COMP 350 – Selected Topics – Introduction to DevOps

Project 2 (due date June 7th)

Hakan Ayral, Spring 2024

Notes: This The project can be done individually or as a team of 2. You may discuss the problems with other teams and post questions to the discussion forum, but the submitted work must be your own. Any material you use from external sources such as the internet should be properly cited in your report.

Contact TA: Javid Baydamirli jbaydamirli21@ku.edu.tr

Github Classroom link: https://classroom.github.com/a/nuHyAiLl

Description: You are expected to create a container environment using Docker Compose to demonstrate an Ansible configuration management setup. (This project is estimated to take 2-3 hours)

Requirements:

- Your Compose environment should consist of two images and three containers.
    1. The **1st image** will be for the **control node** for ansible; there will be **one container** launched from that image.
    2. The **2nd image** will be for the **managed nodes**; there will be **two containers** launched from that image.
- The image for the **control node** should contain:
    1. Python
    2. ssh client
    3. ansible (*installed and ready to run*)
    4. your ansible playbook
    5. your ansible inventory (*pointing to the other two containers for managed nodes*)
    6. your ansible configuration file (*properly setup to point to the inventory and ssh key*)
    7. ssh key pair (i.e. public + private)
    8. some kind of shell (*pick a base image with a shell or install your own*)
- The image for **managed nodes** should contain:
    1. Python
    2. ssh server (OpenSSH)
    3. ssh public key only
    4. some kind of shell (*pick a base image with a shell or install your own*)
- Port mapping
    1. Managed nodes should expose and map the ssh port to the outside
    2. Managed nodes should expose and map port 80 inside to port 80 outside eventhough they don't have a web server (yet! 😐).
- Managed nodes should have the ssh server running during their lifetime to allow ansible to connect from the control node.

- All three containers should have their shell as their main process and running at all times (*i.e.* ***docker attach*** *to the container should let you access the already running shell*)
    1. This means you should have the ssh server and the shell running at the same time on the managed nodes.
- Your **playbook** should have a **Play** which performs the following 3 tasks on each machine on the inventory (*this steps will be performed by ansible when play from the playbook is executed, you should not perform these in your docker file!*)
    1. Install **Nginx** through **apt**
    2. Create a **virtual host file** in `/etc/nginx/sites-enabled/` which serves a static web page (**index.html**) through port 80.
    3. Create an **index.html** file on the location you specified on the virtual host file above

Expected behaviour:

- After **git clone** 'ing your repo, it is expected to be able to use **docker compose up** on your project directory and get the 3 containers up and running.
- At this point the managed node containers should have the shell and ssh server running, but should not have Nginx or the virtual host file or the index file.
- The control node container should only have the shell running.
- User should be able to docker attach to the control node and access the already running shell.
- The user should be able to execute the play in the ansible playbook against all inventory.
- After execution both managed nodes should have Nginx installed, running, and serving the html file on port 80 accessible form outside of the container.

GitHub submissions: We will be using GitHub classroom for this assignment. You must push your work to your assigned GitHub repository. We will be checking the commits throughout the course of the project and during evaluation. Note that you still need to upload a zip file to Blackboard. (link: https://classroom.github.com/a/nuHyAiLl)

Deliverables:

- You are expected to upload your complete project (except any compiled binaries that can be reproduced) as a zip file to Blackboard before the deadline. It should be ready to be launched by just typing **`docker compose up`**.
    - Before submitting your project clone your project to a clean location and try **`docker compose up`** yourself on your project; one project 1 there were many submitted projects which doesn't work when cloned form scratch! (*Your project working on your local does not mean that it will work when freshly cloned.*)
- You must push your work to GitHub classroom in addition to Blackboard submission; the TA and I will be checking the commits throughout the and during project evaluation. You should keep your GitHub repo updated from the start to the end of the project.