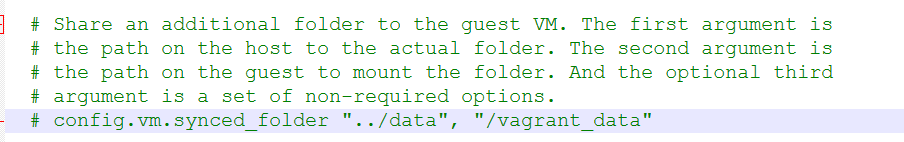
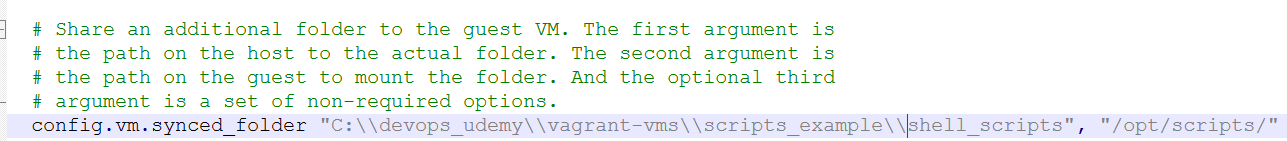
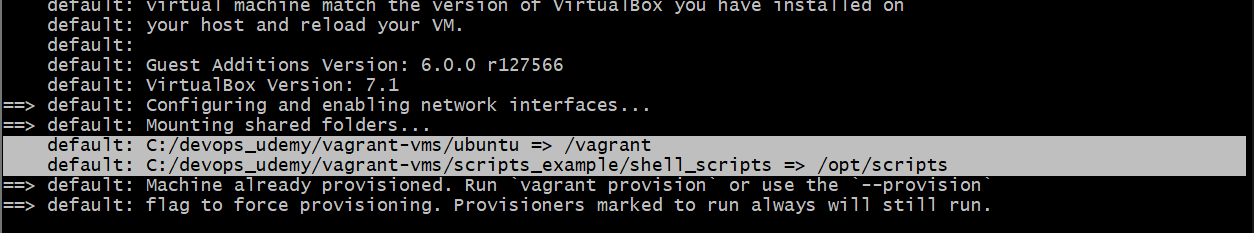
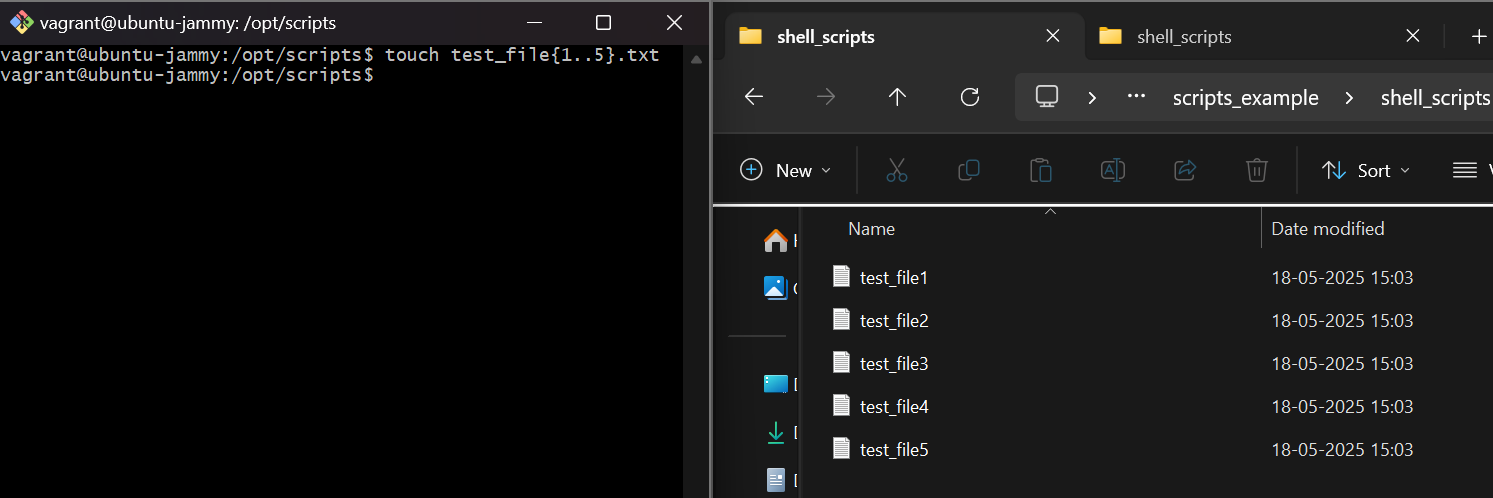
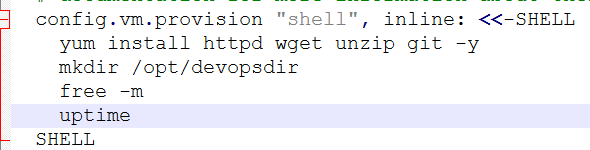
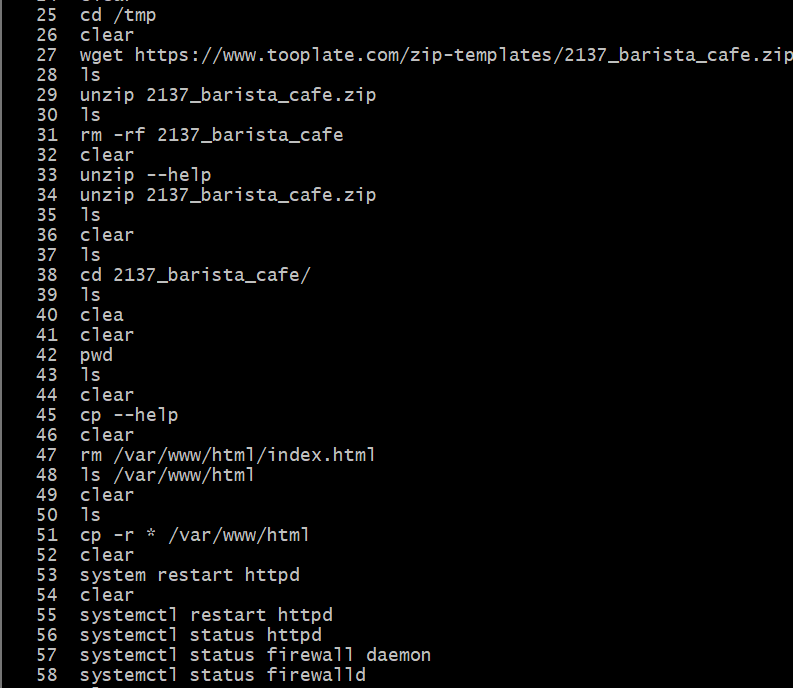
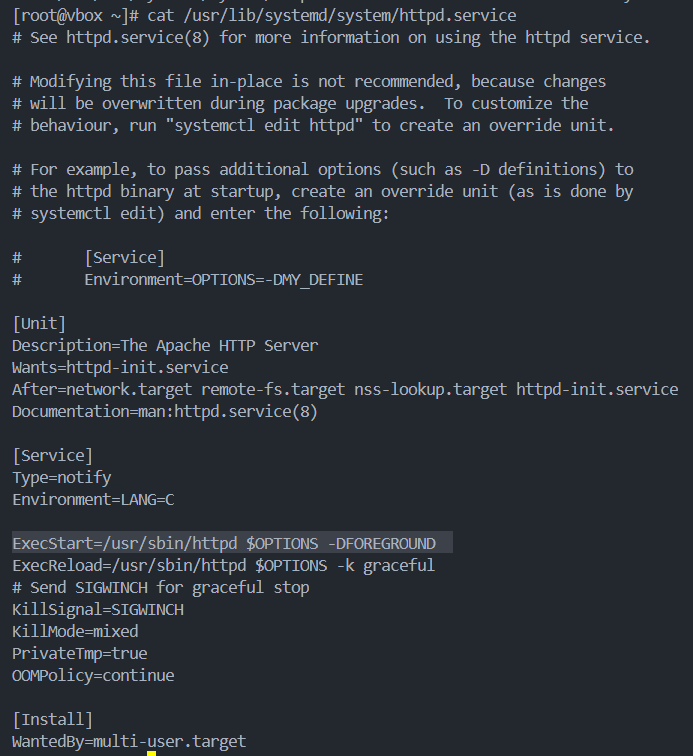
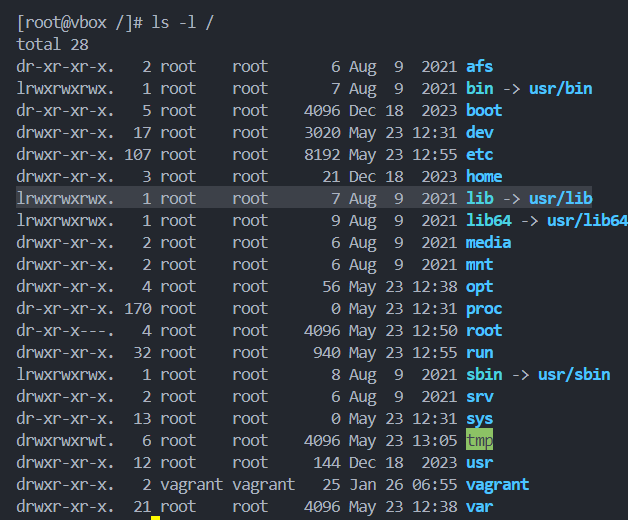
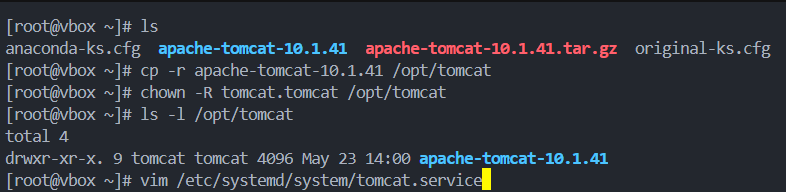
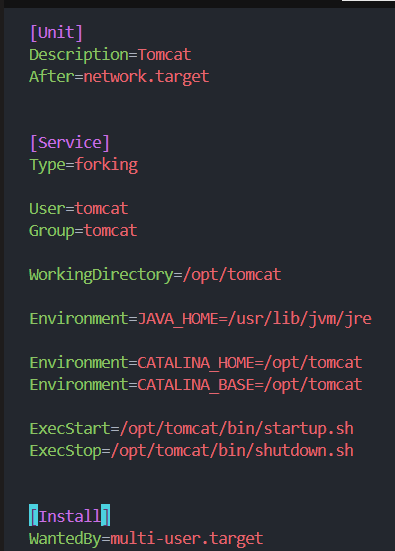
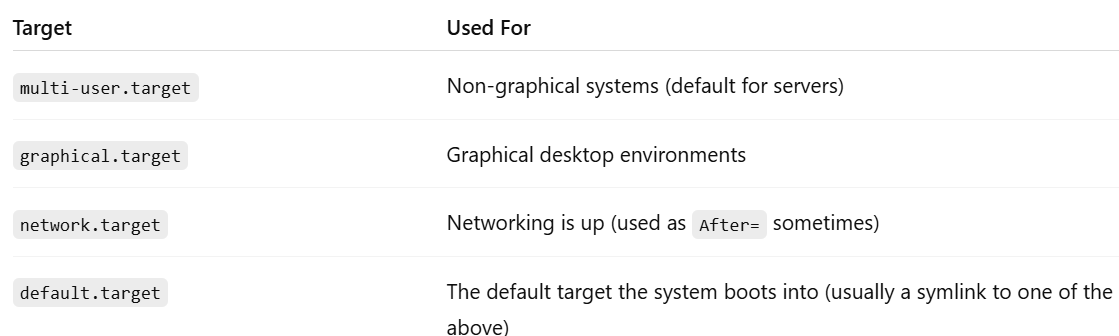
* **Configuring the Vagrantfile**
  + When u write “vagrant up”, at the end of the config a shared path is given I.e. /c/devops\_udemy/vagrant-vms/ubuntu => /vagrant
  + Here the first path is of the physical machine, and 2nd path is of the virtual machine.
  + Whatever u create inside that virtual machine on that path, same will be appeared in the physical machine as well and vice-versa.
    - 
      * It is the Vagrantfile.
    - 
      * I edited this. (remember: 2 backward slash in windows)
      * Note: You need to create folder in host(windows) machine. In guest(linux) machine, it’ll be automatically gets created.
    - 
      * Now, you can see 2 sync paths. One is default and one is that we created.
      * 
  + **Provisioning:**
    - 
    - Whatever we write inside that -SHELL and SHELL, will be executed while loading the Vagrantfile.
    - “**inline:**” means we are writing the command in the same file only
    - *These commands will be executed while creating the os only, not while reloading.*
    - To provision during the reload as well, use the flag *--provision.* i.e. **vagrant reload --provision**
* **Website**
  + 
  + Download the zip file (if you want to get from any website, if you have already then no need)
  + Move it to /var/www/html directory.
  + Start **httpd**
  + Now u can check in the browser using your guest machine IP (not using the NAT ip).
* **Multiple VM:**
  + vagrant up <vm name> (as multiple vms are there)
  + vagrant ssh <vm name>
  + vagrant destroy <vm name>
    - vagrant destroy (If not specified any vm, it’ll destroy all)
  + Ex:
  + 
* **Systemctl & Tomcat:**
  + **Apache Tomcat:**
    - Free, open-source Java servlet container.
    - It hosts Java-based web apps.
    - Dynamic content & handling web requests.
  + 
    - You can see, 3 things are there ([Unit], [Service], [Install])
    - When we run **systemctl start httpd** , it basically runs this command which is there next to “ExecStart”
  + 
    - **/lib** is nothing but a link to **/usr/lib**
  + 
    - This is how, we can run **tomcat**. (bin/startup.sh contains the code to start the tomcat. It’s just a shell script so we can directly run this to start tomcat).
    - *Write* ***ip addr show*** *and take an ip and paste this in browser. Do not forget to give the port* ***8080****. It’ll show display the default apache tomcat page. Ex:* ***192.168.60.226:8080***
    - **Abdd**
* **Systemctl work:**
  + When we execute the command **systemctl start <service>**, It’ll check the file **<service>.service** inside one of the following directories:
    - **/etc/systemd/system**
    - **/run/systemd/system**
    - **/usr/lib/systemd/system** or **/lib/systemd/system**
  + When you download a service from a package installer like **yum** or **dnf**, It creates a file like **<service>.service**, so you can give any commands using systemctl.
  + But tomcat is not installed by default, You’ll have to download this package from outside. So, you’ll have to run it’s **startup.sh** file to start tomcat.
* **To automate the enabling of tomcat even after reboot:**
  + **useradd --home-dir /opt/tomcat --shell /sbin/nologin tomcat**
  + 
  + 
    - **Environment=<variable\_name>=<variable\_value>**
    - It is used to define environmental variables.
  + **systemctl daemon-reload**
* Values of **WantedBy**
  + 
* All commands:

