

### Nodes

master01	192.16- 8.192.88	Michael
master02	192.16- 8.192.89	Jens
master03	192.16- 8.192.90	Jarek
worker01	192.16- 8.192.91	Kai
worker02	192.16- 8.192.92	Reiner
worker03	192.168.192.93	
worker04	192.168.192.94	
haproxy load balancer	192.168.192.95 <sup>1</sup>	

<sup>1</sup>) Nur innerhalb des Clusters erreichbar.  
kubectl auf den master nodes greift über den haproxy auf die API zu.

Zuständigkeiten wurden am 05.09.2023 in Mattermost festgelegt.

### MetalLB Pools

production	192.168.254.188 – 192.168.254.192
testing	192.168.254.193 – 192.168.254.197

Annotation `metallb.universe.tf/address-pool` um Services vom Typ `LoadBalancer` eine IP aus dem entsprechenden Pool zuweisen zu lassen. `metallb.universe.tf/loadBalancerIPs` um eine konkrete IP-Adresse zuzuweisen. → MetalLB Usage Dokumentation

### Kubernetes Dashboard

Neuen Login-Token generieren

```
kubect1 -n kubernetes-dashboard  
create token admin-user
```

Dashboard: <https://192.168.254.188/#/login>

### k9s

### kubect1

**Kubernetes Version ausgeben**  
k version --short

**Liste aller Ressourcen inkl. Abkürzungen**  
k api-re sources

**Konfiguration anzeigen**  
k config view [--minify]  
oder mit Highlighting am Mac  
k config view | bat -l yml

**Context anzeigen**  
k config current -c context

**Context setzen**  
k config use-context <context-name>  
k config current -c context

**Temporär einen anderen Context benutzen**  
k --context =<context-name> get nodes

**Netzwerkprobleme debuggen**  
k run -i --tty debug2 --rm \  
--image=alpine -- sh \  
-c "apk add --no-cache \  
curl bind-tools && sh"

### Helm

### kubect1 get Shortnames

	<b>cj</b>	cronjobs
	<b>cm</b>	configmaps
	<b>crd, crds</b>	customresourcedefinitions
	<b>cs</b>	componentstatuses
	<b>csr</b>	certificatesigningrequests
	<b>deploy</b>	deployments
	<b>ds</b>	daemonsets
	<b>ep</b>	endpoints
	<b>ev</b>	events
	<b>hpa</b>	horizontalpodautoscalers
	<b>ing</b>	ingresses
	<b>limits</b>	limitranges
	<b>netpol</b>	networkpolicies
	<b>no</b>	nodes
	<b>ns</b>	namespaces
	<b>pc</b>	priorityclasses
	<b>pdb</b>	poddisruptionbudgets
	<b>po</b>	Pods
	<b>pv</b>	persistentvolumes
	<b>pvc</b>	persistentvolumeclaims
	<b>quota</b>	resourcequotas
	<b>rc</b>	replicationcontrollers
	<b>rs</b>	replicasets
	<b>sa</b>	serviceaccounts
	<b>sc</b>	storageclasses
	<b>sts</b>	statefulsets
	<b>svc</b>	services

Eine Liste aller im Cluster verfügbaren Resource inkl. Shortnames:  
kubect1 api-re sources [-o wide]

<code>k9s info</code>	K9s runtime info
<code>k9s -n ns</code>	Start in ns namespace
<code>k9s --context ctx</code>	Start in ctx context
<code>k9s --readonly</code>	Read only mode
<code>?</code>	Show shortcuts help

### Repos anzeigen

```
helm repo ls
```

### Repo hinzufügen

```
helm repo add <re po- nam e> <ur l>
```

### Charts in einem Repo auflisten

```
helm search repo traefik
```

### Ältere Versionen eines Charts auflisten

```
helm search repo traefi k/t raefik \
  --versions
```

### Installierte Charts auflisten

```
helm ls [--all -na mes paces]
```

### Chart installieren

```
helm install <re lea se> \
  [--nam espace <na mes pac e>] \
  [--set key1=v all ,ke y2= val2] \
  [--values <ya ml- fil e/u rl>] \
  [--dry-run --debug] \
  [--dep end enc y-u pdate ] \
  <re po> /<c har t>
```

### Chart upgraden inkl. Rollback

```
helm upgrade <re lea se> <ch art> \
  --atomic
```

### Chart deinstallieren

```
helm uninstall <re lea se>
```

### Chart Innereien anzeigen

```
helm show all <re po> /<c har t>
```

### Chart Standard-Werte anzeigen

```
helm show values <re po> /<c har t>
```

Mehr Kommandos: Offizielles Cheatsheet



By DASKAJA  
cheatography.com/daskaja/

Not published yet.  
Last updated 8th September, 2025.  
Page 1 of 2.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>