Tribhuvan University

Institute of Science and Technology

Asian College of Higher Studies

Ekantakuna, Lalitpur, Nepal

An Internship Report

On

“DevOps Engineer “

At

F1Soft International Pvt. Ltd.

Submitted by

Saurab Tharu (T.U. Exam Roll No. 24256/076)

An Internship Report Submitted in partial fulfillment of the requirement of Bachelor of Science in Computer Science & Information Technology (BSc.CSIT) 8th Semester of Tribhuvan University, Nepal

June, 2024

MENTOR'S RECOMMENDATION

This is to recommend that **Saurab Tharu** has submitted the internship report entitled **"An Internship report on DevOps Engineer at F1Soft Pvt. Ltd."** for the fulfillment of the requirement of the Bachelor's degree of Computer Science and Information Technology (BSc. CSIT) is processed for the evaluation.

…….

Sagar Khatiwada

Mentor

F1Soft Pvt. Ltd.

Pulchowk, Lalitpur

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this report has been prepared under my supervision by **Saurab Tharu** entitled **An Internship Report on DevOps Engineer at F1Soft International Pvt. Ltd.** in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology of Tribhuvan University is processed for the evaluation.

. . . . . .

Er. Bidur Sapkota

Supervisor

Department of Computer Science and Information Technology

Asian College of Higher Studies

Ekantakuna, Lalitpur

ASIAN COLLEGE OF HIGHER STUDIES

ACHS

COLLEGE OF IT AND COMPUTER SCIENCE

(Affiliated to Tribhuvan University)

LETTER OF APPROVAL

This is to certify that this internship report prepared by **Saurab Tharu** entitled **"An Internship report on DevOps Engineer at F1Soft Pvt. Ltd"** has been submitted to the Department of Computer Science for acceptance in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology. In our opinion, it is satisfactory in the scope and quality as a project for the required degree

Signature of Supervisor	**Signature of Mentor**
.....	.....
Er. Bidur Sapkota Asian College of Higher Studies	**Mr. Sagar Khatiwada** F1Soft International Pvt. Ltd
Signature of HOD	**Signature of External Examiner**
.....	.....
Er. Pranaya Nakarmi Asian College of Higher Studies	**.....** IOST, Tribhuvan University

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this internship report. Special gratitude to my supervisor **Er. Bidur Sapkota** for the complete support and guidance throughout the internship period.

Also, I would like to express my special gratitude to our Program Coordinator, **Pranaya Nakarmi** and administrative staff whose all-time encouragement helped me coordinate the internship tasks systematically.

I would like to express my sincere thanks to my mentors **Mr. Sagar Khatiwada** and **Mr. Avash Shakya** of F1Soft International Pvt. Ltd. for sharing their valuable knowledge and guiding me during the internship period, and making me learn new skills and abilities. I am also grateful to the entire staff of F1Soft International Pvt. Ltd for their constant support and guidance

With all due respect and gratitude, I would like to give a word of thanks to the members of the IT department of Asian College of Higher Studies, who enc

ouraged me to perform work activities.

With Regards,
Saurab Tharu (24256/76)

ABSTRACT

The report provides an overview of the DevOps Engineer internship at F1Soft International, where involvement included setting up and managing CI/CD pipelines, automating deployment processes, and implementing infrastructure solutions using bare-metal servers. Kubernetes (K3s) was utilized for container orchestration and application deployment along with SSL certification provisioning. Furthermore, system reliability and performance were improved through monitoring tools like Prometheus, Grafana, and Uptime Kuma within the CI/CD pipeline and infrastructure.

The true meaning of DevOps culture, which encompasses automation, collaboration, and continuous learning, was taught during the internship. Collaboration with both the development team and the operations team ensured a smooth transition between development and deployment workflows. This experience has been very helpful because it has built upon previous practical knowledge of DevOps practices, thereby providing better preparation for future employment opportunities within this field.

Keywords : *DevOps, CI/CD, Automation, Bare-Metal Infrastructure, Kubernetes, Bash Scripting.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CI/CD	Continuous Integration and Continuous Development
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
HAProxy	High Availability
HTTPS	Hypertext Transfer Protocol Secure
IAC	Infrastructure as Code
ITOPS	IT Operation
LVM	Logical Volume Manager
NFS	Network File System
SELINUX	Secure Linux
SSL	Secure Socket Layer
TCP/IP	Transfer Control Protocol

Chapter 1: Introduction

1.1. Introduction

Over the course of my DevOps internship at F1Soft International, the chance was given to fully embrace the DevOps culture and observe how it has been used effectively to bridge the gap between software development and IT operations. F1Soft International is one of the most well-known names in innovative financial solutions, offering a wide array of innovative and advanced solutions to commercial banks, development banks, financial institutions, and several other large enterprises. The company is committed to using advanced technology with the view of enhancing financial inclusion and making digital experience seamless.The primary responsibility, as a member of the DevOps team, has been to learn how to streamline and enhance the development and operational processes, so that the resulting systems could be robust, efficient, and reliable.

The main duty included understanding the implementation of DevOps practices to automate and simplify different parts of the software delivery process. This entailed creation as well as optimization of Continuous Integration/Continuous Deployment (CI/CD) pipelines which automate integration of code changes frequently and reliably should be deployed; they are important in reducing time-to-market while increasing overall development team's productivity.

Additionally, there was an opportunity was given to observe how corporations utilize automation for handling repetitive tasks in their environments. Automation not only saves time but also reduces human error, thereby increasing reliability and yielding consistent results. Practical experience was gained in automating routine operations, configuring infrastructures without any challenges, and managing applications seamlessly deployed across different systems or platforms.

Real-world experience with DevOps, complementing the theoretical knowledge I previously acquired, has been one of the best outcomes of this internship. A better understanding of the collaborative and iterative nature of DevOps was also achieved. F1Soft has been able to produce quality software faster by focusing on continuous improvement, automation, and ensuring effective communication between development and operations teams. These skills and ideas will help in handling future challenges in this area, thereby contributing to becoming an even better engineer.

1.2. Problem Statement

F1Soft International, like many tech-driven organizations, faced challenges related to the manual processes in software deployment, the scalability of their infrastructure, and the efficiency of their operational workflows. The primary issues included:

i) **Manual Deployment Processes**:
   The existing deployment processes were largely manual, leading to inconsistencies, longer deployment times, and higher risk of errors.

ii) **Infrastructure Scalability**:
With a growing user base, the need for scalable infrastructure became critical. The current setup struggled to efficiently handle the increased load, affecting performance and user experience.

iii) **Operational Efficiency**:
The lack of automated workflows resulted in slower response times to incidents and less efficient use of resources.

Addressing these problems was crucial for maintaining F1Soft's competitive edge, ensuring customer satisfaction, and supporting the company's growth objectives.

1.3. Objectives

The primary objectives of my internship at F1Soft International were:

i) **Gain Professional Experience**:
Work in a real-world corporate environment to understand team dynamics, project management, and effective communication within a professional setting.

ii) **Develop Problem-Solving Skills**:
Tackle real-world challenges and develop solutions, enhancing critical thinking and problem-solving abilities.

iii) **Automate Deployment Processes**:
Implement CI/CD pipelines to automate the build, test, and deployment processes, reducing deployment time and errors.

iv) **Improve Operational Efficiency**:
Develop and integrate automated monitoring and alerting systems to enhance incident response times and operational efficiency.

1.4. Scope and Limitation

1.4.1. Scope

The scope of my internship included the following key areas:

i) **CI/CD Pipeline Implementation**:
Setting up automated pipelines for continuous integration and deployment on bare-metal servers.

ii) **Bare-Metal Infrastructure Management**:
Designing and deploying scalable solutions using physical servers.

iii) **Monitoring and Alerting**:
Implementing tools like Prometheus and Grafana for monitoring and setting up alerting mechanisms.

iv) **Security Enhancements**:
Adding security checks within the CI/CD pipeline and ensuring infrastructure compliance with security standards.

1.4.2. Limitations

Despite the comprehensive scope, there were some limitations during my internship:

i) **Time Constraints**:
The duration of the internship was limited, which restricted the depth of exploration and implementation of certain advanced DevOps practices and tools.

ii) **Resource Availability**:
Access to certain hardware and software resources was limited, which occasionally hindered the implementation and testing of specific solutions on a larger scale.

iii) **Learning Curve**:
The complexity of some tools and technologies, especially those I was unfamiliar with, required significant time to learn, reducing the time available for hands-on application.

iv) **Assigned Task Scope**:
The tasks assigned were predetermined, leaving limited room to explore additional areas of personal or emerging interest within the DevOps field.

1.5. Report Organization

This report is structured into four main chapters, each detailing different aspects of my internship experience at F1Soft International. Here is a brief overview of each chapter:

i) **Chapter 1: Introduction**
This chapter introduces the work completed during my internship. It outlines the problem statement, the objectives of the internship, the scope and limitations of the project, and provides an overview of the report's organization.

ii) **Chapter 2: Organization Details and Literature Review**
In this chapter, a comprehensive introduction to F1Soft International has been provided. This includes an overview of the organization, its hierarchy, the various domains in which it operates, and a detailed description of the department where internship has been completed. Additionally, this chapter includes a literature review or related study, highlighting relevant theories and frameworks that underpin the works that have been performed during the internship.

iii) **Chapter 3: Internship Activities**
This chapter delves into the specifics of my internship activities. It outlines my roles and responsibilities, provides a weekly log of the technical activities, describes the involved projects, and details the technical tasks and activities have been completed successfully. This section offers an in-depth look at the hands-on experience obtained.

iv) **Chapter 4: Conclusion and Learning Outcomes**

A brief overview of the experience gained during the internship is also stated in this last part, as well as the main conclusions. It mentions my skills and knowledge, challenges I faced and how I dealt with them. Additionally, the section talks about what the future holds in terms of career development after such an opportunity.

Chapter 2: Background Study and Literature Review

2.1. Introduction to Organization

F1Soft International is a leading financial technology company based in Nepal. Established in 2004, the company specializes in providing innovative digital financial solutions to a diverse range of clients, including banks, financial institutions, and enterprises in Nepal. F1Soft's mission is to revolutionize the financial services industry by leveraging cutting-edge technology to enhance financial inclusion and provide seamless digital experiences to its users. The company's portfolio includes a wide array of products and services such as mobile banking, payment gateways, digital wallets, and enterprise solutions, all designed to meet the evolving needs of the modern financial ecosystem. F1Soft is recognized for introducing mobile banking and mobile financial services in the country.

The Mobile Banking and Internet Banking platforms developed by F1Soft are currently used by over 90% of the banks in Nepal, serving in excess of 19 million people. The systems contribute to nearly 80% of the total digital payments in Nepal. The company's efforts in fintech innovation have been duly recognized by various national and international bodies, including the 2014 International Business Awards (Bronze), the 2013 FNCCI Service Excellence Award, and the 2012 Red Herring Top 100 Asia Award. In 2017, F1Soft was recognized as Nepal's Highest Tax Payer (IT Sector) by the Government of Nepal. F1Soft is also working with the Government of Nepal and a few other organizations to identify and develop digital financial solutions that benefit and improve the lives of the unbanked and underbanked across the country.

F1Soft is the market leader in the fintech industry due to its unwavering dedication to perfection, constant forward thinking and focus on the client. Financial inclusion for millions is being championed by F1Soft with its strong infrastructure and smart team that yearns to see this achieved through providing safe trustworthy easy to use financial solutions.

Table 2.1 : Company Details

Official name	F1Soft
Type of business	Fintech
Location	Pulchowk, lalitpur
Year of establishment	2004
Key service areas	Digital Payment Solutions, Banking Solutions
Staff size	600
Location of clients	Nepal

Expertise in	Financial Software Development, Digital Payment Systems, Data Analytics
Noteworthy mentions	Innovation in Digital Payments,Core Banking Solutions

2.2. Organizational Hierarchy

F1Soft International prioritizes innovation through its structure. The Board sets direction, and Executive Management implements it. Specialized departments handle tasks: Product Development builds software, Sales & Marketing drives growth, Customer Support keeps users happy, DevOps streamlines processes, and Finance & Admin manages the company's well-being. This structure fosters collaboration towards F1Soft's goals.

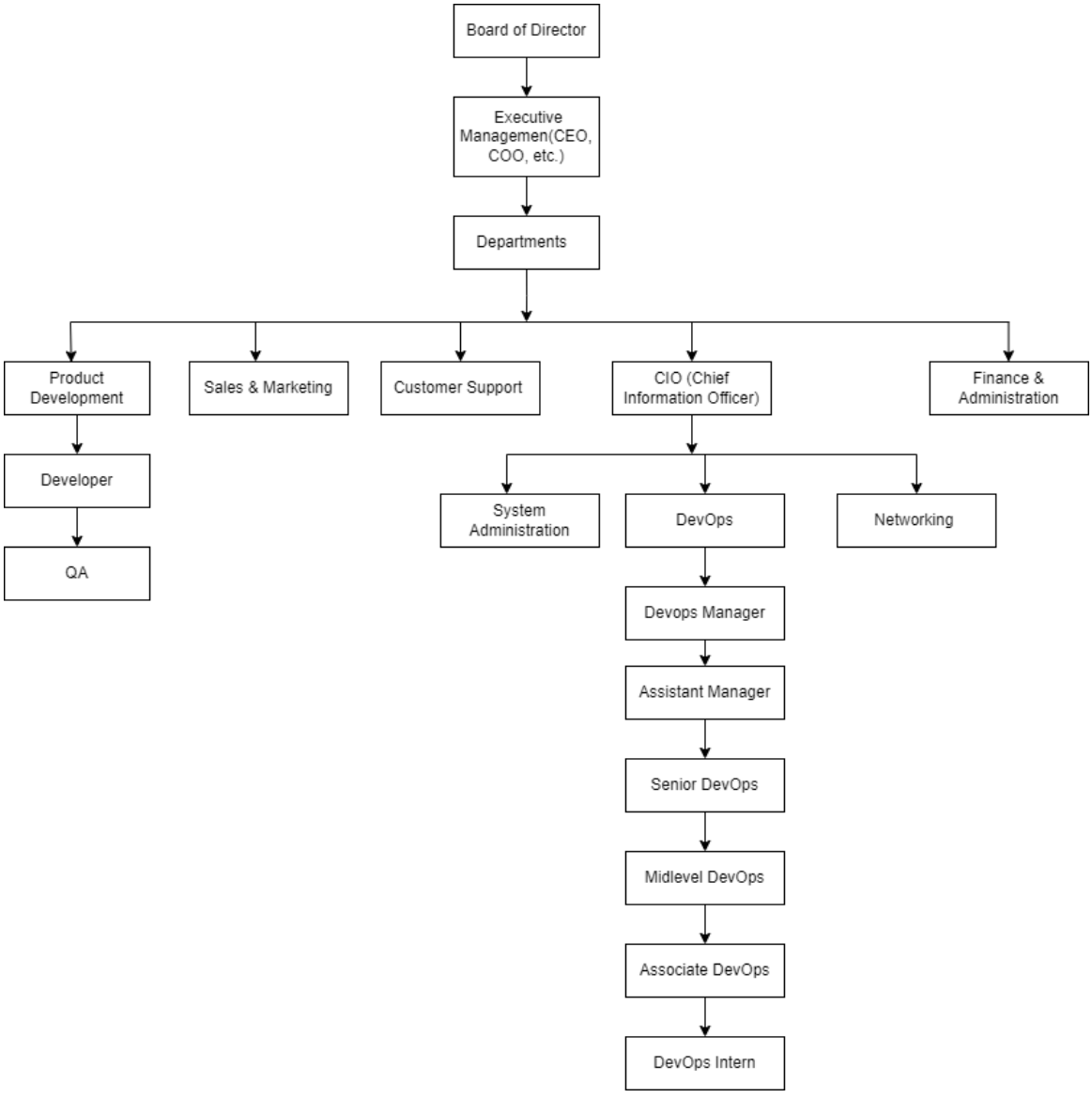


Figure 2.1 : Organizational Hierarchy

2.3. Working Domains of Organization

F1Soft International Pvt. Ltd is among the top technology firms located in Nepal that mainly focuses on financial technology solutions (fintech). Since its establishment in 2004, F1Soft has revolutionized digital financial services thus changing Nepalese and regional financial sector landscape. The company operates principally in:

i) **Digital Payment Solutions** - F1Soft is popularly known for its inclusive digital payment systems which consist of mobile wallets and payment gateways. Notable products under this category are eSewa (Nepal's first & leading digital wallet) and Fonepay (a prominent payment gateway that facilitates secure online transactions).

ii) **Core Banking Solutions** – It offers strong core banking systems that are meant to improve the operational efficiency of banks. These solutions help in various banking operations like account management, transaction processing among others.

iii) **Mobile Banking** – Advanced mobile banking applications provided by F1Soft allow users to do multiple banking activities using their smartphones. These applications have features such as fund transfer, balance inquiry, bill payment etc., thus making it more convenient for the user.

iv) **Internet Banking** – Secure internet banking platforms provided by the company ensures that customers have a user-friendly online banking experience. This allows them to manage their finance transfer money pay bills from home comforts or while at work through different services available on these platforms.

v) **Send or transfer money**: F1Soft also has experience with money transfer systems. They make it easy to send funds between countries by streamlining the process for their clients who want to do so quickly and securely while saving on costs too!

vi) **For businesses**: F1Soft offers many other types of technology products besides just those related to finance such as app development tools etc., through its Enterprise Solutions arm catering for all different industries' needs ranging from software engineering through systems integration right up until consultancy service provision around Information Technology.

2.4. Description of Intern Department

During my internship at F1Soft International Pvt. Ltd., I was placed in the dynamic DevOps department, which plays a crucial role in the company's IT infrastructure and operations. The DevOps team is responsible for ensuring seamless integration and deployment processes, enabling continuous delivery and integration (CI/CD) of applications. This involves managing infrastructure automation, monitoring system performance, and enhancing deployment efficiency through streamlined processes and tools. Each team within the department is led by a dedicated team lead who oversees operations and delegates responsibilities to team members. Under the guidance of System Support and Deployment Department, the department fosters a collaborative and energetic environment that enables its teams to deliver exceptional results.

As a DevOps intern, I had the opportunity to work under the guidance of my mentors, Sagar Khatiwada and Avash Shakya who provided invaluable assistance throughout my tenure. My responsibilities included assisting in the setup and maintenance of CI/CD pipelines, working with tools like Docker, Kubernetes, and Ansible for infrastructure automation, and implementing monitoring tools to track system performance. Additionally, I wrote scripts to automate routine tasks, improving overall efficiency in deployment and maintenance processes. This hands-on experience in DevOps practices, coupled with the support and mentorship from my team, significantly enhanced my technical skills and prepared me for a future career in the DevOps field. The collaborative and energetic environment at F1Soft allowed me to develop professionally and contribute effectively to the team's objectives.

2.5. Literature Review

The adoption of DevOps practices has significantly transformed the software development and IT operations landscape, promoting a culture of collaboration, continuous integration, and automation. DevOps culture thrives on the breaking down of walls between development and operations teams thus enabling faster and more reliable software releases (Len Bass, 2015). This kind of transformation is supported by a shift towards this culture which is fostered by processes and tools of automation where quality can be delivered at speed without sacrificing stability of operations or efficiency in running such systems within an organization.

Continuous Integration/Continuous Deployment (CI/CD) is one such central pillar among other things that make up DevOps (Farley, 2015). CI/CD pipelines automate integration testing deployment, speeding up production cycles through reduction of manual labour errors and general slowness associated with them thus ultimately boosting overall productivity levels within development teams. Also, it sets a ground for receiving quick responses from clients during different stages (feedback loops) because developers can detect any problem at an early stage before proceeding further.

Additionally, you will note that if we put in place DevOps practices then system monitoring and incident management become easier than ever before. There are continuous monitoring tools such as Prometheus, Grafana or ELK stack (Elasticsearch, Logstash, Kibana) among others which offer visibility into the performance and health status of a system real time (Ebert et al., 2016). Through them organizations can easily find anomalies proactively as well as respond quickly when incidents occur so as to improve reliability while reducing downtime for those depending on these systems most times in businesses world wide. More still, an effective monitoring combined with logging forms strong pillars towards achieving success through ensuring high availability levels & performances are maintained always within any given environment setting under consideration taking cognizance that downtime may translate into huge losses especially financially or even worse loss of lives due failure deliver mission critical services.

Chapter 3: Internship Activities

3.1. Roles and Responsibilities

While working as a DevOps Engineer intern for F1Soft International, my main focus was on bringing together software development and IT operations. I had the following tasks:

i) **CI/CD Pipeline Implementation**:
I automated software build, test, and deployment processes by setting up Jenkins CI/CD pipelines.

ii) **Infrastructure Management**:
I designed infrastructure solutions that could be scaled using physical servers.

iii) **Monitoring and Alerting**:
To ensure system reliability and performance, I implemented monitoring tools such as Grafana, Prometheus, Uptime Kuma as well as set up alerting systems.

iv) **Containerization**:
I utilized Kubernetes (K3S) for managing Docker containers which were orchestrated using Rancher.

v) **Automation and Scripting**:
My duties also involved writing Bash scripts that would automate routine tasks like deployment processes' execution or system checks for anomalies detection.

vi) **Documentation & Reporting**:
I kept records of every procedure undertaken along with their configurations before finally compiling performance reports at the end of each month.

vii) **Collaboration**:
Additionally, I worked hand in hand with developers, sysadmins and other team players so as to smoothen integration points between development and deployment workflows

viii) **Continuous learning**:
Staying updated with industry trends and applying new knowledge to improve existing systems.

3.2. Weekly log

Following table shows the weekly activities the intern performed throughout their internship period.

Table 3.1 : Weekly Log

Week	Activities

Week 1	• Orientation • Installed CentOS with automatic as well as manual partition and got familiar with Linux commands • Learned hosting HTML, CSS on nginx and tomcat webserver • Setup nginx for reverse proxy and load balancing
Week 2	• Explored about HAProxy for load balancing • Configured firewalld in CentOS for allowing traffic • Got familiar with SELinux policies • Hosted tomcat website with Dockerfile • Learned basic commands used in redis for retrieving and inserting data • Configured keepalived for loadbalancing and high-availability • Configured self hosted git server without GUI • Configured nginx hosted website with SSL certificate
Week 3	• Learned to create own Certificate Autority and generate SSL certificate for client • Setup mysql database • Installed and configured MinIO for managing unstructured object data • Installed monit for process monitoring and configured to send alert email • Hosted git service like github, gitlab • Configured uptime-kuma for monitoring services over HTTP/S, TCP, DNS and other protocols • Learned to create bash script for send email using curl • Achieved file synchronization between master with worker nodes using lsyncd • Learned about Network File System for sharing files between nodes within a network • Learned about traefik for reverse proxy and load balancing
Week 4	• Configured MySQL replication for data backup • Setup self-hosted registry for docker images i.e. Harbor • Configured jenkins to push to the self-hosted registry • Learned to create ansible playbooks • Configured grafana, loki and prometheus for log collection and their visualization • Studies about cryptography and PKI • Setup k3s with rancher to manage kubernetes cluster
Week 5	• Deploy website in nginx using k3s pod • Learned about kubernetes concepts • Deployed sidecar contanier in k3s

	• Learned to manage LVM partition of linux system • Deployed a website synchronized with git repo • Learned about awk and sed command
Week 6	• Configured k3s deployed website with SSL certificate using k3s TLS Secrets • Learned about ArgoCD for continuous delivery

3.3. Description of the Project(s) Involved During Internship

One of the highlights which really helped me in understanding DevOps principles was working on two minor projects during my internship both using local infrastructure.

Project 1: Setting up Jenkins and Self-Hosted Docker Registry

In this project, my main goal was to automate the build and deployment processes by using Jenkins as well as a self-hosted docker registry.

i) **Under Jenkins Configuration**: I installed Jenkins on a local server then did its setup for managing CI/CD pipelines which were continuous integration and continuous deployment. This involved creating Jenkins jobs that would automate building and testing phases of application development.

ii) **Setting Up Self-Hosted Docker Registry**: I established an internal docker registry within a server environment so that we could host our docker images locally instead of relying on external registries where we have less control over them or their security levels.

iii) **Image Push To Registry**: Thereafter, I configured jenkins pipelines; these would pull source codes from different repositories then build corresponding docker images before pushing these artifacts into my own registry which I had set up before hand thus ensuring all necessary files for deployment are stored in one place for both security

The result was a complete automation of CI/CD pipeline which eliminated manual intervention, reduced errors and made deployment faster within our premises.

Project 2: Using git-sync with Nginx Deployed In K3S

During this project work centered around syncing websites hosted on self hosted git services (Gitea/ Gogs) with nginx servers deployed inside K3s clusters as well adding SSL certificates deployment along side Mysql replication also .

i) **Git-sync Implementation**: Configured git-sync within the local K3s environment to automatically pull the latest changes from the Git repository (hosted on a local Gitea or Gogs instance). This ensured that the website content was always up-to-date with the latest code changes.

ii) **Nginx Deployment**: Deployed Nginx as a reverse proxy in the local K3s cluster to serve the website. Configured Nginx to serve the synchronized content from the Git repository.

iii) **SSL Certificates**: Utilized Kubernetes Ingress with TLS secrets to implement SSL certificates for secure HTTPS access to the website. This was done using Let's Encrypt to automatically issue and renew certificates.

iv) **Monitoring and Logging**: Implemented monitoring and logging solutions using Grafana, Prometheus, and Loki. These tools were configured to monitor the performance and health of the applications, collect logs, and visualize metrics, enabling proactive management and troubleshooting of the deployed applications.

The project demonstrated a robust, automated deployment process with real-time synchronization, secure access, and reliable data replication, all managed within the local infrastructure.

These projects provided hands-on experience with several key DevOps tools and practices, such as Jenkins, Docker, K3s, git-sync, Nginx, SSL certificates. The practical knowledge gained from these projects significantly enhanced my skills in automating, managing, and securing deployments in a production environment, all within the scope of local infrastructure.

3.4. Tasks / Activities Performed

i) Pushing docker image to self hosted container registry

In this task, Docker images have been pushed to self hosted container registry using self hosted jenkins by running shell script. Given steps were being followed to do the task:

- Install Jenkins and it's dependencies using following shell script:

File: Jenkins-install.sh

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat/jenkins.
repo
sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
sudo dnf upgrade
# Add required dependencies for the jenkins package
sudo dnf install fontconfig java-17-openjdk
sudo dnf install jenkins
sudo systemctl enable jenkins
sudo systemctl start jenkins

JENKINS_PORT=8080
PERM="--permanent"
SERV="$PERM --service=jenkins"

firewall-cmd $PERM --new-service=jenkins
firewall-cmd $SERV --set-short="Jenkins ports"
```

```
firewall-cmd $SERV --set-description="Jenkins port exceptions"
firewall-cmd $SERV --add-port=$JENKINS_PORT/tcp
firewall-cmd $PERM --add-service=jenkins
firewall-cmd --zone=public --add-service=http --permanent
firewall-cmd --reload
```

- Install and configure Harbor which is self hosted registry using docker with following script and create a new project from Harbor Dashboard.

File: Harbor-install-and-configure.sh

```
mkdir -p harbor-setup; cd harbor-setup
wget https://github.com/goharbor/harbor/releases/download/v2.11.0/harbor-online-installer-v2.11.0.tgz
tar xvf harbor-online-installer-v2.11.0.tgz
openssl genrsa -out ca.key 4096

echo -e "\n------------\n Creating Root CA Certificate \n-----------------\n "
openssl req -x509 -new -nodes -sha512 -days 3650 \
  -subj  "/C=NP/ST=Bagmati  Pradesh/L=Kathmandu/O=MeroOrganization/OU=Personal/CN=merodomain.com" \
 -key ca.key \
 -out ca.crt

echo -e "\n------------\n Generating Server Certificate \n-----------------\n"
openssl genrsa -out meroharbor.com.key 4096
openssl req -sha512 -new \
      -subj  "/C=NP/ST=Mero  State/L=Mero  Thau/O=Mero  Organization/OU=Personal/CN=meroharbor.com" \
    -key meroharbor.com.key \
    -out meroharbor.com.csr

cat > meroharbor.com.v3.ext <<-EOF
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
[alt_names]
DNS.1=meroharbor.com
EOF

echo  -e  "\n------------\n  Signing  Server  Certificate  with  CA's  Certificate \n-----------------\n"
```

```
openssl x509 -req -sha512 -days 3650 \
    -extfile meroharbor.com.v3.ext \
    -CA ca.crt -CAkey ca.key -CAcreateserial \
    -in meroharbor.com.csr \
    -out meroharbor.com.crt

echo -e "\n------------\n Converting crt to cert \n-----------------\n"
openssl x509 -inform PEM -in meroharbor.com.crt -out meroharbor.com.cert
mkdir -p /etc/docker/certs.d/meroharbor.com:443/
cp meroharbor.com.cert /etc/docker/certs.d/meroharbor.com:443/
cp meroharbor.com.key /etc/docker/certs.d/meroharbor.com:443/
cp ca.crt /etc/docker/certs.d/meroharbor.com:443/
cd harbor; cp harbor.yml.tmpl harbor.yml
sed    -i    's|hostname:    reg.mydomain.com|hostname:    meroharbor.com|;
s|certificate:    /your/certificate/path|certificate:    /etc/docker/certs.d/
meroharbor.com:443/meroharbor.com.cert|;   s|private_key:   /your/private/key/
path|private_key: /etc/docker/certs.d/meroharbor.com:443/meroharbor.com.key|'
harbor.yml
./prepare
docker compose up -d
```

Following picture provide the output of the above shell script:

Figure 3.1 : Harbor Installation

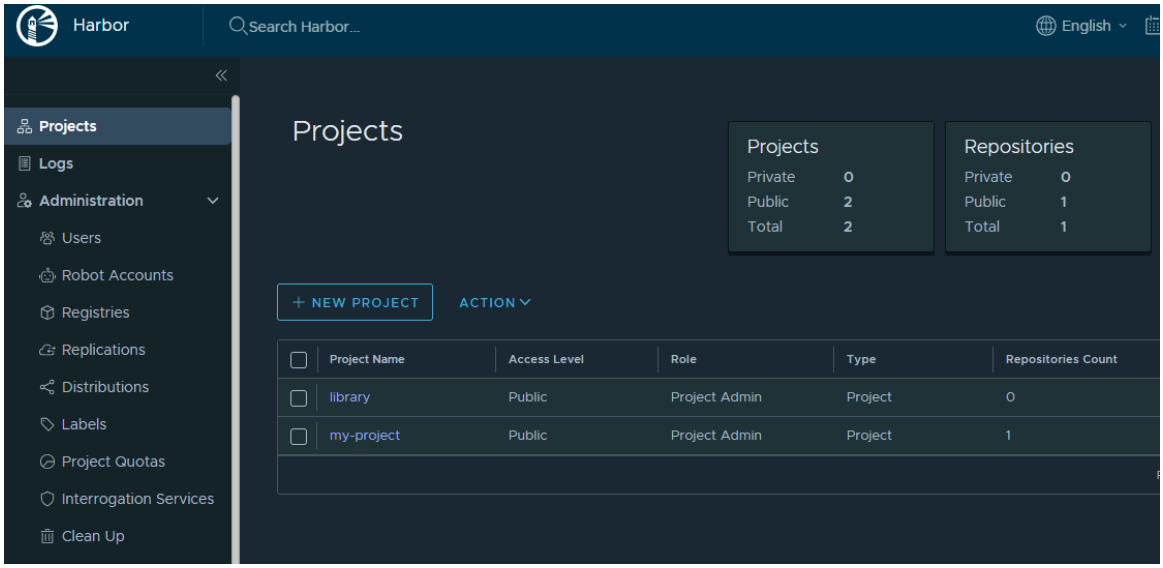


Figure 3.2 : Harbor Project Creation

- Create Dockerfile to build docker image

File: Dockefile

```
FROM nginx:latest
COPY ./mysite /usr/share/nginx/html/
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

Step 4: Create a Freestyle Project in Jenkins

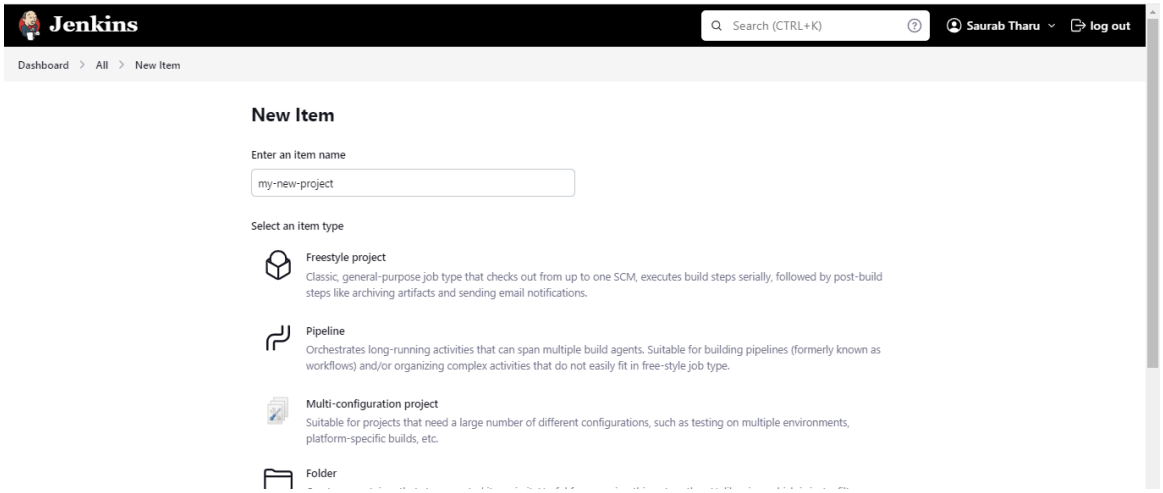


Figure 3.3 : Creating Jenkins Freestyle Project

- Configure the jenkins project by adding **"Exectue shell"** build step and use the following shell script and save the configuration.

```
DOCKER_REGISTRY='meroharbor.com:443'
DOCKER_REPO='my-project'
IMAGE_NAME='my_web'
TAG="build-${BUILD_NUMBER}"

cd /var/lib/jenkins/workspace/MyWeb-DockerFile/
ls -al

docker build -t ${DOCKER_REGISTRY}/${DOCKER_REPO}/${IMAGE_NAME}:${TAG} .

# here password to login is stored in some.txt file
cat some.txt | docker login ${DOCKER_REGISTRY} -u saurab --password-stdin

docker push ${DOCKER_REGISTRY}/${DOCKER_REPO}/${IMAGE_NAME}:${TAG}
docker logout
```

- Finally build the project and check the **"Console Output"** to verify if build was successful or not.

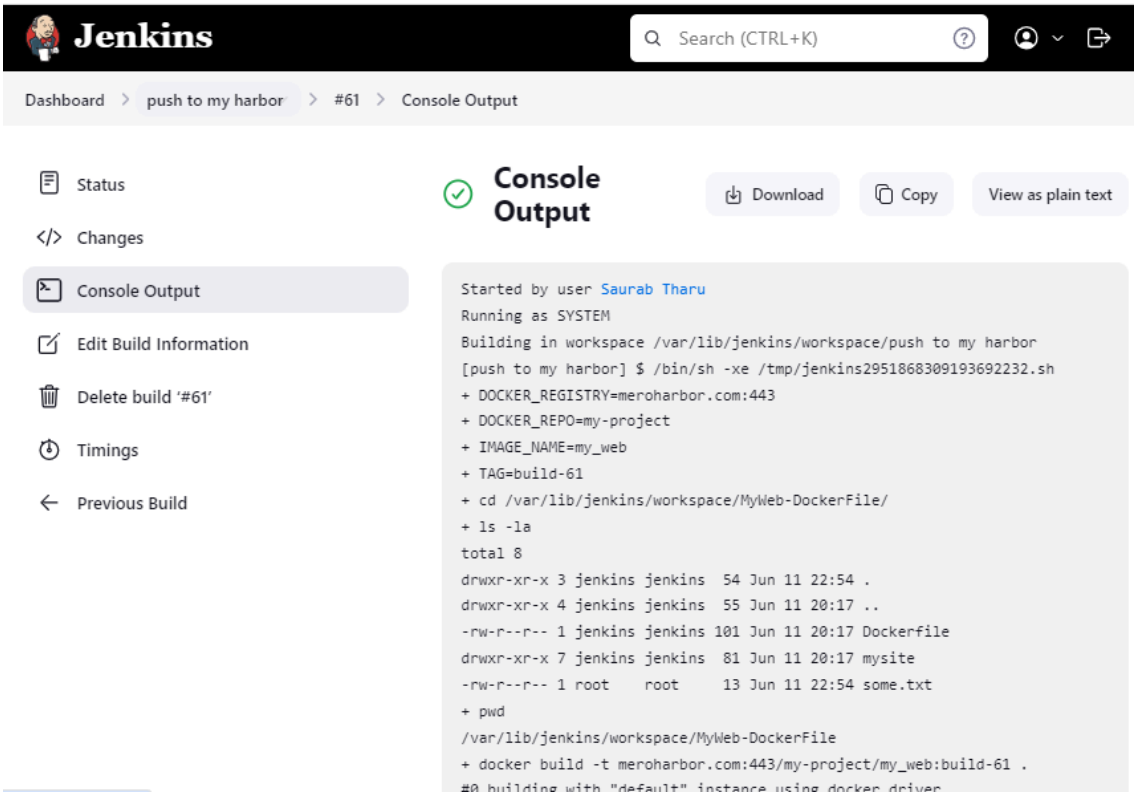


Figure 3.4 : Jenkins Console Output

ii) Dockerizing nginx hosted website and providing SSL certificate to the website

Hosting a website using Nginx and securing it with an SSL certificate is one of the fundamental task DevOps needs to understand. This involved preparing a Dockerfile and configuring the Nginx server to serve the website content and setting up virtual hosts. To ensure secure communication, SSL certificates were generated and installed. openssl tool is used for generating certificates which includes certificate.crt file and private key .key file. Following files were used for dockerization of nginx website:

File: generate_certificate.sh

```
#!/bin/bash

set -x
openssl genrsa -out ca.key 4096

echo -e "\n------------\n Creating Root CA Certificate \n----------------\n "
openssl req -x509 -new -nodes -sha512 -days 3650 \
  -subj  "/C=NP/ST=Bagmati  Pradesh/L=Kathmandu/O=MeroOrganization/OU=Personal/
CN=meroCA.com" \
 -key ca.key \
 -out ca.crt

echo -e "\n------------\n Generating Server Certificate \n----------------\n"
openssl genrsa -out merowebsite.com.key 4096
openssl req -sha512 -new \
      -subj  "/C=NP/ST=Mero  State/L=Mero  Thau/O=Mero  Organization/OU=Personal/
CN=merowebsite.com" \
    -key merowebsite.com.key \
    -out merowebsite.com.csr

cat > merowebsite.com.v3.ext <<-EOF
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

[alt_names]
DNS.1=merowebsite.com
EOF
```

```
echo -e "\n------------\n Signing Server Certificate with CA's Certificate
\n-----------------\n"
openssl x509 -req -sha512 -days 3650 \
    -extfile merowebsite.com.v3.ext \
    -CA ca.crt -CAkey ca.key -CAcreateserial \
    -in merowebsite.com.csr \
    -out merowebsite.com.crt
```

File: nginx.conf

```
server {
    listen       443;
    server_name  merowebsite.com;

    ssl on;
    ssl_certificate     /usr/share/nginx/certificates/merowebsite.com.crt;
    ssl_certificate_key /usr/share/nginx/certificates/merowebsite.com.key;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
}
```

File: Dockerfile

```
FROM nginx:1.10.1-alpine
COPY jd /usr/share/nginx/html
COPY merowebsite.com.crt /usr/share/nginx/certificates/
COPY merowebsite.com.key /usr/share/nginx/certificates/
COPY nginx.conf /etc/nginx/conf.d/default.conf
EXPOSE 443
CMD ["nginx", "-g", "daemon off;"]
```

iii) Configuring up load balancing

Suppose we have four website hosted using nginx at port 1111, 2222, 3333, 4444 with following nginx configuration

File: website-for-load-balancing.conf

```
##############################################################
#   This is for loadbalancing
##############################################################

# localhost:1111
server {
    listen       1111;
    server_name  localhost;
    root         /usr/share/nginx/digital-trend;
    index  index.html index.htm
}

# localhost:2222
server {
    listen       2222;
    server_name  localhost;
    root         /usr/share/nginx/jackpiro;
    index  index.html index.htm
}

# localhost:3333
server {
    listen       3333;
    server_name  localhost;
    root         /usr/share/nginx/memorial;
    index  index.html index.htm
}

# localhost:4444
server {
    listen       4444;
    server_name  localhost;
    root         /usr/share/nginx/shiphile;
    index  index.html index.htm
}
```

a) Using Nginx

The following Nginx configuration can be used to set up a load balancer that distributes traffic across multiple backend servers. This setup ensures that requests are balanced between the servers specified in the upstream block.

File: haproxy.cfg

```
upstream myapp1 {
    server localhost:1111;
    server localhost:2222;
    server localhost:3333;
    server localhost:4444;
}
server {
    listen 80;
    server_name nginx-load-balancer.com;              # domain-name
    location / {
        proxy_pass http://myapp1;
    }
}
```

b) Using HAProxy

The following configuration can be used for HAProxy to load balance traffic across multiple nodes, ensuring that if any node goes down, the other nodes will continue to serve the website. This setup distributes incoming traffic using the round-robin algorithm and checks the health of each node to maintain high availability.

File: haproxy.cfg

```
frontend main
    bind *:80
    timeout client 60s
    mode http
    default_backend backend_name # <- name of backend specified below
 backend backend_name
    timeout connect 30s
    timeout server 100s
    mode http
    balance roundrobin
    http-request set-header Host load-balanced-website.com
    server node1  127.0.0.1:1111  check
    server node2  127.0.0.1:2222  check
    server node3  127.0.0.1:3333  check
    server node4  127.0.0.1:4444  check
```

v) Configuring k3s for synchronizing deployed website with code in Gitea

K3s, a lightweight Kubernetes distribution, was used for managing containerized applications efficiently. Synchronizing a deployed website with a GitHub repository ensures continuous integration and deployment, allowing the website to reflect the latest code changes automatically.At first setting up the environment involves installing k3s, a lightweight Kubernetes distribution that simplifies cluster setup and management. By running a single command, we install k3s and initiate the server. Configuring `kubectl` ensures that we can manage the k3s cluster efficiently, setting the stage for deploying applications. This streamlined setup process demonstrates k3s's ease of use and quick deployment capabilities.

To install k3s, execute the following command in your terminal. This command downloads and installs the k3s binary and starts the k3s server.

```
$ curl -sfL https://get.k3s.io | sh -
```

Deploying a sample website on k3s involves creating a deployment configuration file (`deployment.yaml`) that defines the desired state of the application, including the number of replicas, container images, and service specifications. Applying this configuration with `kubectl` deploys the website on the k3s cluster.

File: nginx-deploymentgit.yaml

```yaml
apiapiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy-resturan
  namespace: nginx
  labels:
    app: nginx-cc
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-cc
  template:
    metadata:
      labels:
        app: nginx-cc
    spec:
      containers:
      - name: git-sync
        image: registry.k8s.io/git-sync/git-sync:v4.2.3
```

```
        volumeMounts:
        - name: www-data
          mountPath: /data
        env:
        - name: GITSYNC_REPO
          value: "http://192.168.33.11:3000/saurab/website-for-git-sync.git"
        - name: GIT_SYNC_BRANCH
          value: "main"
        - name: GITSYNC_ROOT
          value: /data
        - name: GIT_SYNC_DEST
          value: "html"
        - name: GITSYNC_ONE_TIME
          value: "false"
        - name: GIT_SYNC_COMMAND_AFTER
          value: "nginx -s reload"
        securityContext:
          runAsUser: 0
      - name: nginx-resturan
        image: nginx
        ports:
        - containerPort: 80
        volumeMounts:
        - mountPath: "/usr/share/nginx/"
          name: www-data
      volumes:
      - name: www-data
        emptyDir: {}
```

This `nginx-ingress.yaml` file defines an Ingress resource named nginx-ingress using the Traefik controller, with TLS termination via tls-secret for roltu.com. It routes all HTTP requests to roltu.com to the nginx-service on port 80.

File: ingnx-ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
  annotations:
    kubernetes.io/ingress.class: "traefik"
```

```yaml
spec:
  tls:
    - secretName: tls-secret
      hosts:
        - "roltu.com"
  rules:
    - host: roltu.com
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: nginx-service
                port:
                  number: 80
```

This `nginx-service.yaml` defines a Kubernetes `Service` named `nginx-service` in the `nginx` namespace. It uses a `NodePort` to expose port 80, forwarding traffic from the service on port 80 to the pods labeled `app: nginx-cc` on port 80, accessible via node IP on port 30002.

File: nginx-namespace.yaml

```yaml
apiVersion: v1
kind: Namespace
metadata:
    name: nginx
```

File: nginx-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: nginx
spec:
  type: NodePort
  selector:
    app: nginx-cc
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30002
```

Apply the configuration to the k3s cluster using kubectl using commands as below:

```
$ kubectl apply -f nginx-namespace.yaml
$ kubectl apply -f nginx-deploymentgit.yaml
$ kubectl apply -f nginx-ingress.yaml
$ kubectl apply -f nginx-service.yaml
```

The above deployment file to work we need a git repo so for that self hosted Gitea was used. To install the Gitea following bash script is used:

File: gitea-installation.sh

```
  #!/bin/bash
# Create a directory and navigate into it
mkdir gitea
cd gitea
wget -O gitea https://dl.gitea.com/gitea/1.22.0/gitea-1.22.0-linux-amd64
chmod +x gitea
useradd git
groupadd gitservice
chown -R root:gitservice gitea
chmod g+s gitea
mkdir -p /usr/local/bin/gitea
mv gitea /usr/local/bin/gitea

cat > /etc/systemd/system/gitea.service <<-EOF
[Unit]
Description=Gitea (Git with a cup of tea)
After=syslog.target
After=network.target
#After=mysqld.service
#After=postgresql.service
#After=memcached.service
#After=redis.service

[Service]
# Modify these two values and uncomment them if you have
# repos with lots of files and get an HTTP error 500 because
# of that
#LimitMEMLOCK=infinity
#LimitNOFILE=65535
RestartSec=2s
Type=simple
```

```
User=git
Group=gitservice
WorkingDirectory=/usr/local/bin/gitea
ExecStart=/usr/local/bin/gitea/gitea web
Restart=always
Environment=USER=git HOME=/home/git
# If you want to bind Gitea to a port below 1024 uncomment
# the two values below
#CapabilityBoundingSet=CAP_NET_BIND_SERVICE
#AmbientCapabilities=CAP_NET_BIND_SERVICE

[Install]
WantedBy=multi-user.target
EOF

systemctl start gitea.service
systemctl enable gitea.service
```

After installation, the gitea can be accessed using ipaddress along with the port as below:

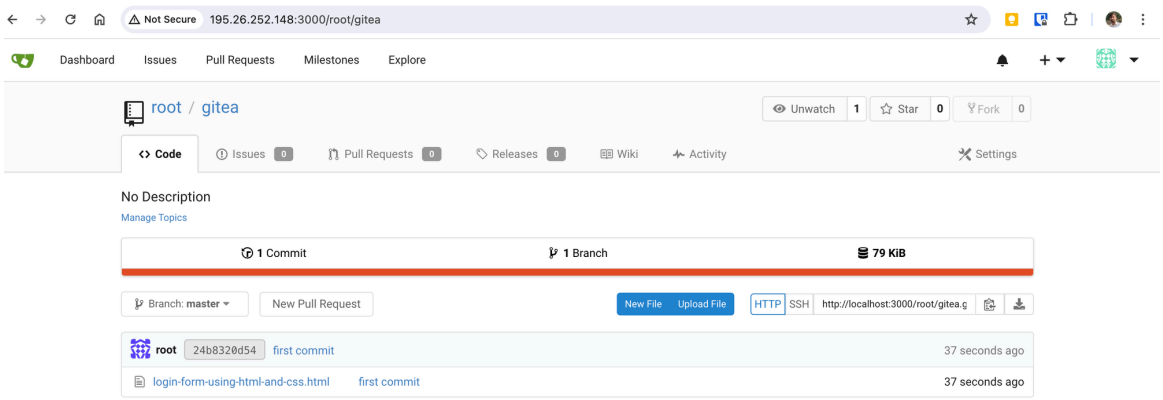


Figure 3.5 : Git Repo Locally Hosted

Chapter 4: Conclusion and Learning Outcomes

4.1. Conclusion

In conclusion, the internship at F1Soft International as a DevOps Engineer exposed me to various aspects of DevOps. This included knowledge gain in diverse DevOps areas like deployment of website with SSL/TLS certificate, Virtual Machine networking, containerization and utilizing Kubernetes (K3s) for container orchestration, implementation of monitoring tools such as Prometheus and Grafana.

Practical task were done using theoretical knowledge acquired from the program. At the end of it all, adopting different DevOps methodologies that were learnt brought about better system performance and reliability. With the help of mentors and other colleagues, my interpersonal skills were also sharpened during this internshipr period henceforth having a huge positive influence on me both professionally and personally.

4.2. Learning Outcome

Here are the key areas where I gained substantial knowledge and practical experience:

i. **Technical Skills**

   Developed my technical skills like operating system troubleshooting, managing bare-metal infrastructure, automating repeated tasks with bash script. Got opportunity to dig deep into using linux command line as well as monitoring tools like monit, prometheus, and grafana. Learned about monitoring various process of linux and send email alert if needed. Along with this various other tools like ansible, used for DevOps were learned.

ii. **Professional Development**

   Enhanced the ability to work effectively in a team-oriented environment. Problem-solving skills were improved by addressing real-world technical challenges and implementing practical solutions. Professional growth was further shaped by the mentorship and guidance received from experienced professionals.

iii. **Time Management**

   Effective time management was crucial during the internship, as multiple tasks were assigned simultaneously. Work was prioritized, achievable goals were set, and deadlines were consistently met.

iv. **Documentation**

   The importance of thorough documentation was emphasized throughout the internship. Detailed records of configurations, processes, and performance metrics were maintained, ensuring transparency and reproducibility. Comprehensive reports and documentation were compiled to communicate project progress, improving technical writing and communication skills.

v. **Continuous Learning**

A habit of continuous learning was cultivated during the internship, encouraging the intern to stay updated with the latest industry trends and advancements. Self-directed learning was regularly engaged in, exploring new tools and technologies to enhance existing systems and processes

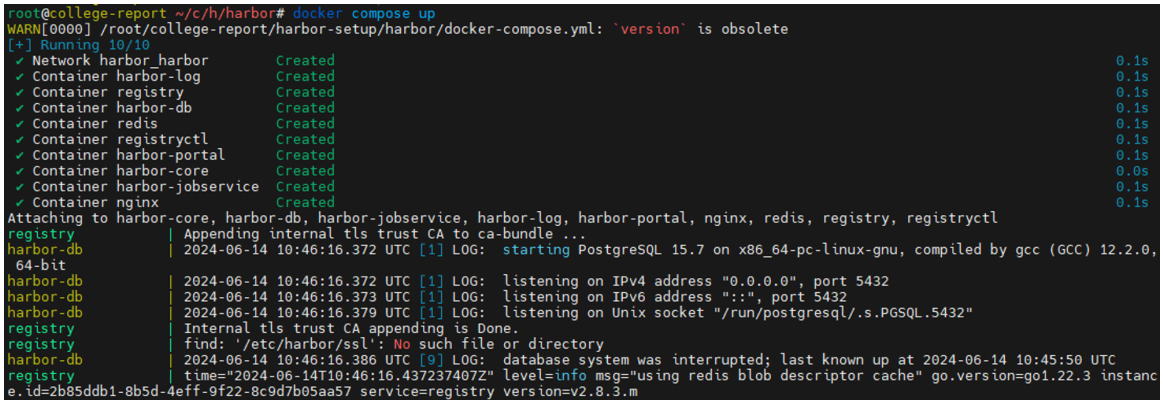
References

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software*, *33*(3), 94–100. https://doi.org/10.1109/MS.2016.68

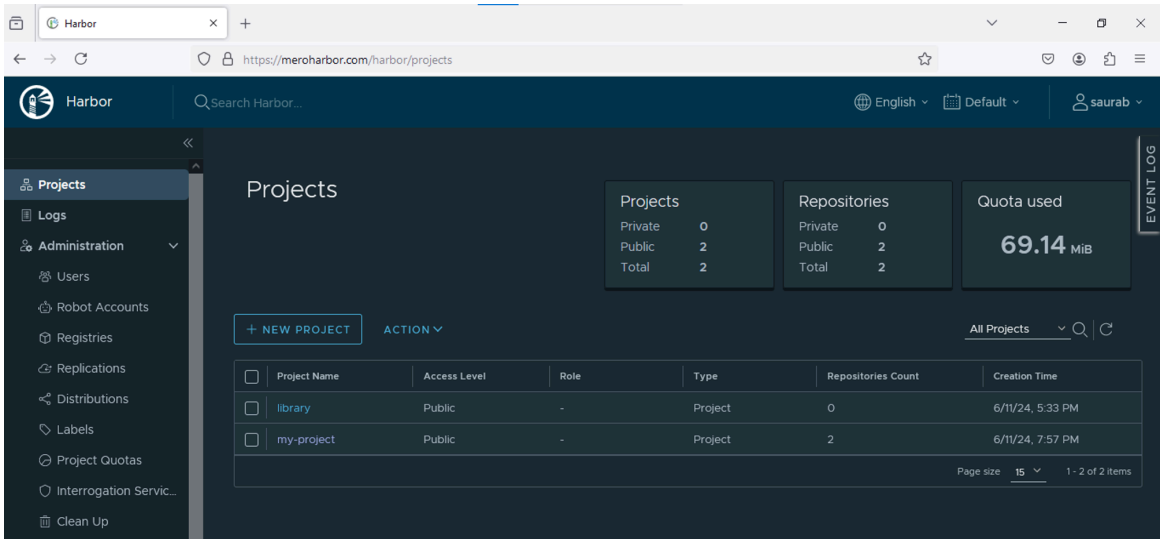
Farley, J., David;Humble. (2015). *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation)* (Tenth printing, Vol. 0). Addison-Wesley Professional.

Len Bass, L. Z., Ingo Weber. (2015). *DevOps: A Software Architect's Perspective* (1st ed., Vol. 0). Addison-Wesley Professional.

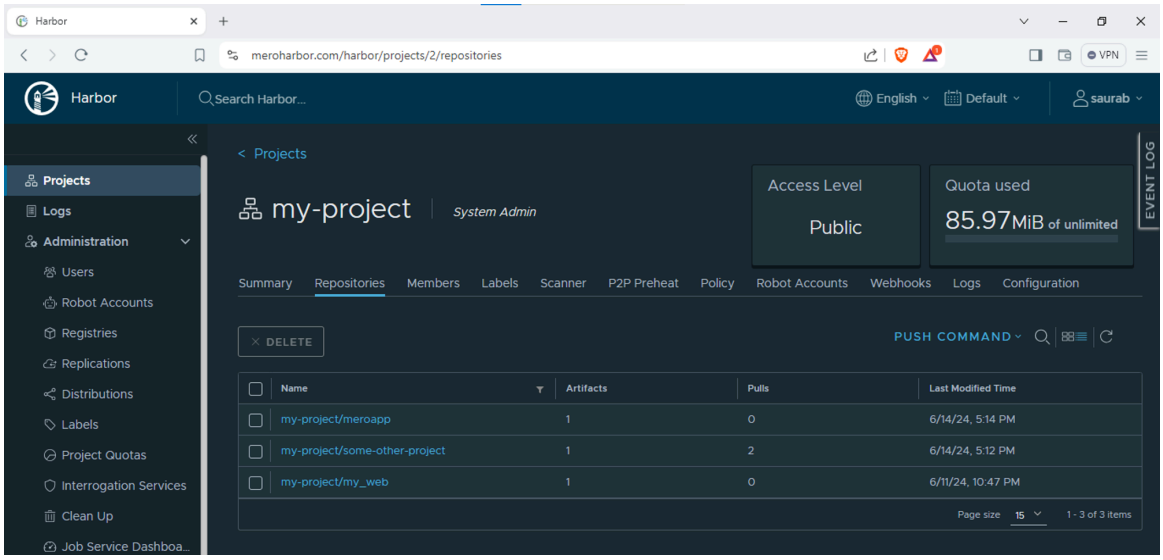
Appendices



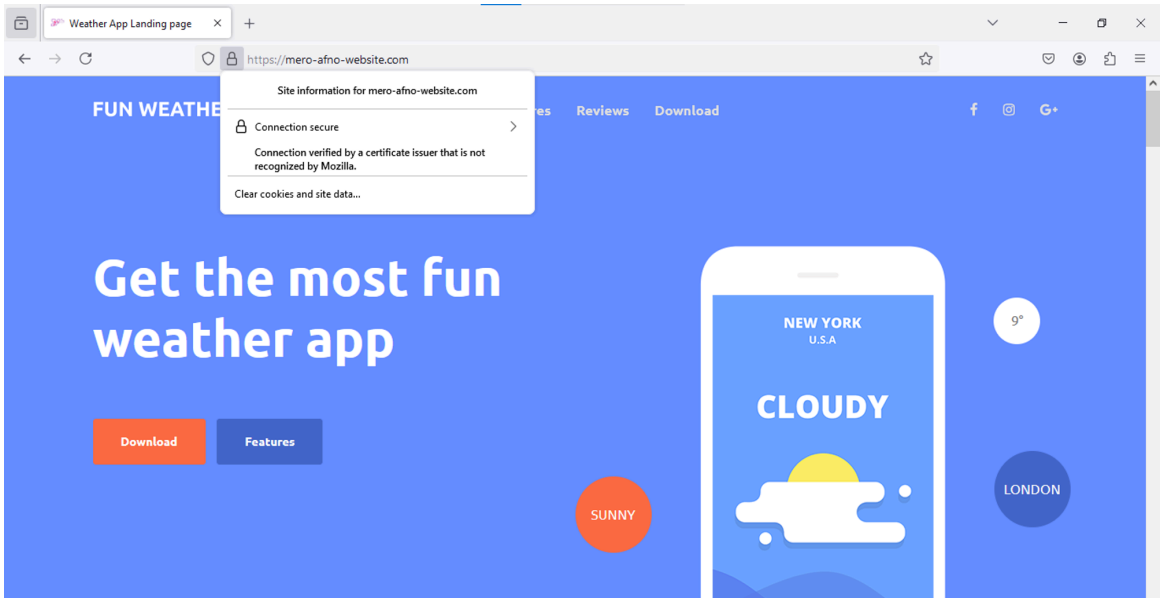
Initializing Locally Hosted Container Registry



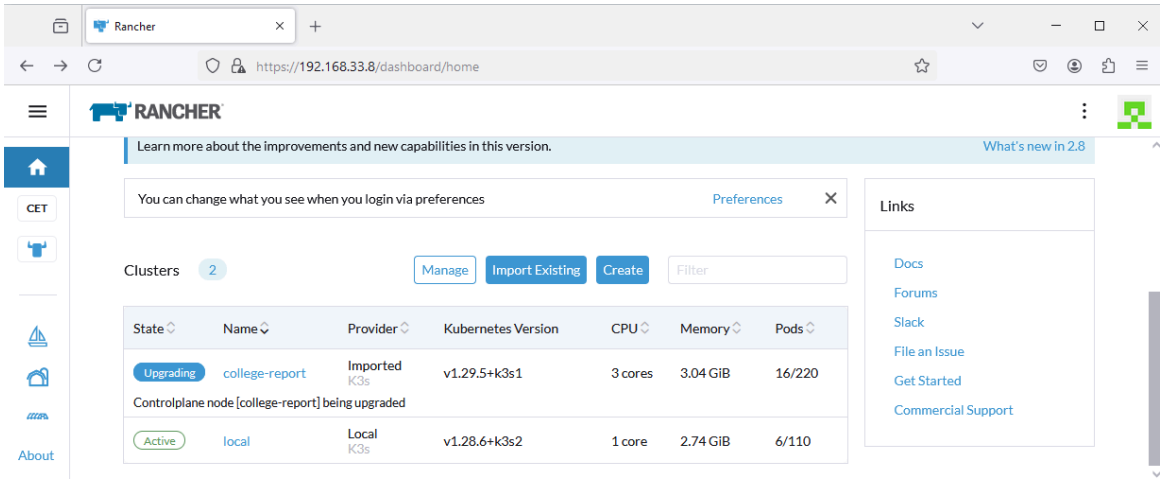
Dashboard of Locally Hosted Container Registry



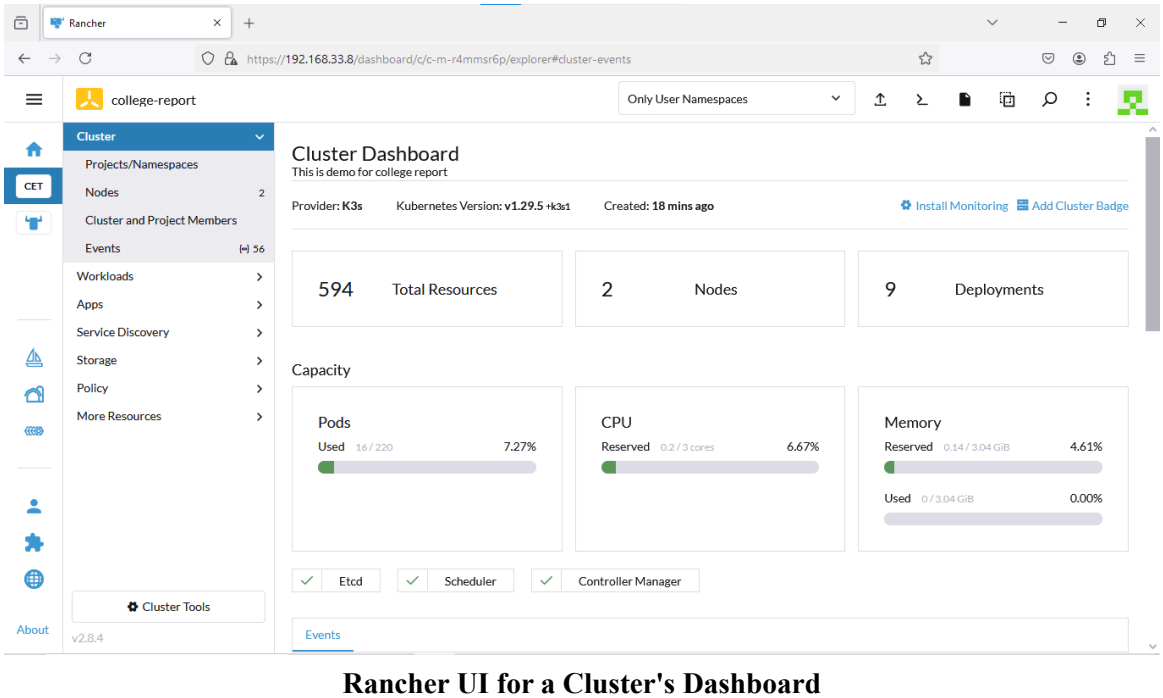
Example of Docker Images Pushed to the Locally Hosted Container Registry



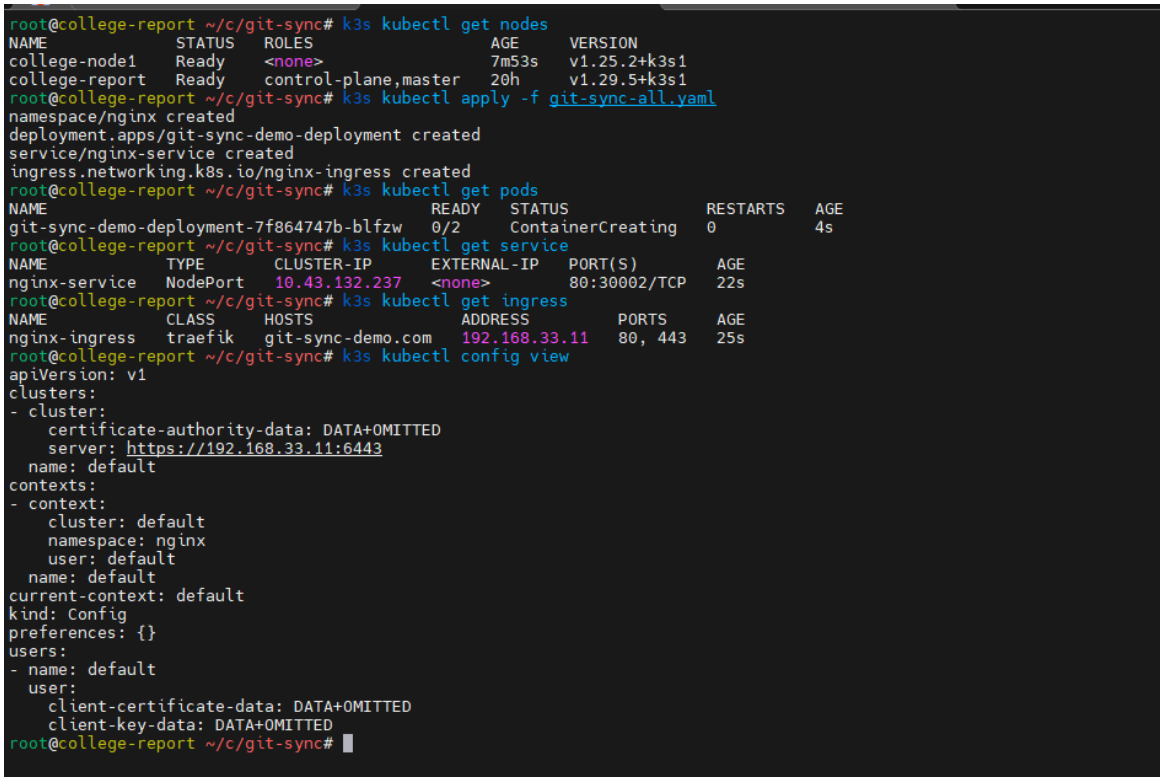
Provisioning of SSL Certificate to Website



Rancher Dashboard for K3S cluster mangement



Rancher UI for a Cluster's Dashboard



K3S Cluster Managenemtn using CLI