Lab 1 – Ansible Setup

Task – Install ansible and run basic command

Prerequisite: An EC2 instance with ports 80, 22, open running Ubuntu 22.

1) Run the following commands to install ansible (or add these to a Linux script and run that)

```
sudo apt update
sudo apt install software-properties-common
sudo apt-add-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

2) Run the following command to locally, install nginx, and start the service. Once the service has started, curl localhost or access the public IP with your browser

```
ansible 127.0.0.1 -m apt -a "name=nginx state=present update_cache=true" --become
```

Stretch goal – Using Ansible command line, install unzip.

Lab 2 – Inventory setup

Task – Create an inventory and run a task

Prerequisites: Two EC2 instances (named Ansible and Host) with ports 80, 22, open running Ubuntu 22.
Ansible must be able to SSH connect to Host, so ensure an ssh key pair has been generated within Ansible and Host has the public key saved. Check you can connect to Host from Ansible with an ssh test (ssh -I <private key> ubuntu@<public ip>).

1) Within the Ansible EC2 create a file called inventory.yaml that contains the following (setting the IP, user and private key path):

```
all:
  children:
    demo:
      hosts:
        <IP of second machine>
      vars:
        ansible_user: ubuntu
        ansible_ssh_private_key_file=~/.ssh/ansible_id_rsa
```

2) Create a playbook.yaml in the same location and add the following:

```
- hosts: demo
  tasks:
  - name: "Ping"
    ping:
```

3) Run your ansible with the command

```
ansible-playbook -v -i inventory.yaml playbook.yaml
```

Stretch goal – Create a third EC2 and call it host_nginx. Modify the inventory.yaml to also have a group of hosts called **nginx**. In your playbook.yaml, add a new set of tasks to run the "Ping" task on the nginx host.

Lab 3 – Playbook Setup

Task – Creating a more complex playbook

Prerequisites: Ansible Host and 2 EC2 instances with ports 80, 22 open. Ansible host should be able to SSH connect to the other 2 instances.

Using Ansible inventory and playbook, **both** instances should have **wget** installed, **one** should have **unzip** and **other** should have **nginx**.

Use the Ansible docs for how to set this up using apt
https://docs.ansible.com/ansible-core/devel/index.html

Stretch goal – Add a task to uninstall unzip from all machines after it is installed on A.

Task 2 – Further Ansible tasks

Using Ansible setup the following environment:

- Install nginx on the host EC2 without nginx
- Replace the default HTML with a new HTML stored in the Ansible EC2
- Print out the JSON object of installing nginx
- Restart the nginx service
- Install Docker (Preferably the module way which will require researching)

Lab 4 – Ansible Variables

Task – Create a load balancer between two instances

Prerequisites: Ansible Host EC2, 2x EC2 instances with port 80 and 22 open, Ansible Host can ssh to the other instances.

Using a combination of Global vars, playbook vars, and handlers, create a new playbook that does the following tasks:

- Prints your name to the console
- Clones down a repo specified at command line
- Installs a package specified at command line (unzip if no specified package)
- Installs nginx and uses handlers to restart
- Creates a new file and appends text (command line chosen)

Lab 5 – Ansible roles

Task – Create Roles for Ansible Tasks

Prerequisites: Ansible Host with one EC2 instance (A) with ports 22 and 80 open. Ansible Host should be able to SSH into A and A should be a fresh instance.

Using ansible roles in a structure like below (role called nginx)

```
nginx/
|-- README.md
|-- tasks/
    |-- main.yaml
|-- handlers/
    |-- main.yaml
```

Create a role for the following tasks:

- Developer environment setup, create a group of users called ‘Devs’, add a new user with your name, install zsh shell and curl
- Install and setup Docker
- Install and setup Jenkins
- Using nginx as a webserver which hosts a custom.html

Use the main playbook.yaml to run the roles listed above, all roles can be run on the same host (A)