作业要求：在GCE 虚拟机上，部署 Nginx，然后跑起一段 html 來

作业结果：浏览器浏览 http://34.67.225.198/，即可看到所制作的 html 页面。

一. 下载、安装 Nginx

1. 浏览 Nginx 官网的说明文档：http://nginx.org/en/docs/

Installation on Linux

For Linux, nginx packages from nginx.org can be used.

2. 在 GCE 虚拟机(Ubuntu)上，参照安装步骤说明进行 nginx package 的安装：

http://nginx.org/en/linux_packages.html#Ubuntu：

Ubuntu

Install the prerequisites:

```
sudo apt install curl gnupg2 ca-certificates lsb-release
```

To set up the apt repository for stable nginx packages, run the following command:(设置安装源为 nginx 稳定版。我选择了安装这个稳定版。)

```
echo "deb http://nginx.org/packages/ubuntu `lsb_release -cs` nginx" \
    | sudo tee /etc/apt/sources.list.d/nginx.list
```

If you would like to use mainline nginx packages, run the following command instead:

(设置安装源为 nginx 主线版。我没有选择安装这个主线版。)

```
echo "deb http://nginx.org/packages/mainline/ubuntu `lsb_release -cs` nginx" \
    | sudo tee /etc/apt/sources.list.d/nginx.list
```

Next, import an official nginx signing key so apt could verify the packages authenticity:

```
curl -fsSL https://nginx.org/keys/nginx_signing.key | sudo apt-key add -
```

Verify that you now have the proper key:

```
sudo apt-key fingerprint ABF5BD827BD9BF62
```

The output should contain the full fingerprint 573B FD6B 3D8F BC64 1079 A6AB

ABF5 BD82 7BD9 BF62 as follows:

(这是上条command之后的输出信息，不是要输入的command.)

```
pub   rsa2048 2011-08-19 [SC] [expires: 2024-06-14]
      573B FD6B 3D8F BC64 1079  A6AB ABF5 BD82 7BD9 BF62
uid   [ unknown] nginx signing key <signing-key@nginx.com>
```

To install nginx, run the following commands:

```
sudo apt update
sudo apt install nginx
```

然后开始以下自动安装过程了...直至安装完毕:

daveyang163@instance-3:~$ sudo apt install nginx

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following package was automatically installed and is no longer required:

grub-pc-bin

Use 'sudo apt autoremove' to remove it.

The following NEW packages will be installed:

nginx

0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.

Need to get 846 kB of archives.

After this operation, 3,031 kB of additional disk space will be used.

Get:1 http://nginx.org/packages/ubuntu xenial/nginx amd64 nginx amd64 1.18.0-1~xenial [846 kB]

Fetched 846 kB in 1s (550 kB/s)

Selecting previously unselected package nginx.

(Reading database ... 101548 files and directories currently installed.)

Preparing to unpack .../nginx_1.18.0-1~xenial_amd64.deb ...

Thanks for using nginx!

Please find the official documentation for nginx here:

* http://nginx.org/en/docs/

Please subscribe to nginx-announce mailing list to get

the most important news about nginx:

* http://nginx.org/en/support.html

Commercial subscriptions for nginx are available on:

* http://nginx.com/products/

Unpacking nginx (1.18.0-1~xenial) ...

Processing triggers for man-db (2.7.5-1) ...

Processing triggers for ureadahead (0.100.0-19.1) ...

Processing triggers for systemd (229-4ubuntu21.27) ...

Setting up nginx (1.18.0-1~xenial) ...

Processing triggers for ureadahead (0.100.0-19.1) ...

Processing triggers for systemd (229-4ubuntu21.27) ...

daveyang163@instance-3:~$

然后需要重启GCE: `sudo reboot now`

Connected, host fingerprint: ssh-rsa 0 B7:AE:3D:9C:0F:8A:8E:C0:D8:06:C7:5C:58:43:F5:5B:92:84:EA:D1:49:9F:20:C1:E8:5E:1A:A0:3A:DD:84:8E

Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1061-gcp x86_64)

* Documentation: https://help.ubuntu.com

* Management: https://landscape.canonical.com

* Support: https://ubuntu.com/advantage

1 package can be updated.

1 update is a security update.

New release '18.04.4 LTS' available.

Run 'do-release-upgrade' to upgrade to it.

Last login: Thu May 7 08:35:31 2020 from 74.125.41.162

查看 nginx 状态: `service nginx status`

如果 nginx 已经启动, 则会看到如下输出： （如果 nginx 没有启动, 可以再次 start nginx: `sudo nginx`）

● nginx.service - nginx - high performance web server

Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)

Active: active (running) since Thu 2020-05-07 08:43:35 UTC; 33s ago

Docs: http://nginx.org/en/docs/

Process: 1520 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf (code=exited, status=0/SUCCESS)

Main PID: 1554 (nginx)

Tasks: 2

Memory: 3.1M

CPU: 10ms

CGroup: /system.slice/nginx.service

├─1554 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.con

└─1555 nginx: worker process

May 07 08:43:35 instance-nginx systemd[1]: Starting nginx - high performance web server...

May 07 08:43:35 instance-nginx systemd[1]: nginx.service: Can't open PID file /var/run/nginx.pid (yet?) after start

May 07 08:43:35 instance-nginx systemd[1]: Started nginx - high performance web server.

这时用浏览器访问 GCE 的外部 IP 地址（http://34.67.225.198/，注意是http, 而不是https），就可以看到 nginx 的 welcom 画面了：

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

二. Copy html example files to Nginx 默认配置的目录，修改 nginx.conf, 然后 reload nginx.

1. 上传 html 所需文件到 GCE user 目录：

```
daveyang163@instance-3:/$ ls
```

14c9d9ee-2d97-4d94-b102-1cde17ac43e5.png

1.css eaba6153-25e3-4769-a6bb-3e6634b3fc2c.png

3c6d1007-e159-43d5-9298-69d81ddc7381.png index.html

ae3d5be3-c871-4977-9de2-fa5d054e5263.png KKMyfEdqnMFZLdtZLktD.png

b34085d2-dbf4-4331-bbb2-e8fd490122c3.png kmzXYyMiWhGHpTSCoHoG.jpg

2. 在 GCE root 目录下创建 nginx 所需 data/www 和 data/images 目录：

```
daveyang163@instance-nginx:~$ sudo mkdir /data
daveyang163@instance-nginx:~$ sudo mkdir /data/www
daveyang163@instance-nginx:~$ sudo mkdir /data/imgages
daveyang163@instance-nginx:~$ sudo mkdir /data/www/css
daveyang163@instance-nginx:~$ sudo mkdir /data/www/js
```

3. copy html 所需文件到 data/www 和 data/images 目录：

```
daveyang163@instance-nginx:~$ sudo cp *.png /data/images
daveyang163@instance-nginx:~$ sudo cp *.html /data/www
```

```
daveyang163@instance-nginx:~$ sudo cp *.css /data/www/css
```

```
daveyang163@instance-nginx:~$ sudo cp *.js /data/www/js
```

4. 修改 /etc/nginx/nginx.conf:

```
daveyang163@instance-nginx:~$ cd /etc/nginx
```

```
daveyang163@instance-3:/etc/nginx$ sudo vim nginx.conf
```

按 i 进入编辑模式。

在最后一行（结尾的}前一行），加入如下配置：

```
server {
        listen 80 default_server;
        index index.html;
        location / {
                root /data/www;
        }
        location /images/ {
                root /data;
       }
  }
```

按 ESC 退出编辑模式;

按:wq 写入退出 vim.

5. Reload nginx

```
daveyang163@instance-3:/etc/nginx$ sudo nginx -s reload
```

6. 浏览器浏览 http://34.67.225.198/，即可看到自己制作的 html 页面了。

附： 部署、应用 Nginx 的官网说明

来源： http://nginx.org/en/docs/beginners_guide.html

The way nginx and its modules work is determined in the configuration file. By default, the configuration file is named nginx.conf and placed in the directory /usr/local/nginx/conf, /etc/nginx, or /usr/local/etc/nginx.

Starting, Stopping, and Reloading Configuration

To start nginx, run the executable file. Once nginx is started, it can be controlled by invoking the executable with the -s parameter. Use the following syntax:

```
nginx -s signal
```

Where signal may be one of the following:

- stop — fast shutdown
- quit — graceful shutdown
- reload — reloading the configuration file
- reopen — reopening the log files

For example, to stop nginx processes with waiting for the worker processes to finish serving current requests, the following command can be executed:

```
nginx -s quit
```

This command should be executed under the same user that started nginx.

Changes made in the configuration file will not be applied until the command to reload configuration is sent to nginx or it is restarted. To reload configuration, execute:

```
nginx -s reload
```

Once the master process receives the signal to reload configuration, it checks the syntax validity of the new configuration file and tries to apply the configuration provided in it. If this is a success, the master process starts new worker processes and sends messages to old worker processes, requesting them to shut down. Otherwise, the master process rolls back the changes and continues to work with the old configuration. Old worker processes, receiving a command to shut down, stop accepting new connections and continue to service current requests until all such requests are serviced. After that, the old worker processes exit.

A signal may also be sent to nginx processes with the help of Unix tools such as the kill utility. In this case a signal is sent directly to a process with a given process ID. The process ID of the nginx master process is written, by default, to the nginx.pid in the directory /usr/local/nginx/logs or /var/run. For example, if the master process ID is 1628, to send the QUIT signal resulting in nginx's graceful shutdown, execute:

```
kill -s QUIT 1628
```

For getting the list of all running nginx processes, the ps utility may be used, for example, in the following way:

```
ps -ax | grep nginx
```

For more information on sending signals to nginx, see Controlling nginx.

Configuration File's Structure

nginx consists of modules which are controlled by directives specified in the configuration file. Directives are divided into simple directives and block directives. A simple directive consists of the name and parameters separated by spaces and ends with a semicolon (;). A block directive has the same structure as a simple directive, but instead of the semicolon it ends with a set of additional instructions surrounded by braces ({ and }). If a block directive can have other directives inside braces, it is called a context (examples: events, http, server, and location).

Directives placed in the configuration file outside of any contexts are considered to be in the main context. The events and http directives reside in the main context, server in http, and location in server.

The rest of a line after the # sign is considered a comment.

Serving Static Content

An important web server task is serving out files (such as images or static HTML pages). You will implement an example where, depending on the request, files will be served from different local directories: /data/www (which may contain

HTML files) and /data/images (containing images). This will require editing of the configuration file and setting up of a server block inside the http block with two location blocks.

First, create the /data/www directory and put an index.html file with any text content into it and create the /data/images directory and place some images in it.

Next, open the configuration file. The default configuration file already includes several examples of the server block, mostly commented out. For now comment out all such blocks and start a new server block:

```
http {
    server {
    }
}
```

Generally, the configuration file may include several server blocks distinguished by ports on which they listen to and by server names. Once nginx decides which server processes a request, it tests the URI specified in the request's header against the parameters of the location directives defined inside the server block.

Add the following location block to the server block:

```
location / {
    root /data/www;
}
```

This location block specifies the "/" prefix compared with the URI from the request. For matching requests, the URI will be added to the path specified in the root directive, that is, to /data/www, to form the path to the requested file on the local file system. If there are several matching location blocks nginx selects the one with the longest prefix. The location block above provides the shortest prefix, of length one, and so only if all other location blocks fail to provide a match, this block will be used.

Next, add the second location block:

```
location /images/ {
    root /data;
}
```

It will be a match for requests starting with `/images/` (location `/` also matches such requests, but has shorter prefix).

The resulting configuration of the `server` block should look like this:

```
server {
    location / {
        root /data/www;
    }

    location /images/ {
        root /data;
    }
}
```

This is already a working configuration of a server that listens on the standard port 80 and is accessible on the local machine at `http://localhost/`. In response to requests with URIs starting with `/images/`, the server will send files from the `/data/images` directory. For example, in response to the `http://localhost/images/example.png` request nginx will send the `/data/images/example.png` file. If such file does not exist, nginx will send a response indicating the 404 error. Requests with URIs not starting with `/images/` will be mapped onto the `/data/www` directory. For example, in response to the `http://localhost/some/example.html` request nginx will send the `/data/www/some/example.html` file.

To apply the new configuration, start nginx if it is not yet started or send the `reload` signal to the nginx's master process, by executing:

```
nginx -s reload
```

In case something does not work as expected, you may try to find out the reason in `access.log` and `error.log` files in the directory `/usr/local/nginx/logs` or `/var/log/nginx`.