TASK 1

a. What is DevOps? How does DevOps Work?

DevOps is an approach to software development that combines development (Dev) and IT operations (Ops) to create a culture of collaboration and communication. The main goal of DevOps is to improve the speed and efficiency of the software development lifecycle by automating software delivery and infrastructure changes. DevOps teams work together throughout the entire software development process, using a variety of tools and technologies to automate the different stages of the process, such as version control systems, configuration management tools, and continuous integration and deployment tools.

DevOps involves several stages, including planning, coding, building, testing, deployment, and monitoring. DevOps teams use automation and continuous integration and delivery (CI/CD) to make the software development process faster, more efficient, and more reliable. The benefits of DevOps include increased collaboration between teams, improved quality and reliability of software releases, and reduced costs and time for software development and maintenance. Overall, DevOps is an approach that emphasizes collaboration, communication, and automation to improve the software development process.

b. Describe the DevOps lifecycle.

The DevOps lifecycle typically includes the following stages:

- Plan: In this stage, the development and operations teams work together to plan the software development process. This involves identifying the requirements, setting goals and objectives, and deciding on the tools and technologies to be used.
- Code: In this stage, the development team writes and tests the code for the software application. They use version control systems to manage code changes and ensure that the code is of high quality and meets the requirements.
- Build: In this stage, the code is built into an executable package, such as a binary or Docker image. This involves compiling the code and packaging it with any dependencies and libraries required for the application.
- Test: In this stage, the software application is tested to ensure that it meets the requirements and is of high quality. This involves automated testing, such as unit testing and integration testing, as well as manual testing by the development and operations teams.
- Deploy: In this stage, the software application is deployed to the production environment. This involves configuring the infrastructure, setting up the application, and deploying the code.
- Operate: In this stage, the operations team manages the software application in the production environment. This involves monitoring the application, troubleshooting issues, and making any necessary changes to the infrastructure or code.
- Monitor: In this stage, the performance and usage of the software application are monitored. This involves tracking metrics such as response time, uptime, and user activity, and making any necessary improvements to the application or infrastructure.

c. Describe DevOps Principles.

There are several key principles that underpin the DevOps approach to software development:

- Collaboration: DevOps emphasizes collaboration between development and operations teams, as well as other stakeholders in the software development process. This involves breaking down silos and fostering a culture of communication and cooperation.
- Continuous Integration and Delivery: DevOps emphasizes continuous integration and delivery (CI/CD), which involves automating the build, test, and deployment processes. This enables faster, more frequent releases of software, while also improving quality and reducing errors.
- Automation: DevOps emphasizes automation of manual processes, such as testing, deployment, and infrastructure management. Automation reduces the risk of errors and improves efficiency, allowing teams to focus on higher-level tasks.
- Infrastructure as Code: DevOps emphasizes the use of infrastructure as code (IaC), which involves managing infrastructure in the same way as software code. This enables teams to treat infrastructure as a version-controlled asset, with changes tracked and managed like any other code.
- Monitoring and Logging: DevOps emphasizes monitoring and logging of software applications and infrastructure, with metrics and logs used to identify and troubleshoot issues. This enables teams to identify problems quickly and make improvements to the software and infrastructure.
- Security: DevOps emphasizes security throughout the software development lifecycle, with security testing integrated into the CI/CD pipeline. This helps to ensure that software is secure and compliant with regulations.

d. Explain the benefits of DevOps.

The benefits of DevOps include:

- Faster time to market: DevOps enables quicker releases by automating testing and integrating CI/CD pipelines.
- Improved collaboration: DevOps fosters communication between teams, breaking down silos and promoting a unified approach.
- Higher quality software: DevOps automation and testing improve reliability, reducing errors and defects.
- Lower costs: DevOps automation streamlines processes, reducing time and effort and enabling faster delivery.
- Improved security: DevOps security practices reduce the risk of breaches and ensure compliance with regulations.
- Greater scalability and flexibility: DevOps enable fast and efficient scaling and changes to meet business requirement.

TASK 2

**a. Installing VMware Workstation or VirtualBox.**
1. Download the installation file
2. Run the installation file
3. Accept the license agreement
4. Choose the installation location
5. Choose the setup type
6. Configure shortcuts
7. Install VMware Workstation
8. Finish the installation

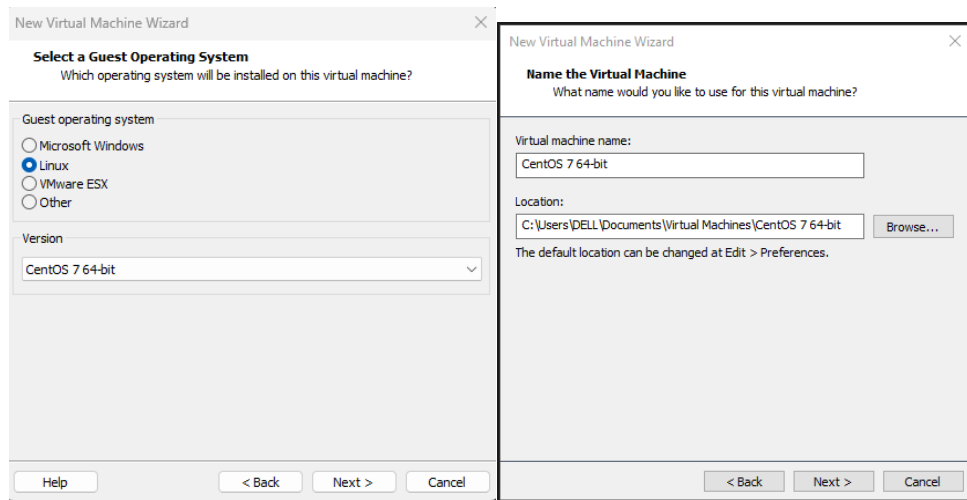**b. Creating Virtual Machine of CentOS 7.**
1. Open VMware Workstation
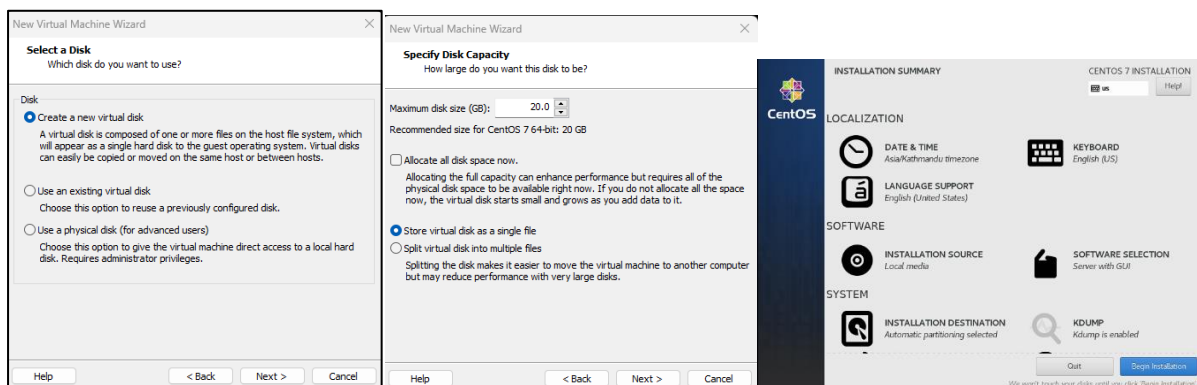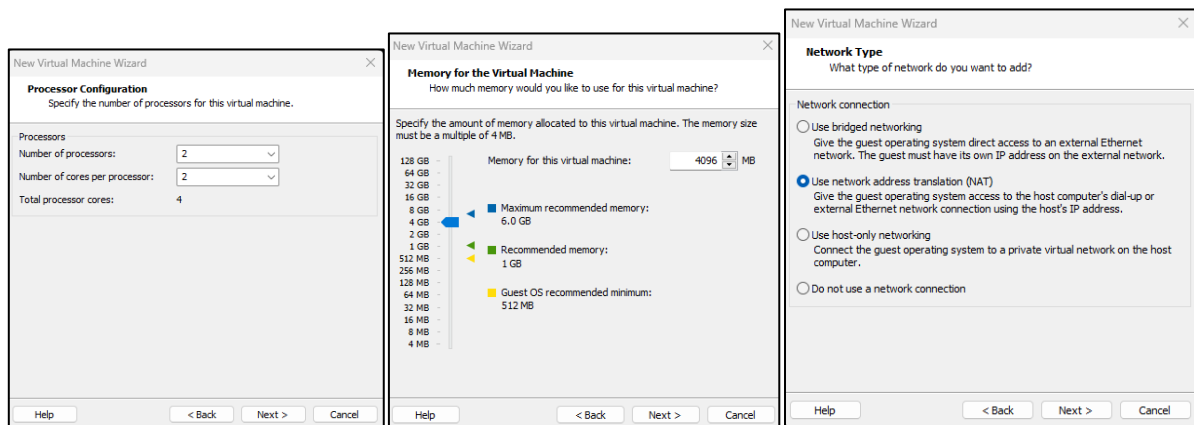2. Create a new virtual machine
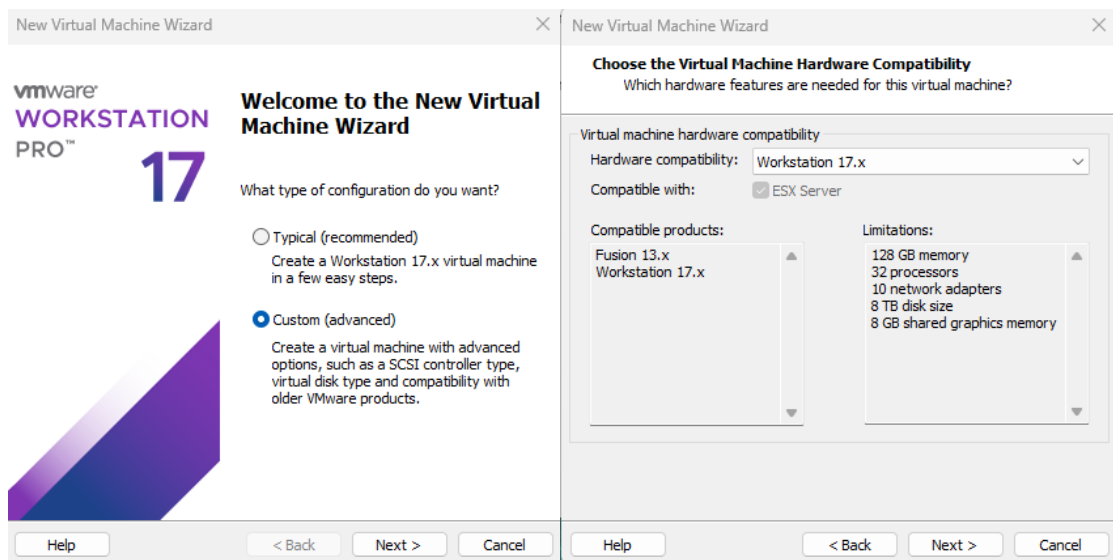
3. Choose the installation media
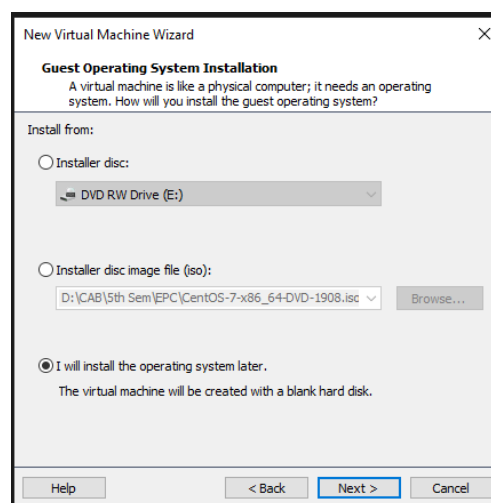
4. Select the guest operating system

5. Configure the virtual machine hardware.

6. Customize other settings
7. Finish creating the virtual machine
8. Start the virtual machine
9. Install CentOS 7
10. Complete the installation.

c. Assign the hostname of Linux machine as <yourname>.devops.com

```
[root@localhost ~]# hostnamectl set-hostname sudesh.devops.com
[root@localhost ~]# hostnamectl
   Static hostname: sudesh.devops.com
         Icon name: computer-vm
           Chassis: vm
        Machine ID: f1baa5c585e5489d8caa8cbc6249cee9
           Boot ID: d4ebc14c54104b8898595f664248a4ba
    Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-1062.el7.x86_64
      Architecture: x86-64
[root@localhost ~]#
```

d. Configure your network interface with static ip address and start the network service.

```
[root@sudesh /]# ping sudesh.devops.com -c 3
PING sudesh.devops.com (192.168.122.1) 56(84) bytes of data.
64 bytes from sudesh.devops.com (192.168.122.1): icmp_seq=1 ttl=64 time=0.316 ms
64 bytes from sudesh.devops.com (192.168.122.1): icmp_seq=2 ttl=64 time=0.101 ms
64 bytes from sudesh.devops.com (192.168.122.1): icmp_seq=3 ttl=64 time=0.105 ms

--- sudesh.devops.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 0.101/0.174/0.316/0.100 ms
[root@sudesh ~]# cat > /etc/sysconfig/network-scripts/ifcfg-ens33
TYPE="Ethernet"
BOOTPRO="static"
NAME="ens33"
DEVICE="ens33"
ONBOOT="yes"
IPADDR="192.168.187.151"
NETMASK=255.255.255.0"
GATEWAY="192.168.187.2"
DNS1="8.8.8.8"
```

```
[root@sudesh network-scripts]# systemctl restart network
```

e. Map your static ip address to your hosts name in configuration file /etc/hosts

```
[root@sudesh /]# vi /etc/hosts
[root@sudesh /]# systemctl restart network
```

TASK 3

a. Write brief history of Linux.

Linux is a free and open-source operating system created by Linus Torvalds in 1991. It was made as an alternative to proprietary operating systems like Windows and macOS. Linux is known for its stability, security, and flexibility, and is widely used in many industries. It is available in many different versions called distributions, and the Linux community continues to contribute to its development.

b. Describe briefly about the following Linux Filesystems Hierarchy: /boot, /root, /user, /home, /usr/bin, /bin, /user/sbin, /sbin, /usr/lib64, /lib64, /usr/lib, /lib, /dev, /etc, /media, /mnt, /opt, /proc, /tmp, /var, /run

- /boot: Contains files needed for the boot process, including the Linux kernel and bootloader.
- /root: Home directory for the root user, which is the system administrator.
- /user: A directory that typically contains subdirectories for user-specific data and programs.
- /home: Home directories for regular users on the system.
- /usr/bin: Contains executable files for system-wide use.
- /bin: Contains essential executable files required for booting and system maintenance.
- /usr/sbin: Contains system administrator executables that are not essential for the system boot process.
- /sbin: Contains essential system administrator executables required for system booting and maintenance.
- /usr/lib64: Contains shared libraries for 64-bit applications.
- /lib64: Contains shared libraries for 64-bit essential executables.
- /usr/lib: Contains shared libraries for system-wide use.
- /lib: Contains shared libraries for essential executables.
- /dev: Contains device files for hardware devices.
- /etc: Contains system configuration files.
- /media: Mount point for removable media devices.
- /mnt: Mount point for temporary filesystems.
- /opt: Contains optional software packages.
- /proc: A virtual filesystem that contains system process information.
- /tmp: A directory for temporary files.
- /var: Contains variable files, including logs and other data that changes frequently.
- /run: Contains system runtime data.

c. Login from root user then create folders according to following tree structure.

```
[root@sudesh /]# mkdir training
[root@sudesh /]# cd training
[root@sudesh training]# mkdir {linux,oracle,datacenter}
[root@sudesh training]# cd linux/
[root@sudesh linux]# mkdir {RHCSA,RHCE}
[root@sudesh linux]# cd /training/oracle/
[root@sudesh oracle]# mkdir {OCA,OCP}
[root@sudesh oracle]# cd /training/datacenter/
[root@sudesh datacenter]# mkdir {VSA,VCP}
[root@sudesh datacenter]# tree /training
```

d. **Copy all the files that have .conf filename extensions inside /etc directory to /root/backup directory.**

```
[root@sudesh /]# mkdir /root/backup
[root@sudesh /]# cp /etc/*.conf /root/backup/
[root@sudesh /]# ls /root/backup/
asound.conf                      ld.so.conf       request-key.conf
brltty.conf                      libaudit.conf    resolv.conf
chrony.conf                      libuser.conf     rsyncd.conf
dleyna-server-service.conf       locale.conf      rsyslog.conf
dnsmasq.conf                     logrotate.conf   sestatus.conf
dracut.conf                      man_db.conf      sos.conf
e2fsck.conf                      mke2fs.conf      sudo.conf
fprintd.conf                     mtools.conf      sudo-ldap.conf
```

e. **Create a directory /root/selected then move all files of /root/backup directory that have 'o' or 'a' as the second character of their file name to /root/selected directory.**

```
[root@sudesh /]# mkdir /root/selected
[root@sudesh /]# mv /root/backup/?[oa]* /root/selected/
[root@sudesh /]# ls /root/selected/
host.conf      logrotate.conf  radvd.conf
locale.conf   man db.conf      sos.conf
```

f. **Remove second character with r in path /root/backup.**

```
[root@sudesh /]# rm -f /root/backup/?r*
[root@sudesh /]# ls /root/backup/
asound.conf                      nsswitch.conf
chrony.conf                      numad.conf
dleyna-server-service.conf       oddjobd.conf
dnsmasq.conf                     pbm2ppa.conf
e2fsck.conf                      pnm2ppa.conf
fprintd.conf                     request-key.conf
fuse.conf                        resolv.conf
GeoIP.conf                       rsyncd.conf
idmapd.conf                      rsyslog.conf
ipsec.conf                       sestatus.conf
kdump.conf                       sudo.conf
ksmtuned.conf                    sudo-ldap.conf
ld.so.conf                       sysctl.conf
libaudit.conf                    tcsd.conf
libuser.conf                     updatedb.conf
mke2fs.conf                      usb_modeswitch.conf
mtools.conf                      vconsole.conf
nfs.conf                         wvdial.conf
nfsmount.conf                    yum.conf
```

g. **Remove all files and directories in path /root/backup.**

```
[root@sudesh /]# rm -rf /root/backup/*
[root@sudesh /]# ls /root/backup/
```

TASK 4

a. **Create a user named student.**

```
[root@sudesh /]# passwd student
Changing password for user student.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too
 simplistic/systematic
Retype new password:
passwd: all authentication tokens updated successfully.
```

b. **Login from student user then create files and folders according to following tree structure.**
   [where, d→ directory and f→ file]

```
[root@sudesh ~]# cd d1
[root@sudesh d1]# mkdir {d2,d3,d4}
[root@sudesh d1]# touch f1
[root@sudesh d1]# cd d2
[root@sudesh d2]# mkdir {d5,d6}
[root@sudesh d2]# cd d5
[root@sudesh d5]# touch f2
[root@sudesh d5]# touch f3
[root@sudesh d5]# cd ..
[root@sudesh d2]# cd d6
[root@sudesh d6]# touch {f4,f5}
[root@sudesh d6]# cd..
bash: cd..: command not found...
[root@sudesh d6]# cd ..
[root@sudesh d2]# cd ..
[root@sudesh d1]# cd d3
[root@sudesh d3]# mkdir d8
[root@sudesh d3]# cd ..
[root@sudesh d1]# cd d4
[root@sudesh d4]# mkdir d7
[root@sudesh d4]# cd d7
[root@sudesh d7]# touch f6
[root@sudesh d7]# cd ~
[root@sudesh ~]# tree d1
```

```
d1
├── d2
│   ├── d5
│   │   ├── f2
│   │   └── f3
│   └── d6
│       ├── f4
│       └── f5
├── d3
│   └── d8
├── d4
│   └── d7
│       └── f6
└── f1
```

c. **Change the permission of the file f1 so that the owner will get full permission, group**
   member will get read and execute permission and others will get read-only permissions.

```
[root@sudesh ~]# cd d1
[root@sudesh d1]# chmod 754 f1
[root@sudesh d1]# ls -l f1
-rwxr-xr--. 1 root_root 0 Apr 10 23:02 f1
```

d. **Change permission of the file f2 such that the owner's and group members will get read**
   and write permission but others will get no permission.

```
-rw-rw----. 1 student student
-rw-rw-r--. 1 student student
```

e. **Change permission of directory d3 such that all categories of users will get full permissions.**

```
drwxr-xr-x. 2 student student
```

TASK 5

a. Create group for each department (production, marketing, sales)

```
[root@sudesh ~]# groupadd production
[root@sudesh ~]# groupadd marketing
[root@sudesh ~]# groupadd sales
```

b. Create user account (user1, user2, user3, user4, user5, user6, manager, boss) for each employee assigning them respective group.

```
[root@sudesh ~]# useradd -G production user1
[root@sudesh ~]# useradd -G production user2
[root@sudesh ~]# useradd -G marketing user3
[root@sudesh ~]# useradd -G marketing user4
[root@sudesh ~]# useradd -G sales user5
[root@sudesh ~]# useradd -G sales user6
[root@sudesh ~]# useradd -G sales,marketing,production,manager
useradd: group 'manager' does not exist
[root@sudesh ~]# useradd -G sales,marketing,production manager
[root@sudesh ~]# useradd boss
```

c. Create common directory (production, marketing and sales) for each department.

```
[root@sudesh ~]# cd /home
[root@sudesh home]# mkdir prod mkt sales
```

d. Change ownership of group directories such that boss will become the owner and the respective groups will be group owner.

```
[root@sudesh home]# chown boss:production prod
[root@sudesh home]# chown boss:sales sales
[root@sudesh home]# chown boss:marketing mkt
```

e. Change the permission of the group directories such that only the owner and group member will get full permission and others will not get any permission.

```
[root@sudesh home]# chmod 770 prod mkt sales
```

TASK 6
Package and Service Management, and Firewall Configuration in CentOS 7:
a) Enable EPEL repository (epel-release package) and verify the repo configuration in
/etc/yum.repos.d

```
[root@sudesh ~]# yum install epel-release -y
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: centos.excellmedia.net
 * extras: centos.excellmedia.net
 * updates: centos.excellmedia.net
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package            Arch         Version       Repository        Size
================================================================================
Installing:
 epel-release       noarch       7-11          extras            15 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 15 k
Installed size: 24 k
Downloading packages:
epel-release-7-11.noarch.rpm                          |  15 kB   00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : epel-release-7-11.noarch                              1/1
  Verifying  : epel-release-7-11.noarch                              1/1

Installed:
  epel-release.noarch 0:7-11

Complete!
```

```
[root@sudesh ~]# cat /etc/yum.repos.d/epel.repo

[epel]
name=Extra Packages for Enterprise Linux 7 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/7/$basearch
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-7&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7

[epel-debuginfo]
name=Extra Packages for Enterprise Linux 7 - $basearch - Debug
#baseurl=http://download.fedoraproject.org/pub/epel/7/$basearch/debug
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-7&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
gpgcheck=1

[epel-source]
name=Extra Packages for Enterprise Linux 7 - $basearch - Source
#baseurl=http://download.fedoraproject.org/pub/epel/7/SRPMS
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-source-7&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
gpgcheck=1
```

b) Install firewalld package as well as start and enable firewall services.

```
[root@sudesh ~]# yum install firewall -y
        Loaded plugins: fastestmirror, langpacks
        Loading mirror speeds from cached hostfile
         * base: centos.excellmedia.net
         * epel: epel.mirror.angkasa.id
         * extras: centos.excellmedia.net
         * updates: centos.excellmedia.net
        Resolving Dependencies
        --> Running transaction check
        ---> Package firewalld.noarch 0:0.6.3-13.el7_9 will be installed
        --> Finished Dependency Resolution

        Dependencies Resolved

        ================================================================
         Package          Arch          Version            Repository      Size
        ================================================================
        Installing:
         firewalld         noarch        0.6.3-13.el7_9     updates         449 k

        Transaction Summary
        ================================================================
        Install  1 Package

        Total download size: 449 k
        Installed size: 1.9 M
        Downloading packages:
        firewalld-0.6.3-13.el7_9.noarch.rpm                | 449 kB   00:00
        Running transaction check
        Running transaction test
        Transaction test succeeded
        Running transaction
          Installing : firewalld-0.6.3-13.el7_9.noarch                       1/1
          Verifying  : firewalld-0.6.3-13.el7_9.noarch                       1/1

        Installed:
          firewalld.noarch 0:0.6.3-13.el7_9

        Complete!
```

c) Install httpd package as well as start and enable httpd services.

```
[root@sudesh ~]# yum install httpd -y
        Loaded plugins: fastestmirror, langpacks
        Loading mirror speeds from cached hostfile
         * base: centos.excellmedia.net
         * epel: epel.mirror.angkasa.id
         * extras: centos.excellmedia.net
         * updates: centos.excellmedia.net
        Resolving Dependencies
        --> Running transaction check
        ---> Package httpd.x86_64 0:2.4.6-98.el7.centos.7 will be installed
        --> Finished Dependency Resolution

        Dependencies Resolved

        ================================================================
         Package          Arch          Version            Repository      Size
        ================================================================
        Installing:
         httpd            x86_64        2.4.6-98.el7.centos.7  updates     2.7 M

        Transaction Summary
        ================================================================
        Install  1 Package

        Total download size: 2.7 M
        Installed size: 9.4 M
        Downloading packages:
        httpd-2.4.6-98.el7.centos.7.x86_64.rpm             | 2.7 MB   00:01
        Running transaction check
        Running transaction test
        Transaction test succeeded
        Running transaction
          Installing : httpd-2.4.6-98.el7.centos.7.x86_64                    1/1
          Verifying  : httpd-2.4.6-98.el7.centos.7.x86_64                    1/1

        Installed:
          httpd.x86_64 0:2.4.6-98.el7.centos.7

        Complete!
```

```
[root@sudesh ~]# systemctl start httpd
[root@sudesh ~]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/
httpd.service to /usr/lib/systemd/system/httpd.service.
[root@sudesh ~]# ▮
```

d) Add the following services and ports to allow packets through the firewall. **[Service = http, smtp port = 25 /tcp, 25/udp, 110/tcp]**

```
firewall-cmd --zone=public --add-service=http --permanent

firewall-cmd --zone=public --add-port=25/tcp --permanent

firewall-cmd --zone=public --add-port=25/udp --permanent

firewall-cmd --zone=public --add-port=110/tcp --permanent

firewall-cmd --reload
```

e) Remove the following services and ports to block packets through the firewall. **[Service = smtp port = 25 /tcp, 25/udp]**

```
firewall-cmd --zone=public --remove-port=25/tcp --permanent

firewall-cmd --zone=public --remove-port=25/udp --permanent

firewall-cmd --reload
```

TASK 7
Bash Shell Scripting:
a) Write a command to find the path of the bash shell.

```
[root@sudesh ~]# which bash
/usr/bin/bash
[root@sudesh ~]#
```

b) Write a script named helloworld.sh to display "Hello World".

```
[root@sudesh ~]# cat > helloworld.sh
echo "Hello World"
^C
[root@sudesh ~]# chmod +x helloworld.sh
[root@sudesh ~]# ./helloworld.sh
Hello World
[root@sudesh ~]#
```

c) Write a script named age.sh to prompt for age and display it.

```
[root@sudesh ~]# cat > age.sh
read -p "Timro umer kati ho?" age
echo "Timro umer $age ho"
^C
[root@sudesh ~]# chmod +x age.sh
[root@sudesh ~]# ./age.sh
Timro umer kati ho? sweet 16
Timro umer sweet 16 ho
[root@sudesh ~]#
```

d) Write a script named guesspass.sh to guess admin password. [password = Redhat123]

```
[root@sudesh ~]# cat > guesspass.sh
read -p "k hola ta password?" pass
if [$pass = Redhat123 ]
then
echo "Bilkul Sahi"
else
echo "Maaf Karna Galat Guess"
fi
^C
[root@sudesh ~]# chmod +x guesspass.sh
[root@sudesh ~]# ./guesspass.sh
k hola ta password? Cabrocks
./guesspass.sh: line 2: [Cabrocks: command not found
Maaf Karna Galat Guess
```

e) Write a script to calculate simple interest.

```
[root@sudesh ~]# cat > interest.sh
read -p "Kitna paisa?" p
read -p "Kitna saal?" t
read -p "Kitna rate?" r
i = $(((p*t*r)/100))
echo "Interest = $i"
^C
[root@sudesh ~]# chmod +x interest.sh
```

f) Write a script to determine the type of tringle by reading the lengths of its sides.

```
[root@sudesh ~]# cat > triangle.sh
read -p "Length_1:"s1
read -p "Length_2:"s2
read -p "Length_3:"s3
if ((s1+s2>s3)) && ((s1+s3>s2)) && ((s2+s3>s1))
then if ["$s1" -eq "$s2" ] && ["$s2" -eq "$s3"]
then
echo "Equilateral Triangle"
elif [ "$s1" -eq "s2" ] || ["$s1" -eq "$s3" ] || ["$s2" -eq "$s3"]
then
echo "Isosceles Triangle"
else
echo "Scalene Triangle"
fi
else
echo"Not a triangle"
fi
^C
[root@sudesh ~]# chmod +x triangle.sh
[root@sudesh ~]# ./triangle.sh

Enter length of side 1: 3
Enter length of side 2: 4
Enter length of side 3: 5
It's a scalene triangle.
```

g) Write a script to determine if a user-inputted number is positive, negative, or Zero.

```
[root@sudesh ~]# cat > number.sh
echo "Enter a number:"
read num
if [ $num -gt 0]
then
echo "The nuber is positive."
else
echo "The number is negative."
fi
^C
[root@sudesh ~]# chmod +x number.sh
[root@sudesh ~]# ./number.sh
Enter a number:

-3
The number is negative.
```

h) Write a script to print the first 10 elements of Fibonacci series.

```
[root@sudesh ~]# cat > fibo.sh
read -p "Enter the number of elements in Fibonacci series:"n
a=0
b=1
i=0
echo "Fibonacci series with $n elements:"
while [ $i - lt $n ]
do
echo
echo -n "$a"
temp = $((a+b))
a = $b
b = $temp
i = $((i+1))
done
^C
[root@sudesh ~]# chmod +x fibo.sh
[root@sudesh ~]# ./fibo.sh

Enter the number of elements in Fibonacci series: 5
Fibonacci series with 5 elements:
0 1 1 2 3
```

i) Create a shell script named "bg.sh" inside /root directory which when execute with parameter 'boy', the output should be 'girl', when execute with the parameter 'girl, the output should be 'boy' & when execute with some other parameter or no parameter the output should be "enter boy or girl only".

```
[root@sudesh ~]# cat > bg.sh
if [ "$1" == "boy" ]; then
echo "girl"
elif [ "$l" == "girl" ]; then
echo "boy"
else
echo "enter boy or girl only"
fi
^C
[root@sudesh ~]# chmod +x bg.sh
[root@sudesh ~]# ./bg.sh boy
 girl
```