

# 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

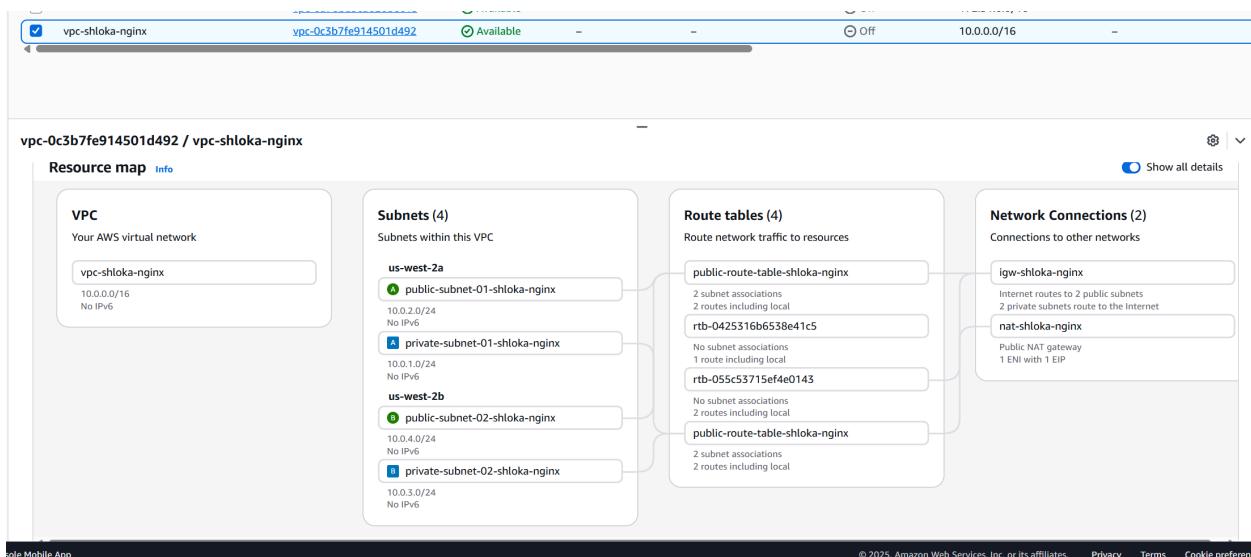
- Created a NAT Gateway: **nat-shloka-nginx**
- **Reason:** Private EC2 could not install packages → user-data failed. NAT fixed this.

### 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 :

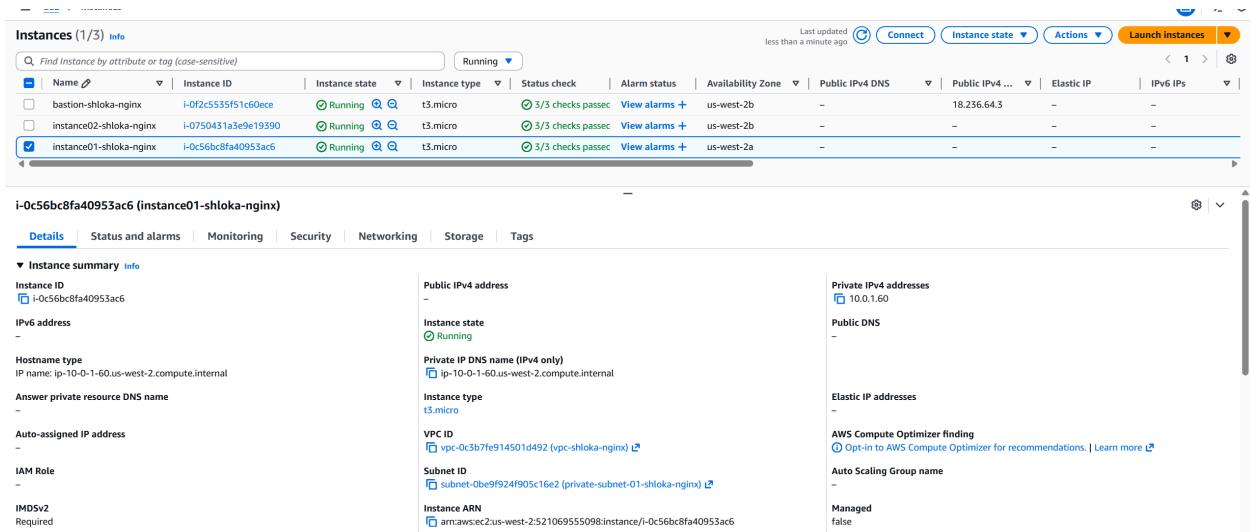
```
#!/bin/bash
sudo apt update -y
sudo apt install -y nginx
```

```
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



## Step 3: Configure Security Groups

## 1. Bastion SG:

- SSH (22): My IP only

The screenshot shows the AWS Management Console interface for managing security group rules. The specific security group is named "sg-097b6ff7e7c5911df - bastion-SG-shloka-nginx". The "Inbound rules" tab is selected, showing one rule entry:

Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0b61a9de9ee4b26ec	IPv4	HTTP	TCP	80	103.178.142.43/32

## 2. ALB SG:

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

The screenshot shows the AWS Management Console interface for managing security group rules. The specific security group is named "sg-023be87f7a84ac582 - ALB-SG-shloka-nginx". The "Inbound rules" tab is selected, showing two rule entries:

Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-070f50125bac3bdb	IPv4	HTTPS	TCP	443	0.0.0.0/0
sgr-0719c7b9ca181ad73	IPv4	HTTP	TCP	80	0.0.0.0/0

## 3. Private EC2 SG:

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

The screenshot shows the AWS Management Console interface for managing security group rules. The specific security group is named "sg-04bc00e774c8964cb - web-SG-shloka-nginx". The "Inbound rules" tab is selected, showing two rule entries:

Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0fcfc9d4da60118	-	SSH	TCP	22	<a href="#">sg-097b6ff7e7c5911df / bastion-SG-shloka-nginx</a>
sgr-06287598dcbeb66c	-	Custom TCP	TCP	8443	<a href="#">sg-023be87f7a84ac582 / ALB-SG-shloka-nginx</a>

## Step 3: Create Target Group & Load Balancer

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

The screenshot shows the AWS CloudWatch Metrics Insights interface. A query is being constructed to search for CloudWatch Metrics Insights metrics. The query includes filters for 'CloudWatch Metrics Insights Metrics' and 'CloudWatch Metrics Insights Metrics'. It also specifies a time range from 'Last hour' to 'Last 24 hours' and a metric name of 'CloudWatch Metrics Insights Metrics'. The results table is currently empty.

## 2. ALB

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

The screenshot shows the AWS CloudWatch Metrics Insights interface. A query is being constructed to search for CloudWatch Metrics Insights metrics. The query includes filters for 'CloudWatch Metrics Insights Metrics' and 'CloudWatch Metrics Insights Metrics'. It also specifies a time range from 'Last hour' to 'Last 24 hours' and a metric name of 'CloudWatch Metrics Insights Metrics'. The results table is currently empty.

## 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.

Record name	Type	Routing policy	Alias	Value/Route traffic to	TTL (s)	Health check	Evaluate health	Records
shlokamdar.in	A	Simple	-	Yes	dualstack.alb-shlokamdar-nginx-7...	-	-	1
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	-	1
shlokamdar.in	SOA	Simple	-	No	ns-849.awsdns-42.net.awsd...	900	-	1
_c447e143e23...	CNAME	Simple	-	No	_f47962c3cee8f6881202b16...	300	-	1

## Step 5: HTTPS Setup (ACM Certificate)

- Requested certificate for:
  - shlokamdar.in

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**