Intro

在本任务中，您需要通过 Docker 来搭建 Nginx 反向代理服务器。

具体来说，您将使用 Docker 命令、Dockerfile 指令和相关字段的知识。您需要通过将给出的 Linux 命令转换为 Dockerfile 指令来编写来编译一个属于你的 Docker 镜像。

总而言之，首先请根据以下信息，给出一个 Dockerfile。

Background

Docker

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中，然后发布到任何流行的 Linux 或 Windows 机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。

你可以认为 Docker 是一种特殊的虚拟机。在运维中，我们可以保存或者构建一个特殊用途的 Docker 容器的雏形，并在需要的时候实时进行部署。这样的「雏形」被称为 Docker Image (镜像)。构建 Docker 镜像最普遍的方式就是通过编写 Dockerfile。

构建一个 Docker 镜像并运行的主要流程如下：

1. 编写 Dockerfile；
2. 执行 `docker build` 命令，根据 Dockerfile 构建一个镜像；
3. 执行 `docker run` 命令，运行构建好的镜像。

Dockerfile

Dockerfile 应至少包含以下字段：

- FROM
- RUN
- CMD

FROM：定制的镜像都是基于 FROM 的镜像，即从 FROM 的镜像为基础开始进行后续构建。

RUN：用于执行后面跟着的命令行命令。

CMD：类似于 RUN 指令，用于运行程序，但二者运行的时间点不同：CMD 在docker run 时运行，RUN 是在 docker build 时运行。

下面是一个极简的 Dockerfile 实例，它输出两行信息，并进入死循环。

```
FROM ubuntu:18.04

RUN echo 'Hello Docker!'
RUN echo 'Hello LanUnion!'

CMD while $true; do sleep 1; done;
```

将该文件保存并命名为 dockerfile，执行：

```
docker build -t lanunion/hello .
```

我们就得到了一个标签为 `lanunion/hello` 的镜像，可以通过 `docker images` 命令来查看。

如果我们想运行这个镜像，就可以直接通过

```
docker run -it -d --name Hello lanunion/hello
```

启动目标 Docker 容器，并给该容器命名为「Hello」。

Nginx

Nginx (读作 “engine X”) 是一款 web 服务器，它还可以用作反向代理、负载均衡器、邮件代理和HTTP缓存。该软件由伊戈尔·赛索夫 (Igor Sysoev) 开发，并于 2004 年公开发布。Nginx 是免费和开源的软件，在 BSD 许可协议的条款下发布。大部分 web 服务器使用 Nginx，通常用作负载均衡器。Nginx 可以通过 FastCGI、脚本 SCGI 处理程序、WSGI 应用服务器部署来提供网络上的动态 HTTP 内容，它还可以作为一个软件负载均衡器。Nginx默认配置文件为 `nginx.conf`。

在 Linux 中安装 Nginx 与 PHP-FPM

以下环境为 Ubuntu 18.04

安装依赖

```
# 更新软件源
apt-get update

# 运行此命令以安装构建依赖项
apt-get install -y curl gcc make autoconf libc-dev zlib1g-dev pkg-config apt-utils

# 运行此命令以安装其他依赖项和库
apt-get install -y gnupg2 dirmngr wget apt-transport-https lsb-release ca-certificates software-properties-common
```

安装 Nginx、PHP 以及 PHP-FRM

运行此命令以添加 PHP 存储库，其中可以找到最新版本的 PHP，并更新包列表。

```
add-apt-repository ppa:ondrej/php && apt-get update
```

运行此命令安装Nginx、PHP 7.4 和 PHP-FPM 7.4。

```
apt-get install nginx php7.4 php7.4-cli php7.4-fpm php7.4-json php7.4-pdo php7.4-pgsql php7.4-zip php7.4-gd php7.4-mbstring php7.4-curl php7.4-xml php7.4-bcmath php7.4-json
```

运行 Nginx

```
nginx -s reload
```