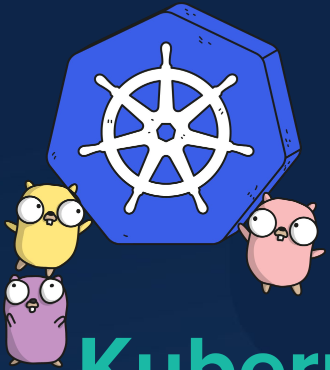
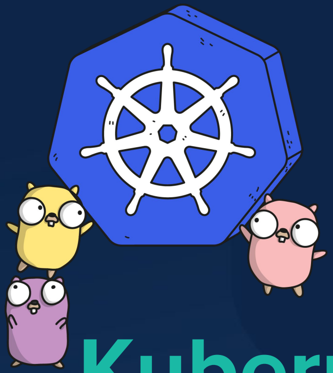


Kubernetes for n00bs

Paul Czarkowski

@pczarkowski

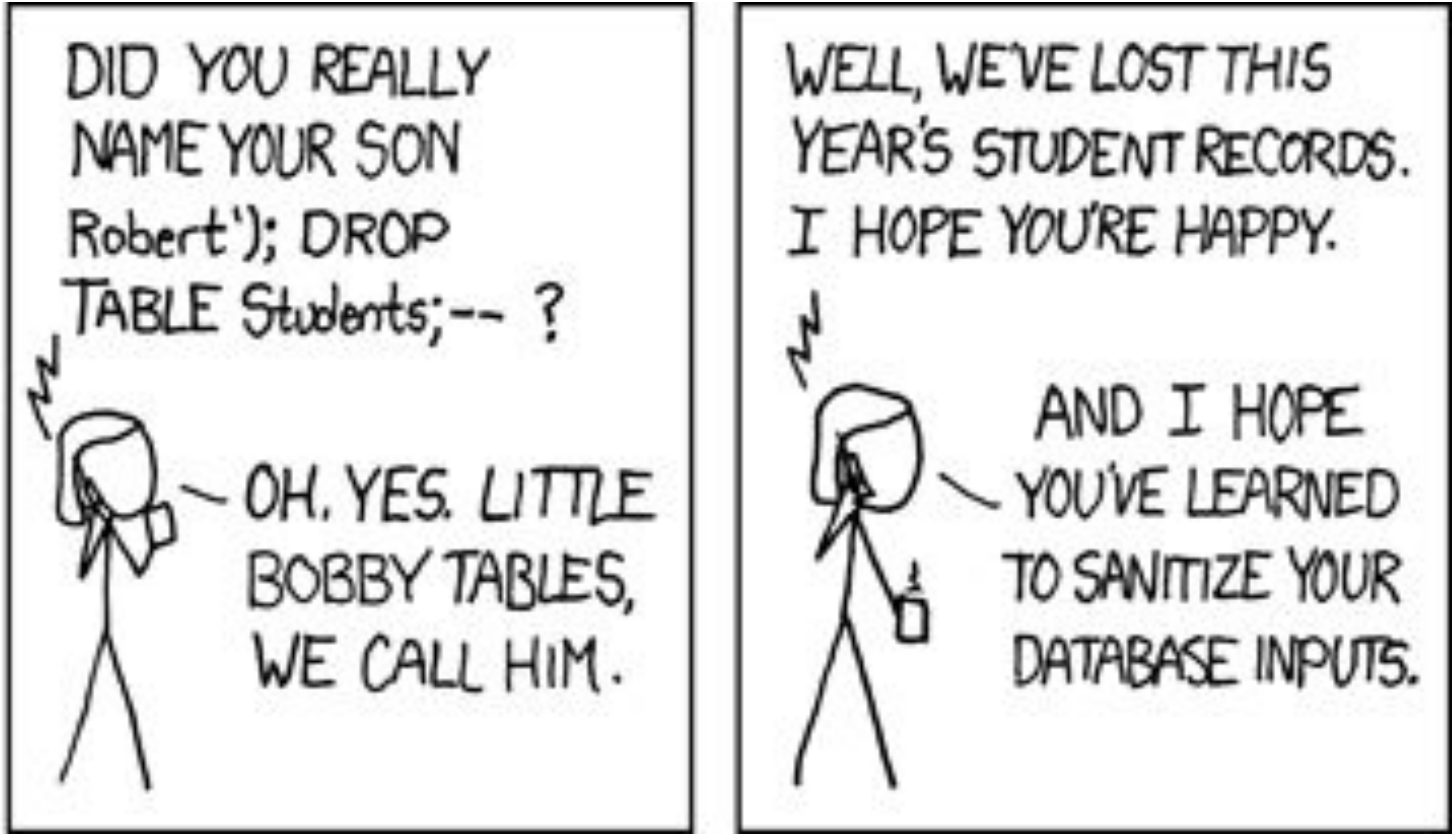


Kubernetes for n00bs

Paul Czarkowski-Golejewski

@pczarkowski

Pivotal


DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

https://xkcd.com/327/

I'M AN
DEVELOPER ADVOCATE
imgflip.com

Agenda

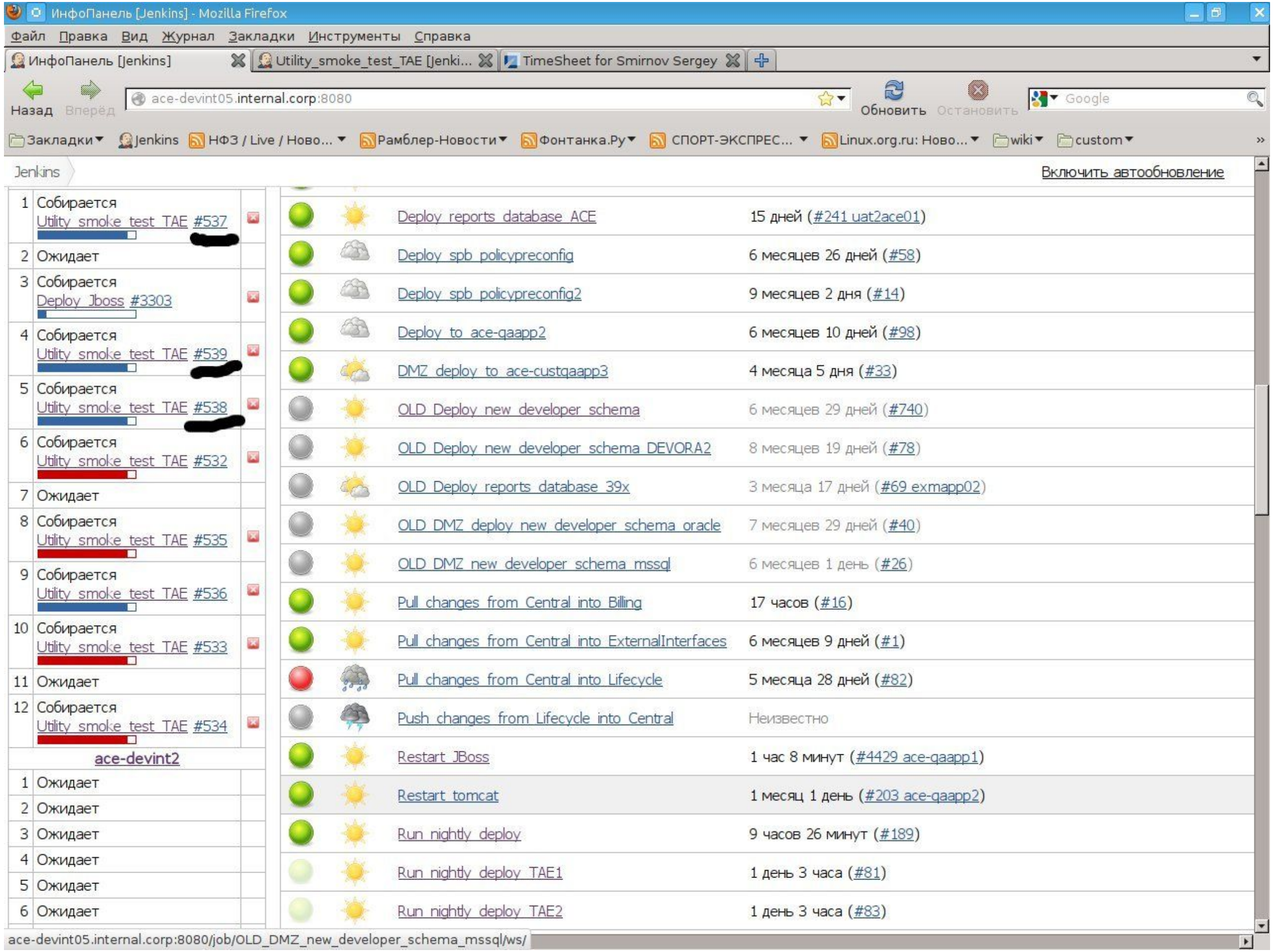
- Who I Am ✓
- Deploying Apps
- What is Kubernetes
- Demo
- Q & A

Deploying Apps

redhat 5.1
LINUX
redhat
LINUX
INSTALLATION GUIDE
LINUX
LINUX

3com
10/100 Managed
Network Interface Card
Simple, Reliable Network Connectivity
for Desktop PCs
• Intelligent management features
• Excellent performance
• Outstanding reliability
• Simple installation and configuration
3C905CX-TX-M
3com
Installation CD
16-0764-002

```go
package main

...
...

func main() {
   fmt.Println("starting hello world app")
   healthHandler := health.NewHandler()
   http.Handle("/health/", healthHandler)
   http.HandleFunc("/", serve)
   http.ListenAndServe(":8080", nil)
}

...
...
```



ИнфоПанель [Jenkins] - Mozilla Firefox

1	Собирается Utility_smoke_test_TAE #537 [redacted]	✖
2	Ожидает	
3	Собирается Deploy_Jboss #3303	✖
4	Собирается Utility_smoke_test_TAE #539 [redacted]	✖
5	Собирается Utility_smoke_test_TAE #538 [redacted]	✖
6	Собирается Utility_smoke_test_TAE #532	✖
7	Ожидает	
8	Собирается Utility_smoke_test_TAE #535	✖
9	Собирается Utility_smoke_test_TAE #536	✖
10	Собирается Utility_smoke_test_TAE #533	✖
11	Ожидает	
12	Собирается Utility_smoke_test_TAE #534	✖
	ace-devint2	
1	Ожидает	
2	Ожидает	
3	Ожидает	
4	Ожидает	
5	Ожидает	
6	Ожидает	

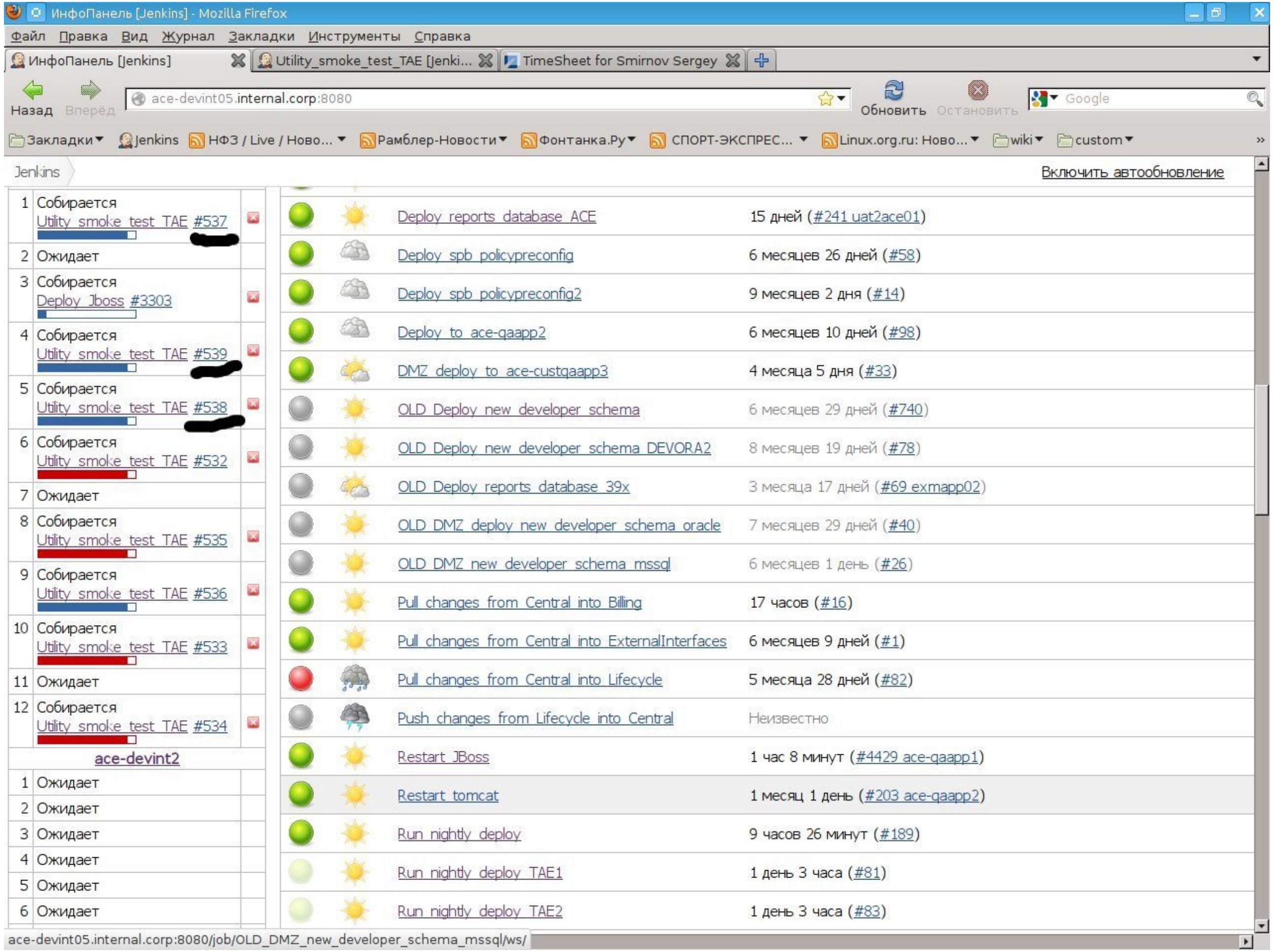
🟢	☀	Deploy_reports_database_ACE	15 дней (#241 uat2ace01)
🟢	☁	Deploy_spb_policypreconfig	6 месяцев 26 дней (#58)
🟢	☁	Deploy_spb_policypreconfig2	9 месяцев 2 дня (#14)
🟢	☁	Deploy_to_ace-qaapp2	6 месяцев 10 дней (#98)
🟢	⛅	DMZ_deploy_to_ace-custqaapp3	4 месяца 5 дня (#33)
⚪	☀	OLD_Deploy_new_developer_schema	6 месяцев 29 дней (#740)
⚪	☀	OLD_Deploy_new_developer_schema_DEVORA2	8 месяцев 19 дней (#78)
⚪	⛅	OLD_Deploy_reports_database_39x	3 месяца 17 дней (#69 exmapp02)
⚪	☀	OLD_DMZ_deploy_new_developer_schema_oracle	7 месяцев 29 дней (#40)
⚪	☀	OLD_DMZ_new_developer_schema_mssql	6 месяцев 1 день (#26)
🟢	☀	Pull_changes_from_Central_into_Billing	17 часов (#16)
🟢	☀	Pull_changes_from_Central_into_ExternalInterfaces	6 месяцев 9 дней (#1)
🔴	🌧	Pull_changes_from_Central_into_Lifecycle	5 месяца 28 дней (#82)
⚪	⛈	Push_changes_from_Lifecycle_into_Central	Неизвестно
🟢	☀	Restart_JBoss	1 час 8 минут (#4429 ace-qaapp1)
🟢	☀	Restart_tomcat	1 месяц 1 день (#203 ace-qaapp2)
🟢	☀	Run_nightly_deploy	9 часов 26 минут (#189)
⚪	☀	Run_nightly_deploy_TAE1	1 день 3 часа (#81)
⚪	☀	Run_nightly_deploy_TAE2	1 день 3 часа (#83)

ace-devint05.internal.corp:8080/job/OLD_DMZ_new_developer_schema_mssql/ws/

```yaml
- name: install ntp
  package:
    name: ntp

- name: configure ntp
  template:
    src: ntp.conf
    dest: /etc/ntp.conf
  notify: restart ntp

- name: start ntp
  service:
    name: ntp
    state: started
```

```ruby
# -*- mode: ruby -*-
# vi: set ft=ruby :
# Vagrantfile API/syntax version. Don't touch unless you know what you're doing!
VAGRANTFILE_API_VERSION = "2"
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
 # https://vagrantcloud.com/ubuntu
 config.vm.box = "ubuntu/xenial64"
 config.vm.network "private_network", type: "dhcp"
 # Forward ports
 config.vm.network "forwarded_port", guest: 8080, host: 8080 # hello world
 config.vm.provider "virtualbox" do |v|
   v.memory = 4096
   v.cpus = 2
 end
```

```
variable "region" {
  default = "europe-west1-d" // We're going to need it in several places in this config
}

provider "google" {
  credentials = "${file("account.json")}"
  project     = "my-project"
  region      = "${var.region}"
}

resource "google_compute_instance" "test" {
  count        = 1 // Adjust as desired
  name         = "test${count.index + 1}" // yields "test1", "test2", etc. It's also the machine's name and hostname
  machine_type = "f1-micro" // smallest (CPU &amp; RAM) available instance
  zone         = "${var.region}" // yields "europe-west1-d" as setup previously. Places your VM in Europe

  disk {
    image = "debian-7-wheezy-v20160301" // the operative system (and Linux flavour) that your machine will run
  }

  network_interface {
    network = "default"
    access_config {
      // Ephemeral IP - leaving this block empty will generate a new external IP and assign it to the machine
    }
  }
}
```



ace-devint05.internal.corp:8080

1	Собирается Utility_smoke_test_TAE #537 [redacted]
2	Ожидает
3	Собирается Deploy_Jboss #3303
4	Собирается Utility_smoke_test_TAE #539 [redacted]
5	Собирается Utility_smoke_test_TAE #538 [redacted]
6	Собирается Utility_smoke_test_TAE #532
7	Ожидает
8	Собирается Utility_smoke_test_TAE #535
9	Собирается Utility_smoke_test_TAE #536
10	Собирается Utility_smoke_test_TAE #533
11	Ожидает
12	Собирается Utility_smoke_test_TAE #534

ace-devint2

1	Ожидает
2	Ожидает
3	Ожидает
4	Ожидает
5	Ожидает
6	Ожидает

●	☀	Deploy_reports_database_ACE	15 дней (#241 uat2ace01)
●	☁	Deploy_spb_policypreconfig	6 месяцев 26 дней (#58)
●	☁	Deploy_spb_policypreconfig2	9 месяцев 2 дня (#14)
●	☁	Deploy_to_ace-qaapp2	6 месяцев 10 дней (#98)
●	⛅	DMZ_deploy_to_ace-custqaapp3	4 месяца 5 дня (#33)
○	☀	OLD_Deploy_new_developer_schema	6 месяцев 29 дней (#740)
○	☀	OLD_Deploy_new_developer_schema_DEVORA2	8 месяцев 19 дней (#78)
○	⛅	OLD_Deploy_reports_database_39x	3 месяца 17 дней (#69 exmapp02)
○	☀	OLD_DMZ_deploy_new_developer_schema_oracle	7 месяцев 29 дней (#40)
○	☀	OLD_DMZ_new_developer_schema_mssql	6 месяцев 1 день (#26)
●	☀	Pull_changes_from_Central_into_Billing	17 часов (#16)
●	☀	Pull_changes_from_Central_into_ExternalInterfaces	6 месяцев 9 дней (#1)
●	⛈	Pull_changes_from_Central_into_Lifecycle	5 месяца 28 дней (#82)
○	⛈	Push_changes_from_Lifecycle_into_Central	Неизвестно
●	☀	Restart_JBoss	1 час 8 минут (#4429 ace-qaapp1)
●	☀	Restart_tomcat	1 месяц 1 день (#203 ace-qaapp2)
●	☀	Run_nightly_deploy	9 часов 26 минут (#189)
○	☀	Run_nightly_deploy_TAE1	1 день 3 часа (#81)
○	☀	Run_nightly_deploy_TAE2	1 день 3 часа (#83)

ace-devint05.internal.corp:8080/job/OLD_DMZ_new_developer_schema_mssql/ws/

```
$ curl http://my-application.com

Hello World!
```

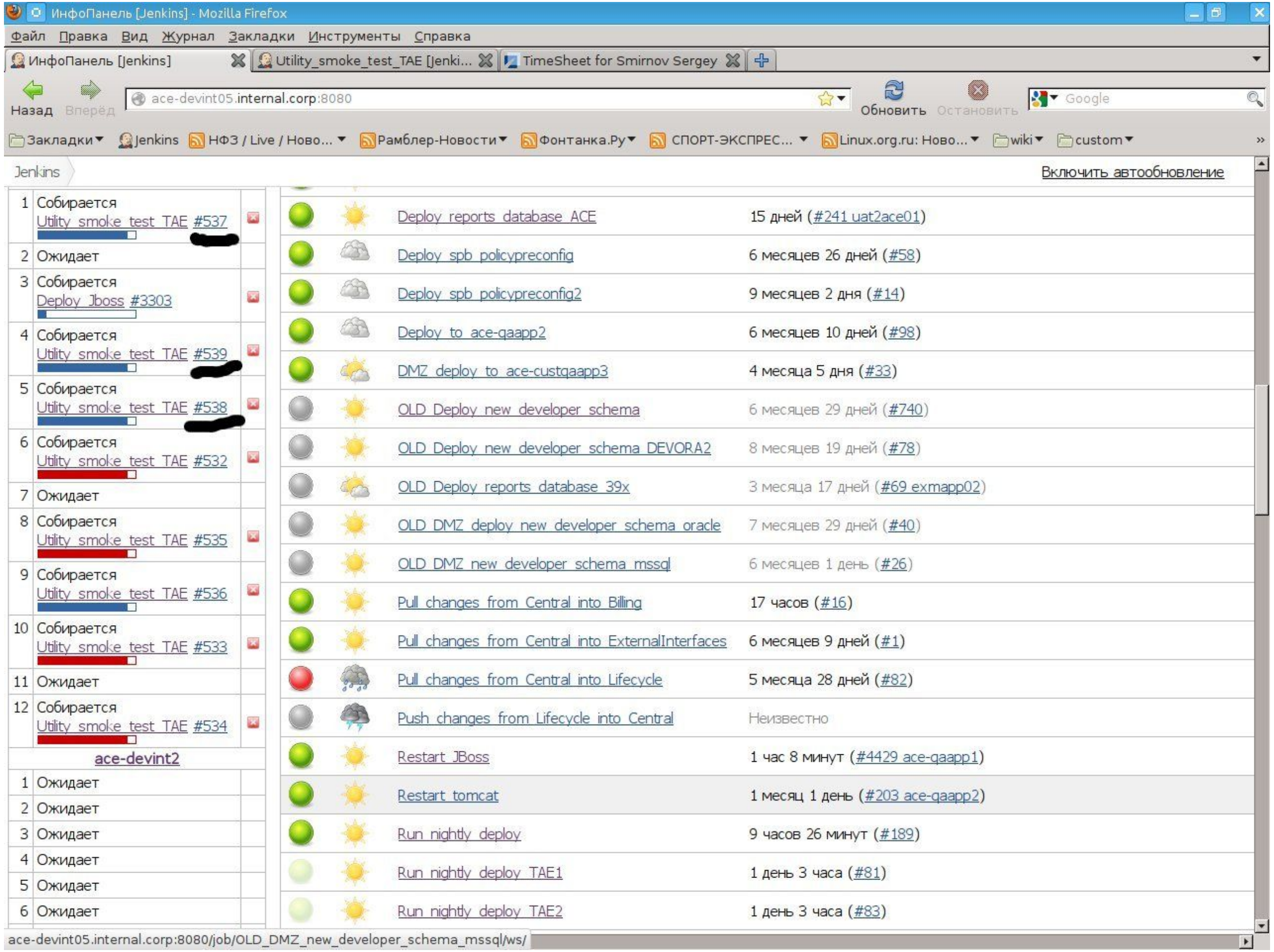
Pivotal

```go
package main

...
...

func main() {
   fmt.Println("starting hello world app")
   healthHandler := health.NewHandler()
   http.Handle("/health/", healthHandler)
   http.HandleFunc("/", serve)
   http.ListenAndServe(":8080", nil)
}

...
...
```

```
FROM golang:1.8
WORKDIR /go/src/app
COPY . .
RUN go-wrapper download
RUN go-wrapper build

EXPOSE 8080
ENTRYPOINT ["/hello-world"]
```

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
 labels:
   app: hello-world
 name: hello-app
spec:
 replicas: 2
 template:
   metadata:
     labels:
       app: hello-world
   spec:
     containers:
     - image: paulczar/hello-world
       name: hello-world
```

```
$ minikube start

$ docker build -t hello-world .

$ kubectl apply -f deployment.yaml

$ curl http://localhost:8080
Hello World!
```



1	Собирается Utility_smoke_test_TAE #537
2	Ожидает
3	Собирается Deploy_Jboss #3303
4	Собирается Utility_smoke_test_TAE #539
5	Собирается Utility_smoke_test_TAE #538
6	Собирается Utility_smoke_test_TAE #532
7	Ожидает
8	Собирается Utility_smoke_test_TAE #535
9	Собирается Utility_smoke_test_TAE #536
10	Собирается Utility_smoke_test_TAE #533
11	Ожидает
12	Собирается Utility_smoke_test_TAE #534
	ace-devint2
1	Ожидает
2	Ожидает
3	Ожидает
4	Ожидает
5	Ожидает
6	Ожидает

Deploy_reports_database_ACE	15 дней (#241 uat2ace01)
Deploy_spb_policypreconfig	6 месяцев 26 дней (#58)
Deploy_spb_policypreconfig2	9 месяцев 2 дня (#14)
Deploy_to_ace-qaapp2	6 месяцев 10 дней (#98)
DMZ_deploy_to_ace-custqaapp3	4 месяца 5 дня (#33)
OLD_Deploy_new_developer_schema	6 месяцев 29 дней (#740)
OLD_Deploy_new_developer_schema_DEVORA2	8 месяцев 19 дней (#78)
OLD_Deploy_reports_database_39x	3 месяца 17 дней (#69 exmapp02)
OLD_DMZ_deploy_new_developer_schema_oracle	7 месяцев 29 дней (#40)
OLD_DMZ_new_developer_schema_mssql	6 месяцев 1 день (#26)
Pull_changes_from_Central_into_Billing	17 часов (#16)
Pull_changes_from_Central_into_ExternalInterfaces	6 месяцев 9 дней (#1)
Pull_changes_from_Central_into_Lifecycle	5 месяца 28 дней (#82)
Push_changes_from_Lifecycle_into_Central	Неизвестно
Restart_JBoss	1 час 8 минут (#4429 ace-qaapp1)
Restart_tomcat	1 месяц 1 день (#203 ace-qaapp2)
Run_nightly_deploy	9 часов 26 минут (#189)
Run_nightly_deploy_TAE1	1 день 3 часа (#81)
Run_nightly_deploy_TAE2	1 день 3 часа (#83)

ace-devint05.internal.corp:8080/job/OLD_DMZ_new_developer_schema_mssql/ws/

What is Kubernetes ?

BRACE YOURSELVES
KUBERNETES IS COMING
memegenerator.net

CMA CGM
eco container
bamboo flooring
CMA CGM
eco container
bamboo flooring
CRONOS
seaco
tex
DOR 40 t
Gulf Port Cranes

What is Docker ?

Popularized Linux Containers

Originated in **2013** by a small PaaS company called DotCloud.

Provided an **easy to use interface** to the [already existing] **Linux Containers**

Linux containers are **like lightweight VMs** that use the built in Linux features instead of virtualizing the hardware.

Most linux containers contain a **single application** rather than a whole operating system.

100s of Containers per server vs a handful of VMs.

Easy to share artifacts called **Images**.

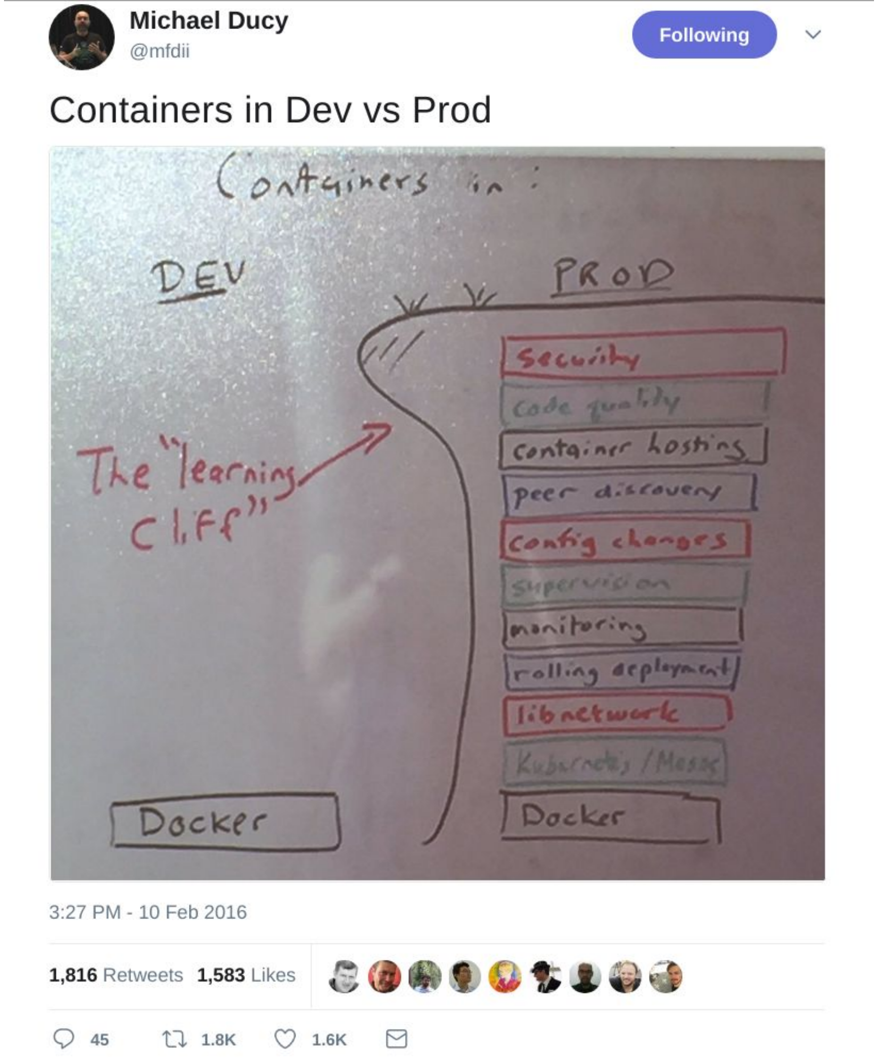
Friendly to Developer and Operator workflows alike.

You tell **Docker** how to build a container image via a fairly simple **Dockerfile** which should generally live alongside your code in your version control system.

A build/test system (ex. Jenkins, Travis, Concourse) should be used to **build and tag images** based on **code changes** and **test results** and push those images to a Registry.

There are a plethora of Registries to choose from and most have a decent UI, Access Controls, and even vuln scanning.

- Docker Registry (either public in form of Docker Hub, or privately run)
- Your Cloud Provider (most public clouds have a Registry service)
- Harbor (extends opensource registry to have enterprise features)
- Artifactory (general purpose artifact repository manager)
- Quay (one of the earliest third party registries)



Michael Ducy
@mfdii

Containers in Dev vs Prod

3:27 PM - 10 Feb 2016

1,816 Retweets 1,583 Likes

45 1.8K 1.6K

What is Kubernetes ?

A container orchestration system.

Greek for “Helmsman” or “Pilot”

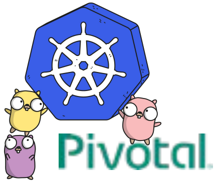
A Borg like platform using Docker as the execution engine originally built by a small team of Google engineers (Joe Beda, Brendan Burns and Craig McLuckie) and Open Sourced in 2014.

GIFEE (Google Infrastructure For Everybody Else).

Production ready! (for some definition of the word production.)

Has a rapid release cycle of a new minor version every three months. (version 1.9 at writing of this)

First project donated to the Cloud Native Compute Foundation.



What is Kubernetes ?

An IaaS for Containers (CaaS)

Abstracts away your infrastructure and provides a **declarative language** for the user to declare their **desired state** and then makes that **actual state**

Linux **containers** instead of **VMs**.

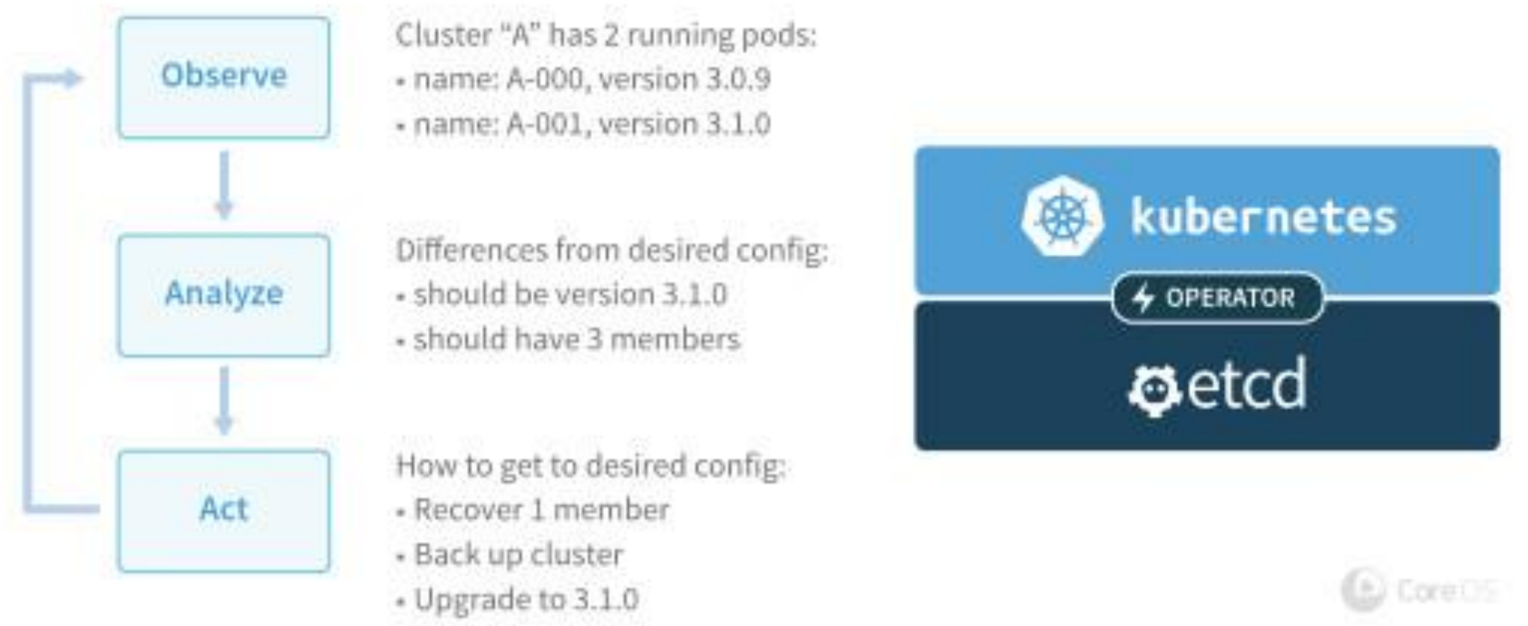
Applications not **Operating Systems**.

Provides a **consistent user experience** for providing **Compute**, **Network** and **Storage** resources and running applications that consume them.

Extends **Compute**, **Network** and **Storage** resources with Controllers that create, monitor and perform actions on them to create higher level abstractions.

Controllers are effectively a infinite loop that interacts with the kubernetes API to ensure the actual state of a resource matches the declared state.

```
#!/bin/bash
while true; do
  count=$(kubectl get pods | grep nginx | wc -l)
  if $count < 5; then
     kubectl run --image=nginx nginx
  fi
  sleep 120
done
```



Observe

Cluster “A” has 2 running pods:
- name: A-000, version 3.0.9
- name: A-001, version 3.1.0

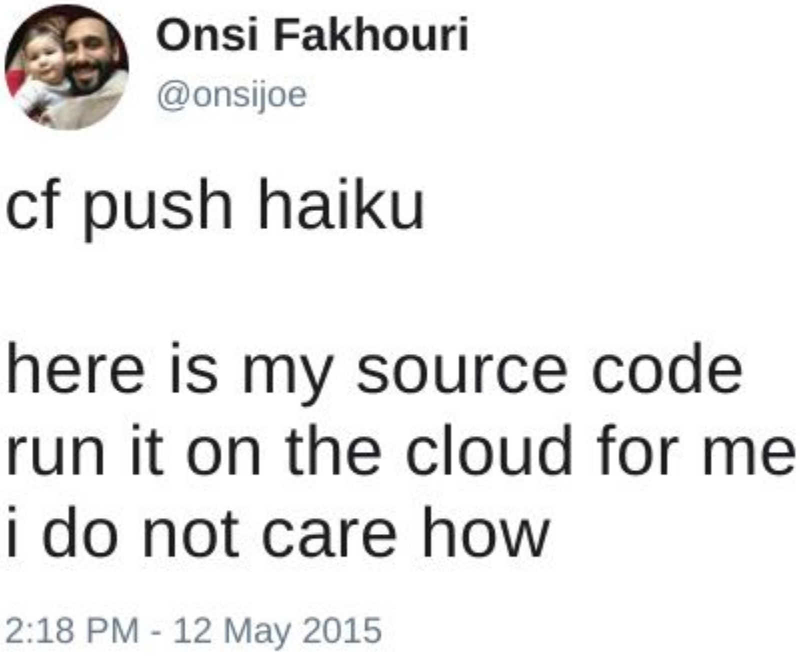
Analyze

Differences from desired config:
- should be version 3.1.0
- should have 3 members

Act

How to get to desired config:
- Recover 1 member
- Back up cluster
- Upgrade to 3.1.0

Pivotal



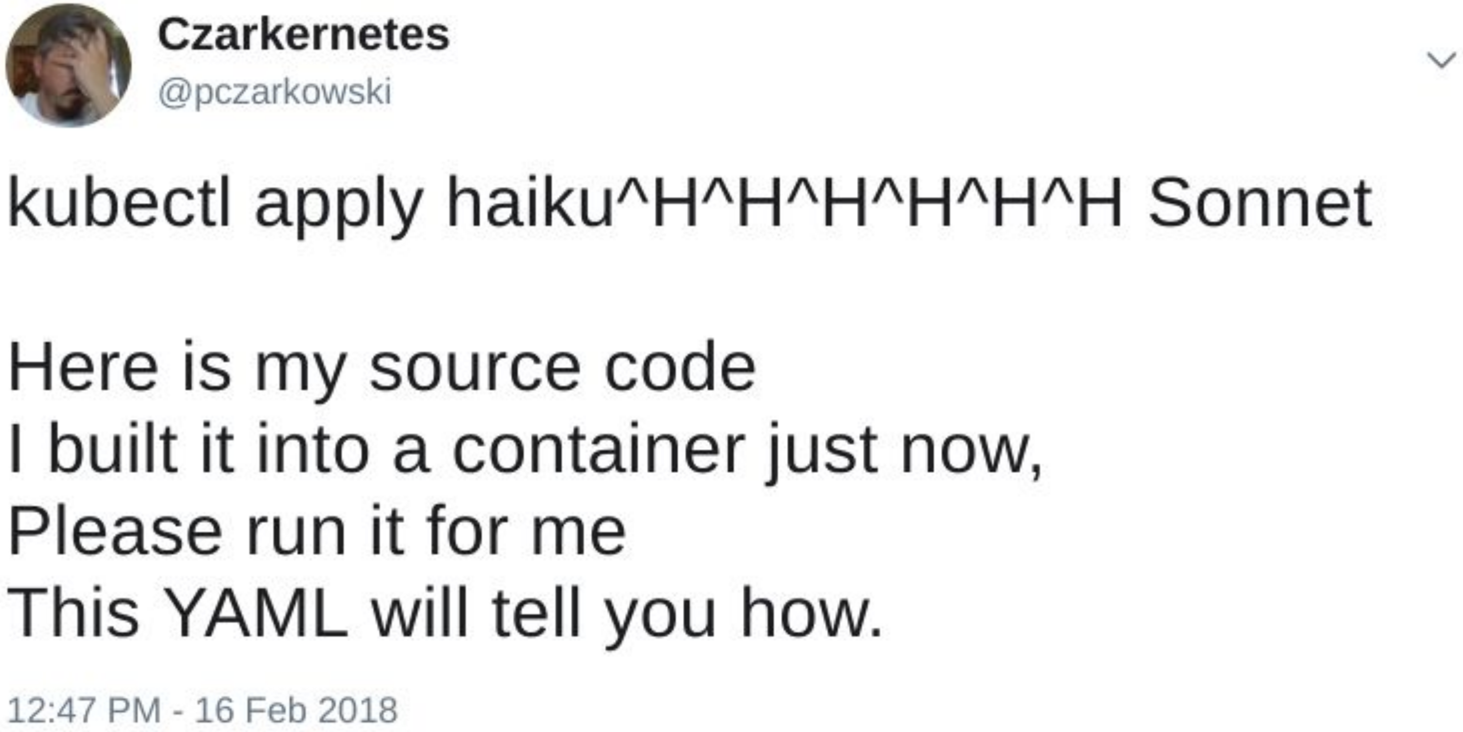
Onsi Fakhouri
@onsijoe

cf push haiku

here is my source code
run it on the cloud for me
i do not care how

2:18 PM - 12 May 2015

https://twitter.com/onsijoe/status/598235841635360768

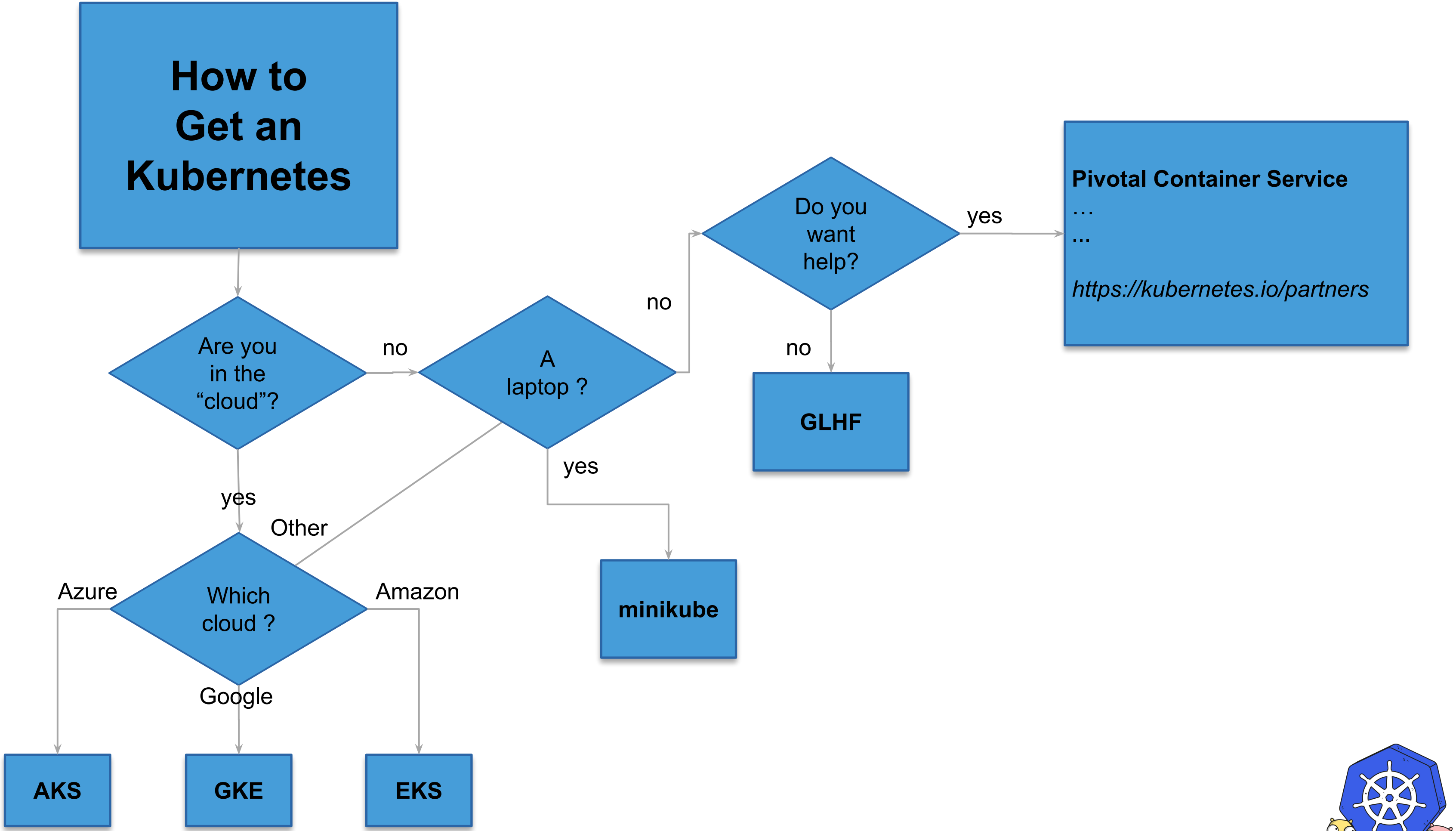


Czarkernetes
@pczarkowski

kubectl apply haiku^H^H^H^H^H^H Sonnet

Here is my source code
I built it into a container just now,
Please run it for me
This YAML will tell you how.

12:47 PM - 16 Feb 2018



How to Get an Kubernetes

Are you in the “cloud”?

yes

Which cloud ?

Azure

AKS

Google

GKE

Amazon

EKS

Other

no

A laptop ?

yes

minikube

no

Do you want help?

no

GLHF

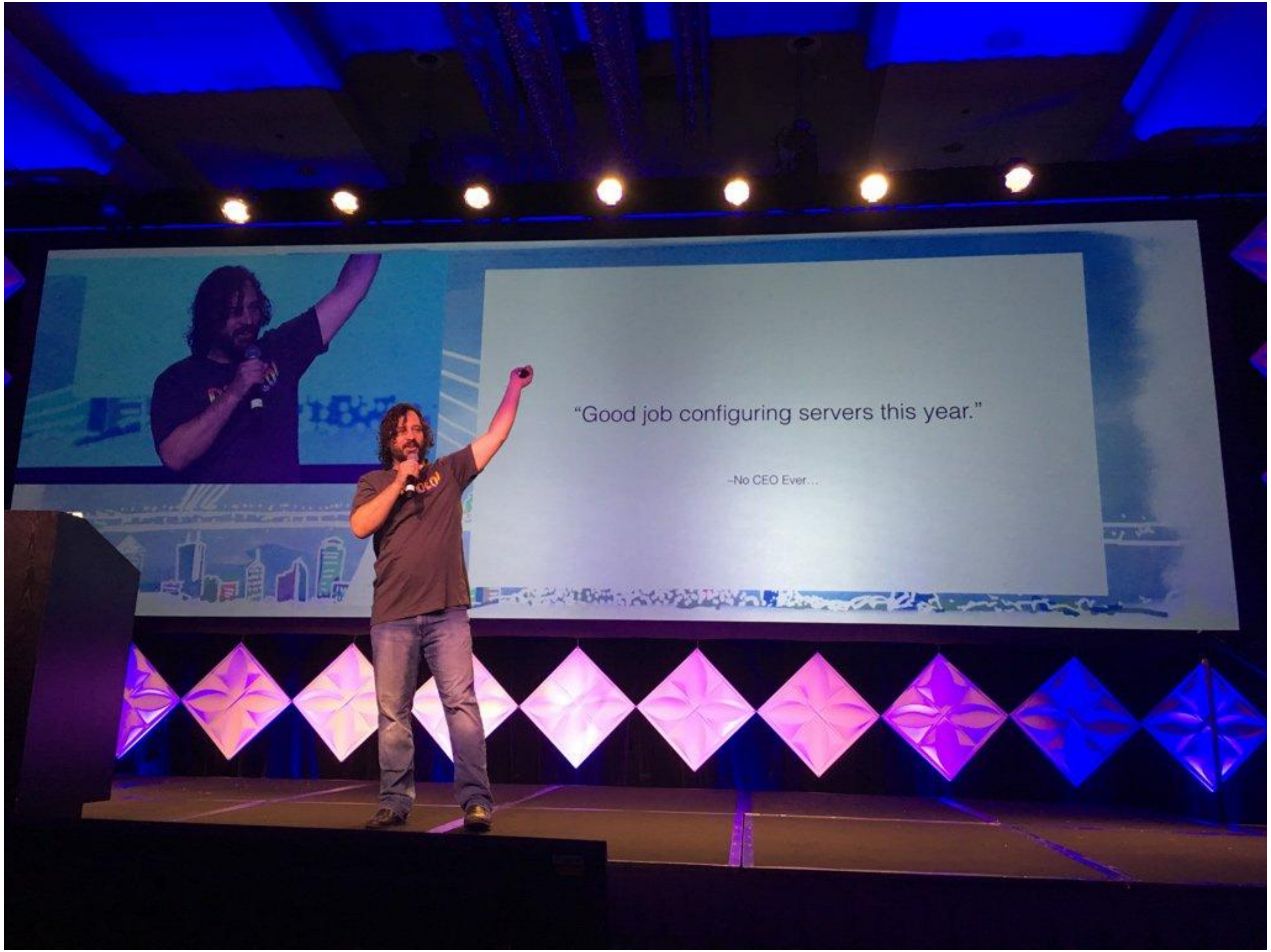
yes

Pivotal Container Service

…

...

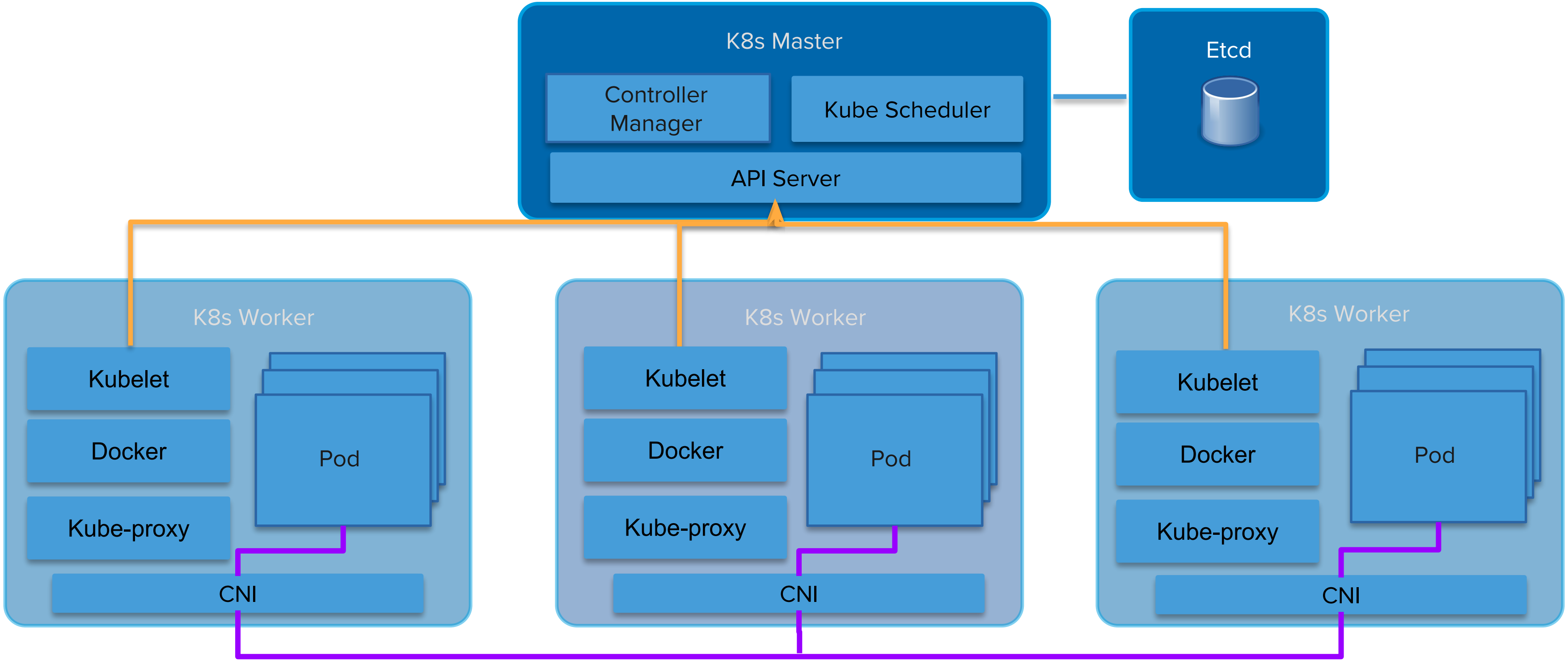
https://kubernetes.io/partners



"Good job configuring servers this year."

–No CEO Ever…

Logical Kubernetes Architecture



Pivotal

Pods (Compute)

one or more application containers that are **tightly coupled**, sharing **network** and **storage**.

Example: a web front-end Pod that consists of an NGINX container and a PHP-FPM container with a shared unix socket and a “init” container to transform their config files based on environment variables.

deployment a controller that ensures a set number of **replicas** of a Pod is running and provides **update and upgrade workflows** for your Pods.

Example: cloud native Node app that scales horizontally and upgrades 2 pods at a time.

statefulset a controller that manages **stateful application** Deployments by providing **sticky identity** for pods and **strict ordering** and **uniqueness**.

Example: Cassandra database. First pod is ‘cassandra-0’ thus all other pods in the set can be told to cluster to ‘cassandra-0’ and it will form a ring, plus the storage will survive pod restarts.

Service (network)

tracks Pods based on metadata and provides connectivity and service discovery (DNS, Env variables) for them.

Published as

ClusterIP (default) exposes service on a **cluster-internal IP**.
NodePort **extends** ClusterIP to expose services on each node’s IP via a **static port**.
LoadBalancer **extends** NodePort to configure a cloud provider’s load balancer using the cloud-controller-manager.

Ingress is a controller that manages an external entity to provide load balancing, SSL termination and name-based virtual hosting to services based on a set of rules.

Volumes (Storage)

Is [effectively] a **Directory**, possibly with data in it, available to **all containers** in a Pod.

Usually **Shares lifecycle** of a Pod (Created when Pod is created, destroyed when Pod is destroyed).

Can be mounted from local disk, or from a network storage device such as a EBS volume, iscsi, NFS, etc.

ConfigMaps/Secrets (user-data)

Provides **key-value pairs** to be injected into a pod much like user-data is injected into a Virtual Machine in the cloud.

Allows you to do **last minute configuration** of applications running on Kubernetes such as setting a database host, or a admin password.

ConfigMaps store values as **strings**, Secrets store them as **byte arrays** (serialized as base64 encoded strings).

Secrets are [currently] **not encrypted** by default. This is likely to **change**.

Can be injected as files in a Volume, or as Environment Variables.

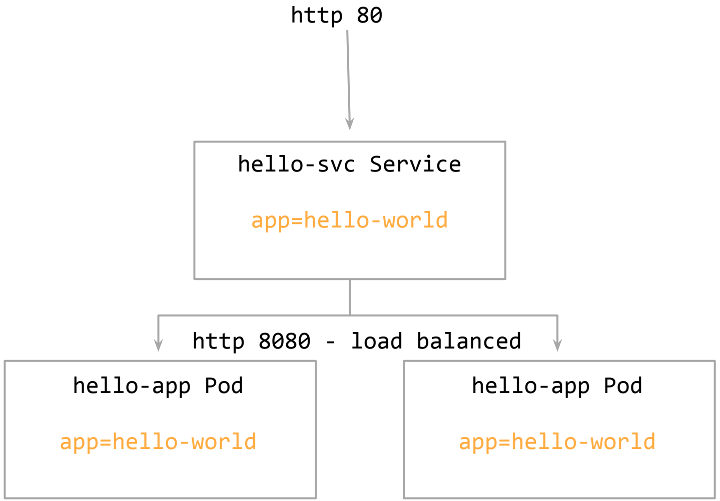
Kubernetes Manifest

```
apiVersion:
kind:
metadata:
spec:
```

Kubernetes Manifest

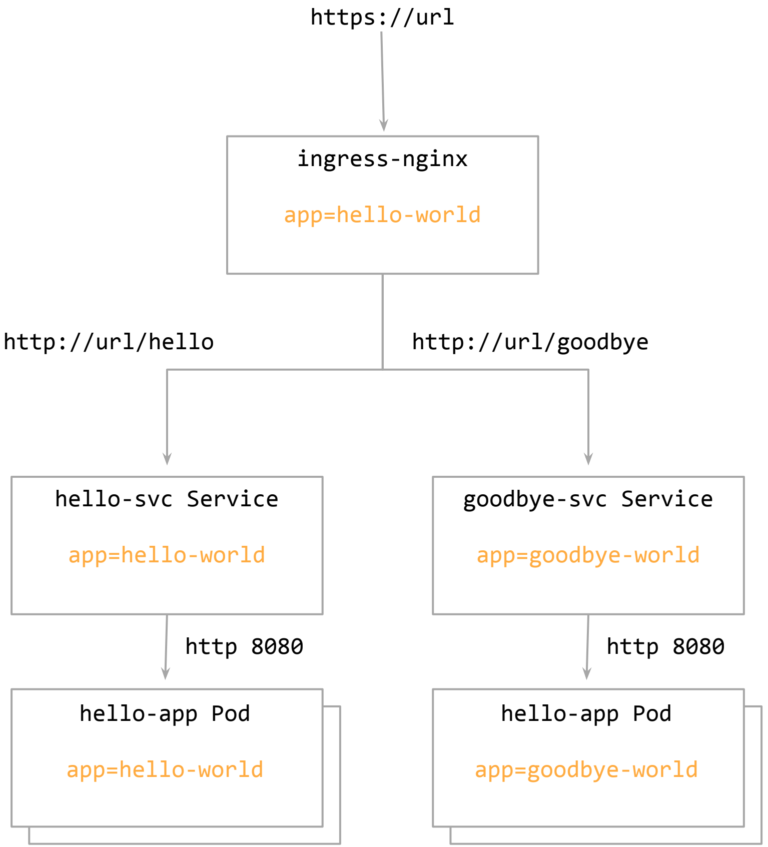
```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
 labels:
   app: hello-world
 name: hello-app
spec:
 replicas: 2
 template:
   metadata:
     labels:
       app: hello-world
   spec:
     containers:
     - image: paulczar/hello-world
       name: hello-world
```

```yaml
apiVersion: v1
kind: Service
metadata:
 name: hello-svc
spec:
 ports:
 - port: 80
   protocol: TCP
   targetPort: 8080
 selector:
   app: hello-world
 type: NodePort
```



Kubernetes Manifest

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: hello-goodbye
spec:
 rules:
 - http:
     paths:
     - path: /hello
       backend:
         serviceName: hello-svc
         servicePort: 80
 - http:
     paths:
     - path: /goodbye
       backend:
         serviceName: goodbye-svc
         servicePort: 81
```


https://url

ingress-nginx

app=hello-world

http://url/hello

http://url/goodbye

hello-svc Service

app=hello-world

goodbye-svc Service

app=goodbye-world

http 8080

http 8080

hello-app Pod

app=hello-world

hello-app Pod

app=goodbye-world

```
$ kubectl apply -f manifests/

deployment "hello-app" created
service "hello-svc" created
deployment "goodbye-app" created
service "goodbye-svc" created
ingress "hello-goodbye" created
```

```
$ curl -k https://$(minikube ip)/hello
Hello World!

$ curl -k https://$(minikube ip)/goodbye
Goodbye Cruel world!
```

Pivotal

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
 name: hello-app
 labels:
   app: hello-world
...
...
   spec:
     containers:
     - image: paulczar/hello-world
       name: hello-world
       volumeMounts:
       - name: config
         mountPath: /etc/hello
     volumes:
       - name: config
         configMap:
           name: hello-cm
```

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
 name: hello-cm
data:
 db: user:pass@host/db
```

```yaml
apiVersion: v1
kind: Service
metadata:
 name: hello-svc
 labels:
   app: hello-world
spec:
 ports:
 - port: 81
   protocol: TCP
   targetPort: 8080
 selector:
   app: hello-world
 type: NodePort
```

Helm is the package manager for Kubernetes

Provides tooling to **template, package, share**, and run Kubernetes manifests for a given application in the form of Charts.

Helm Client a CLI that helps you develop and run Charts.

Tiller Server runs in your cluster and translates Helm Charts into Running Applications.

~ 150 community managed Helm Charts at https://hub.kubeapps.com/

```
.
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── ingress.yaml
│   ├── NOTES.txt
│   └── service.yaml
└── values.yaml
```

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
 name: {{ .Chart.name}}-app
 labels:
   app: {{ .Chart.name}}
...
...
   spec:
     containers:
     - image: paulczar/hello-world
       name: hello-world
       volumeMounts:
       - name: config
         mountPath: /etc/hello
     volumes:
       - name: config
         configMap:
           name: {{ .Chart.name}}-cm
```

```
apiVersion: v1
kind: ConfigMap
metadata:
 name: {{ .Chart.name}}-cm
data:
 db: {{ .Value.db }}
```

```
apiVersion: v1
kind: Service
metadata:
 name: {{ .Chart.name}}-svc
 labels:
   app: {{ .Chart.name}}-world
spec:
 ports:
 - port: {{ .Value.port }}
   protocol: TCP
   targetPort: 8080
 selector:
   app: {{ .Chart.name}}-world
 type: NodePort
```

```
$ helm install --name staging . \
  --set db=’user:pass@staging.mysql/dbname’

$ helm install --name production . \
  --set db=’user:pass@production.mysql/dbname’
```

DEMO

Just Enough Modernization for Kubernetes (JEMFORK)

I. Codebase — One codebase tracked in revision control, many deploys

II. Dependencies — Explicitly declare and isolate dependencies

III. Config — Store config in the environment

IV. Backing Services — Treat backing services as attached resources

V. Build, release, run — Strictly separate build and run stages

VI. Processes — Execute the app as one or more stateless processes

Just Enough Modernization for Kubernetes (JEMFORK)

VII. Port binding — Export services via port binding

VIII. Concurrency — Scale out via the process model

IX. Disposability — Maximize robustness with fast startup and graceful shutdown

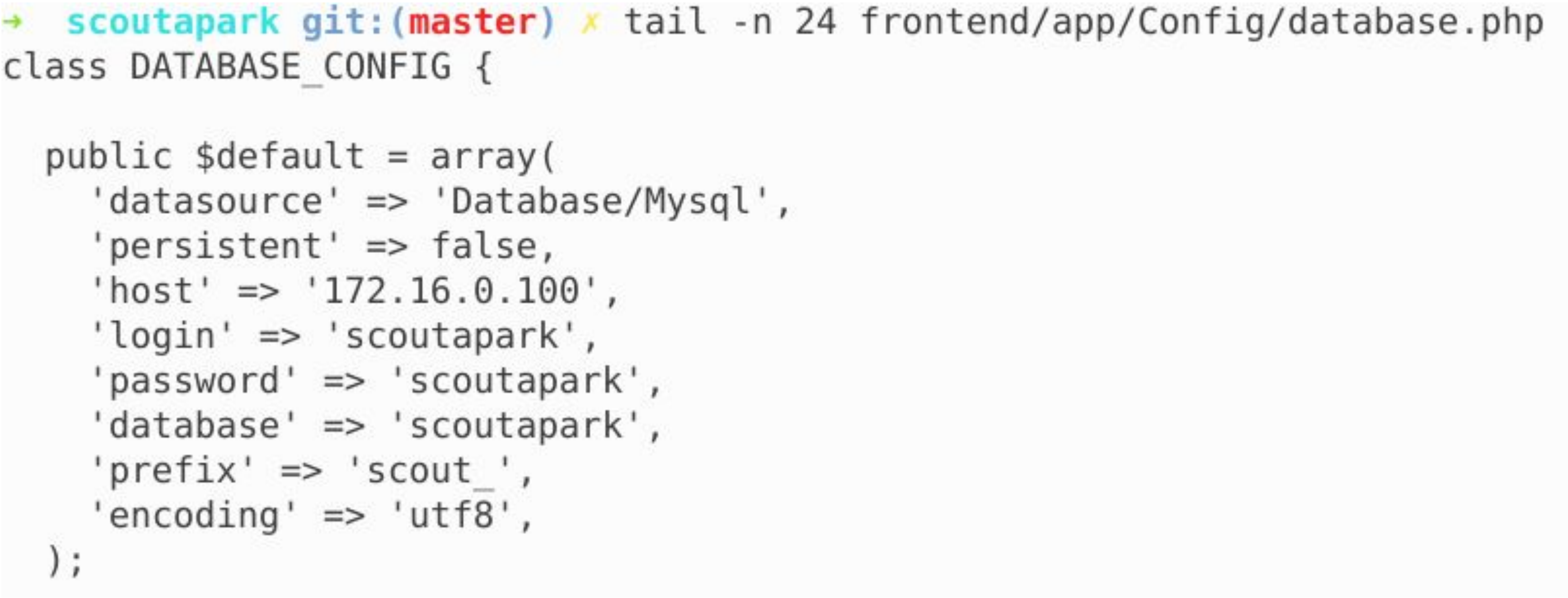
X. Dev/prod parity — Keep development, staging, and production as similar as possible

XI. Logs — Treat logs as event streams

XII. Admin processes — Run admin/management tasks as one-off processes

Just Enough Modernization for Kubernetes (JEMFORK)

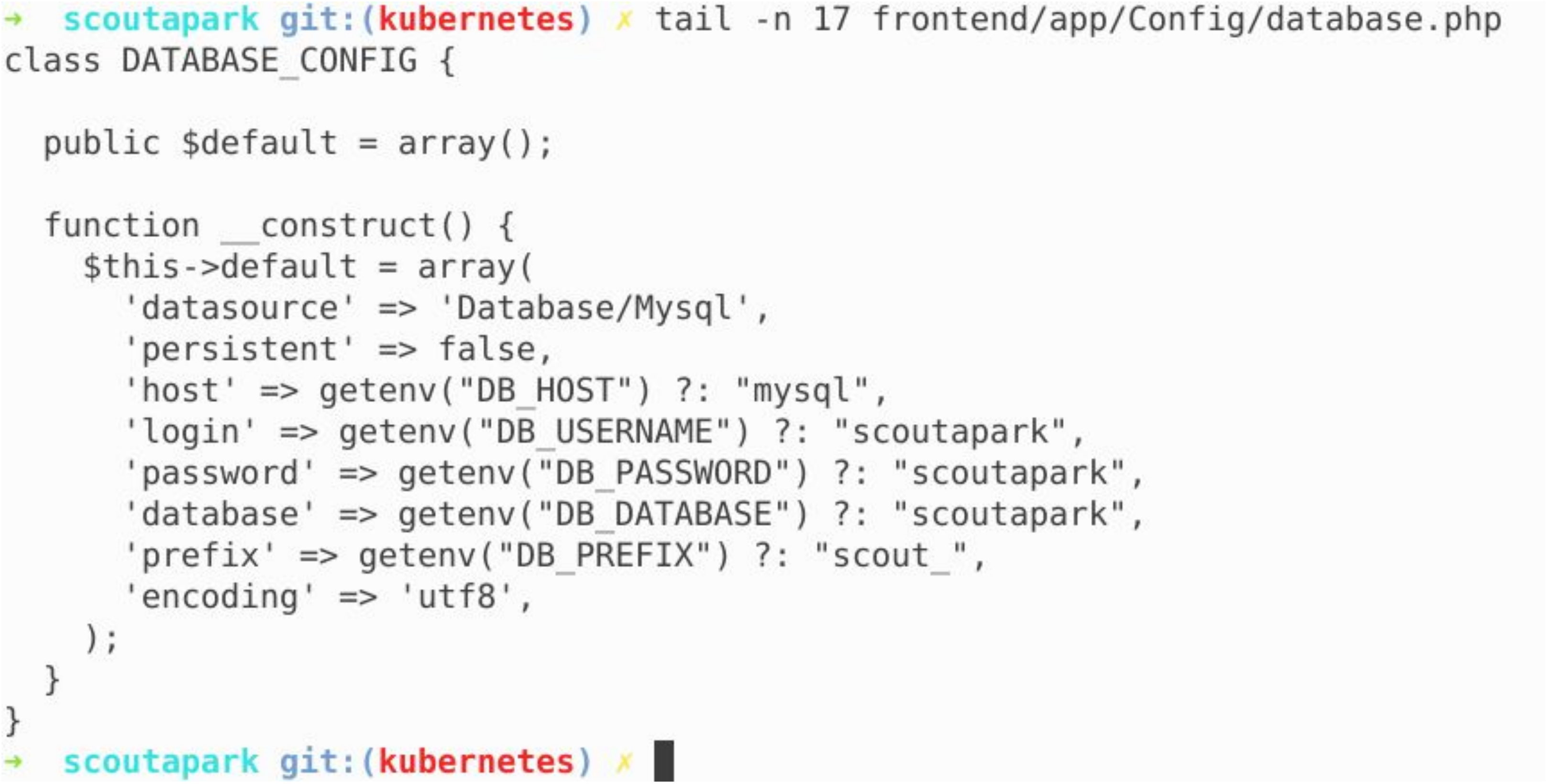
III. Config — Store config in the environment



```
➜  scoutapark git:(master) ✗ tail -n 24 frontend/app/Config/database.php
class DATABASE_CONFIG {

  public $default = array(
    'datasource' => 'Database/Mysql',
    'persistent' => false,
    'host' => '172.16.0.100',
    'login' => 'scoutapark',
    'password' => 'scoutapark',
    'database' => 'scoutapark',
    'prefix' => 'scout_',
    'encoding' => 'utf8',
  );
```

Just Enough Modernization for Kubernetes (JEMFORK)

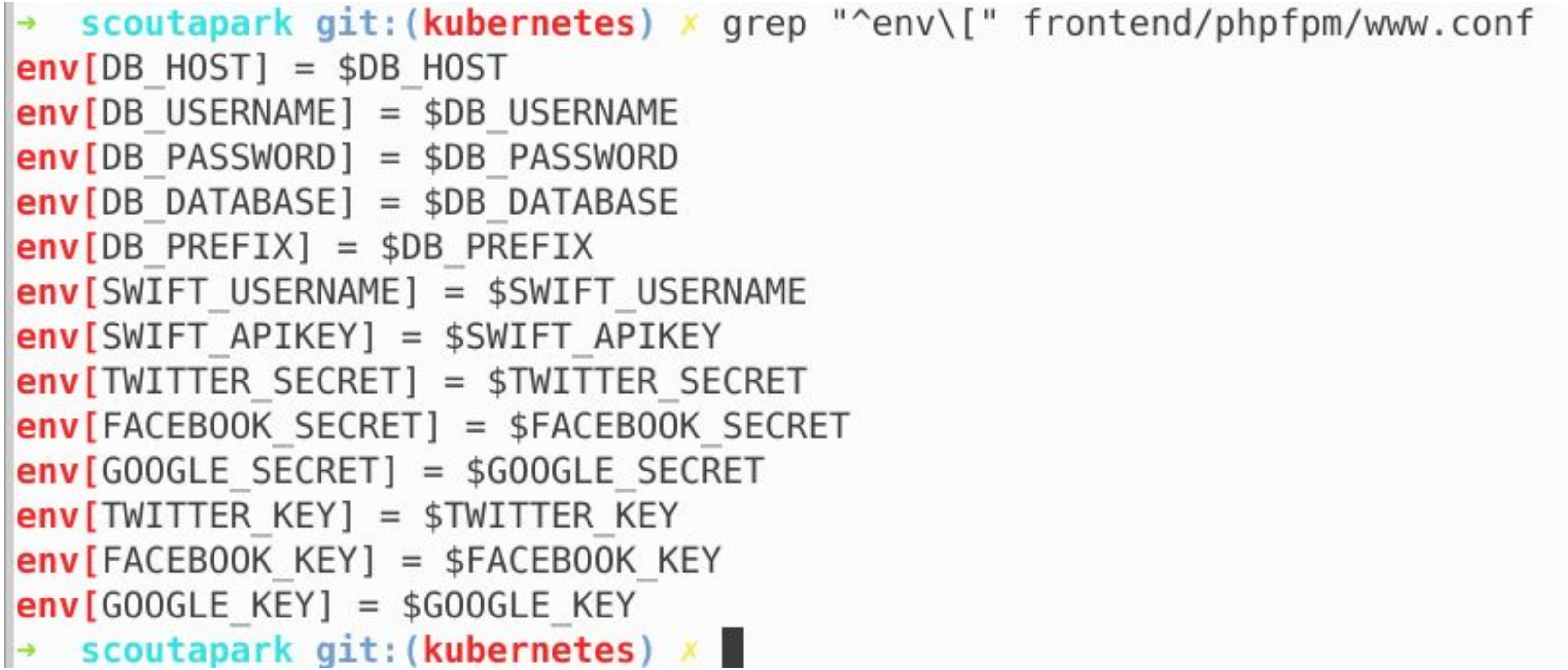
Environment Variables



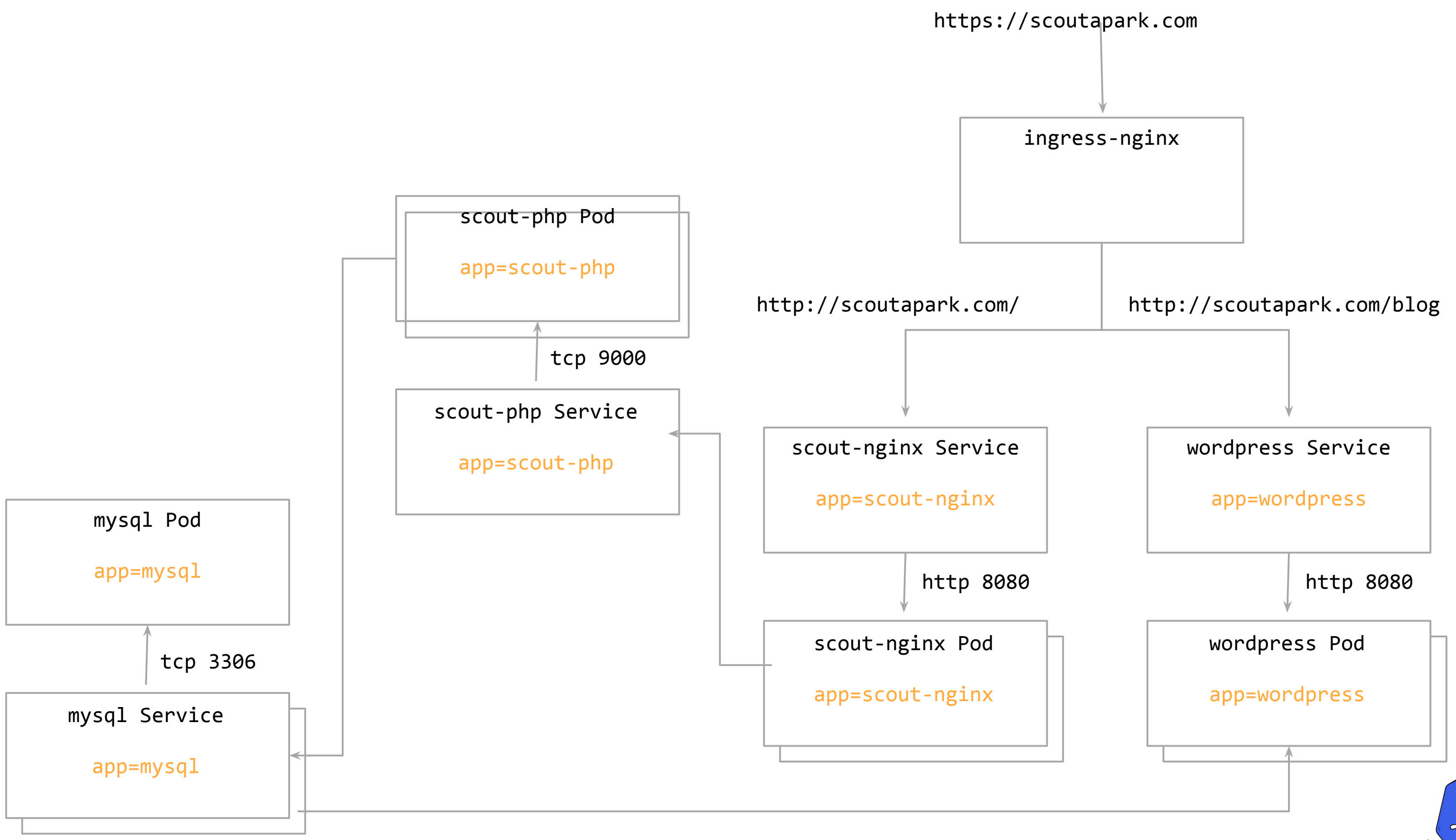
```
➜  scoutapark git:(kubernetes) ✗ tail -n 17 frontend/app/Config/database.php
class DATABASE_CONFIG {

  public $default = array();

  function __construct() {
    $this->default = array(
      'datasource' => 'Database/Mysql',
      'persistent' => false,
      'host' => getenv("DB_HOST") ?: "mysql",
      'login' => getenv("DB_USERNAME") ?: "scoutapark",
      'password' => getenv("DB_PASSWORD") ?: "scoutapark",
      'database' => getenv("DB_DATABASE") ?: "scoutapark",
      'prefix' => getenv("DB_PREFIX") ?: "scout_",
      'encoding' => 'utf8',
    );
  }
}
➜  scoutapark git:(kubernetes) ✗ █
```

Just Enough Modernization for Kubernetes (JEMFORK)

Environment Variables



```
→  scoutapark git:(kubernetes) ✗ grep "^env\[" frontend/phpfpm/www.conf
env[DB_HOST] = $DB_HOST
env[DB_USERNAME] = $DB_USERNAME
env[DB_PASSWORD] = $DB_PASSWORD
env[DB_DATABASE] = $DB_DATABASE
env[DB_PREFIX] = $DB_PREFIX
env[SWIFT_USERNAME] = $SWIFT_USERNAME
env[SWIFT_APIKEY] = $SWIFT_APIKEY
env[TWITTER_SECRET] = $TWITTER_SECRET
env[FACEBOOK_SECRET] = $FACEBOOK_SECRET
env[GOOGLE_SECRET] = $GOOGLE_SECRET
env[TWITTER_KEY] = $TWITTER_KEY
env[FACEBOOK_KEY] = $FACEBOOK_KEY
env[GOOGLE_KEY] = $GOOGLE_KEY
→  scoutapark git:(kubernetes) ✗ 
```


https://scoutapark.com
ingress-nginx
http://scoutapark.com/
http://scoutapark.com/blog
scout-php Pod
app=scout-php
tcp 9000
scout-php Service
app=scout-php
scout-nginx Service
app=scout-nginx
wordpress Service
app=wordpress
http 8080
http 8080
mysql Pod
app=mysql
tcp 3306
mysql Service
app=mysql
scout-nginx Pod
app=scout-nginx
wordpress Pod
app=wordpress

<INSERT DEMO HERE>

Further Reading

Next Steps … Further reading.

- Kubernetes Docs, specifically the tutorials and troubleshooting sectiong
  - https://kubernetes.io/docs/home/
  - https://kubernetes.io/docs/tutorials/kubernetes-basics/
  - https://kubernetes.io/docs/tasks/debug-application-cluster/troubleshooting/

- Writing your first Helm Chart
  - https://medium.com/@pczarkowski/writing-your-first-helm-chart-f3433344f824

- Pivotal's Enterprise Kubernetes Offering
  - https://pivotal.io/platform/pivotal-container-service

- Kelsey Hightower's Kubecon Keynote showing CI/CD pipeline
  - https://www.youtube.com/watch?v=07jq-5VbBVQ

Q & A

Pivotal®

Transforming How The World Builds Software