Bash script assignment02

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

#!/bin/bash

# Set the directory path to check

DIR="/path/to/your/directory"

# Check if the directory exists

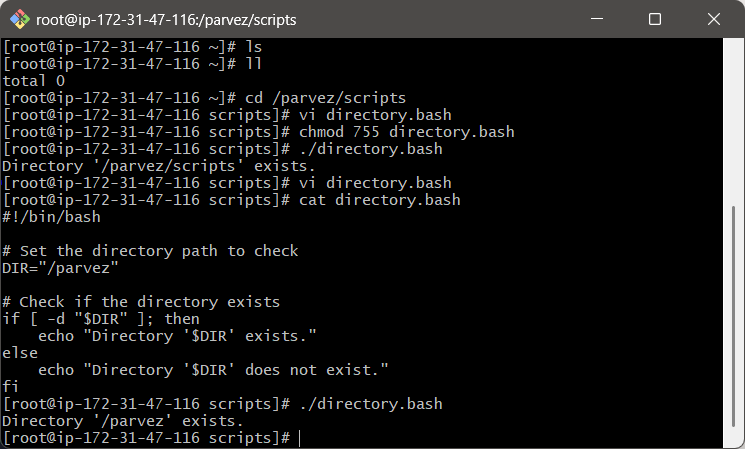
if [ -d "$DIR" ]; then

echo "Directory '$DIR' exists."

else

echo "Directory '$DIR' does not exist."

fi



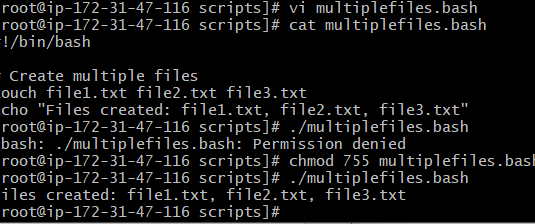
1. Create a bash script to create multiple files.

#!/bin/bash

# Create multiple files

touch file1.txt file2.txt file3.txt

echo "Files created: file1.txt, file2.txt, file3.txt"



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

#!/bin/bash

NOW=$(date +"%a")

case $NOW in

Mon)

echo "Full backup";;

Tue|Wed|Thu|Fri)

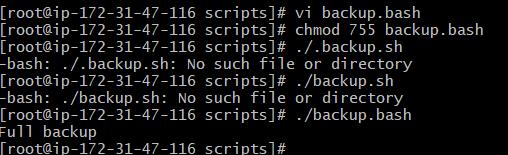
echo "Partial backup";;

Sat|Sun)

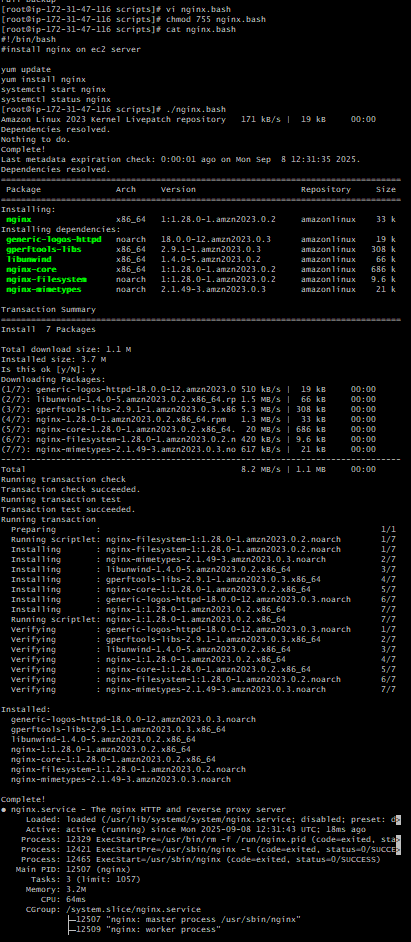
echo "No backup";;

\*) ;;

Esac



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

#!/bin/bash

#install nginx on ec2 server

yum update

yum install nginx

systemctl start nginx

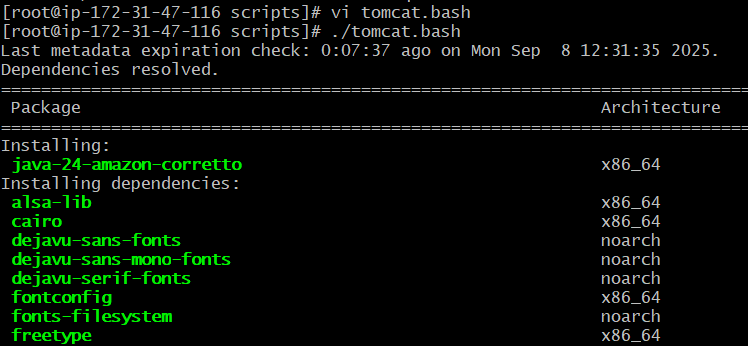
systemctl status nginx

1. Create a bash script to install Apache Tomcat on an EC2 server.

#!/bin/bash

yum install java

yum install tomcat



1. Create a bash script to check if the Nginx service is running, if not running then script should start the service.

#!/bin/bash

# check if nginx service is active

if systemctl is-active --quiet nginx; then

echo " nginx is running."

else

echo " nginx is not running. attempting to start it..."

sudo systemctl start nginx

# re-check status after attempting to start

if systemctl is-active --quiet nginx; then

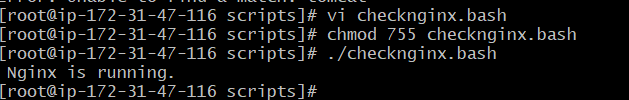
echo " nginx started successfully."

else

echo " failed to start nginx. check logs with: sudo journalctl -u nginx"

fi

fi



1. Create a bash script for a calculator.

#!/bin/bash

echo " simple bash calculator"

# read first number

read -p "enter first number: " num1

# read operator

read -p "enter operator (+, -, \*, /): " op

# read second number

read -p "enter second number: " num2

# perform calculation

case $op in

+) result=$(echo "$num1 + $num2" | bc);;

-) result=$(echo "$num1 - $num2" | bc);;

\\*) result=$(echo "$num1 \* $num2" | bc);;

/)

if [ "$num2" == "0" ]; then

echo " error: division by zero"

exit 1

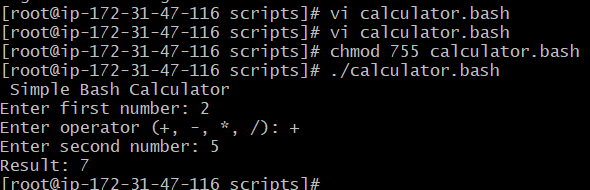
fi

result=$(echo "scale=2; $num1 / $num2" | bc);;

\*) echo " invalid operator"; exit 1;;

esac

echo "result: $result"



1. Create a bash script to check if a directory exists, if not then create a directory.

#!/bin/bash

# Set your target directory

DIR="./DIRECTORY”

# Check if the directory exists

if [ -d "$DIR" ]; then

echo "Directory '$DIR' already exists."

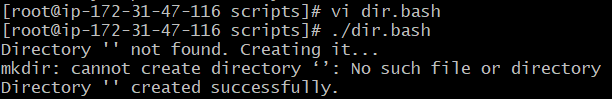
else

echo " Directory '$DIR' not found. Creating it..."

mkdir -p "$DIR"

echo " Directory '$DIR' created successfully."

Fi



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

#!/bin/bash

# Usage: delete last 3 lines of file

FILE="$1"

if [[ ! -f "$FILE" ]]; then

echo "Error: File '$FILE' not found."

exit 1

fi

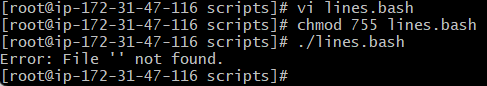
# Create a temporary file excluding the last 3 lines

head -n -3 "$FILE" > "${FILE}.tmp"

# Replace original file with trimmed version

mv "${FILE}.tmp" "$FILE"

echo "Removed last 3 lines from '$FILE'."



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

#!/bin/bash

# CPU threshold

THRESHOLD=80

# Get current CPU idle percentage using top

CPU\_IDLE=$(top -bn2 | grep "Cpu(s)" | tail -n1 | awk -F',' '{print $4}' | awk '{print $1}')

CPU\_USAGE=$(echo "100 - $CPU\_IDLE" | bc)

# Convert to integer

CPU\_INT=${CPU\_USAGE%.\*}

# Check threshold

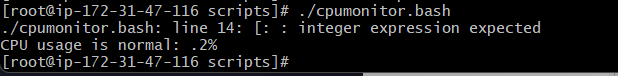
if [ "$CPU\_INT" -gt "$THRESHOLD" ]; then

echo "Warning: CPU usage is at ${CPU\_USAGE}% (Threshold: ${THRESHOLD}%)"

else

echo " CPU usage is normal: ${CPU\_USAGE}%"

fi



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

#!/bin/bash

# Threshold percentage

THRESHOLD=80

# Get disk usage info

df -H --output=pcent,target | tail -n +2 | while read -r line; do

# Extract percentage and mount point

USAGE=$(echo "$line" | awk '{print $1}' | tr -d '%')

MOUNT=$(echo "$line" | awk '{print $2}')

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

echo "Warning: Disk usage on $MOUNT is ${USAGE}% (Threshold: ${THRESHOLD}%)"

# Optional: trigger email or log here

fi

done



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

#!/bin/bash

# define the memory usage threshold (80%)

threshold=80

# get total memory and used memory in mb

# free -m outputs memory in mb

# awk is used to extract the relevant numbers from the 'mem:' line

total\_memory=$(free -m | awk 'nr==2 {print $2}')

used\_memory=$(free -m | awk 'nr==2 {print $3}')

# calculate memory usage percentage

# 'bc' is used for floating-point arithmetic

if [[ "$total\_memory" -gt 0 ]]; then

memory\_percentage=$(echo "scale=2; ($used\_memory \* 100) / $total\_memory" | bc)

else

memory\_percentage=0

fi

# compare memory usage with the threshold

# 'echo' is used with 'bc' to perform the comparison

# the result is then evaluated in an 'if' statement

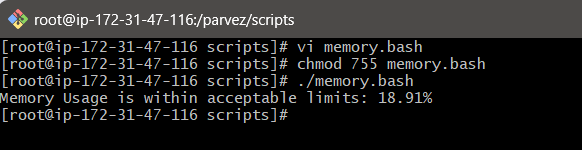
if (( $(echo "$memory\_percentage > $threshold" | bc -l) )); then

echo "warning: high memory usage!"

echo "current memory usage: ${memory\_percentage}% (${used\_memory}mb / ${total\_memory}mb)"

else

echo "memory usage is within acceptable limits: ${memory\_percentage}%"

fi