

Senpiper Technologies DevOps Internship

ASSIGNMENT

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

Answer – By using command –

“The successful response log statements with status code 200 are – 18
(HTTP/1.1 = 16 and HTTP/1.0 = 2)”

Minimal Command: `cat access.log | grep "200" | wc -l` (# Not a better approach)

It is possible that the string 200 will occur elsewhere in log file. So, we should ensure that we only match HTTP success code. So that –

Command:

```
cat access.log | grep 'HTTP/1.1\" 200\\|HTTP/1.0\" 200' | wc -l
```

Or:

```
grep -e "HTTP/1.1\" 200" -e "HTTP/1.0\" 200" access.log | wc -l
```

We can also find HTTP status 200 using “**awk**” command but for that log should be in same format (In access.log file, logs aren’t in same format.)
If logs modified in `["%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i"]` this format, This command will find the HTTP status which match with 200 status.)

Command: `awk '($9 ~ /200/)' access.log | wc -l`

Question 2 - Write command to find the filenames containing words DEBUG, ERROR, and INFO in any directory of the filesystem.

Answer -

This will check whole Linux filesystem for the given words (**sensitive**) and return filename with relative path (It also return that filename, whose name contains these words in prefix or suffix) -

Command -

```
find /* -type f \( -name "*ERROR*" -o -name "*DEBUG*" -o -name "*INFO*" \)
```

Question 3 - What would be the sed command to convert the string input "Ab1Cd2Ef3Gh4Ij5....." to "abcdefghij...."?

Answer -

First remove numerical Integers from string and then convert the string to required string by changing that in lower case -

Command -

```
echo "Ab1Cd2Ef3Gh4Ij5....." | sed 's/[0-9]*//g' | sed 's/[A-Z]/\L&/g'
```

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

Answer – Using Python –

1. Install the python3 and configure it in your Operating System (**taking Linux**)
2. Create a file with name “**check-parenthesis.py**” using vim editor and insert this program –

```
#!/usr/bin/python3

def BalancedBrackets(Str):
    # stack for storing opening brackets
    stack = []

    # Loop for checking string
    for char in Str:
        # if its opening bracket, so push it in the stack
        if char == '{' or char == '(' or char == '[':
            stack.append(char)
        # else if its closing bracket then check if the stack is empty then
        # return false or pop the top most element from the stack
        # and compare it
        elif char == '}' or char == ')' or char == ']':
            if len(stack) == 0:
                return False
            top_element = stack.pop() # pop
            # function to compare whether two
            # brackets are corresponding to each other
            if not Compare(top_element, char):
                return False
        # lastly, check that stack is empty or not
    if len(stack) != 0:
        return False

    return True

# Function which compare the top_element and char –
def Compare(opening, closing):
    if opening == '(' and closing == ')':
        return True
    if opening == '[' and closing == ']':
```

```
    return True
if opening == '{' and closing == '}':
    return True
return False
```

Taking Input string from user -

```
input_string = input("Dear User, Please enter the input string with  
parenthesis: ")
print(BalancedBrackets(input_string))
```

3. Save and exit using “:wq”.
4. Now, we have to give permission to this file for execution so, to provide permission use:

Command: `chmod 777 check-parenthesis.py`

5. Now, You can run this script using:

Command: `./check-parenthesis.py`

6. Input any string containing parenthesis or not (Taking “{123(456[.768.])}”)
7. Done – If given string has its opening and closing bracket, It return TRUE otherwise it return FALSE.

Run this code online here -

<https://colab.research.google.com/drive/1LJTGisr9u5bOiOuUKUorOMehIttydQWD?usp=sharing>

Question 5 - Write steps to create and publish a Docker image to the Docker repository.

Answer – For creating a Docker Image, There are two ways:

1. By Using Dockerfile (Building Customized Webserver) –

a. Creating an “index.html” file and inserting some html which will be shown on the homepage (use :wq to exit) –

Command: vim index.html

Html Command:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Assignment</title>
  </head>
  <body>
    <h1>This is the Assignment for the DevOps role.</h1>
    <p> You are hitting a container </p>
  </body>
</html>
```

b. Creating Dockerfile through vi editor and insert data (for save and exit : esc -> :wq) –

Command: vim Dockerfile

Data:

```
FROM ubuntu:latest
ENV DEBIAN_FRONTEND=noninteractive
RUN apt-get update
RUN apt-get install apache2 -y
RUN apt-get install apache2-utils -y
RUN apt-get clean
COPY index.html /var/www/html/
EXPOSE 80
CMD ["apache2ctl","-D","FOREGROUND"]
```

c. Building Docker image from Dockerfile –

Command: docker build -t nirdeshkumar02/webserver:latest .

d. Push this image to docker repository / hub –

First, we have to login in docker hub and provide username and password

Command: docker login

Now, Push this image to docker hub (docker image name should be same as docker repository)

Command: docker push nirdeshkumar02/webserver:latest

e. Launching the webserver with new customized image –

Pull image from docker hub and run it on port 80

Command: docker run -d -p 80:80 nirdeshkumar02/webserver:latest

To see result on web browser –

http://<host-ip>:80

2. From Docker Container (Building Customized Webserver) –

a. First, We pull the latest image of Ubuntu and create container from it –

Command: docker run -it --name webserver ubuntu:latest

b. Now, I'm inside the ubuntu. Update apt –

Command: apt update -y

c. After updating, Install apache2 server –

Command: apt install -y apache2

d. After installing apache2, Install apache2 server – utilities

Command: apt install -y apache2-utils

e. Now, create a webpage at location “/var/www/html/index.html”. and exit from the container –

Command: vim /var/www/html/index.html

Html Command:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Assignment</title>
  </head>
  <body>
    <h1>This is the Assignment for the DevOps role.</h1>
    <p> You are hitting a container </p>
  </body>
</html>
```

exit from vi editor: “:wq”

also from container using: exit

f. Now, here, we have container in which apache server is installed and our page is configured. Now, create a customized Docker image from this container –

Command: docker commit webserver nirdeshkumar02/webserver:latest

g. Push this image to docker repository / hub –

```
# First, we have to login in docker hub and provide username and password
```

Command: docker login

```
# Now, Push this image to docker hub (docker image name should be same as docker repository)
```

Command: docker push nirdeshkumar02/webserver:latest

h. Launching the webserver with new customized image –

```
# Pull image from docker hub and run it on port 80
```

Command: docker run -d -p 80:80 nirdeshkumar02/webserver:latest
apache2ctl -D FOREGROUND

```
# To see result on web browser –
```

```
http://<host-ip>:80
```

Bonus Question (Non Mandatory)

Question 6 - Given the following snippet of nginx server configuration:

```
server {
```

```
    listen 80 default_server;
```

```
    root /var/www/html;
```

```
    server_name *.senpiper.com senpiper.com;
```

```
} Write a location block for requests “/api” that will serve the requests from  
backend controller “/welcome/home”.
```


Answer:

```
Server {  
    listen 80 default_server;  
    root /var/www/html;  
    server_name *.senpiper.com senpiper.com;  
    location /api {  
        root /welcome/home;  
        try_files $uri =404;  
    }  
}
```