# AutoTest Studio入门系列10：Jenkins集成AutoTest Studio

随着敏捷开发的盛行，在很多的项目中都有着自己的CI环境，本节我们将介绍如何实现在Jenkins集成AutoTest Studio。在阅读本节内容前，假定你已经对Jenkins有一定的了解，并且本文的内容也仅仅是围绕Jenkins和AutoTest Studio集成，对一些基本操作不会做过多描述。

Jdk和Jenkins安装包：

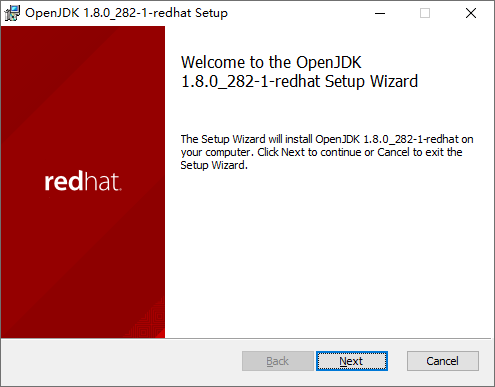
java-1.8.0-openjdk-1.8.0.282-1.b08.dev.redhat.windows.x86\_64.msi

Jenkins.war

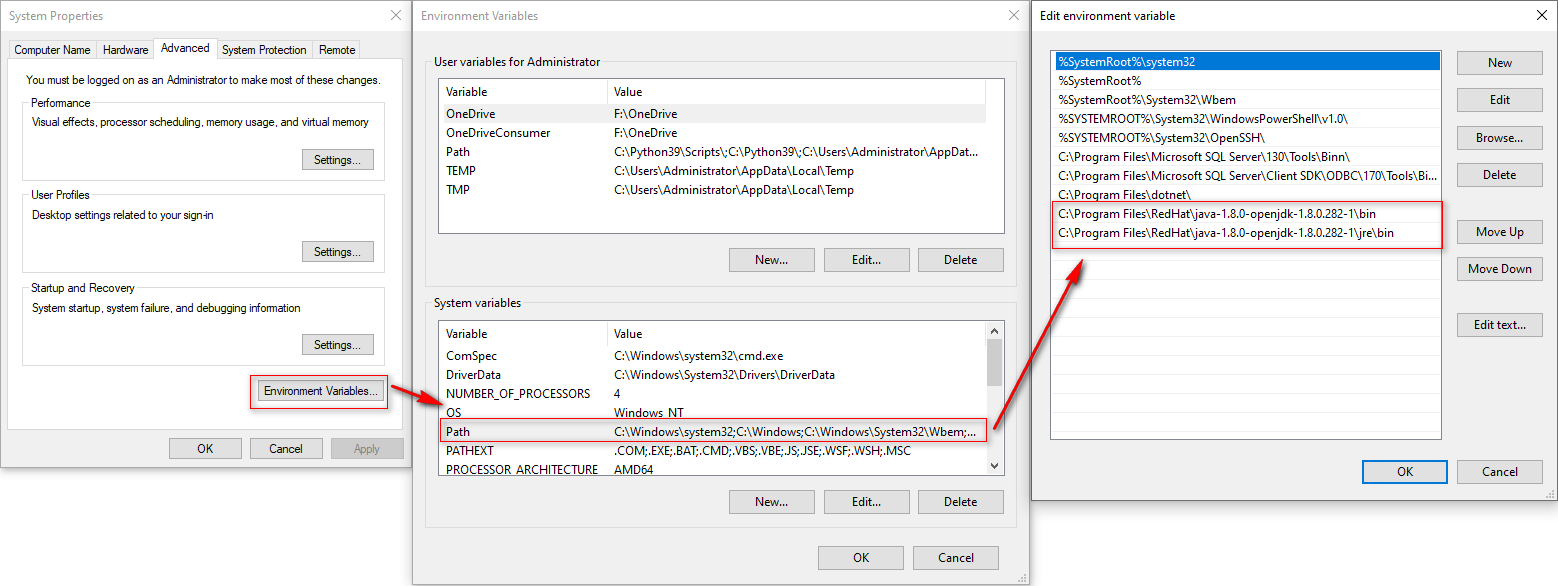
**注意**:jdk可以选用其它的jdk版本，但在本文的demo中Jenkins必须使用war包以命令行方式启动，因为AutoTest Studio是桌面应用程序，需要桌面环境中运行，如果Jenkins以服务形式安装，将无法正常运行AutoTest Studio。

**安装jdk**

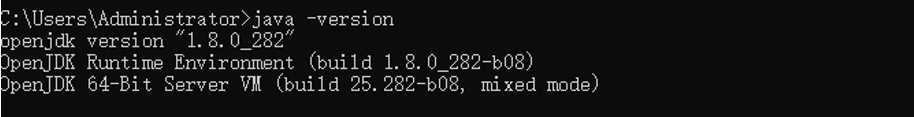
Jdk的安装非常简单，一直点击“Next”按钮即可。



安装完成后，检查系统环境变量，本文中使用jdk安装包会自动配置环境变量，如果其他jdk的版本没有配置环境变量，需要手工配置。

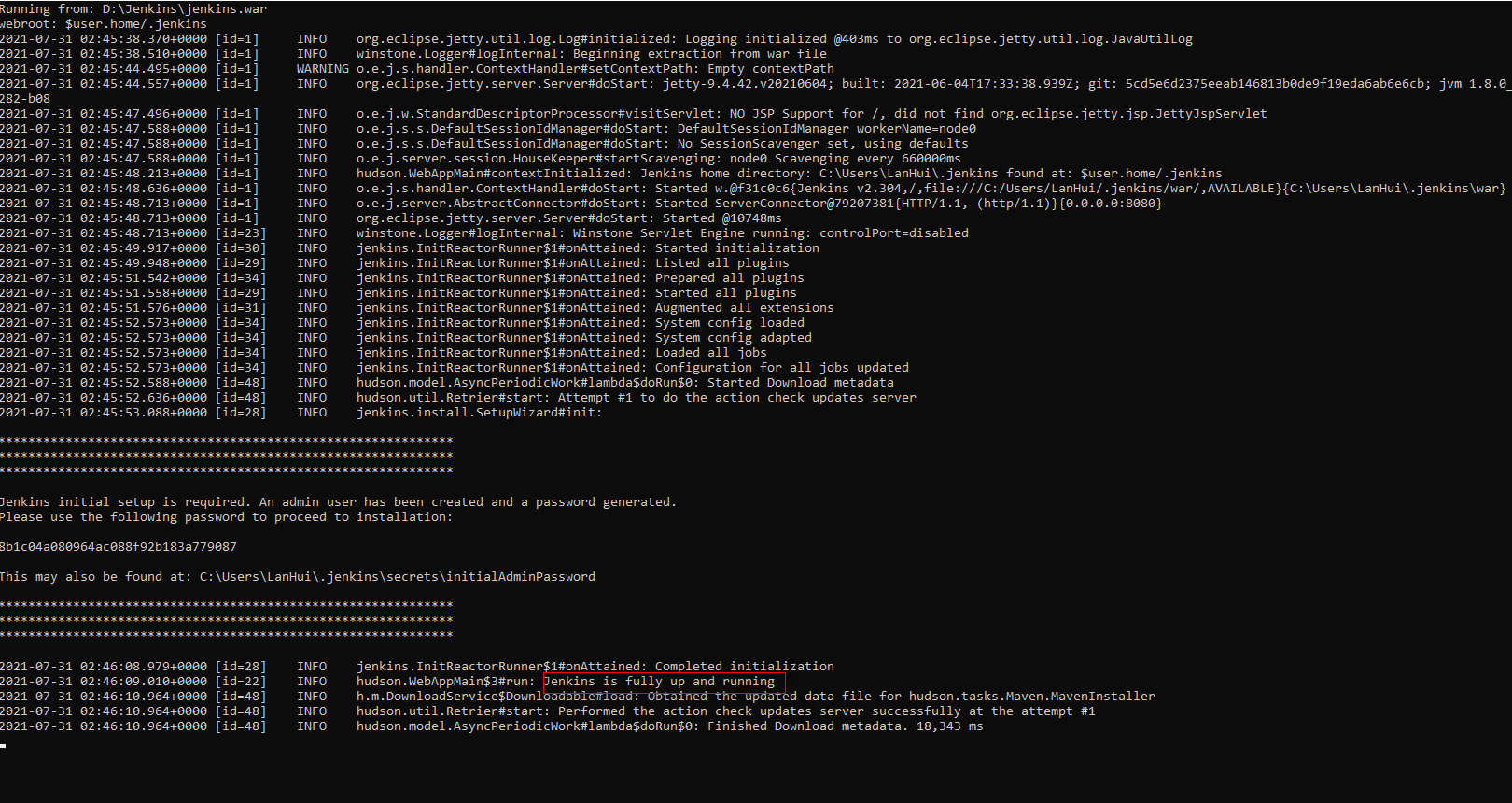


在cmd中运行“java -version”返回如下版本信息，说明jdk安装正常，并且已经配置好环境变量。

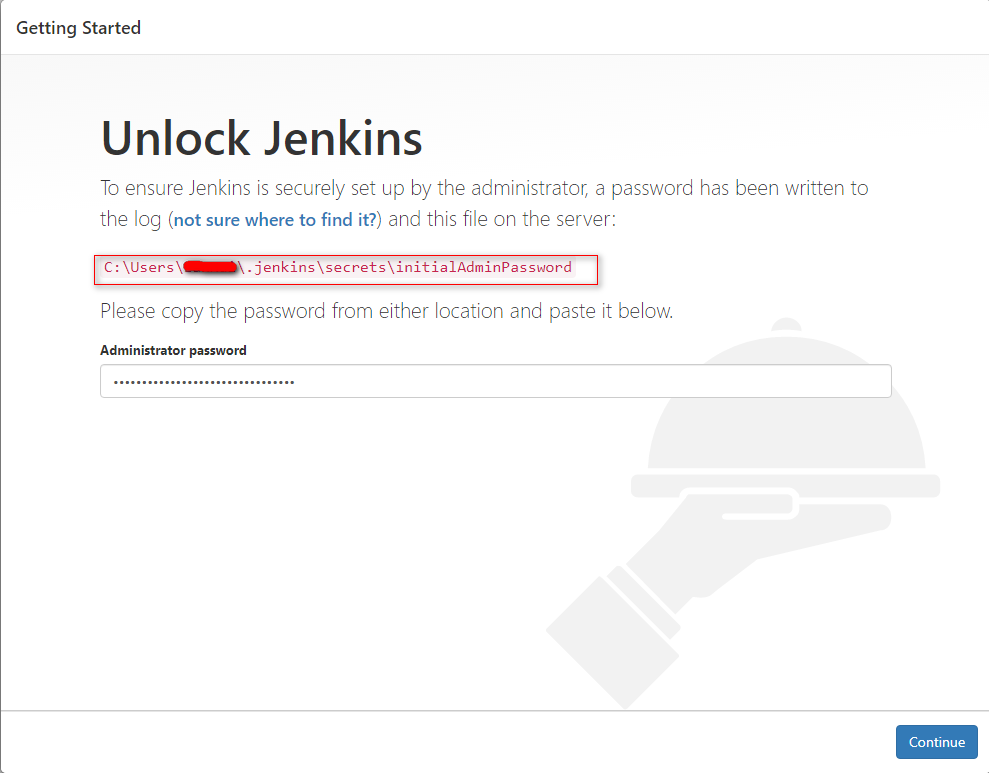


**启动Jenkins**

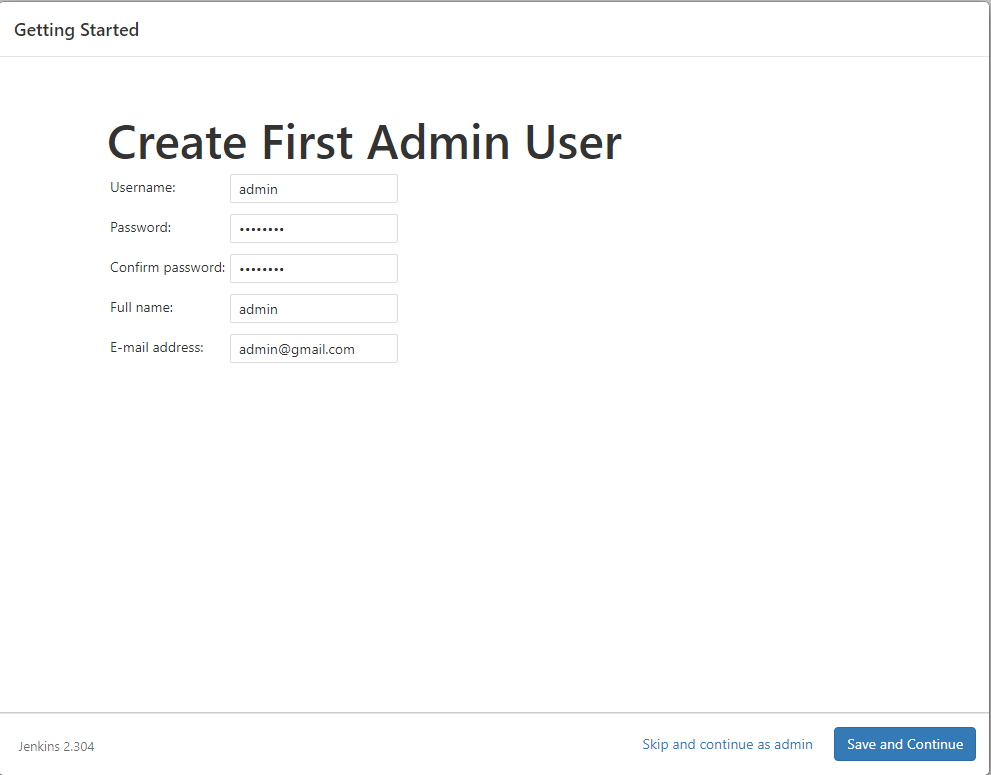
通过命令行方式启动Jenkins，启动命令“*java -jar "D:\Jenkins\Jenkins.war*"”，当启动信息中出现“Jenkins is fully up and running.”说明Jenkins已经正常启动。



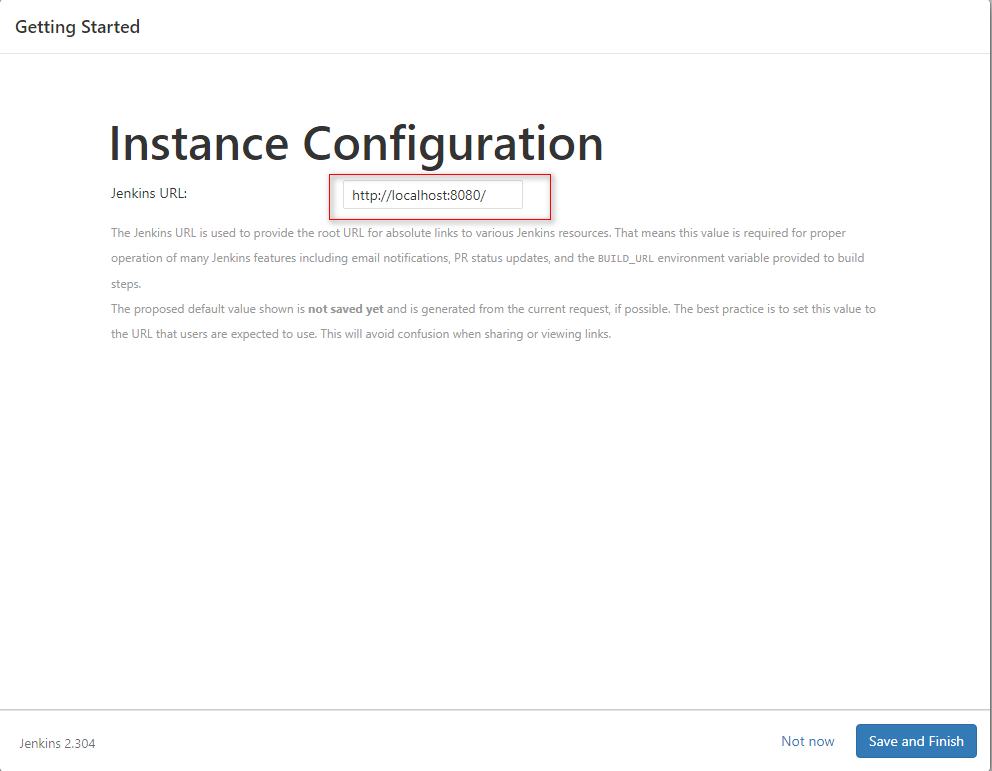
使用浏览器登陆“http://localhost:8080”。Administart password在提示的文件中，如下图所示。



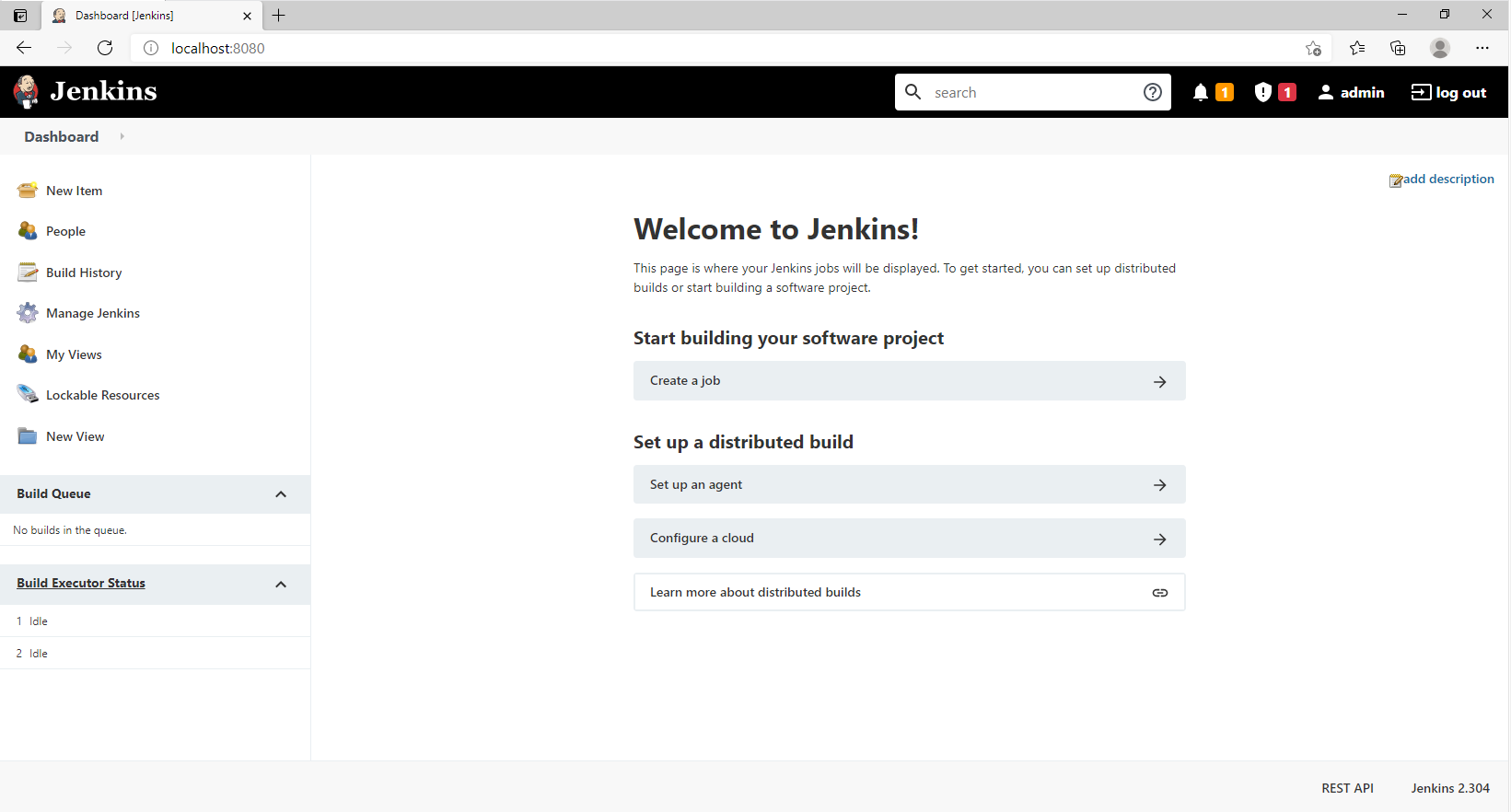
首次使用Jenkins时，需要安装默认的插件，按照提示安装即可。安装完成后，需要创建一个用户。



用户创建完成后，Jenkins会要配置登录ip地址和端口，这里我们使用默认的URL,即“http://localhost:8080/”；

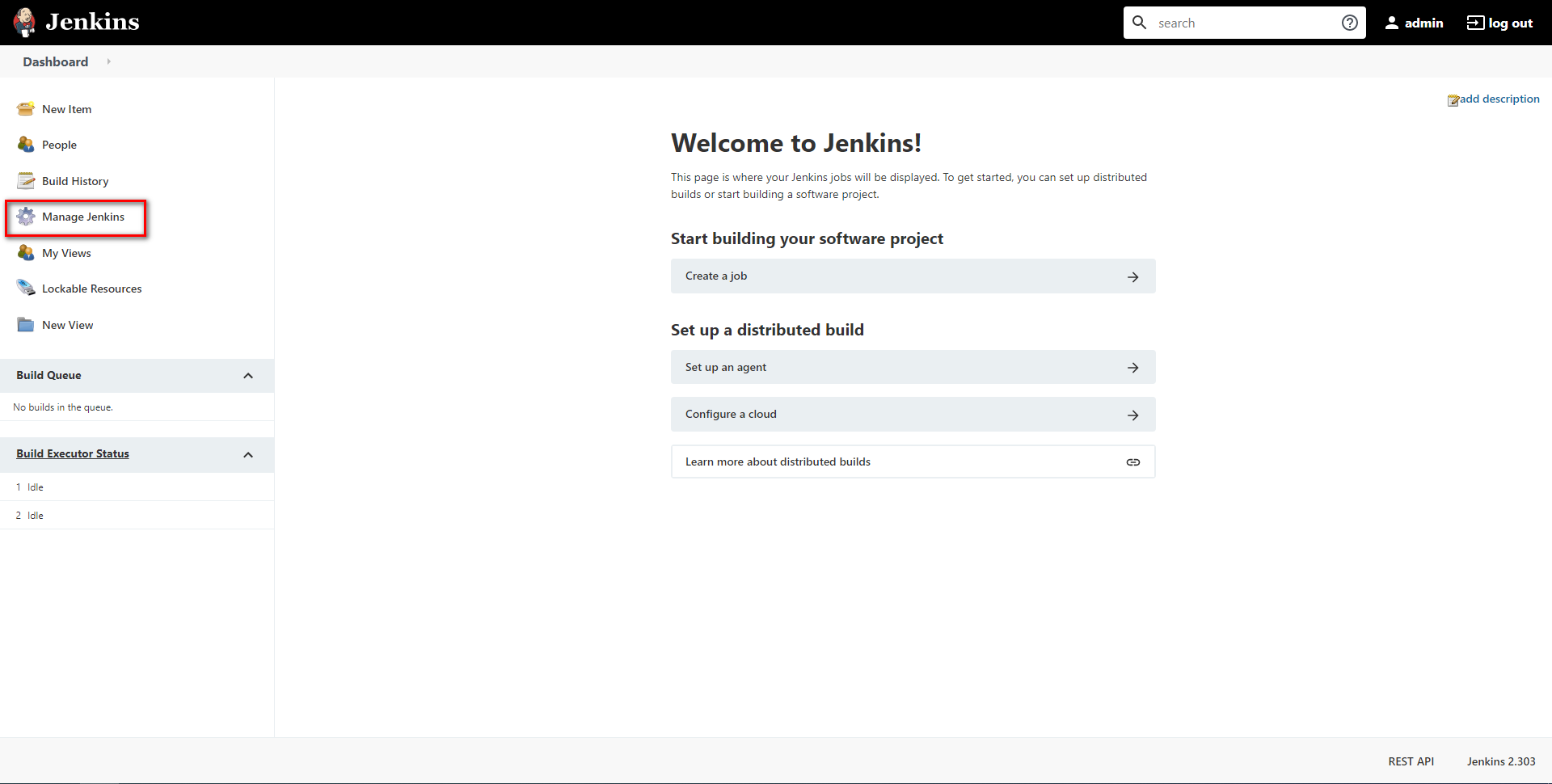


Jenkins基本配置完成以后，进入到Dashboard界面，如下图所示。

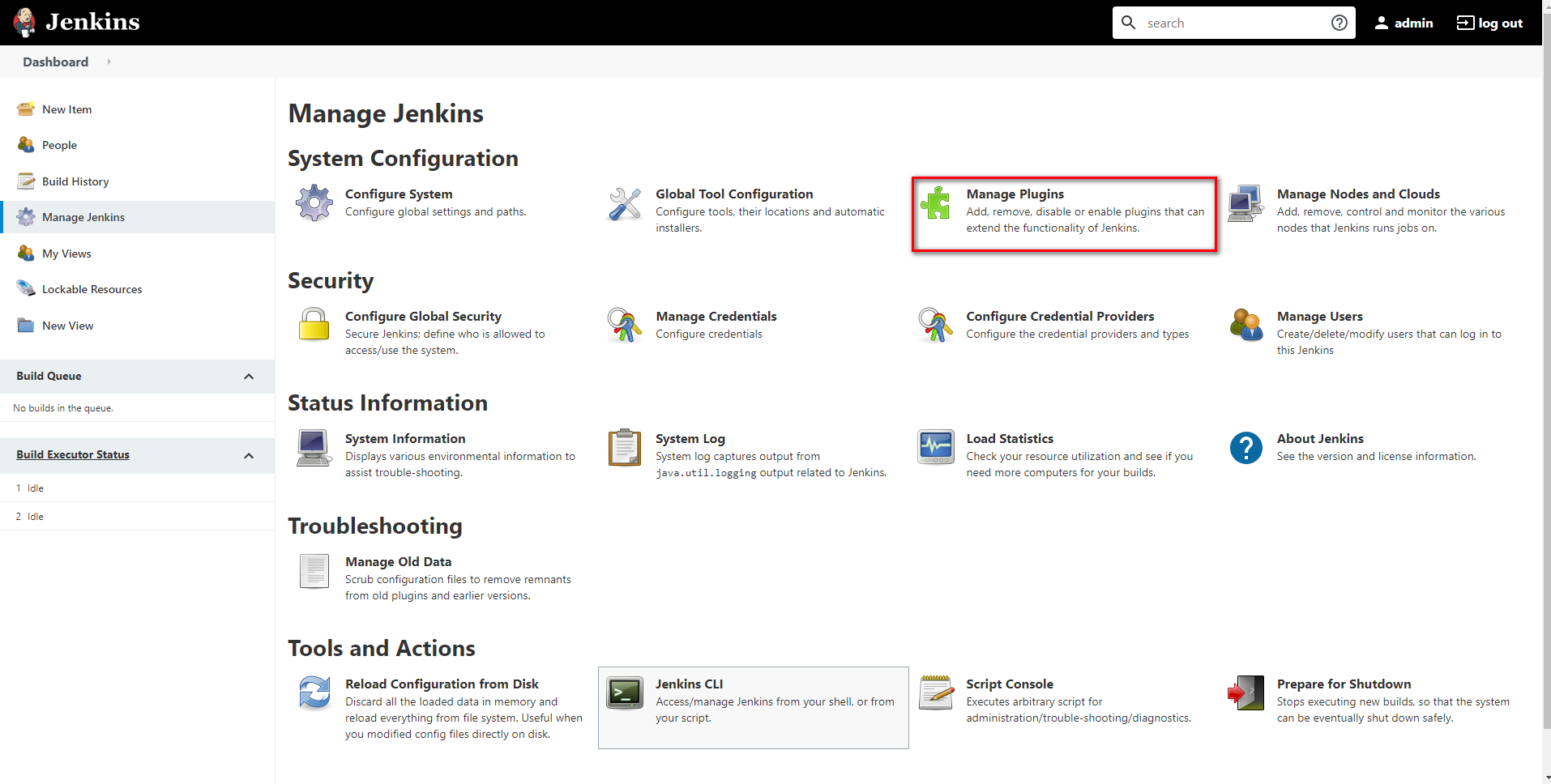


**安装HTML Publisher插件**

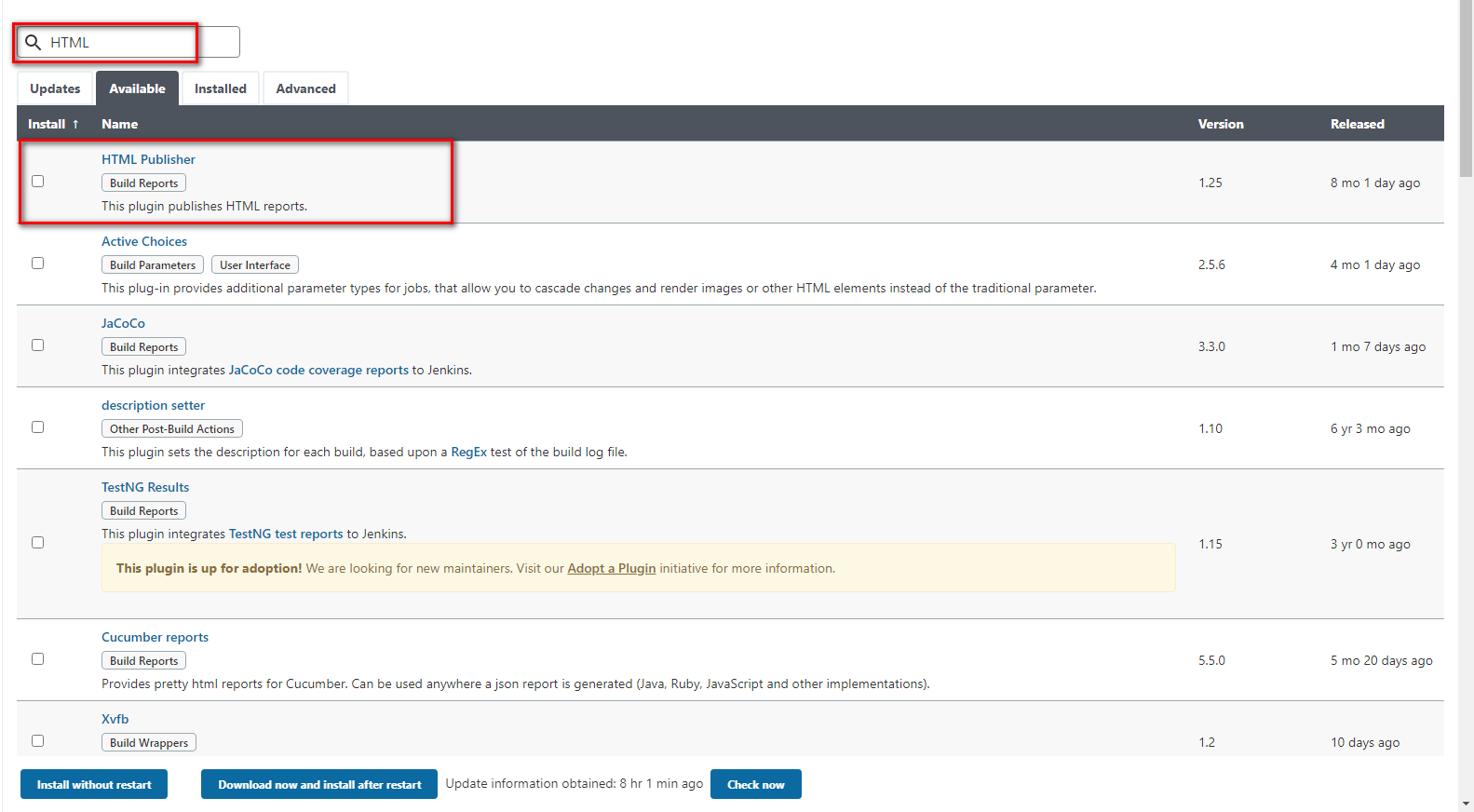
AutoTest Studio使用“HTML Publisher”插件在Jenkins中发布测试报告，因此我们需要安装HTML Publisher插件。在Jenkins的Dashboard界面中依次点击“Manage Jenkins”->“Manage Plugins”，进入插件管理模块。



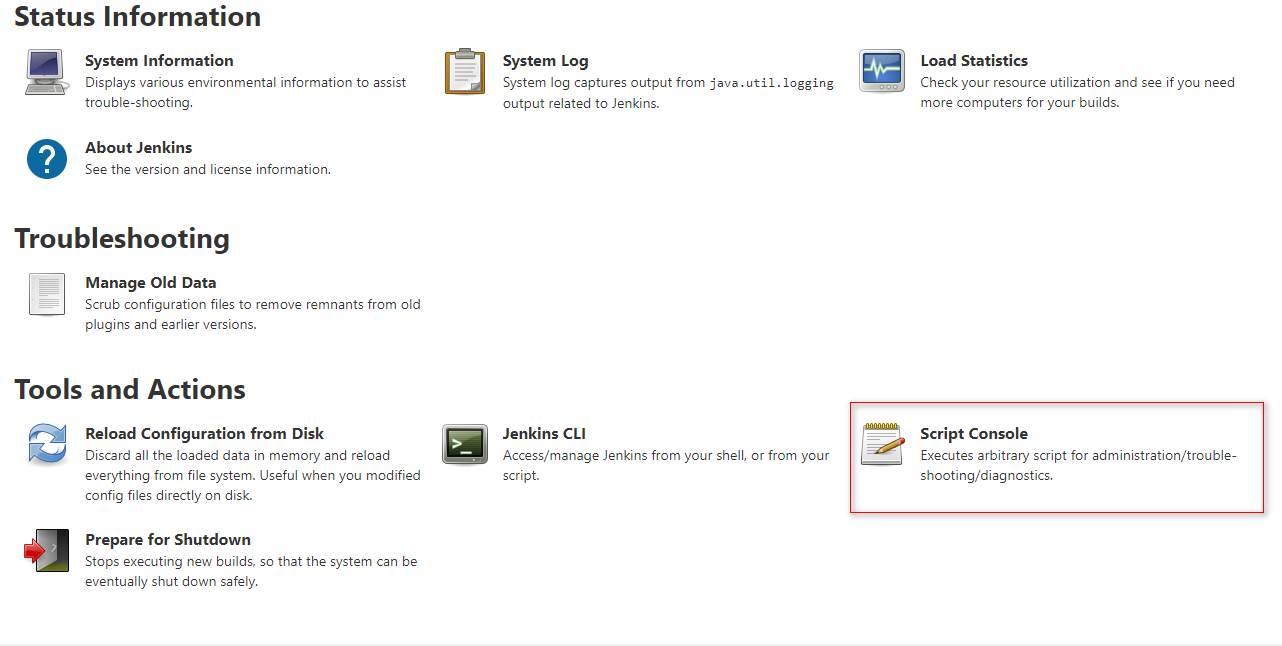
选择“Manage Plugins”



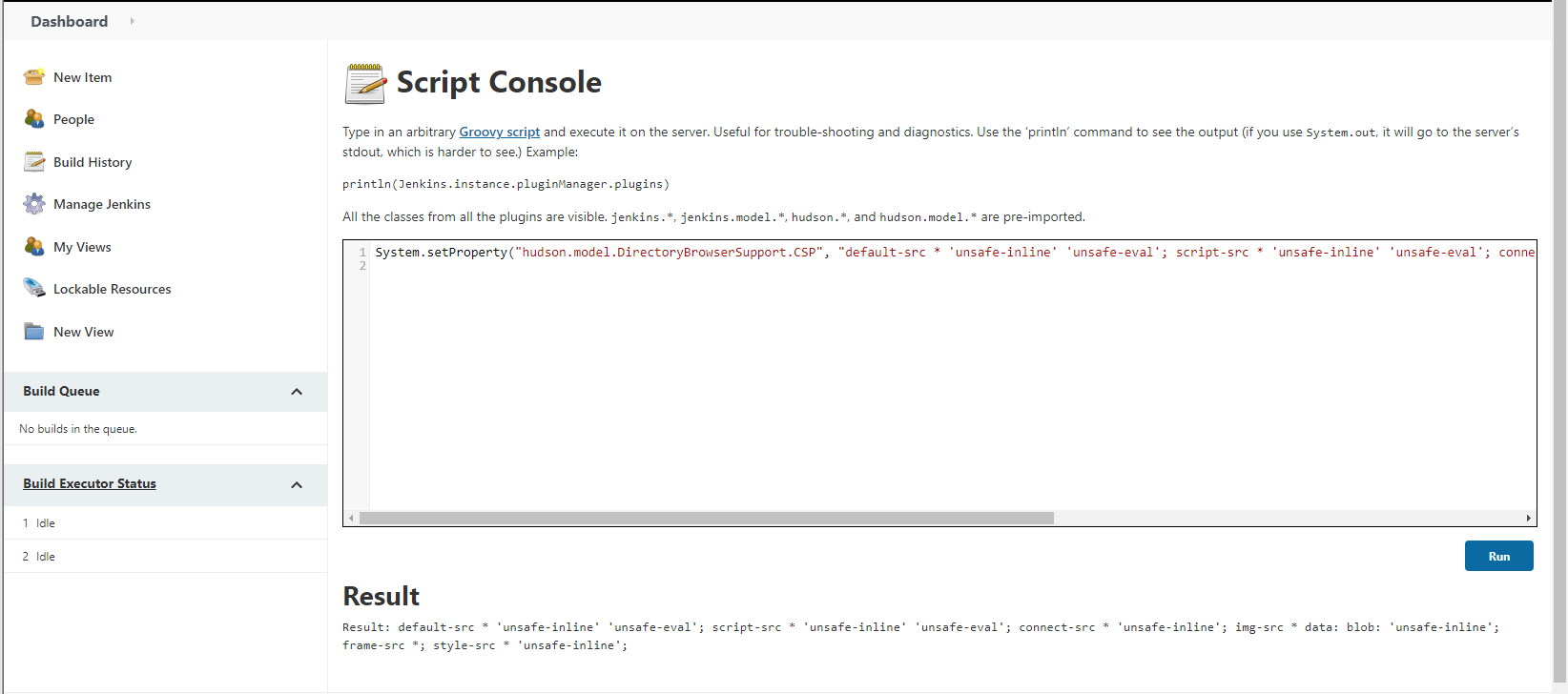
在搜索栏中输出“HTML”后，在搜索结果的列表中选中“HTML Publisher”插件，点击底部“Install without restart”按钮.



安装HTML Publisher插件完成以后，由于Jenkins出于安全考虑，禁止运行Html文件中js和css运行，因此我们还需要修改Jenkins中CSP安全机制，在Dashboard中依次点击“Manage Jenkins”->“Script console”，进入“Script console”。

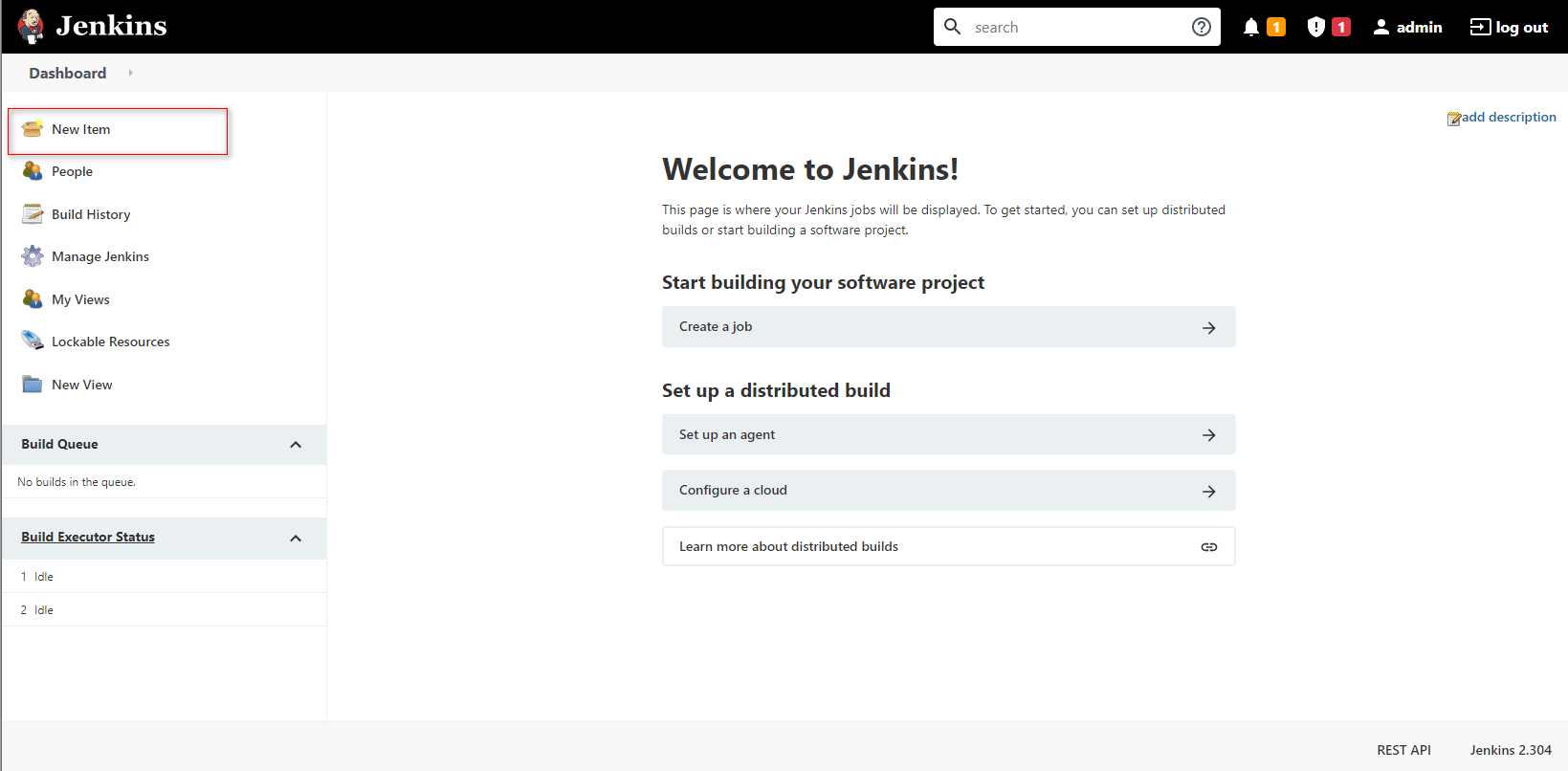


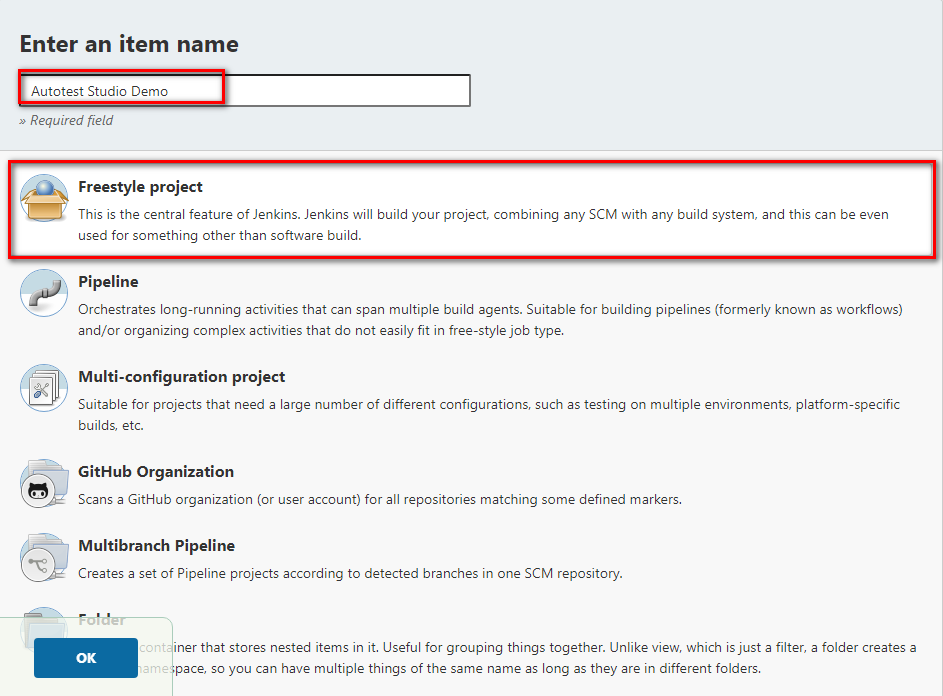
运行代码 *System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "default-src \* 'unsafe-inline' 'unsafe-eval'; script-src \* 'unsafe-inline' 'unsafe-eval'; connect-src \* 'unsafe-inline'; img-src \* data: blob: 'unsafe-inline'; frame-src \*; style-src \* 'unsafe-inline';")* ，运行结果如下图所示。

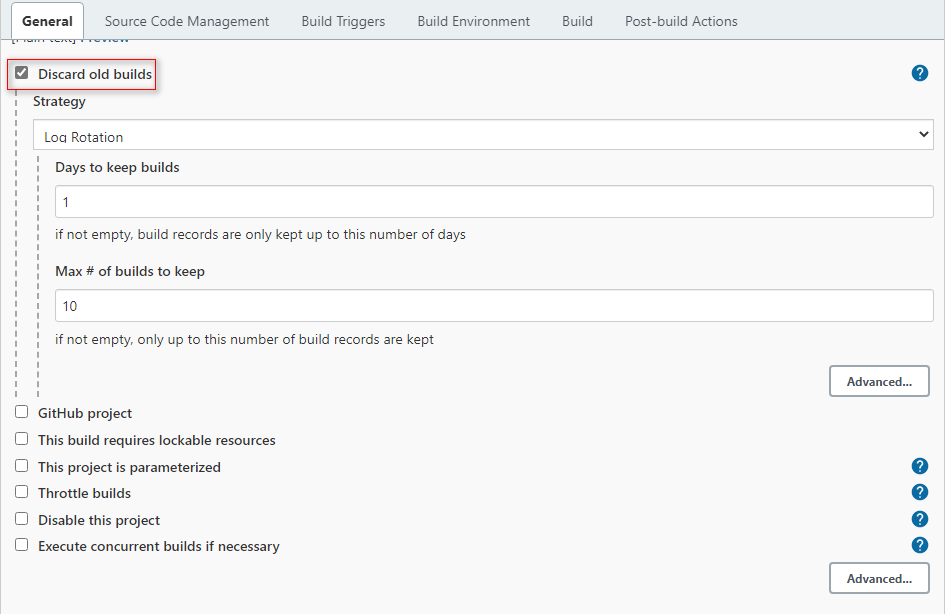


**新建一个Item**

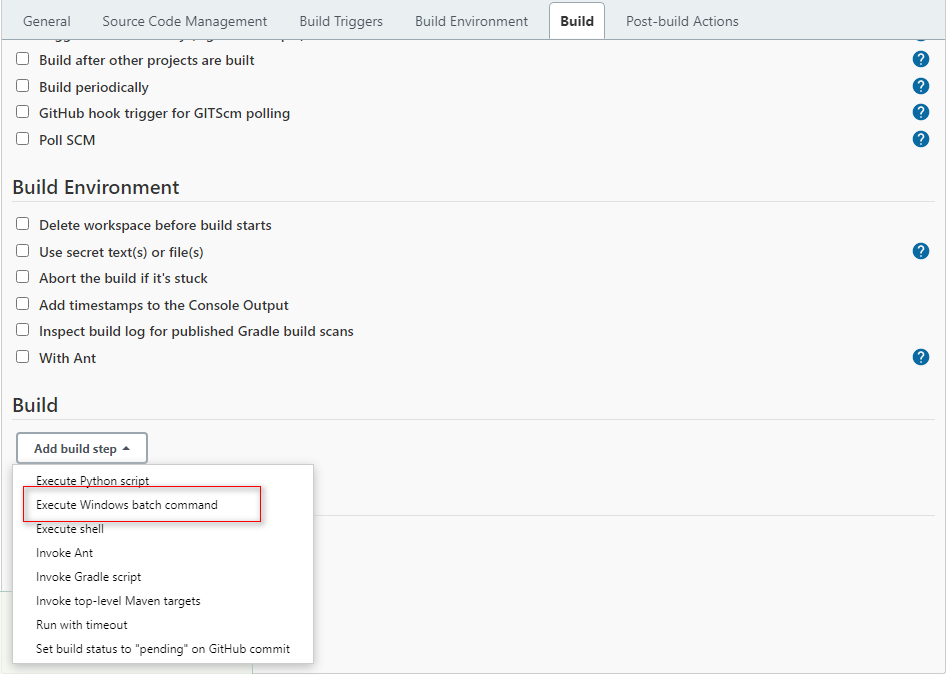
创建一个新的Item，名称为“AutoTest Stdio Demo”，风格为“Freestyle project”。







在Build中选择“Execute Windows batch command”添加AutoTest Studio调用命令。



调用命令*"C:\Program Files\AutoTest Studio\AutoTest Studio.exe" "C:\Users\Demouser\Documents\AutoTest Studio\Projects\ats\_example\ats\_example.tpro" -r -q* .

命令含义：

**"C:\Program Files\AutoTest Studio\AutoTest Studio.exe"**，AutoTest Studio.exe路径。

**"C:\Users\Demouser\Documents\AutoTest Studio\Projects\ats\_example\ats\_example.tpro"**，项目路径，在任务运行前需要在项目中选择要执行的脚本，AutoTest Studio会自动保存到项目文件中。

**-r**，启动后立即运行任务。必选参数。

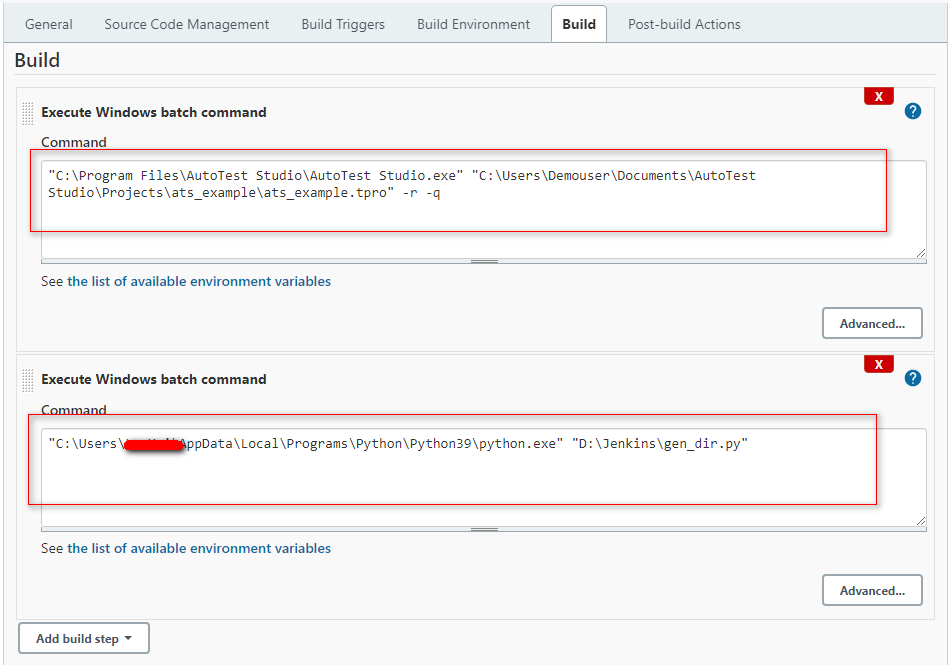
**-q**，任务运行完成后，退出AutoTest Studio程序。必选参数。

由于AutoTest Studio的报告和日志目录是动态产生的，因此我们需要通过一个脚本将任务输出文件“output.json”中日志目录解析出来，并映射到workspace中。

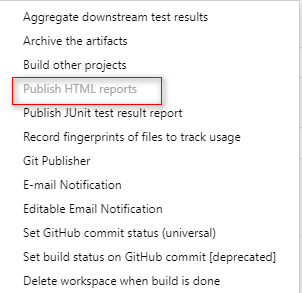
将以下python代码保存到一个文件中，如：“D:\Jenkins\jen\_dir.py”，但脚本位置没有限制。

|  |
| --- |
| import json import os   file=".\\output.json" target\_report\_dir="report\_logs" if not os.path.exists(file):  print("\"output.json\" does not exist.")  exit(-1)  fd=open(file,"r") log\_json=json.loads(fd.read()) fd.close() log\_dir=log\_json["LogDirectory"]  cmd='rmdir /s /q "{0}" & mklink /j "{0}" "{1}"'.format(target\_report\_dir,log\_dir) os.system(cmd) |

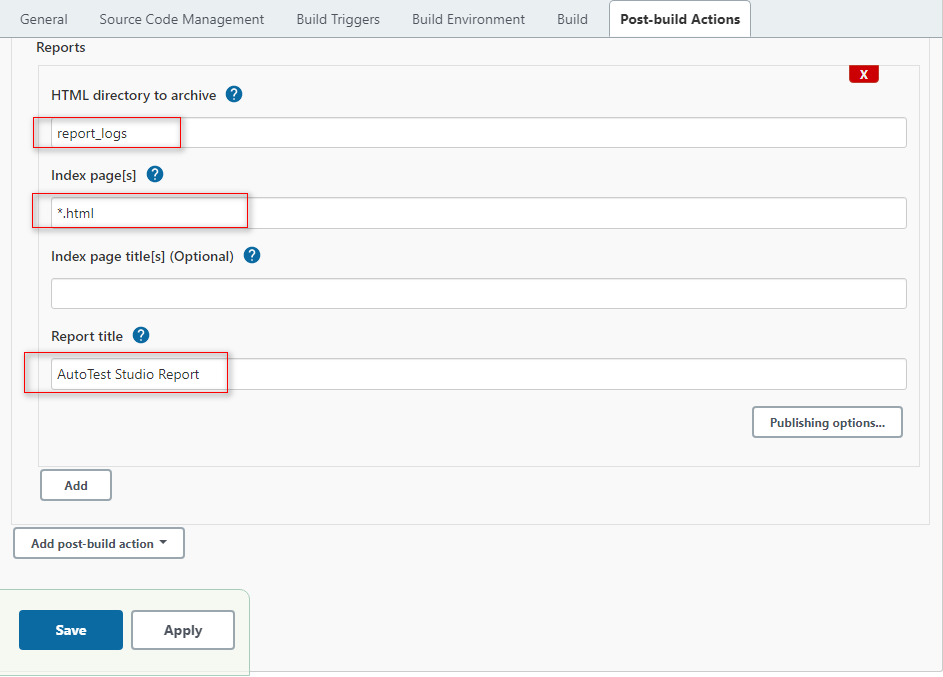
这两条构建命令如下图所示。



在“Post-build Actions”中通过“Publish HTML reposts”插件将AutoTest Studio的测试报告发到布到Jenkins中。



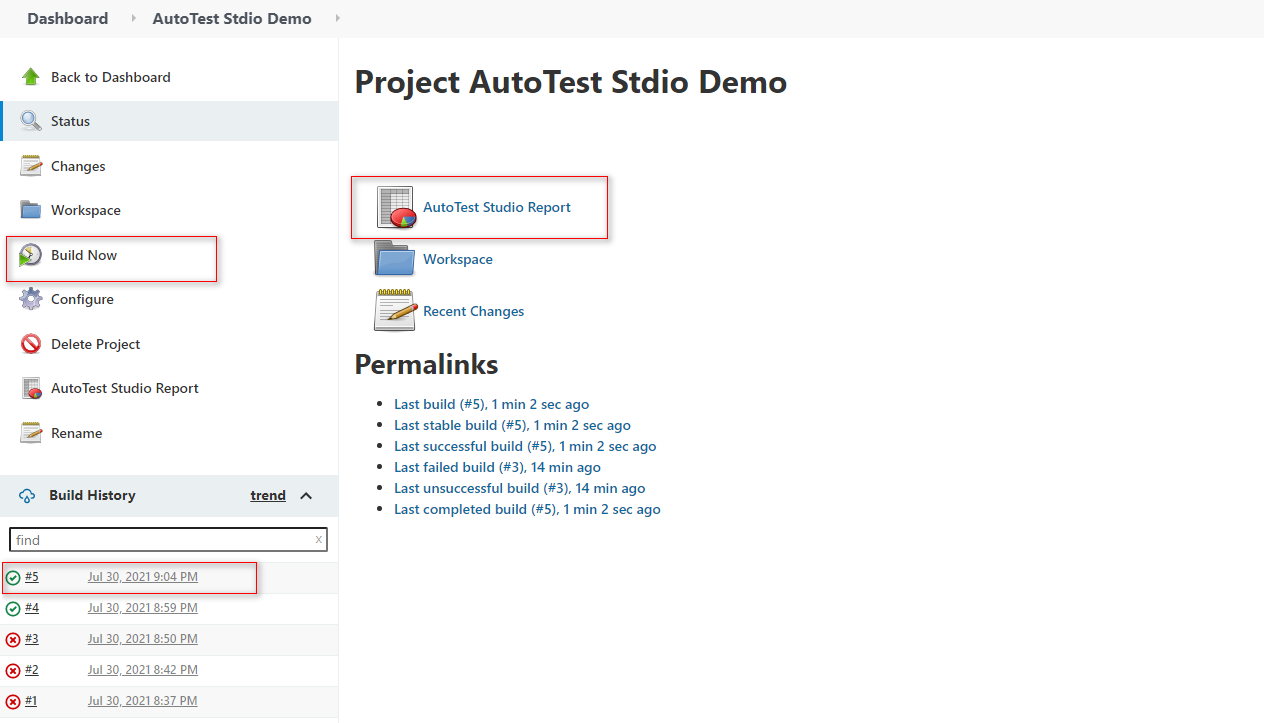
“Publish HTML reposts”插件配置参数如下，注意目录“report\_logs”必须上面脚本中命令保持一致。



配置完成，点击底部的“Save”按钮，保存配置。

**执行构建任务**

在Dashboard中点击“Bulid New”执行一个构建操作。



构建操作完成后，我们点击“AutoTest Studio Report”打开测试报告，可以看到如下图所示的内容，这个页面是由“Publish HTML report”插件抓取的，在页面的右上角点击“Zip”可以下载本次任务运行的测试报告和日志。

