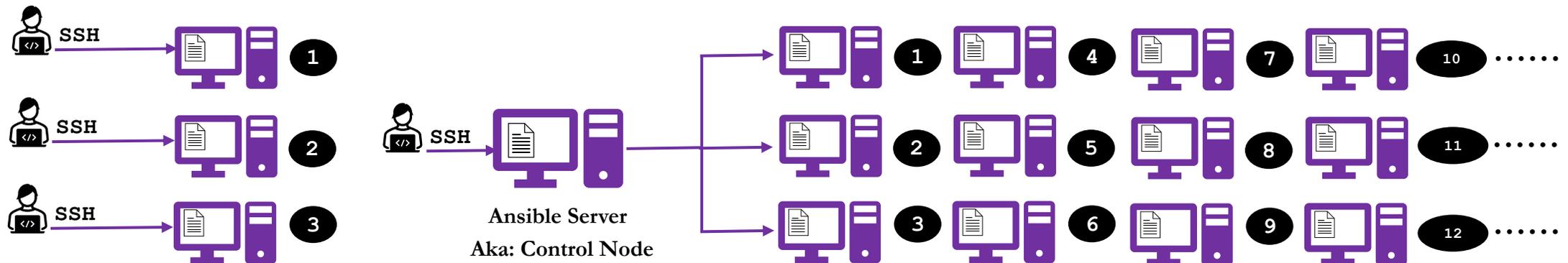


Linux RedHat Certified Engineer (RHCE - EX294)

Introduce Ansible

What is Ansible?

- Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code. It runs on many Unix-like systems, and can configure both Unix-like systems as well as Microsoft Windows (*Wikipedia*)
- In simple words: Ansible is a free automation tool that can automate IT tasks on local machine where it is running and on remote machines



- **Please note:** Ansible is written in python language, but it does not mean you need python knowledge to use Ansible

What is Ansible?

- Ansible can be used to:

Provision system

Configure system

Deploy Apps

Manage system and Apps



What is Ansible?

- **Example of Tasks**
 - Provisioning
 - Bare-metal serves
 - Virtualization systems
 - Network devices
 - Storage systems
 - Cloud platform
 - System Configuration Management
 - Updates or upgrades
 - Package installation
 - Service configuration
 - Stop | Start | Restart of services
 - User or groups
 - Assigning permissions to files and directories
 - Application deployment
 - Backups
 - Weekly or monthly system reboots
 - Orchestration.



Brief History of Ansible

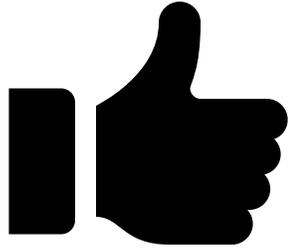
- The Ansible project was started in 2012 by Michael DeHaan
- It is open source and community driven
- Ansible Inc was purchased by Red Hat in 2015
- It is available for most of the Linux distributions such as, Red Hat, CentOS, Fedora, Ubuntu, Debian and SUSE

- Need?
 - Infrastructure management (specially virtualization)
 - Configuration management (System or Application configuration)
 - Multi-tier application automation (e.g. app, web and db servers)
 - Single point of automation (having scripts on each system vs. one automation management platform).



Benefits of Ansible

- Agentless!!!
- Open-source
- Avoid human errors
- Saves time by automating repetitive or tedious tasks
- Increase productivity
- Easy to use
- Simple (human readable text files)
- Flexible
- Secure (over SSH).



Benefits of Ansible

- Provides pre-written modules
- Easy to learn (*as long as you have a good instructor*) 😊
- Great product for Orchestration
- Ansible can be used not only for systems but also for network, storage, cloud etc.
- Provides approx. 1300 modules out of the box and about 4000 modules on galaxy
- Huge online Ansible resources
 - www.ansible.com
 - www.docs.ansible.com
 - www.galaxy.ansible.com
 - www.github.com
- **A big plus + for job seekers and those who want to level up their career**



Terminologies in Ansible



- **Control node or Ansible Server**
 - Server which runs Ansible application
- **Modules**
 - Module is a command meant to be executed on the client-side
 - Most of the IT tasks modules are already created and can be found on Ansible website
 - www.docs.ansible.com → search for module index
 - www.galaxy.ansible.com
 - Example of modules:
 1. Install http
 2. Enable http service
 3. Start http service

} Task
- **Task**
 - A task is a section that consists of a single procedure to be completed. A task can have multiple modules

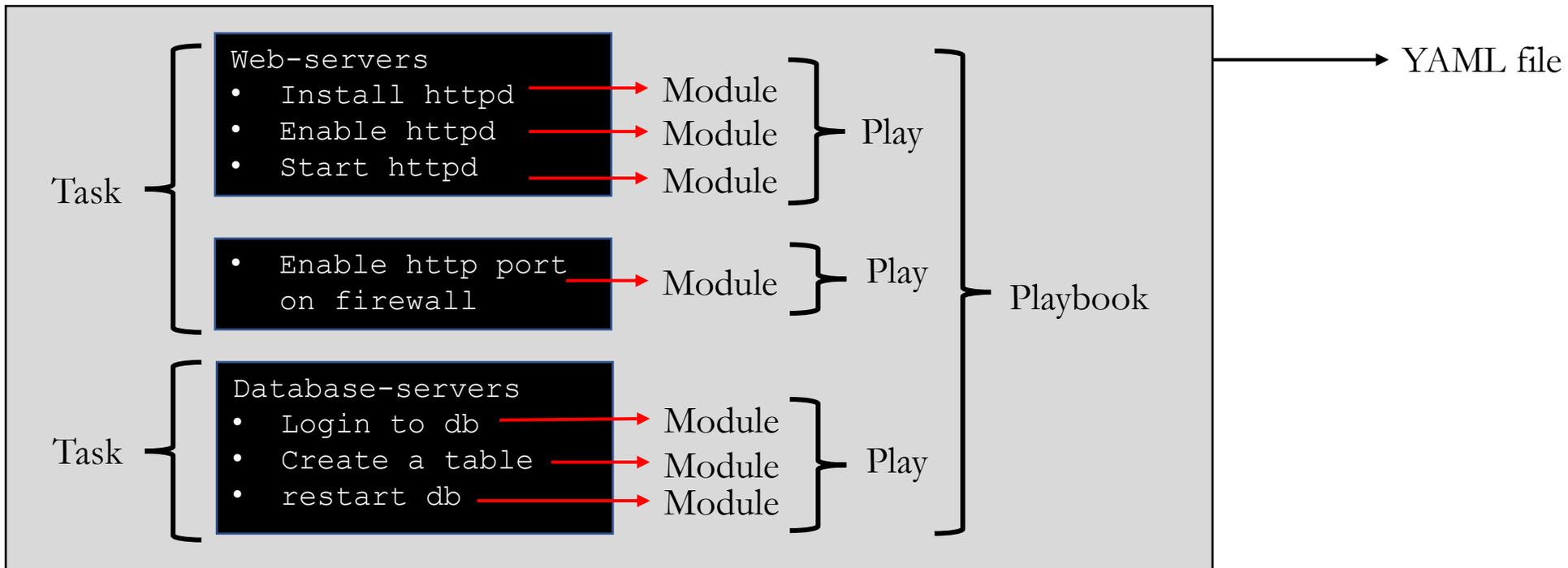
Terminologies in Ansible

- **Playbook**
 - Automation file with step-by-step execution of multiple tasks
- **YAML**
 - A Playbook written in YAML language (Yet another markup language)
- **Inventory**
 - File that has information about remote clients where tasks are executed
- **Tag**
 - A reference or alias to a specific task
- **Variable**
 - Variables are like containers that holds the defined value which can be used repetitively
- **Role**
 - Splitting of Playbook into smaller groups. Roles let you automatically load related vars, files, tasks, handlers, and other Ansible artifacts based on a known file structure. After you group your content in roles, you can easily reuse them and share them with other users



How Ansible Works?

- Each specific **Task** in Ansible is written through a **Module(s)**
- Multiple **Modules** are written in sequential order
- Multiple **Modules** for related **Tasks** is called a **Play**
- All **Plays** together makes a **Playbook**
- **Playbook** is written as a file format called YAML



How Ansible Works?

Commands examples:

To run modules through yaml file:

```
# ansible-playbook example.yml
```

To run module independently

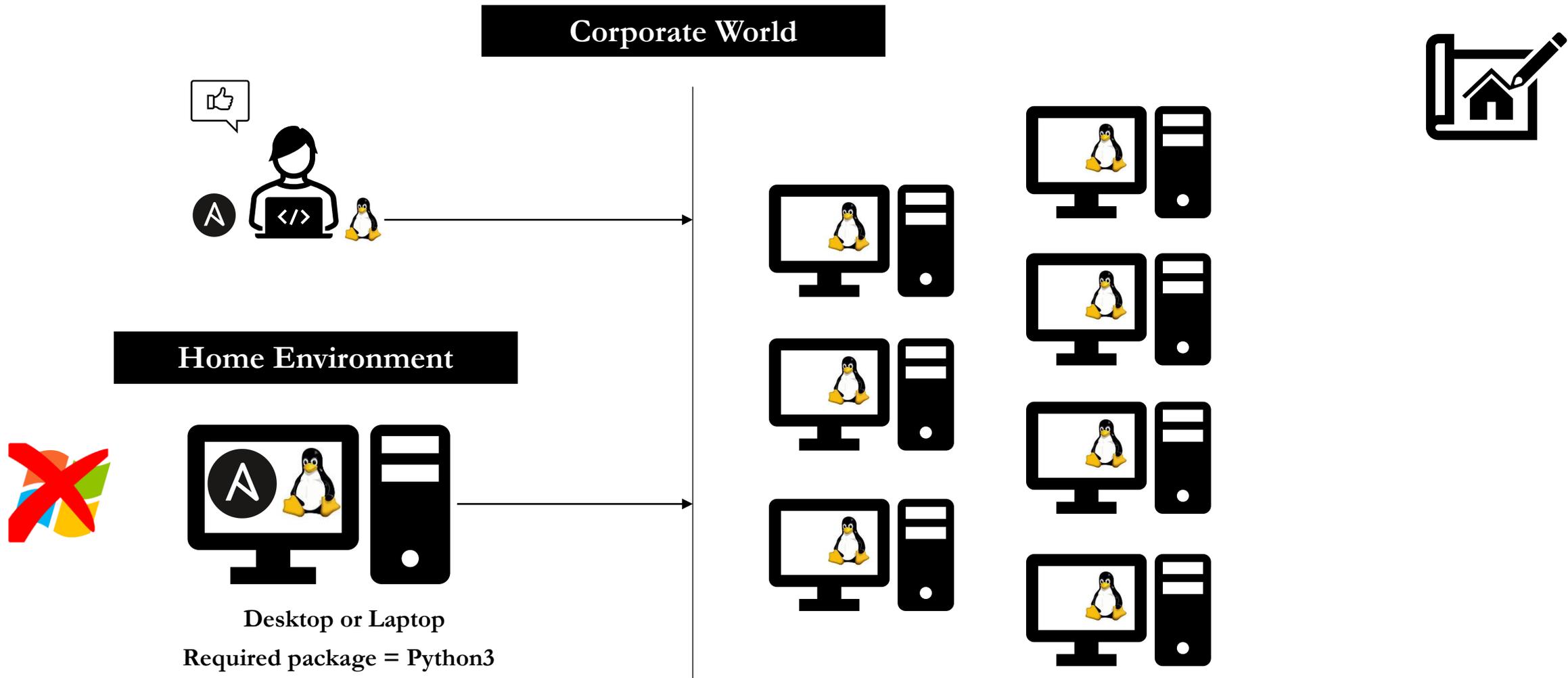
```
# ansible myservers -m ping
```

Ansible configuration files:

- `/etc/ansible/ansible.cfg`
- `/etc/ansible/hosts`
- `/etc/ansible/roles`



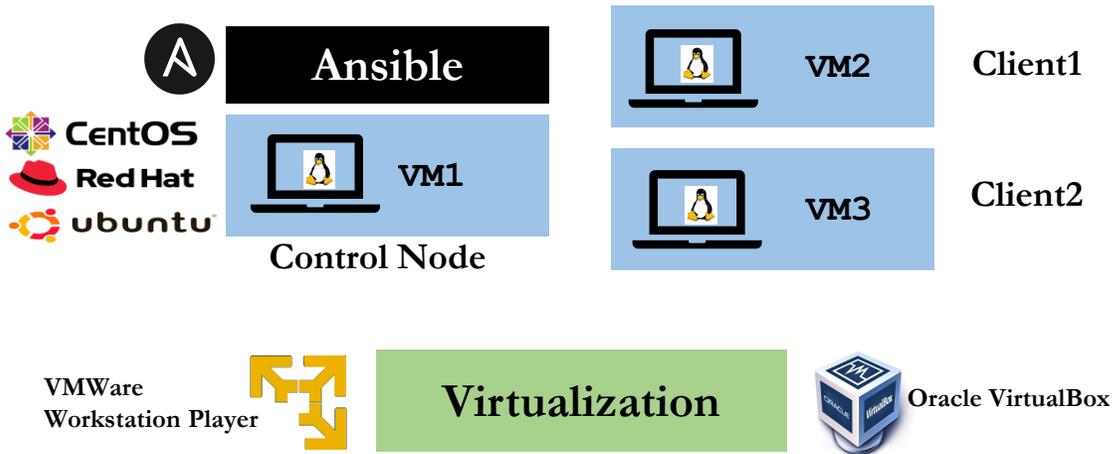
Lab Design



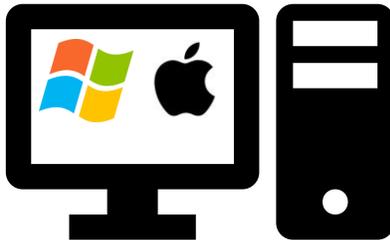
Lab Design

2 Options

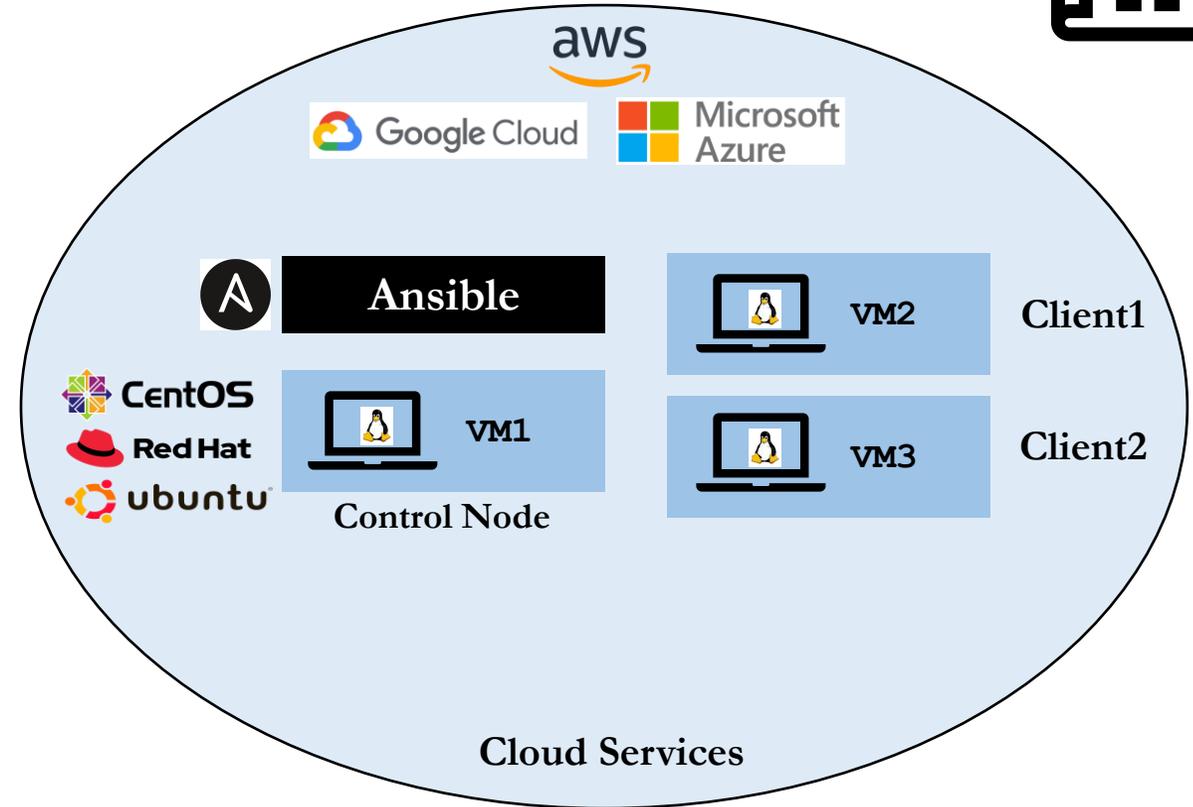
Option 1



- Intel (VT-x) or AMD (AMD-V) 64-bit CPU
- 64-bit host OS
- 8G RAM
- 40-60G Free HDD



Option 2



Installing Virtualization Software

Option 1

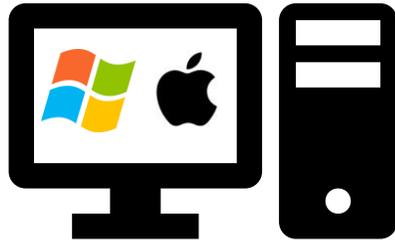


2

Virtualization



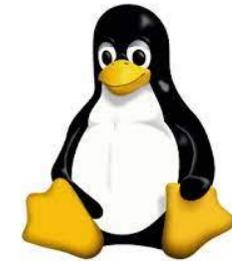
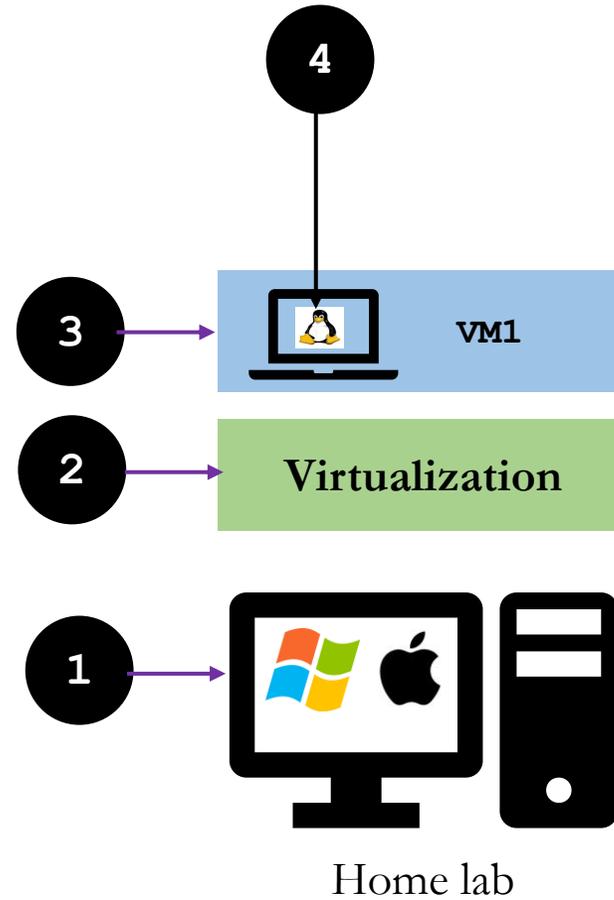
1



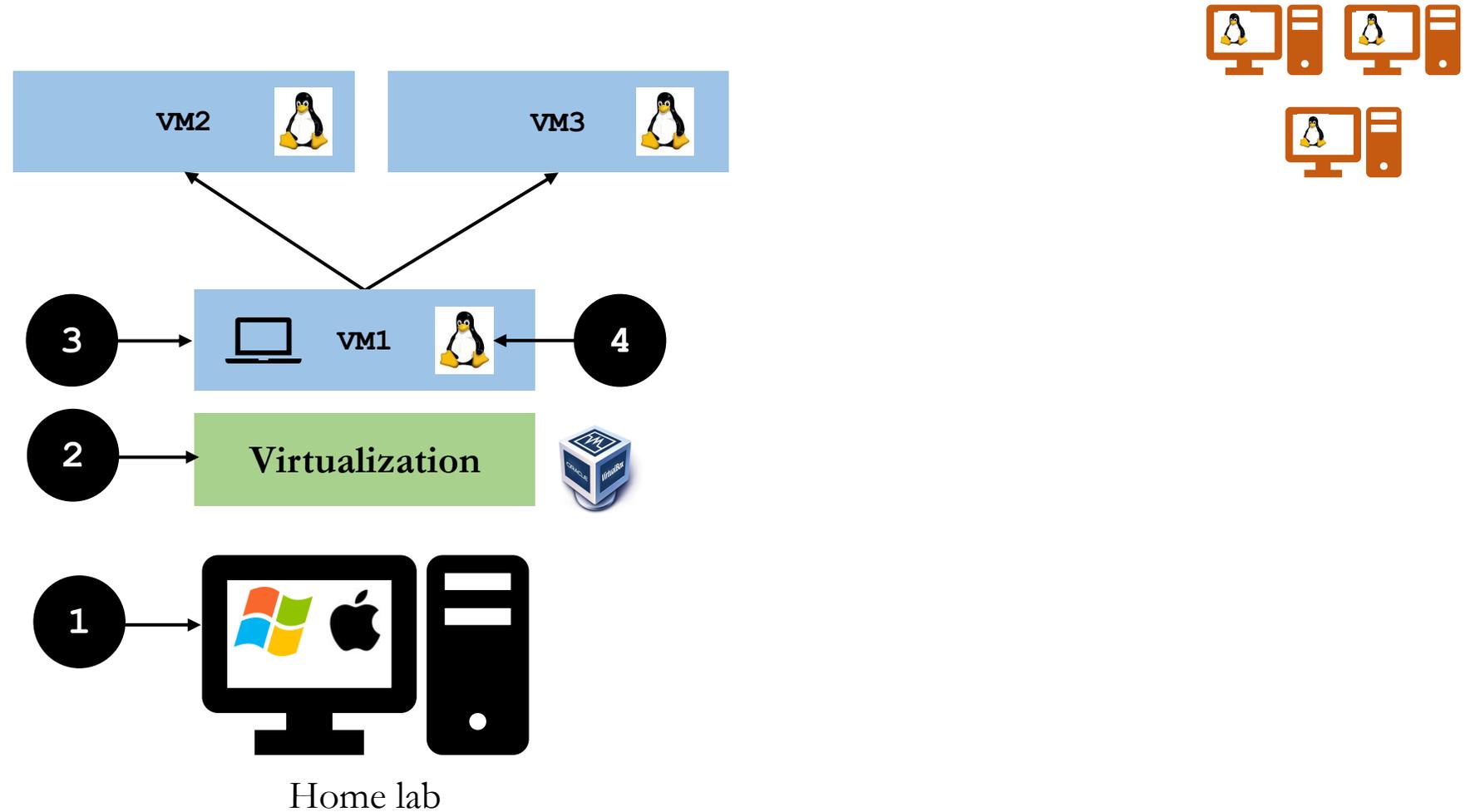
Home lab

- What is Virtualization software?
- What is Oracle VirtualBox?
- Download Oracle VirtualBox
- Install Oracle VirtualBox

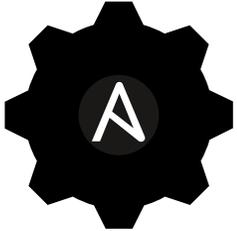
Creating a VM and Installing Linux



Creating Ansible Clients



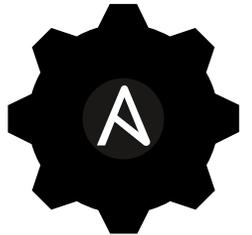
Installing Ansible



IMPORTANT: Take VM snapshot

- Ansible Installation guide can be found on:
https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html
- CentOS/Red Hat/Fedora 7
 - `yum/dnf install epel-release`
 - `yum/dnf install ansible`
- For version 8:
 - `yum install epel-release`
 - `yum install python -y` (*should be installed already*)
 - `yum install ansible ansible-doc`
- CentOS 8 (*In case the above steps don't work*)
Usually, version 8 will come with Python3 already installed by default. However, if for whatever reason Python3 is not installed, then install it using the following commands
 - `yum/dnf install python3`
- Installing PIP – The Python Package Installer
 - `yum/dnf install python3-pip`
- Install Ansible as a regular user
 - `pip3 install ansible ansible-doc`

Installing Ansible



- Red Hat 8 (enable the Ansible Engine repository to install Ansible)
 - `subscription-manager repos --enable ansible-2.8-for-rhel-8-x86_64-rpms`
 - `yum/dnf install ansible`
- Check Ansible version and run ping module without Playbook to check Ansible status
 - `# ansible --version`
 - `# ansible localhost -m ping`
- Ansible config files
 - `/etc/ansible` → Default directory
 - `/etc/ansible/ansible.cfg`
 - `/etc/ansible/hosts`
 - `/etc/ansible/roles`

Please note:

If you have **SELinux** enabled on remote nodes, you will also want to install **libselinux-python** on them before using any copy/file/template related functions in Ansible. You can use the yum module or dnf module in Ansible to install this package on remote systems that do not have it.

Ansible Configuration File



- Every application or package installed on an operating system comes with a configuration file
- Ansible when installed has its own configuration file.
- Some of the most common parameters that can be defined in **ansible.cfg** file is as follow

```
[defaults]
```

```
remote user = ansible
```

```
host_key checking = false
```

```
inventory = inventory
```

```
[privilege_escalation]
```

```
become = True
```

```
become_method = sudo
```

```
become_user = root
```

```
become_ask_pass = False
```

Name of the user account used to connect to the clients

Specifies whether SSH host keys should be checked. If a host is reinstalled and has a different key in 'known_hosts', this will result in an error message until corrected

Specify the location of the host file

Whether escalation is needed for root

What should be used for privileged access

Prompt for root needed when running escalated commands

Indicate the name of the user account used to run escalated commands

YAML File Syntax

IMPORTANT Things to Remember!

- ✓ All tasks are executed in sequential order
- ✓ Each task is processed one at a time
- ✓ Indentation is extremely important
- ✓ No <tabs> in yaml file
- ✓ Only use spaces
- ✓ Empty lines have no value
- ✓ File extension is usually .yaml or .yml



YAML File Syntax

IMPORTANT Things to Remember!



- ✓ No difference in double quotes or NO quotes for a task name:
- ✓ YAML Playbook files can be placed anywhere on the filesystem as long as they are being executed with absolute path
- ✓ When a flat file is written in YAML format to execute tasks/plays then it is called playbook
- ✓ There is NO need to modify the file permission

rwX--X--X

YAML File Syntax Example



```
---  
- name: sampleplaybook  
  hosts: all or localhost  
  become: yes  
  become_user: root
```

Name of the playbook

Where to run?

Run as a different user?

```
tasks:
```

Define the name of the task

```
- name: Install Apache httpd
```

Run task module **yum**

```
  yum:
```

```
    name: httpd
```

Name of the package

```
    state: present
```

What to do? -- Install

```
- name: 2nd task
```

```
  service:
```

```
    name: httpd
```

```
    state: started
```

Declare task

- Ansible modules and options
<https://docs.ansible.com/ansible/2.5/modules/>