

# The App Developer's Kubernetes Toolbox

Adnan Abdulhusein (@prydonius)  
Bitnami

# \$ whoami

## Adnan Abdulhussein (@prydonius)



- Software Engineer @ VMware
- ❤️ DevOps, Developer Tools, OSS
- SIG-Apps co-chair
- Helm & Charts core maintainer

# Agenda

- Overview of Development Environments
- Development Environments for Kubernetes
- Developer Tools Landscape



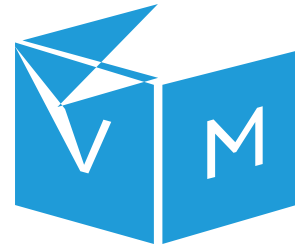
# Development Environments

## Overview

# Development Environments



Local Machine



VMs  
(vagrant)



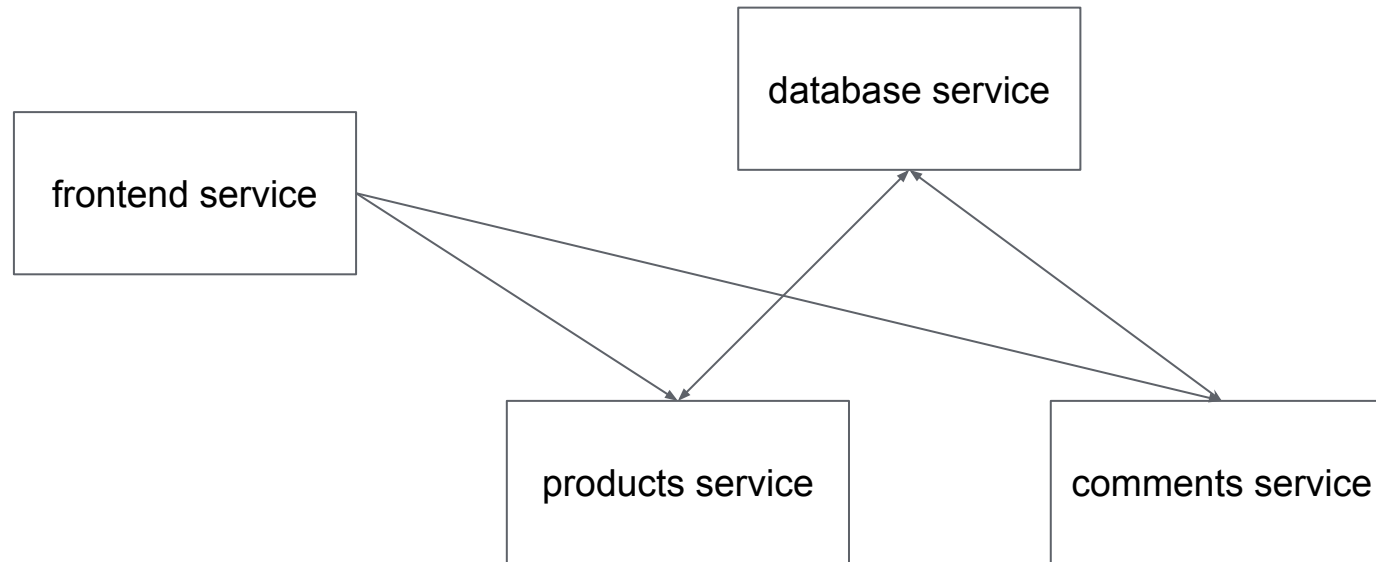
Containers  
(docker)



Goal: get as close to production as possible










# Developing on Kubernetes



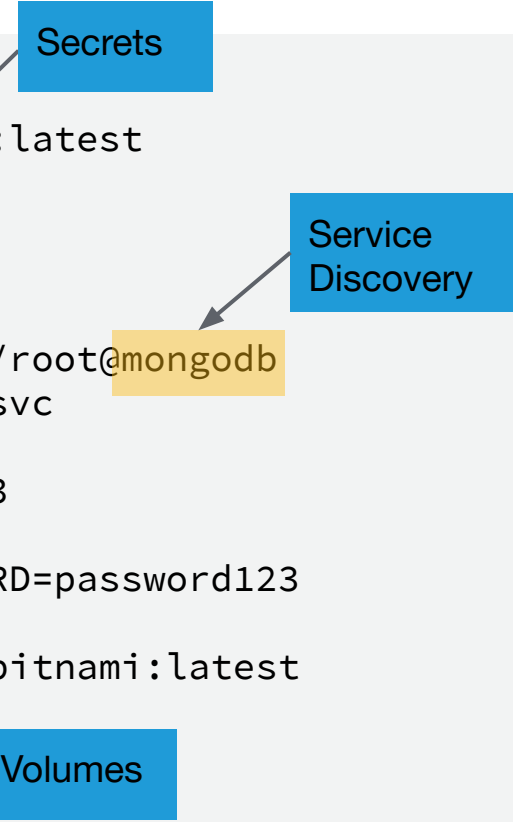
# Developing on Kubernetes...

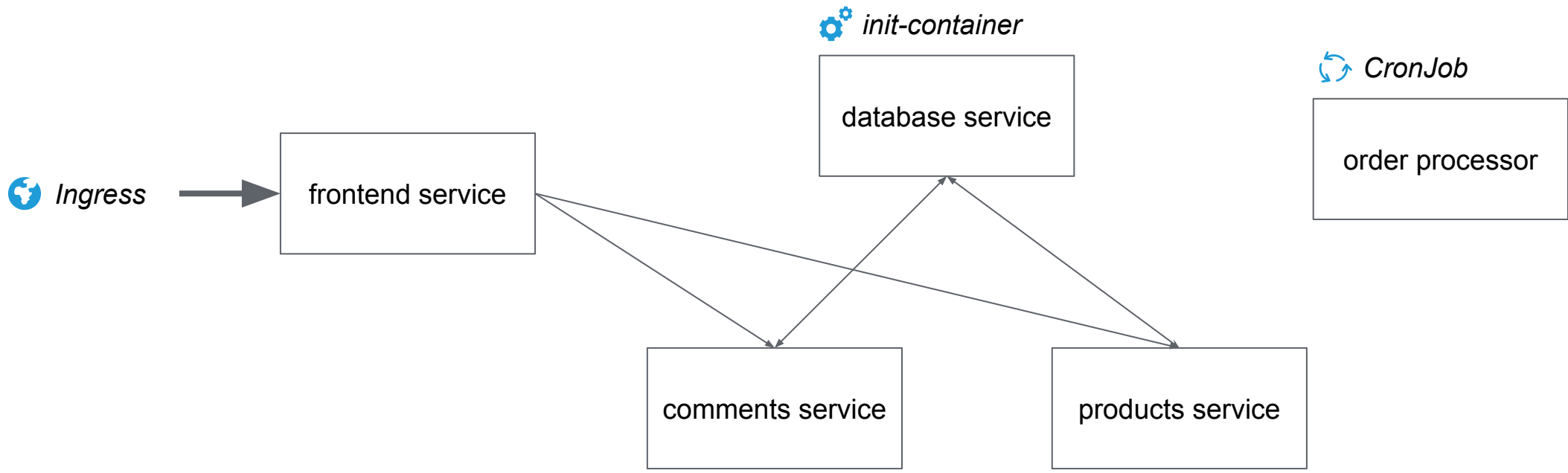
## ...with Docker Compose

```
services:
  ui:
    tty: true
    build: ./docker
    ports:
      - 4200:4200
    volumes:
      - ./app
    command: "ng serve"
  api:
    tty: true
    image: bitnami/monocular-api:latest
    volumes:
      - ./config.yaml:/config/monocular.yaml
    environment:
      - ENVIRONMENT=development
      - MONOCULAR_AUTH_SIGNING_KEY=secret
```



```
ratesvc:
  image: kubeapps/ratesvc:latest
  environment:
    - JWT_KEY=secret
  command:
    - /ratesvc
    - --mongo-url=mongodb://root@mongodb
    - --mongo-database=ratesvc
mongodb:
  image: bitnami/mongodb:3
  environment:
    - MONGODB_ROOT_PASSWORD=password123
auth:
  image: kubeapps/oauth2-bitnami:latest
  ...
volumes:
  monocular-data:
```

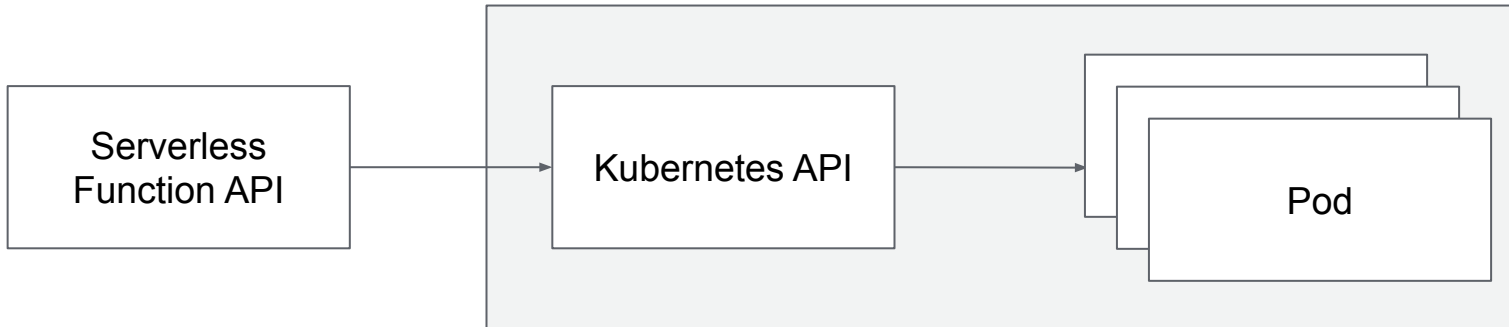






```
import requests
import urlparse

def handler(event, context):
    ticker = event['data']['crypto']
    path = urlparse.urljoin('https://api.coinmarketcap.com/v1/ticker/', ticker)
    return requests.get(path).json()[0]['price_usd']
```



## Developing Kubernetes Extensions

Example: a custom API for managing Serverless Functions on Kubernetes

The application needs to talk to a Kubernetes API server to create, update, scale and delete Pods that run the functions.

A photograph of a person's hands typing on a white keyboard in front of an Apple iMac monitor. The background is a white surface with a repeating geometric pattern of light gray cubes. A blue diagonal shape is in the top right, and a green diagonal shape is in the bottom right.

# Kubernetes Developer Tools

A look at the landscape





# First things first

Getting a Kubernetes environment to work with

# Minikube

```
tstromberg — ~ — fish -i — 72x13
~ minikube start
🤗 minikube v0.34.0 on darwin (amd64)
🔥 Creating virtualbox VM (CPUs=2, Memory=2048MB, Disk=20000MB) ...
📶 "minikube" IP address is 192.168.99.183
🐳 Configuring Docker as the container runtime ...
✨ Preparing Kubernetes environment ...
🚜 Pulling images required by Kubernetes v1.13.3 ...
🚀 Launching Kubernetes v1.13.3 using kubeadm ...
🔑 Configuring cluster permissions ...
🤔 Verifying component health .....
💖 kubectl is now configured to use "minikube"
🏄 Done! Thank you for using minikube!
```

- Single-node Kubernetes environment
- Supports VirtualBox, VMware, Hyperkit and others
- Supports addons: NGINX Ingress, Dashboard, etc.
- Flexible and configurable

# Docker for Desktop



- Simpler to enable
- Reuses existing Docker environment
- Can use local images directly in Kubernetes
- Not available on Linux
- Kubernetes version determined by Docker for Desktop version

# Other Local Options

microk8s - <https://microk8s.io>

- Easy to install on Ubuntu (or other Linux OS with Snap)
- Installs separate binaries for tools (e.g. `microk8s.kubectl`)

Rancher k3s - <https://k3s.io>

- Very lightweight Kubernetes distribution (40mb binary, 512mb RAM)
- Works with Docker Compose for a quick way to setup and teardown clusters



# Remote Options

## Managed Kubernetes Services (Google Kubernetes Engine, Azure Kubernetes Service, etc.)

Google Cloud Platform Launchpad Test 2

### Create a Kubernetes cluster

**Cluster templates**  
Select a template with preconfigured settings, or customize a template to suit your needs

- Clone an existing cluster  
Select one of your existing clusters to populate fields
- Standard cluster  
Continuous integration, web serving, backends. Best choice for further customization or if you are not sure what to choose.
- Your first cluster**  
Experimenting with Kubernetes Engine, deploying your first application. Affordable choice to get started.
- CPU intensive applications  
Web crawling or anything else that requires more CPU.
- Memory intensive applications  
Databases, analytics, things like Hadoop, Spark, ETL or anything else that requires more memory.
- GPU Accelerated Computing  
Machine learning, video transcoding, scientific computations or anything else that is compute-intensive and can utilize GPUs.
- Highly available

**'Your first cluster' template**  
Experimenting with Kubernetes Engine, deploying your first application. Affordable choice to get started.

**Key fields for this cluster template**

Cluster version	1.12.5-gke.10 (latest)
Machine type	g1-small
Autoscaling	Disabled
Stackdriver Logging & Monitoring	Disabled
Boot disk size	30GB

You will be billed for the 1 node (VM instance) in your cluster [Learn more](#)

**Name**  
your-first-cluster-1

**Location type**  
 Zonal  
 Regional

**Zone**  
us-central1-a

**Master version**  
1.12.5-gke.10

**Node pools**  
Node pools are separate instance groups running Kubernetes in a cluster. You may add node pools in different zones for higher availability, or add node pools of different type machines. To add a node pool, click Edit. [Learn more](#)

**pool-1**

**Number of nodes**  
1

**Machine type**  
Customize to select cores, memory and GPUs  
small (1 shared ... 1.7 GB memory [Customize](#)

Boot disk: Standard, 30 GB Auto-upgrade: On

[Create](#) [Cancel](#) Equivalent REST or [command line](#)

- Spin up and tear down quickly
- Production Kubernetes distributions
- Cloud features: Public Load Balancers, Storage
- Limited configuration



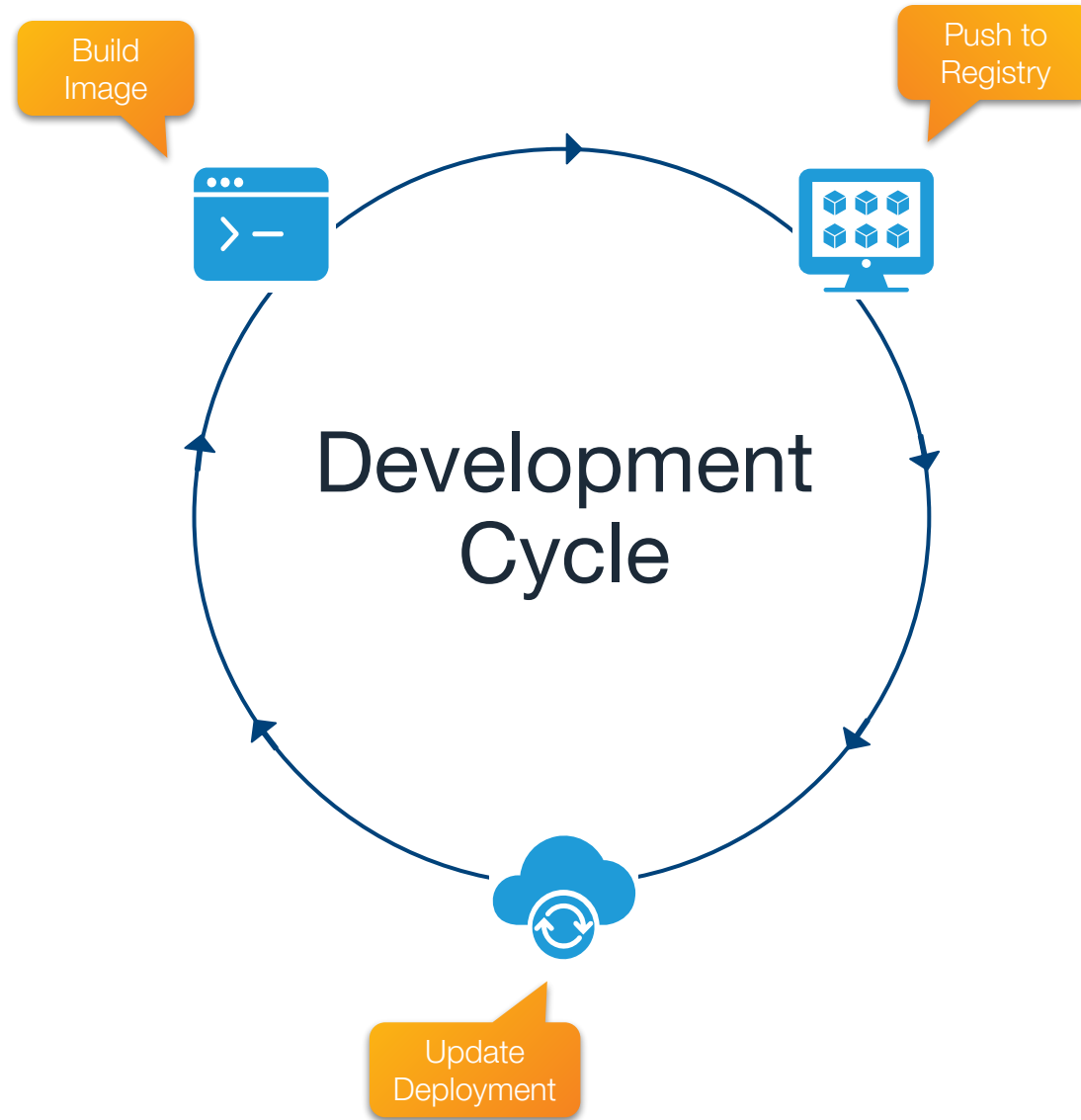
# Remote Options

## Shared Development Environments

- Long running internal clusters
- Internal Load Balancers (over VPN)
- Ingress + DNS (my-service.k8s.int.mycompany.com)
- Closely replicate production environments
- Easy to tread on each other's toes







# Automating the cycle

Draft (Microsoft) - <https://draft.sh>

- Automatically generates Dockerfile and Helm chart
- "Packs" used to define configurations for different runtimes

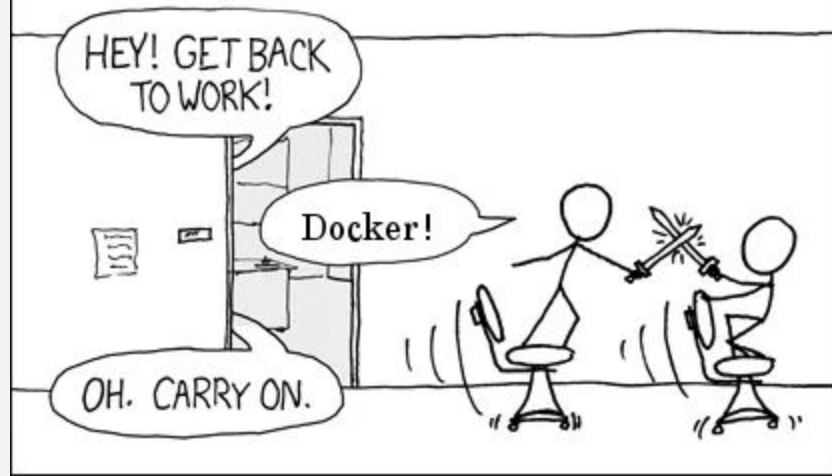
Skaffold (Google) - <https://skaffold.dev>

- Works with any directory containing a Dockerfile
- Pluggable build/push/deploy pipelines (e.g. remote builds with Google Container Builder)
- Works with any deployment tool (e.g. kubectl, Helm, etc.)



THE #1 PROGRAMMER EXCUSE  
FOR LEGITIMATELY SLACKING OFF:

"My Docker containers are building!"





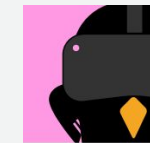
# I already have a dev server for that!

Telepresence (CNCF) - <https://telepresence.io>

- Replaces a Deployment with a two-way proxy
- Makes a local process feel like it's running in a Pod in the cluster
- Access to environment vars, volumes and service discovery

Cloud Native Development (Okteto) - <https://github.com/cloudnativdevelopment/cnd>

- Replaces a Deployment with a Pod to run a dev server
- Syncs local file changes with the Pod

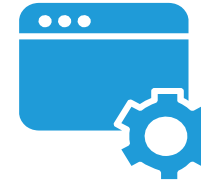


**TELEPRESENCE**

**okteto**

# Building Kubernetes extensions

- Custom Kubernetes controllers can watch API resources and do something when resources are changed
- Kubernetes extensions are written in Go
- Make use of Kubernetes libraries (client-go, apimachinery)
- Tools that help you build extensions:
  - Kubebuilder - <https://github.com/kubernetes-sigs/kubebuilder>
  - Operator SDK - <https://github.com/operator-framework/operator-sdk>
  - metacontroller - <https://metacontroller.app>



# Next Steps

## Learn More about Developing on Kubernetes

- <https://bitnami.com/kubernetes>
- <https://docs.bitnami.com/kubernetes/>
- <https://github.com/kubernetes/community/tree/master/sig-apps>

Thank You!

Q&A

