ondia

Kubernetes Secrets and ConfigMaps

AGENDA

- Secrets
- ConfigMaps

CLARUSWAY©
WAY TO REINVENT YOURSELF

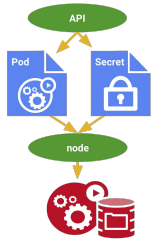
1 Secrets

Why env exist?

```
1   API_KEY="XXX_YYY"
2   PASSWORD="mypassword"
3   HOST="111.121.130.17"
4   PORT="3360"
5   DATABASE="mydb"
6   USER="Mickey"
```


Secrets

A **Secret** is an object that contains a small amount of **sensitive data such as a password**, **a token**, or **a key**. Such information might otherwise be put in a Pod specification or in an image.



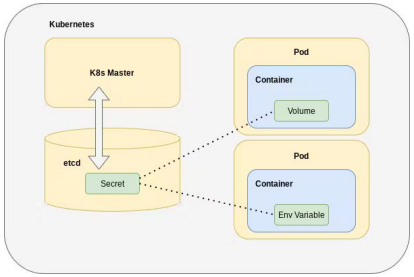
Secrets

Opaque is the default Secret type if omitted from a Secret configuration file. When you create a Secret using kubectl, you will use the generic subcommand to indicate an Opaque Secret type.

Built-in Type	Usage
Opaque	arbitrary user-defined data
kubernetes.io/service-account-token	ServiceAccount token
kubernetes.io/dockercfg	serialized ~/.dockercfg file
kubernetes.io/dockerconfigjson	serialized ~/.docker/config.json file
kubernetes.io/basic-auth	credentials for basic authentication
kubernetes.io/ssh-auth	credentials for SSH authentication
kubernetes.io/tls	data for a TLS client or server
bootstrap.kubernetes.io/token	bootstrap token data

Secrets

Secrets can be mounted as data volumes or exposed as environment variables to be used by a container in a Pod.

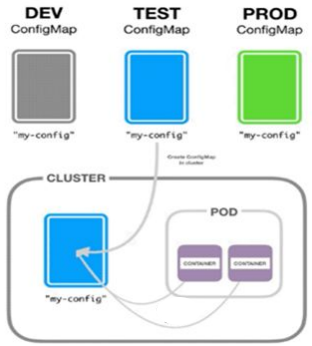


2

ConfigMaps

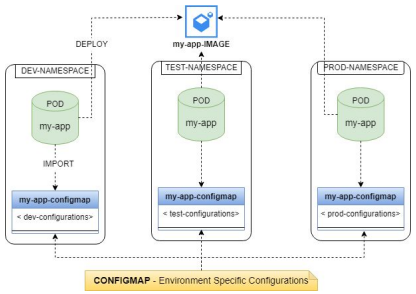
ConfigMaps

- A **ConfigMap** is an API object used to store **non-confidential data in key-value pairs**. Pods can consume ConfigMaps **as environment variables, command-line arguments, or as configuration files in a volume**.



ConfigMaps

- A ConfigMap allows you to **decouple environment-specific configuration** from your container images, so that your applications are easily portable.

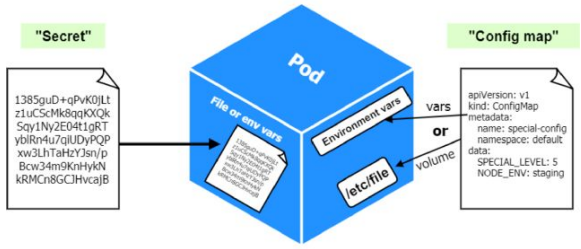



```
env:
  - name: APP_COLOR
    value: pink
```

```
env:
  - name: APP_COLOR
    valueFrom:
      configMapKeyRef:
```

```
env:
  - name: APP_COLOR
    valueFrom:
      secretKeyRef:
```

1. Plain Key Value
2. ConfigMap
3. Secrets



ConfigMaps Samples

- User can create configMap via **Literal** or **from Files**.

- ***Via File:*** A **path to a directory** containing one or more configuration files, indicated using the **—from-file** flag.

kubectl create configmap [NAME] --from-file [/PATH/TO/FILE.PROPERTIES] --from-file [/PATH/TO/FILE2.PROPERTIES]

- User **can also put complete directory**, containing multiple files.

kubectl create configmap [NAME] --from-file [/PATH/TO/DIRECTORY]

ConfigMaps Samples

➢ ***Via Literal Values:*** To create a ConfigMap from literal values **—from-literal**.

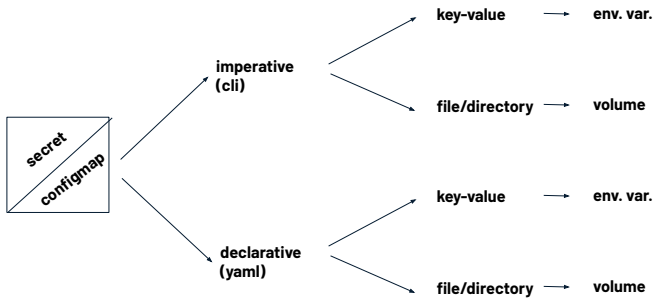
kubectl create configmap literal-data --from-literal key1=value1 --from-literal key2=value2

```
kubectl create configmap special-config --from-literal=special.how=very
--from-literal=special.type=charm
```

➢ Get ConfigMap via CLI.

kubectl get configmaps <confifig-map_Name> -o yaml/json

Handson Workflow



ondia

THANKS!

Any questions?

Powered by Clarusway

Control Plane Components