**Task#1**

**How to boot a USB for Windows 10 using CMD?**

**Step 1**

Connect a USB drive to a laptop or a computer.

**Step 2**

Download the Windows 10 ISO file.

**Step 3**

Open a command prompt (run as administrator)

Follow these commands:

1. diskpart
2. list disk
3. select disk (which drives you to want to boot)
4. clean
5. create partition primary
6. format fs=fat32 quick
7. active
8. assign (The name of the USB drive)
9. exit
10. The last step is the copy/paste the required packages into the disk

Your USB is successfully booted and you can plug in every system and run booted files.

**Task #2**

**Create a virtual box on a Host Machine**

1. Install virtual box.
2. Install Ubuntu 22.04.2 desktop LTS version.
3. Then installed the virtual box and open the virtual box and adjust the storage and RAM.
4. Go to a storage setting and pick up an Ubuntu ISO file and ok.
5. Open a virtual machine on a Ubuntu Linux OS
6. Go to the setting Network and change the NAT to bridge Adapter then press ok.

**Task #3**

**Install apache2, PHP, and MySQL servers on Ubuntu OS**

1. Open the cmd terminal on the Ubuntu server

Fallows the following commands to install the apache2, PHP, and MySQL servers on Ubuntu OS.

* apt update
* apt install apache2
* apt install php
* apt install mysql-server

Extra commands are used

* how to get the serial number of any computer
* wmic bios get serialnumber
* systeminfo
* find / -name “name of package ” find the package
* cat /etc/os-release find the operating system of the virtual machine
* ps –aux |grep (find the error)
* kill -9 ( kill errors)
* systemctl apache2 status ( check the status the apache2 is running or not )
* netstat –tulpn check the ports are up
* ping [ip address] connect to another computer through cli
* apt update
* apt upgrade -y
* apt install nginx install nginx
* systemctl status nginx check status to nginx running
* clear
* apt install htop
* apt install iftop
* apt install mload
* apt install nload
* 10 htop
* 11 atop
* 12 apt install atop
* 13 atop
* 14 iftop
* 15 top
* 16 nload
* 17 clear
* 18 apt install inkscpae
* 19 apt install inkscapae
* 20 apt install inkscape
* 21 sudo apt update
* 22 apt list --upgradable
* 23 apt upgrade
* 24 clear
* 25 ls
* 26 ls -al
* 27 touch bz.txt make an empty file
* 28 ls
* 30 ls --help
* 35 ls -l
* 38 pwd
* 39 cd ..
* 40 ls –a
* 46 cd etc
* 48 cd nginx
* 51 cd nginx
* 52 cd ../../
* 53 cd home
* 54 lsF
* 55 cd afaq
* 56 ls
* 58 cd Desktop
* 60 cat bz.txt
* 61 vi bz.txt
* 63 cd Desktop
* 65 cat bz.txt
* 66 vi bz.txt
* 67 cd Desktop
* 68 touch afaq.txt
* 69 cat afaq.txt
* 70 vi afaq.txt
* 71 cat afaq.txt
* 73 cat afaq.txt > hs.txt
* 74 ls
* 75 ls -l
* 76 rmtouch bz.txt
* 77 cat afaq.txt
* 78 cat bz
* 79 cat bz.txt
* 80 cat afaq.txt hs.txt > aa.txt
* 82 cat aa.txt
* 83 cd etc
* 84 cd /etc
* 85 cat passwd
* 86 ls -al
* 87 cd ..
* 88 ls -al
* 89 clear
* 90 exit
* 91 cd ..
* 92 apt install php php-mysql apache2 mysql-server
* 93 cd /war/www/html
* 94 cd /var/www/html
* 95 wget http:/wordpress.org/latest.zip
* 96 wget https:/wordpress.org/latest.zip
* 97 wget https://wordpress.org/latest.zip
* 104 cat docker.file
* 105 vi docker.file
* 106 cat docker.file
* 107 mkdir htmlapplication
* 109 cd htmlapplication
* 110 touch html.txt
* 112 vi html.txt
* 113 cat html.txt
* 116 mv docker.file htmlapplication
* 118 cd htmlapplication
* 120 docker bulid -t htmlapp:0.1 .
* 121 apt install docker
* 122 apt install docker.io
* 123 apt install podman-docker
* 124 docker bulid -t htmlapp:0.1 .
* 125 apt update
* 126 apt upgrade
* 127 apt install apache2
* 128 apt install php
* 129 apt install mysqual
* 130 apt install mysql
* 131 dpkg -l
* 132 apt update
* 135 apt install mysql-server
* 136 mysql -u root -p
* 137 mysqldump -u root -p testdb >testdb.sql
* 138 mysql -u root -p
* 139 mysqldump -u root -p testdb < testdb.sql
* 140 mysql -u root -p
* 141 ysql -u root -p testdb < testdb.sql
* 142 mysql -u root -p testdb < testdb.sql
* 143 ip a
* 144 apt install netstat tools
* 145 ip a
* 146 ip a /all
* 147 apt install ssh
* 148 netstat -tulpn
* 149 apt install net-tools
* 150 reboot
* 152 cd Desktop
  + 1 154 cd ..
* 155 ls
* 156 cd afaq
* 157 ls
* 158 cd ..
* 159 afaqb
* 160 cd afaqb
* 161 ls
* 162 ip a
* 163 cd ..
* 164 ls
* 165 cd afaqb
* 166 cd Desktop
* 167 ls
* 168 vi afaq3
* 169 nano afaq4
* 170 ls
* 171 cat afaq4
* 172 nano afaq4
* 173 ip a
* 174 user
* 175 echo user
* 176 echo $USER
* 177 vi /etc/hostname
* 178 cat afaq
* 179 cat afaq.txt
* 180 cat > afaq
* 181 touch afaq1 afaq2
* 182 nano afaq2
* 183 vi afaq2
* 184 nano afaq1
* 185 cat afaq1 afaq2
* 186 nano afaq1
* 187 nano afaq2
* 188 ls
* 189 cd afaq1
* 190 cat > afaq3 afaq1 afaq2
* 191 ls
* 192 cat >> file3
* 193 nano afaq3
* 194 cat afaq afaq1 afaq2 afaq3 >afaq3
* 195 cat afaq afaq1 afaq2 >afaq3
* 196 nano afaq3
* 197 stat afaq1
* 198 stat afaq2
* 199 stat afaq3
* 200 ls
* 201 cat afaq afaq1 afaq2 afaq3 file3
* 202 cat afaq afaq1 afaq2 afaq3 >file3
* 203 nano file3
* 204 touch afaq afaq1 afaq2 afaq3 file3
* 205 stat afaq3
* 206 touch -a afaq1
* 207 vi file3
* 208 ip a
* 209 restart
* 210 reboot
* 211 ls -i
* 212 ls -l
* 213  history
* apt list --installed

**Task #4**

* **Create a MySQL database using cmd**
* Open the Window power shell and connect the Ubuntu server through the ping command
* mysql –u root –p (connect the server of the MySQL database)
* show databases;
* create database [database-name ];
* use database;
* **Backup database**
* Mysqldump –u root –p database name > name of new database.sql
* **Again restore the database**
* Mysql –u root –p database name < database.sql

If the MySQL is not working (access denied for user 'root'@'localhost')

* dpkg-reconfigure mysql-server

**Task#5**

**How to connect a host machine to a virtual machine?**

* To use the ping command we connect a virtual machine
* ping 192.168.200.180
* ssh hostname@ip address
* ssh [afaq@192.168.200.180](mailto:afaq@192.168.200.180)

**task#06**

study IP and MIC Binding

**task#07**

What is the difference between VM and containers?

In summary, virtual machines provide full OS virtualization with dedicated resources, while containers offer lightweight application-level virtualization with shared resources and faster startup times. Both have their use cases depending on factors such as performance requirements, resource utilization, portability, and isolation needs.

Create 4 virtual machines in the server to centos 7 cts with 1cpu, 2GB,8GB hard disk

MariaDB 192.168.200.23/24

MongoDB 192.168.200.24/24

Neo4j 192.168.200.25/24

Minio 192.168.200.26/24

Task #08

How to check an ip trace route

* tracers facebook.com
* tracert google .com

**Task #09**

=====configure a free letsencrypt ssl certificate with apache2 on ubuntu =====  
1- Install the Certbot package:  
sudo apt-get update  
sudo apt-get install certbot python3-certbot-apache

2- Make sure that your domain is pointing to the server where you are installing the SSL certificate.  
3- sudo a2enmod ssl  
4- sudo systemctl restart apache2  
5- Run the Certbot command to obtain the SSL certificate:  
sudo certbot --apache -d example.com       (Replace example.com with your domain name)

6- Follow the instructions provided by Certbot to obtain the SSL certificate.  
7- Certbot will ask if you want to redirect HTTP traffic to HTTPS. Answer yes if you want to do so.  
8- Verify that the SSL certificate has been installed correctly by visiting your website using HTTPS. You should see a lock icon in your browser indicating that the connection is secure.  
9- Certbot will automatically renew your SSL certificate before it expires. You can verify this by running the following command:  
sudo certbot renew --dry-run

10- Done

**Connect the aws pem file to the Window power shell**

* ssh –I [pem file path] ec2-user@[ip of the instance ]

**Task#10**

How to create a DNS server and configure it on the Ubuntu server

To create an AWS Amplify account, you'll need an existing Amazon Web Services (AWS) account. If you don't have one, you can sign up for an AWS account first. Once you have your AWS account ready, follow these steps to create an AWS Amplify account:

1. Open the AWS Management Console: Go to the AWS Management Console by visiting the AWS website at <https://aws.amazon.com/console/>.
2. Sign in to your AWS account: Use your AWS account credentials (email address and password) to sign in.
3. Navigate to the AWS Amplify service: Once you're signed in, you can either search for "Amplify" in the AWS Management Console search bar or find it under the "Developer Tools" section.
4. Access the AWS Amplify console: Click on the "Amplify" service to access the AWS Amplify console.
5. Set up the Amplify service: You'll see a welcome screen with options to get started in the Amplify console. Click on the "Get Started" button.
6. Select a Git provider: Amplify supports various Git providers like AWS CodeCommit, GitHub, and BitbuScket. Choose the provider you prefer and follow the instructions to connect your Git repository to Amplify.
7. Configure your Amplify app: Once your Git repository is connected, you'll need to configure your Amplify app. Provide the necessary details such as the name of your app, the framework you're using, and other project-specific information.
8. Review and create the app: Double-check the configuration settings you've entered and click on the "Create app" button to create your Amplify app.