Maven私服搭建过程文档

# 简介

# Maven

## Maven私服

## Docker

# 材料

可以连接外网的centos

内外网相同的centos

一台与外网同局域网的部署了IDEA的电脑

# 私服搭建过程

## 外网操作

### 下载和安装docker

使用yum命令下载相应的包。

[root@localhost docker]# yum install --downloadonly --downloaddir=. docker

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

\* base: mirrors.zju.edu.cn

\* extras: mirrors.zju.edu.cn

\* updates: mirrors.huaweicloud.com

Resolving Dependencies

--> Running transaction check

---> Package docker.x86\_64 2:1.13.1-74.git6e3bb8e.el7.centos will be installed

--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================================================================

Package Arch Version Repository Size

=====================================================================================================================================

Installing:

docker x86\_64 2:1.13.1-74.git6e3bb8e.el7.centos extras 16 M

Transaction Summary

=====================================================================================================================================

Install 1 Package

Total download size: 16 M

Installed size: 57 M

Background downloading packages, then exiting:

docker-1.13.1-74.git6e3bb8e.el7.centos.x86\_64.rpm | 16 MB 00:00:35

exiting because "Download Only" specified

在外网安装docker。

[root@localhost docker]# yum -y install docker

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

\* base: mirrors.zju.edu.cn

\* extras: mirrors.zju.edu.cn

\* updates: mirrors.huaweicloud.com

Resolving Dependencies

--> Running transaction check

---> Package docker.x86\_64 2:1.13.1-74.git6e3bb8e.el7.centos will be installed

--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================================================================

Package Arch Version Repository Size

=====================================================================================================================================

Installing:

docker x86\_64 2:1.13.1-74.git6e3bb8e.el7.centos extras 16 M

Transaction Summary

=====================================================================================================================================

Install 1 Package

Total download size: 16 M

Installed size: 57 M

Downloading packages:

docker-1.13.1-74.git6e3bb8e.el FAILED

http://mirrors.zju.edu.cn/centos/7.5.1804/extras/x86\_64/Packages/docker-1.13.1-74.git6e3bb8e.el7.centos.x86\_64.rpm: [Errno 12] Timeout on http://mirrors.zju.edu.cn/centos/7.5.1804/extras/x86\_64/Packages/docker-1.13.1-74.git6e3bb8e.el7.centos.x86\_64.rpm: (28, 'Resolving timed out after 30568 milliseconds')

Trying other mirror.

docker-1.13.1-74.git6e3bb8e.el7.centos.x86\_64.rpm | 16 MB 00:00:39

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Installing : 2:docker-1.13.1-74.git6e3bb8e.el7.centos.x86\_64 1/1

Verifying : 2:docker-1.13.1-74.git6e3bb8e.el7.centos.x86\_64 1/1

Installed:

docker.x86\_64 2:1.13.1-74.git6e3bb8e.el7.centos

Complete!

### 禁用SELinux

> 修改/etc/selinux/config 文件

> 将SELINUX=enforcing改为SELINUX=disabled

[root@localhost docker]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - No SELinux policy is loaded.

SELINUX=disabled

# SELINUXTYPE= can take one of three two values:

# targeted - Targeted processes are protected,

# minimum - Modification of targeted policy. Only selected processes are protected.

# mls - Multi Level Security protection.

SELINUXTYPE=targeted

[root@localhost docker]# reboot

通过reboot命令重启电脑即可。

### 修改docker国内源

外网配置docker国内源，默认国外的下载太慢。

[root@localhost docker]# vim /etc/docker/daemon.json

配置内容如下：

{

"registry-mirrors": ["https://registry.docker-cn.com"]

}

### 启动docker并拉取nexus镜像

[root@localhost /]# systemctl restart docker 重启docker服务

[root@localhost /]# docker pull sonatype/nexus3 拉取nexus3

Using default tag: latest

Trying to pull repository docker.io/sonatype/nexus3 ...

latest: Pulling from docker.io/sonatype/nexus3

Digest: sha256:19d186d5bc8be1ea4f7bae72756baa830e79bf20aae0e9e7b1a0c7d3ce7ac136

Status: Image is up to date for docker.io/sonatype/nexus3:latest

由于之前已经下载过，因此Status为Imaeg is up to date for docker.io/sonatype/nexus3:latest。表示已经下载过该镜像

### 保存镜像

首先查看一下本地的镜像，使用docker images命令。

对于没有私有仓库来说，将本地镜像放到其它服务器上执行时，我们可以使用save和load方法，前者用来把镜像保存一个tar文件，后台从一个tar文件恢复成一个镜像，这个功能对于我们开发者来说还是很方便的！

[root@localhost tmp]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/sonatype/nexus3 latest 777b20c20405 2 weeks ago 505 MB

[root@localhost tmp]# docker save -o nexus\_iso.tar sonatype/nexus3

[root@localhost tmp]# ls

abrt

docker

nexus\_iso.tar

可以看到nexus\_iso.tar压缩包已经生成了。之后则可以在内网使用docker load命令把压缩包转换成镜像。

### 创建数据目录，关联启动nexus镜像

首先创建数据目录

[root@localhost nexus-data]mkdir –p /some/dir/nexus-data

[root@localhost nexus-data]# pwd

/some/dir/nexus-data

[root@localhost nexus-data] chown –R 200

使用下载的镜像通过docker命令创建容器

[root@localhost tmp]# docker run -d -p 8081:8081 --name nexus -v /some/dir/nexus-data:/nexus-data sonatype/nexus3

/usr/bin/docker-current: Error response from daemon: Conflict. The container name "/nexus" is already in use by container b24e7dcd1aa99a3ef8c6c25860de5b0e6022a32fda31e54789161f477e08f756. You have to remove (or rename) that container to be able to reuse that name..

See '/usr/bin/docker-current run --help'.

[root@localhost tmp]# docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

b24e7dcd1aa9 sonatype/nexus3 "sh -c ${SONATYPE\_..." 17 hours ago Exited (255) 49 minutes ago 0.0.0.0:8081->8081/tcp nexus

[root@localhost tmp]# docker start b24e7dcd1aa9

b24e7dcd1aa9

[root@localhost tmp]#

此时既可以通过网页进行访问nexus，即私服网页

访问网址虚拟机IP:8081,由于本次安装的虚拟机IP为192.168.0.165，如下访问：

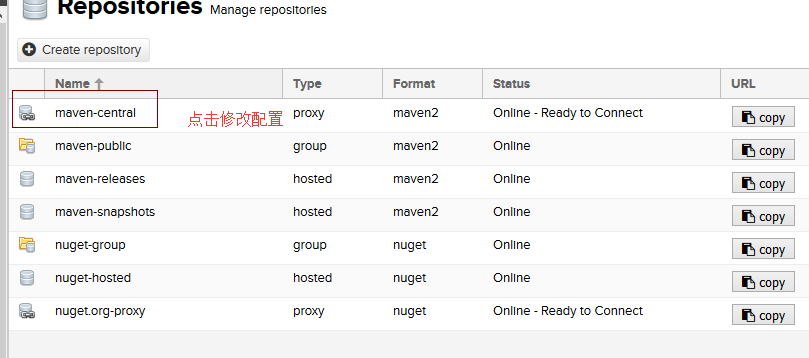
192.168.0.165:8081



### 修改中央仓库代理配置

登陆网页，用户名为admin，密码为admin123

点击settings->Repository->Repositories，弹出如下界面：



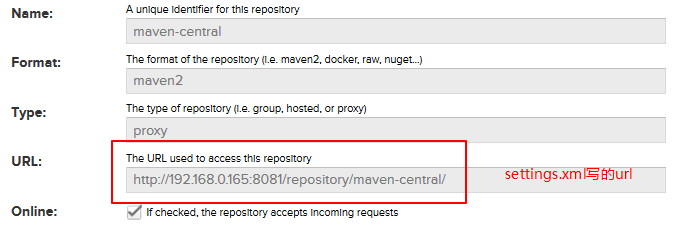
点击maven-central修改：

修改Remote storage:

http://maven.aliyun.com/nexus/content/groups/public



然后保存访问私服的url即可



## IDEA触发下载

打开IDEA配置自定义的settings.xml，settings.xml内容如下：

<?xml version="1.0" encoding="UTF-8"?>

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">

<localRepository>C:\apache-maven-3.3.9\conf\repository</localRepository>

<mirrors>

<mirror>

<id>my mavenn</id>

<name>my maven</name>

<url>http://192.168.0.165:8081/repository/maven-central/</url>

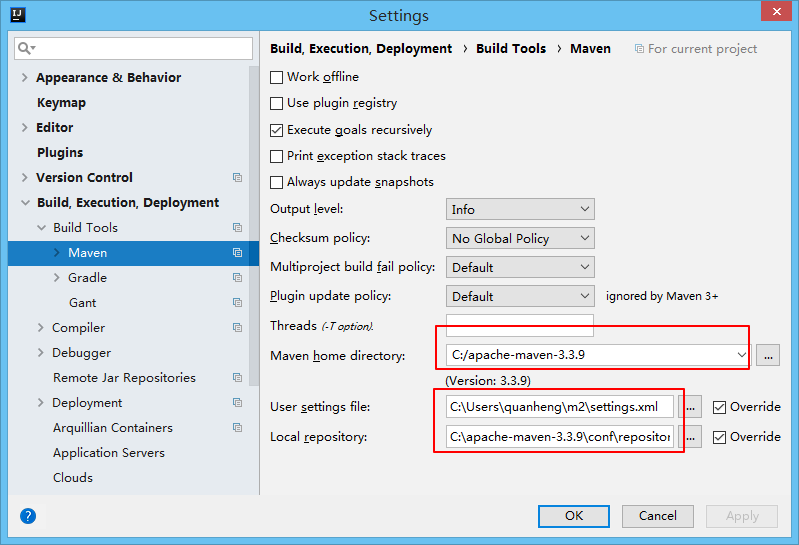
<mirrorOf>central</mirrorOf>

</mirror>

</mirrors>

</settings>

在IDEA中配置自定义的settings.xml文件。



## Import Changes

在pom.xml文件上，右击maven->Reimport,即可看到正在私服下载相关依赖，如果私服不存在，即会从阿里云上下载到私服。



此时可以看到很多标红的，是因为自己把本地仓库删掉，IDEA找不到相应的库。使用命令Import Changes触发下载依赖

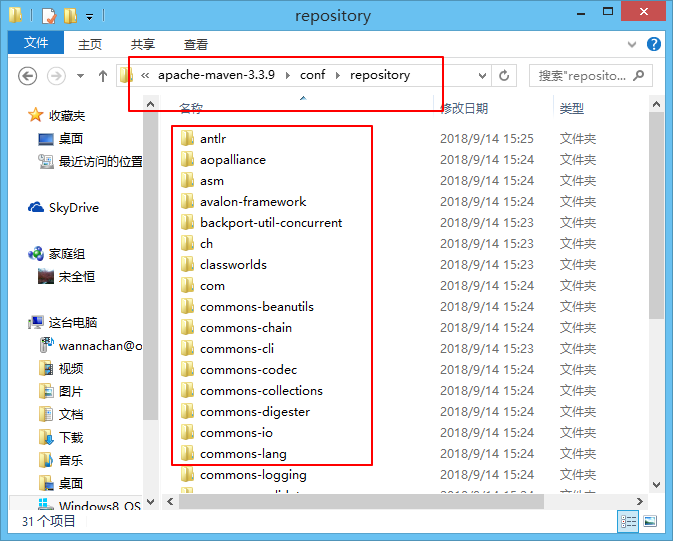
待下载完成，则会变成黑色。

## 验证

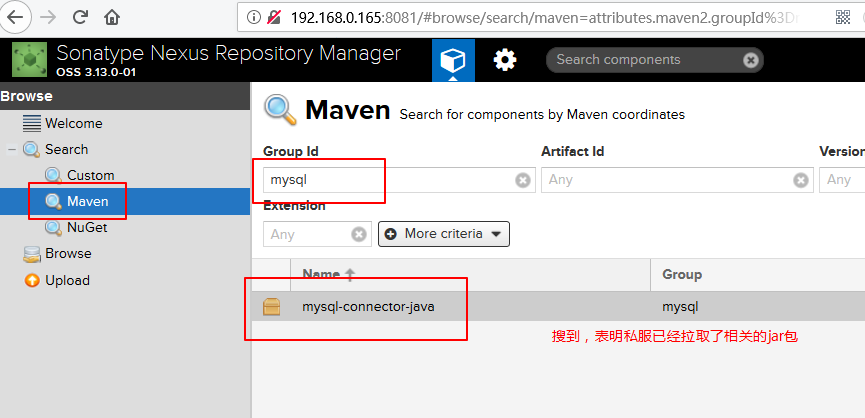
之前自己删除了本地仓库

C:\apache-maven-3.3.9\conf\repository

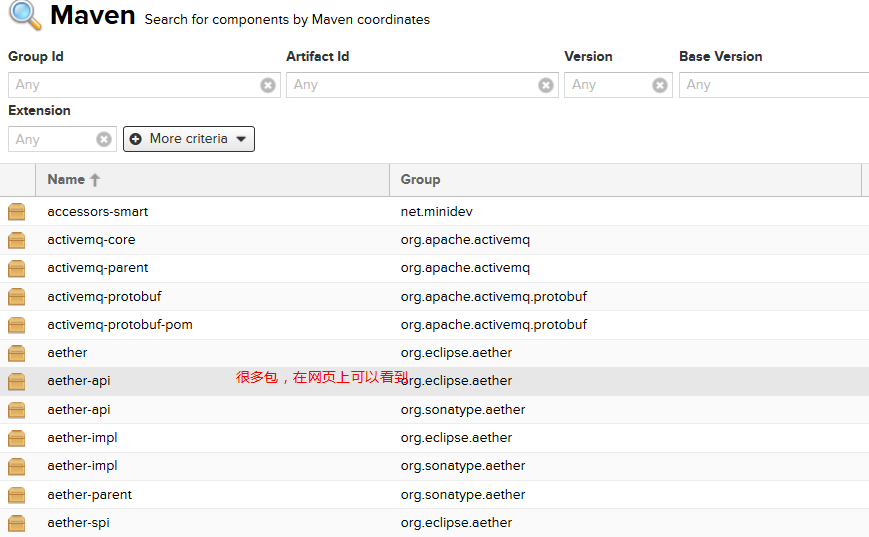
待下载完成之后可以看到：



浏览私服网址



至此，就已经在私服下载了一些包



此时便可以把私服挪进内网了。还要进行如下的步骤：

## 打包

由于目录/some/dir/nexus-data目录为由镜像nexus所起的容器的关联容器，因此，只要把该数据目录打包，拷贝进入内网。

首先stop容器

[root@localhost nexus-data]# docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

b24e7dcd1aa9 sonatype/nexus3 "sh -c ${SONATYPE\_..." 18 hours ago Up 50 minutes 0.0.0.0:8081->8081/tcp nexus

[root@localhost nexus-data]# docker stop b24e7dcd1aa9

b24e7dcd1aa9

然后执行tar命令把数据目录备份

[root@localhost nexus-data]# pwd

/some/dir/nexus-data

[root@localhost nexus-data]# tar cvf nexus.tar /some

/some/dir/nexus-data/cache/bundle96/bundle.info

/some/dir/nexus-data/cache/bundle96/version0.0/

/some/dir/nexus-data/cache/bundle96/version0.0/bundle.jar

/some/dir/nexus-data/cache/bundle96/version0.0/revision.location

/some/dir/nexus-data/cache/bundle97/

/some/dir/nexus-data/cache/bundle97/bundle.info

/some/dir/nexus-data/cache/bundle97/version0.0/

/some/dir/nexus-data/cache/bundle97/version0.0/bundle.jar

/some/dir/nexus-data/cache/bundle97/version0.0/revision.location

/some/dir/nexus-data/cache/bundle98/

/some/dir/nexus-data/cache/bundle98/bundle.info

/some/dir/nexus-data/cache/bundle98/version0.0/

然后把所有的文件拷贝进入内网，拷贝的材料如下：

1， 下载不安装的docker

2， 打包的/some/dir/nexus-data目录nexus.tar

3， docker命令导出的镜像tar包

## 内网操作

在相同Centos版本比如在开发机安装的Centos版本如下：

[root@localhost docker]# cat /etc/redhat-release

CentOS Linux release 7.4.1708 (Core)

首先要在内网安装docker,cd到rpm所在目录

### 安装docker

执行命令

[root@localhost tmp]# rpm -Uvh \*

warning: audit-2.8.1-3.el7.x86\_64.rpm: Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEY

preparing… ############################### [100%]

Updating / installing…

1:libsepol-2.5-8.1.el7 ############################## [100%]

2:libselinux-2.5-l2.el7 ############################## [100%]

…

3:docker-2:1.13.1-63.git94f4240.el7 ################################### [100%]

使用docker load –i nexus\_iso.tar，装载镜像。

[root@localhost ~]# docker load -i nexus.tar

Cannot connect to the Docker daemon. Is the docker daemon running on this host?

该问题遇到，请重启docker服务

[root@localhost ~]# service docker restart

Redirecting to /bin/systemctl restart docker.service

[root@localhost ~]# docker load -i nexus\_iso.tar

1d31b5806ba4: Loading layer [=======================>] 208.3MB/208.3 MB

147340bc3324: Loading layer [=======================>] 5.12 kB/5.12kB

147340bc3324: Loading layer [=======================>] 306.6 MB/306.6B

Loaded image: docker.io/sonatype/nexus3:latest

### 解压数据目录

操作如下：

tar xvf nexus.tar

mv ./some / # 拷贝到根目录下生成目录/some/dir/nexus-data/

chown -R 200 /some/dir/nexus-data/

### 运行镜像

[root@localhost ~]# doker run –d –p 8081:8081 --name -v /some/dir/nexus-data/:/nexus-data sonatype/nexus3

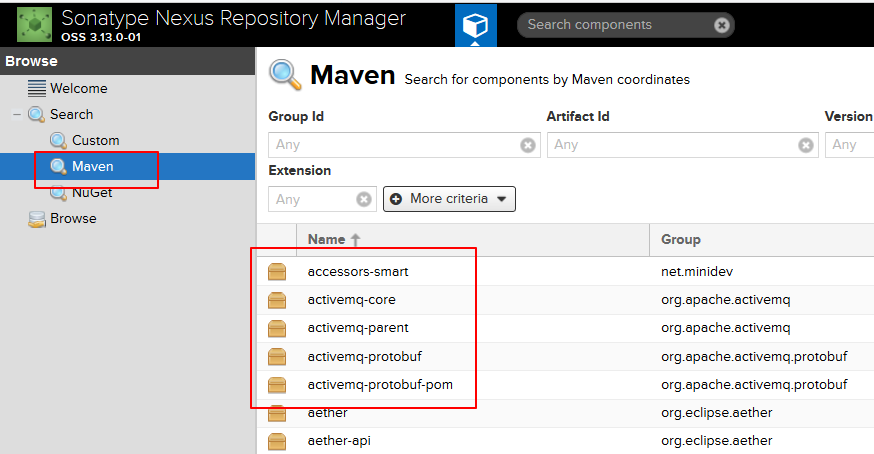
c8ffadffad126739783ff83733rj3fhfjhajfdafh8238

### 网页验证

使用内网虚拟机IP登陆sonatype nexus Repository

192.168.95.101:8081

登录名和密码依然为admin和admin123



这表示内部在内网已经运行成功了。

### docker容器管理

使用docker ps -a，可以看到由镜像模板装载的容器列表。

[root@localhost ~]# docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

50336317ab6f sonatype/nexus3 "sh -c ${SONATYPE\_..." 2 days ago Exited (137) 20 minutes ago

可以使用docker restart/start/stop 50336317ab6f对该容器进行重启，停止，重新启动。

### docker镜像管理

下载镜像

[root@kvm ~]# docker pull centos

latest: Pulling from centos

47d44cb6f252: Already exists

6fdebd7b0eb5: Already exists

a63aae4d216f: Already exists

Digest: sha256:381f21e4c7b3724c6f420b2bcfa6e13e47ed155192869a2a04fa10f944c78476

Status: Image is up to date for centos:latest

查看下载到本地的镜像

[root@kvm ~]# docker images

REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE

centos              latest              bb3d629a7cbc        3 weeks ago         196.6 MB

删除镜像

[root@kvm ~]# docker rmi centos:test-img2

Untagged: centos:test-img2

[root@kvm ~]# docker images

REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE

centos              latest              bb3d629a7cbc        3 weeks ago         196.6 MB

test-img            latest              bb3d629a7cbc        3 weeks ago         196.6 MB

# 再次导入

再经过一段时间的开发之后，或许在外网又重新下载了许多依赖包。需要重新再次导入内网。操作如下：

将外网的数据拷贝重新拷贝进入内网

1. 外网docker stop 容器id
2. tar命令对目录/some/dir/nexus-data目录压缩
3. 把数据目录压缩包拷贝进入内网。
4. 内网根据容器id停止运行
5. 解压缩，并修改权限chown -R 200 /some/dir/nexus-data
6. 使用命令docker run重新拉起容器运行  
   docker run -d -p 8081:8081 --name nexus -v /some/dir/nexus-data:/nexus-data sonatype/nexus3

# 总结

至此，内网搭建maven私服的过程总结完毕。整个过程并不复杂。对jar包依赖的解决，maven私服的确是一个不错的方式。