Name : Chinmay Chaudhari **Div : D15C** **Roll No : 6**

Automated Deployment with Monitoring

Introduction

This case study outlines the implementation of a Continuous Integration/Continuous Deployment (CI/CD) pipeline using Jenkins for a simple web application deployed on an Amazon EC2 instance. Additionally, we will set up Nagios for monitoring the application's availability, ensuring that any downtime or issues are promptly identified.

Problem Statement

The objective is to automate the deployment process of a web application, enhancing efficiency and reducing the potential for human error during deployment. Furthermore, it's essential to implement a monitoring solution to keep track of the application's health and performance.

Technologies Used

- **Jenkins**: An open-source automation server that facilitates CI/CD.
- **Amazon EC2**: A cloud computing service that provides scalable computing capacity.
- **Nagios**: A powerful monitoring system that enables organizations to identify and resolve IT infrastructure issues.
- **Vercel**: A platform for frontend developers, used here for analytics and monitoring the application's performance and usage.

Implementation Steps

1. **Setting Up the Environment**:
    - Provision an EC2 instance where the web application will be deployed.
    - Install necessary software and dependencies on the EC2 instance.

2. **Jenkins Configuration**:
    - Install Jenkins on a dedicated server or the EC2 instance itself.
    - Create a new pipeline job in Jenkins that will:
        - Clone the web application repository.
        - Build the application.
        - Deploy the built application to the EC2 instance using SSH.

3. **Nagios Installation and Configuration**:
    - Install Nagios on a separate server or the same EC2 instance.
    - Configure Nagios to monitor the HTTP status of the deployed web application.

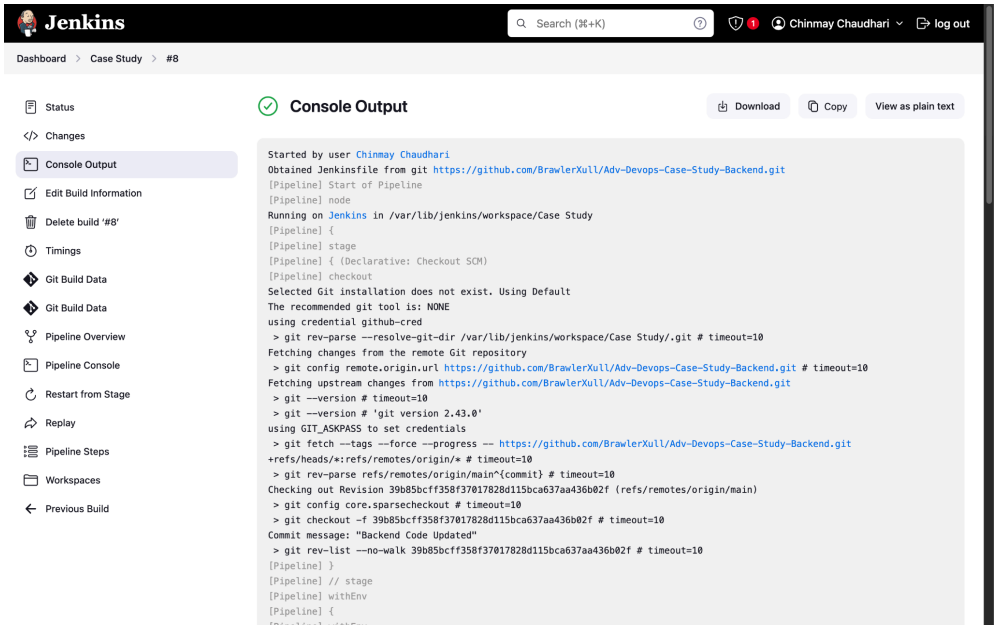
4. **Vercel Analytics Integration**:
    - Integrate Vercel for tracking user interactions and performance metrics.
    - Use the data gathered from Vercel to inform future deployments and enhancements.

5. **Testing the Pipeline**:
    - Trigger a build in Jenkins to deploy the application.
    - Verify the deployment by accessing the web application through a browser.

Name : Chinmay Chaudhari **Div : D15C** **Roll No : 6**

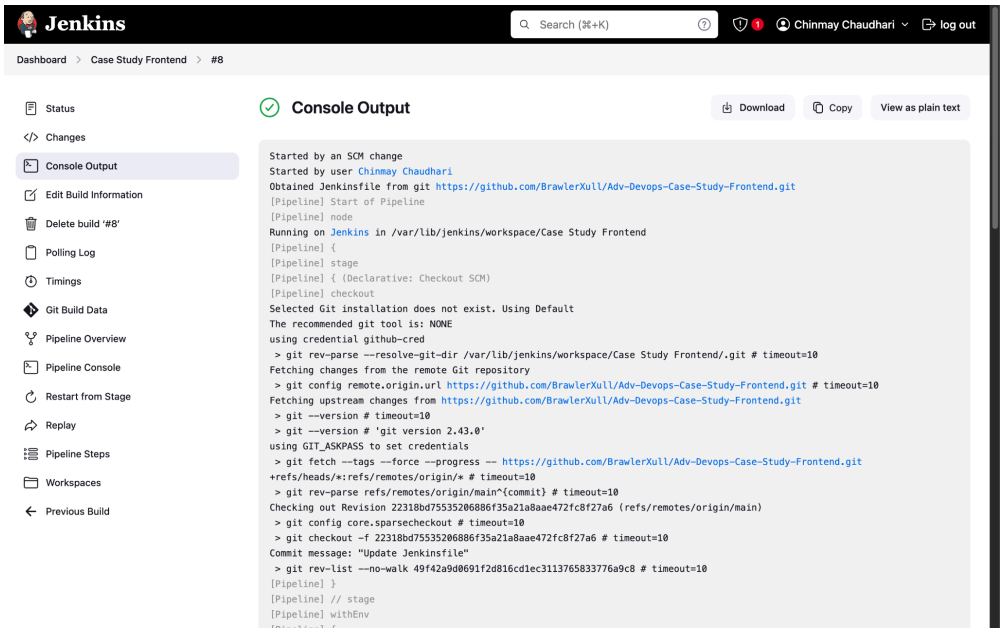
Part I - Backend

Setup a Jenkins Pipeline to automate the process of deployment of our web app's backend



Part II - Frontend

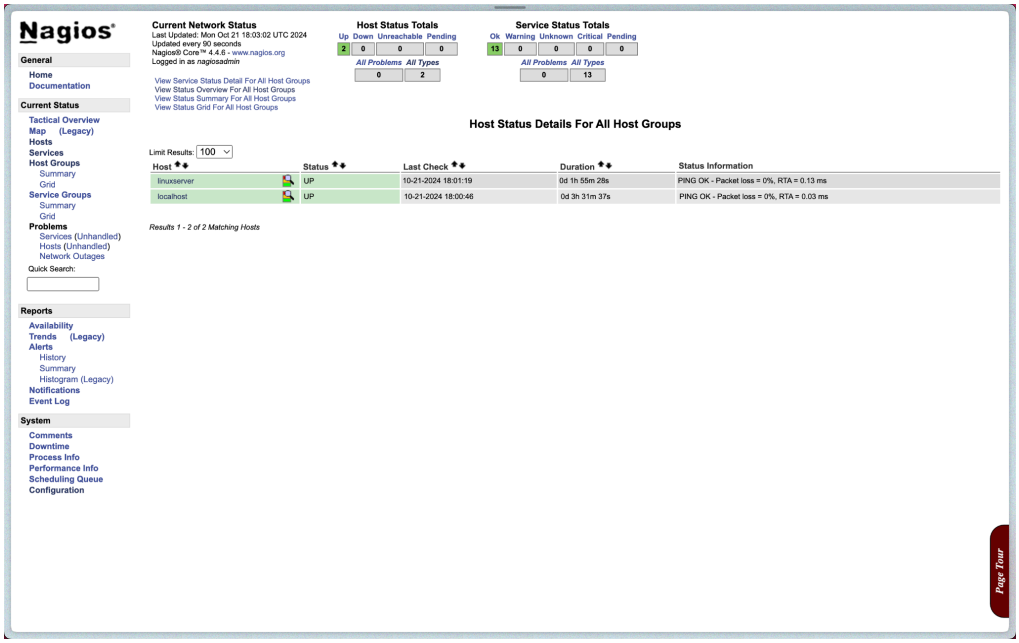
Setup a Jenkins Pipeline to automate the process of deployment of our web app's frontend.



Name : Chinmay Chaudhari **Div : D15C** **Roll No : 6**

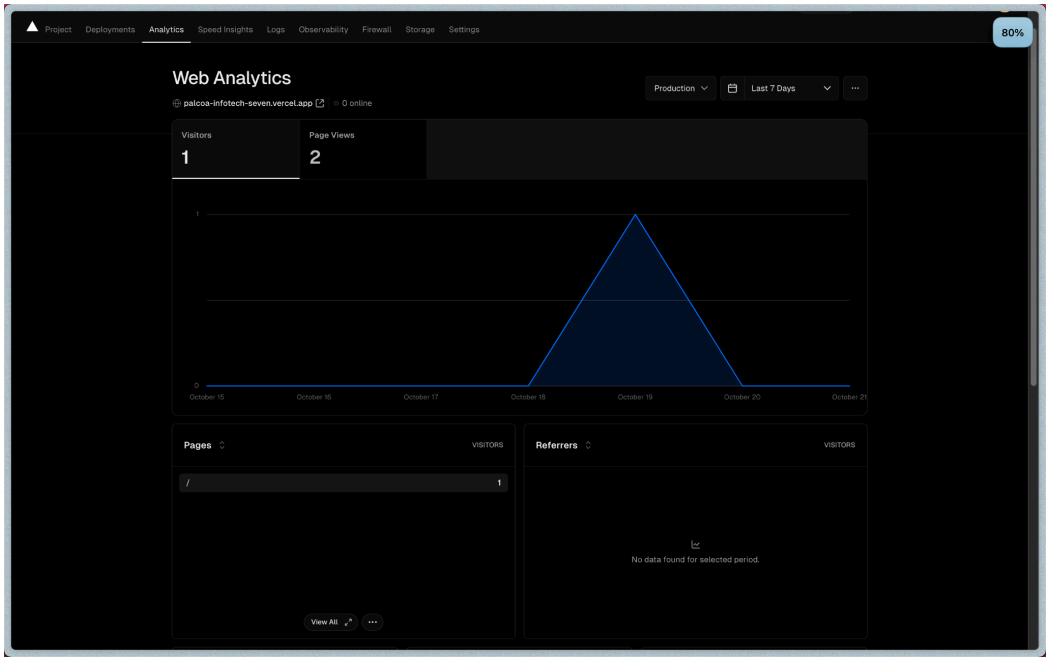
Part III - Nagios

Install Nagios and go to hosts section so see the running webapps



Part IV - Additional Integration (Mini Project)

Login to vercel using your github account and setup the analytics for our webapp.



Outcome

The successful implementation of this CI/CD pipeline results in a streamlined process for deploying web applications, reducing the time from development to production. The addition of Nagios monitoring provides immediate visibility into the application's status, allowing for proactive management of any issues.

Conclusion

In this following Experiment we learnt the below things

1. Setting up the Jenkins pipeline for deployment of web applications.
2. Using SCM Pipeline script while setting up the jenkins pipeline.
3. Integrating Nagios for Real Time monitoring of our web server.
4. Setting up custom config files for our Nagios project.
5. Integrating Vercel to view analytics of our frontend project.