(2019S) COMP-5413-SC - Topics in Cloud Comp & Edge Co

PROJECT REPORT

LUGRAM

A PHOTO SHARING BLOG WEB APPLICATION USING AMAZON WEB SERVICES

Submitted by

HARIKA KUDARVALLI (0887033)

Submitted to

Dr. KEVIN PANG

15 JUNE 2019

INDEX

1. INTRODUCTION

This Photo Sharing System is a web-based application using Amazon Web Services. The primary goal is to replicate a current popular photo sharing application “Instagram”. The application allows users to share and upload photos online. They can be uploaded with hash tags and statuses as well. The users can login with username and password and authentication process will take place. The uploaded photos can be rated by other users as well with a time stamp. Other users can like and comment other user’s photos. For developing the application HTML and CSS have been used. Javascript is used to connect to Amazon Web Services. Multiple services have been used like Amazon S3, AWS API Gateway, AWS IAM, Amazon CloudWatch, Amazon DynamoDB, AWS Lambda, Amazon Cognito.

2. TECHNOLOGY

2.1 HyperText Markup Language

Hypertext Markup Language (HTML) is the major markup language used to display Web pages on the Internet. In other words, Web pages are composed of HTML, which is used to display text, images or other resources through a Web browser. HTML is plain text, meaning it is not compiled and may be read by humans. The file extension for an HTML file is .htm or .html. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

```
<!doctype html>
<html lang="en">
  <head>
    <title>LUGram &mdash; Share with the Community</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <link href="https://fonts.googleapis.com/css?family=Josefin+Sans:300, 400,700|Inconsolata:400,700" rel="stylesheet">

    <link rel="stylesheet" href="css/bootstrap.css">
    <link rel="stylesheet" href="css/animate.css">
    <link rel="stylesheet" href="css/owl.carousel.min.css">

    <link rel="stylesheet" href="fonts/ionicons/css/ionicons.min.css">
    <link rel="stylesheet" href="fonts/fontawesome/css/font-awesome.min.css">
    <link rel="stylesheet" href="fonts/flaticon/font/flaticon.css">

    <!-- Theme Style -->
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>


    <div class="wrap">

      <header role="banner">
        <div class="top-bar">
          <div class="container">
            <div class="row">
              <div class="col-9 social">
                <a href="#"><span class="fa fa-twitter"></span></a>
                <a href="#"><span class="fa fa-facebook"></span></a>
                <a href="#"><span class="fa fa-instagram"></span></a>
                <a href="#"><span class="fa fa-youtube-play"></span></a>
              </div>
```

2.2 Cascading Style Sheets

CSS (Cascading Style Sheets) is used to style and layout web pages — for example, to alter the font, color, size and spacing of your content, split it into multiple columns, or add animations and other decorative features. CSS mastery with the basics of how it works, including selectors and properties, writing CSS rules, applying CSS to HTML, how to specify length, color, and other units in CSS, cascade and inheritance, and debugging CSS.

```
html {
  font-family: sans-serif;
  line-height: 1.15;
  -webkit-text-size-adjust: 100%;
  -ms-text-size-adjust: 100%;
  -ms-overflow-style: scrollbar;
  -webkit-tap-highlight-color: transparent; }

@-ms-viewport {
  width: device-width; }

article, aside, figcaption, figure, footer, header, hgroup, main, nav, section {
  display: block; }

body {
  margin: 0;
  font-family: "Josefin Sans", -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,
  font-size: 1rem;
  font-weight: 400;
  line-height: 1.5;
  color: #212529;
  text-align: left;
  background-color: #fff; }

[tabindex="-1"]:focus {
  outline: 0 !important; }

hr {
  -webkit-box-sizing: content-box;
  box-sizing: content-box;
  height: 0;
  overflow: visible; }

h1, h2, h3, h4, h5, h6 {
  margin-top: 0;
  margin-bottom: 0.5rem; }
```

2.3 JavaScript

JavaScript (JS) is a lightweight interpreted or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Read more about JavaScript.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. Most websites use it, and major web browsers have a dedicated JavaScript engine to execute it.

2.4 Amazon Web Services

Amazon Web Services provides services from dozens of data centers spread across availability zones (AZs) in regions across the world. An AZ represents a location that typically contains multiple physical data centers, while a region is a collection of AZs in geographic proximity connected by low-latency network links. An AWS customer can spin up virtual machines (VMs) and replicate data in different AZs to achieve a highly reliable infrastructure that is resistant to failures of individual servers or an entire data center.

3. LITERATURE SURVEY

3.1 Performance Analysis of High-Performance Computing Applications on the Amazon Web Services Cloud by K. R. Jackson

Cloud computing has seen tremendous growth, particularly for commercial web applications. The on-demand, pay-as-you-go model creates a flexible and cost-effective means to access compute resources. For these reasons, the scientific computing community has shown increasing interest in exploring cloud computing. However, the underlying implementation and performance of clouds are very different from those at traditional supercomputing centers. It is therefore critical to evaluate the performance of HPC applications in today's cloud environments to understand the tradeoffs inherent in migrating to the cloud. This work represents the most comprehensive evaluation to date comparing conventional HPC platforms to Amazon EC2, using real applications representative of the workload at a typical supercomputing center. Overall results indicate that EC2 is six times slower than a typical mid-

range Linux cluster, and twenty times slower than a modern HPC system. The interconnect on the EC2 cloud platform severely limits performance and causes significant variability.

3.2 What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software by Tim O'Reilly

This paper was the first initiative to try to define Web 2.0 and understand its implications for the next generation of software, looking at both design patterns and business modes. Web 2.0 is the network as platform, spanning all connected devices; Web 2.0 applications are those that make the most of the intrinsic advantages of that platform: delivering software as a continually-updated service that gets better the more people use it, consuming and remixing data from multiple sources, including individual users, while providing their own data and services in a form that allows remixing by others, creating network effects through an architecture of participation, and going beyond the page metaphor of Web 1.0 to deliver rich user experiences.

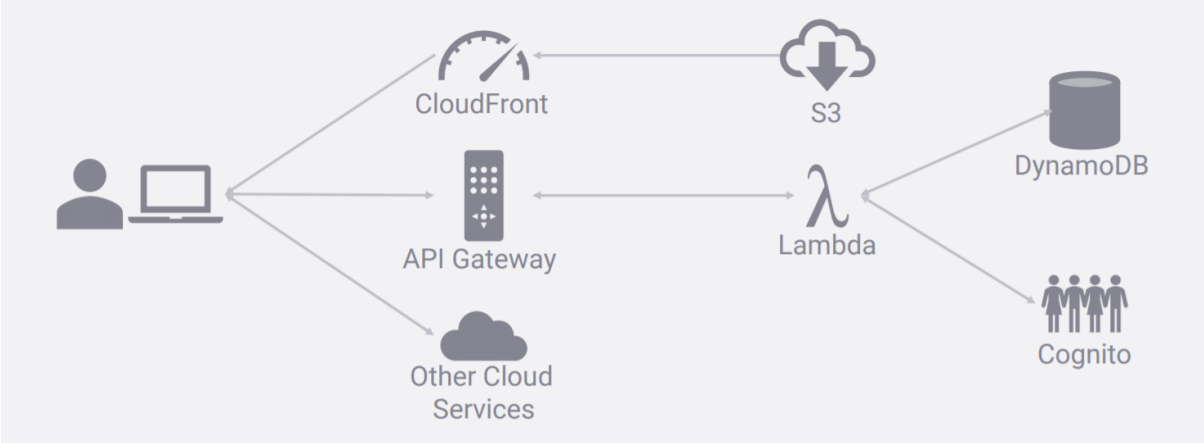
3.3 Digital Images, Photo-Sharing, and Our Shifting Notions of Everyday Aesthetics by Susan Murray

In this article, the author argues that the social use of digital photography, as represented on Flickr, signals a shift in the engagement with the everyday image, as it has become less about the special or rarefied moments of domestic living and more about an immediate, rather fleeting, display and collection of one's discovery and framing of the small and mundane. In this way, photography is no longer just the embalmer of time that André Bazin once spoke of, but rather a more alive, immediate, and often transitory practice/form. In addition, the everyday image becomes something that even the amateur can create and comment on with relative authority and ease, which works to break down the traditional bifurcation of amateur versus professional categories in image-making.

4. OBJECTIVES

1. Easy accessibility.
2. Developing a social network platform among university students.
3. User friendly UI design.
4. Upload blur-less images.
5. Using Amazon Web Services for computing, storage, security, authentication and accessibility roles.
6. Minimal programming required.

5. DESIGN



Homepage is designed in the following way:

1. Welcome message and photo of user.
2. Followed by series of uploaded photos.
3. Each photo with its ratings and comments.
4. Each photo will have timestamp.

Following are the activities associated with the application:

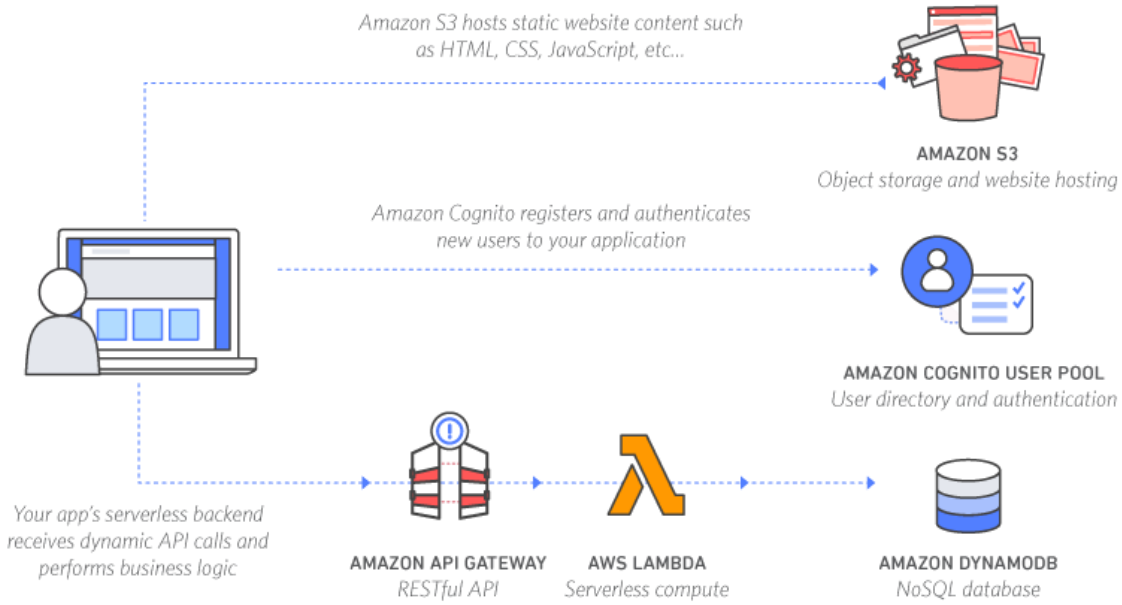
1. Registering as user.
2. Adding and viewing friends
3. User activity
4. Visitor Activity
5. Viewing user using tag names
6. Photo search
7. Recommendations
8. Comments on other photos
9. Multiple ways for uploading

PHASE 1

amazon web services

PHASE 2

6. ARCHITECTURE



6.1 Web Application

Developing a web application using HTML, CSS and JavaScript for photo sharing platform.

6.2 Amazon S3

Similar objects are grouped into buckets and are accessible globally. It is often used for backup functionality.

6.3 Amazon API Gateway

Developing and publishing APIs and using it to trigger Lambda functions.

6.4 AWS IAM

Multiple interaction among different AWS services in our application, we define the roles and policies for accessibility.

6.5 AWS Lambda

The web application will be deployed on AWS Lambda which runs the code without using servers.

6.6 Amazon DynamoDB

DynamoDB is a NOSQL database which has been used to save the entire database of the application.

6.7 Amazon Cognito

Multi factor authentication and password policy can be defined in Cognito for user authentication.

7. METHODOLOGY

7.1 Web Application

A web application (or "web app" for short) is any computer program that performs a specific function by using a web browser as its client. The application can be as simple as a message board or a contact form on a website or as complex as a word processor or a multi-player mobile gaming app that you download to your phone.

A web application relieves the developer of the responsibility of building a client for a specific type of computer or a specific operating system, so anyone can use the application along as they have internet access. Since the client runs in a web browser, the user could be using an IBM-compatible or a Mac. They can be running Windows XP or Windows Vista. They can even be using Internet Explorer or Firefox, though some applications require a specific Web browser.

Web applications commonly use a combination of server-side script (ASP, PHP, etc) and client-side script (HTML, Javascript, etc.) to develop the application. The client-side script deals with the presentation of the information while the server-side script deals with all the hard stuff like storing and retrieving the information.

1. LUGram is a photo sharing blog application.
2. The users can sign up and login with secured credentials.
3. Photos can be uploaded in different ways.
4. Each photo has a small description and tags.
5. Posts can be commented and liked by fellow users.
6. Each post has a timestamp.

7.2 Cloud Computing

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams.

A cloud service has three distinct characteristics that differentiate it from traditional web hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet, have accelerated interest in cloud computing.

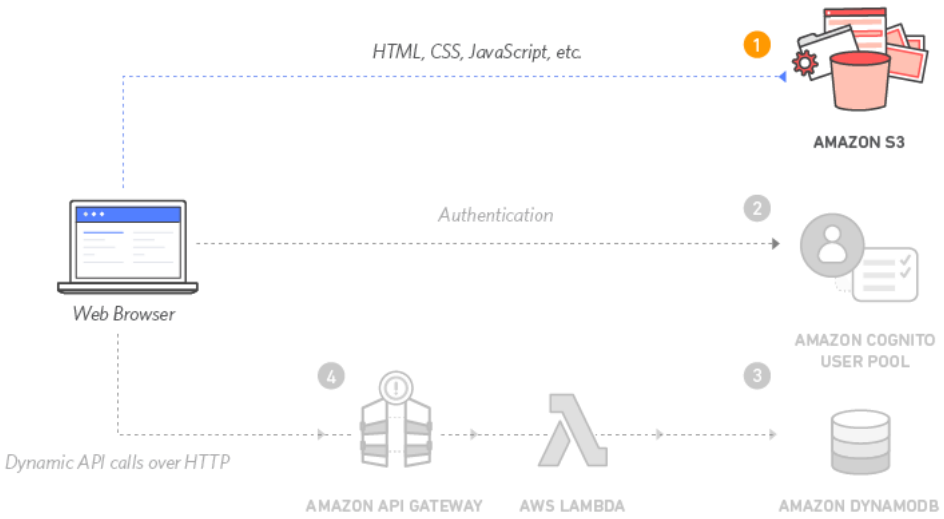
Cloud can be private or public. A public cloud sells services to anyone on the Internet. (Currently, Amazon Web Services is the largest public cloud provider.) A private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services.

7.3 Amazon S3

If your web application is static, it may be easier or cheaper to host it on Amazon Simple Storage Service (S3) instead of an EC2 instance. In S3, you can store sets of files in web-accessible folders, called buckets. S3 allows you to designate any bucket as a website. You define an index document (in other words, a start page) and a policy stating who can access the site. You can then interact with your pages from the bucket as if they were hosted on a traditional web server.

Not all web applications are appropriate for S3. If your app uses a server-side scripting language such as PHP, JSP, or ASP.NET, then you should host the app on your own EC2 instance. However, client-side scripting languages such as JavaScript are appropriate for S3. Applications built with the out-of-the-box Esri Web APIs and viewers should work on S3.

A benefit of hosting your page in S3 is the potential to use the Amazon CloudFront delivery service. This is an Amazon web service that hosts your content on various servers around the world, optimizing your file delivery speed among geographically dispersed users.



Basic steps involve:

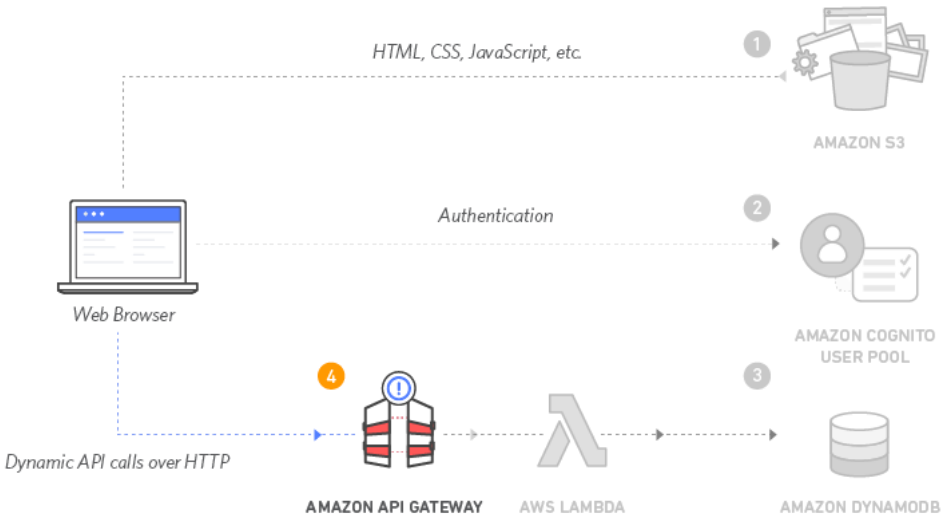
1. Select a region.
2. Create an S3 bucket.
3. Upload content.
4. Add a Bucket Policy to allow public access.

5. Enable website hosting.
6. Validating implementation.

7.4 Amazon API Gateway

An API gateway takes all API calls from clients, then routes them to the appropriate microservice with request routing, composition, and protocol translation. Typically it handles a request by invoking multiple microservices and aggregating the results, to determine the best path. It can translate between web protocols and web-unfriendly protocols that are used internally.

An ecommerce site might use an API gateway to provide mobile clients with an endpoint for retrieving all product details with a single request. It invokes various services, like product info and reviews, and combines the results.



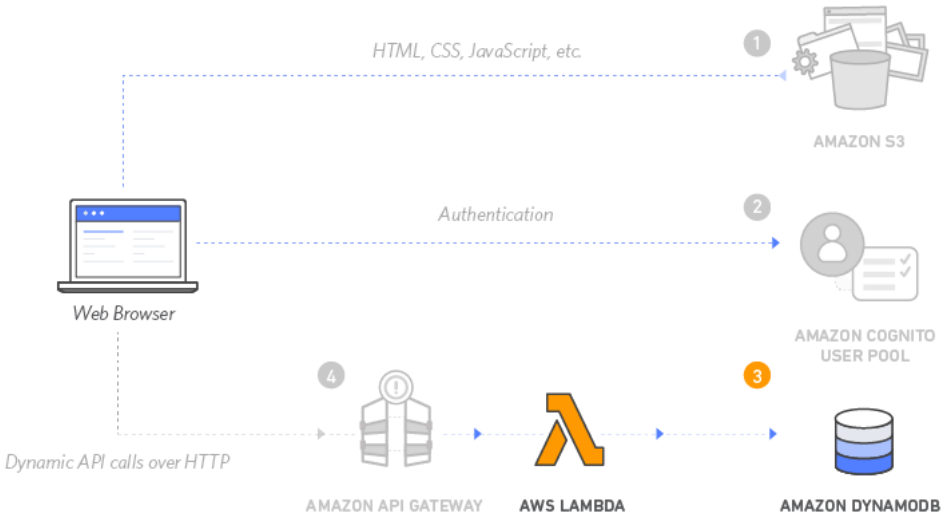
Basic steps involve:

1. Create a new REST API.
2. Create a Cognito User Pools Authorizer.
3. Create a new resource and method.
4. Deploy API.
5. Update website configuration.
6. Validating implementation.

7.5 AWS Lambda

A decade ago, cloud servers abstracted away physical servers. And now, “Serverless” is abstracting away cloud servers. Another advantage of going serverless is that you no longer need to keep a server running all the time. The “server” suddenly appears when you need it, then disappears when you’re done with it. Now you can think in terms of functions instead of servers, and all your business logic can now live within these functions.

In the case of AWS Lambda Functions, this is called a trigger. Lambda Functions can be triggered in different ways: an HTTP request, a new document upload to S3, a scheduled Job, an AWS Kinesis data stream, or a notification from AWS Simple Notification Service (SNS).



Basic steps involve:

1. Create an IAM role for Lambda Function.
2. Attach Policy to IAM role.
3. Create Lambda Function for handling requests.
4. Testing implementation.

7.6 Amazon DynamoDB

DynamoDB is a fully-managed NoSQL database that stores the data in key-value pairs. There are no schemas so every record can have a different structure. The only restriction is that the field(s) defined as the partition key must be present in all the records.

Based on this partition key, DynamoDB store data in different drives. An efficient distribution will make accessing the data as fast as possible, so it's important to choose a good partition key.

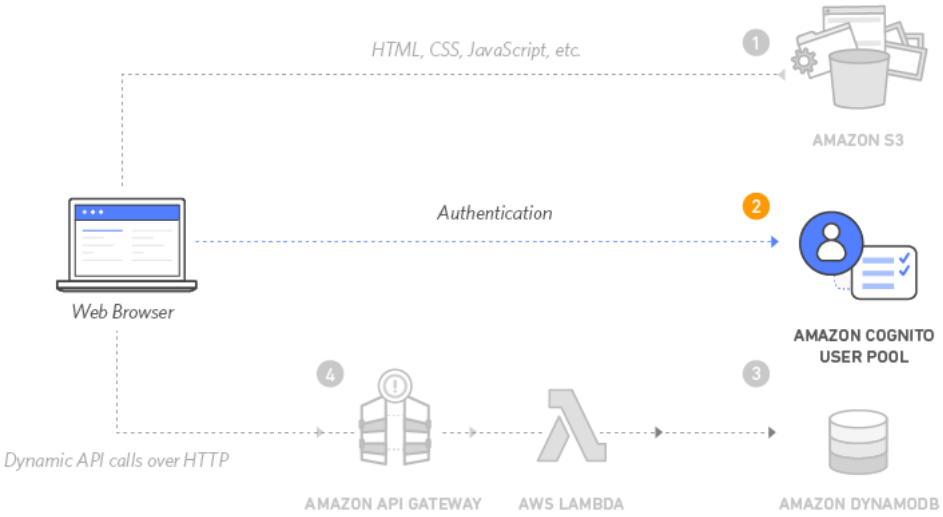
This way, the partition key can become the primary key, but you can also use a combination of a partition key and a sort key as a primary key. For example, if you have multiple records with the same course ID (the partition key), you can add a timestamp as a sort key to form a unique combination. In addition, you can also have secondary indexes for any other field (or combination of fields) to make queries more efficient.

7.7 Amazon Cognito

When users visit the website they will first register a new user account. However, you can configure Amazon Cognito to require additional attributes in your own applications.

After users submit their registration, Amazon Cognito will send a confirmation email with a verification code to the address they provided. To confirm their account, users will return to your site and enter their email address and the verification code they received. You can also confirm user accounts using the Amazon Cognito console if want to use fake email addresses for testing.

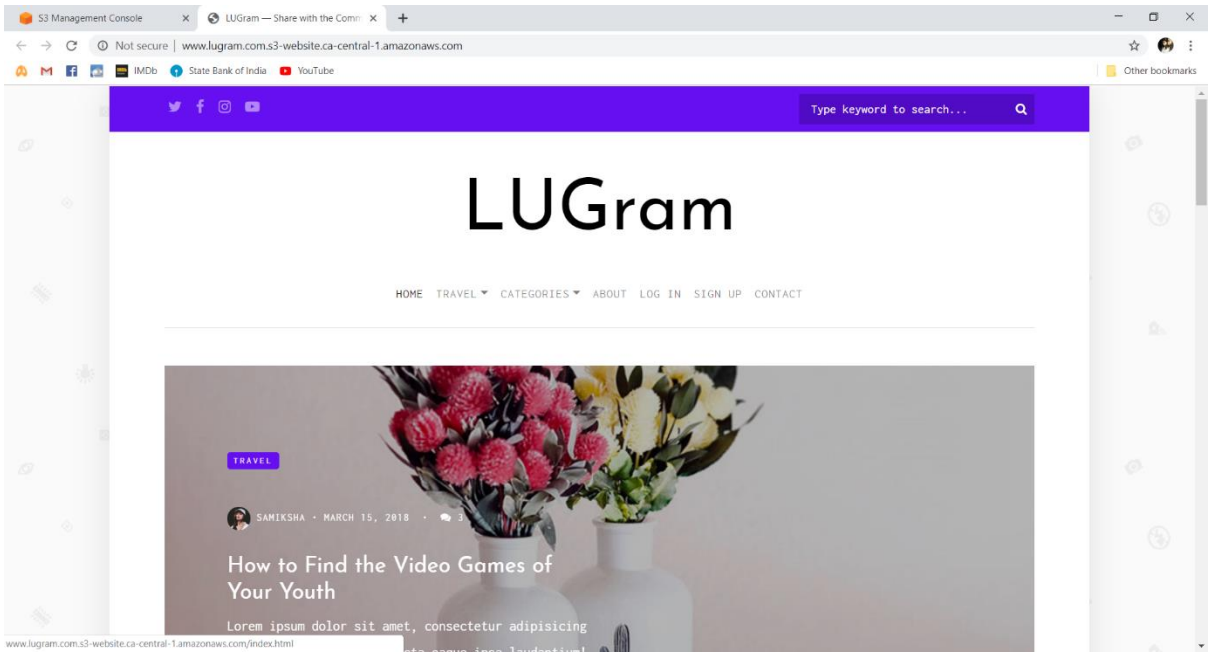
After users have a confirmed account (either using the email verification process or a manual confirmation through the console), they will be able to sign in. When users sign in, they enter their username (or email) and password. A JavaScript function then communicates with Amazon Cognito, authenticates using the Secure Remote Password protocol (SRP), and receives back a set of JSON Web Tokens (JWT). The JWTs contain claims about the identity of the user and will be used in the next module to authenticate against the RESTful API you build with Amazon API Gateway.

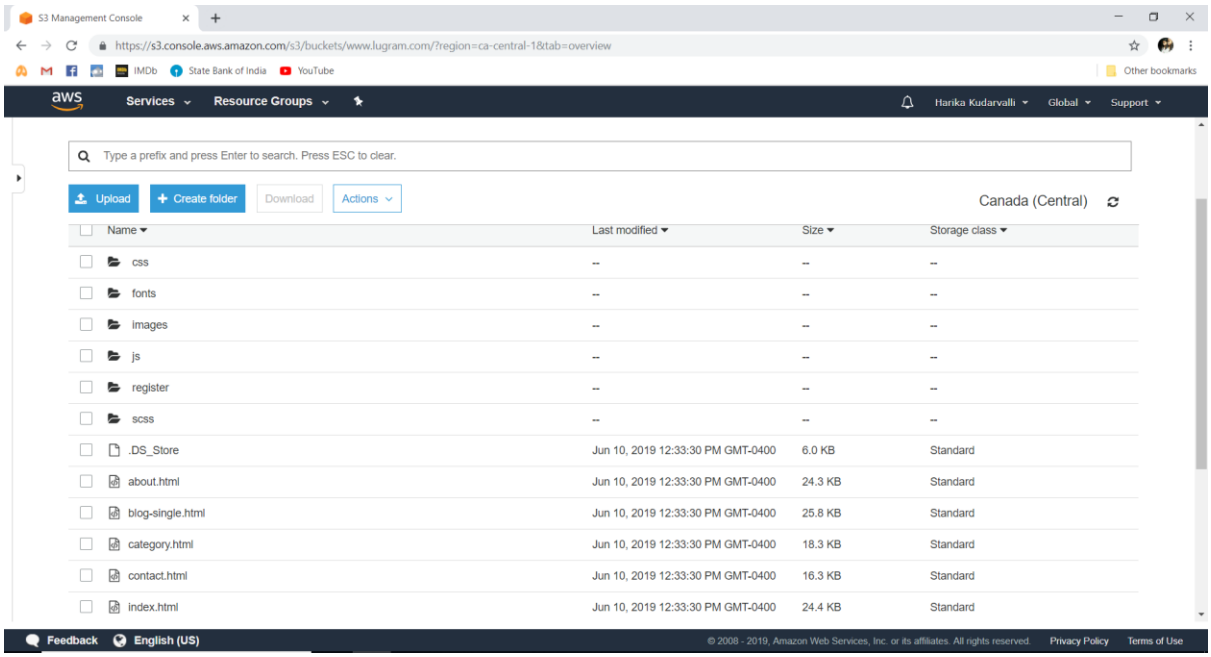


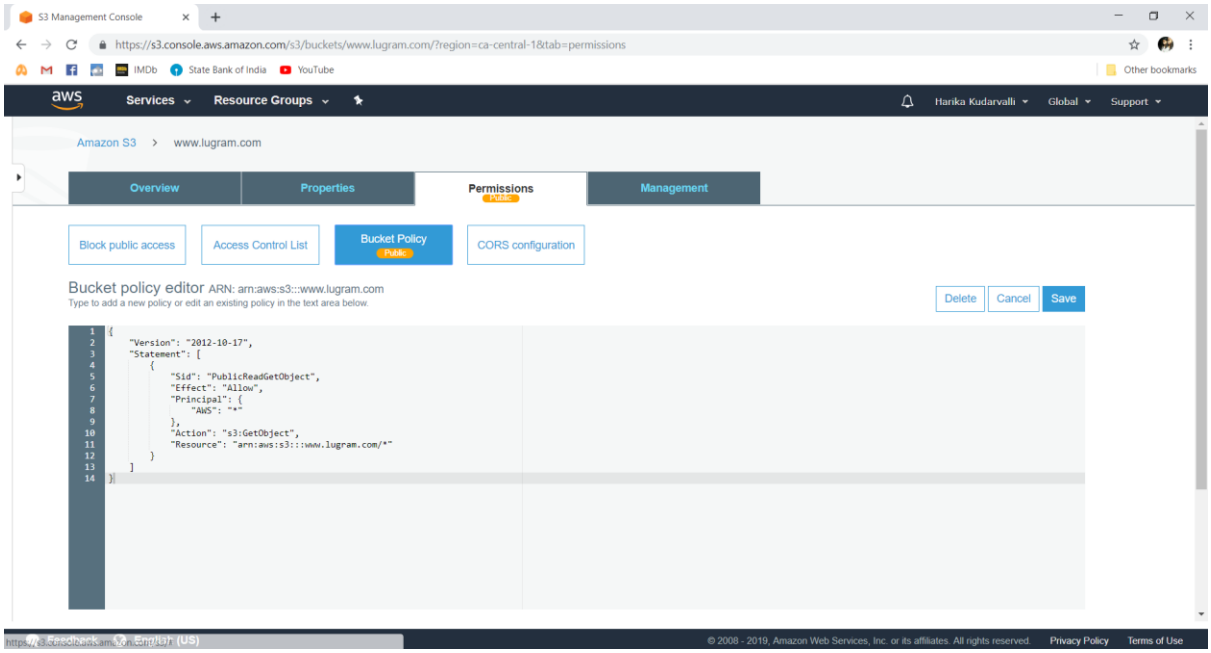
Basic steps involve:

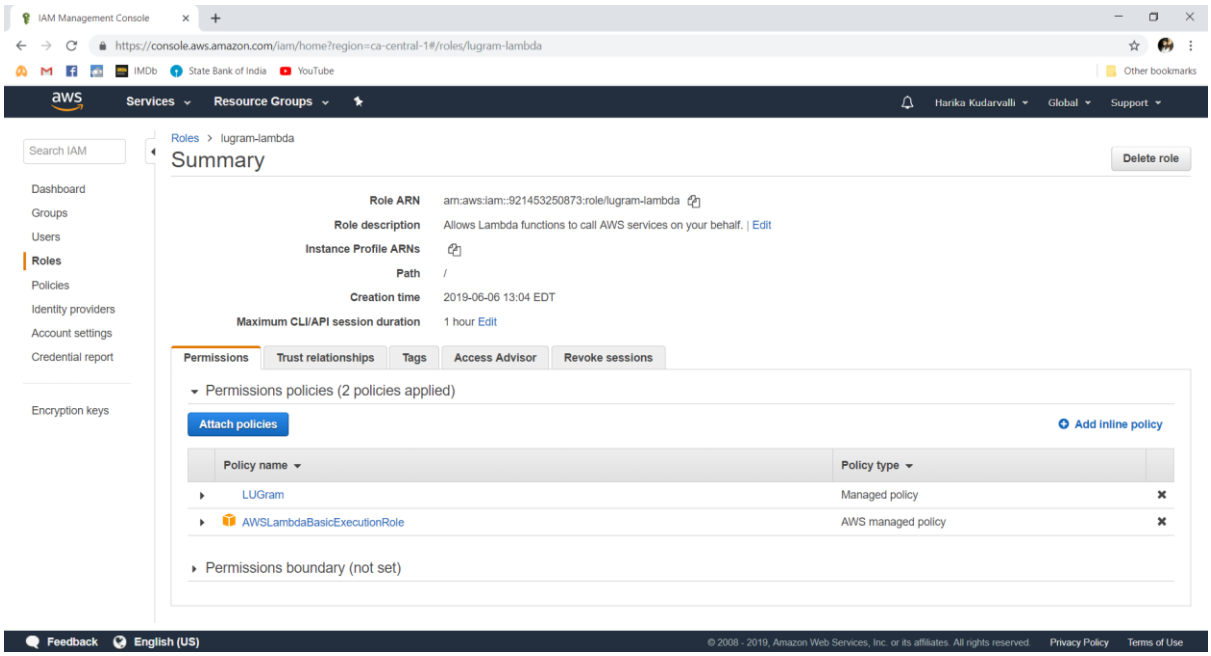
1. Create an Amazon Cognito User Pool.
2. Change settings accordingly.
3. Update config.js file in S3 bucket.
4. Testing implementation.

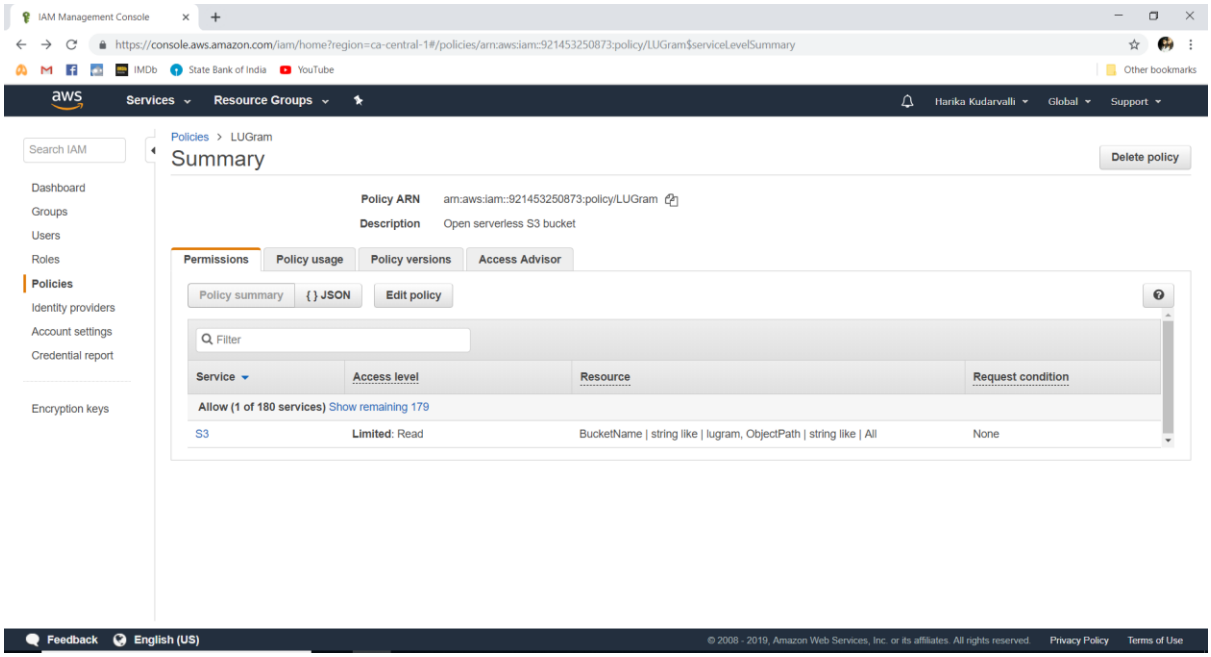
7. CODE

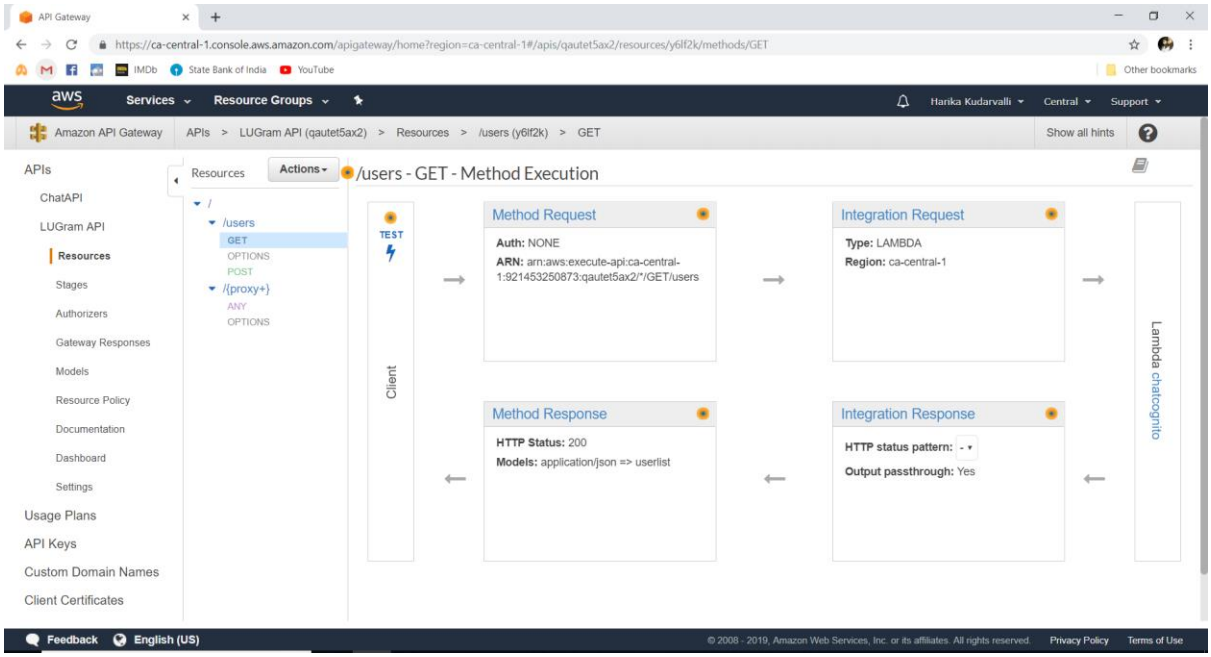

S3 Management Console
LUGram — Share with the Comm
Not secure | www.lugram.com.s3-website.ca-central-1.amazonaws.com
IMDb
State Bank of India
YouTube
Other bookmarks
Type keyword to search...
LUGram
HOME TRAVEL CATEGORIES ABOUT LOG IN SIGN UP CONTACT
TRAVEL
SAMIKSHA · MARCH 15, 2018
How to Find the Video Games of Your Youth
Lorem ipsum dolor sit amet, consectetur adipisicing
www.lugram.com.s3-website.ca-central-1.amazonaws.com/index.html

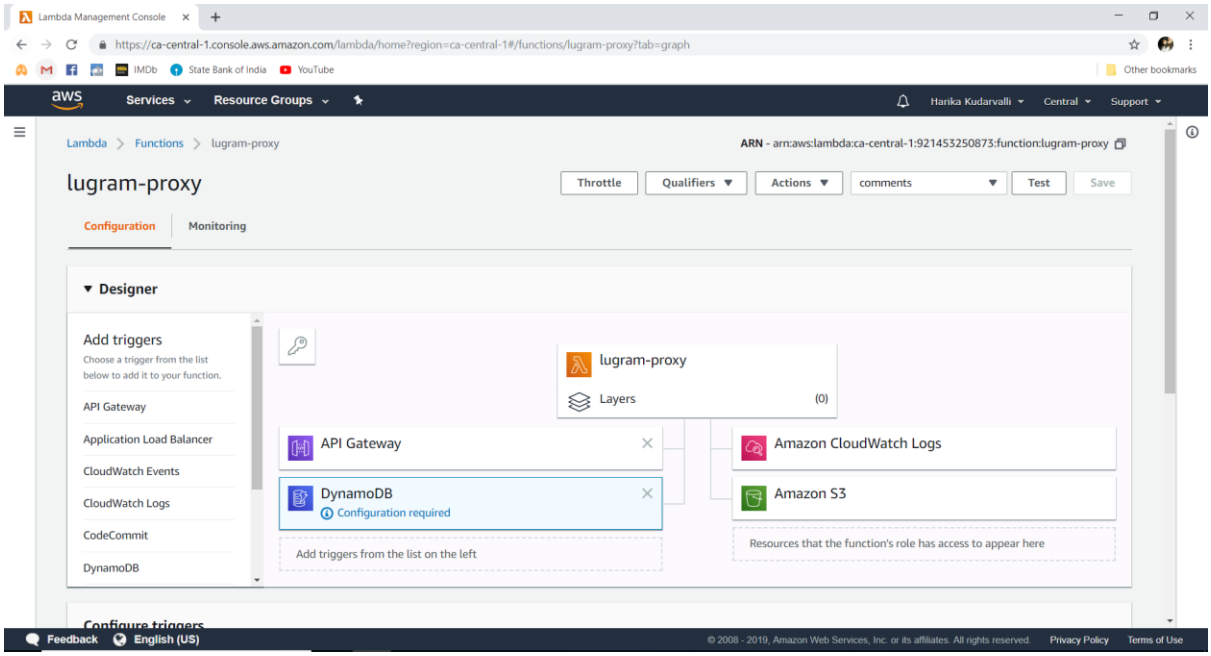

S3 Management Console
https://s3.console.aws.amazon.com/s3/buckets/www.lugram.com/?region=ca-central-1&tab=overview
IMDb
State Bank of India
YouTube
Other bookmarks
aws
Services
Resource Groups
Harika Kudarvalli
Global
Support
Type a prefix and press Enter to search. Press ESC to clear.
Upload
+ Create folder
Download
Actions
Canada (Central)
Name
Last modified
Size
Storage class
css -- -- --
fonts -- -- --
images -- -- --
js -- -- --
register -- -- --
scss -- -- --
.DS_Store Jun 10, 2019 12:33:30 PM GMT-0400 6.0 KB Standard
about.html Jun 10, 2019 12:33:30 PM GMT-0400 24.3 KB Standard
blog-single.html Jun 10, 2019 12:33:30 PM GMT-0400 25.8 KB Standard
category.html Jun 10, 2019 12:33:30 PM GMT-0400 18.3 KB Standard
contact.html Jun 10, 2019 12:33:30 PM GMT-0400 16.3 KB Standard
index.html Jun 10, 2019 12:33:30 PM GMT-0400 24.4 KB Standard
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

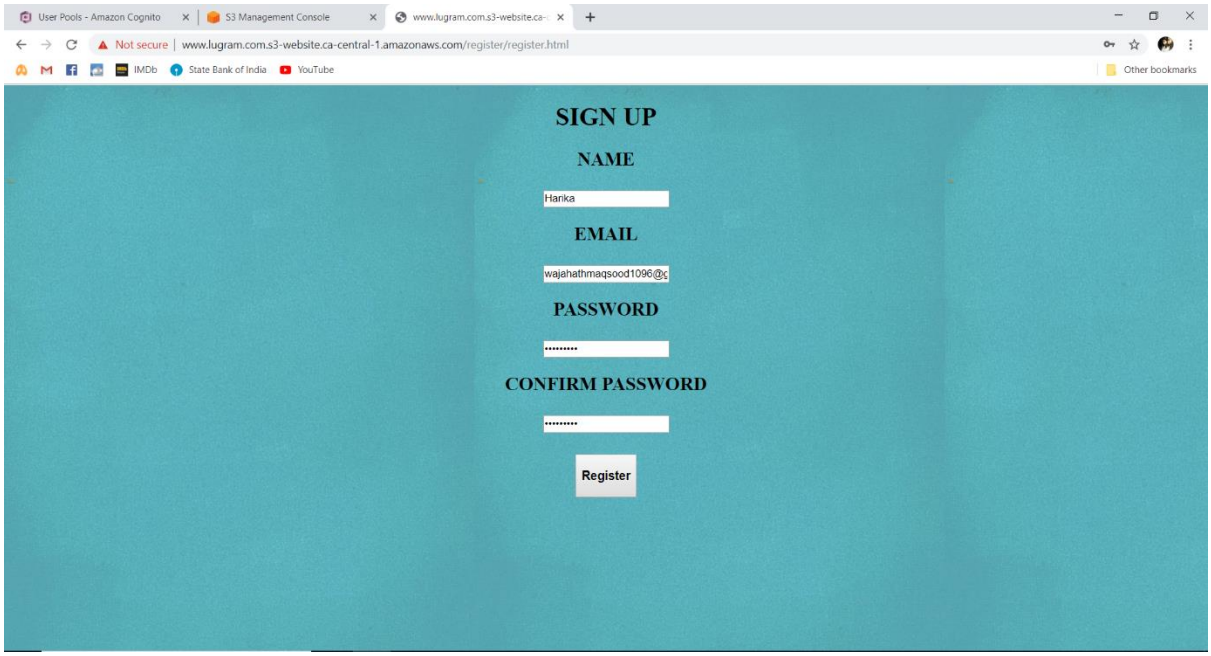

S3 Management Console
https://s3.console.aws.amazon.com/s3/buckets/www.lugram.com/?region=ca-central-1&tab=permissions
IMDb
State Bank of India
YouTube
Other bookmarks
aws
Services
Resource Groups
Harika Kudarvalli
Global
Support
Amazon S3 > www.lugram.com
Overview
Properties
Permissions
Management
Block public access
Access Control List
Bucket Policy
Public
CORS configuration
Bucket policy editor ARN: arn:aws:s3:::www.lugram.com
Type to add a new policy or edit an existing policy in the text area below.
Delete
Cancel
Save
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "PublicReadGetObject",
"Effect": "Allow",
"Principal": {
"AWS": "*"
},
"Action": "s3:GetObject",
"Resource": "arn:aws:s3:::www.lugram.com/*"
}
]
}
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

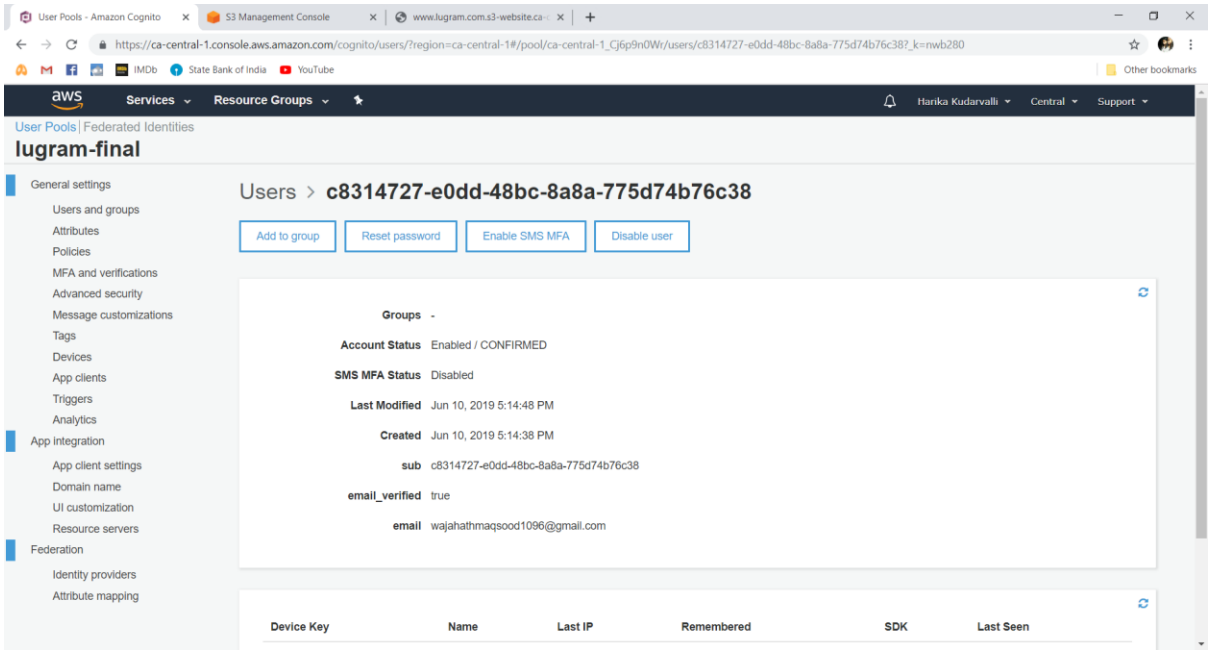

IAM Management Console
https://console.aws.amazon.com/iam/home?region=ca-central-1#/roles/lugram-lambda
IMDb
State Bank of India
YouTube
Other bookmarks
aws
Services
Resource Groups
Harika Kudarvalli
Global
Support
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Roles > lugram-lambda
Summary
Delete role
Role ARN
arn:aws:iam::921453250873:role/lugram-lambda
Role description
Allows Lambda functions to call AWS services on your behalf. | Edit
Instance Profile ARNs
Path
/
Creation time
2019-06-06 13:04 EDT
Maximum CLI/API session duration
1 hour Edit
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (2 policies applied)
Attach policies
Add inline policy
Policy name
Policy type
LUGram
Managed policy
AWSLambdaBasicExecutionRole
AWS managed policy
Permissions boundary (not set)
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

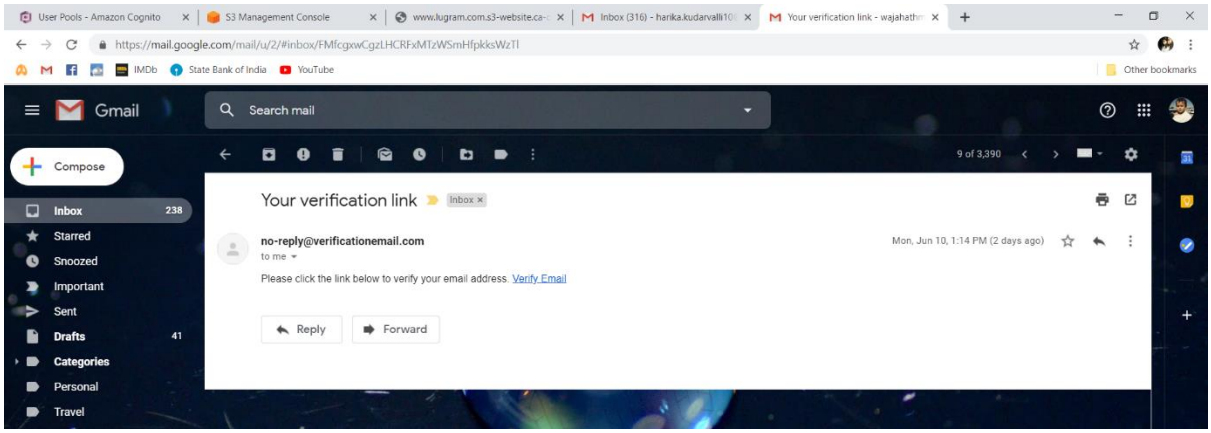

IAM Management Console
https://console.aws.amazon.com/iam/home?region=ca-central-1#/policies/arn:aws:iam::921453250873:policy/LUGram$serviceLevelSummary
Services
Resource Groups
Harika Kudarvalli
Global
Support
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Policies > LUGram
Summary
Delete policy
Policy ARN arn:aws:iam::921453250873:policy/LUGram
Description Open serverless S3 bucket
Permissions
Policy usage
Policy versions
Access Advisor
Policy summary
{ } JSON
Edit policy
Filter
Service
Access level
Resource
Request condition
Allow (1 of 180 services) Show remaining 179
S3
Limited: Read
BucketName | string like | lugram, ObjectPath | string like | All
None
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

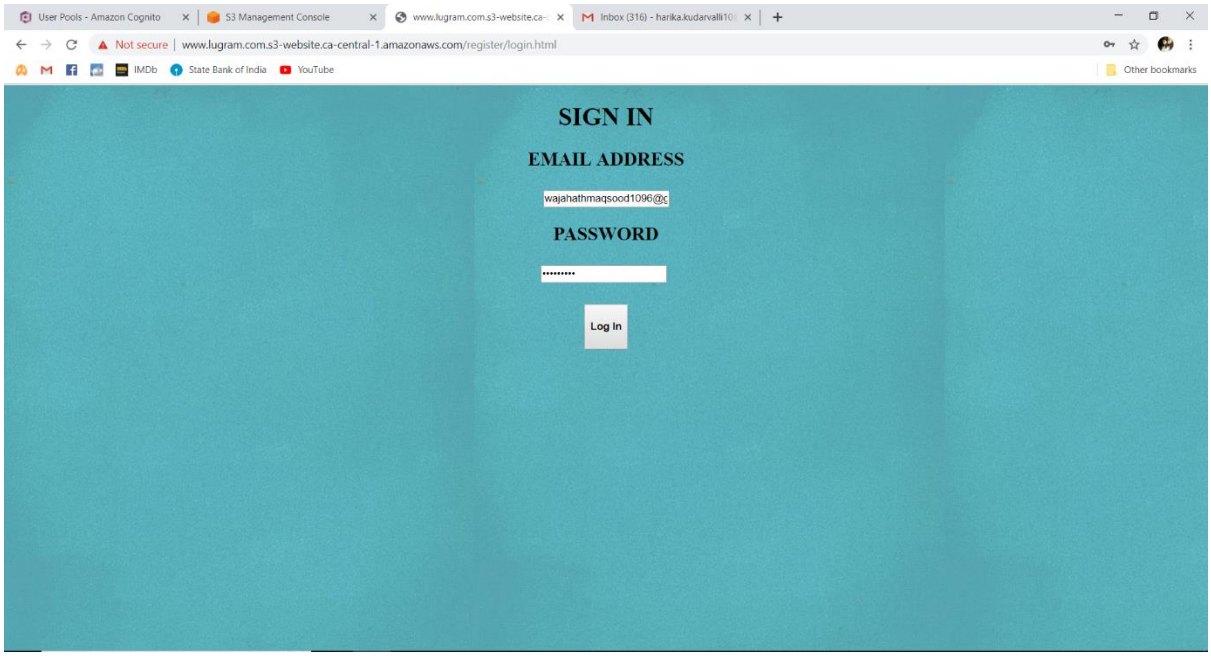

API Gateway
https://ca-central-1.console.aws.amazon.com/apigateway/home?region=ca-central-1#/apis/qautet5ax2/resources/y6lf2k/methods/GET
Services
Resource Groups
Harika Kudarvalli
Central
Support
Amazon API Gateway
APIs > LUGram API (qautet5ax2) > Resources > /users (y6lf2k) > GET
Show all hints
APIs
ChatAPI
LUGram API
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys
Custom Domain Names
Client Certificates
Resources
Actions
/users - GET - Method Execution
/
/users
GET
OPTIONS
POST
/{proxy+}
ANY
OPTIONS
TEST
Client
Method Request
Auth: NONE
ARN: arn:aws:execute-api:ca-central-1:921453250873:qautet5ax2/*/GET/users
Integration Request
Type: LAMBDA
Region: ca-central-1
Lambda chatcognito
Method Response
HTTP Status: 200
Models: application/json => userlist
Integration Response
HTTP status pattern: -
Output passthrough: Yes
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

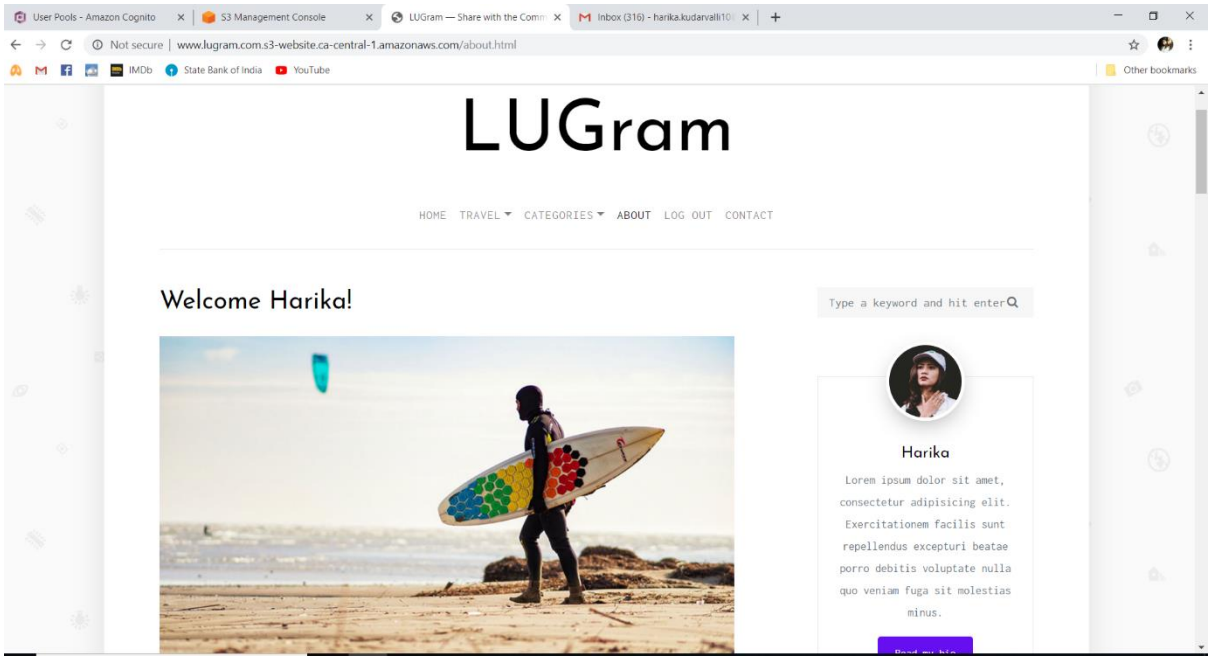

Lambda Management Console
https://ca-central-1.console.aws.amazon.com/lambda/home?region=ca-central-1#/functions/lugram-proxy?tab=graph
aws
Services
Resource Groups
Harika Kudarvalli
Central
Support
Lambda > Functions > lugram-proxy
ARN - arn:aws:lambda:ca-central-1:921453250873:function:lugram-proxy
lugram-proxy
Throttle
Qualifiers
Actions
comments
Test
Save
Configuration
Monitoring
Designer
Add triggers
Choose a trigger from the list below to add it to your function.
API Gateway
Application Load Balancer
CloudWatch Events
CloudWatch Logs
CodeCommit
DynamoDB
lugram-proxy
Layers
(0)
API Gateway
DynamoDB
Configuration required
Add triggers from the list on the left
Amazon CloudWatch Logs
Amazon S3
Resources that the function's role has access to appear here
Configure triggers
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


User Pools - Amazon Cognito
S3 Management Console
www.lugram.com.s3-website.ca-
Not secure
www.lugram.com.s3-website.ca-central-1.amazonaws.com/register/register.html
SIGN UP
NAME
Harika
EMAIL
wajahathmaqsood1096@g
PASSWORD
CONFIRM PASSWORD
Register


User Pools - Amazon Cognito
S3 Management Console
www.lugram.com.s3-website.ca-
https://ca-central-1.console.aws.amazon.com/cognito/users/?region=ca-central-1#/pool/ca-central-1_Cj6p9n0Wr/users/c8314727-e0dd-48bc-8a8a-775d74b76c38?_k=nwb280
IMDb
State Bank of India
YouTube
Other bookmarks
aws
Services
Resource Groups
Harika Kudarvalli
Central
Support
User Pools | Federated Identities
lugram-final
General settings
Users and groups
Attributes
Policies
MFA and verifications
Advanced security
Message customizations
Tags
Devices
App clients
Triggers
Analytics
App integration
App client settings
Domain name
UI customization
Resource servers
Federation
Identity providers
Attribute mapping
Users > c8314727-e0dd-48bc-8a8a-775d74b76c38
Add to group
Reset password
Enable SMS MFA
Disable user
Groups -
Account Status Enabled / CONFIRMED
SMS MFA Status Disabled
Last Modified Jun 10, 2019 5:14:48 PM
Created Jun 10, 2019 5:14:38 PM
sub c8314727-e0dd-48bc-8a8a-775d74b76c38
email_verified true
email wajahathmaqsood1096@gmail.com
Device Key
Name
Last IP
Remembered
SDK
Last Seen


User Pools - Amazon Cognito
S3 Management Console
www.lugram.com.s3-website.ca-
Inbox (316) - harika.kudarvalli10
Your verification link - wajahath
https://mail.google.com/mail/u/2/#inbox/FMfcgxwCgzLHCRFxMTzWSmHfpkksWzTl
IMDb
State Bank of India
YouTube
Other bookmarks
Gmail
Search mail
Compose
Inbox 238
Starred
Snoozed
Important
Sent
Drafts 41
Categories
Personal
Travel
9 of 3,390
Your verification link
Inbox ×
no-reply@verificationemail.com
to me
Mon, Jun 10, 1:14 PM (2 days ago)
Please click the link below to verify your email address. Verify Email
Reply
Forward


User Pools - Amazon Cognito
S3 Management Console
Inbox (316) - harika.kudarvalli10
Not secure | www.lugram.com.s3-website.ca-central-1.amazonaws.com/register/login.html
IMDb
State Bank of India
YouTube
Other bookmarks
SIGN IN
EMAIL ADDRESS
wajahathmaqsood1096@c
PASSWORD
Log In


User Pools - Amazon Cognito
S3 Management Console
LUGram — Share with the Comm
Inbox (316) - harika.kudarvalli10
Not secure | www.lugram.com.s3-website.ca-central-1.amazonaws.com/about.html
IMDb
State Bank of India
YouTube
Other bookmarks
LUGram
HOME TRAVEL CATEGORIES ABOUT LOG OUT CONTACT
Welcome Harika!
Type a keyword and hit enter
Harika
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Exercitationem facilis sunt repellendus excepturi beatae porro debitis voluptate nulla quo veniam fuga sit molestias minus.

9. RESULTS

Click the following text for the link.

PRESENTATION DEMO

10. FUTURE WORK

To develop a messaging service among different users.

11. CONCLUSION

LUGRAM is a photo sharing web application which has been hosted by Amazon Web Services. The AWS platform allows the user to create, program, deploy and amend changes to the application all at one stop. In brief, Amazon S3 is used to deploy the application, AWS Lambda has been used for triggering functions and Amazon Cognito for security and user's authentication. This application can be accessed publicly. Users can sign up and log in using secured credentials. Individual posts and photos can be posted with description and tags. This application has been designed for our university students who can post updates about different events being conducted all over the campus. It is a user - friendly design and can be accessed by students and teachers regardless. AWS has enabled us to host a website without directly engaging with servers. It manages the back end of the application which is useful for users who are unaware about technological terms. As mentioned in future work, the application can also have interacting messaging service among the users. In conclusion, creating applications is much more easier using Amazon Web Services which provides a user friendly, highly malleable and structured framework.

REFERENCES

1. Build a Serverless web application ,<https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/>
2. Performance Analysis of High-Performance Computing Applications on the Amazon Web Services Cloud by K. R. Jackson
3. What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software by Tim O'Reilly
4. Digital Images, Photo-Sharing, and Our Shifting Notions of Everyday Aesthetics by Susan Murray
5. Using RESTful web-services and cloud computing to create next generation mobile applications by Jason H. Christensen
6. Automatic creation of bidirectional online album links in a peer-to-peer photo sharing network by Alfredo C. Issa
7. Instagram at the museum: communicating the museum experience through social photo sharing by Alexandra Weilanman
8. Serverless applications with AWS, < https://www.manning.com/livevideo/serverless-applications-with-AWS>