

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

```
[ironman@awslinux ~]$ cat testscripts/check-script
#!/bin/bash

if [ -e /home/ironman ]
then
echo "It exist"
else
echo "It doesn't exist"
fi
[ironman@awslinux ~]$ bash /home/ironman/testscripts/check-script
It exist
[ironman@awslinux ~]$ |
```

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

```
[ironman@awslinux ~]$ vi testscripts/files.sh
[ironman@awslinux ~]$ cat testscripts/files.sh
#!/bin/bash

DIR="/home/ironman/"

for i in {1..5}; do
    touch "$DIR/file$i"
done

[ironman@awslinux ~]$ ls
'\      testscripts
[ironman@awslinux ~]$ bash testscripts/files.sh
[ironman@awslinux ~]$ ls -l
total 20
-rw-r--r--. 1 ironman superheroes    88 Nov  4 06:45 '\
-rw-r--r--. 1 ironman superheroes     0 Nov  4 06:51 file1
-rw-r--r--. 1 ironman superheroes     0 Nov  4 06:51 file2
-rw-r--r--. 1 ironman superheroes     0 Nov  4 06:51 file3
-rw-r--r--. 1 ironman superheroes     0 Nov  4 06:51 file4
-rw-r--r--. 1 ironman superheroes     0 Nov  4 06:51 file5
drwxr-xr-x. 2 ironman superheroes 16384 Nov  4 06:51 testscripts
```

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

```
[ironman@awslinux ~]$ vi testscripts/script_backup.sh
[ironman@awslinux ~]$ chmod a+x testscripts/script_backup.sh
[ironman@awslinux ~]$ cat testscripts/script_backup.sh
#!/bin/bash

source="/home/ironman/testscripts"
backup="/home/ironman/script_backup"

#Creating new directory

mkdir -p "$backup"

#Copying all content in backup DIR

cp -r "$source" "$backup"

#Give output

echo "content of '$source' have been copied to '$backup'."

[ironman@awslinux ~]$ ls
testscripts
[ironman@awslinux ~]$ bash testscripts/script_backup.sh
content of '/home/ironman/testscripts' have been copied to '/home/ironman/script_backup'.
[ironman@awslinux ~]$ ls
script_backup testscripts
```

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

```
[batman@ip-172-31-2-182 ~]$ cat test_script/nginx.sh
#!/bin/bash

#updating system

sudo yum update systemd -y

#installing the web server nginx

sudo yum install nginx -y
echo

#checking status of the nginx service
echo "status----->>>>>>>"
sudo systemctl status nginx.service | grep "Active"

sleep 2

#starting and enabling the service
sudo systemctl start nginx.service
echo "Starting nginx.service...."

#checking the state of nginx service
sleep 1
sudo systemctl status nginx.service | grep "Active"
echo

echo "#####"
```

```
Installed:
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64      libunwind-1.4.0-5.amzn2023.0.2.x86_64
  nginx-1:1.24.0-1.amzn2023.0.4.x86_64                  nginx-core-1:1.24.0-1.amzn2023.0.4.x86_64         nginx-filesystem-1:1.24.0-1.amzn2023.0.4.noarch
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!

status----->>>>>>>
  Active: inactive (dead)
Starting nginx.service....
  Active: active (running) since Mon 2024-11-04 10:04:02 UTC; 1s ago

#####
[batman@ip-172-31-2-182 ~]$
```

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

```
[batman@ip-172-31-2-182 test_script]$ vi tomcat.sh
[batman@ip-172-31-2-182 test_script]$ cat tomcat.sh
#!/bin/bash
#updating system daemon

sudo yum update systemd

#installing java development kit for running apache-tomcat
echo "Installing Java jdk.....>"
echo
sudo yum install java -y

#Downloading latest version of tomcat
wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.31/bin/apache-tomcat-10.1.31.tar.gz

#Moving file to different location and untar the file
sudo mkdir -p /opt/tomcat
sudo mv apache-tomcat-10.1.31.tar.gz /opt/tomcat
cd /opt/tomcat
sudo tar xvf apache-tomcat-10.1.31.tar.gz

#changing ownership for the file
sudo chown -R $USER:$USER /opt/tomcat/apache-tomcat-10.1.31

#starting tomcat
bash /opt/tomcat/apache-tomcat-10.1.31/bin/startup.sh

[batman@ip-172-31-2-182 test_script]$ chmod a+x tomcat.sh
[batman@ip-172-31-2-182 test_script]$ ./tomcat.sh
```

```

Installed:
alsa-lib-1.2.7.2-1.amzn2023.0.2.x86_64
dejavu-sans-mono-fonts-2.37-16.amzn2023.0.2.noarch
fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch
google-noto-fonts-common-202106-2.amzn2023.0.2.noarch
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
javapackages-filesystem-6.0.0-7.amzn2023.0.6.noarch
libSM-1.2.4-3.amzn2023.0.1.x86_64
libXau-1.0.11-6.amzn2023.0.1.x86_64
libXinerama-1.1.5-6.amzn2023.0.1.x86_64
libXt-1.3.0-3.amzn2023.0.1.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
pixman-0.43.4-1.amzn2023.0.4.x86_64
cairo-1.17.6-2.amzn2023.0.1.x86_64
dejavu-serif-fonts-2.37-16.amzn2023.0.2.noarch
freetype-2.13.2-5.amzn2023.0.1.x86_64
google-noto-sans-vf-fonts-202106-2.amzn2023.0.2.noarch
java-22-amazon-corretto-1:22.0.2+9-1.amzn2023.1.x86_64
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libX11-1.8.10-2.amzn2023.0.1.x86_64
libXext-1.3.6-1.amzn2023.0.1.x86_64
libXrandr-1.5.4-3.amzn2023.0.1.x86_64
libXtst-1.2.5-1.amzn2023.0.1.x86_64
libpng-2:1.6.37-10.amzn2023.0.6.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
dejavu-sans-fonts-2.37-16.amzn2023.0.2.noarch
fontconfig-2.13.94-2.amzn2023.0.2.x86_64
giflib-5.2.1-9.amzn2023.0.1.x86_64
graphite2-1.3.14-7.amzn2023.0.2.x86_64
java-22-amazon-corretto-headless-1:22.0.2+9-1.amzn2023.1.x86_64
libICE-1.1.1-3.amzn2023.0.1.x86_64
libX11-common-1.8.10-2.amzn2023.0.1.noarch
libXi-1.8.2-1.amzn2023.0.1.x86_64
libXrender-0.9.11-6.amzn2023.0.1.x86_64
librotli-1.0.9-4.amzn2023.0.2.x86_64
libxcb-1.17.0-1.amzn2023.0.1.x86_64
complete!

```

```

Using CATALINA_BASE:   /opt/tomcat/apache-tomcat-10.1.31
Using CATALINA_HOME:   /opt/tomcat/apache-tomcat-10.1.31
Using CATALINA_TMPDIR: /opt/tomcat/apache-tomcat-10.1.31/temp
Using JRE_HOME:        /usr
Using CLASSPATH:        /opt/tomcat/apache-tomcat-10.1.31/bin/bootstrap.jar:/opt/tomcat/apache-tomcat-10.1.31/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat 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.

```

[batman@ip-172-31-2-182 test_script]$ vi nginx-service.sh
[batman@ip-172-31-2-182 test_script]$ chmod a+x nginx-service.sh
[batman@ip-172-31-2-182 test_script]$ ./nginx-service.sh
active
Nginx service is running.
[batman@ip-172-31-2-182 test_script]$ systemctl stop nginx
Failed to stop nginx.service: Access denied
See system logs and 'systemctl status nginx.service' for details.
[batman@ip-172-31-2-182 test_script]$ sudo systemctl stop nginx
[sudo] password for batman:
[batman@ip-172-31-2-182 test_script]$ ./nginx-service.sh
inactive
Nginx service is not running. Starting the service...
active
Nginx service started successfully.
[batman@ip-172-31-2-182 test_script]$ cat nginx-service.sh
#!/bin/bash

# Check the status of the Nginx service

if systemctl is-active nginx; then
    echo "Nginx service is running."
else
    echo "Nginx service is not running. Starting the service..."
    sudo systemctl start nginx

    # Check if the service started successfully
    if systemctl is-active nginx; then
        echo "Nginx service started successfully."
    else
        echo "Failed to start Nginx service."
    fi
fi

[batman@ip-172-31-2-182 test_script]$ |

```

7) Create a bash script for calculator.

```

[batman@ip-172-31-2-182 test_script]$ vi calculator.sh
[batman@ip-172-31-2-182 test_script]$ chmod a+x calculator.sh
[batman@ip-172-31-2-182 test_script]$ cat calculator.sh
#!/bin/bash

# Function to perform the calculation
calculate() {
    case $2 in
        +) echo "Result: $((($1 + $3)))" ;;
        -) echo "Result: $((($1 - $3)))" ;;
        \*) echo "Result: $((($1 * $3)))" ;;
        /)
            if [ $3 -eq 0 ]; then
                echo "Error: Division by zero!"
            else
                echo "Result: $((($1 / $3)))"
            fi
            ;;
        *) echo "Invalid operator. Please use +, -, *, or /." ;;
    esac
}

# User input
echo "Welcome to the Bash Calculator!"
read -p "Enter the first number: " num1
read -p "Enter the operator (+, -, *, /): " operator
read -p "Enter the second number: " num2

# Perform the calculation
calculate $num1 $operator $num2

[batman@ip-172-31-2-182 test_script]$ ./calculator.sh
Welcome to the Bash Calculator!
Enter the first number: 245
Enter the operator (+, -, *, /): +
Enter the second number: 180
Result: 425
[batman@ip-172-31-2-182 test_script]$

```

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

```

[batman@ip-172-31-2-182 test_script]$ vi new_dir.sh
[batman@ip-172-31-2-182 test_script]$ ./new_dir.sh
bash: ./new_dir.sh: Permission denied
[batman@ip-172-31-2-182 test_script]$ chmod a+x new_dir.sh
[batman@ip-172-31-2-182 test_script]$ ./new_dir.sh
./new_dir.sh: line 8: echo: No such directory: command not found
Creating new directory named imp.logs
[sudo] password for batman:
sudo: a password is required
ERROR: Couldn't create a Directory
[batman@ip-172-31-2-182 test_script]$ vi new_dir.sh
[batman@ip-172-31-2-182 test_script]$ ./new_dir.sh
No such directory
Creating new directory named imp.logs
[sudo] password for batman:
Directory is created successfully
[batman@ip-172-31-2-182 test_script]$ cat new_dir.sh
#!/bin/bash

Dir="/tmp/logfiles/imp.logs"

if [ -d $Dir ]; then
    echo "the directory exists"
else
    echo "No such directory"
    sleep 1
    echo "Creating new directory named imp.logs"
    sudo mkdir -p "$Dir"

#checking again if directory is created or not
sleep 1

if [ -d $Dir ]; then
    echo "Directory is created successfully"
else
    echo "ERROR: Couldn't create a Directory"

fi
fi

[batman@ip-172-31-2-182 test_script]$ cd /tmp/logfiles/imp.logs/
[batman@ip-172-31-2-182 imp.logs]$ cd ..
[batman@ip-172-31-2-182 logfiles]$ ls
imp.logs

```