

## Dev-Ops Home Challenge

### Task 1.a :

Given a Linux OS and bash shell.

Using oneliner bash command, create a histogram of relative volume per hour with any 24 span from:

[https://poloniex.com/public?ommand=returnTradeHistory&currencyPair=BTC\\_ETH&start=1580515200&end=1583020800](https://poloniex.com/public?ommand=returnTradeHistory&currencyPair=BTC_ETH&start=1580515200&end=1583020800)

Allowed commands are: for, echo, tr, tail, grep, head, sed, awk, wc, curl, cat, sort, uniq

Example output:

```
00: #####
01: #####
02: #####
...
23: #####
```

### Task 1.b :

Given a Linux OS and bash shell.

Using a oneliner bash command, find the top 3 biggest files in the file system.

### Task 2:

Run two *Nginx* docker containers in active-passive mode by using *Keepalived*.

The containers must be based on an Ubuntu **16:04 image**.

The goal is to ensure one Nginx service is running at all times.

*Nginx* containers should be called *keepalived\_master/keepalived\_backup*.

Both containers should contain

- keepalived installed and configured
- nginx with a simple web page (*hello I'm master/backup!*).
- Master's Nginx webpage is served on port 8880 to the host machine.
- Slave's Nginx webpage is served on port 8881 to the host machine.

To run everything use `docker-compose up`

**Bonus Points:** Place both Nginx containers behind a load-balancer (HAProxy) in order to have one url and port for both containers.

#### Test:

Visit *localhost:8880* and you should see "hello I'm master!"

Visit *localhost:8881* should not be possible!

Running the command: `docker pause keepalived_master`

Visit again *localhost:8880* should not be possible!

Visit again *localhost:8881* and you should see "hello I'm backup!"

### Task 3

Calculate what would be the LCU capacity and the monthly cost for operating an AWS Application Load blancer in Frankfurt region with the following details:

- 26.6 new connections / s
- 27 active connections / s (each connection lasting 50 ms)
- processed 15000000 Bytes per minute

### Task 4

Write Terrafor, Cloudformation or Ansible playbook/roles to launch 3 ec2, t2.small instances with a tag: "env=task", all of the instances should also have security group "ssh" assigned to them where access on port 2222 is allows to all ip's

### Task 5

How would you scale a monolithic API service that is running in the cloud which fails to process API calls fast enough?

### Task 6

Write a python script that processes the files from [here](#) (you can manually extract everything in advance) and count how many times the term “**bitcoin**” was searched in the **english** wikipedia version.

\* the format of the file content is explained [here](#).

### Delivery:

Task 1: command that prints the results, and an explanation of every section.

Task 2: Instructions on how to run and test. Dockerfile, docker-compose, keepalived, nginx, Haproxy configuration files and any other relevant files.

Task 3: Written solution that also includes the calculation.

Task 4: Script file

Task 5: Written solution

Task 6: Python script file

**GOOD LUCK!**