DeployZen

Document Information :

Name: DeployZen - Fast, easy, Automate Deployment

Team Members:

G.L.Sudhitha	-	23P31A4225
G.N.R.L.Jayanthi	-	23P31A4227
I.Amrutha Varshini	-	23P31A4229
koonisetti Mahesh	-	23P31A0528
J.DYNS.Gowrish	-	23P31A0537
M.Bharghav Sai	-	23P31A0503

Abstract

DeployZen is a cloud-based platform that simplifies static website deployment for students and developers. It allows users to upload a ZIP file or connect a GitHub repository, and automatically handles hosting on AWS S3 using Lambda, DynamoDB, and a Flask backend. The frontend, built with HTML, CSS, and basic Tailwind, provides a simple user interface. Features include secure authentication via AWS Cognito and a public project showcase through ZenHub. DeployZen removes the complexity of cloud setup, making static site deployment fast, easy, and beginner-friendly.

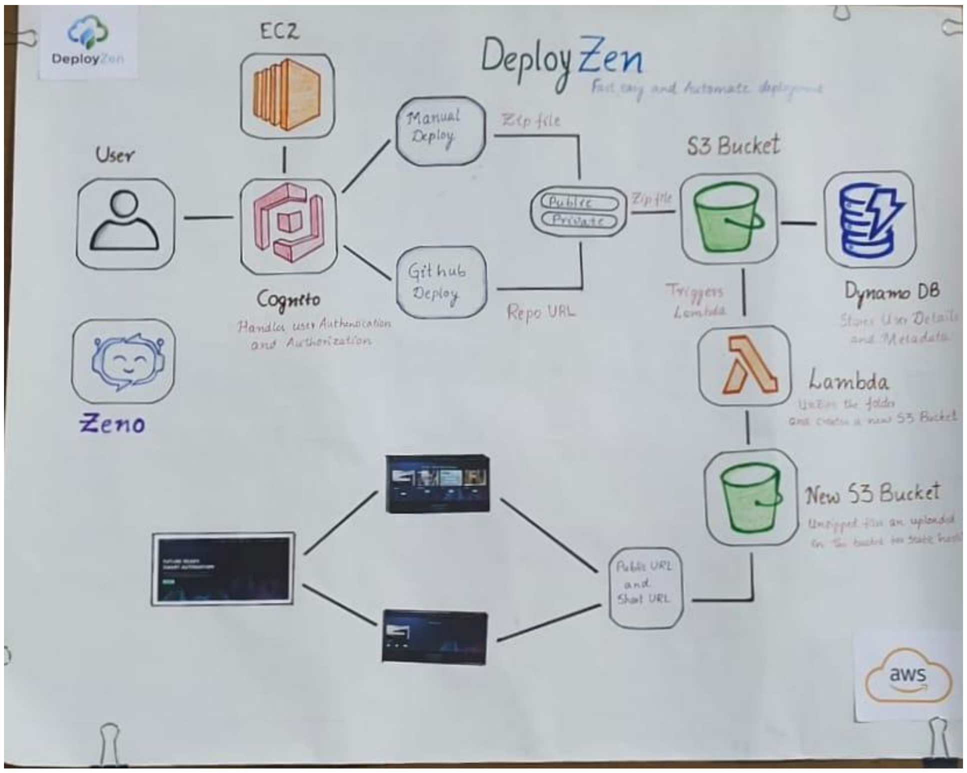
Problem Overview

We chose to build DeployZen to solve the common difficulties faced by beginners and students when deploying static websites to the cloud. Although platforms like AWS offer powerful hosting options, the manual setup—such as configuring S3 buckets, setting permissions, and managing public URLs—can be complex and time-consuming. Our goal was to create a simple, automated solution that makes static site deployment quick, accessible, and beginner-friendly.

Objectives

- Enable quick and seamless static website deployment
- Provide both **ZIP file** and **GitHub repo** upload options
- Automate deployment using AWS Lambda
- Store project metadata in DynamoDB
- Secure user access with AWS Cognito
- Display deployed websites on a public "ZenHub" page

Architecture


DeployZen
ECZ
DeployZen
Fast easy and Automate deployment
User
Manual Deploy
Zip file
S3 Bucket
Public
Private
Zip file
Cognito
Handles user Authentication and Authorization
Github Deploy
Repo URL
Triggers Lambda
Dynamo DB
Stores User Details and Metadata
Zeno
Lambda
Unzips the folder and creates a new S3 Bucket
New S3 Bucket
Unzipped files are uploaded in the bucket for static hosting
Public URL and Short URL
aws

Services Used And Workflow

1. **Amazon Cognito – User Login & Registration**

We used Cognito for handling user sign-up and login. It manages:

Secure user authentication

Email verification through token-based confirmation

Session and token management.

2. Amazon EC2 – Hosting the Flask Application

Our Flask-based web application is hosted on an EC2 instance. It handles:

Frontend rendering with HTML templates

Backend logic and routing

API endpoints for GitHub/manual deployment, updates, and redirection

EC2 gives full control over the hosting environment.

3. Amazon S3 – File Storage & Static Site Hosting

S3 is used in two stages:

To temporarily store uploaded .zip files from users

To host unzipped website files as static websites

The hosted website becomes publicly accessible via a full URL.

4. AWS Lambda – Unzipping & Hosting Logic

When a .zip file is uploaded to S3, a Lambda function is triggered:

It extracts the zip contents

Creates a new S3 bucket (per deployment)

Uploads the unzipped files

Enables static hosting on the new bucket

Updates the full URL in DynamoDB

This makes the process automatic and scalable without manual hosting steps.

5. Amazon DynamoDB – Deployment Metadata Storage

DynamoDB is used to store all the important metadata:

Unique short Id

Original filename

short URL and the final full URL

This allows us to track and manage user deployments efficiently.

6. GitHub – Public Repository Deployment Source

For users who choose GitHub deployment, we:

Extract the username and repo name from the URL

Download the .zip from main branch

Upload it to S3 for further processing

This enables fast deployment of existing GitHub projects.

7. Amazon Lex – Chatbot Integration

We built a rule-based chatbot using Lex that:

Helps users understand the platform

Guides them step-by-step in choosing deployment methods

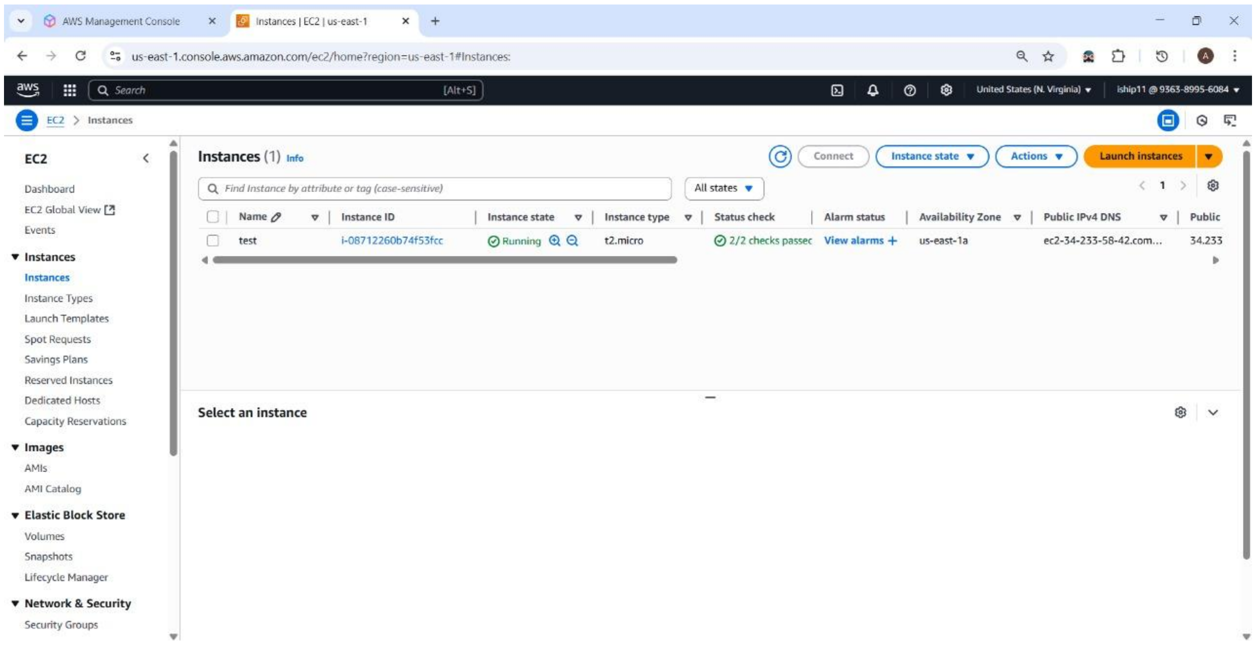
Answers basic FAQs

It improves user experience and makes the platform beginner-friendly.

Project Execution

Step1: EC2 Instance Setup & Flask App Hosting

We launched an EC2 instance and installed necessary packages (Python, pip, Flask). The Flask app handles routing, file uploads, redirections, and UI rendering

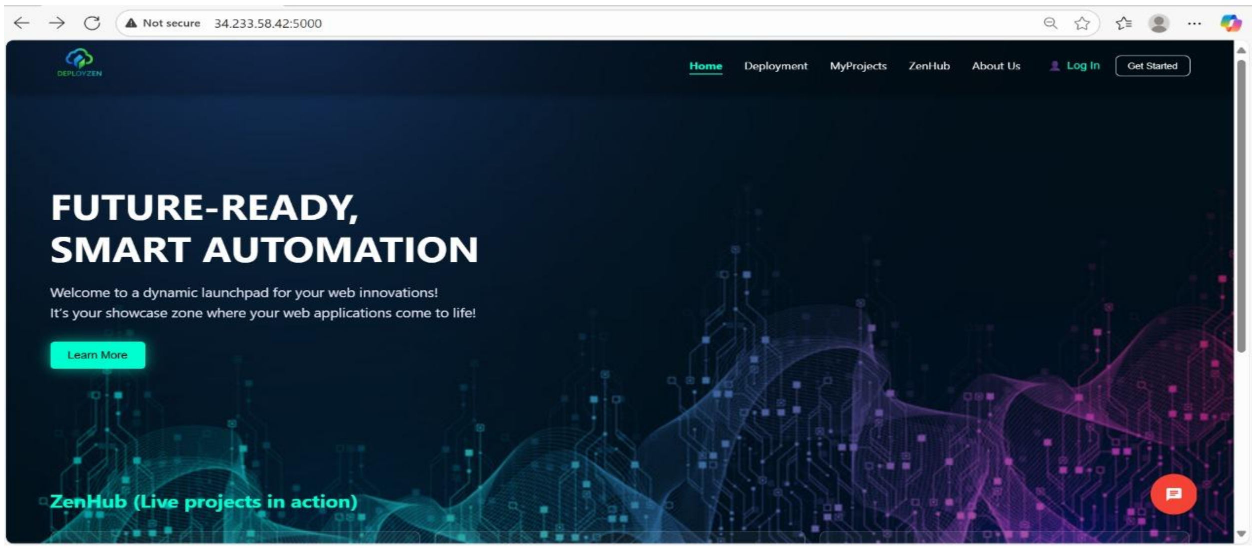


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
test	i-08712260b74f53fcc	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	ec2-34-233-58-42.com...	34.233

Step 2: Frontend Template Design

We created HTML templates for home page, deployment pages (manual, GitHub), showcase, update, etc. basic TailwindCSS was used for styling.

➢ **Home Page:**

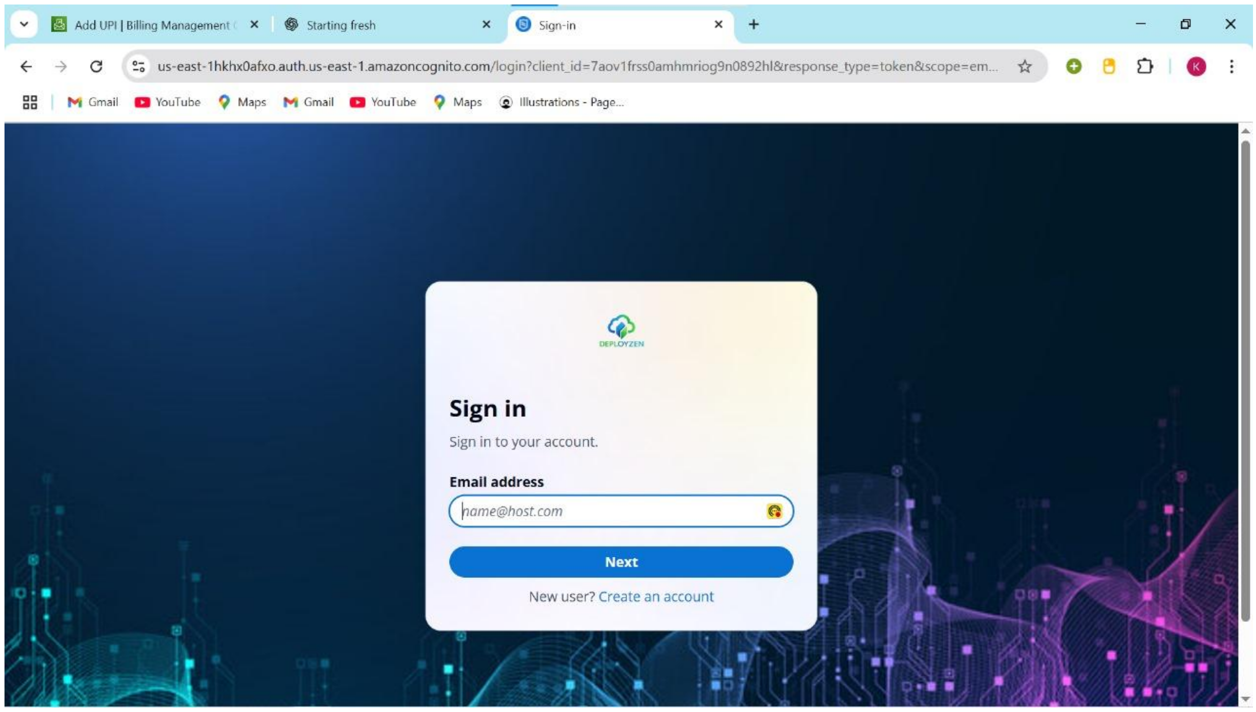


FUTURE-READY, SMART AUTOMATION

Welcome to a dynamic launchpad for your web innovations!
It's your showcase zone where your web applications come to life!

ZenHub (Live projects in action)

➢ **User Sign/Sinup:**


Add UPI | Billing Management
Starting fresh
Sign-in
us-east-1hkhx0afxo.auth.us-east-1.amazoncognito.com/login?client_id=7aov1frss0amhmriog9n0892hl&response_type=token&scope=em...
Gmail
YouTube
Maps
Gmail
YouTube
Maps
Illustrations - Page...
DEPLOYZEN

Sign in

Sign in to your account.

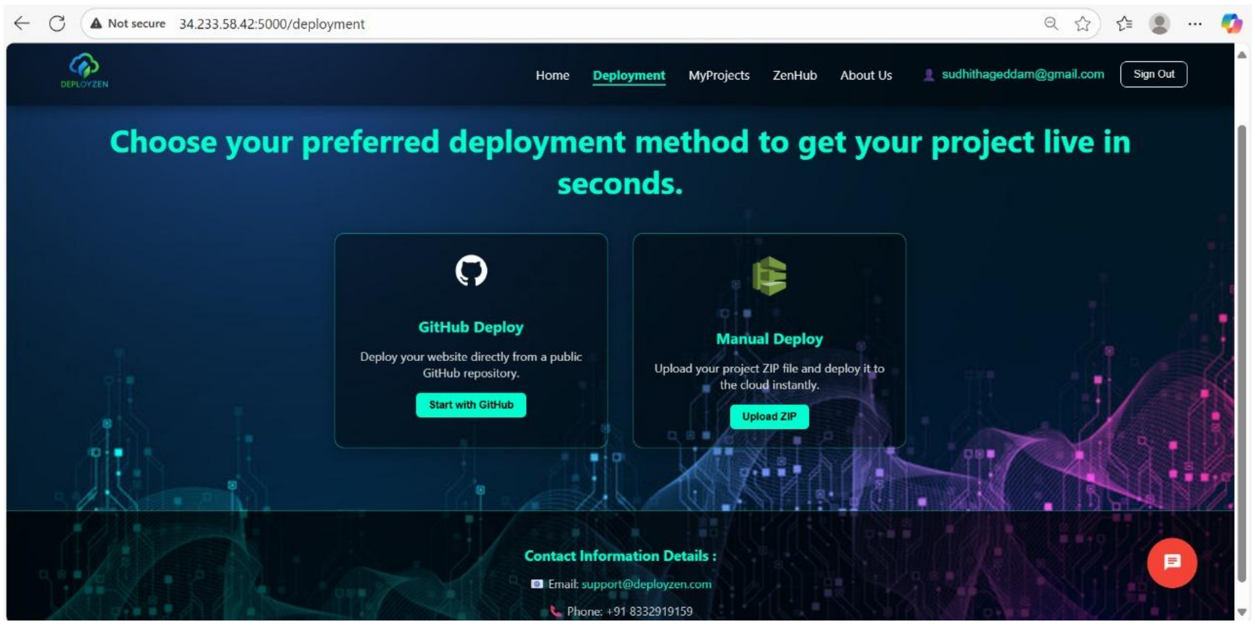
Email address

name@host.com

Next

New user? Create an account

➢ **Deployment page:**



Not secure 34.233.58.42:5000/deployment

DEPLOYZEN Home Deployment MyProjects ZenHub About Us sudhithageddam@gmail.com Sign Out

Choose your preferred deployment method to get your project live in seconds.

GitHub Deploy

Deploy your website directly from a public GitHub repository.

Start with GitHub

Manual Deploy

Upload your project ZIP file and deploy it to the cloud instantly.

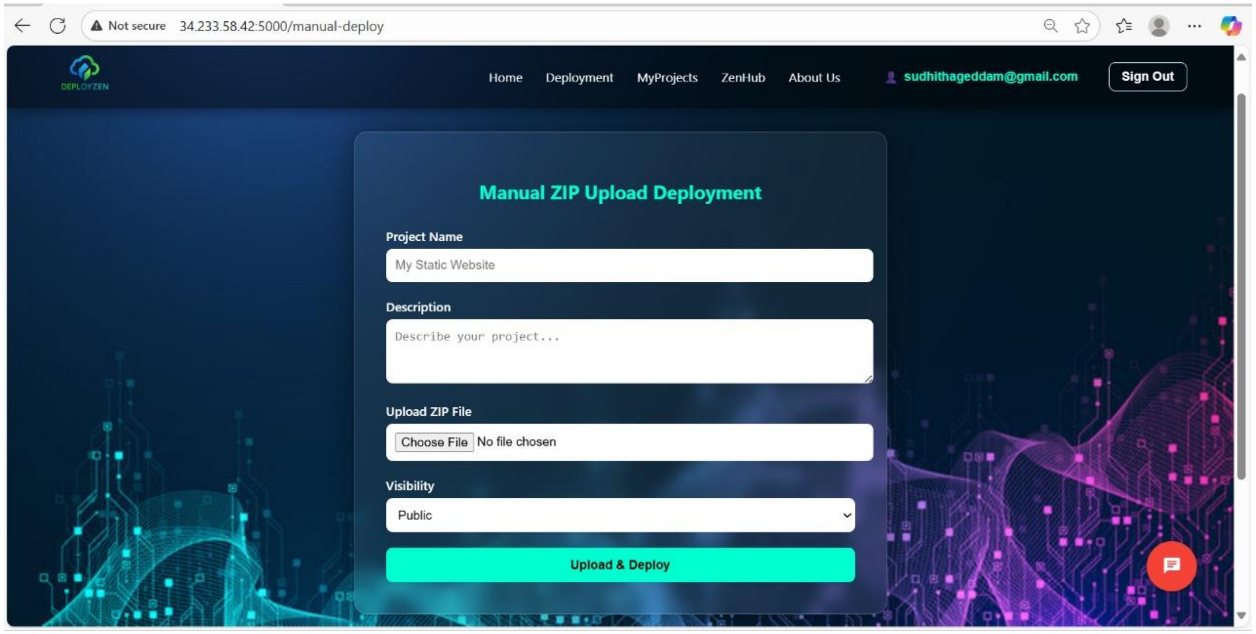
Upload ZIP

Contact Information Details :

Email: support@deployzen.com

Phone: +91 8332919159

➢ **Manual deploy:**



Not secure 34.233.58.42:5000/manual-deploy

DEPLOYZEN Home Deployment MyProjects ZenHub About Us sudhithageddam@gmail.com Sign Out

Manual ZIP Upload Deployment

Project Name

My Static Website

Description

Describe your project...

Upload ZIP File

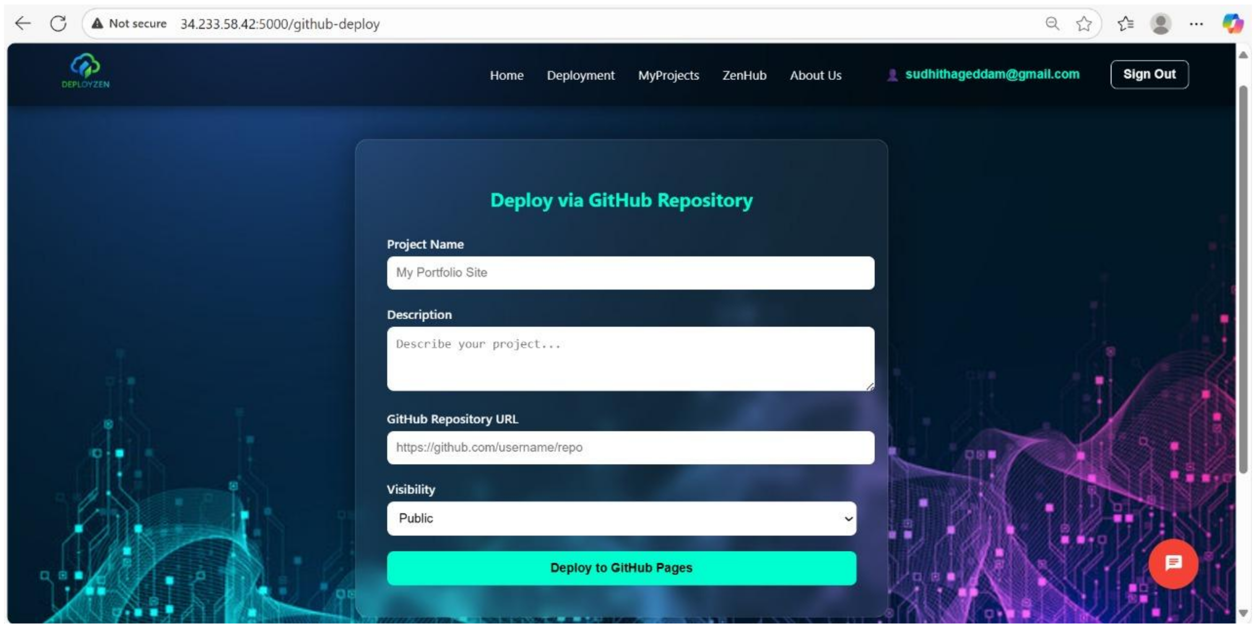
Choose File No file chosen

Visibility

Public

Upload & Deploy

➢ Github Rep Deploy:



Not secure 34.233.58.42:5000/github-deploy

Deploy via GitHub Repository

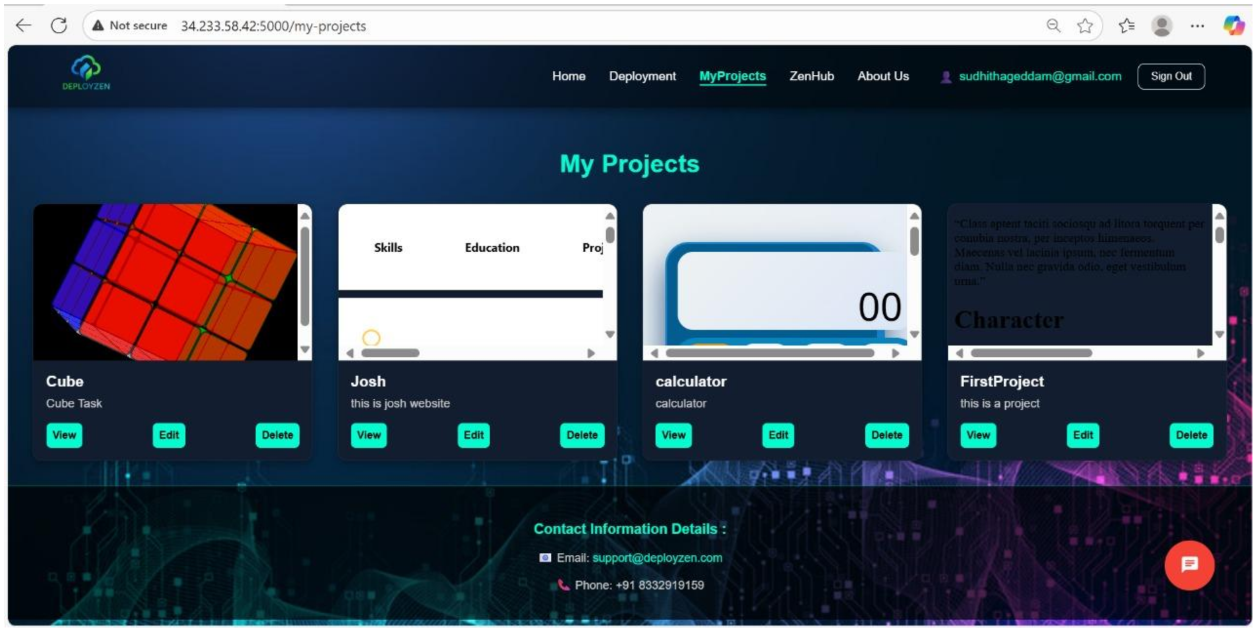
Project Name: My Portfolio Site

Description: Describe your project...

GitHub Repository URL: https://github.com/username/repo

Visibility: Public

➢ My Project Section:



Not secure 34.233.58.42:5000/my-projects

My Projects

Cube
Cube Task

Josh
this is josh website

calculator
calculator

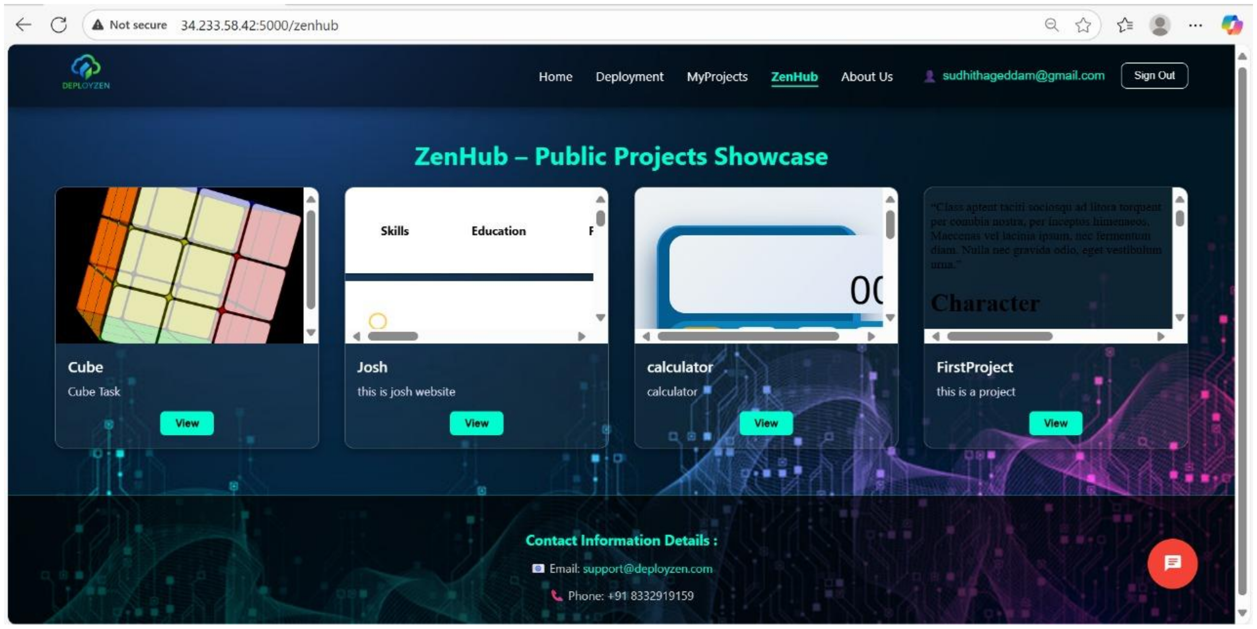
FirstProject
this is a project

Contact Information Details :

Email: support@deployzen.com

Phone: +91 8332919159

➢ **ZenHub Page:**



ZenHub – Public Projects Showcase

Cube
Cube Task
View

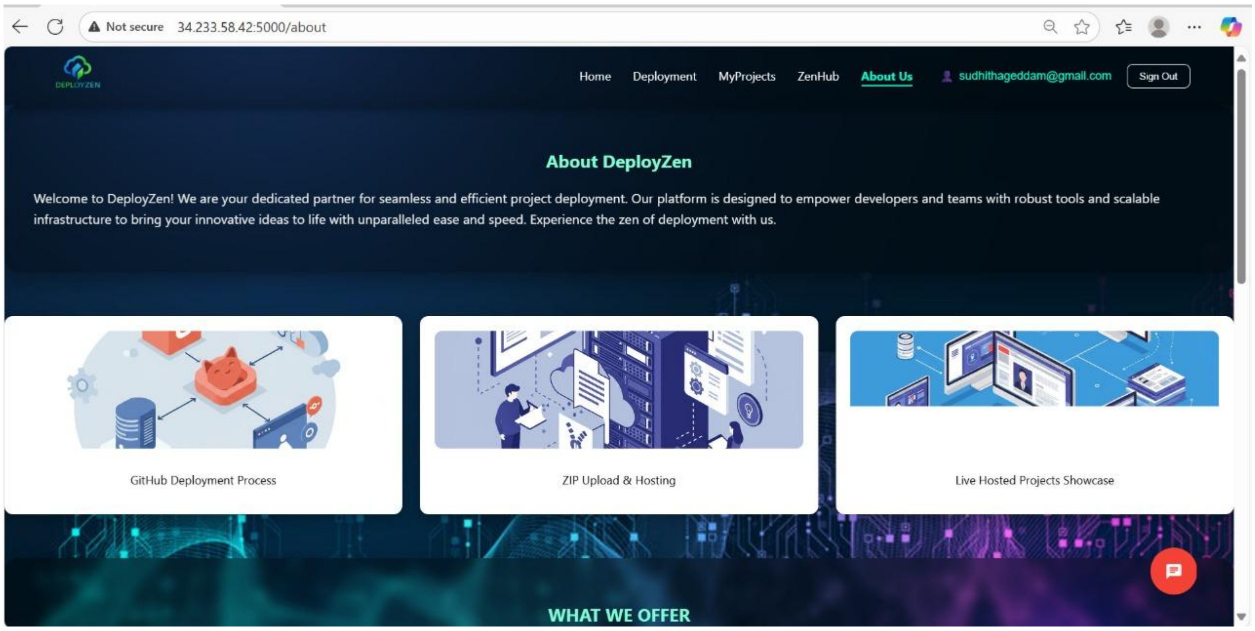
Josh
this is josh website
View

calculator
calculator
View

FirstProject
this is a project
View

Contact Information Details :
Email: support@deployzen.com
Phone: +91 8332919159

➢ **About Us Page:**



About DeployZen

Welcome to DeployZen! We are your dedicated partner for seamless and efficient project deployment. Our platform is designed to empower developers and teams with robust tools and scalable infrastructure to bring your innovative ideas to life with unparalleled ease and speed. Experience the zen of deployment with us.

GitHub Deployment Process

ZIP Upload & Hosting

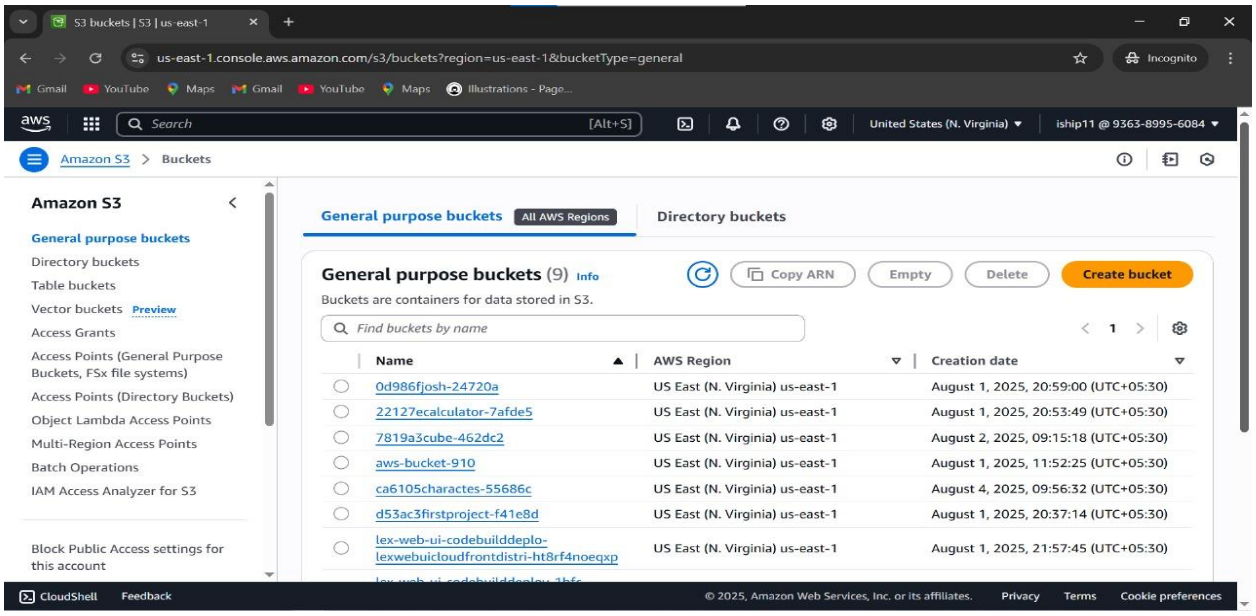
Live Hosted Projects Showcase

WHAT WE OFFER

Step 3: S3 & DynamoDB Integration:

The uploaded ZIP files (from manual or GitHub deployment) are stored in Amazon S3. We used DynamoDB to save metadata like filename, short URL, and status.

AWS Console showing files in S3 bucket

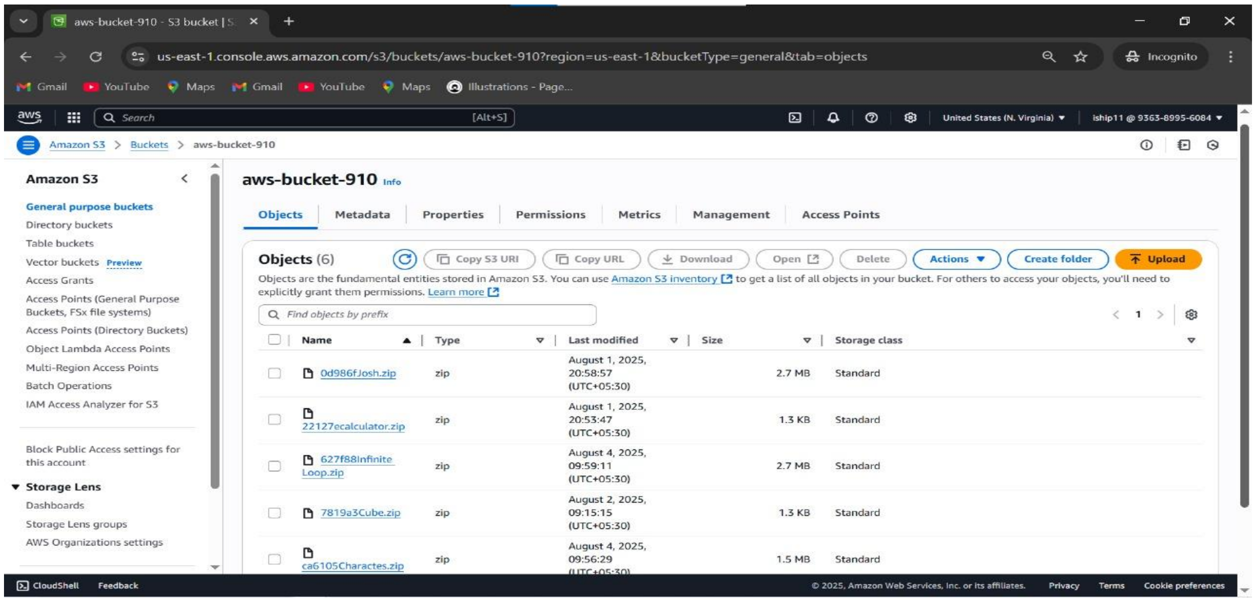


us-east-1.console.aws.amazon.com/s3/buckets?region=us-east-1&bucketType=general

General purpose buckets (9)

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
0d986fjosh-24720a	US East (N. Virginia) us-east-1	August 1, 2025, 20:59:00 (UTC+05:30)
22127ecalculator-7afde5	US East (N. Virginia) us-east-1	August 1, 2025, 20:53:49 (UTC+05:30)
7819a3cube-462dc2	US East (N. Virginia) us-east-1	August 2, 2025, 09:15:18 (UTC+05:30)
aws-bucket-910	US East (N. Virginia) us-east-1	August 1, 2025, 11:52:25 (UTC+05:30)
ca6105charactes-55686c	US East (N. Virginia) us-east-1	August 4, 2025, 09:56:32 (UTC+05:30)
d53ac3firstproject-f41e8d	US East (N. Virginia) us-east-1	August 1, 2025, 20:37:14 (UTC+05:30)
lex-web-ui-codebuilddeplo-lexwebuicloudfrontdistri-ht8rf4noeqxp	US East (N. Virginia) us-east-1	August 1, 2025, 21:57:45 (UTC+05:30)



us-east-1.console.aws.amazon.com/s3/buckets/aws-bucket-910?region=us-east-1&bucketType=general&tab=objects

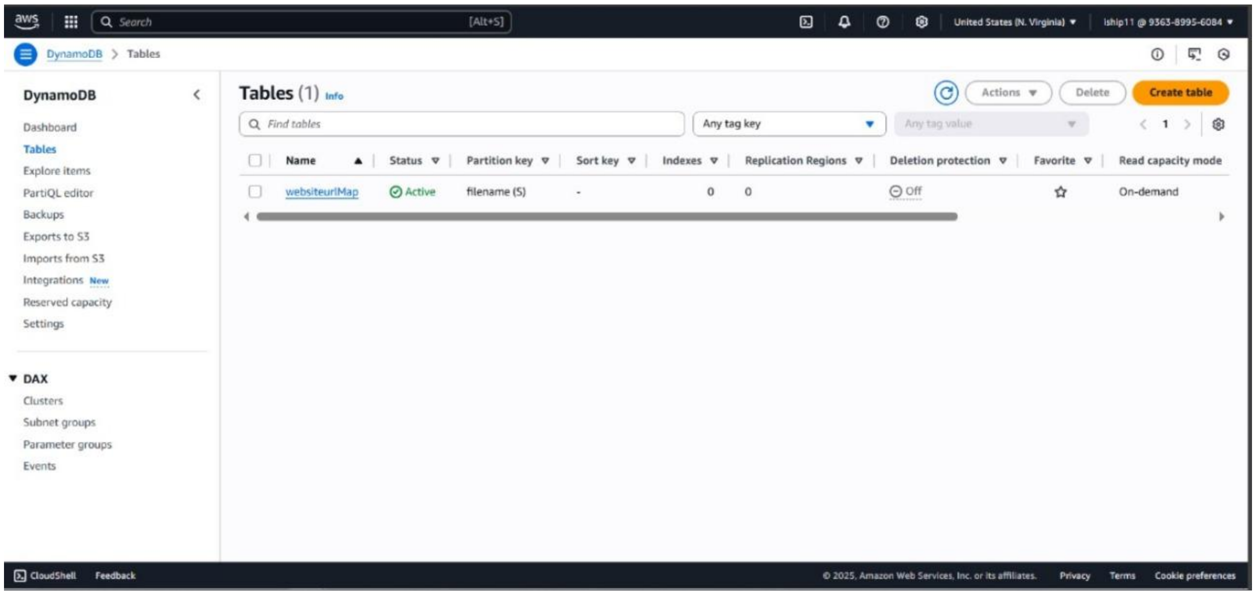
aws-bucket-910

Objects (6)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

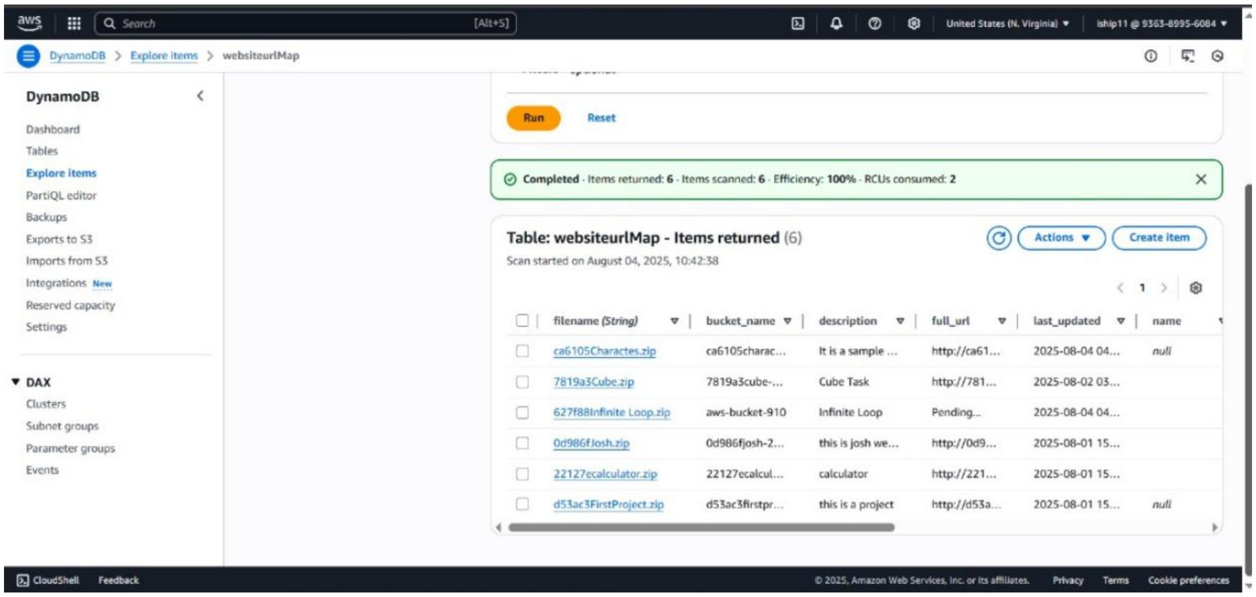
Name	Type	Last modified	Size	Storage class
0d986fJosh.zip	zip	August 1, 2025, 20:58:57 (UTC+05:30)	2.7 MB	Standard
22127ecalculator.zip	zip	August 1, 2025, 20:53:47 (UTC+05:30)	1.3 KB	Standard
627f88Infinite Loop.zip	zip	August 4, 2025, 09:59:11 (UTC+05:30)	2.7 MB	Standard
7819a3Cube.zip	zip	August 2, 2025, 09:15:15 (UTC+05:30)	1.3 KB	Standard
ca6105Charactes.zip	zip	August 4, 2025, 09:56:29 (UTC+05:30)	1.5 MB	Standard

DynamoDB table with items.



Tables (1)

	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode
	websiteurlMap	Active	filename (S)	-	0	0	Off	☆	On-demand



Completed · Items returned: 6 · Items scanned: 6 · Efficiency: 100% · RCUs consumed: 2

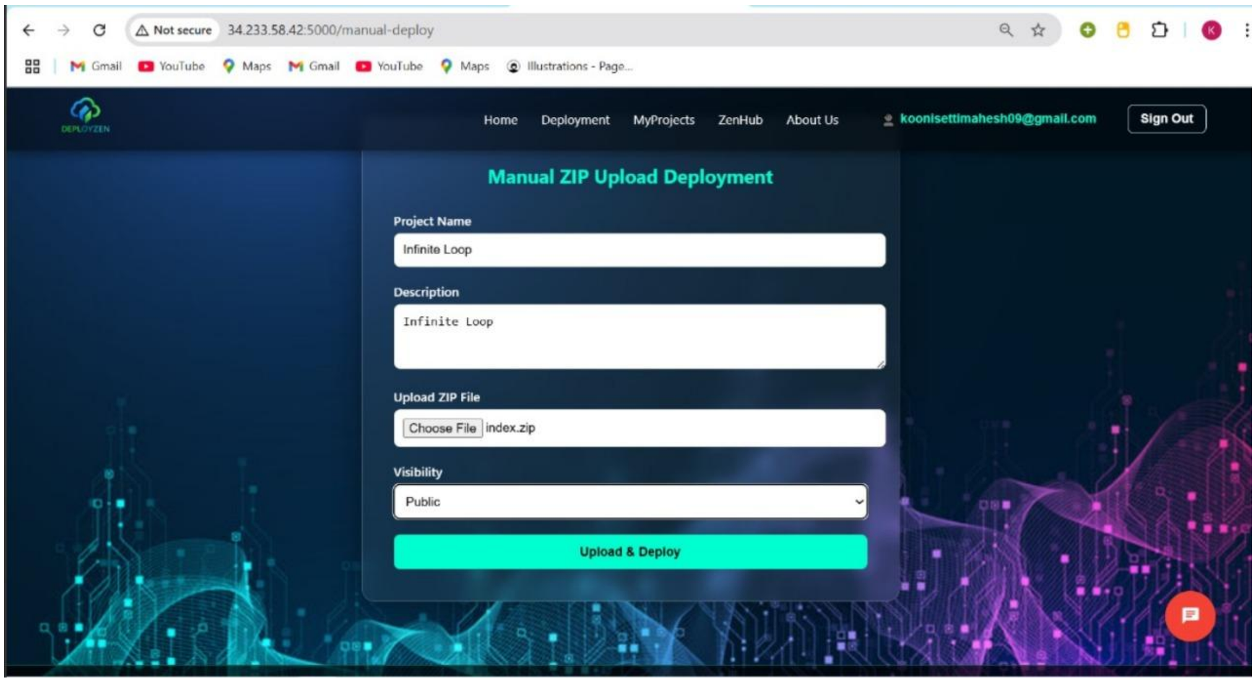
Table: websiteurlMap - Items returned (6)

Scan started on August 04, 2025, 10:42:38

filename (String)	bucket_name	description	full_url	last_updated	name
ca6105Charactes.zip	ca6105charac...	It is a sample ...	http://ca61...	2025-08-04 04...	null
7819a3Cube.zip	7819a3cube-...	Cube Task	http://781...	2025-08-02 03...	
627f88Infinite Loop.zip	aws-bucket-910	Infinite Loop	Pending...	2025-08-04 04...	
0d986fJosh.zip	0d986fjosh-2...	this is josh we...	http://0d9...	2025-08-01 15...	
22127ecalculator.zip	22127ecalcul...	calculator	http://221...	2025-08-01 15...	
d53ac3FirstProject.zip	d53ac3firstpr...	this is a project	http://d53a...	2025-08-01 15...	null

Step 4: Manual Deployment Flow

Users upload a ZIP file, which is stored in S3 and saved in DynamoDB. A short URL is generated and a Lambda function is triggered for deployment.



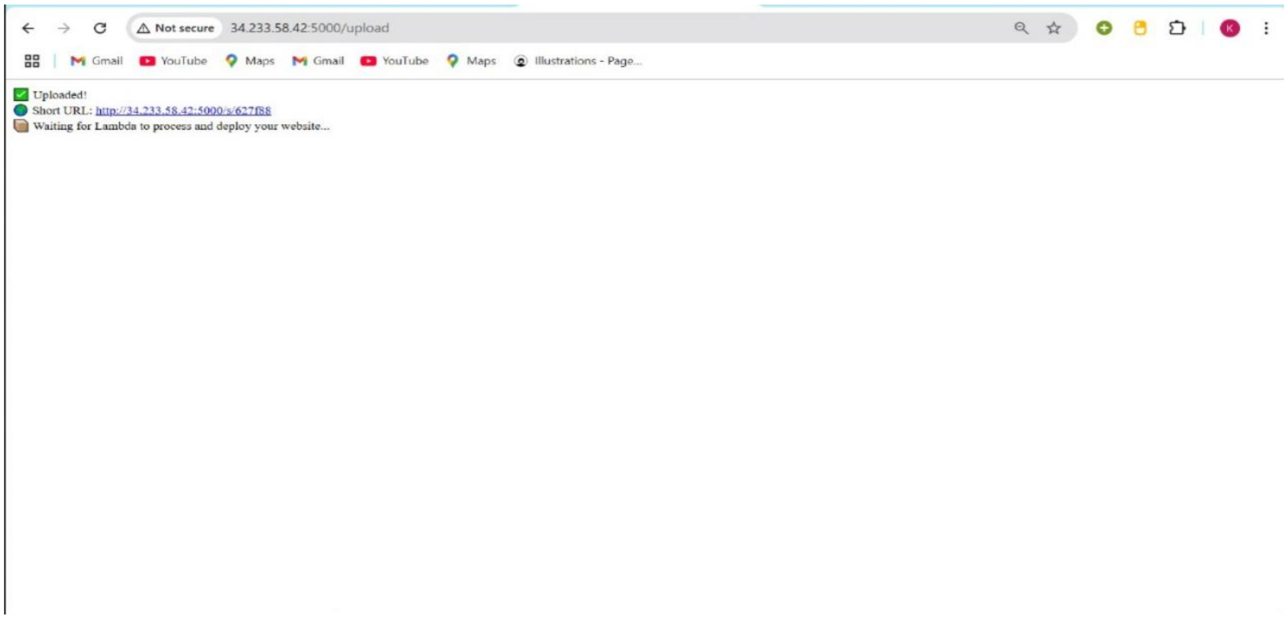
Manual ZIP Upload Deployment

Project Name: Infinite Loop

Description: Infinite Loop

Upload ZIP File: Choose File index.zip

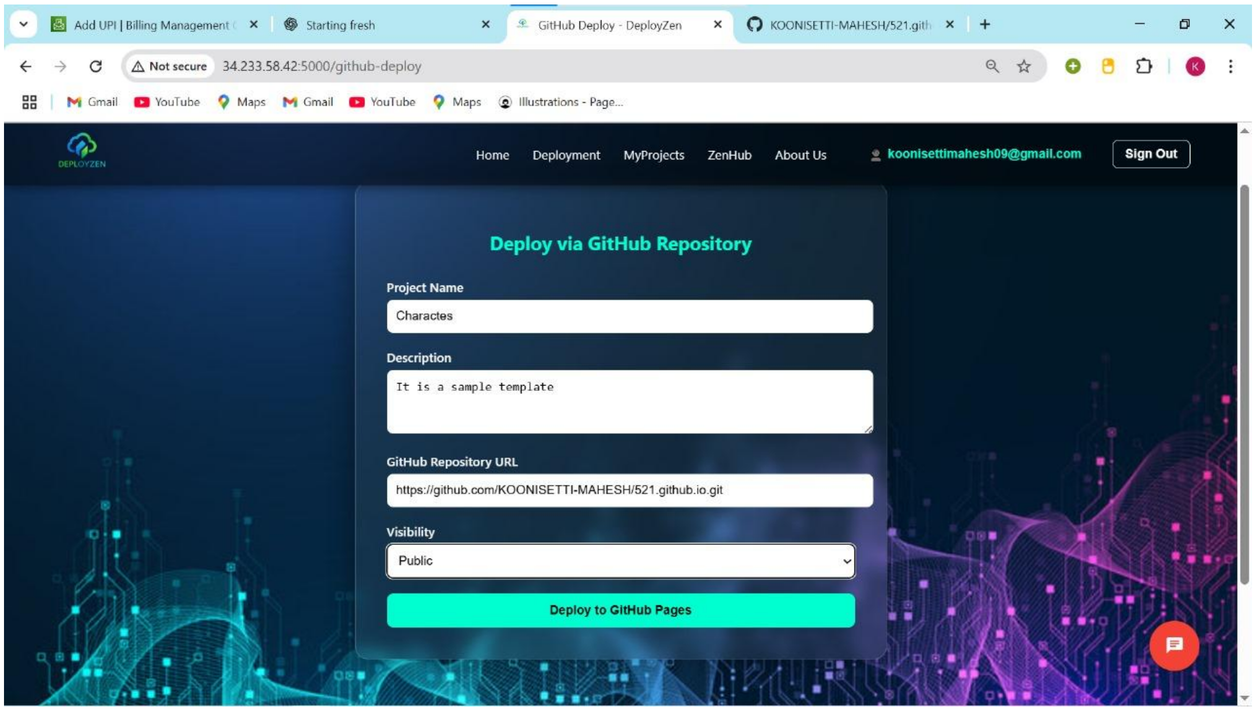
Visibility: Public



```
✅ Uploaded!
🌐 Short URL: http://34.233.58.42:5000/s/627f88
📦 Waiting for Lambda to process and deploy your website...
```

Step 5: GitHub Deployment Flow

Users paste a GitHub repo URL. The backend fetches the ZIP from GitHub, uploads it to S3, and saves metadata. Lambda handles the rest.



Deploy via GitHub Repository

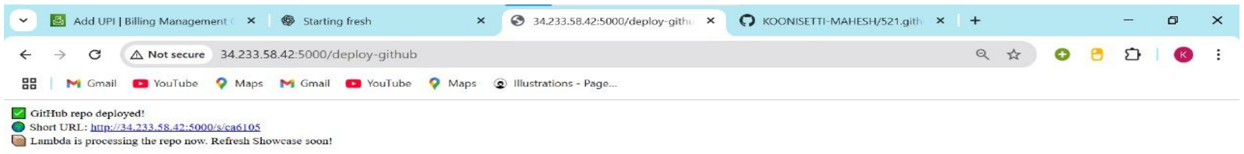
Project Name: Charactes

Description: It is a sample template

GitHub Repository URL: https://github.com/KOONISETTI-MAHESH/521.github.io.git

Visibility: Public

Deploy to GitHub Pages

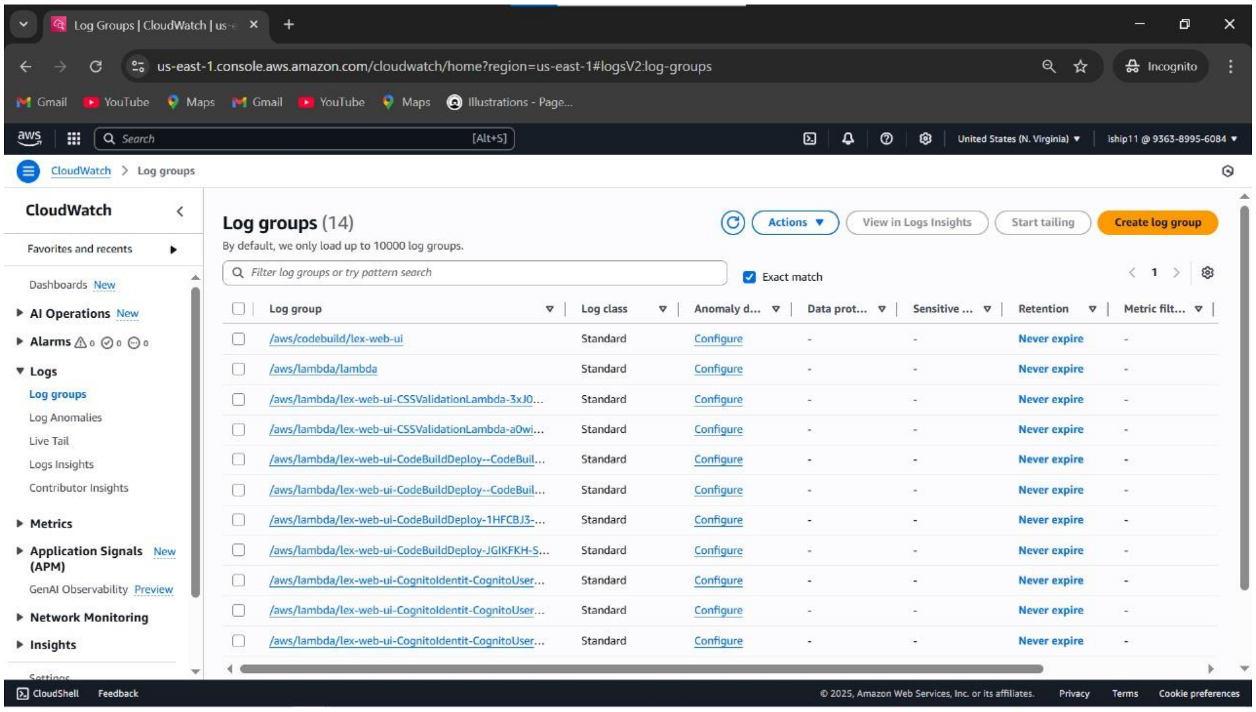


```
✅ GitHub repo deployed!
🌍 Short URL: http://34.233.58.42:5000/s/ca6105
📦 Lambda is processing the repo now. Refresh Showcase soon!
```

Step 6: Lambda Function for Hosting

Lambda gets triggered when a file is uploaded to the S3 bucket. It unzips the file, creates a new bucket, uploads files, and enables static website hosting.

CloudWatch logs of Lambda execution.


Log Groups | CloudWatch | us-
us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups
CloudWatch > Log groups
Actions
View in Logs Insights
Start tailing
Create log group

Log groups (14)

By default, we only load up to 10000 log groups.

Filter log groups or try pattern search ✓ Exact match

Log group	Log class	Anomaly d...	Data prot...	Sensitive ...	Retention	Metric filt...
/aws/codebuild/lex-web-ui	Standard	Configure	-	-	Never expire	-
/aws/lambda/lambda	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CSSValidationLambda-3xJ0...	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CSSValidationLambda-a0wi...	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CodeBuildDeploy--CodeBuil...	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CodeBuildDeploy--CodeBuil...	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CodeBuildDeploy-1HFCBJ3-...	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CodeBuildDeploy-JGIKFKH-S...	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CognitoIdentit-CognitoUser...	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CognitoIdentit-CognitoUser...	Standard	Configure	-	-	Never expire	-
/aws/lambda/lex-web-ui-CognitoIdentit-CognitoUser...	Standard	Configure	-	-	Never expire	-

Step 7: IAM Roles and Policies – Secure Service Access

➢ Lambda Execution Role

Roles:

- Grants Lambda access to:
- Read/write from S3
- Perform operations on DynamoDB
- Log to CloudWatch

Attached Policies:

- AmazonS3FullAccess
- AmazonDynamoDBFullAccess
- CloudWatchLogsFullAccess (or a custom least-privilege policy)

➢ EC2 Instance Role

Roles:

- Assigned to the EC2 instance running the Flask backend, enabling it to:
  - Upload and manage user files in Amazon S3
  - Send logs and monitoring data to CloudWatch

Attached Policies:

- AmazonS3FullAccess
- CloudWatchAgentServerPolicy

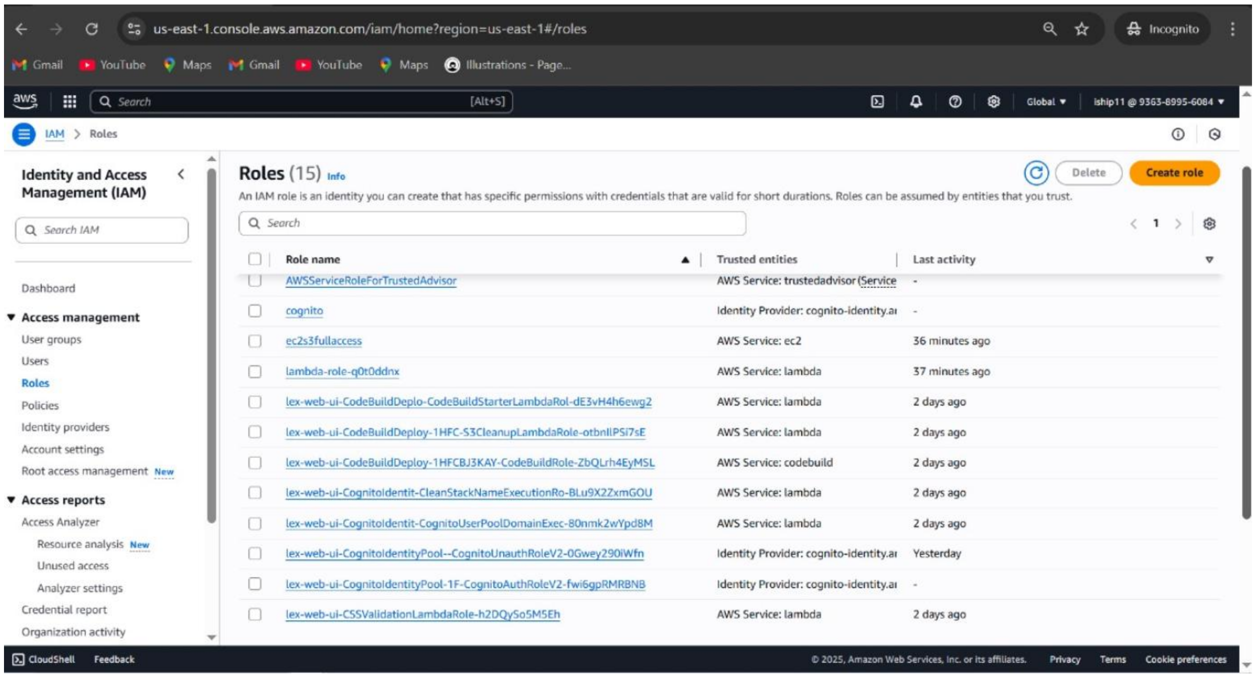
➢ Amazon Cognito Roles

Roles:

- Generated automatically during Cognito identity pool configuration:
  - Authenticated Role: Grants signed-in users controlled access to AWS resources
  - Unauthenticated Role (optional): For guest users with restricted access

Attached Policies:

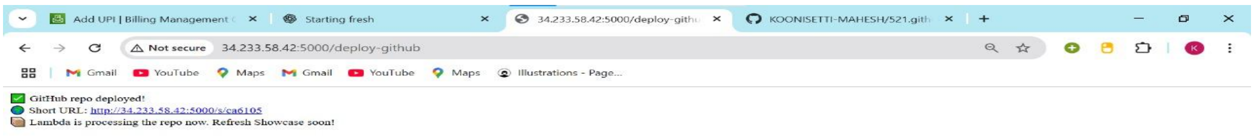
- Custom fine-grained policies, e.g., to allow invoking specific Lambda functions or accessing certain APIs



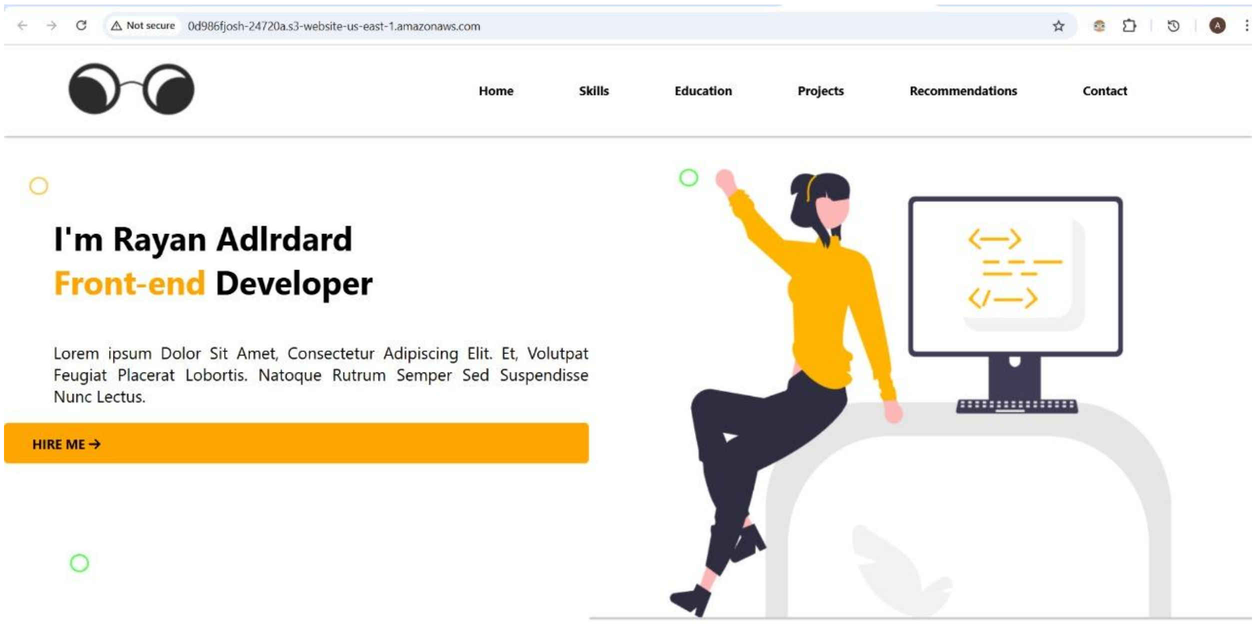
Role name	Trusted entities	Last activity
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
cognito	Identity Provider: cognito-identity.ar	-
ec2s3fullaccess	AWS Service: ec2	36 minutes ago
lambda-role-q0t0ddnx	AWS Service: lambda	37 minutes ago
lex-web-ui-CodeBuildDeplo-CodeBuildStarterLambdaRol-dE3vH4h6ewg2	AWS Service: lambda	2 days ago
lex-web-ui-CodeBuildDeploy-1HFC-S3CleanupLambdaRole-otbnlIPSi7sE	AWS Service: lambda	2 days ago
lex-web-ui-CodeBuildDeploy-1HFCBJ3KAY-CodeBuildRole-ZbQLrh4EyMSL	AWS Service: codebuild	2 days ago
lex-web-ui-CognitoIdentit-CleanStackNameExecutionRo-BLu9X2ZxmGOU	AWS Service: lambda	2 days ago
lex-web-ui-CognitoIdentit-CognitoUserPoolDomainExec-80nmk2wYpd8M	AWS Service: lambda	2 days ago
lex-web-ui-CognitoIdentityPool--CognitoUnauthRoleV2-0Gwey290iWfn	Identity Provider: cognito-identity.ar	Yesterday
lex-web-ui-CognitoIdentityPool-1F-CognitoAuthRoleV2-fwi6gpRMRBNB	Identity Provider: cognito-identity.ar	-
lex-web-ui-CSSValidationLambdaRole-h2DQySo5M5Eh	AWS Service: lambda	2 days ago

Step 8: URL Shortening & Redirection

Each deployment gets a unique short ID (using UUID). The Flask app redirects users from this short URL to the actual hosted site URL.



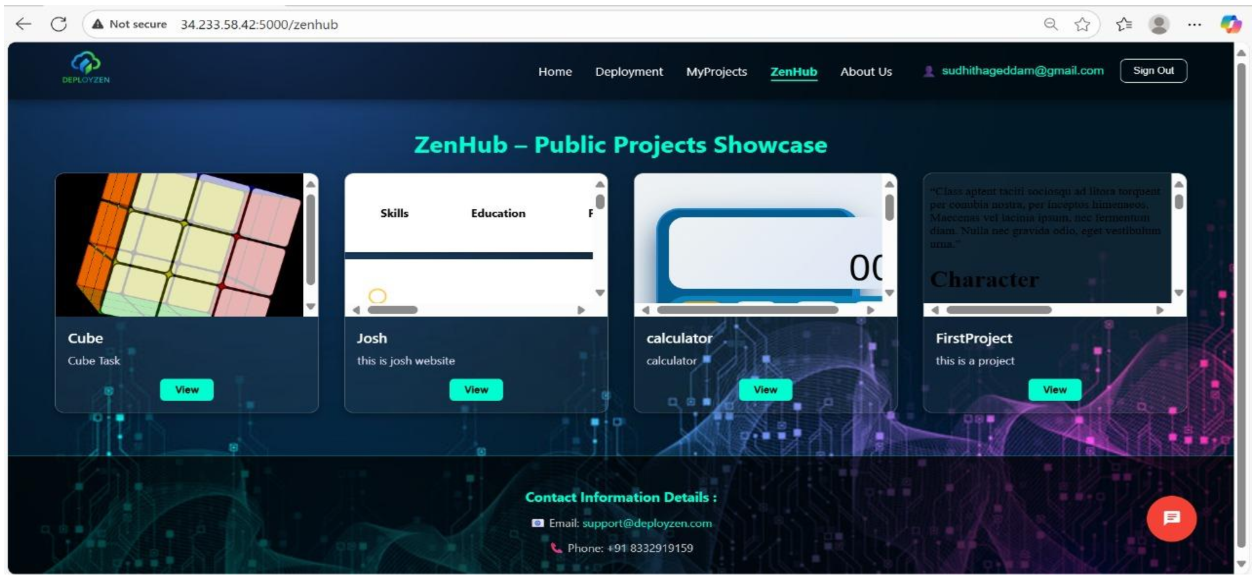
```
✅ GitHub repo deployed!
🌐 Short URL: http://34.233.58.42:5000/s/ca6105
📦 Lambda is processing the repo now. Refresh Showcase soon!
```


Not secure 0d986fjosh-24720a.s3-website-us-east-1.amazonaws.com
Home
Skills
Education
Projects
Recommendations
Contact
I'm Rayan Adlrdard
Front-end Developer
Lorem ipsum Dolor Sit Amet, Consectetur Adipiscing Elit. Et, Volutpat Feugiat Placerat Lobortis. Natoque Rutrum Semper Sed Suspendisse Nunc Lectus.
HIRE ME →

Step 9: Showcase & Update Flow

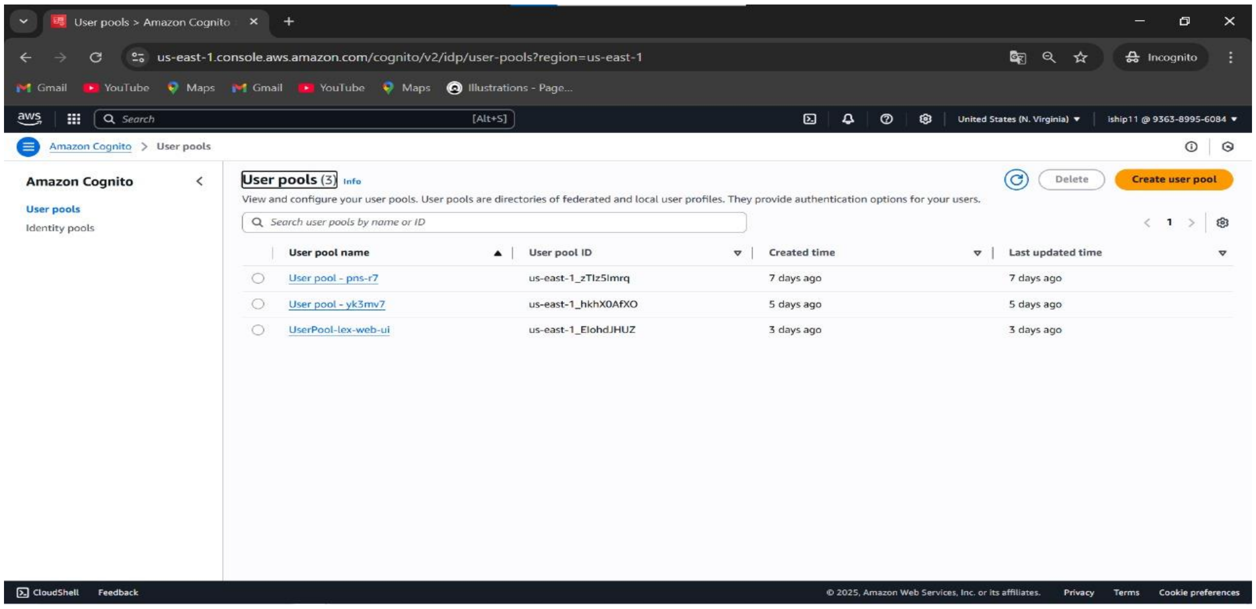
If a project is marked public, it's shown in the ZenHub (Showcase).

Showcase page.



Step 10: User Login with Amazon Cognito

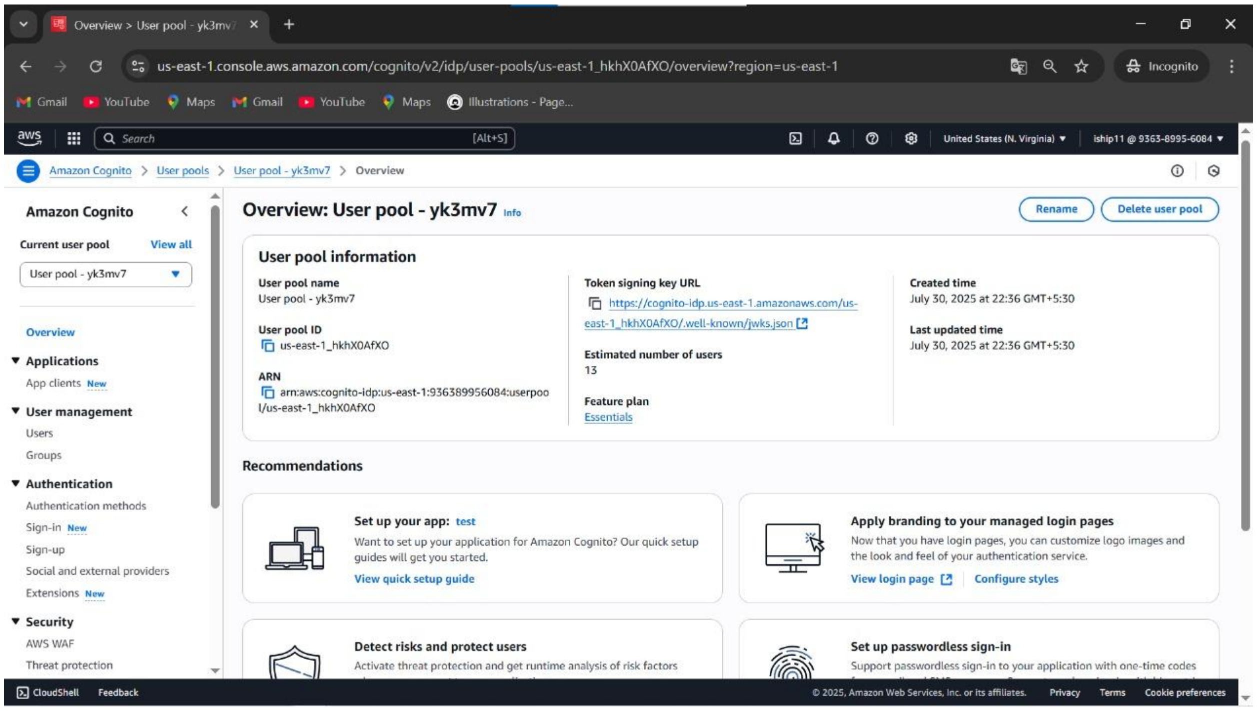
We used Cognito for login/signup and email verification. It securely handles user sessions and token management.


User pools > Amazon Cognito
us-east-1.console.aws.amazon.com/cognito/v2/idp/user-pools?region=us-east-1
Amazon Cognito
User pools
Identity pools
Delete
Create user pool
CloudShell
Feedback

User pools (3) Info

View and configure your user pools. User pools are directories of federated and local user profiles. They provide authentication options for your users.

User pool name	User pool ID	Created time	Last updated time
User pool - pns-r7	us-east-1_zTIz5lmrq	7 days ago	7 days ago
User pool - yk3mv7	us-east-1_hkhX0AfXO	5 days ago	5 days ago
UserPool-lex-web-ui	us-east-1_ElohdJHUZ	3 days ago	3 days ago



Overview: User pool - yk3mv7 Info

User pool information

User pool name
User pool - yk3mv7

User pool ID
us-east-1_hkhX0AfXO

ARN
arn:aws:cognito-idp:us-east-1:936389956084:userpool/us-east-1_hkhX0AfXO

Token signing key URL
https://cognito-idp.us-east-1.amazonaws.com/us-east-1_hkhX0AfXO/.well-known/jwks.json

Estimated number of users
13

Feature plan
Essentials

Created time
July 30, 2025 at 22:36 GMT+5:30

Last updated time
July 30, 2025 at 22:36 GMT+5:30

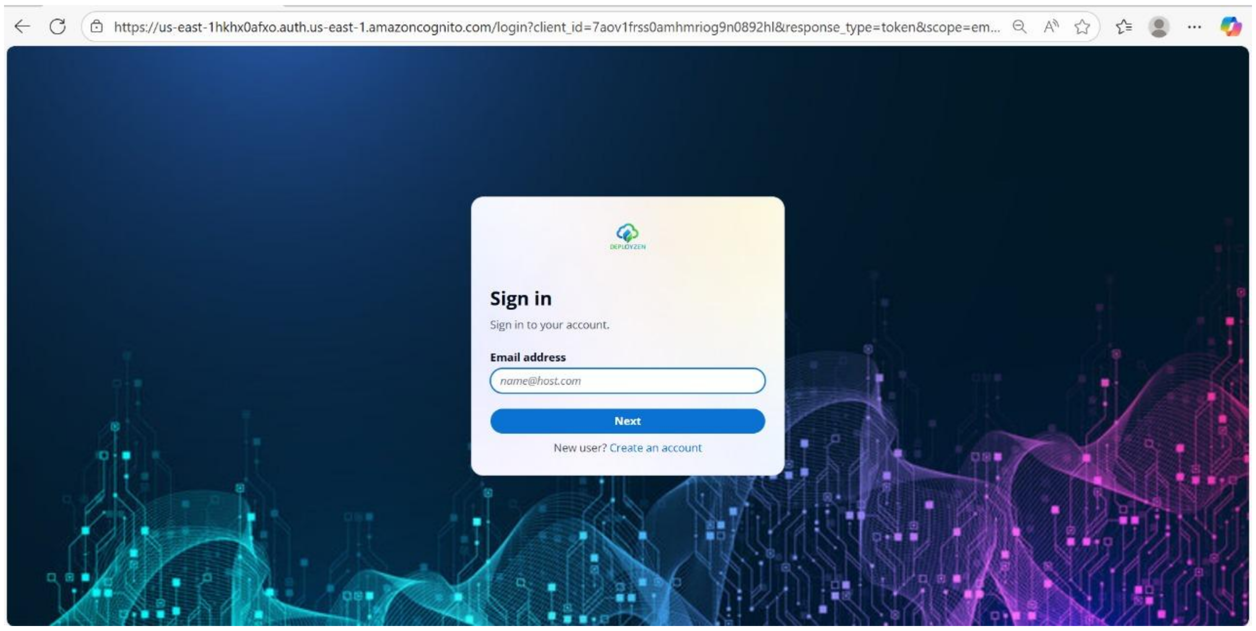
Recommendations

Set up your app: test
Want to set up your application for Amazon Cognito? Our quick setup guides will get you started.
View quick setup guide

Apply branding to your managed login pages
Now that you have login pages, you can customize logo images and the look and feel of your authentication service.
View login page
Configure styles

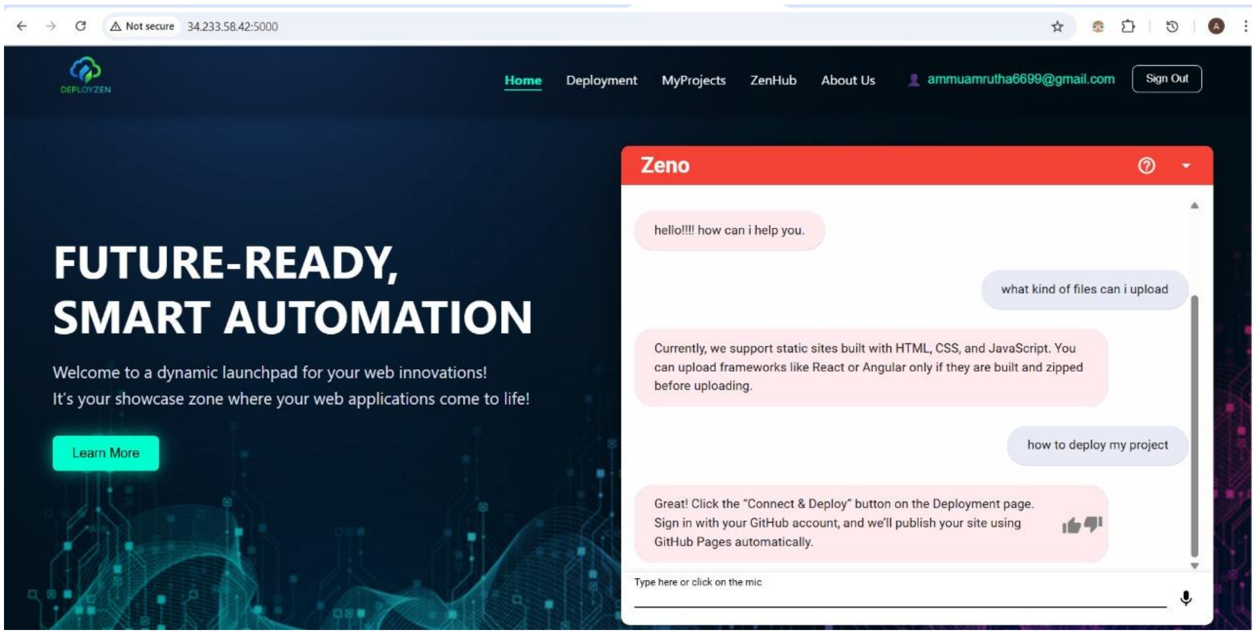
Detect risks and protect users
Activate threat protection and get runtime analysis of risk factors

Set up passwordless sign-in
Support passwordless sign-in to your application with one-time codes


https://us-east-1hkhx0afxo.auth.us-east-1.amazoncognito.com/login?client_id=7aov1frss0amhmriog9n0892hl&response_type=token&scope=em...
DEPLOYZEN
Sign in
Sign in to your account.
Email address
name@host.com
Next
New user? Create an account

Step 11: Chatbot Integration

Amazon Lex chatbot was added to assist users with deployment queries, acting like a smart assistant.


Not secure 34.233.58.42:5000
DEPLOYZEN
Home
Deployment
MyProjects
ZenHub
About Us
ammuamrutha6699@gmail.com
Sign Out
FUTURE-READY,
SMART AUTOMATION
Welcome to a dynamic launchpad for your web innovations!
It's your showcase zone where your web applications come to life!
Learn More
Zeno
Type here or click on the mic

hello!!!! how can i help you.

what kind of files can i upload

Currently, we support static sites built with HTML, CSS, and JavaScript. You can upload frameworks like React or Angular only if they are built and zipped before uploading.

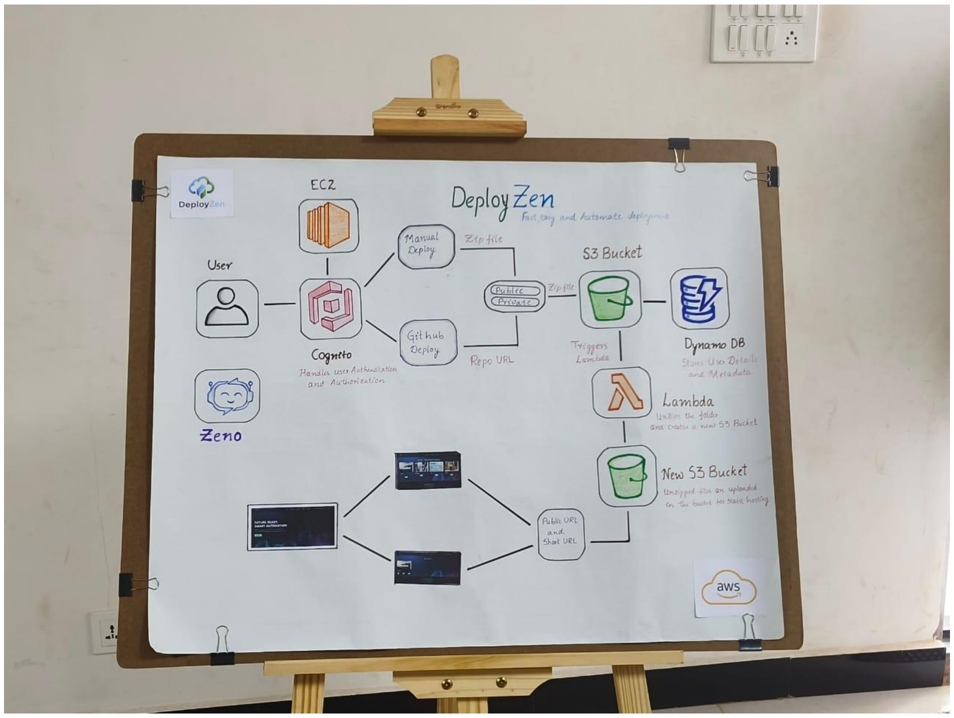
how to deploy my project

Great! Click the "Connect & Deploy" button on the Deployment page. Sign in with your GitHub account, and we'll publish your site using GitHub Pages automatically.

Tech Stack

Technology	Purpose
Amazon EC2	Hosting the Flask backend application.
Amazon S3	Storing uploaded ZIP files and hosting extracted static websites.
AWS Lambda	Automatically triggered to unzip files and deploy content.
Amazon DynamoDB	Used to store metadata like short IDs, URLs, and filenames.
Amazon Cognito	Managing user authentication, signup, and login securely.
Amazon Lex	Providing chatbot support for guiding users through deployment.
Flask (Python)	Backend framework to manage routes, logic, and AWS integration.
Tailwind CSS	Used for building a clean and responsive frontend UI. HTML For structuring the web pages.
UUID (Python)	Generating unique short IDs for URL shortening.

PICTURES


DeployZen
DeployZen
Fast, easy and Automate deployment
EC2
User
Cognito
Handles user Authentication and Authorization
Zeno
Manual Deploy
Zip file
Github Deploy
Repo URL
Public
Private
Zip file
S3 Bucket
Triggers Lambda
Dynamo DB
Stores User Details and Metadata
Lambda
Unzips the folder and creates a new S3 Bucket
New S3 Bucket
Unzipped files are uploaded on the bucket for static hosting
Public URL and Short URL
aws

Conclusion

DeployZen simplifies the process of deploying static websites by offering an easy-to-use interface and automation behind the scenes. By integrating multiple AWS services like EC2, S3, Lambda, Cognito, and DynamoDB, the project achieves seamless deployment using both manual ZIP uploads and GitHub repositories.

The addition of a chatbot makes the platform user-friendly, especially for beginners with minimal technical knowledge. This project not only deepened our understanding of cloud services but also gave us hands-on experience with real-world architecture, automation, and user interaction.

In the future, this platform can be enhanced further by adding analytics, user-based dashboards, and support for more file types or frameworks. Overall, DeployZen represents a powerful and scalable solution for effortless static web hosting.