

# 0x03. AirBnB clone - Deploy static

DevOpsPythonSysAdminScriptingCI/CD

- By: Guillaume, CTO at Holberton School
- Weight: 1
- Project over - took place from Sep 28, 2022 6:00 AM to Sep 30, 2022 6:00 AM
- An auto review will be launched at the deadline

## *In a nutshell...*

- **Auto QA review:** 0.0/51 mandatory & 0.0/35 optional
- **Altogether: 0.0%**
  - Mandatory: 0.0%
  - Optional: 0.0%
  - Calculation:  $0.0\% + (0.0\% * 0.0\%) == 0.0\%$

## Concepts

*For this project, we expect you to look at these concepts:*

- CI/CD
- AirBnB clone

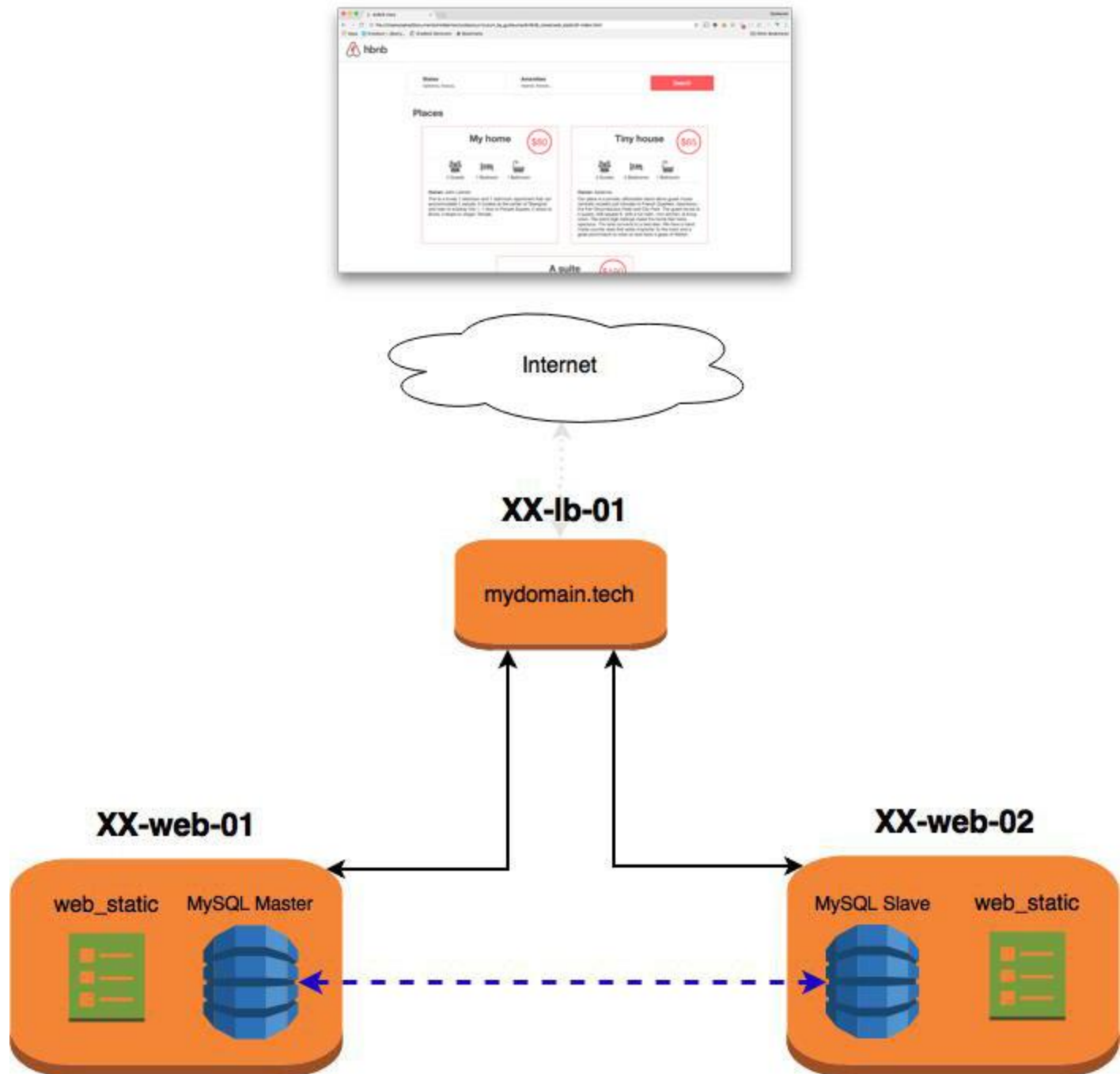
## Background Context

Ever since you completed project **0x0F. Load balancer** of the SysAdmin track, you've had 2 web servers + 1 load balancer but nothing to distribute with them.

It's time to make your work public!

In this first deployment project, you will be deploying your **web\_static** work. You will use **Fabric** (for Python3). Fabric is a Python library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks. It provides a basic suite of operations for executing local or remote shell commands (normally or via **sudo**) and uploading/downloading files, as well as auxiliary functionality such as prompting the running user for input, or aborting execution. This concept is important: execute commands locally or remotely. Locally means in your laptop (physical laptop or inside your Vagrant), and Remotely means on your server(s). Fabric is taking care of all network connections (SSH, SCP etc.), it's an easy tool for transferring, executing, etc. commands from locale to a remote server.

Before starting, please fork the repository **AirBnB\_clone\_v2** from your partner if you don't have it



## Resources

Read or watch:

- [How to use Fabric](#)
- [How to use Fabric in Python](#)

- Fabric and command line options
- CI/CD concept page
- Nginx configuration for beginners
- Difference between root and alias on NGINX
- Fabric for Python 3
- Fabric Documentation

## Learning Objectives

At the end of this project, you are expected to be able to **explain to anyone, without the help of Google:**

### General

- What is Fabric
- How to deploy code to a server easily
- What is a `tgz` archive
- How to execute Fabric command locally
- How to execute Fabric command remotely
- How to transfer files with Fabric
- How to manage Nginx configuration
- What is the difference between `root` and `alias` in a Nginx configuration

### Copyright - Plagiarism

- You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
- You will not be able to meet the objectives of this or any following project by copying and pasting someone else's work.
- You are not allowed to publish any content of this project.
- Any form of plagiarism is strictly forbidden and will result in removal from the program.

## Requirements

### Python Scripts

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted/compiled on Ubuntu 14.04 LTS using `python3` (version 3.4.3)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/python3`
- A `README.md` file at the root of the folder of the project is mandatory
- Your code should use the `PEP 8` style (version `1.7.*`)

- Your Fabric file must work with **Fabric 3** version **1.14.post1** (installation instruction below)
- All your files must be executable
- The length of your files will be tested using **wc**
- All your functions (inside and outside a class) should have documentation (**python3 -c 'print(\_\_import\_\_("my\_module").my\_function.\_\_doc\_\_)'** and **python3 -c 'print(\_\_import\_\_("my\_module").MyClass.my\_function.\_\_doc\_\_)'**)
- A documentation is not a simple word, it's a real sentence explaining what's the purpose of the module, class or method (the length of it will be verified)

## Bash Scripts

- Allowed editors: **vi**, **vim**, **emacs**
- All your files will be interpreted on Ubuntu 14.04 LTS
- All your files should end with a new line
- A **README.md** file at the root of the folder of the project is mandatory
- All your Bash script files must be executable
- Your Bash script must pass **Shellcheck** (version **0.3.3-1~ubuntu14.04.1** via **apt-get**) without any errors
- The first line of all your Bash scripts should be exactly **#!/usr/bin/env bash**
- The second line of all your Bash scripts should be a comment explaining what is the script doing

## More Info

### Install Fabric for Python 3 - version 1.14.post1

```
$ pip3 uninstall Fabric
$ sudo apt-get install libffi-dev
$ sudo apt-get install libssl-dev
$ sudo apt-get install build-essential
$ sudo apt-get install python3.4-dev
$ sudo apt-get install libpython3-dev
$ pip3 install pyparsing
$ pip3 install appdirs
$ pip3 install setuptools==40.1.0
$ pip3 install cryptography==2.8
$ pip3 install bcrypt==3.1.7
$ pip3 install PyNaCl==1.3.0
$ pip3 install Fabric3==1.14.post1
```

## Quiz questions

Great! You've completed the quiz successfully! Keep going! ([Show quiz](#))

# Tasks

## 0. Prepare your web servers

mandatory

Score: 0.0% (Checks completed: 0.0%)

Write a Bash script that sets up your web servers for the deployment of `web_static`. It must:

- Install Nginx if it not already installed
- Create the folder `/data/` if it doesn't already exist
- Create the folder `/data/web_static/` if it doesn't already exist
- Create the folder `/data/web_static/releases/` if it doesn't already exist
- Create the folder `/data/web_static/shared/` if it doesn't already exist
- Create the folder `/data/web_static/releases/test/` if it doesn't already exist
- Create a fake HTML file `/data/web_static/releases/test/index.html` (with simple content, to test your Nginx configuration)
- Create a symbolic link `/data/web_static/current` linked to the `/data/web_static/releases/test/` folder. If the symbolic link already exists, it should be deleted and recreated every time the script is ran.
- Give ownership of the `/data/` folder to the `ubuntu` user AND group (you can assume this user and group exist). This should be recursive; everything inside should be created/owned by this user/group.
- Update the Nginx configuration to serve the content of `/data/web_static/current/` to `hbnb_static` (ex: `https://mydomainname.tech/hbnb_static`). Don't forget to restart Nginx after updating the configuration:
  - Use `alias` inside your Nginx configuration
  - Tip

Your program should always exit successfully. **Don't forget to run your script on both of your web servers.**

In optional, you will redo this task but by using Puppet

```
ubuntu@89-web-01:~/$ sudo ./0-setup_web_static.sh
ubuntu@89-web-01:~/$ echo $?
0
ubuntu@89-web-01:~/$ ls -l /data
total 4
drwxr-xr-x 1 ubuntu ubuntu    4096 Mar  7 05:17 web_static
ubuntu@89-web-01:~/$ ls -l /data/web_static
```

```
total 8
lrwxrwxrwx 1 ubuntu ubuntu 30 Mar 7 22:30 current -> /data/web_static/releases/test
drwxr-xr-x 3 ubuntu ubuntu 4096 Mar 7 22:29 releases
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar 7 22:29 shared
ubuntu@89-web-01:~/ $ ls /data/web_static/current
index.html
ubuntu@89-web-01:~/ $ cat /data/web_static/current/index.html
<html>
  <head>
</head>
  <body>
    Holberton School
  </body>
</html>
ubuntu@89-web-01:~/ $ curl localhost/hbnb_static/index.html
<html>
  <head>
</head>
  <body>
    Holberton School
  </body>
</html>
ubuntu@89-web-01:~/ $
```

## Repo:

- GitHub repository: `AirBnB_clone_v2`
- File: `0-setup_web_static.sh`

Done? Help Check your code Ask for a new correction Get a sandbox QA Review

### 1. Compress before sending

mandatory

Score: 0.0% (Checks completed: 0.0%)

Write a Fabric script that generates a `.tgz` archive from the contents of the `web_static` folder of your AirBnB Clone repo, using the function `do_pack`.

- Prototype: `def do_pack():`
- All files in the folder `web_static` must be added to the final archive
- All archives must be stored in the folder `versions` (your function should create this folder if it doesn't exist)
- The name of the archive created must be `web_static_<year><month><day><hour><minute><second>.tgz`
- The function `do_pack` must return the archive path if the archive has been correctly generated. Otherwise, it should return `None`

```
guillaume@ubuntu:~/AirBnB_clone_v2$ fab -f 1-pack_web_static.py do_pack
Packing web_static to versions/web_static_20170314233357.tgz
[localhost] local: tar -cvzf versions/web_static_20170314233357.tgz web_static
web_static/
web_static/.DS_Store
web_static/0-index.html
web_static/1-index.html
web_static/100-index.html
web_static/2-index.html
web_static/3-index.html
web_static/4-index.html
web_static/5-index.html
web_static/6-index.html
web_static/7-index.html
web_static/8-index.html
web_static/images/
web_static/images/icon.png
web_static/images/icon_bath.png
web_static/images/icon_bed.png
web_static/images/icon_group.png
web_static/images/icon_pets.png
web_static/images/icon_tv.png
web_static/images/icon_wifi.png
web_static/images/logo.png
web_static/index.html
web_static/styles/
web_static/styles/100-places.css
```

```
web_static/styles/2-common.css
web_static/styles/2-footer.css
web_static/styles/2-header.css
web_static/styles/3-common.css
web_static/styles/3-footer.css
web_static/styles/3-header.css
web_static/styles/4-common.css
web_static/styles/4-filters.css
web_static/styles/5-filters.css
web_static/styles/6-filters.css
web_static/styles/7-places.css
web_static/styles/8-places.css
web_static/styles/common.css
web_static/styles/filters.css
web_static/styles/footer.css
web_static/styles/header.css
web_static/styles/places.css
web_static packed: versions/web_static_20170314233357.tgz -> 21283Bytes

Done.

guillaume@ubuntu:~/AirBnB_clone_v2$ ls -l versions/web_static_20170314233357.tgz
-rw-rw-r-- 1 guillaume guillaume 21283 Mar 14 23:33 versions/web_static_20170314233357.tgz
guillaume@ubuntu:~/AirBnB_clone_v2$
```

## Repo:

- GitHub repository: [AirBnB\\_clone\\_v2](#)
- File: [1-pack\\_web\\_static.py](#)

Done? Help Check your code Ask for a new correction Get a sandbox QA Review

## 2. Deploy archive!

mandatory

Score: 0.0% (Checks completed: 0.0%)

Write a Fabric script (based on the file [1-pack\\_web\\_static.py](#)) that distributes an archive to your web servers, using the function [do\\_deploy](#):



- Prototype: `def do_deploy(archive_path):`
- Returns `False` if the file at the path `archive_path` doesn't exist
- The script should take the following steps:
  - Upload the archive to the `/tmp/` directory of the web server
  - Uncompress the archive to the folder `/data/web_static/releases/<archive filename without extension>` on the web server
  - Delete the archive from the web server
  - Delete the symbolic link `/data/web_static/current` from the web server
  - Create a new the symbolic link `/data/web_static/current` on the web server, linked to the new version of your code (`/data/web_static/releases/<archive filename without extension>`)
- All remote commands must be executed on your both web servers (using `env.hosts = ['<IP web-01>', 'IP web-02']` variable in your script)
- Returns `True` if all operations have been done correctly, otherwise returns `False`
- You must use this script to deploy it on your servers: `xx-web-01` and `xx-web-02`

In the following example, the SSH key and the username used for accessing to the server are passed in the command line. Of course, you could define them as Fabric environment variables (ex: `env.user = ...`)

**Disclaimer:** commands execute by Fabric displayed below are linked to the way we implemented the archive function `do_pack` - like the `mv` command - depending of your implementation of it, you may don't need it

```
guillaume@ubuntu:~/AirBnB_clone_v2$ fab -f 2-do_deploy_web_static.py do_deploy:archive_path=versions/web_static_20170315003959.tgz -i my_ssh_private_key -u ubuntu
[52.55.249.213] Executing task 'do_deploy'
[52.55.249.213] put: versions/web_static_20170315003959.tgz -> /tmp/web_static_20170315003959.tgz
[52.55.249.213] run: mkdir -p /data/web_static/releases/web_static_20170315003959/
[52.55.249.213] run: tar -xzf /tmp/web_static_20170315003959.tgz -C /data/web_static/releases/web_static_20170315003959/
[52.55.249.213] run: rm /tmp/web_static_20170315003959.tgz
[52.55.249.213] run: mv /data/web_static/releases/web_static_20170315003959/web_static/* /data/web_static/releases/web_static_20170315003959/
[52.55.249.213] run: rm -rf /data/web_static/releases/web_static_20170315003959/web_static
[52.55.249.213] run: rm -rf /data/web_static/current
[52.55.249.213] run: ln -s /data/web_static/releases/web_static_20170315003959/ /data/web_static/current
New version deployed!
[54.157.32.137] Executing task 'deploy'
[54.157.32.137] put: versions/web_static_20170315003959.tgz -> /tmp/web_static_20170315003959.tgz
```

```
[54.157.32.137] run: mkdir -p /data/web_static/releases/web_static_20170315003959/
[54.157.32.137] run: tar -xzf /tmp/web_static_20170315003959.tgz -C /data/web_static/
releases/web_static_20170315003959/
[54.157.32.137] run: rm /tmp/web_static_20170315003959.tgz
[54.157.32.137] run: mv /data/web_static/releases/web_static_20170315003959/web_stati
c/* /data/web_static/releases/web_static_20170315003959/
[54.157.32.137] run: rm -rf /data/web_static/releases/web_static_20170315003959/web_s
tatic
[54.157.32.137] run: rm -rf /data/web_static/current
[54.157.32.137] run: ln -s /data/web_static/releases/web_static_20170315003959/ /data
/web_static/current
New version deployed!
```

Done.

Disconnecting from 54.157.32.137... done.

Disconnecting from 52.55.249.213... done.

guillaume@ubuntu:~/AirBnB\_clone\_v2\$

guillaume@ubuntu:~/AirBnB\_clone\_v2\$ curl 54.157.32.137/hbnb\_static/0-index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <meta charset="UTF-8" />
```

```
    <title>AirBnB clone</title>
```

```
  </head>
```

```
  <body style="margin: 0px; padding: 0px;">
```

```
    <header style="height: 70px; width: 100%; background-color: #FF0000">
```

```
    </header>
```

```
    <footer style="position: absolute; left: 0; bottom: 0; height: 60px; width: 1
00%; background-color: #00FF00; text-align: center; overflow: hidden;">
```

```
      <p style="line-height: 60px; margin: 0px;">Holberton School</p>
```

```
    </footer>
```

```
  </body>
```

```
</html>
```

guillaume@ubuntu:~/AirBnB\_clone\_v2\$

## Repo:

- GitHub repository: `AirBnB_clone_v2`
- File: `2-do_deploy_web_static.py`

Done? Help Check your code Ask for a new correction Get a sandbox QA Review

### 3. Full deployment

mandatory

Score: 0.0% (Checks completed: 0.0%)

Write a Fabric script (based on the file `2-do_deploy_web_static.py`) that creates and distributes an archive to your web servers, using the function `deploy`:

- Prototype: `def deploy():`
- The script should take the following steps:
  - Call the `do_pack()` function and store the path of the created archive
  - Return `False` if no archive has been created
  - Call the `do_deploy(archive_path)` function, using the new path of the new archive
  - Return the return value of `do_deploy`
- All remote commands must be executed on both of web your servers (using `env.hosts = ['<IP web-01>', 'IP web-02']` variable in your script)
- You must use this script to deploy it on your servers: `xx-web-01` and `xx-web-02`

In the following example, the SSH key and the username used for accessing to the server are passed in the command line. Of course, you could define them as Fabric environment variables (ex: `env.user = ...`)

```
guillaume@ubuntu:~/AirBnB_clone_v2$ fab -f 3-deploy_web_static.py deploy -i my_ssh_private_key -u ubuntu
[52.55.249.213] Executing task 'deploy'
Packing web_static to versions/web_static_20170315015620.tgz
[localhost] local: tar -cvzf versions/web_static_20170315015620.tgz web_static
web_static/
web_static/0-index.html
web_static/1-index.html
web_static/100-index.html
web_static/2-index.html
web_static/3-index.html
web_static/4-index.html
web_static/5-index.html
web_static/6-index.html
```

```
web_static/7-index.html
web_static/8-index.html
web_static/images/
web_static/images/icon.png
web_static/images/icon_bath.png
web_static/images/icon_bed.png
web_static/images/icon_group.png
web_static/images/icon_pets.png
web_static/images/icon_tv.png
web_static/images/icon_wifi.png
web_static/images/logo.png
web_static/index.html
web_static/styles/
web_static/styles/100-places.css
web_static/styles/2-common.css
web_static/styles/2-footer.css
web_static/styles/2-header.css
web_static/styles/3-common.css
web_static/styles/3-footer.css
web_static/styles/3-header.css
web_static/styles/4-common.css
web_static/styles/4-filters.css
web_static/styles/5-filters.css
web_static/styles/6-filters.css
web_static/styles/7-places.css
web_static/styles/8-places.css
web_static/styles/common.css
web_static/styles/filters.css
web_static/styles/footer.css
web_static/styles/header.css
web_static/styles/places.css
web_static packed: versions/web_static_20170315015620.tgz -> 27280335Bytes
[52.55.249.213] put: versions/web_static_20170315015620.tgz -> /tmp/web_static_20170315015620.tgz
```

```
[52.55.249.213] run: mkdir -p /data/web_static/releases/web_static_20170315015620/
[52.55.249.213] run: tar -xzf /tmp/web_static_20170315015620.tgz -C /data/web_static/
releases/web_static_20170315015620/
[52.55.249.213] run: rm /tmp/web_static_20170315015620.tgz
[52.55.249.213] run: mv /data/web_static/releases/web_static_20170315015620/web_stati
c/* /data/web_static/releases/web_static_20170315015620/
[52.55.249.213] run: rm -rf /data/web_static/releases/web_static_20170315015620/web_s
tatic
[52.55.249.213] run: rm -rf /data/web_static/current
[52.55.249.213] run: ln -s /data/web_static/releases/web_static_20170315015620/ /data
/web_static/current
New version deployed!
[54.157.32.137] Executing task 'deploy'
[54.157.32.137] put: versions/web_static_20170315015620.tgz -> /tmp/web_static_201703
15015620.tgz
[54.157.32.137] run: mkdir -p /data/web_static/releases/web_static_20170315015620/
[54.157.32.137] run: tar -xzf /tmp/web_static_20170315015620.tgz -C /data/web_static/
releases/web_static_20170315015620/
[54.157.32.137] run: rm /tmp/web_static_20170315015620.tgz
[54.157.32.137] run: mv /data/web_static/releases/web_static_20170315015620/web_stati
c/* /data/web_static/releases/web_static_20170315015620/
[54.157.32.137] run: rm -rf /data/web_static/releases/web_static_20170315015620/web_s
tatic
[54.157.32.137] run: rm -rf /data/web_static/current
[54.157.32.137] run: ln -s /data/web_static/releases/web_static_20170315015620/ /data
/web_static/current
New version deployed!

Done.
Disconnecting from 54.157.32.137... done.
Disconnecting from 52.55.249.213... done.
guillaume@ubuntu:~/AirBnB_clone_v2$
guillaume@ubuntu:~/AirBnB_clone_v2$ curl 54.157.32.137/hbnb_static/0-index.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
```

```
<title>AirBnB clone</title>
</head>
<body style="margin: 0px; padding: 0px;">
  <header style="height: 70px; width: 100%; background-color: #FF0000">
    </header>

  <footer style="position: absolute; left: 0; bottom: 0; height: 60px; width: 100%; background-color: #00FF00; text-align: center; overflow: hidden;">
    <p style="line-height: 60px; margin: 0px;">Holberton School</p>
  </footer>
</body>
</html>
guillaume@ubuntu:~/AirBnB_clone_v2$
```

## Repo:

- GitHub repository: [AirBnB\\_clone\\_v2](#)
- File: [3-deploy\\_web\\_static.py](#)

Done? Help Check your code Ask for a new correction Get a sandbox QA Review

Done with the mandatory tasks? Unlock 2 advanced tasks now!