

# Tugas 2: Kubernetes

Nama: Muhammad Amir Al Aqwa

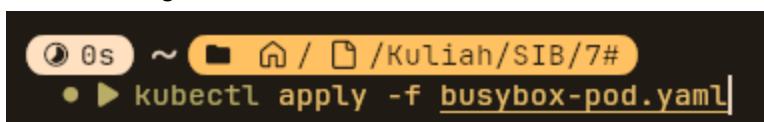
## Praktik Kubernetes

### 1. Menjalankan Image BusyBox dan Print Pesan ke Standard Output

Pertama, saya membuat Pod sederhana dengan image busybox yang akan mencetak pesan ke standard output, dengan membuat file baru bernama busybox-pod.yaml. Dan berikut isinya:

```
13  apiVersion: v1
12  kind: Pod
11  metadata:
10  | name: busybox-pod
  9  | labels:
  8  | | name: busybox
  7  spec:
  6  | restartPolicy: Never
  5  | containers:
  4  | | name: busybox-container
  3  | | image: busybox:latest
  2  | | command:
  1  | | | - echo
14  | | | - Hello, World from BusyBox!
```

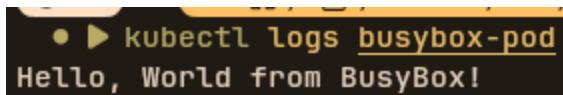
- Jalankan Log:



A terminal window showing the command `kubectl apply -f busybox-pod.yaml` being entered.

pod/busybox-pod created

- Screenshot Log BusyBox-Pod:



A terminal window showing the command `kubectl logs busybox-pod` being run, with the output "Hello, World from BusyBox!" displayed.

### 2. Membuat Pod dengan 2 Container dan Mengakses Nginx

Selanjutnya, saya membuat Pod yang terdiri dari dua kontainer: satu nginx untuk server web dan satu busybox yang akan mengakses kontainer nginx tersebut.

- File YAML (`nginx-sleep-pod.yaml`):

```
16  apiVersion: v1
15  kind: Pod
14  metadata:
13    name: nginx-sleep-pod
12    labels:
11      app: nginx-sleep
10  spec:
9    containers:
8      - name: nginx-container
7        image: nginx:latest
6        ports:
5          - containerPort: 80
4          - name: sleep-container
3        image: busybox:latest
2        command:
1          - sleep
17          - infinity
```

- Perintah:

```
• ➔ kubectl apply -f nginx-sleep-pod.yaml
```

- Pemeriksaan Akses Nginx (melalui log busybox):

```
• ➔ kubectl logs nginx-sleep-pod -c nginx-container
```

- File YAML (nginx-service.yaml):

```
1  apiVersion: v1
1  kind: Service
2  metadata:
3    name: nginx-service
4  spec:
5    type: NodePort
6    ports:
7      - protocol: TCP
8        targetPort: 80
9        port: 80
10   selector:
11     app: nginx-sleep
```

- Terapkan layanan nginx dengan perintah ini:

```
• ➔ kubectl apply -f nginx-service.yaml
```

- Berhasil Akses Nginx:

```
● ▶ kubectl logs nginx-sleep-pod -c nginx-container
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/09/19 15:09:01 [notice] 1#1: using the "epoll" event method
2025/09/19 15:09:01 [notice] 1#1: nginx/1.29.1
2025/09/19 15:09:01 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u1)
2025/09/19 15:09:01 [notice] 1#1: OS: Linux 6.16.7-arch1-1
2025/09/19 15:09:01 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/09/19 15:09:01 [notice] 1#1: start worker processes
2025/09/19 15:09:01 [notice] 1#1: start worker process 29
2025/09/19 15:09:01 [notice] 1#1: start worker process 30
2025/09/19 15:09:01 [notice] 1#1: start worker process 31
2025/09/19 15:09:01 [notice] 1#1: start worker process 32
2025/09/19 15:09:01 [notice] 1#1: start worker process 33
2025/09/19 15:09:01 [notice] 1#1: start worker process 34
2025/09/19 15:09:01 [notice] 1#1: start worker process 35
2025/09/19 15:09:01 [notice] 1#1: start worker process 36
2025/09/19 15:09:01 [notice] 1#1: start worker process 37
2025/09/19 15:09:01 [notice] 1#1: start worker process 38
2025/09/19 15:09:01 [notice] 1#1: start worker process 39
2025/09/19 15:09:01 [notice] 1#1: start worker process 40
```

busybox-container yang menunjukkan respons HTML dari Nginx.)

### 3. Penjelasan Fungsi Perintah kubectl

#### kubectl describe pod <nama-pod>

**Fungsi:** Menampilkan informasi detail tentang sebuah Pod, seperti status, events, dan konfigurasi. Perintah ini sangat berguna untuk debugging karena menyediakan detail tentang apa yang terjadi pada Pod, terutama jika ada masalah.

#### kubectl logs <nama-pod>

**Fungsi:** Digunakan untuk melihat log yang dihasilkan oleh kontainer di dalam Pod. Ini sangat penting untuk memantau perilaku aplikasi dan melacak kesalahan.

#### kubectl exec

**Fungsi:** Memungkinkan kita untuk menjalankan sebuah perintah langsung di dalam kontainer yang sedang berjalan. Perintah ini mirip dengan ssh ke dalam kontainer untuk debugging interaktif.