Infrastructure Integration 2

Team name: team10

server ip address: 10.134.178.10

1. User accounts for team members

Executed: Jasper **Reviewed:** Vlad

a. Create accounts for all team members

```
root@team10int2: ~
# tail -n 5 /etc/passwd
team:x:1004:1004::/home/team:/bin/bash
Vasil:x:1005:1005::/home/Vasil:/bin/bash
Jasper:x:1006:1006::/home/Jasper:/bin/bash
Vlad:x:1007:1007::/home/Vlad:/bin/bash
Panagiotis:x:1008:1008::/home/Panagiotis:/bin/bash
```

Comment: I first gave myself sudo rights with the command “sudo su -” after that I used the command “useradd *username*” to add all the users. Command “passwd *username*” was used to set everybody’s default password.

b. Add them to a secondary group which you create for your team

```
root@team10int2: ~
# tail -n 1 /etc/group
team10:x:991:
```

Comment: Used the command “groupadd -r *groupname*” to add a new group. The command “usermod -g *groupname username*” was used to add all users to their group.

c. Grant sudo rights to the group members. Do so by adding a config file in the configuration "drop-in" directory sudoers.d

```
root@team10int2: ~
# cat /etc/sudoers.d/team10
%team10 ALL=(ALL:ALL) ALL
```

Comment: Used the command “touch /etc/sudoers.d/filename”. After that, I edited the file in vim and added the following line “%team10 ALL=(ALL:ALL) ALL”. I ran the following command to validate the sudoers file “visudo -c”. To test the sudo rights by logging in as a member of the group and running a command with sudo “sudo whoami”. If the command returns “root” the sudo rights have been granted successfully.

d. Change the permissions of this file so that only the owner root can read/write the config file

```
-rw-------. 1 root root 26 Feb 13 14:04 team10
```

Comment: The command “chmod u=rw,g=,o= team10” was used to change the permissions of the file so only the root owner can read and write this file.

e. Test and prove the results
   Comment: It’s shown in each part already.

f. Remove the initial account when tasks have been completed successfully
   Comment: The command “userdel team” was used to delete the default initial server account we were provided with.

g. Nice extra: perform the user creation tasks in a bash script

```
#!/bin/bash

# Define a function to log messages using logger command
log() {
  # Get the current date and time
  local datetime=$(date +"%Y-%m-%d %H:%M:%S")
  # Get the name of the script
  local scriptname=$(basename $0)
  # Get the message level and text
  local level=$1
  local message=$2
  # Write the message to a custom logfile using tee command
  echo "$datetime $scriptname $level: $message" | tee -a
/var/log/usercreation.log
  # Write the message to system log using logger command
with tag usercreation and priority user.$level
  logger -t usercreation -p user.$level "$message"
}

# Check if the script is run as root
if [ $(id -u) -ne 0 ]; then
  log err "You need to run this script as root"
  exit
fi

# Read the username and password from arguments
username=$1
password=$2

# Check if the username and password are provided
if [ -z "$username" ] || [ -z "$password" ]; then
  log err "Usage: $0 username password"
  exit
fi
```

```
# Create the user
useradd $username

# Check if the user creation was successful
if grep "$username" /etc/passwd >/dev/null 2>&1; then
    echo "User created successfully."
else
    echo "User creation failed."
fi

# Set password for the user
echo "$username:$password" | chpasswd

# Check if the password was set successful
passwd_status=$(passwd -S $username | cut -d " " -f 2)
if [ "$passwd_status" == "PS" ]; then
    echo "User $username created successfully with password
$password."
else
    echo "Failed to set password for user $username."
fi

# Add the user to the teams group
usermod -a -G team10 $username

# Check if the user was added to the group team successful
if id -nG $username | grep -qw team10
then
    echo "User was addes successful to the team group."
else
    echo "Failed to add user to the group team."
fi
```

Comment: I created a file using vim with the name “usercreation.sh”. Afterwards I edited the file in vim to create my bash script. To give everybody access to the bash script, I created a directory named script inside the “opt” folder. Afterwards I changed the permissions of the file so that everybody can execute the file. Also changed the group and user to team10 and root.

h. Add checks in the script which check the results, write a meaningful logfile, use rsyslog and logger to log messages.
Comment: This was all done inside the bash script.

Username	**Default password**

Vasil	initial01
Jasper	initial01
Vlad	initial01
Panagiotis	initial01

2. Passwordless ssh access

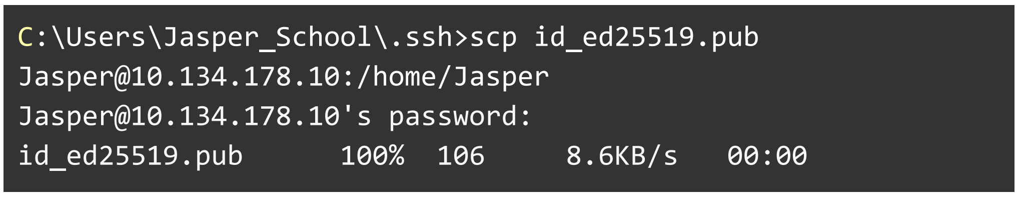
Executed: Jasper **Reviewed:** Vlad

a. Create an ed25519 key pair on your native operating system (look up the option to create a ed25519 key)

```
C:\Users\Jasper_School>ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\Jasper
School/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Jasper
School/.ssh/id_ed25519
Your public key has been saved in C:\Users\Jasper
School/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:zr5Jkf4bVNlHoY2drsnJNBLjy5t9FLV2h1SG/9Far28 jasper
school@Laptop-JM
The key's randomart image is:
+--[ED25519 256]--+
|               +=|
|             oBoo|
|           oo+.*=|
|         ...o o**|
|        S .o o.+B|
|       + o. * *.o|
|        = .o *.. |
|       o o .+ ..E|
|        +.o+ ..o.|
+----[SHA256]-----+
```

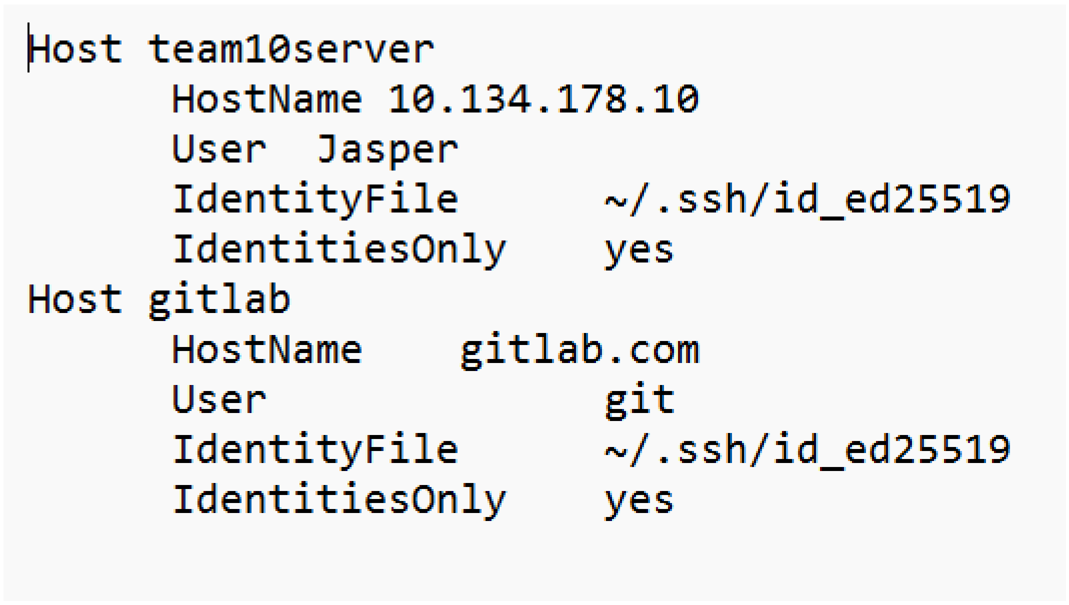
Comment: I used the command “ssh-keygen -t ed25519” to create the key pair.

b. Transfer the public key to your user account on the server in the correct folder/file


```
C:\Users\Jasper_School\.ssh>scp id_ed25519.pub
Jasper@10.134.178.10:/home/Jasper
Jasper@10.134.178.10's password:
id_ed25519.pub      100%  106     8.6KB/s   00:00
```

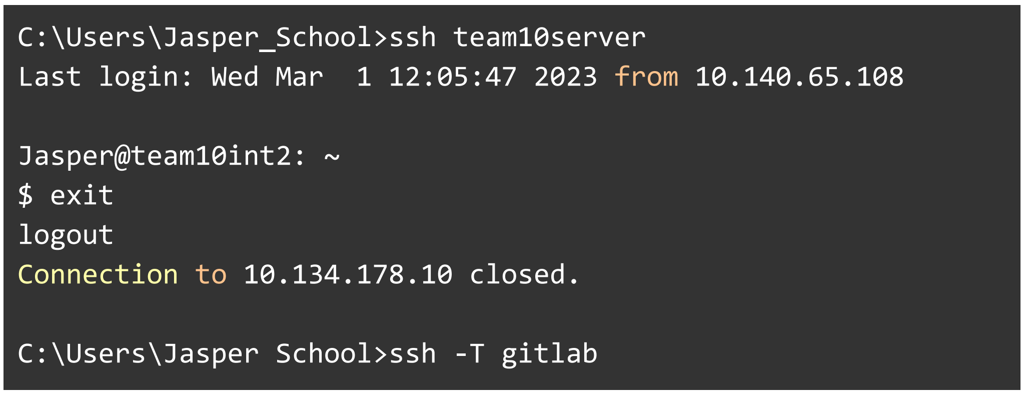
Comment: To copy the contents of the public key, I used the command "scp id_ed25519.pub Jasper@10.134.178.10:/home/Jasper"
After that I created a file called "authorized_keys" and used the command "cat id_ed25519.pub >> authorized_keys" to move the contents of the key to authorized_keys.

c. Create a "config" file on your native operating system (windows?) with a reference to the private key and the ip address of your team server.


```
Host team10server
      HostName 10.134.178.10
      User  Jasper
      IdentityFile     ~/.ssh/id_ed25519
      IdentitiesOnly   yes
Host gitlab
      HostName   gitlab.com
      User             git
      IdentityFile     ~/.ssh/id_ed25519
      IdentitiesOnly   yes
```

Comment: I created a file named config with the command "type nul > filename". Then edited the file in the notepad and pasted the contents of the public key in it. I also created a second host for our ssh connection with gitlab.

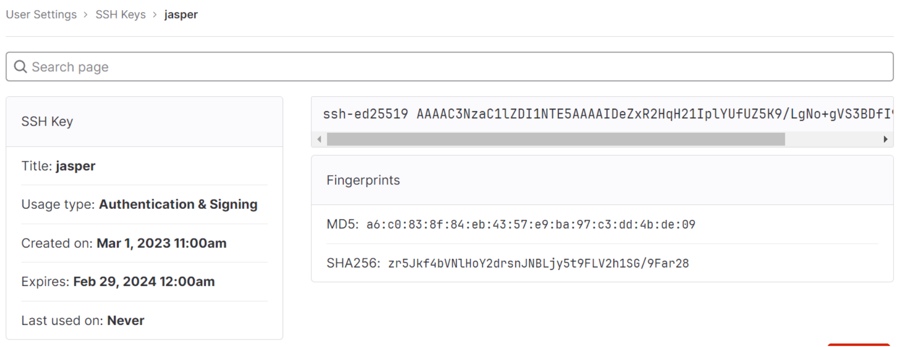
d. It is important that you can login from your native operating system using either the command line 'ssh' command and/or putty


```
C:\Users\Jasper_School>ssh team10server
Last login: Wed Mar  1 12:05:47 2023 from 10.140.65.108

Jasper@team10int2: ~
$ exit
logout
Connection to 10.134.178.10 closed.

C:\Users\Jasper School>ssh -T gitlab
```

```
Welcome to GitLab, @Jasper_Marichal!
```

Comment: You will use the command "ssh team10server" to login in our team server. The command "ssh -T gitlab" is used for our connection with gitlab.

e. You can use passwordless authentication with gitlab too.
Generate a key pair and add the public key to your gitlab configuration.



User Settings › SSH Keys › jasper

Search page

SSH Key

Title: **jasper**

Usage type: **Authentication & Signing**

Created on: **Mar 1, 2023 11:00am**

Expires: **Feb 29, 2024 12:00am**

Last used on: **Never**

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDeZxR2HqH21IplYUfUZ5K9/LgNo+gVS3BDfI
```

Fingerprints

```
MD5: a6:c0:83:8f:84:eb:43:57:e9:ba:97:c3:dd:4b:de:09
SHA256: zr5Jkf4bVNlHoY2drsnJNBLjy5t9FLV2h1SG/9Far28
```

Comment: I copied the private key id_ed25519.pub and pasted it in gitlab. You first go to user settings, then you go to the section SSH keys and there you create your new key.

Teacher: Make sure every user has added his/her key. Added everybody's ssh connection except for team1 who is using our server to store a remote backup.

3. Additional network configuration

Executed: Vlad **Reviewed:** Jasper

a. Add a new network connection named "instructors" and configure this connection to use **ens19** with a static ipv4 address

```
root@team10int2: ~
# nmcli con add con-name instructors ifname ens19 type
ethernet ipv4.addresses 192.168.1.10/24
Connection 'instructors'
(a12d0825-36bb-48d9-a5bd-08fd2614c325) successfully added.
```

Comment: I've created a new connection named "instructors", on the **ens19** interface, with the static ip 192.168.1.10 and the subnet mask 255.255.255.0

```
root@team10int2: ~
# nmcli con up instructors
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/3)
```

```
root@team10int2: ~
# nmcli con show --active
NAME           UUID                TYPE          DEVICE
ens18          4b465f2d-9419…      ethernet      ens18
instructors    a12d0825-36bb…      ethernet      ens19
```

Comment: Activated the connection "instructors" and checked if it is active using the **show –active** command

b. Check connectivity with the instructor's server on 192.168.1.250
Comment: To check connectivity you will need to ping to the server. "ping 192.168.1.250"

```
root@team10int2: ~
# ping 192.168.1.250
PING 192.168.1.250 (192.168.1.250) 56(84) bytes of data.
64 bytes from 192.168.1.250: icmp_seq=1 ttl=64 time=0.598 ms
--- 192.168.1.250 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
2034ms
rtt min/avg/max/mdev = 0.313/0.433/0.598/0.120 ms
```

c. To complete this task, you will copy a funny picture (jpg format) of your team to the home folder of user `team` on your instructor's server 192.168.1.250

```
Vlad@team10int2: ~
$ sftp team@192.168.1.250
team@192.168.1.250's password:
Connected to 192.168.1.250.
sftp> put team10.jpg
Uploading team10.jpg to /home/team/team10.jpg
team10.jpg             100%   32KB  23.3MB/s   00:00
```

Comment: Uploaded the jpg file using **sftp**

Teacher: The second interface should have only one IPv4 address. You still have 2 ip's on the ens19 connection. I needed to delete the 10.134.177.24/21 ip address on the ens19 connection. I did that my executing the command "ip addr del 10.134.177.24/21 dev ens19" after that I checked with the command "ip a" if it was successful and yes it was. c

```
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
state UP group default qlen 1000
    link/ether 46:a0:b4:4e:fe:8a brd ff:ff:ff:ff:ff:ff
    altname enp0s19
    inet 192.168.1.10/24 brd 192.168.1.255 scope global noprefixroute
ens19
       valid_lft forever preferred_lft forever
    inet6 fe80::dfda:269c:c79d:47b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Teacher: ens19 still has 2 ip's inet 10.134.177.24/21 brd 10.134.183.255 scope global dynamic noprefixroute ens19 valid_lft 1636sec preferred_lft 1636sec I couldn't find the file for ens19 so wasn't able to change. I have the file instructors.nmconnection here is the interface-name specified as ens19 and an ip addr set, only one.

Nice -challenging- extra:

d. Write and schedule a script which checks at intervals if some team uploaded a new picture to the home folder of the server backend.

```
#!/bin/bash

# This script checks for new images in the home folder of the
server backend and performs an action if there are any.

# Login to the server backend with ssh and run inotifywait
sshpass -p initial01 ssh team@192.168.1.250
"./bin/inotifywait -m ~ -e create -e moved_to" | while read
dir action file; do

# Send a push notification to your device if a new file has
been added to the folder with curl
curl -s \
    --form-string "token=anq7r5kadmify7gi9yp3t5hbq3xddc" \
    --form-string "user=uydnbw2k8b2egfoad6nsfuvihraio4" \
    --form-string "message=A new file '$file' has been added
to '$dir'via '$action'" \
    https://api.pushover.net/1/messages.json
done
```

Comment: First I went with root to the /opt/script folder. Here I created using vim a file named "newpicture.sh". Then I continued making the bash script. To schedule the script I used crontab. I first made the script executable then I edited the crontab file. I entered with the command "crontab -e" and added the line "0 0 */2 * * /opt/script/newpicture.sh". This means that the script will run at 0 minutes and 0 hours every two days (/2) of every month() and every day of week (*).

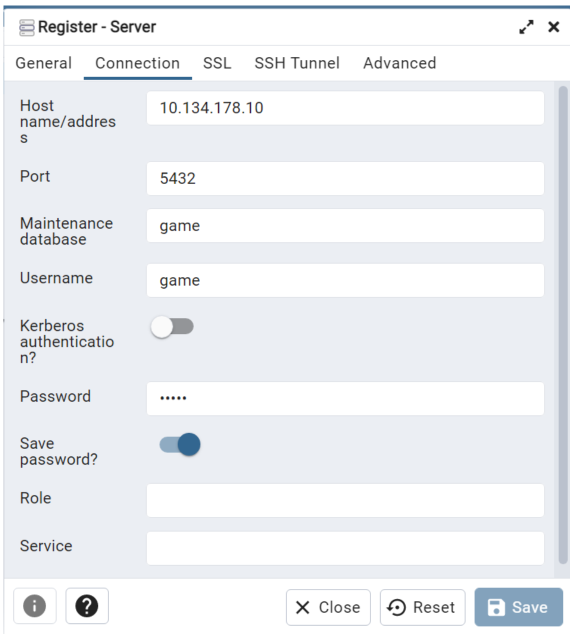
e. When a new picture arrives, send a message to your phone using the "pushover service" (free trial)

Comment: First I needed to install and create an account for pushover. I also needed to apply for an API code. After filling in my name and uploading an Icon I got one. After that I needed to install curl on my server. This was done by the command “yum install curl” and afterwards checked by “ rpm -qa | grep curl”. This is a tool that can send HTTP requests from the command line. In the bash script I added the part of curl to send messages.
Teacher: errors in the cron jobs: /opt/script/newpicture.sh: line 6: sshpass: command not found) I installed inotifywait-tools and added the right directory to the script code. Works only as a user not as root.

4. Install PostgreSQL

Executed: Jasper **Reviewed:**

a. Follow the instructions in *Assignment 5: Network Services* from the infrastructure 1 course. (Steps were taken from the document that is linked.)
Comment:
- To install package postgresql-server “sudo dnf install postgresql-server -y”.
- After that initialize the postgresql service with “sudo /usr/bin/postgresql-setup –initdb”.
- Check if postgresql service is running: systemctl status postgresql.
- Enable and start the postgresql service, if necessary: systemctl start postgresql and systemctl enable postgresql
- Edit the configuration file /var/lib/pgsql/data/postgresql.conf and add the line: #listen_addresses = 'localhost' # what IP address(es) … listen_addresses = '10.134.178.10''
- Edit the configuration file /var/lib/pgsql/data/pg_hba.conf and add the line: host all all all md5
  We used three times all bc ip address 10.0.0.0/8 gave errors.
- Add user ‘game’ and create database ‘game’ see step b
- Finally we’re ready to try to make a new pgadmin connection from your host operating system to your new database server.
- If If you have problems connecting, check if there is a firewall running: ”sudo firewall-cmd --state”
- Check the services that are allowed to be accessed by the firewall: ”sudo firewall-cmd --list-services”
- If there is a firewall running and the postgresql is not listed in the allowed services, then add the postgresql service to the firewall settings: ”sudo firewall-cmd --add-service=postgresql --permanent”
- Reload the firewall service: “sudo firewall-cmd --reload”
- Check that the postgres service is accessible by listing the firewall services: “sudo firewall-cmd --list-services”


Register - Server

General | Connection | SSL | SSH Tunnel | Advanced

Host name/address	10.134.178.10
Port	5432
Maintenance database	game
Username	game
Kerberos authentication?	off
Password	•••••
Save password?	on
Role	
Service	

b. Create a new user "game" with password '7sur7' on your postgresql server and grant it all permissions to a new database "game".
Use this user and database for your game.
Comment:
- Switch to the postgres user: “sudo su - postgres”
- Run the createuser command with the -P option to prompt for a password: “createuser -P game”
- Enter ‘7sur7’ as the password when prompted
- Run the createdb command to create a new database ‘game’: “createdb game”
- Run the psql command to connect to the database: “psql game”
- Run the GRANT command to give all permissions to the user ‘game’: “GRANT ALL ON DATABASE game TO game;”

5. Webserver

Executed: Vlad **Reviewed:** Jasper

a. Install apache webserver on your server.

```
root@team10int2: ~
# yum install httpd
Last metadata expiration check: 1:00:47 ago on Mon 13 Feb
2023 03:12:34
Dependencies resolved.
============================================================
Package        Arch       Version       Repository     Size
============================================================
Installing:
 httpd         x86_64    2.4.53-7.el9  appstream     48 k

[...]

root@team10int2: ~
# service httpd start
Redirecting to /bin/systemctl start httpd.service
```

Comment: Installed apache using “yum”. After that we started the server using “service httpd start”.

b. Which command can you use in a terminal to see that the web server is accepting connections on port 80?

```
root@team10int2: ~
# ss -tuwln | grep LISTEN
tcp   LISTEN 0    128         0.0.0.0:22        0.0.0.0:*
tcp   LISTEN 0    128               [::]:22           [::]:*
```

Comment: Checked what ports are listening (accepting connections)

c. Make sure the web server starts automatically on reboot

```
root@team10int2: ~
# systemctl enable httpd.service
Created symlink
/etc/systemd/system/multi-user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
```

d. Open the firewall to accept connections on port 80.

```
root@team10int2: ~
# firewall-cmd --zone=public --permanent --add-service=http
success

root@team10int2: ~
# firewall-cmd --reload
success
```

e. Copy your website to the new web server (in the default DocumentRoot)

```
➜  ~ rsync -av Website10/ Vlad@10.134.178.10:/home/Vlad
building file list ... done
./
favicon.ico
index.html
```

```
css/
css/global.css
css/index.css
img/
img/background.jpg
img/cards.svg
js/

sent 677591 bytes  received 1080 bytes  452447.33 bytes/sec
total size is 673732  speedup is 0.99
```

Comment: Modified it and now runs with git. Easier to pull the project and add changes.

f. Change file permissions and ownership to root apache rw-r--r–

```
Vlad@team10int2: /var/www
$ sudo chown root:apache html/*

Vlad@team10int2: /var/www
$ sudo chmod 644 html/*
```

Comment: Changed the ownership of the website files to root and the group ownership to apache. Changed the permissions of the website files to rw-r–r–

Change folder permissions and ownership to root apache rwxr-xr-x

```
Vlad@team10int2: /var/www
$ sudo chown root:apache html

Vlad@team10int2: /var/www
$ sudo chmod 755 html
```

g. Gitlab connection to update the website on the server
Comment: After finishing all infra tasks we modified this part so we can pull our site from git. This is easier to update our side. Otherwise we would have to delete and upload everything every time. Here is how we did it:
First installed git on the server afterwards, "**sudo git clone --single-branch *LINK TO GIT***" and now git pull and git update are active to get changes.

Nice extra:

h. You might accidentally also upload contents from the .git folder to the web folder on the server. To avoid serving up contents from the .git folder, make a .htaccess file, in which you redirect requests for the .git folder to a "404" error page.

Comment: I created a .htaccess file in our website's root directory and added the following line to it: "RedirectMatch 404 /\\.git" This will hide not only the contents of our Git repo but also its existence.
After that I checked that the permissions of the file were 755, user: root and group: apache.

6. RSYSLOG configuration

Executed: Jasper **Reviewed:**

a. Configure the syslog facility `local6` to log messages to the logfile /var/log/team.log
Comment: Cd to /var/log directory. Create a team.log file with the command "touch team.log".

b. Use the drop in directory rsyslog.d to add this new configuration to the syslog service.
Comment: Cd to /etc/rsyslog.d/ directory and touch local6.conf to create file. After that I added the following content: "local6.* /var/log/team.log". After creating this file, I restarted rsyslog service for the changes to take effect.

c. Test the newly configured syslog facility using the command: `logger`
Comment: To test the newly configured syslog facility I used the command "logger -p local6.info "Test message".

```
# cat /var/log/team.log
Mar  7 18:14:03 team10int2 root[83084]: Test message
```

d. Make the system journal database persistent so that it is available after reboot.
Comment: To make system journal database persistent so that it is available after reboot, you need to create /var/log/journal directory and assign proper permissions.

```
sudo mkdir -p /var/log/journal
sudo systemd-tmpfiles --create --prefix /var/log/journal
```

After creating this directory and assigning proper permissions, restart systemd-journald service for changes to take effect.
Teacher: Your system journal is not persistent on reboot. I edited the /etc/systemd/journald.conf file. Here I changed the line #Storage=auto to #Storage=persistent. After that I restarted the systemd-journald to let the changes take effect.

7. Create a Backup script

Executed: Panagiotis/Jasper **Reviewed:** Jasper

a. Create a bash backup script for your server: `/opt/script/backup.sh`
Change them to the right permissions and ownership.

Comment: First, we used the command "touch /opt/script/backup.sh" to create the bash script. After that we changed the ownership with "chown root:team10 backup.sh" and the permissions with "chmod u=rwx,g=rw,o=r backup.sh". The permissions of the /opt/script folder were already set in a previous assignment. (Extra of task 1 and 3).

Comment: I used the command "vim backup.sh" to edit the backup script. We made an agreement with team 1 to use their server and ours. For the account on team1 server, I had to create a passwordless ssh access to let the backup script run automatically(See task 2). Then I created the script taking the requirements in consideration.

Script:

```
#!/bin/bash

#Define variables
BACKUP_DIR="/var/www/html"
BACKUP_FILE="backup-$(date +%Y-%m-%d-%H-%M-%S).tar.gz"
EXTERNAL_DISK="/dev/sdb1"
MOUNT_POINT="/mnt/backup"
REMOTE_SERVER="10.134.178.1"
REMOTE_USER="backupuser"
REMOTE_DIR="backup"

# Check if script is started with root credentials
if [ $(id -u) -ne 0 ]; then
    logger -p local6.err -t backup "Error: Script needs to be
run as root."
    exit
fi

# Check command line parameter
if [ "$#" -eq 0 ]; then
    logger -p local6.err -t backup "Error: Command line
parameter required (now or restore)."
    exit
fi
if [ "$1" == "now" ]; then
    logger -p local6.info -t backup "INFO: Starting backup
process..."

    # Create backup
    tar czf "$BACKUP_FILE" "$BACKUP_DIR"

    # Calculate hash values
    md5sum "$BACKUP_FILE" > "$BACKUP_FILE.md5"
    sha256sum "$BACKUP_FILE" > "$BACKUP_FILE.sha256"
    sha512sum "$BACKUP_FILE" > "$BACKUP_FILE.sha512"

    # Check if external disk is already mounted
    if mountpoint -q "$MOUNT_POINT"; then
        logger -p local6.info -t backup "INFO: External disk
already mounted."
    else
        # Mount external disk
        mount "$EXTERNAL_DISK" "$MOUNT_POINT"
        if [ $? -ne 0 ]; then
            logger -p local6.err -t backup "Error: Failed to
mount external disk."
            exit
        fi
    fi

    # Copy backup file to external disk
    cp "$BACKUP_FILE" "$MOUNT_POINT"
    cp "$BACKUP_FILE.md5" "$MOUNT_POINT"
    cp "$BACKUP_FILE.sha256" "$MOUNT_POINT"
    cp "$BACKUP_FILE.sha512" "$MOUNT_POINT"

    # Unmount external disk
    umount "$MOUNT_POINT"
    if [ $? -ne 0 ]; then
        logger -p local6.err -t backup "Error: Failed to
unmount external disk."
        exit
    fi

    # Copy backup file to remote server
    scp "$BACKUP_FILE" "$BACKUP_FILE.md5"
"$BACKUP_FILE.sha256" "$BACKUP_FILE.sha512"
"$REMOTE_USER@$REMOTE_SERVER:$REMOTE_DIR"
    if [ $? -ne 0 ]; then
        logger -p local6.err -t backup "Error: Failed to copy
backup file to remote server."
        exit
    fi

  # Calculate hash value of remote backup file

  REMOTE_MD5=$(ssh "$REMOTE_USER@$REMOTE_SERVER" "md5sum
$REMOTE_DIR/$BACKUP_FILE")
  REMOTE_MD5=$(echo "$REMOTE_MD5" | cut -d ' ' -f 1)
  if [ "$REMOTE_MD5" == "$(cat $BACKUP_FILE.md5 | cut -d ' ' -f 1)" ]; then
      logger -p local6.info -t backup "Backup completed successfully."
      # Delete backup file from current directory
      rm "$BACKUP_FILE" "$BACKUP_FILE.md5" "$BACKUP_FILE.sha256" "$BACKUP_FILE.sha512"
  else
      logger -p local6.err -t backup "Error: Remote backup file hash value does not match local hash value."
      exit
  fi
fi
if [ "$1" == "restore" ]; then
    logger -p local6.info -t backup "INFO: Starting restore..."

    # Check hash value of backup file
    md5sum -c "$BACKUP_FILE.md5" > /dev/null 2>&1
    if [ $? -ne 0 ]; then
        logger -p local6.err "Error: Backup file hash value does not match."
        exit
    fi

    # Restore files to a folder
    RESTORE_DIR="/var/www/html"
    tar xzf "$BACKUP_FILE" -C "$RESTORE_DIR"
    logger -p local6.info -t backup "INFO: Restore completed successfully."
fi
```

Teacher: /opt/script/backup.sh: line 88: syntax error: unexpected end of file)
We needed to add a fi statement at the end in the script, and also changed the REMOTE_DIR from /backup to just backup.
Teacher: Check the permissions of the script folder (sgid) The permissions of the script folder were already right, changed permissions of some scripts. We gave the folder sgid permissions with the line “chmod g+s script/”

8. Schedule the backup script

Executed: Jasper **Reviewed:**

a. Use the crontab of user root to schedule the backup script to run at an appropriate time.

Comment: To schedule the script I used crontab. I first made the script executable then I edited the crontab file. I entered with the command “crontab -e” and added the line “0 0 * * 0 /opt/script/backup.sh now”. This means that the script will run every sunday at 12am.
Teacher: Error in cron job, we added “now” to the end of the cron job to let the script know we want to run it.
Teacher: Make a backup everyday ruChanged the crontab to 0 9 * * * /opt/script/wrapper.sh so it will make a backup everyday at 9am.

b. Make a wrapper script which can save different versions of the backup, and deletes older ones. Implement a backup rotation scheme.
Comment: I created a wrapper script that runs the backup.sh script and afterwards checks how many files are in there that end with .tar.gz. If there are more than three it deletes the oldest one with his hash files. This is also done on the external disk. The crontab is changed from /opt/script/backup.sh now to /opt/script/wrapper.sh so it will run the wrapper script that calls the backup.sh script.

Wrapper script:
```
#!/bin/bash

# create new backup
sh /opt/script/backup.sh now

# delete oldest backups if more than 3 backups exist on the
remote server
num_backups=$(ssh backupuser@10.134.178.1 "ls backup/*.tar.gz
| wc -l")
if [ $num_backups -gt 3 ]; then
    ssh backupuser@10.134.178.1 "ls -1t backup/*.tar.gz |
tail -n +4 | xargs -I {} sh -c 'rm -- {}; rm -- {}.md5; rm --
{}.sha256; rm -- {}.sha512'"
fi

# delete oldest backups if more than 3 backups exist on the
external disk
    disk_device="/dev/sdb1"
    mount_point="/mnt/backup"
    # Mount the disk
    mount "$disk_device" "$mount_point"
    # Delete
    num_backups=$(ls "$mount_point"/*.tar.gz | wc -l)
    if [ $num_backups -gt 3 ]; then
        for file in $(ls -1t "$mount_point"/*.tar.gz | tail
-n +4); do
            rm -- "$file"
            rm -- "${file}.md5"
            rm -- "${file}.sha256"
            rm -- "${file}.sha512"
        done
    fi
    # Unmount the disk
    umount "$disk_device"
```

9. Demo backup script/ wrapper script