3(a)

Aim:

Description:

Static websites have fixed content with no backend processing. They can contain HTML Pages, image, style and all files that are needed to render a website. However, static websites do not use server side scripting or a database. If you want your static webpages to provide interactivity and run programming logic, you can use javascript that runs in the users web browser.

* choose StartLab to launch our lab
* wait until the lab status: ready
* choose AWS

Task1: creating a bucket in Amazon S3.

* In AWS management Console, on the Services menu, choose S3.
* Choose create bucket.
* For Bucket name, enter: website-<123> | Replace <123> with a random number.

- Verify the AWS Region is set to us-east-1
* In the object ownership section, select ACLs enabled, then verify Bucket owner preferred is selected.
* Clear Block all public access, then select the box that states I acknowledge that the current settings may result in this bucket and the objects with in becoming public.
* Choose create bucket.
* Choose the name of new bucket.
* Choose the properties tab.
* Scroll to the tags panel.
* Choose edit then Add tag and enter
  - Key: Department
  - Value: Marketing
* Choose Save changes to save the tag
* Stay in the properties console.
* Scroll to the static website hosting panel.
* Choose edit.
* Configure the settings.
* Save changes

* In the static website hosting panel, choose the link under Bucket website endpoint.

Task 2: Uploading content to bucket.

* Right-click each of these links and download the files to your computer.
  - Index.html
  - Script.js
  - Style.css
* Return to amazon S3 console and in the website <123> bucket you created earlier, choose the object tab.
* Choose upload
* Choose Add files
* Locate and select the three files that you downloaded
* If prompted, choose ☑ I acknowledge that existing objects with the same name will be overwritten.
* Choose upload.

your files are uploaded to the bucket.

- Choose close.

Task 3: Enable access to the objects

* Return to the browser tab that showed the 403 forbidden message
* Refresh the webpage.
* Select all three objects.
* In the Actions menu, choose make public via ACL.
* A list of three objects is displayed.
* choose Make public
* Return to web browser tab that has the 403 forbidden
* Refresh the webpage.

Task 4: updating the website

* On your computer, load the index.html file into a text editor.
* Find the text served from Amazon S3 console and upload the index.html file that you just edited.
* At top of these instructions, choose submit.
* choose End lab and select yes to confirm that you want to end the lab.

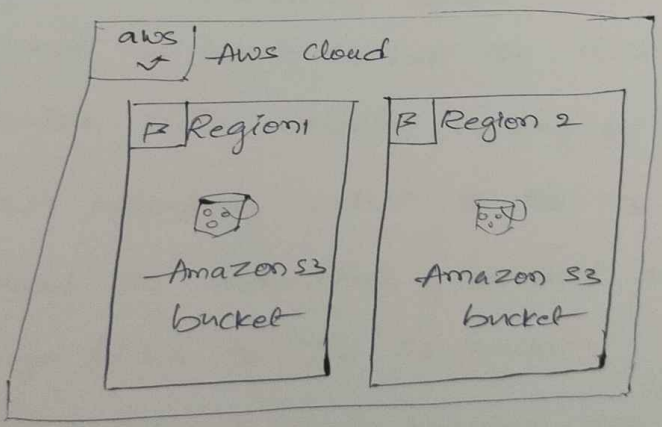
3(b)

Aim :-

Description :-

Frank and Martha are a husband and wife team who own and operate a small cafe business that sells desserts and coffee. Their daughter Sofia and their other employee Nikhil - who is secondary school student also work at the cafe, the cafe currently doesn't have a web marketing strategy. The cafe doesn't have a web presence yet and it doesn't currently use any cloud computing services.

Architecture :-



* At the top of these instructions, choose Start lab to launch the lab.
* Wait until the lab status ready.
* Choose Aws.

Task 1 : Extracting the files that we need for this lab.

* Download the zip file you need for this lab by opening this Amazon S3 link.
* On Computer, extract the files. Notice that your have an index.html file, and two folders that Contain Cascading Style sheets (CSS) and Image files.

Task 2 : Creating an S3 bucket to host the static website.

* Open the Amazon S3 console.
* Create a bucket to host the static website.
* Enable static website hosting on the bucket.

Task 3 : uploading content to the S3 bucket.

* upload the index.html file and the CSS and Images folder to the S3 bucket.
* In a separate web browser tab, open the endpoint link for static website.

* Access the questions of this lab.
* In the page we load, answer the first question.

question 1: when viewing the website after Task 2. do you see the page in the browser?

Task 4: creating a bucket policy to grant public read access.

* Create a bucket policy that grants read only permission to public anonymous users by using the bucket policy editor.
* Confirm that the website for the cafe is now Policy accessible

Task 5: Enabling versioning on the S3 bucket

* In the S3 Console, enable versioning on your S3 bucket.
* In your favourite text editor, open the index.html file
* Modify the file according to the given instructions
* Save the changes.
* upload the update file to the S3 bucket.
* Reload the web browser tab with your website and notice the changes.

* To see the latest version of index.html file, go to bucket and choose list versions.
* Return to browser tab with the multiple choice questions for the lab.

Task 6: Setting lifecycle policies.

* Configure two rules in website bucket's lifecycle configuration. Do not configure two transitions in a single rule.
  - In one rule, move previous versions of all source bucket objects to S3 standard-IA after 30 days.
  - In the other rule, delete previous versions of objects after 365 days.

Task 7: Enabling Cross Region Replication.

* In a different Region that your source bucket, create a second bucket and enable versioning on it.
* On your source S3 bucket, enable region replication.
  - Replace the entire source bucket.
  - use the cafe Role for the AWS Identity and Access Management (IAM) role.

Version : 2012-10-17

Resource :

Effect : Allow

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

Question 3: Do you see the objects from your source bucket in the destination bucket?

* Make a minor change to the index.html file and upload the new versions.

* Verify that the source bucket now has three versions of index.html file.

* Goto your source bucket and delete the latest version.

* Choose the submit.

* Choose the End Lab and select Yes to confirm that you want to end the lab.

4(a)

Aim:
Introducing Amazon EFS

Description:-
This lab introduces you to Amazon EFS by the AWS management Console. After completing this we should be able to log in to the AWS management console and create an Amazon EFS file system mount file system to your EC2 instance.

Procedure:
* Choose start lab to launch our lab.
* Wait until the lab status reads
* Choose AWS

Task1 : Creating a security group to access your EFS file system
* In the AWS Management Console on the Services menu choose EC2.
* Copy the Security group ID of the EFS client Security group to your text editor.
* Choose Create security group then configure
  - Security group name: EFS mount Target.

* Inbound rules section, choose Add rule then configure.
  - Type: NFS
  - Source: Custom
  - Choose: create security group

Task 2: Creating an EFS file system

* On the services menu, choose EFS
* Choose Create file system
* In the Create file system window, choose Customize.
* Choose Next
* Choose Create.

Task 3: Connecting to your EC2 instance via SSH

* Above these instructions that you are currently reading, choose the Details dropdown menu and then select show.
* Choose the download PRK button and save the labuser.ppk file.
* To use SSH to access the EC2 instance, you must use 'PUTTY'.

* Open putty.exe
* To keep the PUTTY section open for a longer period of time, configure the PUTTY TimeOut.
* Back on PUTTY, in the Connection list, expand SSH.
* Choose Auth and Expand Credentials
* When you are prompted with login as ec2-user.

Task4: Creating a new directory and mounting the EFS file System.

* In your SSH session, make a newdirectory by entering sudo choose EFS.
* In the Amazon EFS console, on the top right Corner on the page, choose Attach to open the Amazon EC2 mount instructions.
* copy the entire command in the using the NFS client section.

Task5: Examining the performance behaviour of your new EFS file system.

* Examine the write performance characteristics of your file system by entering fio command.

* The fio command will take 5-10 minutes to complete.
* In the AWS management console on the Services menu choose cloud watch.
* Choose metrics.
* In all metrices, choose EFS
* Choose File System metrics
* Select the checkbox for dataLink attribute TotalBytes.

Congratulations! you created an EFS file System.

Submitting your work

* At the top of these instructions, choose Submit.
* Choose End lab then select yes to confirm that you want to end the lab.

4(b)

Aim: Creating a Dynamic website for the cafe.

Description:

After the cafe launched the first version of their website, customers told the cafe staff how nice the website looks. They agreed that their business strategy and decisions should focus on delighting their customer and providing them with the best possible cafe experience.

Accessing the AWS management console:

1) At the top of these instructions choose start lab.

2) Wait until you see the status: ready

3) Choose AWS.

A business request for the cafe: preparing an EC2 instance of hosting a website.

Task1: Analyzing the Existing EC2 Instance

* In the search box next to services, search for and select EC2.
* Then choose Instances.
* Access the questions for this lab.
* In the page that you loaded, answer the first four questions.

Task2: Connecting to the IDE on the EC2 instance

* In the search box search for and select Cloud 9.

In the Environments page, notice the cafe website environment. It indicates that it is for type EC2 instance.

* Choose Open.

You are now connected to the AWS cloud 9 IDE that is running on the EC2 instance that you observed earlier.

Task3: Analyzing the Lamp stack environment and confirming that the website is accessible.

* Choose and observe the OS version.

In the AWS cloud 9 bash terminal run this command cat/proc/version.

* Start the web server and the database and also set them to start automatically after any future EC2 instance restart.

* Configure the EC2 instance so that you can use the AWS Cloud 9 editor to edit web server file.

*Creating a simple test webpage.

- In the file browser expand the cafe webroot directory and highlight the html directory
- Choose File > New File
- Chose File > Save and save the file.

In the second part of the lab, you will take on the role of sofia and install the Cafe application on the EC2 instance. You now know PHP is installed.

Task 5: Testing the web application

→ Test by placing an order

- In the browser tab where you have the page open Choose menu.
- Submit an order for at least one of the menu items.
- Return to the menu page and place another order, then go the order history.

Task 6: Creating an AMI and launching another EC2 Instance

→ Set a static internal hostname and create a new key pair on the EC2 Instance.

→ choose Actions > Images and Templates > Create Image.

Task 7:- Verifying the new cafe instance

→ Return to the EC2 console for the oreg on Region & verify that the new prod cafe server instance is running.

→ Copy the IPV4 public IP address and load it in a web browser.

→ Load the menu paging.

→ Place an order to verify that the website is working as intended.

Submitting your work:-

→ At the top of these instructions submit and when prompted choose Yes.

Lab Complete:-

→ At the top of this page, choose END lab and then choose Yes.

5(a)

Aim: To create an amazon RDS database

Descaiption:

Creating a database can be a complex process that requires either a database administrator or a systems administrator. In the cloud, you can simplify this process by using Amazon RDS. Launch a database using Amazon RDS. Configure a web application to connect to the database instance.

steps followed by creating an Amazon RDS database.

* choose start lab to launch the lab.
* wait until the lab status: ready.
* choose AWS.

Task1: Creating an Amazon RDS database

* choose RDS
* choose create database
* Select MySQL
* set the templates and availability and durability options:
- under the templates select Dev/Test.

- Under the Availability and durability section, select DB instance
* Under the settings section
  - DB instance identifier: inventory-db
  - Username: admin
  - password: lab-password
* Under the settings DB instance class
  - select Burstable classes
  - select db.t3.micro
* Under the Connectivity section,
  - Virtual private cloud (VPC): lab VPC
  - Existing vpc security groups.
* Expand the Additional configuration panel,
  - Initial database name: inventory
  - clear the Enable Enhanced Monitoring option
* choose Create database

Task 2: Configure web application communication with a database instance:

* choose EC2
* choose Instances
* select the App server instance.
* In the details tab, copy the public IPV4 address
* paste the ip address in new browser & enter.

* Choose Settings
* Return to the AWS management console
* Choose RDS
* Choose Database
* Choose Inventory.db
* Return to browser tab with the Inventory application,
  - End point: paste the endpoint you copied earlier
  - Database: Inventory
  - Username: admin
  - password: lab-password
  - choose save
* Add inventory, edit and delete Inventory information.
* Insert new records into the table. Ensure that the table has 5 or more Inventory records before submitting the work.
* Choose Submit.
* Choose End lab and select Yes to confirm that you want to end the lab

3

5(b)

Aim:- To Migrating a database to Amazon RDS.

Description:-

The cafe currently uses a single EC2 instance to host their webserver, database and application code. You will migrate data from a database on an amazon EC2 instance to amazon RDS Create an RDS database instance. Export data from MariaDB database by using MySQL dump Connect a SQL client to an RDS database.

Steps:-

* choose startlab
* wait until labstatus: ready
* choose Aws.

Task1: creating an RDS instance.

Create an RDS instance that compiles with these specifications.

- Engine type: MariaDB
- Templates: Dev/Test
- DB instance: Identifier: Cafe Database.

- Availability zone : Choose us-east-1a
- Database port : keep the default Top port of 3306.

Task 2 : Analyzing the existing cafe application deployment.

* Choose Running instance
* Test the cafe application.
* Connect the EC2 instance.
  - Choose Session Manager.
  - Start a session & connect to the Cafeserver.
  - Enter the Following commands:

```
bash
sudo su
su ec2-user
who am i
cd /home/ec2-user/
```

Task 3 : Working with the database on the EC2 instance :

* Observe details of the database that runs on EC2 instance. service mariadb status
  mysql -- version
* Return to browser tab with the AWS system manager.
* choose parameter store
* Connect to the database

mysql -u root -p

* observe the data in the existing database.

Show database
use cafe-db
show tables
select * from orders;

* Exit the SQL client
* capture Existing data in a file
* Confirm that mysqldump succeded

Task 4:- Working with the RDS database.

* Return to the RDS.
* Access the questions in this tab
* Now, answer the first four questions.
* Establish a network connection from terminal running.
* Run the show database

Task 5 :- Importing the data into the RDS database instance.

* Import the data that you exported in tasks to run RDS database instance.
* Confirm that the data was imported
  - Run this command
  - Enter the password for the RDS instance

- Exit the SQL client;

Task 6: Connecting the cafe application to the new database.

* Return to AWS Systems Manager
* choose Parameter store.
* Connect the cafe application to the RDS instance.
* Confirm that you web application.
* Choose Submit
* choose End lab and choose Yes to confirm you to end the lab.