**1、下载镜像准备**

1、下载nexus3镜像

docker pull sonatype/nexus3

2、创建文件夹

mkdir /opt/nexus-data

3、文件夹授权

chmod 777 /opt/nexus-data

4、启动镜像

docker run -d -p 8081:8081 -p 8082:8082 -p 8083:8083 -p 8084:8084 -v /opt/nexus-data:/nexus-data --name nexus3 sonatype/nexus3

映射端口对应的用途：

8081：nexus3网页端

8082：docker(hosted)私有仓库，可以pull和push

8083：docker(proxy)代理远程仓库，只能pull

8084：docker(group)私有仓库和代理的组，只能pull

使用参数 -v 建立宿主机与Docker目录映射关系，/nexus-data：docker里存nexus数据目录，所以将数据目录存放到宿主机/opt/nexus-data

**2、登录准备**

登录网址输入账号密码 ：默认账号是admin

密码去下面文件去找：/opt/nexus-data/admin.password

计算机生成了可选文字:
Sonetype Nexus Repository Manager 
OSS 3 ． 25 · 0 ． 03 
0 Welcome 工 s ” № ” № “ 
0 Welcome 
， Seerch 
Browse 
Slgn In 
、 Our edmln user 0 ， w01 丨 00 ed In 
'nexus-dete/edmln.pessword on 1 ， e server. 
Cencel 
O 
劌 Sign in 

进入后点击next

计算机生成了可选文字:
Sonetype Nexus Repository Manager 
OSS 3 ． 25 · 0 ． 03 
0 Welcome 工 s ” № ” № “ 
0 Welcome 
， Search 
Browse 
I-I ， 0 匕 
Setu p 
Thls W ， ， 1 WIII ， e 丨 0 you complete requlred et 」 0 tasks. 
admin Sign ～ ' 
user slgned In: edmln 

输入新的密码

计算机生成了可选文字:
Sonetype Nexus Repository Manager 
OSS 3 ． 25 · 0 ． 03 
0 Welcome 工 s ” № ” № “ 
0 Welcome 
， Seerch 
Browse 
I-I ， 0 匕 
Please choose a password for the admin user 
New pessword: 
Confirm password: 
Beck 
2 0 ， 4 
admin 
Sign 

选择禁止匿名访问

计算机生成了可选文字:
Sonetype Nexus Repository Manager 
OSS 3 ． 25 · 0 ． 03 
0 Welcome 工 s ” № ” № “ 
0 Welcome 
， Seerch 
Browse 
I-I ， 0 匕 
Configure Anonymous Access 
3 0 ， 4 
O 
． admin 
Sign 
Enable enonymous eccess means 1 ， by de L 丨 ： users 0 匕 n se 以 ， browse ， nd 00w ， 0 匕 0 componen 蔭 
from reposltorles 纛 ， 10 」 t cre ， entl 引 P 以 e conslder ， h ， securlty Impllcetlons for your orgenlzetlon. 
0 驚 ， b anonymous ， ： C ， S ； 以 ， 0 」 be ， 10 en 纛 1 ， 0 酽 e as It WII' requlre cr ， d ， n ， for 丨 users 以 n ， or 
匕 ul ℃ ℃ 0 且 
More 
C) nable anonymous 00e 
（ 巨 ） D 尽 匕 ， e nonymo 」 以 access 
Beck 

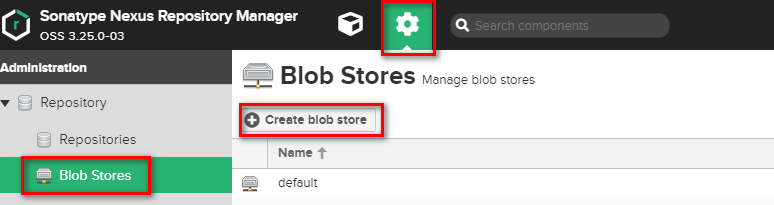
点击finlsh

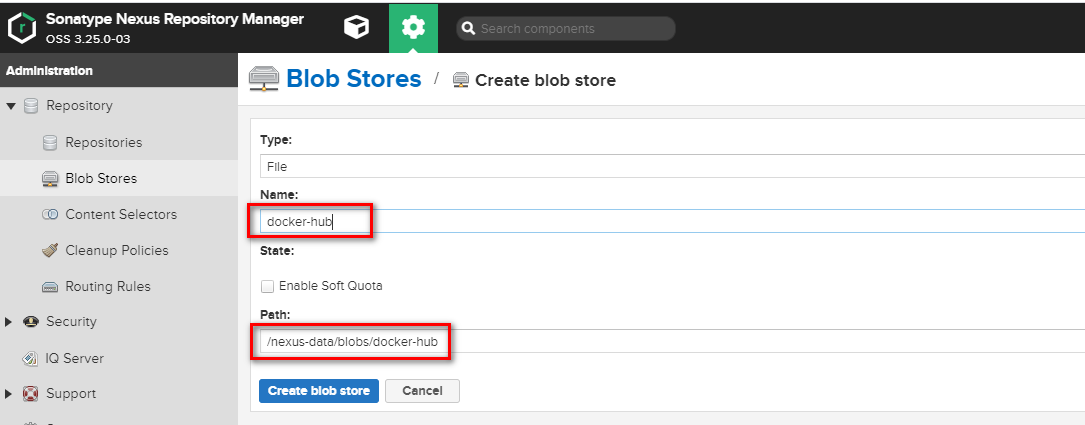
计算机生成了可选文字:
Sonetype Nexus Repository Manager 
OSS 3 ． 25 · 0 ． 03 
0 Welcome 工 s ” № ” № “ 
0 Welcome 
， Search 
Browse 
I-I ， 0 匕 
Com plete 
The setup 1 ， k 以 have been completed, enjoy 」 n § Nexus Repo ℃ ， Manager! 
4 0 ， 4 
admin 
Sign 

**3、配置仓库**

2.1、创建blob存储

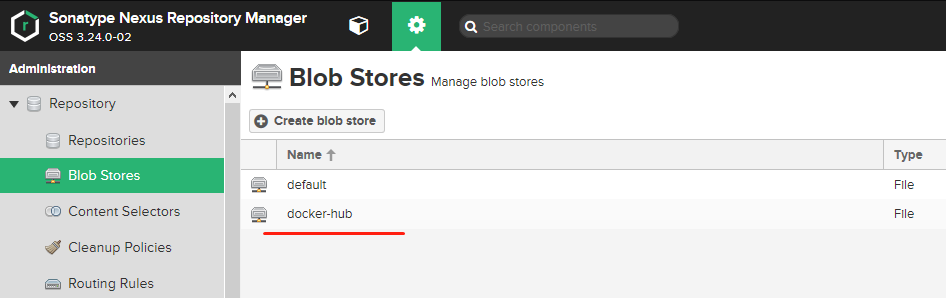
登陆之后，先创建一个用于存储镜像的空间。





定义一个name，下边的内容会自动补全。

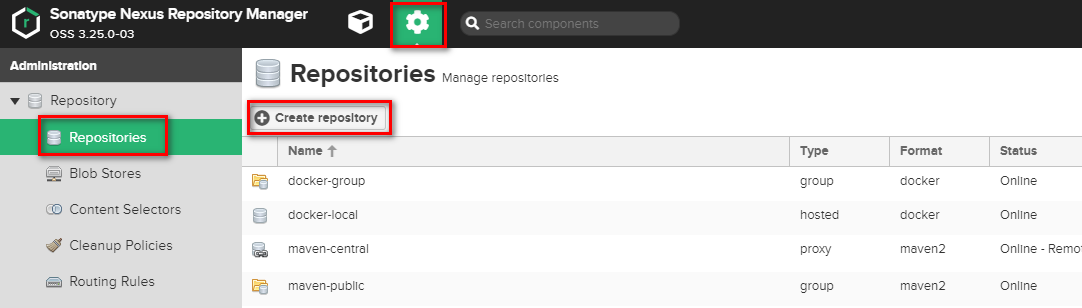
然后保存。



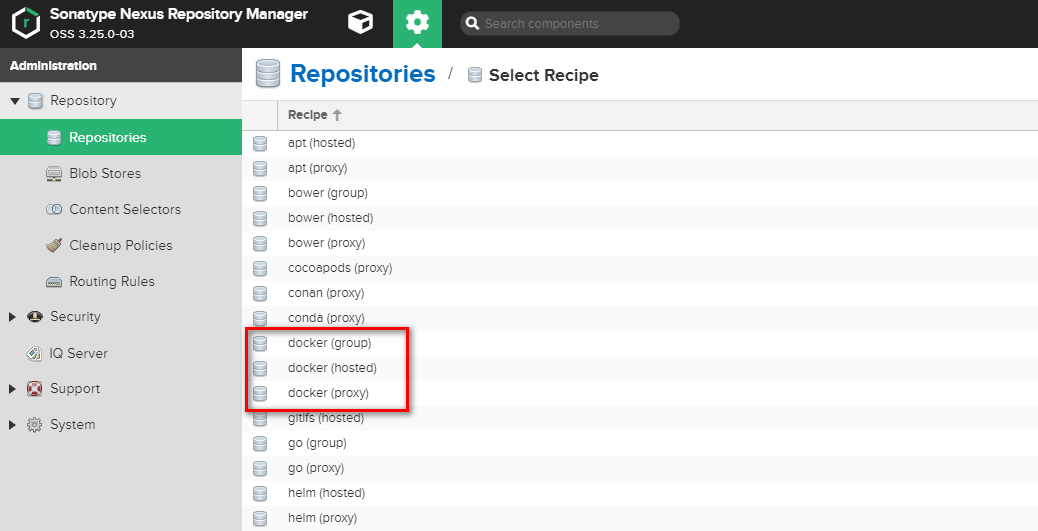
**4、创建一个hosted类型的docker仓库**

Hosted类型仓库用作我们的私有仓库，替代harbor的功能。

点击步骤如下：



这里我们看到docker类型有三种：



* hosted : 本地存储，即同 docker 官方仓库一样提供本地私服功能。
* proxy : 提供代理其他仓库的类型，如 docker 中央仓库。
* group : 组类型，实质作用是组合多个仓库为一个地址。

先来创建一个hosted类型的私有仓库。

点击 Repository下面的 Repositories – Create repository – docker(hosted) :

Name: 定义一个名称docker-local

Online: 勾选。这个开关可以设置这个Docker repo是在线还是离线。

Repository Connectors

把HTTP这里勾选上，然后设置端口为8083。

Allow anonymous docker pull

不勾选。这样的话就不允许匿名访问了，执行docker pull或 docker push之前，都要先登录：docker login

Docker Registry API Support

Docker registry默认使用的是API v2, 但是为了兼容性，我们可以勾选启用API v1。

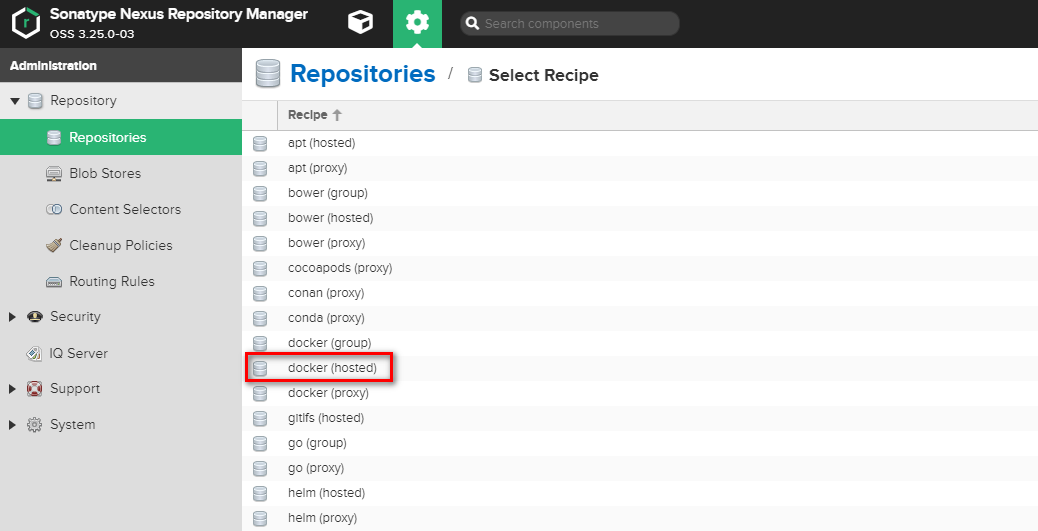
Storage：

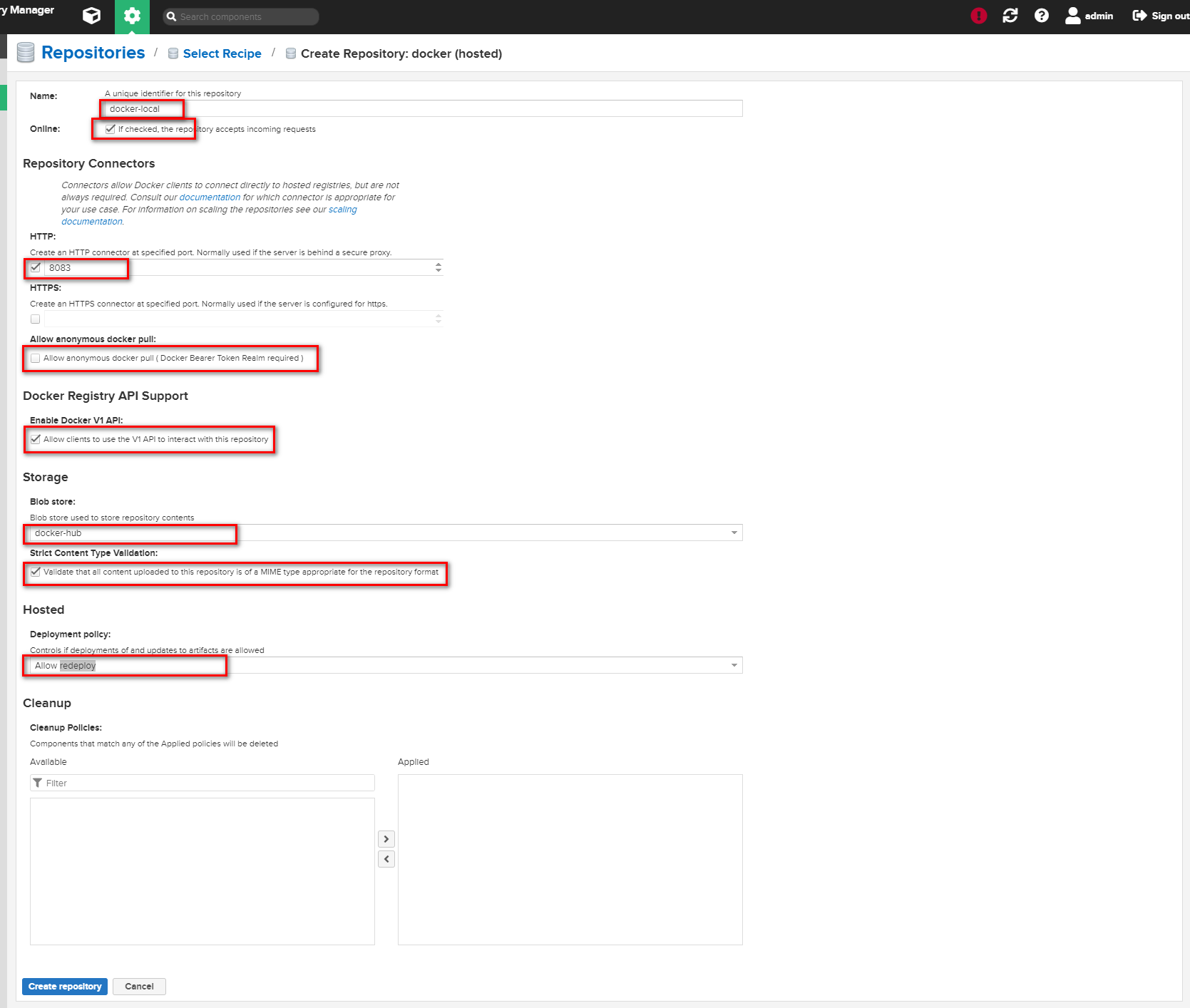
Blob store：我们下拉选择前面创建好的专用blob：docker-hub。

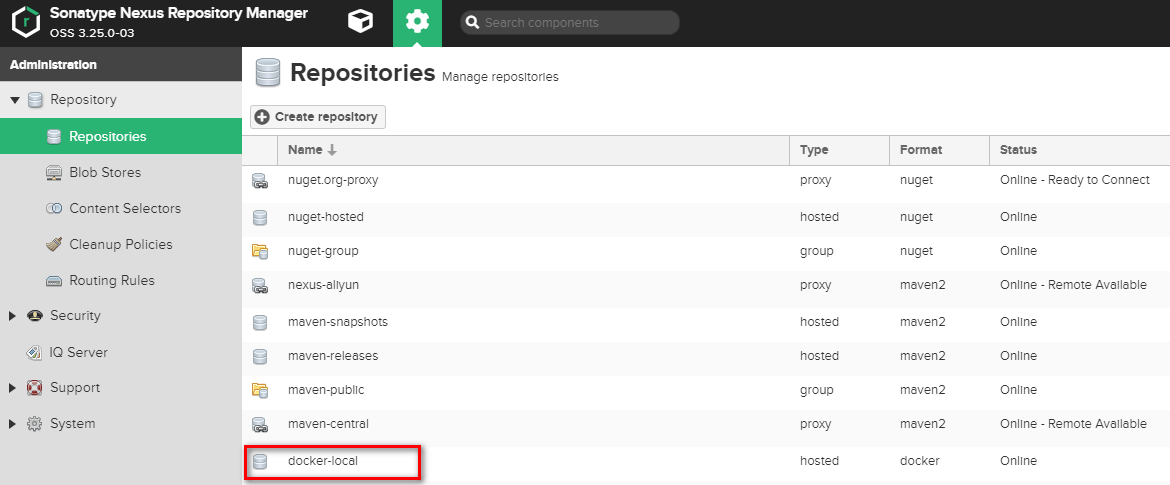
Hosted：

开发环境，我们运行重复发布，因此Delpoyment policy 我们选择Allow redeploy。

创建Hosted ：整体配置截图如下：







**5、创建一个proxy类型的docker仓库**

proxy类型仓库，可以帮助我们访问不能直接到达的网络，如另一个私有仓库，或者国外的公共仓库，如官方的dockerhub镜像库。

创建一个proxy类型的仓库

Name: proxy-docker-hub

Repository Connectors: 不设置。

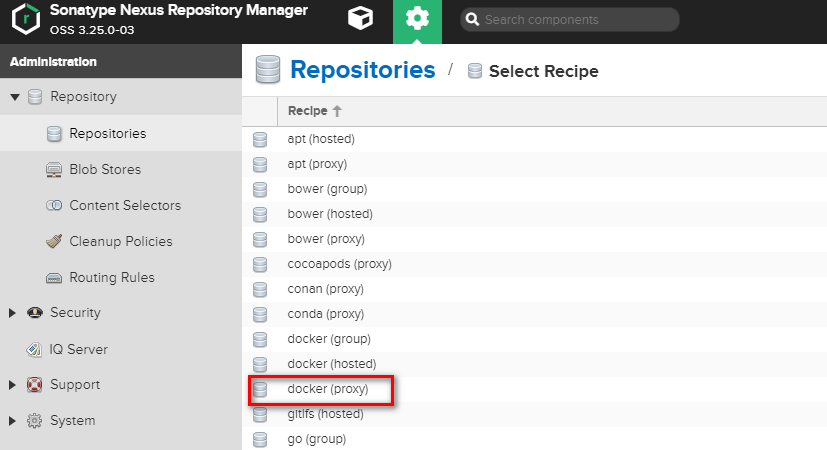
Proxy

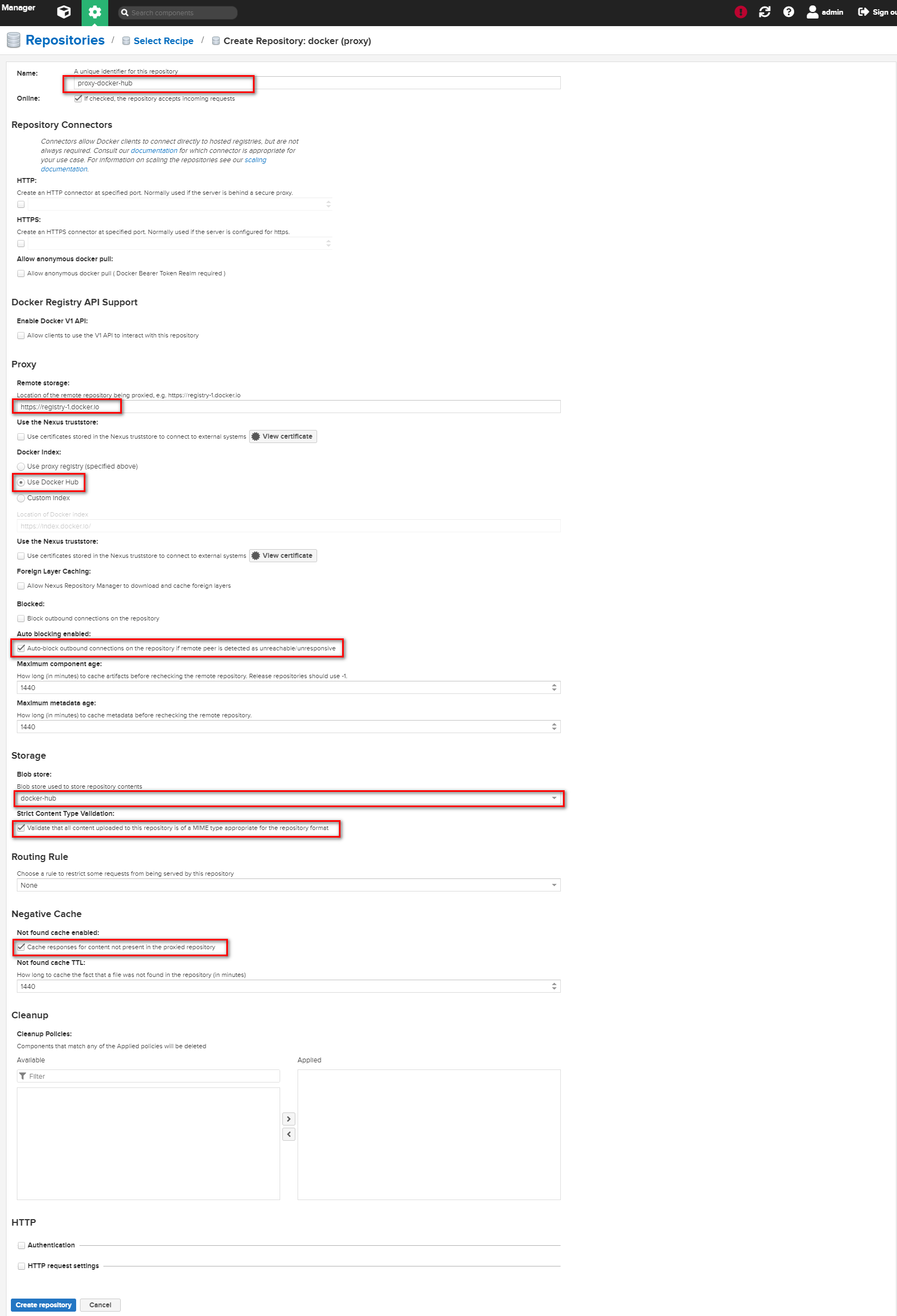
Remote Storage: docker hub的proxy，这里填写: <https://registry-1.docker.io> 这个是官方默认的一个链接

Docker Index： Use Docker Hub

Storage：idocker-hub

整体配置截图如下：





**6、创建一个group类型的docker仓库**

group类型的docker仓库，是一个聚合类型的仓库。它可以将前面我们创建的3个仓库聚合成一个URL对外提供服务，可以屏蔽后端的差异性，实现类似透明代理的功能。

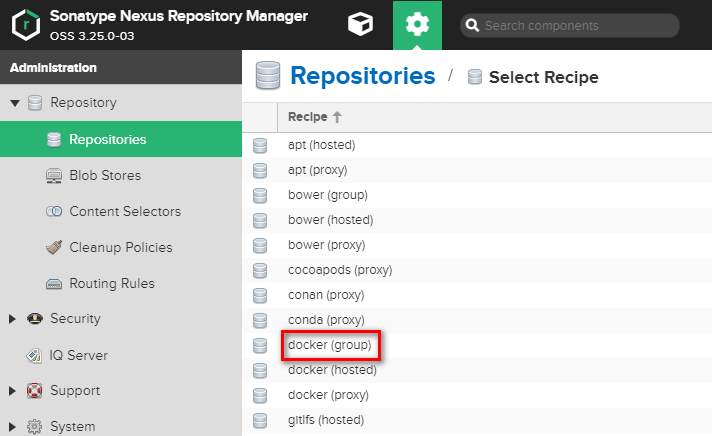
name：docker-group

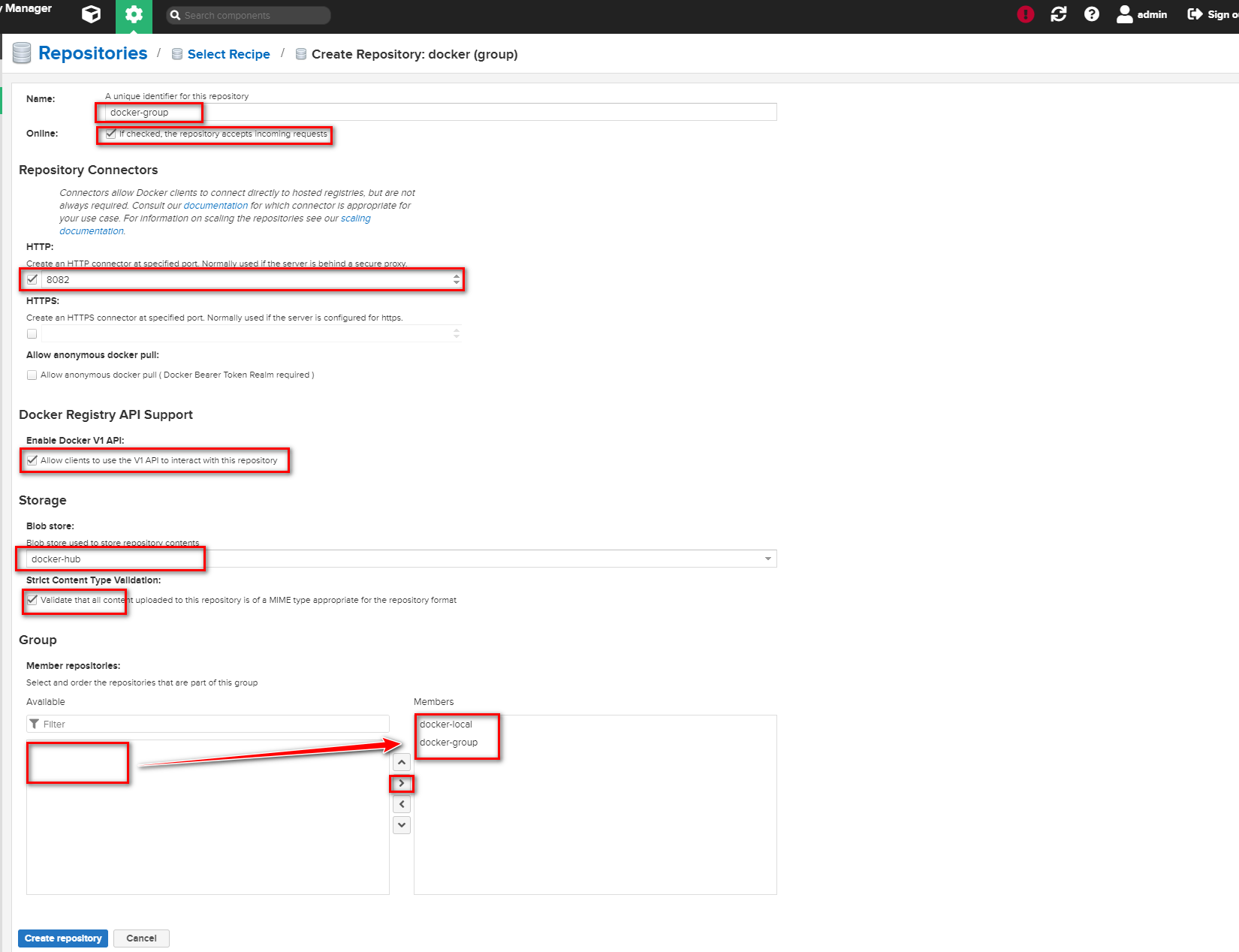
Repository Connectors：启用了一个监听在8082端口的http连接器；

Storage：选择专用的blob存储idocker-hub。

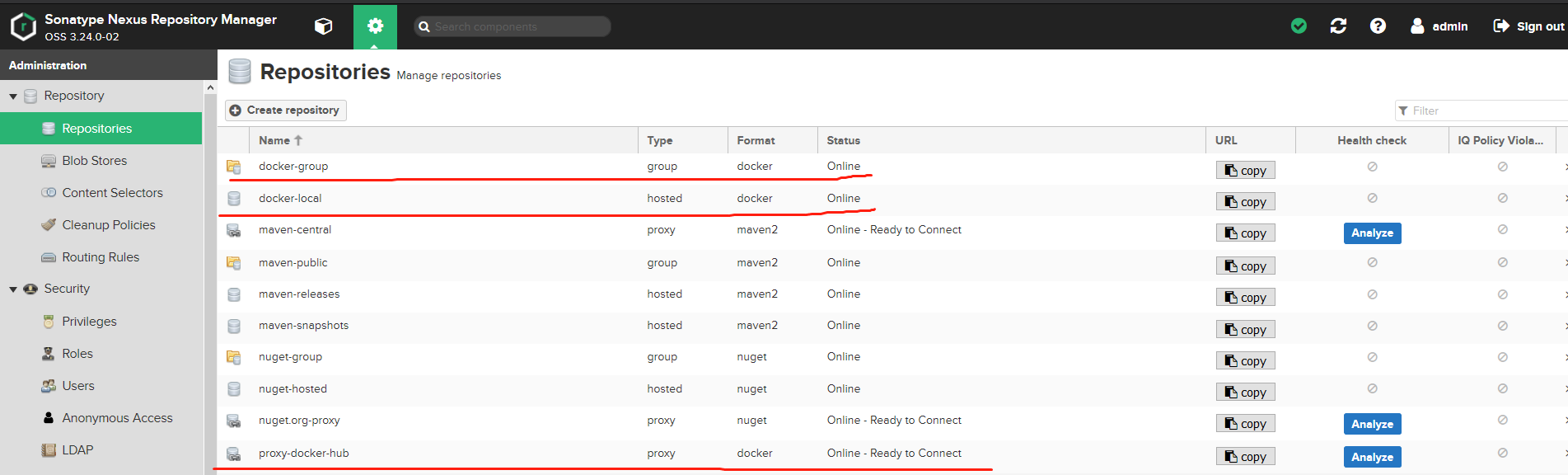
group : 将左边可选的3个仓库，添加到右边的members下。

整体配置截图如下：





 最终效果



**7、测试镜像上传**

1、下载一个测试镜像

docker pull openjdk:8-jdk-alpine

2、修改镜像tag

docker tag openjdk:8-jdk-alpine 192.168.1.50:8083/openjdk:v1

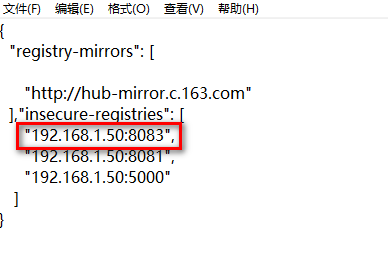
3、上传时出现这个问题需要配置一下docker

docker push 192.168.1.50:8083/openjdk:v1

计算机生成了可选文字:
Iroot@localhost SSI ] # docker push 192.168.1.5e:8@83/openjdk:v1 
The push refers t 0 repository 1192.168.1.5e:8@83/openjdkl 
Get https://192 · 158 · 1 · ： 8e83 / v2 / ： http ： server gave HTTP response to HTTPS client 
Iroot@localhost ssll# . 

配置/etc/docker/daemon.json，添加docker仓库的地址

"insecure-registries": ["192.168.1.50:8083","192.168.1.50:5000"]



出现以下状况问题是没有登录仓库

计算机生成了可选文字:
Iroot@localhost SSI ] # docker push 192.168.1.5e:8@83/openjdk:v1 
The push refers t 0 repository [ 1g2 ． № 8 ． 1.50:8083/openjdkl 
ceafgeIebef5 ： Prepa rIng 
gbgb7f3d56ae ： Prepa rIng 
flbS933fe4bS ： Prepa rIng 
n 0 basic auth ℃ redentials 

执行登录命令

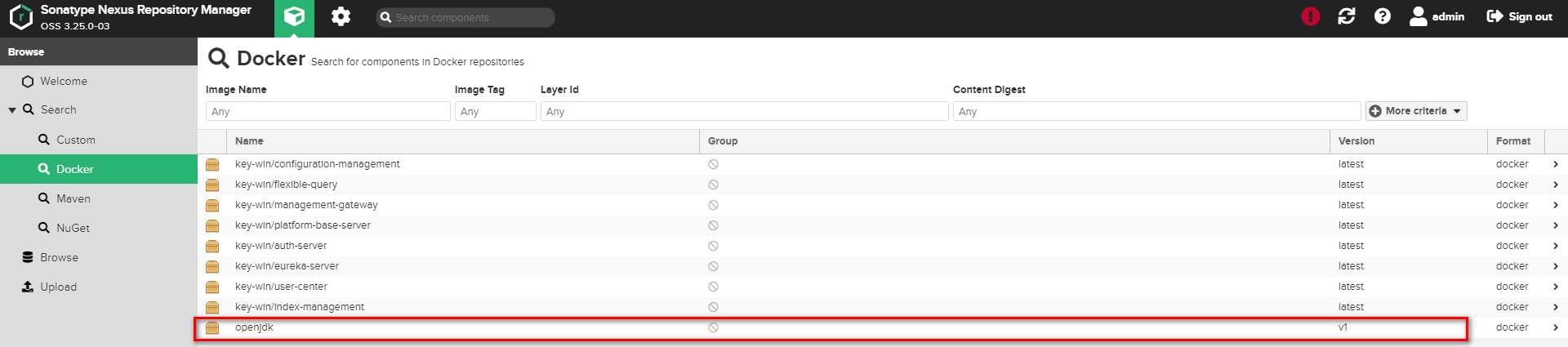
docker login -u admin -p admin123 192.168.1.50:8083

计算机生成了可选文字:
Iroot@localhost SSI ] # docker login u admin -p a 而 in123 1g2 n58 ． 1 ． 50 吒 e83 
-password Via the 〔 廴 I is In S ec u re ． Use -password-stdin 
WAÆING ！ USIng 
W 尉 'I № ！ Yo u r password will be sto red unenc rypted in / root/ .docker/config ． json ． 
Configu re a c redential helper t 0 renove this warnlng. See 
https ：//do℃s.docker.com/engine/reference/commandline/login/#c redentials -sto re 
Login Succeeded 

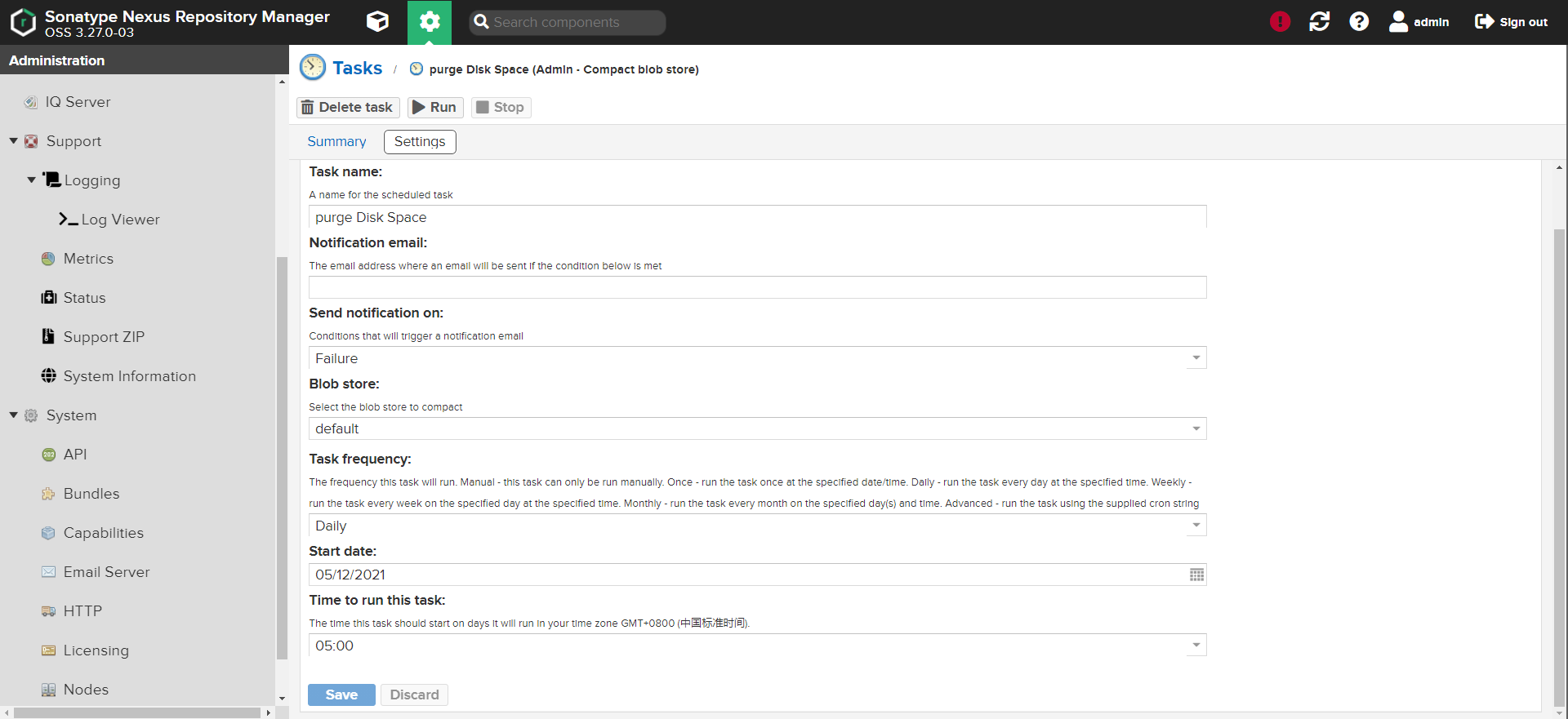
再次执行上传镜像

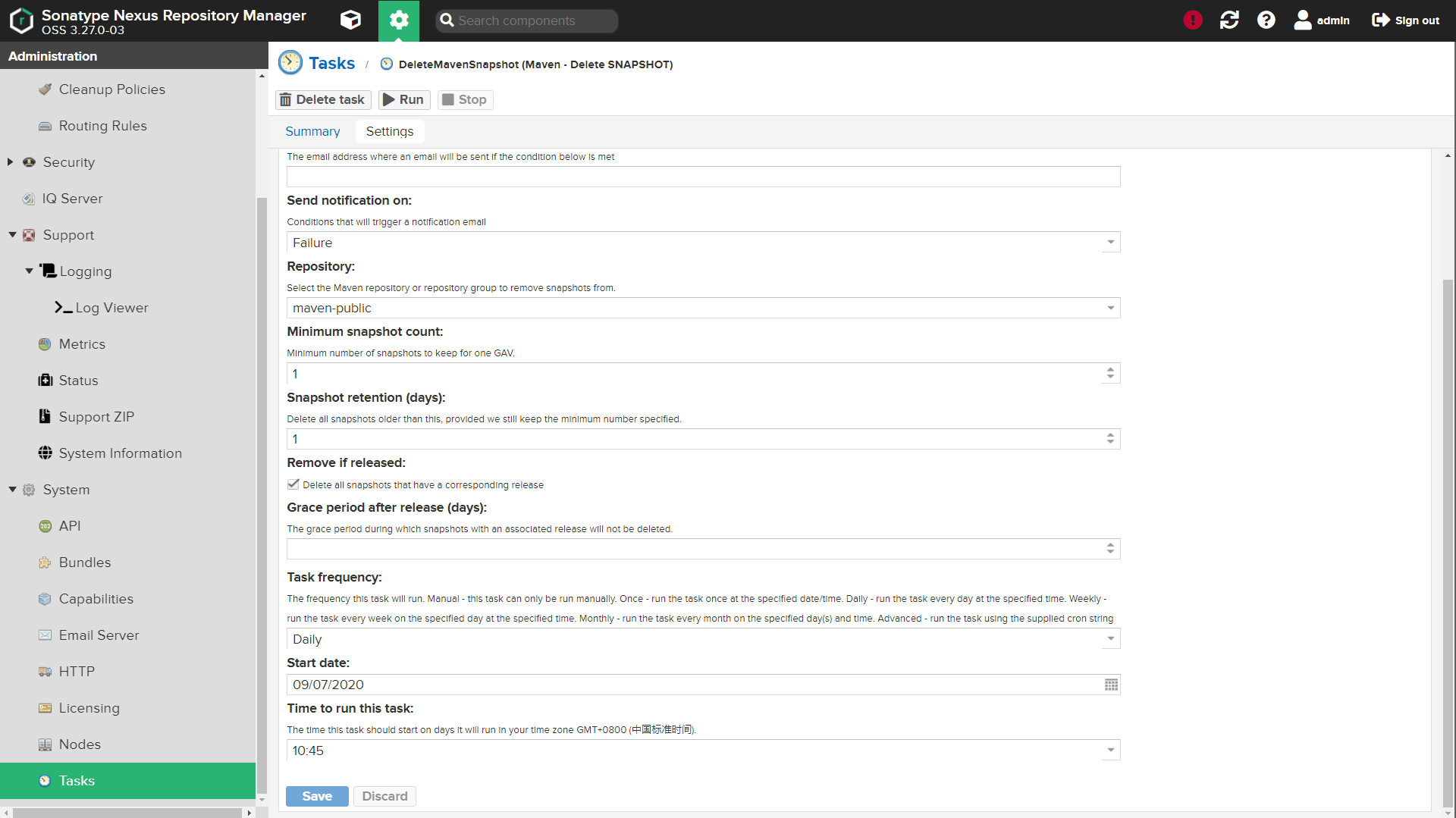
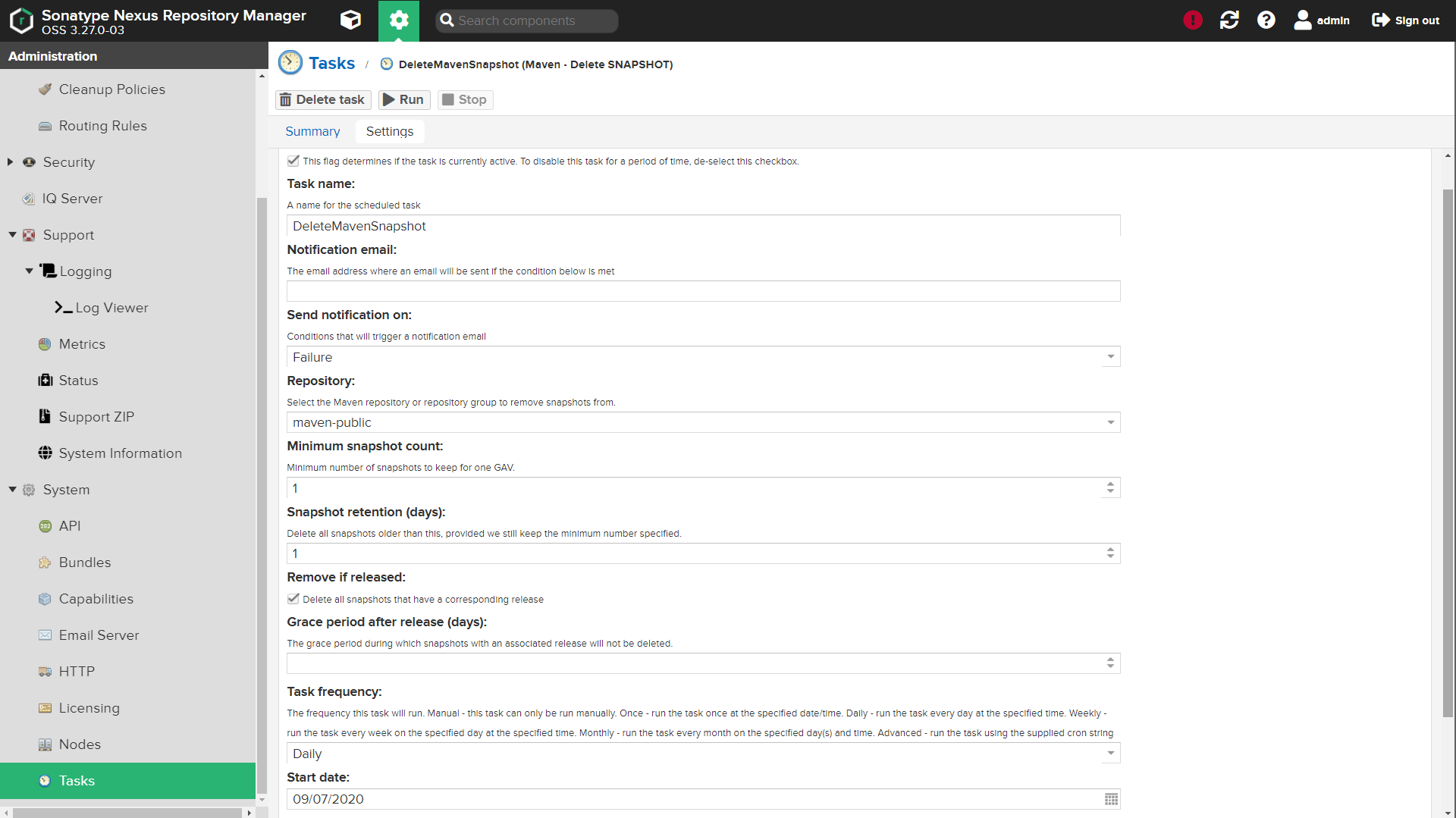
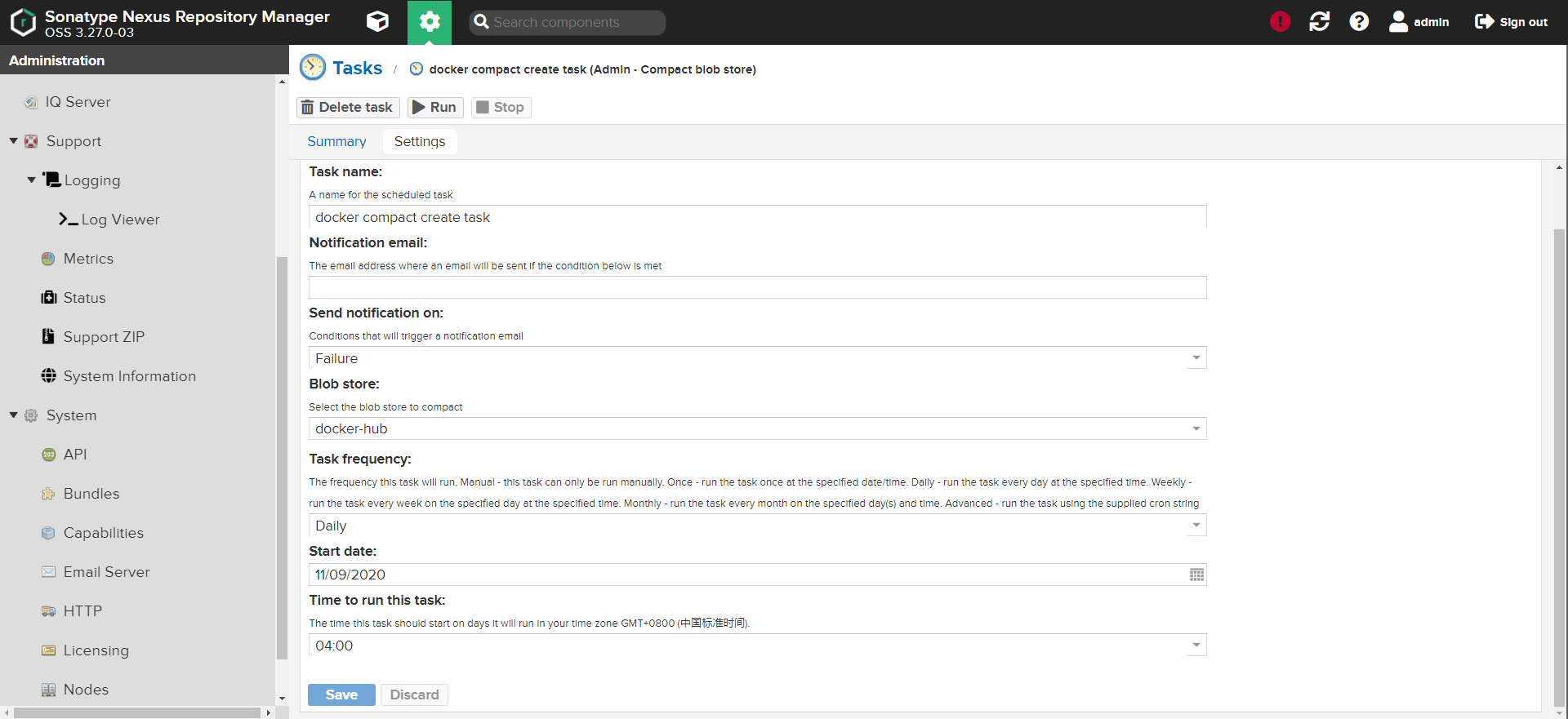
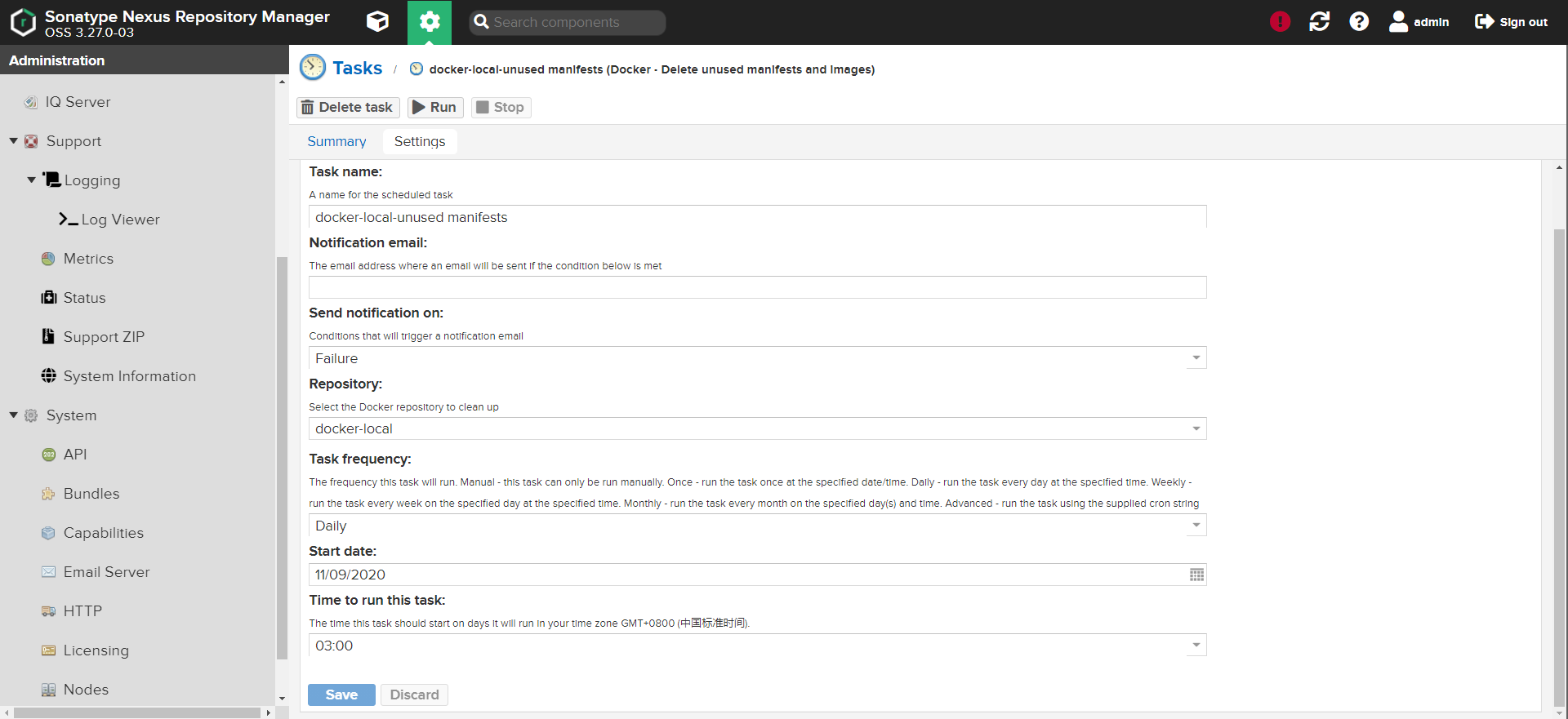
计算机生成了可选文字:
[ root@100a1h0st SSI ] # docker push 192.168.1.5e:8@83/openjdk:v1 
The push refers t 0 repository [ 1g2 ． 158 ． 1.50:8083/openjdkl 
ceafge1ebef5 ： Pushed 
gbgb7f3d56ae ： Pushed 
flb5933fe4b5 ： Pushed 
VI ： digest ： sha256 ： 44b3cea35gcg47527e2b5275cee85c71a81f2efc5e75f5ebb5a13f1ge15d ℃ e71 Slze. 
． g47 
Iroot@localhost ssll# . 

仓库查看已经上传成功



**8、清理Nexus3空间**





du /opt/nexus-data/blobs/ --max-depth=1 -h

df -h