### **安装 Docker**

yum -y install docker

开启docker服务systemctl start docker

**Docker Compose 安装**

Docker Compose 是 Docker 的独立产品，因此需要安装 Docker 之后在单独安装 Docker Compose .

#下载

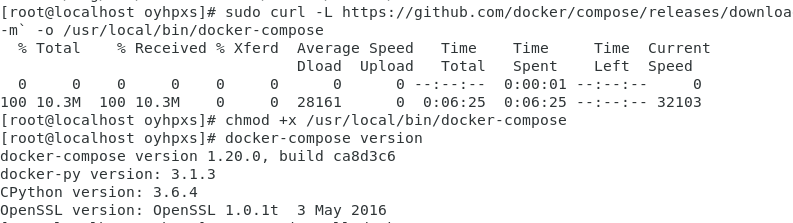
sudo curl -L https://github.com/docker/compose/releases/download/1.20.0/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

#安装

chmod +x /usr/local/bin/docker-compose

#查看版本

docker-compose version



**step\_01：准备镜像**

项目部署需要3个基础镜像外加一个izone的构建镜像（可以自己拉取代码构建，也可以拉取我构建好的镜像），拉取镜像的命令：

# 3个基础镜像

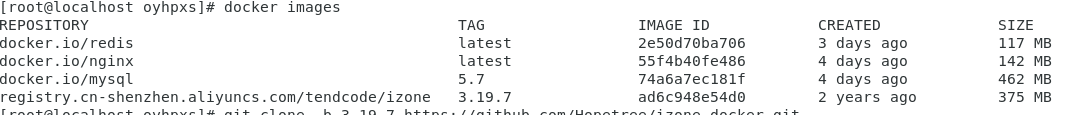
docker pull nginx

docker pull redis

docker pull mysql:5.7

# 1个自构建镜像，镜像对应的编排版本也使用这个tag，后续会说到

docker pull registry.cn-shenzhen.aliyuncs.com/tendcode/izone:3.19.7



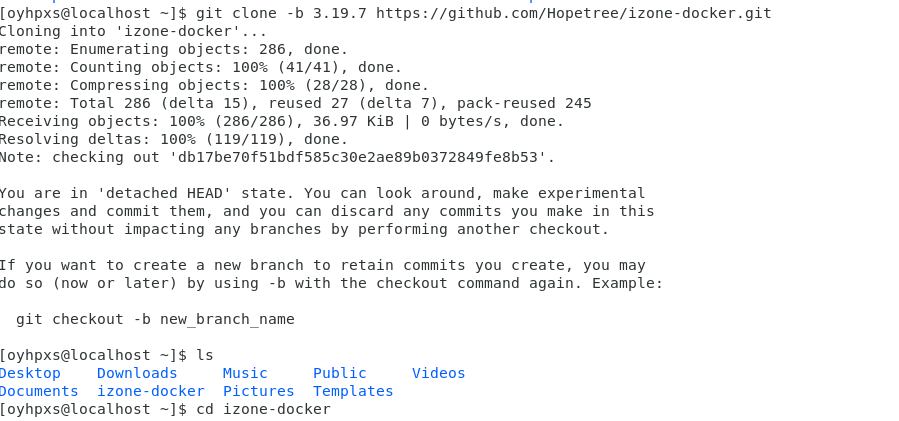
**安装Git**

yum -y install git

**step\_02：拉取项目代码**

为了避免挂载目录权限问题，后续命令全部使用root账号执行。拉取代码，直接拉取跟镜像的tag一样的项目tag：

git clone -b 3.19.7 <https://github.com/Hopetree/izone-docker.git>



**step\_03：创建环境变量文件**

项目依赖两个环境变量文件，必须创建，第一个文件是 .env，这个文件是docker-compose 默认使用的环境变量问题，可以把参数传递到 docker-compose 中。进入izone-docker 目录，创建两个问题，文件内容如下：

# .env

# db

MYSQL\_IMAGE=mysql:5.7

MYSQL\_ROOT\_PASSWORD=1314520@abc

# redis

REDIS\_IMAGE=redis

# web

IZONE\_IMAGE=registry.cn-shenzhen.aliyuncs.com/tendcode/izone:3.19.7

IZONE\_MYSQL\_NAME=izone

# nginx

NGINX\_IMAGE=nginx

然后创建一个 izone.env 文件，这个里面的环境变量都会传递给 izone 使用，至于izone 可以设置哪些环境变量，可以去 izone 项目的 settings 文件中查看。这里可以虽然传入几个变量（其实可以一个也不传，因为环境变量不传的时候有默认值使用）：

# izone.env

# 个性化配置

IZONE\_SECRET\_KEY=#!kta!9e0)24p@9#=\*=ra$r!0k0+p5@w+a%7g1bboo9+9080

IZONE\_TOOL\_FLAG=True

IZONE\_API\_FLAG=False

IZONE\_DEBUG=False

IZONE\_SITE\_END\_TITLE=izone

IZONE\_SITE\_DESCRIPTION=izone 博客

IZONE\_SITE\_KEYWORDS=Django博客,Python web开发,个人博客

IZONE\_GITHUB=https://github.com/

# 邮箱配置

IZONE\_EMAIL\_HOST=smtp.163.com

IZONE\_EMAIL\_HOST\_USER=your-email

IZONE\_EMAIL\_HOST\_PASSWORD=77777

IZONE\_EMAIL\_PORT=465IZONE\_EMAIL\_USE\_SSL=True

IZONE\_DEFAULT\_FROM\_EMAIL=izone博客 <test@163.com>

IZONE\_ACCOUNT\_EMAIL\_VERIFICATION=optional

# 非必须设置

IZONE\_CNZZ\_PROTOCOL=''

IZONE\_BEIAN=''

IZONE\_SITE\_VERIFICATION=''

# 配置管理员邮箱，格式：name|test@test.com 多组用户用英文逗号隔开

IZONE\_ADMIN\_EMAIL\_USER=name|test@test.com

两个文件创建成功之后可以检查一下文件是合格，执行命令：

Plain Text

docker-compose config

没有报错就说明格式没问题，可以查看一下环境变量是否传错

**step\_04：修改 nginx配置**

由于项目默认给的nginx配置是我自己项目的，所以不适合其他人使用，这里把项目代码中nginx/conf.d/ 目录中的配置文件全部删除，然后在nginx/conf.d/ 目录中创建一个通用的配置（使用 localhost）：

cd nginx/conf.d && rm -f server\_izone.conf

vi nginx.conf

default.conf 的内容如下：

server {

# 端口和域名

listen 80;

server\_name localhost;

# static 和 media 的地址

location /static/ {

root /opt/izone;

}

location /media/ {

root /opt/izone;

}

# web 服务使用80端口，并且添加别名跟本地域名保持一致

location / {

proxy\_pass http://web:8000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

# 其他配置

client\_max\_body\_size 1m;

client\_header\_buffer\_size 128k;

client\_body\_buffer\_size 1m;

proxy\_buffer\_size 32k;

proxy\_buffers 64 32k;

proxy\_busy\_buffers\_size 1m;

proxy\_temp\_file\_write\_size 512k;

}

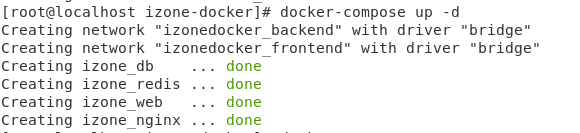
临时关闭：

[root@localhost ~]# getenforce  
Enforcing

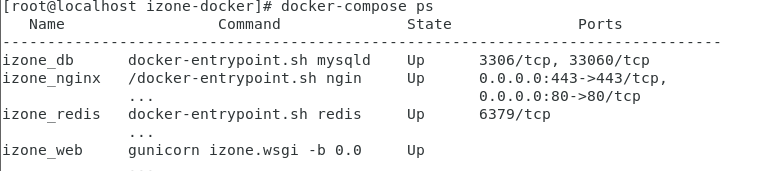
[root@localhost ~]# setenforce 0  
[root@localhost ~]# getenforce  
Permissive

**step\_05：部署容器**

退回到 izone-docker 目录，启动容器的命令（加 -d 是后台运行），可以看到输出



然后可以查看一下容器运行状态，如果都是up 就没问题：



**step\_06：初始化数据库**

虽然容器都起来了，但是izone 项目的数据库里面是空的，所以要先初始化一下数据库，直接在izone-docker 目录下执行如下命令：

# 一条一条的输入

# 初始化数据

docker exec -it izone\_web python manage.py makemigrations

docker exec -it izone\_web python manage.py migrate

# 收集静态文件

docker exec -it izone\_web python manage.py collectstatic

# 初始化搜索索引

docker exec -it izone\_web python manage.py update\_index

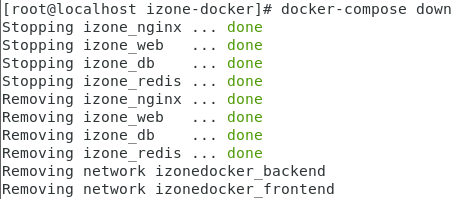
# 创建管理员账号

docker exec -it izone\_web python manage.py createsuperuser

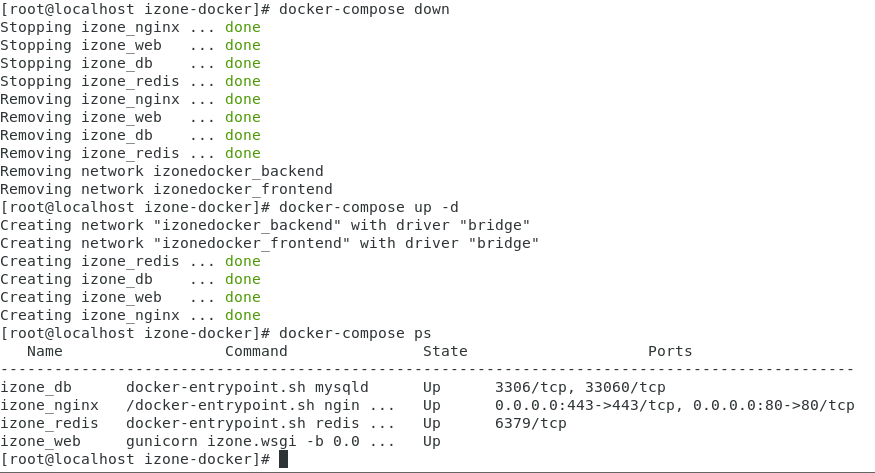
**step7：重启服务，浏览页面**

在izone-docker 目录下，依次执行关闭和启动命令，重启服务：

[root@CentOS-2 izone-docker]# docker-compose down



[root@CentOS-2 izone-docker]# docker-compose up -d



浏览器输入服务器（或者虚拟机）的IP地址查看页面

