



PARSHVANATH CHARITABLE TRUST'S
A.P. Shah Institute of Technology
Thane, 400615

Academic Year: 2021-22
Department of Computer Engineering

CSL605: SKILL BASED LAB COURSE: CLOUD COMPUTING

Mini Project Report

- **Title of Project** - APSIT Community
- **Year and Semester** - T.E. (Sem VI)
- **Group Members Name and Roll No.** -
 - Vaidik Vadhavana – 20102197 / 58
 - Abhishek Tiwari – 20102064 / 54

Table of Contents

Sr. No.	Topic	Page No.
1.	Problem Definition	3
2.	Introduction	3
3.	Description (Include the cloud services used in the project, methodologies used and software requirements)	4
4.	Implementation details with screen-shots (stepwise)	5
5.	Learning Outcome	8

Problem Definition:

The APSIT Community application was developed to create a platform where students and faculty members of APSIT (A. P. Shah Institute of Technology) can connect with each other, share their knowledge, and collaborate on projects. The primary goal of the application was to provide a single platform where students can post their queries and get solutions from faculty members or other students. The application also provides a platform for faculty members to post important announcements and notices for students.

The development of the APSIT Community application was not without its challenges. The primary challenge was to host the application on the cloud and ensure that it is scalable, secure, and available to users 24x7. Additionally, the application should be easy to deploy and maintain, even with a small team of developers.

To address these challenges, our team decided to use AWS cloud services. The team chose AWS because it provides a wide range of cloud services that can be used to develop, deploy, and manage applications. Our team also chose AWS because it is a reliable and secure cloud platform that can handle the needs of a large-scale application.

The APSIT Community application was designed using a microservices architecture, which made it easy to scale and maintain individual components of the application. The frontend and backend applications were decoupled, which made it easy to deploy and scale each component independently. The application was containerized using Docker, which made it easy to deploy and manage the application on AWS Beanstalk.

Introduction:

The APSIT Community is a web application designed to facilitate collaboration and knowledge sharing among students and faculty members. It has been developed using Next.js and MUI for the frontend, and Python and Flask for the backend. The application has been deployed on AWS using various cloud services, including Amplify, Beanstalk, S3, and IAM.

The frontend of the application was developed using Next.js, which is a React-based web framework that provides server-side rendering of React components. It is ideal for developing scalable and performant web applications. MUI, or Material-UI, is a popular React UI framework that provides a set of pre-designed components for creating modern and visually appealing user interfaces.

The backend of the application was developed using Python, a high-level programming language that is easy to read and write. Flask, a lightweight web framework for Python, was used to develop the API endpoints for the application. This allows for easy integration with the frontend and provides a seamless user experience.

The application has been deployed on AWS using various cloud services. Amplify was used to deploy the frontend application and to provide continuous deployment by integrating with GitHub. Beanstalk was used to deploy the backend application, which allows for easy scaling and management of the application. S3 was used to store the static content of the website, including images, CSS, and JavaScript files. IAM was used to create different users with different policies attached, providing granular access control for the application.

In conclusion, the APSIT Community is a modern web application developed using Next.js, MUI, Python, and Flask. It has been deployed on AWS using various cloud services, making it scalable, secure, and easy to deploy and maintain. The frontend and backend have been developed using modern and performant technologies, providing a seamless user experience.

Description:

The APSIT Community is a web application developed using Next.js, MUI, Python, and Flask. The application has been deployed on AWS using various cloud services, making it scalable, secure, and easy to deploy and maintain.

Cloud services used in the project:

- **Amplify:** Used to deploy the frontend application and provide continuous deployment by integrating with GitHub. Amplify automatically builds and deploys the application when new code is pushed to the GitHub repository.
- **Beanstalk:** Used to deploy the backend application, which allows for easy scaling and management of the application. Beanstalk automates the deployment of the application, manages the underlying infrastructure, and provides monitoring and logging capabilities.
- **S3:** Used to store the static content of the website, including images, CSS, and JavaScript files. S3 is a highly scalable and durable object storage service that provides low-latency access to data from anywhere on the web.
- **IAM:** Used to create different users with different policies attached, providing granular access control for the application. IAM enables users to manage access to AWS resources and services securely.

Software requirements for the project:

- **Next.js:** A React-based web framework that provides server-side rendering of React components. It is used to develop the frontend of the application.
- **MUI:** A React UI framework that provides a set of pre-designed components for creating modern and visually appealing user interfaces. It is used to style and design the frontend of the application.
- **Python:** A high-level programming language that is easy to read and write. It is used to develop the backend of the application.
- **Flask:** A lightweight web framework for Python that is used to develop the API endpoints for the application. It enables easy integration with the frontend and provides a seamless user experience.

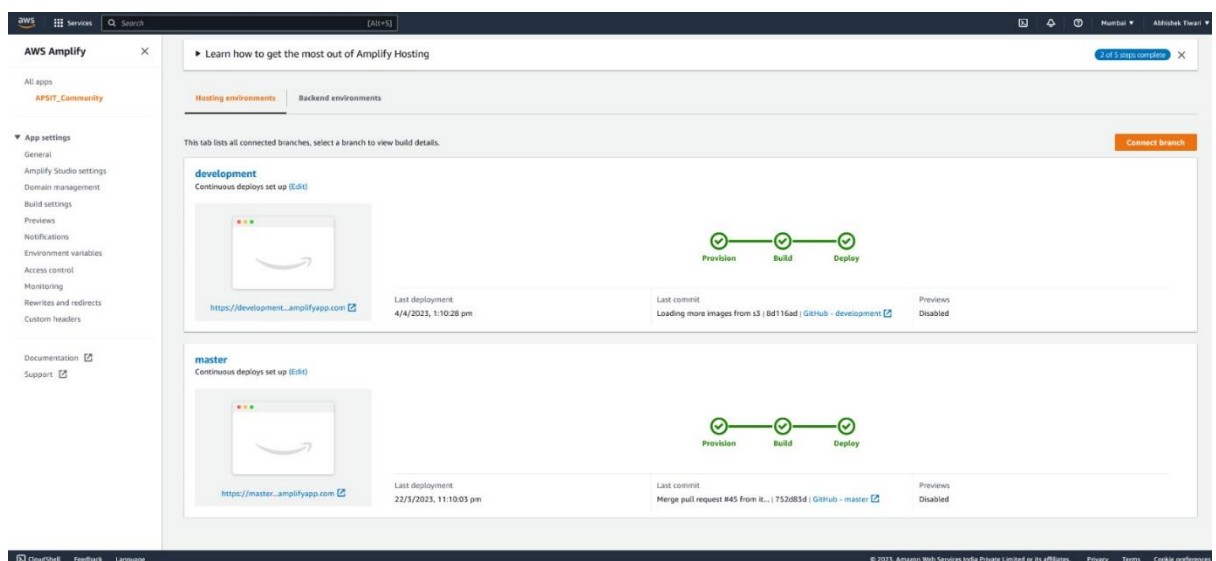
In conclusion, the APSIT Community is a modern web application that has been developed using various cloud services, agile methodology, and modern software technologies. The cloud services used in the project provide scalability, security, and ease of deployment and

maintenance. The agile methodology enabled the team to deliver high-quality software in a timely and efficient manner. Finally, the modern software technologies used in the project provide a seamless user experience and enable easy integration between the frontend and backend of the application.

Implementation details:

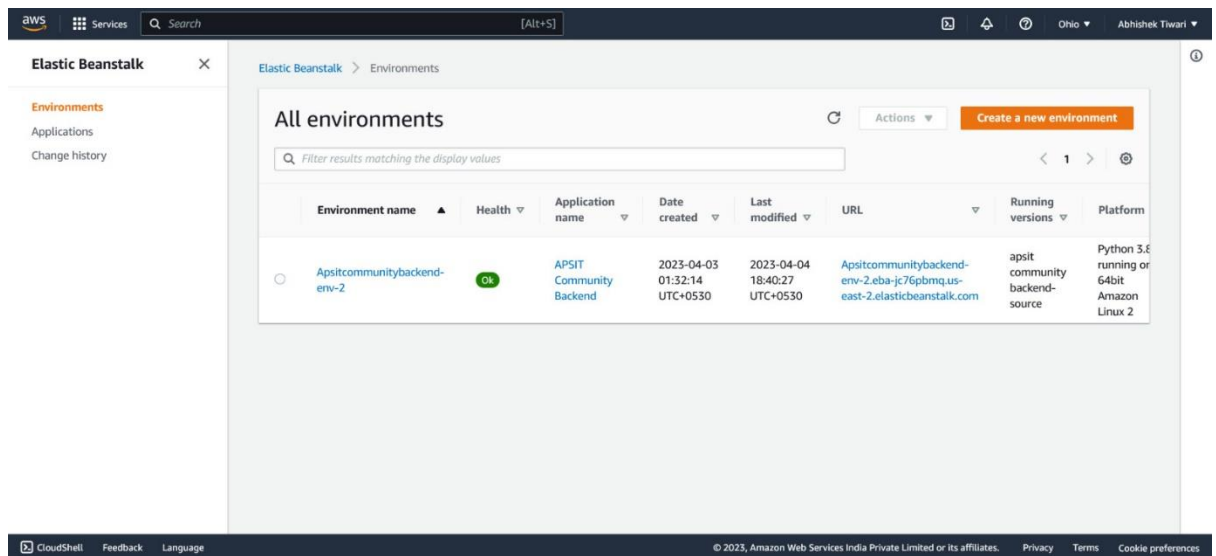
The APSIT Community web application has been deployed on AWS using various cloud services. The following steps outline how each service was implemented:

1. Amplify:



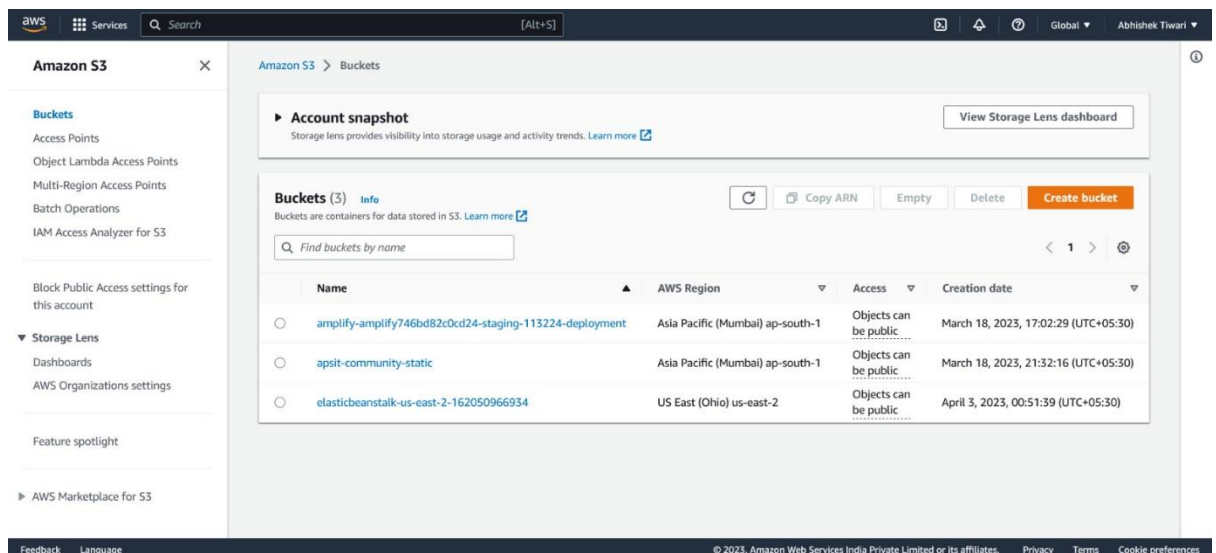
- Create an Amplify app from the Amplify console in the AWS Management Console.
- Connect the app to the GitHub repository where the frontend code is stored.
- Configure the build settings and environment variables in the Amplify console.
- Enable continuous deployment to automatically build and deploy the application when new code is pushed to the repository.

2. Beanstalk:



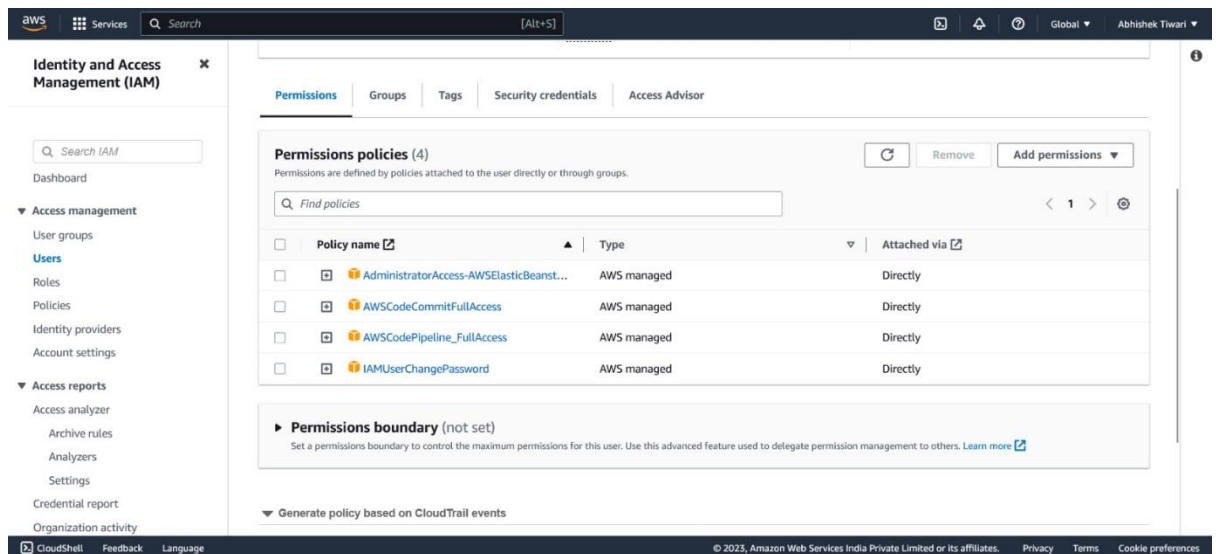
- Create an Elastic Beanstalk environment from the Elastic Beanstalk console in the AWS Management Console.
- Select the Python platform and configure the environment with the required resources and settings.
- Upload the backend code to the environment and deploy it.

3. S3:



- Create an S3 bucket from the S3 console in the AWS Management Console.
- Upload the static content of the website, including images, CSS, and JavaScript files, to the bucket.
- Configure the bucket settings to allow public access to the content.

4. IAM:



- Create an IAM user from the IAM console in the AWS Management Console.
- Attach policies to the user that define the user's permissions and access to AWS resources.
- Create an IAM group and add the user to the group to manage permissions for a group of users.

Overall, these steps allow for the deployment of the APSIT Community web application on AWS using various cloud services. By following these steps, the application can be scaled, secured, and easily maintained. The Amplify service provides continuous deployment and integration with GitHub, while Elastic Beanstalk allows for easy scaling and management of the backend. S3 provides a scalable and durable storage solution for the static content of the website, and IAM enables granular access control for users and groups.

Learning Outcomes:

The deployment of the APSIT Community web application on AWS using various cloud services has provided several learning outcomes. Some of these include:

1. Understanding the different cloud services available in AWS: The project has helped us gain a deeper understanding of the various cloud services offered by AWS and their capabilities. This has enabled us to choose the appropriate services for our application's specific requirements and needs.
2. Learning to deploy an application in the cloud: The deployment of the APSIT Community web application has provided us with hands-on experience in deploying an application on AWS. We have learned the importance of scalability, reliability, and security when deploying an application in the cloud.
3. Knowledge of DevOps practices: The use of continuous deployment and integration through Amplify has exposed us to DevOps practices. We have learned how DevOps

practices can improve application development and deployment processes by enabling faster, more reliable, and more secure deployments.

4. Familiarity with AWS console and CLI: Deploying the APSIT Community web application on AWS has made us more familiar with the AWS Management Console and CLI. This has enabled us to understand how to navigate and manage AWS resources effectively.

Overall, the deployment of the APSIT Community web application on AWS using various cloud services has provided valuable insights and learning outcomes. These will be useful in future application development and deployment projects, and in our general understanding of cloud computing and its related services.