Table of Contents

* [Custom (Named) Certificates Lab](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_custom_named_certificates_lab)
* [1. Validate DNS and Existing Named Certificates](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_validate_dns_and_existing_named_certificates)
  + [1.1. Validate DNS](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_validate_dns)
  + [1.2. Verify Named Certificate Does Not Exist](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_verify_named_certificate_does_not_exist)
* [2. Locate Named Certificate and Update Inventory File](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_locate_named_certificate_and_update_inventory_file)
  + [2.1. Locate Named Certificate](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_locate_named_certificate)
  + [2.2. Edit Inventory File](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_edit_inventory_file)
* [3. Deploy and Validate Named Certificate](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_deploy_and_validate_named_certificate)
* [4. Configure Web Browser](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_configure_web_browser)
  + [4.1. Copy Intermediate CA Certificate](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_copy_intermediate_ca_certificate)
  + [4.2. Configure Firefox Web Browser CA on Mac](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_configure_firefox_web_browser_ca_on_mac)
  + [4.3. Configure Chrome Web Browser CA on Mac](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_configure_chrome_web_browser_ca_on_mac)
  + [4.4. Access OpenShift Web Console **certtest** Address](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_access_openshift_web_console_code_certtest_code_address)
* [5. Summary](https://www.opentlc.com/labs/ocp_operations/02_04_Named_Certificates_Lab.html#_summary)

Custom (Named) Certificates Lab

**Background**

MitziCom desires that multiple hostnames resolve securely to the Master API and web console. This is not a common practice when a customer is alreadying using their own custom certificates as in other labs, but it can be the case when customers are only prepared to issue serving certificates, and not signing certificates.

Custom (named) certificates refer only to the public hostname of the Master API server and web console. Multiple named certificates can be used if there are multiple hostnames of the Master API and web console. The **openshift\_master\_named\_certificates**Ansible variable is additive for each run of Ansible. It caches all the named certificates on the master hosts. A special second variable —  **openshift\_master\_overwrite\_named\_certificates** — can be used to replace all the previously cached named certificates with the named certificate presently configured in the Ansible inventory file. This variable is dangerous to leave in an inventory file - because it may inadvertantly replace certificates which were not meant to be replaced. Instead, call the ansible playbook with that variable on the command line. This lab will show you how.

It’s your job to deploy certificates that enable certificate based security for additional hostnames access the Master API and web console.

1. Validate DNS and Existing Named Certificates

1.1. Validate DNS

In this section, you use a test domain name that has been configured to also resolve to your master’s public IP. This test domain name was automatically set up by the system. Test that it resolves to make sure it is working in DNS. The test domain is **certtest.${GUID}.example.opentlc.com**.

If this were a Highly Availabile system, you would use the load balancer’s public IP address for the named certificate. This is important to consider when planning an HA deployment.

1. Log in to your OpenShift Bastion host and become root.

sudo -i

1. Set an environment variable to equal your environment’s **GUID**:

export GUID=$(hostname | cut -f2 -d.); echo $GUID

1. Set an environment variable to equal the hostname of the new named certificate:

export NAMED\_CERT=certtest.${GUID}.example.opentlc.com; echo ${NAMED\_CERT}

1. Make sure that the correct DNS is used:

host ${NAMED\_CERT} && echo "DNS for ${NAMED\_CERT} is OK."

**Sample Output**

certtest.${GUID}.example.opentlc.com has address 34.233.202.129

DNS for certtest.${GUID}.example.opentlc.com is OK.

1.2. Verify Named Certificate Does Not Exist

In this section, you verify that the named certificate does not already exist in the OpenShift cluster.

1. Remotely connect using SSH first to your bastion host, and run the following Ansible command and examine the named certificates configuration setting:

ansible masters -m shell -a 'ls /etc/origin/master/named\_certificates/\*'

1. The output should indicate one of the following results:
   * Directory not found.
   * Directory is empty.
   * Directory does not contain a certificate file named "certtest."

2. Locate Named Certificate and Update Inventory File

2.1. Locate Named Certificate

1. Check to see if there are certs already generated on your bastion host by examining for the certs generation directory:

ls /root/certs

1. If there are is no certs on your bastion host, generate them as follows:
2. cd /root/
3. git clone https://github.com/newgoliath/certs/
4. cd certs
5. ./ca\_create.sh

./create\_env\_certs.sh -g ${GUID}

1. Make sure it created the named certificate is on your bastion host:

file /root/certs/ca/intermediate/newcerts/${GUID}/certtest.${GUID}.example.opentlc.com.cert.pem

Should have the proper output:

/root/certs/ca/intermediate/newcerts/GUID/certtest.GUID.example.opentlc.com.cert.pem: PEM certificate

1. And make sure it created the private key on your bastion host:

file /root/certs/ca/intermediate/newcerts/${GUID}/certtest.${GUID}.example.opentlc.com.key.pem

Should have the proper output:

/root/certs/ca/intermediate/newcerts/GUID/certtest.GUID.example.opentlc.com.key.pem: PEM RSA private key

2.2. Edit Inventory File

In this section, you edit the inventory file, **/etc/ansible/hosts** to include the named certificate.

1. First, find the following section of your Ansible inventory:
2. openshift\_master\_cluster\_method=native
3. openshift\_master\_cluster\_hostname=master.GUID.example.opentlc.com
4. openshift\_master\_cluster\_public\_hostname=master.GUID.example.opentlc.com
5. openshift\_master\_default\_subdomain=apps.GUID.example.opentlc.com

openshift\_set\_hostname=True

1. Second, directly below that section, add the following single long line. BE SURE TO REPLACE GUID with your guid, and be careful not to break the line:

openshift\_master\_named\_certificates=[{"certfile": "/root/certs/ca/intermediate/newcerts/GUID/certtest.GUID.example.opentlc.com.cert.pem", "keyfile": "/root/certs/ca/intermediate/newcerts/GUID/certtest.GUID.example.opentlc.com.key.pem", "cafile": "/root/certs/ca/intermediate/certs/intermediate.cert.pem"}]

1. Save the changes to the inventory file.

3. Deploy and Validate Named Certificate

1. Deploy the named certificate. It takes about 2 minutes in the OCP-Operations environment:

ansible-playbook -i /etc/ansible/hosts -f 20 /usr/share/ansible/openshift-ansible/playbooks/byo/openshift-cluster/redeploy-master-certificates.yml -e openshift\_master\_overwrite\_named\_certificates=true

1. The playbook should complete with output similar to what follows:
2. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. infranode1.GUID.internal : ok=56 changed=2 unreachable=0 failed=0
4. localhost : ok=14 changed=0 unreachable=0 failed=0
5. master1.GUID.internal : ok=180 changed=20 unreachable=0 failed=0
6. node1.GUID.internal : ok=56 changed=2 unreachable=0 failed=0
7. node2.GUID.internal : ok=56 changed=2 unreachable=0 failed=0
8. node3.GUID.internal : ok=56 changed=2 unreachable=0 failed=0

support1.GUID.internal : ok=38 changed=2 unreachable=0 failed=0

4. Configure Web Browser

4.1. Copy Intermediate CA Certificate

1. On the bastion host, copy the Intermediate certificate to your user’s home directory in order to **scp** it to your laptop.
2. sudo -i
3. cp /root/certs/ca/intermediate/certs/intermediate.cert.pem /home/<OpenTLC username>/

chown <OpenTLC username> /home/<OpenTLC username>/intermediate.cert.pem

1. On your laptop, download the following file via scp from your bastion to your laptop using **scp**. Don’t forget to change the ${guid} to your GUID:

scp -i <Your OpenTLC private key> <OpenTLC username>@bastion.${GUID}.example.opentlc.com:intermediate.cert.pem .

* + This is the Intermediate CA certificate that signed most of OpenShift’s server certificates. Note the location it was downloaded.

4.2. Configure Firefox Web Browser CA on Mac

1. In Firefox, navigate to **about:preferences#advanced**.
2. Navigate to **Certificates → View Certificates → Authorities**:
3. Click **Import** and select the Intermediate CA certificate file from your download location.
4. Check the **Trust this CA to identify websites** box and click **OK.**
5. Click **OK** again to close the certificates box.
6. Browse to your master’s public hostname at **https://master.$GUID.example.opentlc.com/**, substituting your GUID for the **$GUID** in the URL.
   * Note that you do not get a certificate error. You are able to log in and browse without warnings or errors.

4.3. Configure Chrome Web Browser CA on Mac

In this section, you add the Intermediate CA certificate to your operating system’s certificate management application.

1. In Chrome, navigate to **chrome://settings/?search=Manage%20certificates** and click **Manage Certificates**.
   * Expect your operating system’s certificate management box to appear.

|  |  |
| --- | --- |
|  | The instructions that follow are for the macOS operating system. |

1. To import the Intermediate CA file into the **System** keychain, click the **System** keychain, and then click the padlock above **Click to unlock the System keychain**:
2. Enter your password and click **Modify Keychain**.
3. At the bottom of the box, click the section "Certificates" and then the **+** to add the certificate.
4. From the file selection dialog box, select the Intermediate CA certificate file and click **Open**.
5. If prompted for your macOS password, enter it to import the certificate.
6. Select the name of the new Intermediate CA from the list and type **command+I** to invoke the **Get Info** dialog for this certificate.
7. Click the triangle to expand the **Trust** section.
8. Select the list box next to **Secure Sockets Layer (SSL)** and set it to **Always Trust**, and close the window.
9. Enter your password again if prompted.
10. Close the **Keychain Access** window.

4.4. Access OpenShift Web Console **certtest** Address

1. Browse to your master’s alternative hostname at **https://certtest.$GUID.example.opentlc.com/**, substituting your GUID for the **$GUID** in the URL.
   * Note that you do not get a certificate error. You are able to log in and browse without warnings or errors.
   * However, the OpenShift Web Console immediately redirects you to **https://master.GUID.example.opentlc.com** This is normal. It’s happening because the **openshift\_master\_cluster\_public\_hostname** has not been changed to the **certtest**domain name. If it is, there will be no redirects to the **master** hostname.
   * Replace **master** in the navigation bar with **certtest**, and hit Enter. Note that there are still no certificate requests.

5. Summary

This lab introduces the idea of **named certificate**. Named certificates are are used to enable custom certificates for OpenShift Master API hostnames. It shows how to install and test named certificates.

Build Version: be32934b41da1e34c27e21e79bdf57d30c732a3e : Last updated 2018-01-19 05:59:04 EST