# AAF Certificate Management for Dummies

The steps that were followed to generate certificates to be used to deploy the CMSO Spring Boot services in OOM with HTTPS enabled in Windriver Lab. This assumes you already have access to the Windriver Lab.

## Updating the oof.onap Certificate in AAF

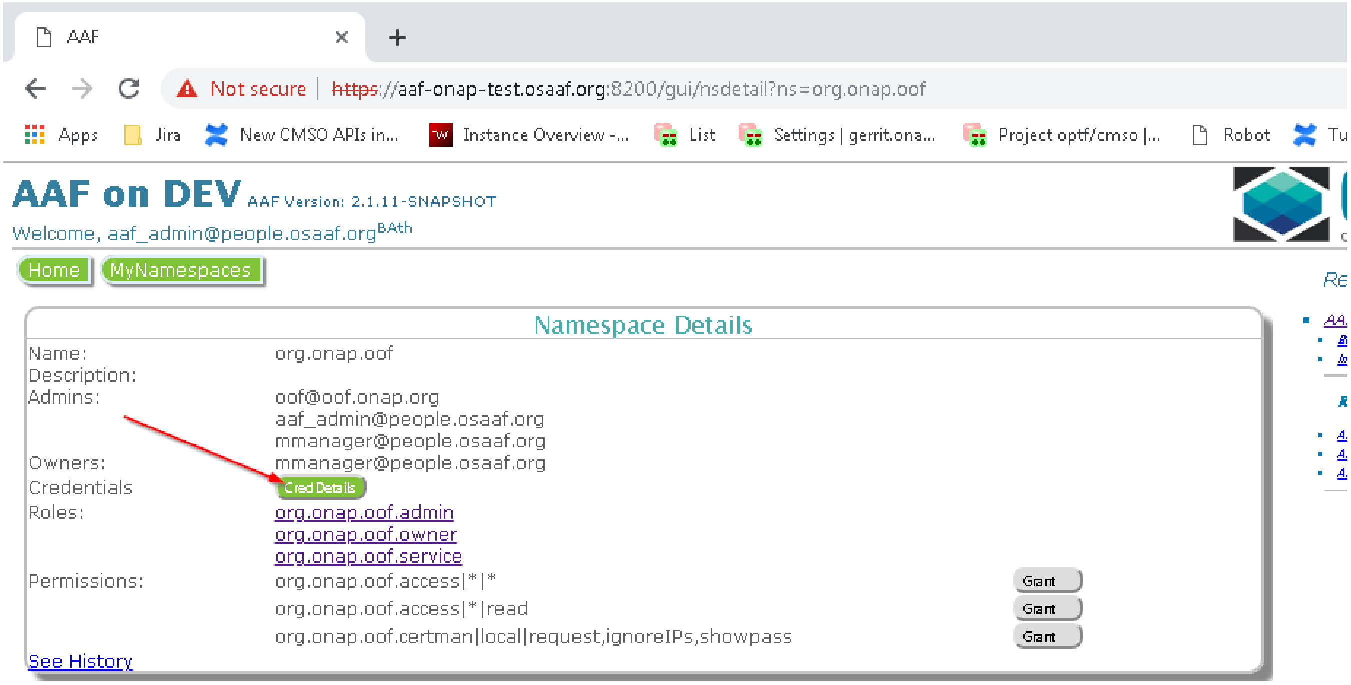
In order to access AAF https://aaf-onap-test.osaaf.org:8200/gui/home you will need to update your /etc/hosts file (C:\Windows\System32\drivers\etc\hosts on Windows)

```
10.12.5.145 aaf-onap-test.osaaf.org
```

Note that the IP can change. Not sure how to find out where it is running when it does. Found it here https://docs.onap.org/en/latest/submodules/aaf/authz.git/docs/sections/configuration/AAF_4.1_config.html, but this could change too 😉

### Navigating to the AAF Certificate

From here: https://aaf-onap-test.osaaf.org:8200/gui/nsdetail?ns=org.onap.oof

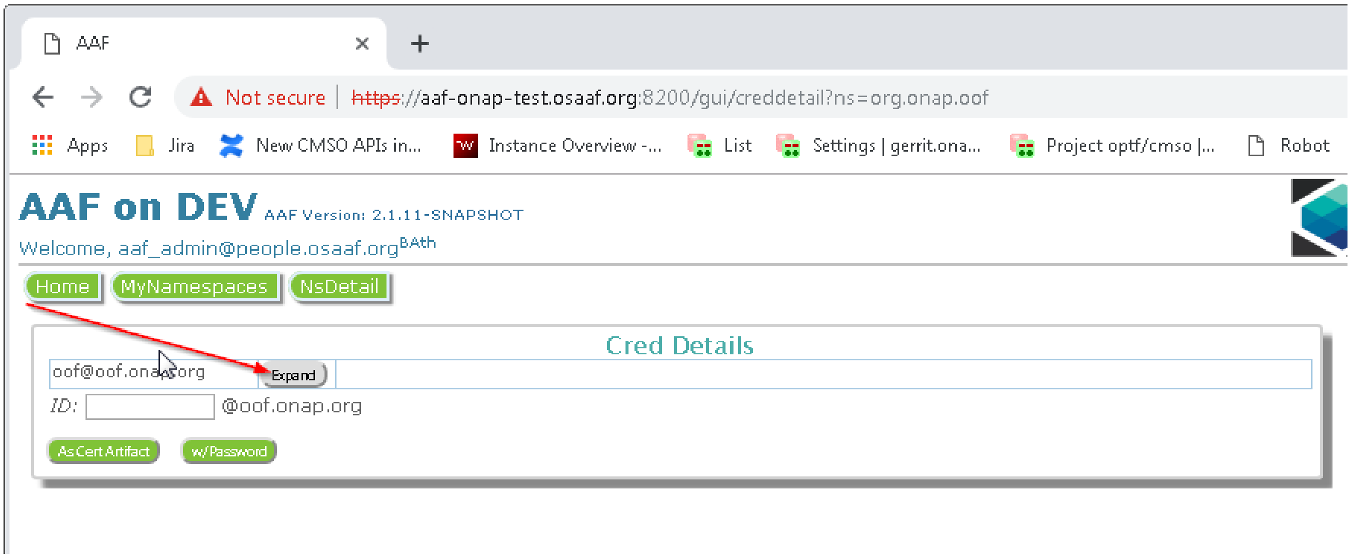


AAF on DEV AAF Version: 2.1.11-SNAPSHOT

Welcome, aaf_admin@people.osaaf.org BAth

Namespace Details

Name: org.onap.oof
Description:
Admins: oof@oof.onap.org
aaf_admin@people.osaaf.org
mmanager@people.osaaf.org
Owners: mmanager@people.osaaf.org
Credentials Cred Details
Roles: org.onap.oof.admin
org.onap.oof.owner
org.onap.oof.service
Permissions: org.onap.oof.access|*|*
org.onap.oof.access|*|read
org.onap.oof.certman|local|request,ignoreIPs,showpass



~~https~~://aaf-onap-test.osaaf.org:8200/gui/creddetail?ns=org.onap.oof

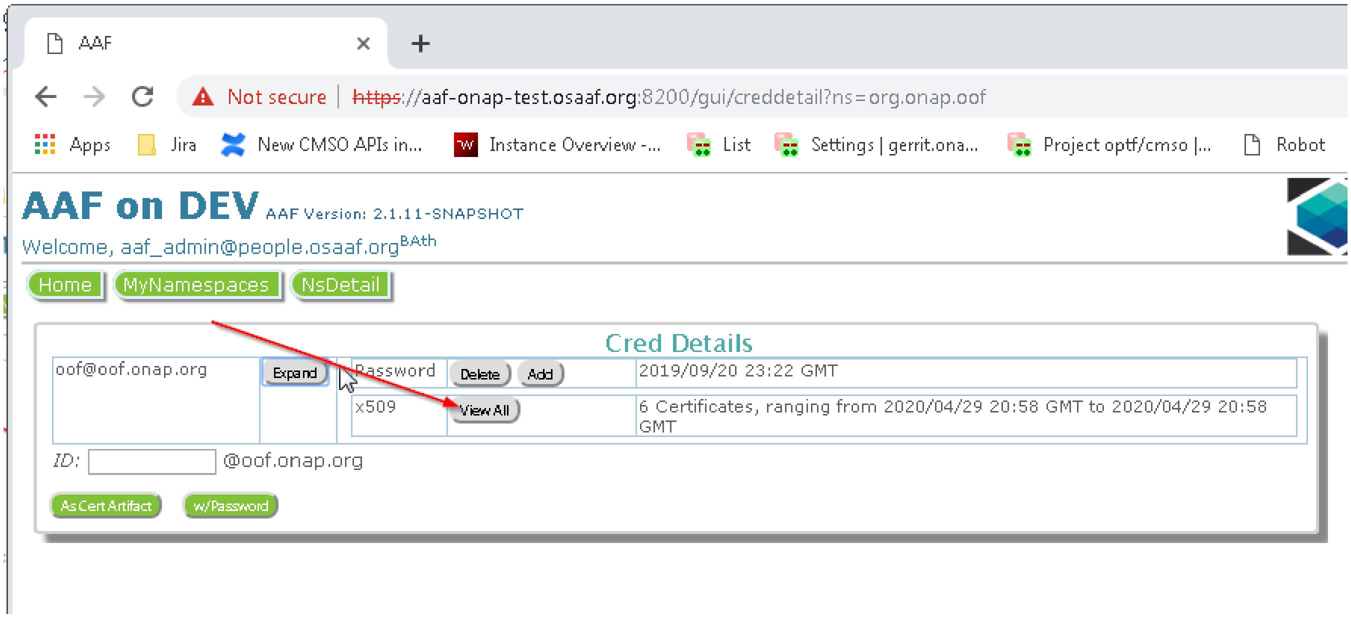
AAF on DEV AAF Version: 2.1.11-SNAPSHOT

Welcome, aaf_admin@people.osaaf.org BAth

Cred Details

oof@oof.onap.org

*ID:* @oof.onap.org



~~https~~://aaf-onap-test.osaaf.org:8200/gui/creddetail?ns=org.onap.oof

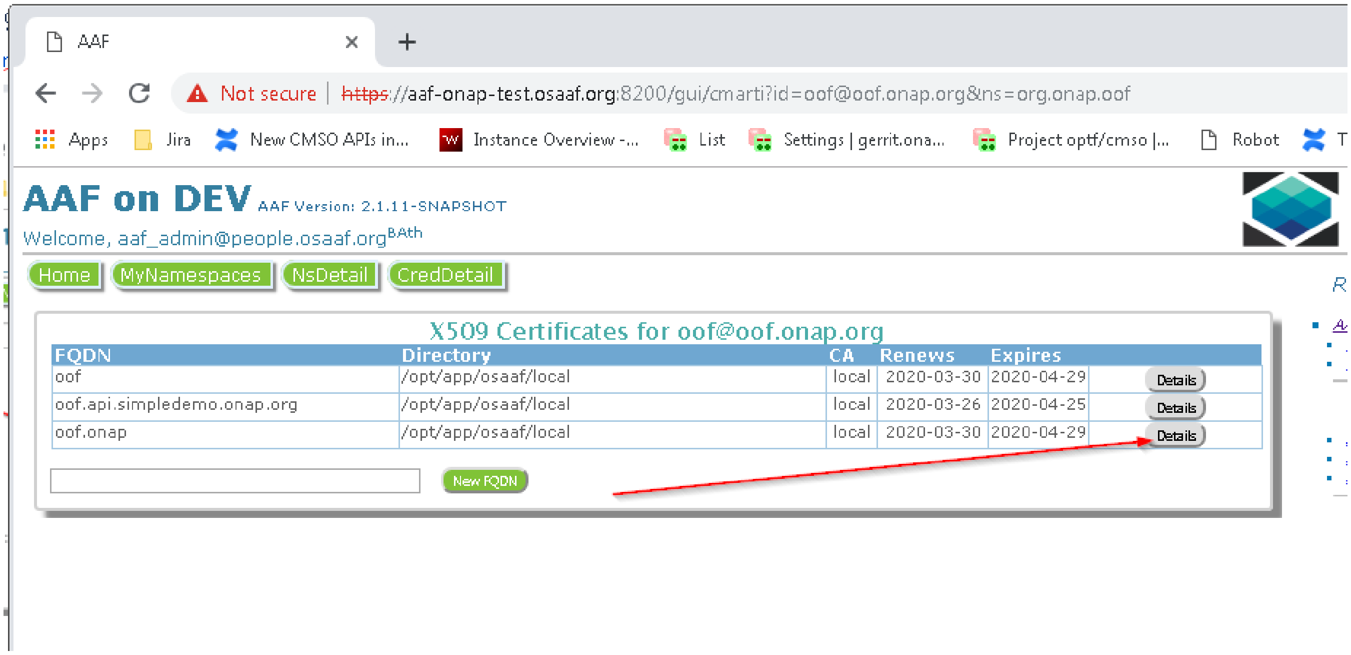
AAF on DEV AAF Version: 2.1.11-SNAPSHOT

Welcome, aaf_admin@people.osaaf.org BAth

Cred Details

| | | | | |
|---|---|---|---|---|
| oof@oof.onap.org | Expand | Password | Delete Add | 2019/09/20 23:22 GMT |
| | | x509 | View All | 6 Certificates, ranging from 2020/04/29 20:58 GMT to 2020/04/29 20:58 GMT |

*ID:* @oof.onap.org

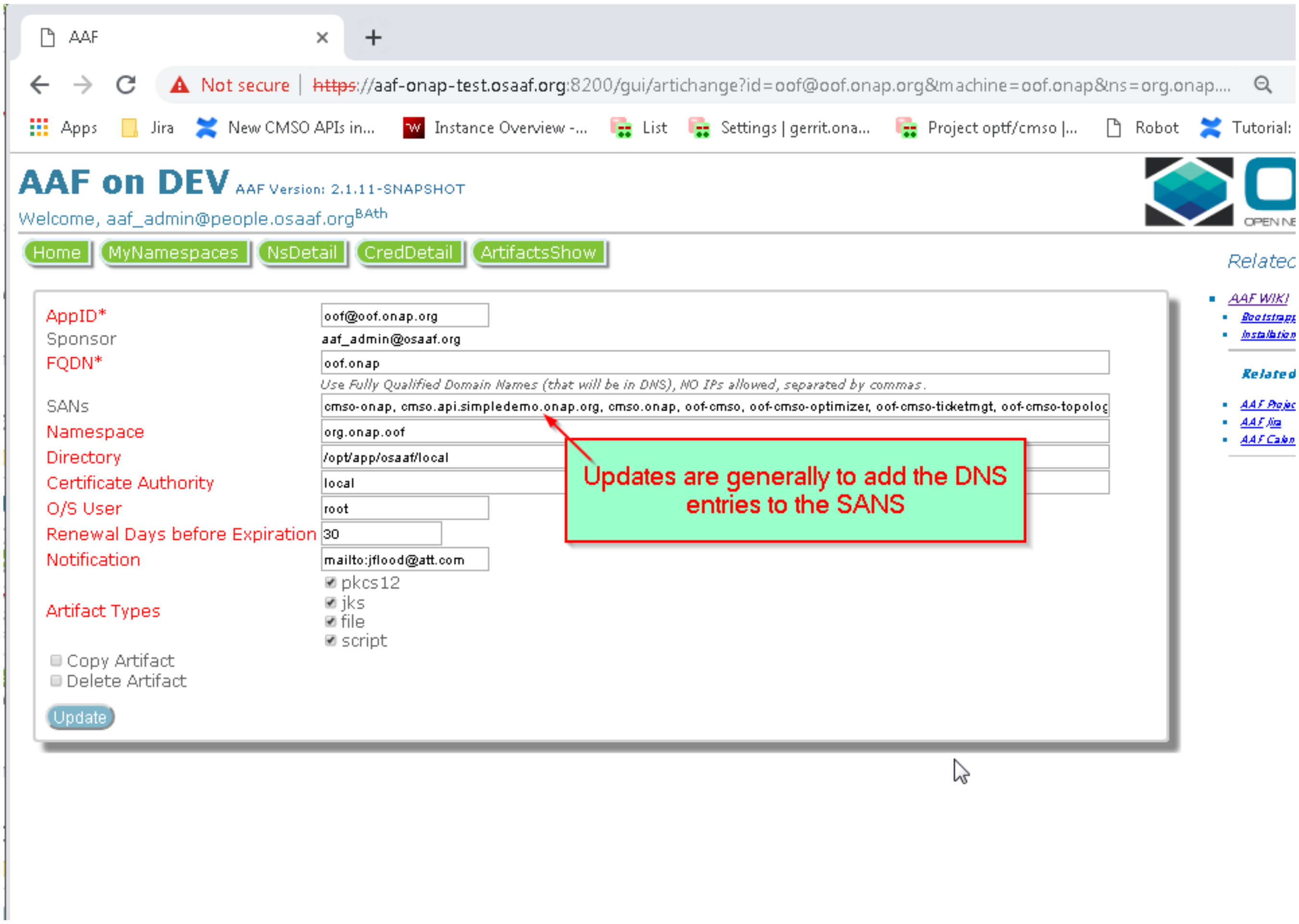


AAF on DEV AAF Version: 2.1.11-SNAPSHOT

Welcome, aaf_admin@people.osaaf.org BAth

X509 Certificates for oof@oof.onap.org

| FQDN | Directory | CA | Renews | Expires | |
|---|---|---|---|---|---|
| oof | /opt/app/osaaf/local | local | 2020-03-30 | 2020-04-29 | Details |
| oof.api.simpledemo.onap.org | /opt/app/osaaf/local | local | 2020-03-26 | 2020-04-25 | Details |
| oof.onap | /opt/app/osaaf/local | local | 2020-03-30 | 2020-04-29 | Details |



AAF on DEV AAF Version: 2.1.11-SNAPSHOT

Welcome, aaf_admin@people.osaaf.org BAth

| | |
|---|---|
| AppID* | oof@oof.onap.org |
| Sponsor | aaf_admin@osaaf.org |
| FQDN* | oof.onap<br>*Use Fully Qualified Domain Names (that will be in DNS), NO IPs allowed, separated by commas.* |
| SANs | cmso-onap, cmso.api.simpledemo.onap.org, cmso.onap, oof-cmso, oof-cmso-optimizer, oof-cmso-ticketmgt, oof-cmso-topolog |
| Namespace | org.onap.oof |
| Directory | /opt/app/osaaf/local |
| Certificate Authority | local |
| O/S User | root |
| Renewal Days before Expiration | 30 |
| Notification | mailto:jflood@att.com |
| Artifact Types | ☑ pkcs12<br>☑ jks<br>☑ file<br>☑ script |

☐ Copy Artifact
☐ Delete Artifact

Updates are generally to add the DNS entries to the SANS

Or, if you want the oof.onap certificate, just click on https://aaf-onap-test.osaaf.org:8200/gui/artichange?id=oof@oof.onap.org&machine=oof.onap&ns=org.onap.oof

# Downloading the AAF certificate Artifacts

This must be done from a host that can run docker and has it's etc hosts updated as above:

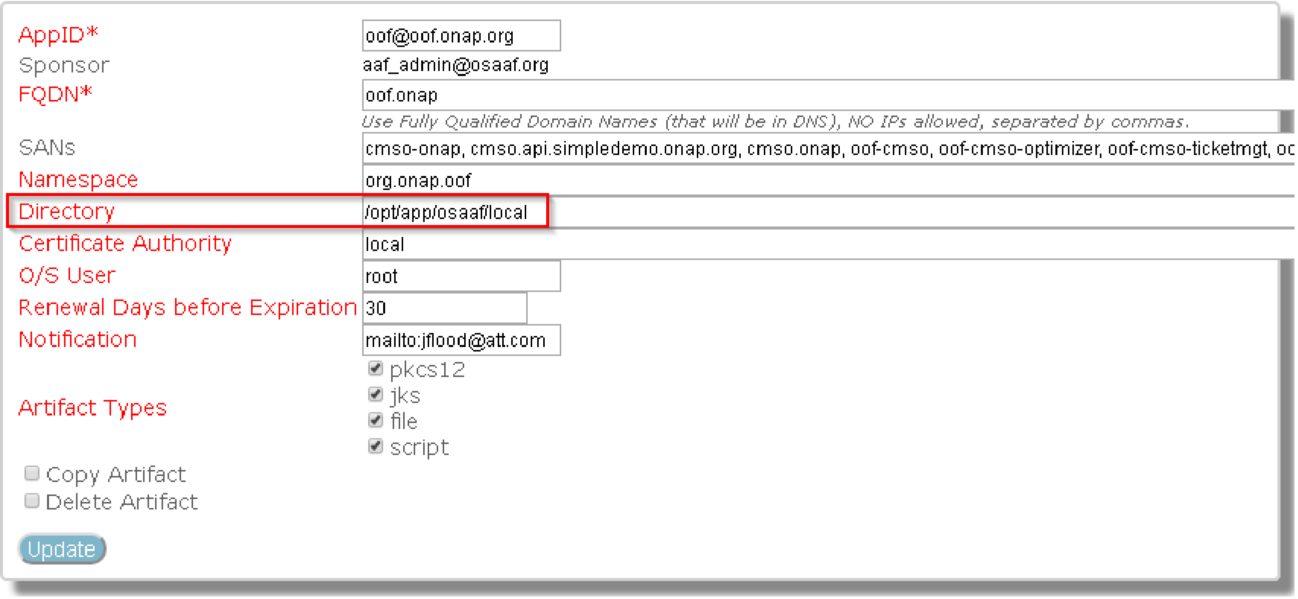
```
10.12.5.145 aaf-onap-test.osaaf.org
```

This is the Reader's Digest version of https://docs.onap.org/en/latest/submodules/aaf/authz.git/docs/sections/configuration/AAF_4.1_config.html

- Download https://gerrit.onap.org/r/gitweb?p=aaf/authz.git;a=blob_plain;f=auth/docker/agent.sh;hb=HEAD
- Rename file to **agent.sh**

Running **agent.sh** will prompt for many arguments which will be stored in **./aaf.props** file. Subsequent executions of **agent.sh** will pull the values from **aaf.props** so if you need to change a value, it should be done in **aaf.props** (or delete it and start over)

**agent.sh** will run a docker image which will download all the artifacts to the **Directory** /opt/app/osaaf/local in the docker image.



| | |
|---|---|
| AppID* | oof@oof.onap.org |
| Sponsor | aaf_admin@osaaf.org |
| FQDN* | oof.onap<br>*Use Fully Qualified Domain Names (that will be in DNS), NO IPs allowed, separated by commas.* |
| SANs | cmso-onap, cmso.api.simpledemo.onap.org, cmso.onap, oof-cmso, oof-cmso-optimizer, oof-cmso-ticketmgt, oc |
| Namespace | org.onap.oof |
| Directory | /opt/app/osaaf/local |
| Certificate Authority | local |
| O/S User | root |
| Renewal Days before Expiration | 30 |
| Notification | mailto:jflood@att.com |
| Artifact Types | ☑ pkcs12<br>☑ jks<br>☑ file<br>☑ script |

□ Copy Artifact
□ Delete Artifact

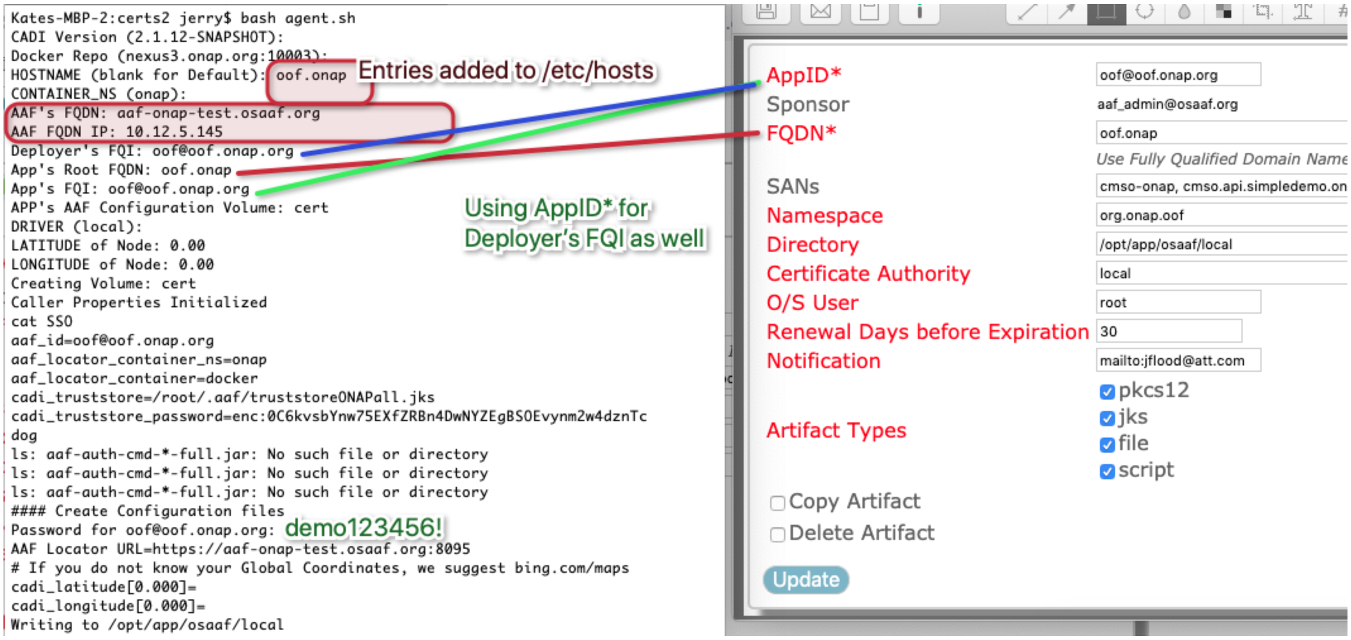
The Application FQDN is the **FQDN *** oof.onap so we will add our Windriver lab VPN IP that to our /etc/hosts file as well

```
10.12.5.145 aaf-onap-test.osaaf.org
10.12.25.177 oof.onap
```

On Unix host, run ifconfig while connected to the VPN

```
ifconfig -a|grep 10.12
inet 10.12.25.177 --> 10.12.25.178 netmask 0xffffffff
```

These files will not be accessible by default when the script is done because it is going to a docker volume that is in the VOLUME attribute in **aaf.props**. For some reason, the script does not allow the VOLUME to be a local folder. The docker volume will automatically be created, in the case below docker volume **cert** (volume can be any name) created with the local driver.



```
Kates-MBP-2:certs2 jerry$ bash agent.sh
CADI Version (2.1.12-SNAPSHOT):
Docker Repo (nexus3.onap.org:10003):
HOSTNAME (blank for Default): oof.onap
CONTAINER_NS (onap):
AAF's FQDN: aaf-onap-test.osaaf.org
AAF FQDN IP: 10.12.5.145
Deployer's FQI: oof@oof.onap.org
App's Root FQDN: oof.onap
App's FQI: oof@oof.onap.org
APP's AAF Configuration Volume: cert
DRIVER (local):
LATITUDE of Node: 0.00
LONGITUDE of Node: 0.00
Creating Volume: cert
Caller Properties Initialized
cat SSO
aaf_id=oof@oof.onap.org
aaf_locator_container_ns=onap
aaf_locator_container=docker
cadi_truststore=/root/.aaf/truststoreONAPall.jks
cadi_truststore_password=enc:0C6kvsbYnw75EXfZRBn4DwNYZEgBSOEvynm2w4dznTc
dog
ls: aaf-auth-cmd-*-full.jar: No such file or directory
ls: aaf-auth-cmd-*-full.jar: No such file or directory
ls: aaf-auth-cmd-*-full.jar: No such file or directory
#### Create Configuration files
Password for oof@oof.onap.org:
AAF Locator URL=https://aaf-onap-test.osaaf.org:8095
# If you do not know your Global Coordinates, we suggest bing.com/maps
cadi_latitude[0.000]=
cadi_longitude[0.000]=
Writing to /opt/app/osaaf/local
```

AppID* oof@oof.onap.org

Sponsor aaf_admin@osaaf.org

FQDN* oof.onap

Use Fully Qualified Domain Name

SANs cmso-onap, cmso.api.simpledemo.on

Namespace org.onap.oof

Directory /opt/app/osaaf/local

Certificate Authority local

O/S User root

Renewal Days before Expiration 30

Notification mailto:jflood@att.com

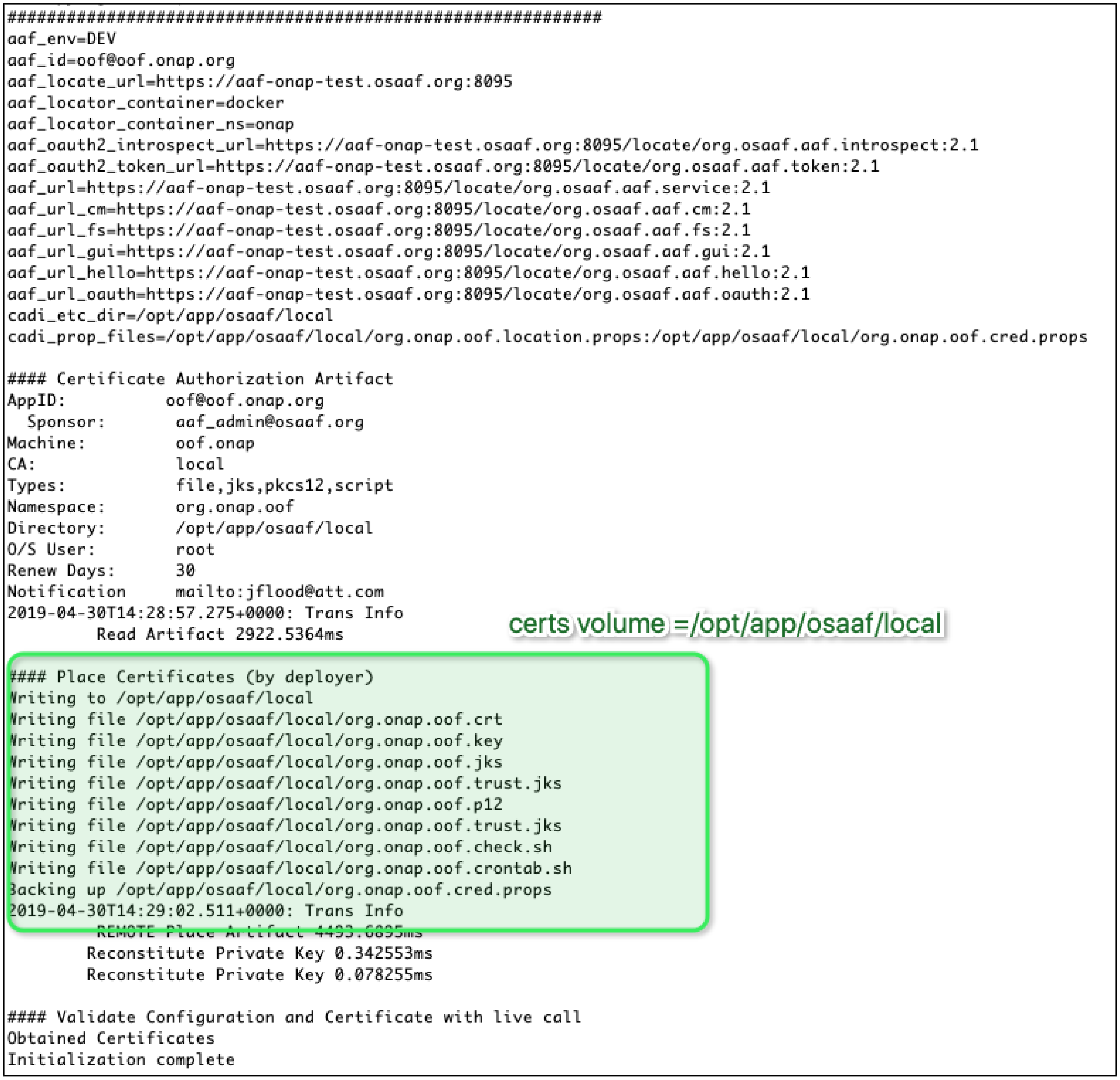
Artifact Types ☑ pkcs12 ☑ jks ☑ file ☑ script

□ Copy Artifact

□ Delete Artifact

Entries added to /etc/hosts
Using AppID* for Deployer's FQI as well
demo123456!
Update

WIth luck this will have generated all of the artifacts in the docker **cert** volume:



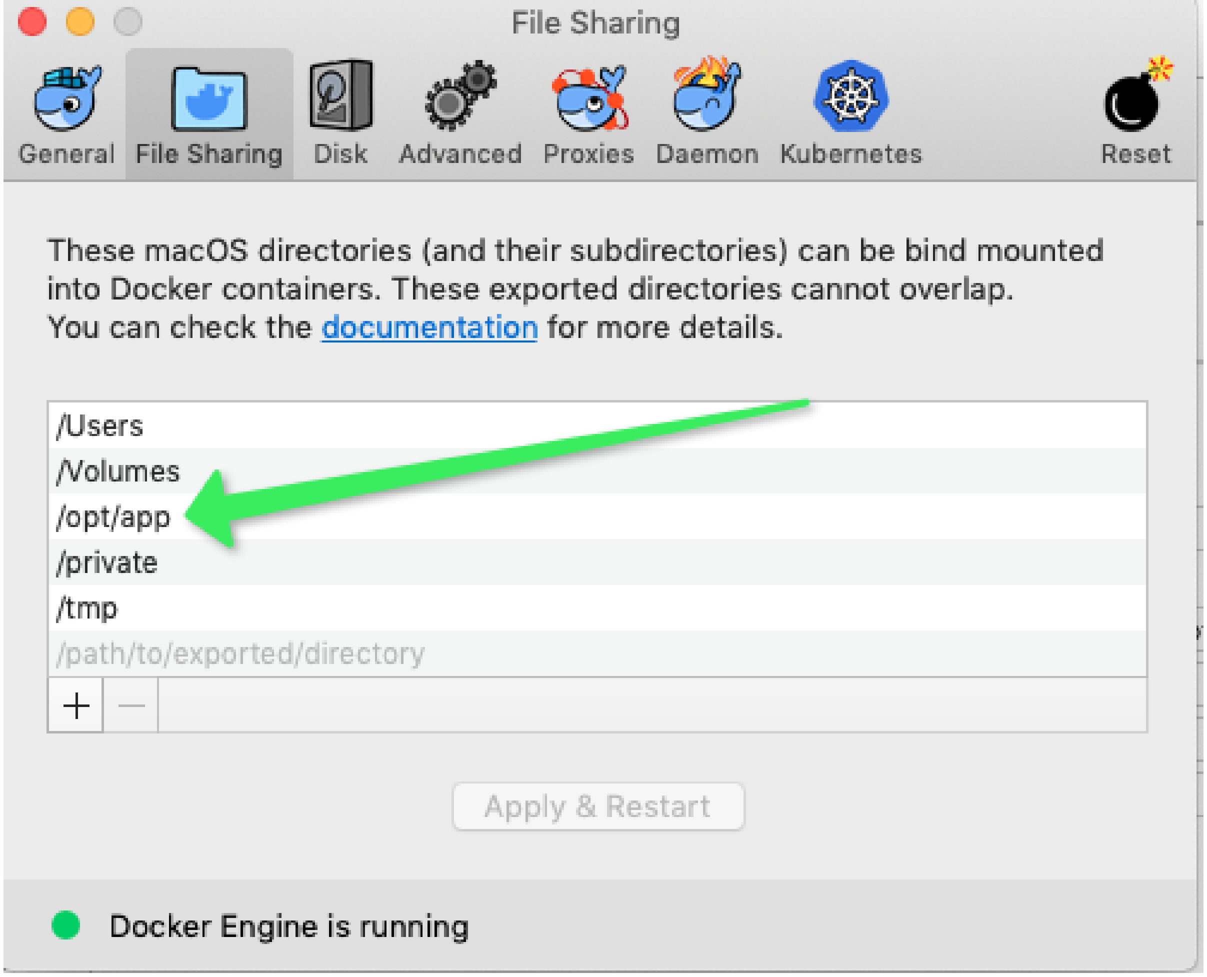
```
##############################################################
aaf_env=DEV
aaf_id=oof@oof.onap.org
aaf_locate_url=https://aaf-onap-test.osaaf.org:8095
aaf_locator_container=docker
aaf_locator_container_ns=onap
aaf_oauth2_introspect_url=https://aaf-onap-test.osaaf.org:8095/locate/org.osaaf.aaf.introspect:2.1
aaf_oauth2_token_url=https://aaf-onap-test.osaaf.org:8095/locate/org.osaaf.aaf.token:2.1
aaf_url=https://aaf-onap-test.osaaf.org:8095/locate/org.osaaf.aaf.service:2.1
aaf_url_cm=https://aaf-onap-test.osaaf.org:8095/locate/org.osaaf.aaf.cm:2.1
aaf_url_fs=https://aaf-onap-test.osaaf.org:8095/locate/org.osaaf.aaf.fs:2.1
aaf_url_gui=https://aaf-onap-test.osaaf.org:8095/locate/org.osaaf.aaf.gui:2.1
aaf_url_hello=https://aaf-onap-test.osaaf.org:8095/locate/org.osaaf.aaf.hello:2.1
aaf_url_oauth=https://aaf-onap-test.osaaf.org:8095/locate/org.osaaf.aaf.oauth:2.1
cadi_etc_dir=/opt/app/osaaf/local
cadi_prop_files=/opt/app/osaaf/local/org.onap.oof.location.props:/opt/app/osaaf/local/org.onap.oof.cred.props

#### Certificate Authorization Artifact
AppID:          oof@oof.onap.org
  Sponsor:       aaf_admin@osaaf.org
Machine:         oof.onap
CA:              local
Types:           file,jks,pkcs12,script
Namespace:       org.onap.oof
Directory:       /opt/app/osaaf/local
O/S User:        root
Renew Days:      30
Notification     mailto:jflood@att.com
2019-04-30T14:28:57.275+0000: Trans Info
         Read Artifact 2922.5364ms

#### Place Certificates (by deployer)
Writing to /opt/app/osaaf/local
Writing file /opt/app/osaaf/local/org.onap.oof.crt
Writing file /opt/app/osaaf/local/org.onap.oof.key
Writing file /opt/app/osaaf/local/org.onap.oof.jks
Writing file /opt/app/osaaf/local/org.onap.oof.trust.jks
Writing file /opt/app/osaaf/local/org.onap.oof.p12
Writing file /opt/app/osaaf/local/org.onap.oof.trust.jks
Writing file /opt/app/osaaf/local/org.onap.oof.check.sh
Writing file /opt/app/osaaf/local/org.onap.oof.crontab.sh
Backing up /opt/app/osaaf/local/org.onap.oof.cred.props
2019-04-30T14:29:02.511+0000: Trans Info
         REMOTE Place Artifact 4493.6895ms
        Reconstitute Private Key 0.342553ms
        Reconstitute Private Key 0.078255ms

#### Validate Configuration and Certificate with live call
Obtained Certificates
Initialization complete
```

certs volume =/opt/app/osaaf/local

## Retrieving the artifacts from the docker volume

Note that when we retrieve the artifacts, the various passwords will be encrypted and can only be unencrypted by cadi. The cadi **showpass** command expects the artifacts to be in **/opt/app/osaaf/local** folder, so the following command can be used to put the files from the **cert** docker volume into your local **/opt/app/osaaf/local** folder.

**Important note on the Mac, had to update the docker preferences to include /opt/app as a mountable folder.**



These macOS directories (and their subdirectories) can be bind mounted into Docker containers. These exported directories cannot overlap. You can check the documentation for more details.

/Users
/Volumes
/opt/app
/private
/tmp
/path/to/exported/directory

Docker Engine is running

```
sudo docker run -v /opt/app/osaaf:/tmp/osaaf -v cert:/opt/app/osaaf/local ubuntu cp -rf /opt/app/osaaf/local/local /tmp/osaaf
```

In essence the above runs a docker container (ubuntu) with the docker **cert** volume mounted and the local **/opt/app/osaaf/local** folder mounted as a different volume so we can copy the artifacts from the docker volume to our local host.

# Retrieving the artifact passwords

The agent.sh can be used to decrypt  the passwords in the **org.onap.oof.cred.props** file.

There are several things to address.

1. Copy the truststoreONAPall.jks to the agent.sh folder
2. Make sure the CADI jar agent.sh is in the local folder. You will get this error: For local use, you need to have 'aaf-cadi-aaf-2.1.12-SNAPSHOT-full.jar'
3. The AAF account you are using needs to have **showpass** privieges in AAF, so we added those privileges to the **admin** role in the namespace

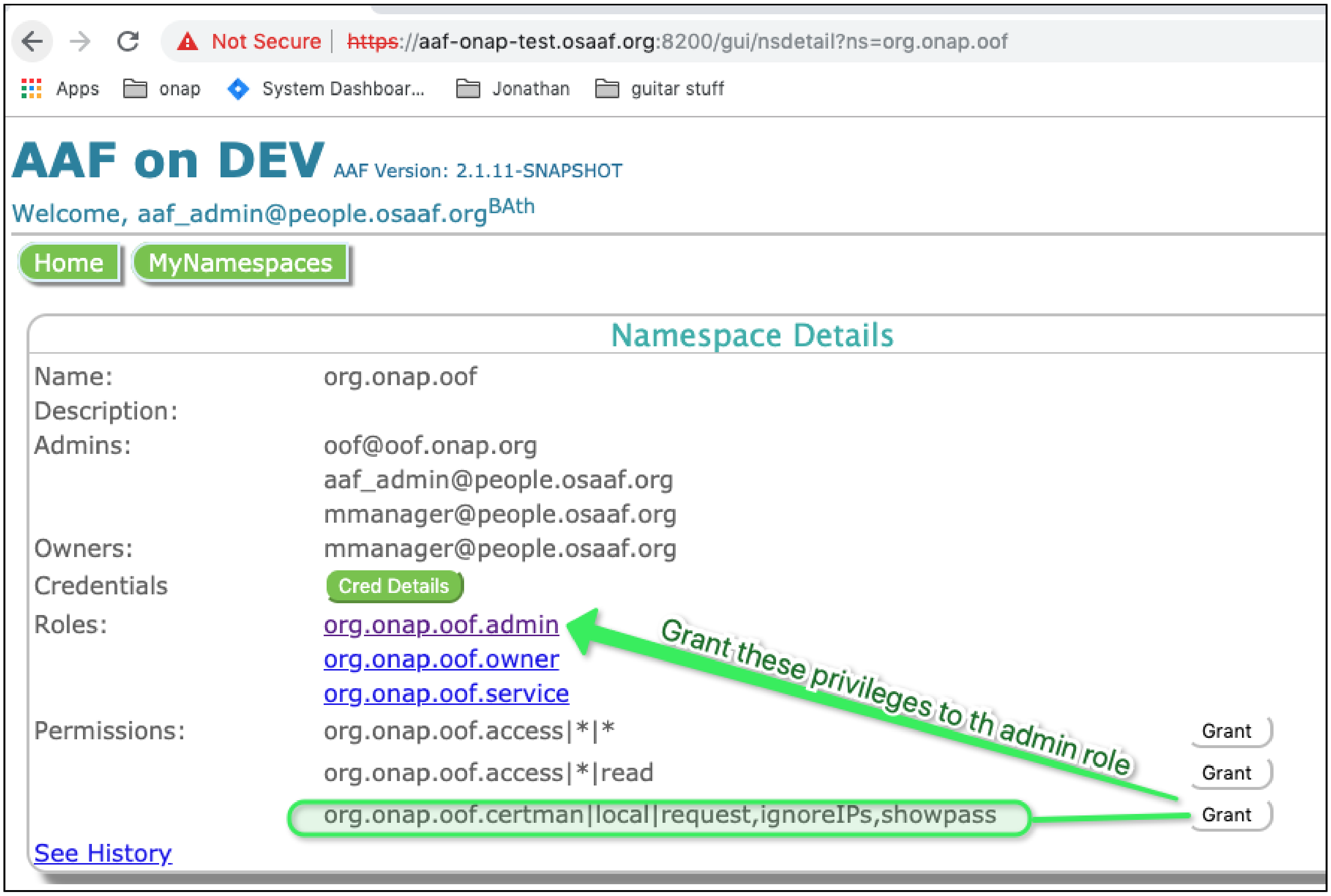
For 1:

```
cp /opt/app/osaaf/local/truststoreONAPall.jks ./
```

For 2:

I had a local version that was different than the version in agent.sh so I copied the jar I had to the **agent.sh** folder and updated VERSION in **aaf.props**

For 3:



AAF on DEV AAF Version: 2.1.11-SNAPSHOT

Welcome, aaf_admin@people.osaaf.org BAth

Namespace Details

| | |
|---|---|
| Name: | org.onap.oof |
| Description: | |
| Admins: | oof@oof.onap.org<br>aaf_admin@people.osaaf.org<br>mmanager@people.osaaf.org |
| Owners: | mmanager@people.osaaf.org |
| Credentials | Cred Details |
| Roles: | org.onap.oof.admin<br>org.onap.oof.owner<br>org.onap.oof.service |
| Permissions: | org.onap.oof.access\|*\|*<br>org.onap.oof.access\|*\|read<br>org.onap.oof.certman\|local\|request,ignoreIPs,showpass |

See History

Grant these privileges to th admin role

So finally you can run

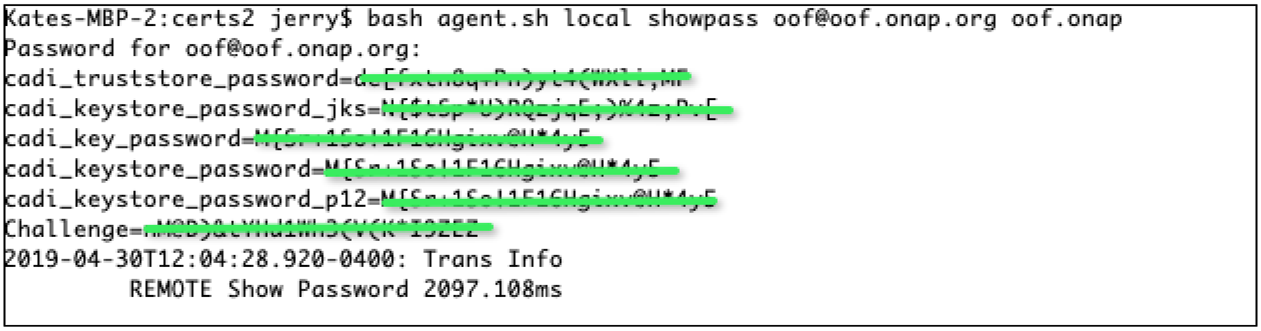
```
bash agent.sh local showpass oof@oof.onap.org oof.onap
```

This will prompt for oof@oof.onap.org password which is the default demo123456!

Note that if you fat finger the password you get a goofy error

org.onap.aaf.cadi.CadiException: org.onap.aaf.cadi.LocatorException: No Entries found for 'https://aaf-onap-test.osaaf.org:8095/locate/AAF_NS.cm:2.1'

which can send you on a wild goose chase.



```
Kates-MBP-2:certs2 jerry$ bash agent.sh local showpass oof@oof.onap.org oof.onap
Password for oof@oof.onap.org:
cadi_truststore_password=[redacted]
cadi_keystore_password_jks=[redacted]
cadi_key_password=[redacted]
cadi_keystore_password=[redacted]
cadi_keystore_password_p12=[redacted]
Challenge=[redacted]
2019-04-30T12:04:28.920-0400: Trans Info
         REMOTE Show Password 2097.108ms
```

Not sure why I bothered to obscure the passwords 😉

# Using certificates with Spring Boot Application

1. Copy these 2 artifacts to a folder accessible to your application
   a. truststoreONAPall.jks (for outgoing HTTPS requests)
   b. org.onap.<app>.jks (i.e. org.onap.oof.jks) (for incoming HTTPS requests)
2. Add the following to the JVM args

a. -Dserver.ssl.key-store=<*folder*>/org.onap.oof.jks
b. -Dserver.ssl.key-store-password=<*cadi_keystore_password_jks*>
c. -Djavax.net.ssl.trustStore=<*folder*>/truststoreONAPall.jks