1- Create a repository file.

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

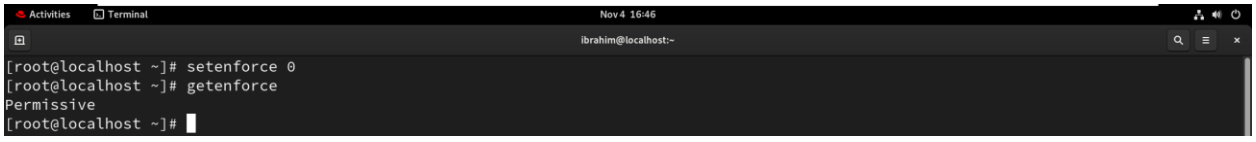
http://classroom.example.com/content/rhel9.0/x86_64/dvd/BaseOS


```
[root@localhost ibrahim]# 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

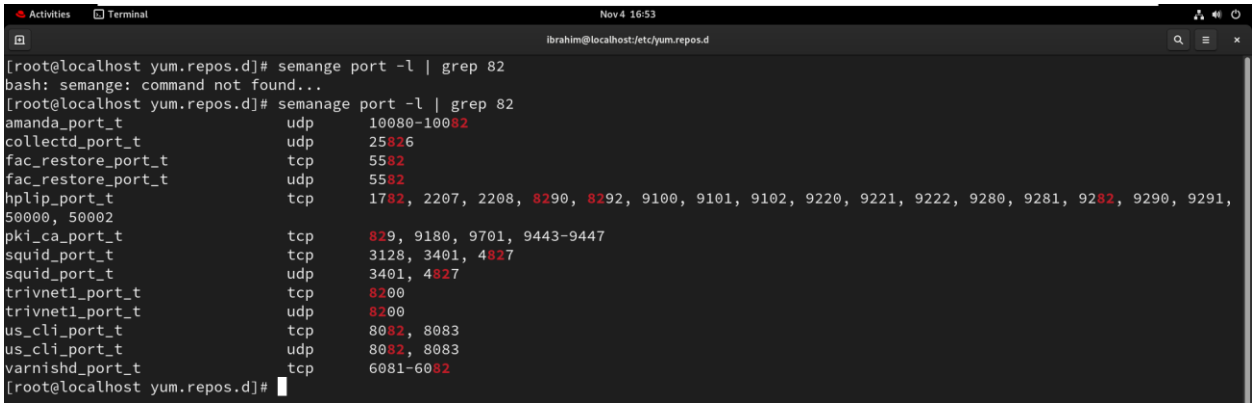
a- The webserver can serve all the existing HTML file located at /var/www/html directory (Don't alter or remove any files in this directory)

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



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

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



>> 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

```
[root@localhost html]# semanage port -a -t http_port_t  -p tcp 82
[root@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
[root@localhost html]#
```

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

```
 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
```

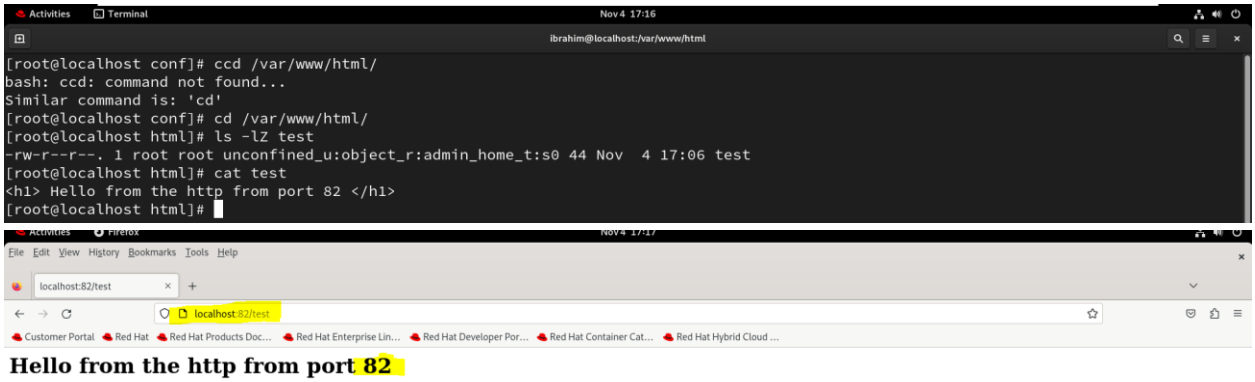
>> Restart the httpd service

```
[root@localhost conf]# systemctl restart httpd
[root@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

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

>> check if everything is running



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.

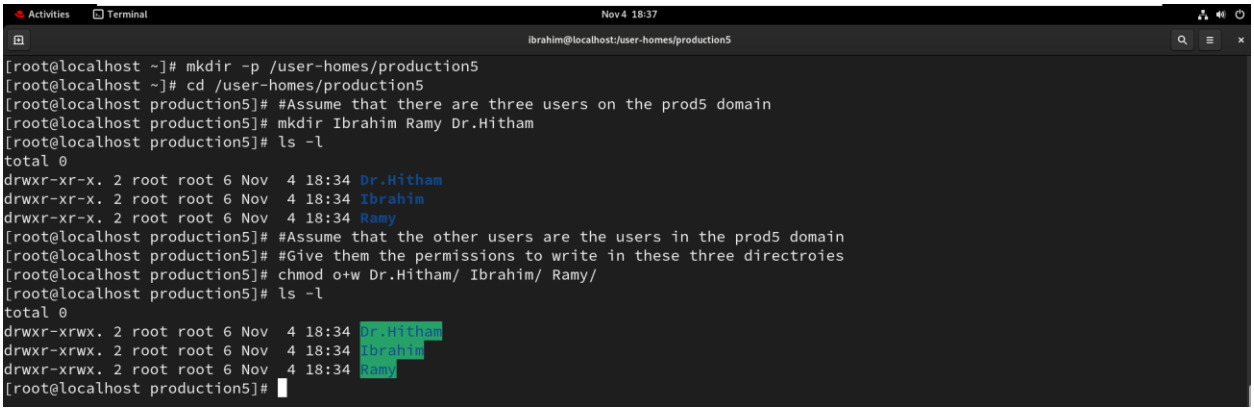
>>In this case (server shares multiple entries under a directory) we must use the indirect map.

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

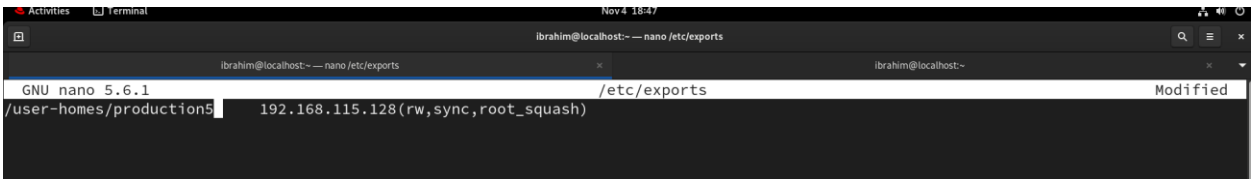
RHEL 9.4 as a server

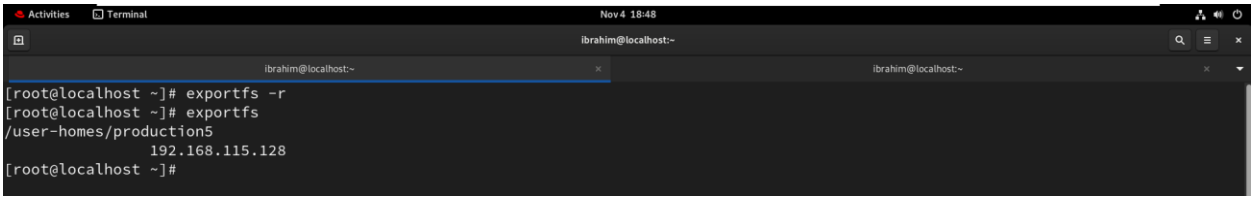
Ubuntu as a client

Start with the server-side configurations:



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

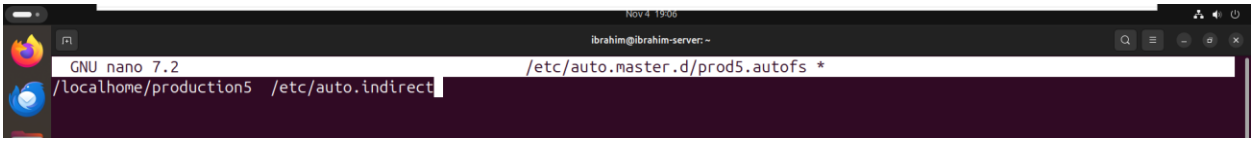




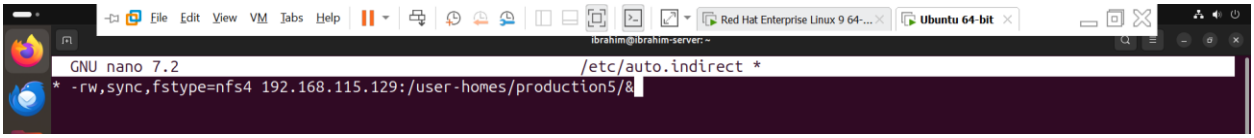
>> Now configuration of the client-side:

>> the autofs service is already installed

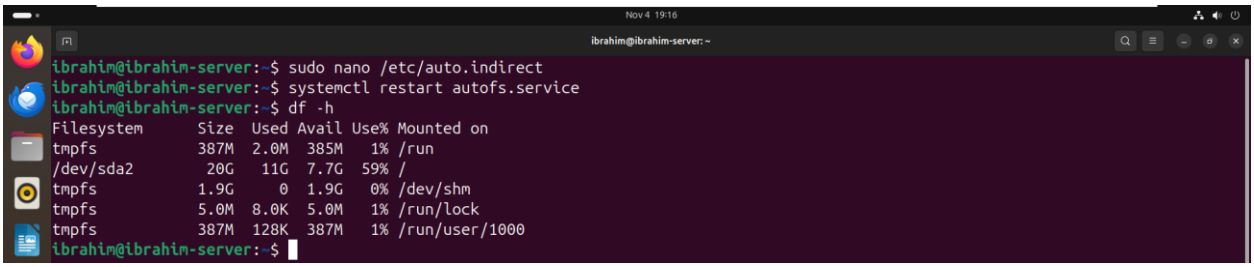
>>Go to the /etc/auto.master.d/ directory and create a master file



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

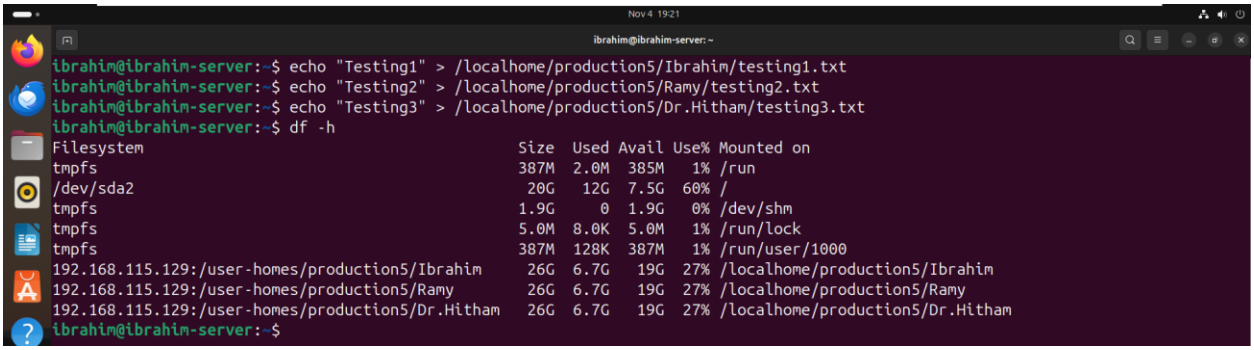


>> This line tell the autofs that mount any subdirectory under the /user-homes/production5/ on the client with the same name on the server under the /localhome/production5 directory.



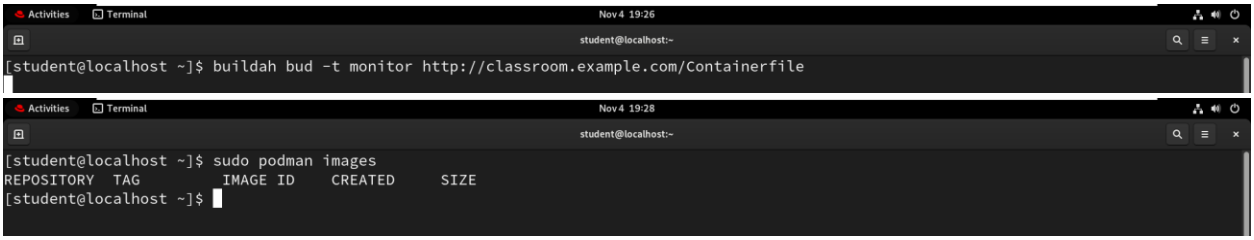
>> there is no entries for the needed directories but once you used or created a file in these directories it will appear in the df -h command after thar by default if you did not these directories for 5min it will disappear again.

>> let's validate



>> see the reflection on the server-side

4- Build a container as student user



>> 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

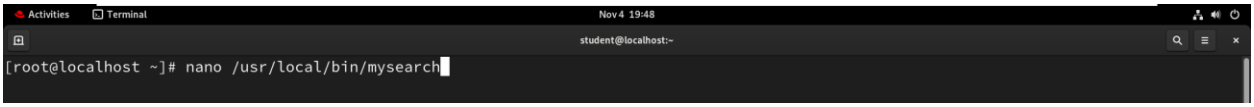
1) So After I built the image from the containerfile then I need to run the container
   >> podman run -d --name ascii2pdf -v /opt/files:/opt/incomingdirectory -v /opt/processed:/opt/outgoingdirectory monitor
2) Create a service file named container-ascii2pdf.service
   >>sudo nano /etc/systemd/system/container-ascii2pdf.service
   [Unit]
   Description=ASCII to PDF Converter Container
   After=network.target

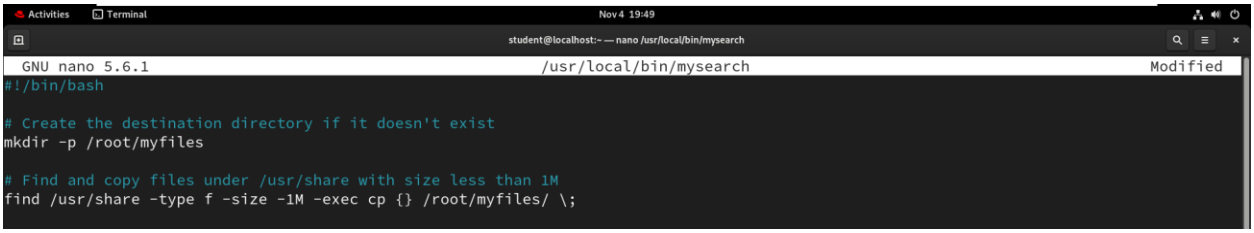
   [Service]
   Restart=always
   ExecStart=/usr/bin/podman start -a ascii2pdf
   ExecStop=/usr/bin/podman stop ascii2pdf

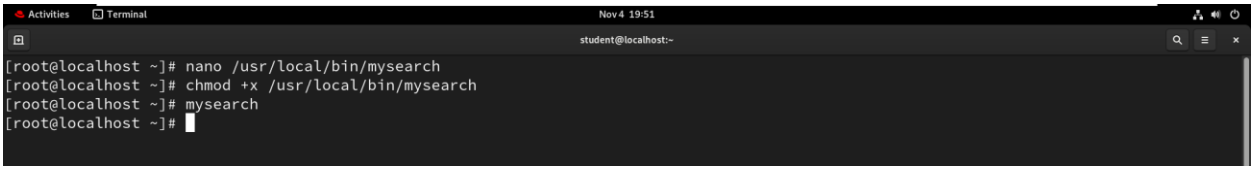
   [Install]
   WantedBy=multi-user.target

   >> 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







>>Verify the results