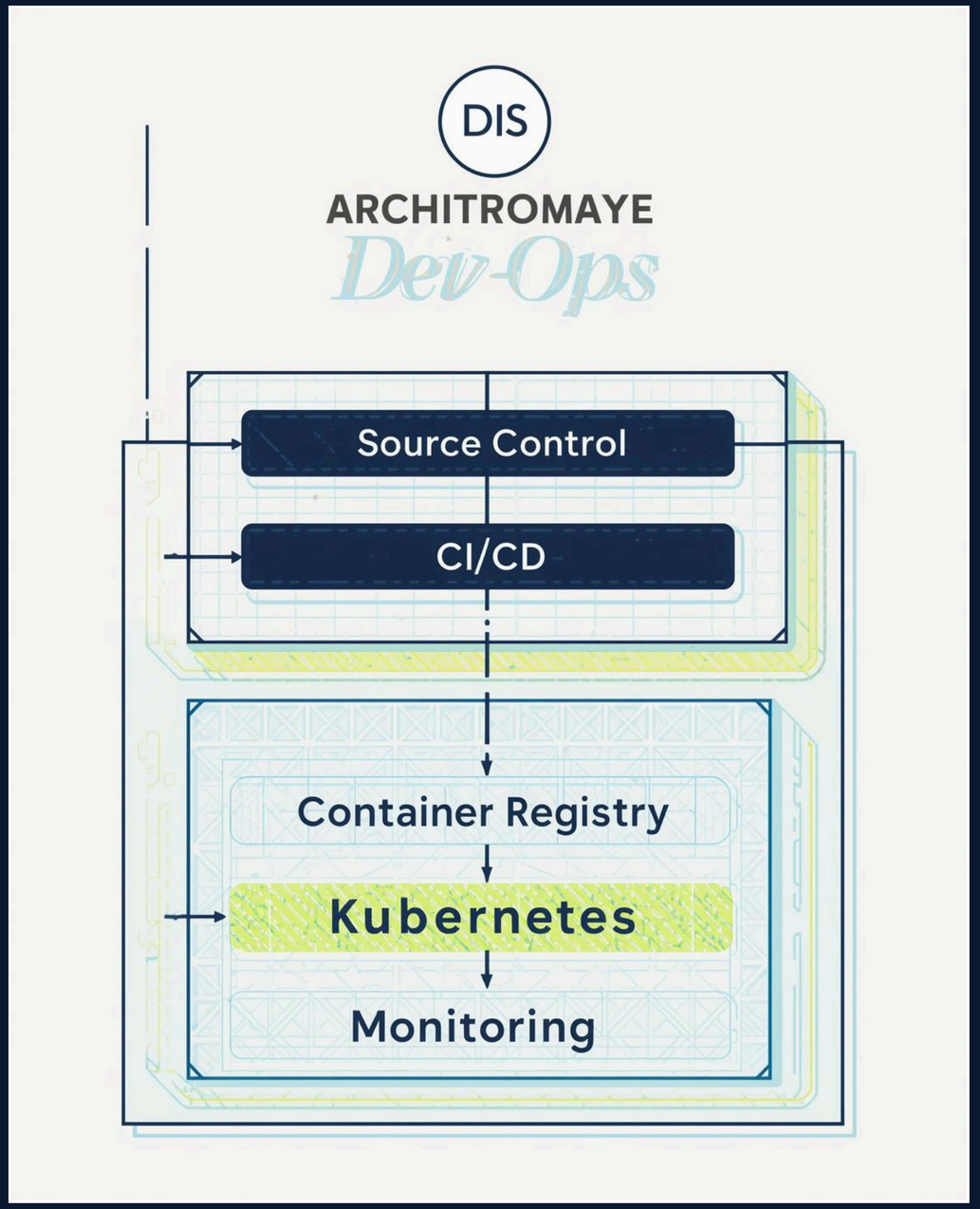
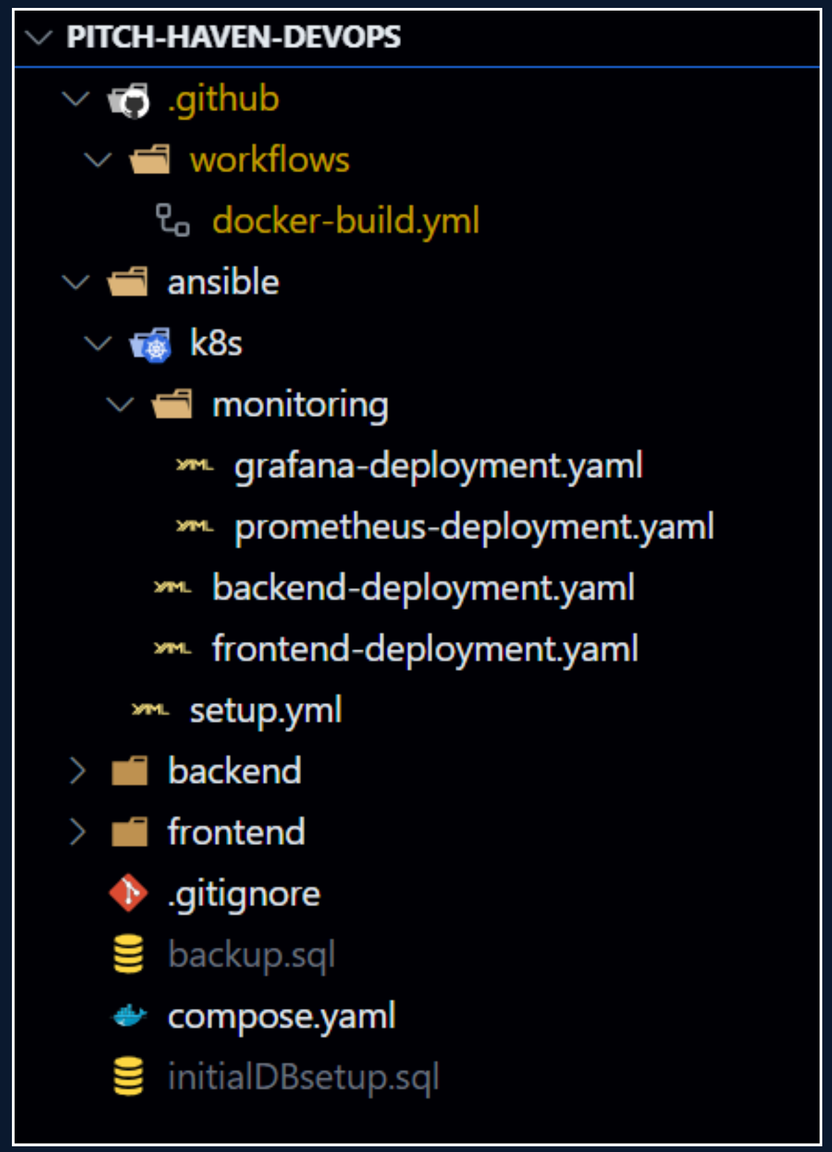


PitchHaven DevOps Solutions

Architecture Overview

Presented by Nikunj Maheshwari

This presentation outlines the architecture for a beginner-friendly DevOps pipeline.



Project Overview

Introducing the PitchHaven DevOps Setup Architecture for Beginner-Friendly Pipelines

A comprehensive guide to understanding the architectural scope of PitchHaven's DevOps setup.

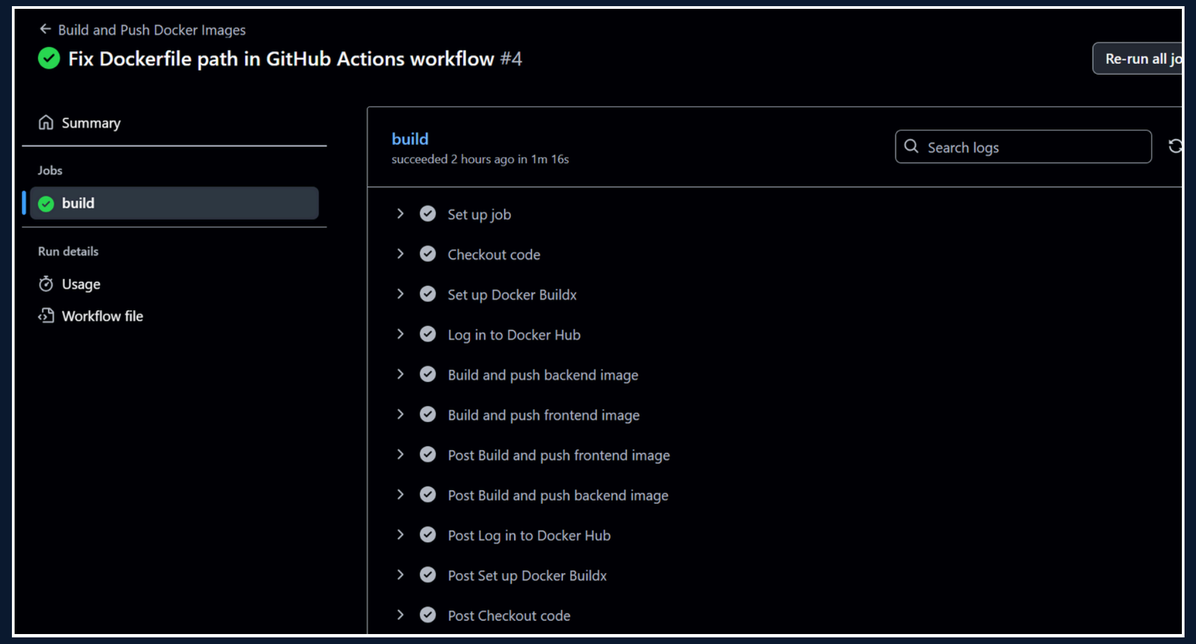
This page outlines the fundamental goal of the PitchHaven DevOps pipeline, which focuses on supporting continuous integration, containerization, and orchestration. The architecture diagram visually represents the end-to-end components and workflow.

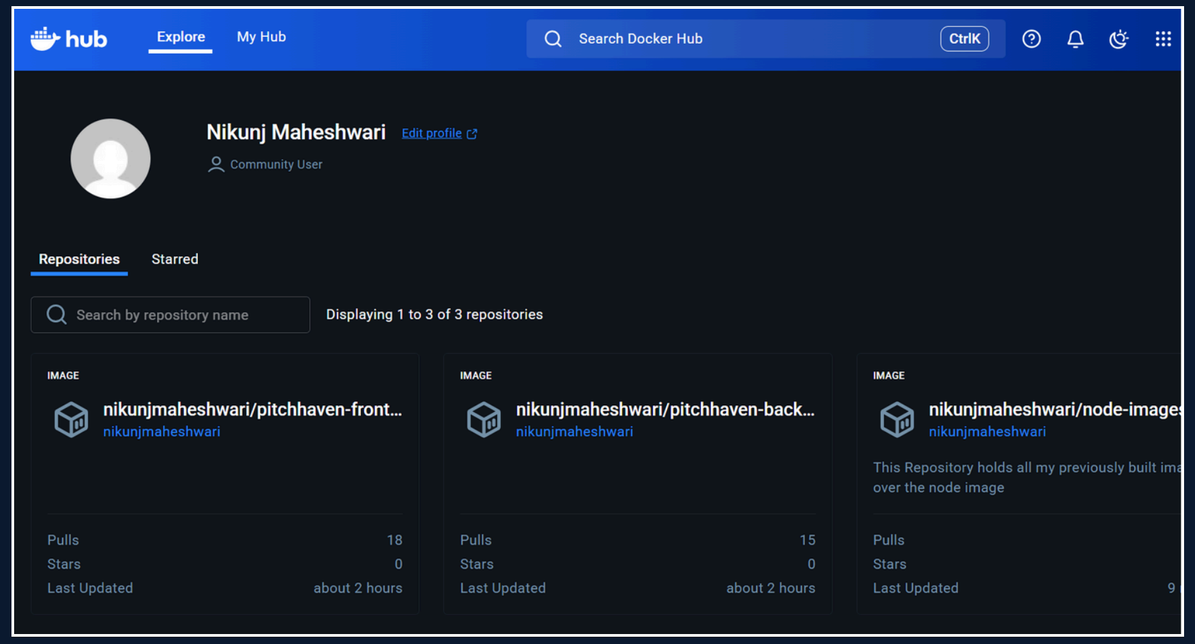
Continuous Integration & Deployment

Streamlining the DevOps Pipeline with Automation and Efficiency

This section details the **automated workflow** of continuous integration and deployment in the PitchHaven project. Utilizing GitHub Actions, the process builds Docker images for both frontend and backend applications, pushes them to Docker Hub, and deploys seamlessly to Kubernetes clusters.

Deployment Pipeline Details: GitHub Actions Workflow and Docker Hub Repository





GitHub Actions

Workflow automates build and deployment processes for seamless delivery.

Docker Hub

Centralized repository for storing and managing Docker images effectively.

```yaml
- name: Minimal DevOps Setup for PitchHaven
  hosts: localhost
  become: yes
  tasks:
    - name: Ensure 'htop' is installed
      apt:
        name: htop
        state: present
        update_cache: yes

    - name: Create a new user 'pitchdev'
      user:
        name: pitchdev
        state: present
        shell: /bin/bash

    - name: Create a test directory
      file:
        path: /home/nimahe/pitchhaven_test
        state: directory
        mode: '0755'

    - name: Create a test file
      copy:
```

Configuration Management with Ansible

Streamlining Infrastructure as Code for Efficient Deployment and Management

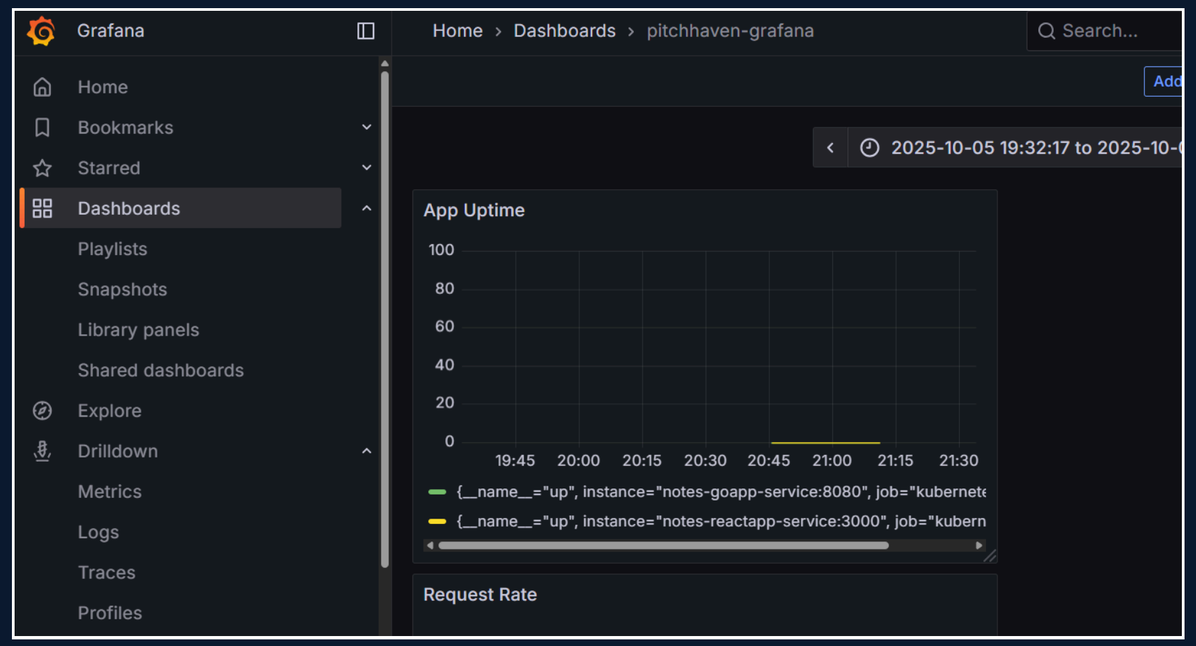
Ansible playbooks automate package installations, user creation, and directory setups, ensuring **repeatable environments**. This approach simplifies configurations, making deployments faster and reducing human error, which is essential for maintaining consistency in modern DevOps practices.

Monitoring with Prometheus & Grafana

Visualizing Service Metrics for Enhanced Insights and Performance

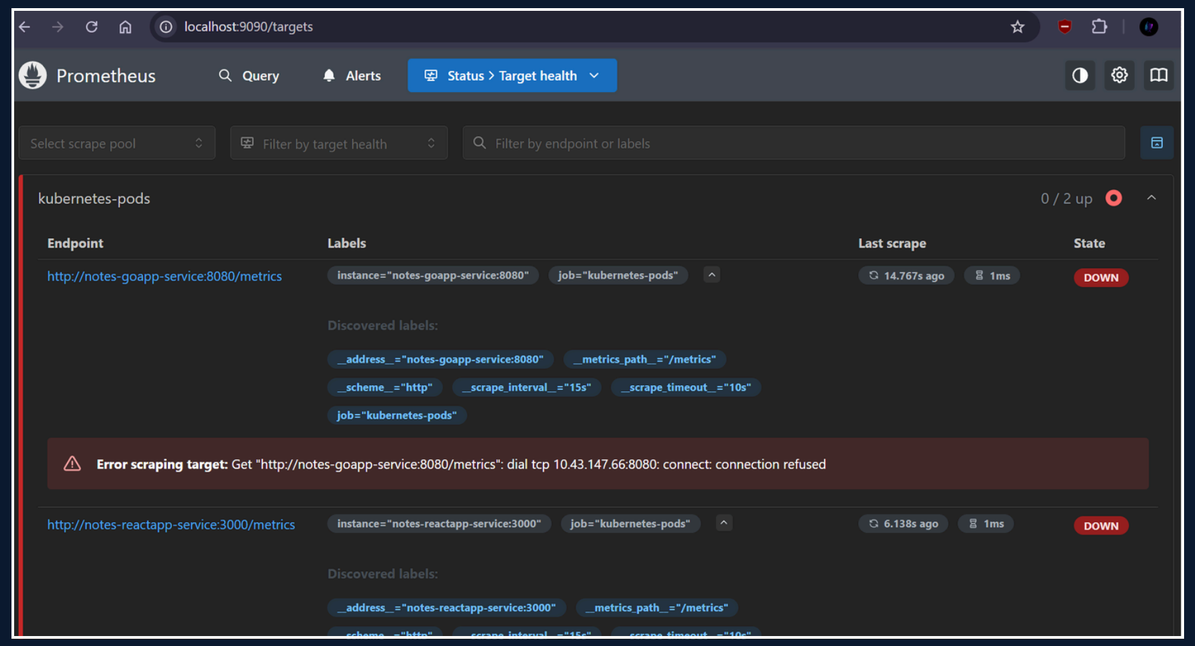
This section explores the integration of **Prometheus** for metric collection and **Grafana** for data visualization within the PitchHaven DevOps setup. By capturing key performance indicators such as uptime, request rate, and error rates, we can gain actionable insights into service health and performance dynamics.

Visualizing Metrics with Grafana and Prometheus



Grafana Dashboard

An overview of live metrics displaying service health and performance.



Prometheus Targets

Active targets monitored for effective performance metrics gathering.

Challenges & Lessons Learned

Overcoming Obstacles in the DevOps Process

Key challenges encountered

Throughout the project, we faced **Docker pull errors** that impacted our ability to retrieve images promptly. Additionally, Kubernetes rollout issues arose, causing delays in deployment. These challenges highlighted the importance of robust error handling and rapid troubleshooting in a DevOps environment.

Valuable lessons gained

We discovered that implementing **CI/CD** significantly simplifies deployment processes, reducing manual intervention. Furthermore, using Ansible allowed us to ensure a **repeatable environment setup**, facilitating consistency across deployments and enhancing overall efficiency in our workflow.