Rust and Skaffold for Iterative Development on Kubernetes
宋子豪 Gilbert Song

Staff Software Engineer

- Apache Mesos PMC/Committer
- Leading Containerization at Mesosphere
- Twitter: @gilbert_songs
Gastón Kleiman

Staff Software Engineer

- Apache Mesos PMC/Committer
- Resource Management Tech Lead at Mesosphere
- Previously at AWS OpsWorks
- Twitter: @kleimang
Outline

Modern development techniques

Iterative Development

CI/CD

Immutable infrastructure + Containers

Challenges

Tools

Rust, Kubernetes, Skaffold

Demo
Continuous/Iterative Development

- Design, develop, and test in repeated cycles
- Testing a new iteration must be quick & cheap
Continuous Integration/Continuous Deployment

- Automated testing as well as deployment of a web application
- Containers make it possible to replicate the same environment during development and in production
Pets -> Cattle -> Immutable Infrastructure
Challenges

- Building, pushing, and deploying isn’t easy
- Giant binaries/containers
- Applications need to be safe (concurrency + memory safety)
Rust

- Created at Mozilla as a “safe, concurrent, practical language”
- Low-level (static) language, performance similar to C++
- Strong emphasis on safety, control of the memory layout, concurrency
- Great & friendly community
Challenges with Rust

• Relatively young language
• Brings extra safety, but you have to pay some upfront price
• Many engineers are new to it; successfully integrating it into your stack can cost significant engineering time
Kubernetes

- Orchestrates containers across many different machines... or on your local dev machine
- Enables you to use immutable infrastructure
- Useful abstractions/tools to easily deploy or scale services
- DC/OS Multi-Kubernetes-Engine
Skaffold

- CLI tool that facilitates iterative development for Kubernetes applications
- Handles the workflow for **building, pushing, and deploying** an application
- Can be used to iterate on an application source code locally then deploy to local or remote Kubernetes clusters
Skaffold

- Build
- Push
- Deploy
Skaffold (Local)

Build → Push → Deploy

Local Docker Daemon → None → Minikube
Skaffold (Remote)

- Build
  - Google Container Builder
- Push
  - Docker Registry on DC/OS
- Deploy
  - MKE on DC/OS
Using Skaffold

No need to have each new developer spend days setting up a new dev environment

1. Create k8s manifests (pod specs, ingress settings, etc)
2. Tell skaffold via skaffold.yaml how to build and deploy the application
3. Use “skaffold dev” on your local machine to deploy to a minikube cluster
4. Make your CI/CD pipeline run “skaffold run” when new code is checked-in
Demo Flow

Save  Push  Trigger

Skaffold

Build  Push  Deploy
Demo

```
dirty-9bf47e7: digest: sha256:b66fedb5fe6e6c7f63d4cee5de9148767381e26c96422f097f6699819a1346875 size: 3882
Build complete in 3.013927954s
Starting deploy...
Ingress.extensions/rust-web-demo unchanged
service/rust-web-demo unchanged
deployment.apps/rust-web-demo configured
Deploy complete in 1.434400547s
watching for changes...
Port Forwarding rust-web-demo:74d5d4bd8f-52rt5 8000 -> 8000
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] => address: 0.0.0.0
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] => port: 8000
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] => log: normal
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] => workers: 8
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] => secret key: generated
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] => limits: forms = 32K1B
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] => tls: disabled
Mounting '/':
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] Rocket has launched from http://0.0.0.0:8000
GET /:
[rust-web-demo:74d5d4bd8f-52rt5 rust-web-demo] => Outcome: Success
Port Forwarding rust-web-demo:74d5d4bd8f-jrp7k 8000 -> 8000

> EOF
shane@rocketship ~/Code/demo $ curl -w "\n" -H "Host: ${DEMO_DOMAIN}" -H "PUBLIC_NODE_IP" Rocket Webserver!
shane@rocketship ~/Code/demo $```
Future Work

• Make the service highly-available:
  • Use an HA postgres setup
  • Zero-downtime deployments
• Utilize more of the dependencies’ capabilities and increase efficiency:
  • Use connection pools for Diesel
  • Use Rocket managed state
Resources

• Mesosphere Kubernetes Engine (MKE) -
  https://mesosphere.com/product/kubernetes-engine/

• https://github.com/shaneutt/dcos-k8s-rust-skaﬀold-demo
Special Thanks

Shane Utt

Jörg Schad
THANK YOU!

ANY QUESTIONS?

https://mesosphere.com/resources/running-kubernetes-oreilly-ebook/
Make it insanely easy to build and scale world-changing technology