

Kubernetes

Architecture & Introduction



Dr. Stefan Schimanski, Sergiusz Urbaniak



#k8sber @the1stein @mieszkoman

hash tag #k8sber
follow @kubernetesBER

Who are we?

Engineers @ Mesosphere

Working on **Kubernetes-on-Mesos**

<https://github.com/kubernetes/kubernetes/tree/master/contrib/mesos>



Stefan
[@the1stein](#)
[sttts@github](#)



Sergiusz
[@mieszkoman](#)
[s-urbaniak@github](#)

Who/What is Mesosphere?

The company behind:

- **Apache Mesos** (<https://mesos.apache.org/>)
A distributed systems kernel.
- **Marathon** (<https://mesosphere.github.io/marathon/>)
An Apache Mesos framework
for long-running applications.
- **Chronos** (<http://mesos.github.io/chronos/>)
A distributed cron replacement.

DCOS

(<https://mesosphere.com/>)

A data center operating system.

Next 45 Minutes

- Archeology: before and without Kubernetes
- Deployment: kube-up, DCOS, GKE
- Core Architecture: the **apiserver**, the **kubelet** and the **scheduler**
- Compute Model: the **pod**, the **service** and the **controller**


No whitepaper talk. We dive deep!



googlearchive / container-agent

How it all started

Python based **agent** to deploy **declarative pods** using **Docker**.


**initial import**
proppey committed on May 16, 2014

...

**Merge pull request #12 from jbeda**
jbeda committed on Jun 8, 2014

**Add a deprecation notice**
thockin committed on Jun 20, 2014

Kubernetes:

**First commit**
jbeda committed on Jun 7, 2014

googlearchive / container-agent


Watch 103 Star 206 Fork 16

[Code](#) [Issues 2](#) [Pull requests 2](#) [Pulse](#) [Graphs](#)

Simple agent for running containers based on a declarative manifest.

13 commits 1 branch 0 releases 7 contributors

Branch: master [New pull request](#) [New file](#) [Find file](#) [SSH](#) [git@github.com:googlearchive/co](#) [Download ZIP](#)

 elibixby Merge pull request #19 from jonparrott/patch-1 Latest commit c6cead2 on Sep 23, 2015

container_agent	initial import	2 years ago
manifests	initial import	2 years ago
tests	initial import	2 years ago
.gitignore	initial import	2 years ago
.travis.yml	Enable automated testing with Python 3.4	2 years ago
CONTRIBUTING.md	initial import	2 years ago

A Pod was described as a ...

[...] **container group** defined by the manifest **to share:**

- **Network Namespaces**
- **Volumes**

This creates a runtime environment where:

- Containers can connect to a service running in other containers of the same group using localhost and a fixed port.
- Containers of the same group can't run services on the same ports.
- Containers of the same group can mount shared volumes defined in the manifest.

A Pod was described as a ...

Still true with
Kubernetes

[...] **container group** defined by the manifest **to share:**

- **Network Namespaces**
- **Volumes**

This creates a runtime environment where:

- Containers can connect to a service running in other containers of the same group using localhost and a fixed port.
- Containers of the same group can't run services on the same ports.
- Containers of the same group can mount shared volumes defined in the manifest.

Back to the roots

A Pod in Docker terms:

```
$ docker run -it --name mypod busybox sleep 99999999
```

```
$ docker run -it --ipc=container:mypod --net=container:mypod  
-v /tmp/mypod:/tmp/mypod  
-w /tmp/mypod ubuntu python3 -m http.server 80
```

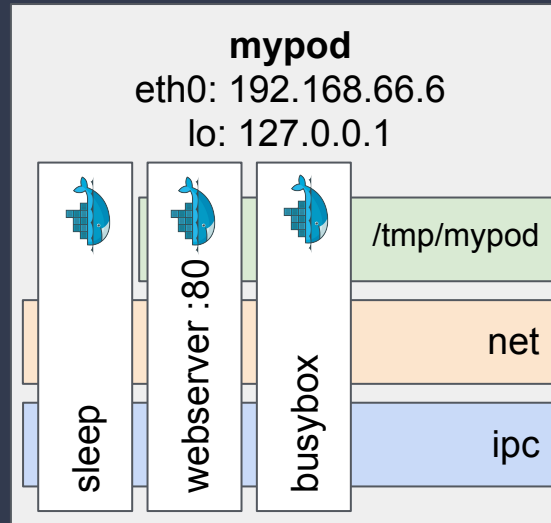
```
$ docker run -it --ipc=container:mypod --net=container:mypod  
-v /tmp/mypod:/tmp/mypod  
sttts/busybox-curl /bin/sh
```

“root”, “pause” or
“infrastructure” container

“webserver”

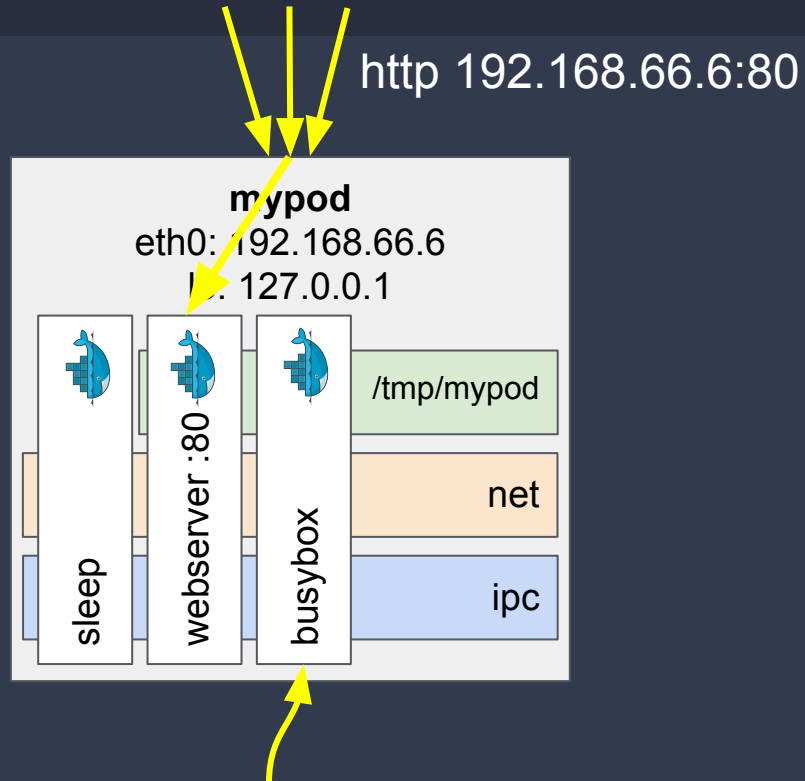
“busybox”, our worker

A Pod in Docker terms - visually



- **Note:** different OSe, resource constraints, restart policies

A little Twitter Mirror Pod



```
while true; do
  curl -k 'https://mobile.twitter.com/search?q=%23kubernetes%20%23berlin&s=typd' \
    > /tmp/mypod/index.html
  sleep 10
done
```

Get a Kubernetes cluster: mesos/docker with kube-up

```
$ git clone git@github.com:kubernetes/kubernetes.git
$ cd kubernetes
$ build/run.sh hack/build-go.sh && make
$ KUBERNETES_PROVIDER=mesos/docker cluster/kube-up.sh
$ alias kubectl=_output/local/bin/darwin/amd64/kubectl
$ kubectl get pods
```

Package for Mesosphere's DCOS

```
$ dcos config prepend package.sources \  
  https://github.com/mesosphere/multiverse/archive/version-1.x.zip
```

```
$ dcos package install kubernetes
```

```
$ dcos kubectl create -f nginx.yml  
pods/nginx
```

```
$ dcos kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	1/1	Running	0	1m



Google Kontainer Engine

- install the **gcloud cli**
- prepare your Google Cloud account with a **project** with enabled GCE APIs, e.g. “kube-test-1154”

```
$ gcloud components install kubectl
```

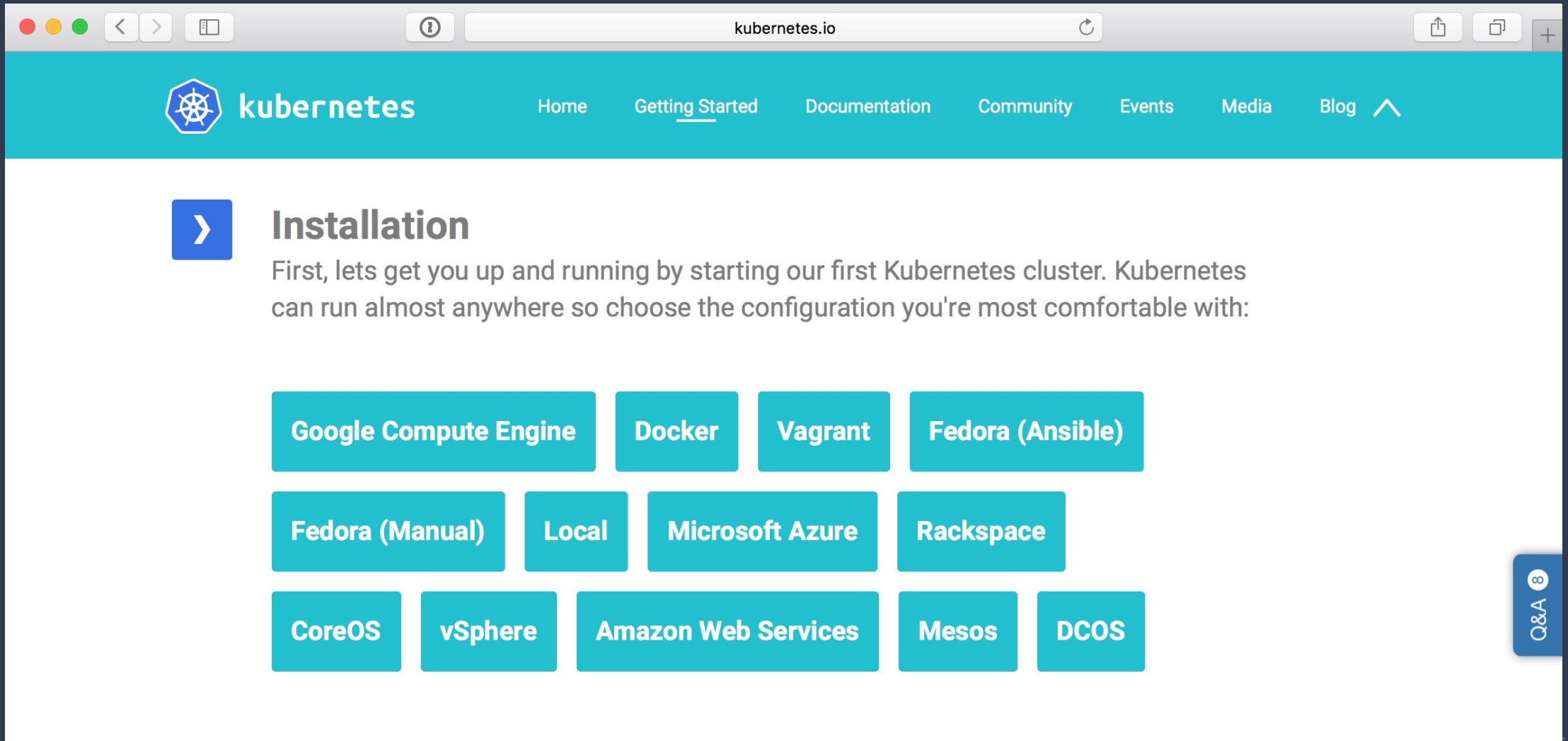
```
$ gcloud container clusters create demo-cluster --num-nodes=2
```

```
$ gcloud config set container/cluster demo-cluster
```

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	1/1	Running	0	1m

Getting Started Guides



The screenshot shows a web browser window with the URL `kubernetes.io`. The page features a teal header with the Kubernetes logo and navigation links: Home, Getting Started (underlined), Documentation, Community, Events, Media, and Blog. Below the header, a blue square icon with a white right-pointing chevron is followed by the section title **Installation**. The text below reads: "First, lets get you up and running by starting our first Kubernetes cluster. Kubernetes can run almost anywhere so choose the configuration you're most comfortable with:". Below this text is a grid of 13 teal buttons with white text, arranged in three rows. The buttons are: Google Compute Engine, Docker, Vagrant, Fedora (Ansible), Fedora (Manual), Local, Microsoft Azure, Rackspace, CoreOS, vSphere, Amazon Web Services, Mesos, and DCOS. On the right side of the page, there is a vertical blue sidebar with the text "Q&A" and a circular icon containing the number "8".

Installation

First, lets get you up and running by starting our first Kubernetes cluster. Kubernetes can run almost anywhere so choose the configuration you're most comfortable with:

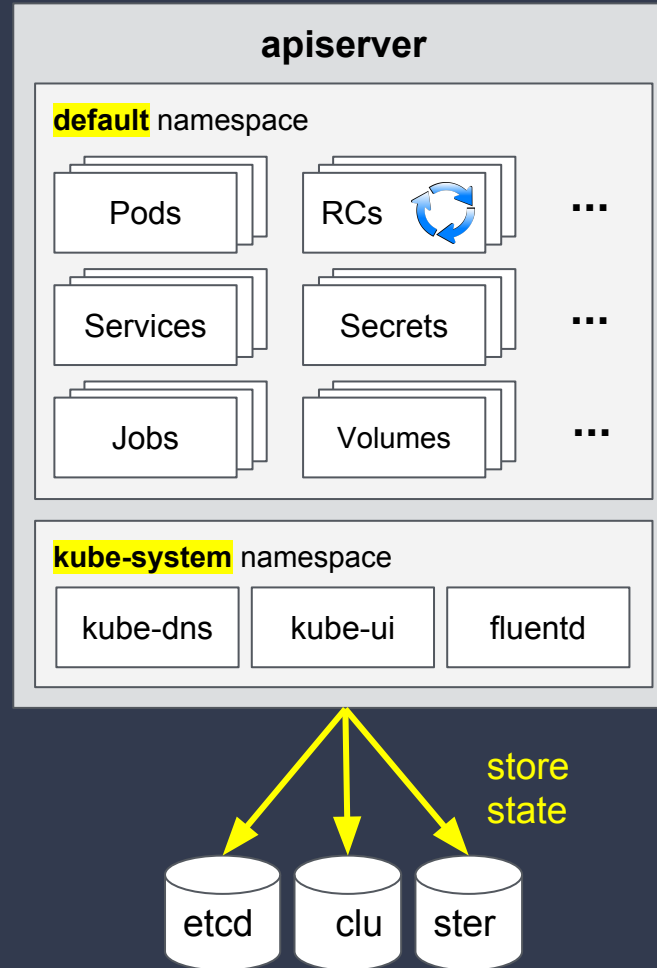
- Google Compute Engine
- Docker
- Vagrant
- Fedora (Ansible)
- Fedora (Manual)
- Local
- Microsoft Azure
- Rackspace
- CoreOS
- vSphere
- Amazon Web Services
- Mesos
- DCOS

Q&A 8

Get a Kubernetes cluster: mesos/docker with kube-up

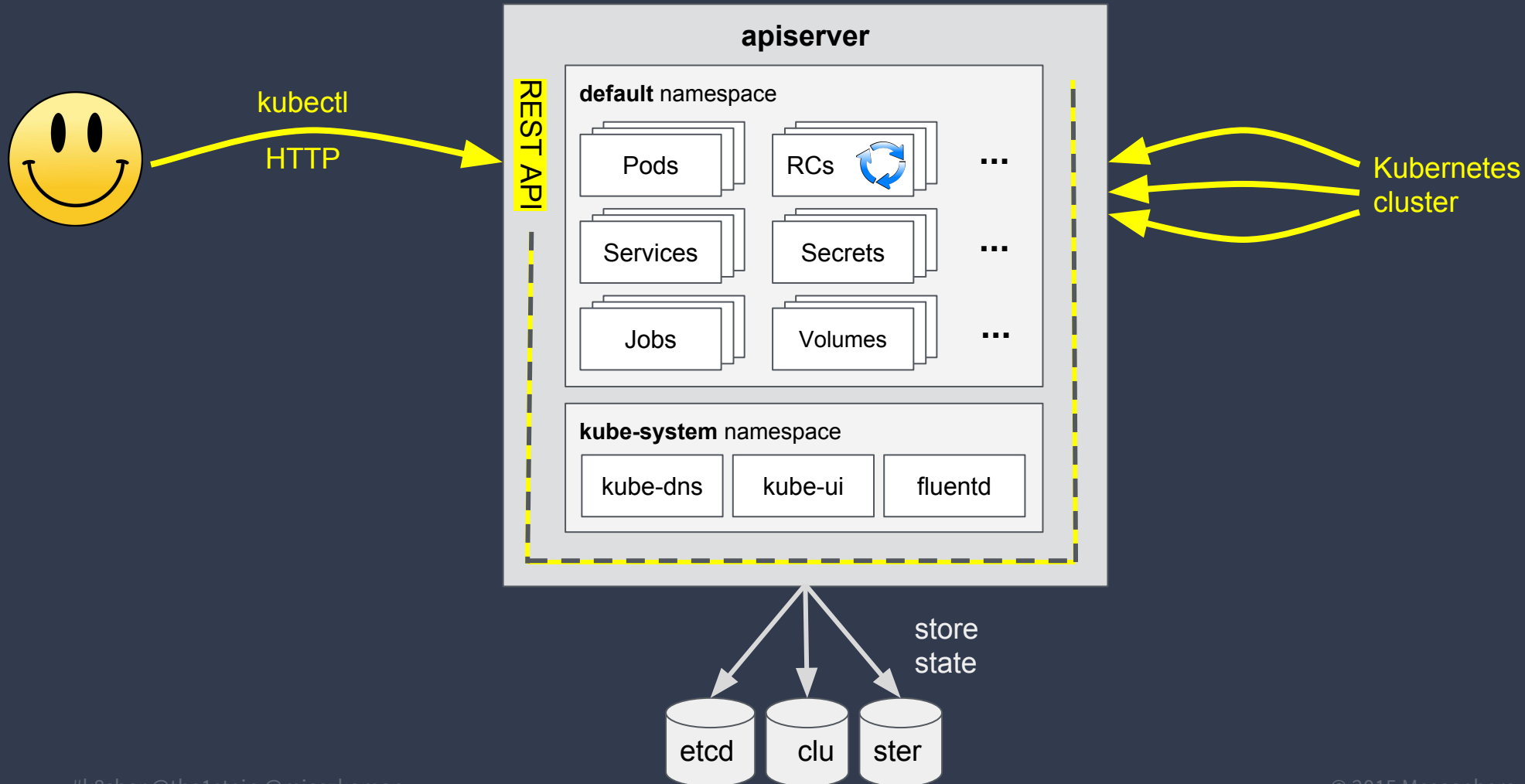
```
$ git clone git@github.com:kubernetes/kubernetes.git
$ cd kubernetes
$ build/run.sh hack/build-go.sh && make
$ KUBERNETES_PROVIDER=mesos/docker cluster/kube-up.sh
$ alias kubectl=_output/local/bin/darwin/amd64/kubectl
$ kubectl get pods
```

Kubernetes Core Architecture - the apiserver

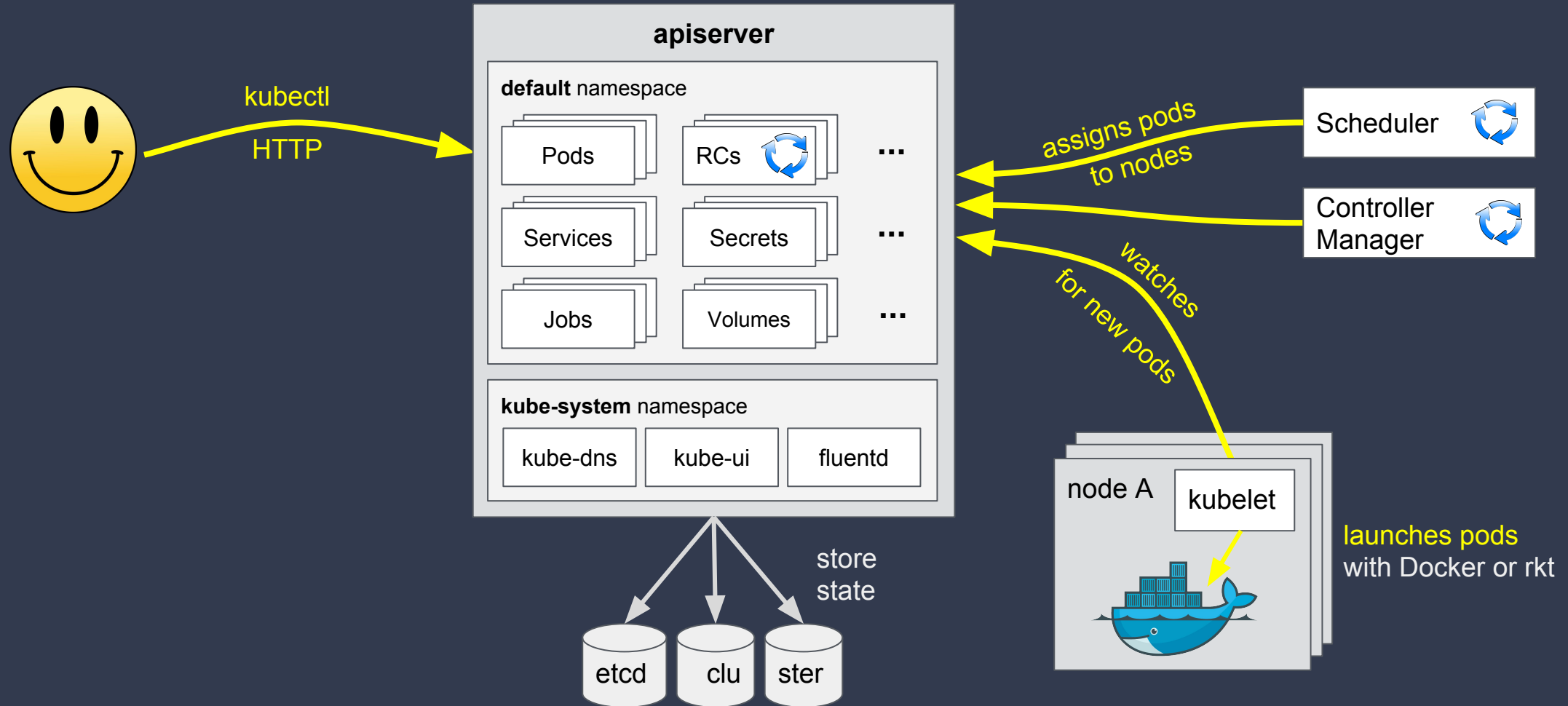


- all the **cluster state**
- **no logic**, only storage and API

Kubernetes Core Architecture - the apiserver



Kubernetes Core Architecture - the components



kubectl - interacting with a Kubernetes cluster

Usage:

```
kubectl [flags]
kubectl [command]
```

Available Commands:

get	Display one or many resources
describe	Show details of a specific resource or group of resources
create	Create a resource by filename or stdin
replace	Replace a resource by filename or stdin.
patch	Update field(s) of a resource by stdin.
delete	Delete resources by filenames, stdin, resources and names, or by resources and label selector.
edit	Edit a resource on the server
apply	Apply a configuration to a resource by filename or stdin
namespace	SUPERSEDED: Set and view the current Kubernetes namespace
logs	Print the logs for a container in a pod.
rolling-update	Perform a rolling update of the given ReplicationController.
scale	Set a new size for a Replication Controller.
attach	Attach to a running container.
exec	Execute a command in a container.
port-forward	Forward one or more local ports to a pod.
proxy	Run a proxy to the Kubernetes API server
run	Run a particular image on the cluster.
stop	Deprecated: Gracefully shut down a resource by name or filename.
expose	Take a replication controller, service or pod and expose it as a new Kubernetes Service
autoscale	Auto-scale a replication controller

Same Pod in Kubernetes

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
    - name: webserver
      image: ubuntu
      workingDir: /tmp/mypod
      command: ["python3", "-m", "http.server", "80"]
      volumeMounts:
        - name: htdocs
          mountPath: /tmp/mypod
    - name: busybox
      image: sttts/busybox-curl
      tty: true
      stdin: true
      volumeMounts:
        - name: htdocs
          mountPath: /tmp/mypod
  volumes:
    - name: htdocs
      hostPath:
        path: /tmp/mypod
```

#k8sber @the1stein @mieszkoman

```
$ kubectl create -f mypod.yaml
pod "mypod" created
```

Documentation

Documentation landing page: <http://kubernetes.io/v1.1>

The screenshot shows the Kubernetes v1.1 documentation landing page. The top navigation bar is blue with the Kubernetes logo and links to Home, Getting Started, Documentation (underlined), Community, Events, Media, and Blog. Below the navigation bar is a Google Custom Search bar with a Search button. The left sidebar contains a version selector set to v1.1, followed by links to Version 1.1, What is Kubernetes?, Getting Started (with an Overview link), and User's Guide (with an Application Administration link and an Overview and Concepts link). The main content area features the heading "What is Kubernetes?" followed by a paragraph stating that Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts. Below this is a paragraph stating that with Kubernetes, you can quickly and efficiently respond to customer demand, followed by a bulleted list of three points: scaling applications, rolling out new features, and optimizing hardware use. The page ends with a partial sentence about the goal of fostering an ecosystem.

kubernetes

Home Getting Started Documentation Community Events Media Blog

Google™ Custom Search Search

Version: v1.1

Version 1.1

[What is Kubernetes?](#)

Getting Started

Overview

► Installing Kubernetes and Creating Clusters

User's Guide

Application Administration

► Overview and Concepts

Application Administration Prerequisites

What is Kubernetes?

Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts.

With Kubernetes, you are able to quickly and efficiently respond to customer demand:

- Scale your applications on the fly.
- Seamlessly roll out new features.
- Optimize use of your hardware by using only the resources you need.

Our goal is to foster an ecosystem of components and tools that relieve the burden of running applications in

Documentation - Schema

JSON/YAML/REST schema: <http://kubernetes.io/v1.1/docs/api-reference/v1/definitions.html>

Persistent Volumes

► Multi-tier Applications

Examples

► Setup and Configuration

Updating Live Pods

Reference

API Reference

► API Basics

► Kubernetes API Reference

Operations

Definitions

Extensions API:

Extensions:

Operations

Extensions:

Definitions

v1.Pod


Pod is a collection of containers that can run on a host. This resource is created by clients and scheduled onto hosts.













Name	Description	Required	Schema	Default
kind	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: http://kubernetes.io/v1.1/docs/devel/api-conventions.html#types-kinds	false	string	
apiVersion	APIVersion defines the versioned schema of this representation of an object.	false	string	

Examples - learn from them


Deployment examples: <https://github.com/kubernetes/kubernetes/tree/master/examples>

Branch: master ▾ **kubernetes** / **examples** /

 **k8s-merge-robot** Merge pull request #17932 from GertiPoppel/DocContribution ...

..	
 aws_ebs	run hack/update-generated-docs.sh
 blog-logging	run hack/update-generated-docs.sh
 cassandra	Correct permissions on cassandra run script and fix provider.
 celery-rabbitmq	run hack/update-generated-docs.sh
 cephfs	run hack/update-generated-docs.sh
 cluster-dns	run hack/update-generated-docs.sh
 elasticsearch	run hack/update-generated-docs.sh
 experimental/persistent-vol...	run hack/update-generated-docs.sh
 explorer	run hack/update-generated-docs.sh
 extensions	Horizontal pod autoscaling example moved to user guide.
 fibre_channel	run hack/update-generated-docs.sh
 flexvolume	Add support for flex volume. Flex volume adds support for thirdparty(...

Branch: master ▾ **kubernetes** / **examples** / **guestbook** / **all-in-one** / **guestbook-all-in-one.yaml**

 **linzichang** Update guestbook example according to config best practices

1 contributor

177 lines (176 sloc) | 3.93 KB Raw

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: redis-master
5    labels:
6      app: redis
7      tier: backend
8      role: master
9  spec:
10   ports:
11     # the port that this service should serve on
12     - port: 6379
13       targetPort: 6379
14   selector:
15     app: redis
16     tier: backend
17     role: master
18 ---
19 apiVersion: v1
20 kind: ReplicationController
21 metadata:
22   name: redis-master
```

Browsing the API

Open <https://192.168.65.65:6443/swagger-ui/>

GET

/api/v1/namespaces/{namespace}/endpoints

list or watch objects of kind Endpoints

POST

/api/v1/namespaces/{namespace}/endpoints

create a Endpoints

Response Class (Status)

Model

Model Schema

```
{
  "kind": "",
  "apiVersion": "",
  "metadata": {
    "name": "",
    "generateName": "",
    "namespace": "",
    "selfLink": "",
    "uid": "",
    "resourceVersion": "",
    "generation": 0
  }
}
```

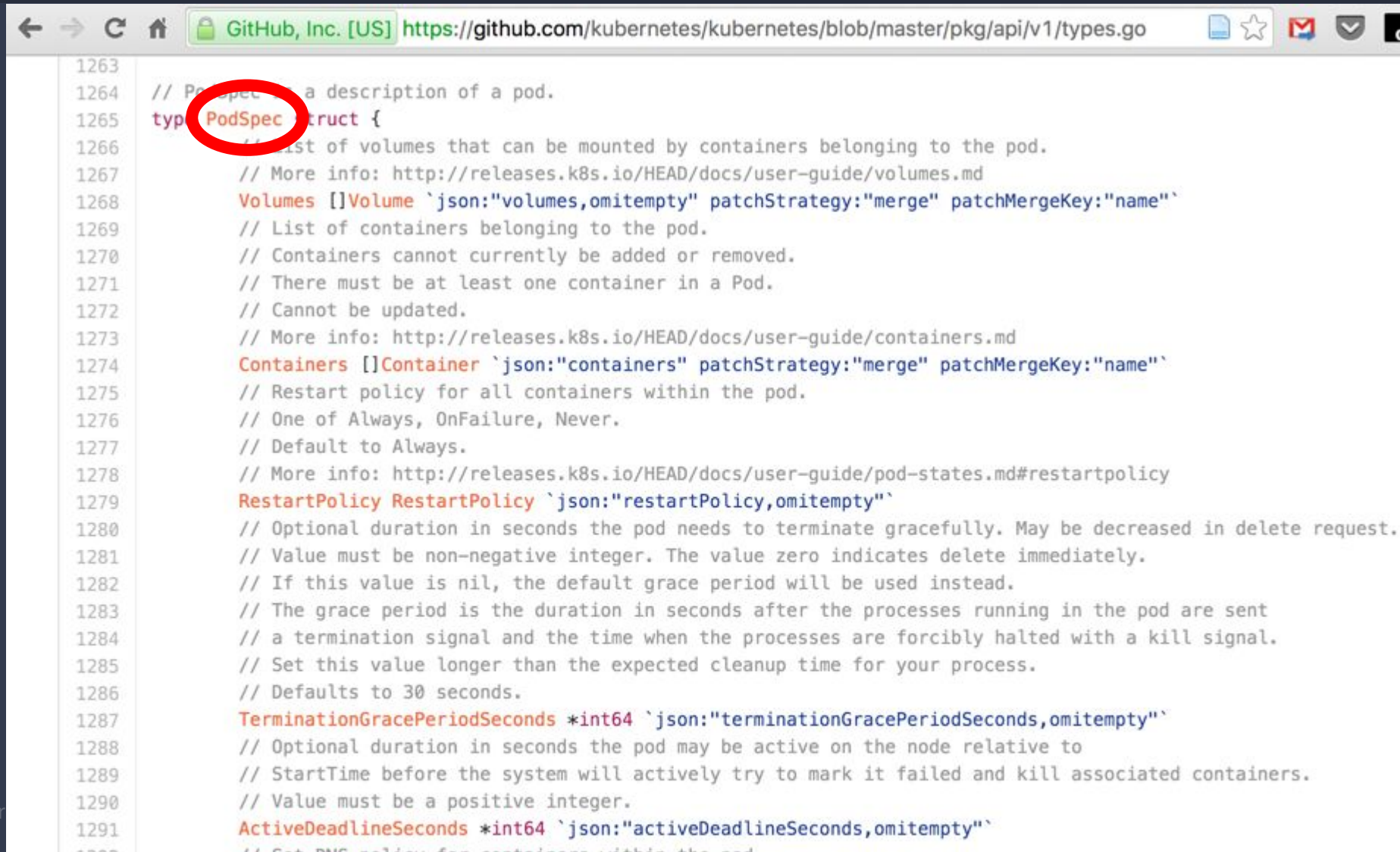
Response Content Type

application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
pretty	(empty)	If 'true', then the output	query	string

The source of truth: **api/v1/types.go**



```
1263 // PodSpec is a description of a pod.
1264 type PodSpec struct {
1265     // List of volumes that can be mounted by containers belonging to the pod.
1266     // More info: http://releases.k8s.io/HEAD/docs/user-guide/volumes.md
1267     Volumes []Volume `json:"volumes,omitempty" patchStrategy:"merge" patchMergeKey:"name"`
1268     // List of containers belonging to the pod.
1269     // Containers cannot currently be added or removed.
1270     // There must be at least one container in a Pod.
1271     // Cannot be updated.
1272     // More info: http://releases.k8s.io/HEAD/docs/user-guide/containers.md
1273     Containers []Container `json:"containers" patchStrategy:"merge" patchMergeKey:"name"`
1274     // Restart policy for all containers within the pod.
1275     // One of Always, OnFailure, Never.
1276     // Default to Always.
1277     // More info: http://releases.k8s.io/HEAD/docs/user-guide/pod-states.md#restartpolicy
1278     RestartPolicy RestartPolicy `json:"restartPolicy,omitempty"`
1279     // Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request.
1280     // Value must be non-negative integer. The value zero indicates delete immediately.
1281     // If this value is nil, the default grace period will be used instead.
1282     // The grace period is the duration in seconds after the processes running in the pod are sent
1283     // a termination signal and the time when the processes are forcibly halted with a kill signal.
1284     // Set this value longer than the expected cleanup time for your process.
1285     // Defaults to 30 seconds.
1286     TerminationGracePeriodSeconds *int64 `json:"terminationGracePeriodSeconds,omitempty"`
1287     // Optional duration in seconds the pod may be active on the node relative to
1288     // StartTime before the system will actively try to mark it failed and kill associated containers.
1289     // Value must be a positive integer.
1290     ActiveDeadlineSeconds *int64 `json:"activeDeadlineSeconds,omitempty"`
1291     // Set DNS policy for containers within the pod.
```

On the wire (JSON only)

```
{
  "kind": "Pod",
  "apiVersion": "v1",
  "metadata": {
    "name": "mypod",
    "namespace": "default",
    "selfLink": "/api/v1/namespaces/default/pods/mypod",
    "uid": "37d657dc-b47a-11e5-938a-0242c0a84127",
    "resourceVersion": "5792",
    "creationTimestamp": "2016-01-06T13:34:32Z",
    "annotations": {
      "k8s.mesosphere.io/bindingHost": "192.168.65.44",
      "k8s.mesosphere.io/executorId": "cf3cd4adb282b475_k8sm-executor",
      "k8s.mesosphere.io/offerId": "20160106-081826-641837248-5050-1-03680",
      "k8s.mesosphere.io/slaveId": "20160106-081826-641837248-5050-1-S1",
      "k8s.mesosphere.io/taskId": "pod.380df08e-b47a-11e5-8201-0242c0a8412b"
    }
  },
  "spec": {
    "volumes": [
      {
        "name": "htdocs",
        "hostPath": {
          "path": "/tmp/mypod"
        }
      }
    ],
    {
      "name": "default-token-b3a3b",
      "secret": {
```

```
$ curl -k -u admin:admin \
https://192.168.65.39:6443/api/v1/namespaces/default/pods/mypod
```



Labels & Annotations

- in **metadata block** of every API resource

```
"labels": {  
  "git": "cb52d79578b1379e32b7e6a119ed16232ef1b13b",  
  "env": "prod",  
  "sla": "super-premium",  
  "app": "webshop",  
  "tier": "frontend"  
}
```

for
user/admin

```
"annotations": {  
  "k8s.mesosphere.io/bindingHost": "192.168.65.44",  
  "k8s.mesosphere.io/executorId": "k8sm-executor",  
  ...  
}
```

for tooling

- labels can be used **to filter objects** serverside (annotations can't)

```
$ kubectl get nodes -l gen=2011
```

Getting the Pod IP

```
$ kubectl get pod mypod -o yaml
```

```
...
```

```
hostIP: 192.168.65.68
```

```
phase: Running
```

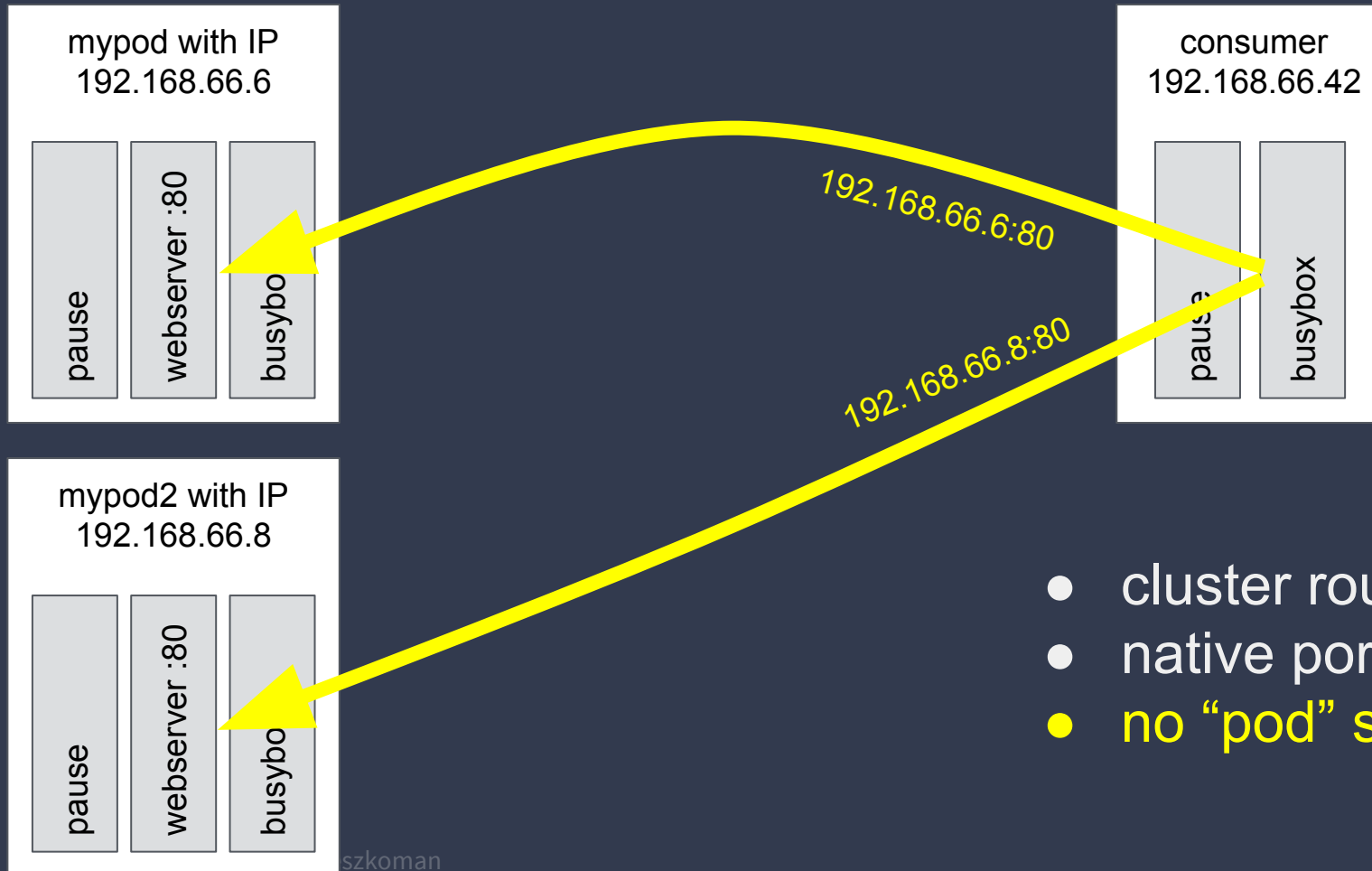
```
podIP: 192.168.66.11
```

```
startTime: 2016-01-07T08:33:55Z
```

```
$ kubectl get pod mypod -o template --template='{{.status.podIP}}'
```

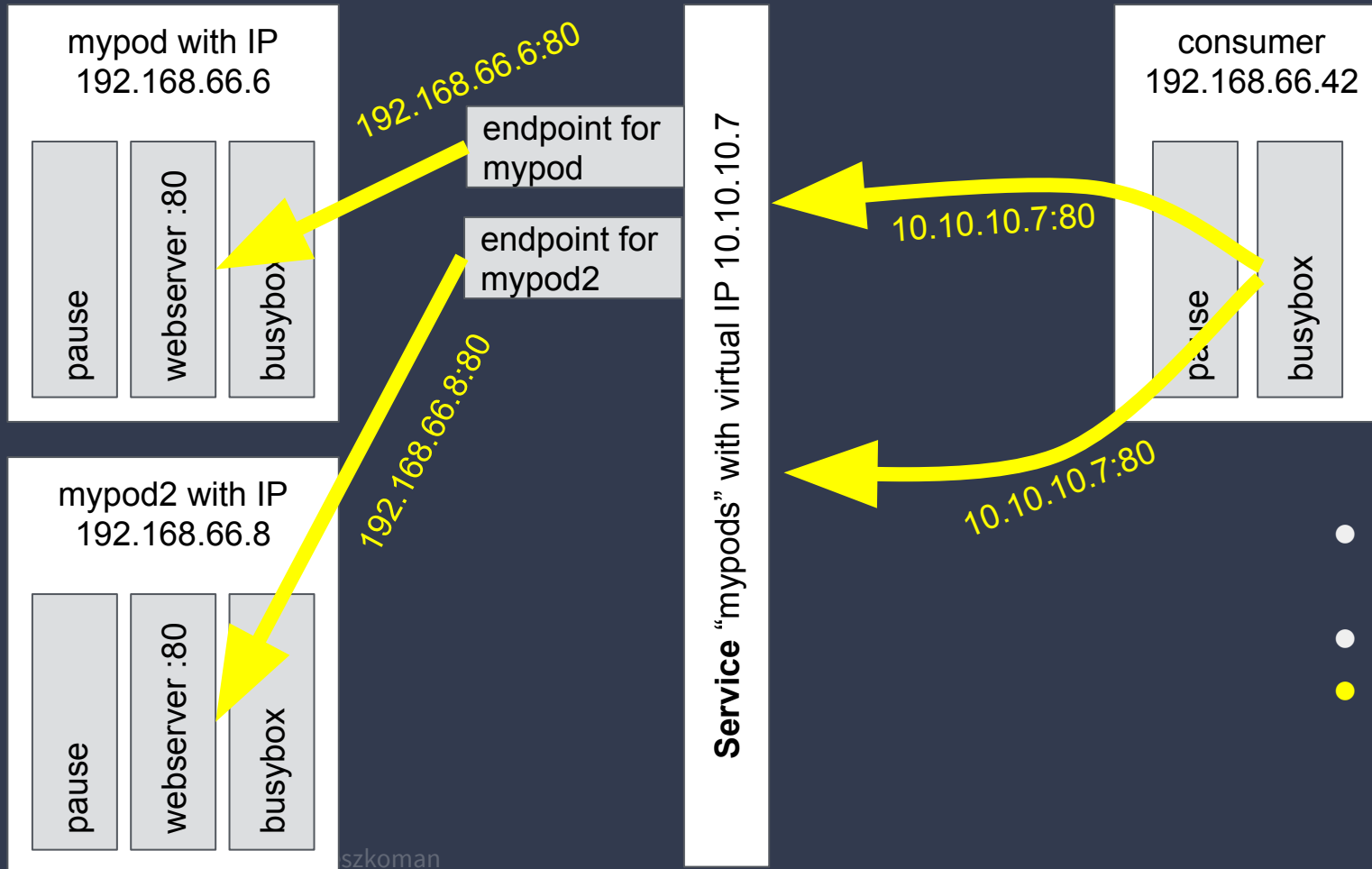
```
192.168.66.12
```

From Pod to Pod



- cluster routable IPs
- native ports without conflicts
- no “pod” service discovery available

A Service



- a service as a **static API object**
\$ kubectl create -f service.yaml
- virtual, but **static IP** (usually 10.x.y.z)
- **no service discovery necessary**



Defining a Service

```
$ cat mypods-service.yaml
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: mypods
```

```
spec:
```

```
  ports:
```

```
  - port: 80
```

```
  selector:
```

```
    name: mypod
```

- match pods with these labels
- a service without any pod is completely valid
- services exist on their own

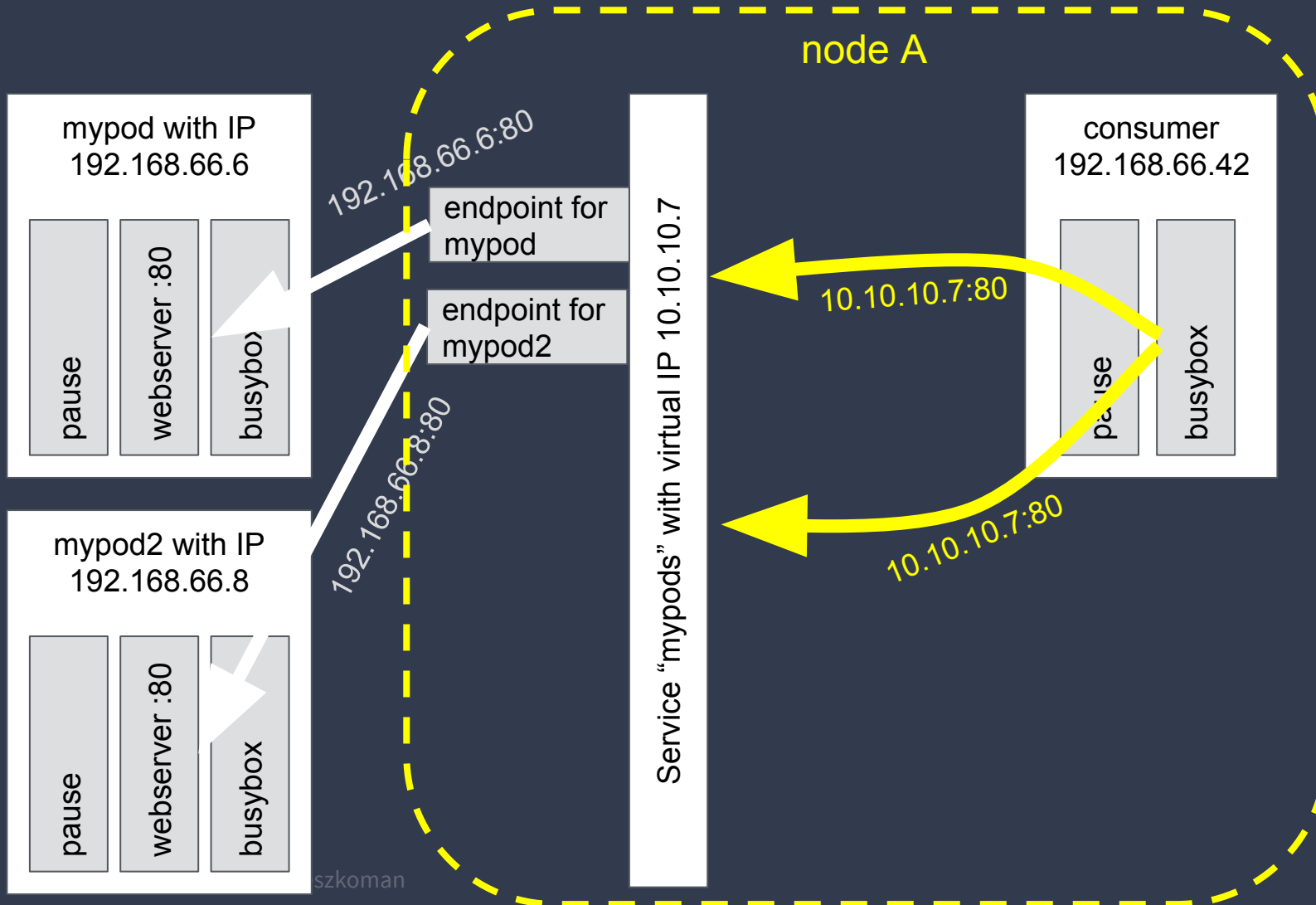
```
$ kubectl create -f mypods-service.yaml
```

```
service "mypods" created
```

```
$ kubectl get services
```

NAME	CLUSTER_IP	EXTERNAL_IP	PORT(S)	SELECTOR	AGE
k8sm-scheduler	10.10.10.34	<none>	10251/TCP	<none>	6h
kubernetes	10.10.10.1	<none>	443/TCP	<none>	6h
mypods	10.10.10.32	<none>	80/TCP	name=mypods	5m

A Service, available on each node

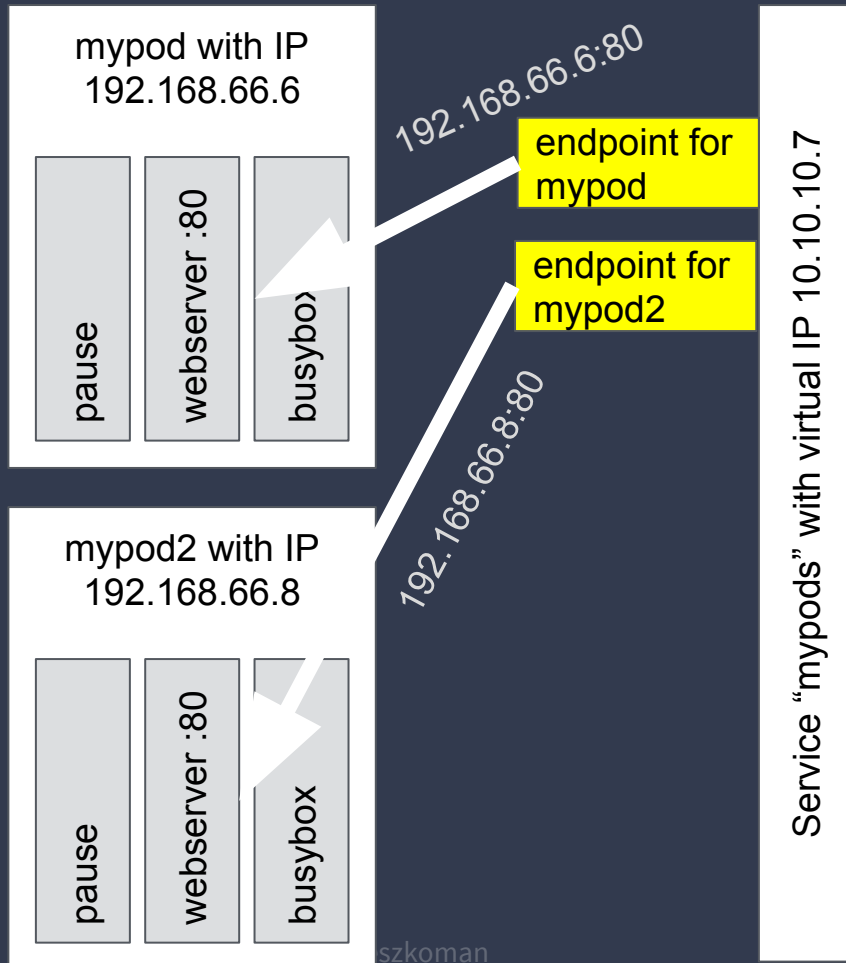


- Implemented by **kube-proxy**
 - in user-mode (simple proxy)
 - or using iptables
- **on every node!**

! traffic from a pod is intercepted on the host of the pod, not the destination of the connection.

Note: In addition kube-proxy makes services available for other hosts, e. g. for internet clients

Service Endpoints



- 1 endpoint for each matching pod of a service
- endpoint = pod-IP:port, e.g. 192.168.66.6:80

```
$ kubectl get endpoints kube-ui
```

NAME	ENDPOINTS	AGE
mypods	192.168.65.68:8001	2s

Special in Kuberentes-Mesos because it runs without overlay network here. Otherwise, :80 would be visible here.

Behind the scenes - Linux iptables

```
$ iptables -t nat -n -L
```

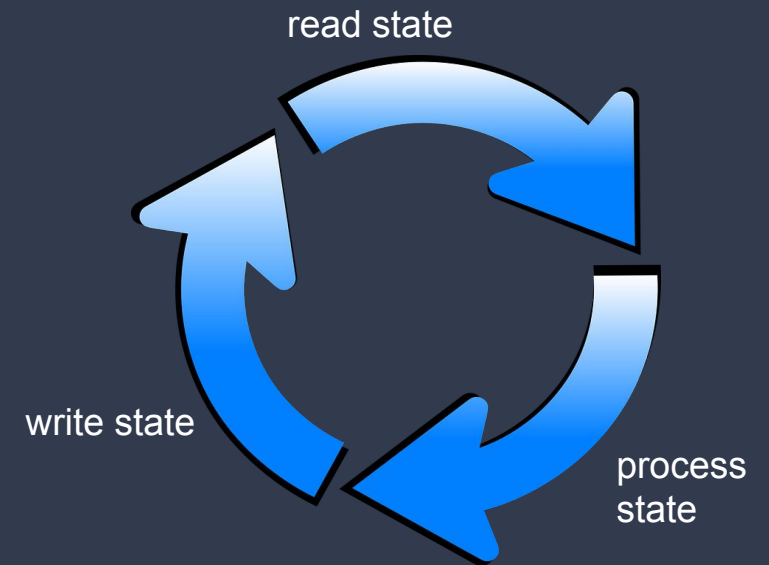
```
...
```

```
Chain KUBE-PORTALS-CONTAINER (1 references)
```

target	prot	opt	source	destination	
REDIRECT	tcp	--	0.0.0.0/0	10.10.10.3	/* kube-system/kube-ui: */ tcp dpt:80 redir ports 45123
REDIRECT	tcp	--	0.0.0.0/0	10.10.10.181	/* default/k8sm-scheduler: */ tcp dpt:10251 redir ports 59722
REDIRECT	tcp	--	0.0.0.0/0	10.10.10.1	/* default/kubernetes: */ tcp dpt:443 redir ports 55836
REDIRECT	udp	--	0.0.0.0/0	10.10.10.10	/* kube-system/kube-dns:dns */ udp dpt:53 redir ports 57016
REDIRECT	tcp	--	0.0.0.0/0	10.10.10.10	/* kube-system/kube-dns:dns-tcp */ tcp dpt:53 redir ports 42340

State plus Control

- **state** consists of **all API objects**, stored on the API-Server
- core **Kubernetes architecture principle**:
logic is in **control loops** (“controllers”) which
 - are **state-less** (can recover from failure)
 - and **decoupled** (communication via API-Server)



A Control Loop to run Pods

INPUT:

REPLICAS: number of mypods that should be running

ALGORITHM

1. `running` := number of running pods with label `name=mypod`
2. **If** `running` > `REPLICAS` => delete some mypods
3. **If** `REPLICAS` > `running` => launch some mypods
4. **Goto** 1

Getting # of running mypods

```
REPLICAS = int(sys.argv[1])

master = os.getenv("KUBERNETES_MASTER", "http://localhost:8080")
kubeclient = toolkit.KubeHTTPClient(api_server=master, debug=False)

while True:
    # count mypods
    response = kubeclient.execute_operation(method="GET",
        ops_path="/api/v1/namespaces/default/pods?labelSelector=name%3Dmypod")
    mypods = response.json()['items']
    running_mypods = len(mypods)
    print "{} running".format(running_mypods)

    ...
```

If running > REPLICAS => delete some mypods

```
if running_mypods > REPLICAS:
    to_delete = running_mypods - REPLICAS
    print " Too many are running. Deleting {} pods:".format(to_delete)
    for pod in mypods[:to_delete]:
        print "    Deleting pod {}".format(pod['metadata']['name'])
        kubectl.delete_resource(pod['metadata']['selfLink'])
```

If REPLICAS > running => launch some mypods

```
elif REPLICAS > running_mypods:
    to_launch = REPLICAS - running_mypods

    for n in range(0, to_launch):
        mypod_spec, _ = util.load_yaml(filename="mypod.yaml")
        mypod_spec["metadata"]["name"] += "-" + shortuuid.uuid()[0:4].lower()

        print "Launching pod {}".format(mypod_spec["metadata"]["name"])
        response = kubectl.execute_operation(method='POST',
            ops_path = "/api/v1/namespaces/default/pods",
            payload = util.serialize_tojson(mypod_spec))
```

Kubernetes Replication Controllers - “RCs”

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: mypod
spec:
  replicas: 10
  selector:
    name: mypod
  template:
    metadata:
      labels:
        name: mypod
    spec:
      containers:
      - name: webserver
        image: ubuntu
        ...
```

```
$ kubectl create -f mypod-rc.yaml
```

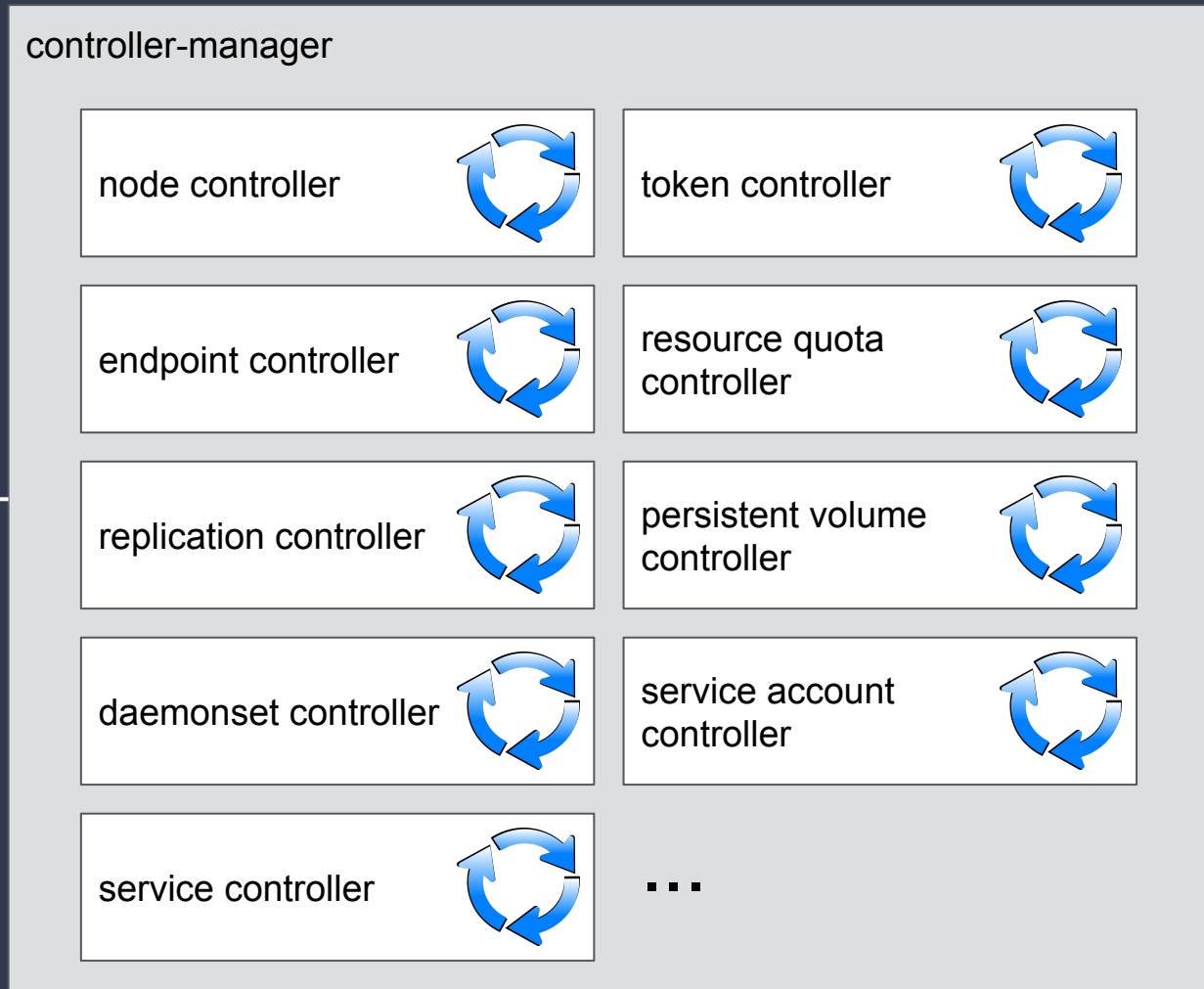
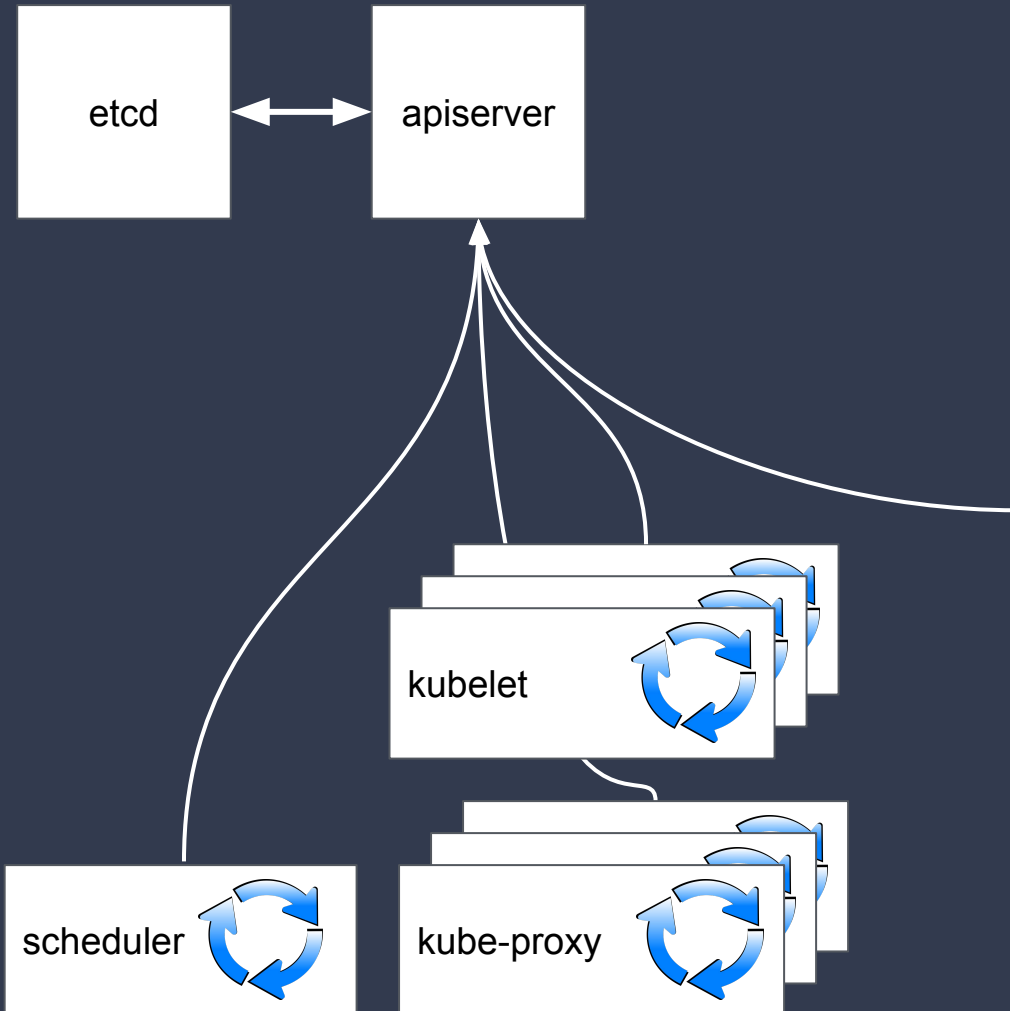
```
$ kubectl scale --replicas=5 rc/mypod
```


Other Controllers

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: mypod
spec:
  selector:
    name: mypod
  template:
    metadata:
      labels:
        name: mypod
    spec:
      nodeSelector:
        env=dmz
      containers:
        - name: webserver
          image: ubuntu
          ...
```

- **DaemonSets** run pods once per node
- **MyCassandraController** might run Cassandra instances
- **MyGaleraController** might run MySQL Galera instances
- **MyHaproxyUpdateController** might write haproxy.conf
- **MyNodeUpdateController** might do rolling security updates on nodes or of containers
- **MyBitcoinScheduler** might schedule bitcoin pods for spare resources

Controllers, controllers everywhere ...



Of course: we are hiring in Hamburg, Berlin & San Francisco!



Watches

```
$ kubectl get pods -l "name=mypod" -w --v=9
```

=> GET <https://192.168.65.39:6443/api/v1/watch/namespaces/default/pods?labelSelector=name%3Dmypod>

```
{"type":"ADDED","object":{"kind":"Pod","apiVersion":"v1", ...}}
```

```
{"type":"MODIFIED","object":{"kind":"Pod","apiVersion":"v1", ...}}
```

```
{"type":"DELETED","object":{"kind":"Pod","apiVersion":"v1",...}}
```