DCAE CertService integration

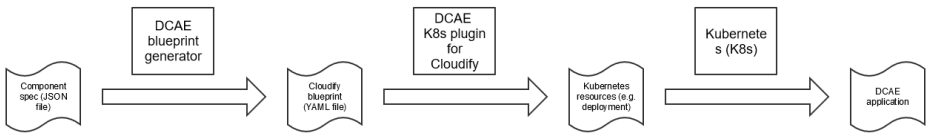
Page describes planned Guilin contribution

Overview

In Frankfurt release AAF was enhanced by Certificate Management Protocol ver. 2 (CMPv2) support. Such support is handled by new AAF's microservice called CertService. CertService provides certificates signed by external CMPv2 server - further on such certificates are called operators certificates. Operators certificates are meant to secure external ONAP traffic - traffic between network functions (xNFs) and ONAP.

Together with CertService, AAF provides CertService's client, which is a docker image meant to be used to call CertService API in a secure way. CertService's client should be invoked by other ONAP components as Init container, which has a dedicated role - acquire certificate and pass it to application container. Call to CertService can be controlled by environment variables which are passed to this init container.

DCAE components are instantiated in ONAP by Cloudify which consumes Cloudify blueprints. Valid Cloudify blueprints are generated from component specs by DCAE blueprint generator. Following diagram presents end to end flow.



Goal

Goal of this feature is to integrate DCAE with CertService to acquire operator certificates meant to protect external traffic between DCAE's components (VES collector, HV-VES, RestConf collector and DFC) and xNFs. For that reason K8s plugin which creates K8s resources from Cloudify blueprints must be enhanced with new TLS properties support. New TLS properties are meant to control CertService's client call in init containers section and environment variables which are passed to it.

This feature doesn't influence ONAP internal traffic which nowadays is protected by certificates issued by AAF's CertMan. If any affected component doesn't distinguish between internal and external communication, such needs to be introduced. So extra goal of this feature is to clearly fence these two communications and still use certificates issued by AAF's CertMan in internal ONAP traffic, while use operators certificates in external traffic.

Current state

Within Frankfurt release only one of the previously listed DCAE components (DFC) clearly distinguish between external and internal traffic. Other components (VES collector and HV-VES) uses certificates enrolled by AAF's CertMan to protect both - external and internal traffic.

DFC

As already mentioned, DFC protects external and internal traffic using two different pairs of certificate and trusted certificates. Such are configured by properties:

```
# External traffic:
dmaap.ftpesConfig.keyCert: /opt/app/datafile/config/cert.jks
dmaap.ftpesConfig.keyPasswordPath: /opt/app/datafile/config/jks.pass
dmaap.ftpesConfig.trustedCa: /opt/app/datafile/config/trust.jks
dmaap.ftpesConfig.trustedCaPasswordPath: /opt/app/datafile/etc/cert/trust.pass

# Internal traffic:
dmaap.security.enableDmaapCertAuth: false
dmaap.security.keyStorePasswordPath: /opt/app/datafile/etc/cert/key.pass
dmaap.security.keyStorePath: /opt/app/datafile/etc/cert/key.p12
dmaap.security.trustStorePasswordPath: /opt/app/datafile/etc/cert/trust.pass
dmaap.security.trustStorePath: /opt/app/datafile/etc/cert/trust.jks
```

VES collector

VES collector protects both external and internal traffic using the same certificate and trusted certificates. Such are configured by properties:

```
# Current one is actually external one:
collector.keystore.file.location: /opt/app/dcae-certificate/cert.jks
collector.keystore.passwordfile: /opt/app/dcae-certificate/jks.pass
collector.truststore.file.location: /opt/app/dcae-certificate/trust.jks
collector.truststore.passwordfile: /opt/app/dcae-certificate/trust.pass
```

HV-VES collector

HV-VES collector protects both external and internal traffic using the same certificate and trusted certificates. Such are configured by properties:

```
# Current one is actually external one:
security.keys.keyStoreFile: /etc/ves-hv/ssl/cert.jks
security.keys.keyStorePasswordFile: /etc/ves-hv/ssl/jks.pass
security.keys.trustStoreFile: /etc/ves-hv/ssl/trust.jks
security.keys.trustStorePasswordFile: /etc/ves-hv/ssl/trust.pass
```

RestConf collector

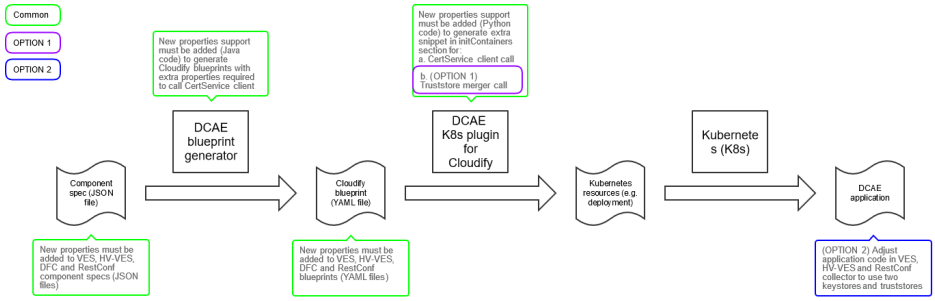
RestConf collector has two keystores and one truststore. One keystore is used to protect RestConf's REST API while second is used to protect communication between RestConf collector and external controllers. The same truststore is used to protect both communications. Right now its is unknown (?) if the same truststore is used when RestConf collector communicates with DMaaP.

```
# Keystore used to protect RestConf's REST API:
collector.keystore.file.location: "/opt/app/restconfcollector/etc/sdnc.p12"
collector.keystore.passwordfile: "/opt/app/restconfcollector/etc/passwordfile"

# Seems this alias is used to lookup correct certificate from keystore used to protect RestConf's REST API:
collector.rcc.keystore.alias: "dynamically generated"

# Keystore used to protect communication between RestConf collector and external controllers:
collector.rcc.keystore.file.location: "/opt/app/restconfcollector/etc/keystore"
collector.rcc.keystore.passwordfile: "/opt/app/restconfcollector/etc/rcc_passwordfile"

# Truststore used to protect both external communications
collector.rcc.truststore.file.location: "/opt/app/restconfcollector/etc/truststore.onap.client.jks"
collector.rcc.truststore.passwordfile: "/opt/app/restconfcollector/etc/trustpasswordfile"
```

Way forward

Overview

Following diagram presents contribution overview.



Blueprint generator and K8s plugin

So, to implement goal of this feature, both: blueprint generator and K8s plugin must be enhanced to support following new blueprint properties in new external_cert section.

```
external_cert:
        use_external_tls
        external_cert_directory
        ca_name
        external_certificate_parameters:
                common_name
                sans
```

Meaning of properties is described in following table. CertService's client properties are described in details on a dedicated page.

* - property available in blueprint inputs, so can be changed every deployment

** - property available in blueprint, doesn't need to be changed every deployment

Group	Property name	Component spec type	Blueprint type (input* /blueprint**)	Default	Description
external_cert	use_external_tls	input	input	true	A boolean that indicates whether the component uses AAF CertService to acquire operator certificate to protect external (between xNFs and ONAP) traffic. For a time being only operator certificate from CMPv2 server is supported
	external_cert_directory	hardcoded in BP Generator	blueprint	/opt/app/dcae-certificate /external	Directory where operator certificate and trusted certs should be created
	ca_name	hardcoded in BP Generator	input	RA	Name of Certificate Authority configured on CertService side (in cmpServers.json). Default RA_TEST corresponds to default CMPv2 testing configuration.
	output_type	hardcoded in BP Generator	input	P12	Certificate output type
external_cert: external_certificate_parameters	common_name	hardcoded in BP Generator	input	<Specific for every blueprint>	Common name which should be present in certificate. Specific for every blueprint (e.g. dcae-ves-collector for VES)
	sans	hardcoded in BP Generator	input	<Specific for every blueprint>	List of Subject Alternative Names (SANs) which should be present in certificate. Delimiter - : Should contain common_name value and other FQDNs under which given component is accessible, e.g. if xNFs uses ves-collector in request URL, such should be also present in SANs - e.g. dcae-ves-collector: ves-collector.

If new properties are provided by blueprint and use_external_tls is set to true, K8s plugin must be able to create init containers section and within it add information about CertService's client image and pass all other variables as environment variables. Section very similar to example described on a dedicated page.

Configuration specific only for K8s plugin

Additionally only K8s plugin must be enhanced to support extra properties in K8s plugin configuration listed in following table. All such parameters must be configured using appropriate global helm CMPv2 properties and stored in K8s plugin configuration file.

Group	Property name	Origin	Default	Description
external_cert	image_tag	global helm value	nexus3.onap.org:10001/onap/org.onap.aaf.certservice.aaf-certservice-client:$VERSION	CertService client image name and version
	request_url	global helm value	https://aaf-cert-service:8443/v1/certificate/	URL to Cert Service API
	timeout	global helm value	30000	Request timeout. Needs to be taken from global CMPv2 helm variable
	country	global helm value	US	Country name in **ISO 3166-1 alpha-2** format, for which certificate will be created. Needs to be taken from global CMPv2 helm variable
	organization	global helm value	Linux-Foundation	Organization name, for which certificate will be created. Needs to be taken from global CMPv2 helm variable
	state	global helm value	California	State name, for which certificate will be created. Needs to be taken from global CMPv2 helm variable
	organizational_unit	global helm value	ONAP	Organizational unit name, for which certificate will be created. Needs to be taken from global CMPv2 helm variable
	location	global helm value	San-Francisco	Location name, for which certificate will be created. Needs to be taken from global CMPv2 helm variable

DCAE component specs

Each component described above has its own component spec. Each has to be updated with all properties described in blueprint generator section.

DCAE bluerprints

Cloudify blueprints must be adjusted to take advantage of new K8s plugin functionality and must provide extra properties which controls CertService's client call.

Take into account X.509 certificates from CMPv2 server

There are two options to proceed with using certificates from CMPv2 server:

Option 1 (DCAE extra init container (aka trust merger))

Keep application intact and implement truststores merger and invoke it as new init container to provide to application one truststore with multiple trust anchors taken from multiple truststores and **one keystore with certificate from CMPv2 server**.

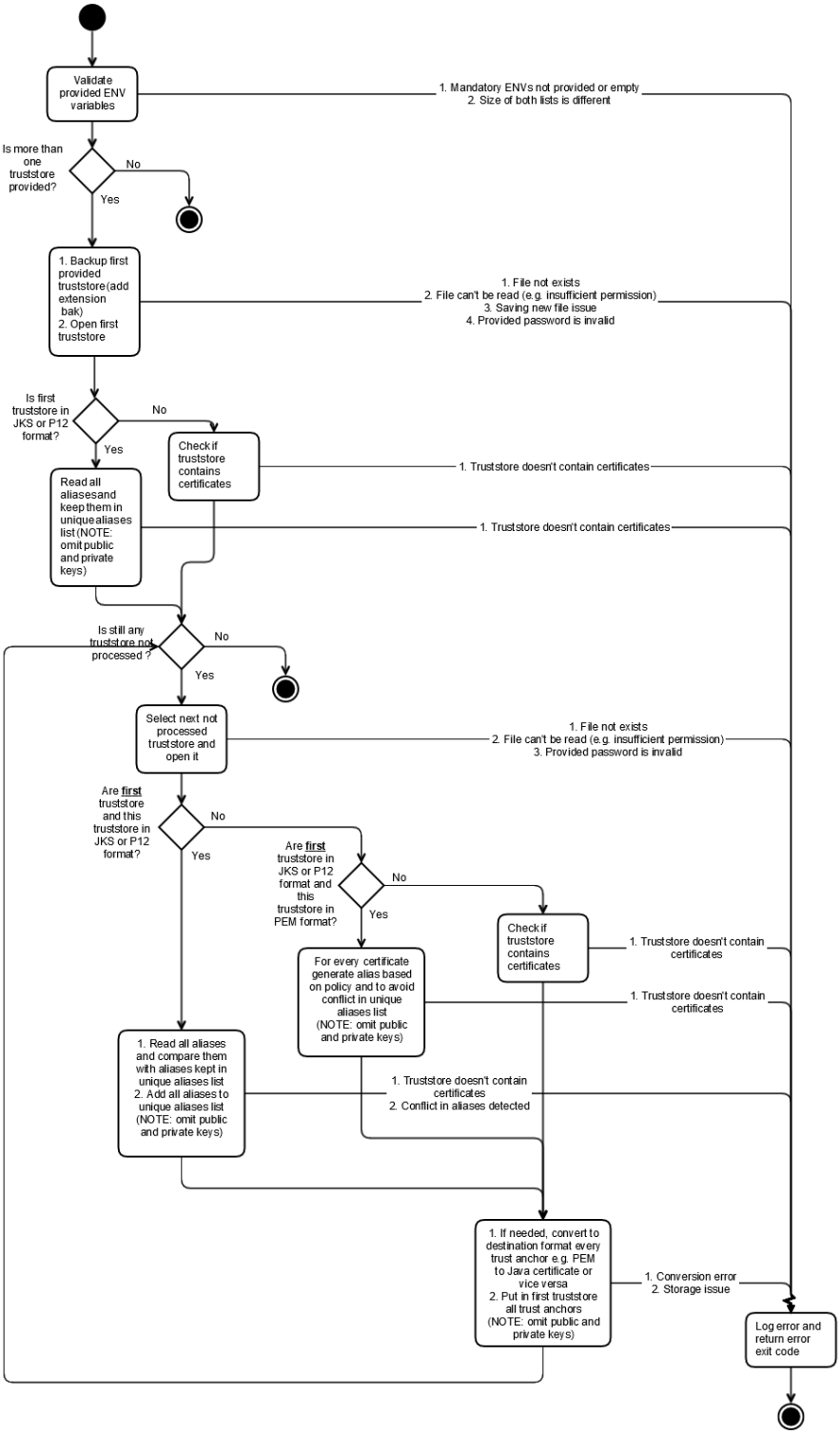
Optionally adjust components (e.g. DFC) which use different certificates internally and externally to support the same truststore and keystore on both traffics.

Truststore merger properties

Property name	Example	Description
TRUSTSTORES_PATHS	/etc/dcae/truststore.jks:/etc/dcae/cacert.pem:/etc/dcae/truststore2.p12	List of truststores to be merged. Certificates from all provided truststores will be added to first provided truststore after success execution.

TRUSTSTORES_PASSWORDS_PATHS	/etc/dcae/truststore.pass::/etc/dcae/truststore2.pass	List of passwords to provided truststores - order must be the same as in truststores WARNING: PEM is not protected by password so its value should be empty

Truststore merger flow



Policy to generate new aliases for certificates from PEM files

Use as prefix pem-trusted-certificate- and $INDEX

Extra K8s plugin property

Group	Property name	Origin	Default	Description
truststore_merger	image_tag	global helm value	nexus3.onap.org:10001/onap/org.onap.dcae.truststore-merger:$VERSION	Truststore merger image name and version

Option 2 (Adjust DCAE components to support two internal and external truststores and keystores)

Components which don't distinguish between external and internal traffic must be adjusted to support different certificates and trusted certificates on both traffics separately.

Additionally both blueprint generator and K8s plugin must be adjusted to add extra properties to Config Binding Service (CBS). Such properties must be read by applications.

Following table presents four new properties stored in CBS.

Group	Property name	Default	Description
properties: application_config	external_keystore_path	/opt/app/dcae-certificate/external/keystore.jks	Path to keystore with external certificate
	external_keystore_password_path	/opt/app/dcae-certificate/external/keystore.pass	Path to password for keystore with external certificate
	external_truststore_path	/opt/app/dcae-certificate/external/truststore.jks	Path to truststore with external trust anchors
	external_truststore_password_path	/opt/app/dcae-certificate/external/truststore.pass	Path to password for truststore with external trust anchors

DCAE multisite deployment support

There are two ways to support DCAE multisite deployment:

- One which requires direct connectivity between EDGE cloud and CMPv2 server (which isn't so extraordinary if xNFs also use CMPv2 protocol to enroll certificates)
- One which doesn't require direct connectivity between EDGE cloud and CMPv2 server, but requires direct connectivity between EDGE cloud and central ONAP deployment.

Each option has its own benefits. Each requires different approach and procedure.

To correctly support first option, instance of CertService (server part) has to be deployed on every EDGE cloud, where DCAE collectors are expected to be running. Nothing else is required.

To correctly support second option, secret with certificate for CertService client has to be copied from central ONAP deployment to EDGE clouds, where DCAE collectors are expected to be running. On central ONAP deployment CertService has to be exposed outside K8s cluster. On every EDGE cloud proxy service is also required.