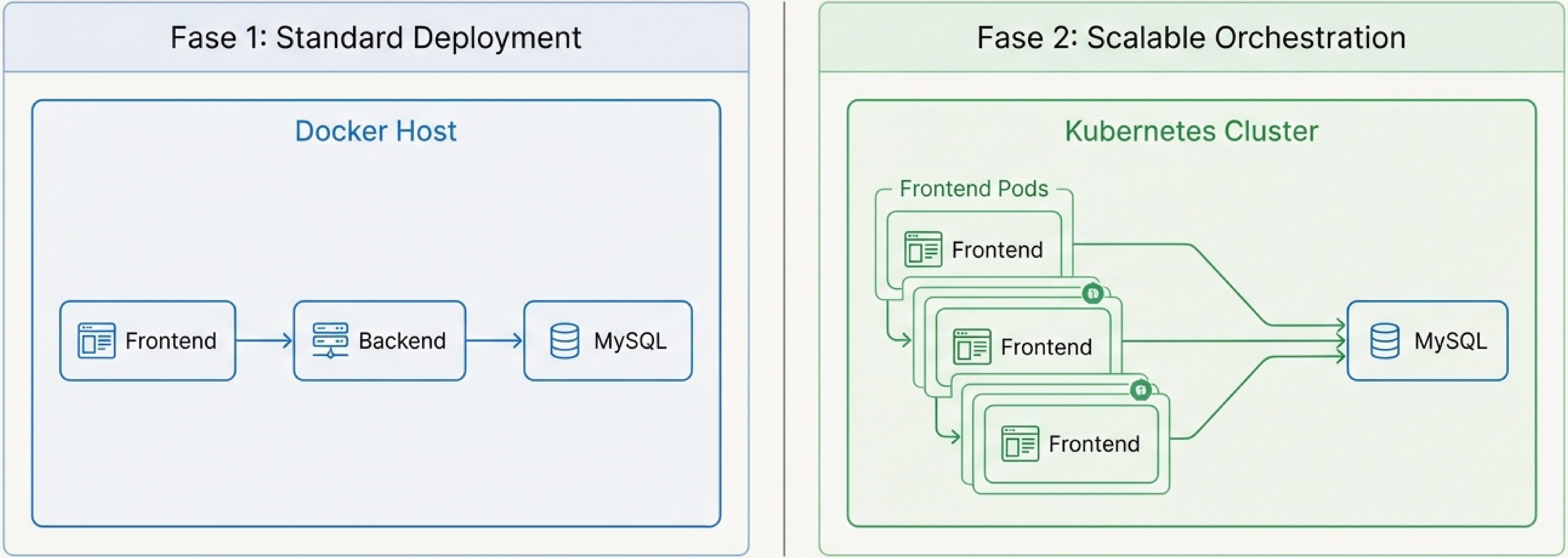
DevOps Portfolio: Project Wayshub

Evolusi Arsitektur dari Docker Monolithic ke Hybrid Kubernetes Cluster

Fauzan Prima Mukti

Ringkasan Evolusi Arsitektur



Project Wayshub adalah platform video streaming yang berevolusi menjadi sistem **Hybrid,** di mana workload Frontend berjalan paralel di **Docker dan Kubernetes** (k3s) dengan mengonsumsi Backend Database **yang sama.**

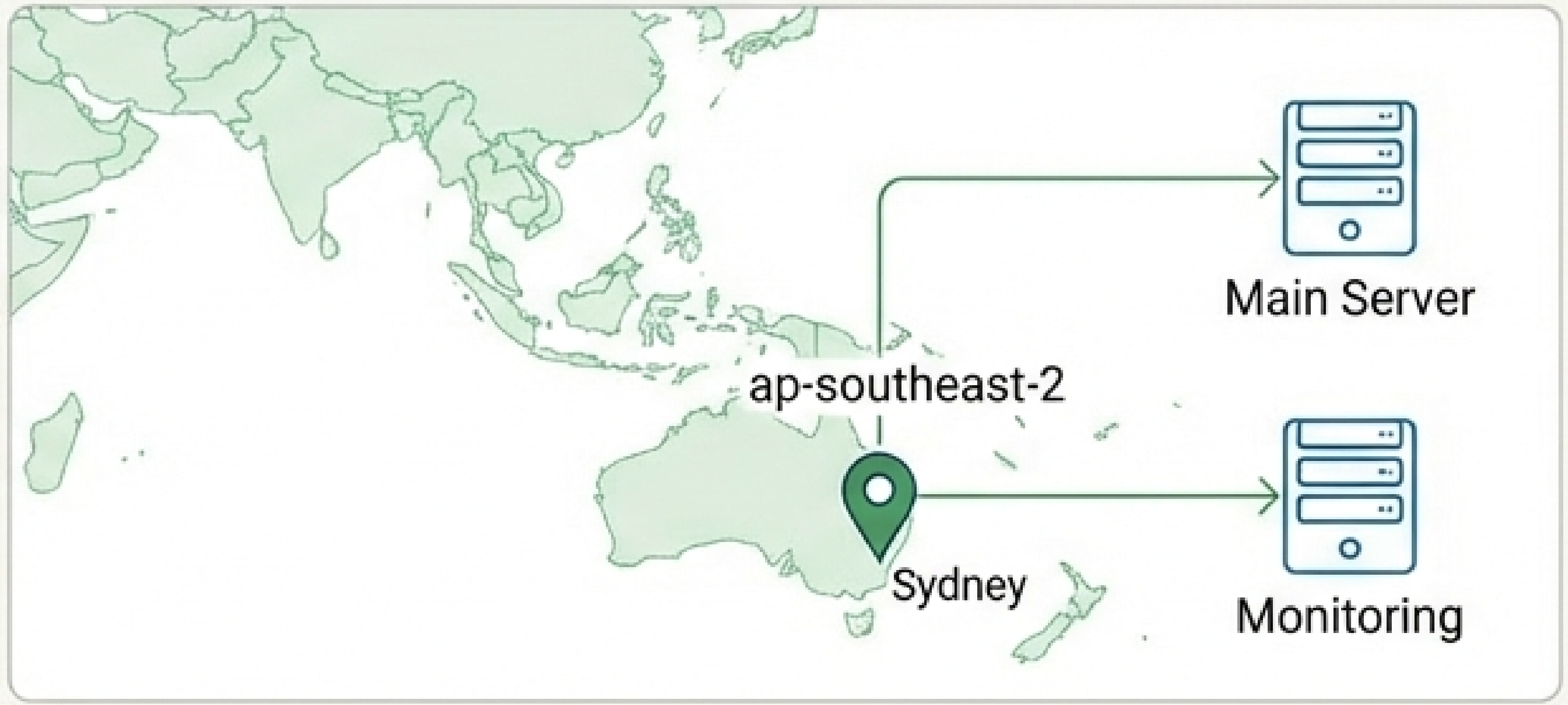
Provisioning Infrastruktur AWS dengan Terraform

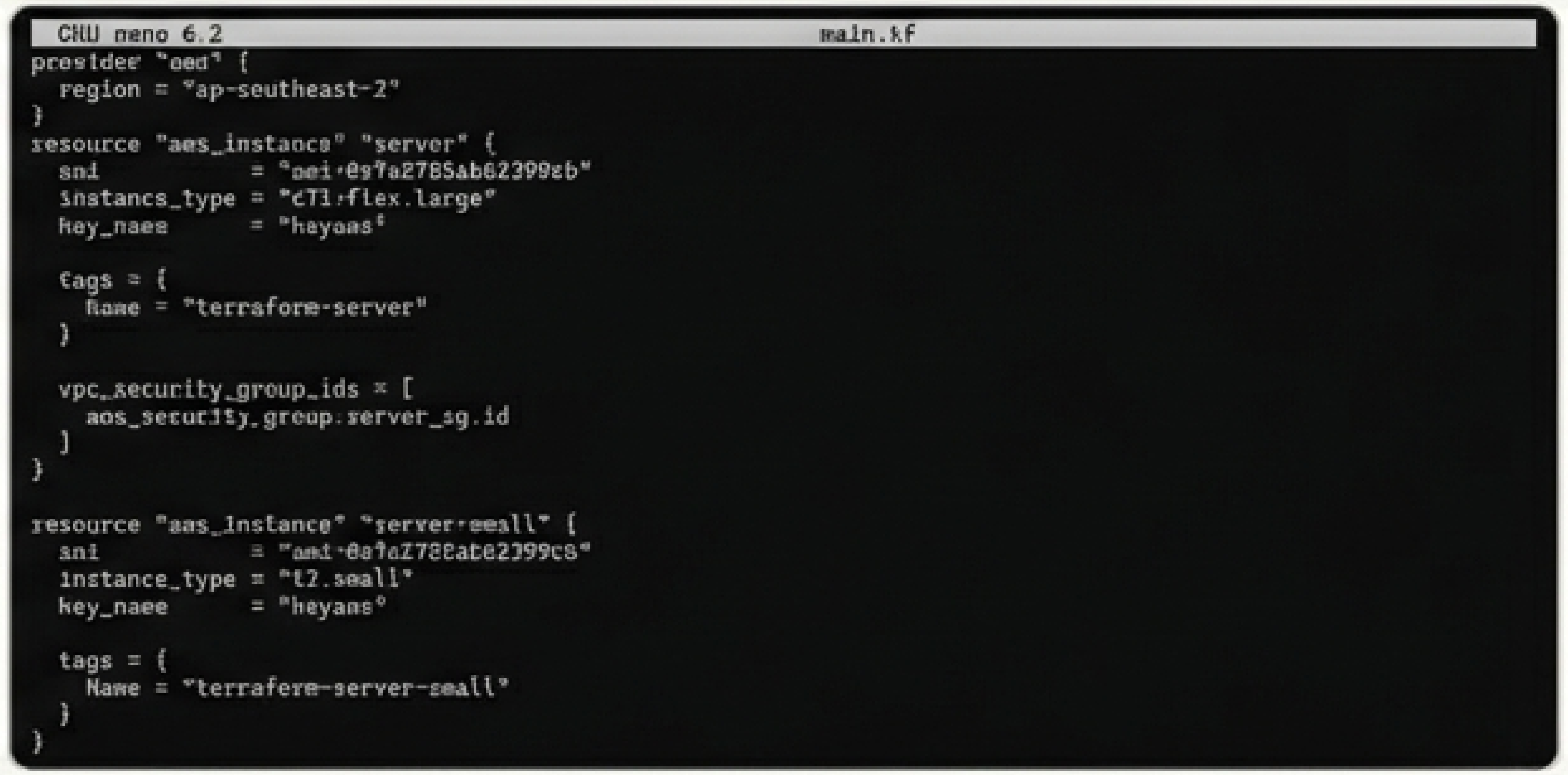
Provider: AWS (Region ap-southeast-2)

Compute:

- EC2 `c7i-flex.large` (Main Server)
- EC2 `t3.small` (Monitoring/Registry)

Security: Security Group `allow-ssh-http` membuka port 22, 80, 3000, 5000, 9090.





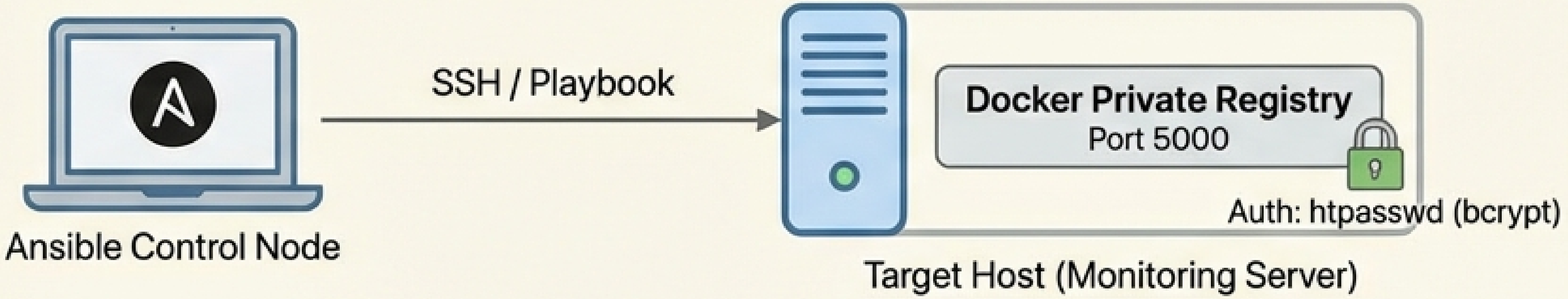
```
presidee "oed" {
  region = "ap-seutheast-2"
}
resource "aes_instance" "server" {
  sni           = "oei-e9fa2785ab62399eb"
  instancs_type = "c71:flex.large"
  hey_naee      = "heyaas"

  tags = {
    Rame = "terrafore-server"
  }

  vpc_security_group_ids = [
    aos_secutity_group:server_sg.id
  ]
}

resource "aas_instance" "server-eeall" {
  sni           = "ami-0afa2788ate2399c8"
  instance_type = "t2.seall"
  hey_naee      = "heyans"

  tags = {
    Name = "terrafere-server-seall"
  }
}
```

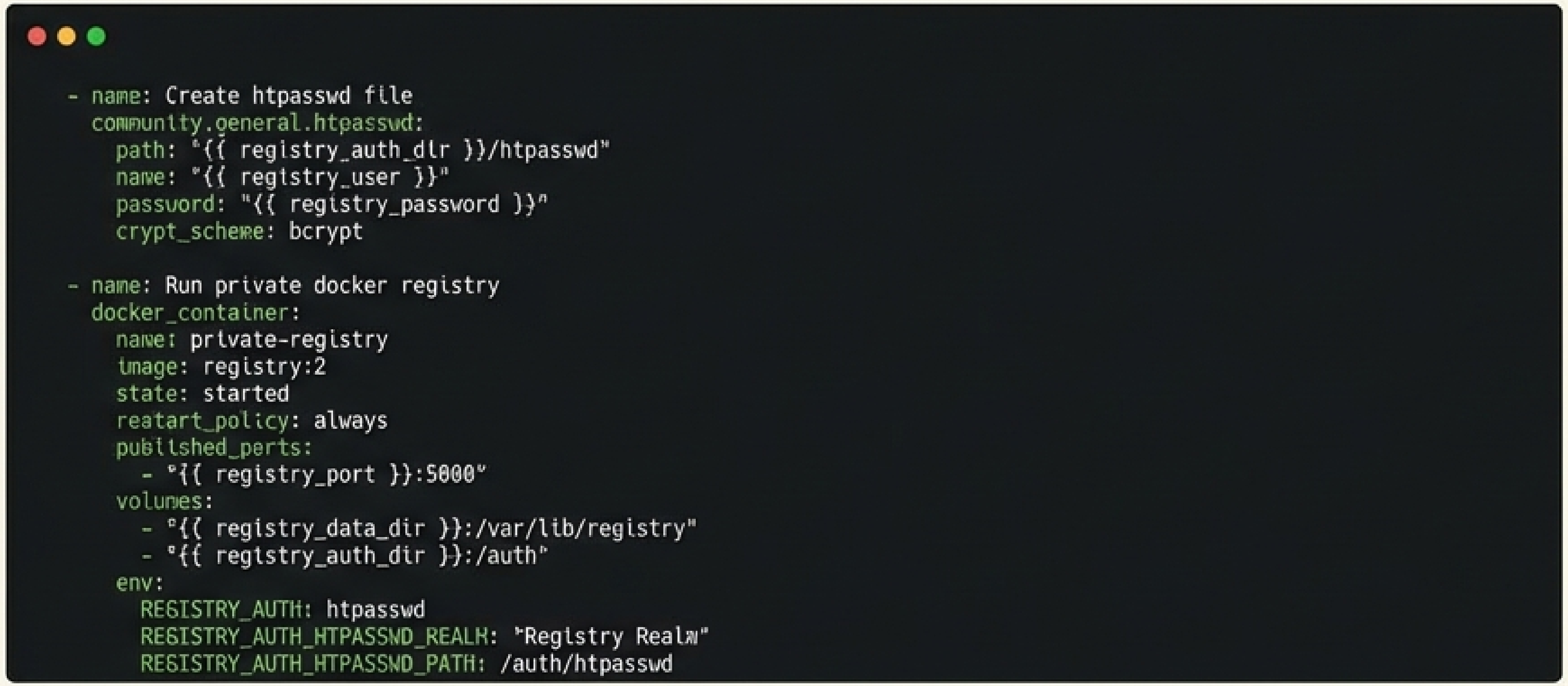
Configuration Management & Private Registry via Ansible

Otomatisasi: Ansible Playbook menjamin setup environment yang konsisten.

Private Registry: Setup Docker Registry pada port 5000 dengan autentikasi wajib.

Persistence: Volume Docker dikonfigurasi untuk menyimpan image dan file auth secara permanen.





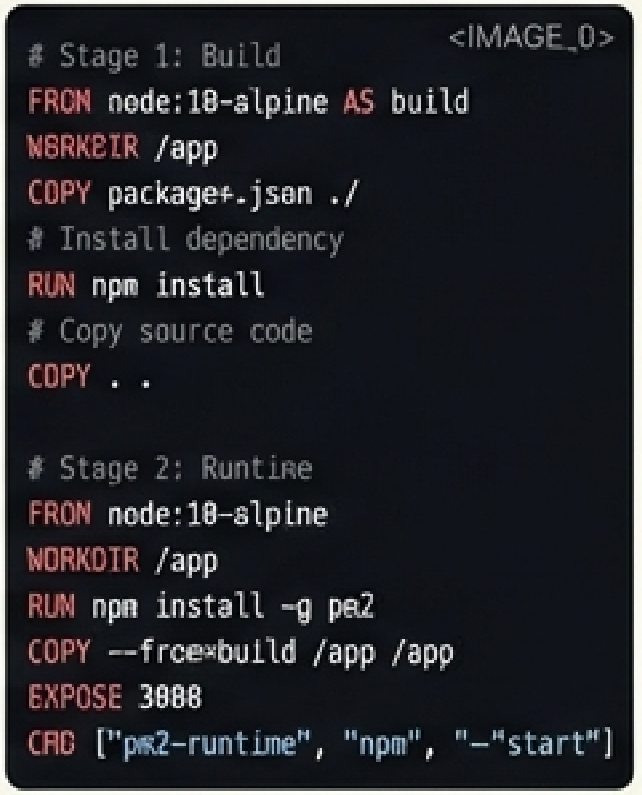
```
- name: Create htpasswd file
  community.general.htpasswd:
    path: "{{ registry_auth_dir }}/htpasswd"
    name: "{{ registry_user }}"
    passvord: "{{ registry_password }}"
    crypt_scheme: bcrypt

- name: Run private docker registry
  docker_container:
    name: private-registry
    image: registry:2
    state: started
    reatart_policy: always
    published_perts:
      - "{{ registry_port }}:5000"
    volumes:
      - "{{ registry_data_dir }}:/var/lib/registry"
      - "{{ registry_auth_dir }}:/auth"
    env:
      REGISTRY_AUTH: htpasswd
      REGISTRY_AUTH_HTPASSWD_REALM: "Registry Realm"
      REGISTRY_AUTH_HTPASSWD_PATH: /auth/htpasswd
```

Strategi Containerization & Multi-Stage Build

Frontend

→ Port **3000**

Multi-stage build: Install dependencies -> Run via **pm2-runtime** on **node:10-alpine**



<IMAGE_0>

```
# Stage 1: Build
FRON node:18-alpine AS build
N8RK8IR /app
COPY package+.jsen ./
# Install dependency
RUN npm install
# Copy source code
COPY . .

# Stage 2: Runtime
FRON node:18-alpine
NORKDIR /app
RUN npm install -g pm2
COPY --frce×build /app /app
EXPOSE 3808
CRD ["pm2-runtime", "npm", "—"start"]
```

Backend

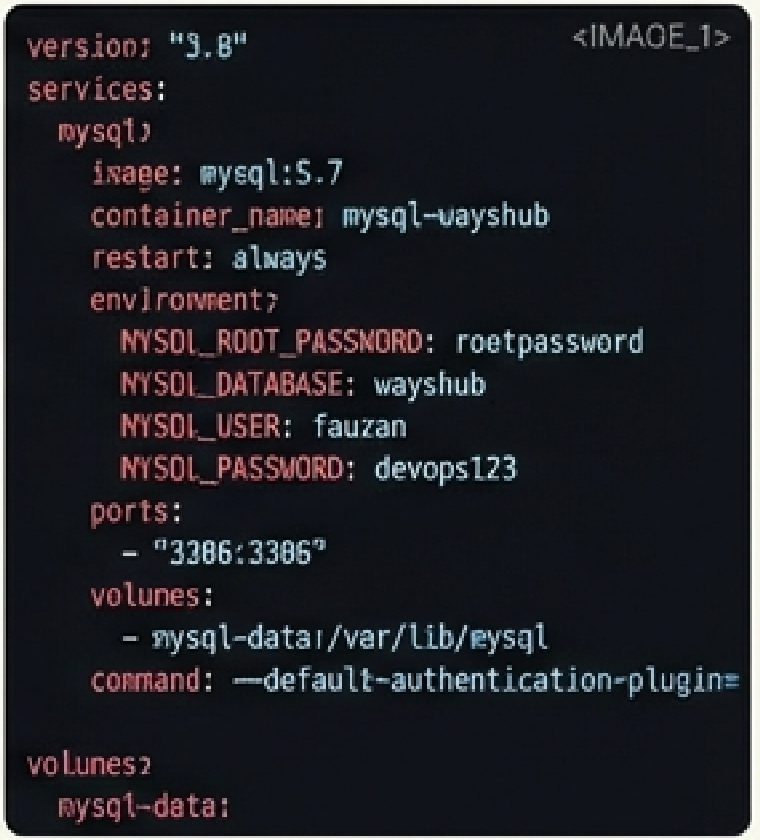
→ Port **5000**

Configured via **config.json** for Dev/Test/Prod environments

Database

→ Port **3306**

MySQL 5.7 with persistent volume **`mysql-data`**

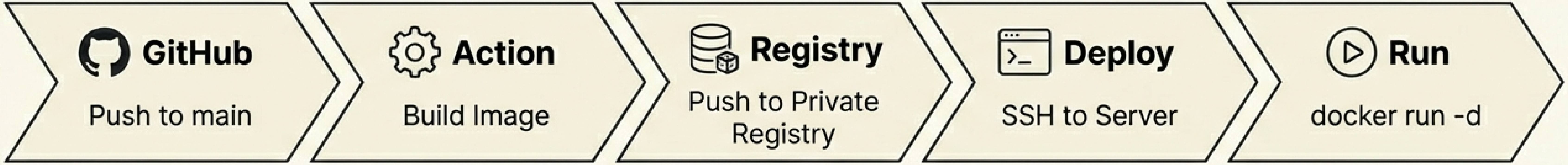


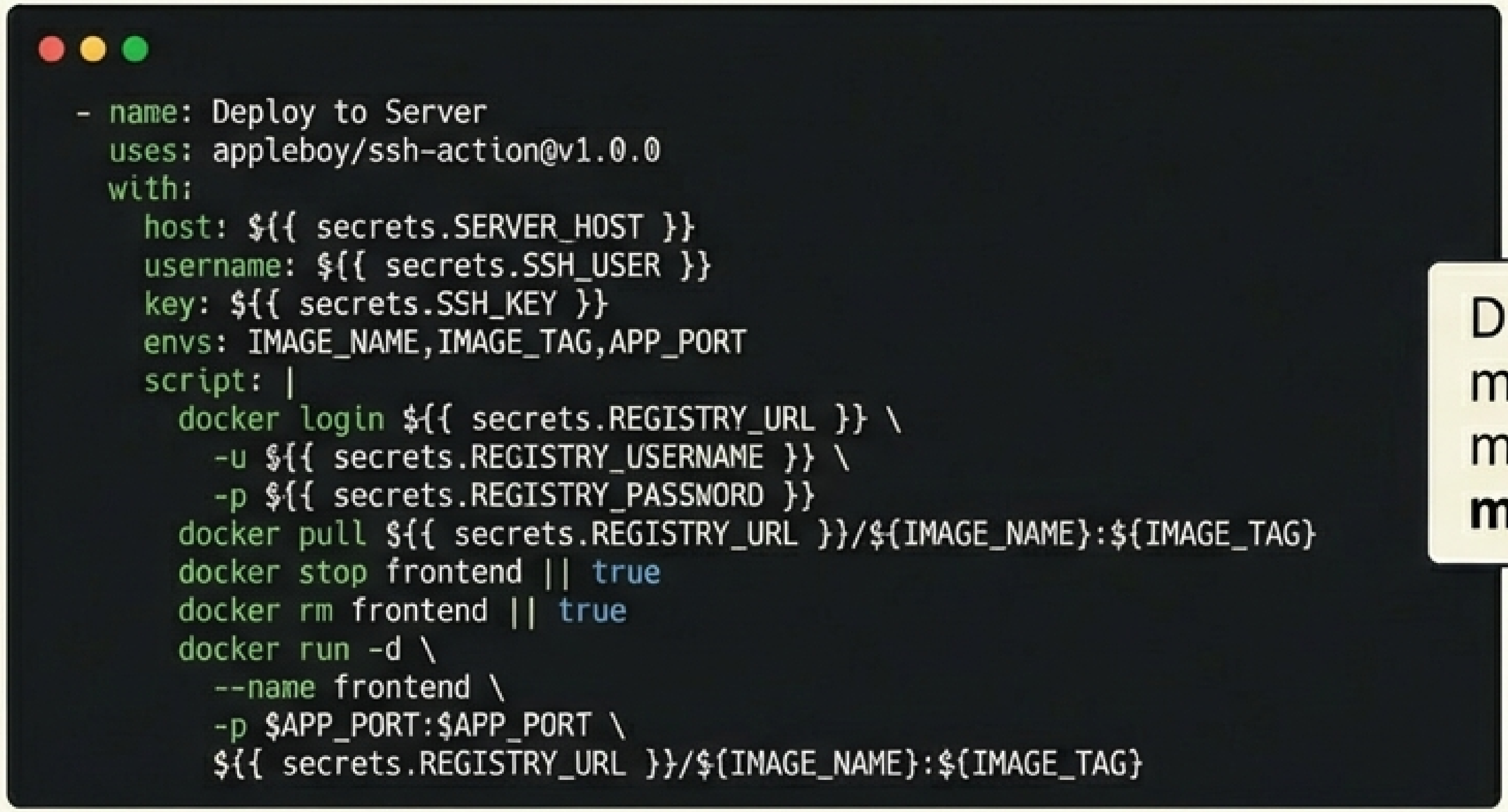
<IMAOE_1>

```
version: "3.8"
services:
  mysql:
    image: mysql:5.7
    container_name: mysql-wayshub
    restart: always
    environment:
      NYSOL_ROOT_PASSNORD: roetpassword
      NYSOL_DATABASE: wayshub
      NYSOL_USER: fauzan
      NYSOL_PASSWORD: devops123
    ports:
      - "3386:3306"
    volumes:
      - mysql-data:/var/lib/mysql
    command: --default-authentication-plugin=

volumes:
  mysql-data:
```

Engineering Editorial

CI/CD Pipeline - Docker Workflow



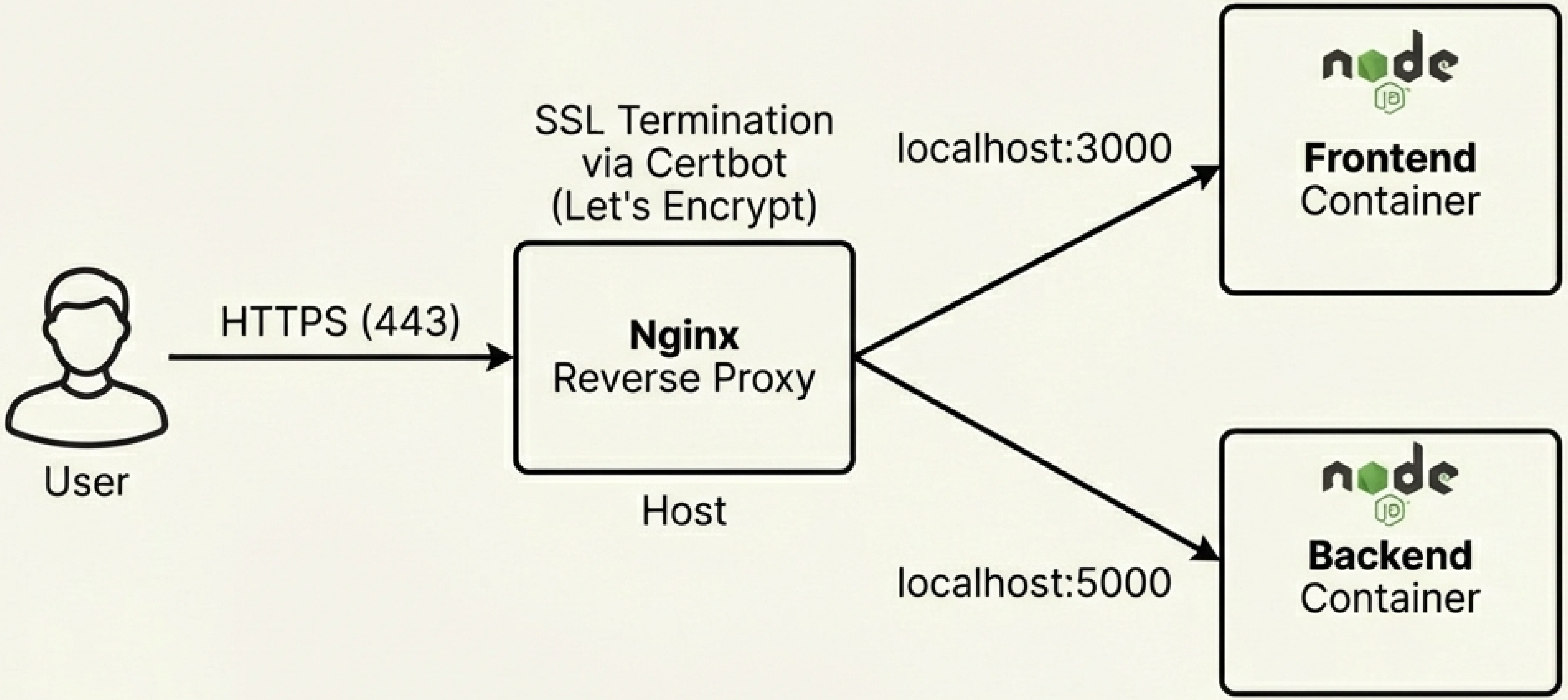


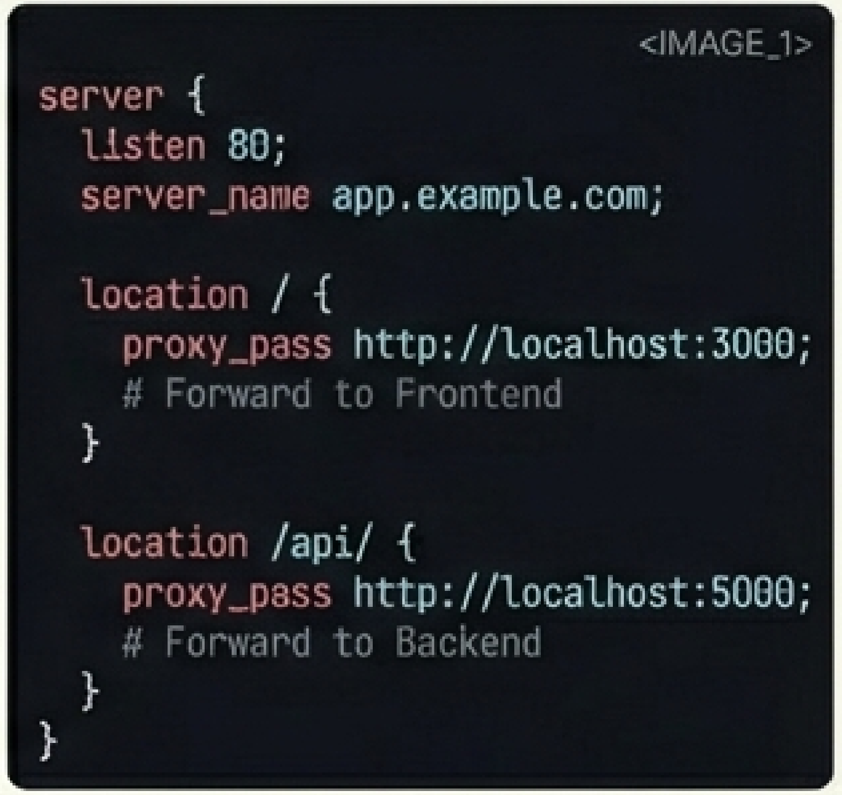
```
- name: Deploy to Server
  uses: appleboy/ssh-action@v1.0.0
  with:
    host: ${{ secrets.SERVER_HOST }}
    username: ${{ secrets.SSH_USER }}
    key: ${{ secrets.SSH_KEY }}
    envs: IMAGE_NAME,IMAGE_TAG,APP_PORT
    script: |
      docker login ${{ secrets.REGISTRY_URL }} \
        -u ${{ secrets.REGISTRY_USERNAME }} \
        -p ${{ secrets.REGISTRY_PASSWORD }}
      docker pull ${{ secrets.REGISTRY_URL }}/${IMAGE_NAME}:${IMAGE_TAG}
      docker stop frontend || true
      docker rm frontend || true
      docker run -d \
        --name frontend \
        -p $APP_PORT:$APP_PORT \
        ${{ secrets.REGISTRY_URL }}/${IMAGE_NAME}:${IMAGE_TAG}
```

Deployment dilakukan secara imperative menggunakan perintah SSH remote untuk menghentikan container lama dan **menjalankan container baru**.

Engineering Editorial

Docker Networking: Reverse Proxy & SSL



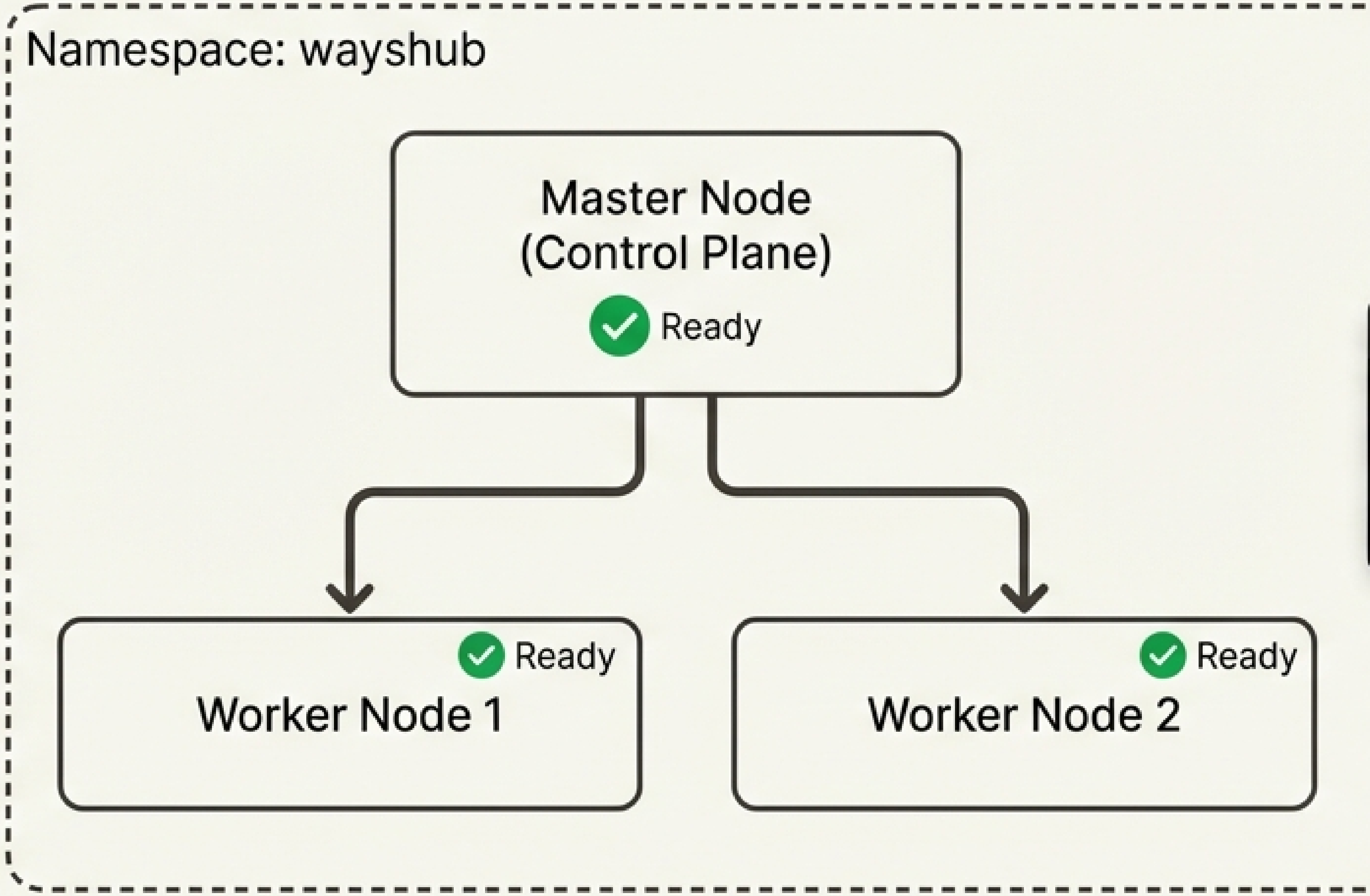


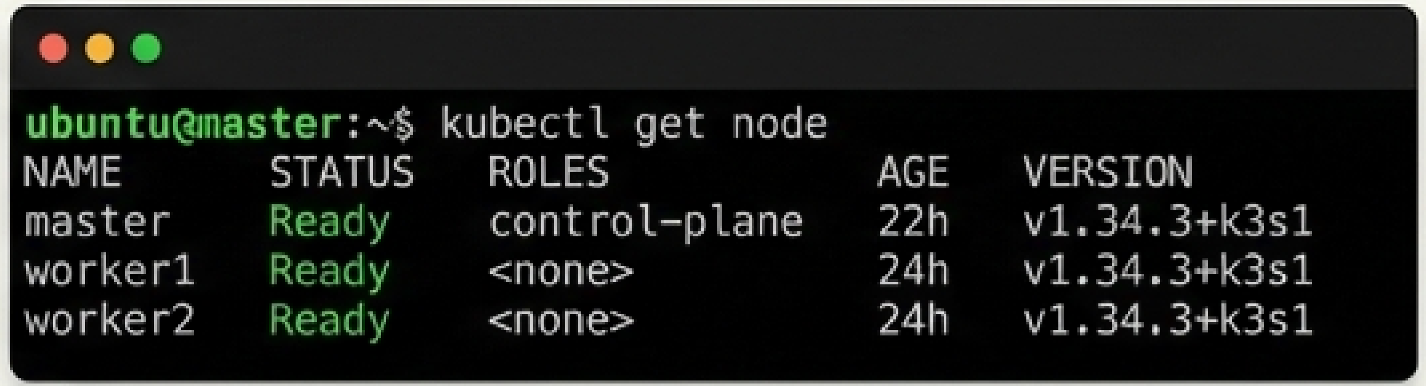
```
server {
  listen 80;
  server_name app.example.com;

  location / {
    proxy_pass http://localhost:3000;
    # Forward to Frontend
  }

  location /api/ {
    proxy_pass http://localhost:5000;
    # Forward to Backend
  }
}
```

Engineering Editorial

Evolusi ke Container Orchestration (Phase 2)

Migrasi ke Kubernetes (k3s) bertujuan untuk meningkatkan availability dan memvalidasi kemampuan orkestrasi.



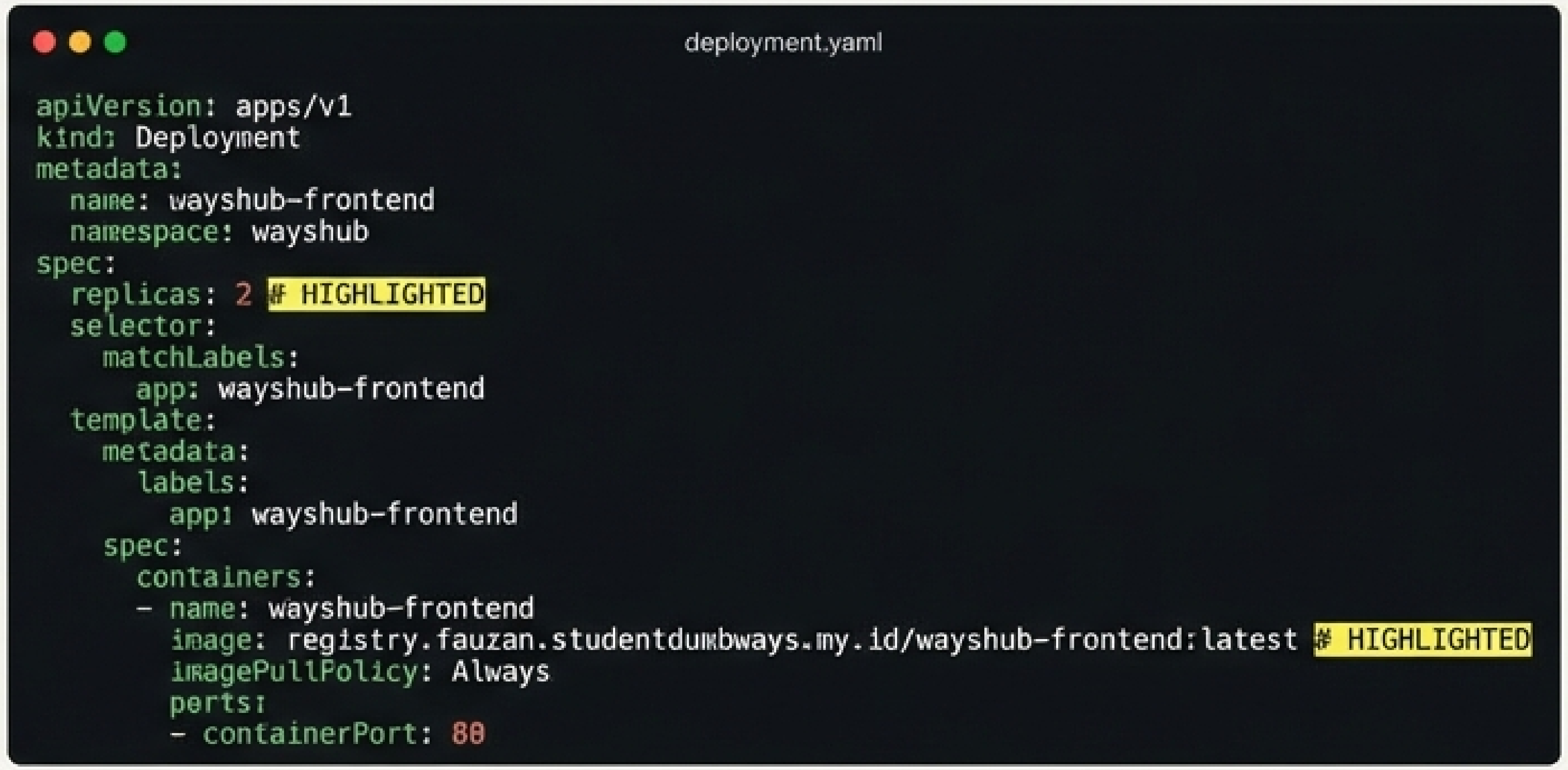


```
ubuntu@master:~$ kubectl get node
NAME       STATUS   ROLES           AGE   VERSION
master     Ready    control-plane   22h   v1.34.3+k3s1
worker1    Ready    <none>          24h   v1.34.3+k3s1
worker2    Ready    <none>          24h   v1.34.3+k3s1
```

Kubernetes Workload: Deployment & Services

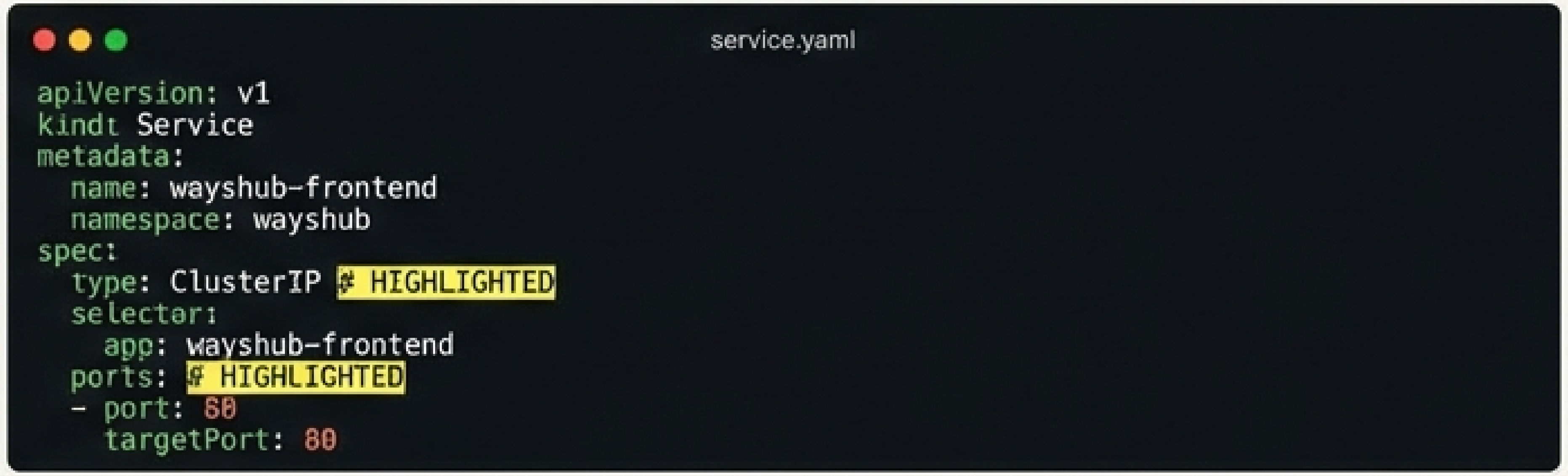
Deployment Strategy:

- Replicas: 2 (Redundancy)
- Image: Pull dari Private Registry
- Strategy: Rolling Update



deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wayshub-frontend
  namespace: wayshub
spec:
  replicas: 2 # HIGHLIGHTED
  selector:
    matchLabels:
      app: wayshub-frontend
  template:
    metadata:
      labels:
        app: wayshub-frontend
    spec:
      containers:
      - name: wayshub-frontend
        image: registry.fauzan.studentdumbways.my.id/wayshub-frontend:latest # HIGHLIGHTED
        imagePullPolicy: Always
        ports:
        - containerPort: 80
```

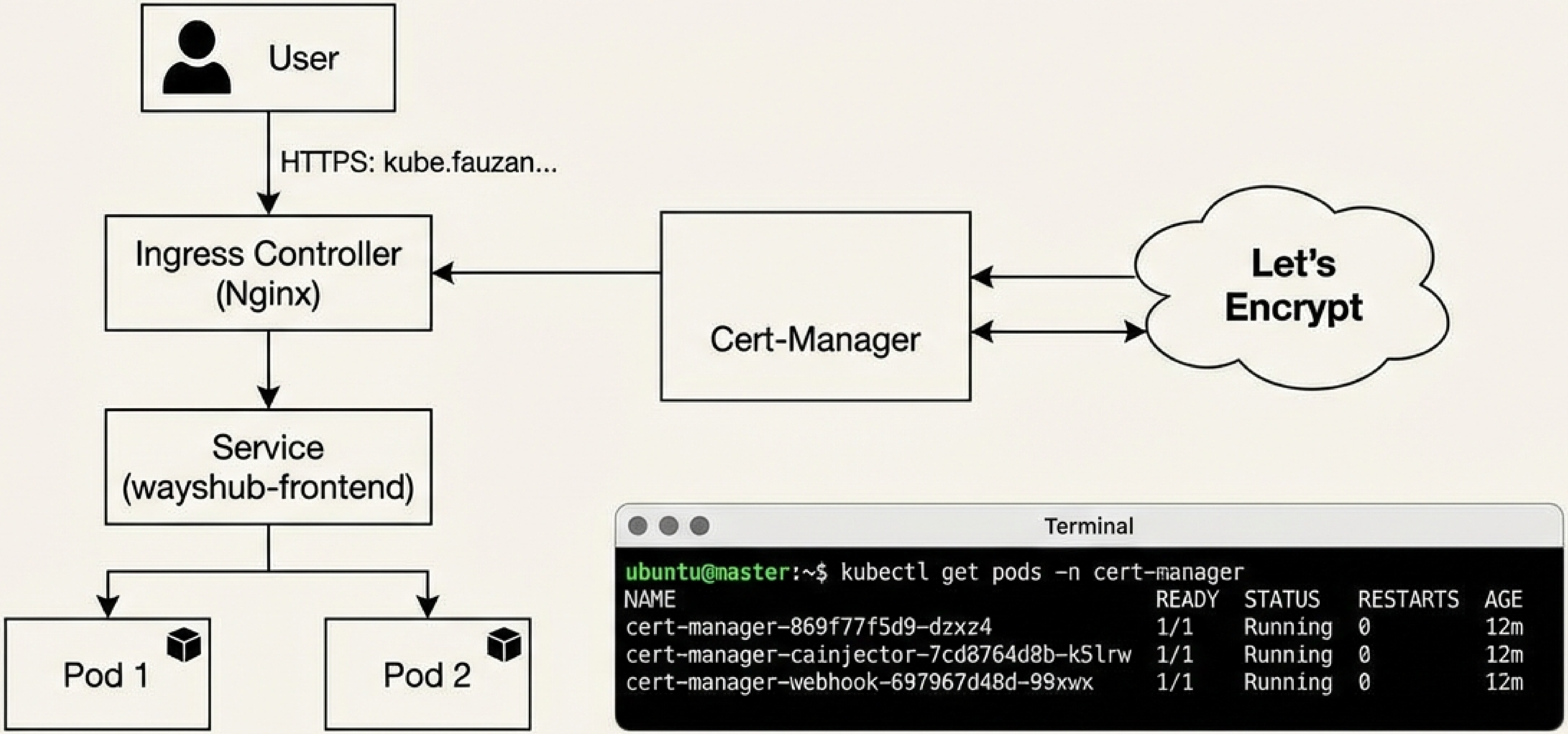


service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: wayshub-frontend
  namespace: wayshub
spec:
  type: ClusterIP # HIGHLIGHTED
  selector:
    app: wayshub-frontend
  ports: # HIGHLIGHTED
  - port: 80
    targetPort: 80
```

Engineering Editorial

Kubernetes Networking: Ingress & Cert-Manager

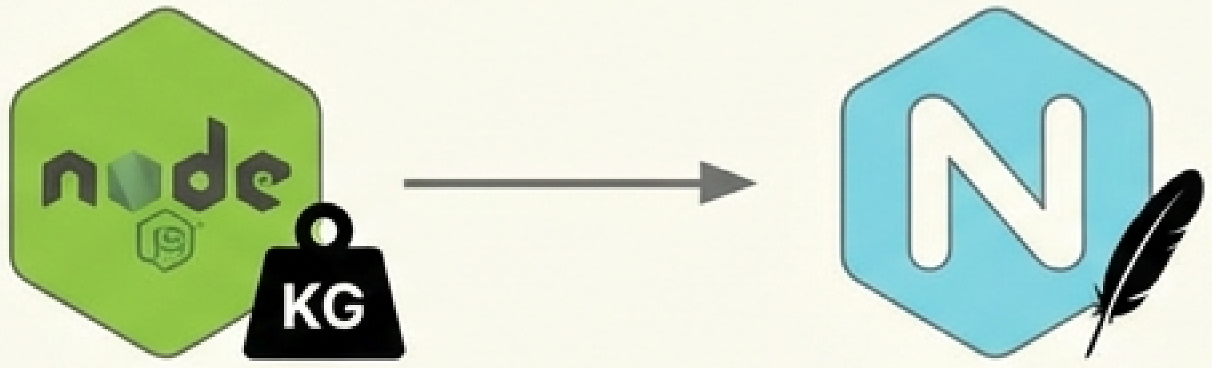


```
ubuntu@master:~$ kubectl get pods -n cert-manager
NAME                                       READY   STATUS    RESTARTS   AGE
cert-manager-869f77f5d9-dzxz4              1/1     Running   0          12m
cert-manager-cainjector-7cd8764d8b-k5lrw   1/1     Running   0          12m
cert-manager-webhook-697967d48d-99xwx      1/1     Running   0          12m
```

NotebookLM

Optimasi Container untuk Kubernetes

Refactoring Dockerfile untuk environment orkestrasi. Mengganti runtime Node.js yang berat dengan Nginx Alpine untuk serving static files.



Before: Node.js
(Heavy)

After: Nginx Alpine
(Lightweight/Static)

Refactored Dockerfile

```
# Stage 1: Build
FROM node:10-alpine AS build
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build

# Stage 2: Runtime
FROM nginx:alpine
COPY --from=build /app/build /usr/share/nginx/html
COPY nginx.conf /etc/nginx/conf.d/default.conf
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

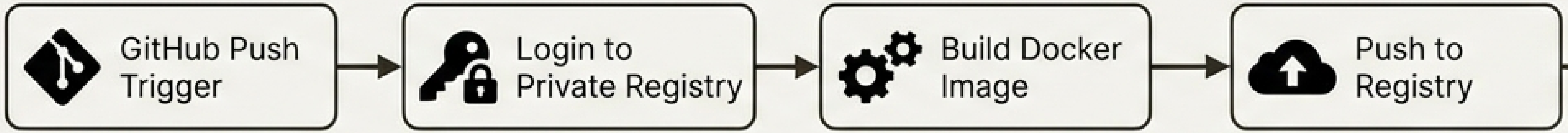
nginx.conf

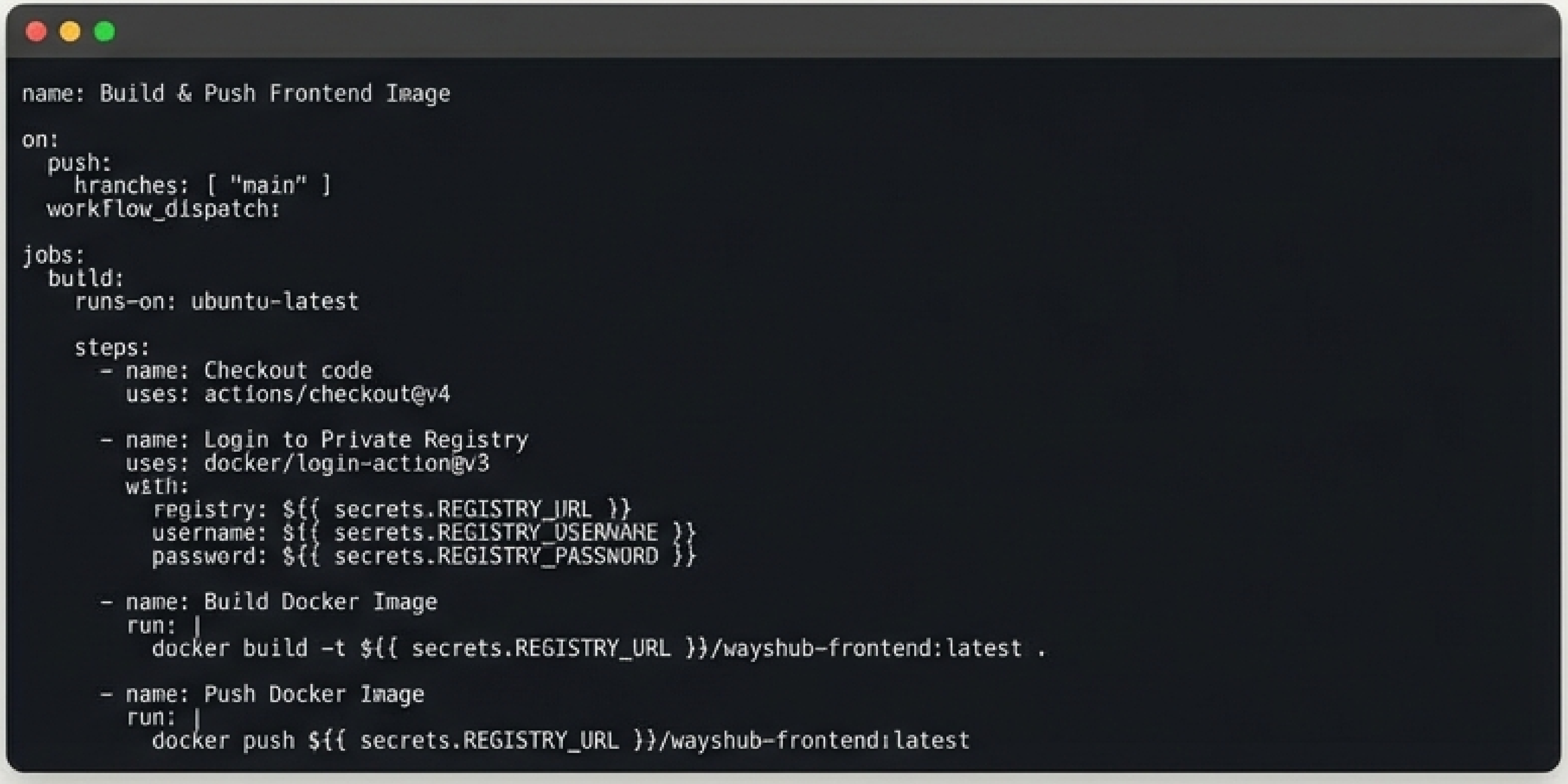
```
server {
    listen 80;
    server_name _;

    root /usr/share/nginx/html;
    index index.html;

    location / {
        try_files $uri $uri/ /index.html;
    }

    gzip on;
    gzip_types text/plain text/css application/json application/javascript
}
```

Engineering Editorial

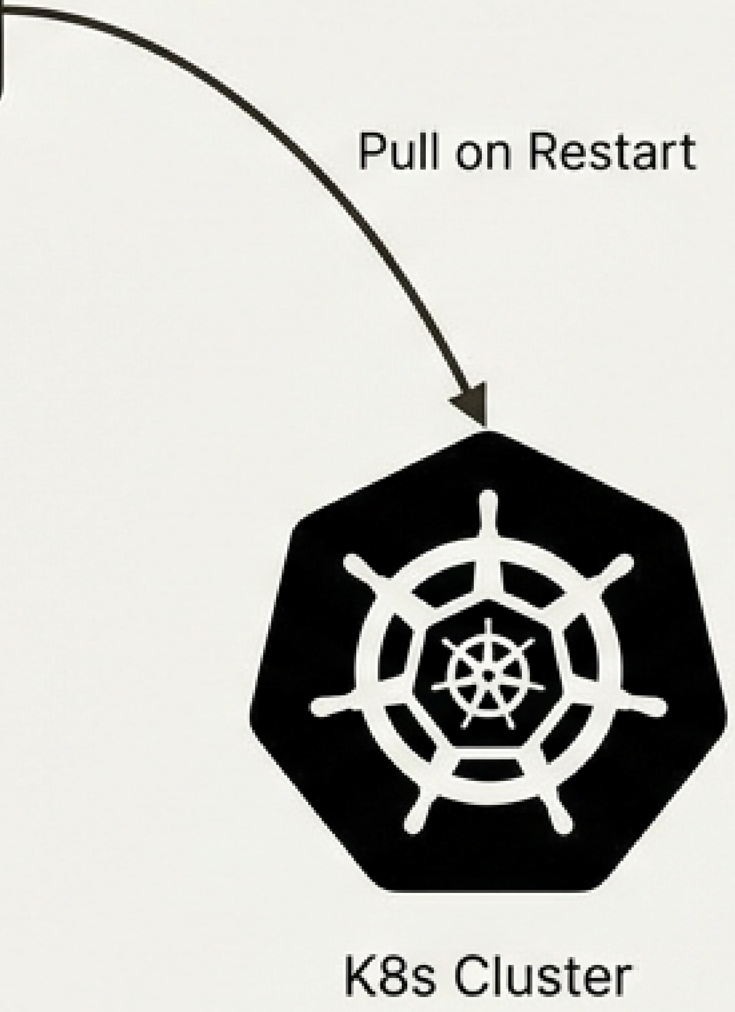
CI/CD Pipeline - Build & Push (K8s Ready)

Pipeline disederhanakan: Fokus pada Build & Push. Kubernetes akan menarik image terbaru berdasarkan `imagePullPolicy: Always`.





```
name: Build & Push Frontend Image

on:
  push:
    hranches: [ "main" ]
  workflow_dispatch:

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Login to Private Registry
        uses: docker/login-action@v3
        with:
          registry: ${{ secrets.REGISTRY_URL }}
          username: ${{ secrets.REGISTRY_USERNAME }}
          password: ${{ secrets.REGISTRY_PASSNORD }}

      - name: Build Docker Image
        run: |
          docker build -t ${{ secrets.REGISTRY_URL }}/wayshub-frontend:latest .

      - name: Push Docker Image
        run: |
          docker push ${{ secrets.REGISTRY_URL }}/wayshub-frontend:latest
```

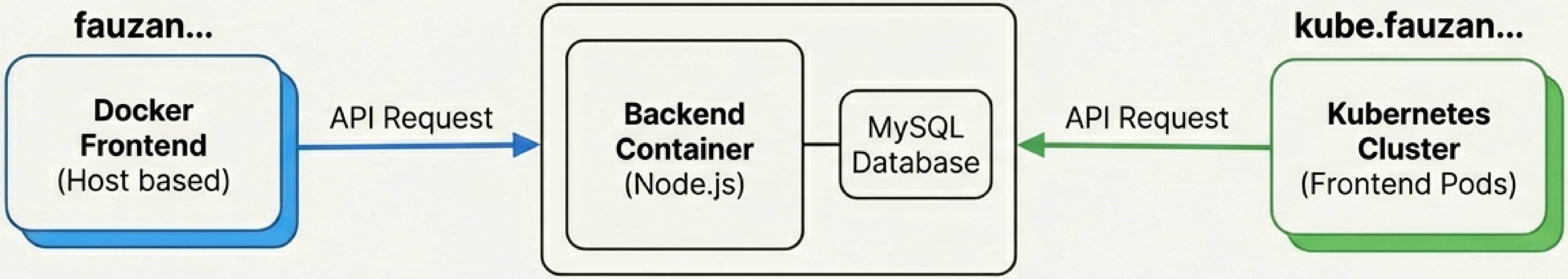


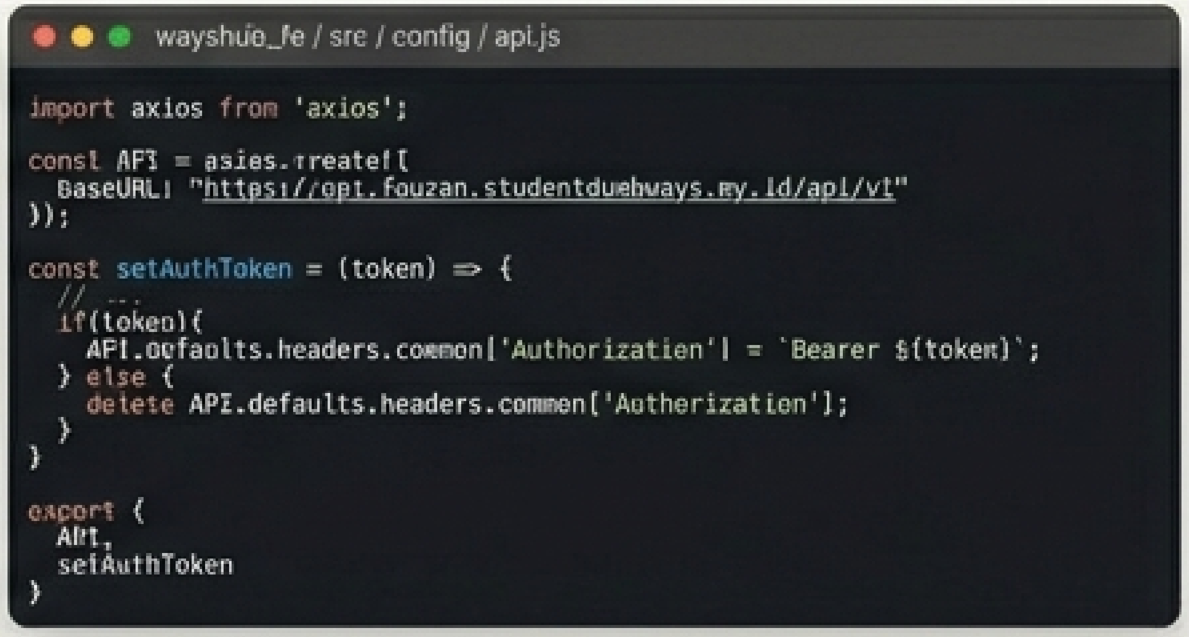
NotebookLM

Engineering Editorial

Ekosistem Hybrid: Unified Backend Strategy

Fleksibilitas arsitektur dibuktikan dengan dua frontend berbeda platform yang mengonsumsi satu backend API yang sama.


fauzan...
Docker Frontend (Host based)
API Request
Backend Container (Node.js)
MySQL Database
API Request
kube.fauzan...
Kubernetes Cluster (Frontend Pods)

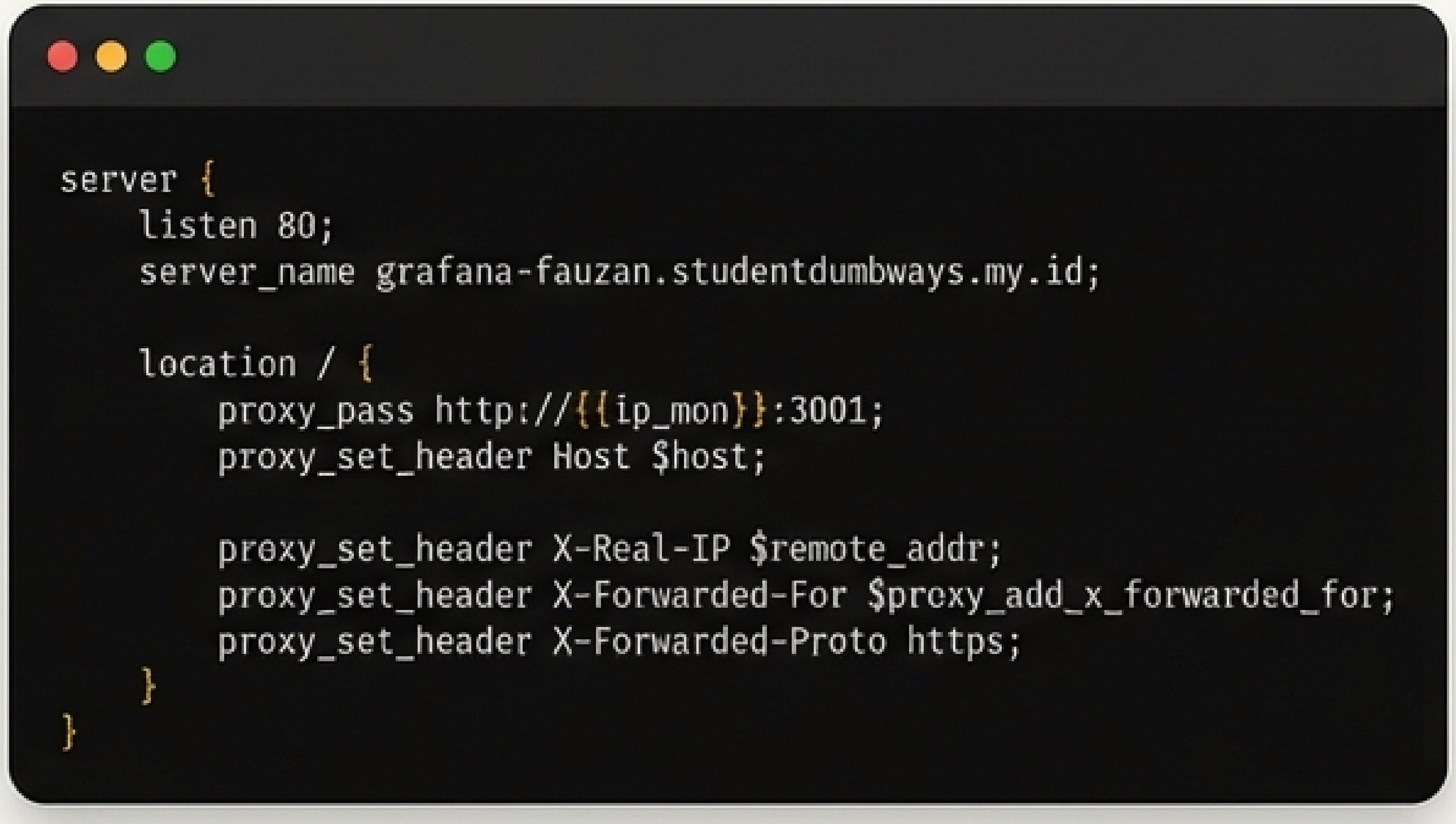


```
import axios from 'axios';

const AFI = asies.rreate![
  BaseURL: "https://opi.fouzan.studentduebways.my.id/api/vt"
});

const setAuthToken = (token) => {
  // ...
  if(token){
    API.oefaolts.headers.common['Authorizatien'] = `Bearer ${token}`;
  } else {
    delete API.defaults.headers.commen['Aotherization'];
  }
}

export {
  AΠt,
  setAuthToken
}
```

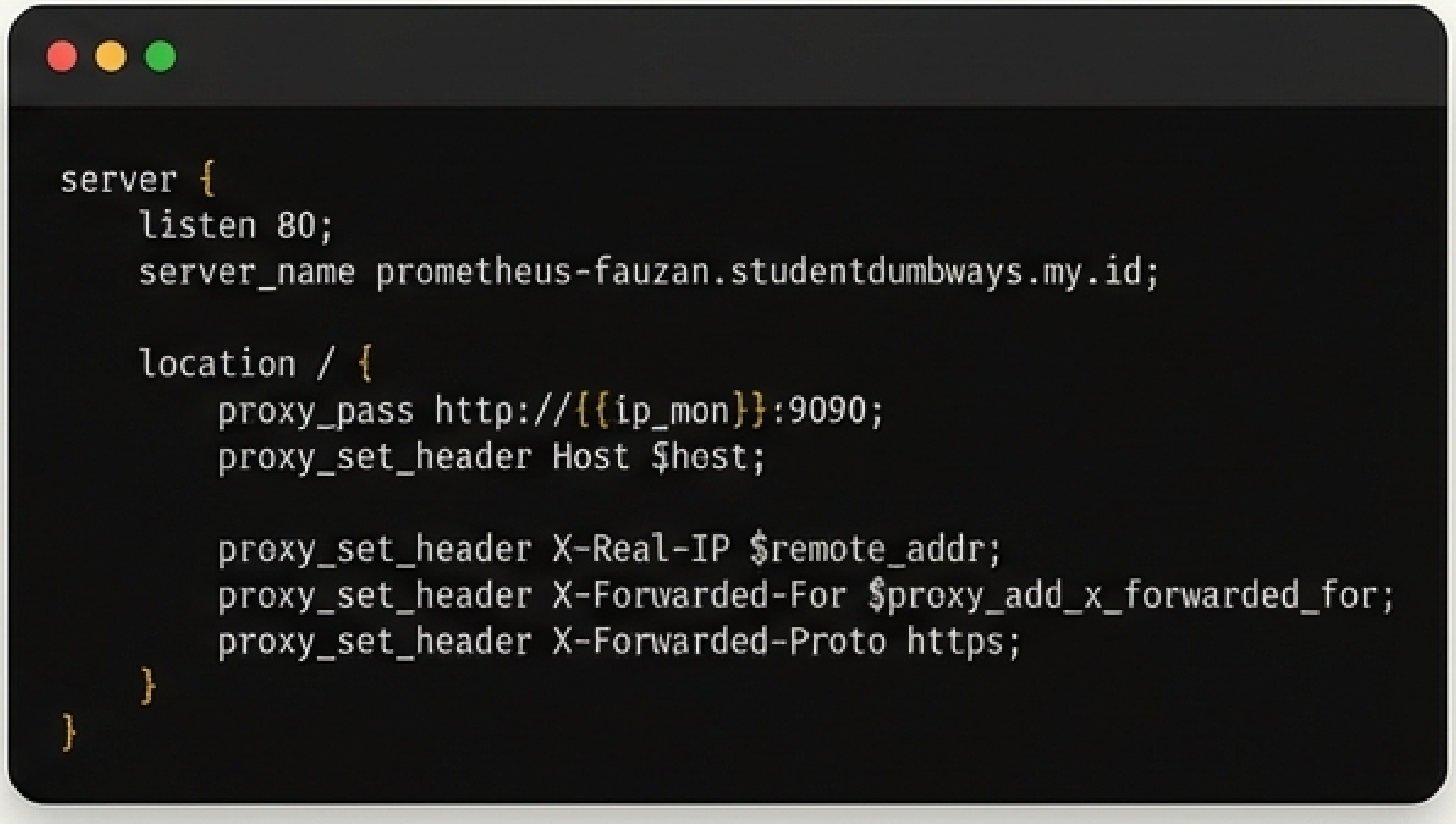
NotebookLM

Monitoring & Observability

Full-stack monitoring menggunakan Prometheus dan Grafana yang di-proxy melalui Nginx.



```
server {
    listen 80;
    server_name grafana-fauzan.studentdumbways.my.id;

    location / {
        proxy_pass http://{{ip_mon}}:3001;
        proxy_set_header Host $host;

        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
    }
}
```



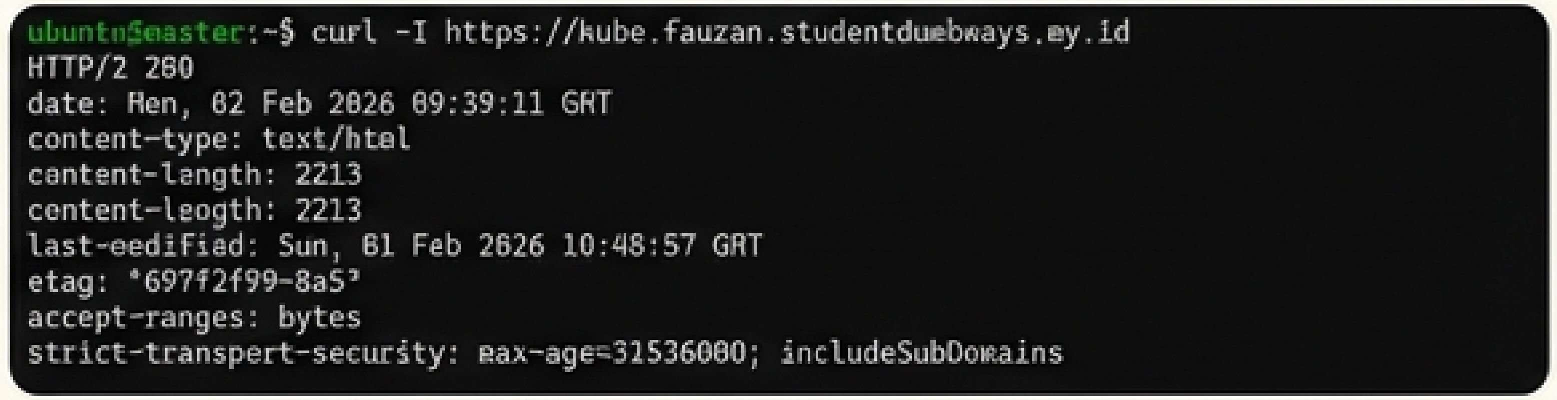
```
server {
    listen 80;
    server_name prometheus-fauzan.studentdumbways.my.id;

    location / {
        proxy_pass http://{{ip_mon}}:9090;
        proxy_set_header Host $host;

        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
    }
}
```

Endpoint: grafana.fauzan... & prometheus.fauzan...

Verifikasi Deployment & Hasil Akhir

HTTPS Verification

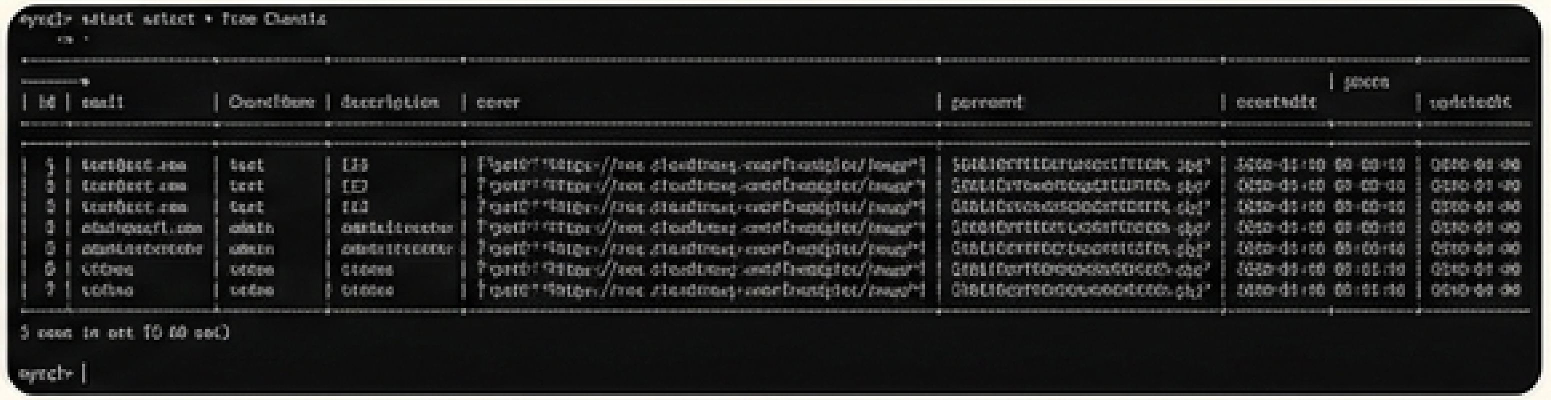
<IMAGE 0>



```
ubuntu@master:~$ curl -I https://kube.fauzan.studentdumbways.my.id
HTTP/2 200
date: Mon, 02 Feb 2026 09:39:11 GMT
content-type: text/html
content-length: 2213
content-leogth: 2213
last-modified: Sun, 01 Feb 2026 10:48:57 GMT
etag: "697f2f99-8a5"
accept-ranges: bytes
strict-transport-security: max-age=31536000; includeSubDomains
```

Data Consistency

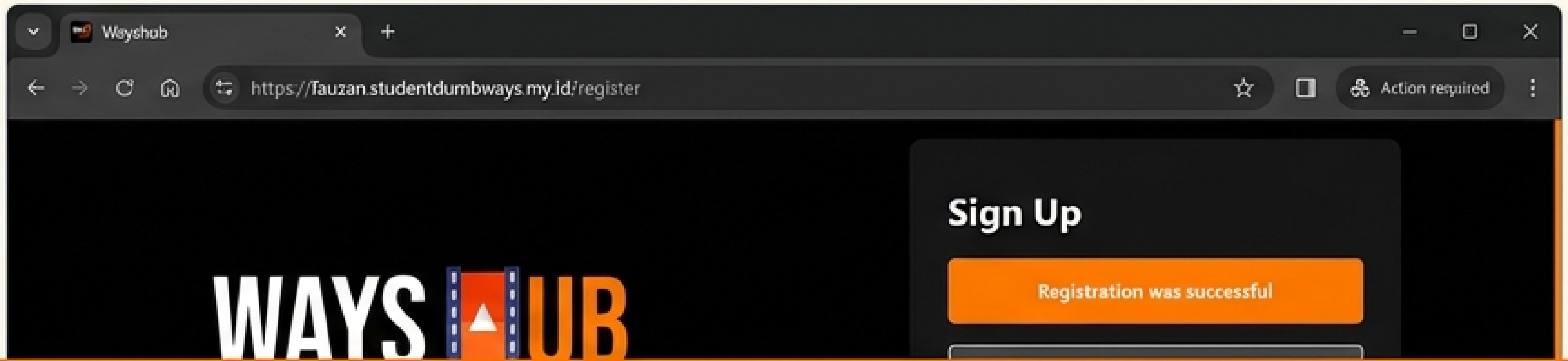
<IMAGE_1>



```
mysql> select select * from Channels
[illegible]
mysql> |
```

Live Application

<IMAGE_2>



Wayshub
https://fauzan.studentdumbways.my.id/register
Action required
Sign Up
Registration was successful
WAYS UB

Status: Live & Operational. Dual-stack Deployment Successful.