

docker 安装 以及简单项目部署

1.0 准备阶段

1.0.1 购买服务器

首先我们确保我们的运行是在Linux环境运行,所以我们需要一台linux服务器,没有小伙伴可以购买阿里云学生机。当然我们也为了没钱的小伙伴准备了人性化的活动,阿里云高校计划。(当然规定是全日制在校学生才可以)

阿里云高校计划传送门: <https://developer.aliyun.com/adc/student/>

阿里云学生机购买传送门: <https://promotion.aliyun.com/ntms/act/campus2018.html>

1.0.2 准备下载连接工具

当买好阿里云服务器后 我们需要去xftp官网申请一下xftp 和xshell (学生免费)

这款工具简而言之就是方便我们连接远程我们的服务器。

冲!我们的准备工作就完成的。

2.0 docker 安装

2.1 docker官网

地址: <https://docs.docker.com/engine/install/centos/>

2.1.1 运行命令

```
//清除服务器上docker
sudo yum remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-engine

-----

//安装工具
sudo yum install -y yum-utils

-----

//配置镜像(官网镜像太慢了)所以我们找到一个阿里的镜像地址 听懂了没有?
sudo yum-config-manager --add-repo http://mirrors.aliyun.com/docker-
ce/linux/centos/docker-ce.repo

-----

sudo yum install docker-ce docker-ce-cli containerd.io

-----
```

```
//启动docker
```

```
sudo systemctl start docker
```

```
//输入命令 查看当前容器
```

```
docker ps
```

```
//出现CONTAINER ID
```

```
IMAGE
```

```
COMMAND
```

```
CREATED
```

```
STATUS
```

```
PORTS
```

```
NAMES
```

代表安装docker成功！ 下班 byebye。

2.1.2 配置阿里镜像加速（这一步可以不用配 配了就会更快更高效 便捷）

```
//阿里镜像加速教程
```

```
https://developer.aliyun.com/article/692983
```

2.1.3

我们需要去阿里云去开放我们的安全组规则

例如你需要访问你服务器8089端口 这时候你就需要配置服务器安全组规则其中一个8089

当然一个个端口开放比较麻烦 偷懒的小技巧是 开放所有端口（可能有安全问题）



docker 安装完毕！

3.0 docker安装GitLab：

```
//下拉最新gitLab
```

```
docker pull gitlab/gitlab-ce:latest
```

```
# 启动
```

```
docker run --detach \
```

```
--publish 443:443 --publish 80:80 --publish 222:22 \
```

```
--name gitlab \
```

```
--memory 4g \
```

```
--restart always \
--volume /srv/gitlab/config:/etc/gitlab \
--volume /srv/gitlab/logs:/var/log/gitlab \
--volume /srv/gitlab/data:/var/opt/gitlab \
gitlab/gitlab-ce:latest
```

修改ssh端口

```
vi /etc/ssh/sshd_config
```

找到 #port 讲其改为 port 任意端口 注意不要和已知端口冲突

2.重启sshd服务

```
systemctl restart sshd
```

#3.重启container

#查看当前运行容器

```
docker ps
```

#找到containerId

```
docker restart containerId
```

重启gitlab至此gitlab搭建完成

--hostname gitlab.example.com: 设置主机名或域名

--detach 如果在docker run后面追加-d=true或者-d, 那么容器将会运行在后台模式。此时所有I/O数据只能通过网络资源或者共享卷组来进行交互。

--publish 8443:443: 将http: 443映射到外部端口8443

--publish 8880:80: 将web: 80映射到外部端口8880

--publish 8222:22: 将ssh: 22映射到外部端口8222

--name gitlab: 运行容器名

--restart always: 自动重启

--volume /srv/gitlab/config:/etc/gitlab: 挂载目录

--volume /srv/gitlab/logs:/var/log/gitlab: 挂载目录

--volume /srv/gitlab/data:/var/opt/gitlab: 挂载目录

--privileged=true 使得容器内的root拥有真正的root权限。否则, container内的root只是外部的一个普通用户权限

参考

https://segmentfault.com/a/1190000019772866?utm_source=sf-related

4.0结语

这是docker容器技术初探。简简单单容器部署jar包。

不足之处在于 每次部署需要终止容器 删除镜像 替换jar包 或者.sh文件有问题就需要吧镜像容器全部删除

重新运行docker build 以及docker run 命令 比较麻烦 很讨厌 下次分享更简单部署

