Debugging Python Scripts

1 hour 30 minutesFree

Introduction

Imagine one of your colleagues has written a Python script that's failing to run correctly. They're asking for your help to debug it. In this lab, you'll look into why the script is crashing and apply the problem-solving steps that we've already learned to get information, find the root cause, and remediate the problem.

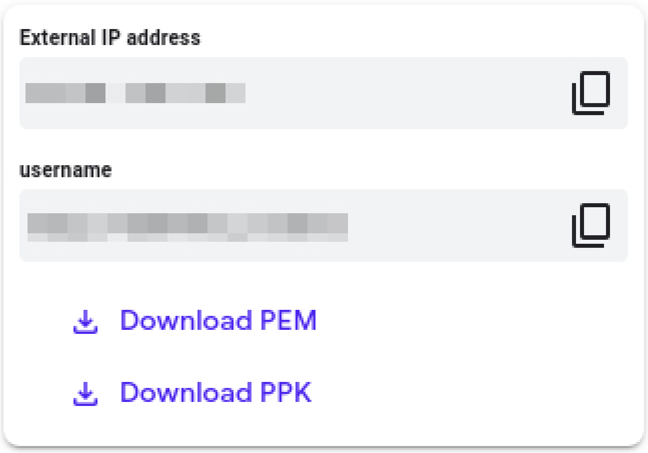
You'll have 90 minutes to complete this lab.

Start the lab

You'll need to start the lab before you can access the materials in the virtual machine OS. To do this, click the green “Start Lab” button at the top of the screen.

Note: For this lab you are going to access the **Linux VM** through your **local SSH Client**, and not use the **Google Console** (**Open GCP Console** button is not available for this lab).

After you click the “Start Lab” button, you will see all the SSH connection details on the left-hand side of your screen. You should have a screen that looks like this:



External IP address

[redacted]

username

[redacted]

Accessing the virtual machine

Please find one of the three relevant options below based on your device's operating system.

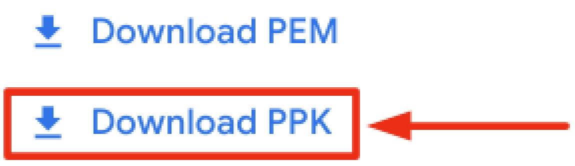
Note: Working with Qwiklabs may be similar to the work you'd perform as an **IT Support Specialist**; you'll be interfacing with a cutting-edge technology that requires multiple steps to access, and perhaps healthy doses of patience and persistence(!). You'll also be using **SSH** to enter the labs -- a critical skill in IT Support that you’ll be able to practice through the labs.

Option 1: Windows Users: Connecting to your VM

In this section, you will use the PuTTY Secure Shell (SSH) client and your VM’s External IP address to connect.

Download your PPK key file

You can download the VM's private key file in the PuTTY-compatible **PPK** format from the Qwiklabs Start Lab page. Click on **Download PPK**.

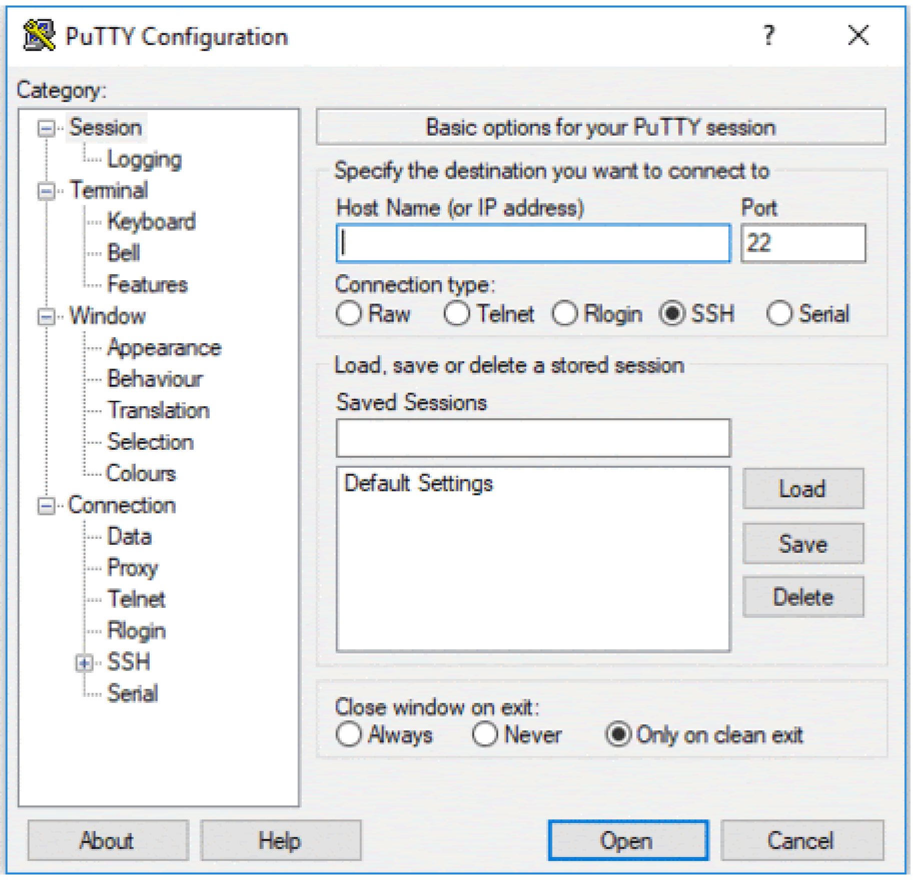


Connect to your VM using SSH and PuTTY

1. You can download Putty from here

2. In the **Host Name (or IP address)** box, enter username@external_ip_address.

Note: Replace **username** and **external_ip_address** with values provided in the lab.



Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Close window on exit: Always Never [✓ Only on clean exit]

3. In the **Connection** list, expand **SSH**.

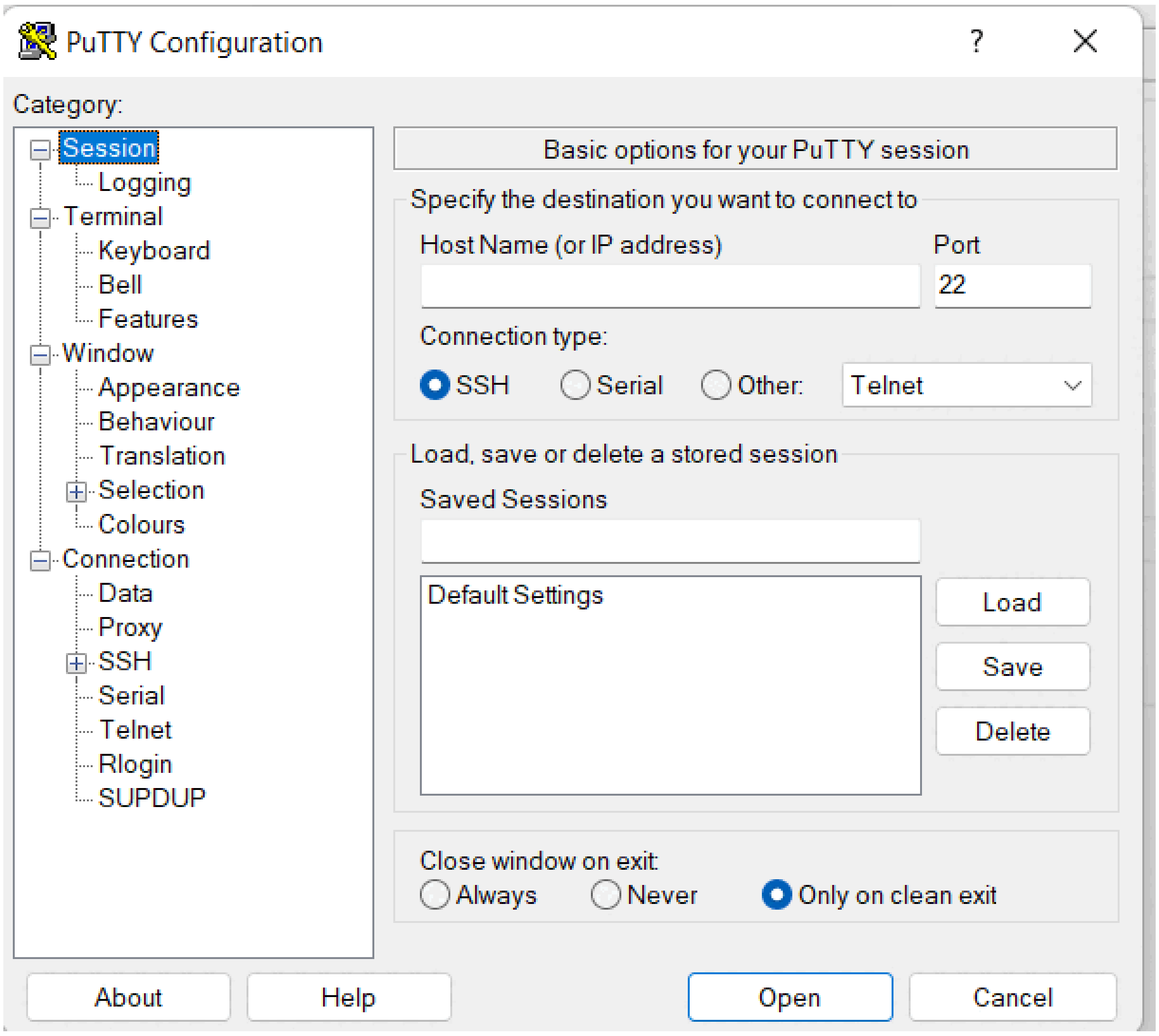
4. Then expand **Auth** by clicking on + icon.

5. Now, select the **Credentials** from the **Auth** list.

6. In the **Private key file for authentication** box, browse to the PPK file that you downloaded and double-click it.

7. Click on the **Open** button.

Note: PPK file is to be imported into PuTTY tool using the Browse option available in it. It should not be opened directly but only to be used in PuTTY.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address)

Port 22

Connection type: [✓ SSH] Serial Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

8. Click **Yes** when prompted to allow a first connection to this remote SSH server. Because you are using a key pair for authentication, you will not be prompted for a password.

Common issues

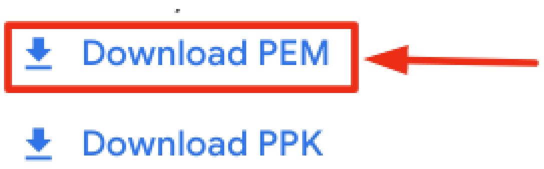
If PuTTY fails to connect to your Linux VM, verify that:

- You entered **<username>**@**<external ip address>** in PuTTY.
- You downloaded the fresh new PPK file for this lab from Qwiklabs.
- You are using the downloaded PPK file in PuTTY.

Option 2: OSX and Linux users: Connecting to your VM via SSH

Download your VM's private key file.

You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.



Connect to the VM using the local Terminal application

A **terminal** is a program which provides a **text-based interface for typing commands**. Here you will use your terminal as an SSH client to connect with lab provided Linux VM.

1. Open the Terminal application.

- To open the terminal in Linux use the shortcut key **Ctrl+Alt+t**.
- To open terminal in **Mac** (OSX) enter **cmd + space** and search for **terminal**.

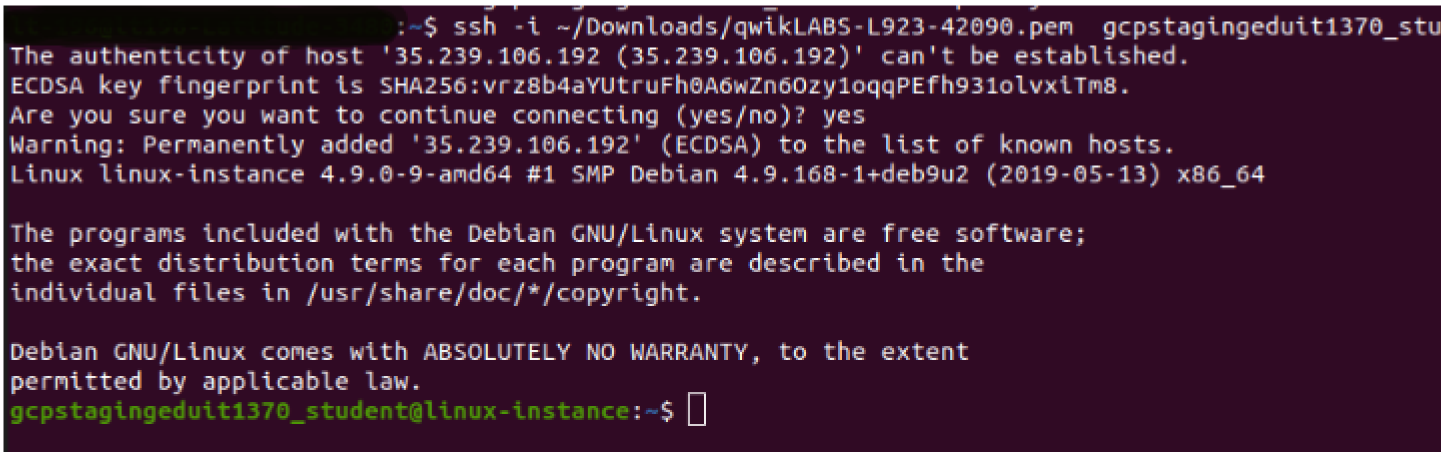
2. Enter the following commands.

Note: Substitute the **path/filename for the PEM** file you downloaded, **username** and **External IP Address**.

You will most likely find the PEM file in **Downloads**. If you have not changed the download settings of your system, then the path of the PEM key will be **~/Downloads/qwikLABS-XXXXX.pem**

```
chmod 600 ~/Downloads/qwikLABS-XXXXX.pem
```
Copied!
content_copy

```
ssh -i ~/Downloads/qwikLABS-XXXXX.pem username@External Ip Address
```
Copied!
content_copy



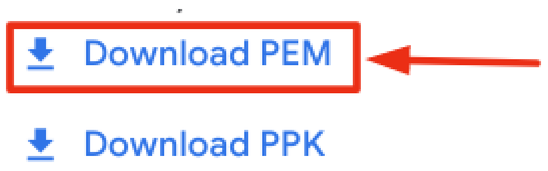
```
[redacted]:~$ ssh -i ~/Downloads/qwikLABS-L923-42090.pem  gcpstagingeduit1370_stu
The authenticity of host '35.239.106.192 (35.239.106.192)' can't be established.
ECDSA key fingerprint is SHA256:vrz8b4aYUtruFh0A6wZn6Ozy1oqqPEfh931olvxiTm8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '35.239.106.192' (ECDSA) to the list of known hosts.
Linux linux-instance 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u2 (2019-05-13) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
gcpstagingeduit1370_student@linux-instance:~$ 
```

Option 3: Chrome OS users: Connecting to your VM via SSH

Note: Make sure you are not in **Incognito/Private mode** while launching the application.

Download your VM's private key file.

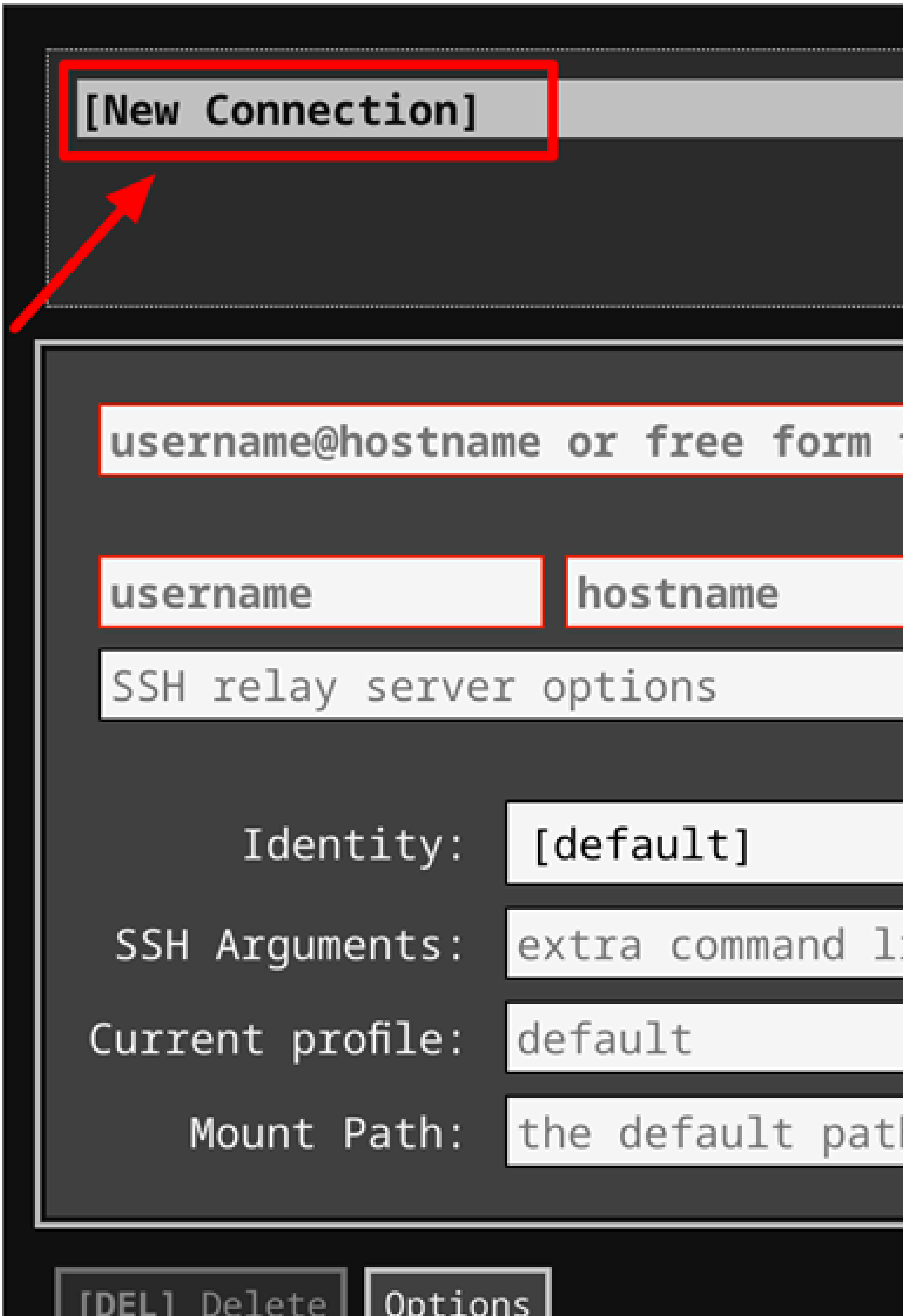
You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.



Connect to your VM

1. Add Secure Shell from here to your Chrome browser.

2. Open the Secure Shell app and click on **[New Connection]**.


[New Connection]

username@hostname or free form

username

hostname

SSH relay server options

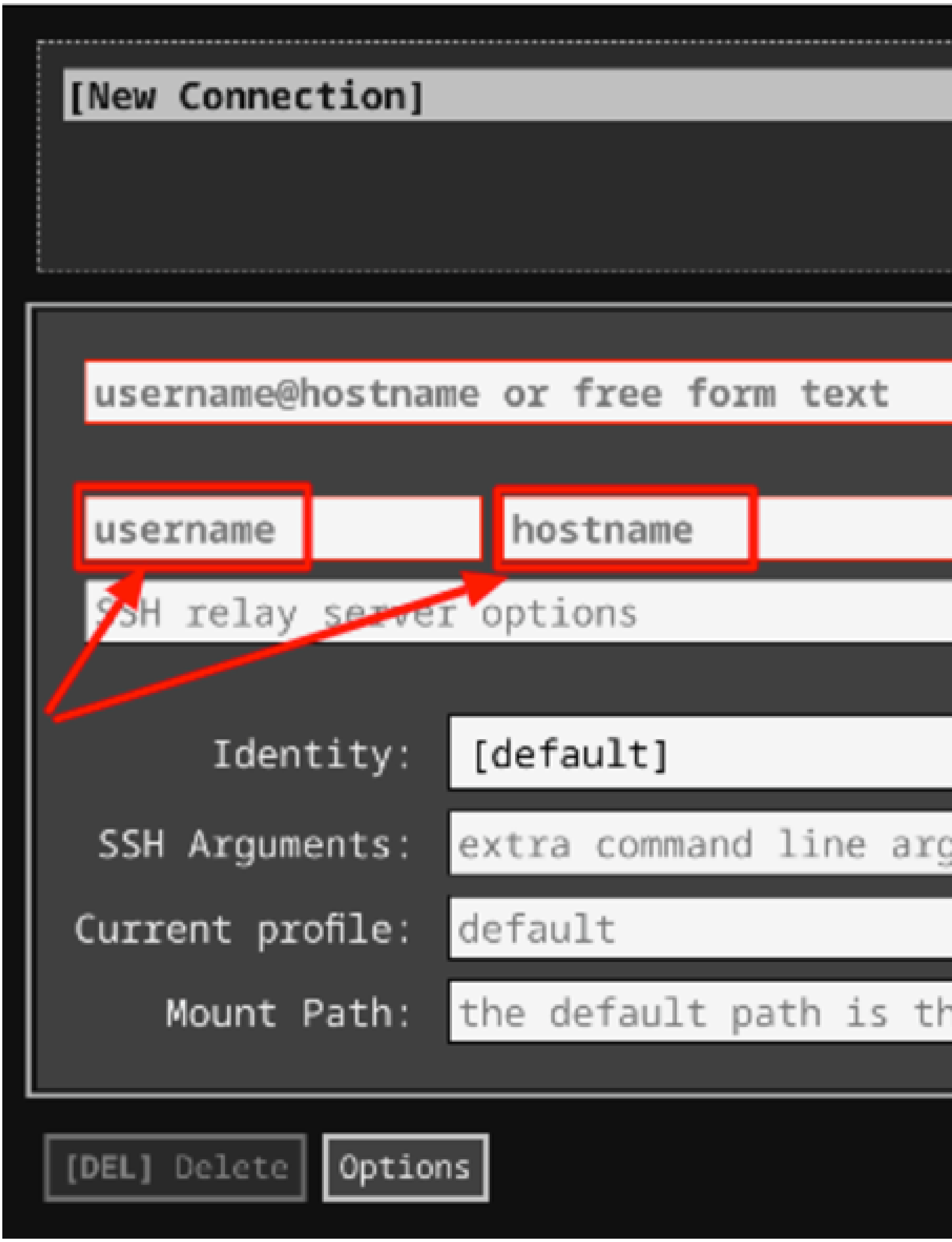
Identity: [default]

SSH Arguments: extra command l

Current profile: default

Mount Path: the default pat

3. In the **username** section, enter the username given in the Connection Details Panel of the lab. And for the **hostname** section, enter the external IP of your VM instance that is mentioned in the Connection Details Panel of the lab.


[New Connection]

username@hostname or free form text

username

hostname

SSH relay server options

Identity: [default]

SSH Arguments: extra command line arg

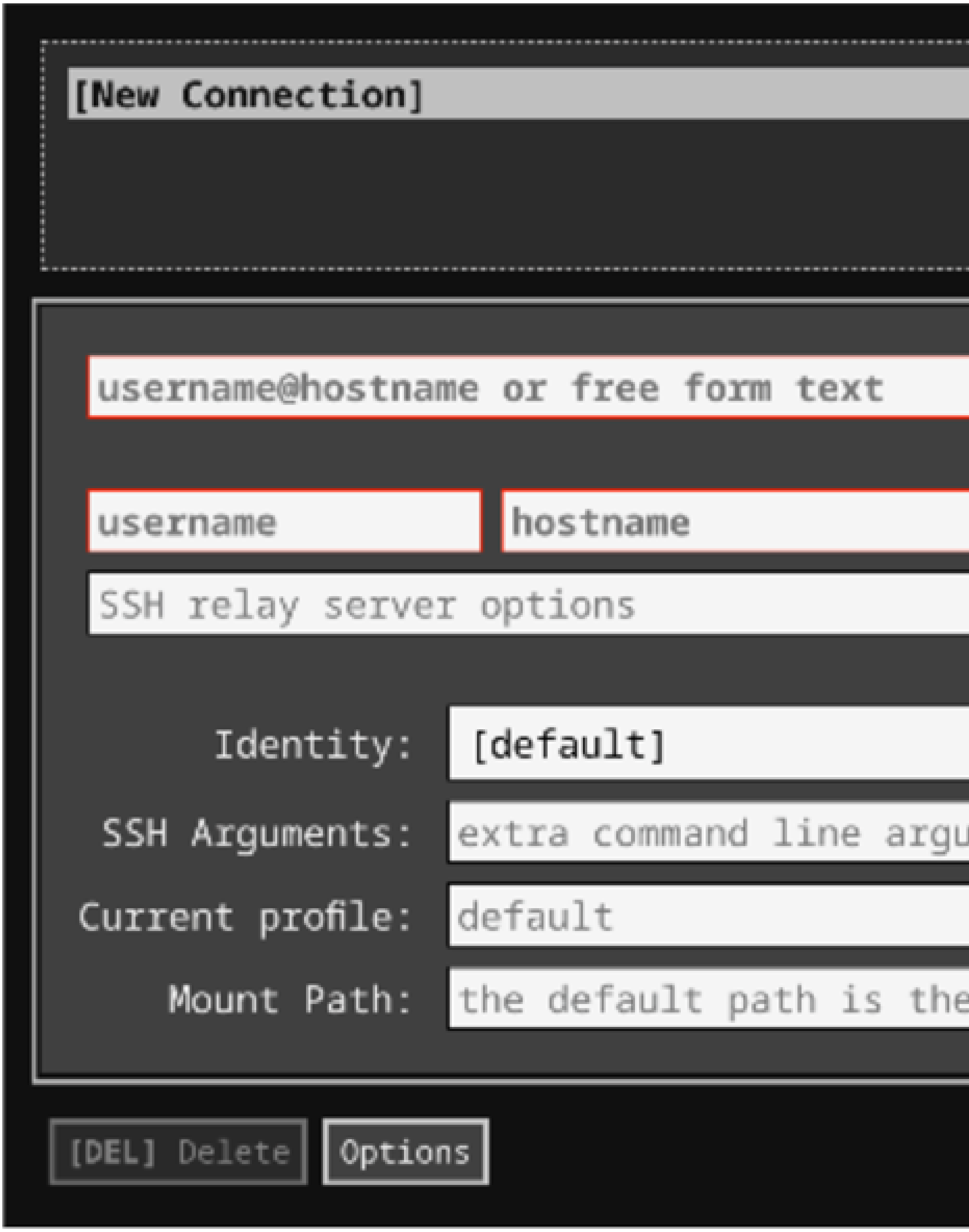
Current profile: default

Mount Path: the default path is th

4. In the **Identity** section, import the downloaded PEM key by clicking on the **Import…** button beside the field. Choose your PEM key and click on the **OPEN** button.

Note: If the key is still not available after importing it, refresh the application, and select it from the **Identity** drop-down menu.

5. Once your key is uploaded, click on the **[ENTER] Connect** button below.



[New Connection]

username@hostname or free form text

username

hostname

SSH relay server options

Identity: [default]

SSH Arguments: extra command line argu

Current profile: default

Mount Path: the default path is the

6. For any prompts, type **yes** to continue.

7. You have now successfully connected to your Linux VM.

You're now ready to continue with the lab!

Reproduce the error

The script sent by your colleague is in the **scripts** directory. Let's navigate to **scripts** directory using the following command:

```
cd ~/scripts
```

Copied!

content_copy

Now, use the following command to list all the files in this directory:

```
ls
```

Copied!

content_copy

You should now be able to see the file named **greetings.py** which was sent by your colleague.

To view the contents of the file, use the following command:

```
cat greetings.py
```

Copied!

content_copy

Let's update the file's permissions.

```
sudo chmod 777 greetings.py
```

Copied!

content_copy

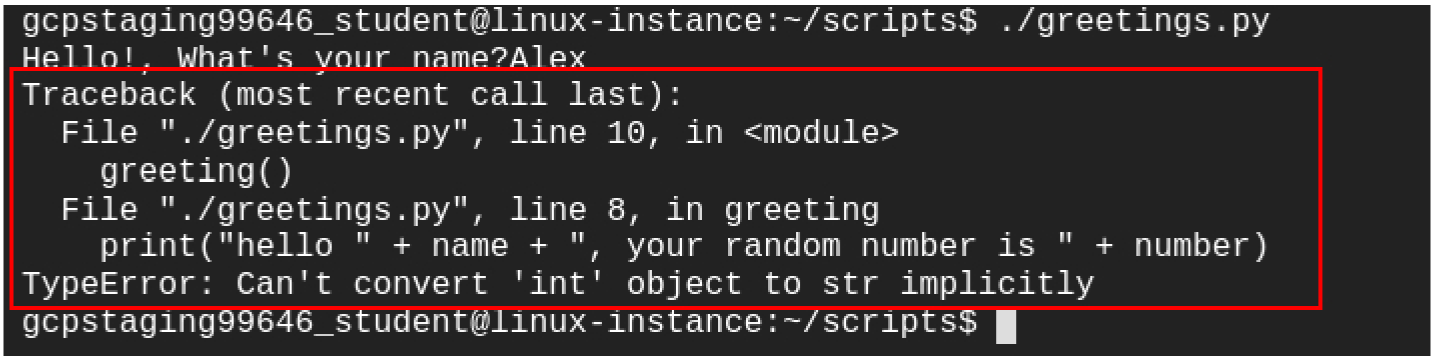
Now let's reproduce the error by running the file using the following command:

```
./greetings.py
```

Copied!
content_copy

Enter your name at the prompt.

The output should throw an error as shown below:



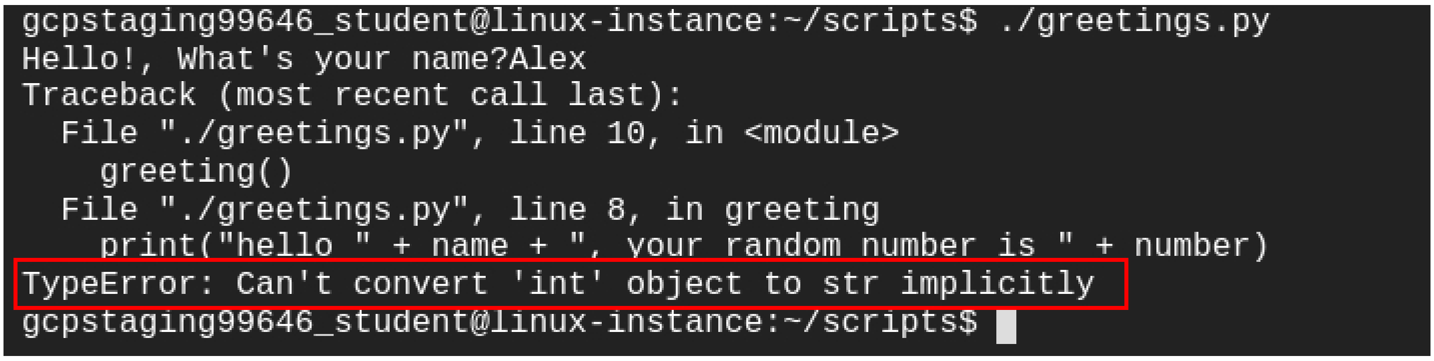
```
gcpstaging99646_student@linux-instance:~/scripts$ ./greetings.py
Hello!, What's your name?Alex
Traceback (most recent call last):
  File "./greetings.py", line 10, in <module>
    greeting()
  File "./greetings.py", line 8, in greeting
    print("hello " + name + ", your random number is " + number)
TypeError: Can't convert 'int' object to str implicitly
gcpstaging99646_student@linux-instance:~/scripts$
```

Great job! You have successfully reproduced the error.

Find the root cause of the issue

Now that we have successfully reproduced the error, let's find its root cause.

The error message indicates that something in the code is trying to concatenate a string and an integer.



```
gcpstaging99646_student@linux-instance:~/scripts$ ./greetings.py
Hello!, What's your name?Alex
Traceback (most recent call last):
  File "./greetings.py", line 10, in <module>
    greeting()
  File "./greetings.py", line 8, in greeting
    print("hello " + name + ", your random number is " + number)
TypeError: Can't convert 'int' object to str implicitly
gcpstaging99646_student@linux-instance:~/scripts$
```

When we look at the code, we can see that there are two different data types used, string and int. The variable **name** takes string values and the variable **number** stores integer (int) values.

So, the print statement within the script concatenates both string and integer values, which is causing the error.

```
print("hello " + name + ", your random number is " + number)
```

So, we can conclude that the root cause of the issue is within the print statement, which is trying to concatenate two different data types (i.e., string and int).

Debug the issue

In the previous section, we found the root cause of the issue, now let's debug the issue.

The print statement within the script is trying to concatenate two different data types. In Python, you can't add two different data types directly. So in this case, we can't add a string data type with an int data type. To add them, we have to turn the number into a string using str() function.

Once the integer value is converted to a string data type using str() function, the concatenate operation will work because the data types will be similar.

str() function takes in an integer as a parameter and converts it into string data type. So in our case, we will pass the variable **number** to the str() function i.e., **str(number)**.

Open the greetings.py file using nano editor.

```
nano greetings.py
```

Copied!
content_copy

Replace the print statement within the script with the following statement:

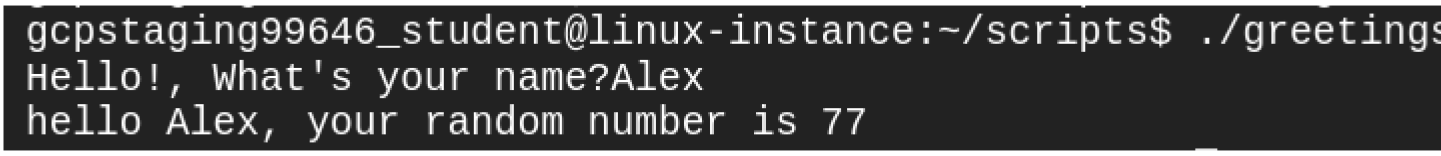
```
    print("hello " + name + ", your random number is " + str(number))
```
Copied!
content_copy

Save the file by pressing Ctrl-o, followed by the Enter key and Ctrl-x.

Now run the file again.

```
./greetings.py
```
Copied!
content_copy

Enter your name for the prompt. You should now see the correct output.



```
gcpstaging99646_student@linux-instance:~/scripts$ ./greetings
Hello!, What's your name?Alex
hello Alex, your random number is 77
```

Click Check my progress to verify the objective.

Debug the issue

Check my progress

Congratulations!

You successfully debugged your colleague's python script. You reproduced the error, found its root cause, and applied the remediation to the issue. You can now close the RDP/SSH window. The lab will automatically end when the time runs out, or you can end it manually.

End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.