# 全球通大厦

## yum -y install lrzsz

## 参考文档：

<https://www.bilibili.com/read/cv17617673/>

## 1：离线部署docker

### 1.1：下载docker的压缩文件：

<https://download.docker.com/linux/static/stable/x86_64/>

### 1.2：上传文件到/usr/local：rz命令

### 1.3：解压docker文件：tar -zxvf docker

### 1.4：将解压出来的docker文件内容移动到 /usr/bin/ 目录下：cp docker/\* /usr/bin/

### 1.5：在/etc/systemd/system目录下创建docker.service文件：vim /etc/systemd/system/docker.service

#### 1.5.1：在文件中配置以下内容：

[Unit]

Description=Docker Application Container Engine

Documentation=https://docs.docker.com

After=network-online.target firewalld.service

Wants=network-online.target

[Service]

Type=notify

# the default is not to use systemd for cgroups because the delegate issues still

# exists and systemd currently does not support the cgroup feature set required

# for containers run by docker

ExecStart=/usr/bin/dockerd

ExecReload=/bin/kill -s HUP $MAINPID

# Having non-zero Limit\*s causes performance problems due to accounting overhead

# in the kernel. We recommend using cgroups to do container-local accounting.

LimitNOFILE=infinity

LimitNPROC=infinity

LimitCORE=infinity

# Uncomment TasksMax if your systemd version supports it.

# Only systemd 226 and above support this version.

#TasksMax=infinity

TimeoutStartSec=0

# set delegate yes so that systemd does not reset the cgroups of docker containers

Delegate=yes

# kill only the docker process, not all processes in the cgroup

KillMode=process

# restart the docker process if it exits prematurely

Restart=on-failure

StartLimitBurst=3

StartLimitInterval=60s

[Install]

WantedBy=multi-user.target

### 1.6：添加文件权限：

chmod +x /etc/systemd/system/docker.service

### 1.7：重载unit配置文件：systemctl daemon-reload

### 1.8：启动Docker：systemctl start docker

### 1.9：设置开机自启：systemctl enable docker.service

### 1.10：查看Docker状态：systemctl status docker

### 1.11：查看docker版本：docker --version

## nginx容器安装：

### 1：上传压缩的nginx镜像至/usr/local：rz命令

### 2：将nginx压缩镜像还原：docker load命令：

### docker load -i nginx.tar

### 3：利用数据卷挂载配置文件以及html文件

#### 3.1：新建html数据卷：

docker volume create html

#### 3.2：查看html数据卷具体信息：docker volume inspect html

#### 3.2：新建目录：mkdir -p /usr/local/nginx/

#### 3.3：新建conf目录：mkdir -p /usr/local/nginx/conf

#### 3.4：新建html目录：mkdir -p /usr/local/nginx/html

#### 3.4：新建log目录：mkdir -p /usr/local/nginx/log

#### 3.5：运行nginx容器：docker run --name mynginx -p 80:80 -d nginx

#### 3.6：复制容器的nginx.conf文件：

docker cp mynginx:/etc/nginx/nginx.conf /usr/local/nginx/conf/nginx.conf

#### 3.6：复制容器的default.conf文件：

docker cp mynginx:/etc/nginx/conf.d /usr/local/nginx/conf/conf.d

#### 3.7：复制容器的html文件：

docker cp mynginx:/usr/share/nginx/html /usr/local/nginx/

#### 3.7：停止nginx：docker stop mynginx

#### 3.8：删除nginx：docker rm mynginx

#### 3.9：运行容器同时挂载数据卷：

docker run -d --name mynginx -p 80:80 -v /usr/local/nginx/html:/usr/share/nginx/html -v /usr/local/nginx/conf/nginx.conf:/etc/nginx/nginx.conf -v /usr/local/nginx/log:/var/log/nginx -v /usr/local/nginx/conf/conf.d:/etc/nginx/conf.d --privileged=true nginx

## Redis容器安装：

### 1：上传压缩的redis镜像至/usr/local：rz命令

### 2：将redis压缩镜像还原：docker load命令：

### docker load -i redis.tar

### 3：利用数据卷挂载配置文件

#### 3.1：新建目录：mkdir -p /home/redis/myredis

#### 3.2：在myredis中新建data目录：mkdir /home/redis/myredis/data

#### 3.3：上传配置文件myredis.conf到/home/redis/myredis目录：rz命令

#### 3.4：启动容器：

docker run --restart=always --log-opt max-size=100m --log-opt max-file=2 -p 6379:6379 --name myredis -v /home/redis/myredis/myredis.conf:/etc/redis/redis.conf -v /home/redis/myredis/data:/data -d redis redis-server /etc/redis/redis.conf --appendonly yes --requirepass 7EwjICun6l

## MySQL容器安装（bug）：

## 1：上传压缩的MySQL镜像至/usr/local：rz命令

## 2：将MySQL压缩镜像还原：docker load命令

### docker load -i musql.tar

## 3：创建目录/tem/mysql/data：mkdir -p /tem/mysql/data

## 4：创建目录/tem/mysql/conf：mkdir -p /tem/mysql/conf

docker run -p 3306:3306 --name mysql-server -v /tem/mysql/conf:/etc/mysql/conf.d -v /tem/mysql/logs:/logs -v /tem/mysql/data:/var/lib/mysql -e MYSQL\_ROOT\_PASSWORD=IFCYyueanmy123456 -d mysql --lower\_case\_table\_names=1

## MongoDB容器安装：

### 1：上传压缩的MongoDB镜像：rz 命令

### 2：将MongoDB压缩镜像还原：

Docker load -i mongodb.tar

2：创建一个文件夹，用作mongodb的数据目录挂载：

mkdir /usr/local/mongodb

2.2：创建一个文件夹，用作/etc/mongod.conf.orig配置文件挂载

Mkdir /usr/local/mymongo

2.3：运行mongodb容器

docker run -itd --name mongo -v /usr/local/mongodb:/data/db -e MONGO\_INITDB\_ROOT\_USERNAME=gdyaadmin -e MONGO\_INITDB\_ROOT\_PASSWORD=gdya1994 -p 27017:27017 mongo --auth

2.4：拷贝/etc/mongod.conf.orig配置文件到/usr/local/mymongo

docker cp mongo:/etc/mongod.conf.orig /usr/local/mymongo

2.5：停止mongo：docker stop mongo

2.6删除mongo：docker rm mongo

3：安装mongo：

docker run -itd --name mongo -v /usr/local/mongodb:/data/db -v /usr/local/mymongo/mongod.conf.orig:/etc/mongod.conf.orig -e MONGO\_INITDB\_ROOT\_USERNAME=gdyaadmin -e MONGO\_INITDB\_ROOT\_PASSWORD=gdya1994 -p 27017:27017 mongo --auth

5：配置外部可以连接操作服务器中的 MongoDB：

5.1：进入mongo：docker exec -it mongo容器id或mongo容器名bash

5.2：更新源：apt-get update

5.3：安装vim：apt-get install vim

5.4：修改mongo的配置文件：vim /etc/mongod.conf.orig

5.4.1：将`bindIp: 127.0.0.1`注释掉；或者改成`bindIp: 0.0.0.0`。即可开启远程连接

5.4.2：开启权限认证

security：

authorization: enabled

6：在5.1执行之后，进入mongo命令行：mongo

7：切换到admin：use admin

8；认证：db.auth(‘gdyaadmin’, ‘gdya1994’)

9：显示数据库：show dbs

10：显示表：show tables

11：在执行6之后，退出mongo：exit

12：在11之后，退出mongo容器：exit

13：开放27017端口

14：使用navicat连接mongo：



15：新建gdya数据库，运行脚本进行数据导入

docker cp mongo:/etc/mongod.conf.orig /usr/local/mymongo/

docker run -itd --name mongo -v /usr/local/mongodb:/data/db -v /usr/local/mymongo/mongod.conf.orig:/etc/mongod.conf.orig -e MONGO\_INITDB\_ROOT\_USERNAME=gdyaadmin -e MONGO\_INITDB\_ROOT\_PASSWORD=gdya1994 -p 27017:27017 mongo --auth

## RabbitMQ容器安装：

### 1：上传压缩的RabbitMQ镜像：rz 命令

### 2：将RabbitMQ压缩镜像还原:docker load命令

### 3：运行RabbitMQ容器