

Forum:[Hack Forums](#)

Thread: [Opening a Terminal on Your Host Machine from an Ubuntu VM Using Python](#)

Clumsy - July 14, 2024, 2:14 pm

When working within an Ubuntu virtual machine (VM), there might be instances where you need to execute a command on your host machine. This guide provides a step-by-step approach to achieve this using Python.

Opening a Terminal on the Host Machine from Within the VM

First, we will explore a straightforward method using Python's subprocess module to open a terminal within the same machine. This method works seamlessly if you are operating entirely within your VM.

Here's a simple script to open a GNOME Terminal.

Code import subprocess

```
# The command to execute
```

```
cmd = "gnome-terminal" # This assumes you are using GNOME Desktop Environment. Replace it with the appropriate command for your DE.
```

```
# Run the command using subprocess
```

```
subprocess.run(cmd, shell=True)
```

Challenges with VM and Host Communication

Running the above script within an Ubuntu VM might not open a terminal on the host machine due to the separation between the host and the VM. To bridge this gap, we can use SSH (Secure Shell) to remotely execute commands on the host machine.

Using Paramiko for SSH

To execute commands on the host machine from within the VM, we will use the Paramiko library, which allows us to establish an SSH connection and run commands remotely.

First, install the Paramiko library using pip.

Code pip install paramiko

Next, use the following Python script to open a terminal on your host machine.

Code import paramiko

```
# SSH credentials
```

```
ssh_user = "your_username"
```

```
ssh_pass = "your_password"
```

```
ssh_host = "your_host_ip"
```

```
# Create an SSH client and connect to the host
```

```
client = paramiko.SSHClient()
```

```
client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
client.connect(ssh_host, username=ssh_user, password=ssh_pass)
```

```
# Open a shell
chan = client.get_transport().open_session()
chan.exec_command("gnome-terminal") # Adjust the command for your DE if needed.
```

Explanation

Using Subprocess

We use the subprocess module to execute the command gnome-terminal.

This script is straightforward but limited to the local environment where it's executed.

Using SSH with Paramiko

We install the Paramiko library to facilitate SSH connections.

The script connects to the host machine using the provided SSH credentials.

Once connected, it executes the command to open a GNOME Terminal on the host machine.

This approach ensures that you can control and execute commands on your host machine from within your Ubuntu VM, enabling a more integrated and flexible workflow.

Disclaimer

The scripts and instructions provided in this article are intended for educational and informational purposes only. The execution of commands on remote systems via SSH should only be done with proper authorization. Unauthorized access to computer systems is illegal and unethical.

The user is responsible for ensuring compliance with all applicable laws and regulations regarding the use of SSH and remote command execution. The author and publisher are not responsible for any misuse, damage, or legal consequences resulting from the use of the information provided in this article.

Use these techniques at your own risk and ensure that you have the necessary permissions to access and execute commands on any remote systems you interact with.