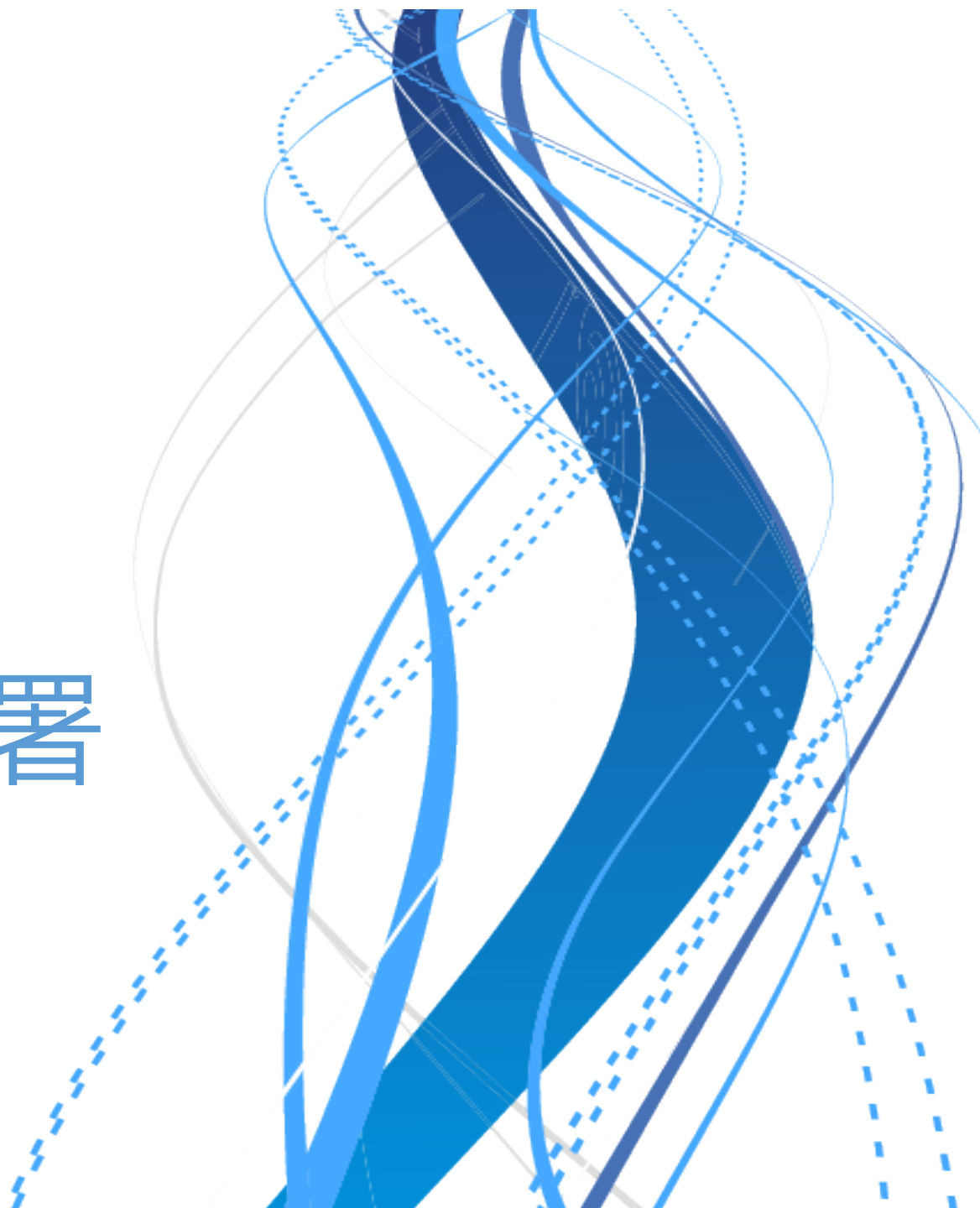




Linux内网服务器部署

汇报人:XX

日期:xxx



1 2、Xftp文件夹名称乱码

3 4、镜像文件上传

5 5、JDK镜像安装

7 7、postgresql镜像安装

9 9、postgresql

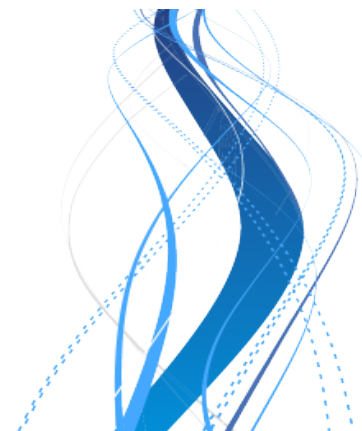
2 3、准备工作

4 5、解压文件

6 6、nginx镜像安装

8 8、redis镜像安装

10 10、后端部署



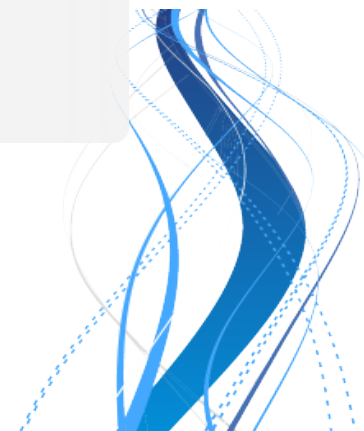
Linux内网服务器部署

1. centos7-linux安装docker(离线方式)

进入将docker-18.06.3-ce.tgz文件上传到centos7-linux系统上：用ftp工具上传即可

解压

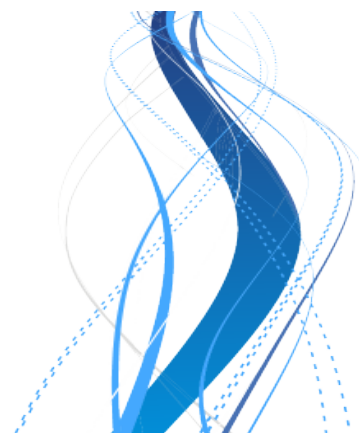
```
tar zxvf docker-18.06.3-ce.tgz
```



Linux内网服务器部署

- ▶ 将解压出：来的docker文件复制到 /usr/bin/ 目录

```
1 [root@localhost java]# tar -zxvf docker-18.06.3-ce.tgz
```



Linux内网服务器部署

```
cp docker/* /usr/bin/
```

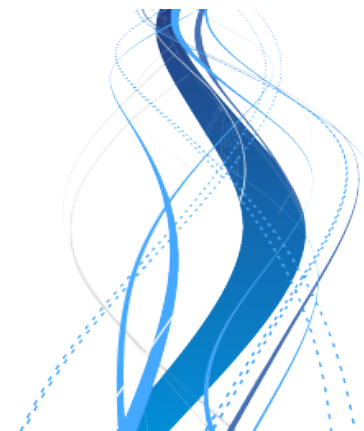
▶ 进入/etc/systemd/system/
目录：复制docker.service
文件至该目录下

▶ cp docker.service
/etc/systemd/system

▶ 打开do: cker.service
文件

▶ vi /e:
tc/systemd/system/docker.servic
e

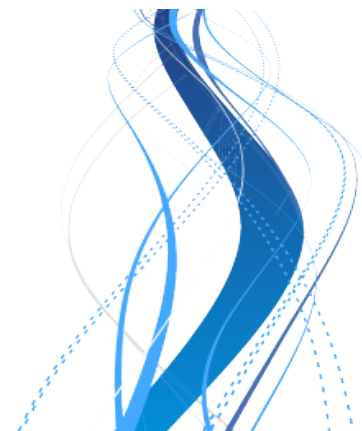
```
1 [root@localhost java]# cp docker/* /usr/bin/
```



Linux内网服务器部署

```
1 [root@localhost system]# vi docker.service
```

▶ 修改一：
insecure-
registry=192.168.
200.128 此处改为
你自己服务器ip



Linux内网服务器部署

给doc:
ker.service文
件添加执行权
限

```
chmod 777  
/etc/systemd/  
system/docker.  
service
```


```
1 [Unit]
2 Description=Docker Application Container Engine
3 Documentation=https://docs.docker.com
4 After=network-online.target firewalld.service
5 Wants=network-online.target
6
7 [Service]
8 Type=notify
9 # the default is not to use systemd for cgroups because the delegate issues still
10 # exists and systemd currently does not support the cgroup feature set required
11 # for containers run by docker
12 ExecStart=/usr/bin/dockerd --selinux-enabled=false --insecure-registry=192.168.200.128
13 ExecReload=/bin/kill -s HUP $MAINPID
14 # Having non-zero Limit*s causes performance problems due to accounting overhead
15 # in the kernel. We recommend using cgroups to do container-local accounting.
16 LimitNOFILE=infinity
17 LimitNPROC=infinity
18 LimitCORE=infinity
19 # Uncomment TasksMax if your systemd version supports it.
20 # Only systemd 226 and above support this version.
21 #TasksMax=infinity
22 TimeoutStartSec=0
23 # set delegate yes so that systemd does not reset the cgroups of docker containers
24 Delegate=yes
25 # kill only the docker process, not all processes in the cgroup
26 KillMode=process
27 # restart the docker process if it exits prematurely
```

Linux内网服务器部署

```
1 [root@localhost system]# chmod 777 /etc/systemd/system/docker.service
```

重新加载：配置
文件(每次有修改
docker.service
文件时都要重新
加载下)

```
systemctl  
daemon-reload
```

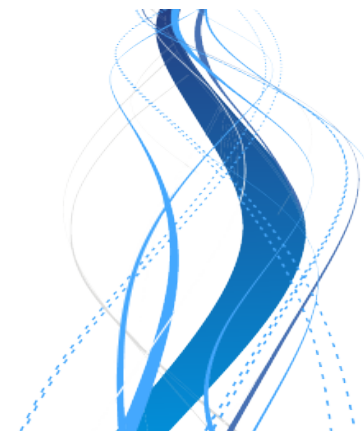


Linux内网服务器部署

启动docker

```
systemctl  
start docker
```

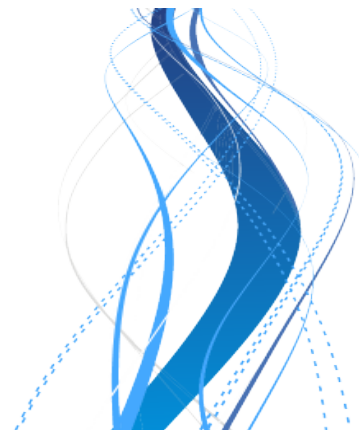
```
1 [root@localhost system]# systemctl daemon-reload
```



Linux内网服务器部署

```
1 [root@localhost system]# systemctl start docker
```

▶ 查看docker状态



Linux内网服务器部署

systemctl status docker

如果启动失败，需要注意虚拟机内存是否给够了，过小会启动失败

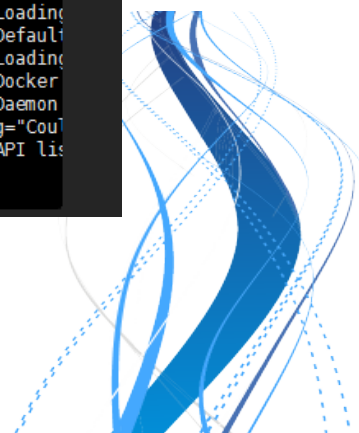
vi

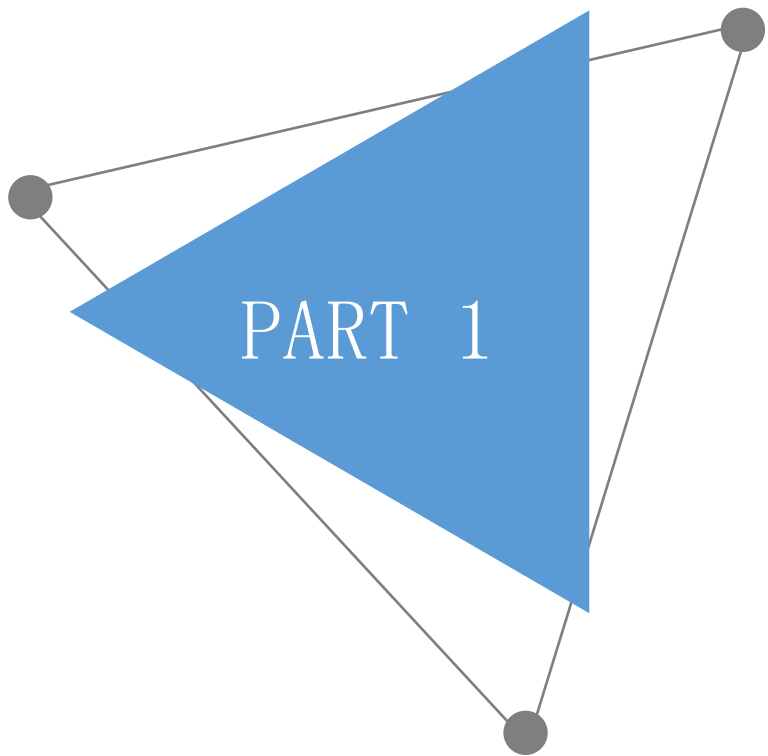
/etc/docker/daemon.json

```
{ "graph": "mnt/docker-  
data", "storage-  
driver": "overlay"  
}
```

出现下面这个界面就代表docker安装成功。

```
[root@localhost system]# systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/etc/systemd/system/docker.service; enabled; vendor preset: disabled)  
   Active: inactive (dead)  
     Docs: https://docs.docker.com  
[root@localhost system]# systemctl start docker  
[root@localhost system]# systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/etc/systemd/system/docker.service; enabled; vendor preset: disabled)  
   Active: active (running) since 四 2020-05-14 16:50:32 CST; 2s ago  
     Docs: https://docs.docker.com  
  Main PID: 2249 (dockerd)  
    Tasks: 17  
   Memory: 21.1M  
   CGroup: /system.slice/docker.service  
           └─2249 /usr/bin/dockerd --selinux-enabled=false --insecure-registry=192.168.200.128  
             └─2255 docker-containerd --config /var/run/docker/containerd/containerd.toml  
  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.492379906+08:00" level=info msg="pickfi  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.492951532+08:00" level=info msg="pickfi  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.492975666+08:00" level=info msg="Loading  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.678916697+08:00" level=info msg="Default  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.798966587+08:00" level=info msg="Loading  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.830836003+08:00" level=info msg="Docker  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.831139821+08:00" level=info msg="Daemon  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.855238884+08:00" level=warning msg="Cou  
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.863758382+08:00" level=info msg="API lis  
5月 14 16:50:32 localhost.localdomain systemd[1]: Started Docker Application Container Engine.  
Hint: Some lines were ellipsized, use -l to show in full.
```

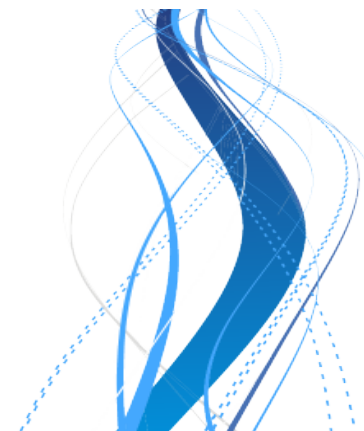


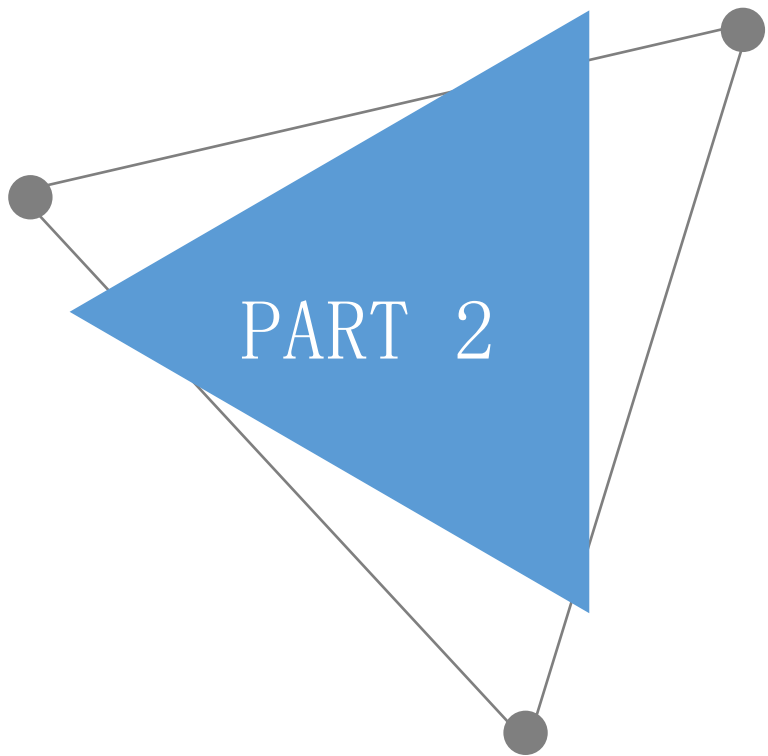


2、Xftp文件夹名称乱码

2、Xftp文件夹名称乱码

左上角 文件->当会话属性->选项->编码改为utf-8





3、准备工作

3、准备工作

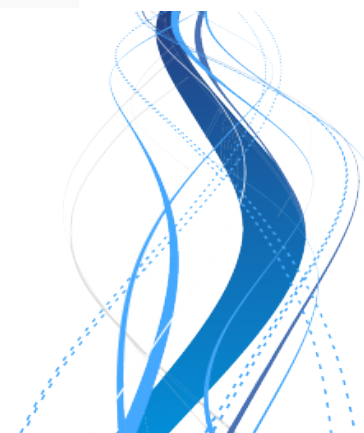
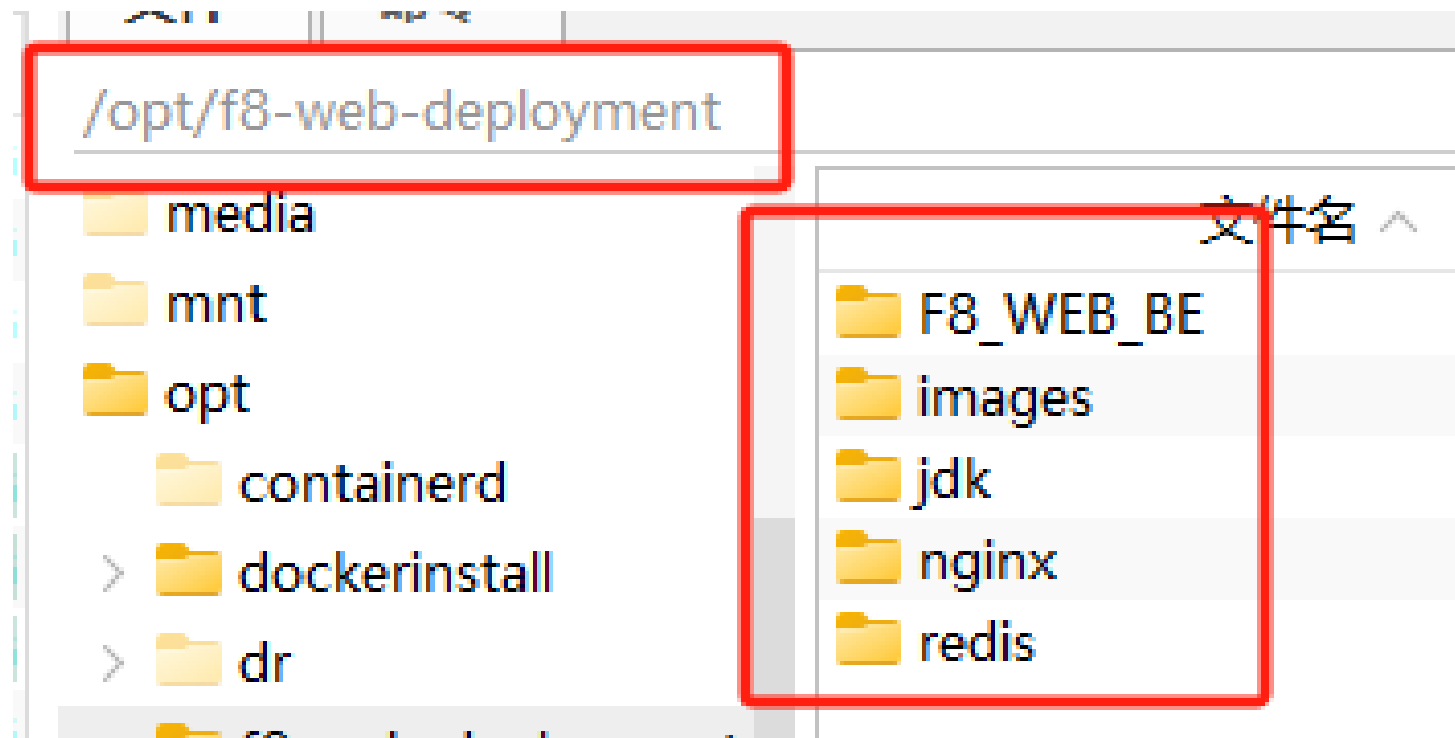
创建以下目录

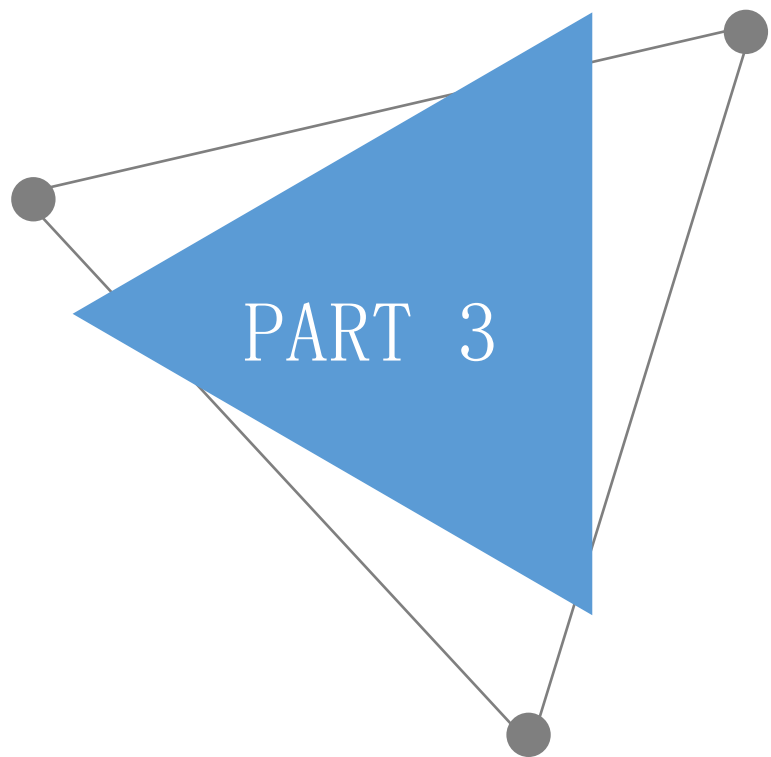
```
Cd /opt
```

```
mkdir f8-web-  
deployment
```

```
Cd /opt/f8-web-  
deployment
```

```
mkdir jdk F8_WEB_BE  
nginx redis
```

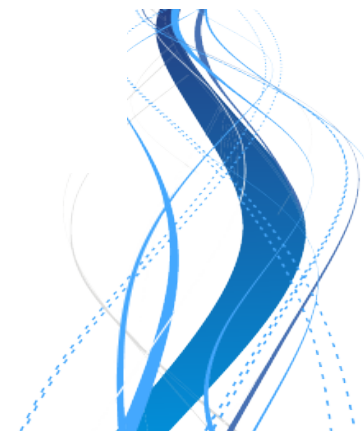
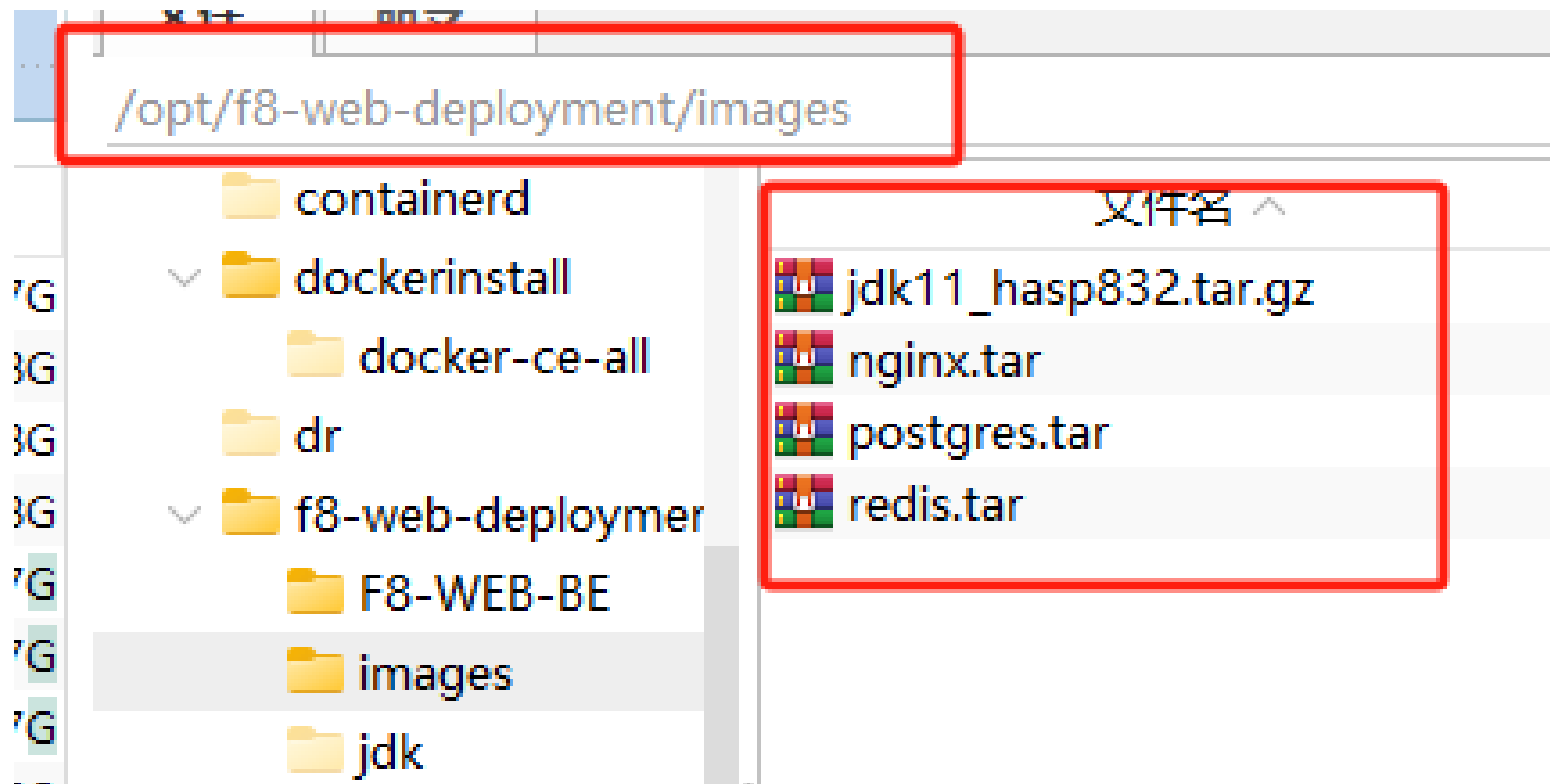


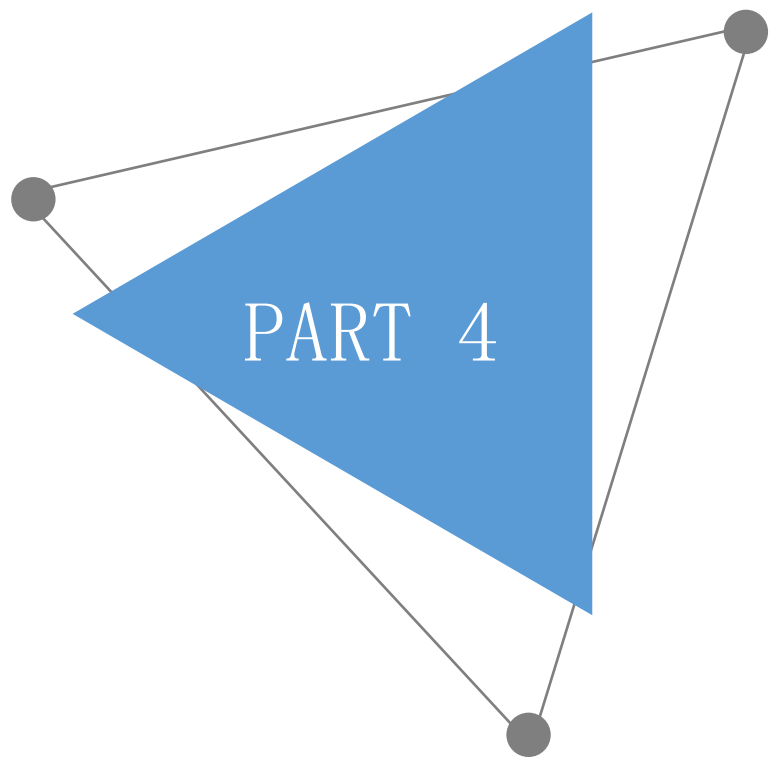


4、镜像文件上传

4、镜像文件上传

新建f8-web-deployment/images文件夹，上传4个镜像文件





5、解压文件

5、解压文件

打开文件夹：`cd /opt/f8-web-deployment/images`
解压jdk镜像文件：`gzip -d jdk11_hasp832.tar.gz`

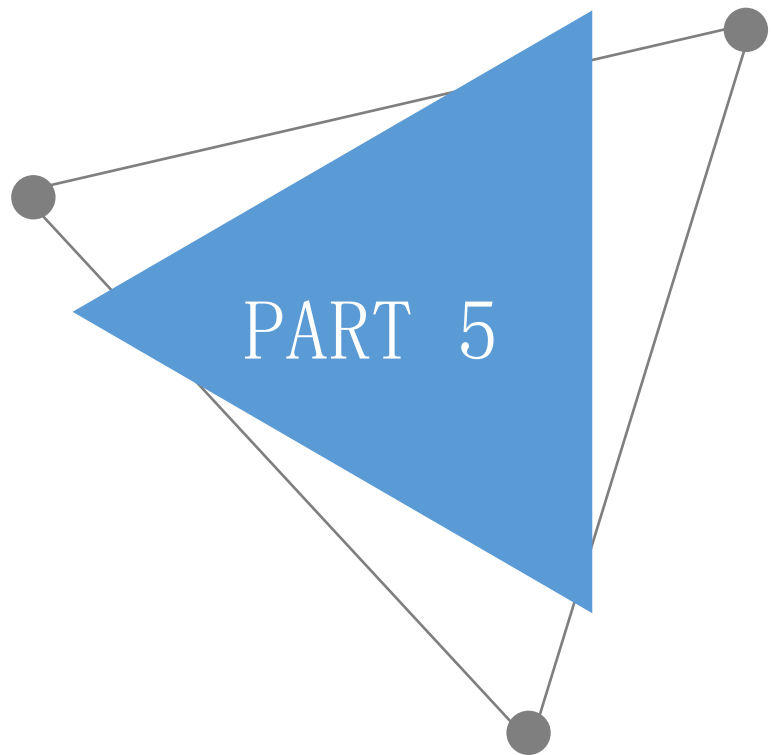
```
[root@ecs-ec82 bin]# cd /opt/f8-web-deployment/images
[root@ecs-ec82 images]# gzip -d jdk11_hasp832.tar.gz
[root@ecs-ec82 images]#
```

命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双击CTRL切换)

文件 命令

/opt/f8-web-deployment/images

文件名	大小
jdk11_hasp832.tar	473.9 MB
jdk11_hasp832.tar.gz	510.3 MB
nginx.tar	130.7 MB
postgres.tar	277 MB
redis.tar	102.6 MB

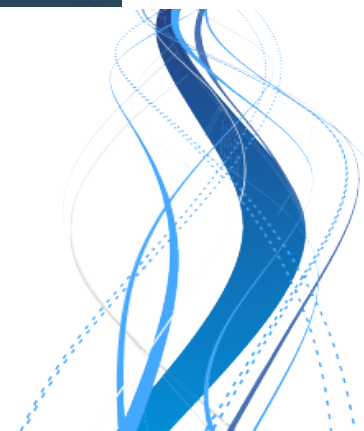


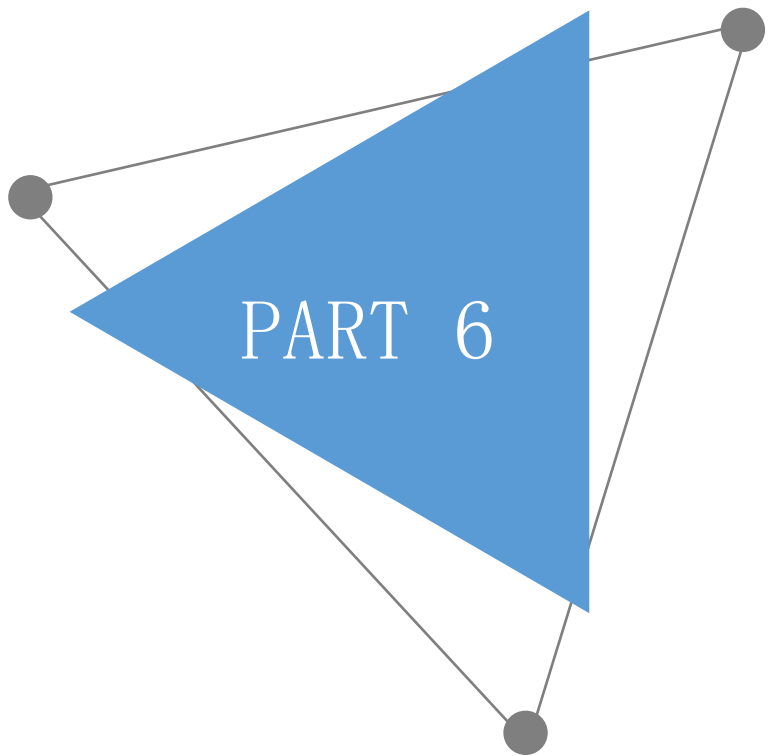
5、JDK镜像安装

5、JDK镜像安装

```
docker load -i  
jdk11_hasp832.tar.g  
z
```

```
[root@ecs-ec82 images]# docker load -i jdk11_hasp832.tar  
174f56854903: Loading layer [=====>] 211.7MB/211.7MB  
d893424c36dc: Loading layer [=====>] 396.6MB/396.6MB  
3e580fb317a6: Loading layer [=====>] 308.4MB/308.4MB  
b5e778844cd9: Loading layer [=====>] 2.048kB/2.048kB  
ade88fd1891e: Loading layer [=====>] 2.56kB/2.56kB  
4c797cle9511: Loading layer [=====>] 147.9MB/147.9MB  
abd893d36e58: Loading layer [=====>] 16.16MB/16.16MB  
Loaded image: jdk11:hasp832
```



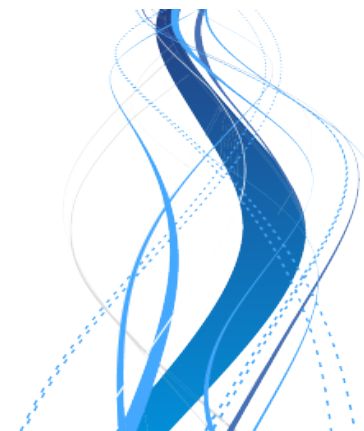


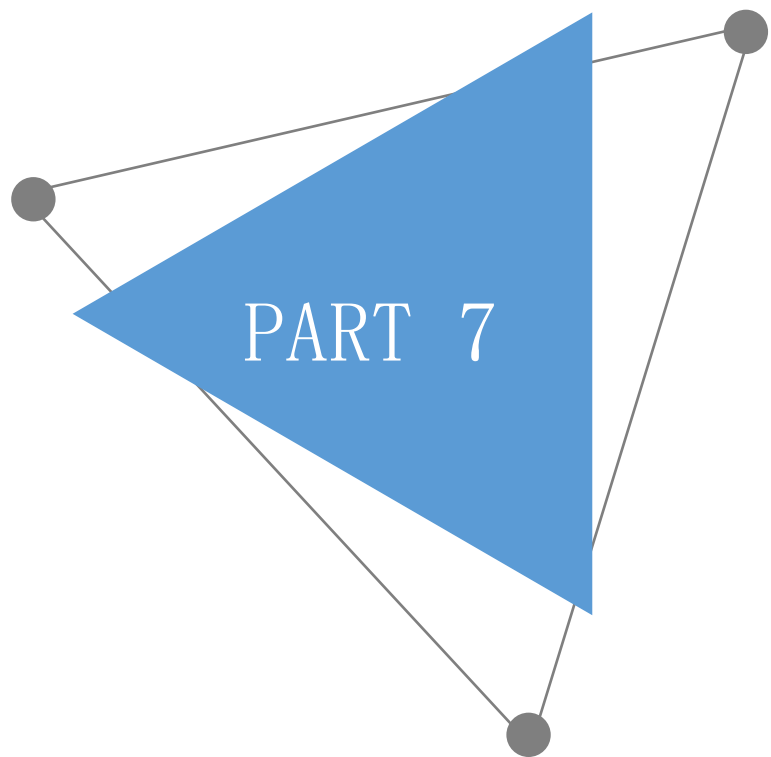
6、nginx镜像安装

6、nginx镜像安装

```
docker load -i  
nginx.tar
```

```
[root@ecs-ec82 images]# docker load -i nginx.tar  
Loaded image: nginx:1.18
```





7、postgresql镜像安装

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/748120021016006072>