

Nama : Alpian Roymundus Siringo-ringo

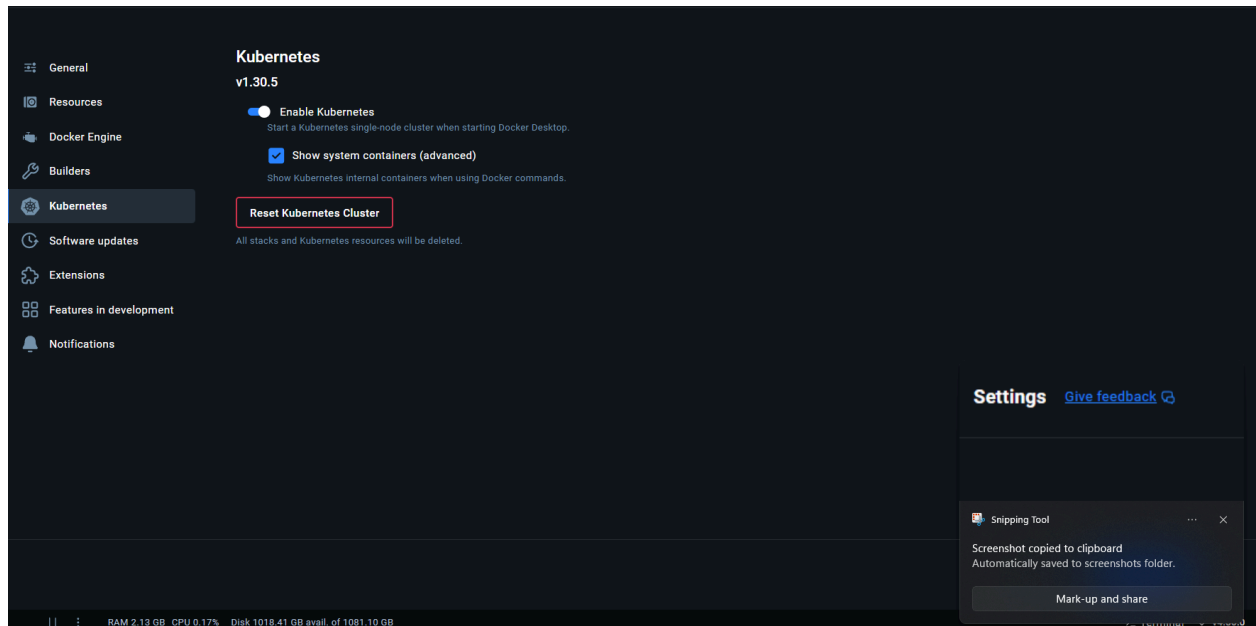
NIM : 11211009

Sistem Terdistribusi A

Tugas Latihan Kubernetes

1. Menginstal Kubernetes di Local

a. Menggunakan Docker Dekstop



2. Menginstal kubectl di Local

Lakukan penginstalan dengan mengikuti dokumentasi berikut: <https://kubernetes.io/docs/tasks/tools/>

```
Command Prompt
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

C:\Users\alpia>curl.exe -LO "https://dl.k8s.io/release/v1.31.0/bin/windows/amd64/kubectl.exe"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 138 100 138 0 0 153 0 --:--:-- --:--:-- --:--:-- 154
100 55.2M 100 55.2M 0 0 883k 0 0:01:03 0:01:03 --:--:-- 1536k

C:\Users\alpia>curl.exe -LO "https://dl.k8s.io/v1.31.0/bin/windows/amd64/kubectl.exe.sha256"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 138 100 138 0 0 275 0 --:--:-- --:--:-- --:--:-- 280
100 64 100 64 0 0 78 0 --:--:-- --:--:-- --:--:-- 0

C:\Users\alpia>CertUtil -hashfile kubectl.exe SHA256
SHA256 hash of kubectl.exe:
a618de26c86421a394de7041f9d0a87752dd4e555894d2278421cf12097fa531
CertUtil: -hashfile command completed successfully.

C:\Users\alpia>type kubectl.exe.sha256
a618de26c86421a394de7041f9d0a87752dd4e555894d2278421cf12097fa531
C:\Users\alpia>kubectl version
Client Version: v1.31.0
Kustomize Version: v5.4.2
Server Version: v1.30.5

C:\Users\alpia>
```

3. Node

Node dalam Kubernetes adalah mesin pekerja (worker machine) yang bertugas menjalankan aplikasi dalam bentuk pod. Sebelumnya, Node juga dikenal dengan istilah "Minion." Node dapat berupa Virtual Machine (VM) atau mesin fisik, tergantung pada infrastruktur yang digunakan. Setiap Node dilengkapi dengan beberapa komponen utama yang memastikan operasionalnya dalam kluster Kubernetes. ****Kubelet**** adalah agen yang berjalan di Node untuk memastikan bahwa kontainer dalam pod dikelola sesuai definisi yang diberikan. ****Kube-proxy**** adalah komponen jaringan yang mengelola aturan jaringan dan mendukung komunikasi antar-pod serta antar-service di kluster. Selain itu, setiap Node memiliki ****container manager****, seperti Docker atau container runtime lainnya, yang bertugas menjalankan dan mengelola kontainer. Kombinasi komponen-komponen ini membuat Node mampu menjalankan aplikasi terdistribusi dalam ekosistem Kubernetes.

a. Melihat Semua Node

```
Terminal
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\alpia> kubectl get node
NAME                STATUS    ROLES    AGE   VERSION
docker-desktop      Ready    control-plane  28m   v1.30.5
PS C:\Users\alpia>
```

b. Melihat Detail Node

```
Terminal
PS C:\Users\alpia> kubectl describe node docker-desktop
Name:          docker-desktop
Roles:         control-plane
Labels:        beta.kubernetes.io/arch=amd64
               beta.kubernetes.io/os=linux
               kubernetes.io/arch=amd64
               kubernetes.io/hostname=docker-desktop
               kubernetes.io/os=linux
               node-role.kubernetes.io/control-plane=
               node.kubernetes.io/exclude-from-external-load-balancers=
Annotations:   kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/cri-dockerd.sock
               node.alpha.kubernetes.io/ttl: 0
               volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp: Wed, 27 Nov 2024 12:02:04 +0800
Taints:        <none>
Unschedulable: false
Lease:
  HolderIdentity:  docker-desktop
  AcquireTime:     <unset>
  RenewTime:       Wed, 27 Nov 2024 12:37:00 +0800
Conditions:
  Type            Status  LastHeartbeatTime             LastTransitionTime             Reason                           Message
  ----            -
  MemoryPressure  False   Wed, 27 Nov 2024 12:36:46 +0800 Wed, 27 Nov 2024 12:02:02 +0800 KubeletHasSufficientMemory      kubelet has sufficient memory available
  DiskPressure    False   Wed, 27 Nov 2024 12:36:46 +0800 Wed, 27 Nov 2024 12:02:02 +0800 KubeletHasNoDiskPressure        kubelet has no disk pressure
  PIDPressure     False   Wed, 27 Nov 2024 12:36:46 +0800 Wed, 27 Nov 2024 12:02:02 +0800 KubeletHasSufficientPID         kubelet has sufficient PID available

Kubernetes running RAM 4.79 GB CPU 1.52% Disk 1016.50 GB avail. of 1081.10 GB
```

4. Pod

Pod adalah unit terkecil yang dapat dideploy di kluster Kubernetes, berfungsi sebagai wadah bagi satu atau lebih container yang saling berbagi jaringan, penyimpanan, dan konteks eksekusi. Secara sederhana, Pod merepresentasikan aplikasi yang berjalan di kluster Kubernetes. Pod memastikan container di dalamnya dapat berkomunikasi satu sama lain dengan mudah dan dikelola sebagai satu kesatuan, sehingga mempermudah orkestrasi dan deployment aplikasi.

a. Melihat Semua Pod

```
Terminal

PS C:\Users\alpia> kubectl get pod
No resources found in default namespace.
PS C:\Users\alpia>
```

b. Melihat Detail Pod

```
Terminal

PS C:\Users\alpia> kubectl describe pod docker-desktop
Error from server (NotFound): pods "docker-desktop" not found
PS C:\Users\alpia>
```

5. Membuat Pod

```
nginx.yaml x

nginx.yaml
You, 58 minutes ago | 1 author (You)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx
5  spec:
6    containers:
7      - name: nginx
8        image: nginx
9        ports:
10       - containerPort: 80
You, 58 minutes ago • add ng
```

a. Membuat Pod

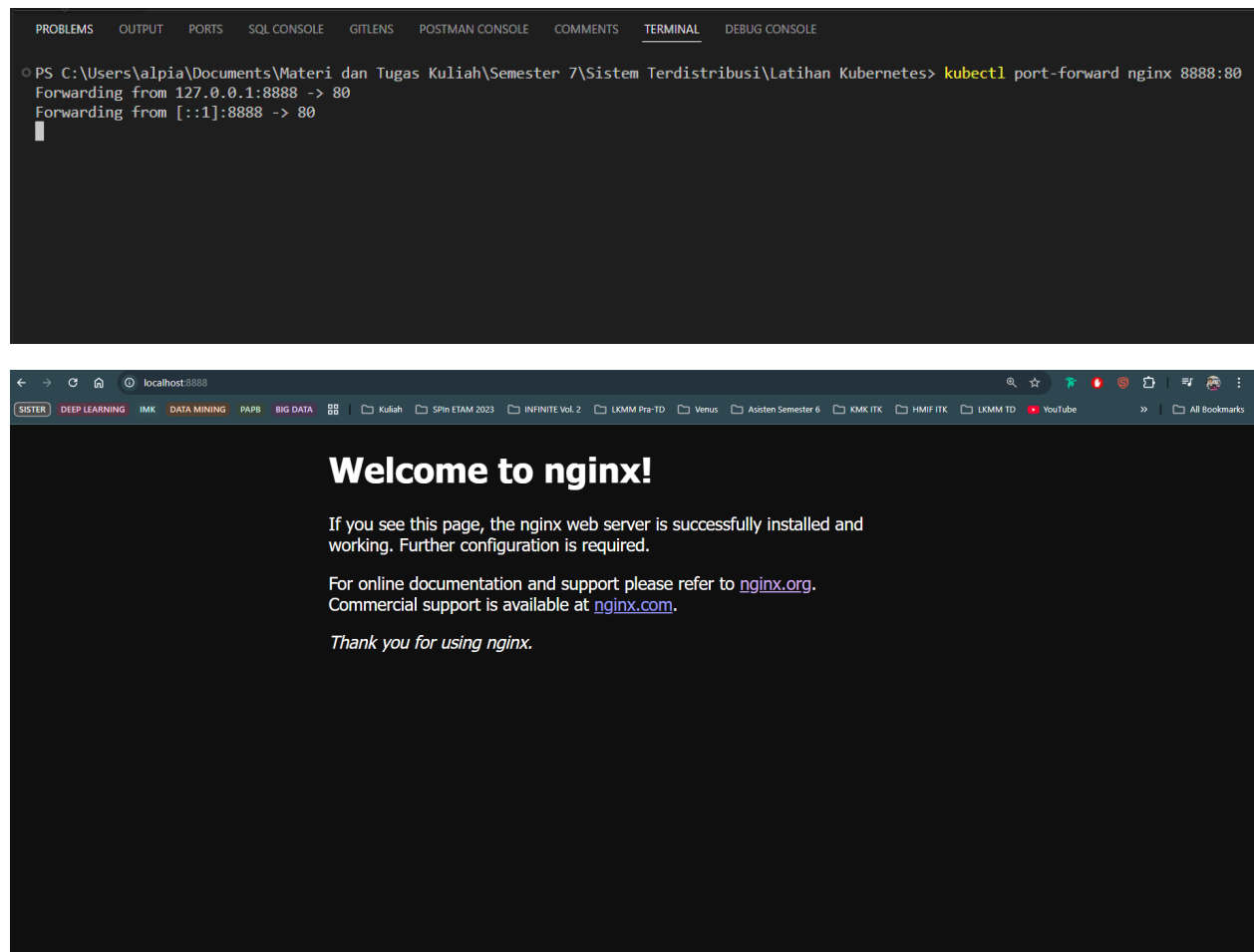
```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
● PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx.yaml
pod/nginx created
○ PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> 
```

b. Melihat Pod

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
● PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME READY STATUS RESTARTS AGE
nginx 1/1 Running 0 104s
● PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
nginx 1/1 Running 0 116s 10.1.0.7 docker-desktop <none> <none>
○ PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> 
```

```
● PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe pod nginx
Name: nginx
Namespace: default
Priority: 0
Service Account: default
Node: docker-desktop/192.168.65.3
Start Time: Wed, 27 Nov 2024 14:26:00 +0800
Labels: <none>
Annotations: <none>
Status: Running
IP: 10.1.0.7
IPs:
  IP: 10.1.0.7
Containers:
  nginx:
    Container ID: docker://f3aba17c3864be4fad280578158cca0386d2b6048f8c7ed250c2fd44b8832714
    Image: nginx
    Image ID: docker-pullable://nginx@sha256:0c86dddac19f2ce4fd716ac58c0fd87bf69bfd4edabfd6971fb885bafd12a00b
    Port: 80/TCP
    Host Port: 0/TCP
    State: Running
```

c. Mengakses Pod



6. Label

Label adalah mekanisme di Kubernetes untuk memberikan tanda atau atribut pada resource, termasuk Pod, guna mengorganisir dan memberikan informasi tambahan. Label memudahkan pengelompokan resource berdasarkan kriteria tertentu, seperti versi aplikasi atau lingkungan (contoh: `env: production`). Selain Pod, label juga dapat diterapkan pada resource lain seperti Replication Controller, Replica Set, dan Service, menjadikannya alat yang fleksibel untuk mengelola dan mengontrol operasi dalam kluster

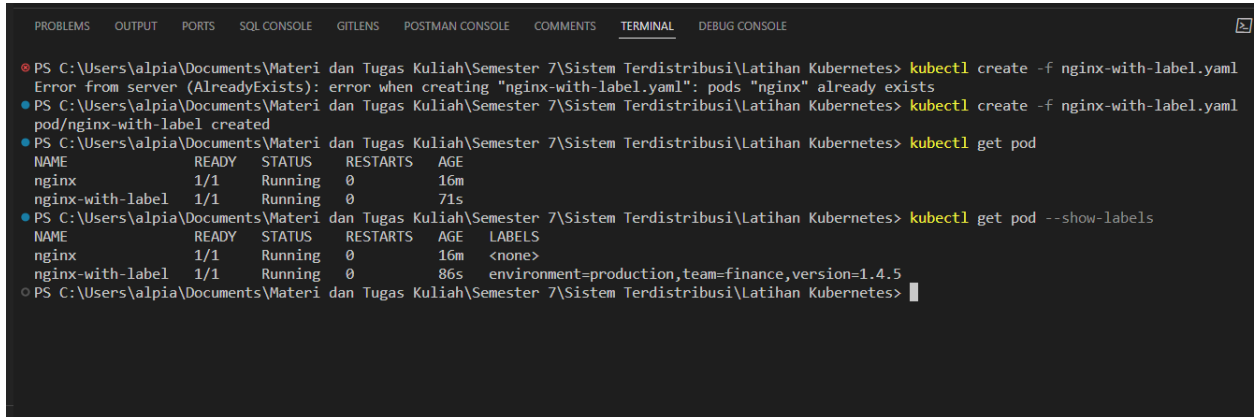
Kubernetes.



```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: nginx-with-label
5   labels:
6     team: finance
7     version: 1.4.5
8     environment: production
9 spec:
10  containers:
11  - name: nginx
12    image: nginx
13    ports:
14  - containerPort: 80
```

a. Menambahkan Label di Pod

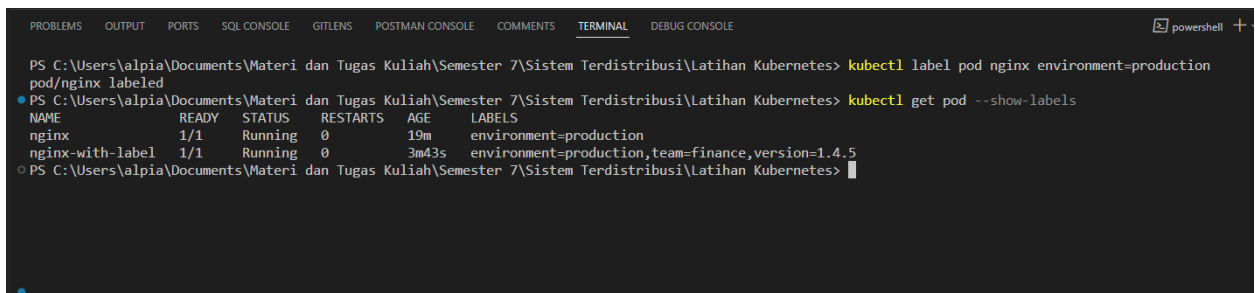
Catatan: Nama Pod harus berbeda atau unik.



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx-with-label.yaml
Error from server (AlreadyExists): error when creating "nginx-with-label.yaml": pods "nginx" already exists
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx-with-label.yaml
pod/nginx-with-label created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx         1/1     Running   0           16m
nginx-with-label 1/1     Running   0           71s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod --show-labels
NAME          READY   STATUS    RESTARTS   AGE   LABELS
nginx         1/1     Running   0           16m   <none>
nginx-with-label 1/1     Running   0           86s   environment=production,team=finance,version=1.4.5
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Menambahkan atau Mengubah Label di Pod

Menambahkan Label di Pod



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl label pod nginx environment=production
pod/nginx labeled
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod --show-labels
NAME          READY   STATUS    RESTARTS   AGE   LABELS
nginx         1/1     Running   0           19m   environment=production
nginx-with-label 1/1     Running   0          3m43s   environment=production,team=finance,version=1.4.5
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

Mengubah Label di Pod

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl label pod nginx environment=production pod/nginx labeled
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod --show-labels
NAME          READY STATUS RESTARTS AGE LABELS
nginx         1/1   Running 0       19m   environment=production
nginx-with-label 1/1   Running 0       3m43s environment=production,team=finance,version=1.4.5
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl label pod nginx environment=qa
error: 'environment' already has a value (production), and --overwrite is false
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl label pod nginx environment=qa --overwrite pod/nginx labeled
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod --show-labels
NAME          READY STATUS RESTARTS AGE LABELS
nginx         1/1   Running 0       19m   environment=qa
nginx-with-label 1/1   Running 0       4m34s environment=production,team=finance,version=1.4.5
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

c. Mencari Pod dengan Label

i. Mencari Pod dengan Menuliskan Key

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pods -l environment
NAME          READY STATUS RESTARTS AGE
nginx         1/1   Running 0       22m
nginx-with-label 1/1   Running 0       6m56s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

ii. Mencari Pod dengan menuliskan key dan value

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pods -l team=rnd
No resources found in default namespace.
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pods -l team=finance
NAME          READY STATUS RESTARTS AGE
nginx-with-label 1/1   Running 0       9m2s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```


iii. Mencari Pod dengan label yang tidak mempunyai Key pada Podnya

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GIT LENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pods -l '!team'
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   0           26m
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

d. Mencari Pod dengan Beberapa Label

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pods -l environment,team=finance
NAME            READY   STATUS    RESTARTS   AGE
nginx-with-label 1/1     Running   0           15m
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

7. Annotation

Annotation di Kubernetes mirip dengan label, tetapi tidak dapat digunakan untuk melakukan filtering atau seleksi resource. Annotation biasanya digunakan untuk menambahkan informasi tambahan dalam ukuran besar, seperti metadata atau konfigurasi non-identifikasi. Dengan kapasitas hingga 256 KB, annotation cocok untuk menyimpan detail seperti checksum, URL dokumentasi, atau catatan debugging. Annotation dapat diterapkan pada berbagai resource Kubernetes, memberikan fleksibilitas dalam menyimpan data

deskriptif tanpa memengaruhi operasi kluster.

```
nginx-with-annotation.yaml X
nginx-with-annotation.yaml
You, 59 minutes ago | 1 author (You)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx-with-annotation
5    labels:
6      team: product
7      version: 1.0.1
8      environment: development
9    annotations:
10     description: Ini adalah aplikasi yang dibuat oleh tim product
11     apapun: Apapun itu....
12 spec:
13   containers: You, 59 minutes ago • add nginx
14     - name: nginx
15       image: nginx
16       ports:
17         - containerPort: 80
```

a. Menambahkan Annotation ke Pod

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl annotate pod nginx-with-label description="Ini adalah aplikasi tim finance"
pod/nginx-with-label annotated
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl annotate pod nginx-with-label description="Ini adalah aplikasi team finance" --overwrite
pod/nginx-with-label annotated
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe pod nginx-with-label
Name:          nginx-with-label
Namespace:     default
Priority:       0
Service Account: default
Node:          docker-desktop/192.168.65.3
Start Time:    Wed, 27 Nov 2024 14:41:21 +0800
Labels:        environment=production
               team=finance
               version=1.4.5
Annotations:   description: Ini adalah aplikasi team finance
Status:        Running
IP:            10.1.0.8
IPs:           10.1.0.8
2/18 needs reviewers 0 0 0 0 Connect Live Share Ln 13, Col 14 Spaces: 2 UTF-8 CRLF YAML Go Live
```

8. Namespace

```
finance-namespace.yaml X
finance-namespace.yaml
You, 59 minutes ago | 1 author (You)
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: finance You, 59 minutes ago • add finance namespace
```

Namespace di Kubernetes digunakan untuk mengorganisir resource ketika jumlahnya sudah terlalu

banyak atau ketika perlu memisahkan resource berdasarkan multi-tenant, tim, atau lingkungan (seperti development, staging, dan production). Dengan namespace, resource yang memiliki nama sama dapat dikelola tanpa konflik, karena berada di ruang lingkup yang berbeda. Namespace membantu dalam pengelolaan skala besar, memberikan isolasi logis, dan mempermudah pengaturan akses serta kontrol terhadap resource dalam kluster Kubernetes.

a. Melihat Namespace

```
Terminal
PS C:\Users\alpia> kubectl get ns
NAME                STATUS   AGE
default              Active   5h
kube-node-lease      Active   5h
kube-public          Active   5h
kube-system          Active   5h
PS C:\Users\alpia>
```

b. Melihat Pod di Namespace

```
Terminal
PS C:\Users\alpia> kubectl get pods --namespace default
NAME                READY   STATUS    RESTARTS   AGE
nginx               1/1    Running   0           158m
nginx-with-annotation 1/1    Running   0           10m
nginx-with-label    1/1    Running   0           142m
PS C:\Users\alpia> kubectl get pods --namespace kube-system
NAME                READY   STATUS    RESTARTS   AGE
coredns-55cb58b774-f87pr 1/1    Running   0           5h2m
coredns-55cb58b774-m55z8 1/1    Running   0           5h2m
etcd-docker-desktop    1/1    Running   0           5h2m
kube-apiserver-docker-desktop 1/1    Running   0           5h2m
kube-controller-manager-docker-desktop 1/1    Running   0           5h2m
kube-proxy-278wr       1/1    Running   0           5h2m
kube-scheduler-docker-desktop 1/1    Running   0           5h2m
storage-provisioner     1/1    Running   0           5h2m
vpknit-controller      1/1    Running   0           5h2m
PS C:\Users\alpia>
```

9. Membuat Namespace

Namespace di Kubernetes memungkinkan Pod dengan nama yang sama untuk berjalan di namespace yang berbeda tanpa konflik. Namun, namespace tidak dirancang untuk sepenuhnya mengisolasi resource. Pod di satu namespace tetap dapat berkomunikasi dengan Pod di namespace lain, kecuali ada pembatasan tambahan, seperti Network Policy. Namespace lebih berfungsi sebagai pengelompokan logis resource daripada sebagai mekanisme isolasi penuh dalam kluster Kubernetes.

a. Membuat Namespace

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f finance-namespace.yaml
namespace/finance created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Membuat Pod di Namespace

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + - []
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx.yaml
Error from server (AlreadyExists): error when creating "nginx.yaml": pods "nginx" already exists
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx.yaml -n namespace finance
pod/nginx created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

c. Menghapus Namespace

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + - [] ...
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete namespace finance
namespace "finance" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

10. Menghapus Pod

a. Menghapus Pod

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + - [ ]

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx         1/1    Running   0          172m
nginx-with-annotation  1/1    Running   0          24m
nginx-with-label    1/1    Running   0          157m
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete pod nginx
pod "nginx" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx-with-annotation  1/1    Running   0          25m
nginx-with-label    1/1    Running   0          157m
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Menghapus Pod Menggunakan Label

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + - [ ]

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod --show-labels
NAME          READY   STATUS    RESTARTS   AGE   LABELS
nginx-with-annotation  1/1    Running   0          28m   environment=development,team=product,version=1.0.1
nginx-with-label    1/1    Running   0          160m   environment=production,team=finance,version=1.4.5
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete pod -l environment=development
pod "nginx-with-annotation" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod --show-labels
NAME          READY   STATUS    RESTARTS   AGE   LABELS
nginx-with-label    1/1    Running   0          161m   environment=production,team=finance,version=1.4.5
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

c. Menghapus Semua Pod di Namespace

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + - [ ]

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete pod --all --namespace finance
No resources found
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete pod --all --namespace default
pod "nginx-with-label" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod --namespace default
No resources found in default namespace.
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

11. Probe

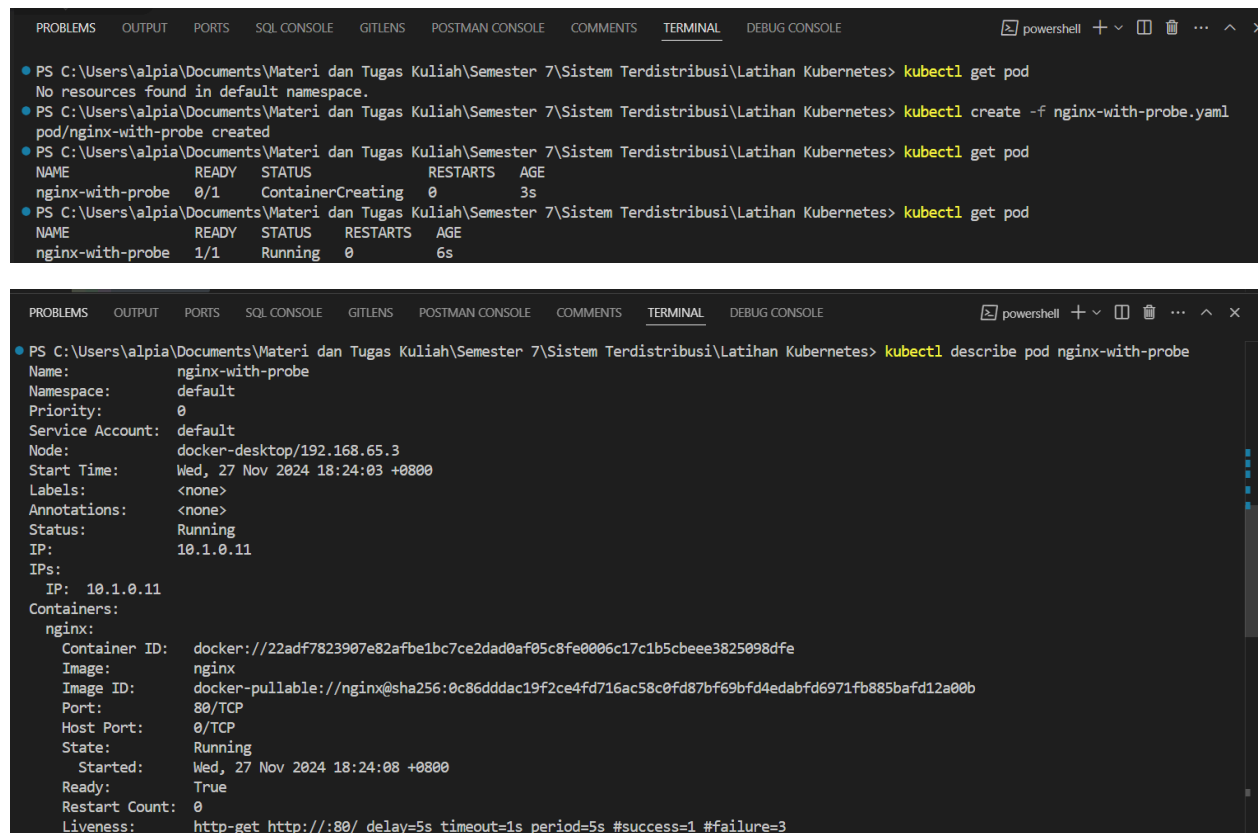
Kubelet menggunakan tiga jenis probe untuk memonitor kesehatan dan kesiapan Pod: *liveness probe* memastikan Pod dalam keadaan hidup, dan jika tidak merespon, kubelet akan me-restart Pod. *Readiness probe* digunakan untuk memeriksa apakah Pod siap menerima traffic, sehingga membantu pengaturan

load balancer atau service routing. ****Startup probe**** digunakan untuk memastikan Pod yang membutuhkan waktu lama untuk startup berjalan sepenuhnya sebelum dilakukan pengecekan liveness dan readiness, mencegah kubelet mematikan Pod sebelum selesai berfungsi dengan sempurna. Probes ini memastikan stabilitas dan efisiensi operasional Pod dalam Kubernetes.

```
nginx-rc.yaml
1  apiVersion: v1
2  kind: ReplicationController
3  metadata:
4    name: nginx-rc
5  spec:
6    replicas: 3
7    selector:
8      app: nginx
9    template:
10     metadata:
11       labels:
12         app: nginx
13     spec:
14       containers:
15         - name: nginx
16           image: nginx
17           ports:
18             - containerPort: 80
```

```
nginx-with-probe.yaml U X
nginx-with-probe.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx-with-probe
5  spec:
6    containers:
7      - name: nginx
8        image: nginx
9        ports:
10         - containerPort: 80
11        livenessProbe:
12          httpGet:
13            path: /
14            port: 80
15          initialDelaySeconds: 5
16          periodSeconds: 5
17          timeoutSeconds: 1
18          successThreshold: 1
19          failureThreshold: 3
```

a. Melihat Detail Probe



The first screenshot shows the execution of `kubectl get pod` in a PowerShell terminal. The output indicates that no resources were found in the default namespace. Subsequently, the command `kubectl create -f nginx-with-probe.yaml` is executed, resulting in the pod `nginx-with-probe` being created. A second `kubectl get pod` command shows the pod's status as `ContainerCreating` with 0 restarts and an age of 3 seconds.

The second screenshot shows the execution of `kubectl describe pod nginx-with-probe`. The output provides detailed information about the pod, including its name, namespace, priority, service account, node, start time, labels, annotations, status, IP, and containers. The `nginx` container is shown as `Running` with its own IP and various configuration details.

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
No resources found in default namespace.
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx-with-probe.yaml
pod/nginx-with-probe created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx-with-probe 0/1     ContainerCreating  0          3s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx-with-probe 1/1     Running   0          6s

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe pod nginx-with-probe
Name:          nginx-with-probe
Namespace:     default
Priority:       0
Service Account: default
Node:          docker-desktop/192.168.65.3
Start Time:    Wed, 27 Nov 2024 18:24:03 +0800
Labels:        <none>
Annotations:   <none>
Status:        Running
IP:            10.1.0.11
IPs:           IP: 10.1.0.11
Containers:
  nginx:
    Container ID:  docker://22adf7823907e82afbe1bc7ce2dad0af05c8fe0006c17c1b5cbeee3825098dfe
    Image:         nginx
    Image ID:      docker-pullable://nginx@sha256:0c86dddac19f2ce4fd716ac58c0fd87bf69bfd4edabfd6971fb885bafd12a00b
    Port:          80/TCP
    Host Port:     0/TCP
    State:         Running
      Started:     Wed, 27 Nov 2024 18:24:08 +0800
    Ready:         True
    Restart Count: 0
    Liveness:      http-get http://:80/ delay=5s timeout=1s period=5s #success=1 #failure=3
```

12. Membuat Replication Controller

Replication Controller di Kubernetes bertugas untuk memastikan bahwa jumlah Pod yang berjalan selalu sesuai dengan yang telah ditentukan. Jika sebuah Pod mati atau hilang, misalnya karena Node yang gagal, Replication Controller akan secara otomatis membuat Pod baru untuk menggantikan Pod yang hilang tersebut. Replication Controller biasanya digunakan untuk mengelola lebih dari satu Pod, menjaga konsistensi jumlah Pod yang diinginkan. Jika jumlah Pod yang berjalan kurang dari yang diinginkan, Replication Controller akan menambah Pod baru, dan jika lebih, ia akan menghapus Pod yang berlebih untuk menjaga stabilitas aplikasi.

a. Melihat Replication Controller

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx-rc.yaml
replicationcontroller/nginx-rc created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get rc
NAME          DESIRED  CURRENT  READY  AGE
nginx-rc      3        3        2      10s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx-rc-4f4fd 1/1     Running   0          67s
nginx-rc-7v4jc 1/1     Running   0          67s
nginx-rc-q2mnq 1/1     Running   0          67s
nginx-with-probe 1/1     Running   0          175m
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

13. Menghapus Replicatoin Controller

Ketika kita menghapus Replication Controller di Kubernetes, secara default, Pod yang dipilih oleh label selector Replication Controller tersebut akan ikut terhapus. Namun, jika kita ingin menghapus Replication Controller tanpa menghapus Pod yang terkait, kita bisa menggunakan opsi `--cascade=false`. Opsi ini mencegah penghapusan Pod yang dikelola oleh Replication Controller, memungkinkan Pod untuk tetap berjalan meskipun Replication Controller sudah dihapus.

a. Menghapus Replication Controller

i. Menghapus semua pod

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get rc
NAME          DESIRED  CURRENT  READY  AGE
nginx-rc      3        3        3      4m26s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete rc nginx-rc
replicationcontroller "nginx-rc" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get rc
No resources found in default namespace.
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx-with-probe 1/1     Running   0          3h
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

ii. Menghapus Replication Controller tanpa menghapus pod

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete rc nginx-rc --cascade=false
warning: --cascade=false is deprecated (boolean value) and can be replaced with --cascade=orphan.
replicationcontroller "nginx-rc" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get rc
No resources found in default namespace.
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx-rc-j89qg 1/1     Running   0          118s
nginx-rc-khk47 1/1     Running   0          118s
nginx-rc-nhrtr 1/1     Running   0          118s
nginx-with-probe 1/1     Running   0          3h3m
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

14. Replica Set

Pada awalnya, Replication Controller digunakan untuk menjaga jumlah replica Pod dan me-reschedule Pod yang mati. Namun, kini telah diperkenalkan resource baru bernama Replica Set, yang merupakan

generasi baru dari Replication Controller dan berfungsi sebagai penggantinya. Replica Set menawarkan fitur yang lebih fleksibel dan lebih baik dalam pengelolaan Pod replica. Penggunaan Replication Controller kini sudah tidak direkomendasikan lagi, karena Replica Set lebih direkomendasikan untuk memastikan ketersediaan Pod dalam kluster Kubernetes.

```
nginx-rs.yaml
4   name: nginx
5   spec:
6     replicas: 3
7     selector:
8       matchLabels:
9         app: nginx
10    template:
11      metadata:
12        name: nginx
13        labels:
14          app: nginx
15      spec:
16        containers:
17          - name: nginx
18            image: nginx
19            ports:
20              - containerPort: 80
```

a. Membuat Replica Set

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GIT LENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx-rs.yaml
replicaset.apps/nginx created
• PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
nginx-dmrhn   1/1     Running   0           7m46s
nginx-wd2b9   1/1     Running   0           7m46s
nginx-x4fh2   1/1     Running   0           7m46s
• PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get rs
NAME   DESIRED   CURRENT   READY   AGE
nginx   3         3         3       8m7s
• PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> |
```

15. Label Selector Match Expression

Pada Replica Set, selector digunakan untuk menentukan Pod mana yang akan dikelola. Sebelumnya, selector pada Replication Controller hanya menggunakan `matchLabels`, yang berarti Pod yang dipilih harus memiliki label yang sama persis. Namun, di Replica Set, selain `matchLabels`, kita juga bisa menggunakan `matchExpressions`. `matchExpressions` memungkinkan penggunaan kondisi yang lebih kompleks, seperti menggunakan operator logika (e.g., `In`, `NotIn`, `Exists`, `DoesNotExist`) untuk memilih Pod berdasarkan nilai label tertentu. Fitur ini memberikan fleksibilitas lebih dalam memilih Pod

yang sesuai dengan kriteria yang lebih spesifik.

```
nginx-rs-match-expression.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: nginx
5  spec:
6    replicas: 3
7    selector:
8      matchExpressions:
9        - key: app
10         operator: In
11         values:
12           - nginx
13        - key: env
14         operator: In
15         values:
16           - prod
17           - qa
18           - dev
19    template:
20      metadata:
21        name: nginx
22        labels:
23          app: nginx
24          env: prod
25      spec:
26        containers:
27          - name: nginx
28            image: nginx
29            ports:
30              - containerPort: 80
```

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f nginx-rs-match-expression.yaml
replicaset.apps/nginx created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get rs
NAME      DESIRED  CURRENT  READY  AGE
nginx     3         3         3      14s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

16. Daemon Set

DemonSet di Kubernetes digunakan untuk memastikan bahwa sebuah Pod berjalan di setiap Node dalam kluster, dengan satu Pod per Node. Berbeda dengan Replica Set atau Replication Controller yang dapat menjalankan Pod secara acak di Node, DaemonSet memastikan distribusi Pod yang konsisten di semua Node. Secara default, DaemonSet akan menjalankan Pod di semua Node yang ada dalam kluster, namun kita juga bisa membatasi agar Pod hanya dijalankan di Node tertentu dengan menggunakan label atau seleksi. Hal ini sangat berguna untuk aplikasi yang perlu dijalankan di setiap Node, seperti agent

monitoring atau log collector.

```
daemon-nginx.yaml
1  apiVersion: apps/v1
2  kind: DaemonSet
3  metadata:
4    name: daemon-nginx
5    labels:
6      name: daemon-nginx
7  spec:
8    selector:
9      matchLabels:
10       name: daemon-nginx
11    template:
12      metadata:
13        name: daemon-nginx
14        labels:
15          name: daemon-nginx
16      spec:
17        containers:
18        - name: nginx
19          image: nginx
20          ports:
21            - containerPort: 80
22          readinessProbe:
23            httpGet:
24              path: /
25              port: 80
26            initialDelaySeconds: 0
27            periodSeconds: 10
28            failureThreshold: 3
29            successThreshold: 1
30            timeoutSeconds: 1
```

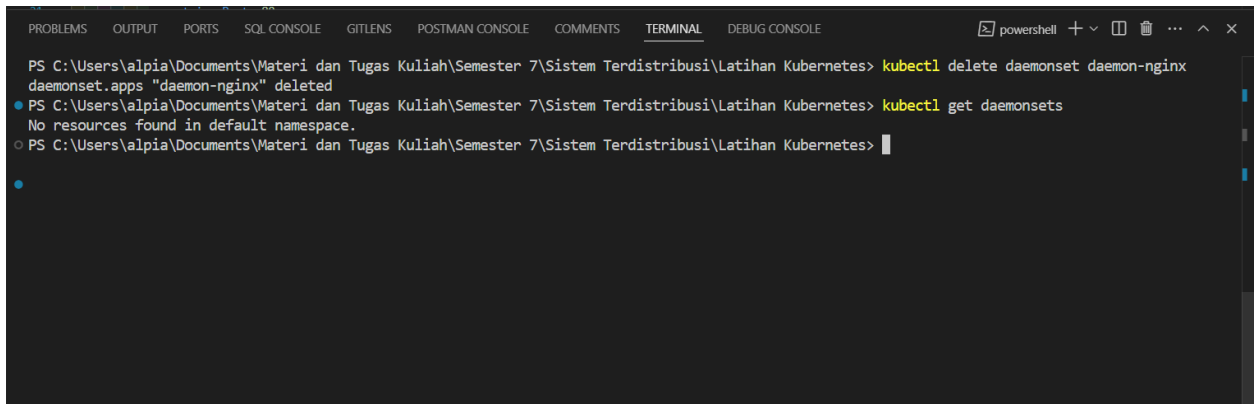
a. Membuat Daemon Set

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f daemon-nginx.yaml
daemonset.apps/daemon-nginx created
```

b. Melihat Semua Daemon Set

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get daemonsets
NAME          DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
daemon-nginx  1        1        1      1            1          <none>         26s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

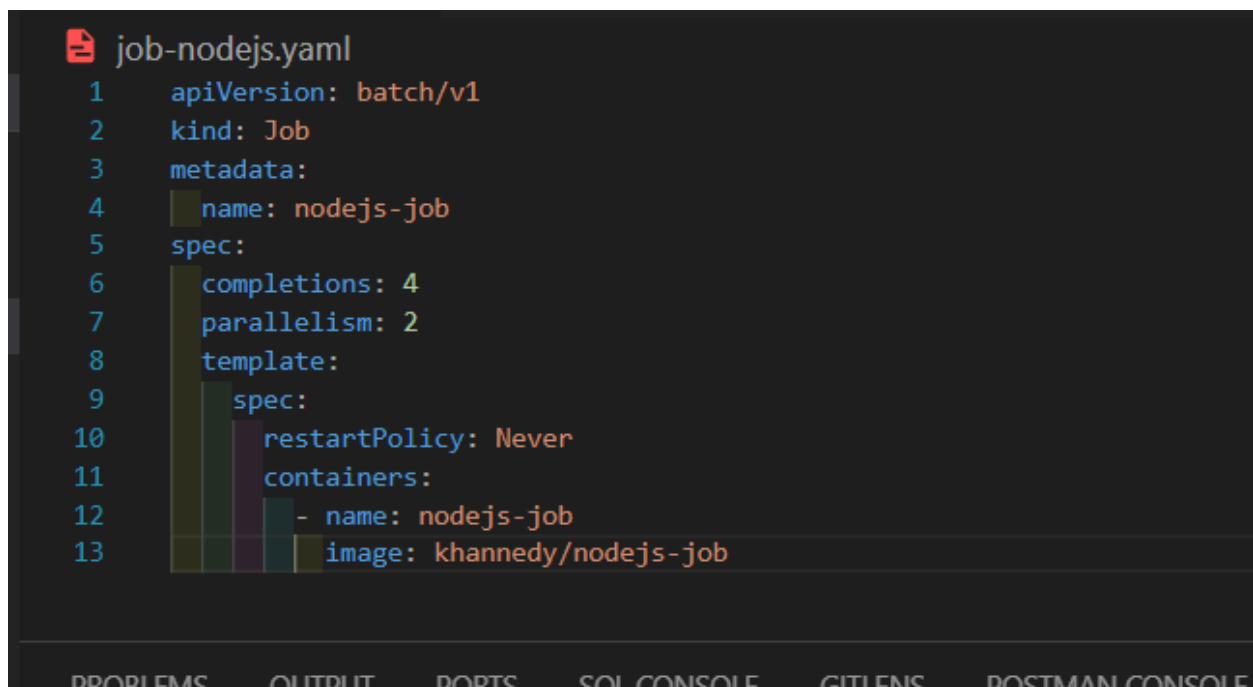
c. Menghapus Daemon Set



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete daemonset daemon-nginx
daemonset.apps "daemon-nginx" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get daemonsets
No resources found in default namespace.
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

17. Job

Job di Kubernetes digunakan untuk menjalankan Pod yang hanya perlu berfungsi sekali, menjalankan tugas tertentu, dan kemudian berhenti setelah selesai. Berbeda dengan Replication Controller, Replica Set, dan Daemon Set yang secara otomatis menjalankan ulang Pod jika mati, Job memastikan bahwa Pod hanya berjalan untuk menyelesaikan pekerjaannya. Setelah tugas selesai, Pod yang terkait dengan Job akan mati, dan tidak ada upaya untuk menjalankannya ulang. Job cocok untuk menjalankan proses batch atau tugas yang bersifat sementara dan tidak membutuhkan keberlanjutan.



```
job-nodejs.yaml
1  apiVersion: batch/v1
2  kind: Job
3  metadata:
4    name: nodejs-job
5  spec:
6    completions: 4
7    parallelism: 2
8    template:
9      spec:
10     restartPolicy: Never
11     containers:
12     - name: nodejs-job
13       image: khannedy/nodejs-job
```

a. Membuat Job

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + - [ ] [ ] ... ^
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f job-nodejs.yaml
job.batch/nodejs-job created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> 
```

b. Melihat Semua Job

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get jobs
NAME          STATUS    COMPLECTIONS  DURATION  AGE
nodejs-job    Running  2/4           13s       13s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> 
```

c. Menghapus Job

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete job nodejs-job
job.batch "nodejs-job" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> 
```

18. Cron Job

CronJob di Kubernetes adalah resource yang memungkinkan penjadwalan tugas untuk dijalankan secara berulang, mirip dengan aplikasi Cron di sistem operasi Unix. CronJob memungkinkan kita untuk menentukan jadwal eksekusi tugas, sehingga aplikasi atau Pod dapat berjalan secara otomatis pada waktu yang telah ditentukan. Berbeda dengan Job yang hanya berjalan sekali, CronJob akan menjalankan tugas secara berkala sesuai dengan jadwal yang ditetapkan, seperti setiap hari, mingguan, atau sesuai interval

lainnya, menjadikannya ideal untuk tugas yang perlu dijalankan secara teratur.

```
crontab-nodejs.yaml
1  apiVersion: batch/v1beta1
2  kind: CronJob
3  metadata:
4    name: nodejs-cronjob
5    labels:
6      name: nodejs-cronjob
7  spec:
8    schedule: "* * * * *"
9    jobTemplate:
10     spec:
11       template:
12         metadata:
13           name: nodejs-cronjob
14           labels:
15             name: nodejs-cronjob
16         spec:
17           restartPolicy: Never
18           containers:
19             - name: nodejs-cronjob
20               image: khannedy/nodejs-job
```

a. Membuat Cron Job

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f crontab-nodejs.yaml
crontab.batch/nodejs-cronjob created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get cronjobs
NAME          SCHEDULE    TIMEZONE    SUSPEND    ACTIVE    LAST SCHEDULE    AGE
nodejs-cronjob * * * * *   <none>      False     0         <none>      9s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get cronjobs
NAME          SCHEDULE    TIMEZONE    SUSPEND    ACTIVE    LAST SCHEDULE    AGE
nodejs-cronjob * * * * *   <none>      False     1         7s         35s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get cronjobs
NAME          SCHEDULE    TIMEZONE    SUSPEND    ACTIVE    LAST SCHEDULE    AGE
nodejs-cronjob * * * * *   <none>      False     0         38s        66s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get cronjobs
NAME          SCHEDULE    TIMEZONE    SUSPEND    ACTIVE    LAST SCHEDULE    AGE
nodejs-cronjob * * * * *   <none>      False     0         45s        73s
```

b. Mendapatkan Semua Cron Job

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get all
NAME                                READY    STATUS    RESTARTS    AGE
pod/nginx-9pp2g                     1/1     Running   1 (61m ago)  12h
pod/nginx-g9kv4                     1/1     Running   1 (61m ago)  12h
pod/nginx-qkr5s                     1/1     Running   1 (61m ago)  12h
pod/nodejs-cronjob-28879347-vc82d    0/1     Completed 0            62s
pod/nodejs-cronjob-28879348-f6l2r    0/1     ContainerCreating 0            2s

NAME          TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1     <none>         443/TCP    22h

NAME          DESIRED    CURRENT    READY    AGE
replicaset.apps/nginx  3          3          3        12h

NAME          SCHEDULE    TIMEZONE    SUSPEND    ACTIVE    LAST SCHEDULE    AGE
cronjob.batch/nodejs-cronjob  * * * * *   <none>      False     1         2s            90s
```

c. Menghapus Cron Job

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete cronjobs nodejs-cronjob
cronjob.batch "nodejs-cronjob" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

19. Node Selector

Node Selector di Kubernetes memungkinkan kita untuk mengarahkan Pod agar berjalan pada Node tertentu yang memiliki spesifikasi khusus, seperti Node dengan GPU atau hardisk SSD. Dengan menggunakan label pada Node dan mendefinisikan selector pada Pod, kita dapat memastikan bahwa Pod hanya akan dijalankan pada Node yang memenuhi kriteria tersebut. Fitur ini berguna ketika kita membutuhkan sumber daya khusus yang hanya tersedia di beberapa Node, memastikan bahwa aplikasi berjalan pada Node yang paling sesuai dengan kebutuhan spesifikasinya.

a. Menambahkan Label ke Node

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get nodes
NAME STATUS ROLES AGE VERSION
docker-desktop Ready control-plane 22h v1.30.5
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl label node docker-desktop gpu=true
node/docker-desktop labeled
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

Hasil Informasi Label

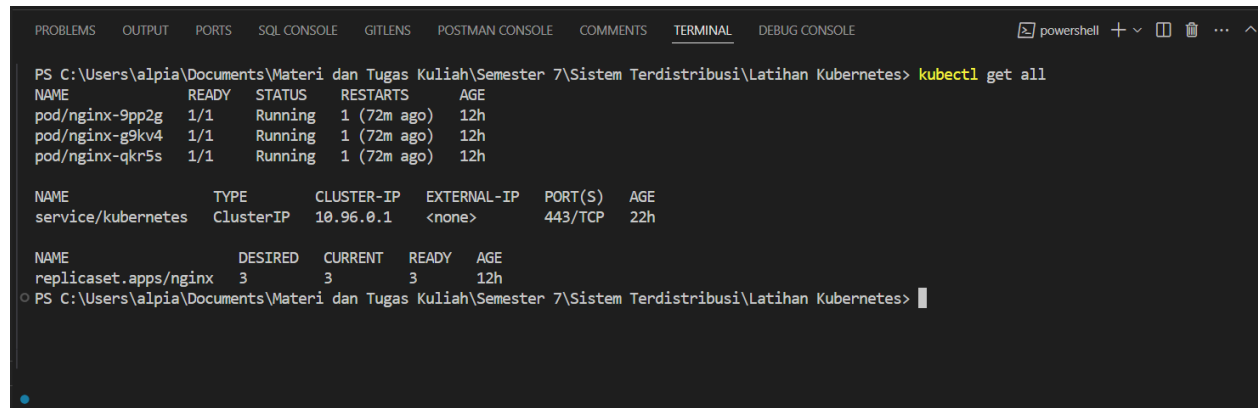
```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe node docker-desktop
Name: docker-desktop
Roles: control-plane
Labels: beta.kubernetes.io/arch=amd64
       beta.kubernetes.io/os=linux
       gpu=true
       kubernetes.io/arch=amd64
       kubernetes.io/hostname=docker-desktop
       kubernetes.io/os=linux
       node-role.kubernetes.io/control-plane=
       node.kubernetes.io/exclude-from-external-load-balancers=
Annotations: kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/cri-dockerd.sock
             node.alpha.kubernetes.io/ttl: 0
             volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp: Wed, 27 Nov 2024 12:02:04 +0800
Taints: <none>
Unschedulable: false
```

20. All

Service di Kubernetes adalah resource yang berfungsi sebagai titik akses stabil untuk menghubungkan client dengan satu atau lebih Pod di belakangnya. Service memiliki IP dan port yang tetap, yang memungkinkan client untuk mengaksesnya tanpa harus mengetahui lokasi spesifik dari Pod yang menjalankan aplikasi. Service secara otomatis akan mendistribusikan trafik ke Pod yang ada, bahkan jika Pod bertambah, berkurang, atau berpindah. Dengan demikian, Service menyediakan lapisan abstraksi

yang memastikan komunikasi tetap lancar antara client dan Pod, meskipun terjadi perubahan pada Pod di backend.

a. Menampilkan Semua Resource

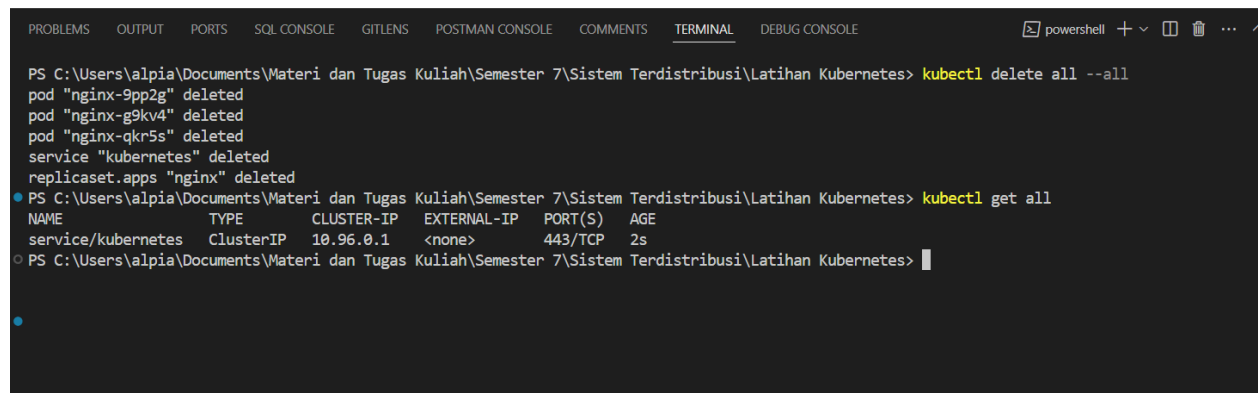


```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get all
NAME                READY   STATUS    RESTARTS   AGE
pod/nginx-9pp2g      1/1     Running   1 (72m ago)  12h
pod/nginx-g9kv4      1/1     Running   1 (72m ago)  12h
pod/nginx-qkr5s      1/1     Running   1 (72m ago)  12h

NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    22h

NAME                DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx  3         3         3       12h
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Menghapus Semua Resource



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete all --all
pod "nginx-9pp2g" deleted
pod "nginx-g9kv4" deleted
pod "nginx-qkr5s" deleted
service "kubernetes" deleted
replicaset.apps "nginx" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get all
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    2s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

21. Membuat Service

Service di Kubernetes mendistribusikan trafik ke Pod yang ada di belakangnya dengan cara seimbang, memastikan beban kerja tersebar secara merata di antara Pod yang relevan. Untuk mengetahui Pod mana yang akan diterima trafik, Service menggunakan label selector, yang memungkinkan Service untuk memilih Pod berdasarkan label tertentu yang diterapkan pada Pod tersebut. Dengan cara ini, Service dapat mengelola dan mengarahkan trafik ke Pod yang sesuai, meskipun jumlah Pod dapat berubah seiring

waktu.

```
Latihan Kubernetes > service-nginx.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: nginx
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        name: nginx
10   template:
11     metadata:
12       name: nginx
13       labels:
14         name: nginx
15     spec:
16       containers:
17         - name: nginx
18           image: nginx
19           ports:
20             - containerPort: 80
21
22   ---
23
24   apiVersion: v1
25   kind: Service
26   metadata:
27     name: nginx-service
28   spec:
29     selector:
30       name: nginx
31     ports:
32       - port: 8080
33         targetPort: 80
34
35   ---
36
37   apiVersion: v1
38   kind: Pod
39   metadata:
40     name: curl
41     labels:
42       name: curl
43   spec:
44     containers:
```

a. MembuatService

```
PROBLEMS  OUTPUT  PORTS  SQL CONSOLE  GITLENS  POSTMAN CONSOLE  COMMENTS  TERMINAL  DEBUG CONSOLE  powershell - Latihan Kubernetes + - [ ] ... ^ x
● PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f service-nginx.yaml
replicaset.apps/nginx created
service/nginx-service created
pod/curl created
○ PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> |
```

b. Melihat Service

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	14m
nginx-service	ClusterIP	10.111.132.96	<none>	8080/TCP	71s

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/curl	1/1	Running	0	76s
pod/nginx-lclkc	1/1	Running	0	76s
pod/nginx-n9m2b	1/1	Running	0	76s
pod/nginx-rrwgl	1/1	Running	0	76s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	14m
service/nginx-service	ClusterIP	10.111.132.96	<none>	8080/TCP	76s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/nginx	3	3	3	76s

c. Menghapus Service

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	85s
nginx-service	ClusterIP	10.104.75.107	<none>	8080/TCP	21s

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete service nginx-service
```

```
service "nginx-service" deleted
```

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

d. Mengakses Service dari Dalam Cluster

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl exec -it curl -- /bin/sh
```

```
/ # curl http://10.111.132.96:8080
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Welcome to nginx!</title>
```

```
<style>
```

```
html { color-scheme: light dark; }
```

```
body { width: 35em; margin: 0 auto;
```

```
font-family: Tahoma, Verdana, Arial, sans-serif; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to nginx!</h1>
```

```
<p>If you see this page, the nginx web server is successfully installed and
```

```
working. Further configuration is required.</p>
```



```
<p>For online documentation and support please refer to
```

```
<a href="http://nginx.org/">nginx.org</a>.<br/>
```

```
Commercial support is available at
```

```
<a href="http://nginx.com/">nginx.com</a>.</p>
```



```
<p><em>Thank you for using nginx.</em></p>
```

```
</body>
```

```
</html>
```


```
/ #
```

22. Mengakses Service

Untuk mengakses Pod lain melalui Service di Kubernetes, kita bisa menggunakan DNS internal yang disediakan oleh Kubernetes, sehingga tidak perlu memasukkan IP address secara manual. Setiap Service

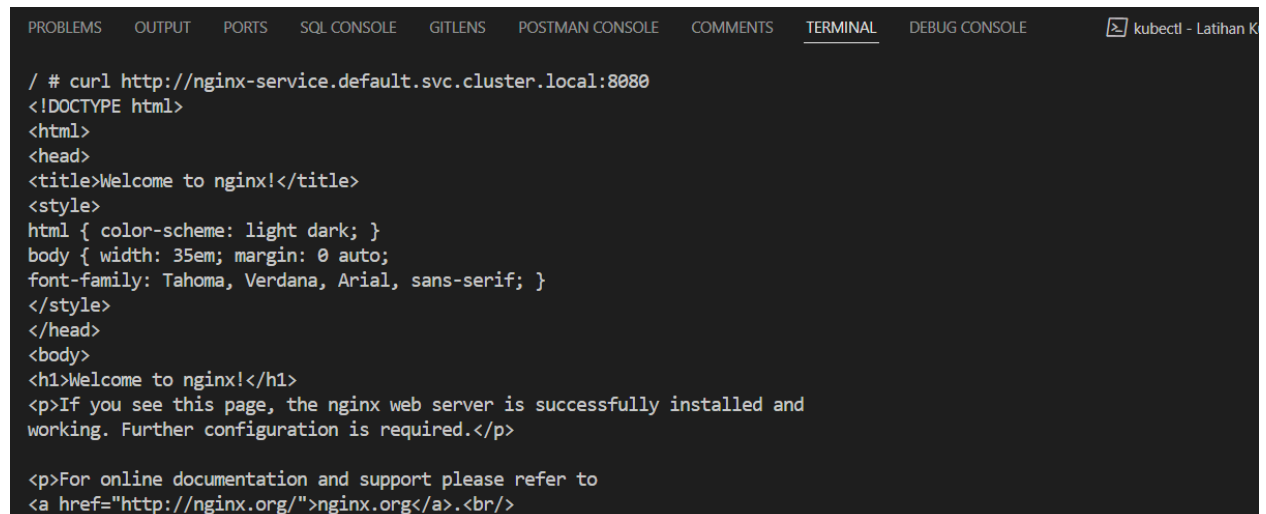
di Kubernetes secara otomatis mendapatkan nama DNS yang berbentuk `*<nama-service>.<namespace>.svc.cluster.local*`. Aplikasi di Pod dapat menggunakan nama DNS ini untuk mengakses Service tanpa perlu mengetahui IP address-nya. Dengan cara ini, Kubernetes secara otomatis akan mengarahkan trafik ke IP Pod yang relevan di belakang Service, sehingga memudahkan komunikasi antar Pod dalam cluster tanpa perlu konfigurasi manual setiap kali IP berubah.

a. Melihat Environment Variable



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl exec -it curl -- /bin/sh
/ # env
KUBERNETES_SERVICE_PORT=443
KUBERNETES_PORT=tcp://10.96.0.1:443
HOSTNAME=curl1
SHLV=1
HOME=/root
PKG_RELEASE=1
TERM=xterm
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
NGINX_VERSION=1.17.10
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
NGINX_SERVICE_SERVICE_HOST=10.99.19.66
NGINX_SERVICE_PORT_8080_TCP_ADDR=10.99.19.66
KUBERNETES_PORT_443_TCP_PORT=443
NJS_VERSION=0.3.9
KUBERNETES_PORT_443_TCP_PROTO=tcp
NGINX_SERVICE_PORT_8080_TCP_PORT=8080
```

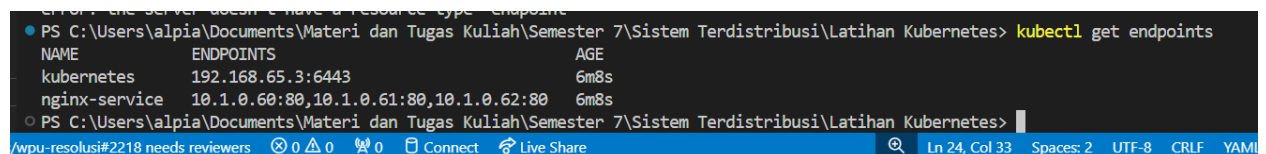
b. Mengakses Menggunakan DNS



```
/ # curl http://nginx-service.default.svc.cluster.local:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
```

c. Melihat Semua Endpoint



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get endpoints
NAME           ENDPOINTS                                     AGE
kubernetes     192.168.65.3:6443                             6m8s
nginx-service  10.1.0.60:80,10.1.0.61:80,10.1.0.62:80        6m8s
```

23. External Service

Service di Kubernetes umumnya digunakan sebagai gateway untuk mengakses Pod internal di dalam cluster. Namun, Service juga dapat berfungsi sebagai gateway untuk aplikasi eksternal di luar Kubernetes

cluster. Dengan menggunakan tipe Service seperti `LoadBalancer` atau `NodePort`, aplikasi eksternal dapat mengakses layanan yang ada di dalam cluster. Ini memungkinkan komunikasi dua arah antara aplikasi internal Kubernetes dan sistem atau layanan yang berada di luar cluster, memfasilitasi integrasi dengan layanan eksternal.

```
service-example.yaml U X
Latihan Kubernetes > service-example.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: example-service
5    labels:
6      name: example-service
7  spec:
8    type: ExternalName
9    externalName: example.com
10   ports:
11   - port: 80
12
13   ---
14
15  apiVersion: v1
16  kind: Pod
17  metadata:
18    name: curl
19    labels:
20      name: curl
21  spec:
22    containers:
23    - name: curl
24      image: khannedy/nginx-curl
```

a. Melihat Service Endpoint

```
PROBLEMS  OUTPUT  PORTS  SQL CONSOLE  GITLENS  POSTMAN CONSOLE  COMMENTS  TERMINAL  DEBUG CONSOLE  powershell - Latihan Kubernetes
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe service example-service
Name:         example-service
Namespace:    default
Labels:       name=example-service
Annotations:  <none>
Selector:     <none>
Type:         ExternalName
IP Families:  <none>
IP:           <none>
IPs:          <none>
External Name: example.com
Port:         <unset> 80/TCP
TargetPort:   80/TCP
Endpoints:    <none>
Session Affinity: None
Events:       <none>
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

24. Sevice Load Balancer

Cloud provider seperti Google Cloud atau Amazon Web Services (AWS) menawarkan layanan Cloud

LoadBalancer yang dapat digunakan oleh Kubernetes untuk mengekspos Service ke dunia luar. Dengan tipe Service 'LoadBalancer', Kubernetes akan membuat sebuah LoadBalancer yang diteruskan ke NodePort untuk mendistribusikan trafik secara merata ke Node dalam cluster. Namun, layanan ini tidak dapat diuji di lingkungan lokal seperti Minikube, karena Minikube tidak menyediakan integrasi langsung dengan Cloud LoadBalancers. Sebagai alternatif, di lingkungan lokal, kita bisa menggunakan tipe Service lain seperti 'NodePort' atau 'Ingress' untuk meniru perilaku load balancing.

```
service-nginx-loadbalancer.yaml U X
Latihan Kubernetes > service-nginx-loadbalancer.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: nginx
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        name: nginx
10   template:
11     metadata:
12       name: nginx
13       labels:
14         name: nginx
15     spec:
16       containers:
17         - name: nginx
18           image: nginx
19           ports:
20             - containerPort: 80
21
22   ---
23
24   apiVersion: v1
25   kind: Service
26   metadata:
27     name: nginx-service
28   spec:
29     type: LoadBalancer
30     selector:
31       name: nginx
32     ports:
33       - port: 80
34         targetPort: 80
```

25. Ingress

Ingress adalah resource Kubernetes yang digunakan untuk mengekspos Service ke dunia luar, namun dengan cara yang lebih fleksibel dibandingkan dengan LoadBalancer atau NodePort. Dengan menggunakan Ingress, client hanya perlu mengetahui satu IP address, yaitu IP address dari Ingress itu sendiri. Ketika client mengirimkan request, pemilihan service yang tepat ditentukan berdasarkan hostname dari request tersebut. Namun, perlu diingat bahwa Ingress hanya mendukung protokol HTTP dan tidak dapat digunakan untuk protokol lain seperti TCP atau UDP. Ingress memungkinkan pengaturan routing yang lebih canggih, seperti path-based routing dan virtual hosting.

```

service-nginx-ingress.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: nginx
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        name: nginx
10   template:
11     metadata:
12       name: nginx
13       labels:
14         name: nginx
15     spec:
16       containers:
17         - name: nginx
18           image: nginx
19           ports:
20             - containerPort: 80
21
22 ---
23
24 apiVersion: v1
25 kind: Service
26 metadata:
27   name: nginx-service
28 spec:
29   selector:
30     name: nginx
31   ports:
32     - port: 80
33       targetPort: 80
34
35 ---

```

service-nginx-ingress.yaml

a. Menjalankan Ingress di Minikube

minikube addons list

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> minikube addons list

ADDON NAME	PROFILE	STATUS	MAINTAINER
ambassador	minikube	disabled	3rd party (Ambassador)
auto-pause	minikube	disabled	minikube
cloud-spanner	minikube	disabled	Google
csi-hostpath-driver	minikube	disabled	Kubernetes
dashboard	minikube	disabled	Kubernetes
default-storageclass	minikube	enabled ✓	Kubernetes
efk	minikube	disabled	3rd party (Elastic)
freshpod	minikube	disabled	Google
gcp-auth	minikube	disabled	Google
gvisor	minikube	disabled	minikube
headlamp	minikube	disabled	3rd party (kinvolk.io)
helm-tiller	minikube	disabled	3rd party (Helm)
inacel	minikube	disabled	3rd party (InAccel [info@inacel.com])
ingress	minikube	disabled	Kubernetes
ingress-dns	minikube	disabled	minikube
inspektor-gadget	minikube	disabled	3rd party (inspektor-gadget.io)
istio	minikube	disabled	3rd party (Istio)
istio-provisioner	minikube	disabled	3rd party (Istio)
kong	minikube	disabled	3rd party (Kong HQ)
kubeflow	minikube	disabled	3rd party
kubevirt	minikube	disabled	3rd party (KubeVirt)
logviewer	minikube	disabled	3rd party (unknown)
metallb	minikube	disabled	3rd party (MetalLB)
metrics-server	minikube	disabled	Kubernetes
nvidia-device-plugin	minikube	disabled	3rd party (NVIDIA)

minikube addons enable ingress

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> minikube addons enable ingress
ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
After the addon is enabled, please run "minikube tunnel" and your ingress resources would be available at "127.0.0.1"
  Using image registry.k8s.io/ingress-nginx/controller:v1.11.2
  Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.3
  Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.3
Verifying ingress addon...
The 'ingress' addon is enabled
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

kubectl get pods --namespace kube-system

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pods --namespace kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-6f6b679f8f-g9fvd           1/1     Running   0           7m42s
etcd-minikube                      1/1     Running   0           7m47s
kube-apiserver-minikube             1/1     Running   0           7m47s
kube-controller-manager-minikube    1/1     Running   0           7m47s
kube-proxy-fzwdk                   1/1     Running   0           7m43s
kube-scheduler-minikube             1/1     Running   0           7m47s
storage-provisioner                 1/1     Running   0           7m44s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Membuat Ingress

```
PROBLEMS  OUTPUT  PORTS  SQL CONSOLE  GITLENS  POSTMAN CONSOLE  COMMENTS  TERMINAL  DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f service-nginx-ingress.yaml
1
>>
replicaset.apps/nginx created
service/nginx-service created
ingress.networking.k8s.io/nginx-ingress created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

c. Melihat Ingress

```
PROBLEMS  OUTPUT  PORTS  SQL CONSOLE  GITLENS  POSTMAN CONSOLE  COMMENTS  TERMINAL  DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get ingresses
NAME      CLASS  HOSTS                ADDRESS      PORTS  AGE
nginx-ingress  nginx  nginx.khannedy.local  192.168.49.2  80     18s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

d. Menghapus Ingress

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete ingress nginx-ingress
ingress.networking.k8s.io "nginx-ingress" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

e. Melihat Ip Minikube

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + v
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> minikube ip
192.168.49.2
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

f. Setting Host

```
service-nginx.yaml U service-nginx-ingress.yaml U hosts X
C: > etc > hosts
1 192.168.49.2 nginx.khannedy.local
```


26. Multi Container Pod

```
multi-container-pod.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: nginx
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        name: nginx
10   template:
11     metadata:
12       name: nginx
13       labels:
14         name: nginx
15     spec:
16       containers:
17         - name: nginx
18           image: nginx
19           ports:
20             - containerPort: 80
21         - name: nodejs-web
22           image: khannedy/nodejs-web
23           ports:
24             - containerPort: 3000
25
26   ---
27
28   apiVersion: v1
29   kind: Service
30   metadata:
31     name: nginx-service
32   spec:
33     selector:
34       name: nginx
35     ports:
36       - port: 8080
37         targetPort: 80
38       - port: 3000
39         targetPort: 3000
39
```

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f multi-container-pod.yaml
replicaset.apps/nginx created
service/nginx-service created
pod/curl created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

```
replicaset.apps/nginx
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl exec -it curl -- /bin/sh
/ # curl http://10.111.22.225:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
```

```
</html>
/ # curl http://10.111.22.225:3000
Hello World/ #
```

you-resolusi#2218 needs reviewers ⓘ 0 ⚠ 0 📶 0 📄 Com

27. Config Map

```
configmap.yaml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: nodejs-env-config
5  data:
6    APPLICATION: My Cool Application
7    VERSION: 1.0.0
8
9  ---
10
11  apiVersion: apps/v1
12  kind: ReplicaSet
13  metadata:
14    name: nodejs-env
15  spec:
16    replicas: 3
17    selector:
18      matchLabels:
19        name: nodejs-env
20    template:
21      metadata:
22        name: nodejs-env
23        labels:
24          name: nodejs-env
25      spec:
26        containers:
27          - name: nodejs-env
28            image: khannedy/nodejs-env
29            ports:
30              - containerPort: 3000
31            envFrom:
32              - configMapRef:
33                name: nodejs-env-config
34
35  ---
36
37  apiVersion: v1
38  kind: Service
```

a. Membuat ConfigMap

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f configmap.yaml
configmap/nodejs-env-config created
replicaset.apps/nodejs-env created
service/nodejs-env-service created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> 
```

b. Melihat ConfigMap

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get configmaps
NAME          DATA  AGE
kube-root-ca.crt    1      8h
nodejs-env-config  2     46s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe configmap kube-root-ca.crt
Name:         kube-root-ca.crt
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/description: Contains a CA bundle that can be used to verify the kube-apiserver when using internal endpoints such as the internal service IP or
              kubern...

Data
====
ca.crt:
-----BEGIN CERTIFICATE-----
MIIDBjCCAc6gAwIBAgIBATANBgkqhkiG9w0BAQsFADAVMRMwEQYDVQDEwptaw5p
a3ViZUNBM4XDTE0MTEyNzE1NDk1N1oXDTE0MTEyNzE1NDk1N1owFTETMBEGA1UE
```

c. Menghapus ConfigMap

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete configmap kube-root-ca.crt
configmap "kube-root-ca.crt" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> |
```

28. Secret

ConfigMap dan Secret keduanya digunakan untuk menyimpan konfigurasi di Kubernetes, namun keduanya memiliki perbedaan penting dalam hal keamanan. ConfigMap digunakan untuk menyimpan data konfigurasi yang tidak sensitif, sementara Secret digunakan untuk menyimpan data sensitif seperti username, password, API key, atau secret key. Data dalam Secret disimpan dengan cara yang lebih aman, yakni hanya didistribusikan ke Node yang membutuhkan dan disimpan di memory, bukan di storage fisik. Selain itu, Secret juga dienkripsi saat disimpan di etcd pada master node, menjaga kerahasiaannya. Secara sederhana, ConfigMap cocok untuk konfigurasi umum, sementara Secret digunakan untuk informasi yang

perlu dilindungi.

```
secret.yaml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: nodejs-env-config
5  data:
6    APPLICATION: My Cool Application
7
8  ---
9
10 apiVersion: v1
11 kind: Secret
12 metadata:
13   name: nodejs-env-secret
14 stringData:
15   VERSION: 1.0.0
16
17 ---
18
19 apiVersion: apps/v1
20 kind: ReplicaSet
21 metadata:
22   name: nodejs-env
23 spec:
24   replicas: 3
25   selector:
26     matchLabels:
27       name: nodejs-env
28   template:
29     metadata:
30       name: nodejs-env
31     labels:
32       name: nodejs-env
33   spec:
```

a. Membuat Secret

```
PROBLEMS  OUTPUT  PORTS  SQL CONSOLE  GITLENS  POSTMAN CONSOLE  COMMENTS  TERMINAL  DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f secret.yaml
configmap/nodejs-env-config created
replicaset.apps/nodejs-env created
service/nodejs-env-service created
```

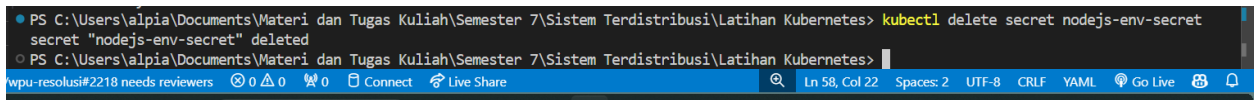
b. Melihat Secret

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get secrets
NAME                TYPE      DATA      AGE
nodejs-env-secret   Opaque    1          5m10s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe secret nodejs-env-secret
Name:         nodejs-env-secret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type: Opaque

Data
====
VERSION: 5 bytes
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

c. Menghapus Secret



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete secret nodejs-env-secret
secret "nodejs-env-secret" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

29. Downward API

Downward API di Kubernetes memungkinkan akses ke informasi dinamis seputar Pod dan Node, seperti nama Pod, namespace, status, dan informasi lainnya, melalui environment variable. Meskipun disebut API, Downward API bukanlah RESTful API, melainkan cara untuk mengakses data tersebut langsung di dalam Pod. Dengan menggunakan Downward API, aplikasi yang berjalan di dalam Pod dapat memperoleh informasi terkait lingkungan atau status saat itu tanpa perlu konfigurasi manual atau pengaturan eksternal, menjadikannya sangat berguna untuk aplikasi yang memerlukan data dinamis.

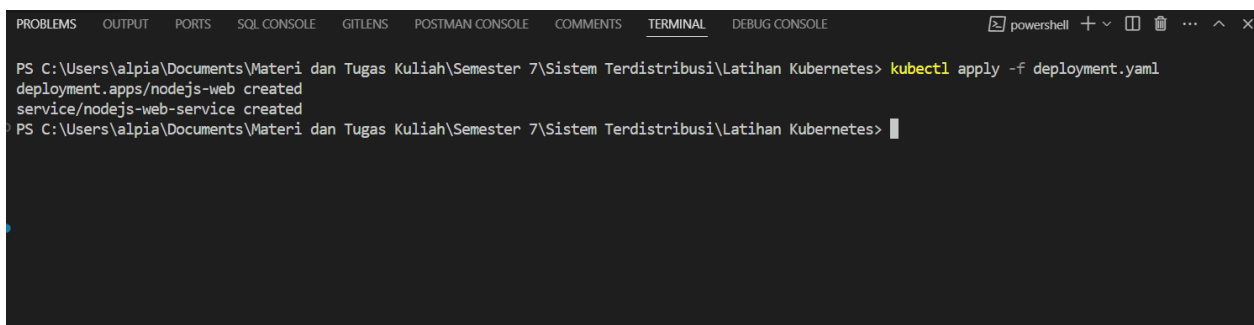
Example: <https://github.com/khannedy/belajar-kubernetes/blob/master/examples/downward-api.yaml>

30. Deployment

Deployment di Kubernetes adalah resource yang digunakan untuk melakukan deployment dan update aplikasi secara deklaratif menggunakan file konfigurasi, yang mengurangi risiko downtime. Ketika kita membuat sebuah Deployment, Kubernetes secara otomatis membuat ReplicaSet, yang selanjutnya akan membuat Pod untuk menjalankan aplikasi. Deployment memungkinkan kita untuk mengelola versi aplikasi dan melakukan update dengan cara yang lebih terstruktur dan otomatis, memastikan aplikasi selalu tersedia dan terhindar dari kesalahan manual dalam proses deployment.

Example: <https://github.com/khannedy/belajar-kubernetes/blob/master/examples/deployment.yaml>

a. Membuat Deployment



```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl apply -f deployment.yaml
deployment.apps/nodejs-web created
service/nodejs-web-service created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Melihat Deployment

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nodejs-web    3/3     3            3           23s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe deployments nodejs-web
Name:          nodejs-web
Namespace:     default
CreationTimestamp: Fri, 29 Nov 2024 08:29:34 +0800
Labels:        name=nodejs-web
Annotations:    deployment.kubernetes.io/revision: 1
Selector:      name=nodejs-web
Replicas:      3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  name=nodejs-web
  Containers:
    nodejs-web:
      Image:      khannedy/nodejs-web:1
      Port:      3000/TCP
      Host Port:  0/TCP
      Environment: <none>
      Mounts:      <none>
      Volumes:     <none>
      Node-Selectors: <none>
      Tolerations:  <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Pod Template:
```

c. Menghapus Deployment

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete deployments nodejs-web
deployment.apps "nodejs-web" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

31. Update Deployment

Untuk memperbarui Deployment di Kubernetes, kita cukup menjalankan perintah `apply` dengan konfigurasi terbaru. Setelah itu, Kubernetes secara otomatis membuat ReplicaSet baru dan menyalakan Pod baru berdasarkan konfigurasi yang diperbarui. Setelah Pod baru siap, Pod lama akan dihapus secara otomatis oleh Deployment, memastikan proses update berjalan lancar tanpa downtime. Dengan cara ini,

aplikasi tetap tersedia selama proses pembaruan, dan perubahan dilakukan dengan cara yang seamless.

```
deployment.yaml deployment-update.yaml U X
deployment-update.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nodejs-web
5    labels:
6      name: nodejs-web
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       name: nodejs-web
12   template:
13     metadata:
14       name: nodejs-web
15       labels:
16         name: nodejs-web
17     spec:
18       containers:
19         - name: nodejs-web
20           image: khannedy/nodejs-web:2
21           ports:
22             - containerPort: 3000
```

a. Update Deployment

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
minikube + - [] ... ^

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl apply -f deployment.yaml
deployment.apps/nodejs-web created
service/nodejs-web-service unchanged
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get all

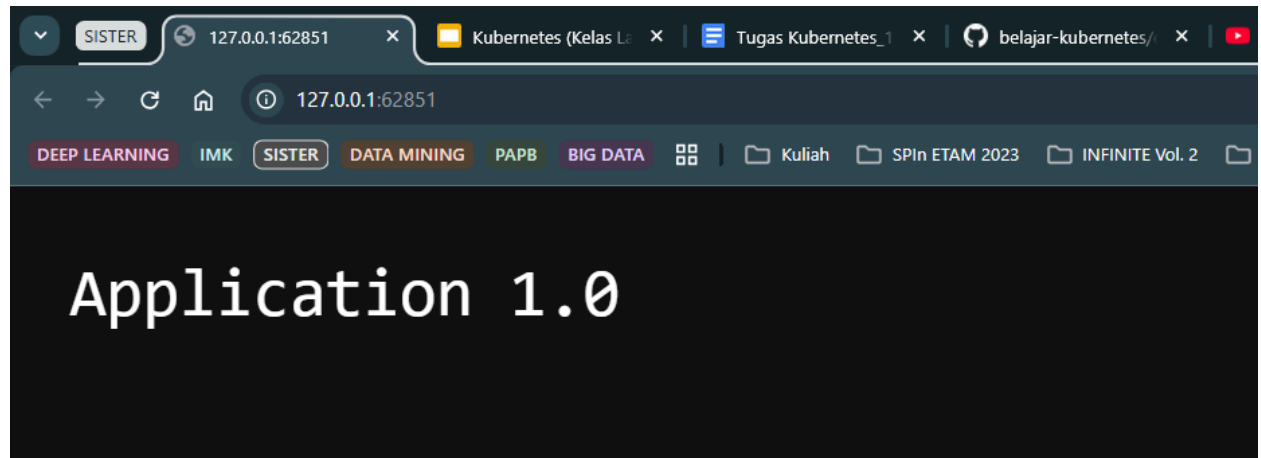
NAME                                READY   STATUS    RESTARTS   AGE
pod/nodejs-web-7db88c745f-kbnw7     1/1     Running   0           52s
pod/nodejs-web-7db88c745f-tdjdx     1/1     Running   0           52s
pod/nodejs-web-7db88c745f-v55x5     1/1     Running   0           52s

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/kubernetes                  ClusterIP     10.96.0.1    <none>        443/TCP          9m31s
service/nodejs-web-service          NodePort      10.97.28.93  <none>        3000:30001/TCP   5m39s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-web          3/3     3            3           52s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nodejs-web-7db88c745f 3         3         3       52s

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> minikube service nodejs-web-service
```

```

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl apply -f deployment-update.yaml
deployment.apps/nodejs-web configured
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/nodejs-web-7db88c745f-kbnw7     1/1     Running   0           2m32s
pod/nodejs-web-7db88c745f-tjdx      1/1     Running   0           2m32s
pod/nodejs-web-7db88c745f-v55x5     1/1     Running   0           2m32s
pod/nodejs-web-84b4c98bfb-q4rhs     0/1     ContainerCreating 0           7s

NAME                                TYPE               CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/kubernetes                 ClusterIP          10.96.0.1    <none>         443/TCP          11m
service/nodejs-web-service         NodePort           10.97.28.93  <none>         3000:30001/TCP   7m19s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-web          3/3     1             3           2m32s

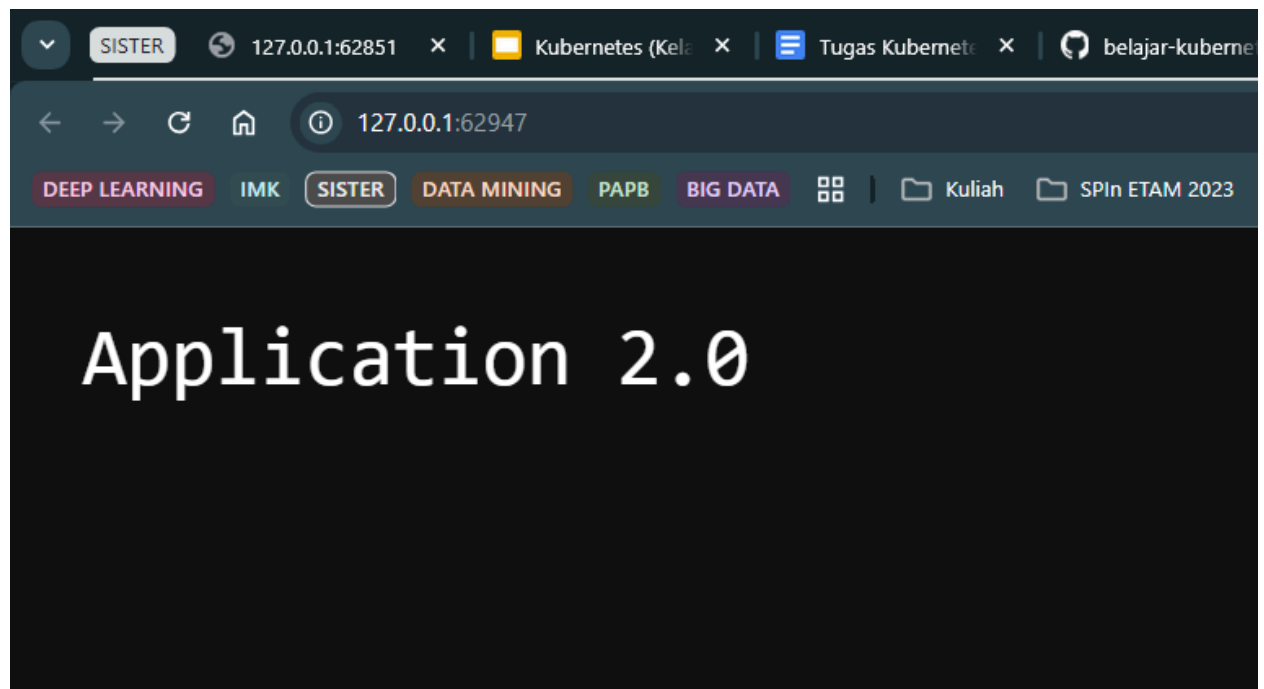
NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nodejs-web-7db88c745f 3         3         3       2m32s
replicaset.apps/nodejs-web-84b4c98bfb 1         1         0       7s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/nodejs-web-7db88c745f-kbnw7     1/1     Terminating 0           2m45s
pod/nodejs-web-7db88c745f-tjdx      1/1     Terminating 0           2m45s
pod/nodejs-web-7db88c745f-v55x5     1/1     Terminating 0           2m45s
pod/nodejs-web-84b4c98bfb-7c224     1/1     Running      0           9s
pod/nodejs-web-84b4c98bfb-m9qtp     1/1     Running      0           11s
pod/nodejs-web-84b4c98bfb-q4rhs     1/1     Running      0           20s

NAME                                TYPE               CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/kubernetes                 ClusterIP          10.96.0.1    <none>         443/TCP          11m
service/nodejs-web-service         NodePort           10.97.28.93  <none>         3000:30001/TCP   7m32s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-web          3/3     3             3           2m45s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nodejs-web-7db88c745f 0         0         0       2m45s

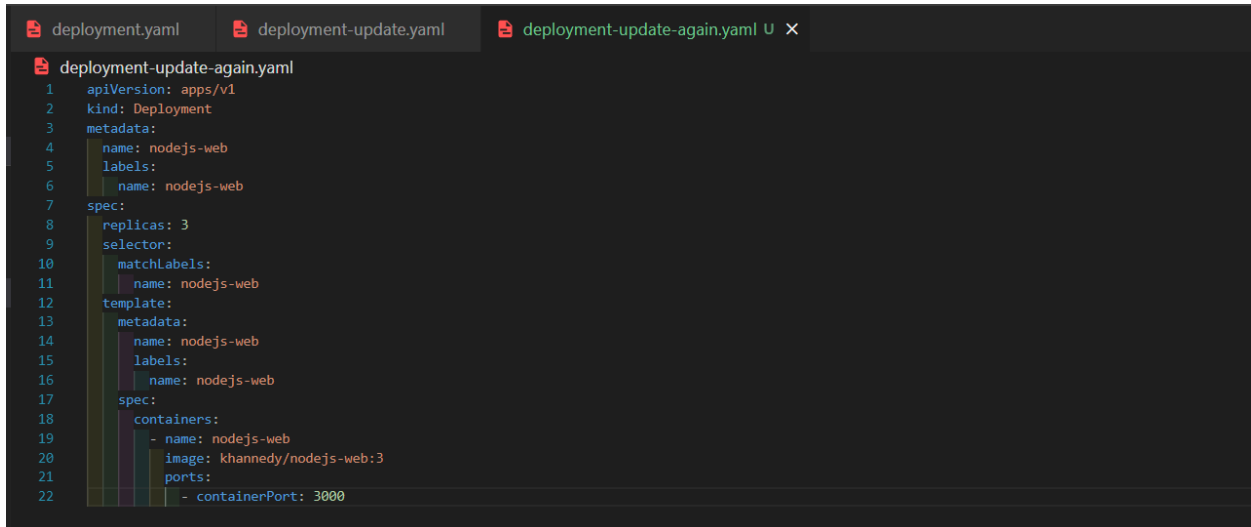
```



32. Rollback Deployment

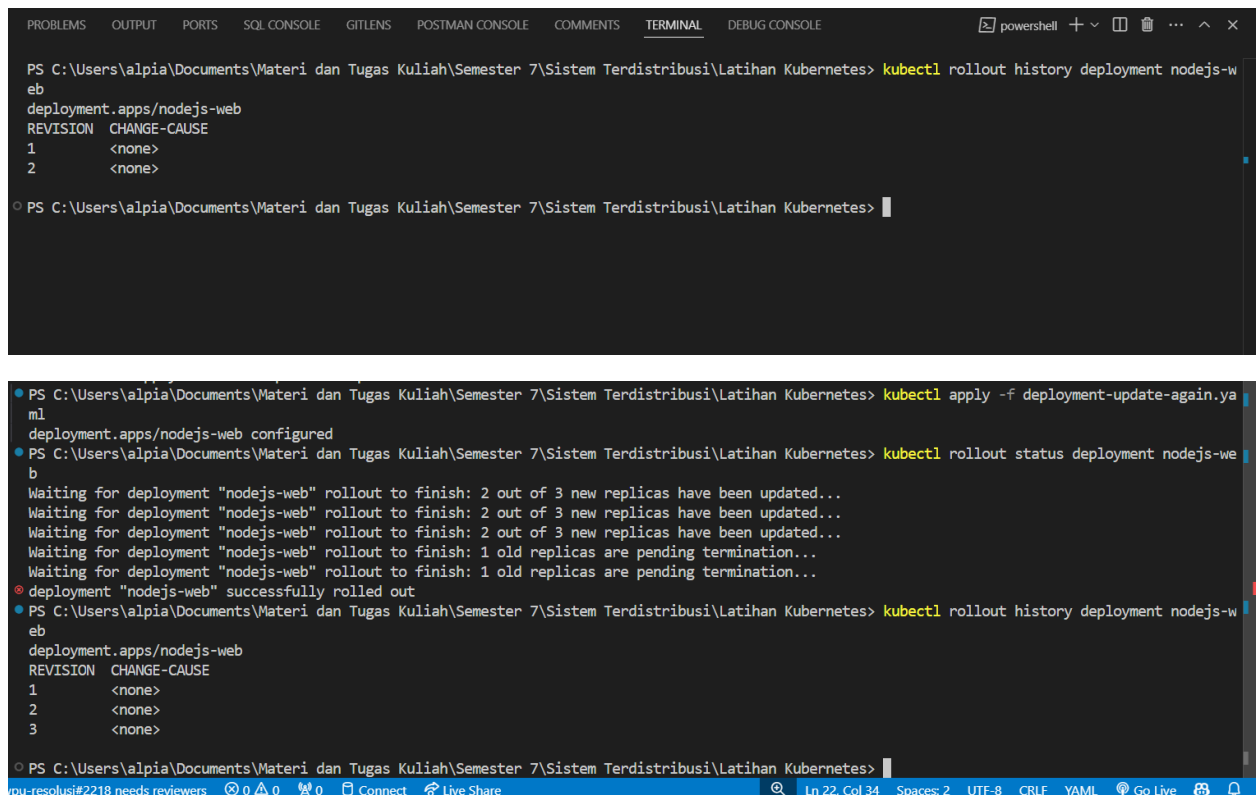
Jika terjadi masalah saat melakukan deploy aplikasi terbaru menggunakan Deployment, cara paling mudah untuk mengembalikan ke versi sebelumnya adalah dengan melakukan rollback. Salah satu cara manual adalah dengan membuat Deployment baru dan mengatur versi aplikasi ke versi sebelumnya. Namun, Kubernetes menyediakan fitur rollout yang memudahkan proses ini. Dengan fitur rollout, kita

bisa dengan mudah melakukan rollback ke versi Deployment sebelumnya tanpa perlu membuat konfigurasi baru, sehingga meminimalkan downtime dan mempercepat pemulihan dari kesalahan deployment.



```
deployment-update-again.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nodejs-web
5    labels:
6      name: nodejs-web
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       name: nodejs-web
12    template:
13     metadata:
14       name: nodejs-web
15       labels:
16         name: nodejs-web
17     spec:
18       containers:
19         - name: nodejs-web
20           image: khannedy/nodejs-web:3
21           ports:
22             - containerPort: 3000
```

a. Rollback Deployment



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl rollout history deployment nodejs-web
deployment.apps/nodejs-web
REVISION  CHANGE-CAUSE
1          <none>
2          <none>

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl apply -f deployment-update-again.yaml
deployment.apps/nodejs-web configured
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl rollout status deployment nodejs-web
Waiting for deployment "nodejs-web" rollout to finish: 2 out of 3 new replicas have been updated...
Waiting for deployment "nodejs-web" rollout to finish: 2 out of 3 new replicas have been updated...
Waiting for deployment "nodejs-web" rollout to finish: 2 out of 3 new replicas have been updated...
Waiting for deployment "nodejs-web" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "nodejs-web" rollout to finish: 1 old replicas are pending termination...
deployment "nodejs-web" successfully rolled out
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl rollout history deployment nodejs-web
deployment.apps/nodejs-web
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
3          <none>

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```


Hasilnya kembali lagi ke versi 2.0



33. Persistent Volume

Persistent Volume (PV) adalah konsep di Kubernetes yang mirip dengan Volume, namun dengan cara kerja yang lebih terstruktur. PV menyediakan cara untuk mengelola penyimpanan yang persisten dan dapat dipakai oleh Pod, meskipun Pod tersebut dihapus atau dipindah. Berbeda dengan Volume biasa yang terikat langsung pada Pod, PV memungkinkan pemisahan antara penyimpanan dan Pod, sehingga data tetap tersedia meskipun Pod dihentikan atau di-restart. Meskipun pembuatan PV sedikit lebih rumit, manfaatnya termasuk kemudahan dalam mengelola penyimpanan, fleksibilitas dalam jenis penyimpanan yang digunakan, dan kemampuan untuk berbagi data antar Pod dalam cluster Kubernetes.

Example: <https://github.com/khannedy/belajar-kubernetes/blob/master/examples/persistent-volume.yaml>

a. Membuat Persistent Volume / Claim

A screenshot of a terminal window with a dark background. The terminal shows the following commands and output:

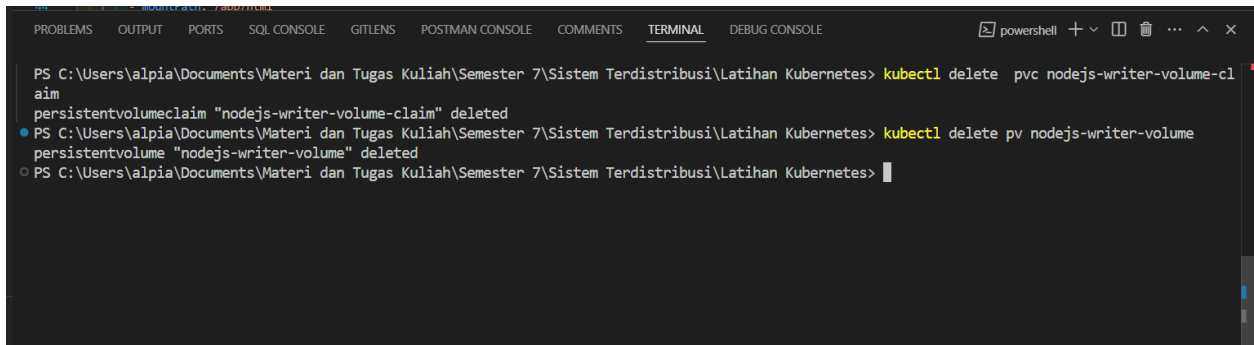
```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f persistent-volume.yaml
persistentvolume/nodejs-writer-volume created
persistentvolumeclaim/nodejs-writer-volume-claim created
pod/nodejs-writer created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Melihat Persistent Volume / Claim

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pv
NAME                                CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM                                STORAGECLASS
nodejs-writer-volume               5Gi       RWO           Retain          Available
pvc-b8f302ff-bf99-40bc-bc7b-58558136cc73  1Gi       RWO           Delete          Bound   default/nodejs-writer-volume-claim  standard
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe pv nodejs-writer-volume
Name: nodejs-writer-volume
Labels: <none>
Annotations: <none>
Finalizers: [kubernetes.io/pv-protection]
StorageClass:
Status: Available
Claim:
Reclaim Policy: Retain
Access Modes: RWO
VolumeMode: Filesystem
Capacity: 5Gi
Node Affinity: <none>
Message:
Source:
  Type: HostPath (bare host directory volume)
  Path: /data/location
  HostPathType:
Events: <none>
```

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pvc
NAME                                STATUS  VOLUME                                CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
nodejs-writer-volume-claim          Bound   pvc-b8f302ff-bf99-40bc-bc7b-58558136cc73  1Gi       RWO           standard      <unset>                 105s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe pvc nodejs-writer-volume-claim
Name: nodejs-writer-volume-claim
Namespace: default
StorageClass: standard
Status: Bound
Volume: pvc-b8f302ff-bf99-40bc-bc7b-58558136cc73
Labels: <none>
Annotations: pv.kubernetes.io/bind-completed: yes
              pv.kubernetes.io/bound-by-controller: yes
              volume.beta.kubernetes.io/storage-provisioner: k8s.io/minikube-hostpath
              volume.kubernetes.io/storage-provisioner: k8s.io/minikube-hostpath
Finalizers: [kubernetes.io/pvc-protection]
Capacity: 1Gi
Access Modes: RWO
VolumeMode: Filesystem
Used By: nodejs-writer
Events:
  Type      Reason              Age    From              Message
  ----      -
  Normal    ExternalProvisioning  2m8s   persistentvolume-controller   Waiting for a volume to be created either by the external provisioner 'k8s.io/minikube-hostpath' or manually by the system administrator. If volume creation is delayed, please verify that the provisioner is running and correctly registered.
  Normal    Provisioning         2m8s   k8s.io/minikube-hostpath_minikube_25b58738-b3a7-4ea9-a324-e9834c8cd209   External provisioner is provisioning volume for claim "default/nodejs-writer-volume-claim"
  Normal    ProvisioningSucceeded 2m8s   k8s.io/minikube-hostpath_minikube_25b58738-b3a7-4ea9-a324-e9834c8cd209   Successfully provisioned volume pvc-b8f302ff-bf99-40bc-bc7b-58558136cc73
```

c. Menghapus Persistent Volume / Claim



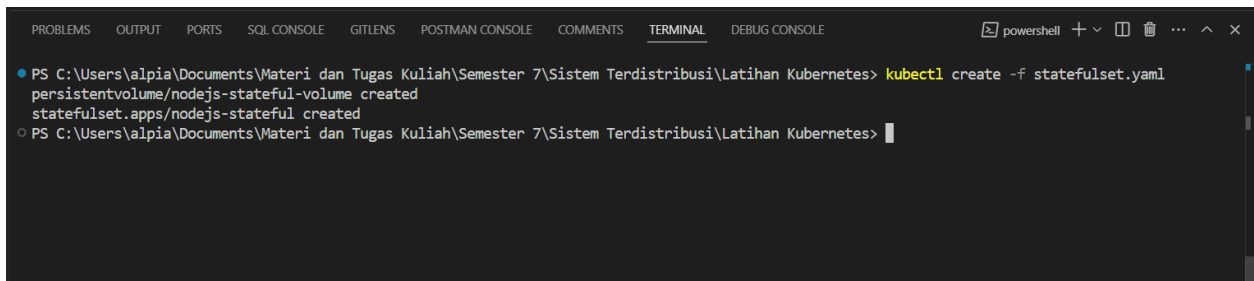
```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete pvc nodejs-writer-volume-claim
persistentvolumeclaim "nodejs-writer-volume-claim" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete pv nodejs-writer-volume
persistentvolume "nodejs-writer-volume" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

34. Membuat StatefulSet

Untuk aplikasi yang stateful, seperti database, Kubernetes menyediakan solusi yang lebih tepat daripada hanya menggunakan PersistentVolume. Aplikasi stateful membutuhkan penyimpanan yang terisolasi untuk setiap Pod, sehingga data yang disimpan di masing-masing Pod tetap terjaga meskipun Pod tersebut di-restart atau diperbarui. Untuk memenuhi kebutuhan ini, Kubernetes menawarkan StatefulSet, yang dirancang untuk aplikasi yang memerlukan state, dengan cara memberikan identitas yang stabil untuk setiap Pod dan mengelola volume penyimpanan yang terpisah untuk setiap Pod. Berbeda dengan Deployment, yang cocok untuk aplikasi stateless, StatefulSet memungkinkan setiap Pod memiliki PersistentVolume yang independen, sehingga data tidak hilang ketika Pod dihentikan atau diganti.

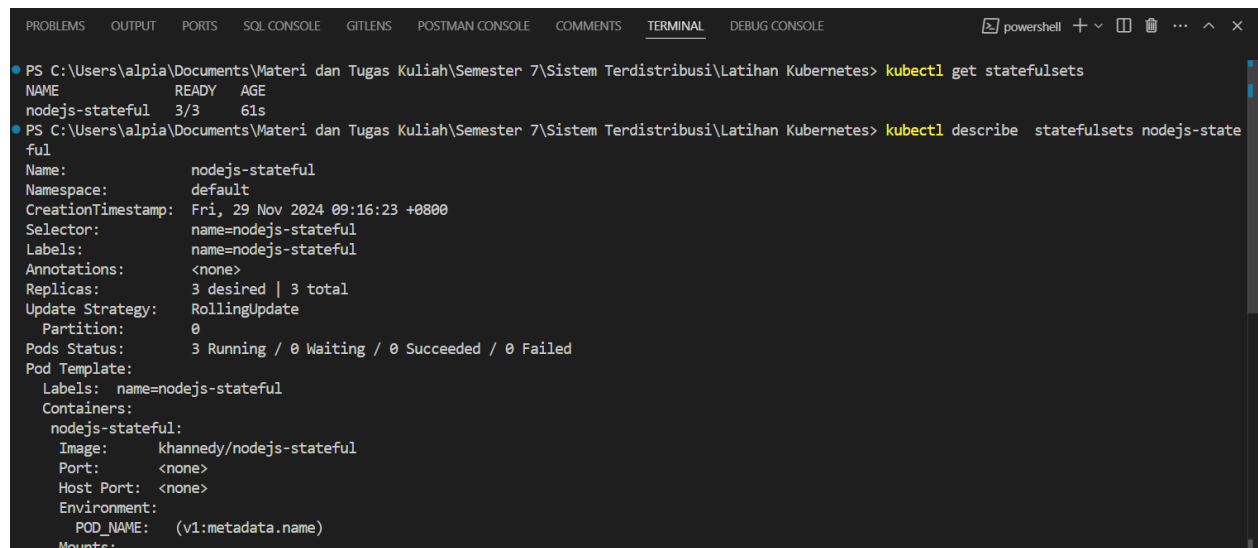
Example: <https://github.com/khannedy/belajar-kubernetes/blob/master/examples/statefulset.yaml>

a. Membuat StatefulSet



```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f statefulset.yaml
persistentvolume/nodejs-stateful-volume created
statefulset.apps/nodejs-stateful created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Melihat StatefulSet

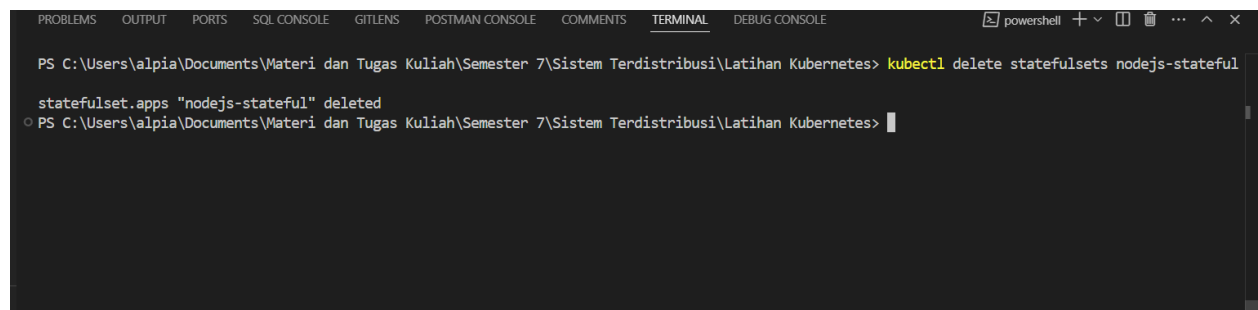
A terminal window with a dark background and light text. The terminal shows two commands and their outputs. The first command is 'kubectl get statefulsets', which outputs a table with columns NAME, READY, and AGE. The second command is 'kubectl describe statefulsets nodejs-stateful', which outputs detailed information about the StatefulSet, including Name, Namespace, CreationTimestamp, Selector, Labels, Annotations, Replicas, Update Strategy, Partition, Pods Status, and Pod Template.

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get statefulsets
NAME                READY   AGE
nodejs-stateful    3/3     61s

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe statefulsets nodejs-stateful
Name:                nodejs-stateful
Namespace:           default
CreationTimestamp:   Fri, 29 Nov 2024 09:16:23 +0800
Selector:            name=nodejs-stateful
Labels:              name=nodejs-stateful
Annotations:         <none>
Replicas:            3 desired | 3 total
Update Strategy:     RollingUpdate
Partition:           0
Pods Status:        3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  name=nodejs-stateful
  Containers:
    nodejs-stateful:
      Image:      khannedy/nodejs-stateful
      Port:       <none>
      Host Port:  <none>
      Environment:
        POD_NAME:  (v1:metadata.name)
  Mounts:

```

c. Menghapus StatefulSet

A terminal window with a dark background and light text. The terminal shows a single command 'kubectl delete statefulsets nodejs-stateful' and its output, which is 'statefulset.apps "nodejs-stateful" deleted'.

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete statefulsets nodejs-stateful
statefulset.apps "nodejs-stateful" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

35. Kubernetes Dashboard

Kubernetes Dashboard adalah aplikasi open-source berbasis web yang memungkinkan pengguna untuk mengelola dan memantau objek Kubernetes secara grafis. Dengan menggunakan Kubernetes Dashboard, pengguna dapat dengan mudah melakukan berbagai operasi, seperti membuat, mengedit, dan menghapus Pod, Deployment, Service, serta objek lainnya, tanpa harus menggunakan terminal. Dashboard ini sangat berguna bagi pengguna yang lebih nyaman dengan antarmuka grafis atau ketika menggunakan Kubernetes di cloud provider yang sudah menyediakan akses ke dashboard, sehingga memudahkan manajemen cluster Kubernetes, baik untuk lingkungan cloud maupun on-premise.

a. Kubernetes Dashboard di Minikube

```
PROBLEMS OUTPUT PORTS SQL CONSOLE GITLENS POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
powershell + - [ ] [X] ...

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> minikube addons enable dashboard
💡 dashboard is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
  ▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
💡 Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube addons enable metrics-server

🌟 The 'dashboard' addon is enabled
• PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> minikube addons list
```

ADDON NAME	PROFILE	STATUS	MAINTAINER
ambassador	minikube	disabled	3rd party (Ambassador)
auto-pause	minikube	disabled	minikube
cloud-spanner	minikube	disabled	Google
csi-hostpath-driver	minikube	disabled	Kubernetes
dashboard	minikube	enabled ✓	Kubernetes
default-storageclass	minikube	enabled ✓	Kubernetes
efk	minikube	disabled	3rd party (Elastic)
freshpod	minikube	disabled	Google
gcp-auth	minikube	disabled	Google
gvisor	minikube	disabled	minikube
headless	minikube	disabled	3rd party (Kubernetes)

```
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get all --namespace kubernetes-dashboard

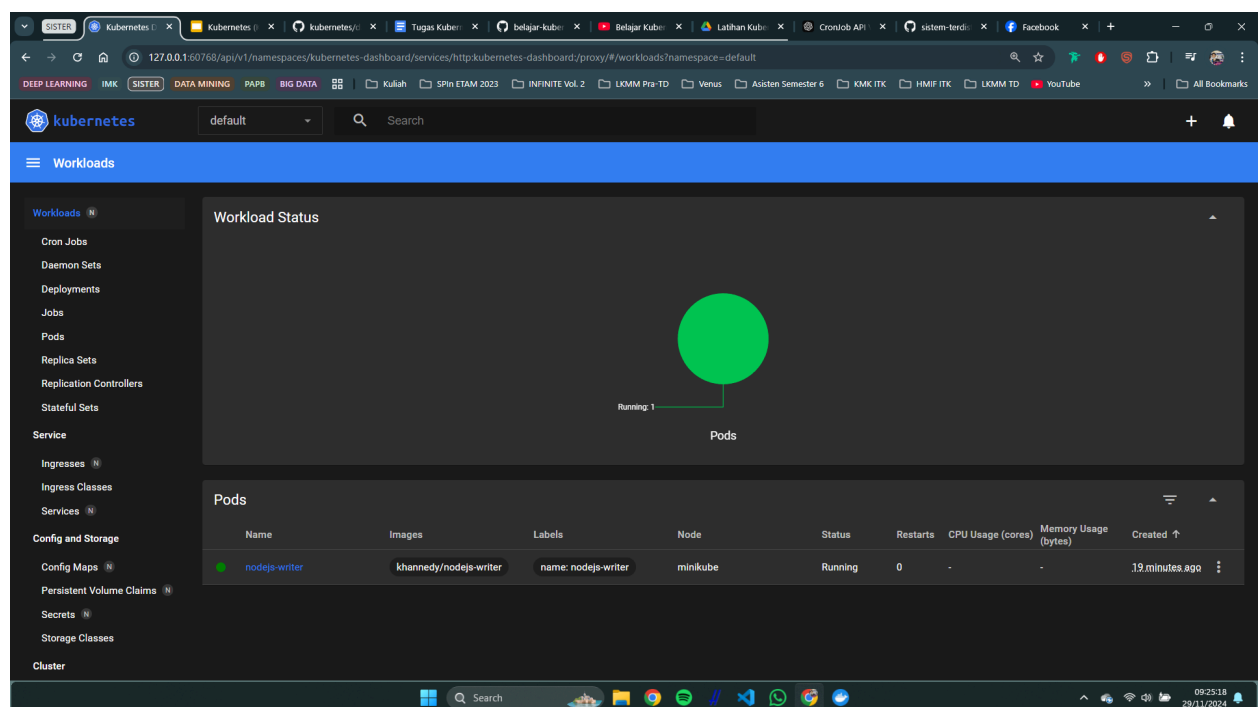
NAME                                     READY   STATUS    RESTARTS   AGE
pod/dashboard-metrics-scraper-c5db448b4-xqcp9   1/1     Running   0           103s
pod/kubernetes-dashboard-695b96c756-7cv89       1/1     Running   0           103s

NAME                                     TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/dashboard-metrics-scraper      ClusterIP     10.97.94.245 <none>        8000/TCP    103s
service/kubernetes-dashboard           ClusterIP     10.110.247.78 <none>        80/TCP      103s

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/dashboard-metrics-scraper  1/1     1             1           104s
deployment.apps/kubernetes-dashboard       1/1     1             1           103s

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/dashboard-metrics-scraper-c5db448b4  1         1         1       103s
replicaset.apps/kubernetes-dashboard-695b96c756      1         1         1       103s

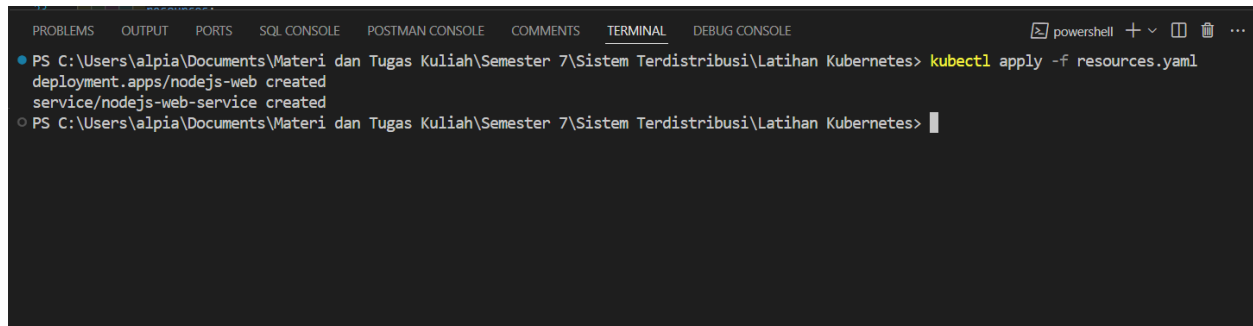
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```



36. Computational Resources

Kubernetes menyediakan mekanisme untuk mengontrol penggunaan sumber daya CPU dan memori melalui Request dan Limit. Request adalah jumlah sumber daya yang dijamin akan diberikan kepada container saat dijalankan, sehingga Kubernetes akan memastikan hanya menjalankan container di Node yang memiliki cukup sumber daya tersebut. Sedangkan Limit adalah batas maksimum sumber daya yang dapat digunakan oleh container, sehingga container tidak akan bisa melebihi batas tersebut, mencegah penggunaan sumber daya yang berlebihan dan mempengaruhi performa Pod lain di Node yang sama. Dengan cara ini, Kubernetes membantu menghindari perebutan sumber daya antar Pod yang dapat menyebabkan penurunan kinerja.

Example: <https://github.com/khannedy/belajar-kubernetes/blob/master/examples/resources.yaml>



```
PROBLEMS OUTPUT PORTS SQL CONSOLE POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl apply -f resources.yaml
deployment.apps/nodejs-web created
service/nodejs-web-service created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

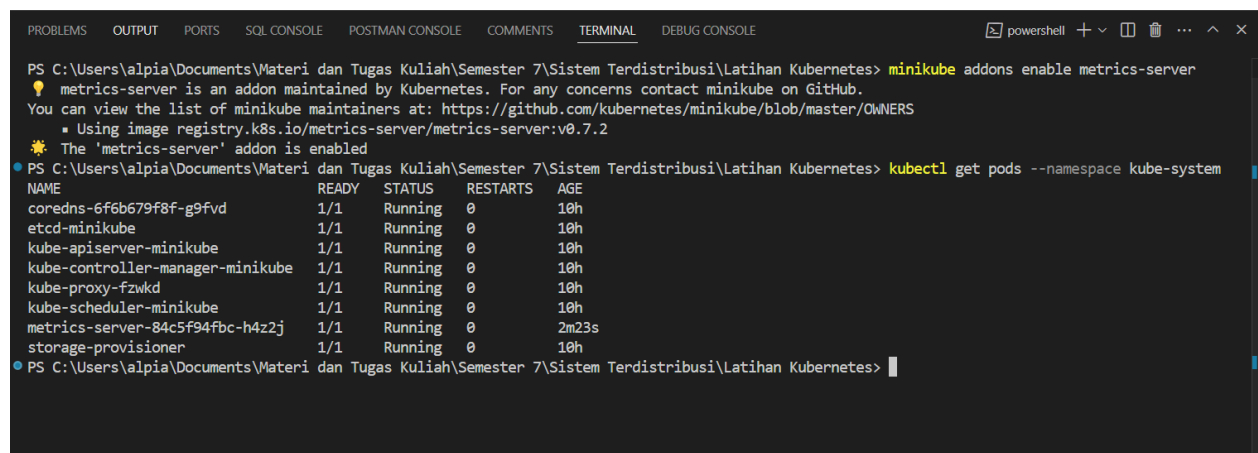
37. Horizontal Pod Autoscaler

Application scaling terbagi menjadi dua jenis: Vertical Scaling dan Horizontal Scaling. Vertical Scaling dilakukan dengan cara meningkatkan kapasitas sumber daya yang digunakan oleh aplikasi, seperti menambah CPU atau memori, tetapi ada batasan karena Pod di Kubernetes hanya bisa menggunakan sumber daya yang tersedia di Node. Sebaliknya, Horizontal Scaling melibatkan penambahan Pod baru untuk mendistribusikan beban kerja, yang lebih efektif dalam jangka panjang karena tidak memerlukan upgrade Node dan memungkinkan aplikasi untuk lebih mudah menyesuaikan dengan peningkatan kebutuhan sumber daya. Horizontal Scaling adalah solusi yang lebih fleksibel dan skalabel dalam menghadapi lonjakan beban.

Example:

<https://github.com/khannedy/belajar-kubernetes/blob/master/examples/horizontal-pod-autoscaler.yaml>

a. Metrics Server di Minikube



```
PROBLEMS OUTPUT PORTS SQL CONSOLE POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> minikube addons enable metrics-server
⚠ metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  ▪ Using image registry.k8s.io/metrics-server/metrics-server:v0.7.2
🌟 The 'metrics-server' addon is enabled
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get pods --namespace kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-6f6b679f8f-g9fvd            1/1     Running   0           10h
etcd-minikube                       1/1     Running   0           10h
kube-apiserver-minikube             1/1     Running   0           10h
kube-controller-manager-minikube    1/1     Running   0           10h
kube-proxy-fzwkd                    1/1     Running   0           10h
kube-scheduler-minikube             1/1     Running   0           10h
metrics-server-84c5f94fbc-h4z2j     1/1     Running   0           2m23s
storage-provisioner                 1/1     Running   0           10h
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes>
```

b. Membuat Horizontal Pod Autoscaler

```
PROBLEMS OUTPUT PORTS SQL CONSOLE POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + v [ ] [ ] ... ^ x

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl create -f horizontal-pod-autoscaler.yaml
deployment.apps/nodejs-web created
service/nodejs-web-service created
horizontalpodautoscaler.autoscaling/hpa-name created
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> [ ]
```

c. Melihat Horizontal Pod Autoscaler

```
PROBLEMS OUTPUT PORTS SQL CONSOLE POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + v [ ] [ ] ... ^ x

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl get hpa
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
hpa-name Deployment/nodejs-web cpu: <unknown>/70%, memory: <unknown>/70% 3 5 3 84s
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl describe hpa hpa-name
Name: hpa-name
Namespace: default
Labels: <none>
Annotations: <none>
CreationTimestamp: Fri, 29 Nov 2024 10:07:14 +0800
Reference: Deployment/nodejs-web
Metrics:
  resource cpu on pods (as a percentage of request): <unknown> / 70%
  resource memory on pods (as a percentage of request): <unknown> / 70%
Min replicas: 3
Max replicas: 5
Deployment pods: 3 current / 3 desired
Conditions:
  Type Status Reason Message
  ----
  AbleToScale True SucceededGetScale the HPA controller was able to get the target's current scale
  ScalingActive False FailedGetResourceMetric the HPA was unable to compute the replica count: failed to get cpu utilization: did not receive metrics for targeted pods (pods might be unready)
Events:
  Type Reason Age From Message
  ----
  Normal SuccessfulRescale 89s horizontal-pod-autoscaler New size: 3; reason: Current number of replicas below Spec.M
```

d. Menghapus Horizontal Pod Autoscaler

```
PROBLEMS OUTPUT PORTS SQL CONSOLE POSTMAN CONSOLE COMMENTS TERMINAL DEBUG CONSOLE powershell + v [ ] [ ] ... ^ x

PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> kubectl delete hpa hpa-name
horizontalpodautoscaler.autoscaling "hpa-name" deleted
PS C:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem Terdistribusi\Latihan Kubernetes> [ ]
```