Mobile Security Solution on Cloud

Team No - 28

(Mentor: Prateek Mehta)

Course Name: Cloud Computing

Professor: Dr. Vasudeva Varma

Akshay Dharphale 201405565

Sameer Jain 201405537

Shipra Banga 201201004

Umang Sharma 201201118

Introduction

This project aims at using android devices in place of CCTV cameras as they are cheaper alternatives. The device captures the picture, uploads in a predefined interval (session) to the cloud.

The user can view all his uploaded images through a web interface and filter those using appropriate filters. Components:

```
Android Application - Secure
Web Application - SecureUI
Rest Backend service - RestService
```

AWS Service used:

```
S3 - Stores the captured images.
RDS - Store the user and session tables.
EC2 - Virtual server that runs tomcat.
EBS - Provides the block storage for the EC2 instance.
CloudFront - Content delivery network to boost the image retrieval.
```

Deployment:

Secure.apk - Deployed to a compatible android phone.

SecureUI.war - Deployed to a HTTP Server (Apache Tomcat) running in Amazon EC2.

RestService.war - Deployed to a HTTP Server (Apache Tomcat) running in Amazon EC2.

1. Problem:

Designing a cloud framework which receive image data from mobile device periodically and save on cloud platform.

2. Proposed framework:

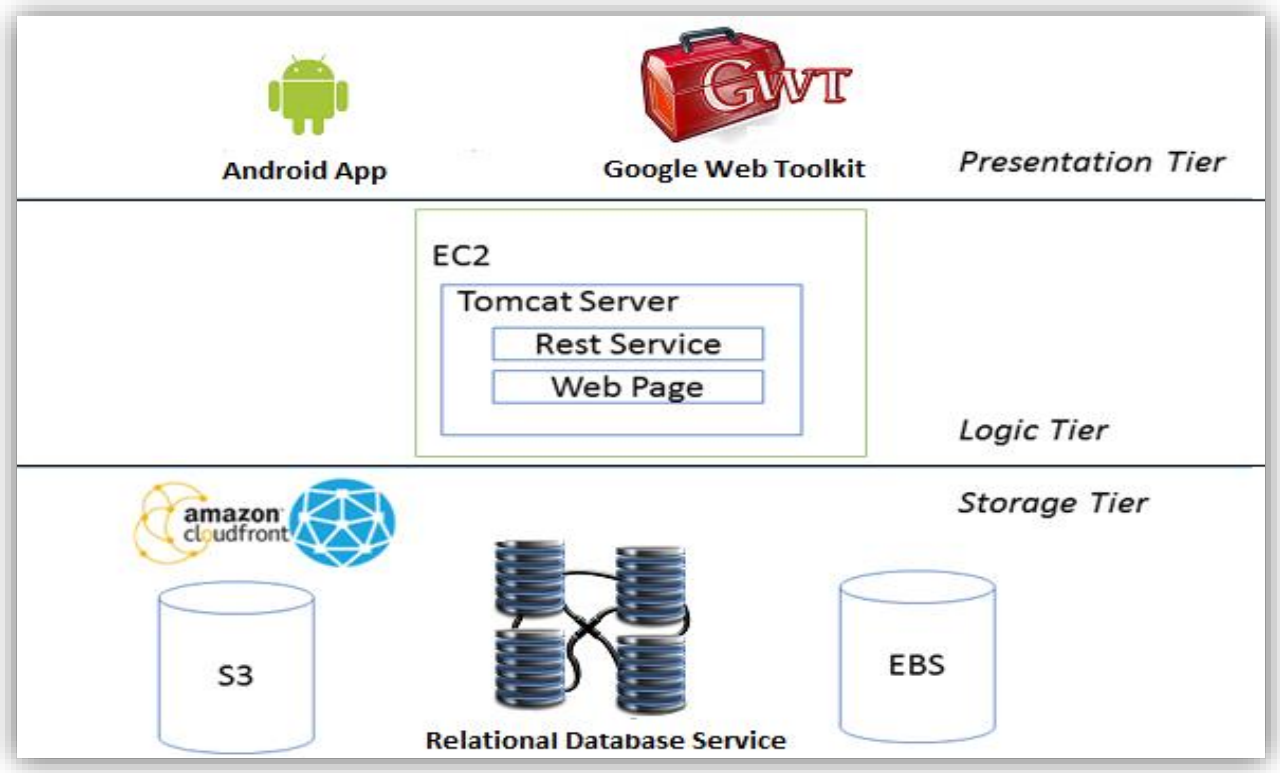

Figure 1 – Architecture Diagram

3. AWS Services Used:

3.1. Relational Database service (RDS):

RDS is a web service that makes it easy to set up, operate, and scale a relational database in the cloud.

We use RDS for storing following information.

3.1.1 Device Info

For every device (User), we maintain a table named User in the database which has the following fields:

ID, NAME, PASSWORD, CLICK DELAY IN SECONDS, UPLOAD INTERVAL IN SECONDS

CLICK DELAY IN SECONDS is an adjustable attribute which defines the time interval between two successive clicks by the camera.

UPLOAD INTERVAL is also an adjustable attribute which defines a session of time so that images captured in that session are to be uploaded altogether at the end of that session.

A sample tuple is given below.

ID	**NAME**	**PASSWORD**	**CLICK DELAY IN SECONDS**	**UPLOAD INTERVAL IN SECONDS**
Bakul	Bakul	12345	60	1800

3.1.2 Session Info

For every session we maintain a table named Session in the database which has the following fields:
ID, USERID, START TIME, END TIME and LOCATION.

A sample tuple is given below.

ID	**USER ID**	**START TIME**	**END TIME**	**LOCATION**
d07d7163-ebf6-404c-b3e7-643e3c003813	Bakul	2015-11-23 08:00:00	2015-11-23 08:29:59	West-IIITH

3.1.3 Configuration

Engine	MySQL 5.6.19a
Storage	5GB
Availability zone	US West (Oregon)
Configuration	1 VCPU, 1GB RAM

3.2 Simple Storage Service (S3):

S3 is a file storage web service, used to store the images captured from the device. All the images taken in a session are stored in a folder named with the session identifier. Every session for a device reside in a directory named with the device identifier.

3.2.1 Storage Hierarchy

The directory structure starts with the bucket name (‘mobilesecurity’), followed by the device name (‘bakul_nivas’), and followed by session identifier.

File name follows the pattern: IMAGE-ID_SESSION-ID_SNAPED-AT-TIMESTAMP.jpg

3.3 Elastic compute cloud (EC2) & Elastic Block Storage (EBS):

EC2 provides virtual servers on the cloud. An instance is created, attached to EBS, which provides memory for the server. We installed Apache Tomcat on the machine, which is HTTP server to host the back end service and web page.

3.3.1 Configuration

Public DNS	ec2-52-10-153-235.us-west-2.compute.amazonaws.com
Availability zone	US West -2b
EBS size	8GB
Amazon Machine Image	Amazon Linux AMI (HVM) 3.14 Kernel

3.4 Amazon Cloud Front

Amazon CloudFront is a content delivery network that speeds up distribution of static and dynamic web content, for example, .html, .css, .php, and image files, to end users. We use it only to power our images so that they will be served faster i.e. the distribution is created for the complete S3 bucket (‘mobilesecurity’) which host all the images.

4. Project Components

4.1 Android Application (Secure.apk)

Secure is the android application that captures and uploads photos periodically to cloud. This app. is developed completely in Java, using Android SDK.

The application automatically starts capturing pictures, and uploading them periodically (defined by uploadIntervalInSeconds in the device table).

If the uploadIntervalInSeconds is 30 minutes and the clickDelayInSeconds is 1 minute, it clicks pictures every minute, and starts uploading every 30 minute, by creating a session (/post/createSessionInfo) and then it pushes the image (/post/imageInfo). For this example, there are 31 (1+30) Rest calls are made for a complete transfer of images taken in a session.

4.2 Web Page (SecureUI.war)

SecureUI is the front end web application, used by security administrator to view images stored in cloud, in gallery form. It's completely built in Java using Google Web Toolkit (GWT), which is a compiler that converts Java code to efficient Javascript code. This service is deployed in tomcat running in the EC2 instance.

The web page allows one to:

- Login to a specific device. (/post/loginInfo)
- Login as a security admin user. (/post/loginInfo)
- Signup for a new device. (/post/signUpInfo)
- Search images by specifying filters such as start date of session, end date of session, device name and device location.

4.3 Rest Back End Service (RestService.war)

RestService is a REST based back end service, developed completely in Java. It uses Jersey library to parse and generate the JSON request and responses respectively. This service accepts request from both android application and web page.

5. Problems faced:

- **Connecting to EC2, RDS instances.** (Firewall issues) – We were unable to connect to the EC2 and RDS nodes as those ports were in blocked state. We solved by using an open network (idea data card).
- **Remote connection to EC2.** - We solved by using tools such as Putty (to remotely execute commands on a EC2 instance) and WinSCP (to transfer the service and web app. wars)