

## KumoScale Ansible Playbooks

This section describes the KumoScale Ansible playbooks available and how to use them to perform various functions.

Ansible playbooks can be created using one or more of the modules documented in [KumoScale Ansible Modules](#). The **examples** directory contains a large variety of example playbooks that can be used as-is, or as a reference for creating your own. Below is a table listing the playbooks defined in the file **examples/***<playbook name>*.yml with a link to more information on the playbook.

KumoScale Software Example Playbooks	
Playbook	Description
<a href="#">ks_connect</a>	Automates the process supported by the <a href="#">ks_connect module</a> for connecting or disconnecting an initiator to a target as a block device.
<a href="#">ks_info</a>	Performs the process supported by the <a href="#">ks_info module</a> for getting information about KumoScale Provisioner entities (system, host, volume).
<a href="#">ks_mirror</a>	Automates the process supported by <a href="#">ks_mirror module</a> for creating or deleting the resiliency configuration (replication) from the attached volumes.
<a href="#">ks_publish</a>	Automates the process supported by the <a href="#">ks_publish module</a> for publishing or unpublishing volumes.
<a href="#">ks_replica</a>	Automates the process supported by the <a href="#">ks_replica module</a> for adding or removing a replica to a resilient volume.
<a href="#">ks_snapshot</a>	Automates the process supported by the <a href="#">ks_snapshot module</a> for creating or deleting a snapshot of a volume.
<a href="#">ks_token</a>	Automates the process supported by the <a href="#">ks_token module</a> for generating a token using the current authorization mode.
<a href="#">ks_volume</a>	Automates the process supported by the <a href="#">ks_volume module</a> for creating, deleting, or expanding volumes.
<a href="#">create_snapshot_volume</a>	Creates a volume from a snapshot.
<a href="#">create_volume_replicated</a>	Creates one or more resilient volumes, creates a file system over them, mounts them, and connects them to a host via a target.
<a href="#">create_volume_simple</a>	Creates one or more non-resilient volumes, creates a file system over them, mounts them, and connects them to an initiator via a target.
<a href="#">delete_snapshot_volume</a>	Deletes a volume of a snapshot.
<a href="#">delete_volume_replicated</a>	Deletes one or more resilient volumes (it assumes that the volumes are connected to initiators and have a mounted file system).
<a href="#">delete_volume_simple</a>	Deletes one or more non-resilient volumes (it assumes that the volumes are connected to initiators and have a mounted file system).
<a href="#">multiattach_replicated</a>	Attaches or detaches an existing replicated volume to or from additional hosts.
<a href="#">multiattach_simple</a>	Attaches or detaches an existing simple volume to or from additional hosts.
<a href="#">install_host</a>	Prepares the initiator to run Ansible and installs the KumoScale agent.
<a href="#">uninstall_host</a>	Uninstalls the KumoScale agent from the initiator.

### Using Playbooks

To run a playbook, use the command **ansible-playbook** as documented at <https://docs.ansible.com>. Examples of how to use this command with KumoScale playbooks are provided in subsequent sections. The playbooks retrieve their input parameters from the command line and from the variable file, **vars.yml**, documented in [The Ansible Variable File](#). The playbooks check input parameters for validity and compliance with the KumoScale configuration. An example vars.yml file can be found in **examples**, the same subdirectory as the playbooks.

Every playbook accepts initiator (host) names from the following sources:

- Ansible configuration file, **/etc/ansible/hosts**, modified at installation time.
- Variable declaration file, **yml**, under hosts\_group.
- External parameters (-limit option at the command line or corresponding parameter)

The next sections detail how to use each playbook so that you can better understand when and how to use them in your deployment then show examples of how you can use them together to perform more complex functions.

## Example Playbooks

This section describes the sequence of events in the playbooks that support volume management.

### Get Information on an Entity with ks\_info

**Description:** Automates the process corresponding to the ks\_info module for getting information on any entity (host, system, volume).

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

#### Modules Used by the Playbook

- ks\_token
- ks\_info

#### Example

To get information on the entity set in vars.yml:

```
ansible-playbook -v ks_info.yml
```

### Generate a Token with ks\_token

**Description:** Automates the process corresponding to the ks\_token which generates a token using the current authentication method.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

#### Modules Used by the Playbook

- ks\_token
- ks\_info

#### Example

To generate a token using parameters defined in vars.yml:

```
ansible-playbook -v ks_token.yml
```

### Create, Expand, or Delete Volumes with ks\_volume

**Description:** Automates the process corresponding to the ks\_volume module which creates, deletes, or expands volumes or snapshot volumes.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

#### Modules Used by the Playbook

- ks\_token
- ks\_info
- ks\_volume

#### Examples

To create a new, or expand an existing volume using parameters defined in vars.yml:

```
ansible-playbook -v ks_volume.yml -e state=present
```

To delete a volume using parameters defined in vars.yml:

```
ansible-playbook -v ks_volume.yml -e state=absent
```

### Publish or Unpublish Volumes with ks\_publish

**Description:** Automates the process corresponding to the ks\_publish module which creates a target for a specific initiator (if none exists), attaches a volume as a namespace to it, and grants access to the initiator.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

#### Modules Used by the Playbook

- ks\_token
- ks\_info
- ks\_publish

#### Examples

To publish an existing volume using parameters defined in vars.yml:

```
ansible-playbook -v ks_publish.yml -e state=present
```

To unpublish a volume using parameters defined in vars.yml:

```
ansible-playbook -v ks_publish.yml -e state=absent
```

### Connect or Disconnect Volumes with ks\_connect

**Description:** Automates the process corresponding to the ks\_connect module which creates a target for a specific initiator (if none exists), attaches a volume as a namespace to it, and grants access to the initiator.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

**Modules Used by the Playbook**

- ks\_token
- ks\_info
- ks\_connect

**Examples**

To connect an existing volume to an initiator defined in vars.yml:

```
ansible-playbook -v ks_connect.yml -e state=present
```

To disconnect an existing volume from an initiator defined in vars.yml:

```
ansible-playbook -v ks_connect.yml -e state=absent
```

**Create or Break Mirroring with ks\_mirror**

**Description:** Automates the process corresponding to the ks\_mirror module which creates the resiliency configuration (replication) from the attached volumes.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

**Modules Used by the Playbook**

- ks\_token
- ks\_info
- ks\_mirror

**Examples**

To create the mdadm mirror on an initiator specified in vars.yml:

```
ansible-playbook -v ks_mirror.yml -e state=present
```

To delete the mdadm mirror on an initiator specified in vars.yml:

```
ansible-playbook -v ks_mirror.yml -e state=absent
```

**Add or Remove a Volume with ks\_replica**

**Description:** Automates the process corresponding to the ks\_replica module which adds or removes a replica to a resilient volume.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

**Modules Used by the Playbook**

- ks\_token
- ks\_info
- ks\_replica

**Examples**

To add a replica to the Provisioner with volume uuid =123e4567-####-####-####-#####440000:

```
ansible-playbook -v ks_replica.yml -e vol_uuid==123e4567-####-####-####-#####440000 state=present
```

To delete the replica with uuid 9876g123-####-####-####-#####099999 from the Provisioner:

```
ansible-playbook -v ks_replica.yml -e vol_uuid=123e4567-####-####-####-#####440000 -e replica_uuid=9876g123-####-####-####-#####099999 -e state=absent
```

**Create and Delete Snapshots with ks\_snapshot**

**Description:** Automates the process corresponding to the ks\_snapshot module which creates a snapshot of a volume or deletes one.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

**Modules Used by the Playbook**

- ks\_token
- ks\_info
- ks\_snapshot

**Examples**

To create a snapshot defined in vars.yml:

```
ansible-playbook -v ks_snapshot.yml -e state=present
```

To delete a snapshot specified in vars.yml:

```
ansible-playbook -v ks_snapshot.yml -e state=absent
```

Create a Replicated Volume with create\_volume\_replicated

**Description:** Provisions a resilient volume by executing the following steps.

1. Create an authentication token.

2. Create a replicated volume.

3. Publish the replicated volume to the corresponding initiator.

4. Connect the replicated volume to all specified initiators.
- If the volume already exists, the operation stops with an informative message.

• The number of defined initiators (hosts) must be more than 1.

• Each step (except for generating a token) is repeated according to the number of replicas defined for the volume.

• All mounted volumes should be recovered after booting.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

Modules Used by the Playbook

- ks\_token

• ks\_info

• ks\_volume

• ks\_publish

• ks\_connect

• *filesystem* (built-in Ansible module)

• *mount* (built-in Ansible module)

To create a replicated volume specified in vars.yml:

```
ansible-playbook -v create_volume_replicated.yml
```

Delete a Replicated Volume with delete\_volume\_replicated

**Description:** Deletes a resilient volume configuration, by executing the following steps:

1. Generate an authentication token.

2. Stop mirroring the corresponding replicated volume.

3. Disconnect the corresponding replicated volume from the corresponding initiator.

4. Unpublish the corresponding replicated volume from the corresponding initiator.

5. Delete the corresponding replicated volume.
- If the volume being deleted is mounted the operation stops with an informative message.

• The number of defined initiators (hosts) must be more than 1.

• Each step (except for generating a token) is repeated according to the number of replicas defined for the volume.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

Modules Used by the Playbook

- ks\_token

• ks\_info

• ks\_volume

• ks\_publish

• ks\_mirror

To delete the volume specified in vars.yml:

```
ansible-playbook -v delete_volume_replicated.yml
```

Create a Simple Volume using create\_volume\_simple

**Description:** Provisions a non-resilient volume by executing the following steps:

1. Generate an authentication token.

2. Create a simple volume.

3. Publish the simple volume to the corresponding initiator.

4. Connect the simple volume to the corresponding initiator.
- If the volume being created already exists, the operation stops with an informative message.

• The number of defined initiators (hosts) must be equal to 1.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

Modules Used by the Playbook

- ks\_token

• ks\_info

• ks\_volume

- `ks_publish`
- `ks_connect`
- *filesystem* (built-in Ansible module)
- *mount* (built-in Ansible module)

**Example**

To create a simple volume on the initiator myhost:

```
ansible-playbook -v create_volume_simple.yml --limit myhost
```

**Delete a Simple Volume using `delete_volume_simple`**

**Description:** Deletes the configuration of a non-resilient volume by executing the following steps:

1. Generate an authentication token.
  2. Disconnect the corresponding simple volume from the corresponding initiator.
  3. Unpublish the corresponding simple volume from the corresponding initiator
  4. Delete the corresponding simple volume.
- If the simple volume is mounted, the operation stops with an informative message.
  - The number of defined hosts must be 1.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

**Modules Used by the Playbook**

- `ks_token`
- `ks_info`
- `ks_volume`
- `ks_publish`
- `ks_connect`
- *filesystem* (built-in Ansible module)
- *mount* (built-in Ansible module)

**Example**

To delete a simple volume from an initiator with the name myhost:

```
ansible-playbook -v delete_volume_simple.yml --limit myhost
```

**Attach or detach a replicated volume using `multiattach_replicated`**

**Description:** Attaches or detaches an existing replicated volume to or from initiators by executing the following steps:

Attach (state=Present)

1. Generate an authentication token.
2. Publish an existing replicated volume to the additional initiator.
3. Connect an existing replicated volume to the additional initiator.

Detach (state=Absent)

1. Generate an authentication token.
  2. Disconnect an existing replicated volume from the corresponding initiator.
  3. Unpublish an existing replicated volume from the corresponding initiator.
- If the replicated volume does not exist, the operation stops with an informative message.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

**Modules Used by the Playbook**

- `ks_token`
- `ks_info`
- `ks_publish`
- `ks_connect`
- *filesystem* (built-in Ansible module)
- *mount* (built-in Ansible module)

**Example**

To create a multiattached replicated volume (full flow)

```
ansible-playbook -v multiattach_replicated.yml -e state=present
```

To delete a multiattached replicated volume (full flow)

```
ansible-playbook -v multiattach_replicated.yml -e state=absent
```

Attach or detach a simple volume using multiattach\_simple

**Description:** Attaches or detaches an existing simple volume to or from initiators by executing the following steps:

Attach (state=Present)

1. Generate an authentication token.
2. Publish an existing simple volume to the additional initiator.
3. Connect an existing simple volume to the additional initiator.

Detach (state=Absent)

1. Generate an authentication token
2. Disconnect an existing simple volume from the corresponding initiator.
3. Unpublish an existing simple volume from the corresponding initiator.

If the simple volume does not exist, the operation stops with an informative message.  
**Required Parameters** You will need to set the parameters for the modules used by the playbook.

Modules Used by the Playbook

- ks\_token
- ks\_info
- ks\_publish
- ks\_connect
- *filesystem* (built-in Ansible module)
- *mount* (built-in Ansible module)

Example

To create a multiattached simple volume (full flow)

```
ansible-playbook -v multiattach_simple.yml -e state=present
```

To delete a multiattached simple volume (full flow)

```
ansible-playbook -v multiattach_simple.yml -e state=absent
```

Create a Snapshot Volume using create\_snapshot\_volume

**Description:** Creates a snapshot volume from existing snapshots by executing the following steps:

1. Generate the authentication token.
2. Create snapshot volumes per existing snapshot.
3. Publish the snapshot volume created to the corresponding initiator.
4. Connect the published snapshot volume to the corresponding initiator.

The playbook supports creating

- a snapshot volume from both replicated and single snapshot volume.
- A writable or read-only volume.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

Modules Used by the Playbook

- ks\_token
- ks\_info
- ks\_publish
- ks\_connect
- *filesystem* (built-in Ansible module)
- *mount* (built-in Ansible module)

**Examples** See [Examples of Provisioning and Management with Playbooks](#) for examples using this playbook.

Delete a Snapshot Volume using delete\_snapshot\_volume

**Description:** Deletes a snapshot volume from existing snapshots by executing the following steps:

1. Generate the authentication token.
  2. Disconnect the snapshot volume from the corresponding initiator.
  3. Unpublish the snapshot volume from the corresponding initiator.
  4. Delete the snapshot volume.
- If the snapshot volume is mounted, the operation stops with an informative message.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

Modules Used by the Playbook

- ks\_token

- ks\_info
- ks\_volume
- ks\_publish
- ks\_connect
- *filesystem* (built-in Ansible module)
- *mount* (built-in Ansible module)

**Examples** See [Examples of Provisioning and Management with Playbooks](#) for examples using this playbook.

**Installing the KumoScale agent with install\_host**

For details on using this playbook works, see [Ansible Module Installation and Configuration](#).

**Modules Used by the Playbook**

- ks\_token
- ks\_info
- ks\_host

**Uninstalling the KumoScale agent with uninstall\_host**

**Description:** Uninstalls the KumoScale agent rpm or deb from a remote host.

**Required Parameters** You will need to set the parameters for the modules used by the playbook.

**Modules Used by the Playbook**

- ks\_token
- ks\_host

**Example**

```
ansible-playbook -v uninstall_host.yml
```

**Examples of Provisioning and Management with Playbooks**

**Create or Delete a Writeable Snapshot of Replicated Volume**

Below is an example of the playbook sequences to create and delete a writable snapshot of replicated volume.

To create a writable snapshot of replicated volume:

```
ansible-playbook -v create_volume_replicated.yml
ansible-playbook -v ks_snapshot.yml -e "snap_uid=#####-1def-4725-b001-##### state=present"
ansible-playbook -v create_snapshot_volume.yml -e "vol_name=vol2 snap_uid=#####-1def-4725-b001-#####
reservedSpacePercentage=20" -e '{"writable": true}'
```

To delete a writable snapshot of replicated volume:

```
ansible-playbook -v delete_snapshot_volume.yml -e "vol_name=vol2"
ansible-playbook -v ks_snapshot.yml -e "snap_uid=#####-1def-4725-b001-##### state=absent"
ansible-playbook -v delete_volume_replicated.yml
```

**Create or Delete a Writeable Snapshot of Volume Simple**

Below is an example of the playbook sequences to create and delete a writable snapshot of simple volume.

To create a writable snapshot of simple volume:

```
ansible-playbook -v create_volume_simple.yml
ansible-playbook -v ks_snapshot.yml -e "snap_uid=#####-1def-4725-b001-##### state=present"
ansible-playbook -v create_snapshot_volume.yml -e "vol_name=vol2 snap_uid=#####-1def-4725-b001-#####
reservedSpacePercentage=20" -e '{"writable": true}'
```

To delete a writable snapshot of simple volume:

```
ansible-playbook -v delete_snapshot_volume.yml -e "vol_name=vol2"
ansible-playbook -v ks_snapshot.yml -e "snap_uid=#####-1def-4725-b001-##### state=absent"
ansible-playbook -v delete_volume_simple.yml
```

**Create or Delete a Read-only Snapshot of Volume Replicated**

Below is an example of the playbook sequences to create and delete a readable snapshot of replicated volume.

To create a read-only snapshot of replicated volume:

```
ansible-playbook -v create_volume_replicated.yml
ansible-playbook -v ks_snapshot.yml -e "snap_uid=#####-1def-4725-b001-##### state=present"
ansible-playbook -v create_snapshot_volume.yml -e "vol_name=vol2 snap_uid=#####-1def-4725-b001-#####
reservedSpacePercentage=0" -e '{"writable": false}'
```

To delete a read-only snapshot of replicated volume:

```
ansible-playbook -v delete_snapshot_volume.yml -e "vol_name=vol2"
ansible-playbook -v ks_snapshot.yml -e "snap_uid=#####-1def-4725-b001-##### state=absent"
ansible-playbook -v delete_volume_replicated.yml
```

Create or Delete a Read-only Snapshot of Simple Volume

Below is an example of the playbook sequences to create and delete a snapshot of simple volume.

To create a read-only only snapshot of simple volume:

```
ansible-playbook -v create_volume_simple.yml
ansible-playbook -v ks_snapshot.yml -e "snap_uid=#####-1def-4725-b001-##### state=present"
ansible-playbook -v create_snapshot_volume.yml -e "vol_name=vol2 snap_uid=#####-1def-4725-b001-#####
reservedSpacePercentage=0" -e '{"writable": false}'
```

To delete a readable only snapshot of simple volume:

```
ansible-playbook -v delete_snapshot_volume.yml -e "vol_name=vol2"
ansible-playbook -v ks_snapshot.yml -e "snap_uid=#####-1def-4725-b001-##### state=absent"
ansible-playbook -v delete_volume_simple.yml
```

See Next: [Ansible Variable File](#)

---

---