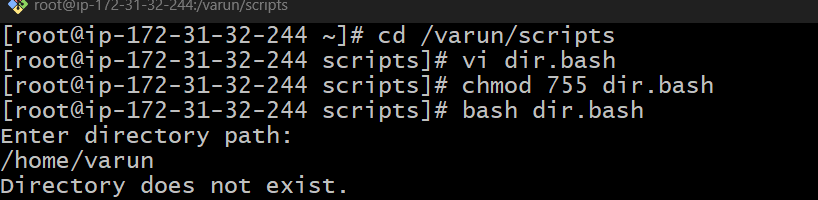
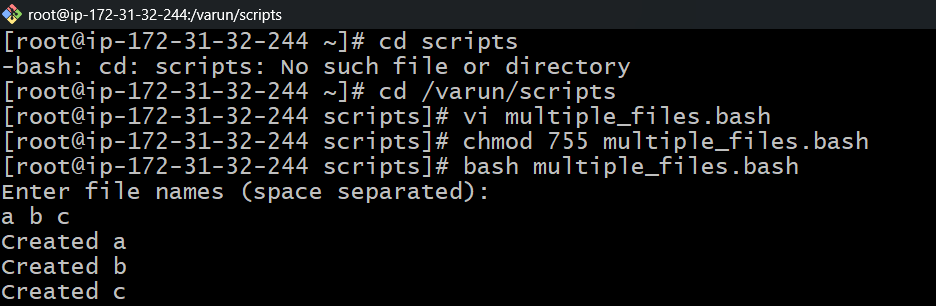
1. **Create a Bash script to check if a directory is available or not.**

****

#!/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 that will create mul�ple files?**

****

#!/bin/bash

echo "Enter file names (space separated):"

read files

for file in $files

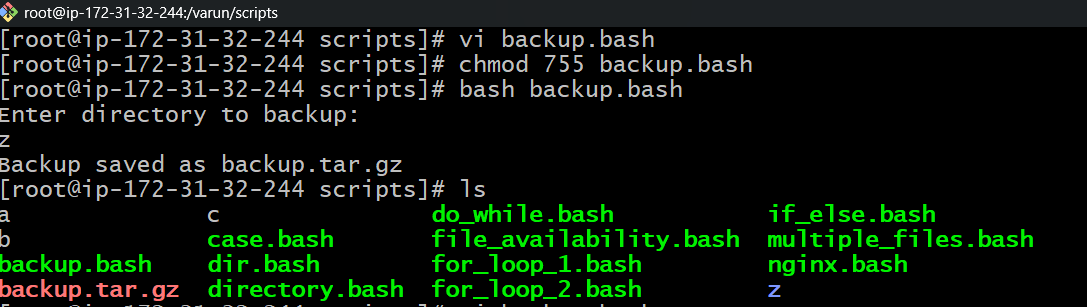
do

touch "$file"

echo "Created $file"

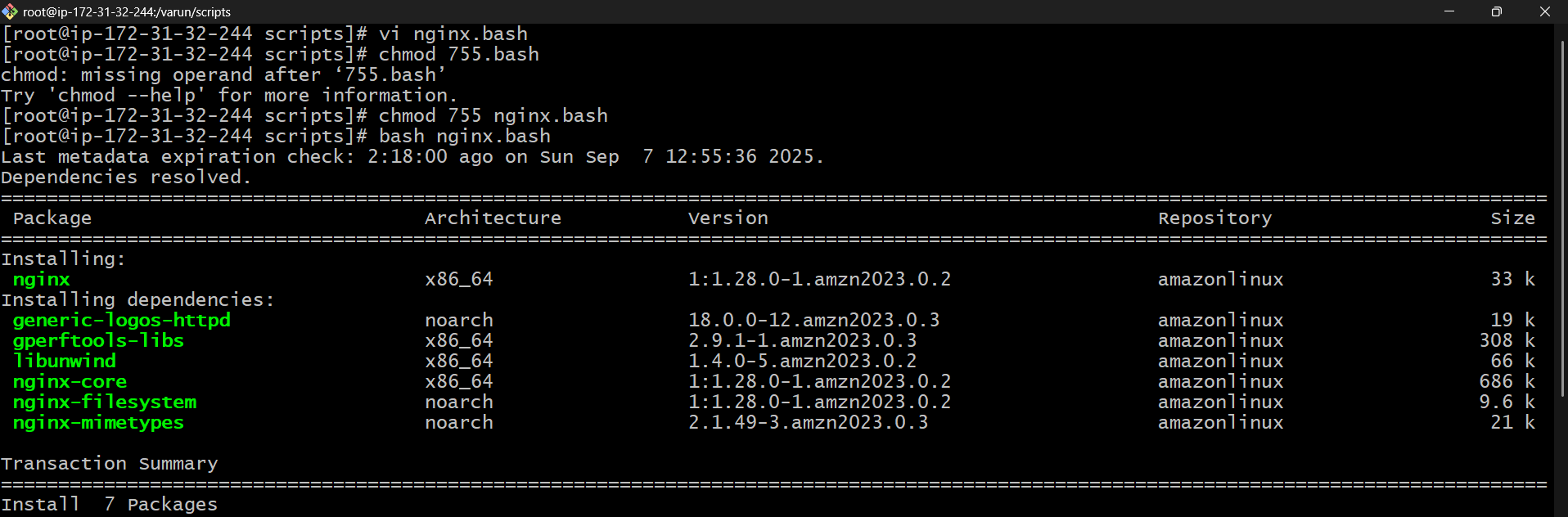
done

1. **Create a bash script to take a backup of a directory.**

****

#!/bin/bash   
echo "Enter directory to backup:"   
read dir   
 2  
tar -czf backup.tar.gz "$dir"   
echo "Backup saved as backup.tar.gz"

**4) Create a bash script to install nginx on an EC2 server.**

****

#!/bin/bash

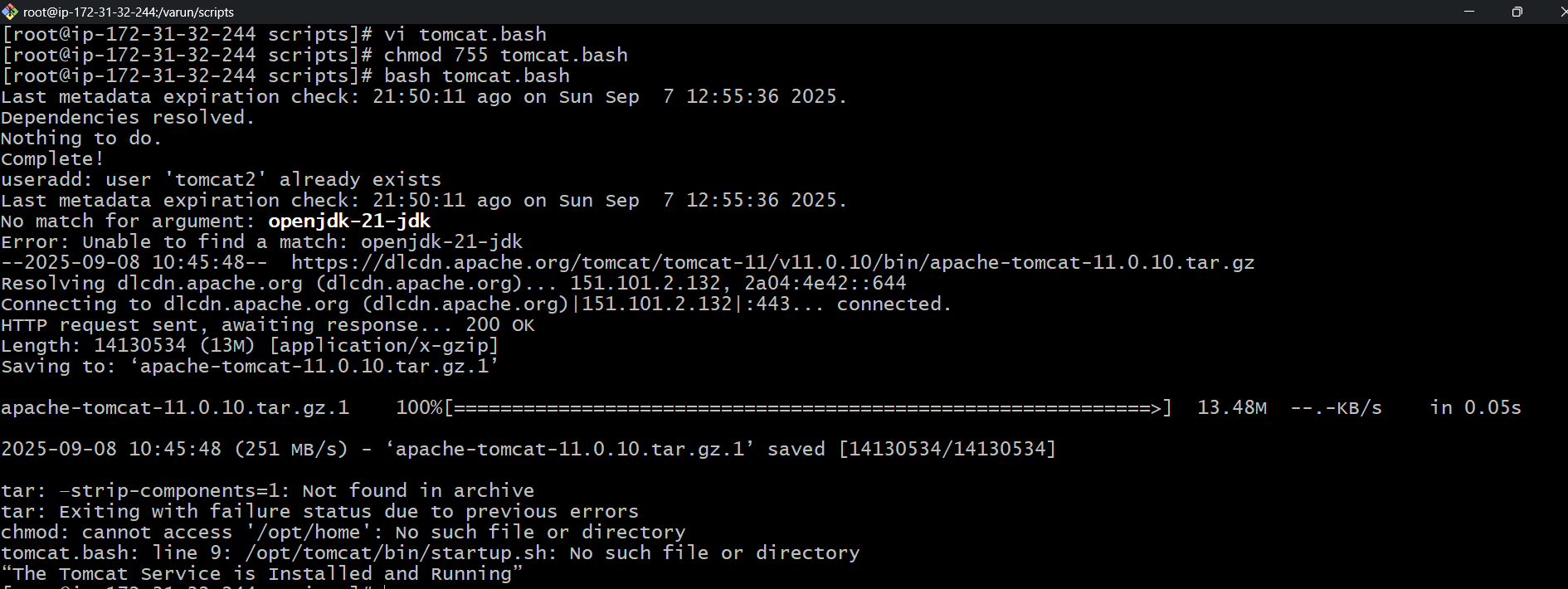
sudo yum install -y nginx

sudo systemctl start nginx

sudo systemctl enable nginx

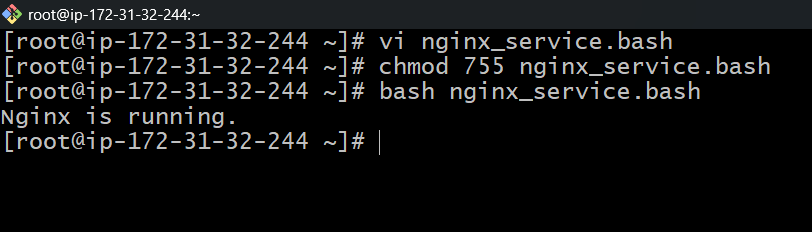
sudo systemctl status nginx

**5) Create a bash script to install Apache Tomcat on an EC2 server.**

****

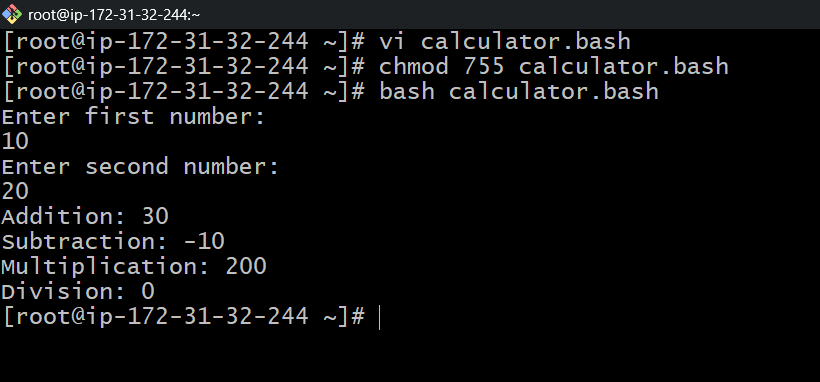
#!/bin/bash   
yum update   
useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat2   
yum install openjdk-21-jdk -y   
wget https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.10/bin/apache-tomcat-11.0.10.tar.gz   
tar xvf apache-tomcat-11.0.10.tar.gz -C /opt/tomcat –strip-components=1   
chown -R tomcat2: /opt/tomcat   
chmod -R 700 /opt/home   
/opt/tomcat/bin/startup.sh   
echo “The Tomcat Service is Installed and Running”

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

****

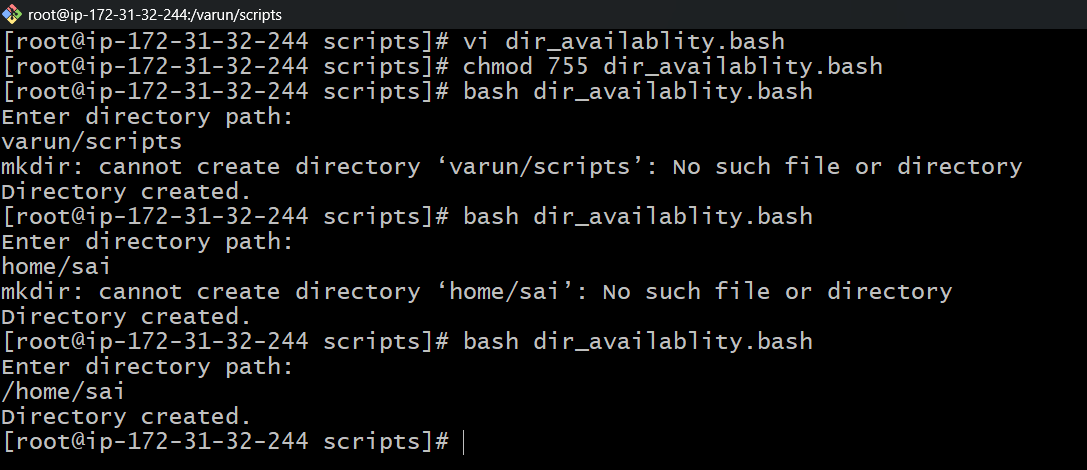
#!/bin/bash   
   
if systemctl is-active --quiet nginx; then   
 echo "Nginx is running."   
else   
 echo "Nginx is not running. Starting..."   
 sudo systemctl start nginx   
fi

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

****

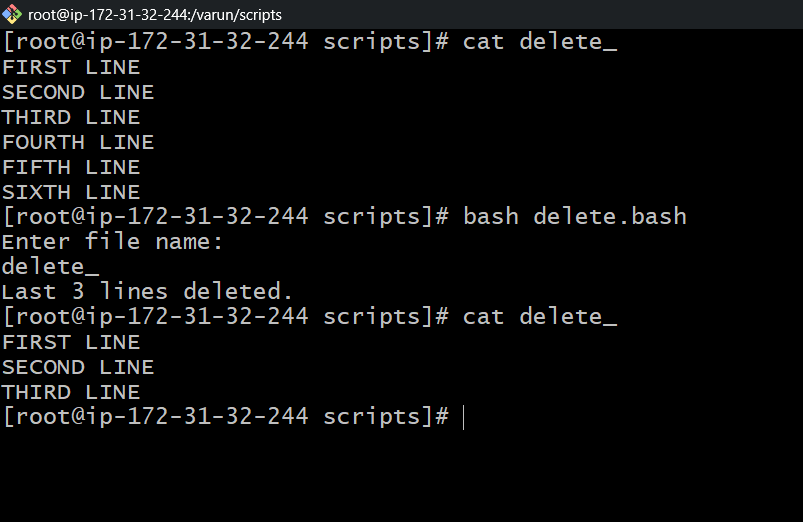
#!/bin/bash   
   
echo "Enter first number:"   
read a   
echo "Enter second number:"   
read b   
   
echo "Addition: $((a + b))"   
echo "Subtraction: $((a - b))"   
echo "Multiplication: $((a \* b))"   
echo "Division: $((a / b))

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



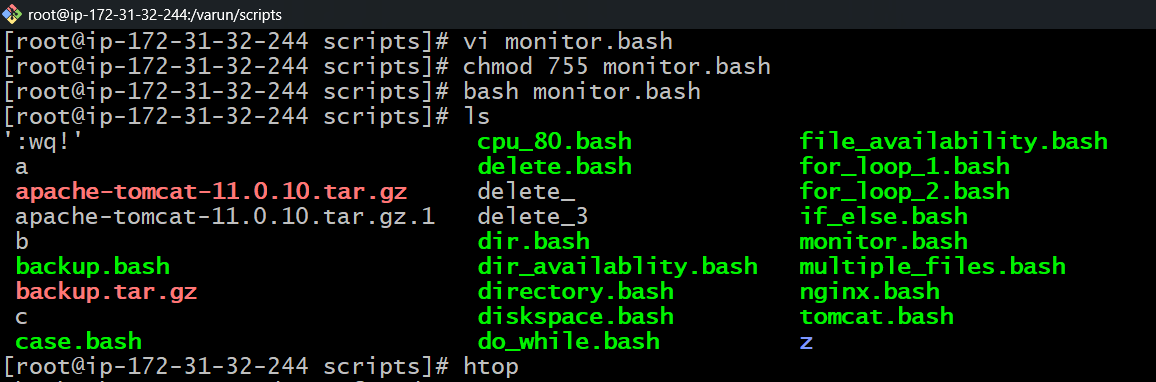
#!/bin/bash   
   
echo "Enter directory path:"   
read dir   
   
if [ -d "$dir" ]; then   
 echo "Directory already exists."   
else   
 mkdir "$dir"   
 echo "Directory created."   
fi

**9) Create a bash script to delete the last 3 lines of a file**

****

#!/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. If the usage is more than 80%, then send an email notification.**

****

**#!/bin/bash**

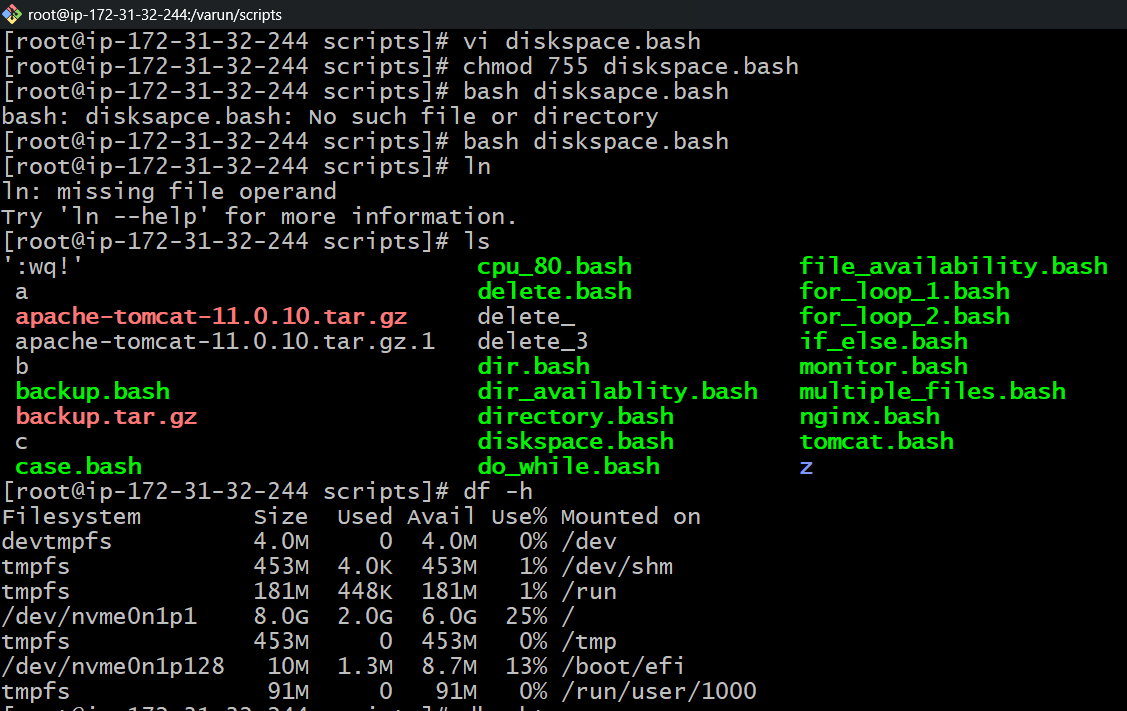
**usage=$(df / | tail -1 | awk '{print $5}' | sed 's/%//')**

**if [ "$usage" -gt 80 ]; then**

**echo "Disk usage high: $usage%”**

**fi**

**11) Bash script to monitor disk space, and if it is more than 80%,**

****

**#!/bin/bash**

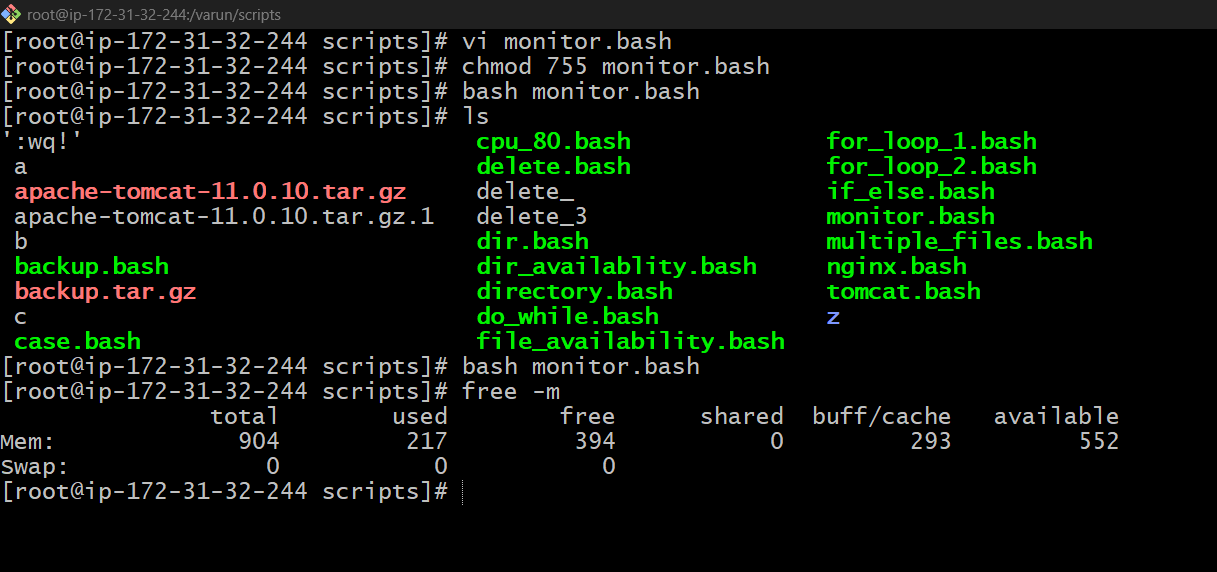
**usage=$(df / | tail -1 | awk '{print $5}' | sed 's/%//')**

**if [ "$usage" -gt 80 ]; then**

**echo "Disk usage high: $usage%”**

**fi**

**12) Bash script to monitor memory, and if it is more than 80%**

****

**#!/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%"**

**fi**