# Lab Instructions: File Transfer Automation Using Python Paramiko and SFTP

## Step 1: Prerequisites

1. Install Python:

- Ensure Python (3.6 or later) is installed on your system.

- Verify the installation:

python3 --version

2. Install Paramiko:

- Paramiko is required for SFTP file transfers. Install it using pip:

pip install paramiko

3. Create a Virtual Environment:

- Use a Python virtual environment to isolate your project.

python3 -m venv venv

- Activate the virtual environment:

source venv/bin/activate # For Linux/macOS

.\venv\Scripts\activate # For Windows

- Install Paramiko within the virtual environment:

pip install paramiko

## Step 2: Create the Python Script

1. Create a Python script:

- Create a file named sftp\_transfer.py in your working directory and add the following content:

**import paramiko  
  
def sftp\_file\_transfer\_with\_component(hostname, port, username, password, local\_file\_path, remote\_file\_path, mode="upload"):  
 try:  
 # Initialize Transport object  
 print(f"Connecting to {hostname} via SFTP...")  
 transport = paramiko.Transport((hostname, port))  
   
 # Authenticate with username and password  
 transport.connect(username=username, password=password)  
 print(f"Connected to {hostname}!")  
  
 # Initialize the SFTP client  
 sftp = paramiko.SFTPClient.from\_transport(transport)  
   
 if mode == "upload":  
 # Upload the file  
 print(f"Uploading {local\_file\_path} to {remote\_file\_path}...")  
 sftp.put(local\_file\_path, remote\_file\_path)  
 print("File uploaded successfully.")  
 elif mode == "download":  
 # Download the file  
 print(f"Downloading {remote\_file\_path} to {local\_file\_path}...")  
 sftp.get(remote\_file\_path, local\_file\_path)  
 print("File downloaded successfully.")  
 else:  
 print("Invalid mode! Use 'upload' or 'download'.")  
  
 except Exception as e:  
 print(f"An error occurred: {e}")  
 finally:  
 # Close the SFTP session and Transport connection  
 if 'sftp' in locals():  
 sftp.close()  
 print("SFTP session closed.")  
 if 'transport' in locals():  
 transport.close()  
 print("SFTP connection closed.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Replace these with your remote host details  
 hostname = "192.168.1.166" # Remote host's IP or domain  
 port = 22 # Default SFTP/SSH port  
 username = "rps" # Your username  
 password = "rps@123" # Your password  
  
 # File paths  
 local\_file\_path = "/home/rps/samplefile.txt" # Local file path  
 remote\_file\_path = "/home/rps/secondfiletransferred.txt" # Remote file path  
  
 # Choose operation mode: "upload" or "download"  
 mode = "upload" # Change to "download" to fetch files from the remote server  
  
 # Automate file transfer using SFTP component  
 sftp\_file\_transfer\_with\_component(hostname, port, username, password, local\_file\_path, remote\_file\_path, mode)**

## Step 3: Run the Script

1. Execute the script:

- Run the script using the following command:

python sftp\_transfer.py

2. Choose Operation Mode:

- Ensure the `mode` variable in the script is set to either 'upload' or 'download'.

- If set to 'upload', the file specified in `local\_file\_path` will be uploaded to the path specified in `remote\_file\_path`.

- If set to 'download', the file specified in `remote\_file\_path` will be downloaded to the path specified in `local\_file\_path`.

## Step 4: Verify the File Transfer

1. Verify the transfer on the remote server:

- Log in to the remote server and check if the file exists at the specified path.

2. Verify the transfer on the local machine:

- Check if the file exists at the specified path on your local machine.

## Step 5: Verify Virtual Environment

1. Check Active Virtual Environment:

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

2. Deactivate When Done:

- Deactivate the virtual environment to exit:

deactivate