

# TASK ON BASH SCRIPT 02

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

```
#!/bin/bash
# Define the directory to check
DIR_PATH="/path/to/directory"

# Check if the directory exists
if [ -d "$DIR_PATH" ]; then
    echo "Directory '$DIR_PATH' exists."
else
    echo "Directory '$DIR_PATH' does not exist."
fi
```

```
[root@ip-172-31-2-53 ~]# sudo -i
[root@ip-172-31-2-53 ~]# vi dire1.bash
[root@ip-172-31-2-53 ~]# chmod 755 dire1.bash
[root@ip-172-31-2-53 ~]# ./dire1.bash
Directory '/path/to/directory' does not exist.
```

## 2) Create a bash script which will create multiple files:

```
#!/bin/bash
# Ask for the number of files to create
read -p "Enter the number of files to create: " FILE_COUNT
read -p "Enter the prefix for file names: " PREFIX

# Loop to create files
for ((i=1; i<=FILE_COUNT; i++))
do
    FILE_NAME="${PREFIX}${i}.txt"
    touch "$FILE_NAME"
    echo "File '$FILE_NAME' created."
done

echo "All files have been created successfully!"
```

```
[root@ip-172-31-2-53 ~]# vi dire1.bash
[root@ip-172-31-2-53 ~]# ./dire1.bash
Enter the number of files to create: 5
Enter the prefix for file names: f
File 'f1.txt' created.
File 'f2.txt' created.
File 'f3.txt' created.
File 'f4.txt' created.
File 'f5.txt' created.
All files have been created successfully!
[root@ip-172-31-2-53 ~]#
```

### 3) Create a bash script to take backup of a directory:

```
MINGW64:/c/Users/naren
#!/bin/bash

# Define source directory and backup destination
echo "Enter the directory to back up: "
read SOURCE_DIR
echo "Enter the backup destination directory: "
read BACKUP_DIR

# Create backup filename with timestamp
TIMESTAMP=$(date +%Y%m%d_%H%M%S")
BACKUP_FILE="backup_${TIMESTAMP}.tar.gz"

# Ensure backup directory exists
mkdir -p "$BACKUP_DIR"

# Create the backup
if tar -czf "$BACKUP_DIR/$BACKUP_FILE" "$SOURCE_DIR" 2>/dev/null; then
    echo "Backup successful: $BACKUP_DIR/$BACKUP_FILE"
else
    echo "Backup failed! Check the source directory and permissions."
    exit 1
fi

~
~
~
```

```
MINGW64:/c/Users/naren
$ vi dir1.bash

naren@narendar MINGW64 ~ (master)
$ chmod 577 d
Desktop/ Documents/ dir1.bash

naren@narendar MINGW64 ~ (master)
$ chmod 577 dir1.bash

naren@narendar MINGW64 ~ (master)
$ ./dir1.bash
Enter the directory to back up:
abc1
Enter the backup destination directory:

mkdir: cannot create directory '': No such file or directory
Backup failed! Check the source directory and permissions.
```

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

```
aws
Search [Alt+S]
Asia Pacific (Mumbai) narendar p

#!/bin/bash
yum update -y
sudo amazon-linux-extras install nginx1

echo " installing nginx "

systemctl start nginx

systemctl status nginx
~
~
```

```
aws [Search] [Alt+S] Asia Pacific (Mumbai) narendar p

61 dnsmasq2.85 available [ =stable ]
62 kernel-5.15 available [ =stable ]
63 postgresql14 available [ =stable ]
64 firefox available [ =stable ]
65 lustre available [ =stable ]
67 awscli1 available [ =stable ]
68 php8.2 available [ =stable ]
69 dnsmasq available [ =stable ]
70 unbound1.17 available [ =stable ]
72 collectd-python3 available [ =stable ]
# Note on end-of-support. Use 'info' subcommand.
# installing nginx
* nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2025-03-04 13:24:12 UTC; 8ms ago
   Process: 3849 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Process: 3846 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
   Process: 3845 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
   Main PID: 3852 (nginx)
   CGroup: /system.slice/nginx.service
           └─3852 nginx: master process /usr/sbin/nginx
             └─3854 nginx: worker process

Mar 04 13:24:12 ip-172-31-4-178.ap-south-1.compute.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Mar 04 13:24:12 ip-172-31-4-178.ap-south-1.compute.internal nginx[3846]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Mar 04 13:24:12 ip-172-31-4-178.ap-south-1.compute.internal nginx[3846]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Mar 04 13:24:12 ip-172-31-4-178.ap-south-1.compute.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
[root@ip-172-31-4-178 ~]#
```

## 5)Create a bash script to install Apache Tomcat in ec2 server:

```
aws [Search] [Alt+S] Asia Pacific (Mumbai) narendar p

#!/bin/bash
yum update -y

wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.100/bin/apache-tomcat-9.0.100.tar.gz

tar -xvzf apache-tomcat-9.0.100.tar.gz

./apache-tomcat-9.0.100/bin/startup.sh

aws [Search] [Alt+S] Asia Pacific (Mumbai) narendar p

Verifying : libxslt-1.1.28-6.amzn2.x86_64 22/28
Verifying : python-lxml-3.2.1-4.amzn2.0.6.x86_64 23/28
Verifying : python-javapackages-3.4.1-11.amzn2.noarch 24/28
Verifying : libxslt-1.2.3-1.amzn2.0.2.x86_64 25/28
Verifying : alsa-lib-1.1.4-1-2.amzn2.x86_64 26/28
Verifying : libICE-1.0.9-9.amzn2.0.2.x86_64 27/28
Verifying : javapackages-tools-3.4.1-11.amzn2.noarch 28/28

Installed:
  java-11-amazon-corretto.x86_64 1:11.0.26+4-1.amzn2

Dependency Installed:
  alsa-lib.x86_64 0:1.1.4-1-2.amzn2          dejavu-fonts-common.noarch 0:2.33-6.amzn2      dejavu-sans-fonts.noarch 0:2.33-6.amzn2
  fontpackages-filesystem.noarch 0:1.44-8.amzn2  dejavu-serif-fonts.noarch 0:2.33-6.amzn2      fontconfig.x86_64 0:2.13.0-4.3.amzn2
  javapackages-tools.noarch 0:3.4.1-11.amzn2  libICE.x86_64 0:1.0.9-9.amzn2.0.2             java-11-amazon-corretto-headless.x86_64 1:11.0.26+4-1.amzn2
  libX11.x86_64 0:1.6.7-3.amzn2.0.5          libXtst.x86_64 0:1.2.3-1.amzn2.0.2             libESM.x86_64 0:1.2.2-2.amzn2.0.2
  libXext.x86_64 0:1.3.3-3.amzn2.0.2          libXi.x86_64 0:1.7.9-1.amzn2.0.2             libXau.x86_64 0:1.0.8-2.1.amzn2.0.2
  libXrandr.x86_64 0:1.5.1-2.amzn2.0.3         libXrender.x86_64 0:0.9.10-1.amzn2.0.2         libXinerama.x86_64 0:1.1.3-2.1.amzn2.0.2
  libXtst.x86_64 0:1.2.3-1.amzn2.0.2          libxcb.x86_64 0:1.12-1.amzn2.0.2             libXt.x86_64 0:1.1.5-3.amzn2.0.2
  log4j-cve-2021-44228-hotpatch.noarch 0:1.3-7.amzn2  python-javapackages.noarch 0:3.4.1-11.amzn2  libxslt.x86_64 0:1.1.28-6.amzn2
                                                                                               python-lxml.x86_64 0:3.2.1-4.amzn2.0.6

Complete!
[root@ip-172-31-14-100 ~]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.100/bin/apache-tomcat-9.0.100.tar.gz
--2025-03-04 13:38:11-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.100/bin/apache-tomcat-9.0.100.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: 12780034 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.100.tar.gz'

100%[=====>] 12,780,034 --.-K/s in 0.1s
```

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

```
#!/bin/bash

systemctl is-active --quiet nginx

if [ $? = 0 ]
then
    echo "nginx is running"
else
    echo "nginx is not running"
    echo "nginx is starting"
    systemctl start nginx
    systemctl status nginx
fi
```

```
apache-tomcat-9.0.100/bin/shutdown.sh
apache-tomcat-9.0.100/bin/startup.sh
apache-tomcat-9.0.100/bin/tool-wrapper.sh
apache-tomcat-9.0.100/bin/version.sh
Neither the JAVA_HOME nor the JRE_HOME environment variable is defined
At least one of these environment variable is needed to run this program
[root@ip-172-31-4-178 ~]# vi script.bash
[root@ip-172-31-4-178 ~]# chmod 755 script.bash
[root@ip-172-31-4-178 ~]# ./script.bash
systemctl: unrecognized option '--quiet'
nginx is not running
nginx is starting
* nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2025-03-04 13:24:12 UTC; 59min ago
   Process: 3849 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Process: 3846 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
   Process: 3845 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
   Main PID: 3852 (nginx)
   CGroup: /system.slice/nginx.service
           └─3852 nginx: master process /usr/sbin/nginx
             └─3854 nginx: worker process

Mar 04 13:24:12 ip-172-31-4-178.ap-south-1.compute.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Mar 04 13:24:12 ip-172-31-4-178.ap-south-1.compute.internal nginx[3846]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Mar 04 13:24:12 ip-172-31-4-178.ap-south-1.compute.internal nginx[3846]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Mar 04 13:24:12 ip-172-31-4-178.ap-south-1.compute.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
[root@ip-172-31-4-178 ~]#
```

7) Create a bash script for calculator:

```
[root@ip-172-31-4-178 ~]# cat script.bash
#!/bin/bash

echo "Simple Bash Calculator"

echo "Enter first number: "
read num1

echo "Enter second number: "
read num2

echo "Select operation:"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
read choice

case $choice in
    1) result=$((num1 + num2))
       echo "Result: $num1 + $num2 = $result";;
    2) result=$((num1 - num2))
       echo "Result: $num1 - $num2 = $result";;
    3) result=$((num1 * num2))
       echo "Result: $num1 * $num2 = $result";;
    4) if [ $num2 -ne 0 ]; then
        result=$((echo "scale=2; $num1 / $num2" | bc))
        echo "Result: $num1 / $num2 = $result"
      else
        echo "Error: Division by zero is not allowed."
      fi;;
    *) echo "Invalid choice";;
esac
```

```

[root@ip-172-31-4-178 ~]# vi script.bash
[root@ip-172-31-4-178 ~]# chmod 755 script.bash
[root@ip-172-31-4-178 ~]# ./script.bash
Simple Bash Calculator
Enter first number:
8
Enter second number:
5
Select operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
1
Result: 8 + 5 = 13
[root@ip-172-31-4-178 ~]# 4
-bash: 4: command not found
[root@ip-172-31-4-178 ~]# vi script.bash
[root@ip-172-31-4-178 ~]# █

```

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



```

aws | Search [Alt+S]
~
echo "Enter directory name: "
read dir_name

# Check if directory exists
if [ -d "$dir_name" ]; then
    echo "Directory '$dir_name' already exists."
else
    # Create directory
    mkdir "$dir_name"
    echo "Directory '$dir_name' has been created."
fi

~
~
~
~
~

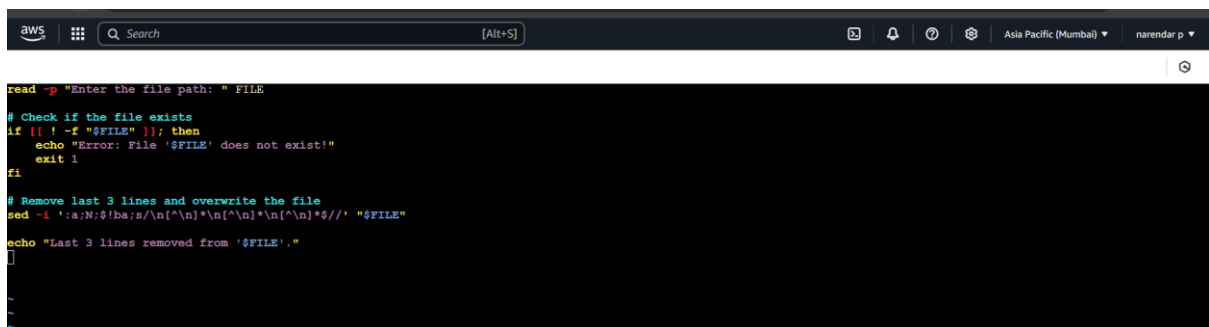
```

```

[root@ip-172-31-4-178 ~]# vi script.bash
[root@ip-172-31-4-178 ~]# chmod 755 script.bash
[root@ip-172-31-4-178 ~]# ./script.bash
Enter directory name:
script.bash
mkdir: cannot create directory 'script.bash': File exists
Directory 'script.bash' has been created.
[root@ip-172-31-4-178 ~]# ls
apache-tomcat-9.0.100  apache-tomcat-9.0.100.tar.gz  script.bash
[root@ip-172-31-4-178 ~]# ./script.bash
Enter directory name:
apache-tomcat-9.0.100
Directory 'apache-tomcat-9.0.100' already exists.
[root@ip-172-31-4-178 ~]# █

```

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



```

aws | Search [Alt+S]
~
read -p "Enter the file path: " FILE

# Check if the file exists
if [ ! -f "$FILE" ]; then
    echo "Error: File '$FILE' does not exist!"
    exit 1
fi

# Remove last 3 lines and overwrite the file
sed -i ':a;N;$!ba;s/\n[^\\n]*\\n[^\\n]*\\n[^\\n]*$//' "$FILE"

echo "Last 3 lines removed from '$FILE'."

~
~
~
~
~

```

```
[root@ip-172-31-84-72 Bash_tasks]# ./deleteLast#lines.bash
give me the file name to delete last three lines
mynumbers.txt
1
2
3
4
5
6
7
8
9
0
2
3
4
5
6
7
removing last 3 lines
...
removed last three lines in the given file
1
2
3
4
5
6
7
8
9
0
2
3
4
[root@ip-172-31-84-72 Bash_tasks]# |
```

## Cron job

```
# april 5th midnight (/opt/myScripts/Bash_tasks)
0 0 5 4 * /opt/myScripts/Bash_tasks/checkDirExist.bash

#5th of every november,jan,june if it is a thursday
0 0 5 1,6,11 4 /opt/myScripts/Bash_tasks/createLast#lines.bash

#at 5th and 27th minute of 9,10,11 hrs every day
5,27 9,10,11 * * * /opt/myScripts/Bash_tasks/calculator.bash

#34 min of 9th hour on 15th august
34 9 15 8 * //opt/myScripts/Bash_tasks/createBackup.bash

#Every Midnight
0 0 * * * /opt/myScripts/Bash_tasks/myFile.bash

#every weekend
59 23 * * * /opt/myScripts/Bash_tasks/file2.bash

#on every reboot
@reboot /opt/myScripts/Bash_tasks/calculator|
~
~
~
```