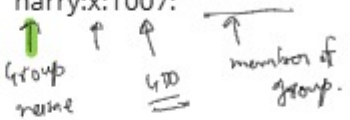
date
ls
pwd
mkdir
cd
cat
nano
rm
rmdir
adduser
cp
mv
userdel
groupadd
groupdel

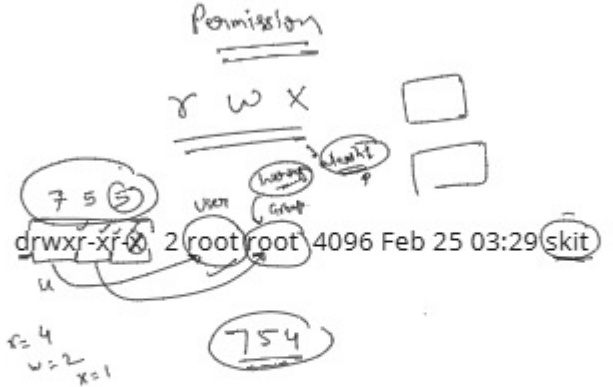
cat /etc/group

harry:x:1007:

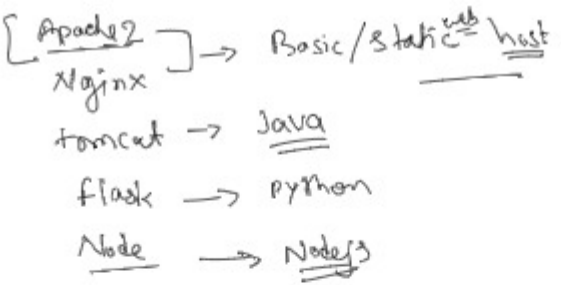


sudo usermod -aG harry akash1
ubuntu@ip-172-31-1-183:~$ tail /etc/group
jenkins:x:113:
u1:x:1001:
u2:x:1002:
team:x:1003:
ssl-cert:x:114:
docker:x:115:jenkins,ubuntu
hey22:x:1004:
akash:x:1005:
akash1:x:1006:
harry:x:1007:akash1

Task ⇒ how to remove user from group



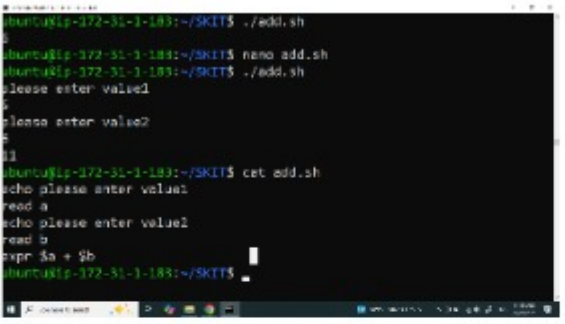
ifconfig
or
ip a

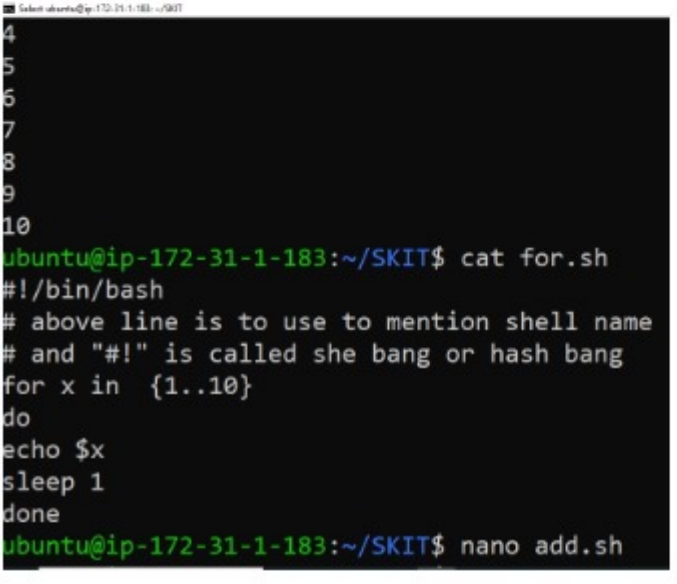


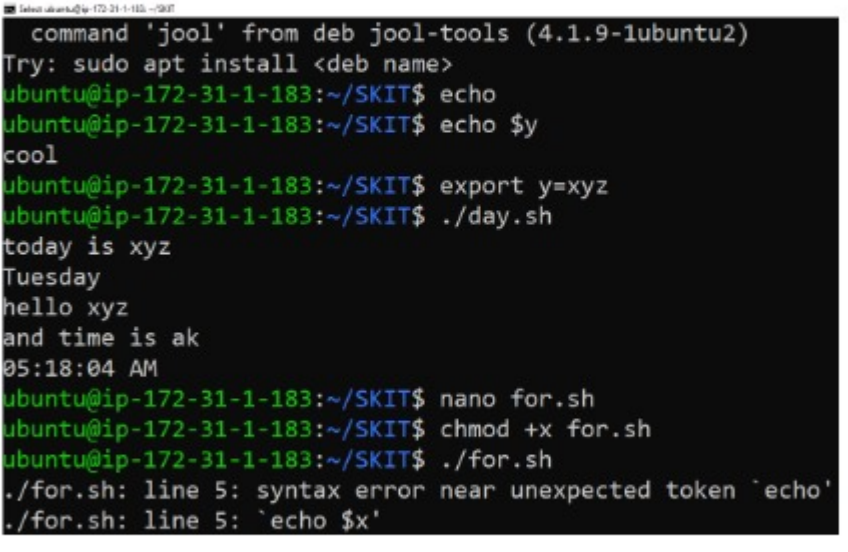
http://3.109.155.226/

http://jenkins.akdev.live

```
sudo adduser harry
 1312  cat /etc/group
 1313  sudo usermod -aG harry akash1
 1314  tail /etc/group
 1315  cd /
 1316  pwd
 1317  sudo mkdir skit
 1318  ls -l
 1319  cd /skit
 1320  cd ..
 1321  sudo chmod 754 skit
 1322  ls -ld skit
 1323  cd skit
 1324  history
 1325  sudo chmod 755 skit
 1326  cd skit
 1327  ip a
 1328  sudo apt install apache2
 1329  ping google.com
 1330  cd /var/www/html/
 1331  ls
 1332  sudo rm *
 1333  sudo rm -r  *
 1334  ls
 1335  touch index.html
 1336  sudo touch index.html
 1337  ls
 1338  sudo nano index.html
 1339  curl ifconfig.me
 1340  sudo nano index.html
 1341  systemctl status apache2
 1342  curl 127.0.0.1
 1343  ls
 1344  nano hello.html
 1345  sudo nano hello.html
 1346  curl 127.0.0.1/hello.html
 1347  curl 127.0.0.1
 1348  ls
 1349  unzip oxer.zip
 1350  sudo unzip oxer.zip
 1351  ls
 1352  cd oxer-html/
 1353  ls
 1354  cd ..
 1355  ls
 1356  sudo cp oxer-html/.  .\
 1357  sudo cp oxer-html/.  .
 1358  sudo cp -r oxer-html/.  .
 1359  ls
```


ubuntu@ip-172-31-1-183:~/SKIT$ ./add.sh
a
ubuntu@ip-172-31-1-183:~/SKIT$ nano add.sh
ubuntu@ip-172-31-1-183:~/SKIT$ ./add.sh
please enter value1
a
please enter value2
b
11
ubuntu@ip-172-31-1-183:~/SKIT$ cat add.sh
echo please enter value1
read a
echo please enter value1
read b
expr $a + $b
ubuntu@ip-172-31-1-183:~/SKIT$

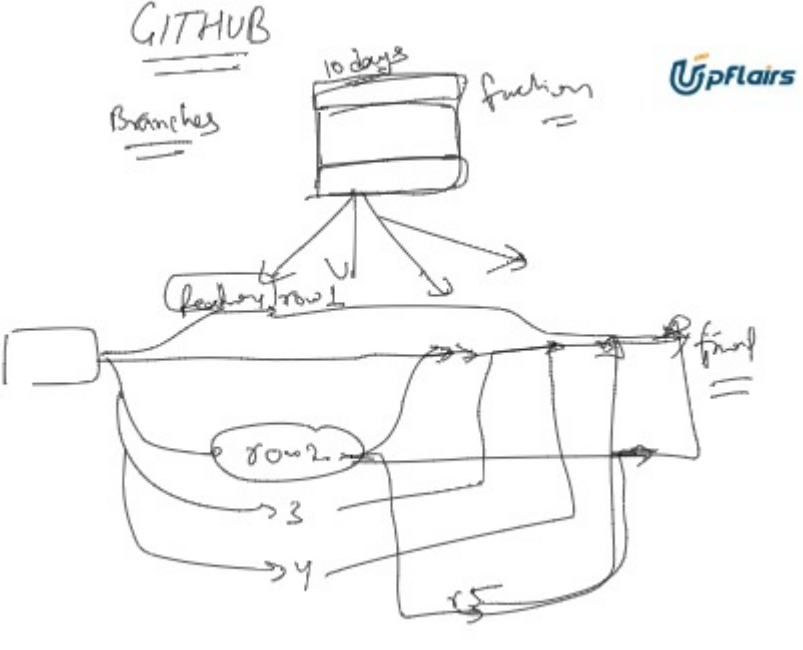

4
5
6
7
8
9
10
ubuntu@ip-172-31-1-183:~/SKIT$ cat for.sh
#!/bin/bash
above line is to use to mention shell name
and "#!" is called she bang or hash bang
for x in {1..10}
do
echo $x
sleep 1
done
ubuntu@ip-172-31-1-183:~/SKIT$ nano add.sh


command 'jool' from deb jool-tools (4.1.9-1ubuntu2)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-1-183:~/SKIT$ echo
ubuntu@ip-172-31-1-183:~/SKIT$ echo $y
cool
ubuntu@ip-172-31-1-183:~/SKIT$ export y=xyz
ubuntu@ip-172-31-1-183:~/SKIT$ ./day.sh
today is xyz
Tuesday
hello xyz
and time is ak
05:18:04 AM
ubuntu@ip-172-31-1-183:~/SKIT$ nano for.sh
ubuntu@ip-172-31-1-183:~/SKIT$ chmod +x for.sh
ubuntu@ip-172-31-1-183:~/SKIT$ ./for.sh
./for.sh: line 5: syntax error near unexpected token `echo'
./for.sh: line 5: `echo $x'

```
00  export y=xyz
 1401  ./day.sh
 1402  nano for.sh
 1403  chmod +x for.sh
 1404  ./for.sh
 1405  nano for.sh
 1406  ./for.sh
 1407  nano for.sh
 1408  ./for.sh
 1409  cat for.sh
 1410  nano for.sh
 1411  ./for.sh
 1412  cat for.sh
 1413  nano add.sh
 1414  chmod +x add.sh
 1415  ./add.sh
 1416  nano add.sh
 1417  ./add.sh
 1418  nano add.sh
 1419  ./add.sh
 1420  nano add.sh
 1421  ./add.sh
 1422  nano add.sh
 1423  ./add.sh
 1424  cat add.sh
 1425  ./add.sh
 1426  nano add.sh
 1427  ./add.sh
 1428  cat add.sh
 1429  crontab -e
 1430  date
 1431  date >myfile.txt
 1432  cat myfile.txt
 1433  date >myfile.txt
 1434  cat myfile.txt
 1435  date >>myfile.txt
 1436  cat myfile.txt
 1437  date >>myfile.txt
 1438  cat myfile.txt
 1439  crontab -e
 1440  cat myfile.txt
 1441  crontab -e
 1442  cat myfile.txt
 1443  tail -f myfile.txt
 1444  crontab -e
```

1. **1st position:** Minute
2. **2nd position:** Hour
3. **3rd position:** Day of the month
4. **4th position:** Month
5. **5th position:** Day of the week

2 * * * * date >>/home/ubuntu/SKIT/myfile.txt


GITHUB
Branches
10 days
feature
UpFlairs
feature row1
final
row2
3
4
5

Upflairs

github.com/gaur95/skit_workshop

◆ Linux File Permissions

1. File Permission Basics

Each file and directory in Linux has three types of permissions for three different categories:

1. **Owner (u) –** The user who owns the file
2. **Group (g) –** Users in the same group as the file owner
3. **Others (o) –** Everyone else

2. Permission Types

SymbolPermissionNumeric ValuerRead4wWrite2xExecute1

Example: -rwxr-xr--

1. **Owner:** rwx (read, write, execute)
2. **Group:** r-x (read, execute)
3. **Others:** r-- (read only)

3. Changing Permissions & Ownership

1. **Check Permissions:** ls -l filename
2. **Change Permissions:** chmod 755 filename
3. **Change Ownership:** chown user:group filename
4. **Recursive Permission Change:** chmod -R 755 directory/

1. What is Apache Web Server?

Apache is an open-source, cross-platform web server that serves websites and applications over HTTP/HTTPS.

2. Varieties of Apache Web Server

1. **Apache HTTP Server (httpd)** – Standard version for Linux servers.
2. **Apache Tomcat** – Serves Java-based applications (JSP & Servlets).
3. **Apache Traffic Server** – High-performance caching proxy server.
4. **Apache Caddy** – Modern web server with automatic HTTPS.
5. **Apache Nginx (Reverse Proxy)** – Used as a load balancer and proxy with Apache for better performance.

3. Basic Apache Commands

1. Install Apache: sudo apt install apache2 -y
2. Start Service: sudo systemctl start apache2
3. Enable Auto Start: sudo systemctl enable apache2
4. Restart Service: sudo systemctl restart apache2
5. Check Status: sudo systemctl status apache2

Basic Shell Scripting

1. What is Shell Scripting?

A **shell script** is a file containing a sequence of commands that automate tasks in Linux.

2. Creating a Shell Script

Create a script file: nano script.sh

Add the shebang (#!), which tells the system to use Bash:

```
#!/bin/bash
echo "Hello, World!"
```

Save and exit (CTRL + X → Y → ENTER).

Make it executable: chmod +x script.sh

Run the script: ./script.sh

3. Variables in Shell Script

```
#!/bin/bash
name="Akash"
echo "Hello, $name!"
```

4. Conditional Statements

```
#!/bin/bash
if [ $1 -gt 10 ]; then
 echo "Number is greater than 10"
else
   echo "Number is 10 or less"
fi
```

5. Looping

1. **For Loop**

```
for i in {1..5};
do
   echo "Iteration $i"
done
```

ex:-

```
#!/bin/bash
for i in {1..5}; do
   echo "Iteration $i"
done
```

Cron Job in Linux (Task Scheduling)

A **cron job** is a scheduled task in Linux that runs automatically at specific intervals. It is managed by the **cron daemon (crond)**.

📌 **1. Basic Cron Syntax**

A cron job follows this format:

```
* * * * * command_to_execute
| | | | |
| | | | └── Day of the week (0 - 7) [0 & 7 = Sunday]
| | | └──── Month (1 - 12)
| | └────── Day of the month (1 - 31)
| └──────── Hour (0 - 23)
└────────── Minute (0 - 59)
```

Example:

```
30 2 * * * /home/user/backup.sh
```

🔹 **Runs backup.sh at 2:30 AM daily.**

📌 **2. How to Edit Cron Jobs**

To edit cron jobs for the current user:

```
crontab -e
```

This opens the cron file in the default editor (like nano or vim).

📌 **3. Examples of Cron Jobs**

✅ **Run a script every minute**

```
* * * * * /home/user/script.sh
```

✅ **Run a script every day at 1 AM**

```
0 1 * * * /home/user/daily.sh
```

✅ **Run a script every Monday at 6 AM**

```
0 6 * * 1 /home/user/monday-task.sh
```

What is Git?

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

1. Tracking code changes
2. Tracking who made changes
3. Coding collaboration

What does Git do?

1. Manage projects with **Repositories**
2. **Clone** a project to work on a local copy
3. Control and track changes with **Staging** and **Committing**
4. **Branch** and **Merge** to allow for work on different parts and versions of a project
5. **Pull** the latest version of the project to a local copy
6. **Push** local updates to the main project

	 GitHub
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the **Git** repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

GIT BRANCH