Lab Instructions: Deploy a Web Server Using Fabric

Objective:

Automate the deployment of an Nginx-based web server and create an index.html file directly on the remote machine.

Prerequisites

1. Install Python (Ignore if Already Done)

Ensure Python 3.6 or later is installed.
Verify installation using:

python3 --version

2. Install fabric (Ignore if Already Done)

Fabric uses Paramiko for SSH automation. Install it using:

pip install fabric

3. Create a Virtual Environment (If Not Already Created)

To isolate your project, create a virtual environment:

python3 -m venv venv

Activate the virtual environment:

source venv/bin/activate # For Linux/macOS

venv\Scripts\activate # For Windows

Step 2: Create the Fabfile

1. Remove Any Previous Fabfile

Ensure no previous fabfile exists by removing it:

rm fabfile.py

2. Create a New Fabfile

Use the nano editor to create a fabfile:

nano fabfile.py

Add the following Python script to automate the deployment:

```
from fabric import task

@task
def install_nginx(c):
    """Installs Nginx on the remote server."""
    c.run("sudo apt update -y")
    c.run("sudo apt install -y nginx")
    print(" Nginx installed successfully.")

@task
def create_html(c):
    """Creates a sample index.html file directly on the remote server."""
    html_content = "<h1>Hi There, this is Fabric. Deployed using Fabric!</h1>"
    c.run(f'echo "{html_content}" | sudo tee /var/www/html/index.html')
    print(" HTML file created successfully.")

@task
def configure_firewall(c):
    """Configures firewall rules to allow web traffic."""
    c.run("sudo ufw allow 'Nginx Full'")
    print("Firewall configured successfully.")

@task
def restart_nginx(c):
    """Restarts the Nginx service."""
    c.run("sudo systemctl restart nginx")
    print("Nginx restarted successfully.")

@task
def deploy(c):
    """Full deployment: Install Nginx, create HTML file, configure firewall, restart Nginx."""
    install_nginx(c)
    create_html(c)
    configure_firewall(c)
    restart_nginx(c)
    print(f" Deployment complete! Visit http://{REMOTE_HOST}/")
```

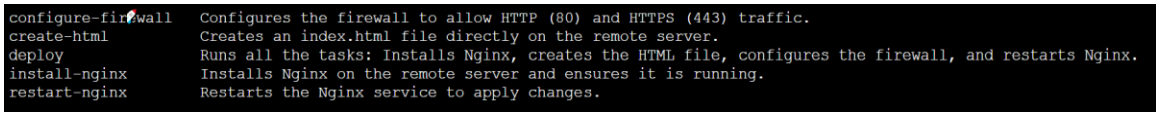
Press **Ctrl+O** and then press **Enter** to save the file and click **Ctrl+X** to close the file.

Step 3: List Tasks in Fabfile

Use the fab command to list available tasks in the fabfile.py:

fab --list

Expected Output: You should see the following tasks listed:



Step 4: Execute Tasks

1. Run the full deployment task:

   fab -H myserver --prompt-for-login-password deploy

2. Replace **myserver** with your remote machine IP address.

3. When it prompts for password, please **enter the password** of your remote_user.

Step 5: Verify the Output

1. Open a new tab in a browser and browse the following URL:

   http://{REMOTE_HOST}

2. Replace {REMOTE_HOST}, with your remote machine IP address.

3. You will see the following output in the browser:

   Hi There, this is Fabric. Deployed using Fabric!

Step 6: Verify Virtual Environment

Ensure the virtual environment is active. The prompt should include (venv).

To deactivate when done, run:

deactivate