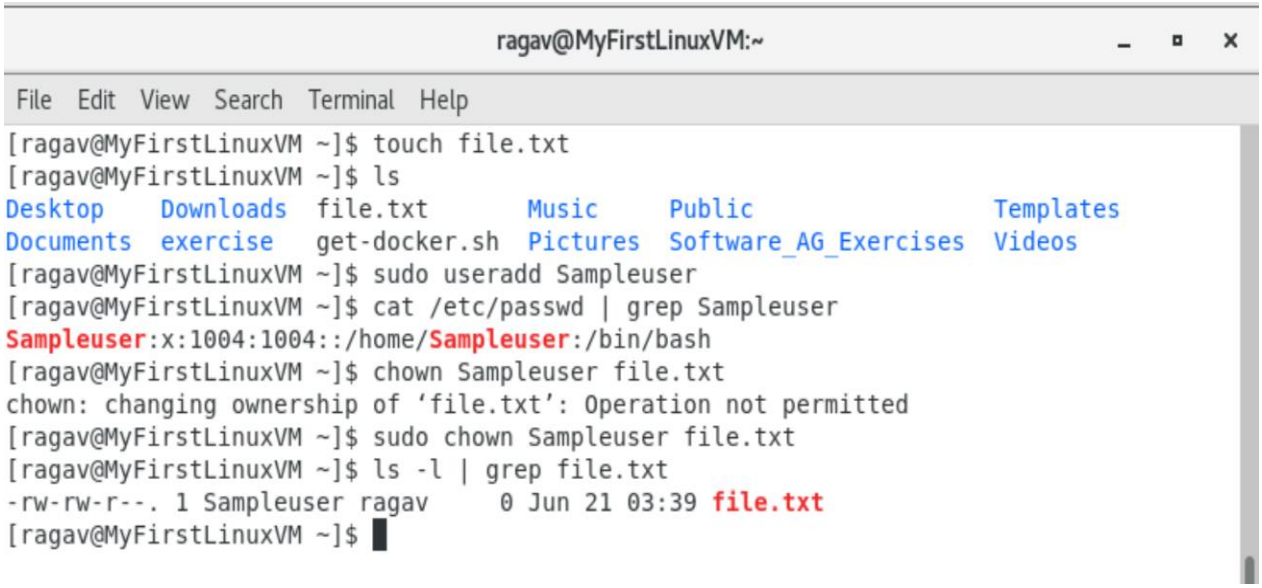
SOFTWARE AG EXERCISES
LINUX AND BASH SHELL SCRIPTING & DOCKER

Linux and bash shell scripting :

1) Create an file named file.txt, create an user sampleuser. Change the ownership of the file to sampleuser
Solution:

```
# creating file using touch command
touch file.txt

# for creating user you have to switch to root.
sudo useradd sampleuser

# change ownership of file.txt using chown
chown sampleuser file.txt
```

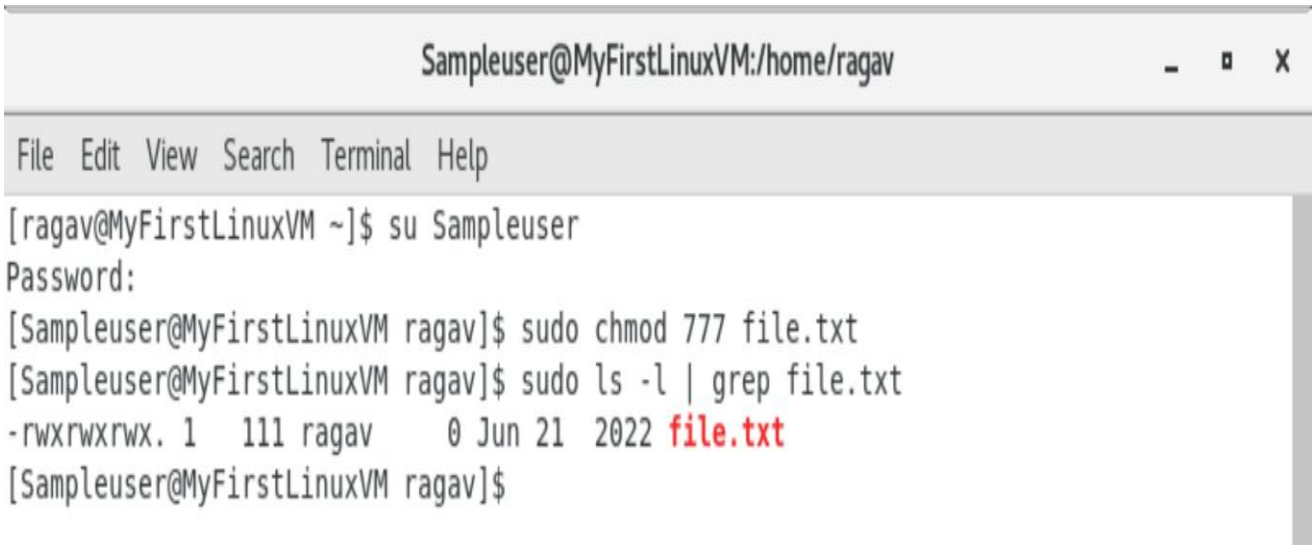
Output:



2) Switch to sample user in terminal then change the permission of the file.txt to the user and group as rwx.
Solution:

```
# switch user using su command
su sampleuser

# change the permission of the file.txt to the user and group as rwx using chmod command
chmod 777 file.txt
```

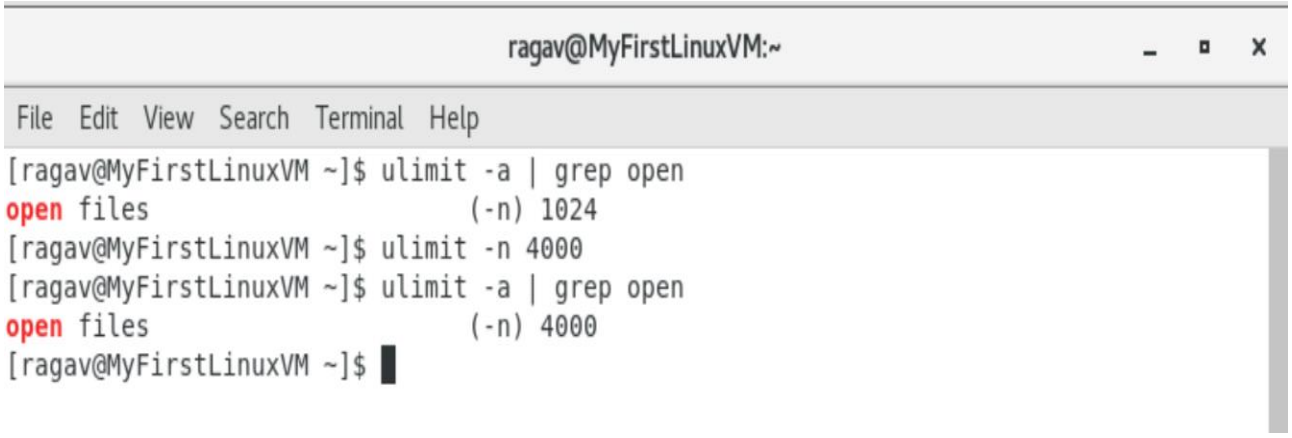
Output:



3) Increase the open file limit
Solution:

```
# show the current limitation use the below  command
ulimit -a | grep open

# for editing the limit to 4000 use the below command
ulimit -n 4000
# In above command 4000 can be replaced by any desired value.
```

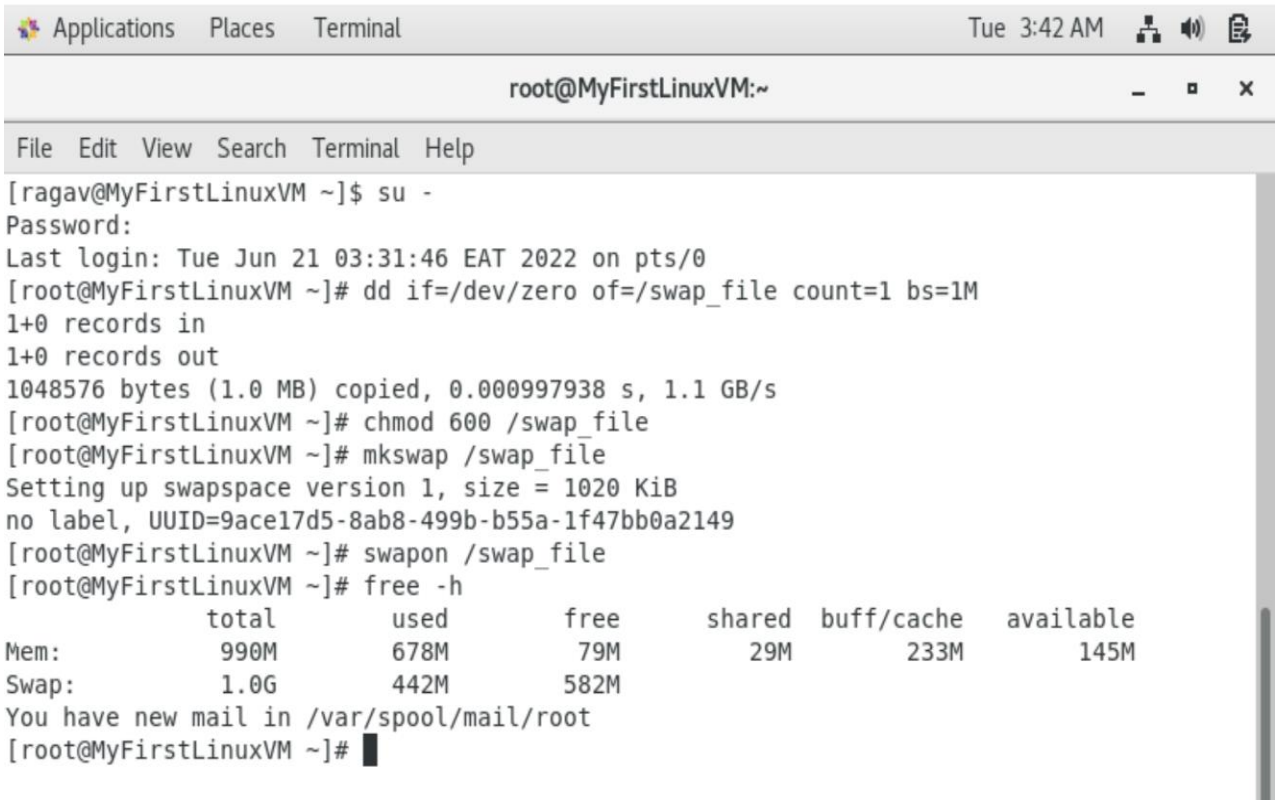
Output:



4) Increase the swapsize by adding new swap storage
Solution:

```
# create a file that will be used for swap
dd if=/dev/zero of=/swap_file count=1 bs=1M

# secure the swap file
chmod 600 /swap_file

# for setting up the file as Linux swap area use mkswap command
sudo mkswap /swap_file

# enable the swap using swapon command:
sudo swapon /swap_file

# for verifing that the swap is active, use free command
free -h
```

Output:

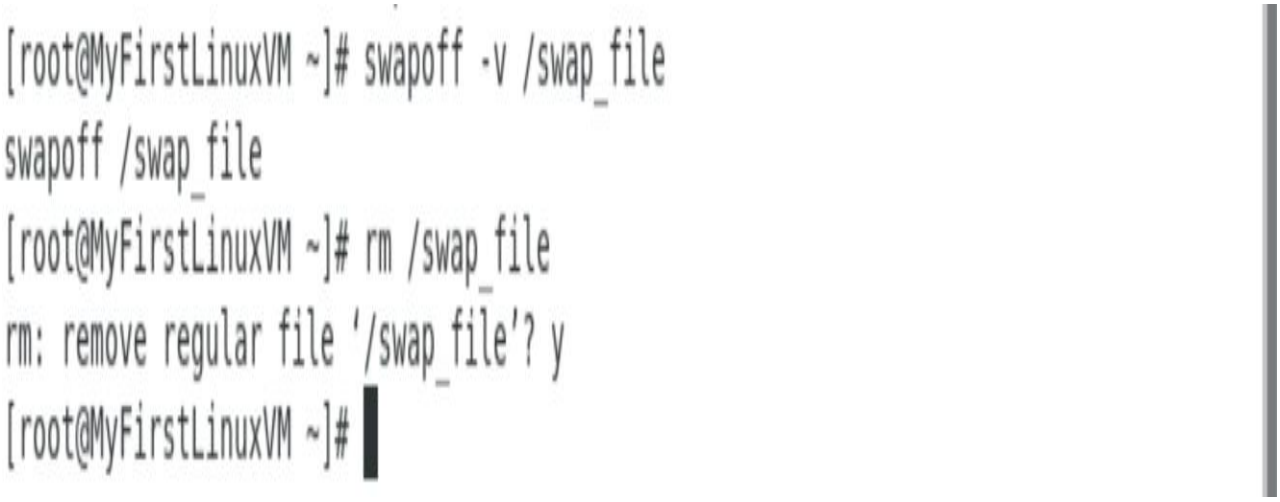


5) Remove the swap storage

Solution:

```
#1) First, deactivate the swap using below command

swapoff -v /swap_file

#2) Remove the swap file entry /swapfile swap swap defaults 0 0 from the
/etc/fstab file.

#3) Finally, delete the actual swapfile file using the rm command.

rm /swap_file
```

Output:



6) Write an script that will read the file content.txt that contains the names,age,job title and print the value in this order “name(age) – jobtitle”

Contents.txt file contents

Raj 30 Developer
Ram 25 Ops
Gokul 21 Intern
Sundar 45 CEO
Anil 38 SRE

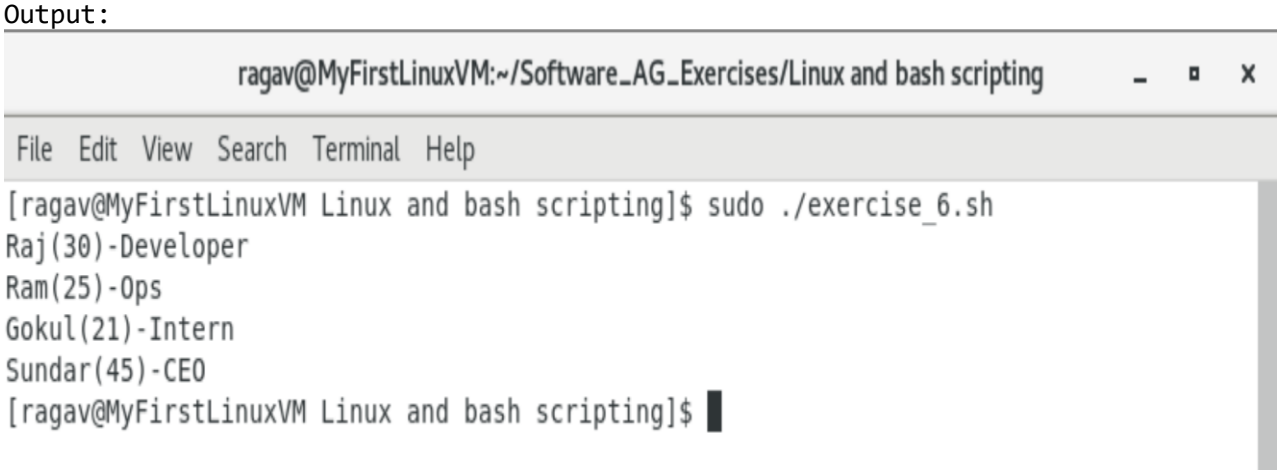
Output example

Raj(30) – Developer

Solution:

```
#!/bin/bash
################################################################################
################################################################################
##########################

#Author : Ragavendra Vigneshwaran R
#Date : 20-06-2022
#Description : This file will write an script that will read the file
content.txt that contains the names,age,job title and print the value in this
order “name(age) – jobtitle”
<<description
Contents.txt file contents
Raj 30 Developer
Ram 25 Ops
Gokul 21 Intern
Sundar 45 CEO .
Output example:
Raj(30) – Developer
description
#Date modified : 20-06-2022

################################################################################
################################################################################
##########################

# to write the contents in the "content.txt" use the following:
echo 'Raj 30 Developer
Ram 25 Ops
Gokul 21 Intern
Sundar 45 CEO' > content.txt

sudo chmod 777 content.txt
# For desired output use below command
awk '{print $1 "("$2")-"$3 }' content.txt
```

Output:



7) Write an script that will get the company name as arguments(command line arguments) and print their products by parsing the following json file

```
{
“Ibm”: [“RedHat”,”Mainframe”,”IBM cloud”,”RHEL”],
“google”: [“k8s”,”kaggle”,”firebase”],
“microsoft”: [“windows”,”azure”,”office365”],
“amazon”: [“AWS”,”AWS Gov”,”Amplify”],
“softwareag”: [“webmethods”,”AIRS”,”ADABAS”,”CumIOT”]
}
```

Solution:

```
#!/bin/bash
################################################################################
################################################################################
###################

#Author : Ragavendra Vigneshwaran R
#Date : 20-06-2022
#Description : This file will write an script that will get the company name
as arguments(command line arguments) and print their products by parsing the
following json file
<<description
{
“Ibm”: [“RedHat”,”Mainframe”,”IBM cloud”,”RHEL”],
“google”: [“k8s”,"kaggle”,”firebase”],
“microsoft”: [“windows”,”azure”,”office365”],
“amazon”: [“AWS”,”AWS Gov”,”Amplify”],
“softwareag”: [“webmethods”,”AIRS”,”ADABAS”,”CumIOT”]
}
description
#Date modified : 20-06-2022

################################################################################
################################################################################
###################
# to write json in the file
echo '
{
“Ibm”: ["RedHat","Mainframe","IBM cloud","RHEL"],
“google”: ["k8s","kaggle","firebase"],
“microsoft”: [ "windows","azure","office365"],
“amazon”: ["AWS","AWS Gov","Amplify"],
“softwareag”: ["webmethods","AIRS","ADABAS","CumIOT"]
}
'> company.json

# for printing their products by parsing the following json file
cat company.json | jq -c ".$1"
```

8) Create an script that will calculate the sum of prime numbers between 0 to an given number(to be given as an argument) name it child.sh, create another script named parent.sh which check if the child script exists in current dir and if exists change permission to execute for current user and call the child script from the parent script passing the number as argument

Solution:

child.sh:

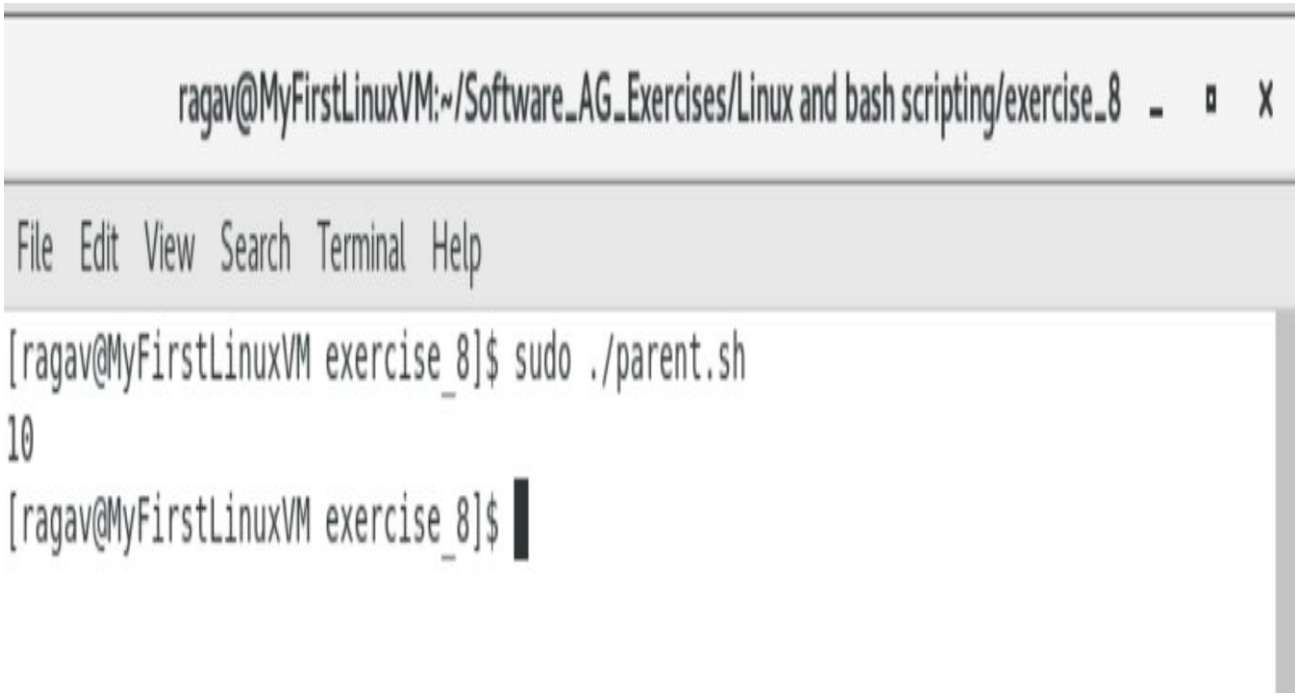
```
#!/bin/bash
################################################################################
######

# Author : Ragavendra Vigneshwaran R
# Date created : 20-06-2022
# Description : This file will create sum of prime numbers between 0 to given
number
# Date modified: 20-06-2022

################################################################################
######

n=$1
sum=0
for (( i=2; i<=$n; i++ ))
do
 c=0
 for(( j=1; j<=$i; j++ ))
 do
 if(( $i%j == 0))
 then
 c=$c+1
 #echo "$c"
 fi
 done
 if(( $c == 2 ))
 then
 sum=$((sum + i));
 fi
done
echo "$sum"
```

parent.sh:

```
#!/bin/bash
################################################################################
################################################################################
#################################################################
```

```
# Author : Ragavendra Vigneshwaran R
# Date created : 20-06-2022
# Description : This file will check if the child script exists in current dir
and if exists change permission to execute for current user and call the child
script from the parent script passing the number as argument.
# Date modified: 20-06-2022

#########################################################################
#########################################################################
#########################################################################

if [ -f "child.sh" ]
then
 chmod u+x child.sh
 ./child.sh 5
else
 echo "child.sh is not exist in current directory"
fi
```

Output:

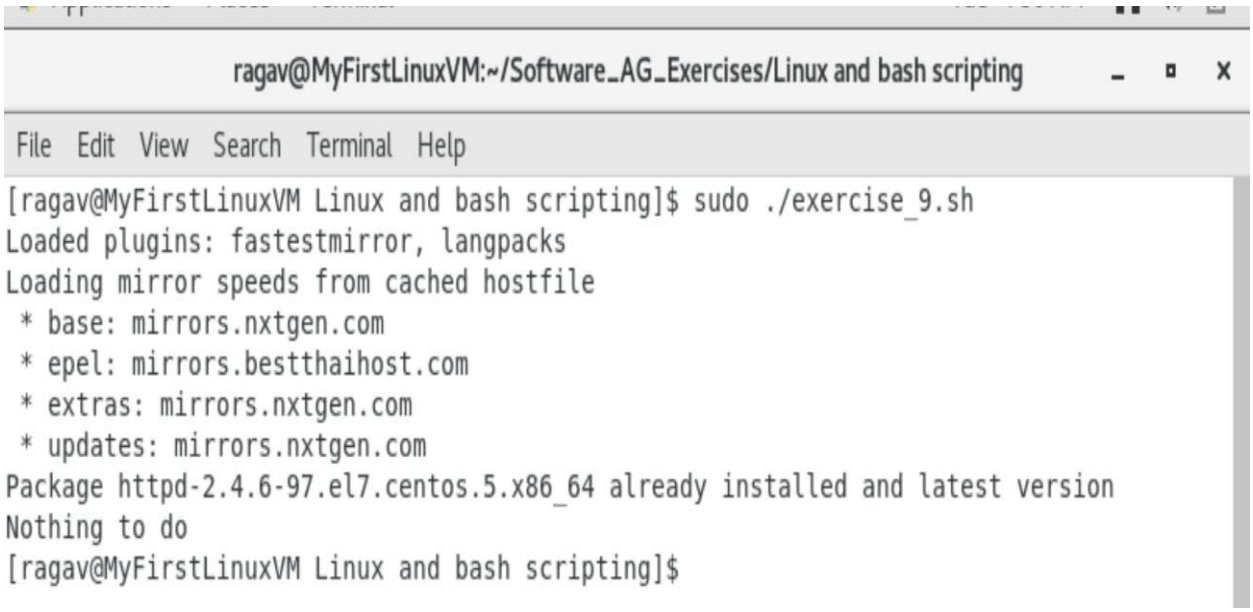


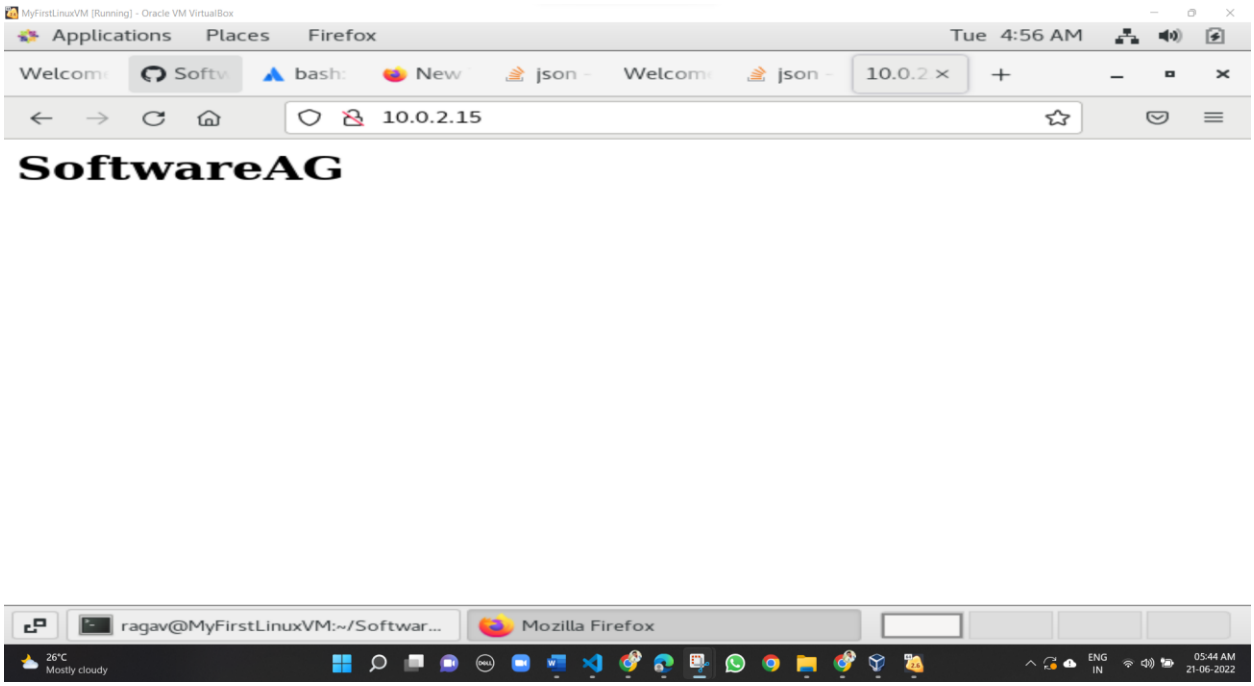
9) Install the apache httpd server(centos preferred), and the home page should say “SoftwareAG”

Solution:

```
#!/bin/bash
################################################################################
############################

#Author : Ragavendra Vigneshwaran R
#Date : 20-06-2022
#Description : This file will install the apache httpd server, and the home
page should say “SoftwareAG”.
#Date modified : 20-06-2022

################################################################################
############################
# install httpd using below command
sudo yum install httpd

# start the apache service using systemctl command
sudo systemctl start httpd

# change the content in "/var/www/html/index.html" file for printing
"SoftwareAG" in the home page.
sudo echo '
<html>
<head>
<h1>SoftwareAG</h1>
</head>
</html>
'>/var/www/html/index.html
```

Output:





10) Make the Https server an Linux systemd service and write an script to stop, start and restart the server via systemctl

Solution:

servicescript.sh:

```
#!/bin/bash
################################################################################
############################

#Author : Ragavendra Vigneshwaran R
#Date : 20-06-2022
#Description : This file will make the Https server an Linux systemd service.
#Date modified : 20-06-2022

################################################################################
############################

while true
do
python3 -m http.server 8081
done

[service]
ExecStart=/home/ragav/Software_AG_Exercises/Linux and Bash
scripting/servicescript.sh
```

start_the_server.sh:

```
#!/bin/bash
################################################################################
############################

#Author : Ragavendra Vigneshwaran R
#Date : 20-06-2022
#Description : This file will start the server using systemctl.
#Date modified : 20-06-2022

################################################################################
############################

sudo systemctl start servicescript
```

stop_the_server.sh:

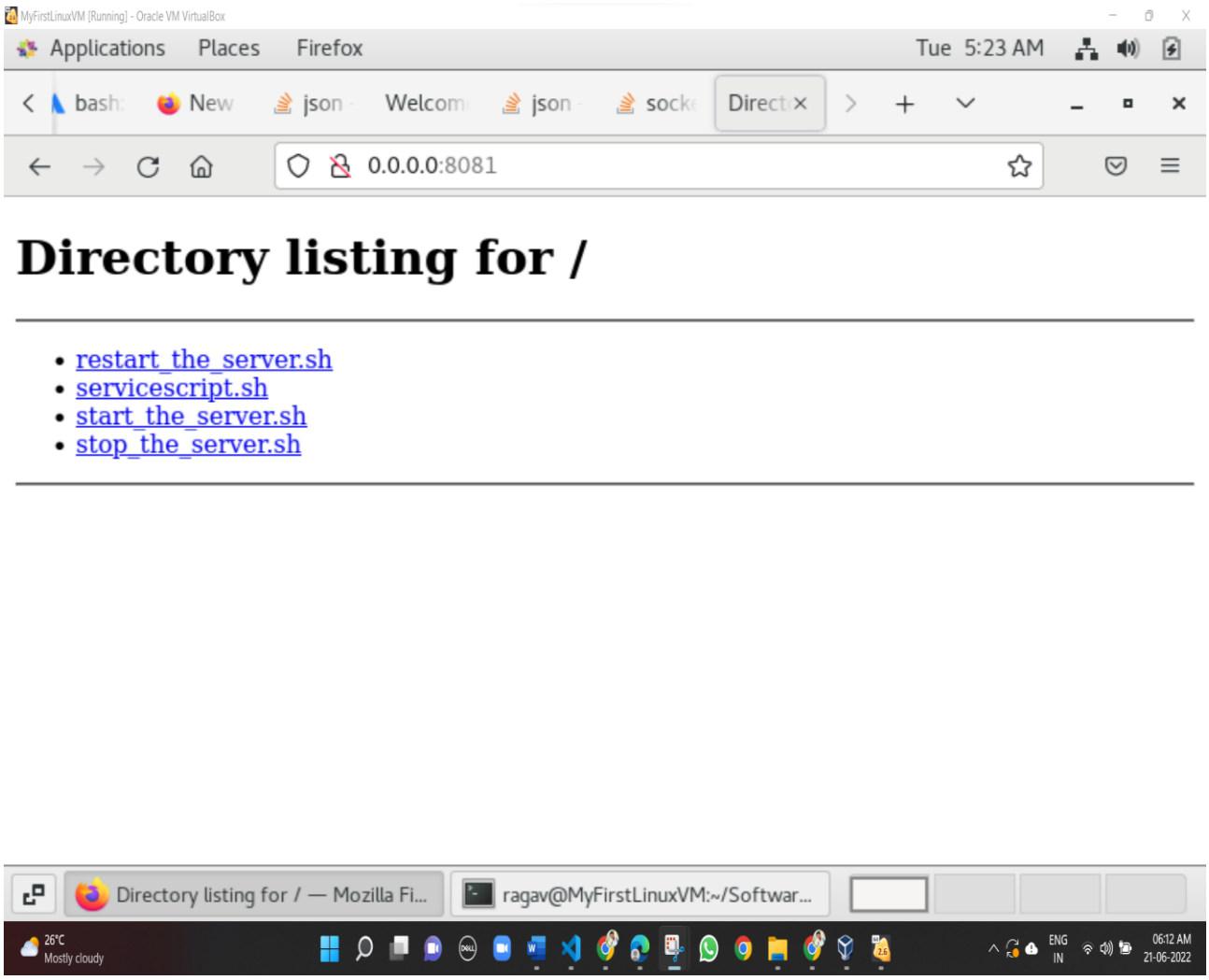
```
#!/bin/bash
################################################################################
############################

#Author : Ragavendra Vigneshwaran R
#Date : 20-06-2022
#Description : This file will stop the server using systemctl.
#Date modified : 20-06-2022

################################################################################
############################

sudo systemctl stop servicescript
```

restart_the_server.sh:

```
#!/bin/bash
################################################################################
############################

#Author : Ragavendra Vigneshwaran R
#Date : 20-06-2022
#Description : This file will restart the server using systemctl.
#Date modified : 20-06-2022

################################################################################
############################

sudo systemctl restart servicescript
```

Output:

```
[ragav@MyFirstLinuxVM exercise_10]$ sudo ./servicescript.sh
Serving HTTP on 0.0.0.0 port 8081 (http://0.0.0.0:8081/) ...
```



DOCKER:

1) Create an docker volume named sampledata.

Solution:

$ docker volume create sampledata

2) Spin up an ubuntu docker container, mount the sampledata docker volume. Pass the env variable os=ubuntu to the docker container when bringing it up

Answer:

$ docker run -d -it --name ubuntu1 --mount source=sampledata,destination=bin/data --env OS='ubuntu' ubuntu

3) Create an dir names /sampledir mount it to an nginx:1.19.0 docker container

Answer:

$ mkdir sampledir

$ docker run -d -it --name nginx1 -v /Users/hariv/sampledir:/var/www/html nginx

4) Create an docker file to create an image with httpd2 server in it, on running the image as an container the server should start up

Answer:

Dockerfile

```
FROM httpd:2.4
COPY ./website/ /usr/local/apache2/htdocs/
```

Then, run the commands to build and run the Docker image:

```
$ docker build -t my-apache2 .
$ docker run -dit --name my-running-app -p 8080:80 my-apache2
```

5) Create an container out of the image you created in earlier step expose the 80 port to the parent host

Answer:

```
$ docker run -d –name apache-container -p 88:80 myapache2
```

6) Create an bridge network and spin up two containers in the network one with the image you created.

Answer:

```
$ docker network create -d bridge mynetwork
$ docker network connect mynetwork container-01
$ docker network connect mynetwork container-02
```

7) Create an docker compose file with an nginx and an redis server both connected via an bridge network

Answer:

DockerFile:

```
version: '2.0'
services:
   nginx:
     image: nginx
     networks:
      - mynetwork
     ports:
      - "82:80"
   redis:
     image: redis
     ports:
      - "3000:6379"
     networks:
      - mynetwork
networks:
     mynetwork:
```

Scenario Based Docker exercises:

i)Write Docker file to create an image with the apache httpd2 server make the start of the httpd2 server as the container entry point.

ii)Create an git repository with some html files for our webserver, clone the repo to an local linux machine.

iii)Create an bash script which will pull the contents from the GitHub Repo every 5 hours (use Linux Cron) to the local cloned repo, The script should log all these activities to an log file with timestamp (file which keep track of all script runs)

iv)Spin up the container out of the image you create from the docker file, expose the port 80, mount the dir where you have cloned the git repo to the dir inside the container from where httpd2 server reads the HTML files (/var/www/html)

Do all the above steps inside an AWS linux EC2 machine and share the publicip/DNS of the VM, by allowing the traffic to reach your webpage

```
First I have launched an EC2 instance with Ubuntu AMI.

Next step is to install the git and docker in the ec2 instance by using the
commands.

$ sudo apt-get update
$ sudo apt-get install docker
$ sudo apt-get install git

Public IP address : 13.52.163.2

i) Docker file to create an image with the apache httpd2 server make the start
of the httpd2 server as the container entry point.
DockerFile
FROM ubuntu:latest
RUN apt-get -y update
RUN apt-get install -y apache2
RUN apt-get install -y apache2-utils
EXPOSE 80
COPY html_pages/index.html /var/www/html/index.html
ENTRYPOINT ["/usr/sbin/apache2ctl"]
CMD ["-D","FOREGROUND"]

ii) Next step is to clone the repository into the ec2 directory

$ sudo git clone https://github.com/Ragavendra-Vigneshwaran-R/html_pages.git
```

iii) bash script which will pull the contents from the GitHub Repo every 5 hours (use Linux Cron) to the local cloned repo, The script should log all these activities to an log file with timestamp (file which keep track of all script runs)

```
updating_log.sh
git pull
head -n 1 /var/log/lastlog/ >> log.txt
date>>log.txt
```

```
crontab -l
* */5 * * * pull.sh
```

iv) Build the image

```
$ sudo docker build -t apache-server ./DockerFile
```

Next step is to spin up the container

```
$ sudo docker run -p 80:80 -v /home/ubuntu/html_pages/:/var/www/html apache-server
```