8.7.2 Managing Quotas

Click one of the buttons to take you to that part of the video.

Manage Quotas 0:00-0:34

In this demonstration, we're going to practice managing quotas. In this scenario, we have a shared folder in '/mnt/shared', that many users use to share files with each other.

This folder has its own partition, and its own file system. As you can see here, /dev/sdb1 is mounted in /mnt/shared. Because this is a shared folder, we want to establish quotas on this file system to make sure everybody plays nice and doesn't use more space than they should.

Install Quota Package 0:35-1:01

The first thing we need to do is verify that the quota package itself is installed on the system, because without this package installed, you can't set up quotas. I do need to switch to my root user account first, and we use the 'rpm' command with the '-qi' option to query whether or not the quota package is currently installed on the system.

And as you can see, it has been installed, so we don't need to install it. If it wasn't installed, then you could use yum or dnf to pull it down from a repository and install it on the system.

Set Up Quotas 1:02-2:51

The first thing we need to do to set up quotas is to first enable quotas for the specific file system in question. This is done by editing the file system's mount options in the /etc/fstab file. Let's use the 'vi' editor. Edit the '/etc/fstab' file and we want to edit the mount options for this file system right here: /dev/sdb1.

We go in to insert mode, come over to the mount options field. We'll continue to use the defaults. We just want to add additional mount options that will be applied when this file system is mounted by the kernel.

In this scenario, we want to enable both user and group quotas, so let's first enable user quotas by entering 'usrjquota' (j stands for journaled quotas) '=', then we have to specify the name of the file that will contain our quota limits: 'aquota.user'. Remember, I said we also want to enable group quotas as well, so we add another option 'grpjquota=aquota.group'.

Our user quotas will be saved in this file, our group quotas will be saved in this file. Then we have to specify the format that we will use for our quotas. Let's enter 'jqfmt=vfsv0'.

Let's quickly review our syntax and make sure that we typed it in correctly, because if you type these options incorrectly, the more likely the kernel's not going to mount this file system. In this situation it wouldn't be a big deal, because you'd come back and edit it, and fix whatever needs to happen and mount it.

If, for some reason, you decided to enable quotas on your root file system, that all of a sudden gets kind of interesting, because you can't mount your root file system, so you can't go in and edit your fstab file.

Apply Mount Options 2:52-6:55

Okay, everything looks good. I'm going to write my changes to the file. At this point, the mount options that I just specified haven't been applied. They're not applied until the next time that partition is mounted in the file system.

There's a couple different ways that you can make this happen--one would be to just reboot the system. Another is to just tell the kernel to remount just that file system. We'll say 'mount -o remount', then name the file system that we want to remount: '/mnt/shared'.

We can verify that these options were applied correctly by typing just 'mount'. See the options that were used when that file system was mounted, and it's displayed here. Notice that /dev/sdb1 is mounted in /mnt/shared, and our quota options have been applied. We're good to go there.

At this point, we need to scan the file system to view our current disk space usage per user, and then create the appropriate user and group quota files. We need to create these two files that we specified right here and right here.

To do this, we run the 'quotacheck' command with the '-avmug' options. It's easy to remember; just remember av mug.

The -a option right here specifies that all file systems that have quotas enabled should be checked. -v says operate in verbose mode so that we can see exactly what's going in the output of the command. -m specifies that we should check file systems even if they're currently mounted in the file system. -u specifies that we check for users, and -g specifies that we check for groups.

Go ahead and press Enter. We do see some error messages displayed; don't worry about these. They're displayed because the two quota files that we specified in our mount options up here didn't exist, so the quota check command had to go out and create them.

At this point, let's verify that they were actually created 'ls /mnt/shared', and we see that we now have aquota.group, and aquota.user. That's exactly what should happen.

At this point, we're ready to actually enable quotas on the file system. This is done by running 'quotaon -av'. Notice that group quotas were turned on, and user quotas were turned on for the /dev/sdb1 file system, which is mounted in /mnt/shared.

At this point, we're ready to generate our quota report. This is done by running the 'repquota' command, and we use the '-av' options. Understand that at this point, no quotas have actually been defined in the file system, so the output of this command will just simply display how much space and how many files have been created by each user.

Go ahead and press Enter, and in the output of this command we can see that two users currently own files in /dev/shared: the root user and the rtracy user.

Currently the root user has used 84 blocks. The rtracy user has used 304. Root owns 11 files. rtracy owns 2 files. Also notice that the block and file limits are listed in the output of the command. They're all set to 0--basically meaning none have been set up yet.

Let's go ahead and fix that now by setting quotas for the rtracy user. To do that, we enter 'edquota', and then '-u' to specify the user that we want to set quotas for, the 'rtracy' user. I'll press my Insert key twice to go into replace mode.

Currently, the only file system with quotas enabled is /dev/sdb1; we just did that a minute ago. Under blocks, we specify how much disk space that the user can use, by limiting the number of blocks that they can consume for their files. Over here under inodes, we limit the number of files that they can create within this file system.

Notice also that we have hard and soft limits that we can define. A soft limit is one that the user is allowed to exceed temporarily for a period of time. By default, this is 7 days. We can also specify a hard limit that the user is not allowed to exceed at all.

Set Up Block Limits 6:56-7:39

Let's go ahead and set up block limits for the rtracy user account. Let's set up a soft block limit of '10000' and a hard block limit of '15000'. These represent the maximum number of blocks that the rtracy user will be allowed to create in /mnt/shared, effectively limiting the amount of space that he is allowed to consume.

You can also limit the number of files that the rtracy user is allowed to create. Let's limit rtracy to just '800' files for a soft limit. For our hard limit, let's set it to '1000'. Be aware that these are actually very small limits; this is just a demonstration system.

In a production environment, you'd have to adjust these limits as appropriate for the type of work that your end users actually do.

Review Limits 7:40-8:42

Let's go ahead and save our changes and 'exit' the editor. Let's review the new limits that we just set. Let's run the 'repquota' command again, and this time we see not only the usage for blocks and inodes, but we also see the limits that have now been set.

Notice that root does not have any limits. Root can create as many files as root needs to, but the rtracy user does have hard and soft block limits set, and hard and soft inode limits set. Look up here, notice that the grace time is currently set to 7 days for both inodes and for blocks. This is the amount of time that's applied to both soft limits here.

You don't have to stick with 7 days if you don't want to; you can change that. That's done again using the 'edquota' command. This time we'll use the '-t' option. Here we can specify the grace period that we used for blocks. Let's change it to '5', and for inodes change it to '5' as well, and 'exit' the editor to save our changes.

Set Limits for Groups 8:43-9:41

If we run the 'repquota' command again, we see that the grace time has been set to 5 for both inodes and blocks. In addition to setting limits for users, you can also set limits for groups. Remember we enabled both user and group quotas.

In this scenario, let's say we want to create limits for any users who are members of the group named users on this system. To do this, we run 'edquota' again. This time instead of specifying '-u' for users, we specify '-g' for groups, and then the name the group for whom we want to establish limits. The name of the group is users.

They're set up in exactly the same way as we set it up for our users a minute ago. Let's set up a soft block limit for the user's group of '15000', a hard block limit of '20000', a soft inode limit of '1000', and a hard inode limit of '1500'.

Test the Quotas 9:42-11:09

Let's save our changes, 'exit' the editor. At this point, we're ready to test the quotas that we have just set. Recall earlier that we set a fairly small file space usage limit for the rtracy user. Let's go ahead and 'exit' out of my root user account, and then as rtracy let's create a very big file in /mnt/shared, containing just junk, to test whether or not the quota will be applied.

To do this, I'm going to use the 'dd' command. We'll use the input file of '/dev/0', and an output file of '/mnt/shared', and we'll name the file 'bigfile.bin'. We'll use a block size of '1024', and we'll specify a count of '100000'.

Don't get too uptight about what this command exactly does, just understand that what we're doing is we're taking a bunch of junk data out of here and we're writing it to this file. And we're just going to keep doing it over and over and over 100,000 times, which will create a very large file in /mnt/shared.

Hit Enter. Notice that it started working, started trading the file, but then we hit an error. That's because we exceeded my disk space quota. This user was blocked, prevented, from creating an excessively large file that would steal all the disk space in the file system from the other users.

Summary 11:10-11:16

That's it for the demonstration. In this demo we talked about establishing quotas in the Linux file system. To do this, we set up both user and group quotas.