

Capstone Project Devops

GITHUB Repo: https://github.com/xrahulcrx/task_manager_capstone

Task Manager DevOps Pipeline (Jenkins + SonarQube + DockerHub + k3d)

Description

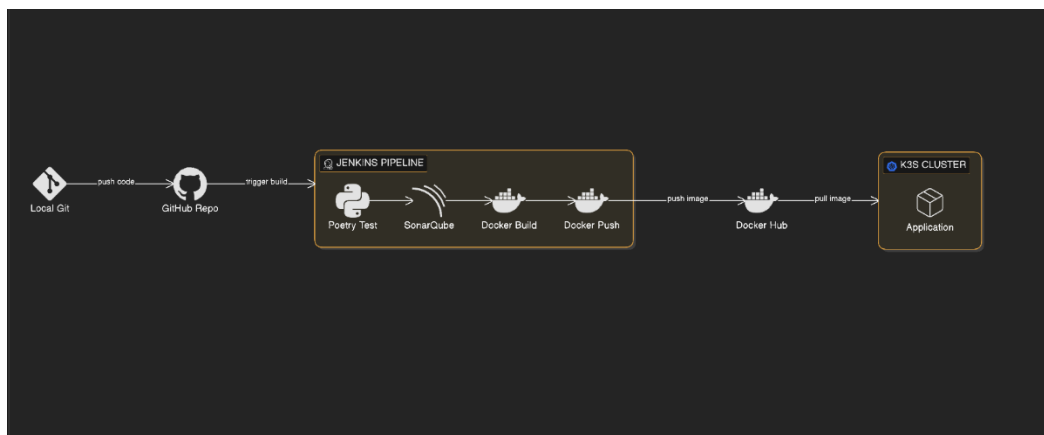
This project contains a full CI/CD pipeline using:

- GitHub
- Jenkins (Pipeline)
- SonarQube (Code Quality + Quality Gate)
- DockerHub (Build + Push Image)
- k3d (K3s Kubernetes cluster running in Docker)
- Kubernetes Deployment + Service from k8s/ folder

Architecture Overview

CI Flow

1. Checkout code from GitHub
2. Install dependencies using Poetry
3. Run unit tests using Pytest
4. Run SonarQube scan + Quality Gate check
5. Build Docker image
6. Push Docker image to DockerHub
7. Create/Reuse k3d cluster
8. Deploy app to k3d using Kubernetes manifests



Prerequisites

System Requirements

1. Docker Desktop (Windows)
2. WSL Ubuntu (recommended)
3. Git installed
4. Jenkins + SonarQube running using Docker Compose

Installation

Project Repo

Clone the repository:

```
git clone https://github.com/xrahulcrx/task_manager_capstone  
cd task_manager_capstone/infra
```

Setup Infrastructure (Jenkins + SonarQube)

```
chmod +x start.sh  
./start.sh
```

Run Manually to compose the docker

```
docker compose up -d --build
```

Checks

```
docker ps  
systemctl status docker
```

Jenkins Initial Setup

Get Jenkins Admin Password

```
docker exec -it jenkins cat /var/jenkins_home/secrets/initialAdminPassword
```

Open Jenkins URL:

```
http://localhost:8080/
```

SonarQube Setup

Open SonarQube

```
http://localhost:9000/
```

Generate Sonar Token

In SonarQube:

1. Go to My Account → Security
2. Generate a token
3. Copy it (needed in Jenkins)

Configure SonarQube Webhook

Go to Administration → Configuration → Webhooks

Create a webhook with:

Webhook URL:

`http://jenkins:8080/sonarqube-webhook/`

This is required for Jenkins Quality Gate stage.

Jenkins Plugins Required

1. GitHub Plugin - Integrates GitHub with Jenkins
2. SonarQube Scanner Plugin
3. Quality Gates Plugin (Sonar Quality Gates)

Jenkins Credentials Setup

Add Sonar Token in Jenkins

Go to:

Manage Jenkins → Credentials → (Global)

Add:

Kind: Secret Text

ID: sonar-token

Secret: (paste SonarQube token)

Configure SonarQube in Jenkins

Go to:

Manage Jenkins → System → SonarQube installations

Add SonarQube Installation:

Name: sonarqube

Server URL: http://sonarqube:9000

Authentication Token: select sonar-token

Add DockerHub Credentials in Jenkins

Generate DockerHub token:

DockerHub → Account Settings → Security → New Access Token

Then in Jenkins credentials add:

Kind: Username with Password

ID: dockerhub-creds

Username: your DockerHub username

Password: DockerHub token

Create Jenkins Pipeline Job

Create a Pipeline Job

In Jenkins:

New Item → Pipeline

Under Pipeline → Definition select:

Pipeline script from SCM

SCM: Git

Repo URL: https://github.com/xrahulcrx/task_manager_capstone

Branch: main

Script Path: Jenkinsfile

Save → Build Now

Enable Poll SCM (Auto Build on Git Changes)

Go to your Jenkins Job

Task-Manager-Pipeline (your pipeline job)

Click Configure - Under Build Triggers

Tick Poll SCM

Add schedule (cron)

Example: poll every 2 minutes

H/2 * * * *

Save

Application URLs

After successful deployment:

SonarQube Dashboard:

<http://localhost:9000/dashboard?id=task-manager-fastapi>

Task Manager Application:

<http://localhost:30080/docs>

Tasks Application list view:

<http://localhost:30080/tasks>

Check published Repo

<https://hub.docker.com/repository/docker/xrahulcrx/task-manager-fastapi/general>