

# Hints on using Cinder Manage with OpenStack Newton and later



**openstack.**®



# CINDER

*an OpenStack Community Project*

In this post, I'm going to look at one of the most frequently misunderstood and misused commands available in the Cinder Block Storage project of OpenStack® is `cinder manage`. This is not the same as `cinder-manage` - maybe there was some regret at the naming of these two commands as their similarity does lead to confusion? Specifically, I'm going to be looking at OpenStack Newton and subsequent releases, as the Newton release made things easier to understand and use this feature.

Let's clarify the difference between these two similar commands before we go any further.

- `cinder-manage` (with the hyphen) is a command to get information about the current state of Cinder and to manipulate the Cinder database
- `cinder manage` (no hyphen) is the ability to import an external block storage volume into the control of Cinder.

The latter is the one I will cover in this post.

## Use case for `cinder manage`

As this command allows you to manage previously created volumes on a backend array, it facilitates the ability to import block devices that have been used in other environments, maybe OpenStack, maybe bare-metal. These imported volumes could be replica volumes from a remote OpenStack cloud that you want to use for Disaster Recovery testing.

## What can you import

Helpfully there is an OpenStack Cinder command provided from the Newton release to allow you to list volumes on a backend that are capable of being imported into Cinder for its management. All you need to know is some OpenStack configuration information of the storage backed you are interested in importing from.

Firstly, you must find the name of the storage pool you are using within OpenStack that references the backend you are going to import from. The reference to pools is OpenStack terminology and doesn't mean you can only use this feature with arrays that support storage pool. Again, this is another unnecessary confusion that arises from Cinder terminology. Remember it is hard trying to come up with generic names for features that are meaningful, but that doesn't match with one of the feature names used by one of the

eighty or so supported storage backends from forty or so different vendors...

Run the following command to get this pool information.

```
[crayon-6424cb58b075c967042500/]
```

In this case, we only have one storage backend configured, that being a Pure Storage® FlashArray.

Now we have the name of the storage pool related to this backend array we can have a look at the volumes that exist on this array, both from our existing OpenStack cloud and those created from other applications. Here we use the helpful command that became available from OpenStack Newton, `cinder manageable-list`:

```
[crayon-6424cb58b0763966665406/]
```

What this command is showing is that the array has, in total, eight volumes, four of which are already managed by this OpenStack cloud. Of the remaining volumes, three look like OpenStack volumes but are not managed - this is because they were created by another OpenStack cloud attached to the same Pure Storage FlashArray, and one remaining volume which is the one we are going to bring into our OpenStack environment.

## Managing the external volume

To bring this volume under our OpenStack Cinder control we use the `cinder manage` command. The syntax of this looks a little complex when you reference the [CLI description](#), but when you see it used it becomes very simple.

```
[crayon-6424cb58b0765443405511/]
```

Let's break this down.

- The parameter `--id-type` should be name, as we are going to reference the name field in the `manageable-list` output for the volume we want to manage
- `--volume-type` should be set to be what you want the volume type to be when managed by your OpenStack cloud
- `--name` is what the imported volume will be called after making it available to your environment
- next, we give the pool name of the backend array
- finally, we provide the name of the unmanaged volume we want to manage as detailed in the first volume of the `manageable-list` output

Now let's look at backend array again and see that the volume is now managed by our OpenStack Cinder environment:

```
[crayon-6424cb58b0768153803926/]
```

Notice that the name of the volume has been changed to conform to the standard required by the driver of the backend storage array you are using. In the case of the Pure Storage FlashArray driver, the volume name is the `cinder_id` appended by `-cinder`. In the example, the Pure Storage FlashArray performed the rename automatically on the backend array, so there is no need to do any management on your Pure Storage FlashArray. For different backends, you would need to consult with the owner of that driver.

Finally, from the standard Cinder view we see the volume with the name and volume type we defined in the `manage` command:

```
[crayon-6424cb58b076a075018354/]
```

So, not as hard or complicated as some may think.

## Going back

So now we have managed an external volume, how about we unmanage an OpenStack Cinder volume. This is more simple to achieve as we have to do is specify the Cinder ID or name of the volume.

Let's unmanage the volume we just managed:

```
[crayon-6424cb58b0770939852016/]
```

Looking at the manageable list again we can see how the volume is now outside of Cinder's control and that the volume has again been renamed, this time to the original name on the backend array and appended with -unmanaged:

```
[crayon-6424cb58b0771349024648/]
```

I hope that this short post has been helpful, as I know I have been asked about how this feature works several times.