ORACLE
Cloud Native Labs

Container Native Development Tools Compared: Draft, Skaffold, and Tilt

Mickey Boxell – Oracle Cloud

#OracleCloudNative
cloudnative.oracle.com

ORACLE Cloud Native Labs

@mickeyboxell


KubeCon
CloudNativeCon
North America 2019
Nov. 18 - 21, 2019
San Diego, CA
kubecon.io

Kubernetes Forums

Sponsorships available: Prospectus

Who am I?

Mickey Boxell

Product Manager, Cloud Advocate, etc.

Oracle Cloud Native Labs

Share best practices and build original solutions and content for developers with a key focus on cloud native/container native, open source, and DevOps

http://cloudnative.oracle.com/

ORACLE Cloud Native Labs

@mickeyboxell

Microservice Environments

- Distributed
- Container-based
- Polyglot
- Scalable
- Ephemeral

ORACLE Cloud Native Labs

Development Workflow

- Step 1: Write code
- Step 2: Build code
- Step 3: Run code
- Step 4: Identify issues and return to Step 1

ORACLE Cloud Native Labs

Container Native Development Workflow

- Step 1: Write code
- Step 2: Build code

Step 2.1: Build a container image

Step 2.2: Push the image to a registry

- Step 3: ~~Run code~~ Deploy to Kubernetes cluster
- Step 4: Identify issues and return to Step 1

ORACLE Cloud Native Labs

Traditional Deployment: Helidon/Java

```
$ mvn archetype:generate -DinteractiveMode=false \
        -DarchetypeGroupId=io.helidon.archetypes \
        -DarchetypeArtifactId=helidon-quickstart-se \
        -DarchetypeVersion=1.1.1 \
        -DgroupId=io.helidon.examples \
        -DartifactId=helidon-quickstart-se \
```

ORACLE Cloud Native Labs

Traditional Deployment: Helidon/Java

```
$ cd helidon-quickstart-se

$ mvn package

$ java -jar target/helidon-quickstart-se.jar
```

ORACLE Cloud Native Labs

Container Native Deployment: Helidon/Java

```
$ docker build -t helidon-quickstart-se .

$ docker run --rm -p 8080:8080 helidon-quickstart-se:latest
```

ORACLE Cloud Native Labs

Local Kubernetes Cluster Deployment: Helidon/Java

$ kubectl apply -f app.yaml

ORACLE Cloud Native Labs

Remote Kubernetes Cluster Deployment: Helidon/Java

```
$ docker tag \ helidon-quickstart-se:latest \ <region-code>.ocir.io/<tenancy-name>/<repo-name>/<image-name>:<tag>
$ docker push \ <region-code>.ocir.io/<tenancy-name>/<repo-name>/<image-name>:<tag>
$ kubectl apply -f app.yaml*
```

* modified with a container image matching the registry

ORACLE Cloud Native Labs

@mickeyboxell

The Whole Flow

Step 1: Write code

Step 2: Build code AND build the image AND push the image to a registry

```
$ mvn package
$ docker build -t helidon-quickstart-se .
$ docker tag \ helidon-quickstart-se:latest \ <region-code>.ocir.io/<tenancy-name>/<repo-name>/<image-name>:<tag>
$ docker push \ <region-code>.ocir.io/<tenancy-name>/<repo-name>/<image-name>:<tag>
```

Step 3: Deploy to Kubernetes Cluster

```
$ kubectl apply -f app.yaml
```

ORACLE Cloud Native Labs

That seems like a lot of typing

Of the same set of commands

ORACLE Cloud Native Labs

Over and over

Why Did I Care?

- Simple code changes took too much time & too many keystrokes
- e.g. Was my endpoint zipkin.monitoring:9411 or 10.0.32.4:9411/zipkin or something else?
- Each change required me to: build code, build image, tag image, push image, apply manifest

ORACLE Cloud Native Labs

Why Not Just Use CI/CD?

- You need tools that operate at a high speed
- You can't take a CI/CD system that takes minutes and make it take seconds or milliseconds
- Every second matters to developer productivity
- This is a different problem from "how do you ship?"

ORACLE Cloud Native Labs

When Does This Take Place?

- The inner loop of the container native development workflow: the period of time during which you are writing code, but have not yet pushed it to a version control system
- More simply: “when you’re iterating on code pre-commit”

ORACLE Cloud Native Labs

“What you do a few times a day is different from what you do hundreds of times a day“ – Dan Bentley, Tilt

ORACLE Cloud Native Labs

Why Deploy To A Cluster?

- Run diagnostic tools – logging, tracing, etc.
- Run integration and dependency tests

ORACLE Cloud Native Labs

Why Deploy To A Remote Cluster?

- Resource exhaustion
- Match test environment to production environment
- Compliance – not everyone has the option of a local cluster

ORACLE Cloud Native Labs

There’s even more going on under the covers

Dockerfile

```
# 1st stage, build the app
FROM maven:3.5.4-jdk-9 as build
WORKDIR /helidon
# Create a first layer to cache the "Maven World" in the local repository. Incremental docker builds will always resume after that, unless you update the pom
ADD pom.xml .
RUN mvn package –DskipTests
# Do the Maven build! Incremental docker builds will resume here when you change sources
ADD src src
RUN mvn package –DskipTests
RUN echo "done!”
```

```
# 2nd stage, build the runtime image
FROM openjdk:8-jre-slimWORKDIR /helidon
# Copy the binary built in the 1st stage
COPY --from=build /helidon/target/helidon-quickstart-se.jar ./
COPY --from=build /helidon/target/libs ./libsCMD ["java", "-jar", "helidon-quickstart-se.jar"]
```

ORACLE Cloud Native Labs

So why not take a similar approach to push and deploy?

Build, Push, Deploy Tools

ORACLE Cloud Native Labs

@mickeyboxell

What Are These Tools?

- Draft by Microsoft Azure

- Skaffold by Google

- Tilt by Windmill Engineering

ORACLE Cloud Native Labs

@mickeyboxell

What Do These Tools Do?

- Build code
- Build an image of your project
- Push the image to a registry service of your choice
- Deploy the image onto a Kubernetes cluster
- Save you time and clicks!
- And they are all open source

ORACLE Cloud Native Labs

Pre-Requisites

- Docker
- Kubernetes cluster
  - Local: Docker For Desktop/Minikube/etc.
  - Remote: Oracle Container Engine for Kubernetes (OKE)
- Kubectl
- An image registry service
  - Oracle Cloud Infrastructure Registry (OCIR)

ORACLE Cloud Native Labs

Sample Application

helidon.io

- Helidon Framework - Java libraries for writing microservices
- Quickstart-SE sample application/archetype
  - And a colorful front end ☺

ORACLE Cloud Native Labs

Draft

ORACLE Cloud Native Labs

@mickeyboxell

Draft

- Low barrier to entry: Draft packs
  - draft create: boilerplate artifacts to run existing apps in K8s
  - Dockerfile, Helm charts

ORACLE Cloud Native Labs

@mickeyboxell

Using Draft

Pre-Reqs: Docker, Kubectl, Helm

- draft init – install packs/plugins and configure $DRAFT_HOME
- draft create – create boilerplate based on application language
- draft config set registry phx.ocir.io/oracle-cloudnative/draft - creates .draft directory and config.toml
- docker login
- draft up + draft delete - make registry public or use imagepullsecrets

ORACLE Cloud Native Labs

@mickeyboxell

Using Draft

- Port forward: draft connect
- Logs: draft logs

ORACLE Cloud Native Labs

@mickeyboxell

Draft

- Boilerplate is helpful to get started
- No watch/continuous deployment feature
- Helm can be overly-complicated and is the only deployment option
- VS Code integration
- Not actively being worked on ☹

ORACLE Cloud Native Labs

@mickeyboxell

Skaffold

ORACLE Cloud Native Labs

@mickeyboxell

Skaffold

SKAFFOLD

- Flexible
- Many build options (Dockerfile locally, Dockerfile in-cluster with Kaniko, Dockerfile on the cloud, Jib Maven/Gradle locally, etc.)
- Many deploy options (kubectl, Helm, Kustomize)
- Many image tag policies

ORACLE Cloud Native Labs

Using Skaffold

Pre-Reqs: Docker, Kubectl

- vi skaffold.yaml – specifies workflow steps
- skaffold config set default-repo phx.ocir.io/oracle-cloudnative/skaffold – creates .skaffold file
- docker login
- skaffold run + skaffold delete or skaffold dev - make registry public or use imagepullsecrets + change image spec in app.yaml

ORACLE Cloud Native Labs

@mickeyboxell

Using Skaffold

SKAFFOLD

- Logs: skaffold run –tail
- Port-forward: automatic based on pod spec configuration or with --port-forward flag

ORACLE Cloud Native Labs

Skaffold

SKAFFOLD

- Profiles feature
  - A set of settings stored in skaffold.yaml that overrides the build, test, and deploy sections of your current configuration
  - skaffold run -p [PROFILE]
- Deploy multiple microservices at once
- File sync – copy changed files to a container to avoid a full rebuild
- Deploy once with skaffold run or continuously with skaffold dev

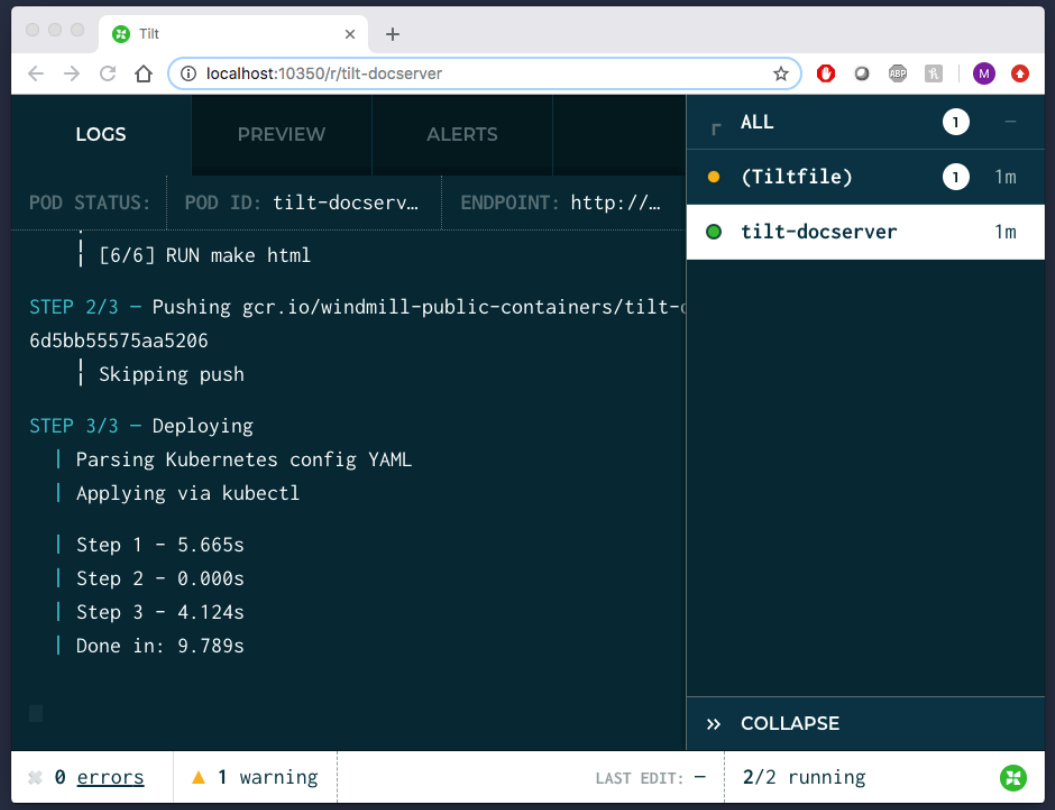
ORACLE Cloud Native Labs

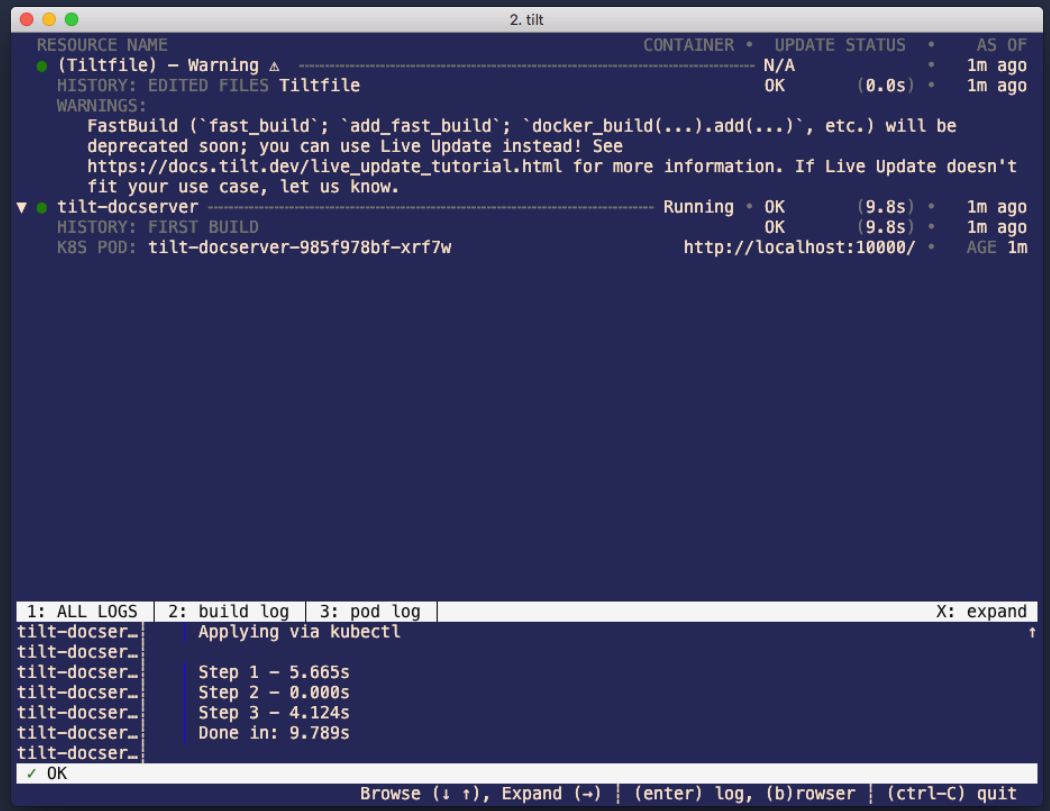
@mickeyboxell

Tilt

Tilt

- Heads up display and browser UI





ORACLE Cloud Native Labs

@mickeyboxell

Using Tilt

Pre-Reqs: Docker, Kubectl

- vi Tiltfile – specifies workflow steps
- Set registry path in the Tiltfile or tilt_option.json
- docker login
- tilt up + tilt down - make registry public or use imagepullsecrets + change image spec in app.yaml

ORACLE Cloud Native Labs

Using Tilt

- B opens a port forward based on Tiltfile resource URL
  - Browser UI includes resource preview page
- Logs available on the UI – X to expand logs

Tilt

- Heads up display and browser UI
- Python Skylark config file – concise and extensible
- LiveUpdate: update a running container in place
  - Instead of building a new image and redeploying from scratch
- Deploys multiple microservices - sample application “servantes”
- No single deploy option
- Dedicated, focused development team

ORACLE Cloud Native Labs

@mickeyboxell

Key Takeaways

- Developer productivity - automate away countless manual steps
- Client-side tools – aside from Helm/Tiller
- These tools can deploy to both local and remote clusters
  - The registry step can be bypassed for local clusters
- Useful as a step before pushing to source control and/or CI
  - They are meant to complement, not replace a CI/CD system

ORACLE Cloud Native Labs

Differentiators

- Getting started boilerplate

- Flexibility

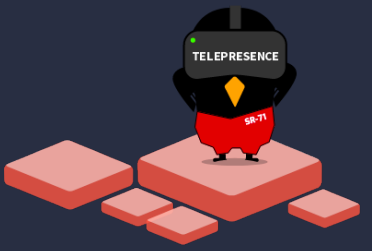
- Heads up display

ORACLE Cloud Native Labs

@mickeyboxell

Additional Development Tools

- Visual Studio Code – Kubernetes Tools Extension:
  - Visually interact with your cluster, run commands
  - Simplify yaml creation



- Telepresence:
  - Connect a locally running service to a remote cluster

- Code Server - in-cluster IDE – I don't think this is a great idea ☺
- Ksync – file sync between local directory and a running container

KSYNC

ORACLE Cloud Native Labs

Stay Connected

Medium: https://medium.com/oracledevs

Twitter: @mickeyboxell

Linkedin: https://www.linkedin.com/in/mickeyboxell/

Try Oracle Cloud: https://cloud.oracle.com/tryit

#OracleCloudNative
cloudnative.oracle.com

ORACLE Cloud Native Labs