Set Up SUSE Enterprise Storage for Veeam in About 30 Minutes

Written by:
John S. Tonello
Global Technical Marketing Manager, SUSE
In this guide, you'll learn how to use a SUSE Enterprise Storage 6.0 cluster (based on Ceph Nautilus) as a Veeam backup repository target, including how to set up an RBD client machine on a low-resource Linux host and implement a live back-up.

Feel free to try this in your own lab. Here's what you'll need:

- A running SUSE Enterprise Storage cluster (this example uses the latest version, 6.0)
- Veeam Backup and Replication 9.5 (community edition or licensed)
- A Windows machine (on which to run Veeam)
- Two Linux machines (one for the RBD client, one for backup)
- A network that ties everything together (this example uses a 10.128.1.0/24 subnet)
- About 30 minutes

This test case is very forgiving, so feel free to run it on virtual machines, bare-metal servers or a mix.

Your Ceph Cluster

Whether you're new to Ceph or a seasoned pro, the principles of using Veeam should be familiar to you. You start by setting up the storage itself (Ceph), tell Veeam about it, and then set up some one-off or recurring backups. Veeam offers more options than I can go into here, but it readily handles backups for full systems, volumes or files.

If you manage your Ceph cluster via the command line, great. You’ll have a chance to use those skills if you want to. But, I’ll take advantage of the Ceph Dashboard, introduced in SUSE Enterprise Storage 6.0, to create the necessary dedicated pool and an RBD block image.

Here are the steps:

- Create a storage pool – This will be called veeam-pool.
- Create an RBD image in the veeam-pool – This will be called veeam-storage.
- Map the veeam-storage image on a Linux host.
- Add that Linux host as a Veeam repository target.
- Run a backup of a Linux system.
A typical Ceph cluster has three monitor nodes. Mine have the following DNS names and IPs:
- mon1.example.com (10.128.1.141)
- mon2.example.com (10.128.1.142)
- mon3.example.com (10.128.1.143)

Since I have a DNS server running, these nodes, which will be used later as part of the RBD mapping commands, are discoverable by any machine on my 10.128.1.0/24 subnet. If you’re not using DNS, just make sure that you edit the hosts files on all the Linux and Windows machines you use, to ensure that everything can communicate.

If you don’t already have a Ceph cluster, take a look at this related SUSE Guide, Deploy a fully functional SUSE Enterprise Storage cluster test environment in about 30 minutes.
Create Your Ceph Resources

Create the storage pool

From the Ceph Dashboard, click on the Pools menu and then on the green Create button. In the new window, provide a name for your pool (veeam-pool) and the pool type (replicated). For placement groups, consider a small number if you have a small test cluster. Replication size will default to 3, but be sure to click the green pencil next to Applications and choose rbd. When you’re done, click the green Create Pool button.

You can create the same pool from the command line of your SUSE Enterprise Storage master/admin node with these commands:

```
# ceph osd pool create veeam-pool 8 \
    replicated replicated_rule 3
```

Then associate the RBD application:

```
# ceph osd pool application enable \ 
    veeam-pool rbd
```

Check to confirm that it was created with:

```
# ceph osd pool ls
```

---

Figure 2. Create the veeam-pool in the Ceph Dashboard.

---

Figure 3. The new veeam-pool now appears in the Pool list in the Ceph Dashboard.
Create the RBD Image
Now you can create a new RBD image, which will become the shared block device that Veeam will use as its backup repository target. In Ceph Dashboard, click Block → Images, and click the green Create button.

In the new window, enter the name of the image (veeam-storage) and select the pool you just created (veeam-pool). Enter a size appropriate to your needs. This example is 20Gb.

Click the green Create RBD button to save it.

If you want to do this via the command line, execute the following commands on your SUSE Enterprise Storage master/admin node:

```
# rbd create --size 20480 / veeam-pool/veeam-storage
```

You can check that it was created from the command line:

```
# rbd list --pool veeam-pool
```

That’s it on the storage cluster side of things for now.

Create a Linux Target Node
In order for Veeam to connect to your newly created veeam-pool/veeam-storage, you need to map and mount it on a Linux system that supports RBD kernel features. This can be any modern Linux, such as openSUSE Leap 15.1 or CentOS 7. I used a SUSE Linux Enterprise Server 15 SP1 VM and gave it minimal resources.

The kernel module works by creating devices in your Linux system's /dev folder; those devices can then be formatted and mounted like any other block storage device, such as /dev/sda1.

First, make sure RBD is installed and enabled on your target Linux host:

```
# which rbd
```

If rbd isn’t found on your system, install it:

```
# zypper in ceph-common
```

Now, load the kernel module:

```
# modprobe rbd
```

If you don’t get any errors, you’re ready to proceed.
Copy client keyring and .conf files
The connection between your Ceph cluster and Linux target node is made via the auth token called keyrings that was created automatically when you created your RBD image. On your SUSE Enterprise Storage master/admin node, keyring files are found in /etc/ceph/. Copy the /etc/ceph/ceph.client.admin.keyring and /etc/ceph/ceph.conf to your Linux target node:

```bash
cephadm:$ sudo scp -r /etc/ceph/* root@target-node.example.com:/etc/ceph/
```

These files will provide the proper credentials for connecting the image and creating the /dev/rdb0 device. The ceph.conf file contains information about your Ceph cluster’s monitor nodes, which is why name resolution is important.

Map the RBD device
With everything in place, run the rbd map command:

```bash
# rbd map --pool veeam-pool veeam-storage \
   --id admin \
   --keyring /etc/ceph/ceph.client.admin.keyring
```

After a moment, you should see simple output:

```
/dev/rbd0
```

This RBD device, which represents the 20Gb image you created earlier, is now available on your Linux target node.

Format the RBD block device
In order to store data on the RBD block device, you must format it. To format it with the xfs filesystem, do:

```bash
# mkfs.xfs /dev/rbd0
```

```
meta-data=/dev/rbd0 isize=512 agcount=8, agsize=32768 blks
  =sectsz=512 attr=2, projid32bit=1
  =crc=1 finobt=1, sparse=0, rmapbt=0
  =reflink=0
```

Mount the device to a local directory
If you want the RBD device to be mounted on a particular directory on your Linux target node, first create it and then mount it:

```
$ mkdir /storage
$ sudo mount /dev/rbd0 /storage
```

Check that everything worked as expected by running the following command on the Linux target node:

```
$ df -Th
You should see something like:
/dev/rbd0    xfs   20G  1G  19G   5% /storage
```

Map the device and mount it at boot time
The current configuration is a working setup and you could switch over to Veeam and start the process of adding your new RBD
share as a backup repository. However, on the reboot of your target Linux node, your device and mount won't be there. That would be bad, so you need to make your mapped settings permanent with `rbdmap` and `fstab`.

When you installed `ceph-common` on your Linux target node earlier, the `rbdmap` service was installed, too. Its config file is `/etc/ceph/rbdmap` and takes only a few arguments.

Edit `/etc/ceph/rbdmap` and add the following:

```
veeam-pool/veeam-storage  id=admin,keyring=/etc/ceph/ceph.client.admin.keyring
```

Note the similarity with the original `rbd map` command that was run earlier. Save the file and then enable the `rbdmap` service:

```
$ sudo systemctl enable rbdmap.service
```

With the RBD mount in place, edit your Linux target node's `/etc/fstab` in the usual way, adding your mount with the "noauto" option so that the host doesn't try to mount the `/dev/rbd0` device to `/storage` before the device is ready:

```
$ sudo vi /etc/fstab

... /dev/rbd0 /storage xfs defaults,noauto 0 0 ...

Save the file and you're done. If you want to test everything, go ahead and reboot your Linux target host. Then log in and execute `df -Th`. You should see something like this:

```
$ df -Th | grep rbd

/dev/rbd0  xfs  20G  1G  19G  5% /storage
```

### Set up Veeam Backup

The remaining steps are done on the Windows machine where you installed Veeam. This can be a Windows server or Windows 10 desktop, which I used in this example. Specifically, you'll be installing and using Veeam Backup and Replication, either the community or licensed edition.

Also, any guest OS support is valid as long as the original vendor supports that OS as part of the product lifecycle.

### Set Up a Veeam Backup Server

Start in the main Veeam Backup and Replication app view and select “Backup Infrastructure” from the left-hand navigation pane. Select “Managed Servers” in the frame above and select “Linux” (if the entry already exists) or choose Linux in the new window. Then click the “Add Server” button in the top left to begin adding your Linux target host.

Enter the DNS name or IP address of your target host. After clicking Next, you'll be asked to enter credentials for it. To be more secure, you can add a user with sudo privileges on the Linux target host, or you can just use root and the password. If you use a non-root user, Veeam offers options for automatically escalating privileges, which is handy so long as you've given that user sudo privileges. Using root is not recommended in production.

After you click Finish, you’ll return to the Backup Infrastructure view, where you’ll see your newly added target node.
Figure 5. Add your Linux target host in Veeam Backup and Replication.

Figure 6. Add credentials for Linux target node and view a summary of the connection.
Set up a Veeam Backup Repository

You're not quite done adding backup infrastructure. You now must create a Backup Repository using the Backup Server you just created. While still in the "Backup Infrastructure" view, click "Backup Repositories" in the upper pane and click Add Repository.

Although it might seem counterintuitive, select "Direct attached storage" from the menu and then choose "Linux." Give your repository a name and select your repository server from the drop-down menu.

Click the Populate button, which tells Veeam to connect to the Backup Server and display all the volumes on it. About midway down you'll see your /storage (dev/rbd0) entry. Select it and click Next.

Veeam will ask for a path on that /storage, suggesting /storage/backups. You can edit it to your liking; Veeam will create the directory for you on the Backup Server.
Use the default setting for the mount server (likely the name of the Windows host on which you set up Veeam). Then review the settings on the next screen and click Apply.

Figure 8. Select your backup server and click Populate

Add Backup Targets to Inventory
Before you can back up a system, you need to add it to your Veeam inventory. When adding Linux or Windows hosts to back up, it’s easiest to use Veeam Protection Groups, which can contain one or more individual computers to which you can apply the same backup rules and actions.

In the main Veeam view, click “Inventory” in the left-hand pane, and then in the main window to the right, click Add Server.
You’ll be adding a Protection Group. Click that and give it a name. The default is fine.

Click Add and enter the host name or IP address of the system you want to back up. My backup target is an openSUSE Leap 15.1 virtual machine at 10.128.1.136.

Select credentials from the drop-down list or add new ones. Again, to be more secure, you can add a user with sudo privileges on the machine; if you’re in a non-production environment, just use root and the password.

Before proceeding to the next step, you can click the “Test Now” button to ensure that Veeam can communicate with the machine you want to back up. On the next screen, leave the defaults or adjust the settings to suit your needs. Notice that Veeam will install its backup agent on the computer automatically.

After you click Finish, you’ll return to the Protection Group listed under the “Physical & Cloud Infrastructure” menu. You’re now ready to create a backup job.

Add a Backup Job
Veeam is now fully configured to use your SUSE Enterprise Storage cluster and the RBD block image you created. It’s connected to your Linux target host and it has mapped the RBD device you added there. That target node, in turn, is connected to your veeam-pool/veeam-storage and the block storage is ready to be used.

To perform a backup job, click on Home in Veeam’s lower-left navigation pane, which will reveal the “Backup Job” button in the top left. Click that button, select “Linux computer...” and choose the default settings: Type: “Server” and Mode: “Managed by backup server”. Give your job a name and add your computer using the Protection Group you just created.

Veeam offers three options for the backup type. Use the default “Entire computer” before going on to select your new backup repository.

In the drop-down menu, you’ll see all available repositories, including the one you created earlier (including its size).

On the next screen, set any automatic backup schedule that suits your needs and review the summary. Tick the “Run the job when I click Finish” box at the bottom of the Summary page if you want to run this backup right away. Veeam will install its agent on the system, scan the machine to be backed up and begin the backup.

Depending on the size of the machine you’re backing up, this could take some time.
time. Veeam is smart enough to only back up what’s in use on the system, not the entire disk. In my example, this added up to about 6.8 gigabytes. Ignore the slow throughput; this was done in a lab with very modest equipment.

Back on your SUSE Enterprise Storage cluster, you should see the results of the backup activity via the Ceph Dashboard. Note that the storage cluster is aware of the Veeam agent’s activity and reported the results accordingly.
Figure 15. Activity recorded on the SUSE Enterprise Storage cluster following the successful backup.

Resources

- SUSE Enterprise Storage product page
- Deploy a SUSE Enterprise Storage cluster test environment in about 30 minutes
- A more technical guide to SUSE Enterprise Storage and Veeam