Certifik8s: All you need to know about certificates in Kubernetes

Alexander Brand
@alexbrand

Background

Early 2016

Started working on Kubernetes and getting involved with the community

November 2016

Initial release of the Kismatic Enterprise Toolkit (KET)

April 2017

RBAC support in Kubernetes goes to beta

May 2017

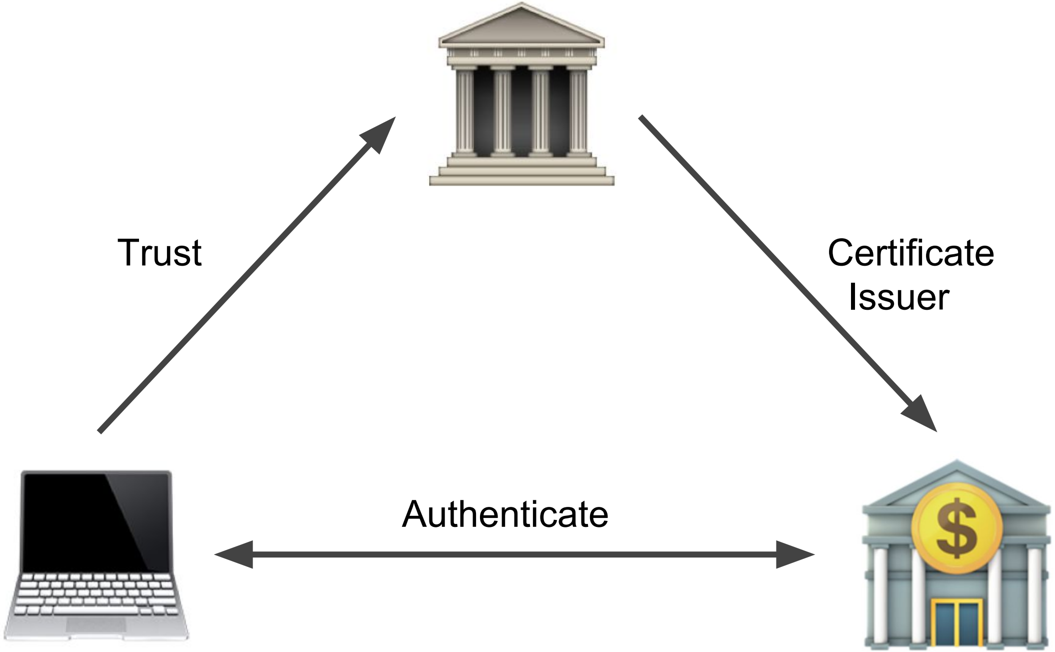
Revamped certificate generation process in KET

Agenda

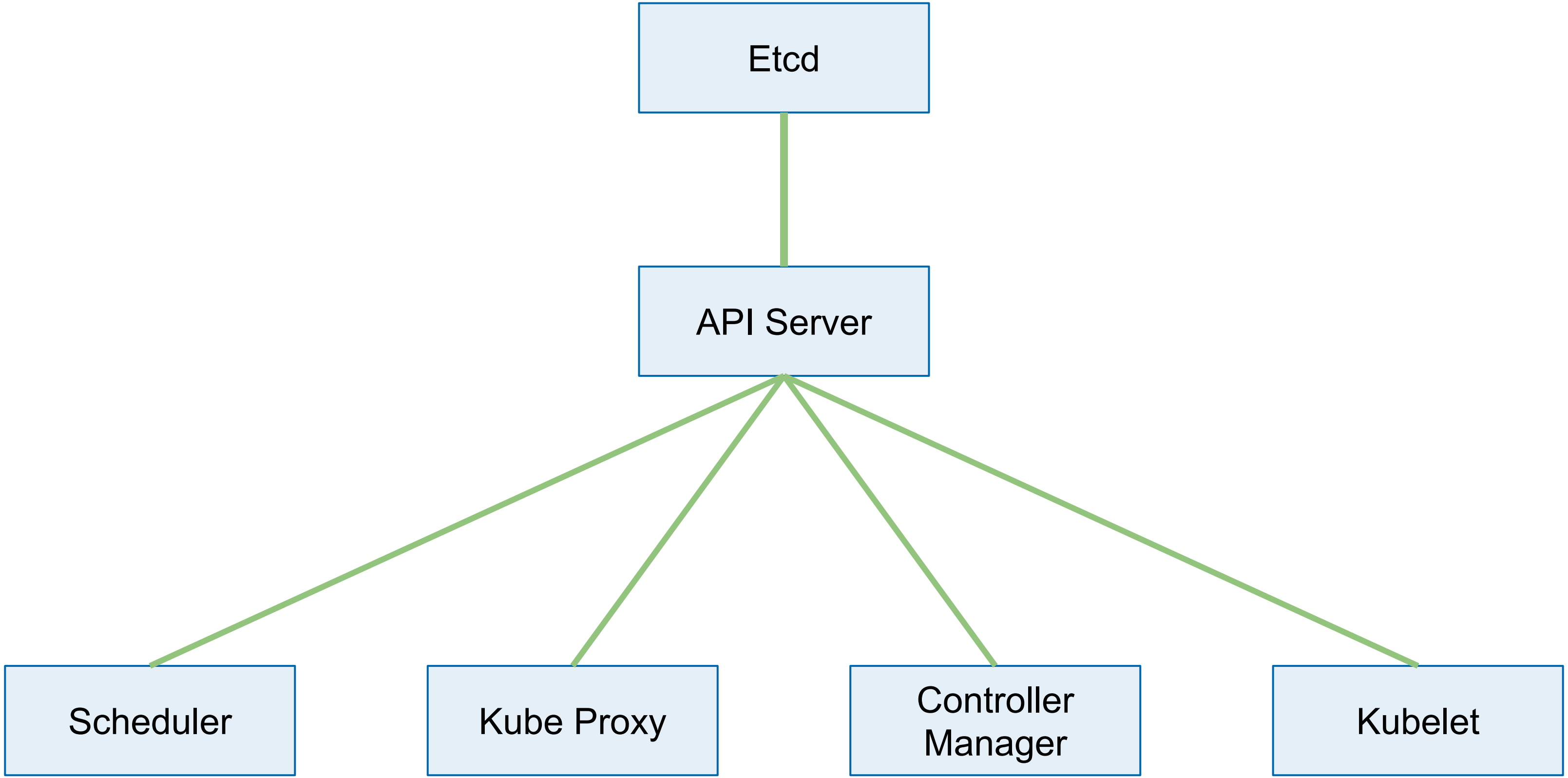
- Cluster Certificate Authority (CA)
- API Server HTTPS and High Availability
- Kubelet HTTPS
- X.509 Client Certificate Authentication Strategy
- Certificate Generation API
- Kubelet Cert Bootstrapping and Cert Rotation
- Further topics

Certificates Refresher

- Certificates enable the authentication of parties in a conversation
- Client authenticates the server and the server can authenticate the client
- Enable the first steps in the TLS Handshake
- Certificate Authority
- Certificate Signing Request


Trust
Certificate Issuer
Authenticate

Why do I need certificates in Kubernetes?


Etcd
API Server
Scheduler
Kube Proxy
Controller Manager
Kubelet

Cluster CA

Cluster CA

- Cluster Certificate Authority is the trusted root for the entire cluster
- All cluster certificates are signed by the Cluster CA
- Used by components to validate API server, etc

API Server HTTPS

API Server HTTPS

- Serving certificate and key are required for HTTPS
- Serving certificate is signed by Cluster CA
- Components authenticate the API server
- Configured using `--tls-cert-file` and `--tls-private-key-file` flags

https://kubernetes.io/docs/reference/generated/kube-apiserver/

HA Considerations

- Multiple API servers must be fronted with a load balancer
- Each master has its own certificate
- Load balancer’s DNS name and IP address should be part of the certificate’s Subject Alternative Name (SAN) field
- Clients will complain otherwise `NET::ERR_CERT_COMMON_NAME_INVALID`

Kubelet HTTPS

Kubelet HTTPS

- The Kubelet exposes an API over HTTPS
- Consumed by API server when getting logs, metrics, exec, etc.
- Serving certificate and key are required for HTTPS
- Certificate is signed by Cluster CA
- API server authenticates the Kubelet

Kubelet HTTPS

- Access to the Kubelet API is protected by authentication and authorization
- The Kubelet authenticates clients using client certificates
- API server has a Kubelet client certificate that is signed by Cluster CA

https://kubernetes.io/docs/admin/kubelet-authentication-authorization/

X.509 Client Cert Authentication

X.509 Client Cert Authentication

- Strategy for authenticating requests that present a client certificate
- Mainly used by Kubernetes components, but can also be used for end user authentication
- Any request that presents a client certificate signed by the Cluster CA is authenticated
- User is obtained from Common Name (CN) field
- Groups are obtained from Organization field

https://kubernetes.io/docs/admin/authentication/#x509-client-certs

X.509 Client Cert Authentication

- Each Kubernetes core component has its own client certificate

Component	Common Name	Organizations
Controller Manager	system:kube-controller-manager	
Scheduler	system:kube-scheduler	
Kube Proxy	system:kube-proxy	
Kubelet	system:node:${hostname}	system:nodes

https://kubernetes.io/docs/admin/authorization/rbac/#core-component-roles

Kubelet Client Certificates

- Each Kubelet on the cluster has its own identity
- Achieved by having Kubelet-specific client certificates
- Enables the use of the Node Authorizer and Node Restriction Admission Plugin
- Limit Kubelet read and write access to resources that are related to the node itself and pods bound to the node

https://kubernetes.io/docs/admin/authorization/node/

Certificate Generation API

Certificate Generation API

- Kubernetes offers an API to request certificates

  `certificates.k8s.io/v1beta1`
- Clients create a certificate signing request and send it to the API server
- The requesting user is stored as part of the CSR resource
- CSR remains in a pending state, until approved by a cluster admin
- The certificate is issued once the CSR request is approved

https://kubernetes.io/docs/tasks/tls/managing-tls-in-a-cluster/

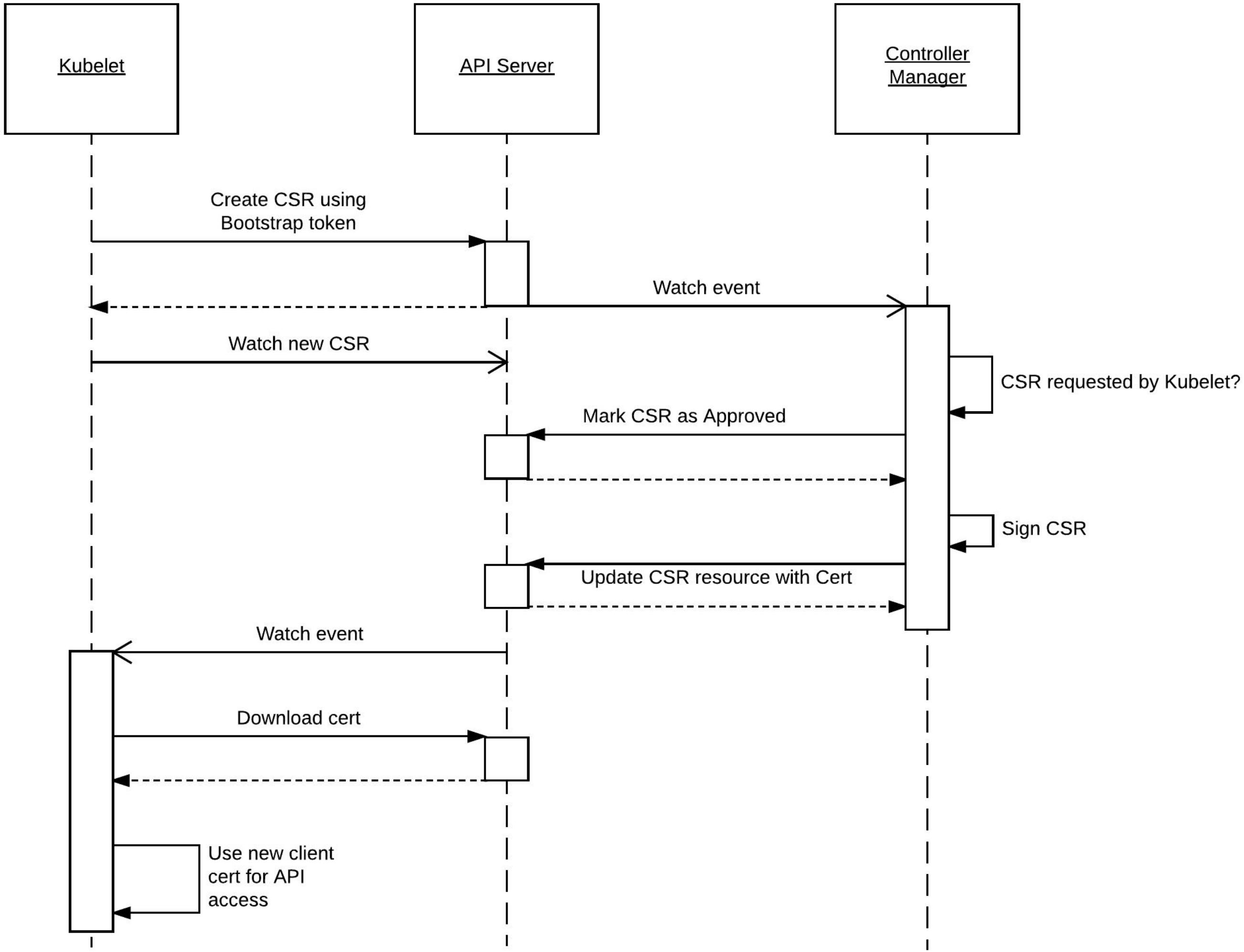
Demo

Kubelet Cert Bootstrapping and Rotation

Kubelet Cert Bootstrapping

- The Kubelet needs a client certificate to access the API server
- It also needs a serving certificate for its own API
- Instead of the admin having to generate certificates for each Kubelet, the Kubelet can request certificates as it starts up
- Built on top of the Certificates API and Bootstrap token authenticator

https://kubernetes.io/docs/admin/kubelet-tls-bootstrapping/


Kubelet
API Server
Controller Manager
Create CSR using Bootstrap token
Watch event
Watch new CSR
CSR requested by Kubelet?
Mark CSR as Approved
Sign CSR
Update CSR resource with Cert
Watch event
Download cert
Use new client cert for API access

Kubelet Cert Bootstrapping Steps

1. Kubelet creates CSR using Bootstrap token
2. `CSRApprovingController` approves the CSR automatically
3. `CSRSignerController` signs the CSR
4. Kubelet downloads the generated certificate and starts using it

https://kubernetes.io/docs/admin/kubelet-tls-bootstrapping/

Kubelet Cert Rotation

- As of Kubernetes 1.8, the Kubelet can request a new client certificate when the current one is nearing expiration (Beta)
- It can also rotate the serving certificate (Alpha, must be enabled with feature flag)

https://github.com/kubernetes/features/issues/266

https://github.com/kubernetes/features/issues/267

Further Topics

Certificate Revocation List

- Kubernetes does not currently support CRLs
- Can use RBAC to “revoke” them
- Discussion around CRLs

  https://github.com/kubernetes/kubernetes/pull/33519

Ingress

- TLS can be configured for services exposed using Ingress
- Define a secret with a certificate and private key, and reference it in the ingress resource
- kube-lego: Auto cert generation using Let’s Encrypt for Ingress

https://kubernetes.io/docs/concepts/services-networking/ingress/#tls

https://github.com/jetstack/kube-lego

Workload Identity

- Kubernetes Container Identity Working Group
- Allow containers to prove their identity
- Use cases include accessing external systems, service-to-service mutual TLS, etc.
- https://docs.google.com/document/d/1uH60pNr1-jBn7N2pEcddk6-6NTnmV5qepwKUJe9tMRo/edit#

Summary

Summary

- Certificates are key to the functioning of a secure Kubernetes cluster
- Kubernetes is a distributed system - components run on different nodes and talk to each other over the network
- Certificates enable Kubernetes components to perform mutual authentication
- Kubernetes offers an API for requesting/generating certificates
- Kubelets are capable of using this API for cert bootstrapping and rotation

Component	Certificate	Purpose
API server	Cluster CA	Authenticate clients, TLS
API server	Etcd CA	Etcd server authentication
API server	Etcd client cert	Etcd client authentication
API server	Serving certificate	Serving API over HTTPS
API server	Kubelet client cert	Authenticating against Kubelet
Controller Manager	Client certificate	Authenticating against API server
Controller Manager	Cluster CA	Embedding in service account secrets
Scheduler	Client certificate	Authenticating against API server
Kubelet	Serving certificate	Serving API over HTTPS
Kubelet	Client certificate	Authenticating against API server
Kubelet	Cluster CA	Authenticating clients
Kube Proxy	Client certificate	Authenticating against API server

Thanks!