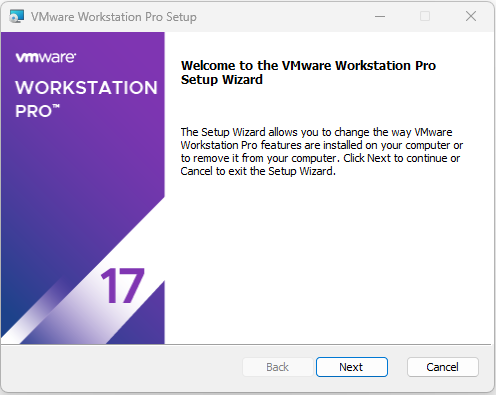
# **Task 5**

## **Cent OS 7 (server/client) installation over VMware:**

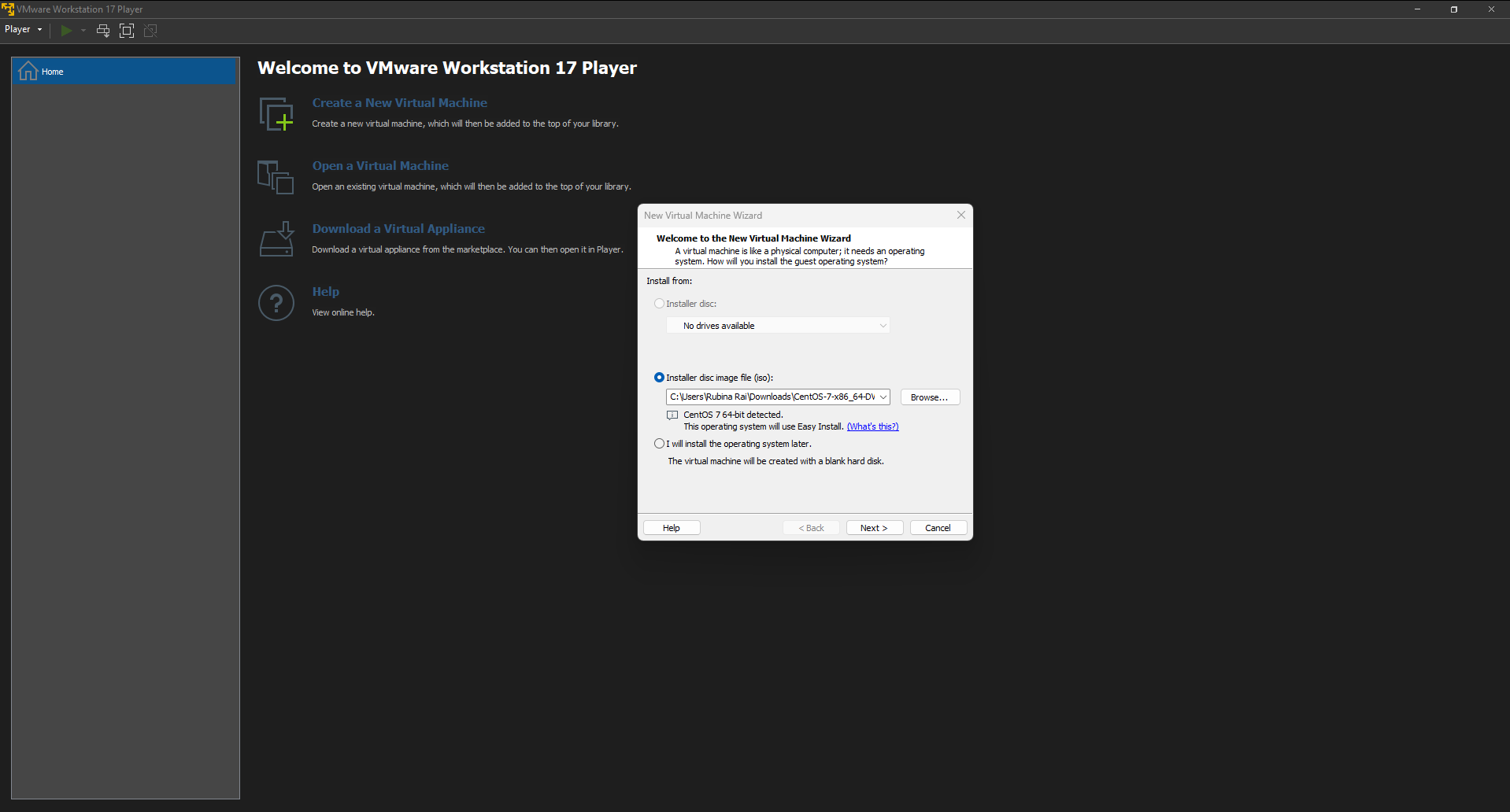
1. **Installing VMware Workstation**  
   To begin the installation of VMware Workstation on my device, I first visited the official Broadcom website. After accessing the page, I registered for an account, which involved providing basic information like my email address and creating a secure password. Once my account was successfully created, I logged in to access the VMware Workstation download page.  
   A screenshot of a computer

   AI-generated content may be incorrect.  
     
   From there, I selected the appropriate version for my operating system and began the download process. After the file finished downloading, I followed the on-screen instructions to complete the installation of VMware Workstation on my system.  
     
   
2. **Creating Virtual Machine of CentOS 7**  
   After successfully installing VMware Workstation, I proceeded to create a new virtual machine (VM) for CentOS 7.
   1. First, I downloaded the ISO file for CentOS 7 from the vault of the official CentOS website.

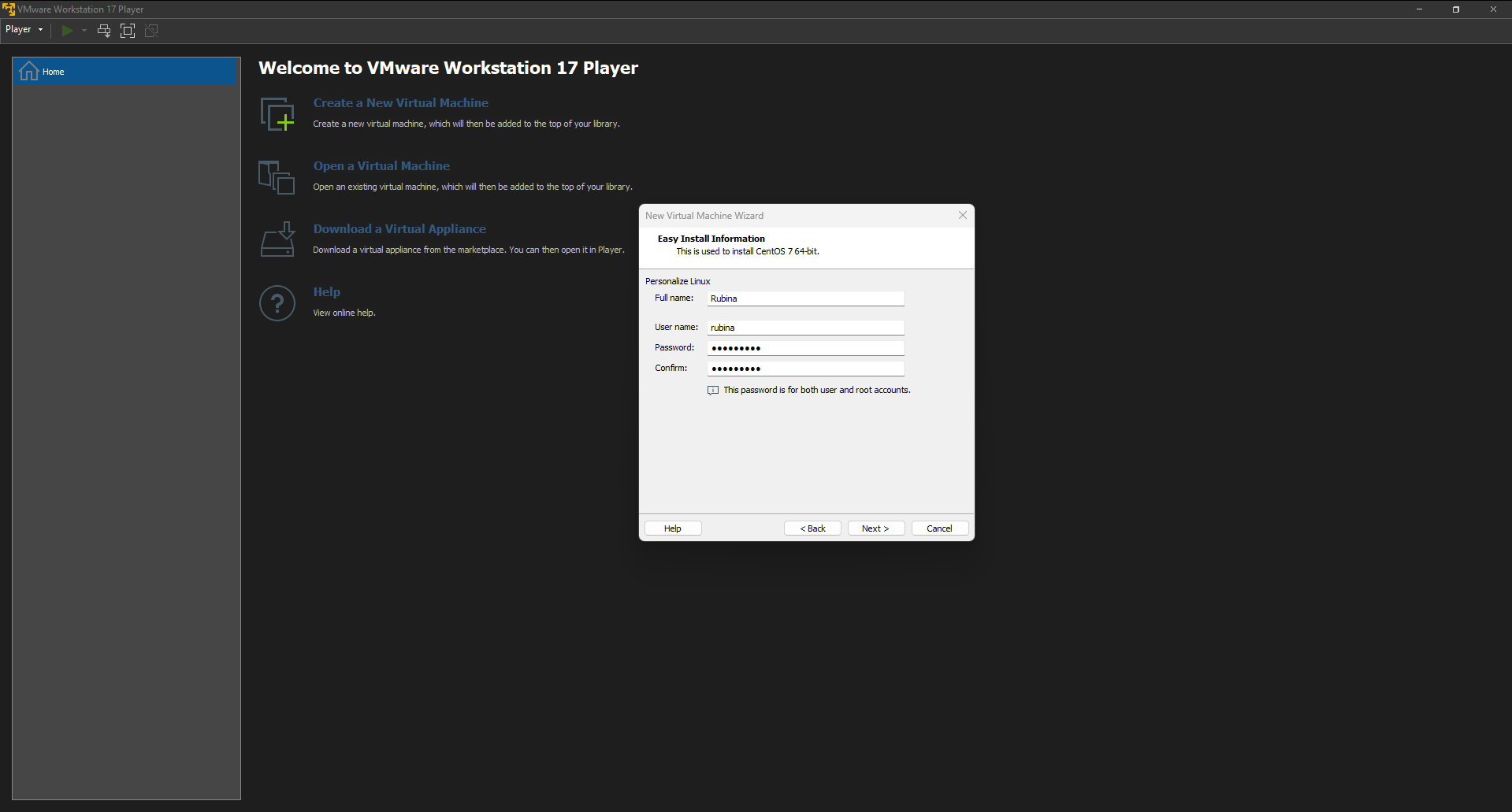
A screenshot of a computer

AI-generated content may be incorrect.

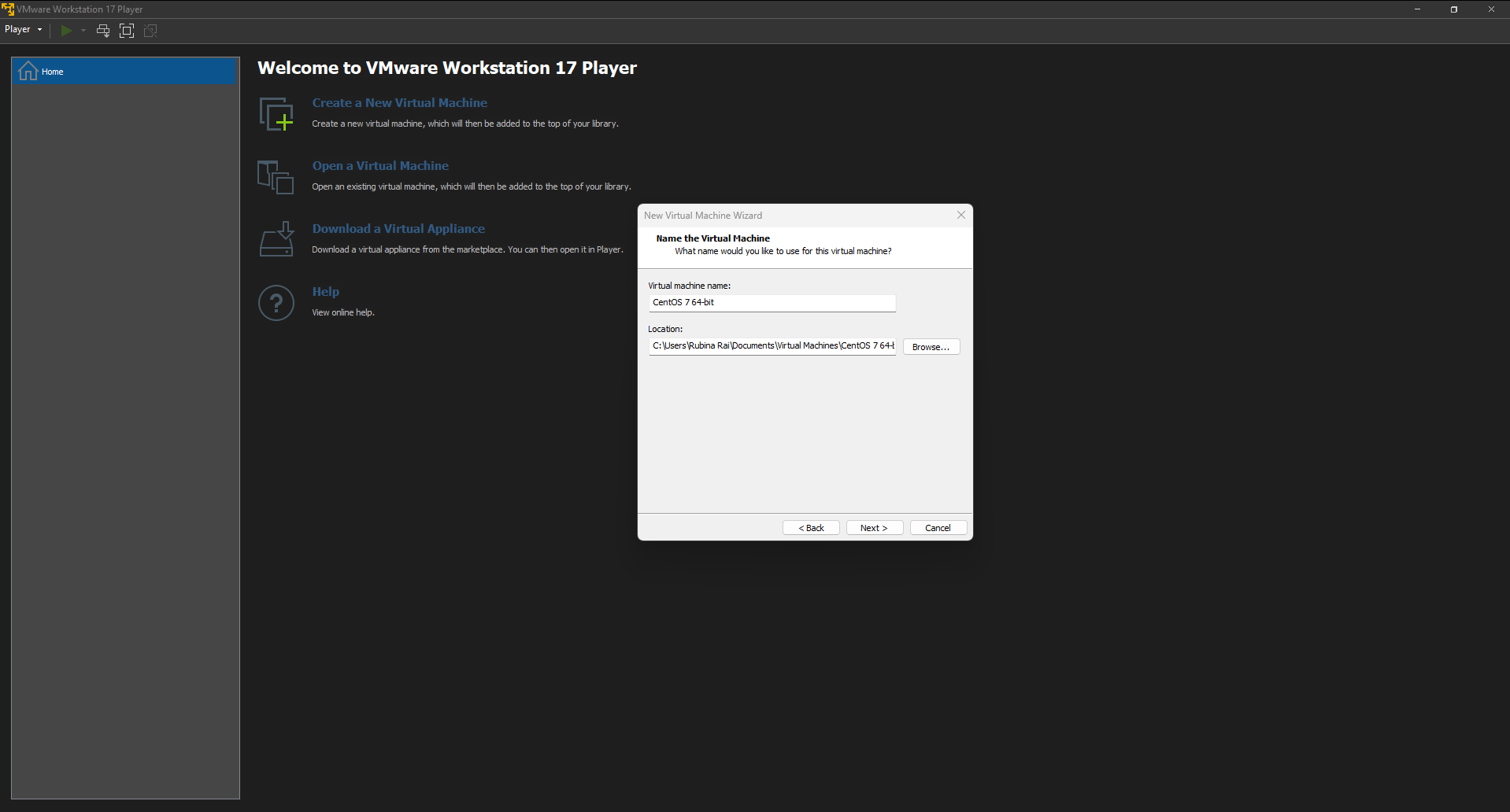
* 1. I started VMware Workstation application and selected the “Create a New Virtual Machine” option and then I selected the CentOS 7 ISO file that I had previously downloaded as the disc image file.



* 1. Then I named the virtual machine and entered information of root user account to provide administrative access to the system.



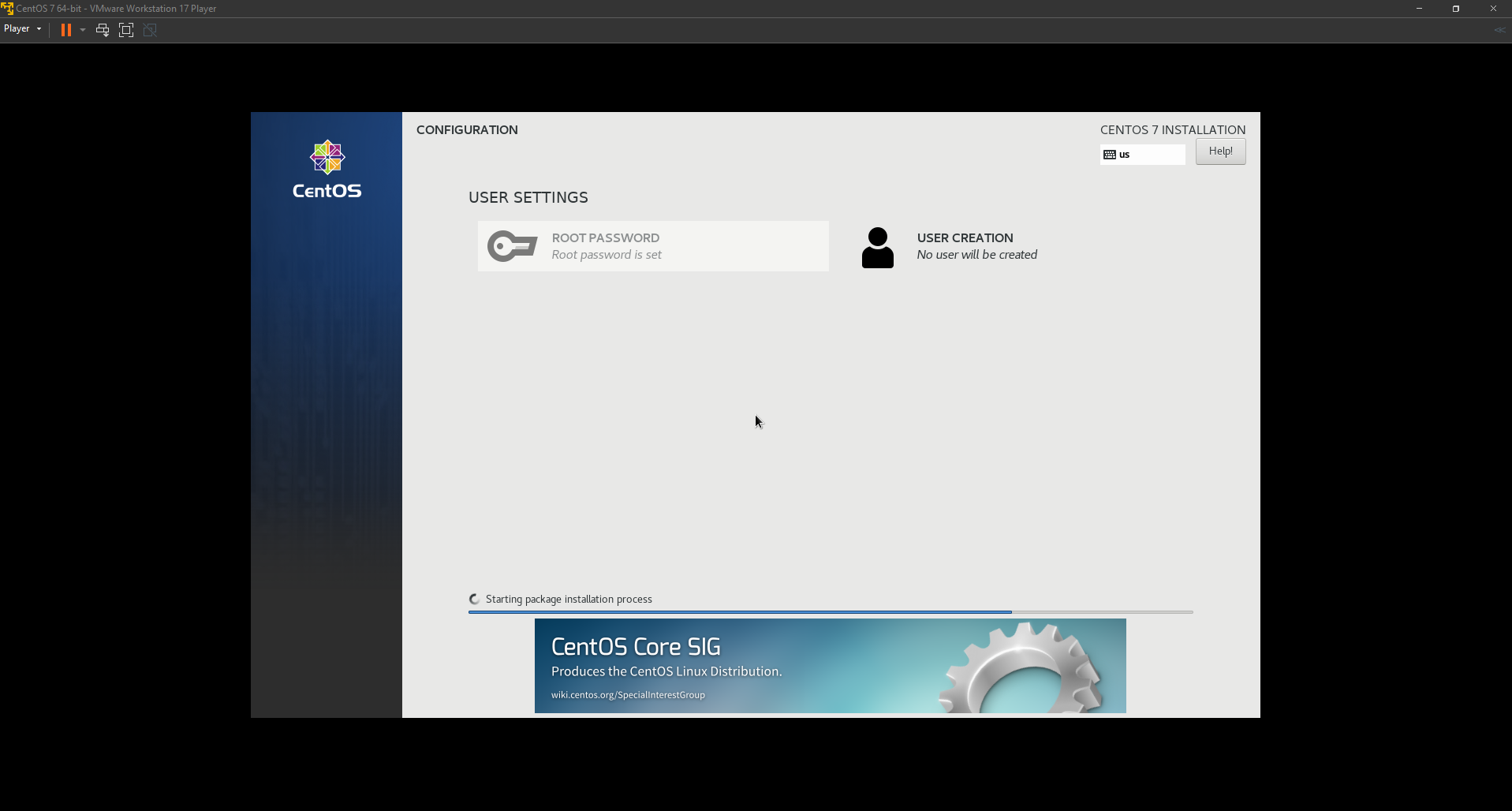
* 1. After which I entered the name of the virtual machine and selected the location where its file will be stored.



* 1. For the final step of the setup, I entered the disk size of the virtual machine and completed the setup.



* 1. After the setup completed, I powered on the virtual machine and the installation of the necessary packages automatically started.

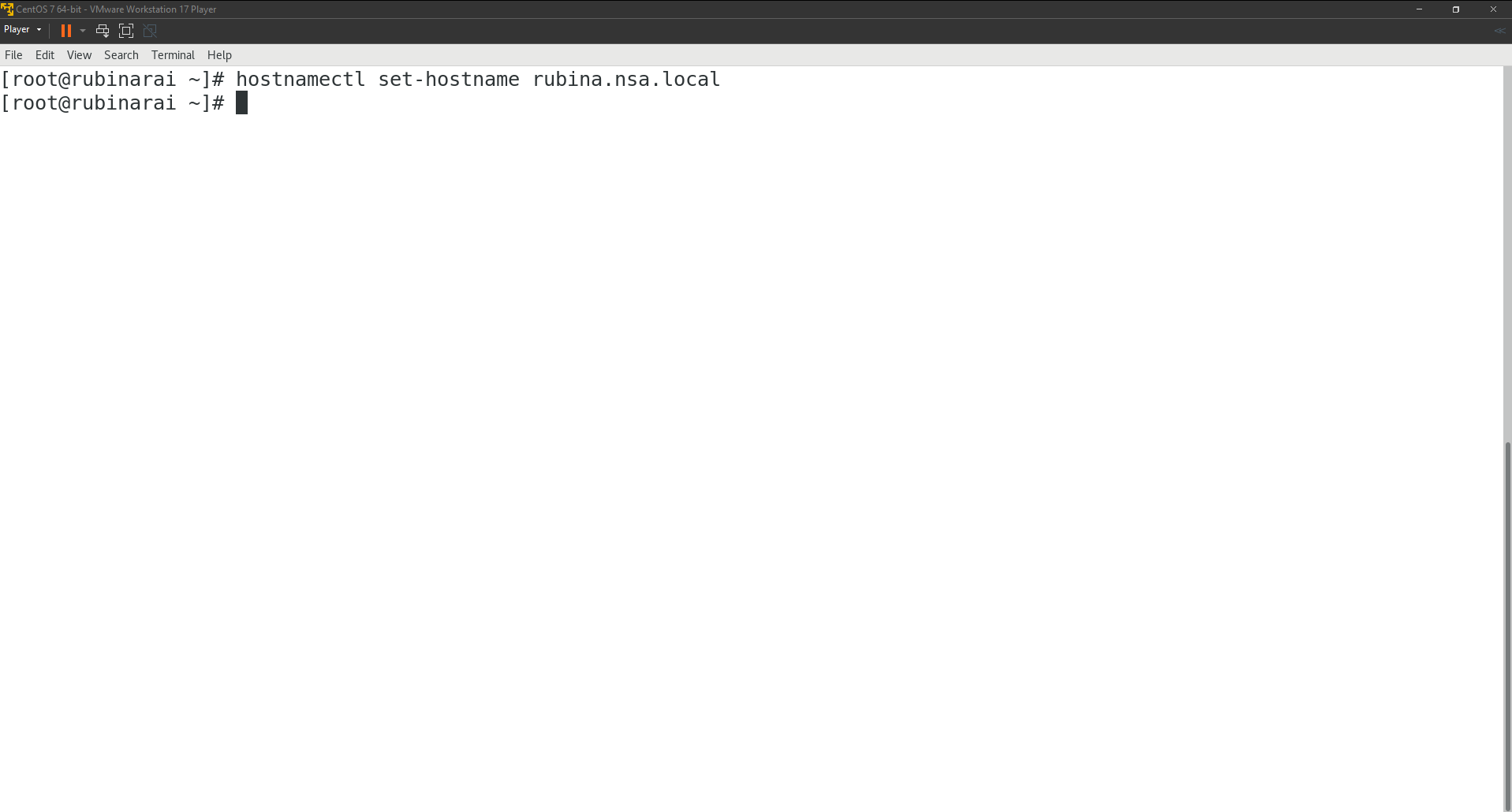


* 1. After the installation completed, I could now use my virtual machine.



1. Assign the hostname of Linux machine as <yourname>.nsa.local
   1. I applied the new hostname using the following command:

hostnamectl set-hostname rubina.nsa.local



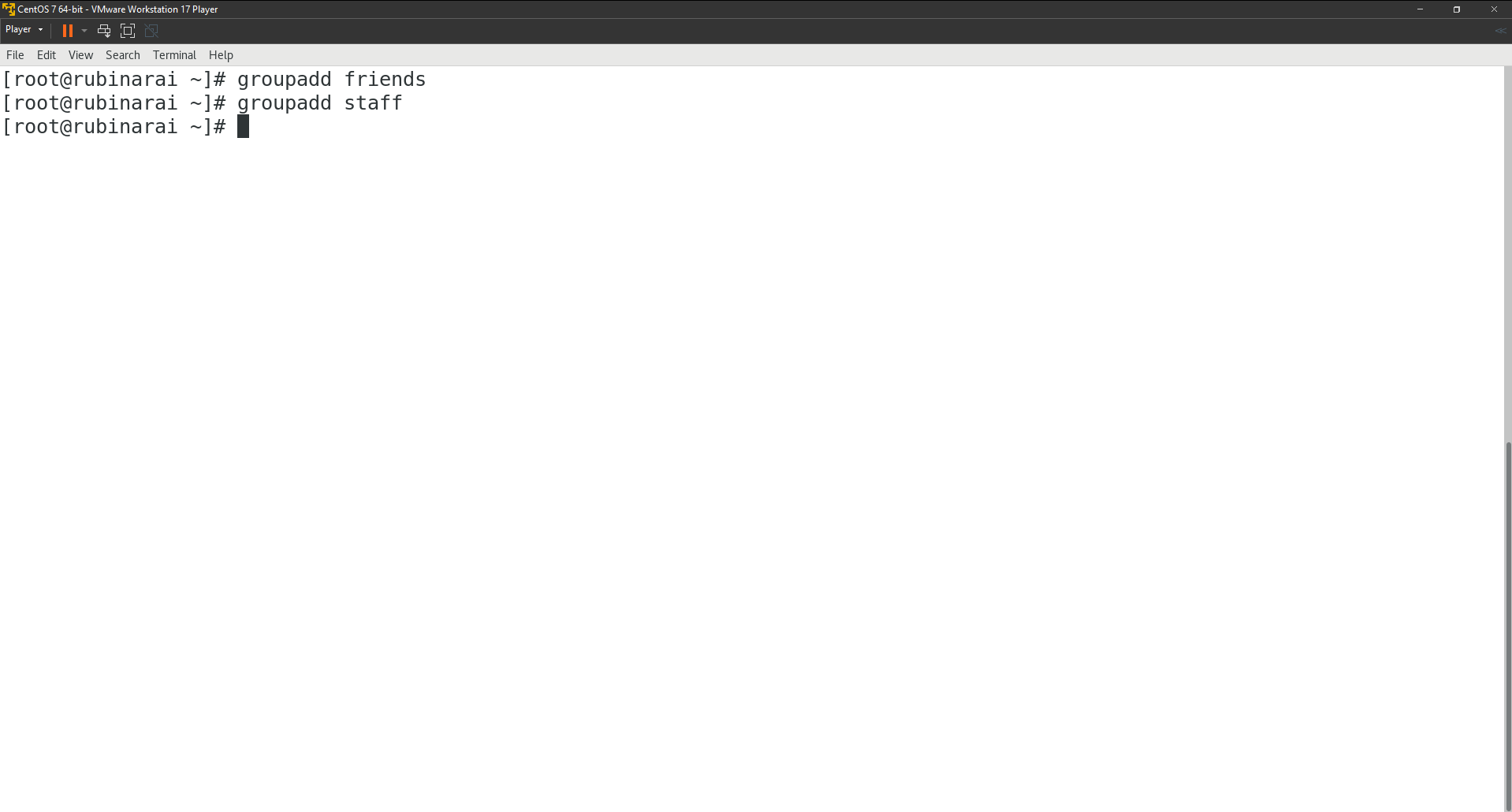
* 1. Then I checked whether the hostname was correctly applied using the following command.  
     hostnamectl

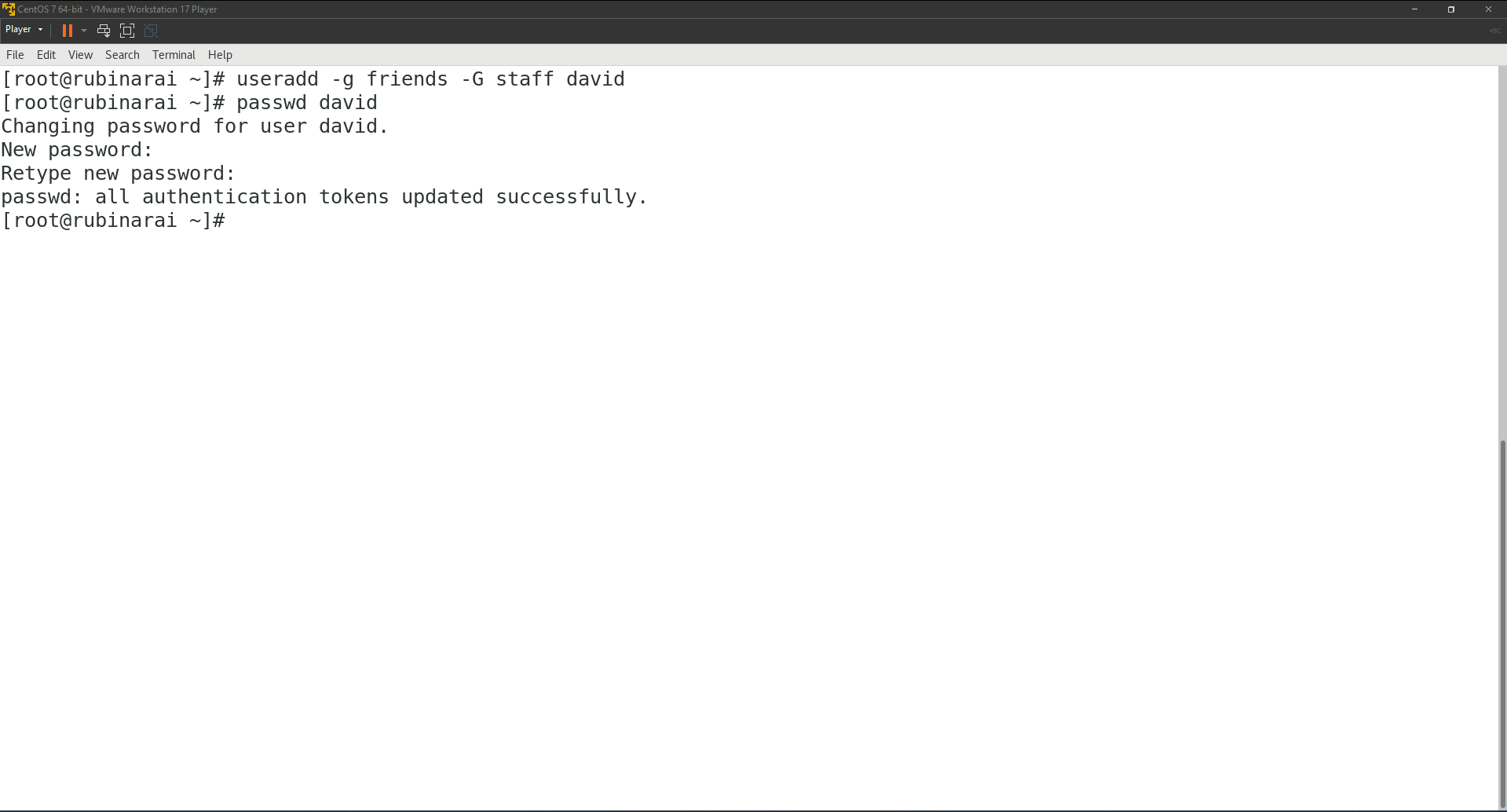


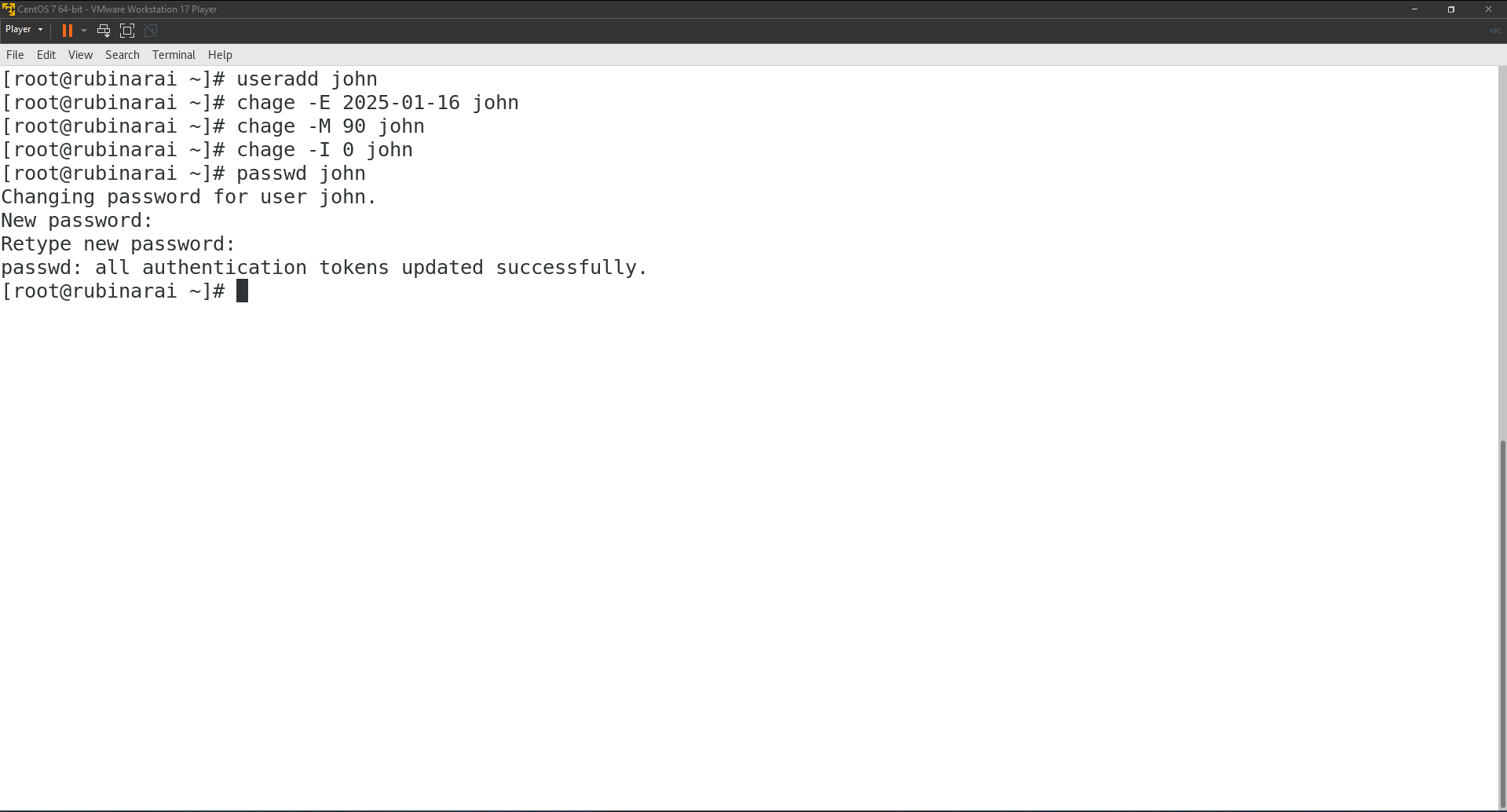
# **Task 6**

## **Users, Groups, Permission and other privileges:**

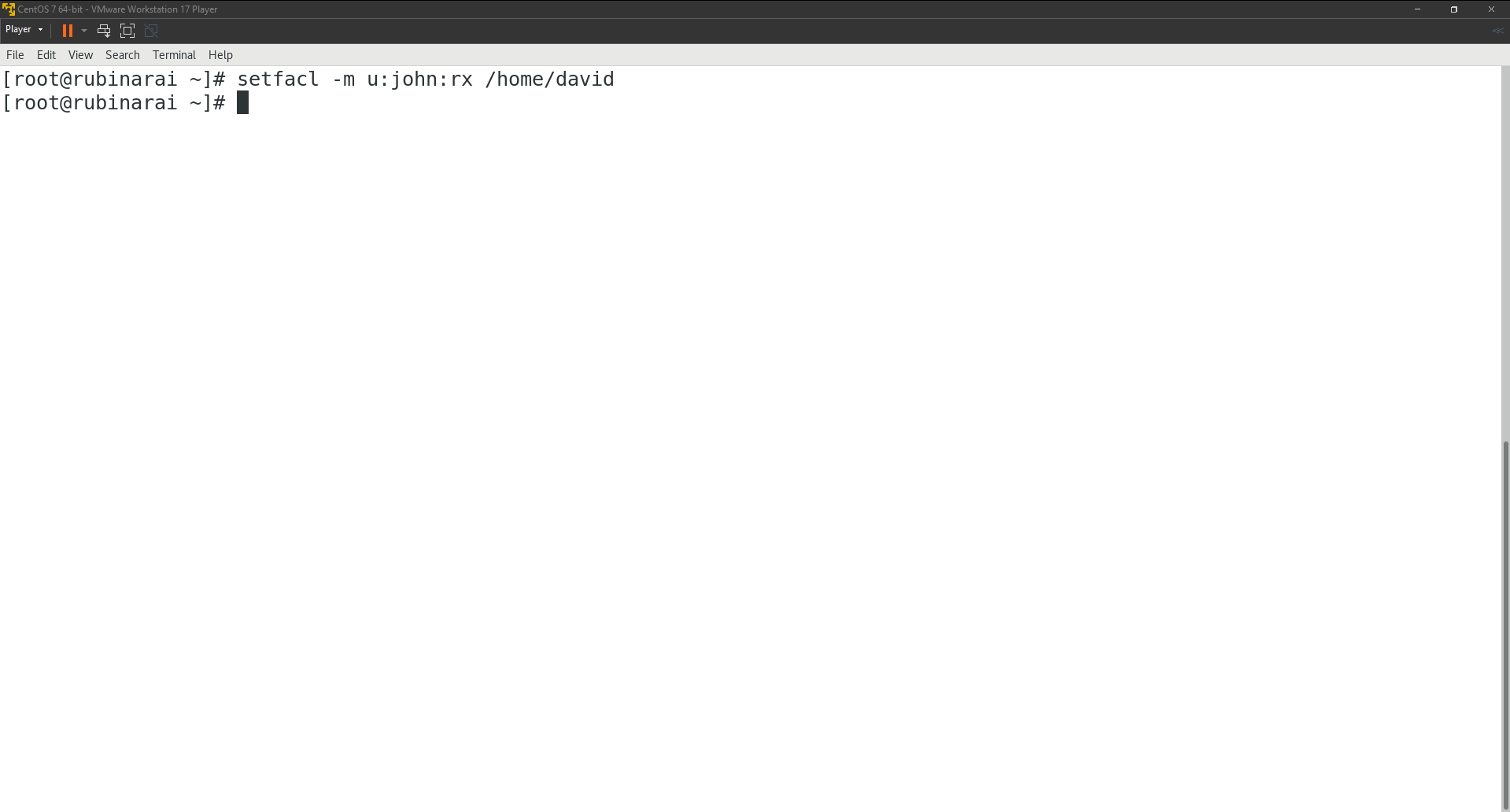
1. Create groups friends and staff.  
   I created two groups ‘friends’ and ‘staff’ using the following commands respectively.  
   groupadd friends  
   groupadd staff

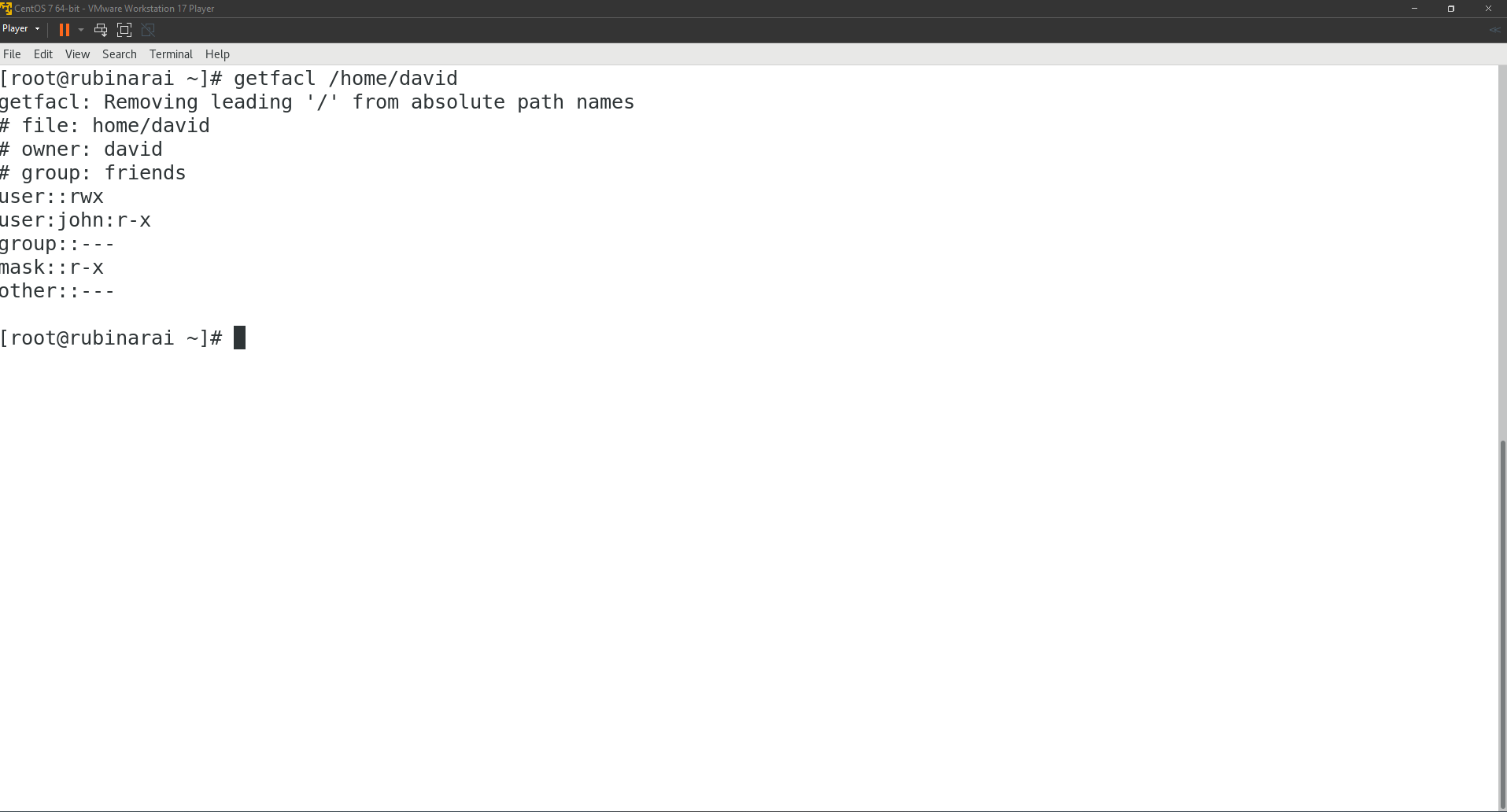


1. Create a user ‘david’ with friends as the primary group and staff as the secondary group  
   I created a user ‘david’ with friends as the primary group and staff as the secondary group using:  
   useradd -g friends -G staff david  
   Updated the password of the user ‘david’ using:  
   passwd david  
   
2. Create another user ‘john’ such that his account will expire on 16th Jan 2025, will have to renew password within every 90 days and will not get login shell.  
     
   Added user ‘john’:  
   useradd john  
   Set the expiration date of the user ‘john’ to **16th Jan 2025**.  
   chage -E 2025-01-16 john  
   Enforced password renewal every 90 days.  
   chage -M 90 john  
   Prevent login if the password is not renewed.  
   chage -I 0 john  
   Updated the password of the user ‘john’ using:  
   passwd john

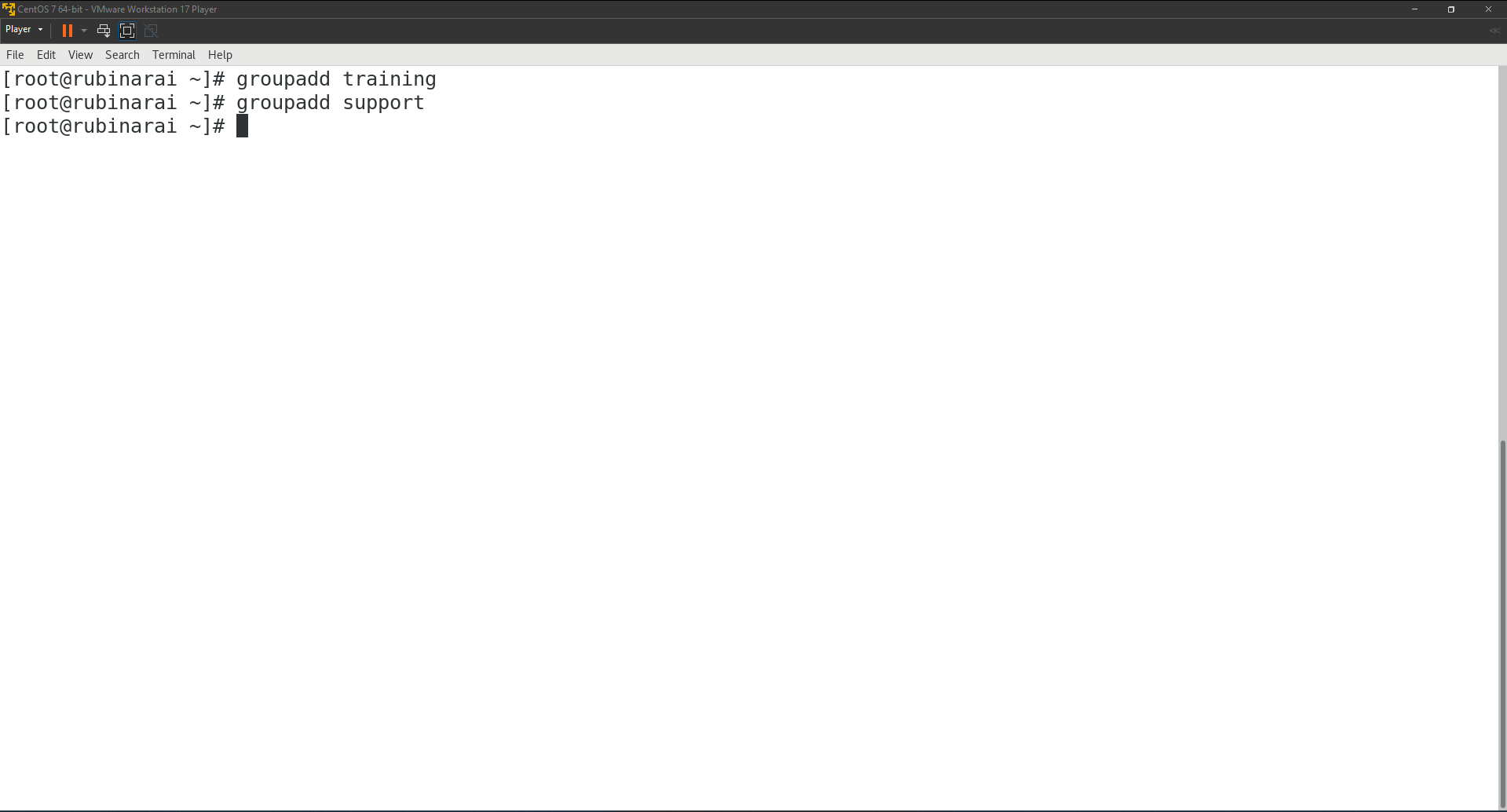


1. Assign read and execute permission on the home directory of david to john.  
   setfacl -m u:john:rx /home/david

 5. Verifying whether the permissions have been applied correctly using:

getfacl /home/david  


1. Create group for each department (training, support)  
   groupadd training   
   groupadd support

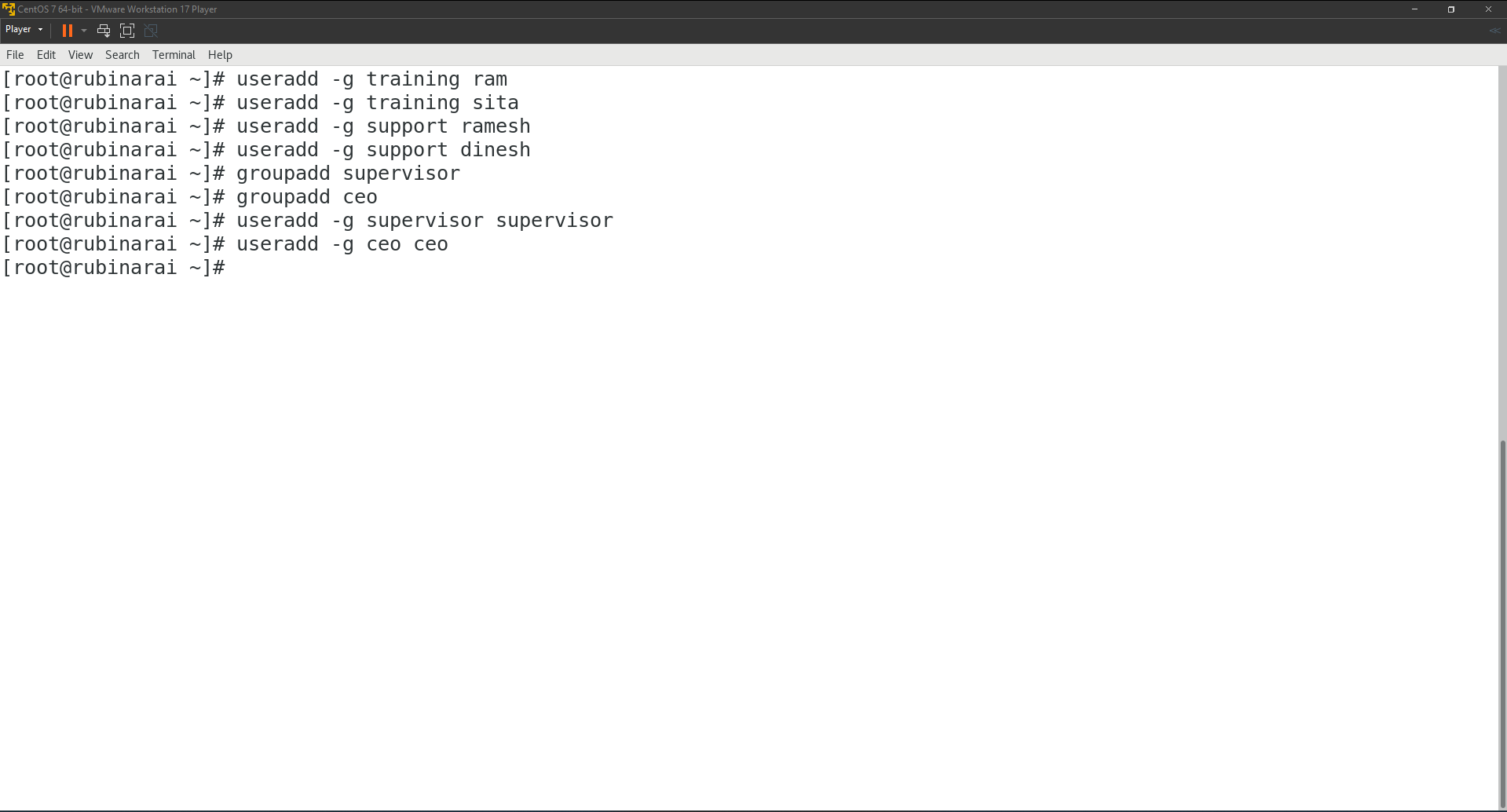


1. Create user account (ram, sita, ramesh, dinesh, supervisor, ceo) for each employee assigning them respective group.

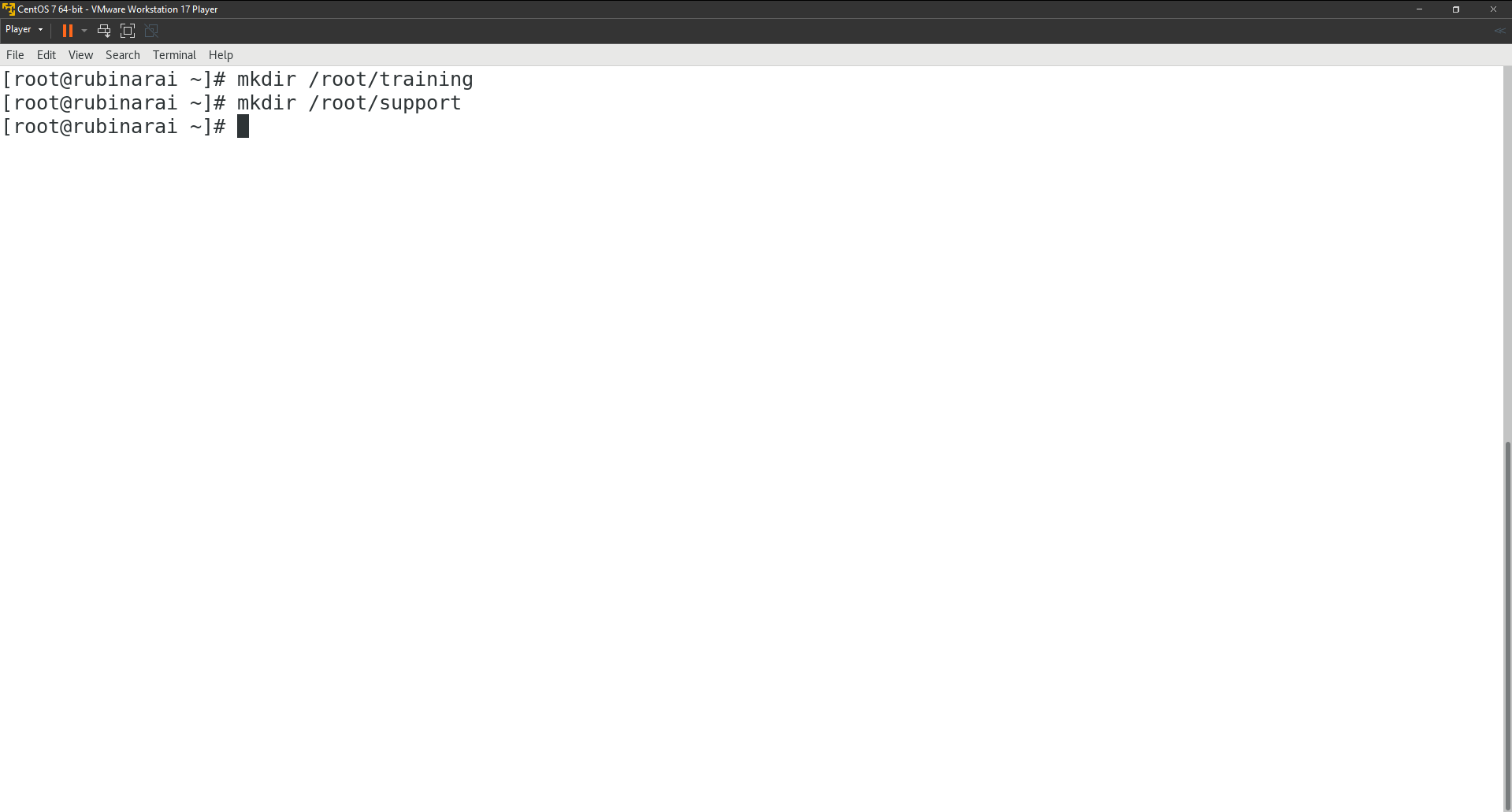
Creating ram and sita user and assigning them to training group.  
useradd -g training ram  
useradd -g training sita  
Creating ramesh and dinesh user and assigning them to training group.  
useradd -g support ramesh  
useradd -g support dinesh

8. Creating users and groups for supervisor and ceo and assigning each user to their respective group.

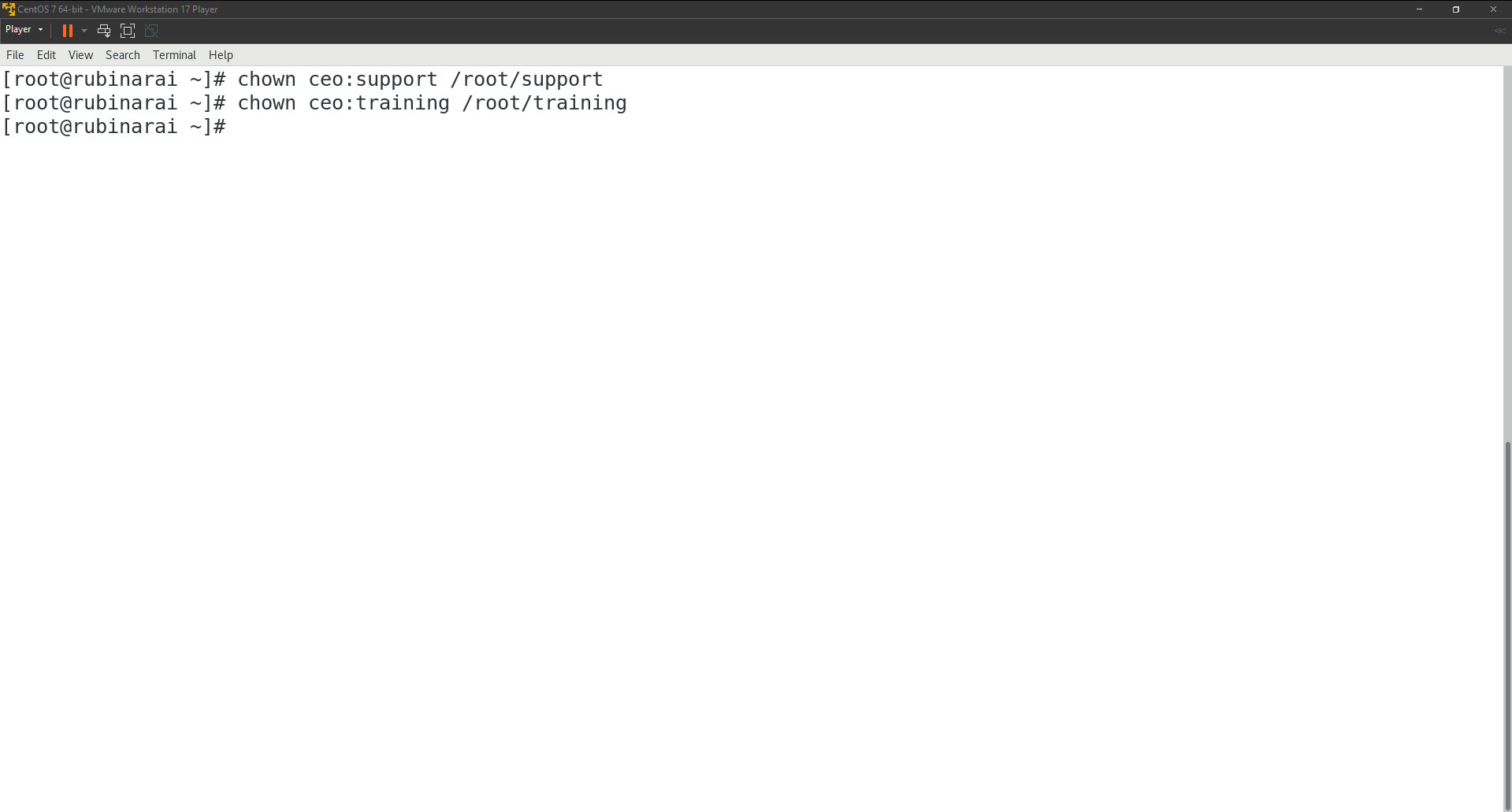
groupadd supervisor  
groupadd ceo  
useradd -g supervisor supervisor  
useradd -g ceo ceo



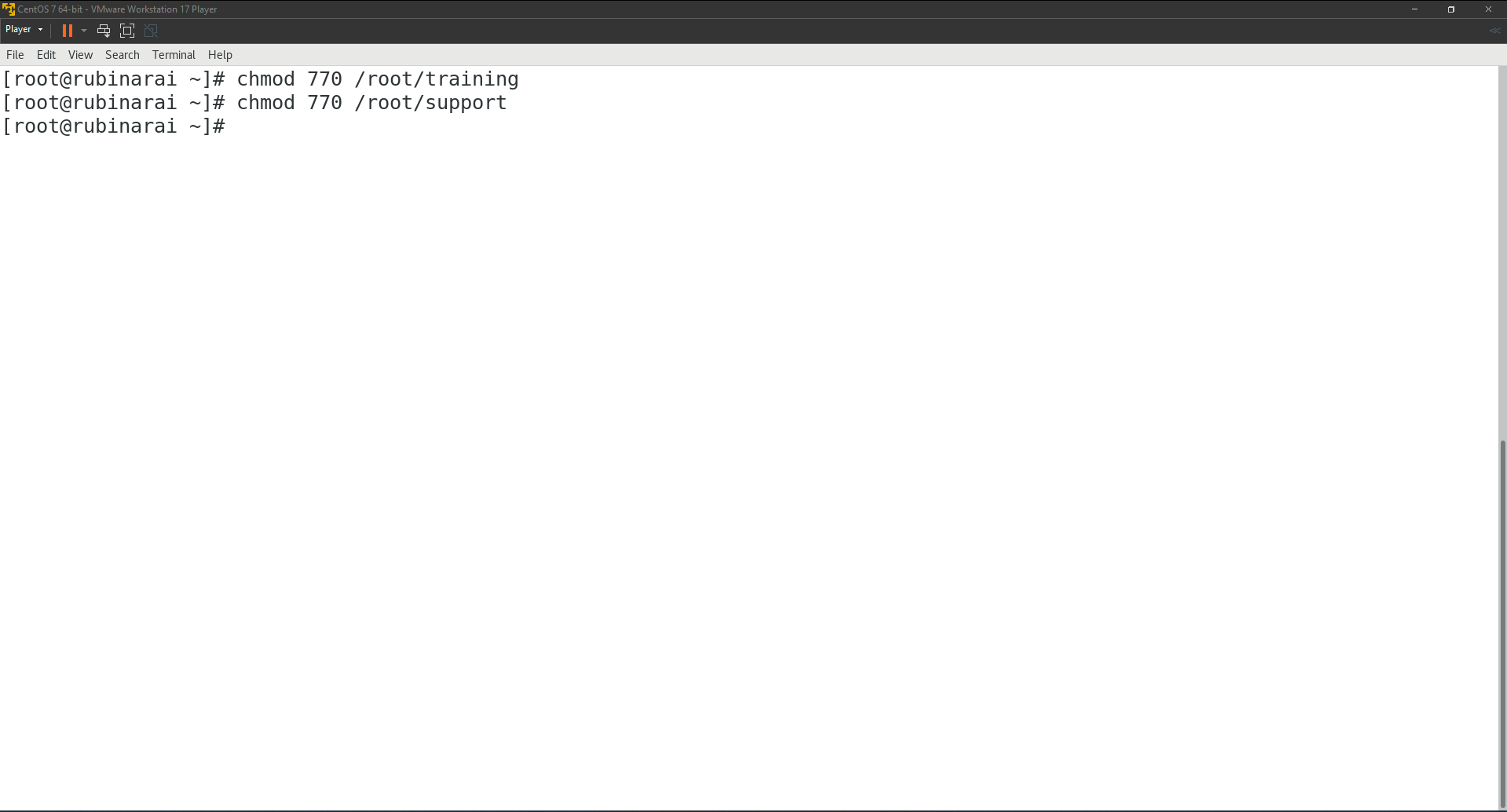
1. Create common directory (/root/training and /root/support) for each department.  
   mkdir /root/training  
   mkdir /root/support



1. Change ownership of group directories such that ceo will become the owner and the respective groups will be group owner.  
   chown ceo:training /root/training  
   chown ceo:support /root/support



1. Change the permission of the group directories such that only the owner and group member will get full permission and other will not get any permission.  
   chmod 770 /root/training  
   chmod 770 /root/support



12. Verifying if the permissions have been applied correctly:

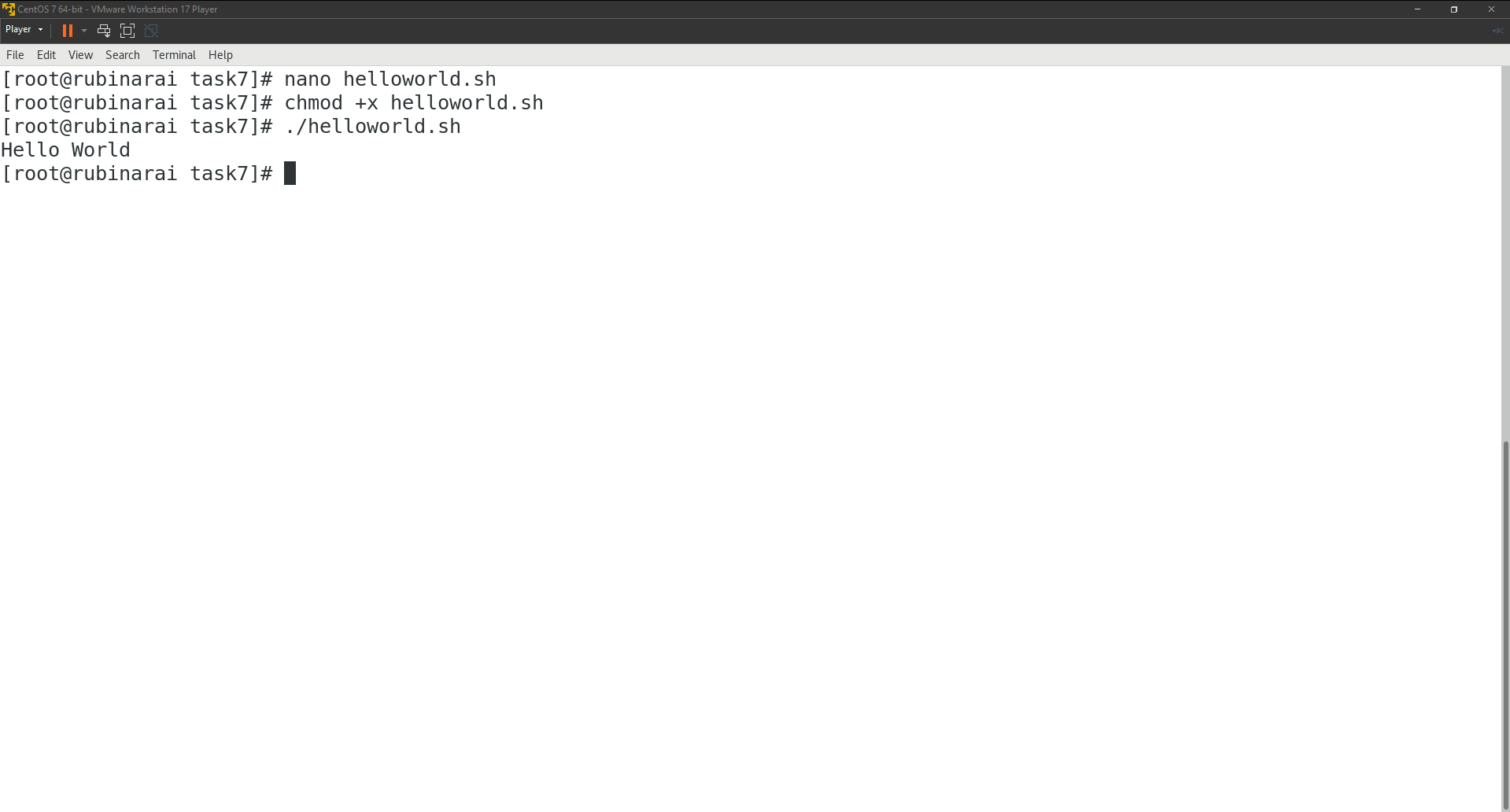
ls -ld /root/training /root/support

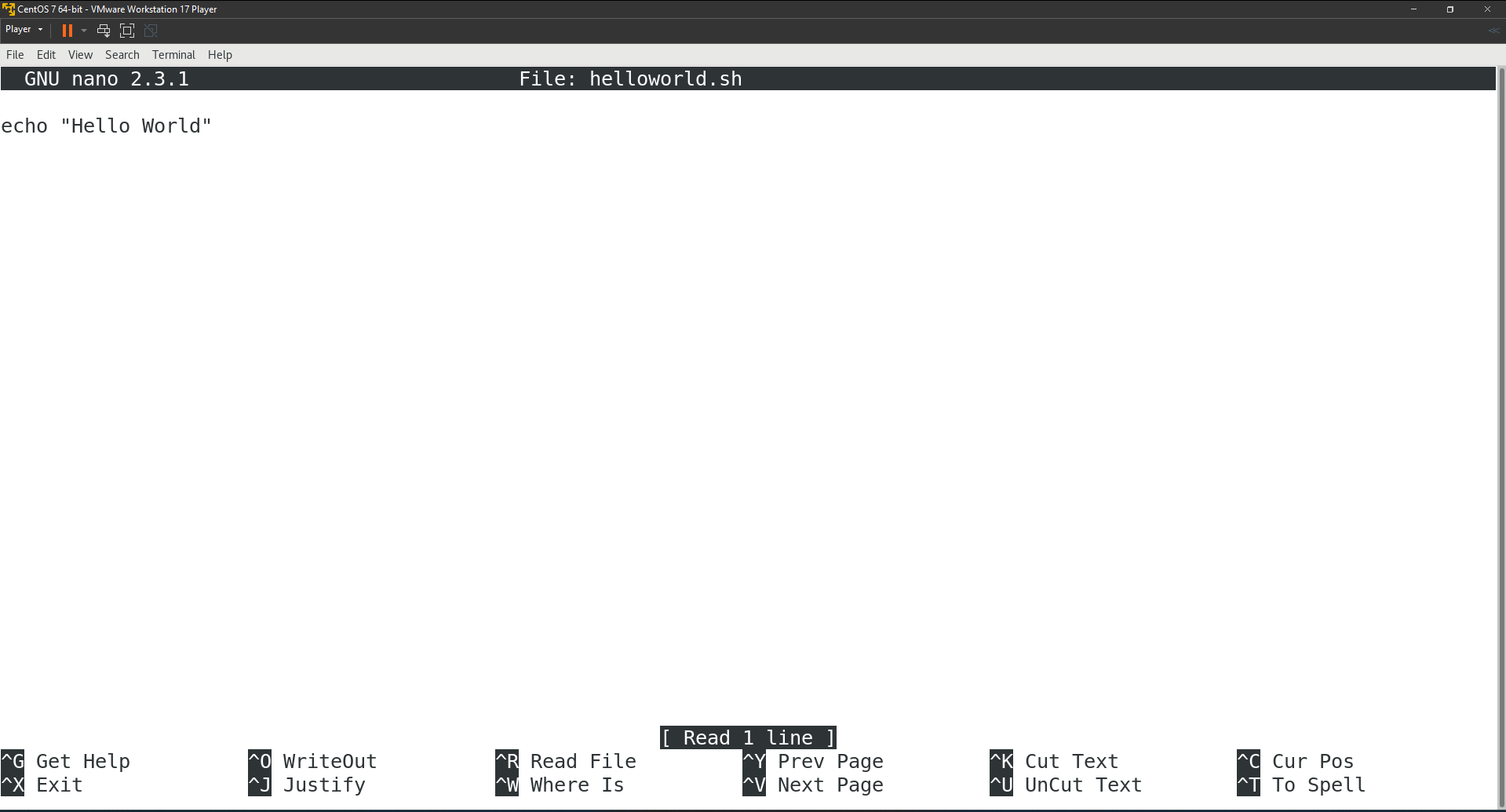


# **Task 7**

## **Bash Shell Scripting (Shell programming fundamentals)**

1. Write a script named helloworld.sh to display “Hello World”.  
   Creating helloworld.sh file:  
   nano helloworld.sh  
   Making the file executable using:  
   chmod +x helloworld.sh  
   Running the script  
   ./helloworld.sh



1. Content of helloworld.sh file:  
   echo "Hello World"  
   
2. Write a script named age.sh to prompt for age and display it.  
   Creating age.sh file:  
   nano age.sh  
     
   Making the file executable using:  
   chmod +x age.sh  
     
   Running the script  
   ./age.sh  
     
     
     
   Content of age.sh file:  
   read -p "Enter your age: " age

echo "Your age is $age"

1. Write a script to calculate simple interest.

Creating simple\_interest.sh file:  
nano simple\_interest.sh  
Making the file executable using:  
chmod +x simple\_interest.sh  
Running the script  
./ simple\_interest.sh

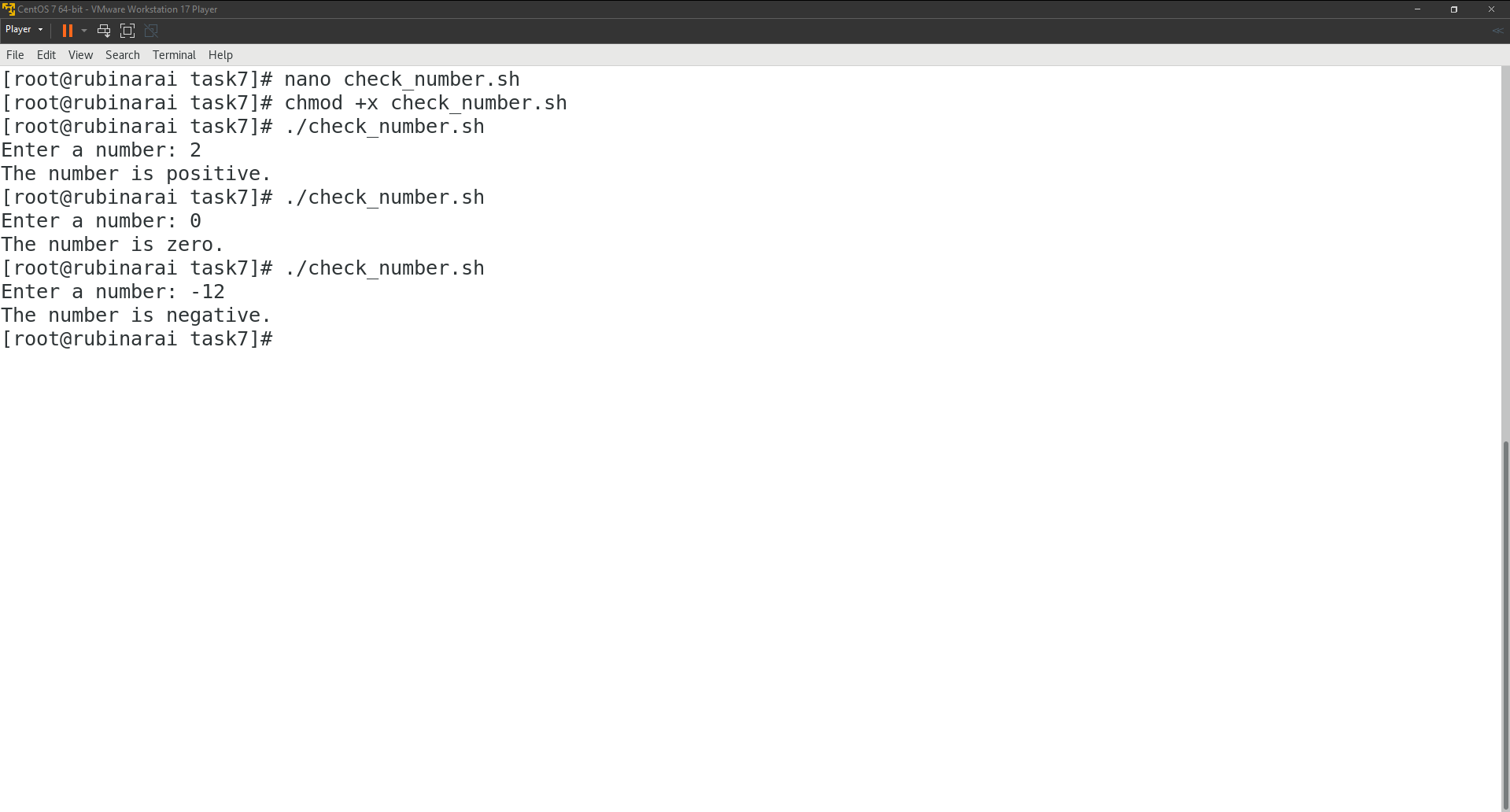
  
Content of age.sh file:  
read -p "Enter the principal amount (P): " principal

read -p "Enter the rate of interest (R): " rate

read -p "Enter the time in years (T): " time

interest=$(echo "scale=2; ($principal \* $rate \* $time) / 100" | bc)

echo "The simple interest is: $interest"  
  


1. Write a script to determine if a user-inputted number is positive, negative, or zero.  
   Creating check\_number.sh file:  
   nano check\_number.sh  
   Making the file executable using:  
   chmod +x check\_number.sh  
   Running the script  
   ./check\_number.sh  
     
   Content of check\_number.sh file:  
   read -p "Enter a number: " num

if (( $(echo "$num > 0" | bc -l) )); then

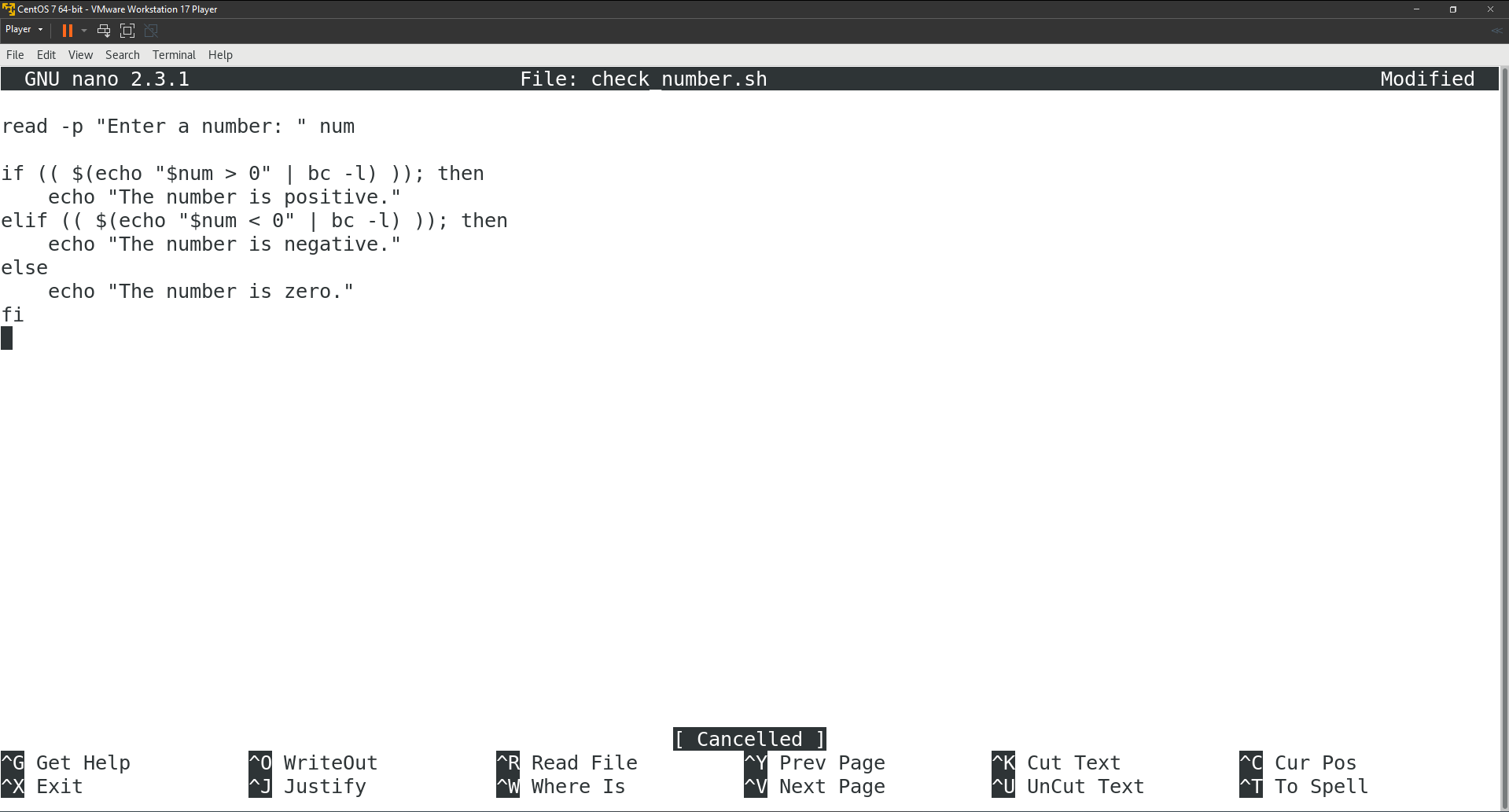
echo "The number is positive."

elif (( $(echo "$num < 0" | bc -l) )); then

echo "The number is negative."

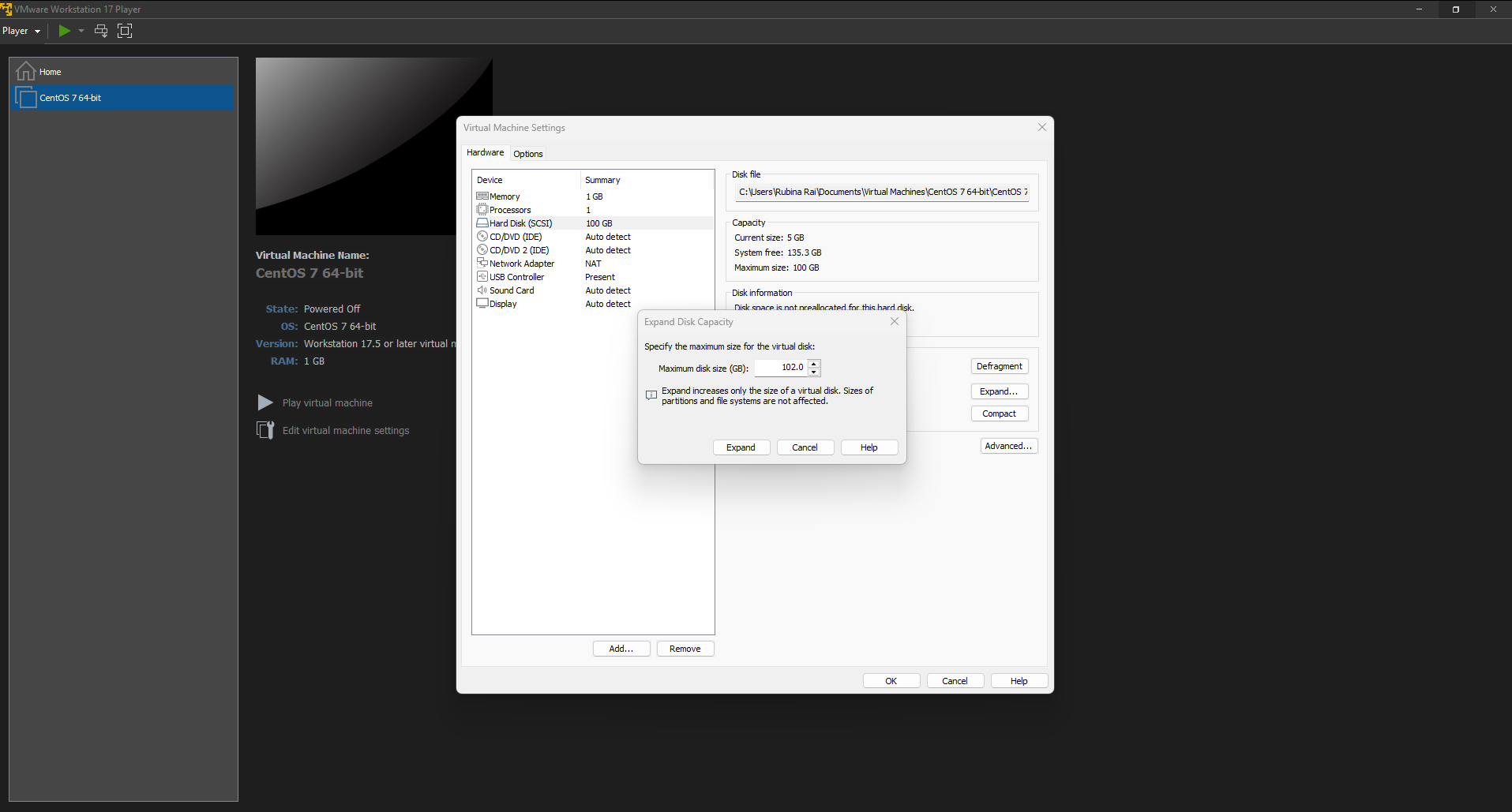
else

echo "The number is zero."

fi  


# **Task 8**

**Disk (File System and Quota) Management**

1. Create a partition of 500MB and format it with ext4 filesystem then mount it on /data1 directory permanently.
   1. I turned off my virtual machine and expanded its disk size by 2.0GB.  
      
   2. I booted my virtual machine and ran the following command to create a new partition:  
      fdisk /dev/sda  
      Inside fdisk:

n → new

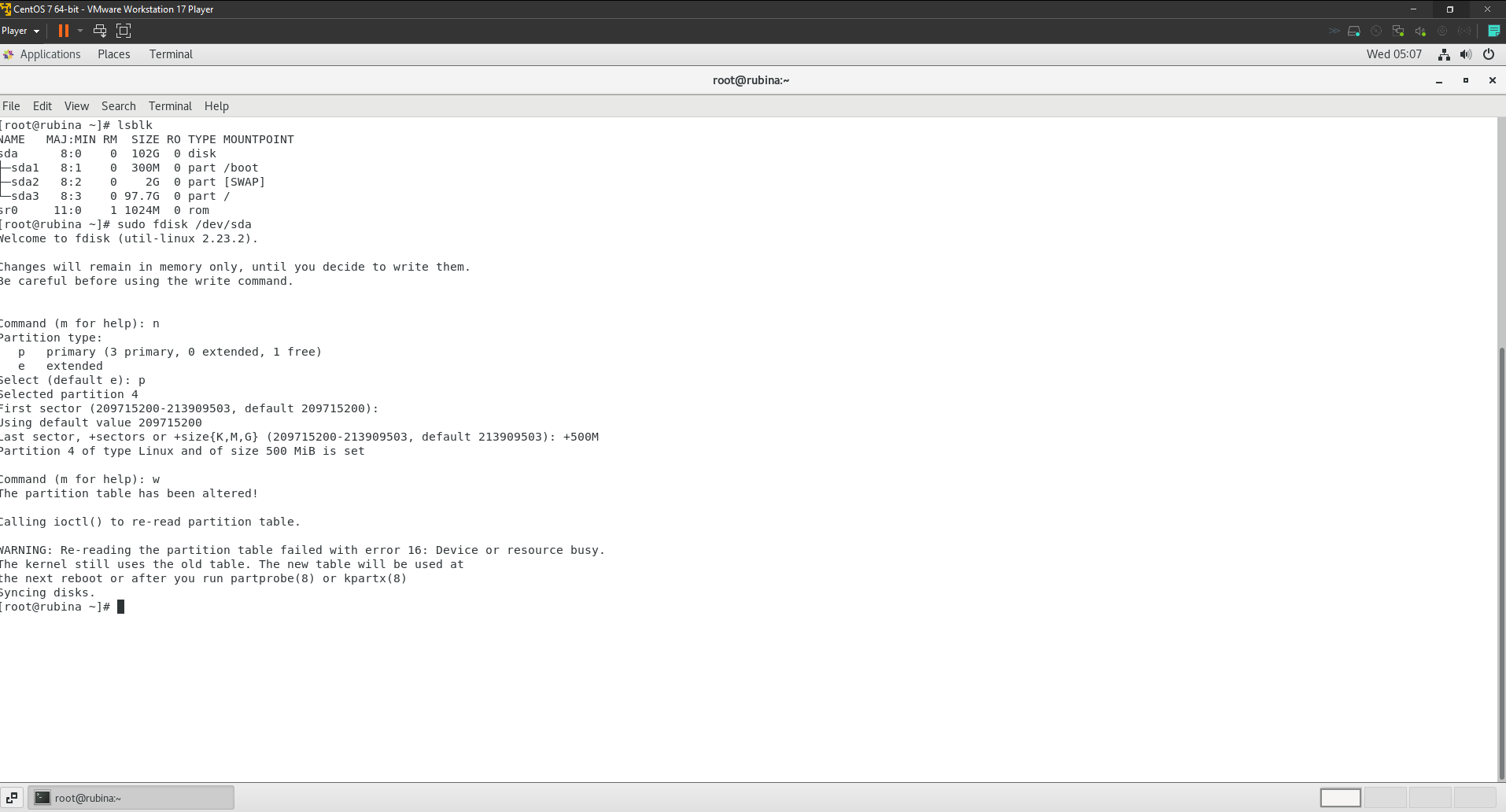
p → primary

Enter → default partition number

Enter → default first sector

+500M → size

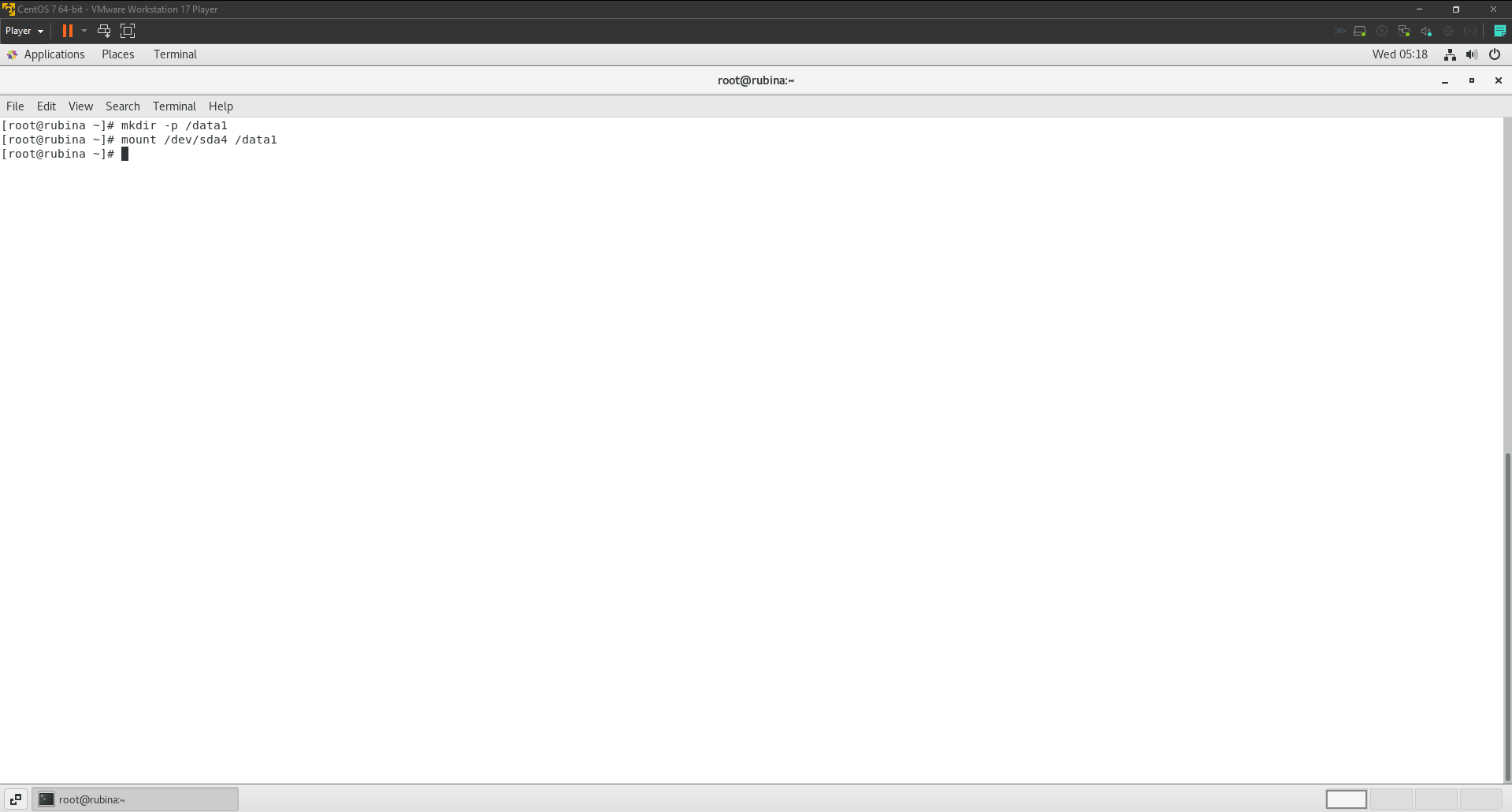
w → write and exit

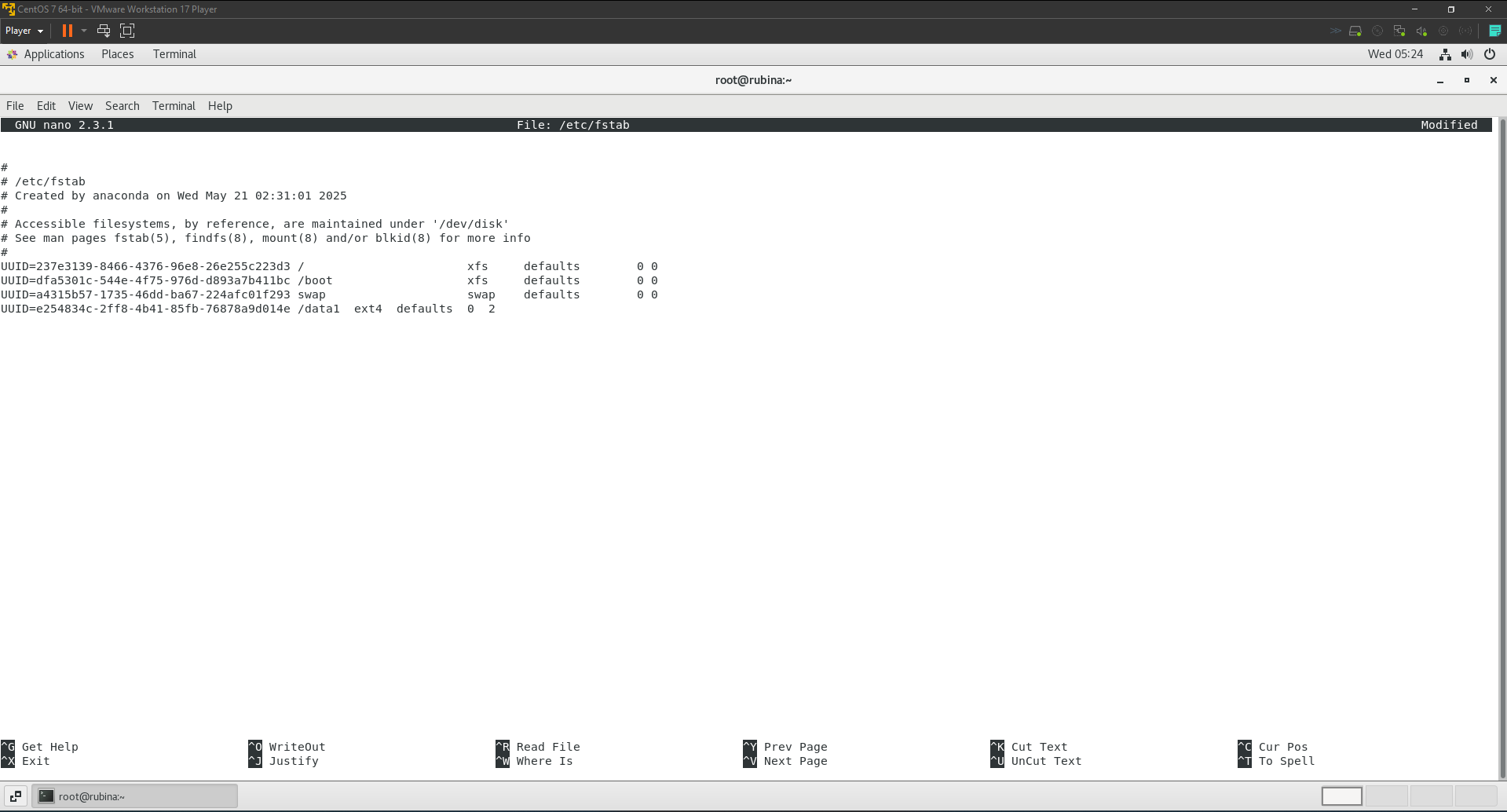
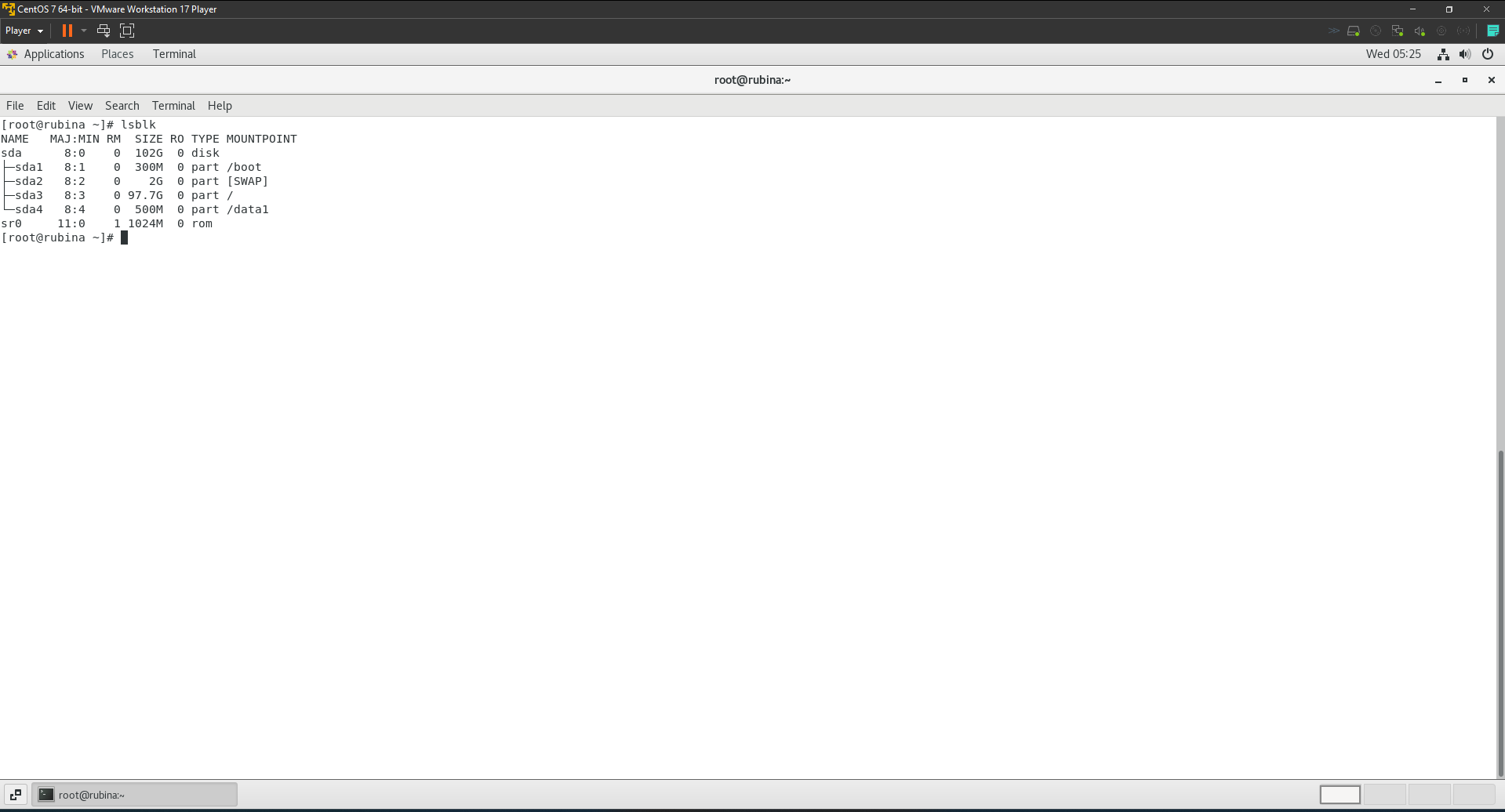


* 1. Since the disk was still being used, I had to reboot the virtual machine for the new partition to be displayed. Alternatively, you also can use partprobe command.  
     After rebooting the virtual machine, I used the following command to format the disk with **ext4** filesystem.  
     mkfs.ext4 /dev/sda4



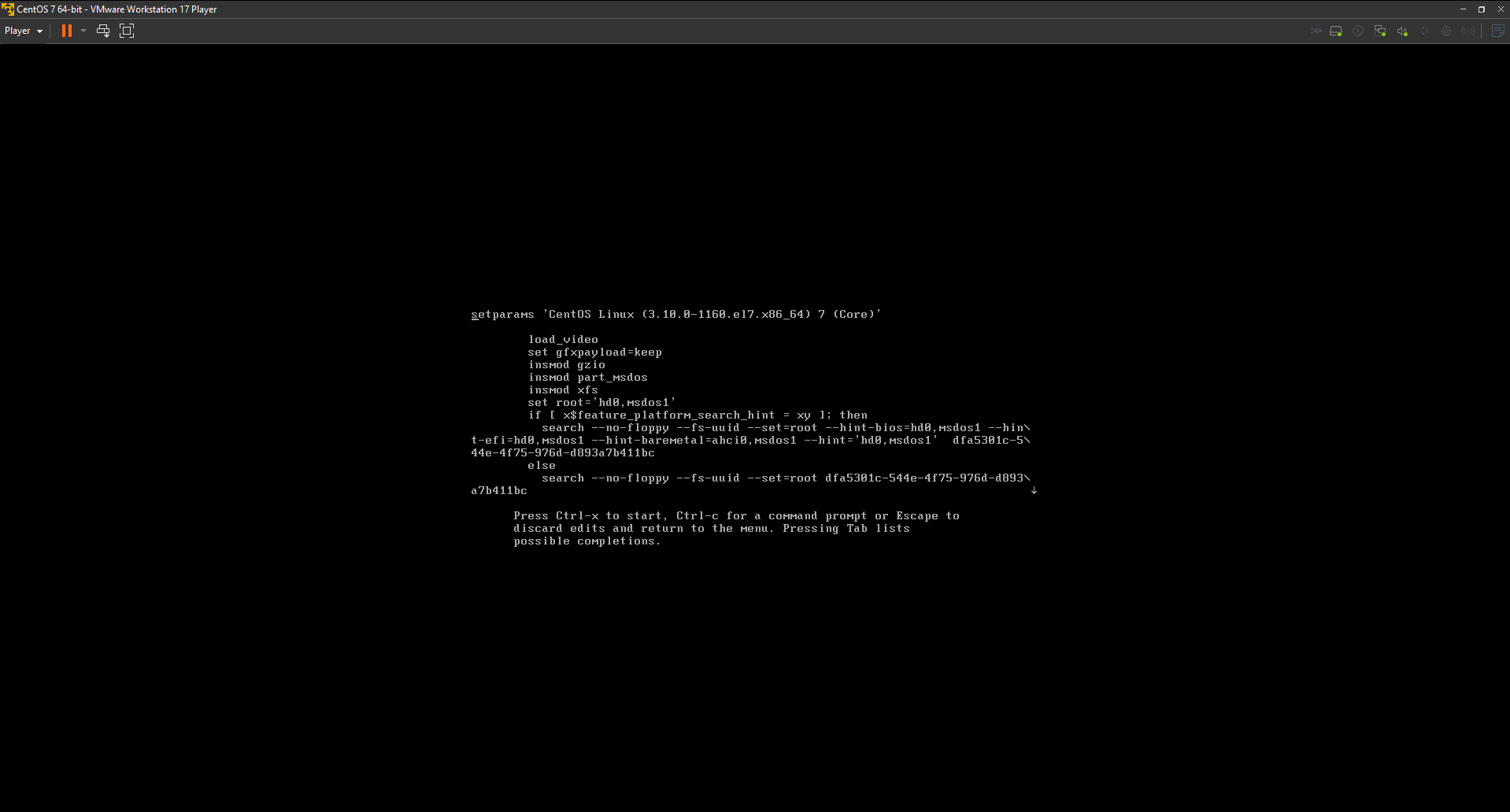
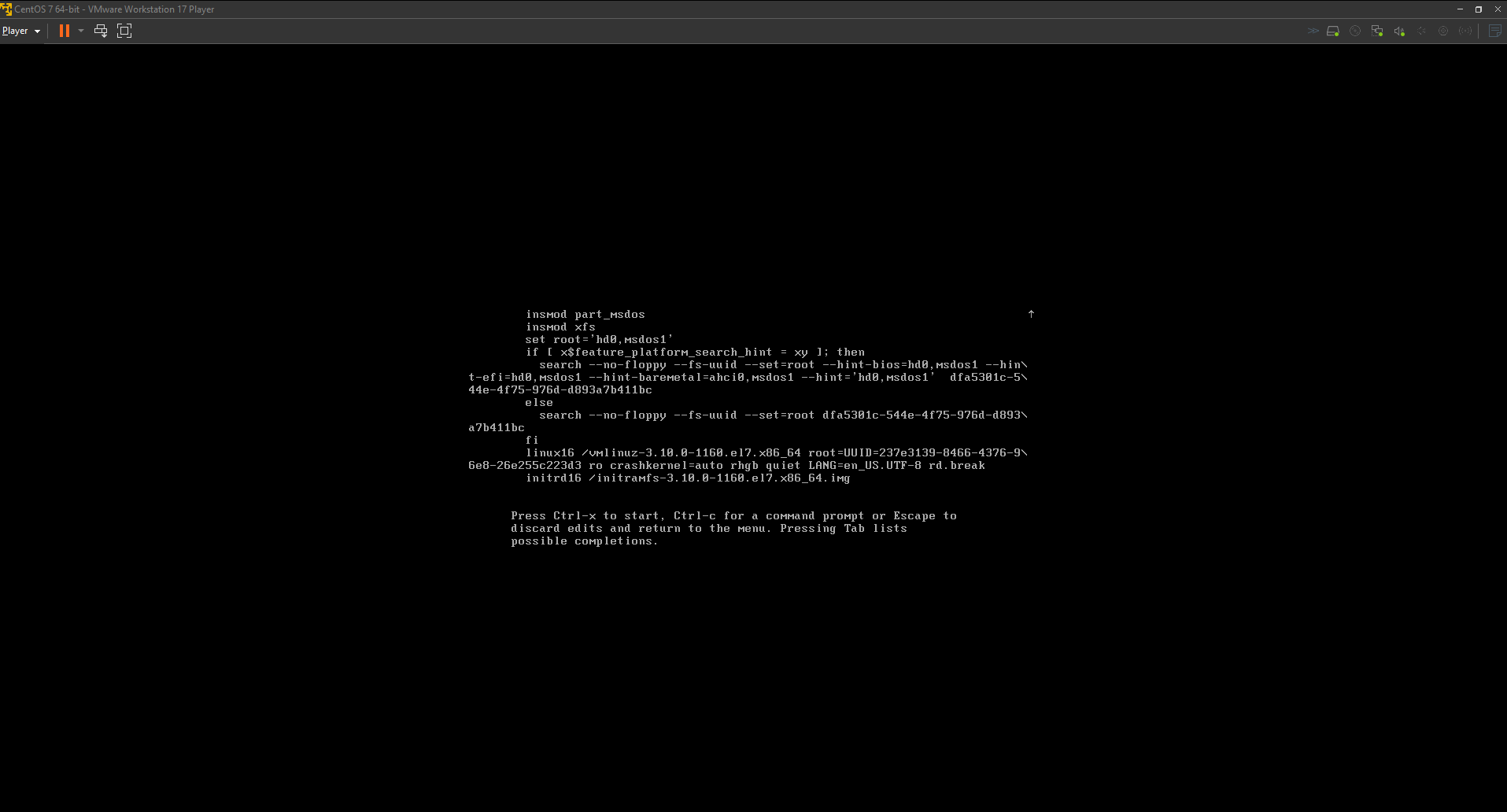
* 1. I created a mountpoint and mounted the partition on **/data1** using the following commands.  
     mkdir -p /data1  
     mount /dev/sda4 /data1

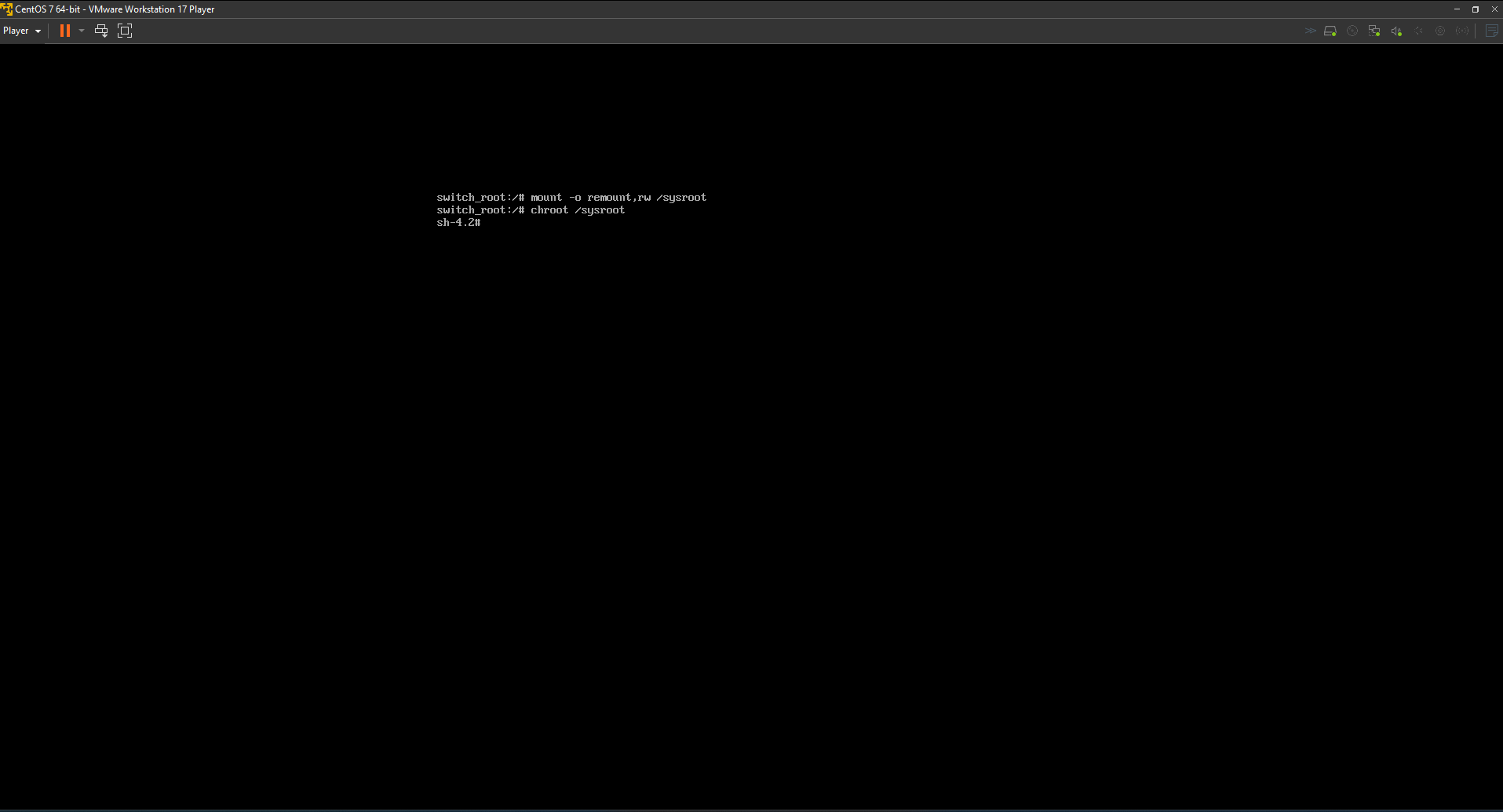


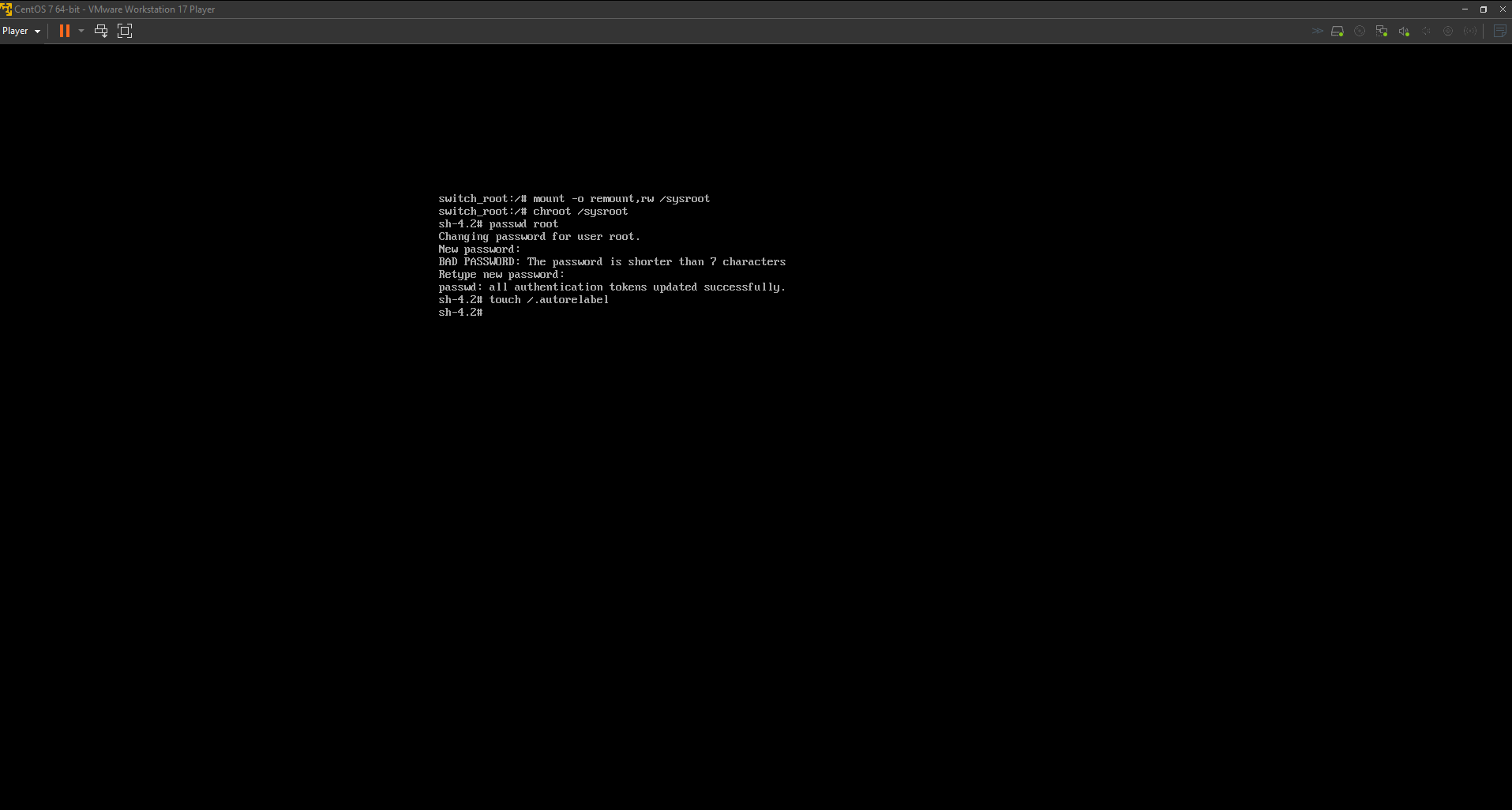
* 1. I followed the following steps to make the mount permanent.
     1. Find UUID:  
        blkid /dev/sda4
     2. Adding the UUID in /etc/fstab  
        nano /etc/fstab  
        
     3. Verifying if the partition has been created correctly.  
        

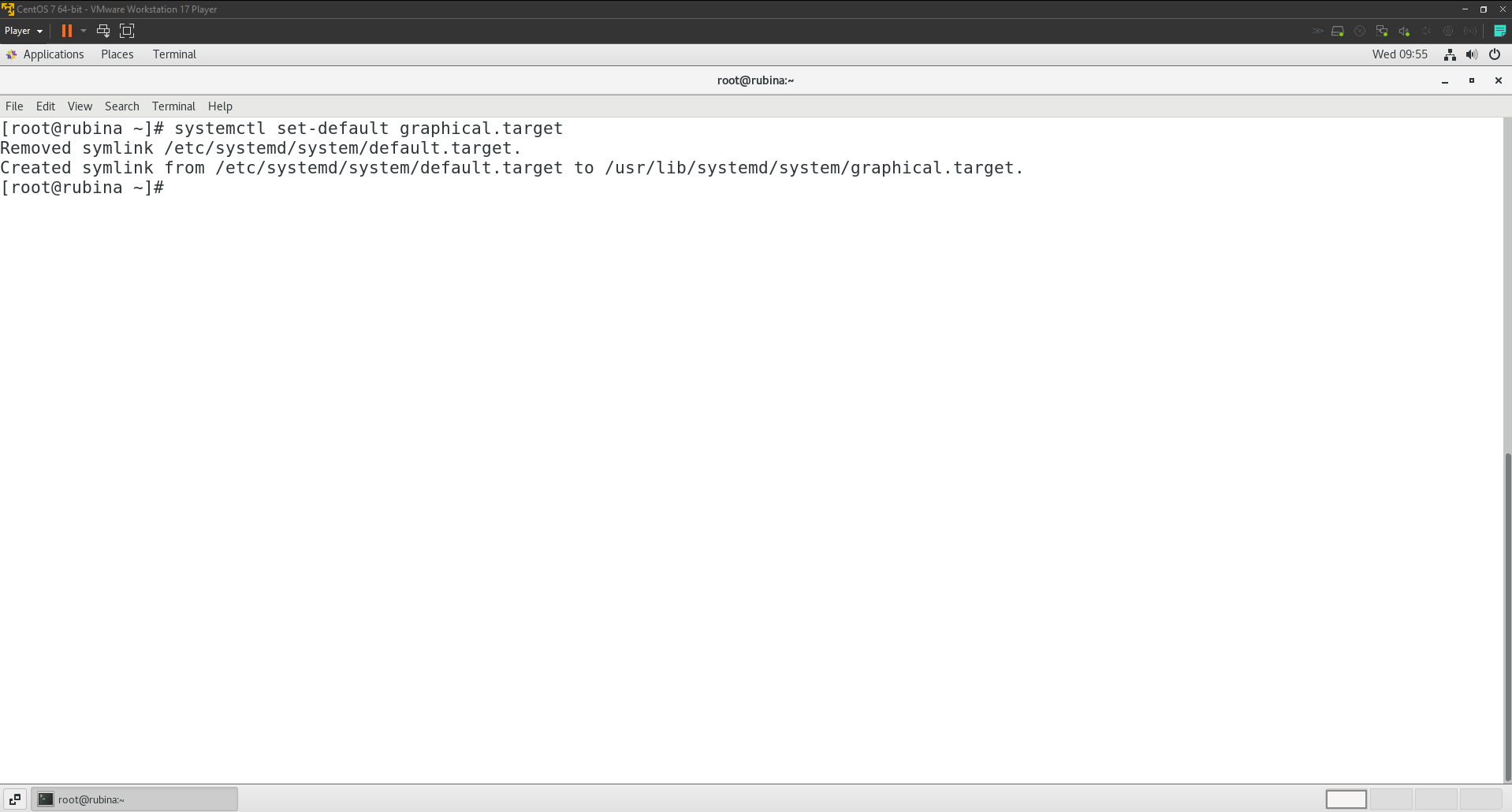
# **Task 9**

## **Boot Process and Service Management:**

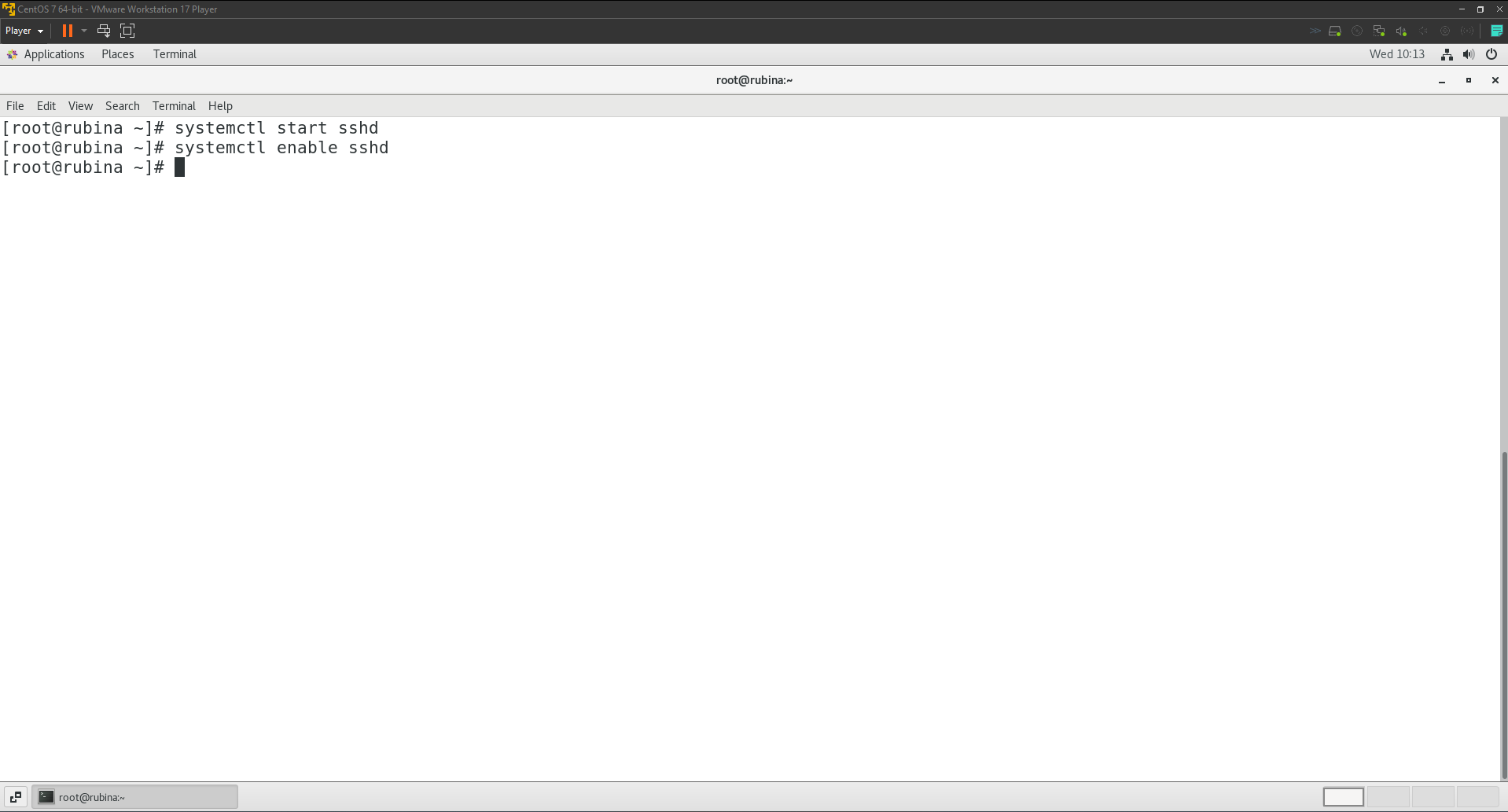
1. Recover the root password with “nsa13” as new password.
   1. I rebooted the virtual machine and at the **GRUB menu,** I pressed ‘e’ to edit the boot parameters  
      
   2. I searched for the line starting with linux16 and appended rd.break to that line. After which I press ctrl + x to boot with this option.  
      The purpose of the rd.break command is to interrupt the boot process and drop you into a **dracut shell.** From there, you can manually inspect or modify things before continuing the boot.  
      
   3. I mounted the real root filesystem in read-write mode so that I could make changes to it. I then switched the shell's environment to the installed system on the disk, allowing me to interact with it as if it were the live system, using the following commands:  
      mount -o remount,rw /sysroot

chroot /sysroot  


* 1. I finally updated the root password using the following command:  
     passwd root  
     I also ran the following command so that the system will automatically perform a full filesystem relabel on the next reboot. This step helps avoid login problems after reboot.  
     touch /.autorelabel  
     

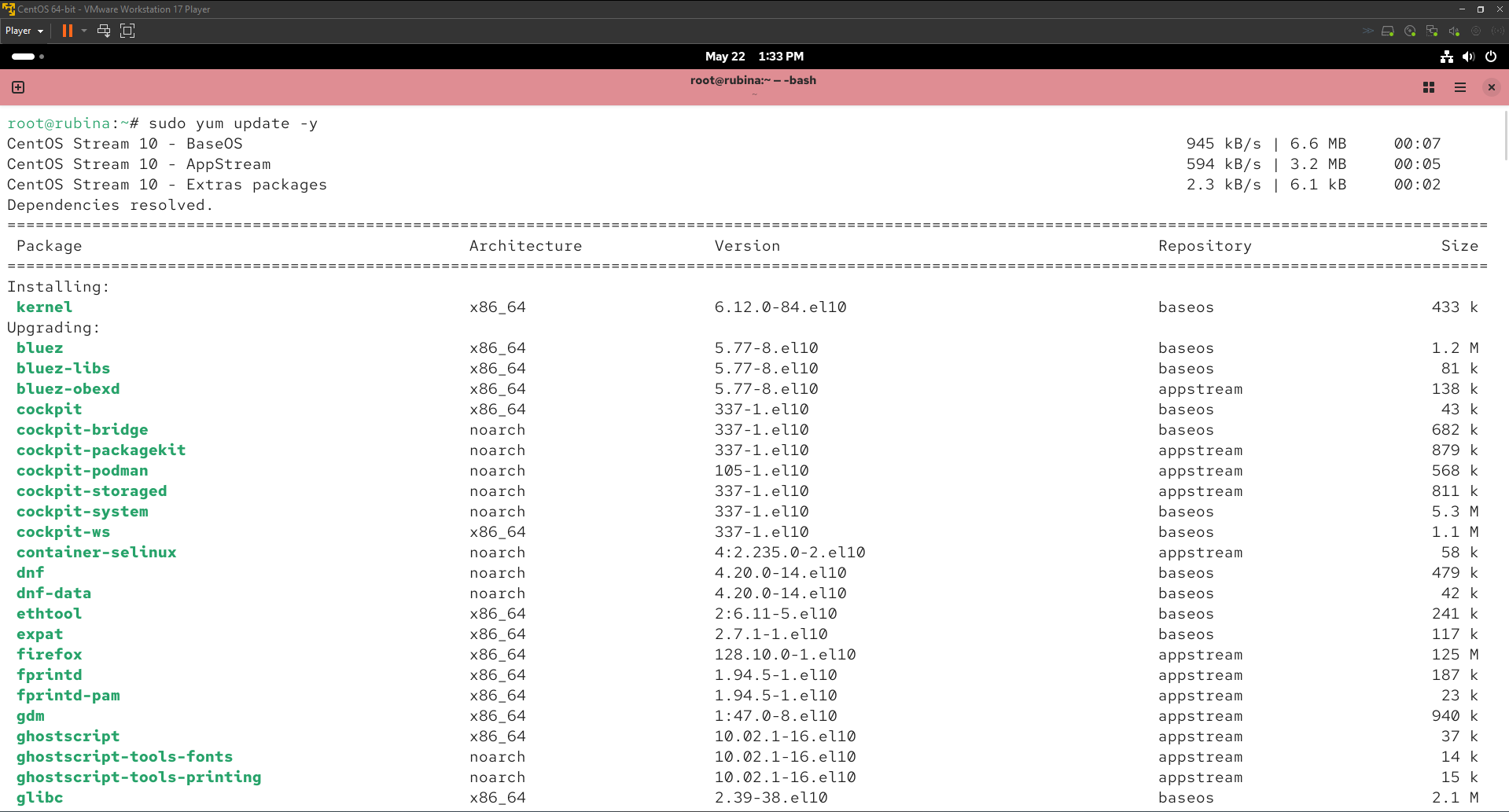
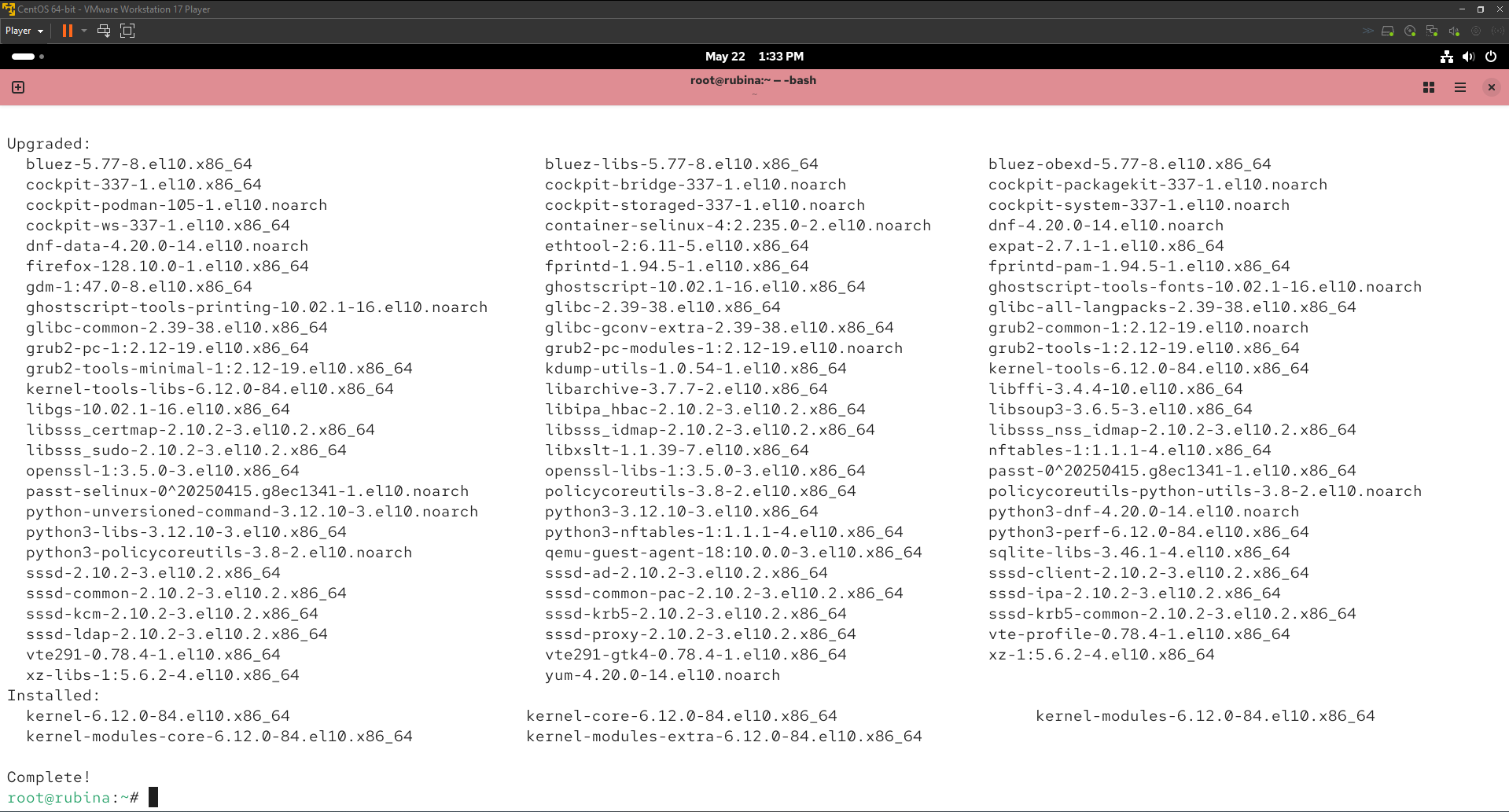
1. Suppose default boot mode of your system is in “multi-user.target”. Configure your system to boot in GUI mode by default i.e., “graphical.target”  
   To change the default boot mode to GUI (i.e., graphical.target), I used the following command:  
   systemctl set-default graphical.target  
   
2. Suppose ssh service is disabled in your system, check the status of the ssh service, then start the ssh service. Also, configure the ssh service to start automatically in boot time.

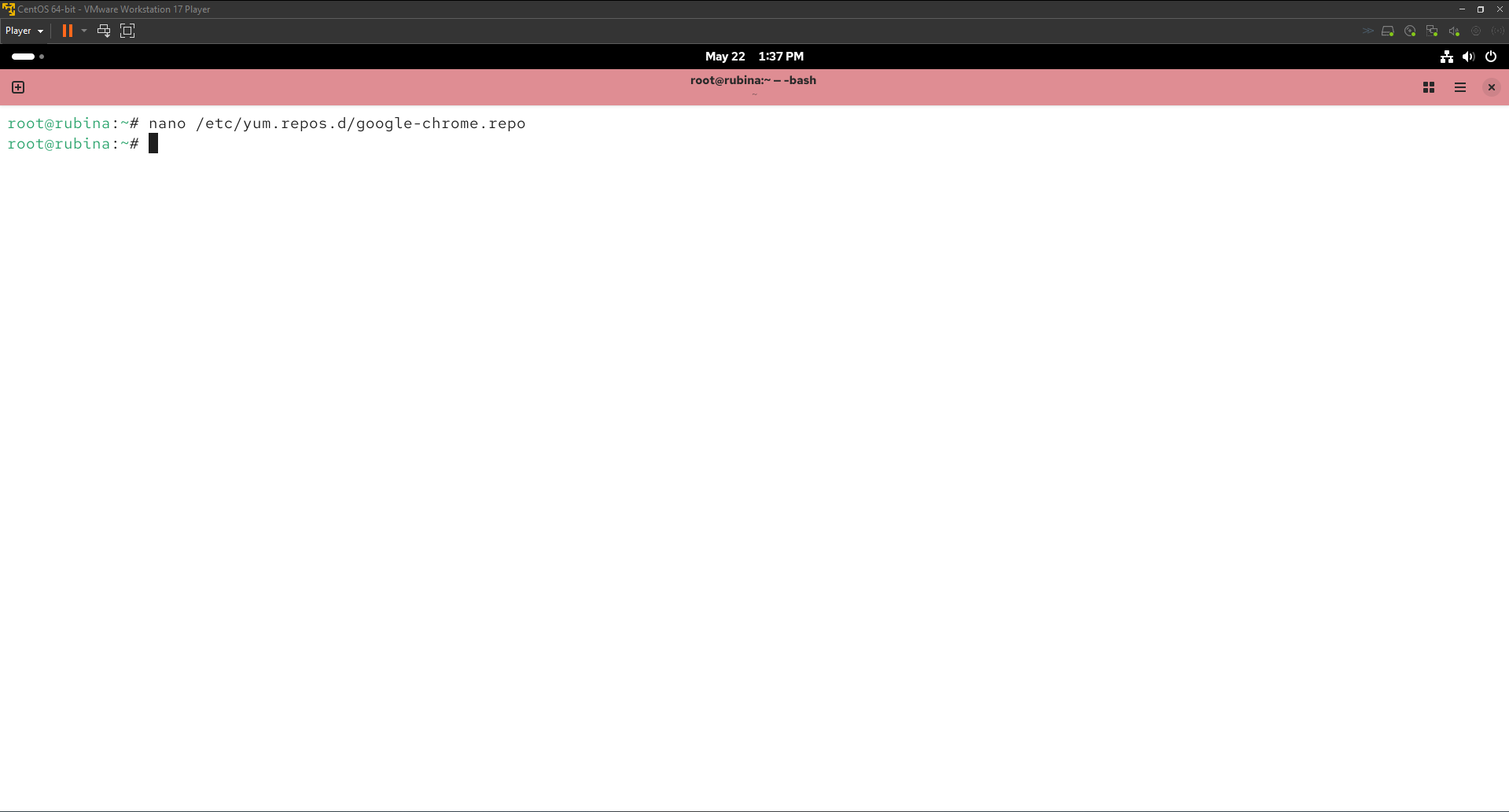
Note: SSH service is typically called sshd on CentOS, not ssh  
  
I used the following command to start the ssh service:  
systemctl status sshd

And the following command to enable SSH service to start on boot:  
systemctl enable sshd  


# **Task 10**

## **Red Hat Package Management (YUM Repository)**

1. Search a yum repo for google chrome, configure your system with respective remote repository then install google chrome in your system. Before installing google chrome update all the existing yum repositories.
   1. I updated existing yum repositories using the following command:  
      yum update -y  
        
      
   2. Then I used the following command to create a new repo file for google chrome:  
      sudo nano /etc/yum.repos.d/google-chrome.repo

  
This is the content of the repo file:  
[google-chrome]

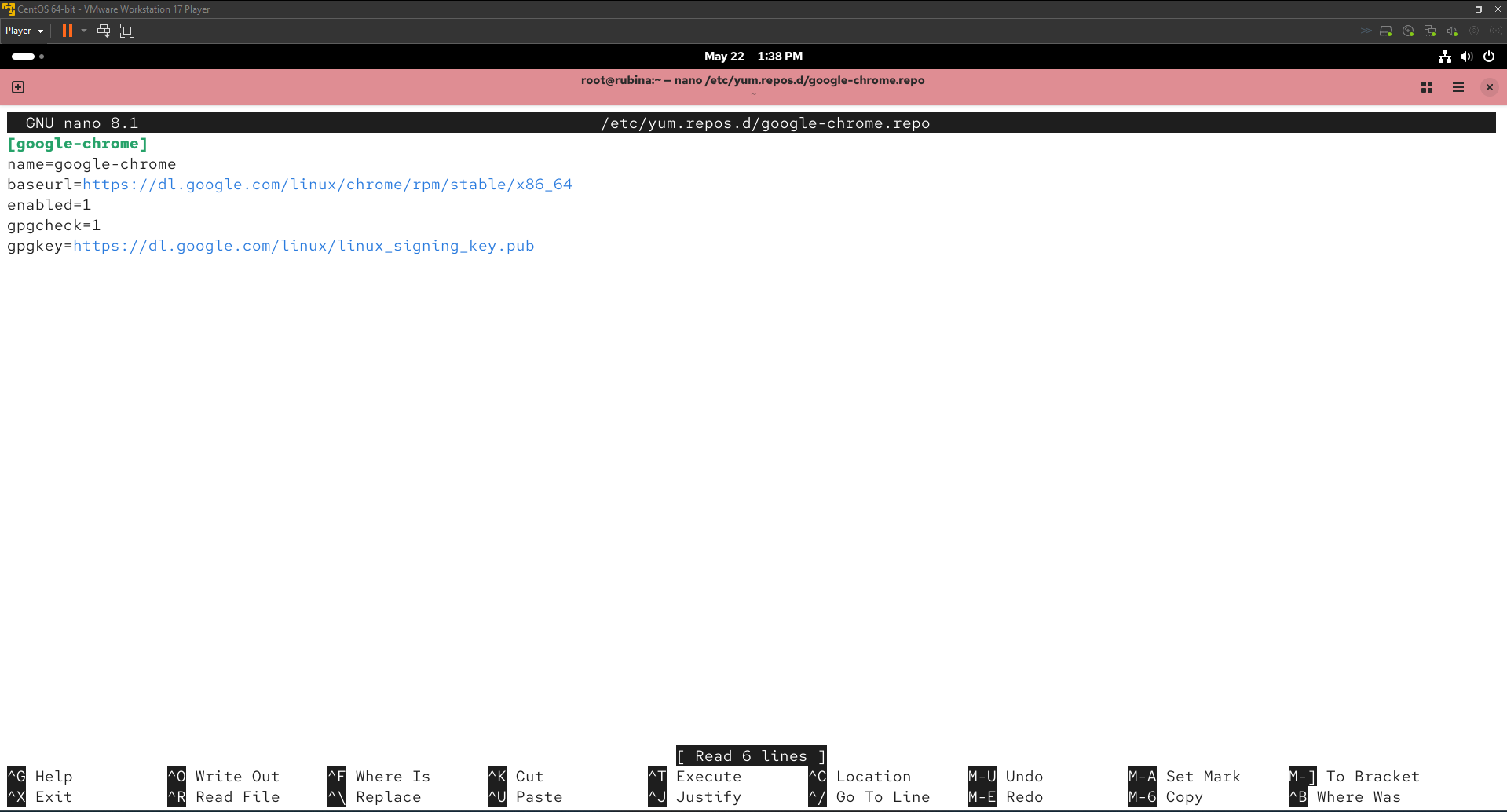
name=google-chrome

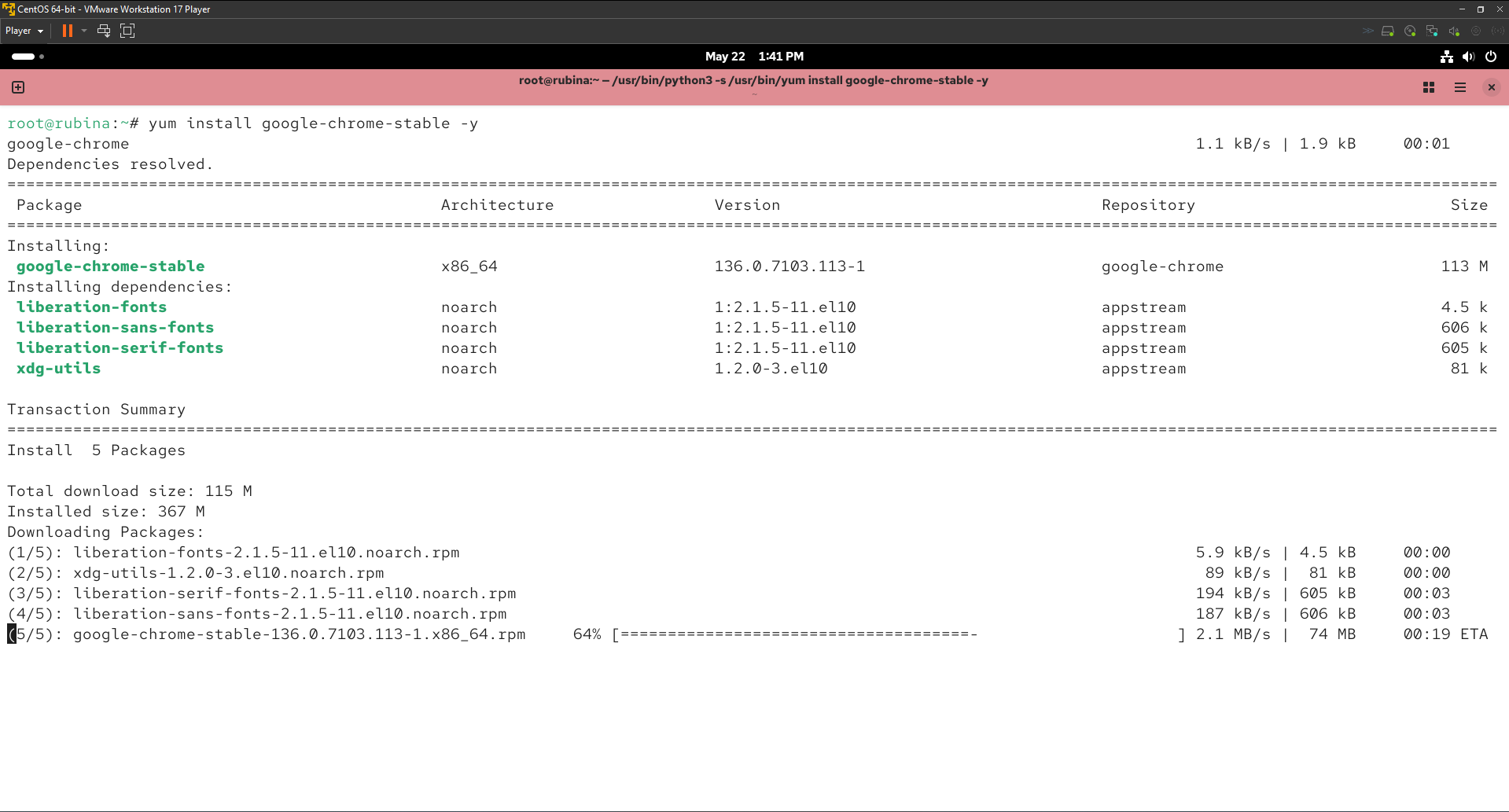
baseurl=https://dl.google.com/linux/chrome/rpm/stable/x86\_64

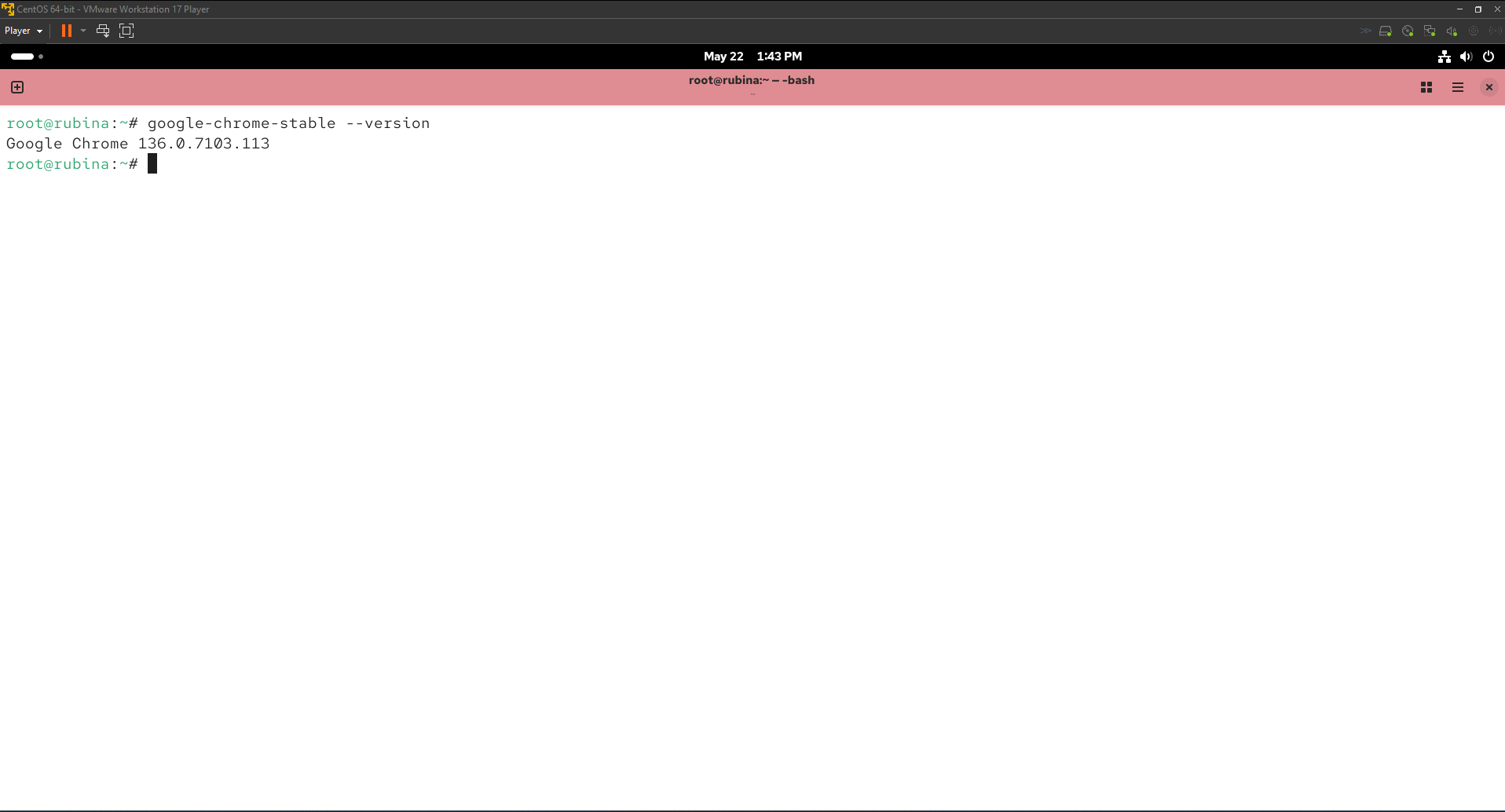
enabled=1

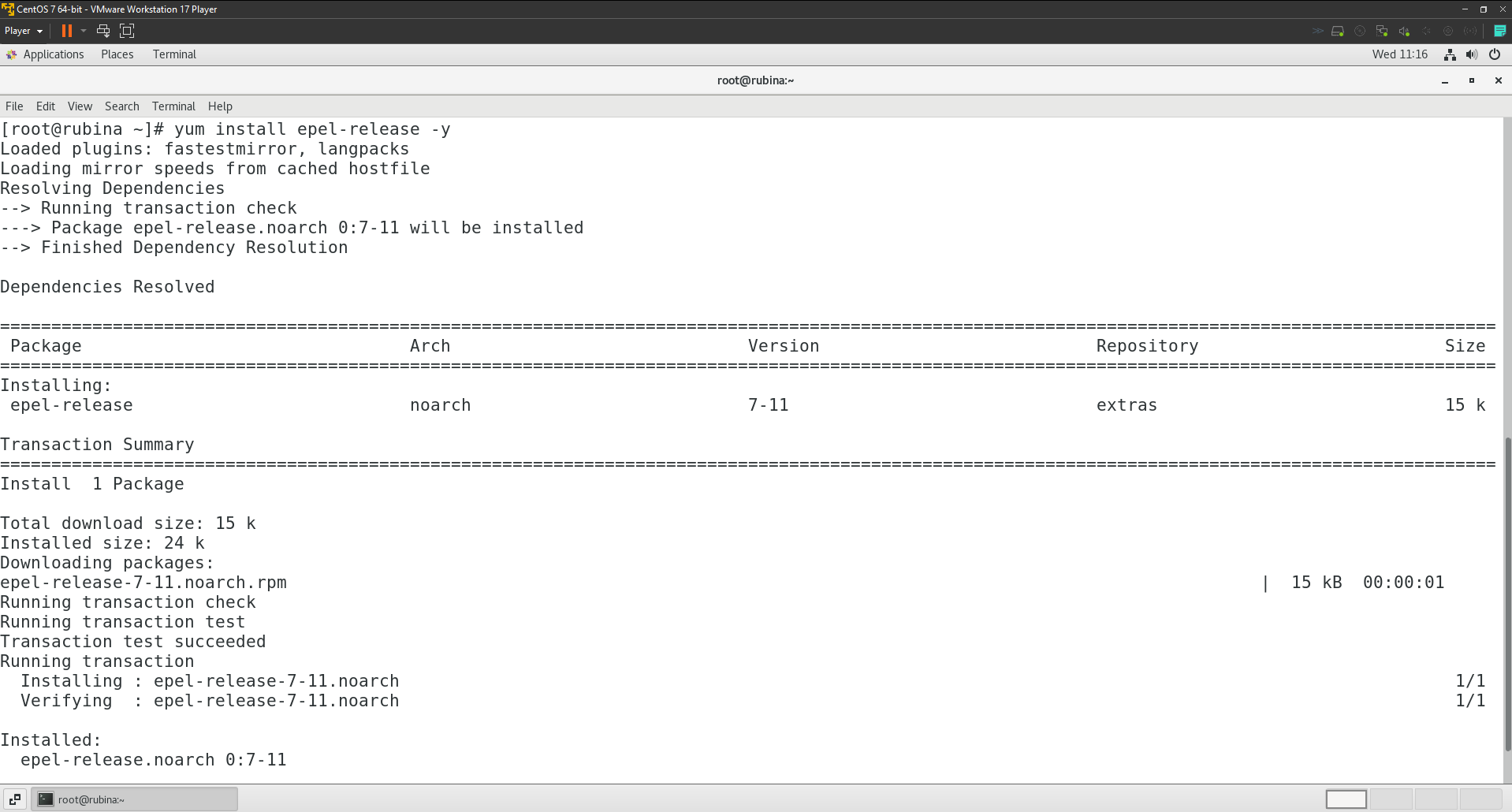
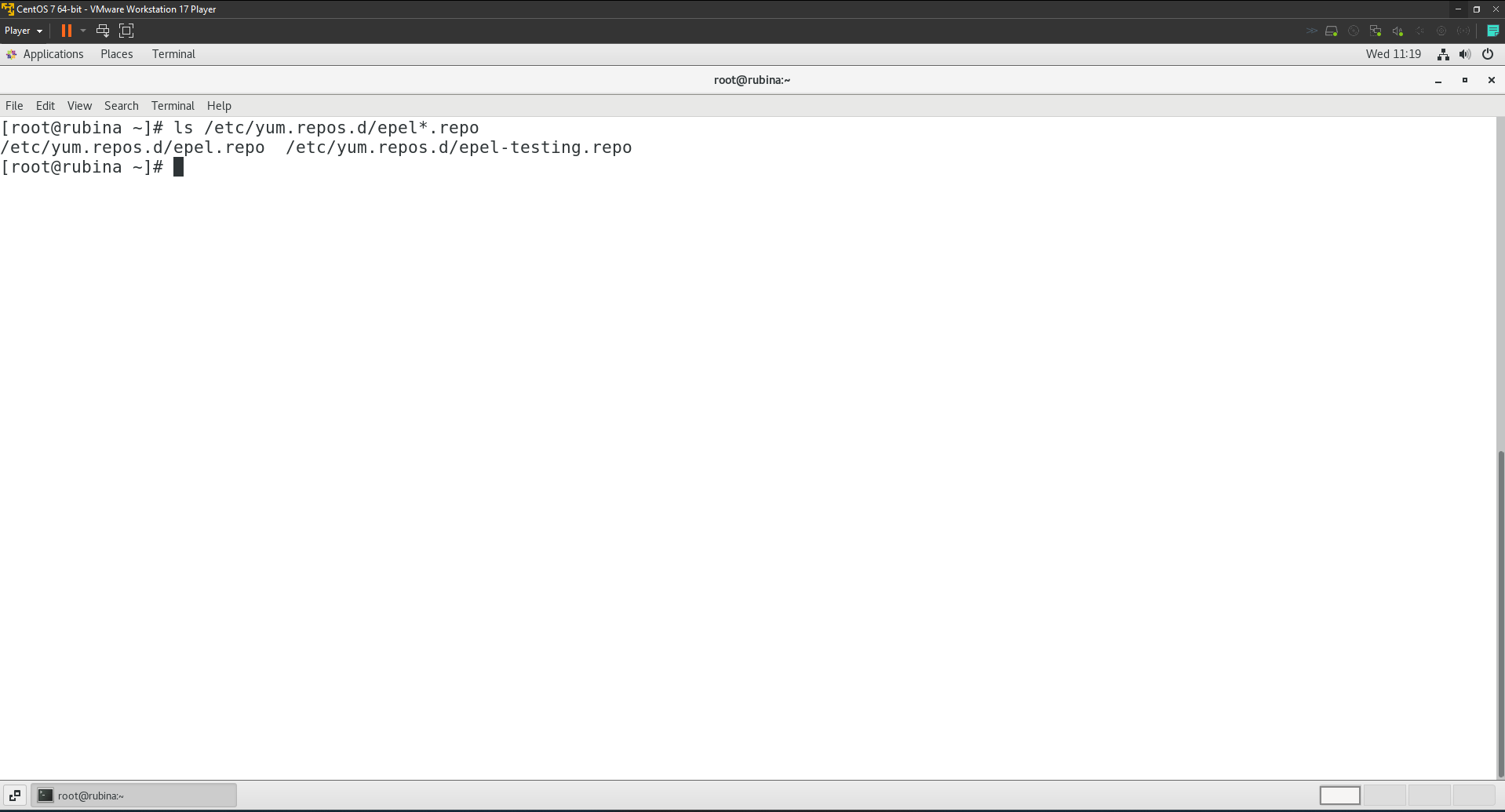
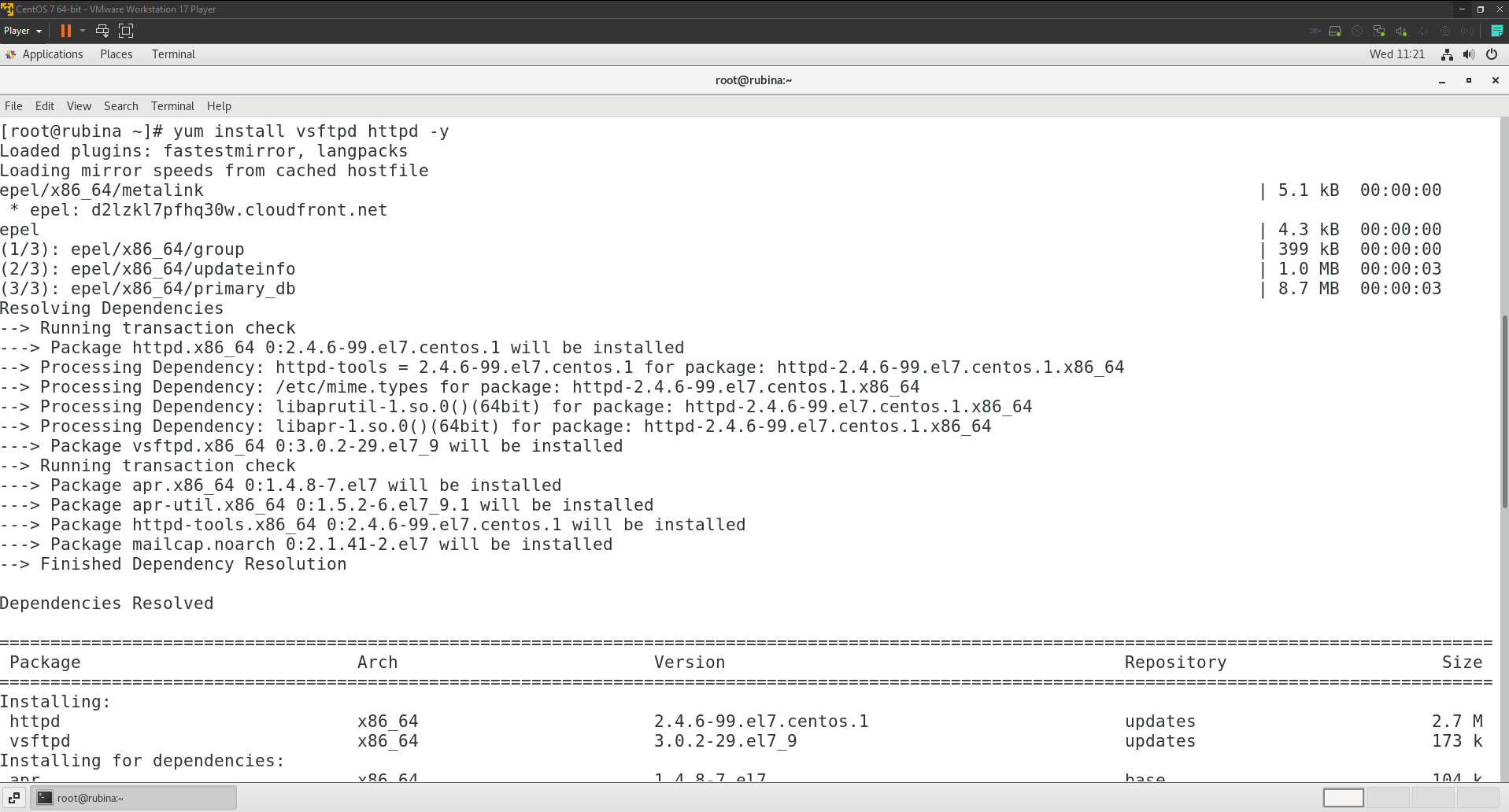
gpgcheck=1

gpgkey=https://dl.google.com/linux/linux\_signing\_key.pub

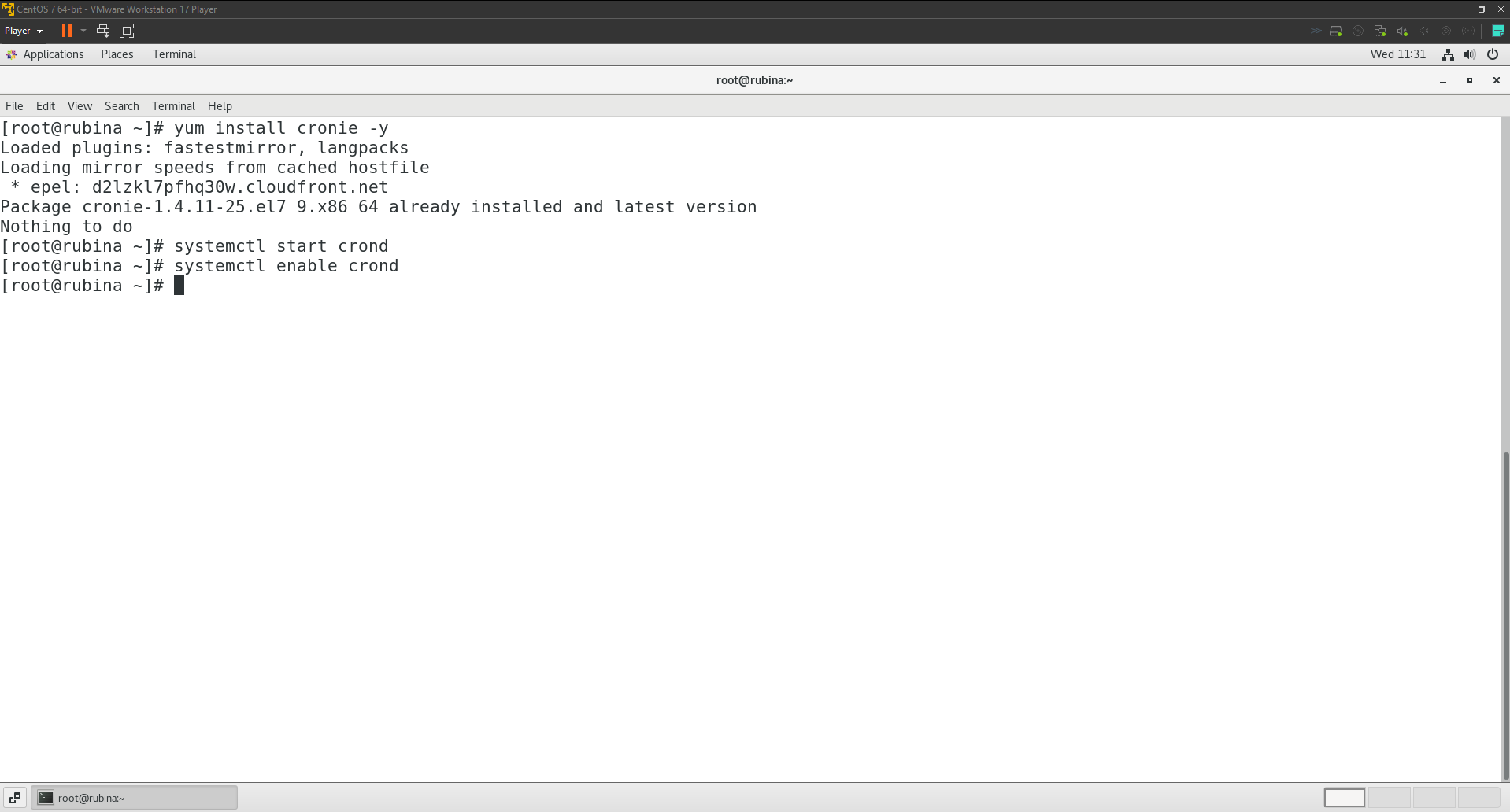


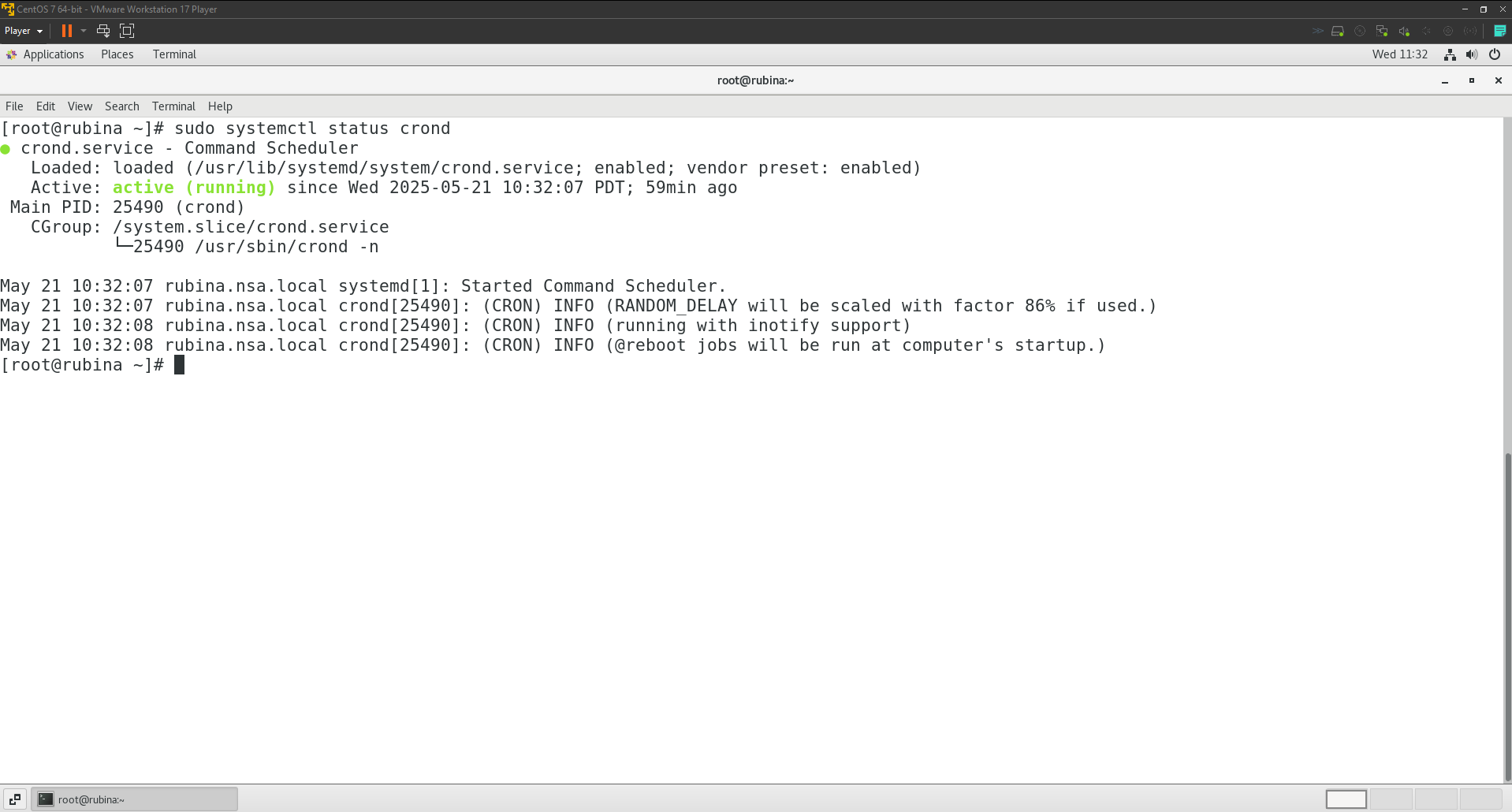
* 1. After creating the repo for chrome, I installed it using this command.  
     yum install google-chrome-stable -y  
       
     Verifying the installation



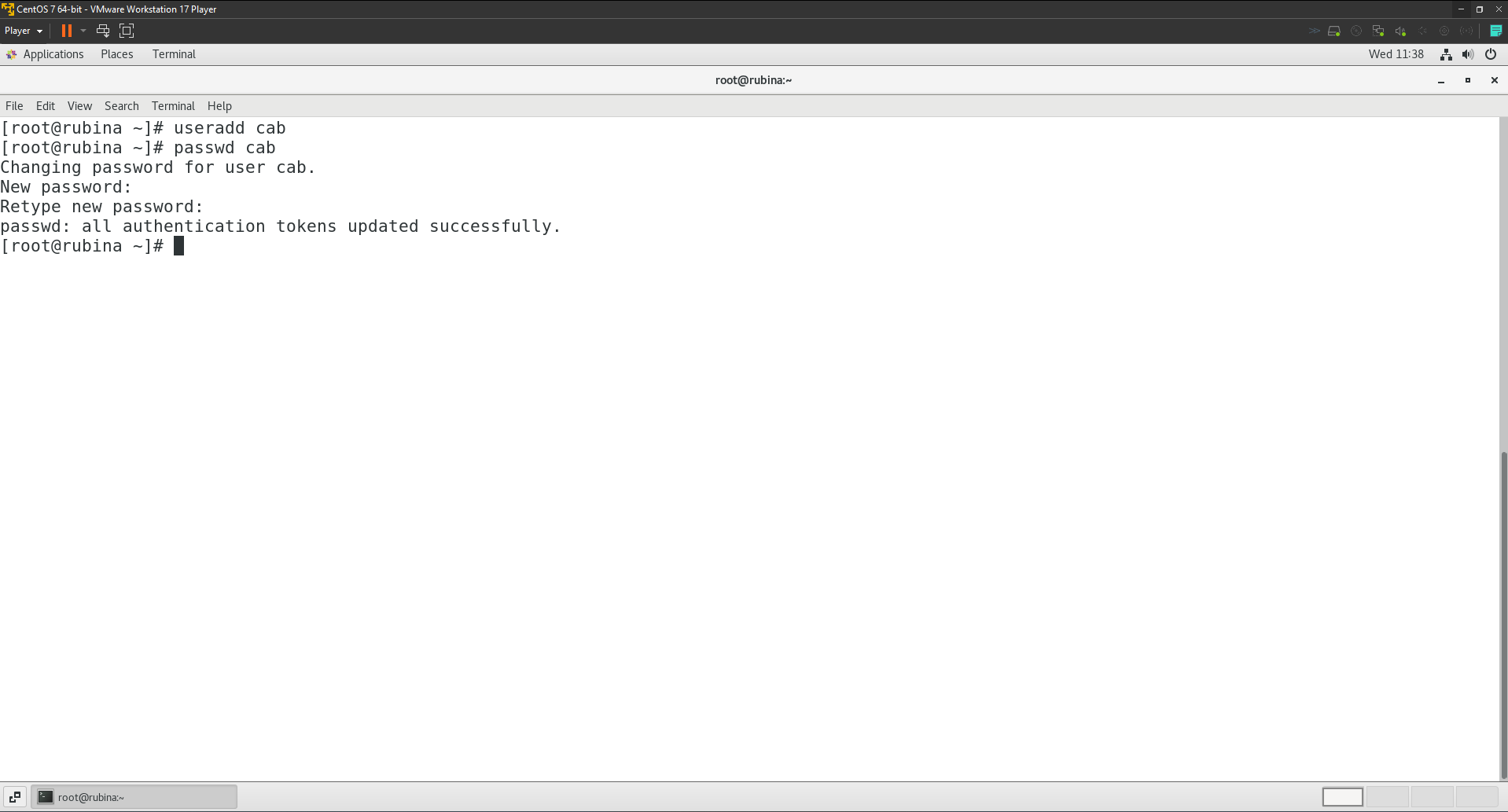
1. Enable EPEL repository (epel-release package) and verity the repo configuration in /etc/yum.repos.d then install vsftpd, httpd package using YUM command.
   1. Firstly, I installed epel-release package using following command:  
      yum install epel-release -y  
      
   2. Secondly, I verified the repo configuration in using:  
      ls /etc/yum.repos.d/epel\*.repo  
      
   3. Lastly, I installed vsftpd and httpd package using the following command:  
      yum install vsftpd httpd -y  
      
2. Uninstall the package httpd.  
   yum remove httpd -y  
   

# **Task 11**

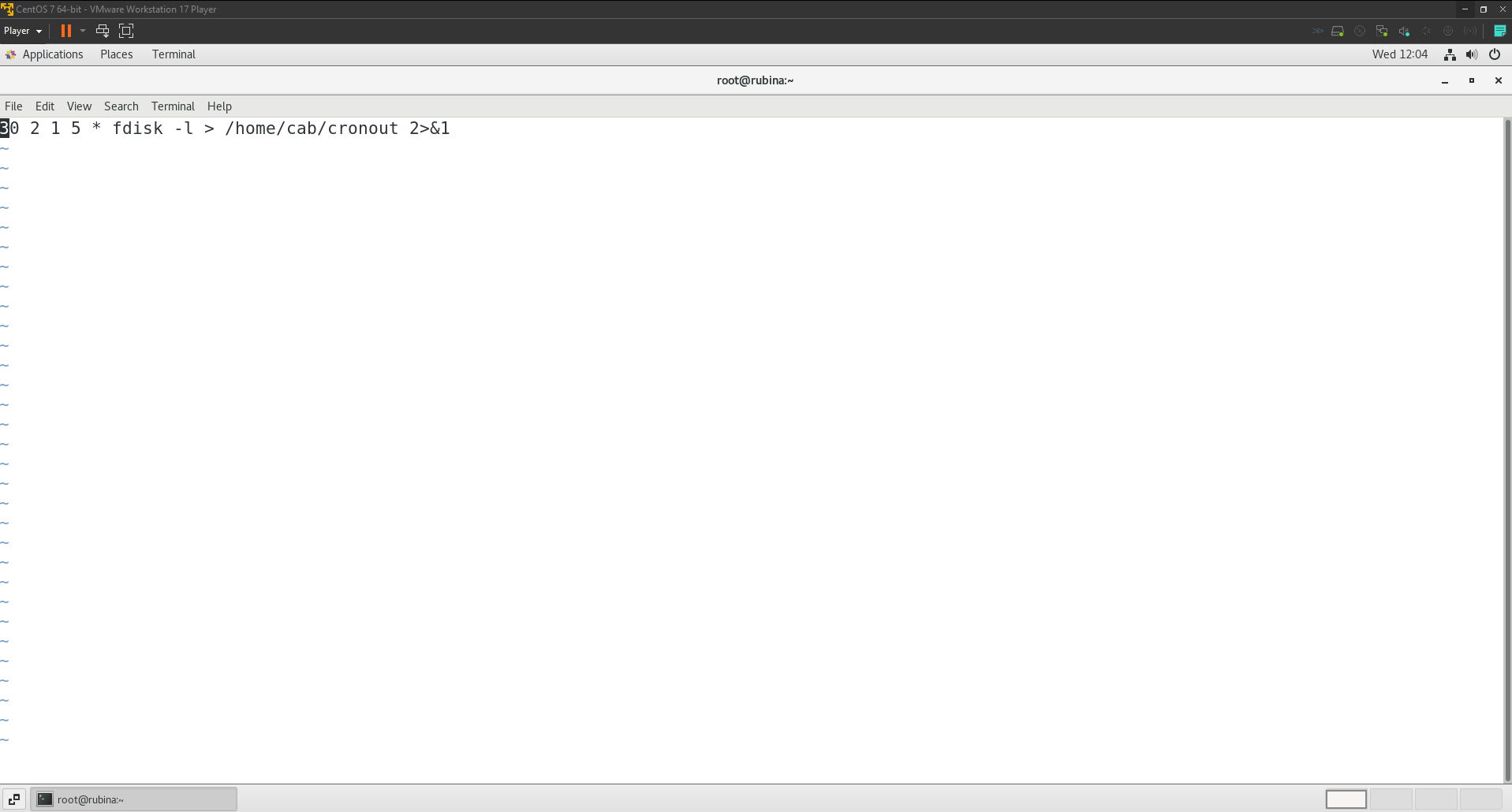
1. Install, start and enable the required packages and services required for the cron job automations.
   1. I installed the cronie package, started the cron service and enabled the cron service to start at the boot using the following commands respectively:  
      yum install cronie -y  
      systemctl start crond  
      systemctl enable crond  
      
   2. Verifying if the cron is running:

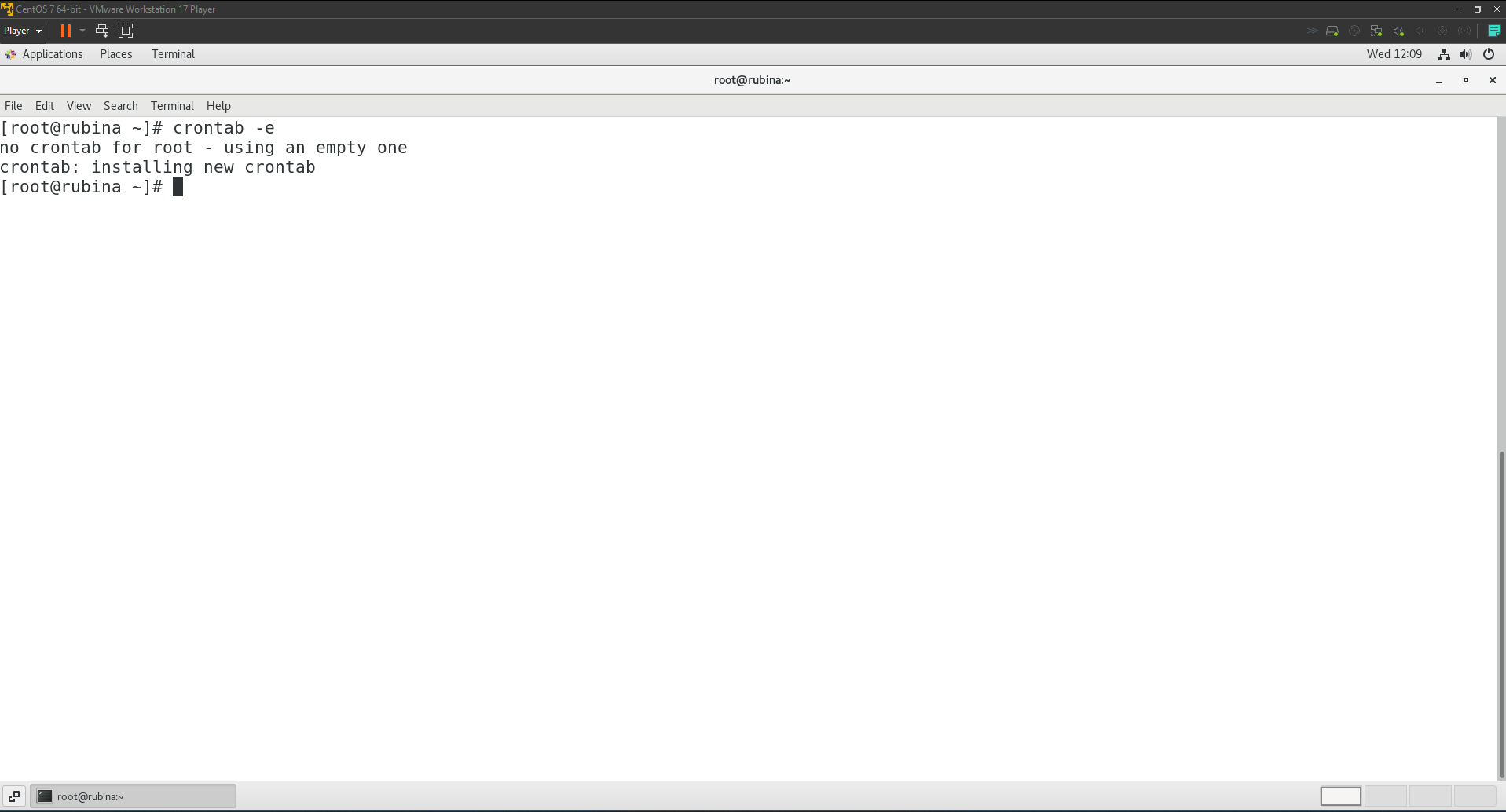
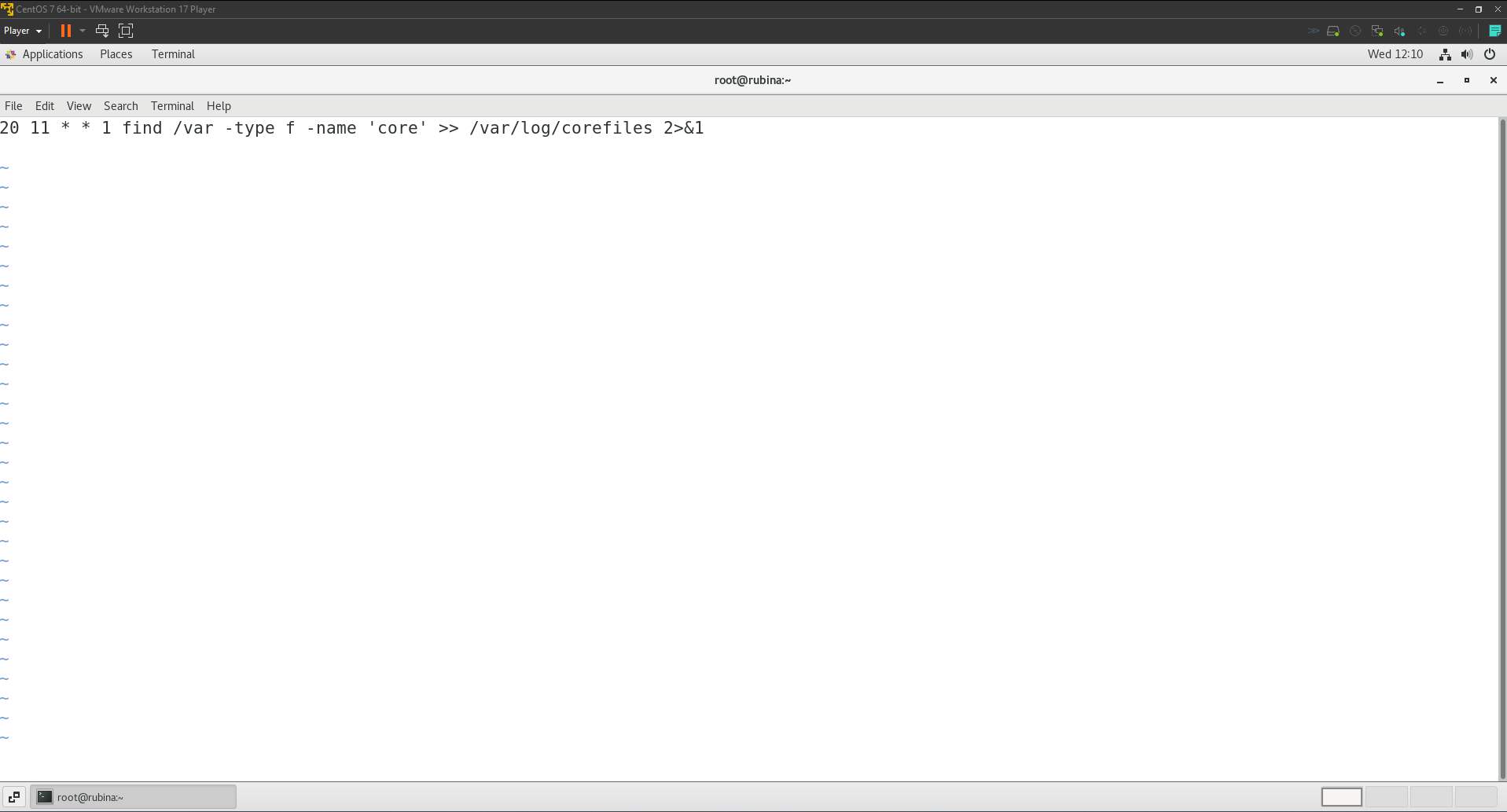
systemctl status crond  


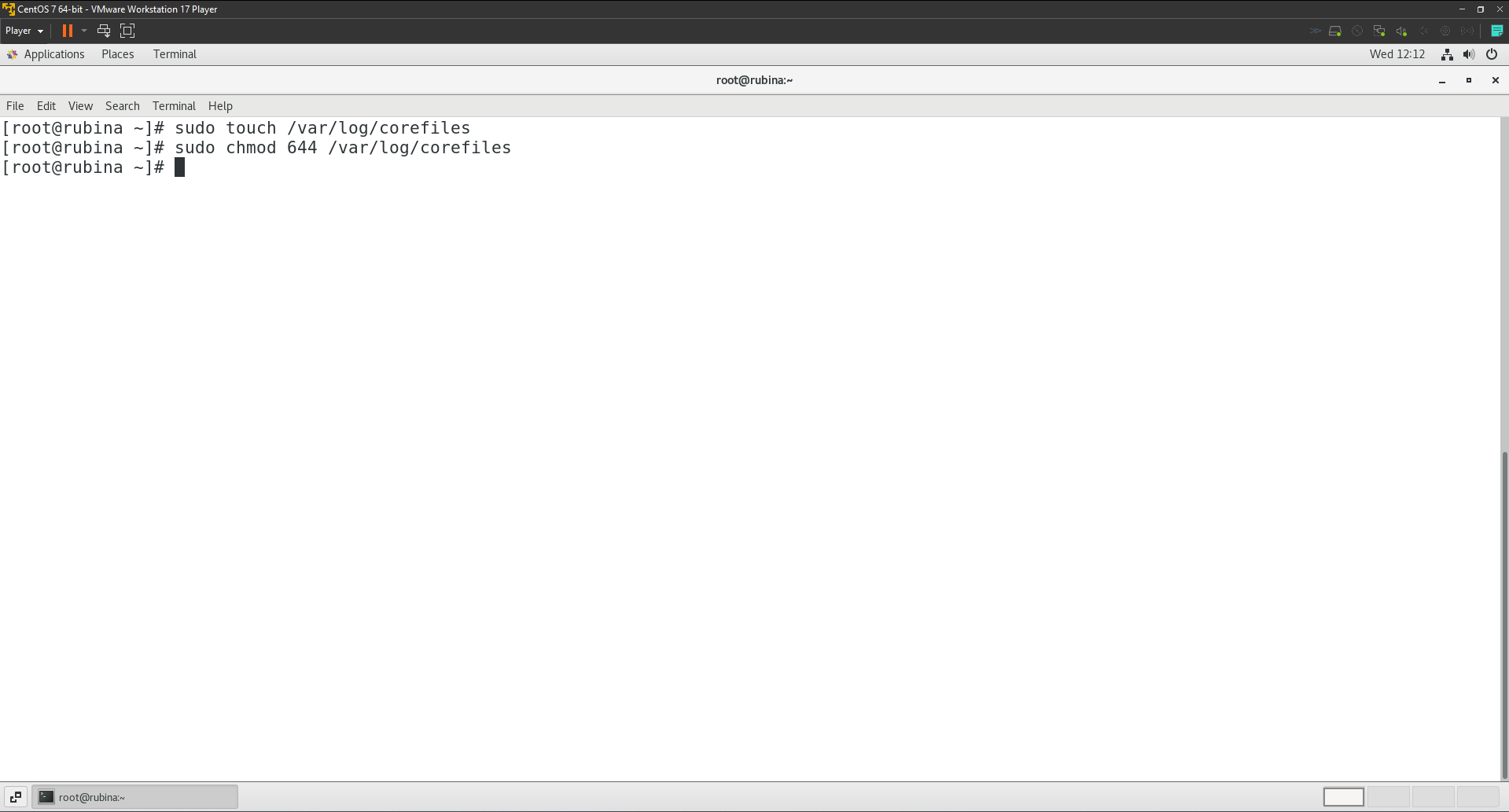
1. Schedule a cron job as user cab so that the information of used and free disk space will be saved into the file /home/cab/cronout at 2:30 AM on the first day of every May. [Hint: fdisk –l]
   1. I created user ‘cab’ using the following commands:  
      useradd cab

passwd cab  


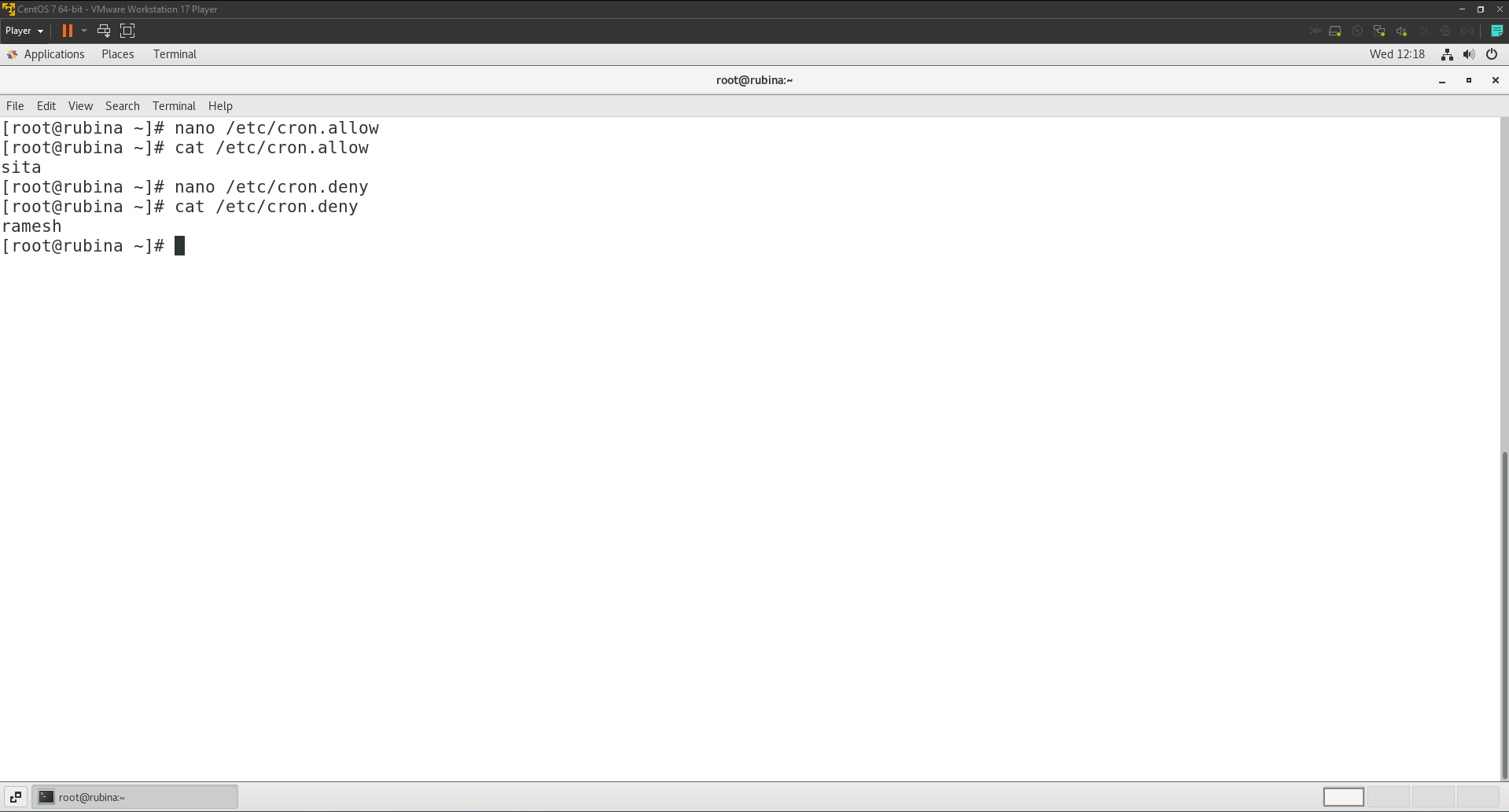
* 1. After which I edited the crontab of ‘cab’ user and added the cron job  
     sudo crontab -u cab -e  
     

Content of cron job   
30 2 1 5 \* fdisk -l > /home/cab/cronout 2>&1  


1. Set up a cron job as the root user to search for core files in the /var directory and list them in /var/log/corefiles. The job should run every Monday at 11:20 am system time.
   1. crontab -e  
      Content of cron job  
      20 11 \* \* 1 find /var -type f -name 'core' >> /var/log/corefiles 2>&1
   2. Ensuring that /var/log/corefiles exists and is writable by the root user.

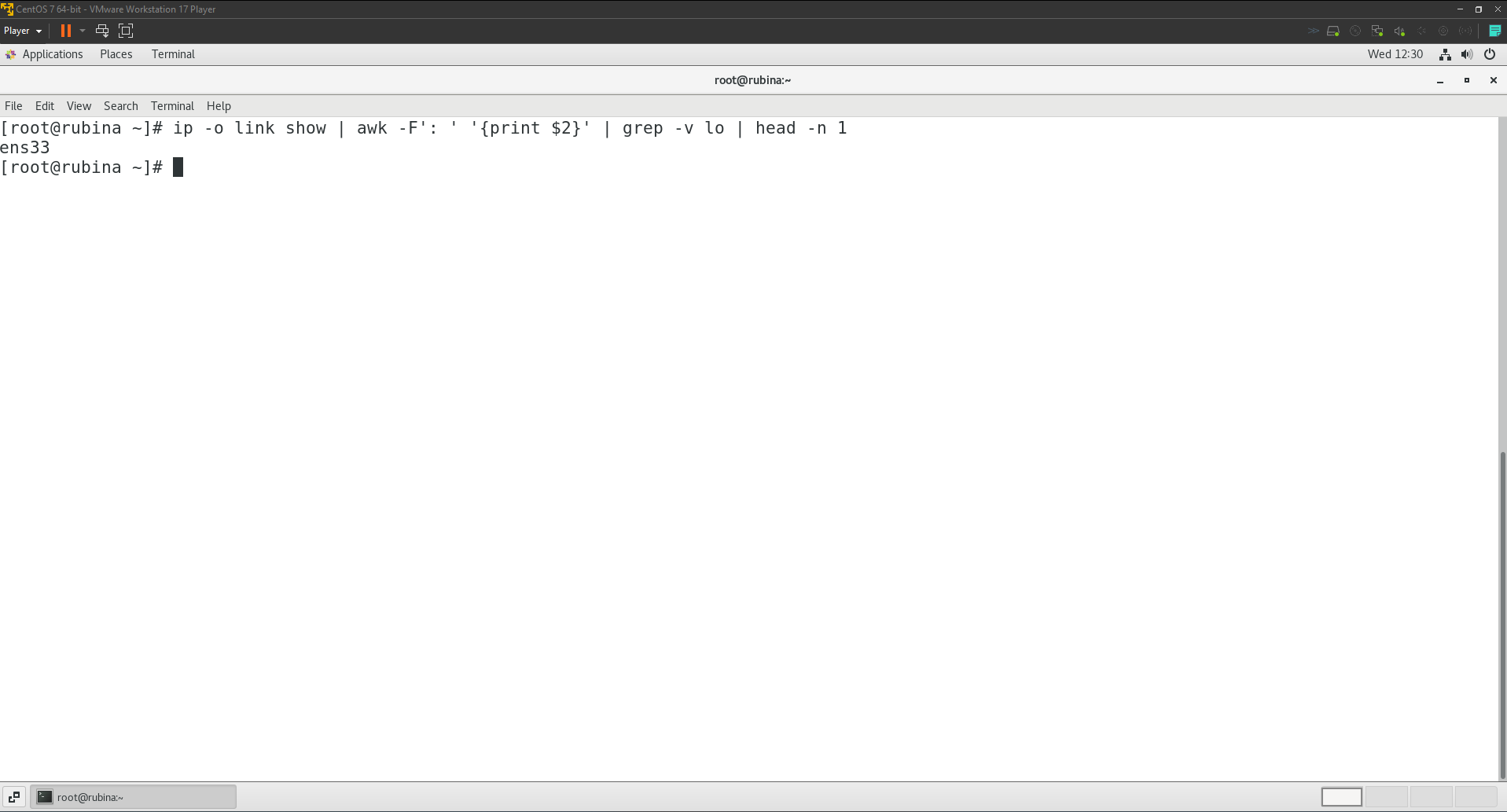
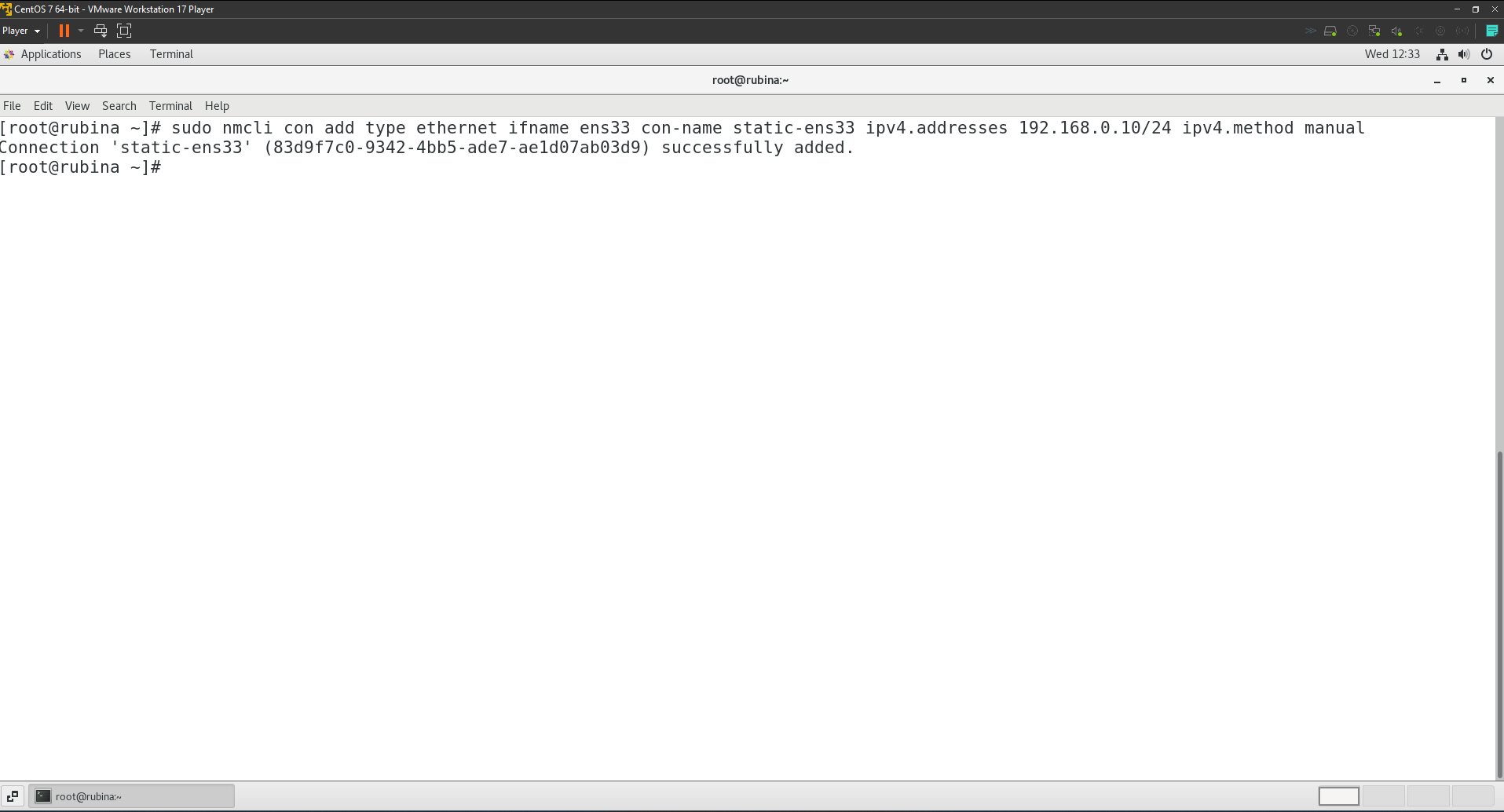
sudo touch /var/log/corefiles  
sudo chmod 644 /var/log/corefiles  


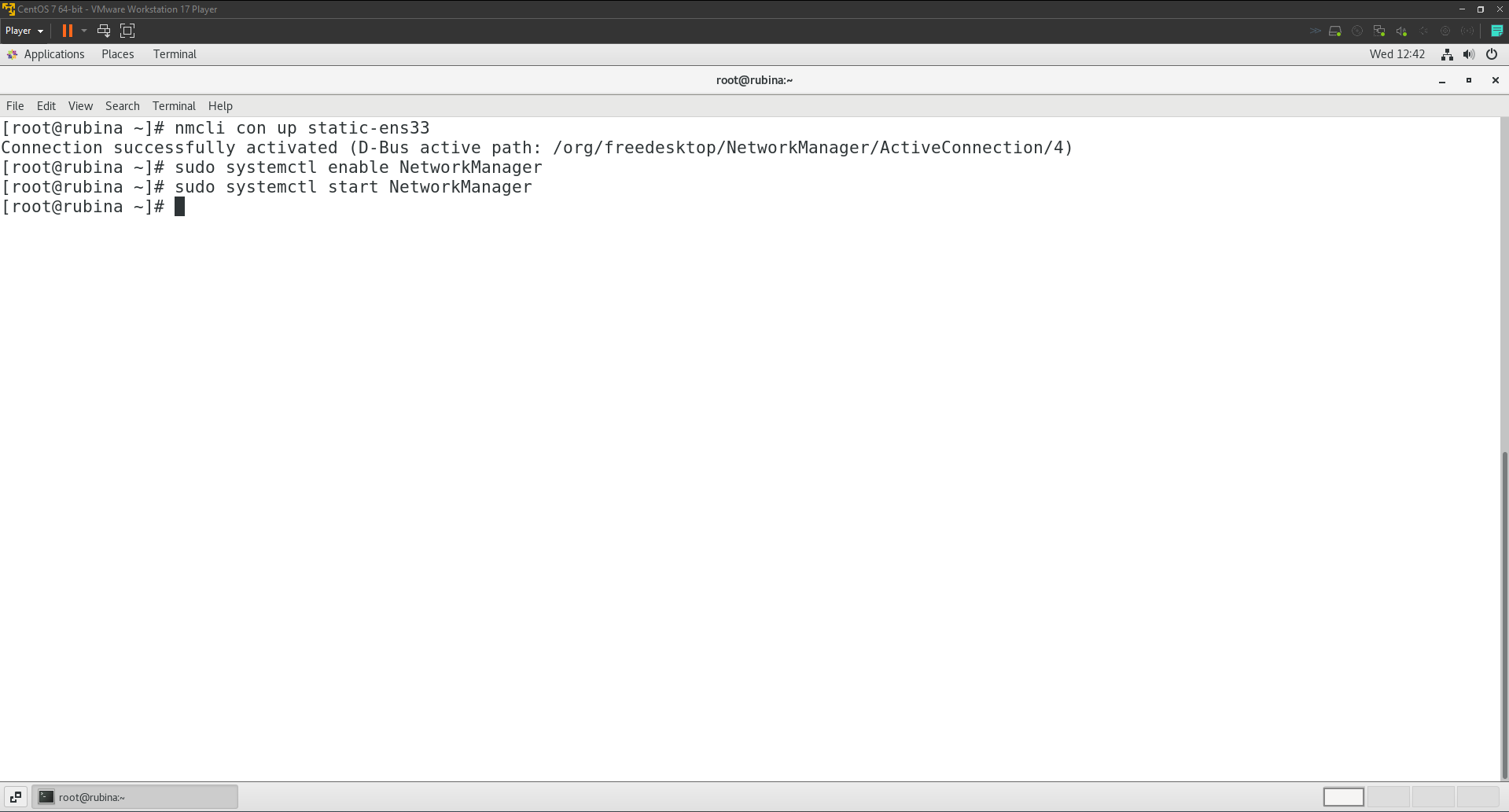
1. Enable cron access for user sita and deny for user ramesh.  
   To allow the user ‘sita’ to cron access, I added her name to /etc/cron.allow file. Likewise, to deny the user ‘ramesh’ cron access, I added his name to /etc/cron.deny.

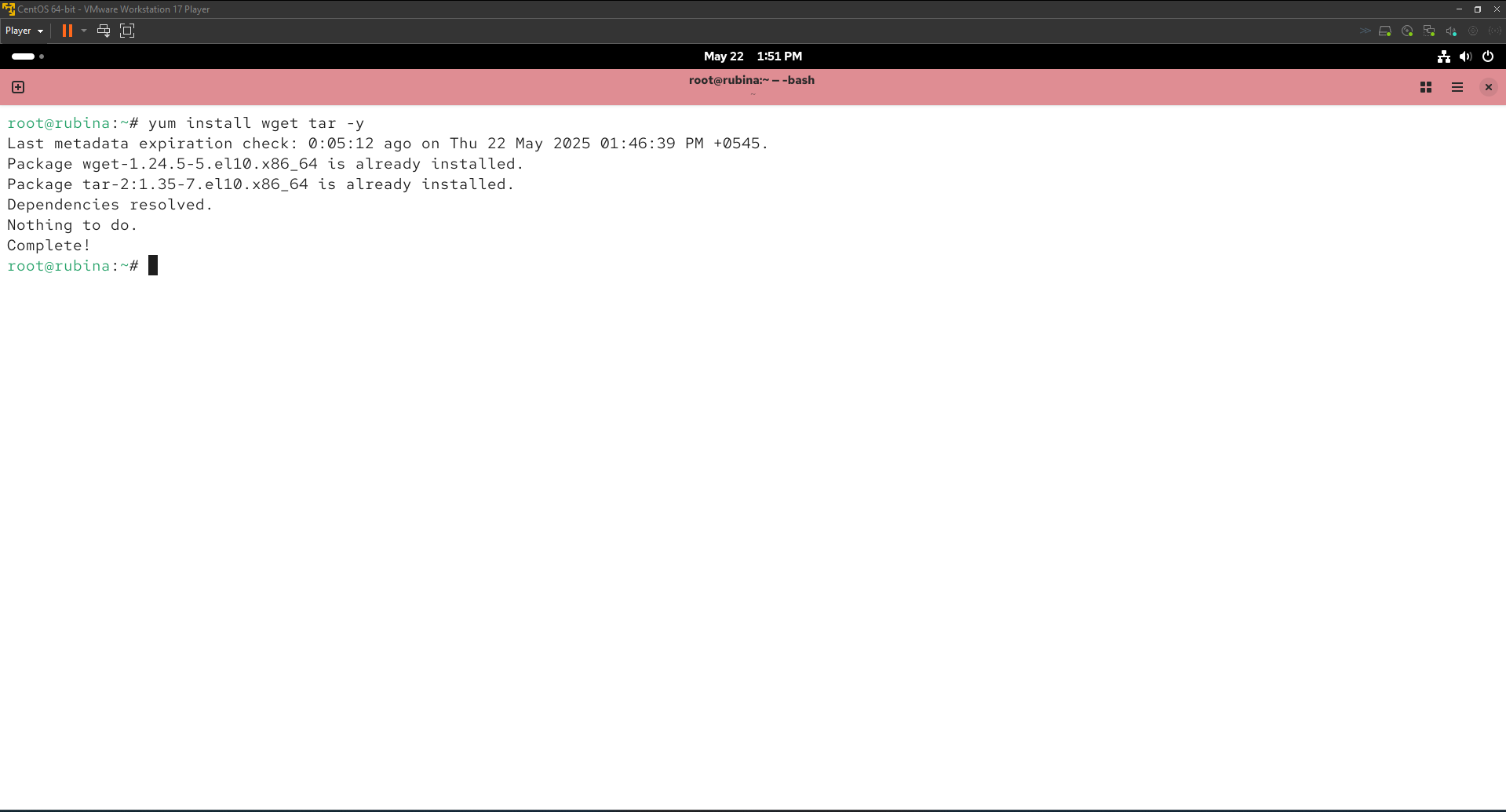
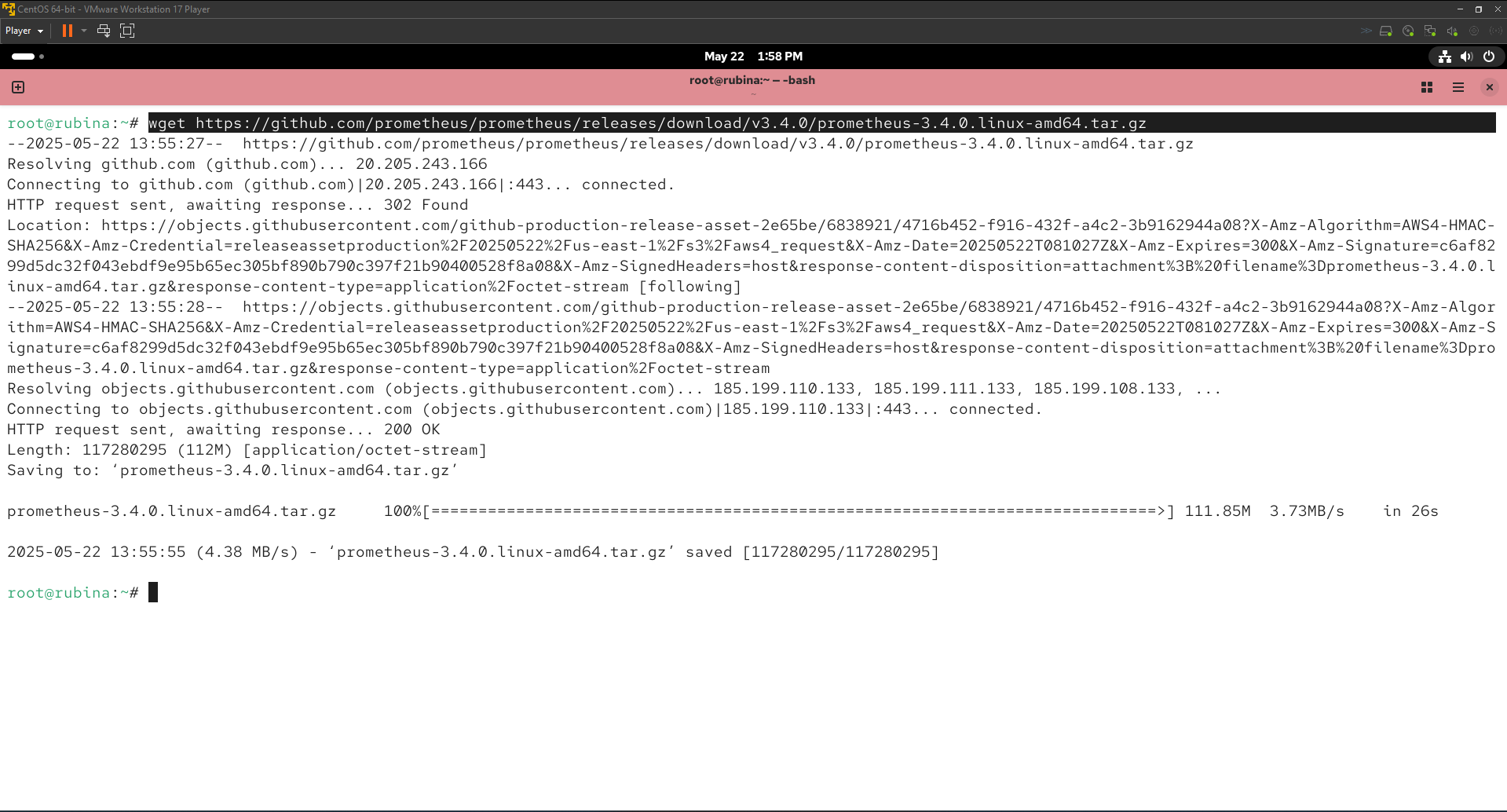
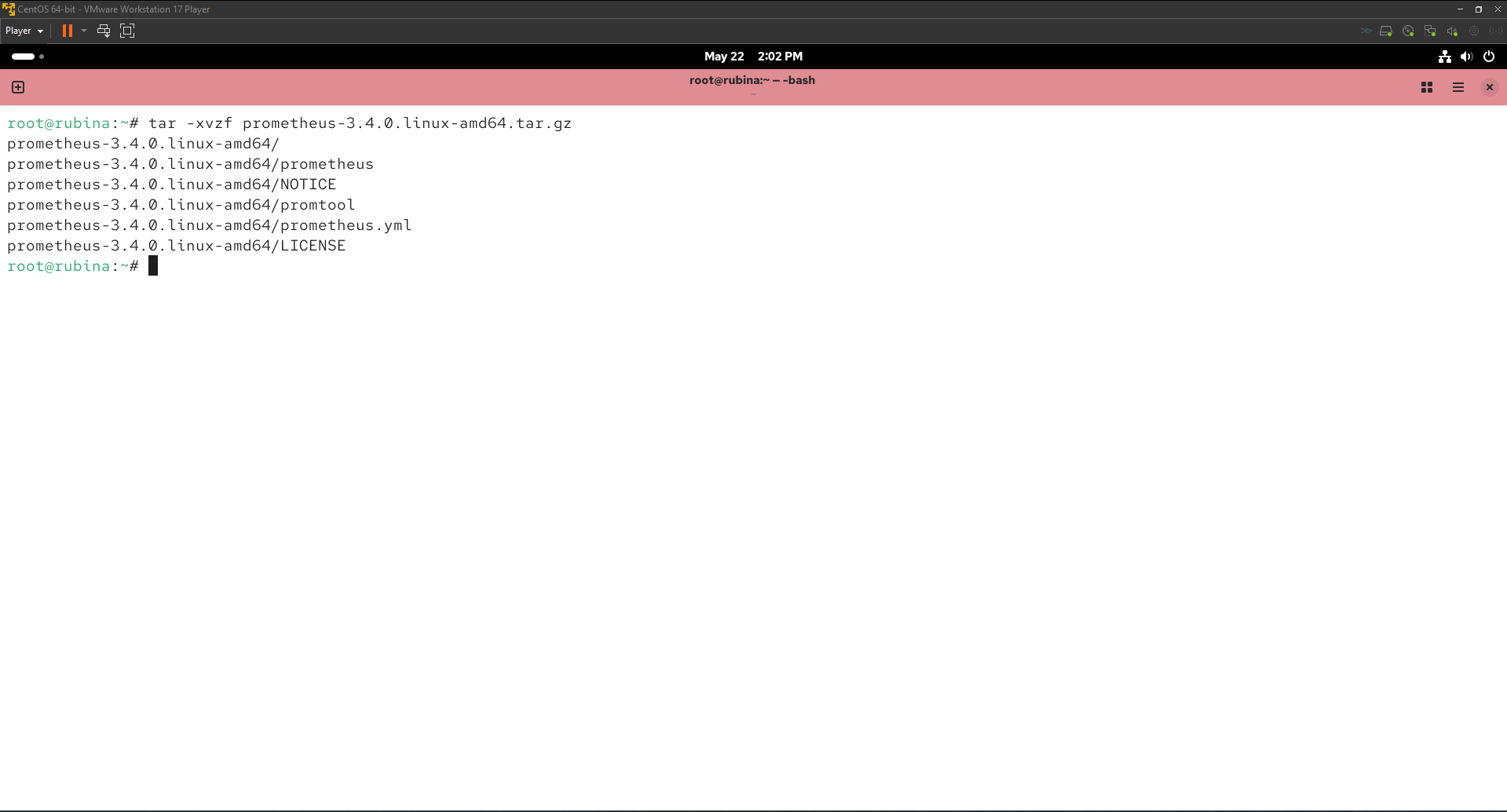


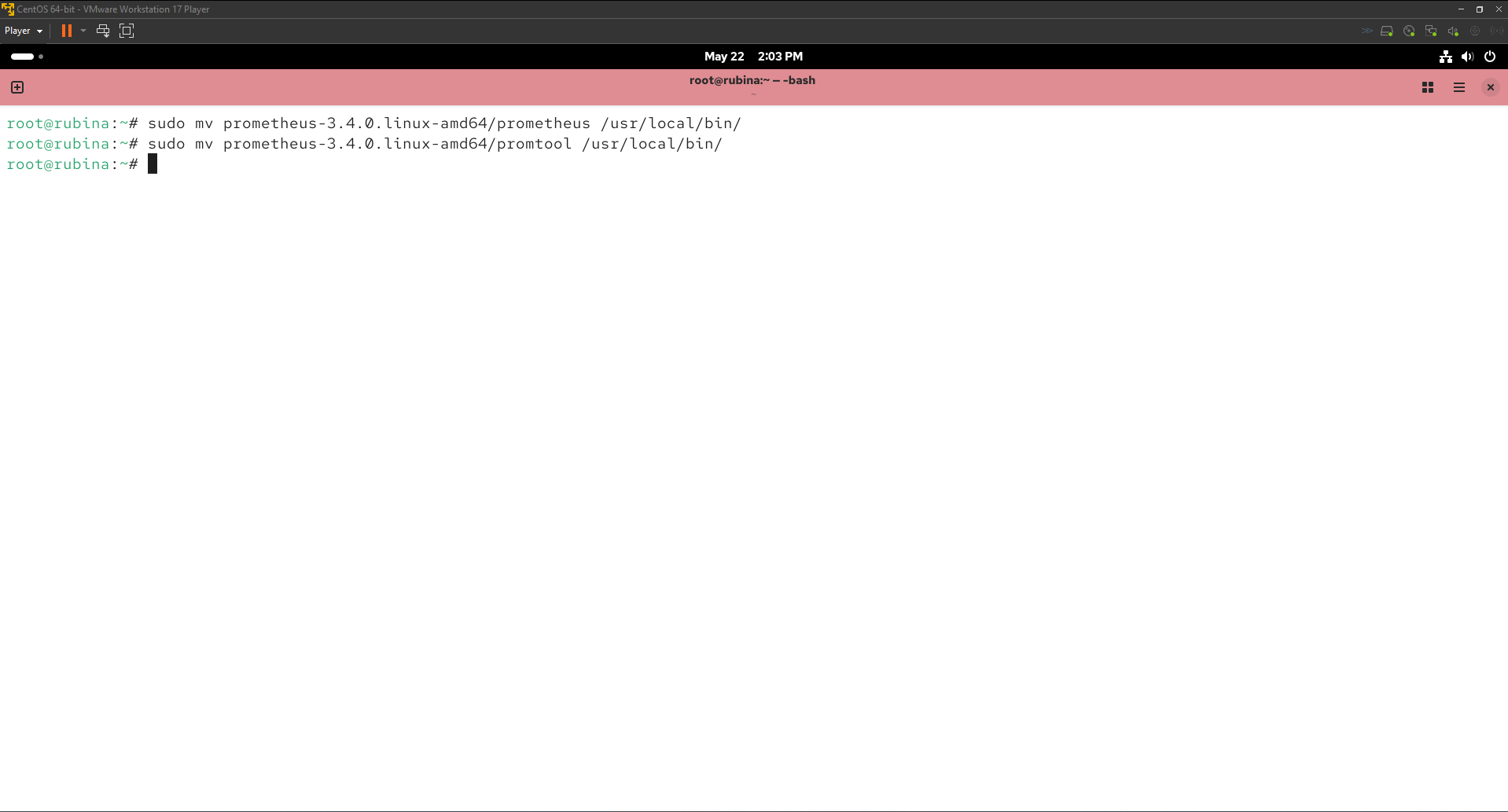
# **Task 12**

## **Network configuration**

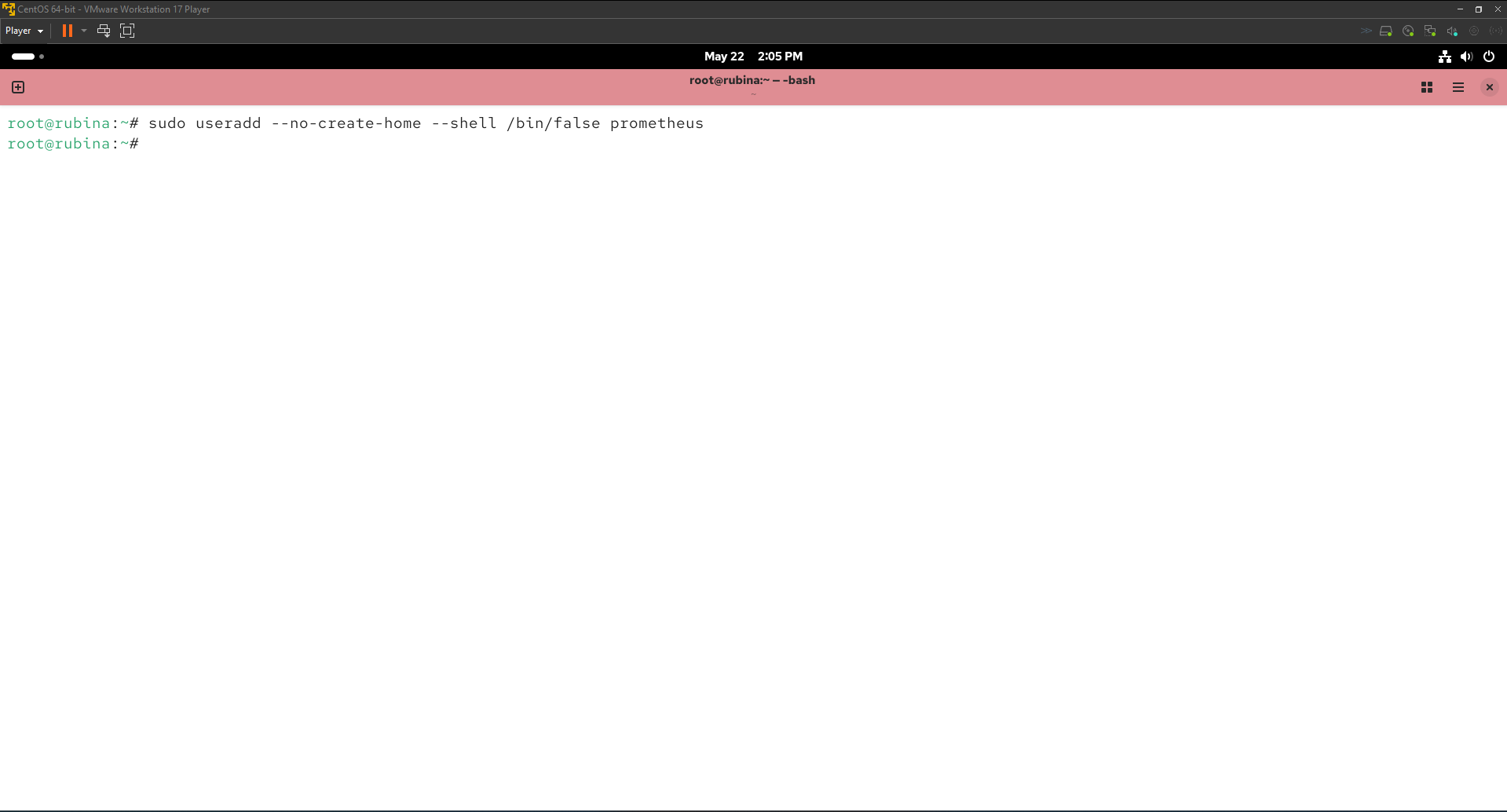
1. Configure the network interface with ipv4 address 192.168.0.10/24. Also start and enable the respective services.
   1. I searched for my actual network interface name using the following command:  
      ip -o link show | awk -F': ' '{print $2}' | grep -v lo | head -n 1  
      
   2. Then I added a static IP address to my actual network interface using the following command:  
      nmcli con add type ethernet ifname ens33 con-name static-ens33 ipv4.addresses 192.168.0.10/24 ipv4.method manual  
      
   3. Finally, I started and enabled the connection as well as NetworkManager service using the following commands:  
      nmcli con up static-ens33  
      sudo systemctl enable NetworkManager

sudo systemctl start NetworkManager  


1. Configure Zabbix or Nagios or Cacti or Prometheus or Observium server in your system. Also start and enable the respective services.  
   I chose to install Prometheus server in my system. The steps I followed are listed below.
   1. I updated the system packages to ensure that my system is up to date before I began the installation process using the following command.  
      sudo yum update -y  
      
   2. After which I downloaded Prometheus using the following command.  
      wget <https://github.com/prometheus/prometheus/releases/download/v3.4.0/prometheus3.4.0.linux-amd64.tar.gz>  
      
   3. After downloading the Prometheus tarball, I extracted it using the following command.  
      tar -xvzf prometheus-3.4.0.linux-amd64.tar.gz  
      
   4. Then I moved the extracted Prometheus binaries to /usr/local/bin for easier execution using the following commands.  
      sudo mv prometheus-3.4.0.linux-amd64/prometheus /usr/local/bin/  
      sudo mv prometheus-3.4.0.linux-amd64/promtool /usr/local/bin/



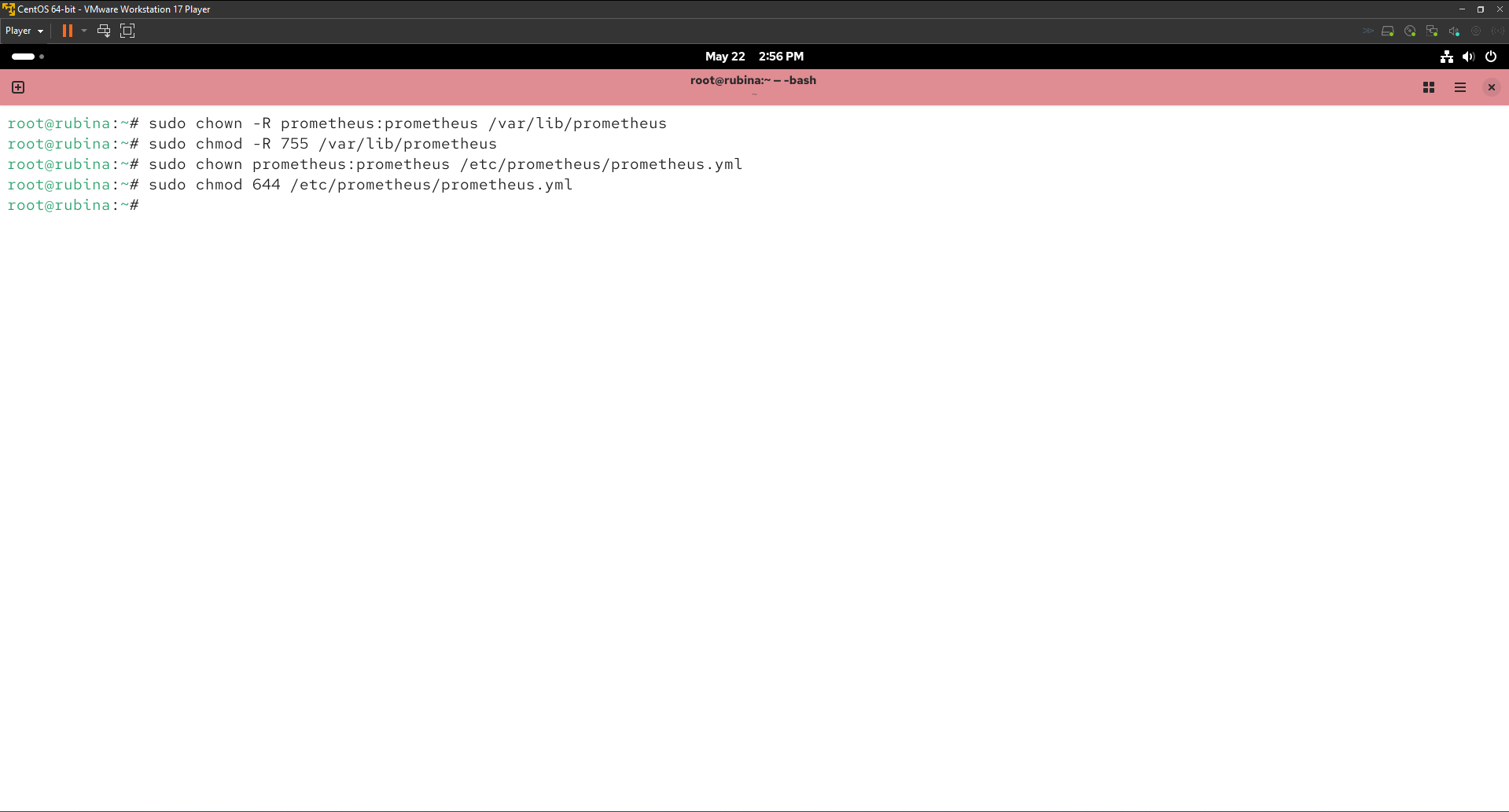
* 1. Since it is a good practice to run Prometheus under a non-privileged user, I created a new user ‘prometheus’ specifically for Prometheus using the following command.  
     sudo useradd --no-create-home --shell /bin/false prometheus



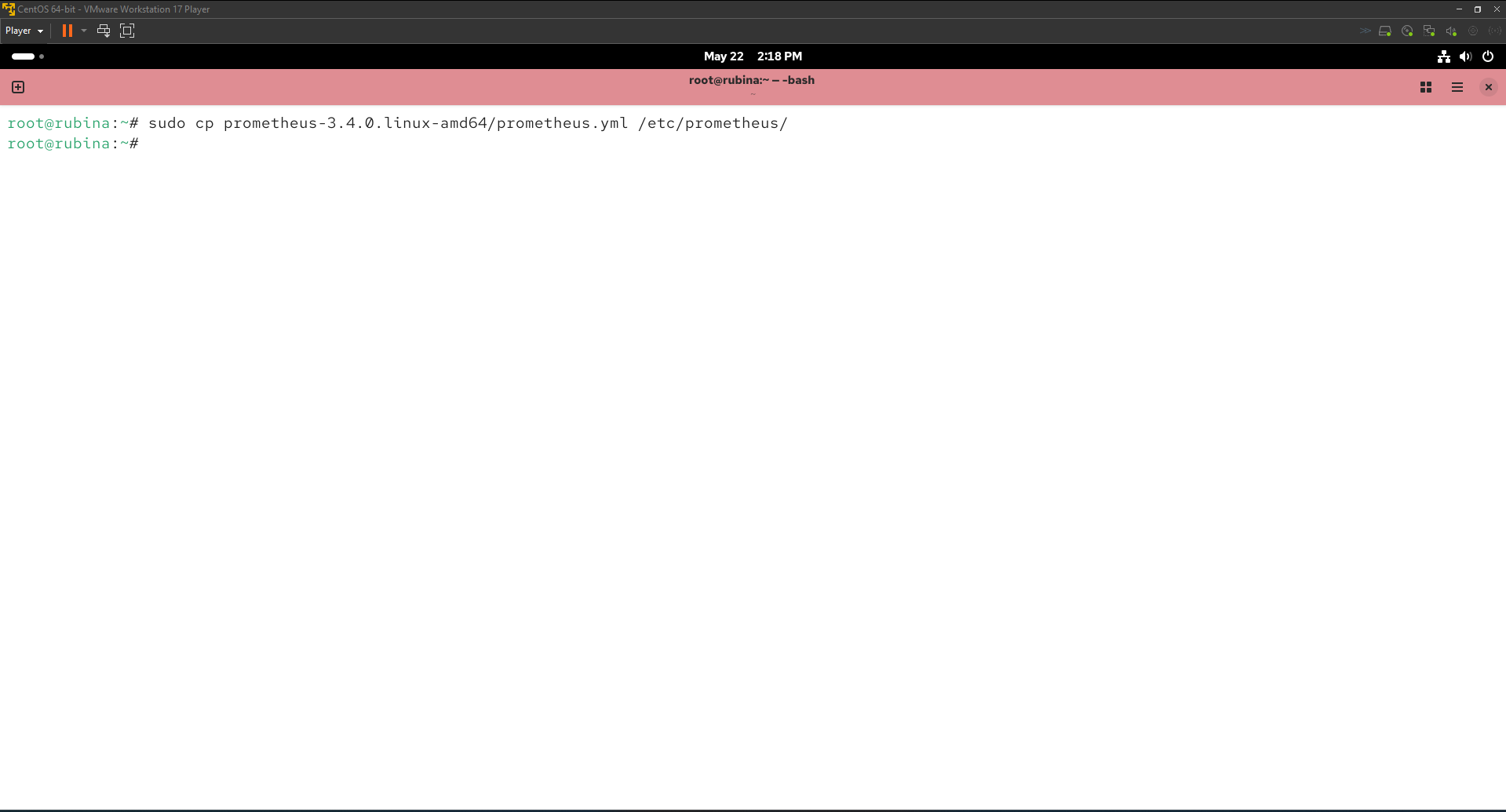
* 1. Since Prometheus needs certain directories to store its data and configuration files, I used the following command to create and give the ‘prometheus’ user necessary permissions to the aforementioned directories.  
     sudo mkdir /etc/prometheus

sudo mkdir /var/lib/Prometheus  
sudo chown prometheus:prometheus /etc/prometheus

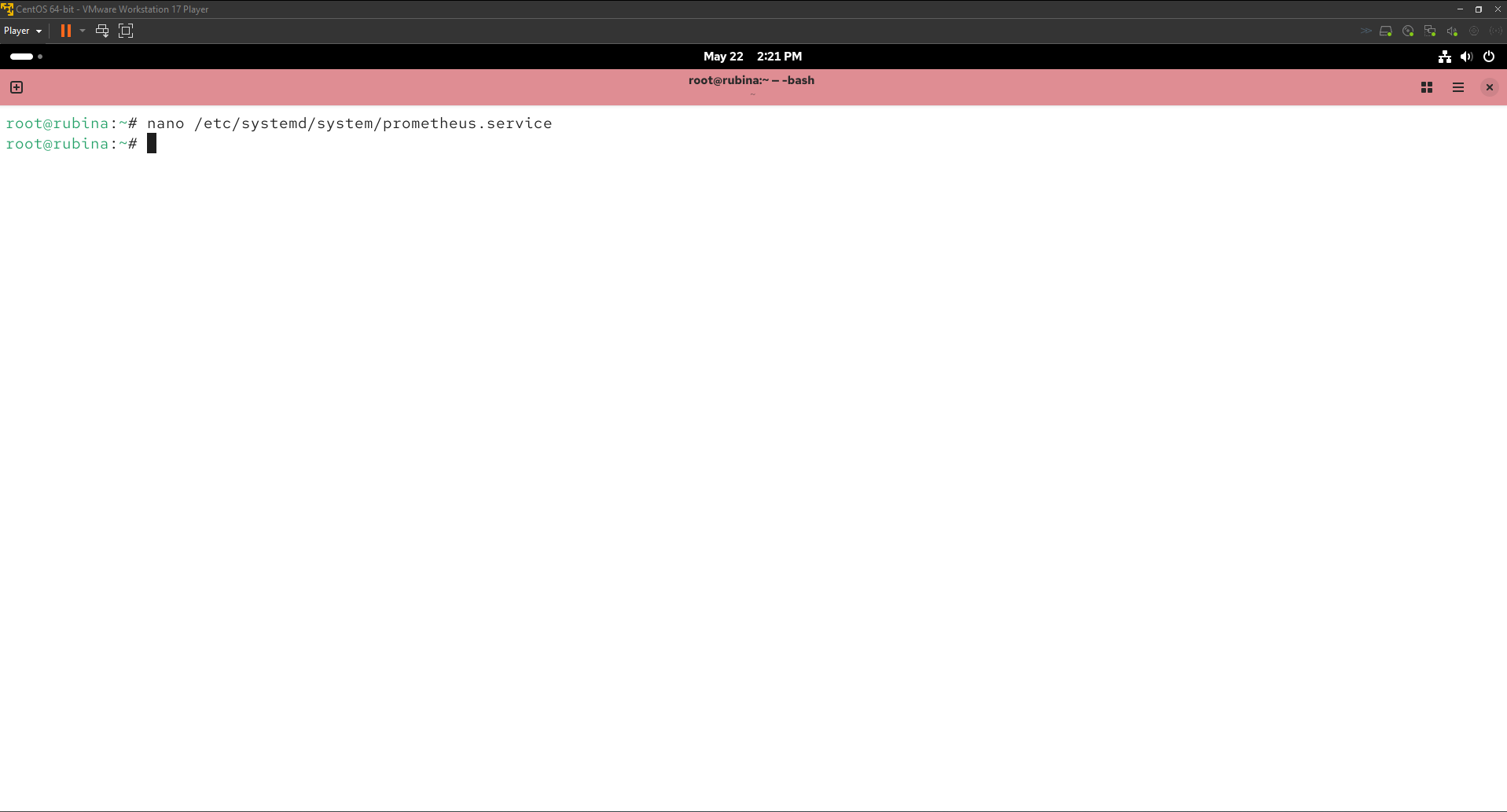
sudo chown prometheus:prometheus /var/lib/Prometheus  
sudo chmod -R 755 /var/lib/Prometheus  
sudo chmod 644 /etc/prometheus/prometheus.yml



* 1. After which I used the following command to copy the configuration (prometheus.yml) to /etc/Prometheus.  
     sudo cp prometheus-3.4.0.linux-amd64/prometheus.yml /etc/prometheus/



* 1. Since we need a new system service file to make Promethues run as a service, I created a prometheus.service using the following command.  
     nano /etc/systemd/system/prometheus.service

  
Content of the prometheus.service file  
[Unit]

Description=Prometheus Monitoring System and Time Series Database

Documentation=https://prometheus.io/docs/introduction/overview/

After=network.target

[Service]

User=prometheus

Group=prometheus

ExecStart=/usr/local/bin/prometheus \

--config.file=/etc/prometheus/prometheus.yml \

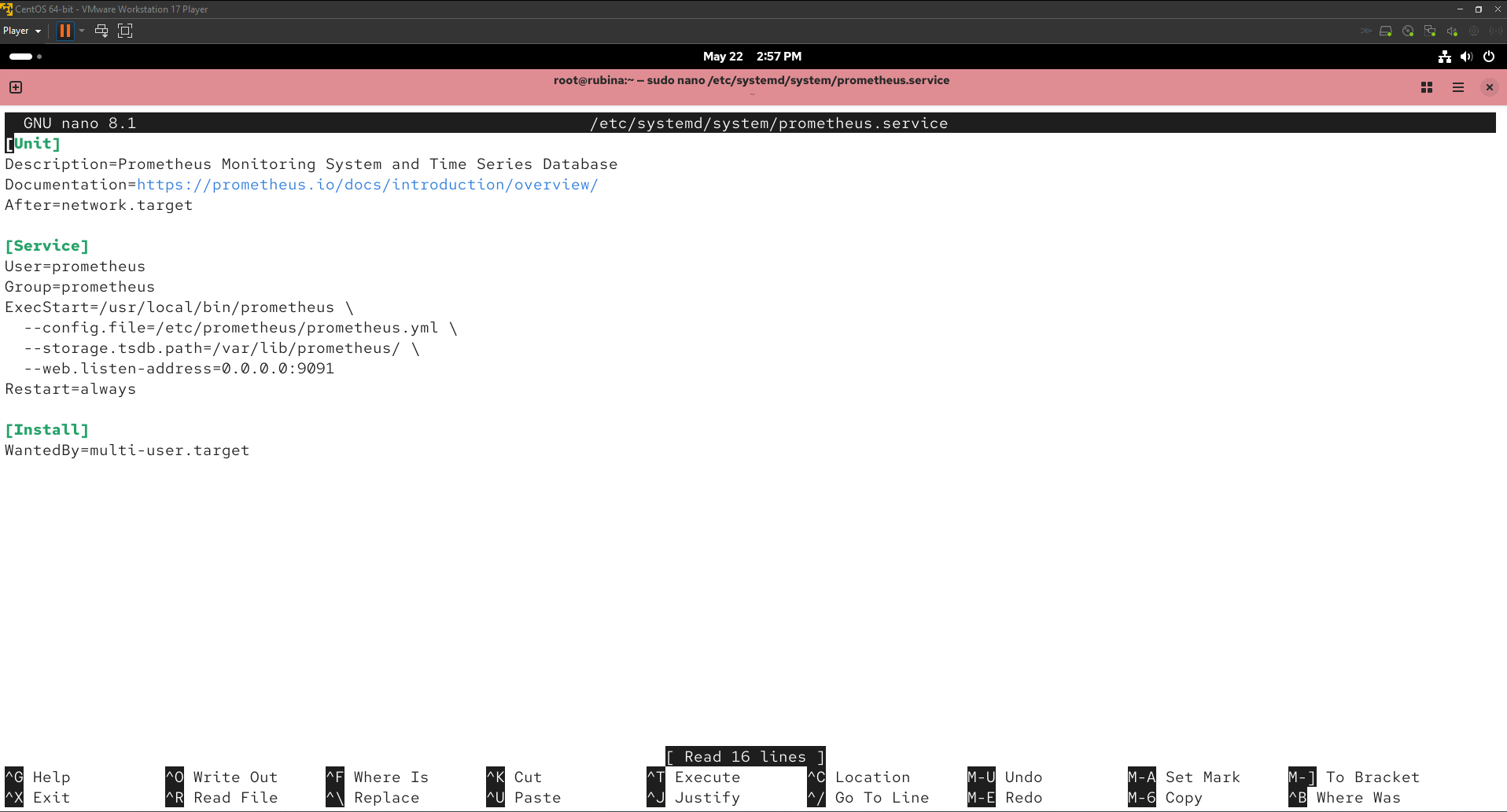
--storage.tsdb.path=/var/lib/prometheus/ \

--web.listen-address=0.0.0.0:9091

Restart=always

[Install]

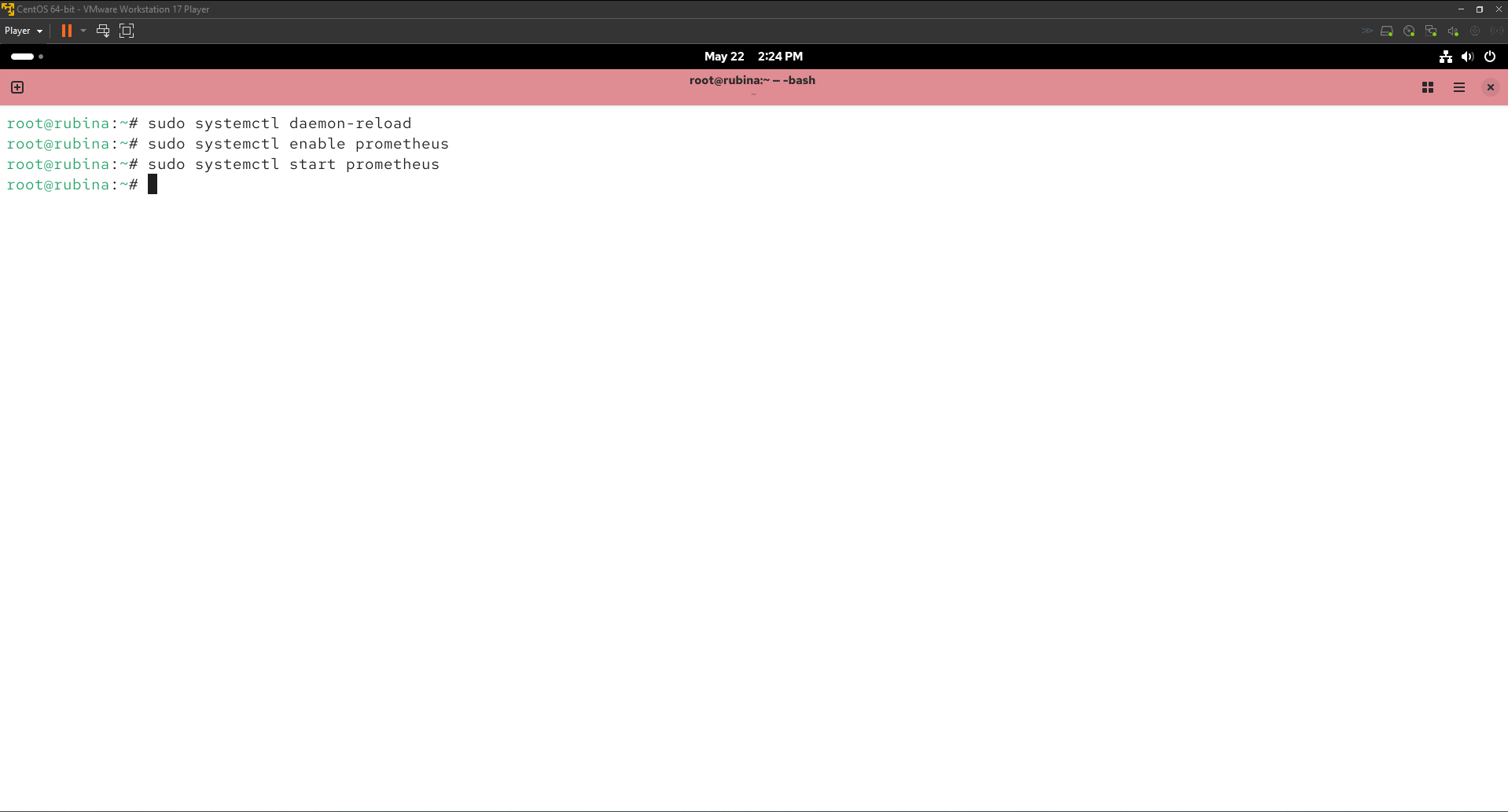
WantedBy=multi-user.target

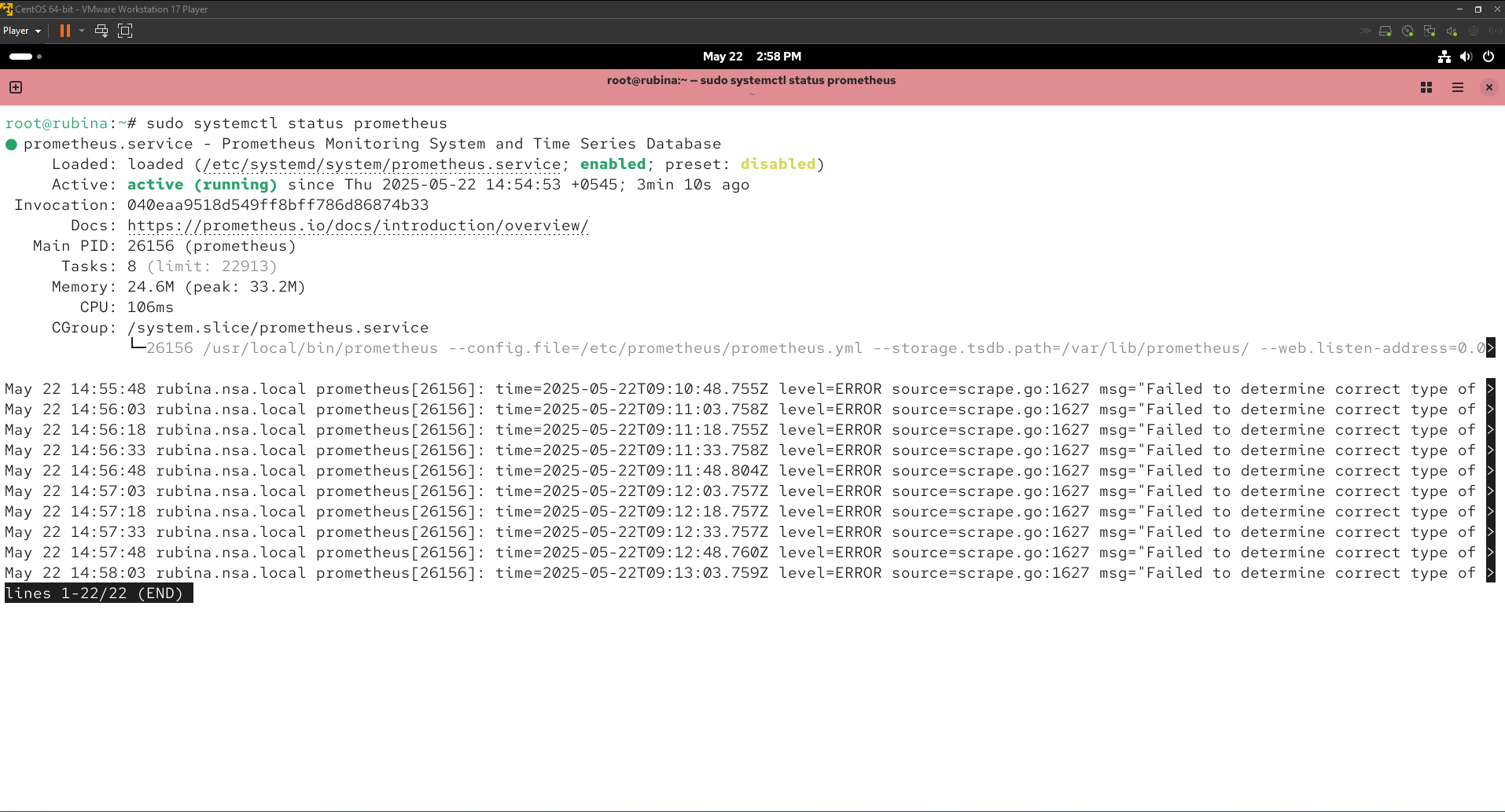


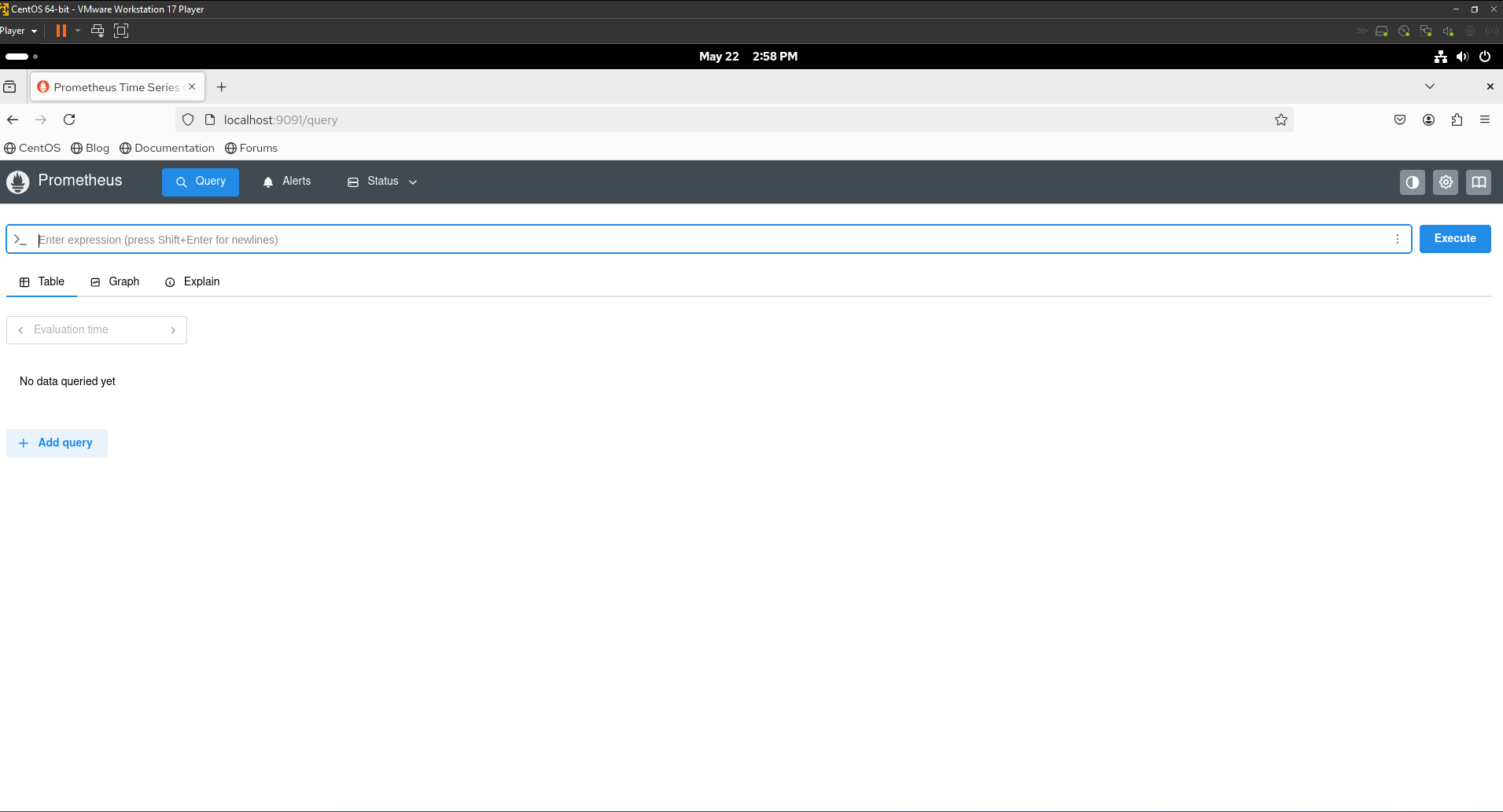
* 1. After creating the service file, I reloaded the system to apply the new configuration, enabled Promethues to start on boot and started the service using the following commands respectively.  
     sudo systemctl daemon-reload

sudo systemctl enable prometheus

sudo systemctl start Prometheus

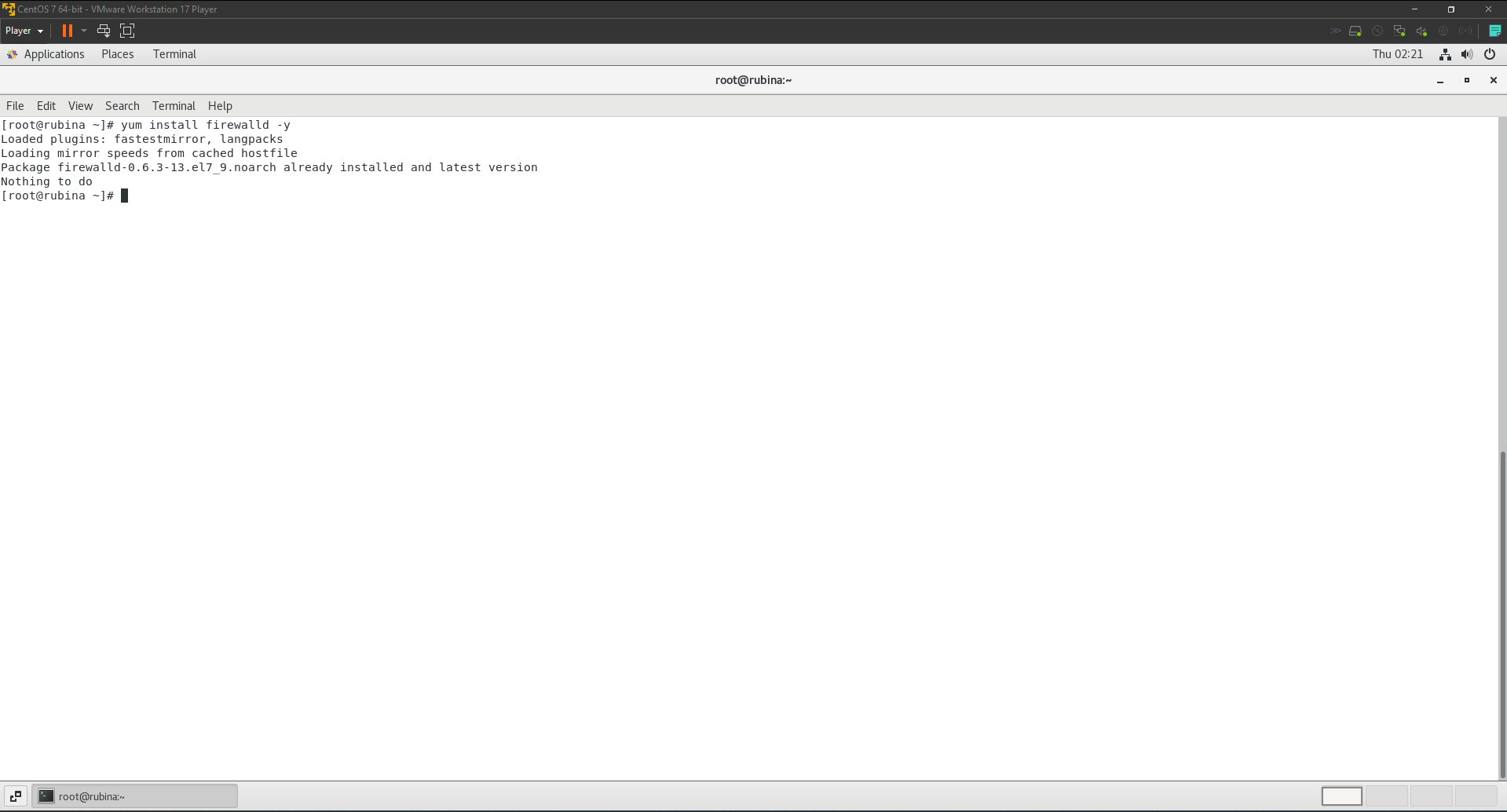


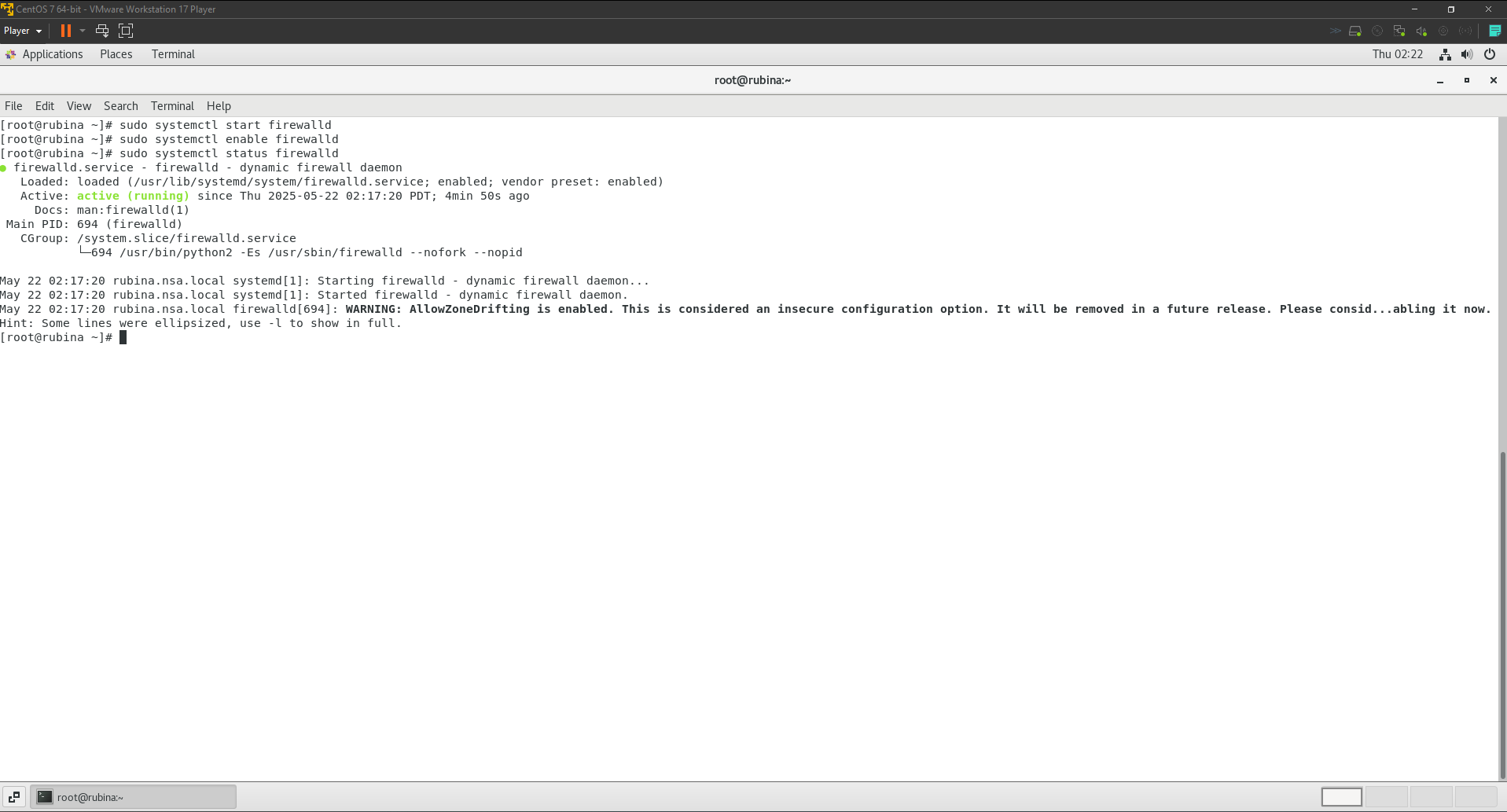
Verifying if Prometheus is working.  


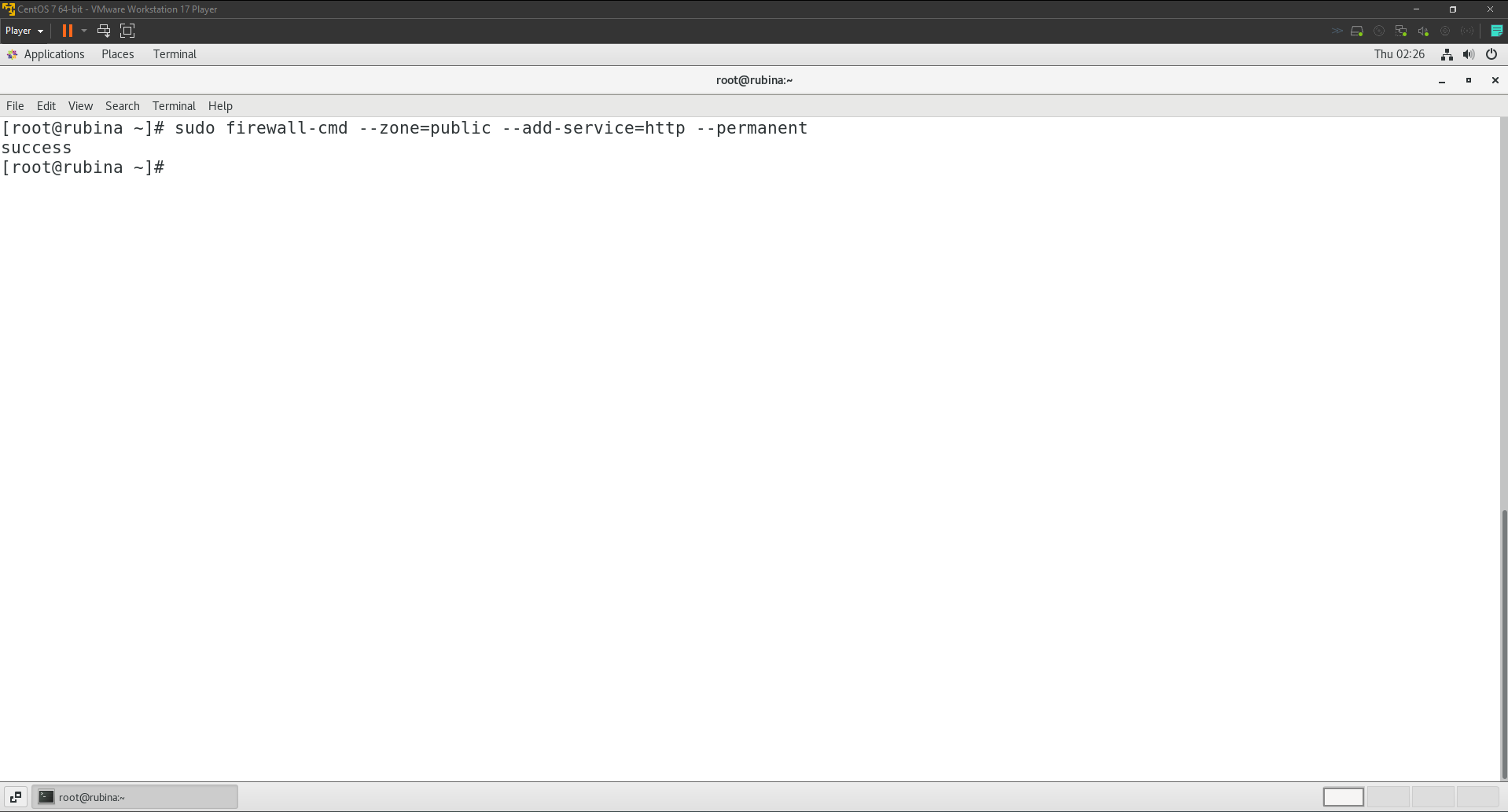


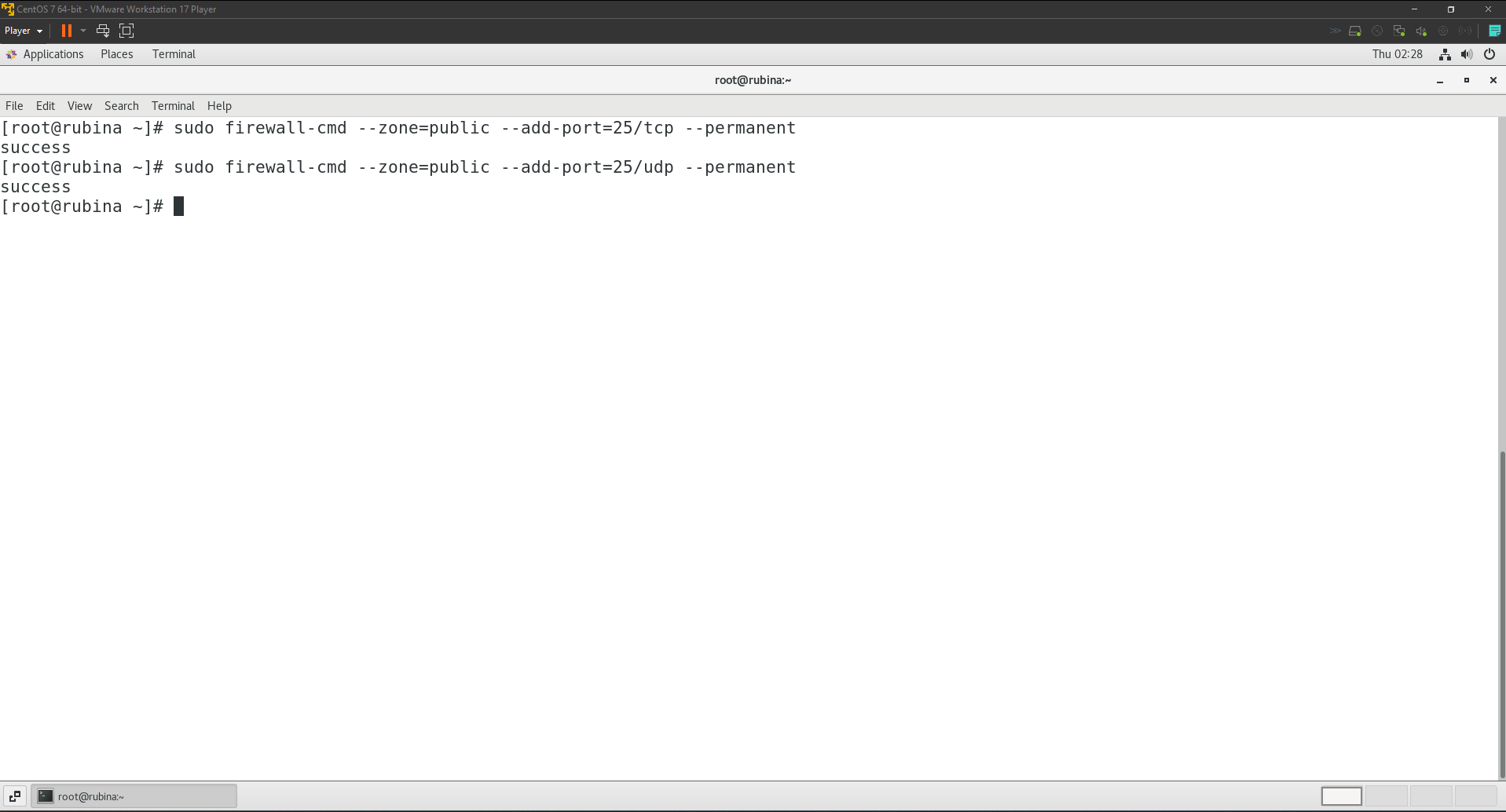
# **Task 13**

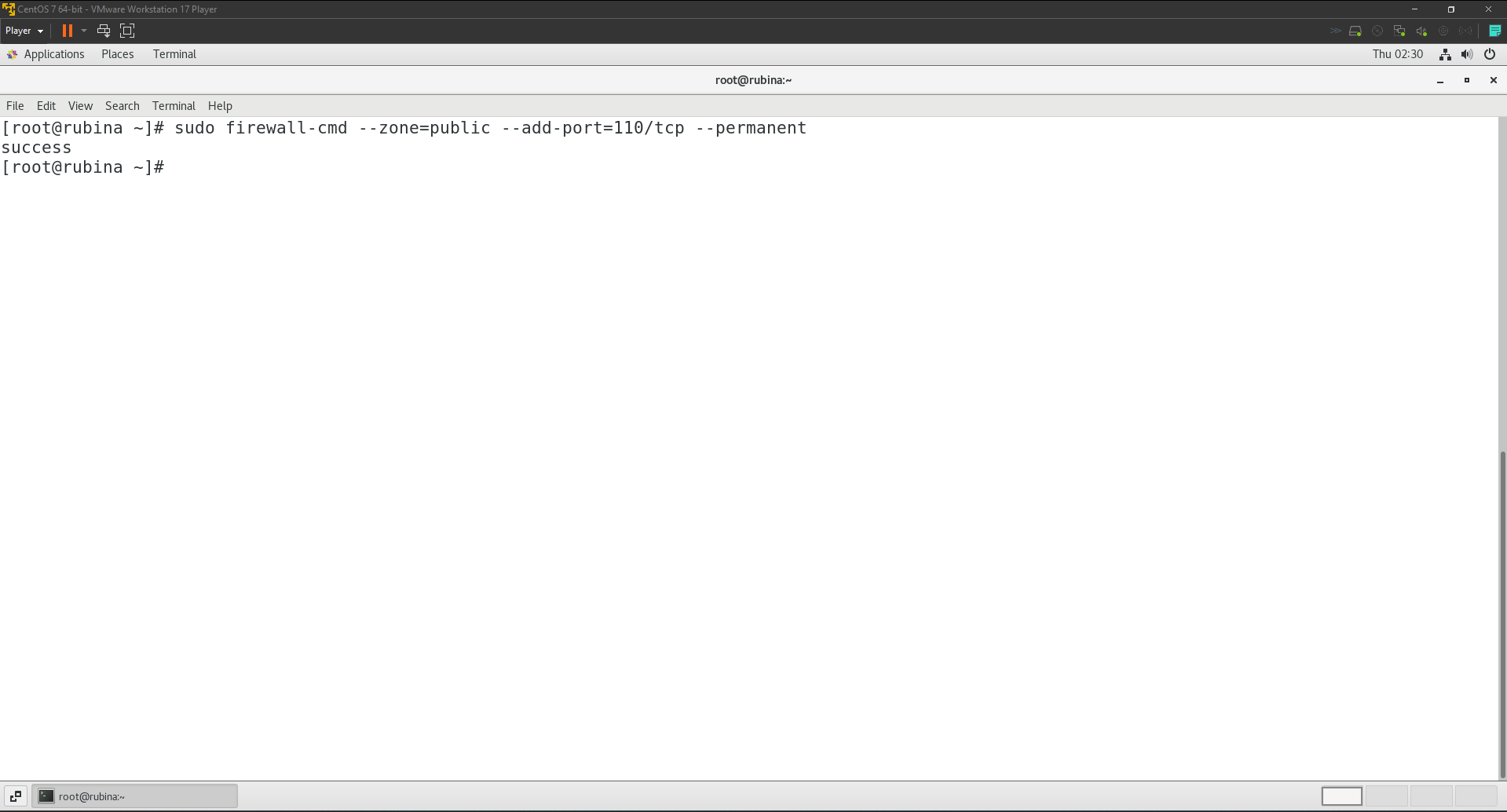
## **Firewall Configuration in CentOS 7:**

1. Install firewalld package as well as start and enable firewall services.
   1. I ran the following command to install firewalld package.   
      
   2. I used the following commands to start, enable firewalld on boot and check the status using the following commands respectively.  
      systemctl start firewalld  
      systemctl enable firewalld

systemctl status firewalld  


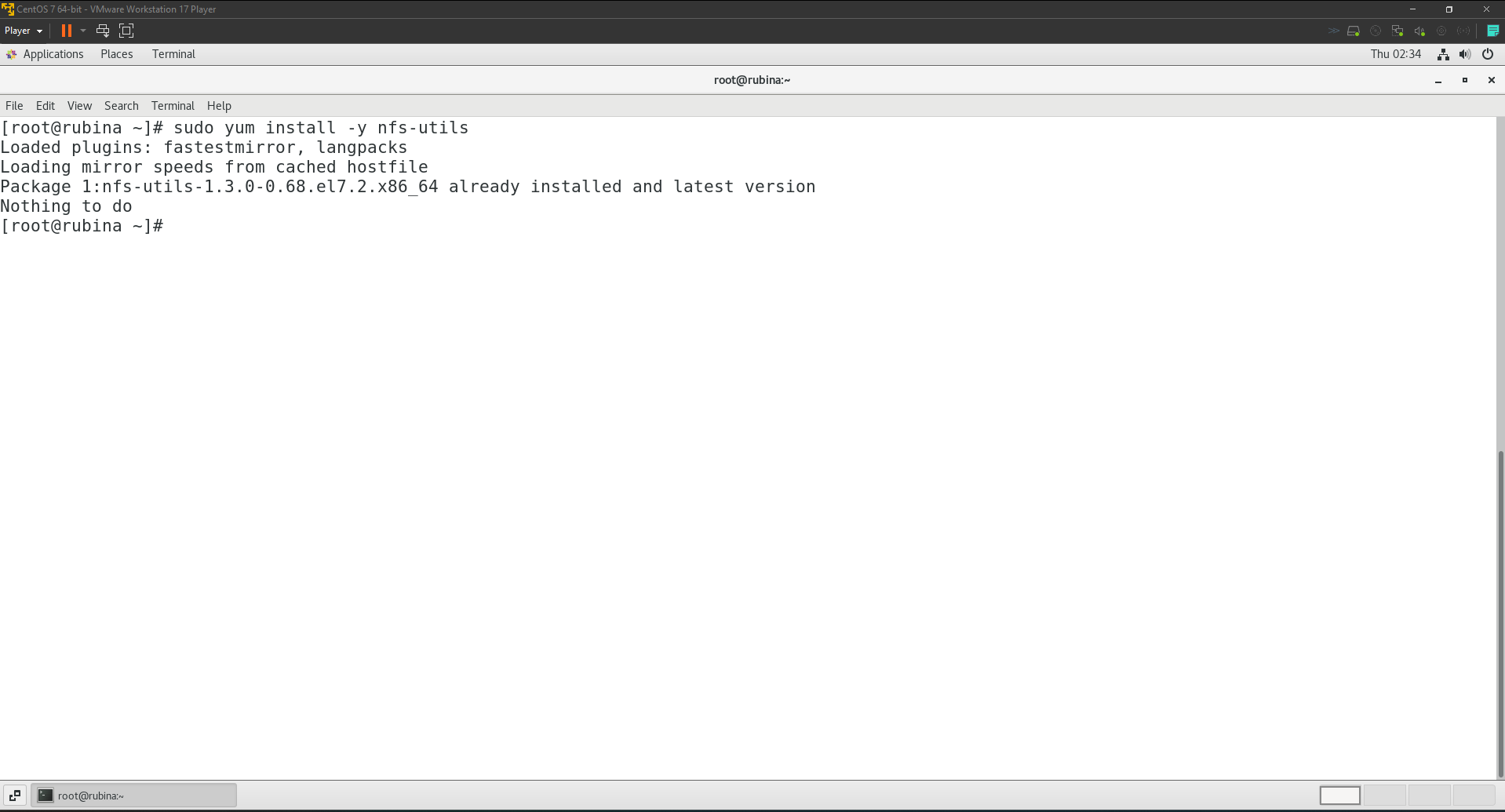
1. Add the following services and ports to allow packets through the firewall. [Service = http, smtp port = 25 /tcp, 25/udp, 110/tcp].
   1. I used the following command to allow HTTP traffic through the firewall.  
      sudo firewall-cmd --zone=public --add-service=http –permanent  
      
   2. I used the following command to allow SMTP traffic on port 25 for both TCP and UDP.  
      sudo firewall-cmd --zone=public --add-port=110/tcp –permanent  
      sudo firewall-cmd --zone=public --add-port=25/udp --permanent



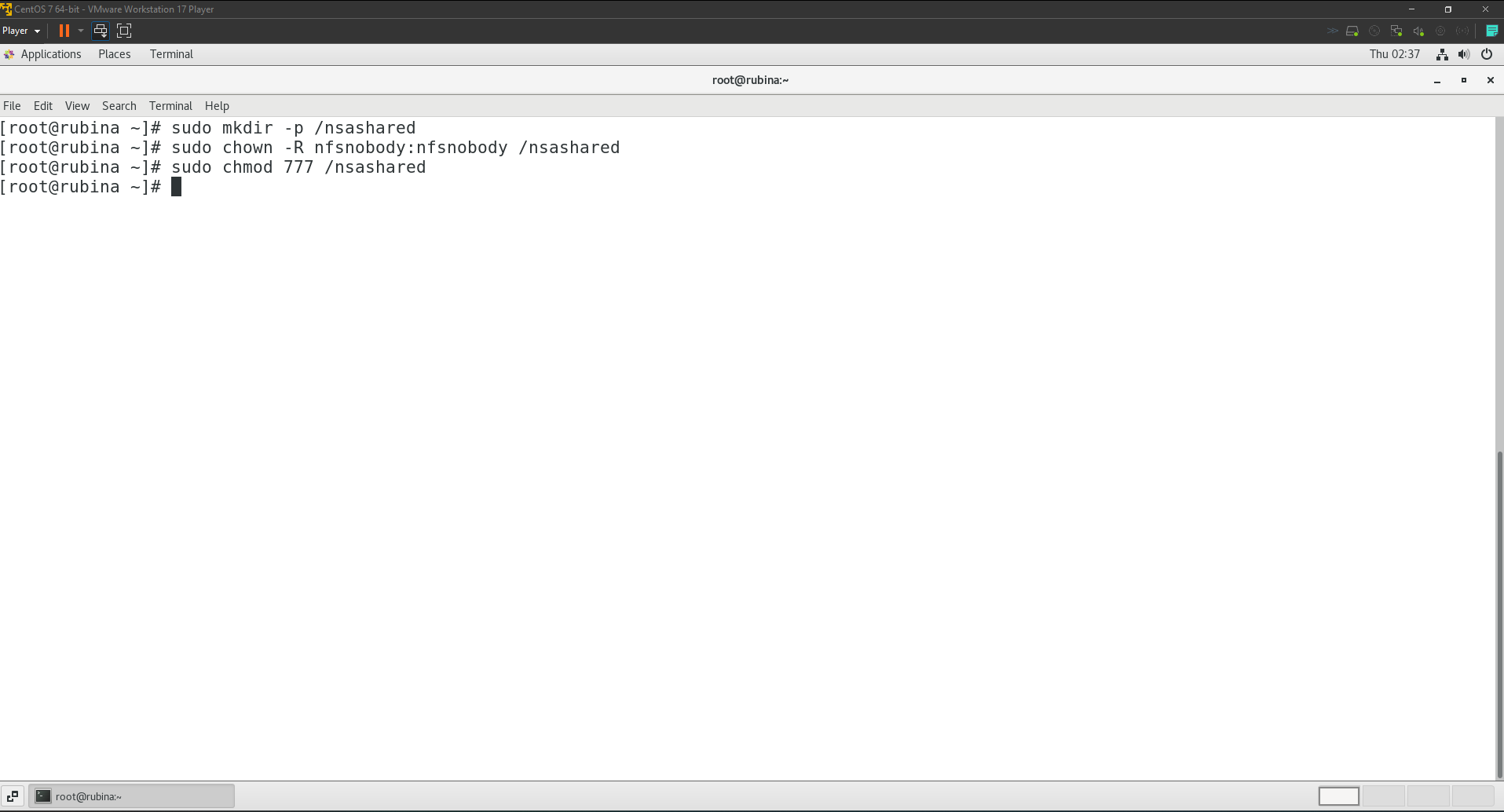
* 1. I used the following command to allow POP3 traffic on port 110/tcp.  
     sudo firewall-cmd --zone=public --add-port=110/tcp –permanent  
     
  2. I reloaded the firewall and verified the changes using the following commands.  
     firewall-cmd –reload  
     firewall-cmd --zone=public --list-all  
     

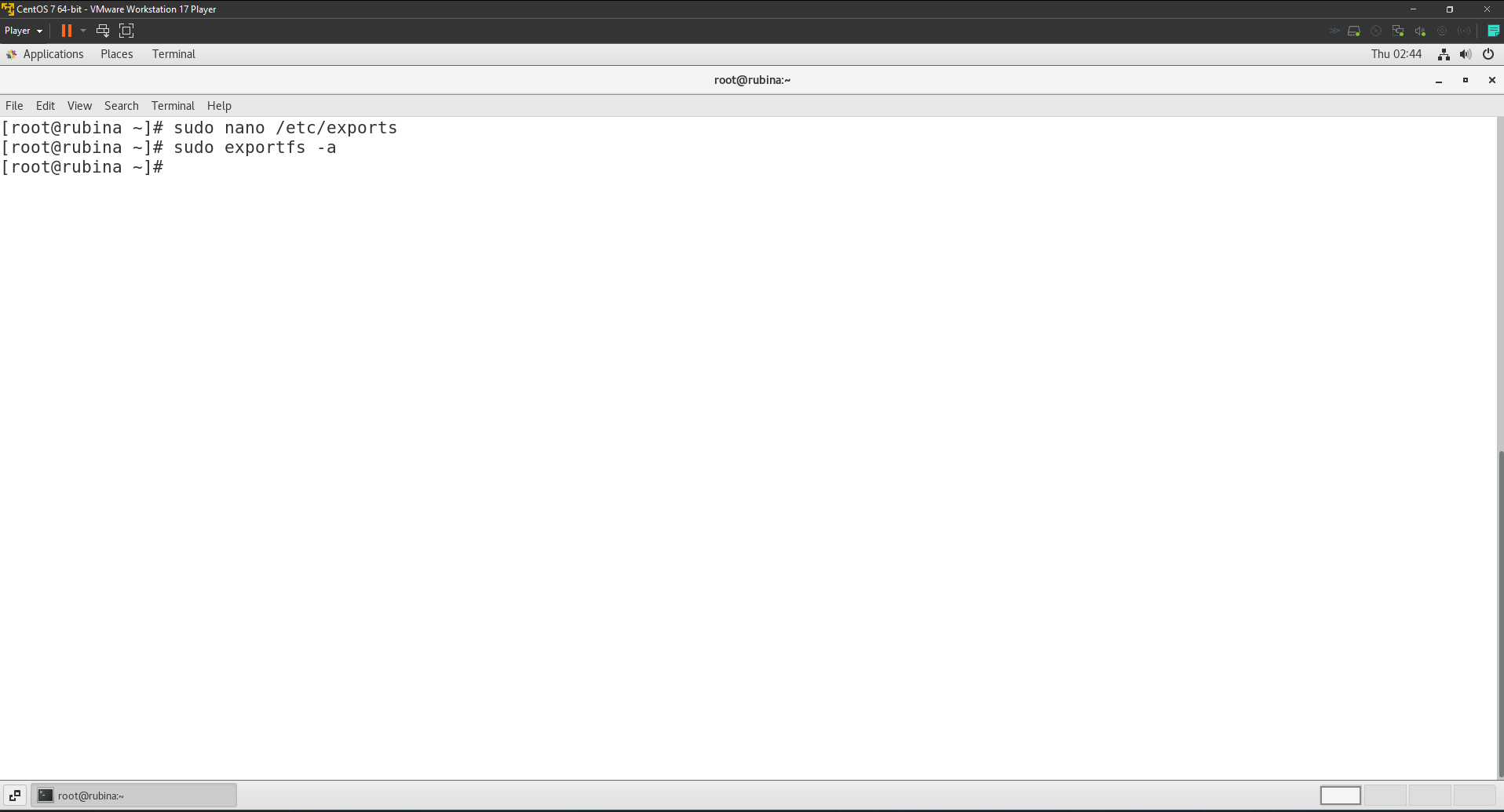
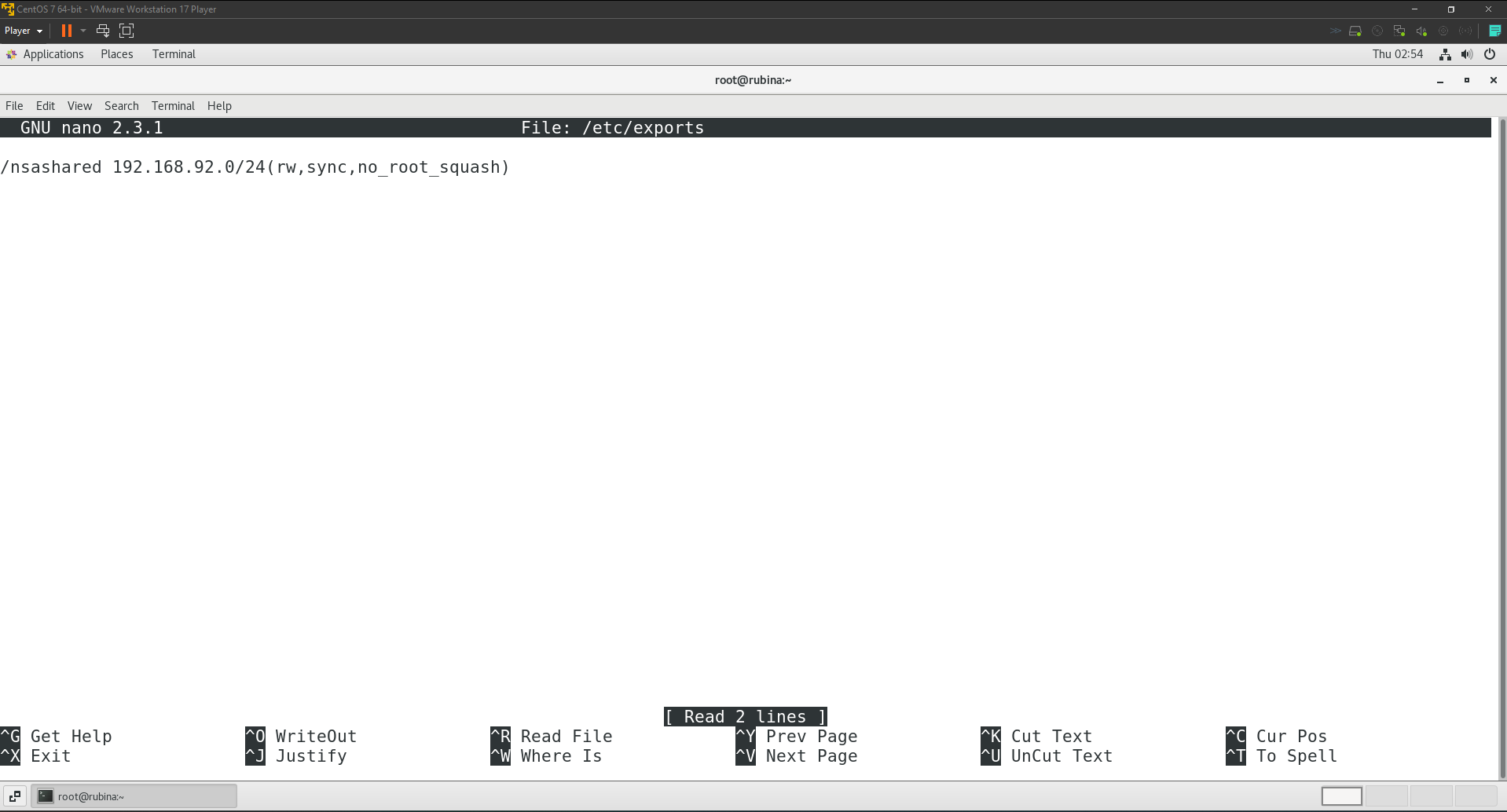
# **Task 14**

## **NFS Server Configuration**:

1. Create & Share /nsashared directory in read/write mode with NFS to users on the local network.  
   To create and share the /nsashared directory using **NFS** (Network File System) on **CentOS 7**, I followed these steps.
   1. First of all, I installed NFS utilities using the following command.  
      sudo yum install -y nfs-utils  
      
   2. After installing the NFS utilities, I created a shared directory ‘nsashared’ and set appropriate permission so users can read/write using the following command.  
      sudo mkdir -p /nsashared  
      sudo chown -R nobody:nogroup /nsashared

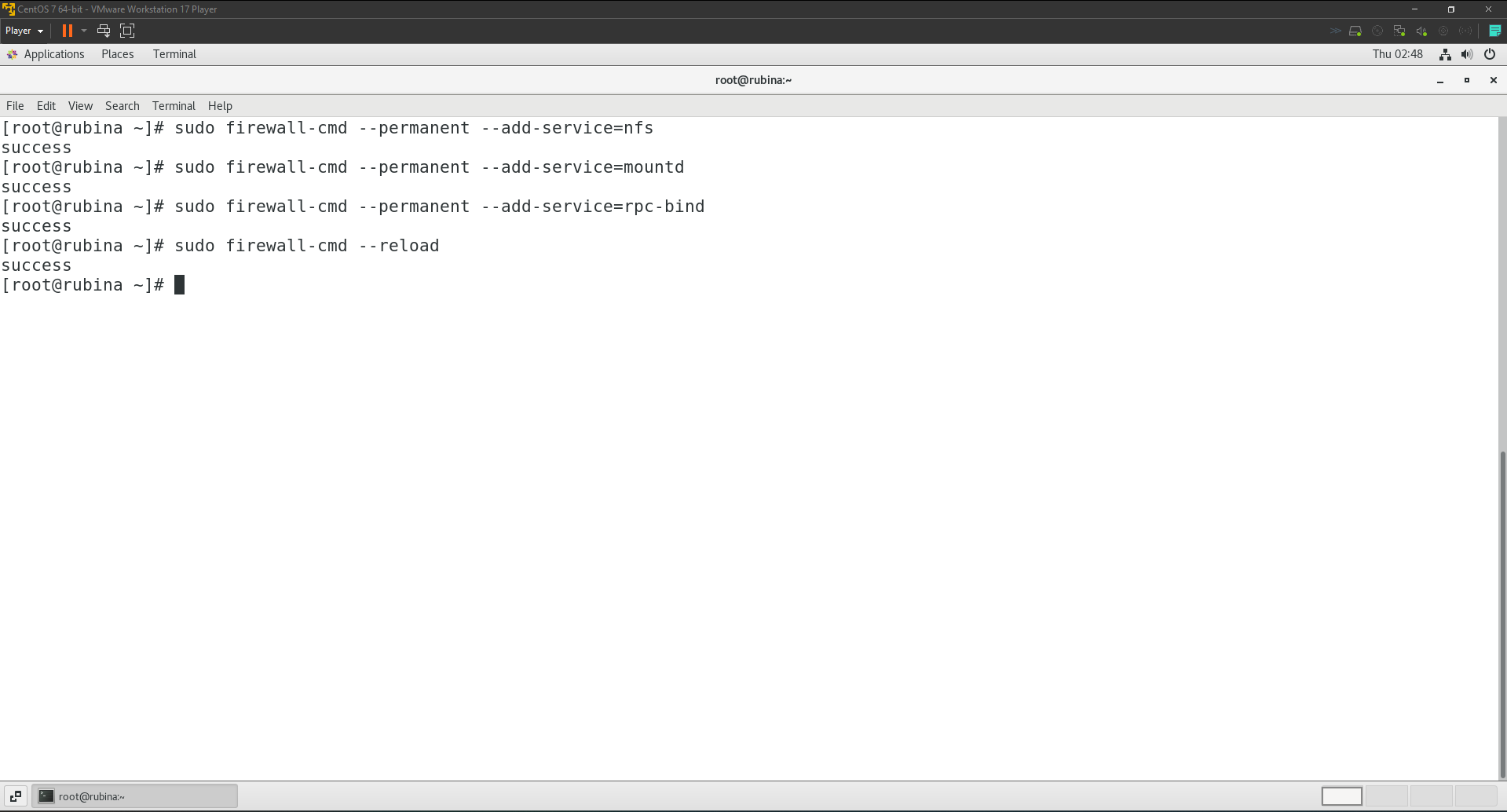
sudo chmod 777 /nsashared



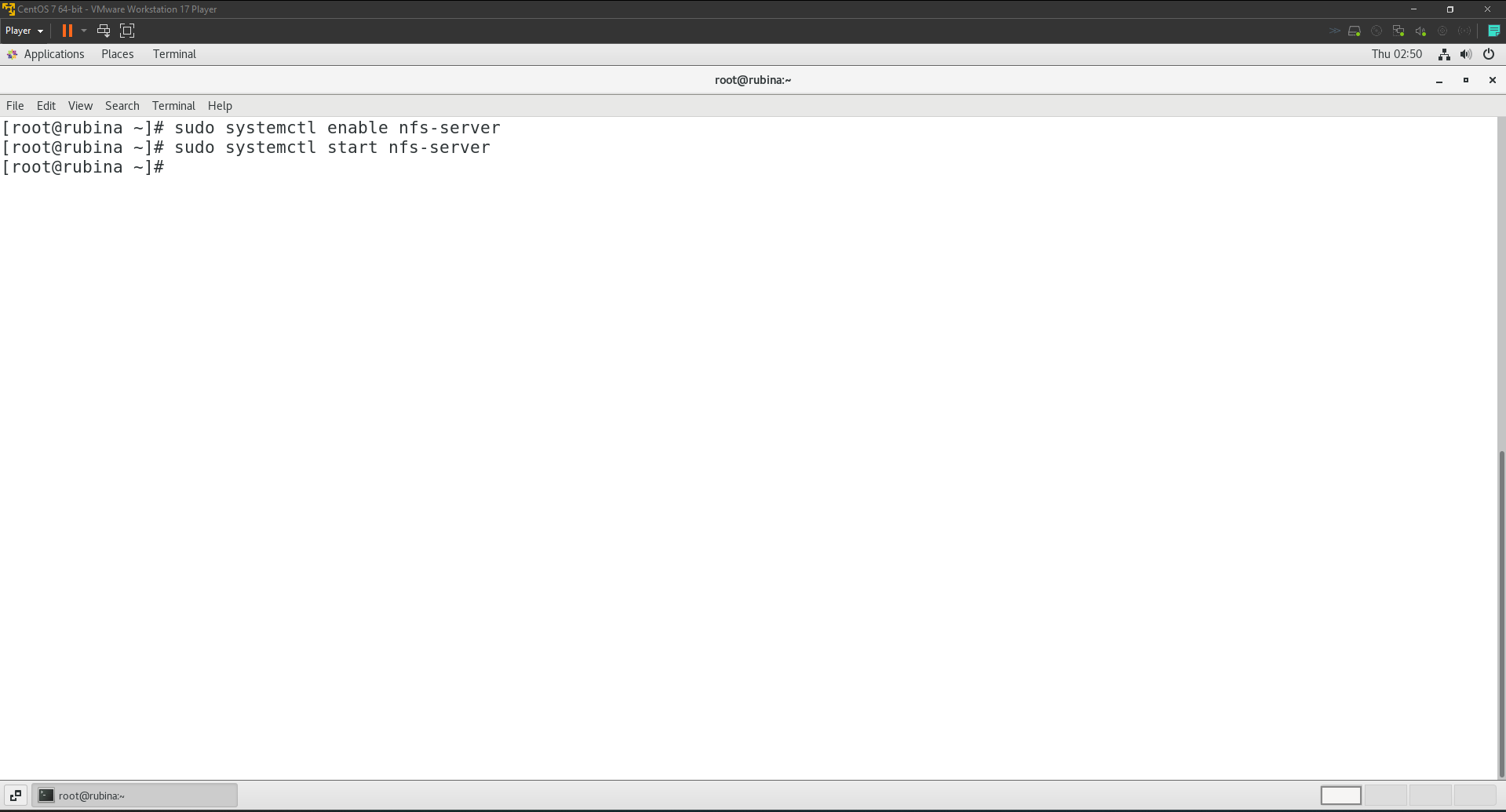
* 1. Then I modified the /etc/exports file to allow read/write to all users on the local network before exporting it using the following command.  
     sudo nano /etc/exports  
     sudo exportfs -a  
       
       
     Content of exports file.  
     /nsashared 192.168.92.0/24 (rw,sync,no\_root\_squash)  
     
  2. Since I also have a firewall running, I allowed NFS traffic by running the following commands.  
     sudo firewall-cmd --permanent --add-service=nfs

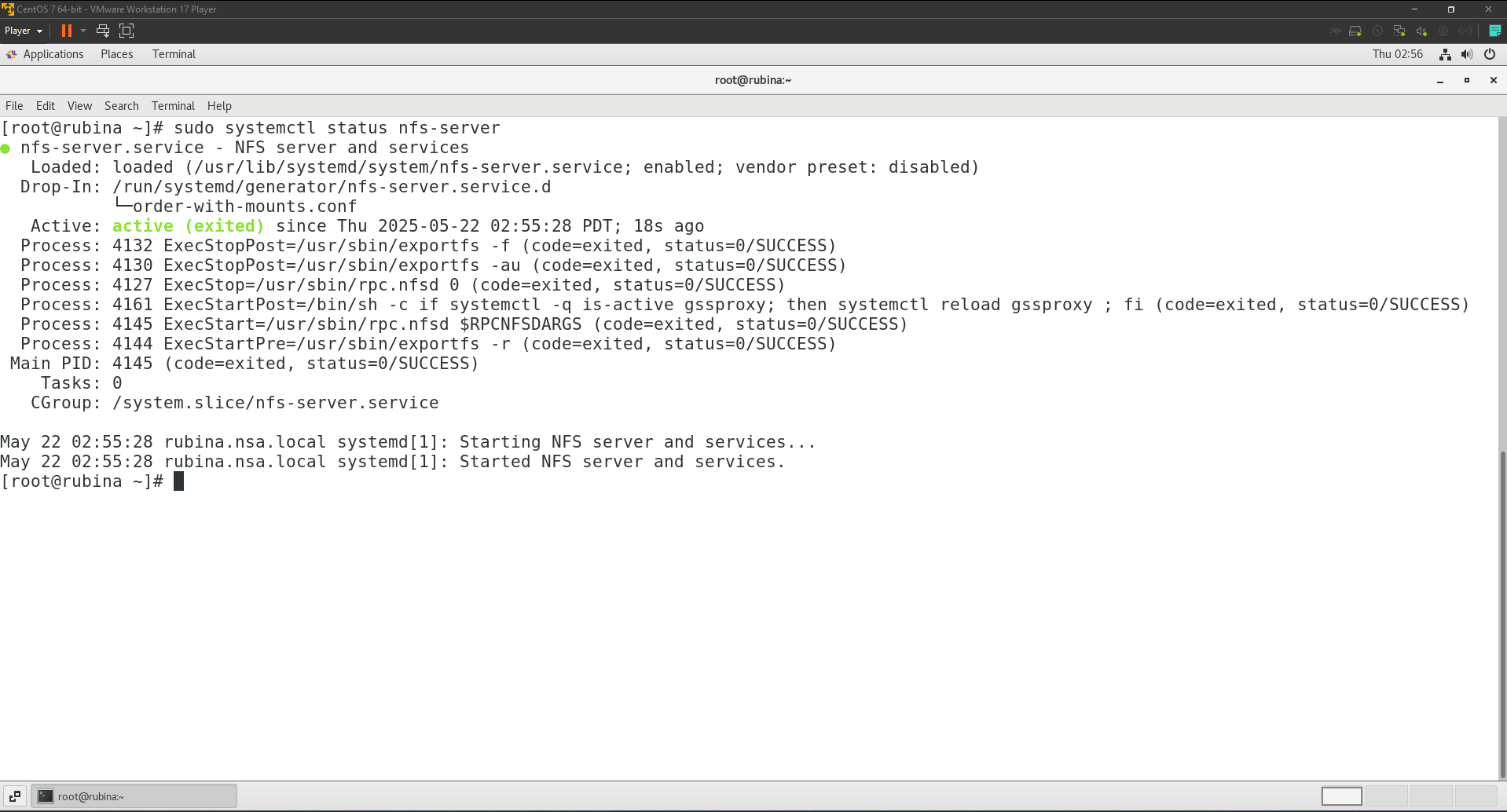
sudo firewall-cmd --permanent --add-service=mountd

sudo firewall-cmd --permanent --add-service=rpc-bind

sudo firewall-cmd –reload  


* 1. Finally, I enabled the NFS server to start on boot and started it using the following commands respectively.  
     sudo systemctl enable nfs-server

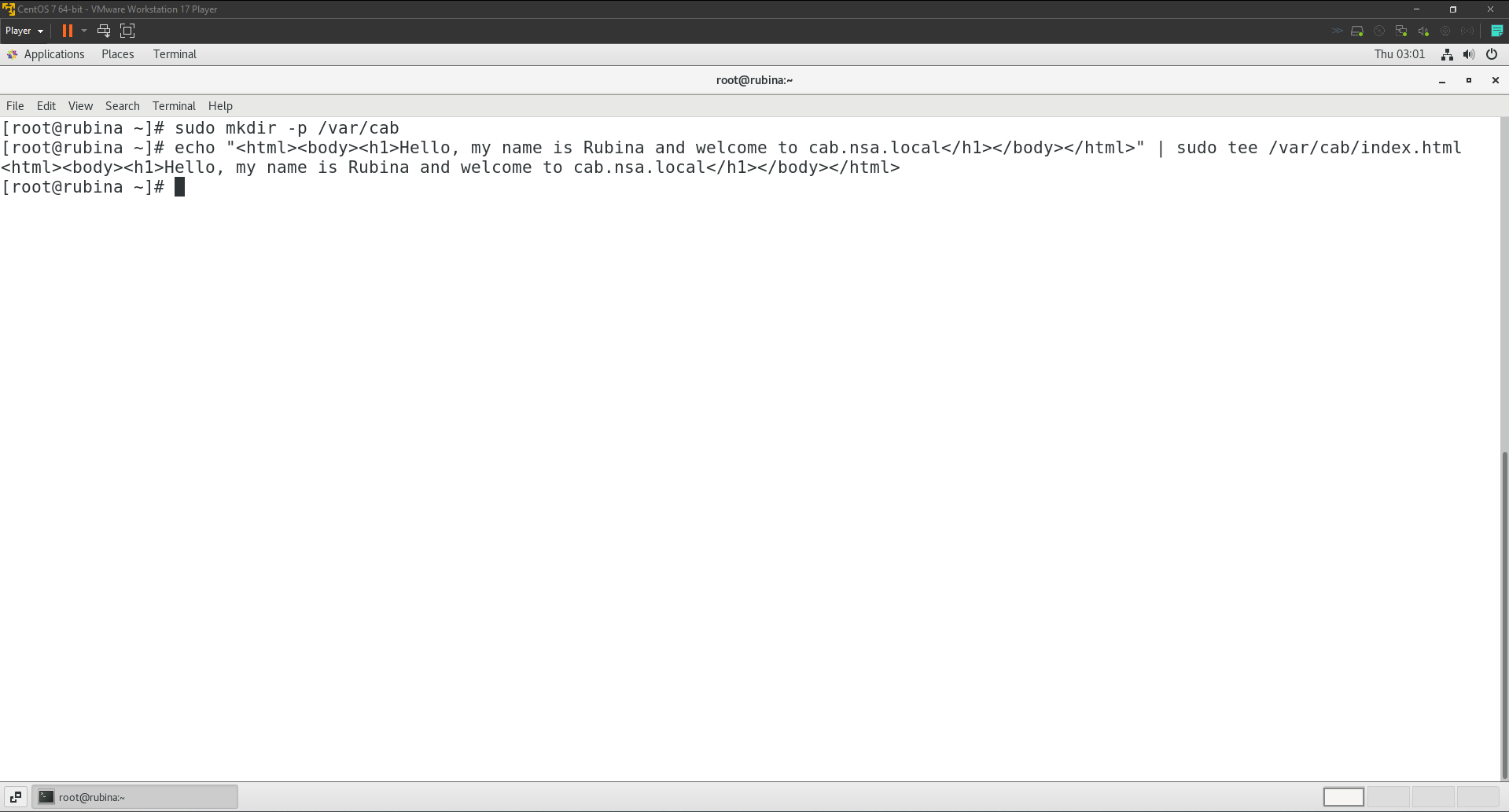
sudo systemctl start nfs-server  
  
Verifying if NFS server is working correctly.

sudo systemctl status nfs-server  


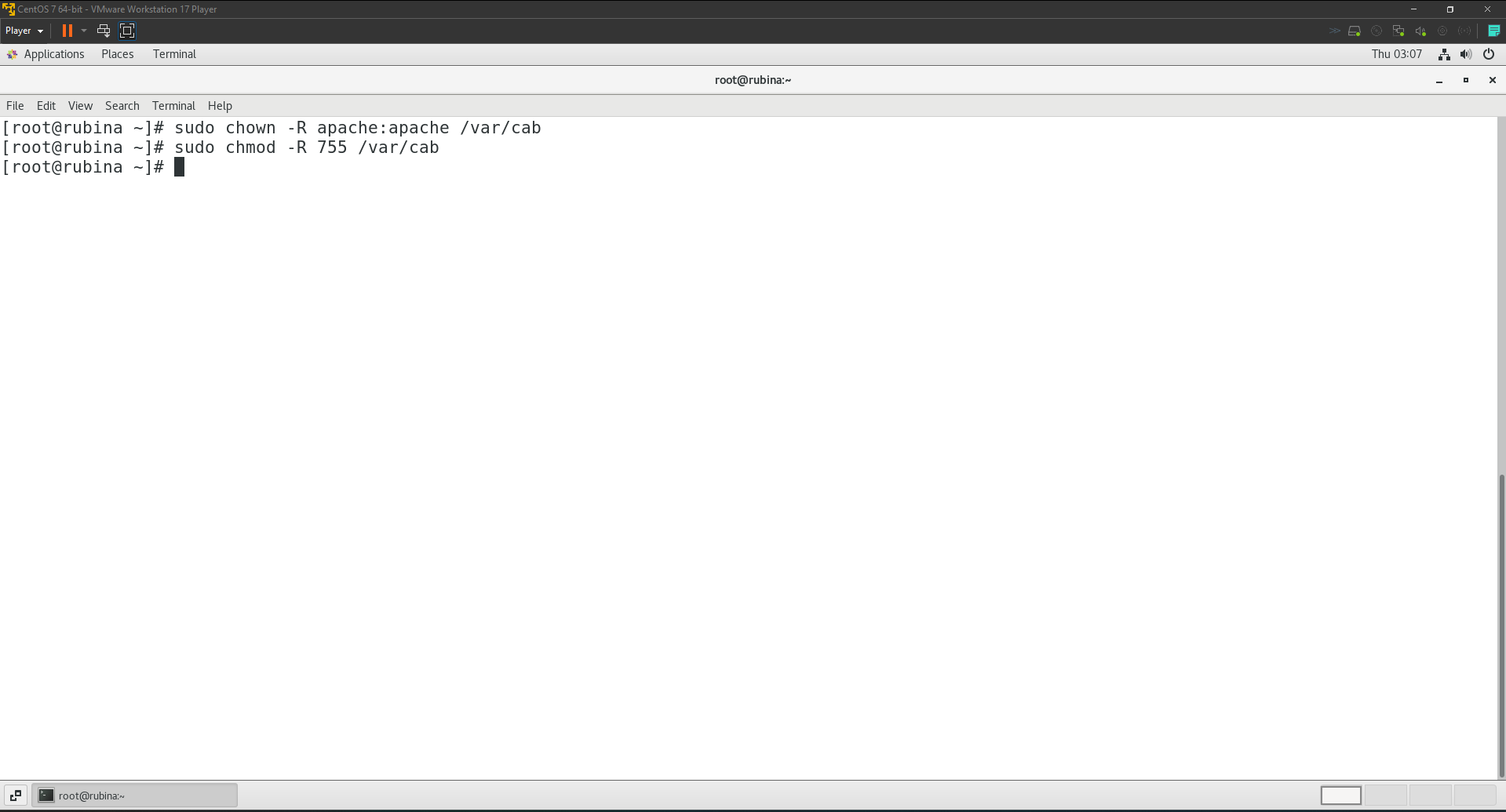
# **Task 15**

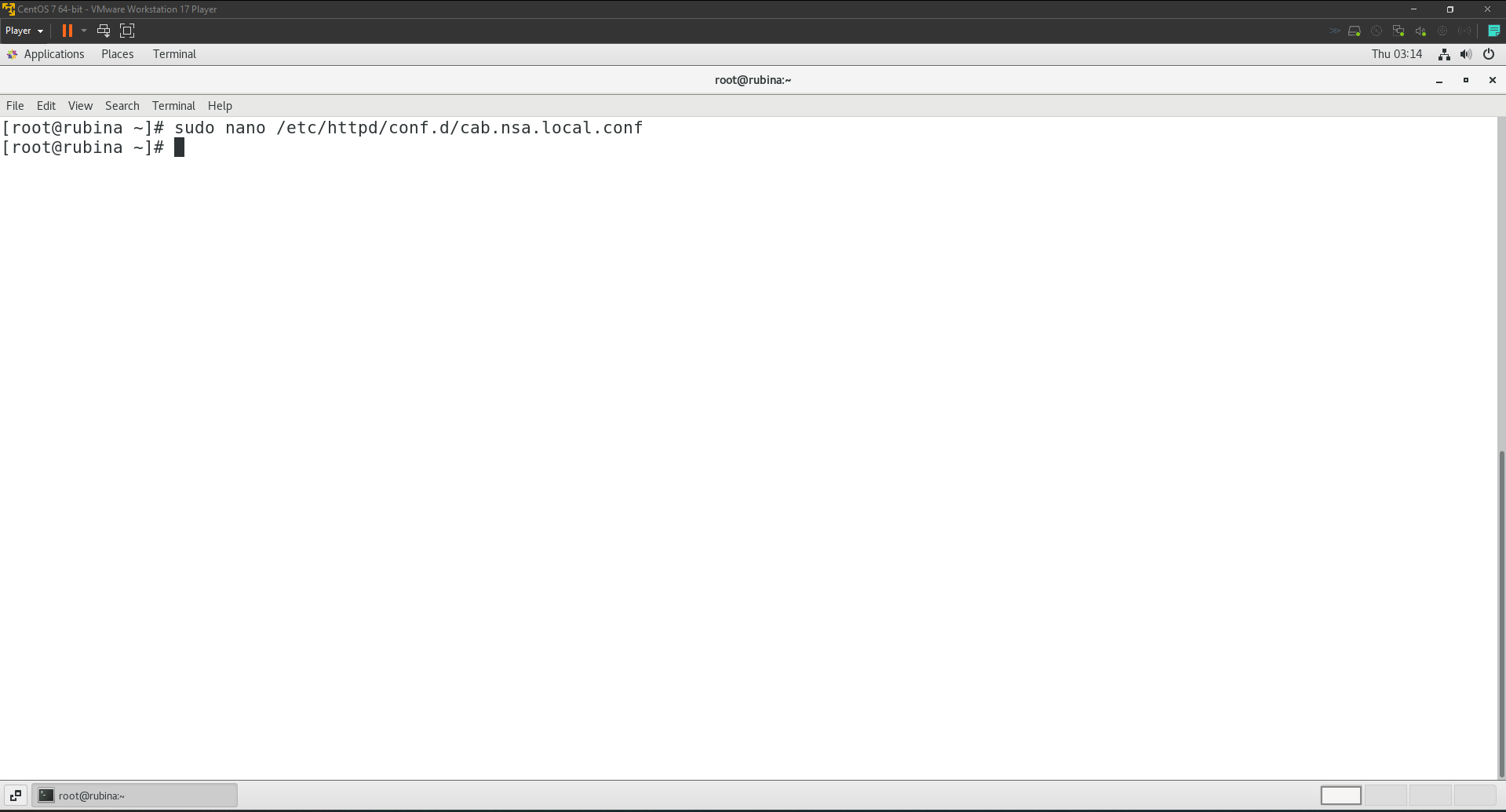
## **Apache Web Server Configuration:**

1. Set up a simple virtual host called cab.nsa.local with index.html placed in DocumentRoot /var/cab. Update the hosts table for name resolution.
   1. I started with installing apache using the following command.  
      sudo yum install httpd -y  
      
   2. I created the directory in DocumentRoot and created a simple index.html file using the following commands respectively.

sudo mkdir -p /var/cab  
  
echo "<html><body><h1>Hello, my name is Rubina and welcome to cab.nsa.local</h1></body></html>" | sudo tee /var/cab/index.html  


* 1. I set up correct ownership and permissions for Apache to correct ownership and permissions for Apache access the folder using the following command.  
     sudo chown -R apache:apache /var/cab

sudo chmod -R 755 /var/cab  


* 1. Then I created a virtual host config for cab.nsa.local with /var/cab as the root using the following command.  
     sudo nano /etc/httpd/conf.d/cab.nsa.local.conf  
       
     Content of the virtual host config file.  
     <VirtualHost \*:80>

ServerAdmin [webmaster@cab.nsa.local](mailto:webmaster@cab.nsa.local)

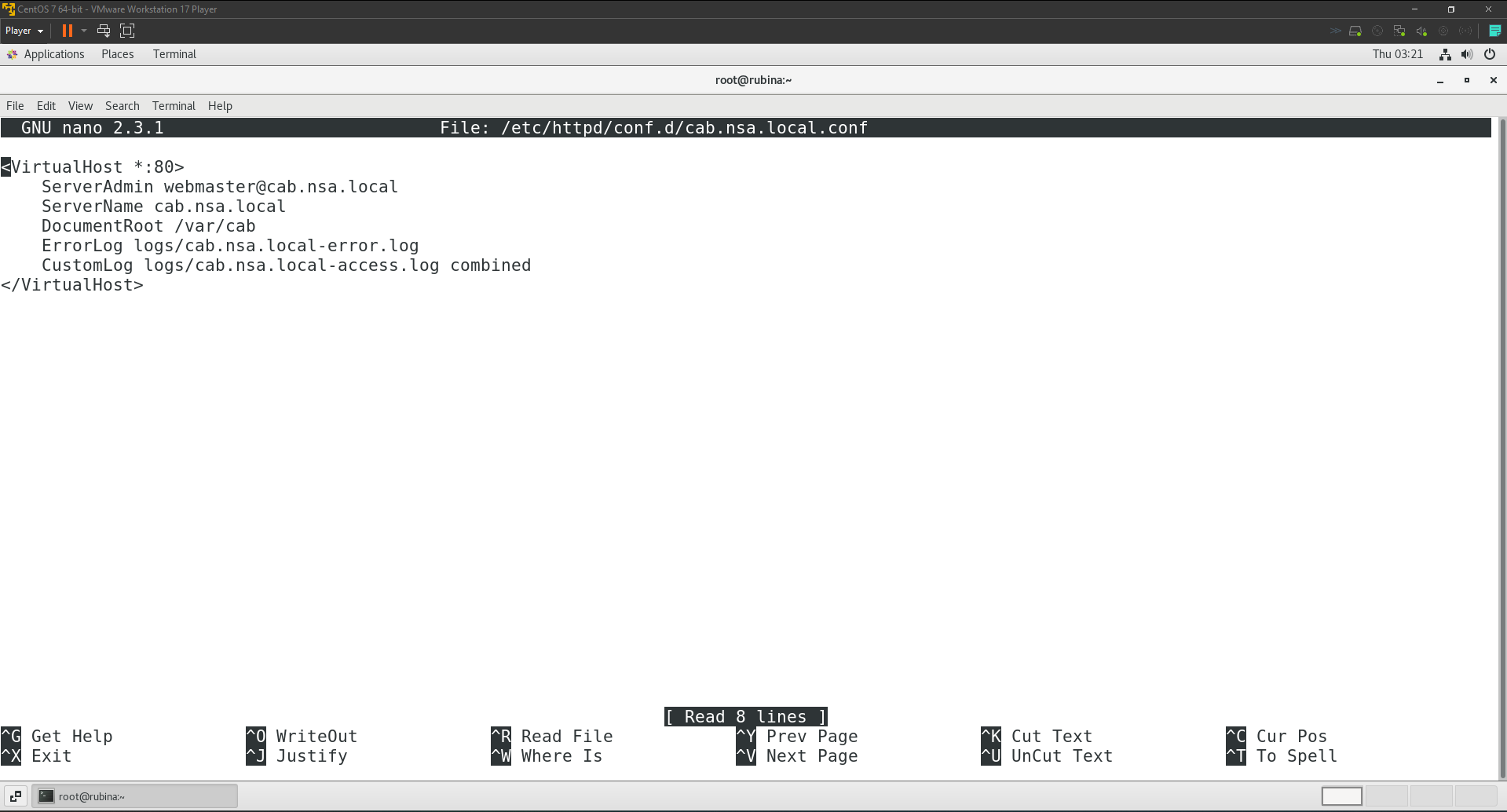
ServerName cab.nsa.local

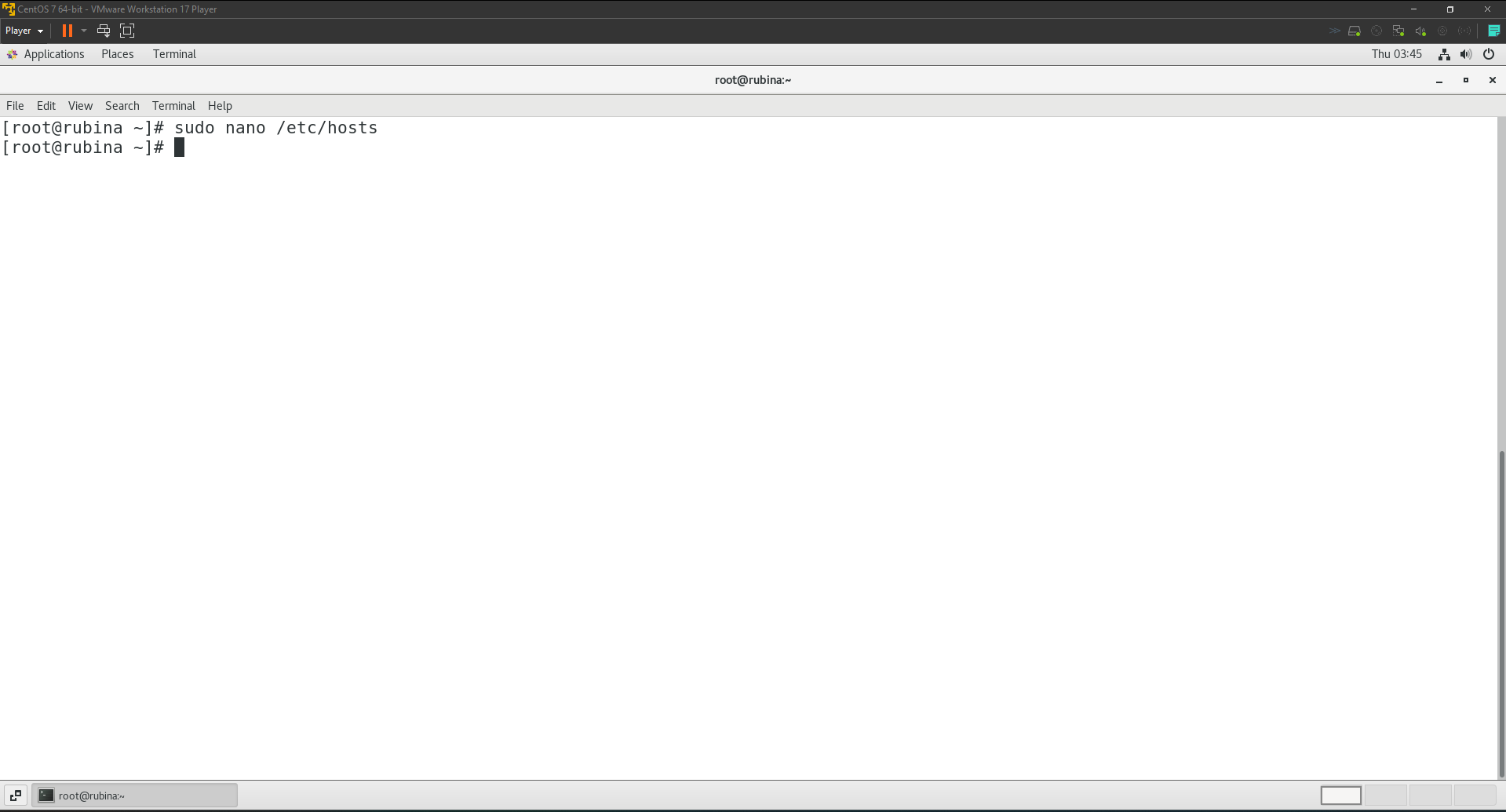
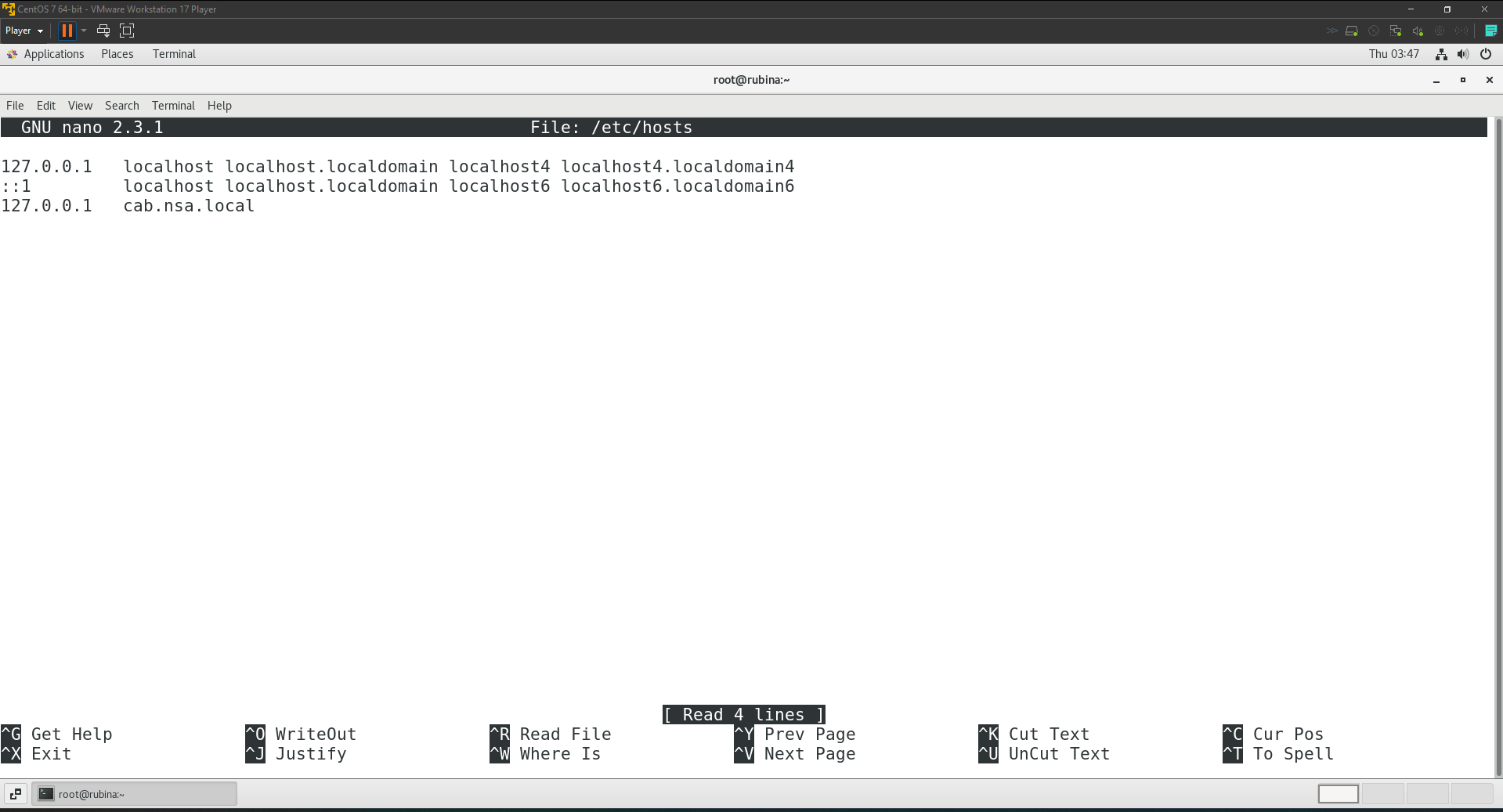
DocumentRoot /var/cab

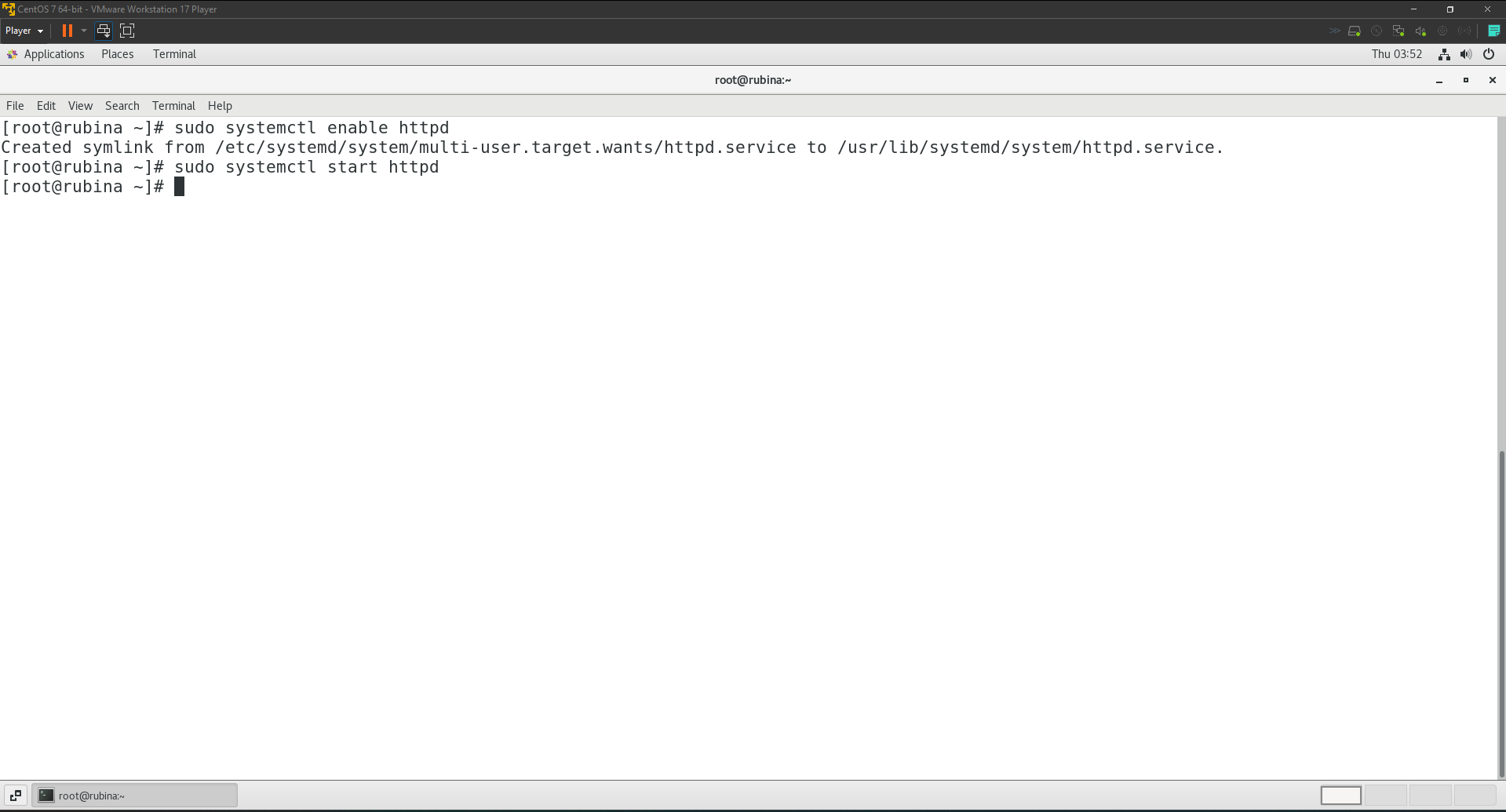
ErrorLog logs/cab.nsa.local-error.log

CustomLog logs/cab.nsa.local-access.log combined

</VirtualHost>



* 1. I updated /etc/hosts file for local name resolution using the following command.  
     sudo nano /etc/hosts  
       
       
     Content of the hosts file.  
     127.0.0.1 cab.nsa.local  
     
  2. I used the following commands to enable Apache to start on boot and start the service respectively.  
     sudo systemctl enable httpd

sudo systemctl start httpd  
  
  
Testing the site to see if it is not working or not.  
