

DevOps

Name: Vijay B

Email: b.vjy05@gmail.com

Mobile: 7010972083

1. Find the count of log statements in the attached(as access.log.zip) file “access.log” with successful response (status code 200).

Answer: 18

The command to find the number of successful responses is:

grep -c ' 200 ' access.log

```
root@DESKTOP-34M0MOJ:/home/vijay# grep -c ' 200 ' access.log
18
root@DESKTOP-34M0MOJ:/home/vijay#
```

2. Write a command to find the files which contain words DEBUG, ERROR, and INFO in any directory of the filesystem.

Answer: **grep -rIE 'DEBUG|ERROR|INFO' /**

```
root@DESKTOP-34M0MOJ:/home/vijay# grep -rIE 'DEBUG|ERROR|INFO' /
/var/cache/man/index.db
/var/cache/jenkins/war/scripts/hudson-behavior.js
/var/cache/jenkins/war/scripts/yui/container/container-debug.js
/var/cache/jenkins/war/scripts/yui/container/container-min.js
/var/cache/jenkins/war/scripts/yui/storage/storage-debug.js
/var/cache/jenkins/war/scripts/yui/storage/storage-min.js
/var/cache/jenkins/war/scripts/yui/connection/connection-debug.js
```

3. What would be the sed command to convert the string input "Ab1Cd2Ef3Gh4Ij5....." to "a-bc-de-fg-hi-j...."?

Answer: `echo "Ab1Cd2Ef3Gh4Ij5....." | sed 's/[0-9]//g; s/[A-Za-z]/L&/g; s/\([a-z]\)\([a-z]\)/\1-\2/g'`

```
root@DESKTOP-34M0MOJ:/home/vijay# echo "Ab1Cd2Ef3Gh4Ij5....." | sed 's/[0-9]//g; s/[A-Za-z]/L&/g; s/\([a-z]\)\([a-z]\)/\1-\2/g'
a-bc-de-fg-hi-j.....
root@DESKTOP-34M0MOJ:/home/vijay#
```

4. Write a script/program to return true if the opening and closing braces are complete, otherwise false.

e.g:

a. Input string:

List { information {{about the FILEs (the current directory by default). Sort entries alphabetically if} none of -cftuvSUX }nor --sort } is specified.

Output: True

b. Input string:

Performs { the specified action on the files. { Valid actions are view, cat (uses only "copious output" rules { and sends output to STDOUT) , compose, com-posetyped, edit and print. If no action is specified, the action will be determined by how the program was called.}

Output: False

Answer:

```
#!/bin/bash

echo "Please enter a string:"
read input_string

open_count=$(echo "$input_string" | grep -o '{' | wc -l)
close_count=$(echo "$input_string" | grep -o '}' | wc -l)

if [[ "$open_count" -eq "$close_count" ]]; then
    echo "True"
else
    echo "False"
fi
```

A output:

```
root@DESKTOP-34M0MOJ:/home/vijay# ./task4
Please enter a string:
List { information {{about the FILEs (the current directory by default). Sort entries alphabetically if} none of -cftuvSUX }nor --sort } is specified.
True
root@DESKTOP-34M0MOJ:/home/vijay#
```

B output

```
root@DESKTOP-34M0MOJ:/home/vijay# ./task4
Please enter a string:
Performs { the specified action on the files. { Valid actions are view, cat (uses only "copious output" rules { and send s output to STDOUT) , compose, com-posetyped, edit and print. If no action is specified, the action will be determined b y how the program was called.}
False
root@DESKTOP-34M0MOJ:/home/vijay#
```

5. Write steps to create and publish a docker image to the docker repository.

e.g., Run tomcat image with a customized landing page, and should be accessible at: <http://localhost:8080/>

Answer:

Step 1: Create a Dockerfile using the Tomcat image and copy a customized landing page index.html onto it.

Dockerfile:

```
FROM tomcat
RUN rm -rf /usr/local/tomcat/webapps/ROOT
COPY index.html /usr/local/tomcat/webapps/ROOT/index.html
EXPOSE 8080
CMD ["catalina.sh", "run"]
```

Step 2: Build the Docker image using the command:

docker build -t mytomcat .

Step 3: To publish a Docker image to the Docker repository, first log in to Docker Hub and push the image using the command:

docker login

```
root@DESKTOP-34M0M0J:/home/vijay# docker login -u vjyguvi
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@DESKTOP-34M0M0J:/home/vijay# |
```

docker push vjyguvi/senpiper:latest

```
root@DESKTOP-34M0M0J:/home/vijay# docker push vjyguvi/senpiper:latest
The push refers to repository [docker.io/vjyguvi/senpiper]
9bbb68dda9a4: Preparing
5f70bf18a086: Preparing
832cbd20bac2: Preparing
b2fe4bd67e07: Preparing
4e5b554b7345: Preparing
39cf0ac89a5a: Waiting
f844dcf94898: Waiting
3359bc3d7a6a: Waiting
4b7c01ed0534: Waiting
unauthorized: access token has insufficient scopes
root@DESKTOP-34M0M0J:/home/vijay# |
```

Step 4: Run the Docker image using the command:

docker run -d -p 8080:8080 vjyguvi/senpiper

```
root@DESKTOP-34M0M0J:/home/vijay# docker run -d -p 8080:8080 vjyguvi/senpiper
f7b95242eb06167891f17725df00c97359345b9ff024b352322ce1a90c465603
root@DESKTOP-34M0M0J:/home/vijay#
```

Output:

