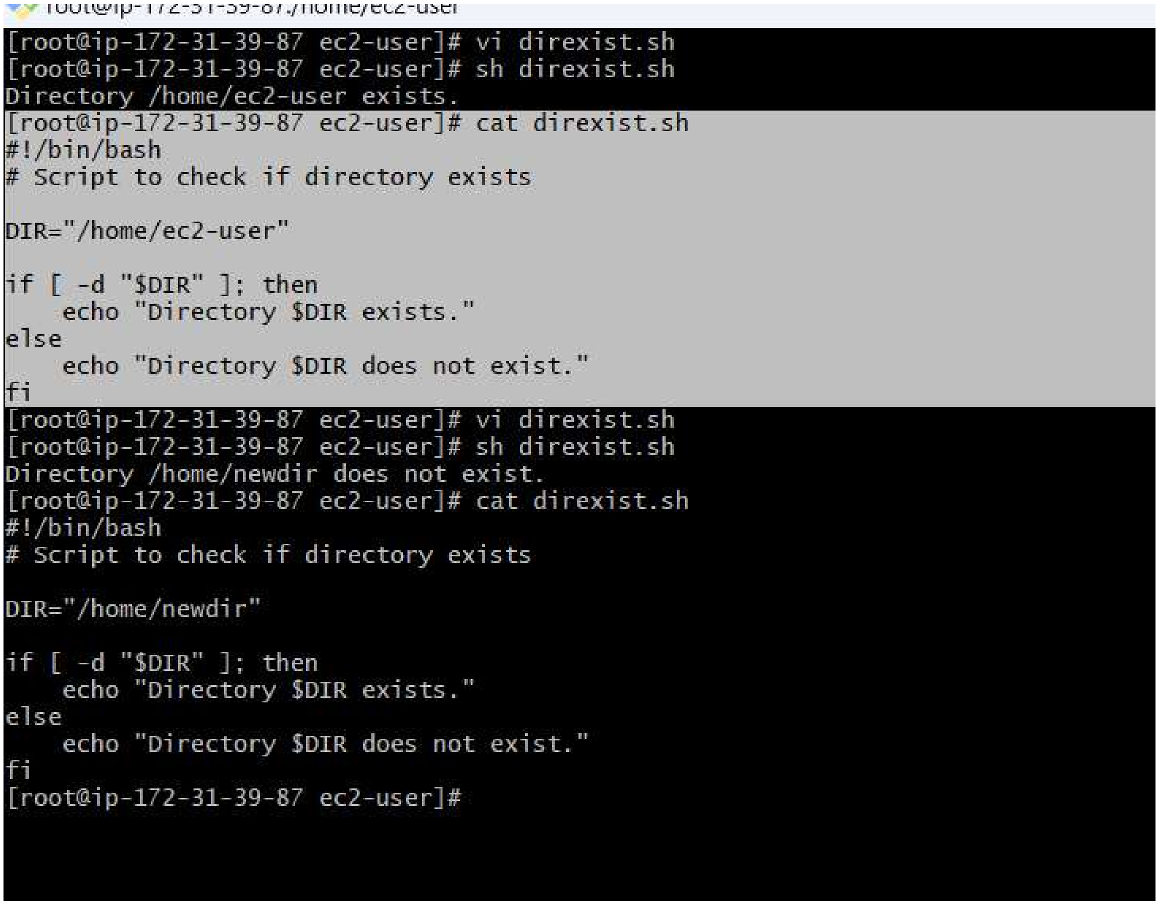
1) Create on Bash script to check if a directory is available or not.

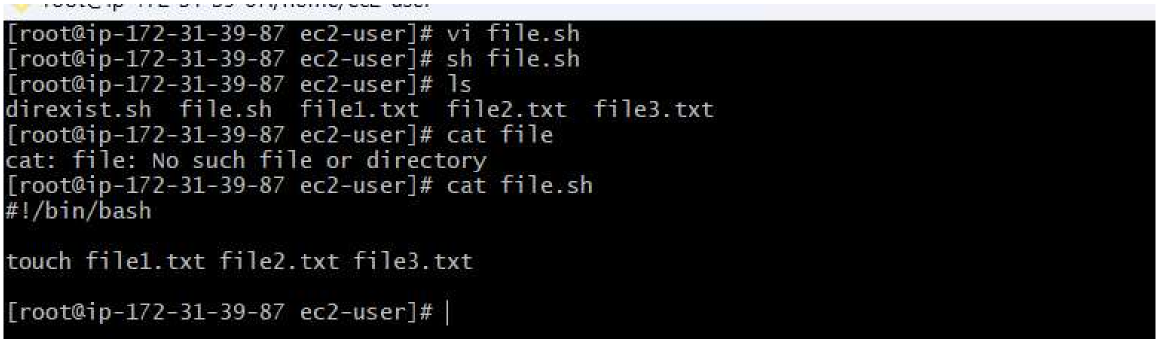


```
[root@ip-172-31-39-87 ec2-user]# vi direxist.sh
[root@ip-172-31-39-87 ec2-user]# sh direxist.sh
Directory /home/ec2-user exists.
[root@ip-172-31-39-87 ec2-user]# cat direxist.sh
#!/bin/bash
# Script to check if directory exists

DIR="/home/ec2-user"

if [ -d "$DIR" ]; then
    echo "Directory $DIR exists."
else
    echo "Directory $DIR does not exist."
fi
[root@ip-172-31-39-87 ec2-user]# vi direxist.sh
[root@ip-172-31-39-87 ec2-user]# sh direxist.sh
Directory /home/newdir does not exist.
[root@ip-172-31-39-87 ec2-user]# cat direxist.sh
#!/bin/bash
# Script to check if directory exists

DIR="/home/newdir"

if [ -d "$DIR" ]; then
    echo "Directory $DIR exists."
else
    echo "Directory $DIR does not exist."
fi
[root@ip-172-31-39-87 ec2-user]#
```

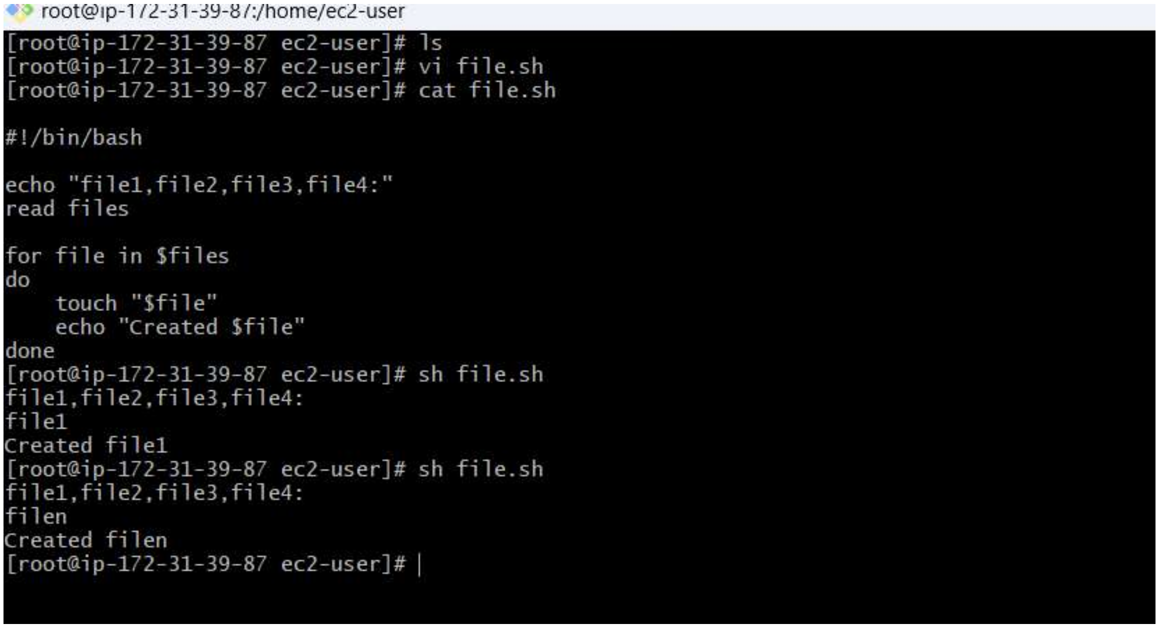
```
#!/bin/bash


echo "Enter directory path:"

read dir


if [ -d "$dir" ]; then

  echo "Directory exists."

else

  echo "Directory does not exist."
```

fi

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



```
[root@ip-172-31-39-87 ec2-user]# vi file.sh
[root@ip-172-31-39-87 ec2-user]# sh file.sh
[root@ip-172-31-39-87 ec2-user]# ls
direxist.sh  file.sh  file1.txt  file2.txt  file3.txt
[root@ip-172-31-39-87 ec2-user]# cat file
cat: file: No such file or directory
[root@ip-172-31-39-87 ec2-user]# cat file.sh
#!/bin/bash

touch file1.txt file2.txt file3.txt

[root@ip-172-31-39-87 ec2-user]# |
```



```
[root@ip-172-31-39-87 ec2-user]# ls
[root@ip-172-31-39-87 ec2-user]# vi file.sh
[root@ip-172-31-39-87 ec2-user]# cat file.sh

#!/bin/bash

echo "file1,file2,file3,file4:"
read files

for file in $files
do
    touch "$file"
    echo "Created $file"
done
[root@ip-172-31-39-87 ec2-user]# sh file.sh
file1,file2,file3,file4:
file1
Created file1
[root@ip-172-31-39-87 ec2-user]# sh file.sh
file1,file2,file3,file4:
filen
Created filen
[root@ip-172-31-39-87 ec2-user]# |
```

#!/bin/bash

echo "Enter file names (space separated):"

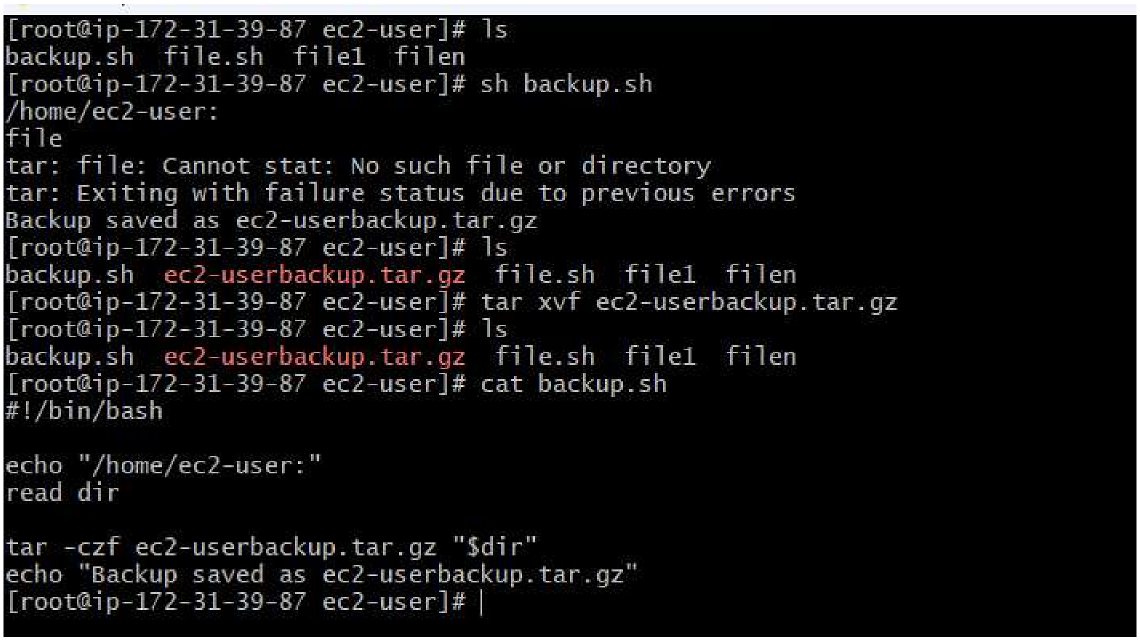
read files

for file in $files

do

```
    touch "$file"

    echo "Created $file"

done
```

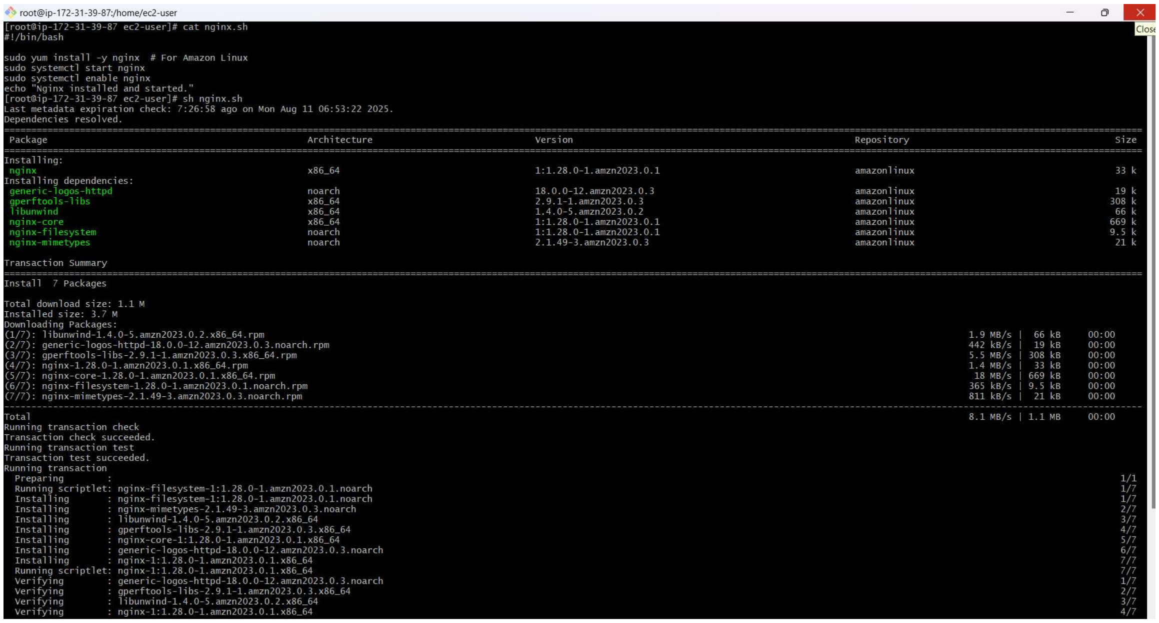
3) Create a bash script to take backup of a directory.



```
[root@ip-172-31-39-87 ec2-user]# ls
backup.sh  file.sh  file1  filen
[root@ip-172-31-39-87 ec2-user]# sh backup.sh
/home/ec2-user:
file
tar: file: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
Backup saved as ec2-userbackup.tar.gz
[root@ip-172-31-39-87 ec2-user]# ls
backup.sh  ec2-userbackup.tar.gz  file.sh  file1  filen
[root@ip-172-31-39-87 ec2-user]# tar xvf ec2-userbackup.tar.gz
[root@ip-172-31-39-87 ec2-user]# ls
backup.sh  ec2-userbackup.tar.gz  file.sh  file1  filen
[root@ip-172-31-39-87 ec2-user]# cat backup.sh
#!/bin/bash

echo "/home/ec2-user:"
read dir

tar -czf ec2-userbackup.tar.gz "$dir"
echo "Backup saved as ec2-userbackup.tar.gz"
[root@ip-172-31-39-87 ec2-user]# |
```

```
#!/bin/bash


echo "Enter directory to backup:"

read dir


tar -czf backup.tar.gz "$dir"

echo "Backup saved as backup.tar.gz"
```

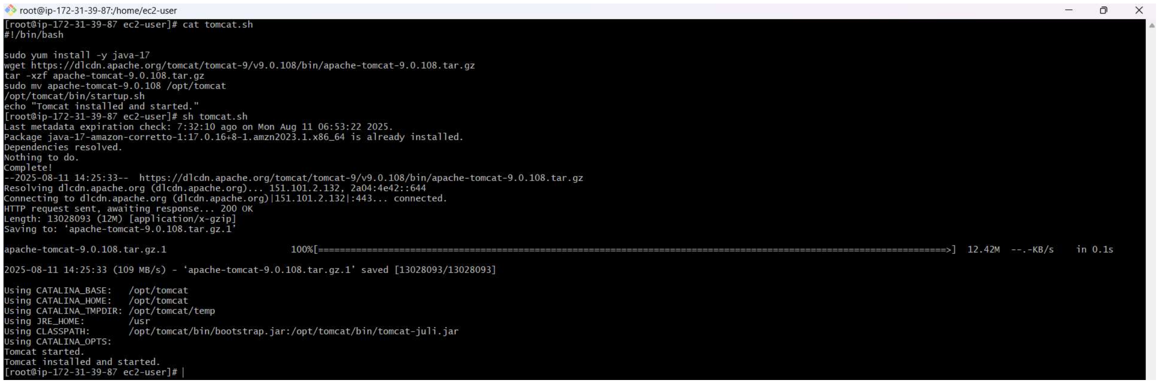
4) Create a bash script to install nginx in ec2 server.



```
[root@ip-172-31-39-87 ec2-user]# cat nginx.sh
#!/bin/bash

sudo yum install -y nginx  # For Amazon Linux
sudo systemctl start nginx
sudo systemctl enable nginx
echo "Nginx installed and started."
[root@ip-172-31-39-87 ec2-user]# sh nginx.sh
Last metadata expiration check: 7:26:58 ago on Mon Aug 11 06:53:22 2025.
Dependencies resolved.
==============================================================================================================================================
 Package                         Architecture              Version                                  Repository                        Size
==============================================================================================================================================
Installing:
 nginx                           x86_64                    1:1.28.0-1.amzn2023.0.1                  amazonlinux                       33 k
Installing dependencies:
 generic-logos-httpd             noarch                    18.0.0-12.amzn2023.0.3                   amazonlinux                       19 k
 gperftools-libs                 x86_64                    2.9.1-1.amzn2023.0.3                     amazonlinux                      308 k
 libunwind                       x86_64                    1.4.0-5.amzn2023.0.2                     amazonlinux                       66 k
 nginx-core                      x86_64                    1:1.28.0-1.amzn2023.0.1                  amazonlinux                      669 k
 nginx-filesystem                noarch                    1:1.28.0-1.amzn2023.0.1                  amazonlinux                      9.5 k
 nginx-mimetypes                 noarch                    2.1.49-3.amzn2023.0.3                    amazonlinux                       21 k

Transaction Summary
==============================================================================================================================================
Install  7 Packages

Total download size: 1.1 M
Installed size: 3.7 M
Downloading Packages:
(1/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                                                                1.9 MB/s |  66 kB     00:00
(2/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                                                    442 kB/s |  19 kB     00:00
(3/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                                                          5.5 MB/s | 308 kB     00:00
(4/7): nginx-1.28.0-1.amzn2023.0.1.x86_64.rpm                                                                   1.4 MB/s |  33 kB     00:00
(5/7): nginx-core-1.28.0-1.amzn2023.0.1.x86_64.rpm                                                               18 MB/s | 669 kB     00:00
(6/7): nginx-filesystem-1.28.0-1.amzn2023.0.1.noarch.rpm                                                        365 kB/s | 9.5 kB     00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm                                                         811 kB/s |  21 kB     00:00
----------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                                           8.1 MB/s | 1.1 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                                        1/1
  Running scriptlet: nginx-filesystem-1:1.28.0-1.amzn2023.0.1.noarch                                                                        1/7
  Installing       : nginx-filesystem-1:1.28.0-1.amzn2023.0.1.noarch                                                                        1/7
  Installing       : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                                                           2/7
  Installing       : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                                                                  3/7
  Installing       : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                                                            4/7
  Installing       : nginx-core-1:1.28.0-1.amzn2023.0.1.x86_64                                                                              5/7
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                                                      6/7
  Installing       : nginx-1:1.28.0-1.amzn2023.0.1.x86_64                                                                                   7/7
  Running scriptlet: nginx-1:1.28.0-1.amzn2023.0.1.x86_64                                                                                   7/7
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                                                      1/7
  Verifying        : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                                                            2/7
  Verifying        : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                                                                  3/7
  Verifying        : nginx-1:1.28.0-1.amzn2023.0.1.x86_64                                                                                   4/7
```

```
#!/bin/bash

sudo yum install -y nginx  # For Amazon Linux

sudo systemctl start nginx

sudo systemctl enable nginx

echo "Nginx installed and started."
```

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



```
[root@ip-172-31-39-87 ec2-user]# cat tomcat.sh
#!/bin/bash

sudo yum install -y java-17
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.108/bin/apache-tomcat-9.0.108.tar.gz
tar -xzf apache-tomcat-9.0.108.tar.gz
sudo mv apache-tomcat-9.0.108 /opt/tomcat
/opt/tomcat/bin/startup.sh
echo "Tomcat installed and started."
[root@ip-172-31-39-87 ec2-user]# sh tomcat.sh
Last metadata expiration check: 7:32:10 ago on Mon Aug 11 06:53:22 2025.
Package java-17-amazon-corretto-1:17.0.16+8-1.amzn2023.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
--2025-08-11 14:25:33--  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.108/bin/apache-tomcat-9.0.108.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13028093 (12M) [application/x-gzip]
Saving to: ‘apache-tomcat-9.0.108.tar.gz.1’

apache-tomcat-9.0.108.tar.gz.1                     100%[=============================================================================================================>]  12.42M  --.-KB/s    in 0.1s

2025-08-11 14:25:33 (109 MB/s) - ‘apache-tomcat-9.0.108.tar.gz.1’ saved [13028093/13028093]

Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
Tomcat installed and started.
[root@ip-172-31-39-87 ec2-user]# |
```

#!/bin/bash

sudo yum install -y java-11-openjdk wget

wget https://downloads.apache.org/tomcat/tomcat-10/v10.1.14/bin/apache-tomcat-10.1.14.tar.gz

tar -xzf apache-tomcat-10.1.14.tar.gz

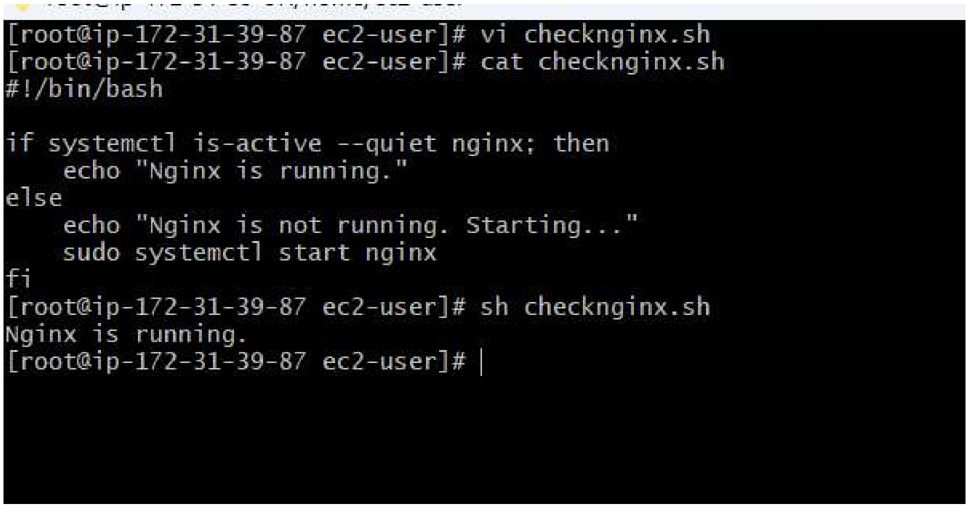
sudo mv apache-tomcat-10.1.14 /opt/tomcat

/opt/tomcat/bin/startup.sh

echo "Tomcat installed and started."

6) Create a bash script to check list if nginx service is running or not,if not running

then script should start the service.



```
[root@ip-172-31-39-87 ec2-user]# vi checknginx.sh
[root@ip-172-31-39-87 ec2-user]# cat checknginx.sh
#!/bin/bash

if systemctl is-active --quiet nginx; then
    echo "Nginx is running."
else
    echo "Nginx is not running. Starting..."
    sudo systemctl start nginx
fi
[root@ip-172-31-39-87 ec2-user]# sh checknginx.sh
Nginx is running.
[root@ip-172-31-39-87 ec2-user]# |
```

#!/bin/bash

if systemctl is-active --quiet nginx; then

    echo "Nginx is running."

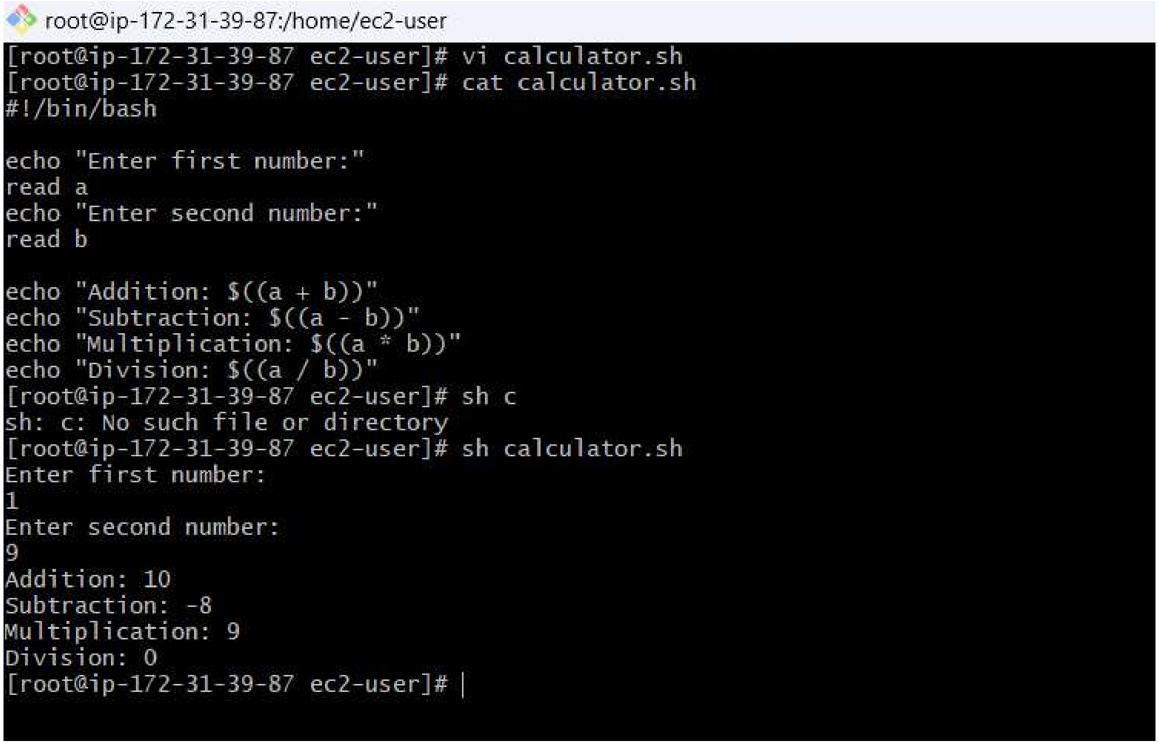
else

    echo "Nginx is not running. Starting..."

    sudo systemctl start nginx

fi

7) Create a bash script for calculator.



root@ip-172-31-39-87:/home/ec2-user

```
[root@ip-172-31-39-87 ec2-user]# vi calculator.sh
[root@ip-172-31-39-87 ec2-user]# cat calculator.sh
#!/bin/bash

echo "Enter first number:"
read a
echo "Enter second number:"
read b

echo "Addition: $((a + b))"
echo "Subtraction: $((a - b))"
echo "Multiplication: $((a * b))"
echo "Division: $((a / b))"
[root@ip-172-31-39-87 ec2-user]# sh c
sh: c: No such file or directory
[root@ip-172-31-39-87 ec2-user]# sh calculator.sh
Enter first number:
1
Enter second number:
9
Addition: 10
Subtraction: -8
Multiplication: 9
Division: 0
[root@ip-172-31-39-87 ec2-user]# |
```

#!/bin/bash

echo "Enter first number:"

read a

echo "Enter second number:"

read b

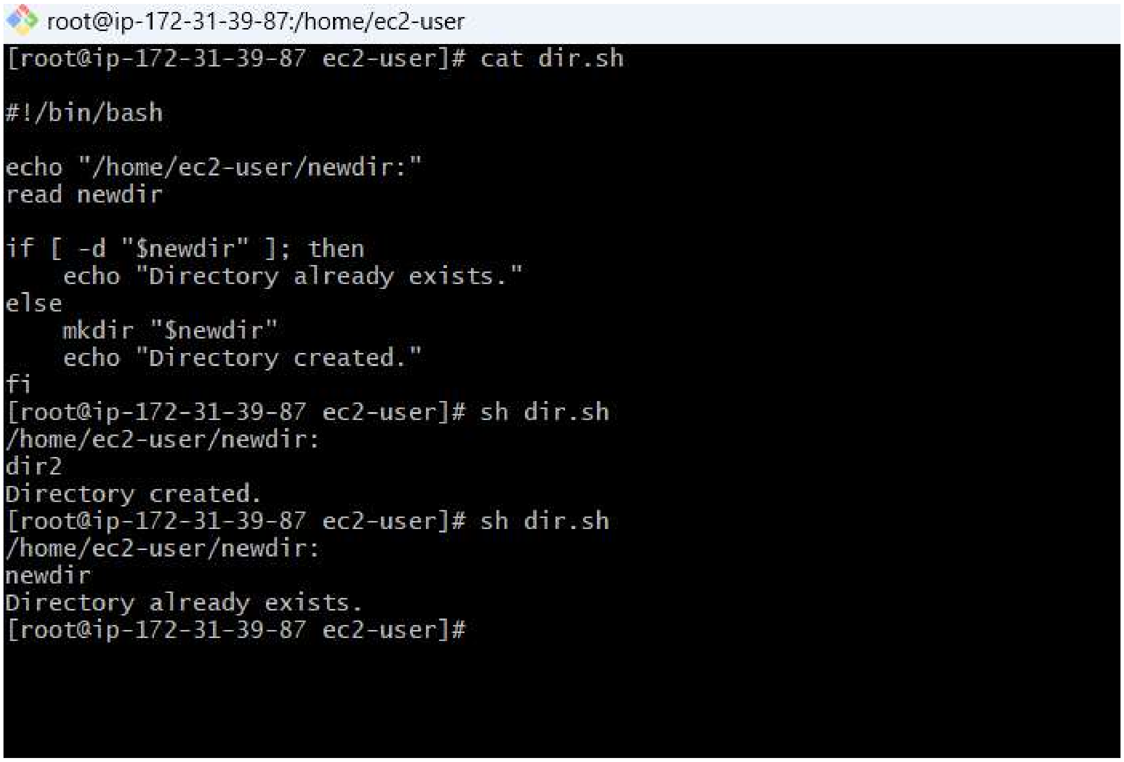
echo "Addition: $((a + b))"

echo "Subtraction: $((a - b))"

echo "Multiplication: $((a * b))"

echo "Division: $((a / b))"

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



root@ip-172-31-39-87:/home/ec2-user

```
[root@ip-172-31-39-87 ec2-user]# cat dir.sh

#!/bin/bash

echo "/home/ec2-user/newdir:"
read newdir

if [ -d "$newdir" ]; then
    echo "Directory already exists."
else
    mkdir "$newdir"
    echo "Directory created."
fi
[root@ip-172-31-39-87 ec2-user]# sh dir.sh
/home/ec2-user/newdir:
dir2
Directory created.
[root@ip-172-31-39-87 ec2-user]# sh dir.sh
/home/ec2-user/newdir:
newdir
Directory already exists.
[root@ip-172-31-39-87 ec2-user]#
```

#!/bin/bash

echo "Enter directory path:"

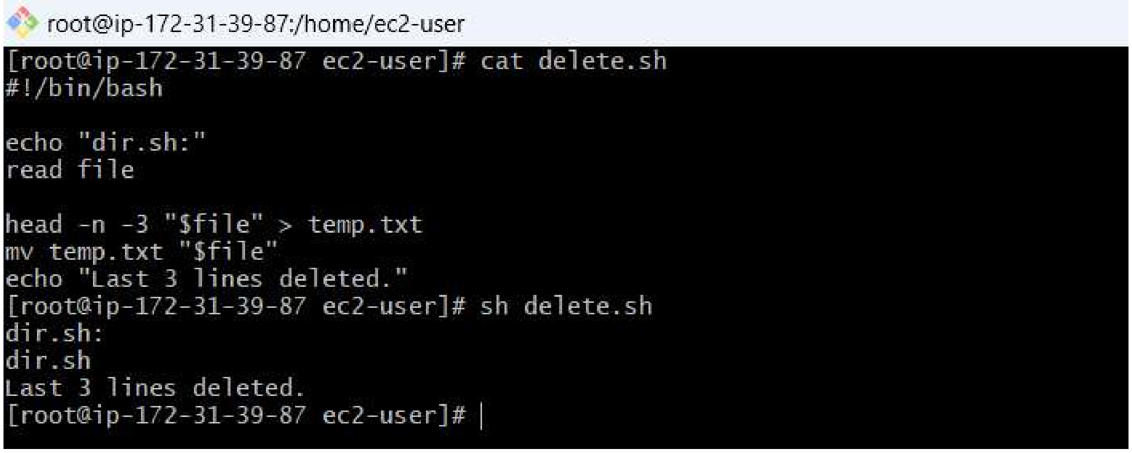
read dir

if [-d "$dir"]; then

    echo "Directory already exists."

```
else

    mkdir "$dir"

    echo "Directory created."

fi
```

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

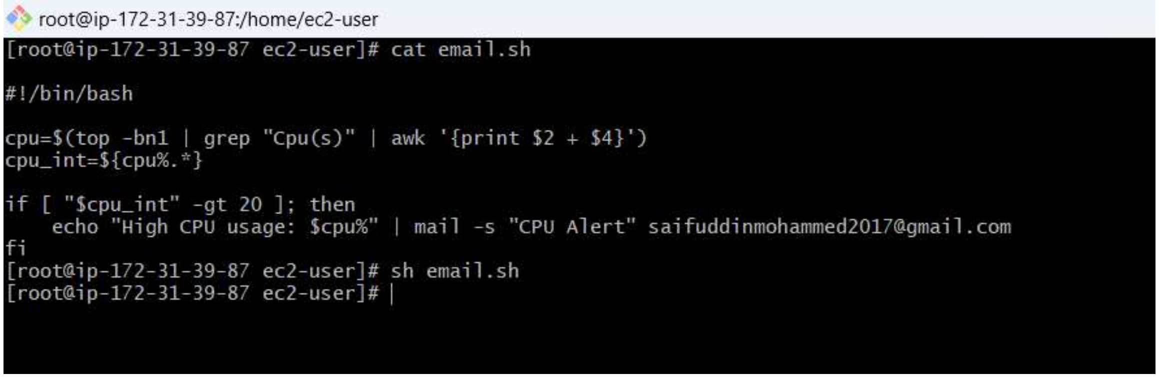


root@ip-172-31-39-87:/home/ec2-user

```
[root@ip-172-31-39-87 ec2-user]# cat delete.sh
#!/bin/bash

echo "dir.sh:"
read file

head -n -3 "$file" > temp.txt
mv temp.txt "$file"
echo "Last 3 lines deleted."
[root@ip-172-31-39-87 ec2-user]# sh delete.sh
dir.sh:
dir.sh
Last 3 lines deleted.
[root@ip-172-31-39-87 ec2-user]# |
```

```
#!/bin/bash


echo "Enter file name:"

read file


head -n -3 "$file" > temp.txt

mv temp.txt "$file"

echo "Last 3 lines deleted."
```

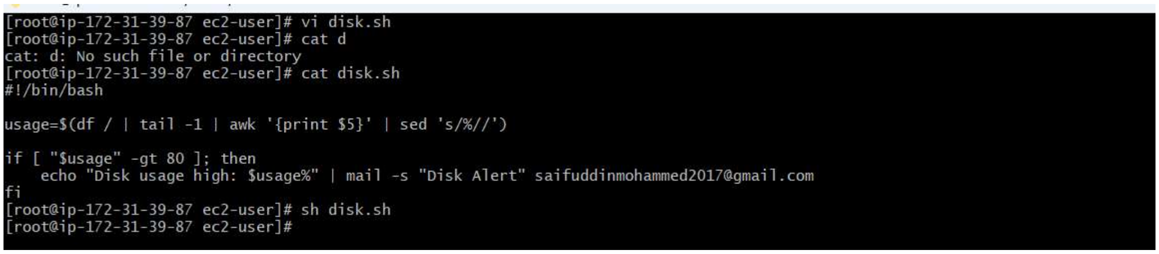
10) Bash script to monitor cpu and if it is more than 80% then send email notification.



```
[root@ip-172-31-39-87 ec2-user]# cat email.sh

#!/bin/bash

cpu=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}')
cpu_int=${cpu%.*}

if [ "$cpu_int" -gt 20 ]; then
    echo "High CPU usage: $cpu%" | mail -s "CPU Alert" saifuddinmohammed2017@gmail.com
fi
[root@ip-172-31-39-87 ec2-user]# sh email.sh
[root@ip-172-31-39-87 ec2-user]# |
```

```
#!/bin/bash



cpu=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}')

cpu_int=${cpu%.*}



if [ "$cpu_int" -gt 80 ]; then

    echo "High CPU usage: $cpu%" | mail -s "CPU Alert" user saifuddinmohammed2017.com

fi
```

11) Bash script to monitor disk space and if it is more than 80% then send email notification.

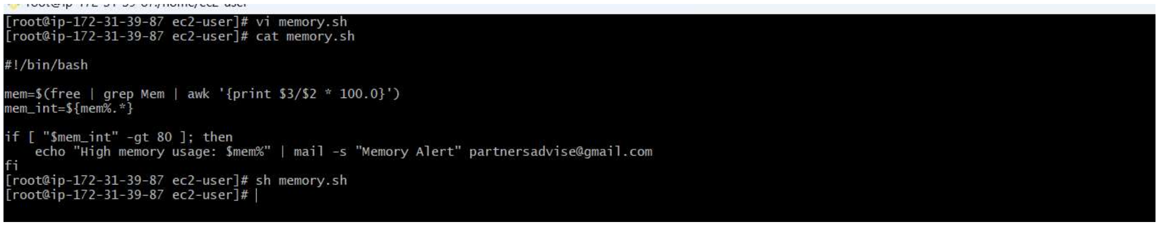


```
[root@ip-172-31-39-87 ec2-user]# vi disk.sh
[root@ip-172-31-39-87 ec2-user]# cat d
cat: d: No such file or directory
[root@ip-172-31-39-87 ec2-user]# cat disk.sh
#!/bin/bash

usage=$(df / | tail -1 | awk '{print $5}' | sed 's/%//')

if [ "$usage" -gt 80 ]; then
    echo "Disk usage high: $usage%" | mail -s "Disk Alert" saifuddinmohammed2017@gmail.com
fi
[root@ip-172-31-39-87 ec2-user]# sh disk.sh
[root@ip-172-31-39-87 ec2-user]#
```

```
#!/bin/bash



usage=$(df / | tail -1 | awk '{print $5}' | sed 's/%//')
```

```
if [ "$usage" -gt 80 ]; then
    echo "Disk usage high: $usage%" | mail -s "Disk Alert" saifuddinmohammed2017.com
fi
```

12) Bash script to monitor memory and if it is more than 80% then send email notification.



```
[root@ip-172-31-39-87 ec2-user]# vi memory.sh
[root@ip-172-31-39-87 ec2-user]# cat memory.sh

#!/bin/bash

mem=$(free | grep Mem | awk '{print $3/$2 * 100.0}')
mem_int=${mem%.*}

if [ "$mem_int" -gt 80 ]; then
    echo "High memory usage: $mem%" | mail -s "Memory Alert" partnersadvise@gmail.com
fi
[root@ip-172-31-39-87 ec2-user]# sh memory.sh
[root@ip-172-31-39-87 ec2-user]# |
```

```
#!/bin/bash


mem=$(free | grep Mem | awk '{print $3/$2 * 100.0}')

mem_int=${mem%.*}


if [ "$mem_int" -gt 80 ]; then

    echo "High memory usage: $mem%" | mail -s "Memory Alert" partnersadvise@gmail.com

fi
```