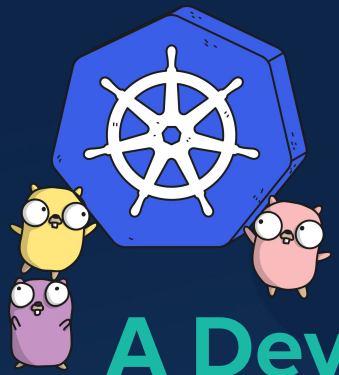


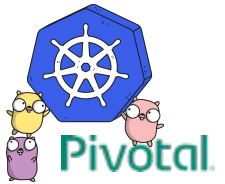
A DevOps Guide to Kubernetes

Paul Czarkowski

@pczarkowski

Pivotal


A DEVOP HAS NO NAME
memegenerator.net


Pivotal

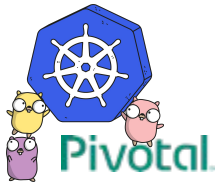
a sprinkle of lime
@moonpolysoft

Following

9:47 AM - 19 Mar 2018

107 Retweets **699** Likes

20 107 699

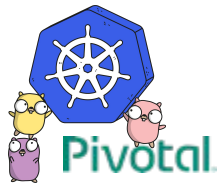


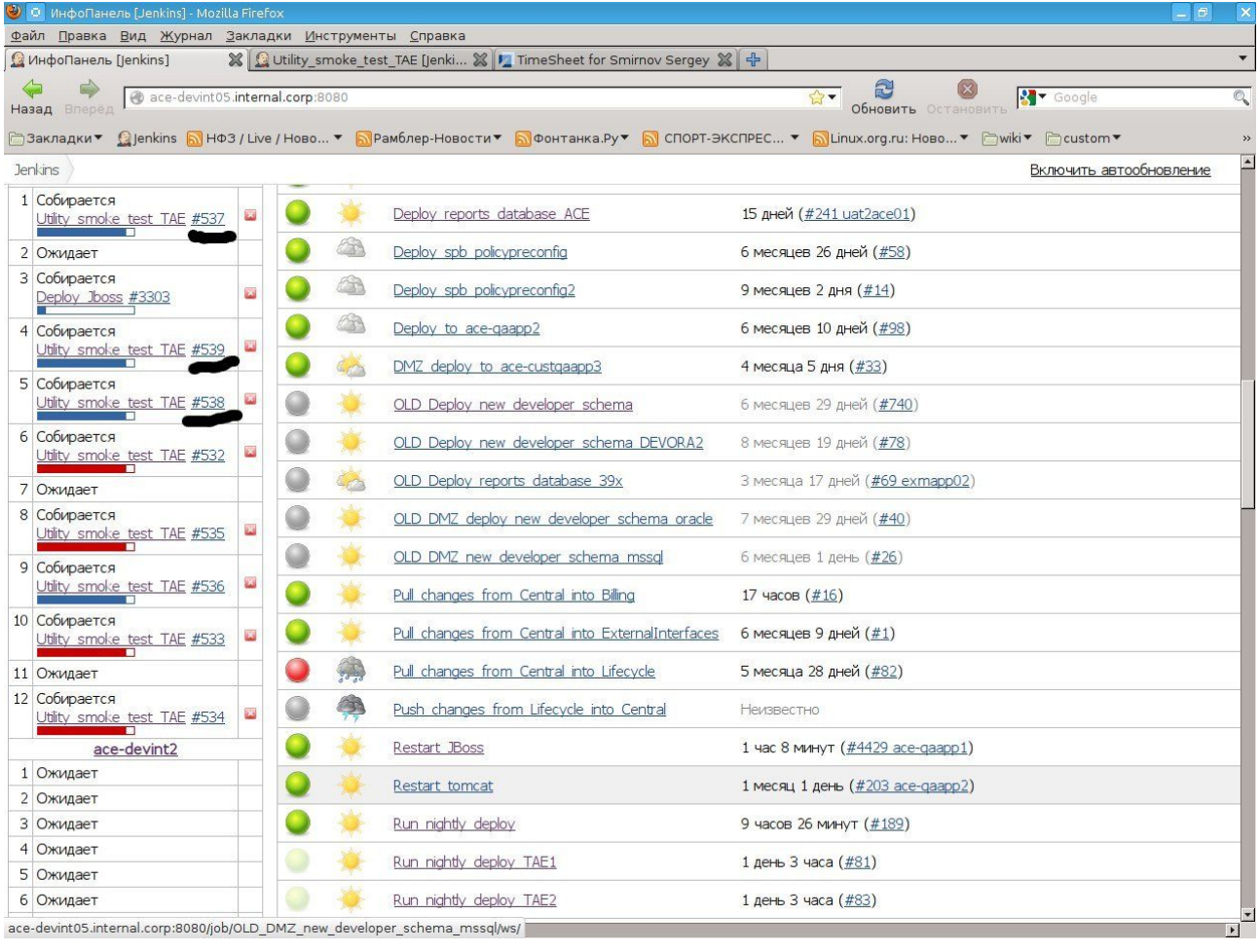

redhat
5.1
LINUX
LINUX
LINUX
LINUX
redhat

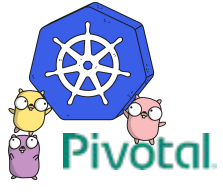

3com
10/100 Managed
Network Interface Card
Simple, Reliable Network Connectivity
for Desktop PCs
• Intelligent management features
• Excellent performance
• Outstanding reliability
• Simple installation and configuration
3CR990-TX-M
3com
Installation CD
16-0764-002

Pivotal

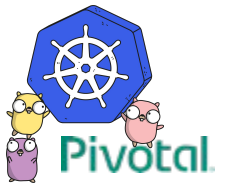
```
package main

...
...

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

...
...
```



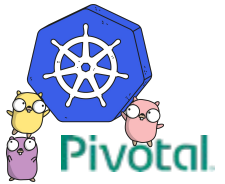

ИнфоПанель [Jenkins] - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
ИнфоПанель [Jenkins]
Utility_smoke_test_TAE [Jenki...
TimeSheet for Smirnov Sergey
Назад
Вперёд
ace-devint05.internal.corp:8080
Обновить
Остановить
Google
Закладки
Jenkins
НФЗ / Live / Ново...
Рамблер-Новости
Фонтанка.Ру
СПОРТ-ЭКСПРЕС...
Linux.org.ru: Ново...
wiki
custom
Jenkins
Включить автообновление
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/


Pivotal

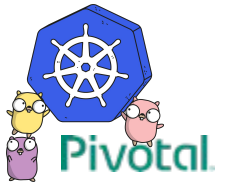
```yaml
- name: install ntp
  package:
    name: ntp

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

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

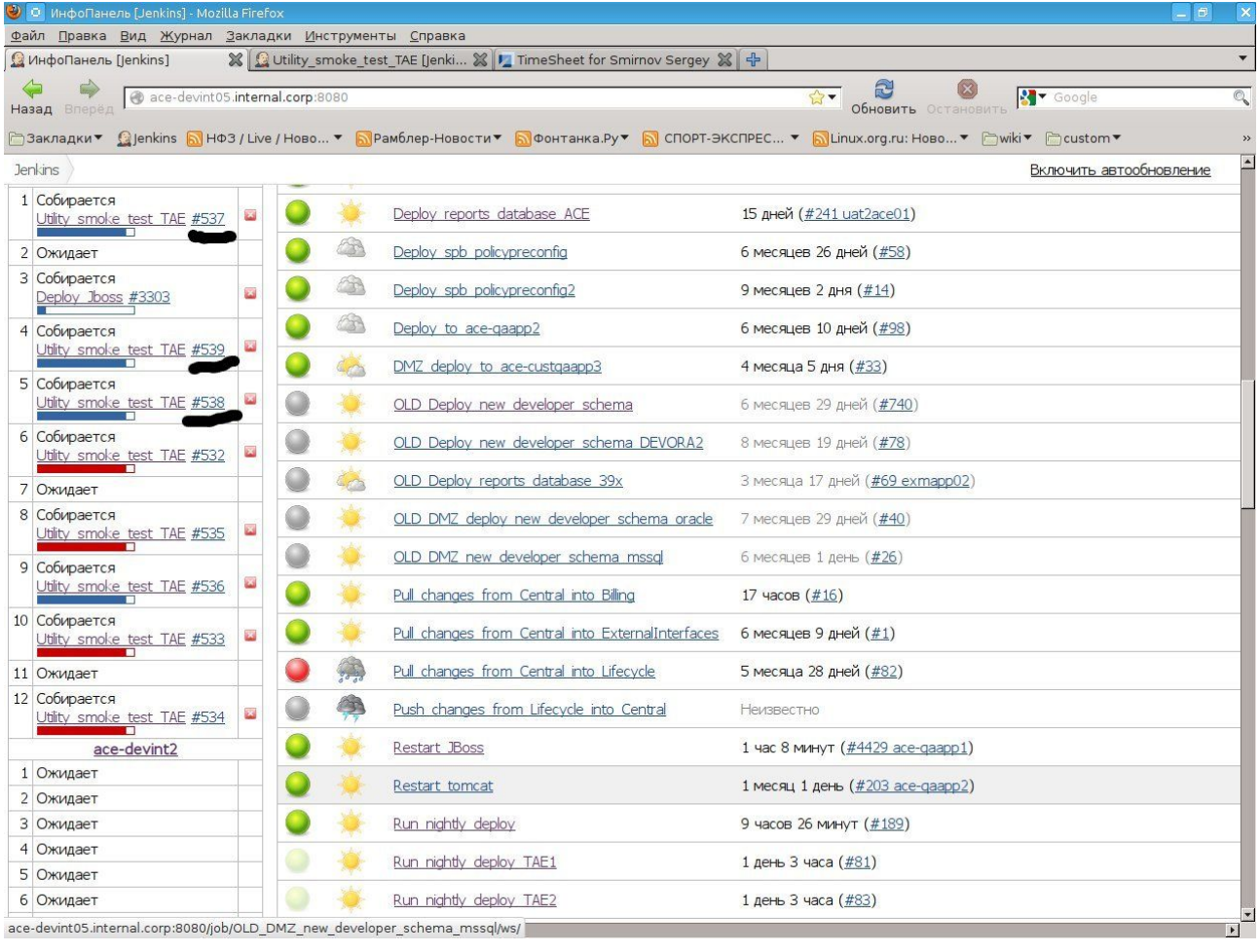


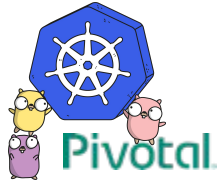
```
# -*- mode: ruby -*-
# vi: set ft=ruby :
# Vagrantfile API/syntax version. Don't touch unless you know what you're doing!
VAGRANTFILE_API_VERSION = "2"
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
 # https://vagrantcloud.com/ubuntu
 config.vm.box = "ubuntu/xenial64"
 config.vm.network "private_network", 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
```



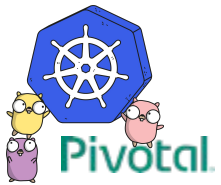
```
variable "region" {
  default = "europe-west1-d" // We're going to need it in several places in this config
}

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

resource "google_compute_instance" "test" {
  count        = 1 // Adjust as desired
  name         = "test${count.index + 1}" // yields "test1", "test2", etc. It's also the machine's name and hostname
  machine_type = "f1-micro" // smallest (CPU &amp;amp; RAM) available instance
  zone         = "${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
    }
  }
}
```

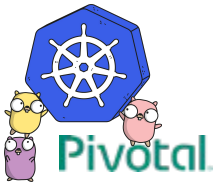



ИнфоПанель [Jenkins] - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
ИнфоПанель [Jenkins]
Utility_smoke_test_TAE [Jenki...
TimeSheet for Smirnov Sergey
Назад
Вперёд
ace-devint05.internal.corp:8080
Обновить
Остановить
Google
Закладки
Jenkins
НФЗ / Live / Ново...
Рамблер-Новости
Фонтанка.Ру
СПОРТ-ЭКСПРЕС...
Linux.org.ru: Ново...
wiki
custom
Jenkins
Включить автообновление
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/

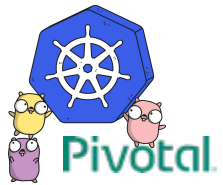

Pivotal

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

Hello World!
```

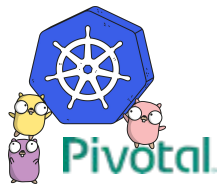



Pivotal

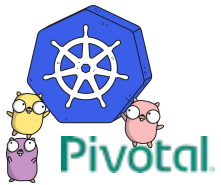
```
package main

...
...

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

...
...
```



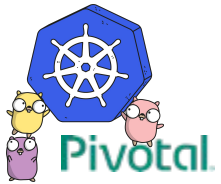
```
FROM golang:1.8
WORKDIR /go/src/app
COPY . .
RUN go-wrapper download
RUN go-wrapper build

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

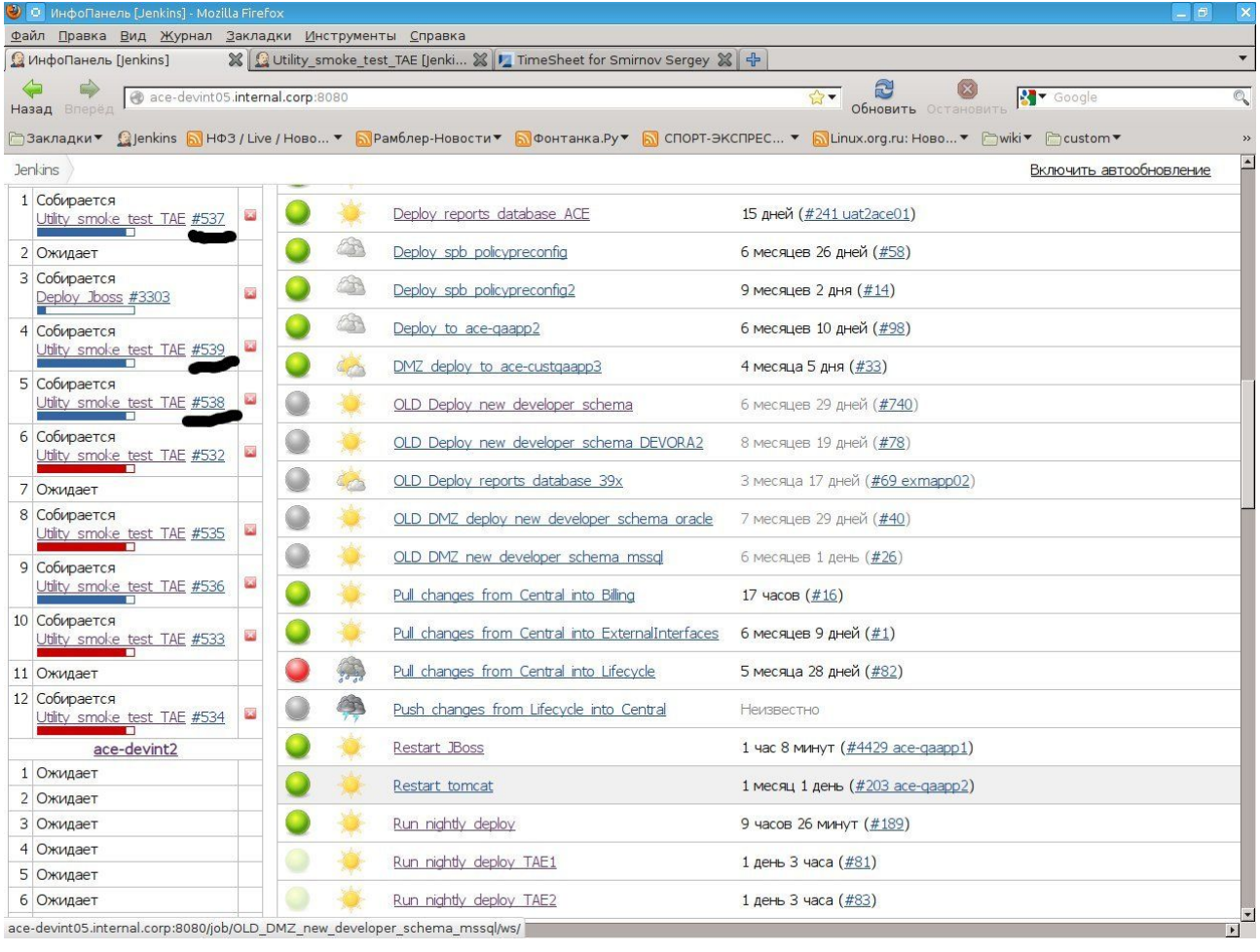


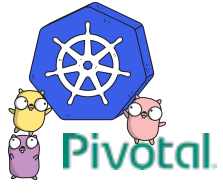
```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
 labels:
   app: hello-world
 name: hello-app
spec:
 replicas: 2
 template:
   metadata:
     labels:
       app: hello-world
   spec:
     containers:
     - image: paulczar/hello-world
       name: hello-world
```

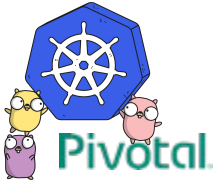


```
$ minikube start

$ docker build -t hello-world .

$ kubectl apply -f deployment.yaml

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




ИнфоПанель [Jenkins] - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
ИнфоПанель [Jenkins]
Utility_smoke_test_TAE [Jenki...
TimeSheet for Smirnov Sergey
Назад
Вперёд
ace-devint05.internal.corp:8080
Обновить
Остановить
Google
Закладки
Jenkins
НФЗ / Live / Ново...
Рамблер-Новости
Фонтанка.Ру
СПОРТ-ЭКСПРЕС...
Linux.org.ru: Ново...
wiki
custom
Jenkins
Включить автообновление
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/


Pivotal

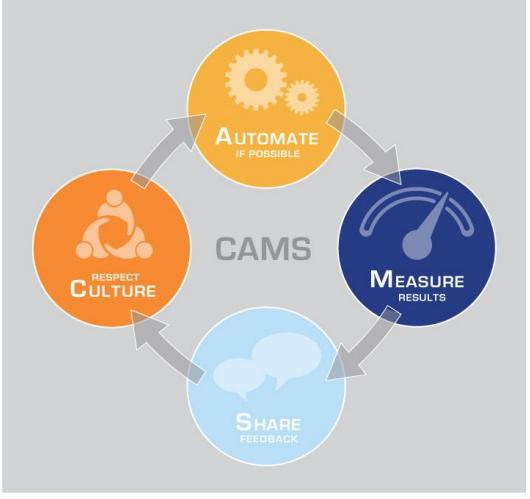

Pivotal

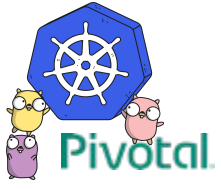
Culture - Increased collaboration between Development and Operations (and the rest of the business) and an attitude of shared responsibility.

Automation - Increases velocity, but just as importantly reduces defects and creates consistency and repeatability.

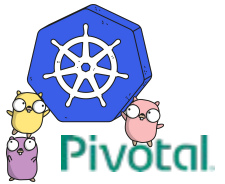
Measurement - Need to measure in order to ensure that improvement is happening.

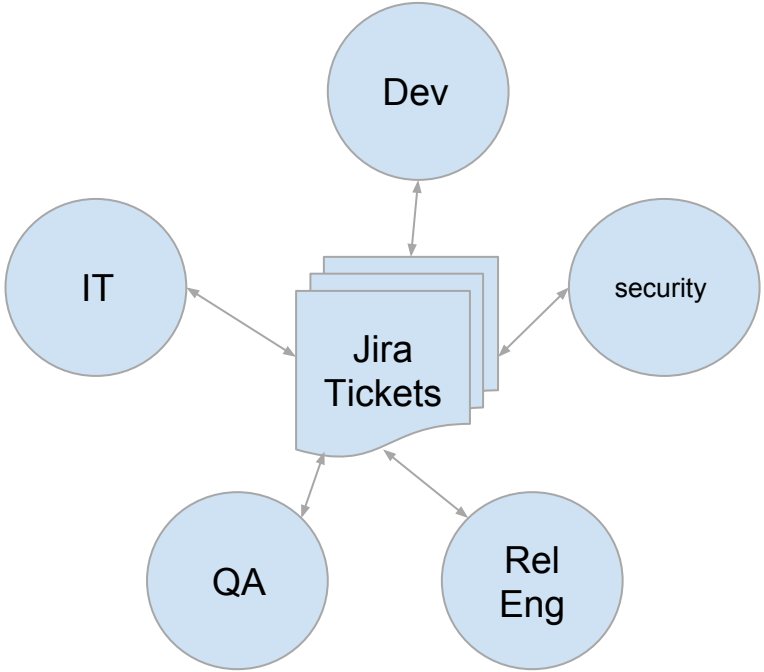
Sharing - As we share tools, discoveries and lessons new opportunities to collaborate will be discovered and duplicate work can be eliminated.

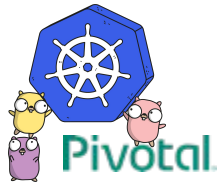


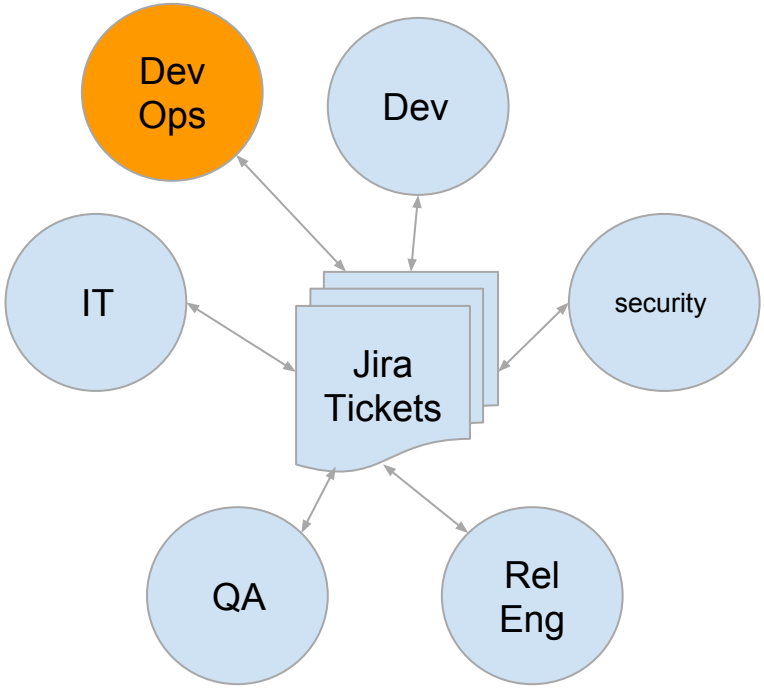


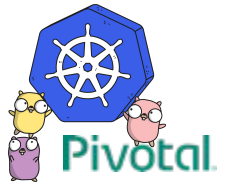

SAY DEVOPS
ONE MORE TIME

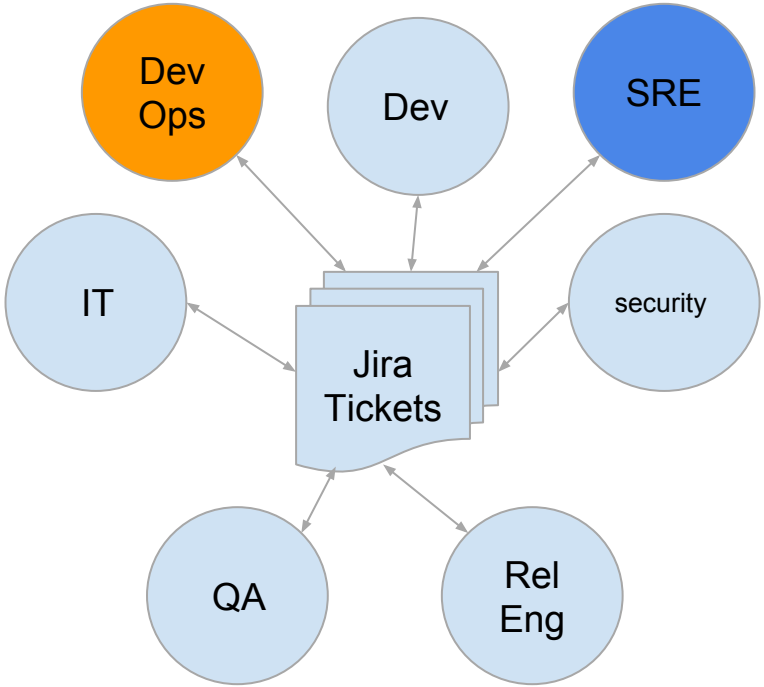

Pivotal

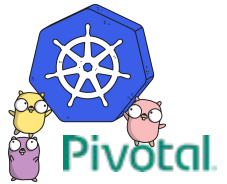

Dev
IT
security
Jira
Tickets
QA
Rel
Eng


Pivotal


Dev
Ops
Dev
IT
security
Jira
Tickets
QA
Rel
Eng


Pivotal

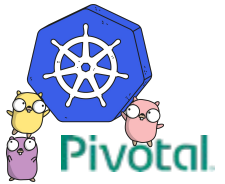

Dev
Ops
Dev
SRE
IT
Jira
Tickets
security
QA
Rel
Eng


Pivotal

Work, Work, Work

On about this date in 1801, Britain's first census was begun. In a subsequent survey conducted in 1881, residents were asked to furnish their "rank, profession, or occupation." Some of the more puzzling responses, as preserved by the London Genealogical Society, included:

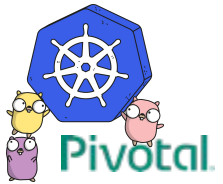
- Colourist of artificial fish
- Knight of the Thimble
- Disinfector of railways
- Examiner of underclothing
- Invisible net maker
- Electric bath attendant
- Proprietor of midgets
- Fifty-two years an imbecile
- Knocker-up of workpeople
- Maker of sand views
- Gymnast to house painter
- Turnip shepherd
- Emasculator
- Sampler of drugs
- Fatuous pauper
- Drowner
- Count as female
- Fish-bender
- Goldfish-catcher
- Cow-banger
- Running about
- Grape-dryer
- Beef twister
- Random waller

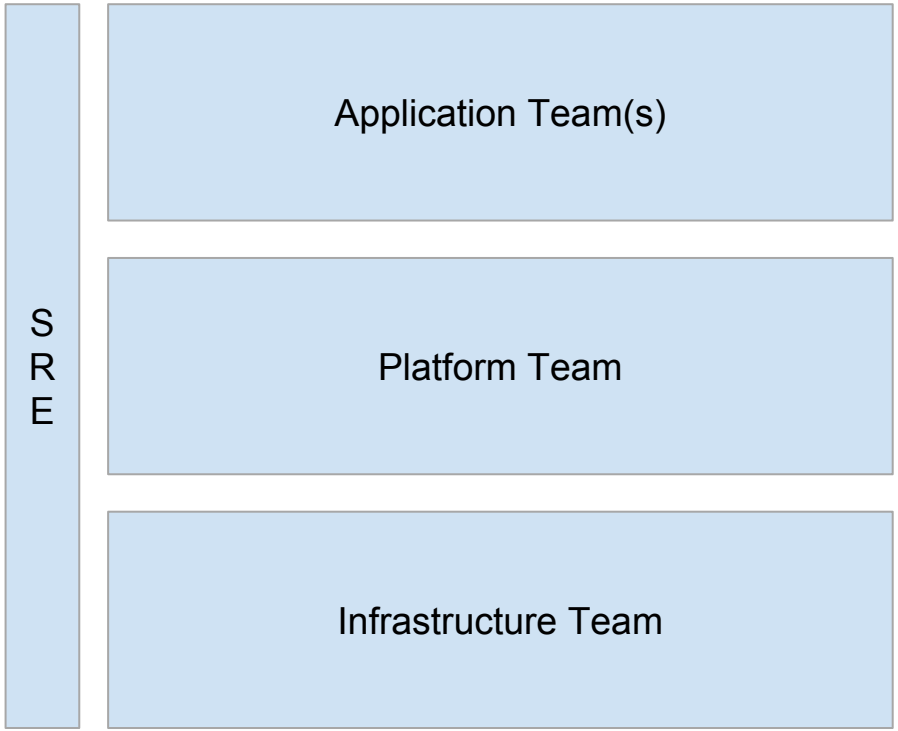


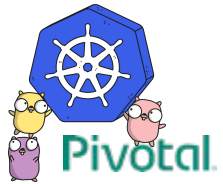
Application Team(s)

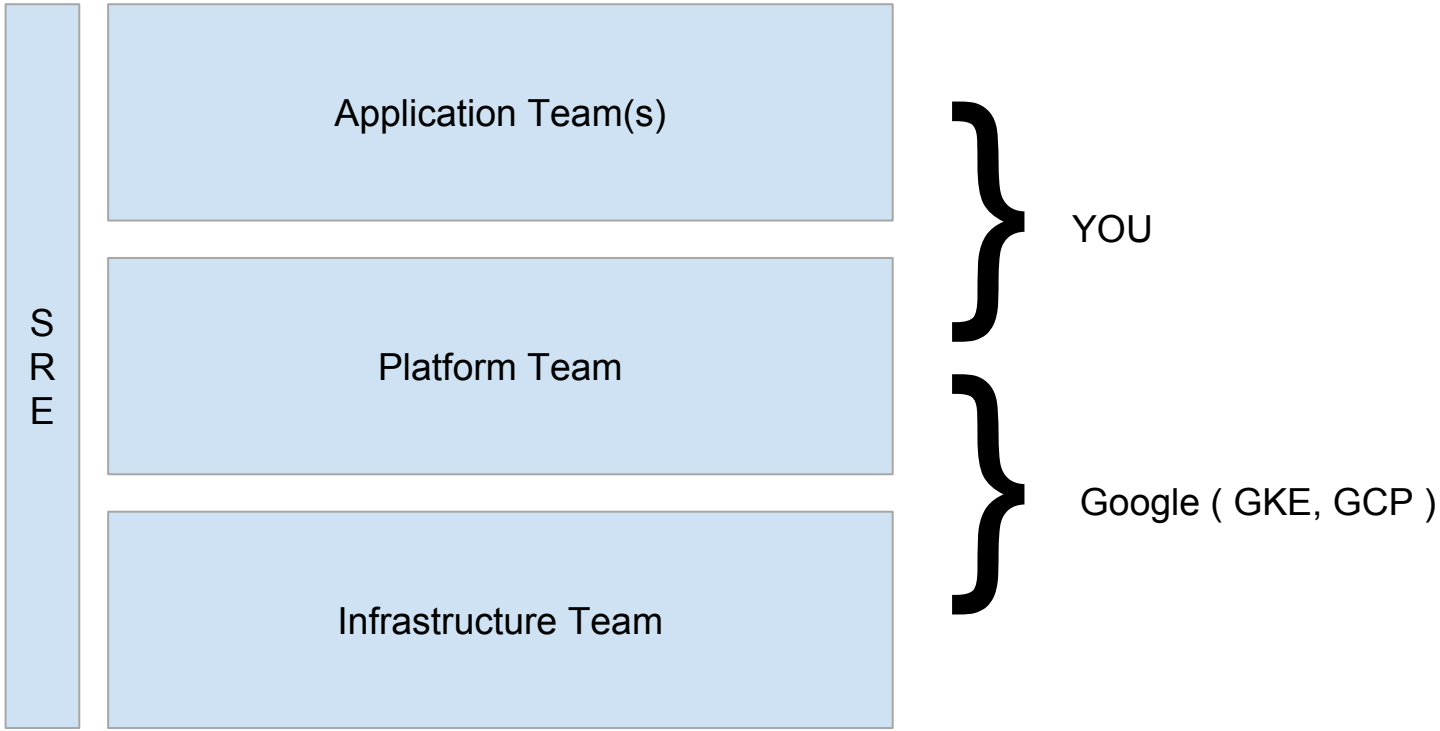
Platform Team

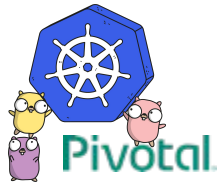
Infrastructure Team




S
R
E
Application Team(s)
Platform Team
Infrastructure Team


Pivotal


S
R
E
Application Team(s)
Platform Team
Infrastructure Team
YOU
Google (GKE, GCP)

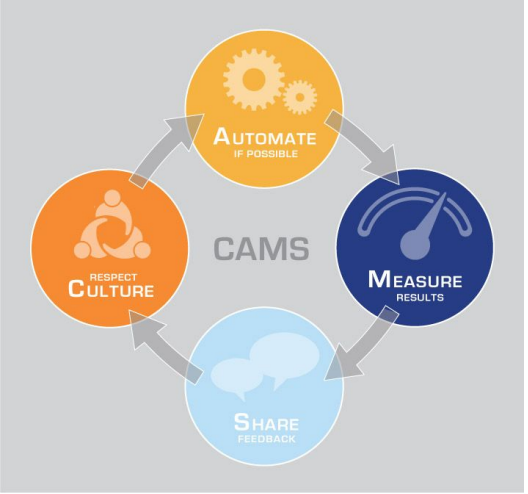

Pivotal

Culture - Increased collaboration between Development and Operations (and the rest of the business) and an attitude of shared responsibility.

Automation - Increases velocity, but just as importantly reduces defects and creates consistency and repeatability.

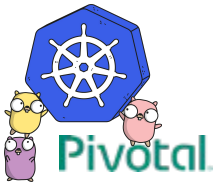
Measurement - Need to measure in order to ensure that improvement is happening.

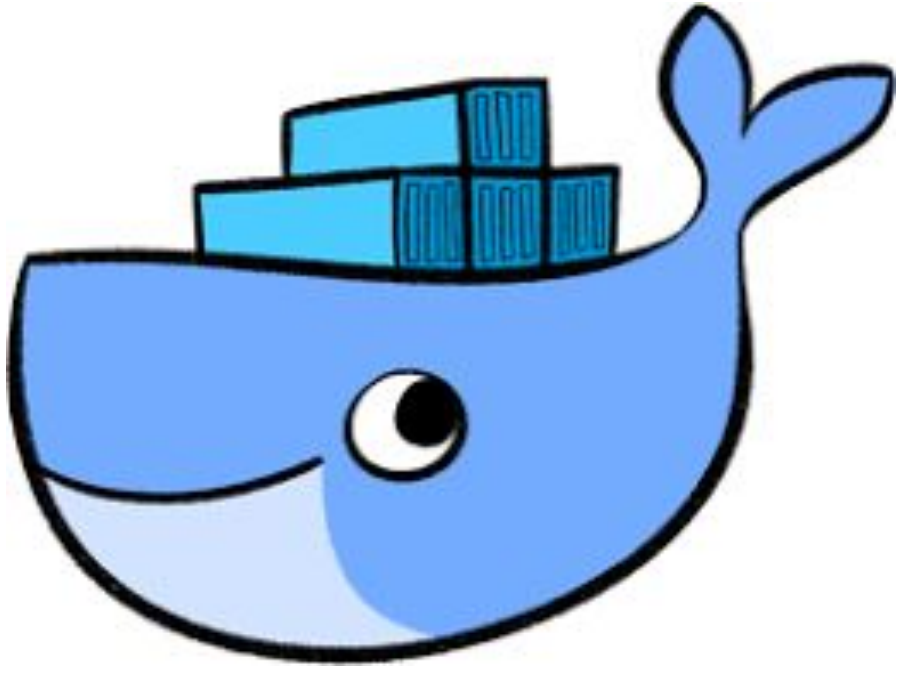
Sharing - As we share tools, discoveries and lessons new opportunities to collaborate will be discovered and duplicate work can be eliminated.

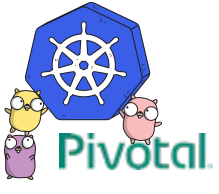


Pivotal


BRACE YOURSELVES
KUBERNETES IS COMING


Pivotal




Pivotal

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**

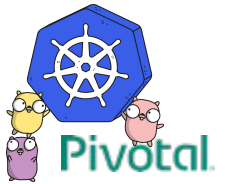
Linux containers are **like lightweight VMs** that use the built in Linux features instead of virtualizing the hardware.

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.

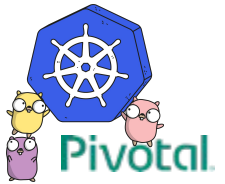


You tell **Docker** how to build a container image via a fairly simple **Dockerfile** which should generally live alongside your code in your version control system.

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)



A container orchestration system.

Greek for “Helmsman” or “Pilot”

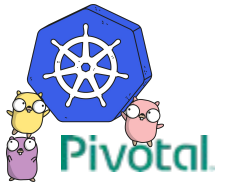
A Borg like platform using Docker as the execution engine originally built by a small team of Google engineers (Joe Beda, Brendan Burns and Craig McLuckie) and Open Sourced in 2014.

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.



An IaaS for Containers (CaaS)

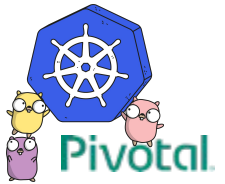
Abstracts away your infrastructure and provides a **declarative language** for the user to declare their **desired state** and then makes that **actual state**

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

Applications not **Operating Systems**.

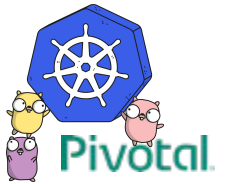
Provides a **consistent user experience** for providing **Compute**, **Network** and **Storage** resources and running applications that consume them.

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.

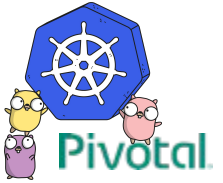
```
#!/bin/bash
while true; do
  count=$(kubectl get pods | grep nginx | wc -l)
  if $count < 5; then
     kubectl run --image=nginx nginx
  fi
  sleep 120
done
```

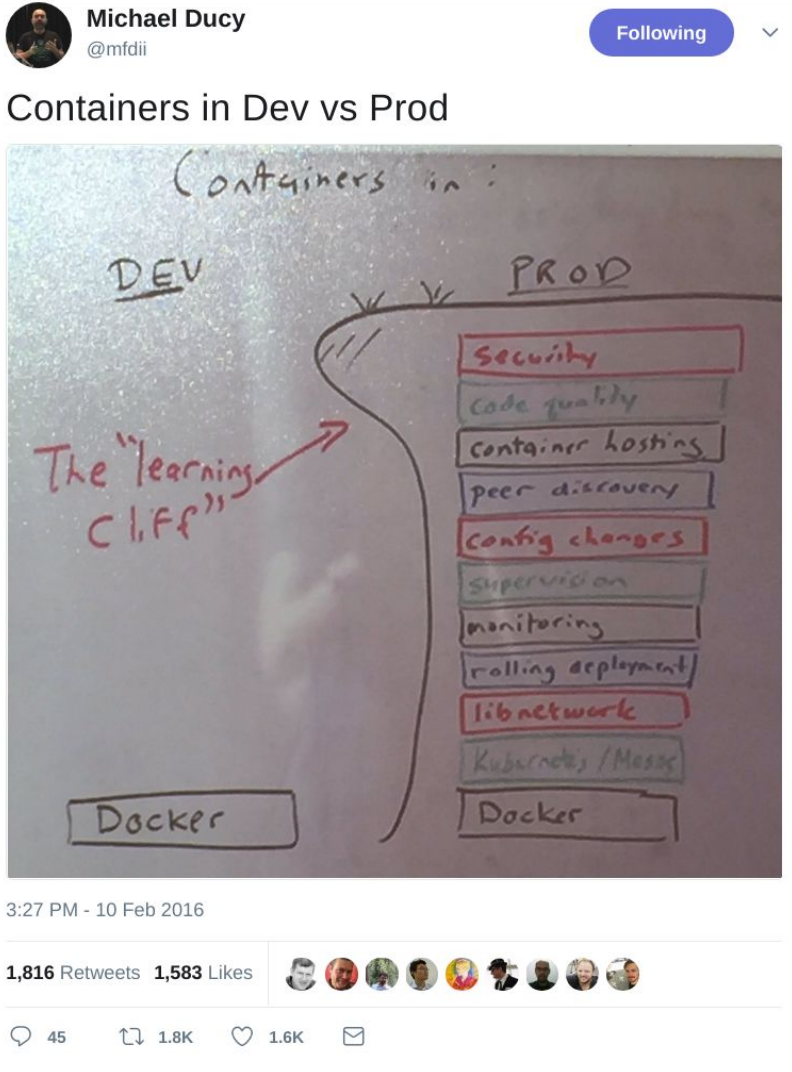


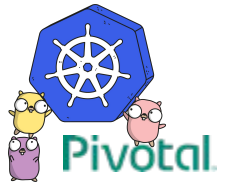
Cluster "A" has 2 running pods:
- name: A-000, version 3.0.9
- name: A-001, version 3.1.0

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

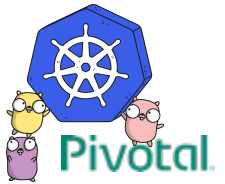
How to get to desired config:
- Recover 1 member
- Back up cluster
- Upgrade to 3.1.0




Michael Ducy
@mfdii
Following
Containers in Dev vs Prod
Containers in:
DEV
PROD
The "learning cliff"
Security
Code quality
Container hosting
peer discovery
Config changes
Supervision
monitoring
rolling deployment
libnetwork
Kubernetes / Mesos
Docker
Docker
3:27 PM - 10 Feb 2016
1,816 Retweets 1,583 Likes
45
1.8K
1.6K


Pivotal


IT'S NOT A PAAS!!!
memegenerator.net


Pivotal

Onsi Fakhouri

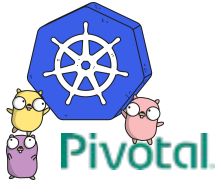
@onsijoe

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

Pivotal


THE DEVOPS
IS COMING FROM INSIDE THE KUBERNETES
negenerator.net

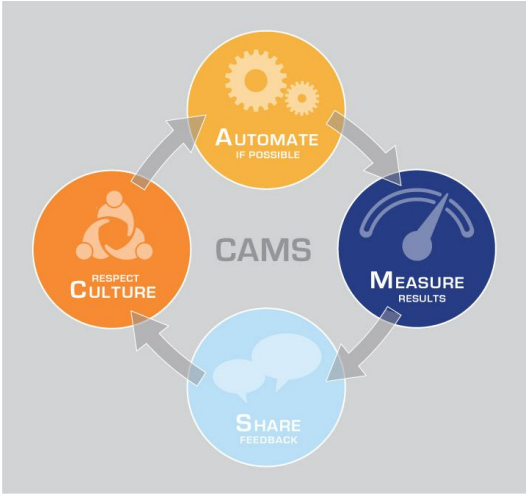
Pivotal

Culture - Increased collaboration between Development and Operations (and the rest of the business) and an attitude of shared responsibility.

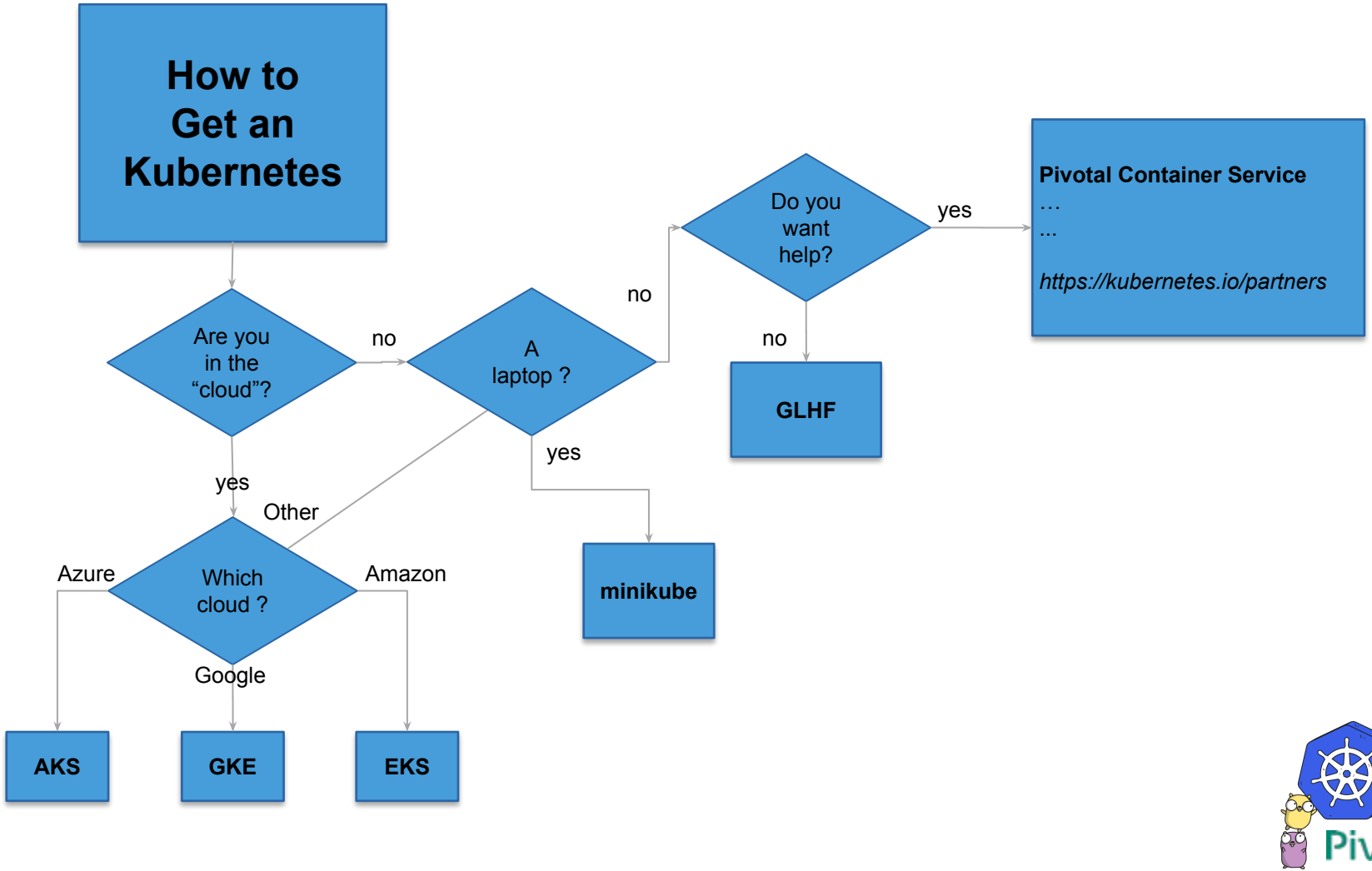
Automation - Increases velocity, but just as importantly reduces defects and creates consistency and repeatability.

Measurement - Need to measure in order to ensure that improvement is happening.

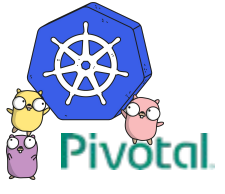
Sharing - As we share tools, discoveries and lessons new opportunities to collaborate will be discovered and duplicate work can be eliminated.

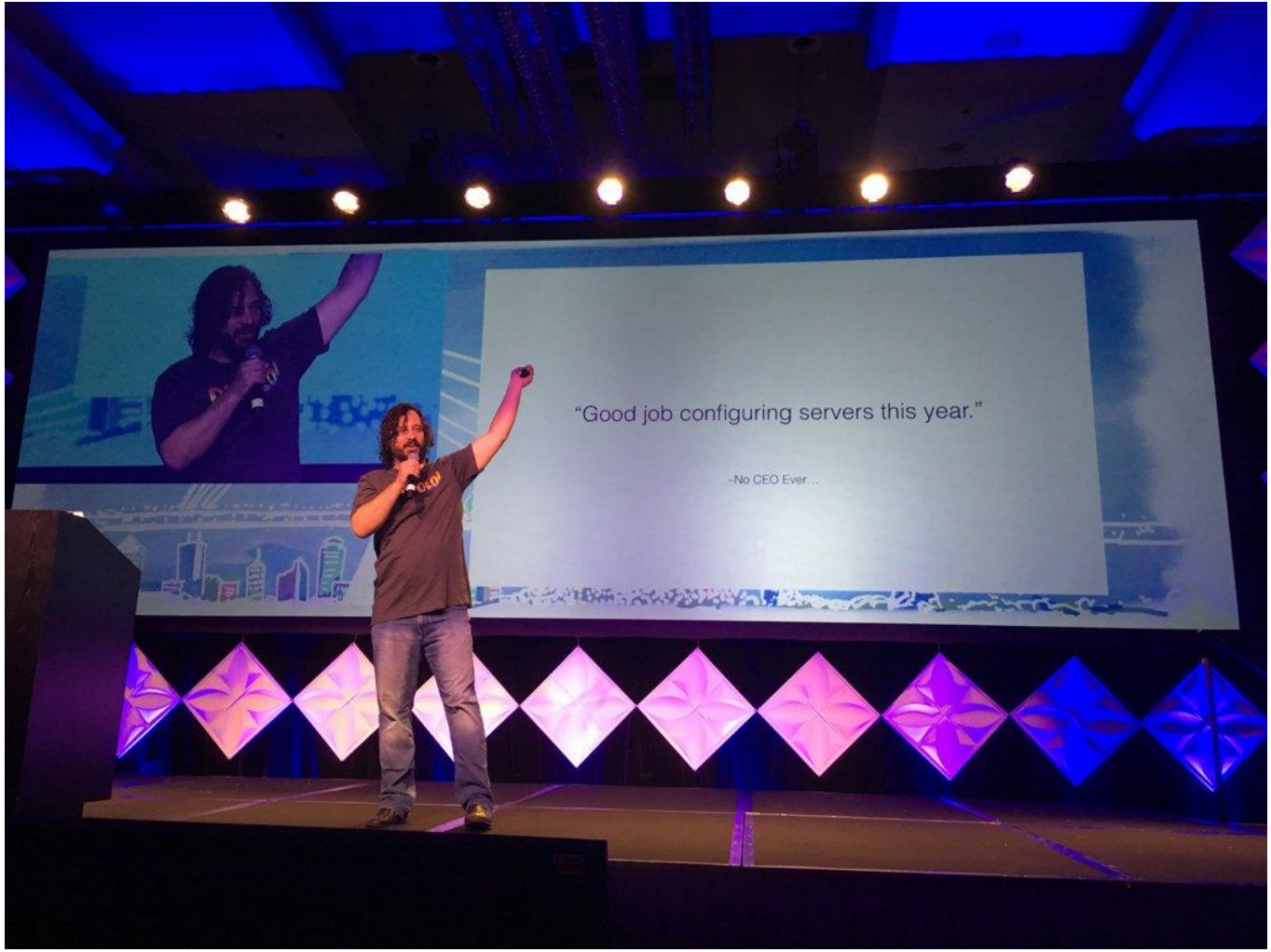


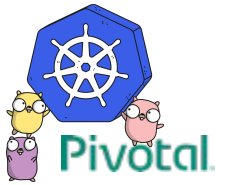
Pivotal

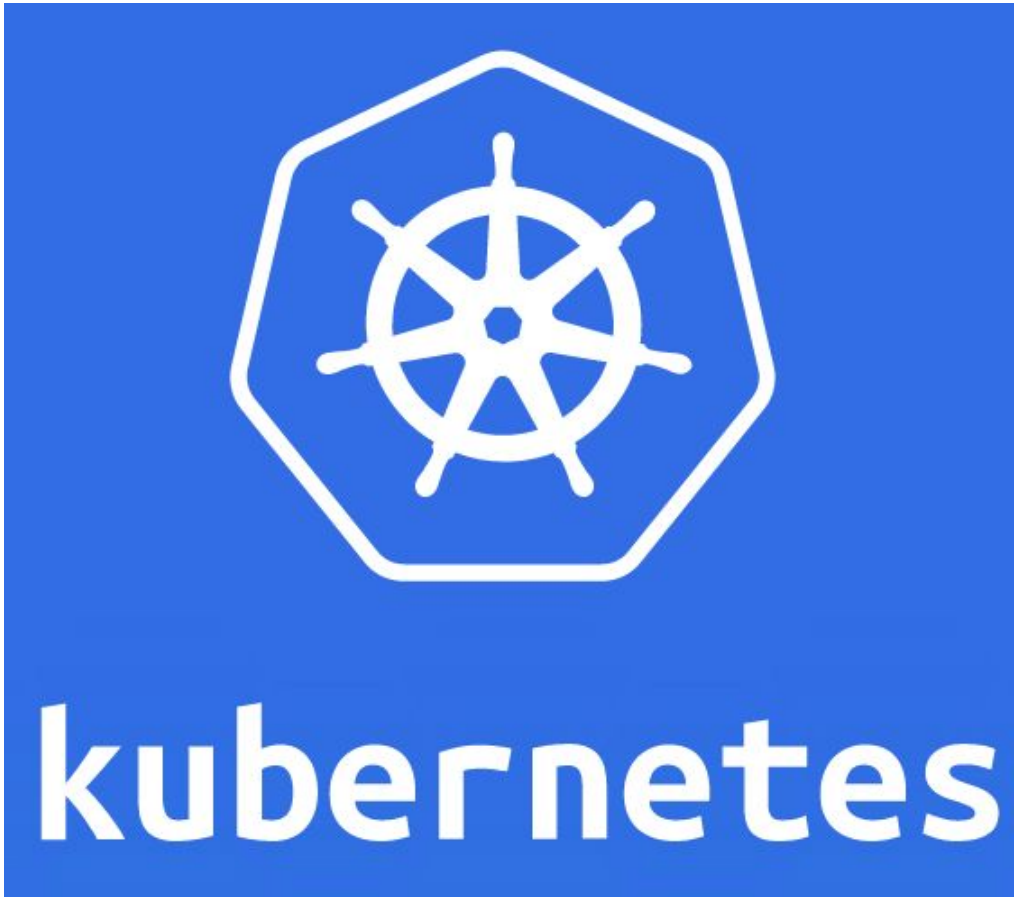

How to Get an Kubernetes
Are you in the "cloud"?
no
A laptop ?
Do you want help?
yes
Pivotal Container Service
…
...
https://kubernetes.io/partners
no
GLHF
yes
minikube
yes
Other
Which cloud ?
Azure
Amazon
Google
AKS
GKE
EKS

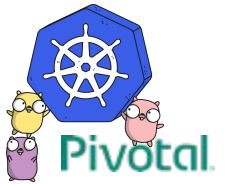

ONES DOES NOT SIMPLY INSTALL
AN KUBERNETES
memegenerator.net


Pivotal

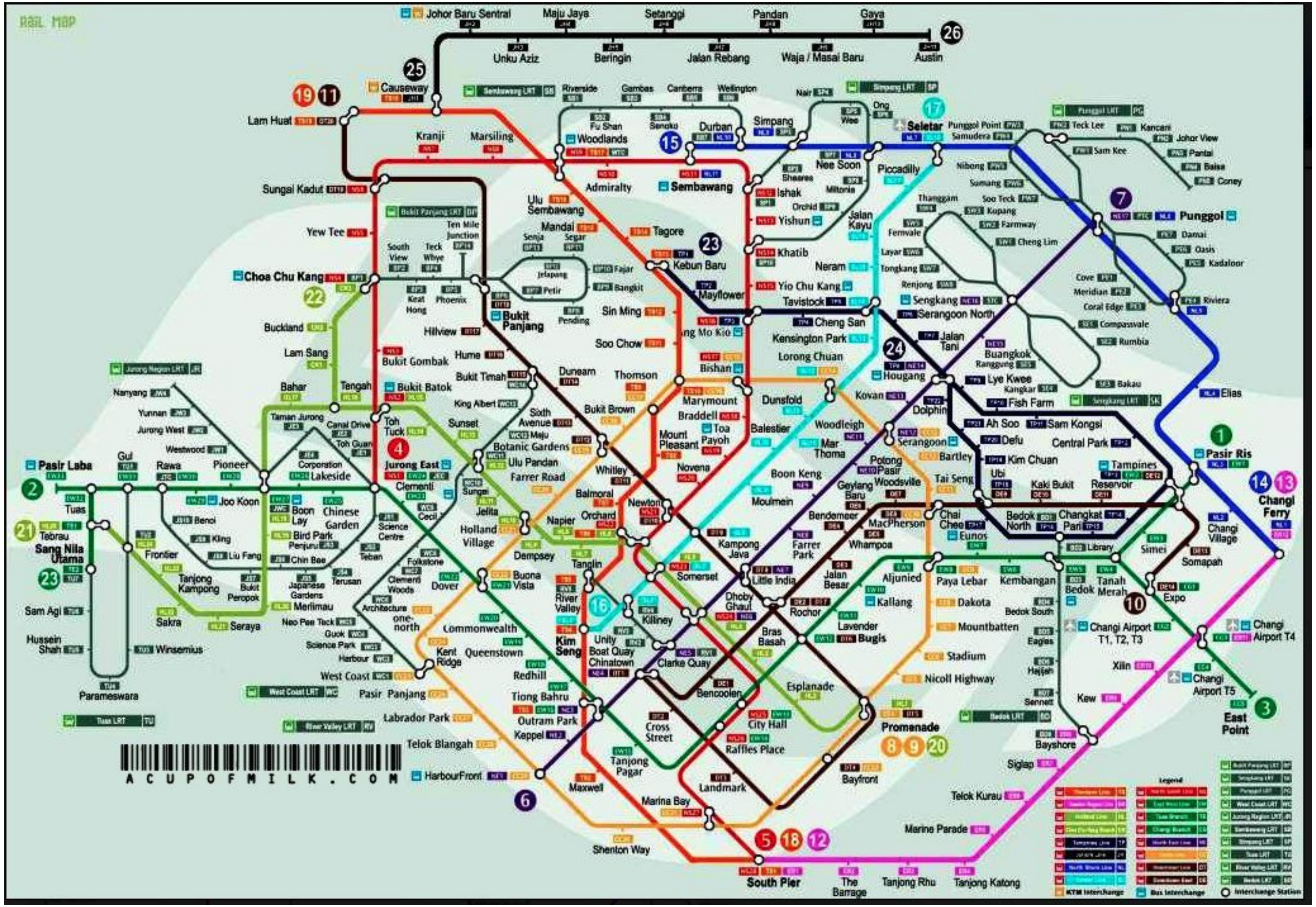

"Good job configuring servers this year."
-No CEO Ever…


Pivotal

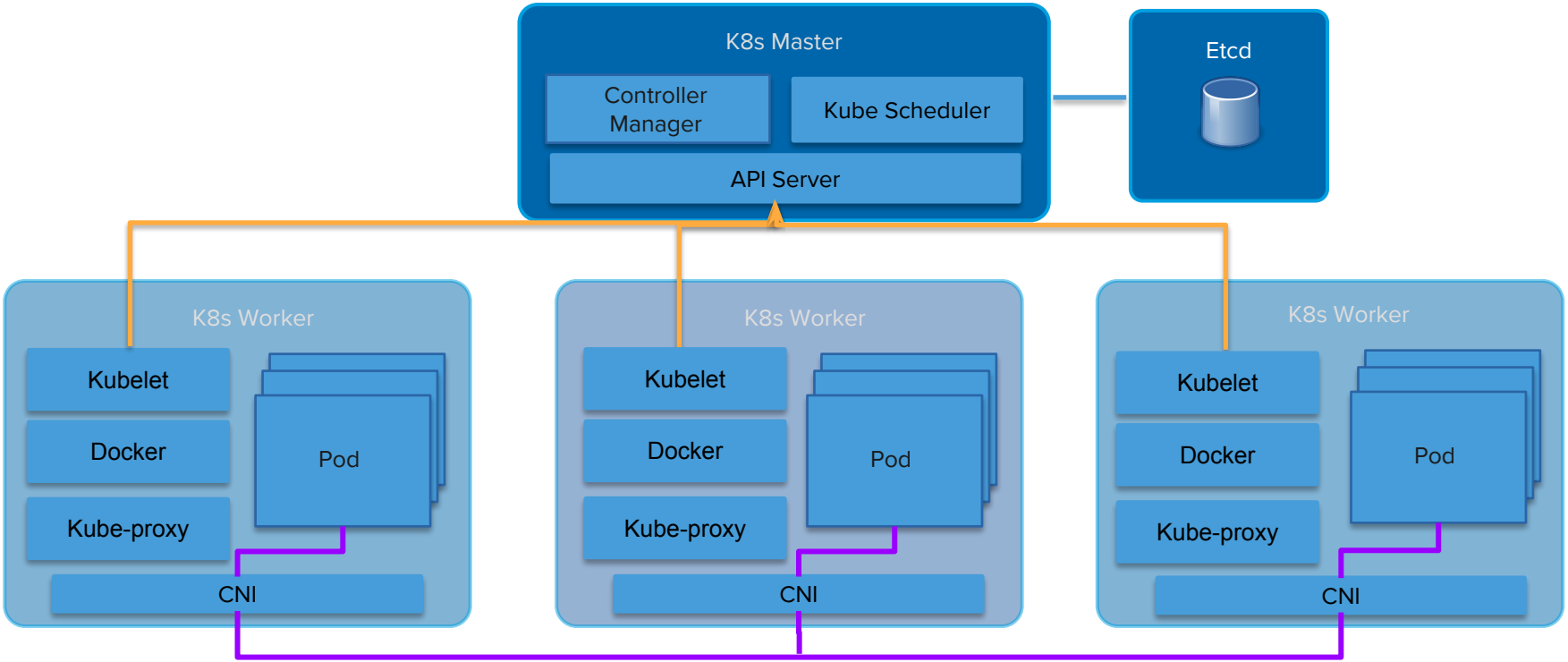

kubernetes


Pivotal

Detailed Kubernetes Architecture



Logical Kubernetes Architecture



Pivotal

Pods (Compute)

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

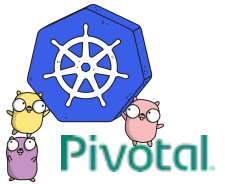
Example: a web front-end Pod that consists of an NGINX container and a PHP-FPM container with a shared unix socket and a "init" container to transform their config files based on environment variables.

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.



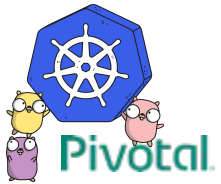
Service (network)

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.

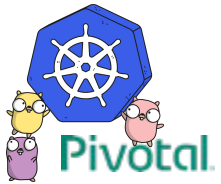


Volumes (Storage)

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)

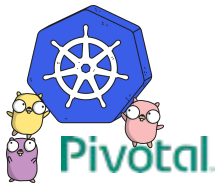
Provides **key-value pairs** to be injected into a pod much like user-data is injected into a Virtual Machine in the cloud.

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).

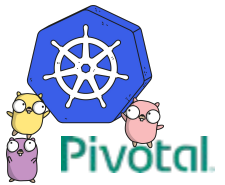
Secrets are [currently] **not encrypted** by default. This is likely to **change**.

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



Kubernetes Manifest

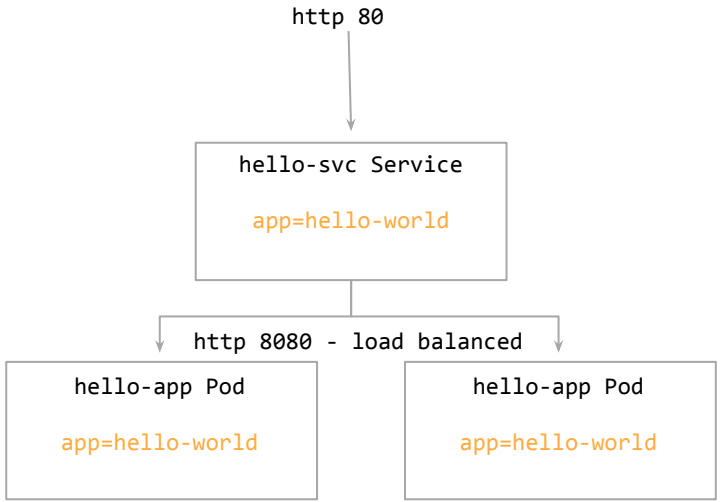
```
apiVersion:
kind:
metadata:
spec:
```

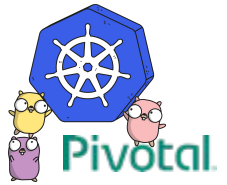


Kubernetes Manifest

```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
```

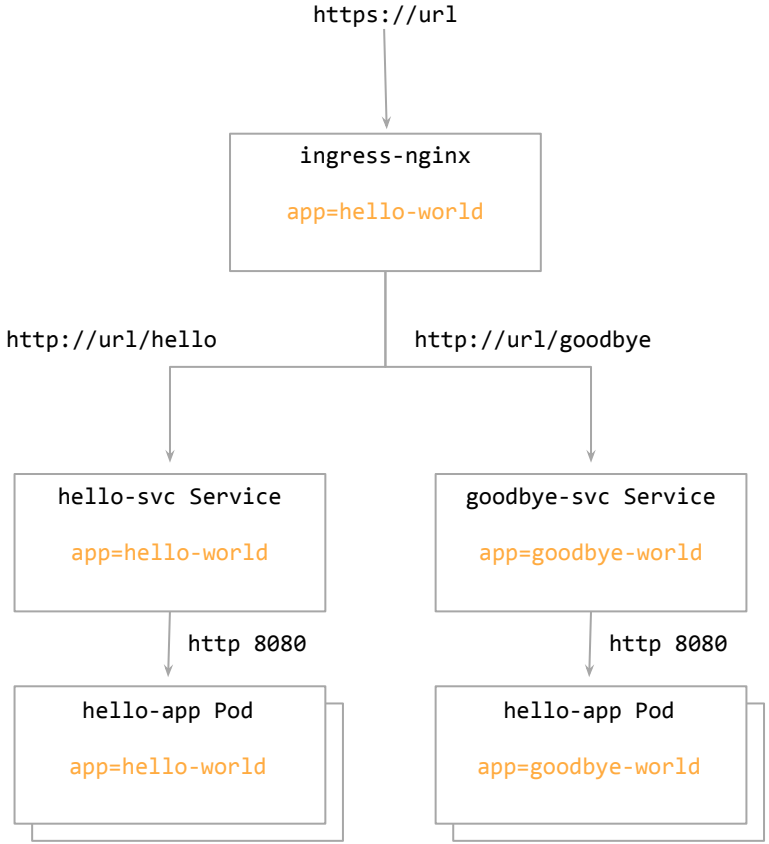
```yaml
apiVersion: v1
kind: Service
metadata:
 name: hello-svc
spec:
 ports:
 - port: 80
   protocol: TCP
   targetPort: 8080
 selector:
   app: hello-world
 type: NodePort
```

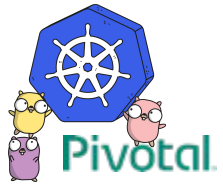




Kubernetes Manifest

```yaml
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
```



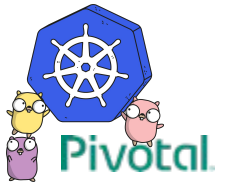


$ kubectl apply -f manifests/

```
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!

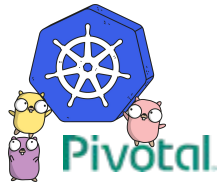
$ curl -k https://$(minikube ip)/goodbye
Goodbye Cruel world!



```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
```



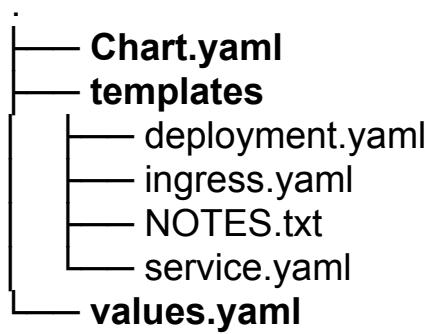
Helm is the package manager for Kubernetes

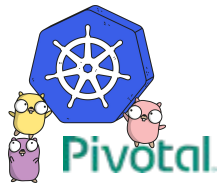
Provides tooling to **template, package, share**, and run Kubernetes manifests for a given application in the form of Charts.

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/

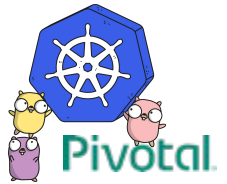




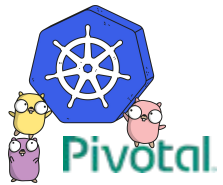
```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
 name: {{ .Chart.name}}-app
 labels:
   app: {{ .Chart.name}}
...
...
   spec:
     containers:
     - image: paulczar/hello-world
       name: hello-world
       volumeMounts:
       - name: config
         mountPath: /etc/hello
     volumes: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

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

```yaml
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’
```



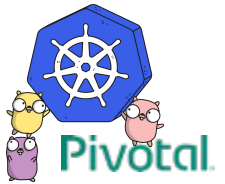
Next Steps … Further reading.

- 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/

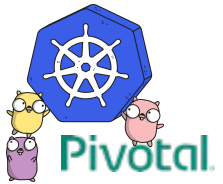
- Writing your first Helm Chart
  - https://medium.com/@pczarkowski/writing-your-first-helm-chart-f3433344f824

- 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



Questions?



Transforming How The World Builds Software