



Kubernetes

Diego Pacheco

About Me



- ❑ Cat's Father
- ❑ Principal Software Architect
- ❑ Agile Coach
- ❑ SOA/Microservices Expert
- ❑ DevOps Practitioner
- ❑ Speaker
- ❑ Author



diegopacheco



@diego_pacheco



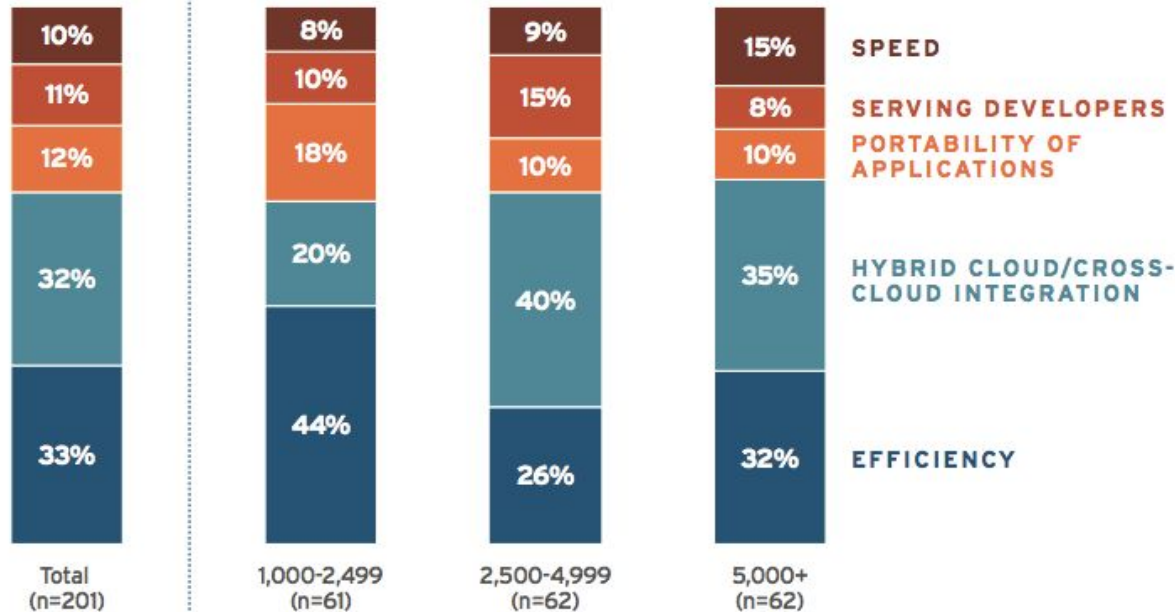
<http://diego-pacheco.blogspot.com.br/>



Why Containers?

Figure 3: Primary driver of container user by company size

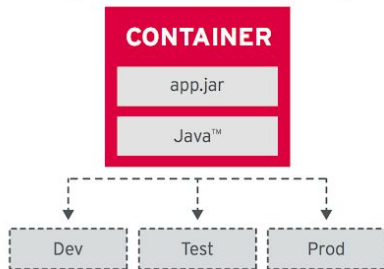
Q. What is the primary driver for containers in your organization?



Source: 451 Research

Containers Principles

Image Immutability Principle



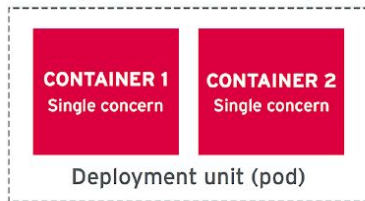
High Observability Principle



Lifecycle Conformance Principle



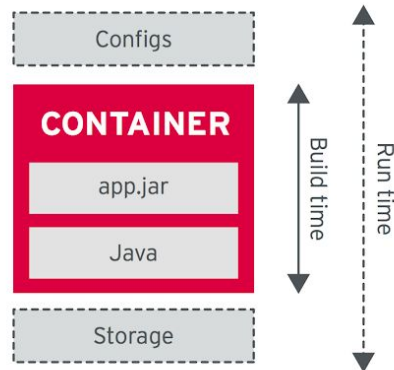
Single Concern Principle



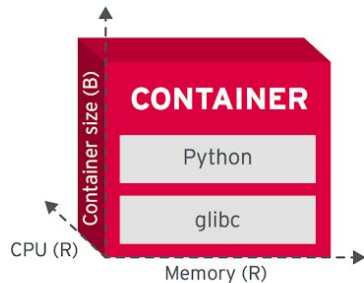
Process Disposability Principle



Self-Containment Principle



Runtime Confinement Principle



Kubernetes (K8s)

Container cluster management



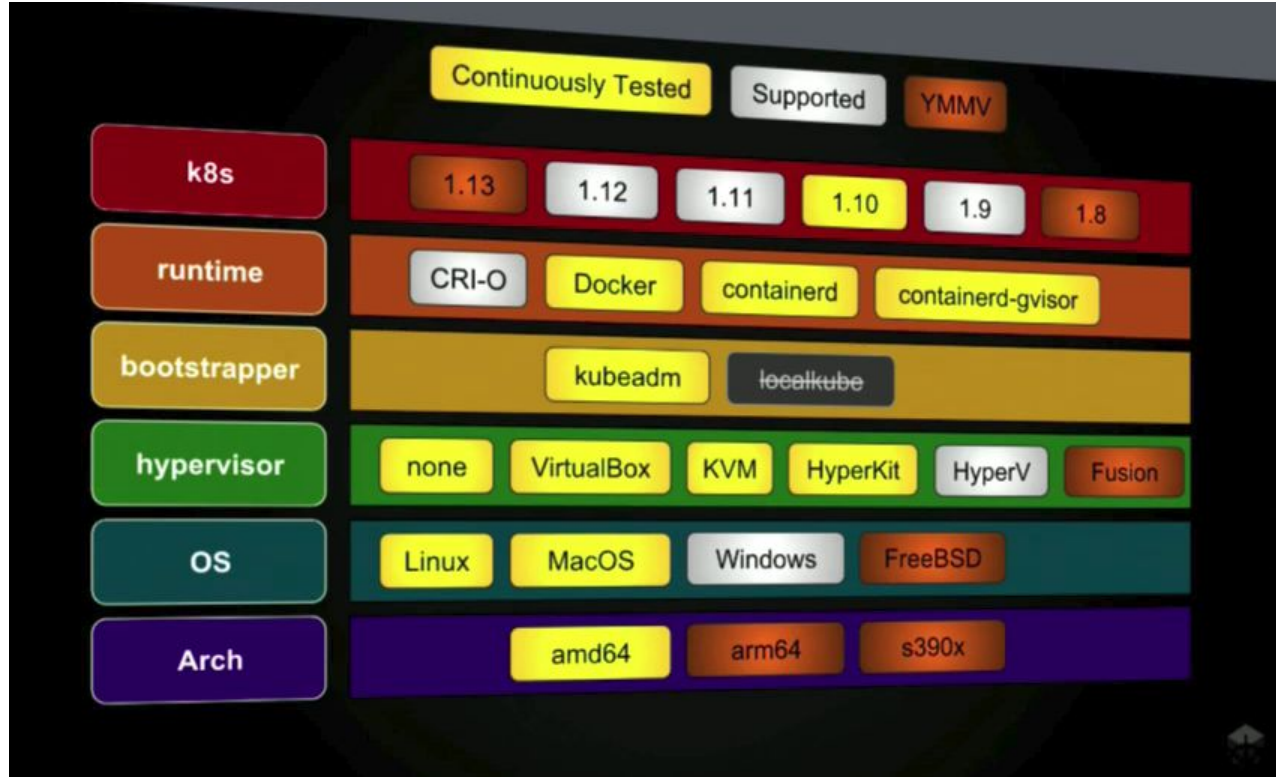
kubernetes




- Distributed configuration
- Service Discovery
- Loadbalancing
- Versioning/Routing
- Deployments
- Scaling/Autoscaling
- Liveness/Health checking
- Self healing



Another Layer of Abstraction



K8s Dashboard

 **kubernetes**

Search

+ CREATE

Overview

Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace

default

Overview

Workloads

Daemon Sets

Deployments

Jobs

Pods

Replica Sets



Replication Controllers

Stateful Sets




Discovery and Load Balancing

Ingresses



Deployments

Name	Labels	Pods	Age	Images
 show-the-awesome	app: show-the-awesome	2 / 2	15 minutes	wordpress
 redis-test	app: redis-test	1 / 1	21 minutes	redis

Pods

Name	Node	Status	Restarts	Age
 show-the-awesome-254335071-2bz4c	minikube	Running	0	15 minutes
 show-the-awesome-254335071-wmmd3	minikube	Running	0	15 minutes
 redis-test-795539674-d0tzf	minikube	Running	0	21 minutes

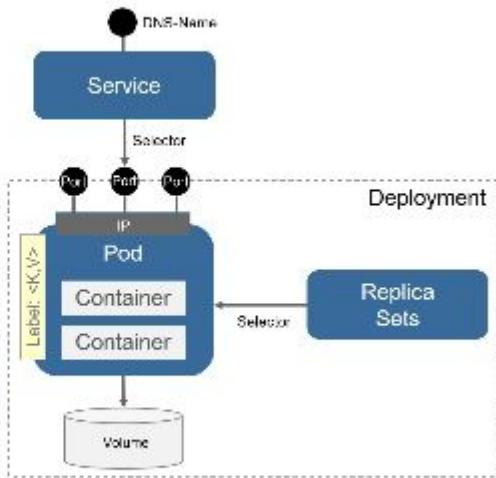
Replica Sets

Name	Labels	Pods	Age	Images
 show-the-awesome-254335071	app: show-the-awesome pod-template-hash: 2543350...	2 / 2	15 minutes	wordpress
 redis-test-795539674	app: redis-test pod-template-hash: 7955396...	1 / 1	21 minutes	redis

Concepts

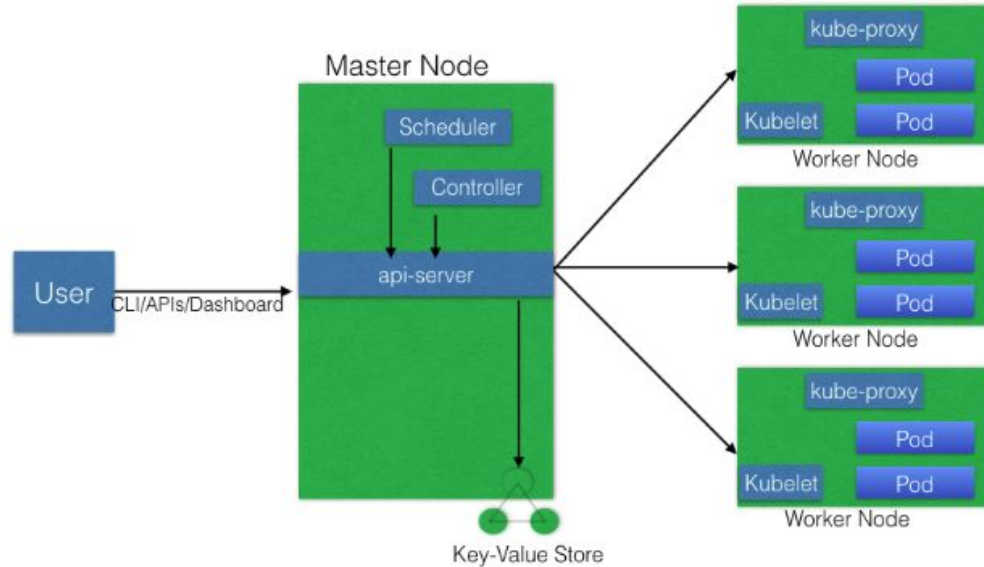
Main Kubernetes concepts

- **Services** are an abstraction for a logical set of Pods.
- **Pods** are the smallest deployable units of computing.
- **Deployments** provide declarative updates for Pods and RCs.
- **Replica Sets** ensure specified number of Pods are running.
- **Labels** are key/value pairs attached to objects used for identification.

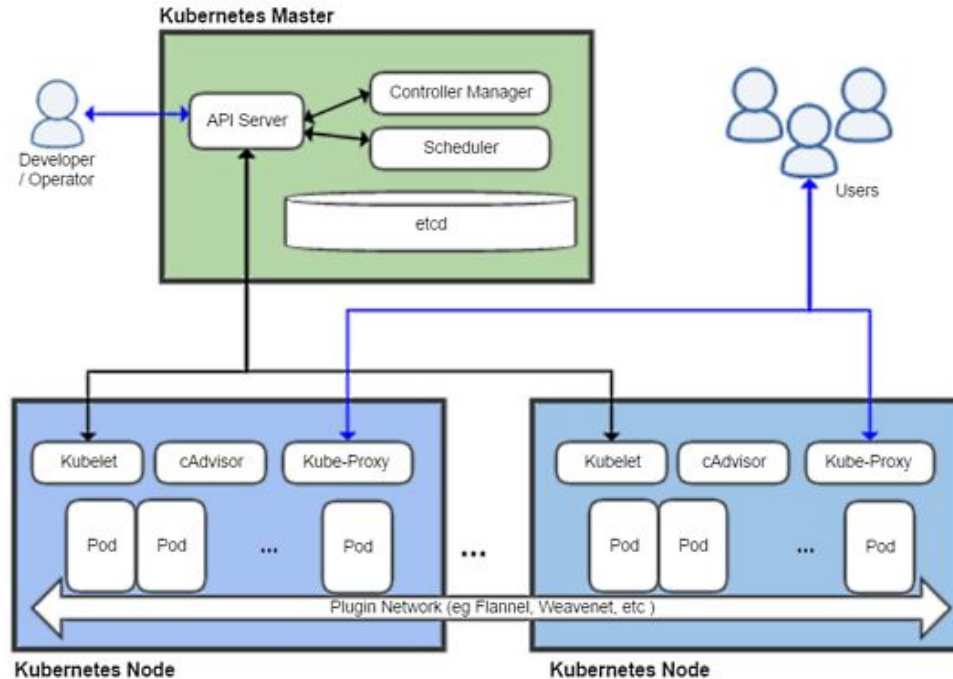


K8s Architecture

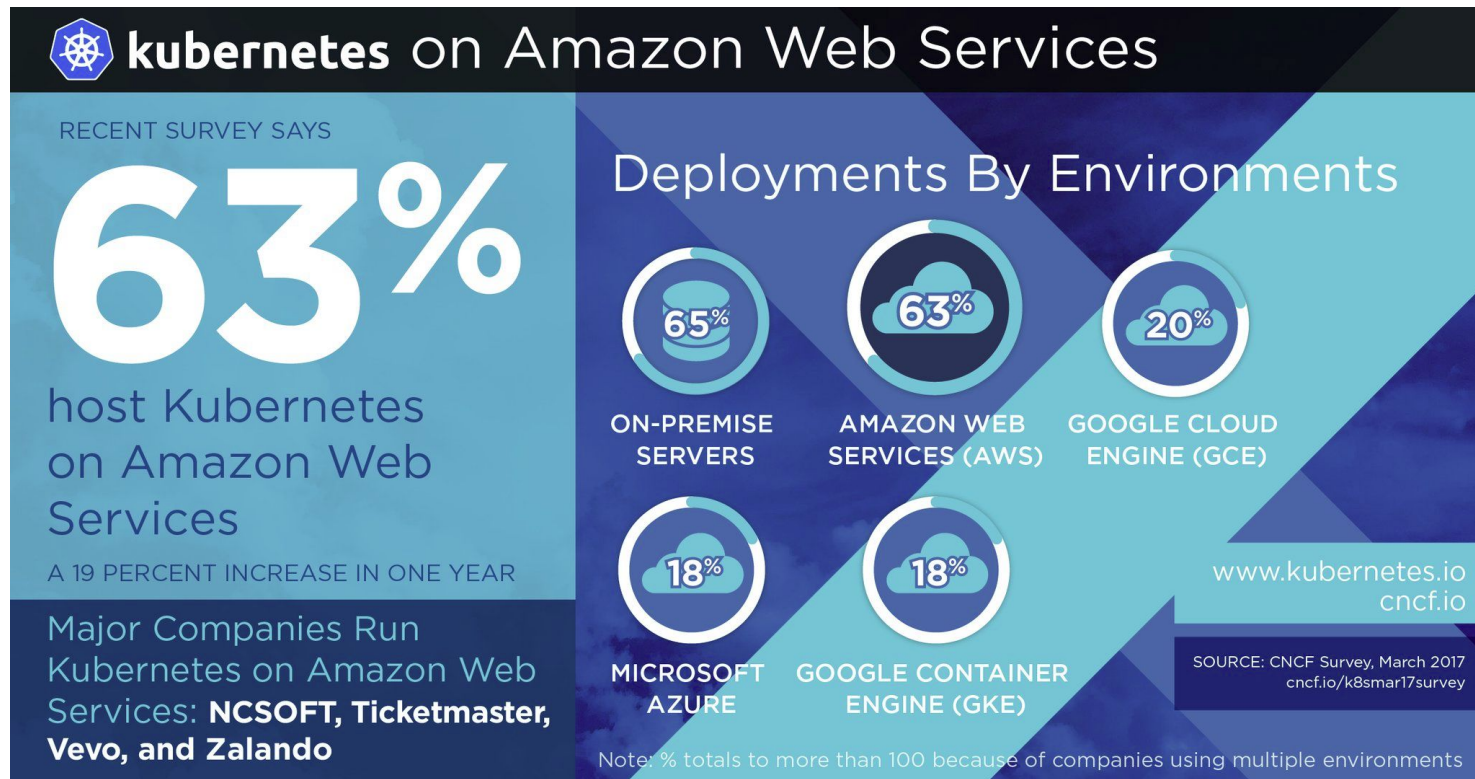
Architecture



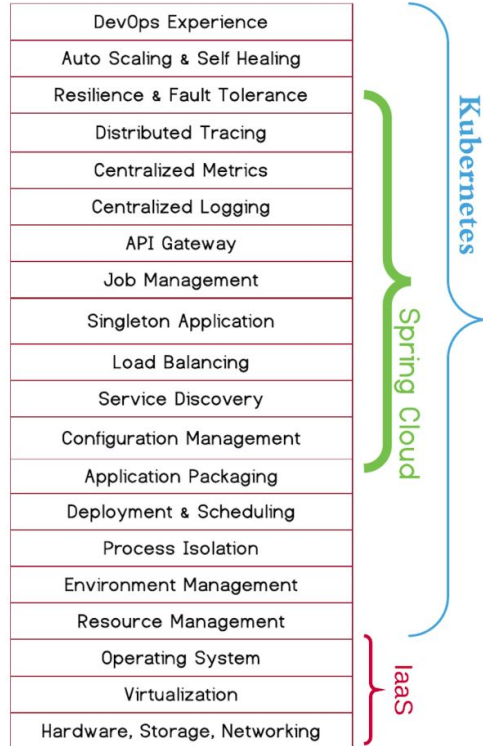
K8s Architecture



Where People are Running k8s?



Container Native VS Cloud-Native



Multi-Region?

Multi-region == Multi-Kubernetes-cluster

- Kubernetes is not designed to span WANs
 - Originally didn't even want to have to span datacenters/AZs within a region, but the community fought for that and made it happen
- Try to run a single k8s cluster across regions at your own risk



minikube

Minikube Architecture



VM

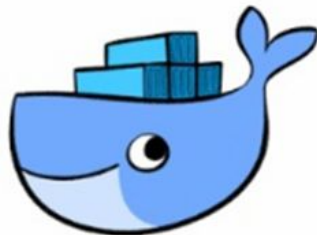


Master



Node

localkube



Container
Runtime

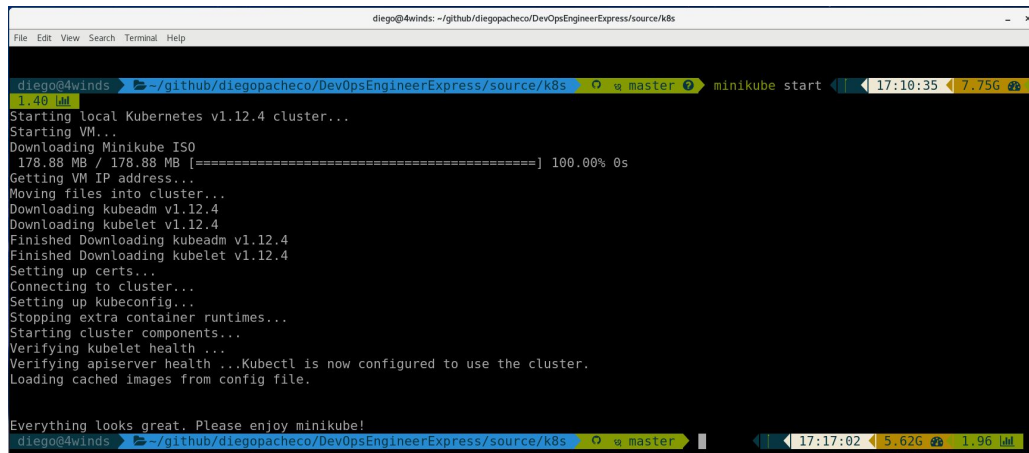
Minikube Features

- Minikube supports Kubernetes features such as:
 - DNS
 - NodePorts
 - ConfigMaps and Secrets
 - Dashboards
 - Container Runtime: Docker, [rkt](#), [CRI-O](#) and [containerd](#)
 - Enabling CNI (Container Network Interface)
 - Ingress

Minikube - Installation

install-minikube.sh x

```
1  #!/bin/bash
2
3  curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.32.0/minikube-linux-amd64 && \
4  chmod +x minikube && sudo cp minikube /usr/local/bin/ && rm minikube
5
6  minikube start
```



The screenshot shows a terminal window titled "diego@4winds: ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s". The terminal output shows the execution of "minikube start" which successfully creates a local Kubernetes v1.12.4 cluster. The output includes progress bars for downloading the Minikube ISO and kubeadm/kubelet binaries. The status bar at the bottom of the terminal window shows the time as 17:17:02, 5.62G of memory used, and 1.96 MB of disk space used.

```
diego@4winds ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s
File Edit View Search Terminal Help

diego@4winds ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s 17:10:35 7.75G
minikube start
Starting local Kubernetes v1.12.4 cluster...
Starting VM...
Downloading Minikube ISO
178.88 MB / 178.88 MB [=====] 100.00% 0s
Getting VM IP address...
Moving files into cluster...
Downloading kubeadm v1.12.4
Downloading kubelet v1.12.4
Finished Downloading kubeadm v1.12.4
Finished Downloading kubelet v1.12.4
Setting up certs...
Connecting to cluster...
Setting up kubeconfig...
Stopping extra container runtimes...
Starting cluster components...
Verifying kubelet health ...
Verifying apiserver health ...Kubectl is now configured to use the cluster.
Loading cached images from config file.

Everything looks great. Please enjoy minikube!
diego@4winds ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s 17:17:02 5.62G 1.96 MB
```

Minikube - Deploy

deploy.sh x

```
1  #!/bin/bash
2
3  kubectl run hello-minikube --image=k8s.gcr.io/echoserver:1.10 --port=8080 deployment.apps/hello-minikube
4  kubectl expose deployment hello-minikube --type=NodePort service/hello-minikube
5
6  kubectl get pod
7  curl $(minikube service hello-minikube --url)
```

Lets deploy Redis on K8s

! redis-service.yaml x

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: redis
5    labels:
6      app: redis
7      role: master
8      tier: backend
9  spec:
10   type: LoadBalancer
11   ports:
12     - port: 6379
13     targetPort: 6379
14   selector:
15     app: redis
16     role: master
17     tier: backend
```

! redis-deployment.yaml x

```
1  apiVersion: extensions/v1beta1
2  kind: Deployment
3  metadata:
4    name: redis
5  spec:
6    replicas: 1
7    template:
8      metadata:
9        labels:
10         app: redis
11         role: master
12         tier: backend
13      spec:
14        containers:
15          - name: redis
16            image: redis
17            resources:
18              requests:
19                cpu: 100m
20                memory: 100Mi
21            ports:
22              - containerPort: 6379
23
```

Lets deploy Redis on K8s

```
2 eval $(minikube docker-env)
3
4 kubectl create -f redis/
5
6 kubectl get deployments
7
8 kubectl get pods
9
```

redis-cli -h 192.168.99.100 -p 32037

File Edit View Search Terminal Help

diego@4winds ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s master kubectl get svc

NAME	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
hello-minikube	10.100.40.88	<nodes>	8080:30032/TCP	17m
kubernetes	10.96.0.1	<none>	443/TCP	22m
redis	10.99.157.228	<pending>	6379:32037/TCP	5m

diego@4winds ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s master redis-cli -h 192.168.99.100 -p 32037

192.168.99.100:32037> keys *

1) "kubernetes"

192.168.99.100:32037> get kubernetes

"works"

192.168.99.100:32037> █

Lets deploy Redis on K8s

```
diego@4winds: ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s
File Edit View Search Terminal Help
diego@4winds ~ ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s master kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
hello-minikube-6b75d57bbd-tt4kd     1/1      Running   0           23m
redis-6669d7cbdb-7498g              1/1      Running   0           10m
diego@4winds ~ ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s master kubectl logs redis-6669d7cbdb-7498g
1:C 28 Dec 2018 19:33:06.081 # o000o000o000o Redis is starting o000o000o000o
1:C 28 Dec 2018 19:33:06.081 # Redis version=5.0.3, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 28 Dec 2018 19:33:06.081 # Warning: no config file specified, using the default config. In order to specify a config file use re
dis-server /path/to/redis.conf
1:M 28 Dec 2018 19:33:06.083 * Running mode=standalone, port=6379.
1:M 28 Dec 2018 19:33:06.083 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is se
t to the lower value of 128.
1:M 28 Dec 2018 19:33:06.084 # Server initialized
1:M 28 Dec 2018 19:33:06.084 * Ready to accept connections
1:M 28 Dec 2018 19:36:55.797 # Possible SECURITY ATTACK detected. It looks like somebody is sending POST or Host: commands to Redis.
This is likely due to an attacker attempting to use Cross Protocol Scripting to compromise your Redis instance. Connection aborted.
diego@4winds ~ ~/github/diegopacheco/DevOpsEngineerExpress/source/k8s master 17:43:04 5.35G 1.08
```

minikube dashboard

127.0.0.1:43713/ap/v1/namespaces/kube-system/services/http:kubernetes-dashboard:/proxy/#/overview?namespace=default

Apps Speed Dial 2 Engineering Tech default Links Work util SPE-SOS CM CM.Cassandra.KT DevOps.Engineer DevOps.Engineer Other bookmarks

kubernetes Search + CREATE

Overview

Cluster

- Namespaces
- Nodes
- Persistent Volumes
- Roles
- Storage Classes

Namespace

- default


Overview

Workloads

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets


Workloads

Workloads Statuses




100.00%

Deployments



100.00%

Pods



100.00%

Replica Sets

Deployments

Name	Labels	Pods	Age	Images
redis	app: redis tier: backend	1 / 1	12 minutes	redis
hello-minikube	run: hello-minikube	1 / 1	24 minutes	k8s.gcr.io/echoserver:1.10

Pods

Name	Node	Status	Restarts	Age
redis-6660d7chdh-7409n	minikube	Running	0	12 minutes



Kubernetes

Diego Pacheco