

Know-How

kubectl Plugins

<https://krew.sigs.k8s.io/plugins/>

Plugin "neat" - Beispiele

```
kubectl get deployment <deploymentname> -n <namespace> -o yaml | kubectl  
neat -f - > deployment.yaml
```

Private Registry/Repository

Abfrage Registry Katalog

```
curl -k https://registry.tuxnet.lan/v2/_catalog
```

Oder mit dem schöneren Tool **reg**

```
reg ls -k registry.tuxnet.lan
```

Docker Images in die private registry kopieren

```
skopeo copy docker://k8s.gcr.io/metrics-server/metrics-server:v0.5.0  
docker://registry.tuxnet.lan/metrics-server/metrics-server:v0.5.0
```

Befüllen der privaten Registry

```
oci-local -u -r registry.tuxnet.lan
```

Löschen von Registry Einträgen/Images mittels crane

```
crane delete "registry.tuxnet.lan/iot/fhem@$(crane digest  
registry.tuxnet.lan/iot/fhem:3.3.0)"
```

Konfiguration der CRE (containerd) für die private Registrie

/etc/containerd/config.toml

```
[plugins."io.containerd.grpc.v1.cri".registry]
  [plugins."io.containerd.grpc.v1.cri".registry.mirrors]
[plugins."io.containerd.grpc.v1.cri".registry.mirrors."registry.tuxnet.lan"]
  endpoint = ["https://registry.tuxnet.lan:32568"]
```

Kubernetes Master-Node welche Ordner/Files auf weitere Master-Nodes kopieren

- Vorbereitungen zum Joinen der weiteren control Nodes
- Sicherung der wesentlichen Dateien, insbesondere der Kubernetes Cluster PKI

```
sudo tar cf kubestrap.tar /etc/kubernetes/admin.conf
/etc/kubernetes/audit.yaml /etc/kubernetes/manifests/lb.yaml
/etc/kubernetes/pki/ca.* /etc/kubernetes/pki/front-proxy-ca.*
/etc/kubernetes/pki/sa* /etc/kubernetes/pki/etcd/ca.*
```

Entpacken dann entsprechend

```
sudo tar xf kubestrap.tar -C /
```

Ausgabe des Token Hash's

Gibt ein kubeadm Befehl zurück zum joinen einer worker Node

```
kubeadm token create --dry-run --print-join-command
```

Erstellt ein kubeadm join Kommando zum hinzufügen einer weiteren control-plan Node.

```
echo $(kubeadm token create --print-join-command) --control-plane --
certificate-key $(kubeadm init phase upload-certs --upload-certs | grep -vw
-e certificate -e Namespace)
```

Zertifikat-Anfragen von Nodes genehmigen

Auflisten aller Anfragen

```
kubectl get csr
```

Anfragen genehmigen

```
kubectl certificate approve <cert-name>
```

Ändern des clusterweiten Konfiguration

Anzeige der aktuellen Cluster Konfig

```
kubectl get cm -n kube-system kubeadm-config -oyaml
```

Ändern der Konfiguration

```
kubectl edit cm -n kube-system kubeadm-config
```

Updating resources

```
kubectl set image deployment/frontend www=image:v2          # Rolling
update "www" containers of "frontend" deployment, updating the image
kubectl rollout history deployment/frontend                  # Check the
history of deployments including the revision
kubectl rollout undo deployment/frontend                      # Rollback
to the previous deployment
kubectl rollout undo deployment/frontend --to-revision=2     # Rollback
to a specific revision
kubectl rollout status -w deployment/frontend               # Watch
rolling update status of "frontend" deployment until completion
kubectl rollout restart deployment/frontend                  # Rolling
restart of the "frontend" deployment
```

new command or entry point for a container

```
apiVersion: v1
```

```
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: golang
  name: golang
  namespace: testing-chantal
spec:
  containers:
  - name: golang
    image: golang:bookworm
    tty: true
    stdin: true
    command: ["/bin/sh"]
    resources: {}
    volumeMounts:
    - name: bin
      mountPath: /go/psp-mig
  imagePullSecrets:
  - name: regcred
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  volumes:
  - name: bin
    persistentVolumeClaim:
      claimName: pvc-golang
status: {}
```

Kubelet image delete

Herausfinden ab welche Werte Images vom Kubelet gelöscht werden

```
k get --raw /api/v1/nodes/dbaas-node2/proxy/configz | yq -P

...
imageMinimumGCAge: 2m0s
imageGCHighThresholdPercent: 85
imageGCLowThresholdPercent: 80
...
```

Visualize your Kubernetes workloads

k9s

<https://k9scli.io/>

<https://github.com/derailed/k9s/releases>

Octant

<https://octant.dev/>

<https://github.com/vmware-archive/octant/releases>

Cronjob disable

Um einen Kubernetes cronjob zu deaktivieren

```
kubectl patch cronjobs job-name -p '{"spec" : {"suspend" : true }}'
```

kubectl debug

Mittels kubectl die Kubernetes Nodes betreten

```
k debug no/<nodename> -it --image cgr.dev/chainguard/bash -- chroot /host
```

Mittels kubectl Ephemeral Container in einem Pod erstellen

```
kubectl debug pod/<podname> -it --image=curlimages/curl:latest -- sh
```

get pods im Status ungleich running

Wenn man sich alle Pods im Cluster anschauen möchte welche den Status ungleich running haben kann man das mittels field-selector machen.

```
k get pods -A --field-selector=status.phase!=Running
```

delete resources hang on state terminating

```
kubectl get namespace "stucked-namespaces" -o json \
| tr -d "\n" | sed "s/\"finalizers\": \[[^\]]+\]/\"finalizers\": []/" \
| kubectl replace --raw /api/v1/namespaces/stucked-namespaces/finalize -f -
```

Liste Anzahl Pods per Node

```
kubectl get pods --all-namespaces -o=json | jq -r '.items[] |  
select(.status.phase == "Running") | .spec.nodeName' | sort | uniq -c | awk  
'{print $2, $1}'
```

From:

<https://www.cooltux.net/> - TuxNet DokuWiki

Permanent link:

<https://www.cooltux.net/doku.php?id=it-wiki:kubernetes:know-how>

Last update: **2025/09/07 14:40**

