CertService and K8s Cert-Manager integration

Problem statement

ONAP has a lot of components which enroll certificates:

- Legacy AAF CertMan which uses SCEP protocol or own internal Certificate Authority - mostly used by AT&T and integrated with several ONAP components
- New CertService which uses CMPv2 to enroll certificates - integrated with ONAP bordering components to protect external traffic
- K8s Cert-Manager which is OOM way forward to enroll certificates for ONAP components and de-facto industry standard for K8s based clouds

It is time to unify them and use forward just one of them.

K8s Cert-Manager Introduction

K8s Cert-Manager is an industry standard to issue X.509 certificates to K8s workloads. It provides simple, reliable, elastic and efficient way to issue certificates within K8s environment. Simple cause it relies on K8s custom resource definition (CRD) mechanism, reliable cause without secret created by Cert-Manager K8s workload won't start, elastic cause it can provide certificates from many sources, including external issuers and efficient - cause it may deliver hundreds of certificates per day.

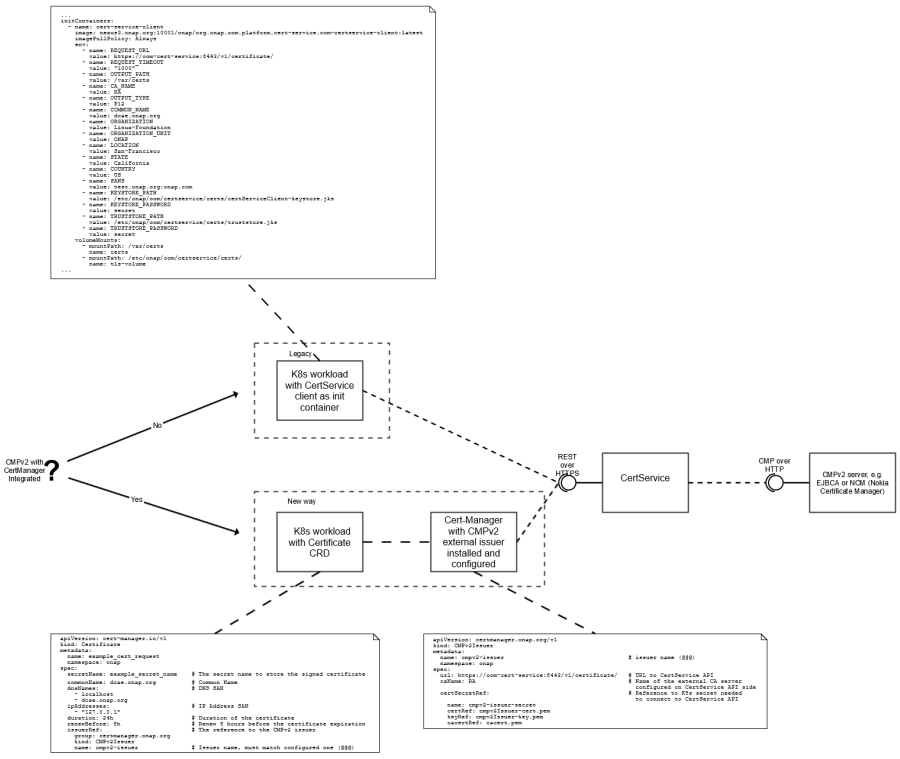
Brief introduction to Cert-Manager usage

As previously mentioned, Cert-Manager consumes Certificate CRD to issue certificates. But before first certificate is issued, Issuer or ClusterIssuer CRD has to be configured first. For basic use cases that's all. One can use very rich functionality to configure issued certificate and use various types of issuers.

Underneath, out of Certificate CRD, Cert-Manager creates CertificateRequest CRD which is more suitable for M2M processing as it contains Certificate Signing Request (CSR). CertificateRequest CRD is further on consumed by Issuer which processes CSR stored there and in return puts signed certificate and trusted certs in CertificateRequest's status and marks CertificateRequest as **Ready**. Cert-Manager notices such state change and from CertificateRequest marked as Ready creates K8s secret originally requested in Certificate CRD. Such K8s secret is ready to be mounted to K8s workload as any other secret.

Way forward

CertService was implemented some time back. It provides basic certificate enrollment functionality using CMPv2 over HTTP. Cause in ONAP we have a lot of components which issue certificates, it is wise to harmonize them and use forward just one of them. As a way forward, CertService should be a backend proxy service for K8s Cert-Manager. The same functionality which is currently implemented in CertService client should be implemented in Cert-Manager's external issuer, except parts which are already implemented in Cert-Manager and are generic for all issuers (e.g. input parameters validation, conversion to different format, etc). If possible the same input parameters which nowadays are used by CertService client should also be used by Certificate CRD. Following diagram presents new setup.



As a consequence, all existing usages should be adjusted to use new way and create Certificate CRD instead of calling CertService client as init container.

In fact, Cert-Manager's external issuer is a K8s operator which reconciles two CRDs:

- Cert-Manager's CertificateRequest CRD
- New Issuer CRD

External issuer notices that CertificateRequest CRD with a reference to New Issuer CRD was created and processes CSR stored there and in result puts signed certificate and trusted certs and marks CertificateRequest as Ready. Simple as that.

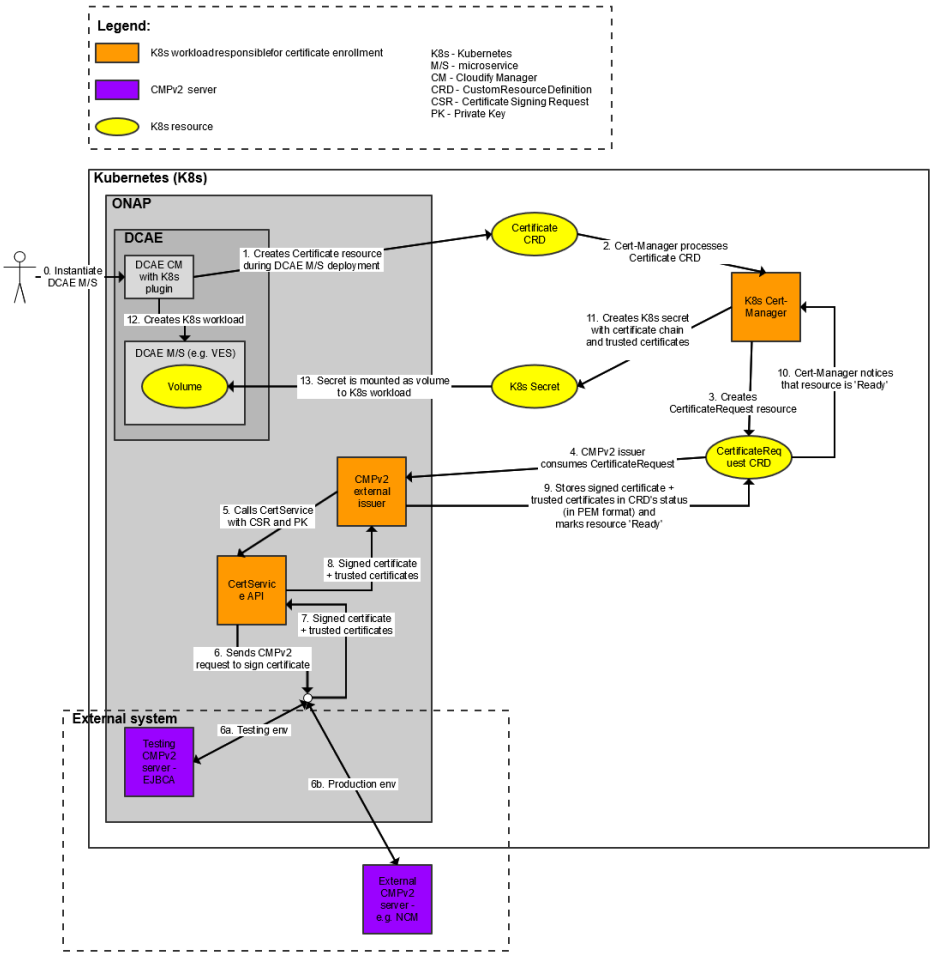
Usages

Helm templates

In helm templates the way forward is simple. Certificate CRD must be added and K8s workload must be enhanced to mount secret created out of Certificate CRD.

DCAE blueprints

The same functionality in DCAE is more complicated cause K8s Cloudify plugin code must be extended to create Certificate CRD instead of adding init container. Following diagram presents flow for DCAE microservice deployment when CMPv2 and Cert-Manager integration is enabled.



Limitations

After detailed check found out that K8s Cert-Manager doesn't correctly handle issuer's response which contains multiple trusted certificates, aka root CAs. For that following community bugs were reported:

1. Add multiple trustedCertEntries to truststores
2. JKS and PKCS12 Keystores are inconsistent

Future

CertService API enhancements

1. Add support within CertService API for parameters sent in CSR which are supported by Cert-Manager's Certificate API
2. CMPv2 over HTTPS support

Cert-Manager native support

There is an open feature request (FR) to support CMPv2 natively in Cert-Manager - https://github.com/jetstack/cert-manager/issues/2619

If such would be implemented, it is beneficial to use it instead of our custom solution.