Google Cloud

An Illustrated Intro to Containers and Kubernetes

Kaslin Fields
Developer Advocate,
Google

Christina Webber
Ada Developers Academy
Graduate

Christina Webber

Ada Graduate

Kaslin Fields

Developer Advocate
Google

Kaslin Fields

@kaslinfields - Twitter

- Developer Advocate at Google
- CNCF Ambassador Gen 1
- K8s SIG ContribEx
- Cloud Native, DevOps, Kubernetes
- Tech Comic Creator
  - (https://kaslin.rocks)

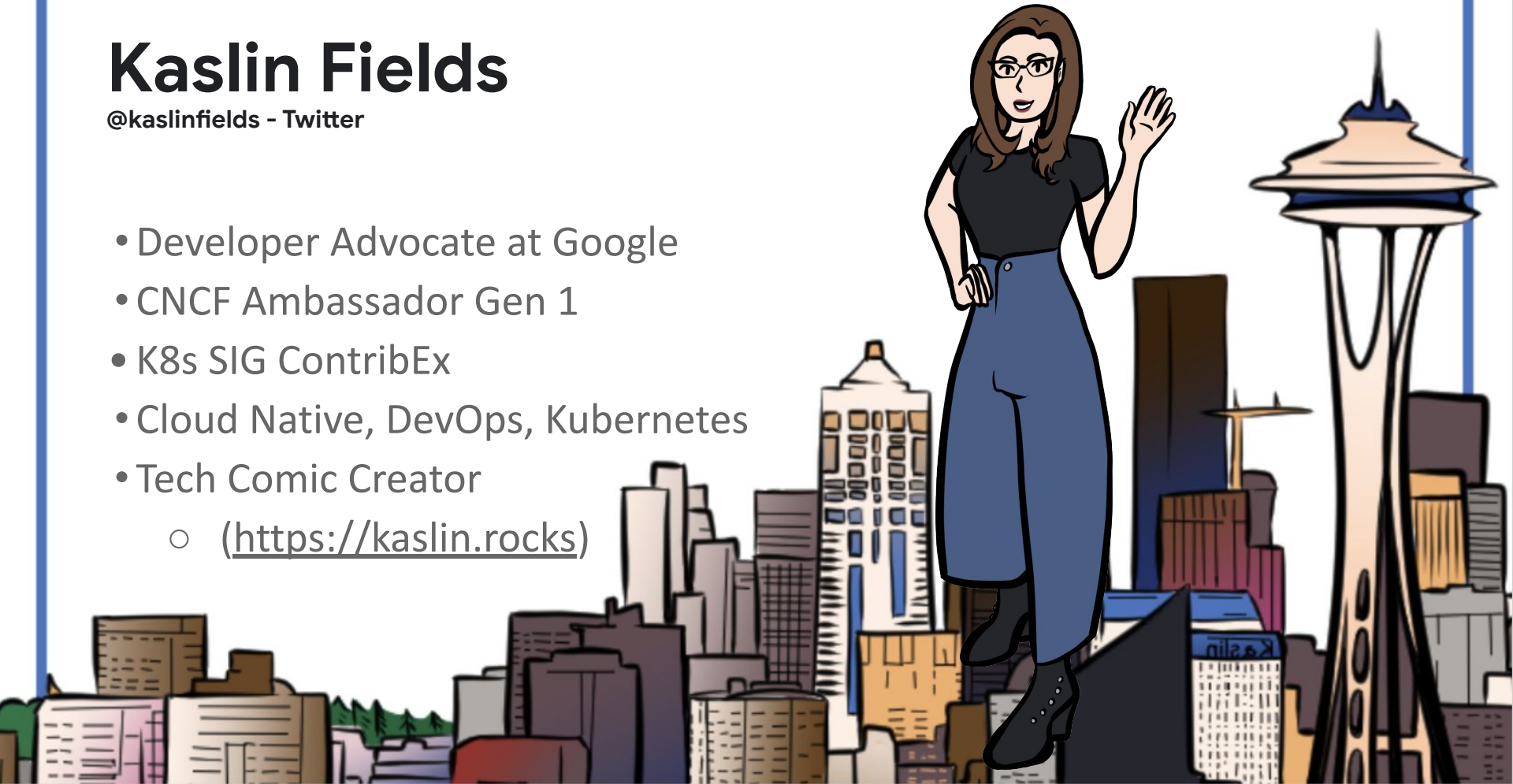


Table of Contents

Intro to Containers	01
Open Source Kubernetes and History	02
Kubernetes Basics	03
Ways to Run Kubernetes	04

01

Intro to Containers

Proprietary + Confidential

01

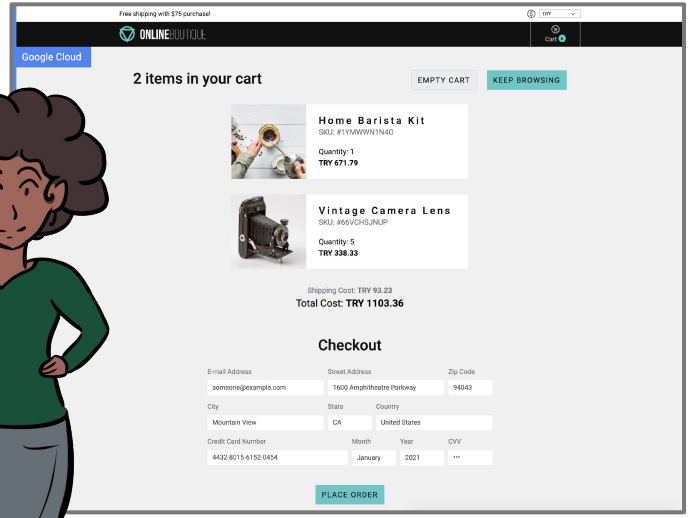
Intro to Containers

It's All About the Apps

Packaging Mechanism

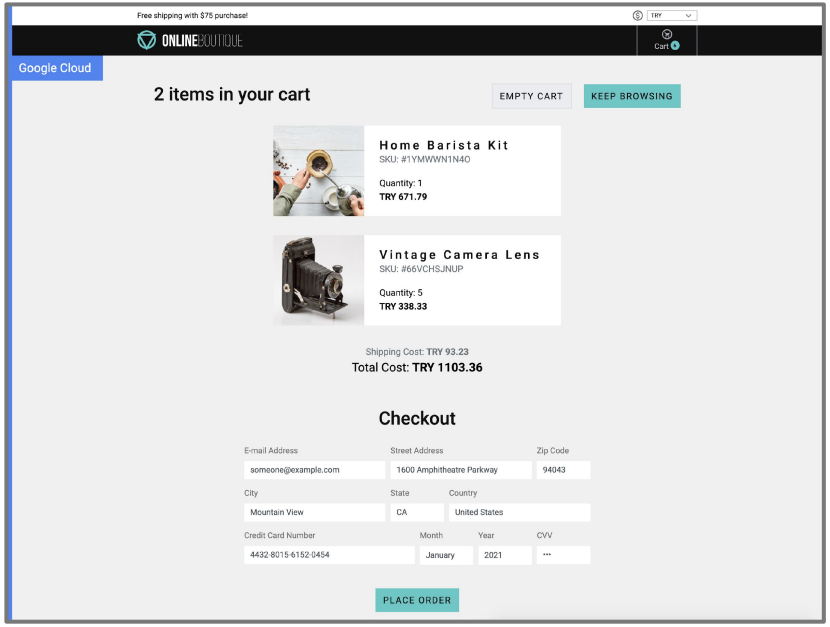
Isolation Mechanism

Why do we care about containers?



https://github.com/GoogleCloudPlatform/microservices-demo

We Care About Containers Because We Care About APPS!

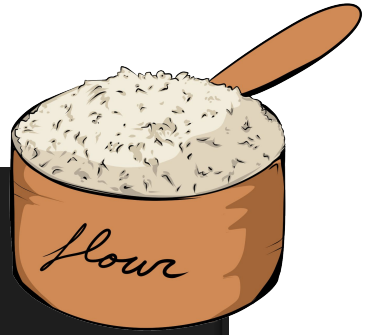


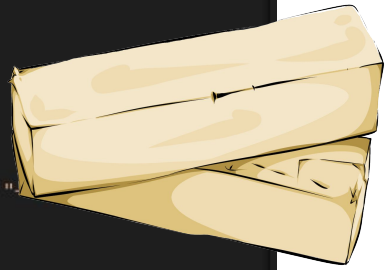
https://github.com/GoogleCloudPlatform/microservices-demo

Container Files

Dockerfile > ...

```
# Use base golang image from Docker Hub
FROM golang:1.14

# Download the dlv (delve) debugger for go (you can comment this out if unused)
RUN go get -u -v github.com/go-delve/delve/cmd/dlv

WORKDIR /src/hello-world

# Install dependencies in go.mod and go.sum
COPY go.mod go.sum ./
RUN go mod download

# Copy rest of the application source code
COPY . ./

# Compile the application to /app.
RUN go build -o /app -v ./cmd/hello-world

# If you want to use the debugger, you need to modify the entrypoint to the
# container and point it to the "dlv debug" command:
#   * UNCOMMENT the following ENTRYPOINT statement,
#   * COMMENT OUT the last ENTRYPOINT statement
# Start the "dlv debug" server on port 3000 of the container.
ENTRYPOINT ["dlv", "exec", "/app", "--continue", "--accept-multiclient", "--api-version=2", "--headless", "-

# If you want to run WITHOUT the debugging server:
#   * COMMENT OUT the previous ENTRYPOINT statements,
#   * UNCOMMENT the following ENTRYPOINT statement.
# ENTRYPOINT ["/app"]
```





Container Images

```
kaslin-macbookpro2:go-hello-world-1 kaslin$ vi Dockerfile
kaslin-macbookpro2:go-hello-world-1 kaslin$ docker build .
Sending build context to Docker daemon  60.42kB
Step 1/8 : FROM golang:1.14
1.14: Pulling from library/golang
57df1a1f1ad8: Extracting  44.04MB/50.4MB
57df1a1f1ad8: Pull complete
71e126169501: Pull complete
1af28a55c3f3: Pull complete
03f1c9932170: Pull complete
f4773b341423: Pull complete
6e11e1e4650d: Pull complete
7bdfcd218115: Pull complete
Digest: sha256:2586a80bea5329c08ef5e4e48a120f1fe947c4f928f51f189dbdb3fe8b40c9d2
Status: Downloaded newer image for golang:1.14
 ---> d6747a138341
Step 2/8 : RUN go get -u -v github.com/go-delve/delve/cmd/dlv
 ---> Running in 538fd5575f80
github.com/go-delve/delve (download)
```

Container Images

```
kaslin-macbookpro2:go-hello-world-1 kaslin$ vi Dockerfile
kaslin-macbookpro2:go-hello-world-1 kaslin$ docker build .
Sending build context to Docker daemon  60.42kB
Step 1/8 : FROM golang:1.14
1.14: Pulling from library/golang
57df1a1f1ad8: Extracting  44.04MB/50.4MB
57df1a1f1ad8: Pull complete
71e126169501: Pull complete
1af28a55c3f3: Pull complete
03f1c9932170: Pull complete
f4773b341423: Pull complete
6e11e1e4650d: Pull complete
7bdfcd218115: Pull complete
Digest: sha256:2586a80bea5329c08ef5e4e48a120f1fe947c4f928f51f189dbdb3fe8b40c9d2
Status: Downloaded newer image for golang:1.14
 ---> d6747a138341
Step 2/8 : RUN go get -u -v github.com/go-delve/delve/cmd/dlv
 ---> Running in 538fd5575f80
github.com/go-delve/delve (download)
```

```
Removing intermediate container 50adcdbfbc48
 ---> c39d9adc95d6
Step 8/8 : ENTRYPOINT ["dlv", "exec", "/app", "--continue", "--accept-multiclient",
api-version=2", "--headless", "--listen=:3000", "--log"]
 ---> Running in 5c2bb774e81c
Removing intermediate container 5c2bb774e81c
 ---> bd0a9b2b1ff7
Successfully built bd0a9b2b1ff7
```

Container Images

```
kaslin-macbookpro2:go-hello-world-1 kaslin$ vi Dockerfile
kaslin-macbookpro2:go-hello-world-1 kaslin$ docker build .
Sending build context to Docker daemon  60.42kB
Step 1/8 : FROM golang:1.14
1.14: Pulling from library/golang
57df1a1f1ad8: Extracting  44.04MB/50.4MB
57df1a1f1ad8: Pull complete
71e126169501: Pull complete
1af28a55c3f3: Pull complete
03f1c9932170: Pull complete
f4773b341423: Pull complete
6e11e1e4650d: Pull complete
7bdfcd218115: Pull complete
Digest: sha256:2586a80bea5329c08ef5e4e48a120f1fe947c4f928f51f189dbdb3fe8b40c9d2
Status: Downloaded newer image for golang:1.14
 ---> d6747a138341
Step 2/8 : RUN go get -u -v github.com/go-delve/delve/cmd/dlv
 ---> Running in 538fd5575f80
github.com/go-delve/delve (download)
```

```
Removing intermediate container 50adcbfbc48
 ---> c39d9adc95d6
Step 8/8 : ENTRYPOINT ["dlv", "exec", "/app", "--continue", "--accept-multiclient",
api-version=2", "--headless", "--listen=:3000", "--log"]
 ---> Running in 5c2bb774e81c
Removing intermediate container 5c2bb774e81c
 ---> bd0a9b2b1ff7
Successfully built bd0a9b2b1ff7
```

```
kaslin-macbookpro2:go-hello-world-1 kaslin$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>              bd0a9b2b1ff7        24 seconds ago      903MB
golang              1.14                d6747a138341        2 weeks ago         810MB
kubecon             latest              0e3139269256        8 weeks ago         43.3MB
<none>              <none>              339ab11b124e        8 weeks ago         1.5GB
alpine              latest              a24bb4013296        4 months ago        5.57MB
golang              1.12-alpine         76bddfb5e55e        7 months ago        346MB
kaslin-macbookpro2:go-hello-world-1 kaslin$
```

Containers

```
kaslin-macbookpro2:go-hello-world-1 kaslin$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>              bd0a9b2b1ff7        24 seconds ago      903MB
golang              1.14                d6747a138341        2 weeks ago         810MB
kubecon             latest              0e3139269256        8 weeks ago         43.3MB
<none>              <none>              339ab11b124e        8 weeks ago         1.5GB
alpine              latest              a24bb4013296        4 months ago        5.57MB
golang              1.12-alpine         76bddfb5e55e        7 months ago        346MB
kaslin-macbookpro2:go-hello-world-1 kaslin$
```

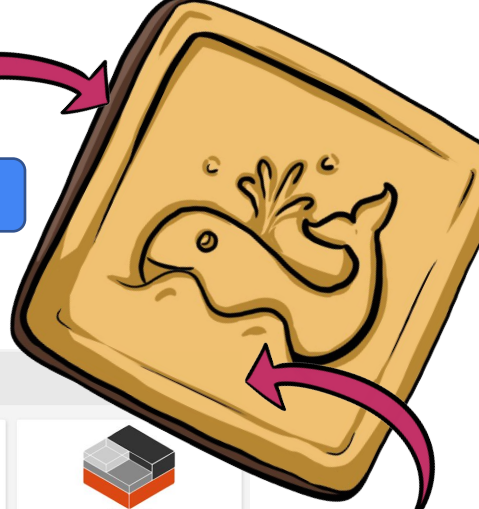
```
kaslin-macbookpro2:go-hello-world-1 kaslin$ docker run bd0
2020-09-27T21:31:08Z info layer=debugger launching process with args: [/app]
API server listening at: [::]:3000
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
    NAMES
52b2db87f120        bd0                 "dlv exec /app --con…"   24 seconds ago      Up 24 seconds
    quizzical_snyder
```

Distributing Your Apps Made Easy!

Containers are:

Fast. Lightweight. Secure.



Application

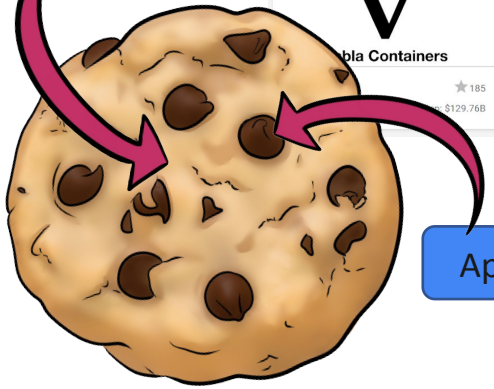
Depend-encies

Runtime - Container Runtime (12)

containerd	CRI-O	Firecracker	gVisor	Kata Containers	lxd
Cloud Native Computing Foundation (CNCF)	Cloud Native Computing Foundation (CNCF)	Amazon Web Services	Google	OpenStack	Canonical
★ 5,165	★ 2,282	★ 10,593	★ 9,513	★ 1,769	★ 2,497
		MCap: $1.02T	MCap: $996.9B		Funding: $12.8M

bla Containers	Pouch	runc	Singularity	SmartOS	Unik
	Alibaba Cloud	Open Container Initiative (OCI)	Sylabs	Joyent	Solo.io
★ 185	★ 4,131	★ 6,640	★ 1,510	★ 1,361	★ 2,089
$129.76B	MCap: $579.08B			Funding: $131M	Funding: $13.5M

Depend-encies



Application

02

Open Source Kubernetes

Proprietary + Confidential

02

Open Source K8s & History

Intro to the CNCF

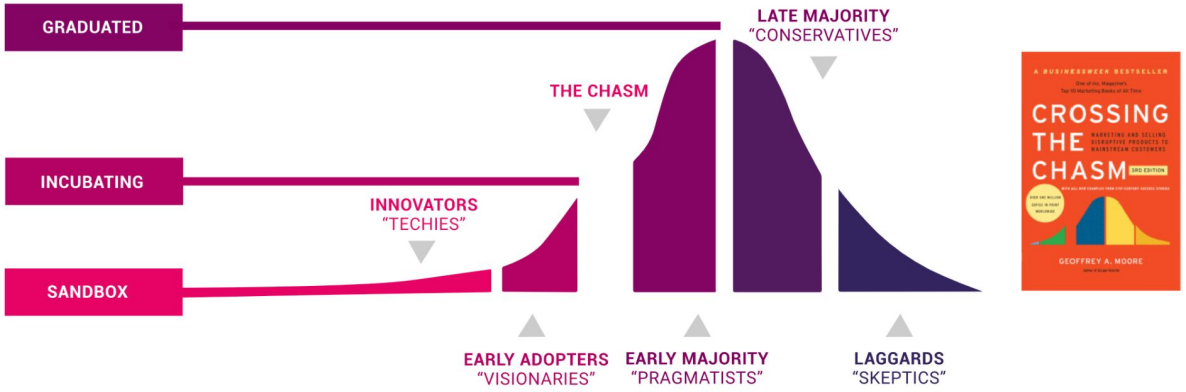
The Cloud Native Ecosystem

Kubernetes Fun Facts

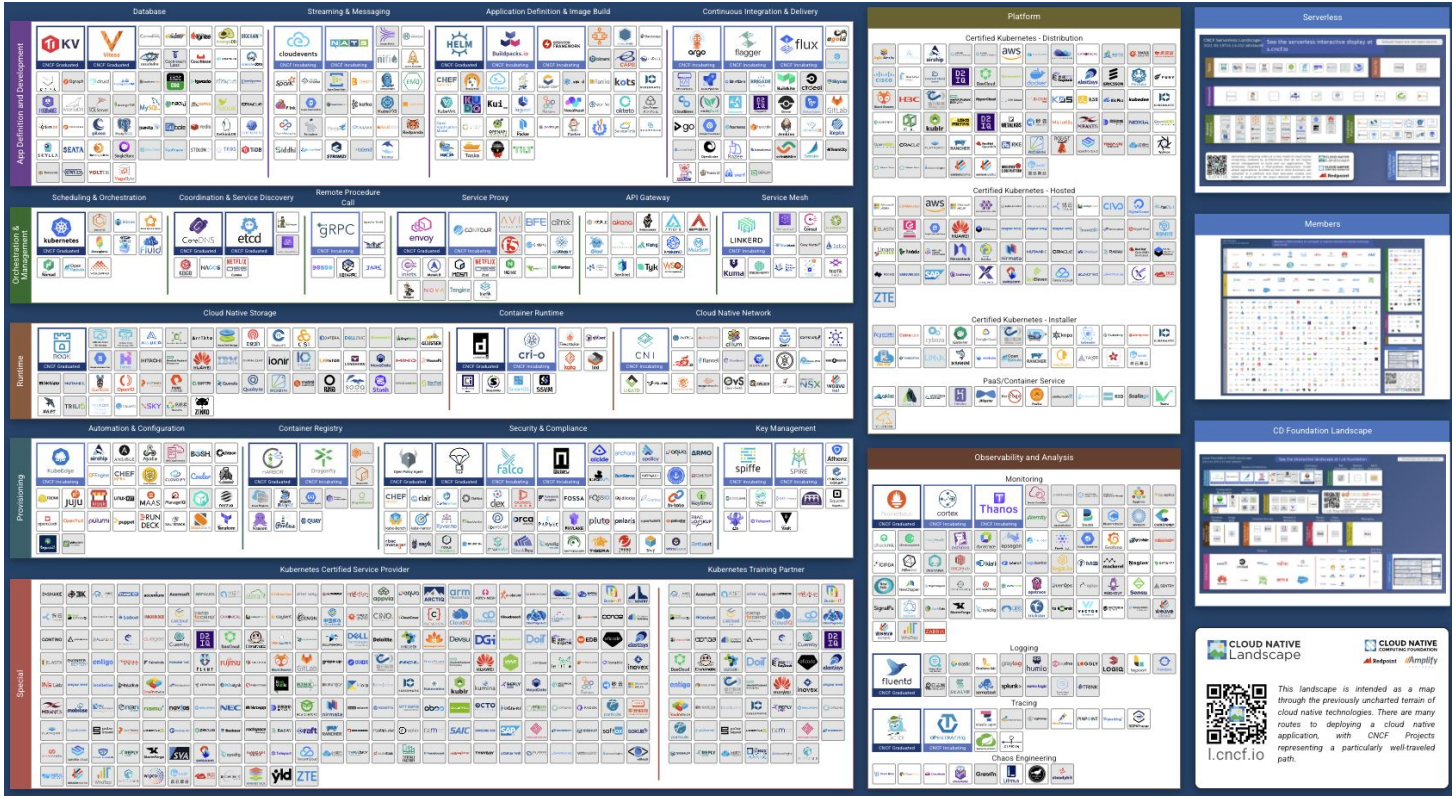
Kubernetes Origins

CLOUD NATIVE
COMPUTING FOUNDATION

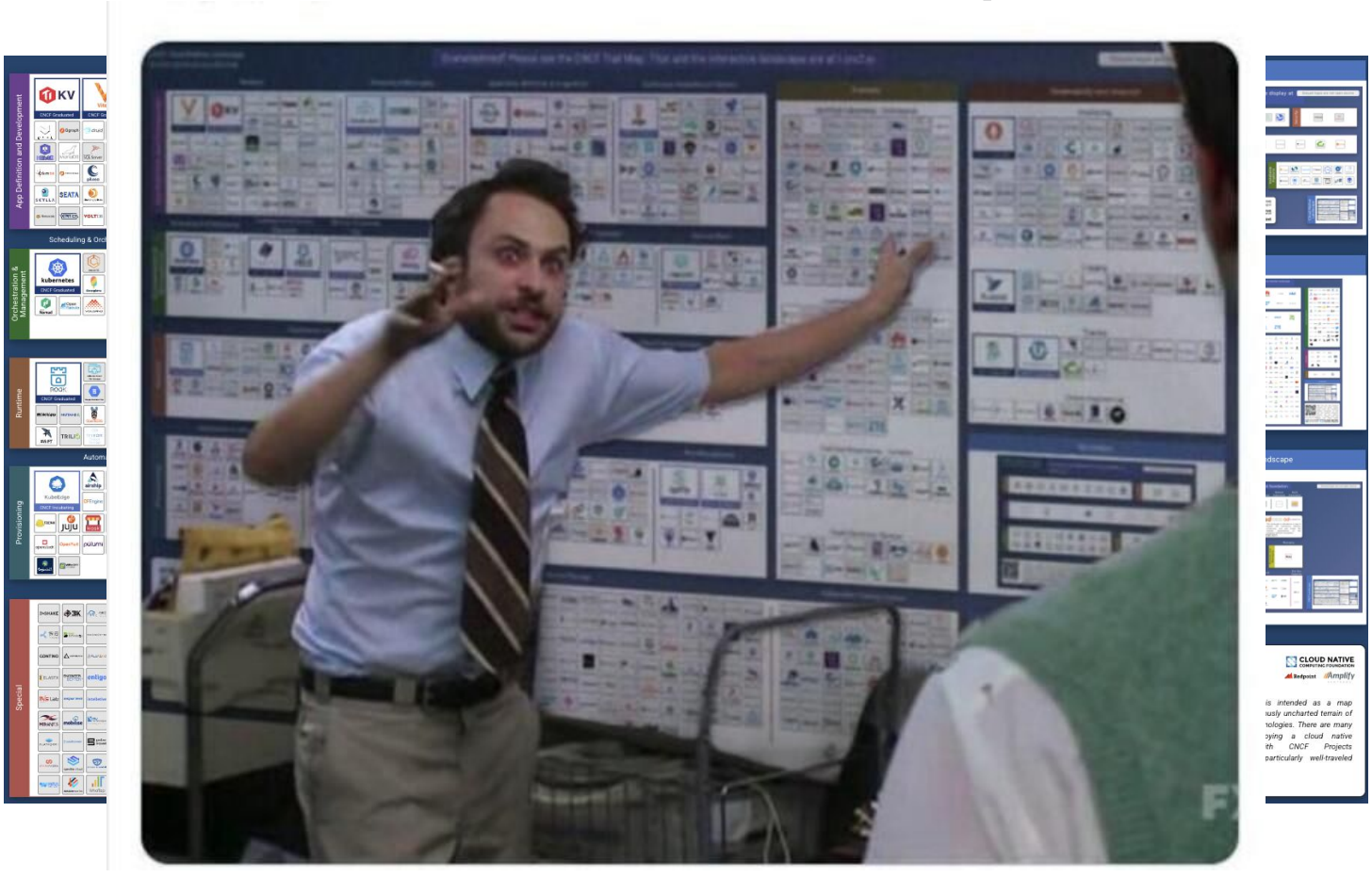
Cloud Native Computing Foundation (CNCF) serves as the vendor-neutral home for many of the fastest-growing open source projects, including Kubernetes, Prometheus, and Envoy.



Welcome to the Cloud Native Landscape!

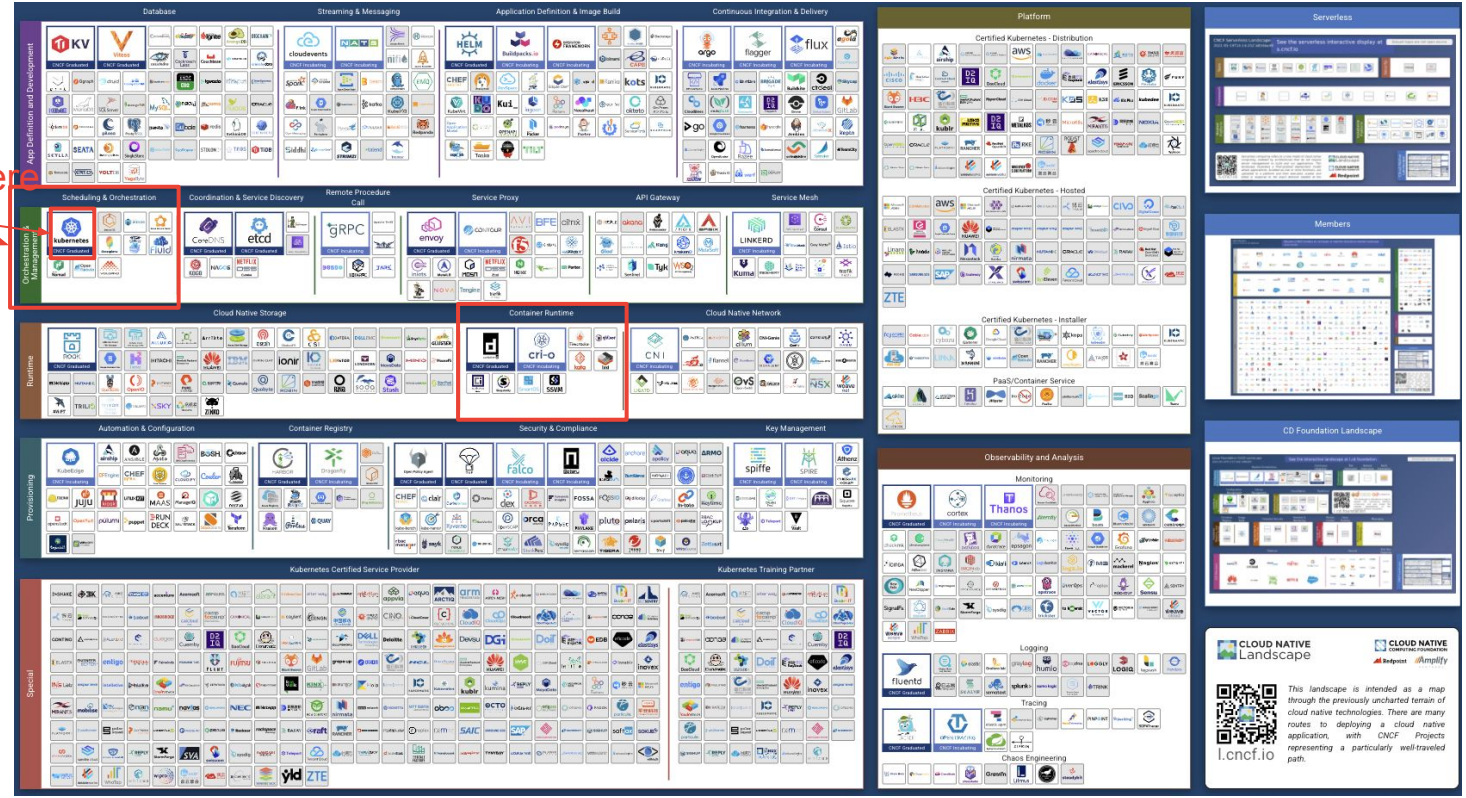


Welcome to the Cloud Native Landscape!



Welcome to the Cloud Native Landscape!

You are here



Greek for “Helmsman”; also the root of the words “governor” and “cybernetic”

- Manages container clusters
- Inspired and informed by Google’s experiences and internal systems
- Supports multiple cloud and bare-metal environments
- Supports multiple container runtimes
- **100% Open source**, written in Go

Manage applications, not machines

Additional Fun Facts!

Ian Coldwater 📦 💥
@IanColdwater

Replying to @bink

The Kubernetes logo is a heptagon and the wheel has seven points on purpose!

Kubernetes is descended from Borg at Google, and its internal code name was that of an ex-Borg character on Star Trek, Seven of Nine 🤓

6:52 PM · Jul 12, 2021 · Twitter for iPhone

21 Retweets **7** Quote Tweets **207** Likes

Check out the Documentary!

03

Kubernetes Basics

Proprietary + Confidential

03

Kubernetes Basics

Container Orchestration - Containers at Scale

Pods

Replication Controllers

Scheduling

Services

and More!

I have this analogy where I talk about containers as cookies.

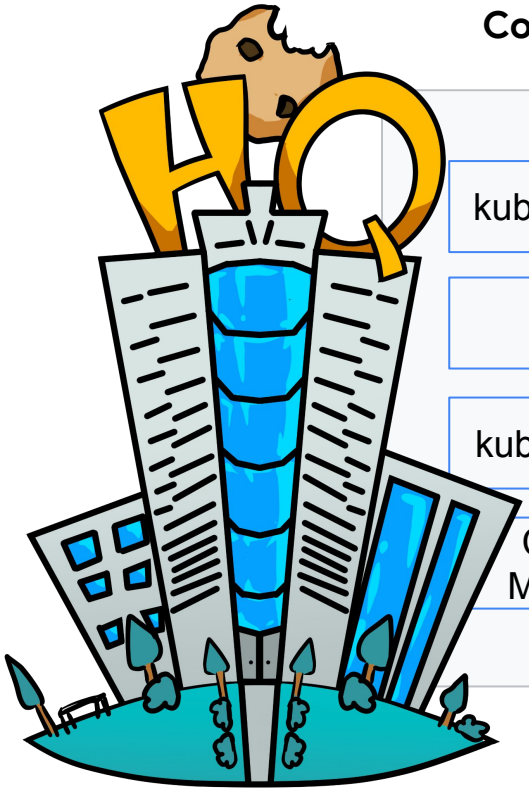
Art by Kaslin! (@kaslinfields on twitter, @kaslin on github)

Container Orchestrator = Containers at Scale

Art by Kaslin! (@kaslinfields on twitter, @kaslin on github)

Container Orchestrator = Containers at Scale

Art by Kaslin! (@kaslinfields on twitter, @kaslin on github)

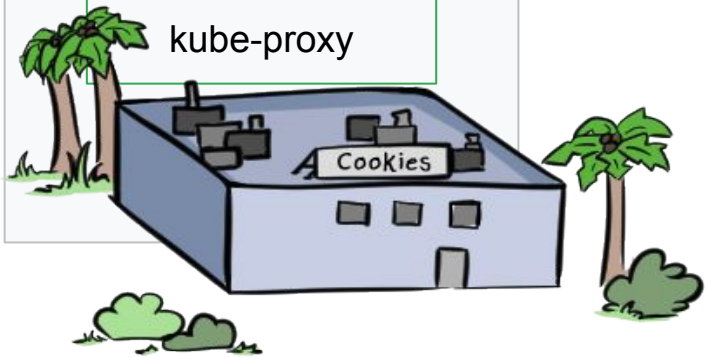


Control Plane

- kube-api-server
- etcd
- kube-scheduler
- Controller Manager(s)

Worker Node

- kubelet
- Container Runtime
- kube-proxy

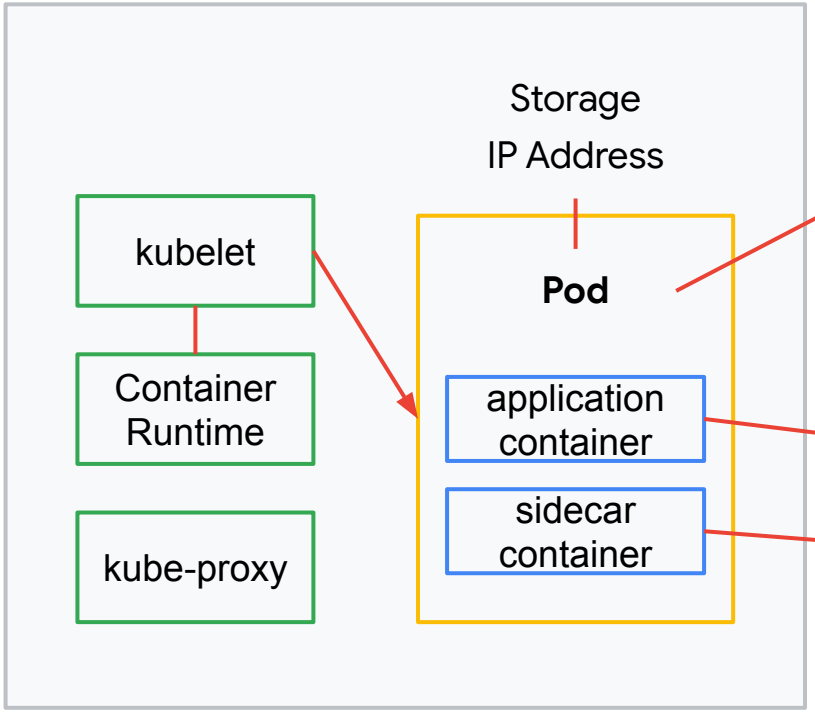


In Kubernetes, Containers are Packaged in Pods

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
    - name: nginx-container
      image: nginx
    - name: backend-container
      image: redis
```

Art by Kaslin! (@kaslinfields on twitter, @kaslin on github)

In Kubernetes, Containers are Packaged in Pods




```yaml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
    - name: nginx-container
      image: nginx
    - name: backend-container
      image: redis
```

Kubernetes Replication Controllers: Repeatable, Predictable, HA Pods

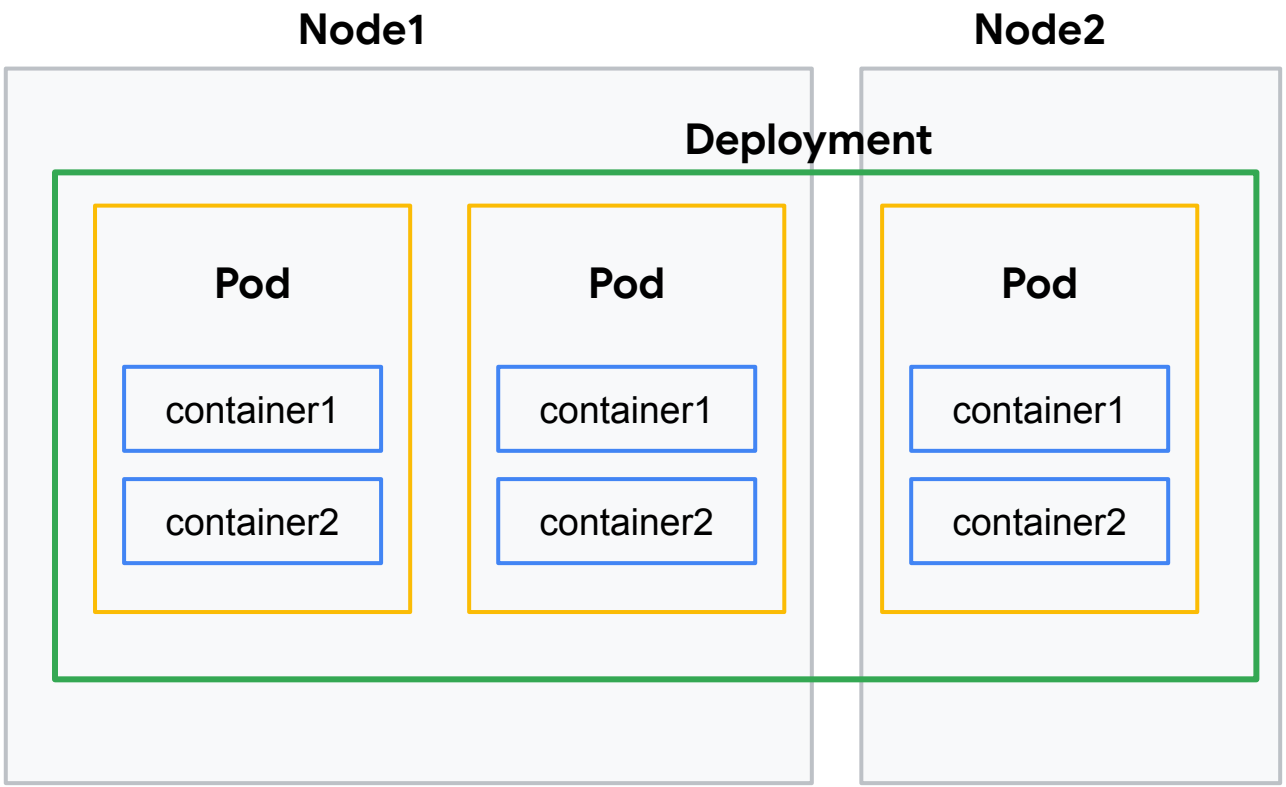
Types of Replication Controllers

- Deployment
- Job
- CronJob
- DaemonSet
- StatefulSet

```
kaslin-macbookpro2:gateway kaslin$ kubectl scale --replicas=3 -f demo-app.yaml
deployment.apps/foo scaled
```

Art by Kaslin! (@kaslinfields on twitter, @kaslin on github)

Kubernetes Replication Controllers



Types of Replication Controllers

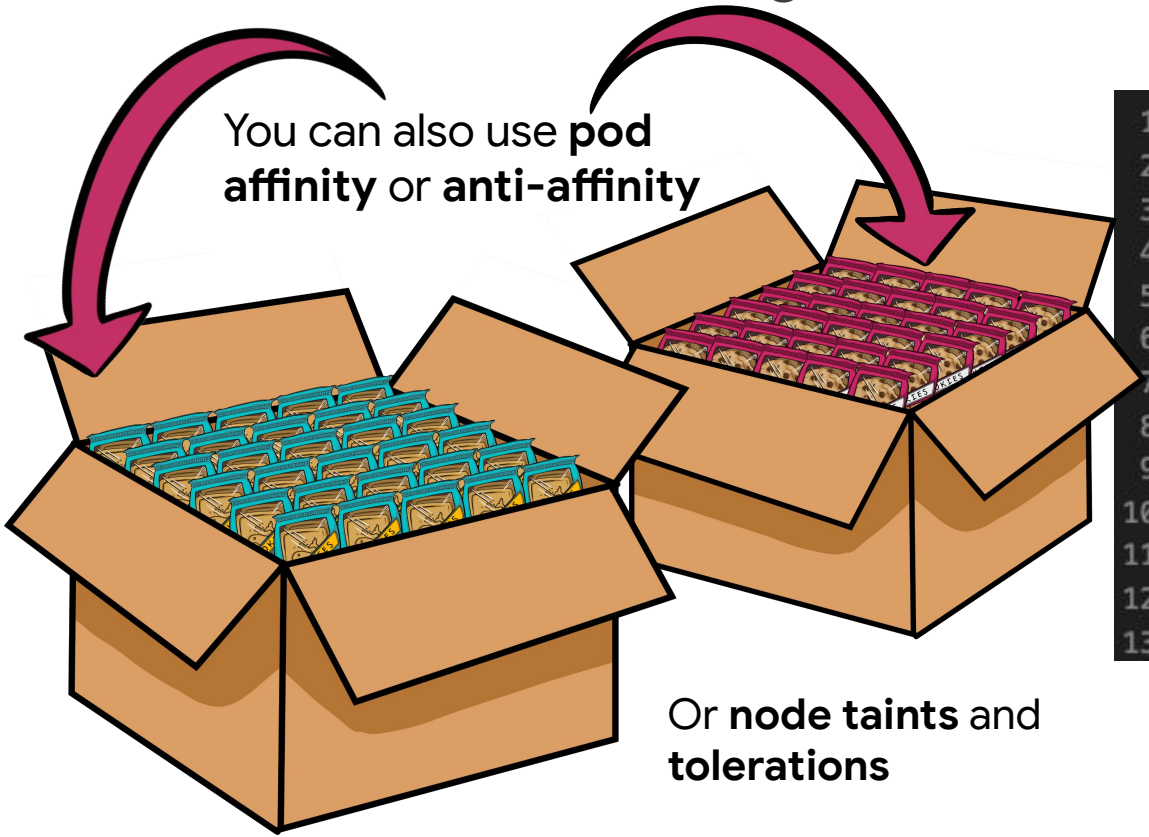
- Deployment
- Job
- CronJob
- DaemonSet
- StatefulSet

Kubernetes Scheduling

Art by Kaslin! (@kaslinfields on twitter, @kaslin on github)

Google Cloud

Kubernetes Scheduling

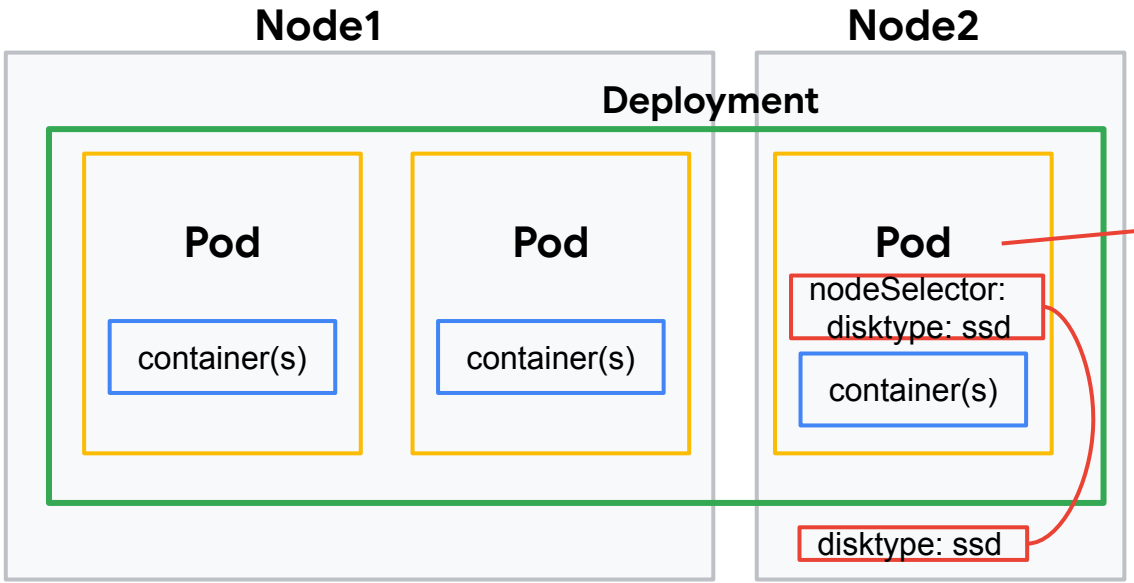



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  nodeSelector:
    disktype: ssd
```

Art by Kaslin! (@kaslinfields on twitter, @kaslin on github)

Google Cloud

Kubernetes Scheduling




```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  nodeSelector:
    disktype: ssd
```

Kubernetes Services

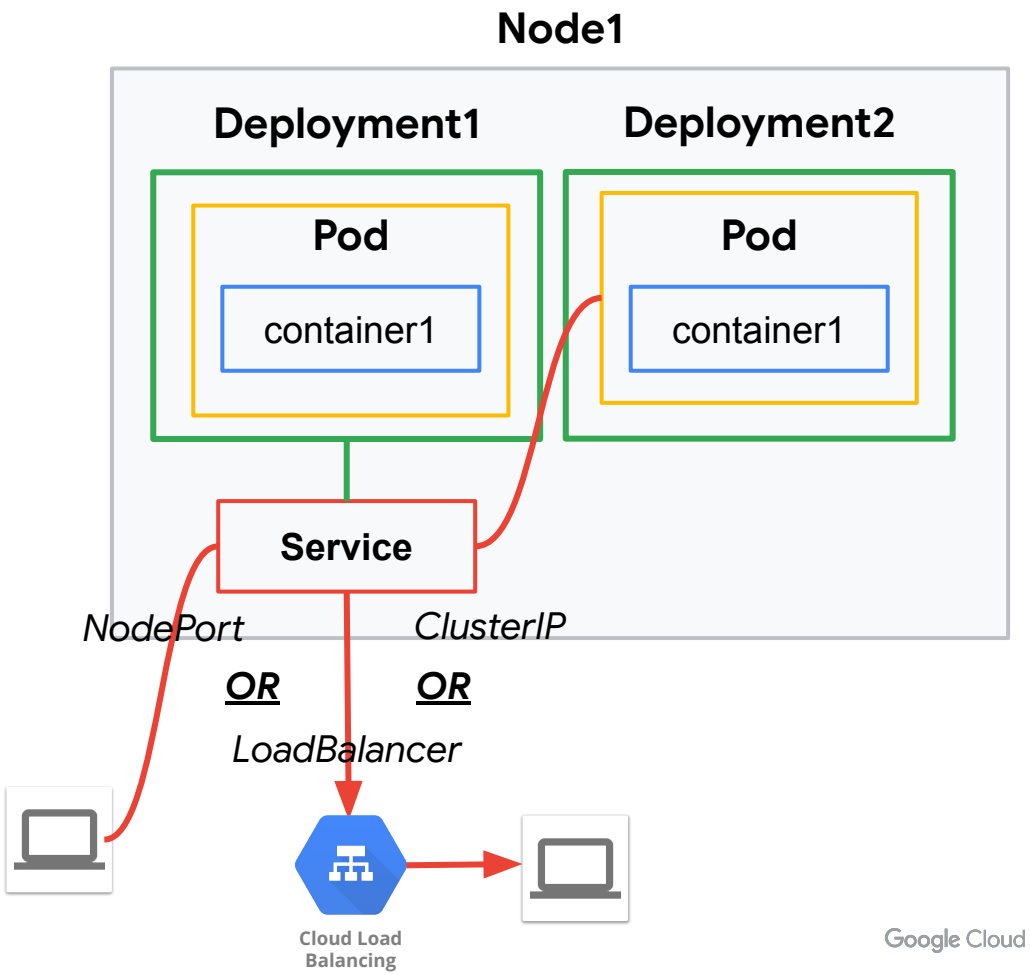
```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: MyApp
  ports:
    - port: 80
      targetPort: 9376
```

Types of Services:

- ClusterIP
- NodePort
- Loadbalancer

Kubernetes Services

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: MyApp
  ports:
    - port: 80
      targetPort: 9376
```



There are a lot more Kubernetes concepts to learn!

- RCs (ie. Deployments, StatefulSets, etc.)
- Ingress
- Persistent Storage
- Namespaces
- and More...

Check out [Kubernetes Essentials from Google Cloud on YouTube](#)!

05

How to Run Kubernetes

Options for Running Kubernetes

- **Minikube**
  - Tool for running a single-node local cluster
- **KIND** (Kubernetes IN Docker)
  - Lighter weight local clusters
- **Cloud Provider Managed Services** (like Google Kubernetes Engine)
  - Get a cluster with one click
  - Control Plane managed and operated by Cloud Provider
- **Self-Hosted DIY**
  - Raspberry pis or other hardware
  - Kubeadm can make it easier
  - Can be multi-node
  - K3s is a cool project meant for “Edge” use cases

Thank You!

Demo Time!