**Task 1: Tom needs to create new accounts for Jack and Jane with their own respective directories. Also, Tom needs to make a dir ‘script’ inside /home where all the users can edit and view files.**

1. Creating two users and assigning them their respective directories i.e., /home/jane and /home/jack. After that we will set a password for the accounts.

sudo useradd -m jane

sudo useradd -m jack

sudo passwd jane

sudo passwd jack

1. Now, we will create a group and add jack,jane and tom to it.

sudo groupadd net\_admins

sudo usermod -G net\_admins -a jack

sudo usermod -G net\_admins -a jane

sudo usermod -G net\_admins -a tom

1. Since we want ‘script’ to be accessible to jane and jack, so, we will change the owner rights of ‘script’ dir. Now all the members of the group will be able to access the directory.

sudo chown -R tom:net\_admins script

sudo chmod -R 770 script

**Task 2: Create a cronjob that deletes empty files and directories from /tmp folder at 00:00 everyday.**

On the terminal,open crontab to add new entry and add following commands:

crontab -e #To add new entry to cron tables

0 0 \* \* \* find /tmp -empty -type f -delete

#This command finds empty files( -empty -type f) in /tmp first, then deletes them. This cron entry runs at 12:00am.

Cron syntax: minute hour dayofmonth month dayofweek

0 0 \* \* \* find /tmp -empty -type d -delete

#This command finds empty directories( -empty -type d) in /tmp first, then deletes them.

**Task 3: Create another cron job to upgrade the system at 6:00 am on every tuesday.**

Creating a bash script upgrade-pkg.sh at /home/tom/. To create script use : vim upgrade-pkg.sh

#!/bin/bash

(sudo apt-get update && sudo apt-get upgrade) > upgrade.txt

Now esc and :wq to save and quit.

Open crontab and cron entry for 6:00 am on tuesday(2):

crontab -e

0 6 \* \* 2 /home/tom/upgrade-pkg.sh

**Task 4: To connect to the remote system and install git as SCM.( Make sure local and remote systems are connected to each other. For the experiment, I connected to VMs by building an internal network between them.)**

1. Build ssh connect to remote system say simar:

ls -al ~/.ssh/id\_\*.pub

#Open terminal and type above cmd to check if already have key-pair. If you see No such file or directory or no matches found it means that you do not have an SSH key and you can proceed with the next step and generate a new one.

ssh-keygen -t rsa -b 4096 -C "[tom123@gmail.com](mailto:tom123@gmail.com)"

Enter file in which to save the key (/home/tom/.ssh/id\_rsa):

#The above command will generate a new 4096 bits SSH key pair with your email address as a comment. Now, a prompt will ask you for a location to store these keys. Press Enter to accept the default file location and file name:

Enter passphrase (empty for no passphrase):

#You can either enter a passphrase or press enter to leave it empty.

ssh-copy-id simar@10.0.2.8

Enter simar’s password

# Now we will copy our generated public key to a remote machine server. This command will append the key authorize\_keys file in /home/simar/.ssh/

ssh simar@10.0.2.8

#Access the system from root account or account with sudo rights.

1. Now we will access the remote system and install git:

ssh simar@10.0.2.8

sudo apt-get update

sudo apt-get install git

git config --global user.name "simar"

git config --global user.email "simar1[2@gmail.com](mailto:tom12@gmail.com)"

#Installing git and setting username and password

mkdir Testing\_repo

cd Testing\_repo

#making a local git directory

git init

git clone [https://github.com/simisag/Testing-repo.git](https://github.com/tom/test.git)

#Initializing local git repository and cloning github repository on your local repository

git config --global credential.helper store

git push [https://github.com/simisag/Testing-repo.git](https://github.com/tom/test.git)

Enter username

Enter password

#To store git credentials permanently in a text file, so that administrator does not have to enter a password every time. It will ask you for your username and password to store. Your username and password are stored

git add .

git commit -m “Testing”

git push origin master

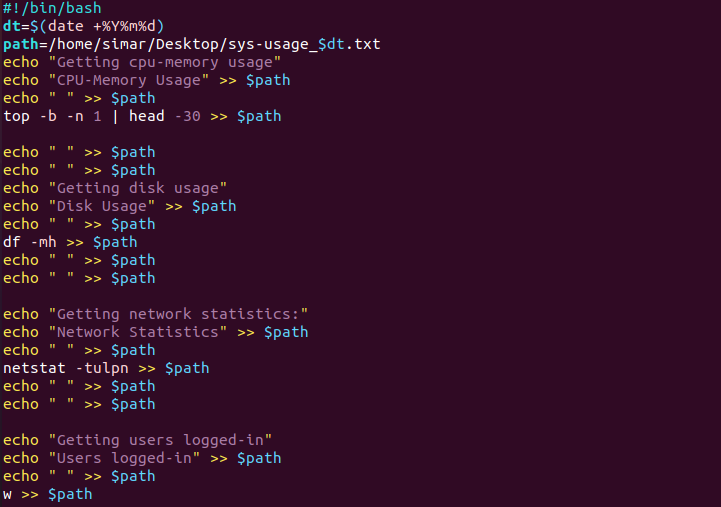
#Create any file and then add to staging area and commit and push to the branch.

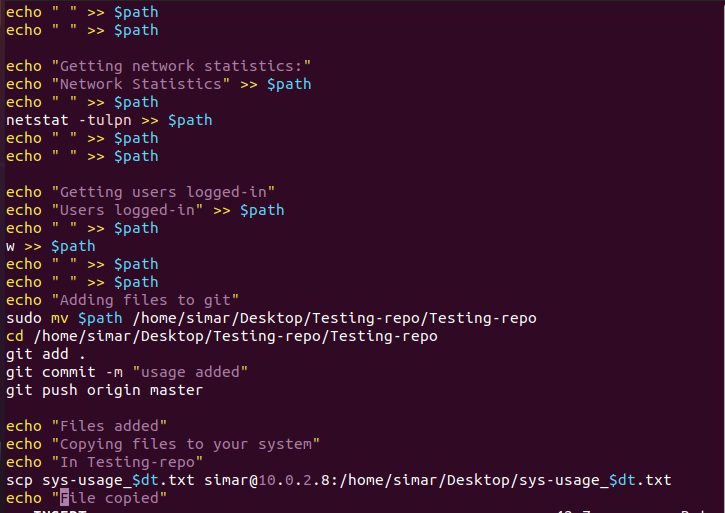
**Task 5: To create a script on Tom’s system that fetches CPU load ,memory usage,network usage, disk usage and active logged-in users data of the remote server to his system. Also this data should be moved to the git repository and pushed to the remote repository.**

1. Create a script sys.sh at /home/tom/Desktop/script. This script will append the output of all the commands to a file which will be created at /home/simar/Desktop with name sys-usage\_<current-date>.txt and each day one file will be created.
2. Now, move the created file to git repository Testing-repo and add push the file to remote repository.
3. Finally, we will copy sys-usage\_<current-date>.txt to tom’ s system using scp command.
4. To run this script, go to the script location:

sudo chmod 700 sys.sh

cat sys.sh | ssh simar@10.0.2.8





\