OPTIMIZING MICROSERVICES DEVELOPMENT IN KUBERNETES

Chakradhar Rao Jonagam
Biqmind Head Software Architect, CNCF Ambassador
API World, 28 Oct

biqmind

HELLO

"Chak" Jonagam
Biqmind Head Software Architect

@debianmaster

CAPE

CLOUD NATIVE COMPUTING FOUNDATION AMBASSADOR

- Kubernetes Specialist
- Architect & lead for CAPE, a new K8s Tool (cape.sh)
- Cloud Native Compute Foundation (CNCF) ambassador

TODAY

Our Story/Challenges

- Our developer journey
- Challenges uncovered
- Our idea of a great dev environment

The Solution

- Possible approaches
- Pros and cons with each approach
- Tool universe
- Demo

OUR STORY / CHALLENGES

OUR DEVELOPER JOURNEY

Microservices

LanternEdge Overview

A positioning solution that eliminates inefficiencies of personnel management, increases safety and productivity of workers at hazardous workplaces

- 20+ microservices
- Edge focused application
- Primarily nodejs
- Message Queue

K8s Operators

CAPE overview

A multi-cluster application and data management Kubernetes operator

- Kubernetes Operator
- Golang / React
- Distributed in nature
- Needs to be tested on various versions of k8s
- Mix of monolith and microservices

CAPE

Microservices

OUR DEVELOPER JOURNEY

For Dev

For Operations

Microservices

OUR DEVELOPER JOURNEY

Adopting K8s In lower envs

Clone all 24 repos

Create CAGE specific Deployment config

Use CAGE to run all 24 services with host path mount into container for hot reload

Sync CAGE specific changes to K8s/AKS deployment configs

Sync changes to edge deployment specific configs

cage


Microservices
K8s
for developers is a
work-in-progress

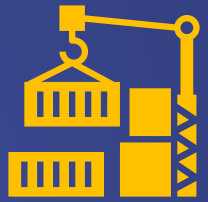
CHALLENGES THAT WE UNCOVERED IN SPRINT RETROS

What our team considers a great dev environment

A GREAT DEV ENVIRONMENT

Smooth set up & onboarding

Easy on laptop



Dependencies (Nodejs/golang) in docker not on laptop

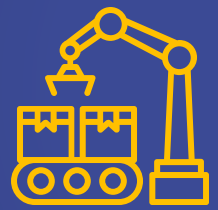
Faster feedback

Ability to debug in a running environment

Does not require user to open multiple terminals / context switching

A GREAT DEV ENVIRONMENT

See all service logs in one place



Production-like environment

Easy branch switching

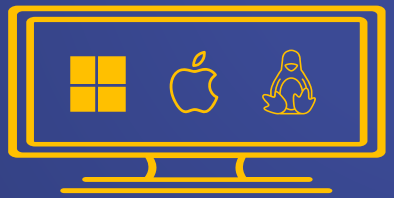
Ability to switch between local and remote clusters transparently

GOOD TO HAVE

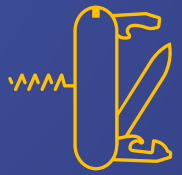
Simple syntax bootstrap services for local development

Non opinionated

Easy to replicate



Works seamlessly on Mac, Windows and Linux environments



Tooling that works with both interpreted and compiled languages

POSSIBLE APPROACHES

Dev Pods

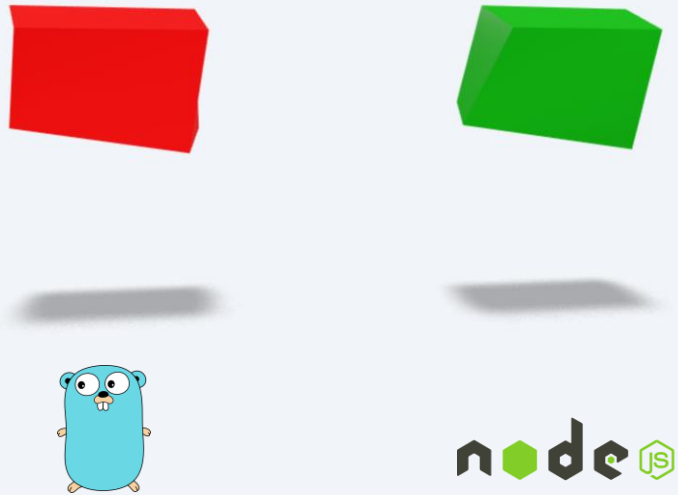
Dev Spaces

Visual Studio Code Remote
- Containers

SAMPLE APP

REACT THREE FIBER

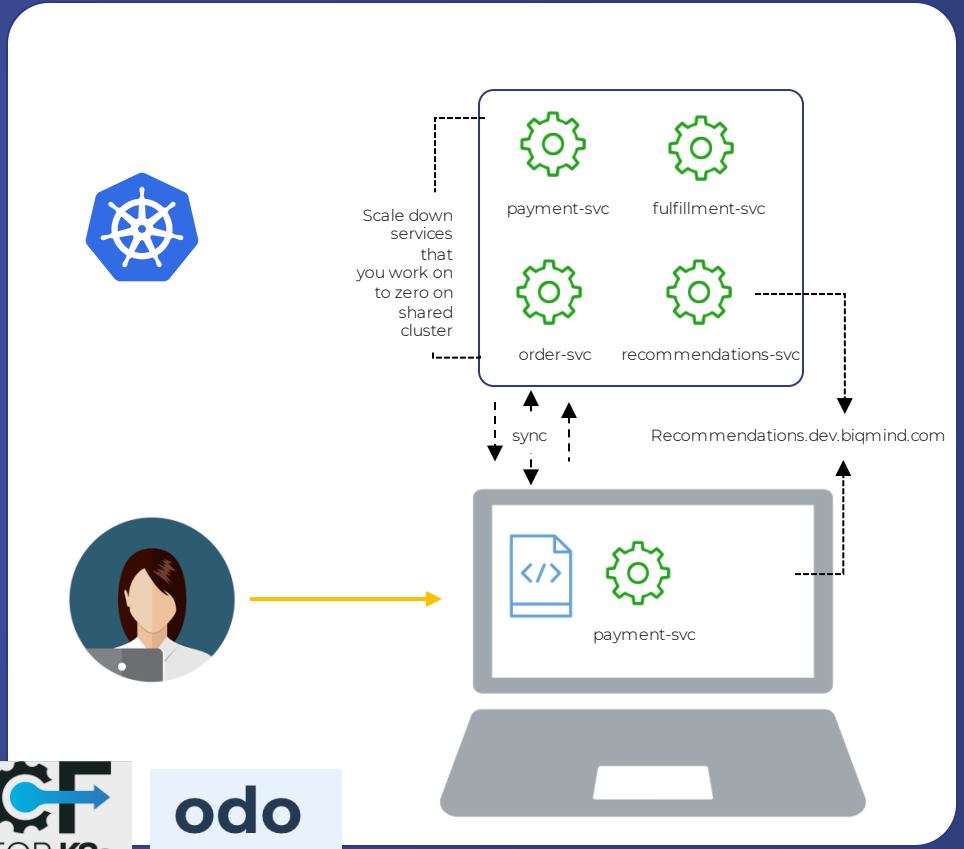
EP. 1



CF PUSH/BUILD PACKS APPROACH

- Run all dependent services on shared K8s cluster
- Use cf push or equivalent to push your changes to remote container
- Abstracts away k8s from developers
- Easy to start with for developers

- Feedback loop is higher
- Mostly suited for single service development at a time
- Promotion of docker images differ from dev to upper envs
- No full control on base image w/o significant effort



CF FOR K8s

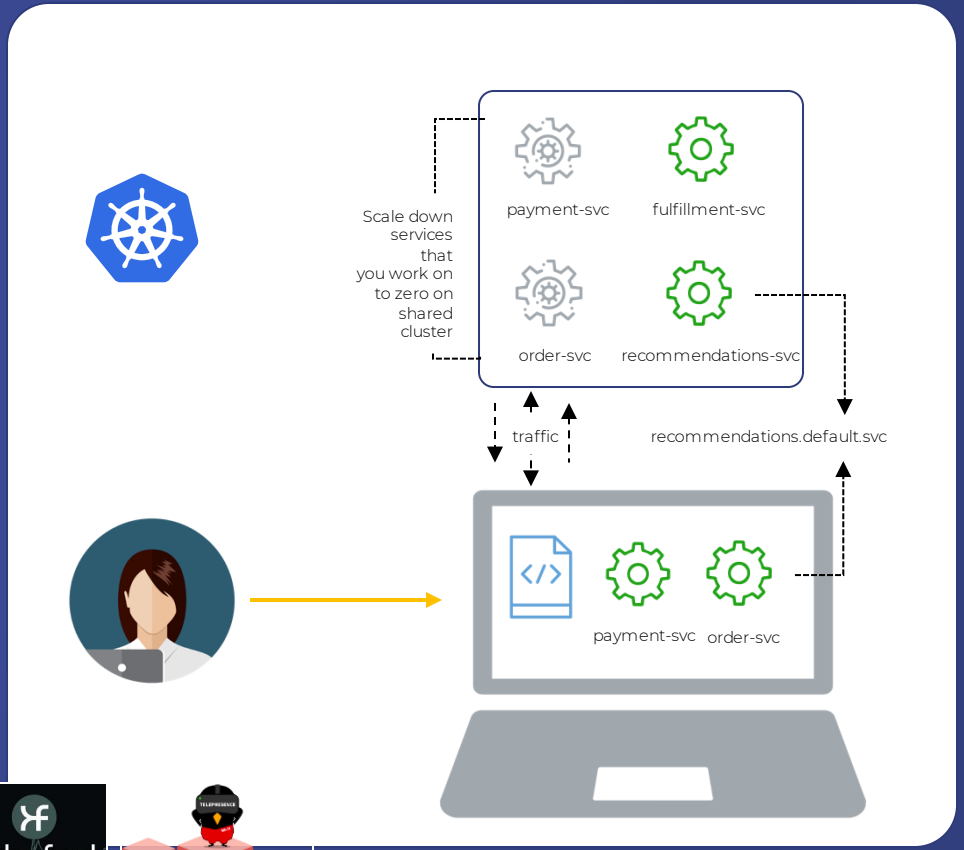
odo

biqmind

PROXY CLUSTER TRAFFIC TO LOCAL

- Run all dependent services on shared K8s cluster
- Run services that you are working on, on your laptop
- Connect dependency services from laptop to shared cluster transparently, no extra config required
- Establish private network between shared namespace and services on laptop using kube-fwd, telepresence
- Scale down the services that you are working on, on the shared cluster to avoid conflicts

- npm link works
- Scaling down services that you are working on is manual for kubefwd but automated for telepresence
- Dependencies on laptop need to be configured
- Most tooling suitable for single service only

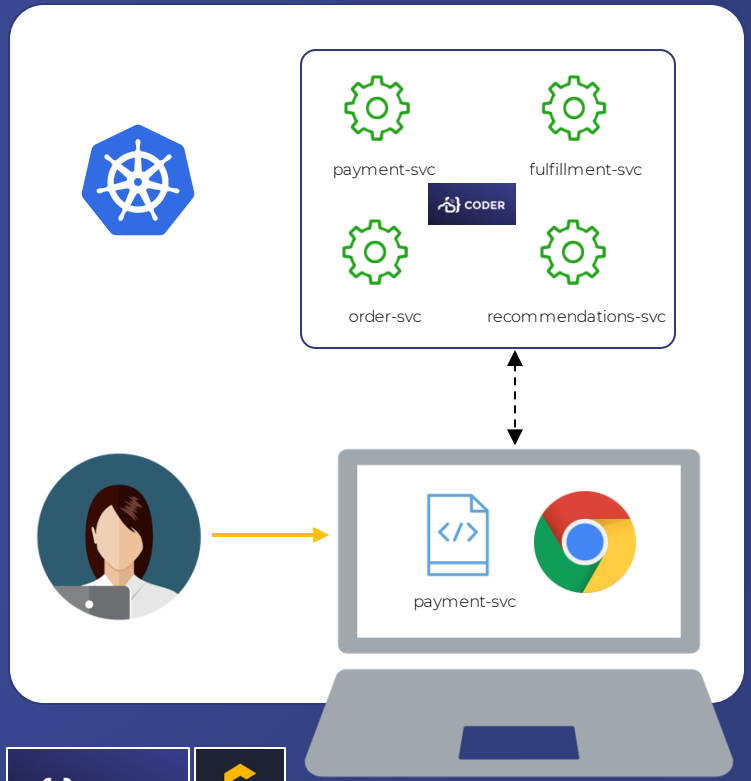


kubefwd

VSCODE WEB ALONGSIDE SERVICES

- Run all services in K8s namespace
- Deploy code-server in same K8s cluster (preferably in same namespace)
- Open browser to access code editor
- Clone repos, start services
- Can embed vscode-web inside desktop app

- Not a native editor experience
- Few additional steps are required to create services, dependencies, scale down conflicting services etc



CODER

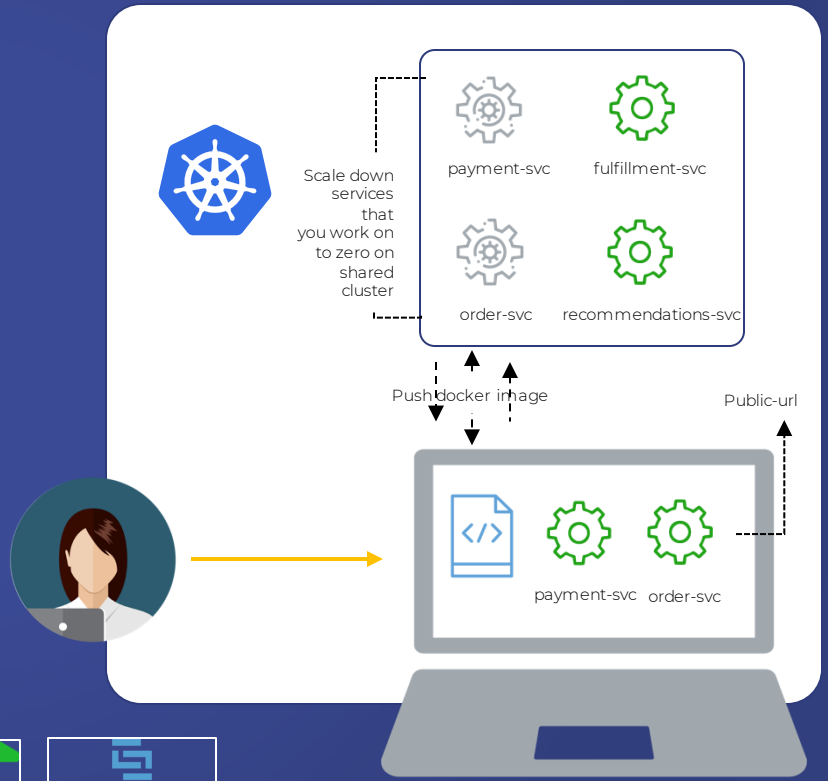
Eclipse Che

biqmind

DOCKER BUILD, PUSH, TAG DEPLOY

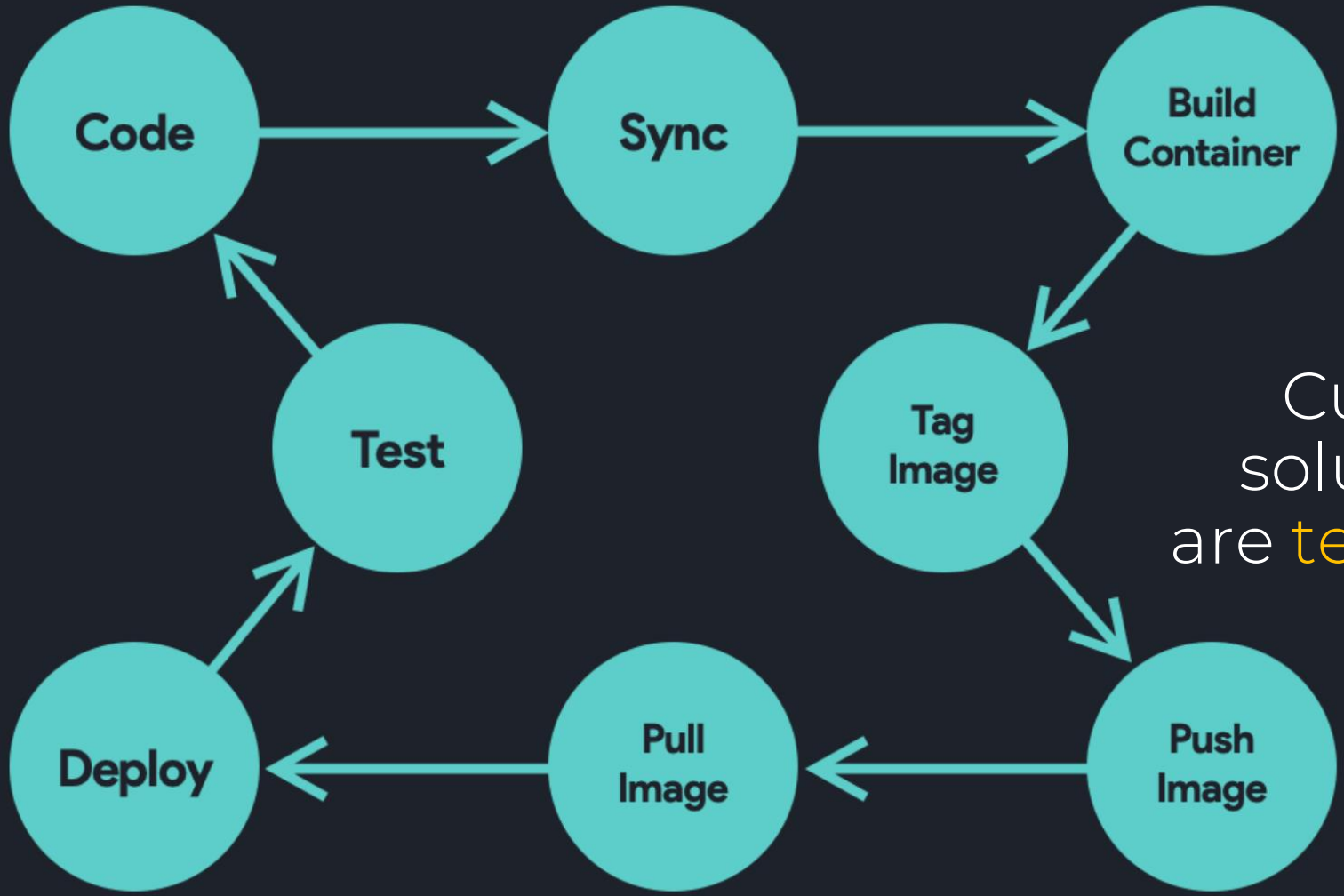
- Run all dependent services on shared K8s cluster
- Run services that you are working on, on your laptop
- Build docker image of your application
- Tag and push image to registry
- Deploy newly built image on the existing deployment

- Long build times.
- Less flexible



TILT

SKAFFOLD



Current solutions are tedious

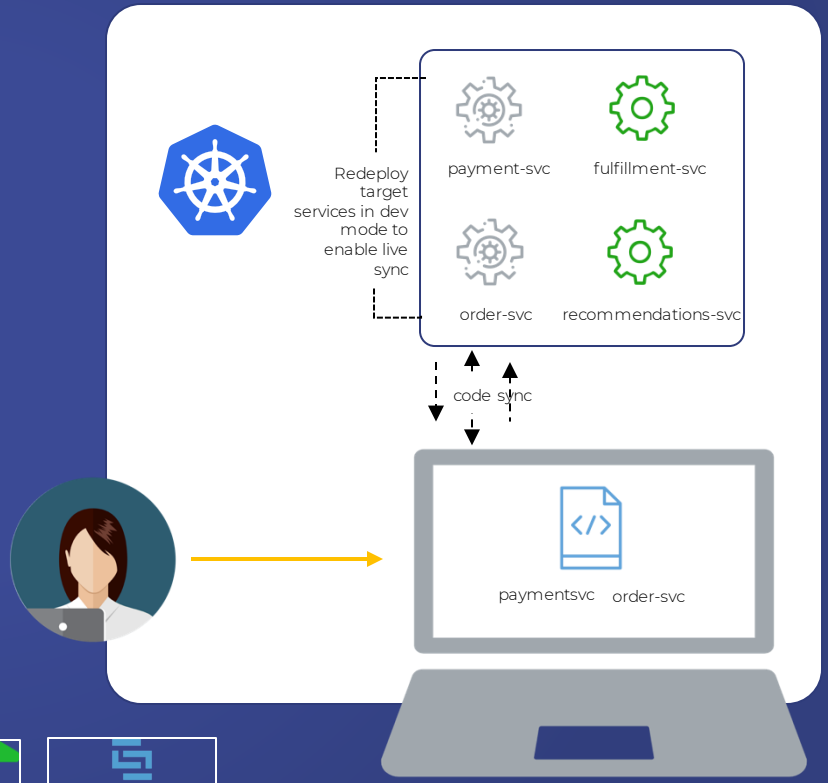
Credits: @pchico83

RECOMMENDED

HOT SYNC TO REMOTE CONTAINER

- Run all dependent services on shared K8s cluster
- Use okteto/tilt to transparently convert services that you want to work on into dev mode
- No private network between laptop and k8s cluster is needed as all the services are running on the same cluster
- All changes in the local environment are synced to remote container

- Not great for sync if the cluster is too far from dev machine
- npm link does not work

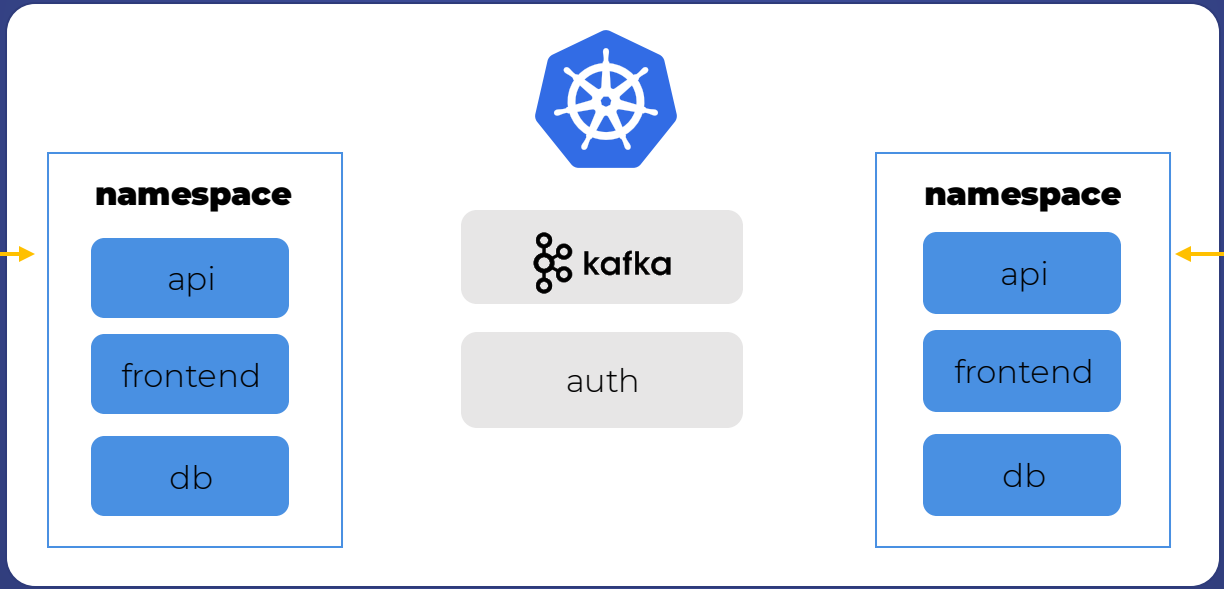


RECOMMENDED

HOT SYNC TO REMOTE CONTAINER

- Each developer works on her/his own namespace (self-service)
- Access to platform services

Developer 1



Developer 2

HIGH LIGHTS

Microservices

okteto

- Simple to start with
- Works great if you have containerized workloads already
- Less opinionated
- Less features
- Suitable for working on single service at a time
- Debug in running environments
- No need for registry

TILT

- Most flexible
- Good speed
- Custom command buttons
- UI for centralized service logs
- Error messages are clear
- *Registry is required*
- *Eats up space on docker and needs frequent cleaning*

SKAFFOLD

- Large community
- Google backed project
- Can work w/ or w/o registry
- Can build either locally or on remote cluster
- Good java support with jib
- *Eats up space on docker and needs frequent cleaning*

FOR OPERATORS DEVELOPMENT AND TESTING

kubebuilder

Metacontroller

OPERATOR FRAMEWORK

Operators Build and Test Tools

https://github.com/debianmaster/actions-k3s

KUTTL

biqmind

VIRTUAL KUBERNETES FOR OPERATORS (K8S ON K8S)

Operators

```
➜ loft cat values.yaml
virtualCluster:
  image: rancher/k3s:v1.18.6-k3s1
  extraArgs:
    - --service-cidr=10.43.0.0/16 # THE CLUSTER SERVICE CIDR HERE
storage:
  size: 5Gi

syncer:
 #extraArgs: ["--disable-sync-resources=ingresses"]
 image: loftsh/virtual-cluster%
```

Input

loft

```
➜ loft helm install c1 virtualcluster --repo https://charts.devspace.sh/ \
  --namespace c1 \
  --values values.yaml \
  --create-namespace \
  --wait
```

Install vk8s

```
➜ loft kubectl exec c1-0 -n c1 -c syncer -- cat /root/.kube/config > c1.yaml
```

Get KC

```
➜ loft kubectl get nodes --kubeconfig=c1.yaml
```

Access vK8s

TESTING OPERATORS

Operators

KUTTL

Input

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: example-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

Test

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: example-deployment
status:
  readyReplicas: 3
```

Run Test

```
kubectl kuttl test --start-kind=true ./tests/e2e/
```

TESTING OPERATORS ON VARIOUS K8S VERSIONS

Operators

```yaml
name: Matrix Testing
on: push
jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        k8s_version: [v0.8.1,v0.9.0,v1.16.9-k3s1,v1.17.4-k3s1,v1.18.2-k3s1]
    steps:
    - uses: debianmaster/actions-k3s@master
      id: k3s
      with:
        version: ${{ matrix.k8s_version }}
    - name: Test on k3s
      run: |
        kubectl get nodes
```

Your logic goes here

https://github.com/debianmaster/actions-k3s

Testing against different k8s versions

CLOSING THOUGHTS

- Pick one of these three tools: skaffold, tilt, okteto
  - Okteto works great for existing containerized workloads as it's less opinionated
  - Skaffold/Tilt for net new projects
- Start Kubernetes first instead of docker-compose to k8s for net new
- Not everything requires an operator: use this approach only for applications that require complex setup and upgrades

CAPE.SH
PROMO CODE:
CAPE5K

Follow us on Twitter @capesuperhero,
DM us your name, address and T-shirt size.
We will send you a free CAPE T-shirt!

Let's connect!

Chakradhar Jonagam
chakradhar.jonagam@biqmind.com
@debianmaster – Twitter
@debianmaster – GitHub

THANK YOU

biqmind