|  |
| --- |
| Brylee Curran |
| Frank Li |
| Stella Xu |
| Saiful Sufri |

Assignment 2 – Task 2

158.383 Information Technology Project

2018

# Task 2

## Concept description

Question 2 is about upgrading a system in a safer manner and avoiding downtime in doing so. We divided this process into four steps:

* Step 1: Use a load balancer (such as Nginx) to manage user’s requests.
* Step 2: Start the new system (and synchronize the data from old system if we need).
* Step 3: Make the load balancer point to the new system and remove the old system from load balancer.
* Step 4: Shutdown the old system.

DevOps is not only just about software deployment; it is also a set of processes and methods for communication and collaboration between departments in a company. Sometimes a company has requirements on "deployment 10 times a day" (“Devops”, n.d.), so it is very important to upgrade the system in a safer manner and in a way that avoids downtime, which will save a lot of time and provide better user experience. The development team and the operational team can have a more collaborative and efficient relationship between each other. As a result of improved teamwork, the efficiency of the organization as a whole is improved and the risk of the production environment, which is accompanied by frequent changes, is reduced.

Docker is an open source application container engine that allows developers to package their applications and rely on packages into a portable container that can then be distributed to any popular Linux machine and can also be virtualized. The container is completely sandbox and has no interfaces with each other.

In this assignment, we use one of it, called docker image. This has been challenging for our group. Additionally, we had to think about how the load is balanced. In a file called “wp” is how to implement it. Nginx can not only be used as a powerful web server but also as a reverse proxy server, and Nginx can also according to the operation rules for dynamic and static page separation, can according to the poll, IP hash, URL hash, weight and so on a variety of ways to the back-end server load balancing, at the same time also supports the backend server health check. Although there are many ways to implement load balancing, we have used proxy\_pass, upstream to implement it. Specifies the polling probability that weight is proportional to the access rate for cases where back-end server performance is uneven. To address some of the problems with ip\_hash, you can use the third-party module upstream\_hash, which is mostly used as url\_hash, but does not prevent it from being used for session sharing.

Reference:

*Devops.* (n.d.) Retrieved from https://en.wikipedia.org/wiki/DevOps.

## Upgrading the system in a safe manner without downtime instructions

**1. Launch Instance:**

* Select t2.micro Ubuntu server 16.04.
* Use all default settings.
* Create a key pair and name it anything you want (I have called it “a2”).
* Create a folder on your computer (e.g. I have created a folder on my desktop and called it “383a2”).
* Download the key pair and then place the key pair into your newly created folder.
* Launch the instance.

**2. Connect to your Instance:**

* Open the terminal and cd into the folder you created earlier e.g.

../Desktop/屏幕快照%202018-10-05%20上午10.36.42.png

* Open SSH client and ensure key is not publicly viewable by changing the permissions of your key pair e.g. enter the command into the terminal chmod 400 a2.pem.
* Download our zip file called “final.zip” and place this file into the folder you created earlier (i.e. the 383a2 folder). Ensure that this file is still zipped once downloaded.



* Connect to your instance via the terminal e.g.



**3. Upgrade the system:**

* Copy and paste into the terminal the following red lines:

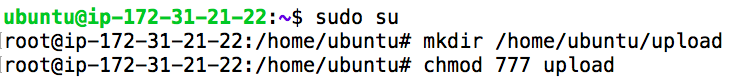
sudo su

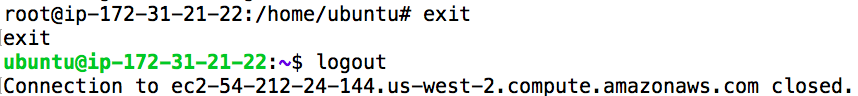
mkdir /home/ubuntu/upload

chmod 777 upload

exit

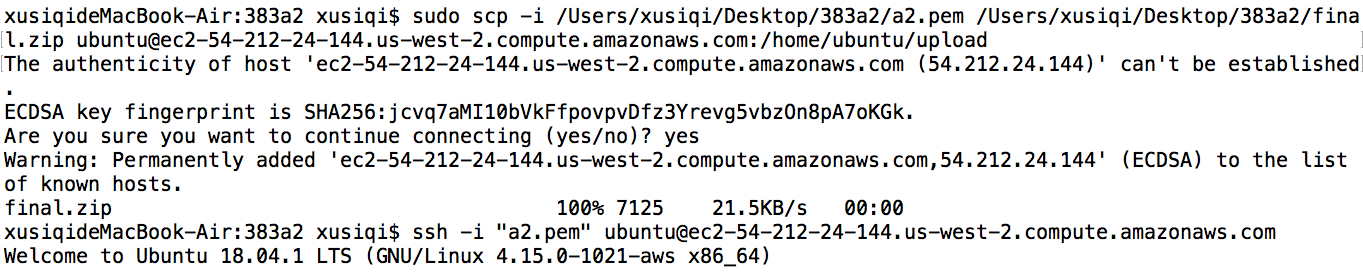
logout

The output should look like the following:

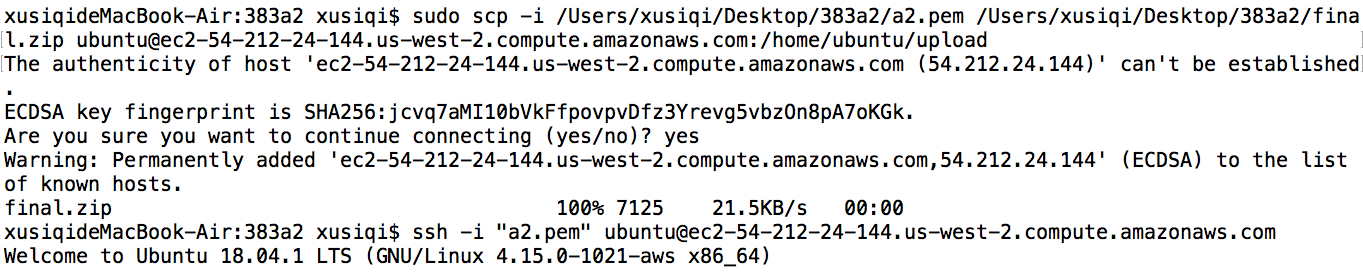


* Enter your own details into the following command and then copy and paste it into the terminal:

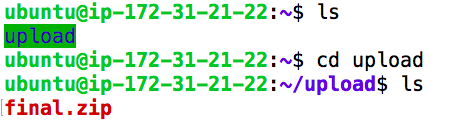
sudo scp –i path\_to\_keypair path\_to\_zipfile hostname@public\_ip:/home/hostname/upload

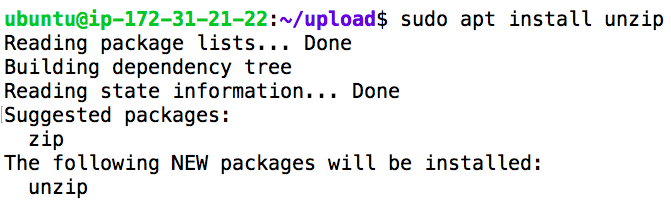
e.g.

The output should look like the following:

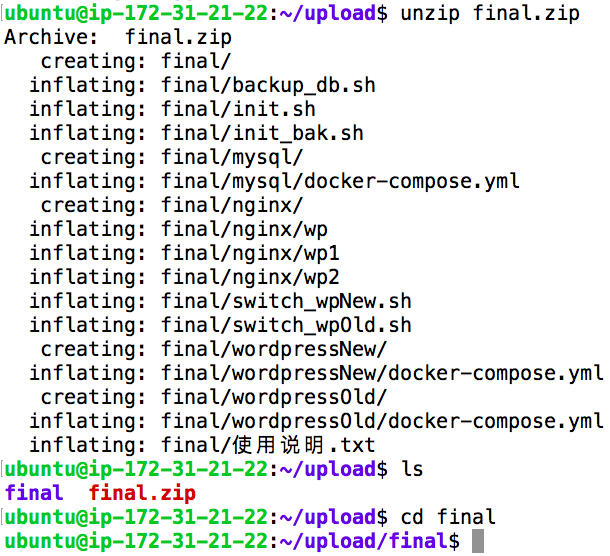


* Now the zip file has been put into the ec2 server and under the upload
* Connect to the instance again (e.g. ssh -i "key\_pair.pem" ubuntu@ec2-54-200-246-70.us-west-2.compute.amazonaws.com)
* By using the ls command in the terminal, you can see the zip file. Firstly cd into “upload” by using the cd upload command, and then ls command.



* Download “unzip” command by using: sudo apt install unzip
* Then unzip the zipped file by using: unzip final.zip command and then cd to “final”

The output should be:

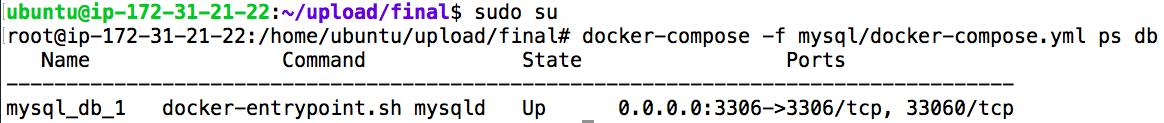


* Copy and paste the following into the terminal:

sudo bash init.sh

sudo su

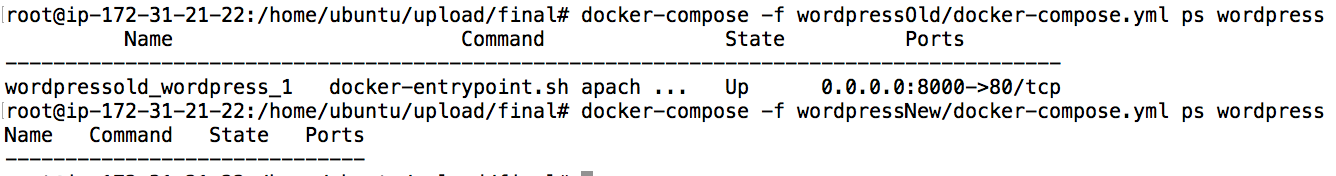
docker-compose -f mysql/docker-compose.yml ps db



* Then copy and paste the following, which can show which server is running and prove our success:

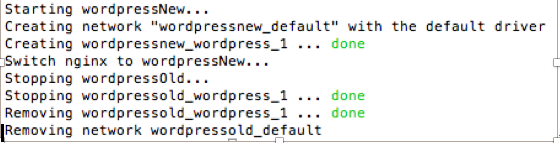
docker-compose -f wordpressOld/docker-compose.yml ps wordpress

docker-compose -f wordpressNew/docker-compose.yml ps wordpress



su ubuntu

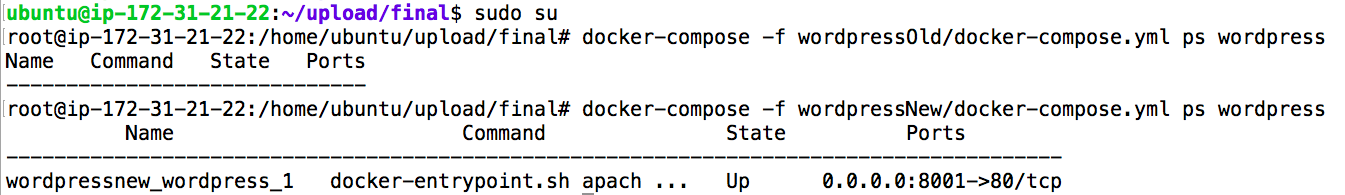
sudo bash switch\_wpNew.sh



sudo su

docker-compose -f wordpressOld/docker-compose.yml ps wordpress

docker-compose -f wordpressNew/docker-compose.yml ps wordpress



**-----You can see that the old server has been shut down and the port is switched from 8000 to 8001. You have completed the upgrade successfully.**