# AutoTest Studio Getting started tutorials 10: Jenkins integrates AutoTest Studio

With the prevalence of agile development, many projects have their own CI environment. In this section, we will introduce how to integrate AutoTest Studio in Jenkins. Before reading this section, it is assumed that you already have a certain understanding of Jenkins, and the content of this article is only around the integration of Jenkins and AutoTest Studio, and will not describe some basic operations too much.

Jdk and Jenkins installation package:

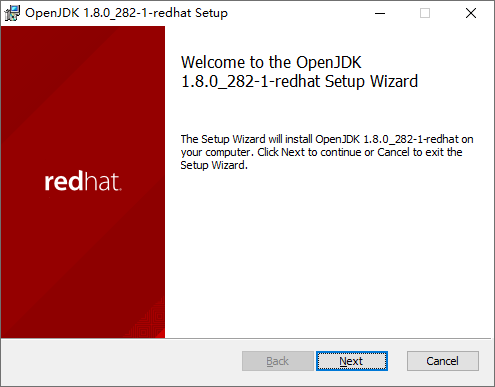
java-1.8.0-openjdk-1.8.0.282-1.b08.dev.redhat.windows.x86\_64.msi

Jenkins.war

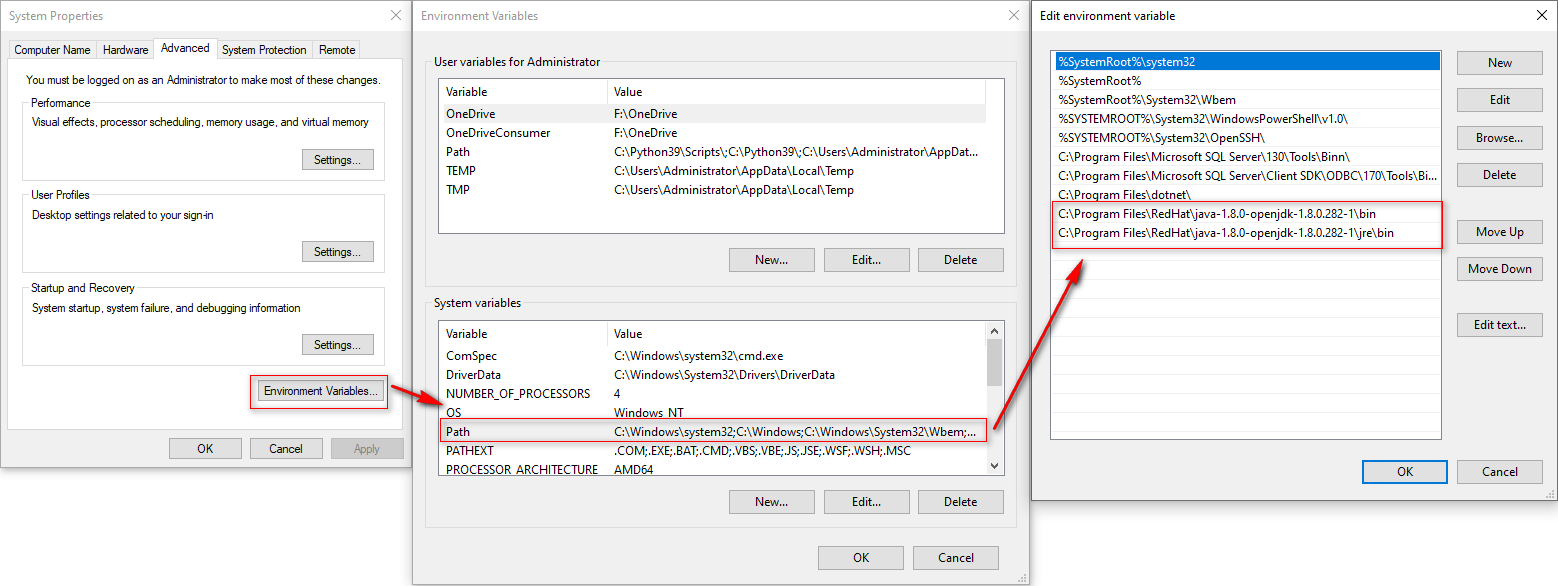
Note: jdk can use other jdk versions, but in the demo of this article, Jenkins must be started by the command line using the war package, because AutoTest Studio is a desktop application and needs to be run in a desktop environment. If Jenkins is installed as a service, it will not be able to Run AutoTest Studio normally.

**Install jdk**

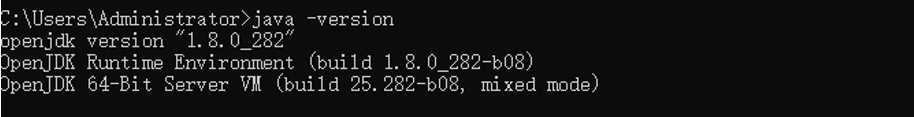
The installation of Jdk is very simple, just click the "Next" button all the time.



After the installation is complete, check the system environment variables. The jdk installation package used in this article will automatically configure the environment variables. If the environment variables are not configured for other jdk versions, you need to configure them manually.

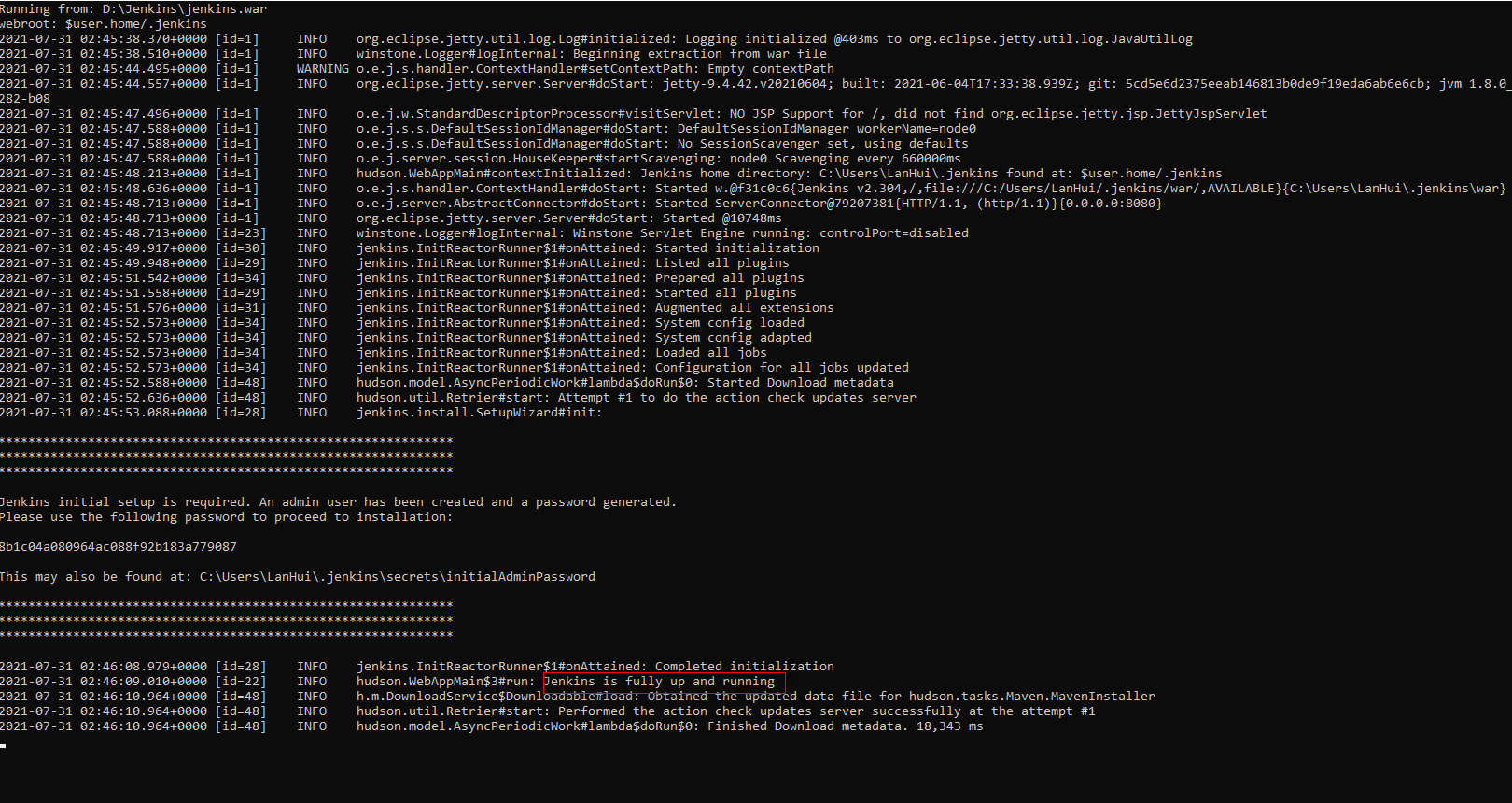


Running "**java -version**" in cmd returns the following version information, indicating that the jdk is installed normally and the environment variables have been configured.

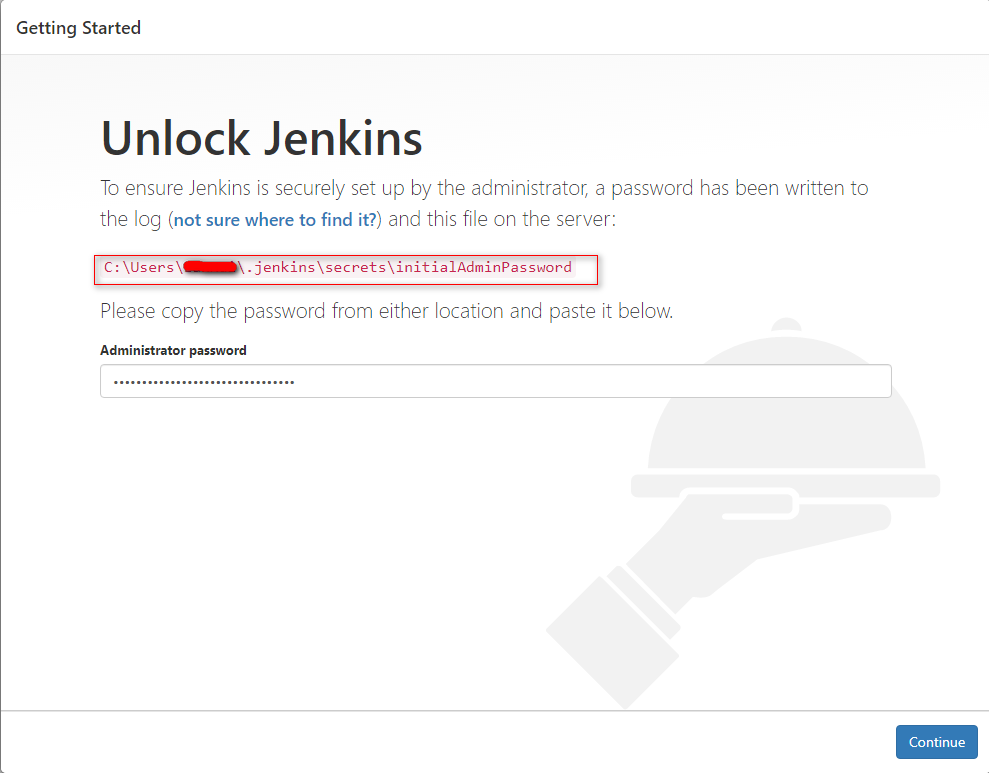


**Start Jenkins**

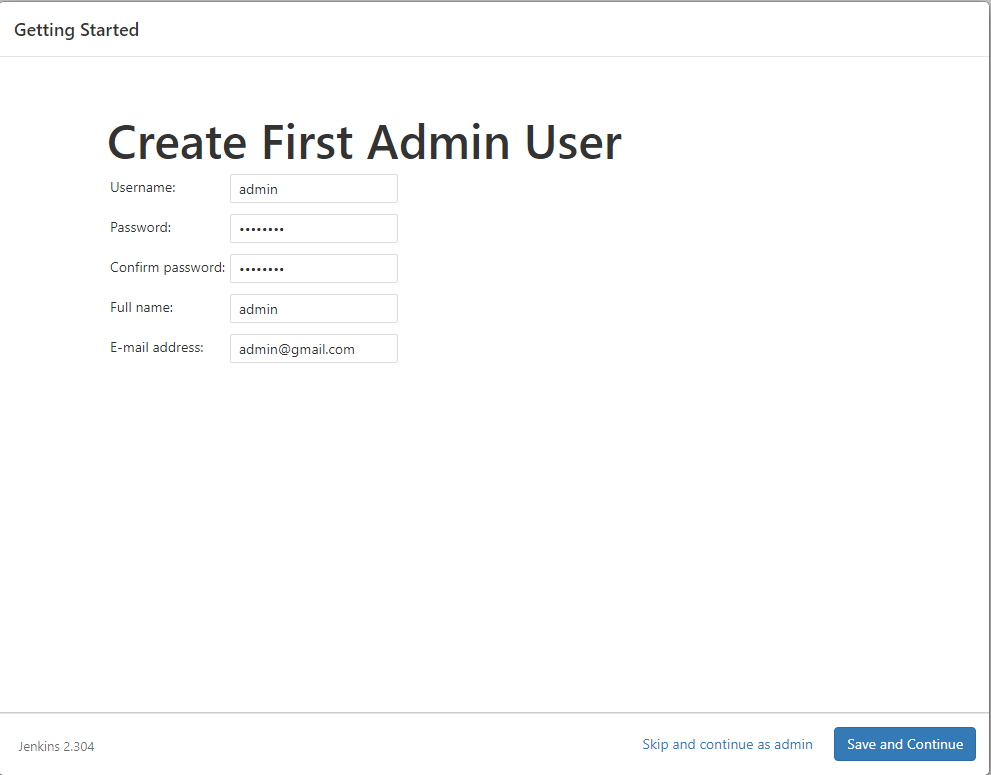
Start Jenkins through the command line, start the command "*java -jar "D:\Jenkins\Jenkins.war*"", when "**Jenkins is fully up and running**." appears in the startup message, it means that Jenkins has been started normally.



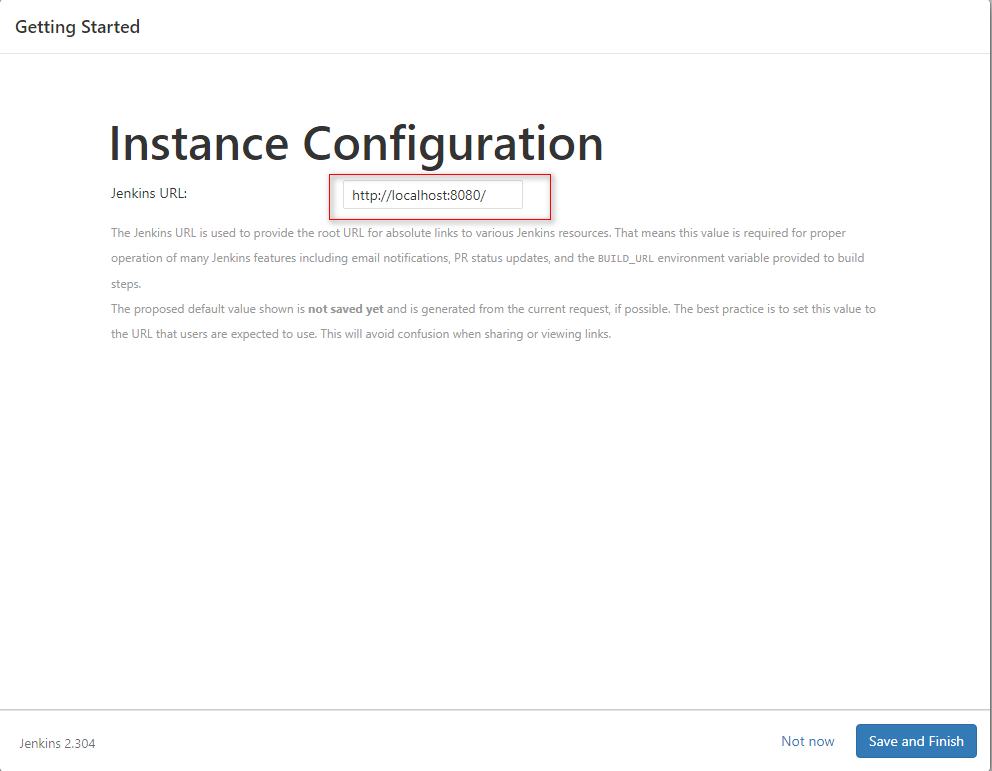
Use a browser to log in to "http://localhost:8080". Administart password is in the prompt file, as shown in the figure below.



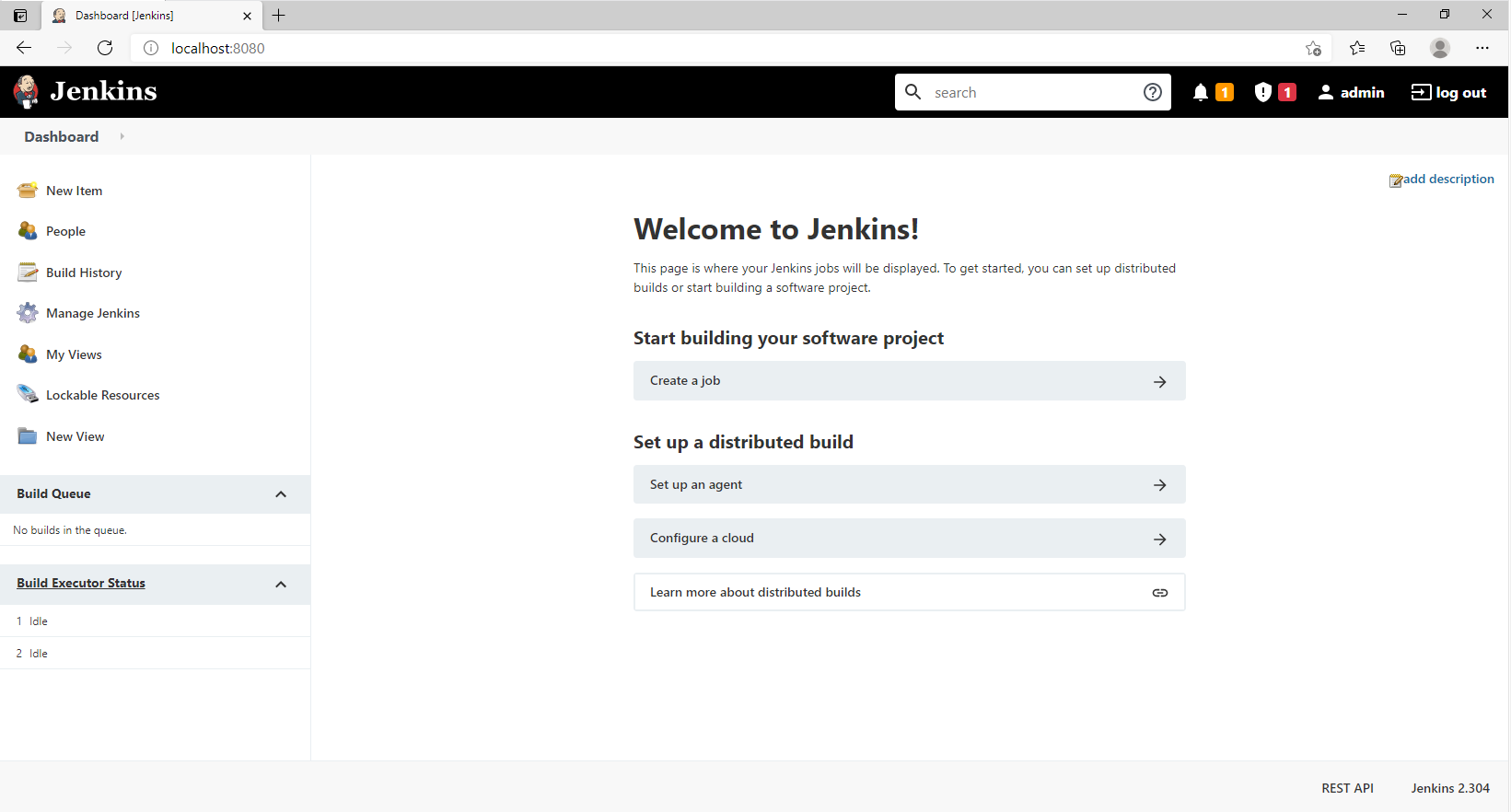
When using Jenkins for the first time, you need to install the default plug-in, just follow the prompts to install it. After the installation is complete, you need to create a user.



After the user is created, Jenkins will configure the login ip address and port. Here we use the default URL, which is "http://localhost:8080/";

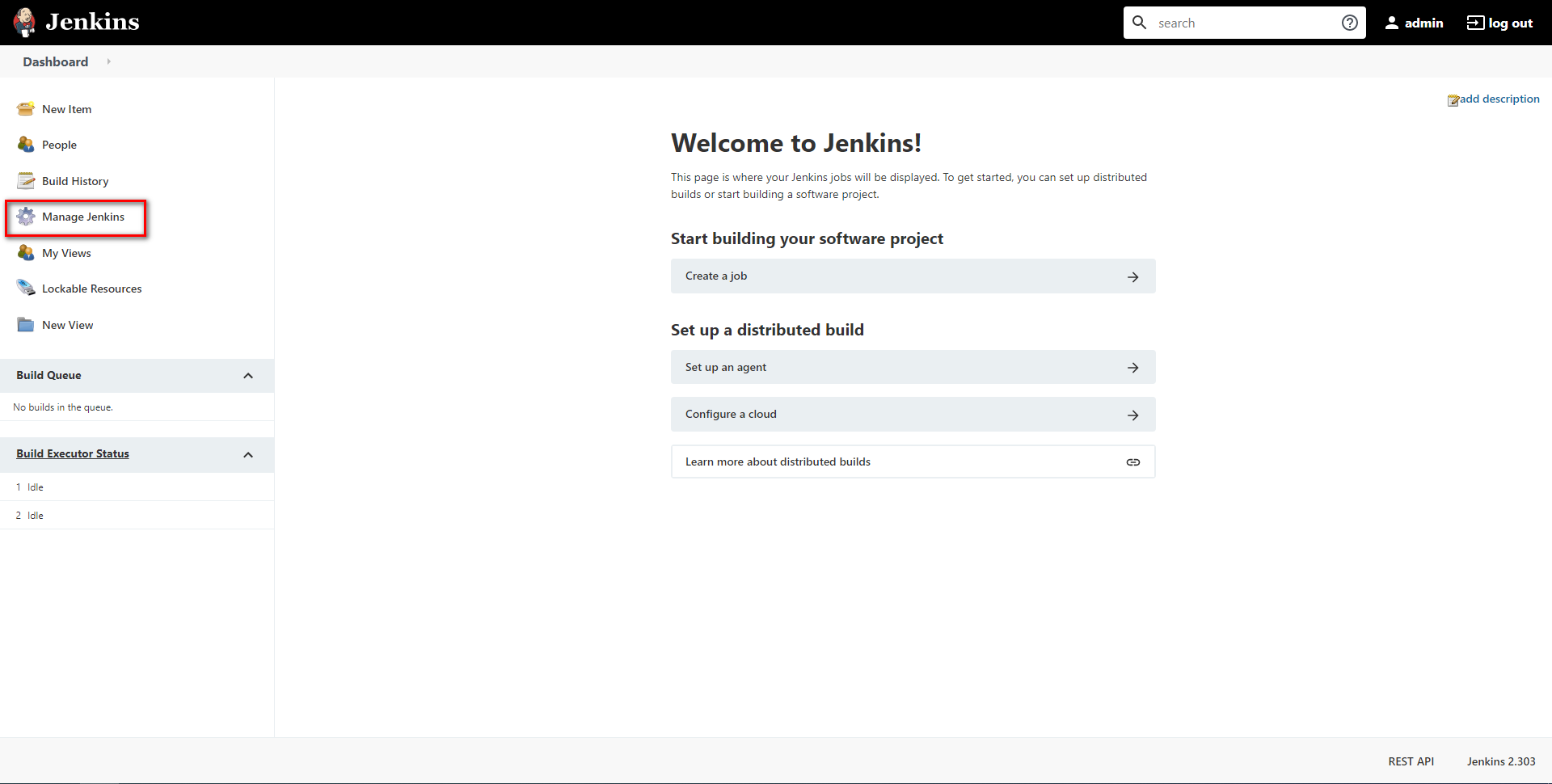


After the basic configuration of Jenkins is completed, enter the Dashboard interface, as shown in the following figure.

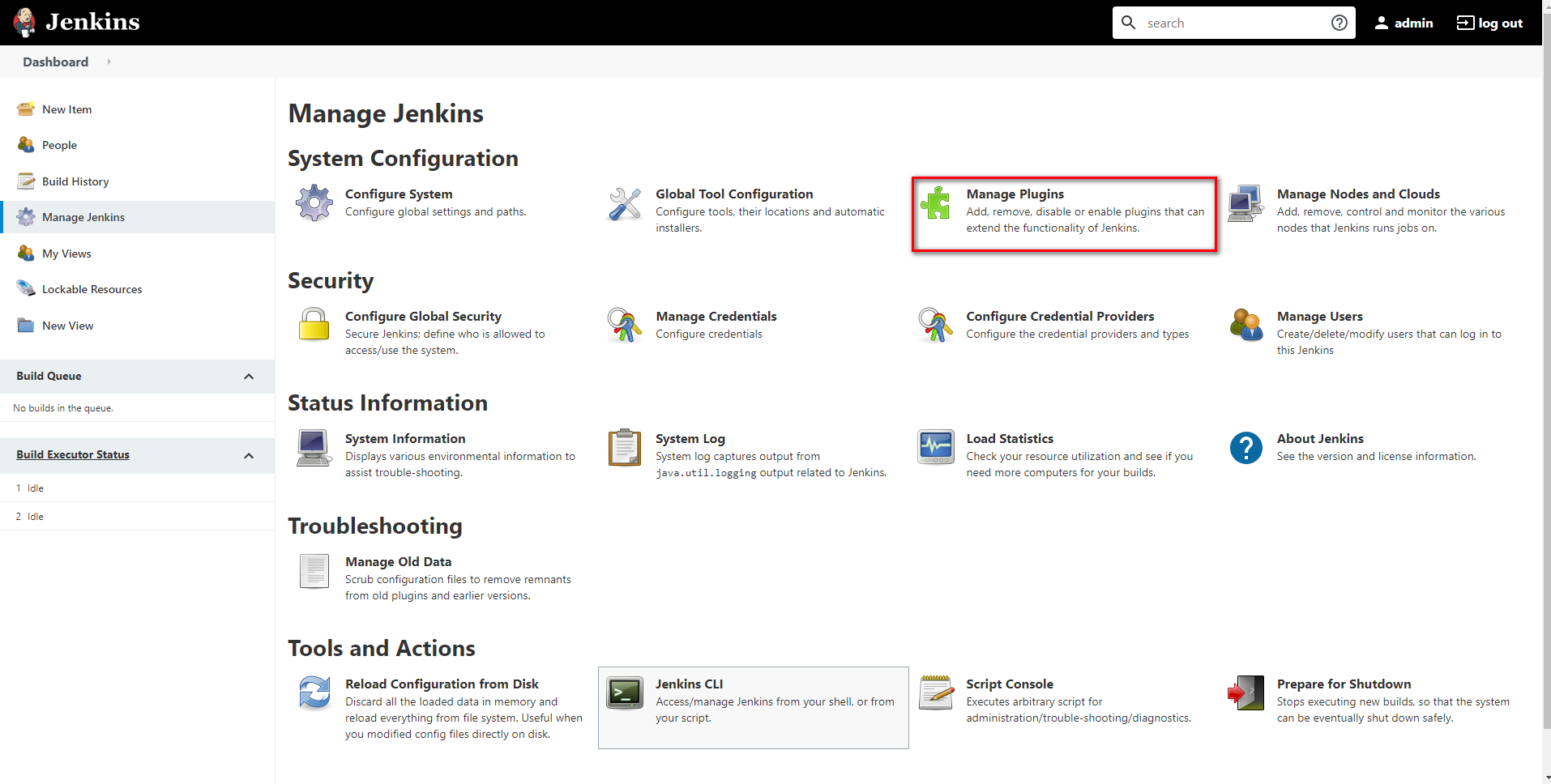


**Install HTML Publisher plug-in**

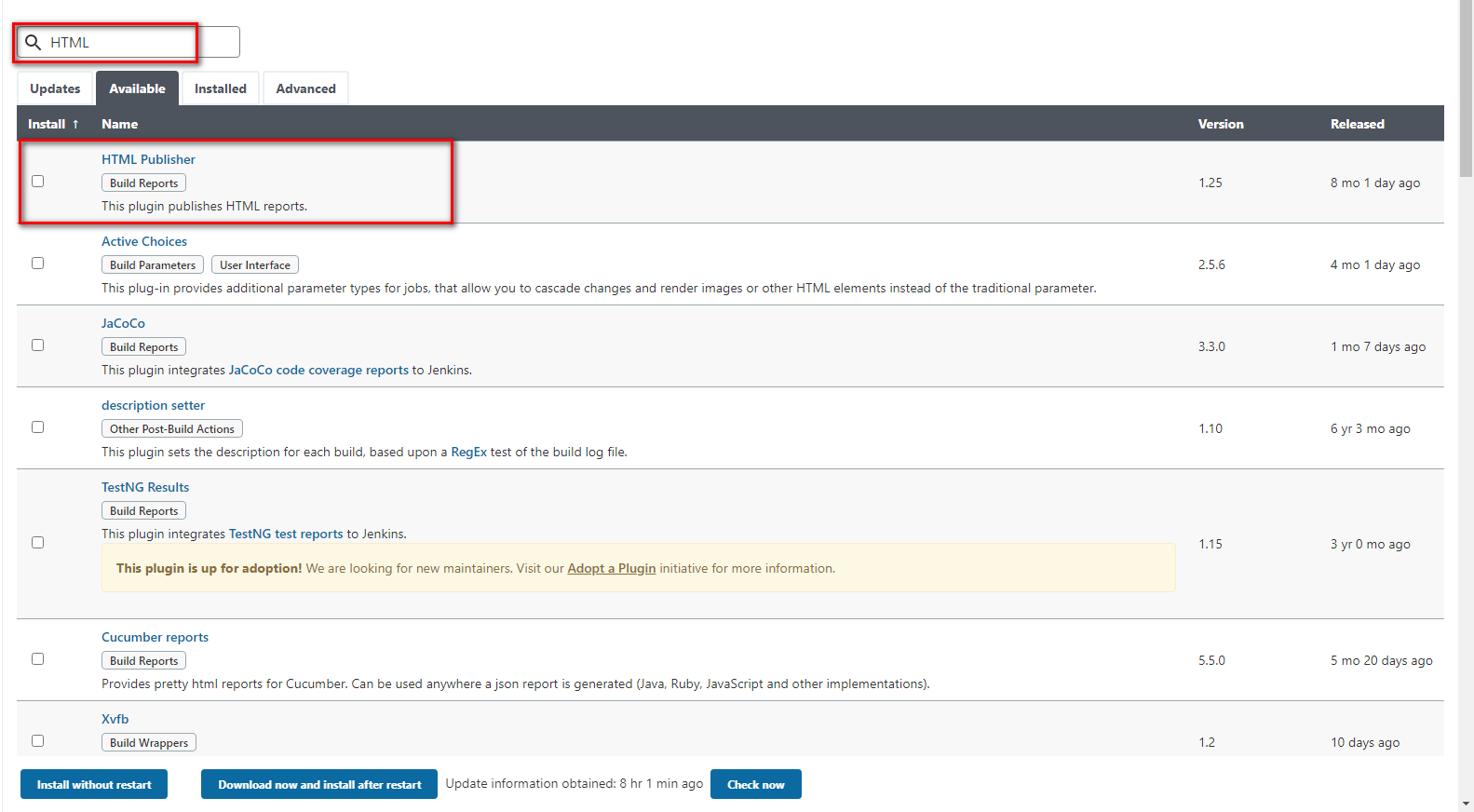
AutoTest Studio uses the "**HTML Publisher**" plug-in to publish test reports in Jenkins, so we need to install the HTML Publisher plug-in. In the Dashboard interface of Jenkins, click "**Manage Jenkins**" -> "**Manage Plugins**" to enter the plug-in management module.



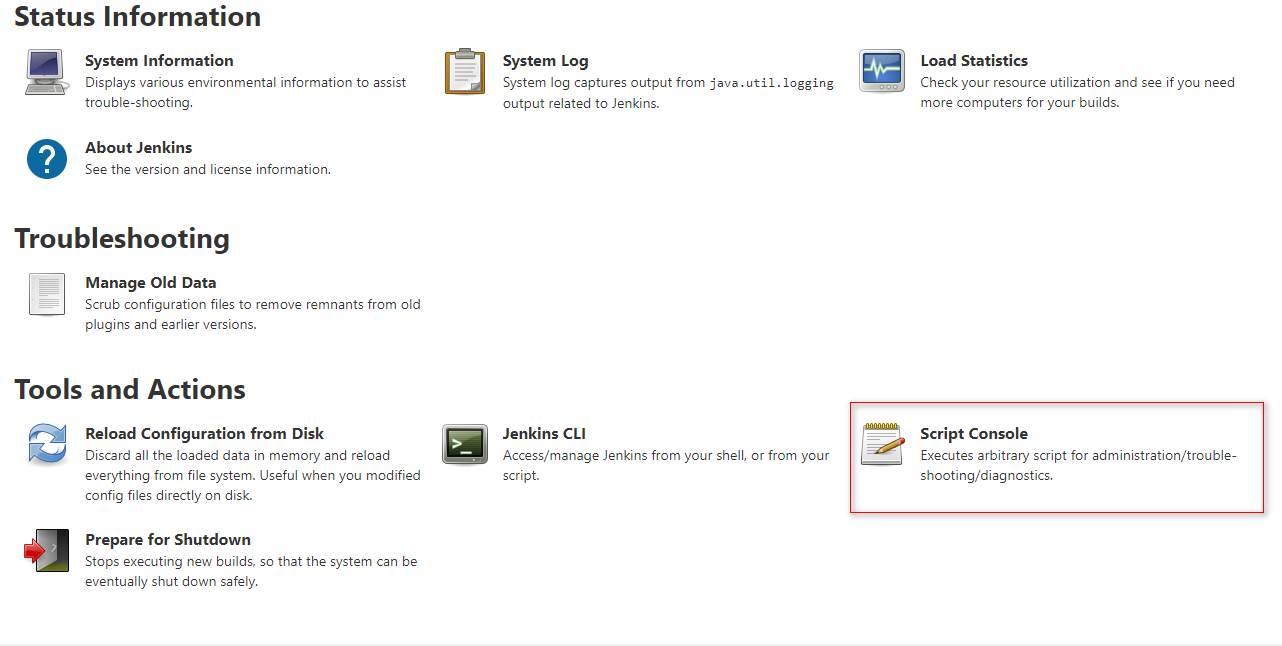
Select "**Manage Plugins**"



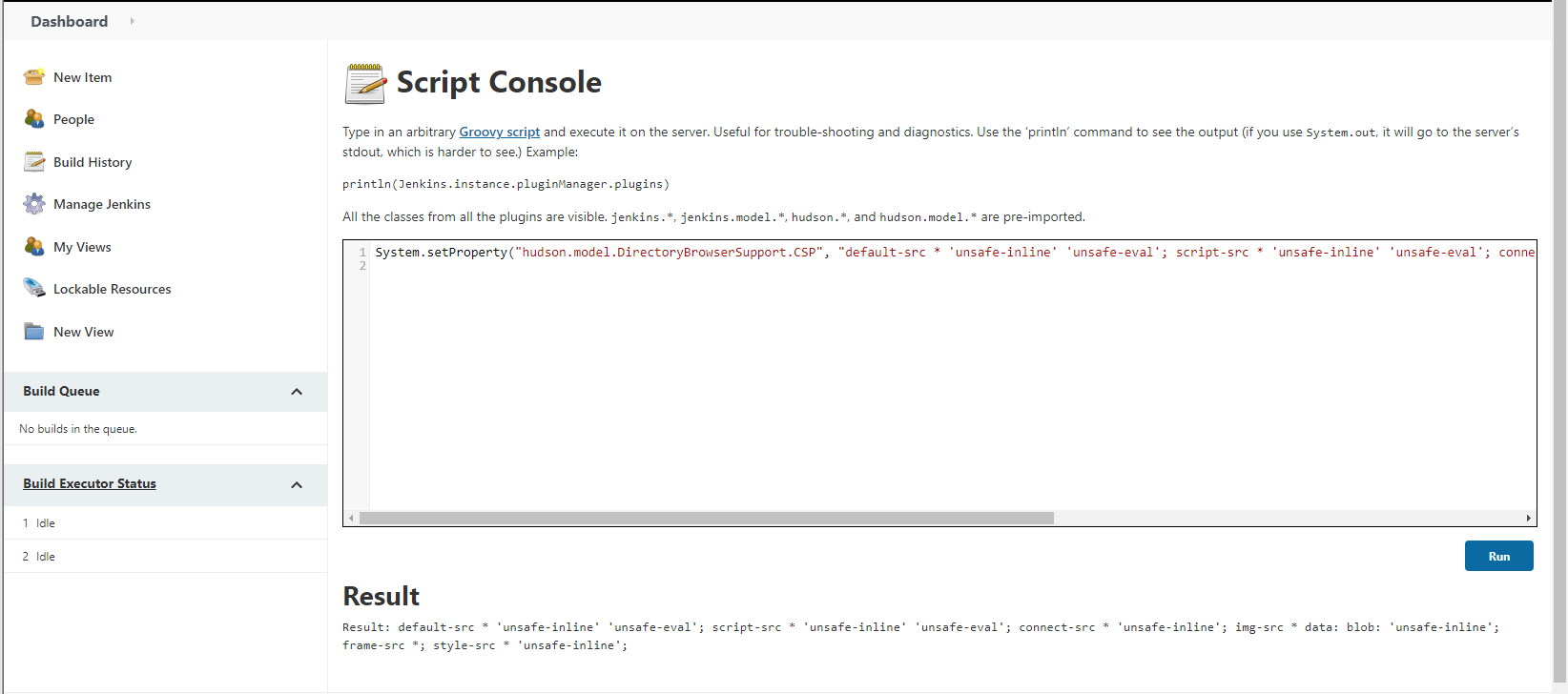
After outputting "**HTML**" in the search bar, select the "**HTML Publisher**" plugin in the list of search results, and click the "**Install without restart**" button at the bottom.



After installing the HTML Publisher plug-in, because Jenkins prohibits running js and css in Html files for security reasons, we also need to modify the CSP security mechanism in Jenkins, and click "**Manage Jenkins**" -> "**Script console**" in the Dashboard. , Enter "**Script console**".

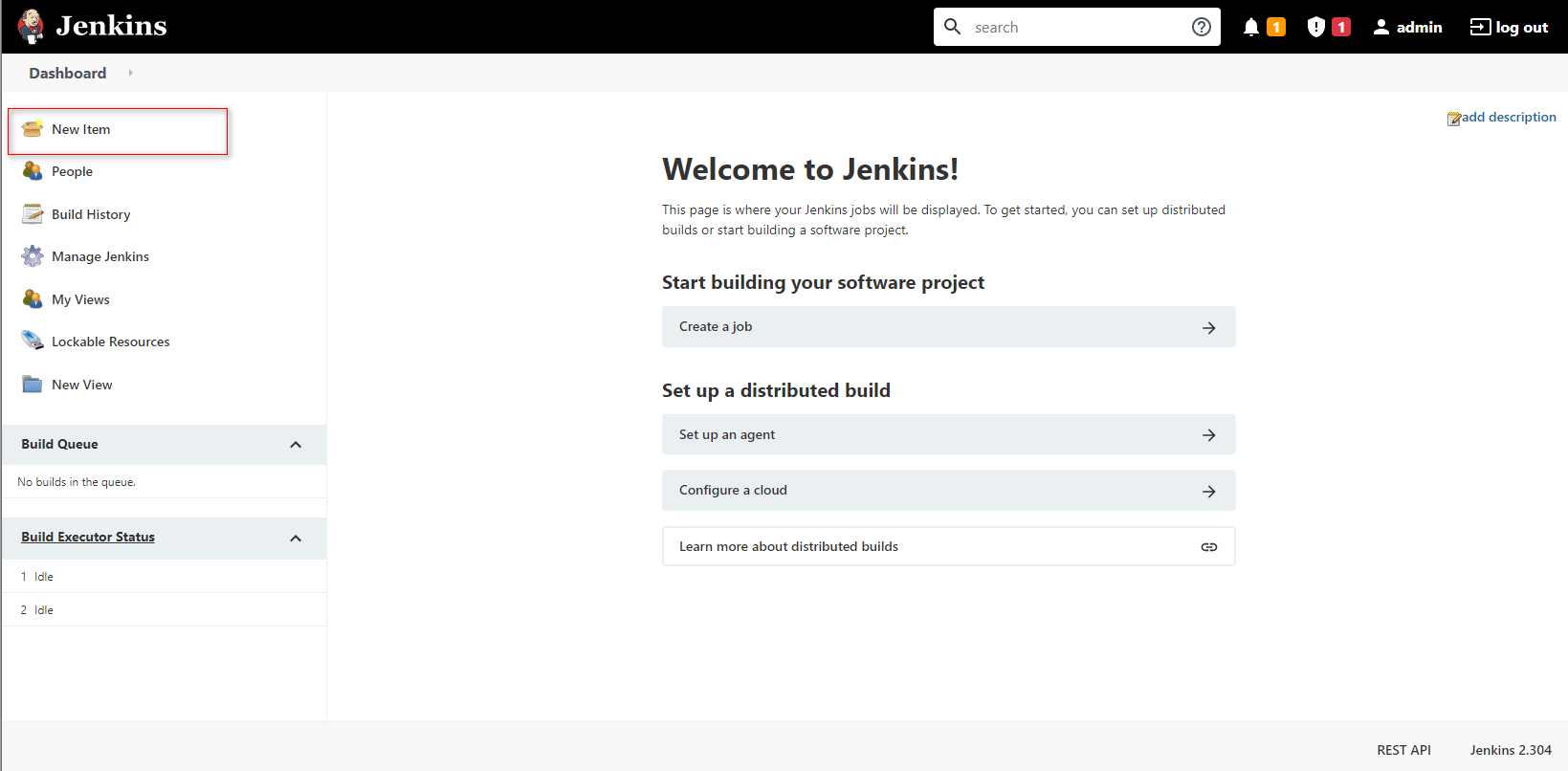


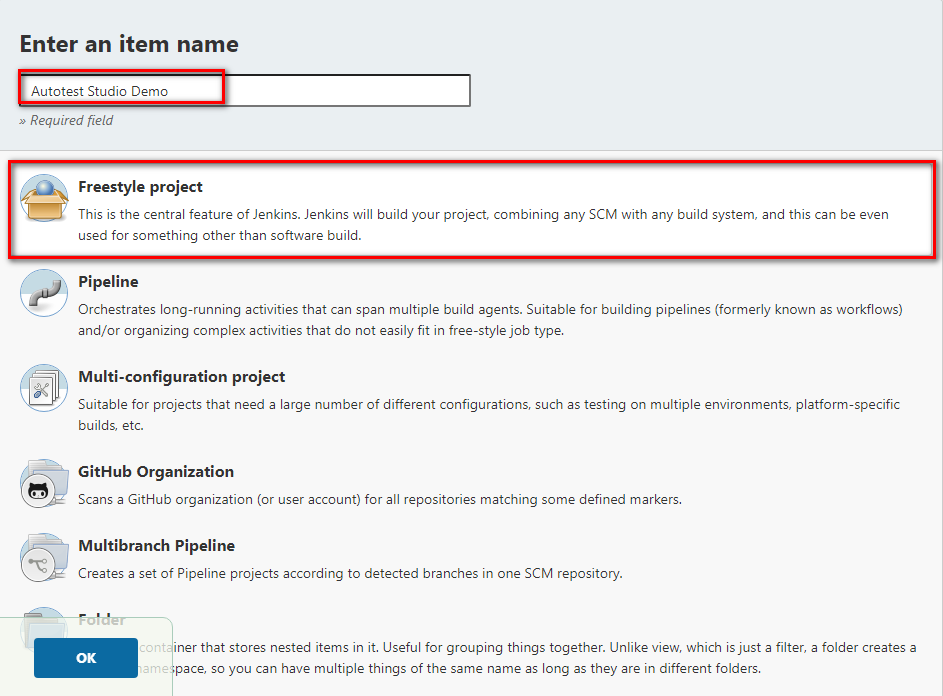
Run the code *System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "default-src \*'unsafe-inline''unsafe-eval'; script-src \*'unsafe-inline''unsafe-eval'; connect-src \*'unsafe-inline'; img-src \* data: blob:'unsafe-inline'; frame-src \*; style-src \*'unsafe-inline';")*, the running result is shown in the figure below.

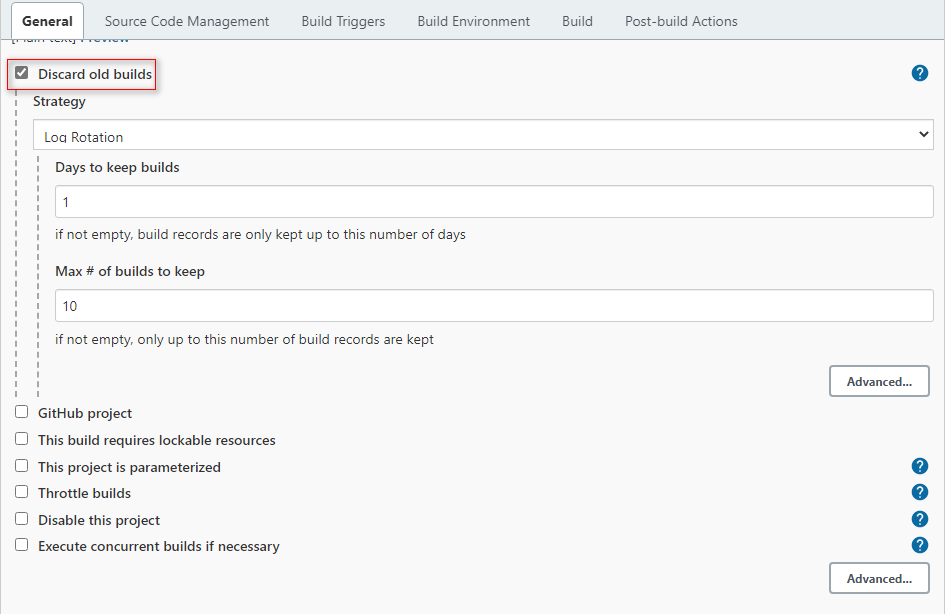


**Create a new item**

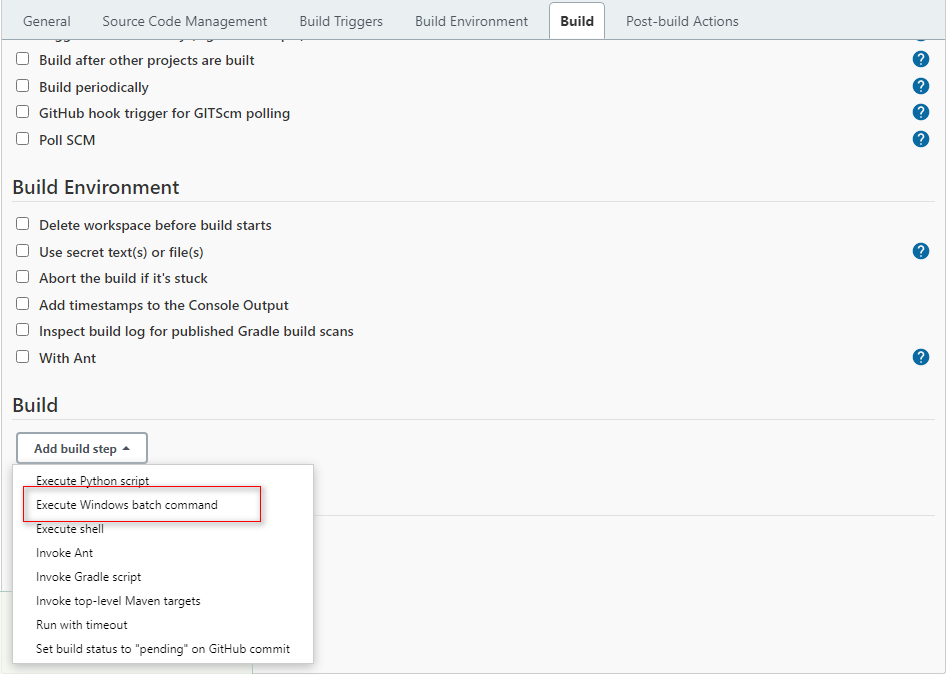
Create a new Item with the name "AutoTest Stdio Demo" and the style "**Freestyle project**".







Select "**Execute Windows batch command**" in Build to add AutoTest Studio call command.



Invoke the command "C:\Program Files\AutoTest Studio\AutoTest Studio.exe" "C:\Users\Demouser\Documents\AutoTest Studio\Projects\ats\_example\ats\_example.tpro" -r -q.

Command meaning:

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

**"C:\Users\Demouser\Documents\AutoTest Studio\Projects\ats\_example\ats\_example.tpro"**, the project path, you need to select the script to be executed in the project before the task runs, and AutoTest Studio will automatically save it to the project file.

**-r**, run the task immediately after startup. Required parameters.

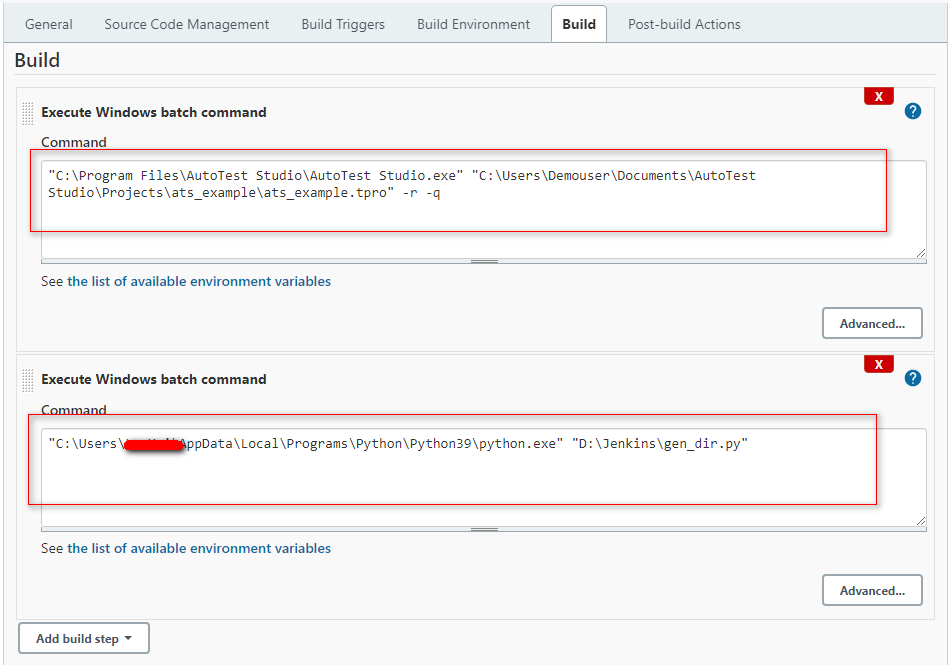
**-q**, after the task is completed, exit the AutoTest Studio program. Required parameters.

Since the report and log directories of AutoTest Studio are generated dynamically, we need to parse the log directory in the task output file "output.json" through a script and map it to the workspace.

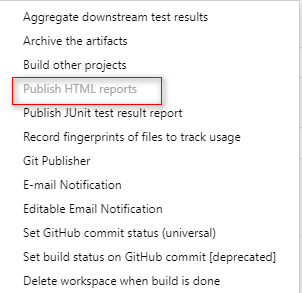
Save the following python code to a file, such as: "D:\Jenkins\jen\_dir.py", but there is no restriction on the location of the script.

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)

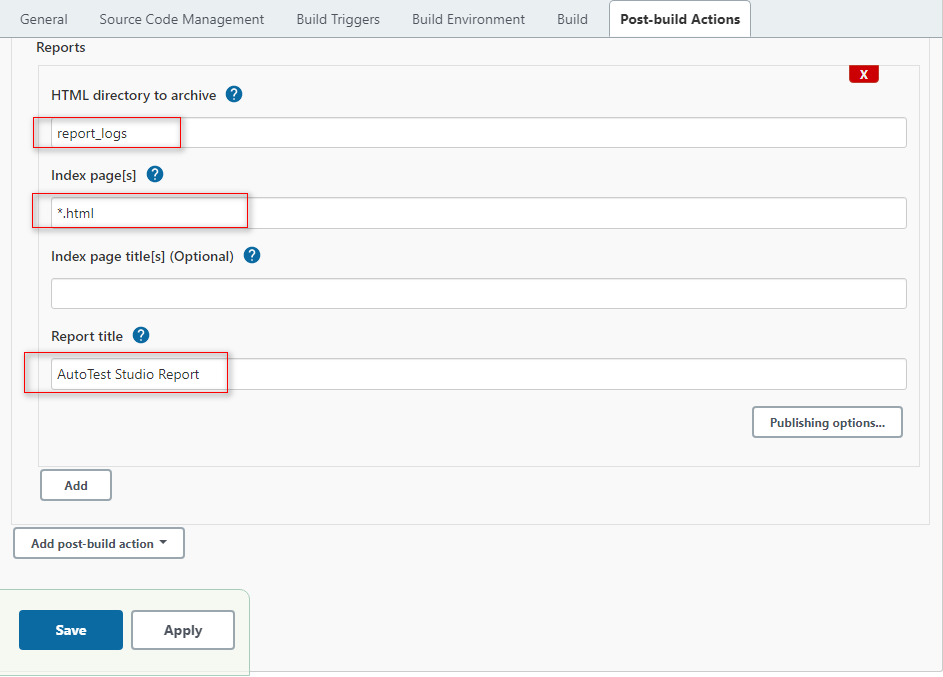
These two build commands are shown in the figure below.



In "**Post-build Actions**", use the "**Publish HTML reposts**" plug-in to send AutoTest Studio test reports to Jenkins.



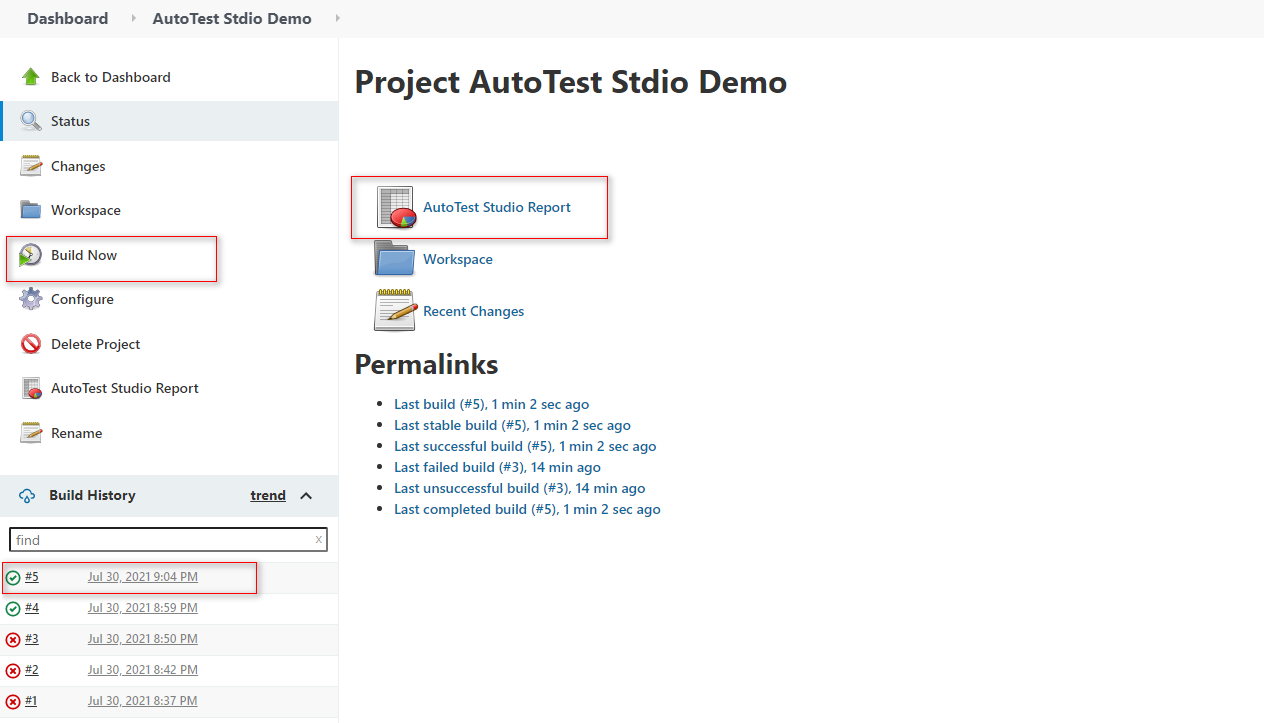
The "**Publish HTML reposts**" plug-in configuration parameters are as follows. Note that the directory "**report\_logs**" must be consistent with the commands in the above script.



After the configuration is complete, click the "Save" button at the bottom to save the configuration.

**Perform build tasks**

Click "Bulid New" in the Dashboard to perform a build operation.



After the build operation is completed, we click "**AutoTest Studio Report**" to open the test report, and you can see the content as shown in the figure below. This page is captured by the "Publish HTML report" plugin. Click "**Zip**" in the upper right corner of the page. Download the test report and log of this task run.

