

Kubeadm CA ... ~~10~~ 11 let poté

Tomáš Kukrál

About me

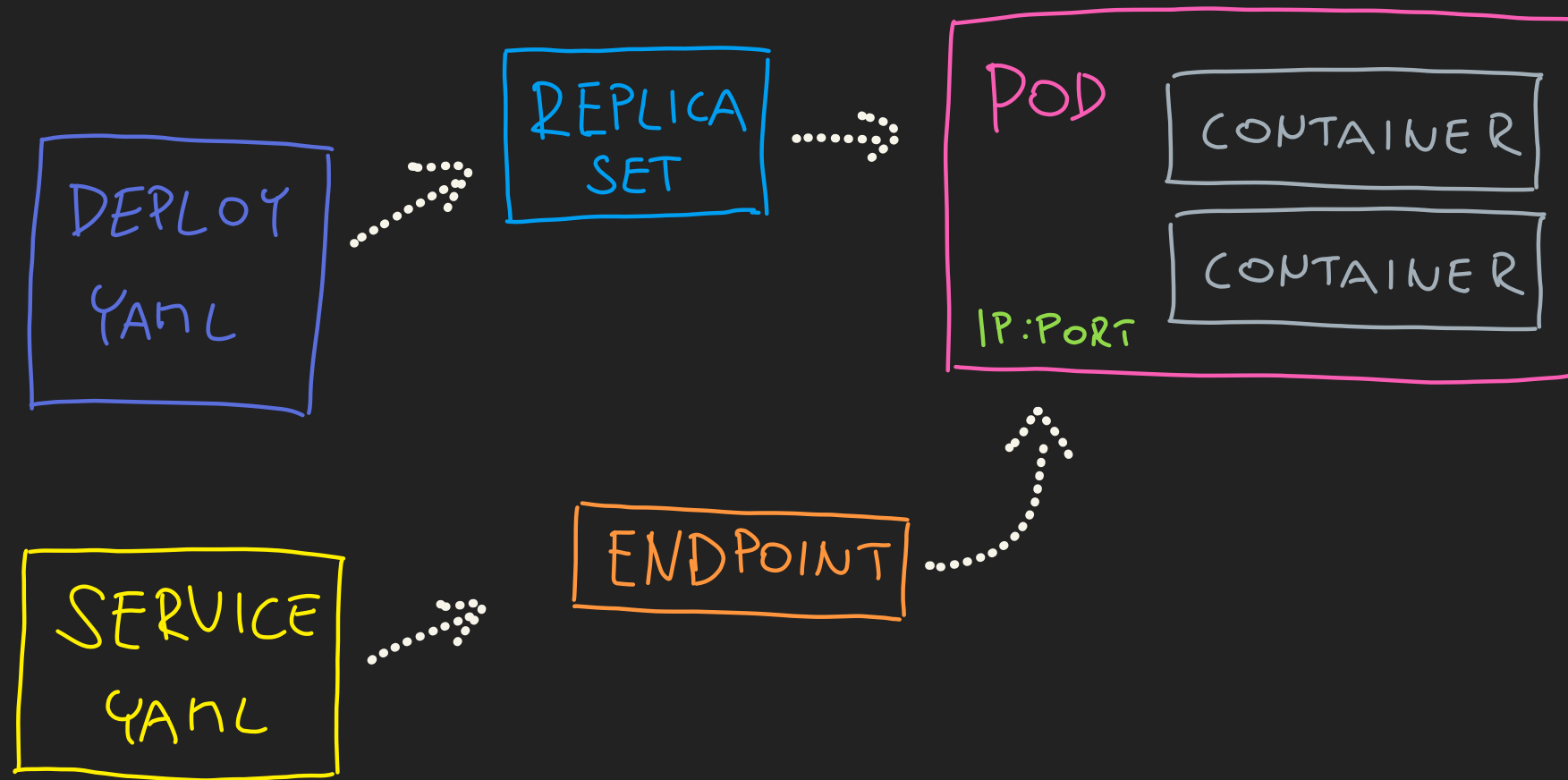
Tomáš Kukrál

[linkedin.com/in/tomaskukral](https://www.linkedin.com/in/tomaskukral)

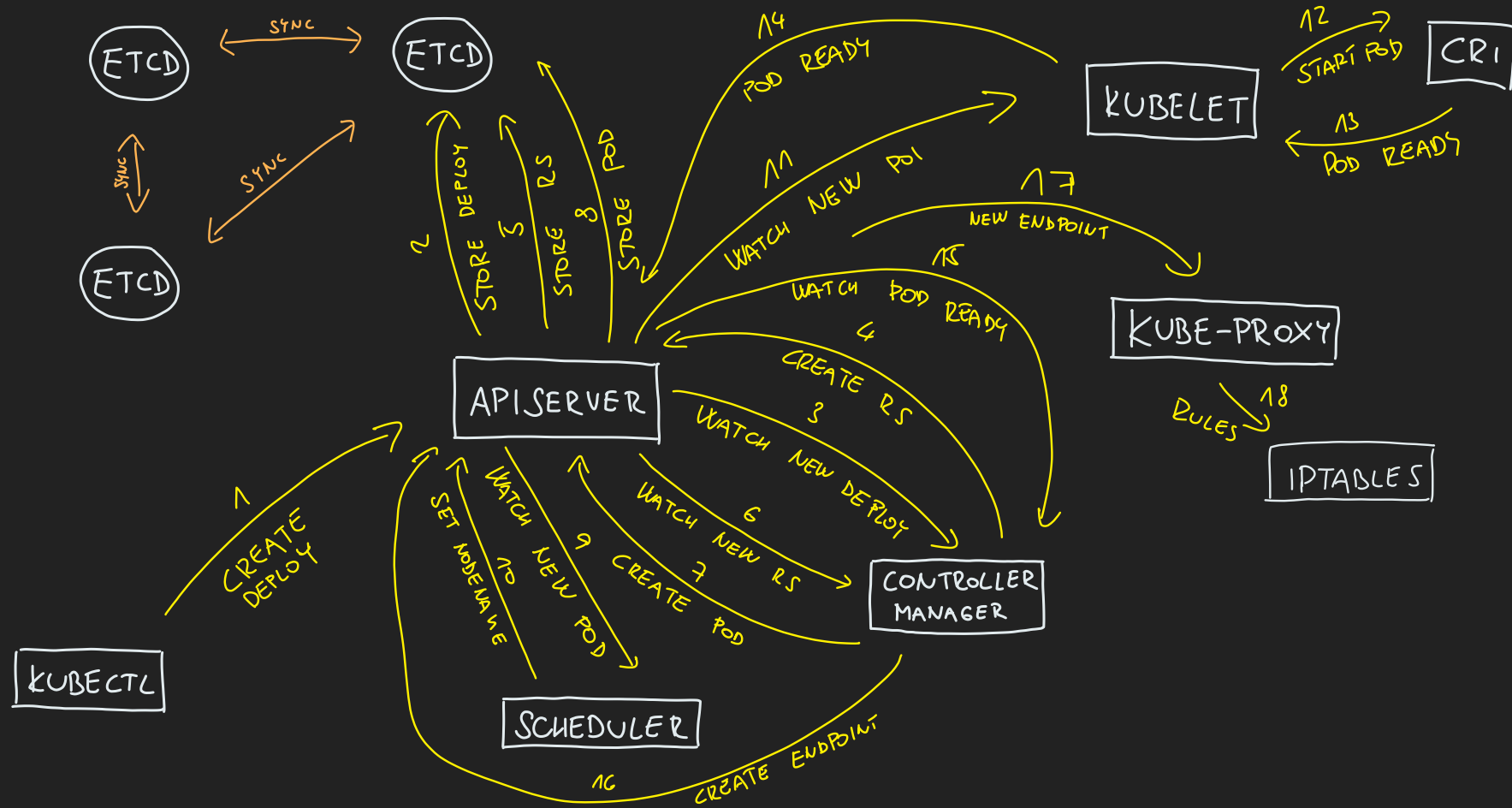
second-foundation.eu, datascript.cz

ex CVUT FIT, tcpCloud, Mirantis, Volterra, F5

Kubernetes in 42 seconds



Components



PKI structure

Authorities (Kubernetes, etcd, front-proxy) and certificates

- Server certificates
 - apiserver
 - etcd (server, client, peer)
 - kubelet
 - (front-proxy)
- Client certificates
 - kubelet -> apiserver
 - apiserver -> etcd
 - controller-manager -> apiserver
 - scheduler -> apiserver
 - kube-proxy -> apiserver
 - kubeconfig
 - (front-proxy)

```
[  
/etc/kubernetes/pki/ca.crt  
/etc/kubernetes/pki/ca.key  
    /etc/kubernetes/pki/apiserver.crt  
    /etc/kubernetes/pki/apiserver-etcd-client.crt  
    /etc/kubernetes/pki/apiserver-etcd-client.key  
    /etc/kubernetes/pki/apiserver.key  
    /etc/kubernetes/pki/apiserver-kubelet-client.crt  
    /etc/kubernetes/pki/apiserver-kubelet-client.key  
/etc/kubernetes/pki/etcd/ca.crt  
/etc/kubernetes/pki/etcd/ca.key  
    /etc/kubernetes/pki/etcd/healthcheck-client.crt  
    /etc/kubernetes/pki/etcd/healthcheck-client.key  
    /etc/kubernetes/pki/etcd/peer.crt  
    /etc/kubernetes/pki/etcd/peer.key  
    /etc/kubernetes/pki/etcd/server.crt  
    /etc/kubernetes/pki/etcd/server.key  
/etc/kubernetes/pki/front-proxy-ca.crt  
/etc/kubernetes/pki/front-proxy-ca.key  
    /etc/kubernetes/pki/front-proxy-client.crt  
    /etc/kubernetes/pki/front-proxy-client.key  
  
/etc/kubernetes/pki/sa.key  
/etc/kubernetes/pki/sa.pub  
]
```

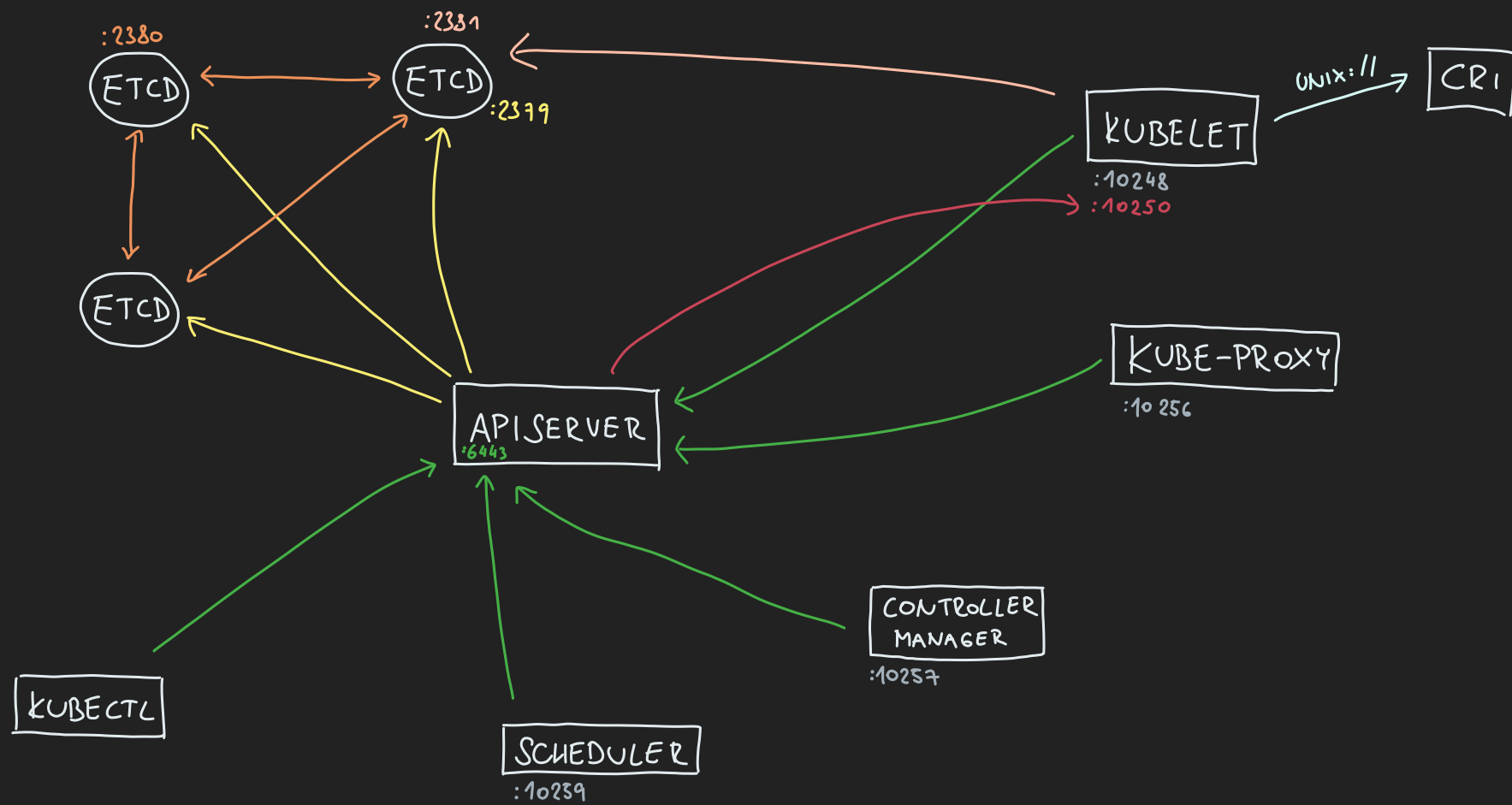
Certificate generation

kubeadm

- Native Kubernetes
- Blackbox
- One shot
- Renew on update

DYI

- Mint CAs
- Sign certificates (CSR)
- Monitor and rotate
- Trust chain



Validity

kubeadm certs check-expiration				
CERTIFICATE	EXPIRES	RESIDUAL TIME	CERTIFICATE AUTHORITY	EXTERNALLY MANAGED
admin.conf	Oct 28, 2026 20:06 UTC	29d	ca	no
apiserver	Oct 28, 2026 20:06 UTC	29d	ca	no
apiserver-etcd-client	Oct 28, 2026 20:06 UTC	29d	etcd-ca	no
apiserver-kubelet-client	Oct 28, 2026 20:06 UTC	29d	ca	no
controller-manager.conf	Oct 28, 2026 20:06 UTC	29d	ca	no
etcd-healthcheck-client	Oct 28, 2026 20:06 UTC	29d	etcd-ca	no
etcd-peer	Oct 28, 2026 20:06 UTC	29d	etcd-ca	no
etcd-server	Oct 28, 2026 20:06 UTC	29d	etcd-ca	no
front-proxy-client	Oct 28, 2026 20:06 UTC	29d	front-proxy-ca	no
scheduler.conf	Oct 28, 2026 20:06 UTC	29d	ca	no
super-admin.conf	Oct 28, 2026 20:06 UTC	29d	ca	no
CERTIFICATE AUTHORITY	EXPIRES	RESIDUAL TIME	EXTERNALLY MANAGED	
ca	Sep 28, 2026 20:11 UTC	4m32s	no	
etcd-ca	Sep 28, 2026 20:11 UTC	4m32s	no	
front-proxy-ca	Sep 28, 2026 20:11 UTC	4m32s	no	

- 10k days ... 27 years
- kubeadm ... 10 years

```
[  
apiVersion: kubeadm.k8s.io/v1beta4  
kind: ClusterConfiguration  
certificateValidityPeriod: 8760h # 1y  
caCertificateValidityPeriod: 87600h # 10y  
]
```

Certificate rotation

Rotation:

- Control plane
- Nodes - auto/manual
- Inside pods: token, ca.crt
- in-cluster: CNI, CSI, Ingress, ArgoCD, Grafana, Prometheus, Cert-manager, ...
- User's kubeconfig

- Automatic
- Manual
- Missed

```
Input Format: PEM
Valid: 2024-09-04 11:51 UTC to 2054-09-04 12:01 UTC
Subject:
    CN=ca
Issuer:
    CN=ca
```

Node certificate renewal

- Automatic renewal, if
 - kubelet is running
 - kubelet has valid bootstrap kubeconfig
 - kubelet has valid client certificate (kubelet.conf)
 - kubelet has valid CA certificate (ca.crt)
 - `rotateCertificates: true`



Common problems

- Control plane cert expired

```
Unable to connect to the server: tls: failed to verify certificate: x509: certificate has expired or is not yet valid: current time 2026-10-02T14:28:37Z is after 2026-10-02T14:17:12Z
```

- Node cert expired

```
Oct 02 14:29:50 ip-10-255-0-10.eu-central-1.compute.internal kubelet[14699]: E1002 14:29:50.924342 14699 kubelet_node_status.go:479] "Error updating node status, will retry" err="error getting node \"ip-10-255-0-10.eu-central-1.compute.internal\": Get \"https://10.255.0.10:6443/api/v1/nodes/ip-10-255-0-10.eu-central-1.compute.internal?timeout=10s\": tls: failed to verify certificate: x509: certificate has expired or is not yet valid: current time 2026-10-02T14:29:50Z is after 2026-10-02T14:17:12Z"
```

- Authority expired

Demo time!

Kubeadm does not support rotation or replacement of CA certificates out of the box.

For more information about manual rotation or replacement of CA, see manual rotation of CA certificates. [doc](#)

Commands executed during demo

```
# stop kubelet
systemctl stop kubelet

# stop control plane daemons
crictl stop $(crictl ps -q --name kube-scheduler)
crictl stop $(crictl ps -q --name kube-controller-manager)
crictl stop $(crictl ps -q --name kube-apiserver)
crictl stop $(crictl ps -q --name etcd)

# make sure it have stopped
crictl ps --name kube-

# delete all certificates and kubeconfig
rm -rfv /etc/kubernetes/pki/* /etc/kubernetes/{admin,controller-manager,kubelet,scheduler,super-admin}.conf

# generate certificates (and ca)
kubeadm init phase certs all --config /etc/kubernetes/config.yml

# generate kubeconfigs
kubeadm init phase kubeconfig all --config /etc/kubernetes/config.yml

# check certificates
kubeadm certs check-expiration

# restart kubelet
systemctl restart kubelet

# check node status
kubectl get no

# delete old node certificates
rm -v /var/lib/kubelet/pki/kubelet-client*
kubectl delete csr --all

# generate new node certificate
export node NODE=ip-10-255-0-10.eu-central-1.compute.internal
kubeadm kubeconfig user --config /etc/kubernetes/config.yml --org system:nodes --client-name system:node:$NODE --validity-period 600s | tee /etc/kubernetes/kubelet.conf

# restart kubelet
systemctl restart kubelet
```

```
# wait for certificate to be generated
ls -l /var/lib/kubelet/pki
kubect! get csr -o yam! | yq .items[0].status.certificate -r | base64 -d | certigo dump

# inspect kubelet auto certificate
certigo dump /var/lib/kubelet/pki/kubelet-client-current.pem
# inspect kubelet certificate
cat /etc/kubernetes/kubelet.conf | yq '.users[0].user."client-certificate-data"' -r | base64 -d | certigo dump --format PEM

# kubelet config with file paths
cat /etc/kubernetes/kubelet-files.conf | tee /etc/kubernetes/kubelet.conf
# restart kubelet
systemctl restart kubelet

# fix workload
kubect! -n traefik rollout restart deploy,sts,ds
kubect! -n kube-system rollout restart deploy,sts,ds

# fix kubeadm configuration
kubect! -n kube-public get cm cluster-info -o yam! | yq .data.kubeconfig -r | yq '.clusters[0].cluster."certificate-authority-data"' -r | base64 -d | certigo dump --format PEM

# fix kubeadm bootstrap/join
base64_encoded_ca="$(base64 -w0 /etc/kubernetes/pki/ca.crt)"
kubect! get cm/cluster-info --namespace kube-public -o yam! | sed "s/\\(certificate-authority-data:\\).*/\\1 ${base64_encoded_ca}/" | kubect! apply -f -
```

Improvements

- Kubeadm single place for CA
- Kubeadm documentation
- Kubeadm documentation
- `CN=kubernetes` configurable
- Monitoring: node certs, control plane certs, pod certs
- CA automation
- Validity monitoring
- Cattle vs Pet cluster
- Exercise!

BA