Certificates management

Intro

Certificates management in TDS allows you to easily enable automated SSL/TLS certicates deployment and renewals.

Project certificates view allows you to list and manage SSL/TLS certificates in your project.

In server settings view it allows you to configure certificate and DNS settings.

Enabling certificates creates /data/ssl folder with relevant files:

- server.key - private key
- server.crt - certificate signed by relevant CA (Let's Encrypt or in some cases TDS)
- ca-bundle.crt - chain of intermediate and root certificates that signed the certificate - more at CAcertificateschain
  - usually used by Apache
  - order mus be always:
    - intermediate
    - root
  - shall not contain other certificates
- fullchain.crt - chain of server cerfiticate, intermediate and root certificate - more at CAcertificateschain
  - usually used by Nginx
  - order mus be always:
    - server certificate
    - intermediate
    - root

You can optionally enable those certificate files when deploying server.

Self-managed PaaS applications (Gerrit, Jenkins, SonarQube) from TDS are automatically configured to use those certificate files.

You can also create hooks if you want to execute some commands after new certificate is deployed. Typically it is restart of some service. Also you can use hooks which can be automatically executed after each new certificate deploy - look for more in Certificatedeploymenthooks chapter.

Managing certificates via portal

Server certificates management

Generic server certificates feature description

- Usually it is sufficient to deploy server with certificate enabled.
- Certificate can be enable also later via server settings.
- In server settings you can also change which certificate you would like to have deployed on that particular server.

Server DNS and certificates settings view description

In server settings view it allows you to configure certificate and DNS settings.

There is multiple features available:

- certificates management
  - enabling >> signs new certificate or uses existing default certificate to deploy to agreed location on server automatically
  - disabling >> turns off automated certificate management, we do not touch the certificates on server after disabling as that might lead to unexpected issues. Certificate is not renewed after expiration.
- support for DNS alias for server and for certificate

        - if alias belongs to area domain we are able to sign certificate for it, otherwise we just sign certificate by TDS CA >> you can change it to your custom certificate
- support for wildcard subdomain DNS record for server and certificate for it
    - if subdomain is under FQDN managed by TDS, we are able to register wildcard DNS record >> otherwise we just store such setting into database without actually registering anything to DNS or without signing certificate as we would not be able allowed to do so anyway due to DNS and certificates signing protection

Project certificates management

Project certificates view allows you to list and manage SSL/TLS certificates in your project.

You can get there by opening particular portal project, then opening "Certificates" menu.

Every certificate shows its usage/assignment on servers and domains or subject alternative names (SANs) that it is applicable for.

We utilise Let's Encrypt CA for automated signing in most of the cases, however in some cases own certificates need to be deployed. You can add/import custom certificates here and use them on servers.

For that purpose you can go to your project >> Certificates and then click (+) button to add new custom certificate.

There you can usually see:

- Certificate
- Key
- CA certificates chain
    - this is chain of CA certificates starting with intermediate certificates and ending with root.
    - more at CAcertificateschain

CA certificates chain

This needs to be clean without extra new lines, spaces and it needs to concatenate CA certificates exactly in chain of trust:

- intermediate 1
    - certificate that signed our server certificate
- root
    - certificate which signs intermediate certificates

Sometimes there can be following differences:

- extra/secondary intermediate 2 if applicable - sometimes CA signs server certificate with secondary intermediate. That secondary intermediate is signed by primary intermediate certificate
- intermediate certificates completely missing - this is generally used in cases with self-signed CA when root certificate signs certificates.

How to make some CA certificates trusted

- Let's Encrypt - usually it is publicly trusted
    - However on some old systems you can see How to make new Lets Encrypt CA certificates trusted
- Self-signed CA
    - In principle you need to put CA certificates of your own certificate authority into relevant CA stores:
        - OS CA store
        - browser CA store

Certificates deployment config

Generally reload of httpd/apache2 is called during each certificates update. However in some cases we must various extra operations.

For that purpose hooks folder has been introduced with following default value:

CERT_HOOKS_LOCATION=/data/ssl/hooks

As soon as there is anything executable present in hooks folder, it is automatically executed. Remember to handle also httpd/apache2 restart yourself as regular certificate update script skips apache restarts in cases when hooks are used.

If you like you can override hooks path by providing CERT_HOOKS_LOCATION variable in /data/configs/tdscertdeploy.conf config file.

Suppressing certificates deployment

Create following file which will make sure your certificates will not be touched:

```
mkdir -p /data/configs
touch /data/configs/tdscertdeploy.conf
```

Content of file:

```
CERT_AUTO_DEPLOY=false
```

To get certificates deployed automatically again during next renewal periods, just remove that file and automated certificate deployment will work.

Certificate deployment hooks

Hooks shall be bash scripts made executable and placed in folder /data/ssl/hooks folder. It will be automatically executed every time when new certificate is deployed.

You can place as many scripts into hooks folder as you like, they are executed in alphabetical order.

Nginx certificates hook example

For Nginx web server it is recommended to have server certificate and intermediate certificates bundled in file configured by "ssl_certificate" directive: http://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_certificate

Example of setting of correct certificate path in nginx files:

```
# Example of configuring recommended path to complete chain
grep 'ssl_certificate /' /etc/nginx/sites-available/*
sed -i 's#ssl_certificate /.*#ssl_certificate /data/ssl/fullchain.crt;#' /etc/nginx/sites-available/*
sed -i 's#ssl_certificate_key /.*#ssl_certificate_key /data/ssl/server.key;#' /etc/nginx/sites-available/*
grep 'ssl_certificate /' /etc/nginx/sites-available/*
```

This is recommended setup verified by users:

```
# Preparing hook:
mkdir -p /data/ssl/hooks/
touch /data/ssl/hooks/nginx.sh
chmod +x /data/ssl/hooks/nginx.sh
echo '#!/bin/sh
cat /data/ssl/server.crt > /data/ssl/fullchain.crt
cat /data/ssl/ca-bundle.crt >> /data/ssl/fullchain.crt
systemctl restart nginx' > /data/ssl/hooks/nginx.sh
cat /data/ssl/hooks/nginx.sh

 # Finally executing the hook to verify that it works
/data/ssl/hooks/nginx.sh
```