Tenderd - DevOps Assignment

Docmentation

Overview

This services is a microservices-based application designed to manage and streamline various services through a unified platform. The primary objective of this project is to implement a robust CI/CD pipeline, deploy the microservices to a Kubernetes cluster, and integrate monitoring and logging solutions to ensure the application's performance and reliability.

This documentation outlines the project structure, CI/CD pipeline, deployment process, and the integration of GCP services for additional functionality.

Project Repositories

The project is divided into two repositories:

Microservices Repository:

Contains the source code for the microservices. * Repository: Tenderd-DevOps-Assignement-3 * Structure: `Tenderd-DevOps-Assignement-3 â”œâ”€â”€ frontend-svc â”, â”œâ”€â”€ Dockerfile â”, â”œâ”€â”€ node_modules â”, â”œâ”€â”€ package.json â”, â”œâ”€â”€ package-lock.json â”, â”œâ”€â”€ src â”, â”œâ”€â”€ tests â”œâ”€â”€ order-svc â”, â”œâ”€â”€ Dockerfile â”, â”œâ”€â”€ package.json â”, â”œâ”€â”€ package-lock.json â”, â”œâ”€â”€ src â”œâ”€â”€ user-svc â”, â”œâ”€â”€ Dockerfile â”, â”œâ”€â”€ package.json â”, â”œâ”€â”€ package-lock.json â”, â”œâ”€â”€ src â”œâ”€â”€ docker-compose.yaml â”œâ”€â”€ README.md`

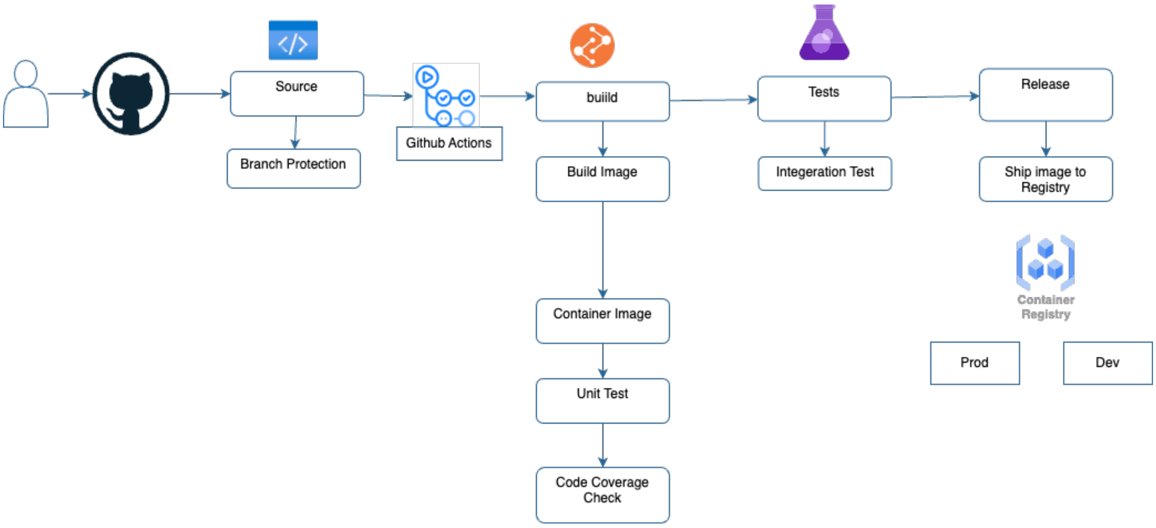
Infrastructure Repository:

Contains the infrastructure as code (IaC) for deploying the microservices to Kubernetes. * Repository: Tenderd-DevOps-Assignement-Infra-3 * Structure: ``` Tenderd-DevOps-Assignement-Infra-3 â”œâ”€â”€ helm-charts â”, â”œâ”€â”€ frontend-svc-chart â”, â”œâ”€â”€ templates â”, â”œâ”€â”€ values.yaml â”œâ”€â”€ terraform â”, â”œâ”€â”€ dev â”, â”, â”œâ”€â”€ main.tf â”, â”, â”œâ”€â”€ outputs.tf â”, â”, â”œâ”€â”€ variables.tf â”œâ”€â”€ k8-status.sh â”œâ”€â”€ README.md

CI/CD Pipeline

The CI/CD pipeline is implemented using GitHub Actions, GCP, Terraform and Docker. The pipeline performs the following tasks:

Tenderd-DevOps-Assignement-3:



Deployment Process:

- Code Checkout: Retrieves the latest code from the repository.
- Docker Buildx Setup: Configures Docker Buildx for multi-platform builds.
- Docker Hub Login: Authenticates with Docker Hub using credentials stored in GitHub Secrets.
- Dependency Installation and Testing: Installs dependencies and runs tests for each microservice.
- Docker Image Build and Push: Builds Docker images for each microservice and pushes them to Docker Hub.

    Github Actions (3):

1- tenderd-build-push-images-hub

``` name: CI

on: push: branches: - dev-1 pull_request: branches: - main

jobs: btenderd-build-push-images-hub: runs-on: ubuntu-latest

```
steps:
  - name: Checkout code
    uses: actions/checkout@v3

  - reset of step ...
```

```

tenderd-close-stale-issues

``` name: tenderd-close-stale-issues

on: schedule: - cron: '30 1 * * *'

jobs: tenderd-close-stale-issues: runs-on: ubuntu-latest steps: - uses: actions/stale@v9 with: stale-issue-message: 'This issue is stale because it has been open 30 days with no activity. Remove stale label or comment or this will be closed in 5 days.' days-before-stale: 30 days-before-close: 5 ```

###### tenderd-apply-branch-protection

``` name: tenderd-apply-branch-protection

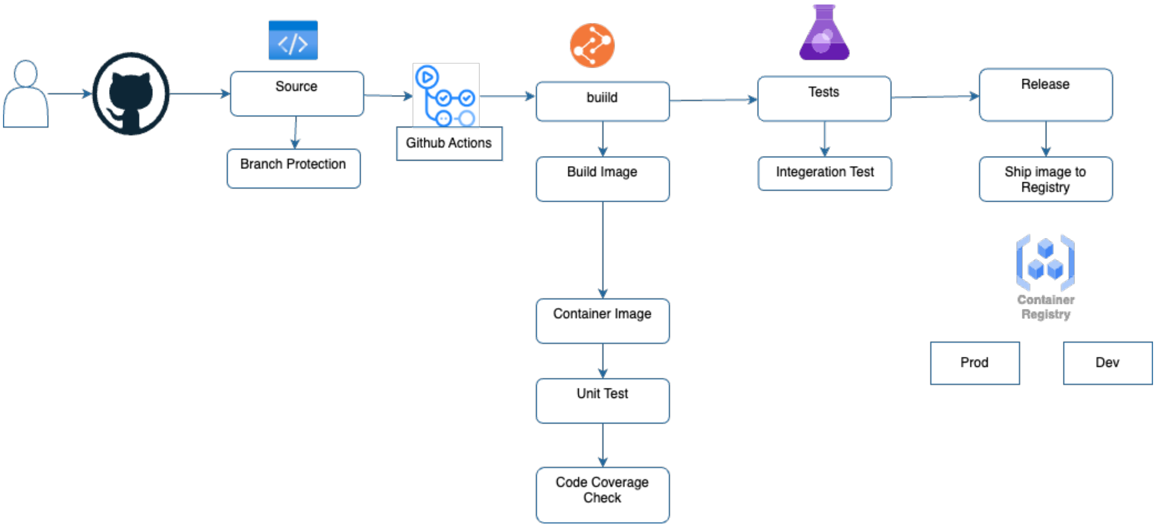
on: push: branches: - main

jobs: tenderd-apply-branch-protection: runs-on: ubuntu-latest

```
steps:
- name: Apply branch protection rules
  run: |
    echo "Applying relaxed branch protection rules to ${{ github.repository }}"

    curl ...
  env:
    MY_GITHUB_PAT: ${{ secrets.TENDERD_TOKEN }}
```

Tenderd-DevOps-Assignement-Infra-1:



Deployment Process:

- The deployment process uses Helm charts and Terraform to manage Kubernetes resources and GCP infrastructure.
- Terraform: Manages GCP resources, such as creating a GKE cluster and provisioning necessary infrastructure.
- Helm Charts: Define Kubernetes resources for each microservice. ``` Frontend-svc-chart example:

Chart.yaml

apiVersion: v2 name: frontend-svc description: A Helm chart for Kubernetes version: 0.1.0 appVersion: 1.16.0

terraform statment to deploy resrouce example: provider "google" { project = var.project region = var.region }

resource "google*container*cluster" "primary" { name = "primary-cluster" location = var.region

node*config { machine*type = "e2-medium" } } ```

Github Actions (5):

deploy-upgrade-helm-charts-dev

``` name: tenderd-deploy-dev run-name: ${{ github.actor }} has triggered the pipeline for to build tendered GKE kubernetes cluster and deploy/upgrade helm charts services.

on: push: branches: - 'dev'

defaults: run: shell: bash working-directory: ./terraform/dev-gke

permissions: contents: read id-token: write

jobs: tenderd-deploy-upgrade-helm-charts-dev: runs-on: ubuntu-latest needs: deploy-gke-k8-cluster-dev steps: - name: Checkout uses: actions/checkout@v2

```
  - reset of step ...
```

```

tenderd-prevision-gke-cluster-dev

``` tenderd-prevision-gke-cluster-dev: runs-on: ubuntu-latest permissions: id-token: write contents: read
steps: - name: Checkout uses: actions/checkout@v2 - name: Configure GCP credentials id: auth uses: google-github-actions/auth@v2 with: workload*identity*provider: ${{ secrets.WORKLOAD*IDENTITY*PROVIDER }} create*credentials*file: true service*account: ${{ secrets.SA }} token*format: "access*token" access*token_lifetime: "120s"

```
  - reset of step ...
```

```

##### tenderd-close-stale-issues-prs

``` name: 'Close stale issues and PRs'

on: schedule: - cron: '30 1 * * *'

jobs: tenderd-close-stale-issues-prs: runs-on: ubuntu-latest steps: - uses: actions/stale@v9 with: stale-issue-message: 'This issue is stale because it has been open 30 days with no activity. Remove stale label or comment or this will be closed in 5 days.' days-before-stale: 30 days-before-close: 5 ```

##### tenderd-apply-branch-protection:

``` name: Apply Branch Rules

on: push: branches: - main

jobs: tenderd-apply-branch-protection:: runs-on: ubuntu-latest

~~~
steps:
- name: Apply branch protection rules
  run: |
    echo "Applying relaxed branch protection rules to ${{ github.repository }}"

    curl ...
      https://api.github.com/repos/${{ github.repository }}/branches/main/protection
  env:
    MY_GITHUB_PAT: ${{ secrets.TENDERD_TOKEN }}
```
~~~

destroy-dev

``` name: tenderd-destroy-dev run-name: ${{ github.actor }} has triggered the pipeline for Terraform

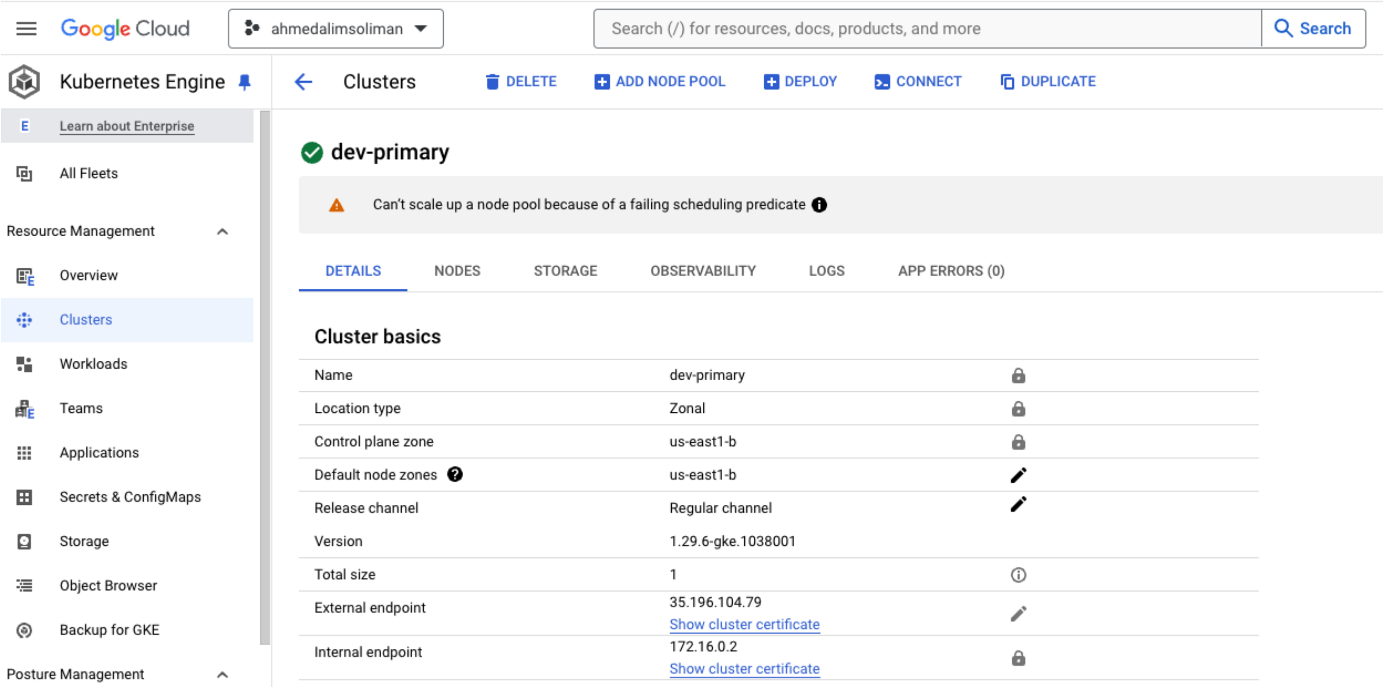
on: push: branches: - 'dev'

defaults: run: shell: bash working-directory: ./terraform/dev-gke permissions: contents: read

jobs: destroy-dev: runs-on: ubuntu-latest permissions: id-token: write contents: read steps: - name: "Checkout" uses: actions/checkout@v3 - reset of step ... ```

**Images**

- Created GKE by Github Actions + Terraform on Google Cloud



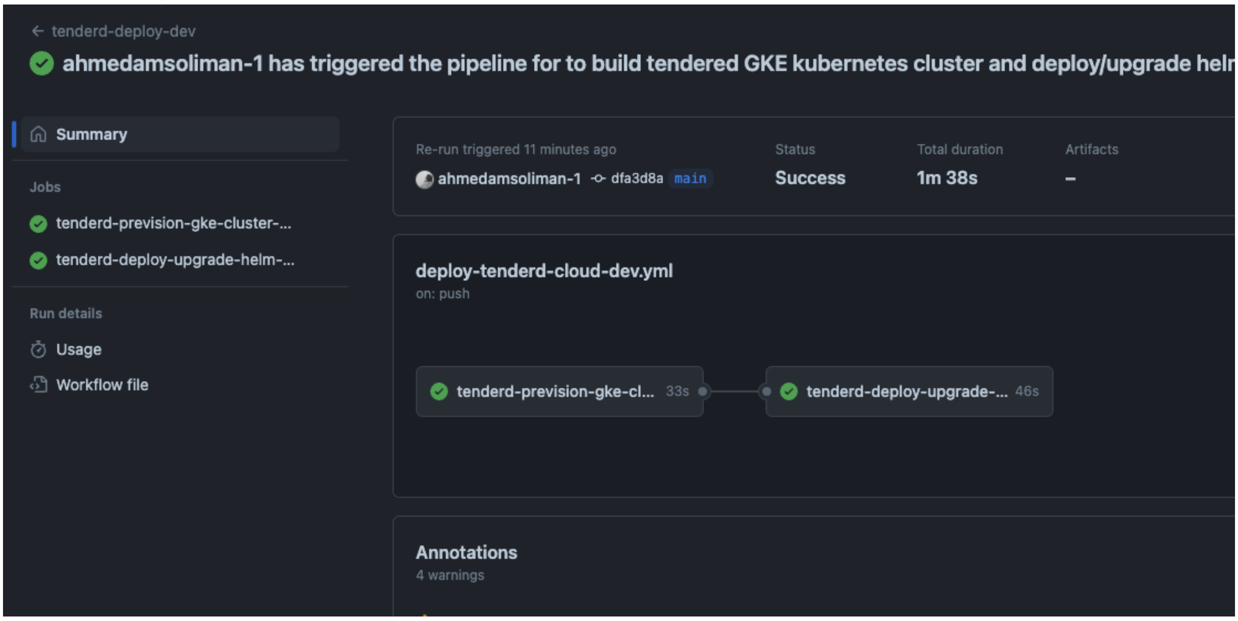
dev-primary

Can't scale up a node pool because of a failing scheduling predicate

Cluster basics

| | |
|---|---|
| Name | dev-primary |
| Location type | Zonal |
| Control plane zone | us-east1-b |
| Default node zones | us-east1-b |
| Release channel | Regular channel |
| Version | 1.29.6-gke.1038001 |
| Total size | 1 |
| External endpoint | 35.196.104.79<br>Show cluster certificate |
| Internal endpoint | 172.16.0.2<br>Show cluster certificate |

- Github Actions


tenderd-deploy-dev
Summary
Jobs
tenderd-prevision-gke-cluster-...
tenderd-deploy-upgrade-helm-...
Run details
Usage
Workflow file
tenderd-prevision-gke-cl... 33s
tenderd-deploy-upgrade-... 46s

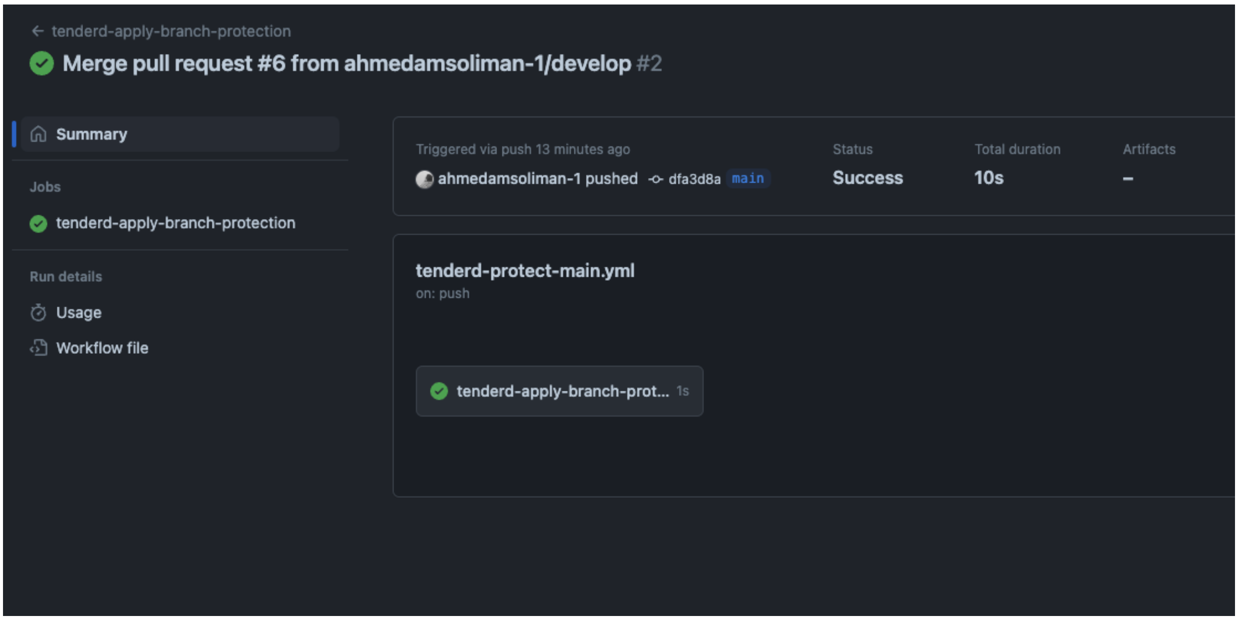
ahmedamsoliman-1 has triggered the pipeline for to build tendered GKE kubernetes cluster and deploy/upgrade hel

Re-run triggered 11 minutes ago
ahmedamsoliman-1 dfa3d8a main

| Status | Total duration | Artifacts |
|---|---|---|
| Success | 1m 38s | – |

deploy-tenderd-cloud-dev.yml
on: push

Annotations
4 warnings
```



← tenderd-apply-branch-protection

✓ Merge pull request #6 from ahmedamsoliman-1/develop #2

Summary

Jobs

✓ tenderd-apply-branch-protection

Run details

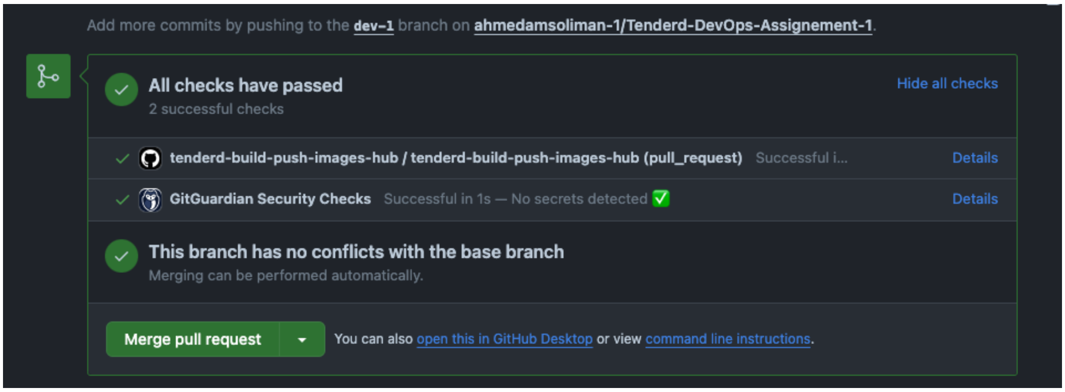
Usage

Workflow file

Triggered via push 13 minutes ago	Status	Total duration	Artifacts
ahmedamsoliman-1 pushed -o- dfa3d8a main	Success	10s	–

tenderd-protect-main.yml
on: push

✓ tenderd-apply-branch-prot... 1s



Add more commits by pushing to the **dev-1** branch on **ahmedamsoliman-1/Tenderd-DevOps-Assignement-1**.

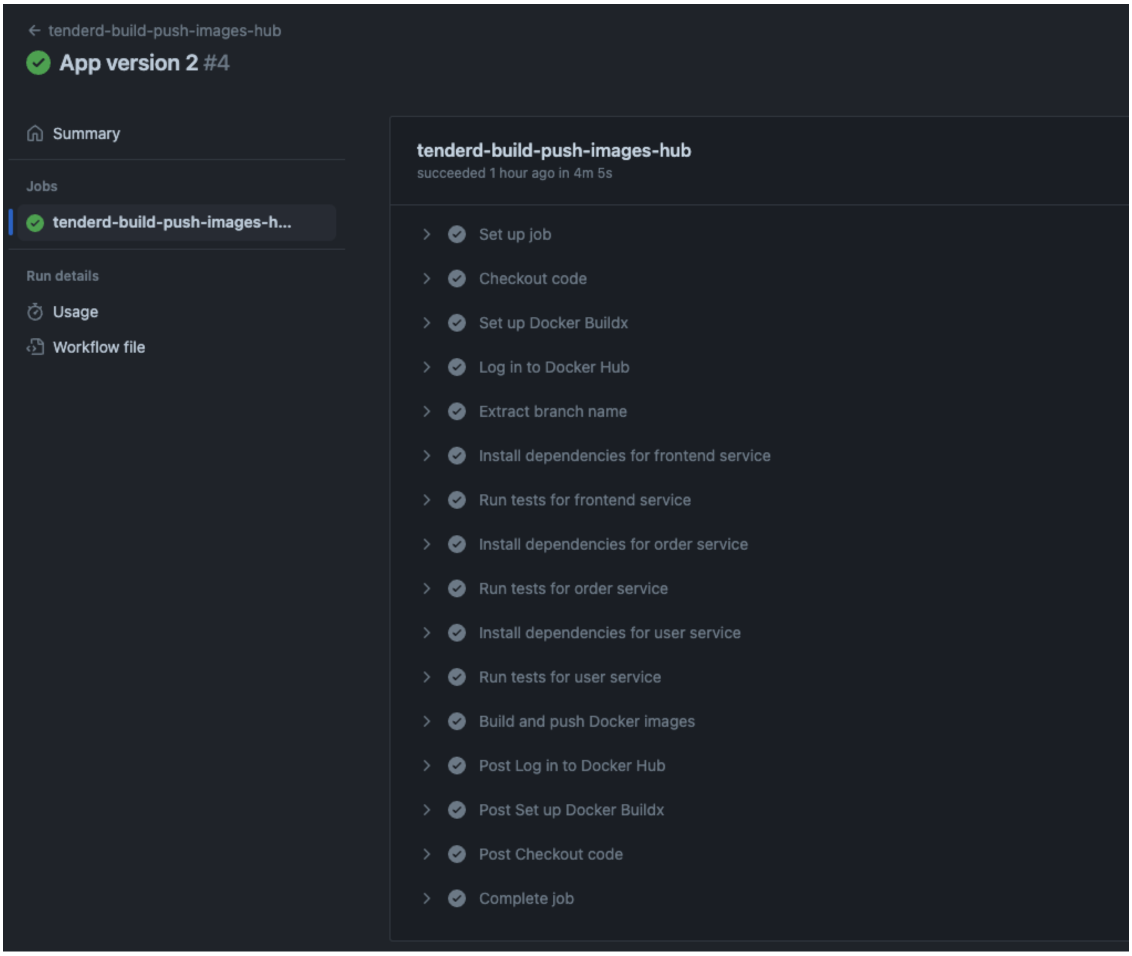
✓ **All checks have passed** Hide all checks
2 successful checks

✓ **tenderd-build-push-images-hub / tenderd-build-push-images-hub (pull_request)** Successful i... Details

✓ **GitGuardian Security Checks** Successful in 1s — No secrets detected ✅ Details

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request ▾ You can also open this in GitHub Desktop or view command line instructions.



← tenderd-build-push-images-hub

✓ App version 2 #4

Summary

Jobs

✓ tenderd-build-push-images-h...

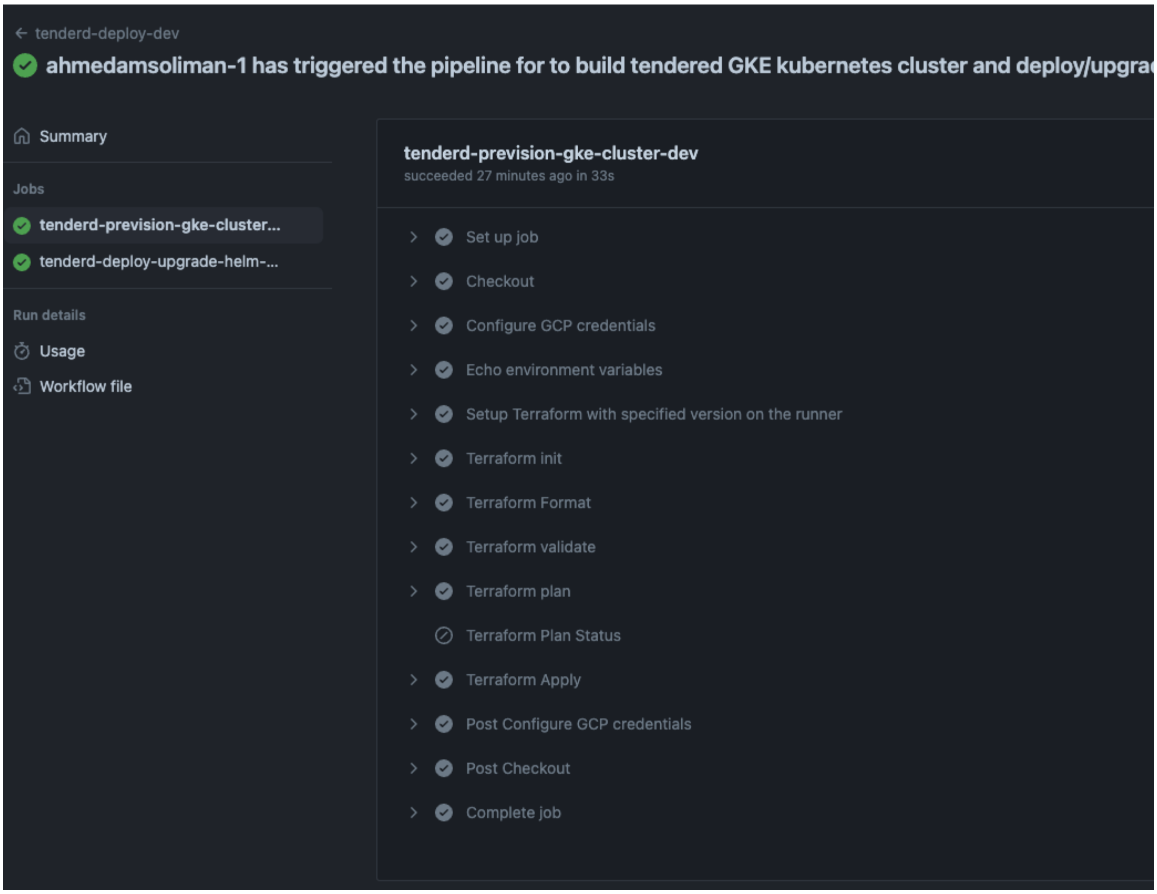
Run details

Usage

Workflow file

tenderd-build-push-images-hub
succeeded 1 hour ago in 4m 5s

- ✓ Set up job
- ✓ Checkout code
- ✓ Set up Docker Buildx
- ✓ Log in to Docker Hub
- ✓ Extract branch name
- ✓ Install dependencies for frontend service
- ✓ Run tests for frontend service
- ✓ Install dependencies for order service
- ✓ Run tests for order service
- ✓ Install dependencies for user service
- ✓ Run tests for user service
- ✓ Build and push Docker images
- ✓ Post Log in to Docker Hub
- ✓ Post Set up Docker Buildx
- ✓ Post Checkout code
- ✓ Complete job



← tenderd-deploy-dev

✓ ahmedamsoliman-1 has triggered the pipeline for to build tendered GKE kubernetes cluster and deploy/upgra

Summary

Jobs

✓ tenderd-prevision-gke-cluster...

✓ tenderd-deploy-upgrade-helm-...

Run details

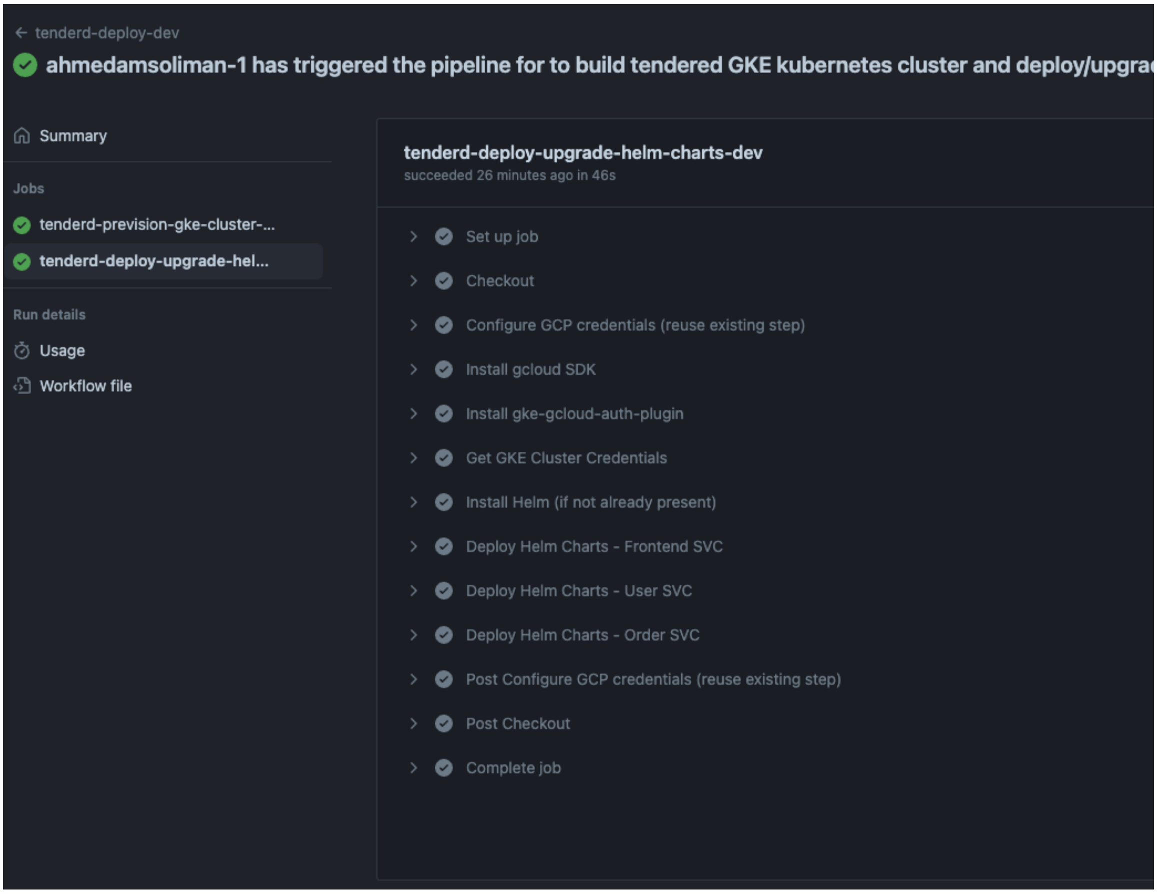
Usage

Workflow file

tenderd-prevision-gke-cluster-dev

succeeded 27 minutes ago in 33s

- ✓ Set up job
- ✓ Checkout
- ✓ Configure GCP credentials
- ✓ Echo environment variables
- ✓ Setup Terraform with specified version on the runner
- ✓ Terraform init
- ✓ Terraform Format
- ✓ Terraform validate
- ✓ Terraform plan
- ⊘ Terraform Plan Status
- ✓ Terraform Apply
- ✓ Post Configure GCP credentials
- ✓ Post Checkout
- ✓ Complete job



← tenderd-deploy-dev

✓ ahmedamsoliman-1 has triggered the pipeline for to build tendered GKE kubernetes cluster and deploy/upgra

Summary

Jobs

✓ tenderd-prevision-gke-cluster-...

✓ tenderd-deploy-upgrade-hel...

Run details

Usage

Workflow file

tenderd-deploy-upgrade-helm-charts-dev

succeeded 26 minutes ago in 46s

- ✓ Set up job
- ✓ Checkout
- ✓ Configure GCP credentials (reuse existing step)
- ✓ Install gcloud SDK
- ✓ Install gke-gcloud-auth-plugin
- ✓ Get GKE Cluster Credentials
- ✓ Install Helm (if not already present)
- ✓ Deploy Helm Charts - Frontend SVC
- ✓ Deploy Helm Charts - User SVC
- ✓ Deploy Helm Charts - Order SVC
- ✓ Post Configure GCP credentials (reuse existing step)
- ✓ Post Checkout
- ✓ Complete job

Monitoring

Monitoring and logging are integrated using Prometheus and Grafana for metrics, and ELK stack (Elasticsearch, Logstash, and Kibana) for logs.

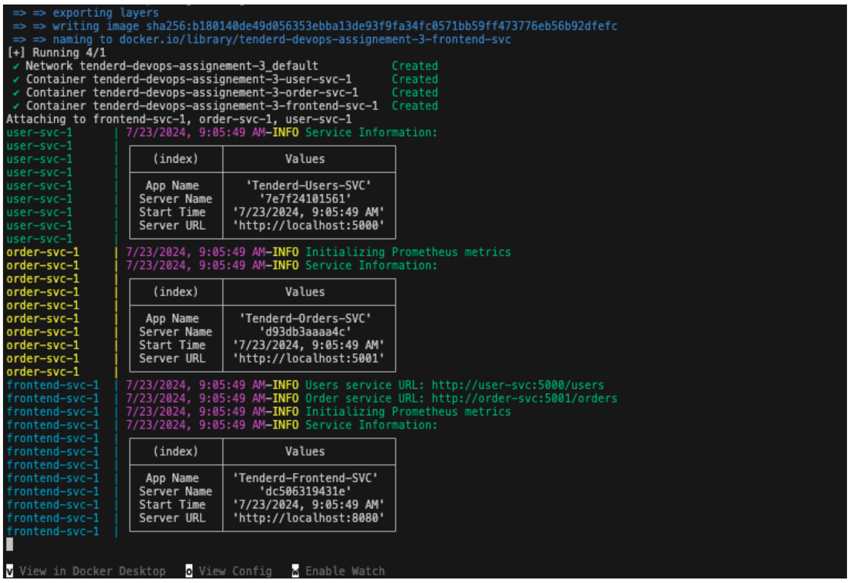
Logging

Logging is not supported yet (possibility to capture and visualize using ELK stack (Elasticsearch, Logstash, and Kibana))

Docker Compose for Development

The docker-compose.yaml file in micro services repository is used for local development and testing. It simplifies the process of running and managing the microservices locally by defining the necessary services, networks. - Configuration: ``` version: '3' services: frontend-svc: build: context: ./frontend-svc ports: - "3000:3000" depends_on: - order-svc - user-svc order-svc: build: context: ./order-svc ports: - "3001:3001"

user-svc: build: context: ./user-svc ports: - "3002:3002"



```
 => => exporting layers
 => => writing image sha256:b180140de49d056353ebba13de93f9fa34fc0571bb59ff473776eb56b92dfefc
 => => naming to docker.io/library/tenderd-devops-assignement-3-frontend-svc
[+] Running 4/1
 ✓ Network tenderd-devops-assignement-3_default           Created
 ✓ Container tenderd-devops-assignement-3-user-svc-1      Created
 ✓ Container tenderd-devops-assignement-3-order-svc-1     Created
 ✓ Container tenderd-devops-assignement-3-frontend-svc-1  Created
Attaching to frontend-svc-1, order-svc-1, user-svc-1
user-svc-1      | 7/23/2024, 9:05:49 AM-INFO Service Information:
user-svc-1      |
user-svc-1      | ┌─────────────┬──────────────────────────┐
user-svc-1      | │   (index)   │          Values          │
user-svc-1      | ├─────────────┼──────────────────────────┤
user-svc-1      | │  App Name   │   'Tenderd-Users-SVC'    │
user-svc-1      | │ Server Name │     '7e7f24101561'       │
user-svc-1      | │ Start Time  │ '7/23/2024, 9:05:49 AM'  │
user-svc-1      | │ Server URL  │ 'http://localhost:5000'  │
user-svc-1      | └─────────────┴──────────────────────────┘
user-svc-1      |
order-svc-1     | 7/23/2024, 9:05:49 AM-INFO Initializing Prometheus metrics
order-svc-1     | 7/23/2024, 9:05:49 AM-INFO Service Information:
order-svc-1     |
order-svc-1     | ┌─────────────┬──────────────────────────┐
order-svc-1     | │   (index)   │          Values          │
order-svc-1     | ├─────────────┼──────────────────────────┤
order-svc-1     | │  App Name   │  'Tenderd-Orders-SVC'    │
order-svc-1     | │ Server Name │     'd93db3aaaa4c'       │
order-svc-1     | │ Start Time  │ '7/23/2024, 9:05:49 AM'  │
order-svc-1     | │ Server URL  │ 'http://localhost:5001'  │
order-svc-1     | └─────────────┴──────────────────────────┘
order-svc-1     |
frontend-svc-1  | 7/23/2024, 9:05:49 AM-INFO Users service URL: http://user-svc:5000/users
frontend-svc-1  | 7/23/2024, 9:05:49 AM-INFO Order service URL: http://order-svc:5001/orders
frontend-svc-1  | 7/23/2024, 9:05:49 AM-INFO Initializing Prometheus metrics
frontend-svc-1  | 7/23/2024, 9:05:49 AM-INFO Service Information:
frontend-svc-1  |
frontend-svc-1  | ┌─────────────┬──────────────────────────┐
frontend-svc-1  | │   (index)   │          Values          │
frontend-svc-1  | ├─────────────┼──────────────────────────┤
frontend-svc-1  | │  App Name   │ 'Tenderd-Frontend-SVC'   │
frontend-svc-1  | │ Server Name │     'dc506319431e'       │
frontend-svc-1  | │ Start Time  │ '7/23/2024, 9:05:49 AM'  │
frontend-svc-1  | │ Server URL  │ 'http://localhost:8080'  │
frontend-svc-1  | └─────────────┴──────────────────────────┘
frontend-svc-1  |

v View in Docker Desktop   o View Config   w Enable Watch
```

networks: default: driver: bridge ```

- see REAEME.md for more details.

Google Cloud Platform

This services deployed in Google Cloud Platform (GCP) using Google Kubernetes Engine (GKE). This feature is integrated into the CI/CD pipeline to ensure secure and efficient deployment. Also GCP bucket is used to host terrafom lock file, for Github Actions to keep track the status of deployed resources and to be used later to destroy resources (with GitHub Actions as well)

services used:

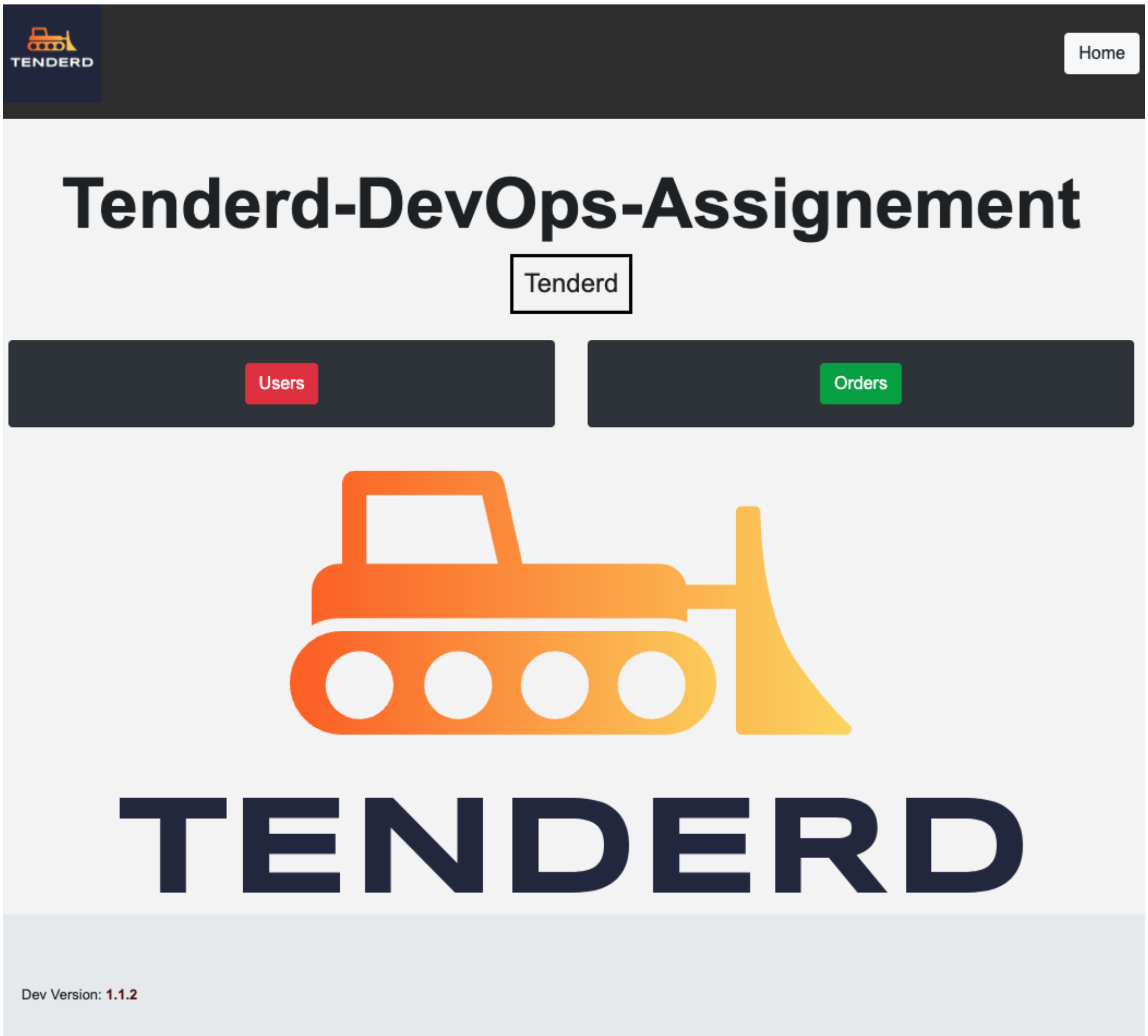
`- Google Kubernetes Engine - Google Services and load balancing and ingress - IAM and Admin - Google VPC - Google DNS - Google Storage`

Future Enhacments

- Enhancing Logging: Integrate ELK stack for comprehensive log management.
- Security Scanning: Add security scanning tools for Docker images, and use cert manager to use https protocol.

Check the current services running

- Tenderd Frontend - URL Go
- Grafana Dashboards for running apps and k8 ifra Go
- Promethues Dashboards Go



Tenderd-DevOps-Assignement

Tenderd

Dev Version: 1.1.2