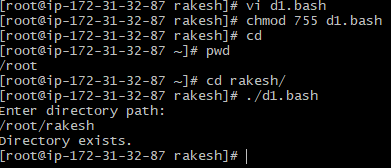
**BASH SCRIPTING – 02**

**NAME: RAKESH – 15**

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

* Create a directory
* Create a bash script to check the directory exist
* Give the permission for bash script
* Run the bashscript



#!/bin/bash

echo "Enter directory path:"

read dir

if [ -d "$dir" ]; then

echo "Directory exists."

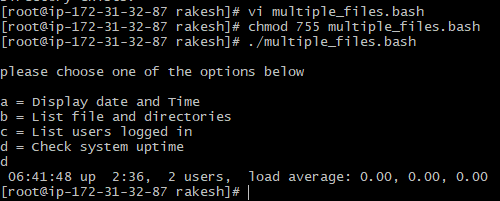
else

echo "Directory does not exist."

Fi

1. **Create a bash script to create multiple files**

* Create a bash script of multiple files
* Give the permission chmod 755 f1.bash
* Run the bah script

****

#!/bin/bash

echo

echo please choose one of the options below

echo

echo 'a = Display date and Time'

echo 'b = List file and directories'

echo 'c = List users logged in '

echo 'd = Check system uptime'

read choices

case $choices in

a) date;;

b) ls;;

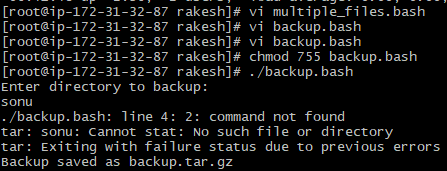
c) who;;

d) uptime;;

\*)echo invalid choice - Bye

esac

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



#!/bin/bash

echo "Enter directory to backup:"

read dir

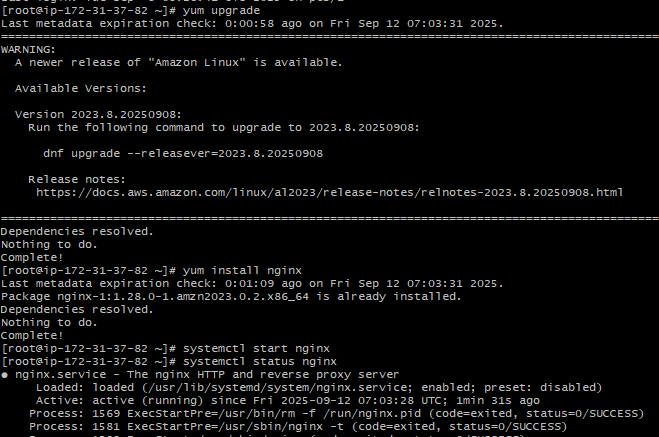
sonu

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

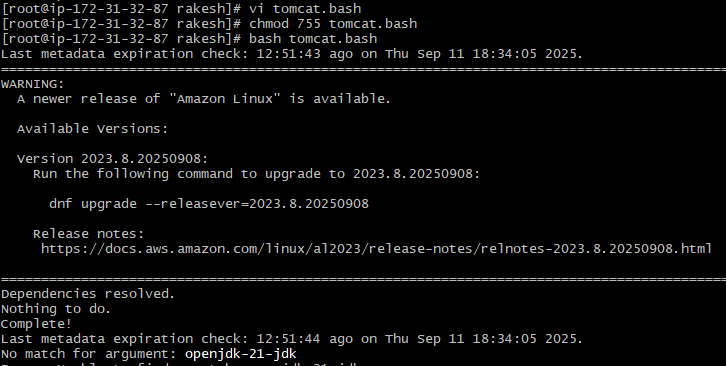
echo "Backup saved as backup.tar.gz"

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

* Create a Nginx file in EC2 server
* Yum upgrade
* Yum install nginx
* Systemctl start nginx
* Systemctl status nginx

****

1. **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

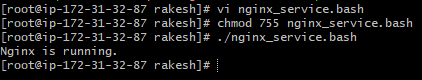
chown -R tomcat2: /opt/tomcat

chmod -R 700 /opt/home

/opt/tomcat/bin/startup.sh

echo “The Tomcat Service is Installed and Running”

1. **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."

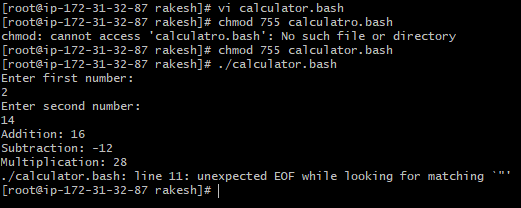
else

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

sudo systemctl start nginx

fi

1. **Create a bash script for a calculator.**

****

#!/bin/bash

echo "Enter first number:"

read a

echo "Enter second number:"

read b

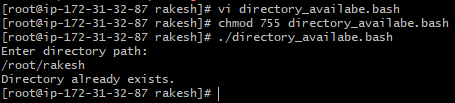
echo "Addition: $((a + b))"

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

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

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

1. **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."

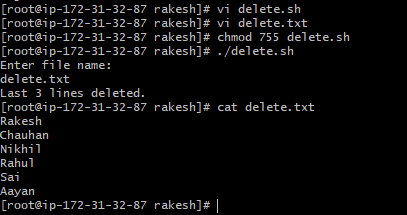
else

mkdir "$dir"

echo "Directory created."

fi

1. **Create a bash script to delete the last 3 lines of a file**



#!/bin/bash

echo "Enter file name:"

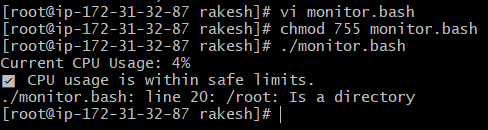
read file

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

mv temp.txt "$file"

echo "Last 3 lines deleted."

1. **Bash script to monitor CPU. If the usage is more than 80%, then send an email notification**



#!/bin/bash

# Set the threshold

THRESHOLD=80

# Get the CPU usage using top command and extract the idle percentage

CPU\_IDLE=$(top -bn1 | grep "Cpu(s)" | awk '{print $8}' | cut -d'.' -f1)

# Calculate actual CPU usage

CPU\_USAGE=$((100 - CPU\_IDLE))

echo "Current CPU Usage: $CPU\_USAGE%"

# Check if CPU usage exceeds threshold

if [ "$CPU\_USAGE" -gt "$THRESHOLD" ]; then

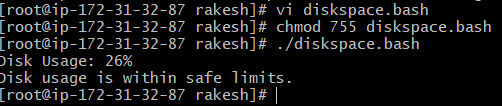
echo "⚠️ Warning: CPU usage is above ${THRESHOLD}%!"

else

echo "✅ CPU usage is within safe limits."

fi

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

****

#!/bin/bash

THRESHOLD=80

# Get disk usage percentage of root partition

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

echo "Disk Usage: $USAGE%"

if [ "$USAGE" -gt "$THRESHOLD" ]; then

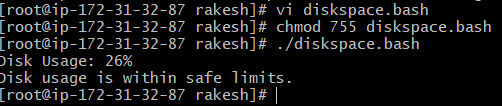
echo "Disk usage is above ${THRESHOLD}%!"

else

echo "Disk usage is within safe limits."

fi

1. **Bash script to monitor memory, and if it is more than 80%**

****

#!/bin/bash

THRESHOLD=80

# Get disk usage percentage of root partition

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

echo "Disk Usage: $USAGE%"

if [ "$USAGE" -gt "$THRESHOLD" ]; then

echo "Disk usage is above ${THRESHOLD}%!"

else

echo "Disk usage is within safe limits."

fi