**1.1 OVERVIEW:**

The Project consists of two main web applications i) Manager App ii) User app which runs on AWS EC2 instance. The Manager controls the worker instances. Manual scaling and autoscaling has been made available. The manager also has the privilege to monitor the details of workers instances. The working pool size is maintained maximum of 6 and minimum of 1. AWS services like RDS, S3, CloudWatch have been used.

**2.1 DESCRIPTION:**

Requirement is to develop a manger application instance with control over user application worker instances in EC2, where the manager can resize the worker pool on demand. A web application for storing images and transforming them by applying different filters will run in worker instance. The application provides the following functionality through a web interface.

1. Login page: This page provides authentication to access the webpage. It also includes support for password recovery.

2. User management: Only user with administrator rights will be able to create or delete a user account. All users will have the permission to change the password. The user accounts will have access to all features on the website.

3. Image upload: An authenticated user should be allowed to upload a new image by selecting it from their local file system.

4. Image transformation: When a new image is uploaded by an authenticated user, the image is automatically converted to image thumbnail. The image is also transformed using Image Magick libraries blur (radius=0, sigma=8), shade (gray=True, azimuth=286.0, elevation=45.0), spread(radius=8.0)

5. Browsing: Authenticated users will be able to browse through thumbnails of their previously uploaded images. Clicking on a photo's thumbnail produces the full resolution version of the photo as well as its 3 transformations.

The Manager-Application on the main instance will have the following controls.

1. The Manager will be able to see the list of workers in real time. Under each worker there should be two charts showing CPU Utilization and HTTP request for past 30min.
2. The main page has a chart which shows the number of workers in past 30 min.
3. On the main page there will be a link to load balanced user-app entry URL.
4. There are two buttons which is used to change the worker pool size.
5. An Auto-scaling policy runs in the main instance to manage worker pool based on certain criteria.
6. The manager application will be able to terminate the manager instance and all worker instances.
7. The manager application will be able to all the application data stored in S3 and database.

**3.1 PREREQUISITES:**

The following are the required software’s and libraries that are to be installed.

**3.1.1** **APPLICATIONS:**

* Flask
* Python

**3.1.2 PYTHON PACKAGES USED:**

* passlib
* mysql.connector
* wand
* werkzeug
* PIL
* flask
* OS
* urllib
* random
* string
* Flask-Mail
* jsonify

Additional to the libraries, we have created a DB with two table as shown in 6.1 DATABASE SCHEMA, and one admin user who have access to login to the webpage.

**4.1 STARTING EC2 INSTANCE:**

Step1: Login to the AWS IAM User account with the credentials provided in “credentials.txt” through the given link.

<https://aws.amazon.com/console/>

Step 2: Search and select EC2 instance in the search bar.

Step 3: Select the instance with instance type as t2.medium and choose the option Start instance from Instance State option on top right to start the instances. (Instance Name – Manager, Worker1)

The instance is connected to cloud9 environment. To access the instance using cloud9, navigate to the cloud9 page in the AWS Management Console. Under ‘Your Environment’ section you will find the Manager and Worker1 environment. Before accessing the IDE, the public IP address must be edited to the new one after starting the instance. The instance is launched with ami-080ff70d8f5b80ba5 containing all the required dependencies for the application to get executed. The EC2 instance can also be accessed by SSH-ing using the key-pair (ece1779.pem). To ssh into the instance, store the pem file in the .ssh directory. Then navigate to that directory and use the command “ssh -i ece1779.pem ubuntu@IP\_ADDRESS” – replace the IP\_ADDRESS with the public IP address of the instance. The application runs on port 5000 and is accessible from outside the instance using the public IP address. The public IP address changes every time the instance is started. The IP address can be seen under the field “Public IPv4 address” within instance summary. The IP is linked to a DNS name through which the user application can be accessed. Mention the webpage url

**5.1 SETTING CLOUDWATCH AGENT**:

CloudWatch agent is used to read the log for html request and create log group. To install and setup cloud watch following steps are used.

Step1: Download cloudwatch agent using the command

“wget <https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb>”

Step2: Install the agent using the command “sudo dpkg -i -E ./amazon-cloudwatch-agent.deb”

Step3: Setup cloudwatch wizard with the required specification using command

“sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard”

Step4: Start the cloudwatch agent using command.

“sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s”

When there is http request the log group gets updated with the http request, which is used to plot the Http count graph.

**6.1 AMAZON S3 and RDS:**

**7.1 RUNNING APPLICATION:**

Step 1: In the terminal, navigate to project folder using command “cd assignment1” from home directory

Step 2: Activate the virtual environment with command “source venv/bin/activate”

Step 3: Execute the run.sh file using command “./run.sh”

The user ‘admin’ is the only user with permission to create new users. The username and password of admin is given as below.

Username: admin

Password: admin

* 1. **APPLICATION ARCHITECTURE:**

Diagram

Description automatically generated

Fig.1 Overall Architecture

Diagram

Description automatically generated

Fig.2 User Application Architecture

**9.1 DATABASE SCHEMA:**

Below is the database model that represents the structure storing the data of the user. Two tables “user\_accounts” and “image\_path” are created in RDS “assignment1”.

The function sha256\_crypt from passlib.hash module is used to securely encrypt and store password into the database. The User data and path of the uploaded image is stored in DB in the table “image\_path” under the column “path”, instead of the image itself. The image can be found in S3. If two images with same name is uploaded, the second image will be considered as another image by the webpage and will be saved with a random string attached with the original name to differentiate between the two images.

Diagram

Description automatically generated

user (Primary key in user\_accounts and foreign key in image\_path)– Stores the username of the user logging into the webpage

password – Stores password for corresponding the user in encrypted form

email – Stores the mail address of the corresponding user

is\_admin – Boolean value which is set to “1” for admin user and set to “0” for normal user.

path – Stores the path of the images processed by the corresponding user.

**10.1 AUTOSCALING:**

Custom Autoscaling collects data from CloudWatch metrics and checks for the threshold breach every 5minutes. Based on the threshold value the instance gets added or deleted. The algorithm validates all the running instances and based on the average threshold decides to scale up or scale down the instance count in the ratio of 2.0 or 0.5 respectively.

**11.1** **MANAGER APPLICATION WEBPAGE FUNCTION:**

The Manager application is provided with authentication support for the manager. Only the manager user with valid credentials can login into the webpage.

Graphical user interface

Description automatically generated

**HOME PAGE:**

This is the main page which has options for manager to Create and Terminate instance, delete the data in RDS and S3, stop the main and user application. There is also a link to see the details of worker instances. This page displays a chart which shows the worker pool size. Manager will also be able to scale the instance manually.

A picture containing graphical user interface

Description automatically generated

**All INSTANCE DETAIL PAGE:**

This page displays all the worker instance running under the manager instance. This page also has buttons to see the complete details of the instance.

**INSTANCE PAGE:**

This page is specific for each running instance. This page displays the selected instance details and displays a chart for CPU Utilization and HTTP.

**11.1 USER APPLICATION WEBPAGE FUNCTION:**

**LOGIN PAGE:**

Graphical user interface

Description automatically generated

The login page has two buttons namely login and signup. Signup button is exclusive for admin when the admin wants to create a new user. Both requires username and password of the user.

**WELCOME PAGE:**

Text

Description automatically generated

The welcome page after logging in has option for the user to upload the required image.

**UPLOAD PAGE:**

Graphical user interface, website

Description automatically generated

The upload page provides option for user to upload the image and display the thumbnail of the images uploaded by the users. Authentication is required for this page to check the logged in user and to display the thumbnail.

Graphical user interface

Description automatically generated

A picture containing text, mammal

Description automatically generated

The above images are thumbnails of the images uploaded by the user admin.

When clicked on the thumbnail the four transformed images are displayed as shown below.

Text

Description automatically generated

**PASSWORD RECOVERY:**

The application provides option for the user to reset password through mail link.

Graphical user interface, application

Description automatically generated

Email is sent to the user mail ID with subject “Reset Password” and a password link in the body of the mail as shown below.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

The link takes the user to password recovery page through which the users can change their password, which gets stored in the database.

**NEW USER PAGE:**

Graphical user interface, application

Description automatically generated

The admin has the rights to create new user. The user created by the admin is stored in DB with corresponding values getting stored in required field.