**docker-dev-ops环境搭建**

****

**2018-2-24  
何锡捷**

**目录**

[1. 文档主题 2](#_Toc212875781)

[2. 安装docker集群环境 2](#_Toc95495365)

[2.1. docker安装 2](#_Toc820315246)

[2.2. gluster安装 2](#_Toc193325782)

[3. 持续集成环境部署 2](#_Toc83660163)

[3.1. 环境部署 2](#_Toc1622054403)

[3.2. 配置gogs 5](#_Toc1710936203)

[3.3. 配置nexus 6](#_Toc898730491)

[3.4. 配置jenkins 10](#_Toc1710872886)

[3.5. 配置docker 14](#_Toc1982045319)

[4. 持续集成环境配置 15](#_Toc469344169)

[5. nexus安装 15](#_Toc560012952)

[6. gogs安装 16](#_Toc1864343110)

[7. 安装jenkins 16](#_Toc80756393)

# 文档主题

本文旨在介绍docker基础与实践，可作为docker入门培训教程。

# 安装docker集群环境

## docker安装

## gluster安装

# 持续集成环境部署

## 环境部署

1. 创建持续存储目录，执行

mkdir /sharefs/portainer\_data /sharefs/gogs\_data /sharefs/gogs\_mysql /sharefs/jenkins\_data /sharefs/nexus\_data -p

chmod 777 /sharefs/portainer\_data /sharefs/gogs\_data /sharefs/gogs\_mysql /sharefs/jenkins\_data /sharefs/nexus\_data

1. 创建文件/tmp/dev\_ops.yml内容如下：

version: "3.2"

services:

agent:

image: "portainer/agent"

environment:

AGENT\_CLUSTER\_ADDR: "tasks.agent"

volumes:

- "/var/run/docker.sock:/var/run/docker.sock"

- "/var/lib/docker/volumes:/var/lib/docker/volumes"

networks:

- agent\_network

deploy:

mode: global

placement:

constraints: [node.platform.os == linux]

portainer:

image: "portainer/portainer"

command: "-H tcp://tasks.agent:9001 --tlsskipverify"

ports:

- "9000:9000"

volumes:

- "portainer\_data:/data"

networks:

- agent\_network

deploy:

mode: replicated

replicas: 1

placement:

constraints: [node.role == manager]

gogs:

image: "gogs/gogs"

ports:

- "9005:3000"

volumes:

- "gogs\_data:/data"

depends\_on:

- gogs\_mysql

- portainer

gogs\_mysql:

image: "mysql:5.6"

environment:

MYSQL\_ROOT\_PASSWORD: "123456"

MYSQL\_DATABASE: "gogs"

ports:

- "3306:3306"

volumes:

- "gogs\_mysql\_data:/var/lib/mysql"

depends\_on:

- portainer

jenkins:

image: "jenkins/jenkins"

ports:

- "9006:8080"

volumes:

- "jenkins\_data:/var/jenkins\_home"

depends\_on:

- portainer

deploy:

placement:

constraints: [node.role == manager]

nexus:

image: "sonatype/nexus3"

ports:

- "9007:8081"

- "9008:8082"

- "9009:8083"

volumes:

- "nexus\_data:/nexus-data"

depends\_on:

- portainer

volumes:

portainer\_data:

driver: local

driver\_opts:

type: "none"

o: "bind"

device: "/sharefs/portainer\_data"

gogs\_data:

driver: local

driver\_opts:

type: "none"

o: "bind"

device: "/sharefs/gogs\_data"

gogs\_mysql\_data:

driver: local

driver\_opts:

type: "none"

o: "bind"

device: "/sharefs/gogs\_mysql"

jenkins\_data:

driver: local

driver\_opts:

type: "none"

o: "bind"

device: "/sharefs/jenkins\_data"

nexus\_data:

driver: local

driver\_opts:

type: "none"

o: "bind"

device: "/sharefs/nexus\_data"

networks:

agent\_network:

driver: overlay

attachable: true

1. 在集群manager节点上执行命令：

docker stack deploy -c /tmp/dev\_ops.yml dev\_ops

1. 环境部署完成,使用下面命令查看部署状态

watch docker service ps dev\_ops\_gogs dev\_ops\_gogs\_mysql dev\_ops\_jenkins dev\_ops\_nexus dev\_ops\_portainer dev\_ops\_agent

正常结果应该都是running，如下图:



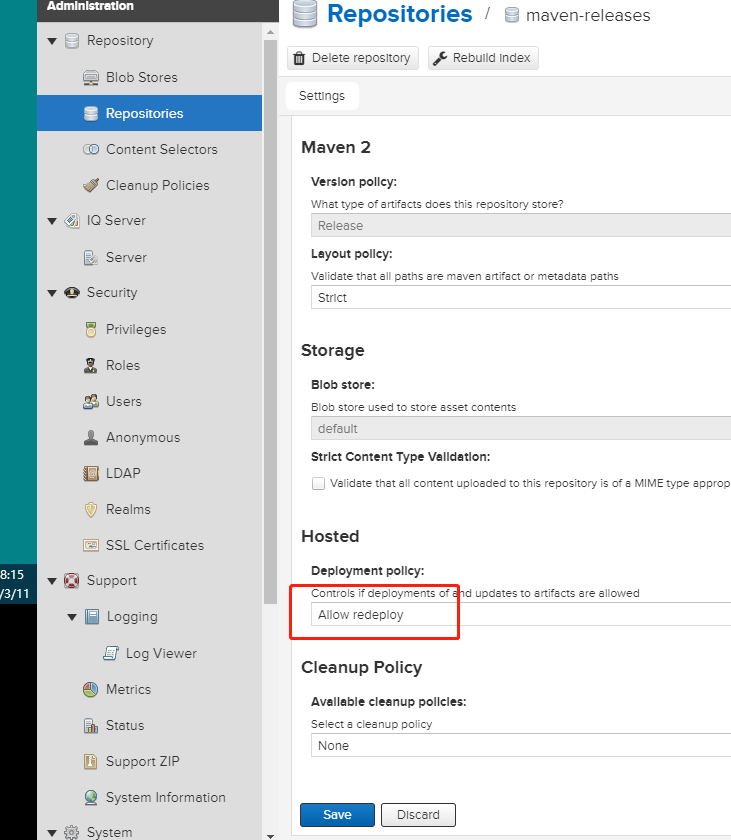
## 配置gogs

1. 访问swarm-manager-01:9005
2. 在页面上配置数据库信息后执行安装，如下图，密码是123456



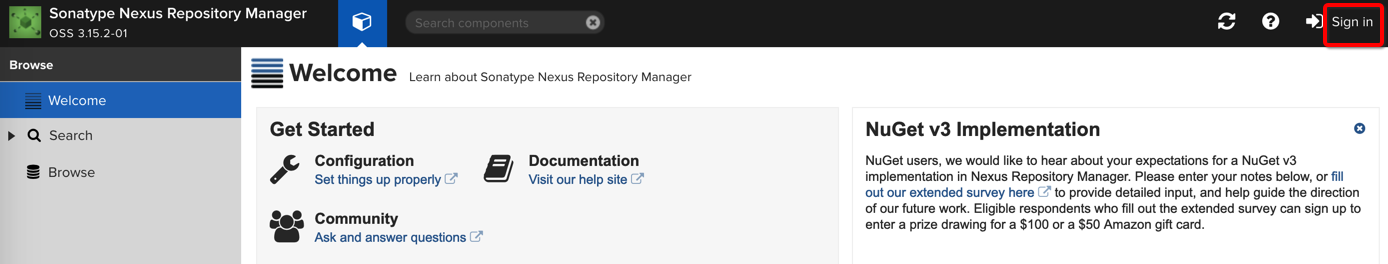
## 配置nexus

配置maven可覆盖发布

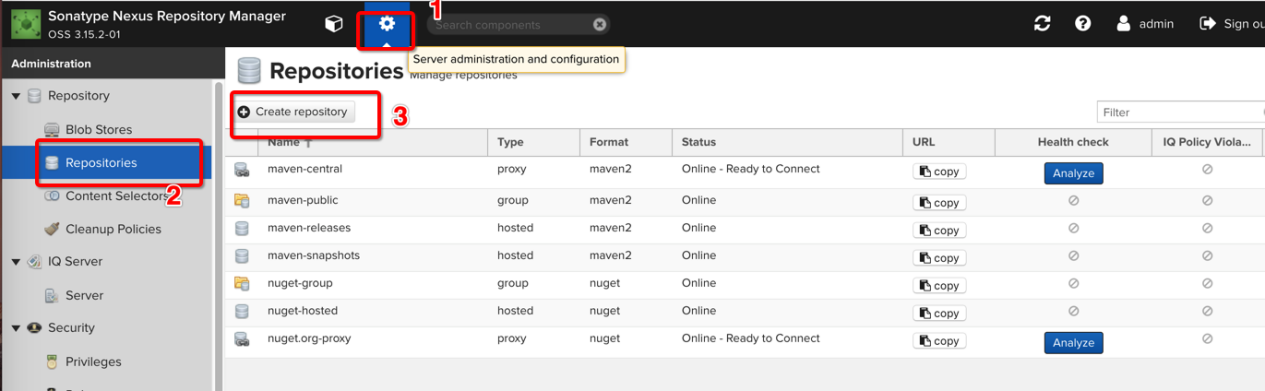


访问swarm-manager-01:9007

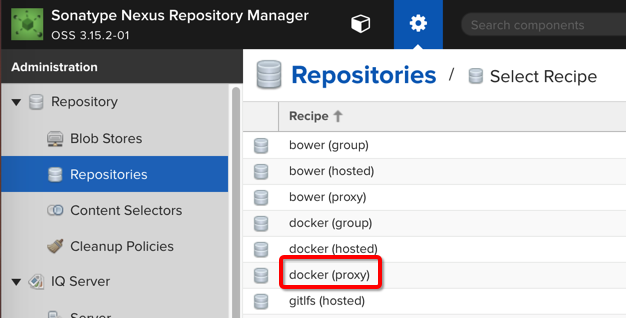
1. 进入系统后点击右上角登录即可，默认用户名密码是：admin/admin123



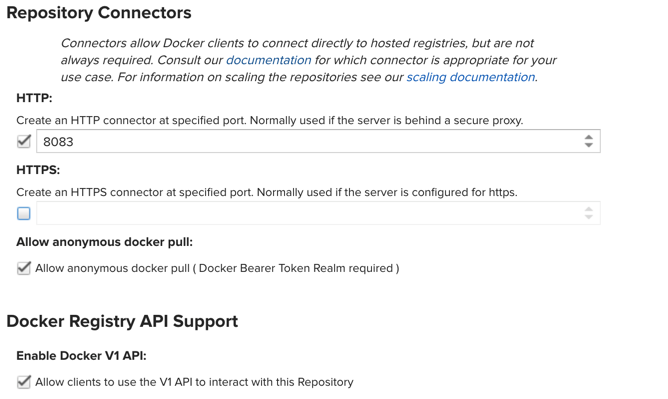
1. 创建docker仓库



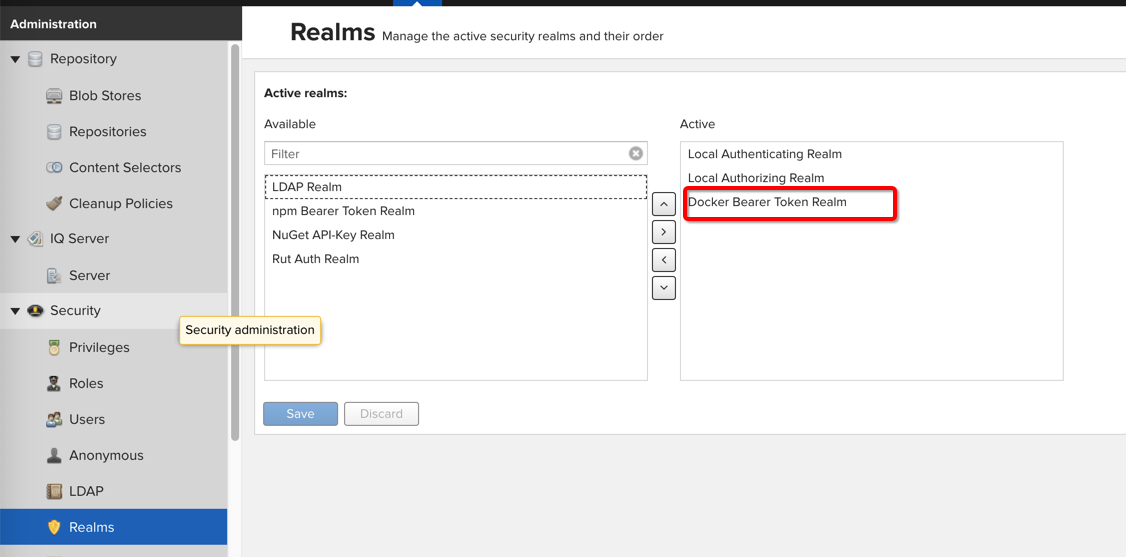
1. 按上图进入创建页面后，选择docker(proxy)



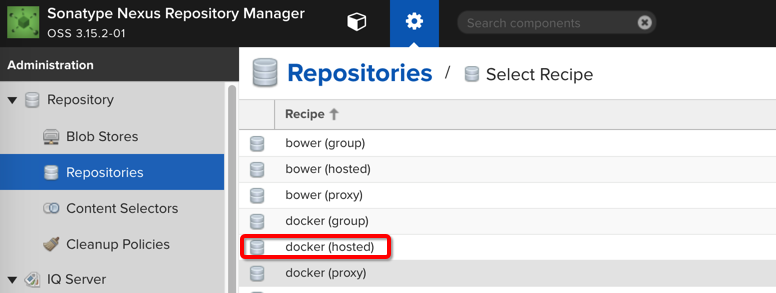
1. 填写名称docker-proxy, proxy为https://registry.docker-cn.com，并且按下图配置connector



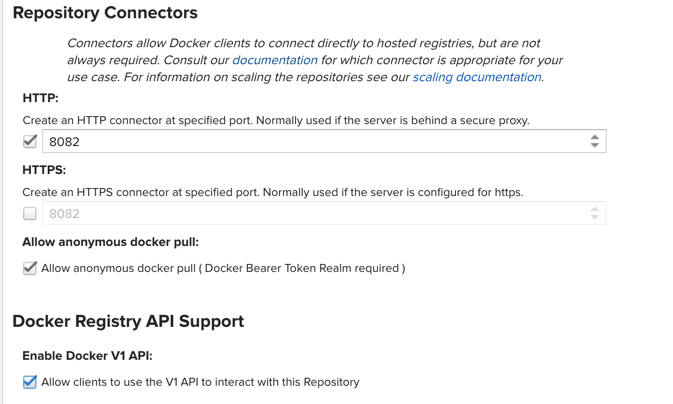
1. 修改安全设置



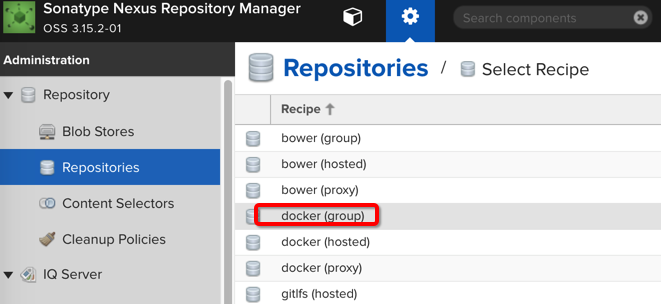
1. 执行保存
2. 新增docker本地仓库



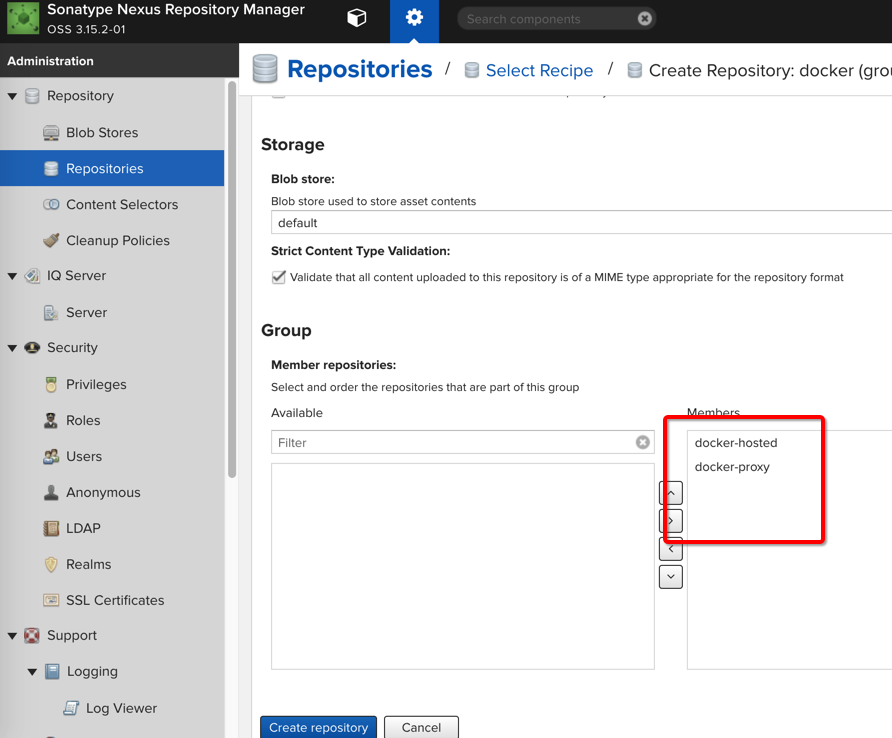
1. 填写名称docker-hosted后，点击创建， 按下图填写connectors部分



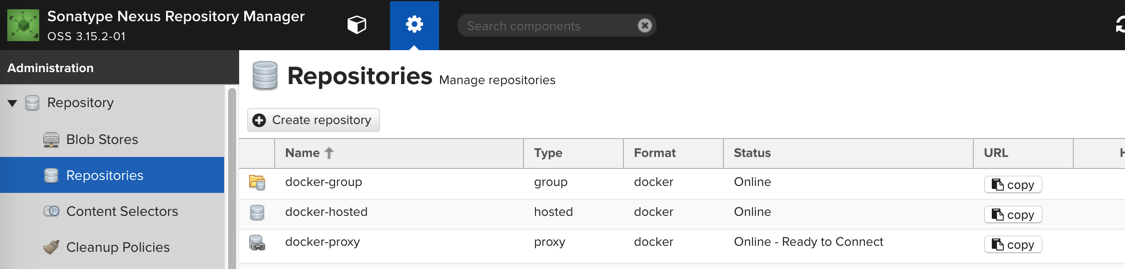
1. 点击保存
2. 新增docker，group仓库



1. 填写名称docker-group后，将上面新增的两个仓库加入当前仓库，点击创建



1. 最终结果



## 配置jenkins

访问swarm-manager-01:9006

1. 在服务器上执行

cat /sharefs/jenkins\_data/secrets/initialAdminPassword



1. 在页面上输入命令展示的密钥



1. 选择安装推荐插件



1. 填写用户名并保存
2. 进入插件配置中心



1. 找到[Maven Integration plugin](https://wiki.jenkins.io/display/JENKINS/Maven+Project+Plugin)插件并安装
2. 进入系统配置



1. 安装maven工具



1. 配置maven配置文件/var/jenkins\_home/maven\_settings.xml



1. 在服务器上创建/sharefs/jenkins\_data/maven\_settings.xml，内容如下(修改红色部分)：

<?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>/var/jenkins\_home/.m2</localRepository>

  <pluginGroups>

  </pluginGroups>

  <proxies>

  </proxies>

  <servers>

    <server>

      <id>dev-releases</id>

      <username>admin</username>

      <password>admin123</password>

    </server>

    <server>

      <id>dev-snapshots</id>

      <username>admin</username>

      <password>admin123</password>

    </server>

  </servers>

  <mirrors>

      <mirror>

        <id>hnykx</id>

        <name>hnykx maven</name>

        <url>http://192.168.56.7:9007/repository/maven-public/</url>

        <mirrorOf>central</mirrorOf>

      </mirror>

  </mirrors>

  <profiles>

  </profiles>

</settings>

## 配置docker

修改/etc/docker/daemon.json为

{

  "registry-mirrors": [

    "https://registry.docker-cn.com"

  ],

  "insecure-registries": [

    "192.168.56.7:9007",

    "192.168.56.7:9008",

    "192.168.56.7:9009"

  ]

}

重启docker

service docker restart

执行

 docker login 192.168.56.7:9008

------------------

拉取镜像例子：

docker pull 192.168.56.7:9009/redis

推送镜像例子：

docker tag redis:latest 192.168.56.7:9008/myredis:latest

docker push 192.168.56.7:9008/myredis:latest

# 持续集成环境配置

# nexus安装

mkdir /sharefs/nexus -p && chown -R 200:200 /sharefs/nexus && chmod 777 /sharefs/nexus # 需要修改成200，不然容器会爆没有权限的错误

docker service create -p8100:8081 -d --mount type=bind,source=/sharefs/nexus,destination=/nexus-data sonatype/nexus3

访问8100端口，默认用户名密码是:admin/admin123

# gogs安装

mkdir /sharefs/gogs

chmod 777 /sharefs/gogs

docker service create -d \

--publish 8102:3000 \

--name gogs \

--mount type=bind,source=/sharefs/gogs,destination=/data \

gogs/gogs

# 安装jenkins

mkdir /sharefs/jenkins -p

chmod 777 /sharefs/jenkins

docker service create -d --mount type=bind,source=/sharefs/jenkins,destination=/var/jenkins\_home \

-p 8101:8080 \

jenkins/jenkins:lts-alpine