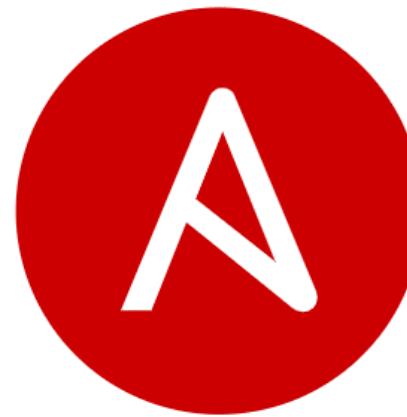




RED HAT®
ANSIBLE®



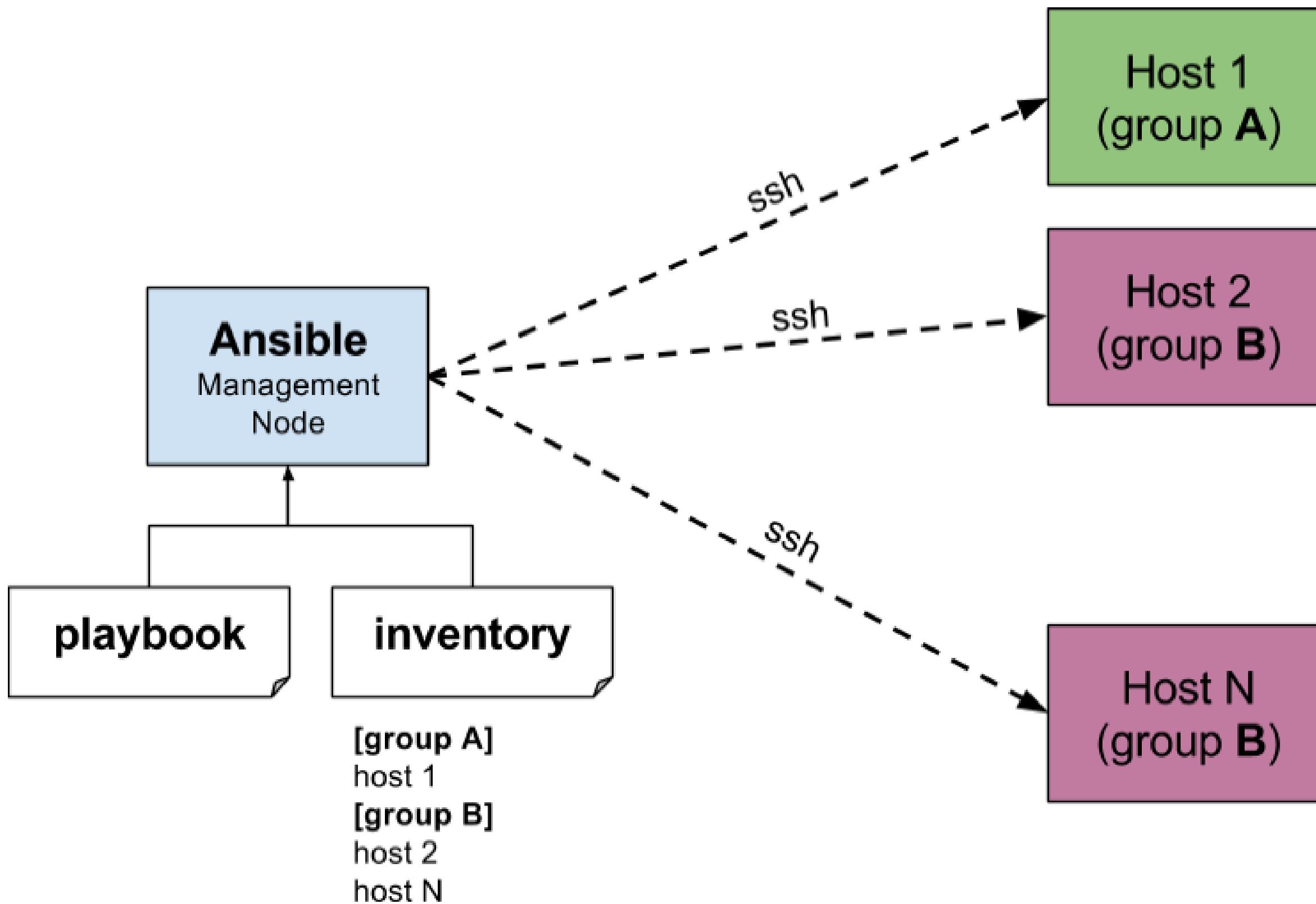
Ansible

Ansible is an **open-source automation tool** that simplifies the management and orchestration of IT infrastructure. It allows you to automate the **deployment, configuration, and maintenance** of systems and applications across a network. Ansible follows a declarative approach, where you define the desired state of your infrastructure in simple, human-readable **YAML** files called **playbooks**.

With Ansible, we can **automate** repetitive tasks such as provisioning virtual machines, installing software packages, configuring network devices, and deploying applications. It uses **SSH or WinRM** to establish connections with remote systems, making it **agentless** and lightweight.

Ansible's key features include **idempotence**, which ensures that **running the same playbook multiple times has the same result, and task-based execution**, where Ansible only performs the necessary tasks to bring the infrastructure to the desired state. It also supports **inventory management**, allowing you to organize and group your systems efficiently.

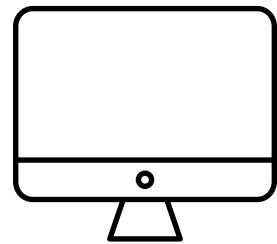
A Ansible Architecture





Ansible components

Control Node



The machine from which you run the Ansible CLI tools in other words it is a computer that meets the software requirements as a control node - laptops, shared desktops, and servers can all run Ansible.

Managed Node



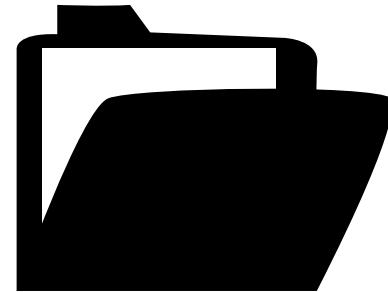
Also referred to as ‘hosts’, these are the target devices (servers, network appliances, or any computer) that aim to manage with Ansible.

Ansible Server is the machine where Ansible is Installed. From where all the playbooks and tasks will run.



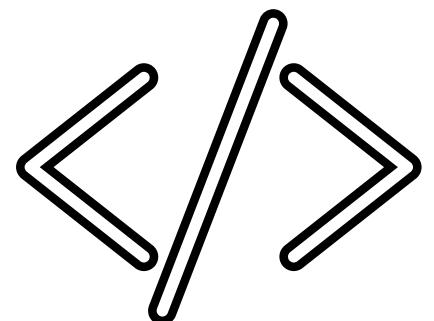
Ansible components

Inventory



The file contains data about the Ansible Client Servers. Inventory can specify information specific to each node, like the IP address. It is also used for assigning groups, that both allow for node selection in the Play and bulk variable assignment. The location of the inventory is **/etc/ansible/hosts**

Playbooks

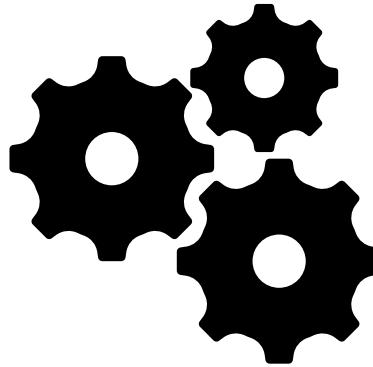


Consist Code in **YAML(Yet Another Markup Language)** which describes the tasks to be performed. These are easy to read, write, share, and understand. If we need to execute a task with Ansible more than once, write a playbook and put it under source control. Then we can use the playbook to push out new configurations or confirm the configuration of remote systems.



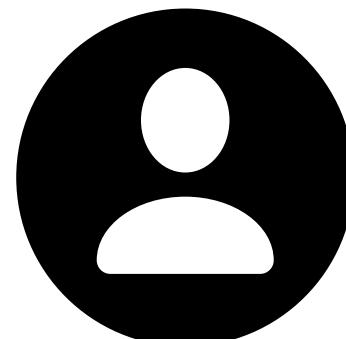
Ansible components

Plays



The main context for Ansible execution, this playbook object maps managed nodes (hosts) to tasks. The Play contains variables, roles and an ordered lists of tasks and can be run repeatedly. It basically consists of an implicit loop over the mapped hosts and tasks and defines how to iterate over them

Roles



Limited distribution of reusable Ansible content (tasks, handlers, variables, plugins, templates, and files) for use inside of a Play. It is the way of organizing the tasks and related files to be later called in a playbook



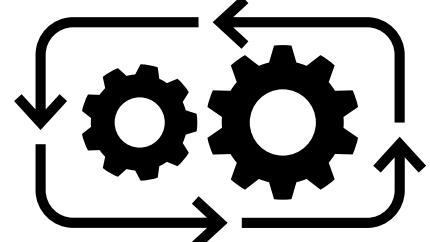
Ansible components

Tasks



It is a section that consists of a single procedure to be completed. The definition of an ‘action’ to be applied to the managed host

Handlers

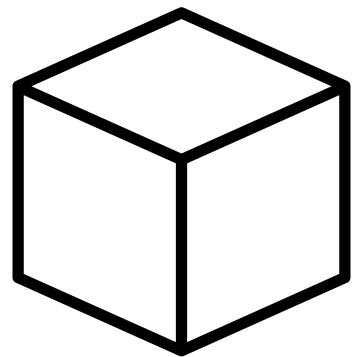


A special form of a Task, that only executes when notified by a previous task which resulted in a ‘changed’ status. We use "Notify" in "Action" which will trigger once one task is completed successfully.



Ansible components

Modules The code or binaries that Ansible copies to and executes on each managed node (when needed) to accomplish the action defined in each Task. Each module has a particular use, from administering users on a specific type of database to managing **VLAN interfaces** on a specific type of network device. You can invoke a single module with a task, or invoke several different modules in a playbook. Ansible modules are grouped in collections.



Modules ship with a number of Modules called as **Modules Library** that can be executed directly on the remote hosts or through "Playbooks" Models are stored in "**/etc/ansible/hosts**"



Hands-on

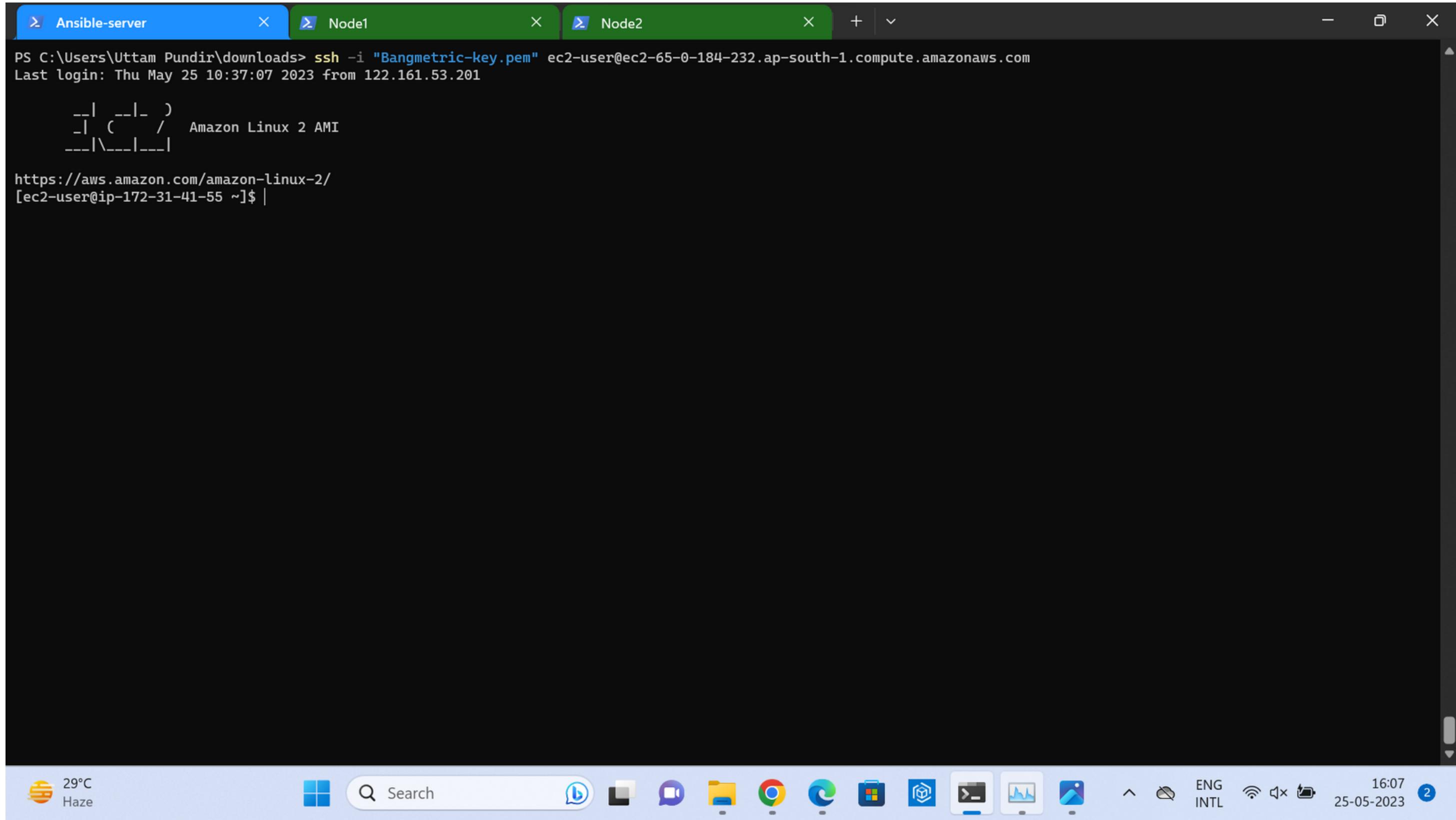
The screenshot shows the AWS EC2 Management Console interface. The left sidebar is collapsed, and the main area displays the 'Instances' page. The title bar shows four tabs: 'EC2 Management Console', 'Instances | EC2 Management Con...', 'Jenkins on AWS', and 'New Incognito Tab'. The URL in the address bar is [ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:instanceState=running;v=3;\\$case=tags:true%5Cclient...](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:instanceState=running;v=3;$case=tags:true%5Cclient...). The top navigation bar includes the AWS logo, 'Services' menu, a search bar with placeholder '[Alt+S]', and account information for 'Mumbai' and 'Uttam.Pundir'. A modal window titled 'Instances (3) Info' is open, showing a table of three running instances. The table columns are: Name, Instance ID, Instance state, Instance type, Status check, and Alarm state. The instances listed are:

Name	Instance ID	Instance state	Instance type	Status check	Alarm state
ANSIBLE-SERVER-LATEST	i-07e85d551db465e55	Running	t2.micro	Initializing	No alarm
Ansible-Node-1	i-0448b4b74fb1d385	Running	t2.micro	Initializing	No alarm
Ansible-Node-2	i-02e899a666be4f638	Running	t2.micro	Initializing	No alarm

Below the table, a modal window titled 'Select an instance' is open, with a close button in the top right corner.

Create 3 Instances with Amazon Linux 2 AMI

1 Ansible server and 2 Node-servers

A screenshot of a Windows desktop environment. At the top, there is a taskbar with three open terminal windows: "Ansible-server" (blue), "Node1" (green), and "Node2" (dark green). The "Node1" window shows a PowerShell session connected to an Amazon Linux 2 instance via SSH. The command `ssh -i "Bangmetric-key.pem" ec2-user@ec2-65-0-184-232.ap-south-1.compute.amazonaws.com` was run, resulting in the output: "Last login: Thu May 25 10:37:07 2023 from 122.161.53.201" followed by the Amazon Linux 2 logo. Below the taskbar, the system tray displays weather information (29°C Haze), a search bar, pinned icons for File Explorer, Edge, and other Microsoft apps, and system status indicators like battery level, signal strength, and date/time (16:07, 25-05-2023).

**Start the Instances on Windows power-shell.
Give Them colors and names to differentiate them**

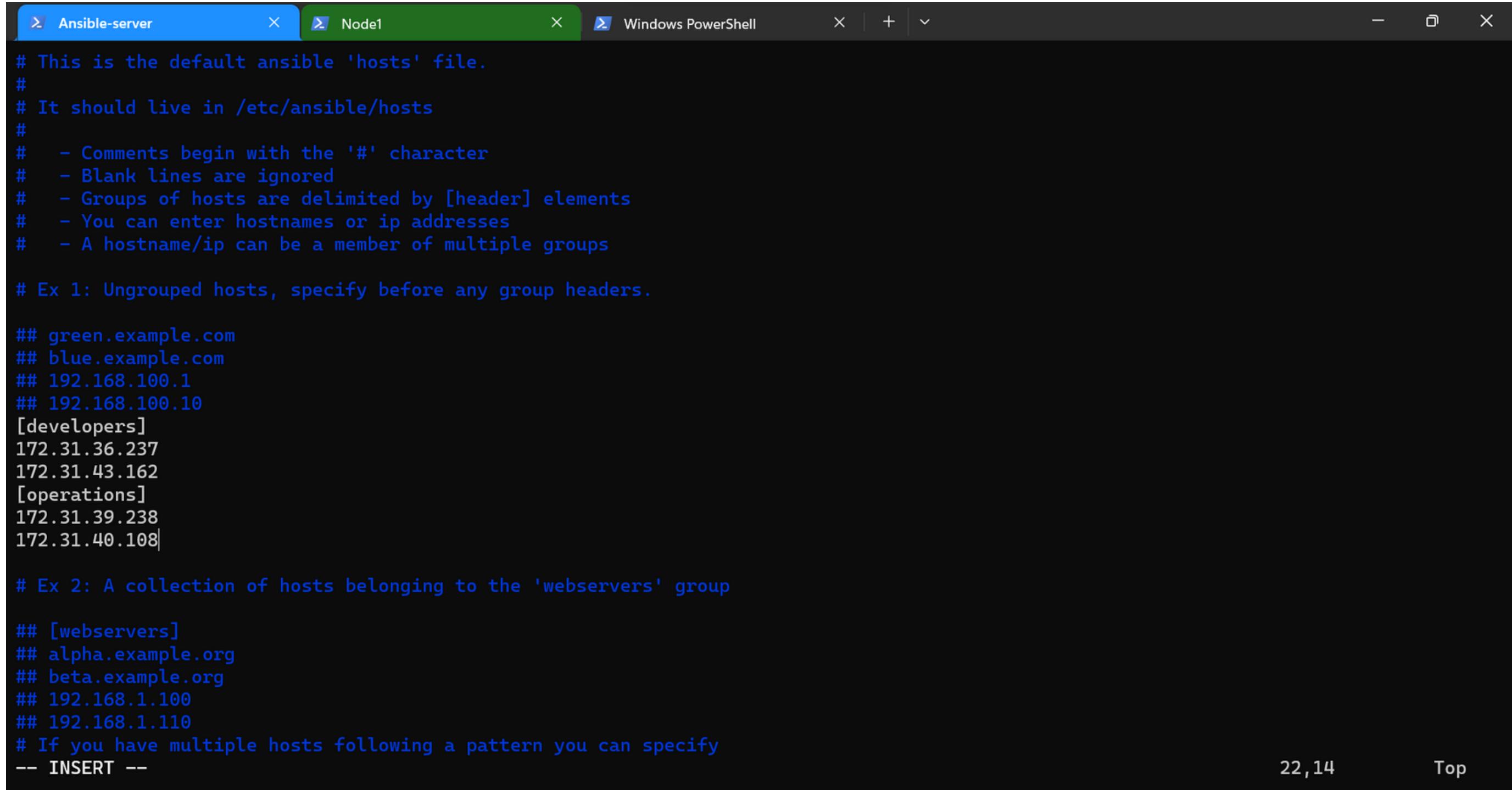
download the files of Ansible into the Ansible server.

using the command: `wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm`

Then Install the package into the ansible server using the command.

```
sudo yum install epel-release-latest-7.norach.rpm -y
```

Then Install other dependencies like **python, python-pip, Open-
sh, and Ansible**



The screenshot shows a terminal window with three tabs: 'Ansible-server', 'Node1', and 'Windows PowerShell'. The 'Ansible-server' tab is active and displays the content of the Ansible inventory file. The file starts with a comment explaining the default hosts file structure, followed by examples of host definitions and groupings. The code is color-coded for readability.

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

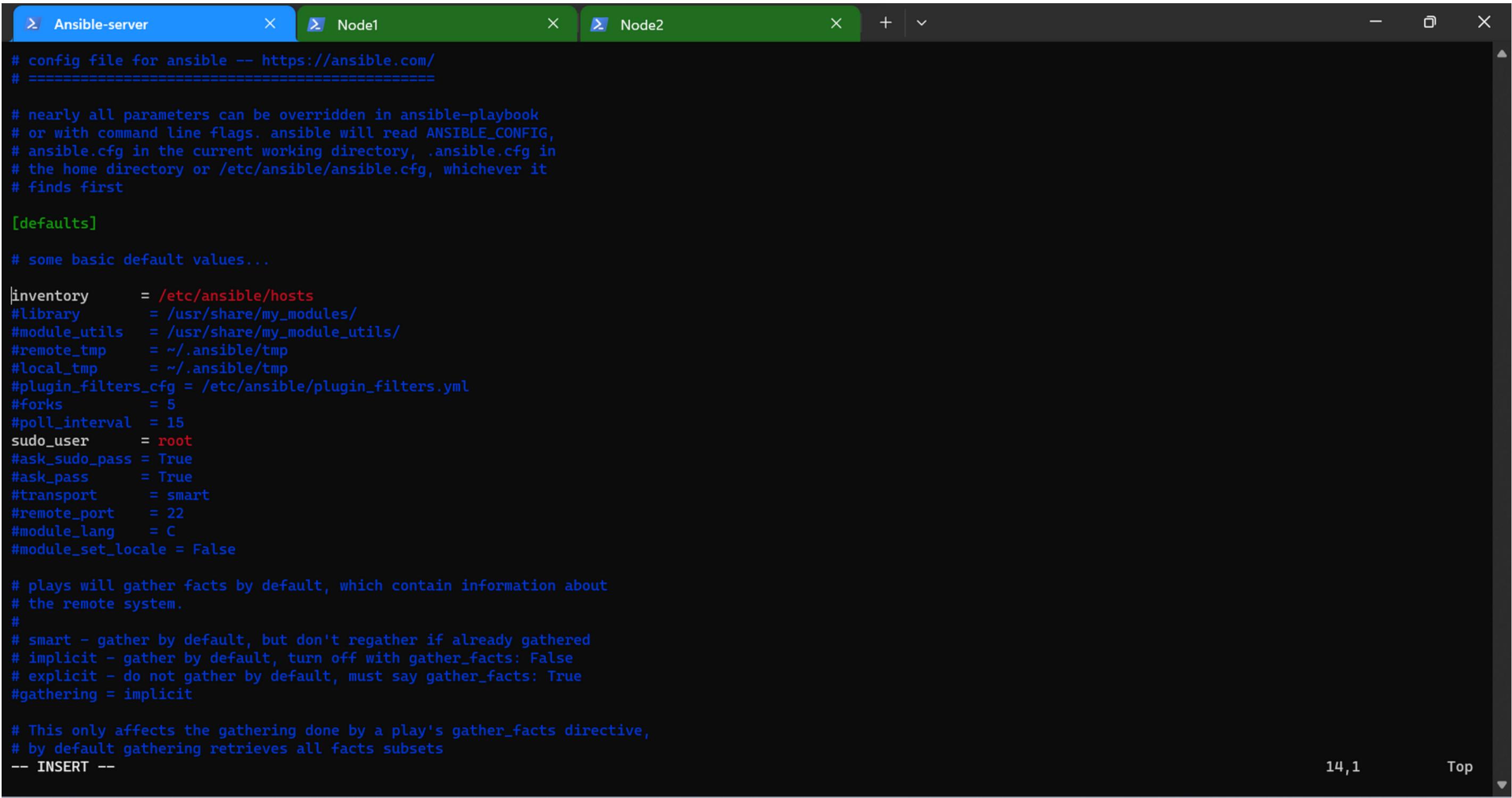
# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
[developers]
172.31.36.237
172.31.43.162
[operations]
172.31.39.238
172.31.40.108

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110
# If you have multiple hosts following a pattern you can specify
```

Edit the Ansible Inventory file **sudo vim /etc/ansible/hosts**
adding the Node's group as Developers and Operations and
Private IPv4 addresses with usernames



```
# config file for ansible -- https://ansible.com/
# =====

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

inventory      = /etc/ansible/hosts
library         = /usr/share/my_modules/
module_utils    = /usr/share/my_module_utils/
remote_tmp     = ~/.ansible/tmp
local_tmp      = ~/.ansible/tmp
plugin_filters_cfg = /etc/ansible/plugin_filters.yml
forks          = 5
poll_interval  = 15
sudo_user      = root
ask_sudo_pass  = True
ask_pass        = True
transport      = smart
remote_port    = 22
module_lang    = C
module_set_locale = False

# plays will gather facts by default, which contain information about
# the remote system.
#
# smart - gather by default, but don't regather if already gathered
# implicit - gather by default, turn off with gather_facts: False
# explicit - do not gather by default, must say gather_facts: True
#gathering = implicit

# This only affects the gathering done by a play's gather_facts directive,
# by default gathering retrieves all facts subsets
-- INSERT --
```

Edit the `/etc/ansible/ansible.cfg` file and uncomment the
#inventory= /etc/ansible/hosts
sudo_user = root

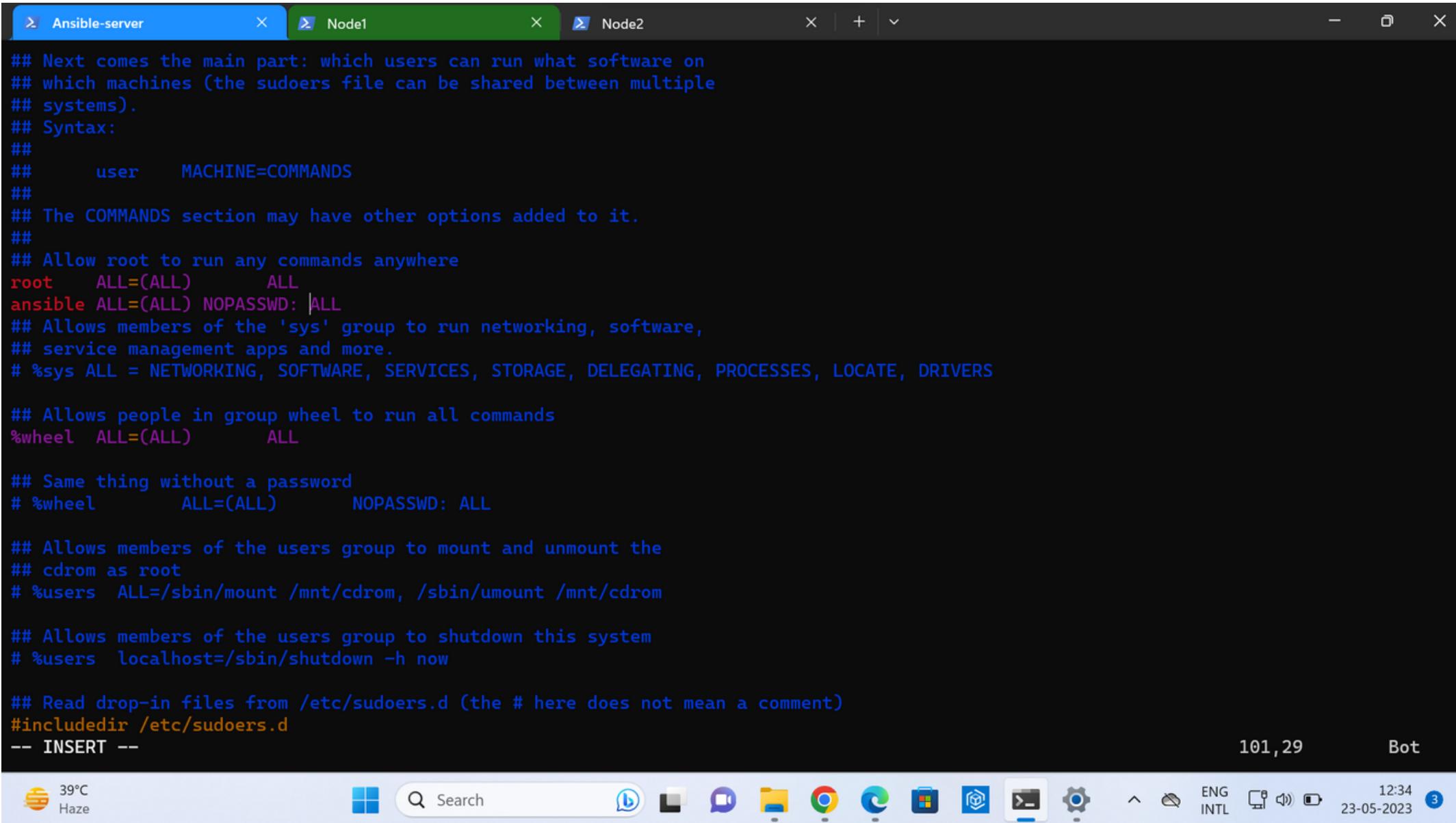
The screenshot shows a terminal window with three tabs: 'Ansible-server' (blue bar), 'Node1' (green bar), and 'Node2' (dark blue bar). The 'Ansible-server' tab displays a list of system users and services from /etc/passwd. The list includes root, bin, daemon, adm, lp, sync, shutdown, halt, mail, operator, games, ftp, nobody, systemd-network, dbus, rpc, libstoragemgmt, sshd, rpcuser, nfsnobody, rngd, chrony, ec2-instance-connect, postfix, tcpdump, ec2-user, ansible, and apache. The 'Node1' and 'Node2' tabs show similar lists, indicating they are running the same configuration.

```
root:x:0:0:root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
libstoragemgmt:x:999:997:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
rngd:x:998:996:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
chrony:x:997:995::/var/lib/chrony:/sbin/nologin
ec2-instance-connect:x:996:994::/home/ec2-instance-connect:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
ansible:x:1001:1001:/home/ansible:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
~
~
~
~
~
~
~
~
```

```
ansible:x:1001:1001:/home/ansible:/bin/bash
```

Add users to the Instances as Ansible-user using the command **sudo adduser USERNAME** and then add the user password using the command **sudo passwd USERNAME**. I name it as **Ansible** in all of the machines.

I can check the list of all the created users using the command **sudo vim /etc/passwd**



```
Ansible-server      Node1          Node2
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##       user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
ansible ALL=(ALL) NOPASSWD: ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL

## Same thing without a password
# %wheel      ALL=(ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
-- INSERT --
101,29           Bot
```

39°C Haze Search ⬤ 📂 🌐 📺 🎙 🚧 ENG INTL 12:34 23-05-2023

After adding my Ansible user and logging in as an Ansible user to my machines, I must need the **SUDO Privileges**. For That, i need to edit the "**visudo**" file as a root user and add one more line allowing Asible as a sudo user with ALL the privileges.

sudo visudo

[ansible ALL=(ALL) NOPASSWD: ALL]

```
# Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no
```

To allow the SSH connection between the different nodes. The sshd_config file must be edited

sudo vim /etc/ssh/sshd_config

uncomment:

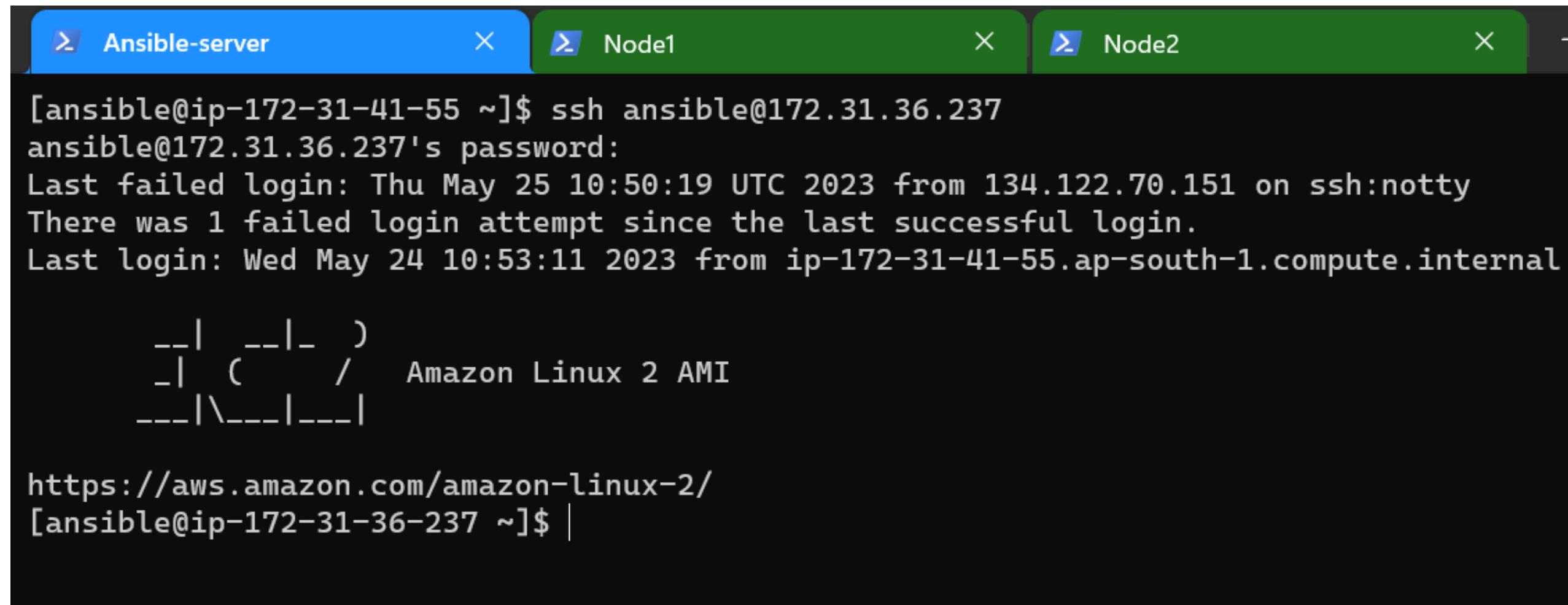
Permitrootlogin yes

Password Authentication Yes

Comment-out:

#Password Authentication No

Now the Ansible server can access these servers, using their username and private IP ex: **ansible@172.14.223.32**. But the Ansible server admin must know the password for each user.



```
[ansible@ip-172-31-41-55 ~]$ ssh ansible@172.31.36.237
ansible@172.31.36.237's password:
Last failed login: Thu May 25 10:50:19 UTC 2023 from 134.122.70.151 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Wed May 24 10:53:11 2023 from ip-172-31-41-55.ap-south-1.compute.internal

__| __|_
_| (   /  Amazon Linux 2 AMI
___\_\_|\__|_|
https://aws.amazon.com/amazon-linux-2/
[ansible@ip-172-31-36-237 ~]$ |
```

ssh ansible@172.31.36.237

If a Company have 500 Nodes, to SSH them to the Ansible server, the user needs passwords for each user.

That causes security conflicts. For that, Ansible has **Trust Relationship Concept**.

Here we create a Public key using **SSH-keygen** in our Ansible server. And then we will copy the Public key of the Ansible server to other nodes from **.ssh** directory using the command.

sudo ssh-copy-id ansible@<IPv4>

```
[ansible@ip-172-31-41-55 ~]$ ssh-copy-id ansible@172.31.36.237
```

This needed to be done for each node, this will send the server's Public SSH-key and make a trusted relation with the server nodes.

```
[root@rhel-8 ~]# ansible database_servers -m user -a "name=james password=redhat"
[WARNING]: The input password appears not to have been hashed. The 'password' argument must be encrypted for this module
to work properly.

173.82.202.239 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1001,
    "home": "/home/james",
    "name": "james",
    "password": "NOT_LOGGING_PASSWORD",
    "shell": "/bin/sh",
    "state": "present",
    "system": false,
    "uid": 1001
}
```

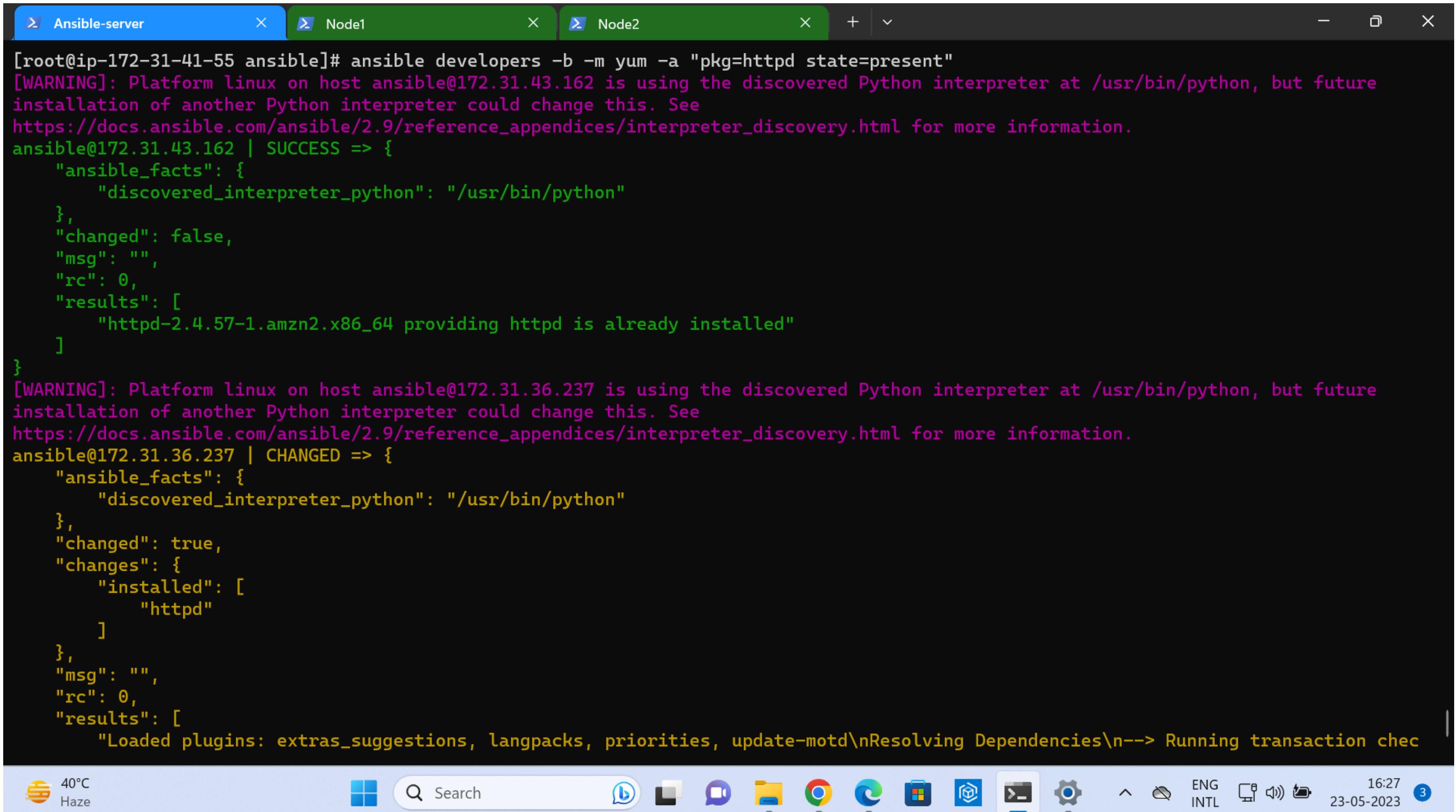
```
7  ansible developers[0] -b -m copy -a "src=AnsibleDocsToShear dest=/temp"
8  ansible developers -b -m copy -a "src=AnsibleDocsToShear dest=/temp"
```

Ansible Adhoc Commands:- Adhoc commands are temporary ways to connect the nodes without any idempotency. These are Individual Commands to run or perform quick tasks.

ANSIBLE AD HOC COMMANDS - SYNTAX

Host Group	Module	Arguments to the module
ansible	webserver	-m yum -a "name=httpd state=latest"
ansible	allservers	-m shell -a " find /opt/oracle -type f -mtime +10 -name *.log "
ansible	appserver	-m user -a "name=saravak group=admins append=yes shell=bin/bash"

Modules in Ansible:-



```
[root@ip-172-31-41-55 ansible]# ansible developers -b -m yum -a "pkg=httpd state=present"
[WARNING]: Platform linux on host ansible@172.31.43.162 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ansible@172.31.43.162 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "httpd-2.4.57-1.amzn2.x86_64 providing httpd is already installed"
    ]
}
[WARNING]: Platform linux on host ansible@172.31.36.237 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ansible@172.31.36.237 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "changes": {
        "installed": [
            "httpd"
        ]
    },
    "msg": "",
    "rc": 0,
    "results": [
        "Loaded plugins: extras_suggestions, langpacks, priorities, update-motd\\nResolving Dependencies\\n--> Running transaction chec
```

40°C Haze

Search

16:27 23-05-2023

```
Ansible-server
Node1
Node2

}

[root@ip-172-31-41-55 ansible]# ansible developers -b -m service -a "name=httpd state=started"
[WARNING]: Platform linux on host ansible@172.31.43.162 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.
ansible@172.31.43.162 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "name": "httpd",
    "state": "started",
    "status": {
        "ActiveEnterTimestampMonotonic": "0",
        "ActiveExitTimestampMonotonic": "0",
        "ActiveState": "inactive",
        "After": "tmp.mount system.slice remote-fs.target systemd-journald.socket -.mount nss-lookup.target network.target httpd-init
.service basic.target",
        "AllowIsolate": "no",
        "AmbientCapabilities": "0",
        "AssertResult": "no",
        "AssertTimestampMonotonic": "0",
        "Before": "shutdown.target",
        "BlockIOAccounting": "no",
        "BlockIOWeight": "18446744073709551615",
        "CPUAccounting": "no",
        "CPUQuotaPerSecUsec": "infinity",
        "CPUSchedulingPolicy": "0",
        "CPUSchedulingPriority": "0",
        "CPUSchedulingResetOnFork": "no",
        "CPUShares": "18446744073709551615",
        "CanIsolate": "no",
    }
}
```

```
[root@ip-172-31-41-55 ansible]# ansible developers -b -m user -a "name=Uttam-Pundir-BM"
[WARNING]: Platform linux on host ansible@172.31.43.162 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ansible@172.31.43.162 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1003,
    "home": "/home/Uttam-Pundir-BM",
    "name": "Uttam-Pundir-BM",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1003
}
[WARNING]: Platform linux on host ansible@172.31.36.237 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ansible@172.31.36.237 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1002,
    "home": "/home/Uttam-Pundir-BM",
    "name": "Uttam-Pundir-BM",
}
```

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
libstoragemgmt:x:999:997:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
rngd:x:998:996:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
chrony:x:997:995:/var/lib/chrony:/sbin/nologin
ec2-instance-connect:x:996:994::/home/ec2-instance-connect:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
tcpdump:x:72:72::/sbin/nologin
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
ansible:x:1001:1001::/home/ansible:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
Uttam-Pundir-BM:x:1002:1002::/home/Uttam-Pundir-BM:/bin/bash
~
~
-- INSERT -- W10: Warning: Changing a readonly file
```

The screenshot shows a terminal window with three tabs: "Ansible-server" (blue bar), "Node1" (green bar, active tab), and "Node2" (dark blue bar). The "Node1" tab displays the following Ansible command output:

```
"ping": "pong"
}
[root@ip-172-31-41-55 ansible]# ansible developers -m setup
[WARNING]: Platform linux on host ansible@172.31.36.237 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ansible@172.31.36.237 | SUCCESS => {
  "ansible_facts": {
    "ansible_all_ipv4_addresses": [
      "172.31.36.237"
    ],
    "ansible_all_ipv6_addresses": [
      "fe80::9a:84ff:fe2b:c8a6"
    ],
    "ansible_apparmor": {
      "status": "disabled"
    },
    "ansible_architecture": "x86_64",
    "ansible_bios_date": "08/24/2006",
    "ansible_bios_version": "4.11.amazon",
    "ansible_cmdline": {
      "BOOT_IMAGE": "/boot/vmlinuz-5.10.178-162.673.amzn2.x86_64",
      "biosdevname": "0",
      "console": "ttyS0,115200n8",
      "net.ifnames": "0",
      "nvme_core.io_timeout": "4294967295",
      "rd.emergency": "poweroff",
      "rd.shell": "0",
      "ro": true,
      "root": "UUID=902c5313-0003-4b1a-a5ef-6981dc3fd4a7"
    },
    "ansible_date_time": {
```

Cheatsheet

Modules	Module Categories							
System	User	Group	Iptables	Mount	Ping	Systemd	Service	Hostname
Commands	Command	Expect	Raw	Script	Shell			
Files	Acl	Archive	FInd	Copy	Replace	Stat	File	Unarchive
Database	MySQL	MongoDB	MSSQL	PostgreSQL	ProxySQL	Vertica		
Cloud	Amazon	Azure	Google	Linode	Openstack	VMware	Docker	Atomic
Windows	Win_copy	Win_command	Win_msi	Win_ping	Win_msq	Win_shell	Win_path	Win_service

Playbooks

```
Ansible-server      Node1
--- # My first Testing Playbook
- hosts: developers
  user: ansible
  become: yes
  connection: ssh
  gather_facts: yes
```

}

Targets

```
[root@ip-172-31-41-55 ansible]# ansible-playbook target.yml

PLAY [developers] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host ansible@172.31.36.237 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.
ok: [ansible@172.31.36.237]
[WARNING]: Platform linux on host ansible@172.31.43.162 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.
ok: [ansible@172.31.43.162]

PLAY RECAP ****
ansible@172.31.36.237 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansible@172.31.43.162 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ip-172-31-41-55 ansible]# |
```

```
- hosts: developers
  user: ansible
  become: yes
  connection: ssh
  gather_facts: yes
  tasks:
    - name: Install Httpd server in linux systems for developers
      action: yum name=httpd state=installed
```

```
[root@ip-172-31-41-55 ansible]# ansible-playbook target.yml

PLAY [developers] ****

TASK [Gathering Facts] ****
[WARNING]: Platform linux on host ansible@172.31.36.237 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.
ok: [ansible@172.31.36.237]
[WARNING]: Platform linux on host ansible@172.31.43.162 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.
ok: [ansible@172.31.43.162]

PLAY RECAP ****
ansible@172.31.36.237 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansible@172.31.43.162 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
[root@ip-172-31-41-55 ansible]# cat target.yml
--- # My first Testing Playbook
- hosts: developers
  user: ansible
  become: yes
  connection: ssh
  gather_facts: yes
```

variables

```
Ansible-server ✘ Node1 ✘ Node2
--- # THIS IS THE YAML Playbook that contains TASKS, VARIABLES, TARGETS
- hosts: developers
  user: ansible
  become: yes
  connection: ssh
  gather_facts: yes
  vars:
    pkgname: httpd
  tasks:
    - name: Install the HTTPD server into the Linux
      action: yum name='{{pkgname}}' state=installed
~ ~
```

Handlers

The image shows a terminal window with three tabs: "Ansible-server", "Node1", and "Node2".

Ansible-server Tab:

```
--- # HANDLER PLAYBOOK
- hosts: developers
  user: ansible
  become: yes
  connection: ssh
  gather_facts: yes
  tasks:
    - name: Install the HTTPD servers for Linux
      action: yum name=httpd state=installed
      notify: restart HTTPD
  handlers:
    - name: restart HTTPD
      action: service name=httpd state=restarted
```

Node1 Tab:

```
ok: [ansible@172.31.43.162]

TASK [Install the HTTPD servers for Linux] *****
changed: [ansible@172.31.36.237]
changed: [ansible@172.31.43.162]

RUNNING HANDLER [restart HTTPD] *****
changed: [ansible@172.31.43.162]
changed: [ansible@172.31.36.237]

PLAY RECAP *****
ansible@172.31.36.237 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansible@172.31.43.162 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Node2 Tab:

```
[root@ip-172-31-41-55 ansible]# ansible-playbook handler.yml

PLAY [developers] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host ansible@172.31.43.162 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible@172.31.43.162]
[WARNING]: Platform linux on host ansible@172.31.36.237 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible@172.31.36.237]

TASK [Install the HTTPD servers for Linux] *****
changed: [ansible@172.31.43.162]
changed: [ansible@172.31.36.237]

RUNNING HANDLER [restart HTTPD] *****
changed: [ansible@172.31.36.237]
changed: [ansible@172.31.43.162]

PLAY RECAP *****
ansible@172.31.36.237 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansible@172.31.43.162 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Loops

```
Ansible-server X Node1 X
--- # Loops playbook in YAML
- hosts: developers
  user: ansible
  become: yes
  connection: ssh
  tasks:
    - name: add list of users in the servers\
      user: name='{{item}}' state=present
      with_items:
        - Uttam
        - Manik
        - Mayank
        - Mukul
        - Aniket
        - Varnit
        - Vinayak
        - Bangmetric
~
```

```
Ansible-server X Node1 X Node2 X + ▾
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
libstoragemgmt:x:999:997:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
rngd:x:998:996:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
chrony:x:997:995::/var/lib/chrony:/sbin/nologin
ec2-instance-connect:x:996:994::/home/ec2-instance-connect:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
ansible:x:1001:1001::/home/ansible:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
Uttam-Pundir-BM:x:1002:1002::/home/Uttam-Pundir-BM:/bin/bash
Uttam:x:1003:1003::/home/Uttam:/bin/bash
Manik:x:1004:1004::/home/Manik:/bin/bash
Mayank:x:1005:1005::/home/Mayank:/bin/bash
Mukul:x:1006:1006::/home/Mukul:/bin/bash
Aniket:x:1007:1007::/home/Aniket:/bin/bash
Varnit:x:1008:1008::/home/Varnit:/bin/bash
Ambuj:x:1009:1009::/home/Ambuj:/bin/bash
```



Troubleshooting

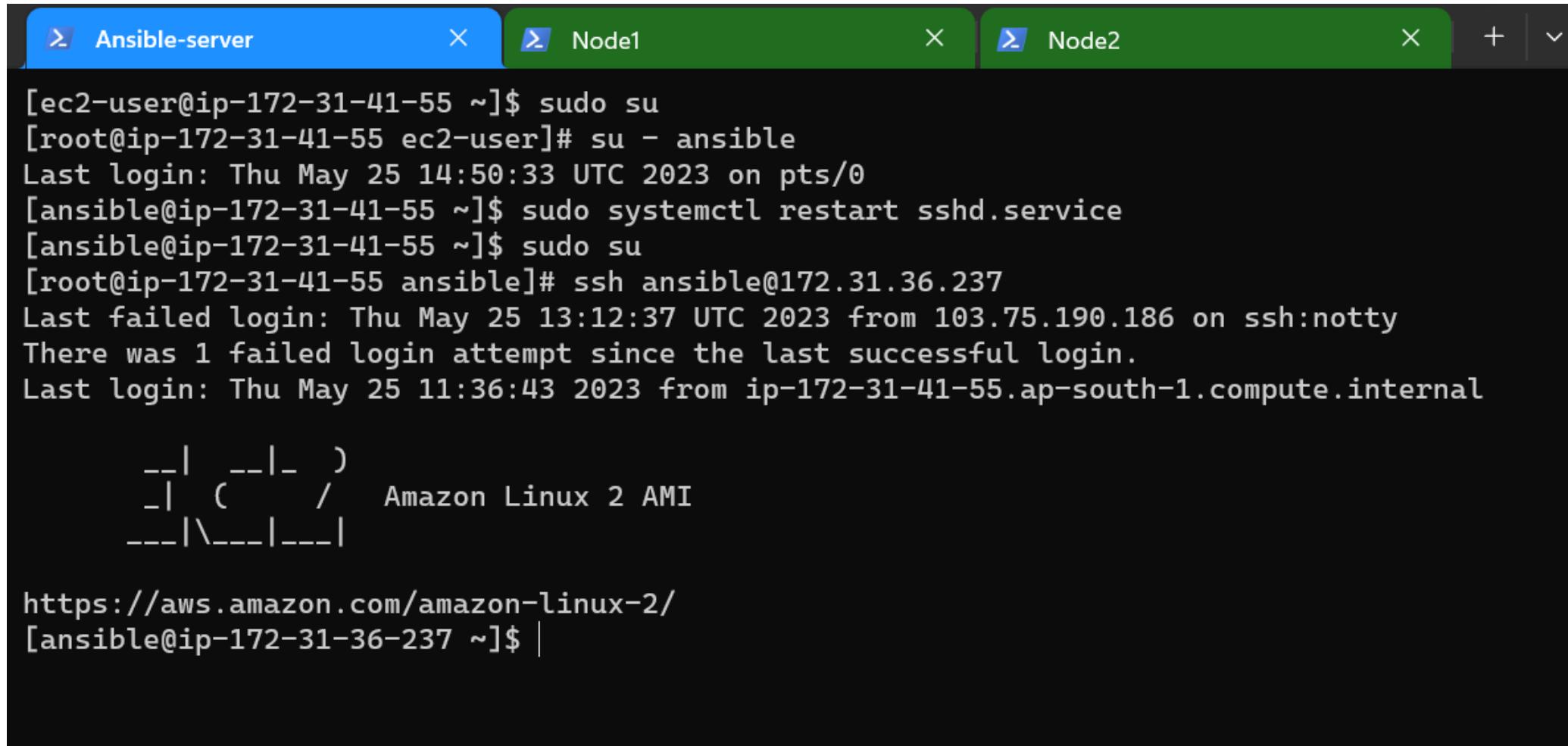
```
Connection to 172.31.0.78 closed.  
[ansible@ip-172-31-13-192 ~]$ ansible all -a "ls"  
172.31.7.67 | UNREACHABLE! => {  
    "changed": false,  
    "msg": "Failed to connect to the host via ssh: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).",  
    "unreachable": true  
}  
172.31.0.78 | UNREACHABLE! => {  
    "changed": false,  
    "msg": "Failed to connect to the host via ssh: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).",  
    "unreachable": true  
}  
5
```

This is the error message we will get when we need to run some specific command that requires taking configuration from all the nodes. The error says we need a password to connect to the node for a proper ssh connection between the Ansible server.

```
[ec2-user@ip-172-31-41-55 ~]$ sudo su
[root@ip-172-31-41-55 ec2-user]# su - ansible
Last login: Thu May 25 14:50:33 UTC 2023 on pts/0
[ansible@ip-172-31-41-55 ~]$ sudo systemctl restart sshd.service
[ansible@ip-172-31-41-55 ~]$ |
```

The first solution is that as an Ansible user, we have to restart the **sshd** service, It will also **restart** the **sshd_config** files as well.

It will solve this error



```
[ec2-user@ip-172-31-41-55 ~]$ sudo su
[root@ip-172-31-41-55 ec2-user]# su - ansible
Last login: Thu May 25 14:50:33 UTC 2023 on pts/0
[ansible@ip-172-31-41-55 ~]$ sudo systemctl restart sshd.service
[ansible@ip-172-31-41-55 ~]$ sudo su
[root@ip-172-31-41-55 ansible]# ssh ansible@172.31.36.237
Last failed login: Thu May 25 13:12:37 UTC 2023 from 103.75.190.186 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Thu May 25 11:36:43 2023 from ip-172-31-41-55.ap-south-1.compute.internal

--| --|- )
_| ( /  Amazon Linux 2 AMI
---|\---|
```

https://aws.amazon.com/amazon-linux-2/
[ansible@ip-172-31-36-237 ~]\$ |

```
[root@ip-172-31-36-237 ansible]# ssh-copy-id ansible@172.31.36.237
```

if it doesn't resolve this error then we can use the above command to again copy the public ID of Ansible servers. we can also check the **/etc/ansible/hosts** file if all nodes are added to the inventory or not.

```
#LoginGraceTime 2m
|PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

To Avoid this error we must check these lines of
/etc/ssh/ssh_config must be uncommented
 PermitRootLogin yes
 PasswordAuthentication Yes

```
# To disable tunneled clear text passwords, change to no here!
|PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

```
[root@ansible-controller ~]# ansible all -m ping -v
Using /etc/ansible/ansible.cfg as config file
host-two | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
host-one | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[root@ansible-controller ~]#
```

we can use this command to check whether the Ansible server has a proper connection or not. If the status Shows SUCCESS then its good else we have to troubleshoot again.

```
[ansible@ip-172-31-41-55 ~]$ ansible-playbook handler.yml --check

PLAY [developers] ****

TASK [Gathering Facts] ****
fatal: [ansible@172.31.36.237]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).", "unreachable": true}
fatal: [ansible@172.31.43.162]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).", "unreachable": true}

PLAY RECAP ****
ansible@172.31.36.237      : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
ansible@172.31.43.162      : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

This error shows the same problem that the SSH connection with the Ansible-Server I have followed solution number 1 to solve this error.

```
[ansible@ip-172-31-41-55 ~]$ ansible-playbook handler.yml
ERROR! We were unable to read either as JSON nor YAML, these are the errors we got from each:
JSON: No JSON object could be decoded

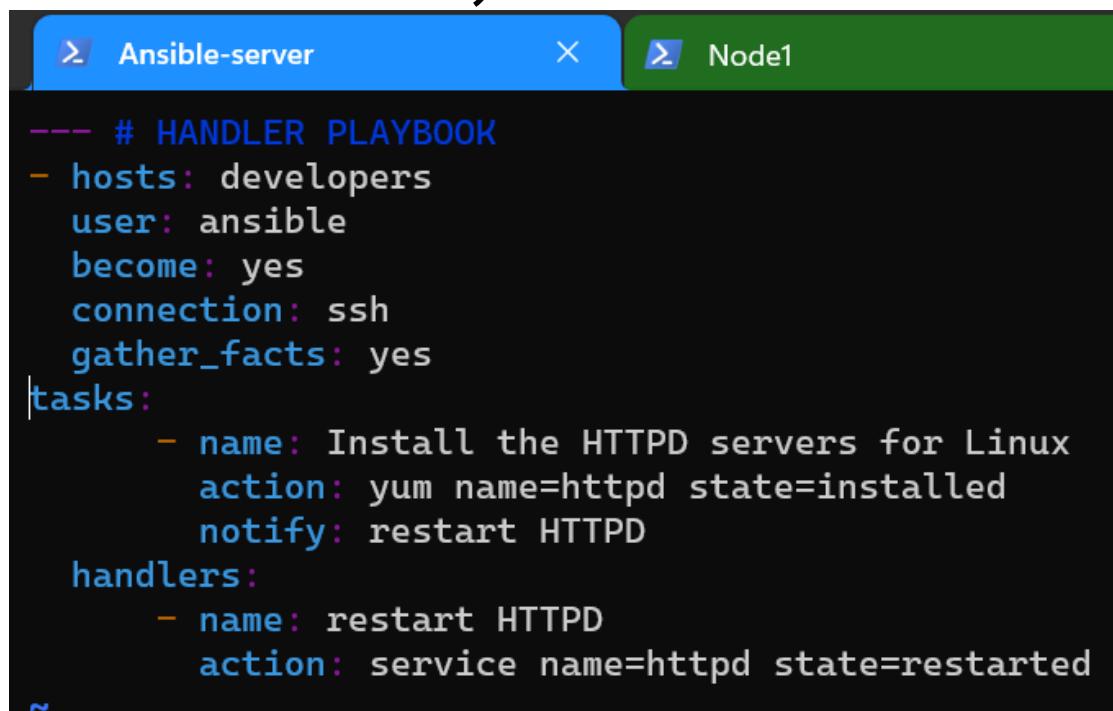
Syntax Error while loading YAML.
  did not find expected '-' indicator

The error appears to be in '/home/ansible/handler.yml': line 7, column 1, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

    gather_facts: yes
tasks:
^ here
```

while running the playbook if we got this error it is because of an indentation error, which means the YAML code is not properly written in order.



The terminal window shows the command `ansible-playbook handler.yml` being run. The title bar of the terminal window says "Ansible-server". The terminal output displays an error message indicating that the YAML file is invalid due to incorrect indentation. The error message points to line 7 of the file, specifically highlighting the word "here" as the point of failure.

```
Ansible-server
Node1

--- # HANDLER PLAYBOOK
- hosts: developers
  user: ansible
  become: yes
  connection: ssh
  gather_facts: yes
  tasks:
    - name: Install the HTTPD servers for Linux
      action: yum name=httpd state=installed
      notify: restart HTTPD
  handlers:
    - name: restart HTTPD
      action: service name=httpd state=restarted
```

Here the task is not properly arranged It
should be proper,
Solution of this error message is simple we
just need to give correct spacings

The screenshot shows the AWS Management Console with the EC2 service selected. The user is in the 'Launch an instance' section, specifically the 'Select AMI' step. A search bar at the top is empty. Below it, four tabs are visible: 'Quickstart AMIs (47)', 'My AMIs (0)', 'AWS Marketplace AMIs (7888)', and 'Community AMIs (500)'. The 'AWS Marketplace AMIs' tab is selected. On the left, a sidebar contains filters for 'Free tier only', 'OS category' (selected), 'Architecture' (selected), and 'Platform'. The main area displays two AMI options: 'Amazon Linux 2023 AMI' and 'Amazon Linux 2 AMI (HVM)'. The 'Amazon Linux 2023 AMI' is highlighted with a blue border. Its details show it's a 64-bit (x86) AMI with a root device of 'amazon'. To the right of the details is a 'Select' button and a radio button for '64-bit (x86)' which is selected. Below the AMI cards is a terminal window showing the command-line process of installing the EPEL repository on an Amazon Linux 2023 instance. The terminal output includes:

```
Warning: Permanently added 'ec2-13-126-176-220.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.  
#  
~\_ #####_      Amazon Linux 2023  
~~ \_######\_\_  
~~   \##|_|  
~~   \#/ _-- https://aws.amazon.com/linux/amazon-linux-2023  
~~     V~' '--->  
~~     /  
~~..._/_/  
~/m/'  
[ec2-user@ip-172-31-15-110 ~]$ sudo wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm  
--2023-05-26 05:49:47-- https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm  
Resolving dl.fedoraproject.org (dl.fedoraproject.org)... 38.145.60.24, 38.145.60.22, 38.145.60.23  
Connecting to dl.fedoraproject.org (dl.fedoraproject.org)|38.145.60.24|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 15608 (15K) [application/x-rpm]  
Saving to: 'epel-release-latest-7.noarch.rpm'  
  
epel-release-latest-7.noarch.rpm 100%[=====] 15.24K 78.1KB/s in 0.2s  
  
2023-05-26 05:49:49 (78.1 KB/s) - 'epel-release-latest-7.noarch.rpm' saved [15608/15608]  
  
[ec2-user@ip-172-31-15-110 ~]$ ls  
epel-release-latest-7.noarch.rpm  
[ec2-user@ip-172-31-15-110 ~]$ sudo yum install epel-release-latest-7.noarch.rpm  
Last metadata expiration check: 0:01:28 ago on Fri May 26 05:48:43 2023.  
Error:  
Problem: conflicting requests  
- nothing provides redhat-release >= 7 needed by epel-release-7-14.noarch
```

If we are using the Amazon Linux 2023 AMI It will show an error installing Ansible Package.

The screenshot shows the AWS Lambda console. At the top, there's a search bar with 'Search' and a 'Create Function' button. Below the search bar, there are tabs for 'Quickstart Functions (1)', 'My Functions (0)', 'AWS Marketplace Functions (257)', and 'Community Functions (500)'. A sidebar on the left has sections for 'Refine results' (with 'Clear all filters') and 'OS category' (listing 'All Linux/Unix', 'All Windows', and 'Architecture' options like '64-bit (Arm)' and '32-bit (x86)'). The main area displays a single function named 'Ansible' with a status of 'Running'. It shows deployment packages for 'aws_lambda_function-1' and 'aws_lambda_function-2'. The configuration tab is selected, showing details like 'Runtime: Python 3.9', 'Handler: lambda_function.lambda_handler', and 'Role: Lambda execution role (arn:aws:lambda:ap-south-1:123456789012:role/lambda-execution-role)'. The code tab contains the Lambda function code.

The screenshot shows a terminal window titled 'Ansible-server' with the command: `PS C:\Users\Uttam Pundir\downloads> ssh -i "Bangmetric-key.pem" ec2-user@ec2-65-0-184-232.ap-south-1.compute.amazonaws.com`. The output shows the user has logged in from 49.249.21.10 at Fri May 26 05:53:00 2023. The terminal then displays the Amazon Linux 2 AMI logo and the URL `https://aws.amazon.com/amazon-linux-2/`. Another terminal window titled 'Node1' shows the command: `[ec2-user@ip-172-31-41-55 ~]$ ls epel-release-latest-7.noarch.rpm`.

To solve this error we can use Amazon Linux 2 AMI. "or" Enable the "epel" using the command to avoid any problem:

sudo yum-config-manager --enable epel
sudo amazon-linux-extras install ansible2