Docker安装Jenkins，部署SpringCloud项目(完整、详细)
======================================

一、 安装JDK环境
===========

路径：`/usr/lib/jdk/jdk1.8.0_201`

添加环境变量：

```
JAVA_HOME=/usr/lib/jdk/jdk1.8.0_201
PATH=$JAVA_HOME/bin:$PATH
CLASSPATH=.:$JAVA_HOME/jre/lib/ext:$JAVA_HOME/lib/tools.jar
export PATH JAVA_HOME CLASSPATH
```

二、 安装Maven环境
=============

maven路径：`/usr/lib/maven/apache-maven-3.5.3`

maven仓库路径：`/usr/lib/maven/repos`

1. 需要修改配置文件：

```
<mirror>
<id>alimaven</id>
<mirrorOf>*</mirrorOf>
<name>aliyun maven</name>
<url>http://maven.aliyun.com/nexus/content/groups/public/</url>
</mirror>
```

```
<localRepository>/usr/lib/maven/repos</localRepository>
```

2. 添加环境变量

```
vim /etc/profile

#在profile中添加如下配置：
export M2_HOME=/usr/lib/maven/apache-maven-3.5.3
export PATH=${PATH}:$JAVA_HOME/bin:$M2_HOME/bin

source /etc/profile        #重新加载配置文件
```

三、安装Jenkins、配置
==============

1. 拉取镜像

```
docker pull jenkins/jenkins:lts
```

2. 构建容器运行

```
docker run \
--name jenkins \
--memory=2048m \
-d \
-p 10240:8080 \
-u root \
-v /var/jenkins_mount:/var/jenkins_home \
-v /usr/lib/maven/apache-maven-3.5.3:/usr/local/maven \
-v /usr/lib/jdk/jdk1.8.0_201:/usr/local/jdk \
-v /etc/localtime:/etc/localtime \
jenkins/jenkins:lts
```

3. 初始登录密码

```
docker logs jenkins
```

4. 安装推荐插件（失败重试）

单独安装

`SSH`： 用来系统配置SSH remote hosts

`Git Parameter`： 用来构建时选择分支

`Publish Over SSH`：用来远端发布

5. 添加全局Credentials凭据

5.1 服务器登录凭据

![1689347783782.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/ee0482ecb1ab450895a503d13952051e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1883&h=830&s=50292&e=png&b=fefefe)

5.2 Git凭据

使用用户名密码或sshkey

![1689347911739.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/5239ae4d201c4cc6a306520f98c2dad8~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1897&h=806&s=42998&e=png&b=fefefe)

6. 系统配置 SSH remote hosts

![1689349174357.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/f99801ed77d64f9d98640af53436a70c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1842&h=865&s=51045&e=png&b=ffffff)

7.系统配置Publish over SSH（远端发布使用）

![1689921622186.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/58d90f5670bb44f49e69ea207d7108f6~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1645&h=817&s=167482&e=png&b=fefefe)

![1689921741545.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/08ff8ce4059644ee911c454ed88ef44b~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1575&h=567&s=35382&e=png&b=ffffff)

8.全局工具配置

Git不需要配置

8.1 JDK

![1689348181240.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/454be7738bcd4342bd5a77863fc10bd5~tplv-k3u1fbpfcp-jj-

mark:3024:0:0:0:q75.awebp#?w=1469&h=416&s=17777&e=png&b=fefefe)

8.2 MAVEN

![1689348246271.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/c9b26303436d447bbb0567c8c64d3c67~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1484&h=450&s=19091&e=png&b=ffffff)

四、项目构建
======

![1689646759983.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/99e28b5c6fa24ceb91185d89509118e6~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1643&h=838&s=101925&e=png&b=fbfbfb)

1. General 参数化构建

构建之后参数可以作为环境变量使用

1.1 Git参数（选择分支）

下载插件Git Parameter

![1689645510936.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/3ac748a562dc4dae8ed1d1250da50e62~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1256&h=555&s=28348&e=png&b=ffffff)

1.2 profile（选择运行环境）

添加参数，选择选项参数

![1689645627665.png](https://p1-juejin.byteimg.com/tos-cn-i-

k3u1fbpfcp/643eb46138344f11be2af68d5a39a2e2~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1216&h=453&s=17433&e=png&b=ffffff)

1.3 module（选择打包模块）

添加参数，选择选项参数

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/74af67a5d4cb46edb31588da48d7398e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1218&h=501&s=31131&e=png&b=ffffff)

1.4 选择构建后执行位置

添加参数，选择选项参数

![1689920257451.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/de994fc8226a48a2acb0180d8e02bcbb~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1220&h=450&s=19161&e=png&b=ffffff)

2. 源码管理 git

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/0b599a00152b4fdbb7ae08488c0b68aa~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1269&h=747&s=56415&e=png&b=ffffff)

3. Build Steps

3.1 调用顶层 Maven 目标

![1689646241817.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/26c1862f711b47f8bc260a55a3a68e53~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1249&h=412&s=21652&e=png&b=ffffff)

```
-U
-pl ${module}
-am
-P ${env}
clean
-Dmaven.test.skip=true
package
install
```

参数	全称	说明
-pl	—projects	选项后可跟随{groupId}:{artifactId}或者所选模块的相对路径(多个模块以逗号分隔)
-am	—also-make	表示同时处理选定模块所依赖的模块，向下处理
-amd	—also-make-dependents	表示同时处理依赖选定模块的模块，向上处理
-N	—Non-recursive	表示不递归子模块
-rf	—resume-from	表示从指定模块开始继续处理
-DskipTests		跳过测试
-U		全部更新代码

3.2 Execute shell script on remote host using ssh

执行服务器中脚本

`/home/../shell/run.sh ${module} ${env} ${location}`

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/59dbe06442684af1a06309bed963348c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1249&h=354&s=23693&e=png&b=ffffff)

3.3 Send files or execute commands over SSH

需要提前安装插件`Publish Over SSH`

![1689920680845.png](https://p9-juejin.byteimg.com/tos-cn-i-

k3u1fbpfcp/b01bd5c3b8d049e39046cd37a2b283b4~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1239&h=790&s=65776&e=png&b=ffffff)

```
Source files：准备发送的文件，该文件是相对于这个项目的workspace目录，也就是$JENKINS_HOME/workspace/xxxx/
例如要发送$JENKINS_HOME/workspace/lumitech-dev/shumiweb/target/shumiweb-1.0.0-SNAPSHOT.jar到目标目录，则设置Source files为shumiweb/target/shumiweb-1.0.0-SNAPSHOT.jar

Remove prefix：例如设置为shumiweb/target，如果不填会创建shumiweb/target/目录层级。
```

更多示例见官网：[wiki.jenkins.io/display/JEN…](http://cxyroad.com/”https://wiki.jenkins.io/display/JENKINS/Publish+Over#PublishOver-examples”)

4. 构建

![1689922036191.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1154b0edca314725941afc471458440a~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1894&h=913&s=88738&e=png&b=ffffff)

★5. 说明

打包都是在Jenkins服务器进行的，

选择local代表在Jenkins所在服务器发布，

选择remote代表在把jar包发送到远端服务器，在远端服务器进行发布。

需要在服务器上提前准备好shell脚本

脚本位置：`/home/../shell/run.sh`

 `/home/../shell/rollback.sh`

 `/home/../shell/backup.sh`

jar包位置：`/home/../xxx-${env}/web-jar/`

备份jar包位置：`/home/../xxx-${env}/backup/`

五、shell脚本
=========

1. run.sh

```
module=$1                                   #模块名
env=$2                                      #运行环境
location=$3                                 #执行位置 local本地执行 remote远端执行
WEB_PATH=’/home/../xxx-’$env           #项目根路径(路径自行定义)
JAR_PATH=$WEB_PATH’/web-jar/’          #jar包路径
LOG_PATH=$WEB_PATH’/web-jar/’          #log路径
SHELL_PATH=$WEB_PATH’/shell/’          #shell脚本路径
BACKUP_PATH=$WEB_PATH’/backup/’        #备份jar包路径
APP_NAME=$module’-1.0.0-SNAPSHOT.jar’  #当前项目名称
DES_APP=$JAR_PATH$APP_NAME             #jar包全路径
DES_LOG=$LOG_PATH’nohup-’$module’.out’ #nohup全路径
PROJECT_NAME=”xxxxx”                   #项目名称

#打印路径
echo [INFO]’$APP_NAME’:$APP_NAME
echo [INFO]’$WEB_PATH:’$WEB_PATH
echo [INFO]’$BACKUP_PATH:’$BACKUP_PATH
echo [INFO]’$DES_APP:’$DES_APP
echo [INFO]’$DES_LOG:’$DES_LOG

# 创建目录
mkdir -p $WEB_PATH
mkdir -p $JAR_PATH
mkdir –p $LOG_PATH
mkdir –p $BACKUP_PATH

#kill进程
killProgress() {
  pid=$(ps –ef | grep $DES_APP | grep –v grep | awk ’{print $2}’)
  if [ –n ”$pid” ]; then
    kill –9 $pid
    echo [INFO] ”>>>>>>>>>>> ”$DES_APP” ”$pid” has bean killed <<<<<<<<<<<”
  fi
}

# 备份
fileSize=$(du –s $BACKUP_PATH) #项目所在路径的大小
if [[ ${fileSize%%/*} –ne 0 ]]; then
  echo [INFO] ”>>>>>>>>>>> 开始备份 <<<<<<<<<<<”
  echo ’backup file –>’$BACKUP_PATH$APP_NAME
  cp –r $DES_APP $BACKUP_PATH
  echo [INFO] ”>>>>>>>>>>> 结束备份 <<<<<<<<<<<”
fi

#本地才执行该操作
if [[ $location = ”local” ]]; then
  # 删除应用服务器旧的代码
  rm –rf $DES_APP
  echo [INFO] ”>>>>>>>>>>> 原jar包已删除 <<<<<<<<<<<”

  # 拷贝jar包
  cp /var/jenkins_mount/workspace/$PROJECT_NAME/$module/target/$APP_NAME $JAR_PATH
  echo [INFO] ”>>>>>>>>>>> jar包拷贝成功 <<<<<<<<<<<”
fi

#kill进程
killProgress

#运行jar包
echo [INFO] ”>>>>>>>>>>> 开始启动项目 <<<<<<<<<<<”
source /etc/profile
echo ”” >$DES_LOG
nohup java –jar $DES_APP >>$DES_LOG 2>&1 &

# 日志监测部分
echo [INFO] ”>>>>>>>>>>>> 日志所在位置”$DES_LOG” 开始输出日志 <<<<<<<<<<<”
sleep 2
```

```
second=0
lastShellRow=0
while true; do
  nowShellRow=$(cat $DES_LOG | wc –l)
  showRow=$((nowShellRow – lastShellRow))
  if [[ $showRow –ge 0 ]]; then
    tail –n$showRow $DES_LOG
  fi
  sum=$(cat $DES_LOG | grep ’Started application in’ | wc –l)
  sum_jvm=$(cat $DES_LOG | grep ’JVM running for’ | wc –l)
  failed_sign=$(cat $DES_LOG | grep ’Application run failed\|APPLICATION FAILED TO START’ | wc –l)
  if [ $second –ge 800 ]; then
    echo [ERROR] ”>>>>>>>>>>>> 部署等待时间过长 退出部署 <<<<<<<<<<<<<”
    killProgress

    echo [INFO] ”>>>>>>>>>>>> 准备回退版本 <<<<<<<<<<<<”
    bash /home/shumi/shell/rollback.sh $APP_NAME $JAR_PATH $BACKUP_PATH $DES_LOG

    exit 1
    break
  fi
  if [ $sum –gt 0 ] || [ $sum_jvm –gt 0 ]; then
    echo [INFO] ”>>>>>>>>>>>> 项目启动花费 $second 秒 <<<<<<<<<<<<”
    break
  fi
  if [[ $failed_sign –gt 0 ]]; then
    echo [INFO] ”>>>>>>>>>>>> 项目启动花费 $second 秒 <<<<<<<<<<<<”
    echo [ERROR] ”>>>>>>>>>>>>> 项目启动失败 <<<<<<<<<<<<<”
    killProgress

    echo [INFO] ”>>>>>>>>>>>> 准备回退版本 <<<<<<<<<<<<”
    bash /home/shumi/shell/rollback.sh $APP_NAME $JAR_PATH $BACKUP_PATH $DES_LOG

    exit 1
    break
  fi
  second=$((second + 2))
  lastShellRow=$nowShellRow
  sleep 2
done
```

2.rollback.sh

```
APP_NAME=$1     #jar包名
JAR_PATH=$2    #web–jar路径
BACKUP_PATH=$3 #备份路径
DES_LOG=$4      #nohup.out位置

echo [INFO]’$APP_NAME’:$APP_NAME
echo [INFO]’$JAR_PATH:’$JAR_PATH
echo [INFO]’$BACKUP_PATH:’$BACKUP_PATH
echo [INFO]’$DES_LOG:’$DES_LOG

#kill进程
killProgress(){
pid=$(ps –ef | grep $DES_APP | grep –v grep | awk ’{print $2}’)
if [ –n ”$pid” ]; then
  kill –9 $pid
  echo [INFO] ”>>>>>>>>>>> ”$DES_APP” ”$pid” has bean killed <<<<<<<<<<<”
fi
}

#备份jar包移动回去
rm –rf $JAR_PATH$APP_NAME
cp $BACKUP_PATH$APP_NAME $JAR_PATH


echo [INFO] ”>>>>>>>>>>>> 开始回滚 <<<<<<<<<<<”
source /etc/profile
echo ”” >$DES_LOG
nohup java –jar $JAR_PATH$APP_NAME >>$DES_LOG 2>&1 &

echo [INFO] ”>>>>>>>>>>>> 日志所在位置”$DES_LOG” 开始输出日志 <<<<<<<<<<<”
# 日志监测部分
sleep 2
second=0
lastShellRow=0
while true; do
  nowShellRow=$(cat $DES_LOG | wc –l)
  showRow=$((nowShellRow – lastShellRow))
  if [[ $showRow –ge 0 ]]; then
    tail –n$showRow $DES_LOG
  fi
  sum=$(cat $DES_LOG | grep ’Started application in’ | wc –l)
  sum_jvm=$(cat $DES_LOG | grep 'JVM running for' | wc -l)
  failed_sign=$(cat $DES_LOG | grep 'Application run failed\|Failed to\|APPLICATION FAILED TO START' | wc -l)
  if [ $second -ge 800 ]; then
    echo [ERROR] ">>>>>>>>>>> 部署等待时间过长 退出部署 <<<<<<<<<<<"
    killProgress
    exit 1
    break
  fi
  if [ $sum -gt 0 ] || [ $sum_jvm -gt 0 ]; then
    echo [INFO] ">>>>>>>>>>> 项目回退版本启动花费 $second 秒 <<<<<<<<<<<"
    echo [INFO] ">>>>>>>>>>> 项目回退版本启动成功 <<<<<<<<<<<"
    break
  fi
  if [[ $failed_sign -gt 0 ]]; then
    echo [INFO] ">>>>>>>>>>> 项目回退版本启动花费 $second 秒 <<<<<<<<<<<"
    echo [ERROR] ">>>>>>>>>>> 项目回退版本启动失败 <<<<<<<<<<<"
    echo [INFO] ">>>>>>>>>>> 请手动操作回退版本！ <<<<<<<<<<<"
    killProgress
    exit 1
    break
  fi
  second=$((second + 2))
  lastShellRow=$nowShellRow
  sleep 2
done
```

3.backup.sh

```
module=$1                                   #模块名
env=$2                                      #运行环境
WEB_PATH='/home/../xxx-'$env        #项目根路径
JAR_PATH=$WEB_PATH'/web-jar/'           #jar包路径
LOG_PATH=$WEB_PATH'/web-jar/'           #log路径
SHELL_PATH=$WEB_PATH'/shell/'           #shell脚本路径
BACKUP_PATH=$WEB_PATH'/backup/'         #备份jar包路径
APP_NAME=$module'-1.0.0-SNAPSHOT.jar'  #当前项目名称
DES_APP=$JAR_PATH$APP_NAME              #jar包全路径
```

```
# 创建目录
mkdir -p $WEB_PATH
mkdir -p $JAR_PATH
mkdir -p $BACKUP_PATH

# 备份
fileSize=$(du -s $BACKUP_PATH) #项目所在路径的大小
if [[ ${fileSize%%/*} -ne 0 ]]; then
  echo [INFO] ”>>>>>>>>>>> 开始备份 <<<<<<<<<<”
  echo ’backup file ->’$BACKUP_PATH$APP_NAME
  cp -r $DES_APP $BACKUP_PATH
  echo [INFO] ”>>>>>>>>>>> 结束备份 <<<<<<<<<<”
fi
```

六、使用方法
======

[http://ip:10240/](http://cxyroad.com/ ”http://ip:10240/”)

用户名：`admin`
密码：`xxx`

1. 选择项目

![image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/4e325905e7f94ea3a1f51ed6b4d9cfc2~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1900&h=624&s=90799&e=png&b=ffffff)

2. 构建

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/29967be4f3e14cec852868a388c9a911~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1895&h=651&s=67982&e=png&b=ffffff)

3. 选择参数

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/75b3b7ee9cee4dc189a953d9329b1acd~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1876&h=899&s=101656&e=png&b=ffffff)

4. 查看构建日志

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/07306f0fff71449ca4b1a3b4438f02b5~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1893&h=638&s=69242&e=png&b=ffffff)

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1fb43c32e02f4e49af823eca59dc1106~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1896&h=818&s=215992&e=png&b=f5f5f5)

PS
==

shell脚本或者其他地方，涉及到路径相关，灵活修改即可。

使用docker-compose部署springcloud项目，后续更新...

原文链接: https://juejin.cn/post/7359157380045750299