Project Documentation: CI/CD Pipeline for index\_html App Deployment

# Overview

This project demonstrates the setup and deployment of a simple HTML-based application through a fully automated CI/CD pipeline using modern DevOps tools:

- GitHub – Source code versioning

- Jenkins – Continuous Integration and Continuous Deployment

- Docker – Application containerization

- Helm – Kubernetes deployment packaging

- ArgoCD – GitOps-based Continuous Delivery

# Step-by-Step Process

## Part A: Project Initialization

1. Created a project directory named index\_html and navigated into it.  
2. Added a simple HTML application file: index.html.  
3. Created a Dockerfile to containerize the application.  
4. Created a Jenkinsfile to define the build pipeline logic.  
5. Created a Helm chart named mycharts to manage Kubernetes deployment.  
6. Added a secondary Jenkinsfile inside the mycharts directory for Jenkins task 2.

## Part B: Version Control with GitHub

Pushed the index\_html project directory to GitHub. Used a secondary branch named 'user' to perform development, testing, and integration.

## Part C: Jenkins CI – Task 1 (index\_html\_1)

Created a Jenkins item named index\_html\_1. Configured it with Pipeline: SCM and selected the user branch. Used a Jenkinsfile from the root of the project to:  
- Clone the repository  
- Build and tag the Docker image  
- Push the image to Docker Hub  
- (Optional) Remove any pre-existing Docker containers

## Part D: Jenkins CD – Task 2 (index\_html\_2)

Created a second Jenkins item named index\_html\_2, configured to trigger after index\_html\_1 completes. Configured this as a pipeline job with Pipeline: SCM and selected the user branch. Located the Jenkinsfile for this task inside the mycharts directory. The job’s purpose:  
- Update the image tag in the Helm chart values.yaml to v1  
- Commit and push the update back to GitHub

## Part E: ArgoCD Deployment Setup

Launched an application in ArgoCD using:  
- Git repo: https://github.com/AbhishekRangra/index\_html.git  
- Branch: main  
- Path: mycharts  
Initially, the values.yaml under mycharts had no real image (tag: "latest"), so the deployment had no actual running container image.

NOTE: in another scenario, Jenkins Task 1 standalone could have created an image with tag: ”Latest” and in that case end result would have been Argocd pulling same image but with an updated tag: “v1” instead of running application with a blank image.

## Part F: Full CI/CD Pipeline Execution

1. Ran index\_html\_1 in Jenkins:  
 - Successfully built and pushed the Docker image abhishekrangra/index\_html:v1.  
2. This automatically triggered index\_html\_2:  
 - Updated the image tag in mycharts/values.yaml to "v1".  
 - Committed and pushed the update to the user branch.  
3. Created and merged a pull request from user into main.  
4. Now, main reflects the updated image tag (v1).  
5. Synced the ArgoCD application:  
 - ArgoCD pulled the updated Helm chart from main.  
 - Deployed the app using the real Docker image.  
6. ✅ Application successfully deployed via GitOps and CI/CD integration.

# Conclusion

This project exemplifies a production-grade CI/CD pipeline setup with GitOps-driven deployment using ArgoCD. It also demonstrates a clean separation of responsibilities across pipeline tasks, image management, and Helm-driven Kubernetes deployments.

A screenshot of a computer

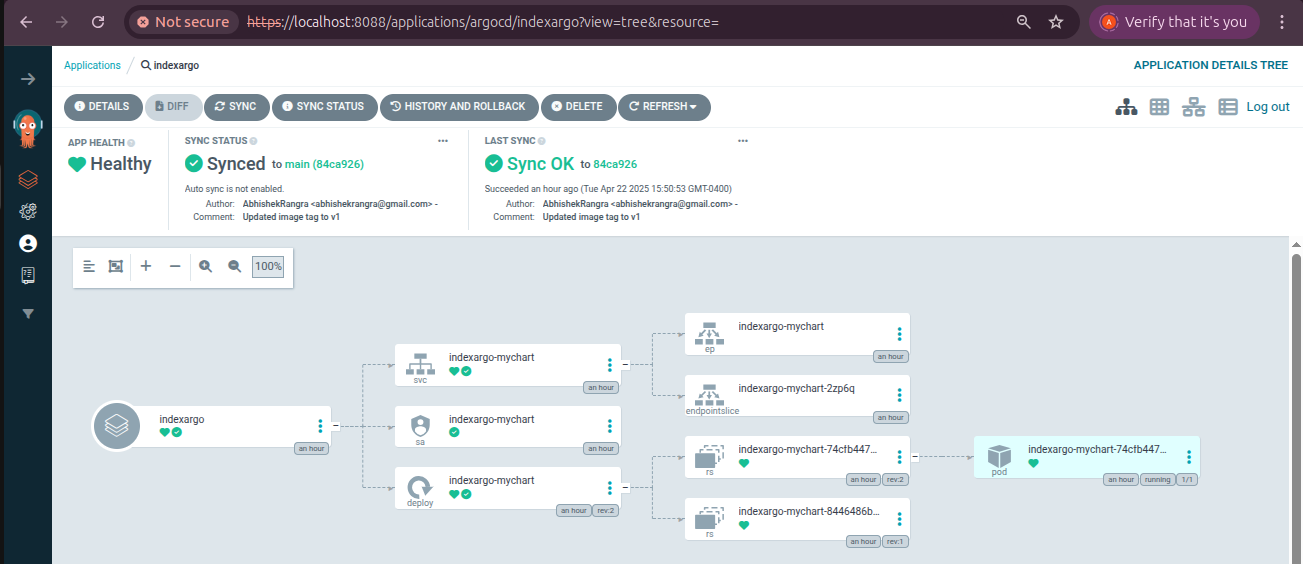
AI-generated content may be incorrect.

Jenkins Task 1: index\_html\_1

A screenshot of a computer

AI-generated content may be incorrect.

Jenkins Task 2: index\_html\_2



Final result in Argocd