# What is Three-Tier Architecture?

* Three-tier architecture is just a fancy way to split an app into three parts so everything’s neat and manageable.
* Think of it like this:
* **Presentation Layer:** The bit users actually see — the buttons, screens, website stuff.
* **Application Layer:** The brain behind the scenes that does the thinking and handles the rules.
* **Data Layer:** The safe where all the information is stored and managed.
* The best thing about this setup? Each layer lives separately. So different teams can work on each part without stepping on each other’s toes. And if you need to upgrade or expand one part, you don’t risk breaking the others.

# How to Set Up a Three-Tier Architecture

## Step 1: Create a Virtual Network

* First off, I logged into Azure and created a resource group called **ThreeTierAppRG**.
* Then, I made a virtual network named **ThreeTierVNet** with this range:
* 10.0.0.0/16

## Inside that network, I made three subnets:

## WebSubnet → 10.0.1.0/24

## AppSubnet → 10.0.2.0/24

## DBSubnet → 10.0.3.0/24

## Step 2: Set Up Network Security Groups (NSGs)

These are basically traffic cops for each subnet. I set up one NSG for each subnet to control who can talk to whom.

**(a) WebNSG**

* Created an NSG called **WebNSG** in the same resource group.
* Inbound Rules:
  + Allow SSH (port 22) only from my own public IP.
  + Allow HTTP (port 80) from anywhere.
  + Allow traffic from AppSubnet (10.0.2.0/24).
* Outbound Rules:
  + Allow traffic going to AppSubnet.
* Then I linked this NSG to **WebSubnet**.

**(b) AppNSG**

* Allowed SSH from my public IP.
* Allowed traffic from WebSubnet.
* Outbound Rules:
  + Allowed traffic back to WebSubnet.
  + Allowed traffic to DBSubnet.
* Connected this NSG to **AppSubnet**.

**(c) DBNSG**

## Allowed SSH only from AppSubnet.

## Blocked all outbound traffic completely.

## Linked this NSG to DBSubnet.

## Step 3: Deploy Virtual Machines

I made one Linux VM in each subnet. (I skipped Windows because my machine can’t handle it — it’s just too slow.)

**(a) Web Tier VM**

* Resource Group: ThreeTierAppRG
* VM Name: WebLinuxVM
* Region: Same as the resource group
* OS: Ubuntu LTS
* Size: Standard B1s (or any lightweight option)
* Auth: SSH key
* Username: azureuser
* Public inbound ports: None
* Networking:
  + VNet: ThreeTierVNet
  + Subnet: WebSubnet
  + Public IP: Yes (for SSH)
* Clicked **Create** and let Azure do its thing.

**(b) App Tier VM**

Same as above, but:

* Same as the Web VM, except:
* VM Name: AppLinuxVM
* Subnet: AppSubnet
* No public IP.

**(c) DB Tier VM**

Same as above, but:

* VM Name: DBLinuxVM
* Subnet: DBSubnet
* No public IP.

## Step 4: Install Apache2

## I wanted a web server running, so I installed Apache2 — but only on the Web and App VMs, because the DB VM can’t reach the internet.

* On **WebLinuxVM**:

sudo apt-get update

sudo apt-get install apache2 -y

sudo systemctl enable apache2

sudo systemctl start apache2

* On **AppLinuxVM**:

ssh azureuser@<AppVM\_Private\_IP>

Then installed Apache2 the same way.

## Step 5: Testing Connections

* I tested how the tiers could talk:
* **Web Tier → App Tier** → Works fine.
* **Web Tier → DB Tier** → Blocked, as planned.
* **App Tier → Web Tier** → Works.
* **App Tier → DB Tier** → Works.
* **DB Tier → Anyone else** → Blocked.