INTERNSHIP REPORT

SHAWN L. D'SOUZA

1910

Zapcom Solutions Pvt Ltd.

Goa University

REPORT OF INTERNSHIP DONE AT ZAPCOM SOLUTIONS PVT LTD

SUBMITTED BY

SHAWN L. D'SOUZA

1910

UNDER THE GUIDANCE OF

Mr. Deepak Puranam

(Head of Delivery, Zapcom)

Mr. Saravanakkumar S R

(Director- Platform Architect, Zapcom)

Mr. Tarun Rapeta

(Software Engineer, Zapcom)

Internship Letter

3rd June 2022

INTERNSHIP LETTER

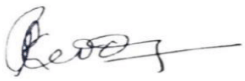
This is to certify that **Mr. Shawn Dsouza**, student of MCA, 6th Semester from Goa University is working as an Intern with our Organization at Bangalore location since 10-Jan-2022, his six months internship period will get over on 30-June-2022.

During the period of his Internship program with us, he is found punctual, hardworking and inquisitive.

We wish him a successful career ahead.

Yours Sincerely,

For **ZapCom Solutions Pvt Ltd**.



SRINIVAS KOTHAKOTA COO

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

www.zapcg.com
CIN: U72200KA2014PTC075530

USA
ZapCom Group, Inc.
7033 Village Parkway, Ste.211
Dublin, CA 94568
Ph: +1-925-350-7003

GOA UNIVERSITY

GOA BUSINESS SCHOOL

CERTIFICATE OF EVALUATION

This is to certify that **Mr. Shawn L. D'souza** 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 fulfilment 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

First of all, I would like to thank the almighty God for giving me the courage and strength throughout the internship period.

I extend my supreme gratitude to Mr. Kishore Pallamereddy (Executive Chairman & CEO, Zapcom) and Zapcom Solutions Pvt Ltd. for giving me the opportunity to intern at Zapcom and broaden my perception on how the real world in the field of Information and Technology looks like.

My sincerest gratitude to Mr. Saravanakkumar S R (Director- Platform Architect, Zapcom) and Mr. Tarun Rapeta (Software Engineer, Zapcom) for being my mentor and giving me the necessary guidance and support.

I would also like to thank the HR and Accounts team in Zapcom for making me understand the company policies and helping me in many other ways.

I would like to extend my gratitude to Mr. Nagarjuna Reddy (Tech Lead , Zapcom) for helping me and guiding me in the technologies I was working on.

I thank Mr. Ramdas Karmali (Prof. MCA, Goa Business School, Goa University, Mr. Jarret Stevan Anthony Fernandes (Assistant Prof, and TPO, MCA, Goa Business School, Goa University) and all the faculty, non-teaching staff of MCA department, Goa University for their constant encouragement and support during the course.

I would like to thank my family and friends for being supportive and for those friends in Bangalore who made me feel comfortable in a new location. The care, support and encouragement they provided me during this internship cannot be forgotten.

Finally, I would like to express my gratitude towards the Zapcom family who were always ready to help me and guide me in all aspects of life. They have transformed me into a new and renewed person ready to face challenges that come my way.

- Shawn L. D'souza

Table of Contents

Introduction

This report is a short description of my full-time on-site internship at Zapcom, Bangalore.

I joined Zapcom as an Intern on 10th January 2022 and have been here since then. This report contains necessary information about the organization, the proof of concept (POC) projects I have worked on , the other tasks and online courses that I have completed during this internship period.

In the content that will follow, I will talk about the company, the work here, the culture, etc. Then, I shall elaborate on the POC's I have worked on and tasks that I have completed.

This report highlights my learning experience and my contributions to the organization as an intern. This will describe the knowledge that I gained by successfully completing the tasks that were assigned to me.

I'll also be talking about the tools and technologies that were used followed by my internship timeline.

I shall conclude by sharing my experience and how it has helped me to grow, both, on the personal and professional front.

Company Profile

ZapCom Group is a US based Product and technology Start-up, focusing on Travel & Logistics, Ecommerce & Retail, Banking & Fintech domains. It is a Venture floated by a couple of KEY global leaders who had led MNC's globally and worked in the US for over 20 years. Currently Zapcom is located in 5 different global locations.

Zapcom are passionate about building digital products and platforms that can bend revenue and cost curves. They build, operate and optimize technology for their clients by taking a data-centric approach to creating products, platforms, and teams that drive delightful experiences and measurable business value.

The approach that defines Zapcom with their work include Solution co-creation, Elasticity, Security first, Metrics Driven, Automate, Product Mindset, and Customized Pods.

Certification Courses and Self Study Completed During Internship

Udemy Certifications:

- Kubernetes for the Absolute Beginners – Hands on
- Hashicorp certified: terraform Associate- 50 practical Demos
- Certified Kubernetes Administrator (CKA) with Practice Tests
- Ultimate AWS Certified Solutions Architect Associate 2022
- Docker for absolute Beginner – Hands On – DevOps

YouTube Tutorials:

- Python – Flask
- Jenkins (CI/CD)
- Docker , Docker compose
- Github ,Github Actions
- Gitlab
- BitBucket
- SonarQube
- Azure Devops Intro
- Helm Charts – Kubernetes
- EFK Stack for logging in Kubernetes
- AWS Cloudwatch , Cloudfront, Cloudformation, Lambda function, API Gateway
- Ingress Controller

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.

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.

AWS CloudTrail is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account. Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail.

AWS Amplify is a set of purpose-built tools and features that lets frontend web and mobile developers quickly and easily build full-stack applications on AWS, with the flexibility to leverage the breadth of AWS services as your use cases evolve.

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

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries

SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL

PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins.

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.

GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.

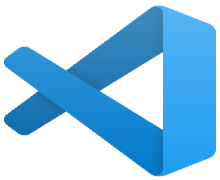
GitLab Inc. is the open-core company that provides GitLab, the DevOps software that combines the ability to develop, secure, and operate software in a single application.

Bitbucket is a Git-based source code repository hosting service owned by Atlassian. Bitbucket offers both commercial plans and free accounts with an unlimited number of private repositories

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

Postman is an application used for API testing. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface.

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data.



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

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps

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.

Fluentd is a cross platform open-source data collection software project originally developed at Treasure Data.

Kibana is a proprietary data visualization dashboard software for Elasticsearch, whose free and open source successor in OpenSearch is OpenSearch Dashboards.

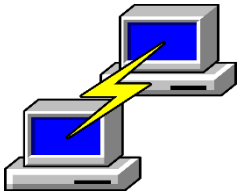
Azure DevOps Server is a Microsoft product that provides version control, reporting, requirements management, project management, automated builds, testing and release management capabilities. It covers the entire application lifecycle, and enables DevOps capabilities.

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

Ngrok is the programmable network edge that adds connectivity, security, and observability to your apps with no code changes

PGAdmin is a web-based GUI tool used to interact with the Postgres database sessions, both locally and remote servers as well.

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.

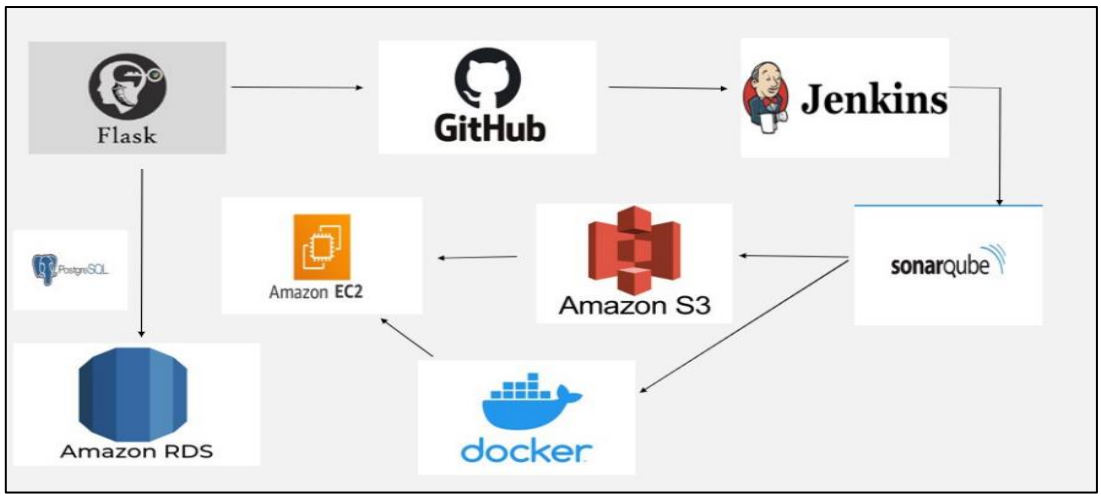
POC PROJECT 1 – Weather App (Flask)

Problem Statement:

Build a sample application using a programming language of your choice , dockerize it and deploy application to a cloud environment using CI/CD pipeline.

Description:

The application is built using python's micro web framework flask and connected to AWS RDS PostgreSQL database. This application allows users to add weather of their favourite cities over the world in their profile by querying the city name in the search bar provided and can even delete them from their profile. This application makes use of OpenWeather API to fetch the city along with weather details. The application is deployed to AWS EC2 instance by a CI/CD pipeline set up in Jenkins. The pipeline is set up with SonarQube for quality checks of the code and web hooks for GitHub to pull the code from GitHub repo.

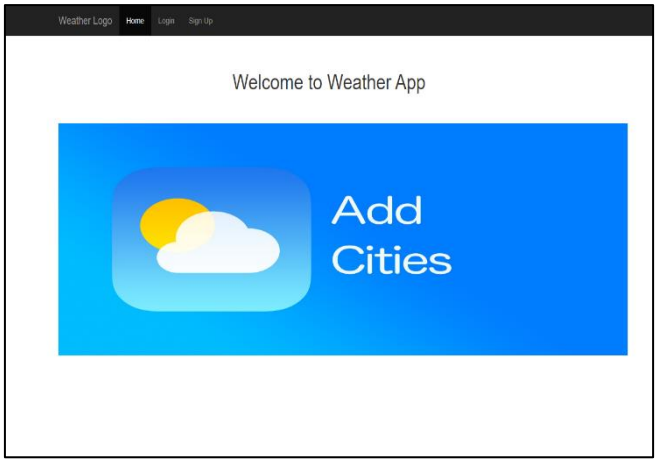


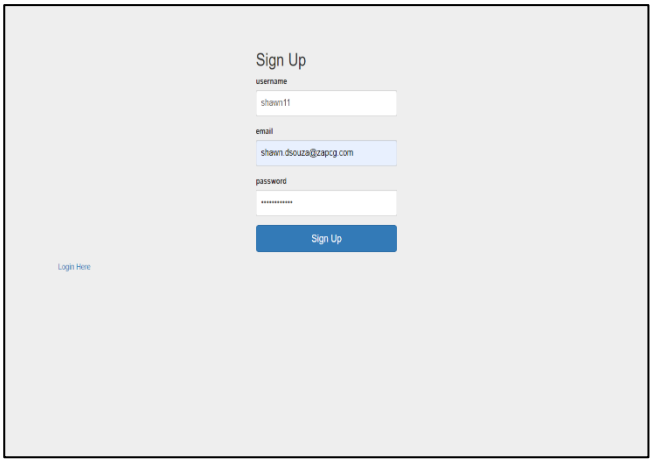
Pipeline flow

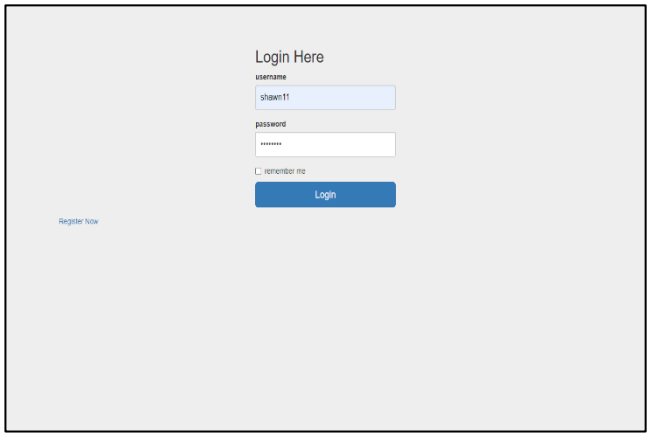
Tools and Technologies used:

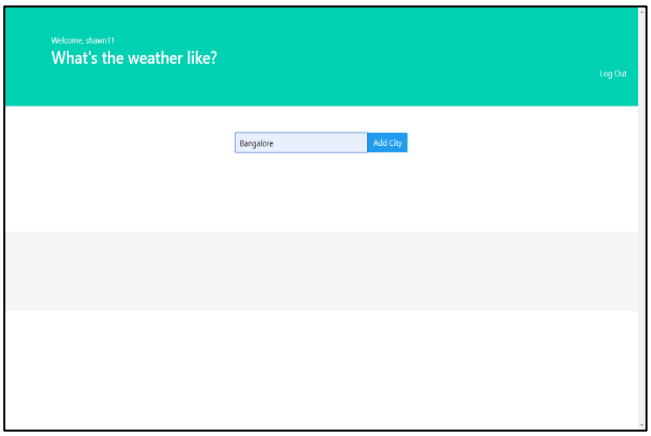
- Flask
- PostgreSQL
- AWS RDS
- AWS S3 Bucket, CLI
- AWS EC2 Instance
- SQLAlchemy
- Git & Github
- SonarQube
- Jenkins
- Docker
- VS Code

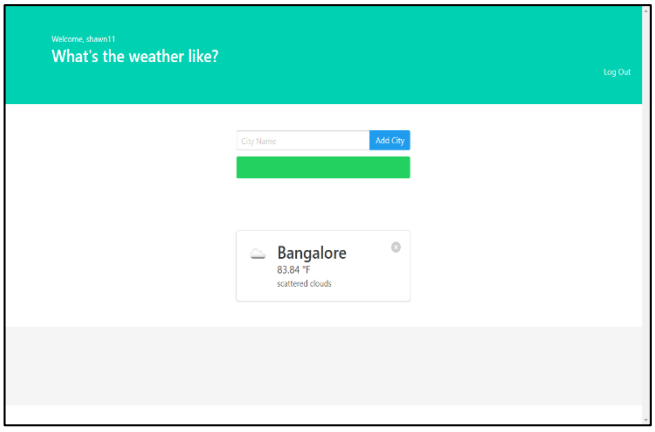
Screenshots:

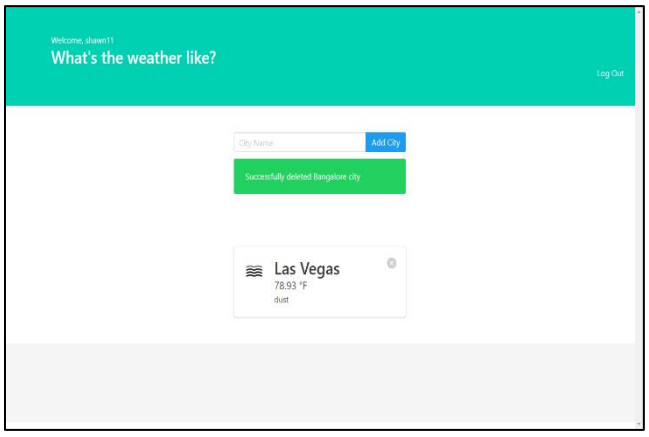


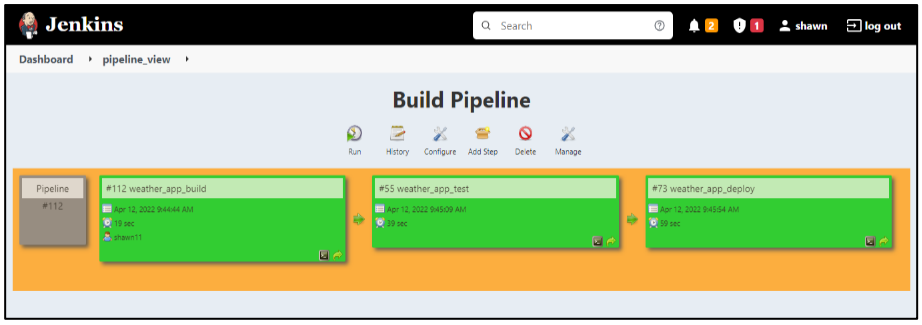












POC PROJECT 2 – Voting App Deployment

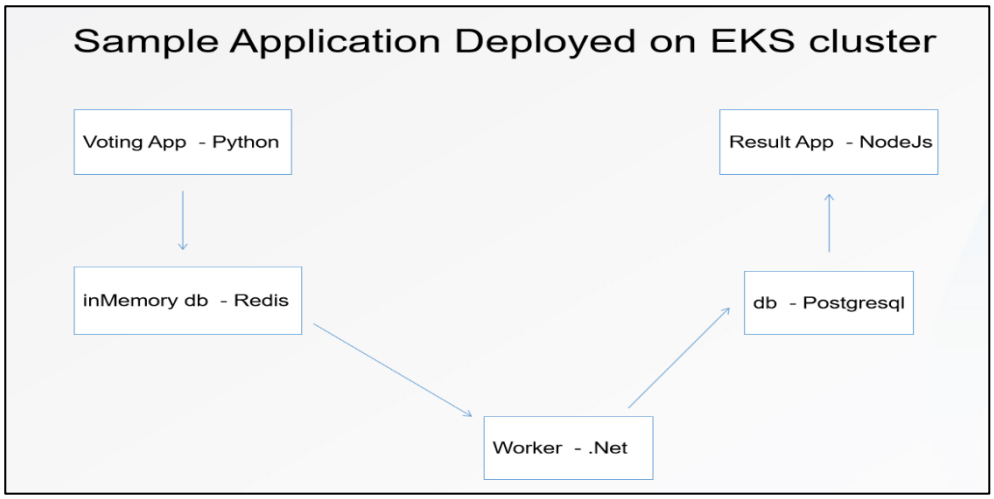
Problem Statement:

Search for an full stack application online and deploy it on AWS EKS cluster.

- Understand the working of the application
- Dockerize the application
- Deploy the application in local cluster and test it
- Configure the EKS cluster and application using Terraform,Kubernetes and helm charts

Description:

The voting app is a microservice application. This application has two frontend screens, namely the voting screen frontend built in python and result screen built in Nodejs. This microservice application makes use of redis to store the data in memory passed from the voting frontend screen, then from redis the data is passed to Backend application to do the vote calculations built in dot net core 2 and PostgreSQL Database to store the data which is then sent to the result frontend screen. This application is configured and deployed to EKS cluster using Terraform and Helm charts.

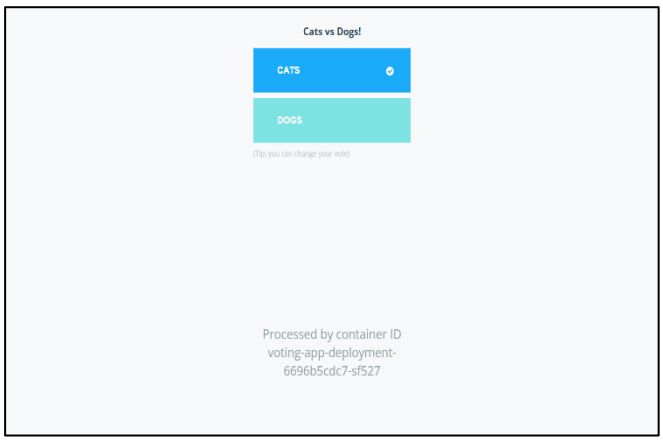


Application working Flow

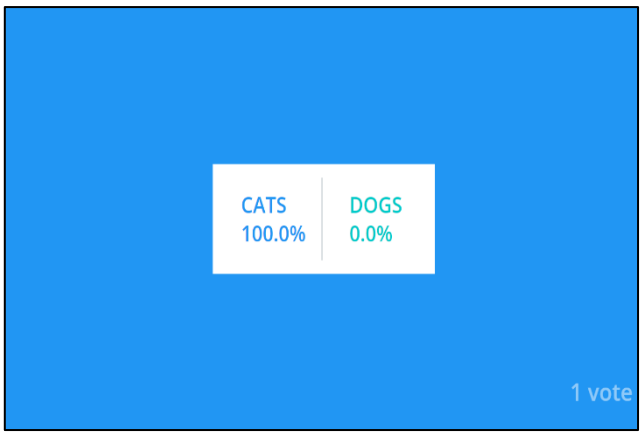
Tools and Technologies used:

- Terraform
- Kubernetes
- Helm charts
- AWS EKS
- Docker Desktop
- Docker
- VS Code
- AWS CLI

Screenshots:



Vote Frontend



Result Frontend

```
PS C:\Users\capcom\Desktop\Terraform_practice\Task3_updated - Copy> kubectl get svc
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
db               ClusterIP   10.100.174.193   <none>        5432/TCP       4m17s
kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP        47h
redis            ClusterIP   10.103.127.20    <none>        6379/TCP       4m17s
result-service   NodePort    10.103.167.43    <none>        80:30005/TCP   4m17s
voting-service   NodePort    10.96.174.105    <none>        80:30004/TCP   4m17s
PS C:\Users\capcom\Desktop\Terraform_practice\Task3_updated - Copy> kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
postgres-deployment-67455455d9-r9nwt  1/1     Running   0          18m
redis-deployment-5f864f88d4-ggztb     1/1     Running   0          18m
result-app-deployment-6d45cb6b4b-2r54r   1/1     Running   0          18m
voting-app-deployment-6696b5cdc7-n7x9w   1/1     Running   0          18m
worker-app-deployment-6d4dcbc8d7-plqwl   1/1     Running   0          18m
PS C:\Users\capcom\Desktop\Terraform_practice\Task3_updated - Copy>
```

cmd view of Pods & Services running in Kubernetes cluster (local)

POC PROJECT 3 – Property Buzz App Deployment

Problem Statement:

Search for an application and deploy it on AWS EKS cluster.

- Understand the working of the application
- Dockerize the application
- Deploy the application in local cluster and test it
- Configure the EKS cluster and application using Terraform,Kubernetes and helm charts(if required)
- Configure pods to terminate gracefully when in a live transaction
- Deploy EFK stack in eks cluster for logs

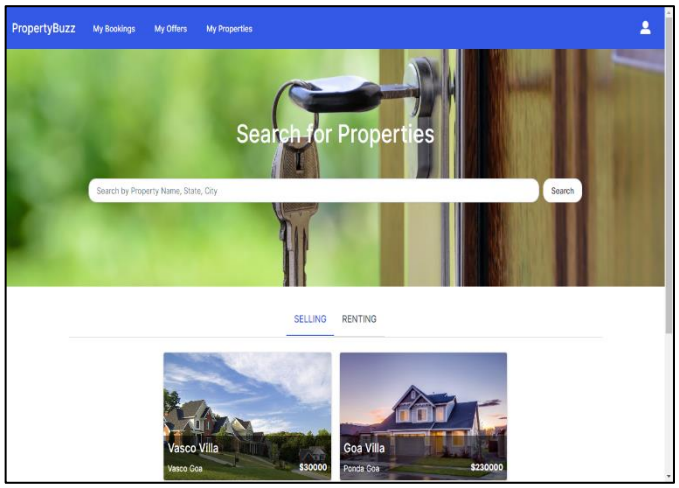
Description:

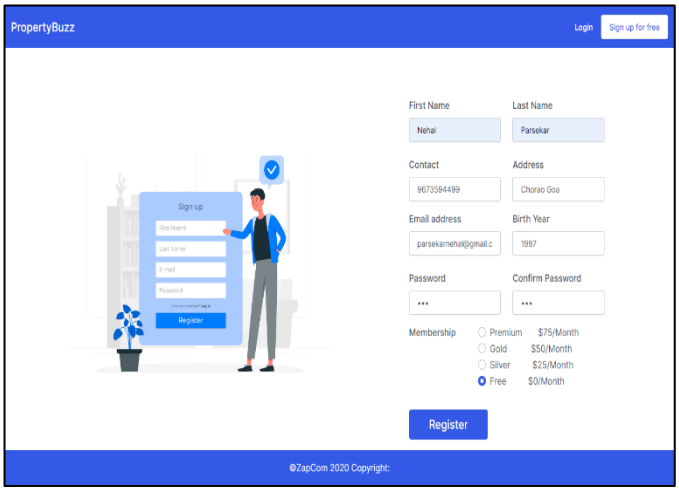
This is a Web API application that allows users to buy and rent properties. The API is built in Dot Net core 6 and front end is built in React and uses MS SQL database. This application is configured and deployed to EKS cluster using Terraform and Kubernetes. The database is deployed in docker container running in AWS EC2 instance.

Tools and Technologies used:

- Terraform
- Kubernetes
- AWS EKS
- Docker Desktop
- Docker
- VS Code
- AWS CLI
- EFK stack

Screenshots:






```
PS C:\Users\capcom\Desktop\Terraform_practice\PropertyApp_eks> kubectl get pods -n test-app
NAME                                        READY   STATUS    RESTARTS   AGE
property-app-deployment-58cb6f4686-rscbr    1/1     Running   0          47h
property-frontend-deployment-874fc97d6-2rffl   1/1     Running   0          45h
PS C:\Users\capcom\Desktop\Terraform_practice\PropertyApp_eks> kubectl get svc -n test-app
NAME                TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)
     AGE
frontend-app-service   LoadBalancer   172.20.97.74     aa417d86e4b3f44e3896470d31ddf07b-492162497.us-east-1.elb.amazonaws.com   3000:31209/T
CP   47h
property-service       LoadBalancer   172.20.196.189   a218ac3cc77bc410faac6c04044cf2a0-2088825227.us-east-1.elb.amazonaws.com   5000:30661/T
CP   47h
```

cmd view of Pods & Services running in Kubernetes cluster (EKS)

Other Tasks

AWS EC2 instance and EKS Deployment:

Deploy EC2 instance with security groups and EKS cluster using Terraform.

Tools and Technologies used:

- Terraform
- Kubernetes
- AWS CLI
- Docker Desktop
- AWS Management Console

Deploy Nginx in local Kubernetes cluster:

Deploy nginx using terraform and display custom html page instead of default nginx page

Tools and Technologies used:

- Terraform
- Kubernetes
- Docker Desktop

Deploy MySQL, Mongodb in local Kubernetes cluster:

Deploy MySQL and MongoDB in local cluster using Terraform

Tools and Technologies used:

- Terraform
- Kubernetes
- Docker Desktop

Deploy EFK stack in local Kubernetes cluster:

Deploy EFK stack using terraform and Hem charts

Tools and Technologies used:

- Terraform
- Kubernetes
- Docker Desktop , Helm Charts

Internship Timeline

- **Week 1**

  - Interaction with Global HR, CEO and CCO of Zapcom
  - Fun Session with HR Team
  - Interaction with mentor assigned to me and was assigned tasks.
  - Setup terraform, aws cli locally
  - Read through documentation of terraform
  - Read through documentation of aws – ec2 instance
  - Created AWS Iam user to authenticate aws cli with access keys of aws account
  - Created basic aws EC2 instance using terraform configuration
  - Enrolled for Udemy course to start learning Kubernetes hands on

- **Week 2**

  - Read about Kubernetes , EKS cluster
  - Learning Kubernetes
  - Installed Minikube to use Kubernetes locally.
  - Went through sample configurations online of eks cluster
  - Completed Intro to Kubernetes course
  - Started learning Terraform (udemy course)

- **Week 3**

  - Learning Terraform – (udemy course)
  - Reading documentation of terraform

- **Week 4**

  - Completed Terraform course
  - Practicing Kubernetes and terraform learnings
  - Started with AWS course – (udemy)

- **Week 5**

  - Learning AWS - (udemy course)
  - Reading through docs/articles

- **Week 6**

  - Learning AWS - (udemy course)
  - Practicing the learnings in AWS hands on
  - Completed AWS Course
  - Started with Kubernetes Certified Kubernetes Administrator Course
  - Learning Docker – (youtube/Documentation)
  - Setup docker locally

- **Week 7**

  - Learning Kubernetes CKA hands on (CKA course on udemy)
  - Provisioning Virtual Machines using Vagrant/ Virtual Box
  - Completed with Kubernetes CKA course

- **Week 8**

  - Working with kubeadm in virtual machine
  - Setup docker desktop on local system
  - Practiced docker
  - Practiced Kubernetes
  - Learnt Pod Security Policies
  - Learnt Ingress and egress controllers
  - Learnt Cluster roles and role bindings
  - Read about Flannel and Calico in kubernetes
  - Learnt Aws kubernetes networking
  - Learnt Aws cloud controller, Event Bridge , Cloudfront, Api Gateway

- **Week 9**

  - Practice Terraform
  - Set up eks cluster using Terraform
  - Interacted with Delivery Head of Zapcom
  - Error Debugging in Terraform configuration
  - Started learning Flask
  - Started working with small flask application (Weather App) –POC 1
  - Created docker and docker compose file for Version1 of the application

- **Week 10**

  - Worked with flask Application Version 2
  - Learnt about git branches
  - Pushed code on github
  - Error Debugging in flask application
  - Structured the python code into proper modules
  - Setup Jenkins locally
  - Read about CI/CD implementation in Jenkins
  - Error Handling in flask application
  - Started learning Jenkins –(youtube)
  - Altered docker compose files

- **Week 11**

  - Learning Jenkins
  - Started implementing pipeline script in Jenkins for flask Application
  - Building pipeline using freestyle project jobs
  - Setup ec2 instance using terraform
  - Unit testing in Flask
  - Adding webhooks to Jenkins
  - Completed CI part of pipeline
  - deployed app to ec2 instance without pipeline
  - Copied files from local machine to S3 bucket using aws cli
  - Synced files from S3 bucket to ec2 instance using aws cli

- **Week 12**

  - Logging errors in flask application
  - Researched about Sonarqube
  - Shifted Database from local Postgresql db to RDS db & configured with flask app
  - Deployed app again with changes to ec2 instance without pipeline
  - Completed the deployment of flask app through Jenkins
  - Error Debugging
  - Researched about usage of tools such as GitLab, BitBucket , & Azure DevOps
  - Used the above tools to know the working.

- **Week 13**

  - Set up Sonarqube on local system
  - Integrated Sonarqube with Jenkins
  - Researched about github actions and tried creating a pipeline for flask app
  - Altered some part in Jenkins pipeline
  - Logging of flask app in terminal and to a file simultaneously
  - Edited docker compose file
  - Error debugging

- **Week 14**

  - Task assigned to install nginx and mysql using kubectl commands in local Kubernetes cluster
  - Completed task assigned
  - Demo Presentation of flask application with CI/CD pipeline to Delivery Head of Zapcom.
  - Installed nginx and mysql in local cluster using terraform configuration
  - Error Debugging in terraform configuration

- **Week 15**

  - Mogodb deployment on local cluster using terraform configuration
  - Configured eks cluster using terraform and deployed nginx in it
  - Mogo-express deployment on local cluster using terraform
  - Configured mogodb and mongo-express running inside local cluster to connect to each other
  - Error Debugging
  - Added env variables and structuring eks cluster configuration

- **Week 16**

  - Added workspaces, tfvars , outputs and modules to eks cluster configuration
  - Deploy sample application in local cluster using terraform
  - Error debugging
  - Learning Helm charts in Kubernetes

- **Week 17**

  - Learnt how to use helm in terraform
  - Used helm as terraform resource for sample deployments to local Kubernetes cluster
  - Used structure of modules , workspaces in application deployment configuration
  - Deployed Application to eks cluster
  - Configured ALB for eks cluster
  - Researched about efk stack for logging in kubernetes

- **Week 18**

  - Researched about working of different load balancers in eks cluster
  - Used helm charts in the configuration of eks cluster and application deployment
  - Configured NLB instead ALB in eks cluster
  - Confgured ec2 instance and s3 bucket using terraform with some additional resources
  - Error debugging
  - Demo presentation of Sample Application deployment in eks cluster with terraform , helmand Kubernetes configuration to delivery head of Zapcom
  - Read about DevSecOps

- **Week 19**

  - EFK stack configuration using helm charts in terraform to deploy in local cluster
  - Error debugging of configuration
  - Researched how we can use DevSecOps in real world
  - Researched how we can gracefully terminate pods in between a live transaction
  - Set up dot net application locally to check working before deploying to kubernetes cluster
  - Researched about git branching strategies
  - Researched about implementation of DevSecOps in Azure Dev Ops pipeline
  - Dockerize Dot net api and react front end application with docker compose
  - Altering dot net code with developer of the code

- **Week 20**

  - Error debugging in Docker compose
  - Pushed docker images to docker hub
  - Application worked locally in Docker container
  - Configured dot net & react application using terraform to run in local Kubernetes cluster and connect to db in local docker container
  - Error Debugging in terraform configuration
  - Hosted MS SQL database in docker container in EC2 instance
  - Changed terraform configuration to connect to database running in ec2 instance
  - Configured dot net api in local cluster and react app on AWS amplify for testing purpose
  - Deployed Dot net api and react app in EKS cluster using terraform configuration
  - Error debugging in configuration

- **Week 21**

  - Debugging errors
  - New configuration for eks cluster
  - AWS api gateway integration with aws lambda function
  - Deployed app to ec2 instance on apache server and connected ec2 to nlb and it to api gateway
  - Researched about aws cloudfront service
  - Implementation of private and public ec2 instances
  - Researched about aws cloud formation service
  - Aws lambda function integration with api gateway and cloudfront
  - Aws event bridge
  - Researched about aws cloudwatch
  - Created dashaboards of ec2 metrics and and vpc flow logs
  - Researched about Prometheus and Grafana
  - Ingress resources an ingress controller
  - wsl setup on windows and setup minikube Kubernetes cluster in it

My Internship Experience at Zapcom

My Internship at Zapcom has been a great and learning experience. The culture and energy embodied and displayed by every single employee made it so exciting to come to work every day and learn about this incredible industry.

An un-dying hunger to learn more is something that is cultivated in employees here. Each intern had a mentor whom we worked under the whole internship period. My mentor and I bonded very quickly, and he didn't hesitate to show me the ropes of DevOps and all the required tools and technologies that goes into it.

At ZapCom, I was never restricted to just my mentor. I was provided freedom to interact, consult seek help and guidance, etc. from any colleague. During this period I had multiple demo presentations to showcase my learnings (POC's) to my mentor and Delivery head of Zapcom.

At ZapCom, there were multiple events like :

- Happy Hours
- Fun Fridays
- Potluck
- Holi Celebration

Between the experience gained and the people I was able to work with, this internship was more than I could have ever hoped for. Overall, I'm very happy to be part of ZapCom. My best of wishes to everyone here.

References

- https://www.terraform.io/docs
- https://kubernetes.io/docs/home/
- https://helm.sh/docs/
- https://docs.docker.com/
- https://flask.palletsprojects.com/en/2.1.x/
- https://docs.sqlalchemy.org/en/14/
- https://docs.sonarqube.org/latest/
- https://docs.aws.amazon.com/
- https://docs.aws.amazon.com/ec2/?id=docs_gateway
- https://docs.aws.amazon.com/eks/?id=docs_gateway
- https://www.postgresql.org/docs/
- https://kubernetes.io/docs/concepts/workloads/pods/
- https://docs.docker.com/network/
- https://www.terraform.io/language/resources/syntax
- https://kubernetes.io/docs/concepts/cluster-administration/networking/
- https://projectcalico.docs.tigera.io/getting-started/kubernetes/
- https://devopscube.com/setup-efk-stack-on-kubernetes/
- https://azure.microsoft.com/en-us/services/devops/
- https://stackoverflow.com/
- https://www.youtube.com/
- https://www.google.com/