**Cloud Computing Homework 4**

**Annette Chiu**

Improve "to-do list" application's design by using web service. To be more specific, leave the application front-end and overall structure unchanged, but provide the data service through a typical RESTful API instead of supporting directly by a database as in the solution posted for the previous homework.

For example, the original code for root URL

@app.route("/")

def show\_list():

db = get\_db()

cur = db.execute('SELECT what\_to\_do, due\_date, status FROM entries')

entries = cur.fetchall()

tdlist = [dict(what\_to\_do=row[0], due\_date=row[1], status=row[2])

for row in entries]

return render\_template('index.html', todolist=tdlist)

will become

@app.route("/")

def show\_list():

with urllib.request.urlopen('http://localhost:6000/api/items') as response:

resp = response.read()

resp = json.loads(resp)

return render\_template('index.html', todolist=resp)

, assuming the servce runs on port 6000.

Of course, you also need to implement the service itself, which will be similar in structure to the original app, except returning JSON instead of HTML. The corresponding function would be like this

@app.route("/api/items")

def get\_items():

db = get\_db()

cur = db.execute('SELECT what\_to\_do, due\_date, status FROM entries')

entries = cur.fetchall()

tdlist = [dict(what\_to\_do=row[0], due\_date=row[1], status=row[2])

for row in entries]

return jsonify(tdlist)

1. Using Docker container technology and kubernetes, deploy the 'to-do list' app to a cluster on Google Cloud Platform. You need to describe how you create the Docker image, e.g. using a *Dockerfile*, and how you create the cluster and deploy the containers to it, e.g. using a script (bash script on Linux/Mac, or batch file on Windows) that includes all the relevant commands for this procedure.

**Hint:** To avoid too much complexity, only deploy the web app part using kubernetes, and run API service on a single VM. Use the external IP of API in the app's code

**Question 2**

Create and start a container

$ docker create -t -i fedora bash

6d8af538ec541dd581ebc2a24153a28329acb5268abe5ef868c1f1a261221752

$ docker start -a -i 6d8af538ec5

bash-4.2#

Initialize volumes

As of v1.4.0 container volumes are initialized during the docker create phase (i.e., docker run too). For example, this allows you to create the data volume container, and then use it from another container:

$ docker create -v /data --name data ubuntu

240633dfbb98128fa77473d3d9018f6123b99c454b3251427ae190a7d951ad57

$ docker run --rm --volumes-from data ubuntu ls -la /data

total 8

drwxr-xr-x 2 root root 4096 Dec 5 04:10 .

drwxr-xr-x 48 root root 4096 Dec 5 04:11 ..

Similarly, create a host directory bind mounted volume container, which can then be used from the subsequent container:

$ docker create -v /home/docker:/docker --name docker ubuntu

9aa88c08f319cd1e4515c3c46b0de7cc9aa75e878357b1e96f91e2c773029f03

$ docker run --rm --volumes-from docker ubuntu ls -la /docker

total 20

drwxr-sr-x 5 1000 staff 180 Dec 5 04:00 .

drwxr-xr-x 48 root root 4096 Dec 5 04:13 ..

-rw-rw-r-- 1 1000 staff 3833 Dec 5 04:01 .ash\_history

-rw-r--r-- 1 1000 staff 446 Nov 28 11:51 .ashrc

-rw-r--r-- 1 1000 staff 25 Dec 5 04:00 .gitconfig

drwxr-sr-x 3 1000 staff 60 Dec 1 03:28 .local

-rw-r--r-- 1 1000 staff 920 Nov 28 11:51 .profile

drwx--S--- 2 1000 staff 460 Dec 5 00:51 .ssh

drwxr-xr-x 32 1000 staff 1140 Dec 5 04:01 docker

Set storage driver options per container.

$ docker create -it --storage-opt size=120G fedora /bin/bash

This (size) will allow to set the container rootfs size to 120G at creation time. This option is only available for the devicemapper, btrfs, overlay2, windowsfilter and zfsgraph drivers. For the devicemapper, btrfs, windowsfilter and zfs graph drivers, user cannot pass a size less than the Default BaseFS Size. For the overlay2 storage driver, the size option is only available if the backing fs is xfs and mounted with the pquotamount option. Under these conditions, user can pass any size less than the backing fs size.