Cloud Computing (Lab 6)

Name : Sarosh Majeed

Reg no : 2023-BSE-059

Task 1 – Switch to root with su - and back to a normal user

Goal: Demonstrate switching to the root account using su - and exiting back to your normal user.

Set a root password (Ubuntu root is disabled by default; this enables su - temporarily for the lab):

sudo passwd root

Enter a temporary root password for the lab

```
sarosh@ubuntu:~$ whoami
sarosh
sarosh@ubuntu:~$ sudo passwd root
[sudo] password for sarosh:
New password:
Retype new password:
passwd: password updated successfully
sarosh@ubuntu:~$ _
```

Switch to root and verify:

su - whoami id

```
sarosh@ubuntu:~$ su -
Password:
root@ubuntu:~# whoami
root
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~#
```

Switch back to your normal user: exit whoami

```
root@ubuntu:~# exit
logout
sarosh@ubuntu:~$ whoami
sarosh
sarosh@ubuntu:~$
```

Task 2 – Create user tom and verify in passwd/group/shadow

Goal: Create a user named tom, then verify the account in system files.

Create user tom (interactive, sets password and home directory): sudo adduser tom

```
sarosh@ubuntu:~$ sudo adduser tom
Adding user `tom' ...
Adding new group `tom' (1002) ...
Adding new user `tom' (1002) with group `tom' ...
Creating home directory `/home/tom' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for tom
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
sarosh@ubuntu:~$
```

Verify tom in system files (view and visually confirm presence): cat /etc/passwd

```
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
syslog:x:106:113::/home/syslog:/usr/sbin/nologin
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
tss:x:109:116:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:110:117::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:111:118:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:113:65534::/run/sshd:/usr/sbin/nologin
sarosh:x:1000:1000:SAROSH MAJEED:/home/sarosh:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
lab4user:x:1001:1001:,,,:/home/lab4user:/bin/bash
rtkit:x:114:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
avahi:x:115:120:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
saned:x:116:123::/var/lib/saned:/usr/sbin/nologin
colord:x:117:124:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
pulse:x:118:125:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
xrdp:x:119:128::/run/xrdp:/usr/sbin/nologin
lightdm:x:120:129:Light Display Manager:/var/lib/lightdm:/bin/false
tom:x:1002:1002:,,,:/home/tom:/bin/bash
sarosh@ubuntu:~$
```

cat /etc/group

sudo cat /etc/shadow

```
tom:$y$j9T$mMPqNtEUM63mw8ihStgMM0$7nbUXKghVFfMv33irqLbUY.yJaaUqTbCUvQAXsJcBz6:20396:0:99999:7:::
sarosh@ubuntu:~$ _
```

Notes: /etc/shadow stores password hashes (not plaintext). You must use sudo to read it.

Task 3 – Create groups; change tom's primary and secondary groups

Goal: Create groups developer, devops, and designer. Change tom's primary group and manage secondary groups.

Create groups and verify by viewing /etc/group (visually confirm entries exist):

sudo groupadd developer

sudo groupadd devops

sudo groupadd designer

cat /etc/group

```
tom:x:1002:
developer:x:1003:
devops:x:1004:
designer:x:1005:
sarosh@ubuntu:~$
```

Change tom's primary group to designer and verify:

sudo usermod -g designer tom

id tom

```
sarosh@ubuntu:~$ sudo usermod -g designer tom
sarosh@ubuntu:~$ id tom
uid=1002(tom) gid=1005(designer) groups=1005(designer)
sarosh@ubuntu:~$ _
```

Add secondary groups developer and devops to tom and verify:

sudo usermod -aG developer,devops tom

id tom

groups tom

```
sarosh@ubuntu:~$ sudo usermod -aG developer, devops tom
usermod: group ' ' does not exist
sarosh@ubuntu:~$ sudo usermod -aG developer,devops tom
sarosh@ubuntu:~$ id tom
uid=1002(tom) gid=1005(designer) groups=1005(designer),1003(developer),1004(devops)
sarosh@ubuntu:~$ groups tom
tom : designer developer devops
sarosh@ubuntu:~$
```

Replace all secondary groups so only tom (user's own group) remains and verify:

sudo usermod -G tom tom

id tom

groups tom

```
sarosh@ubuntu:~$ sudo usermod -G tom tom
sarosh@ubuntu:~$ id tom
uid=1002(tom) gid=1005(designer) groups=1005(designer),1002(tom)
sarosh@ubuntu:~$ groups tom
tom : designer tom
sarosh@ubuntu:~$
```

Task 4 – Create/delete users (Jerry, Scooby) and groups (jolly, anime)

Goal: Create users using both adduser and useradd, demonstrate login/password/home directory differences, then delete users/groups.

Create users: **sudo adduser Jerry** **sudo useradd Scooby**

```
sarosh@ubuntu:~$ sudo adduser jerry
Adding user `jerry' ...
Adding new group `jerry' (1006) ...
Adding new user `jerry' (1003) with group `jerry' ...
Creating home directory `/home/jerry' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for jerry
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
sarosh@ubuntu:~$ sudo useradd scooby
sarosh@ubuntu:~$
```

Try to log in as Scooby immediately (expected authentication failure because there is no password yet): **su – Scooby**

```
sarosh@ubuntu:~$ su - scooby
Password:
su: Authentication failure
sarosh@ubuntu:~$
```

Set a password for Scooby: **sudo passwd Scooby**

```
sarosh@ubuntu:~$ sudo passwd scooby
New password:
Retype new password:
passwd: password updated successfully
sarosh@ubuntu:~$ _
```

Try logging in as Scooby again (home directory still missing; expect a message such as "No directory, logging in with HOME=/"): **su - Scooby**

```
sarosh@ubuntu:~$ su - scooby
Password:
su: warning: cannot change directory to /home/scooby: No such file or directory
$ _
```

Show that Scooby's home directory does not exist yet and what /etc/passwd says:

exit

cat /etc/passwd

ls -ld /home/Scooby

```
tom:x:1002:1005:,,,:/home/tom:/bin/bash
jerry:x:1003:1006:,,,:/home/jerry:/bin/bash
scooby:x:1004:1007::/home/scooby:/bin/sh
sarosh@ubuntu:~$ ls -ld /home/scooby
ls: cannot access '/home/scooby': No such file or directory
sarosh@ubuntu:~$ _
```

Manually create Scooby's home directory and set proper ownership and permissions:

sudo mkdir -p /home/Scooby

sudo chown Scooby:Scooby /home/Scooby

sudo chmod 750 /home/Scooby

ls -ld /home/Scooby

```
sarosh@ubuntu:~$ sudo mkdir -p /home/scooby
sarosh@ubuntu:~$ sudo chown scooby:scooby /home/scooby
sarosh@ubuntu:~$ sudo chmod 750 /home/scooby
sarosh@ubuntu:~$ ls -ld /home/scooby
drwxr-x--- 2 scooby scooby 4096 Nov  4 20:20 /home/scooby
sarosh@ubuntu:~$
```

Log in as Scooby again and verify you land in the correct home directory:

su - Scooby

pwd

ls -la

```
sarosh@ubuntu:~$ su - scooby
Password:
$ pws
-sh: 1: pws: not found
$ pwd
/home/scooby
$ ls -la
total 8
drwxr-x--- 2 scooby scooby 4096 Nov  4 20:20 .
drwxr-xr-x 7 root   root   4096 Nov  4 20:20 ..
$
```

Verify users in system files and observe shell of Scooby: exit cat /etc/passwd

```
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
syslog:x:106:113::/home/syslog:/usr/sbin/nologin
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
tss:x:109:116:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:110:117::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:111:118:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:113:65534::/run/sshd:/usr/sbin/nologin
sarosh:x:1000:1000:SAROSH MAJEED:/home/sarosh:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
lab4user:x:1001:1001:,,,:/home/lab4user:/bin/bash
rtkit:x:114:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
avahi:x:115:120:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
saned:x:116:123::/var/lib/saned:/usr/sbin/nologin
colord:x:117:124:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
pulse:x:118:125:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
xrdp:x:119:128::/run/xrdp:/usr/sbin/nologin
lightdm:x:120:129:Light Display Manager:/var/lib/lightdm:/bin/false
tom:x:1002:1005:,,,:/home/tom:/bin/bash
jerry:x:1003:1006:,,,:/home/jerry:/bin/bash
scooby:x:1004:1007::/home/scooby:/bin/sh
sarosh@ubuntu:~$
```

Change the shell from /bin/sh to /bin/bash

sudo usermod -s /bin/bash Scooby

su - Scooby

```
sarosh@ubuntu:~$ sudo usermod -s /bin/bash scooby
sarosh@ubuntu:~$ su - scooby
Password:
scooby@ubuntu:~$
```

Create groups:

sudo addgroup jolly

sudo groupadd anime

```
scooby@ubuntu:~$ exit
logout
sarosh@ubuntu:~$ sudo addgroup jolly
Adding group `jolly' (GID 1008) ...
Done.
sarosh@ubuntu:~$ sudo addgroup anime
Adding group `anime' (GID 1009) ...
Done.
sarosh@ubuntu:~$
```

Verify groups: cat /etc/group

```
systemd-timesync:x:105:
input:x:106:
sgx:x:107:
kvm:x:108:
render:x:109:
lxd:x:110:sarosh
_ssh:x:111:
crontab:x:112:
syslog:x:113:
uuidd:x:114:
tcpdump:x:115:
tss:x:116:
landscape:x:117:
fwupd-refresh:x:118:
sarosh:x:1000:
lab4user:x:1001:
rtkit:x:119:
avahi:x:120:
netdev:x:121:
scanner:x:122:saned
saned:x:123:
colord:x:124:
pulse:x:125:
pulse-access:x:126:
ssl-cert:x:127:
xrdp:x:128:
lightdm:x:129:
nopasswdlogin:x:130:
tom:x:1002:tom
developer:x:1003:
devops:x:1004:
designer:x:1005:
jerry:x:1006:
scooby:x:1007:
jolly:x:1008:
anime:x:1009:
sarosh@ubuntu:~$ _
```

Delete groups and users:

sudo delgroup jolly

sudo groupdel anime

cat /etc/group

sudo deluser --remove-home Jerry

sudo userdel -r Scooby

```
Removing user `jerry' ...
Warning: group `jerry' has no more members.
Done.
sarosh@ubuntu:~$ sudo userdel -r scooby
userdel: scooby mail spool (/var/mail/scooby) not found
```

cat /etc/passwd

```
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
messagebus:x:104:
systemd-timesync:x:105:
input:x:106:
sgx:x:107:
kvm:x:108:
render:x:109:
lxd:x:110:sarosh
_ssh:x:111:
crontab:x:112:
syslog:x:113:
uuidd:x:114:
tcpdump:x:115:
tss:x:116:
landscape:x:117:
fwupd-refresh:x:118:
sarosh:x:1000:
lab4user:x:1001:
rtkit:x:119:
avahi:x:120:
netdev:x:121:
scanner:x:122:saned
saned:x:123:
colord:x:124:
pulse:x:125:
pulse-access:x:126:
ssl-cert:x:127:
xrdp:x:128:
lightdm:x:129:
nopasswdlogin:x:130:
tom:x:1002:tom
developer:x:1003:
devops:x:1004:
designer:x:1005:
sarosh@ubuntu:~$ _
```

Task 5 – Create user Student; create files; set owner/group; identify file types

Create Student: **sudo adduser Student**

```
sarosh@ubuntu:~$ sudo adduser student
Adding user `student' ...
Adding new group `student' (1006) ...
Adding new user `student' (1003) with group `student' ...
Creating home directory `/home/student' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for student
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
sarosh@ubuntu:~$ _
```

Switch to Student and create files:

su - Student

touch file1

mkdir -p dir1

touch dir1/file2

ls –l

```
sarosh@ubuntu:~$ su - student
Password:
student@ubuntu:~$ touch file1
student@ubuntu:~$ mkdir -p dir1
student@ubuntu:~$ touch dir1/file2
student@ubuntu:~$ ls -l
total 4
drwxrwxr-x 2 student student 4096 Nov  4 20:40 dir1
-rw-rw-r-- 1 student student    0 Nov  4 20:40 file1
student@ubuntu:~$
```

Change owner then group for file1 (separate commands):

sudo chown tom file1

ls -l file1

```
sarosh@ubuntu:~$ sudo chown tom /home/student/file1
sarosh@ubuntu:~$ ls -l /home/student/file1
ls: cannot access '/home/student/file1': Permission denied
sarosh@ubuntu:~$
```

sudo chgrp devops file1

ls -l file1

```
sarosh@ubuntu:~$ sudo chgrp devops /home/student/file1
sarosh@ubuntu:~$ ls -l /home/student/file1
ls: cannot access '/home/student/file1': Permission denied
sarosh@ubuntu:~$ ls -ld /home/student
drwxr-x--- 3 student student 4096 Nov  4 20:41 /home/student
sarosh@ubuntu:~$ _
```

Identify files/directories and show /dev/null:

ls -l

ls -l dir1

ls -l /dev/null

file file1 dir1 /dev/null

```
sarosh@ubuntu:~$ su - student
Password:
student@ubuntu:~$ ls -l /home/student
total 4
drwxrwxr-x 2 student student 4096 Nov  4 20:40 dir1
-rw-rw-r-- 1 tom     devops     0 Nov  4 20:40 file1
student@ubuntu:~$ ls -l /home/student/dir1
total 0
-rw-rw-r-- 1 student student 0 Nov  4 20:40 file2
student@ubuntu:~$ ls -l /dev/null
crw-rw-rw- 1 root root 1, 3 Nov  4 19:39 /dev/null
student@ubuntu:~$ file /home/student/file1 /home/student/dir1 /dev/null
/home/student/file1: empty
/home/student/dir1:  directory
/dev/null:           character special (1/3)
student@ubuntu:~$ _
```

Exit Student: exit

```
sarosh@ubuntu:~$
```

Task 6 – Change permissions using symbolic mode

Target file: ~/file1 (run these as the Student user)

Ensure Student and file present:

su - Student

cd ~

ls -l file1

```
sarosh@ubuntu:~$ su - student
Password:
student@ubuntu:~$ cd ~
student@ubuntu:~$ pwd
/home/student
student@ubuntu:~$ ls -l
total 4
drwxrwxr-x 2 student student 4096 Nov  4 20:40 dir1
-rw-rw-r-- 1 student student    0 Nov  4 20:40 file1
student@ubuntu:~$
```

Remove all permissions:

chmod -rwx file1

ls -l file1

```
student@ubuntu:~$ chmod -rwx file1
student@ubuntu:~$ ls -l file1
---------- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

Add read to all:

chmod +r file1

ls -l file1

```
student@ubuntu:~$ chmod +r file1
student@ubuntu:~$ ls -l file1
-r--r--r-- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$ _
```

Add execute to user:

chmod u+x file1

ls -l file1

```
student@ubuntu:~$ chmod u+x file1
student@ubuntu:~$ ls -l file1
-r-xr--r-- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

Add write to user and group:

chmod ug+w file1

ls -l file1

```
student@ubuntu:~$ chmod ug+w file1
student@ubuntu:~$ ls -l file1
-rwxrw-r-- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$ _
```

Remove all permissions (explicit):

chmod ugo-rwx file1

ls -l file1

```
student@ubuntu:~$ chmod ugo-rwx file1
student@ubuntu:~$ ls -l file1
---------- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$ _
```

Task 7 – Change permissions using "set" symbolic form (u= g= o=)

Ensure you are Student:

su - Student

cd ~

ls -l file1

```
student@ubuntu:~$ su - student
Password:
student@ubuntu:~$ cd ~
student@ubuntu:~$ ls -l file1
---------- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$ _
```

Set all to rwx:

chmod u=rwx,g=rwx,o=rwx file1

ls -l file1

```
student@ubuntu:~$ chmod u=rwx,g=rwx,o=rwx file1
student@ubuntu:~$ ls -l file1
-rwxrwxrwx 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

Remove execute from group and others:

chmod g=rw,o=rw file1

ls -l file1

```
student@ubuntu:~$ chmod g=rw,o=rw file1
student@ubuntu:~$ ls -l file1
-rwxrw-rw- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$ _
```

Remove all permissions:

chmod u=,g=,o= file1

ls -l file1

```
student@ubuntu:~$ chmod u=,g=,o= file1
student@ubuntu:~$ ls -l file1
---------- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

Task 8 – Change permissions using numeric (octal) mode

Ensure you are Student:

su - Student

cd ~

ls -l file1

```
student@ubuntu:~$ su - student
Password:
student@ubuntu:~$ cd ~
student@ubuntu:~$ ls -l file1
---------- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

Run each command and capture screenshot after each ls:

chmod 777 file1

ls -l file1

```
student@ubuntu:~$ chmod 777 file1
student@ubuntu:~$ ls -l file1
-rwxrwxrwx 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

chmod 700 file1

ls -l file1

```
student@ubuntu:~$ chmod 700 file1
student@ubuntu:~$ ls -l file1
-rwx------ 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

chmod 744 file1

ls -l file1

```
student@ubuntu:~$ chmod 744 file1
student@ubuntu:~$ ls -l file1
-rwxr--r-- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

chmod 640 file1

ls -l file1

```
student@ubuntu:~$ chmod 640 file1
student@ubuntu:~$ ls -l file1
-rw-r----- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$
```

chmod 664 file1

ls -l file1

```
student@ubuntu:~$ chmod 664 file1
student@ubuntu:~$ ls -l file1
-rw-rw-r-- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$ _
```

chmod 775 file1

ls -l file1

```
student@ubuntu:~$ chmod 775 file1
student@ubuntu:~$ ls -l file1
-rwxrwxr-x 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$ _
```

chmod 750 file1

ls -l file1

```
student@ubuntu:~$ chmod 750 file1
student@ubuntu:~$ ls -l file1
-rwxr-x--- 1 student student 0 Nov  4 20:40 file1
student@ubuntu:~$ _
```

Task 9 – Practice pipes, pagers, grep, and redirects with /var/log/syslog

less:

sudo cat /var/log/syslog | less

 # quit q #arrow keys to move

```
Nov  4 19:33:24 ubuntu systemd[1]: rsyslog.service: Sent signal SIGHUP to main process 893 (rsyslogd
) on client request.
Nov  4 19:33:24 ubuntu systemd[1]: Started Modem Manager.
Nov  4 19:33:24 ubuntu systemd[1]: logrotate.service: Deactivated successfully.
Nov  4 19:33:24 ubuntu systemd[1]: Finished Rotate log files.
Nov  4 19:33:24 ubuntu udisksd[900]: Acquired the name org.freedesktop.UDisks2 on the system message
 bus
Nov  4 19:33:24 ubuntu multipathd[562]: sda: triggering change event to reinitialize
Nov  4 19:33:24 ubuntu multipath: sda: failed to get sysfs uid: Invalid argument
Nov  4 19:33:24 ubuntu multipath: sda: failed to get sgio uid: No such file or directory
Nov  4 19:33:24 ubuntu multipathd[562]: sda: failed to get udev uid: No data available
Nov  4 19:33:24 ubuntu multipathd[562]: sda: failed to get path uid
Nov  4 19:33:24 ubuntu multipathd[562]: uevent trigger error
Nov  4 19:33:24 ubuntu snapd[896]: overlord.go:271: Acquiring state lock file
Nov  4 19:33:24 ubuntu snapd[896]: overlord.go:276: Acquired state lock file
Nov  4 19:33:24 ubuntu snapd[896]: daemon.go:247: started snapd/2.63+22.04ubuntu0.1 (series 16; clas
sic) ubuntu/22.04 (amd64) linux/5.15.0-160-generic.
Nov  4 19:33:24 ubuntu kernel: [   53.965361] loop6: detected capacity change from 0 to 8
Nov  4 19:33:24 ubuntu snapd[896]: daemon.go:340: adjusting startup timeout by 50s (pessimistic esti
mate of 30s plus 5s per snap)
Nov  4 19:33:25 ubuntu systemd[1]: tmp-sysecheck\x2dmountpoint\x2d2505362919.mount: Deactivated succe
ssfully.
Nov  4 19:33:25 ubuntu snapd[896]: backends.go:58: AppArmor status: apparmor is enabled and all feat
ures are available
Nov  4 19:33:25 ubuntu xrdp[969]: [INFO ] starting xrdp with pid 969
Nov  4 19:33:25 ubuntu xrdp[969]: [INFO ] address [0.0.0.0] port [3389] mode 1
Nov  4 19:33:25 ubuntu xrdp[969]: [INFO ] listening to port 3389 on 0.0.0.0
Nov  4 19:33:25 ubuntu xrdp[969]: [INFO ] xrdp_listen_pp done
Nov  4 19:33:25 ubuntu lxd.activate[894]: => Starting LXD activation
Nov  4 19:33:25 ubuntu lxd.activate[894]: ==> Loading snap configuration
Nov  4 19:33:25 ubuntu lxd.activate[894]: ==> Checking for socket activation support
Nov  4 19:33:25 ubuntu kernel: [   54.771820] kauditd_printk_skb: 34 callbacks suppressed
Nov  4 19:33:25 ubuntu kernel: [   54.771825] audit: type=1400 audit(1762284805.720:46): apparmor="S
TATUS" operation="profile_replace" profile="unconfined" name="/usr/lib/snapd/snap-confine" pid=1070
comm="apparmor_parser"
Nov  4 19:33:25 ubuntu kernel: [   54.808540] audit: type=1400 audit(1762284805.768:47): apparmor="S
:
```

more: sudo cat /var/log/syslog | more

 # quit q #spacebar to move

```
espace-capture-helper" pid=1070 comm="apparmor_parser"
Nov  4 19:33:25 ubuntu systemd[1]: Started Snap Daemon.
Nov  4 19:33:25 ubuntu dbus-daemon[883]: [system] Activating via systemd: service name='org.freedesk
top.timedate1' unit='dbus-org.freedesktop.timedate1.service' requested by ':1.13' (uid=0 pid=896 com
m="/usr/lib/snapd/snapd " label="unconfined")
Nov  4 19:33:25 ubuntu systemd[1]: Starting Time & Date Service...
Nov  4 19:33:26 ubuntu dbus-daemon[883]: [system] Successfully activated service 'org.freedesktop.ti
medate1'
Nov  4 19:33:26 ubuntu systemd[1]: Started Time & Date Service.
Nov  4 19:33:26 ubuntu systemd[1]: Finished Wait until snapd is fully seeded.
Nov  4 19:33:26 ubuntu systemd[1]: Starting Cloud-init: Config Stage...
Nov  4 19:33:26 ubuntu systemd[1]: Condition check resulted in Auto import assertions from block dev
ices being skipped.
Nov  4 19:33:26 ubuntu lxd.activate[894]: ==> Setting LXD socket ownership
Nov  4 19:33:26 ubuntu lxd.activate[894]: ==> Setting LXD user socket ownership
Nov  4 19:33:26 ubuntu lxd.activate[894]: ==> LXD never started on this system, no need to start it
now
Nov  4 19:33:26 ubuntu systemd[1]: snap.lxd.activate.service: Deactivated successfully.
Nov  4 19:33:26 ubuntu systemd[1]: Finished Service for snap application lxd.activate.
Nov  4 19:33:26 ubuntu systemd[1]: snap.lxd.activate.service: Consumed 1.531s CPU time.
Nov  4 19:33:26 ubuntu systemd[1]: Reached target Multi-User System.
Nov  4 19:33:26 ubuntu systemd[1]: Reached target Graphical Interface.
Nov  4 19:33:26 ubuntu systemd[1]: Starting Record Runlevel Change in UTMP...
Nov  4 19:33:26 ubuntu systemd[1]: systemd-update-utmp-runlevel.service: Deactivated successfully.
Nov  4 19:33:26 ubuntu systemd[1]: Finished Record Runlevel Change in UTMP.
Nov  4 19:33:27 ubuntu ModemManager[945]: <info>  [base-manager] couldn't check support for device '
/sys/devices/pci0000:00/0000:00:11.0/0000:02:01.0': not supported by any plugin
Nov  4 19:33:27 ubuntu systemd[1]: dmesg.service: Deactivated successfully.
Nov  4 19:33:27 ubuntu cloud-init[1128]: Cloud-init v. 25.1.4-0ubuntu0~22.04.1 running 'modules:conf
ig' at Tue, 04 Nov 2025 19:33:27 +0000. Up 55.17 seconds.
Nov  4 19:33:27 ubuntu systemd[1]: Finished Cloud-init: Config Stage.
Nov  4 19:33:27 ubuntu systemd[1]: Starting Cloud-init: Final Stage...
Nov  4 19:33:28 ubuntu cloud-init[1138]: Cloud-init v. 25.1.4-0ubuntu0~22.04.1 running 'modules:fina
l' at Tue, 04 Nov 2025 19:33:28 +0000. Up 56.37 seconds.
Nov  4 19:33:29 ubuntu cloud-init[1138]: Cloud-init v. 25.1.4-0ubuntu0~22.04.1 finished at Tue, 04 N
ov 2025 19:33:29 +0000. Datasource DataSourceNone.  Up 56.67 seconds
--More--
```

grep failures/errors: sudo grep -E 'fail|error' /var/log/syslog | head

 # E enables extended regex **# head shows only the first 10 result**

```
Nov  4 19:33:26 ubuntu lxd.activate[894]: ==> Setting LXD socket ownership
Nov  4 19:33:26 ubuntu lxd.activate[894]: ==> Setting LXD user socket ownership
Nov  4 19:33:26 ubuntu lxd.activate[894]: ==> LXD never started on this system, no need to start it
now
Nov  4 19:33:26 ubuntu systemd[1]: snap.lxd.activate.service: Deactivated successfully.
Nov  4 19:33:26 ubuntu systemd[1]: Finished Service for snap application lxd.activate.
Nov  4 19:33:26 ubuntu systemd[1]: snap.lxd.activate.service: Consumed 1.531s CPU time.
Nov  4 19:33:26 ubuntu systemd[1]: Reached target Multi-User System.
Nov  4 19:33:26 ubuntu systemd[1]: Reached target Graphical Interface.
Nov  4 19:33:26 ubuntu systemd[1]: Starting Record Runlevel Change in UTMP...
Nov  4 19:33:26 ubuntu systemd[1]: systemd-update-utmp-runlevel.service: Deactivated successfully.
Nov  4 19:33:26 ubuntu systemd[1]: Finished Record Runlevel Change in UTMP.
Nov  4 19:33:27 ubuntu ModemManager[945]: <info>  [base-manager] couldn't check support for device '
/sys/devices/pci0000:00/0000:00:11.0/0000:02:01.0': not supported by any plugin
Nov  4 19:33:27 ubuntu systemd[1]: dmesg.service: Deactivated successfully.
Nov  4 19:33:27 ubuntu cloud-init[1128]: Cloud-init v. 25.1.4-0ubuntu0~22.04.1 running 'modules:conf
ig' at Tue, 04 Nov 2025 19:33:27 +0000. Up 55.17 seconds.
Nov  4 19:33:27 ubuntu systemd[1]: Finished Cloud-init: Config Stage.
Nov  4 19:33:27 ubuntu systemd[1]: Starting Cloud-init: Final Stage...
Nov  4 19:33:28 ubuntu cloud-init[1138]: Cloud-init v. 25.1.4-0ubuntu0~22.04.1 running 'modules:fina
l' at Tue, 04 Nov 2025 19:33:28 +0000. Up 56.37 seconds.
Nov  4 19:33:29 ubuntu cloud-init[1138]: Cloud-init v. 25.1.4-0ubuntu0~22.04.1 finished at Tue, 04 N
ov 2025 19:33:29 +0000. Datasource DataSourceNone.  Up 56.67 seconds
sarosh@ubuntu:~$ sudo grep -E 'fail|error' /var/log/syslog | head

Nov  4 19:33:24 ubuntu multipath: sda: failed to get sysfs uid: Invalid argument
Nov  4 19:33:24 ubuntu multipath: sda: failed to get sgio uid: No such file or directory
Nov  4 19:33:24 ubuntu multipathd[562]: sda: failed to get udev uid: No data available
Nov  4 19:33:24 ubuntu multipathd[562]: sda: failed to get path uid
Nov  4 19:33:24 ubuntu multipathd[562]: uevent trigger error
Nov  4 19:33:34 ubuntu multipath: sda: failed to get sysfs uid: Invalid argument
Nov  4 19:33:34 ubuntu multipath: sda: failed to get sgio uid: No such file or directory
Nov  4 19:33:34 ubuntu multipathd[562]: sda: failed to get udev uid: No data available
Nov  4 19:33:34 ubuntu multipathd[562]: sda: failed to get path uid
Nov  4 19:33:34 ubuntu multipathd[562]: uevent trigger error
sarosh@ubuntu:~$
sarosh@ubuntu:~$ _
```

redirect: sudo grep -i systemd /var/log/syslog > ~/syslog_systemd.txt

 # > creates or overwrites the file.

```
Nov  5 21:25:45 ubuntu systemd-timesyncd[731]: Network configuration changed, trying to establish co
nnection.
Nov  5 21:25:56 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.57:123 (
ntp.ubuntu.com).
Nov  5 21:26:06 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.58:123 (
ntp.ubuntu.com).
Nov  5 21:26:16 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 91.189.91.157:123 (n
tp.ubuntu.com).
Nov  5 21:26:26 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.56:123 (
ntp.ubuntu.com).
Nov  5 21:40:45 ubuntu systemd-timesyncd[731]: Network configuration changed, trying to establish co
nnection.
Nov  5 21:40:55 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 91.189.91.157:123 (n
tp.ubuntu.com).
Nov  5 21:41:05 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.57:123 (
ntp.ubuntu.com).
Nov  5 21:41:15 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.56:123 (
ntp.ubuntu.com).
Nov  5 21:41:26 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.58:123 (
ntp.ubuntu.com).
Nov  5 21:54:37 ubuntu systemd[1]: Started Getty on tty6.
Nov  5 21:55:45 ubuntu systemd-timesyncd[731]: Network configuration changed, trying to establish co
nnection.
Nov  5 21:55:55 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.58:123 (
ntp.ubuntu.com).
Nov  5 21:56:05 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 91.189.91.157:123 (n
tp.ubuntu.com).
Nov  5 21:56:16 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.56:123 (
ntp.ubuntu.com).
Nov  5 21:56:16 ubuntu systemd[1]: Starting Daily apt upgrade and clean activities...
Nov  5 21:56:23 ubuntu systemd[1]: apt-daily-upgrade.service: Deactivated successfully.
Nov  5 21:56:23 ubuntu systemd[1]: Finished Daily apt upgrade and clean activities.
Nov  5 21:56:23 ubuntu systemd[1]: apt-daily-upgrade.service: Consumed 7.043s CPU time.
Nov  5 21:56:26 ubuntu systemd-timesyncd[731]: Timed out waiting for reply from 185.125.190.57:123 (
ntp.ubuntu.com).
sarosh@ubuntu:~$
sarosh@ubuntu:~$ _
```

append: **sudo grep -i network /var/log/syslog >> ~/syslog_systemd.txt**

cat ~/syslog_systemd.txt

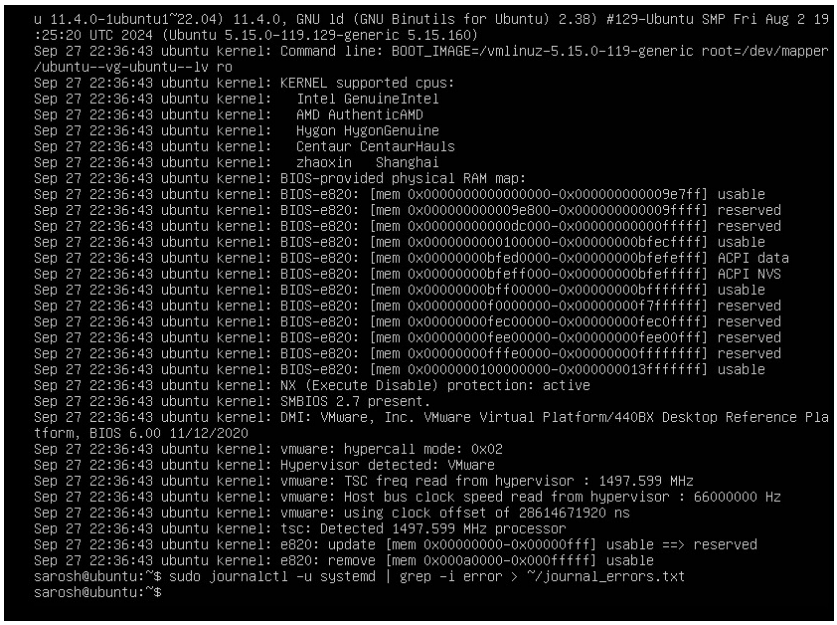
 # >> appends (adds at the end without deleting old

```
Nov  5 21:10:48 ubuntu systemd-networkd[878]: ens33: DHCPv4 address 192.168.138.131/24 via 192.168.1
38.2
Nov  5 21:10:48 ubuntu systemd-timesyncd[731]: Network configuration changed, trying to establish co
nnection.
Nov  5 21:10:48 ubuntu systemd-networkd[878]: ens33: Gained IPv6LL
Nov  5 21:10:48 ubuntu systemd-timesyncd[731]: Network configuration changed, trying to establish co
nnection.
Nov  5 21:10:48 ubuntu systemd[1]: Finished Wait for Network to be Configured.
Nov  5 21:10:48 ubuntu systemd[1]: Starting Cloud-init: Network Stage...
Nov  5 21:10:48 ubuntu systemd[1]: Finished Cloud-init: Network Stage.
Nov  5 21:10:48 ubuntu systemd[1]: Reached target Network is Online.
Nov  5 21:10:48 ubuntu systemd[1]: Starting Dispatcher daemon for systemd-networkd...
Nov  5 21:10:48 ubuntu avahi-daemon[892]: Network interface enumeration completed.
Nov  5 21:10:48 ubuntu kernel: [   22.579321] drop_monitor: Initializing network drop monitor servic
e
Nov  5 21:10:48 ubuntu kernel: [   30.574415] e1000: Intel(R) PRO/1000 Network Driver
Nov  5 21:10:48 ubuntu kernel: [   31.718528] e1000 0000:02:01.0 eth0: Intel(R) PRO/1000 Network Con
nection
Nov  5 21:10:48 ubuntu kernel: [   37.885799] systemd[1]: Listening on Network Service Netlink Socke
t.
Nov  5 21:10:48 ubuntu kernel: [   44.580502] audit: type=1400 audit(1762377040.528:5): apparmor="ST
ATUS" operation="profile_load" profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-client.act
ion" pid=738 comm="apparmor_parser"
Nov  5 21:10:48 ubuntu kernel: [   44.580525] audit: type=1400 audit(1762377040.528:6): apparmor="ST
ATUS" operation="profile_load" profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-helper" pi
d=738 comm="apparmor_parser"
Nov  5 21:10:49 ubuntu networkd-dispatcher[904]: No valid path found for iwconfig
Nov  5 21:10:49 ubuntu networkd-dispatcher[904]: No valid path found for iw
Nov  5 21:10:50 ubuntu systemd[1]: Started Dispatcher daemon for systemd-networkd.
Nov  5 21:11:43 ubuntu systemd[1302]: Listening on GnuPG network certificate management daemon.
Nov  5 21:25:45 ubuntu systemd-timesyncd[731]: Network configuration changed, trying to establish co
nnection.
Nov  5 21:40:45 ubuntu systemd-timesyncd[731]: Network configuration changed, trying to establish co
nnection.
Nov  5 21:55:45 ubuntu systemd-timesyncd[731]: Network configuration changed, trying to establish co
nnection.
sarosh@ubuntu:~$
```

Alternative (journalctl) if needed:

sudo journalctl | less

sudo journalctl -u systemd | grep -i error > ~/journal_errors.txt



Task 10 – Script setup.sh – variables, command substitution, file/dir checks, permissions (use vim)

Goal: Using vim, write a script named setup.sh that implements each numbered step below. After writing the code for each step, run the script and capture screenshots showing the vim editor (script content) and the script output for that step. Students must add the code for each step into the same file setup.sh step-by-step (i.e., write 1., save, run and screenshot; then append 2., save, run and screenshot; and so on).

For each step you MUST:

Open vim and edit setup.sh

Insert only the code shown for that step (append to the existing file)

Save and quit vim (:wq)

Make the file executable if not already: chmod +x setup.sh

Run the script: ./setup.sh

Capture two screenshots:

One showing the vim editor with the script content after you added the step (use the vim screen before :wq)

One showing the terminal output after running the script (show the command and the output)

Start in your Student home directory (recommended).

Include bash shebang

Code to add (enter in vim as the first line of the file):

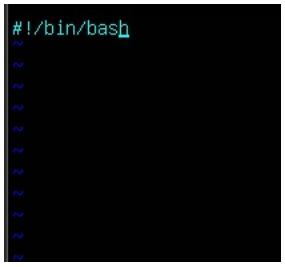
#!/bin/bash

Steps:

vim setup.sh → add the shebang line → save and quit

chmod +x setup.sh

./setup.sh



script run output (likely no output but show ./setup.sh run)

Define variable var1 and echo it

Code to append:

 # Define and show var1

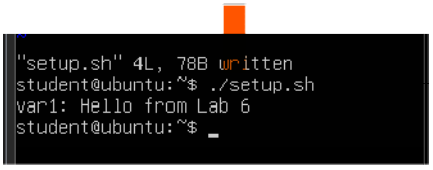
var1="Hello from Lab 6"

echo "var1: $var1"

Steps:

vim setup.sh → append the code above → save and quit

./setup.sh



Save output of ls -l into variable allFiles and echo it

Code to append:

 # Save ls -l to variable and display

allFiles="$(ls -l)"

echo "allFiles (ls -l):"

echo "$allFiles"

Steps:

vim setup.sh → append the code above → save and quit

./setup.sh

```
#!/bin/bash
# Define and show var1
var1="Hello from Lab 6"
echo "var1: $var1"
allFiles="$(ls -l)"
echo "allFiles (ls -l):"
echo "$allFiles"
```

```
"setup.sh" 8L, 141B written
student@ubuntu:~$ ./setup.sh
var1: Hello from Lab 6
allFiles (ls -l):
total 8
drwxrwxr-x 2 student student 4096 Nov  4 20:40 dir1
-rwxr-x--- 1 student student    0 Nov  4 20:40 file1
-rwxrwxr-x 1 student student  141 Nov  5 23:13 setup.sh
student@ubuntu:~$
```

If directory dir1 exists echo a message; else create it

Code to append:

 # Directory check

```
if [ -d "dir1" ]; then
  echo "Directory dir1 exists."
else
  echo "Directory dir1 does not exist. Creating..."
  mkdir -p "dir1"
  echo "Directory dir1 created."
fi
```

Steps:

vim setup.sh → append the code above → save and quit

./setup.sh script run output showing directory message or creation

```
#!/bin/bash
# Define and show var1
var1="Hello from Lab 6"
echo "var1: $var1"
allFiles="$(ls -l)"
echo "allFiles (ls -l):"
echo "$allFiles"

# Directory check

if [ -d "dir1" ]; then
          echo "Directory dir1 exists."
  else
            echo "Directory dir1 does not exist. Creating..."
              mkdir -p "dir1"
                echo "Directory dir1 created."
fi
```

```
"setup.sh" 25L, 352B written
student@ubuntu:~$ ./setup.sh
var1: Hello from Lab 6
allFiles (ls -l):
total 8
drwxrwxr-x 2 student student 4096 Nov  4 20:40 dir1
-rwxr-x--- 1 student student    0 Nov  4 20:40 file1
-rwxrwxr-x 1 student student  352 Nov  5 23:16 setup.sh
Directory dir1 exists.
student@ubuntu:~$
```

If file dir1/file2 does not exist, create it

Code to append:

 # File check

```
if [ -f "dir1/file2" ]; then
  echo "file2 already exists."
else
  echo "file2 does not exist. Creating..."
  touch "dir1/file2"
  chmod a-rwx "dir1/file2"
  echo "file2 created."
fi
```

Steps:

vim setup.sh → append the code above → save and quit

./setup.sh script run output showing file creation message or existence

```
# Directory check
if [ -d "dir1" ]; then
        echo "Directory dir1 exists."
  else
          echo "Directory dir1 does not exist. Creating..."
            mkdir -p "dir1"
              echo "Directory dir1 created."
fi

# File check
if [ -f "dir1/file2" ]; then
        echo "file2 already exists."
  else
          echo "file2 does not exist. Creating..."
            touch "dir1/file2"
              chmod a-rwx "dir1/file2"
                echo "file2 created."
fi
```

```
"setup.sh" 44L, 578B written
student@ubuntu:~$ ./setup.sh
var1: Hello from Lab 6
allFiles (ls -l):
total 8
drwxrwxr-x 2 student student 4096 Nov  4 20:40 dir1
-rwxr-x--- 1 student student    0 Nov  4 20:40 file1
-rwxrwxr-x 1 student student  578 Nov  5 23:18 setup.sh
Directory dir1 exists.
file2 already exists.
student@ubuntu:~$
```

Check read, write, execute permissions on dir1/file2; grant missing user perms and show final ls

Code to append:

```
# Permission checks for dir1/file2 (user permissions)
f="dir1/file2"
if [ ! -r "$f" ]; then
  echo "Read permission missing; granting to user..."
  chmod u+r "$f"
fi

if [ ! -w "$f" ]; then
  echo "Write permission missing; granting to user..."
  chmod u+w "$f"
fi

if [ ! -x "$f" ]; then
  echo "Execute permission missing; granting to user..."
  chmod u+x "$f"
fi

echo "Final permissions for $f:"
ls -l "$f"
```

Steps:

vim setup.sh → append the code above → save and quit

./setup.sh **script run output showing the permission grants and final ls -l dir1/file2**

```
if [ ! -r "$f" ]; then
          echo "Read permission missing; granting to user..."
            chmod u+r "$f"
fi


if [ ! -w "$f" ]; then
          echo "Write permission missing; granting to user..."
            chmod u+w "$f"
fi


if [ ! -x "$f" ]; then
          echo "Execute permission missing; granting to user..."
            chmod u+x "$f"
fi


echo "Final permissions for $f:"
ls -l "$f"
```

```
"setup.sh" 83L, 1021B written
student@ubuntu:~$ ./setup.sh
var1: Hello from Lab 6
allFiles (ls -l):
total 8
drwxrwxr-x 2 student student 4096 Nov  4 20:40 dir1
-rwxr-x--- 1 student student    0 Nov  4 20:40 file1
-rwxrwxr-x 1 student student 1021 Nov  5 23:19 setup.sh
Directory dir1 exists.
file2 already exists.
Execute permission missing; granting to user...
Final permissions for dir1/file2:
-rwxrw-r-- 1 student student 0 Nov  4 20:40 dir1/file2
student@ubuntu:~$
```

Important notes for Task 10:

Students MUST add the code incrementally exactly as described above (do not replace the file each time — append).

Capture the vim editor screen (showing your code in the buffer) before you save and quit for each step (these are the task10_b*_vim.png files).

After saving, run the script and capture the terminal output for that run (these are the task10_b*_run.png files).

If a step reports that a directory/file already exists, that is acceptable — still capture the output screenshot showing the script's message.

Task 11 – Script setup.sh – argument comparisons (eq, ne, gt, lt, ge, le) and string checks

Updated: replace the previous single-script approach with an incremental exercise. Students will overwrite setup.sh and then add each individual if-test one-by-one. After adding each if-test they must run the script with example arguments and capture screenshots. This teaches the individual comparison operators and makes each if statement a separate step.

Important overall instructions

Start by overwriting setup.sh (vim setup.sh) and add only what the step asks (do not add all tests at once).

After editing in vim, save (:wq), make executable (chmod +x setup.sh) if needed, then run the script with the example commands shown for each step.

For each step capture two screenshots:

A vim screenshot showing the current file buffer with the newly added lines (before :wq) — name as specified for the step.

A terminal screenshot showing the commands you ran (chmod +x setup.sh if necessary) and the script outputs for the example invocations — name as specified for the step.

For the numeric comparisons, set a variable num=$1 at the top of the file before adding the individual if-tests (this will be the initial step). For string checks, set str=$2 before adding the string if-tests.

create file with shebang and set num and str variables

In vim create/overwrite setup.sh and insert:

#!/bin/bash

num=$1

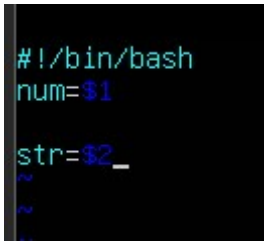
str=$2

Save and quit (:wq)

Make executable and run with examples:

chmod +x setup.sh

./setup.sh 10 Student



add the -eq test (equal)

Append to setup.sh:

```
if [ "$num" -eq 10 ]; then
  echo "$num is equal to 10 (-eq)."
else
  echo "$num is NOT equal to 10 (-eq)."
fi
```

Save and quit; then run these commands (capture both in one terminal screenshot):

./setup.sh 10 Student

./setup.sh 7 Student

```
#!/bin/bash
num=$1

str=$2
if [ "$num" -eq 10 ]; then

        echo "$num is equal to 10 (-eq)."

  else

          echo "$num is NOT equal to 10 (-eq)."

fi
```

```
"setup.sh" 15L, 150B written
student@ubuntu:~$ ./setup.sh 10 Student
10 is equal to 10 (-eq).
student@ubuntu:~$ ./setup.sh 7 Student
7 is NOT equal to 10 (-eq).
student@ubuntu:~$
```

add the -ne test (not equal)

Append to setup.sh:

if ["$num" -ne 10]; then

echo "$num is not equal to 10 (-ne)."

else

echo "$num is equal to 10 (-ne false)."

fi

Save and quit; run:

./setup.sh 7 Student

./setup.sh 10 Student

```
#!/bin/bash
num=$1

str=$2
if [ "$num" -eq 10 ]; then

          echo "$num is equal to 10 (-eq)."

  else

            echo "$num is NOT equal to 10 (-eq)."

fi

if [ "$num" -ne 10 ]; then

  echo "$num is not equal to 10 (-ne)."

else

          echo "$num is equal to 10 (-ne false)."

          fi
```

```
"setup.sh" 25L, 279B written
student@ubuntu:~$ ./setup.sh 7 Student
7 is NOT equal to 10 (-eq).
7 is not equal to 10 (-ne).
student@ubuntu:~$ ./setup.sh 10 Student
10 is equal to 10 (-eq).
10 is equal to 10 (-ne false).
student@ubuntu:~$
```

add the -gt test (greater than)

Append:

```
if [ "$num" -gt 10 ]; then
  echo "$num is greater than 10 (-gt)."
else
  echo "$num is NOT greater than 10 (-gt)."
fi
```

Run:

./setup.sh 12 Student

./setup.sh 9 Student

```
#!/bin/bash
num=$1

str=$2
if [ "$num" -eq 10 ]; then
          echo "$num is equal to 10 (-eq)."
  else
            echo "$num is NOT equal to 10 (-eq)."
fi
if [ "$num" -gt 10 ]; then
  echo "$num is greater than 10 (-gt)."
  else
            echo "$num is NOT greater than 10 (-gt)."
            fi
            if [ "$num" -ne 10 ]; then
  echo "$num is not equal to 10 (-ne)."
else
          echo "$num is equal to 10 (-ne false)."
          fi
```

```
"setup.sh" 35L, 433B written
student@ubuntu:~$ ./setup.sh 12 Student
12 is NOT equal to 10 (-eq).
12 is greater than 10 (-gt).
12 is not equal to 10 (-ne).
student@ubuntu:~$ ./setup.sh 9 Student
9 is NOT equal to 10 (-eq).
9 is NOT greater than 10 (-gt).
9 is not equal to 10 (-ne).
student@ubuntu:~$
```

add the -lt test (less than)

Append:

```
if [ "$num" -lt 10 ]; then
  echo "$num is less than 10 (-lt)."
```

else

echo "$num is NOT less than 10 (-lt)."

fi

Run:

./setup.sh 5 Student

./setup.sh 11 Student

```
          echo "$num is NOT equal to 10 (-eq)."
fi
if [ "$num" -gt 10 ]; then
  echo "$num is greater than 10 (-gt)."
  else
          echo "$num is NOT greater than 10 (-gt)."
          fi
          if [ "$num" -ne 10 ]; then
  echo "$num is not equal to 10 (-ne)."
else
         echo "$num is equal to 10 (-ne false)."
         fi
if [ "$num" -lt 10 ]; then
  echo "$num is less than 10 (-lt)."
else
  echo "$num is NOT less than 10 (-lt)."
fi
```

```
"setup.sh" 46L, 549B written
student@ubuntu:~$ ./setup.sh 5 Student
5 is NOT equal to 10 (-eq).
5 is NOT greater than 10 (-gt).
5 is not equal to 10 (-ne).
5 is less than 10 (-lt).
student@ubuntu:~$ ./setup.sh 11 Student
11 is NOT equal to 10 (-eq).
11 is greater than 10 (-gt).
11 is not equal to 10 (-ne).
11 is NOT less than 10 (-lt).
student@ubuntu:~$
```

add the -ge test (greater than or equal)

Append:

if ["$num" -ge 10]; then

echo "$num is greater than or equal to 10 (-ge)."

else

echo "$num is NOT greater than or equal to 10 (-ge)."

fi

Run:

./setup.sh 10 Student

./setup.sh 8 Student

```
          fi
          if [ "$num" -ne 10 ]; then
  echo "$num is not equal to 10 (-ne)."
else
         echo "$num is equal to 10 (-ne false)."
         fi
if [ "$num" -lt 10 ]; then
  echo "$num is less than 10 (-lt)."
else
  echo "$num is NOT less than 10 (-lt)."
fi
if [ "$num" -ge 10 ]; then
  echo "$num is greater than or equal to 10 (-ge)."
else
         echo "$num is NOT greater than or equal to 10 (-ge)."
         fi
```

```
"setup.sh" 56L, 701B written
student@ubuntu:~$ ./setup.sh 10 Student
10 is equal to 10 (-eq).
10 is NOT greater than 10 (-gt).
10 is equal to 10 (-ne false).
10 is NOT less than 10 (-lt).
10 is greater than or equal to 10 (-ge).
student@ubuntu:~$ ./setup.sh 8 Student
8 is NOT equal to 10 (-eq).
8 is NOT greater than 10 (-gt).
8 is not equal to 10 (-ne).
8 is less than 10 (-lt).
8 is NOT greater than or equal to 10 (-ge).
student@ubuntu:~$ _
```

add the -le test (less than or equal)

Append:

```
if [ "$num" -le 10 ]; then
  echo "$num is less than or equal to 10 (-le)."
else
  echo "$num is NOT less than or equal to 10 (-le)."
fi
```

Run:

./setup.sh 10 Student

./setup.sh 12 Student

```
if [ "$num" -le 10 ]; then
  echo "$num is less than or equal to 10 (-le)."
else
          echo "$num is NOT less than or equal to 10 (-le)."
          fi
_
```

```
"setup.sh" 66L, 847B written
student@ubuntu:~$ ./setup.sh 10 Student
10 is equal to 10 (-eq).
10 is NOT greater than 10 (-gt).
10 is equal to 10 (-ne false).
10 is NOT less than 10 (-lt).
10 is greater than or equal to 10 (-ge).
10 is less than or equal to 10 (-le).
student@ubuntu:~$ ./setup.sh 12 Student
12 is NOT equal to 10 (-eq).
12 is greater than 10 (-gt).
12 is not equal to 10 (-ne).
12 is NOT less than 10 (-lt).
12 is greater than or equal to 10 (-ge).
12 is NOT less than or equal to 10 (-le).
student@ubuntu:~$ _
```

string equality test (=)

Ensure str=$2 exists at top (1.). Append:

```
if [ "$str" = "Student" ]; then
  echo "Second argument equals 'Student' ( = )."
else
  echo "Second argument does NOT equal 'Student' ( = )."
fi
```

Run:

```
./setup.sh 10 Student
./setup.sh 10 Test
```

```
if [ "$str" = "Student" ]; then
  echo "Second argument equals 'Student' ( = )."
else
          echo "Second argument does NOT equal 'Student' ( = )."
          fi
```

```
"setup.sh" 76L, 1002B written
student@ubuntu:~$ ./setup.sh 10 Student
10 is equal to 10 (-eq).
10 is NOT greater than 10 (-gt).
10 is equal to 10 (-ne false).
10 is NOT less than 10 (-lt).
10 is greater than or equal to 10 (-ge).
10 is less than or equal to 10 (-le).
Second argument equals 'Student' ( = ).
student@ubuntu:~$ ./setup.sh 10 Test
10 is equal to 10 (-eq).
10 is NOT greater than 10 (-gt).
10 is equal to 10 (-ne false).
10 is NOT less than 10 (-lt).
10 is greater than or equal to 10 (-ge).
10 is less than or equal to 10 (-le).
Second argument does NOT equal 'Student' ( = ).
student@ubuntu:~$ _
```

string inequality test (!=)

Append:

```
if [ "$str" != "Student" ]; then
  echo "Second argument is not equal to 'Student' ( != )."
else
  echo "Second argument equals 'Student' ( != false)."
fi
```

Run:

```
./setup.sh 10 Test
./setup.sh 10 Student
```

```
if [ "$str" != "Student" ]; then
  echo "Second argument is not equal to 'Student' ( != )."
else
         echo "Second argument equals 'Student' ( != false)."
         fi
```

```
"setup.sh" 86L, 1166B written
student@ubuntu:~$ ./setup.sh 10 Test
10 is equal to 10 (-eq).
10 is NOT greater than 10 (-gt).
10 is equal to 10 (-ne false).
10 is NOT less than 10 (-lt).
10 is greater than or equal to 10 (-ge).
10 is less than or equal to 10 (-le).
Second argument does NOT equal 'Student' ( = ).
Second argument is not equal to 'Student' ( != ).
student@ubuntu:~$ ./setup.sh 10 Student
10 is equal to 10 (-eq).
10 is NOT greater than 10 (-gt).
10 is equal to 10 (-ne false).
10 is NOT less than 10 (-lt).
10 is greater than or equal to 10 (-ge).
10 is less than or equal to 10 (-le).
Second argument equals 'Student' ( = ).
Second argument equals 'Student' ( != false).
student@ubuntu:~$
```

check if second argument is empty (zero-length)

Append:

if [-z "$str"]; then

echo "Second argument is empty (zero-length)."

else

echo "Second argument is not empty."

fi

Run:

./setup.sh 10

./setup.sh 10 Student

```
if [ -z "$str" ]; then
  echo "Second argument is empty (zero-length)."
else
         echo "Second argument is not empty."
         fi
```

```
"setup.sh" 94L, 1292B written
student@ubuntu:~$ ./setup.sh 10
10 is equal to 10 (-eq).
10 is NOT greater than 10 (-gt).
10 is equal to 10 (-ne false).
10 is NOT less than 10 (-lt).
10 is greater than or equal to 10 (-ge).
10 is less than or equal to 10 (-le).
Second argument does NOT equal 'Student' ( = ).
Second argument is not equal to 'Student' ( != ).
Second argument is empty (zero-length).
student@ubuntu:~$ ./setup.sh 10 Student
10 is equal to 10 (-eq).
10 is NOT greater than 10 (-gt).
10 is equal to 10 (-ne false).
10 is NOT less than 10 (-lt).
10 is greater than or equal to 10 (-ge).
10 is less than or equal to 10 (-le).
Second argument equals 'Student' ( = ).
Second argument equals 'Student' ( != false).
Second argument is not empty.
student@ubuntu:~$
```

Important Note Task 11 :

Students MUST follow the incremental approach: add one test at a time and capture the vim buffer and run output screenshots for each step.

Using the two example runs per step in a single terminal screenshot helps demonstrate both true and false results for the operator being taught.

If a non-numeric value is used for $1 during integer comparisons the shell may print an error; that is expected here because we are demonstrating the operator behavior step-by-step. (If you want to avoid runtime errors, you can add integer validation before the comparisons — but for this exercise we are isolating each if-test into its own step.)

Task 12 – Script setup.sh – print all arguments with a for loop

Create the script with shebang and basic structure

Open vim and overwrite setup.sh:

vim setup.sh

Insert these lines (first step — shebang and a short comment):

#!/bin/bash

 # Script to demonstrate printing all user-entered arguments using $*

Save and quit (:wq)

```
#!/bin/bash
# Script to demonstrate printing all user-entered arguments using $*
```

```
student@ubuntu:~$ chmod +x setup.sh
student@ubuntu:~$
student@ubuntu:~$ ./setup.sh
```

Append the for loop using $* and print each argument

Re-open setup.sh in vim and append the following lines:

 # Print all arguments using $*

echo "Printing all arguments using \$*:"

for arg in $*; do

** echo "Argument: $arg"**

done

Save and quit (:wq)

Make the script executable and run it with example arguments:

chmod +x setup.sh

./setup.sh one "two words" three

```
#!/bin/bash
# Script to demonstrate printing all user-entered arguments using $*

# Print all arguments using $*

echo "Printing all arguments using \$*:"

for arg in $*; do

          echo "Argument: $arg"

  done
```

```
"setup.sh" 22L, 221B written
student@ubuntu:~$ ./setup.sh one "two words" three

Printing all arguments using $*:
Argument: one
Argument: two
Argument: words
Argument: three
student@ubuntu:~$
student@ubuntu:~$ _
```

Task 13 – Script setup.sh – while loop summation and functions

Clear the previous code of setup.sh and write a new script, step-by-step, that:

Starts with a shebang line

Implements an interactive while loop that prompts the user to enter numbers and keeps a running total until the user types q to quit; after each input the script echoes "Total Score: <current_total>"

Implements a function sum_two() that runs its own interactive while loop doing the same accumulation and echoes the running totals

Adds a second function that takes two numeric arguments, sums them, and returns the result via echo (demonstrated in the script)

Important: if you move the while-loop logic into the sum_two() function, delete the standalone while-loop code to avoid running the same loop twice

Add the shebang line

Open vim and overwrite setup.sh with the shebang line:

#!/bin/bash

Save and quit (:wq)

Make executable and run (no output expected):

chmod +x setup.sh

./setup.sh

```
#!/bin/bash
```

```
"setup.sh" 10L, 24B written
student@ubuntu:~$ chmod +x setup.sh
student@ubuntu:~$ ./setup.sh
student@ubuntu:~$ _
```

Add the while-loop summation (interactive)

Re-open setup.sh in vim and append the while-loop:

While-loop summation (interactive)

```
sum=0
while true; do
 read -p "Enter a number (or 'q' to quit): " input
 if [ "$input" = "q" ]; then
  break
 fi

 sum=$((sum + input))
 echo "Total Score: $sum"
done
echo "Final total: $sum"
```

Save and quit (:wq)

Run the script and demonstrate a short session (example): enter 5, then 7, then q

./setup.sh

 # interactively enter:

 # 5

 # 7

 # q

```
#!/bin/bash

le-loop summation (interactive)
sum=0
while true; do
         read -p "Enter a number (or 'q' to quit): " input
           if [ "$input" = "q" ]; then
                      break
                        fi


                          sum=$((sum + input))
                            echo "Total Score: $sum"
                    done
                    echo "Final total: $sum"
```

```
"setup.sh" 34L, 313B written
student@ubuntu:~$ ./setup.sh
Enter a number (or 'q' to quit): 5
Total Score: 5
Enter a number (or 'q' to quit): 7
Total Score: 12
Enter a number (or 'q' to quit): q
Final total: 12
student@ubuntu:~$ _
```

Add the interactive summation function and demonstrate it

Re-open setup.sh in vim and append the function sum_two() which contains its own interactive while-loop:

 # Function to accumulate scores interactively

```
sum_two() {
 sum=0
 while true; do
   read -p "Enter a number (or 'q' to quit): " input
   if [ "$input" = "q" ]; then
     break
   fi


   sum=$((sum + input))
```

```
    echo "Total Score: $sum"
  done
  echo "Function final total: $sum"
}
```

 # Demonstrate the function

```
echo "Now calling sum_two function:"
sum_two
```

Save and quit (:wq)

Important: If you have the standalone while-loop from step 2 and you place this function into the script, delete the standalone loop to avoid executing the same interactive logic twice when running the script.

Run the script and demonstrate a short session (example): enter 3, 4, q when prompted by the function:

```
./setup.sh
```

 # when prompted by the function enter:

 # 3

 # 4

 # q

```
#!/bin/bash

# Function to accumulate scores interactively
sum_two() {
        sum=0
          while true; do
                read -p "Enter a number (or 'q' to quit): " input
                    if [ "$input" = "q" ]; then
                                break
                                    fi


                                        sum=$((sum + input))
                                            echo "Total Score: $sum"
                                              done
                                                echo "Function final total: $sum"
                                        }


                                        # Demonstrate the function
                                        echo "Now calling sum_two function:"
```

```
"setup.sh" 46L, 501B written
student@ubuntu:~$ ./setup.sh
Now calling sum_two function:
Enter a number (or 'q' to quit): 3
Total Score: 3
Enter a number (or 'q' to quit): 4
Total Score: 7
Enter a number (or 'q' to quit): q
Function final total: 7
student@ubuntu:~$
```

Add a function that takes two numeric arguments, sums them, and returns the result (echo)

Re-open setup.sh in vim and append the following function and demonstration. This function accepts two numeric arguments, adds them, and return the sum. The script then captures that output and displays it.

 # Function that sums two arguments and returns the result

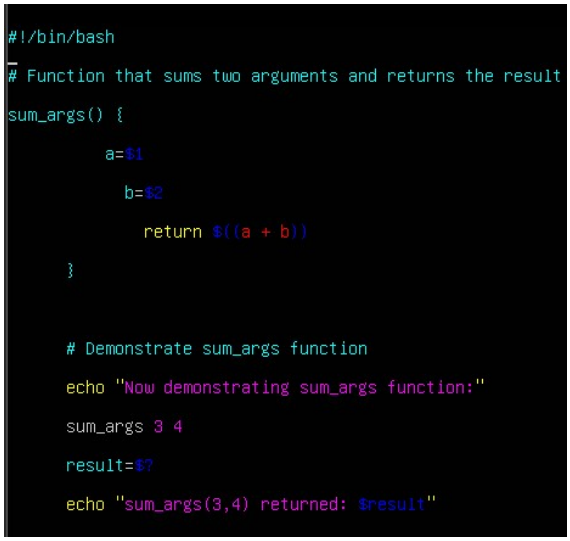
```
sum_args() {
  a=$1
  b=$2
  return $((a + b))
}
# Demonstrate sum_args function
echo "Now demonstrating sum_args function:"
sum_args 3 4
result=$?
echo "sum_args(3,4) returned: $result"
```

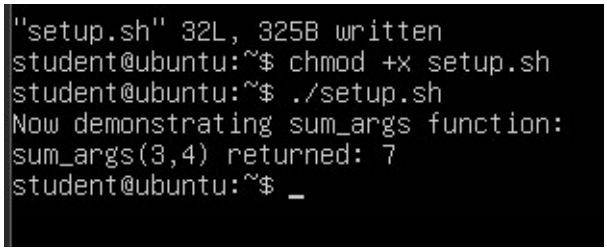
Save and quit (:wq)

Run the script and capture the demonstration output:

```
chmod +x setup.sh
./setup.sh
```

 # Observe the output that shows "sum_args(3,4) returned: 7"





Notes for Task 13:

Overwrite previous contents of setup.sh at the start of Task 13 (step 1).

Add code incrementally, save, run, and capture both the vim buffer and the run output screenshots for each numbered step.

The while-loop and the functions are interactive; include the user inputs in the run screenshots to demonstrate the behavior.

If you decide to use the function-based approach for interactive summation, remove the earlier standalone while-loop to avoid duplicate interaction.

Task 14 – Codespaces GUI — fork repo, run start-desktop.sh, open VNC, stop GUI

Goal: Fork the specified repository to your GitHub account, open it in GitHub Codespaces, run the provided script to start a desktop GUI, connect to the GUI via the Codespaces forwarded port (6080) -> vnc.html, and then stop the GUI using the provided stop script.

Important notes before starting:

GitHub Codespaces must be enabled for your account/org. Codespaces availability and billing may apply.

The instructions below assume you have permission and capacity to create a Codespace for your fork.

If Codespaces is not available, you may perform this step on another cloud environment that exposes the same port and scripts, but the screenshot filenames below assume Codespaces.

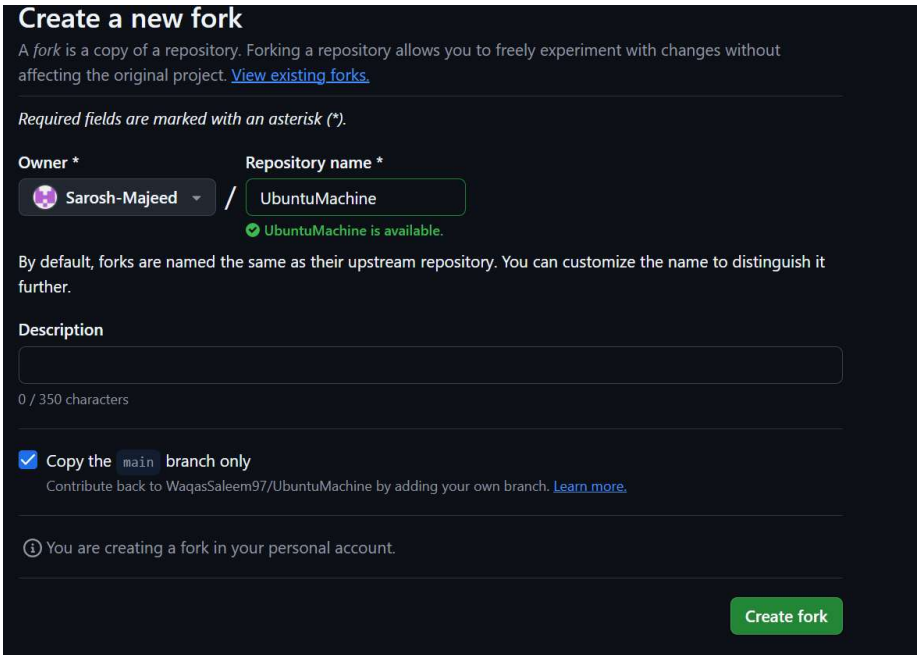
Steps:

Fork the repository to your GitHub account

Open the repo URL in your browser:

Ubuntu Machine

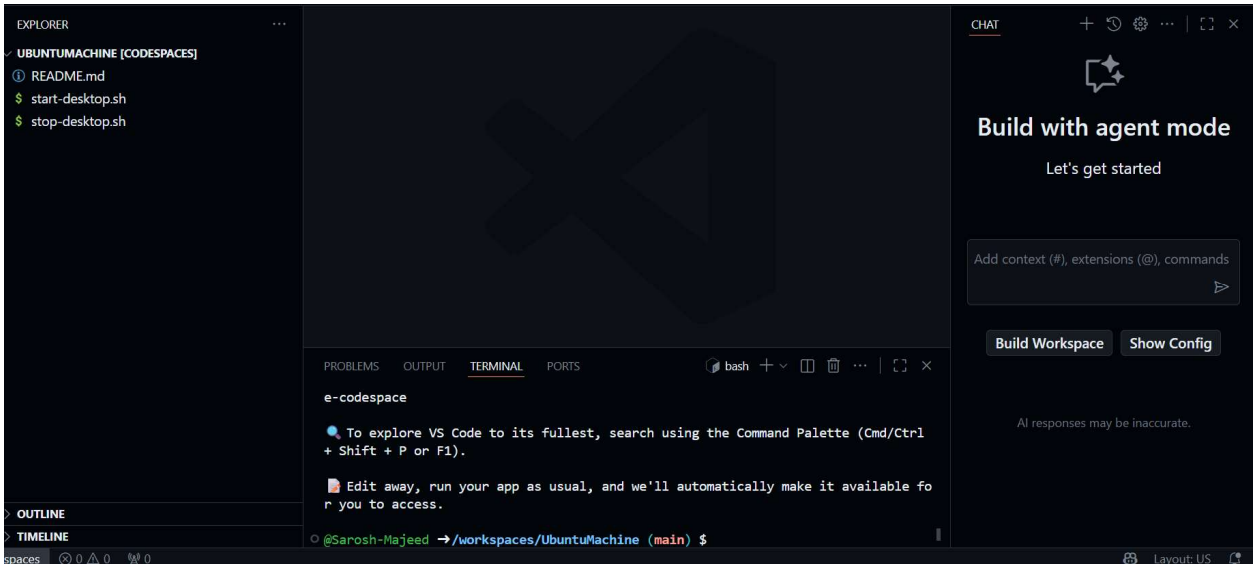
Click "Fork" (top-right) and fork it to your account.



Open a Codespace on your fork

In your forked repository on GitHub, click the green "Code" button → "Open with Codespaces" → "Create codespace on main" (or appropriate branch).

Wait for the Codespace to initialize.



Verify the start script is present and executable (capture evidence)

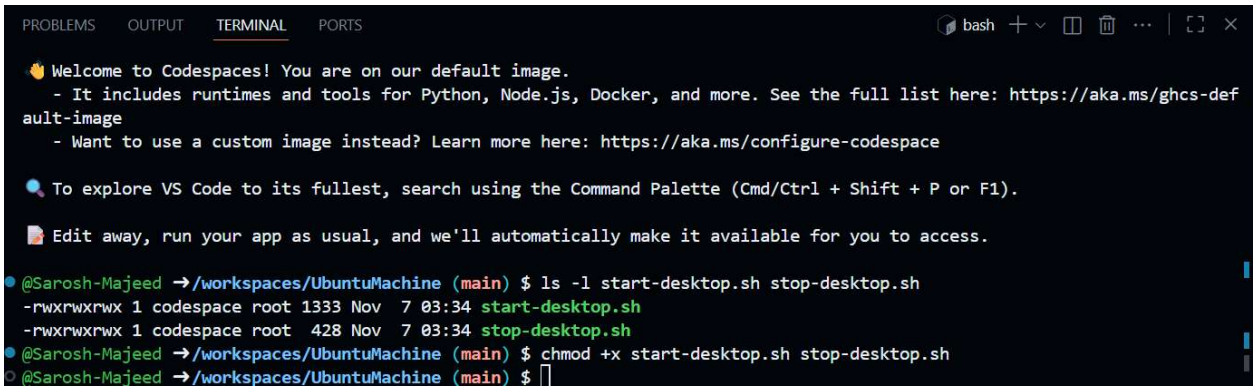
In the Codespace terminal list files in the repo root and show the start script and stop script exist:

ls -l start-desktop.sh stop-desktop.sh

If not executable, make it executable:

chmod +x start-desktop.sh stop-desktop.sh

Save a screenshot showing the ls -l output (file listing) and the chmod command if applied:



Run the start script inside the Codespace terminal

In the Codespace terminal run:

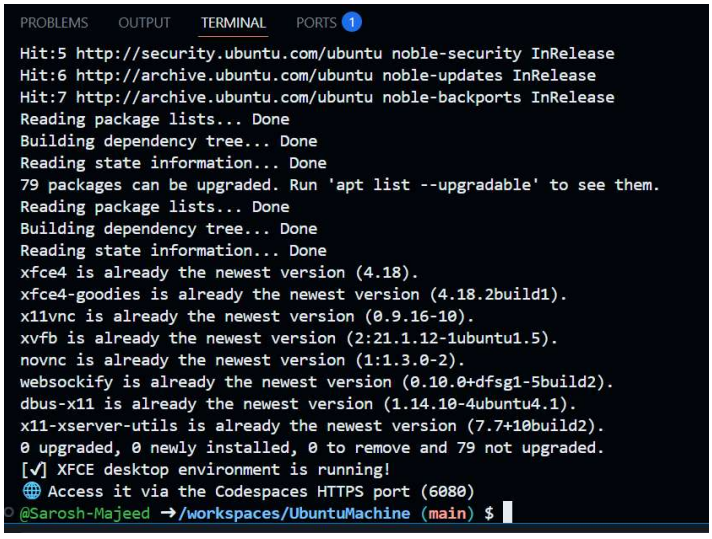
 # Ensure the start script is executable

chmod +x start-desktop.sh

 # Start the desktop GUI

./start-desktop.sh

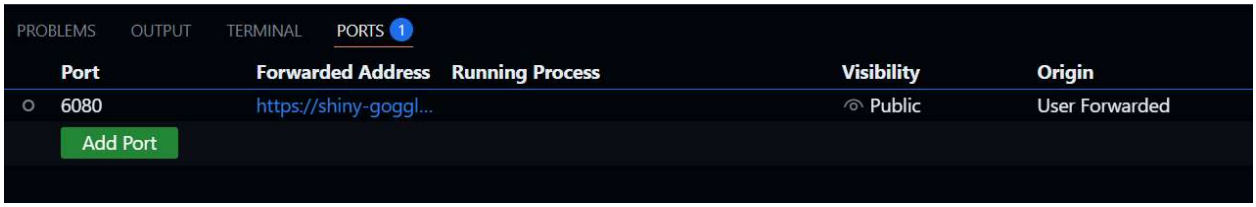
Capture the terminal output showing successful start messages.



Verify forwarded ports in Codespaces (Ports view)

Open the Codespaces "Ports" panel / view and confirm port 6080 is forwarded and visible.

Save a screenshot of the Ports view showing port 6080 and its status:



Open forwarded port 6080 and connect to VNC HTML page

In the Codespaces UI, open the forwarded port's preview URL or copy the forwarded URL and open it in your browser.

Visit the port 6080 address and click the vnc.html link.

When prompted for a password enter:

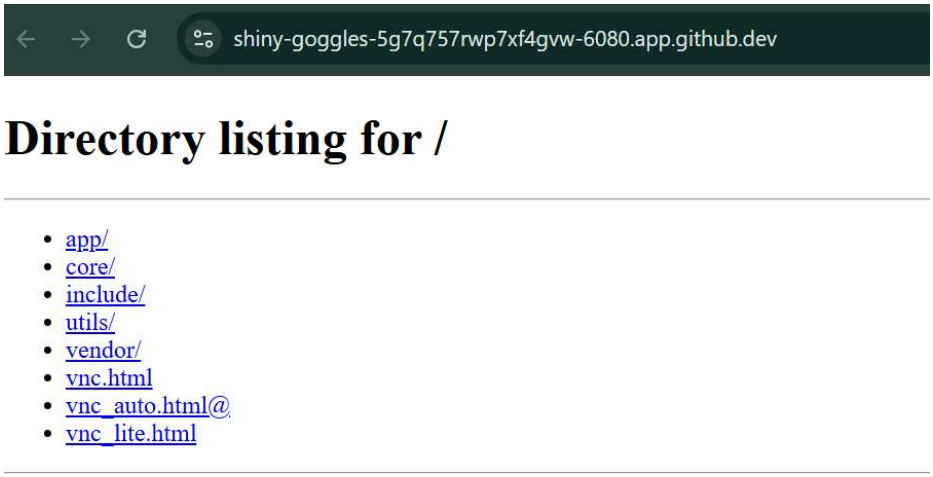
codespace

Capture screenshots of:

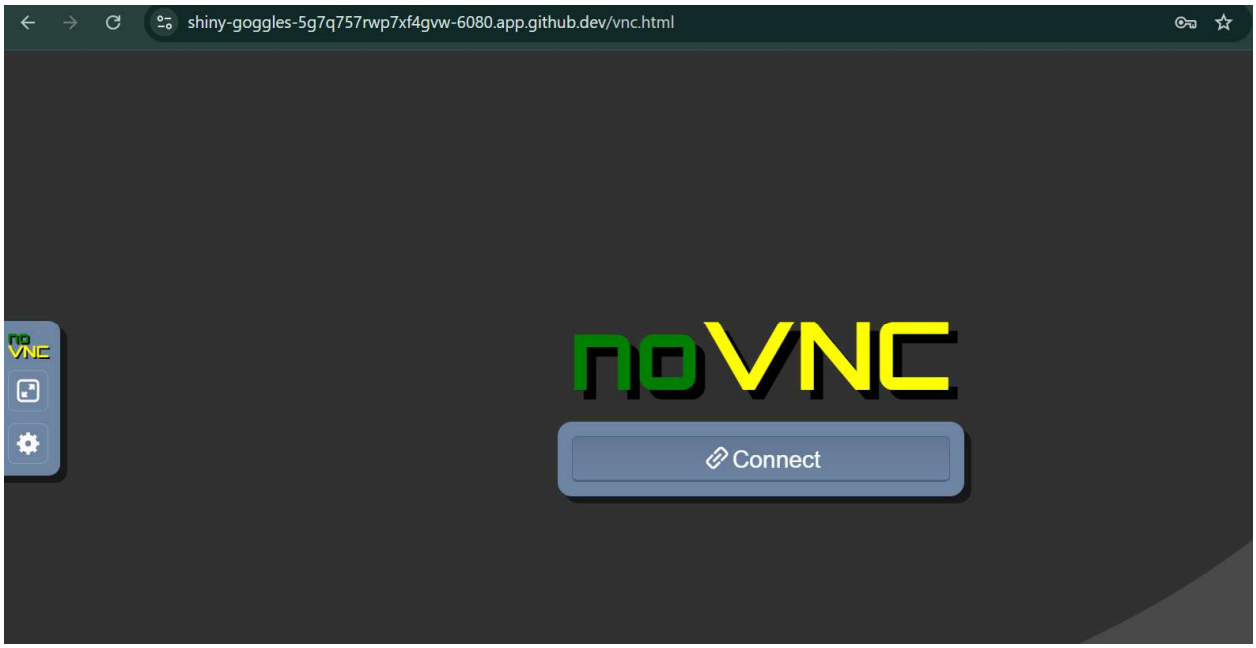
The browser showing the forwarded port URL in the address bar / Codespaces preview

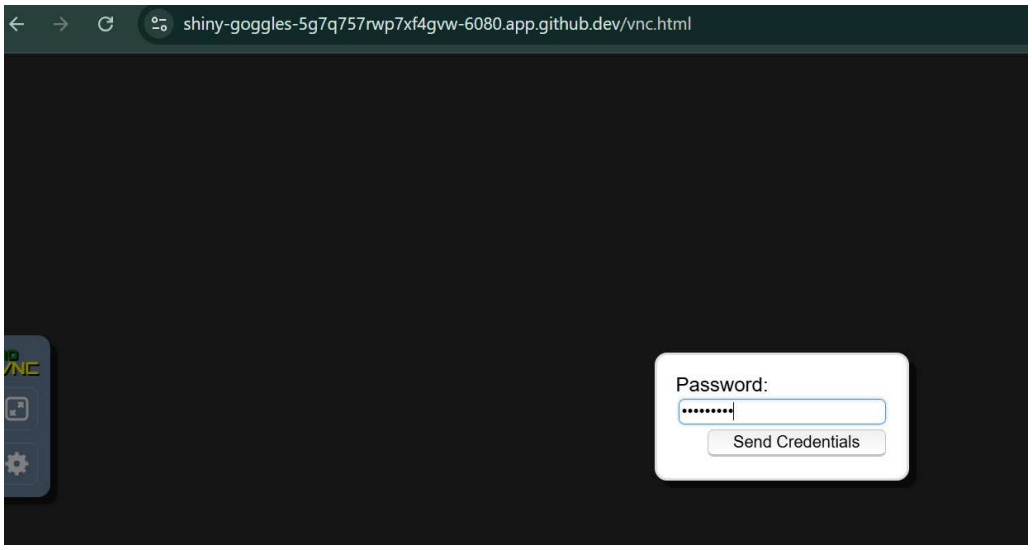
The VNC password prompt (showing password field; do NOT include typed password in a screenshot): task14_vnc_password_prompt.png

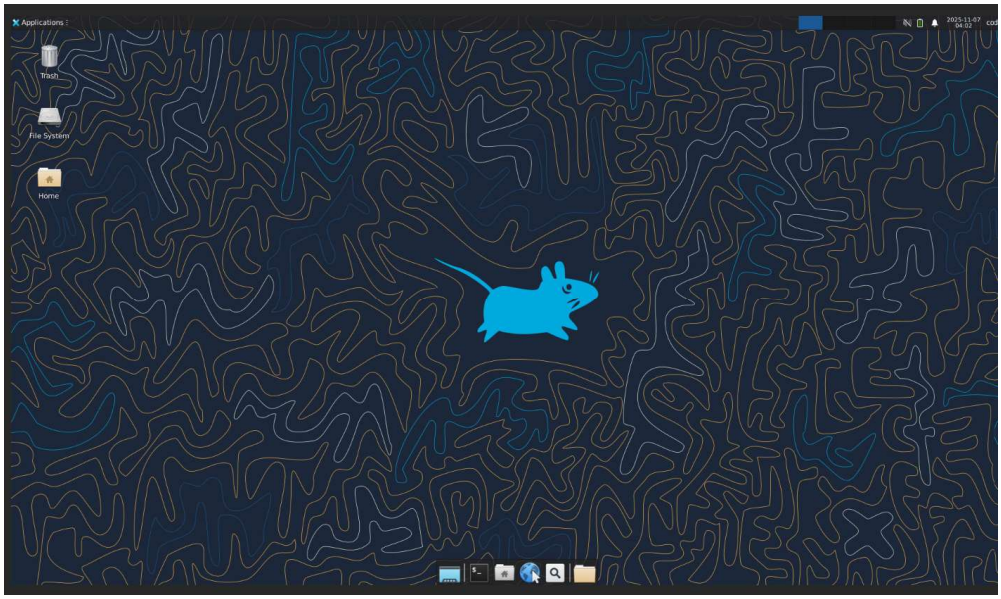
The VNC session after successful connection showing the GUI/desktop: task14_vnc_desktop.png



(Optional) A focused screenshot of vnc.html UI showing the "Connect" button before/after connecting: task14_vnc_connect.png


shiny-goggles-5g7q757rwp7xf4gvw-6080.app.github.dev/vnc.html
noVNC
Connect


shiny-goggles-5g7q757rwp7xf4gvw-6080.app.github.dev/vnc.html
Password:
Send Credentials


Applications
Trash
File System
Home

Stop the GUI

When finished, return to the Codespace terminal and run:

./stop-desktop.sh

Capture the terminal output that shows the GUI stopping and any cleanup messages.

```
@Sarosh-Majeed ➜/workspaces/UbuntuMachine (main) $ ./stop-desktop.sh
[*] Stopping noVNC server...

Terminating WebSockets proxy (19781)
[*] Stopping x11vnc server...
In exit
Terminating child 23137
Process Process-16:
07/11/2025 04:03:29 client_count: 0
07/11/2025 04:03:29 Restored X server key autorepeat to: 1
07/11/2025 04:03:29 Client 127.0.0.1 gone
07/11/2025 04:03:29 Statistics            events    Transmit/ RawEquiv ( saved)
07/11/2025 04:03:29  ServerCutText       :      1 |        16/       16 (  0.0%)
07/11/2025 04:03:29  FramebufferUpdate   :      4 |         0/        0 (  0.0%)
07/11/2025 04:03:29  LastRect            :      1 |        12/       12 (  0.0%)
07/11/2025 04:03:29  tight               :     57 |    633397/  8297520 ( 92.4%)
07/11/2025 04:03:29  RichCursor          :      1 |      2400/     2400 (  0.0%)
07/11/2025 04:03:29  ExtendedDesktopSize :      1 |        32/       32 (  0.0%)
07/11/2025 04:03:29  TOTALS              :     65 |    635857/  8299980 ( 92.3%)
```

Troubleshooting tips:

If port 6080 is not visible, check Codespaces "Ports" view and forward it manually.

If the VNC page fails to connect, verify the start-desktop.sh completed without errors and that the VNC server is listening on the expected port inside the Codespace.

If Codespaces is unavailable for your account, consider forking and running the same scripts on another cloud VM that forwards port 6080 and adapt screenshots/file names accordingly.
