

Cloud Network & Access Control Architecture for Internal Application

Role: Cloud Network & Security Intern

Cloud Provider: AWS

Region: ap-south-1 (Mumbai)

1. Objective

The goal of this project is to design a **secure cloud network architecture** that allows employees to access an internal application while protecting it from external threats.

The solution must:

- Restrict public exposure
 - Enforce network segmentation
 - Allow access only through defined ports
 - Use basic firewall and security group rules (no IDS/IPS)
-

2. High-Level Architecture Overview

Architecture Pattern Used:

Public Subnet + Bastion Host + Private Application Subnet

Access Flow:

User Laptop

↓ (SSH)

Bastion Host (Public Subnet)

↓ (SSH / App Port)

Application Server (Private Subnet)

This ensures:

- No direct public access to the application server
- A single controlled administrative access path
- Least-privilege network access

3. VPC Design

3.1 Virtual Private Cloud (VPC)

- VPC Name:** Internal-App-VPC
- CIDR Block:** 10.0.0.0/16
- DNS Resolution:** Enabled
- DNS Hostnames:** Enabled

Purpose:

Provides isolated networking for the internal application and administrative resources.

Name	VPC ID	State	Encryption c...	Encryption control...	Block Public...	IPv4 CIDR	IPv6 CIDR
vpc-092be9b5ba332826e	vpc-092be9b5ba332826e	Available	-	-	Off	10.0.0.0/24	-
vpc-091f1e7be55b7a19	vpc-091f1e7be55b7a19	Available	-	-	Off	172.31.0.0/16	-
Internal-App-VPC	vpc-0568c7f72f2647c	Available	-	-	Off	10.0.0.0/16	-

vpc-0568c7f72f2647c / Internal-App-VPC

Details

VPC ID	vpc-0568c7f72f2647c	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	-	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
Encryption control ID	-	Encryption control mode	-
Block Public Access			
Off			
DHCP option set			
dopt-0b933b7ae2fd2b0f4			
IPv4 CIDR			
10.0.0.0/16			
Route 53 Resolver DNS Firewall rule groups			
-			
DNS hostnames			
Disabled			
Main route table			
-			
IPv6 pool			
-			
Owner ID			
194722416808			

4. Subnet Design & Segmentation

4.1 Public Subnet

- Subnet Name:** Public-Subnet
- CIDR:** 10.0.1.0/24
- Purpose:** Hosts the Bastion Host
- Internet Access:** Yes (via Internet Gateway)

4.2 Private Application Subnet

- **Subnet Name:** Private-App-Subnet
- **CIDR:** 10.0.2.0/24
- **Purpose:** Hosts internal application server
- **Internet Access:** No direct internet access

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIE
private	subnet-03f33b67aae23a3af	Available	vpc-092be9b5ba332820e my...	Off	10.0.0.32/28	-	-
public	subnet-046ff1d005e96920d	Available	vpc-092be9b5ba332820e my...	Off	10.0.0.16/28	-	-
<input checked="" type="checkbox"/> Public-Subnet	subnet-00aa2320869af2f64	Available	vpc-0568c7f722f2647c inter...	Off	10.0.1.0/24	-	-
<input checked="" type="checkbox"/> Private-App-Subnet	subnet-0ef84be9dad441784	Available	vpc-0568c7f722f2647c inter...	Off	10.0.2.0/24	-	-

✓ This satisfies **network segmentation** and **internal-only application hosting**.

5. Internet Gateway & Routing

5.1 Internet Gateway (IGW)

- Attached to **Internal-App-VPC**
- Used only by the public subnet

5.2 Route Tables

Public Route Table (Public-RT)

- 10.0.0.0/16 → local

- 0.0.0.0/0 → Internet Gateway
- Associated with **Public-Subnet**

Private Route Table (Private-RT)

- 10.0.0.0/16 → local
- **No route to Internet Gateway**
- Associated with **Private-App-Subnet**

The screenshot shows the AWS VPC Route Tables console. On the left, there's a navigation sidebar with sections like VPC dashboard, Virtual private cloud, Security, and PrivateLink and Lattice. The main area is titled "Route tables (1/4) Info". It lists four route tables: "rtb-0df883150ecd5b995" (Main Yes, VPC vpc-091f51e7be55b7a19), "rtb-07f1eb3fe5e0724" (Main Yes, VPC vpc-092be9b5ba332028e), "Public-RT" (selected, Main No, VPC vpc-0568c77f722f2f647c), and "rtb-08a7806e102ff7fc04" (Main Yes, VPC vpc-0568c77f722f2f647c). The "Public-RT" row has a "Create route table" button. Below the table, there are tabs for Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags. The "Subnet associations" tab is selected, showing "Explicit subnet associations" for the Public-Subnet with CIDR 10.0.1.0/24 and "Subnets without explicit associations" for the Private-App-Subnet with CIDR 10.0.2.0/24.

✓ This ensures **private subnet isolation**.

6. EC2 Instances

6.1 Bastion Host

- **Subnet:** Public-Subnet
- **Public IP:** Assigned
- **Private IP:** 10.0.1.166
- **OS:** Amazon Linux 2023

Purpose:

Acts as the **only administrative entry point** into the private network.

6.2 Application Server

- Subnet:** Private-App-Subnet
- Public IP:** None
- Private IP:** 10.0.2.10

Purpose:

Hosts the internal application and is **not directly accessible from the internet**.

7. Security Groups (Firewall Rules)

7.1 Bastion Host Security Group

Inbound Rules:

- SSH (22) → Allowed from **Admin Public IP only**

Outbound Rules:

- All traffic allowed (default)

Security Benefit:

Prevents open SSH access from the internet.

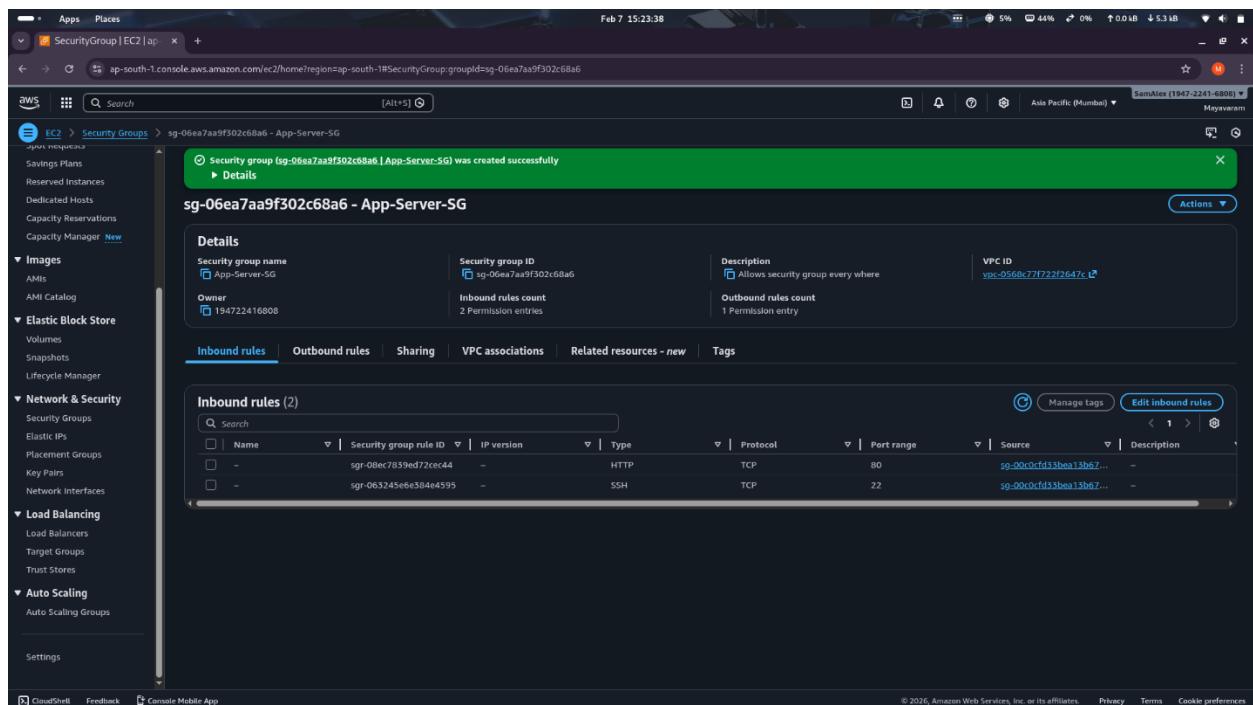
7.2 Application Server Security Group (App-Server-SG)

Inbound Rules:

- SSH (22) → Allowed **only from Bastion Host Security Group**
- HTTP (80) → Allowed **only from Bastion Host Security Group**

Outbound Rules:

- All traffic allowed (default)



The screenshot shows the AWS EC2 Security Groups console. A success message at the top indicates that the security group was created successfully. The main details page for 'sg-06ea7aa9f302c68a6 - App-Server-SG' shows the following information:

- Security group name:** App-Server-SG
- Security group ID:** sg-06ea7aa9f302c68a6
- Description:** Allows security group everywhere
- VPC ID:** vpc-056bc77f72ff2647c
- Inbound rules count:** 2 Permission entries
- Outbound rules count:** 1 Permission entry

The 'Inbound rules' tab is selected, displaying two entries:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
sgr-08ec7859ed72cc44	-	-	HTTP	TCP	80	sg-00c0cf133bea13b67...	-
sgr-063245e6e384e4595	-	-	SSH	TCP	22	sg-00c0cf133bea13b67...	-

✓ This enforces **defined ports only** and **restricted access paths**.

8. Access Method (Important for Submission)

Step 1: SSH to Bastion Host

```
ssh -i "key_pair.pem" ec2-user@<Bastion_Public_IP>
```

Step 2: From Bastion → Application Server

```
ssh ec2-user@10.0.2.10
```

- ✓ Confirms **one administrative access path**.

9. Simulated Common Mistakes & Corrections

Mistake 1: Opening SSH (22) to 0.0.0.0/0

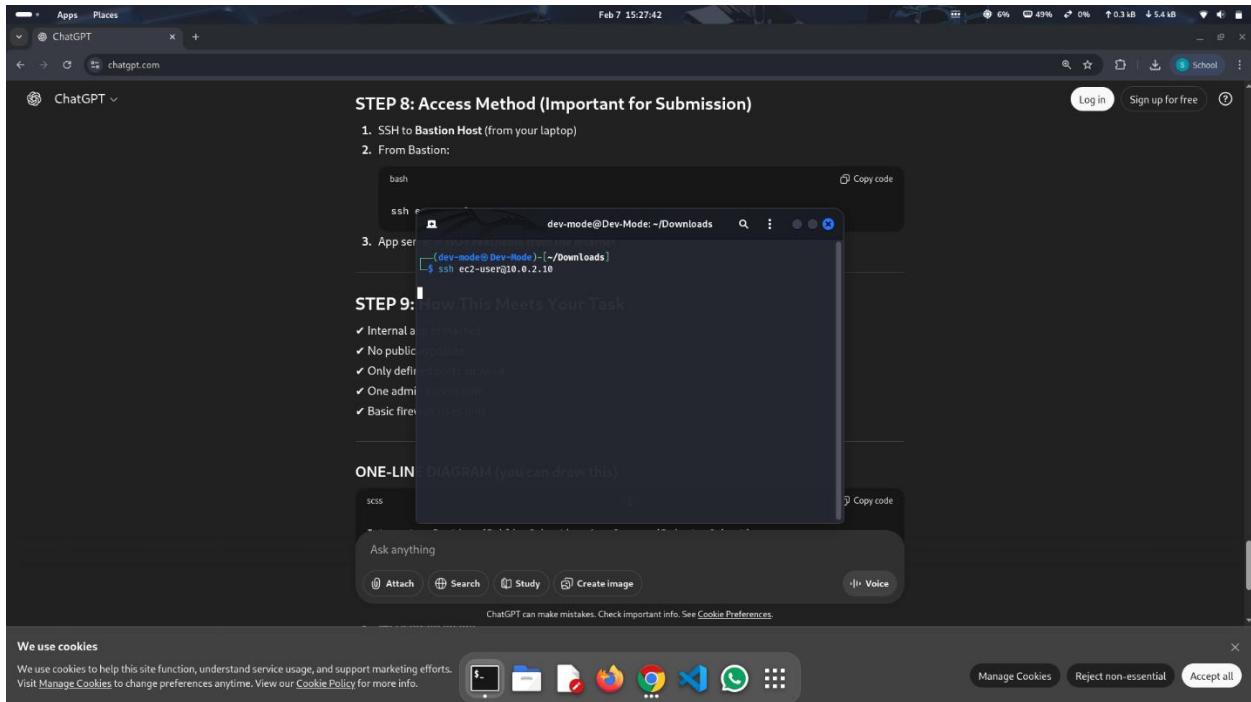
- ✗ Risk: Anyone on the internet could attempt access
 - ✓ Fix: Restricted SSH to admin IP and Bastion SG only

Mistake 2: Public IP on Application Server

- ✗ Risk: Direct external exposure
 - ✓ Fix: Removed public IP and placed instance in private subnet

Mistake 3: Internet Gateway Route on Private Subnet

- ✗ Risk: Unnecessary internet exposure
 - ✓ Fix: Used separate private route table without IGW



10. How This Meets Task Requirements

- ✓ Internal application protected
- ✓ No direct public exposure
- ✓ Network segmentation using subnets
- ✓ Only defined ports allowed
- ✓ One administrative access path (Bastion Host)
- ✓ Basic firewall & security group rules only
- ✓ Fully documented and diagrammable

11. Final Architecture Diagram (Textual)

Internet



[Internet Gateway]



[Public Subnet]



[Bastion Host]



[Private App Subnet]



[Application Server]

The screenshot shows the AWS CloudWatch Metrics interface with the following details:

- Instances (3/3) Info:** A table listing three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Employees	i-081b1e070b4c61041	Stopping	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-	43.205.143.150	-
Web App Server	i-0fe05569c02851a4	Stopping	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-	-	-
Bastion-Host	i-0255050b6d459f5b6	Stopping	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-	13.201.86.185	-
- Monitoring:** A section displaying six time-series charts for the selected instances over a 1-hour period:
 - CPU utilization (%)
 - Network in (bytes)
 - Network out (bytes)
 - Network packets in (count)
 - Metadata no token (count)
 - CPU credit usage (count)
 - CPU credit balance (count)