DevOps Engineer Technical Assessment

Objective

The objective was to deploy simple web application securely "Hello World" on AWS using VPC, private EC2 instances with Nginx as service, and Application Load Balancer(ALB). Everything done with best practices with no direct internet access to your servers Using AWS console.

Architecture Diagram

- **Traffic flow**

  Internet → ALB → EC2 (Private)

- **VPC (10.0.0.0/16)**

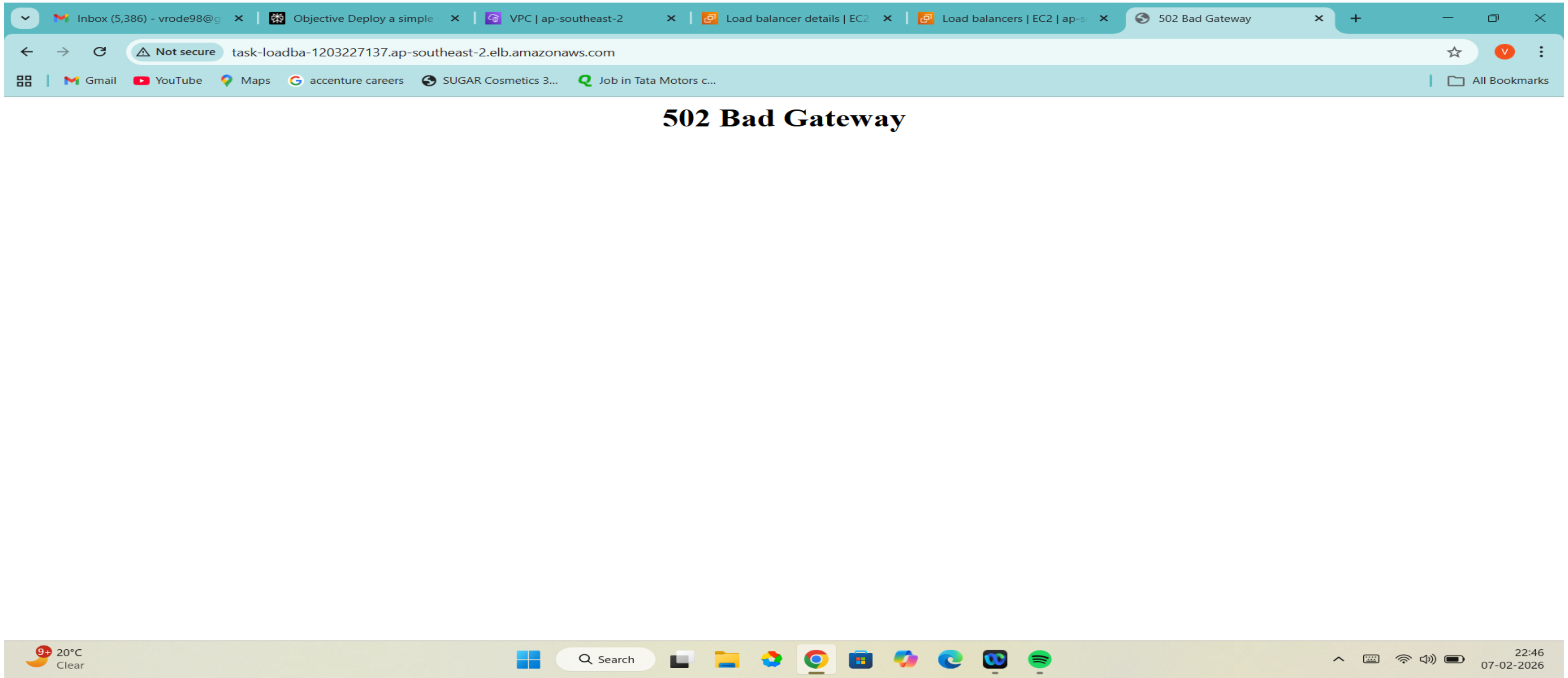
2 Public Subnets

- Public Subnet 1 (AZ-a) – 10.0.1.0/24
- Public Subnet 2 (AZ-b) – 10.0.2.0/24
- Application Load Balancer

2 Private Subnets

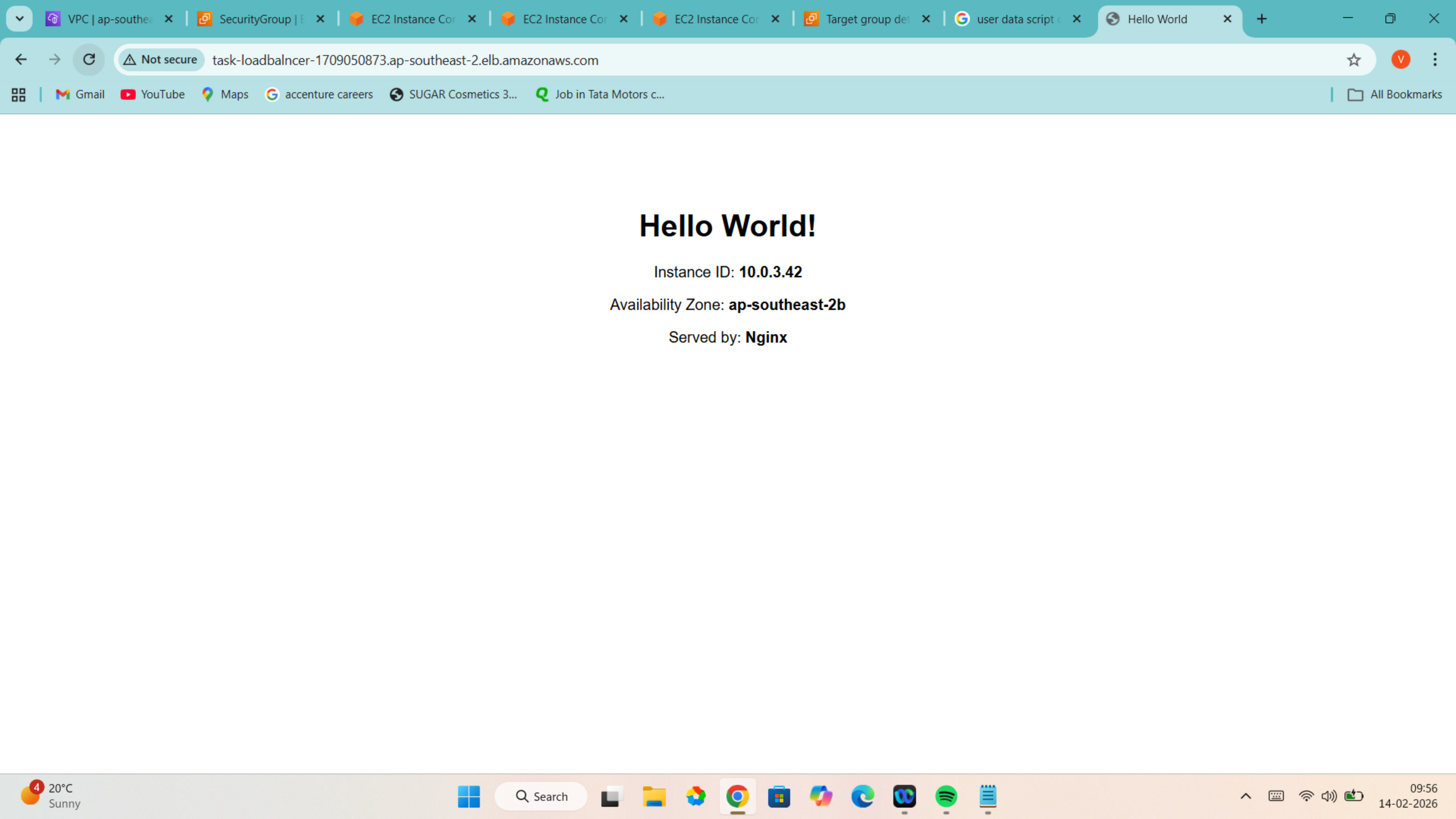
- Private Subnet 1 (AZ-a) – 10.0.3.0/24 → EC2 + Nginx
- Private Subnet 2 (AZ-b) – 10.0.4.0/24 → EC2 + Nginx

Working Application



502 Bad Gateway

As working of loadbancer URL, it showing 502 Bad Gateway means alb is not getting nginx. It will resolve

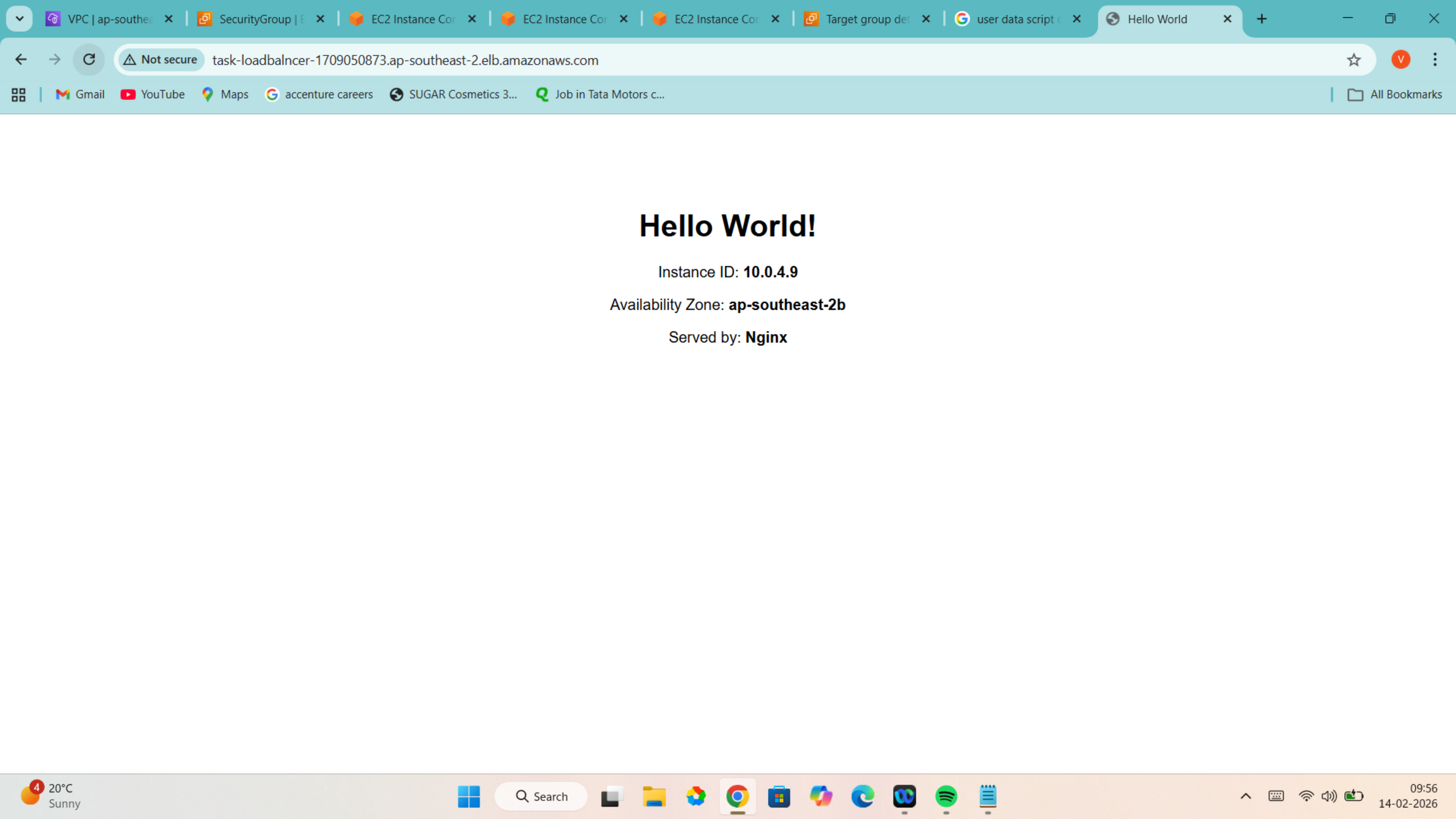


Hello World!

Instance ID: **10.0.3.42**

Availability Zone: **ap-southeast-2b**

Served by: **Nginx**



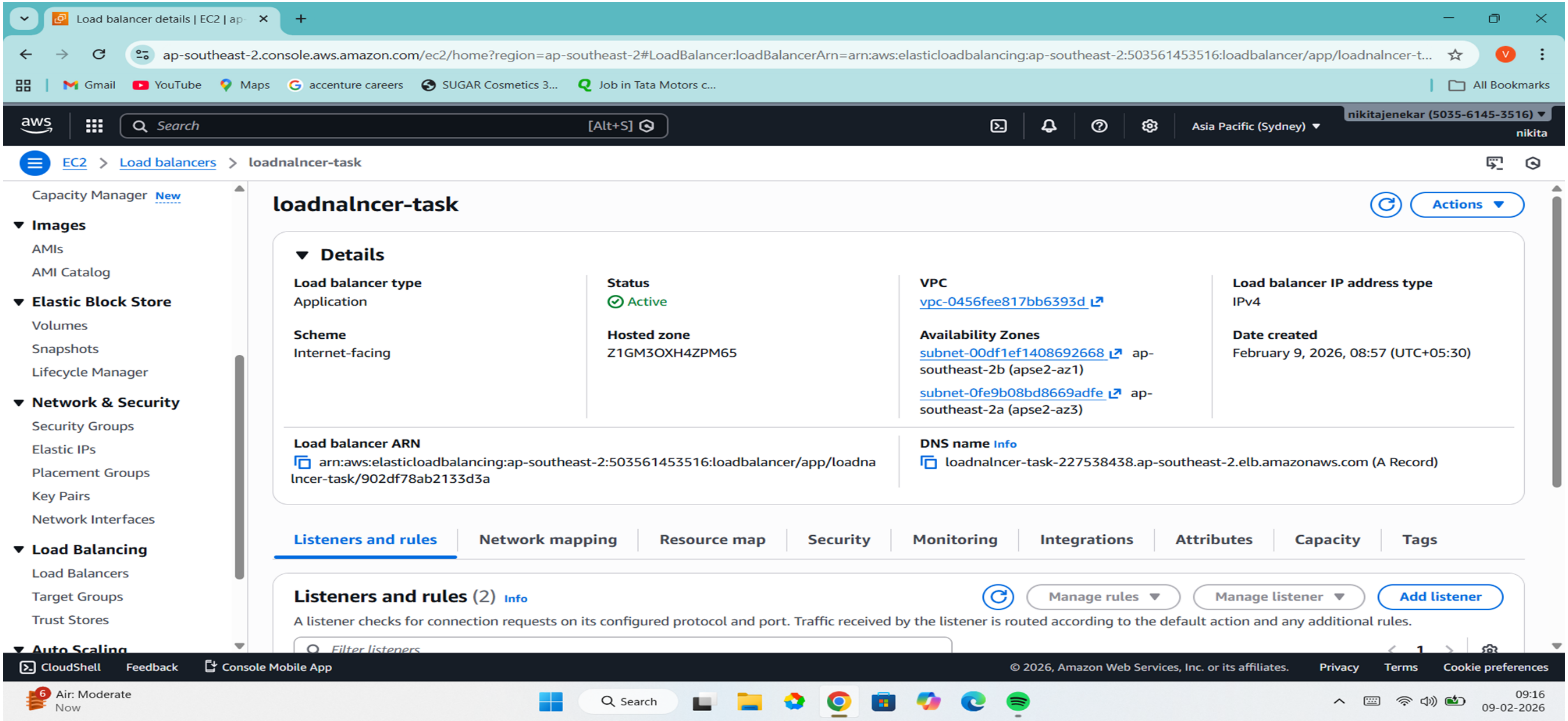
Hello World!

Instance ID: **10.0.4.9**

Availability Zone: **ap-southeast-2b**

Served by: **Nginx**

Load balancer URL



loadnalncer-task

Details

Load balancer type
Application

Status
Active

VPC
vpc-0456fee817bb6393d

Load balancer IP address type
IPv4

Scheme
Internet-facing

Hosted zone
Z1GM3OXH4ZPM65

Availability Zones
subnet-00df1ef1408692668 ap-southeast-2b (apse2-az1)
subnet-0fe9b08bd8669adfe ap-southeast-2a (apse2-az3)

Date created
February 9, 2026, 08:57 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:ap-southeast-2:503561453516:loadbalancer/app/loadnalncer-task/902df78ab2133d3a

DNS name Info
loadnalncer-task-227538438.ap-southeast-2.elb.amazonaws.com (A Record)

Listeners and rules (2) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Step-by-step guide to perform task

- I login to aws console with help of login credentials and went to VPC dashboard to create vpc, create VPC → Resources to create - Vpc only → give name – (Task-01) to identify easily. Set IPv4 CIDR – 10.0.0.0/16 → create vpc and vpc created.
- Now create 2 public and 2 private subnet such that 1 public & 1 private in same availability zone and another 2 in different availability zone.

Create subnet → select Vpc Id (to Create subnets in this VPC that created) → give name as public-01 → select availability zone to where this subnet will reside (ap-southeast-2a) → IPv4 subnet CIDR block – 10.0.1.0/24 then go to create subnet ,subnet is created.

Follow the same procedure for creating subnet for,

Public-02—(ap-southeast-2b) –(10.0.2.0/24)

Private-01—(ap-southeast-2a) –(10.0.3.0/24)

Private-02 –(ap-southeast-2a) –(10.0.4.0/24)

- Now create internet gateway to enables communication between VPC and the internet. Create internet gateway → name for gateway – (gateway-task) → Create internet gateway. Now select the gateway and go to action and attach to the vpc for communication.
- Create route table to determine where network traffic from your subnet or gateway is directed.

Go to create route table → give name (rt-public) → select the vpc to connect for this route table → create route table → select public1 subnet go to → Actions to edit subnet association → select subnet save association. And save it.

Now in route table → go routes → edit route → select destination (0.0.0.0/0) → target (internet gateway) → select gateway Id → save changes.

- Do same for rt-private.

- Now create security groups to act as a virtual firewall that controls the traffic for one or more instances.

SG Name	Inbound Rules	Purpose
loaderbalnce-sg	HTTP 80 from 0.0.0.0/0	ALB (internet-facing)
ec2-sg	HTTP 80 from loaderbalnce-sg only	EC2 (ALB → EC2 only)
VPC-sg	HTTP 80 from vpc i.d	Ec2 (vpc → private instance)

Launch EC2 Instances (Private Subnets)

- Now launch instance Give name (Private-01) select AMI: Amazon Linux 2023
→ Instance type: t2.micro → Network: VPC as created (task-01) → Subnet: Private-01
→Auto-assign public IP: Disabled →Security group: ec2-sg and vpc-sg
Now add user data script to host desired page
Below are the script,

```
#!/bin/bash
yum update -y
yum install -y nginx

# Start and enable nginx
systemctl start nginx
systemctl enable nginx
```

Now launch instance tab and instance will be created with running status.
Take SSH with help of endpoint as instance is in private state
Do changes in nginx config. File index.html by taking nginx path /usr/share/nginx/html
Vim index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Hello World</title>
```

- Now create loadbalancer so give name as loadbalnce-task → select segurity groups for ALB (internet-facing) → add target groups for this we have to create target group and attach to loadbalncer. Then go to register ,select both instance and create target group.
- It will take some time to active ate the status of loadbalncer after showing active copy the DNs of loadbalncer and hit in new tab to host
- Finally it will show the desired web page

But in my case, I apologise that error has occurs showing 502 Bad Gateway , means nginx is not getting request from ALB and I will resolve it update soon.

As describe above by taking loadblancer DNS, got the desired result that is

By ping got request to private instance1

By ping again getting private instance 2

List of AWS Resources Created

Resource Type	Count	Names/IDs Example
VPC	1	Task-01
Subnets	4	public-01 public-02 private-01 private-02
Internet Gateway	1	Gateway-task
Route Tables	2	public-rt, private-rt
Security Groups	3	loaderbalnce-sg ec2-sg Vpc-sg
EC2 Instances	2	Private-01 Private-02
ALB	1	Loadnancer-task
Target Group	1	Tg-task
Endpoint	1	Endpoint for connecting to ssh
Nat-gateway	1	Gateway-task

Security group configurations

SG Name	Inbound Rules	Purpose
Loadnancer-task	HTTP 80 from 0.0.0.0/0	ALB (internet-facing)
ec2-sg	HTTP 80 from Loadnancer-task only	EC2 (ALB → EC2 only)
VPC-sg	HTTP 80 from vpc i.d	Ec2 (vpc → private instance)

Configuration Files

Nginx Configuration Files

```
#!/bin/bash
yum update -y
yum install -y nginx
# Start and enable nginx
systemctl start nginx
systemctl enable nginx
```

HTML page code

```html
<!DOCTYPE html>
<html>
<head>
    <title>Hello World</title>
</head>
<body style="text-align: center; font-family: Arial; margin-top: 100px;">
    <h1>Hello World!</h1>
    <p>Instance ID: <strong>i-1234567890abcdef0</strong></p>
    <p>Availability Zone: <strong>us-east-1a</strong></p>
    <p>Served by: <strong>Nginx</strong></p>
</body>
</html>
```

Thank You