Configuration Management

It is a process of handling changes systematically to ensure system integrity over a time.
It involves maintaining consistency of system changes by tracking and control changes in software/ hardware / networking and infra.
Why?
I have created 1 Python project and I want to run it over 10 servers (10 AWS Instances)

- create manually 10 servers.
- login to each server
- install python dependencies
- install github
- clone github repository
- then run python project.

These above process we can manage with help of ansible.
we can automate the configuration process.

let's Setup Ansible step by Step
Install Ansible
-- sudo apt update
-- sudo apt install ansible -y
-- ansible --version

Run Ansible Script (Playbook) without any Inventory.
first-playbook.yml

```
---
- name: Sample Playbook run with localhost
hosts: localhost
connection: local

tasks:
- name: say Hello
debug:
msg: "This a first Hello From Ansible"
- name: another Hello
debug:
```

```
msg: "Sample Debug Message"
```

To Run this you can run below command in wsl
ansible-playbook first-playbook.yml
also, above live will give warning like no inventory found and detect localhost later. To get result without warning you can run below command
ansible-playbook first-playbook.yml -i localhost, --connection=local

What is inventory File
File which tells Ansible to manage machines like how to connect with them.
It is having list of host with connection variables like username, ip, key to connect.
When we run our playbook it will take the server name from inventory file and run the entire playbook on all your servers.
How to Create
3 ways

- INI : traditional format (simple)
- YAML: (more structured format / modern)
- Dynamic (which we can generate by script)

```
all:
children:
web:
hosts:
aws1:
ansible_host:
aws2:
ansible_host:
db:
hosts:
db1:
ansible_host:
```

let's create one AWS Instance (ubuntu)
create unbuntu instance --> download key file --> network configuration --> SSH and HTTP port must be open --> Size 8 GB --> Launch

Then Create inventory file as mentioned.
Then check the Inventory file code using below command
ansible-inventory -i inventory.yml --list
let's ping the inventory file to check connectivity:

ansible -i inventory.yml all -m ping
(ansible is the adhoc command to directly run some command in your instance shell)
-- i form mentioning inventory file
all means i want to run mentioned command to all hosts.
-m (module) i want to run module which is ping.
To ping only one host use the host name:
sudo ansible -i inventory.yml aws1 -m ping

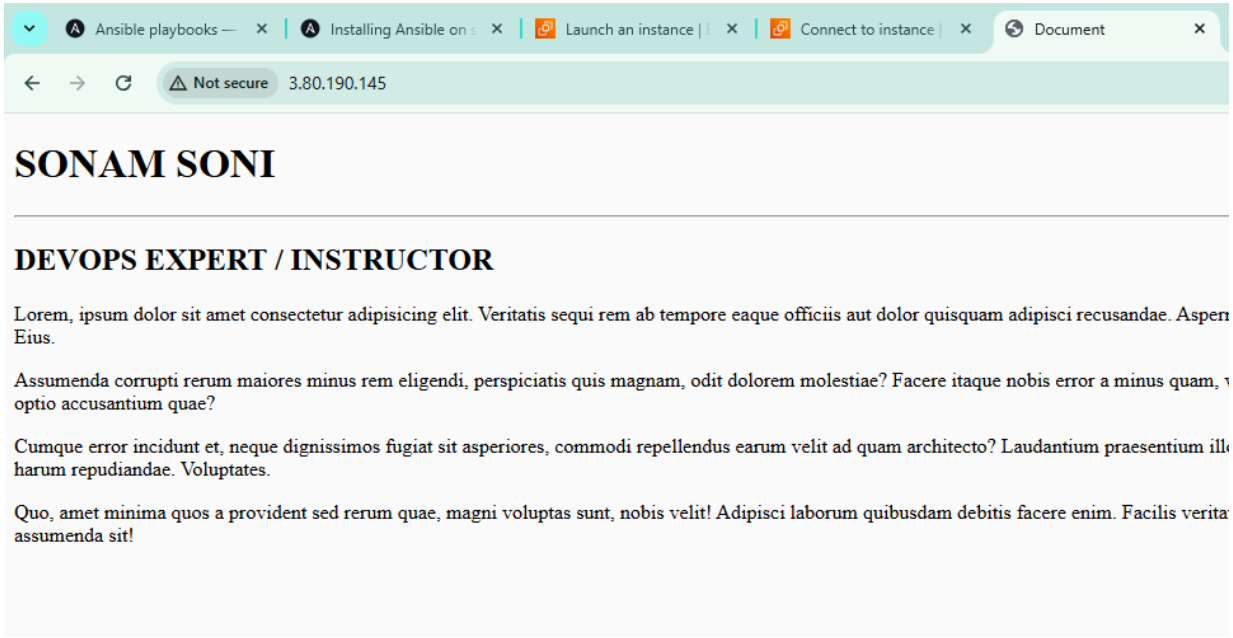
Let's Create One Playbook to install NGINX and copy index.html file and start the server.

```
---
- name: install nginx Server
hosts: aws1
become: yes
tasks:
- name: Update APT package index
apt:
update_cache: yes
- name: Install Nginx
apt:
name: nginx
state: present
- name: Copy HTML file to default location
copy:
src: index.html
dest: /var/www/html/index.html
# dest: /usr/share/nginx/html/index.html
# location for Linux
mode:
- name: Start Nginx
service:
name: nginx
state: started
enabled: yes
```

Above Code is OK for Ubuntu Instance. If you are creating instance which is amazon Linux then make sure you change apt to yum.
also file location of html as mentioned in comment.
use below command to run:
sudo ansible-playbook -i inventory.yml nginx.yml

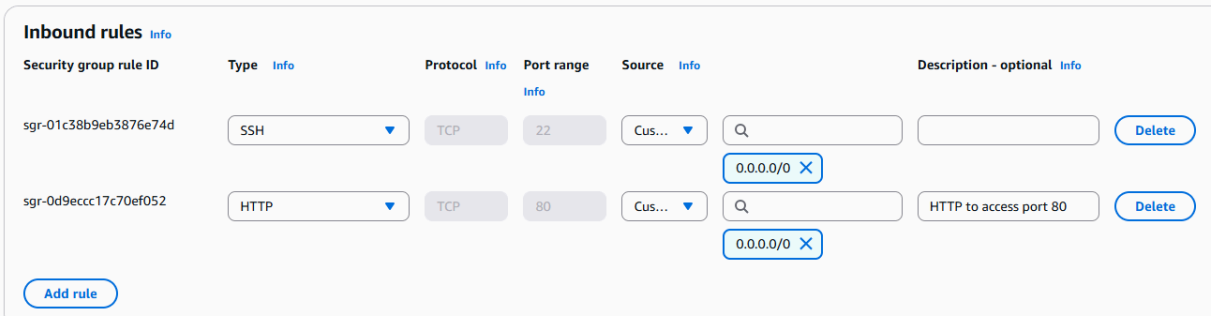
```
sonam@DESKTOP-4F8ELLU:/mnt/d/PhysicsWalla/Devops-March/Github/May-June-2025/session-27-ansible$ sudo ansible-playbook -i inventory.yml nginx
.yml

PLAY [install nginx Server] ****************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [aws1]

TASK [Update APT package index] ************************************************************************************
changed: [aws1]

TASK [Install Nginx] ***********************************************************************************************
ok: [aws1]

TASK [Copy HTML file to default location] **************************************************************************
changed: [aws1]

TASK [Start Nginx] *************************************************************************************************
ok: [aws1]

PLAY RECAP *********************************************************************************************************
aws1                       : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Once its started you can check the application deployed or not by accessing public ip / DNS in browser.



Incase If you are not able to see the output in browser make sure you are accessing via http not https.

still if you not able to access make sure your port 80 is open from anywhere in your instance.



Then again check the results.

Activity: Create Playbook which install Git and Curl in your instance using Ansible
Also verify the version of Git and curl to understand installation done successfully.