Process Text Files with Python Dictionaries and Upload to Running Web Service

googlecoursera.qwiklabs.com/focuses/52112

Introduction

You're working at a company that sells second-hand cars. Your company constantly collects feedback in the form of customer reviews. Your manager asks you to take those reviews (saved as .txt files) and display them on your company's website. To do this, you'll need to write a script to convert those .txt files and process them into Python dictionaries, then upload the data onto your company's website (currently using Django).

What you'll do

- Use the Python OS module to process a directory of text files
- Manage information stored in Python dictionaries
- Use the Python requests module to upload content to a running Web service
- Understand basic operations for Python requests like GET and POST methods

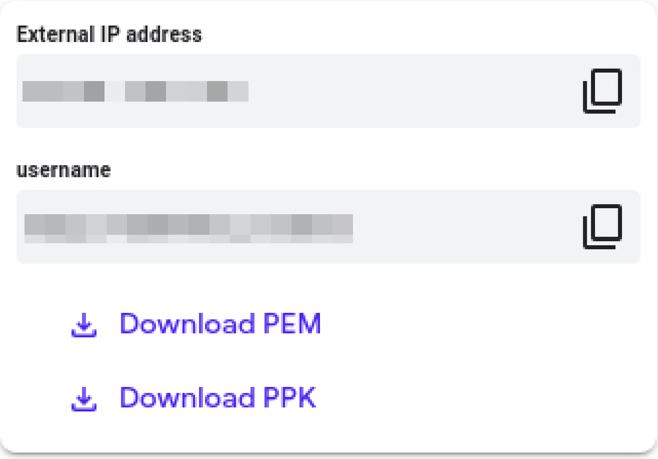
You'll have 90 minutes to complete this lab.

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.

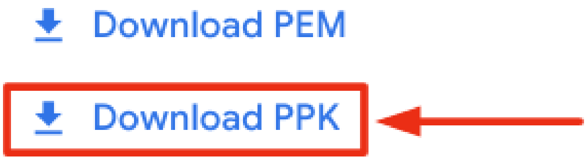
Option 1: Windows Users: Connecting to your VM

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**.

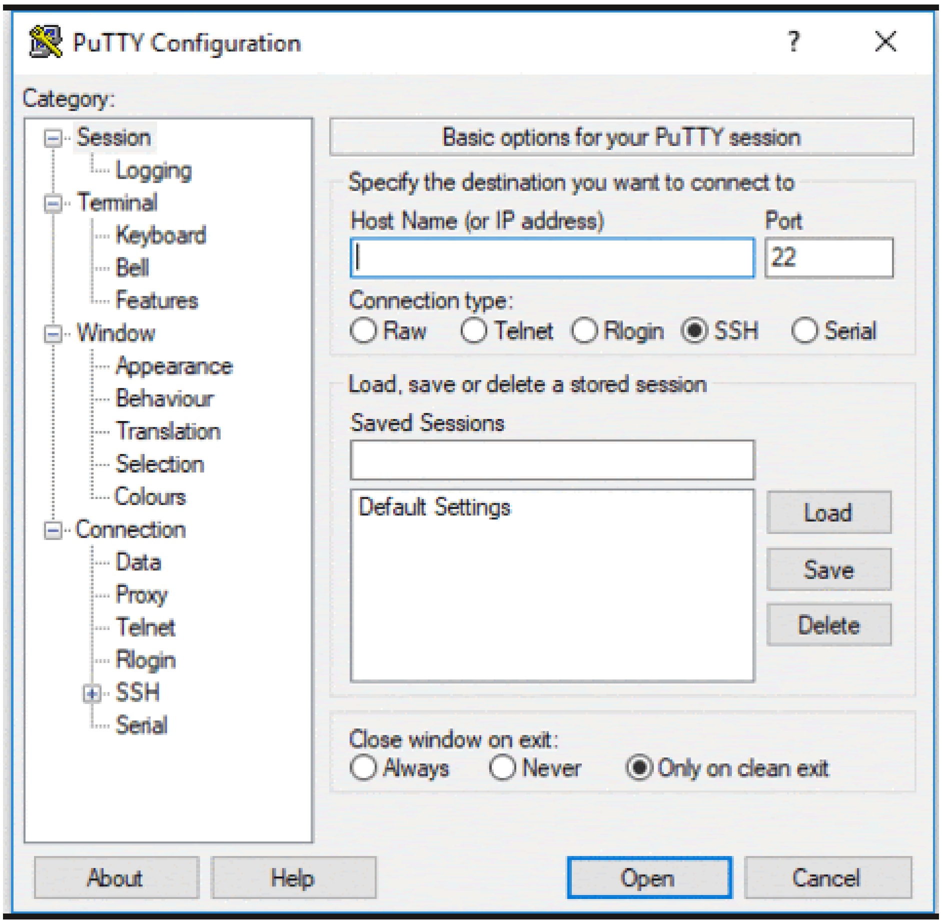
Connect to your VM using SSH and PuTTY



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.



Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

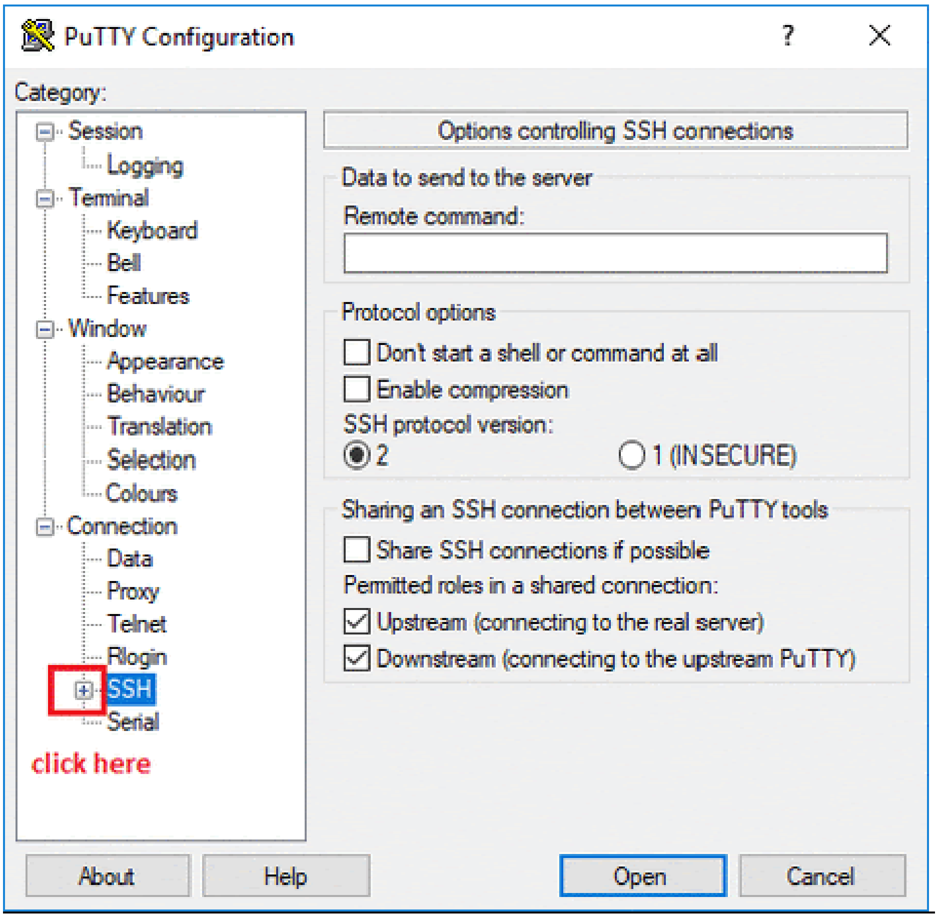
3. In the **Category** list, expand **SSH**.

4. Click **Auth** (don't expand it).

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

6. 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.



click here

Options controlling SSH connections

Data to send to the server

Remote command:

Protocol options

□ Don't start a shell or command at all

□ Enable compression

SSH protocol version:

● 2 ○ 1 (INSECURE)

Sharing an SSH connection between PuTTY tools

□ Share SSH connections if possible

Permitted roles in a shared connection:

☑ Upstream (connecting to the real server)

☑ Downstream (connecting to the upstream PuTTY)

7. 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.

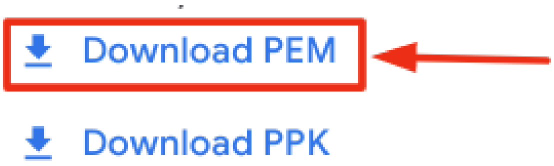
Option 2: OSX and Linux users: Connecting to your VM via SSH

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.

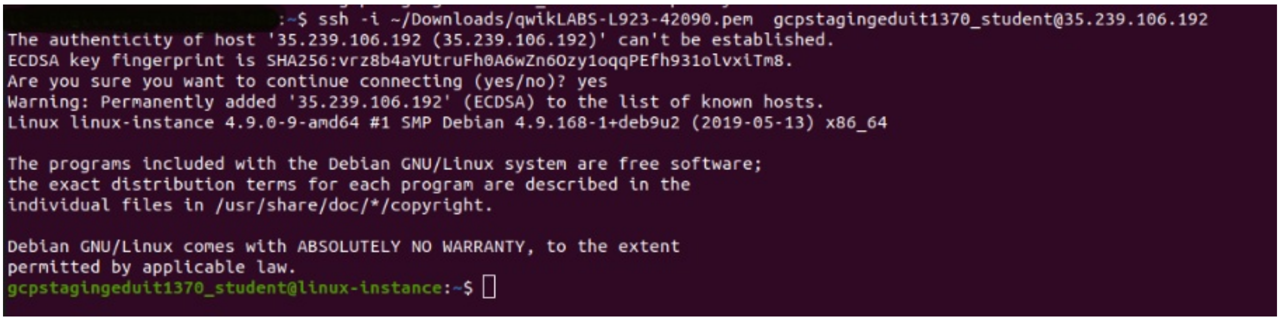


1. Open the Terminal application.
    - To open the terminal in Linux use the shortcut key **Ctrl+Alt+t**.
    - To open terminal in **Mac** (OSX) enter **cmd + space** and search for **terminal**.
2. Enter the following commands.

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

ssh -i ~/Downloads/qwikLABS-XXXXX.pem username@External Ip Address



```
[redacted]:~$ ssh -i ~/Downloads/qwikLABS-L923-42090.pem  gcpstagingeduit1370_student@35.239.106.192
The authenticity of host '35.239.106.192 (35.239.106.192)' can't be established.
ECDSA key fingerprint is SHA256:vrz8b4aYUtruFh0A6wZn6Ozy1oqqPEfh931olvxlTm8.
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

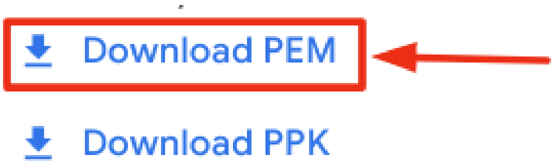
Note: Make sure you are not in **Incognito/Private mode** while launching the application.
Download your VM's private key file.

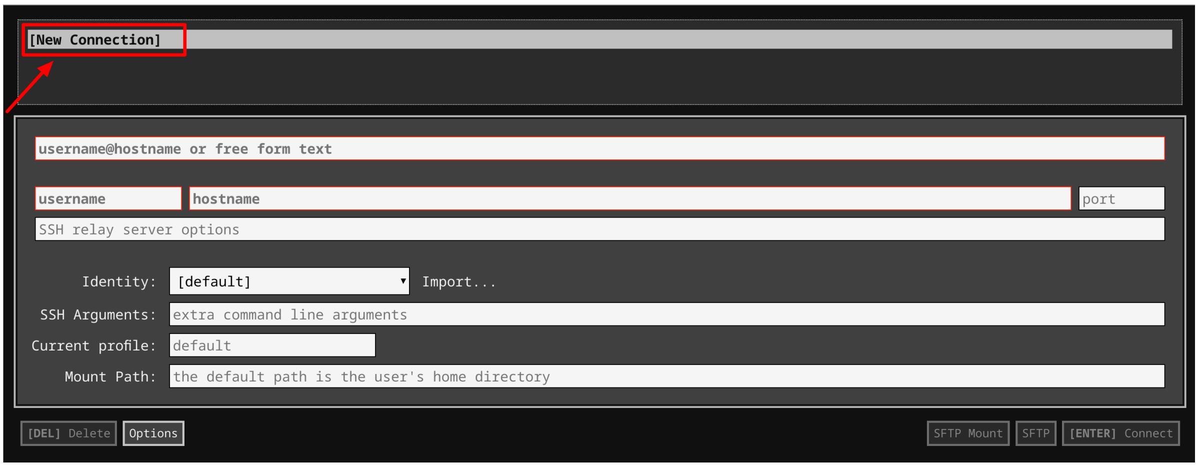
You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.

Connect to your VM

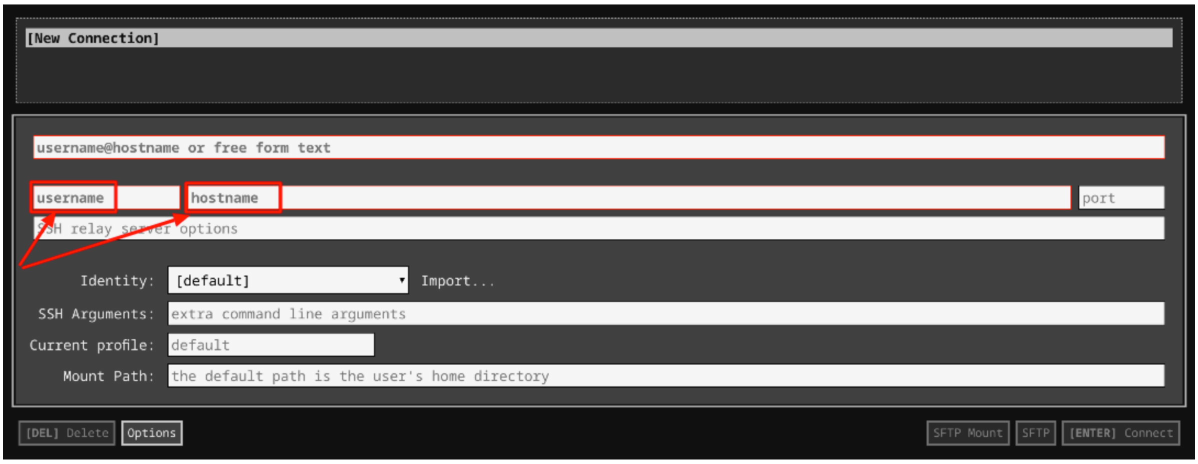
1. Add Secure Shell from here to your Chrome browser.

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





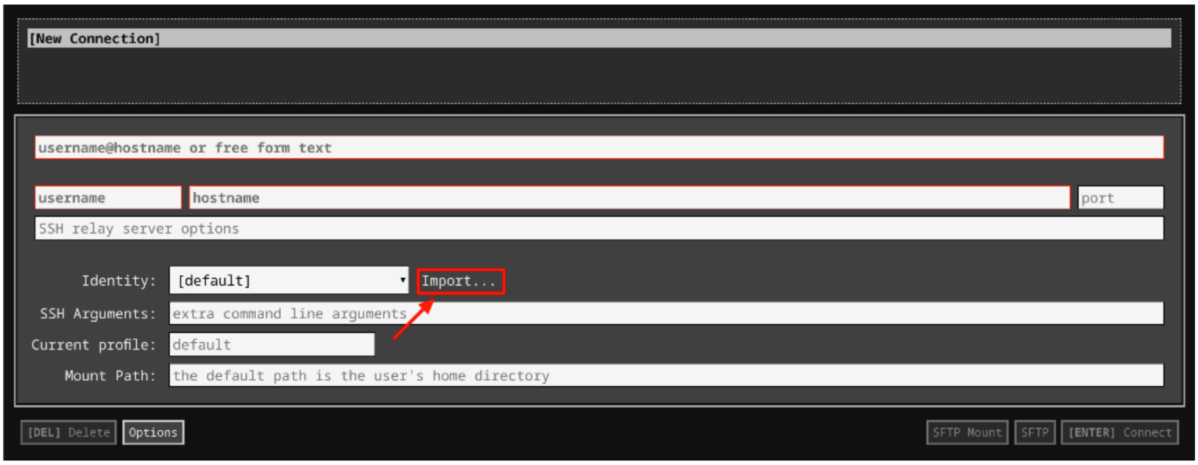
3. In the **username** section, enter the username given in the Connection Details Panel of the lab. And for the **hostname** section, enter the external IP of your VM instance that is mentioned in the Connection Details Panel of the lab.



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 port

SSH relay server options

Identity: [default] Import...

SSH Arguments: extra command line arguments

Current profile: default

Mount Path: the default path is the user's home directory

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

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

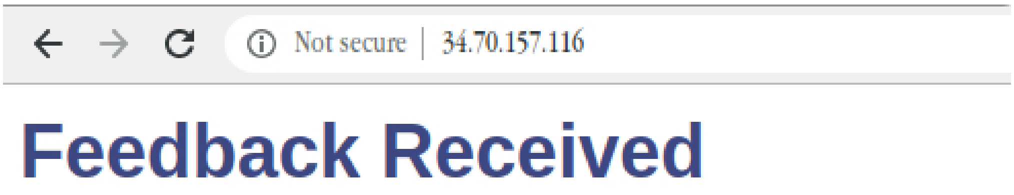
You're now ready to continue with the lab!

Web server corpweb

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. A Web framework is a set of components that provide a standard way to develop websites quickly and easily.

For this lab, a Django web server `corpweb` is already configured under `/projects/corpweb` directory. You can check it out by visiting the external IP address of the `corpweb` VM. The external IP address can be found in the connection details panel. Enter the `corpweb` external IP address in a new separate browser tab.

Output:

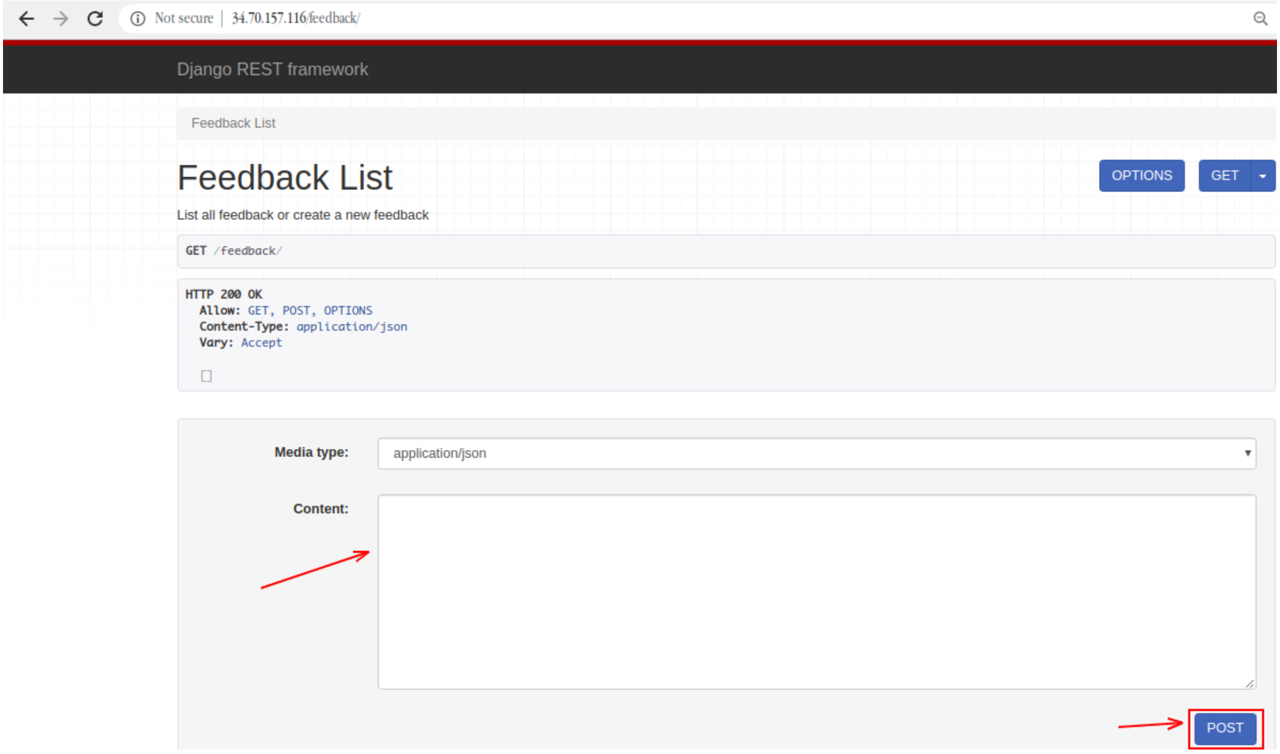


Not secure | 34.70.157.116

Feedback Received

You'll see that there's currently no feedback.

Now, append `/feedback` to the external IP address of `corpweb` VM opened in the browser tab.



Feedback List

List all feedback or create a new feedback

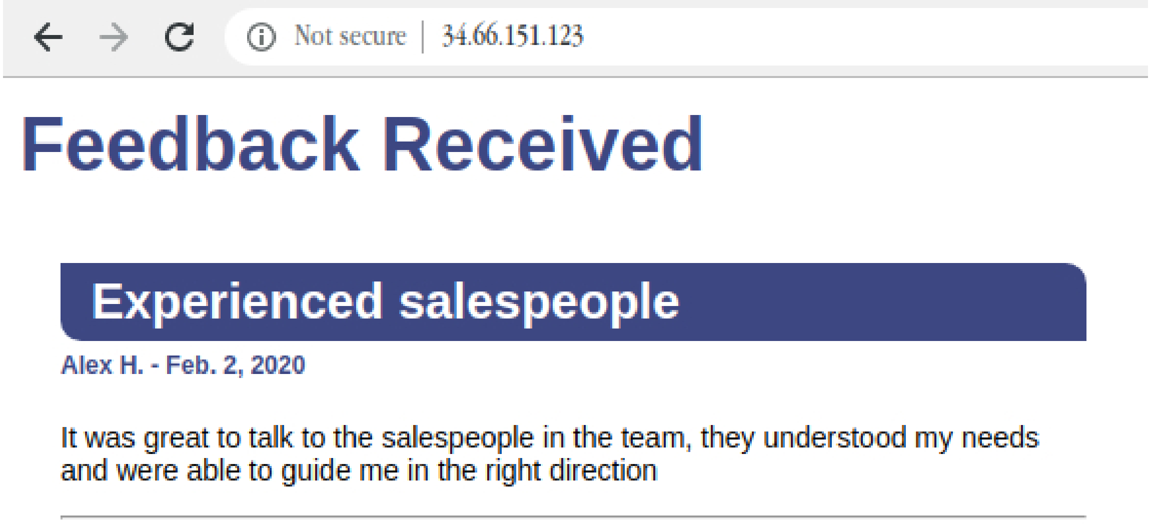
```
GET /feedback/
```

```
HTTP 200 OK
  Allow: GET, POST, OPTIONS
  Content-Type: application/json
  Vary: Accept
```

This is a web interface for a REST end-point. Through this end-point, you can enter feedback that can be displayed on the company's website. You can use this end-point in the example below. Start by copying and pasting the following JSON to the **Content** field on the website, and click **POST**.

{"title": "Experienced salespeople", "name": "Alex H.", "date": "2020-02-02", "feedback": "It was great to talk to the salespeople in the team, they understood my needs and were able to guide me in the right direction"}

Now, go back to the main page by removing the `/feedback` from the URL. You can see that the feedback that you just entered is displayed on the webpage.



Feedback Received

Experienced salespeople

Alex H. - Feb. 2, 2020

It was great to talk to the salespeople in the team, they understood my needs and were able to guide me in the right direction

The whole website is stored in `/projects/corpweb`. You're free to look around the configuration files. Also, there's no need to make any changes to the website; all interaction should be done through the REST end-point.

Process text files and upload to running web server

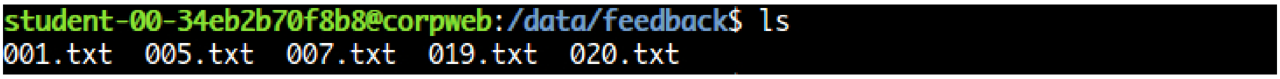
In this section, you'll write a Python script that will upload the feedback automatically without having to turn it into a dictionary.

Navigate to `/data/feedback` directory, where you'll find a few .txt files with customer reviews for the company.

```
cd /data/feedback
```

```
ls
```

Output:

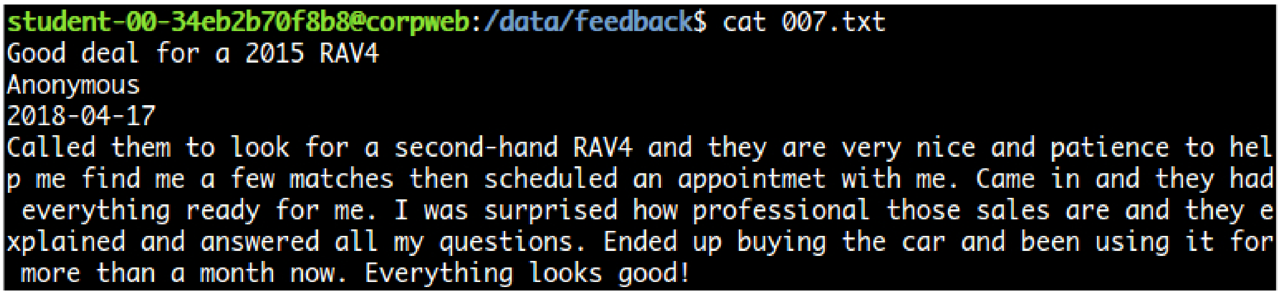


```
student-00-34eb2b70f8b8@corpweb:/data/feedback$ ls
001.txt  005.txt  007.txt  019.txt  020.txt
```

Use the `cat` command to view these files. For example:

```
cat 007.txt
```

Output:



```
student-00-34eb2b70f8b8@corpweb:/data/feedback$ cat 007.txt
Good deal for a 2015 RAV4
Anonymous
2018-04-17
Called them to look for a second-hand RAV4 and they are very nice and patience to hel
p me find me a few matches then scheduled an appointmet with me. Came in and they had
 everything ready for me. I was surprised how professional those sales are and they e
xplained and answered all my questions. Ended up buying the car and been using it for
 more than a month now. Everything looks good!
```

They're all written in the same format (i.e. `title, name, date,` and `feedback).`

Here comes the challenge section of the lab, where you'll write a Python script that uploads all the feedback stored in this folder to the company's website, without having to turn it into a dictionary one by one.

Now, navigate back to the `home` directory and create a Python script named `run.py` using the following command:

```
cd ~
```

```
nano run.py
```

Add the shebang line:

```
#! /usr/bin/env python3
```

The following are a few libraries that will be required for the script. Import them using:

```
import os
import requests
```

The script should now follow the structure:

- List all .txt files under `/data/feedback` directory that contains the actual feedback to be displayed on the company's website.

  Hint: Use os.listdir() method for this, which returns a list of all files and directories in the specified path.
- You should now have a list that contains all of the feedback files from the path `/data/feedback`. Traverse over each file and, from the contents of these text files, create a dictionary by keeping `title`, `name`, `date`, and `feedback` as keys for the content value, respectively.

- Now, you need to have a dictionary with keys and their respective values (content from feedback files). This will be uploaded through the Django REST API.

- Use the Python `requests` module to post the dictionary to the company's website. Use the request.post() method to make a POST request to `http://<corpweb-external-IP>/feedback`. Replace `<corpweb-external-IP>` with corpweb's external IP address.

- Make sure an error message isn't returned. You can print the status_code and text of the response objects to check out what's going on. You can also use the response `status_code 201` for created success status response code that indicates the request has succeeded.

Save the `run.py` script file by pressing Ctrl-o, the Enter key, and Ctrl-x.

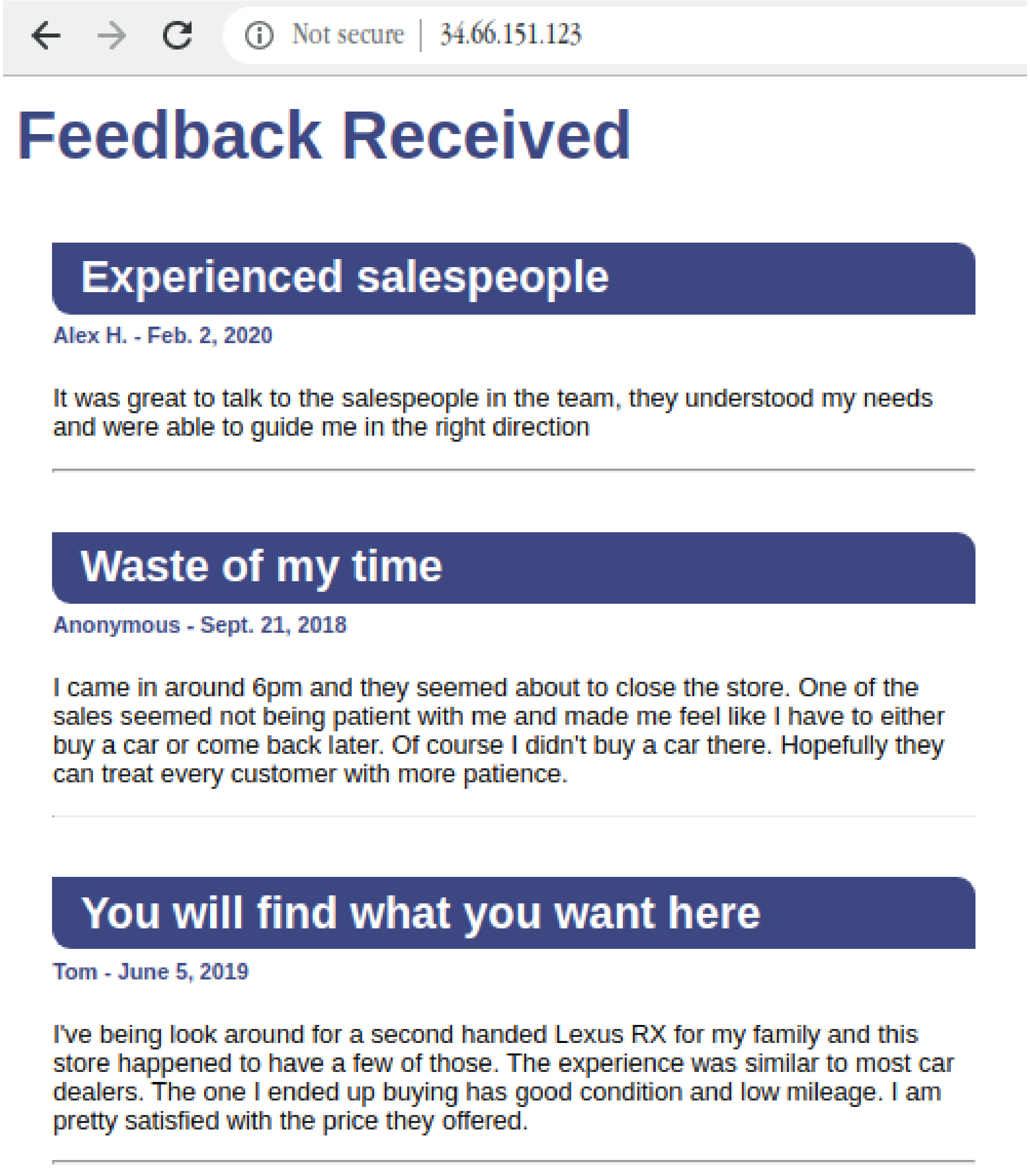
Grant executable permission to the run.py script.

```
chmod +x ~/run.py
```

Now, run the `run.py` script:

```
./run.py
```

Your POST requests should have successfully uploaded the feedback on the company's website. Now, visit the website again using the `corpweb` external IP address or just refresh the page if already opened, and you should be able to see the feedback.



Feedback Received

Experienced salespeople

Alex H. - Feb. 2, 2020

It was great to talk to the salespeople in the team, they understood my needs and were able to guide me in the right direction

Waste of my time

Anonymous - Sept. 21, 2018

I came in around 6pm and they seemed about to close the store. One of the sales seemed not being patient with me and made me feel like I have to either buy a car or come back later. Of course I didn't buy a car there. Hopefully they can treat every customer with more patience.

You will find what you want here

Tom - June 5, 2019

I've being look around for a second handed Lexus RX for my family and this store happened to have a few of those. The experience was similar to most car dealers. The one I ended up buying has good condition and low mileage. I am pretty satisfied with the price they offered.

Click *Check my progress* to verify the objective. Process text files and upload to running web server

Congratulations!

Nice work! You've successfully displayed customer reviews on the company website by writing a Python script that automatically uploads feedback in the form of a text file to a running web server.

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.