

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps

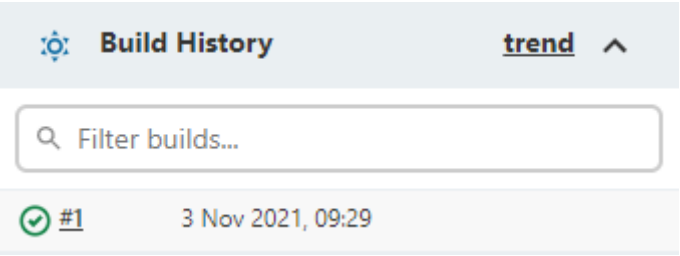
Builds

Contents

- [Overview](#)
- [Build Status](#)
- [Build Steps](#)
- [Console Output](#)
- [Post-build Actions](#)
 - [Artifacts](#)
 - [Accessing Artifacts](#)
- [Tutorial](#)
 - [Create a Job](#)
 - [Add a Build Step](#)
 - [Run the Job](#)
 - [Make the Build Fail](#)
 - [Abort a build](#)
 - [Fix the Build and Create Artifacts](#)
 - [Finish Up](#)
- [Exercises](#)

Overview

A build is a result of an execution of a project (job) in Jenkins. Builds for projects can be seen on the **Build History** section of the project dashboard:



Builds are named by the order that they were executed, so the first build ever executed will be called **#1** and the second **#2**, etc.

Build Status

The status of the build can have a few different values:

- **SUCCESS**: The build succeeded.
If all the build steps complete successfully, this will be the build status.
- **FAILURE**: The build failed.
If any of the steps exit with a non-zero status (if they throw an error), then the build status will go to failed.
- **ABORTED**: The build aborted before it finished.
This exit status is more uncommon; it must be set either by yourself or plugins that are being used in the project.

Build Steps

Build steps are effectively where you configure what your job is going to do. Depending on your situation, this could accomplish many different tasks:

- **Java**
If you have a Java project that you are wanting to automate builds for, you may be running commands or using plugins for running Maven, Gradle or Ant.

Jenkins Introduction

- Installation Wizard
- Jobs
- Freestyle Project
- Builds
- Plugins

Jenkins Pipeline

Markdown

IDE Cheatsheet

- **Docker**

Docker images can be built for your project.

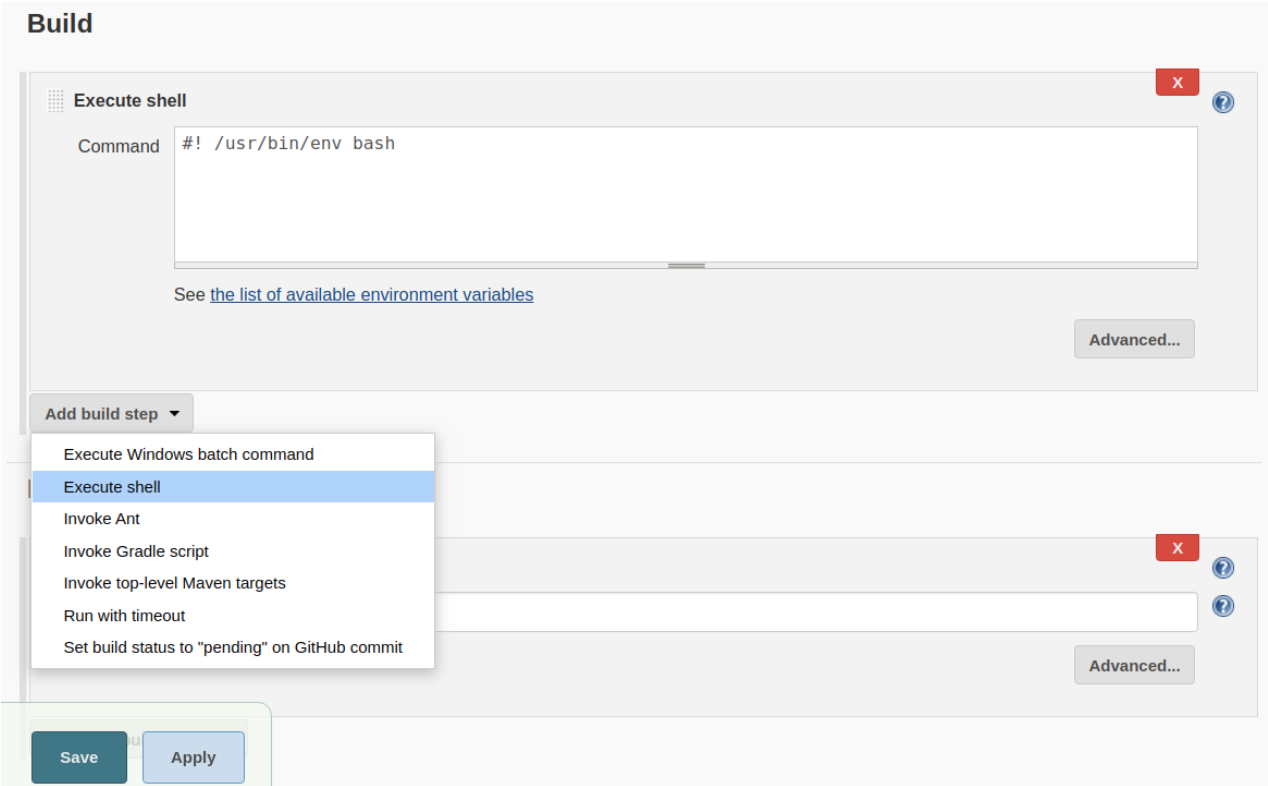
- **NodeJS**

Installing dependencies for your node application before packaging it up for deployment at a later date.

- **Any Other Platforms**

With the combination of shell scripting and plugins that are available for Jenkins, the sky truly is the limit with what you can automate in a Jenkins' build.

We can add build steps by selecting the **Add build step** drop-down button and then selecting the type of build step that we would like to use:



The type of build steps that are available will depend on the Jenkins' Plugins that you have installed.

For the example shown above, we can see that some plugins must be installed for Gradle, Maven and GitHub.

Most of the time you can get what you need out of a build step from shell scripting.

It's preferable to separate your shell scripts into different build steps, to keep you project more maintainable.

For example you could have separate build steps for compiling, testing and deployment.

Console Output

The console output is likely one of the main parts of a build that you will be checking for information and debugging purposes.

This section includes the output for any shell scripts and plugins that have been executed in the build step for a project

The console output can be navigated to quickly by selecting the blue or red bulb (🟢) on the build history.

Here is a very simple console output example:



Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/builds
[builds] $ /bin/sh -xe /tmp/jenkins10109130200644771413.sh
+ echo test
test
Finished: SUCCESS
```

Post-build Actions

Post-build actions are plugins that can be run after you execute all the build steps.

You can add a post-build action by selecting the **Add post-build action** drop-down button and then selecting a post-build action from the list.

Theses actions can be set to run, or not run, depending on the success of the build steps.

Even if the build failed, you can still run post-build actions as an immediate response to the build failing.

For instance, you may want to send an email notification on a build failure.

If the build succeeds, you might want to store the built application files as an artifact for access later on.

Artifacts

Build artifacts are immutable files that are created as result of the build. You have complete control over which files you would like to store as an artifact, and this can be configured in the **Post-build Actions** section of the project configuration.

The example here will archive any files and store them as artifacts for the project:



Accessing Artifacts

Once the job has executed, the artifacts are then available on the job dashboard:

Project My Project



[Workspace](#)



[Last Successful Artifacts](#)



[artifact-1571670785.zip](#)

712 B [view](#)



[Recent Changes](#)

Tutorial

We'll have a look at creating some builds here, focusing on build step configuration, statuses and storing artifacts.

Create a Job

First, we will need a new Jenkins job to work on, so create one called anything that you like.

Add a Build Step

Let's add a build step that we know will succeed. Select the **Add build step** button and add an **Execute shell** build step.

Add the following script into the **Command** field:

```
#!/bin/bash
echo "Hello from the Jenkins job named: ${JOB_NAME}"
```

Run the Job

Save the job and then build it; you should then have one successful build in your history for that job.

Once you navigate to the console output, you should see an output like this:

Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/builds
[builds] $ /bin/bash /tmp/jenkins4212785633751696761.sh
Hello from the Jenkins job named: builds
Finished: SUCCESS
```

Make the Build Fail

Now that the last build succeeded, let's see what a failed build looks like!

All we have to do is make it so that the script we added to the command box "fails".

Jenkins will treat any script or application exiting with a non-zero status as a failure.

So to create a failed build, let's add `exit 1` to the script box, which will make the script exit with a code of 1:

```
#!/bin/bash
echo "Hello from the Jenkins job named: ${JOB_NAME}"
exit 1
```

Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/builds
[builds] $ /bin/bash /tmp/jenkins15479069577723093425.sh
Hello from the Jenkins job named: builds
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

Abort a build

We can, of course, remove the `exit 1` from the build step to fix it.

We are now going to make the build run for 10 minutes, this will give us plenty of time to abort it.

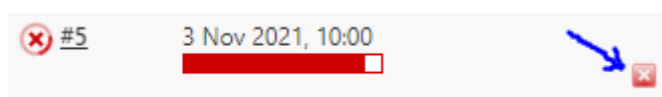
Here is the code that should be in your script box now:

```
#!/bin/bash
echo "Hello from the Jenkins job named: ${JOB_NAME}"
sleep 600
```

Click **Save** and **Build Now** to begin the build.

We then want to abort the build, we do this by clicking the red **x** on the in-progress build.

Note: While the build is in progress, it will have the status of the previous build, but with a spinning icon.



This will result in an **aborted** build, the console output will look like this:

Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/builds
[builds] $ /bin/bash /tmp/jenkins15945283049457593992.sh
Hello from the Jenkins job named: builds
Terminated
Build was aborted
Aborted by admin
Finished: ABORTED
```

Fix the Build and Create Artifacts

We can, of course, remove the `sleep 600` from the build step so it doesn't take 10 minutes to complete.

After that, let's change the script to create several files, and then put them in a zip archive called `archive.zip`:

```
#!/bin/bash
echo "Hello from the Jenkins job named: ${JOB_NAME}"
touch 1.txt 2.txt 3.txt 4.txt 5.txt
zip archive.zip *.txt
```

Next, a `Post-build Action` must be configured to archive the zip files. Please refer to the [Artifacts](#) section for configuring this.

Finish Up

Now try to run the job. You should see artifacts on the project dashboard. If they don't show up, try refreshing the page.

Exercises

There are no exercises for this module.