Application Configuration

» Secrets

Overview

View the list of secrets.

```
kubectl get secrets
```

Creating Secrets

Create a Secret from literal values.

```
kubectl create secret generic my-secret --from-literal password=supersecret --from-literal apikey=confidential
```

View the detail for the new secret.

```
kubectl get secrets
kubectl get secret my-secret
kubectl get secret my-secret -o yaml
```

Next, create a directory with some configuration files.

```
mkdir secrets
```

```
cat > secrets/safe.properties <<EOF
secret.one=foo
secret.two=bar
secret.three=baz
EOF
```

```
cat > secrets/secure.properties <<EOF
password.one=hello
password.two=world
EOF
```

```
ls -la secrets
```

Create a Secret from the directory contents and view the detail.

```
kubectl create secret generic dir-secret --from-file secrets
kubectl describe secret dir-secret
kubectl get secret dir-secret -o yaml
```

Finally, create a Secret from a single file and view the detail.

```
kubectl create secret generic file-secret --from-file secrets/secure.properties
kubectl describe secret file-secret
kubectl get secret file-secret -o yaml
```

Using Secrets

Consume a Secret from environment variables. Define a manifest to specify individual keys.

```
cat > secrets-env.yaml <<EOF
apiVersion: v1
kind: Pod
metadata:
  name: secrets-env
spec:
  containers:
    - name: my-container
      image: busybox
      command: [ "/bin/sh", "-c", "env" ]
      env:
        - name: SECRET_PASSWORD
          valueFrom:
            secretKeyRef:
              name: my-secret
              key: password
        - name: SECRET_APIKEY
          valueFrom:
            secretKeyRef:
              name: my-secret
              key: apikey
  restartPolicy: Never
EOF
```

Create the Pod and view the result.

```
kubectl create -f secrets-env.yaml
kubectl get pods --show-all
kubectl logs secrets-env
```

Use a Secret for command-line arguments. Define a manifest to specify the command-line arguments.

```
cat > secrets-cmd.yaml <<EOF
apiVersion: v1
kind: Pod
metadata:
  name: secrets-cmd
spec:
  containers:
    - name: my-container
      image: busybox
      command: [ "/bin/sh", "-c", "echo PASSWORD: \$(SECRET_PASSWORD) APIKEY: \$(SECRET_APIKEY)" ]
      env:
        - name: SECRET_PASSWORD
          valueFrom:
            secretKeyRef:
              name: my-secret
              key: password
        - name: SECRET_APIKEY
          valueFrom:
            secretKeyRef:
              name: my-secret
              key: apikey
  restartPolicy: Never
EOF
```

Create the Pod and view the result.

```
kubectl create -f secrets-cmd.yaml
kubectl get pods --show-all
kubectl logs secrets-cmd
```

Clean Up

```
kubectl delete pod secrets-cmd
kubectl delete pod secrets-env
```