0x03. AirBnB clone - Deploy static

Deadline: Python Sysadmin Scripting CORS

Concepts

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

- CORS
- 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 `Nginx` the Python/3. Fabric is a Python library and command-line tool for streamlining the use of SSH for application deployment or system administration tasks. It provides a basic suite of operations for executing commands on remote servers and transferring files over SSH. Local pseudoglobbering/wildcards, as well as auxiliary functionality such as prompting for password/passphrase input, is also provided. The intent is to provide a lightweight, non-magical, Pythonic way of scripting such tasks. Fabric 1.x is a stable, mature, well-documented, well-supported library. Fabric 2 is a newer, backward-incompatible rewrite that adds support for SSH v2 and other features, but is currently lacking some of all network conveniences (SSH, SCP etc.). It is an ideal tool for maintaining, executing etc. commands from locale to a remote server.

Before starting, please fork the repository `AirBnB_clone_v4` 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
- CORS command page
- Fabric Documentation for Beginners
- Differences between 1x0 and also 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 `fabfile.py`
- 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 `add` and `alias` in 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 20.04 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/local/bin/fabric3`
- A `README.md` file at the root of the folder of the project is mandatory
- All your files must be executable, `pep8` style compliant
- Your Fabric file must end with `env.hosts = ['localhost']` (Installation instructions 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 (`python -c "print(my_module.my_function.__doc__)"` and `python -c "print(my_module.MyClass.__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 20.04 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 `0.3.8` or `0.7.0`) 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

```
$ pip install Fabric
$ sudo apt-get install libffi-dev
$ sudo apt-get install libssl-dev
$ sudo apt-get install build-essential
$ sudo apt-get install libssl-dev
$ sudo apt-get install libpython3-dev
$ pip install paramiko
$ pip install wheel
$ pip install setuptools==41.1.0
$ pip install cryptography==2.8
$ pip install bcrypt==3.1.7
$ pip install pyasn1==0.4.8
$ pip install fabric3==1.14.post1
```

Video library 2 total

Search by title

Deploy Static Fabric
Code With Fabric
alx

Display static files with Fabric

Quiz questions

Great! You've completed the quiz successfully! Keep going! Show quiz

Tasks

0. Prepare your web servers

mandatory

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

- Install Nginx if 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/releases/test/` if it doesn't already exist
- Create the folder `/data/web_static/shared/` 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 `/data/web_static/current` 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 owned by this user/group.
- Update the Nginx configuration to serve the content of `/data/web_static/current/` to `hbnb_static` (ex: `https://mydomain.tech/hbnb_static`). Do not forget to restart Nginx after updating the configuration:
  - `location /hbnb_static/` to serve your Nginx configuration

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

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

```
ubuntu@89-ed-01:~$ sudo ./0-setup_web_static.sh
ubuntu@89-ed-01:~$ echo $?
0
ubuntu@89-ed-01:~$ ls -l /data
total 4
drwxr-xr-x 3 data data  4096 Mar 7 15:17 web_static
ubuntu@89-ed-01:~$ ls -l /data/web_static/
total 8
lrwxrwxrwx 1 data data  34 Mar 7 15:18 current -> /data/web_static/releases/test/
drwxr-xr-x 2 root root 4096 Mar 7 15:18 releases
drwxrwxrwx 2 root root 4096 Mar 7 15:18 shared
ubuntu@89-ed-01:~$ ls /data/web_static/releases/test/
index.html
ubuntu@89-ed-01:~$ cat /data/web_static/releases/test/index.html
<html>
  <head>
  </head>
  <body>
    Holberton School
  </body>
</html>
ubuntu@89-ed-01:~$ curl localhost:9000/hbnb_static/index.html
<html>
  <head>
  </head>
  <body>
    Holberton School
  </body>
</html>
ubuntu@89-ed-01:~$
```

Repo:

- GitHub repository: `AirBnB_clone_v3`
- File: `0-setup_web_static.sh`

Done! | Check your code | Ask for a new correction | My Results

1. Compress before sending

mandatory

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_yyyyMMddHHmmss.tgz`
- The function `do_pack` must return the archive path if the archive has been correctly generated. Otherwise, it should return `None`

```
guillaume@ubuntu:~$(hbtn_venv_20.04) fab -f 1-pack_web_static.py do_pack
Packing web_static to versions/web_static_20170324201207.tgz
[localhost] local: tar -cvzf versions/web_static_20170324201207.tgz web_static
web_static/
web_static/_os_store
web_static/styles/1-hbnb.less
web_static/styles/3-footer.less
web_static/styles/4-common.less
web_static/styles/6-media.less
web_static/styles/2-header.less
web_static/styles/5-filters.less
web_static/styles/7-places.less
web_static/styles/8-places_extra.less
web_static/styles/9-places_search.less
web_static/styles/10-places_owner.less
web_static/styles/0-reset.less
web_static/styles/11-places_amenities.less
web_static/styles/12-places_reviews.less
web_static/styles/13-places_map.less
web_static/styles/14-media_queries.less
web_static/styles/15-icons.less
web_static/styles/16-icon_fonts.less
web_static/styles/17-icon_fonts_extra.less
web_static/styles/18-icon_fonts_owner.less
web_static/styles/19-icon_fonts_search.less
web_static/styles/20-icon_fonts_places.less
web_static/styles/21-icon_fonts_amenities.less
web_static/styles/22-icon_fonts_filters.less
web_static/styles/23-icon_fonts_reviews.less
web_static/styles/24-icon_fonts_map.less
web_static/styles/25-icon_fonts_media.less
web_static/styles/26-icon_fonts_icons.less
web_static/styles/27-icon_fonts_extra.less
web_static/styles/28-icon_fonts_owner.less
web_static/styles/29-icon_fonts_search.less
web_static/styles/30-icon_fonts_places.less
web_static/styles/31-icon_fonts_amenities.less
web_static/styles/32-icon_fonts_filters.less
web_static/styles/33-icon_fonts_reviews.less
web_static/styles/34-icon_fonts_map.less
web_static/styles/35-icon_fonts_media.less
web_static/styles/36-icon_fonts_icons.less
web_static/styles/37-icon_fonts_extra.less
web_static/styles/38-icon_fonts_owner.less
web_static/styles/39-icon_fonts_search.less
web_static/styles/40-icon_fonts_places.less
web_static/styles/41-icon_fonts_amenities.less
web_static/styles/42-icon_fonts_filters.less
web_static/styles/43-icon_fonts_reviews.less
web_static/styles/44-icon_fonts_map.less
web_static/styles/45-icon_fonts_media.less
web_static/styles/46-icon_fonts_icons.less
web_static/styles/47-icon_fonts_extra.less
web_static/styles/48-icon_fonts_owner.less
web_static/styles/49-icon_fonts_search.less
web_static/styles/50-icon_fonts_places.less
web_static/styles/51-icon_fonts_amenities.less
web_static/styles/52-icon_fonts_filters.less
web_static/styles/53-icon_fonts_reviews.less
web_static/styles/54-icon_fonts_map.less
web_static/styles/55-icon_fonts_media.less
web_static/styles/56-icon_fonts_icons.less
```

```
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_2017
0315015620/
[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_static/* /data/web_static/rel
eases/web_static_20170315015620/
[52.55.249.213] run: rm -rf /data/web_static/releases/web_static_20170315015620/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_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_20170315015620.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_2017
0315015620/
[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_static/* /data/web_static/rel
eases/web_static_20170315015620/
[54.157.32.137] run: rm -rf /data/web_static/releases/web_static_20170315015620/web_static
[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? | Check your code | Ask for a new correction | Get a sandbox | QA Review

4. Keep it clean!

#advanced

Write a Fabric script (based on the file `3-deploy_web_static.py`) that deletes out-of-date archives, using the function `do_clean`:

- Prototype: `def do_clean(number=0):`
- `number` is the number of the archives, including the most recent, to keep.
  - If `number` is 0 or 1, keep only the most recent version of your archive.
  - if `number` is 2, keep the most recent, and second most recent versions of your archive.
  - etc.
- Your script should:
  - Delete all unnecessary archives (all archives minus the number to keep) in the `versions` folder
  - Delete all unnecessary archives (all archives minus the number to keep) in the `/data/web_static/releases` folder of both of your web servers
- All remote commands must be executed on both of your web servers (using the `env.hosts = ['<IP web-01>', 'IP web-02']` variable in your script)

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$ ls -ltr versions
-rw-r--r-- 1 vagrant vagrant 27280335 Mar 15  2017 web_static_20170315015414.tgz
-rw-r--r-- 1 vagrant vagrant 27280335 Mar 15  2017 web_static_20170315015448.tgz
-rw-r--r-- 1 vagrant vagrant 27280335 Mar 15  2017 web_static_20170315015507.tgz
-rw-r--r-- 1 vagrant vagrant 27280335 Mar 15  2017 web_static_20170315015620.tgz
guillaume@ubuntu:~/AirBnB_clone_v2$ fab -f 100-clean_web_static.py do_clean:number=2 -i my_ssh_private_key -u
ubuntu > /dev/null 2>&1
guillaume@ubuntu:~/AirBnB_clone_v2$ ls -ltr versions
-rw-r--r-- 1 vagrant vagrant 27280335 Mar 15  2017 web_static_20170315015507.tgz
-rw-r--r-- 1 vagrant vagrant 27280335 Mar 15  2017 web_static_20170315015620.tgz
guillaume@ubuntu:~/AirBnB_clone_v2$
```

Repo:

- GitHub repository: `AirBnB_clone_v2`
- File: `100-clean_web_static.py`

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

5. Puppet for setup

#advanced

Redo the task #0 but by using Puppet:

```
ubuntu@89-web-01:~/$ puppet apply 101-setup_web_static.pp
....
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 root root   30 Mar 7 22:30 current -> /data/web_static/releases/test
drwxr-xr-x 3 root root 4096 Mar 7 22:29 releases
drwxr-xr-x 2 root root 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: `101-setup_web_static.pp`

Done! | Check your code | QA Review

Copyright © 2024 ALX, All rights reserved.