Final Assignment:
Infrastructure as Code or Containerizing Apps

OSYS1000, Winter 2024
NSCC Lunenburg Campus

___/20

Note: This assignment provides 2 different options to complete – both of the same value. Complete the option of your choice –you are not expected to complete both!

Note: Both options require you to create a web page and serve it over Apache HTTP. This does not need to be a fancy web page, any basic yet custom page will do as this is not a web design course.

Option 1: Linux Scripting Project

Objective:

Develop an idempotent BASH script to automate user management, group creation, directory structure setup, web server installation, and FTP server installation on a Linux system. Submit the entire script (or preferably, provide a GitHub link).

Tasks:

1. **User Management:**
    - Create users: Michael, Dwight, Jim, Phyllis, Andy, Stanley, Pam, Kevin, Oscar, Angela, Meredith, Creed, Kelly, and Toby.
2. **Group Creation:**
    - Create groups: manager, accounting, sales, support, hr.
3. **Group Assignments:**
    - Assign users to groups:
        - Michael to manager group
        - Dwight, Jim, Phyllis, Andy, Stanley to sales
        - Kevin, Oscar, Angela to accounting
        - Pam, Meredith, Creed to support
        - Kelly and Toby to HR
4. **Directory Structure Setup:**
    - Create a directory structure allowing each group to have separate directories.

- Set permissions to allow group members to add/modify files within their respective directories, while restricting access for others.
- Ensure Michael and Toby have access to all files.

5. **Web Server Installation:**
    - Install a web server (Apache HTTP, not Nginx) and ensure it's running.
    - Serve a basic web page on port 80.
6. **FTP Server Installation:**
    - Install an FTP server and ensure it's running.

Rubric:

Criteria	Excellent (5)	Good (3)	Needs Improvement (1)
User Management			
Creates all specified users	All users created with correct attributes.	All users created, minor attribute discrepancies.	Not all users created or significant discrepancies.
Group Creation			
Creates all specified groups	All groups created accurately.	All groups created, minor discrepancies.	Not all groups created or significant discrepancies.
Group Assignments			
Assigns users to correct groups	All users assigned to correct groups without error.	All users assigned, minor errors in assignment.	Incorrect user-group assignments or significant errors.
Directory Structure Setup			
Sets up directory structure with correct permissions	Directories created with appropriate permissions and ownership.	Directories created, minor issues with permissions or ownership.	Incorrect permissions or ownership, or directories not properly created.
Web Server Installation			
Installs and configures web server	Web server installed, running, and serving basic webpage correctly.	Web server installed and running, minor configuration issues.	Web server not installed or significant configuration errors.
FTP Server Installation			
Installs and configures FTP server	FTP server installed, running, and accessible.	FTP server installed and running, minor configuration issues.	FTP server not installed or significant configuration errors.

Script Quality			
Script is well-documented, organized, and efficient	Code is clear, concise, and well-commented. Follows best practices.	Code is mostly clear, with some comments and organization.	Code lacks clarity, comments, or organization.
Bonus (Optional)			
Implements additional functionality	Additional features implemented with excellence.	Bonus features added, minor issues or limitations.	Bonus features attempted but not fully functional or significant issues.

Note:

- Students may specify the distribution of Linux this script should run on (e.g., RHEL, Ubuntu). Ensure compatibility and specify any distribution-specific commands or configurations in the script.
- Ensure the script is well-documented and includes comments explaining the purpose of each section and any non-obvious commands.

Option 2: Containerized Web Application

Objective:

Develop a containerized application using Apache HTTP to serve a basic website on port 8080. The output of this assignment should include a GitHub repository containing a Dockerfile, a docker-compose.yml file for developer workflow, and the Docker image should be built and uploaded to Docker Hub for public consumption. Please provide the GitHub repo URL and the DockerHub URL.

Tasks:

1. **GitHub Repository Setup:**
    - Create a GitHub repository to host the project.
    - Include a Dockerfile for building the Apache HTTP server container.
    - Ensure the repository is well-structured with clear documentation.
2. **Dockerfile Creation:**
    - Develop a Dockerfile that sets up Apache HTTP server within a container.
    - Configure the Dockerfile to copy the website files into the appropriate directory within the container.
    - Expose port 8080 for accessing the website.
3. **docker-compose.yml File:**
    - Create a docker-compose.yml file to facilitate the developer workflow.
    - Define the service for the Apache HTTP server.
    - Configure volume mounts for local development.
4. **Image Building and Pushing:**
    - Build the Docker image locally using the Dockerfile.
    - Tag the built image appropriately for Docker Hub.
    - Push the image to Docker Hub for public consumption.

Rubric:

Criteria	Excellent (5)	Good (3)	Needs Improvement (1)
GitHub Repository			
Repository structure and	Well-structured repository with clear	Repository structure and documentation	Repository structure or documentation

documentation	documentation and README.	present, minor improvements needed.	lacks clarity or completeness.
Dockerfile			
Dockerfile completeness and correctness	Dockerfile sets up Apache HTTP server correctly within a container.	Dockerfile is functional but may have minor issues.	Dockerfile has significant issues or functionality is incomplete.
docker-compose.yml			
docker-compose.yml completeness and correctness	docker-compose.yml facilitates local development effectively.	docker-compose.yml is functional but may have minor issues.	docker-compose.yml has significant issues or functionality is incomplete.
Image Building and Pushing			
Image built and pushed to Docker Hub	Docker image built successfully, tagged appropriately, and pushed to Docker Hub.	Docker image built and tagged correctly but may have minor issues with pushing.	Docker image not built or tagged correctly, or significant issues with pushing.
Documentation			
Clear instructions and usage documentation	Clear instructions provided for building, running, and accessing the web server.	Instructions provided but may lack clarity or completeness.	Instructions are unclear or incomplete.
Bonus (Optional)			
Additional features or improvements	Additional features or improvements implemented with excellence.	Bonus features added, minor issues or limitations.	Bonus features attempted but not fully functional or significant issues.

Note:

- Ensure the Dockerfile and docker-compose.yml files are well-commented to explain each step and configuration.
- Test the Docker image and docker-compose setup thoroughly to ensure the web server functions as expected.
- Provide clear instructions for developers on how to use the Docker image and docker-compose setup in the README file.
- Consider best practices for Docker image optimization and security.