# Ansible Project (ansible handson-04 in US cohort)

## Part 1 - Build the Infrastructure

* Get to the AWS Console and spin-up 3 EC2 Instances with Ubuntu 22.04 AMI.
* Configure the security groups as shown below:
  + Controller Node ----> Port 22 SSH
  + Target Node1 -------> Port 22 SSH, Port 3306 MYSQL/Aurora
  + Target Node2 -------> Port 22 SSH, Port 80 HTTP

## Part 2 - Install Ansible on the Controller Node

* Run the terraform files in github repo.
* Connect to your Controller Node.
* Optionally you can connect to your instances using VS Code.
* Check Ansible's installation with the command below.

$ ansible --version

* Show and exlain the files (ansible.cfg, inventory.ini) that created by terraform.

## Part 3 - Pinging the Target Nodes

* Make a directory named ansible-lesson under the home directory and cd into it.

mkdir ansible-lesson

cd ansible-lesson

* Copy the phonebook app files (phonebook-app.py, requirements.txt, init.sql, templates) to the control node from your github repository.
* Do not forget to change db server private ip in phonebook-app.py. (app.config['MYSQL\_DATABASE\_HOST'] = "<db\_server private ip>")
* Create a file named ping-playbook.yml and paste the content below.

touch ping-playbook.yml

- name: ping them all

hosts: all

tasks:

- name: pinging

ansible.builtin.ping:

* Run the command below for pinging the servers.

ansible-playbook ping-playbook.yml

* Explain the output of the above command.

## Part4 - Install, Start, Enable Mysql and Run The Phonebook App.

* Create a playbook name db\_config.yml and configure db\_server.

- name: db configuration

become: true

hosts: db\_server

vars:

hostname: cw\_db\_server

db\_name: phonebook\_db

db\_table: phonebook

db\_user: remoteUser

db\_password: clarus1234

tasks:

- name: set hostname

ansible.builtin.shell: "hostnamectl set-hostname {{ hostname }}"

- name: Installing Mysql and dependencies

ansible.builtin.package:

name: "{{ item }}"

state: present

update\_cache: yes

loop:

- mysql-server

- mysql-client

- python3-mysqldb

- libmysqlclient-dev

- name: start and enable mysql service

ansible.builtin.service:

name: mysql

state: started

enabled: yes

- name: creating mysql user

community.mysql.mysql\_user:

name: "{{ db\_user }}"

password: "{{ db\_password }}"

priv: '\*.\*:ALL'

host: '%'

state: present

- name: copy the sql script

ansible.builtin.copy:

src: /home/ubuntu/ansible-lesson/phonebook/init.sql

dest: ~/

- name: creating phonebook\_db

community.mysql.mysql\_db:

name: "{{ db\_name }}"

state: present

- name: check if the database has the table

ansible.builtin.shell: |

echo "USE {{ db\_name }}; show tables like '{{ db\_table }}'; " | mysql

register: resultOfShowTables

- name: DEBUG

ansible.builtin.debug:

var: resultOfShowTables

- name: Import database table

community.mysql.mysql\_db:

name: "{{ db\_name }}" # This is the database schema name.

state: import # This module is not idempotent when the state property value is import.

target: ~/init.sql # This script creates the products table.

when: resultOfShowTables.stdout == "" # This line checks if the table is already imported. If so this task doesn't run.

- name: Enable remote login to mysql

ansible.builtin.lineinfile:

path: /etc/mysql/mysql.conf.d/mysqld.cnf

regexp: '^bind-address'

line: 'bind-address = 0.0.0.0'

backup: yes

notify:

- Restart mysql

handlers:

- name: Restart mysql

ansible.builtin.service:

name: mysql

state: restarted

* Explain what these tasks and modules.
* Run the playbook.

ansible-playbook db\_config.yml

* Open up a new Terminal or Window and connect to the db\_server instance and check if MYSQL is installed, started, and enabled.

mysql --version

* Or, you can do it with ad-hoc command.

ansible db\_server -m shell -a "mysql --version"

* Create another playbook name web\_config.yml and configure web\_server.

- name: web server configuration

hosts: web\_server

vars:

hostname: cw\_web\_server

tasks:

- name: set hostname

become: yes

ansible.builtin.shell: "hostnamectl set-hostname {{ hostname }}"

- name: Installing python for python app

become: yes

ansible.builtin.package:

name:

- python3

- python3-pip

state: present

update\_cache: yes

- name: copy the app file to the web server

ansible.builtin.copy:

src: /home/ubuntu/ansible-lesson/phonebook/phonebook-app.py

dest: ~/

- name: copy the requirements file to the web server

ansible.builtin.copy:

src: /home/ubuntu/ansible-lesson/phonebook/requirements.txt

dest: ~/

- name: copy the templates folder to the web server

ansible.builtin.copy:

src: /home/ubuntu/ansible-lesson/phonebook/templates

dest: ~/

- name: install dependencies from requirements file

become: yes

ansible.builtin.pip:

requirements: /home/ubuntu/requirements.txt

- name: run the app

become: yes

ansible.builtin.shell: "nohup python3 phonebook-app.py &"

* Explain what these tasks and modules.
* Run the playbook.

ansible-playbook web\_config.yml

* Check if you can see the website on your browser.