Understanding the SnapDrive for UNIX daemon

Snapdrive for Unix

NetApp
March 24, 2021

Table of Contents

Understanding the SnapDrive for UNIX daemon

Before you run any SnapDrive for UNIX command, you must understand the web services and daemon and how to use them. All the SnapDrive for UNIX commands work using the daemon service. Before you can use SnapDrive for UNIX on your AIX host, you must start the daemon, which enables SnapDrive for UNIX to integrate seamlessly and securely with other NetApp and non-NetApp products.

What the Web service and daemon are

The SnapDrive for UNIX Web service provides a uniform interface for all the NetApp SnapManager and third-party products to integrate seamlessly with SnapDrive for UNIX. To use command-line interface (CLI) commands in SnapDrive for UNIX, you need to start the daemon.

Various NetApp SnapManager products use the command-line interface (CLI) to communicate with SnapDrive for UNIX. Using the CLI puts a constraint on the performance and manageability of SnapManager and SnapDrive for UNIX. When you use the SnapDrive for UNIX daemon, all the commands work as a unique process. Daemon service does not affect the way SnapDrive for UNIX commands are used.

The SnapDrive for UNIX Web service allows third-party applications to integrate with SnapDrive for UNIX seamlessly. They interact with SnapDrive for UNIX using APIs.

When you start the daemon, SnapDrive for UNIX daemon first checks whether the daemon is running. If the daemon is not running, it starts the daemon. If the daemon is already running and you try to start it, SnapDrive for UNIX displays the message:

```
snapdrive daemon is already running
```

You can check the status of the daemon to see whether SnapDrive for UNIX is running or not. You should check the status before deciding to start the daemon. If a user other than the root user tries to check the status, SnapDrive for UNIX checks the credentials of the user and displays the message:

```
snapdrive daemon status can be seen only by root user
```

When you try to stop the daemon, SnapDrive for UNIX checks your credentials. If you are a user other than root user, SnapDrive for UNIX displays the message

```
snapdrive daemon can be stopped only by root user
```

After you stop the daemon, you must restart the SnapDrive for UNIX daemon for any changes to the configuration file or any module to take effect. If a user other than the root user tries to restart the SnapDrive for UNIX daemon, SnapDrive for UNIX checks the credentials of the user and displays the message

```
snapdrive daemon can be restarted only by root user
```

Checking the status of the daemon

You can check the status of the daemon to see whether the daemon is running. If the daemon is already running, you do not need to restart it until the SnapDrive for UNIX

configuration file has been updated.

You must be logged in as a root user.

Steps

1. Check the status of the daemon:

   ```
   snapdrived status
   ```

Starting the SnapDrive for UNIX daemon

You must start and run the SnapDrive for UNIX daemon before you can use any SnapDrive for UNIX command.

You must be logged in as a root user.

Steps

1. Start the daemon:

   ```
   snapdrived start
   ```

Changing the default daemon password

SnapDrive for UNIX is assigned a default daemon password, which you can change later. This password is stored in an encrypted file with read and write permissions assigned to only the root user. After the password is changed, all the client applications must be notified manually.

You must be logged in as the root user.

Steps

1. Change the default password:

   ```
   snapdrived passwd
   ```

2. Enter the password.
3. Confirm the password.

Stopping the daemon

If you change the SnapDrive for UNIX configuration file, you must stop and restart the daemon. You can stop the daemon nonforcibly or forcibly.

Nonforcibly stopping the daemon

If your SnapDrive for UNIX configuration file is changed, you must stop the daemon for the configuration file changes to take effect. After the daemon is stopped and restarted, the changes in the configuration file take effect. Nonforcibly stopping the daemon allows

all queued commands to complete execution. After the stop request is received, no new commands are executed.

You must be logged in as a root user.

1. Enter the following command to nonforcibly stop the daemon:

   ```
   snapdrived stop
   ```

Forcibly stopping the daemon

You can forcibly stop the daemon when you do not want to wait for all the commands to complete execution. After the request to forcibly stop the daemon is received, the SnapDrive for UNIX daemon cancels any commands that are in execution or in queue. When you forcibly stop the daemon, the state of your system might be undefined. This method is not recommended.

You must be logged in as a root user.

Steps

1. Forcibly stop the daemon:

   ```
   snapdrived -force stop
   ```

Restarting the daemon

You must restart the daemon after you stop it so that changes that you make to the configuration file or to the other modules take effect. The SnapDrive for UNIX daemon restarts only after completing all the commands that are in execution and in queue. After the restart request is received, no new commands are executed.

- Ensure that you are logged in as a root user.
- Ensure that no other sessions are running on the same host in parallel. The `snapdrived restart` command hangs the system in such situations.

Steps

1. Enter the following command to restart the daemon:

   ```
   snapdrived restart
   ```

Forcing daemon restart

You can force the daemon to restart. A forceful restart of the daemon stops the execution of all running commands.

Ensure that you are logged in as a root user.

Steps

1. Enter the following command to forcefully restart the daemon:

   ```
   snapdrived -force restart
   ```

   After the force restart request is received, the daemon stops all the commands in execution and in queue. The daemon is restarted only after cancelling execution of all running commands.

Secure daemon communication using HTTPS

You can use HTTPS for secure Web services and daemon communication. Secure communication is enabled by setting some configuration variables in the `snapdrive.conf` file, and generating and installing the self-signed or CA-signed certificate.

You must provide the self-signed or CA-signed certificate at the path specified in the `snapdrive.conf` file. To use HTTPS for communication, you must set the following parameters in the `snapdrive.conf` file:

- `use-https-to-sdu-daemon=on`
- `contact-https-port-sdu-daemon=4095`
- `sdu-daemon-certificate-path=/opt/NetApp/snapdrive/snapdrive.pem`

SnapDrive 5.0 for UNIX and later versions support HTTPS for daemon communication. By default, the option is set to `off`.

Generating self-signed certificates

The SnapDrive for UNIX daemon service requires that you generate a self-signed certificate for authentication. This authentication is required while communicating with the CLI.

Steps

1. Generate an RSA key:

   ```
   $ openssl genrsa 1024 > host.key $ chmod 400 host.key`
   ```

   ```
   # openssl genrsa 1024 > host.key Generating
   RSA private key, 1024 bit long modulus
   ..............................++++++ ...++++++ e is 65537(0x10001)
   # chmod 400 host.key
   ```

2. Create the certificate:

   ```
   $ openssl req -new -x509 -nodes -sha1 -days 365 -key host.key > host.cert
   ```

   The `-new`, `-x509`, and `-nodes` options are used to create an unencrypted certificate. The `-days` option specifies the number of days the certificate remains valid.

3. When asked to fill out the certificate's x509 data, enter your local data:

```
# openssl req -new -x509 -nodes -sha1 -days 365 -key host.key >
host.cert
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN. There are quite a few fields
but you can leave some blank For some fields there will be a default
value, If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:Sunnyvale
Organization Name (eg, company) [Internet Widgits Pty Ltd]:abc.com
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:localhost
Email Address []:postmaster@example.org
```

The `Common Name` value must be *localhost*.

4. Extract metadata (optional).

```
$ openssl x509 -noout -fingerprint -text < host.cert > host.info
```

You can save the certificate metadata for quick reference later.

5. Combine key and certificate data.

SnapDrive for UNIX requires the key and certificate data to be in the same file. The combined file must be protected as a key file.

```
$ cat host.cert host.key > host.pem \

&& rm host.key

$ chmod 400 host.pem
```

```
# cat host.cert host.key >  /opt/NetApp/snapdrive.pem
# rm host.key rm: remove regular file `host.key'? y
# chmod 400 /opt/NetApp/snapdrive.pem
```

6. Add the complete path of the daemon certificate to the *`sdu-daemon-certificate-path`* variable of the `snapdrive.conf` file.

Generating a CA-signed certificate

The SnapDrive for UNIX daemon service requires that you generate a CA-signed certificate for successful daemon communication. You must provide the CA-signed certificate at the path specified in the `snapdrive.conf` file.

- You must be logged in as a root user.
- You must have set the following parameters in the `snapdrive.conf` file to use HTTPS for communication:
  - use-https-to-sdu-daemon=on
  - contact-https-port-sdu-daemon=4095
  - sdu-daemon-certificate-path=`/opt/NetApp/snapdrive/snapdrive.pem`

Steps

1. Generate a new unencrypted RSA private key in a pem format:

   ```
   $ openssl genrsa -out privkey.pem 1024
   ```

   ```
   Generating RSA private key, 1024 bit long modulus
    ..................++++++ ....................................++++++
   e is 65537 (0x10001)
   ```

2. Configure `/etc/ssl/openssl.cnf` to create the CA private key and the certificate vi `/etc/ssl/openssl.cnf`.
3. Create an unsigned certificate using your RSA private key:

   ```
   $ openssl req -new -x509 -key privkey.pem -out cert.pem
   ```

   ```
   You are about to be asked to enter information that will be
   incorporated into your certificate request.
   What you are about to enter is what is called a Distinguished Name or
   a DN.
   There are quite a few fields but you can leave some blank For some
   fields there will be a default value, If you enter '.', the field
   will be left blank.
   -----
   Country Name (2 letter code) [XX]:NY
   State or Province Name (full name) []:Nebraska Locality Name (eg,
   city) [Default City]:Omaha Organization Name (eg, company) [Default
   Company Ltd]:abc.com Organizational Unit Name (eg, section) []:
   Common Name (eg, your name or your server's hostname) []:localhost
   Email Address []:abc@example.org
   ```

4. Use your private key and your certificate to create a CSR:

```
cat cert.pem privkey.pem | openssl x509 -x509toreq -signkey privkey.pem -out
certreq.csr
```

```
Getting request Private Key Generating certificate request
```

5. Sign the certificate with the CA private key by using the CSR that you have just created:

```
$ openssl ca -in certreq.csr -out newcert.pem
```

```
Using configuration from /etc/pki/tls/openssl.cnf Check that the
request matches the signature Signature ok Certificate Details:
         Serial Number: 4096 (0x1000)
         Validity
            Not Before: May 17 06:02:51 2015 GMT
             Not After : May 16 06:02:51 2016 GMT
             Subject:
             countryName               = NY
             stateOrProvinceName       = Nebraska
             organizationName          = abc.com
             commonName                = localhost
             emailAddress              = abc@example.org
             X509v3 extensions:
             X509v3 Basic Constraints:
                 CA:FALSE
             X509v3 Key Usage:
                 Digital Signature, Non Repudiation, Key Encipherment
             Netscape Comment:
                 OpenSSL Generated Certificate
             X509v3 Subject Key Identifier:

FB:B0:F6:A0:9B:F2:C2:BC:50:BF:45:B2:9D:DB:AA:3B:C5:07:5B:7F
             X509v3 Authority Key Identifier:

 keyid:FB:B0:F6:A0:9B:F2:C2:BC:50:BF:45:B2:9D:DB:AA:3B:C5:07:5B:7F

 Certificate is to be certified until May 16 06:02:51 2016 GMT (365
 days) Sign the certificate? [y/n]:y


 1 out of 1 certificate requests certified, commit? [y/n]y Write out
 database with 1 new entries Data Base Updated
```

6. Install the signed certificate and the private key to be used by an SSL server.

```
The newcert.pem is the certificate signed by your local CA that you can
then use in an
ssl server:
( openssl x509 -in newcert.pem; cat privkey.pem ) > server.pem
ln -s server.pem `openssl x509 -hash -noout -in server.pem`.0 # dot-zero
( server.pem refers to location of https server certificate)
```