



# Setup Ansible and Infrastructure



## Step 1: Installing Ansible

1. Login to AWS Console. Then navigate to EC2 and create an instance using Ubuntu Operating System.
2. Then you need to install Ansible on that and then you have to Launch 3 more EC2 instance using CentOS Operating System.
3. Now first launch your EC2 instance with Ubuntu OS. While creating your instance just give you security group a name so that you can remember it if some inbound rules are need to be added further. Then just simply launch your instance.
4. After that you need to launch 3 instances at the same time using CentOS.
5. Now for the OS in the **Application and OS Images section** click on **Browse more AMIs** then go to **Market Place** and search for **CentOS 9**. Choose the below AMI accordingly.
6. Then scroll down to Instance type and make sure to change it to t2.micro because there is no need to use a bigger instance.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Choose an Amazon Machine Image (AMI)' step, the user has searched for 'centos 9'. The results show 'CentOS Stream 9 (x86\_64)' by Amazon Web Services, which is selected. In the 'Instance type' section, 't2.micro' is chosen, and a note says 'The AMI vendor recommends using a t3.small instance (or larger) for the best experience with this product.'

7. After that you need to create a keypair for it.
8. So, now you need to go in the network settings and create a new security group and name it accordingly. Then you need to add a **new port** with **port number 22** and the **Source** should be your **security group** of the **first instance**.
9. After that you can launch your instances.

Instances (4) <a href="#">Info</a>		<a href="#">C</a>	<a href="#">Connect</a>	<a href="#">Instance state</a> ▾	<a href="#">Actions</a> ▾	<a href="#">Launch instances</a> ▾
<a href="#">Find Instance by attribute or tag (case-sensitive)</a>		Any state				
<input type="checkbox"/>	Name ↴	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	ansible-demo	i-09dfc2c67a6ff41c1	<span>Running</span> ⓘ ⓘ	t2.micro	<span>2/2 checks passed</span> <a href="#">View alarms</a> +	us-west-2c
<input type="checkbox"/>	demo-web-01	i-056733a4b9ae53890	<span>Running</span> ⓘ ⓘ	t2.micro	<span>Initializing</span> <a href="#">View alarms</a> +	us-west-2c
<input type="checkbox"/>	demo-web-02	i-04ec508ff35a8bd7	<span>Running</span> ⓘ ⓘ	t2.micro	<span>Initializing</span> <a href="#">View alarms</a> +	us-west-2c
<input type="checkbox"/>	demo-db-01	i-00f787aed1aa5e353	<span>Running</span> ⓘ ⓘ	t2.micro	<span>Initializing</span> <a href="#">View alarms</a> +	us-west-2c

Select an instance

10. Now wait for your instance until it gets to running state then you need to SSH into your instance by using either Putty tool, Command Prompt or even Git Bash.
11. You have to SSH into your first instance or for me it is ansible-demo.
12. Once you are in your machine you need to **install Ansible** into it. You don't need to install **Ansible** on any of the other machine because **Ansible** will be using **SSH** and these **EC2 instances already have SSH**.
13. So, you don't need to do any special setting on the CentOS machines.

```
ubuntu@ip-172-31-12-170: ~  X  +  ▾

System information as of Fri Feb  2 08:14:55 UTC 2024

System load: 0.2080078125      Processes:          101
Usage of /:   20.6% of 7.57GB    Users logged in:     0
Memory usage: 21%                IPv4 address for eth0: 172.31.12.170
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-12-170:~$ |
```

14. Now you need to install Ansible on it. For that you can go to Ansible documentation to see what are the commands to install it. Below is the link to that documentation.

[https://docs.ansible.com/ansible/latest/installation\\_guide/installation\\_distros.html](https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html)

15. First you need to run update command and update your instance. Then you need to run the Ansible installation command.

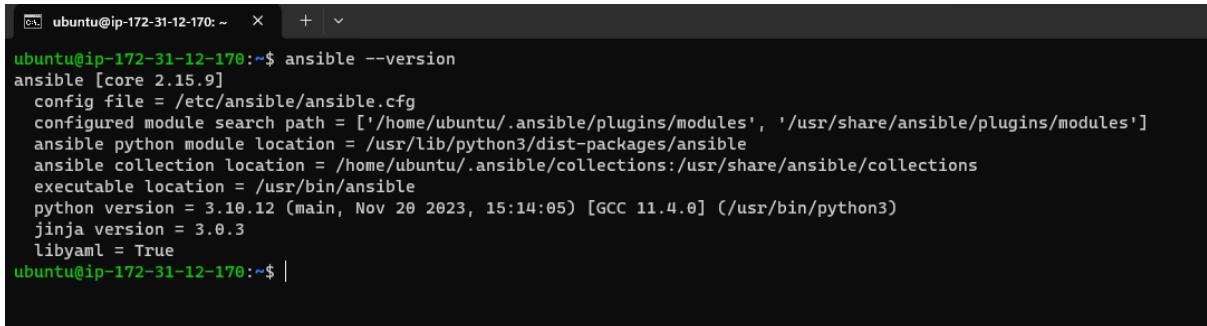
**sudo apt update**

**sudo add-apt-repository --yes --update ppa:ansible/ansible**

**sudo apt install ansible -y**

**ansible --version**

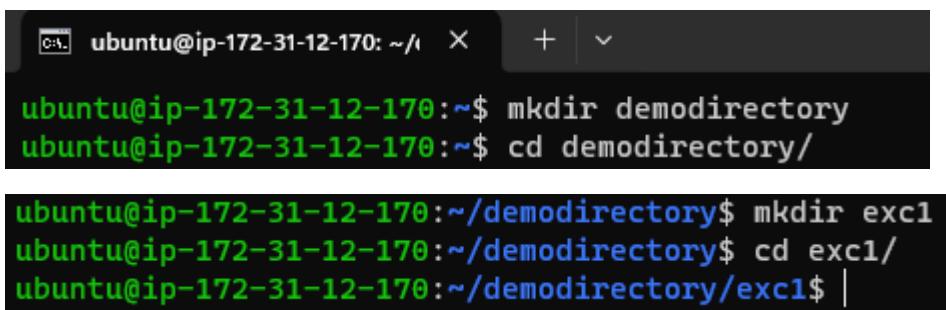
16. After the installation is completed, you can check the version of ansible.



```
ubuntu@ip-172-31-12-170:~$ ansible --version
ansible [core 2.15.9]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
ubuntu@ip-172-31-12-170:~$ |
```

## 💡 Step 2: Inventory and Ping Module

1. Now you are going to use Ansible to connect to the other EC2 instances that you have created using CentOS.
2. Let's say the other **EC2 instances** are your **target machines** and the **first EC2 instance** on which you did **SSH** and **installed Ansible** is your **control machine** which is controlling everything.
3. Now you need to create a directory on your control machine and then CD into it.
4. Now assume this as a git repository but you are going to create different folders for different exercises.



```
ubuntu@ip-172-31-12-170:~/ | × + ▾
ubuntu@ip-172-31-12-170:~$ mkdir demodirectory
ubuntu@ip-172-31-12-170:~$ cd demodirectory/
ubuntu@ip-172-31-12-170:~/demodirectory$ mkdir exc1
ubuntu@ip-172-31-12-170:~/demodirectory$ cd exc1/
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ |
```

5. Now we are going to use YAML format to write the inventory file. Simple write this command shown below. You can give any name you want other then inventory.

**vim inventory**

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ vim inventory
```

- Now you need to write this YAML code in the inventory. First you need to define your host, then what is the private IP of the host then what is its user name and what is the private ssh key file.

**all:**

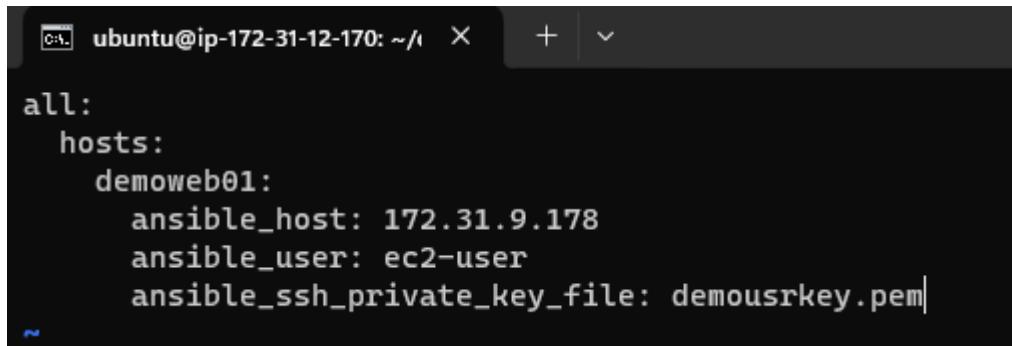
**hosts:**

**web01:**

**ansible\_host: 172.31.9.178**

**ansible\_user: ec2-user**

**ansible\_ssh\_private\_key\_file: demousrkey.pem**



```
ubuntu@ip-172-31-12-170: ~/i + ~
all:
  hosts:
    demoweb01:
      ansible_host: 172.31.9.178
      ansible_user: ec2-user
      ansible_ssh_private_key_file: demousrkey.pem
```

- Now as you can see that we have given a private key file name but we don't have that right now with us.
- So, for that you are going to exit from your instance and you are going to copy the file data and then come back to your instance create a file with the same name, go into the insert mode and paste that data into it then save it.
- Now go to the location of the file choose the file which you have assigned to CentOS instances, open it in notepad then copy it.



```
ubuntu@ip-172-31-12-170: ~/demodirectory/exc1$ ls
inventory
ubuntu@ip-172-31-12-170: ~/demodirectory/exc1$ |
```

- Come back to the instance do a **cat inventory** to see the contents of the inventory. Then create a file with the same name which you have given in the inventory.  
**vim demousrkey.pem**
- Now this will take you to the insert page there you need to paste the contents which you copied earlier.
- Then just save and quit.

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ ls
inventory
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ cat inventory
all:
  hosts:
    demoweb01:
      ansible_host: 172.31.9.178
      ansible_user: ec2-user
      ansible_ssh_private_key_file: demousrkey.pem
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ |
```

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAvB92wsgRzd9NWQuuZzs8R+Ydua/FCkzaY0uB6F2cvaq4s0/Y
0VXVs9WA7k0rvvc60XuYlemJLVmQth4cCRp6nhNb1hLN7oR4Ly7k7PHTQBMcRjN/c
wfq3FLoS0nHwzmXAmoC5GPbsT5cVr4lwJfOMPuevFE/ljkUBLBPCRuwi2wjUwxuf
7kXvAZErMh1q+lhgsD/vZXPNA8QdnU5vEDSLmWMR6HsaDrdDCIEFXLp6+M/guWAN
vpoiktIwreLQdWnLue0PGxFWAtxLPCutxy30Srx0YtsaFL0+bsmnZ6f4A7keZocn
b/QPGliknXZ1u0OYvshFtifpIT8m0Y68hu5NLwIDAQABAoIBAQChjb+2jIxFqUPK
ea6fd05RDHxwj1L+gnt1LE20mH3Xn/oX+zUuqLDqTCr5+wnaab14H9sFm84o9cTn
w3if3vXrrZKSdbUQ6zWkEY GebI6L3+EoQsNZTo7VXOp3EWSES3u481yB50ORgiUc
Z0IJRXsmNl7zaZ2wrPArE0DfGUdhqd3YKiDYuhVZcMsnIQ6JDQSreFyMs6mLU4xg
K/cx7bTDuL9qnVWjhdmKAuZMe6hx/Syi61ZZztQVLE0YBzMwZfg3Fu3TAgAPVls3
XwMzsNnTz+fQ7NXNNzLONUnKus0eA4x96VzzBR6Ti0hIosUGslK9+NKQOK0R0dM1
+EgZZhUxAoGBAN+aEIWNgst320STuVFu4XcErNpkBkoeyzqDp/SDMckZ1Cxeinxk9
WJHt63hCBN7Y3qB0bozrRe0BGemQzmwv5SbqAf njQ9iWLZIfpkY2yadNrW5AY0cq
MuGp37NPZ9hixz5ek89VMWN/3sHR4WhHzDxfQ8MfOPpIXxuNGtpD42JAoGBANDh
Znp5Tur0mPk8i4dT Kfx0KmMBN7NALQUgAfjjh71wlQAPTYyBkk32jX1zzHsGtD00
N4qZjBW1fQGkKatIGdz0I54tPwXvhR0iOs7DnQdQhGGdsPufCsH9Xb4l8GfItvyR
IMa4ZYRJbNiJ30fUIg+aZ+9+vW9q0B/qdll/JU73AoGASGK3toRXiEb k1059r78V
fLILIw3+e0A8ylSUqrNO Cqwx46s+Fg2l/pfnAqdcrEfIymjm4PFJS1Bv8wzNxng6
1r4qpZrM+/DDuupECbby0RZ0bpypwhBCOIulv4aVN5fCqI41i4ko0+0xT82BvJLM
ZivHMHf6dk2iaYXDsoHwZwECgYBXobiZckwKyvSGy/yVVeAHw42PdMpd82md4TUM
OD9daSH8wT0zQvDtKbjN3MCW+dZN97pfeTQHDS3Mgie2GgLz1SemzxAKZqLfhLLA
t7pnZHPIYWnGJyKwZU9D85nKEF9rif0E/bfISDuyKv2nkac+I/vP9Qj106j8S2BV
76aVZwKBgQCu2KmxwtDr siCdrzzFJn3l2ow93wB MtzVQ8B4Mu7Gkyw9j/uWfl6PE
ctNE3ybRvhyU1UVPO7pv0JG3W4A6fwFgoLoLRhyvx Fw3iafVym7nEC9q6g0GMkqC
/rgVdoX7u3prBm0z5jFHRixxw48l0dOd4NremdHT1YTRw/hUJi64qg==
-----END RSA PRIVATE KEY-----
^
^
:wq|
```

13. Now if you do a list of files you will see that you have two files one is inventory and other is demousrkey.

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ ls
demousrkey.pem  inventory
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ |
```

14. Now if you check for the permission, they are changed for demousrkey. They are more secure. Now it is 664.

**ls -l**

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ ls -l
total 8
-rw-rw-r-- 1 ubuntu ubuntu 1679 Feb  2 09:47 demousrkey.pem
-rw-rw-r-- 1 ubuntu ubuntu 142 Feb  2 09:34 inventory
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ |
```

15. Now if you try to SSH into your instance using Ansible you will see that it is asking you for a fingerprint but we don't need it because when you will launch multiple machine it will ask you for the fingerprint each time which is now feasible.

**ansible demoweb01 -m ping -i inventory**

16. For now just say NO.

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ ansible demoweb01 -m ping -i inventory
The authenticity of host '172.31.9.178 (172.31.9.178)' can't be established.
ED25519 key fingerprint is SHA256:M2dlBJfE35mrnz5aipZuz8S3m5rTtnshAyHJHlQvp0U.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? |
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ ansible demoweb01 -m ping -i inventory
The authenticity of host '172.31.9.178 (172.31.9.178)' can't be established.
ED25519 key fingerprint is SHA256:M2dlBJfE35mrnz5aipZuz8S3m5rTtnshAyHJHlQvp0U.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? no
demoweb01 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Host key verification failed.",
    "unreachable": true
}
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ |
```

17. Now we are going to use Ansible configuration to not check the host keys, just accept the connection and say yes by default.

18. For that we need to create an Ansible configuration file.

**sudo cat /etc/ansible/ansible.cfg**

19. Once you write the above command you will see that there is nothing here but if you look at the highlighted statement, this is all you need for now. This command will generate the Ansible configuration file.

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ sudo cat /etc/ansible/ansible.cfg
# Since Ansible 2.12 (core):
# To generate an example config file (a "disabled" one with all default settings, commented out):
#     $ ansible-config init --disabled > ansible.cfg
#
# Also you can now have a more complete file by including existing plugins:
# ansible-config init --disabled -t all > ansible.cfg

# For previous versions of Ansible you can check for examples in the 'stable' branches of each version
# Note that this file was always incomplete and lagging changes to configuration settings

# for example, for 2.9: https://github.com/ansible/ansible/blob/stable-2.9/examples/ansible.cfg
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ |
```

20. Now simply go into the root user. Then go this folder.

```
cd /etc/ansible/
```

21. Now do a listing of files here, you will see that there is an Ansible configuration file.

First you need to create a backup of this file because we don't want to lose it. So, just rename the file.

```
mv ansible.cfg ansible.cfg_backup
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ sudo -i
root@ip-172-31-12-170:~# cd /etc/ansible/
root@ip-172-31-12-170:/etc/ansible# ls
ansible.cfg  hosts  roles
root@ip-172-31-12-170:/etc/ansible# |
```

```
root@ip-172-31-12-170:/etc/ansible# mv ansible.cfg ansible.cfg_backup
root@ip-172-31-12-170:/etc/ansible# |
```

22. Once the backup is created just run the highlighted command from above then do a listing of files again you will see an Ansible configuration file which is just created.

23. Now you need to open this file.

```
vim ansible.cfg
```

```
root@ip-172-31-12-170:/etc/ansible# ansible-config init --disabled -t all > ansible.cfg
root@ip-172-31-12-170:/etc/ansible# ls
ansible.cfg  ansible.cfg_backup  hosts  roles
root@ip-172-31-12-170:/etc/ansible# |
```

24. Once you are in the file you need to search for **host\_key\_checking** then hit **enter** after click on **insert button** on your keyboard.

```
;host_key_checking=True

# (boolean) Facts are available
the main namespace.
# Unlike inside the `ansible_f
;inject_facts_as_vars=True

# (string) Path to the Python
ode. Supported discovery modes
All discovery modes employ a
lling back to a fixed ordered
lable. The fallback behavior w
alled later may change which o
gacy_silent``. The value of ``
nsible releases that always de
/host_key_checking|
```

25. First you need to remove the colon then you are going to write False instead of true.  
After hit ESC save it and quit.

```
host_key_checking=False

# (boolean) Facts are available in
the main namespace.
# Unlike inside the `ansible_facts
;inject_facts_as_vars=True

# (string) Path to the Python inter-
ode. Supported discovery modes are
All discovery modes employ a look-
lling back to a fixed ordered list
lable. The fallback behavior will
alled later may change which one is
gacy_silent``. The value of ``auto-
nsible releases that always default
:wa|
```

26. Now come back to Ubuntu user and run the command again to SSH.

```
ansible demoweb01 -m -ping -i inventory
```

27. Now you will see an error is occurring because of the permission which is 664. This permission is blocking us from SSH.

```
ubuntu@ip-172-31-12-178-:/demodirectory/exc1$ ls
demouserkey.pem inventory
ubuntu@ip-172-31-12-178-:/demodirectory/exc1$ ansible demoweb01 -m ping -i inventory
demoweb01 | UNREACHABLE! > {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Warning: Permanently added '172.31.9.178' (ED25519) to the list of known hosts.\r\nPermissions 0664 for 'demouserkey.pem' are too open.\r\n\r\nIt is required that your private key files are NOT accessible by others.\r\nThis private key will be ignored.\r\n\r\nLoad key \"demouserkey.pem\": bad permissions\r\nnec2-user@172.31.9.178: Permission denied (publickey,gssapi-keyexchange,gssapi-with-mic).",
    "unreachable": true
}
ubuntu@ip-172-31-12-178-:/demodirectory/exc1$ |
```

28. Now just change the permission to 400 of the PEM file, and you will be able to do the SSH.

```
chmod 400 demousrkey.pem
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ chmod 400 demousrkey.pem
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ ansible demoweb01 -m ping -i inventory
demoweb01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-12-170:~/demodirectory/exc1$ |
```

## 🤓 Step 3: More about Inventory

1. We have seen how we can set up inventory file. We did it for just one single host. Now we are going to do it for the other two host and we'll see few more concepts in inventory like groupings and variables.
2. As you can see, we go back to the directory and did a listing of files. Then I copied the contents of exc1 to exc2 and created exc2 simultaneously.
3. Then go to exc2 and do a listing, you will see the same files here too.

```
cp -r exc1/ exc2
```

```
cd exc2/
```

```
ls
```

```
ubuntu@ip-172-31-12-170:~/demodirectory$ ls
exc1
ubuntu@ip-172-31-12-170:~/demodirectory$ cp -r exc1/ exc2
ubuntu@ip-172-31-12-170:~/demodirectory$ cd exc2/
ubuntu@ip-172-31-12-170:~/demodirectory/exc2$ ls
demousrkey.pem  inventory
ubuntu@ip-172-31-12-170:~/demodirectory/exc2$ |
```

4. Now you need to open the inventory file.

```
vim inventory
```

5. Write this code there with your private IP address of instances and the private key file name. Then save it and quit.

```
all:
```

```
hosts:
```

```
demoweb01:
```

```
  ansible_host: 172.31.9.178
```

```
  ansible_user: ec2-user
```

```
  ansible_ssh_private_key_file: demousrkey.pem
```

```
demoweb02:
```

```
ansible_host: 172.31.8.250
ansible_user: ec2-user
ansible_ssh_private_key_file: demouserkey.pem
demodb01:
  ansible_host: 172.31.0.93
  ansible_user: ec2-user
  ansible_ssh_private_key_file: demouserkey.pem

children:
  webservers:
    hosts:
      demoweb01:
      demoweb02:
  dbservers:
    hosts:
      demodb01:
  demoall:
    children:
      webservers:
      dbservers:
```

```
ubuntu@ip-172-31-12-170: ~/. + | ^

all:
  hosts:
    demoweb01:
      ansible_host: 172.31.9.178
      ansible_user: ec2-user
      ansible_ssh_private_key_file: demousrkey.pem
    demoweb02:
      ansible_host: 172.31.8.250
      ansible_user: ec2-user
      ansible_ssh_private_key_file: demousrkey.pem
    demodb01:
      ansible_host: 172.31.0.93
      ansible_user: ec2-user
      ansible_ssh_private_key_file: demousrkey.pem

  children:
    webservers:
      hosts:
        demoweb01:
        demoweb02:
    dbservers:
      hosts:
        demodb01:
    demoall:
      children:
        webservers:
        dbservers:
      ~
      ~
-- INSERT --
```

6. Now if you run the SSH command  
**Ansible demoall -m ping -i inventory**
7. You will see that it is running successfully.

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc2$ ansible demoall -m ping -i inventory
demoweb01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
demoweb02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
demodb01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-12-170:~/demodirectory/exc2$ |
```

8. You can also say just ‘all’ and run the command it will also work.

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc2$ ansible all -m ping -i inventory
demoweb01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
demodb01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
demoweb02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-12-170:~/demodirectory/exc2$ |
```

9. You can also use ‘\*’ but you need to put it in the single quote.

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc2$ ansible '*' -m ping -i inventory  
demoweb01 | SUCCESS => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python3"  
    },  
    "changed": false,  
    "ping": "pong"  
}  
demoweb02 | SUCCESS => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python3"  
    },  
    "changed": false,  
    "ping": "pong"  
}  
demodb01 | SUCCESS => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python3"  
    },  
    "changed": false,  
    "ping": "pong"  
}  
ubuntu@ip-172-31-12-170:~/demodirectory/exc2$ |
```