



Manage certificates

ONTAP Mediator

NetApp
October 19, 2026

Table of Contents

- Manage certificates 1
 - Regenerate a temporary self-signed certificate for the ONTAP Mediator host 1
 - Replace ONTAP Mediator self-signed certificates with trusted third-party certificates 2
 - Step 1: Obtain a certificate from a third-party issuing a CA certificate..... 3
 - Step 2: Generate a server certificate by signing with a third-party CA certification 5
 - Step 3: Replace new third-party CA certificate and server certificate in ONTAP Mediator configuration .. 6
 - Step 4: Optionally, use a different path or name for your third-party certificates 10
- Troubleshoot ONTAP Mediator certificate-related issues 16
 - Verify certificate expiration 16
 - Verify X509v3 extensions in CA certification 16
 - Verify X509v3 extensions in server certificate and subject Alt Names..... 18
 - Verify that a private key matches with a certificate 20
 - Verify that a server certificate is created from a particular CA certificate..... 21

Manage certificates

Regenerate a temporary self-signed certificate for the ONTAP Mediator host

Regenerate a temporary self-signed certificate for the current host when a hostname or IP address change invalidates the existing certificate.



This procedure is only supported on systems running ONTAP Mediator 1.7 or later.

About this task

- Perform this task on the Linux host where you installed ONTAP Mediator.
- You can perform this task only if the generated self-signed certificates have become obsolete due to changes to the hostname or IP address of the host after installing ONTAP Mediator.
- After the temporary self-signed certificate has been replaced by a trusted third-party certificate, you do *not* use this task to regenerate a certificate. The absence of a self-signed certificate will cause this procedure to fail.

Step

To regenerate a new temporary self-signed certificate for the current host, perform the following step:

1. Restart ONTAP Mediator:

```
./make_self_signed_certs.sh overwrite
```

```
[root@xyz000123456 ~]# cd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config
[root@xyz000123456 server_config]# ./make_self_signed_certs.sh overwrite

Adding Subject Alternative Names to the self-signed server certificate
#
# OpenSSL example configuration file.
Generating self-signed certificates
Generating RSA private key, 4096 bit long modulus (2 primes)
.....
.....
.....++++
.....++++
e is 65537 (0x010001)
Generating a RSA private key
.....++++
.....
.....+++
+
writing new private key to 'ontap_mediator_server.key'
-----
```

```
Signature ok
subject=C = US, ST = California, L = San Jose, O = "NetApp, Inc.", OU =
ONTAP Core Software, CN = ONTAP Mediator, emailAddress =
support@netapp.com
Getting CA Private Key
```

Replace ONTAP Mediator self-signed certificates with trusted third-party certificates

If your environment supports it, replace self-signed certificates with trusted certificates issued by a certificate authority.

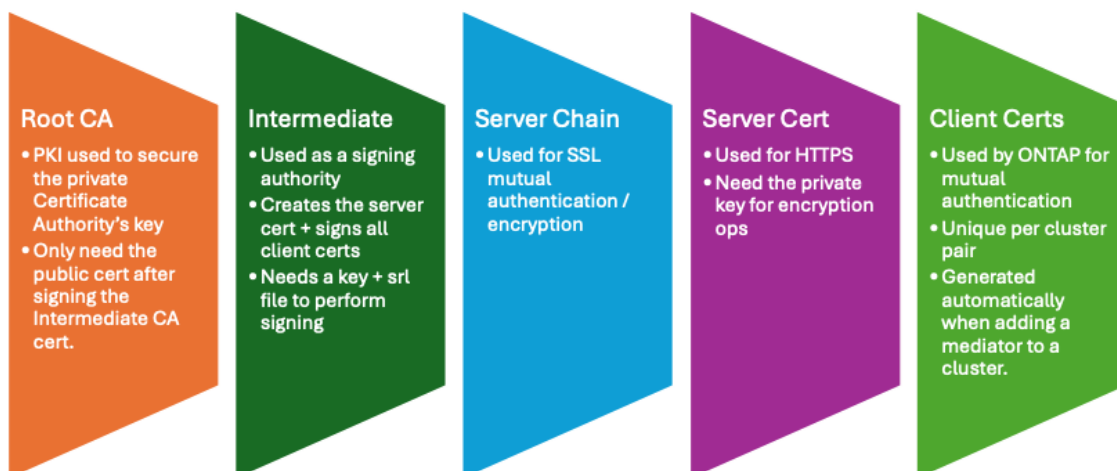


- Third-party certificates are only supported beginning with ONTAP 9.16.1 and in some earlier ONTAP patch releases. See [NetApp Bugs Online Bug ID CONTAP-243278](#).
- Third-party certificates are only supported on systems running ONTAP Mediator 1.7 or later.

About this task

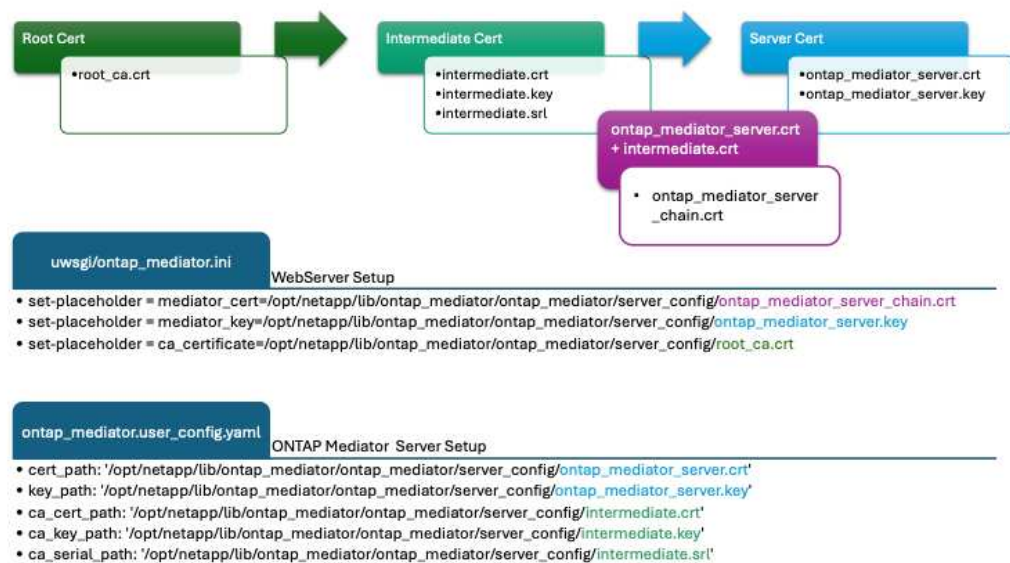
- Perform this task on the Linux host where you installed ONTAP Mediator.
- You can perform this task if the generated self-signed certificates need to be replaced by certificates obtained from a trusted subordinate certificate authority (CA). To accomplish this, you should have access to a trusted public-key infrastructure (PKI) authority.
- The following image shows the purposes of each ONTAP Mediator certificate.

ONTAP Mediator Certificate Purposes



- The following image shows configuration for the web server setup and ONTAP Mediator setup.

ONTAP Mediator Certificates



Step 1: Obtain a certificate from a third-party issuing a CA certificate

You can obtain a certificate from a PKI authority using the following procedure.

The following example demonstrates replacing the self-signed certificate actors with the third-party certificate actors located at /opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/.



The example illustrates the necessary criteria for the certificates required for ONTAP Mediator. You can obtain the certificates from a PKI authority in a way that might be different to this procedure. Adjust the procedure according to your business need.

ONTAP Mediator 1.9 and later

1. Create a private key `intermediate.key` and a configuration file `openssl_ca.cnf` that will be consumed by the PKI authority to generate a certificate.
 - a. Generate the private key `intermediate.key`:
2. Use the private key and configuration file to create a certificate signing request `intermediate.csr`:

Example:

```
openssl req -key <private_key_name>.key -new -out
<certificate_csr_name>.csr -config <config_file_name>.cnf
```

```
[root@mediator_host server_config]# openssl req -key
intermediate.key -new -config openssl_ca.cnf -out intermediate.csr
Enter pass phrase for intermediate.key:
[root@mediator_host server_config]# cat intermediate.csr
-----BEGIN CERTIFICATE REQUEST-----
<certificate_value>
-----END CERTIFICATE REQUEST-----
```

3. Send the certificate signing request `intermediate.csr` to a PKI authority for their signature.

The PKI authority verifies the request and signs the `.csr`, generating the certificate `intermediate.crt`. Additionally, you need to obtain the `root_ca.crt` certificate that signed the `intermediate.crt` certificate from the PKI authority.



For SnapMirror active sync clusters and MetroCluster IP configurations using ONTAP Mediator configured with HTTPS, you must add the `intermediate.crt` and `root_ca.crt` certificates to an ONTAP cluster. See [Configure ONTAP Mediator and clusters for SnapMirror active sync](#) and [Configure the ONTAP Mediator for a MetroCluster IP configuration](#).

ONTAP Mediator 1.8 and earlier

1. Create a private key `ca.key` and a configuration file `openssl_ca.cnf` that will be consumed by the PKI authority to generate a certificate.
 - a. Generate the private key `ca.key`:

Example:

```
openssl genrsa -aes256 -out ca.key 4096
```
 - b. The configuration file `openssl_ca.cnf` (located at `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_ca.cnf`) defines the properties that the generated certificate must have.
2. Use the private key and configuration file to create a certificate signing request `ca.csr`:

Example:

```
openssl req -key <private_key_name>.key -new -out
<certificate_csr_name>.csr -config <config_file_name>.cnf
```

```
[root@mediator_host server_config]# openssl req -key ca.key -new
-config openssl_ca.cnf -out ca.csr
Enter pass phrase for ca.key:
[root@mediator_host server_config]# cat ca.csr
-----BEGIN CERTIFICATE REQUEST-----
<certificate_value>
-----END CERTIFICATE REQUEST-----
```

3. Send the certificate signing request `ca.csr` to a PKI authority for their signature.

The PKI authority verifies the request and signs the `.csr`, generating the certificate `ca.crt`. Additionally, you need to obtain the `root_ca.crt` certificate that signed the `ca.crt` certificate from the PKI authority.



For SnapMirror active sync clusters, you must add the `ca.crt` and `root_ca.crt` certificates to an ONTAP cluster. See [Configure ONTAP Mediator and clusters for SnapMirror active sync](#).

Step 2: Generate a server certificate by signing with a third-party CA certification

ONTAP Mediator 1.9 and later

A server certificate must be signed by the private key `intermediate.key` and the third-party certificate `intermediate.crt`. Additionally, the configuration file `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_server.cnf` contains certain attributes that specify the properties required for server certificates issued by OpenSSL.

The following commands can generate a server certificate.

Steps

1. To generate a server certificate signing request (CSR), run the following command from the `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config` folder:

```
openssl req -config openssl_server.cnf -extensions v3_req -nodes -newkey  
rsa:4096 -sha512 -keyout ontap_mediator_server.key -out  
ontap_mediator_server.csr
```

2. To generate a server certificate from the CSR, run the following command from the `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config` folder:



These files were obtained from a PKI authority. If you are using a different certificate name, replace `intermediate.crt` and `intermediate.key` with the relevant file names.

```
openssl x509 -extfile openssl_server.cnf -extensions v3_req -CA  
intermediate.crt -CAkey intermediate.key -CAcreateserial -sha512 -days 1095  
-req -in ontap_mediator_server.csr -out ontap_mediator_server.crt
```

- The `-CAcreateserial` option is used to generate the `intermediate.srl` files.

ONTAP Mediator 1.8 and earlier

A server certificate must be signed by the private key `ca.key` and the third-party certificate `ca.crt`. Additionally, the configuration file

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_server.cnf` contains certain attributes that specify the properties required for server certificates issued by OpenSSL.

The following commands can generate a server certificate.

Steps

1. To generate a server certificate signing request (CSR), run the following command from the `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config` folder:

```
openssl req -config openssl_server.cnf -extensions v3_req -nodes -newkey  
rsa:4096 -sha512 -keyout ontap_mediator_server.key -out  
ontap_mediator_server.csr
```

2. To generate a server certificate from the CSR, run the following command from the `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config` folder:



These files were obtained from a PKI authority. If you are using a different certificate name, replace `ca.crt` and `ca.key` with the relevant file names.

```
openssl x509 -extfile openssl_server.cnf -extensions v3_req -CA ca.crt  
-CAkey ca.key -CAcreateserial -sha512 -days 1095 -req -in  
ontap_mediator_server.csr -out ontap_mediator_server.crt
```

◦ The `-CAcreateserial` option is used to generate the `ca.srl` files.

Step 3: Replace new third-party CA certificate and server certificate in ONTAP Mediator configuration

ONTAP Mediator 1.10 and later

The certificate configuration is supplied to ONTAP Mediator in the configuration file located at `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml`. The file includes the following attributes:

```
cert_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi  
ator_server.crt'  
key_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi  
ator_server.key'  
ca_cert_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermedia  
te.crt'  
ca_key_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermedia  
te.key'  
ca_serial_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermedia  
te.srl'
```

- `cert_path` and `key_path` are server certificate variables.
- `ca_cert_path`, `ca_key_path`, and `ca_serial_path` are CA certificate variables.

Steps

1. Replace all `intermediate.*` files with the third-party certificates.
2. Create a certificate chain from the `intermediate.crt` and `ontap_mediator_server.crt` certificates:

```
cat ontap_mediator_server.crt intermediate.crt >
ontap_mediator_server_chain.crt
```

3. Update the `/opt/netapp/lib/ontap_mediator/uvicorn/config.json` file.

Update the values of `ssl_keyfile`, `ssl_certfile`, and `ssl_ca_certs`:

```
ssl_keyfile:
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_
server.key
```

```
ssl_certfile:
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_
server_chain.crt
```

```
ssl_ca_certs:
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- The `ssl_keyfile` value is the key path in the `ontap_mediator_server.crt` file, which is `ontap_mediator_server.key`.
 - The `ssl_certfile` value is the path of the `ontap_mediator_server_chain.crt` file.
 - The `ssl_ca_certs` value is the path of the `root_ca.crt` file.
4. Verify that the following attributes of the newly generated certificates are set correctly:
 - Linux Group Owner: `netapp:netapp`
 - Linux permissions: `600`
 5. Restart ONTAP Mediator:

```
systemctl restart ontap_mediator
```

ONTAP Mediator 1.9.1 and 1.9

The certificate configuration is supplied to ONTAP Mediator in the configuration file located at `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml`. The file includes the following attributes:

```
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi
ator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi
ator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermedia
```

```
te.crt'  
ca_key_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermedia  
te.key'  
ca_serial_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermedia  
te.srl'
```

- `cert_path` and `key_path` are server certificate variables.
- `ca_cert_path`, `ca_key_path`, and `ca_serial_path` are CA certificate variables.

Steps

1. Replace all `intermediate.*` files with the third-party certificates.
2. Create a certificate chain from the `intermediate.crt` and `ontap_mediator_server.crt` certificates:

```
cat ontap_mediator_server.crt intermediate.crt >  
ontap_mediator_server_chain.crt
```

3. Update the `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` file.

Update the values of `mediator_cert`, `mediator_key`, and `ca_certificate`:

```
set-placeholder = mediator_cert =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_  
server_chain.crt
```

```
set-placeholder = mediator_key =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_  
server.key
```

```
set-placeholder = ca_certificate =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- The `mediator_cert` value is the path of the `ontap_mediator_server_chain.crt` file.
 - The `mediator_key` value is the key path in the `ontap_mediator_server.crt` file, which is `ontap_mediator_server.key`.
 - The `ca_certificate` value is the path of the `root_ca.crt` file.
4. Verify that the following attributes of the newly generated certificates are set correctly:
 - Linux Group Owner: `netapp:netapp`
 - Linux permissions: `600`
 5. Restart ONTAP Mediator:

```
systemctl restart ontap_mediator
```

ONTAP Mediator 1.8 and earlier

The certificate configuration is supplied to ONTAP Mediator in the configuration file located at `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml`. The file includes the following attributes:

```
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.srl'
```

- `cert_path` and `key_path` are server certificate variables.
- `ca_cert_path`, `ca_key_path`, and `ca_serial_path` are CA certificate variables.

Steps

1. Replace all `ca.*` files with the third-party certificates.
2. Create a certificate chain from the `ca.crt` and `ontap_mediator_server.crt` certificates:

```
cat ontap_mediator_server.crt ca.crt > ontap_mediator_server_chain.crt
```

3. Update the `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` file.

Update the values of `mediator_cert`, `mediator_key`, and `ca_certificate`:

```
set-placeholder = mediator_cert =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server_chain.crt
```

```
set-placeholder = mediator_key =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key
```

```
set-placeholder = ca_certificate =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- The `mediator_cert` value is the path of the `ontap_mediator_server_chain.crt` file.
- The `mediator_key` value is the key path in the `ontap_mediator_server.crt` file, which is `ontap_mediator_server.key`.
- The `ca_certificate` value is the path of the `root_ca.crt` file.

4. Verify that the following attributes of the newly generated certificates are set correctly:

- Linux Group Owner: netapp:netapp
- Linux permissions: 600

5. Restart ONTAP Mediator:

```
systemctl restart ontap_mediator
```

Step 4: Optionally, use a different path or name for your third-party certificates

ONTAP Mediator 1.10 and later

You can use third-party certificates with a different name other than `intermediate.*` or store the third-party certificates in a different location.

Steps

1. Configure the

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.user_config.yaml` file to override the default variable values in the `ontap_mediator.config.yaml` file.

If you obtained `intermediate.crt` from a PKI authority and you store its private key `intermediate.key` at the location

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config`, the `ontap_mediator.user_config.yaml` file should look like the following example:



If you used `intermediate.crt` to sign the `ontap_mediator_server.crt` certificate, the `intermediate.srl` file is generated. See [Step 2: Generate a server certificate by signing with a third-party CA certification](#) for more information.

```
[root@mediator_host server_config]# cat
ontap_mediator.user_config.yaml

# This config file can be used to override the default settings in
ontap_mediator.config.yaml
# To override a setting, copy the property key from
ontap_mediator.config.yaml to this file and
# set the property to the desired value. e.g.,
#
# The default value for 'default_mailboxes_per_target' is 4 in
ontap_mediator.config.yaml
#
# To override this value with 6 mailboxes per target, add the
following key/value pair
# below this comment:
#
# 'default_mailboxes_per_target': 6
```

```
#
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.srl'
```

- a. If you are using a certificate structure where the `root_ca.crt` certificate provides an `intermediate.crt` certificate that signs the `ontap_mediator_server.crt` certificate, create a certificate chain from the `intermediate.crt` and `ontap_mediator_server.crt` certificates:



You should have obtained the `intermediate.crt` and `ontap_mediator_server.crt` certificates from a PKI authority earlier in the procedure.

```
cat ontap_mediator_server.crt intermediate.crt >
ontap_mediator_server_chain.crt
```

- b. Update the `/opt/netapp/lib/ontap_mediator/uvicorn/config.json` file.

Update the values of `ssl_keyfile`, `ssl_certfile`, and `ssl_ca_certs`:

```
ssl_keyfile:
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi
ator_server.key
```

```
ssl_certfile:
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi
ator_server_chain.crt
```

```
ssl_ca_certs:
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- The `ssl_keyfile` value is the key path in the `ontap_mediator_server.crt` file, which is `ontap_mediator_server.key`.
- The `ssl_certfile` value is the path of the `ontap_mediator_server_chain.crt` file.
- The `ssl_ca_certs` value is the path of the `root_ca.crt` file.



For SnapMirror active sync clusters and MetroCluster IP configurations using ONTAP Mediator configured with HTTPS, you must add the `intermediate.crt` and `root_ca.crt` certificates to an ONTAP cluster. See [Configure ONTAP Mediator and clusters for SnapMirror active sync](#) and [Configure the ONTAP Mediator for a MetroCluster IP configuration](#).

c. Verify that the following attributes of the newly generated certificates are set correctly:

- Linux Group Owner: `netapp:netapp`
- Linux permissions: `600`

2. Restart ONTAP Mediator when the certificates are updated in the configuration file:

```
systemctl restart ontap_mediator
```

ONTAP Mediator 1.9.1 and 1.9

You can use third-party certificates with a different name other than `intermediate.*` or store the third-party certificates in a different location.

Steps

1. Configure the

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.user_config.yaml` file to override the default variable values in the `ontap_mediator.config.yaml` file.

If you obtained `intermediate.crt` from a PKI authority and you store its private key `intermediate.key` at the location

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config`, the `ontap_mediator.user_config.yaml` file should look like the following example:



If you used `intermediate.crt` to sign the `ontap_mediator_server.crt` certificate, the `intermediate.srl` file is generated. See [Step 2: Generate a server certificate by signing with a third-party CA certification](#) for more information.

```
[root@mediator_host server_config]# cat
ontap_mediator.user_config.yaml

# This config file can be used to override the default settings in
ontap_mediator.config.yaml
# To override a setting, copy the property key from
ontap_mediator.config.yaml to this file and
# set the property to the desired value. e.g.,
#
# The default value for 'default_mailboxes_per_target' is 4 in
ontap_mediator.config.yaml
#
# To override this value with 6 mailboxes per target, add the
following key/value pair
# below this comment:
```

```
#
# 'default_mailboxes_per_target': 6
#
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.srl'
```

- a. If you are using a certificate structure where the `root_ca.crt` certificate provides an `intermediate.crt` certificate that signs the `ontap_mediator_server.crt` certificate, create a certificate chain from the `intermediate.crt` and `ontap_mediator_server.crt` certificates:



You should have obtained the `intermediate.crt` and `ontap_mediator_server.crt` certificates from a PKI authority earlier in the procedure.

```
cat ontap_mediator_server.crt intermediate.crt >
ontap_mediator_server_chain.crt
```

- b. Update the `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` file.

Update the values of `mediator_cert`, `mediator_key`, and `ca_certificate`:

```
set-placeholder = mediator_cert =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server_chain.crt
```

```
set-placeholder = mediator_key =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key
```

```
set-placeholder = ca_certificate =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- The `mediator_cert` value is the path of the `ontap_mediator_server_chain.crt` file.
- The `mediator_key` value is the key path in the `ontap_mediator_server.crt` file, which

is `ontap_mediator_server.key`.

- The `ca_certificate` value is the path of the `root_ca.crt` file.



For SnapMirror Business Continuity (SM-BC) clusters, you must add the `intermediate.crt` and `root_ca.crt` certificates to an ONTAP cluster. See [Configure ONTAP Mediator and clusters for SnapMirror active sync](#).

c. Verify that the following attributes of the newly generated certificates are set correctly:

- Linux Group Owner: `netapp:netapp`
- Linux permissions: `600`

2. Restart ONTAP Mediator when the certificates are updated in the configuration file:

```
systemctl restart ontap_mediator
```

ONTAP Mediator 1.8 and earlier

You can use third-party certificates with a different name other than `ca.*` or store the third-party certificates in a different location.

Steps

1. Configure the

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.user_config.yaml` file to override the default variable values in the `ontap_mediator.config.yaml` file.

If you obtained `ca.crt` from a PKI authority and you store its private key `ca.key` at the location `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config`, the `ontap_mediator.user_config.yaml` file should look like the following example:



If you used `ca.crt` to sign the `ontap_mediator_server.crt` certificate, the `ca.srl` file is generated. See [Step 2: Generate a server certificate by signing with a third-party CA certification](#) for more information.

```
[root@mediator_host server_config]# cat
ontap_mediator.user_config.yaml

# This config file can be used to override the default settings in
ontap_mediator.config.yaml
# To override a setting, copy the property key from
ontap_mediator.config.yaml to this file and
# set the property to the desired value. e.g.,
#
# The default value for 'default_mailboxes_per_target' is 4 in
ontap_mediator.config.yaml
#
# To override this value with 6 mailboxes per target, add the
following key/value pair
```

```
# below this comment:
#
# 'default_mailboxes_per_target': 6
#
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.srl'
```

- a. If you are using a certificate structure where the `root_ca.crt` certificate provides an `ca.crt` certificate that signs the `ontap_mediator_server.crt` certificate, create a certificate chain from the `ca.crt` and `ontap_mediator_server.crt` certificates:



You should have obtained the `ca.crt` and `ontap_mediator_server.crt` certificates from a PKI authority earlier in the procedure.

```
cat ontap_mediator_server.crt ca.crt > ontap_mediator_server_chain.crt
```

- b. Update the `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` file.

Update the values of `mediator_cert`, `mediator_key`, and `ca_certificate`:

```
set-placeholder = mediator_cert =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi
ator_server_chain.crt
```

```
set-placeholder = mediator_key =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi
ator_server.key
```

```
set-placeholder = ca_certificate =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- The `mediator_cert` value is the path of the `ontap_mediator_server_chain.crt` file.
- The `mediator_key` value is the key path in the `ontap_mediator_server.crt` file, which is `ontap_mediator_server.key`.
- The `ca_certificate` value is the path of the `root_ca.crt` file.



For SnapMirror Business Continuity (SM-BC) clusters, you must add the

ca.crt and root_ca.crt certificates to an ONTAP cluster. See [Configure ONTAP Mediator and clusters for SnapMirror active sync](#).

c. Verify that the following attributes of the newly generated certificates are set correctly:

- Linux Group Owner: netapp:netapp
- Linux permissions: 600

2. Restart ONTAP Mediator when the certificates are updated in the configuration file:

```
systemctl restart ontap_mediator
```

Troubleshoot ONTAP Mediator certificate-related issues

Use these checks to verify certificate properties when troubleshooting certificate issues.

Verify certificate expiration

Use the following command to identify the certificate validity range.

ONTAP Mediator 1.9 and later

```
[root@mediator_host server_config]# openssl x509 -in intermediate.crt
-text -noout
Certificate:
    Data:
    ...
        Validity
            Not Before: Feb 22 19:57:25 2024 GMT
            Not After  : Feb 15 19:57:25 2029 GMT
```

ONTAP Mediator 1.8 and earlier

```
[root@mediator_host server_config]# openssl x509 -in ca.crt -text
-noout
Certificate:
    Data:
    ...
        Validity
            Not Before: Feb 22 19:57:25 2024 GMT
            Not After  : Feb 15 19:57:25 2029 GMT
```

Verify X509v3 extensions in CA certification

Use the following command to verify the X509v3 extensions in the CA certification.

ONTAP Mediator 1.9 and later

The properties defined within **v3_ca** in `openssl_ca.cnf` are displayed as X509v3 extensions in `intermediate.crt`.

```
[root@mediator_host server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@mediator_host server_config]# cat openssl_ca.cnf
...
[ v3_ca ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, cRLSign, digitalSignature, keyCertSign

[root@mediator_host server_config]# openssl x509 -in intermediate.crt
-text -noout
Certificate:
    Data:
    ...
        X509v3 extensions:
            X509v3 Subject Key Identifier:

9F:06:FA:47:00:67:BA:B2:D4:82:70:38:B8:48:55:B5:24:DB:FC:27
            X509v3 Authority Key Identifier:

keyid:9F:06:FA:47:00:67:BA:B2:D4:82:70:38:B8:48:55:B5:24:DB:FC:27

            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
```

ONTAP Mediator 1.8 and earlier

The properties defined within **v3_ca** in `openssl_ca.cnf` are displayed as X509v3 extensions in `ca.crt`.

```
[root@mediator_host server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@mediator_host server_config]# cat openssl_ca.cnf
...
[ v3_ca ]
subjectKeyIdentifier = hash
```

```

authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, cRLSign, digitalSignature, keyCertSign

[root@mediator_host server_config]# openssl x509 -in ca.crt -text
-noout
Certificate:
    Data:
    ...
        X509v3 extensions:
            X509v3 Subject Key Identifier:

            9F:06:FA:47:00:67:BA:B2:D4:82:70:38:B8:48:55:B5:24:DB:FC:27
            X509v3 Authority Key Identifier:

            keyid:9F:06:FA:47:00:67:BA:B2:D4:82:70:38:B8:48:55:B5:24:DB:FC:27

            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign

```

Verify X509v3 extensions in server certificate and subject Alt Names

The `v3_req` properties defined in the `openssl_server.cnf` configuration file are displayed as X509v3 extensions in the certificate.

In the following example, you can obtain the variables in the `alt_names` sections by running the commands `hostname -A` and `hostname -I` on the Linux VM on which ONTAP Mediator is installed.

Check with your network administrator for the correct values of the variables.

ONTAP Mediator 1.9 and later

```

[root@mediator_host server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@mediator_host server_config]# cat openssl_server.cnf
...
[ v3_req ]
basicConstraints          = CA:false
extendedKeyUsage          = serverAuth
keyUsage                  = keyEncipherment, dataEncipherment
subjectAltName             = @alt_names

```

```

[ alt_names ]
DNS.1 = abc.company.com
DNS.2 = abc-v6.company.com
IP.1 = 1.2.3.4
IP.2 = abcd:abcd:abcd:abcd:abcd:abcd

[root@mediator_host server_config]# openssl x509 -in intermediate.crt
-text -noout
Certificate:
    Data:
    ...

    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Extended Key Usage:
            TLS Web Server Authentication
        X509v3 Key Usage:
            Key Encipherment, Data Encipherment
        X509v3 Subject Alternative Name:
            DNS:abc.company.com, DNS:abc-v6.company.com, IP
Address:1.2.3.4, IP Address:abcd:abcd:abcd:abcd:abcd:abcd

```

ONTAP Mediator 1.8 and earlier

```

[root@mediator_host server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@mediator_host server_config]# cat openssl_server.cnf
...
[ v3_req ]
basicConstraints          = CA:false
extendedKeyUsage          = serverAuth
keyUsage                  = keyEncipherment, dataEncipherment
subjectAltName            = @alt_names

[ alt_names ]
DNS.1 = abc.company.com
DNS.2 = abc-v6.company.com
IP.1 = 1.2.3.4
IP.2 = abcd:abcd:abcd:abcd:abcd:abcd

[root@mediator_host server_config]# openssl x509 -in ca.crt -text
-noout
Certificate:
    Data:

```

...

```
X509v3 extensions:
  X509v3 Basic Constraints:
    CA:FALSE
  X509v3 Extended Key Usage:
    TLS Web Server Authentication
  X509v3 Key Usage:
    Key Encipherment, Data Encipherment
  X509v3 Subject Alternative Name:
    DNS:abc.company.com, DNS:abc-v6.company.com, IP
Address:1.2.3.4, IP Address:abcd:abcd:abcd:abcd:abcd:abcd
```

Verify that a private key matches with a certificate

You can verify whether a particular private key matches with a certificate.

Use the following OpenSSL commands on the key and certificate respectively.

ONTAP Mediator 1.9 and later

```
[root@mediator_host server_config]# openssl rsa -noout -modulus -in
intermediate.key | openssl md5
Enter pass phrase for intermediate.key:
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
[root@mediator_host server_config]# openssl x509 -noout -modulus -in
intermediate.crt | openssl md5
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
```

ONTAP Mediator 1.8 and earlier

```
[root@mediator_host server_config]# openssl rsa -noout -modulus -in
ca.key | openssl md5
Enter pass phrase for ca.key:
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
[root@mediator_host server_config]# openssl x509 -noout -modulus -in
ca.crt | openssl md5
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
```

If the `-modulus` attribute for both match, it indicates that the private key and certificate pair are compatible and can work with each other.

Verify that a server certificate is created from a particular CA certificate

You can use the following command to verify that the server certificate is created from a particular CA certificate.

ONTAP Mediator 1.9 and later

```
[root@mediator_host server_config]# openssl verify -CAfile root_ca.crt
--untrusted intermediate.crt ontap_mediator_server.crt
ontap_mediator_server.crt: OK
[root@mediator_host server_config]#
```

ONTAP Mediator 1.8 and earlier

```
[root@mediator_host server_config]# openssl verify -CAfile ca.crt
ontap_mediator_server.crt
ontap_mediator_server.crt: OK
```

If the Online Certificate Status Protocol (OCSP) validation is being used, use the command [openssl-verify](#).

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.