

General Goals

- Install Ansible (On WSL)
- Use Ansible to create an EC2 instance and install Git, Nginx, NodeJS, MongoDB, PM2, and other dependencies needed for the Sparta App (<https://github.com/AmeenahRiffin/tech501-sparta-app>), get the database running on it. Configuring the Port 3000 and the mongoDB port 27017.

Task: 3. Setup Ansible controller and first target node ('app' instance)

a. Task: Create EC2 instances for Ansible controller and first target node

b. Task: Setup dependencies for the Ansible controller and first target node

c. Task: Use other ad hoc commands

d. Task: Do update and upgrade on target nodes using ad hoc commands

e. Task: Consolidate ad hoc commands by copying a file to a target node

f. Task: Create and run playbook to install nginx on target node

g. Task: Create and run playbook to provision app VM

h. (Extension - if time) Task: Create and run playbook to print facts gathered

1. I was not able to install ansible on my windows machine, so I used WSL (Windows Subsystem for Linux) to install it. This was done by using the following command:

```
wsl --install -d Ubuntu
```

This allowed me to specify a distribution to install, which I chose to be Ubuntu. I set my username to AmeenahRiffin and the password to something I could remember.

To login to my ubuntu terminal if I log out, I navigate to the Ubuntu folder, and I used the following command:

```
wsl -d Ubuntu
```

2. Once I had installed WSL, I was in my ubuntu terminal. I then used the following commands to install ansible:

```
sudo apt update && sudo apt upgrade -y  
sudo apt install ansible -y
```

This installed ansible on my ubuntu machine.

```
ameenahriffin@AmeenahMSI:~$ ansible --version
ansible [core 2.16.3]
  config file = None
  configured module search path = ['/home/ameenahriffin/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ameenahriffin/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Feb  4 2025, 14:48:35) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ameenahriffin@AmeenahMSI:~$ |
```

3. I then created a new directory called ansible-sparta-app. In this directory I created a new file called playbook.yml and inventory.ini. This is where I wrote my playbook and inventory.

The commands I used were:

```
mkdir ansible-sparta-app
cd ansible-sparta-app
touch playbook.yml inventory.ini
```





4. Create inventory.ini file with target hosts:

```
[app]
<target-node-ip>

[all:vars]
ansible_user=ubuntu
ansible_ssh_private_key_file=~/.ssh/aws-ansible-key
```

5. For creating EC2 instances:

```
# Launch EC2 instances from AWS Console or CLI
# Ensure to use Ubuntu 18.04 AMI
# Configure security groups to allow SSH access
```

<input type="checkbox"/>	tech501-ameenah-ubuntu-2204-ansible-controller	i-05a00a6935d6f5cc0	Running	 	t3.micro
<input type="checkbox"/>	tech501-ameenah-ubuntu-2204-ansible-target-node-...	i-09f8ce47f77f4a300	Running	 	t3.micro

```

ansible_host=63.35.178.20 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: ssh: Could not resolve hostname ansible_host=63.35.178.20: Name or service not known",
  "unreachable": true
}
ansible_user=ubuntu | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: ssh: Could not resolve hostname ansible_user=ubuntu: Name or service not known",
  "unreachable": true
}
ansible_ssh_private_key_file=~/.ssh/ameenah-aws-key.pem | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: ssh: Could not resolve hostname ansible_ssh_private_key_file=~/.ssh/ameenah-aws-key.pem: Name or service not known",
  "unreachable": true
}

```

(This needed me to put the correct directory for the key, and also format the message so it's all on one line)

```

ameenahriffin@AmeenahMSI:/etc/ansible$ sudo nano hosts
ameenahriffin@AmeenahMSI:/etc/ansible$ ansible all -m ping
app | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}

```

```

ubuntu@ip-172-31-62-146:/etc/ansible$ ansible all -m ping
app | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-62-146:/etc/ansible$ ansible web -m ping
app | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}

```

command:

```

ubuntu@ip-172-31-62-146:/etc/ansible$ ansible web -m command -a "apt update" --become
app | CHANGED | rc=0 >>
Hit:1 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 126 kB in 0s (282 kB/s)
Reading package lists...
Building dependency tree...
Reading state information...
133 packages can be upgraded. Run 'apt list --upgradable' to see them.

```

shell:

```
ubuntu@ip-172-31-62-146:/etc/ansible$ ansible web -m shell -a "apt update && apt upgrade -y" --become
app | CHANGED | rc=0 >>
Hit:1 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists...
Building dependency tree...
Reading state information...
All packages are up to date.
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

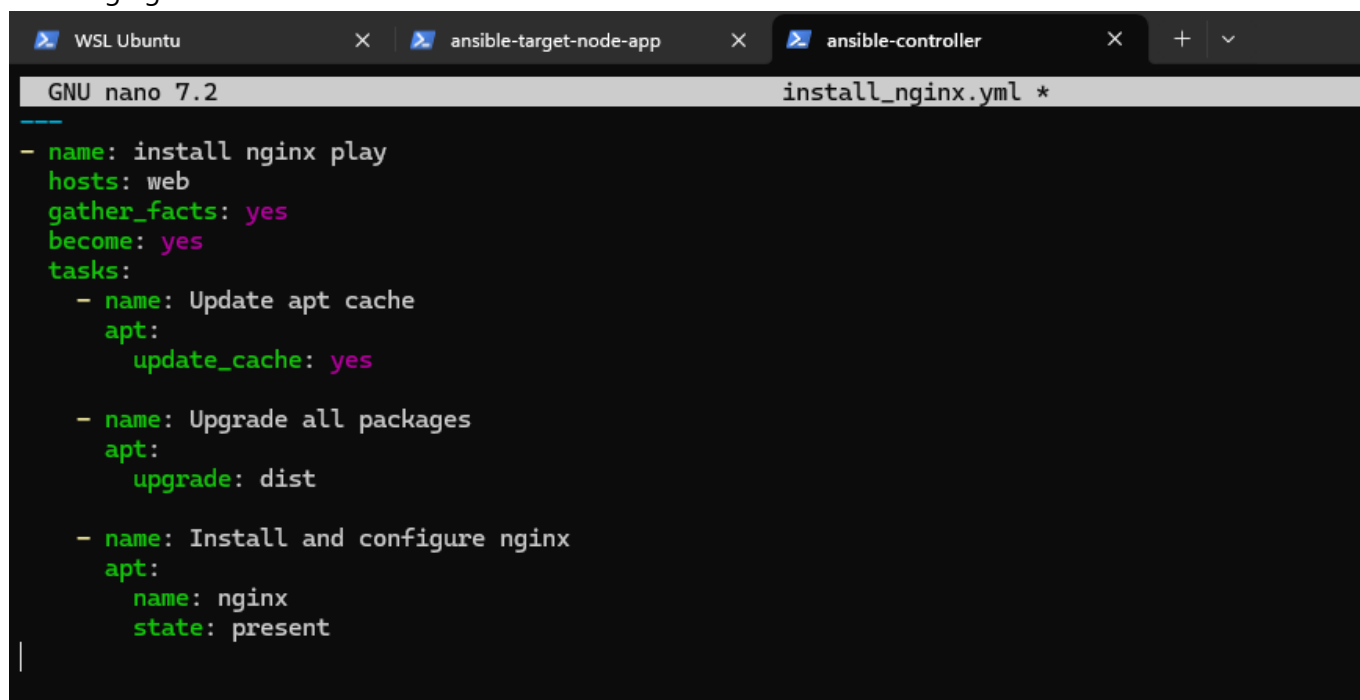
apt:

```
ubuntu@ip-172-31-62-146:/etc/ansible$ ansible web -m apt -a "update_cache=yes upgrade=dist" --become
app | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\n0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.\n",
  "stderr": "",
  "stderr_lines": [],
  "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\n0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.\n",
  "stdout_lines": [
    "Reading package lists...",
    "Building dependency tree...",
    "Reading state information...",
    "Calculating upgrade...",
    "0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded."
  ]
}
```

Copying SSH Key to the target node:

```
ubuntu@ip-172-31-62-146:/etc/ansible$ ansible web -m copy -a "src=~/.ssh/ameenah-aws-key.pem dest=/home/ubuntu/.ssh/ameenah-aws-key.pem owner=ubuntu group=ubuntu mode=0400"
app | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "checksum": "d925969b972c7506b9c6f56085452f715caa582f",
  "dest": "/home/ubuntu/.ssh/ameenah-aws-key.pem",
  "gid": 1000,
  "group": "ubuntu",
  "md5sum": "22893cce57bad747a3c5759e4e5e1034",
  "mode": "0400",
  "owner": "ubuntu",
  "size": 1679,
  "src": "/home/ubuntu/.ansible/tmp/ansible-tmp-1741577392.5390933-4141-129528703242842/source",
  "state": "file",
  "uid": 1000
}
```

Installing nginx:



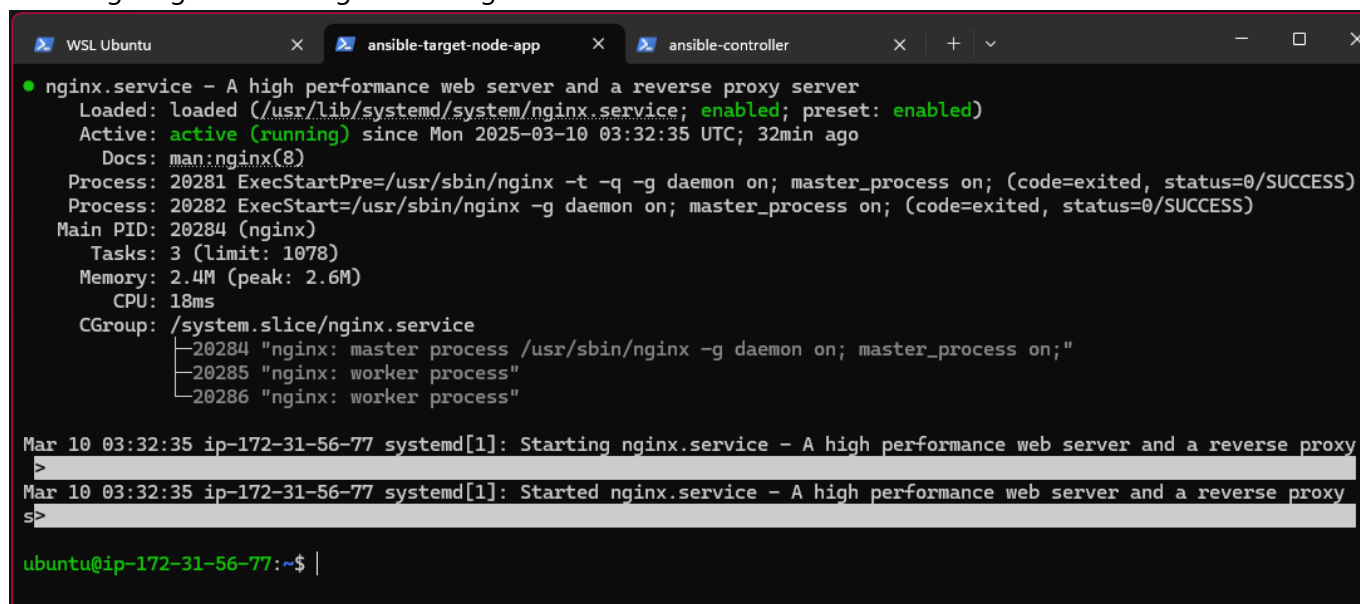
The screenshot shows a terminal window with three tabs: 'WSL Ubuntu', 'ansible-target-node-app', and 'ansible-controller'. The active tab is 'ansible-controller', which displays the content of a file named 'install_nginx.yml'. The file is being edited with 'GNU nano 7.2'. The YAML content defines an Ansible play for installing nginx on a host named 'web'. It includes tasks to update the apt cache, upgrade all packages, and install and configure nginx.

```
GNU nano 7.2 install_nginx.yml *
- name: install nginx play
  hosts: web
  gather_facts: yes
  become: yes
  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Upgrade all packages
      apt:
        upgrade: dist

    - name: Install and configure nginx
      apt:
        name: nginx
        state: present
```

Checking if nginx is running on the target node:

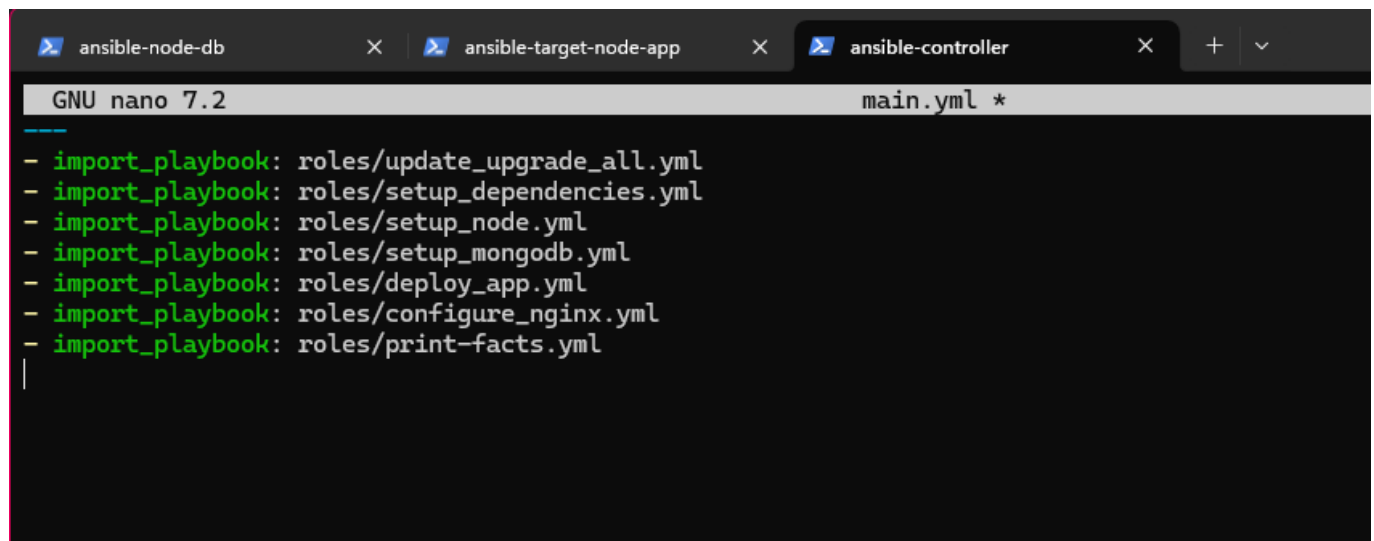


The screenshot shows a terminal window with three tabs: 'WSL Ubuntu', 'ansible-target-node-app', and 'ansible-controller'. The active tab is 'ansible-controller', which displays the output of a command to check the status of the nginx service. The output shows that the service is loaded, active, and running. It also displays the process details for the nginx service, including the master process and worker processes.

```
WSL Ubuntu ansible-target-node-app ansible-controller
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-03-10 03:32:35 UTC; 32min ago
     Docs: man:nginx(8)
  Process: 20281 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 20282 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 20284 (nginx)
    Tasks: 3 (limit: 1078)
   Memory: 2.4M (peak: 2.6M)
      CPU: 18ms
   CGroup: /system.slice/nginx.service
           └─20284 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─20285 "nginx: worker process"
               └─20286 "nginx: worker process"

Mar 10 03:32:35 ip-172-31-56-77 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy
>
Mar 10 03:32:35 ip-172-31-56-77 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy
s>

ubuntu@ip-172-31-56-77:~$
```



```
GNU nano 7.2 main.yml *
- import_playbook: roles/update_upgrade_all.yml
- import_playbook: roles/setup_dependencies.yml
- import_playbook: roles/setup_node.yml
- import_playbook: roles/setup_mongodb.yml
- import_playbook: roles/deploy_app.yml
- import_playbook: roles/configure_nginx.yml
- import_playbook: roles/print-facts.yml
|
```