**NOS125 Final Project:**

## **Summary:**

This playbook was designed for efficiency and reusability. To run a playbook, you need three key files: an inventory file, a configuration (.cfg) file, and the playbook itself in a YAML format (.yml). The inventory file lists all the hosts, which can be grouped together for easier management. For example, servera.lab.example.com and serverb.lab.example.com are grouped as "webservers," allowing the playbook to target both servers simultaneously.

The .cfg file configures how the playbooks behave within the Ansible project directory. Inside the vars directory of the FinalExam folder, there are three .yml files that store the variables required for the playbook. Keeping these variables in separate files makes them easier to update and allows other developers to use them without creating new variable files, which saves time. The variables can also be referenced in loops within the playbook, reducing the need for repetitive tasks. The first file, group\_vars.yml, defines the group names for the playbook. The services\_vars.yml file includes two variables: packages (the packages to be installed) and services (the services to be enabled and started on the web servers). The users\_vars.yml file lists users and their corresponding groups.

The playbook itself, playbook.yml, is the main file that executes a series of tasks. It performs four main actions: installing packages, starting and enabling services, creating user groups, and managing users. At the beginning of the playbook, a name is assigned, the target hosts are specified, root privileges are granted, and the variable files are imported. The first task installs the packages defined in the package variable on the specified hosts using a loop. If any package fails to install, a rescue block displays an error message.

The next task enables and starts the services listed in the services variable, also using a loop. This setup allows developers to install certain packages without enabling all of them. The ignore\_errors option is used here to manage errors; if a service fails to start, Ansible will skip the error and proceed to the next task. The third task creates the groups specified in group\_vars.yml and checks their existence. In this case, ignore\_errors is not used, as the success of this task is crucial for the next one. Finally, the last task creates users and assigns them to their respective groups, again using a loop and the users\_vars.yml file. To run the playbook, use the command ansible-navigator run -m stdout playbook.yml. If the playbook runs successfully, the output will display only green or brown. Any red indicates that a task has failed, and the playbook will stop.

## **Directory layout:**

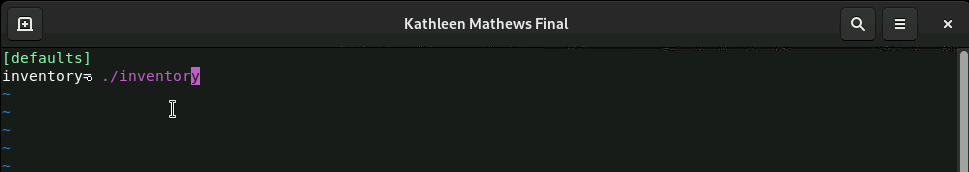
**A screenshot of a computer

Description automatically generated**

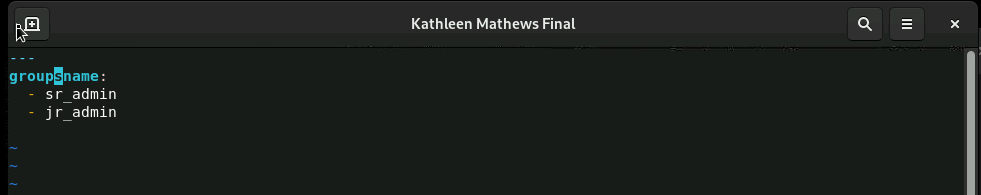
## **inventory:**

****

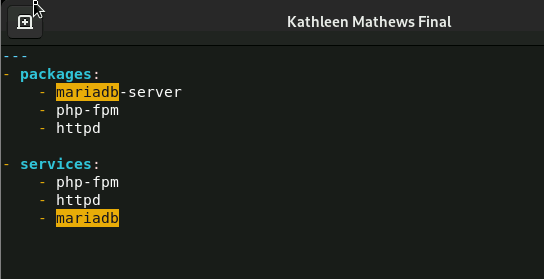
## **ansible.cfg:**

****

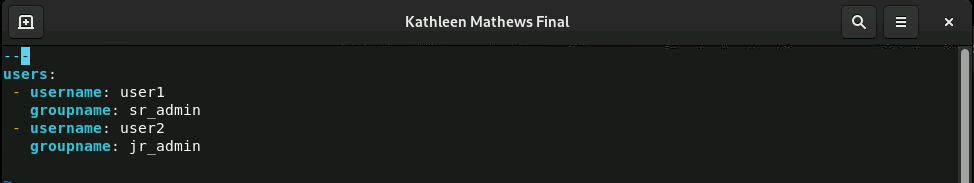
## **~/FinalExam/vars/groups\_vars.yml**

****

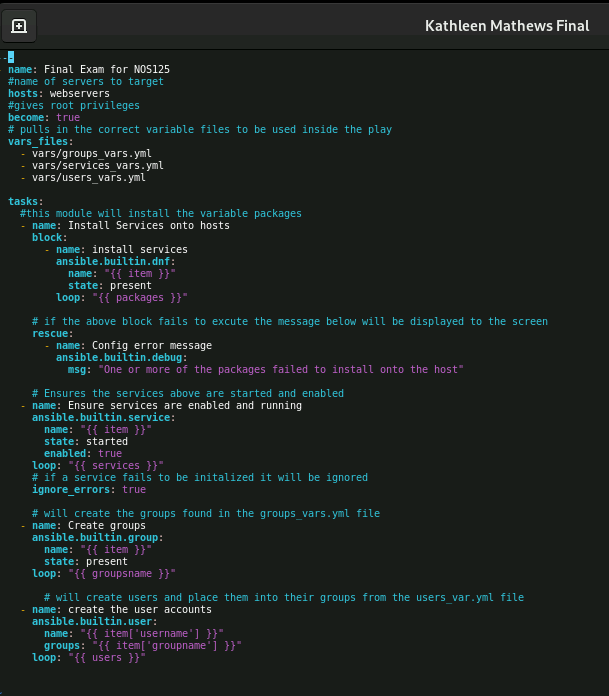
## **~/FinalExam/vars/services\_vars.yml**

****

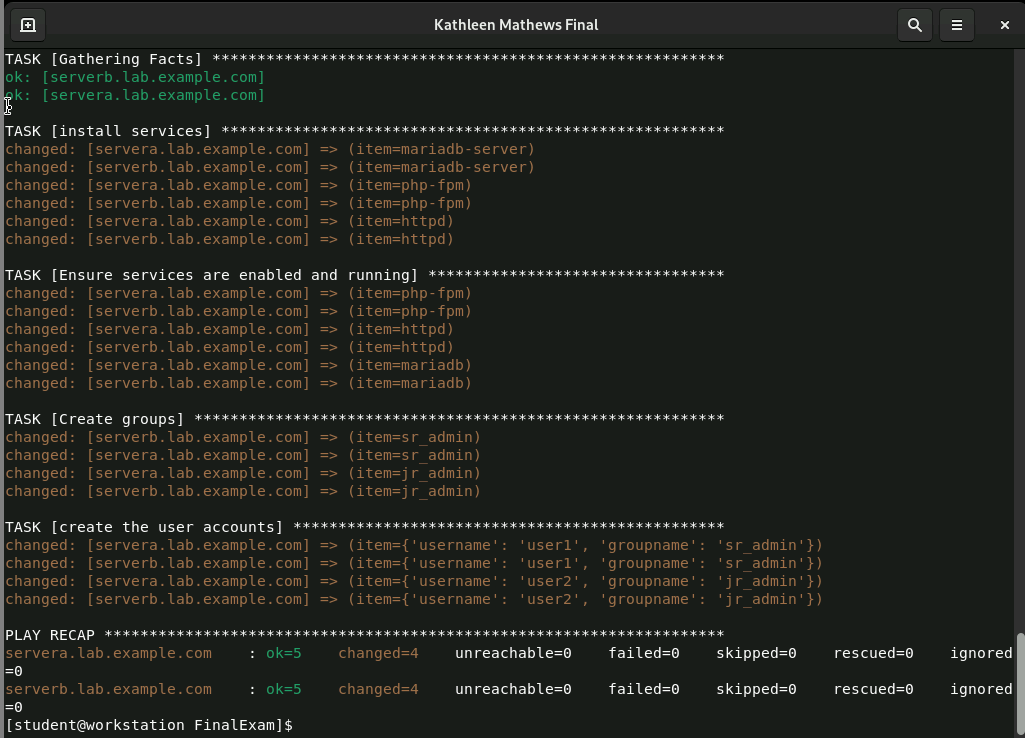
## **~/FinalExam/vars/users\_vars.yml**

****

## **Playbook.yml**

****

## **Successful Execution of the Playbook on webservers:**

****