# 1阿里云容器配置

关于阿里云容器的基本概念请参考

https://help.aliyun.com/document\_detail/25975.html?spm=5176.doc51489.6.544.G19bFk

## 容器服务使用步骤：

第一步：创建集群，选择集群的网络环境，设置集群的节点个数和配置信息。（这里需要一个节点ECS）

第二步：选择镜像或编排模板（如果您的应用由多个镜像承载的服务组成，可以选择一个编排模板）。(这里需要开通Docker镜像仓库)

第三步：创建应用并部署。

第四步：查看部署后应用的状态和相应的服务、容器信息。

## 详细步骤

前提准备

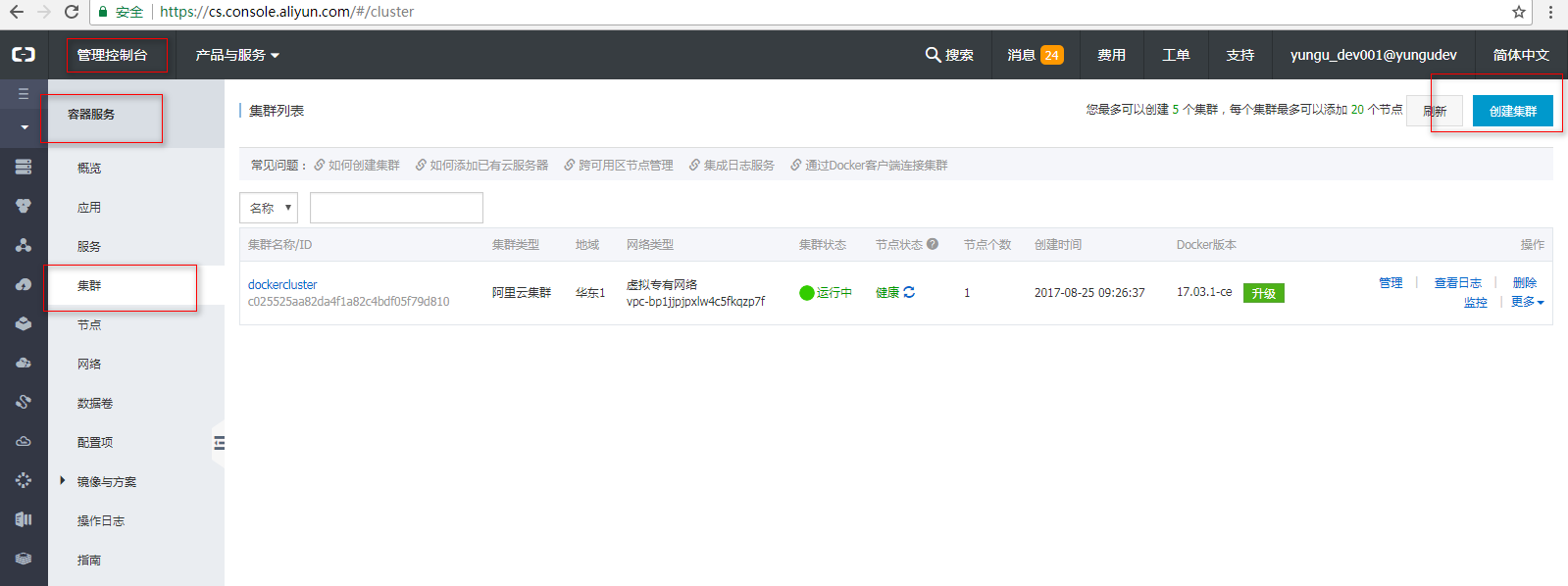
①创建华东1区的交换机，

②开通一台华东1的ecs

③开通负载均衡

### 1集群创建步骤

进入阿里云管理控台-->容器服务-->集群-->右上角创建集群



地域：华东1

可用区：（选择有vpc的区域）

集群模式：默认不选

网络类型：专有网络（可用区选过之后这个地方会自动选择，如果没有网络就创建交换机）

容器起始网段：默认

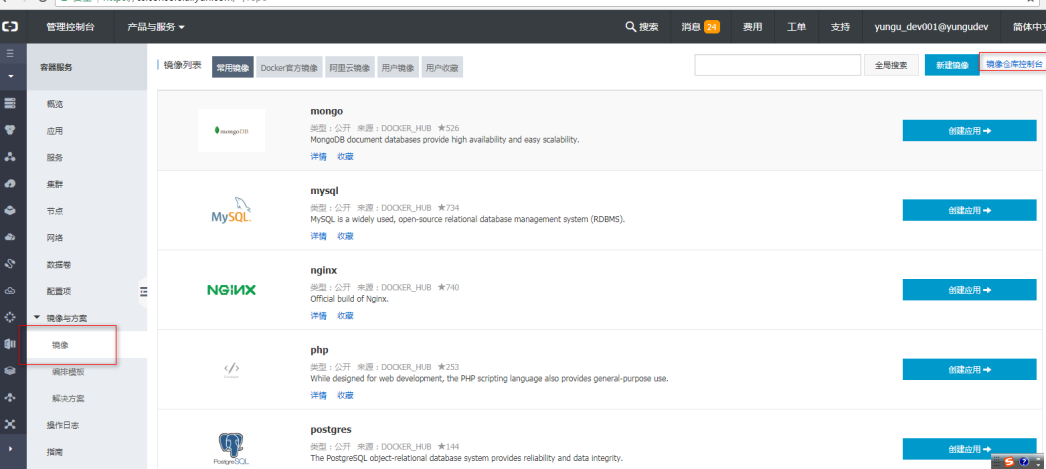
是否新增节点：选择添加已有节点（输入选择已有ecs） 如果没有ecs选择穿件节点

负载均衡：如果之前没有创建负载均衡这里可以选择自动创建负载均衡

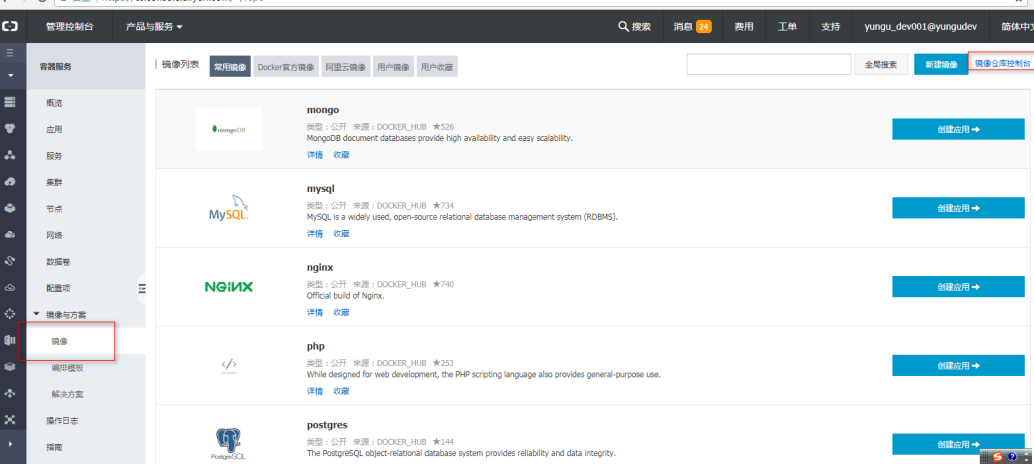
点击创建集群

### 2创建镜像仓库

1容器服务-->镜像与方案-->右上角镜像仓库控制台（如果未开通，需要开通，记住初始密码）



2先创建namespace



3创建镜像仓库

4=3fsd

### 3上传镜像到阿里云镜像仓库

镜像列表-->管理-->参照指南

在有镜像的服务器操作（日常服务器）

vpc上传：

sudo docker login --username=yungu\_dev001@yungudev registry-vpc.cn-hangzhou.aliyuncs.com

$ sudo docker tag [ImageId] registry.cn-hangzhou.aliyuncs.com/yungu-docker/dcsync:[镜像版本号]

$ sudo docker push registry.cn-hangzhou.aliyuncs.com/yungu-docker/dcsync:[镜像版本号]

外网上传：

$ sudo docker login --username=yungu\_dev001@yungudev registry.cn-hangzhou.aliyuncs.com

（成功后会显示successed）登录registry的用户名是您的阿里云账号全名，密码是您开通镜像仓库时设置的密码。

$ sudo docker tag [ImageId] registry.cn-hangzhou.aliyuncs.com/yungu-docker/dcsync:[镜像版本号]

$ sudo docker push registry.cn-hangzhou.aliyuncs.com/yungu-docker/dcsync:[镜像版本号]

注意您的网络环境

从ECS推送镜像时，可以选择走内网，速度将大大提升，并且将不会损耗您的公网流量。

如果您申请的机器是在经典网络，请使用 registry-internal.cn-hangzhou.aliyuncs.com 作为registry的

域名登录, 并作为镜像名空间前缀

如果您申请的机器是在vpc网络的，请使用 registry-vpc.cn-hangzhou.aliyuncs.com 作为registry的域名

登录, 并作为镜像名空间前缀

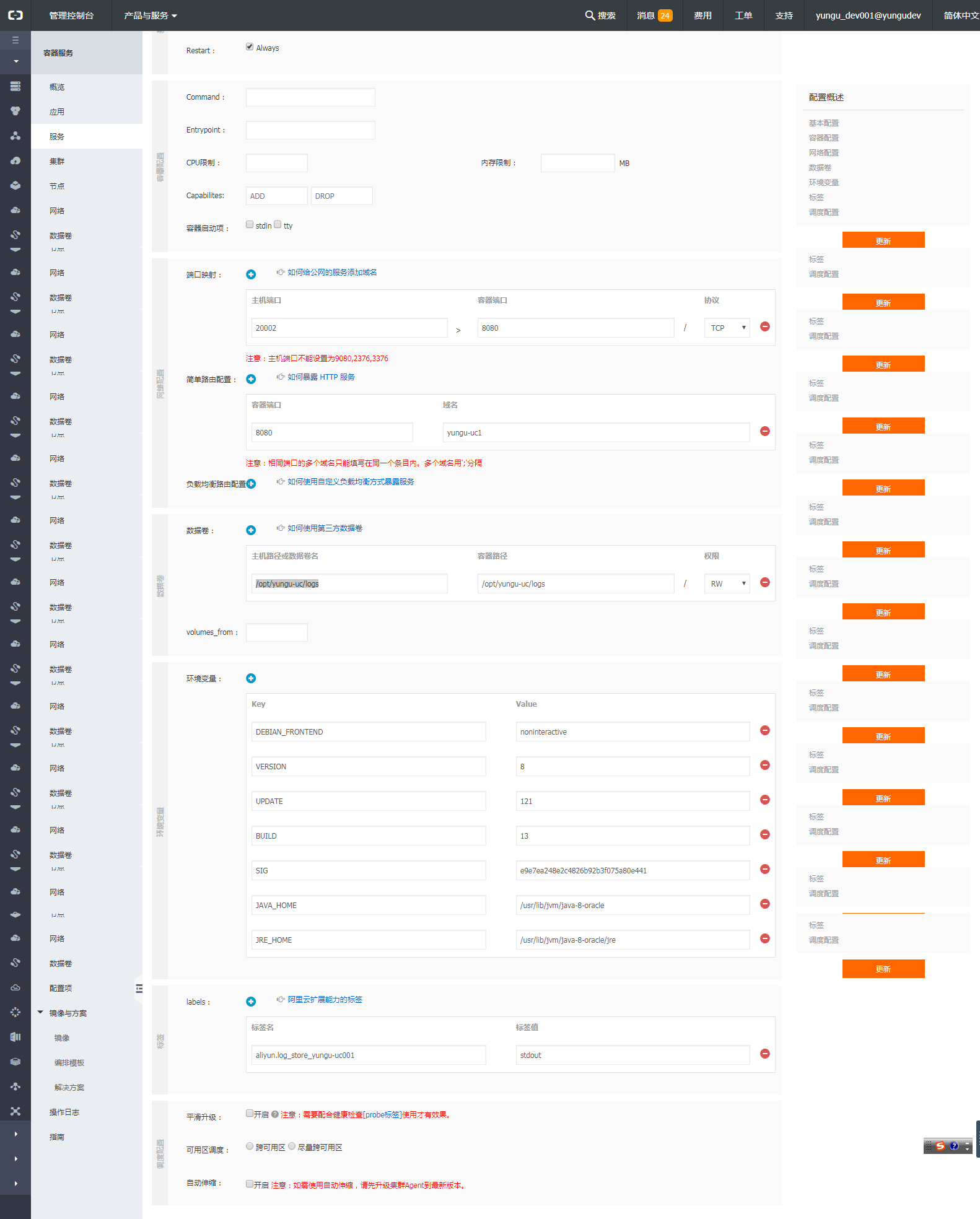
可在 镜像列表-->管理-->镜像版本中查看推送的镜像是否成功

### 4创建应用

容器服务--》应用--》创建应用

填写 相关信息

页面最后选择使用镜像创建



镜像名称：选择镜像（选择自己构建的镜像） 选择镜像版本

Command：不用填

Entrypoint: 不用填

端口映射+：8990:8080（项目启动的端口号:容器端口）TCP

简单路由+： 8080（容器端口） yunug-uc(域名)

如果需要挂载数据卷

可选择数据卷： 如testsso（对应第三方数据卷名，需要创建数据卷） -- /aaa(这里对应docke容器的路径，必须写绝对路径) RW

如/logs （对应宿主机器的路径） -- /logs(这里对应docke容器的路径，必须写绝对路径）RW

labes:

aliyun.log\_store\_dbstdouts stdout

(aliyun.log\_store\_dbstdouts stdout容器集成sls日志相关也可以用编排模板在labels中加入 aliyun.log\_store\_dbstdouts: stdout

需要理解这个label的定义：

aliyun.log\_store\_{name}: {logpath}

其中name表示日志服务中logstore的名字，实际创建的logstore名字是acslog-${app}-${name}，

app是容器服务中的应用名。logpath是容器中日志的路径，stdout是一个特殊的logpath，表示标准输出。参考https://yq.aliyun.com/articles/9068)

其他配置暂时都是默认，之后在更改配置中修改

点击创建

访问：应用--》应用名称--》路由列表--》点击链接

访问不成功：

原因可能

1开通负载均衡,

2集群-->管理-->负载均衡--》绑定slb

也可以在集群创建之前开通负载均衡，创建集群的还是选择默认创建负载均衡，之后会自动绑定slb

### 5创建触发器

阿里云镜像服务为每个仓库提供了Webhook的功能，帮助用户在镜像构建成功后进行消息的推送，实现持续集成的后续流程。如果用户设置了一个容器服务触发器的Webhook，那么当镜像构建成功后，将会自动触发容器服务上应用拉取新的镜像，并进行重新部署。

点击应用--->创建触发器





### 6集成日志服务

点击集群--》管理--》开启日志服务

参考 https://yq.aliyun.com/articles/9068

1如果使用镜像方式在创建应用的第二步--应用配置中

labels：

aliyun.log\_store\_dbstdouts stdout （也可选择其他路径）

2如果使用编排模板在编排模板的labels中加入

aliyun.log\_store\_dbstdouts: stdout

注意：保障检查主账号是否配置了AccessKey

请确保主账号下至少有一个启用状态的AccessKey

### 7挂载外部目录

1挂载宿主机器的目录

如果使用镜像方式在创建应用的第二步--应用配置中

数据卷：/logs(宿主机器目录) /logs（容器目录，必须是绝对路径）

如果使用编排模式：

在模板中加入

volumes:(是个标签与labels同级，可简单了解编排模板的语法)

- /logs:/logs

2挂载第三方数据卷

参考：https://yq.aliyun.com/articles/7581（很简单）

### 8补充

1关于编排模板，如果想创建编排模板，可以在现有的应用中点解更改配置，里面的模板，创建成自己的模板。

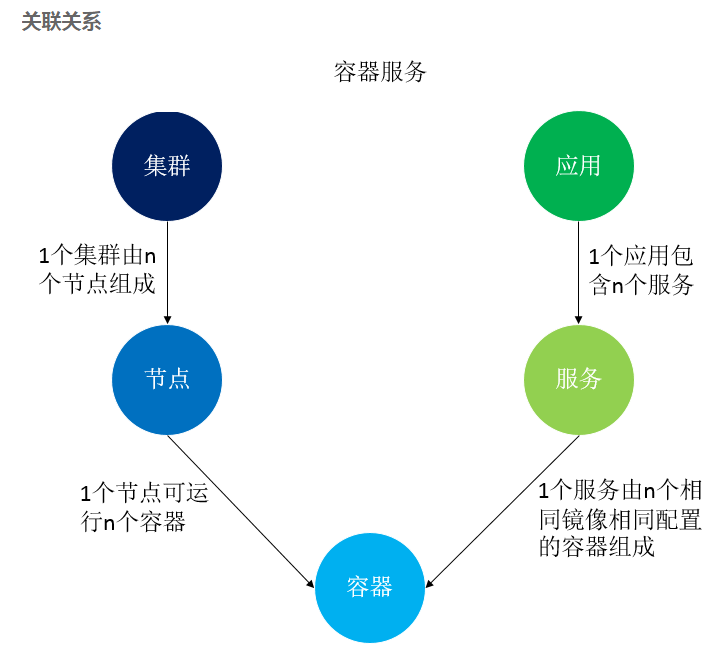
2如果运行中的应用想加入日志，挂载外部目录，可在更改配置中进行修改

3 关于服务和应用的概念

https://help.aliyun.com/document\_detail/25975.html?spm=5176.doc51489.6.544.G19bFk

应用： 一个应用可通过单个镜像或一个编排模板创建，每个应用可包含1个或多个服务。

服务：一组基于相同镜像和配置定义的容器，作为一个可伸缩的微服务。



# 2.安装使用jenkins

参考：http://www.jb51.net/article/114637.htm

http://www.cnblogs.com/yangcb/p/5852380.html

## 2.1安装jdk

yum install java-1.8.0-openjdk

## 2.2yum安装jenkins

### 2.2.1yum的repos中默认是没有Jenkins的，需要先将Jenkins存储库添加到yum repos

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

### 2.2.2yum install jenkins

yum install jenkins

如果网络不好此时可能会下载不下来，可到官网下载jenkins的rpm包，官网地址http://pkg.jenkins-ci.org/redhat-stable/

wget http://pkg.jenkins-ci.org/redhat-stable/jenkins-2.7.3-1.1.noarch.rpm

rpm -ivh jenkins-2.7.3-1.1.noarch.rpm

（jenkins日常环境wget可能下载不下来，可直接下载好rmp包上传到日常）

### 2.2.3配置jenkins

vi /etc/sysconfig/jenkins

JENKINS\_USER="root"

给jenkins root权限

### 2.2.4启动jenkins

service jenkins restart

### 2.2.5在dev-proxy配置jenkins代理

upstream backend\_jenkins {

server 192.168.1.135:8080;

}

server {

listen 80;

#listen 443 ssl;

server\_name jenkins.daily.yungu-inc.org;

#ssl\_certificate /etc/nginx/ssl/214201237270770.pem;

#ssl\_certificate\_key /etc/nginx/ssl/214201237270770.key;

#ssl\_session\_timeout 5m;

#ssl\_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;

#ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;

#ssl\_prefer\_server\_ciphers on;

location / {

proxy\_set\_header HOST $host;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_pass http://backend\_jenkins;

client\_max\_body\_size 500m;

}

}

重启nginx systemctl restart nginx

### 2.2.6访问jenkins

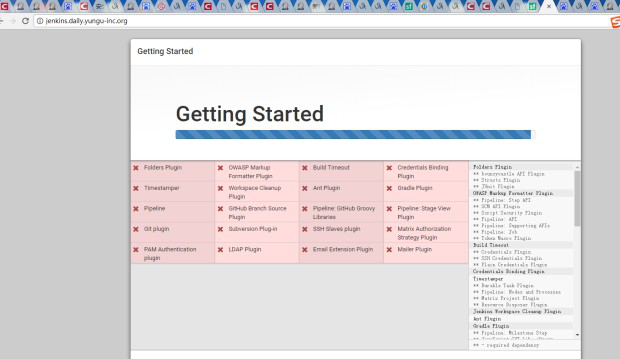
jenkins.daily.yungu-inc.org

1进入jenkins登录界面，Unlock jenkins

2根据提示可以在/var/lib/jenkins/secrets/initialAdminPassword找到密码

3 install suggested plugins选择安装所有插件

在日常服务器，因为网络超时。一个插件也安装不上, 之后可以手动安装



4 耐心等待全部安装完成。安装完成后，页面自动进入了管理员账户注册页面

5 完成后jenkins主页面，新建项目

问题：此时可能因为插件都没有安装的缘故，新建项目时源码管理没有git这个选项

可以选择 系统管理-->插件管理-->可选插件-->选择git相关插件（ [GitLab Plugin](https://wiki.jenkins-ci.org/display/JENKINS/GitLab+Plugin)，[Gitlab Hook Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Gitlab+Hook+Plugin) ,[Gitlab Authentication plugin](http://wiki.jenkins-ci.org/display/JENKINS/Gitlab+OAuth+Plugin) , [GitHub Branch Source Plugin](https://wiki.jenkins-ci.org/display/JENKINS/GitHub+Branch+Source+Plugin) ）

选择直接安装。

如果安装不上，选择高级，上传插件（手动去官网下载），

插件地址http://updates.jenkins-ci.org/download/plugins/

上传一个插件可能依赖其他很多插件。比如安装git-cilent.hpi之前需要安装credentials.hpi,scm-api.hpi,workflow-scm-step.hpi,workflow-sep-api.hpi, ssh-credentials.hpi在https://wiki.jenkins.io/display/JENKINS/Git+Client+Plugin。可以看到他的依赖，上传过程中也可能上传失败。因为是nginx 代理，上传文件大小有限制，此时也要在nginx配置文件上传大小的限制。之后也可能因为网络超时，上传不了。在日常环境下，可以把之间安装好的插件/var/lib/jenkins/plugins直接拷贝到日常环境的/var/lib/jenkins/plugins下

重启jenkins

### 2.2.7安装gradle docker

说明 使用jenkins的插件gradle和docker也可以。用命令执行比较熟悉,所以在安装jenkins的机器上，安装gradle和docker，在构建项目时执行脚本命令

解压gradle tar包，在/root/.bash\_profile，中配置gradle,jdk环境变量，上面已经给了jenkins root权限。

yum install docker

systemctl start docker

## 2.3 jenkins使用

### 2.3.1配置credentials

http://blog.csdn.net/gw569453350game/article/details/51911179

因为要用git管理项目，需要配置用户信息、

1 http下载只需要配置Username with password，

2 ssh 方式需要Private Key：可以指定文件，也可以使用默认的 ~/.ssh，当然也可以直接将私钥复制粘贴到此处（home应用需要配置）

### 2.3.2新建项目

1源码管理选择 git

Repository UR ： gitlab 项目地址

Credentials：刚才配置的

Branch Specifier (blank for 'any') \*/daily/0.0.2 （代码分支）

2构建

选择Execute shell

Command:

source /root/.bash\_profile (加载root的配置文件，前面已经配置jenkins有root权限)

docker build -t registry-vpc.cn-hangzhou.aliyuncs.com/yungu-docker/work:latest .（生成镜像）

docker login -u=yungu\_dev001@yungudev -p=YunguDev@0811

registry-vpc.cn-hangzhou.aliyuncs.com （登录阿里云仓库）

docker push registry-vpc.cn-hangzhou.aliyuncs.com/yungu-docker/work:latest

（上传到阿里云镜像仓库）

保存立即构建

补充：为了确保镜像一更新，应用就重新部署，在阿里云容器-->应用名称-->创建触发器