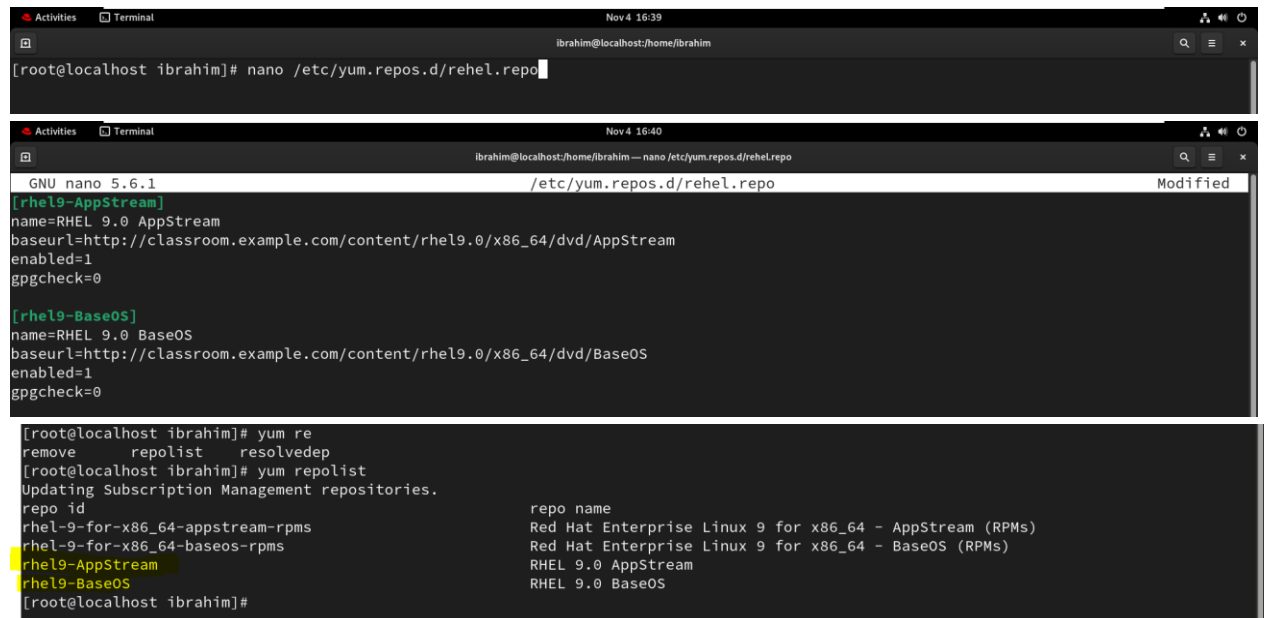


1- Create a repository file.

http://classroom.example.com/content/rhel9.0/x86_64/dvd/AppStream

http://classroom.example.com/content/rhel9.0/x86_64/dvd/BaseOS



```
[root@localhost ibrahim]# nano /etc/yum.repos.d/rhel.repo

GNU nano 5.6.1 /etc/yum.repos.d/rhel.repo Modified
[rhel9-AppStream]
name=RHEL 9.0 AppStream
baseurl=http://classroom.example.com/content/rhel9.0/x86_64/dvd/AppStream
enabled=1
gpgcheck=0

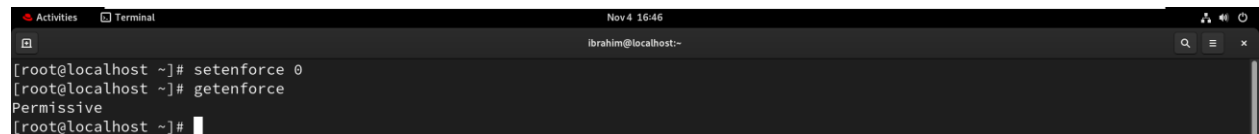
[rhel9-BaseOS]
name=RHEL 9.0 BaseOS
baseurl=http://classroom.example.com/content/rhel9.0/x86_64/dvd/BaseOS
enabled=1
gpgcheck=0

[root@localhost ibrahim]# yum re
remove repolist resolvedep
[root@localhost ibrahim]# yum repolist
Updating Subscription Management repositories.
repo id                                repo name
rhel-9-for-x86_64-appstream-rpms       Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
rhel-9-for-x86_64-baseos-rpms          Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
rhel9-AppStream                        RHEL 9.0 AppStream
rhel9-BaseOS                           RHEL 9.0 BaseOS
[root@localhost ibrahim]#
```

2- Configure the Selinux

- a- The webserver can serve all the existing HTML file located at /var/www/html directory (Don't alter or remove any files in this directory)

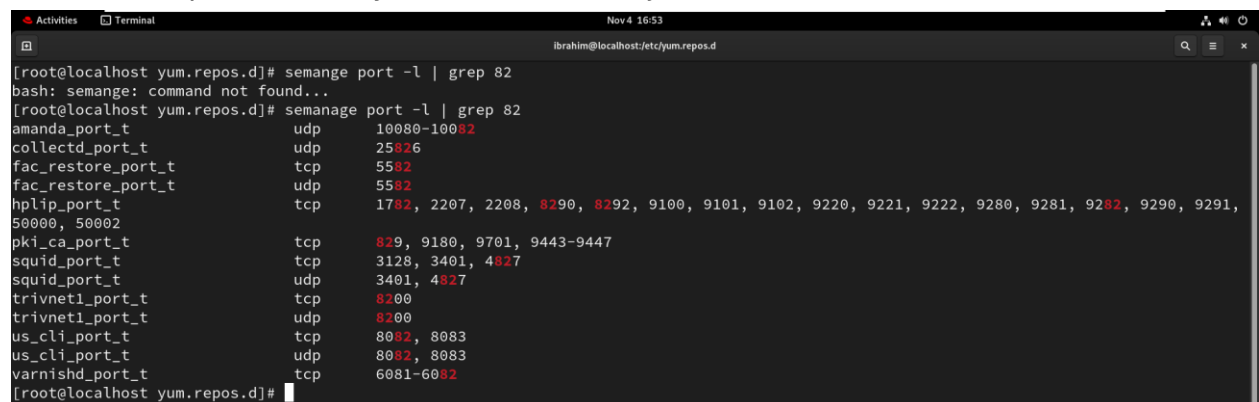
>> In order to achieve this you must set the selinux to the permissive mode



```
[root@localhost ~]# setenforce 0
[root@localhost ~]# getenforce
Permissive
[root@localhost ~]#
```

- b- The webserver can serve the content on port 82.

>> see if this port is used by another selinux object or not



```
[root@localhost yum.repos.d]# semange port -l | grep 82
bash: semange: command not found...
[root@localhost yum.repos.d]# semanage port -l | grep 82
amanda_port_t      udp      10080-10082
collectd_port_t    udp      25826
fac_restore_port_t tcp      5582
fac_restore_port_t udp      5582
hplip_port_t       tcp      1782, 2207, 2208, 8290, 8292, 9100, 9101, 9102, 9220, 9221, 9222, 9280, 9281, 9282, 9290, 9291, 50000, 50002
pki_ca_port_t      tcp      829, 9180, 9701, 9443-9447
squid_port_t       tcp      3128, 3401, 4827
squid_port_t       udp      3401, 4827
trivnet1_port_t    tcp      8200
trivnet1_port_t    udp      8200
us_cli_port_t      tcp      8082, 8083
us_cli_port_t      udp      8082, 8083
varnishd_port_t    tcp      6081-6082
[root@localhost yum.repos.d]#
```

>> As you can see the port 82 is not In use by any selinux object so that we can assigned it to the httpd and it to the selinux database to avoid errors

```
ibrahim@localhost: /var/www/html
[ibrahim@localhost html]# semanage port -a -t http_port_t -p tcp 82
[ibrahim@localhost html]# semanage port -l | grep 82
amanda_port_t      udp      10080-10082
collectd_port_t    udp      25826
fac_restore_port_t tcp      5582
fac_restore_port_t udp      5582
hplip_port_t       tcp      1782, 2207, 2208, 8290, 8292, 9100, 9101, 9102, 9220, 9221, 9222, 9280, 9281, 9282, 9290, 9291, 50000, 50002
http_port_t        tcp      82, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pki_ca_port_t      tcp      829, 9180, 9701, 9443-9447
squid_port_t       tcp      3128, 3401, 4827
squid_port_t       udp      3401, 4827
trivnet1_port_t    tcp      8200
trivnet1_port_t    udp      8200
us_cli_port_t      tcp      8082, 8083
us_cli_port_t      udp      8082, 8083
varnishd_port_t    tcp      6081-6082
[ibrahim@localhost html]#
```

>>Now I am going to change the default port in the http configurations file to 82

```
ibrahim@localhost: /etc/httpd/conf -- nano httpd.conf
GNU nano 5.6.1 httpd.conf Modified
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
#Listen 80
Listen 82
```

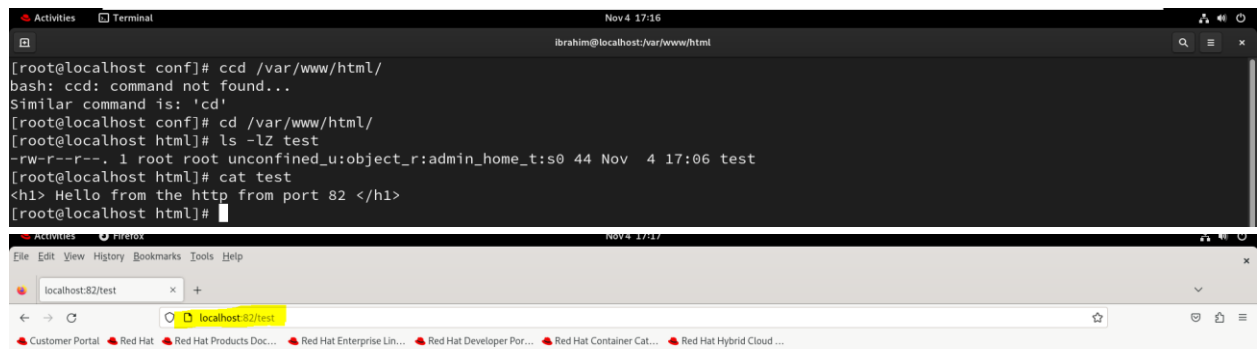
>> Restart the httpd service

```
ibrahim@localhost: /etc/httpd/conf -- systemctl status httpd
[ibrahim@localhost conf]# systemctl restart httpd
[ibrahim@localhost conf]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Mon 2024-11-04 17:14:30 EET; 7s ago
     Docs: man:httpd.service(8)
   Main PID: 4421 (httpd)
    Status: "Started, listening on: port 82"
   Tasks: 177 (limit: 24452)
  Memory: 28.0M
    CPU: 226ms
```

>> Add this port in the firewall to allow the incoming traffic on that port

```
ibrahim@localhost: /var/www/html
[ibrahim@localhost html]# firewall-cmd --add-p
--add-port=      --add-protocol=
[ibrahim@localhost html]# firewall-cmd --permanent --add-port=82/tcp
success
[ibrahim@localhost html]# firewall-cmd --reload
success
[ibrahim@localhost html]#
```

>> check if everything is running



The screenshot shows two windows. The top window is a terminal with the following commands and output:

```
[root@localhost conf]# ccd /var/www/html/
bash: ccd: command not found...
Similar command is: 'cd'
[root@localhost conf]# cd /var/www/html/
[root@localhost html]# ls -lZ test
-rw-r--r--. 1 root root unconfined_u:object_r:admin_home_t:s0 44 Nov  4 17:06 test
[root@localhost html]# cat test
<h1> Hello from the http from port 82 </h1>
[root@localhost html]#
```

The bottom window is a Firefox browser showing the URL `localhost:82/test` with the content `Hello from the http from port 82`.

Hello from the http from port 82

3- Configure autofs to automount the home directories of production5 domain users.

>> Assume that the home directories of production5 domain users are exist on a directory called `/user-homes/production5` on the server.

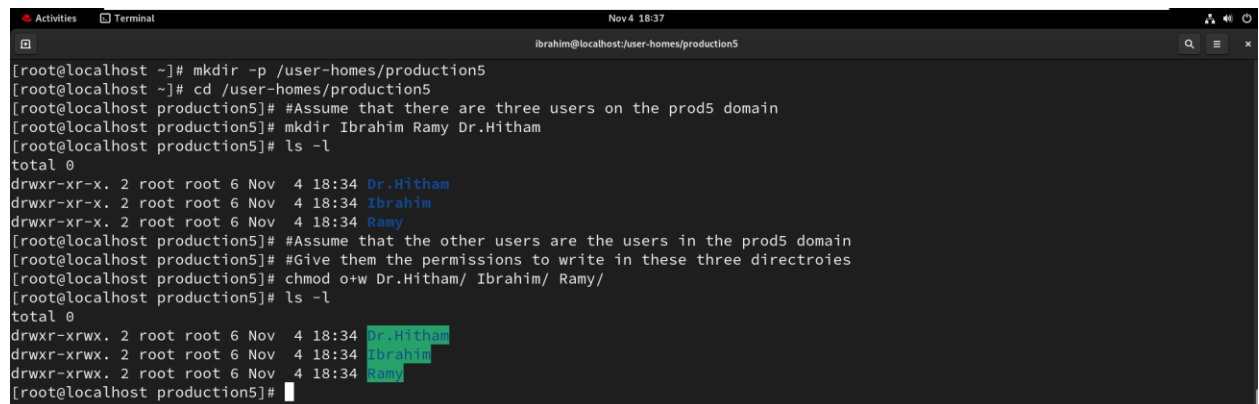
>> In this case (server shares multiple entries under a directory) we must use the indirect map.

>> I have a two vms configured to be an NFS-server and NFS-client which are:

RHEL 9.4 as a server

Ubuntu as a client

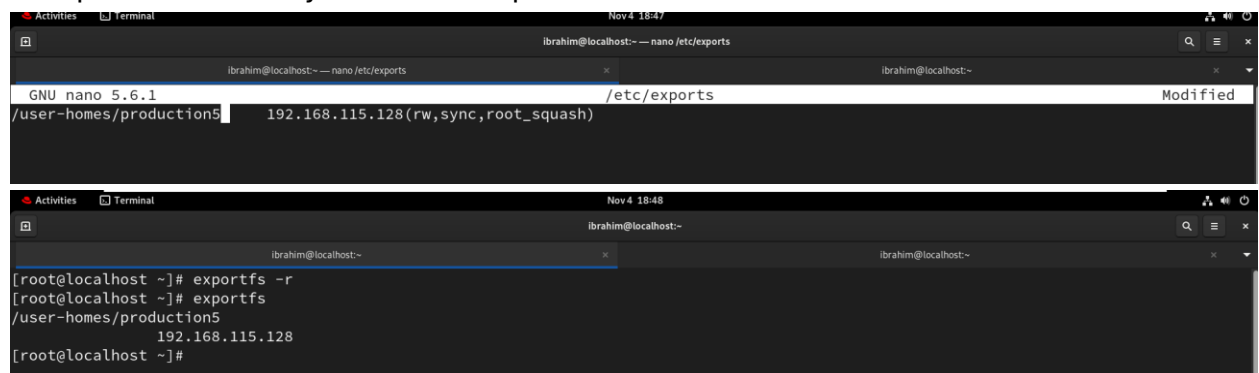
Start with the server-side configurations:



The screenshot shows a terminal window with the following commands and output:

```
[root@localhost ~]# mkdir -p /user-homes/production5
[root@localhost ~]# cd /user-homes/production5
[root@localhost production5]# #Assume that there are three users on the prod5 domain
[root@localhost production5]# mkdir Ibrahim Ramy Dr.Hitham
[root@localhost production5]# ls -l
total 0
drwxr-xr-x. 2 root root 6 Nov  4 18:34 Dr.Hitham
drwxr-xr-x. 2 root root 6 Nov  4 18:34 Ibrahim
drwxr-xr-x. 2 root root 6 Nov  4 18:34 Ramy
[root@localhost production5]# #Assume that the other users are the users in the prod5 domain
[root@localhost production5]# #Give them the permissions to write in these three directroies
[root@localhost production5]# chmod o+w Dr.Hitham/ Ibrahim/ Ramy/
[root@localhost production5]# ls -l
total 0
drwxr-xrwx. 2 root root 6 Nov  4 18:34 Dr.Hitham
drwxr-xrwx. 2 root root 6 Nov  4 18:34 Ibrahim
drwxr-xrwx. 2 root root 6 Nov  4 18:34 Ramy
[root@localhost production5]#
```

>> Export this directory `/user-homes/production5/`



The screenshot shows two terminal windows. The top window shows the `/etc/exports` file being edited in nano:

```
GNU nano 5.6.1 /etc/exports
/user-homes/production5 192.168.115.128(rw,sync,root_squash)
```

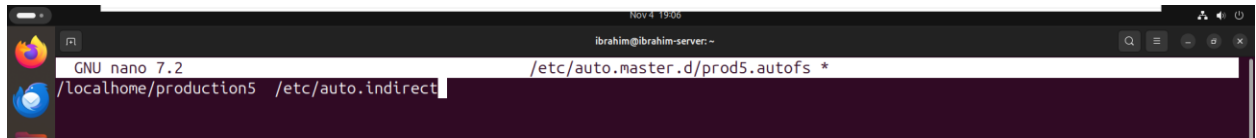
The bottom window shows the following commands and output:

```
[root@localhost ~]# exportfs -r
[root@localhost ~]# exportfs
/user-homes/production5
192.168.115.128
[root@localhost ~]#
```

>> Now configuration of the client-side:

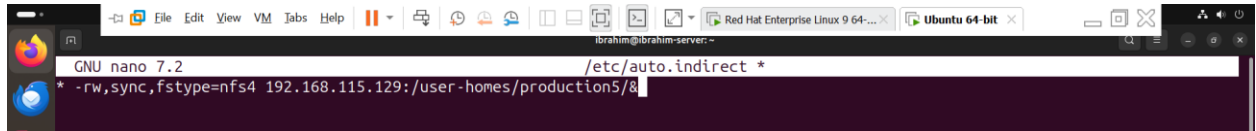
>> the autofs service is already installed

>> Go to the /etc/auto.master.d/ directory and create a master file



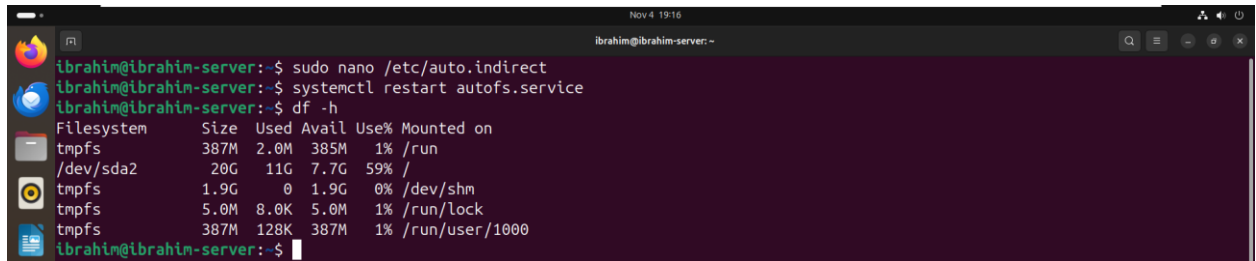
```
ibrahim@ibrahim-server:~$ nano /etc/auto.indirect
GNU nano 7.2 /etc/auto.indirect *
/localhost/production5 /etc/auto.indirect
```

>> base directory as shown and the mount details will be in the mapping file called auto.indirect under /etc directory



```
ibrahim@ibrahim-server:~$ nano /etc/auto.indirect
GNU nano 7.2 /etc/auto.indirect *
* -rw,sync,fstype=nfs4 192.168.115.129:/user-homes/production5/&
```

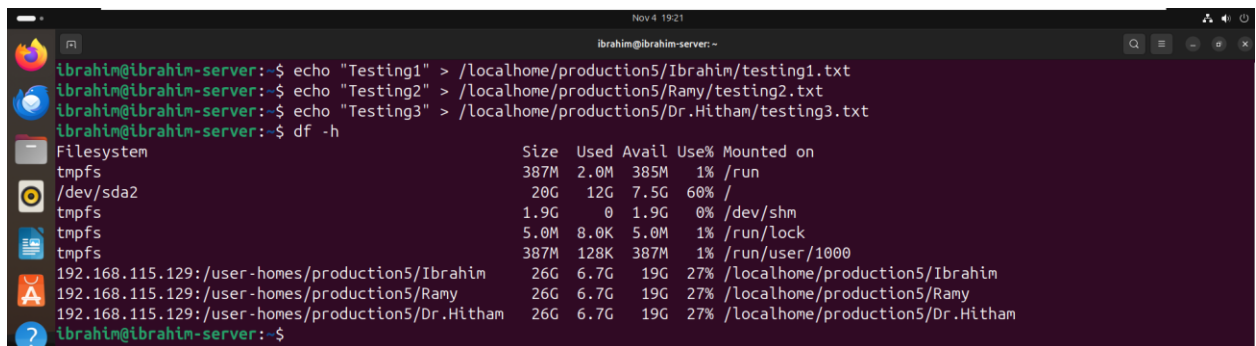
>> This line tell the autofs that mount any subdirectory under the /user-homes/production5/ on the client with the same name on the server under the /localhome/production5 directory.



```
ibrahim@ibrahim-server:~$ sudo nano /etc/auto.indirect
ibrahim@ibrahim-server:~$ systemctl restart autofs.service
ibrahim@ibrahim-server:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            387M  2.0M  385M   1% /run
/dev/sda2        20G   11G   7.7G  59% /
tmpfs            1.9G   0 1.9G   0% /dev/shm
tmpfs            5.0M  8.0K  5.0M   1% /run/lock
tmpfs            387M  128K  387M   1% /run/user/1000
```

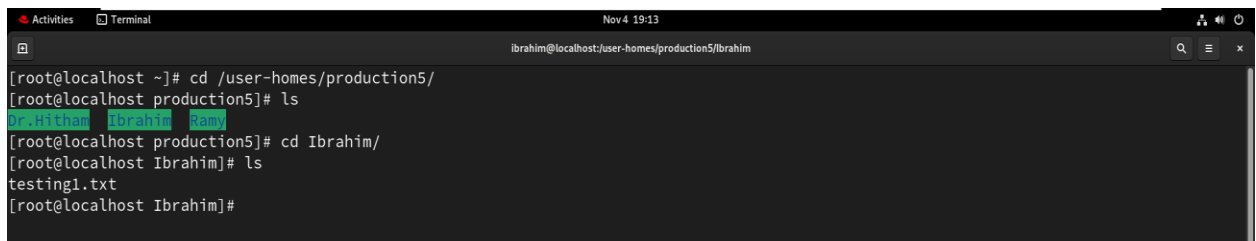
>> there is no entries for the needed directories but once you used or created a file in these directories it will appear in the df -h command after that by default if you did not these directories for 5min it will disappear again.

>> let's validate



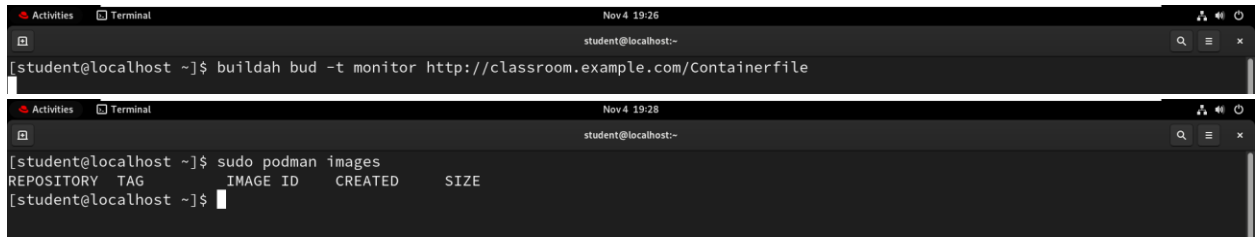
```
ibrahim@ibrahim-server:~$ echo "Testing1" > /localhome/production5/Ibrahim/testing1.txt
ibrahim@ibrahim-server:~$ echo "Testing2" > /localhome/production5/Ramy/testing2.txt
ibrahim@ibrahim-server:~$ echo "Testing3" > /localhome/production5/Dr.Hitham/testing3.txt
ibrahim@ibrahim-server:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            387M  2.0M  385M   1% /run
/dev/sda2        20G   12G   7.5G  60% /
tmpfs            1.9G   0 1.9G   0% /dev/shm
tmpfs            5.0M  8.0K  5.0M   1% /run/lock
tmpfs            387M  128K  387M   1% /run/user/1000
192.168.115.129:/user-homes/production5/Ibrahim  26G   6.7G   19G  27% /localhome/production5/Ibrahim
192.168.115.129:/user-homes/production5/Ramy    26G   6.7G   19G  27% /localhome/production5/Ramy
192.168.115.129:/user-homes/production5/Dr.Hitham 26G   6.7G   19G  27% /localhome/production5/Dr.Hitham
```

>> see the reflection on the server-side



```
[root@localhost ~]# cd /user-homes/production5/
[root@localhost production5]# ls
Dr.Hitham Ibrahim Ramy
[root@localhost production5]# cd Ibrahim/
[root@localhost Ibrahim]# ls
testing1.txt
[root@localhost Ibrahim]#
```

4- Build a container as student user



The image shows two terminal windows. The top window shows the command `buildah bud -t monitor http://classroom.example.com/Containerfile` being executed. The bottom window shows the command `sudo podman images` being executed, resulting in a table of images.

```
student@localhost ~]$ buildah bud -t monitor http://classroom.example.com/Containerfile

student@localhost ~]$ sudo podman images
REPOSITORY TAG IMAGE ID CREATED SIZE
student@localhost ~]$
```

>> in normal case you should find that an image with the name of monitor exist

5- Configure the container(monitor) as a system start-up service and mount volumes persistently

>> In this example I am going to write only the steps as I don not have a valid link for the container

1) So After I built the image from the containerfile then I need to run the container

>> `podman run -d --name ascii2pdf -v /opt/files:/opt/incomingdirectory -v /opt/processed:/opt/outgoingdirectory monitor`

2) Create a service file named `container-ascii2pdf.service`

>>`sudo nano /etc/systemd/system/container-ascii2pdf.service`

[Unit]

Description=ASCII to PDF Converter Container

After=network.target

[Service]

Restart=always

ExecStart=/usr/bin/podman start -a ascii2pdf

ExecStop=/usr/bin/podman stop ascii2pdf

[Install]

WantedBy=multi-user.target

>> Save and exist

3) Enable the service to start on boot with the following command

>> `sudo systemctl enable container-ascii2pdf`

4) Start the service

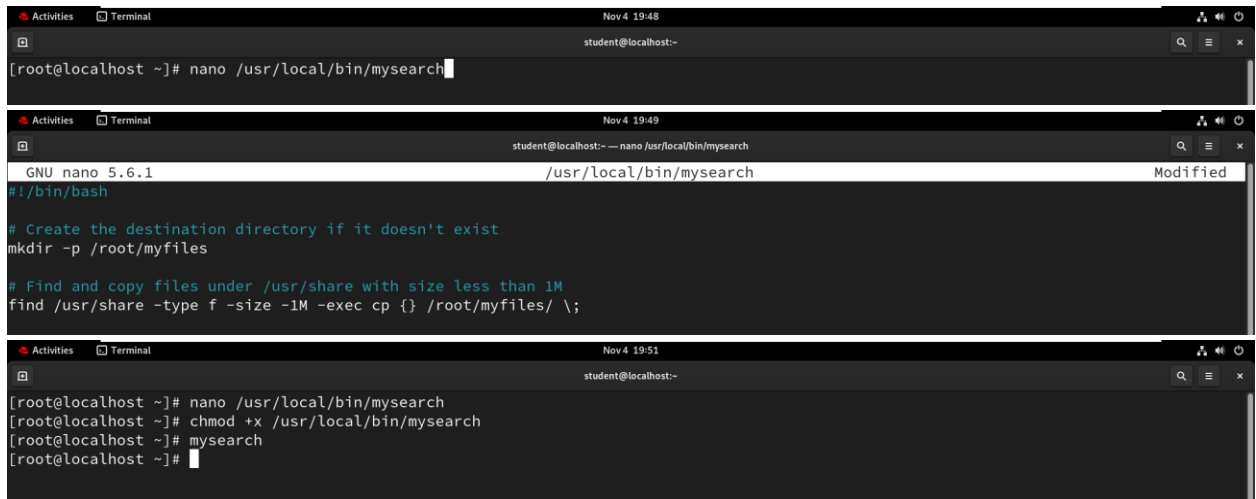
>>`sudo systemctl start container-ascii2pdf`

5) Check the Status

>>`sudo systemctl status container-ascii2pdf`

6) Test the Functionality: Place a file in the `/opt/files` directory and check that it processes into a PDF in `/opt/processed`.

6- Create the script file



The first screenshot shows the terminal at Nov 4 19:48 with the command `[root@localhost ~]# nano /usr/local/bin/mysearch`. The second screenshot, at Nov 4 19:49, shows the nano editor with the following content: `#!/bin/bash`, `# Create the destination directory if it doesn't exist`, `mkdir -p /root/myfiles`, `# Find and copy files under /usr/share with size less than 1M`, and `find /usr/share -type f -size -1M -exec cp {} /root/myfiles/ \;`. The third screenshot, at Nov 4 19:51, shows the terminal after execution: `[root@localhost ~]# nano /usr/local/bin/mysearch`, `[root@localhost ~]# chmod +x /usr/local/bin/mysearch`, `[root@localhost ~]# mysearch`, and `[root@localhost ~]#`.

```
[root@localhost ~]# nano /usr/local/bin/mysearch
```

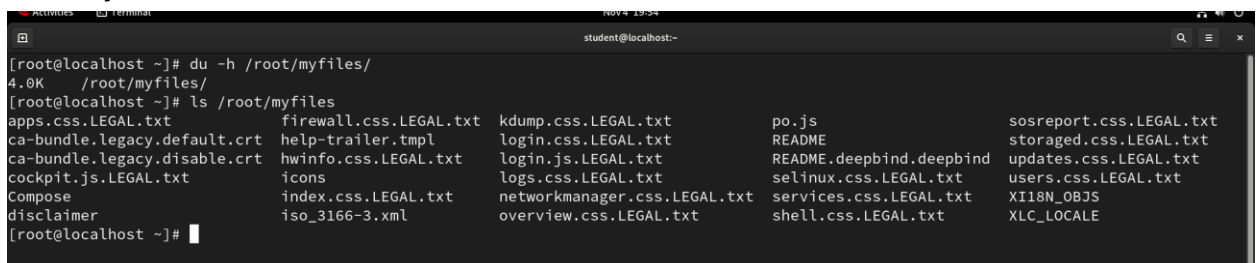
```
GNU nano 5.6.1 /usr/local/bin/mysearch Modified
#!/bin/bash

# Create the destination directory if it doesn't exist
mkdir -p /root/myfiles

# Find and copy files under /usr/share with size less than 1M
find /usr/share -type f -size -1M -exec cp {} /root/myfiles/ \;
```

```
[root@localhost ~]# nano /usr/local/bin/mysearch
[root@localhost ~]# chmod +x /usr/local/bin/mysearch
[root@localhost ~]# mysearch
[root@localhost ~]#
```

>>Verify the results



The screenshot, taken at Nov 4 19:54, shows the terminal output of the script. It displays the directory size and a list of files copied to /root/myfiles.

```
[root@localhost ~]# du -h /root/myfiles/
4.0K /root/myfiles/
[root@localhost ~]# ls /root/myfiles
apps.css.LEGAL.txt      firewall.css.LEGAL.txt  kdump.css.LEGAL.txt    po.js                  sosreport.css.LEGAL.txt
ca-bundle.legacy.default.crt  help-trailer.tmpl      login.css.LEGAL.txt     README                 stored.css.LEGAL.txt
ca-bundle.legacy.disable.crt  hwinfo.css.LEGAL.txt   login.js.LEGAL.txt      README.deepbind.deepbind  updates.css.LEGAL.txt
cockpit.js.LEGAL.txt          icons                  logs.css.LEGAL.txt      selinux.css.LEGAL.txt    users.css.LEGAL.txt
Compose                     index.css.LEGAL.txt    networkmanager.css.LEGAL.txt  services.css.LEGAL.txt  XI18N_OBJS
disclaimer                  iso_3166-3.xml          overview.css.LEGAL.txt    shell.css.LEGAL.txt      XLC_LOCALE
[root@localhost ~]#
```