Role-based access control in SnapDrive for UNIX

Snapdrive for Unix

NetApp
March 24, 2021

Table of Contents

Role-based access control in SnapDrive for UNIX

Role-based access control (RBAC) is used for user login and role permissions. RBAC allows administrators to manage groups of users by defining roles. If you need to restrict access to the database to specific administrators, you must set up administrator accounts for them. Additionally, if you want to restrict the information, these administrators can view, and the operations they can perform, you must apply roles to the administrator accounts you create.

RBAC is used in SnapDrive for UNIX with the help of Operations Manager console. Operations Manager console provides granular access to storage objects such as LUNs, qtrees, volumes, aggregates, and vFiler units.

Related information

Mandatory checks for volume-based SnapRestore

Restoring Snapshot copies on a destination storage system

Snap disconnect procedure

What role-based access control (RBAC) in SnapDrive for UNIX is

RBAC allows SnapDrive administrators to restrict access to a storage system for various SnapDrive operations. This limited or full access for storage operations depends on the role that is assigned to the user.

SnapDrive 4.0 for UNIX and later requires an RBAC access check for all the SnapDrive for UNIX operations. This behavior allows the storage administrators to limit the operations that SnapDrive users can perform depending on their assigned roles. RBAC is implemented using the Operations Manager infrastructure. In releases earlier than SnapDrive 4.0 for UNIX, there was limited access control and only the root user could perform SnapDrive for UNIX operations. SnapDrive 4.0 for UNIX and later provides support for nonroot local users and Network Information System (NIS) users by using the RBAC infrastructure of Operations Manager console. SnapDrive for UNIX does not require the root password of the storage system; it communicates with the storage system using sd-<hostname> user.

By default, Operations Manager console RBAC functionality is not used. You must turn on RBAC functionality by setting the variable *rbac-method=dfm* in the `snapdrive.conf` file and restart the SnapDrive for UNIX daemon.

The following requirements must be fulfilled before you can use this feature:

- Operations Manager console 3.7 or later.
- Operations Manager console server must be present and configured in the IP network that contains the SnapDrive hosts and the storage systems.
- Operations Manager console communication settings must be configured during SnapDrive installation.
- SnapDrive for UNIX daemon should be running.

SnapDrive for UNIX and Operations Manager console interaction

Use of Role-based access control (RBAC) depends on the Operations Manager console infrastructure. The Operations Manager console administrator must create user names for SnapDrive for UNIX use. All storage operation requests are first sent to Operations Manager console for an access check. After Operations Manager console verifies a storage operation from a specific SnapDrive user, the operation is completed.

The following diagram illustrates the entire RBAC for storage operations.

[sdu rbac process 01 aix] | *../media/sdu_rbac_process_01_aix.gif*

1. Operations Manager console administrator adds sd-admin user on Operations Manager console.
2. Operations Manager console administrator creates sd-hostname user on the storage system.
3. Operations Manager console administrator sends sd-admin and sd-hostname credentials to SnapDrive for UNIX administrator.
4. SnapDrive administrator configures SnapDrive with the received user credentials.
5. Operations Manager console performs access check for SnapDrive for UNIX use with the user credentials added by SnapDrive administrator.
6. After the SnapDrive user is authenticated, the user can connect to the storage system.

When a SnapDrive user wants to carry out some storage operation, the user issues the corresponding command at the command line. The request is sent to Operations Manager console for an access check. Operations Manager console checks whether the requested user has the appropriate permissions to carry out the SnapDrive operation. The result of the access check is returned to SnapDrive. Depending on the result, the user is allowed or not allowed to carry out the storage operations on the storage system.

If the user is verified after the access check, the user connects to the storage system as sd-hostname.

sd-hostname and sd-admin are the recommended user names. You can configure SnapDrive for UNIX with other user names.

Configuration of role-based access control in SnapDrive for UNIX

You must complete various tasks to configure Role-Based Access Control (RBAC) for SnapDrive for UNIX. You can use either Operations Manager console or the command-line interface to perform the tasks.

Configuring sd-admin in Operations Manager console

The Operations Manager console administrator can create the sd-admin user.

The Operations Manager console administrator creates a user named, sd-admin, with the capability to perform a core access check on global group (global `DFM.Core.AccessCheck`). After the Operations Manager console administrator configures the sd-admin user, you must manually send the credential information to the

SnapDrive for UNIX administrator. For more information about using Operations Manager console to configure users and roles, see the *Operations Manager Console Administration guide* and the Online Help.

You can use any name in place of sd-admin; however, it is best to use sd-admin.

To create a role in Operations Manager console, select **Setup** > **Roles**. In the sd-admin configuration page, the Operations Manager console administrator must assign `DFM.Database.Write` capability on the global group to sd-admin-role, so that SnapDrive for UNIX can refresh storage entities in Operations Manager console.

Configuring sd-admin using command-line interface

The storage system administrator can configure sd-admin user using command-line interface.

Steps

1. Add a user named sd-admin.

```
# useradd sd-admin
```

```
# passwd sd-admin
Changing password for sd-admin.
New password:
Re-enter new password:
Password changed
```

2. Add an administrator named sd-admin.

```
# dfm user add sd-admin
Added administrator sd-admin.
```

3. Create a role named sd-admin-role.

```
# dfm role create sd-admin-role
Created role sd-admin-role.
```

4. Add a capability to the role created in step 3.

```
# dfm role add sd-admin-role DFM.Core.AccessCheck Global
Added 1 capability to role sd-admin-role.
```

5. The Operations Manager administrator can also grant `DFM.Database.Write` capability on the global group to `<sd-admin>` to enable SnapDrive for UNIX to refresh storage system entities in Operations Manager.

```
# dfm role add sd-admin-role DFM.Database.Write Global
Added 1 capability to role sd-admin-role.
```

6. Add an sd-admin-role role to the sd-admin user.

```
# dfm user role set sd-admin sd-admin-role
Set 1 role for administrator sd-admin.
```

Adding sd-hostname to the storage system

The Operations Manager console administrator can create the sd-hostname user on the storage system using Operations Manager console. After the steps are completed, the Operations Manager console administrator must manually send the credentials to the SnapDrive for UNIX administrator. You can use any name in place of sd-hostname; however it is best to use sd-hostname.

Steps

1. Obtain the root password of the storage system and store the password.

   To add the password for the storage system, select **Management** > **Storage System**.

2. Create an sd-hostname user for each UNIX system.
3. Assign capabilities `api- and login-` to a role, such as sd-role.
4. Include this role (sd-role) in a new usergroup, such as sd-usergroup.
5. Associate this usergroup (sd-usergroup) with the sd-hostname user on the storage system.

Adding sd- hostname to storage system using CLI

The storage system administrator can create and configure the sd-hostname user using the useradmin command.

Steps

1. Add storage.

```
# dfm host add storage_array1
Added host storage_array1.lab.eng.btc.xyz.in
```

2. Set the password for the host.

```
# dfm host password save -u root -p xxxxxxxx storage_array1
Changed login for host storage_array1.lab.eng.btc.xyz.in to root.
Changed Password for host storage_array1.lab.eng.xyz.netapp
.in
```

3. Create a role on the host.

```
# dfm host role create -h storage_array1 -c "api-*,login-*" sd-unixhost-
role
Created role sd-unixhost-role on storage_array1
```

4. Create a usergroup.

```
# dfm host usergroup create -h storage_array1 -r sd-unixhost-role sd-
unixhost-ug
Created usergroup sd-unixhost-ug(44) on storage_array1
```

5. Create a local user.

```
# dfm host user create -h storage_array1 -p xxxxxxxx -g sd-unixhost-ug
sd-unixhost
Created local user sd-unixhost on storage_array1
```

Configuring user credentials on SnapDrive for UNIX

The SnapDrive for UNIX administrator receives user credentials from Operations Manager console administrator. These user credentials need to be configured on SnapDrive for UNIX for proper storage operations.

Steps

1. Configure sd-admin on the storage system.

```
[root]#snapdrive config set -dfm sd-admin ops_mngr_server
Password for sd-admin:
Retype password:
```

2. Configure sd-hostname on the storage system.

```
[root]#snapdrive config set sd-unix_host storage_array1
Password for sd-unix_host:
Retype password:
```

3. Verify step 1 and step 2, using the `snapdrive config list` command.

```
user name       appliance name          appliance type
----------------------------------------------------
sd-admin        ops_mngr_server         DFM
sd-unix_host    storage_array1          StorageSystem
```

4. Configure SnapDrive for UNIX to use Operations Manager console Role-based access control (RBAC) by setting the configuration variable `rbac-method="dfm"` in the `snapdrive.conf` file.

The user credentials are encrypted and saved in the existing `.sdupw` file. The default location of the earlier file is `/opt/NetApp/snapdrive/.sdupw`.

User name formats for performing access checks with Operations Manager console

SnapDrive for UNIX uses the user name formats for performing access checks with Operations Manager console. These formats depends on whether you are a Network Information System (NIS) or a local user.

SnapDrive for UNIX uses the following formats to check whether a user is authorized to perform certain tasks:

- If you are an NIS user running the `snapdrive` command, SnapDrive for UNIX uses the format `<nisdomain>\<username>` (for example, `netapp.com\marc`)
- If you are a local user of a UNIX host such as lnx197-141, SnapDrive for UNIX uses the format `<hostname>\<username>` format (for example, `lnx197-141\john`)
- If you are an administrator (root) of a UNIX host, SnapDrive for UNIX always treats the administrator as a local user and uses the format `lnx197-141\root`.

Configuration variables for role-based access control

You must set the various configuration variables related to role-based access control in the `snapdrive.conf` file.

Variable	Description
contact-http-dfm-port = 8088	Specifies the HTTP port to use for communicating with an Operations Manager console server. The default value is 8088.

Variable	Description
contact-ssl-dfm-port = 8488	Specifies the SSL port to use for communicating with an Operations Manager console server. The default value is 8488.
rbac-method=dfm	Specifies the access control methods. The possible values are `native` and `dfm`. If the value is `native`, the access control file stored in `/vol/vol0/sdprbac/sdhost-name.prbac` is used for access checks. If the value is set to `dfm`, Operations Manager console is a prerequisite. In such a case, SnapDrive for UNIX sends access checks to the Operations Manager console.
rbac-cache=on	SnapDrive for UNIX maintains a cache of access check queries and the corresponding results. SnapDrive for UNIX uses this cache only when all the configured Operations Manager console servers are down. You can set this value to either `on` to enable cache, or to `off` to disable it. The default value is off so that you can configure SnapDrive for UNIX to use Operations Manager console and set the `rbac-method` configuration variable to `dfm`.
rbac-cache-timeout	Specifies the rbac cache timeout period and is applicable only when the *rbac-cache* is enabled. The default value is `24` hrs. SnapDrive for UNIX uses this cache only when all the configured Operations Manager console servers are down.
use-https-to-dfm=on	This variable lets you set SnapDrive for UNIX to use SSL encryption (HTTPS) when it communicates with Operations Manager console. The default value is `on`.

SnapDrive commands and capabilities

In role-based access control (RBAC), a specific capability is required for each operation to be successful. A user must have the correct set of capabilities assigned to carry out storage operations.

The following table lists the commands and the corresponding capabilities required:

Command	Capability
`storage show`	SD.Storage.Read on volume
`storage list`	SD.Storage.Read on volume
`storage create`	• For LUNs inside volumes: `SD.Storage.Write` on Volume • For LUNs inside qtrees: `SD.Storage.Write` on qtree
`storage resize`	`SD.Storage.Write` on LUN
`storage delete`	`SD.Storage.Delete` on LUN
`snap show`	`SD.SnapShot.Read` on volume
`snap list`	`SD.SnapShot.Read` on volume
`snap delete`	`SD.Storage.Delete` on volume
`snap rename`	`SD.Storage.Write` on volume
`snap connect`	• For LUN clones in volume: `SD.SnapShot.Clone` on volume • For LUN clones in qtree: `SD.SnapShot.Clone` on qtree • For traditional volume clones: `SD.SnapShot.Clone` on storage system • For FlexClone volume: `SD.SnapShot.Clone` on the parent volume • For unrestricted Flexclone volumes: `SD.SnapShot.UnrestrictedClone` on the parent volume

Command	Capability
`snap connect-split`	• For LUN clones (LUN cloned and split in volume): `SD.SnapShot.Clone` on volume and `SD.Storage.Write` on volume • For LUN clones (LUN cloned and split in qtree): `SD.SnapShot.Clone` on qtree and `SD.Storage.Write` on qtree • For traditional volume clones which are split: `SD.SnapShot.Clone` on storage system and `SD.Storage.Write` on storage system • For Flex volume clones which are split: `SD.SnapShot.Clone` on the parent volume.
`clone split start`	• For LUN clones where the LUN resides in volume or qtree: `SD.SnapShot.Clone` containing volume or qtree • For volume clones: `SD.SnapShot.Clone` on the parent volume
`snap disconnect`	• For LUN clones where the LUN resides in volume or qtree: `SD.SnapShot.Clone` containing volume or qtree • For volume clones: `SD.SnapShot.Clone` on the parent volume • For deletion of unrestricted volume clones: `SD.SnapShot.DestroyUnrestrictedClone` on the volume
`snap disconnect-split`	• For LUN clones where the LUN resides in volume or qtree: `SD.SnapShot.Clone` on the containing volume or qtree • For volume clones: `SD.Storage.Delete` on the parent volume • For deletion of unrestricted volume clones: `SD.SnapShot.DestroyUnrestrictedClone` on the volume

Command	Capability
`snap restore`	• For LUNs that exist in a volume: `SD.SnapShot.Restore` on volume and `SD.Storage.Write` on LUN • For LUNs which exists in a qtree: `SD.SnapShot.Restore` on qtree and `SD.Storage.Write` on LUN • For LUNs which are not in the volumes: `SD.SnapShot.Restore` on volume and `SD.Storage.Write` on volume • For LUNs which are not in qtree: `SD.SnapShot.Restore` on qtree and `SD.Storage.Write` on qtree • For volumes: `SD.SnapShot.Restore` on storage system for traditional volumes, or `SD.SnapShot.Restore` on aggregate for flexible volumes • For single-file snap restore in volumes: `SD.SnapShot.Restore` on the volume • For single-file snap restore in qtree: `SD.SnapShot.Restore` qtree • For overriding baseline Snapshot copies: `SD.SnapShot.DisruptBaseline` on the volume
`host connect`, `host disconnect`	`SD.Config.Write` on the LUN
`config access`	`SD.Config.Read` on the storage system
`config prepare`	`SD.Config.Write` on at least one storage system
`config check`	`SD.Config.Read` on at least one storage system
`config show`	`SD.Config.Read` on at least one storage system
`config set`	`SD.Config.Write` on storage system
`config set -dfm`, `config set -mgmtpath`,	`SD.Config.Write` on at least one storage system
`config delete`	`SD.Config.Delete` on storage system
`config delete dfm_appliance`, `config delete -mgmtpath`	`SD.Config.Delete` on at least one storage system

Command	Capability
`config list`	`SD.Config.Read` on at least one storage system
`config migrate set`	`SD.Config.Write` on at least one storage system
`config migrate delete`	`SD.Config.Delete` on at least one storage system
`config migrate list`	`SD.Config.Read` on at least one storage system

SnapDrive for UNIX does not check any capability for administrator (root).

Preconfigured roles for ease of user role configuration

Preconfigured roles simplify the task of assigning roles to users.

The following table lists the predefined roles:

Role Name	Description
GlobalSDStorage	Manage storage with SnapDrive for UNIX
GlobalSDConfig	Manage configurations with SnapDrive for UNIX
GlobalSDSnapshot	Manage Snapshot copies with SnapDrive for UNIX
GlobalSDFullControl	Full use of SnapDrive for UNIX

In the preceding table, Global refers to all the storage systems managed by an Operations Manager console.

Automatic storage system update on Operations Manager console

Operations Manager console discovers the storage systems supported on your network. It periodically monitors data that it collects from the discovered storage systems. The data is refreshed at a set interval. The Operations Manager console administrator can configure the refresh interval.

LUN monitoring Interval, qtree monitoring Interval, and vFiler monitoring interval are important fields that decide the frequency of LUN, qtree, and vFiler updates. For example, if a new LUN is created on a storage system, the new LUN is not immediately updated on Operations Manager console. For this reason, and access check issued to Operations Manager console for that LUN to Operations Manager console fails. To avoid this situation, you can modify the LUN monitoring interval to suit your requirements.

1. Select **Setup** > **Options** in Operations Manager console to change the monitoring interval.

2. The Operations Manager console administrator can also forcefully refresh Operations Manager console by executing `dfm host discovery filername` in the command-line interface.
3. The Operations Manager console administrator can also grant `DFM.Database.Write` capability on the global group to sd-admin to enable SnapDrive for UNIX to refresh storage system entities on Operations Manager console.

```
# dfm role add sd-admin-role DFM.Database.Write Global
Added 1 capability to role sd-admin-role.
```

Multiple Operations Manager console servers

SnapDrive for UNIX supports multiple Operations Manager console servers. This feature is required when a group of storage systems is managed by more than one Operations Manager console server. SnapDrive for UNIX contacts the Operations Manager console servers in the same order that the Operations Manager console servers are configured in SnapDrive for UNIX. You can run the `snapdrive config list` command to obtain the configuration order.

The following example shows output for multiple Operations Manager console servers:

```
# snapdrive config list
username    appliance name     appliance type
----------------------------------------------
root        storage_array1     StorageSystem
root        storage_array2     StorageSystem
sd-admin    ops_mngr_server1   DFM
sd-admin    ops_mngr_server2   DFM
```

In the preceding example, storage_array1 is managed by ops_mngr_server1 and storage_array2 is managed by ops_mngr_server2. In this example, SnapDrive for UNIX contacts ops_mngr_server1 first. If ops_mngr_server1 is not able to determine access, SnapDrive for UNIX contacts ops_mngr_server2.

SnapDrive for UNIX contacts the second Operations Manager console only under the following conditions:

- When the first Operations Manager console is unable to determine access. This situation might occur because the first Operations Manager console is not managing the storage system.
- When the first Operations Manager console is down.

Operations Manager console unavailable

SnapDrive for UNIX needs Operations Manager console for access checks. Sometimes Operations Manager console server might not be available for various reasons.

When the RBAC method *`rbac-method = dfm`* is set and Operations Manager console is not available, SnapDrive for UNIX displays the following error message:

```
[root]# snapdrive storage delete -lun storage_array1:/vol/vol2/qtree1/lun1
0002-333 Admin error: Unable to connect to the DFM ops_mngr_server
```

SnapDrive for UNIX can also maintain a cache of the user access check results returned by Operations Manager console. This cache is valid for 24 hours and is not configurable. If Operations Manager console is not available then SnapDrive for UNIX uses the cache to determine access. This cache is used only when all the configured Operations Manager console servers do not respond.

For SnapDrive for UNIX to use the cache for an access check, you must turn on the *rbac-cache* configuration variable must be turned on to maintain the cache of access results. The *rbac-cache* configuration variable is off by default.

To use SnapDrive for UNIX even when Operations Manager console is not available, the server administrator must reset the role-based access control (RBAC) method to *rbac-method = native* in the `snapdrive.conf` file. After you change the `snapdrive.conf` file, you must restart the SnapDrive for UNIX daemon. When *rbac-method = native* is set, only root user can use SnapDrive for UNIX.

RBAC and storage operation examples

Role-based access control allows storage operations depending on the capabilities assigned to you. You receive an error message if you do not have the right capabilities to carry out the storage operation.

Operation with a single filespec on a single storage object

SnapDrive for UNIX displays an error message when you are not an authorized user to create a filespec on a specified volume.

Filespec: Filespec can be a file system, host volume, disk group, or LUN.

```
[john]$ snapdrive storage create -fs /mnt/testfs -filervol
storage_array1:/vol/vol1 -dgsize 100m
0002-332 Admin error:SD.Storage.Write access denied on volume
storage_array1:/vol/vol1 for user unix_host\john on Operations Manager
server ops_mngr_server
```

In this example, John is a nonroot user and is not authorized to create a filespec on the specified volume. John must ask the Operations Manager console administrator to grant `SD.Storage.Write` access on the volume `storage_array1:/vol/vol1`.

Operation with a single filespec on multiple storage objects

SnapDrive for UNIX displays an error message when the administrator does not have the required permission on multiple storage objects to carry out the storage operations.

Filespec: Filespec can be anyone of file system, host volume, disk group, or LUN

```
[root]# snapdrive storage create -fs /mnt/testfs -lun
storage_array1:/vol/vol1/lun2 -lun storage_array1:/vol/vol2/lun2  -lunsize
100m
0002-332 Admin error:SD.Storage.Write access denied on volume
storage_array1:/vol/vol1 for user unix_host\root on Operations Manager
server ops_mngr_server
SD.Storage.Write access denied on volume storage_array1:/vol/vol2 for user
unix_host\root on Operations Manager server ops_mngr_server
```

In this example the filespec spans over two storage system volumes, vol1 and vol2. The administrator (root) of unix_host does not have `SD.Storage.Write` access on both volumes. Therefore, SnapDrive for UNIX shows one error message for each volume. To proceed with `storage create`, the administrator (root) must ask the Operations Manager console administrator to grant `SD.Storage.Write` access on both the volumes.

Operation with multiple filespec and storage objects

The following example shows the error message you would receive when you are not an authorized user to carry out the specific operation.

```
[marc]$ snapdrive storage create -lun storage_array1:/vol/vol1/lun5 lun6
-lun storage_array1:/vol/vol2/lun2 -lunsize 100m
0002-332 Admin error:SD.Storage.Write access denied on volume
storage_array1:/vol/vol1 for user nis_domain\marc on Operations Manager
server ops_mngr_server
SD.Storage.Write access denied on volume storage_array1:/vol/vol2 for user
nis_domain\marc on Operations Manager server ops_mngr_server
```

In this example, three LUNs reside on two storage system volume, vol1 and vol2. User Marc belongs to nis_domain and is not authorized to create filespec on vol1 and vol2. SnapDrive for UNIX displays the two error messages in the preceding example. The error messages show that the user must have `SD.Storage.Write` access on vol1 and vol2.

Operation with multiple storage objects

The following example shows the error message you would receive when you are not an authorized user to carry out the specific operation.

```
[john]$ snapdrive storage show -all

Connected LUNs and devices:

device filename   adapter  path  size  proto  state  clone  lun path
backing Snapshot
---------------  -------  ----  ----  -----  -----  -----  --------
---------------
/dev/sdao            -     -     200m  iscsi  online  No
storage_array1:/vol/vol2/passlun1           -
/dev/sda1            -     -     200m  fcp    online  No
storage_array1:/vol/vol2/passlun2           -

Host devices and file systems:

dg: testfs1_SdDg                dgtype lvm
hostvol: /dev/mapper/testfs1_SdDg-testfs1_SdHv  state: AVAIL
fs: /dev/mapper/testfs1_SdDg-testfs1_SdHv       mount point: /mnt/testfs1
(persistent) fstype jfs2

device filename   adapter  path  size  proto  state   clone  lun path
backing Snapshot
---------------  -------  ----  ----  -----  -----   -----  --------
---------
/dev/sdn             -      P    108m  iscsi  online   No
storage_array1:/vol/vol2/testfs1_SdLun         -
/dev/sdn1             -      P    108m  fcp  online   No
storage_array1:/vol/vol2/testfs1_SdLun1         -

0002-719 Warning: SD.Storage.Read access denied on volume
storage_array1:/vol/vol1 for user unix_host\john on Operations Manager
server ops_mngr_server
```

John is authorized to list storage entities on vol2 but not on vol1. SnapDrive for UNIX displays entities of vol1 and displays a warning message for vol2.

For `storage list`, `storage show`, `snap list`, and `snap show` commands SnapDrive for UNIX displays a warning instead of error.

Operation with multiple Operations Manager console servers managing storage systems

The following output shows the error message you would receive when storage systems are managed by multiple Operations Managers console.

```
[root]# snapdrive storage create -lun storage_array1:/vol/vol1/lun5 lun6
-lun storage_array2:/vol/vol1/lun2 -lunsize 100m
0002-332 Admin error:SD.Storage.Write access denied on volume
storage_array1:/vol/vol1 for user unix_host\root on Operations Manager
server ops_mngr_server1
SD.Storage.Write access denied on volume storage_array2:/vol/vol1 for user
unix_host\root on Operations Manager server ops_mngr_server2
```

storage_array1 is managed by ops_mngr_server1 and storage_array2 is managed by ops_mngr_server2. Administrator of unix_host is not authorized to create filespecs on storage_array1 and storage_array2. In the preceding example SnapDrive for UNIX displays the Operations Manager console used to determine access.