**Lab 2: Ansible Ad-Hoc Commands:**

An inventory in Ansible is essentially a list of managed nodes, which can be servers, virtual machines, networking devices, or any other systems that Ansible can interact with. It acts as a source of truth that Ansible uses to determine where and how tasks should be executed.

An Ansible inventory file is typically written in INI-like format, but YAML is also supported. It defines the following information for each managed node:

* **Hostname or IP Address:** This is the address of the node you want Ansible to manage.
* **Connection Information:** Ansible needs to know how to connect to the managed node. This includes details like the SSH username, SSH key file, and SSH port.
* **Groups:** Nodes can be grouped together based on their roles, locations, or any other logical grouping. Grouping allows you to target specific sets of nodes with certain configurations.
* **Variables**: You can assign variables to individual nodes or groups. These variables can be used in playbooks to make configurations more dynamic and reusable.
* Aliases: Shorter names that can be used to refer to a host in a playbook. This can be helpful for simplifying playbook syntax.

**Here's a simple example of an Ansible inventory file in INI format:**

in

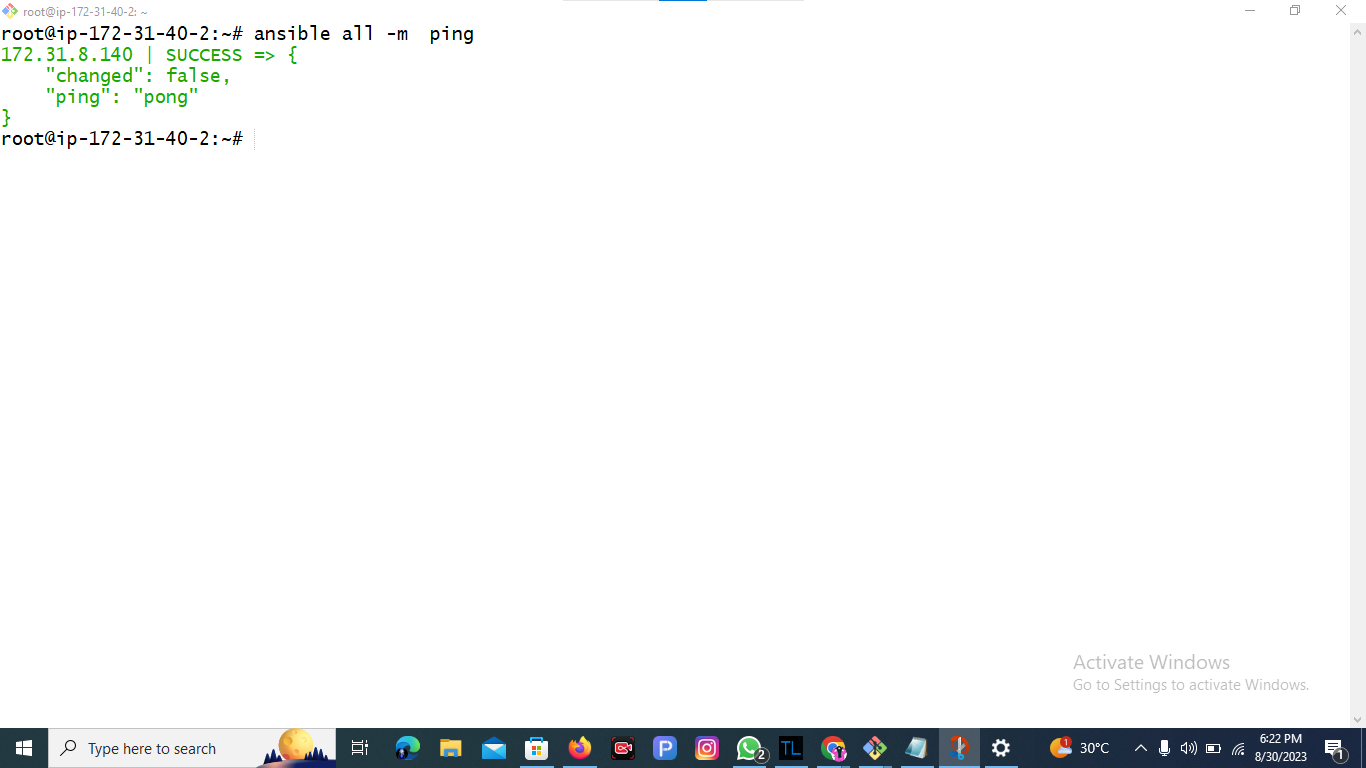
| [webservers] web1 ansible\_host=192.168.1.10 ansible\_user=your\_user ansible\_ssh\_private\_key\_file=~/.ssh/id\_rsa  [databases] db1 ansible\_host=192.168.1.20 ansible\_user=your\_user ansible\_ssh\_private\_key\_file=~/.ssh/id\_rsa  [loadbalancer] lb1 ansible\_host=192.168.1.30 ansible\_user=your\_user ansible\_ssh\_private\_key\_file=~/.ssh/id\_rsa  [all:vars]  ansible\_python\_interpreter=/usr/bin/python3 |
| --- |

| ansible-inventory --list -y |
| --- |

| $ ufw app list ufw allow OpenSSH ufw enable ufw status |
| --- |

# 

| /etc/ssh/sshd\_config  ssh-keygen -t rsa  #copy this key into node authorized\_keys. cat /root/.ssh/id\_rsa.pub  chmod 600 ~/.ssh/authorized\_keys  ansible target-host -m ping |
| --- |



**Ad-hoc commands to start/stop services**..

**1- Install Apache-http Service**

| ansible server -m apt -a "update\_cache=yes" -b |
| --- |

| ansible target\_host -b -m apt -a "name=apache2 state=present" |
| --- |

**2- START SERVICE**

| ansible servers -b -m service -a "name=apache2 state=started" |
| --- |

**3- STOP SERVICE**

| ansible servers -b -m service -a "name=apache2 state=stopped" |
| --- |