**Task Description:**

1. **Create a S3 bucket, with no public access and upload files to the bucket & view the logs using cloudwatch for the uploaded files.**

***SOLUTION:***

**Step 1: Create an S3 Bucket with No Public Access**

1. **Navigate to the S3 Service**:
   * In the AWS Management Console, search for and select **S3**.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Create the S3 Bucket**:
   * Click **Create bucket**.
   * Choose a globally unique name for your bucket.
   * Choose a region for your bucket.
   * **Uncheck** "Block all public access" (to keep it private).
     + Make sure **Block all public access** is enabled, which ensures that no one can publicly access your bucket.
   * Leave the rest of the settings at their default values or adjust them according to your preferences.
   * Click **Create** to create the bucket.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 2: Set Up CloudWatch Logging for S3**

1. **Enable CloudTrail** (for logging S3 events):
   * Navigate to **CloudTrail** in the AWS Management Console.
   * Create a new **Trail** (or use an existing one).
   * Ensure that the trail is set up to log **S3 data events**. This will capture any API operations related to your S3 bucket.
   * Choose the option to send logs to **CloudWatch Logs**.
   * Specify an existing or new CloudWatch log group.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Configure Bucket Logging** (optional, for detailed S3 access logs):
   * Navigate to your **S3 bucket** in the AWS Console.
   * Go to the **Properties** tab.
   * Under **Server access logging**, click **Edit** and enable logging.
   * Choose a target bucket where logs will be stored (this could be the same or a different bucket).

A screenshot of a computer

AI-generated content may be incorrect.

**Step 3: Upload Files to the S3 Bucket**

1. Go to your **S3 bucket**.
2. Click **Upload**.
3. Select the files you want to upload from your local machine.
4. Click **Upload** to start the process.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4: View the Logs in CloudWatch**

1. **Navigate to CloudWatch**:
   * In the AWS Management Console, go to **CloudWatch**.

A screenshot of a computer

AI-generated content may be incorrect.

1. **View Logs in CloudWatch**:
   * In the CloudWatch console, click on **Logs** in the left sidebar.
   * You should see the log group that is associated with your **CloudTrail** or S3 logging.
   * Select the appropriate log group.
   * You can now see detailed logs for the file uploads to the S3 bucket.

A white box with black text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 5: Check CloudTrail for S3 Events**

1. **Navigate to CloudTrail**:
   * Go to **CloudTrail** in the AWS Console.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Search for Events**:
   * Use the **Event History** to filter and search for S3-related events, such as **PutObject** or **GetObject** events.
   * You can filter by the S3 bucket name to view only the logs relevant to your specific bucket.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Launch two ec2-instances and connect it to a application load balancer, where the output traffic from the server must be an load balancer IP address**

***SOLUTION:***

**Launch Two EC2 Instances:**1. In EC2 instances launch 2 instances with Bash script in the additional column ( While creating the instance) in order to automate the server to launch a website automatically.

Bash Script for server1:

#!/bin/bash

sudo apt update -y

sudo apt install apache2 unzip -y

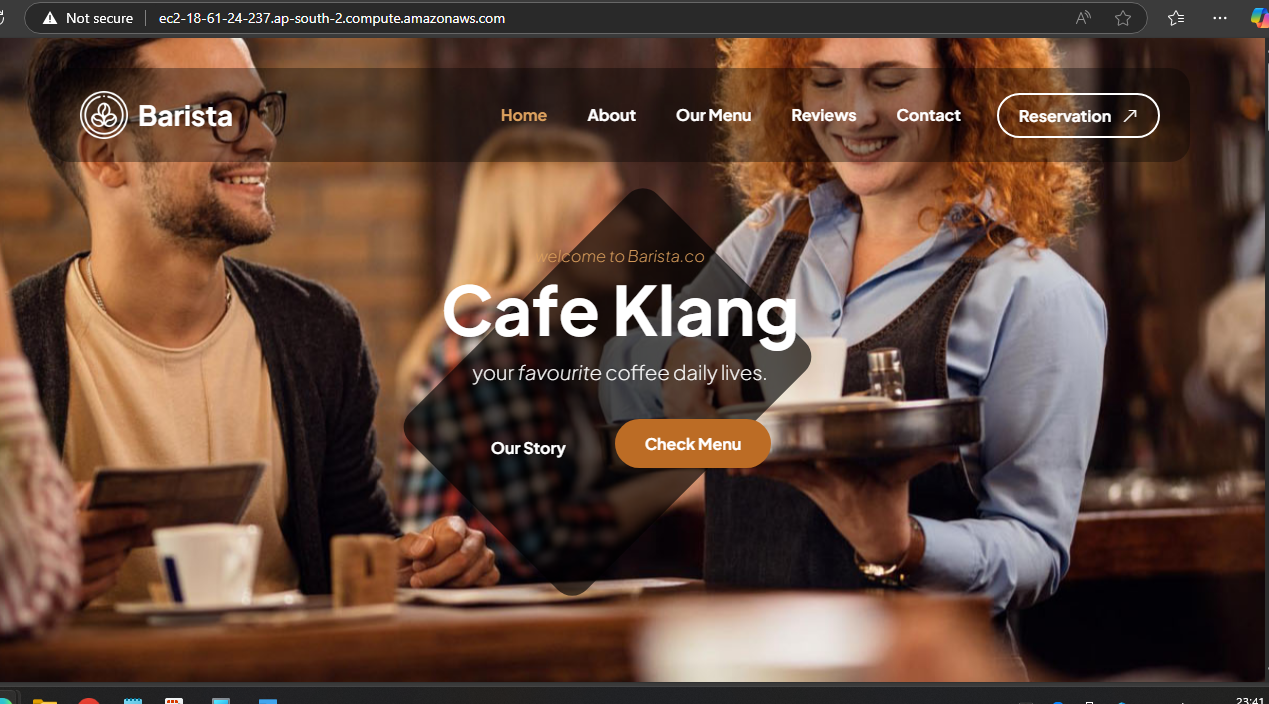
sudo systemctl status apache2

sudo cd /tmp

sudo wget https://www.tooplate.com/zip-templates/2137\_barista\_cafe.zip

sudo unzip 2137\_barista\_cafe.zip

sudo cp -R 2137\_barista\_cafe/\* /var/www/html/



Bash Script for server 2:

#!/bin/bash

sudo apt update -y

sudo apt install apache2 unzip -y

sudo systemctl status apache2

sudo cd /tmp

sudo wget https://www.tooplate.com/zip-templates/2136\_kool\_form\_pack.zip

sudo unzip 2136\_kool\_form\_pack.zip

sudo cp -R 2136\_kool\_form\_pack/\* /var/www/html/

A screenshot of a computer

AI-generated content may be incorrect.

**Two Instances:**

A screenshot of a computer

AI-generated content may be incorrect.

**Create an Application Load Balancer (ALB):**

1. **Create a Target Group:**

* In the Target Group section, select Instances as the target type.
* Choose HTTP as the protocol and the port 80 (or 443 if using HTTPS).
* Choose a health check path (e.g., /).
* Add your two EC2 instances to the target group (using their private IPs).

A computer screen shot of a browser window

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Create the ALB:**

1. In the EC2 Console, navigate to Load Balancers and click on Create Load Balancer.
2. Select Application Load Balancer.
3. Enter a name for the load balancer.
4. Set the Scheme to Internet-facing (since the ALB needs to handle traffic from the internet).
5. Under Listeners, add HTTP (port 80) or HTTPS (port 443).
6. Select the VPC and public subnet where the ALB will be deployed.
7. Create or select a Security Group that allows inbound traffic on port 80 or 443.
8. And select the Target Group (which consist the Instances)

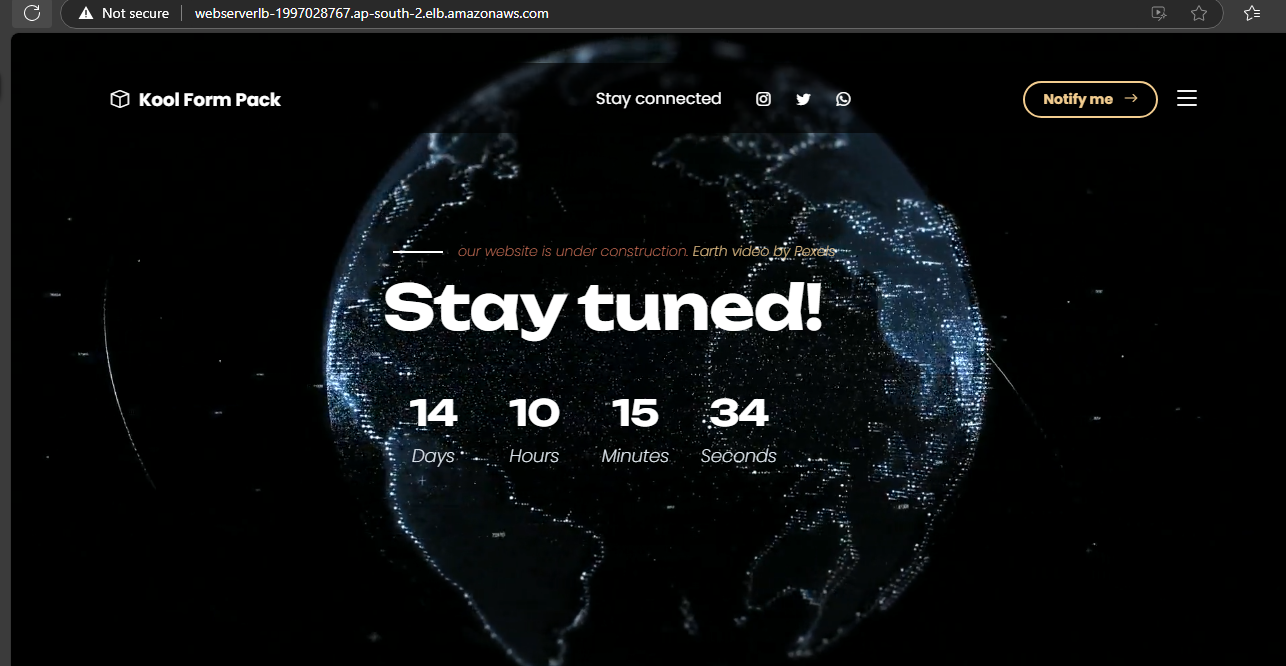
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

DNS NAME: webserverlb-1997028767.ap-south-2.elb.amazonaws.com

 A screenshot of a computer

AI-generated content may be incorrect.

***Result:*** with same DNS using Load balancer we can navigate to the one web server to other by respective to the load to the servers.