Manual For Demo Vedraxx Python

Developmented by La Katuna

Select:

Indications

For init the program, run the script **call_menus.py.**

If there is some error, can be, it has not libraries corresponding installed, it is necessary install:

Paramiko, yaml, Json, requests, select.

1. For choose an option, enter the corresponding number.

```
**_____ Main Menu _____**
Docker-compose(0)
Execute command(1)
Configuration(2)
Insert option:
```

a. **Docker-Compose**
   1. Enter the name file.
   2. For Add some service as engine, permit or random-generator, enter "y" or "yes", but in the case of not add some service, enter "no" or press "Enter" key from keyboard.
   3. For enter data corresponding for environment variables, write and press "Enter" key, but if not wish change data from ".env" downloaded from repository Github, did not write only press "Enter" key.

```
**_____ Docker-compose Menu_____**
Folder Ok!.
0.Installation name: test3
Creating folder...
Created Folder:  .\files_compose\test3
Copying file >>  docker-compose.yaml
Copying file >>  .env
1.Add engine?(optional) yes/not: y
2.Add transit-random-generator?(optional): yes
3.Add transit-permit-ext?(optional):
4.Digit maintainer email: maintainer@lakatuna.com
5.Host/Ip Address:
6.Digits version(lastest or RC): 5.331
File .yaml Created!
Folder Ok!.
File .env created!
No exist this name config: test3 in the file Json, for store in format .env
```

The files '.env' and '.yaml' be create successfully in main folder with name instanced from dataV.py.
In the last line, mean that installation name create recently has not a configuration registred, so go to main menu / Configuration / Add.

b. **Execute Command**
   It has enabled a shell, where the following commands are valid for execute.
   - vedraxx release
   - vedraxx install **name_docker_compose**

- vedraxx up **name_docker_compose**
- vedraxx down **name_docker_compose**
- vedraxx install docker
- vedraxx copy **path_local _folder   path_remote_server**

Other commands:

- back (this command back to menu)
- shell (this command calls a simple shell connected to server remote)
- shell2 (this command calls an interactive shell connected to server remote)

c. Configuration

For choose an option, enter the corresponding number.

```
**_____ Configuration Menu_____**
Add(1)
Update(2)
Delete configuration(3)
Back(4)
Insert option: |
```

For whatever option is chosen, if you wish enter corresponding data, otherwise only press "Enter" key for pass to next item, without need to fill the fields.

The information entered will be store in a file Json, for after convert in file .env; the folder's name is instance from dataV.py.

```
**_____ Add Menu_____**
Configuration Name: test1
This name already exist, try with other name.


Configuration Name: test2
This name already exist, try with other name.


Configuration Name: test3
Add remote server ip: 122.34.33.2
Add remote port: 44
Add registry(optional):
Add registry username(optional):
Add registry password(optional):
Folder Ok!.
Folder Ok!.
File .env created!
```

Recommendation

- Modify the script dataV.py for enter your corresponding information as credentials from server remote, registry, link repository Github, the other fields are not necessary change them.
- Recommended example link registry: **https://registry.domain/v2**, because is necessary get file Json through **"/_catalog/name_service/tags/list"**
- If be throw an error in the program to show menu or get keys ".env", you must generate Json menu and keys_env, so run the script **generateJson_Menu_&_Keys.py**
- If be throw an error in the program to copy directory to server remote, check the path local has not spaces in the name each folder, same for path remote.
- It is recommended that file name docker-compose has not spaces.
- In the repository github, the file .env & docker-compose.yaml must be uploaded to root.
- If the shell is freeze or hold, sometimes it happens due to excess data in the buffer shell, you must wait or reset the program, then, invoke the shell executing the command **"shell"**, after, write the command for verify if the file docker-compose or docker is run or if this was installed.
  - sudo docker version
  - sudo docker-compose - - version
  - sudo docker ps
  - sudo docker images -a

Test Demo

1. Run script call_menus.py.
2. Enter 0 and press enter key.

```
**_____ Main Menu _____**
Docker-compose(0)
Execute command(1)
Configuration(2)
Insert option: |
```

3. Write TestDemo, then press enter key.
4. Wait while cloning repository github with files '.env' & '.yaml'.
5. Write y or yes.
6. Write no or press enter key.
7. Write yes.
8. If you wish write a maintainer email do it, or only press enter key for keep default data of file downloaded.
9. If you wish write a Host/IP do it, or only press enter key for keep default data of file downloaded.
10. If you wish write a new Version do it, or only press enter key for keep default data of file downloaded.
11. Wait created the new files.

```
**_____ Docker-compose Menu_____**
Folder Ok!.
0.Installation name: testDemo
Creating folder...
Created Folder:   .\files_compose\testDemo
C:\Users\gta20\AppData\Local\Temp\tmp_3h1m08j
Cloning into 'C:\Users\gta20\AppData\Local\Temp\tmp_3h1m08j'...
Moving file >>  docker-compose.yaml
Copying file >>  docker-compose.yaml
Moving file >>   .env
Copying file >>   .env
1.Add engine?(optional) yes/not: y
2.Add transit-random-generator?(optional):
3.Add transit-permit-ext?(optional): yes
4.Digit maintainer email:
5.Host/Ip Address: 125.31.22.3
6.Digits version(lastest or RC): 1.005
File .yaml Created!
Folder Ok!.
File .env created!
```

12. Now enter number 2 for go to menu Configuration.
13. Enter number 1 for add a new configuration.
14. Write the same name as called the docker-compose file, in this case testDemo.
15. Write IP 127.58.22.5.

16. If you wish write a Remote Port, do it, or only press enter key, because in the option Update you can modify this field.
17. If you wish write a Registry, do it, or only press enter key, because in the option Update you can modify this field.
18. If you wish write a Registry Username, do it, or only press enter key, because in the option Update you can modify this field.
19. If you wish write a Registry Password, do it, or only press enter key, because in the option Update you can modify this field.
20. Wait while creating file with information.
21. Now enter number 2, for update the config.
22. Write the configuration's name that you want to modify, in this case testDemo.
23. Enter Port 22.
24. Enter Registry: resgistry.example.com/v2.
25. Enter User: admin.
26. Enter Pass: admin.
27. Wait create file .env in the folder called testDemo created in the first steps, and the file Json is update with the new data.

```
**_____ Update Menu_____**
List of all configuration availables:
-  testDemo
Enter the configuration name: testDemo
Add remote port: 22
Add registry(optional): resgistry.example.com/v2
Add registry username(optional): admin
Add registry password(optional): admin
Folder Ok!.
Folder Ok!.
File .env created!
```

28. Enter number 4 for back to main menu.
29. Now, enter option 1, for connect to server remote and execute commands.
30. Enter **vedraxx release** for display all versions of images from registry.

```
> image:  dev.phcodes.company
 -  31
 -  realese.32
 -  5
 -  dev
 -  1005917977
 -  latest
 -  merge
 -  1005926399
```

31. Enter **vedraxx install testDemo** for copy the folder to default path remote server.

```
>>Digit Command: vedraxx install testDemo
Command Ok!
>>  /home/test
Creating folder... >>  /home/test/testDemo
Copying file... >>
local: C:\Users\gta20\Documents\ProjectsPyCharm\Code_Menu2\Code_Menu\files_compose\testDemo\.env to
remote: /home/test/testDemo/.env
Complete precent: 100.00 %Copying file... >>
local: C:\Users\gta20\Documents\ProjectsPyCharm\Code_Menu2\Code_Menu\files_compose\testDemo\testDemo_docker-compose.yaml to
remote: /home/test/testDemo/testDemo_docker-compose.yaml
Complete precent: 100.00 %Copied Succesful in path_remote >>  /home/test/testDemo
```

In the end line, you can see the path remote where it was copied the folder.

32. Execute the next command **shell** for invoke a simple shell for verify if the folder was copied successfully.
33. Enter **cd /home/test/testDemo; ls -a** remember that path used is in the end line of previous command, then enter **back** for back to shell vedraxx.

```
>>Digit Command: shell
cmd shell >> cd /home/test/testDemo; ls -a
Command Ok!
>>  .  ..  .env  testDemo_docker-compose.yaml
cmd shell >> back
Shell closed!
No exist command!
Return menu input >> back
>>Digit Command: |
```

If you do not have installed docker & docker-compose, make this step, otherwise omit it.

34. Enter **vedraxx install docker**
    it will begin download a version docker stable for start install, be patient while execute command.
35. If docker was installed successfully, the terminal will display a hello-world image running in docker, otherwise display command incorrect, then try again or install it manually.

```
Hello from Docker!
This message shows that your installation appears to be working correctly.
To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/
For more examples and ideas, visit:
 https://docs.docker.com/get-started/
$
```

36. For verify that docker-compose was installed correctly, execute **shell2** and write **docker-compose - - version,** after enter **back,** so back to shell vedraxx**.**

```
>>Digit Command: shell
cmd shell >> docker-compose --version
Command Ok!
>>  docker-compose version 1.29.2, build 5becea4c
```

For test docker-compose up & down, there is a folder call wordpress-mysql in the folder files-compose, this file compose only is is a performance, for comprobate that next command works correctly.

37. Execute command **vedraxx up wordpress-mysql**
    This command first will copy the folder to default path remote, after it will download the imagen and will be install it.

```
Creating volume "wordpress-mysql_db_data" with default driver
Pulling db (mysql:8.0.19)...
8.0.19: Pulling from library/mysql
54fec2fa59d0: Pulling fs layer
bcc6c6145912: Pulling fs layer
951c3d959c9d: Pulling fs layer
05de4d0e206e: Pulling fs layer
319f0394ef42: Pulling fs layer
d9185034607b: Pulling fs layer
013a9c64dadc: Pulling fs layer
96d4c3d31f9f: Pulling fs layer
785bc90808da: Pulling fs layer
1339cf094729: Pulling fs layer
beb8f531cc37: Pulling fs layer
2b40c9f6a918: Pulling fs layer05de4d0e206e: Waiting319f0394ef42: Waitingd9185034607b: Waiting013a9c64
013a9c64dadc: Downloading [==================================================>]  2.395kB/2.395kB013a9
54fec2fa59d0: Extracting [====================================>              ]  20.94MB/27.1MB785bc90
bcc6c6145912: Pull complete951c3d959c9d: Extracting [>
d9185034607b: Extracting [==========================>                        ]  7.537MB/13.44MBd91850
785bc90808da: Extracting [=>                                                 ]  2.785MB/113MB785bc908
785bc90808da: Extracting [===========>                                       ]  25.62MB/113MB785bc908
785bc90808da: Extracting [====================>                              ]  45.68MB/113MB785bc908
785bc90808da: Extracting [============================>                      ]  64.06MB/113MB785bc908
785bc90808da: Extracting [====================================>              ]  83.56MB/113MB785bc908
```

If your terminal displayed similar to next image, then your docker-compose up was executed successfully, else try again entering previous command.

```
Starting wordpress-mysql_wordpress_1
Starting wordpress-mysql_db_1
Waiting...
Starting wordpress-mysql_wordpress_1
doneStarting wordpress-mysql_db_1
doneAttaching to wordpress-mysql_wordpress_1, wordpress-mysql_db_1
wordpress_1  | WordPress not found in /var/www/html - copying now...
wordpress_1  | Complete! WordPress has been successfully copied to /var/www/html
wordpress_1  | No 'wp-config.php' found in /var/www/html, but 'WORDPRESS_...' var
wordpress_1  | AH00558: apache2: Could not reliably determine the server's fully
wordpress_1  | AH00558: apache2: Could not reliably determine the server's fully
wordpress_1  | [Fri Jul 16 18:53:04.363819 2021] [mpm_prefork:notice] [pid 1] AH0
wordpress_1  | [Fri Jul 16 18:53:04.364188 2021] [core:notice] [pid 1] AH00094: C
db_1         | 2021-07-16 18:53:04+00:00 [Note] [Entrypoint]: Entrypoint script f
db_1         | 2021-07-16 18:53:05+00:00 [Note] [Entrypoint]: Switching to dedica
db_1         | 2021-07-16 18:53:05+00:00 [Note] [Entrypoint]: Entrypoint script f
db_1         | 2021-07-16 18:53:05+00:00 [Note] [Entrypoint]: Initializing databa
```

38. Let's go verify with an interactive shell if this service is running, for this write command **shell2** and after entering **sudo docker ps,** write back, so back to shell vedraxx.

```
>>Digit Command: shell2
cmd shell2 >> sudo docker ps
Waiting...
Last login: Fri Jul 16 18:52:58 2021 from 179.49.44.209
$ sudo docker ps
[sudo] password for test:
CONTAINER ID   IMAGE             COMMAND              CREATED         STATUS         PORTS                               NAMES
82878ce12243   mysql:8.0.19      "docker-entrypoint.s…"   22 minutes ago  Up 5 minutes   3306/tcp, 33060/tcp                 wordpress-mysql_db_1
47af2566507c   wordpress:latest  "docker-entrypoint.s…"   22 minutes ago  Up 5 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp   wordpress-mysql_wordpress_1
$

Finish command.
```

39. Execute command **vedraxx down wordpress-mysql** for stop the service.

```
Stopping wordpress-mysql_db_1
Stopping wordpress-mysql_wordpress_1
Waiting...
Stopping wordpress-mysql_wordpress_1
doneStopping wordpress-mysql_db_1
doneRemoving wordpress-mysql_db_1
Removing wordpress-mysql_wordpress_1
Removing wordpress-mysql_db_1
doneRemoving wordpress-mysql_wordpress_1
doneRemoving network wordpress-mysql_default
```

40. Let's go verify with an interactive shell if this service was stopped, so write command **shell2** and after entering **sudo docker ps**, then, write **back**, so back to shell vedraxx.

```
>>Digit Command: shell2
cmd shell2 >> sudo docker ps
Waiting...
Last login: Fri Jul 16 19:00:12 2021 from 179.49.44.209
$ sudo docker ps
[sudo] password for test:
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS   PORTS   NAMES
$
```

Copy a local directory to server remote specific, for this, enter to folder files_compose and copy the path local, in this example I worked in Windows.

41. Enter command **vedraxx copy C:\Users\Code_Menu\files_compose /home/test/testDemo**

```
>>Digit Command: vedraxx copy C:\Users\gta20\Documents\ProjectsPyCharm\Code_Menu2\Code_Menu\files_compose /home/test/testDemo
Creating folder... >>  /home/test/testDemo/files_compose
Copying file... >>
local: C:\Users\gta20\Documents\ProjectsPyCharm\Code_Menu2\Code_Menu\files_compose\.env to
remote: /home/test/testDemo/files_compose/.env
Complete precent: 100.00 %Copying file... >>
local: C:\Users\gta20\Documents\ProjectsPyCharm\Code_Menu2\Code_Menu\files_compose\docker-compose.yaml to
remote: /home/test/testDemo/files_compose/docker-compose.yaml
Complete precent: 100.00 %Creating folder... >>  /home/test/testDemo/files_compose/testDemo
Created sub-folder... >>  /home/test/testDemo/files_compose/testDemo
Creating folder... >>  /home/test/testDemo/files_compose/wordpress-mysql
Created sub-folder... >>  /home/test/testDemo/files_compose/wordpress-mysql
Copying file... >>
local: C:\Users\gta20\Documents\ProjectsPyCharm\Code_Menu2\Code_Menu\files_compose\testDemo\.env to
remote: /home/test/testDemo/files_compose/testDemo/.env
```

42. Enter command **shell** after enter **cd /home/test/testDemo; ls**

```
>>Digit Command: shell
cmd shell >> cd /home/test/testDemo/; ls
Command Ok!
>>  files_compose  testDemo_docker-compose.yaml
```

43. Enter **cd /home/test/ testDemo /files_compose; ls**

```
cmd shell >> cd /home/test/testDemo/files_compose; ls
Command Ok!
>>  docker-compose.yaml  testDemo  wordpress-mysql
cmd shell >>
```

44. Enter **back** in each shell, so back to main menu.

Finish.