

Step-by-Step Implementation

| by Shloka Kamdar

Step 1: Set up the virtual private cloud

1. Create VPC

- **Name:** vpc-shloka-nginx
- **Region:** us-west-2 (Oregon)
- **CIDR:** 10.0.0.0/16

Reason: Provides an isolated networking environment for secure infrastructure.

2. Create Subnets (Minimum 2 for HA)

- **private-subnet-01-shloka-nginx:** 10.0.1.0/24 (us-west-2a)
- **public-subnet-01-shloka-nginx:** 10.0.2.0/24 (us-west-2a)
- **private-subnet-02-shloka-nginx :** 10.0.3.0/24 (us-west-2b)
- **public-subnet-02-shloka-nginx:** 10.0.4.0/24 (us-west-2b)

Reason: Separate public-facing resources from private application servers.

3. Create IGW Gateway

- **IGW Name:** igw-shloka-nginx
- Attached to VPC

Reason: Allows internet traffic for ALB and Bastion.

4. Create NAT Gateway

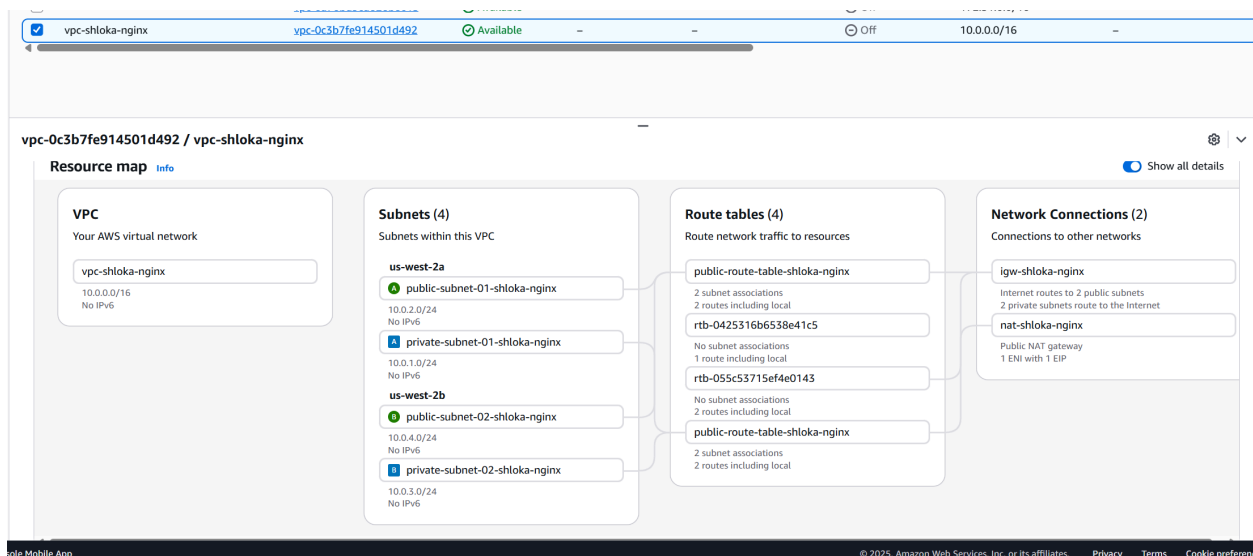
- Created a NAT Gateway: **nat-shloka-nginx**
- **Reason:** Private EC2 could not install packages → user-data failed. NAT fixed this.

5. Create Route Tables

- **Public Route Table:** public-route-table-shloka-nginx
 - 0.0.0.0/0 → Internet Gateway
- **Private Route Table:** private-route-table-shloka-nginx
 - 0.0.0.0/0 → NAT Gateway

Reason: Ensures public resources get internet and private resources get outbound access through NAT.

Resource Map —



Step 2: Launch EC2 Instances

1. 2 Private EC2 instances (for application) in different subnets

- **instance02-shloka-nginx** - private subnet A
- **instance01-shloka-nginx** - private subnet B

User Data Script :

```
#!/bin/bash
sudo apt update -y
sudo apt install -y nginx
```

```
sudo sed -i 's/listen 80 default_server;/listen 8443 default_server;/' /etc/nginx/sites-available/default
sudo sed -i 's/listen \[:\]:80 default_server;/listen \[:\]:8443 default_server;/' /etc/nginx/sites-available/default
```

```
echo "<h1>Shloka — ALB Test on Port 8443</h1>" | sudo tee /var/www/html/index.html
```

```
sudo systemctl enable nginx
sudo systemctl restart nginx
```

2. A Bastion Instance to SSH into other instances in private subnet

- **bastion-shloka-nginx** - public subnet A

The screenshot displays the AWS Management Console. At the top, there's a table of EC2 instances. Below it, the details for the instance 'instance01-shloka-nginx' (ID: i-0c56bc8fa40953ac6) are shown. The instance is in a 'Running' state, using a 't3.micro' instance type, and is located in the 'us-west-2a' availability zone. It has a public IPv4 address of 18.236.64.3. The details panel on the right shows various configuration options like VPC ID, Subnet ID, and Instance ARN.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
bastion-shloka-nginx	i-0f2c5535f1c60ece	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2b	-	18.236.64.3	-	-
instance02-shloka-nginx	i-0750431a3e9e19390	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2b	-	-	-	-
instance01-shloka-nginx	i-0c56bc8fa40953ac6	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2a	-	-	-	-

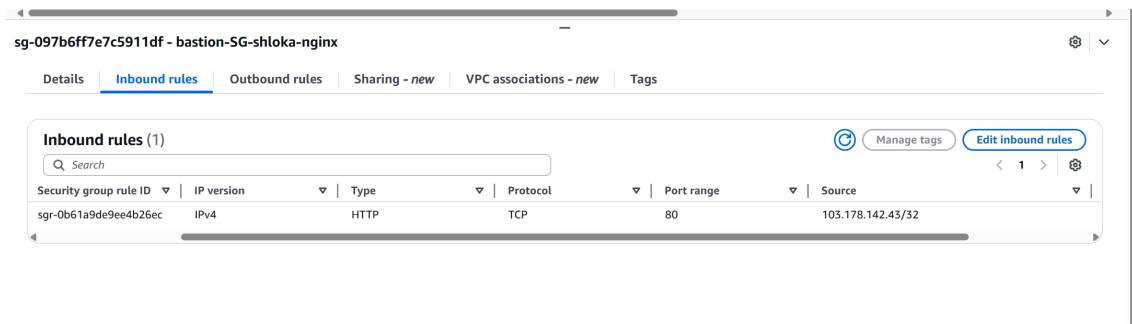
Instance summary

- Instance ID:** i-0c56bc8fa40953ac6
- IPv6 address:** -
- Hostname type:** IP name: ip-10-0-1-60.us-west-2.compute.internal
- Answer private resource DNS name:** -
- Auto-assigned IP address:** -
- IAM Role:** -
- IMDSv2:** Required
- Public IPv4 address:** -
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-10-0-1-60.us-west-2.compute.internal
- Instance type:** t3.micro
- VPC ID:** vpc-0c3b7fe914501d492 (vpc-shloka-nginx)
- Subnet ID:** subnet-0be9f924f905c16e2 (private-subnet-01-shloka-nginx)
- Instance ARN:** arn:aws:ec2:us-west-2:521069555098:instance/i-0c56bc8fa40953ac6
- Private IPv4 addresses:** 10.0.1.60
- Public DNS:** -
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations. | Learn more
- Auto Scaling Group name:** -
- Managed:** false

Step 3: Configure Security Groups

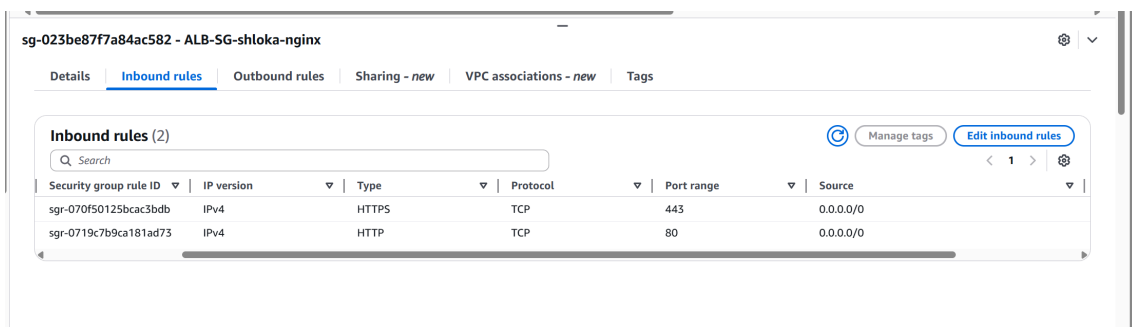
1. Bastion SG:

- SSH (22): My IP only



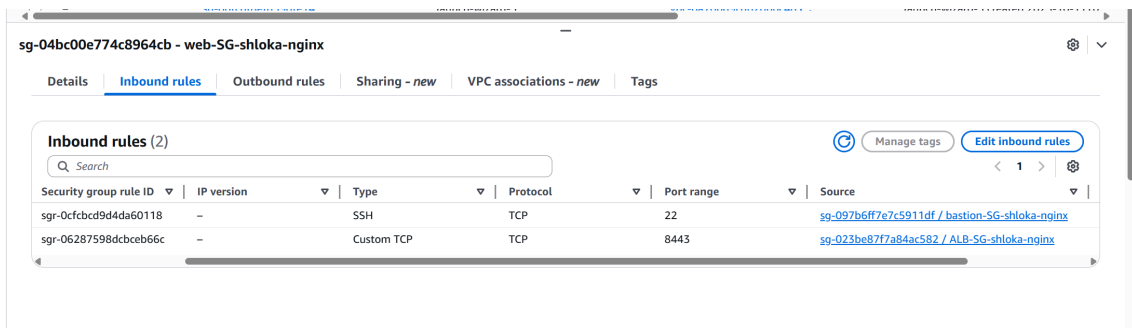
2. ALB SG:

- HTTP (80): 0.0.0.0/0
- HTTPS (443): 0.0.0.0/0



3. Private EC2 SG:

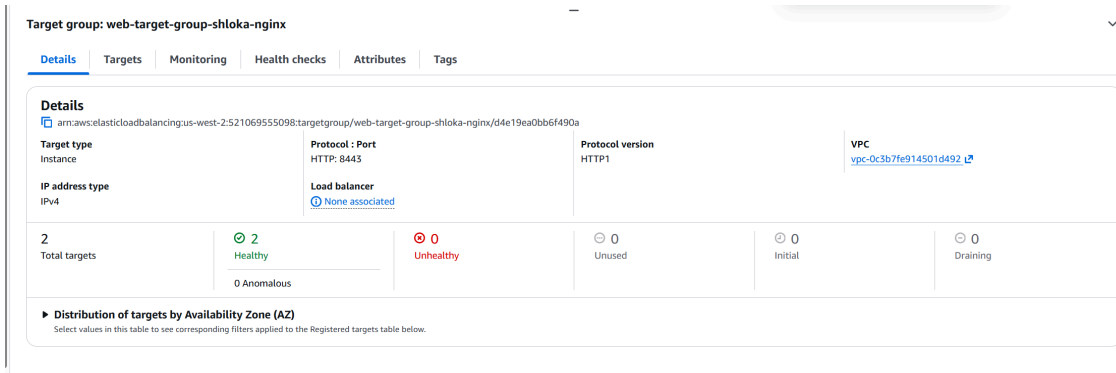
- Allow 8443 **only** from ALB SG
- Allow SSH **only** from Bastion SG



Step 3: Create Target Group & Load Balancer

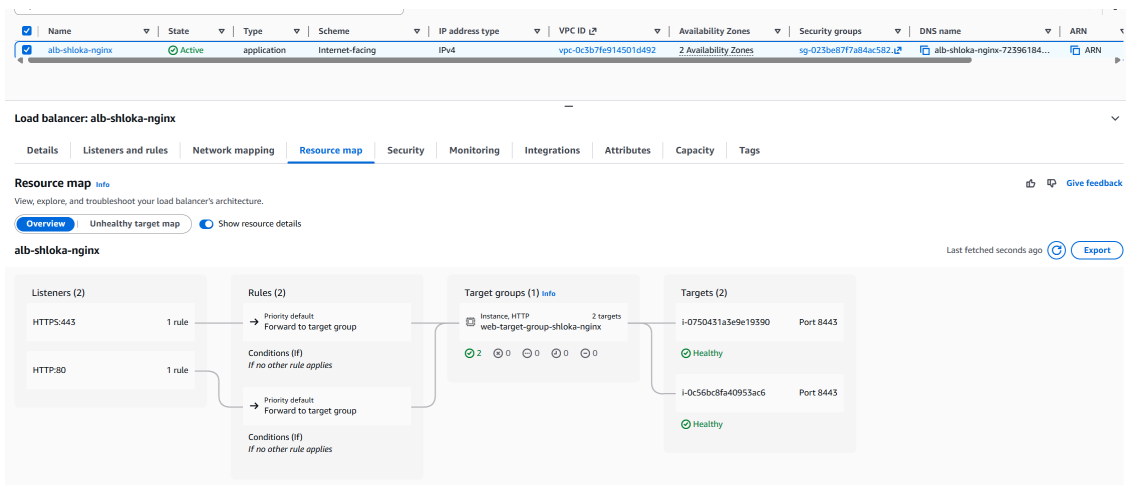
- **Target Group**

- Listens on Port 8443
- Associated with 2 instances (application servers)



2. ALB

- In both the public subnets
- Listener: 80 → forward to TG
- Later updated to 443 for HTTPS



Step 4: Configure Route 53 DNS

- Updated GoDaddy nameservers to Route 53's nameservers

- Created Hosted Zone in route 53
- Created A (Alias) record to ALB's DNS record

Reason: Required for domain → ALB routing.

The screenshot shows the AWS Route 53 console for the hosted zone 'shlokamdar.in'. The 'Hosted zone details' section is visible, along with a list of records. The records table is as follows:

Record name	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL (s...)	Health ...	Evalua...	Recor...
shlokamdar.in	A	Simple	-	Yes	dualstack.alb-shloka-nginx-7...	-	-	Yes	-
shlokamdar.in	NS	Simple	-	No	ns-849.awsdns-42.net. ns-1084.awsdns-07.org. ns-1546.awsdns-01.co.uk. ns-168.awsdns-21.com.	172800	-	-	-
shlokamdar.in	SOA	Simple	-	No	ns-849.awsdns-42.net. awsd...	900	-	-	-
_c447e143e23...	CNAME	Simple	-	No	_f47962c3cee8f6881202b16...	300	-	-	-

Step 5: HTTPS Setup (ACM Certificate)

- Requested certificate for:
 - shlokamdar.in

The screenshot shows the AWS Certificate Manager console. A single certificate is listed with the following details:

Certificate ID	Domain name	Type	Status	In use	Renewal eligibility	Key algorithm
2f75b35d-ec51-451d-864d-0343b0337c99	shlokamdar.in	Amazon Issued	Issued	Yes	Eligible	RSA 2048

- DNS validation (took time due to propagation)
- Attached certificate to ALB Listener on port 443

Result → **Secure HTTPS website working**