0x0F. Load balancer

DevOpsSysAdmin

- By: Sylvain Kalache, co-founder at Holberton School
- Weight: 1
- Project over - took place from Sep 26, 2022 6:00 AM to Sep 27, 2022 6:00 AM
- An auto review will be launched at the deadline

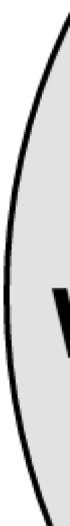
In a nutshell…

- **Auto QA review:** 2.0/4 mandatory & 2.0/2 optional
- **Altogether: 100.0%**
  - Mandatory: 50.0%
  - Optional: 100.0%
  - Calculation: 50.0% + (50.0% * 100.0%) == **100.0%**

Concepts

For this project, we expect you to look at these concepts:

- Load balancer
- Web stack debugging



Background Context

You have been given 2 additional servers:

- gc-[STUDENT_ID]-web-02-XXXXXXXXXX
- gc-[STUDENT_ID]-lb-01-XXXXXXXXXX

Let's improve our web stack so that there is redundancy for our web servers. This will allow us to be able to accept more traffic by doubling the number of web servers, and to make our infrastructure more reliable. If one web server fails, we will still have a second one to handle requests.

For this project, you will need to write Bash scripts to automate your work. All scripts must be designed to configure a brand new Ubuntu server to match the task requirements.

Resources

Read or watch:

- Introduction to load-balancing and HAproxy
- HTTP header
- Debian/Ubuntu HAProxy packages

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted on Ubuntu 16.04 LTS
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project, is mandatory
- All your Bash script files must be executable
- Your Bash script must pass `Shellcheck` (version `0.3.7`) without any error
- The first line of all your Bash scripts should be exactly `#!/usr/bin/env bash`
- The second line of all your Bash scripts should be a comment explaining what is the script doing

Your servers

Name	Username	IP	State	
1609-web-01				Actions Toggle Dropdown
1609-web-02				Actions Toggle Dropdown
1609-lb-01				Actions Toggle Dropdown

Tasks

0. Double the number of webservers

mandatory

Score: 33.33% (*Checks completed: 33.33%*)

In this first task you need to configure `web-02` to be identical to `web-01`. Fortunately, you built a Bash script during your web server project, and they'll now come in handy to easily configure `web-02`. Remember, always try to automate your work!

Since we're placing our web servers behind a load balancer for this project, we want to add a custom Nginx response header. The goal here is to be able to track which web server is answering our HTTP requests, to understand and track the way a load balancer works. More in the coming tasks.

Requirements:

- Configure Nginx so that its HTTP response contains a custom header (on `web-01` and `web-02`)
  - The name of the custom HTTP header must be `X-Served-By`
  - The value of the custom HTTP header must be the hostname of the server Nginx is running on
- Write `0-custom_http_response_header` so that it configures a brand new Ubuntu machine to the requirements asked in this task
  - Ignore SC2154 for `shellcheck`

Example:

```
sylvain@ubuntu$ curl -sI 34.198.248.145 | grep X-Served-By

X-Served-By: 03-web-01

sylvain@ubuntu$ curl -sI 54.89.38.100 | grep X-Served-By

X-Served-By: 03-web-02
```

```
sylvain@ubuntu$
```

If your server’s hostnames are not properly configured (`[STUDENT_ID]-web-01` and `[STUDENT_ID]-web-02`.), follow this tutorial.

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x0F-load_balancer`
- File: `0-custom_http_response_header`

Done? Help Check your code Ask for a new correction Get a sandbox QA Review

1. Install your load balancer

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Install and configure HAproxy on your `lb-01` server.

Requirements:

- Configure HAproxy so that it send traffic to `web-01` and `web-02`
- Distribute requests using a roundrobin algorithm
- Make sure that HAproxy can be managed via an init script
- Make sure that your servers are configured with the right hostnames: `[STUDENT_ID]-web-01` and `[STUDENT_ID]-web-02`. If not, follow this tutorial.
- For your answer file, write a Bash script that configures a new Ubuntu machine to respect above requirements

Example:

```
sylvain@ubuntu$ curl -Is 54.210.47.110
HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Mon, 27 Feb 2017 06:12:17 GMT
Content-Type: text/html
Content-Length: 30
Last-Modified: Tue, 21 Feb 2017 07:21:32 GMT
Connection: keep-alive
ETag: "58abea7c-1e"
X-Served-By: 03-web-01
Accept-Ranges: bytes
```

```
sylvain@ubuntu$ curl -Is 54.210.47.110
HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Mon, 27 Feb 2017 06:12:19 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Mar 2014 11:46:45 GMT
Connection: keep-alive
ETag: "5315bd25-264"
X-Served-By: 03-web-02
Accept-Ranges: bytes

sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x0F-load_balancer`
- File: `1-install_load_balancer`

Done! Help Check your code Get a sandbox QA Review

2. Add a custom HTTP header with Puppet

#advanced

Score: 100.0% (*Checks completed: 100.0%*)

Just as in task #0, we'd like you to automate the task of creating a custom HTTP header response, but with Puppet.

- The name of the custom HTTP header must be `X-Served-By`
- The value of the custom HTTP header must be the hostname of the server Nginx is running on
- Write `2-puppet_custom_http_response_header.pp` so that it configures a brand new Ubuntu machine to the requirements asked in this task

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x0F-load_balancer`
- File: `2-puppet_custom_http_response_header.pp`

Done! Help Check your code Get a sandbox QA Review