

## Project 4 Part C Report

### Logistics

*Estimated implementation time:* 50-60 minutes (having already performed the design work)

*Estimated testing time:* 30-45 minutes (to thoroughly test and debug)

### General Design Modifications

Rack objects were already implemented for parts A and B. To implement the new functionality:

- Add two new parameters to rack objects:
  - `store_max` - The maximum capacity of the rack's local storage.
  - `store_avail` - The rack's remaining local storage.
- Add a data structure to rack objects:
  - `images` - A dictionary of those images copied to the rack's local storage.

Remove the Part B functions not included in the Part C API.

### Command / Function Design Modifications

`aggiestack config --hardware hdwr-config.txt`

- Add parsing code to interpret rack objects and their local storage capacities (`store_max`) and add them to a dictionary of rack objects (this was already implemented for Parts A & B). Set `store_avail` to `store_max` for each rack.

`aggiestack config --images image-config.txt`

- No change here. The program already parses image objects' space values.

`aggiestack config --flavors flavor-config.txt`

- No change here.

`aggiestack show hardware`

- No change here. The program already prints rack information in-line with the hardware output.

`aggiestack show images`

- After printing image object information as before, iterate across rack objects and their `images` collections and print the image names associated with each rack.

`aggiestack server create --image IMAGE -- flavor FLAVOR_NAME INSTANCE_NAME`

- In the step of the `server_create()` function where a host machine is sought after, iterate across rack objects as before. Check if the desired image is contained in each rack's `images` dictionary. If it is, try to add the instance to a machine on that rack as before (continue if this fails). If not, insert the rack object into a temporary array of rack objects insertion-sorted by their available storage parameter, `store_avail`.
- If no host machine is found, iterate across the temporary sorted array of rack objects from highest `store_avail` to lowest `store_avail`, and try to add the instance to a machine on each rack in turn. If successful, add the image to the rack object's `images` collection and decrease its `store_avail` value accordingly. If doing this would exceed the rack's local storage capacity (`store_avail < 0`), remove other images from the rack's local collection and increase `store_avail` accordingly until there is sufficient space.
- Return and break all loops once the instance is hosted and storage values have been updated.

`aggiestack server delete INSTANCE_NAME`

- As before.

`aggiestack server list`

- As before.

`aggiestack admin show imagecaches RACK_NAME`

- Check if the desired rack exists and then print (1) its `store_avail` value and (2) its `images` collection.

`aggiestack admin show hardware`

- As before.

`aggiestack admin show instances`

- As before.