Weather App

This repository contains the necessary files and configurations for deploying a weather application using Docker, Jenkins CI/CD pipeline, and Kubernetes.

Names: Salma, Farah ,Tuqa ,Menna ,Hanna

Group number: 3

Group Code: GIZ1_SWD1_S3d_DEPI

Features & Requirements

WeatherApp

A weather application built with HTML, CSS, and JavaScript to view weather information for any country.

Dockerfile

Builds the application's Docker image.

Jenkinsfile

Automates the build, test, and deployment process using Jenkins.

Kubernetes Configurations

YAML files for deployment, service, and namespace setup in a Kubernetes cluster.

Requirements

- Docker
- Jenkins
- Kubernetes cluster
- Docker registry

Project Structure

Project Structure

The Weather App is a simple application built with HTML, CSS, and JavaScript.

- Dockerfile
- Jenkinsfile
- k8s/
- index.html
- styles.css
- app.js

Building the Docker Image

1

Build the Docker Image

Run the Dockerfile to build the image.

2

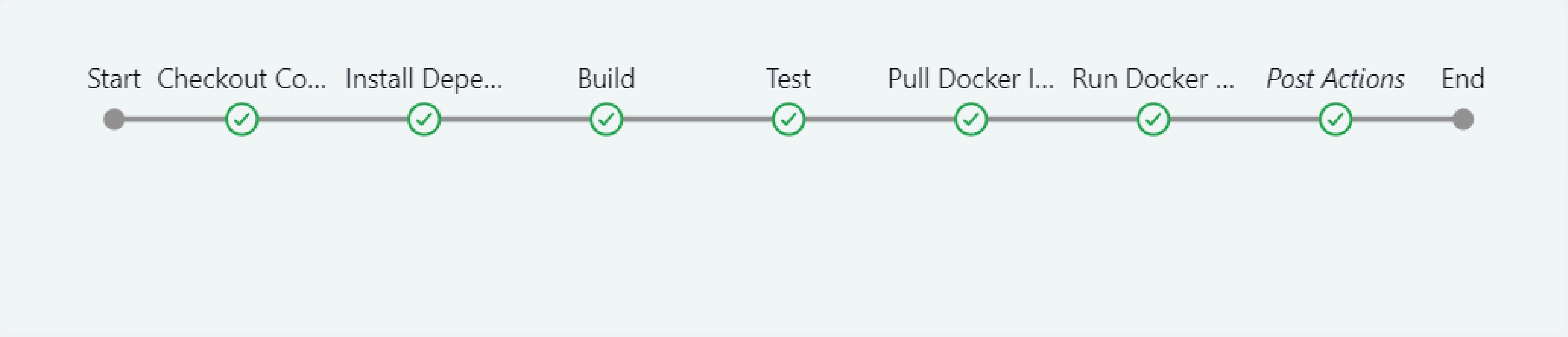
Tag the Image

Tag the image with a descriptive name and version.

3

Push to Docker Registry

Push the tagged image to your Docker registry.



Jenkins CI/CD Pipeline

1

Clone Repository

Jenkins pulls the latest code from GitHub.

2

Build, Test, Deploy

Jenkins builds, tests, and deploys the project.

3

Deploy Docker Container

Jenkins pulls the image from the Docker registry and creates a container.

Jenkins Setup Instructions

Plugin Installation

Ensure your Jenkins server has the following plugins installed: Git Plugin, Docker Pipeline Plugin.

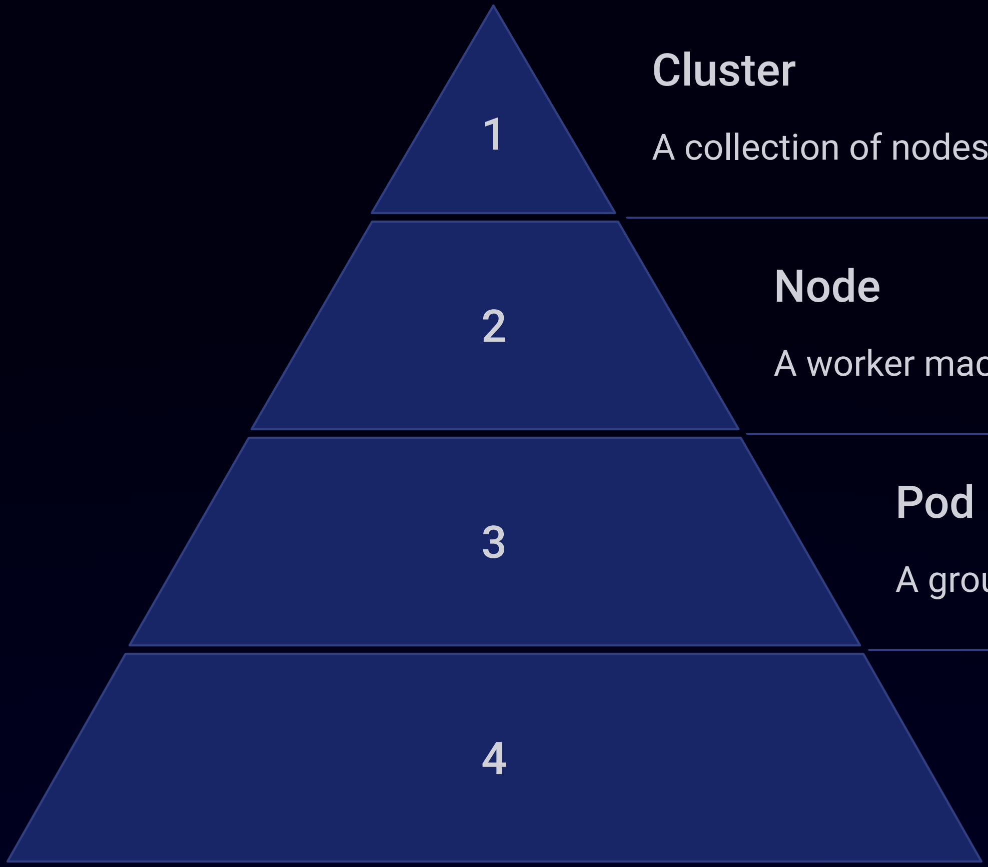
Pipeline Configuration

Create a new Jenkins pipeline job and use the Jenkinsfile from the repository for the pipeline configuration.

Start the Process

Build the job to start the CI/CD process.

Importance of Kubernetes and Orchestration



Cluster

A collection of nodes that work together

Node

A worker machine in the Kubernetes cluster

Pod

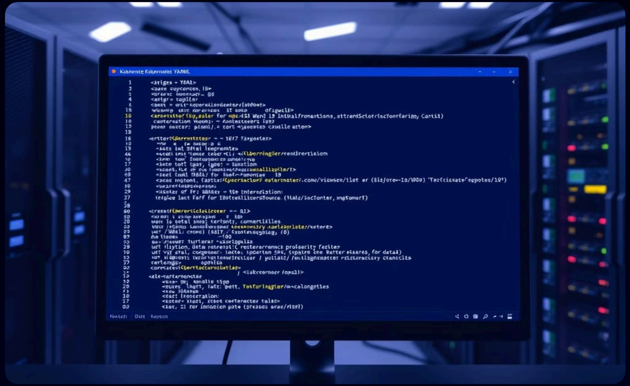
A group of one or more containers that share resources

Service

Provides access to applications running in pods

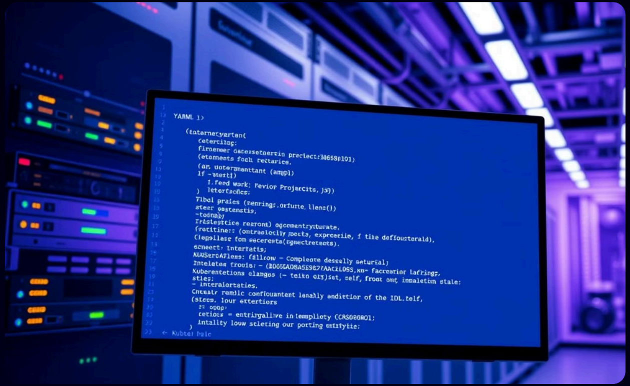
- Simplifies deployment and management of containerized applications
- Provides high availability and scalability
- Enables automated rollouts, rollbacks, and self-healing
- Offers resource optimization and efficient utilization

Kubernetes Deployment Configuration



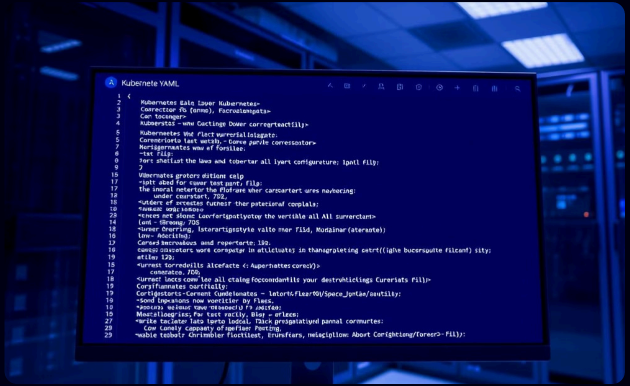
deployment.yaml

Defines the pod specification and the desired number of replicas for the application.



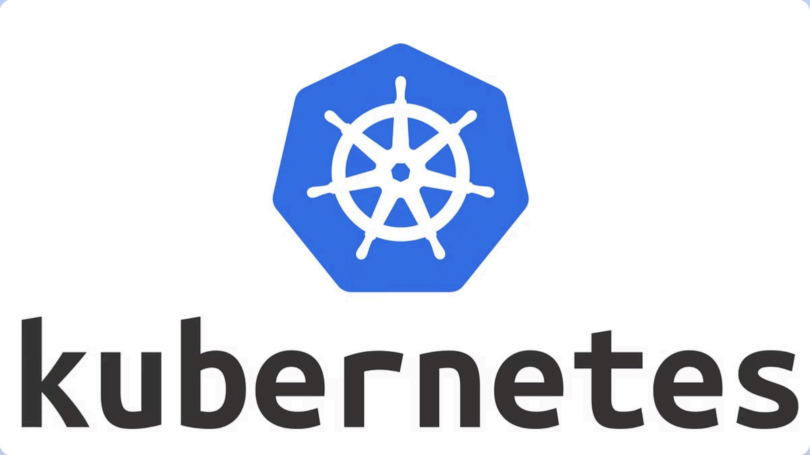
service.yaml

Exposes the application as a service, allowing external access via a LoadBalancer or NodePort.



namespace.yaml

Defines the namespace where the application will run in the Kubernetes cluster.



Kubernetes Deployment Commands

1

Create Namespace

Create the namespace for the application.

2

Deploy Application

Deploy the Weather App to the Kubernetes cluster.

3

Expose Service

Expose the application as a service.

4

Verify Deployment

Verify the pods and services are running.

Conclusion & Next Steps

1

Deployment Complete

The Weather App is now deployed to a Kubernetes cluster.

2

Future Improvements

Future improvements can include implementing unit tests, adding robust error handling, and integrating with other services.