

Local Kubernetes Development

With Minikube, Skaffold and KPT

Fredrik Holmqvist 2024-02-15

Some Facts about Me

Name: Fredrik Holmqvist

- likes distributed systems, containers, VM's and Go
- Works at evroc, building a new European Cloud
<https://evroc.com/>
- Worked on collecting terabytes and terabytes of logs at LogicMonitor
<https://www.logicmonitor.com/>
- Kernel Developer for the open source Haiku Operating System
<https://www.haiku-os.org/>

Disclaimers



Disclaimers:

- This presentation is done by me, NOT related to my work at evroc
- I am not affiliated with any of these tools
- I just like them

Agenda:

- Minikube
 - intro, config and commands
 - hands on
- Skaffold
 - intro
 - small go app with Kubernetes YAML
 - skaffold init, yaml,builders and skaffold dev
- Google Cloud Code for your IDE
 - config, run and modify cycle
- Developing the small app
 - adding persistence
 - debugging
- KPT
 - intro
 - example
- Tips and Tricks
 - multiple skaffold files
- Questions and Answers

Minikube

“minikube quickly sets up a local Kubernetes cluster on macOS, Linux, and Windows”

Get Started:

<https://minikube.sigs.k8s.io/docs/>

After installation, before starting, configure Minikube correctly

Minikube Configuration

Choose Driver for Minikube: <https://minikube.sigs.k8s.io/docs/drivers/>

- > minikube config set driver hyperkit # (Works for intel Macs)
- > minikube config set driver docker # (General, most likely default)
- > minikube config set driver qemu # (Another good general option)

Configure Memory, Disk and CPU

- > minikube config set disk-size 10GB
- > minikube config set memory 4096
- > minikube config set cpus 4
- > minikube config view

Minikube Basic Commands

- > `minikube start` # Creates a new cluster or starts a stopped one
- > `minikube stop` # Stops, but does not remove a running cluster
- > `minikube delete` # Stops and deletes a cluster
- > `minikube pause` # Pauses Kubernetes containers
- > `minikube unpause` # Resumes paused Kubernetes containers
- > `minikube status` # Check status of cluster

Minikube Some Additional Commands

- > `minikube image ls --format=table` # List fetched images with size 🔥
- > `minikube ssh` # SSH into the “machine” for Kubernetes
- > `minikube update-context` # Update your kubeconfig
- > `minikube service <k8s service>` # Access info for a service
- > `minikube cp` # Copy files to and from minikube
- > `minikube ip` # Retrieves the IP address of the specified node
- > `minikube docker-env` # Use the docker in minikube

Skafold

“Skafold handles the workflow for building, pushing and deploying your application, allowing you to focus on what matters most: **writing code**”

Get Started:

<https://skafold.dev/>

Detects and uses some local clusters by default:

<https://skafold.dev/docs/environment/local-cluster/>

Our Go App and skaffold init

Easy start:

> skaffold init # (In project directory)

- Detects app in Go, Java or docker-based and its kubernetes files
- Generates a skaffold.yaml file

Lets try skaffold init on a small Go App, that counts each time a path is accessed:

./cmd/main.go # Go code

./k8s/app.yaml # Kubernetes deployment yaml

Scaffold

It generates a scaffold.yaml file:

metadata:

name: app # The name of this scaffold file

build: # How to build this app

artifacts:

- image: app-image-name # name of the image to build, notice no tag

buildpacks: # Builder for image, this image is big...

builder: gcr.io/buildpacks/builder:v1

Scaffold

Lets use another builder (<https://scaffold.dev/docs/builders/builder-types/>) instead:

metadata:

name: app # The name of this scaffold file

build: # How to build this app

artifacts:

- image: app-image-name # name of the image to build, notice no tag

ko: # Builder for go apps, no need for Dockerfile

main: ./cmd # Since our main is in a different directory

Skaffold

Demo

```
> skaffold dev --port-forward=services
```

Google Cloud Code

“Cloud Code provides a managed experience of using Skaffold in supported IDEs. You can install the Cloud Code extension for **Visual Studio Code** or the plugin for **JetBrains IDEs**.”

Get Started:

<https://cloud.google.com/code>

NOTE: You might need to downgrade version number in skaffold.yaml

Goland Demo

Prerequisites:

- Install plugin (maybe turn off usage tracking?)
- Create a **skaffold.yaml**
- Close and Open Project for plugin to auto detect **skaffold.yaml** file and create run configurations
Or through **Google Cloud Code > Kubernetes > Run..** in **Tools**-menu
- Set **Watch Mode** to **On file save**

Demo: config memu, run and modifying the app

Developing the App

Feature Request:

- **Add persistence**

Lets add a postgres database that stores the counts!

Developing the App

Feature Request Complete:

- Added **postgres kubernetes files**
- Added them in **skaffold.yaml**
- Added **postgres.go** code to application
- Added **postgres connection url** to app.yaml
- Switch from old **count** function to **pgcount** function

Debugging the App

Lets try debugging the app!

Debugging the App

**Oh no! favicon.ico path always gets called..
Someone should really fix that...**

KPT

You can use KPT for transformations of your yaml files

The **The Rationale behind kpt** is a good read: <https://kpt.dev/guides/rationale>

Get Started:

<https://skaffold.dev/docs/renderers/kpt/>

There are many transformations: <https://catalog.kpt.dev/>

NOTE: kpt cli must be installed

KPT

Short Version:

Add the following under **manifests:** in your **skaffold.yaml** and it will set these fields:

transform:

- name: apply-setters

configMap:

- "app-replicas:1"

- "pg-password:c3Ryb25n"

Note: Don't use any spaces after colons in the config strings!!

KPT

Continued:

The fields can be anywhere in the yaml if they are marked:

In app.yaml

```
replicas: 3 # kpt-set: ${app-replicas}
```

In postgres-secret.yaml

```
password: "ZXhhbXBsZQ==" # kpt-set: ${pg-password}
```

Note: This only seems to work for me on skaffold command-line

Tips and Tricks

You can have **several** skaffold.yaml **parts** in skaffold.yaml (use **—** and set different **metadata name**)

You can depend on other skaffold.yaml parts, files or git repos:

requires:

- configs: [metadata_name_in_same_file]
- configs: [metadata_name_in_other_file, metadata_name2_in_other_file]
path: ./skaffold_other.yaml
- configs: [metadata_name_in_repo]

git:

repo: <https://github.com/tqh/skaffold-demo.git>
path: app/skaffold.yaml

Thats it!

Thanks for joining!

fredrik.holmqvist@gmail.com

<https://github.com/tqh/skaffold-demo>

Thats it!

Questions and Answers