Using a 3rd Party Certificate Authority with OpsBridge

Chakraborty, Abhijit

MICRO FOCUS

CONTENTS

1 INTRODUCTION

Micro Focus Operations Bridge Suite (OpsBridge), and specifically the Operations Bridge Manager (OBM) and Operations Agent (OA), provide an inbuilt, self-signed certificate based, secure communication framework. This secure communication framework provides strong security, adhering to industry standards. However, for certain organizations, it is mandatory to use a specific recommended Certificate Authority (CA). For such occasions, it is possible to use a 3rd party CA, in place of the inbuilt certificate management framework, following documented steps. This white paper provides all the necessary steps to use a 3rd party CA for a secure communication environment for OBM and OA. This feature is available from OBM 2018.05 release.

2 WORKING WITH 3RD PARTY CA

This section delves in to the steps to follow, for using a 3rd party CA with OBM, for the purpose of establishing a secure communication framework in a distributed OBM, OA deployment scenario.

The steps provided include, the steps to create certificates on the OBM Server (additional steps required for a distributed OBM setup), and steps to create certificates on the OA system. The subsequent sections provide details of all the steps to should be followed.

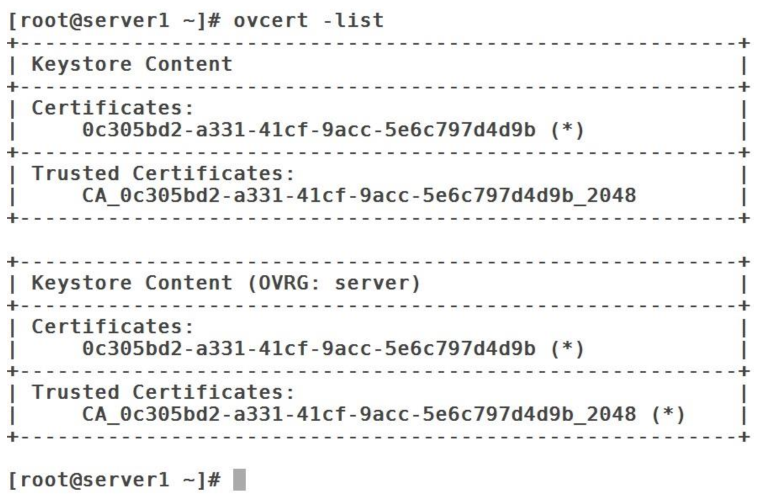
This document assumes fresh installation of OBM and OA systems as the starting point, which are not configured yet.

2.1 CREATING CERTIFICATES ON THE OBM SERVER

This section explains the steps to install certificates from a 3rd party CA on a standalone OBM system.

1. Install OBM using the installation wizard, but do not execute the configuration step.

At this point all the core id's (server OVRG groups and the nodes), will be same (as shown in Figure 1).



```
[root@server1 ~]# ovcert -list
+---------------------------------------------------------------+
| Keystore Content                                              |
+---------------------------------------------------------------+
| Certificates:                                                 |
|     0c305bd2-a331-41cf-9acc-5e6c797d4d9b (*)                  |
+---------------------------------------------------------------+
| Trusted Certificates:                                         |
|     CA_0c305bd2-a331-41cf-9acc-5e6c797d4d9b_2048              |
+---------------------------------------------------------------+

+---------------------------------------------------------------+
| Keystore Content (OVRG: server)                               |
+---------------------------------------------------------------+
| Certificates:                                                 |
|     0c305bd2-a331-41cf-9acc-5e6c797d4d9b (*)                  |
+---------------------------------------------------------------+
| Trusted Certificates:                                         |
|     CA_0c305bd2-a331-41cf-9acc-5e6c797d4d9b_2048 (*)          |
+---------------------------------------------------------------+

[root@server1 ~]#
```

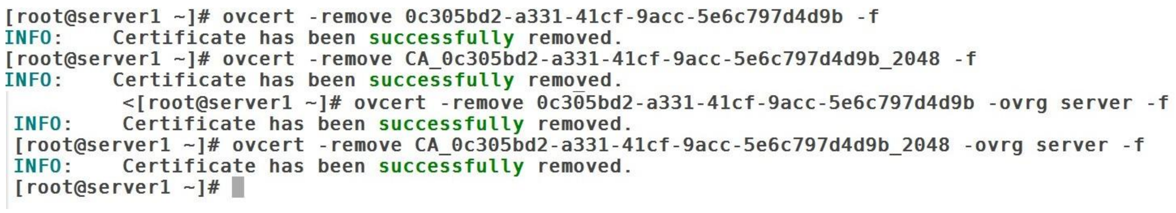
Figure 1

2. Now, remove all the certificates that are present.

    - You can achieve this by running the *"ovcert –remove <certificate id>"* command.
    - You should run the command for both the node and trusted certificate.
    - To remove the certificates in the resource group OVRG: Server, you must suffix the remove command with "-ovrg server".

        - *"ovcert –remove <certificate id> -ovrg server".*

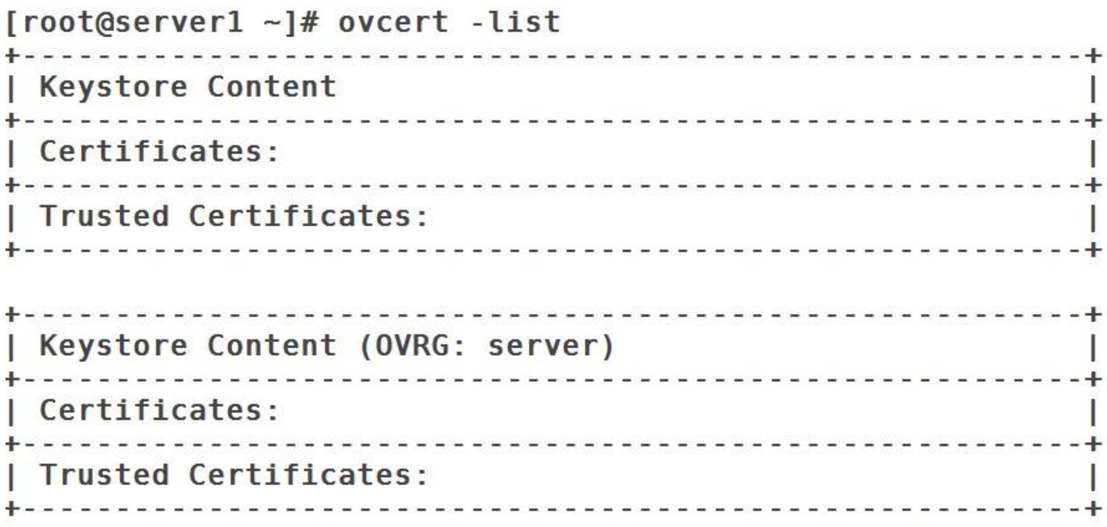
    For example, refer to Figure 2.



```
[root@server1 ~]# ovcert -remove 0c305bd2-a331-41cf-9acc-5e6c797d4d9b -f
INFO:    Certificate has been successfully removed.
[root@server1 ~]# ovcert -remove CA_0c305bd2-a331-41cf-9acc-5e6c797d4d9b_2048 -f
INFO:    Certificate has been successfully removed.
         <[root@server1 ~]# ovcert -remove 0c305bd2-a331-41cf-9acc-5e6c797d4d9b -ovrg server -f
 INFO:    Certificate has been successfully removed.
 [root@server1 ~]# ovcert -remove CA_0c305bd2-a331-41cf-9acc-5e6c797d4d9b_2048 -ovrg server -f
 INFO:    Certificate has been successfully removed.
 [root@server1 ~]#
```

Figure 2

3. The Keystore Content should be empty after removing the certificates. This can be verified using the "*ovcert –list*" command. The output of "*ovcert –list*" command should look as depicted in Figure 3 below.

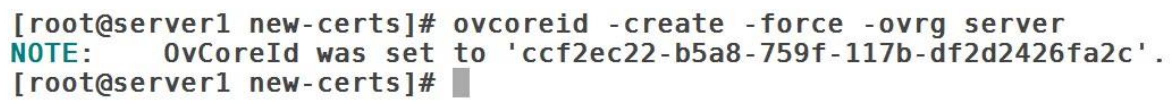


```
[root@server1 ~]# ovcert -list
+---------------------------------------------------------+
| Keystore Content                                        |
+---------------------------------------------------------+
| Certificates:                                           |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
+---------------------------------------------------------+

+---------------------------------------------------------+
| Keystore Content (OVRG: server)                         |
+---------------------------------------------------------+
| Certificates:                                           |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
+---------------------------------------------------------+
```

Figure 3

4. Create the node certificates using the preferred 3rd party CA. It is important to ensure that the server resource group "OVRG" has a different core identifier. A different core identifier can be enforced using the "*ovcoreid –create –force –ovrg server"* command, as shown in Figure 4.



```
[root@server1 new-certs]# ovcoreid -create -force -ovrg server
NOTE:    OvCoreId was set to 'ccf2ec22-b5a8-759f-117b-df2d2426fa2c'.
[root@server1 new-certs]#
```

Figure 4

5. Customer can now issue certificates, specifying the agent core identifier as the common name (CN) in the certificate signing request for the certificates generated. Create the node certificate using the agent core identifier, which is the output of the *"ovcoreid"* command. The server certificate should be created with the core identifier of server, which can be got using the "*ovcoreid*" command with the *"-ovrg server*" option.

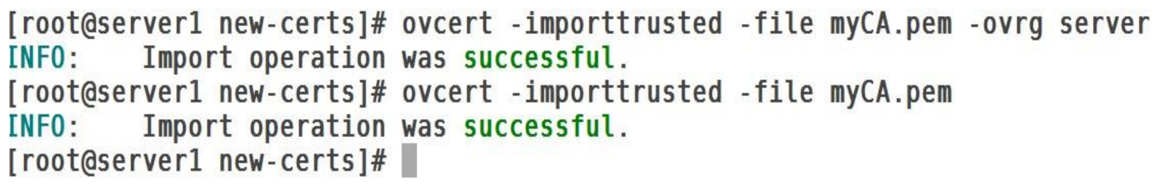
   Note: The certificate management in OBM and OA will be done through the "ovcert" command, which acts as the interface to the Keystore. The ovcert command accepts certificate only in the p12 format. Hence it's important to generate the certificate from 3rd party CA in that format.

6. The next step is to import to the node and server certificates generated.

   a) Import the trusted certificate in to the server and server resource group following the commands below.

      "ovcert –importtrusted –file <certificate file> -ovrg server"
      "ovcert –importtrusted –file <certificate file>"

      For example, as shown in Figure 5.



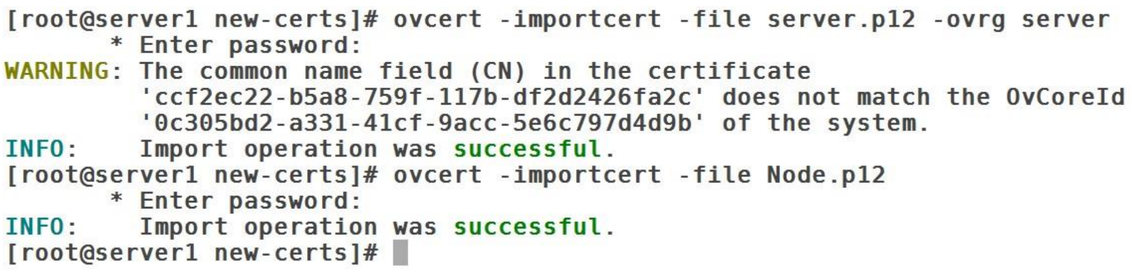
```
[root@server1 new-certs]# ovcert -importtrusted -file myCA.pem -ovrg server
INFO:    Import operation was successful.
[root@server1 new-certs]# ovcert -importtrusted -file myCA.pem
INFO:    Import operation was successful.
[root@server1 new-certs]#
```

Figure 5

   b) The next step after importing the trusted certificate is importing the server and node certificate. This can be done using the commands listed below.

      "ovcert –importcert –file <server certificate file> -ovrg server"
      "ovcert –importcert –file <node certificate file>"

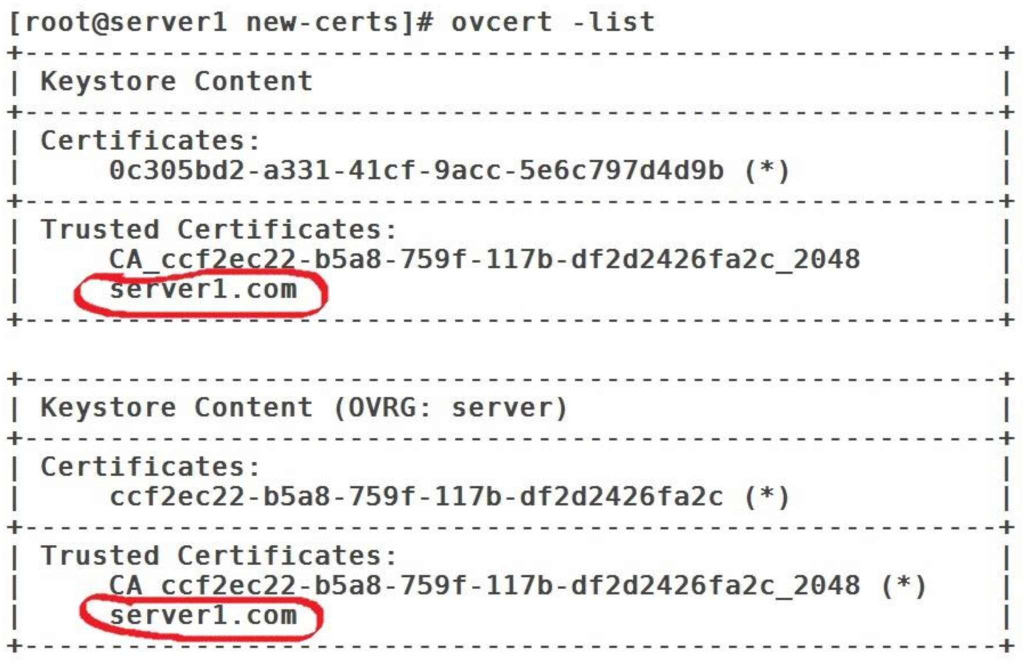
      For example, as shown in Figure 6.



```
[root@server1 new-certs]# ovcert -importcert -file server.p12 -ovrg server
       * Enter password:
WARNING: The common name field (CN) in the certificate
         'ccf2ec22-b5a8-759f-117b-df2d2426fa2c' does not match the OvCoreId
         '0c305bd2-a331-41cf-9acc-5e6c797d4d9b' of the system.
INFO:    Import operation was successful.
[root@server1 new-certs]# ovcert -importcert -file Node.p12
       * Enter password:
INFO:    Import operation was successful.
[root@server1 new-certs]#
```

Figure 6

c) Now execute the config server wizard and re-check the certificate status. After executing the configuration wizard, you should be able to see that the certificate generated by the 3rd party CA is present and active. For example, as show in Figure 7.

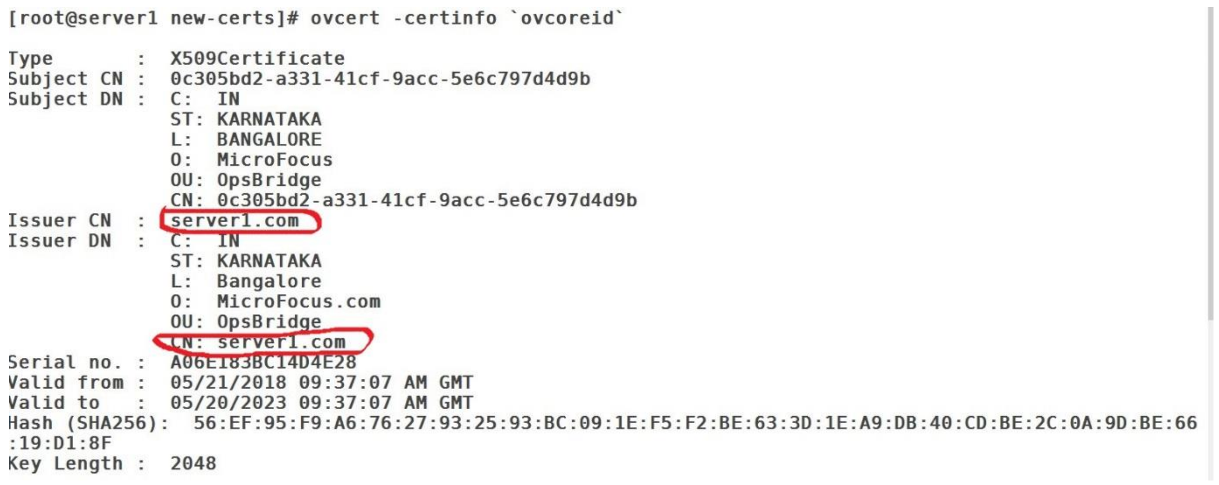


```
[root@server1 new-certs]# ovcert -list
+---------------------------------------------------------+
| Keystore Content                                        |
+---------------------------------------------------------+
| Certificates:                                           |
|     0c305bd2-a331-41cf-9acc-5e6c797d4d9b (*)            |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ccf2ec22-b5a8-759f-117b-df2d2426fa2c_2048        |
|     server1.com                                         |
+---------------------------------------------------------+

+---------------------------------------------------------+
| Keystore Content (OVRG: server)                         |
+---------------------------------------------------------+
| Certificates:                                           |
|     ccf2ec22-b5a8-759f-117b-df2d2426fa2c (*)            |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ccf2ec22-b5a8-759f-117b-df2d2426fa2c_2048 (*)    |
|     server1.com                                         |
+---------------------------------------------------------+
```

Figure 7

Also check the issuer CN of the newly imported certificates. It should be of the 3rd party CA as shown in the figure below, Figure 8.



```
[root@server1 new-certs]# ovcert -certinfo `ovcoreid`

Type         :  X509Certificate
Subject CN  :  0c305bd2-a331-41cf-9acc-5e6c797d4d9b
Subject DN  :  C:  IN
               ST: KARNATAKA
               L:  BANGALORE
               O:  MicroFocus
               OU: OpsBridge
               CN: 0c305bd2-a331-41cf-9acc-5e6c797d4d9b
Issuer CN   :  server1.com
Issuer DN   :  C:  IN
               ST: KARNATAKA
               L:  Bangalore
               O:  MicroFocus.com
               OU: OpsBridge
               CN: server1.com
Serial no.  :  A06E183BC14D4E28
Valid from  :  05/21/2018 09:37:07 AM GMT
Valid to    :  05/20/2023 09:37:07 AM GMT
Hash (SHA256):  56:EF:95:F9:A6:76:27:93:25:93:BC:09:1E:F5:F2:BE:63:3D:1E:A9:DB:40:CD:BE:2C:0A:9D:BE:66
:19:D1:8F
Key Length  :  2048
```

Figure 8

2.2 CREATING CERTIFICATES IN A DISTRIBUTED OBM INSTALLATION

A distributed OBM installation can include multiple Data Processing Systems (DPS) and Gateways.

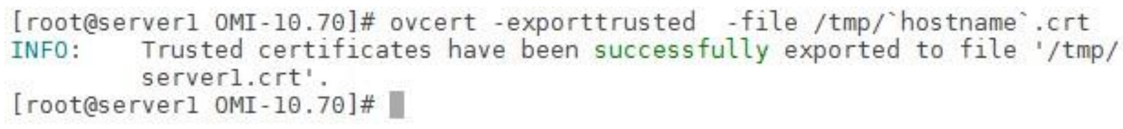
For the purpose explanation, let us consider a setup with two DPS systems and two Gateway systems.

1. For the first DPS system, follow all the steps mentioned in section 2.1

2. Export the trusted certificate to an external file using the command,

   “ovcert –exporttrusted –file <trusted certificate file name>”

   For example, as shown in Figure 9.



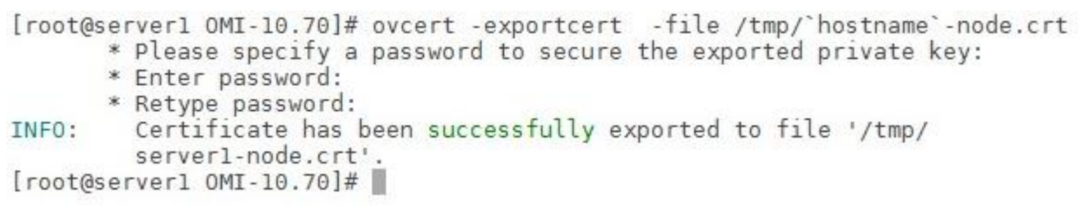
```
[root@server1 OMI-10.70]# ovcert -exporttrusted  -file /tmp/`hostname`.crt
INFO:    Trusted certificates have been successfully exported to file '/tmp/
         server1.crt'.
[root@server1 OMI-10.70]# 
```

Figure 9

3. Export the node certificate to a file executing the command below.

   “ovcert –exportcert -file <node certificate file name>”

   For example, Figure 10.



```
[root@server1 OMI-10.70]# ovcert -exportcert  -file /tmp/`hostname`-node.crt
       * Please specify a password to secure the exported private key:
       * Enter password:
       * Retype password:
INFO:    Certificate has been successfully exported to file '/tmp/
         server1-node.crt'.
[root@server1 OMI-10.70]#
```

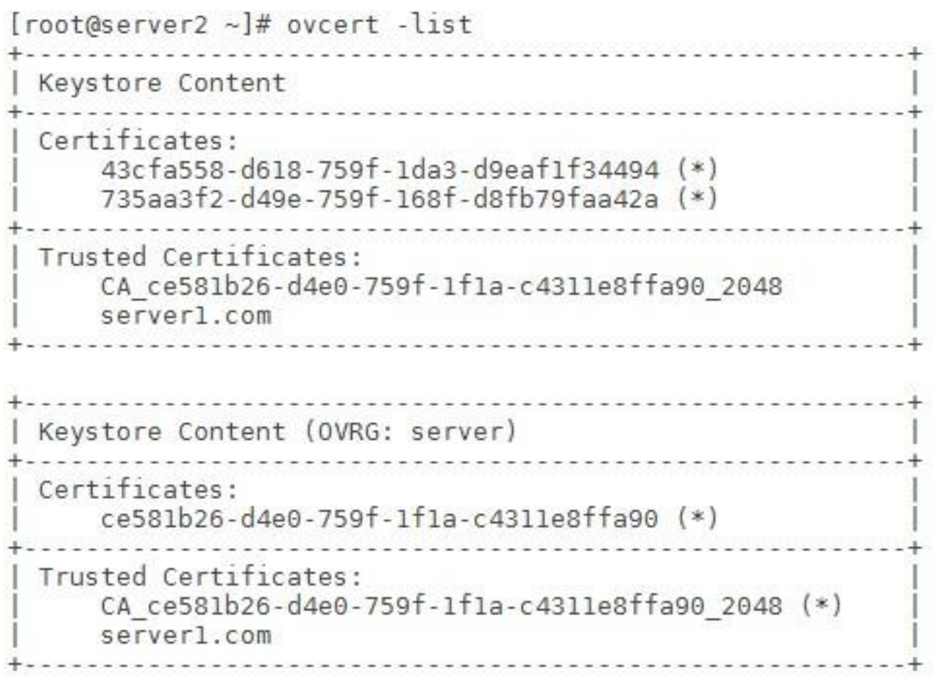
Figure 10

4. Copy both the trusted certificate file and node certificate file to the other DPS system and to both the Gateway systems. (The system should have OBM installed, but not configured).

5. Execute the below mentioned steps.

   a) On the other DPS system, import the trusted and node certificates to the default resource group and execute the configuration wizard.

      The steps are same as that in Section 2.1, steps 6a and 6b. After importing certificates the contents of the Keystore should be similar to that shown in Figure 11.



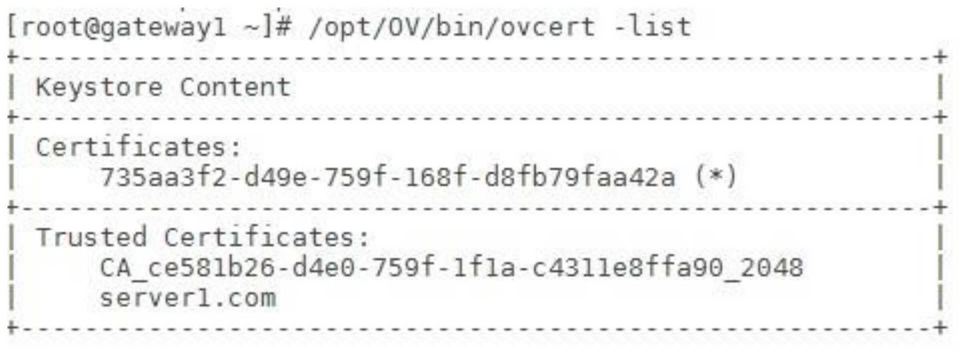
```
[root@server2 ~]# ovcert -list
+---------------------------------------------------------+
| Keystore Content                                        |
+---------------------------------------------------------+
| Certificates:                                           |
|     43cfa558-d618-759f-1da3-d9eaf1f34494 (*)            |
|     735aa3f2-d49e-759f-168f-d8fb79faa42a (*)            |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ce581b26-d4e0-759f-1f1a-c4311e8ffa90_2048        |
|     server1.com                                         |
+---------------------------------------------------------+

+---------------------------------------------------------+
| Keystore Content (OVRG: server)                         |
+---------------------------------------------------------+
| Certificates:                                           |
|     ce581b26-d4e0-759f-1f1a-c4311e8ffa90 (*)            |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ce581b26-d4e0-759f-1f1a-c4311e8ffa90_2048 (*)    |
|     server1.com                                         |
+---------------------------------------------------------+
```

Figure 11

b) On both the gateway systems, import the node certificates and the trusted certificates, following steps similar what was done above.

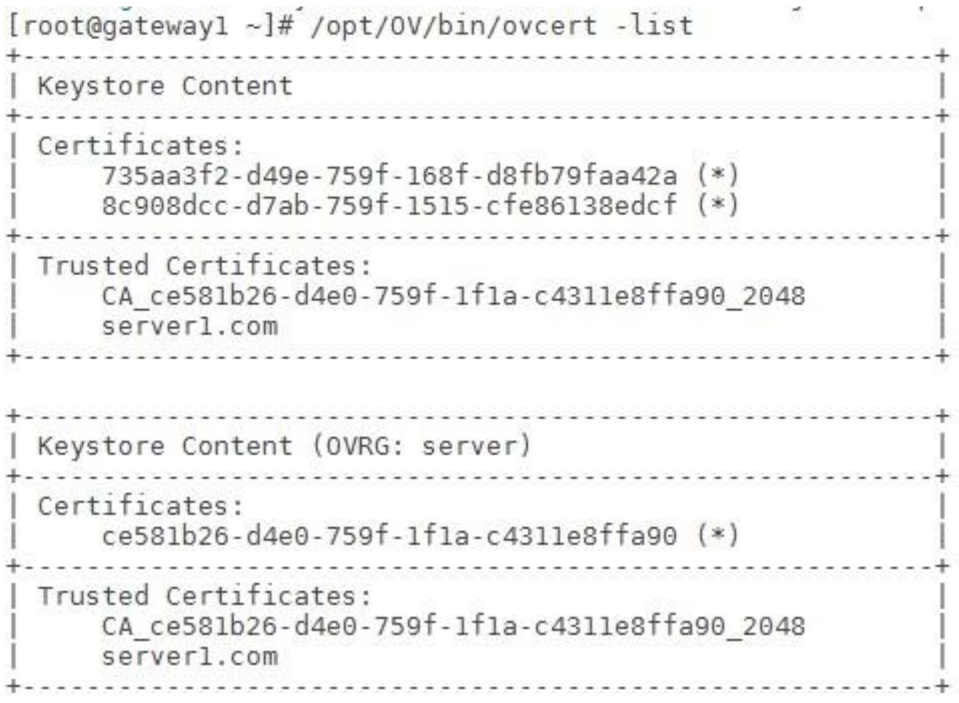
   There will be no resource group on the Gateway by default, it will be created by the configuration wizard. For example, as show in Figure 12.



```
[root@gateway1 ~]# /opt/OV/bin/ovcert -list
+---------------------------------------------------------+
| Keystore Content                                        |
+---------------------------------------------------------+
| Certificates:                                           |
|     735aa3f2-d49e-759f-168f-d8fb79faa42a (*)            |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ce581b26-d4e0-759f-1f1a-c4311e8ffa90_2048        |
|     server1.com                                         |
+---------------------------------------------------------+
```

Figure 12

As a next step, execute the configuration wizard. After running the config wizard, the ovcert –list output will resemble as shown in Figure 13



```
[root@gateway1 ~]# /opt/OV/bin/ovcert -list
+---------------------------------------------------------+
| Keystore Content                                        |
+---------------------------------------------------------+
| Certificates:                                           |
|     735aa3f2-d49e-759f-168f-d8fb79faa42a (*)            |
|     8c908dcc-d7ab-759f-1515-cfe86138edcf (*)            |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ce581b26-d4e0-759f-1f1a-c4311e8ffa90_2048        |
|     server1.com                                         |
+---------------------------------------------------------+

+---------------------------------------------------------+
| Keystore Content (OVRG: server)                         |
+---------------------------------------------------------+
| Certificates:                                           |
|     ce581b26-d4e0-759f-1f1a-c4311e8ffa90 (*)            |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ce581b26-d4e0-759f-1f1a-c4311e8ffa90_2048        |
|     server1.com                                         |
+---------------------------------------------------------+
```

Figure 13

6. Now start the OBM server and check all process are coming up on both GWs and DPS.

2.3 Creating certificates on the OA system

1. Install the agent with the command

   "oainstall.sh -i -a -cs <server-name> -srv <server-name>"

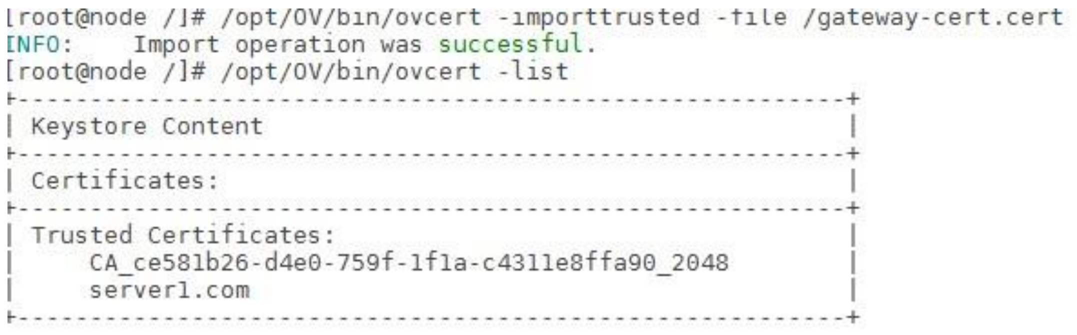
2. Copy the Root CA file (P12 file) created in the step 4 of section 2.1 to the node, using SCP/FTP or any other method of choice, and import the file as the trusted certificate.

   "ovcert –importtrusted –file <trusted certificate file>"

   When adding the node to the gateway system, export the trust certificate from the gateway as follows.

   "ovcert –exporttrusted –file /tmp/gateway-cert.crt"

   Now import the Certificate as shown in the Figure 12 below on the agent node



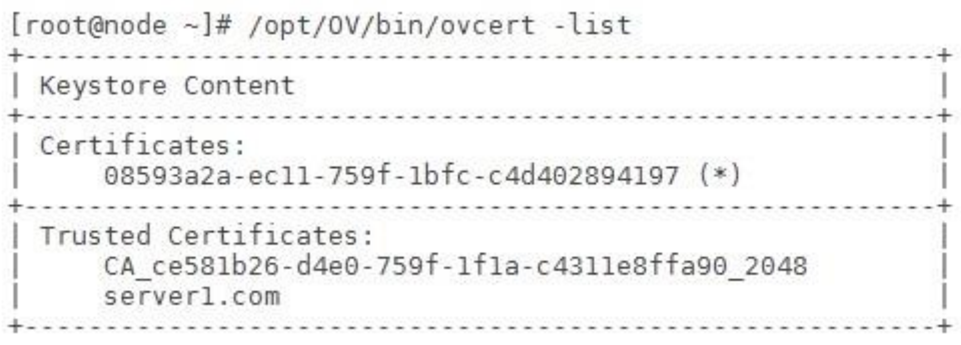
```
[root@node /]# /opt/OV/bin/ovcert -importtrusted -file /gateway-cert.cert
INFO:    Import operation was successful.
[root@node /]# /opt/OV/bin/ovcert -list
+---------------------------------------------------------+
| Keystore Content                                        |
+---------------------------------------------------------+
| Certificates:                                           |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ce581b26-d4e0-759f-1f1a-c4311e8ffa90_2048        |
|     server1.com                                         |
+---------------------------------------------------------+
```

Figure 12

3. Now, using the certificate file (root /intermediate CA) and using the agent node core identifier as the CN , generate the CSR and from the CSR, create the certificate P12 file

4. Import the certificate using the ovcert command
   "ovcert –importcert –file <P12 file>

   After importing the certificate re-start the processed using the "ovc –restart" command. For example, as show in Figure 13.



```
[root@node ~]# /opt/OV/bin/ovcert -list
+---------------------------------------------------------+
| Keystore Content                                        |
+---------------------------------------------------------+
| Certificates:                                           |
|     08593a2a-ec11-759f-1bfc-c4d402894197 (*)            |
+---------------------------------------------------------+
| Trusted Certificates:                                   |
|     CA_ce581b26-d4e0-759f-1f1a-c4311e8ffa90_2048        |
|     server1.com                                         |
+---------------------------------------------------------+
```

Figure 13

2.4 Final Test

To ensure that the certificate installation and creation of secure communication infrastructure is proper end to end, deploy a set of sample monitoring policies from the OBM Server to the OA, and verify if events are flowing in to the OBM system from the OA system.