



PhraseApp

Continuous Delivery with Kubernetes

Tobias Schwab, phraseapp.com



Container Scheduler?

Well, yes, but...

... there is a little bit more

Framework to build **Distributed Applications**

REST API

CNCF: independent from commercial vendors

Linux Kernel for the Cloud?

Getting Started

- [Google Container Engine \(GKE\)](#)
- [minikube](#)
- [kubeadm](#) (alpha in v1.4)
- ...
- [Kubernetes the Hard Way](#)

Pods

Pods

Set of tightly coupled containers running on a **single node**

Pod: Features

- dedicated ip
- shared network interface
- shared file system

Pod: Manifest

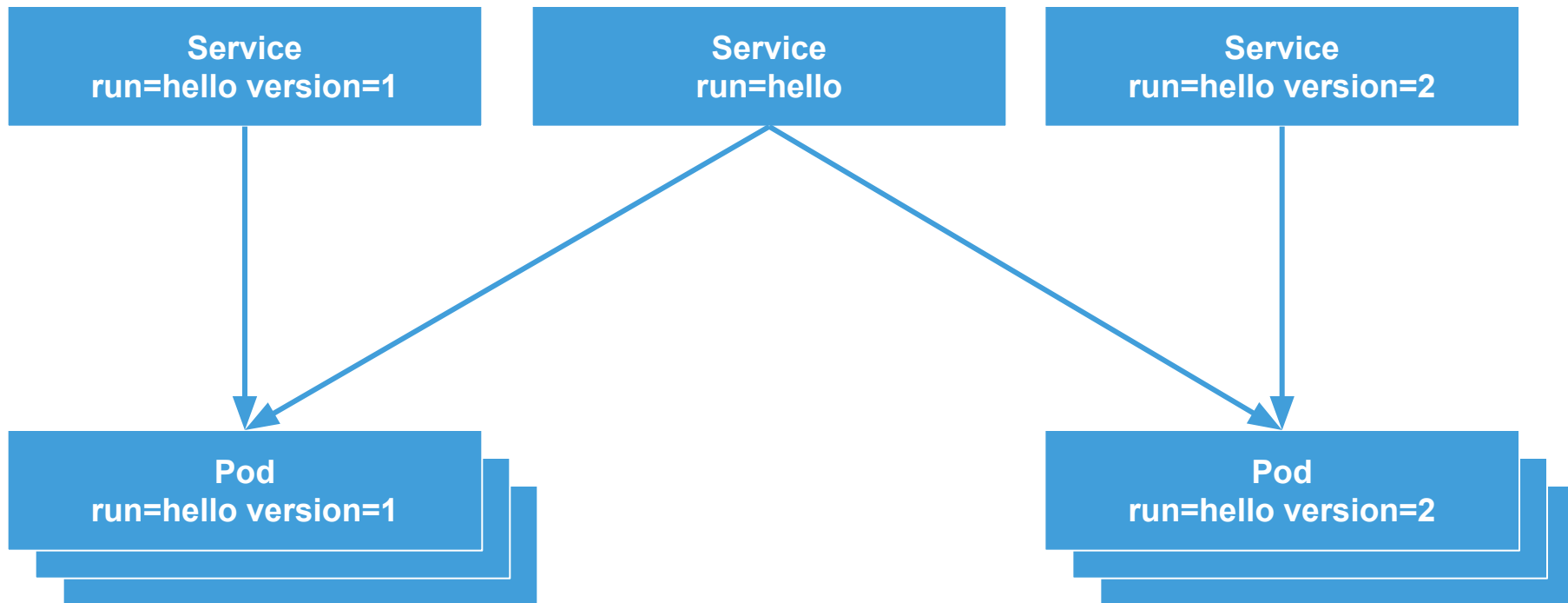
```
kind: Pod
apiVersion: v1
metadata:
  name: hello
  labels:
    run: hello
spec:
  containers:
  - name: hello
    image: quay.io/tobstarr/hello:v1
```

Services

Services

Policy to access a **logical set** of pods

Services



Service: Manifest

```
apiVersion: v1
kind: Service
metadata:
  labels:
    run: hello
  name: hello
spec:
  ports:
    - port: 80
      targetPort: 8080
      protocol: TCP
  selector:
    run: hello
```

Deployments

declarative Pod updates

- image
- ENV
- configuration files
- labels

Replica Sets

Fixed number of pod **replicas** running on **multiple nodes**

Configuration

ConfigMap: **plain text** configuration

- e.g. redis.conf, nginx.conf

Secret: **sensitive** information

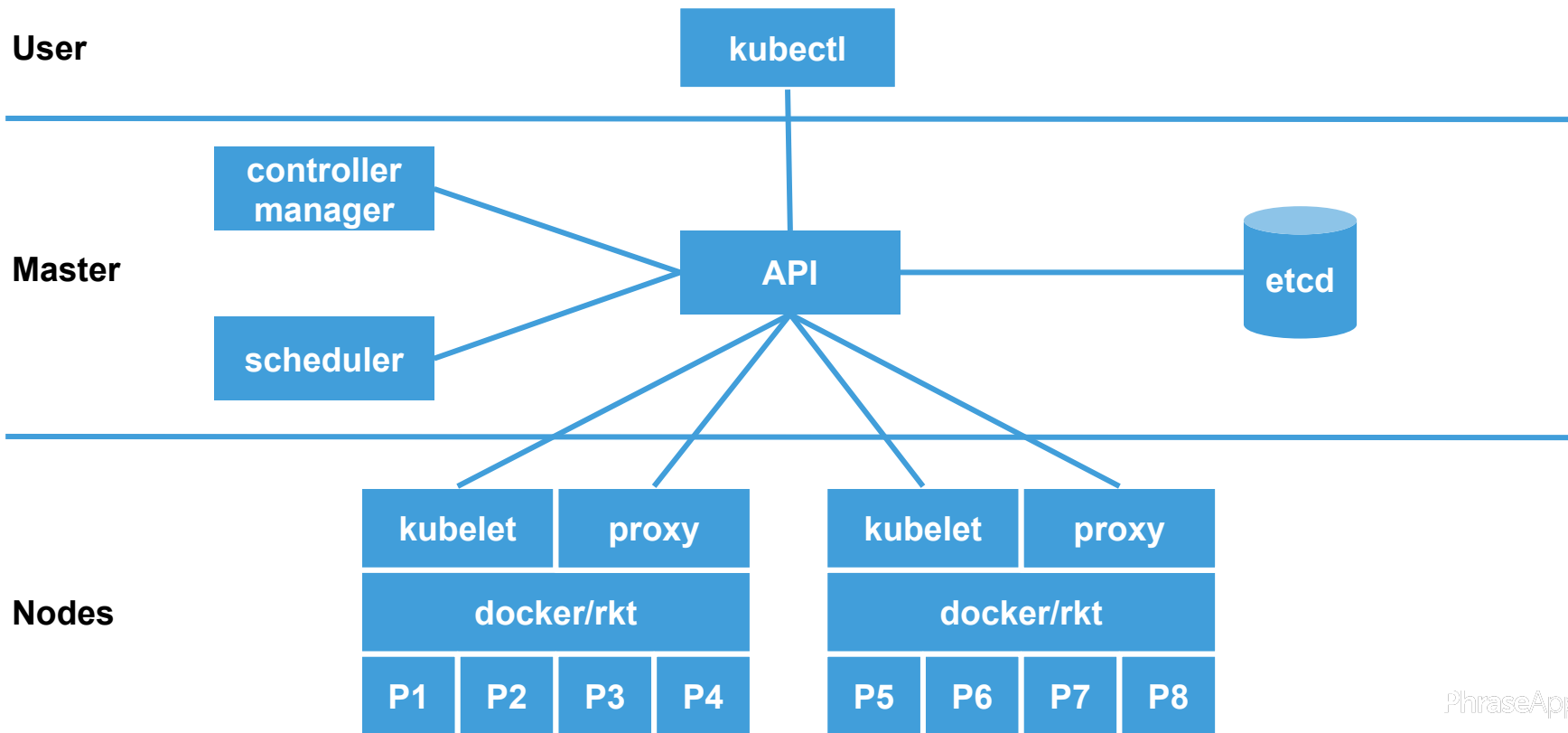
- e.g. TLS certs, registry credentials, etc.

Volumes

Persistent storage beyond container lifetime

- configMap
- secrets
- hostPath
- gcePersistentDisk
- awsElasticBlockStore
- nfs

Architecture



Continuous Delivery

Continuous Delivery

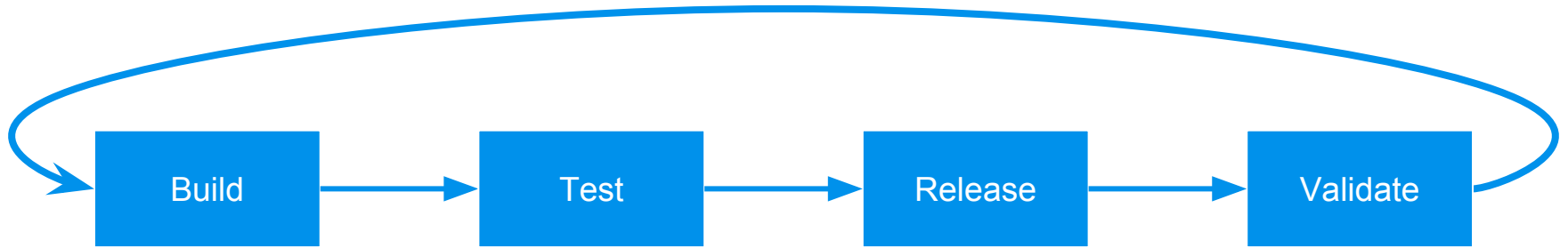
build, test, and release software **faster** and **more frequently**

Continuous Delivery

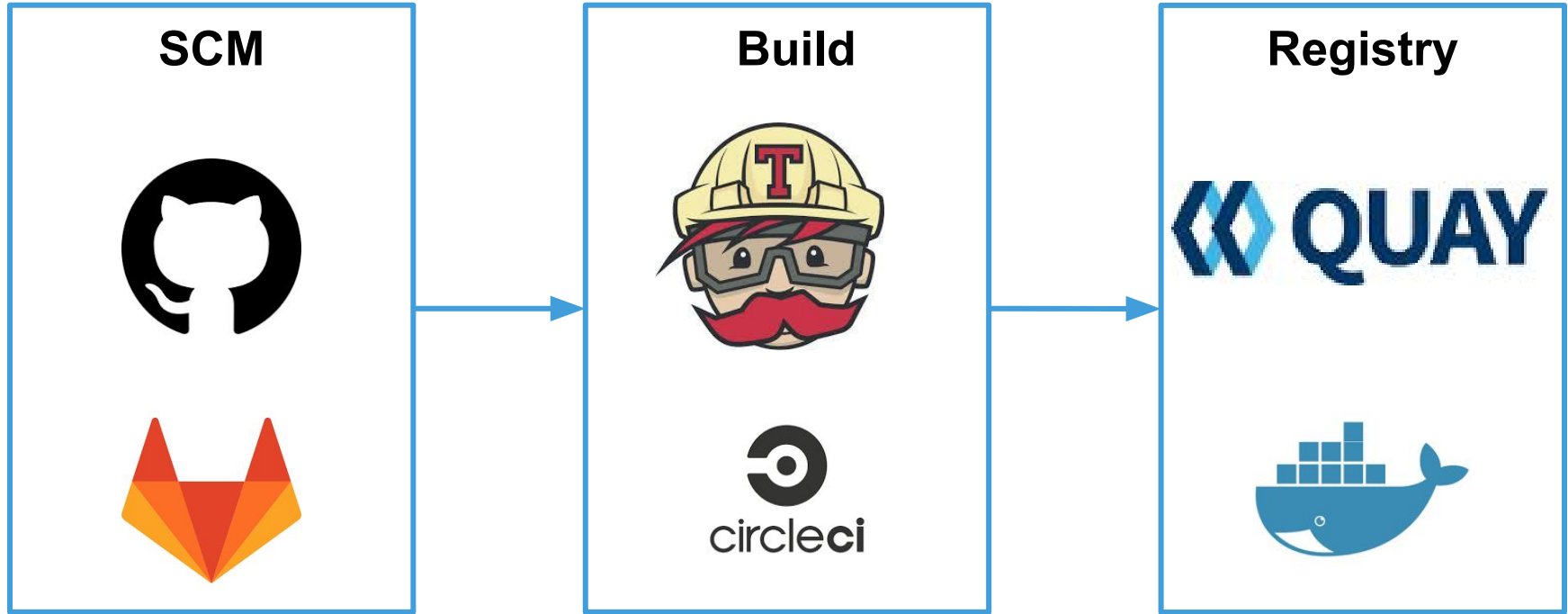
Our Philosophy:

- everyone **can** deploy any time
- you build it, you run it!
- **automation!**

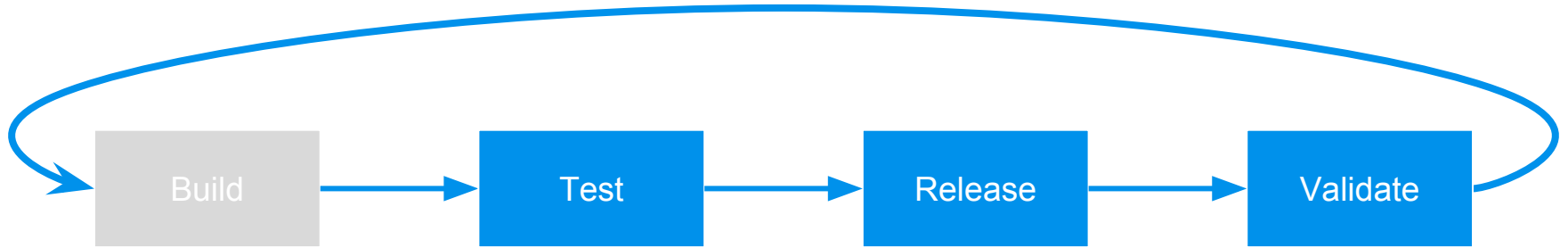
Continuous Delivery Pipeline



Continuous Delivery Pipeline



Build



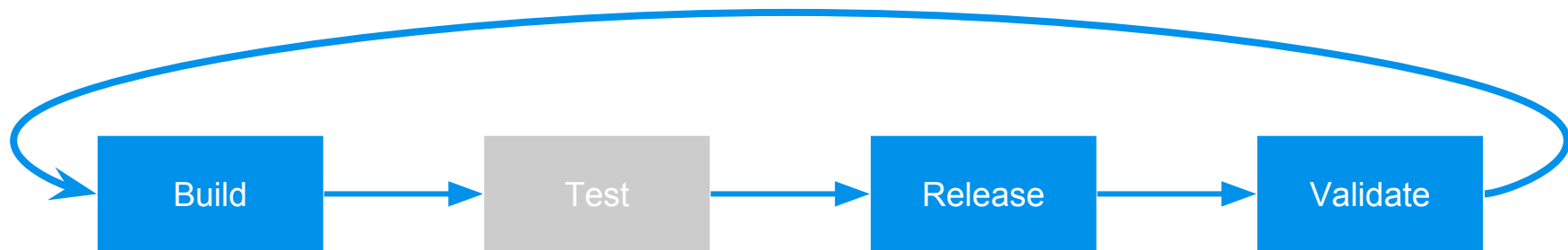
Build

- compile
- bundle assets
- build container image
- push container image to registry

Build: Kubernetes

- build container with
 - /usr/bin/docker
 - /var/run/docker.sockmounted from k8s node
- private docker registry deployment
- registry exposed via *NodePort* service

Test



Test

- unit tests
- integration tests
- acceptance tests

Test: Kubernetes

unit tests:

- **test container**

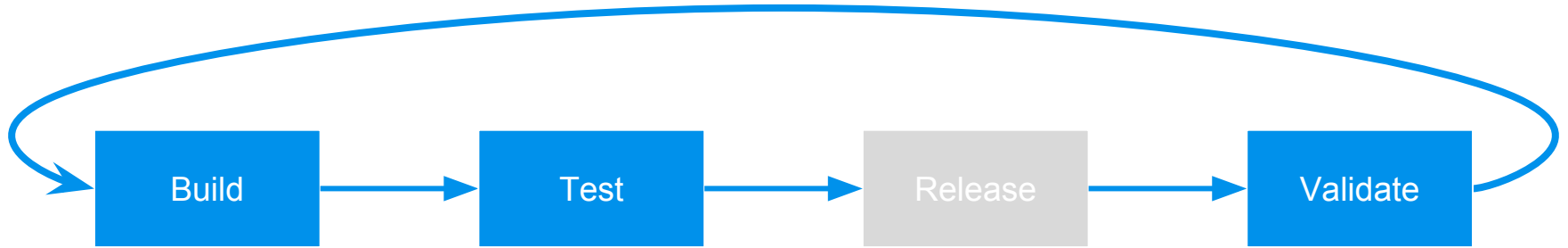
integration tests:

- **test pod with dependencies**

acceptance tests:

- **test pod with e.g. selenium**


Release



Release

```
kubectl set image ...
```

Jenkins Build Pipeline


 **Jenkins**

[Jenkins](#) > [hello](#) > [master](#) > [ENABLE AUTO REFRESH](#)

[Up](#)
[Status](#)
[Changes](#)
[Build Now](#)
[View Configuration](#)
[Full Stage View](#)
[Pipeline Syntax](#)

Pipeline master

Full project name: hello/master

 [Recent Changes](#)

Stage View

Average stage times:

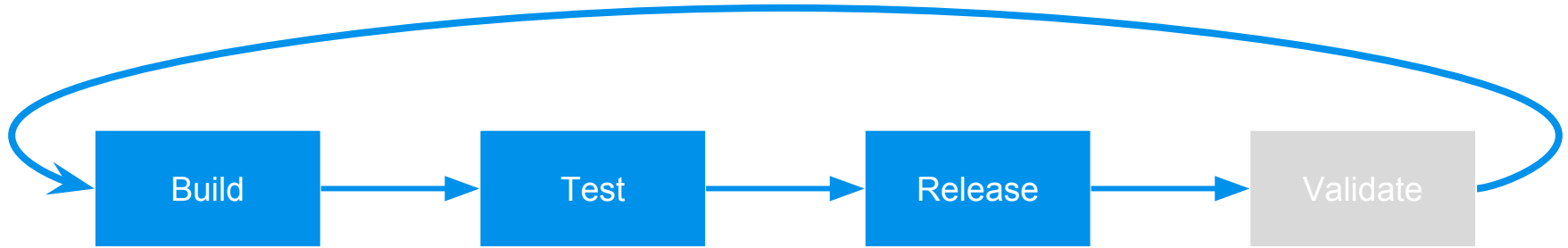
	SCM	Build	Test	Push	Release
#28 Sep 29 15:45 1 commits	915ms	3s	2s	271ms	6s
#27 Sep 29 15:43 2 commits	748ms	4s	2s	549ms	56s aborted
#26 Sep 29 15:27 1 commits	773ms	5s	2s	291ms	281ms
#25 Sep 29 15:25 1 commits	854ms	4s	2s	284ms	280ms

Build History

[trend](#)

- #28 Sep 29, 2016 1:45 PM
- #27 Sep 29, 2016 1:43 PM
- #26 Sep 29, 2016 1:27 PM
- #25 Sep 29, 2016 1:25 PM
- #24 Sep 29, 2016 1:23 PM
- #23 Sep 29, 2016 1:21 PM
- #22 Sep 29, 2016 1:20 PM
- #21 Sep 29, 2016 12:54 PM
- #20 Sep 29, 2016 12:53 PM

Validate



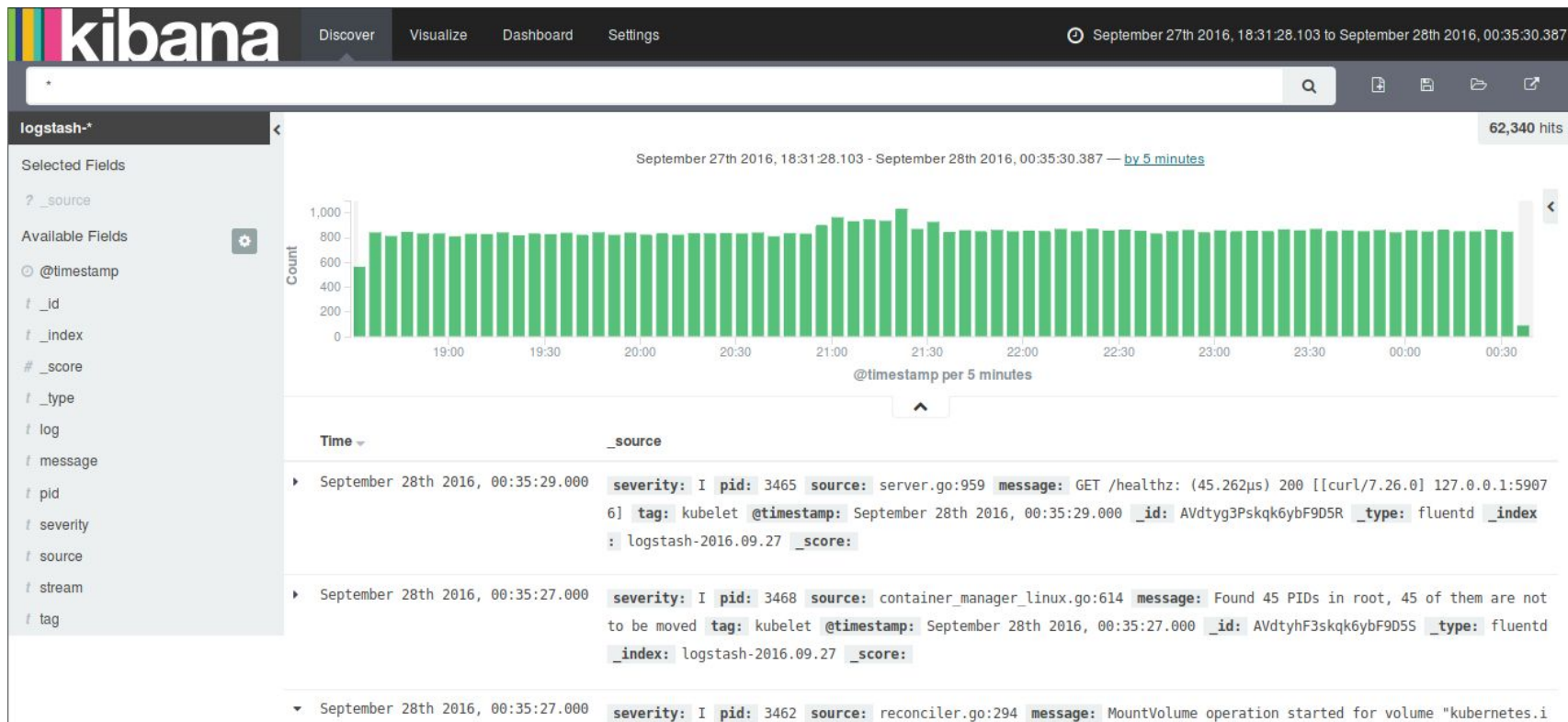
Logging

- apps log to **stdout**
- containers log to local file system
- [fluentd](#)
- [elasticsearch](#)
- [kibana](#)

Logging

- *DaemonSet*: Pod running on selected **nodes** in a cluster
- mount node FS via *VolumeMount*
- fluentd sends logs to ElasticSearch
- display/search with Kibana

Kibana



Monitoring

Cadvisor

- expose container metrics
- running on k8s nodes by default

Prometheus

- pull based monitoring & alerting
- <https://coreos.com/blog/prometheus-and-kubernetes-up-and-running.html>

Prometheus

Prometheus Alerts Graph Status - Help

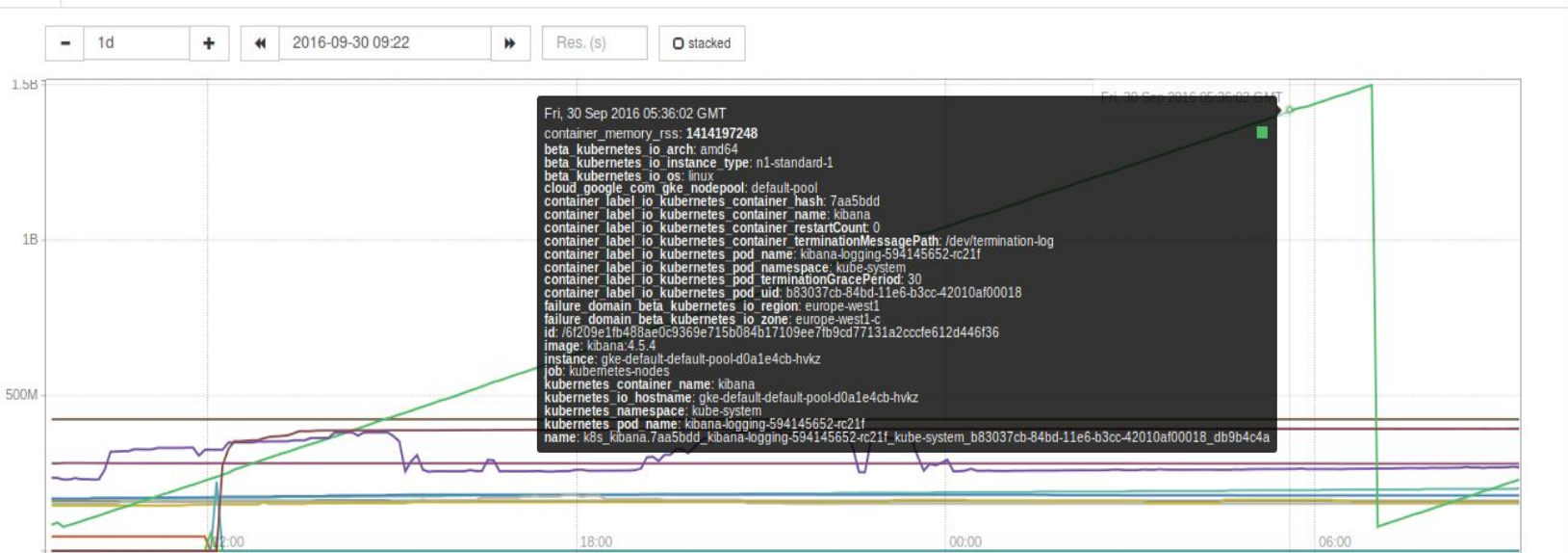
topk(10, container_memory_rss)

Load time: 230ms
Resolution: 345s

Execute

- Insert metric at cursor -

Graph Console



Takeaways

- fully automated CD pipeline
- no **lock-in** or dependency to external services
- failure tolerance towards process **and** node failures
- **scalable!**



PhraseApp

@tobstarr

tobias@phraseapp.com

phraseapp.com