

Programme: BSc Cloud Computing

Module: ACCA5029 Network Programmability

Lecturer: Dr. Muhamamd Azeem Abbas

Assignment Title: Task 2: Report and demonstration

Student Name: Carol Cernea

Student ID: 2113437

Table of Contents

Background of Ansible	3
Advantages.....	3
Disadvantage.....	4
Features of Ansible	5
Creating VM machine.	9
Installation and confirmation of Ansible.....	16
Installation and confirmation of Apache2 Server	18
Installation and confirmation of MySQL.....	20
Installation of MyPhpAdmin	23
References	26

Background of Ansible

Ansible has been created by Red Hat in 2015 and is an open-source tool and give us the ability to configure, managing and administrating the servers from one single location.

Ansible can be used to provision the underlying infrastructure of your environment, virtualized hosts and hypervisors, network devices, and bare metal servers. It can also install services, add compute hosts, and provision resources, services, and applications inside of your cloud.

Ansible works by connecting to your nodes and pushing out small programs—called modules—to these nodes. Modules are used to accomplish automation tasks in Ansible. These programs are written to be resource models of the desired state of the system. Ansible then executes these modules and removes them when finished. Furthermore, you can also use Ansible to run ad-hoc commands. To do this, you will need to run a command or call a module directly from the command line. Moreover, Ansible It is very rapid platform, all processes that are carried out through SSH, and also, agent installation is not necessary, it is open source and free software and is very simple to learn YAML language with a minimum knowledge. [3]

Advantages

Simple to Learn

The foremost mention among advantages of Ansible refers to its simplicity. The simplicity is not only meant for professionals but also for beginners. It is easy to learn, and so, users could learn to use Ansible quickly along with better productivity. Ansible receives the support of comprehensive and easily interpretable documentation.

Therefore, you can learn the logic of Ansible operations and the workflow in a limited period. The absence of a dependency structure can indicate that Ansible jobs run sequentially and terminate when an error is detected. As a result, even while learning the basics of ansible, troubleshooting becomes much simpler. [1]

Easily Understandable Python Language

One of the prominent advantages of Ansible also refers to the language in which it is written. Python is a human-readable language and serves as the basis for Ansible. It provides better facilities for getting up Ansible and running it due to the presence of Python libraries on the majority of Linux distributions by default.

Python is a very good substitute for administrative and scripting tasks, which suggests that engineers and system administrators choose it more. The availability of Ansible modules, which can enhance its usefulness, is another intriguing element of Ansible. Any language can be used to create Ansible modules. The module should deliver data in JSON format, though, as that is the crucial issue here. [1]

No Dependency on Agents

The next important addition among the benefits of Ansible refers to its agentless nature. Ansible manages all the master-agent communications through Standard SSH or Paramiko module. The Paramiko module is a Python implementation of SSH2 and is crucial for managing nodes. In order to

ensure management, Ansible does not need any kind of agents deployed on remote computers. As a result, Ansible significantly reduces maintenance costs and performance deterioration. [1]

Playbooks are written in YAML

The use of Playbooks in Ansible is also another reason for the major advantages of Ansible. Playbooks are Ansible configuration files, and the language for writing them is YAML. The interesting factor, in this case, is that YAML is a better alternative for configuration management and automation.

Ansible is a better configuration management and automation tool since YAML is superior to other formats like JSON. Ansible supports comments and makes it simple to read. The usage of anchors to refer to other items is also included, which is of utmost importance. [1]

Ansible Galaxy

Another notable entry in the Ansible best practices refers to the Ansible Galaxy. Ansible Galaxy is a portal that acts as the central repository for locating, reusing, and sharing Ansible-related content. The best advantage of Ansible Galaxy is in the example of downloading reusable Roles for installing application or server configuration. The downloads are ideal for use in a particular user's playbooks and can contribute substantially to an increase in deployment speed. [1]

Disadvantage

Insufficient User Interface

The first entry in the disadvantages of Ansible is the crude user interface. Ansible was initially a command-line only tool. The first effort of Ansible at making a user interface was with AWX graphical user interface. The other component in the UI was the REST endpoint that is meant for easier infrastructure management.

The AWX later evolved into the Ansible Tower, an online management UI. Ansible Tower provides a team-based workflow tool as well as visual management tools. The Ansible Tower, however, needs significant changes. For instance, the UI can be used to accomplish almost 85% of tasks that can be accomplished through the command line. [1]

Lack of any Notion of State

Another prominent mention among the disadvantages of Ansible is the lack of any notion of state. Ansible does not have any notion of state like other automation tools such as Puppet. Ansible does not track dependencies and simply executes sequential tasks and stops when tasks finish, fail, or any error comes.

For users who require the automation tool to keep a comprehensive catalogue for ordering, these characteristics are not appropriate. The catalogue can assist in getting to a particular state devoid of the influence of alterations in the surroundings. Ansible, however, is lacking it and is at a severe disadvantage. [1]

Limited Windows Support

The next prominent mention among Ansible disadvantages is the half-built Windows support. Ansible version 1.7 supports Windows as well as Linux/Unix nodes. In the case of Windows, Ansible employs a native PowerShell remoting rather than SSH. As a result, a Linux control machine is

mandatory for the management of Windows hosts. The limited support for Windows in Ansible presents one of the formidable setbacks with the configuration management and automation tool. [1]

Ansible does not have Experience

The lack of enterprise support experience also draws down the appeal of Ansible. Ansible does not have a full-fledged working experience with large enterprises like its competitors, such as Puppet and Chef (Read: Chef vs Puppet). Even though Ansible claims the facility of enterprise-grade extended support options, limited practical experience reduces the accountability of Ansible.[1]

Ansible is New to the Market

Finally, you can note one of the most common entries in Ansible advantages and disadvantages as a prominent setback of Ansible. Ansible is new to the market, unlike its renowned competitors. As a result, it does not have a large developer or user community. Furthermore, the new presence of Ansible on the market implies the possibilities of undiscovered bugs, software issues, and edge scenarios. [1]

In summary, Ansible offers a simple and effective solution for configuration management and automation. Ansible, however, is a quiet new tools on the market and must contend with opposition from well-known providers. Moreover, ansible is part of Infrastructure as Code (IaC), One great way to address this challenge is by adopting infrastructure as code (IaC), a solution that can save your IT team time and effort (and therefore money). IaC is a process of managing and provisioning computer data centers through configuration files rather than physical hardware configuration or interactive configuration tools. IaC is available for various hardware and operating systems. [2]

Features of Ansible

Even building an application or running complex infrastructure we can use very well the Ansible. Some of the feature so ansible are:

- Models the IT infrastructure around the systems interrelating with each other, thus ensuring faster end results.
- Module library can reside on any system, without the requirement of any server, daemons or databases.
- No additional setup required, so once you have the instance ready you can work on it straight away.
- Easier and faster to deploy as it doesn't rely on agents or additional custom security infrastructure.
- Uses a very simple language structure called playbooks. Playbooks are almost similar to the plain English language for describing automation jobs.
- Ansible has the flexibility to allow user-made modules that can be written in any programming language such as Ruby, Python. It also allows adding new server side behaviors extending Ansible's connection types through Python APIs. [4]

Playbooks are written in YAML format. YAML stands for Yet Another Markup Language. Playbooks are one of the core features of Ansible and tell Ansible what to execute. They are like a to-do list for

Ansible that contains a list of tasks. [5] All playbooks should contain tasks or a list of tasks to be executed. Tasks are a list of actions one needs to perform. A tasks field contains the name of the task. This works as the help text for the user. It is not mandatory but proves useful in debugging the playbook. Each task internally links to a piece of code called a module. A module that should be executed, and arguments that are required for the module you want to execute. [5]

According to Sarav, AK (Middlewareinventory.com) There are four ways of running playbooks. [6]

Method1: Specify Localhost in your hosts directive of your playbook.

If you are running a playbook which you want to run on localhost (or) in other words you have a playbook you want to run locally. Within the playbook Just mention localhost in the hosts segment where you usually specify the host group. Consider the following playbook.

I have just specified the localhost in the hosts directive or key.

When you run this playbook. Despite you have no entry named localhost in your ansible inventory file /etc/ansible/hosts or custom hosts file specified with -i option. It would work

As an additional measure that it should run in localhost or locally. You can add a one more parameter named connection and set it to local [6]

```
---

- name: "Playing with Ansible and Git"

  hosts: localhost

  connection: local

  tasks:

    - name: "just execute a ls -lrt command"

      shell: "ls -lrt"

      register: "output"

    - debug: var=output.stdout_lines
```

Method2: Using local_action clause in the ansible playbook

This method is most recommended than the other 3 but it is always subjective to use it for your requirements.

Here is the example playbook that runs locally to create an EC2 instance connecting to AWS.

if you notice, we have not specified any hosts in here alternatively we have used local_action clause to specify that this task has to run local. [6]

```
- name: create an ec2 instance
```

```
  local_action:
```

```
    module: ec2
```

```
    image: ami-8caa1ce4
```

```
    instance_type: m3.medium
```

```
    key_name: mykey
```

```
    group:
```

```
      - web
```

```
      - ssh
```

```
    instance_tags:
```

```
      type: web
```

```
      env: production
```

Method3: Add an entry in your Inventory

This is a second method to run ansible playbook locally.

You can also explicitly define your localhost in your inventory file. your inventory file can be at the default location /etc/ansible/hosts or a customized ansible_hosts file in your present working directory. [6]

```

+ Ansible-Examples git:(master) x cat ansible_hosts Local Ansible_hosts file / Inventory
[app]
mwivmapp01
mwivmapp02

localhost ansible_connection=local
+ Ansible-Examples git:(master) x cat /etc/ansible/hosts Default Inventory
[app]
mwivmapp01
mwivmapp02

localhost ansible_connection=local
+ Ansible-Examples git:(master) x

```

Method4: Specify in the Ansible Command line

This is a third method to run ansible playbook locally.

By Default, Ansible would run the playbook on the host group which is mentioned in the playbook with hosts: directive.

But if you want to ignore all those hosts specified in the playbook and run it locally. You can use this method.

In this way. The hosts directive would be ignored, and your task would run only on localhost.

I have a host group named app in my Ansible inventory file, it has two hosts beneath it and the playbook is defined to run on all the hosts in my inventory with hosts: all [6]

```

+ Ansible-Examples git:(master) x cat ansible_hosts ## MY INVENTORY FILE
[app]
mwivmapp01
mwivmapp02

controlmachine ansible_connection=local
+ Ansible-Examples git:(master) x cat Ansible-Local.yml ## MY PLAYBOOK
---
- name: "Playing with Ansible and Git"
  hosts: all
  tasks:

    - name: "just execute a ls -lrt command"
      shell: "ls -lrt"
      register: "output"

    - debug: var=output.stdout_lines

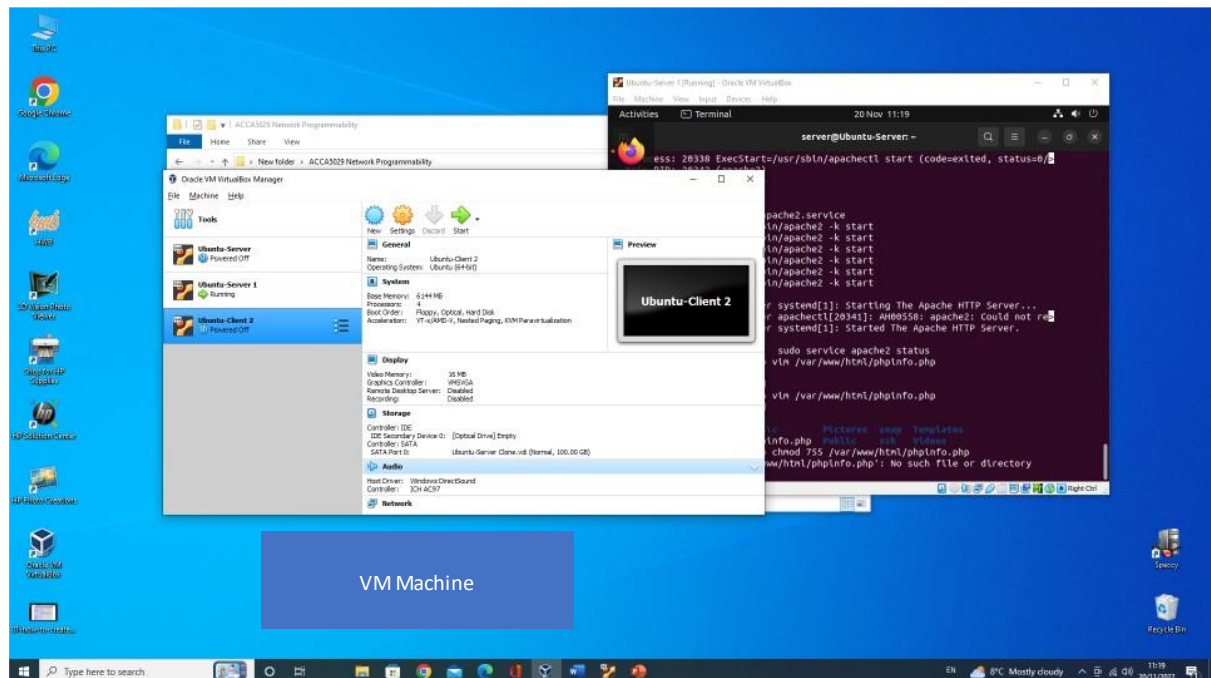
+ Ansible-Examples git:(master) x

```

Creating VM machine.

As we can see in this picture below, we created two Virtual Machines using Oracle VM VirtualBox

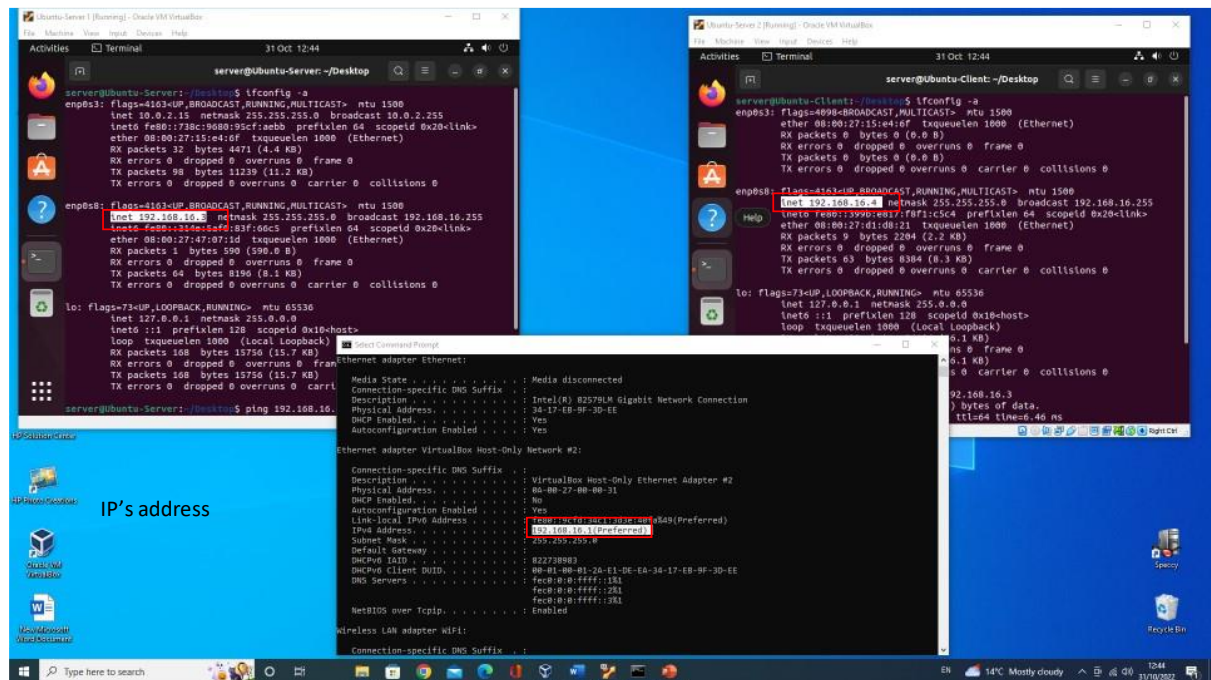
One machine will be Client and the second one Server



IP addresses

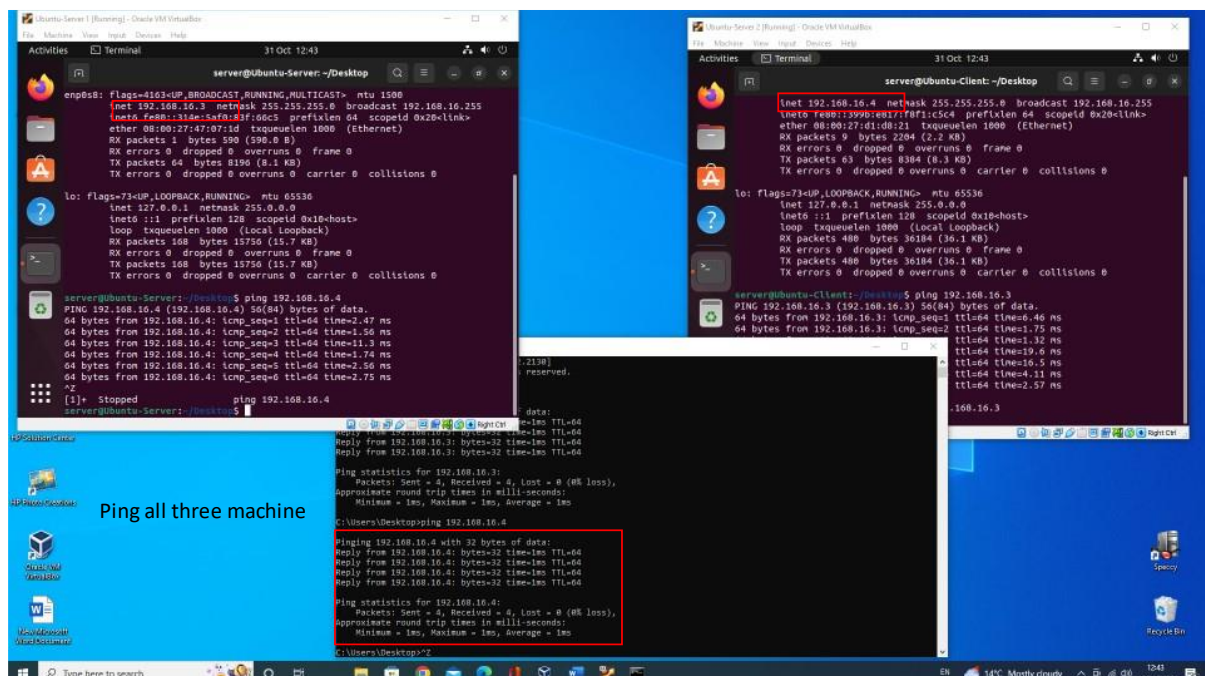
After installing correctly VM machine we bridge the network and as you can see in the picture below using the **ip a** command the bridge was successfully and each machine got its own IP

address

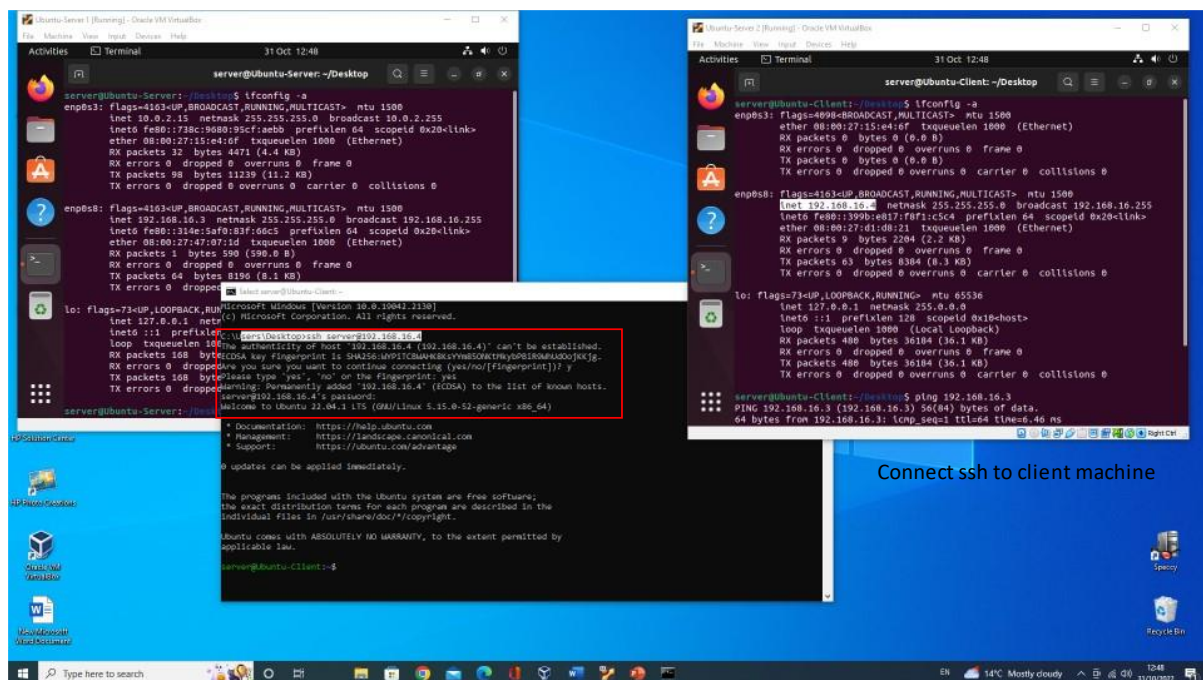


Ping all machines

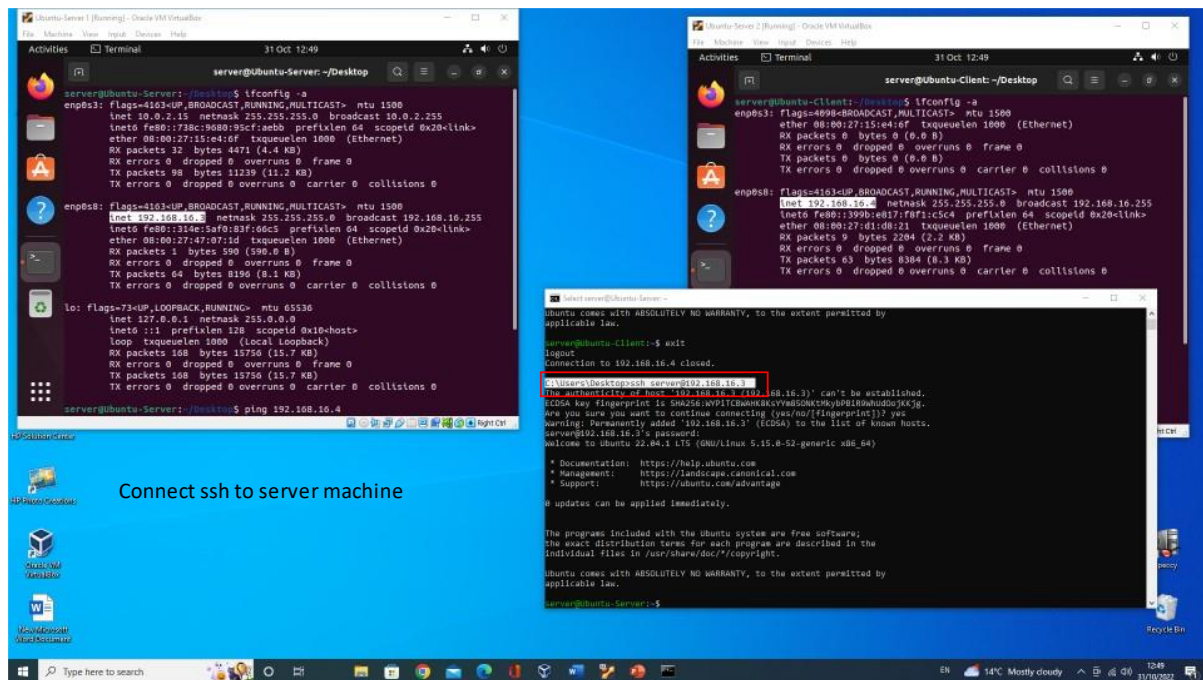
Below is a successfully demonstration on pingging all the machines. This show us that all the machine can communicate.



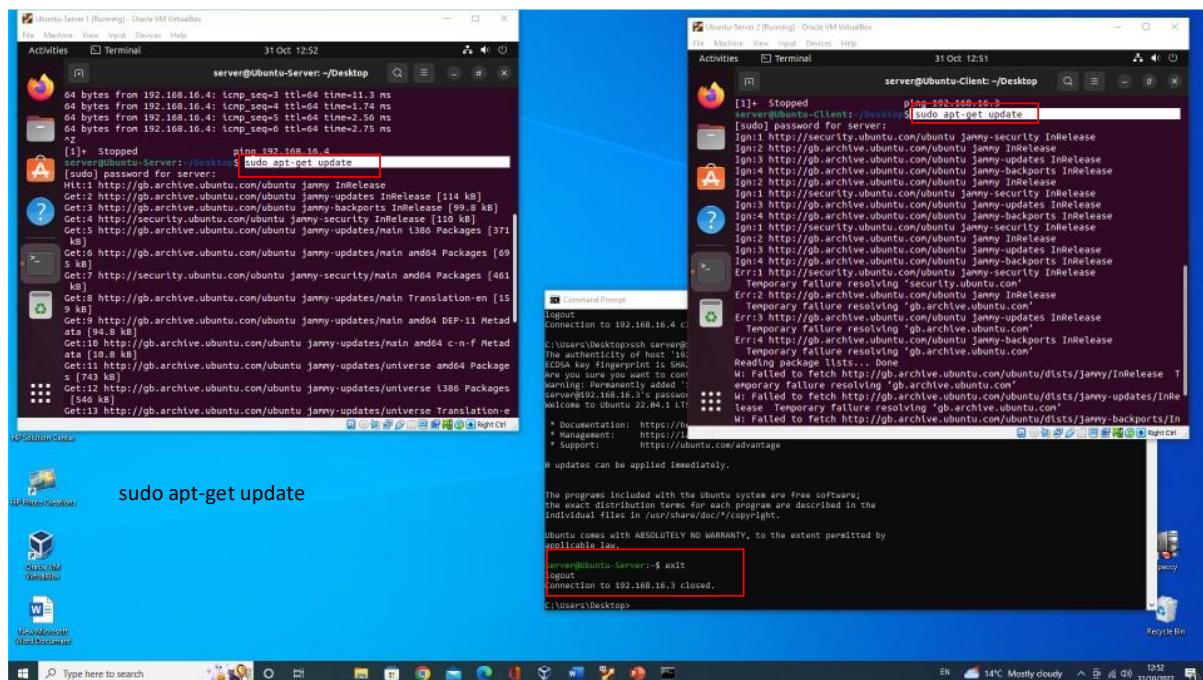
Connect SSH to Ubuntu client



Connect SSH to Ubuntu Server Machine

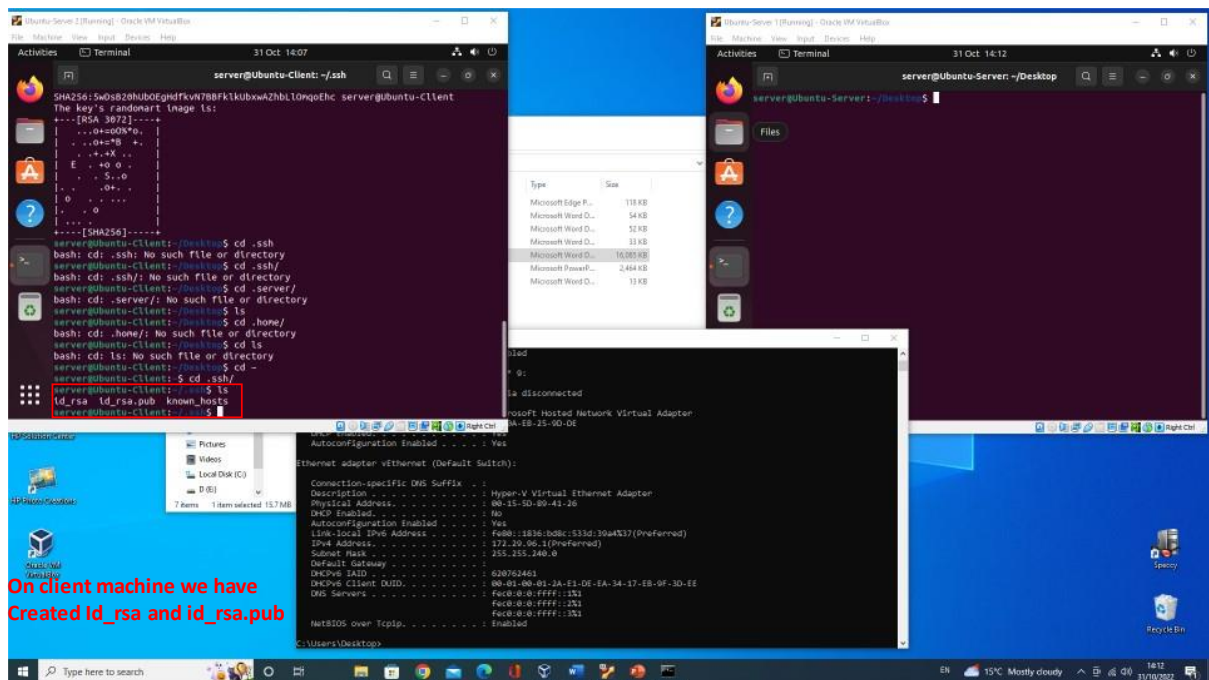


sudo apt-get update



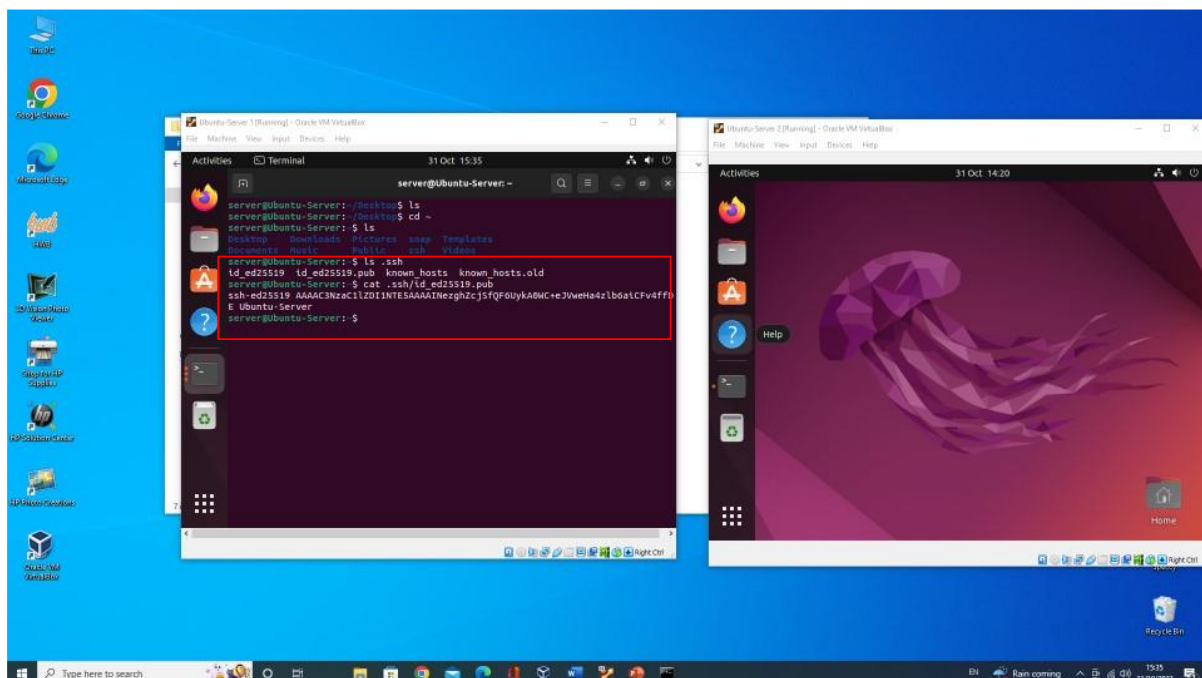
Connect to Client machine

As you can see below we have been using Command Prompt (IP address 192.168.16.1) to connect to Client ubuntu machine (IP address 192.168.16.3)

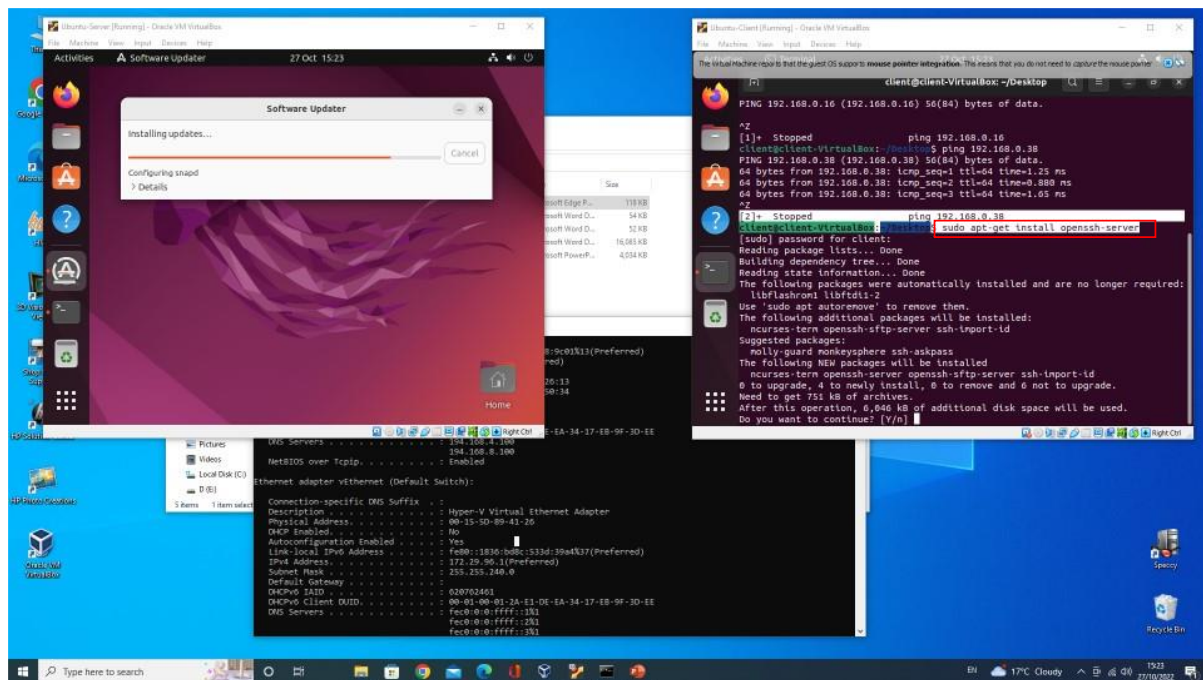


The following command will help us to see if the key is in the ssh director.

ls .ssh

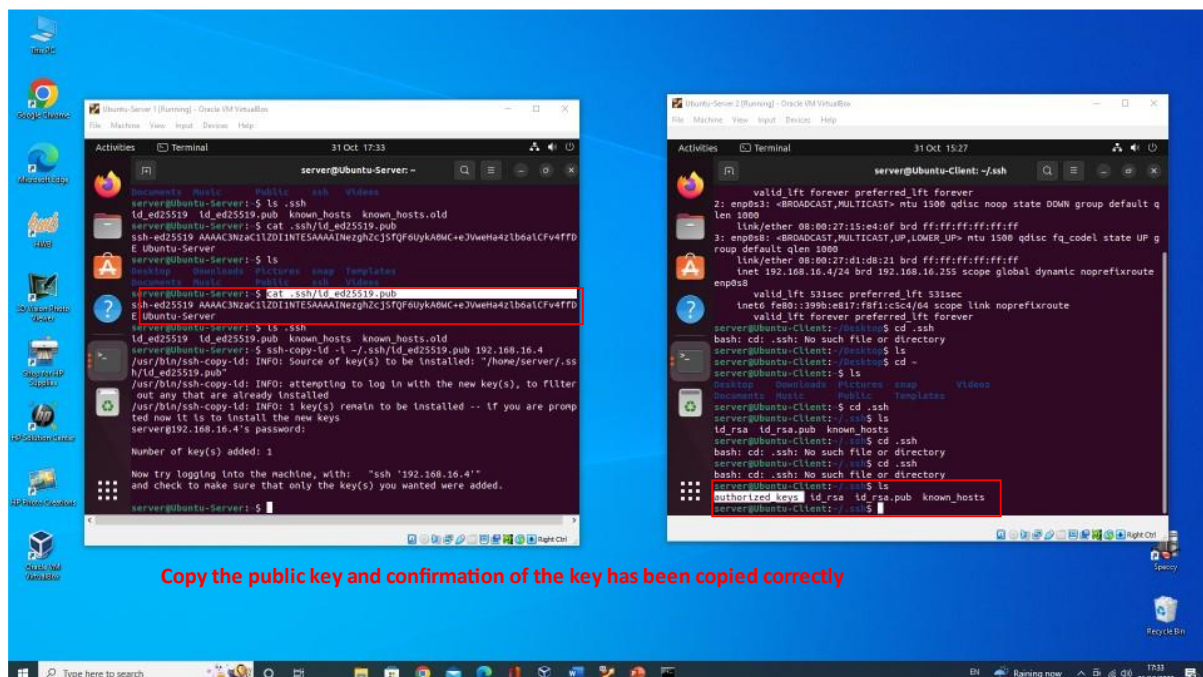


sudo apt-get install openssh-server



Copy the public key, and confirmation that key has been copied correctly

As you can see on the left machine has been given command `cat .ssh/id_ed25519.pub` for copying the public key which has been generated earlier. Furthermore, on the right machine using the command `cd .ssh` to go into this directory we used `ls` command to check if the key has been copied correctly.



Copy the public key and confirmation of the key has been copied correctly

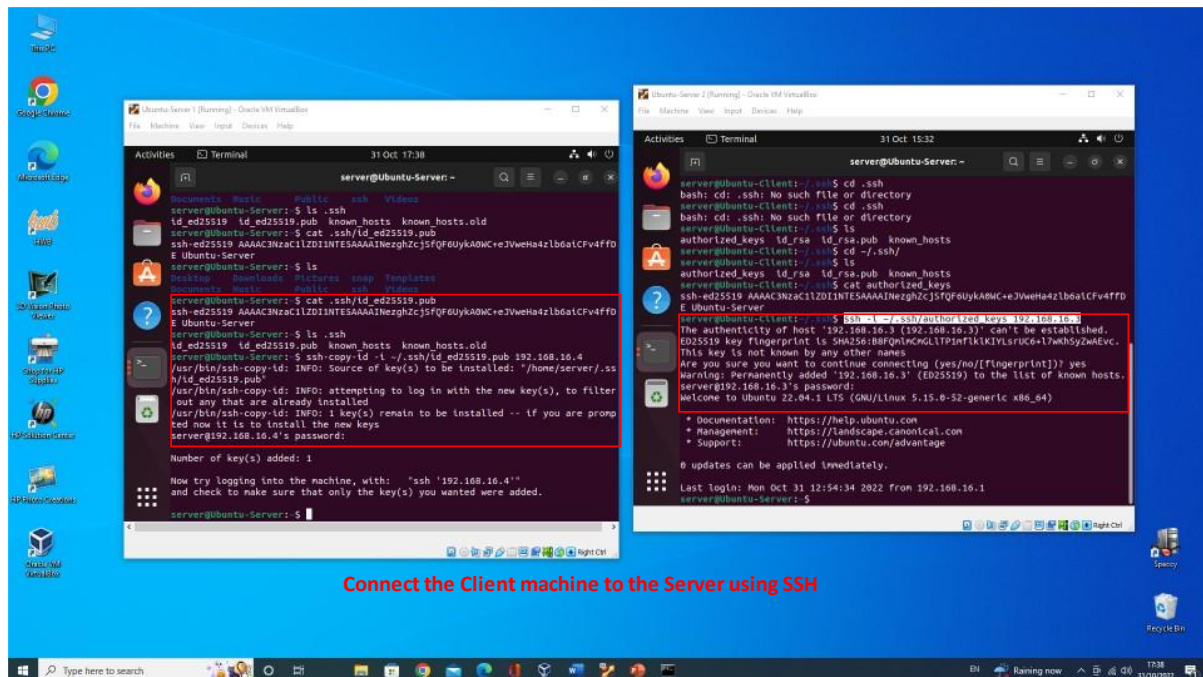
Cat .ssh/id_ed25519.pub

Server command

Ssh-copy-id -I ~/.ssh/id_ed25519.pub 192.168.16.4

Client command

ssh -I ~/.ssh/authorized_key 192.168.16.3



Connect the Client machine to the Server using SSH

Installation and confirmation of Ansible

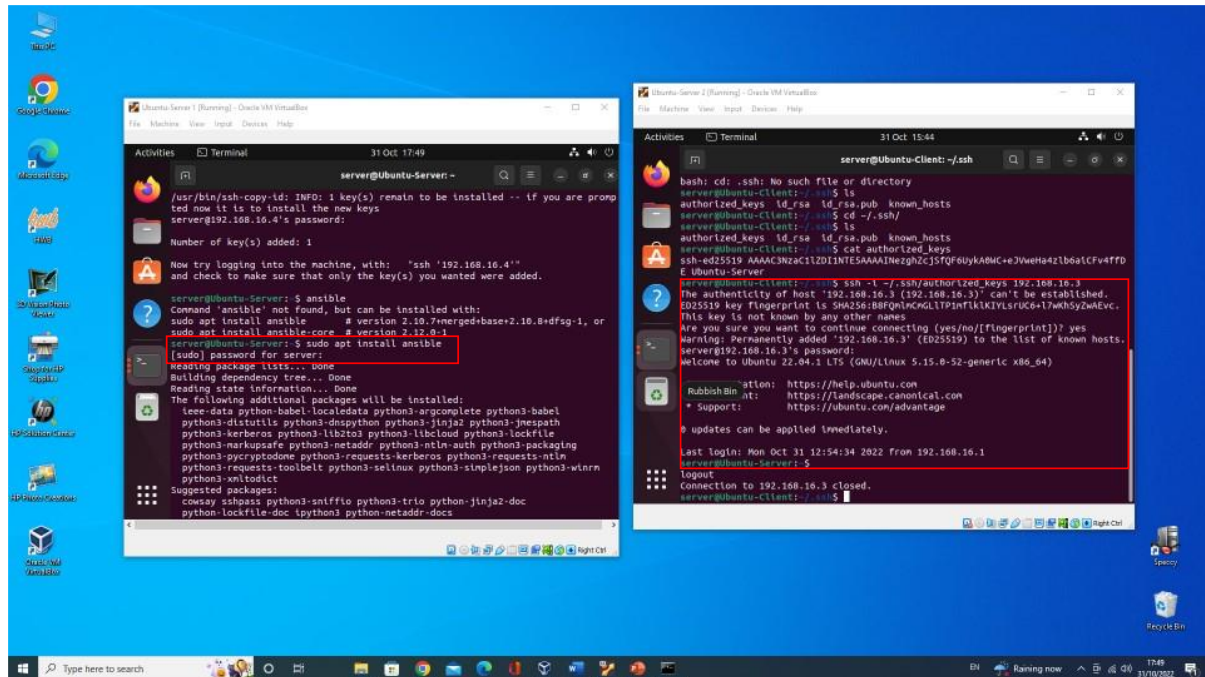
ANSIBLE

Here will be showing how to install the Ansible using apt.

Sudo apt install ansible

After this command is completed usually, we need to add user to sudo group using the following command

Sudo adduser ansible sudo



Ping the machine using ansible from Server Machine

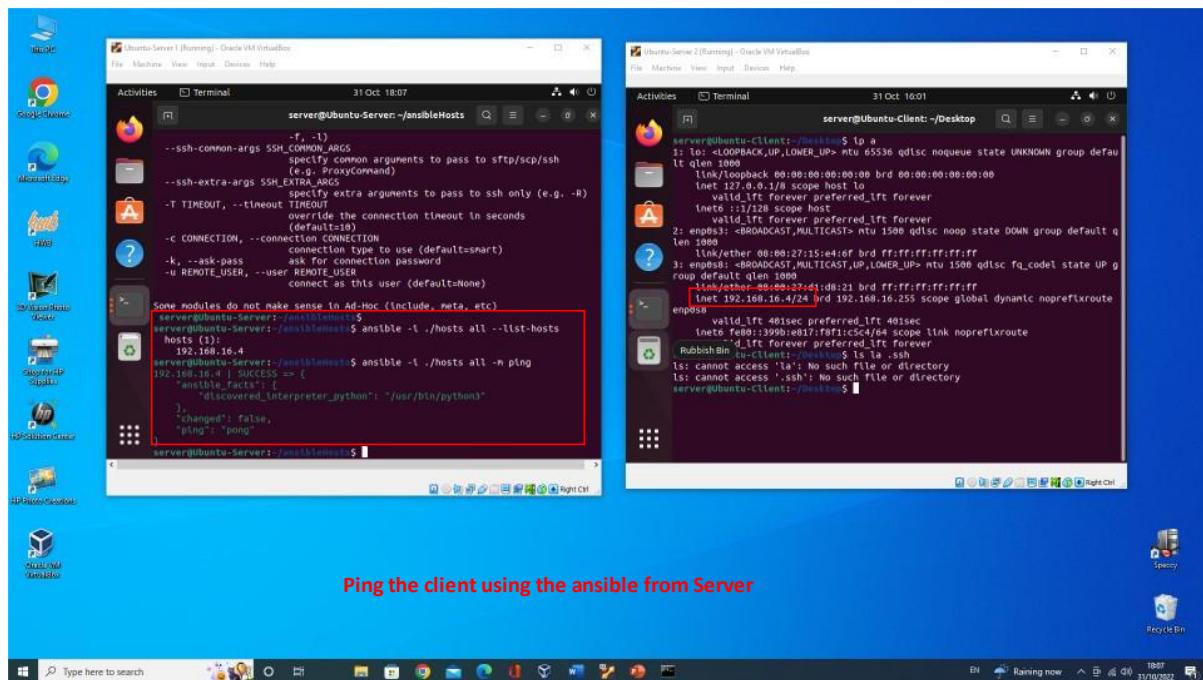
Bellow will be showing how we can connect to the server and how to ping the machine using ansible. The command for checking the hosts is:

```
ansible -i ./hosts all -l list-hosts
```

And to ping the host we used the command

```
ansible -i ./host all -m ping
```

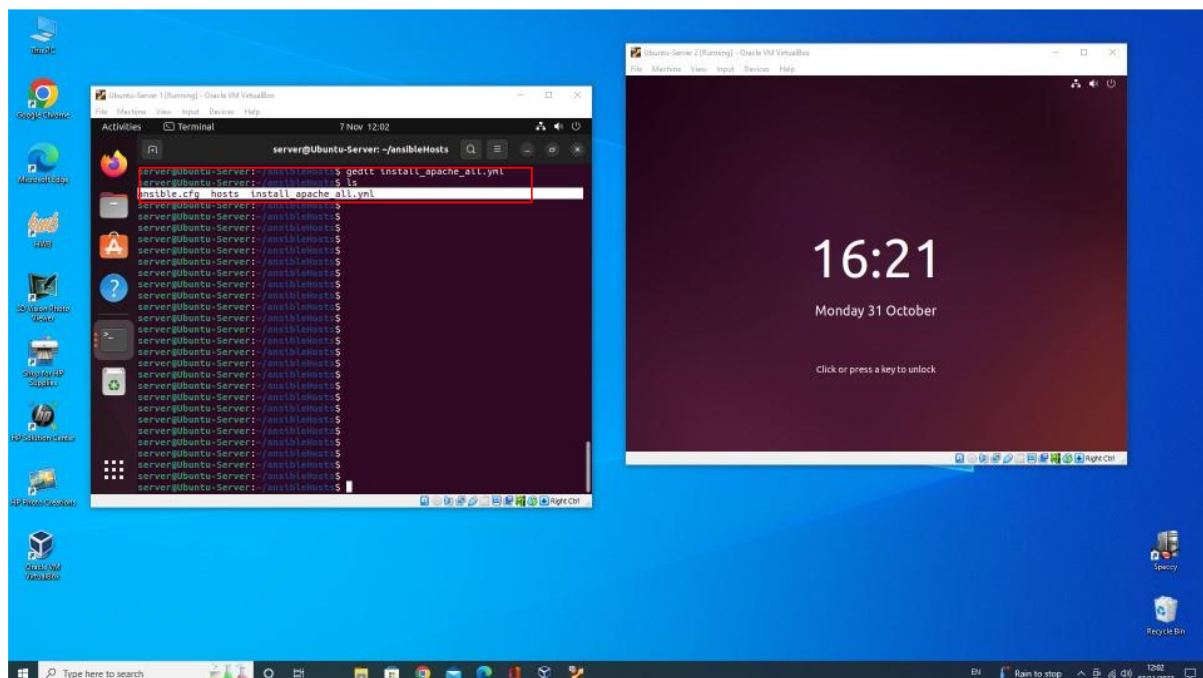
As you can see bellow the command has been successfully and there is one host with IP **192.168.16.4** which is the same IP address as **Client Machine**



Installation and confirmation of Apache2 Server

Install the Apache

Sudo apt-get install apache2

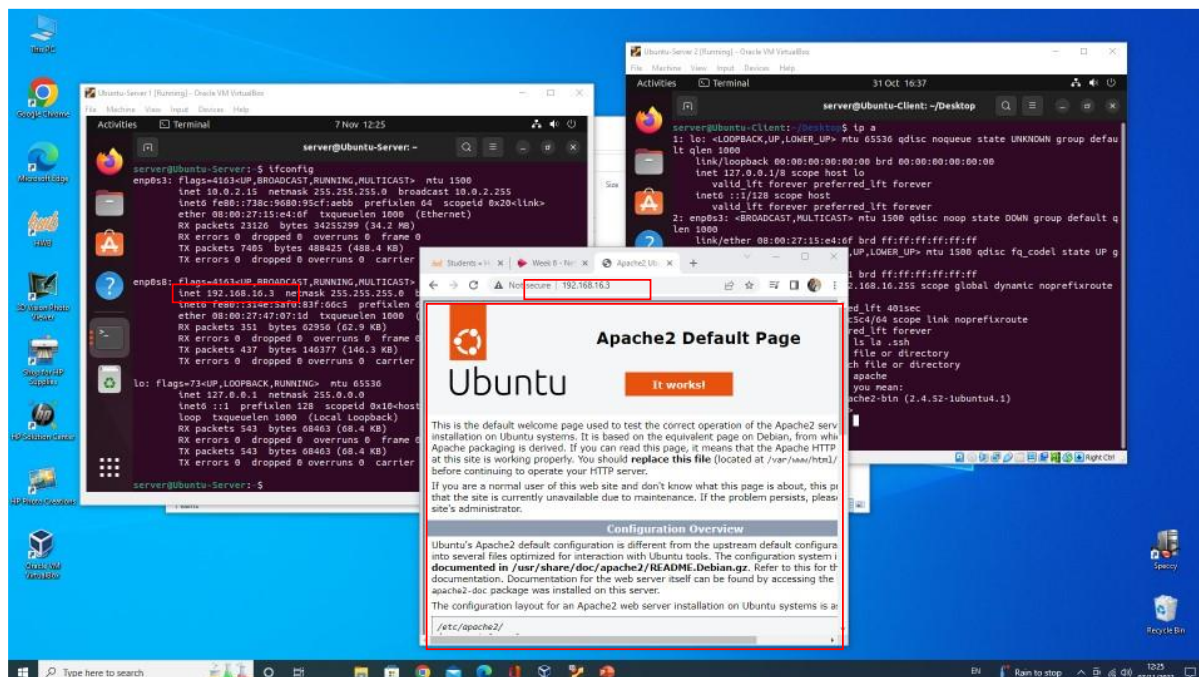


Confirmation that Apache server has been installed correctly,

Home Page

As you can see in the picture bellow the IP address which we are using now to connect to Apache2 Default page is same as Server IP address

192.168.16.3



After installation is done, we will need to start Apache server with the following command

Sudo systemctl start apache2

Furthermore, Apache server we will need to enable to start automatically following next command

Sudo systemctl enable apache2

As soon as Apache server is installed correctly and running properly, we can start the configuration of the Apache. The location of the server will be: **/etc/apache2/apache2.conf**

Now we need to change the **ServerName** directive, this directive will set up the hostname that Apache server will use and we need to match the host of the Apache server. The location have a default location **/var/www/html** and in our case we need to change to **/var/www/html/servername**

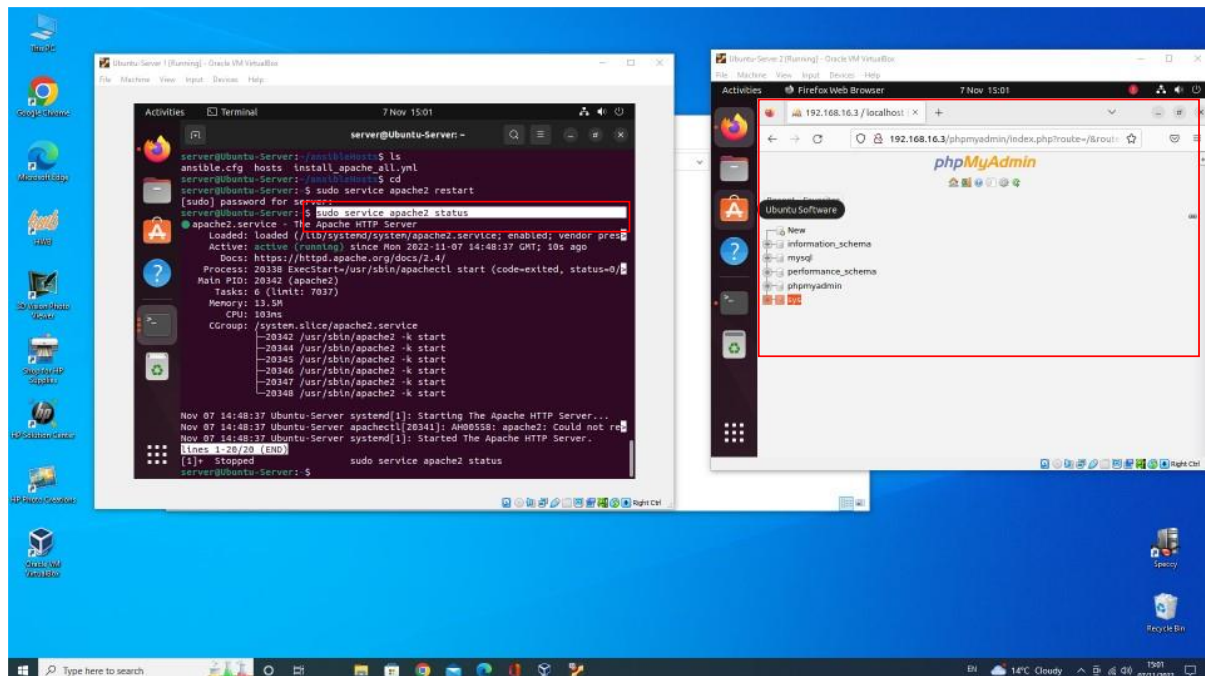
After every change the Apache server need restart. For this we going to use the command

Sudo systemctl restart apache2

Apache status

To check and see the status of Apache server we can use the following command

Sudo service apache2 status



Installation and confirmation of MySQL

Install the MySQL using the command

The screenshot displays a virtual machine environment with the title bar 'Ubuntu Server 1 [Running] - Oracle VM VirtualBox'. The terminal window shows the following commands and output:

```

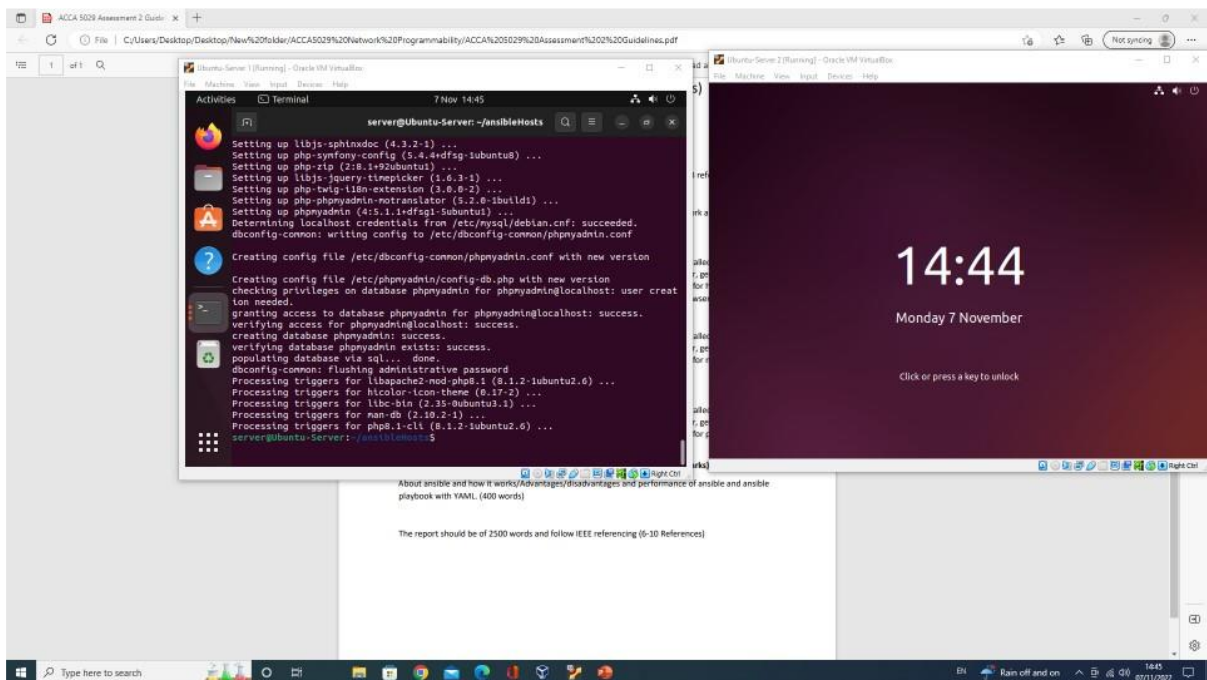
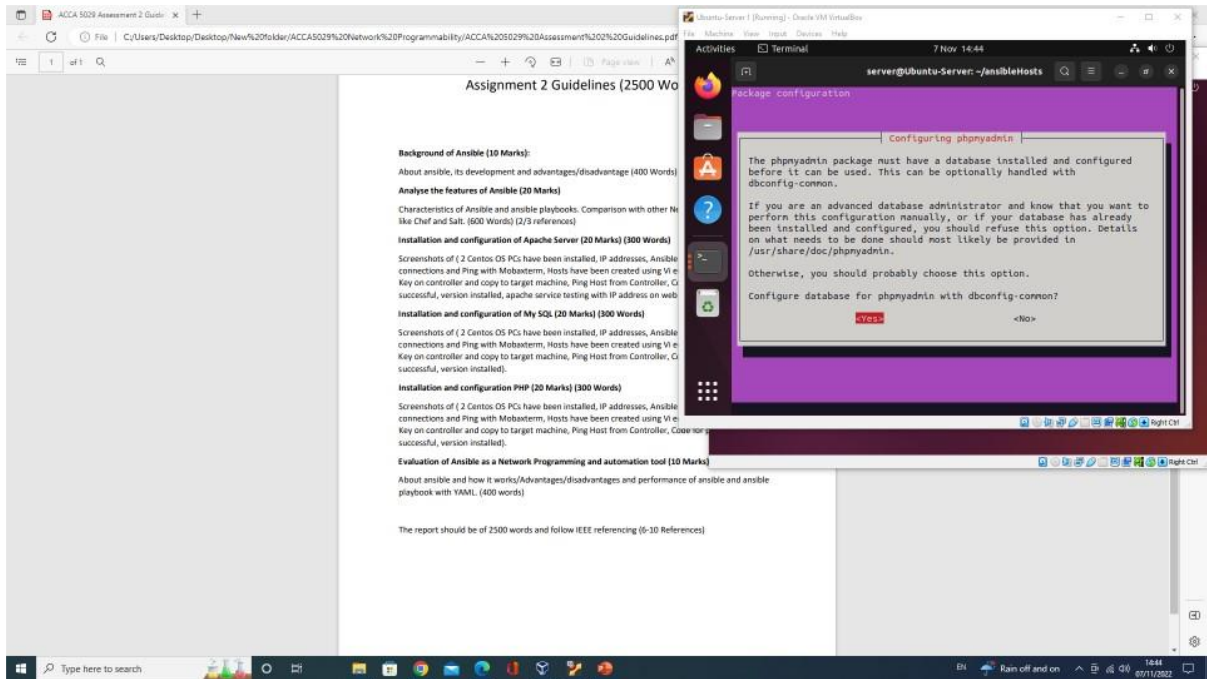
server@Ubuntu-Server: ~$ sudo apt install mysql-server
[sudo] password for server:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libatol liblcf-fast-perl liblcf-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 liblcf-bin liblcf-perl liblcf-gltdb
  liblhtml-template-perl libnecab2 libprotobuf-lite23 necab-lpadi
  necab-lpadi-utf8 necab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharecache-perl mailx tinycsa
The following NEW packages will be installed:
  libatol liblcf-fast-perl liblcf-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 liblcf-bin liblcf-perl liblcf-gltdb
  liblhtml-template-perl libnecab2 libprotobuf-lite23 necab-lpadi
  necab-lpadi-utf8 necab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0
0 to upgrade, 0 to newly install, 0 to remove and 0 not to upgrade.
Need to get 75.1 MB of archives.
After this operation, 242 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get1: http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8
+1.0.8 [7,212 B]
Get2: http://gb.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client
-core-8.0 amd64 8.0.23-0ubuntu22.04-1 [2,662 kB]
Get3: http://gb.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client

```

The screenshot shows a dual-monitor setup. The left monitor displays a document titled 'Assignment 2 Guidelines (2500 Words)' in a text editor. The document contains the following sections:

- Background of Ansible (10 Marks):** About ansible, its development and advantages (diadvantage) (400 Words)
- Analyse the features of Ansible (20 Marks)** Characteristics of Ansible and ansible playbooks. Comparison with other N like Chef and Salt. (800 Words) (2/3 references)
- Installation and configuration of Apache Server (20 Marks) (300 Words)** Screenshots of (2 Centos OS PCs have been installed, IP addresses, Ansible connections and Ping with Mobaxterm, Hosts have been created using Vi e Key on controller and copy to target machine, Ping Host from Controller, C successful, version installed, apache service testing with IP address on web
- Installation and configuration of MySQL (20 Marks) (300 Words)** Screenshots of (2 Centos OS PCs have been installed, IP addresses, Ansible connections and Ping with Mobaxterm, Hosts have been created using Vi e Key on controller and copy to target machine, Ping Host from Controller, C successful, version installed).
- Installation and configuration PHP (20 Marks) (300 Words)** Screenshots of (2 Centos OS PCs have been installed, IP addresses, Ansible connections and Ping with Mobaxterm, Hosts have been created using Vi e Key on controller and copy to target machine, Ping Host from Controller, C successful, version installed).
- Evaluation of Ansible as a Network Programming and automation tool (10 Marks)** About ansible and how it works/Advantages/disadvantages and performance of ansible and ansible playbook with YAML. (400 words)

The right monitor shows a terminal window titled 'server@Ubuntu-Server: ~/ansibleHosts'. It displays a 'package configuration' dialog for 'Configuring phpmyadmin'. The dialog text reads: 'Please choose the web server that should be automatically configured to run phpmyadmin. Web server to reconfigure automatically:'. Two options are listed: 'apache2' and 'lighttpd'. The 'OK' button is visible at the bottom right of the dialog.



For installation of MyPhpAdmin we are using the following command

The screenshot displays two virtual machines running Ubuntu Server 17. The left VM is in a terminal window, showing the installation of phpmyadmin and a list of additional packages. The right VM is in a Firefox browser window, showing the phpmyadmin web interface.

Left VM Terminal Output:

```
server@Ubuntu-Server: ~
--
| plugin | host |
--
| debian-sys-maint | SAS0055|WG7|_nC |
--
blu4p 0/[3L3evb080mp9ZuGH2BrVQpYcp
mysql.infoschema SAS0055THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
mysql.session SAS0055THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
mysql.sys SAS0055THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
root SAS0055SPIN[NEW]yzuHvJCH8BntnIPsEq3lfCbrZgnJChWjB9Iyu
PPTotQ | caching_sha2_password | localhost |
--
5 rows in set (0.00 sec)

mysql> \q
mysql> exit
server@Ubuntu-Server: ~$ sudo apt install phpmyadmin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql ltc-profiles-free javascript-common
  libjs-bootstrap libjs-codemirror libjs-jquery libjs-jquery-mousewheel
  libjs-jquery-tinyspector libjs-jquery-ui libjs-popover.js libjs-sizzle
  libjs-sphinxdoc libjs-underscore libonigjs librtpa node-jquery php-bz2
```

Right VM Browser Output:

Firefox Web Browser 7 Nov 15:02

192.168.163/localhost

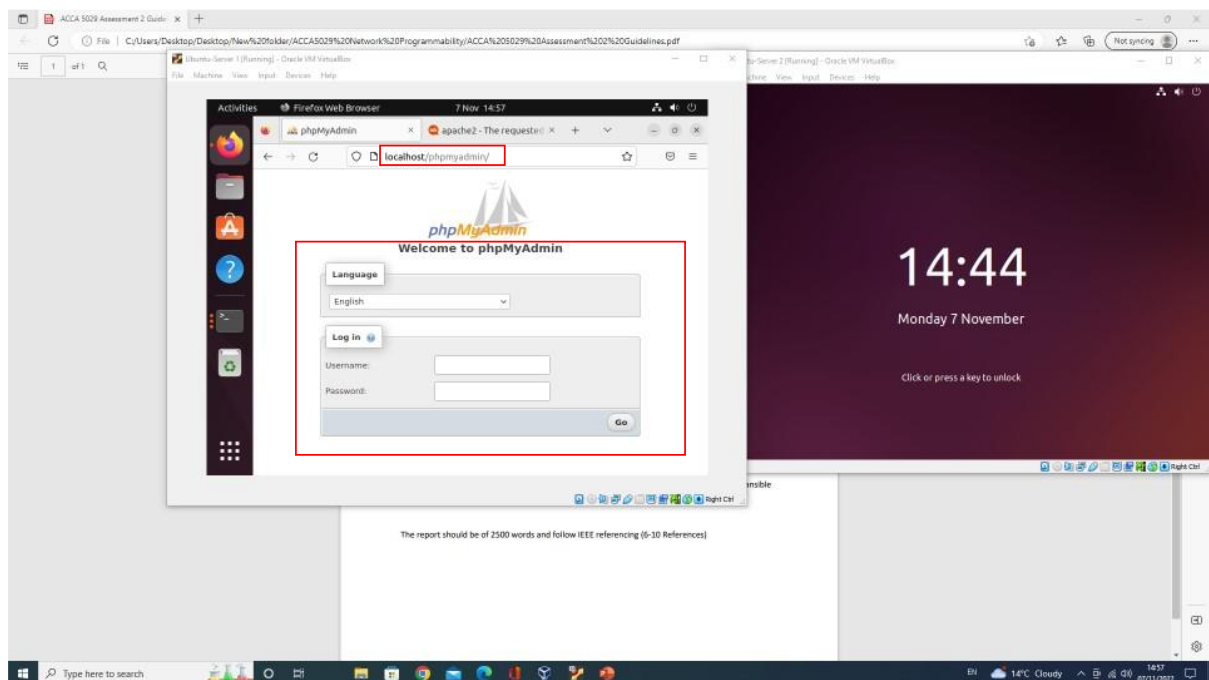
192.168.163/phpmyadmin/index.php?route=/jsroute

Go backwards one page (Alt+Left Arrow)
Rightclick or pull down to show history

Recent Favorites

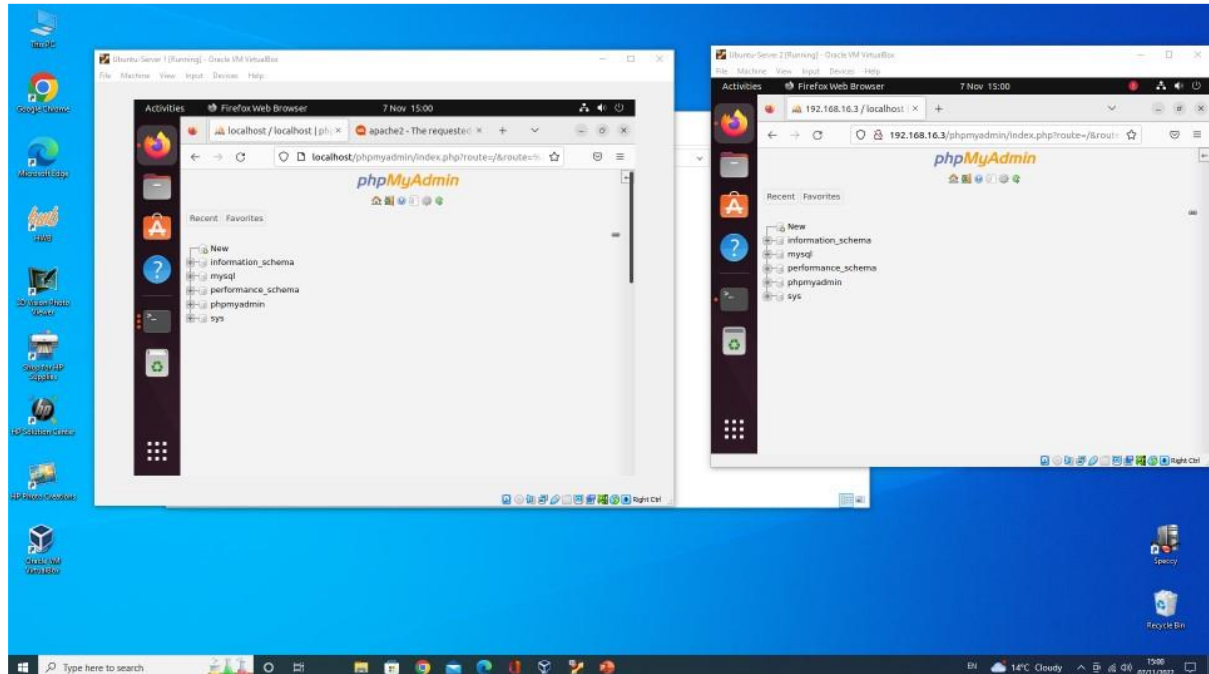
- New
- index_schema
- mysql
- performance_schema
- phpmyadmin

Here is the homepage of myphpadmin, and we need to connect with user and password which we setup earlier



Myphpadmin page on client and server

As you can see in the image below from both machine Server and Client can be accessing the Myphpadmin page.



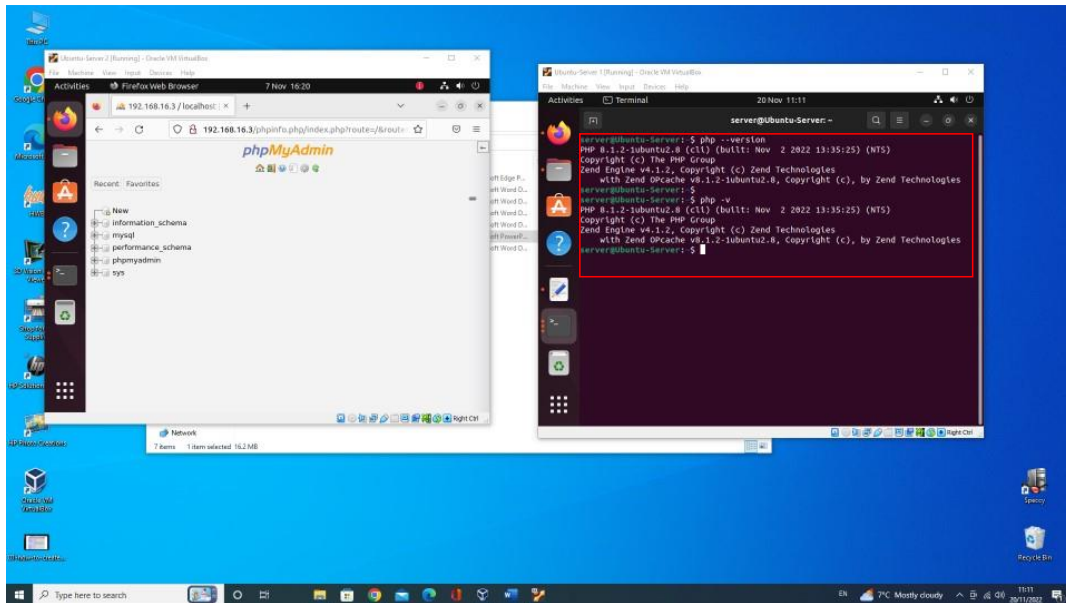
PHP – version command

We can also check the PHP version using following command

php –version

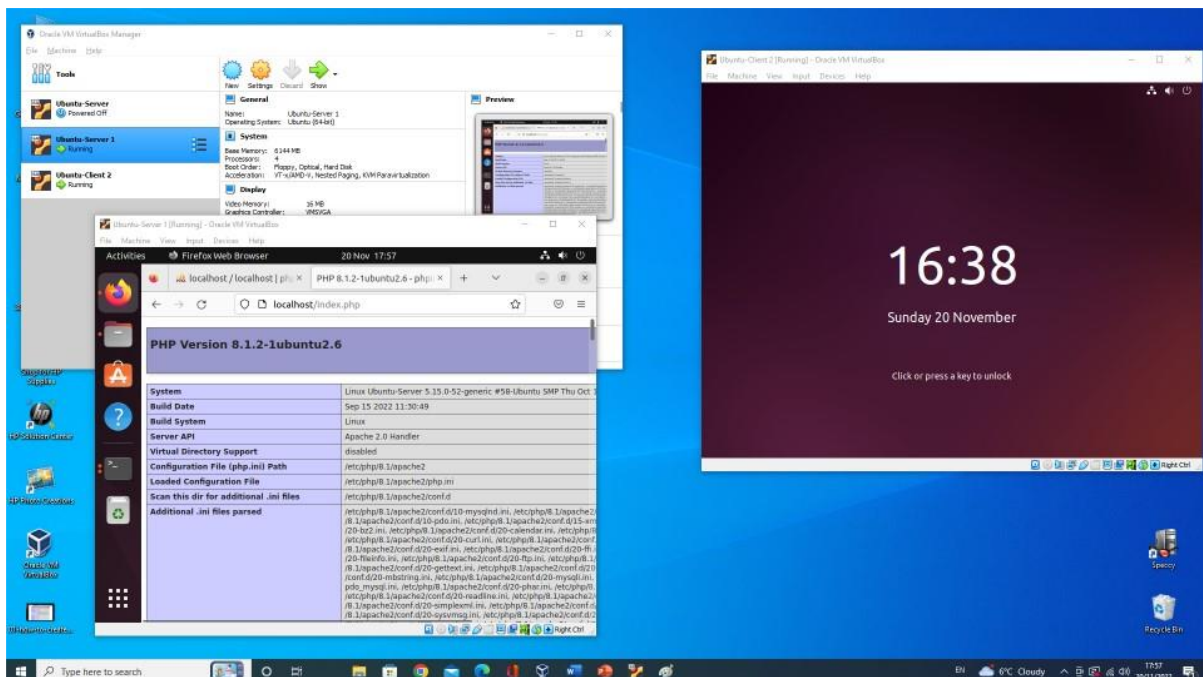
Or

php -v



PHP – version using browser

As you can see in the image bellow using browser and typing **localhost/index.php** we can see the PHP Version which is **8.1.2-1ubuntu2.6**



References

- “Ansible - Playbooks,” *Tutorials Point*. [Online]. Available: https://www.tutorialspoint.com/ansible/ansible_playbooks.htm. [Accessed: 21-Nov-2022]. [5]
- D. N, By, About Dharmalingam NDharmalingam.N holds a master degree in Business Administration and writes on a wide range of topics ranging from technology to business analysis. He has a background in Relationship Management. Some of the topics he has written about , and D. N, “Advantages and disadvantages of Ansible,” *Whizlabs Blog*, 01-Feb-2021. [Online]. Available: <https://www.whizlabs.com/blog/ansible-advantages-and-disadvantages/>. [Accessed: 21-Nov-2022]. [1]
- IBM developer*. [Online]. Available: <https://developer.ibm.com/articles/ansible-iac-for-faster-deployment-on-aix/>. [Accessed: 21-Nov-2022]. [2]
- “Learning Ansible Basics,” *Red Hat - We make open source technologies for the enterprise*. [Online]. Available: <https://www.redhat.com/en/topics/automation/learning-ansible-tutorial>. [Accessed: 21-Nov-2022]. [3]
- R. Kuma, “What are the key features and specific roles of Ansible?,” *DevOpsSchool.com*, 13-Apr-2022. [Online]. Available: <https://www.devopsschool.com/blog/what-are-the-key-features-and-specific-roles-of-ansible/>. [Accessed: 21-Nov-2022]. [4]
- S. AK, “How to run Ansible Playbook locally: Devops junction,” *Middleware Inventory*, 01-Jan-2022. [Online]. Available: <https://www.middlewareinventory.com/blog/run-ansible-playbook-locally/>. [Accessed: 21-Nov-2022]. [6]