SUPINFO Academic Dept.

SUPINFO International University

INSTITUTE OF INFORMATION TECHNOLOGY

1LIN - Linux - Foundations
Project - Rugby Club

Contents

Version 1.0
Last update: 18/02/2019
Use: Students/Instructor
Author: Laurent SANSELME

TABLE OF CONTENTS

1. Objectives

This project is a research project that will help you improving your overall Linux skills. You will explore course topics beyond the initial in-class course scope.

Your final goal is detailed in the "Project" chapter.

2. Project

You work in the IT department of a big Rugby club. For years now, every supporter has had a dedicated space to upload and share their favorite pictures and videos so every fan could see it. After a good decade without any problem, HD (and then 4K) cameras and smartphones have made the storage of such a big quantity of data much harder.

Some supporters were even abusing their storage capacity by uploading personal content just as if it was some kind of clouded-drive.

To avoid those problems in the future, you decide to modify the infrastructure so that the service stays the same for real fans but can not be abused.

You are going to realize a proof-of-concept and present your work to the President of the club. You will prepare and practice your final demonstration.

Needless to say, everything you're going to do will have the appropriate authentication and security options (strong passwords, …).

2.1. Global

You will install a full server for this project. To emulate that for your demonstration, you will create a virtual machine with your favorite VMWare product and install on it the Linux distribution of your choice.

The Physical server used for the actual solution will be the one used when your POC is validated. For that reason, you will replicate it's architecture approximatively. Server has:

- 64GB of RAM so you will use at least 2GB for your VM
- 500GB hard drive for the system so you will use at least 50GB
- 50TB hard drive for the storage so you will use at least 10GB

2.2. Quota limits

To prevent users abusing their storage, you will make a quota solution. In that POC, users won't be able to store more than 500MB of data in their own folder.

Note

Read the script part too. There can be multiple valid ways to have a quota solution

2.3. Users

You will need a certain amount of users to show that your POC is viable distributed as follow.

2.3.1. User accounts

For your realization, you will create 3 user accounts named at your convenience. These users will be used to show how quota work and what future users will leave in different situation in the future.

- The first one will be a new user connecting for the first time with no data at all.
- The second one will be one with 250MB of data stored.
- The last one will be one almost at the limit or just passing the limit so the demo can show what happens when a user wants to abuse his storage capacity.

You will also create a user dedicated specifically for administration purpose

2.3.2. System account

In the next part, you will see that you need a script to be run every hour. That script must be launched by a system user named **rugbadmin**.

Note

Be careful that this user is totally distinct from the user created for administration purpose.

2.4. Script

This script can be the core point of your solution. If you don't rely on a built-in quota solution, this is what will make the magic.

When a user abuse his limitation, you don't want him to loose everything but you also don't want him to do that again. For that reason, you will save his data in a safe place where you will be able to give him back if he wants them or delete them after a while. During that time, he won't be able to access his files and will only see an empty folder.

Every hour, your script is going to verify if a user has exceeded his quota. If so, it will archive everything in the user's storage repository, compress it and name that file with the following convention: **USERNAME-creationdate.extension**

- **USERNAME** being the username of the quota exceeded user in uppercase
- **creationdate** being the date of the day the archive has been created
- **extension** being the extension of the file you generated depending on the archiving method and compression algorithm you will choose

Once you have created this archive, you will move it in a folder that is only accessible to the system user and only them you will remove everything in the user's folder.

The script also keeps track of everything that happened for login purpose but also if reporting is needed.

Note

Maybe cron?

2.5. Sudo

Every time a user wants to recover his data, you will need to connect to the server with your administrative account to give him is archive to download. You don't want everybody to do that and it's the only administrative task that this user is going to have to do.

You know your security and want the privileges to be as low as possible for that administrative user. For that reason, you will use **sudo** to delegate him the **chown** execution rights and the **mv** rights no matter what. Indeed, he has to be able to move the archive in a folder the user will be able to access in order to download and thus have the rights to do so.

Be careful that you want the minimum rights for that administrative user. He doesn't need to be a full root but he needs to enter a password. Don't forget to be sure that a password will always needed to change ownership of files or to move them from place to place.

2.6. AWK

For the moment you don't now which percentage of people try to abuse the system. You want to have better information so you can adapt your fanbase offer. Moreover, you want to be sure that you can have proof of multiple abuse if you need to take measure with a supporter.

Awk is the perfect tool for that purpose.

You want to make a report of which size every archive was, which user it belonged to and how often all that happens by user and with all users.

Note

If you used a built-in quota solution, the logs you will need for your report may be in the /var/log directory.

2.7. SSH

You're the only qualified IT people of the department. Being a sportive association, the rugby club isn't rich. To be able to take pleasant and relaxing holidays, you will implement an SSH server solution so you can administrate the server and recover user's archive even if you're far away.

SSH connection must be only allowed by key and for your administrative user.

2.8. Bonus

You can have 2 extra bonus point if you make a script that allows to generate users, files and show a full demo of actions and reporting. We want it at least to generate 50 users and the reporting must contain at least 50 case of archiving.

3. Conditions And Delivery

Although you can perfectly do this project on your own if you decide to do so, this project has been designed to be done by a group of 4 students. There is no need to declare your group before the delivery: Will be considered part of the group all students mentioned in the final delivery. All members of the group present in the delivery will get the same mark.

You have to hand back the following items on the https://sce.sad.supinfo.com/ platform in a ZIP file:

- Configuration files for each service
- Source code of your script/s
- SHA1 Checksum of all your virtual machines hard disks. Give a list of all virtual machines disk files names (vmdk if you're using VMware) and checksums.
- A list of the group members

You will show your work and do a live demonstration during orals.

Warning

You're going to send checksums of your virtual machine disks. Don't power up the virtual machine after the checksum, only do it when asked by your instructor.

4. Deadline

You must have sent items from Chapter 3 in a ZIP file on the http://sce.sad.supinfo.com/ platform before the date given on the 1LIN exam section. It may vary depending on your location.

5. Graded Items

Table 5.1.

Services	Description	Points	Total Points
Quota limits	User limitation is correctly set	2	2
Users	User and system account correctly created	2	2
Scripting	Runs every hour	1	6
	Compress and archive with correct name	3	
	Logs what happens	2	
Sudo	Administrative user's rights correctly delegated	3	3
Awk	Full report	3	5
	Good design	2	
SSH	Distant access by key for administrative user only	2	2
Bonus	**Testing script with user generation**	**2**	**2**
Total			20/**22**