zapcom.

INTERNSHIP REPORT

KASHINATH BORKER

2009

Zapcom Solutions Pvt Ltd.

Goa University

REPORT OF INTERNSHIP DONE AT ZAPCOM SOLUTIONS PVT LTD

SUBMITTED BY

KASHINATH BORKER

2009

UNDER THE GUIDANCE OF

Mr. Suburamanian A
(Delivery Manager, Zapcom)

Mr. Nagarjuna Reddy
(Tech Lead, Zapcom)

Mr. Tarun Rapeta
(Software Engineer, Zapcom)

Internship Letter

zapcom.

1st June 2023

TO WHOMSOEVER IT MAY CONCERN

This is to inform you that **Mr. Kashinath Borker,** student of Master of Computer Applications (MCA) of Goa University, Goa, is currently undergoing his final semester project (Semester VI/V) at our company, **Zapcom Solutions Pvt. Ltd** from 4th January, 2023.

During his tenure he has met the expectations of his team lead/mentor/guide and found to be regular and sincere.

This letter is being issued on his request to be submitted with the project report at Goa University.

The final internship completion letter will be provided on completing his internship.

For Zapcom Solutions Pvt. Ltd.

Zapcom Solutions Pvt. Ltd. INDIA

Srinivas Reddy Kothakota
Chief Operating Officer

INDIA
Zapcom Solutions Pvt. Ltd
9th Floor, Gamma Tower, Sigma Soft Tech Park,
Whitefield, Bangalore - 560066
Ph: +91-80-67232300

www.zapcg.com

USA
Zapcom Group Inc.
105 Decker Court, Ste. 810 Irving,
TX 75062.
Ph: (972)441-2081

GOA UNIVERSITY

GOA BUSINESS SCHOOL

CERTIFICATE OF EVALUATION

This is to certify that **Mr. Kashinath G. Borker** has been evaluated for the project work titled "**Report of Internship done at Zapcom Solutions Pvt Ltd.**" undertaken at **Zapcom Solutions Pvt Ltd. , Bangalore** in partial fulfillment for the award of the degree in Master of Computer Application.

Examiner 1

Examiner 2
[Kiran Kulkarni]

Place: Goa University

Date: 11th June 2022

Dean, Goa Business School

Acknowledgement

I would like to start by expressing my sincere appreciation to my family, friends, and divine blessings for their unwavering support and strength throughout my internship experience.I am immensely thankful to Mr. Kishore Pallamereddy, the Executive Chairman & CEO of Zapcom, and Zapcom Solutions Pvt Ltd., for granting me the opportunity to intern at Zapcom and gain valuable insights into the world of Information Technology.

I extend my utmost gratitude to Mr. Suburamanian A, the Delivery Manager at Zapcom, and Mr. Tarun Rapeta, a Software Engineer, for their mentorship and unwavering guidance throughout my internship, which proved instrumental in my personal and professional growth.I would also like to extend my appreciation to the HR and Accounts teams at Zapcom for their invaluable assistance in helping me understand the company's policies and providing support in various ways.Furthermore, I am deeply grateful to Mr. Nagarjuna Reddy, the Tech Lead at Zapcom, for his invaluable help and guidance in the technologies I worked on during my internship.I would like to express my heartfelt thanks to Mr. Ramdas Karmali, a Professor of MCA at Goa Business School, Goa University, and Mr. Hanumant Redkar, an Assistant Professor and TPO, MCA at Goa Business School, Goa University, as well as the entire faculty and non-teaching staff of the MCA department at Goa University, for their constant encouragement and support throughout my academic journey.

I would also like to acknowledge the unwavering support of my family and friends, whose care and encouragement played a vital role during my internship. I am especially grateful to my friends in Bangalore, who made me feel at home in a new city.Finally, I would like to express my deep appreciation to the Zapcom family, whose support and guidance have been invaluable in shaping me into a more confident individual, ready to face any challenges that come my way.

Kashinath Borker

INTERNSHIP REPORT

KASHINATH BORKER

2009

Zapcom Solutions Pvt Ltd.

Goa University

REPORT OF INTERNSHIP DONE AT ZAPCOM SOLUTIONS PVT LTD

SUBMITTED BY

KASHINATH BORKER

2009

UNDER THE GUIDANCE OF

Mr. Suburamanian A

(Delivery Manager, Zapcom)

Mr. Nagarjuna Reddy

(Tech Lead, Zapcom)

Mr. Tarun Rapeta

(Software Engineer, Zapcom)

Internship Letter

zapcom.

1st June 2023

TO WHOMSOEVER IT MAY CONCERN

This is to inform you that **Mr. Kashinath Borker,** student of Master of Computer Applications (MCA) of Goa University, Goa, is currently undergoing his final semester project (Semester VI/V) at our company, **Zapcom Solutions Pvt. Ltd** from 4th January, 2023.

During his tenure he has met the expectations of his team lead/mentor/guide and found to be regular and sincere.

This letter is being issued on his request to be submitted with the project report at Goa University.

The final internship completion letter will be provided on completing his internship.

For Zapcom Solutions Pvt. Ltd.

Zapcom Solutions Pvt. Ltd. INDIA

Srinivas Reddy Kothakota
Chief Operating Officer

INDIA
Zapcom Solutions Pvt. Ltd
9th Floor, Gamma Tower, Sigma Soft Tech Park,
Whitefield, Bangalore - 560066
Ph: +91-80-67232300

www.zapcg.com

USA
Zapcom Group Inc.
105 Decker Court, Ste. 810 Irving,
TX 75062.
Ph: (972)441-2081

GOA UNIVERSITY

GOA BUSINESS SCHOOL

CERTIFICATE OF EVALUATION

This is to certify that **Mr. Kashinath G. Borker** has been evaluated for the project work titled "**Report of Internship done at Zapcom Solutions Pvt Ltd.**" undertaken at **Zapcom Solutions Pvt Ltd. , Bangalore** in partial fulfillment for the award of the degree in Master of Computer Application.

Examiner 1

Examiner 2

Place: Goa University

Date: 11th June 2022

Dean, Goa Business School

Acknowledgement

I would like to start by expressing my sincere appreciation to my family, friends, and divine blessings for their unwavering support and strength throughout my internship experience.I am immensely thankful to Mr. Kishore Pallamereddy, the Executive Chairman & CEO of Zapcom, and Zapcom Solutions Pvt Ltd., for granting me the opportunity to intern at Zapcom and gain valuable insights into the world of Information Technology.

I extend my utmost gratitude to Mr. Suburamanian A, the Delivery Manager at Zapcom, and Mr. Tarun Rapeta, a Software Engineer, for their mentorship and unwavering guidance throughout my internship, which proved instrumental in my personal and professional growth.I would also like to extend my appreciation to the HR and Accounts teams at Zapcom for their invaluable assistance in helping me understand the company's policies and providing support in various ways.Furthermore, I am deeply grateful to Mr. Nagarjuna Reddy, the Tech Lead at Zapcom, for his invaluable help and guidance in the technologies I worked on during my internship.I would like to express my heartfelt thanks to Mr. Ramdas Karmali, a Professor of MCA at Goa Business School, Goa University, and Mr. Hanumant Redkar, an Assistant Professor and TPO, MCA at Goa Business School, Goa University, as well as the entire faculty and non-teaching staff of the MCA department at Goa University, for their constant encouragement and support throughout my academic journey.

I would also like to acknowledge the unwavering support of my family and friends, whose care and encouragement played a vital role during my internship. I am especially grateful to my friends in Bangalore, who made me feel at home in a new city.Finally, I would like to express my deep appreciation to the Zapcom family, whose support and guidance have been invaluable in shaping me into a more confident individual, ready to face any challenges that come my way.

Kashinath Borker

Table of Contents

Introduction

This report serves as a brief overview of my full-time on-site internship experience at Zapcom in Bangalore. I commenced my internship journey on January 4th, 2023, and have been actively involved with Zapcom since then. Within this report, I aim to provide essential information about the organization, including the proof of concept (POC) projects I have worked on, as well as tasks assigned to develop Zapcom's intellectual property (IP). Additionally, I will outline other responsibilities I undertook and online courses I completed during the internship period.

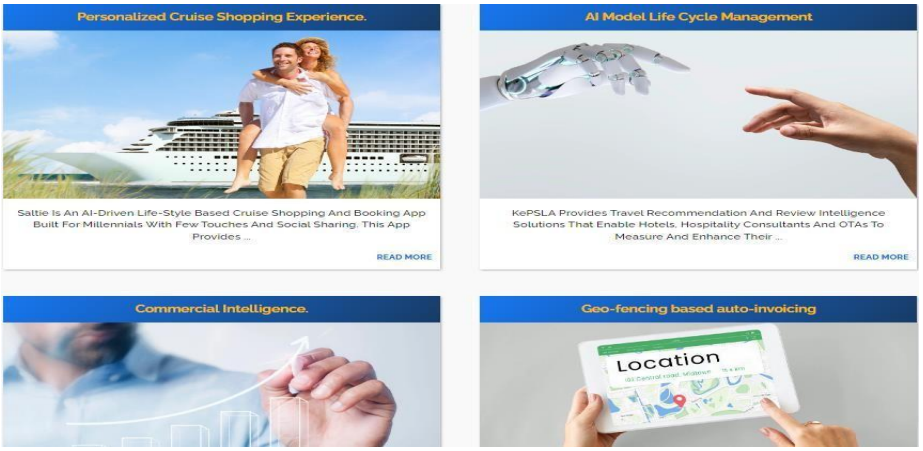
The subsequent content will cover details about the company, work environment, organizational culture, and more. Furthermore, I will delve into specific POCs I contributed to and highlight the tasks I successfully accomplished. This report will emphasize the knowledge I acquired through the completion of assigned projects.

Moreover, I will discuss the tools and technologies employed during my internship and provide a timeline of my activities and progress. Finally, I will conclude by sharing my overall experience and how this opportunity facilitated personal and professional growth.

Throughout this report, I will strive to enhance grammar, correct any spelling errors, and present the information in the most effective and concise manner possible.

Company Profile

ZapCom is a company that focuses on developing and providing products. As an ISO 9001 and 27001 certified company, it upholds high standards in quality management and information security. With a leadership team possessing extensive experience of over 120 years in the Travel and Hospitality domain, ZapCom excels in this industry. The company actively supports and nurtures Start-Up ideas, particularly in the fields of Travel, Hospitality, and Retail.



Zapcom is dedicated to developing digital solutions and platforms that have a profound impact on revenue and cost effectiveness. They specialize in constructing, managing, and enhancing technology for their clients, utilizing a data-driven methodology to design products, platforms, and teams that generate exceptional user experiences and quantifiable business benefits.
Zapcom's work is guided by a distinctive set of principles, which include collaborative solution development, adaptability, prioritizing security, a metrics-oriented approach, automation, a product-centric mindset, and personalized project teams.Through their collaborative approach, Zapcom engages closely with clients to co-create tailored solutions that address their specific needs. They emphasize adaptability, ensuring that their technology solutions can scale and adjust seamlessly to accommodate changing requirements. Zapcom places a strong emphasis on security, prioritizing the protection of data and systems. They utilize a metrics-driven approach to measure performance and make data-informed decisions. Automation plays a vital role in their processes, enabling efficient and consistent execution of tasks. Zapcom adopts a product-centric mindset, considering the end-user experience and overall value delivered. Finally, they form customized project teams, known as pods, to provide personalized attention and expertise to each client engagement.

Certification Courses and Self Study Completed During Internship

Udemy And LinkedIn Learning

- Git Essential Training: The Basics
- Jenkins Essential Training
- Amazon EC2 Deep Dive
- AWS for Developers: Identity Access Management (IAM)
- Learning Groovy
- Kodekloud Udemy Labs - Online Kubernetes Lab for Beginners - Hands On
- Udemy Labs - Certified Kubernetes Administrator with Practice Tests
- Helm Kubernetes Packaging Manager for Developers and DevOps
- Docker for the Absolute Beginner - Hands On - DevOps
- HashiCorp Certified: Terraform Associate 2023
- AWS VPC and Networking in depth: Learn practically in 8 hr
- Jenkins, From Zero To Hero: Become a DevOps Jenkins Master

Tools and Technologies Used

Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.

Amazon Elastic Compute Cloud is a part of Amazon.com's cloud-computing platform, Amazon Web Services, that allows users to rent virtual computers on which to run their own computer applications.

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS and on-premises.

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as cloud-native applications, and mobile apps.

Amazon Virtual Private Cloud (Amazon VPC) gives you full control over your virtual networking environment, including resource placement, connectivity, and security.

Amazon Relational Database Service (RDS) is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud.

An **AWS Identity and Access Management** (IAM) user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS. A user in AWS consists of a name and credentials.

The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

Terraform is an open-source infrastructure as code software tool created by HashiCorp. Users define and provide data center infrastructure using a declarative configuration language known as HashiCorp Configuration Language

Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers

Docker Desktop delivers the speed, choice and security you need for designing and delivering these containerized applications on your desktop.

Minikube is local Kubernetes, focusing on making it easy to learn and develop for Kubernetes.

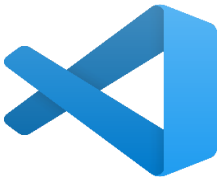
Jenkins is an open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery.

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.

GitLab Inc. is the open-core company that provides GitLab, the DevOps software that combines the ability to develop, secure, and operate software in a single application.

Oracle VM VirtualBox is a type-2 hypervisor for x86 virtualization developed by Oracle Corporation.



Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS.

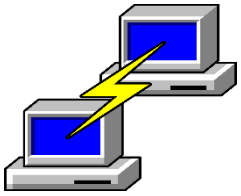
Helm Charts are simply Kubernetes YAML manifests combined into a single package that can be advertised to your Kubernetes clusters

Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

Azure DevOps Server is a Microsoft product that provides version control, reporting, requirements management, project management, automated builds, testing and release management capabilities. It covers the entire application lifecycle, and enables DevOps capabilities.

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells

Windows Subsystem for Linux (WSL) is a compatibility layer for running Linux binary executables natively on Windows 10, Windows 11 etc.



PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. Management Console

Groovy is a powerful and dynamic programming language that runs on the Java Virtual Machine (JVM). It is designed to be a companion language for Java, providing a more expressive and concise syntax while still maintaining seamless compatibility with existing Java code and libraries.e

Maven is a widely used build automation and dependency management tool primarily used for Java projects. It provides a structured and standardized way to manage project dependencies, build processes, and project documentation.

YAML (YAML Ain't Markup Language) is a human-readable data serialization language. It is often used for configuration files, data exchange between languages, and representing structured data. YAML's simple and intuitive syntax makes it popular for various applications.

PROJECT 1

Title: Continuous Web Deployer: Automating CI/CD Pipeline for Efficient Website Deployment

Introduction:
In my first project, I successfully developed a Continuous Web Deployer, utilizing a comprehensive CI/CD pipeline to streamline and automate the deployment process of a simple website. This project involved integrating various tools and technologies commonly associated with DevOps practices, enabling efficient version control, containerization, automation, code quality analysis, artifact management, and build automation using Maven. By implementing GitHub webhooks, the deployment process became seamlessly triggered whenever changes were pushed to the repository.

Tools and Technologies Utilized:

1. Version Control: GitHub
2. Containerization: Docker
3. Automation and Orchestration: Jenkins
4. Code Quality Analysis: SonarQube
5. Artifact Management: Nexus Repository
6. Build Automation: Maven

GitHub for Version Control:
GitHub served as the central repository for hosting and version control of the website's source code. It provided a collaborative environment for developers, enabling them to efficiently manage code changes, track revisions, and facilitate team collaboration.

Docker for Containerization:
Docker played a crucial role in containerizing the application and its dependencies. By creating lightweight and isolated containers, we achieved consistent deployment across different environments, ensuring that the application ran consistently regardless of the host system's configuration.

Jenkins for Automation and Orchestration:
Jenkins, a powerful automation tool, was used to orchestrate the entire CI/CD pipeline. It enabled the automation of repetitive tasks, such as building, testing, and deploying the website, while also integrating with other tools in the pipeline.

SonarQube for Code Quality Analysis:
SonarQube was integrated into the pipeline to analyze the quality of the codebase. It performed static code analysis, identifying code smells, bugs, vulnerabilities, and enforcing coding best practices. This allowed us to maintain a high level of code quality throughout the development process.

Nexus Repository for Artifact Management:
Nexus Repository served as a central repository for storing and managing artifacts generated during the CI/CD pipeline. It facilitated artifact versioning, artifact retrieval, and ensured reliable and efficient artifact management.

GitHub Webhooks for Automation Trigger:
By leveraging GitHub webhooks, the CI/CD pipeline was triggered automatically whenever new changes were pushed to the GitHub repository. This real-time integration reduced manual intervention, allowing for

quicker and more frequent deployments.

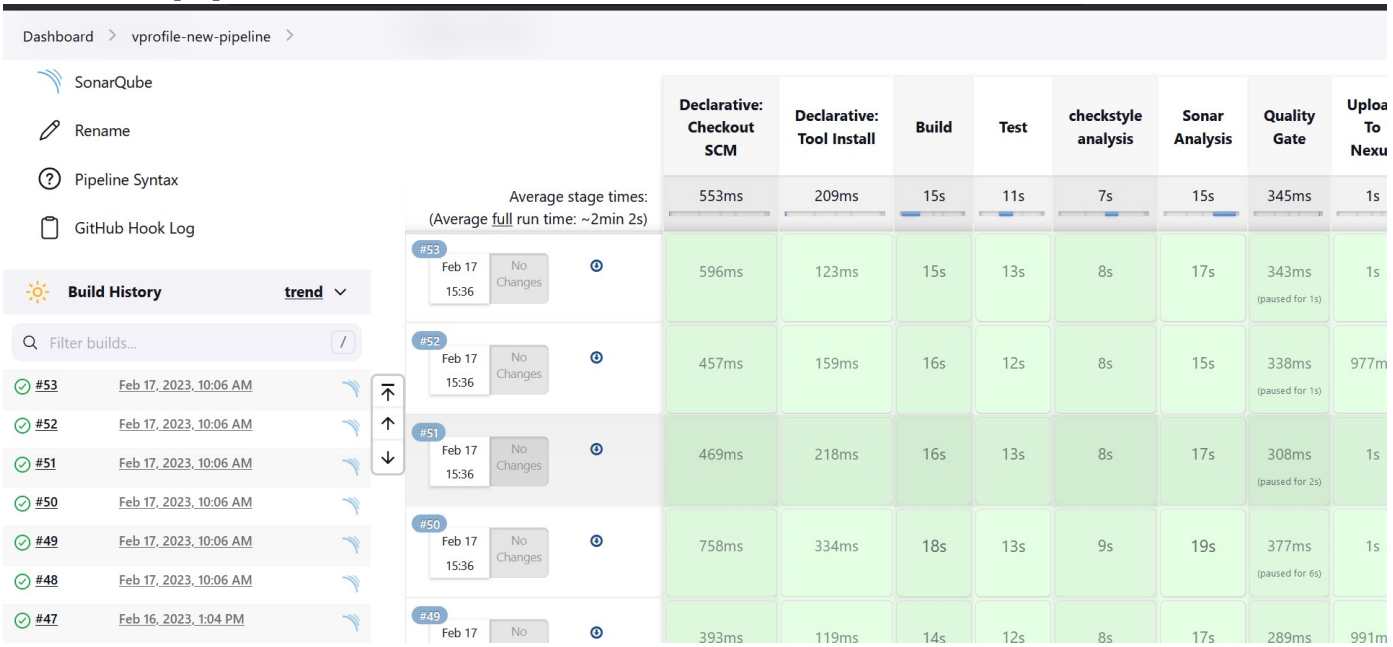
Build Automation with Maven:

Maven, a build automation tool, was utilized to manage the project's build process. It handled dependency management, project structuring, and build configuration, simplifying the build and deployment process. Maven allowed for efficient building of artifacts, ensuring a consistent and reliable build across different environments.

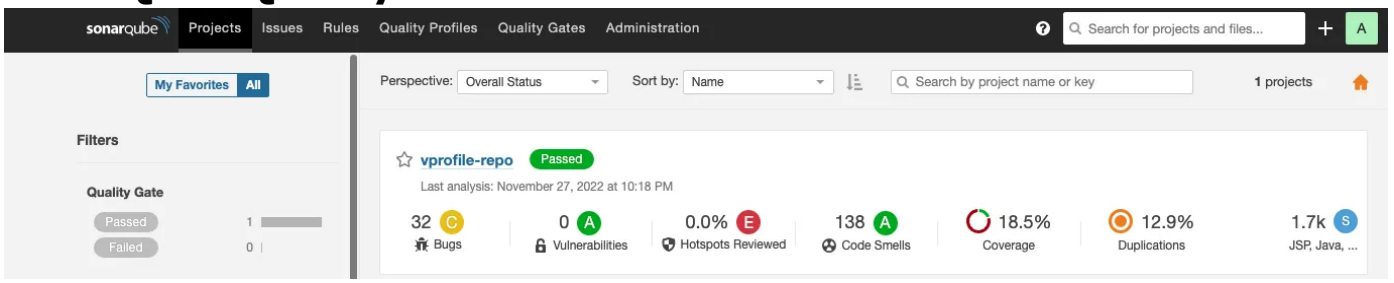
Conclusion:

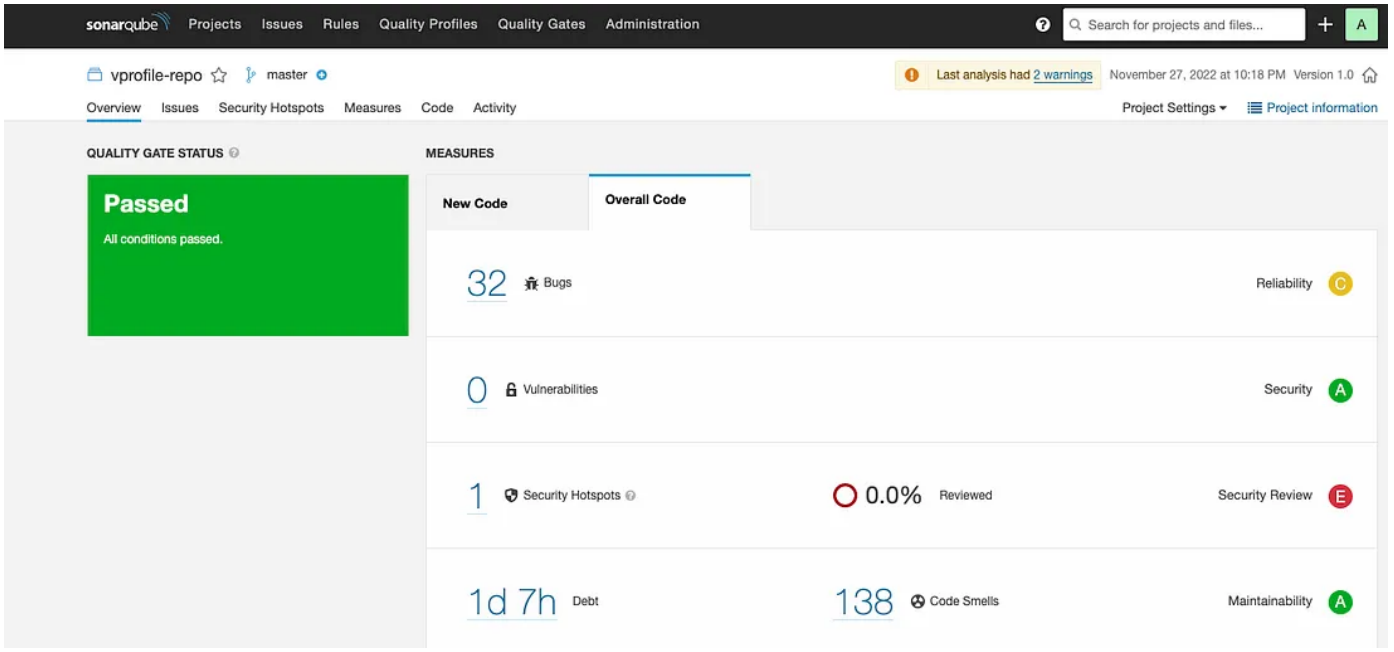
The Continuous Web Deployer project showcased my ability to design and implement an end-to-end CI/CD pipeline using various DevOps tools and technologies, including GitHub, Docker, Jenkins, SonarQube, Nexus Repository, and Maven. By incorporating version control, containerization, automation, code quality analysis, artifact management, and build automation, I successfully achieved efficient and consistent deployment of a simple website. The integration of GitHub webhooks further streamlined the process, ensuring automatic triggers and reducing manual intervention. This project served as a solid foundation for my understanding and practical application of DevOps principles in real-world scenarios, empowering me to deliver projects with improved efficiency and reliability.

Jenkins pipeline



SonarQube Quality Gate Results





PROJECT 2

Title: Scalable Infrastructure Deployment using Terraform: Harnessing AWS Cloud Services

Introduction:
In my second project, I successfully developed a scalable infrastructure deployment solution using Terraform. This project focused on designing and implementing infrastructure on the AWS cloud platform, leveraging a range of services such as EC2, VPC, subnets, routing tables, security groups, IGW, and NAT. I employed modularization techniques to enhance the efficiency, reusability, and scalability of the Terraform codebase, allowing for seamless management of infrastructure resources.

Tools and Technologies Utilized:

1. Infrastructure as Code: Terraform
2. Cloud Platform: Amazon Web Services (AWS)

Designing Infrastructure with AWS Cloud Services:
Using Terraform, I designed and deployed infrastructure on the AWS cloud platform. This included creating Amazon EC2 instances for virtual servers, Virtual Private Cloud (VPC) for network isolation, subnets for segmenting resources, routing tables for network traffic management, security groups for access control, Internet Gateway (IGW) for connecting VPC to the internet, and Network Address Translation (NAT) for outbound internet traffic.

Modularization for Efficiency, Reusability, and Scalability:
To ensure efficient and manageable infrastructure deployment, I modularized the Terraform code. By breaking down the infrastructure into reusable modules, I achieved improved maintainability, ease of collaboration, and scalability. Each module represented a specific component or resource, such as VPC, subnet, or EC2 instance, allowing for easy composition and configuration of the infrastructure.

Terraform's Infrastructure as Code:
By adopting Infrastructure as Code (IaC) principles with Terraform, I eliminated manual infrastructure provisioning and enabled version control. Infrastructure configurations were expressed as code, which could be reviewed, tracked, and modified through code repositories. This approach ensured consistency, repeatability, and the ability to recreate the infrastructure reliably in different environments.

Benefits of Scalable Infrastructure Deployment:
The scalable infrastructure deployment solution offered numerous advantages, including:

1. **Flexibility**: With Terraform and AWS cloud services, the infrastructure could be easily adapted and modified to meet evolving business requirements.

2. **Scalability**: The modular design allowed for effortless scaling of resources, enabling rapid growth and capacity management.

3. **Consistency**: Infrastructure provisioning was automated and consistent across environments, reducing the risk of configuration errors and ensuring reliable deployments.

4. **Cost Optimization**: By leveraging AWS cloud services and utilizing only the necessary resources, cost optimization was achieved, minimizing infrastructure expenses.

Conclusion:
The scalable infrastructure deployment project showcased my proficiency in designing and implementing infrastructure using Terraform and AWS cloud services. By utilizing Infrastructure as Code principles, modularization, and AWS cloud capabilities, I delivered an efficient, reusable, and scalable infrastructure solution. The project's success demonstrates my ability to leverage Terraform and AWS to deploy and manage infrastructure in a consistent, scalable, and cost-effective manner.

PROJECT 3

Title: Automated Deployment of React Web Application: Streamlining with DevOps and Cutting-Edge

TechnologiesIntroduction: In my third project, I successfully leveraged DevOps practices to automate the deployment of a React-based web application using cutting-edge technologies. This project focused on containerization, infrastructure automation, and seamless deployment. By employing Docker for containerization, Jenkins for automation, and modularized Terraform code for infrastructure provisioning, I achieved an efficient, scalable, and automated deployment process.

Tools and Technologies Utilized:

1. Containerization: Docker
2. Automation: Jenkins
3. Infrastructure Provisioning: Terraform
4. Cloud Platform: Amazon Web Services (AWS)
5. Front-End Framework: React

Containerization with Docker: I containerized the React web application using Docker, allowing for consistent and efficient deployment across different environments. Docker enabled encapsulating the application and its dependencies into lightweight, isolated containers, ensuring reliable execution regardless of the host system's configuration. The containerized application was pushed to Docker Hub for streamlined distribution and easy access.

Automated Deployment with Jenkins: To automate the deployment process, I created a Jenkins pipeline on an EC2 instance. The pipeline utilized Jenkins to orchestrate the entire deployment workflow, ensuring seamless execution and minimizing manual intervention. With a single click of a button, the pipeline was triggered, initiating the deployment process.

Infrastructure Provisioning with Terraform: The deployment pipeline fetched my modularized Terraform code from GitHub, allowing for efficient infrastructure provisioning. Using Terraform, I created an infrastructure stack incorporating various AWS cloud services such as EC2, VPC, subnets, routing tables, security groups, IGW, and NAT. The modular design facilitated easy configuration and scalability of the infrastructure components, enabling efficient resource management.

Fast and Efficient Deployment: The deployment pipeline utilized Docker to pull and install the containerized image of the React web application. This streamlined approach significantly reduced deployment time and minimized potential errors. The pipeline exposed the application on port 3000, providing seamless access to the end-user.

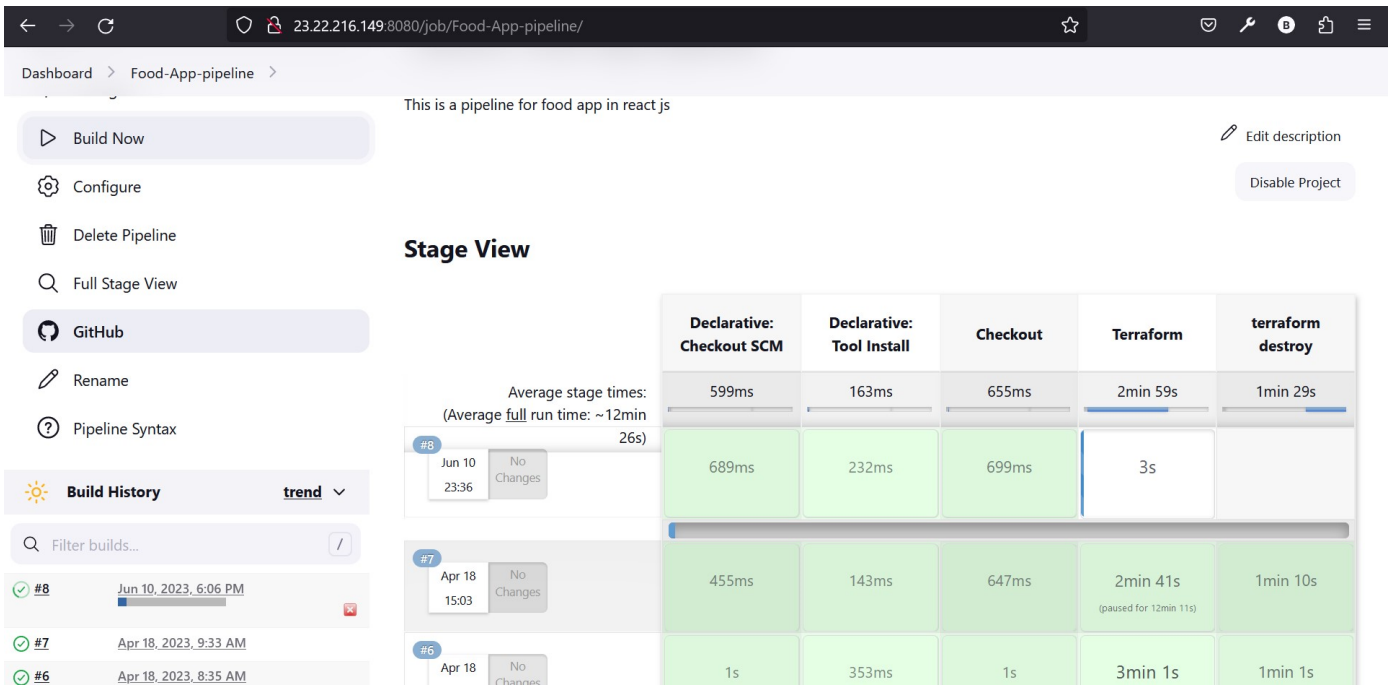
Benefits of DevOps Practices: By incorporating DevOps practices and cutting-edge technologies, the project offered several advantages, including:

1. **Automation**: The automated deployment pipeline reduced manual intervention, minimizing errors and enabling frequent and consistent deployments.
2. **Scalability**: The modularized Terraform code allowed for easy scalability of the infrastructure, accommodating growing user demands.
3. **Efficiency**: Containerization with Docker and streamlined deployment reduced deployment time, enhancing overall efficiency.
4. I**nfrastructure as Code**: Using Terraform, the infrastructure was managed as code, ensuring version control, consistency, and repeatability across different environments.

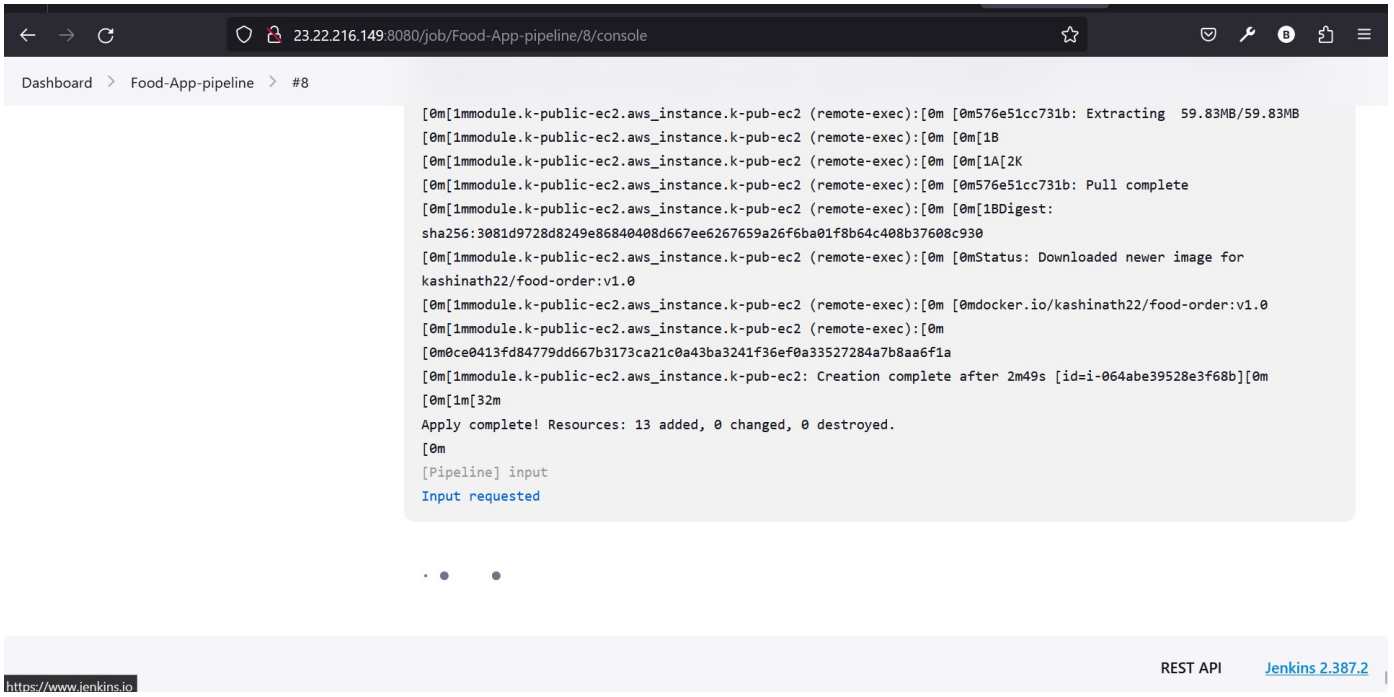
Conclusion: The third project demonstrated my ability to leverage DevOps practices and cutting-edge technologies to automate the deployment of a React web application. By utilizing Docker for containerization, Jenkins for automation, and modularized Terraform code for infrastructure provisioning, I achieved an efficient, scalable, and automated deployment process. The project's success showcased my proficiency in streamlining deployment

workflows and incorporating DevOps principles into real-world scenarios, ultimately improving the efficiency and reliability of the application deployment process.

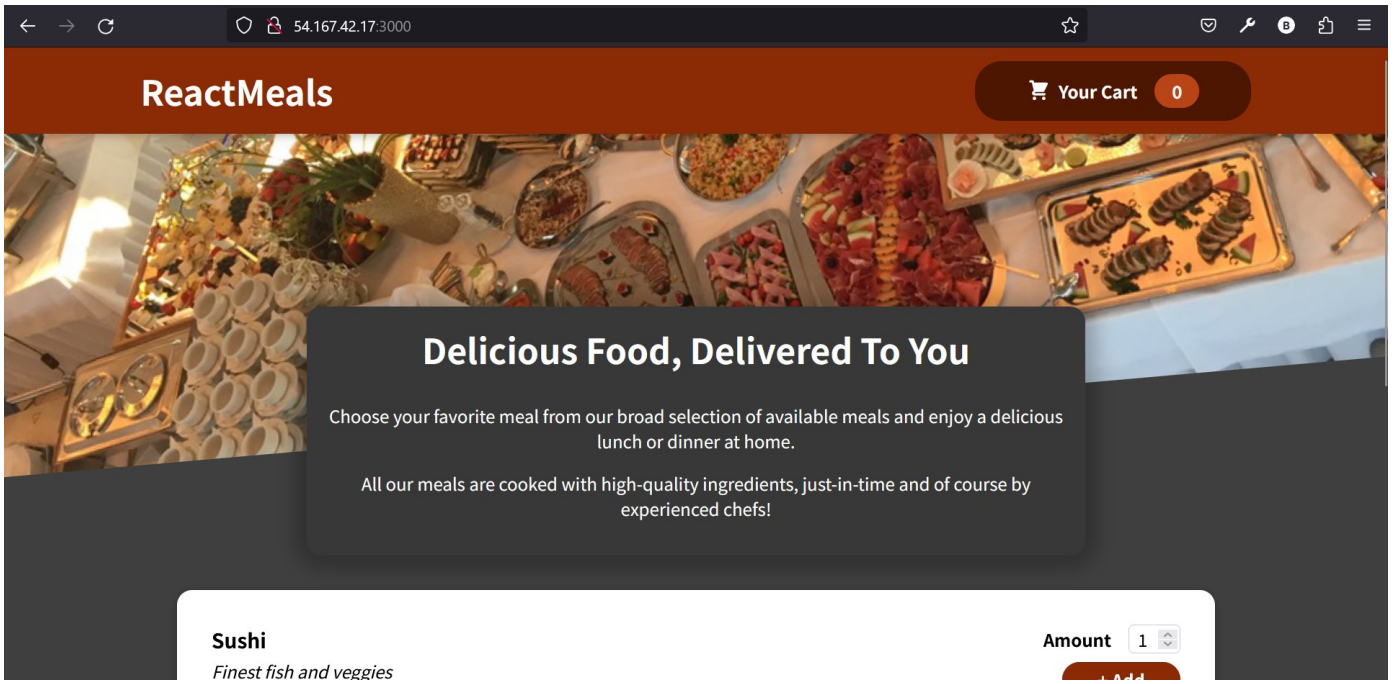
Jenkins Dashboard



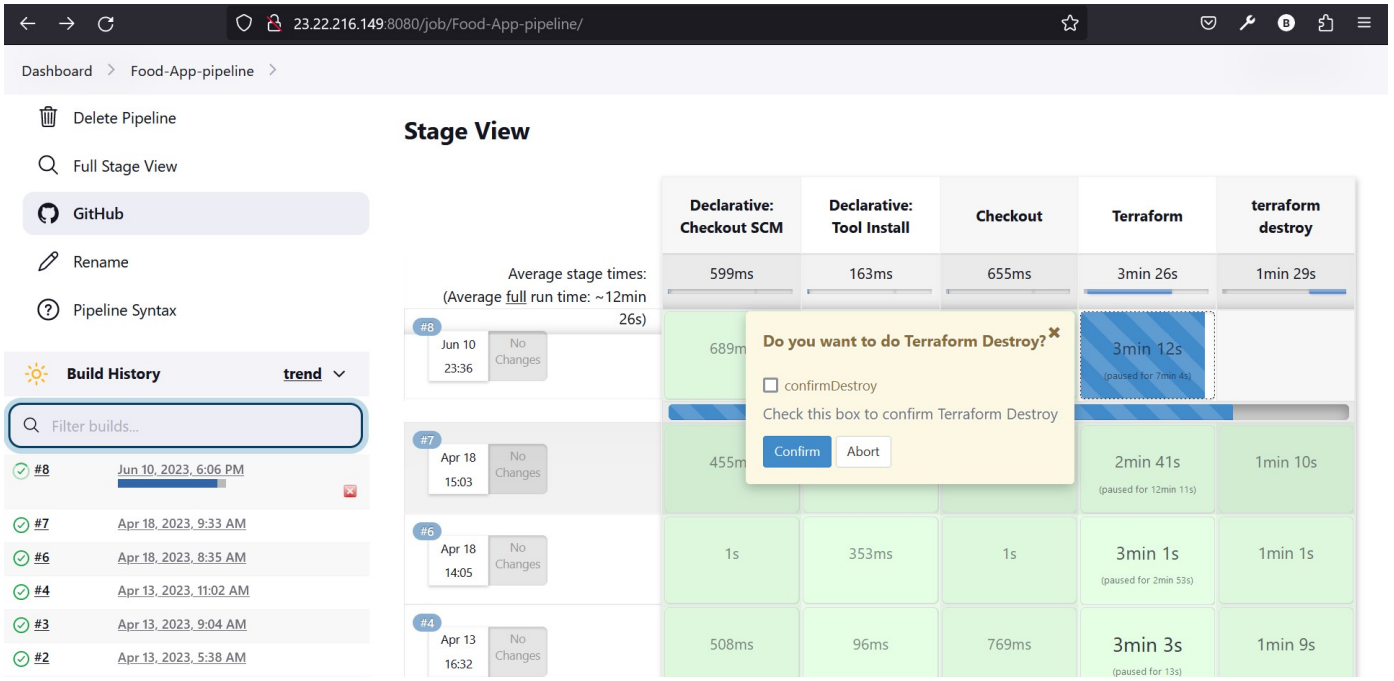
Completion of building infrastructure and deploying the docker image on ec2



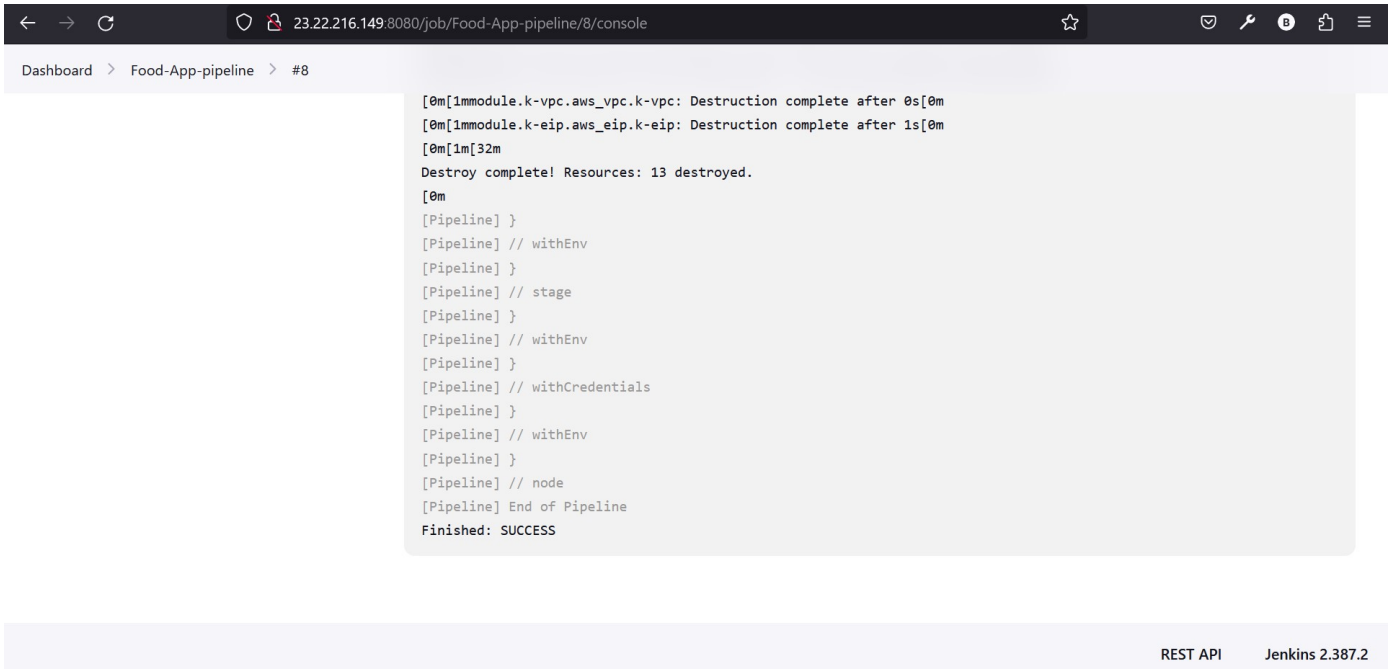
Application Deployed on newly created EC2



Pipeline Asking for Instruction: Should the Infrastructure Be Destroyed or Not?



Infrastructure Destroyed



PROJECT 4

Title: Part 2: Deployment of React Web Application on EKS Cluster with Terraform, NLB Service, IAM Role-Based Access Control, and EBS Storage

Introduction: Building upon the previous phases of the project, I extended the deployment of the React web application on the EKS (Elastic Kubernetes Service) cluster. This phase focused on utilizing Amazon Elastic Block Store (EBS) as the storage class for creating persistent volumes and volume claims. By incorporating EBS, I ensured reliable and persistent storage for the application's data within the cluster.

Tools and Technologies Utilized:

1. Infrastructure Provisioning: Terraform
2. Container Orchestration: Elastic Kubernetes Service (EKS)
3. Load Balancing: Network Load Balancer (NLB)
4. Identity and Access Management: IAM Roles and Role Bindings
5. Object Storage: Amazon S3
6. Persistent Storage: Amazon Elastic Block Store (EBS)

Infrastructure Provisioning with Terraform: Leveraging Terraform, I continued the infrastructure provisioning process by creating the necessary resources, including the EKS cluster, worker nodes, networking components, and namespaces. Terraform provided an automated and scalable approach to deploy the infrastructure consistently across environments, ensuring efficient management.

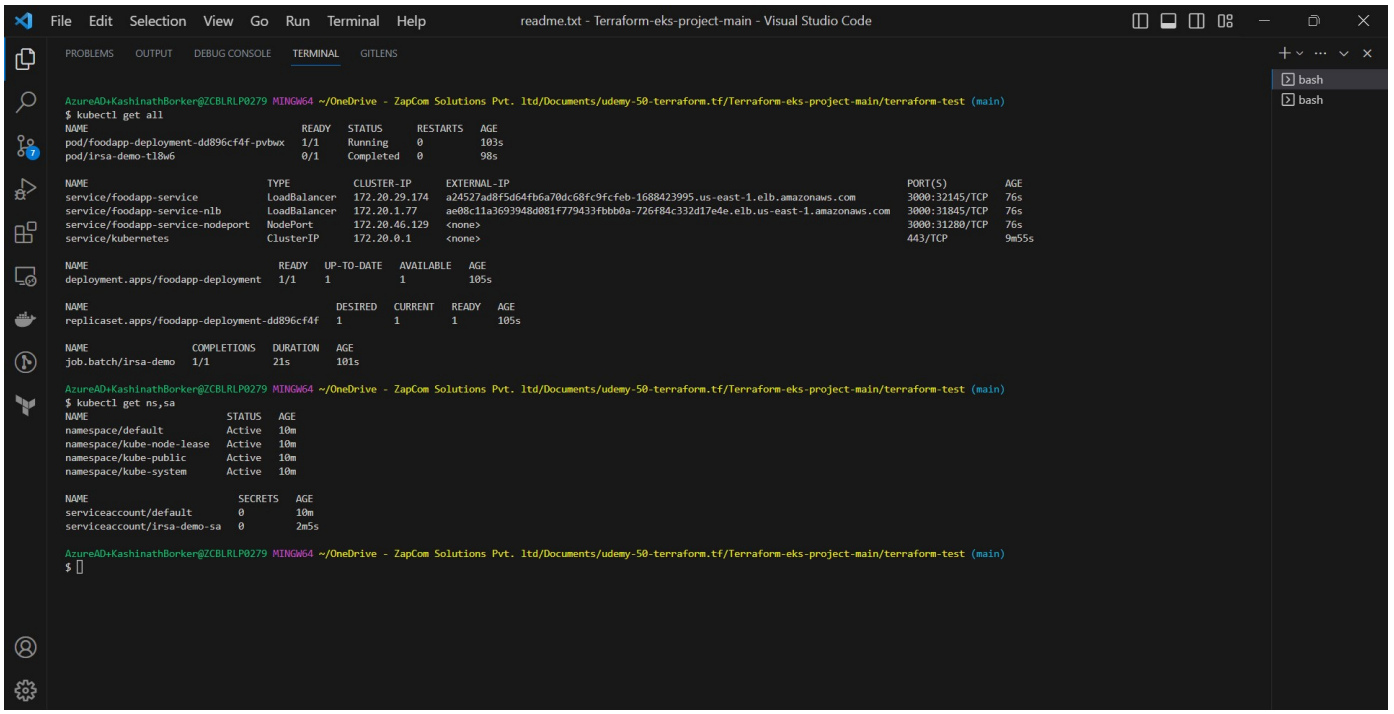
Integration of Amazon EBS for Persistent Storage: To enable persistent storage for the React web application, I utilized Amazon Elastic Block Store (EBS) as the storage class within the EKS cluster. EBS volumes provided reliable and durable block-level storage for containerized applications, allowing data to persist even when containers were terminated or rescheduled.

Creation of Persistent Volumes (PVs) and Volume Claims (PVCs): Using Kubernetes manifests, I defined Persistent Volume (PV) objects representing the EBS volumes and Persistent Volume Claim (PVC) objects representing the application's storage requirements. The PVs were bound to the PVCs, allowing the application to claim and utilize the persistent storage as needed. This ensured that data persisted across application restarts or scaling operations.

Benefits of EBS for Persistent Storage: The integration of EBS for persistent storage provided several advantages for the deployment of the React web application:

1. **Data Durability**: EBS volumes offered durability guarantees, ensuring that data persisted even in the event of container failures or rescheduling.
2. **Scalability**: The integration of EBS allowed for scaling the application's storage independently from the compute resources, enabling efficient resource allocation and improved application performance.
3. **Seamless Integration**: EBS seamlessly integrated with the EKS cluster, allowing the application to utilize persistent storage without additional complex configurations or dependencies.
4. **Data Persistence**: The utilization of EBS volumes ensured that data remained persistent even when pods or containers were terminated, providing a consistent and reliable user experience.

Conclusion: In this phase of the project, I extended the deployment of the React web application on the EKS cluster by incorporating Amazon Elastic Block Store (EBS) for persistent storage. By leveraging Terraform, I provisioned the necessary infrastructure resources, and by integrating EBS as the storage class, I ensured reliable and durable storage for the application's data. The creation of Persistent Volumes (PVs) and Volume Claims (PVCs) enabled the application to utilize and persist data seamlessly. This successful project demonstrates my ability to incorporate advanced technologies and best practices to enhance the scalability, reliability, and storage capabilities of applications deployed on an EKS cluster.



INTERNSHIP TIMELINE

Week 1: Git Fundamentals

Introduction to version control systems
Setting up Git and creating a repository
Basic Git commands: clone, add, commit, push, pull
Branching and merging

Week 2: Linux Basics

Introduction to Linux operating system
Command-line navigation and file management
Permissions and user management
Shell scripting basics

Week 3: AWS Fundamentals

Introduction to Amazon Web Services (AWS)
Setting up an AWS account and IAM users
Exploring key AWS services: EC2, S3, RDS
Understanding security and access control in AWS

Week 4: Networking Concepts

Basics of networking protocols (TCP/IP, DNS, HTTP)
IP addressing and subnets
Network security fundamentals (firewalls, VPNs)
Introduction to load balancing and CDN

Week 5: Docker

Introduction to containerization
Installing and running Docker containers
Building custom Docker images
Managing containers and networks with Docker Compose

Week 6: Jenkins and Groovy Scripting

Introduction to Jenkins and its role in CI/CD
Setting up Jenkins server
Creating and configuring Jenkins jobs
Automating Jenkins workflows using Groovy scripting

Week 7: Working On Project 1 - Continuous Web Deployer

- Reviewing and reinforcing concepts learned so far
- Developing a CI/CD pipeline for automatic deployment of a website
- Utilizing Git, Docker, Jenkins, SonarQube, and Nexus Repository
- Ensuring code quality analysis and artifact management

Week 8: Terraform Fundamentals

- Infrastructure as Code (IaC) principles
- Installing and configuring Terraform
- Creating and managing AWS resources with Terraform
- Understanding Terraform modules and variables

Week 9: Continued learning Advanced Terraform

- Reviewing key Terraform concepts and best practices
- Practicing writing Terraform configurations
- Exploring advanced Terraform features
- Preparing for the Terraform Associate certification exam

Week 10: Worked on Project 2 - Scalable Infrastructure with Terraform

- Designing and implementing scalable infrastructure using Terraform
- Leveraging AWS services: EC2, VPC, subnets, routing tables, security groups, IGW, and NAT
- Modularizing Terraform code for efficiency and reusability

Week 11: Kubernetes Fundamentals

- Introduction to Kubernetes and container orchestration: Understanding the need for container orchestration and an overview of Kubernetes architecture and components. Exploring Kubernetes master and worker nodes.
- Deploying and managing applications with Kubernetes: Creating Kubernetes deployments and pods, managing application scaling and updates, and understanding Kubernetes services for networking and load balancing.

Week 12: YAML and Advanced Kubernetes Concepts

- YAML fundamentals for defining Kubernetes resources: Introduction to YAML syntax and structure, creating YAML manifests for pods, deployments, services, and other resources, and validating and applying YAML configurations with kubectl.
- Advanced Kubernetes Concepts: Exploring networking in Kubernetes with services, ingress, and DNS. Configuring persistent storage with Kubernetes, including volumes and persistent volume claims (PVCs). Understanding Kubernetes security measures such as RBAC (Role-Based Access Control) and Pod Security Policies.

Week 13: Advanced Kubernetes Concepts

- Kubernetes networking and service discovery
- Deploying stateful applications with persistent volumes
- Understanding Kubernetes deployments and stateful sets
- Implementing Kubernetes security best practices

Week 14: Microservices Voting App Deployment

- Designing a microservices architecture for a voting application
- Implementing the microservices voting app on a local Kubernetes cluster
- Testing and validating the microservices voting app
- Exploring security best practices for Kubernetes

Week 15: Learning Helm

- Introduction to Helm package manager
- Installing and configuring Helm
- Managing Kubernetes applications with Helm charts
- Deploying and upgrading applications using Helm

Week 16: Working on Project 3 - Automation and Deployment with Docker, Jenkins, and Terraform

- Leveraging DevOps practices for automated deployment
- Containerizing a React-based web application with Docker
- Creating a Jenkins pipeline for automated deployment
- Integrating Terraform for efficient infrastructure provisioning

Week 17: EKS Cluster Creation with AWS Console

- Exploring Amazon Elastic Kubernetes Service (EKS) cluster creation using the AWS Management Console
- Configuring cluster settings, including VPC, subnets, and security groups
- Integrating worker nodes into the EKS cluster
- Managing scaling options and monitoring the cluster's health and performance

Week 18: EKS Cluster Creation with Terraform

- Leveraging Terraform to automate the provisioning of an EKS cluster
- Defining the necessary resources, such as VPC, subnets, and security groups, using Terraform code
- Automating the creation and configuration of the EKS cluster by running Terraform scripts
- Verifying the successful creation of the EKS cluster using Terraform output and AWS Console

Week 19: Project 3 - Part 2: Deployment on EKS Cluster

- Extending the deployment of the React web application on the EKS cluster
- Configuring network load balancing using AWS Network Load Balancer (NLB)
- Implementing IAM (Identity and Access Management) roles and policies for secure access control within the EKS cluster
- Deploying and scaling the React web application on the EKS cluster using Terraform and Kubernetes manifests

Week 20: Terraform for IAM Roles and Policies and Worker Nodes Deployment

- Understanding IAM roles and policies in AWS and their importance for secure access management
- Leveraging Terraform to define IAM roles and policies for the EKS cluster and associated resources
- Automating the creation, modification, and deletion of IAM roles and policies using Terraform scripts
- Exploring the requirements and considerations for deploying worker nodes on Zapcom's internal compute infrastructure

My Internship Experience at Zapcom

My internship at ZapCom has been an incredible learning experience, and I am immensely grateful for the opportunities I've had. The company's culture is characterized by a vibrant energy and a genuine enthusiasm from every employee, which made each day at work exciting and inspiring. I've been fortunate to learn a great deal about the industry and grow both personally and professionally during my time here.

One aspect that truly stood out at ZapCom is the unwavering commitment to continuous learning. As an intern, I was assigned a mentor who not only provided guidance but also became a trusted friend. My mentor was not only cool and friendly but also generously shared their knowledge and expertise in DevOps, ensuring I understood the intricacies of the field and became familiar with the various tools and technologies involved.

What I truly appreciated about ZapCom was the culture of collaboration and support. While my mentor played a crucial role, I was also encouraged to interact with and seek guidance from any of my colleagues. This inclusive environment allowed me to tap into a wealth of knowledge and experience, facilitating my professional growth.

Throughout the course of my internship, I had the opportunity to demonstrate my learnings through several demo presentations, including proof-of-concepts (POCs), which I shared with my mentor and the Delivery Head of ZapCom. These presentations not only showcased my progress but also allowed me to receive valuable feedback and insights from experienced professionals.

Additionally, ZapCom organized various engaging events and activities to foster team bonding and celebrate our accomplishments. Fun Fridays and games like 'Guess the Items on my Desk?' and 'Catch the Cup if you can' brought joy and a sense of camaraderie among the team. Moreover, we celebrated festivals like Holi, further strengthening the bond within the company.

In conjunction with these experiences, ZapCom also arranged informative tech talks conducted by senior experts. These sessions delved into important topics such as DynamoDB and Agile foundations. Participating in these tech talks provided invaluable opportunities to deepen my understanding of these concepts and stay abreast of industry trends. I was able to engage with the speakers, ask questions, and engage in meaningful discussions, further enriching my learning experience.

Overall, my internship at ZapCom has been a transformative journey. The knowledge I have gained, coupled with the incredible individuals I have had the pleasure of working with, has surpassed my expectations. I am deeply grateful for the opportunity to be part of ZapCom and extend my heartfelt wishes to everyone in the company.

References

- GitHub: https://github.com/docs
- Docker: https://www.docker.com/docs
- Jenkins: https://www.jenkins.io/docs
- SonarQube: https://www.sonarqube.org/docs
- Nexus Repository: https://www.sonatype.com/nexus/repository-oss/docs
- Maven: https://maven.apache.org/docs
- Terraform: https://www.terraform.io/docs
- AWS (Amazon Web Services): https://aws.amazon.com/docs
- EC2 (Elastic Compute Cloud): https://aws.amazon.com/ec2/docs
- VPC (Virtual Private Cloud): https://aws.amazon.com/vpc/docs
- Subnets: https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Subnets.html
- Routing Tables: https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html
- Security Groups: https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html
- IGW (Internet Gateway): https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html
- NAT (Network Address Translation): https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat.html
- EKS (Elastic Kubernetes Service): https://aws.amazon.com/eks/
- NLB (Network Load Balancer): https://aws.amazon.com/elasticloadbalancing/features/#:~:text=Network%20Load%20Balancer%20(NLB)%20is,SSL%20encryption%20and%20SSL%20offload.
- IAM (Identity and Access Management): https://aws.amazon.com/iam/
- Amazon S3 (Simple Storage Service): https://aws.amazon.com/s3/
- EBS (Elastic Block Store): https://aws.amazon.com/ebs/