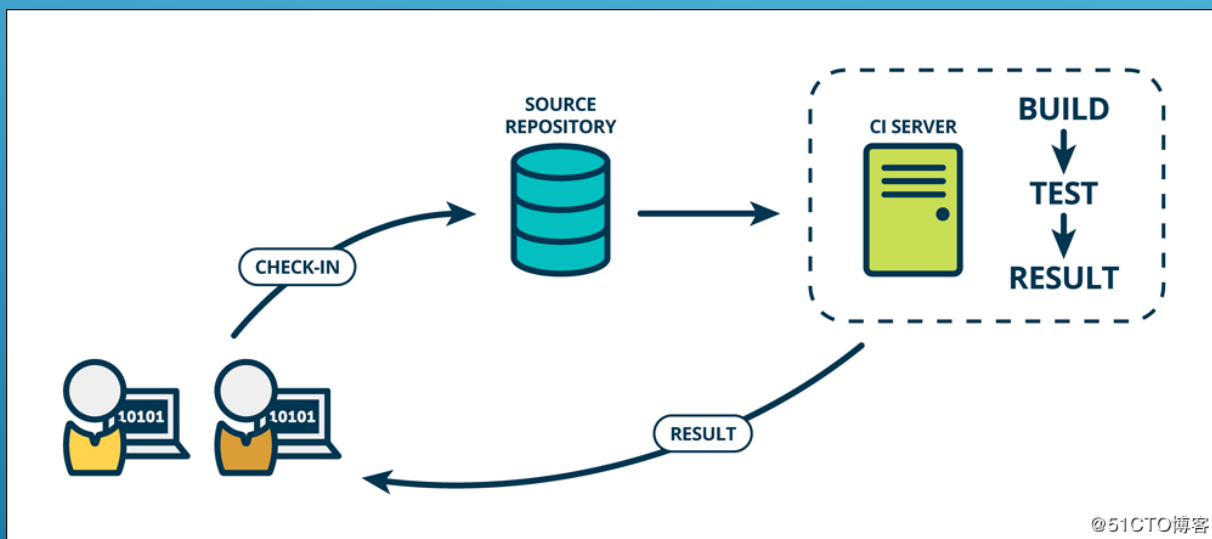


JENKINS部署

ShenJet

<https://github.com/ShenJet>

- ▶ 持续集成是一种软件开发实践，即团队开发成员经常集成他们的工作，通常每个成员至少集成一次，也就意味着每天可能会发生多次集成。每次集成都通过自动化的构建（包括编译，发布，自动化测试）来验证，从而尽快地发现集成错误。许多团队发现这个过程可以大大减少集成的问题，让团队能够更快的开发内聚的软件。



CONTINUOUS INTEGRATION (CI)

- ▶ Jenkins只是一个平台，真正运作的都是插件。这就是jenkins流行的原因，因为jenkins什么插件都有
- ▶ Hudson是Jenkins的前身，是基于Java开发的一种持续集成工具，用于监控程序重复的工作，Hudson后来被收购，成为商业版。后来创始人又写了一个jenkins，jenkins在功能上远远超过hudson
- ▶ Jenkins官网：<https://jenkins.io/>
- ▶ Jenkins下载：<http://updates.jenkins-ci.org/>
- ▶ jenkins的全部镜像：<http://mirrors.jenkins-ci.org/status.html>

JENKINS介绍

- ▶ 1.Centos 7 (ubuntu、docker等都可以)
- ▶ 2.JDK

环境要求

- ▶ 1.下载对应版本的rpm包：<https://jenkins.io/download/>
- ▶ 2.安装rpm包：`rpm -ivh [包名]`
- ▶ 3.安装完最好reboot一下
- ▶ 4.启动：`$ service jenkins start` (默认启动在8080端口)
- ▶ 5.启动后看下端口是否监听：`$ netstat -ano | grep LISTEN`

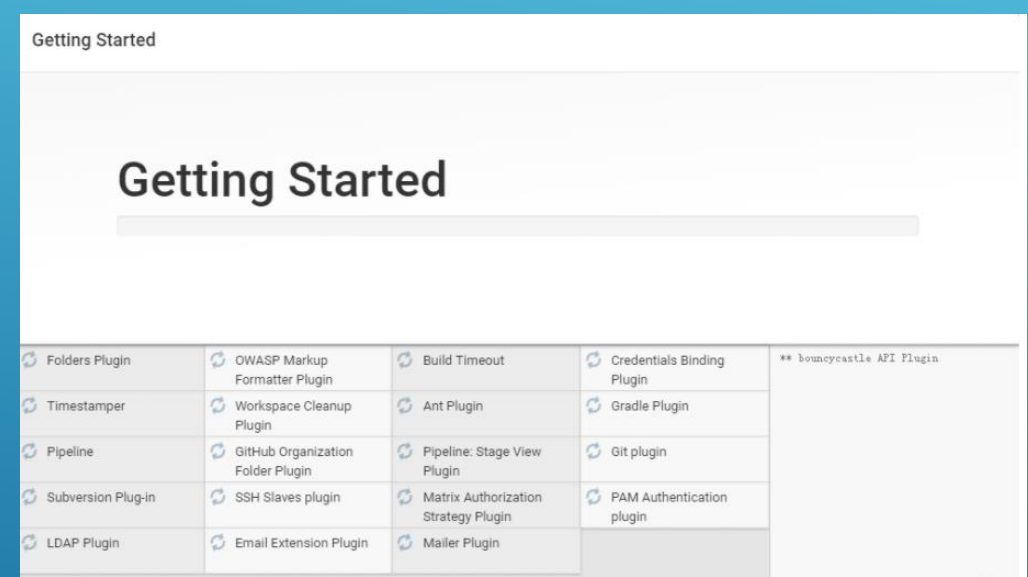
JENKINS部署安装

- ▶ 1. `cat /var/lib/jenkins/secrets/initialAdminPassword`
- ▶ 获取初始登录密钥
- ▶ 2. 登录进系统



JENKINS开始使用1

可以选择系统推荐的插件，直接自动安装
后续可以按需追加安装其他模块



JENKINS开始使用2

- ▶ 前端项目最常用的是js环境以及若干npm相关命令,
- ▶ 因此, 第一步就是要给jenkins追加nodejs插件
- ▶ 首页->系统设置->全局工具配置:

NodeJS

NodeJS 安装

新增 NodeJS

NodeJS

别名

☒ 自动安装

Install from nodejs.org

Version

Force 32bit architecture ☐

Global npm packages to install

Global npm packages refresh hours

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax 'packageName@version'

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

新增安装

删除安装

删除 NodeJS

新增 NodeJS

系统下NodeJS 安装列表

Docker

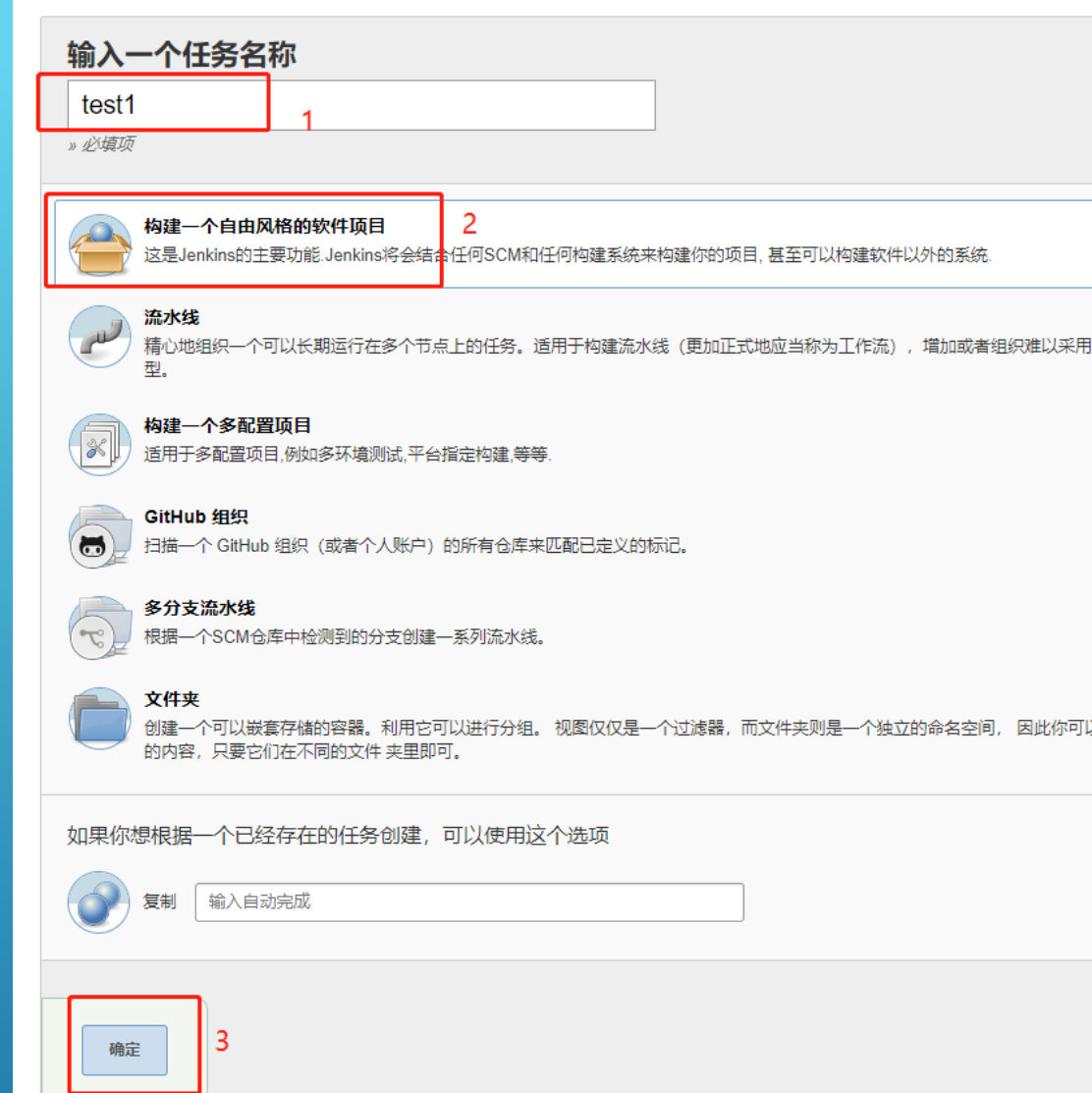
Docker 安装

保存 应用

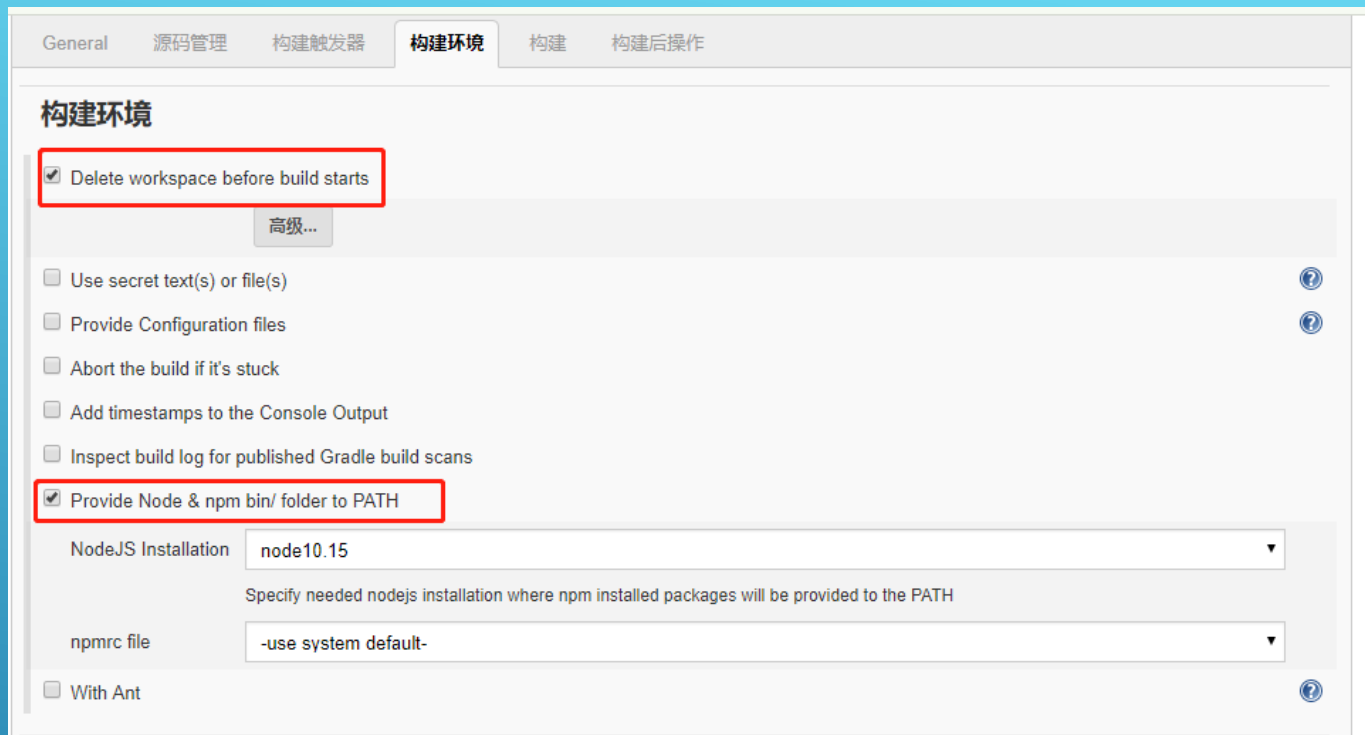
前端项目的CI配置与流程 (1)

► 首页->新建任务

前端项目的CI配置与流程 (2)

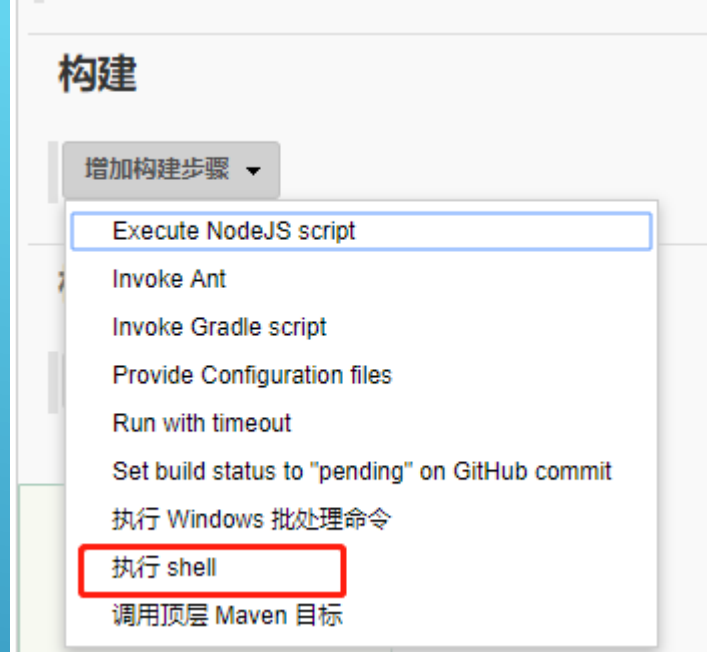


- ▶ 很关键的一步，任务的配置中，
- ▶ 一定要勾选Provide Node...
- ▶ 否则跑不起来



前端项目的CI配置与流程（3）

- ▶ 另一个关键配置，就是构建前的
- ▶ 命令执行，因为库上的源码一般
- ▶ 都是没有node_modules的，
- ▶ 所以至少要执行npm i 和 npm build
- ▶ 两个命令。
- ▶ 举个例子，如图



前端项目的CI配置与流程（4）

- ▶ 最后就是写上仓库地址与登录账户

源码管理

☐ 无
☐ Git
☒ Subversion 我用的SVN

Modules

Repository URL

Credentials 添加

Local module directory

Repository depth

Ignore externals ☒

Cancel process on externals fail ☒

Add module...

Additional Credentials

Check-out Strategy

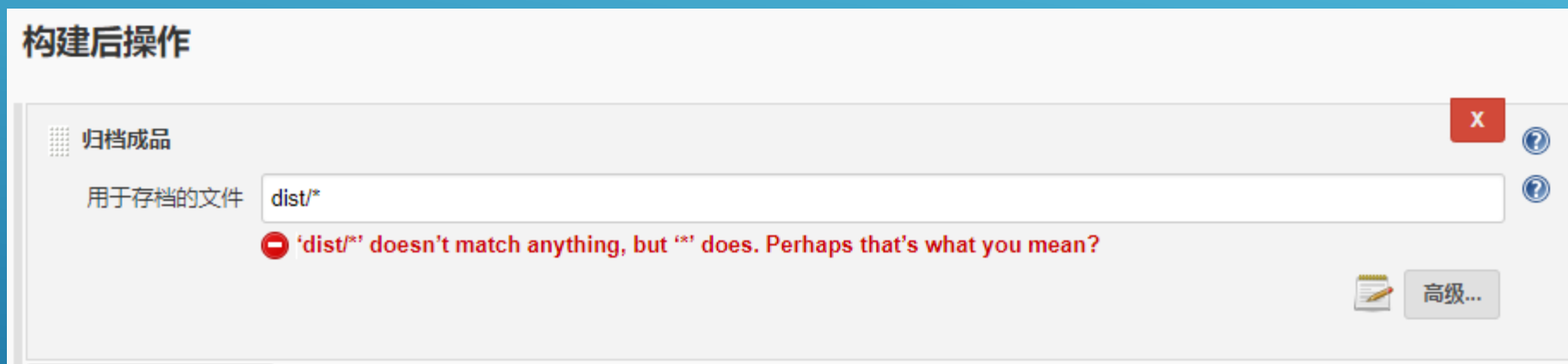
Delete everything first, then perform "svn checkout". While this takes time to execute, it ensures that the workspace is in the pristine state.

Quiet check-out ☒

源码库浏览器

前端项目的CI配置与流程 (5)

- ▶ 最后一步，设置打包后的代码的目录，方便jenkins给你压缩和下载



前端项目的CI配置与流程 (6)

开始构建一个项目：



构建过程输出log：



- 构建成功的结果，
- 可以打成zip包下载



前端项目的CI配置与流程（7）

- ▶ 1.开发者更新到库上后，实施人员或测试人员可以直接上平台自己去按需打包，不再依赖开发者协助打包；(持续、自动地构建/测试软件项目)
- ▶ 2.定时构建打包
- ▶ 3.库上代码如果有问题，也方便找到责任者和版本；
- ▶ 4.团队内管理轻松了很多，降低了很多重复劳动和沟通成本

JENKINS解决的问题

▶ Thanks

▶ Author: ShenJet

▶ <https://github.com/ShenJet>

END

Several white diagonal lines of varying lengths and thicknesses are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.