Don’t -k your services on k8s

Our journey of rolling out TLS for microservices on our Kubernetes clusters

Akshay Chitneni
Tilottama Gaat
VMware
whoami

Production Engineers
VMware CMBU

vRA Cloud - Provides single pane of glass for users to automate their workflows across environments
K8s 1.9 (1.5 - 1.13)
AWS EC2
40 microservices in Prod
~700 pipeline runs in a day
300 dev/sre across 3 geos USA, Bulgaria, India
What is TLS?

1. Ssl hello

2. Server responds with valid cert and public key

3. Client validates certificate and public key

4. Client responds with symmetric key

5. SSL session established
What is TLS? Contd.

Identification:

1. Company asks CA for root certificate
2. CA creates ROOT CA certificate and signs it
3. This Root CA will be installed in the webserver
4. Intermediate CA certificates used along with Root CA to create a chain of trust
5. Browser can now trust the certificates issued by the server
Kubernetes TLS challenges

- Microservices → more endpoints
- Many environments
- Custom Certificate Authorities and certificate signing processes.
- Certificate Renewal is a calendar event.
Problem statement 1.0

- Encrypted communication between microservices using TLS
- Distribute and Manage TLS certs
- Applicable for service environments running in containers (k8s)
- Highly available and fault tolerant
TLS Attempt 1.0

Provisioning Certificates 1.0
- **Diplomav1** - generates Certificates and CA certificates
- Uses Rest API (sanic)
- Vault and Consul

Distributing Certificates 1.0
- Deploy scripts of all services invoke diploma api
Diploma v1 vs Diplomav2
Migration to Diplomav2

Diploma V1

- Deploy_blueprintsService.py
- Deploy_provisioningService.py
- Deploy_GatewayServer.py
- Deploy_Kafka.py

Diploma V2

- Deploy_blueprintsService.py
- Deploy_provisioningService.py
- Deploy_GatewayServer.py
- Deploy_Kafka.py
Problems in Attempt 1.0

Provisioning Certificates 1.0 Problems
• Problems in diplomav1 design
• Not on k8s
• Want to move to diplomav2 but too many hurdles

Distributing Certificates 1.0 Problems
• Each service has deployment code touching diploma
• Want to move towards helm but cannot
• Renewing certs – onus on services
Problem statement 2.0

• Services will need to move to Diploma v2
  • New API for certs
  • Diff auth mechanism

• 40+ services in production
  • Multiple clusters
  • Multiple regions
  • Service specific Cert Config, varies per env
  • Service specific SLA, Release cycle
  • mtls

• Seamless migration without causing any downtime

• Minimal changes to existing deploy scripts.
Where to begin?

• Services use Diploma V1 API
  • Auth tokens
  • Certificate as Kubernetes secret
  • Jenkins jobs for certificate renewal

• Standardize
  • Service access to diploma
  • Auth tokens
  • Labels on secrets
  • Centralized CA config
Chancellor

• K8s controller

• Watches deployments/statefulsets/daemonsets

• Cert config in the annotation
  • CA to use
  • Common name for cert

• Retrieves Cert and creates k8s secret

```
annotations:
cert-request.diploma.vmware.com: '{"caSecretName":"ops-kafka-ca","baseCommonName":"ops-ingest",}'}
```
Interfaces

• CA interface

```
apiVersion: v1
class: Secret
data:  
  ca0.email: xxxxx  
  ca0.expirationTimeEpoch: xxx  
  ca0.name: xxxxxxxx  
  ca0.root: xxxxxxxx  
  ca0.secret: xxxxxxxx  
  ca0.token: xxxxxxxx
```

• Secret Interface

```
apiVersion: v1
kind: Secret
data:  
  chain.pem: xxxxx  
  key.pem: xxxxx  
  root0.pem: xxxxxxx
```
Migration

• Services need to use v2 CA

• Diploma V2
  • New Root CA
  • Not trusted by the services using v1 certs
Migration

• Avoid bad state
  • Services on different CAs that are not trusted by each other.
Expand - Contract

- Onboard to use certs from chancellor
- Services still use V1 CA
Expand - Contract

- Multiple CA config
Expand - Contract

- Move all services to use expanded CA config
• Update CA config in chancellor to prioritize v2 CA over v1 CA for private cert generation.
• Delete v1 CA config in chancellor.
Other features

• Cert renewal
  • Better control to renew

• Debugging
  • K8s events on create/renew of certs
  • Observability metrics
Other Solutions

• Cert-Manager
  • Vault integration
  • Multiple CA support

• Istio
  • Sidecar approach
  • Service account based identity
Thank you

Akshay Chitneni  
https://www.linkedin.com/in/akshayadatta/  
@akshayadatta

Tilottama Gaat  
https://www.linkedin.com/in/tgaat/