KUBERNETES – INJECTING DATA

Github repo..

https://github.com/vsaini44/KubernetesRepo.git

Configuring containerized Application?

App in the containers can be customized using following methodologies :

- **Passing command-line arguments to containers.**
- **Setting custom environment variables for each container.**

Command line Argument in Docker

In a Dockerfile, two instructions define the two parts:

- **ENTRYPOINT defines the executable invoked when the container is started.**
- **CMD specifies the arguments that get passed to the ENTRYPOINT.**

Examples:

```
$ docker run <image>

$ docker run <image> <arguments>
```

Arguments in Kubernetes

In Kubernetes, when specifying a container, you can choose to override both ENTRY- POINT and CMD .

To do that, you set the properties command and args in the container specification, as shown in the following listing.

```
kind: Pod
spec:
  containers:
    - image: some/image
      command: ["/bin/command"]
      args: ["arg1", "arg2", "arg3"]
```

Environment Variables

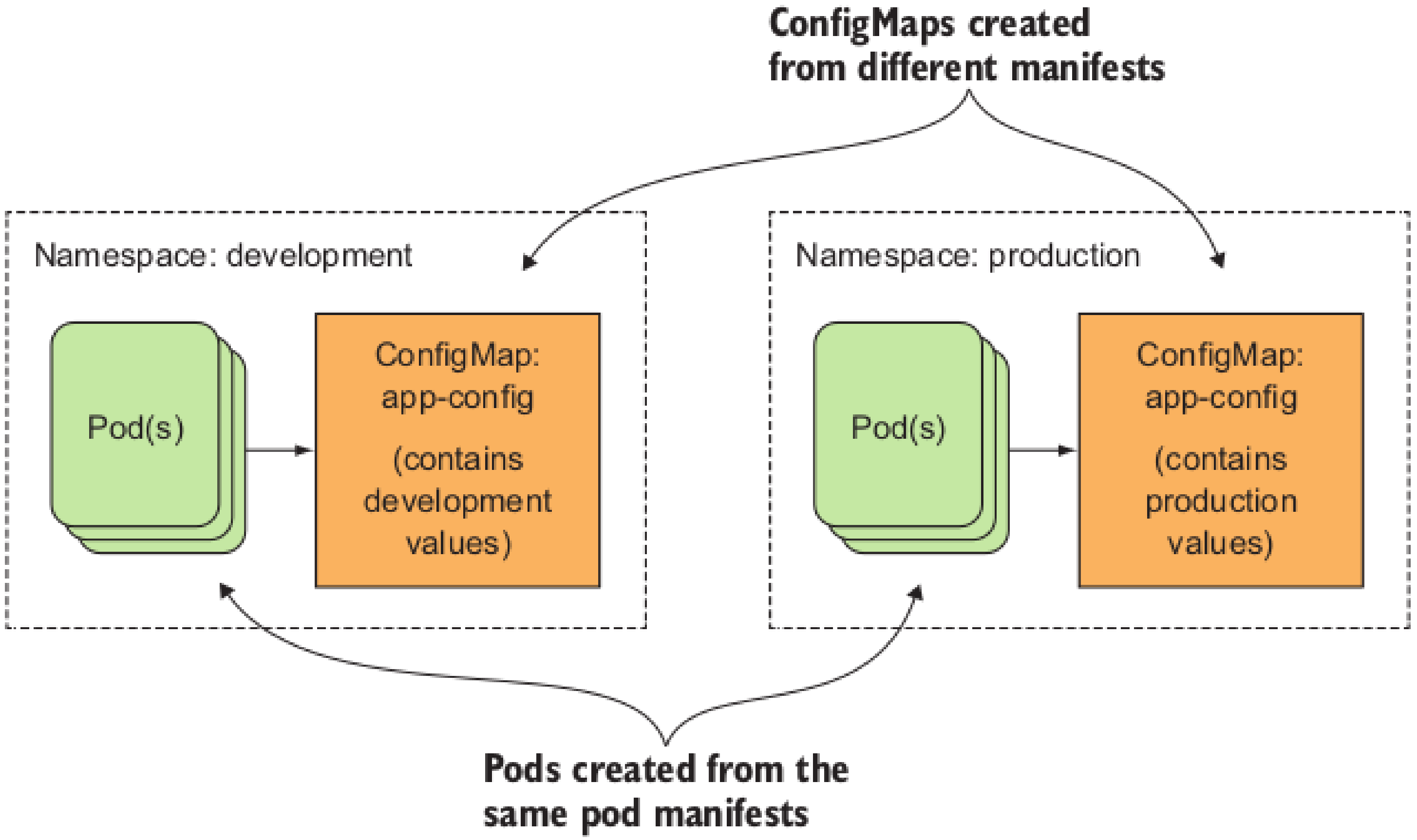
Containerized applications often use environment variables as a source of configuration options. Kubernetes allows you to specify a custom list of environment variables for each container of a pod.

```
kind: Pod
spec:
   containers:
     - image: someimage
       env:
         - name: INTERVAL
           value: "30"
           name: html-generator
```

ConfigMaps

Kubernetes allows separating configuration options into a separate object called a ConfigMap, which is a map containing key/value pairs with the values ranging from short literals to full config files.

ConfigMaps


ConfigMaps created from different manifests
Namespace: development
Pod(s)
ConfigMap: app-config
(contains development values)
Namespace: production
Pod(s)
ConfigMap: app-config
(contains production values)
Pods created from the same pod manifests

Secret

Secrets are much like ConfigMaps—they’re also maps that hold key-value pairs. They can be used the same way as a ConfigMap.

Kubernetes helps keep your Secrets safe by making sure each Secret is only distributed to the nodes that run the pods that need access to the Secret.

Also, on the nodes themselves, Secrets are always stored in memory and never written to physical storage.

Types of Secret ?

Secrets can be created as following types

- **TLS**
- **Generic**
- **Docker-Registry**