

# ENSF 400 - Assignment 2 - Ansible

Feb 27, 2024

This assignment has a full mark of 100. It takes up 5% of your final grade.

## 1 Using Ansible to Set Up a Load Balancer

Extend your work for Lab 5 and add `nginx` as a load balancer in front of the 3 NodeJS servers so that a request will come in through the load balancer's port and be processed by either of the 1 backend services. There are three ways of creating the load balancer service:

1. Create a `Dockerfile` and build an image for Nginx. Mount the Nginx configuration file.
2. Use an existing Nginx image and mount the Nginx configuration file.
3. Use Ansible to install Nginx service on “localhost”.

You may use either of the three ways above to create your load balancer. Additional hints can be found at <https://github.com/denoslab/ensf400-lab5-ansible/blob/main/assignment2/README.md>.

Deliver the following:

- `Dockerfile` of the base image with Alpine Linux for the container ‘loadbalancer’, or using a public, pre-built Nginx container image. Alternatively, Nginx can be installed on “localhost” using Ansible. (5%)
- Nginx config files, which will be mounted to the container `loadbalancer` for setting up the configurations. Alternatively, the config file can be copied to the directory storing Nginx configurations. (10%)
- Updated `docker-compose.yml` with the additional nginx load balancer. Alternatively, demonstrate that Nginx is successfully installed on “localhost” by showing the status of the Nginx service. (5%)
- Updated inventory file including the host `loadbalancer`. When “localhost” is used for installing Nginx, include “localhost” in the `loadbalancer` group. (5%)
- Updated `hello.yml` playbook to include the play for installing the nginx load balancer on host `loadbalancer`. (25%)

## 2 Ansible Python API using ansible-runner

Use `ansible-runner` to translate the steps in Lab 5 into Python code, achieving the same results. Ansible Python API For the Python API, we should use the Python `ansible-runner` library.

## 2.1 References

Using Runner as a Python Module Interface to Ansible Python Library Installation - [https://ansible.readthedocs.io/projects/runner/en/latest/python\\_interface/](https://ansible.readthedocs.io/projects/runner/en/latest/python_interface/)

Complete the following two tasks:

- Load inventory using Python code and print the names, IP addresses, and group names of all hosts. Ping all these hosts and output the results. (20%)
- Run the `hello.yml` playbook using Python code to install the app on the group `app_group`. Use the existing playbook file from the lab and pass it to Python code to run the playbook. Print the results. Read and verify the response of the NodeJS servers from your python code. (30%)

Along with the extended project of Lab 5 based on the previous question, deliver the following:

- A Python file `load_inventory.py` for the first task.
- A Python file `run_playbook.py` for the second task.

## 3 Submission

Use your completed Lab 5 as the base project, complete the two sections above. Include all updated files and directories in the directory of the base project. Compress the project directory to a zip file with the name `ensf400.a2.zip` and submit through D2L.