



Milking the most
out of thousands of
Kubernetes clusters

What to expect from the session



- Intro
- How is CFA using K8s?
- What does our architecture look like?
- How are we engineering around K8s for our business?
- Q&A

Chick-fil-A

RIGHT LANE
MUST
TURN RIGHT

NO CHANGING
LANES

#NYCCFA





AT PEAK HOUR

1 sandwich every 16 seconds

1 box of nuggets every 25 seconds

1 order of waffle fries every 14 seconds

1 car through the drive thru every 22 seconds

267 total transactions

Chick-fil-A Architecture (2017)

Cloud



OAuth Server



MQTT

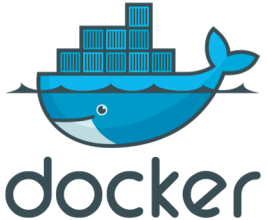


Analytics



Management

Edge



Edge Tools



Local Auth



MSGing



Web Server



Event Fwding



Apps ...



Local Persistence/Storage

Connectivity



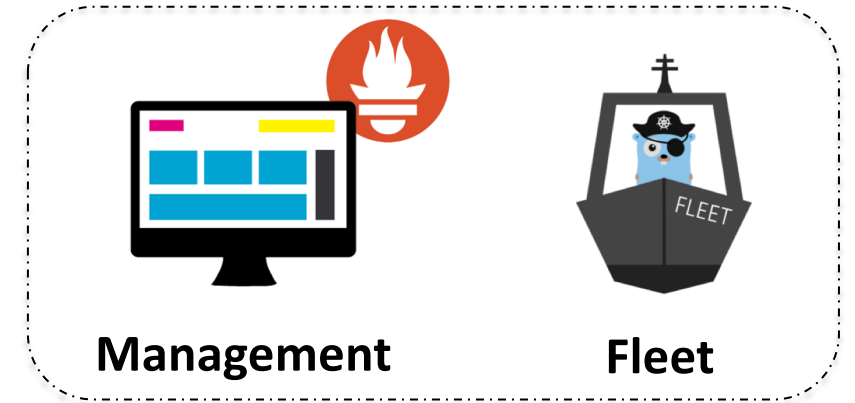
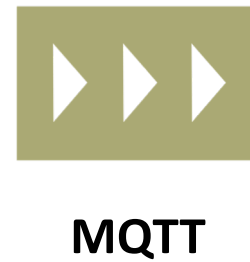
Bluetooth

Things



Chick-fil-A Architecture (Today)

Cloud



Edge



Connectivity



Things



Why Containers? Why Kubernetes?

Idea

Code

**Production
Code**

Value

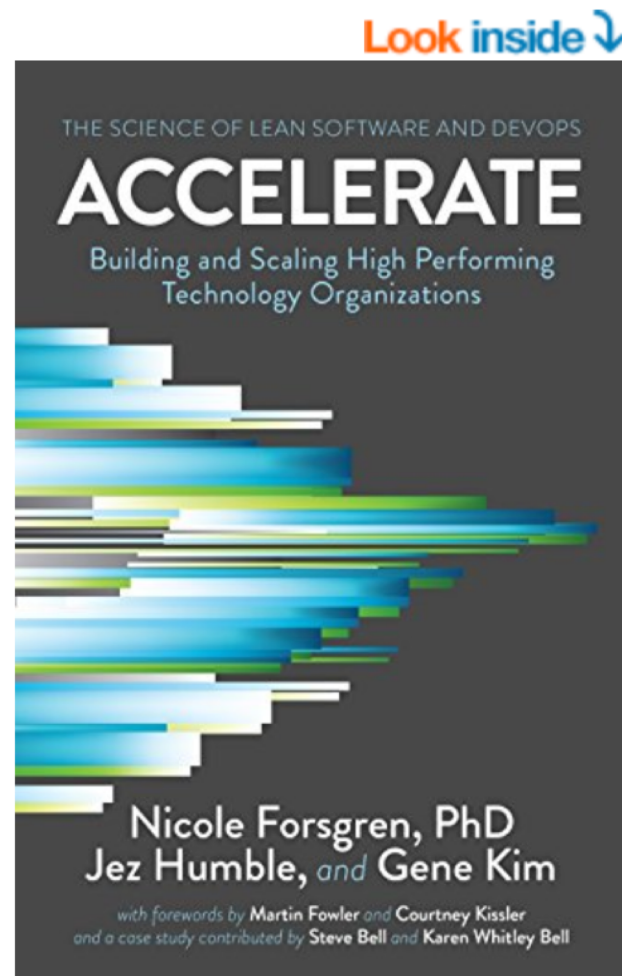
Impact

Optimize for



Accelerate

i You purchased this item on March 28, 2018.
[View this order](#) | You can find this title on your Amazon Kindle [apps](#) and devices, or the [Manage Your Content and Devices](#) page. Need additional help? [Click here](#)



Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations Kindle Edition

by [Nicole Forsgren PhD](#) (Author), [Jez Humble](#) (Author), [Gene Kim](#) (Author)

★★★★★ (5 stars) 21 customer reviews

#1 Best Seller in Information Management


> [See all 3 formats and editions](#)

Kindle
\$10.99

Read with Our **Free App**

Paperback
\$16.96

34 Used from \$12.95
46 New from \$15.07

 **Audiobook**
\$7.49

or 1 credit

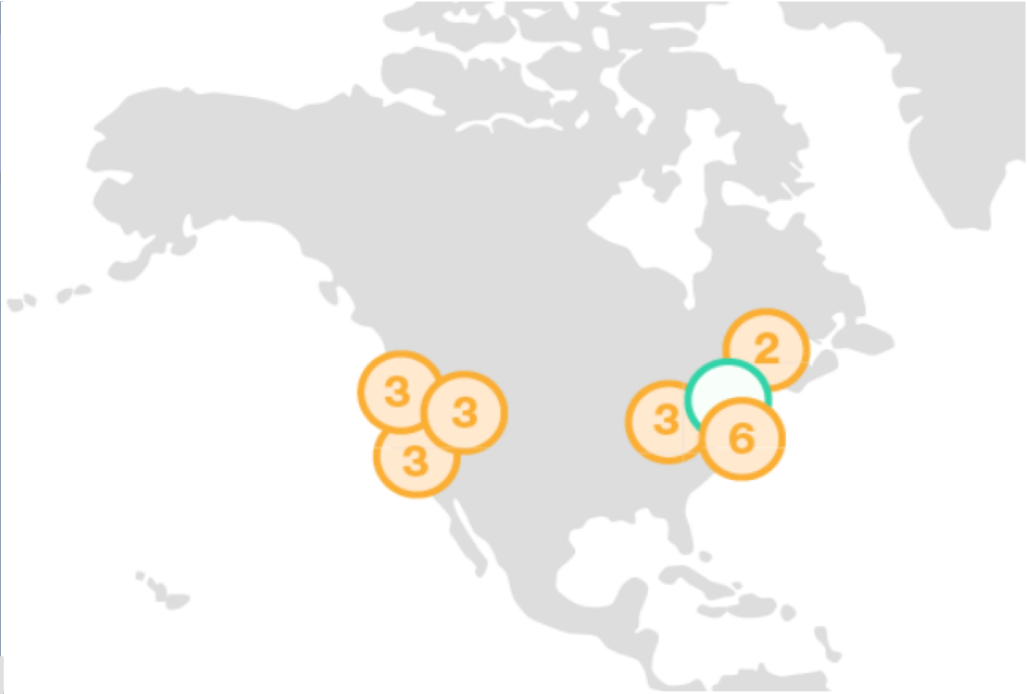
Accelerate your organization to win in the marketplace.

North American Data Centers

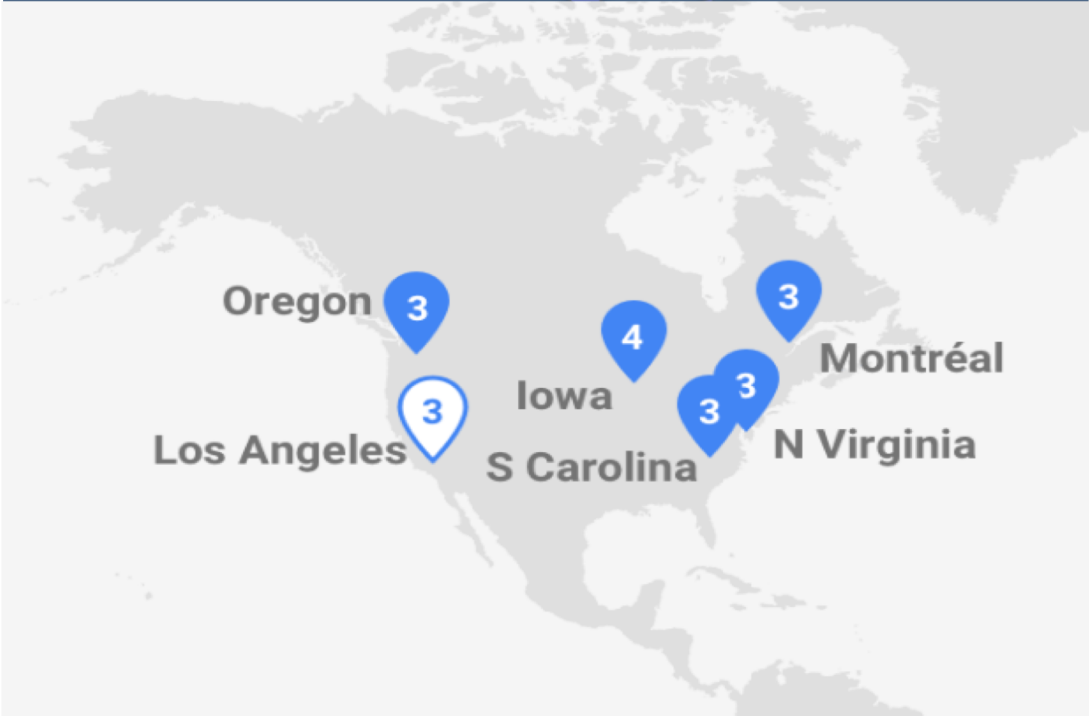
Azure



AWS



Google Cloud

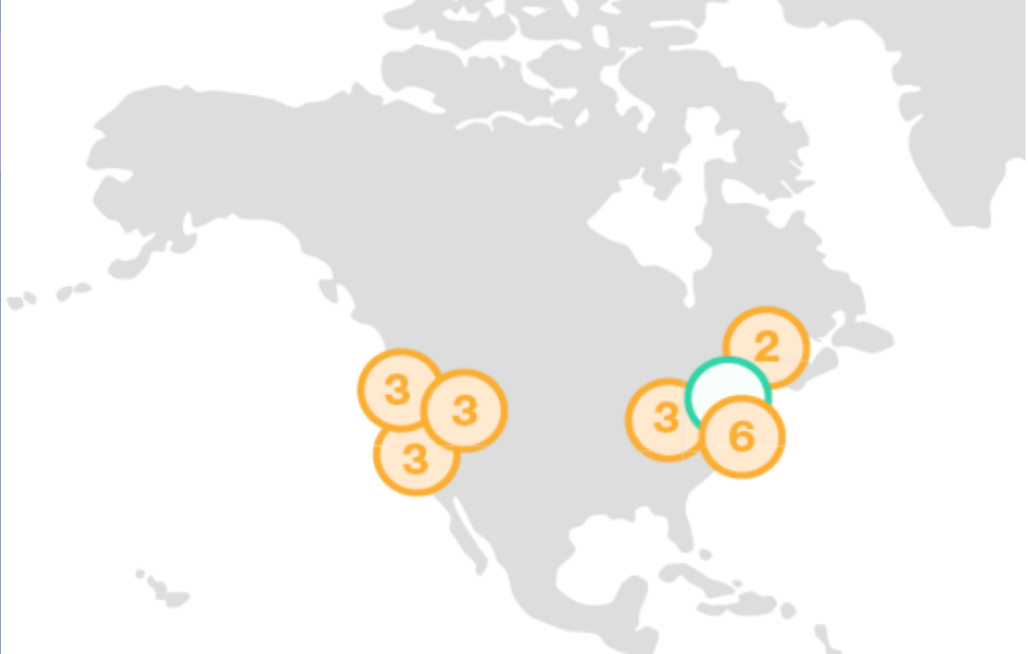


North American Data Centers

Azure



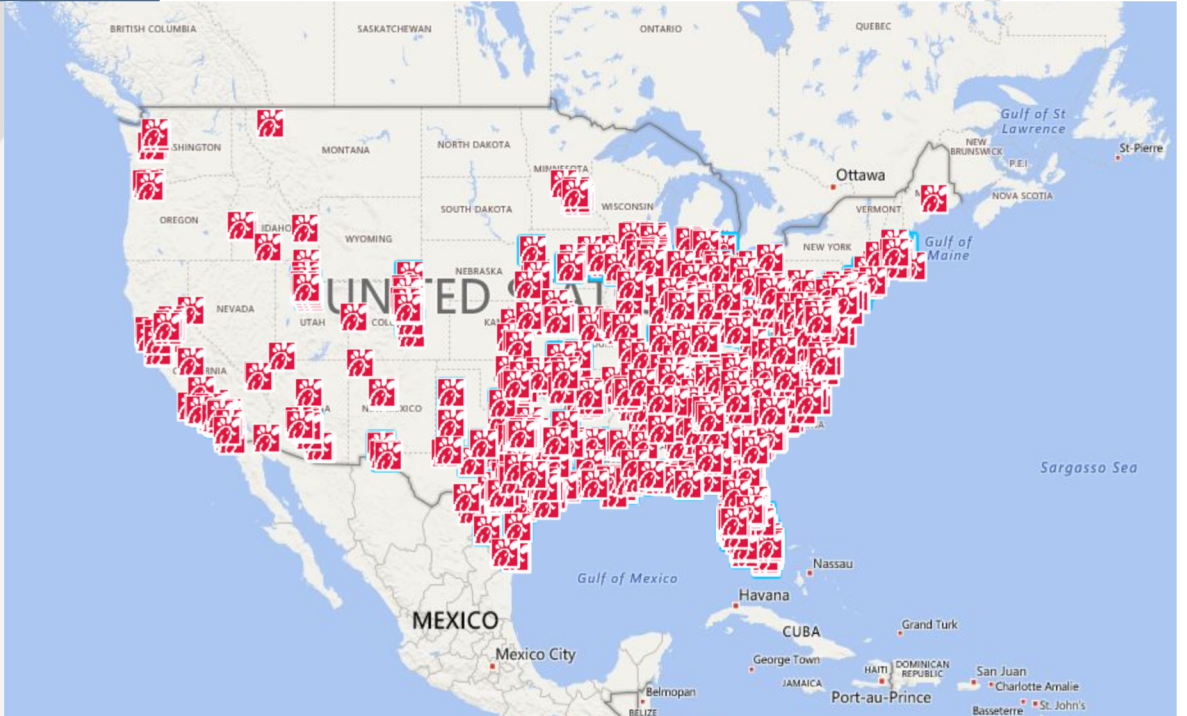
AWS



Google Cloud



Cloud-fil-A



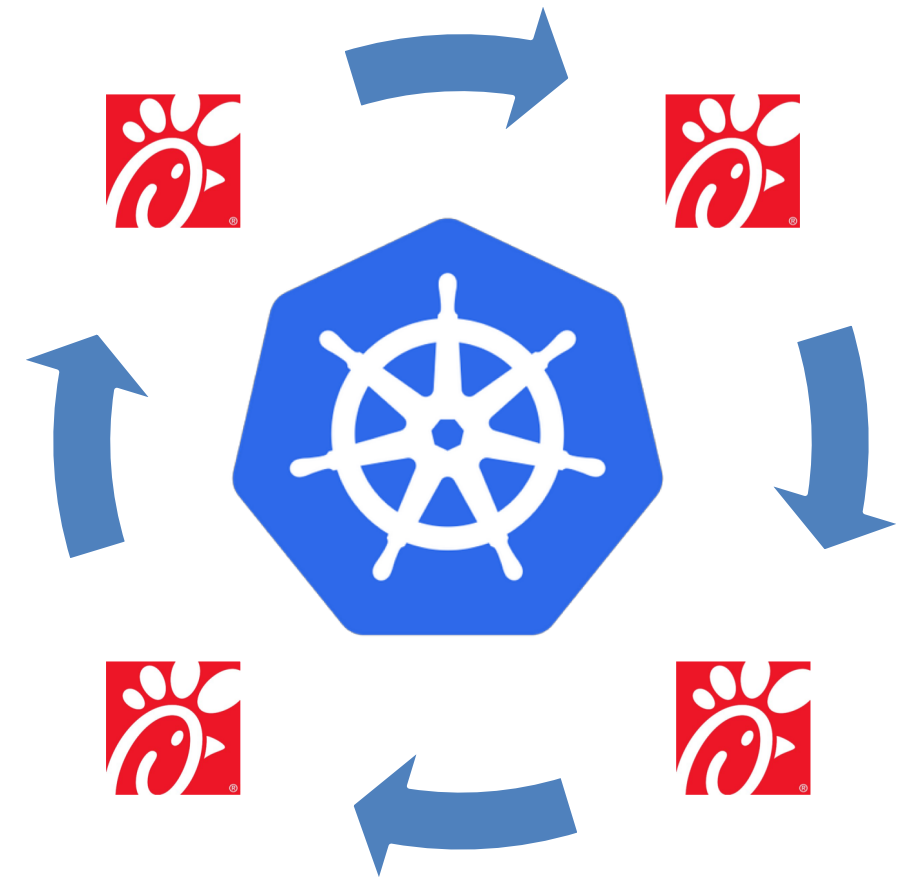
Restaurant “Data Centers”



Intel: Quadcore processor, 8 GB RAM, SSD

Engineering Around K8s

- How we build and repair bare metal clusters
- SRE Lessons Learned
- How we deploy applications to thousands of clusters



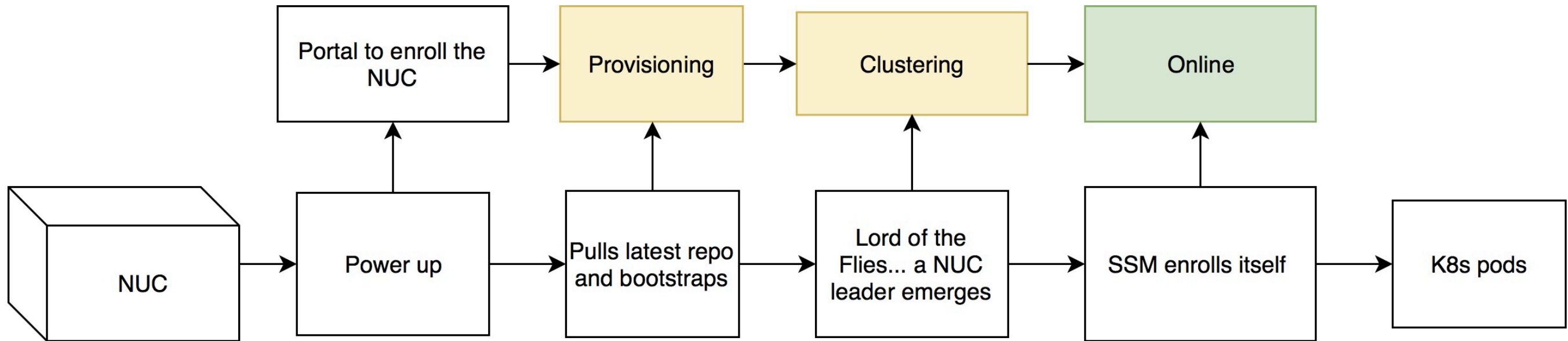
Challenges of Bare Metal K8s clustering at scale

- Goal: #code2prod
- Simple enough for a non-technologist to install
- Manageable remotely
- Automated device discovery and self-clustering
- Self healing & HA

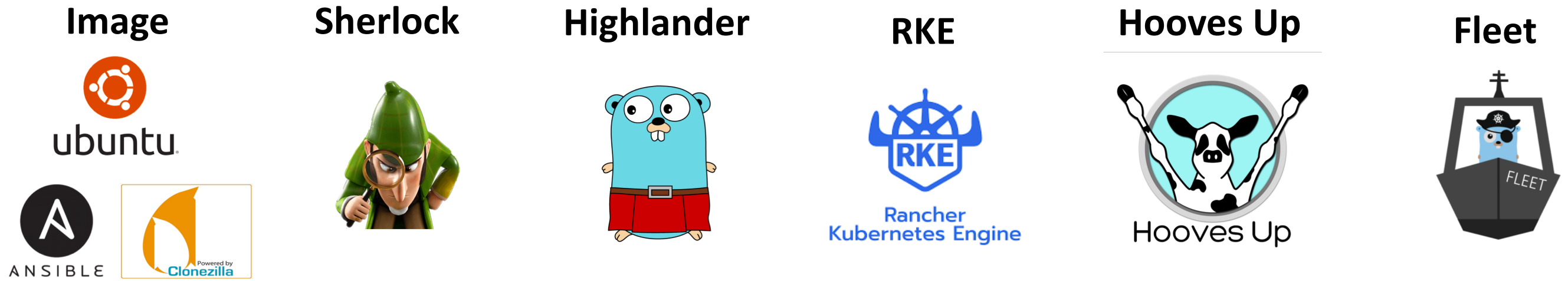


How we Bare Metal Cluster K8s at scale

PROCESS



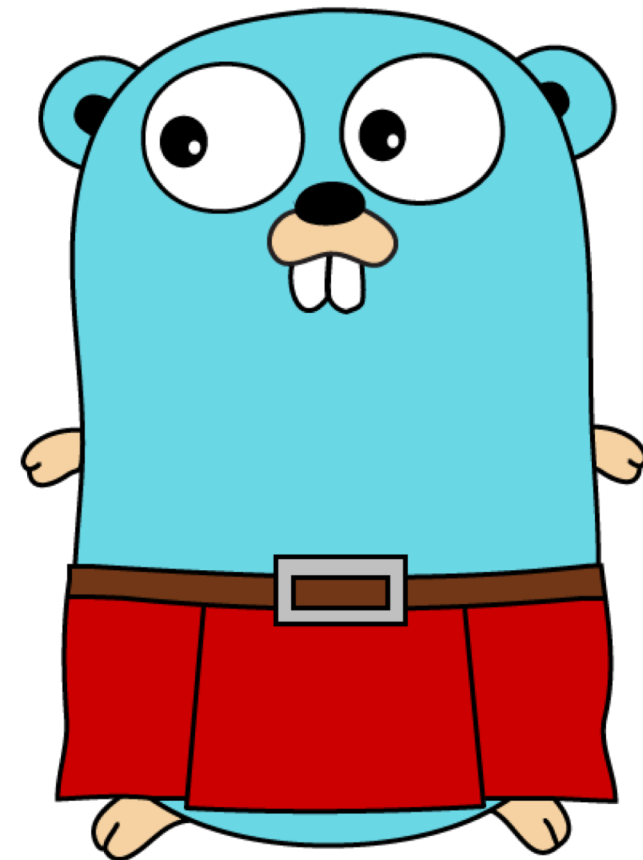
TOOLS



Bootstrapping Clusters

- Highlander
 - Node coordination and clustering leader election using UDP
 - Execute clustering (RKE)
 - Swap KubeDNS for CoreDNS
 - Base OAuth identity negotiation
 - Controller Pods (control plane activity/Istio)

There can be only one



Initializing Clusters

What we considered

- Kops = love it, no bare metal
- Kubespray = slow + brittle
- kubeadmin = maybe in the future
- RKE = fairly simple, works for us

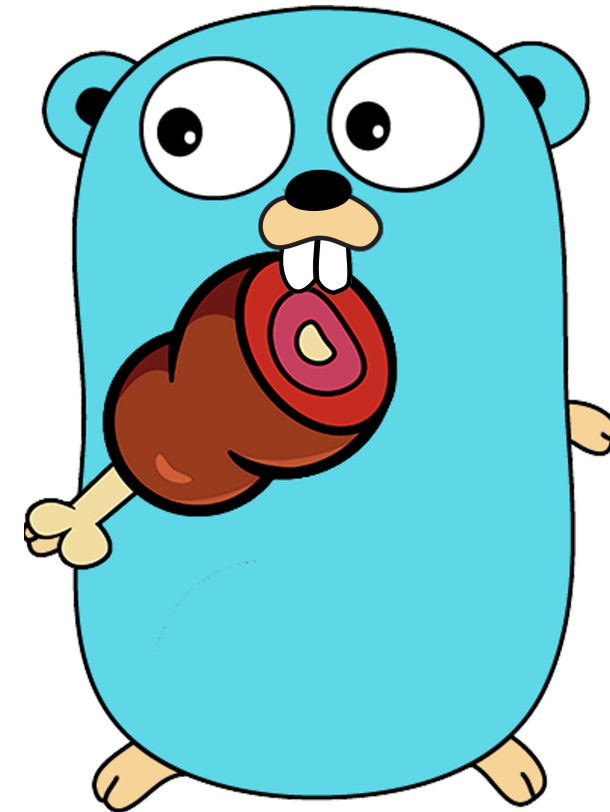


Future State?

- Stick w/ RKE, Kubeadmin, or roll our own to meet our needs

Resetting Cluster State

- Requirement: Need to be able to re-image remotely
- Solution: Overlay FS + HAMS
 - Manages wiping clusters and restoring to base



Hooves Up

- Self-healing AWS SSM Registration
- Free even for non-AWS deployments
- Able to do remote commands and patch reporting/management



Hooves Up

Lessons learned

- Use K8s feature set and don't reinvent the wheel
- MVP. MVP. MVP.
- Ensure aggregated and searchable logging
- Deep health checks are a must --> Use /healthz
- Every service needs “/metrics” endpoint

Minimum Viable Product



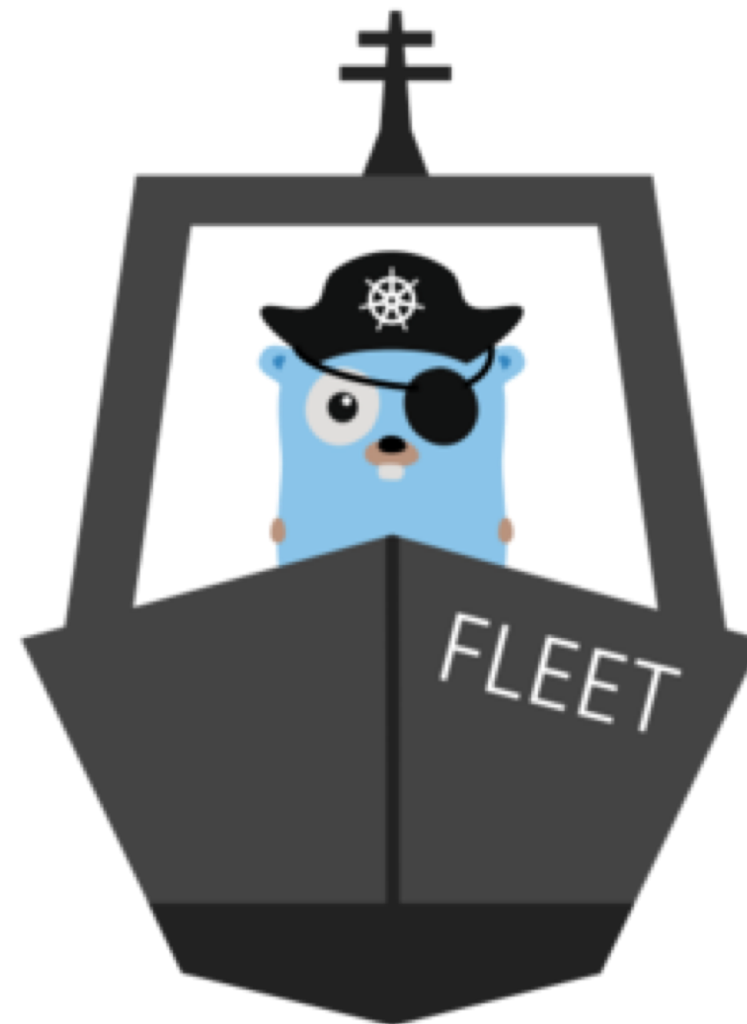
How do we deploy to our restaurants?

- Large number of deployment targets
- Complex success/fail criteria
- Array of application types
- What approaches did we consider?



Introducing Fleet

- **Design Goals**
 - **Simple** to use / reason about
 - Use **declarative** approach
 - Support for **variety** of deployment models (canary, blue/green)
 - Rollout over **flexible time** period
 - **Sane rollback** behaviors
 - Leverage **standard** k8s API
 - Full **visibility**



Fleet Ecosystem Components

- **Fleet Client**
 - Git webhook, REST call, CLI
- **Fleet Server API**
 - Code generation for deployment, service, ingress files
 - Git management for cluster repositories
 - Deployment status tracking
- **Atlas**
 - Repository of deploy-ready, k8s compliant application files
- **Vessel**
 - Deployed on cluster, git pull, kubectl apply, report status
- **Dashboards**

Sample Templates

```
# Deployment.yaml | Source: Generated by Fleet. Yarr!
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  namespace: {{.Namespace}}
  name: {{.Name}}
  labels:
    app: {{.Name}}
    heritage: Fleet
spec:
  replicas: {{.Replicas}}
  selector:
    matchLabels:
      app: {{.Name}}
  template:
    metadata:
      labels:
        app: {{.Name}}
        deploymentHash: {{.DeployHash}}
    spec:
      containers:
        - name: {{.Name}}
          image: {{.Image}}
          imagePullPolicy: always
          ports:
```

```
# Service.yaml | Generated by Fleet, yarr!
apiVersion: v1
kind: Service
metadata:
  name: {{.Name}}
  labels:
    app: {{.Name}}
    heritage: Fleet
spec:
  type: ClusterIP
  ports:
    - port: {{.SourcePort}}
      targetPort: {{.TargetPort}}
      protocol: {{.Protocol}}
      name: {{.ProtocolName}}
  selector:
    app: {{.Name}}
```



atlas 

Fleet Atlas consisting of repository per restaurant location



 Global 

Filter by name...


Last created 

New project 


  **90005.cluster.riot-dev.cfadevelop.com**
90005

 0 
6 days ago

  **00123.cluster.riot.cfahome.com**
00123

 0 
6 days ago

  **90024.cluster.riot-dev.cfadevelop.com**
90024



 0 
3 days ago

Application Configuration

HTTP POST Request

```
{
  "name": "auth-server-edge-3",
  "replicas": 4,
  "namespace": "default",
  "image": "docker.somedomain.com/auth-server-edge:1.0",
  "clusters": [
    "90005.cluster.riot-dev.cfadevelop.com",
    "90024.cluster.riot-dev.cfadevelop.com"
  ]
}
```

K8s config example

 deployment.yaml 842 Bytes 

```
1  # Deployment.yaml | Source: Generated by Fleet. Yarr!
2  apiVersion: apps/v1beta2
3  kind: Deployment
4  metadata:
5    namespace: default
6    name: auth-server-edge-3
7    labels:
8      app: auth-server-edge
9      heritage: Fleet
10 spec:
11   replicas: 4
12   selector:
13     matchLabels:
14       app: auth-server-edge-3
15   template:
16     metadata:
17       labels:
18         app: auth-server-edge-3
19         deploymentHash: 381edeab-78cd-11e8-bc2d-6a00024ede00
20     spec:
21       containers:
22         - name: auth-server-edge-3
23           image: docker.somedomain.com/auth-server-edge:1.0
24           imagePullPolicy: always
25           ports:
```

deployment id 381edeab-78cd-11e8-bc2d-6a00024ede00

parent 2e5aebf9  master

Showing 2 changed files ▾ with 4 additions and 4 deletions

Hide whitespace changes

Inline

Side-by-side

auth-server-edge-3/deployment.yaml 



View file @ 96d9e763

@@ -8,7 +8,7 @@ metadata:

8 app: auth-server-edge

9 heritage: Fleet

10 spec:

11 - replicas: 3

12 selector:

13 matchLabels:

14 app: auth-server-edge-3

@@ -16,7 +16,7 @@ spec:

16 metadata:

17 labels:

18 app: auth-server-edge-3

19 - deploymentHash: db312e36-78cb-11e8-842c-6a00024ede00

20 spec:

21 containers:

22 - name: auth-server-edge-3

@@ -8,7 +8,7 @@ metadata:

8 app: auth-server-edge

9 heritage: Fleet

10 spec:

11 + replicas: 4

12 selector:

13 matchLabels:

14 app: auth-server-edge-3

@@ -16,7 +16,7 @@ spec:

16 metadata:

17 labels:

18 app: auth-server-edge-3

19 + deploymentHash: 381edeab-78cd-11e8-bc2d-6a00024ede00

20 spec:

21 containers:

22 - name: auth-server-edge-3

auth-server-edge-3/metadata.txt 



View file @ 96d9e763

1 - guid=db312e36-78cb-11e8-842c-6a00024ede00

1 + guid=381edeab-78cd-11e8-bc2d-6a00024ede00

Overview

Items

Metrics

Alarms

Capacity

Indexes

Global Tables

Backups

Triggers

Access control

Tags

Create item

Actions



Scan: [Table] dev.fleet.deployment-status: deployHash, gi...

Viewing 1 to 6 items

Scan

[Table] dev.fleet.deployment-status: deployHash, gitHash

+ Add filter

Start search

<input type="checkbox"/>	deployHash	gitHash	cluster	status
<input type="checkbox"/>	51221086-78b1-11e8-bef1-6a	7456de7402a414a204faabf2d9d5d0dba605d917	90024.cluster.riot-dev.cfadevelop.com	Complete
<input type="checkbox"/>	51221086-78b1-11e8-bef1-6a	7da17c5580dcf3318b793c13204afd1eafa3de18	90005.cluster.riot-dev.cfadevelop.com	Complete
<input type="checkbox"/>	93ebd42b-775a-11e8-af36-98	265515e9ae6efff38d541e1721265b8d56465362	90024.cluster.riot-dev.cfadevelop.com	CrashLoopBackoff
<input type="checkbox"/>	93ebd42b-775a-11e8-af36-98	e15d5a8892042f566314c540d50d694bbfbe950c	90005.cluster.riot-dev.cfadevelop.com	Pending
<input type="checkbox"/>	381edeab-78cd-11e8-bc2d-6a	36c657f7d79b468e668944b0bb867fbd2f92701	90005.cluster.riot-dev.cfadevelop.com	Pending
<input type="checkbox"/>	381edeab-78cd-11e8-bc2d-6a	96d9e76354b295825e12dd9b604342aff777d7b7	90024.cluster.riot-dev.cfadevelop.com	Pending

Where you can find us



www.linkedin.com/in/brian-chambers
www.linkedin.com/in/calebrhurd

Medium

<https://medium.com/@cfatechblog>



<https://github.com/chick-fil-a>



@brianchambers21
@calebrhurd

