go to **VPC Dashboard → Network ACLs → Create NACL**
2. **Name**: Public-NACL
3. **VPC**: ThreeTier-VPC
4. Click **Create**
5. Click on the new NACL → **Subnet Associations**
   * Associate: Web-Pub-1, Web-Pub-2

| **Rule #** | **Type** | **Protocol** | **Port Range** | **Source** | **Allow/Deny** |
| --- | --- | --- | --- | --- | --- |
| 100 | HTTP | TCP | 80 | 0.0.0.0/0 | ALLOW |
| 110 | SSH | TCP | 22 | your IP CIDR | ALLOW |
| 120 | Ephemeral | TCP | 1024–65535 | 0.0.0.0/0 | ALLOW |
| \* | ALL | ALL | ALL | 0.0.0.0/0 | DENY |

| **Rule #** | **Type** | **Protocol** | **Port Range** | **Destination** | **Allow/Deny** |
| --- | --- | --- | --- | --- | --- |
| 100 | HTTP | TCP | 80 | 0.0.0.0/0 | ALLOW |
| 110 | Ephemeral | TCP | 1024–65535 | 0.0.0.0/0 | ALLOW |
| \* | ALL | ALL | ALL | 0.0.0.0/0 | DENY |

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Private NACL

Repeat above process for Private-NACL and associate:

* App-Priv-1
* App-Priv-2
* DB-Priv-1
* DB-Priv-2

Inbound rules

| **Rule #** | **Type** | **Port Range** | **Source** | **Allow/Deny** |
| --- | --- | --- | --- | --- |
| 100 | Custom TCP | 3000 | 10.0.1.0/24 | ALLOW |
| 110 | MySQL | 3306 | 10.0.3.0/24 | ALLOW |
| 120 | Ephemeral | 1024–65535 | 10.0.1.0/24 | ALLOW |
| \* | ALL | ALL | 0.0.0.0/0 | DENY |

Outbound rules

| **Rule #** | **Type** | **Port Range** | **Destination** | **Allow/Deny** |
| --- | --- | --- | --- | --- |
| 100 | Ephemeral | 1024–65535 | 0.0.0.0/0 | ALLOW |
| \* | ALL | ALL | 0.0.0.0/0 | DENY |

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Create a Key Pair (for SSH into EC2)

1. Go to **EC2 Dashboard → Key Pairs → Create Key Pair**
2. **Name**: ThreeTier-KeyPair
3. **Key pair type**: RSA (recommended for SSH)
4. **Private key format**: .pem (for Linux/Mac)
5. Click **Create key pair**
6. File will download as: ThreeTier-KeyPair.pem

A screenshot of a computer

AI-generated content may be incorrect.

Phase 3

## Launching Ec2 Instances

Public Instance:

A screenshot of a computer

AI-generated content may be incorrect.

App-tier Instance:

A screenshot of a computer

AI-generated content may be incorrect.

DB-tier-Instance:

A screenshot of a computer

AI-generated content may be incorrect.

Phase 4

## Creating Target Group & Application Load Balancer

Target Group:

A screenshot of a computer

AI-generated content may be incorrect.

Application Load Balancer:

A screenshot of a computer

AI-generated content may be incorrect.

Phase 5

## Database Tier (RDS Free Tier)

**Step 1: Create a DB Subnet Group**

1. Go to **RDS > Subnet groups > Create DB subnet group**
2. Name: db-subnet-group-p2
3. VPC: Select your VPC (vpc-p2)
4. Add **2 subnets** (must be in different AZs):
   * db1-p2 (e.g., 10.0.5.0/24)
   * db2-p2 (e.g., 10.0.6.0/24)

A screenshot of a computer

AI-generated content may be incorrect.

**Step 2: Launch RDS Instance (MySQL)**

1. Go to **RDS > Databases > Create database**
2. Choose **Standard Create**
3. Engine: **MySQL**
4. Version: Latest **Free Tier eligible** (e.g., 8.x)
5. Template: **Free Tier**
6. DB Instance Identifier: dipen-db
7. Master username: admin
8. Password: Choose strong password (store it safely)
9. DB instance size: **db.t3.micro**
10. Storage: 20 GB (General Purpose SSD)

A screenshot of a computer

AI-generated content may be incorrect.

**Note: DB tier EC2 instance has \*\*two SGs attached\*\*:**

**1. db-sg-p2 → Accepts SSH from app tier SG**

**2. private-sg-p2 → Enables outbound MySQL traffic to RDS**

**This dual SG setup allows full connectivity between layers while maintaining modular SG rules.**

## RDS Connection

A screenshot of a computer

AI-generated content may be incorrect.

## Flowchart of RDS Connection

A diagram of a computer network

AI-generated content may be incorrect.

| **Tier** | **Security Group** | **Inbound Allows From** | **Port(s)** |
| --- | --- | --- | --- |
| Web Tier | pub-sg-p2 | My IP (SSH/HTTP) | 22, 80 |
| App Tier | private-sg-p2 | pub-sg-p2 | 22, 80 |
| DB Tier | db-sg-p2 | private-sg-p2 | 22, 3306 |
| RDS | db-sg-p2 | db-sg-p2 (self-ref) | 3306 |

Home Page (Web-tier)

A screenshot of a computer

AI-generated content may be incorrect.

Add Employee

A screenshot of a computer

AI-generated content may be incorrect.

Adding data

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

View Employees

A screenshot of a computer

AI-generated content may be incorrect.

## Files and Purpose

| **File Name** | **Purpose** |
| --- | --- |
| **form.html** | **Employee data input form** |
| **submit-form.php** | **Validates & inserts data to RDS** |
| **view-employees.html** | **Displays data in styled table** |
| **get-employees.php** | **Sends employee data as JSON (API)** |
| **index.html** | **Optional landing/home page** |

**Flow Structure**

**1. Form Submission (Frontend)**

* **File: form.html**
* **Function: Presents a form to collect:**
  + **Name, Email, Role, Department**
* **Action: On submit, sends a POST request to submit-form.php on the Web Tier EC2.**

**2. Submit Data (Backend PHP)**

* **File: submit-form.php**
* **Runs on: Web Tier EC2**
* **Tasks:**
  + **Connects to RDS MySQL using mysqli**
  + **Validates input fields**
  + **Executes INSERT INTO employees (...) VALUES (...)**
  + **Displays a success message with redirect**

**3. View Data (Frontend + Backend)**

* **Frontend File: view-employees.html**
  + **Uses JavaScript to fetch employee data from get-employees.php**
  + **Dynamically updates the table on load**
* **Backend File: get-employees.php**
  + **Fetches all records from the employees table**
  + **Returns data as a JSON response**