REPORT P0

Getting ready for AggieStack

CSCE 678 Distributed Systems and Cloud Computing

Fall 2018

Submitted by

Ashima Sharma: 426008924

Nikhil Gupta: 526003001

Index

Project Repository

https://github.tamu.edu/nikhil-gupta/678-18-c

Contributors:

- Ashima Sharma
- Nikhil Gupta

Tools Used

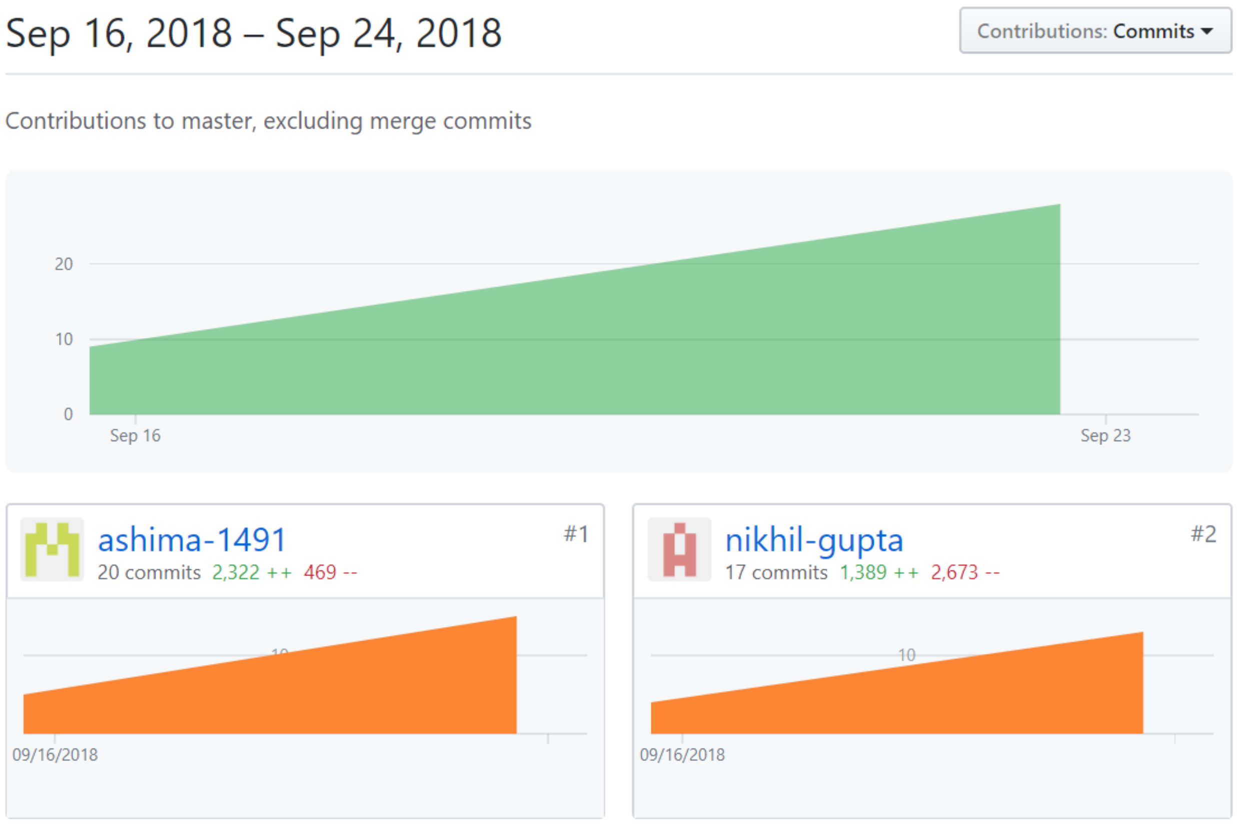
IDE: Eclipse with PyDev plugin

Version Control: GitBash

Language: Python

Database: MongoDB Cloud

Timeline and Architecture



Sep 16, 2018 – Sep 24, 2018

Contributions: Commits

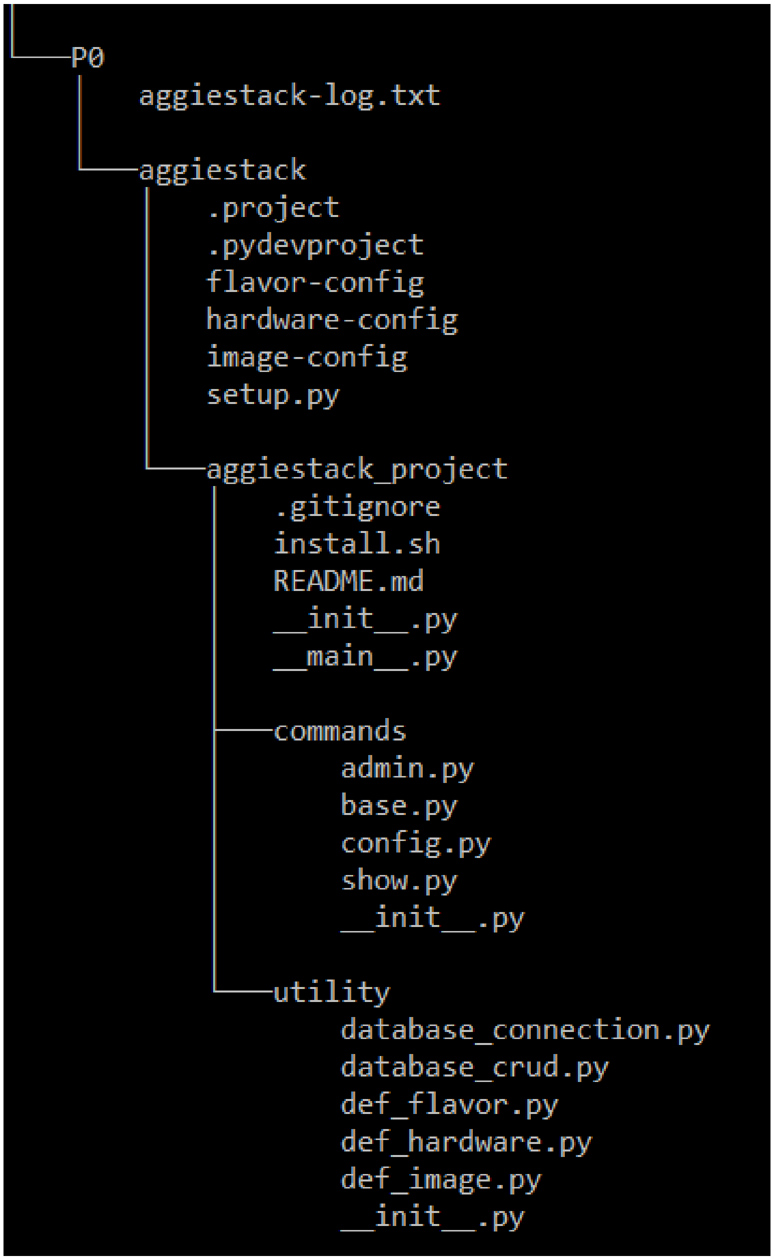
Contributions to master, excluding merge commits

ashima-1491 #1
20 commits 2,322 ++ 469 --

nikhil-gupta #2
17 commits 1,389 ++ 2,673 --

We made this project in a span of 5 days, putting around 10 hours each day per team member. We started researching on how to make CLI apps using python and its libraries to parse text. We studied many online tutorials on how to design a CLI app and found a standard approach of parsing the doc string using “Docopt” and creating commands. We have used MongoDb cloud as a database to store the data from config files. We selected a database over the file system as a storage medium, to keep our application flexible enough so that we can add more functions in the future, hence more scalable. Also, having a database makes the CRUD operations easier. Some important features of our application have been listed below:

- **The data files**: hardware-config, flavor-config, image-config have been kept at the root of the project so that when user gives the config command, the file can be read
- **Handling Duplicity**: If a config file is loaded multiple times, the old duplicate entries will be deleted from the database, and new entries will be inserted from the file.
- **Handling Dynamic configuration for machine**: The RAM, num of disks and number of virtual CPU is stored into two sets of columns: “original” and “current”. The values for “original” will remain intact and the values for “current” will be updated in future. Consequently, admin can see the “current” and “original” values. But a regular user can only see the “original” i.e. default configuration.
- **File system architecture**:
  - Below is the tree structure of the files inside our Project named 678-18-c. Our root folder 678-18-c contains P0 folder.

- 

```
│
└───P0
    │   aggiestack-log.txt
    │
    └───aggiestack
        │   .project
        │   .pydevproject
        │   flavor-config
        │   hardware-config
        │   image-config
        │   setup.py
        │
        └───aggiestack_project
            │   .gitignore
            │   install.sh
            │   README.md
            │   __init__.py
            │   __main__.py
            │
            ├───commands
            │       admin.py
            │       base.py
            │       config.py
            │       show.py
            │       __init__.py
            │
            └───utility
                    database_connection.py
                    database_crud.py
                    def_flavor.py
                    def_hardware.py
                    def_image.py
                    __init__.py
```

- **Logging**:

  As per the requirements of the project, we have a log file aggiestack-log.txt inside folder P0, that includes all the activity performed in the CLI.

  The log will have following properties:

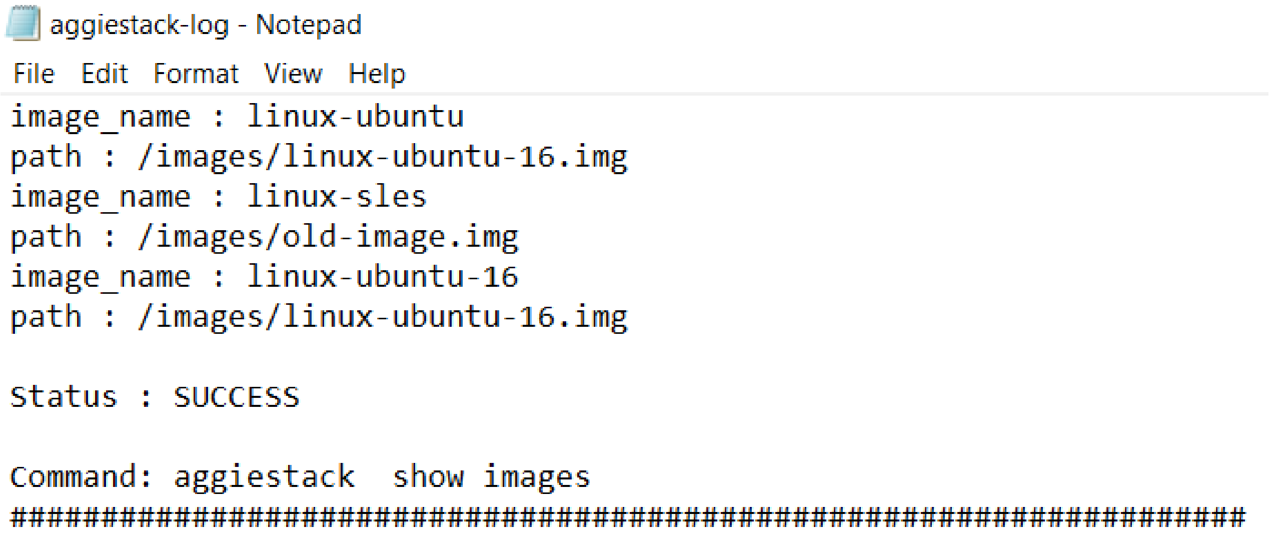
  - Stores any command typed on CLI.

- Since we have put try-except functions in our code, the log will store most of the exceptions or error messages.
- The log will catch errors such as: invalid filename, non-existing configuration files, and invalid format in the files

  The log file will have the following format:
  - Output generated by the command entered in CLI
  - Status - SUCCESS/FAILURE
  - Command entered by the user

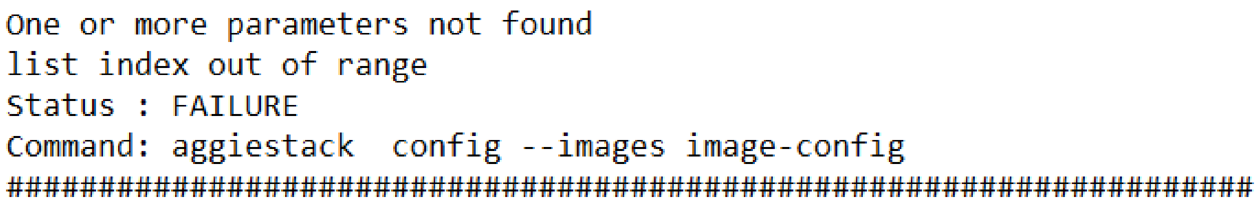
Sample screenshots are attached:

Status: SUCCESS →



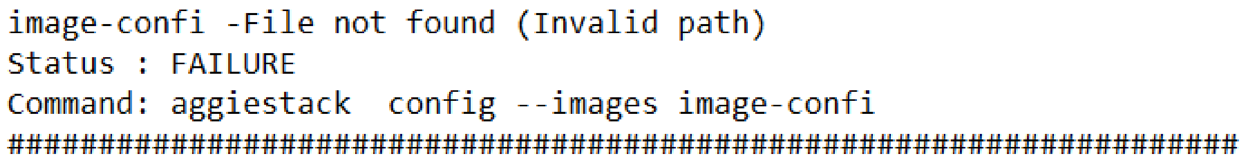
```
image_name : linux-ubuntu
path : /images/linux-ubuntu-16.img
image_name : linux-sles
path : /images/old-image.img
image_name : linux-ubuntu-16
path : /images/linux-ubuntu-16.img

Status : SUCCESS

Command: aggiestack  show images
####################################################################
```

Status: FAILURE →



```
One or more parameters not found
list index out of range
Status : FAILURE
Command: aggiestack  config --images image-config
####################################################################
```

Status: FAILURE →



```
image-confi -File not found (Invalid path)
Status : FAILURE
Command: aggiestack  config --images image-confi
########################################################################
```

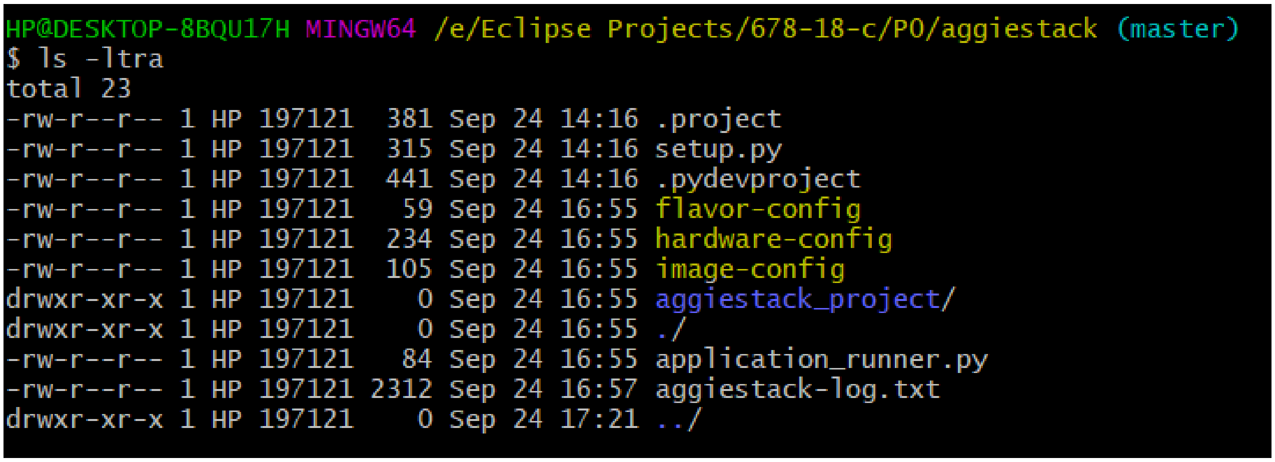
HOW TO RUN THE PROJECT

Note: Our project works fine on Windows machine. We did not test this on a Mac machine due to non-availability of the same.

Method 1: Running via pip install command

1. Download our archive file named:
2. Navigate to path: \678-18-c\P0\aggiestack
3. Run the command: ***pip install .***
4. The above command will install our project.
5. You might have to install some python modules such as dnspython, pymongo etc.
6. Now run any of the following commands :
    a. aggiestack config --hardware <filename>
    b. aggiestack config --images <filename>
    c. aggiestack config --flavors <filename>
    d. aggiestack admin show hardware
    e. aggiestack admin can_host <machinename> <flavor>
    f. aggiestack show hardware
    g. aggiestack show images
    h. aggiestack show flavors
    i. aggiestack show all

7. The above commands will give results depending upon the commands. For the commands that require <filename>, the files have been put in the path: \678-18-c\P0\aggiestack\ as shown in the screenshot given below:



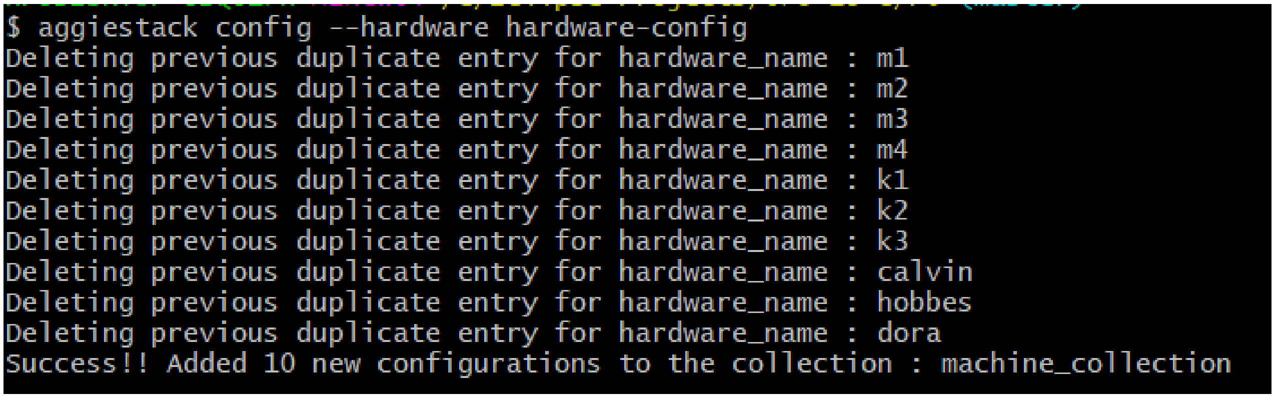
```
HP@DESKTOP-8BQU17H MINGW64 /e/Eclipse Projects/678-18-c/P0/aggiestack (master)
$ ls -ltra
total 23
-rw-r--r-- 1 HP 197121  381 Sep 24 14:16 .project
-rw-r--r-- 1 HP 197121  315 Sep 24 14:16 setup.py
-rw-r--r-- 1 HP 197121  441 Sep 24 14:16 .pydevproject
-rw-r--r-- 1 HP 197121   59 Sep 24 16:55 flavor-config
-rw-r--r-- 1 HP 197121  234 Sep 24 16:55 hardware-config
-rw-r--r-- 1 HP 197121  105 Sep 24 16:55 image-config
drwxr-xr-x 1 HP 197121    0 Sep 24 16:55 aggiestack_project/
drwxr-xr-x 1 HP 197121    0 Sep 24 16:55 ./
-rw-r--r-- 1 HP 197121   84 Sep 24 16:55 application_runner.py
-rw-r--r-- 1 HP 197121 2312 Sep 24 16:57 aggiestack-log.txt
drwxr-xr-x 1 HP 197121    0 Sep 24 17:21 ../
```

8. You can see the logs in the log aggiestack-log.txt stored at location path: \678-18-c\P0\aggiestack

Important Note: The log file will be created where we run the command. So, it's advisable to run the command in the path \678-18-c\P0\aggiestack so it appends all CLI activities in the log file.

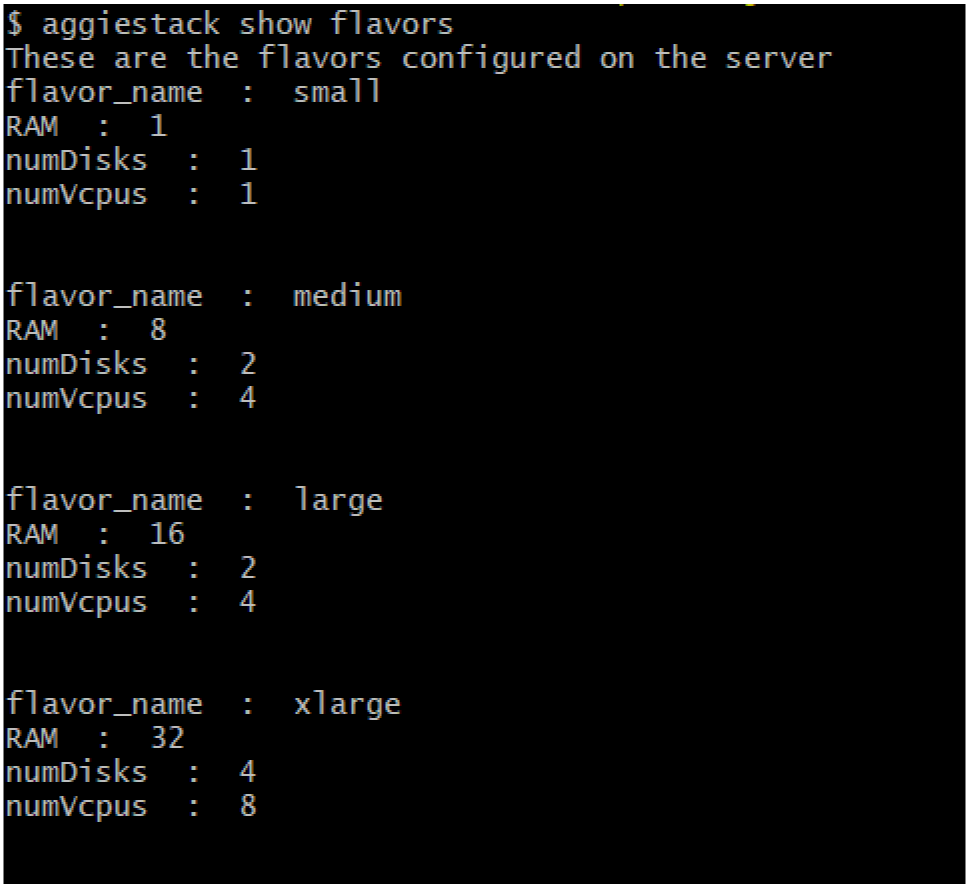
Screenshots:

aggiestack config --hardware hardware-config



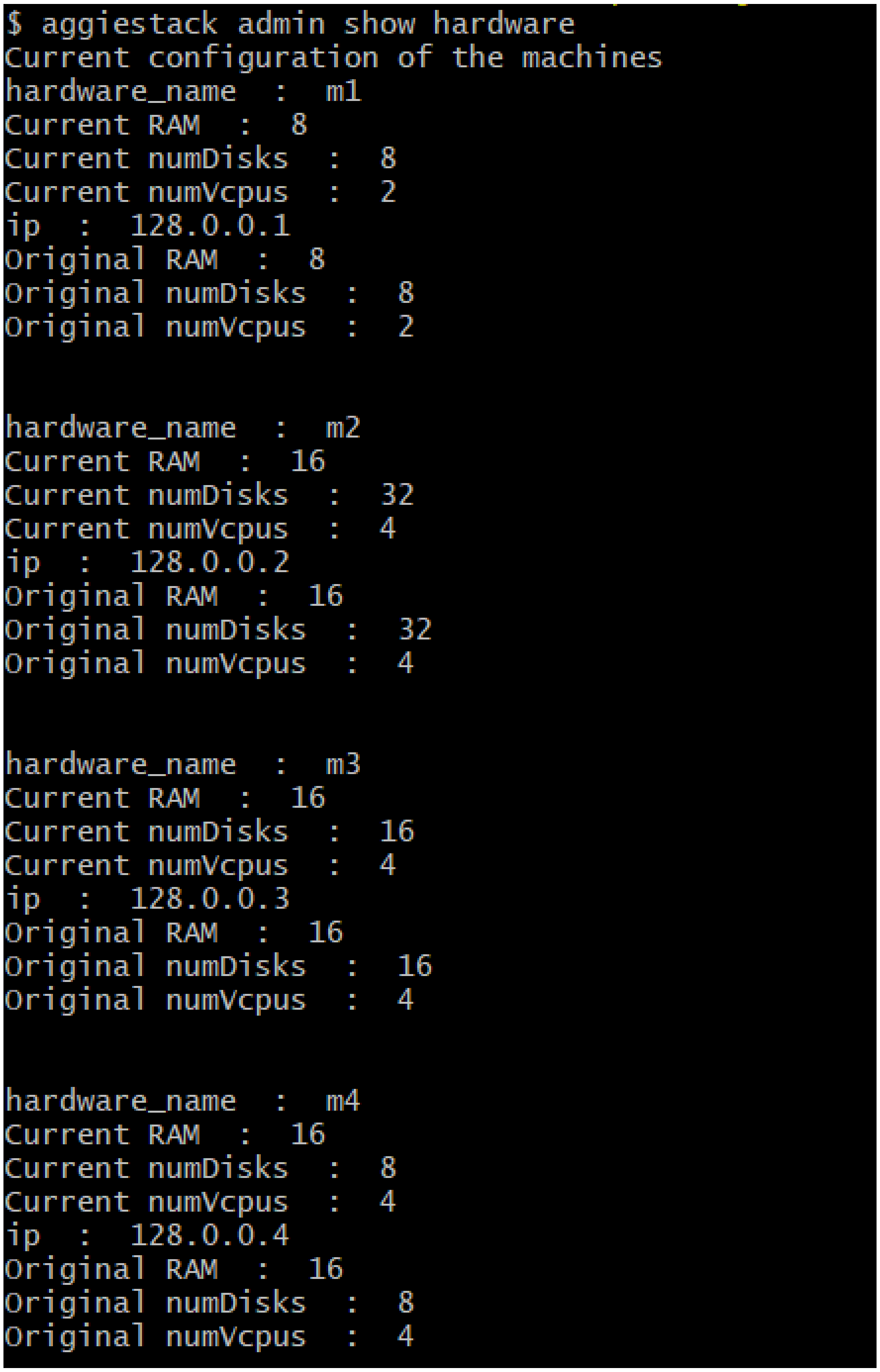
```
$ aggiestack config --hardware hardware-config
Deleting previous duplicate entry for hardware_name : m1
Deleting previous duplicate entry for hardware_name : m2
Deleting previous duplicate entry for hardware_name : m3
Deleting previous duplicate entry for hardware_name : m4
Deleting previous duplicate entry for hardware_name : k1
Deleting previous duplicate entry for hardware_name : k2
Deleting previous duplicate entry for hardware_name : k3
Deleting previous duplicate entry for hardware_name : calvin
Deleting previous duplicate entry for hardware_name : hobbes
Deleting previous duplicate entry for hardware_name : dora
Success!! Added 10 new configurations to the collection : machine_collection
```

aggiestack show flavors



```
$ aggiestack show flavors
These are the flavors configured on the server
flavor_name  :  small
RAM  :  1
numDisks  :  1
numVcpus  :  1


flavor_name  :  medium
RAM  :  8
numDisks  :  2
numVcpus  :  4


flavor_name  :  large
RAM  :  16
numDisks  :  2
numVcpus  :  4


flavor_name  :  xlarge
RAM  :  32
numDisks  :  4
numVcpus  :  8
```

aggiestack admin show hardware



```
$ aggiestack admin show hardware
Current configuration of the machines
hardware_name  :  m1
Current RAM  :  8
Current numDisks  :  8
Current numVcpus  :  2
ip  :  128.0.0.1
Original RAM  :  8
Original numDisks  :  8
Original numVcpus  :  2


hardware_name  :  m2
Current RAM  :  16
Current numDisks  :  32
Current numVcpus  :  4
ip  :  128.0.0.2
Original RAM  :  16
Original numDisks  :  32
Original numVcpus  :  4


hardware_name  :  m3
Current RAM  :  16
Current numDisks  :  16
Current numVcpus  :  4
ip  :  128.0.0.3
Original RAM  :  16
Original numDisks  :  16
Original numVcpus  :  4


hardware_name  :  m4
Current RAM  :  16
Current numDisks  :  8
Current numVcpus  :  4
ip  :  128.0.0.4
Original RAM  :  16
Original numDisks  :  8
Original numVcpus  :  4
```

Method 2: Running locally

In case you do not want to install our CLI, you can also locally run the application and test the commands.

1. Download and unzip the application
2. Open command prompt or git bash
3. Navigate to path: \678-18-c\P0\aggiestack
4. Run the command

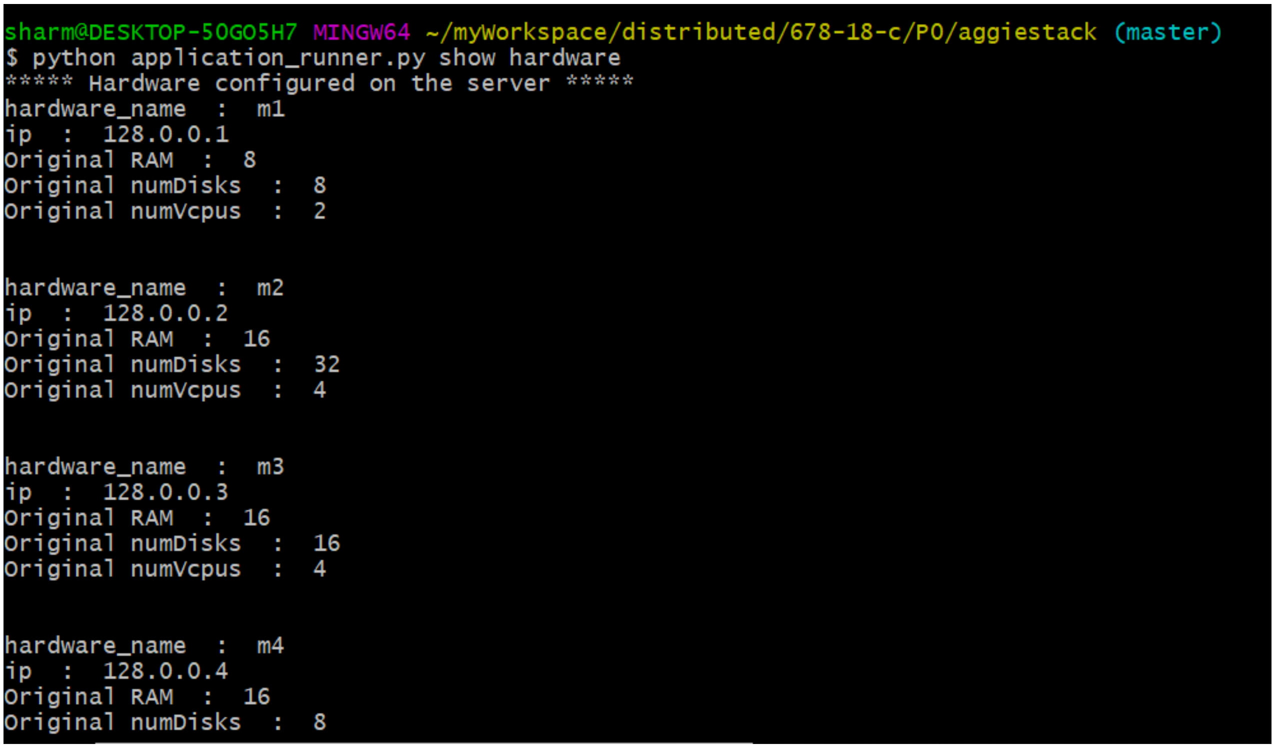
   python application_runner.py <aggiestack cli command without keyword ‘aggiestack’>

   For example:

   python application_runner.py config --hardware hardware-config

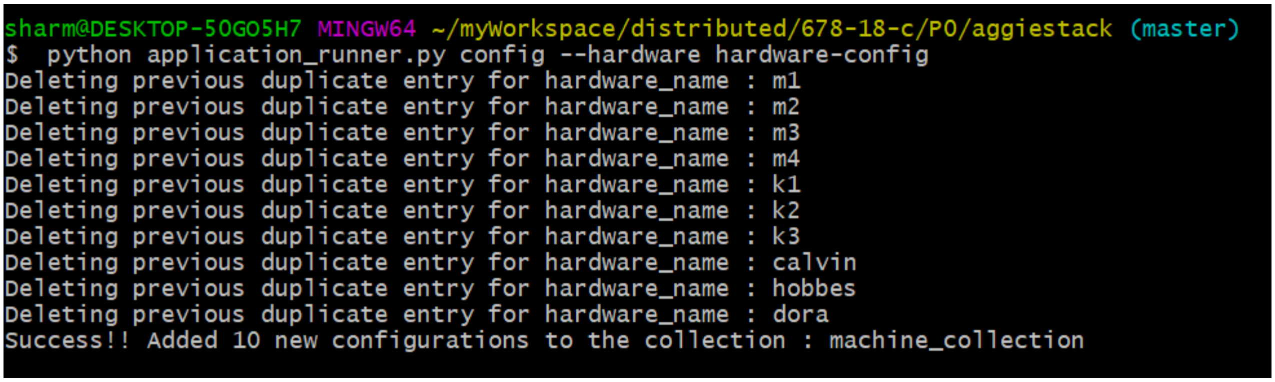
   python application_runner.py show hardware

   python application_runner.py admin show hardware

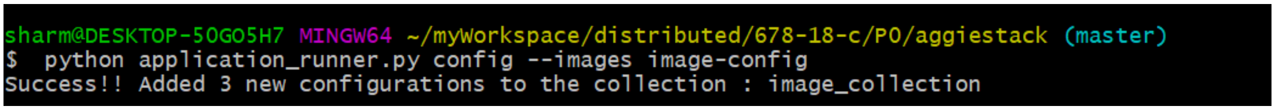


```
sharm@DESKTOP-50GO5H7 MINGW64 ~/myWorkspace/distributed/678-18-c/P0/aggiestack (master)
$ python application_runner.py show hardware
***** Hardware configured on the server *****
hardware_name  :  m1
ip  :  128.0.0.1
Original RAM  :  8
Original numDisks  :  8
Original numVcpus  :  2


hardware_name  :  m2
ip  :  128.0.0.2
Original RAM  :  16
Original numDisks  :  32
Original numVcpus  :  4


hardware_name  :  m3
ip  :  128.0.0.3
Original RAM  :  16
Original numDisks  :  16
Original numVcpus  :  4


hardware_name  :  m4
ip  :  128.0.0.4
Original RAM  :  16
Original numDisks  :  8
```

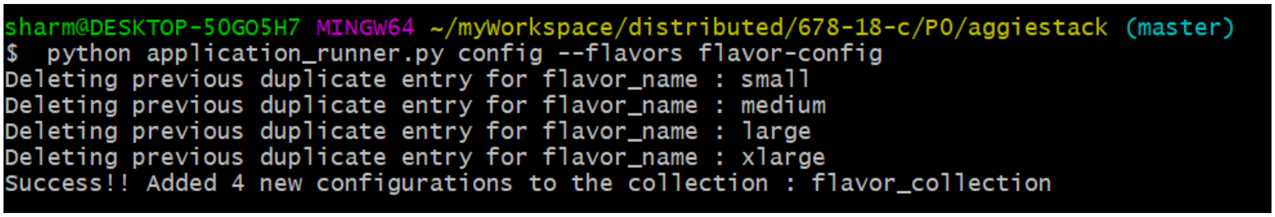
Screenshots for other commands have been attached:



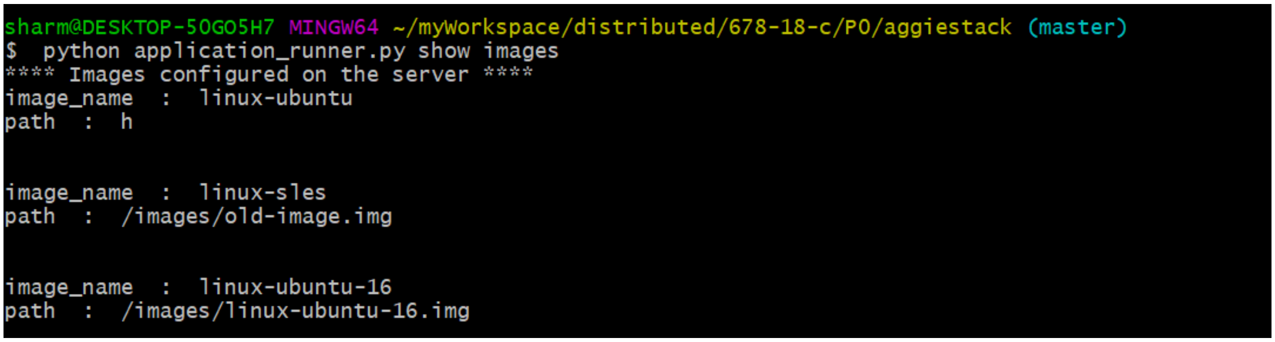
```
sharm@DESKTOP-50GO5H7 MINGW64 ~/myWorkspace/distributed/678-18-c/P0/aggiestack (master)
$  python application_runner.py config --hardware hardware-config
Deleting previous duplicate entry for hardware_name : m1
Deleting previous duplicate entry for hardware_name : m2
Deleting previous duplicate entry for hardware_name : m3
Deleting previous duplicate entry for hardware_name : m4
Deleting previous duplicate entry for hardware_name : k1
Deleting previous duplicate entry for hardware_name : k2
Deleting previous duplicate entry for hardware_name : k3
Deleting previous duplicate entry for hardware_name : calvin
Deleting previous duplicate entry for hardware_name : hobbes
Deleting previous duplicate entry for hardware_name : dora
Success!! Added 10 new configurations to the collection : machine_collection
```



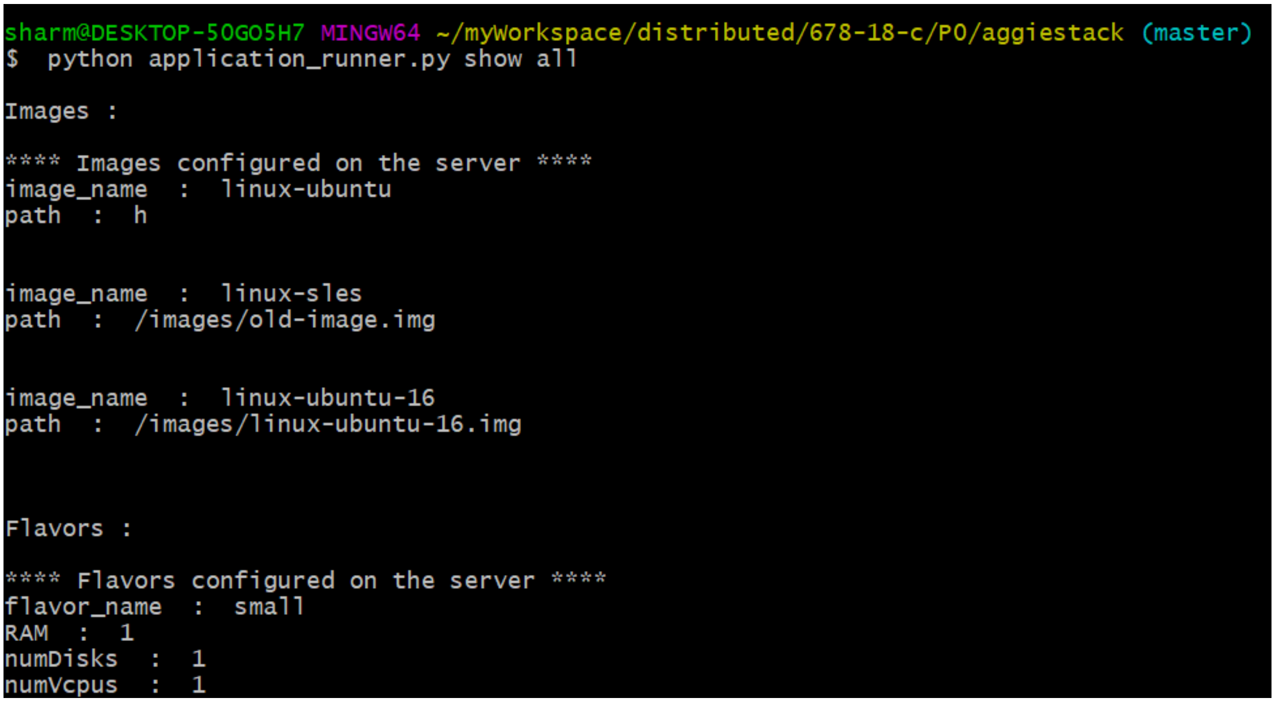
```
sharm@DESKTOP-50GO5H7 MINGW64 ~/myWorkspace/distributed/678-18-c/P0/aggiestack (master)
$  python application_runner.py config --images image-config
Success!! Added 3 new configurations to the collection : image_collection
```



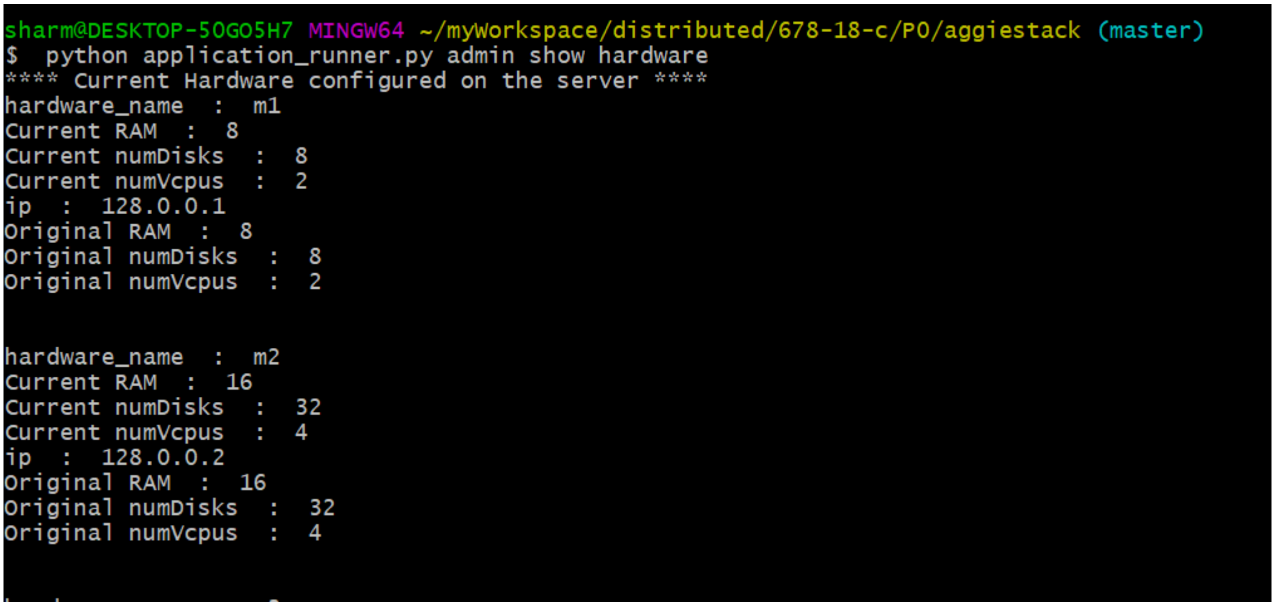
```
sharm@DESKTOP-50GO5H7 MINGW64 ~/myWorkspace/distributed/678-18-c/P0/aggiestack (master)
$  python application_runner.py config --flavors flavor-config
Deleting previous duplicate entry for flavor_name : small
Deleting previous duplicate entry for flavor_name : medium
Deleting previous duplicate entry for flavor_name : large
Deleting previous duplicate entry for flavor_name : xlarge
Success!! Added 4 new configurations to the collection : flavor_collection
```



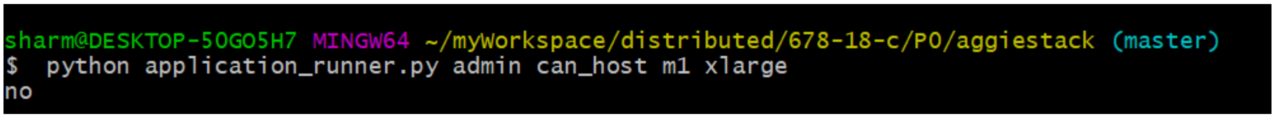
```
sharm@DESKTOP-50GO5H7 MINGW64 ~/myWorkspace/distributed/678-18-c/P0/aggiestack (master)
$  python application_runner.py show images
**** Images configured on the server ****
image_name   :  linux-ubuntu
path   :  h


image_name   :  linux-sles
path   :  /images/old-image.img


image_name   :  linux-ubuntu-16
path   :  /images/linux-ubuntu-16.img
```



```
sharm@DESKTOP-50G05H7 MINGW64 ~/myWorkspace/distributed/678-18-c/P0/aggiestack (master)
$  python application_runner.py show all

Images :

**** Images configured on the server ****
image_name  :  linux-ubuntu
path  :  h


image_name  :  linux-sles
path  :  /images/old-image.img


image_name  :  linux-ubuntu-16
path  :  /images/linux-ubuntu-16.img



Flavors :

**** Flavors configured on the server ****
flavor_name  :  small
RAM  :  1
numDisks  :  1
numVcpus  :  1
```



```
sharm@DESKTOP-50G05H7 MINGW64 ~/myWorkspace/distributed/678-18-c/P0/aggiestack (master)
$  python application_runner.py admin show hardware
**** Current Hardware configured on the server ****
hardware_name  :  m1
Current RAM  :  8
Current numDisks  :  8
Current numVcpus  :  2
ip  :  128.0.0.1
Original RAM  :  8
Original numDisks  :  8
Original numVcpus  :  2


hardware_name  :  m2
Current RAM  :  16
Current numDisks  :  32
Current numVcpus  :  4
ip  :  128.0.0.2
Original RAM  :  16
Original numDisks  :  32
Original numVcpus  :  4
```



```
sharm@DESKTOP-50G05H7 MINGW64 ~/myWorkspace/distributed/678-18-c/P0/aggiestack (master)
$  python application_runner.py admin can_host m1 xlarge
no
```

References:

https://stormpath.com/blog/building-simple-cli-interfaces-in-python

https://github.com/rdegges/skele-cli/blob/master/skele/cli.py

https://www.pythonforbeginners.com/basics/python-docstrings

https://medium.com/@trstringer/the-easy-and-nice-way-to-do-cli-apps-in-python-5d9964dc950d

https://radek.io/2015/01/19/docopt/

https://pypi.org/project/aggiestack/#files

https://www.youtube.com/watch?v=SdCG1LrbMbg&t=222s