31338 Network Servers
Lab 7b: Implementing disk quotas

Aims

1. To implement disk quotas for a Linux server

Introduction

Your task is to set a disk quota for the user *peter* created in an earlier lab exercise. There are two options for completing this lab. One is to use the `/opt` filesystem you created in Lab 2d (Disk partitioning), and set quotas on that filesystem. The other option is to set quotas on your root filesystem (`/`). You can do either or both, but the steps are slightly different. When we created `/opt`, it was an ext4 filesystem. Our root filesystem uses xfs (default since Centos 7). It is worth being familiar with both, so it is recommended to do both tasks.

Task 1: Setting up disk quotas using ext4 (/opt filesystem)

The basic steps are:

1. Edit `/etc/fstab` and add the option to turn on user quotas (`usrquota`) for the opt directory filesystem (`/opt`). Be very careful with this step, as a corrupted fstab file will prevent your system from booting normally. As a safeguard, make a copy of /etc/fstab, calling it, for example, /etc/fstab.bak. This will make recovery easier.

2. Remount the filesystem so that the new options from fstab take effect. A useful command to remount a filesystem when you have changes the options is below (for the `/opt` filesystem in this case):
   ```
   mount –o remount /opt
   ```

3. Run the `quotacheck` command for the `/opt` filesystem.

4. Turn quotas on with the `quotaon` command (this command needs an argument – read the man entry if in doubt)

5. Use `edquota` to edit `peter`'s disk quota. For block usage, set a soft limit (warning) at 400KB and a hard limit at 500KB. You do not need to set limits on inodes (which is a limit on the number of files a user can create).

Test that your disk quotas are working by logging in as the user `peter`, and trying to store more than 500KB in the /opt directory (copy a large file from somewhere else). You may need to create a directory under /opt and make it so that peter can write to that directory (either make him the owner, or change the directory permission). Hint: an easy way to create a large file is to use the dd command, for example, `dd if=/dev/zero of=junk bs=1024 count=600` You can also run the `quota -v` command while logged in as `peter` to show the quota values for `peter`.

As root, run `repquota` to generate a quota report for the `/opt` filesystem.

Carefully document the process you followed, including all of the options to the commands. Also document your testing procedure, and what messages the system showed when the user exceeded his soft limit and his hard limit.

Task 2: Setting up disk quotas using xfs (root filesystem)

To enable user quotas on the root filesystem on Centos, there is an extra step – just adding options to /etc/fstab is not enough any more (once it was). We need to tell the GRUB2 bootloader to load the filesystem with quota support enabled at boot time. This is only needed for xfs quotas on the root filesystem – for setting xfs quotas on any other filesystem you don't need this step.

The steps are:

1. Edit `/etc/fstab` and add the option to turn on user quotas (`usrquota`) for the root filesystem (`/`). As a safeguard, make a copy of /etc/fstab, calling it, for example, /etc/fstab.bak. This will make recovery easier.

2. **(Only needed for root filesystem)** Edit `/etc/default/grub` so that we can add the root filesystem quota options in the GRUB2 bootloader config. In this file, look for the line that starts with GRUB_CMDLINE_LINUX. Edit this line and at the end of the line (but still within the double quotes), add the extra entry:

   ```
   rootflags=usrquota
   ```

3. **(Only needed for root filesystem)** Run the command below. It will read the `/etc/default/grub` file and rebuild the GRUB2 bootloader configuration taking the new options into account.

   ```
   grub2-mkconfig
   ```

4. **(Only needed for root filesystem)** Reboot your system. Run the mount command to verify that the new filesystem options are shown. The "mount -o remount" option won't work this time. Also note that with xfs quotas we **don't** need to run quotacheck to set things up. xfs stores quotas as part of the internal filesystem data, so it manages it automatically.

5. Check that quota accounting and enforcement are both enabled with the command below. If not, you can use the `quotaon` command to turn enforcement on (and `quotaoff` to turn enforcement off).

   ```
   xfs_quota -x -c state
   ```

6. Use `edquota` to edit `peter`'s disk quota. This time, check the current block usage first (it is shown in edquota). Set a soft limit (warning) just a little bit higher than the current usage and a hard limit higher again. For example, if Peter's usage on the root filesystem was 11223 blocks, set the soft quota at 12MB and the hard quota at 13MB. You do not need to set limits on inodes (which is a limit on the number of files a user can create).

Test that your disk quotas are working by logging in as the user `peter`, and trying to store more than the hard limit in his home directory (copy a large file from somewhere else). Hint: an easy way to create a large file is to use the dd command, for example, `dd if=/dev/zero of=junk bs=1024 count=600`
You can also run the `quota -v` command while logged in as `peter` to show the quota values for `peter`.

As root, run `repquota` to generate a quota report for the `/` filesystem.
With xfs, you can also generate a report with: `xfs_quota -x -c report`

Tip: The `xfs_quota` command also has an interactive option. If you run it without options you can view quota information (also available to normal users). If you run `xfs_quota -x` (expert mode), then you can edit quotas, show quota reports, etc.

Carefully document the process you followed, including all of the options to the commands. Also document your testing procedure, and what messages the system showed when the user exceeded his soft limit and his hard limit.

Advanced (optional): Both ext4 and xfs support "project quotas", which means you can limit storage in a specified directory hierarchy within a filesystem. This is a newer concept for Linux – in the past quotas were only for users or groups, and applied to entire filesystems. Project quotas better reflect how organisations work – people typically work in teams on shared projects, and an administrator might want to limit the storage for a particular project rather than a particular user (users work on multiple projects, so per-user quotas are complicated for most organisations).

If this is of interest to you, research how to set up project quotas. Create a quota for the `/share/family` directory and test that different users can create files in that directory up to the project quota limit. You might need to turn off user quotas while testing, otherwise it's tricky to test whether the user quota or project quota is limiting.