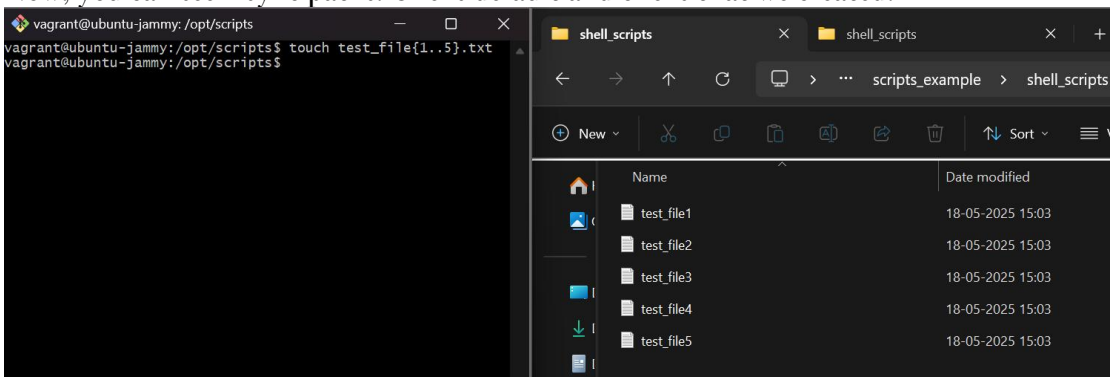
- **Configuring the Vagrantfile**
  - When u write "vagrant up", at the end of the config a shared path is given I.e. /c/devops_udemy/vagrant-vms/ubuntu => /vagrant
  - Here the first path is of the host machine, and 2nd path is of the guest machine.
  - Whatever u create inside that virtual machine on that path, same will be appeared in the physical machine as well and vice-versa.
    -
```
# Share an additional folder to the guest VM. The first argument is
# the path on the host to the actual folder. The second argument is
# the path on the guest to mount the folder. And the optional third
# argument is a set of non-required options.
# config.vm.synced_folder "../data", "/vagrant_data"
```
    - It is the Vagrantfile.
    -
```
# Share an additional folder to the guest VM. The first argument is
# the path on the host to the actual folder. The second argument is
# the path on the guest to mount the folder. And the optional third
# argument is a set of non-required options.
config.vm.synced_folder "C:\\devops_udemy\\vagrant-vms\\scripts_example\\shell_scripts", "/opt/scripts/"
```
    - 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.
    -
```
    default: Virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default:
    default: Guest Additions Version: 6.0.0 r127566
    default: VirtualBox Version: 7.1
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
    default: C:/devops_udemy/vagrant-vms/ubuntu => /vagrant
    default: C:/devops_udemy/vagrant-vms/scripts_example/shell_scripts => /opt/scripts
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
```
    - Now, you can see 2 sync paths. One is default and one is that we created.
    - 

  - **Provisioning:**
    -
```
config.vm.provision "shell", inline: <<-SHELL
  yum install httpd wget unzip git -y
  mkdir /opt/devopsdir
  free -m
  uptime
SHELL
```
    - 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**

```
25  cd /tmp
26  clear
27  wget https://www.tooplate.com/zip-templates/2137_barista_cafe.zip
28  ls
29  unzip 2137_barista_cafe.zip
30  ls
31  rm -rf 2137_barista_cafe
32  clear
33  unzip --help
34  unzip 2137_barista_cafe.zip
35  ls
36  clear
37  ls
38  cd 2137_barista_cafe/
39  ls
40  clea
41  clear
42  pwd
43  ls
44  clear
45  cp --help
46  clear
47  rm /var/www/html/index.html
48  ls /var/www/html
49  clear
50  ls
51  cp -r * /var/www/html
52  clear
53  system restart httpd
54  clear
55  systemctl restart httpd
56  systemctl status httpd
57  systemctl status firewall daemon
58  systemctl status firewalld
```

  - 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:

```
Vagrant.configure("2") do |config|
  # Shared synced folder
  config.vm.synced_folder ".", "/vagrant"

  # Define web01 VM
  config.vm.define "web01" do |web01|
    web01.vm.box = "ubuntu/focal64"
    web01.vm.hostname = "web01"
    web01.vm.network "private_network", ip: "192.168.56.51
    web01.vm.provider "virtualbox" do |vb|
      vb.name = "web01"
      vb.memory = 1024
      vb.cpus = 1
    end
  end

  # Define web02 VM
  config.vm.define "web02" do |web02|
    web02.vm.box = "ubuntu/focal64"
    web02.vm.hostname = "web02"
    web02.vm.network "private_network", ip: "192.168.56.52
    web02.vm.provider "virtualbox" do |vb|
      vb.name = "web02"
      vb.memory = 1024
      vb.cpus = 1
    end
  end

  # Define db01 VM (with provisioning)
  config.vm.define "db01" do |db01|
    db01.vm.box = "centos/7"
    db01.vm.hostname = "db01"
    db01.vm.network "private_network", ip: "192.168.56.53"
    db01.vm.provider "virtualbox" do |vb|
      vb.name = "db01"
      vb.memory = 2048
      vb.cpus = 2
    end

    # Shell provisioning script
    db01.vm.provision "shell", inline: <<-SHELL
      yum -y update
      yum -y install mariadb-server
      systemctl start mariadb
      systemctl enable mariadb
      echo "Database server provisioned and running."
    SHELL
  end
end
```

- **Systemctl & Tomcat:**
  - **Apache Tomcat:**
    - Free, open-source Java servlet container.
    - It hosts Java-based web apps.
    - Dynamic content & handling web requests.

```
[root@vbox ~]# cat /usr/lib/systemd/system/httpd.service
# See httpd.service(8) for more information on using the httpd service.

# Modifying this file in-place is not recommended, because changes
# will be overwritten during package upgrades.  To customize the
# behaviour, run "systemctl edit httpd" to create an override unit.

# For example, to pass additional options (such as -D definitions) to
# the httpd binary at startup, create an override unit (as is done by
# systemctl edit) and enter the following:

#       [Service]
#       Environment=OPTIONS=-DMY_DEFINE

[Unit]
Description=The Apache HTTP Server
Wants=httpd-init.service
After=network.target remote-fs.target nss-lookup.target httpd-init.service
Documentation=man:httpd.service(8)

[Service]
Type=notify
Environment=LANG=C

ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
# Send SIGWINCH for graceful stop
KillSignal=SIGWINCH
KillMode=mixed
PrivateTmp=true
OOMPolicy=continue

[Install]
WantedBy=multi-user.target
```

- 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"

```
[root@vbox /]# ls -l /
total 28
dr-xr-xr-x.   2 root    root       6 Aug  9  2021 afs
lrwxrwxrwx.   1 root    root       7 Aug  9  2021 bin -> usr/bin
dr-xr-xr-x.   5 root    root    4096 Dec 18  2023 boot
drwxr-xr-x.  17 root    root    3020 May 23 12:31 dev
drwxr-xr-x. 107 root    root    8192 May 23 12:55 etc
drwxr-xr-x.   3 root    root      21 Dec 18  2023 home
lrwxrwxrwx.   1 root    root       7 Aug  9  2021 lib -> usr/lib
lrwxrwxrwx.   1 root    root       9 Aug  9  2021 lib64 -> usr/lib64
drwxr-xr-x.   2 root    root       6 Aug  9  2021 media
drwxr-xr-x.   2 root    root       6 Aug  9  2021 mnt
drwxr-xr-x.   4 root    root      56 May 23 12:38 opt
dr-xr-xr-x. 170 root    root       0 May 23 12:31 proc
dr-xr-x---.   4 root    root    4096 May 23 12:50 root
drwxr-xr-x.  32 root    root     940 May 23 12:55 run
lrwxrwxrwx.   1 root    root       8 Aug  9  2021 sbin -> usr/sbin
drwxr-xr-x.   2 root    root       6 Aug  9  2021 srv
dr-xr-xr-x.  13 root    root       0 May 23 12:31 sys
drwxrwxrwt.   6 root    root    4096 May 23 13:05 tmp
drwxr-xr-x.  12 root    root     144 Dec 18  2023 usr
drwxr-xr-x.   2 vagrant vagrant   25 Jan 26 06:55 vagrant
drwxr-xr-x.  21 root    root    4096 May 23 12:38 var
```

- **/lib** is nothing but a link to **/usr/lib**

```
[root@vbox apache-tomcat-10.1.41]# pwd
/root/apache-tomcat-10.1.41
[root@vbox apache-tomcat-10.1.41]# ls bin
bootstrap.jar  catalina-tasks.xml  commons-daemon.jar          configtest.sh  digest.sh     migrate.bat       setclasspath.sh  startup.bat      tomc
catalina.bat   ciphers.bat         commons-daemon-native.tar.gz daemon.sh     makebase.bat  migrate.sh        shutdown.bat     startup.sh       tool
catalina.sh    ciphers.sh          configtest.bat               digest.bat     makebase.sh   setclasspath.bat  shutdown.sh      tomcat-juli.jar  tool
[root@vbox apache-tomcat-10.1.41]# bin/startup.sh
Using CATALINA_BASE:   /root/apache-tomcat-10.1.41
Using CATALINA_HOME:   /root/apache-tomcat-10.1.41
Using CATALINA_TMPDIR: /root/apache-tomcat-10.1.41/temp
Using JRE_HOME:        /
Using CLASSPATH:       /root/apache-tomcat-10.1.41/bin/bootstrap.jar:/root/apache-tomcat-10.1.41/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@vbox apache-tomcat-10.1.41]# pw -ef | grep tomcat
-bash: pw: command not found
[root@vbox apache-tomcat-10.1.41]# ps -ef | grep tomcat
root       10329       1 68 13:16 pts/0    00:00:19 //bin/java -Djava.util.logging.config.file=/root/apache-tomcat-10.1.41/conf/logging.properties -D
Size=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dsun.io.useCanonCaches=false -Dorg.apache.catalina.security.SecurityListener.
reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=A
omcat-10.1.41/bin/bootstrap.jar:/root/apache-tomcat-10.1.41/bin/tomcat-juli.jar -Dcatalina.base=/root/apache-tomcat-10.1.41 -Dcatalina.home=/root/apac
.startup.Bootstrap start
root       10362    7777  0 13:16 pts/0    00:00:00 grep --color=auto tomcat
```

  - 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**

```
[root@vbox ~]# ls
anaconda-ks.cfg  apache-tomcat-10.1.41  apache-tomcat-10.1.41.tar.gz  original-ks.cfg
[root@vbox ~]# cp -r apache-tomcat-10.1.41 /opt/tomcat
[root@vbox ~]# chown -R tomcat.tomcat /opt/tomcat
[root@vbox ~]# ls -l /opt/tomcat
total 4
drwxr-xr-x. 9 tomcat tomcat 4096 May 23 14:00 apache-tomcat-10.1.41
[root@vbox ~]# vim /etc/systemd/system/tomcat.service
```

```
[Unit]
Description=Tomcat
After=network.target


[Service]
Type=forking

User=tomcat
Group=tomcat

WorkingDirectory=/opt/tomcat

Environment=JAVA_HOME=/usr/lib/jvm/jre

Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh


[Install]
WantedBy=multi-user.target
```

- Environment=<variable_name>=<variable_value>
- It is used to define environmental variables.

- **systemctl daemon-reload**

➢ Values of **WantedBy**

Target	Used For
`multi-user.target`	Non-graphical systems (default for servers)
`graphical.target`	Graphical desktop environments
`network.target`	Networking is up (used as `After=` sometimes)
`default.target`	The default target the system boots into (usually a symlink to one of the above)

- All Commands:

```
#Download tomcat
wget https://archive.apache.org/dist/tomcat/tomcat-10/v10.1.28/bin/apache-tomcat-10.1.28.tar.gz
#Extract tomcat
tar xzvf apache-tomcat-10.1.28.tar.gz
#Create tomcat user
useradd --home-dir /opt/tomcat --shell /sbin/nologin tomcat
#Copy files to tomcat home directory
cp -r apache-tomcat-10.1.28/* /opt/tomcat/
#Give tomcat user ownership to tomcat directory
chown -R tomcat.tomcat /opt/tomcat/
#Create system file for tomcat service
vim /etc/systemd/system/tomcat.service

#Paste below content ------------------------------------------------------------
[Unit]
Description=Tomcat
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

WorkingDirectory=/opt/tomcat

Environment=JAVA_HOME=/usr/lib/jvm/jre
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINE_BASE=/opt/tomcat

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
# ------------------------------------------------------------------------------

# Reload system config changes
systemctl daemon-reload
#Start & Enable tomcat service
systemctl start tomcat
systemctl status tomcat
systemctl enable tomcat
```