



**RMIT International University Vietnam
Assignment 2 (AWS cloud implementation)**

Subject Code:	COSC2930
Subject Name:	Cloud Architecting
Location & Campus	SGS Campus
Student name:	Nguyen Quoc An
Student Number:	s3938278
Lecturer's name:	Dr. Nguyen Minh Long

"I declare that in submitting all work for this assessment I have read, understood and agreed to the content and expectations of the Assessment Declaration."

Task A - Static website with S3

1. Create S3 bucket and host a static website

Create S3 bucket.

The screenshot shows the 'Create bucket' configuration page in the AWS S3 console. In the 'General configuration' section, the bucket name is set to 'momandpopcafe.shop' and the AWS Region is set to 'US East (N. Virginia) us-east-1'. Under 'Object Ownership', the 'ACLs enabled' option is selected, and a note states: 'We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.' In the 'Object Ownership' section, the 'Bucket owner preferred' option is selected. At the bottom, there are 'CloudShell' and 'Feedback' buttons, and a copyright notice for 2023, Amazon Web Services, Inc. or its affiliates.

The screenshot shows the 'Block Public Access settings for this bucket' configuration page. It lists four settings: 'Block all public access', 'Block public access to buckets and objects granted through new access control lists (ACLS)', 'Block public access to buckets and objects granted through any access control lists (ACLS)', and 'Block public access to buckets and objects granted through new public bucket or access point policies'. The 'Block all public access' checkbox is checked. A note below states: 'Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.' Below this, a checkbox for acknowledging the risk is checked. The 'Bucket Versioning' section is also visible, with the 'Enable' option selected. The page includes standard AWS navigation and footer elements.

Upload source code

The screenshot shows the AWS S3 console interface for uploading files to a bucket named "momandpopcafe.shop". The "Upload" page displays a list of files and folders to be uploaded, including "index.html", "styles.css", and various image files like "Cafe-Owners.png" and "Cookies.png". The destination is set to "s3://momandpopcafe.shop".

Name	Folder	Type	Size
index.html	-	text/html	2.9 KB
styles.css	css/	text/css	541.0 B
Cafe-Owners.png	images/	image/png	2.7 MB
Coke-Vitrine.png	images/	image/png	3.8 MB
Coffee-and-Pastries...	images/	image/png	3.1 MB
Coffee-Shop.png	images/	image/png	726.8 KB
Cookies.png	images/	image/png	1.4 MB
Cup-of-Hot-Chocola...	images/	image/png	3.6 MB
Strawberry-&-Blueb...	images/	image/png	2.9 MB
Strawberry-Tarts.png	images/	image/png	3.4 MB

Enable static website hosting

The screenshot shows the "Edit static website hosting" configuration page for the same bucket. It enables static website hosting and selects "Host a static website" as the hosting type. The "Index document" is set to "index.html" and the "Error document" is set to "error.html".

Make public via ACL.

The make public action enables public read access in the object access control list (ACL) settings. [Learn more](#)

Specified objects

Name	Type	Last modified	Size
css/	Folder	-	-
images/	Folder	-	-
index.html	html	September 24, 2023, 09:54:26 (UTC+07:00)	2.9 KB

Cancel **Make public**

The website is now accessible by the index.html file

2. Enable bucket versioning, and trigger an SNS topic

Enable versioning when creating the bucket, or in bucket properties.

Successfully edited public access
View details below.

Amazon S3 > Buckets > momandpopcafe.shop

momandpopcafe.shop [Info](#)

Objects **Properties** Permissions Metrics Management Access Points

Bucket overview

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) arn:aws:s3:::momandpopcafe.shop	Creation date September 24, 2023, 09:51:15 (UTC+07:00)
---	---	---

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.
[Learn more](#) **Edit**

Bucket Versioning
Enabled
Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

Tags (0)
You can use bucket tags to track storage costs and organize buckets. [Learn more](#) **Edit**

Key	Value
No tags associated with this resource.	

Create an SNS topic for sending notifications to subscribed emails.

Create topic

Details

Type [Info](#)
Topic type cannot be modified after topic is created

FFO (first-in, first-out)
• Strictly-preserved message ordering
• Exactly-once message delivery
• High throughput, up to 300 publishes/second
• Subscription protocols: SQS

Standard
• Best-effort message ordering
• At-least once message delivery
• Highest throughput in publishes/second
• Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - optional [Info](#)
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

Maximum 100 characters.

► Encryption - optional [Info](#)
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

► Access policy - optional [Info](#)
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

► Data protection policy - optional [Info](#)
This policy defines which sensitive data to monitor and to prevent from being exchanged via your topic.

Add subscription email, and confirm subscription in email.

Important changes for sending text messages (SMS) to US destinations
US mobile carriers have recently changed their regulations, and will require that all toll-free numbers (TFNs) complete a registration process with a regulatory body before September 30, 2022. If you currently have a toll-free number you must register your toll-free number by September 30, 2022 or you will no longer be able to use the toll-free number. [Learn more](#)

[View origination numbers](#)

Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN

Protocol
The type of endpoint to subscribe

Endpoint
An email address that can receive notifications from Amazon SNS.

After your subscription is created, you must confirm it. [Info](#)

► Subscription filter policy - optional [Info](#)
This policy filters the messages that a subscriber receives.

► Redrive policy (dead-letter queue) - optional [Info](#)
Send undeliverable messages to a dead-letter queue.

[Cancel](#) [Create subscription](#)

Create an event notification:

Select:

s3:ObjectCreated:*

s3:ObjectRemoved:*

SNS topic: momandpop-cafe-s3-email-notification

Personal

Services Search [Alt+S]

Global vocabs/user2459378-Nguyen_Quoc_An @ 7743-7158-3665

General configuration

Event name: upload-delete-object-notification
Event name can contain up to 255 characters.

Prefix - optional: Limit the notifications to objects with key starting with specified characters.
images/

Suffix - optional: Limit the notifications to objects with key ending with specified characters.
.jpg

Event types
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

All object create events
s3:ObjectCreated:
 Put s3:ObjectCreated:Put
 Post s3:ObjectCreated:Post
 Copy s3:ObjectCreated:Copy
 Multipart upload completed s3:ObjectCreated:CompleteMultipartUpload

Object removal

All object removal events
s3:ObjectRemoved:
 Permanently deleted s3:ObjectRemoved:Delete
 Delete marker created

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 9:58 AM 24/09/2023

Come back to the SNS topic and edit the “Acess policy” to allow S3 bucket arn to publish messages.

Change:

“StringEquals” to “ArnLike”

“AWS:SourceOwner” to “AWS:SourceArn”

Number to “s3-bucket-arn”

Personal

Services Search [Alt+S]

N. Virginia vocabs/user2459378-Nguyen_Quoc_An @ 7743-7158-3665

Access policy - optional Info This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

JSON editor

```

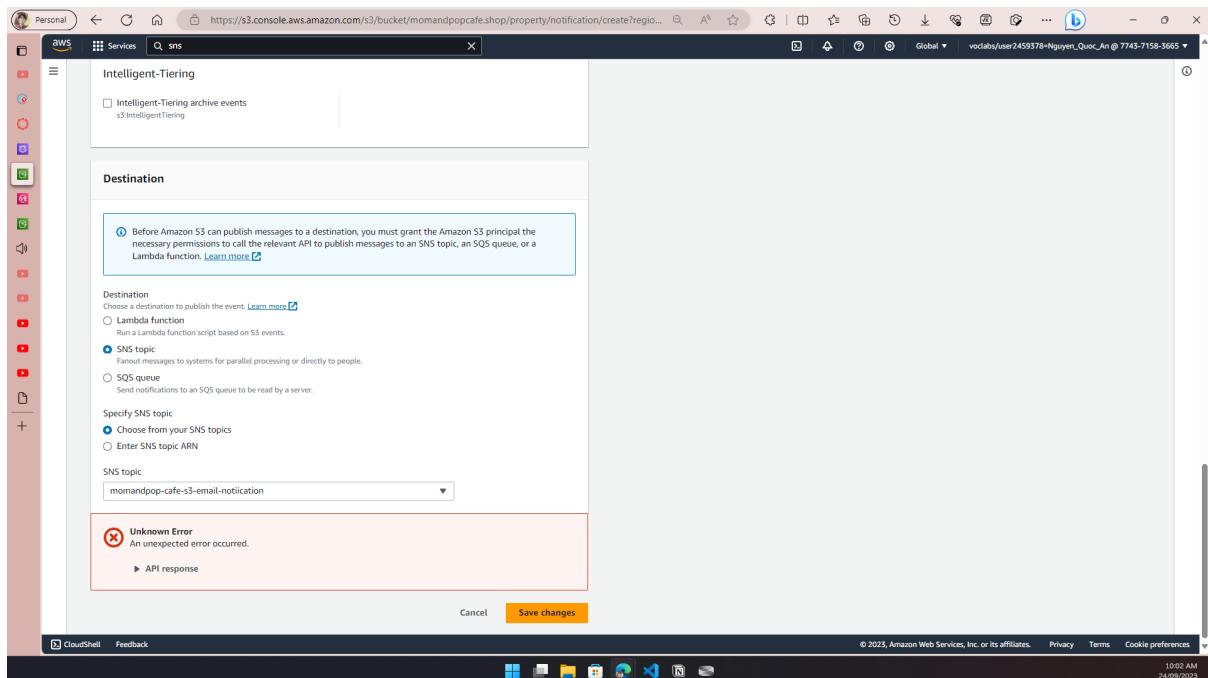
11     "Action": [
12       "SNS:Publish",
13       "SNS:RemovePermission",
14       "SNS:SetTopicAttributes",
15       "SNS:DeleteTopic",
16       "SNS:ListSubscriptionsByTopic",
17       "SNS:GetTopicAttributes",
18       "SNS:AddPermission",
19       "SNS:Subscribe"
20     ],
21     "Resource": "arn:aws:sns:us-east-1:774371583665:momandpop-cafe-s3-email-notification",
22     "Condition": {
23       "ArnLike": {
24         "AWS:SourceArn": "arn:aws:s3:::momandpopcafe.shop"
25       }
26     }
  
```

Data protection policy - optional Info This policy defines which sensitive data to monitor and to prevent from being exchanged via your topic.

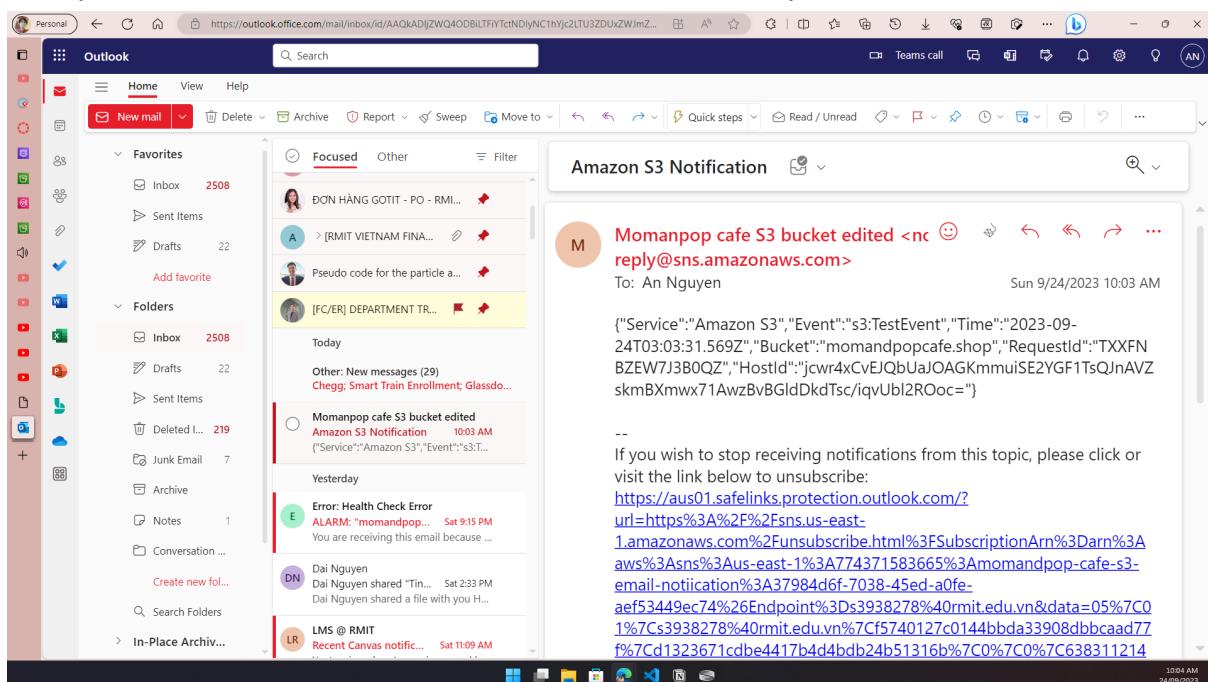
Delivery policy (HTTP/S) - optional Info The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section.

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 10:02 AM 24/09/2023

The destination is the previously created SNS topic.



Now you will receive an email notification when a file is uploaded or deleted.



3. Set life-cycle policy

In the bucket setting, go to "Management" and create a life cycle rule.

Now the file when deleted, will be moved to to Standard-IA storage class in 30 days, One Zone-IA in 90 days, and Glacier Deep Archive in 365 days.

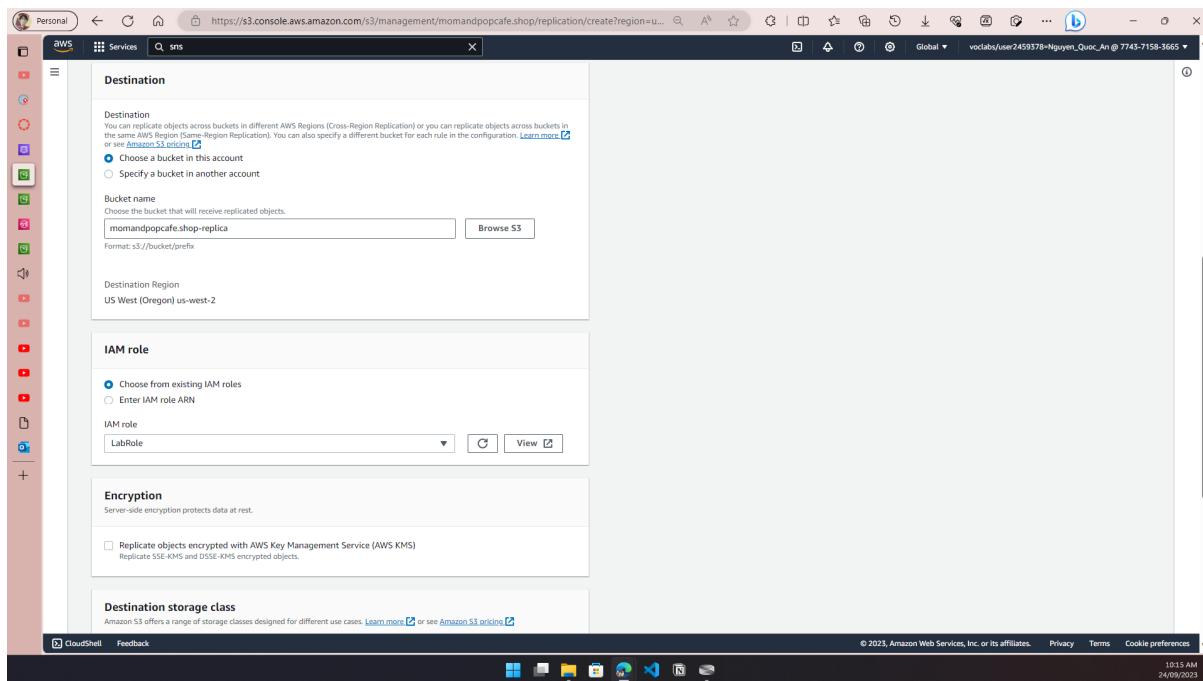
4. Enable cross-region replication

Create another S3 bucket in another region for backup, and enable bucket versioning.

The screenshot shows the 'Create bucket' configuration page in the AWS S3 console. The 'General configuration' section includes fields for 'Bucket name' (set to 'momandpopcafe.shop-replica') and 'AWS Region' (set to 'US West (Oregon) us-west-2'). Below these, there's a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button. The 'Object Ownership' section shows 'ACLs disabled (recommended)' selected, with a note that objects are owned by the account. The 'Block Public Access settings for this bucket' section indicates that public access is granted through ACLs, bucket policies, access point policies, or all. The bottom of the page includes standard AWS navigation links like CloudShell and Feedback.

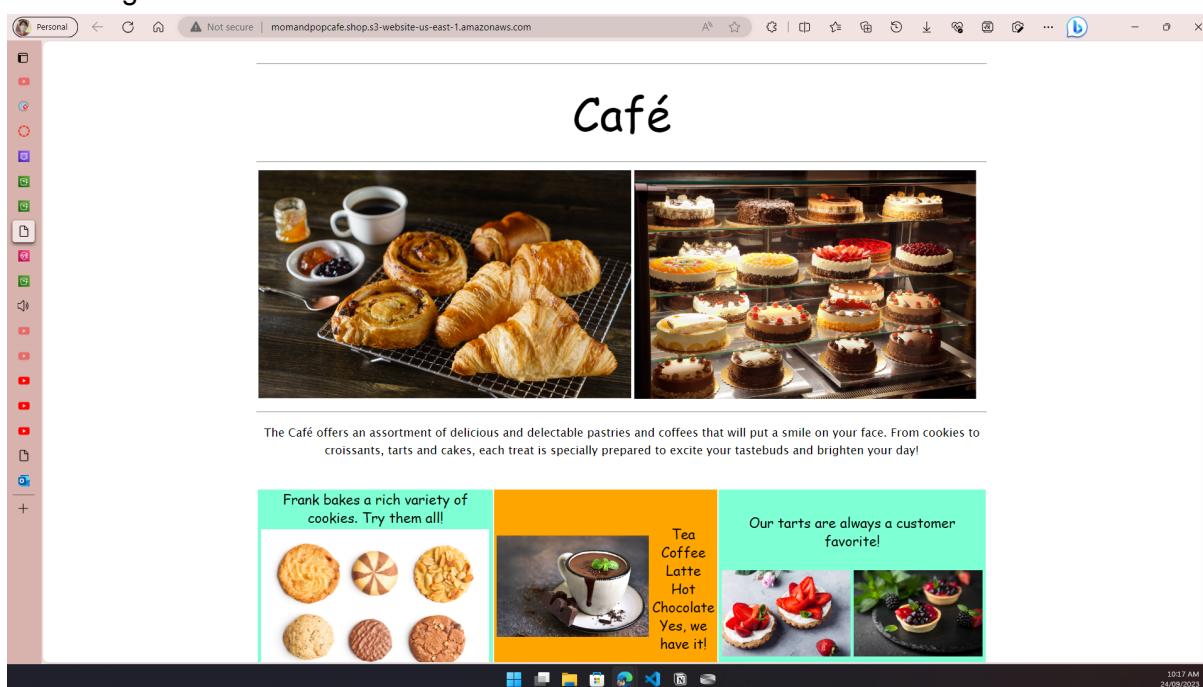
In the “Management” tab of the source bucket, create a replication rule with the destination is the replica bucket and LabRole for IAM role.

The screenshot shows the 'Create replication rule' configuration page in the AWS S3 console. The 'Replication rule configuration' section includes a 'Replication rule name' field set to 'momandpopcafe.shop'. Under 'Status', 'Enabled' is selected. In the 'Priority' section, the value '1' is entered. The 'Source bucket' section shows the source bucket is 'momandpopcafe.shop' in 'US East (N. Virginia) us-east-1'. The 'Destination' section is currently empty. The bottom of the page includes standard AWS navigation links like CloudShell and Feedback.



Upload the source code to the replica bucket to have the current file synchronize.
Now all the file updates, uploads will be replicated to the replica bucket.

Final image of the website:

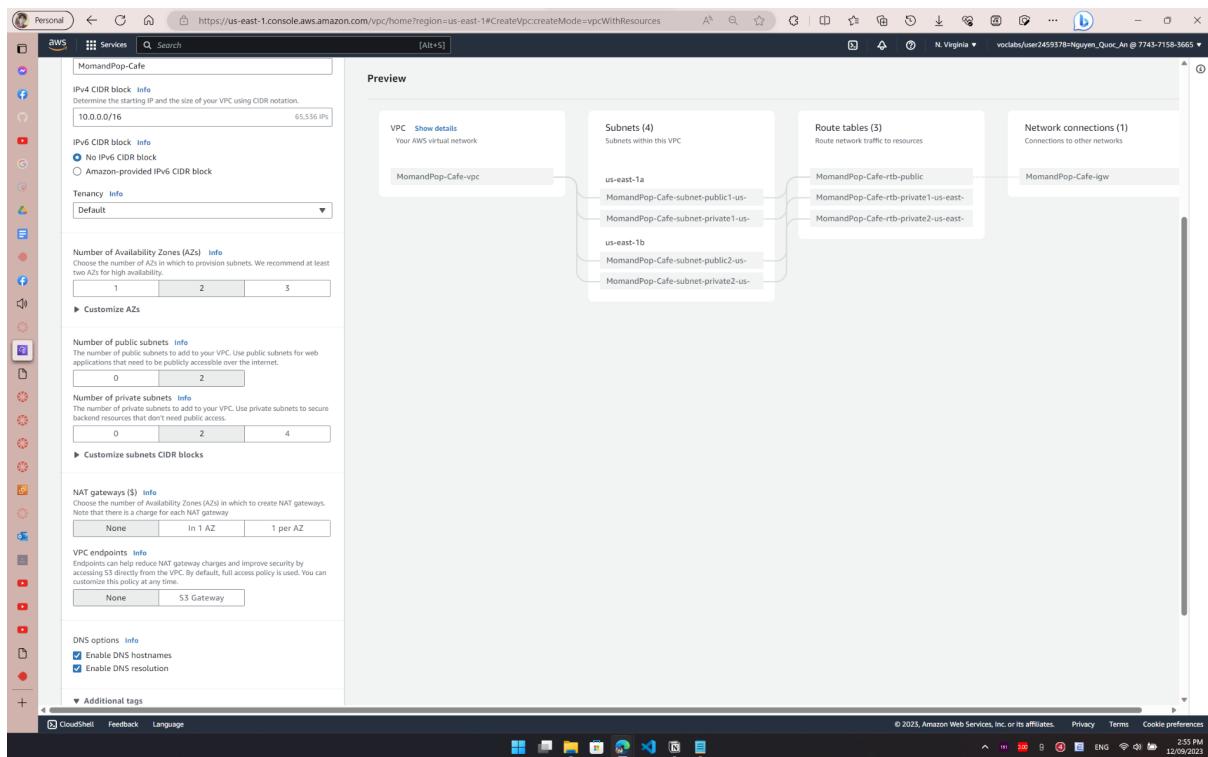


Task B - Highly available Dynamic website

1. Create network environment to launch web server

Create a VPC named MomandPop-Cafe-vpc with:

- 2 public subnets attached to an internet gateway
- 2 private subnets



Create a Security Group for the free tier EC2 instance to host the web server:

Name: MomandPop-Cafe-Web-Server-SG

Inbound Rule: Allow HTTP, HTTPS, and SSH request from 0.0.0.0/0

Outbound Rule: Allow all traffic to 0.0.0.0/0

The screenshot shows the AWS Security Groups creation wizard at the 'Create security group' step. In the 'Basic details' section, the security group name is 'MomandPop-Cafe-Web-Server-SG' and the description is 'Allows HTTP/HTTPS, and SSH traffic from everywhere'. In the 'VPC Info' section, the VPC is set to 'vpc-0c157a92e2d75595f'. Under 'Inbound rules', three rules are defined: one for HTTP (TCP port 80, source 0.0.0.0/0), one for HTTPS (TCP port 443, source 0.0.0.0/0), and one for SSH (TCP port 22, source 0.0.0.0/0). An 'Add rule' button is available. Under 'Outbound rules', there is a table with a single row: Destination 0.0.0.0/0 and Description - optional info.

2. Host website in EC2 instance

Launch a free tier EC2 instance, with the following configuration:

Name: MomandPop-Cafe-Web-Server

AMI: Amazon Linux 2 AMI

Instance type: t2.micro

Key par: vockey

VPC: MomandPop-Cafe-vpc

Subnet: MomandPop-Cafe-subnet-public1-us-east-1a

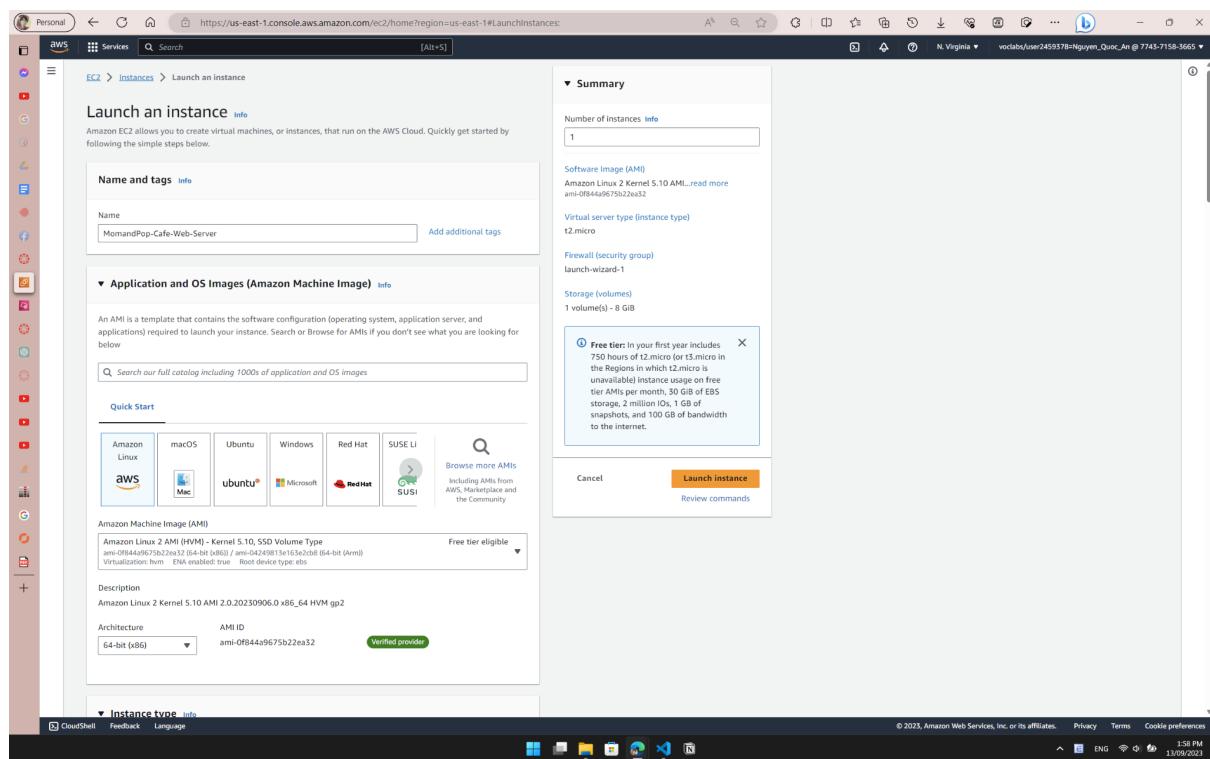
Auto-assign public IP: Enable

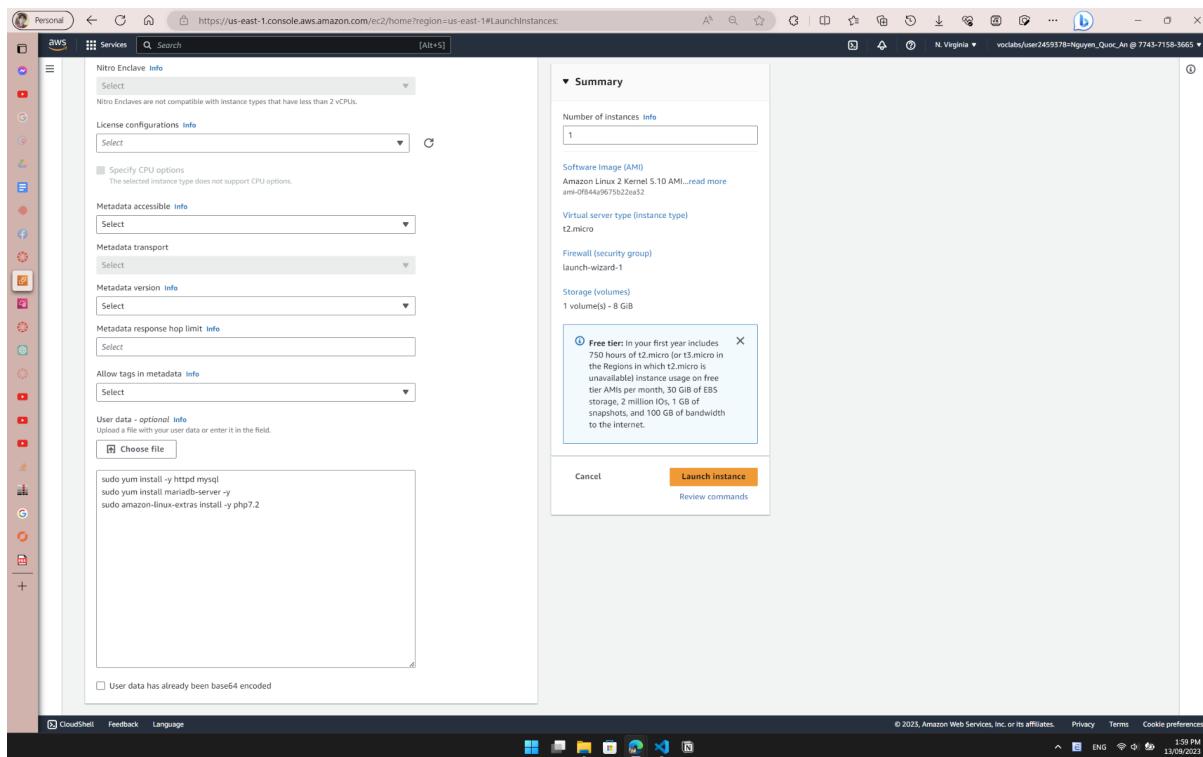
Security Group: MomandPop-Cafe-Web-Server-SG

IAM instance profile: LabInstanceProfile

User Data:

```
#!/bin/bash
# Install Apache Web Server and PHP
sudo yum install -y httpd mysql
sudo yum install mariadb-server -y
sudo amazon-linux-extras install -y php7.2
```





Connect to the created instance through Instance Connect, and run the following command in the terminal:

```
sudo httpd -v  
service httpd status
```

```
mysql --version  
service mysqld status
```

```
php --version
```

If the result indicates no Apache Web Server installed, then re-run the following command to install the web server, database server, and php.

```
#!/bin/bash  
# Install Apache Web Server and PHP  
sudo yum install -y httpd mysql  
sudo yum install mariadb-server -y  
sudo amazon-linux-extras install -y php7.2
```

Then run the following command to start the web server and database:

```
# Turn on web server  
sudo chkconfig httpd on  
sudo service httpd start  
sudo service httpd status
```

```
# Turn on database
sudo systemctl enable mariadb
sudo systemctl start mariadb
sudo systemctl status mariadb
```

The output will look like this:

```
aws Personal Services Search [Alt+S]
71 logstash-19 available [ =stable ]
72 collectd-python3 available [ =stable ]
* Extra topic has reached end of support.
Note on end-of-support: Use 'info' subcommand.
topic httpd-31-42-85.ec2.internal mariadb
Note: Forwarding request to 'systemctl enable httpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-42-85 ~]$ sudo service httpd start
[ec2-user@ip-172-31-42-85 ~]$ sudo service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
     Docs: man:php-fpm(7)
   Active: active (running) since Wed 2023-09-13 07:05:58 UTC; 46ms ago
     Docs: man:systemctl(1)
   Main PID: 5003 (httpd)
     Status: "Processing requests..."
   CGroup: /system.slice/systemd-system-t
           ├ 5003 /usr/sbin/httpd -DFOREGROUND
           ├ 5013 /usr/sbin/httpd -DFOREGROUND
           ├ 5014 /usr/sbin/httpd -DFOREGROUND
           ├ 5015 /usr/sbin/httpd -DFOREGROUND
           ├ 5016 /usr/sbin/httpd -DFOREGROUND
           └ 5017 /usr/sbin/httpd -DFOREGROUND

Sep 13 07:05:58 ip-172-31-42-85.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Sep 13 07:05:58 ip-172-31-42-85.ec2.internal systemd[1]: Started The Apache HTTP Server.
[ec2-user@ip-172-31-42-85 ~]$ sudo systemctl enable mariadb.service to /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-172-31-42-85 ~]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-42-85 ~]$ sudo systemctl status mariadb
● mariadb.service - Mariadb database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
     Active: active (running) since Wed 2023-09-13 07:06:02 UTC; 5s ago
       Docs: man:mysqld(8)
   Process: 5175 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
   Process: 5092 ExecStart=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
   Main PID: 5174 (mysqld_safe)
     Groups: _mysql
   CGroup: /system.slice/mariadb.service
           ├ 5174 /bin/mysqld_safe --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.log --pid-file=/var/run/mariadb/mariadb.pid --socket=/var/lib/mysql/mariadb.sock
           └ 5340 /usr/libexec/mysqld

Sep 13 07:06:00 ip-172-31-42-85.ec2.internal mariadb-prepare-db-dir[5092]: databases and anonymous users created by default. This is
Sep 13 07:06:00 ip-172-31-42-85.ec2.internal mariadb-prepare-db-dir[5092]: strongly recommended for production servers.
Sep 13 07:06:00 ip-172-31-42-85.ec2.internal mariadb-prepare-db-dir[5092]: More information about MySQL can be found at http://mariadb.com/kb or the
Sep 13 07:06:00 ip-172-31-42-85.ec2.internal mariadb-prepare-db-dir[5092]: MySQL manual for more instructions.
Sep 13 07:06:00 ip-172-31-42-85.ec2.internal mariadb-prepare-db-dir[5092]: Please report any problems at http://mariadb.org/jira.
Sep 13 07:06:00 ip-172-31-42-85.ec2.internal mariadb-prepare-db-dir[5092]: The latest information about MariaDB is available at http://mariadb.org/.
Sep 13 07:06:00 ip-172-31-42-85.ec2.internal mariadb-prepare-db-dir[5092]: MySQL is a registered trademark of Oracle Corporation and/or its
Sep 13 07:06:00 ip-172-31-42-85.ec2.internal mysqld_safe[5174]: 230913 07:06:00 mysqld_safe Logging to '/var/log/mariadb/mariadb.log'.
Sep 13 07:06:02 ip-172-31-42-85.ec2.internal systemd[1]: Started MariaDB database server.
[ec2-user@ip-172-31-42-85 ~]$
```

i-050e82d1202fcfffd (MolandPop-Cafe-Web-Server)
PublicIPs: 18.234.59.41 PrivateIPs: 172.31.42.85

Next run the following command to install the web application files and AWS SDK for PHP:

```
# Download Lab files
cd ~/environment
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/setup.tar.gz
tar -zvxf setup.tar.gz
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/db.tar.gz
tar -zvxf db.tar.gz
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/cafe.tar.gz
tar -zvxf cafe.tar.gz
# Download and install the AWS SDK for PHP
wget https://github.com/aws/aws-sdk-php/releases/download/3.62.3/aws.zip
sudo mkdir /var/www/html
sudo unzip aws -d /var/www/html
```

Next, move the cafe application to the host folder:

```
# Move web to host folder  
sudo mv cafe /var/www/html/
```

```
Personal < > Home https://us-east-1.console.aws.amazon.com/ec2-instance-connect/sh/?region=us-east-1&connType=standard&instanceId=ec2-userip-172-31-42-85 N. Virginia vocabs/user2459378:Nguyen_QucAn@7743-7158-3665

aws Services Search [Alt+S]

creating: /var/www/html/Aws/Ec2/Exception/
inflate: /var/www/html/Aws/Ec2/Exception/Ec2Exception.php
inflate: /var/www/html/Aws/Ec2/Exception/Ec2Incompleteabant.php
creating: /var/www/html/Aws/CloudTrail/
inflate: /var/www/html/Aws/CloudTrail/LogFileIterator.php
creating: /var/www/html/Aws/CloudTrail/Exception/
inflate: /var/www/html/Aws/CloudTrail/CloudTrailException.php
inflate: /var/www/html/Aws/CloudTrail/CloudTrailClient.php
inflate: /var/www/html/Aws/CloudTrail/LogRecordIterator.php
inflate: /var/www/html/Aws/CloudTrail/LogRecordHeader.php
inflate: /var/www/html/Aws/IotJobsDataPlane/
inflate: /var/www/html/Aws/IotJobsDataPlane/IotJobsDataPlaneClient.php
creating: /var/www/html/Aws/IotJobsDataPlane/Exception/
inflate: /var/www/html/Aws/IotJobsDataPlane/IotJobsDataPlaneException.php
inflate: /var/www/html/Aws/Sns/MessageValidator.php
creating: /var/www/html/Aws/Sns/Exception/
inflate: /var/www/html/Aws/Sns/SnsException.php
inflate: /var/www/html/Aws/Sns/SnsException/invalidSnsMessageException.php
inflate: /var/www/html/Aws/Sns/SnsClient.php
inflate: /var/www/html/Aws/Sns/Message.php
creating: /var/www/html/Aws/ElasticsearchService/
inflate: /var/www/html/Aws/ElasticsearchService/ElasticsearchServiceClient.php
creating: /var/www/html/Aws/ElasticsearchService/Exception/
inflate: /var/www/html/Aws/ElasticsearchService/ElasticsearchServiceException.php
inflate: /var/www/html/Aws/Neptune/
creating: /var/www/html/Aws/Neptune/Exception/
inflate: /var/www/html/Aws/Neptune/NeptuneException.php
inflate: /var/www/html/Aws/Neptune/NeptuneClient.php
inflate: /var/www/html/Aws/Redshift/
creating: /var/www/html/Aws/Redshift/Exception/
inflate: /var/www/html/Aws/Redshift/RedshiftExceptionRedshiftException.php
inflate: /var/www/html/Aws/Redshift/RedshiftClient.php
inflate: /var/www/html/Aws/MockHandler.php
creating: /var/www/html/Aws/KinesisVideoArchiveMedia/
inflate: /var/www/html/Aws/KinesisVideoArchiveMedia/KinesisVideoArchivedMediaClient.php
inflate: /var/www/html/Aws/KinesisVideoArchiveMedia/Exception/
creating: /var/www/html/Aws/DeviceFarm/
inflate: /var/www/html/Aws/DeviceFarm/Exception/deviceFarmException.php
inflate: /var/www/html/Aws/DeviceFarm/Exception/deviceFarmException.php
inflate: /var/www/html/Aws/DeviceFarm/DeviceFarmClient.php
creating: /var/www/html/Aws/CloudDirectory/
inflate: /var/www/html/Aws/CloudDirectory/CloudDirectoryException.php
inflate: /var/www/html/Aws/CloudDirectory/Exception/CloudDirectoryException.php
inflate: /var/www/html/Aws/CloudDirectory/CloudDirectoryClient.php
creating: /var/www/html/Aws/SageMaker/
inflate: /var/www/html/Aws/SageMaker/SageMakerClient.php
creating: /var/www/html/Aws/SageMaker/Exception/
inflate: /var/www/html/Aws/SageMaker/SageMakerException.php
inflate: /var/www/html/Aws/SageMaker/SageMakerClient.php
[ec2-user@ip-172-31-42-85 ~] $ sudo mv cafe /var/www/html/
[ec2-user@ip-172-31-42-85 ~] $ i-005e82d1202fcffd (MamondPop-Cafe-Web-Server)
PublicIP: 18.234.59.41 PrivateIP: 172.31.42.85

CloudShell Feedback Language
© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
2:09 PM 13/09/2023
```

Next, run the setup file to set up the database credential in parameter stored in AWS System Manager:

```
#Set up parameter store  
cd setup  
sudo ./set-app-parameters.sh
```

The output should display a list of 7 JSON as below:

```
aws Services Search [Alt+S]
infloating: /var/www/html/aws/CloudDirectory/Exception/CloudDirectoryException.php
infloating: /var/www/html/aws/CloudDirectory/CloudDirectoryClient.php
creating: /var/www/html/aws/SageMaker
creating: /var/www/html/aws/SageMaker/SageMakerClient.php
creating: /var/www/html/aws/SageMaker/Exception/SageMakerException.php
infloating: /var/www/html/aws/autoload.php
infloating: /var/www/html/NOTICE.md
[ec2-user@ip-172-31-42-85 ~]$ sudo mv cafe /var/www/html/
[ec2-user@ip-172-31-42-85 ~]$ ls
[ec2-user@ip-172-31-42-85 ~]$ sudo ./set-app-parameters.sh

Setting the default AWS region...
% Total  % Received = Xferd  Average Speed   Time   Time  Current
          Dload Upload Total Spent  Left  Speed
100  10 100  10  0  0 1409  0 :--:-- :--:-- :--:-- 1428
Region = us-east-1
% Total  % Received = Xferd  Average Speed   Time   Time  Current
          Dload Upload Total Spent  Left  Speed
100  40 100  40  0  0 5284  0 :--:-- :--:-- :--:-- 5714
Public DNS = ec2-16-234-59-41.compute-1.amazonaws.com

Setting the application parameter values in the Parameter Store...
{
    "Plan": "standard",
    "Version": 1

    "Plan": "standard",
    "Version": 1
}

Application Parameter Setup script completed.

[ec2-user@ip-172-31-42-85 setup]$ i-005e82d1202fcffd (MomandPop-Cafe-Web-Server)
PrivateIP: 18.234.59.41  PrivateIPs: 172.31.42.85
```

In “Parameter Store” section of AWS System Manager, there will be 7 values created:

The screenshot shows the AWS Systems Manager Parameter Store interface. The URL is https://us-east-1.console.aws.amazon.com/systems-manager/parameters/?region=us-east-1&tab=Table. The page title is "AWS Systems Manager > Parameter Store". The navigation bar includes "My parameters" (selected), "Public parameters", and "Settings". A search bar at the top right has "[Alt+S]". The main content area is titled "My parameters" and contains a table with the following data:

Name	Tier	Type	Last modified
/cafe/currency	Standard	String	Wed, 13 Sep 2023 07:09:44 GMT
/cafe/dbName	Standard	String	Wed, 13 Sep 2023 07:09:45 GMT
/cafe/dbPassword	Standard	String	Wed, 13 Sep 2023 07:09:46 GMT
/cafe/dbUrl	Standard	String	Wed, 13 Sep 2023 07:09:45 GMT
/cafe/dbUser	Standard	String	Wed, 13 Sep 2023 07:09:46 GMT
/cafe/showServerInfo	Standard	String	Wed, 13 Sep 2023 07:09:43 GMT
/cafe/timeZone	Standard	String	Wed, 13 Sep 2023 07:09:43 GMT

At the top right of the table, there are buttons for "View details", "Edit", and "Delete", followed by a "Create parameter" button. Below the table, there are navigation arrows and a refresh icon.

Next run the db setup file to create and set the parameter for the database:

```
# Set up database  
cd ..db/  
sudo ./set-root-password.sh  
sudo ./create-db.sh
```

The output message will indicate successful operation:

A screenshot of an AWS CloudShell terminal window. The terminal shows the execution of several commands to set up a database. The output includes logs for network traffic, application parameter settings, and successful completion of root password and database creation scripts. The terminal also displays the AWS logo and navigation links at the bottom.

```
Setting the default AWS region...  
  % Total    % Received % Xferd  Average Speed   Time   Time  Current  
  100  10  100  10      0     0  1409      0 --:--:-- --:--:-- --:--:-- 1428  
Resuming download from 100...  
  % Total    % Received % Xferd  Average Speed   Time   Time  Current  
  100  40  100  40      0     0  5284      0 --:--:-- --:--:-- --:--:-- 5714  
Public DNS = ec2-18-234-59-41.compute-1.amazonaws.com  
Setting the application parameter values in the Parameter Store...  
{  
  "Tier": "Standard",  
  "Version": 1  
  
  "Tier": "Standard",  
  "Version": 1  
  
Application Parameter Setup script completed.  
(ec2-user@ip-172-31-42-85:~) $ cd ..db/  
(ec2-user@ip-172-31-42-85:~) $ sudo ./set-root-password.sh  
Set Root Password script completed.  
Please check the set-root-password.log file to verify successful execution.  
(ec2-user@ip-172-31-42-85:~) $ sudo ./create-db.sh  
Create Database script completed.  
Please check the create-db.log file to verify successful execution.  
(ec2-user@ip-172-31-42-85:~) $
```

Now that the database has been created, run the following command to check:

```
mysql -u root -p
```

Input the password in “/cafe/dbPassword” parameter store, then run these SQL command:

```
show databases;  
use cafe_db;  
show tables;  
select * from product;  
exit;
```

The data will display as below:

AWS Services Search [Alt+S]

```
[ec2-user:~] ip-172-31-42-85 ~ $ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 5.5.68-MariaDB Server
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cafe_db |
| mysql |
| performance_schema |
| test |
+-----+
5 rows in set (0.00 sec)

MariaDB [(none)]> use cafe_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MariaDB [cafe_db]> show tables;
+-----+
| Tables_in_cafe_db |
+-----+
| order |
| order_item |
| product |
| product_group |
+-----+
4 rows in set (0.00 sec)

MariaDB [cafe_db]> select * from product;
+-----+-----+-----+-----+-----+
| id | product_name | description | price | product_group | image_url |
+-----+-----+-----+-----+-----+
| 1 | Croissant | Fresh, buttery and fluffy... Simply delicious! | 1.50 | 1 | images/Croissants.jpg |
| 2 | Donut | We have more than half-a-dozen flavors! | 1.00 | 1 | images/Donuts.jpg |
| 3 | Chocolate Chip Cookie | Made with Swiss chocolate with a touch of Madagascar vanilla | 2.50 | 1 | images/Chocolate-Chip-Cookies.jpg |
| 4 | Scone | Light and airy, filled with apricot jam | 1.50 | 1 | images/Scones.jpg |
| 5 | Strawberry Blueberry Tart | Bursting with the taste and aroma of fresh fruit | 3.50 | 1 | images/Strawberry-Blueberry-Tarts.jpg |
| 6 | Strawberry Tart | Made with fresh ripe strawberries and a delicious whipped cream | 3.50 | 1 | images/Strawberry-Tarts.jpg |
| 7 | Coffee | Freshly-ground black or blend columbian coffee | 3.00 | 2 | images/coffee.jpg |
| 8 | Latte | Rich and creamy coffee with steamed milk | 3.00 | 2 | images/Latte-with-Chocolate.jpg |
| 9 | Latte | Offered hot or cold and in various delicious flavors | 3.50 | 2 | images/Latte.jpg |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

MariaDB [cafe_db]> exit;
i-005e82d1202fcffd (MamandPop-Cafe-Web-Server)
PublicIPs: 18.234.59.41 PrivateIPs: 172.31.42.85
```

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now access the server through it's public IP address, in the format:
 "http://<public-ip>/cafe":

Café

Home About Us Contact Us Menu Order History

Our café offers an assortment of delicious and delectable pastries and coffees that will put a smile on your face. From cookies to croissants, tarts and cakes, each treat is especially prepared to excite your tastebuds and brighten your day!

Frank bakes a rich variety of cookies. Try them all!

Tea, Coffee, Lattes, and Hot Chocolate. Yes, we have it!

Our tarts are always a customer favorite!

About Us

Click the menu and the data will be displayed:

Croissant \$1.50
Fresh, buttery and fluffy... Simply delicious!
Quantity:

Donut \$1.00
We have more than half-a-dozen flavors!
Quantity:

Chocolate Chip Cookie \$2.50
Made with Swiss chocolate with a touch of Madagascar vanilla
Quantity:

Muffin \$3.00
Banana bread, blueberry, cranberry or apple
Quantity:

Strawberry Blueberry Tart \$3.50
Bursting with the taste and aroma of fresh fruit
Quantity:

Strawberry Tart \$3.50
Made with fresh ripe strawberries and a delicious whipped cream
Quantity:

Make an order and it will be saved in the order history:

Item	Price	Quantity	Amount
Croissant	\$1.50	1	\$1.50

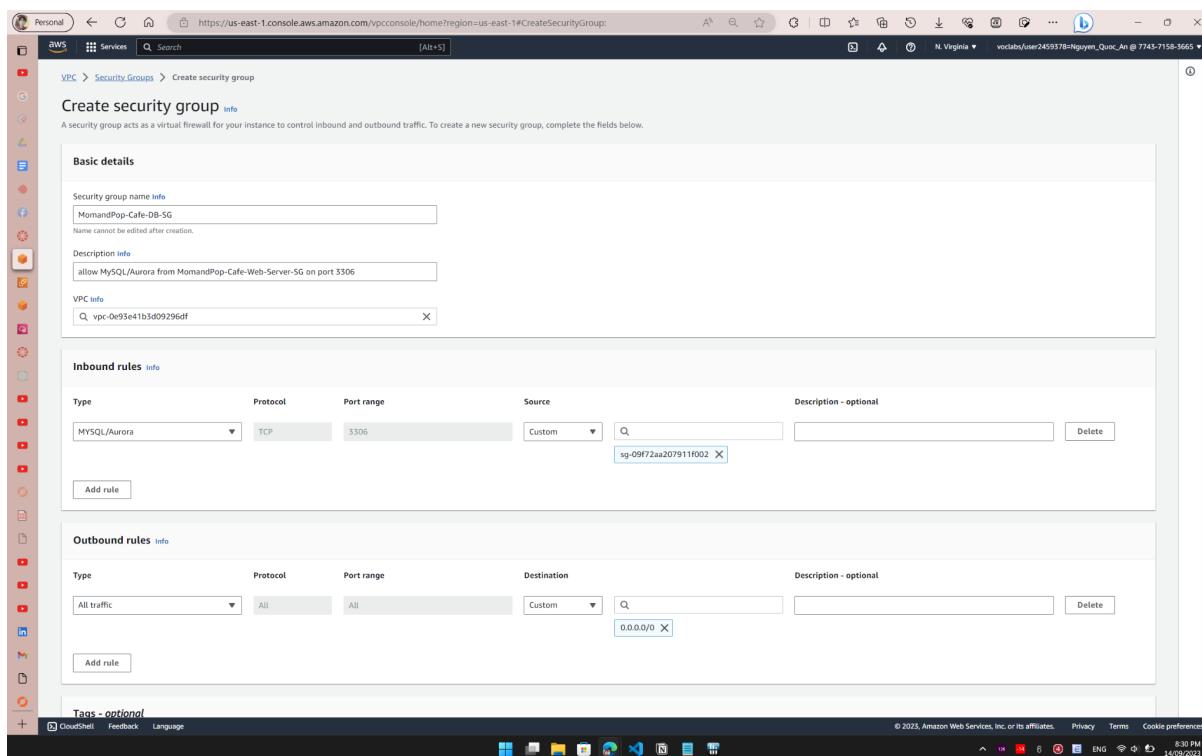
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Until this step, since the PHP web file established a connection to the database through the DNS endpoint of the current EC2 instance, which means if the server is restarted, the public IP and DNS changed, then you will need to update the DNS endpoint in the parameter store “/cafe/dbUrl” of AWS System Parameter Store.

3. Split the database to AWS RDS and Store database connection information in the AWS Systems Manager Parameter Store.

Create a security group for the database instance with the following configuration:

- Name: MomandPop-Cafe-DB-SG
- VPC: MomandPop-Cafe-vpc
- Inbound Rule: allow MySQL/Aurora from MomandPop-Cafe-Web-Server-SG on port 3306
- Outbound Rule: all traffic to 0.0.0.0/0



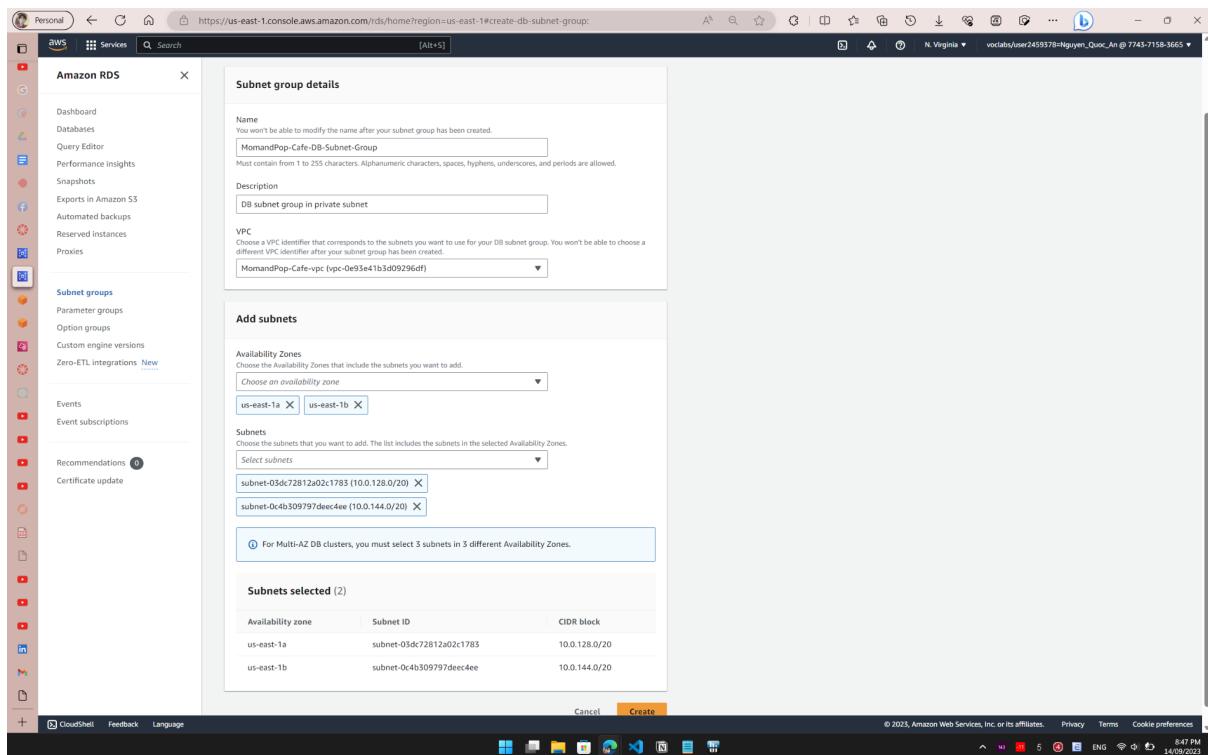
Create a DB subnet group with the following configuration:

Name: MomandPop-Cafe-DB-Subnet-Group

VPC: MomandPop-Cafe-vpc

Availability Zones: us-east-1a and us-east-1b

Subnet: Select the 2 private subnets in the 2 AZs



Create RDS database instance with the following configuration:

- Engine type: MariaDB
- Templates: Dev/Test
- DB instance identifier: MomandPop-Cafe-DB
- Username: admin
- Password: Caf3DbPassw0rd!
- DB Instance Class: db.t3.micro
- Storage type: General Purpose (SSD)
- Allocated storage: 20 GiB
- Do not create a standby instance selected
- VPC: MomandPop-Cafe-vpc
- Subnet Group: MomandPop-Cafe-DB-Subnet-Group
- Security Group: MomandPop-Cafe-DB-SG
- Availability Zone: us-east-1a.
- Database port: TCP port of 3306.
- Disable Enhanced monitoring

The screenshot shows the AWS RDS console for the 'momandpop-cafe-db' database. The left sidebar includes options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update.

Summary tab details:

- DB identifier: momandpop-cafe-db
- CPU: 7.83%
- Status: Backing-up
- Role: Instance
- Engine: MariaDB
- Class: db.t3.micro
- Region & AZ: us-east-1a

Connectivity & security tab details:

Endpoint & port	Networking	Security
Endpoint: momandpop-cafe-db.cbjux8fire.us-east-1.rds.amazonaws.com Port: 3306	Availability Zone: us-east-1a VPC: MomandPop-Cafe-vpc (vpc-0e93e41b3d09296df) Subnet group: momandpop-cafe-db-subnet-group Subnets: subnet-03dc72812a02c1783, subnet-04b309797deec4ee Network type: IPv4	VPC security groups: MomandPop-Cafe-DB-SG (sg-0f402490c20bd955) (Active) Publicly accessible: No Certificate authority: rds-ca-2019 Certificate authority date: August 23, 2024, 00:08 (UTC+07:00) DB instance certificate expiration date: August 23, 2024, 00:08 (UTC+07:00)

Security group rules table:

Security group	Type	Rule
MomandPop-Cafe-DB-SG (sg-0f402490c20bd955)	EC2 Security Group - Inbound	sg-09f72aa207911f002
MomandPop-Cafe-DB-SG (sg-0f402490c20bd955)	CIDR/IP - Outbound	0.0.0.0/0

Connect to the EC2 instance, access mariadb and view the database through:

```
mysql -u root -p
```

```
show databases;
use cafe_db;
show tables;
select * from `order`;
select * from `order_item`;
exit;
```

```

MariaDB [(none)]> use cafe_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

MariaDB [cafe_db]> show tables;
+-----+
| Tables_in_cafe_db |
+-----+
| order           |
| order_item      |
| product         |
| product_group   |
+-----+
4 rows in set (0.00 sec)

MariaDB [cafe_db]> select * from `order`;
+-----+-----+-----+
| order_number | order_date_time | amount |
+-----+-----+-----+
| 1           | 2023-09-14 09:27:54 | 14.00 |
| 2           | 2023-09-14 09:28:02 | 7.00  |
+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [cafe_db]> select * from `order_item`;
+-----+-----+-----+-----+
| order_number | order_item_number | product_id | quantity | amount |
+-----+-----+-----+-----+
| 1           | 1                 | 5          | 4        | 14.00 |
| 2           | 2                 | 9          | 2        | 7.00  |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [cafe_db]> exit;
Bye

```

i-0f6e839950aad9939 (MongodbPop-Cafe-Web-Server)
PublicIPs: 3.236.174.245 PrivateIPs: 10.0.8.64

Dump the entire database to a file using the mysqldump utility:

```
mysqldump --databases cafe_db -u root -p > CafeDbDump.sql
```

Enter the password store the System Manager Parameter Store. After the process is done, you can run “ls” to view the file and “cat CafeDbDump.sql” to view the content of the file.

```

MariaDB [cafe_db]> show tables;
+-----+
| Tables_in_cafe_db |
+-----+
| order           |
| order_item      |
| product         |
| product_group   |
+-----+
4 rows in set (0.00 sec)

MariaDB [cafe_db]> select * from `order`;
+-----+-----+-----+
| order_number | order_date_time | amount |
+-----+-----+-----+
| 1           | 2023-09-14 09:27:54 | 14.00 |
| 2           | 2023-09-14 09:28:02 | 7.00  |
+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [cafe_db]> select * from `order_item`;
+-----+-----+-----+-----+
| order_number | order_item_number | product_id | quantity | amount |
+-----+-----+-----+-----+
| 1           | 1                 | 5          | 4        | 14.00 |
| 2           | 2                 | 9          | 2        | 7.00  |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [cafe_db]> exit;
Bye
[ec2-user@ip-10-0-8-64 ~]$ ^C
[ec2-user@ip-10-0-8-64 ~]$ mysqldump --databases cafe_db -u root -p > CafeDbDump.sql
[ec2-user@ip-10-0-8-64 ~]$ ls
CafeDbDump.sql  cafe.tar.gz  db.tar.gz  setup  setup.tar.gz
[ec2-user@ip-10-0-8-64 ~]$ [ ]

```

i-0f6e839950aad9939 (MongodbPop-Cafe-Web-Server)
PublicIPs: 3.236.174.245 PrivateIPs: 10.0.8.64

Try establishing a connection to the RDS instance created previously:

```
nmap -Pn <rds-endpoint>
```

The output will indicate the port 3306 is open as below:

```
[ec2-user@cafeserver ~]$ nmap -Pn cafedatabase.c4iis9wyj45t.us-east-1.rds.amazonaws.com

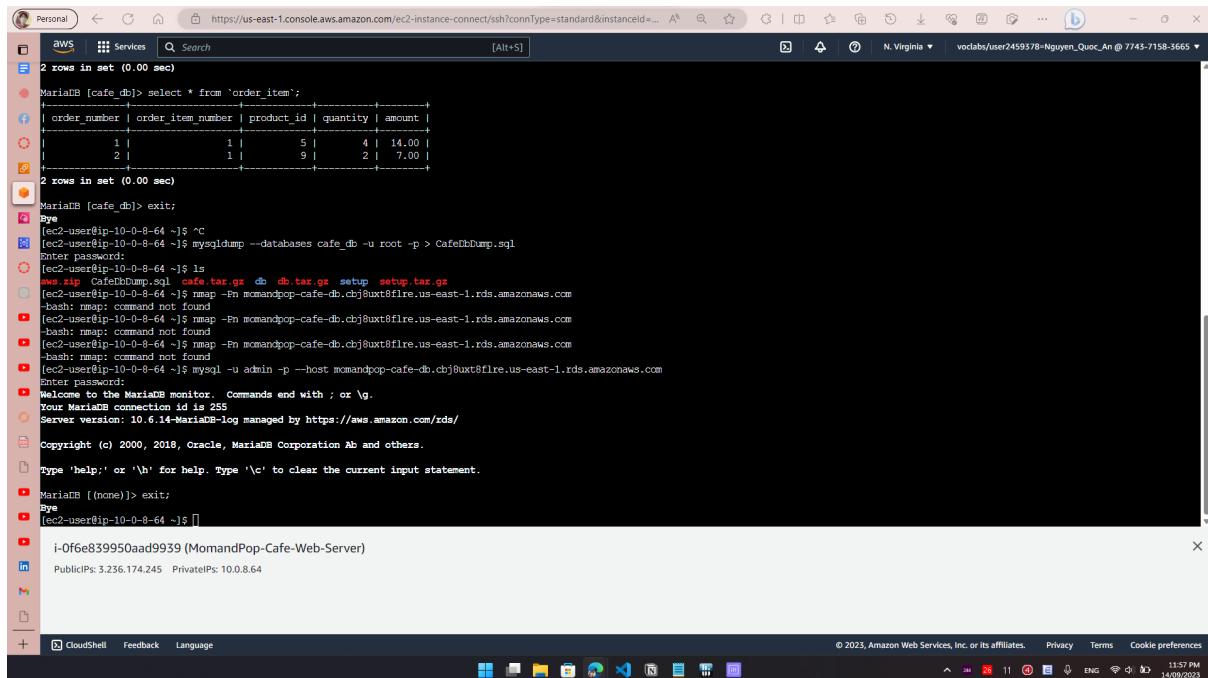
Starting Nmap 6.40 ( http://nmap.org ) at 2023-09-13 10:20 UTC
Nmap scan report for cafedatabase.c4iis9wyj45t.us-east-1.rds.amazonaws.com (10.0.3.116)
Host is up (0.0053s latency).
rDNS record for 10.0.3.116: ip-10-0-3-116.ec2.internal
Not shown: 999 filtered ports
PORT      STATE SERVICE
3306/tcp  open  mysql

Nmap done: 1 IP address (1 host up) scanned in 4.24 seconds
[ec2-user@cafeserver ~]$
```

Then connect to the RDS instance, with the password when creating the instance:

```
mysql -u admin -p --host <rds-endpoint>
```

If success, type "exit;"



```
MariaDB [cafe_db]> select * from `order_item`;
+-----+-----+-----+-----+
| order_number | order_item_number | product_id | quantity | amount |
+-----+-----+-----+-----+
| 1 | 1 | 1 | 5 | 4 | 14.00 |
| 2 | 2 | 1 | 9 | 2 | 7.00 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [cafe_db]> exit;
Bye
[ec2-user@ip-10-0-8-64 ~]$ ^C
[ec2-user@ip-10-0-8-64 ~]$ mysqldump --databases cafe_db -u root -p > CafeDbDump.sql
Enter password:
[ec2-user@ip-10-0-8-64 ~]$ ls
aws.zip CafeDbDump.sql cafe.tar.gz db db.tar.gz setup setup.tar.gz
[ec2-user@ip-10-0-8-64 ~]$ rmap -Pn momandpop-cafe-db.cbj@uxt8fire.us-east-1.rds.amazonaws.com
[bash]: rmap: command not found
[ec2-user@ip-10-0-8-64 ~]$ rmap -Pn momandpop-cafe-db.cbj@uxt8fire.us-east-1.rds.amazonaws.com
[bash]: rmap: command not found
[ec2-user@ip-10-0-8-64 ~]$ rmap -Pn momandpop-cafe-db.cbj@uxt8fire.us-east-1.rds.amazonaws.com
[bash]: rmap: command not found
[ec2-user@ip-10-0-8-64 ~]$ mysql -u admin -p --host momandpop-cafe-db.cbj@uxt8fire.us-east-1.rds.amazonaws.com
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 255
Server version: 10.6.14-MariaDB-log managed by https://aws.amazon.com/rds/
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> exit;
Bye
[ec2-user@ip-10-0-8-64 ~]$
```

Then run this command to copy the data to the RDS instance:

```
mysql -u admin -p --host <rds-endpoint> < CafeDbDump.sql
```

Then connect back to the RDS instance and check the data:

```
show databases;
use cafe_db;
show tables;
select * from `order`;
exit;
```

```

MariaDB [(none)]> exit;
Bye
Enter password:
[ec2-user@ip-10-0-8-64 ~]$ mysql -u admin -p --host mcmdpop-cafe-db.cbj8uxflre.us-east-1.rds.amazonaws.com < CafeDbDump.sql
Your MariaDB connection id is 269
Server version: 10.6.14-MariaDB-log managed by https://aws.amazon.com/rds/
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| cafe_db |
| information_schema |
| innodb |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.00 sec)

MariaDB [(none)]> use cafe_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MariaDB [cafe_db]> show tables;
+-----+
| Tables in cafe_db |
+-----+
| order |
+-----+
1 row in set (0.00 sec)

i-0f6e839950aad9939 (MramidPop-Cafe-Web-Server)
PublicIPs: 3.236.174.245 PrivateIPs: 10.0.8.64

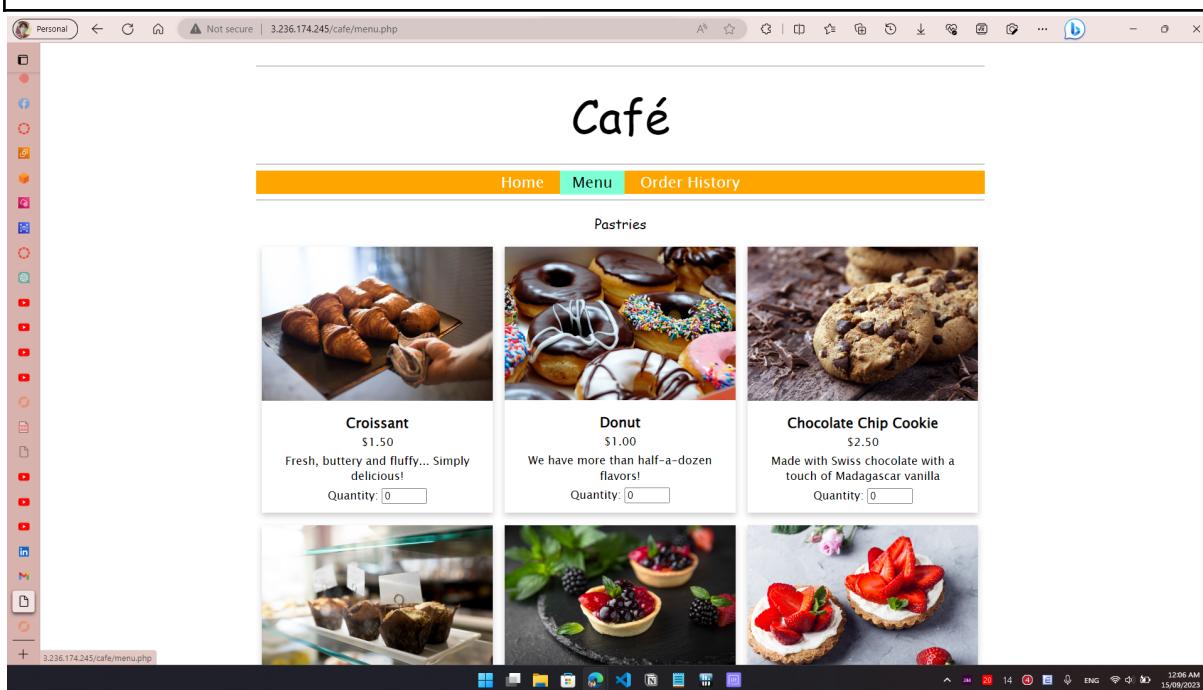
```

Next, go to the parameter store of the AWS System Manager, change the following parameter value:

- /cafe/dbUser : “admin”
- /cafe/dbPassword: RDS instance password (Caf3DbPassw0rd!)
- /cafe/dbUrl: RDS instance endpoint

Check if the web server is still working, then stop the mariadb service:

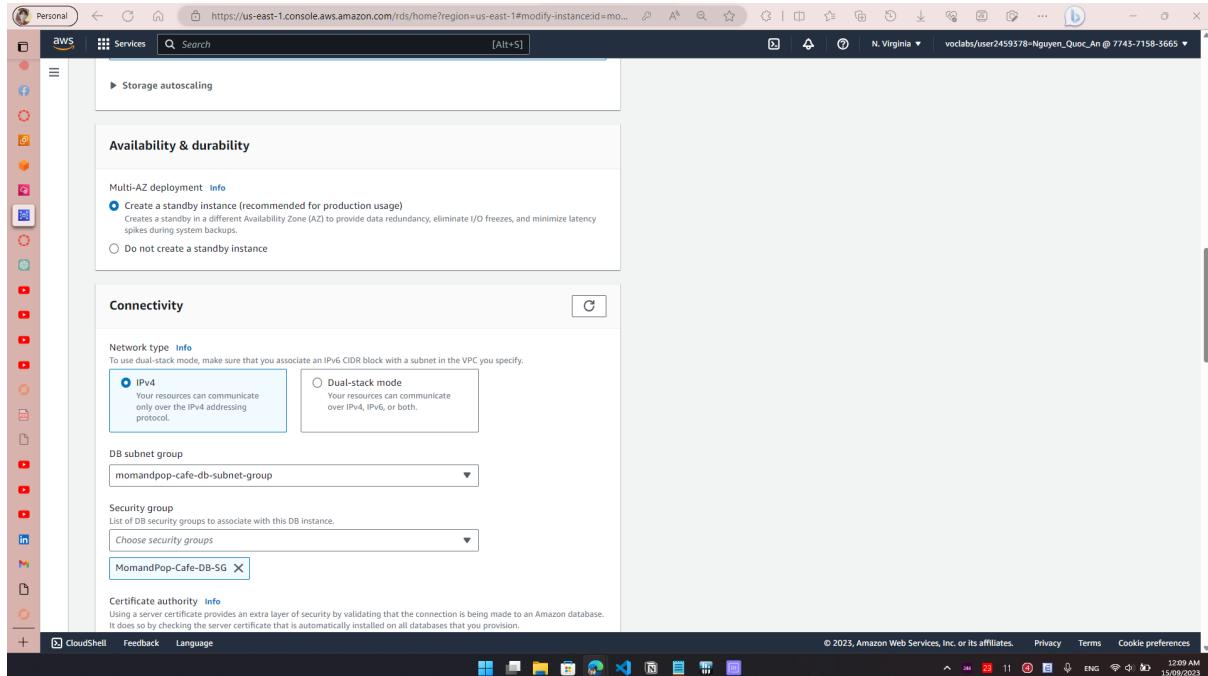
```
sudo service mariadb stop
```



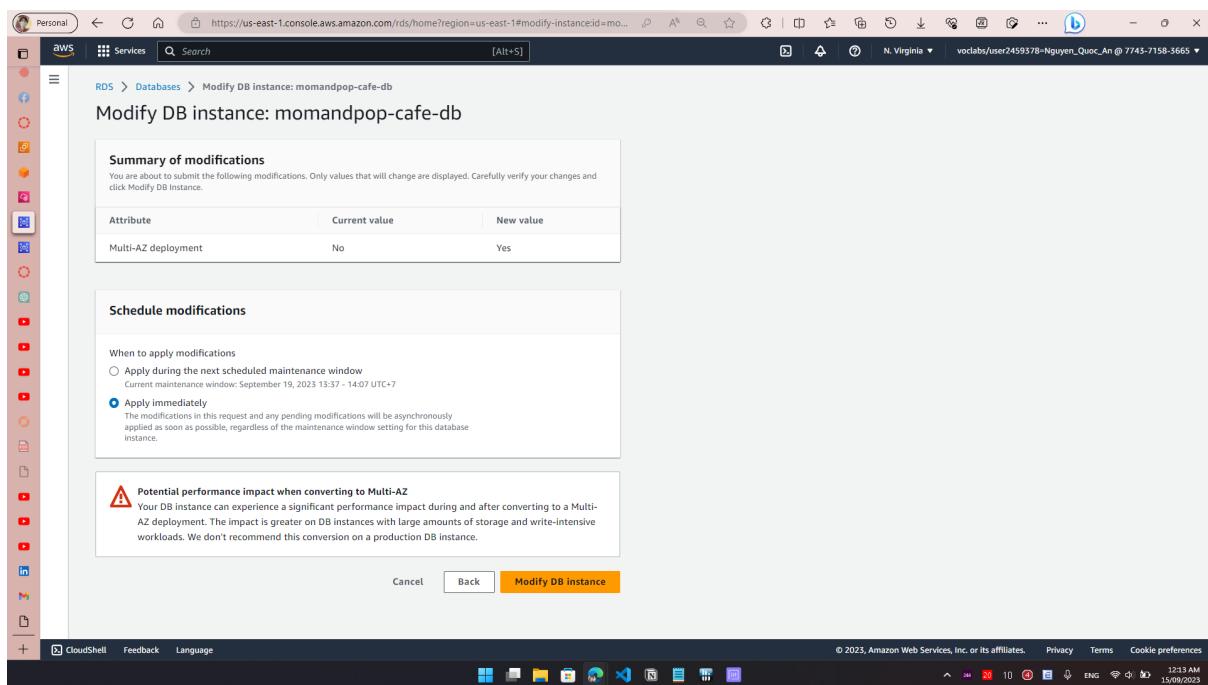
4. Make the database highly available

Go to RDS dashboard, in the left navigation pane, choose Databases, choose momandpop-cafe-db and choose Modify.

Scroll down to the Availability & Durability section. For Multi-AZ deployment, select Create a standby instance.



Under Schedule modifications, select Apply immediately, then Modify DB instance.



5. Implement Application Load Balancer, Auto Scaling Group and scaling policy

Create new Security Group with the following configuration:

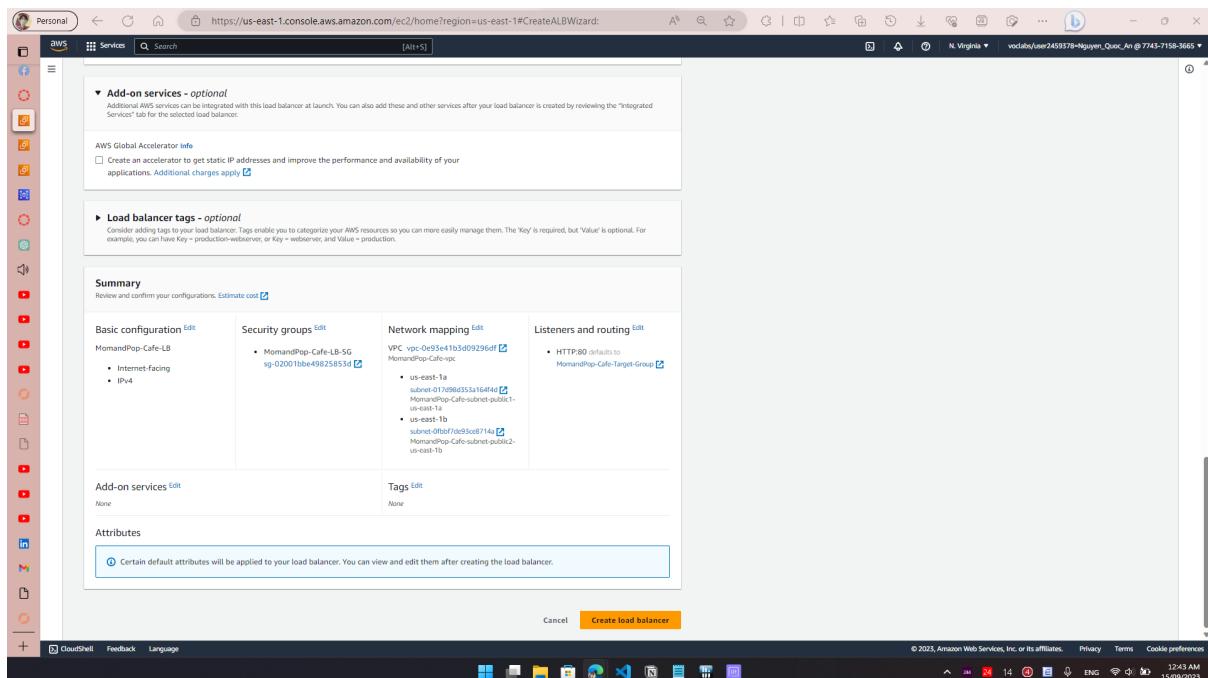
- Name: MomandPop-Cafe-LB-SG

- Inbound rule: HTTP and HTTPS from anywhere IPv4

Create an Application Load balancer with the following configuration:

- Name: MomandPop-Cafe-LB
- VPC: MomandPop-Cafe-vpc
- Network mapping: first AZ -> public 1, second AZ -> public 2
- Security Group: MomandPop-Cafe-LB-SG
- Create new Target Group with the following configuration:
 - Type: Instances
 - Name: MomandPop-Cafe-Target-Group
 - VPC: MomandPop-Cafe-vpc
 - Create target group (no register targets)
- Listeners and routing -> add this target group to default action

-> Create load balancer.



AMI

Create AMI from MomandPop-Cafe-Web-Server instance:

- Select MomandPop-Cafe-Web-Server instance
- In the Actions menu, choose Image and templates > Create image -> Copy AMI ID

Create Launch Templates

Create a launch template with the following configuration:

- Launch template name: MomandPop-Cafe-Web-Server-LT
- Under Auto Scaling guidance, select Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

- In the Application and OS Images (Amazon Machine Image) area, choose My AMIs.
- Amazon Machine Image (AMI): choose *MomandPop-Cafe-Web-Server-Image*
- Instance type: choose *t2.micro*
- Key pair name: choose *vockey*
- Firewall (security groups): choose *Select existing security group*
- Security groups: choose *MomandPop-Cafe-Web-Server-SG*
- *Add a network interface, and enable Auto-assign public IP*
- *Add tag: Key = Name, Value = MomandPop-Cafe-AutoScaling-Instance*
- ***Subnet: MomandPop-Cafe-subnet-public1-us-east-1a***
- Scroll down to the Advanced details area and expand it.
- IAM instance profile: choose *LabInstanceProfile*
- Scroll down to the Detailed CloudWatch monitoring setting. Select *Enable Note:* This will allow Auto Scaling to react quickly to changing utilization.

(*Due to some error in configuration, in this assignment I choose the version 5 of this launch template)

Create Auto Scaling Group

Select the newly created Launch Template, choose Create Auto Scaling Group

- **Auto Scaling group name:** MomandPop-Cafe-ASG
- **Launch template:** MomandPop-Cafe-Web-Server-LT
- **VPC:** MomandPop-Cafe-vpc
- **Availability Zones and subnets:**
 - us-east-1a and MomandPop-Cafe-subnet-public1-us-east-1a
 - us-east-1b and MomandPop-Cafe-subnet-public2-us-east-1b
- In the **Load balancing** panel:
 - Choose **Attach to an existing load balancer**
 - **Existing load balancer target groups:**
MomandPop-Cafe-Target-Group
- In the **Health checks** panel:
 - **Health check grace period:** 90 seconds
- In the **Additional settings** panel:
 - Select **Enable group metrics collection within CloudWatch**
 - **Desired capacity:** 2
 - **Minimum capacity:** 2
 - **Maximum capacity:** 6
- Under **Scaling policies** choose target tracking
 - Metric type: Avarage CPU utilization
 - Target value: 50
- Tag : key: Name, value: MomandPop-Cafe-ASG-Instance

Personal

EC2 > Auto Scaling groups > Create Auto Scaling group

Review [Info](#)

Step 1: Choose launch template

Step 2: Choose instance launch options

Step 3 - optional: Configure advanced options

Step 4 - optional: Configure group size and scaling policies

Step 5 - optional: Add notifications

Step 6 - optional: Add tags

Step 7: Review

Group details

Auto Scaling group name: MomandPop-Cafe-ASG

Launch template

Launch template	Version	Description
MomandPop-Cafe-Web-Server-LT	Default	lt-0deeb0xbcd9e57240d

Step 2: Choose instance launch options

Network

VPC: vpc-0e93e41b5d09296df

Availability Zone	Subnet	CIDR Range
us-east-1a	subnet-017d98d3553a164f4d	10.0.0.0/20
us-east-1b	subnet-0fbfb7de93ce8714e	10.0.16.0/20

Instance type requirements

This Auto Scaling group will adhere to the launch template.

Step 3: Configure advanced options

This Auto Scaling group will adhere to the launch template.

Step 3: Configure advanced options

Load balancing

Load balancer 1

Name	Type	Target group
MomandPop-Cafe-LB	Application/HTTP	MomandPop-Cafe-Target-Group

VPC Lattice integration options

VPC Lattice target groups: -

Health checks

Health check type	Health check grace period
EC2	90 seconds

Additional settings

Monitoring	Default instance warmup
Enabled	Disabled

Step 4: Configure group size and scaling policies

Group size

Desired capacity	Minimum capacity	Maximum capacity
2	2	6

CloudShell Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences N. Virginia vocabs/user2459378-HNguyen_Quoc_An @ 7745-7158-3665 126 AM 15/09/2023

Personal

EC2 > Auto Scaling groups > Create Auto Scaling group

Review [Info](#)

Step 1: Choose launch template

Step 2: Choose instance launch options

Step 3 - optional: Configure advanced options

Step 4 - optional: Configure group size and scaling policies

Step 5 - optional: Add notifications

Step 6 - optional: Add tags

Step 7: Review

Group details

Auto Scaling group name: MomandPop-Cafe-ASG

Launch template

Launch template	Version	Description
MomandPop-Cafe-Web-Server-LT	Default	lt-0deeb0xbcd9e57240d

Step 2: Choose instance launch options

Network

VPC: vpc-0e93e41b5d09296df

Availability Zone	Subnet	CIDR Range
us-east-1a	subnet-017d98d3553a164f4d	10.0.0.0/20
us-east-1b	subnet-0fbfb7de93ce8714e	10.0.16.0/20

Instance type requirements

This Auto Scaling group will adhere to the launch template.

Step 3: Configure advanced options

This Auto Scaling group will adhere to the launch template.

Load balancing

Load balancer 1

Name	Type	Target group
MomandPop-Cafe-LB	Application/HTTP	MomandPop-Cafe-Target-Group

VPC Lattice integration options

VPC Lattice target groups: -

Health checks

Health check type	Health check grace period
EC2	90 seconds

Additional settings

Monitoring	Default instance warmup
Enabled	Disabled

Step 4: Configure group size and scaling policies

Group size

Desired capacity	Minimum capacity	Maximum capacity
2	2	6

CloudShell Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences N. Virginia vocabs/user2459378-HNguyen_Quoc_An @ 7745-7158-3665 126 AM 15/09/2023

Screenshot of the AWS CloudFormation console showing the creation of an Auto Scaling group. The configuration includes:

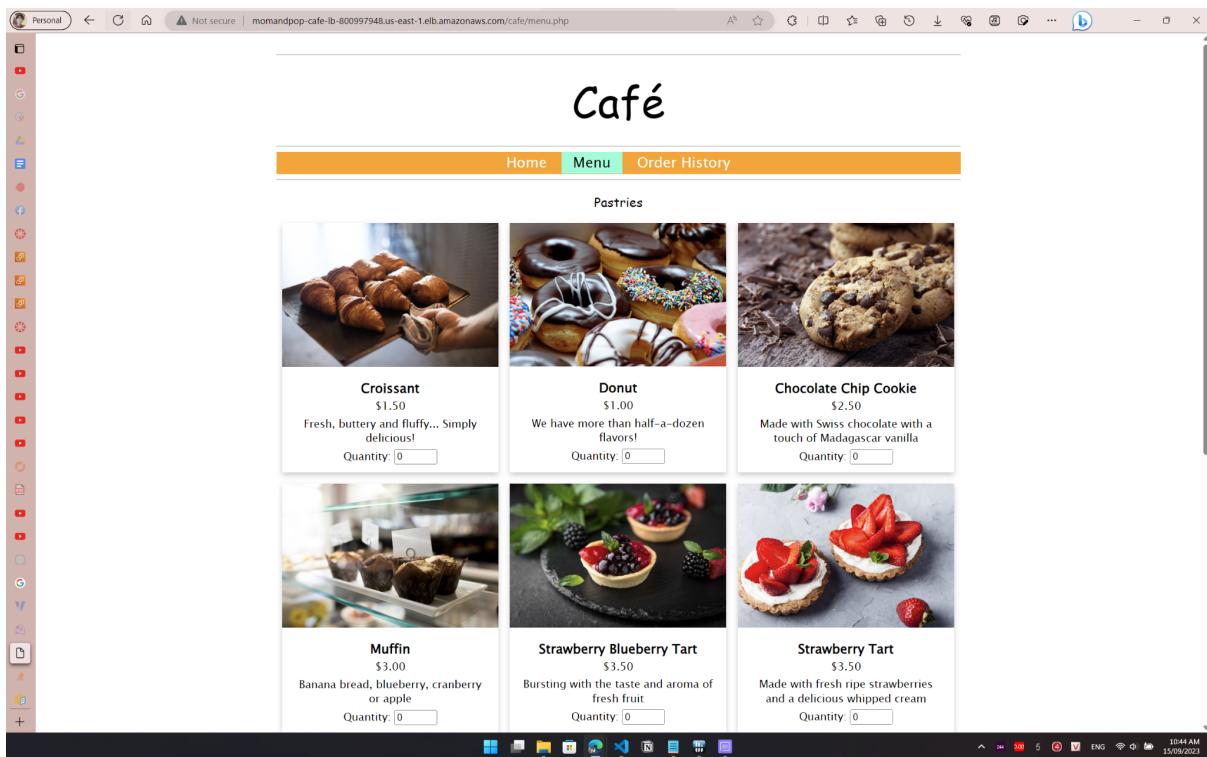
- Desired capacity:** 2
- Scaling policy:** Target tracking scaling (Target Tracking Policy, execute when utilization at 50%)
- Instance scale-in protection:** Enabled
- Step 5: Add notifications:** No notifications
- Step 6: Add tags:** Tag (Name: MomandPop-Cafe-ASG-Instance, Value: Yes)

The "Create Auto Scaling group" button is visible at the bottom right.

After creation, there will be 2 initial instances being launched. Now the website will be accessible through the DNS endpoint of the Application Load Balancer

Screenshot of a web browser displaying the website for "Café". The page features:

- A header with the word "Café" and a navigation menu: Home, About Us, Contact Us, Menu, Order History.
- Two images: one showing pastries like croissants and tarts, and another showing a display case full of various cakes.
- A descriptive text block: "Our café offers an assortment of delicious and delectable pastries and coffees that will put a smile on your face. From cookies to croissants, tarts and cakes, each treat is especially prepared to excite your tastebuds and brighten your day!"
- Three callout boxes:
 - "Frank bakes a rich variety of cookies. Try them all!" with an image of six different cookies.
 - "Tea, Coffee, Lattes, and Hot Chocolate. Yes, we have it!" with an image of a cup of coffee.
 - "Our tarts are always a customer favorite!" with images of tarts topped with strawberries and blackberries.
- An "About Us" section at the bottom.



6. Test Auto Scaling

Initially, as we declare the desired instance in the auto-scaling group is 2, so there will be 2 instances on standby.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
MomandPop-Cafe-ASG-Instance	i-0ba9355c0eb4cd966	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-44-192-107-45.co
MomandPop-Cafe-ASG-Instance	i-0c072af255c0eb90b	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-54-161-209-7.cor
MomandPop-Cafe-Web-Server	i-0f6e839950aad9939	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-44-192-131-23.co

Connect to one of the instances launched by the Auto Scaling Group by instance connect, and input the following code to install stress test program to test load of CPU:

```
sudo yum install
```

```
https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
sudo yum install stress -y
stress --cpu 1 --timeout 600
```

The screenshot shows a Microsoft Edge browser window with the URL <https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c072af255c0e...>. The terminal session displays the following commands and output:

```
Microsoft Edge File Edit View History Favorites Tools Profiles Tab Window Help
Instances | EC2 | us-east-1
us-east-1.console.aws.amazon.com
N. Virginia voclabs/user2459378=Nguyen_Quoc_An @ 7743-7158-3665
Tue Sep 19 8:08 AM

AWS Services Q Search [Option+S]

--> Package stress.x86_64 0:1.0.4-16.el7 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

Package           Arch      Version       Repository   Size
stress            x86_64    1.0.4-16.el7  epel          39 k

Transaction Summary
Install 1 Package

Total download size: 39 k
Installed size: 94 k
Downloading packages:
Warning: /var/cache/yum/x86_64/2/epel/packages/stress-1.0.4-16.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY
Public key for stress-1.0.4-16.el7.x86_64.rpm is not installed
stress-1.0.4-16.el7.x86_64.rpm
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importing GPG key 0x352c64e5:
  Userid : "Fedora EPEL (7) <epel@fedoraproject.org>"
  Fingerprint: 91e0 445e 96f1 f73e 88f8 6a2f aea2 352c 64e5
  Package: epel-release-7-14.noarch (installed)
  From:   /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : stress-1.0.4-16.el7.x86_64
  Verifying   : stress-1.0.4-16.el7.x86_64
  Installed   : stress.x86_64 0:1.0.4-16.el7
Complete!
[root@ip-10-0-31-152 ~]# stress --cpu 1 --timeout 600
stress: info: [26120] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```

The terminal session ends with the command `stress --cpu 1 --timeout 600`. Below the terminal, the AWS CloudShell interface shows the instance details:

- Instance ID: i-0c072af255c0e...
- Public IP: 54.161.209.7
- Private IP: 10.0.31.152

After some moment, alarm will be raise and it can be observed in AWS Cloudwatch, new instances will be launched by the auto scaling group.

The screenshot shows the AWS CloudWatch Metrics interface with the URL <https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:~alarmStateFilter~ALARM>. The left sidebar shows the CloudWatch navigation menu. The main pane displays the Alarms section, listing one alarm:

Name	State	Last state update	Conditions	Actions
TargetTracking-MomandPop-Cafe-ASG-AlarmHigh-10e8fd8a-e563-457f-9090-37ddfb0f406b	In alarm	2023-09-19 04:44:01	CPUUtilization > 25 for 3 datapoints within 3 minutes	Actions enabled

Instances (6) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
MomandPop-Cafe-ASG-Instance	i-0ba9355c0eb4cd966	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-44-192-107-45.co
MomandPop-Cafe-ASG-Instance	i-053407c3816e34bf9	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-54-163-141-78.co
MomandPop-Cafe-ASG-Instance	i-0c072af255c0eb90b	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-161-209-7.cor
MomandPop-Cafe-Web-Server	i-0f6e839950aad9939	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-44-192-131-23.co
MomandPop-Cafe-ASG-Instance	i-0057dc2e9528cef73	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-3-230-2-199.com
MomandPop-Cafe-ASG-Instance	i-0ca3323f247bbc206	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-44-201-54-189.co

Select an instance

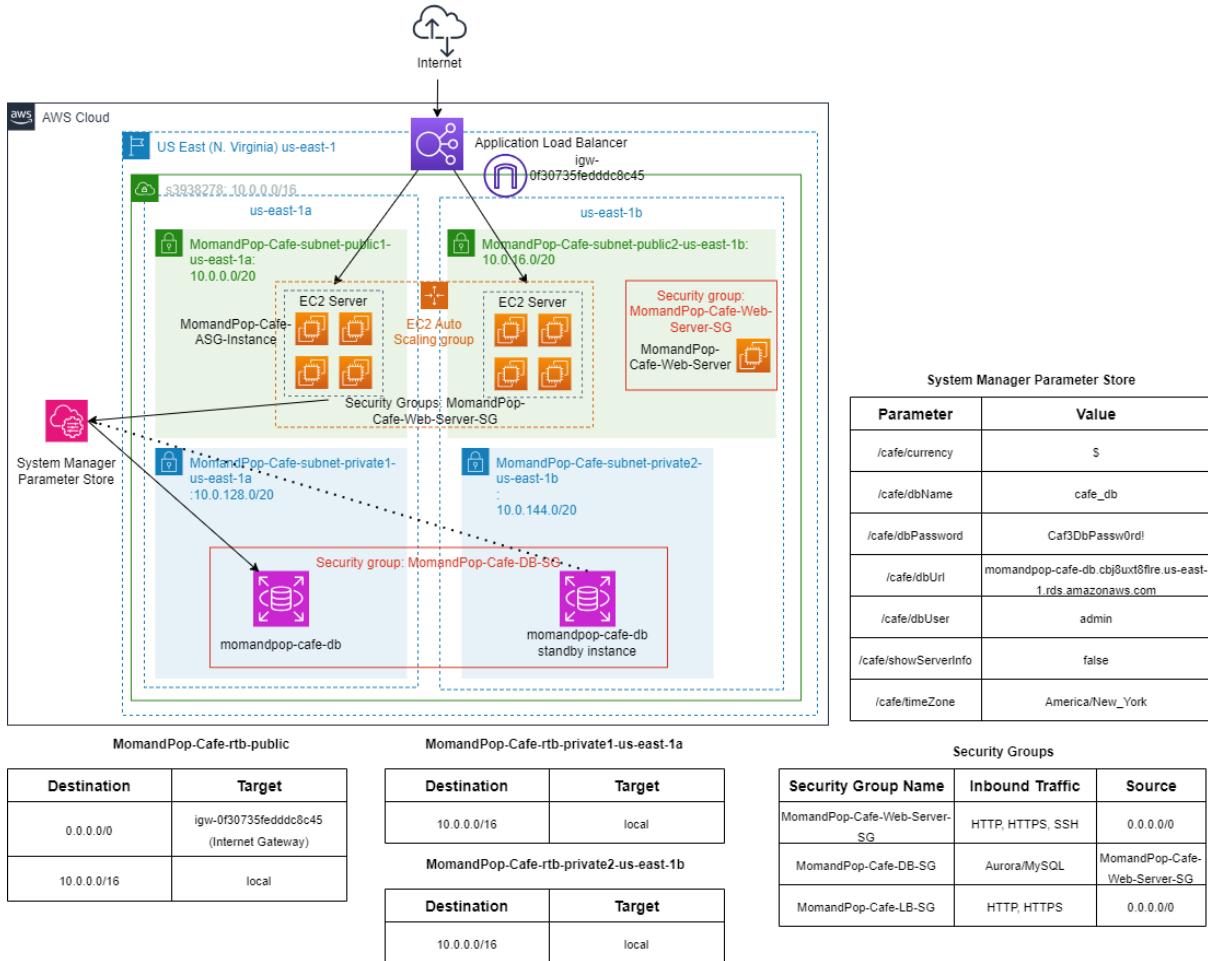
When the stress test is done, and the CPU utilization goes down, the newly launched instances will be terminated back to 2 instances.

Instances (5) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
MomandPop-Cafe-ASG-Instance	i-0ba9355c0eb4cd966	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-44-192-107-45.co
MomandPop-Cafe-ASG-Instance	i-02963a3f0fdcd5a232	Terminated	t2.micro	-	No alarms	us-east-1a	-
MomandPop-Cafe-ASG-Instance	i-0c072af255c0eb90b	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-161-209-7.cor
MomandPop-Cafe-ASG-Instance	i-0e570024c6464711b	Terminated	t2.micro	-	No alarms	us-east-1b	-
MomandPop-Cafe-Web-Server	i-0f6e839950aad9939	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-44-192-131-23.co

Select an instance

7. System Architecture Diagram



Task C - Route 53 Failover Routing

1. Register Domain Name For Website

Register a domain name from a provider, in this assignment, I registered momandpopcafe.shop issued by tenten.vn.

In AWS Route53, create a hosted zone with the name of the domain name

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name [Info](#)
This is the name of the domain that you want to route traffic for.

Valid characters: a-z, 0-9, ! * # \$ % & ' () * + , - / ; < = > ? @ [{ } ^ _ ' { }] ~

Description - optional [Info](#)
This value lets you distinguish hosted zones that have the same name.

Type [Info](#)
The type indicates whether you want to route traffic on the Internet or in an Amazon VPC.
 Public hosted zone
A public hosted zone determines how traffic is routed on the Internet.
 Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

Tags [Info](#)
Apply tags to hosted zones to help organize and identify them.

No tags associated with the resource.
[Add tag](#)
You can add up to 50 more tags.

[Cancel](#) [Create hosted zone](#)

Choose the newly created hosted zone and add an A record, allows alias and route it the application load balancer created in task B.

*The record name have been change to blank later on

Create record [Info](#)

Quick create record [Switch to wizard](#)

Record 1 [Delete](#)

Record name [Info](#) .momandpopcafe.shop
Keep blank to create a record for the root domain.

Record type [Info](#) A – Routes traffic to an IPv4 address and some AWS resources

Alias

Route traffic to [Info](#)
Alias to Application and Classic Load Balancer
US East (N. Virginia)
 [X](#)
Alias hosted zone ID: Z355XDOTRQ7X7K

Routing policy [Info](#) Simple routing Evaluate target health

[Add another record](#)

[Cancel](#) [Create records](#)

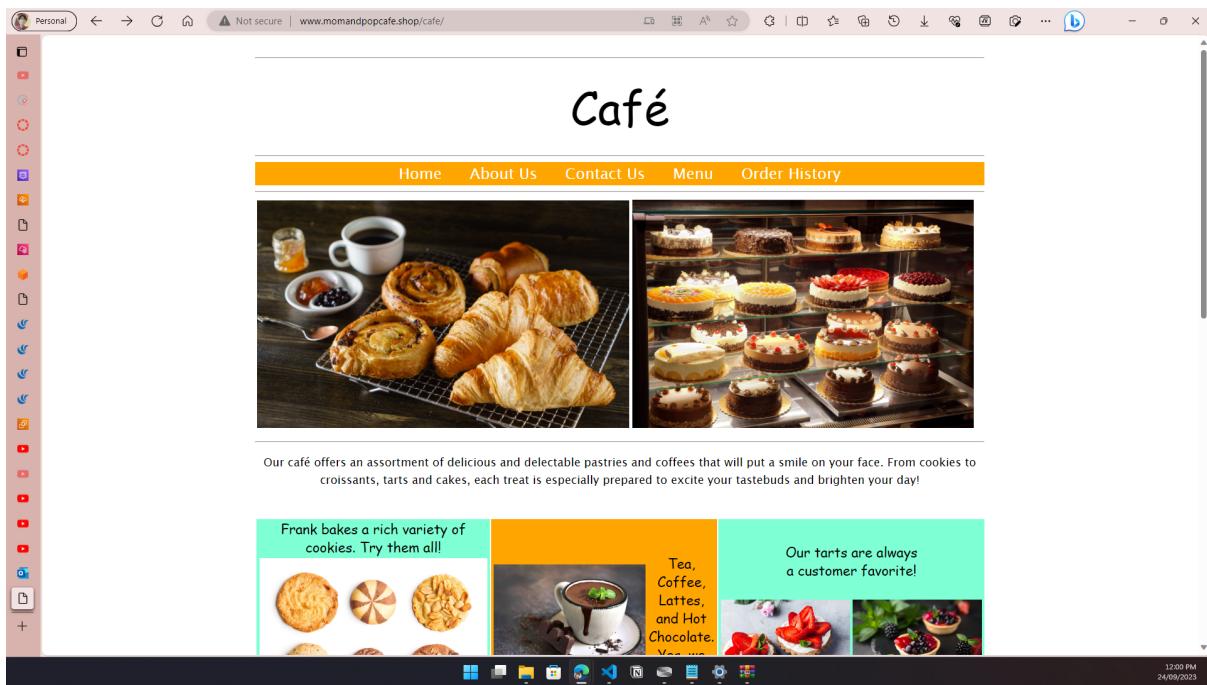
View existing records
The following table lists the existing records in momandpopcafe.shop.

Then add another record for a CNAME:

- Name: [www.momandpopcafe.shop](#)
- Value: momandpopcafe.shop

Then go to the DNS configuration of the domain name provider, add the name server (NS) of the 4 address listed in the hosted zones, and the required NS based on the provider.

Save, and wait for a while, the website will be able to access through the domain:
www.momandpopcafe.shop



*Update:

Later on, I added a CNAME record point toward the DNS endpoint of the application load balancer in tenten.vn's dns service.

2. Create an SNS Topic for sending emails

Create a standard topic, and an email subscription

New Feature
Amazon SNS FIFO topics now support message delivery to Amazon SQS Standard queues.

Amazon.SNS > Topics > Create topic

Create topic

Details

Type [Info](#)
Topic type cannot be modified after topic is created

FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - optional [Info](#)
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

Error: Health Check Error

Maximum 100 characters.

► **Encryption - optional** [Info](#)
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

► **Access policy - optional** [Info](#)
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Create subscription

Details

Topic ARN
arn:aws:sns:us-east-1:774371583665:momandpop-cafe-healthcheck-email-topic

Protocol
The type of endpoint to subscribe
Email

Endpoint
An email address that can receive notifications from Amazon SNS.
s3938278@rmit.edu.vn

After your subscription is created, you must confirm it. [Info](#)

▼ Subscription filter policy - optional [Info](#)
This policy filters the messages that a subscriber receives.
 Subscription filter policy

▼ Redrive policy (dead-letter queue) - optional [Info](#)
Send undeliverable messages to a dead-letter queue.
 Redrive policy (dead-letter queue)

[Cancel](#) [Create subscription](#)

3. Configure Route 53 Health Check

Create a health check with the following configuration:

- Specify endpoint by: Domain
- Protocol: HTTP
- Domain name: www.momandpopcafe.shop
- Port: 80
- Path: cafe

Step 1: Configure health check

Configure health check

Step 2: Get notified when health check fails

Route 53 health checks let you track the health status of your resources, such as web servers or mail servers, and take action when an outage occurs.

Name

What to monitor Endpoint
 Status of other health checks (calculated health check)
 State of CloudWatch alarm

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy.
[Learn more](#)

Specify endpoint by IP address Domain name

Protocol HTTP HTTPS

Domain name *

Port *

Path

Advanced configuration

URL

Health check type Basic - no additional options selected ([View Pricing](#))

* Required [Cancel](#) [Next](#)

Then select the previously created SNS topic.

The screenshot shows the 'Create health check' wizard, Step 2: Get notified when health check fails. It includes fields for creating an alarm (Yes selected) and sending notifications to an Existing SNS topic (selected). The topic name is 'momandpop-cafe-healthcheck-email-topic'. Buttons at the bottom include 'Cancel', 'Previous', and 'Create health check'.

4. Configure Route 53 failover routing to a static website

In the momandpopcafe.shop hosted zone, edit the A record to change

- Routing policy to failover
- Health Check: previously created health check
- RecordID: momandpopcafe.shop-primary

The screenshot shows the 'Hosted zone details' page for the 'momandpopcafe.shop' zone. The 'Records' section lists four records:

Record name	Type	Routing policy	Alias
momandpopcafe.shop	A	Simple	Yes
momandpopcafe.shop	NS	Simple	No
momandpopcafe.shop	SOA	Simple	No
www.momandpopcafe.shop	CNAME	Simple	No

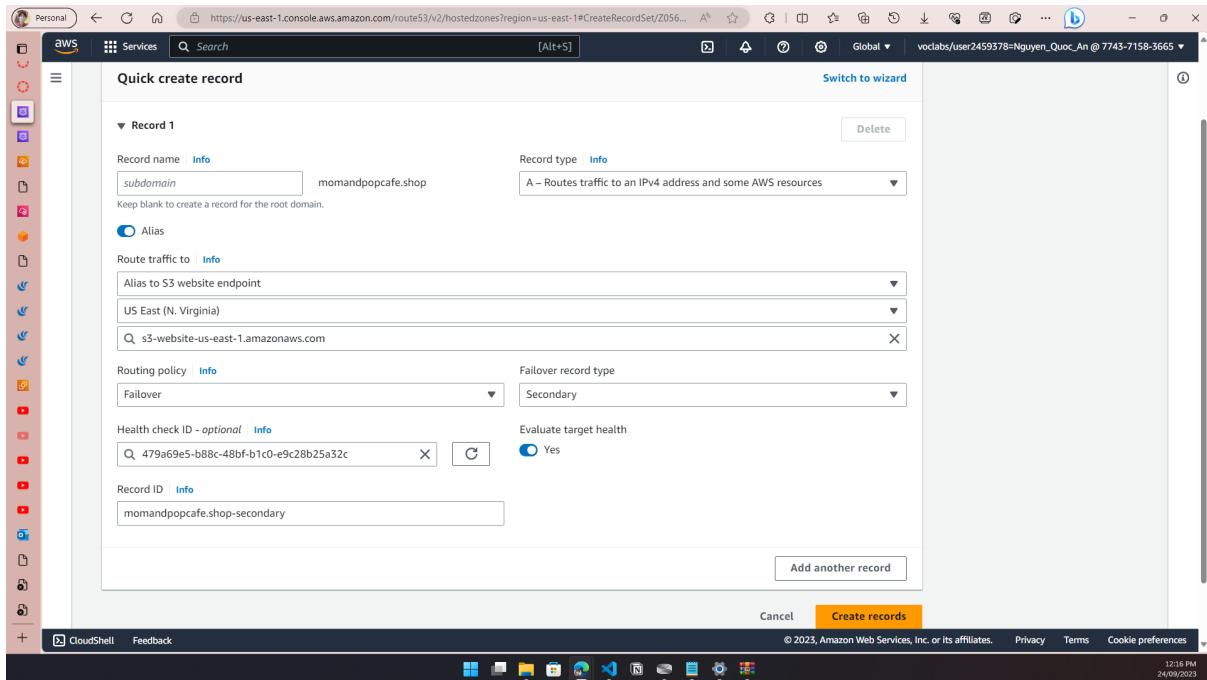
The right panel shows the configuration for the first A record:

- Subdomain: momandpopcafe.shop
- Record type: Info
- Route traffic to: Alias
- Alias to Application and Classic Load Balancer
- Region: US East (N. Virginia)
- Q: dualstack.momandpop-cafe-lb-8009975
- Routing policy: Info
- Failover
- Failover record type: Primary
- Health check ID: Info
- Q: 479a69e5-b88c-48bf-b1c0-e9c28b25a5
- Evaluate target health: No
- Record ID: Info
- momandpopcafe.shop-primary

Then add another A record:

- Routing policy to failover

- Enable alias, and route to S3 bucket created in task A
- Health Check: previously created health check
- RecordID: momandpopcafe.shop-secondary

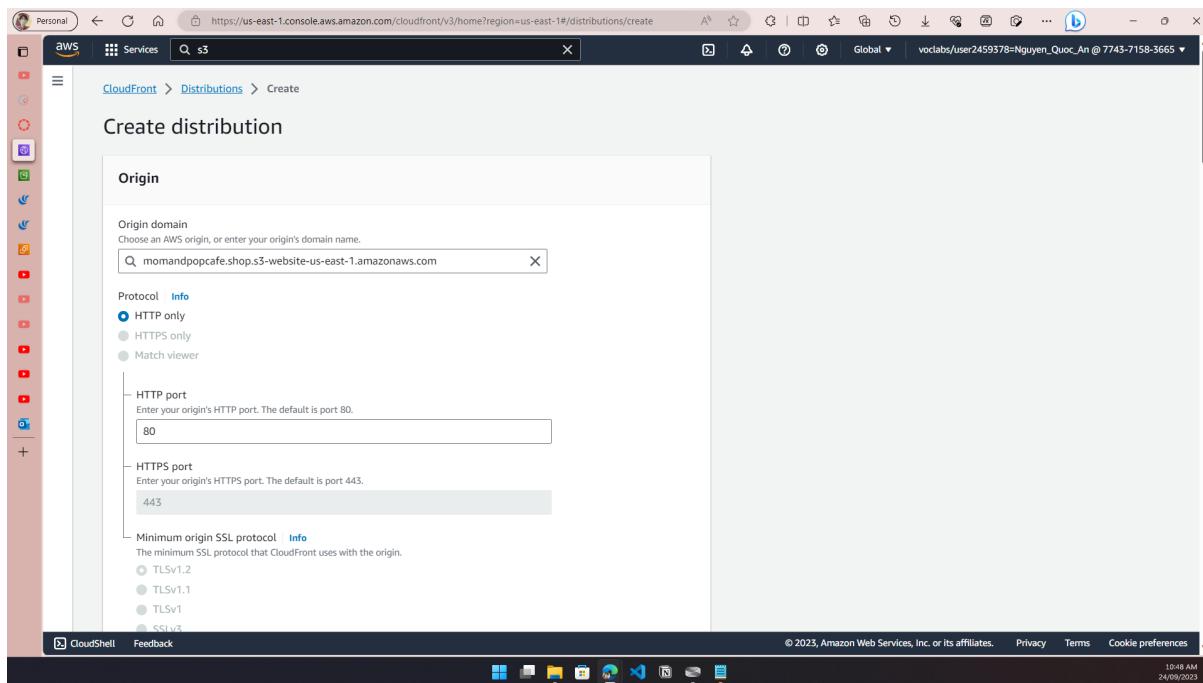


Task D - Caching with CloudFront

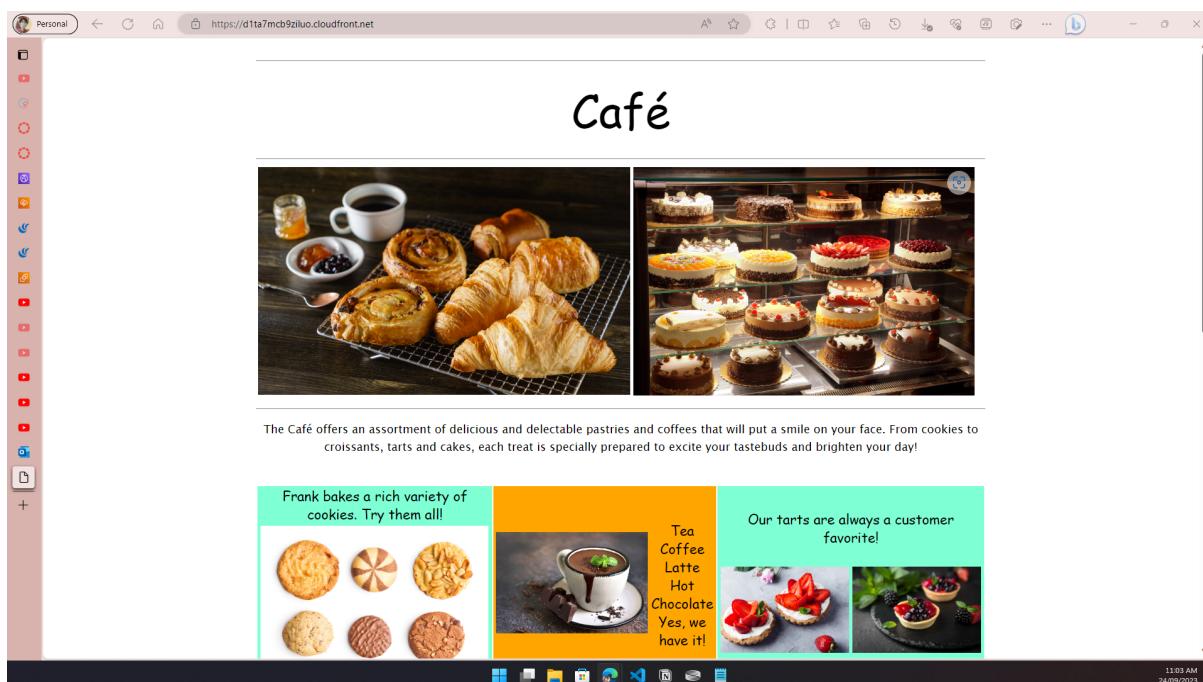
1. Cache static website in task A

Go to AWS CloudFront dashboard and create a distribution with the following config:

- Origin domain: Paste the DNS endpoint of the website in task A (momandpopcafe.shop.s3-website-us-east-1.amazonaws.com)
- Name: Momandpop-Cafe-Static-Website-Distribution
- Origin access: Public
- Default root object: index.html
- WAF: Do not enable security protections
- Leave the other settings as default



Wait for the distribution to be deployed, then the website can be accessed through the distribution domain name

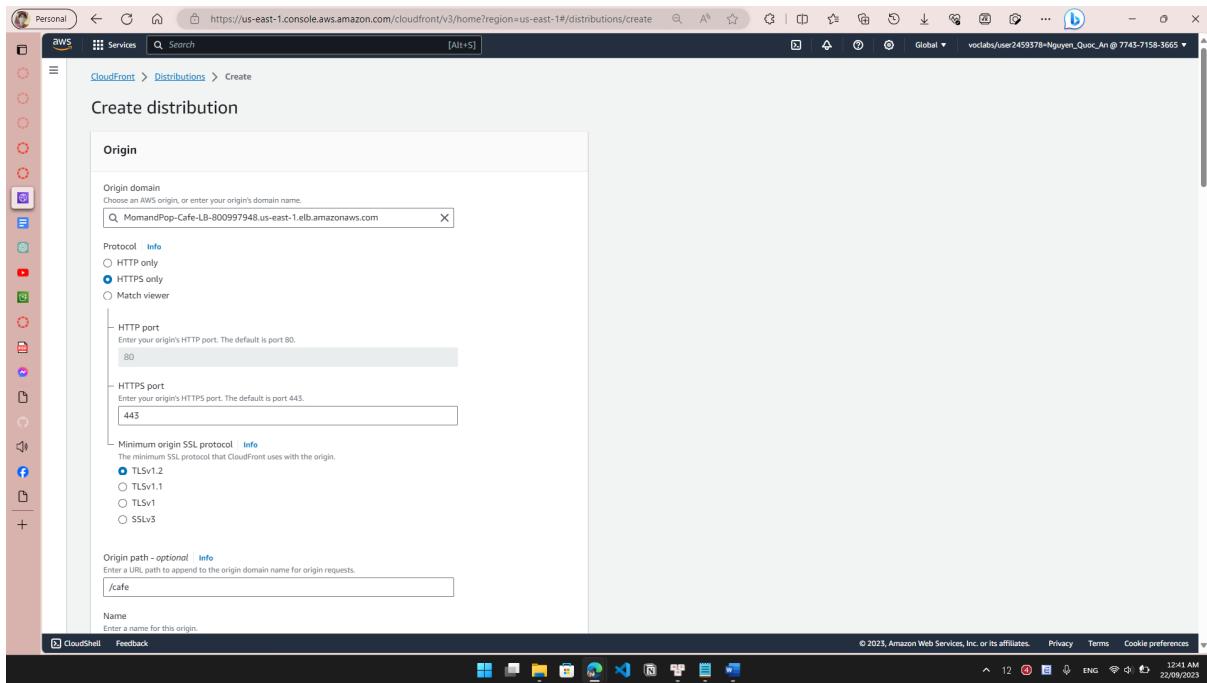


2. Cache dynamic website in task B

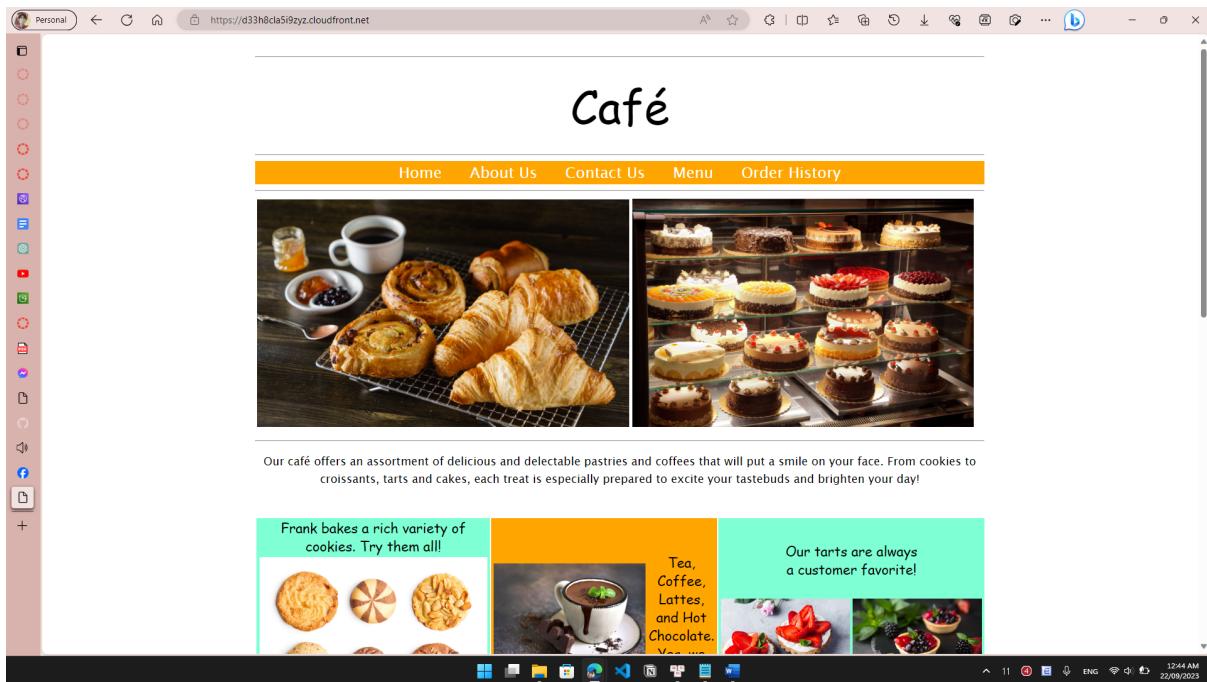
Go to AWS CloudFront dashboard and create a distribution with the following config:

- Origin domain: Choose the load balancer endpoint created in task B (MomandPop-Cafe-LB)
- Origin path: cafe
- Name: Automatic (MomandPop-Cafe-LB-800997948.us-east-1.elb.amazonaws.com)
- Cache policy: CachingOptimized

- WAS: Do not enable security protections
- Leave the other settings as default



Wait for the distribution to be successfully deployed, then the cached website can be accessed through the domain of the distribution.



Task E - Dynamic website with Elastic Beanstalk

1. Create an environment

Create an environment running php in Beanstalk, with LabRole and LabInstanceProfile.

The screenshot shows the 'Configure environment' step in the AWS Elastic Beanstalk console. On the left, a sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main area is titled 'Configure environment' and contains three sections: 'Environment tier' (selected as 'Web server environment'), 'Application information' (with application name 'momandpopcafe-server-environment'), and 'Environment information' (with environment name 'Momandpopcafe-server-environment-env').

Environment tier Info
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

Web server environment
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information Info

Application name

Maximum length of 100 characters.

► Application tags (optional)

Environment information Info
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain
 .us-east-1.elasticbeanstalk.com

Environment description

Platform Info

Platform type

Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Bundle the cafe file into a zip file and upload it.

The screenshot shows the 'Application code' section of the 'Create environment' wizard. The 'Upload your code' option is selected, and a file named 'cafe.tar.gz' is chosen from a local file. A version label '1' is specified.

The screenshot shows the 'Configure service access' step. The 'Service access' section is configured to use an existing service role ('LabRole') and an EC2 key pair ('vockey').

Select the 2 public subnets.

Personal

https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment [Alt+S] N. Virginia vclabs/user2459378=Nguyen_Quoc_An @ 7743-7158-3665

Step 1 [Configure environment](#)

Step 2 [Configure service access](#)

Step 3 - optional **Set up networking, database, and tags**

Step 4 - optional [Configure instance traffic and scaling](#)

Step 5 - optional [Configure updates, monitoring, and logging](#)

Step 6 [Review](#)

Virtual Private Cloud (VPC)

VPC

Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console.

[Learn more](#)

vpc-0e93e41b3d09296df | (10.0.0.0/16) | MomandPop-Cafe-vpc

[Create custom VPC](#)

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

Activated

Instance subnets

Filter instance subnets

Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/> us-east-1a	subnet-01...	10.0.0.0/20	MomandPop-Cafe-subnet-public1-us...
<input type="checkbox"/> us-east-1a	subnet-03...	10.0.128...	MomandPop-Cafe-subnet-private1-u...

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 11:01 AM 24/09/2023

Personal

https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment [Alt+S] N. Virginia vclabs/user2459378=Nguyen_Quoc_An @ 7743-7158-3665

Step 1 [Configure environment](#)

Step 2 [Configure service access](#)

Step 3 - optional **Set up networking, database, and tags**

Step 4 - optional [Configure instance traffic and scaling](#)

Step 5 - optional [Configure updates, monitoring, and logging](#)

Step 6 [Review](#)

Virtual Private Cloud (VPC)

VPC

Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console.

[Learn more](#)

vpc-0e93e41b3d09296df | (10.0.0.0/16) | MomandPop-Cafe-vpc

[Create custom VPC](#)

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

Activated

Instance subnets

Filter instance subnets

Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/> us-east-1a	subnet-01...	10.0.0.0/20	MomandPop-Cafe-subnet-public1-us...
<input type="checkbox"/> us-east-1a	subnet-03...	10.0.128...	MomandPop-Cafe-subnet-private1-u...

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 11:01 AM 24/09/2023

The screenshot shows the AWS Elastic Beanstalk console with the URL <https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment>. The left sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main content area is titled "Configure updates, monitoring, and logging - optional". Under "Monitoring", it says "Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The EnvironmentHealth custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#)". There are two radio button options: "Basic" and "Enhanced", with "Enhanced" selected. Below this are sections for "CloudWatch Custom Metrics - Instance" and "CloudWatch Custom Metrics - Environment", both with dropdown menus labeled "Choose metrics". A note about "Health event streaming to CloudWatch Logs" follows, mentioning retention up to ten years and log deletion on termination. It includes a checkbox for "Log streaming" and a dropdown for "Retention" set to 7 days. At the bottom is a "Lifecycle" section.

I'm not managed to fine a way for the website to run the php file, it seem like php 8 is newer version than the source code file.

Task F - Breaking a Monolithic Application into Microservices

Task G - SNS, SQS and CloudWatch Alarm

2. Create a standard SNS topic

Create a standard SNS topic, and add an email subscription

The screenshot shows the AWS SNS console with the URL <https://us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/create-topic>. The left sidebar shows "Amazon SNS > Topics > Create topic". The main content area has a "New Feature" banner: "Amazon SNS FIFO topics now support message delivery to Amazon SQS Standard queues. [Learn more](#)". The "Create topic" form has a "Details" section. Under "Type", "Info" is selected, and a note says "Topic type cannot be modified after topic is created". Two radio button options are shown: "FIFO (first-in, first-out)" and "Standard". The "Standard" option is selected, with a list of features: "Best-effort message ordering", "At-least once message delivery", "Highest throughput in publishes/second", and "Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints". Below this are fields for "Name" (set to "TopicAlarm") and "Display name - optional" (set to "Lab Topic Alarm"). At the bottom are "CloudShell" and "Feedback" buttons, and a footer with copyright information and links to Privacy, Terms, and Cookie preferences.

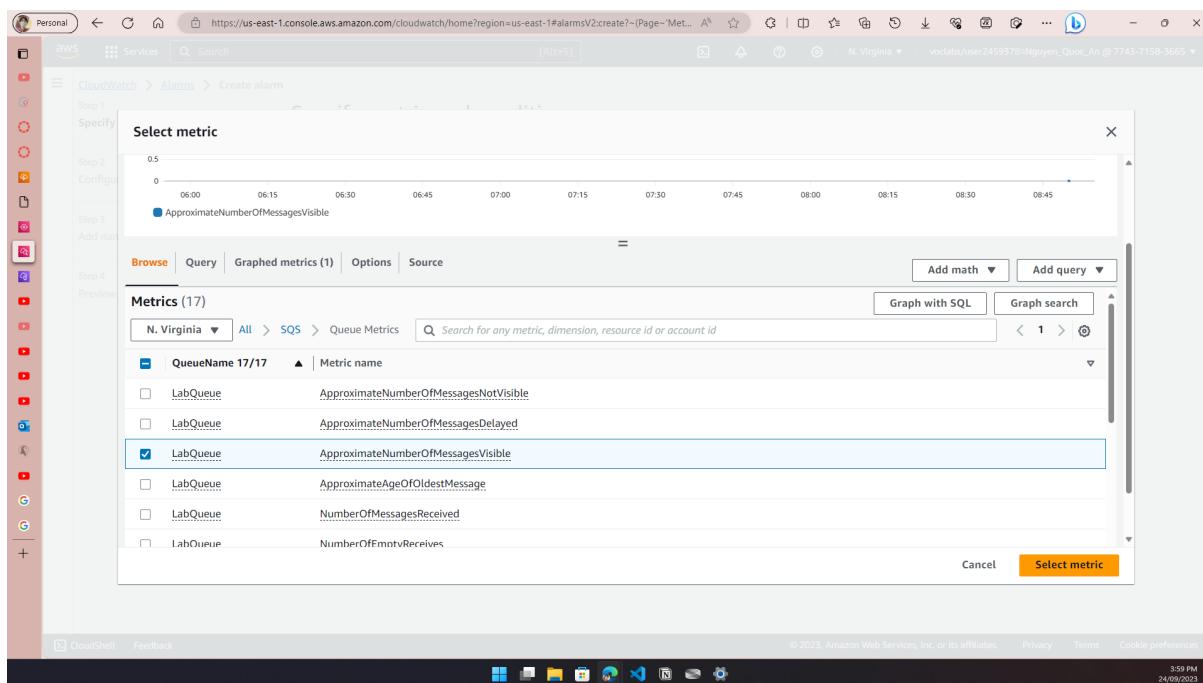
The screenshot shows the AWS SNS console with the 'TopicAlarm' topic selected. The 'Details' section displays the topic's name (TopicAlarm), display name (Lab Topic Alarm), ARN (arn:aws:sns:us-east-1:774371583665:TopicAlarm), and type (Standard). Below this, the 'Subscriptions' tab is active, showing one confirmed subscription to the email address s3938278@rmit.edu.vn.

3. Create a standard SQS

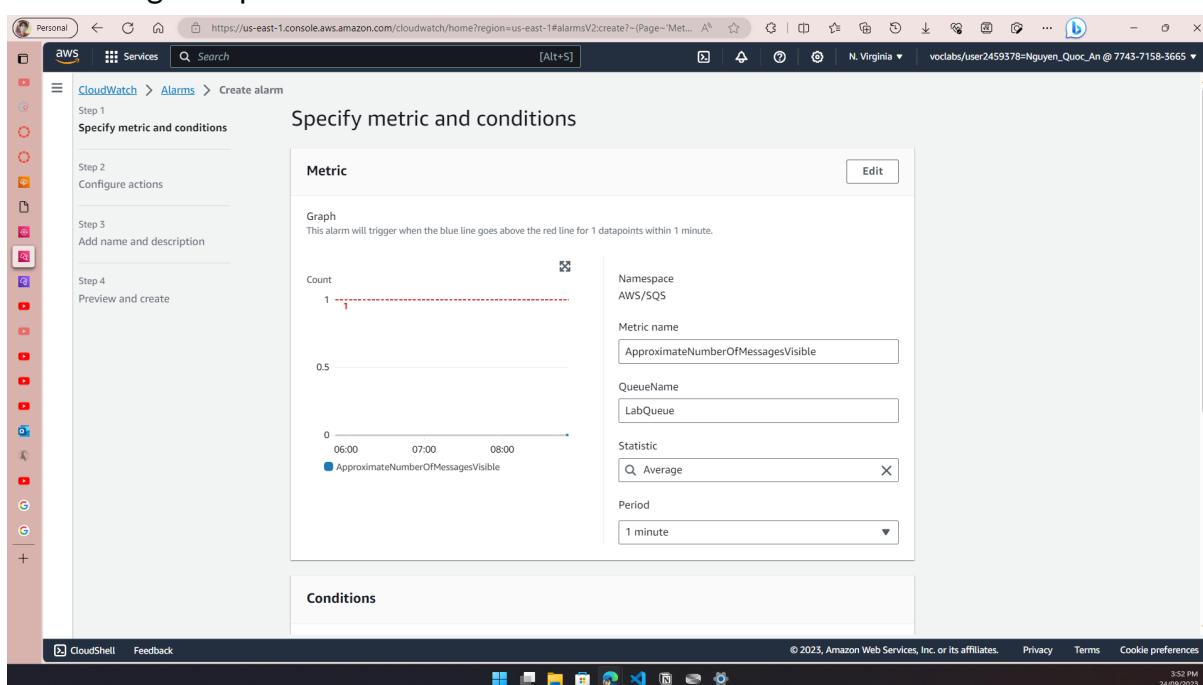
The screenshot shows the AWS SQS console with the 'Create queue' wizard. In the 'Details' step, the 'Type' section is expanded, showing two options: 'Standard Info' (selected) and 'FIFO Info'. 'Standard Info' describes at-least-once delivery and non-preserved message ordering. 'FIFO Info' describes first-in-first-out delivery and preserved message ordering. A note states that the queue type cannot be changed after creation. The 'Name' field is set to 'LabQueue'. In the 'Configuration' step, visibility timeout is set to 30 seconds and message retention period is set to 4 days.

4. Create a CloudWatch alarm

Go to AWS CloudWatch and create an alarm with LabQuese SQS and metric: ApproximateNumberOfMessageVisible.

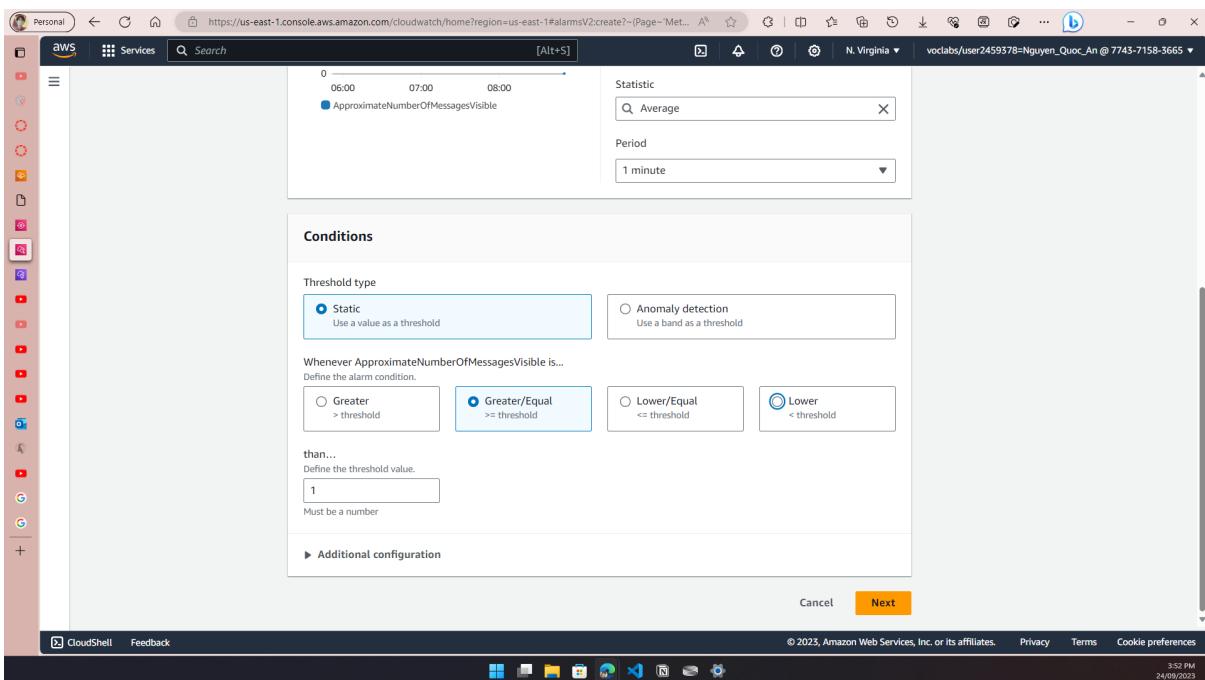


Then change the period to 1 minute

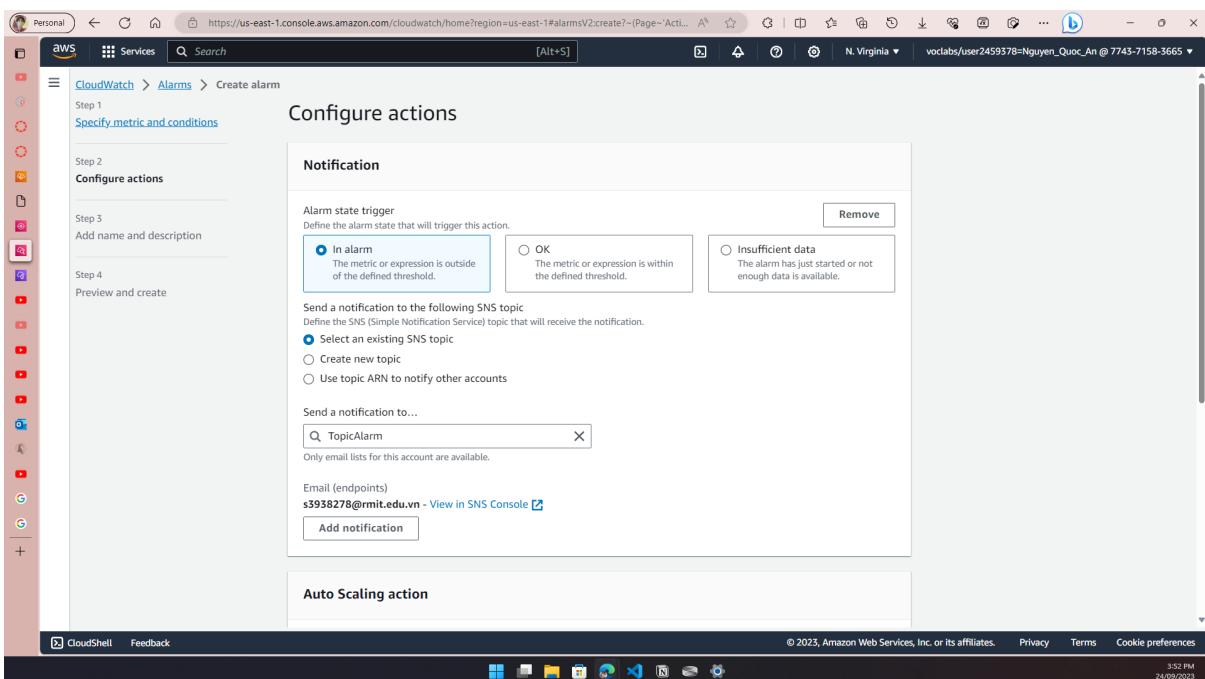


Threshold type: static

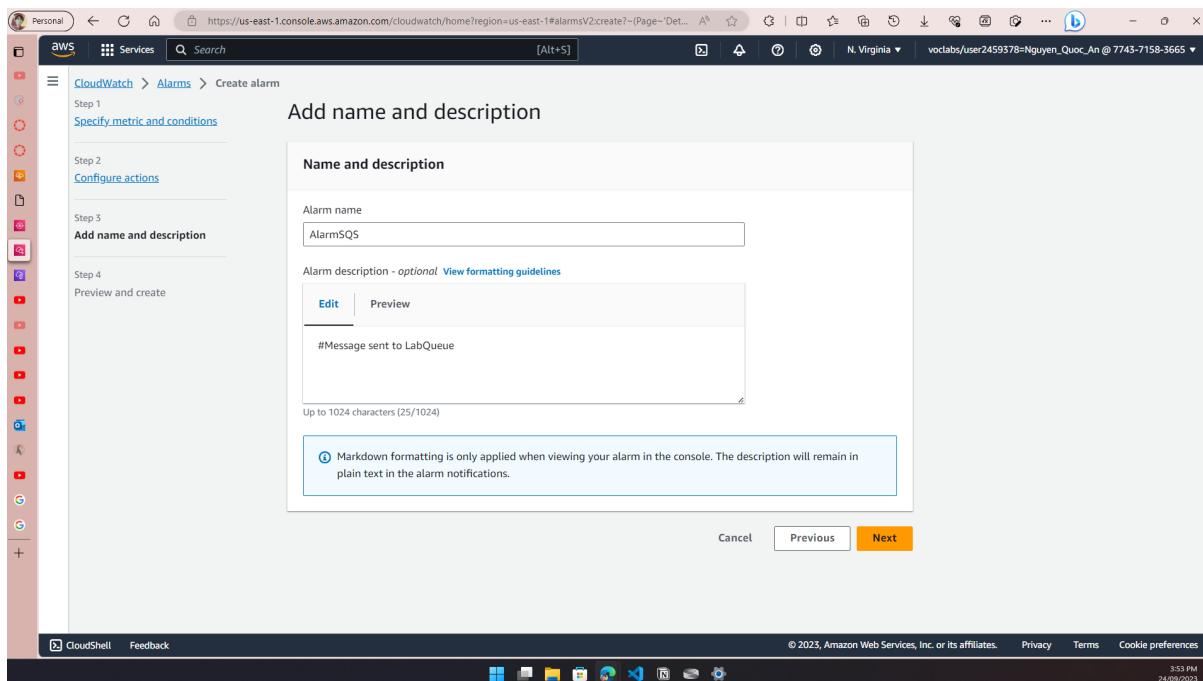
Whenever Greater/Equal than 1



Then select previously created TopicAlarm

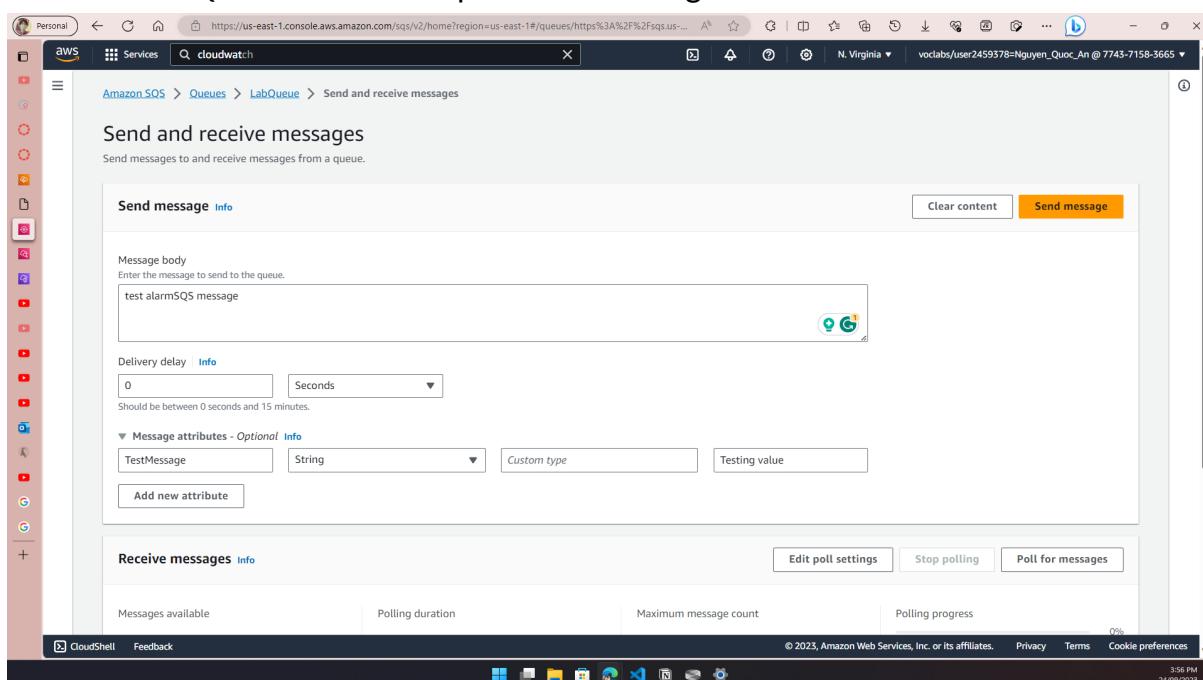


Name: AlarmSQS, and add a custom message and create the alarm.



5. Send and poll messages

Go to the LabQueue to send and poll a test message



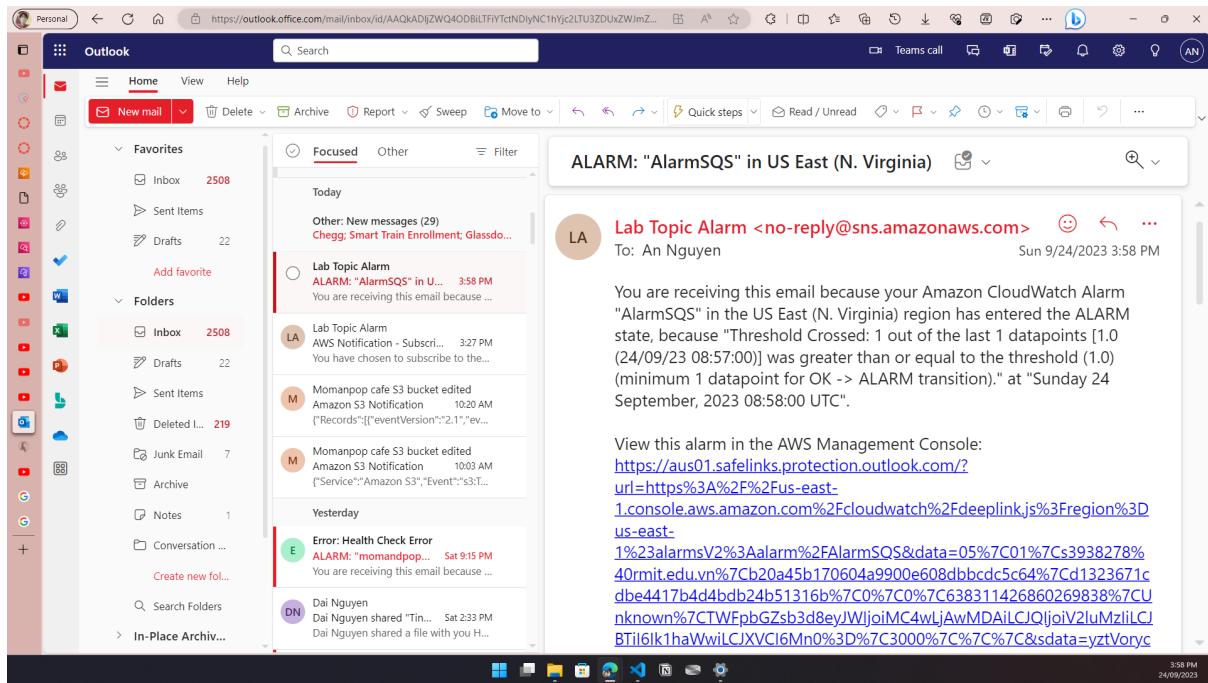
The status of the alarm will change to OK

The screenshot shows the AWS CloudWatch Alarms page. A green banner at the top indicates "Successfully created alarm AlarmSQS." The left sidebar has a "CloudWatch" tab selected, showing various monitoring options like Dashboards, Alarms, Logs, Metrics, and X-Ray traces. The "Alarms" section is expanded, showing 1 alarm in progress and 2 more pending. The main table lists four alarms:

Name	State	Last state update	Conditions	Actions
AlarmSQS	OK	2023-09-24 08:55:00	ApproximateNumberOfMessagesVisible >= 1 for 1 datapoints within 1 minute	Actions enabled
momandpop-cafe-health-check-awsroute53-479a69e5-b88c-48bf-b1c0-e9c28b25a32-Low-HealthCheckStatus	OK	2023-09-24 05:56:40	HealthCheckStatus < 1 for 1 datapoints within 1 minute	Actions enabled
TargetTracking-MomandPop-Cafe-ASG-AlarmLow-426c3e44-dc76-4b83-9a37-be81cd539f9	In alarm	2023-09-22 09:15:16	CPUUtilization < 17.5 for 15 datapoints within 15 minutes	Actions enabled
TargetTracking-MomandPop-Cafe-ASG-AlarmHigh-10e8fd8a-e563-457f-9090-37ddb0f406b	OK	2023-09-19 04:47:01	CPUUtilization > 25 for 3 datapoints within 3 minutes	Actions enabled

After 1 minute, the alarm will turn into in alarm, and the email will be sent

This screenshot is identical to the one above, showing the AWS CloudWatch Alarms page. The green banner still says "Successfully created alarm AlarmSQS." The "Alarms" section shows 1 alarm in progress and 2 more pending. The main table lists the same four alarms. The difference is that the first alarm, "AlarmSQS", is now in an "In alarm" state, indicated by a red triangle icon next to its name. The other three alarms remain in their previous states (OK).



Task H - Okami Restaurant

Okami have 41 restaurant branches over 4 states of the Australia, which is relative large geographical area, so first of all to migrate all the existing restaurant website the AWS Cloud. Okami should implement AWS Route 53 service and geolocation routing as an entry to route all of their request to the nearest server, to ensure the lowest latency for their customer, as well as failover to another server in case one is down.

Next to store and deploy their current website, they can run it on AWS EC2 instances due to their scalability and can be used for general purposes. Since the restaurant chain have 41 branches, it is assumed that traffic will be huge, and spiky especially in eating hours for instance dinner time. So it is suggested that Okami restaurant's website to be placed in to an auto scaling group, that scale by average CPU utilization which reflects the incoming traffic. Scale-in should also be enabled, to make sure the EC2 resources will scale out to serve the high traffic but scale in on lazy hour to save cost. This auto scaling group is also recommended to span over 3 AZs to further increase the website's high availability. This group will be accessed through an application load balancer, so traffic will be spread evenly. Finally the user will access the website by the DNS endpoint of CloudFront aliasing to the load balancer endpoint to assure highest loading speed.

To implement the reservation and order systems and connect with the restaurant's cloud. I recommend Okami to implement them as a Lambda functions, so that it is cheaper, and more time efficient for them as Lambda is serverless service so they do not need to manage those system, also they will only be charged when there is an order

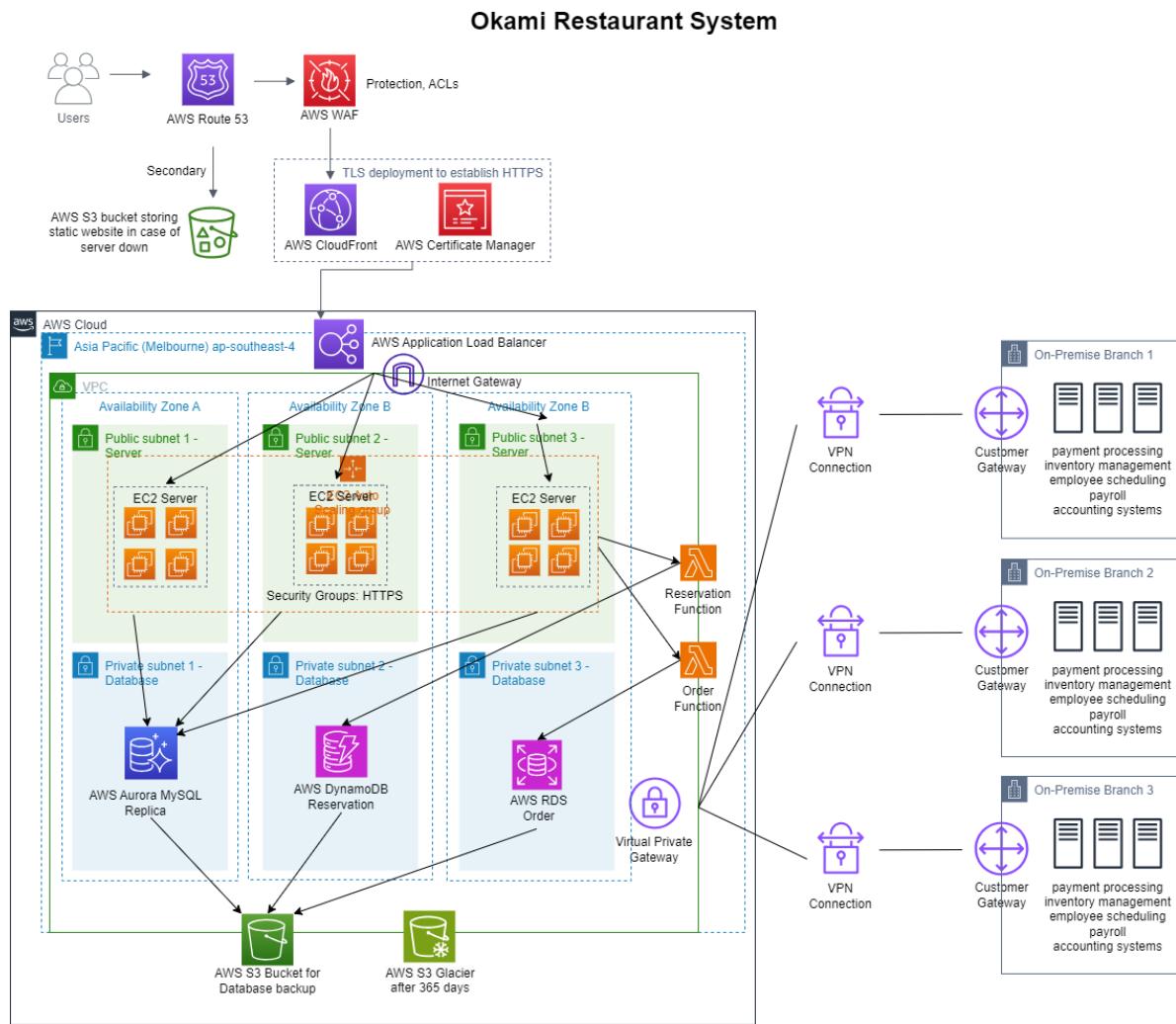
or booking being made, as guests might access the restaurant website for discovering but not booking and ordering anything.

In the case of reservation system, I suggest Okami to integrated Lambda, and save the reservation data to AWS DynamoDB, since this database service offer fast, reliable data read and write performance. It is very flexible as a NoSQL database, which is particularly advantageous for the system, as customers tend to not providing all complete details when filling the reservation form, reducing the risk of system error and simplify system's complexity.

For ordering system, I also suggest Okami to integrate Lambda, but data will be saved to AWS RDS with it's nature of SQL database and strict schema will ensure a strict data integrity for every order being made throughout operations.

All working data generated into the databases, will be backed up annually into an S3 bucket, with life-cycle policy to switch those data in to S3 Glacier for cost saving if data is not accessed in 365 days. Additionally, all Cloudwatch logs will also be stored in those buckets.

Lastly, to established a connection between the central cloud-based system and the local system of every branches. I suggest every branch on premise system connect to the cloud using AWS site-to-site VPN. This will use the IPSec protocol to established a secure VPN tunnels between the cloud and the branch system, and each branch can access through their own customer gateway. This approach will ensure connection of all branches's local system to the cloud without compromising any data privacy and security.



Task I - Capstone Project

The user from the internet will access the website through the application load balancer, placing public, with the ALBSG security Group allow HTTP/HTTPS traffic. This traffic will route to the EC2 auto scaling group placed in 2 public subnets, this use Inventory-App security group, which only allows HTTP/HTTPS traffic from the ALBSG group. Its also allow SSH remote connect traffic on port 22 from Bastion-SG bastion host for dumping database to RDS instance. Those ec2 instances will access the database and run query to the RDS database instance placed in private subnet and Example-DBSG, which allows MySQL/Aurora traffic from Inventory-App security group. The credential to access the RDS database is saved on the System Manager parameter store and retrieved by the EC2 instance.

