

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

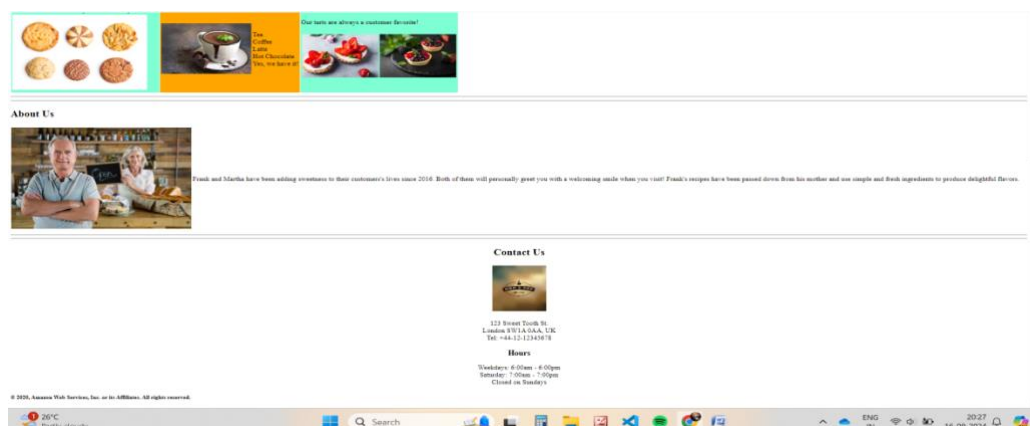
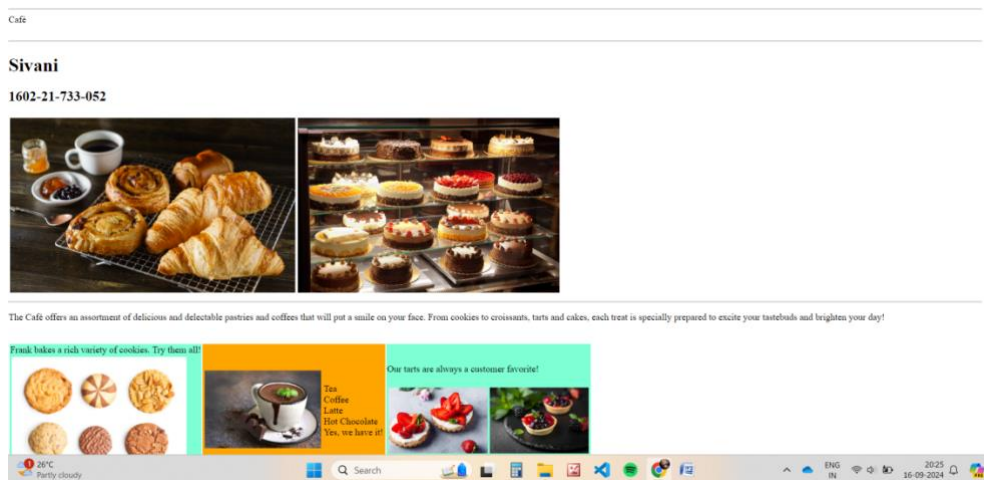
Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS-1

Creating a Static Website for the Cafe:

1. Extracting the files that you need for this lab
2. Creating an S3 bucket to host your static website
 1. **Open the Amazon S3 console.**
Create a bucket in the US East (N. Virginia) us-east-1 AWS Region to host your static website.
Tip: You must clear Block *all* public access and enable ACLs.
 2. **Enable static website hosting on your bucket.**
Tip: You use the index.html file for your index document.
3. Uploading content to your S3 bucket
 1. Upload the index.html file and the CSS and images folders to your S3 bucket.
 2. In a separate web browser tab, open the endpoint link for your static website.
4. Creating a bucket policy to grant public read access
 1. Create a bucket policy that grants read-only permission to public anonymous users by using the bucket policy editor.
 2. Confirm that the website for the café is now publicly accessible.

Output:



VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS-2: Introduction to Amazon EC2

1. Launch Your Amazon EC2 Instance:

- In the AWS Management Console choose Services, choose Compute and then choose EC2.
- Choose the Launch instance menu and select Launch instance.
- Give the instance the name **Web Server**
- In the list of available *Quick Start* AMIs, keep the default Amazon Linux AMI selected.
- Also keep the default Amazon Linux 2023 AMI selected.
- In the *Instance type* panel, keep the default t2.micro selected.
- For Key pair name - *required*, choose vockey.
- Next to Network settings, choose Edit. For VPC, select Lab VPC.
- Under Firewall (security groups), choose Create security group and configure:
- In the *Configure storage* section, keep the default settings. Expand Advanced details.
- For Termination protection, select Enable.
- Scroll to the bottom of the page and then copy and paste the code shown below into the User data box:

```
#!/bin/bash
dnf install -y httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
```

At the bottom of the Summary panel choose Launch instance

2. Monitor Your Instance

- Choose the **Status checks** tab: Notice that both the **System reachability** and **Instance reachability** checks have passed.
- Choose the **Monitoring** tab: Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (five-minute) monitoring is enabled by default. You can also enable detailed (one-minute) monitoring.
- In the Actions menu towards the top of the console, select **Monitor and troubleshoot Get system log**.
- Scroll through the output and note that the HTTP package was installed from the **user data** that you added when you created the instance.
- Choose **Cancel**.
- Ensure **Web Server** is still selected. Then, in the Actions menu, select **Monitor and troubleshoot Get instance screenshot**.
- Choose **Cancel**.

3. Update Your Security Group and Access the Web Server

- Ensure **Web Server** is still selected. Choose the **Details** tab.
- Copy the **Public IPv4 address** of your instance to your clipboard.
- Open a new tab in your web browser, paste the IP address you just copied, then press **Enter**.
- Keep the browser tab open, but return to the **EC2 Console** tab.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

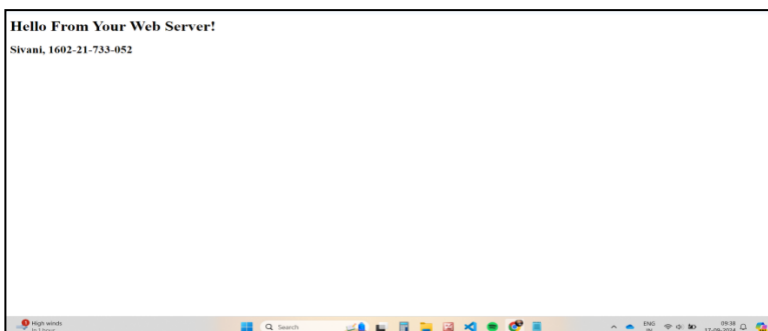
DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

- In the left navigation pane, choose **Security Groups**.
 - Select **Web Server security group**. Choose the **Inbound rules** tab.
 - Choose **Edit inbound rules**, select **Add rule** and then configure: **Type: HTTP**, **Source: Anywhere-IPv4**, Choose **Save rules**
 - Return to the web server tab that you previously opened and refresh the page.
4. Resize Your Instance: Instance Type and EBS Volume
- Stop Your Instance
 1. On the **EC2 Management Console**, in the left navigation pane, choose **Instances** and then select the **Web Server** instance.
 2. In the **Instance state** menu, select **Stop instance**. Choose **Stop**
 3. Your instance will perform a normal shutdown and then will stop running.
 4. Wait for the **Instance state** to display: *Stopped*.
 - Change The Instance Type and enable stop protection
 1. Select the Web Server instance, then in the **Actions** menu, select **Instance settings Change instance type**, then configure: **Instance Type: t2.small**, Choose **Apply**
 2. Select the Web Server instance, then in the **Actions** menu, select **Instance settings Change stop protection**. Select **Enable** and then **Save** the change.
5. Resize the EBS Volume
- With the Web Server instance still selected, choose the **Storage** tab, select the name of the Volume ID, then select the checkbox next to the volume that displays.
 - In the **Actions** menu, select **Modify volume**.
 - The disk volume currently has a size of 8 GiB. You will now increase the size of this disk.
 - Change the size to: **10** **NOTE:** You may be restricted from creating Amazon EBS volumes larger than 10 GB in this lab. Choose **Modify**. Choose **Modify** again to confirm and increase the size of the volume.
6. Start the Resized Instance
- In left navigation pane, choose **Instances**. Select the **Web Server** instance.
 - In the **Instance state** menu, select **Start instance**.

Output:



VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS-3: Introducing Amazon Elastic File System (Amazon EFS)

1. Creating a security group to access your EFS file system

- At the top of the AWS Management Console, in the search box, search for and choose **EC2**.
- In the navigation pane on the left, choose **Security Groups**.
- Copy the **Security group ID** of the *EFSClnt* security group to your text editor.
- The Group ID should look similar to *sg-03727965651b6659b*.
- Choose **Create security group** then configure: **Security group name**: *EFS Mount Target*, **Description**: *Inbound NFS access from EFS clients*, **VPC**: *Lab VPC*
- Under the **Inbound rules** section, choose **Add rule** then configure: **Type**: *NFS*,
 1. **Source**:
 1. *Custom*
 2. In the *Custom* box, paste the security group's **Security group ID** that you copied to your text editor
 2. Choose **Create security group**.

2. Creating an EFS file system

At the top of the AWS Management Console, in the search box, search for and choose **EFS**.

- Choose **Create file system**.
- In the **Create file system** window, choose **Customize**.
- On **Step 1**:
 1. Uncheck **Enable Automatic backups**.
 2. **Lifecycle management**: for **Transition into IA** Select *None*.
 3. In the **Tags optional** section, configure:
 1. **Key**: *Name*
 2. **Value**: *My First EFS File System*
- Choose **Next**.
- For **VPC**, select *Lab VPC*.
- Detach the default security group from each *Availability Zone* mount target by choosing the check box on each default security group.
- Attach the **EFS Mount Target** security group to each *Availability Zone* mount target by choosing **EFS Mount Target** for each *Availability Zone*.
- Choose **Next**.
- On **Step 3**, choose **Next**.
- On **Step 4**: Review your configuration. Choose **Create**.

3. Connecting to your EC2 instance

- To connect to the **EC2 instance**, from the top of this page, choose **i AWS Details** and copy the value for *InstanceSessionURL*.
- Paste it into the new browser tab or window to connect to the EC2 instance using AWS Systems Manager Session Manager.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

4. Creating a new directory and mounting the EFS file system

- In your EC2 terminal session, run the following command to install the required utilities:

```
sudo su -l ec2-user
```

```
sudo yum install -y amazon-efs-utils
```

- Run the following command to create directory for mount: `sudo mkdir efs`.
- At the top of the AWS Management Console, in the search box, search for and choose **EFS**.
- Choose **My First EFS File System**.
- In the **Amazon EFS Console**, on the top right corner of the page, choose **Attach** to open the Amazon EC2 mount instructions.
- In your EC2 terminal session, Copy and run the entire command in the **Using the NFS client** section.
- The mount command should look similar to this example:
- `sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-bce57914.efs.us-west-2.amazonaws.com:/ efs`
- The provided `sudo mount...` command uses the default Linux mount options.
- Get a full summary of the available and used disk space usage by entering: `sudo df -hT`
- Notice the *Type* and *Size* of your mounted EFS file system, similar to the following.
`fs-0e2e45d50de5916b3.efs.us-east-1.amazonaws.com:/ nfs4 8.0E 0 8.0E 0% /home/ec2-user/efs`

5. Examining the performance behavior of your new EFS file system

- Examine the write performance characteristics of your file system by entering:
`sudo fio --name=fio-efs --filesize=10G --filename=./efs/fio-efs-test.img --bs=1M --nrfiles=1 --direct=1 --sync=0 --rw=write --iodepth=200 --ioengine=libaio`
- The `fio` command will take few minutes to complete. The output should look like the example in the following screenshot. Make sure that you examine the output of your `fio` command, specifically the summary status information for this WRITE test.
- Monitoring performance by using Amazon CloudWatch
 - At the top of the AWS Management Console, in the search box, search for and choose **CloudWatch**.
 - In the navigation pane on the left, choose **All Metrics**.
 - In the **All metrics** tab, choose **EFS**.
 - Choose **File System Metrics**.
 - Select the row that has the **PermittedThroughput** Metric Name.
 - You might need to wait 2–3 minutes and refresh the screen several times before all available metrics, including **PermittedThroughput**, calculate and populate.
 - On the graph, If you do not see the line graph, adjust the time range of the graph down to **1h** to display the period during which you ran the `fio` command.
 - Note the Peak *Throughput* Value on the Y-axis (Bytes/Second) line in the graph. The value should be around **3G**.
 - The throughput of Amazon EFS scales as the file system grows. File-based workloads are typically spiky. They drive high levels of throughput for short periods of time, and low levels of throughput the rest of the time. Because of this behavior, Amazon EFS is designed to burst to high throughput levels for periods of time.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

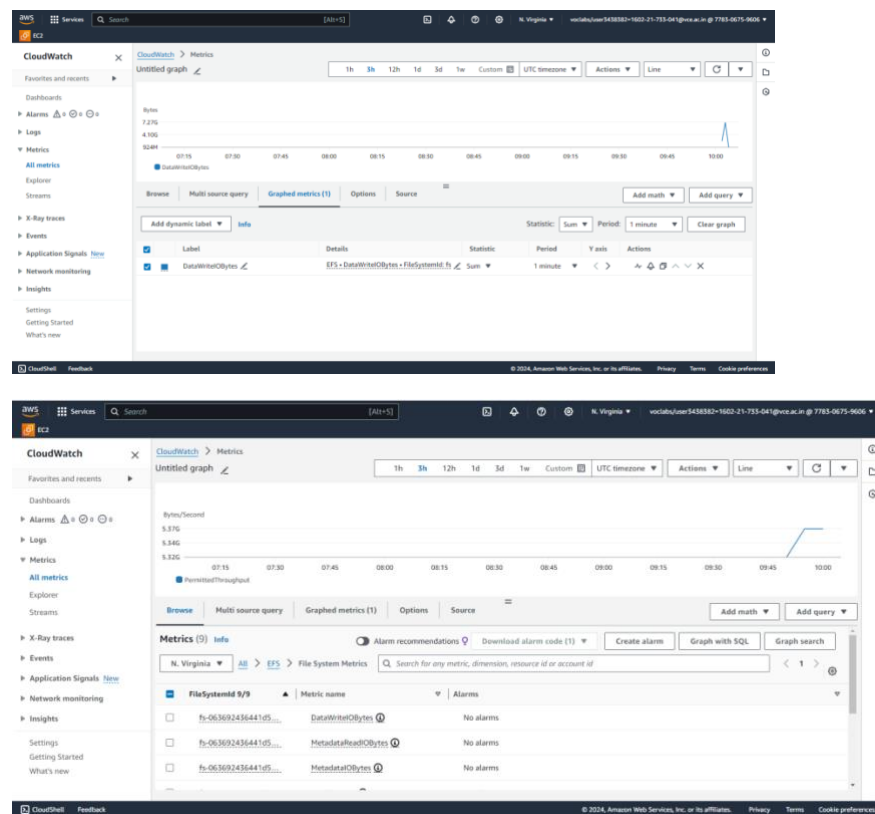
DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

- In the **All metrics** tab, *uncheck* the box for **PermittedThroughput**.
- Select the check box for **DataWriteIOBytes**.
- If you do not see *DataWriteIOBytes* in the list of metrics, use the **File System Metrics** search to find it.
- Choose the **Graphed metrics** tab.
- On the **Statistics** column, select **Sum**.
- On the **Period** column, select **1 Minute**.
- Note the the peak value, which is around 7.6G. Take this number (in bytes) and divide it by the duration in seconds (60 seconds). The result gives you the write throughput (B/s) of your file system during your test.
- The throughput that is available to a file system scales as a file system grows. All file systems deliver a consistent baseline performance of 50 MiB/s per TiB of storage. Also, all file systems (regardless of size) can burst to 100 MiB/s. File systems that are larger than 1T B can burst to 100 MiB/s per TiB of storage. As you add data to your file system, the maximum throughput that is available to the file system scales linearly and automatically with your storage.
- File system throughput is shared across all EC2 instances that are connected to a file system. For more information about performance characteristics of your EFS file system, see the documentation link in the resources section.
- With EFS you can also create access points for application-specific entry points into an EFS file system to provide secured access to shared datasets. Access points can enforce a user identity, including the user's POSIX groups, for all file system requests that are made through the access point. Refer to the section at the bottom for additional information.

Output:



VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS-4: Creating an Amazon RDS database

Task 1: Creating an Amazon RDS database

1. At the top of the AWS Management Console, in the search box, enter and select **RDS**.
2. Choose **Create database**.
3. Under **Engine options**, select **MySQL**.
4. Set the templates and availability and durability options:
5. Under the **Templates** section, select **Dev/Test**.
6. Under the **Availability and durability** section, select **Single DB instance**
7. Under the **Settings** section, configure these options:
DB instance identifier: inventory-db
Master username: admin
8. Under **Credentials management**, choose **Self managed** and configure as follows:
Master password: lab-password
Confirm master password: lab-password
9. Under the **Instance configuration** section, configure these options:
Select **Burstable classes (includes t classes)**.
Select **db.t3.micro**
10. In the **Storage** section next
For **Storage type** choose **General Purpose SSD (gp2)** from the Dropdown menu.
For **Allocated storage** enter 20.
11. Expand **Storage autoscaling**
Clear or Deselect **Enable storage autoscaling**.
12. Under the **Connectivity** section, configure these options:
Virtual Private Cloud (VPC): Lab VPC
DB subnet group: Keep the default selection
- Existing VPC security groups:**
12. Choose **DB-SG**. It will be highlighted.
Remove the *default* security group.
Under **Monitoring** section, Clear (turn off) the **Enable Enhanced monitoring** option
13. Expand the **Additional configuration** panel, then configure these settings:
Initial database name: inventory
14. Choose **Create database**

Task 2: Configuring web application communication with a database instance

15. At the Top of these instructions, from the **i AWS Details** section, copy the value for **AppServerPublicIP**.
16. Open a new web browser tab, paste the IP address you copied into the address bar, and then press ENTER.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

The web application should appear. It does not display much information because the application is not yet connected to the database.

17. Choose **Settings**.

You can now configure the application to use the RDS DB instance you created earlier. You will first retrieve the **Database Endpoint** so that the application knows how to connect to a database.

18. Return to the **AWS Management Console**, but do not close the application tab. (You will return to it soon).

19. From the **Services** menu, choose **RDS** to open the RDS console.

20. In the left navigation pane, choose **Databases**.

21. Choose **inventory-db**.

22. Go to the **Connectivity & Security** section and copy the **Endpoint** to your clipboard.

It should look similar to this example: *inventory-db.crwxbgqad61a.rds.amazonaws.com*

23. Return to the browser tab with the Inventory application, and enter these values:

○ **Endpoint:** Paste the endpoint you copied earlier

○ ****Database:**** `inventory`

○ ****Username:**** `admin`

○ ****Password:**** `lab-password`

○ Choose ****Save****

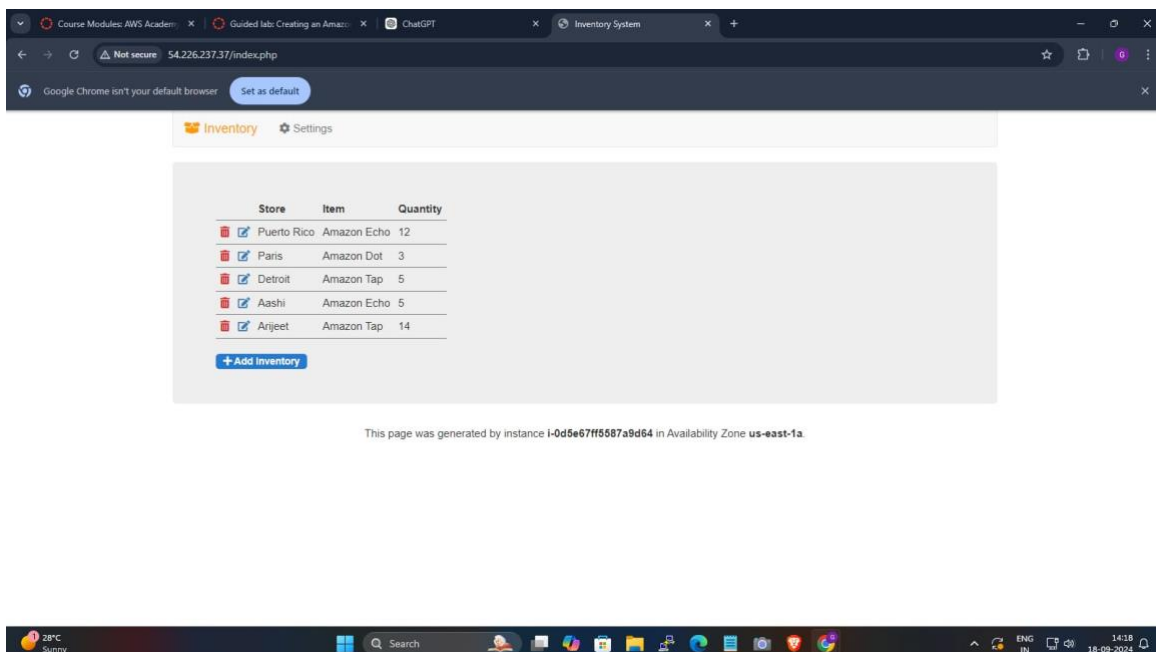
24. Add inventory, edit, and delete inventory information by using the web application.

The inventory information is stored in the Amazon RDS MySQL database that you created earlier in the lab. This means that any failure in the application server will *not* lose any data. It also means that multiple application servers can access the same data.

25. Insert new records into the table. Ensure that the table has 5 or more inventory records before submitting your work.

You have now successfully launched the application and connected it to the database!

Output:



VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS- 5 : Virtual Private Cloud

Task 1: Creating a VPC

1. On the **AWS Management Console**, in the search box, enter and choose **VPC** to open the Amazon VPC console.
2. In the left navigation pane, choose **Your VPCs**.
3. Choose **Create VPC**.
4. On the **Create VPC** page, configure the following options:
 - **Name tag - optional:** Enter **Lab VPC**.
 - **IPv4 CIDR:** Enter **10.0.0.0/16**.
4. Choose **Create VPC**.
5. Choose the **Tags** tab.
6. Choose **Actions**, and choose **Edit VPC settings**.
7. For **DNS settings**, select **Enable DNS hostnames**.
8. Choose **Save**.

Task 2: Creating subnets

Task 2.1: Creating a public subnet

9. In the left navigation pane, choose **Subnets**.
10. Choose **Create subnet**.
11. On the **Create subnet** page, configure the following options:
 - **VPC ID:** Choose **Lab VPC**.
 - **Subnet name:** Enter **Public Subnet**.
 - **Availability Zone:** Choose the first Availability Zone in the list. Do not keep **No preference** as the default.
 - **IPv4 subnet CIDR block:** Enter **10.0.0.0/24**.
12. Choose **Create subnet**.
13. Select **Public Subnet**.
14. Choose **Actions**, and then choose **Edit subnet settings**.
15. On the **Edit subnet settings** page, for **Auto-assign IP settings**, select **Enable auto-assign public IPv4 address**.
16. Choose **Save**.

Task 2.2: Creating a private subnet

17. Choose **Create subnet**.
18. On the **Create subnet** page, configure the following options:
 - **VPC ID:** Choose **Lab VPC**.
 - **Subnet name:** Enter **Private Subnet**.
 - **Availability Zone:** Choose the first Availability Zone in the list. Do not keep **No preference** as the default.
 - **IPv4 subnet CIDR block:** Enter **10.0.2.0/23**
19. Choose **Create subnet**.

Task 3: Creating an internet gateway

20. In the left navigation pane, choose **Internet gateways**.
21. Choose **Create internet gateway**.
22. For **Name tag**, enter **Lab IGW**.
23. Choose **Create internet gateway**. You can now attach the internet gateway to your Lab VPC.
24. Choose **Actions**, and then choose **Attach to VPC**.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

25. For **Available VPCs**, choose **Lab VPC**.

26. Choose **Attach internet gateway**.

Task 4: Configuring route tables

27. In the left navigation pane, choose **Route tables**.

28. Expand the **VPC** column so that you can see which one is used by **Lab VPC**.

29. Select the route table that shows **Lab VPC**.

30. In the **Name** column, choose the edit icon (), and for **Edit Name**, enter **Private Route Table**.

31. Choose **Save**.

32. Choose the **Routes** tab.

33. Choose **Create route table**, and configure these settings:

- **Name - optional:** Enter **Public Route Table**.
- **VPC:** Choose **Lab VPC**.

34. Choose **Create route table**.

35. In the **Routes** tab, choose **Edit routes**.

36. Choose **Add route**, then configure these settings:

- **Destination:** Enter **0.0.0.0/0**.
- **Target:** Choose **Internet Gateway** and then, from the list, choose **Lab IGW**.

37. Choose **Save changes**.

38. Choose the **Subnet associations** tab.

39. Choose **Edit subnet associations**.

40. Select the row with **Public Subnet**.

41. Choose **Save associations**.

Task 5: Creating a security group for the application server

42. In the left navigation pane, choose **Security groups**.

43. Choose **Create security group**.

44. On the **Create security group** page, configure the following options:

- **Security group name:** Enter **App-SG**.
- **Description:** Enter **Allow HTTP traffic**.
- **VPC:** Choose **Lab VPC**.

45. In the **Inbound rules** section, choose **Add rule**, and then configure the following options:

- **Type:** Choose **HTTP**.
- **Source:** Choose **Anywhere-IPv4**.
- **Description - optional:** Enter **Allow web access**.

46. Choose **Create security group**.

Task 6: Launching an application server in the public subnet

47. On the **AWS Management Console**, in the search box, enter and choose **EC2** to open the Amazon EC2 console.

48. Choose **Launch instance**.

49. On the **Launch an instance** page, configure the following options:

- For **Name**, enter **App Server**.
- In the **Application and OS Images (Amazon Machine Image)** section, configure the following options:
 1. For **Quick Start**, keep the default **Amazon Linux** option.
 2. For **Amazon Machine Image (AMI)**, keep the default **Amazon Linux 2023 AMI** option.
- In the **Instance type** section, keep the default **t2.micro** option.
- For **Key pair name - required**, choose **vockey**.
- In the **Network settings** section, choose **Edit**, and then configure the following options:

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

1. For **VPC - required**, choose **Lab VPC**.
 2. For **Subnet**, choose **Public Subnet**.
 3. For **Firewall (security groups)**, choose **Select existing security group**.
 4. For **Common security groups**, choose **App-SG**.
- In the **Configure storage** section, keep the default settings.
 - Expand the **Advanced details** panel, and for **IAM instance profile**, choose **Inventory-App-Role**.
 - In the **User data** box, copy and paste the following code: `#!/bin/bash`

Install Apache Web Server and PHP

```
dnf install -y httpd wget php-fpm php-mysql php-json php php-devel
```

```
dnf install -y mariadb105-server
```

Download Lab files

```
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACACAD-3-113230/06-lab-mod7-guided-VPC/s3/scripts/al2023-inventory-app.zip -O inventory-app.zip
```

```
unzip inventory-app.zip -d /var/www/html/
```

Download and install the AWS SDK for PHP

```
wget https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.zip
```

```
unzip aws.zip -d /var/www/html
```

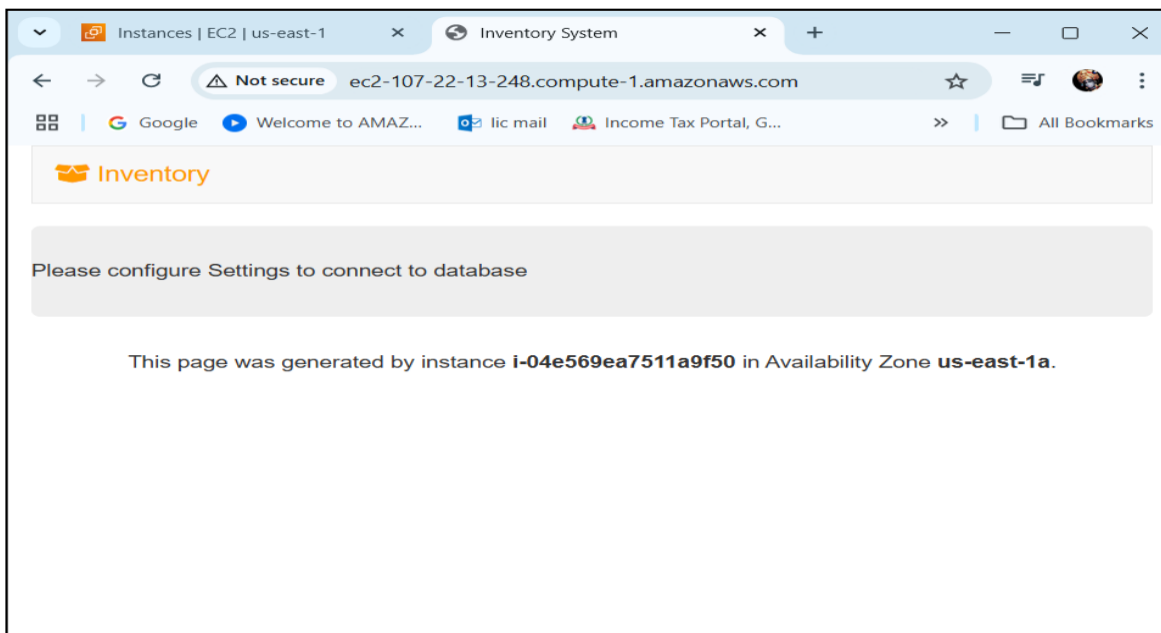
Turn on web server

```
systemctl enable httpd
```

```
systemctl start httpd
```

- In the **Summary** section, choose **Launch instance**.
50. Choose the link to the new instance that you created.
 51. Select **App Server**.
 52. From the **Details** tab, copy the **Public IPv4 DNS** value.
 53. Open a new web browser tab, and enter this public IPv4 DNS value.

OUTPUT:



VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS-6 : Serverless Architecture using Lambda

Task 1: Creating a Lambda function to load data

1. Choose **Create function**.
2. Configure the following settings:
 - For **Function name**, enter `Load-Inventory`.
 - For **Runtime**, choose **Python 3.8**.
 - Expand **Change default execution role**, and configure the following options:
 1. For **Execution role**, choose **Use an existing role**.
 2. For **Existing role**, choose **Lambda-Load-Inventory-Role**.
3. This role gives the Lambda function permission to access Amazon S3 and DynamoDB.
4. Choose **Create function**.
5. In the **Code source** section, in the **Environment** pane, choose **lambda_function.py**.
6. In the code editor for the **lambda_function.py** file, delete all the default code.
7. In the **Code source** editor, copy and paste the following code:

```
# Load-Inventory Lambda function
#
# This function is invoked by an object being created in an Amazon S3 bucket.
# The file is downloaded and each line is inserted into a DynamoDB table.
import json, urllib, boto3, csv
# Connect to S3 and DynamoDB
s3 = boto3.resource('s3')
dynamodb = boto3.resource('dynamodb')
# Connect to the DynamoDB tables
inventoryTable = dynamodb.Table('Inventory');
# This handler is run every time the Lambda function is invoked
def lambda_handler(event, context):
    # Show the incoming event in the debug log
    print("Event received by Lambda function: " + json.dumps(event, indent=2))
    # Get the bucket and object key from the Event
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'])
    localFilename = '/tmp/inventory.txt'
    # Download the file from S3 to the local filesystem
    try:
        s3.meta.client.download_file(bucket, key, localFilename)
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))
        raise e
    # Read the Inventory CSV file
    with open(localFilename) as csvfile:
        reader = csv.DictReader(csvfile, delimiter=',')
        # Read each row in the file
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

```
rowCount = 0
for row in reader:
    rowCount += 1
    # Show the row in the debug log
    print(row['store'], row['item'], row['count'])
    try:
        # Insert Store, Item and Count into the Inventory table
        inventoryTable.put_item(
            Item={
                'Store': row['store'],
                'Item': row['item'],
                'Count': int(row['count'])})
    except Exception as e:
        print(e)
        print("Unable to insert data into DynamoDB table".format(e))
# Finished!
return "%d counts inserted" % rowCount
```

8. At the top of the **Code source** section, choose **File** and then choose **Save**
 - Then **Deploy** your changes.

Task 2: Configuring an Amazon S3 event

9. On the AWS Management Console, in the search box, enter and choose **S3**.
10. Choose **Create bucket**.
11. For **Bucket name** enter **inventory-7** and replace *<number>* with a random number.
12. Choose **Create bucket**.
13. Choose the name of your **inventory-<number>** bucket.
14. Choose the **Properties** tab.
15. In the **Event notifications** section, choose **Create event notification**, and then configure these settings:
 - **Event name:** Enter **Load-Inventory**.
 - **Event types:** Choose **All object create events**.
 - **Destination:** Choose **Lambda function**.
 - **Lambda function:** Choose **Load-Inventory**.
16. Choose **Save changes**.

Task 3: Testing the loading process

17. Download the inventory files by opening (right-clicking) the context menu for these links: These files are the inventory files that you can use to test the system. They are comma-separated values (CSV) files. The following example shows the contents of the Berlin file:

```
store,item,count
Berlin,Echo Dot,12
Berlin,Echo (2nd Gen),19
Berlin,Echo Show,18
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

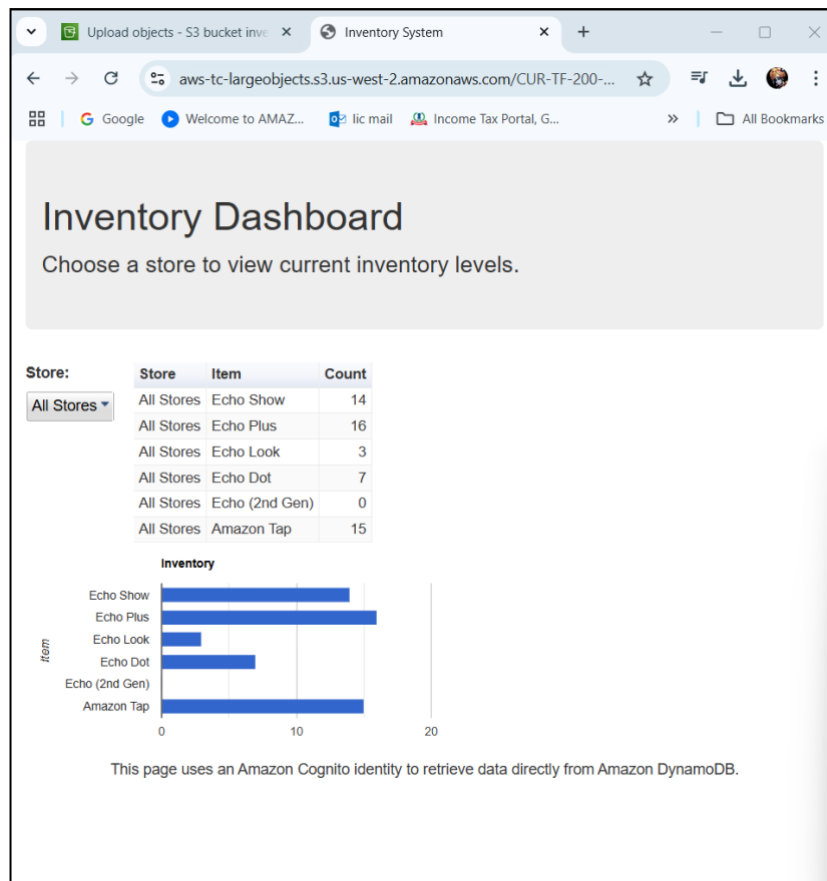
DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

Berlin,Echo Plus,0
Berlin,Echo Look,10
Berlin,Amazon Tap,15

18. Choose the **Objects** tab.
19. Choose **Upload**.
20. Choose **Add files**, and choose one of the inventory .csv files that you downloaded earlier in this task. You can choose any inventory file.
21. Choose **Upload**.
22. At the top of these instructions, choose **AWS Details**.
23. From the window, copy the **Dashboard** URL.
24. Copy and paste the URL into a new web browser tab.



25. On the AWS Management Console, in the search box, enter and choose **DynamoDB**.
26. In the left navigation pane, choose **Tables**.
27. Choose the **Inventory** table.
28. Choose **Explore table items**. The data from the inventory file is displayed. It shows the **Store**, **Item**, and **Count**.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)




(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

Items returned (6)					Actions ▾	Create item
				< 1 >		
<input type="checkbox"/>	Store (String) ▾	Item (String) ▾	Count ▾			
<input type="checkbox"/>	Calcutta	Amazon Tap	15			
<input type="checkbox"/>	Calcutta	Echo (2nd Gen)	0			
<input type="checkbox"/>	Calcutta	Echo Dot	7			
<input type="checkbox"/>	Calcutta	Echo Look	3			
<input type="checkbox"/>	Calcutta	Echo Plus	16			
<input type="checkbox"/>	Calcutta	Echo Show	14			

Task 4: Configuring notifications

29. On the AWS Management Console, in the search box, enter and choose **SNS**.
30. In the **Create topic** section, for **Topic name**, enter **NoStock**.
31. Choose **Next step**.
32. On the **Create topic** page, keep **Standard** selected.
33. Choose **Create topic**.
34. On the **NoStock** topic page, in the **Subscriptions** section, choose **Create subscription**.
35. On the **Create subscription** page, configure these settings:
 - **Protocol**: Choose **Email**.
 - **Endpoint**: Enter your email address.
36. Choose **Create subscription**.
37. To confirm your subscription, open the email message, and choose the **Confirm subscription** link.

Task 5: Creating a Lambda function to send notifications

38. On the AWS Management Console, in the search box, enter and choose **Lambda**.
39. Choose **Create function**.
40. Configure these settings:
 - For **Function name**, enter **Check-Stock**.
 - For **Runtime**, choose **Python 3.8**.
 - Expand **Change default execution role**, and configure the following options:
 1. For **Execution role**, choose **Use an existing role**.
 2. For **Existing role**, choose **Lambda-Check-Stock-Role**.
41. Choose **Create function**.
42. In the **Code source** section, in the **Environment** pane, choose **lambda_function.py**.
43. In the code editor for the **lambda_function.py** file, delete the code.
44. In the **Code source** editor, copy and paste the following code:

```
# Stock Check Lambda function
```

```
#
```

```
# This function is invoked when values are inserted into the Inventory DynamoDB table.
```

```
# Inventory counts are checked and if an item is out of stock, a notification is sent to an SNS Topic.
```


VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

```
import json, boto3
# This handler is run every time the Lambda function is invoked
def lambda_handler(event, context):
    # Show the incoming event in the debug log
    print("Event received by Lambda function: " + json.dumps(event, indent=2))
    # For each inventory item added, check if the count is zero
    for record in event['Records']:
        newImage = record['dynamodb'].get('NewImage', None)
        if newImage:
            count = int(record['dynamodb']['NewImage']['Count']['N'])
            if count == 0:
                store = record['dynamodb']['NewImage']['Store']['S']
                item = record['dynamodb']['NewImage']['Item']['S']
                # Construct message to be sent
                message = store + ' is out of stock of ' + item
                print(message)
                # Connect to SNS
                sns = boto3.client('sns')
                alertTopic = 'NoStock'
                snsTopicArn = [t['TopicArn'] for t in sns.list_topics()['Topics']
                               if t['TopicArn'].lower().endswith(':' + alertTopic.lower())][0]
                # Send message to SNS
                sns.publish(
                    TopicArn=snsTopicArn,
                    Message=message,
                    Subject='Inventory Alert!',
                    MessageStructure='raw'
                )
    # Finished!
    return 'Successfully processed {} records.'.format(len(event['Records']))
```

45. To save your changes, choose **File** and then choose **Save**
 - Then choose **Deploy**.
46. In the **Function overview** section, choose **Add trigger**, and configure these settings:
 - **Select a source:** Choose **DynamoDB**.
 - **DynamoDB table:** Choose **Inventory**.
47. Choose **Add**.

Task 6: Testing the system

48. On the AWS Management Console, in the search box, enter and choose **S3**.
49. Choose the name of your **inventory-<number>** bucket.
50. Choose **Upload**.
51. On the **Upload** page, choose **Add files**, and upload a different inventory file.
52. Return to the **Inventory System** dashboard browser tab, and refresh the page.
53. Try to upload multiple inventory files at the same time.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS- 7 : Deploy a Node.js application on a Docker Container

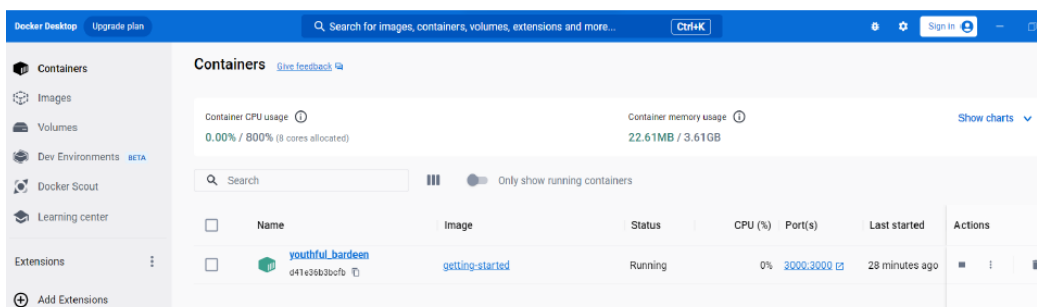
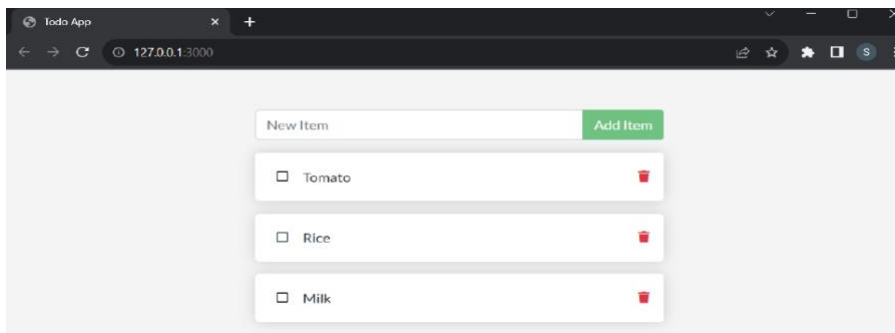
1. Install Docker for Windows.
2. Installing Virtual Studio Code.
3. Git Client Installation.
4. Restart the system.
5. Open Docker for windows.
6. Go to Power shell.
7. Clone the app from github.
8. cd enter; cd .\getting-started\
9. cd app
10. build docker
11. Run docker
12. View docker and check whether container is created and image is created.
13. Open new chrome tab.
14. Copy URL and paste it in the browser.

OUTPUT:

```
96650@DESKTOP-0934VJG MINGW64 ~/Desktop/docker/getting-started-app (main)
$ docker run -dp 127.0.0.1:3000:3000 getting-started
d41e36b3bcfb6a32b8b9da7d3b0a4c5b4c2d7ded0a6bc2a60a3ef4baa4a71ac7

96650@DESKTOP-0934VJG MINGW64 ~/Desktop/docker/getting-started-app (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
d41e36b3bcfb   getting-started "docker-entrypoint.s..." About a minute ago Up About a minute
127.0.0.1:3000->3000/tcp   youthful_bardeen

96650@DESKTOP-0934VJG MINGW64 ~/Desktop/docker/getting-started-app (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
d41e36b3bcfb   getting-started "docker-entrypoint.s..." About a minute ago Up About a minute 127.0.0.1:3000->3000/tcp youthful_bardeen
```



VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS- 8: Streaming dynamic content using Amazon CloudFront.

Task 1: Lab Preparation

In this lab, you will be using a sample video file to configure a dynamic stream. For your convenience, an Amazon Simple Storage Service (Amazon S3) bucket has already been created.

In the AWS Management Console, on the **Services** menu, choose **S3**.

An S3 bucket containing the string **awstrainingreinvent** should be present. Note the Region that the bucket is in, and open the bucket.

Open the **input** folder. It contains a video file named **AmazonS3Sample.mp4**.

Note: From the time you log in to the Amazon S3 console, it can take up to ten minutes for the file to appear in the S3 bucket. If you do not see it, select the circular arrow icon on the upper right of the screen to refresh the contents of the bucket.

Task 2: Create an Amazon CloudFront Distribution

In this task, you will create an Amazon CloudFront distribution that will be used to deliver the multiple bit-rate files generated by Amazon Elastic Transcoder to end-user devices.

On the **Services** menu, choose **CloudFront**.

Choose **Create a CloudFront distribution**.

Under **Origin Settings** section of the page, enter the follow information:

- Select the **Origin domain** field. A list of S3 buckets will appear. Choose the one that was created earlier that has **awstrainingreinvent** as part of the file name.
- Leave **Origin access** as **Public**.
- Under **Web Application Firewall (WAF)** select **Do not enable security protections**.

The warning message under **Custom SSL certificate - optional** can be safely ignored.

Scroll to the bottom of the page, then choose **Create Distribution**.

Task 3: Create an Amazon Elastic Transcoder Pipeline

Create a Pipeline

In this section, you will create a pipeline that will manage the jobs to transcode the input file.

In the AWS Management Console, on the **Services** menu, choose **Elastic Transcoder**.

In the navigation bar of the Amazon Elastic Transcoder console, select the same Region that the S3 bucket was created in.

On the Pipelines page, choose **Create a new Pipeline**.

For **Pipeline Name**, enter InputPipeline

For **Input Bucket**, select the **awstrainingreinvent** S3 bucket.

For **IAM Role**, under **Other roles**, select **AmazonElasticTranscoderRole**. This is a role that was pre-created in this lab's CloudFormation template that uses the managed policy AmazonElasticTranscoderRole. The Elastic Transcoder service will assume this role to access Amazon S3 and Amazon Simple Notification Service (Amazon SNS) resources in your lab account.

In the **Configuration for Amazon S3 Bucket for Transcoded Files and Playlists** section, enter the follow information:

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

- Under **Bucket**, select the **awstrainingreinvent** S3 bucket.
- Under **Storage Class**, select **Standard**.

In the **Configuration for Amazon S3 Bucket for Thumbnails** section, enter the following information:

- Under **Bucket**, select the **awstrainingreinvent** S3 bucket.
- Under **Storage Class**, select **ReducedRedundancy**.

Choose **Create Pipeline**.

Create a Job

In this section, you will create a job under the Amazon Elastic Transcoder pipeline that was just created. The job does the work of transcoding the input file into multiple bit-rates as selected.

On the Pipelines page, choose **Create New Job** to create a transcoding job. You create the job in the pipeline (queue) that you want to use to transcode the video file.

For **Pipeline**, select **InputPipeline**.

For **Output Key Prefix**, enter **output/**.

Amazon Elastic Transcoder will prepend this value to the names of all files that the job will create (including output files, thumbnails, and playlists).

For **Input Key**, select the input file labeled **input/AmazonS3Sample.mp4**.

Configure Output Details

The settings in this section will determine how many output files (bit-rates) are created. You will configure three output files for this demo having three separate bit-rates (2Mbps, 1.5Mbps and 1Mbps). Each output bit-rate will require you to create a separate output details section. This will also output a playlist file for each bit-rate, which lists all the segments that make up the stream.

For **Preset**, select **System preset: HLS 2M**

For **Segment Duration**, enter **10** (which is the HLS default).

For **Output Key**, enter the unique prefix **HLS20M** to name the segments created using this preset.

Click **+ Add Another Output** and repeat the steps above to generate segments for presets **HLS 1.5M** and **HLS 1M** and then provide the respective prefix names:

- **HLS15M**
- **HLS10M**

Caution: Do not create the job yet! Instead, complete the next few steps in this lab which will have you add a playlist to the job.

Configure a Playlist

The playlist will combine all the individual bit-rate playlists and provide a single URL for the devices to playback the stream. To configure a playlist, do the following:

Under **Playlists (Adaptive Streaming)**, choose **Add Playlist**, then configure:

- **Master Playlist Name** **primary**
- **Playlist Format:** **HLSv3**

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

Select all the three outputs, which were entered in the previous section, to include them in this playlist by selecting the + option.

Choose **Create New Job**.

The transcoding process should complete within a minute.

Task 4: Test Playback of the Dynamic (Multi Bit-Rate) Stream

In this module, you will test the playback of the dynamic stream generated in the previous section using an iOS or Android device. You can also use an Android 4.x device to test the below exercise.

Note: Certain browsers may not support this feature. Use the default web browser in the device to test.

Construct the Playback URL

The playback URL that plays through Amazon CloudFront is comprised of two components:

- Amazon CloudFront domain name
- Path of the playlist file in the S3 bucket (output generated by Elastic Transcoder):

http://<CloudFront domain name>/<playlist file path in Amazon S3 bucket>

Obtain an Amazon CloudFront Domain Name

To obtain an Amazon CloudFront domain name:

In the AWS Management Console, on the **Services** menu, choose **CloudFront**.

Select the **Amazon CloudFront** distribution that was previously created, and verify that the **Status** has changed from *InProgress* to *Enabled*.

Proceed to the next step only after the **Status** changes to *Enabled*.

Select the Distribution and under **Settings**. Copy the **Distribution domain name** and paste it into a text editor.

Obtain the Playlist File Path

To obtain the playlist file path:

On the **Services** menu, choose **S3**.

Select the **awstrainingreinvent** S3 bucket.

Open the **output** folder (which contains the output of the transcoding job) and select the **primary.m3u8** playlist file.

This is the file that you will play on your mobile device.

Next, you must create the URL to the file from CloudFront.

In a text editor, construct the URL by appending `/output/primary.m3u8` to the end of your CloudFront domain name.

The new URL should look similar to: `d1ckwesahkbyvu.cloudfront.net/output/primary.m3u8`

Type the URL into the default browser of an iOS or Android device. If you do not have a mobile device available, type the URL into a browser on your computer.

Be aware that standard data rates may apply when playing the video on a mobile device.

The stream should start playing on your device and dynamically request the relevant segments based on your bandwidth and CPU conditions.

You have learned how to use AWS services such as Amazon S3, Amazon Elastic Transcoder, and Amazon CloudFront together to deliver HLS media files to iOS or Android devices.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

You have successfully:

- Learned the basic concepts and terminology of the Amazon Elastic Transcoder and Amazon CloudFront services.
- Created your own Amazon Elastic Transcoder pipeline and Amazon CloudFront distribution.
- Used Amazon Elastic Transcoder to transcode a video file into different HLS formats and distributed it to remote devices using Amazon CloudFront.

OUTPUT:

Pipelines
Jobs
Presets

Create New Pipeline>Create New Job>EditPauseActivateRemove

Filter:

	Name	Input Bucket	Bucket for Transcoded Files	Bucket for Thumbnails	Status
<input type="checkbox"/>	InputPipeline	c92813a2068874f5284453t1w8 5452-awstrainingreinvent- oj6owelzwy5	c92813a2068874f5284453t1w8 5452-awstrainingreinvent- oj6owelzwy5	c92813a2068874f5284453t1w8 5452-awstrainingreinvent- oj6owelzwy5	Active

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS- 9: Running Containers on Amazon Elastic Kubernetes Service (Amazon EKS).

Step 1: Set Up Your AWS Account

1. **Sign Up:** If you don't have an AWS account, go to the [AWS website](#) and sign up.
2. **Access Management:** Set up IAM (Identity and Access Management) roles and policies for users who will manage the EKS cluster.

Step 2: Install Required Tools

1. **AWS CLI:** Install the AWS Command Line Interface (CLI) if you haven't already. Follow the [official instructions](#).
2. **kubect:** Install kubectl, the Kubernetes command-line tool. Follow the instructions [here](#).
3. **eksctl:** Install eksctl, a command-line tool for creating and managing EKS clusters. Follow the instructions [here](#).

Step 3: Configure AWS CLI

1. Run the following command and provide your AWS credentials:

```
bash
```

```
aws configure
```

Enter your access key, secret key, region (e.g., us-west-2), and preferred output format (e.g., json).

Step 4: Create an EKS Cluster

1. Use eksctl to create a cluster. Replace <CLUSTER_NAME> and <REGION> with your preferred name and AWS region.

```
bash
```

```
eksctl create cluster --name <CLUSTER_NAME> --region <REGION> --without-namespace
```

2. This command will create all necessary resources, including the VPC and EKS control plane.

Step 5: Configure kubectl

1. Once your cluster is created, eksctl automatically updates your kubeconfig file. To verify your connection to the EKS cluster, run:

```
bash
```

```
kubectl get svc
```

Step 6: Deploy Applications

1. You can now deploy applications to your cluster using Kubernetes manifests or Helm charts. Here's a basic example of deploying a simple application:

```
yaml
```

```
# example-deployment.yaml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: example-deployment
```

```
spec:
```

```
  replicas: 3
```

```
  selector:
```

```
    matchLabels:
```

```
      app: example
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: example
```

```
  spec:
```

```
    containers:
```


VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

```
- name: example-container
  image: nginx
  ports:
  - containerPort: 80
```

2. Apply the deployment:

```
bash
```

```
kubectl apply -f example-deployment.yaml
```

Step 7: Expose Your Application

1. You may want to expose the application using a service:

```
yaml
```

```
# example-service.yaml
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: example-service
```

```
spec:
```

```
  type: LoadBalancer
```

```
  ports:
```

```
  - port: 80
```

```
    targetPort: 80
```

```
  selector:
```

```
    app: example
```

2. Apply the service: `bash kubectl apply -f example-service.yaml`

Step 8: Monitor Your Cluster: Use AWS Management Console or AWS CLI to monitor your EKS cluster and the workloads running on it.

Additional Considerations

- **IAM Roles:** Ensure that your EKS cluster has the necessary IAM roles assigned for the services you will be using.
- **Networking:** Configure VPC and subnets according to your needs.
- **Cost Management:** Be aware of the costs associated with running EKS and resources in AWS.

These steps provide a high-level overview of setting up Kubernetes on AWS using EKS. You may want to refer to the official [EKS documentation](#) for more detailed information and updates.

OUTPUT:

The screenshot displays the AWS Management Console interface for an EKS cluster named 'demo-cluster'. The top navigation bar shows 'EKS > Clusters > demo-cluster'. On the right, there are buttons for 'Refresh' and 'Delete cluster'. The main content area is divided into two sections: 'Cluster info' and 'Details'.

Cluster info

Status	Kubernetes version	Support type	Provider
Active	1.28	Standard support until November 2024	EKS

Details

API server endpoint	OpenID Connect provider URL	Created
https://3ED55FDCCFAA0A46B6FBB8557604E17.gr7.us-east-1.eks.amazonaws.com	https://oidc.eks.us-east-1.amazonaws.com/id/3ED55FDCCFAA0A46B6FBB8557604E17	37 minutes ago

Cluster ARN is also listed at the bottom right.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS- 10: Implement a distributed application on Hadoop framework to count word frequency with MapReduce.

Implementing a distributed application on the Hadoop framework to count word frequency using MapReduce can be done in a few steps. Here's a simple guide to achieve that:

Step 1: Set Up the Hadoop Environment

1. **Install and configure Hadoop:** Ensure you have a running instance of Hadoop as described in the previous response. Make sure that HDFS and YARN are up and running.

Step 2: Write the MapReduce Code

You need two main components: the Mapper and the Reducer. The mapper will read the input data and emit key-value pairs, while the reducer will aggregate these pairs.

Here's an example in Java:

1. **Create a new Java file called WordCount.java:**

```
java
```

```
import org.apache.hadoop.conf.Configuration;
```

```
import org.apache.hadoop.fs.Path;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Job;
```

```
import org.apache.hadoop.mapreduce.Mapper;
```

```
import org.apache.hadoop.mapreduce.Reducer;
```

```
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
```

```
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

```
import java.io.IOException;
```

```
public class WordCount {
```

```
    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {
```

```
        private final static IntWritable one = new IntWritable(1);
```

```
        private Text word = new Text();
```

```
        public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
```

```
            String[] words = value.toString().split("\\s+");
```

```
            for (String w : words) {
```

```
                word.set(w);
```

```
                context.write(word, one);}}
```

```
    public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
```

```
        private IntWritable result = new IntWritable();
```

```
        public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
```

```
            InterruptedException {
```

```
            int sum = 0;
```

```
            for (IntWritable val : values) {
```

```
                sum += val.get();}
```

```
            result.set(sum);
```

```
            context.write(key, result);}}
```

```
    public static void main(String[] args) throws Exception {
```

```
        Configuration conf = new Configuration();
```

```
        Job job = Job.getInstance(conf, "word count");
```

```
        job.setJarByClass(WordCount.class);
```

```
        job.setMapperClass(TokenizerMapper.class);
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

```
job.setCombinerClass(IntSumReducer.class);
job.setReducerClass(IntSumReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
System.exit(job.waitForCompletion(true) ? 0 : 1);}
```

Step 3: Compile the Code

1. **Set Up the Hadoop Environment:** Make sure to include the Hadoop library in your classpath. You can use tools like Maven to manage dependencies or manually include the Hadoop jar files.
2. Compile the Java file:

```
javac -classpath `hadoopclasspath` -d /path/to/output/directory WordCount.java
```

3. Create a jar file:

```
jar -cvf wordcount.jar -C /path/to/output/directory/ .
```

Step 4: Prepare Input Data

1. **Create a text file** (for example, input.txt) with some text in it. Upload this file to HDFS:

```
hadoop fs -mkdir /input
```

```
hadoop fs -put input.txt /input/
```

Step 5: Run the WordCount Job

1. Execute the MapReduce job using the jar file you created:

```
bash
```

```
hadoop jar wordcount.jar WordCount /input/input.txt /output
```

Step 6: Check the Output

1. After the job completes, check the output directory on HDFS:

```
hadoop fs -ls /output
```

2. **Read the output file** (the filename will typically be part-r-00000):

```
hadoop fs -cat /output/part-r-00000
```

OUTPUT:

```
[training@localhost~]$hadoop fs -ls MRDir1
```

```
Found 3 items
```

```
-rw-r--r-- 1 training supergroup      0 2016-02-23 03:36 /user/training/MRDir1/_SUCCESS
```

```
drwxr-xr-x - training supergroup      0 2016-02-23 03:36 /user/training/MRDir1/_logs
```

```
-rw-r--r-- 1 training supergroup    20 2016-02-23 03:36 /user/training/MRDir1/part-r-00000
```

```
[training@localhost~]$hadoop fs -cat MRDir1/part-r-00000
```

```
BUS  7
```

```
CAR  4
```

```
TRAIN 6
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS- 11: Demonstrate Identity and access management for controlling account access.

Task 1: Configuring an IAM group with policies and an IAM user

In this first task in the lab, you will work as Sofía, the AWS account administrator.

As Sofía, you will create an IAM group and assign AWS managed policies to the group. You will then create a new IAM user, and add that user as a member of the group. Next, you will create an AWS Cloud9 environment on the existing EC2 instance where the development version of the café web application runs. Finally, you will share the development environment with the new user.

Note: The user you are logged in as displays in the upper-right area of the webpage. It should currently indicate that you are logged in as a *voclabs* user. In this lab, imagine that the *voclabs* user is Sofía.

In the console, open the IAM service page.

Create an **IAM group** named `AppDevelopers`, and attach the following IAM policies to it:

- **AmazonEC2ReadOnlyAccess**
- **AWSCloud9EnvironmentMember**

Create an **IAM user** and add the user to the *AppDevelopers* group.

- **User name:** `Nikhil`
- **Access type:** *AWS Management Console access*
- **Custom password:** `@ppD3veloper2020!`
- **Require password reset:** Clear this check box
- Add Nikhil to the *AppDevelopers* group
- **Note:** You don't need to add any tags
- In the **Success** screen, you can *optionally* choose **Download .csv** and save the file to your computer
- Choose **Close**

While still logged in as the *voclabs* user (Sofía), connect to the AWS Cloud9 IDE and set up the café web application.

Open the **AWS Cloud9** service page and under **DEV Cafe Server**, choose **Open IDE**.

The AWS Cloud9 IDE that run on an EC2 instance should now display.

In the Bash terminal window at the bottom of the screen, paste and run these three commands:

```
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod8-challenge/install-cafe-app.sh
chmod +x install-cafe-app.sh
./install-cafe-app.sh
```

Note: You do *not* need to wait for the script to finish running. Instead, go to the next step.

Share the AWS Cloud9 environment with the *Nikhil* user.

- In the top-right corner of the AWS Cloud9 IDE, choose **Share**.
- In the **Share this environment** panel, under **Invite Members**, enter `Nikhil` and choose **Invite**.
- Choose **OK**, choose **OK** again, and then choose **Done**.

Important: Close the AWS Cloud9 IDE browser tab, but leave open at least one other browser tab where you are logged into the AWS Management Console as the *voclabs* user (Sofía).

Task 2: Logging in as Nikhil and testing access

In this task, you will work as Nikhil to test the access permissions that Sofía configured.

Tip: We recommend opening an incognito or private browser tab to log in as *Nikhil*.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

For example, if you are using *Chrome*, choose **File > New Incognito Window**. If you are using *Firefox*, choose **File > New Private Window**.

Alternatively, use a different browser (for example, if you used Chrome to complete Task 1, use Firefox, Edge, or Safari for Task 2).

Note: Using an incognito session, a private session, or a different browser is convenient. You can remain logged in as both the *voclabs* user and as another user (such as *Nikhil*) simultaneously. You can thus switch between these two user-access levels without needing to log out or log in repeatedly. The authenticated session information in the different browser tabs will be isolated from one another.

As *Nikhil*, log in to the AWS Management Console.

In the browser tab where you are logged in as the *voclabs* user (Sofia), open the **IAM** console, choose **Users**, and then choose **Nikhil**.

- Choose the **Security credentials** tab, and in the **Sign-in credentials** section, copy the **Console sign-in link**.
- Paste the link into an incognito or private browser tab (or other browser as explained in the previous tip).
- In the **Sign in as IAM user** screen, enter Nikhil's credentials and choose **Sign in**.
 - **IAM user name:** Nikhil
 - **Password:** @ppD3veloper2020!

Open the **Amazon EC2** console and in a browser tab, load the café web application.

Verify that you are in the correct **Region** (for example, *N. Virginia*) and switch to it, if necessary.

You should be able to view all the details of the EC2 instances.

Locate and copy the **IPv4 Public IP address** of the **aws-cloud9-DEVCafeServer** instance.

In a new browser tab, load `http://<dev-public-ip-address>/cafe`, where `<dev-public-ip-address>` is the IP address that you copied.

The café website should display. Keep this browser tab open for later in the lab.

Test your Amazon EC2 access further by attempting to restart the web server.

Try to reboot the *aws-cloud9-DEVCafeServer* instance

Tip: To find the **Reboot instance** option, select the instance and look in the **Instance state** menu.

Answering questions about Nikhil's Amazon EC2 access

Answers will be checked when you choose the blue **Submit** button at the end of the lab.

Access the questions in this lab.

- Above these instructions, choose the **Details** menu, and then choose **Show**.
- At the bottom of the page, choose the **Access the multiple choice questions** link.

The questions should load in a new browser tab.

In the webpage that you just opened, answer the first two questions:

- **Question 1:** What happened when Nikhil tried to reboot the EC2 instance?
- **Question 2:** Which IAM policy allowed Nikhil to access the AWS Cloud9 environment?

Note: Leave the questions webpage open in your browser tab. You will return to it later in this lab.

Accessing the Development server as Nikhil

Return to the browser tab where you are logged into the AWS Management Console as *Nikhil*.

Tip: Remember that you can see which user you are using in the top-right area of the browser interface.

Browse to the **AWS Cloud9** console, and connect to the AWS Cloud9 IDE on the *DEVCafeServer* EC2 instance.

- From the **Services** menu, choose **AWS Cloud9**.
- On the left, expand the menu by choosing the (menu icon), choose **Environments**. In the **Environments** drop-down it should say **My environments**. Select the drop-down and choose **Shared with me**. The **DEVCafeServer** environment is now listed.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

- Choose **Open**.

Note: You have just connected to the guest OS of the *aws-cloud9-DEVCafeServer* EC2 instance. You observed this instance in the Amazon EC2 console a moment ago. You can use the IDE to browse and edit files on the web server. It also provides a Bash terminal that runs on the instance.

On the development instance of the café website, modify the main heading for the webpage.

Open the main webpage in the editor by going to the file browser, navigating to the

`DEV CafeServer/www/html/cafe` directory, and double-clicking `index.php`.

Modify **line 13** So that it reads:

```
<div class="center">Café; DEV Site</div>
```

To save the change, choose **File > Save** and in the browser, refresh the `http://dev-public-ip-address/cafe/` webpage.

Notice that, while acting as Nikhil, you changed the main heading of the webpage in the development environment.

Test the connectivity of the web application database.

- In the café website, choose **Menu**.
- What message displays?

Return to the browser tab with the multiple-choice questions for this lab, and answer the following question.

- **Question 3:** Which message was displayed on the **Menu** page of the café website's development instance?

Nikhil recalls that the database connection parameters are stored in the AWS Systems Manager Parameter Store.

As *Nikhil*, open the Systems Manager Parameter Store.

- In the console, open the **Systems Manager** service.
- From the menu on the left, choose **Application Management > Parameter Store**.
- What message displays?

Return to the browser tab with the multiple-choice questions for this lab, and answer the following question.

Question 4: Which message was displayed when Nikhil opened the Systems Manager Parameter Store page in the console?

Nikhil alerts Sofia about the issue on the development server that's preventing him from improving the café web application. Sofia is concerned. She asks Nikhil to check if the *production* version of the website is experiencing the same issue.

As *Nikhil*, verify that the production café web application is working correctly.

Open the **Amazon EC2** console and copy the **IPv4 Public IP address** of the **PRODCafeServer** instance.

- In a new browser window, load `http://prod-public-ip-address/cafe/menu.php`.
- Does the webpage display correctly, and can you place orders?

New business requirement: Configuring AWS account access for database administrators (Challenge #2)

Nikhil reports the results of his test to Sofia. She's glad to know that the production site is still functioning well. However, Sofia wants to correct the problem on the development site.

Sofia decides that she will ask Olivia to fix the issue. However, Sofia must first define the AWS account access rights for database administrators. She must then create an IAM user resource so that Olivia can log in to the account.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

Task 3: Configuring IAM for database administrator user access

In this task, you will work as *Sofia* to enable AWS access for Olivia.

Back in the browser where you are logged in as the *voclabs* user (Sofia), create an **IAM group** named **DBAdministrators**, with the following permissions:

- **AmazonRDSReadOnlyAccess**
- **AmazonSSMFullAccess**.

Create an IAM user that's named **Olivia** with access to the **AWS Management Console**.

- Set a custom password: **Db@dministrat0r2020!**
- Clear the requirement to reset the password

Add Olivia to the **DBAdministrators** group.

Task 4: Logging in as the database administrator and resolving the database connectivity issue

In this task, you will work as *Olivia* to resolve the database issue that Nikhil identified. You will also work as *Sofia* to help Olivia resolve some issues.

As *Olivia*, log in to the AWS Management Console.

Tip: Use the incognito session, private session, or other browser type that you used to log in as Nikhil. To do this:

- Choose **Nikhil @ <account-number>** in the top-right area of the console and choose **Sign Out**.
- Then, choose **Log back in**.

The **Sign in as IAM user** screen should display, with the *Account ID* pre-populated.

Note: If the login screen doesn't display the account ID, return to the browser tab where you are still logged in as the *voclabs* (Sofia) user. In the IAM console, choose **Users** and choose **Olivia**. Choose the **Security credentials** tab. In the **Sign-in credentials** section, copy the **Console sign-in link**.

Sign in with Olivia's credentials:

- **IAM user name:** **Olivia**
- **Password:** **Db@dministrat0r2020!**

Verify that the RDS database is running.

- Open the **Amazon RDS** service page and choose **Databases**.
- Verify that the **Status** of the database instance is *Available*.

Olivia observes that the database is running.

She recalls that the development environment connects to the database by using parameters that are stored in the Systems Manager Parameter Store. Olivia wonders if the *DEV CafeServer* EC2 instance has permissions to read the parameters out of the Parameter Store.

Open the **Amazon EC2** console and choose **Instances (running)**.

Return to the browser tab with the multiple-choice questions for this lab, and answer the following question.

Question 5: Why can't Olivia access the EC2 instance details? Olivia tells Sofia that she can't access the EC2 instances, and Sofia goes back to the console to troubleshoot this issue.

You will now work as *Sofia* to review and update Olivia's access to AWS resources.

Return to the browser tab where you are logged in as the *voclabs* user (Sofia).

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

Open the **DBAdministrators** group, and attach these policies:

- **AmazonEC2ReadOnlyAccess**
- **IAMReadOnlyAccess**

Note: Sofia realizes that Olivia needs some IAM permissions if she must access the details of the IAM role that's attached to the EC2 instance.

Still as the *voclabs* user (Sofia), check which services and features Olivia used.

- In the IAM console, open the **Olivia** user, and choose the **Access Advisor** tab
 - Notice that you can see which service areas that Olivia visited. Recent service activity usually appears within 4 hours (as stated in the Access Advisor details). You might not see any **Last accessed data** for Olivia yet.
- You can use this view to gain insight into how a user's permissions might be more open than they should be. This information enables you to more closely align access rights with the [principle of least privilege](#).

Sofia asks Olivia to check her Amazon EC2 access.

As *Olivia*, return to the browser tab where the Olivia user is logged in and refresh the instances page of the **Amazon EC2** console.

- Olivia should now be able to access both running EC2 instances.
- Select the **aws-cloud9-DEV CafeServer** instance.
- In the **Details** tab, find **IAM role** and choose **CafeRole**.
- In the **Permissions** tab, expand the **AmazonSSMManagedInstanceCore** policy to see the permission details in **JSON**.
- Review the policy permissions.

Return to the browser tab with the multiple-choice questions for this lab and answer the following question.

Question 6: Name two specific actions in the policy that allow the café web application on this instance to access the database credentials in the Parameter Store. As *Olivia*, update the **dbUser** value in the Systems Manager Parameter Store.

In the web application on the development café server, refresh the **Menu** page.

- If this webpage isn't already open, load `http://<dev-public-ip-address>/cafe/menu.php` in a browser (where `<dev-public-ip-address>` is the actual IPv4 public IP address of the **aws-cloud9-DEV CafeServer** instance).
- Does the full page of content display correctly now? Can you successfully submit an order?

Congratulations! You acted as a member of the *DBAdministrators* group, and you fixed the website. Nikhil thanks Olivia for resolving the issue, and Olivia also informs Sofia that the issue is resolved.

Task 5: Using the IAM Policy Simulator and creating a custom IAM policy with the visual editor

Return to the browser window where you are logged in as the *voclabs* user (Sofia), and load this URL in a new browser tab: <https://policysim.aws.amazon.com/>

The IAM Policy Simulator page should open.

Choose the **Olivia** user.

In the **IAM Policies** list, make sure that the **IAMReadOnlyAccess** policy is selected. However, *clear* the check boxes of the other policies.

In the **Policy Simulator** section, choose **Select service**. In the **Filter** search box, enter `Ident` and select **Identity and Access Management**.

Choose the **Select All** option (to the right of the **Select actions** menu), and then choose **Run Simulation**.

- In the **Action Settings and Results** panel, a list of actions should display.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

- The **Permission** column displays Olivia's permissions for each action. The *IAMReadOnlyAccess* policy denies Olivia the permissions to perform **Add** or **Create** actions. However, scroll to find the actions that she *can* take.
- The summary (at the top of the list) shows that Olivia is currently allowed to take *57 IAM actions*.

Sofia recalls the only reason that she granted Olivia the *IAMReadOnlyAccess* policy permissions. She wanted to grant Olivia the permissions to observe the details of certain policies. These policies are attached to the IAM role that's attached to the two café server instances. Sofia decides to author a new, more restrictive IAM policy for members of the *DBAdministrators* group.

In the next steps, you will work as Sofia to create this new policy.

Return to the browser tab where you are logged in as the **voclabs** user (Sofia).

In the **IAM** console, choose **Policies** and then choose **Create Policy**.

In the **Visual editor** tab, configure the following settings.

- Select **Choose a service**. Search for and choose **EC2**.
- In the **Actions** search box, search for **IAM** and select **DescribelamInstanceProfileAssociations**.

At the bottom of the screen, choose **Add additional permissions**.

- Select **Choose a service**. Search for and choose **IAM**.
 - In the **Actions** search box, search for **Get** and select the following actions –
 - **GetPolicyVersion**
 - **GetRole**
 - **GetRolePolicy**
 - **GetInstanceProfile**
 - Back in the search box, search for **List** and select the following actions –
 - **ListAttachedRolePolicies**
 - **ListInstanceProfiles**
 - **ListInstanceProfilesForRole**
 - **ListPolicies**
 - **ListRolePolicies**
 - **ListRoles**
- Expand the **Resources** section and for all three resource types (*instance-profile*, *policy*, and *role*) select **Any in this account**.
- Back at the top of the screen, choose the **JSON** tab

This view shows the JSON document that you just composed by using the visual editor.

- Verify that the policy document details match what is shown in the following example:

Unfortunately, in this lab environment, we can't grant you the permissions to create an IAM policy. You will get a permissions error if you choose **Review policy**, give the policy a name, and then choose **Create policy**.

However, a policy that exactly matches the example policy was created for you when you started this lab, and you have now gained experience with using the visual editor. You also experienced how it provides a way to create fine-grained policies without needing to author a JSON policy document from scratch.

Exit the **Create policy** wizard by choosing **Cancel**.

In the **Policies** search box, search for **LimitedIamPolicy**.

Observe that the policy details match the one you worked to build, as shown in the previous screen capture.

Note: The *Sid* elements in a policy are optional. Also, the order in which *Effect*, *Action*, and *Resource* appear in a policy document doesn't affect how the policy works.

Edit the **DBAdministrators** IAM group.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

- **Attach** the **LimitedIamPolicy** policy
- **Remove** the **IAMReadOnlyAccess** policy

Sofia asks Olivia to confirm that she can still access the details of the *CafeRole* IAM role, even with the more limited IAM access rights now granted to her.

Return to the browser tab where you are logged in as *Olivia*, and verify that you can still access the details of *CafeRole*.

- In the **Amazon EC2** console, select the **aws-cloud9-DEV** *CafeServer* instance.
- In the **Details** tab, notice that you can now see that the IAM role attached is named *CafeRole*.
- Still as Olivia, go to the IAM console and choose **Roles**.
- Search for and select the **CafeRole**.
- In the Permissions tab, expand the **AmazonSSMManagedInstanceCore** policy and verify that you can still see the JSON document details.

(Optional) Return to the **IAM Policy Simulator** browser tab where you are logged in as the *voclabs* user (Sofia). Run the simulation again (for what IAM actions Olivia can take).

Before the change to the *DBAdministrators* IAM group, Olivia was allowed to perform 57 actions that are related to the IAM service. However, after the change, her access to the IAM service is now much more limited.

The new policy grants fewer IAM permissions. However, Olivia still has the access that she needs to perform her job functions.

OUTPUT:

```
Visual editor  JSON  Import managed policy

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": [
8         "iam:ListPolicies",
9         "ec2:DescribeInstanceProfileAssociations",
10        "iam:ListRoles",
11        "iam:GetInstanceProfile"
12      ],
13      "Resource": "*"
14    },
15    {
16      "Sid": "VisualEditor1",
17      "Effect": "Allow",
18      "Action": [
19        "iam:GetRole",
20        "iam:ListInstanceProfilesForRole",
21        "iam:GetPolicyVersion",
22        "iam:ListAttachedRolePolicies",
23        "iam:ListRolePolicies",
24        "iam:ListInstanceProfiles",
25        "iam:GetRolePolicy"
26      ],
27      "Resource": [
28        "arn:aws:iam::*:policy/*",
29        "arn:aws:iam::*:instance-profile/*",
30        "arn:aws:iam::*:role/*"
31      ]
32    }
33  ]
34 }
```

IAM Policy Simulator																																																																
Policies		Policy Simulator																																																														
Selected user: Olivia		Identity And Ac... 141 Action(s) se... Select All Deselect All																																																														
AWS Organizations SCPs		Reset Contexts Clear Results Run Simulation																																																														
IAM Policies		Global Settings																																																														
Filter		Action Settings and Results [141 actions selected. 0 actions not simulated. 57 actions allowed. 84 actions denied.]																																																														
IAMReadOnlyAccess		<table><thead><tr><th>Service</th><th>Action</th><th>Resource Type</th><th>Simulation Resource</th><th>Permission</th></tr></thead><tbody><tr><td>Identity And Access Man...</td><td>DetachRolePolicy</td><td>role</td><td>*</td><td>denied</td></tr><tr><td>Identity And Access Man...</td><td>DetachUserPolicy</td><td>user</td><td>*</td><td>denied</td></tr><tr><td>Identity And Access Man...</td><td>EnableMFADevice</td><td>user</td><td>*</td><td>denied</td></tr><tr><td>Identity And Access Man...</td><td>GenerateCredentialsReport</td><td>not required</td><td>*</td><td>allowed</td></tr><tr><td>Identity And Access Man...</td><td>GenerateOrganizationsAcces...</td><td>access-report</td><td>*</td><td>denied</td></tr><tr><td>Identity And Access Man...</td><td>GenerateServiceLastAccess...</td><td>not required</td><td>*</td><td>allowed</td></tr><tr><td>Identity And Access Man...</td><td>GetAccessKeyLastUsed</td><td>user</td><td>*</td><td>allowed</td></tr><tr><td>Identity And Access Man...</td><td>GetAccountAuthorizationGet...</td><td>not required</td><td>*</td><td>allowed</td></tr><tr><td>Identity And Access Man...</td><td>GetAccountPasswordPolicy</td><td>not required</td><td>*</td><td>allowed</td></tr><tr><td>Identity And Access Man...</td><td>GetAccountSummary</td><td>not required</td><td>*</td><td>allowed</td></tr><tr><td>Identity And Access Man...</td><td>GetContextKeysForCustomP...</td><td>not required</td><td>*</td><td>allowed</td></tr></tbody></table>			Service	Action	Resource Type	Simulation Resource	Permission	Identity And Access Man...	DetachRolePolicy	role	*	denied	Identity And Access Man...	DetachUserPolicy	user	*	denied	Identity And Access Man...	EnableMFADevice	user	*	denied	Identity And Access Man...	GenerateCredentialsReport	not required	*	allowed	Identity And Access Man...	GenerateOrganizationsAcces...	access-report	*	denied	Identity And Access Man...	GenerateServiceLastAccess...	not required	*	allowed	Identity And Access Man...	GetAccessKeyLastUsed	user	*	allowed	Identity And Access Man...	GetAccountAuthorizationGet...	not required	*	allowed	Identity And Access Man...	GetAccountPasswordPolicy	not required	*	allowed	Identity And Access Man...	GetAccountSummary	not required	*	allowed	Identity And Access Man...	GetContextKeysForCustomP...	not required	*	allowed
Service	Action	Resource Type	Simulation Resource	Permission																																																												
Identity And Access Man...	DetachRolePolicy	role	*	denied																																																												
Identity And Access Man...	DetachUserPolicy	user	*	denied																																																												
Identity And Access Man...	EnableMFADevice	user	*	denied																																																												
Identity And Access Man...	GenerateCredentialsReport	not required	*	allowed																																																												
Identity And Access Man...	GenerateOrganizationsAcces...	access-report	*	denied																																																												
Identity And Access Man...	GenerateServiceLastAccess...	not required	*	allowed																																																												
Identity And Access Man...	GetAccessKeyLastUsed	user	*	allowed																																																												
Identity And Access Man...	GetAccountAuthorizationGet...	not required	*	allowed																																																												
Identity And Access Man...	GetAccountPasswordPolicy	not required	*	allowed																																																												
Identity And Access Man...	GetAccountSummary	not required	*	allowed																																																												
Identity And Access Man...	GetContextKeysForCustomP...	not required	*	allowed																																																												

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

LAB PROGRAMS- 12: Implement Elastic load balancing and auto scaling service.

Task 1: Create an AMI for Auto Scaling

1. In the **AWS Management Console**, in the search box next to **Services** , search for and select **EC2**.
2. In the left navigation pane, choose **Instances**. First, you will confirm that the instance is running.
3. Wait until the **Status Checks** for **Web Server 1** displays *2/2 checks passed*. If necessary, choose refresh to update the status. You will now create an AMI based upon this instance.
4. Select **Web Server 1**.
5. In the **Actions** menu, choose **Image and templates > Create image**, then configure:
 - **Image name:** WebServerAMI
 - **Image description:** Lab AMI for Web Server
6. Choose **Create image**. A confirmation banner displays the **AMI ID** for your new AMI.

Task 2: Create a Load Balancer

7. In the left navigation pane, choose **Target Groups**.
 - Choose **Create target group**
 - Choose a target type: **Instances**
 - **Target group name**, enter: LabGroup
 - Select **Lab VPC** from the **VPC** drop-down menu.
8. Choose **Next**. The **Register targets** screen appears.
9. Review the settings and choose **Create target group**
10. In the left navigation pane, choose **Load Balancers**.
11. At the top of the screen, choose **Create load balancer**.
12. Under **Application Load Balancer**, choose **Create**
13. Under **Load balancer name**, enter: LabELB
14. Scroll down to the **Network mapping** section, then:
 - For **VPC**, choose **Lab VPC**
 - Choose the **first** displayed Availability Zone, then select **Public Subnet 1** from the Subnet drop down menu that displays beneath it.
 - Choose the **second** displayed Availability Zone, then select **Public Subnet 2** from the Subnet drop down menu that displays beneath it.
 - You should now have two subnets selected: **Public Subnet 1** and **Public Subnet 2**.
15. In the **Security groups** section:
 - Choose the Security groups drop down menu and select **Web Security Group**
 - Below the drop down menu, choose the **X** next to the default security group to remove it. The **Web Security Group** security group should now be the only one that appears.
16. For the Listener HTTP:80 row, set the Default action to forward to **LabGroup**.
17. Scroll to the bottom and choose **Create load balancer**
The load balancer is successfully created. Choose **View load balancer**

Task 3: Create a Launch Template and an Auto Scaling Group

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

18. In the left navigation pane, choose **Launch Templates**.
19. Choose **Create launch template**
20. Configure the launch template settings and create it:
 - **Launch template name:** LabConfig
 - 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 *Web Server AMI*
 - **Instance type:** choose *t2.micro*
 - **Key pair name:** choose *vockey*
 - **Firewall (security groups):** choose *Select existing security group*
 - **Security groups:** choose *Web Security Group*
 - Scroll down to the **Advanced details** area and expand it.
 - Scroll down to the **Detailed CloudWatch monitoring** setting. Select *Enable*
 - Choose **Create launch template**
21. In the Success dialog, choose the **LabConfig** launch template.
22. From the **Actions** menu, choose *Create Auto Scaling group*
23. Configure the details in Step 1 (Choose launch template or configuration):
 - **Auto Scaling group name:** Lab Auto Scaling Group
 - **Launch template:** confirm that the *LabConfig* template you just created is selected.
 - Choose **Next**
24. Configure the details in Step 2 (Choose instance launch options):
 - **VPC:** choose *Lab VPC*
 - **Availability Zones and subnets:** Choose *Private Subnet 1* and then choose *Private Subnet 2*.
 - Choose **Next**
25. Configure the details in Step 3 (Configure advanced options):
 - Choose **Attach to an existing load balancer**
 - **Existing load balancer target groups:** select *LabGroup*.
 - In the **Additional settings** pane:
 - Select **Enable group metrics collection within CloudWatch**
 - Choose **Next**
26. Configure the details in Step 4 (Configure group size and scaling policies - optional):
 - Under **Group size**, configure:
 - **Desired capacity:** 2
 - **Minimum capacity:** 2
 - **Maximum capacity:** 6
 - Under **Scaling policies**, choose *Target tracking scaling policy* and configure:
 - **Scaling policy name:** LabScalingPolicy
 - **Metric type:** *Average CPU Utilization*
 - **Target value:** 60
 - Choose **Next**
27. Configure the details in Step 5 (Add notifications - optional):
 - Auto Scaling can send a notification when a scaling event takes place. You will use the default settings.
 - Choose **Next**
28. Configure the details in Step 6 (Add tags - optional):
 - Choose **Add tag** and Configure the following:
 - **Key:** Name
 - **Value:** Lab Instance

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

- Choose **Next**

29. Configure the details in Step 6 (Review):

- Review the details of your Auto Scaling group
- Choose **Create Auto Scaling group**

Task 4: Verify that Load Balancing is Working

30. In this task, you will verify that Load Balancing is working correctly.

31. In the left navigation pane, choose **Instances**.

32. You should see two new instances named **Lab Instance**. These were launched by Auto Scaling.

33. Next, you will confirm that the new instances have passed their Health Check.

34. In the left navigation pane, choose **Target Groups**.

- Select *LabGroup*
- Choose the **Targets** tab.

35. Two target instances named Lab Instance should be listed in the target group.

36. Wait until the **Status** of both instances transitions to *healthy*.

37. Choose Refresh in the upper-right to check for updates if necessary.

38. *Healthy* indicates that an instance has passed the Load Balancer's health check. This means that the Load Balancer will send traffic to the instance.

39. You can now access the Auto Scaling group via the Load Balancer.

40. In the left navigation pane, choose **Load Balancers**.

- Select the *LabELB* load balancer.

41. In the Details pane, copy the **DNS name** of the load balancer, making sure to omit "(A Record)".

42. It should look similar to: *LabELB-1998580470.us-west-2.elb.amazonaws.com*

43. Open a new web browser tab, paste the DNS Name you just copied, and press Enter.

44. The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.

Task 5: Test Auto Scaling

45. You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

46. Return to the AWS Management Console, but do not close the application tab — you will return to it soon.

47. In the search box next to **Services**, search for and select **CloudWatch**.

48. In the left navigation pane, choose **All alarms**.

49. Two alarms will be displayed. These were created automatically by the Auto Scaling group. They will automatically keep the average CPU load close to 60% while also staying within the limitation of having two to six instances.

- On the **Services** menu, choose **EC2**.
- In the left navigation pane, choose **Auto Scaling Groups**.
- Select **Lab Auto Scaling Group**.
- In the bottom half of the page, choose the **Automatic Scaling** tab.
- Select **LabScalingPolicy**.
- Choose **Actions** and **Edit**.
- Change the **Target Value** to 50.
- Choose **Update**
- On the **Services** menu, choose **CloudWatch**.
- In the left navigation pane, choose **All alarms** and verify you see two alarms.
- 50. Choose the **OK** alarm, which has *AlarmHigh* in its name.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____

51. If no alarm is showing **OK**, wait a minute then choose refresh in the top-right until the alarm status changes.
52. The **OK** indicates that the alarm has *not* been triggered. It is the alarm for **CPU Utilization > 60**, which will add instances when average CPU is high. The chart should show very low levels of CPU at the moment.
53. You will now tell the application to perform calculations that should raise the CPU level.
54. Return to the browser tab with the web application.
55. Choose **Load Test** beside the AWS logo.
56. This will cause the application to generate high loads. The browser page will automatically refresh so that all instances in the Auto Scaling group will generate load. Do not close this tab.
57. Return to browser tab with the **CloudWatch** console.
58. In less than 5 minutes, the **AlarmLow** alarm should change to **OK** and the **AlarmHigh** alarm status should change to *In alarm*.
59. You can choose Refresh in the top-right every 60 seconds to update the display.
60. You should see the **AlarmHigh** chart indicating an increasing CPU percentage. Once it crosses the 60% line for more than 3 minutes, it will trigger Auto Scaling to add additional instances.
61. Wait until the **AlarmHigh** alarm enters the *In alarm* state.
62. You can now view the additional instance(s) that were launched.
63. In the search box next to **Services**, search for and select **EC2**.
64. In the left navigation pane, choose **Instances**.
65. More than two instances labeled **Lab Instance** should now be running. The new instance(s) were created by Auto Scaling in response to the CloudWatch alarm.
- 66. Task 6: Terminate Web Server 1**
67. In this task, you will terminate *Web Server 1*. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.
68. Select **Web Server 1** (and ensure it is the only instance selected).
69. In the **Instance state** menu, choose **Instance State > Terminate Instance**.
70. Choose **Terminate**

OUTPUT:

The screenshot displays the AWS Management Console interface. On the left, the navigation pane shows various services, with 'LOAD BALANCING' and 'Load Balancers' highlighted. The main content area shows the configuration for a Load Balancer named 'my-lb'. The 'Basic Configuration' tab is active, displaying the following details:

Property	Value
Name	my-lb
ARN	
DNS name	us-east-1.elb.amazonaws.com (A Record)
Scheme	internet-facing
Type	application
Creation time	November 10, 2017
Hosted zone	
State	active
VPC	
IP address type	ipv4
AWS WAF Web ACL	

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DSCC LAB

Name: _____ Roll No. 1602-21-733-0 Page No. : _____