DevOps

# Project Deployment Report

## Overview

The project aimed to deploy three applications using both Vagrant + Ansible and Dockerfile + Docker Compose, emphasizing Infrastructure as Code principles. The team utilized Git for version control, and the project was hosted on the GitHub repository named "dockerdemons2."

# Task 3: Vagrant + Ansible Deployment

## Project Structure

* Repository: [dockerdemons2](https://github.com/your-username/dockerdemons2)
* Branch: main

## Approach

* The team employed Vagrant and Ansible for Infrastructure as Code.
* Each team member handled one application setup.
* Advanced Ansible features such as roles, collections, and modules were leveraged for modularity.
* Code was merged into the main branch for seamless collaboration.

## Challenges Faced

1. **Network Configuration:** Configuring Vagrant with specific network setups posed challenges.
2. **Dependency Issues:** Ansible playbooks encountered issues related to application dependencies.

## Learning Notes

1. **Vagrant Multi-Machine:** Understanding the configuration of Vagrant for multi-machine setups.
2. **Ansible Modularity:** Exploring the organizational benefits of Ansible roles.
3. **Dependency Management:** Overcoming challenges in managing application dependencies with Ansible.

# Task 4: Dockerfile + Docker Compose Deployment

## Project Structure

* Repository: [dockerdemons2](https://github.com/your-username/dockerdemons2)
* Branch: main

## Approach

* Dockerfile and Docker Compose were employed for Infrastructure as Code.
* Dockerfiles were modified and optimized for compatibility.
* A modular code structure was maintained for continuous collaboration.

## Challenges Faced

1. **Networking Conflicts:** Docker Compose services faced conflicts on the same network.
2. **Dockerfile Debugging:** Debugging Dockerfile for application3 due to unexpected errors.

## Learning Notes

1. **Docker Networking:** Gaining insights into Docker networking and service isolation.
2. **Dockerfile Optimization:** Fine-tuning Dockerfiles for efficient container builds.
3. **Collaborative Development:** Enhancing collaboration through Docker Compose.

# Conclusion

The successful completion of both tasks demonstrates the team's proficiency in implementing Infrastructure as Code using Vagrant, Ansible, Dockerfile, and Docker Compose. Challenges were overcome through collaborative problem-solving, and key learnings have been documented for future reference. The project lays a foundation for scalable and maintainable deployments in the future.