**目录**

[新增超级root权限用户](#77hhkd1496649579102)

[Linux配置JDK](#68ryar1496647972076)

[Linux网络相关命令](#14ouft1496648141593)

[Nginx 安装与配置](#76udsa1496648950626)

[Nginx安装](#64udrc1496648950902)

[Nginx相关命令](#7tymq1496648950902)

[yum相关命令](#49fihw1496648950902)

[docker安装与配置](#34cwvg1496649270382)

[docker安装](#68zctf1496649290829)

[docker相关配置](#59hqyo1496649307700)

[开放2375端口](#63uehf1496649337532)

[/etc/docker/dame.json配置;](#63jglq1496649349353)

[docker安装registry](#79alhj1496648950902)

[docker 安装showdoc](#4wvuh1496648950902)

[docker相关命令](#69rbig1496648950902)

[停止所有的容器](#38rnxw1496649475796)

[删除所有的容器](#68xyjt1496649482795)

[删除所有的镜像](#14ilci1496649482795)

[进入容器](#86bnih1496649503497)

[docker 安装vim](#46krui1496648950902)

[docker-compose安装流程](#3225-1504574066789)

[gradle安装](#90gkoq1496648950902)

[svn安装](#30vqak1496648950902)

[安装命令](#62wxib1496648950902)

[创建版本库](#95qiby1496648950902)

[版本库配置](#17xwwb1496649808687)

[代码检出](#1fjkt1496649941861)

[版本库迁移](#4fbub1503904888557)

[SVN过滤](#8293-1507511897232)

[alias脚本配置](#23nqsb1502090107271)

[JVM相关查询](#85xisw1502273881298)

[jps -Vv c --查询相关java进程](#75isdc1503966240537)

[mongodb的安装与配置](#99bbqy1503966270218)

[docker安装mongodb](#6xydd1503966280105)

[docker安装gitlab](#67xfoh1503967752813)

[1.切换阿里云的加速地址](#8546-1504078974830)

[2.下载gitlab镜像](#3lteo1503967752894)

[3.创建容器](#2189-1504079039934)

[showdoc安装](#79pxwk1503967753134)

[更换centos yum源为阿里云源](#95tabf1503967753299)

[redis安装](#9229-1505113398469)

[修改linux的时区](#4766-1507512652769)

[linux日志查找](#3910-1507512938222)

[docker安装redis](#4432-1509939196047)

**新增超级root权限用户**

|  |  |
| --- | --- |
| sudo useradd -u 0 -o -g 0 lotusit | 建立一个与Root权限相关的用户;  -u 0  ：建立一个uid=0的帐号；  -o ：允许使用重复的ID号，字母o，不是数字0  -g 0 ：建立一个gid=0的帐号；  lotusit  :帐号名 |

**Linux配置JDK**

export JAVA\_HOME=/yaoche/java/jdk1.8.0\_151
export JRE\_HOME=${JAVA\_HOME}/jre
export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib
export PATH=${JAVA\_HOME}/bin:$PATH

**Linux网络相关命令**

1.关闭防火墙
service iptables stop
2.永久关闭防火墙，重启生效
hkconfig iptables off
3.修改IP
vi /etc/sysconfig/network-scripts/ifcfg-eth0
4.重启网络服务
serivce network restart

**Nginx 安装与配置**

**Nginx安装**

rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm

yum install nginx

**Nginx相关命令**

**yum相关命令**

查看已经安装的应用

yum list installed | grep docker

删除安装的应用

yum -y remove docker-engine.x86\_64

**docker安装与配置**

**docker安装**

yum list installed | grep docker    --查找docker安装包

yum -y remove docker-engine.x86\_64     --清除docker安装包

yum -y install docker                 --安装docker

**docker相关配置**

**开放2375端口**

在/etc/sysconfig/docker的OPTIONS='--selinux-enabled --log-driver=journald --signature-verification=false'

下一行添加OPTIONS='-H tcp://0.0.0.0:2375 -H unix://var/run/docker.sock'

**/etc/docker/dame.json配置;**

**特别注意，不能直接copy到文件中，只能手工敲，不然会有编码错误异常；**

{

    "live-restore": true,

    "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn", "http://e31317bb.m.daocloud.io"],

    "insecure-registries": ["http://192.168.1.28:5000"],

    "graph": "/mnt/resource/docker",

    "hosts": ["tcp://0.0.0.0:2375", "unix:///var/run/docker/docker.sock"]

}

cat /usr/local/nginx/logs/access.log | wc -l

**docker安装registry**

docker run -d -v  /mnt/resource/registry:/var/lib/registry -p 5000:5000 --restart=always --name registry docker.io/registry

**docker 安装showdoc**

docker run -d --name showdoc -p 8081:80 -v /lotus/showdoc/data/Public:/var/www/html/Public -v /lotus/showdoc/data/Sqlite:/var/www/html/Sqlite showdoc

**docker相关命令**

**停止所有的容器**

docker ps -a | grep "Exited" | awk '{print $1 }'|xargs docker stop

**删除所有的容器**

docker ps -a | grep "Exited" | awk '{print $1 }'|xargs docker rm

**删除所有的镜像**

docker images|grep none|awk '{print $3 }'|xargs docker rmi

**进入容器**

docker exec -it registry（容器名） bash

**docker 安装vim**

apt-get update --同步 /etc/apt/sources.list 和 /etc/apt/sources.list.d 中列出的源的索引，这样才能获取到最新的软件包。

apt-get install vim --安装vim命令

**docker-compose安装流程**

1、安装python-pip

yum -y install epel-release

yum -y install python-pip

2、安装docker-compose

pip install docker-compose

待安装完成后，执行查询版本的命令，即可安装docker-compose

docker-compose version

**gradle安装**

wget https://downloads.gradle.org/distributions/gradle-3.4.1-all.zip

uzip gradle-3.4.1-all.zip

vi /etc/profile

--在最后面加入以下

export GRADLE\_HOME=/lotus/gradle/gradle-3.4.1

PATH=$PATH:$GRADLE\_HOME/bin

**svn安装**

**安装命令**

yum -y install subversion

**创建版本库**

svnadmin create platform --platform版本库名称

**版本库配置**

版本库本身配置

授权配置

用户名与密码

**代码检出**

svn checkout svn://42.159.246.182/o2oapp/service/branches/lotus\_1.8 --带根目录

svn checkout  svn://192.168.1.20/platform/trunk/code/lotusgo-html/lotusgo-manager . --不带根目录

svn checkout svn://42.159.246.182/o2oapp/service/branches/lotus\_1.8

**版本库迁移**

svnadmin dump platform >platform.dump --生成备份文件

svnadmin load reponame < yourreponame.dump 加载备份文件

**SVN过滤**

1.进入Eclipse>Preferences>Team>Ignored Resources>Add Pattern

2.加入以下规则至列表中

.project

.factorypath

.classpath

target

target/\*

.settings

.settings/\*

.apt\_generated

.apt\_generated/\*

README.md

.gitignore

.springBeans

**alias脚本配置**

#初始化，传入版本号，从SVN下载代码，并进行构建；

alias initByVersion='function \_initByVersion() { mkdir -p /platform/lotusgo/code;cd /platform/lotusgo/code;svn co svn://192.168.1.20/platform/branches/lotusgo-$1 --username Mina\_Test --password Qe5vSPQJAxowidnS --no-auth-cache --non-interactive;cd lotusgo-$1;mvn clean;mvn install; unset -f \_initByVersion; }; \_initByVersion'

#针对工程全局maven编译

alias mvnGlobalInstall=' function \_mvnGlobalInstall() { cd /platform/lotusgo/code/lotusgo-$1;svn update;mvn clean;mvn install;unset -f \_mvnGlobalInstall; }; \_mvnGlobalInstall'

#针对公用工程的maven编译

alias mvnCommonInstall='function \_mvnCommonInstall() { cd /platform/lotusgo/code/lotusgo-$1;svn update;cd lotusgo-commons/lotusgo-commons-$2;mvn clean;mvn install; unset -f \_mvnCommonInstall; }; \_mvnCommonInstall'

#针对服务工程的maven编译

alias mvnServiceInstall='function \_mvnServiceInstall() { cd /platform/lotusgo/code/lotusgo-$1;svn update;cd lotusgo-service/lotusgo-service-$2;mvn clean;mvn install; unset -f \_mvnServiceInstall; }; \_mvnServiceInstall'

#通过docker构建单个服务镜像

alias dockerBuildSignle='function \_dockerBuildService() { cd /platform/lotusgo/code/lotusgo-$1;svn update;cd lotusgo-service/lotusgo-service-$2;mvn clean install docker:build docker:push -Dmaven.test.skip=true; unset -f \_dockerBuildService; }; \_dockerBuildService'

#通过docker-compose构建服务

alias upgradeSignleService='function \_upgradeSignleService() { sh /platform/lotusgo/bin/app.sh lotusgo-service-$1 upgrade;sh /platform/lotusgo/bin/app.sh lotusgo-service-$1 restart; unset -f \_upgradeSignleService; }; \_upgradeSignleService'

#构建单个服务

alias buildSignleService='function \_buildSignleService() { dockerBuildSignle $1 $2;upgradeSignleService $2; unset -f \_buildSignleService; }; \_buildSignleService'

#自动构建单个服务

alias autoBuildSignleService='function \_autoBuildSignleService() { mvnCommonInstall $1 tools; mvnCommonInstall $1 third; buildSignleService $1 $2; unset -f \_autoBuildSignleService; }; \_autoBuildSignleService'

alias autoBuildAllService='function \_autoBuildAllService() { buildSignleService $1 config;buildSignleService $1 registry;buildSignleService $1 gateway;buildSignleService $1 auth;buildSignleService $1 users;buildSignleService $1 goods;buildSignleService $1 order;buildSignleService $1 pay;buildSignleService $1 third; unset -f \_autoBuildAllService; }; \_autoBuildAllService'

**JVM相关查询**

jps -Vv c --查询相关java进程

jstat -gccause 23238 --查看jvmgc信息

**mongodb的安装与配置**

**docker安装mongodb**

docker pull mongo --拉取最新的mongodb镜像

--先创建容器，然后创建admin的用户跟密码

docker run -d --name mongodb -p 27017:27017 -v /yaoche/mongodb/db:/data/db mongo

--运行mongodb容器

docker exec -it mongodb bash

--创建用户，指定权限

db.createUser( {user: "admin",pwd: "gggLGZwXP6NhP1u",roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]})

--校验用户名密码是否正确，返回1表示正确

db.auth("admin","gggLGZwXP6NhP1u")

--然后停止mongodb容器

docker stop mongodb

--然后在删除mongodb容器

docker rm mongodb

--然后再指定授权模式创建mongodb

docker run -d --name mongodb -p 27017:27017 -v /yaoche/mongodb/db:/data/db mongo --auth

--运行mongodb容器

docker exec -it mongodb bash

--执行命令

mongo

--进行授权校验

db.auth("admin","gggLGZwXP6NhP1u")

--创建用户，指定权限

db.createUser({user:"itc",pwd:"caNEpMR0WtDSgqn1",roles:[{"role":"readWrite","db":"itc"}]})

--校验用户名密码是否正确，返回1表示正确

db.auth("itc","caNEpMR0WtDSgqn1")

./mongo --进入mongodb数据库

show dbs --查看所有数据库

db.createUser(mongodb,9zbByykoid86nb3H) --增加用户

docker exec -it mongodb bash

vim /etc/MongoDB.conf

db.createUser({user:'root',pwd:'9zbByykoid86nb3H',roles:['root']})

db.system.users.find()

db.auth("root","9zbByykoid86nb3H")

mongodb://root:9zbByykoid86nb3H@192.168.1.30/

db.db1.insert({"name":"菜鸟教程"})

db.db2.insert({"name":"菜鸟教程"})

db.col.insert({title: 'MongoDB 教程',

description: 'MongoDB 是一个 Nosql 数据库',

by: '菜鸟教程',

url: 'http://www.runoob.com',

tags: ['mongodb', 'database', 'NoSQL'],

likes: 100

})

db.col.insert({

title: 'MongoDB 教程',

description: 'MongoDB 是一个 Nosql 数据库',

by: '菜鸟教程',

url: 'http://www.runoob.com',

tags: ['mongodb', 'database', 'NoSQL'],

likes: 100

})

insert() 插入文档

update() 更新文档

save() 替换文档

remove() 删除文档

db.col.find().pretty() 显示所有文档

139.219.191.228 27017

root 9zbByykoid86nb3H

svn://42.159.226.200:3691/lotusgo/app/service/branches/lotus\_1.9.7.3

svn checkout svn://192.168.1.20:3691/lotusgo/app/service/branches/lotus\_1.9.7.3

svn://42.159.226.200:3691/lotusgo

**docker安装gitlab**

**1.切换阿里云的加速地址**

首先需要注册一个[阿里云](https://www.aliyun.com/)的帐号，可能还需要其他一点信息。然后进入[容器Hub服务控制台](https://cr.console.aliyun.com/?spm=5176.100239.blogcont29941.12.ZHezpK)，中间有一个加速器。我们点击它之后，阿里云会为我们创建一个专属加速器地址。

然后需要检查Docker客户端的版本，如果小于1.10，只能按照自己系统版本寻找相应的办法了。如果大于等于1.10，就可以直接使用下面的配置方法。配置方法很简单，在/etc/docker/daemon.json中添加一段配置。如果没有该文件则创建。

{
"registry-mirrors": ["<your accelerate address>"]
}

**2.下载gitlab镜像**

docker pull gitlab/gitlab-ce:latest

**3.创建容器**

docker run --detach \

--publish 8082:443 --publish 8083:80 --publish 8084:22 \

--name gitlab \

--restart always \

--volume /lotus/gitlab/config:/etc/gitlab \

--volume /lotus/gitlab/logs:/var/log/gitlab \

--volume /lotus/gitlab/data:/var/opt/gitlab \

gitlab/gitlab-ce:latest

**showdoc安装**

**Docker部署**

本文档主要介绍如何在Docker中部署ShowDoc。阅读文档前你需要对Docker有所了解。如你尚不了解Docker，请在网络上搜索相应入门教程。

**下载代码**

代码地址：<https://github.com/star7th/showdoc>

**构建**

将代码下载至某个目录，并在命令行中切换到该目录下，执行命令：

docker build -t showdoc ./

由于下载的很多依赖是在国外的，所以网络会比较慢。建议自行配置网络代理或者使用国内源。Docker的使用问题可网络上搜索相应教程。这里主要介绍ShowDoc。

**运行**

执行命令

docker run -d --name showdoc -p 4999:80 showdoc

**访问**

在浏览器中访问<http://xxxx.com:4999/install/> 进行程序的初始化(域名请改为你服务器的地址)

如你想改变端口号，可在上一步的执行命令里将4999改为其他端口。

**升级**

如果你之前已经部署过ShowDoc，可参考以下步骤进行升级：

先停止旧版的ShowDoc，按照文章开始的步骤，全新部署一个新版ShowDoc。

全新部署并初始化ShowDoc完毕后，进入之前ShowDoc的目录（/var/www/html/）。将Sqlite/sqlite.db.php（这是原来的数据库文件），以及Public/Uploads/下的所有文件（这些是上传的图片。如没有图片则可忽略之），全部复制并覆盖到新目录的相应文件。

覆盖文件后，用浏览器访问<http://xxx.com/index.php?s=/home/update/db> （请将网址更改为你服务器域名或ip）。看到OK字样便证明成功升级.

**数据与备份**

showdoc的数据库文件和代码是放在一起的。备份的时候备份整个目录（/var/www/html/）即可。删除容器的话记得也先把数据拷贝出来。如果你不放心，请在安装的时候把/var/www/html/Sqlite/showdoc.db.php 或者整份代码挂在容器外面。

docker run -d --name showdoc -p 8081:80 -v /lotus/showdoc/data/Public/Uploads/:/var/www/html/Public/Uploads/ -v /lotus/showdoc/data/Sqlite:/var/www/html/Sqlite showdoc

docker安装elasticsearch

docker pull docker.elastic.co/elasticsearch/elasticsearch:5.5.2

docker run -p 9200:9200 -e "http.host=0.0.0.0" -e "transport.host=127.0.0.1" docker.elastic.co/elasticsearch/elasticsearch:5.5.2

**更换centos yum源为阿里云源**

1.备份原来的yum源

mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup

2.下载阿里云yum源

2、下载新的CentOS-Base.repo 到/etc/yum.repos.d/

CentOS 5

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-5.repo

CentOS 6

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo

CentOS 7

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

3、之后运行yum makecache生成缓存

yum makecache

**redis安装**

通过epel安装

1.yum安装epel

yum install epel

2.yum安装redis

yum install redis

scp lotusit@192.168.1.9:/usr/java/jdk-8u121-linux-x64.gz /usr/java/

**修改linux的时区**

ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

**linux日志查找**

---手机查询会员卡

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]cardSearch[]]' |wc -l

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]cardSearch[]]' |grep \<\?xml |wc -l --成功

--促销计算

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]tr\_total[]]' |wc -l

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]tr\_total[]]' |grep 一元一积分 |wc -l --成功

--查询积分

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]GetScore[]]' |wc -l

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]GetScore[]]' |grep response |wc -l --成功

--总请求

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep request: |wc -l --总

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep request: |grep response |wc -l --失败1

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep request: |grep response |grep CommPoint2 |wc -l --失败2

--cve

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]CVE[]]' |wc -l

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]CVE[]]' |grep FirstName|wc -l --成功

--查询会员卡信息

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]GetMember[]]' |wc -l

cat /lotus/tomcat\_group/tomcat\_1/spring\_log/clpe/clpe.2017-08-18.txt |grep '[[]GetMember[]]' |grep FirstName|wc -l --成功

**docker安装redis**

--拉取redis镜像

docker pull redis

--生成redis容器

docker run --name redis -p 6379:6379 -v /yaoche/redis/data:/data -d redis redis-server --appendonly yes

java -jar /yaoche/itc/itc-cms/target/itc-cms-1.0.0-SNAPSHOT.jar &

java -jar /yaoche/itc/itc-rest/target/itc-rest-1.0.0-SNAPSHOT.jar &

java -jar /yaoche/itc/itc-task/target/itc-task-1.0.0-SNAPSHOT.jar &

**Linux 安装git以及配置码云**

安装GIT

yum -y install git

通过SSH配置码云

ssh-keygen -t rsa -C "yczl77@163.com" --生成key

cat ~/.ssh/id\_rsa.pub --查看公钥

**Linux 安装Maven**

下载安装包

wget http://mirror.bit.edu.cn/apache/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz

解压文件

tar xvf apache-maven-3.5.2-bin.tar.gz

在 /etc/profile下新增以下目录

export MAVEN\_HOME=/yaoche/maven

export PATH=${PATH}:${MAVEN\_HOME}/bin

启用环境参数

source /etc/profile