# Lab Instructions: Automating FTP Operations using pexpect

This document provides step-by-step instructions to automate FTP operations using the `pexpect` Python module.

# Setting Up an FTP Server on Ubuntu- On your Remote machine

## 1. Update Your System

1. Before setting up the FTP server, update the package list on your Ubuntu system to ensure you have the latest updates.
2. Run the following commands:

**sudo apt update  
sudo apt upgrade -y**

## 2. Install vsftpd

1. `vsftpd` (Very Secure FTP Daemon) is a popular and lightweight FTP server for Linux. Install it using the following command:

**sudo apt install vsftpd -y**

## 3. Backup the Configuration File

1. Before making changes, back up the default `vsftpd` configuration file:

**sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bak**

## 4. Configure vsftpd

1. Open the `vsftpd` configuration file in a text editor:

**sudo nano /etc/vsftpd.conf**

1. Modify the following settings:
   * Allow local users to log in:  
      Ensure the line below is uncommented:  
      **local\_enable=YES**
   * Enable file upload and write permissions:  
      Uncomment or add the following line:  
      **write\_enable=YES**
   * Restrict users to their home directories:  
      Uncomment or add the following line:  
      **chroot\_local\_user=YES**
2. Save and exit the file **(Press `CTRL+O`, then `Enter`, and `CTRL+X`).**

## 5. Restart the FTP Service

1. After making changes, restart the `vsftpd` service to apply the configuration:

**sudo systemctl restart vsftpd**

1. Enable the service to start on boot:

**sudo systemctl enable vsftpd**

## 6. Create an FTP User

1. Add a new user account for FTP access:

**sudo adduser ftpuser**

1. Follow the prompts to set a password and user details.
2. Assign the user a home directory (optional):

**sudo mkdir -p /home/ftpuser/ftp/files  
sudo chown -R ftpuser:ftpuser /home/ftpuser/ftp/files**

1. The `/ftp/files` directory is where the FTP user will upload/download files.
2. Restrict the user to their home directory (if not already enabled in the config):

**sudo usermod -d /home/ftpuser/ftp ftpuser**

## 7. Adjust Firewall Rules

1. If you have a firewall enabled, allow FTP traffic using the following commands:

**sudo ufw allow 20/tcp  
sudo ufw allow 21/tcp  
sudo ufw reload**

## 8. Test the FTP Server

1. Connect to the FTP server using the command-line of your **Local Machine.**
2. From your local machine, run:
3. ftp <your-server-ip>
4. Enter the username (`ftpuser`) and password created earlier.

## Step 9: Python Script for FTP Automation( execute the following script in your Local Machine’s virtual environment)

Use the Python script below to automate FTP operations using the `pexpect` module:

import pexpect  
  
def ftp\_automation(host, username, password, commands):  
 try:  
 # Spawn the FTP session  
 ftp\_command = f"ftp {host}"  
 print(f"Connecting to FTP server: {host}")  
 child = pexpect.spawn(ftp\_command, timeout=30)  
   
 # Handle login prompts  
 child.expect("Name .\*:") # Match the prompt for username  
 child.sendline(username)  
 child.expect("Password:") # Match the prompt for password  
 child.sendline(password)  
   
 # Check if login was successful  
 index = child.expect(["ftp>", "Login incorrect", pexpect.EOF, pexpect.TIMEOUT])  
 if index == 1:  
 print("Login failed. Please check your username or password.")  
 return  
 elif index in [2, 3]:  
 print("Error: Connection failed.")  
 return  
   
 print("Login successful. Executing commands...")  
   
 # Execute each command  
 for command in commands:  
 print(f"Executing: {command}")  
 child.sendline(command)  
 child.expect("ftp>")  
 output = child.before.decode()  
 print(f"Output:\n{output}")  
   
 # Close the FTP session  
 child.sendline("bye")  
 child.expect(pexpect.EOF)  
 print("FTP session closed.")  
   
 except pexpect.exceptions.TIMEOUT:  
 print("Error: Operation timed out.")  
 except Exception as e:  
 print(f"An error occurred: {e}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Replace with your FTP server details and commands  
 ftp\_host = "ftp\_host\_ip\_address"  
 ftp\_username = "username"  
 ftp\_password = "Password"  
   
 # List of commands to execute  
 ftp\_commands = [  
 "ls", # List files in the current directory  
 "cd files", # Change to the 'files' directory  
 "put /home/rps/Downloads/samplefile.txt", # Upload a file  
 "get /home/ftpuser/ftp/files/ftpfile.txt", # Download a file  
 "pwd", # Print working directory  
 ]  
   
 # Automate the FTP operations  
 ftp\_automation(ftp\_host, ftp\_username, ftp\_password, ftp\_commands)

## Notes

- Ensure that the `pexpect` library is installed in your Python environment. Install it using:

```bash  
 pip install pexpect  
 ```

- Replace placeholders in the script (e.g., `ftp\_host`, `ftp\_username`, `ftp\_password`) with your FTP server details.