

# Kubernetes

Für DevOps  
und DIY-Entwickler

Slides und Code auf  
<https://github.com/MazderMind/kubernetes-tutorial>

# Inhalt

Erklärungen

Beispiele

Setup

# Inhalt

- Definition und Überblick
- Begriffserklärung
- Beispielsetup – einfach – TCP-Chat in NodeJS
- Beispielsetup – mittel – Python-Anwendung mit Datenbank
- Beispielsetup – komplex – Anwendung mit vielen Dependencies
- Systemsetup – Minikube
- Systemsetup – Google Cloud Engine
- Systemsetup – Debian

# Definition und Überblick

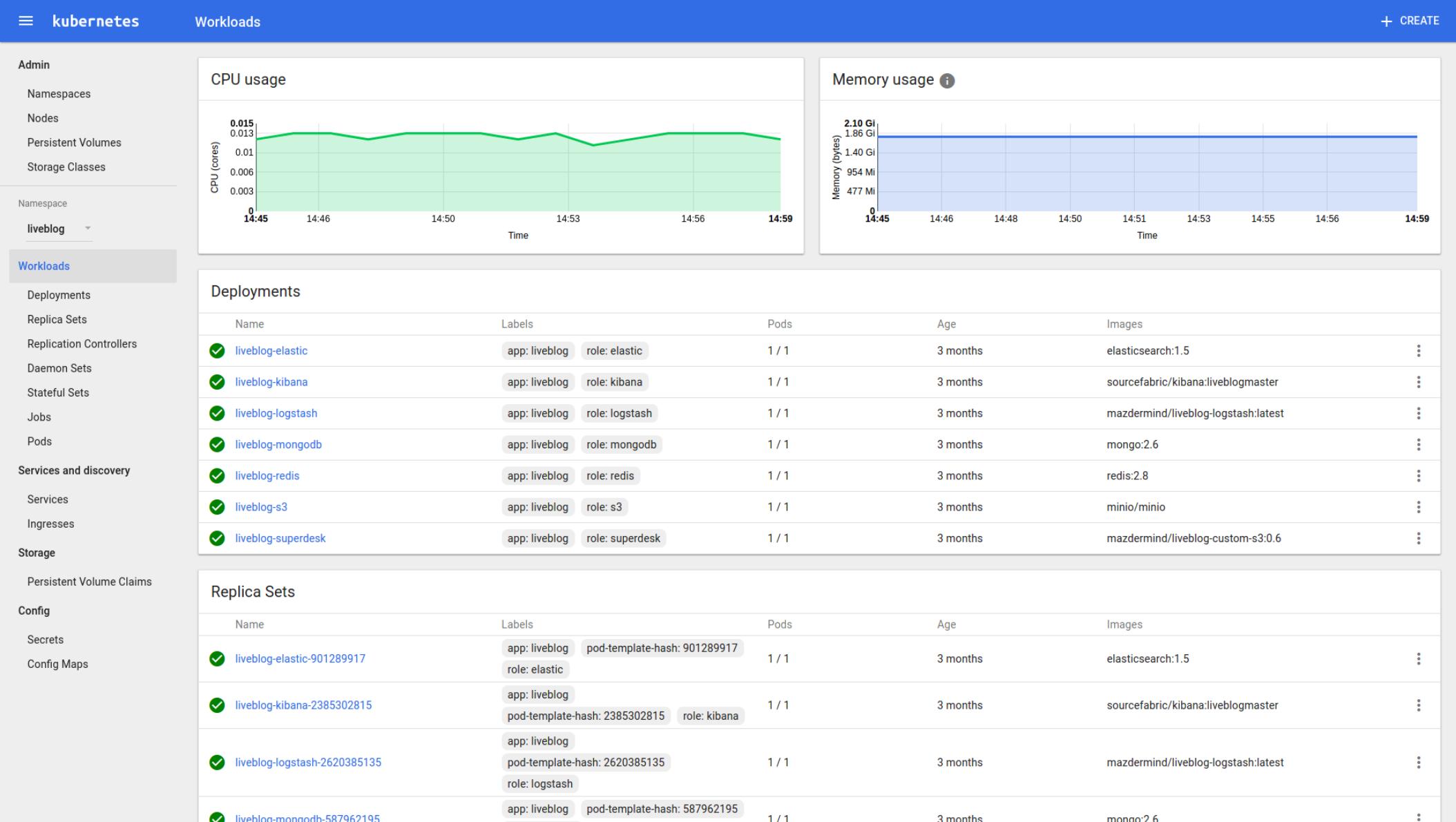
## **Container**

Leichtgewichtige Virtualisierung  
von Anwendungen

# Definition und Überblick

## Kubernetes

System zur Orchestrierung und  
Skalierung von Container-Anwendungen



# Begriffserklärung

## Namespace

Gruppierungselement. Kommunikation zw.  
Namespaces ist möglich aber selten.

z.B. Eine Anwendung  
oder eine Stage einer Anwendung  
(Dev, Test, PreProd, Prod)

# Begriffserklärung

## **Deployment**

deklarative Beschreibung des Zustands  
einer Anwendung

Verwaltet Updates, Scaling,  
Restarts uvm.

# Begriffserklärung

## Pod

1...n Container mit geteiltem Storage,  
gemeinsamen Ressourcen-Limits, IP, Ports

Meist: 1 Container = 1 Pod

# Begriffserklärung

## Pod

Container in einem Pod können via IPC  
(Shared Memory, Pipes, Semaphores, ...)  
miteinander kommunizieren

Beispiel für 2. Container in einem Pod:  
rsyncd-Endpunkt für Backups

# Begriffserklärung

## **Pod**

1 Dienst = 1 Container = 1 Pod

# Begriffserklärung

## **Service**

Sammelt gesunde Pods unter einem gemeinsamen DNS-Namen

Verteilt Anfragen (TCP/UDP LoadBalancer)

# Begriffserklärung

## **Service**

Stellt DNS-Namen und  
Environment-Variablen bereit

Abstrahiert Anbieter und Nutzer innerhalb  
des Clusters

# Begriffserklärung

## **Service**

Veröffentlicht Anwendungen auf  
ISO Layer4 (TCP/UDP IP:Port)

# Beispielsetup – TCP-Chat in NodeJS

- Anwendung
- Dockerfile
- Bauen und Hochladen
- Manifest-Files
- Anwenden
- Zugreifen
- Analysieren
  - kubectl describe
  - kubectl exec
  - kubectl logs
- Aktualisieren



chatServer.js

Raw

```
1 // Load the TCP Library
2 net = require('net');
3
4 // Keep track of the chat clients
5 var clients = [];
6
7 // Start a TCP Server
8 net.createServer(function (socket) {
9
10    // Identify this client
11    socket.name = socket.remoteAddress + ":" + socket.remotePort
12
13    // Put this new client in the list
14    clients.push(socket);
15
16    // Send a nice welcome message and announce
17    socket.write("Welcome " + socket.name + "\n");
18    broadcast(socket.name + " joined the chat\n", socket);
19
20    // Handle incoming messages from clients.
21    socket.on('data', function (data) {
22        broadcast(socket.name + "> " + data, socket);
23    });
24
25    // Remove the client from the list when it leaves
26    socket.on('end', function () {
27        clients.splice(clients.indexOf(socket), 1);
28        broadcast(socket.name + " left the chat.\n");
29    });
30
31    // Send a message to all clients
32    function broadcast(message, sender) {
33        clients.forEach(function (client) {
34            // Don't want to send it to sender
35            if (client === sender) return;
36            client.write(message);
37        });
38        // Log it to the server output too
39        process.stdout.write(message)
40    }
41
42 }).listen(5000);
43
44 // Put a friendly message on the terminal of the server.
45 console.log("Chat server running at port 5000\n");
```

# Dockerfile

```
◀ ▶ Dockerfile x chatServer.js x
1 FROM mhart/alpine-node:8.1
2
3 WORKDIR /app
4 ADD chatServer.js .
5
6 EXPOSE 5000
7 CMD ["node", "chatServer.js"]
8 |
```

```
~/S/k/b/app docker build -t mazdermind/chat-server:v1 .
Sending build context to Docker daemon 4.096 kB
Step 1 : FROM mhart/alpine-node:8.1
--> a5ce2f653420
Step 2 : WORKDIR /app
--> Using cache
--> e216ff369a71
Step 3 : ADD chatServer.js .
--> Using cache
--> 8f23e709ef36
Step 4 : EXPOSE 5000
--> Using cache
--> 5abd752606ce
Step 5 : CMD node chatServer.js
--> Using cache
--> 7a0a5041f3ff
Successfully built 7a0a5041f3ff
~/S/k/b/app docker run --publish 5000:5000 mazdermind/chat-server:v1
Chat server running at port 5000
```

File Edit View Search Terminal Help  
docker /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/app

```
~/S/k/b/app docker build -t mazdermind/chat-server:v1 .
Sending build context to Docker daemon 4.096 kB
Step 1 : FROM mhart/alpine-node:8.1
--> a5ce2f653420
Step 2 : WORKDIR /app
--> Using cache
--> e216ff369a71
Step 3 : ADD chatServer.js .
--> Using cache
--> 8f23e709ef36
Step 4 : EXPOSE 5000
--> Using cache
--> 5abd752606ce
Step 5 : CMD node chatServer.js
--> Using cache
--> 7a0a5041f3ff
Successfully built 7a0a5041f3ff
~/S/k/b/app docker run --publish 5000:5000 mazdermind/chat-server:v1
Chat server running at port 5000
```

```
::ffff:172.17.0.1:43830 joined the chat
::ffff:172.17.0.1:43830> Hello
::ffff:172.17.0.1:43834 joined the chat
::ffff:172.17.0.1:43834> Anyone here?
```

File Edit View Search Terminal Help  
rlwrap /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/app  
~/S/k/b/app rlwrap telnet 127.0.0.1 5000  
Trying 127.0.0.1...  
Connected to 127.0.0.1.  
Escape character is '^]'.  
Welcome ::ffff:172.17.0.1:43830  
Hello  
::ffff:172.17.0.1:43834 joined the chat  
::ffff:172.17.0.1:43834> Anyone here?

Applications ▾ Places ▾ Terminal ▾ Mi Jul 12, 19:09:19

fish /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/app

File Edit View Search Terminal Help

```
~/s/k/b/app docker push mazdermind/chat-server:v1
The push refers to a repository [docker.io/mazdermind/chat-server]
fed725850a21: Pushed
33e79ccc2645: Pushed
6374b506f290: Mounted from mhart/alpine-node
5bef08742407: Mounted from mhart/alpine-node
v1: digest: sha256:ecbd21417d8aa8bd8503363331de4f315b09c9e5c559d7ea2fbda446dfa3c39 size: 1153
~/s/k/b/app
```

mazdermind/chat-server - Docker Hub - Chromium

mazdermind/chat-server x

Secure | https://hub.docker.com/r/mazdermind/chat-server/tags/

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

Search Explore Help Sign up Sign in

PUBLIC REPOSITORY

## mazdermind/chat-server ☆

Last pushed: a minute ago

Repo Info Tags

Tag Name	Compressed Size	Last Updated
v1	22 MB	a minute ago

# Beispielsetup – TCP-Chat in NodeJS

- Anwendung
- Dockerfile
- Bauen und Hochladen
- Manifest-Files
- Anwenden
- Zugreifen
- Analysieren
  - kubectl describe
  - kubectl exec
  - kubectl logs
- Aktualisieren

File Edit Selection Find View Goto Tools Project Preferences Help

~SM/kubernetes-experiments/beispiel-1-tcp-chat/kubernetes/000-tcpchat-namespace.yaml - Sublime Text

```
000-tcpchat-namespace.yaml x 011-tcpchat-chatserver-deploy.yaml x 012-tcpchat-chatserver-svc.yaml x
```

1 apiVersion: v1  
2 kind: Namespace  
3 metadata:  
4 name: tcpchat  
5

1 apiVersion: extensions/v1beta1  
2 kind: Deployment  
3 metadata:  
4 labels:  
5 app: chatserver  
6 name: chatserver  
7 namespace: tcpchat  
8  
9 spec:  
10 replicas: 1  
11 revisionHistoryLimit: 2  
12  
13 selector:  
14 matchLabels:  
15 app: chatserver  
16  
17 template:  
18 metadata:  
19 labels:  
20 app: chatserver  
21  
22 spec:  
23 containers:  
24 name: chatserver  
25 image: mazdermind/chat-server:v1  
26 ports:  
27 containerPort: 5000  
28 name: chat  
29 protocol: TCP  
30

1 kind: Service  
2 apiVersion: v1  
3 metadata:  
4 name: chat  
5 labels:  
6 app: chat  
7 namespace: tcpchat  
8  
9 spec:  
10 selector:  
11 app: chatserver  
12 type: NodePort  
13  
14 ports:  
15 name: chat  
16 port: 5000  
17 protocol: TCP  
18 nodePort: 35000  
20

Line 5, Column 1      UTF-8      Unix      Spaces: 2      YAML

File Edit View Search Terminal Tabs Help

fish /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/kubernetes

```

~/k/nodeport ...
~/k/kubernetes-experiments cd Beispiel-1-tcp-chat/kubernetes/
~/k/b/kubernetes kubectl apply -f 000-tcpchat-namespace.yaml
namespace "tcpchat" created
~/k/b/kubernetes kubectl apply -f 011-tcpchat-chatserver-deploy.yaml
deployment "chatserver" created
~/k/b/kubernetes kubectl apply -f 012-tcpchat-chatserver-svc.yaml
service "chat" created
~/k/b/kubernetes kubectl describe --namespace=tcpchat deployment chatserver
Name:           chatserver
Namespace:      tcpchat
CreationTimestamp:   Wed, 12 Jul 2017 19:18:46 +0200
Labels:          app=chatserver
Selector:        app=chatserver
Replicas:        1 updated | 1 total | 1 available | 0 unavailable
StrategyType:    RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
Conditions:
  Type     Status  Reason
  ----  -----
  Available  True    MinimumReplicasAvailable
OldReplicaSets: <none>
NewReplicaSet:  chatserver-425380256 (1/1 replicas created)
Events:
  FirstSeen  LastSeen  Count  From               SubObjectPath  Type  Reason  Message
  -----  -----  -----  -----  -----  -----  -----  -----
  51s       51s       1      {deployment-controller }          Normal  ScalingReplicaSet  Scaled up replica set chatserver-425380256 to 1
~/k/b/kubernetes kubectl describe --namespace=tcpchat rs chatserver-425380256
Name:           chatserver-425380256
Namespace:      tcpchat
Image(s):       mazdermind/chat-server:v1
Selector:        app=chatserver,pod-template-hash=425380256
Labels:          app=chatserver
                pod-template-hash=425380256
Replicas:        1 current / 1 desired
Pods Status:    1 Running / 0 Waiting / 0 Succeeded / 0 Failed
No volumes.
Events:
  FirstSeen  LastSeen  Count  From               SubObjectPath  Type  Reason  Message
  -----  -----  -----  -----  -----  -----  -----  -----
  1m        1m        1      {replicaset-controller }          Normal  SuccessfulCreate  Created pod: chatserver-425380256-8011c
~/k/b/kubernetes kubectl describe --namespace=tcpchat pod chatserver-425380256-8011c
Name:           chatserver-425380256-8011c
Namespace:      tcpchat
Node:          minikube/192.168.99.100
Start Time:    Wed, 12 Jul 2017 19:18:46 +0200
Labels:          app=chatserver
                pod-template-hash=425380256
Status:         Running
IP:            172.17.0.6
Controllers:   ReplicaSet/chatserver-425380256
Containers:
  chatserver:
    Container ID:  docker://7857ded2c81dd2665e5aaa9d42da780f87ff0e541a5a486b55822ba177da2108
    Image:         mazdermind/chat-server:v1
    Image ID:     docker://sha256:7a0a5041f3ff90618ebbf11b5ee6216f62104aa8169b74045732967387f65205
    Port:          5000/TCP
    State:        Running

```

File Edit View Terminal Tabs Help

kubectl /home/peter/SM/kubernetes-experiments/multicontainer

```
~/s/k/b/kubernetes kubectl describe --namespace=tcpchat rs chatserver-425380256
Name:          chatserver-425380256
Namespace:     tcpchat
Image(s):      mazdermind/chat-server:v1
Selector:      app=chatserver,pod-template-hash=425380256
Labels:        app=chatserver
               pod-template-hash=425380256
Replicas:     1 current / 1 desired
Pods Status:  1 Running / 0 Waiting / 0 Succeeded / 0 Failed
No volumes.
Events:
  FirstSeen   LastSeen   Count   From           SubObjectPath   Type    Reason           Message
  -----       -----       ---   -----         -----           -----   -----   -----
  1m          1m          1   {replicaset-controller }           Normal  SuccessfulCreate  Created pod: chatserver-425380256-8011c
~/s/k/b/kubernetes kubectl describe --namespace=tcpchat pod chatserver-425380256-8011c
Name:          chatserver-425380256-8011c
Namespace:     tcpchat
Node:          minikube/192.168.99.100
Start Time:   Wed, 12 Jul 2017 19:18:46 +0200
Labels:        app=chatserver
               pod-template-hash=425380256
Status:        Running
IP:            172.17.0.6
Controllers:   ReplicaSet/chatserver-425380256
Containers:
  chatserver:
    Container ID:    docker://7857ded2c81dd2665e5aaa9d42da780f87ff0e541a5a486b55822ba177da2108
    Image:          mazdermind/chat-server:v1
    Image ID:       docker://sha256:7a0a5041f3ff90618ebbf11b5ee6216f62104aa8169b74045732967387f65205
    Port:          5000/TCP
    State:         Running
      Started:    Wed, 12 Jul 2017 19:18:57 +0200
    Ready:         True
    Restart Count: 0
    Volume Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-vqpp5 (ro)
    Environment Variables: <none>
Conditions:
  Type     Status
  Initialized  True
  Ready      True
  PodScheduled  True
Volumes:
  default-token-vqpp5:
    Type:      Secret (a volume populated by a Secret)
    SecretName: default-token-vqpp5
QoS Class:  BestEffort
Tolerations: <none>
Events:
  FirstSeen   LastSeen   Count   From           SubObjectPath   Type    Reason           Message
  -----       -----       ---   -----         -----           -----   -----   -----
  1m          1m          1   {default-scheduler }           Normal  Scheduled        Successfully assigned chatserver-425380256-8011c to minikube
  1m          1m          1   {kubelet minikube}        spec.containers[chatserver]  Normal  Pulling         pulling image "mazdermind/chat-server:v1"
  1m          1m          1   {kubelet minikube}        spec.containers[chatserver]  Normal  Pulled          Successfully pulled image "mazdermind/chat-server:v1"
  1m          1m          1   {kubelet minikube}        spec.containers[chatserver]  Normal  Created         Created container with docker id 7857ded2c81d; Security:[seccomp=unconfined]
  1m          1m          1   {kubelet minikube}        spec.containers[chatserver]  Normal  Started        Started container with docker id 7857ded2c81d
~/s/k/b/kubernetes
```

Kubernetes Dashboard - Chromium

localhost:8001/api/v1/proxy/namespaces/kube-system/services/kubernetes-dashboard/#/workload?namespace=tcpchat

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

**kubernetes** Workloads + CREATE

Admin

Namespaces

Nodes

Persistent Volumes

Namespace  
tcpchat

Workloads

Deployments

Name	Labels	Pods	Age	Images
chatserver	app: chatserver	1 / 1	17 hours	mazdermind/chat-server:v2

Replica Sets

Name	Labels	Pods	Age	Images
chatserver-425380256	app: chatserver pod-template-hash: 425380256	0 / 0	17 hours	mazdermind/chat-server:v1
chatserver-507038113	app: chatserver pod-template-hash: 507038113	1 / 1	16 hours	mazdermind/chat-server:v2

Pods

Name	Status	Restarts	Age
chatserver-507038113-1mtgc	Running	0	16 hours

Services and discovery

Services

Ingresses

Storage

Persistent Volume Claims

Config

Secrets

Config Maps

Detailed description: The screenshot shows the Kubernetes Dashboard interface. On the left, there's a sidebar with navigation links for Admin, Namespaces, Nodes, Persistent Volumes, and various workloads like Deployments, Replica Sets, and Pods. The 'tcpchat' namespace is selected. The main content area has three tabs: Deployments, Replica Sets, and Pods. The Deployments tab shows a single deployment named 'chatserver' with 1 pod running, using the image 'mazdermind/chat-server:v2'. The Replica Sets tab shows two replica sets: 'chatserver-425380256' (0 pods) and 'chatserver-507038113' (1 pod running). The Pods tab shows the single pod 'chatserver-507038113-1mtgc' which is running.

Applications ▾ Places ▾ Terminal ▾ Mi Jul 12, 19:31:20

rlwrap /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/kubernetes

File Edit View Search Terminal Tabs Help

kubectl /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/kubernetes

~\$ k/b/kubernetes rlwrap telnet 192.168.99.100 32000

Trying 192.168.99.100...

Connected to 192.168.99.100.

Escape character is '^]'.

Welcome ::ffff:172.17.0.1:52196

::ffff:172.17.0.1:52196 joined the chat

::ffff:172.17.0.1:52196> yay

yoooo

rlwrap /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/kubernetes

File Edit View Search Terminal Help

~/s/k/b/kubernetes rlwrap telnet 192.168.99.100 32000

Trying 192.168.99.100...

Connected to 192.168.99.100.

Escape character is '^]'.

Welcome ::ffff:172.17.0.1:52196

yay

::ffff:172.17.0.1:52196> yoooo

File Edit View Search Terminal Tabs Help

kubectl /home/peter/SM/kubernetes-experiments/multicontainer x fish /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/kubernetes x rlwrap /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/kubernetes x +

```
-/S/k/b/kubernetes kubectl get pods --namespace=tcpchat -l app=chatserver
NAME          READY   STATUS    RESTARTS   AGE
chatserver-425380256-8011c   1/1     Running   0          17m
-/S/k/b/kubernetes
-/S/k/b/kubernetes
-/S/k/b/kubernetes
-/S/k/b/kubernetes kubectl logs --namespace=tcpchat chatserver-425380256-8011c
Chat server running at port 5000

::ffff:172.17.0.1:52194 joined the chat
::ffff:172.17.0.1:52196 joined the chat
::ffff:172.17.0.1:52196> yay
::ffff:172.17.0.1:52194> yoooo
-/S/k/b/kubernetes
-/S/k/b/kubernetes
-/S/k/b/kubernetes kubectl exec --namespace=tcpchat -it chatserver-425380256-8011c sh
/app # ps aux
PID  USER      TIME  COMMAND
 1 root      0:00 node chatServer.js
 69 root      0:00 sh
 73 root      0:00 ps aux
/app #
/app #
/app # ls -la
total 12
drwxr-xr-x  2 root      root      4096 Jul 12 16:51 .
drwxr-xr-x  1 root      root      4096 Jul 12 17:18 ..
-rw-rw-r--  1 root      root     1278 Jul 12 16:43 chatServer.js
/app #
/app #
/app # ↵
-/S/k/b/kubernetes
```

File Edit View Search Terminal Tabs Help

fish /home/peter/SM/kubernetes-experiments/beispiel-1-tcp-chat/app

```

~/k/b/app docker push mazdermind/chat-server:v2
The push refers to a repository [docker.io/mazdermind/chat-server]
c26ebfa04d87: Pushed
33e79ccc2645: Layer already exists
6374b506f290: Layer already exists
5bef08742407: Layer already exists
v2: digest: sha256:c02c49c960c20cb27336bdd58a541c7b4fd7fac4f945b2e113eed8cb29e487b size: 1153
~/k/b/app clear

~/k/b/app kubectl set image --namespace=tcpchat deployment/chatserver chatserver=mazdermind/chat-server:v2
deployment "chatserver" image updated
~/k/b/app kubectl describe --namespace=tcpchat deployment chatserver
Name:          chatserver
Namespace:     tcpchat
CreationTimestamp:   Wed, 12 Jul 2017 19:18:46 +0200
Labels:        app=chatserver
Selector:      app=chatserver
Replicas:     1 updated | 1 total | 1 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
Conditions:
  Type    Status  Reason
  ----  -----
  Available  True    MinimumReplicasAvailable
OldReplicaSets: chatserver-425380256 (0/0 replicas created)
NewReplicaSet:  chatserver-507038113 (1/1 replicas created)
Events:
  FirstSeen  LastSeen  Count  From                    SubObjectPath  Type    Reason
  -----  -----  -----  -----  -----  -----  -----
  24m       24m       1      {deployment-controller }           Normal  ScalingReplicaSet  Scaled up replica set chatserver-425380256 to 1
  24s       24s       1      {deployment-controller }           Normal  ScalingReplicaSet  Scaled up replica set chatserver-507038113 to 1
  24s       24s       1      {deployment-controller }           Normal  ScalingReplicaSet  Scaled down replica set chatserver-425380256 to 0
~/k/b/app kubectl describe --namespace=tcpchat rs chatserver-507038113
Name:          chatserver-507038113
Namespace:     tcpchat
Image(s):      mazdermind/chat-server:v2
Selector:      app=chatserver,pod-template-hash=507038113
Labels:        app=chatserver
               pod-template-hash=507038113
Replicas:     1 current / 1 desired
Pods Status:  1 Running / 0 Waiting / 0 Succeeded / 0 Failed
No volumes.
Events:
  FirstSeen  LastSeen  Count  From                    SubObjectPath  Type    Reason
  -----  -----  -----  -----  -----  -----  -----
  1m        1m        1      {replicaset-controller }           Normal  SuccessfulCreate  Created pod: chatserver-507038113-1mtgc
~/k/b/app kubectl describe --namespace=tcpchat pod chatserver-507038113-1mtgc
Name:          chatserver-507038113-1mtgc
Namespace:     tcpchat
Containers:
  chatserver:
    Container ID:  docker://cc0f3e1a5bb0035b63bcda2f08c86c3f488a526e9d6d1511f288cf5b6408bb0c
    Image:         mazdermind/chat-server:v2
    Image ID:      docker://sha256:21eee9766dd4648b2ec5506956f50d2f2ac9d2e1315a5599bbe60eb86d168ac3
    Port:          5000/TCP
    State:         Running
    Started:      Wed 12 Jul 2017 19:42:53 +0200

```

# Begriffserklärung

## Ingress

ISO Layer 7 (HTTP) Proxy Spezifikation

Wird von einem Ingress-Proxy Interpretiert, welcher bestimmte Domains, oder Pfade via HTTP an Container weiterreicht.

# Begriffserklärung

## Ingress-Proxy

nginx oder traefik

traefik kann z.B. auch SSL Offloading mit  
automatischem *Let's Encrypt*  
Zertifikatsmanagement

Achtung: Production-Ready?

# Begriffserklärung

**Persistent Volume / -Claim**

PVs beschreiben die vorhandenen Ressourcen

PVCs beschreiben die Anforderungen der  
Anwendung

# Begriffserklärung

## **Persistent Volume / -Claim**

Passt kein PV auf einen PVC, kann dynamisch eins erstellt werden

Support für Storage-Klassen (SSD, Disk, Mit/Ohne Backup etc.)

# Beispielsetup #2

- Python-Anwendung
  - Loggt Anfragen
- PostgreSQL-Datenbank
  - PostgreSQL-Service
- Persistent Storage
- Ingress

File Edit View Search Terminal Help docker /home/peter/SM/kubernetes-experiments/beispiel-2-db-anwendung/app

```
~/S/k/b/app (master) $ docker run -e POSTGRES_USER=req -e POSTGRES_PASSWORD=req -e POSTGRES_DB=req postgres:9.6
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /var/lib/postgresql/data ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

  pg_ctl -D /var/lib/postgresql/data -l logfile start

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.
waiting for server to start....LOG:  database system was shut down at 2017-07-12 19:44:49 UTC
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
done
server started
CREATE DATABASE

CREATE ROLE

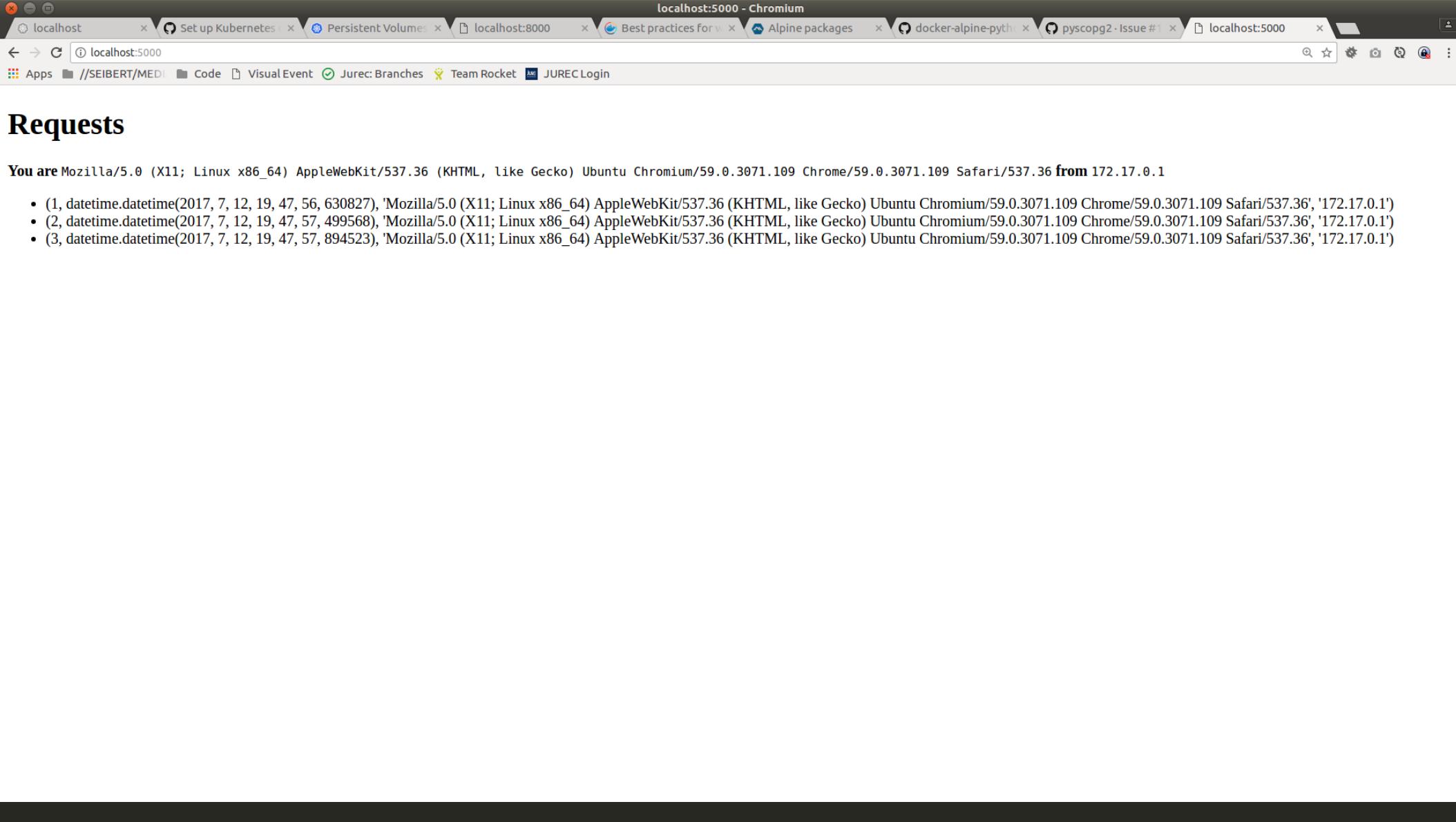
/usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
```

File Edit View Search Terminal Tabs Help

kubectl /home/peter docker /home/peter/SM/kubernetes-experiments/beispiel-2-db-anwendung/app fish /home/peter/SM/kubernetes-experiments/beispiel-2-db-anwendung/app

```
~/S/k/b/app (master) 2 docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
f9420d2df0d5        postgres:9.6      "docker-entrypoint.sh"   2 minutes ago     Up 2 minutes      5432/tcp
~/S/k/b/app (master) 2 docker inspect --format '{{ .NetworkSettings.IPAddress }}' evil_leavitt
172.17.0.2
~/S/k/b/app (master) 2 docker run -e DB_NAME=req -e DB_USER=req -e DB_PASS=req -e DB_HOST=172.17.0.2 -p 5000:5000 -it mazdermind/request-logger:v1
created table if not existed
[NOTICE: relation "reqs" already exists, skipping\n']
```

```
Dockerfile
FROM frolvlad/alpine-python3:latest
WORKDIR /app
RUN apk add --no-cache \
    py3-psycopg2
ADD server.py .
EXPOSE 5000
CMD ["python3", "server.py"]
```



## Requests

You are Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/59.0.3071.109 Chrome/59.0.3071.109 Safari/537.36 from 172.17.0.1

- (1, datetime.datetime(2017, 7, 12, 19, 47, 56, 630827), 'Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/59.0.3071.109 Chrome/59.0.3071.109 Safari/537.36', '172.17.0.1')
- (2, datetime.datetime(2017, 7, 12, 19, 47, 57, 499568), 'Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/59.0.3071.109 Chrome/59.0.3071.109 Safari/537.36', '172.17.0.1')
- (3, datetime.datetime(2017, 7, 12, 19, 47, 57, 894523), 'Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/59.0.3071.109 Chrome/59.0.3071.109 Safari/537.36', '172.17.0.1')

File Edit Selection Find View Goto Tools Project Preferences Help

~SM/kubernetes-experiments/beispiel-2-db-anwendung/kubernetes/001-logger-secrets.yaml - Sublime Text

```
001-logger-secrets.yaml * 010-logger-postgres-pvc.yaml * 011-logger-postgres-deploy.yaml *
```

```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: logger
5   namespace: logger
6 type: Opaque
7 data:
8   db-pass: cmVx # "req" in base64
9   db-user: cmVx # "req" in base64
10  db-name: cmVx # "req" in base64
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
```

!! Reclaim Policy !!

# Beispielsetup #2

- Python-Anwendung
  - Loggt Anfragen
- PostgreSQL-Datenbank
  - PostgreSQL-Service
- Persistent Storage
- Ingress

# Beispielsetup #2

- PostgreSQL-Service & Anwendung
  - Testen mit port-forward

```
File Edit View Search Terminal Tabs Help
~ /S/k/b/kubernetes (master) 2 kubectl port-forward -h
Forward one or more local ports to a pod.

Examples:
# Listen on ports 5000 and 6000 locally, forwarding data to/from ports 5000 and 6000 in the pod
kubectl port-forward mypod 5000 6000

# Listen on port 8888 locally, forwarding to 5000 in the pod
kubectl port-forward mypod 8888:5000

# Listen on a random port locally, forwarding to 5000 in the pod
kubectl port-forward mypod :5000

# Listen on a random port locally, forwarding to 5000 in the pod
kubectl port-forward mypod 0:5000

Options:
-p, --pod='': Pod name

Usage:
kubectl port-forward POD [LOCAL_PORT:]REMOTE_PORT [...[LOCAL_PORT_N:]REMOTE_PORT_N] [options]

Use "kubectl options" for a list of global command-line options (applies to all commands).
~/S/k/b/kubernetes (master) 2 kubectl port-forward --namespace=logger postgres-3535813833-7rx15 32123:5432 &
~/S/k/b/kubernetes (master) 2 Forwarding from 127.0.0.1:32123 -> 5432
Forwarding from [::1]:32123 -> 5432

~/S/k/b/kubernetes (master) 2 psql --host=127.0.0.1 --port=32123 -U req
Handling connection for 32123
psql (9.5.7, server 9.6.3)
WARNING: psql major version 9.5, server major version 9.6.
         Some psql features might not work.
Type "help" for help.

req=# \d
No relations found.
req=#

```

# Beispielsetup #2

```
 10  replicas: 1
11  revisionHistoryLimit: 2
12
13  selector:
14    matchLabels:
15      app: app
16
17  template:
18    metadata:
19      labels:
20        app: app
21
22    spec:
23      containers:
24        - name: app
25          image: mazdermind/request-logger:v1
26
27      env:
28        - name: DB_HOST
29          value: postgres
30
31        - name: DB_NAME
32          valueFrom:
33            secretKeyRef:
34              name: logger
35              key: db-name
36
37        - name: DB_USER
38          valueFrom:
39            secretKeyRef:
40              name: logger
41              key: db-user
42
43        - name: DB_PASS
44          valueFrom:
45            secretKeyRef:
46              name: logger
47              key: db-pass
48
49      ports:
50        - containerPort: 5000
51          name: web
52          protocol: TCP
```

```
File Edit View Search Terminal Tabs Help
File Edit View Search Terminal Tabs Help
~/S/k/b/kubernetes (master) ~ kubectl port-forward --namespace=logger app-3226751085-65nnv 32123:5000
&
~/S/k/b/kubernetes (master) ~ Forwarding from 127.0.0.1:32123 -> 5000
Forwarding from [::1]:32123 -> 5000

~/S/k/b/kubernetes (master) ~ curl -vvv http://127.0.0.1:32123/
* Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 32123 (#0)
> GET / HTTP/1.1
> Host: 127.0.0.1:32123
> User-Agent: curl/7.47.0
> Accept: */*
> Referer:
>
Handling connection for 32123
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: BaseHTTP/0.6 Python/3.6.1
< Date: Thu, 13 Jul 2017 11:46:49 GMT
< Content-type: text/html
<
<!DOCTYPE html><html><body>
  <h1>Requests</h1>
  <p>
    <strong>You are</strong> <tt>curl/7.47.0
    </tt> <strong>from</strong> <tt>127.0.0.1</tt>
  </p>
  <ul><li>(1, datetime.datetime(2017, 7, 13, 11, 4
  1, 36, 54974), 'curl/7.47.0', '127.0.0.1')</li><li>(2, datetime.datetime
  (2017, 7, 13, 11, 41, 38, 605944), 'curl/7.47.0', '127.0.0.1')</li>(
  3, datetime.datetime(2017, 7, 13, 11, 41, 44, 39725), 'curl/7.47.0', '12
  7.0.0.1')</li><li>(4, datetime.datetime(2017, 7, 13, 11, 41, 48, 110167)
  , 'curl/7.47.0', '127.0.0.1')</li><li>(5, datetime.datetime(2017, 7, 13,
  11, 41, 52, 47039), 'curl/7.47.0', '127.0.0.1')</li><li>(6, datetime.da
  tetime(2017, 7, 13, 11, 42, 11, 22505), 'Mozilla/5.0 (X11; Linux x86_64)
  AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/59.0.3071.109 Ch
  rome/59.0.3071.109 Safari/537.36', '127.0.0.1')</li><li>(7, datetime.dat
  etime(2017, 7, 13, 11, 42, 11, 51794), 'Mozilla/5.0 (X11; Linux x86_64)
```

# Beispielsetup #2

- Python-Anwendung
  - Loggt Anfragen
- PostgreSQL-Datenbank
  - PostgreSQL-Service
- Persistent Storage
- Ingress

File Edit View Search Terminal Tabs Help

fish /home/peter/SM/kubernetes-experiments/beispiel-2-db-anwendung/kubernetes

kubectl /home/peter

~/S/k/b/kubernetes (master) ↵

```
kubectl apply -f https://raw.githubusercontent.com/containous/traefik/master/examples/k8s/traefik.yaml
serviceaccount "traefik-ingress-controller" created
daemonset "traefik-ingress-controller" created
~/S/k/b/kubernetes (master) ↵
kubectl apply -f 023-logger-app-ingress.yaml
ingress "app-web" created
~/S/k/b/kubernetes (master) ↵ echo '192.168.99.100 logger.mazdermind.de' | sudo tee -a /etc/hosts
[sudo] password for peter:
192.168.99.100 logger.mazdermind.de
~/S/k/b/kubernetes (master) ↵ curl -vvv http://logger.mazdermind.de
* Rebuilt URL to: http://logger.mazdermind.de/
*   Trying 192.168.99.100...
* Connected to logger.mazdermind.de (192.168.99.100) port 80 (#0)
> GET / HTTP/1.1
> Host: logger.mazdermind.de
> User-Agent: curl/7.47.0
> Accept: */*
> Referer:
>
< HTTP/1.1 200 OK
< Content-Type: text/html
< Date: Thu, 13 Jul 2017 11:54:52 GMT
< Server: BaseHTTP/0.6 Python/3.6.1
< Content-Length: 1450
<
<!DOCTYPE html><html><body>
    <h1>Requests</h1>
    <p>
        <strong>You are</strong> <tt>curl/7.47.0</tt> <strong>from</strong> <tt>192.168.99.1</tt>
    </p>
    <ul><li>(1, datetime.datetime(2017, 7, 13, 11, 41, 36, 54974), 'curl/7.47.0', '127.0.0.1')</li><li>(2, datetime.datetime(2017, 7, 13, 11, 41, 38, 605944), 'curl/7.47.0', '127.0.0.1')</li><li>(3, datetime.datetime(2017, 7, 13, 11, 41, 44, 39725), 'curl/7.47.0', '127.0.0.1')</li><li>(4, datetime.datetime(2017, 7, 13, 11, 41, 48, 110167), 'curl/7.47.0', '127.0.0.1')</li><li>(5, datetime.datetime(2017, 7, 13, 11, 41, 52, 47039), 'curl/7.47.0', '127.0.0.1')</li><li>(6, datetime.datetime(2017, 7, 13, 11, 42, 11, 22505), 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/59.0.3071.109 Chrome/59.0.3071.109 Safari/537.36', '127.0.0.1')</li><li>(7, datetime.datetime(2017, 7, 13, 11, 42, 11, 51794), 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/59.0.3071.109 Chrome/59.0.3071.109 Safari/537.36', '127.0.0.1')</li><li>(8, datetime.datetime(2017, 7, 13, 11, 46, 23, 621921), 'curl/7.47.0', '127.0.0.1')</li><li>(9, datetim
```

# Beispielsetup #2

- Python-Anwendung
  - Loggt Anfragen
- PostgreSQL-Datenbank
  - PostgreSQL-Service
- Persistent Storage
- Ingress
- Next Steps:
  - Startup- und -Teardown-Sequence
  - Automatische SSL-Zertifikate mit *Let's Encrypt*
  - *Skalierung mehrerer App-Instanzen*
  - *Reclaim-Policy der Persistent-Volumes*
  - *Yaml-Dateien als Templates*
    - <https://github.com/kubernetes/helm>

Kubernetes Dashboard - Chromium

127.0.0.1:32123    localhost:8001/api/v1/namespaces/kube-system/services/kubernetes-dashboard/proxy/#!/workload?namespace=liveblog

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

## kubernetes Workloads

**Admin**

- Namespaces
- Nodes
- Persistent Volumes
- Storage Classes

**Namespace**

- liveblog

**Workloads**

- Deployments
- Replica Sets
- Replication Controllers
- Daemon Sets
- Stateful Sets
- Jobs
- Pods
- Services and discovery
- Services
- Ingresses
- Storage
- Persistent Volume Claims

**CPU usage**

**Memory usage**

### Deployments

Name	Labels	Pods	Age	Images
liveblog-elastic	app: liveblog role: elastic	1 / 1	4 months	elasticsearch:1.5
liveblog-kibana	app: liveblog role: kibana	1 / 1	4 months	sourcefabric/kibana:liveblogmaster
liveblog-logstash	app: liveblog role: logstash	1 / 1	4 months	mazdermind/liveblog-logstash:latest
liveblog-mongodb	app: liveblog role: mongodb	1 / 1	4 months	mongo:2.6
liveblog-redis	app: liveblog role: redis	1 / 1	4 months	redis:2.8
liveblog-s3	app: liveblog role: s3	1 / 1	4 months	minio/minio
liveblog-superdesk	app: liveblog role: superdesk	1 / 1	4 months	mazdermind/liveblog-custom-s3:0.6

### Replica Sets

Name	Labels	Pods	Age	Images
liveblog-elastic-901289917	app: liveblog pod-template-hash: 901289917 role: elastic	1 / 1	4 months	elasticsearch:1.5
liveblog-kibana-2385302815	app: liveblog pod-template-hash: 2385302815 role: kibana	1 / 1	4 months	sourcefabric/kibana:liveblogmaster

# Beispielsetups

- Beispiel 1 & 2:  
<https://github.com/MaZderMind/kubernetes-techtalk>
- Beispiel 3:  
<https://github.com/MaZderMind/liveblog-kubernetes>

# Inhalt

- Definition und Überblick
- Begriffserklärung
- Beispielsetup – einfach – TCP-Chat in NodeJS
- Beispielsetup – mittel – Python-Anwendung mit Datenbank
- Beispielsetup – komplex – Anwendung mit vielen Dependencies
- Systemsetup – Minikube
- Systemsetup – Google Cloud Engine
- Systemsetup – Debian

# Systemsetup – Minikube

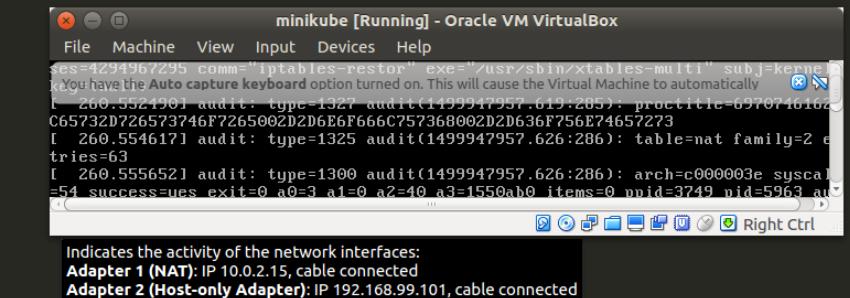
- Lokales Cluster mit einer Node
- Linux, MacOS und eventuell Windows ;)
- u.a. VirtualBox als Hypervisor
- VM hat Host-Only IP als Node-IP
- Optimal zum Entwickeln und Ausprobieren oder für Integration-Tests
  - Alle Beispiele und Screenshots wurden auf Minikube unter Ubuntu gemacht

Applications ▾ Places ▾ VirtualBox ▾

Do Jul 13, 14:12:40

fish /home/peter/SM/kubernetes-experiments/beispiel

```
~/S/k/b/kubernetes (master) 2 minikube start
Starting local Kubernetes cluster...
Starting VM...
SSH-ing files into VM...
Setting up certs...
Starting cluster components...
Connecting to cluster...
Setting up kubeconfig...
Kubectl is now configured to use the cluster.
~/S/k/b/kubernetes (master) 2 kubectl get nodes
NAME STATUS AGE
minikube Ready 18s
~/S/k/b/kubernetes (master) 2 kubectl get pods
No resources found.
~/S/k/b/kubernetes (master) 2 kubectl get pods --namespace=kube-system
NAME READY STATUS RESTARTS AGE
kube-addon-manager-minikube 1/1 Running 0 24s
kube-dns-v20-2mk9m 0/3 ContainerCreating 0 15s
kubernetes-dashboard-mm4jn 1/1 Running 0 15s
~/S/k/b/kubernetes (master) 2
```



# Systemsetup – GCE

- (Offiziell) nur Geschäftliche Nutzung
    - Umsatzsteuer...
  - „One“-Click
  - 12 Monate / 300\$ Spielgeld (Testzeitraum)
  - Größe und Konfiguration der Nodes lässt sich jederzeit Anpassen
  - Beliebig viele Nodes und Storage (modulo \$\$\$)
  - Google...
- 
- Optimal für Produktivumgebungen, wenn Google und \$\$\$ kein Problem sind.

Google Cloud Platform - Chromium

Secure | https://console.developers.google.com/projectcreate?previousPage=%2Fcloud-resource-manager%3FpreviousPage%3D%252F&defaultProjectName&organizationId=0

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

Gefallen Ihnen unsere APIs? Dann lernen Sie unsere Infrastruktur kennen. Registrieren Sie sich und testen Sie die Google Cloud Platform 12 Monate lang, inklusive 300-\$-Gutschrift. [Mehr erfahren](#)

SCHLIESSEN JETZT KOSTENLOS TESTEN

Google APIs

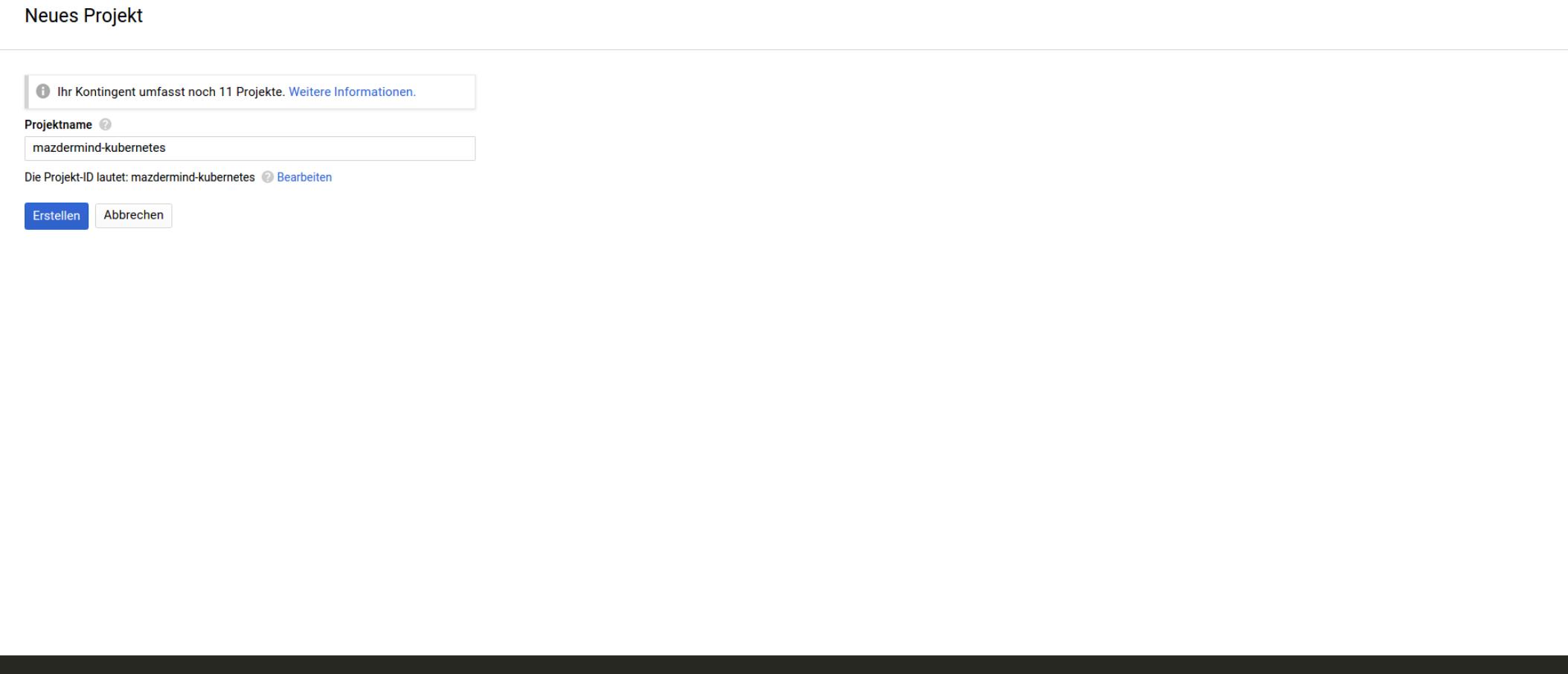
Neues Projekt

Ihr Kontingent umfasst noch 11 Projekte. [Weitere Informationen](#).

Projektname ?  
mazdermind-kubernetes

Die Projekt-ID lautet: mazdermind-kubernetes [Bearbeiten](#)

Erstellen Abbrechen



Google Cloud Platform - Chromium

Secure | https://console.developers.google.com/home/dashboard?project=mazdermind-kubernetes

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

Google APIs mazdermind-kubernetes

ANPASSEN

**Compute Engine**  
Sie sind nicht berechtigt, diese Informationen aufzurufen

**Trace**  
Keine Trace-Daten für die letzten 7 Tage  
→ Jetzt mit Stackdriver Trace starten

**Erste Schritte**

- API APIs aktivieren und Anmeldedaten (z. B. Schlüssel) abrufen
- Vorkonfigurierte Lösung bereitstellen → Fügen Sie der laufenden Anwendung eine dynamische Protokollierung hinzu
- Fehler mithilfe von Fehlerberichten überwachen
- Hello World App bereitstellen
- Cloud Storage-Bucket erstellen
- Cloud SDK installieren

→ Alle Anleitungen

→ Weiter zum Compute Engine-Dashboard

**API APIs**  
Anfragen (Anfrage/s)  
Für dieses Diagramm sind keine Daten verfügbar  
→ Zur API-Übersicht

→ Erfahren Sie, wie Sie Error Reporting einrichten können

**Neuigkeiten**  
Google Cloud Platform now open in London vor 5 Stunden  
Container Engine now runs Kubernetes 1.7 to drive enterprise-ready secure hybrid workloads vor 1 Tag  
Guest post: Loot Crate unboxes Google Container Engine for new Sports Crate venture vor 1 Tag  
→ Alle Neuigkeiten lesen

**Dokumentation**  
Informationen zu Compute Engine  
Informationen zu Cloud Storage  
Informationen zu App Engine

https://console.cloud.google.com/launcher?project=mazdermind-kubernetes

Cloud Launcher - mazdermind-kubernetes - Chromium

Secure | https://console.cloud.google.com/launcher/browse?q=kubernetes&project=mazdermind-kubernetes

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

Registrieren Sie sich jetzt, erhalten Sie ein Guthaben von 300 \$ und testen Sie die Google Cloud Platform 12 Monate lang kostenlos. [Mehr erfahren](#)

SCHLIESSEN JETZT KOSTENLOS TESTEN

Google Cloud Platform mazdermind-kubernetes

Cloud Launcher

Cloud Launcher  X

Launcher > "kubernetes"

Filtern nach 2 Ergebnisse

TYP

Google Cloud Platform (1)

APIs & Dienste (1)

KATEGORIE

Computing (1)

Entwicklerstapel (1)

Entwicklertools (1)

 Container Engine Google One-click Kubernetes clusters, managed by Google Typ Google Cloud Platform

 Codefresh - Container Platform Codefresh Build, test, spin up, and deploy Docker Images to Google (GKE) Typ APIs & Dienste



# Google Cloud Platform

## Google Cloud Platform kostenlos testen

### Kundeninformationen

#### Kontotyp

Geschäftskonto

#### Steuerinformationen

Steuerstatus: Geschäftskonto

#### Name und Adresse

Name des Unternehmens

Peter Körner GbR

Name

Peter Körner

Adresszeile 1

Adresszeile 2



#### Zugriff auf alle Produkte der Cloud Platform

Sie erhalten alles, was Sie für die Erstellung und Ausführung Ihrer Apps, Websites und Dienste benötigen, unter anderem Firebase und die Google Maps API.



#### Guthaben von 300 \$

Wenn Sie sich registrieren, können Sie mit Ihrem Guthaben von 300 \$ die Google Cloud Platform 12 Monate lang nutzen.



#### Keine automatische Gebühr nach Ablauf des Testzeitraums

Wir bitten Sie um Ihre Kreditkarteninformationen, um uns zu vergewissern, dass Sie kein Roboter sind. Ihnen werden keine Kosten in Rechnung gestellt, sofern Sie kein manuelles Upgrade auf ein kostenpflichtiges Konto durchführen.

### Zahlungsoption



#### Monatliche autom. Zahlungen

Sie bezahlen auf monatlicher Basis für diesen Dienst. Ihr Konto wird automatisch belastet, wenn die Zahlung fällig ist.

Dieser Dienst darf nur zu geschäftlichen oder kommerziellen Zwecken genutzt werden. Sie sind dafür verantwortlich, die Umsatzsteuer zu veranlagen und zu melden.

### Zahlungsmethode



Neue Kredit- oder Debitkarte hinzufügen



Bankkonto hinzufügen

**Kostenlose Testversion starten**

Datenschutzerklärung



#### Zugriff auf alle Produkte der Cloud Platform

Sie erhalten alles, was Sie für die Erstellung und Ausführung Ihrer Apps, Websites und Dienste benötigen, unter anderem Firebase und die Google Maps API.



#### Guthaben von 300 \$

Wenn Sie sich registrieren, können Sie mit Ihrem Guthaben von 300 \$ die Google Cloud Platform 12 Monate lang nutzen.

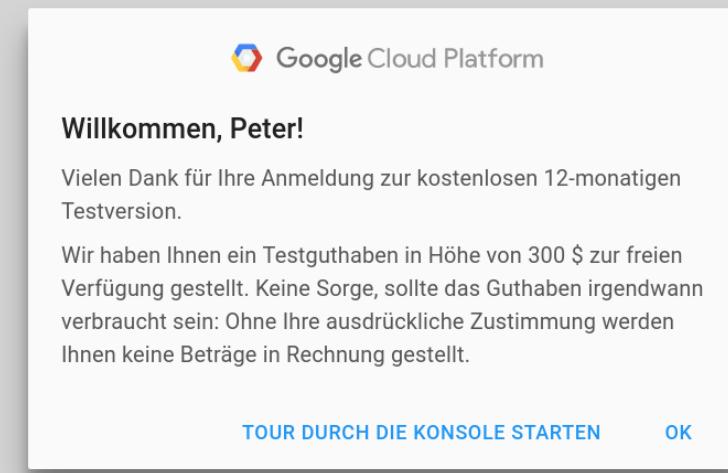


#### Keine automatische Gebühr nach Ablauf des Testzeitraums

Wir bitten Sie um Ihre Kreditkarteninformationen, um uns zu vergewissern, dass Sie kein Roboter sind. Ihnen werden keine Kosten in Rechnung gestellt, sofern Sie kein manuelles Upgrade auf ein kostenpflichtiges Konto durchführen.

## Container Engine

Container Engine wird aktiviert. Dies kann einige Minuten dauern. [Container Engine-Dokumentation](#)



Ihr Guthaben beträgt 263,21 € und Ihr kostenloser Testzeitraum endet in 365 Tagen.

SCHLIESSEN

UPGRADE AUSFÜHREN

≡ Google Cloud Platform mazdermind-kubernetes ▾



Container Engine

## Container Engine Container-Cluster

Container erstellen ein Anwendungspaket, damit die Anwendung problemlos bereitgestellt und in ihrer isolierten Umgebung ausgeführt werden kann. Container werden in Clustern verwaltet, um die VM-Erstellung und -Wartung zu automatisieren. [Mehr erfahren](#)

[Container-Cluster erstellen](#) oder [Schnellstart verwenden](#)

Container Engine - Google Container Engine - Container Engine - mazdermind-kubernetes - Chromium

Secure | https://console.cloud.google.com/kubernetes/add?project=mazdermind-kubernetes

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

Google Cloud Platform mazdermind-kubernetes

Container Engine Container Cluster erstellen

A container cluster is a managed group of uniform VM instances for running Kubernetes. [Mehr erfahren](#)

Name ?  
mazdermind-cluster-1

Beschreibung (optional)  
Testcluster

Zone ?  
europe-west1-c

Clusterversion ?  
1.6.4 (Standard)

Maschinentyp

Kerne  
Basisansicht  
1 vCPU 1 - 8

Speicher  
6 GB 1 - 6,5

Weitere Informationen zur Auswahl des Maschinentyps ↗

Führen Sie für Ihr Konto ein Upgrade aus, um Instanzen mit bis zu 64 Kernen zu erstellen

Knoten-Image ?

← → ⌂ Secure | https://console.cloud.google.com/kubernetes/list?project=mazdermind-kubernetes

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

Google Cloud Platform mazdermind-kubernetes ⌂

Container Engine CLUSTER ERSTELLEN AKTUALISIEREN LÖSCHEN INFOFELD ANZEIGEN

Nach Label oder Namen filtern

Container-Cluster

Name	Zone	Clustergröße	Kerne insgesamt	Speicher insgesamt	Knotenversion	Labels
<input checked="" type="checkbox"/> mazdermind-cluster-1	europe-west1-c	3	3 vCPUs	18,00 GB	1.6.4	<a href="#">Verbinden</a> <a href="#"></a> <a href="#"></a>

Container Engine - mazdermind-kubernetes - Chromium

← → C 🔒 Secure | https://console.cloud.google.com/kubernetes/list?project=mazdermind-kubernetes

Apps //SEIBERT/MEDI Code Visual Event Jurec: Branches Team Rocket JUREC Login

Google Cloud Platform mazdermind-kubernetes

Container Engine

+ CLUSTER ERSTELLEN C AKTUALISIEREN LÖSCHEN

Nach Label oder Namen filtern

Container-Cluster

Name	Zone	Clustergröße	Kerne insgesamt
<input checked="" type="checkbox"/> mazdermind-cluster-1	europe-west1-c	3	3 vCPUs

Mit dem Cluster verbinden

Konfigurieren Sie den Zugriff auf die [kubectl](#)-Befehlszeile, indem Sie folgenden Befehl ausführen:

```
$ gcloud container clusters get-credentials mazdermind-cluster-1 \
--zone europe-west1-c --project mazdermind-kubernetes
```

Starten Sie anschließend einen Proxy, um eine Verbindung zur Kubernetes-Steuerungsebene herzustellen:

```
$ kubectl proxy
```

Öffnen Sie danach die Dashboard-Benutzeroberfläche, indem Sie mit Ihrem Browser hierher navigieren:

<http://localhost:8001/ui>

OK

# Systemsetup – GCE

- gcloud auth application-default login
- gcloud config set project mazdermind-cluster-1
- gcloud container clusters get-credentials \  
mazdermind-cluster-1 \  
--zone europe-west1-c \  
--project mazdermind-kubernetes
- kubectl get nodes
- kubectl proxy



Search

+ CREATE

## ☰ Workloads

## Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

## Namespace

kube-system

## Workloads

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

## Discovery and Load Balancing

Ingresses

Services

## Config and Storage

Config Maps

Persistent Volume Claims

## CPU usage



## Memory usage ⓘ



## Daemon Sets

Name	Labels	Pods	Age	Images
fluentd-gcp-v2.0	addonmanager.kubernetes.io/mode: Reconcile k8s-app: fluentd-gcp kubernetes.io/cluster-service: true version: v2.0	3 / 3	36 minutes	gcr.io/google-containers/fluentd-gcp:2.0.7 gcr.io/google-containers/prometheus-to-sd:v0.1.0

## Deployments

Name	Labels	Pods	Age	Images
event-exporter-v0.1.4	addonmanager.kubernetes.io/mode: Reconcile k8s-app: event-exporter kubernetes.io/cluster-service: true	1 / 1	21 minutes	gcr.io/google-containers/event-exporter:v0.1.0-r2 gcr.io/google-containers/prometheus-to-sd:v0.1.2-r2
heapster-v1.4.0	addonmanager.kubernetes.io/mode: Reconcile k8s-app: heapster kubernetes.io/cluster-service: true	1 / 1	21 minutes	eu.gcr.io/google_containers/heapster-amd64:v1.4.0 eu.gcr.io/google_containers/addon-resizer:1.7

google-clo...tar.gz

Show all

# Systemsetup – Ubuntu/Debian

- Freie Wahl des Betriebssystems?
  - CoreOS

# Systemsetup – Ubuntu/Debian

- <https://kubernetes.io/docs/setup/independent/create-cluster-kubeadm/>
- Versionen von docker, kubectl & kubeadm beachten!
- Evtl. cgroups-memory in grub aktivieren:
  - <https://serverfault.com/questions/790318/cannot-enable-cgroup-enable-memory-swapaccount-1-on-gce-debian-jessie-instance>
- (Noch) Nicht für Produktion – Beta Phase
  - (außer du bist mutig ;)
- Optimal für den Selbermacher oder wenn vorhandene Server genutzt werden müssen / Google / AWS nicht in Frage kommt

# Systemsetup – Ubuntu/Debian

- Netzwerk: Flannel
  - `kubectl create -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel-rbac.yml`
  - `kubectl create -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml`
- Dashboard
  - `kubectl create -f https://git.io/kube-dashboard`
- UFW
  - `DEFAULT_FORWARD_POLICY="ACCEPT"`

# Systemsetup – Ubuntu/Debian

- Storage NFS vs. HostPath
- Dynamic HostPath Provisioning
  - `kubectl create -f https://raw.githubusercontent.com/MaZderMind/hostpath-provisioner/master/manifests/rbac.yaml`
  - `kubectl create -f https://raw.githubusercontent.com/MaZderMind/hostpath-provisioner/master/manifests/deployment.yaml`
  - `kubectl create -f https://raw.githubusercontent.com/MaZderMind/hostpath-provisioner/master/manifests/storageclass.yaml`

# Systemsetup – Ubuntu/Debian

- Traefik hinter Apache (Fließender Übergang)
  - <https://gist.github.com/MaZderMind/b95da0071ba6d4dcf7babfeb477a292>
- `ufw` verhindert Forwards
  - Zum Debuggen: `ufw reload` hilft
  - Langzeitlösung: eigene iptables-rules in die rc.local
    - Doppelt gut: Kommentare
    - <https://gist.github.com/MaZderMind/e0395042b5271c5828b9144562d93b52>
- Unattended Upgrade
  - Cronjob mit `helm update`?

# Fragen (und unfrisierte Gedanken)

**Jetzt oder gerne im Anschluss**

Bei Bier, Mate oder Kaffee