

持续集成与docker配置 (章节)

学习目标

1. 场景分析 (章导学)
 - 1.1 使用场景
- 2 持续集成Jenkins及Docker配置 (小节)
 - 2.1 介绍 (知识点)
 - 2.2 安装及配置 (知识点)
 - 2.2.1 Docker安装及配置
 - 2.2.2 Jenkins安装及配置
 - 2.2.3 Docker简单使用
3. Jenkins插件安装及配置 (小节)
 - 3.1 安装Allure 插件 (知识点)
 - 3.2 配置Allure Commandline (知识点)
 - 3.3 Jenkins 配置 git (知识点)
4. Jenkins 配置及运行
 - 4.1 创建Jenkins项目
 - 4.2 配置Jenkins项目
5. 章总结
 - 重难点
 - 拓展延伸

持续集成与docker配置 (章节)

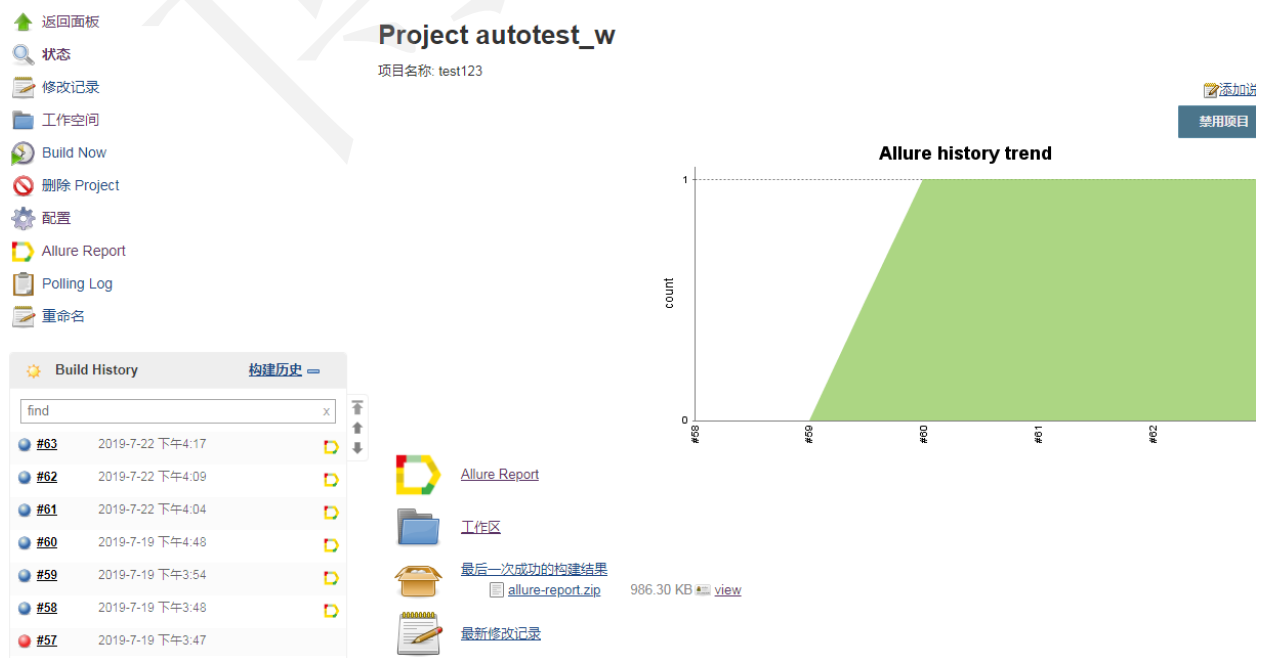
学习目标

- 持续集成Jenkins及Docker介绍及安装【重点】
- 持续集成Jenkins自动化测试配置及运行【重点】

1, 场景分析 (章导学)

1.1 使用场景

- 使用Jenkins实现自动/定时运行接口测试脚本，并生成对应的测试报告



2 持续集成Jenkins及Docker配置 (小节)

2.1 介绍 (知识点)

2.1.1 Jenkins介绍

Jenkins是一个开源软件项目，是基于Java开发的一种工具，可扩展的持续集成、交付、部署（软件/代码的编译、打包、部署）的基于web界面的平台。允许持续集成和持续交付项目，无论用的是什么平台，可以处理任何类型的构建或持续集成。

- 官网: <https://jenkins.io/>
- 官方文档: <https://jenkins.io/zh/doc/>



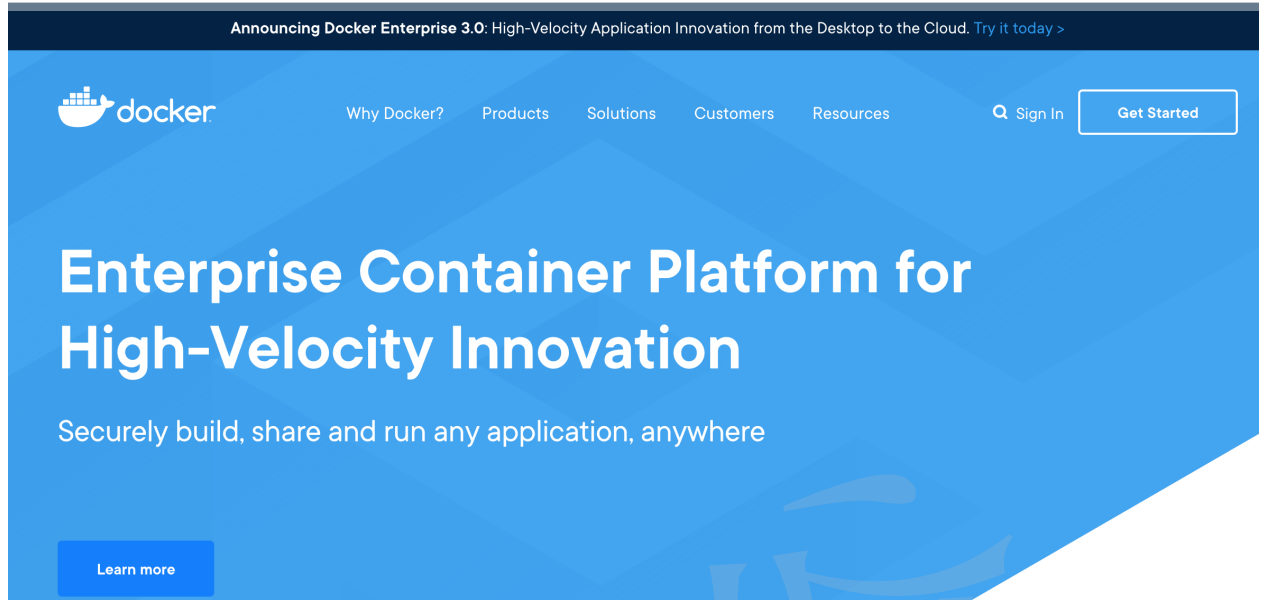
Jenkins特点

- 开源免费
- 安装配置超级简单
- 跨平台，支持所有的平台
- web形式的可视化的管理页面
- 分布式构建
- 丰富的插件支持

2.1.2 Docker介绍

Docker 是一个[开源](https://www.docker.com/)的应用容器引擎，方便快速搭建环境、部署应用

- 官网: <https://www.docker.com/>



Docker应用场景

- 应用的自动化打包和发布
- 自动化测试和持续集成、发布。

2.2 安装及配置 (知识点)

2.2.1 Docker安装及配置

- Windows安装
- CentOS安装

CentOS安装

Docker支持以下的CentOS版本：

- CentOS 7 (64-bit)
- CentOS 6.5 (64-bit) 或更高的版本

安装及配置步骤

1. 查看内核版本`uname -a`,建议3.10以上
2. yum更新`yum update`
3. 安装需要的软件包

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

4. 设置yum源

```
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

5. 查询版本，并安装

```
yum list docker-ce --showduplicates | sort -r
```

6. 安装Docker

```
yum install docker-ce-17.12.1.ce
```

7. 验证安装是否成功

```
docker version
```

8. 启动docker,并加入开机启动

```
systemctl start docker  
systemctl enable docker
```

```
[root@localhost itcast] # systemctl enable docker  
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service  
to /usr/lib/systemd/system/docker.service.  
[root@localhost itcast] # docker version  
Client:  
Version:      18.03.1- ce  
API version:  1.37  
Go version:   gol.9.5  
Git commit:   9ee9f40  
Built:        Thu Apr 26 07:20:16 2018  
OS/Arch:      linux/amd64  
Experimental: false  
Orchestrator: swarm  
  
Server:  
Engine:  
Version:      18.03.1- ce  
API version:  1.37 (minimum version 1.12)  
Go version:   gol.9.5  
Git commit:   9ee9f40  
Built:        Thu Apr 26 07:23:58 2018  
OS/Arch:      linux/amd64  
Experimental: false  
[root@localhost itcast] #
```

2.2.2 Jenkins安装及配置

安装及配置步骤:

1. 下载地址: <https://jenkins.io/zh/download/>
2. 选择docker版本下载

https://jenkins.io/zh/download/

kins

博客 文档 插件 社区 子项目 关于

长期支持版本 (LTS)

LTS (长期支持) 版本每12周从常规版本流中选择，作为该时间段的稳定版本。 [了解更多...](#)

[变更日志](#) | [升级指南](#) | [以前的发行版](#)

部署 Jenkins 2.176.2



下载 Jenkins 2.176.2 for:

Docker
FreeBSD
Gentoo
Mac OS X
OpenBSD
openSUSE

每周更新版

每周都会发布一个新版本，为用户和插件开发人员提供错误修复和功能。

[变更日志](#) | [以前的发行版](#)

下载 Jenkins 2.189 for:

Arch Linux
Docker
FreeBSD
Gentoo
Mac OS X
OpenBSD
openSUSE
Red Hat/Fedora/CentOS


3. 或直接访问<https://hub.docker.com/r/jenkins/jenkins>

← → × https://hub.docker.com/r/jenkins/jenkins ☆

Overview Tags

Jenkins Continuous Integration and Delivery server.

This is a fully functional Jenkins server, based on the weekly and LTS releases.



Jenkins


- To use the latest LTS: `docker pull jenkins/jenkins:lts`
- To use the latest weekly: `docker pull jenkins/jenkins`
- Lighter alpine based image also available

[Read documentation for usage](#)

Docker Pull Command

```
docker pull jenkins/jenkins
```

Owner

 jenkins

4. Centos下安装

◦ 切换root用户

`su root`

◦ 执行命令

`docker pull jenkins/jenkins:lts`

```
[itcast@localhost ~]$ su
```

密码：

```
[root@localhost itcast]# docker pull jenkins/jenkins:lts
```

◦ 安装完成

```
[root@localhost itcast]# docker pull jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
a4d8138d0f6b: Downloading 22.04MB/45.34MB
dbdc36973392: Download complete
a4d8138d0f6b: Pull complete
dbdc36973392: Pull complete
f59d6d019dd5: Pull complete
aaef3e026258: Pull complete
5e86b04a4500: Pull complete
1a6643a2873a: Pull complete
eabf1bdcaba7: Pull complete
4b6fa481404f: Pull complete
9df65d190912: Pull complete
914163c93632: Pull complete
6e4f672a31eb: Pull complete
c3f2013d010d: Pull complete
00a23be5fe00: Pull complete
9e52e93edd76: Pull complete
be52efd60529: Pull complete
0bd368f9cf88: Pull complete
f4e0474291c0: Pull complete
1acfa66cb7b6: Pull complete
ab0a1408265b: Pull complete
Digest: sha256:5a1ac8381051dc430dcc90077f648ed1e1b91149b5808a8dea507639eefc563f
Status: Downloaded newer image for jenkins/jenkins:lts
```

◦ 查看下载完成的镜像

```
[root@localhost itcast]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
jenkins/jenkins	lts	b137a5753eb1	3 weeks ago	567MB

```
[root@localhost itcast]#
```

◦ 启动Jenkins镜像

```
docker run -d -p 80:8080 -p 50000:50000 -v jenkins:/var/jenkins_home -v /etc/localtime:/etc/localtime --name jenkins docker.io/jenkins/jenkins:lts
```

■ 启动参数意义

- d 后台运行镜像
- p 80:8080 将镜像的8080端口映射到服务器的80端口
- p 50000:50000 将镜像的50000端口映射到服务器的50000端口
- v jenkins:/var/jenkins_home /var/jenkins_home目录为jenkins工作目录，我们将硬盘上的一个目录挂载到这个位置，方便后续更新镜像后继续使用原来的工作目录。
- v /etc/localtime:/etc/localtime 让容器使用和服务器同样的时间设置。
- name jenkins 给容器起一个别名

◦ 浏览<http://localhost>并等到Unlock Jenkins页面出现。

- 注意，docker版的jenkins，所以这个路径不是服务器的路径，而是容器内的路径。我们可以通过如下命令从容器中获取到初始化密码

```
docker exec jenkins tail /var/jenkins_home/secrets/initialAdminPassword
```

◦ 继续使用后面步骤设置向导完成设置。

2.2.3 Docker简单使用

- 列出本地主机上的镜像

```
docker images
```

- 查看容器状态

```
docker ps -a
```

- 容器启动命令

```
Docker start container-name(jenkins)
Docker stop container-name
docker restart container-name
```

- 进入正在运行的镜像

docker exec : 在运行的容器中执行命令
语法
-d :分离模式: 在后台运行
-i :即使没有附加也保持STDIN 打开
-t :分配一个伪终端

- 容器内安装命令

- 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。

- 等更新完毕以后再敲命令，即可

```
apt-get install vim
```

FAQ:

1. docker中开放端口IPv4 forwarding is disabled. Networking will not work解决方法

- 编辑 vi /etc/sysctl.conf
- 添加 net.ipv4.ip_forward=1
- 重启network服务 systemctl restart network

2. 处理docker下载慢

- 编辑docker配置文件vi /lib/systemd/system/docker.service
- 将这一行: ExecStart=/usr/bin/dockerd
- 修改为: ExecStart=/usr/bin/dockerd --registry-mirror=<https://u1qbyfsc.mirror.aliyuncs.com>

3. 开放防火墙端口

- 查看已经开放的端口

```
firewall-cmd --list-ports
```

- 开启端口

```
firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

命令含义:

```
-zone #作用域
-add-port=80/tcp #添加端口, 格式为: 端口/通讯协议
-permanent #永久生效, 没有此参数重启后失效
```

```
firewall-cmd --reload
```

#重新载入firewall配置, 使配置生效

4. Docker容器不能访问网络

- 执行命令

#nmcli是redhat7或者centos7之后的命令, 该命令可以完成网卡上所有的配置工作, 并且可以写入配置文件, 永久生效

```
nmcli connection modify docker0 connection.zone trusted
```

#重启docker服务

```
systemctl restart docker.service
```

- 进入容器终端并且的保留为容器终端的输入形式(-it和bash的结合作用)

- `docker exec -it CONTAINER_ID bash`

- 使用ping命令, 验证网络是否连通

```
ping www.baidu.com
```

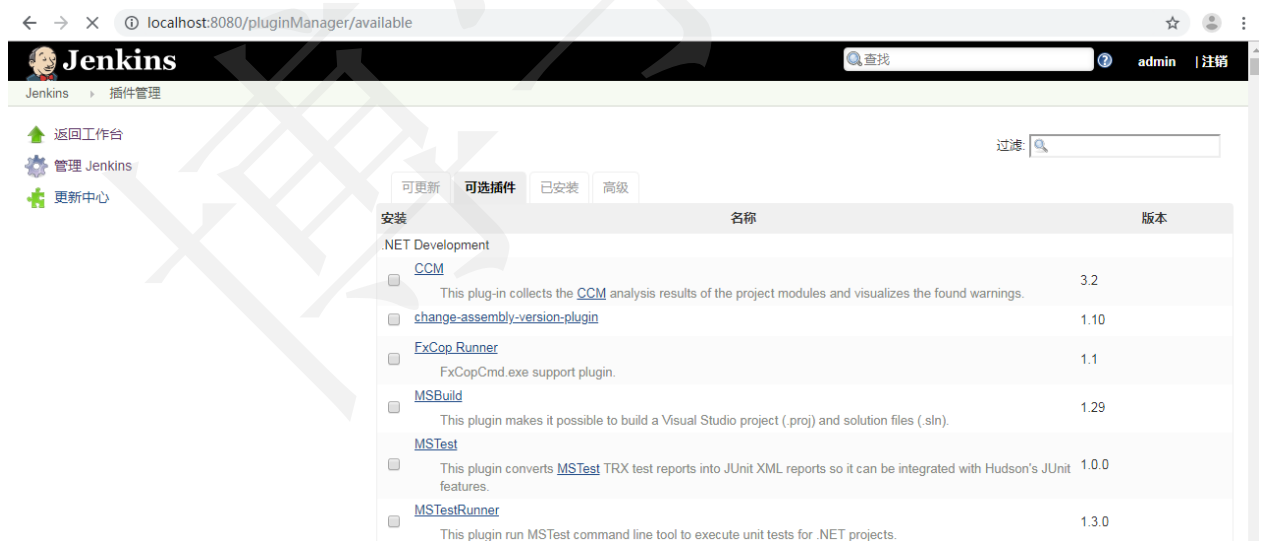
3. Jenkins插件安装及配置 (小节)

3.1 安装Allure 插件 (知识点)

安装步骤

1. 进入jenkins->系统管理 ->管理插件->可选插件

- 直接输入地址: <http://localhost:8080/pluginManager/available>



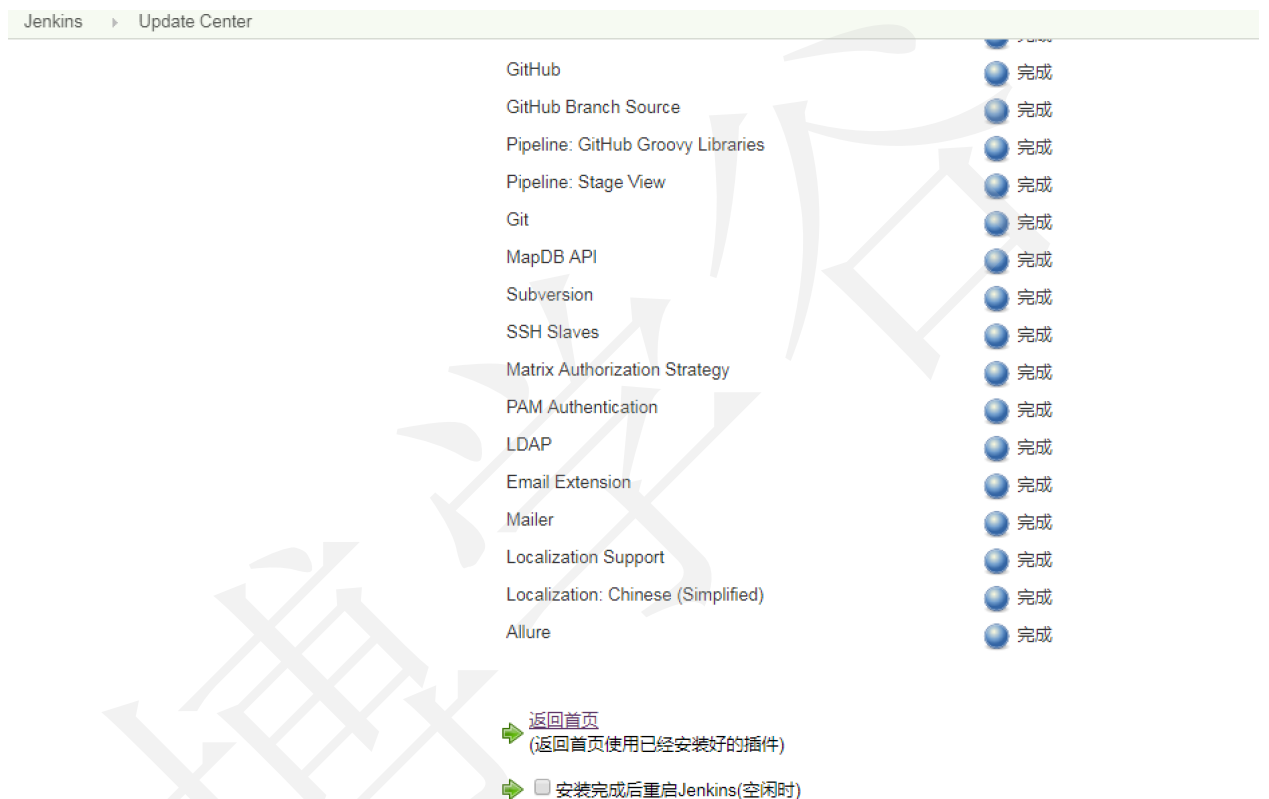
2. 搜索框中输入 allure



3. 第一条选中点击 直接安装 即可



4. 安装完成



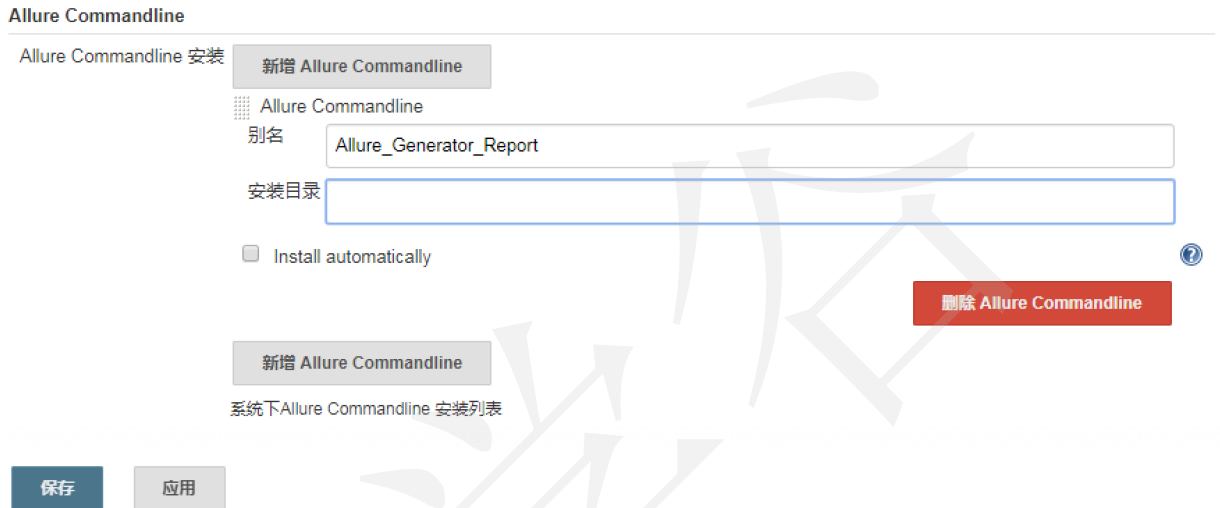
3.2 配置Allure Commandline (知识点)

安装步骤

1. 确保 allure 插件已经安装成功
2. Manage Jenkins——>Global Tool Configuration——>Allure Commandline 部分，点击添加Allure Commandline



3. 填写name(可以随便取名，主要是用来标识),可以选择自动安装或指定安装目录



4. 如果指定输入安装目录，需要下载allure

- 下载 allure.zip 下载地址: <https://dl.bintray.com/qameta/generic/io/qameta/allure/allure/2.7.0>
- 解压到本地目录、配置allure.bat的环境变量Path；在命令行中运行allure，确认环境变量配置成功



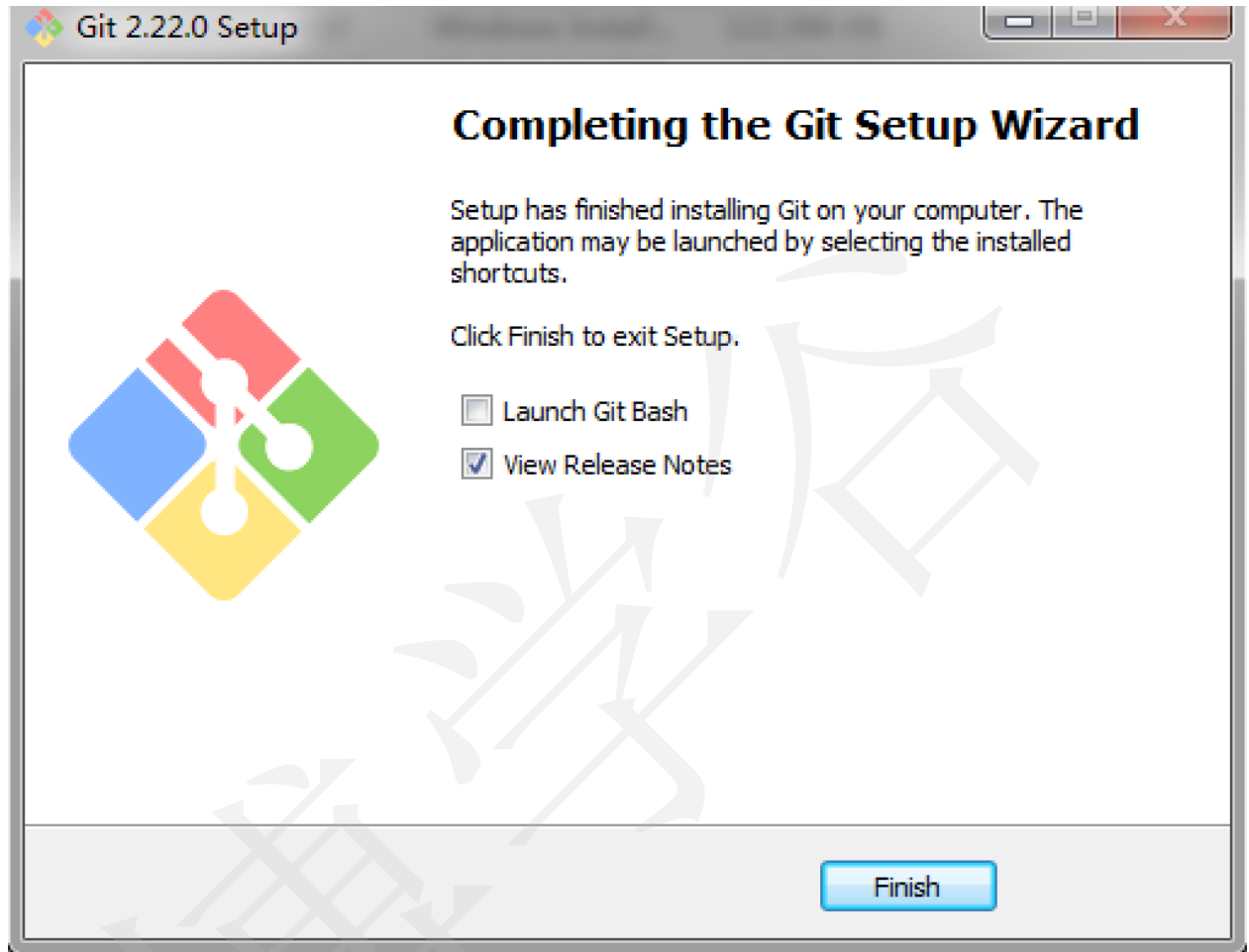
3.3 Jenkins 配置 git (知识点)

安装步骤

下载地址: <https://git-scm.com/downloads>

- Git安装
 - windows

双击安装包，一路Next，安装完成



- linux
 - `yum install git`

- Git配置
 - linux默认已安装完，不需修改配置，如果windows则需要配置
 - Manage Jenkins——>Global Tool Configuration——>Git
 - 点击添加

Maven 配置

默认 settings 提供

Use default maven settings

默认全局 settings 提供

Use default maven global settings

JDK

JDK 安装

新增 JDK

系统下JDK 安装列表

Git

Git installations

Git

Name

git

Path to Git executable

C:\Program Files\Git\bin

☐ Install automatically

Add Git

保存

应用

hinet 的响应

4. Jenkins 配置及运行

4.1 创建Jenkins项目

步骤:

1. 进入Jenkins主页
2. 点击新建
3. 输入项目名称，并点击构建一个自由风格的软件项目。
4. 点击确定

4.2 配置Jenkins项目

General配置

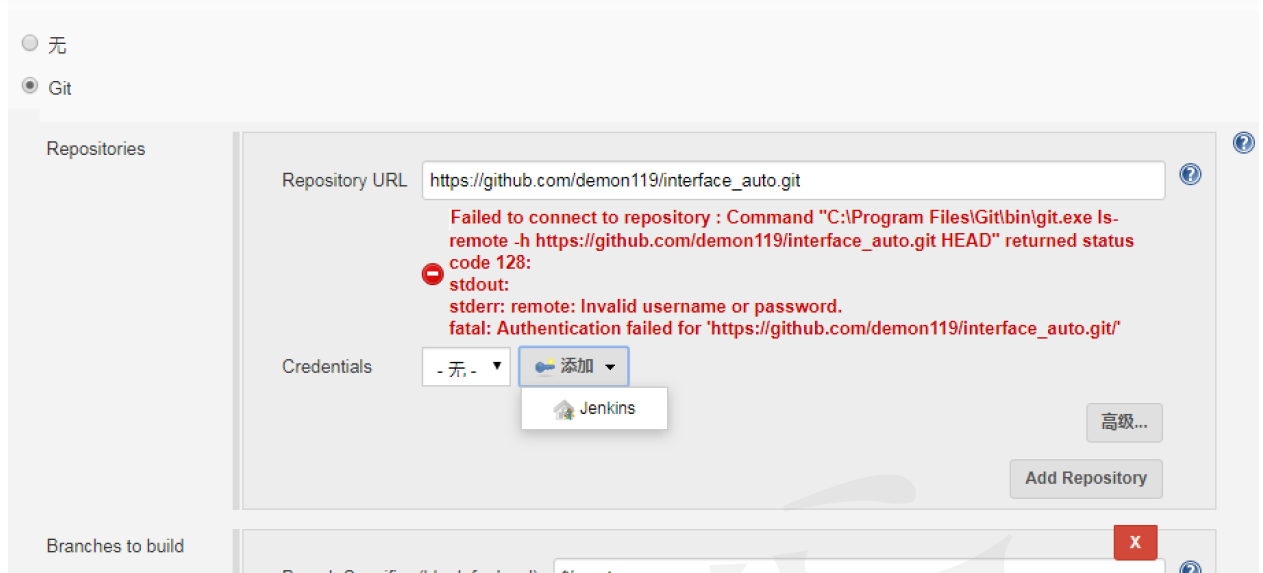
步骤:

- 勾选 GitHub project，输入 Project url

源码管理配置

步骤:

1. 勾选 Git，输入 Repository URL
2. 点击 Add，点击 Jenkins
3. 输入 github 用户名和密码，点击Add



构建触发器 配置

步骤:

- 勾选 Poll SCM, 日程表中填写 */1 * * * *



构建 配置

1. 根据平台选择执行的命令类型

- Windows
 - Execute Windows batch command
- Linux/mac
 - Execute shell

2. 编写代码

- Windows

```
#python安装版本尽量与代码编写版本一致
PATH=$PATH;C:\python37\Scripts;E:\tools\allure-2.7.0\bin
pip install -r requirements.txt
C:\python37\python run.py
```

* Linux

#python安装版本尽量与代码编写版本一致，由于linux自带安装的python2,需要单独安装python3
如果不识别pip或python,则需要使用绝对路径来调用

```
pip3 install -r requirements.txt  
python3 run.py  
````
```

- requirements文件生成
- # 使用以下命令会在项目根目录产生一个 requirements.txt 文件  
# 该文件会列出本虚拟空间中所使用的所有包及其版本  
pip freeze > requirements.txt  
  
# 使用以下命令会安装项目根目录的 requirements.txt 文件中所有的依赖的指定版本  
pip install -r requirements.txt

## 构建后操作-报告 配置

步骤:

1. 点击 增加构建后步骤，选择Allure Report
2. 在 path 中输入 allure 报告的 xml 所在的目录名称

report/result

## 构建执行

配置到报告步骤后，点击保存。此时，我们可以构建这个项目，所谓构建也就是运行的意思。

- 自动构建
- 手动构建

## 邮件

步骤:

1. Jenkins->配置->邮件通知
2. 设置相关信息

|                                              |                                               |   |
|----------------------------------------------|-----------------------------------------------|---|
| SMTP服务器                                      | <input type="text" value="smtp.itcast.cn"/>   | ? |
| 用户默认邮件后缀                                     | <input type="text" value="@itcast.cn"/>       | ? |
| <input checked="" type="checkbox"/> 使用SMTP认证 |                                               | ? |
| 用户名                                          | <input type="text" value="wangxu@itcast.cn"/> |   |
| 密码                                           | <input type="password" value="....."/>        |   |
| 使用SSL协议                                      | <input type="checkbox"/>                      | ? |
| SMTP端口                                       | <input type="text" value="25"/>               | ? |
| Reply-To Address                             | <input type="text"/>                          |   |
| 字符集                                          | <input type="text" value="UTF-8"/>            |   |
| <input type="checkbox"/> 通过发送测试邮件测试配置        |                                               |   |

#### Allure Report

Properties

新增

保存

应用

## 5. 章总结

### 重难点

- Jenkins插件安装及配置【重点】【难点】
- Jenkins配置及运行【重点】【难点】
- 【易错点】：编写shell时，一定要声明路径或者是全路径

### 拓展延伸

**思考：**请思考一下，Jenkins邮件是否可以配置多个人进行接收。(必须提供)

**延伸资料：**给和课程相关的文章链接。(可选)

Windows Docker安装 <https://www.runoob.com/docker/windows-docker-install.html>