BASH SCRIPTING TASKS

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

```
root@ubnginx:/home/omer# ./check_dir.sh
Enter directory path.
/etc/bin
Directory does not exists.
root@ubnginx:/home/omer# cd /etc/bin
bash: cd: /etc/bin: No such file or directory
root@ubnginx:/home/omer# ./check_dir.sh
Enter directory path.
/home/omer
Directory exists.
root@ubnginx:/home/omer# cd /home/omer
root@ubnginx:/home/omer# ls
check_dir.sh
root@ubnginx:/home/omer# _
```

1) Created a bash file with a ".sh" extension and named it "check_dir.sh".
2) Gave execute permission to the file using:
   # chmod 755 check_dir.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
echo "Enter directory path:"
read dir

if [ -d "$dir" ]; then
   echo "Directory exists."
else
   echo "Directory does not exist."
fi
```

4) The output of the task is shown in the screenshot above.

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

```
root@ubnginx:/home/omer# ls
check_dir.sh  multiple_files.sh
root@ubnginx:/home/omer# ./multiple_files.sh
Enter file names (space separated):
1.txt 2.c 3.cpp
Created 1.txt
Created 2.c
Created 3.cpp
root@ubnginx:/home/omer#
```

1) Created a bash file with ".sh" and named it "multiple_files.sh"
2) Gave execute permission using:
   # chmod 755 multiple_files.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
#!/bin/bash

echo "Enter file names (space separated):"

read files

for file in $files

do

   touch "$file"

   echo "Created $file"

done
```

4) The output of the task is shown in the screenshot above.

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

```
root@ubnginx:/home/omer# ls
backup_dir.sh  check_dir.sh  multiple_files.sh
root@ubnginx:/home/omer# ./backup_dir.sh
Enter directory to backup:
/home/omer
tar: Removing leading `/' from member names
tar: /home/omer: file changed as we read it
./backup_dir.sh: line 7: unexpected EOF while looking for matching `
root@ubnginx:/home/omer# cd /home/omer
root@ubnginx:/home/omer# ls
backup_dir.sh  backup.tar.gz  check_dir.sh  multiple_files.sh
root@ubnginx:/home/omer# tar xvf backup.tar.gz
home/omer/
home/omer/check_dir.sh
home/omer/backup.tar.gz
home/omer/backup_dir.sh
home/omer/multiple_files.sh
root@ubnginx:/home/omer# ls
backup_dir.sh  backup.tar.gz  check_dir.sh  home  multiple_files.sh
root@ubnginx:/home/omer# _
```

1) Created a bash file with ".sh" and named it "backup_dir.sh"
2) Gave execute permission using:
   # chmod 755 backup_dir.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

   #!/bin/bash

   echo "Enter directory to backup:"

   read dir

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

   echo "Backup saved as backup.tar.gz"

4) The output of the task is shown in the screenshot above.

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

```
root@ubnginx:/home/omer# ./nginx.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
The nginx is installed and started
root@ubnginx:/home/omer# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-08-10 21:28:53 UTC; 17h ago
       Docs: man:nginx(8)
   Main PID: 16072 (nginx)
      Tasks: 3 (limit: 2210)
     Memory: 2.4M (peak: 5.0M)
        CPU: 44ms
     CGroup: /system.slice/nginx.service
             ├─16072 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─16075 "nginx: worker process"
             └─16076 "nginx: worker process"
```

1) Created a bash file with ".sh" and named it "nginx.sh"
2) Gave execute permission using:
   # chmod 755 nginx.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

   #!/bin/bash

   sudo yum install -y nginx # For Amazon Linux

   sudo systemctl start nginx

   sudo systemctl enable nginx

   echo "Nginx installed and started."

4) To check if the nginx is installed and running, use:
   # systemctl status nginx
5) The output of the task is shown in the screenshot above.

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

```
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
Tomcat installed and started.
```

1) Created a bash file with ".sh" and named it "tomcat.sh"
2) Gave execute permission using:
   # chmod 755 tomcat.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

   #!/bin/bash

   apt update
   useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat2
   apt 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"

4) The output of the task is shown in the screenshot above.

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.

```
[root@ip-172-31-25-98 ec2-user]# vi nginx_state.sh
[root@ip-172-31-25-98 ec2-user]# chmod 755 nginx_state.sh
[root@ip-172-31-25-98 ec2-user]# ls
nginx_state.sh
[root@ip-172-31-25-98 ec2-user]# ./nginx_state.sh
Nginx is running.
[root@ip-172-31-25-98 ec2-user]#
```

1) Created a bash file with ".sh" and named it "nginx_state.sh"
2) Gave execute permission using:
   # chmod 755 nginx_state.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
#!/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
```

4) The output of the task is shown in the screenshot above.

7) Create a bash script for a calculator.

```
[root@ip-172-31-25-98 ec2-user]# ls
cal.sh  nginx_state.sh
[root@ip-172-31-25-98 ec2-user]# ./cal.sh
Enter first number:
76
Enter second number:
24
Addition: 100
Subtraction: 52
Multiplication: 1824
Division: 3
[root@ip-172-31-25-98 ec2-user]#
```

1) Created a bash file with ".sh" and named it "cal.sh"
2) Gave execute permission using:
   # chmod 755 cal.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
#!/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))"
```

4) The output of the task is shown in the screenshot above.

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

```
[root@ip-172-31-25-98 ec2-user]# ls
cal.sh  dir_avail.sh  nginx_state.sh
[root@ip-172-31-25-98 ec2-user]# ./dir_avail.sh
Enter directory path:
/home/omer
Directory created.
[root@ip-172-31-25-98 ec2-user]# ./dir_avail.sh
Enter directory path:
/home/omer
Directory already exists.
[root@ip-172-31-25-98 ec2-user]#
```

1) Created a bash file with ".sh" and named it "dir_avail.sh"
2) Gave execute permission using:
   # chmod 755 dir_avail.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
#!/bin/bash


echo "Enter directory path:"

read dir


if [ -d "$dir" ]; then

   echo "Directory already exists."

else

   mkdir "$dir"

   echo "Directory created."

fi
```

4) The output of the task is shown in the screenshot above.

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

```
[root@ip-172-31-25-98 ec2-user]# ls
cal.sh  del3lines.sh  dir_avail.sh  nginx_state.sh  sample.txt
[root@ip-172-31-25-98 ec2-user]# cat sample.txt
1
2
3
4
5
6
7
8
[root@ip-172-31-25-98 ec2-user]# ./del3lines.sh
Enter file name:
sample.txt
Last 3 lines deleted.
[root@ip-172-31-25-98 ec2-user]# cat sample.txt
1
2
3
4
5
```

1) Created a bash file with ".sh" and named it "del3lines.sh"
2) Gave execute permission using:
   # chmod 755 cal.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
#!/bin/bash


echo "Enter file name:"

read file


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

mv temp.txt "$file"

echo "Last 3 lines deleted."
```

4) The output of the task is shown in the screenshot above.

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

```
[root@ip-172-31-25-98 ec2-user]# ls
cal.sh  cpu.sh  del3lines.sh  dir_avail.sh  nginx_state.sh  sample.txt
[root@ip-172-31-25-98 ec2-user]# ./cpu.sh
[root@ip-172-31-25-98 ec2-user]# htop
[root@ip-172-31-25-98 ec2-user]#
```

1) Created a bash file with ".sh" and named it "cpu.sh"
2) Gave execute permission using:
   # chmod 755 cpu.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
#!/bin/bash

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

if [ "$usage" -gt 80 ]; then
   echo "Disk usage high: $usage%" | mail -s "Disk Alert" iamshaikomerfarooq@gmail.com
fi
```

4) To get the email, mail must be installed and set for use. Done that.
5) Check the CPU usage with the htop, top commands. Once the usage exceeds 80%, an automated email is sent to the email entered in the script.
6) The output of the task is shown in the screenshot above.

11) Bash script to monitor disk space, and if it is more than 80%, then send an email notification.

```
[root@ip-172-31-25-98 ec2-user]# ls
cal.sh  cpu.sh  del3lines.sh  dir_avail.sh  disk_space.sh  nginx_state.sh  sample.txt
[root@ip-172-31-25-98 ec2-user]# ./disk_space.sh
[root@ip-172-31-25-98 ec2-user]# df -h
Filesystem        Size  Used Avail Use% Mounted on
devtmpfs          4.0M     0  4.0M   0% /dev
tmpfs             453M     0  453M   0% /dev/shm
tmpfs             181M  464K  181M   1% /run
/dev/nvme0n1p1    8.0G  1.7G  6.4G  21% /
tmpfs             453M     0  453M   0% /tmp
/dev/nvme0n1p128   10M  1.3M  8.7M  13% /boot/efi
tmpfs               91M     0   91M   0% /run/user/1000
[root@ip-172-31-25-98 ec2-user]# dh -hT
```

1) Created a bash file with ".sh" and named it "disk_space.sh"
2) Gave execute permission using:
   # chmod 755 disk_space.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
#!/bin/bash


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


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

    echo "Disk usage high: $usage%" | mail -s "Disk Alert" iamshaikomerfarooq@gmail.com

fi
```

4) To get the email, mail must be installed and set for use. Done that.
5) Check the disk usage with the "df -h" command. Once the usage exceeds 80%, an automated email is sent to the email entered in the script.
6) The output of the task is shown in the screenshot above.

12) Bash script to monitor memory, and if it is more than 80%, then send an email notification.

```
[root@ip-172-31-25-98 ec2-user]# ls
cal.sh  cpu.sh  del3lines.sh  dir_avail.sh  disk_space.sh  memory_usage.sh  nginx
[root@ip-172-31-25-98 ec2-user]# ./memory_usage.sh
[root@ip-172-31-25-98 ec2-user]# free -m
              total        used        free      shared  buff/cache   available
Mem:            904         190         429           0         285         579
Swap:             0           0           0
```

1) Created a bash file with ".sh" and named it "memory_usage.sh"
2) Gave execute permission using:
   # chmod 755 memory_usage.sh
3) Using the vi editor, insert the script with the necessary commands to execute and achieve the desired output, which is as follows:

```
#!/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%" | mail -s "Memory Alert" iamshaikomerfarooq@gmail.com

fi
```

4) To get the email, mail must be installed and set for use. Done that.
5) Check the memory usage with the "free -m" command. Once the usage exceeds 80%, an automated email is sent to the email entered in the script.
6) The output of the task is shown in the screenshot above.

Shaik Omer Farooq

Batch 14