Lab Instructions: File Transfer Automation Using Python Paramiko and SFTP

1. Ensure that you are in your python virtual environment, it it is not active, use the following command to activate it:

   source my_python_env/bin/activate

Step 1: Prerequisites

1. Install Python **(ignore if it is already done):**

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

- Verify the installation:

  python3 --version

2. Install Paramiko **(ignore if this already done):**

- Paramiko is required for SSH automation. Install it using pip:

  pip install paramiko

3. Create a Virtual Environment **if it is not created** :

- Use a Python virtual environment to isolate your project.

  python3 -m venv venv

- Activate the virtual environment:

  source venv/bin/activate # For Linux/macOS

- Install Paramiko within the virtual environment:

  pip install paramiko

Step 2: Create a sample file called sftpupload.txt in your current working directory.

1. In this step, we will create a text file called sftpupload.txt in our local machine. To do so, execute the following command:

   nano sftpupload.txt

   Enter the following text, **This is a file using for SFTP.**

2. Then press **Ctrl+O** to save the changes, then press **Enter**, then **Ctrl+X** to close the nano editor.

3. Now, lets verify if we have created the localfile.txt successfully, to do so, please enter the following command:

   ls

   Here, you can locate **sftpupload.txt** file which we created.

4. Copy the path of that file and paste it in a text editor (we will use it in the later steps of the lab)

Step 2: Create the Python Script

1. Now, lets create a python script with name **'Automating_File_Transfers_using_SFTP_in_Paramiko_using_SFTP_Client.py' in our present working directory** using nano editor. To do so, please enter the following command:

   nano Automating_File_Transfers_using_SFTP_in_Paramiko_using_SFTP_Client.py

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)
```

- Replace placeholders in the script (e.g., **`hostname`, 'username', `password',** **local_file_path** with your sftpupload.txt file path and **remote_file_path** with the 'remotepath') .

Then press **Ctrl+O** to save the changes, then press **Enter**, then **Ctrl+X** to close the nano editor.

Step 3: Run the Script

1. Execute the script:

- Run the script using the following command:

python **Automating_File_Transfers_using_SFTP_in_Paramiko_using_SFTP_Client.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