

Kubernetes Fundamental - Deployment Local to Production Part 3

Detail Materi



Kubernetes Secret

Indikator :
Mengenal Kubernetes Secret, Implementasi pada
Kubernetes Cluster, dan Mengenal ConfigMap



Kubernetes ConfigMap

Modul Sekolah DevOps Cilsy

Hak Cipta © 2020 **PT. Cilsy Fiolution Indonesia**

Hak Cipta dilindungi Undang-Undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk mecropy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis dan Penerbit.

Penulis : Estu Fardani

Editor: Taufik Maulana, Iqbal Ilman Firdaus & Muhammad Fakhri A

Revisi Batch 9

Penerbit : **PT. Cilsy Fiolution Indonesia**

Web Site : <https://cilsyfiolution.com> , <https://devops.cilsy.id>

Sanksi Pelanggaran Pasal 113 Undang-undang Nomor 28 Tahun 2014 tentang Hak Cipta

1. Setiap orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam pasal 9 ayat (1) huruf i untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan atau pidana denda paling banyak Rp100.000.000,00 (seratus juta rupiah).
2. Setiap orang yang dengan tanpa hak dan atau tanpa izin pencipta atau pemegang hak cipta melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf c, huruf d, huruf f, dan atau huruf h, untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah)
3. Setiap orang yang dengan tanpa hak dan atau tanpa izin pencipta atau pemegang hak melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf a, huruf b, huruf e, dan atau huruf g, untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah)
4. Setiap orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah)



Daftar Isi

Cover.....	1
9. Kubernetes Fundamental & Deployment Local to Production.....	4
Learning Outcomes.....	4
Outline Materi.....	4
9.1. Kubernetes ConfigMap.....	5
9.1.1. Persiapan Umum.....	5
9.1.1.1. Membuat ConfigMap.....	5
9.1.1.2. Mengatur YAML Pod Pesbuk.....	6
9.2. Kubernetes Secret.....	9
9.2.1. Persiapan Umum.....	9
9.2.1.1. Membuat Secret Menggunakan Kubectl.....	9
9.2.1.2. Membuat <i>Secret</i> secara <i>Manual</i>	10
9.2.1.3. Menggunakan <i>Secret</i> pada <i>File</i>	11
9.2.1.4. Mengatur YML <i>Pod</i> Facebook.....	13

9.

Kubernetes Fundamental & Deployment Local to Production

Learning Outcomes

Setelah selesai mempelajari bab ini, peserta mampu :

1. Memahami Konsep dan Arsitektur Kubernetes
2. Mempersiapkan Service AWS untuk Integrasi Kubernetes
3. Memahami Konsep Secrets dan ConfigMaps

Outline Materi

1. Kubernetes Secret
2. Kubernetes ConfigMap



9.1. Kubernetes ConfigMap

Banyak aplikasi memerlukan konfigurasi melalui beberapa kombinasi file config, argumen baris perintah, dan variabel lingkungan. Artefak konfigurasi ini harus dipisahkan dari konten image agar aplikasi yang dipaketkan tetap portabel. API ConfigMap menyediakan mekanisme untuk menyuntikkan data konfigurasi ke kontainer sambil tetap menyimpan agnostik container di kubernetes. ConfigMap dapat digunakan untuk menyimpan informasi seperti properti individu atau informasi seluruh file konfigurasi atau gumpalan JSON.

API ConfigMap menyimpan pasangan kunci nilai-nilai yang dapat dikonsumsi dalam pod atau digunakan untuk menyimpan data konfigurasi untuk komponen sistem seperti pengontrol. ConfigMap mirip dengan Secret, tetapi dirancang untuk lebih mendukung kerja dengan string yang tidak mengandung informasi sensitif.

Catatan: ConfigMap tidak dimaksudkan untuk bertindak sebagai pengganti file properti. ConfigMap dimaksudkan untuk bertindak sebagai referensi ke beberapa file properti. Bahasa lainnya dapat dianggap sebagai cara untuk merepresentasikan sesuatu yang mirip dengan direktori / etc, dan file-file di dalamnya, pada komputer Linux. Salah satu contoh model ini adalah membuat Volume Kubernetes dari ConfigMap, di mana setiap item data dalam ConfigMap menjadi file baru.

9.1.1. Persiapan Umum

Kita akan tetap menggunakan repo social media. Namun bedanya, kita tidak mengubah file config.php tapi akan mengganti isi config.php dengan configmap.

9.1.1.1. Membuat ConfigMap

Buat sebuah configmap dengan nama p-fb-configmap.yml lalu masukan konfigurasi berikut

```
apiVersion: v1
```



```
kind: ConfigMap
metadata:
  name: p-fb-configmap
data:
  config.php: |
    <?php
    $db_host = "php-pesbuk";
    $db_user = "devopscilsy";
    $db_pass = "1234567890";
    $db_name = "dbsosmed";

    try {
      //create PDO connection
      $db = new PDO("mysql:host=$db_host;dbname=$db_name", $db_user, $db_pass);
    } catch(PDOException $e) {
      //show error
      die("Terjadi masalah: " . $e->getMessage());
    }
    ?>
```

script dapat dilihat di
<https://gist.github.com/sdcilsy/b02bf1e03645212e4bc7a75377398207>

Kita bisa mengeksekusi perintah berikut untuk membuat **configmap**

```
controlplane $ kubectl apply -f p-fb-configmap.yaml
configmap/p-fb-configmap created
```

9.1.1.2. Mengatur YAML Pod Pesbuk

Lakukan konfigurasi pada bagian env pada p-fb-pod.yml dengan isi konfigurasi sebagai berikut

```
apiVersion: v1
kind: Pod
metadata:
  name: php-pesbuk
```



```
spec:
  volumes:
    - name: config-fb
      configMap:
        name: p-fb-configmap

  containers:
    - name: php-sosmed
      image: sekolahdevopscilsy/php-sosmed
      ports:
        - containerPort: 80
      volumeMounts:
        - name: config-fb
          mountPath: "/var/www/html/config.php"
          subPath: "config.php"

    - name: mysql-sosmed
      image: sekolahdevopscilsy/mysql-sosmed
      ports:
        - containerPort: 3306
      env:
        - name: MYSQL_ROOT_PASSWORD
          value: "1234567890"
```

script dapat dilihat di
<https://gist.github.com/sdcilsy/bbb6472895f9c49fe40f87fe36928dcf>

Kita bisa mengeksekusi perintah berikut untuk membuat **pod**

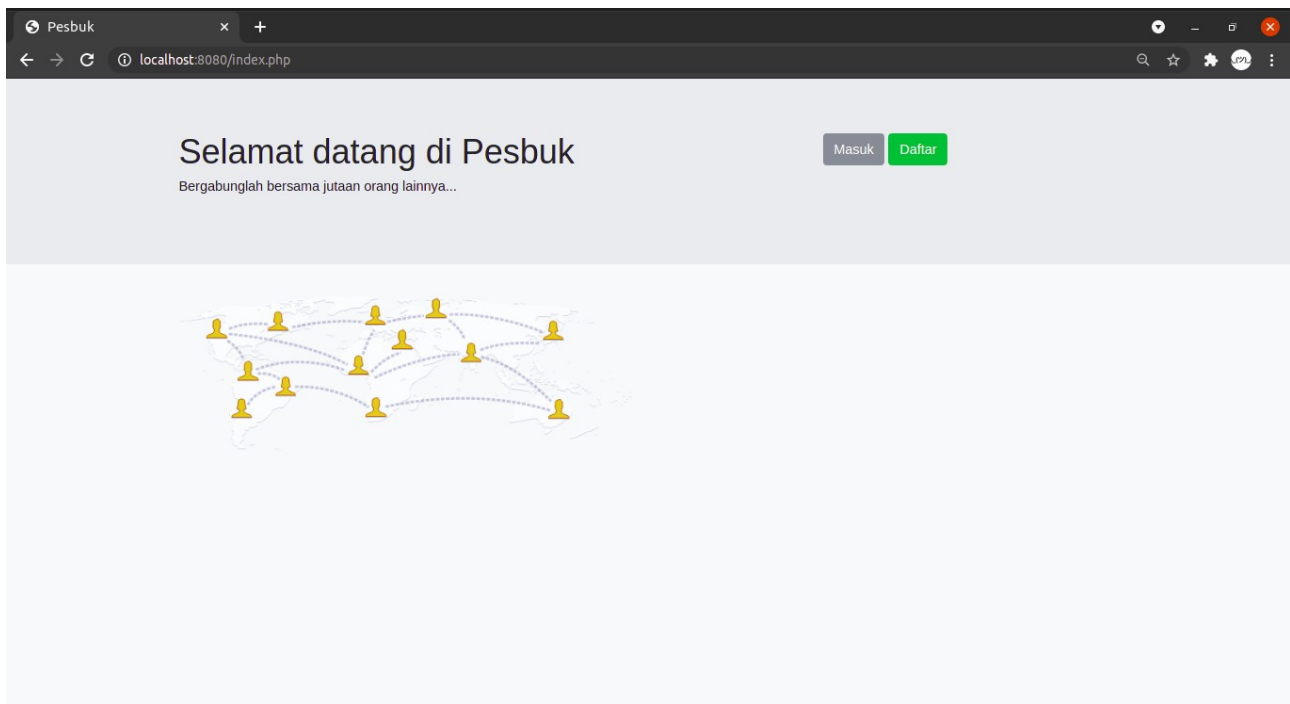
```
controlplane $ kubectl apply -f p-fb-pod.yaml
pod/php-pesbuk created
```

Untuk pengujian, kita bisa menggunakan kubectl port-forward untuk mengekspos pod.

```
controlplane $ kubectl port-forward php-pesbuk 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```



Lalu pada browser kita, akses alamat localhost:8080



Cobalah untuk daftar untuk melihat apakah koneksi ke mysql dapat berjalan ?



9.2. Kubernetes Secret

Kubernetes secret adalah objek yang memungkinkan kita untuk menyimpan dan mengelola informasi sensitif, seperti kata sandi, *token OAuth*, dan kunci ssh. Informasi semacam itu dapat dimasukkan ke dalam spesifikasi Pod atau *container image*. Menyimpan informasi rahasia dalam *Secret* akan menjadi lebih aman dan lebih fleksibel daripada menyimpan dalam plain text, yml, atau di dalam container image.

Untuk menggunakan *Secret*, sebuah Pod membutuhkan referensi ke *secret* yang dituju. Secret dapat digunakan dengan Pod dalam dua cara:

- Sebagai file dalam volume yang dipasang pada satu atau lebih container.
- Menggunakan kubelet saat menarik Image untuk Pod.

9.2.1. Persiapan Umum

9.2.1.1. Membuat Secret Menggunakan Kubectl

Sebelum kita akan membuat Secret di *workspace* dengan menggunakan perintah berikut :

```
#echo -n 'admin' > ./username.txt
#echo -n '1f2d1e2e67df' > ./password.txt
```

Selanjutnya inputkan password yang akan kita buat dengan perintah berikut :

```
#kubectl create secret generic db-user-pass --from-file=./username.txt --from-file=./password.txt
```

Setelah memasukan perintah diatas output yang akan dihasilkan seperti berikut :

```
secret "db-user-pass" created
```

Selanjutnya masukan perintah berikut untuk memeriksa Secret yang sudah ada :

```
#kubectl get secrets
```

Output yang dihasilkan seperti berikut :



NAME	TYPE	DATA	AGE
db-user-pass	Opaque	2	51s

bahkan kita bisa melihat detail secret yang sudah ada dengan menggunakan perintah berikut :

```
#kubectl describe secrets/db-user-pass
```

Maka *output* yang dihasilkan dari perintah diatas sebagai berikut :

```
Name:          db-user-pass
Namespace:     default
Labels:        <none>
Annotations:   <none>

Type:          Opaque

Data
====
password.txt:  12 bytes
username.txt:  5 bytes
```

9.2.1.2. Membuat *Secret* secara *Manual*

kita akan membuat user pada workspace dengan menggunakan perintah berikut:

```
#echo -n 'admin' | base64
```

Output dari perintah diatas sebagai berikut :

```
YWRtaW4=
```

Setelah membuat *user* selanjutnya kita akan membuat *password* dengan memasukan perintah berikut :

```
#echo -n '1f2d1e2e67df' | base64
```

Output dari perintah diatas sebagai berikut :

```
MWYyZDFlMmU2N2Rm
```

Buat sebuah file dengan nama p-secret.yaml yang isinya sebagai berikut:

```
apiVersion: v1
kind: Secret
```



```
metadata:
  name: mysecret
type: Opaque
data:
  username: YWRtaW4=
  password: MWYyZDFLMmU2N2Rm
```

script dapat dilihat di

<https://gist.github.com/sdcilisy/9c7ba0f35c23a656fb7563389c72fbae>

Lakukan kubectl apply dengan menggunakan perintah berikut :

```
#kubectl apply -f ./secret.yaml
```

Output dari perintah diatas sebagai berikut :

```
secret "mysecret" created
```

9.2.1.3. Menggunakan *Secret* pada *File*

Pada praktek ini, kita akan menggunakan repo facebook dengan kredensial **config.php** seperti username dan password, kita masukan kedalam Secret Kubernetes

Kita akan memasukan **username** dan **password** dari mysql kedalam bentuk **base64**. Nantinya, secara otomatis, kubernetes akan mendekripsi username dan password ini menjadi **plain text**.

```
controlplane $ echo -n 'devopscilisy' | base64
ZGV2b3BzY2lsc3k=
controlplane $ echo -n '1234567890' | base64
MTIzNDU2Nzg5MA==
```

Buat *file secret* p-fb-secret.yaml Isi *data* sesuai output pada langkah sebelumnya

```
apiVersion: v1
kind: Secret
metadata:
  name: p-secret-fb
type: Opaque
```



```
data:
  DB_USER: ZGV2b3BzY2lsc3k=
  DB_PASS: MTIzNDU2Nzg5MA==
  MYSQL_ROOT_PASSWORD: MTIzNDU2Nzg5MA==
```

script dapat dilihat di

<https://gist.github.com/sdcilisy/8adc7917ab18916a7f4da2b844200d17>

Lalu kita bisa mengeksekusi perintah berikut untuk membuat **Secret**

```
controlplane $ kubectl apply -f secret.yaml
secret/p-secret-fb created
```

Selanjutnya kita membuat **configmap** file config.php bernama **p-fb-configmap.yaml**. Ubah beberapa variabel seperti **host**, **username**, **password**, dan kita arahkan untuk mengambil ke **variabel environment**.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: p-fb-configmap-secret
data:
  config.php: |
    <?php
    $db_host = getenv("DB_HOST");
    $db_user = getenv("DB_USER");
    $db_pass = getenv("DB_PASS");
    $db_name = "dbsosmed";

    try {
      //create PDO connection
      $db = new PDO("mysql:host=$db_host;dbname=$db_name", $db_user, $db_pass);
    } catch(PDOException $e) {
      //show error
      die("Terjadi masalah: " . $e->getMessage());
    }
    ?>
```

script dapat dilihat di



<https://gist.github.com/sdcilisy/beb8123fb9a8098342dc3f07d36abef3>

9.2.1.4. Mengatur YML *Pod* Facebook

Kita akan melakukan pengaturan pada bagian env dibagian p-fb-pod.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: php-pesbuk-secret
spec:
  volumes:
    - name: config-secret
      configMap:
        name: p-fb-configmap-secret

  containers:
    - name: php-sosmed
      image: sekolahdevopscilisy/php-sosmed
      ports:
        - containerPort: 80
      volumeMounts:
        - name: config-secret
          mountPath: "/var/www/html/config.php"
          subPath: "config.php"
      env:
        - name: DB_HOST
          value: "php-pesbuk-secret"
        - name: DB_USER
          valueFrom:
            secretKeyRef:
              name: p-secret-fb
              key: DB_USER
        - name: DB_PASS
          valueFrom:
            secretKeyRef:
```



```

      name: p-secret-fb
      key: DB_PASS

- name: mysql-sosmed
  image: sekolahdevopscilsy/mysql-sosmed
  ports:
    - containerPort: 3306
  env:
    - name: MYSQL_ROOT_PASSWORD
      valueFrom:
        secretKeyRef:
          name: p-secret-fb
          key: MYSQL_ROOT_PASSWORD

```

script dapat dilihat di

<https://gist.github.com/sdcilsy/99dfd1c2c98fa3a36e4275f03552a633>

Lakukan deployment pada p-fb-deployment.yml dengan menggunakan perintah berikut :

```
#kubectl apply -f p-fb-pod.yaml
```

Menggunakan perintah **kubectl port-forward**, kita dapat menguji apakah pod kita dapat berintegrasi satu sama lain ?



Referensi

<https://kubernetes.io/docs/concepts/configuration/secret/>

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/>

<https://unofficial-kubernetes.readthedocs.io/en/latest/tasks/configure-pod-container/configmap/>

