Basics of Cloud Computing 2018/19 spring

Cus

- Main (https://courses.cs.ut.ee/2019/cloud/spring/Main/HomePage)
- Lectures (https://courses.cs.ut.ee/2019/cloud/spring/Main/Lectures)
- Practicals (https://courses.cs.ut.ee/2019/cloud/spring/Main/Practicals)
- Results (https://courses.cs.ut.ee/2019/cloud/spring/Main/Results)
- Submit Homework (https://courses.cs.ut.ee/2019/cloud/spring/Main/Homework)

Practice 3 - Load balancing

In this lab we will take a look at load balancing web applications running ontop of IaaS cloud instances. You will set up two instances. One with a simple PHP application and second one with a load balancer. You will define a load balancing group which will contain your own application server, together with other student servers.

We will use Apache web server with PHP for the application and nginx for the load balancer.

Exercise 3.1. Setting up an Application Server instance

We will create new instance for a Application server that use Apache 2 web server and PHP. We will also download a simple PHP page that keeps track of how many users have visited the page and from which IP's.

1. Start an instance either from your previous snapshot or create a new clean instance from `Ubuntu 18.04` image.
   1. This time you will run multiple machines at the same time so for flavour please use `m1.xsmall`
   2. For network you can use `provider_64_net`
   3. Make sure you use a `security group` that has `port 80 open`, like the security group you made last practice session!!
2. Log into the instance through SSH
3. Install the required software packages:
   1. Refresh the list of software packages: `sudo apt update`
   2. Install the following ubuntu packages: `apache2 php libapache2-mod-php`
4. Restart the apache web server:
   1. `sudo service apache2 restart`
5. Delete the default Apache HTTP web page file located at: `/var/www/html/index.html`
6. Download index.zip (http://math.ut.ee/~jakovits/index.zip)
   - Unpack it and upload the containing `index.php` file to the instance using `scp` (or just copy its content)
   - Alternatively you can use `wget` to download the zip directly to the instance and `unzip` to unpack it.
     - `wget http://math.ut.ee/~jakovits/index.zip`

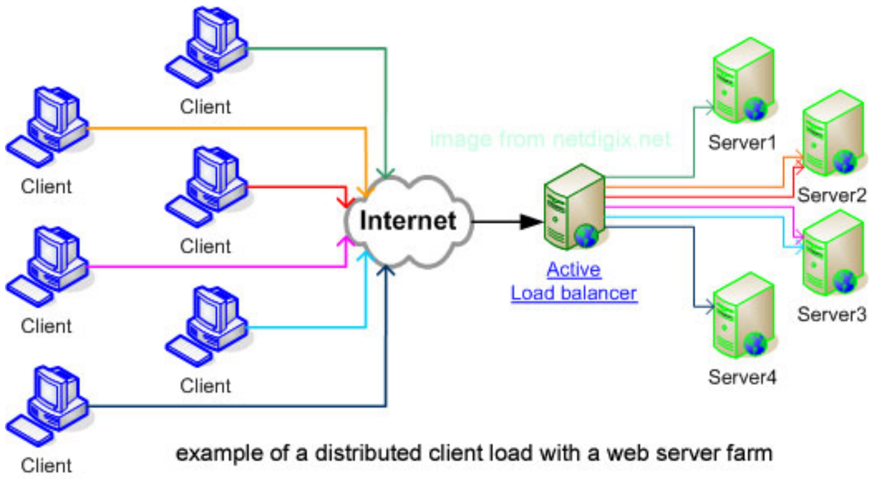
(http://math.ut.ee/~jakovits/index.zip)

7. Copy the `index.php` file into the Apache HTTP application folder `/var/www/html/index.php` (You will have to use sudo as your user does have required permissions in that folder)
8. Create an empty data file: `/var/www/html/data.txt`
    1. This file is used to store the list and count of incoming user requests.
    2. Change the owner of the created file to 'www-data': `sudo chown www-data /var/www/html/data.txt` (Apache web server is running under this user)
9. Access your instance through a web browser using its Public IP to check that the website is working as required
10. Modify `/var/www/html/index.php` to make the web page more personal to you, so that others would recognize that this application server was set up by you when they visit the page.
    1. How you decide to modify it up to you, but there should at least be your Full Name present.
    2. You may want to make it visually very recognizable to differentiate between your server and other students servers more easily.
    3. `nano` is a suitable command line file editor to modify text files.
        - You can use
11. Take a screenshot of your application server web page

Exercise 3.2. Setting up load balancer instance and balancing multiple application servers

In this exercise, we set up another instance and install a NginX load balancer on it.

Load balancer distributes all user requests of the web page across multiple application servers. Load balancers are used to increase capacity (concurrent users) and reliability of applications and are a required component for auto-scaling - meaning changing the number of application servers dynamically based on number of active users.



(image taken from http://www.netdigix.com/linux-loadbalancing.php (http://www.netdigix.com/linux-loadbalancing.php))

- We are using Nginx (http://nginx.org/en/ (http://nginx.org/en/)) as a load balancer.
- Create a new instance (Should also be Ubuntu 18.04) for the load balancer
    - This instance should also have the port 80 open, just like with the first instance
- Install Nginx on the instance
    - Refresh the list of software packages: `sudo apt update`
    - Install the following ubuntu package: `nginx-full`
- Modify Nginx configuration to add your application server IP into the list of managed services

- Download the example nginx load balancer configuration: `wget http://kodu.ut.ee/~jakovits/default (http://kodu.ut.ee/~jakovits/default)`
- Modify the downloaded `default` configuration file
  - Find the **upstream lab_load_balancer** block
  - Modify the `server 172.17.64.144;` line in the configuration file and replace the existing IP (172.17.64.144) with the actual IP of your application server.

- Overwrite the default nginx site configuration on the instance with the modified configuration
  - `sudo cp default /etc/nginx/sites-enabled/default`
- Reload Nginx service
  - `sudo service nginx reload`
- Visit the IP address of the **load balancer** using a web browser and verify that it displays your application server
- Now also add lab supervisor and few other student's application servers under your load balancer to distribute user requests between multiple servers.
  - Create a new `server IP_ADDRESS;` line inside the **upstream lab_load_balancer** block for each additional server you wish to add
- Take a screenshot of visiting your application server web page **through the load balancer** (Browser is accessing load balancer IP address, but your application server content is shown)

Exercise 3.3. Verifying that the load balancing is working properly

Lets test whether the previous steps have been completed successfully by verifying that the load balancing is working and user traffic is being rerouted into your application server.

- Visit the load balancer multiple times. Do you recognize your own application sever?
- Ask other students to visit your load balancer or to add your application server into their load balancer
- Wait until you start seeing requests on your application server tracker from a number of other locations.
- You can check current incoming HTTP connections using the following command: `netstat | grep http`
  - Take a screenshot of the output of **netstat** command. Try to have it display more than 4 connections when other student`s requests are being redirected to your server.

Exercise 3.4. Generating additional user traffic for benchmarking

Lets check whether the load balancer can now handle a higher number of simultaneous user requests.

- Your task is to generate a large number of user requests to the load balancer and verify how many of those requests are sent to your application server by the load balancer.
- To generate a large number of user request, you can either use existing load generation tools, web applets or write a simple script (Python, Java, Bash, JavaScript, etc.) to visit the load balancer page multiple times in a sequence.
- Generate at least 1000 requests.
- Note down how many of those requests end up on your application server. How large percent of your generated user requests ended up visiting your server?
  - Take a screenshot of the tool or a script that you used for load generation.

Deliverables

- This time you should leave your instance running unless otherwise instructed so by the lab assistant

- Screenshot of your application server web page. Browser should be directed at the application server instance IP address, without going through the load balancer. (Exercise 3.1)
- Screenshot of visiting your application server web page through the load balancer. Browser should be directed at the load balancer instance IP address. (Exercise 3.2)
- Screenshot taken of the netstat command in exercise 3.3.
- Screenshot taken in exercise 3.4 of the load generation tool.

Task	lab 3
	If your homework consists of multiple files (for example HTML + CSS + JS) it should be archived before submitting.
File	Choose File no file selected
Comment	
	Submit

Incase of issues, check the following:

1. You might encounter an error stating that something is locked that is Ubuntu running some updates in background so please give it few minutes to complete and try again later, if still no luck ask help from lab instructor. (Use with caution! https://www.tecmint.com/fix-unable-to-lock-the-administration-directory-var-lib-dpkg-lock/ (https://www.tecmint.com/fix-unable-to-lock-the-administration-directory-var-lib-dpkg-lock/))

Institute of Computer Science (http://cs.ut.ee/)
| Faculty of Science and Technology (http://reaalteadused.ut.ee/)
| University of Tartu (http://www.ut.ee/)

In case of technical problems or questions write to: ati.error@ut.ee (mailto:ati.error@ut.ee)

The courses of the Institute of Computer Science are supported by following programs:

European Union Regional Development Fund
Investing in your future

European Union European Social Fund
Investing in your future