Installation Lab Setup

This document provides the steps and a script to create a Python virtual environment, install system updates, ensure the required Python modules are installed, and install the necessary libraries (`pexpect`, `paramiko`, and `fabric`).

Prerequisites

Ensure the following prerequisites are met before executing the script:

- A system running a Debian-based Linux distribution (e.g., Ubuntu).

- Sudo privileges for installing updates and packages.

By leveraging both the remote and local servers, you ensure a balance between centralized control and distributed processing. This setup improves performance, security, reliability, and flexibility of your O&M library.

Part 1: Checking the connection between local host VM and Remote host VM.

1. From the 2 VMs provided, choose 1 VM as your local host, and the other as remote host (Ex: U03 (Remote host), U04 (local host).
2. In your local host machine, open terminal and enter the following command.

   ```
   Ifconfig
   ```
3. Now, you will be able to see **ip address** of the local machine (copy it into your notepad/text editor, we will use it in our upcoming lab activities)
4. Next, enter the following command to get the username of your local machine.

   ```
   whoami
   ```
5. Similarly, repeat the same process in your 'remote machine' and copy the 'IP address and hostname' of the remote machine into a text editor.

6. From your local machine, in the terminal, enter the following command:

   ```
   ping <your-remote-machine-ip-address>
   ```

   You will be able to see packets being delivered, which means your local machine is able to communicate with your remote machine.

7. Now, from your local machine, we will login into your remote host machine using SSH (secure shell). To do so, please enter the following command in the terminal of your local machine:

   ```
   ssh<remote_machine_usrname>@<remote_machine_ip_address>
   ```

8. Now the terminal prompts you to enter the password to enter.
   Enter the password: **rps@123** (since this is terminal the password will not be visible)
9. Enter **yes** if it is asking for confirmation prompts to login.
10. To verify whether you logged in successfully, type ‘**ifconfig’ ,** you will be able to see the IP address of your remote machine.
11. Now, let's close the connection, to do so, please enter the following command:

    exit

    You will see the output stating that **‘Connection closed’**

12. Now, you are in your local host, to verify type the following command:

    ifconfig

    You will see the IP address of the local host.

Part 2: Installing the required libraries using shell script.

1. In your local host machine, create and save the shell script provided below to a file, e.g., `setup_env.sh`.

2. To create a shell script, lets create using **nano** editor, to do so, please type the following command in your terminal:

nano setup_env.sh

The above command will create and open a file called setup_env.sh

3. Copy paste the following script into the setup_env.sh file.

Shell Script

Below is the shell script to create a Python virtual environment, install all updates, ensure the `venv` module is installed, and install the required libraries:

```
#!/bin/bash

# Define the virtual environment name
VENV_NAME="my_python_env"

# Update system packages
echo "Updating system packages..."
sudo apt update && sudo apt upgrade -y

# Install Python3 and venv module if not already installed
echo "Installing Python3 and venv module if not installed..."
sudo apt install -y python3 python3-venv
```

```
# Check if Python3 is installed
if ! command -v python3 &> /dev/null
then
   echo "Python3 is not installed. Please install Python3 before running this script."
   exit 1
fi

# Check if venv module is available
if ! python3 -m ensurepip --version &> /dev/null
then
   echo "Python3 venv module is not available. Please install it before running this script."
   exit 1
fi

# Create the virtual environment
if [ ! -d "$VENV_NAME" ]; then
   echo "Creating virtual environment: $VENV_NAME"
   python3 -m venv $VENV_NAME
else
   echo "Virtual environment $VENV_NAME already exists."
fi

# Activate the virtual environment
source $VENV_NAME/bin/activate

# Upgrade pip
echo "Upgrading pip..."
pip install --upgrade pip

# Install required Python libraries
echo "Installing required libraries: pexpect, paramiko, fabric"
pip install pexpect paramiko fabric asyncio nest_asyncio

# Deactivate the virtual environment
echo "Deactivating virtual environment..."
deactivate

# Confirmation message
echo "Setup complete. Virtual environment $VENV_NAME is ready with the required libraries installed."
```

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

2. Grant execution permissions to the script using the following command:

```bash
chmod +x setup_env.sh
```

3. Execute the script with elevated privileges:

```bash
sudo ./setup_env.sh
```

Activating the Python Virtual Environment

To activate the virtual environment, use the following command:

```bash
source my_python_env/bin/activate
```

Once activated, you can use the Python environment to run your projects and scripts. To deactivate, simply use the `deactivate` command.