# Using ConfigMaps & Secrets in Kubernetes

**Janakiram MSV**
Analyst | Advisor | Architect
Janakiram & Associates
Google Developer Expert

@janakiramm

# Objectives

- What are ConfigMaps
- When to use ConfigMaps
- Use cases for ConfigMap
- Overview of Secrets
- Using Secrets
- Key Takeaways

# Configuring Containerized Applications

- Applications expect configuration from
  - Configuration files
  - Command line arguments
  - Environment variables
- Configuration is always decoupled from applications
  - INI
  - XML
  - JSON
  - Custom Format
- Container Images shouldn’t hold application configuration
  - Essential for keeping containerized applications portable

# Whats are ConfigMaps?

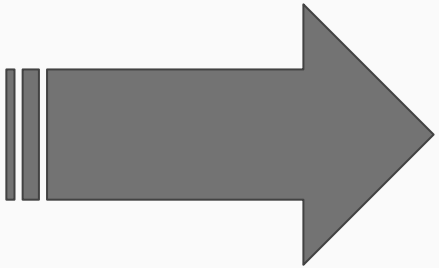
- Kubernetes objects for injecting containers with configuration data
- ConfigMaps keep containers agnostic of Kubernetes
- They can be used to store fine-grained or coarse-grained configuration
  - Individual properties
  - Entire configuration file
  - JSON files
- ConfigMaps hold configuration in Key-Value pairs accessible to Pods
- Similar to */etc* directory and files in Linux OS

# Overview of ConfigMap

```yaml
kind: ConfigMap
apiVersion: v1
metadata:
  creationTimestamp: 2017-03-28T19:14:38Z
  name: example-config
  namespace: default
data:
  example.property.1: hello
  example.property.2: world
  example.property.file: |-
    property.1=value-1
    property.2=value-2
    property.3=value-3
```

# Accessing ConfigMaps from Pods

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: special-config
  namespace: default
data:
  special.how: very
  special.type: charm
```



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: dapi-test-pod
spec:
  containers:
    - name: test-container
      image: gcr.io/google_containers/busybox
      command: [ "/bin/sh", "-c", "env" ]
      env:
        - name: SPECIAL_LEVEL_KEY
          valueFrom:
            configMapKeyRef:
              name: special-config
              key: special.how
        - name: SPECIAL_TYPE_KEY
          valueFrom:
            configMapKeyRef:
              name: special-config
              key: special.type
  restartPolicy: Never
```

# Accessing ConfigMaps from Pods

- Configuration data can be consumed in pods in a variety of ways
- ConfigMap can be used to:
  1. **Populate the value of environment variables.**
  2. **Set command-line arguments in a container.**
  3. **Populate configuration files in a volume.**
- Users and system components may store configuration data in a ConfigMap

# Demo

## Using ConfigMaps

# Using Secrets

- Secret is an object that contains a small amount of sensitive data such as a password, a token, or a key
- Secrets reduce the risk of exposing sensitive data to unwanted entities
- Like ConfigMaps, Secrets are Kubernetes API objects created outside of Pods
- Secrets belong a specific Kubernetes Namespace
- The size of each Secret cannot exceed 1MB

# UsingSecrets

- Secrets are registered with Kubernetes Master
- Secrets can be mounted as Volumes or exposed as environment variables
- Secret is only sent to the Node hosting the Pod that requires access
- Each Secret is stored in tempfs volumes that restrict access to the rest of the applications in the Node
- Communication between the Kubernetes API Server and Node is secured through SSL/TLS

# Demo

## Using Secrets

# Key Things to Remember

- Secrets feature is not entirely foolproof
- API Server stores Secrets in plain text
- During replication across etcd clusters, Secrets are sent in plain text
- Secret definitions may still get exposed to outside world

# Upcoming Webinar

## Scaling and Managing Deployments

## 9AM PST, Tuesday, April 18th

Deployments in Kubernetes bring PaaS-like capabilities to the platform. You easily can pause, resume, update, and resume deployments. They enable rolling upgrades, patching, and rollback of versions. This session highlights the benefits of using deployments in Kubernetes.

- Using Deployments
- Performing Rolling Upgrades and Rollbacks
- Pausing & Resuming Deployments

Register at **kube.live** for the upcoming webinars

# Thank You!

Send your Feedback / Questions / Comments
**jani@janakiram.com**