什么是镜像库

字面意思，镜像库就是集中存放镜像的一个文件服务。镜像库在 CI/CD 中，又称 制品库 。构建后的产物称为**制品**，制品则要放到**制品库**做**中转和版本管理**。常用平台有**Nexus，Jfrog，Harbor**或其他对象存储平台。

在这里，我们选用 Nexus3 作为自己的镜像库。因为其稳定，性能好，免费，部署方便，且支持类型多，是许多制品库的首选选型。

部署 Nexus 服务

在部署 Nexus 之前，需要先下载 Nexus 的安装包（这里需要另外找个托管服务）

shell 复制代码

```
wget https://dependency-fe.oss-cn-beijing.aliyuncs.com/nexus-3.29.0-02-unix.tar.gz
```

下载完成后，解压安装包

shell 复制代码

```
tar -zxvf ./nexus-3.29.0-02-unix.tar.gz
```

解压后，我们可以看到有2个文件夹。分别是 nexus-3.29.0-02 和 sonatype-work 。其中， nexus-3.29.0-02 是nexus主程序文件夹， sonatype-work 则是数据文件。

启动 Nexus

我们进入 nexus-3.29.0-02 下面的 bin 目录，这里就是 nexus 的主命令目录。我们在 bin 目录下，执行 ./nexus start 命令即可启动 nexus ：

shell 复制代码

```
./nexus start
```

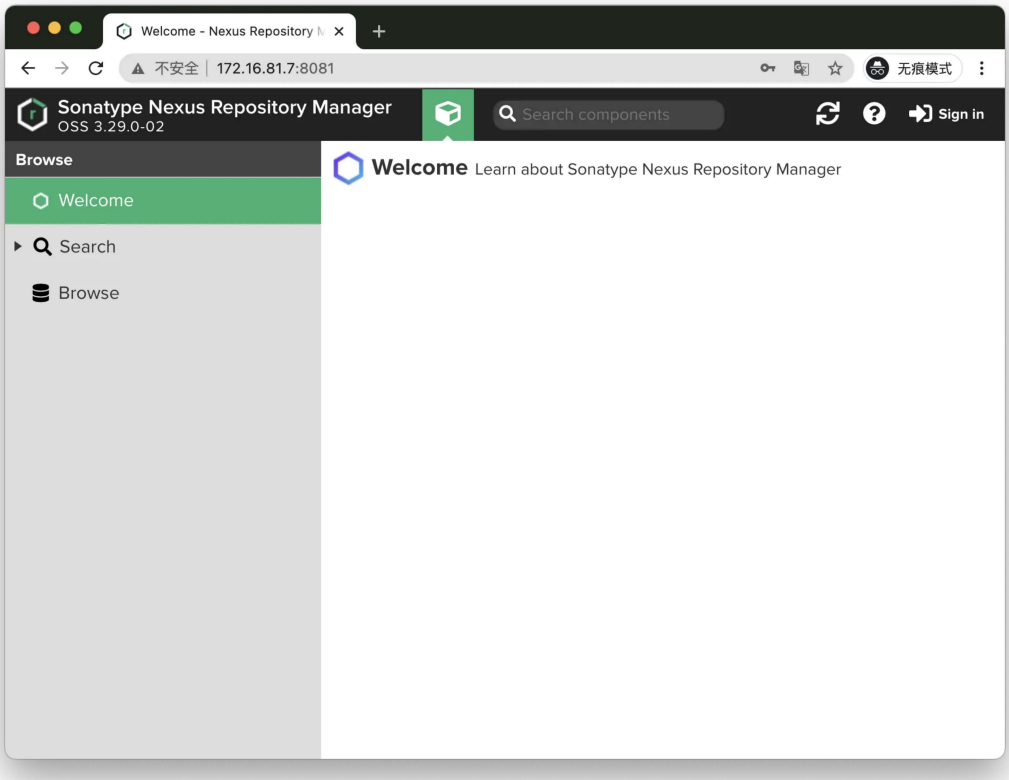
itjc8.com搜集整理

由于 `nexus` 默认服务端口是 `8081` ，稍后我们还需要给镜像库访问单独开放一个 `8082` 端口。这里将 `8081` ， `8082` 端口添加到防火墙放行规则内（没开防火墙则可以略过）：

shell 复制代码

```shell
firewall-cmd --zone=public --add-port=8081/tcp --permanent
firewall-cmd --zone=public --add-port=8082/tcp --permanent
```

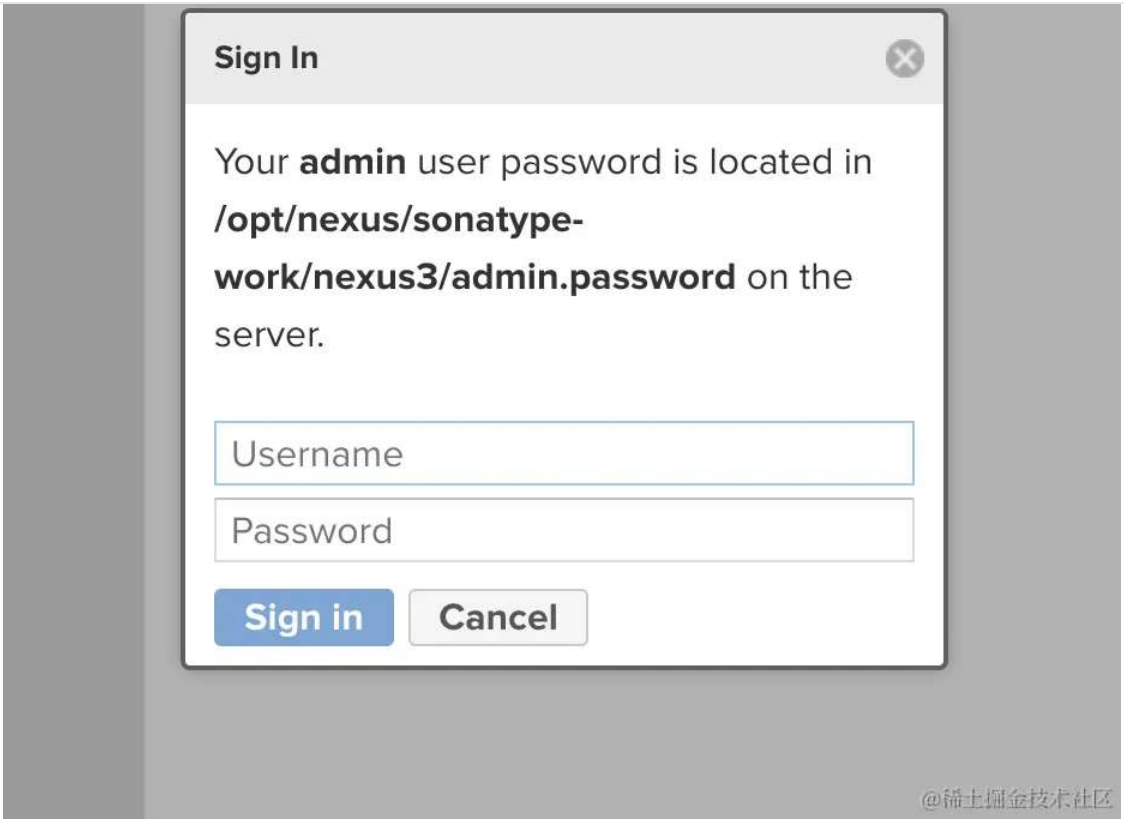
打开浏览器地址栏，访问 `IP:8081` 。启动时间比较长，需要耐心等待。在 `Nexus` 启动后，会进入这个欢迎页面：



配置 Nexus

进入欢迎页后，点击右上角的登录，会打开登录框。这里需要我们输入 默认管理员密码 进行初始化配置。

itjc8.com搜集整理



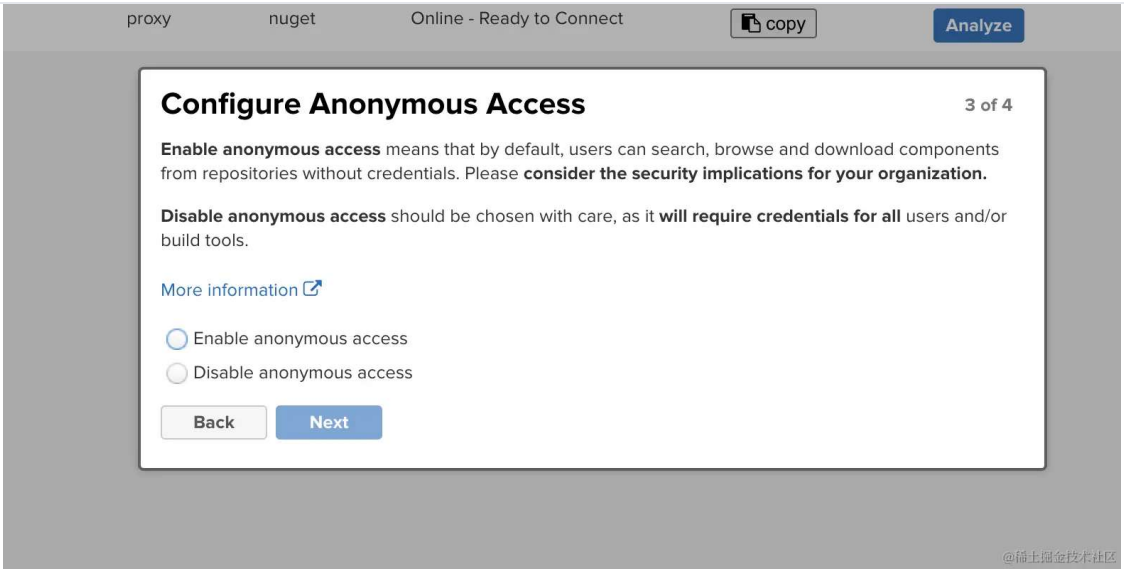
可以在这里找到：

```shell
cat /opt/nexus/sonatype-work/nexus3/admin.password
# 0ee35fa5-d773-432b-8e76-6c10c940ccd9
```

将文件中获取到的密码输入进去，登录用户名是 `admin` 。

接着是修改新密码。修改后，会进入下图这一步。这一步的意思是**是否开启匿名访问。**匿名访问是指：**我们在没有登录的情况下，拉取（推送）制品到制品库，都算匿名访

itjc8.com搜集整理

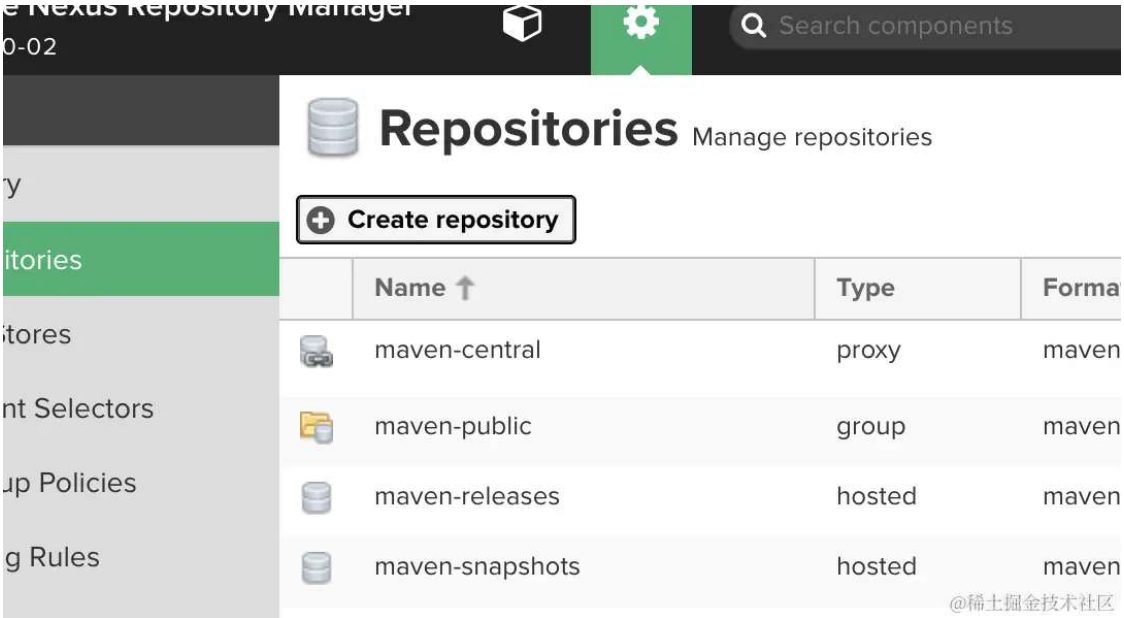


例如，这个制品库也支持 `node` 的 `npm` 私有库。那么我们在没有 `npm login` 登录这个制品库之前，就可以进行 `npm install` `npm publish`，其实是不太安全的。那么任何一个知道制品库地址的人，都可以任意进行推送和获取资源。

这里我们为了测试，可以先允许开启匿名访问。选择 `Enable anonymous access` ，点击下一步。

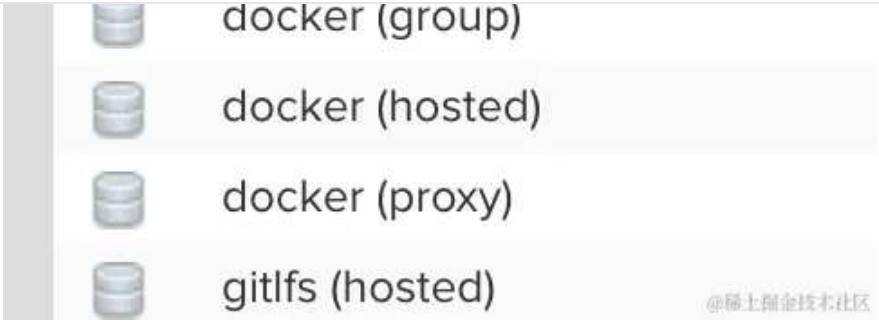
创建一个 Docker 私服

登录完成后，点击页面头部导航栏的**齿轮**图标，选择左侧菜单中的 `Repositories` ，点击 `Create repository` 。



点击后，我们可以看到一个列表，这就是 `Nexus` 所支持的制品库类型。其中有我们要使用的 `Docker` ，也有我们熟悉的 `Npm` 。我们在里面找到 `Docker` ：

itjc8.com搜集整理



但是 `Docker` 有三种，该选哪个呢?

选择制品库的类型

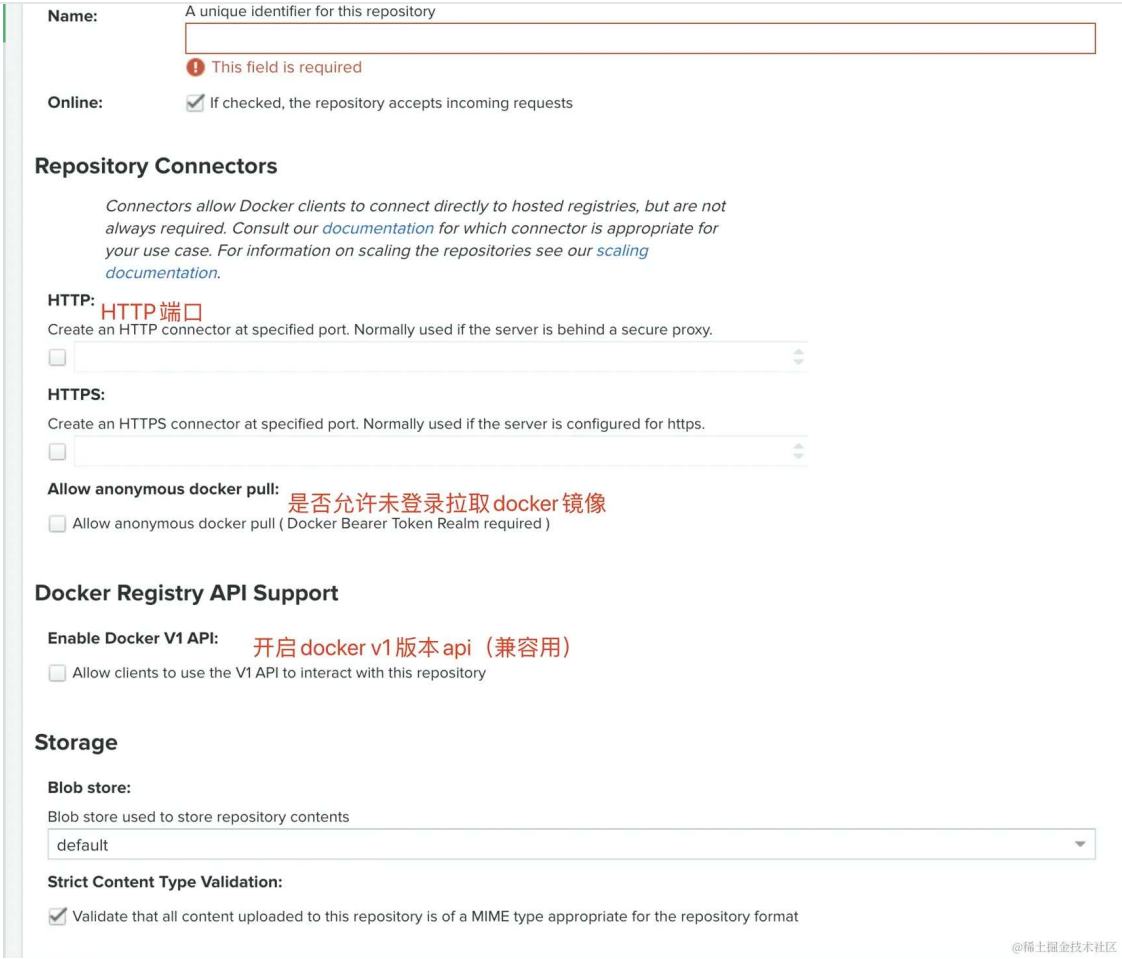
在 `nexus` 中，制品库一般分为以下三种类型：

- proxy: 此类型制品库原则上**只下载，不允许用户推送**。可以理解为**缓存外网制品的制品库**。例如，我们在拉取 `nginx` 镜像时，如果通过 `proxy` 类型的制品库，则它会去创建时配置好的外网 `docker` 镜像源拉取（有点像 `cnpm` ）到自己的制品库，然后给你。第二次拉取，则不会从外网下载。起到 `内网缓存` 的作用。
- hosted：此类型制品库和 `proxy` 相反，原则上 `只允许用户推送，不允许缓存` 。这里只存放自己的私有镜像或制品。
- group：此类型制品库可以将以上两种类型的制品库组合起来。组合后只访问 `group` 类型制品库，就都可以访问。

在这里，我们其实不需要**缓存外网镜像**，那么我们只需要 `hosted` 即可。选择 `docker (hosted)` 。

我们将启动 `Nexus` 镜像时，配置好的 `Docker` 端口（预留了一个 `8082` 端口）填入 `HTTP` 输入框内。这里可以先允许匿名拉取镜像。

itjc8.com搜集整理



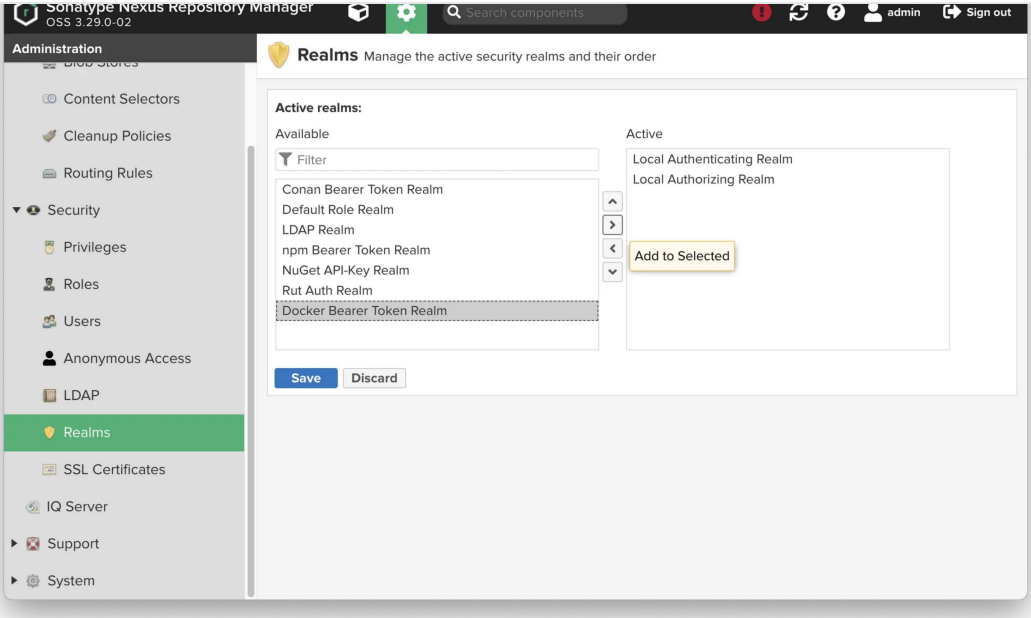
填写完成后，点击最下方的 **Create repository，**保存创建。

给镜像库添加访问权限

在我们创建好镜像库后，还需要配置一步访问权限才可以。

找到页面头部导航栏的 齿轮 图标，选择左侧菜单中的 `Realms` 。找到右边的 `Docker Bearer Token Realm` ，将其添加到右边的 `Active` 内，保存即可。

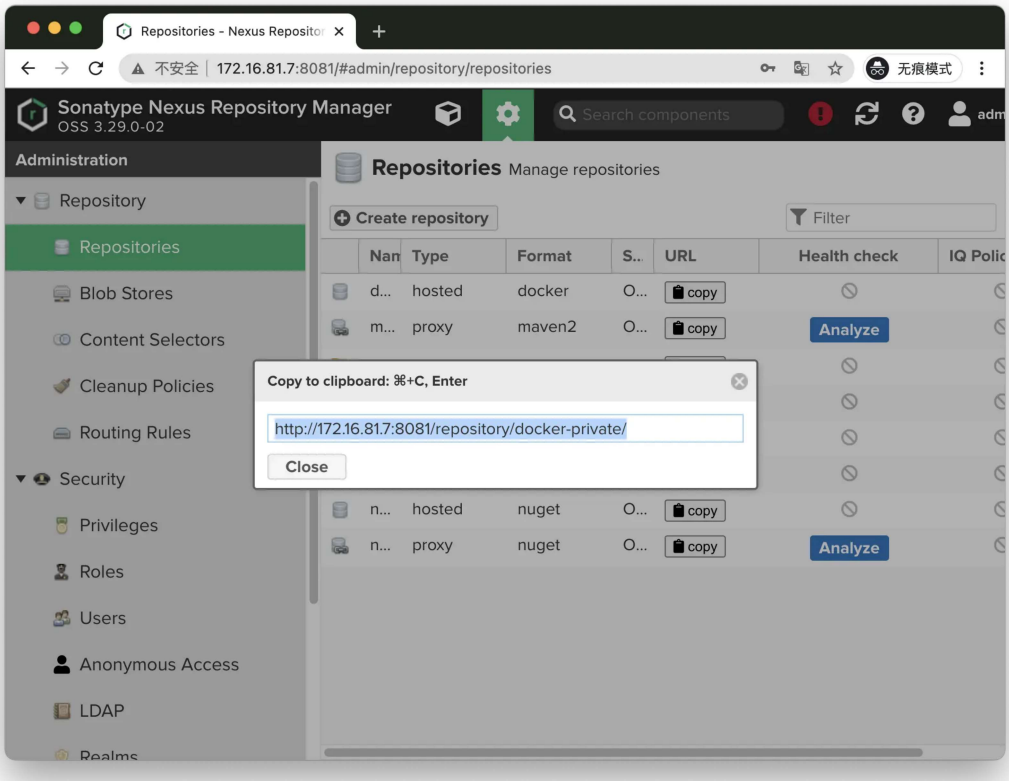
itjc8.com搜集整理



@稀土掘金技术社区

查看获取镜像库地址

找到我们刚刚创建的制品，点击上面的 `copy` ，查看镜像库地址。



@稀土掘金技术社区

登录制品库

itjc8.com搜集整理

找到 `daemon.json` 文件，该文件描述了当前 `docker` 配置的镜像加速地址，和配置过的私服地址。

shell 复制代码

```
vi /etc/docker/daemon.json
```

找到 `insecure-registries` 字段，如果不存在就自己添加一个。值是数组类型，将你的制品库地址填写上去。例如：

json 复制代码

```
{
  "insecure-registries" : [
    "172.16.81.7:8082"
  ],
}
```

> 注意，nexus 显示的镜像库端口为 nexus 服务端口，要替换为自己配置的端口才有效。

保存并退出，重启 Docker

shell 复制代码

```
systemctl restart docker
```

接着使用 `docker login` 命令尝试登录：

shell 复制代码

```
docker login 服务IP:端口
```

如果提示：**Login Succeeded 则代表登录成功。**

```
[root@localhost nexus]# docker login 172.16.81.7:8082
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

@稀土掘金技术社区

itjc8.com搜集整理

在完成镜像库配置后，我们就可以使用 Jenkins 推送自己的镜像到镜像库了。我们找到 Jenkins 任务中设置 Shell 的编辑框，添加一条推送镜像的命令进去：

注意！

`docker` 在推送一个镜像时，**镜像的 Tag (名称:版本号) 开头必须带着镜像库的地址，才可以推送到指定的镜像库。**例如 `jenkins-test` 是不能推送到镜像库的。而 `172.16.81.7:8082/jenkins-test` 则可以推送到镜像库。

那我们怎么才能推送镜像上去呢？我们可以重新制作一份带镜像库地址的镜像。找到 Jenkins 的 Shell 编辑框，j将构建的 Shell 脚本修改为以下内容：

shell 复制代码

```shell
#!/bin/sh -l

npm install --registry=https://registry.npm.taobao.org
npm run build
docker build -t 172.16.81.7:8082/jenkins-test .
docker push 172.16.81.7:8082/jenkins-test
```

这里将**构建的镜像名称加了镜像库的前缀**，推送镜像也是一样，这样才可以将镜像推送到指定镜像库。保存后并重新构建一次。

itjc8.com搜集整理

```
71580c30d988: Preparing
a521e1bbddf5: Preparing
bf381a670956: Preparing
a61993362baf: Preparing
f1b5933fe4b5: Preparing
unauthorized: access to the requested resource is not authorized
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

@稀土掘金技术社区

利用凭据给 Shell 注入镜像库用户名密码

没有权限怎么办呢？我们可以使用 `docker login` 在 `shell` 脚本里面登录下。想直接在命令里写入用户名和密码，可以直接加 `-u 用户名 -p 密码` 即可。例如：

shell 复制代码

```shell
docker login -u "用户名" -p "密码" 172.16.81.7:8082
```

但这样，我们需要在命令里面写死用户名和密码，无论是安全和友好性上，都是不太合适的。这里我们可以借助 Jenkins 的凭据功能，添加一条用户名密码凭据，然后利用 Shell 变量写入在终端内。

找到任务的设置界面 => 构建环境 => 勾选 Use secret text(s) or file(s) => 找到左下角的新增按钮，选择 `Username and password (separated)`

itjc8.com搜集整理

打开后，我们可以添加一条凭据。点击凭据字段下面的添加，弹出以下弹窗，在这里填入你的用户名和密码。ID为凭据名称，描述随意。

添加后，返回下图模块。在这里选择你刚才添加的凭据，用户名变量可以起名为 `DOCKER_LOGIN_USERNAME` ，密码可以起名为 `DOCKER_LOGIN_PASSWORD` 。

itjc8.com搜集整理

接着找到下面的构建，找到 `docker login` 命令，将我们保存的用户名和密码变量填写进去：

shell 复制代码

```
1  docker login -u $DOCKER_LOGIN_USERNAME -p $DOCKER_LOGIN_PASSWORD 172.16.81.7:8082
```

接着保存并构建，提示权限通过，构建成功

```
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
The push refers to repository [172.16.81.7:8082/jenkins-test]
7f580c30d988: Preparing
a521e1bbddf5: Preparing
bf381a670956: Preparing
a61993362baf: Preparing
f1b5933fe4b5: Preparing
a521e1bbddf5: Layer already exists
f1b5933fe4b5: Layer already exists
7f580c30d988: Layer already exists
a61993362baf: Layer already exists
bf381a670956: Layer already exists
latest: digest: sha256:ae081f6633d3e0018a8a6a674f1bad9eb105e97216054e9ac07bc547a8c15c1d size:
1363
Finished: SUCCESS
```

@稀土掘金技术社区

如何推送已有的镜像到仓库呢?

上面是推送我们现场编译的镜像，镜像名称都可以一条龙约定好。可是面对 load /pull 进来的镜像，我们如何推送到自己的镜像库呢?

这里可以使用 `docker tag` 命令给已有的镜像打个标签。在打新Tag时可以在Tag头部加入镜像库地址。如下面格式。

shell 复制代码

itjc8.com搜集整理

> 查看服务器上的docker镜像列表，可以使用 docker images 查看

这样，就可以重新打一个全新的tag，实现 重命名 功能。

```
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
172.16.81.150:8082/local/jenkins    latest              bd695e3e4317        12 hours ago        677MB
local/jenkins                       latest              bd695e3e4317        12 hours ago        677MB
```

@稀土掘金技术社区

接着我们使用 docker push 命令就可以进行推送了：

shell 复制代码

```
1  docker push 172.16.81.150:8082/local/jenkins
```

```
[root@localhost ~]# docker push 172.16.81.150:8082/local/jenkins
The push refers to repository [172.16.81.150:8082/local/jenkins]
6a6be042b357: Pushed
3a070a13db42: Pushed
8926f01eca4b: Pushed
a6b8f7267bb3: Pushed
3cb82385d5ce: Pushed
0cd936bef077: Pushed
ed138baf5b94: Pushed
51485080e8fd: Pushed
0a865f3f860e: Pushed
d7a8d7b4a3ee: Pushed
0030a2aac18e: Pushed
ca8ff06c1037: Pushed
b56c581a374c: Pushed
802845173f8c: Pushed
832986c77b0e: Pushed
d81d8fa6dfd4: Pushed
bd76253da83a: Pushed
e43c0c41b833: Pushed
01727b1a72df: Pushed
69dfa7bd7a92: Pushed
4d1ab3827f6b: Pushed
7948c3e5790c: Pushed
latest: digest: sha256:63a80d866e7069cef67ea2f17e9d0f7e8f7de2aeeeb76e2364813f383c20e4d7 size: 4925
```

@稀土掘金技术社区

‹ 上一章　　下一章 ›

留言

输入评论（Enter换行，Ctrl + Enter发送）

发表评论

itjc8.com搜集整理