

Nama : M. Muammar

project github : <https://github.com/aamdsam/sltr-devop-test>

1. Simple Coding menggunakan express js

```
JS server.js X Dockerfile {} package.json {} deploy.yml 2 docker-compose.yml Dockerfile (Working Tree)

JS server.js > ...
1  const express = require('express');
2  const app = express();
3
4  app.get('/welcome/:name?', (req, res) => {
5    const name = req.params.name || "Anonymous";
6    res.send(`Selamat datang ${name}`);
7  });
8
9  app.listen(5000, () => {
10    console.log('Server berjalan di port 5000');
11  });
```

2. Container Docker

- Docker build image manually

```
JS server.js X Dockerfile X {} package.json {} deploy.yml 2 docker-compose.yml Dockerfile (Working Tree) .dockerignore .gitignore ! configmap.yaml !

Dockerfile
1 FROM node:18-alpine AS builder
2
3 WORKDIR /app
4
5 COPY package.json package-lock.json ./
6 RUN npm install --only=production
7 COPY . .
8
9 FROM node:18-alpine
10 WORKDIR /app
11 COPY --from=builder /app /app
12 EXPOSE 5000
13 CMD ["node", "server.js"]

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE PORTS

See 'docker buildx build --help'.

Usage: docker buildx build [OPTIONS] PATH | URL | -

Start a build
PS D:\Projects\others\test\sltr-devops-test> docker build -t local-testing/welcome .
[+] Building 4.0s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 268B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerignore
=> => transferring context: 76B
=> [internal] load build context
=> => transferring context: 7.01kB
=> [builder 1/5] FROM docker.io/library/node:18-alpine@sha256:e0340f26173b41066d68e3fe9bfbdb6571ab3cad0a4272919a52e36f4ae56925
=> CACHED [builder 2/5] WORKDIR /app
=> CACHED [builder 3/5] COPY package.json package-lock.json ./
=> CACHED [builder 4/5] RUN npm install --only=production
=> [builder 5/5] COPY . .
=> [stage-1 3/3] COPY --from=builder /app /app
=> exporting to image
=> exporting layers
=> writing image sha256:1749df8c799ebff3383bfbdb05dfe4e2a34a5eb4e757f0bd53ed6190cf1b83f0f
=> naming to docker.io/local-testing/welcome

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/a2oc16egrb8vklzg45xeraso3

What's next:
View a summary of image vulnerabilities and recommendations + docker scout quickview
```

- Docker compose run port 8000

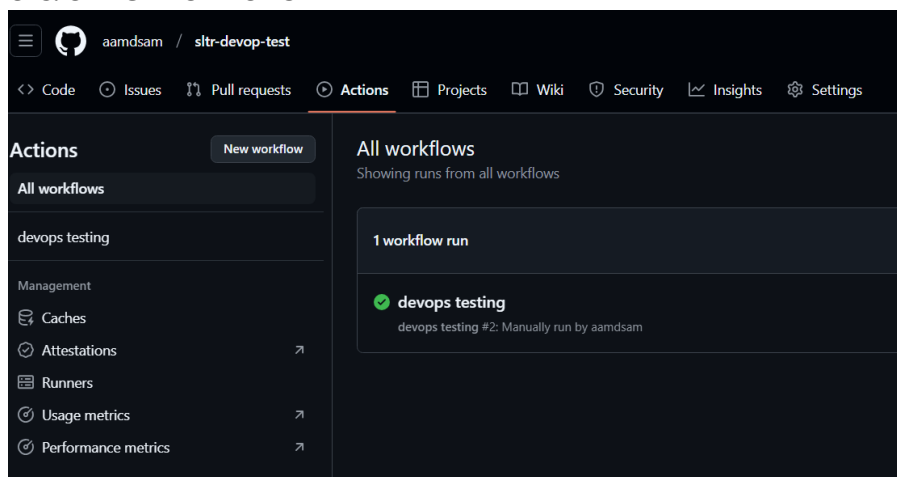
JS server.js Dockerfile package.json deploy.yml 2 docker-compose.yml X Dockerfile

docker-compose.yml1 version: '3.8'2 services:3 welcome-app:4 build: .5 ports:6 - "8000:5000"7 environment:8 - NODE_ENV=production9 restart: unless-stopped10 networks:11 - welcome-network1213 networks:14 welcome-network:15 driver: bridge

TERMINAL PROBLEMS 2 OUTPUT DEBUG CONSOLE PORTS

=> [builder 1/5] FROM docker.io/library/node:18-alpine@sha256:e0340f26173b41066d68e3fe9bfbdb6=> CACHED [builder 2/5] WORKDIR /app=> CACHED [builder 3/5] COPY package.json package-lock.json ./=> CACHED [builder 4/5] RUN npm install --only=production=> [builder 5/5] COPY . .=> [stage-1 3/3] COPY --from=builder /app /app=> exporting to image=> => exporting layers=> => writing image sha256:1749df8c799ebff3383bfbd05dfe4e2a34a5eb4e757f0bd53ed6190cf1b83f0f=> => naming to docker.io/local-testing/welcomeView build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/a2oc16egrb8vkWhat's next:View a summary of image vulnerabilities and recommendations -> docker scout quickviewPS D:\Projects\others\test\sltr-devops-test> docker-compose uptime="2025-02-22T08:23:30+07:00" level=warning msg="D:\Projects\others\test\sltr-devops-test\nored, please remove it to avoid potential confusion"[+] Running 1/0✔ Container sltr-devops-test-welcome-app-1 CreatedAttaching to welcome-app-1welcome-app-1 | Server berjalan di port 5000

3. CI & CD - GITHUB ACTION



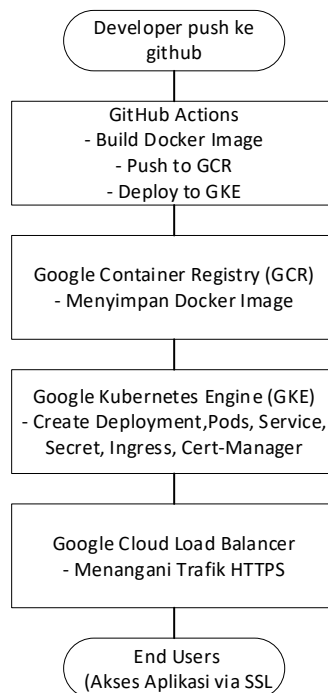
4. KUBERNETES

The image shows a VS Code editor with a project named 'sltr-devops-test'. The file explorer on the left shows a directory structure with files like `deployment.yaml`, `ingress.yaml`, `secret.yaml`, and `service.yaml`. The main editor displays the content of `deployment.yaml`, which defines a Kubernetes deployment for a 'welcome' app using the 'aamdsam/sltr-devops-testing:latest' image, exposing port 5000.

Below the editor, the 'TERMINAL' panel shows the execution of `docker compose up` and `kubectl apply` commands. The output indicates that the container 'welcome-app-1' was created and is running on port 5000. It also shows the successful creation of various Kubernetes resources like certificates, configmaps, and services.

At the bottom, the Docker Desktop overview is visible, showing a summary of the Kubernetes cluster state: 18 Pods (15 Running, 3 Succeeded), 6 Deployments (6 Running), 1 Daemon Set (1 Running), 0 Stateful Sets, 9 Replica Sets (3 Unknown, 6 Running), and 3 Jobs (3 Succeeded). The left sidebar of Docker Desktop provides navigation options for clusters, workloads, and configuration.

5. CLOUD INFRA - GCP



6. Cara Deploy Welcome Test App di Kubernetes (Lokal)

- Pastikan aplikasi seperti git, docker dan Kubernetes sudah terinstall di local PC
- Clone Repository dari GitHub : <https://github.com/aamdsam/sltr-devop-test.git>
- Masuk ke di rektori hasil clone atau directory sltr-devops-test
- Jalankan perintah kubectl apply untuk deploy deployment, service, secret, configmap, ingress dan ssl issuer nya dengan cara : ***kubectl apply -f kubernetes/***
- Pastikan semua deployment, pod, service, secret, configmap, ingress, dan nginx ingress sudah ada dan status running atau tidak ada error
- Apabila nginx-controller belum ada bisa diinstall bisa dibuat menggunakan helm ingress-nginx atau instal manual dengan command kubectl seperti ini: ***kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/main/deploy/static/provider/cloud/deploy.yaml***
- Tambahkan domain local di etc host berikut ini: 127.0.0.1 welcome.local
- Akses domain local dengan url: <https://welcome.local/welcome/test> atau <https://welcome.local/welcome>