**Bash scripting task – 02**

1. **Create a Bash script to check if a directory is available or not.**

**Commands used:**

**vi <file\_name.bash>** - To create and edit bash script.

**chmod 755 <file\_name.bash>** – To change permissions.

**Cat <file\_name.bash>** - To check content once again before execute.

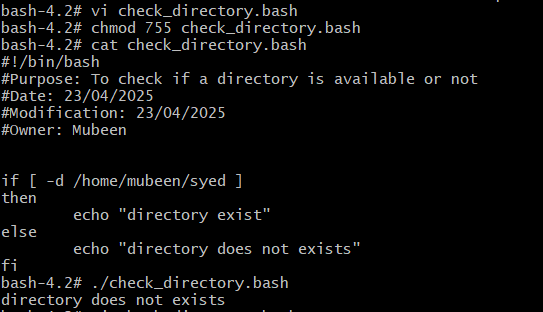
**./ <file\_name.bash>** - To run the script.

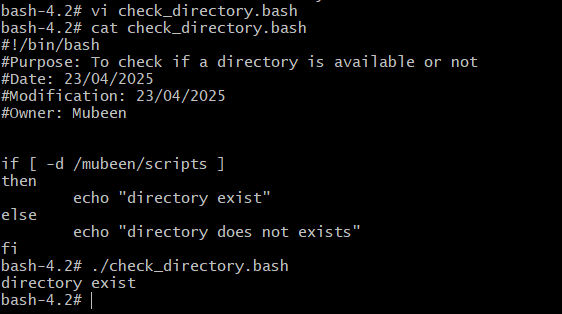
**If-else** condition is used, as question itself indicates ‘if’.

**[ -d <path> ]** – we are using test command, where -d is directory.

If test is successful or else test is failed we can print accordingly using **echo**.

directory exist/directory does not exist.





**2) Create a bash script which will create multiple files.**

**Commands used:**

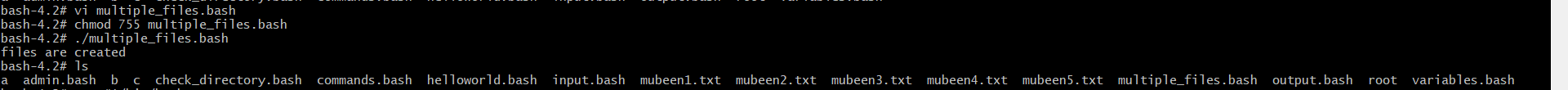
**vi <file\_name.bash>** - To create and edit bash script.

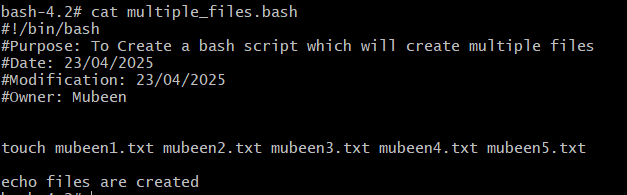
**chmod 755 <file\_name.bash>** – To change permissions.

**Cat <file\_name.bash>** - To check content once again before execute.

**./ <file\_name.bash>** - To run the script.

**touch <file1> <file2> <file3> <file4>** as required to create multiple files and **echo** to print text.





**3) Create a bash script to take backup of a directory.**  
**Commands used:**

**vi <file\_name.bash>** - To create and edit bash script.

**chmod 755 <file\_name.bash>** – To change permissions.

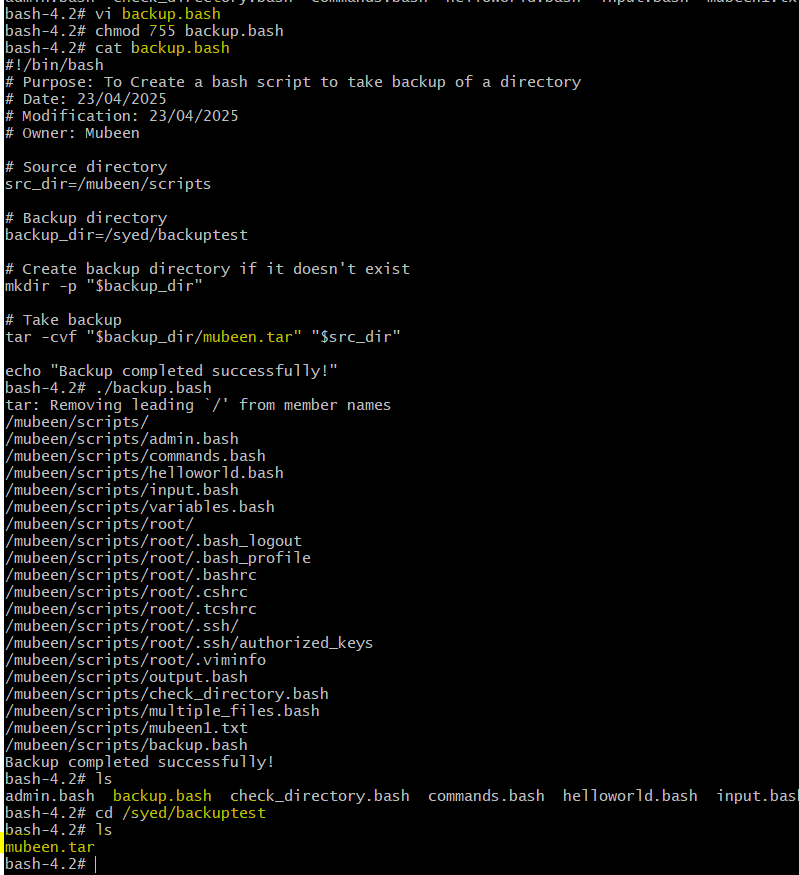
**Cat <file\_name.bash>** - To check content once again before execute.

**./ <file\_name.bash>** - To run the script.

Two path **variables** are given as **src\_dir & backup\_dir.**

**mkdir -p $backup\_dir –** To create a backup path.

**tar -cvf < src\_dir.tar> < backup\_dir >** -This command is used to create backup and **echo** line print for confirmation.



**4) Create a bash script to install nginx in ec2 server.**

**Commands used:**

**vi <file\_name.bash>** - To create and edit bash script.

**chmod 755 <file\_name.bash>** – To change permissions.

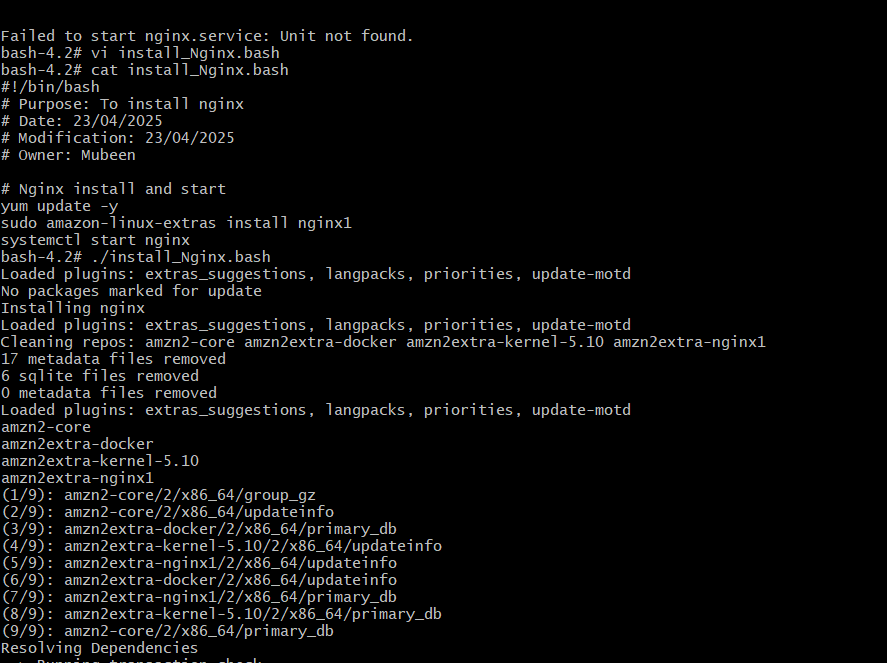
**Cat <file\_name.bash>** - To check content once again before execute.

**./ <file\_name.bash>** - To run the script.

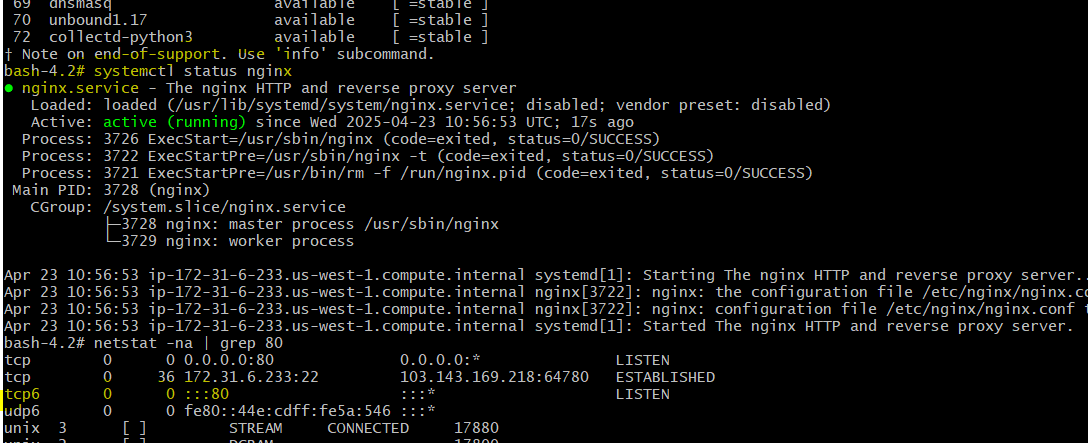
**Yum update -y** – To automatic update all installed packages.

**sudo amazon-linux-extras install nginx1** - Command from EC2 server shown in error to install nginx successfully.

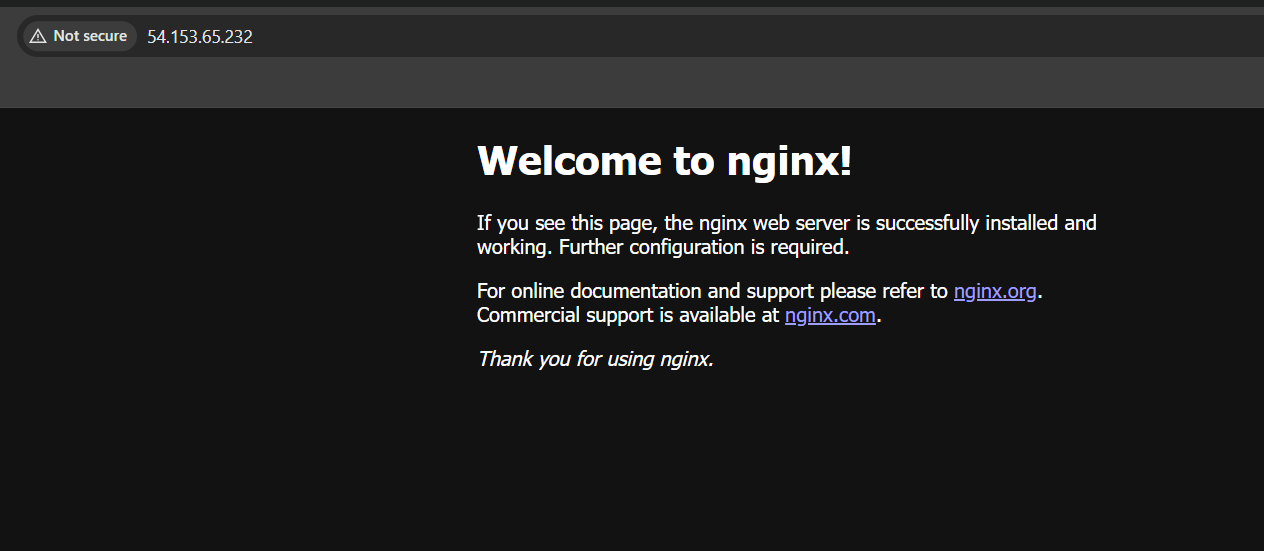
**systemctl start nginx** – To start the service.



**systemctl status nginx** – To check the status of the service whether it is active/inactive/failed.



Testing nginx web server via browser.



**5) Create a bash script to install ApacheTomcat in ec2 server.**

**Commands used:**

**vi <file\_name.bash>** - To create and edit bash script.

**chmod 755 <file\_name.bash>** – To change permissions.

**Cat <file\_name.bash>** - To check content once again before execute.

**./ <file\_name.bash>** - To run the script.

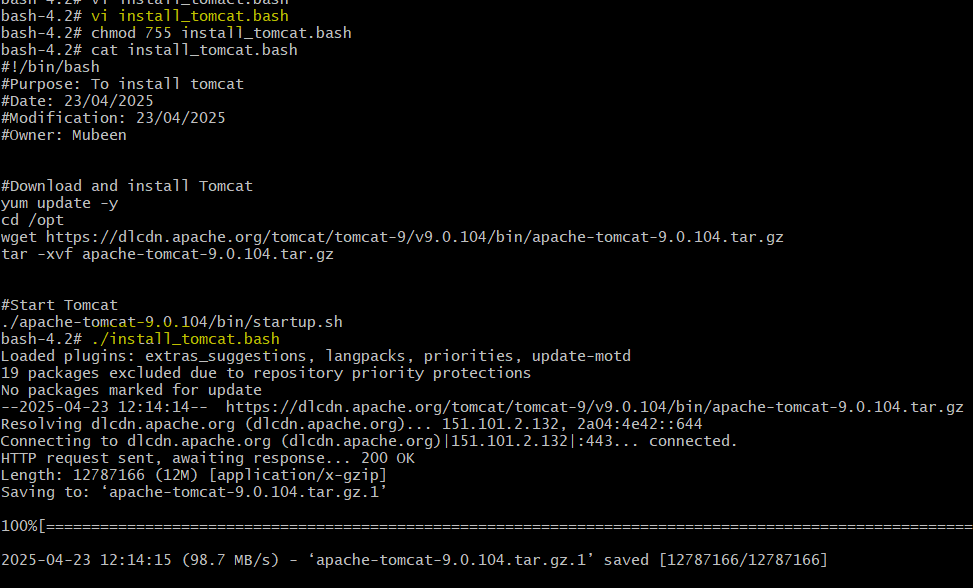
**yum update -y -** To automatic update all installed packages.

**cd /opt –** To change directory to opt for downloading tomcat tar file, because opt is used for third party apps.

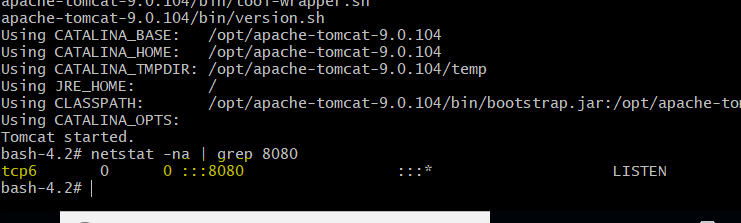
**wget** [**https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.104/bin/apache-tomcat-9.0.104.tar.gz**](https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.104/bin/apache-tomcat-9.0.104.tar.gz) **-** To download the tomcat tar file.

**tar -xvf apache-tomcat-9.0.104.tar.gz –** To extract the tar file.

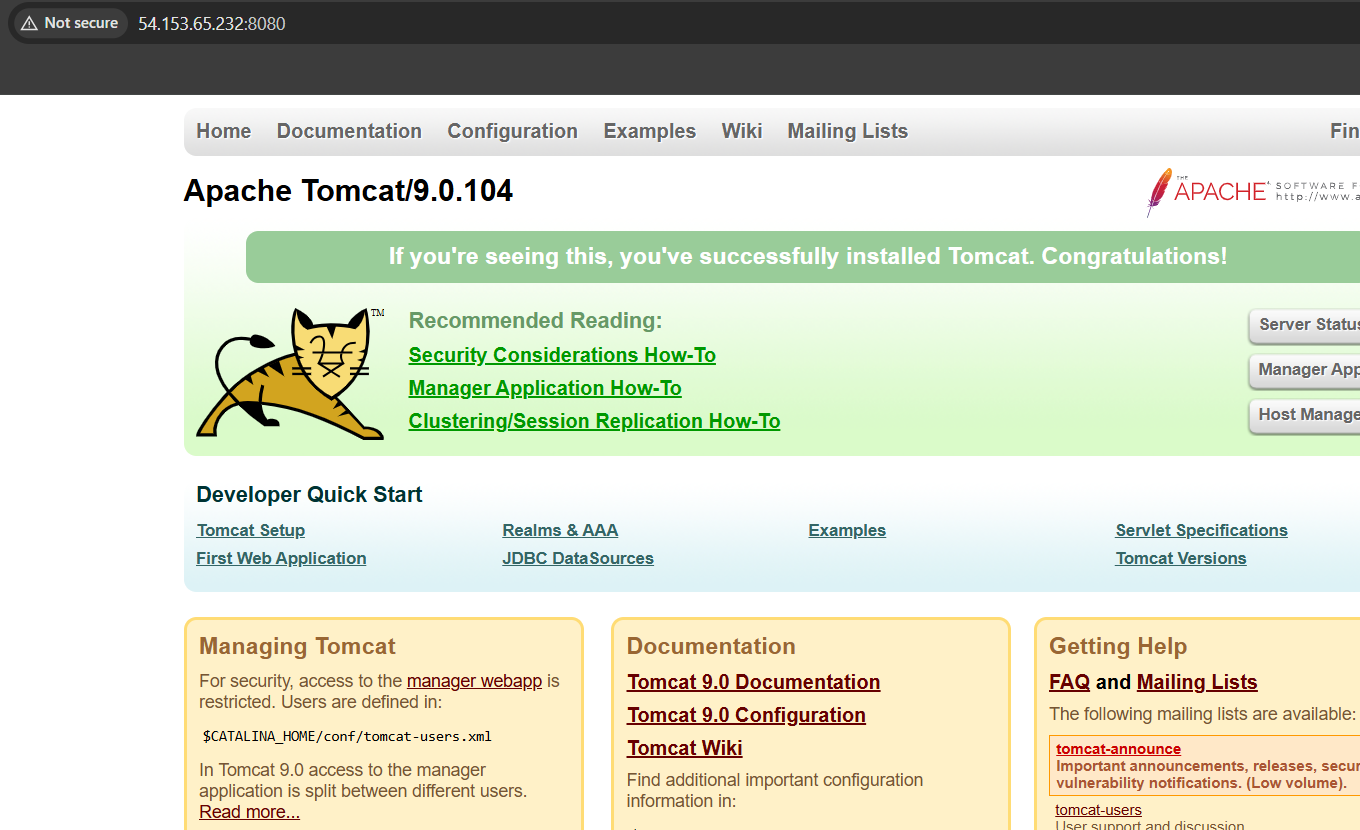
**./apache-tomcat-9.0.104/bin/startup.sh –** To start the Tomcat ./ is used with path.



**netstat -na | grep 8080** – To check the port number whether service is sunning or not.



Testing tomcat app server via browser.



**6) Create a bash script to check list if nginx service is running or not,if not running then script should start the service.**

**Commands used:**

**vi <file\_name.bash>** - To create and edit bash script.

**chmod 755 <file\_name.bash>** – To change permissions.

**Cat <file\_name.bash>** - To check content once again before execute.

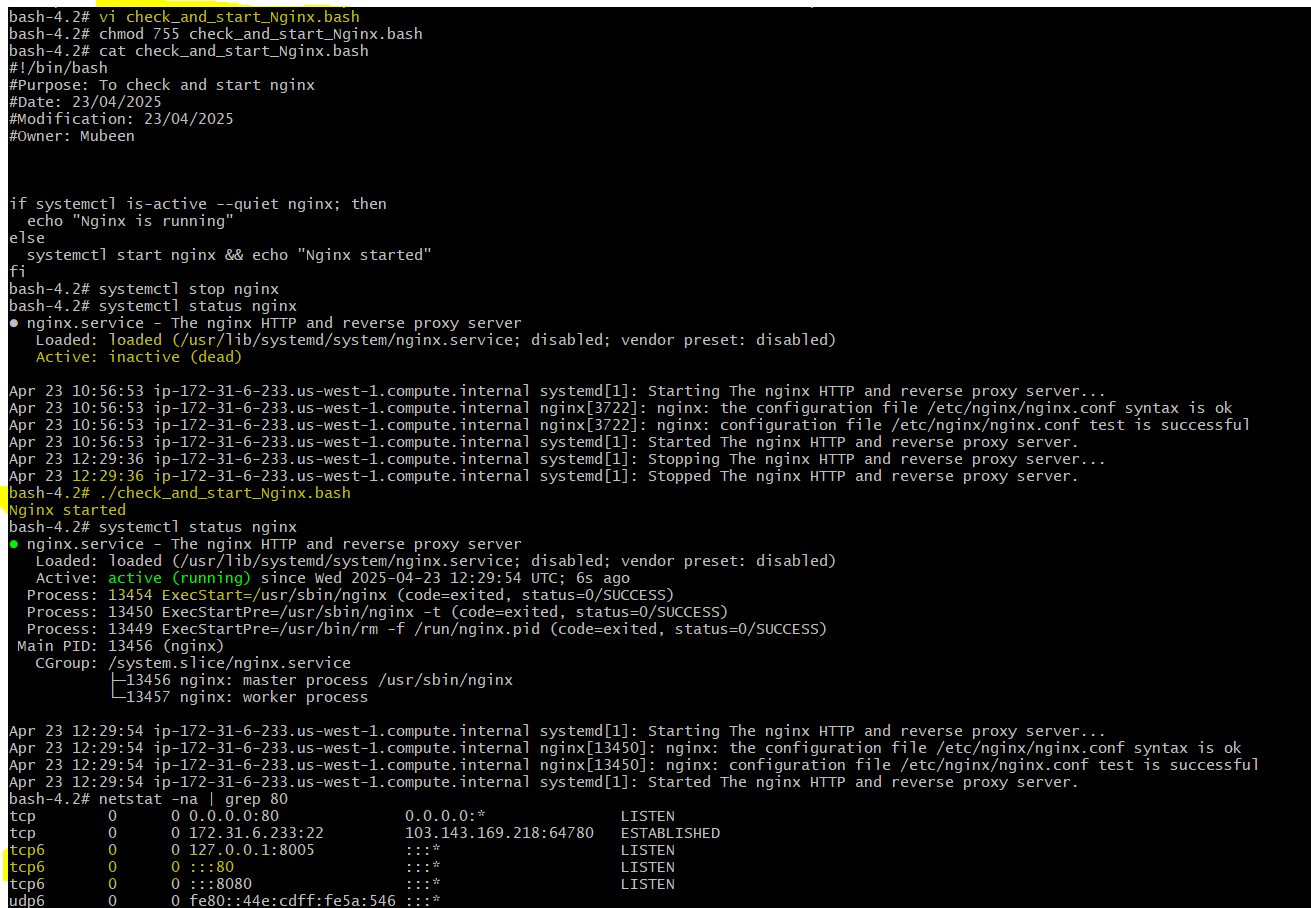
**./ <file\_name.bash>** - To run the script.

**If-else** condition is used, **if** nginx is active then **echo "Nginx is running"** else using command **echo "Nginx started".**

**systemctl is-active --quiet nginx –** Checks whether nginx is active or not.

**systemctl start nginx** – To start the service.

**systemctl status nginx** – To check the status of the service whether it is active/inactive/failed.



**7) Create a bash script for calculator.**

**Commands used:**

**vi <file\_name.bash>** - To create and edit bash script.

**chmod 755 <file\_name.bash>** – To change permissions.

**Cat <file\_name.bash>** - To check content once again before execute.

**./ <file\_name.bash>** - To run the script.

It displays a simple calculator menu using **echo** with Add, subtract, multiply & Division as **choice**.  
Using **read** we get the **choice** from user as (1/2/3/4) as variable c. Again, using the **read** we get 2 variables **n1 & n2** as input from the user.  
**Case** statement is used to match the selected **choice** and perform the operation selected by user.

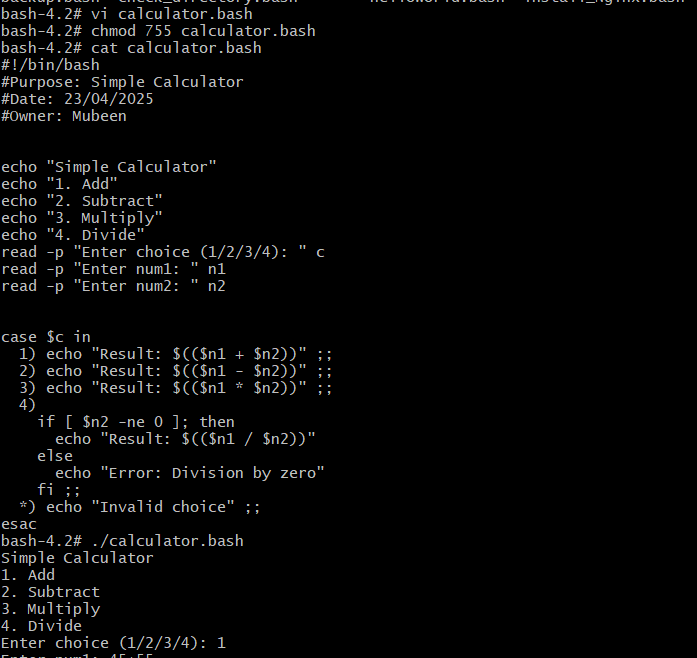
**$(($n1 + $n2))** – used for addition**.**

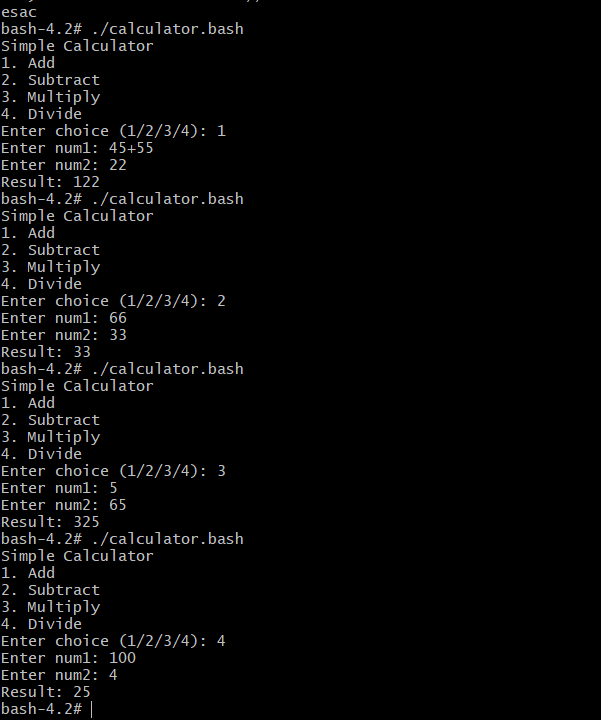
**$(($n1 - $n2))** - used for subtraction.

**$(($n1 \* $n2)) -** used for multiplication. **$(($n1 / $n2))** – used for division.

**[ $n2 -ne 0 ]**  is used to checked second number should not be 0 to avoid errors. If the user provides input for n2 as 0 then **echo "Error: Division by zero"** is printed.

If user selects any other option then 1/2/3/4 by using **echo** it prints “invalid choice”.





**8) Create a bash script to check if directory is available or not, if not then create a directory.**

**Commands used:**

**vi <file\_name.bash>** - To create and edit bash script.

**chmod 755 <file\_name.bash>** – To change permissions.

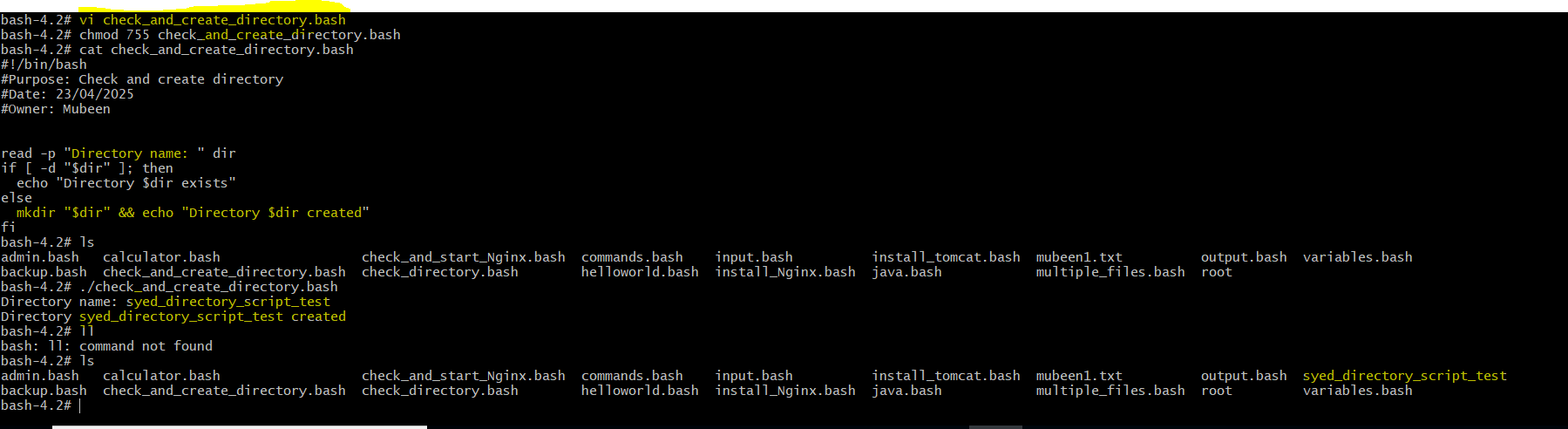
**Cat <file\_name.bash>** - To check content once again before execute.

**./ <file\_name.bash>** - To run the script.

**read -p "Directory name: " dir –** read is to take directory name from the user as input, where dir is variable.

**[ -d <path> ]** – Test command to check whether directory is available, where -d is directory.

**mkdir "$dir" –** To create a directory with variable ‘dir”.  
**If-else** condition is used, as question itself indicates ‘if’. If **echo "Directory $dir exists"** else **echo "Directory $dir created"** ends with fi.



**9) Create bash script to delete last 3 lines for a file.**

**Commands used:**

**vi <file\_name.bash>** - To create and edit bash script.

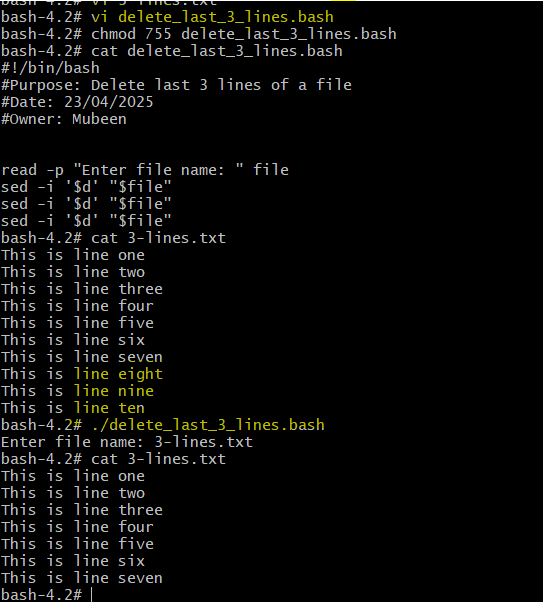
**chmod 755 <file\_name.bash>** – To change permissions.

**Cat <file\_name.bash>** - To check content once again before execute.

**./ <file\_name.bash>** - To run the script.

**“Read input”** is used to get file name to be executed.

**sed -i ‘$d’ <file\_name>** - This command is used for deleting last line so it is entered   
3 times.



**Exercise Crontab Entries:**

1) April 5th Midnight.

2) 5 th of Every November,Jan,June if it is a Thursday.

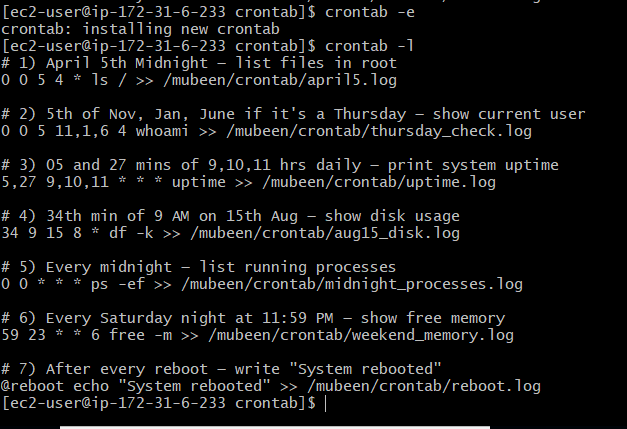
3) At 05 and 27th minutes of 9,10,11 hours everyday.

4) 34 min. of 9th hour on 15th Aug.

5) Every midnight.

6) Every Weekend ( Saturday night 11.59 ).

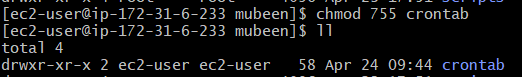
7) After every reboot.



**Commands used:**

crontab -e – To edit the crontab file.  
crontab -l - To list out the content in the crontab file.

Make sure to check user/group and permissions and change accordingly user and group to self ( Mubeen : Mubeen ) and keep permissions 755.



**Content of crontab:**

5 stars \*\*\*\*\*, command/script and the Path.

**5 starts:**

1st star is for minutes (0-59),

2nd star is for hours (0-23),

3rd start is for days (1-31),

4th start is for months (1-12),

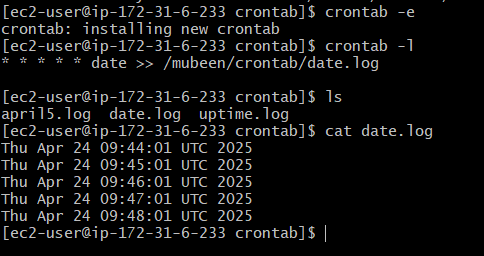
5th star is for weekday (mon -sun).

**Commands/Scripts:**

Any command or script required by user.

**Path:**

Destination where the command/script to be executed.

The log file is only created when the cron job runs. So here is the example to check cron job is running or not.  


**The-End**