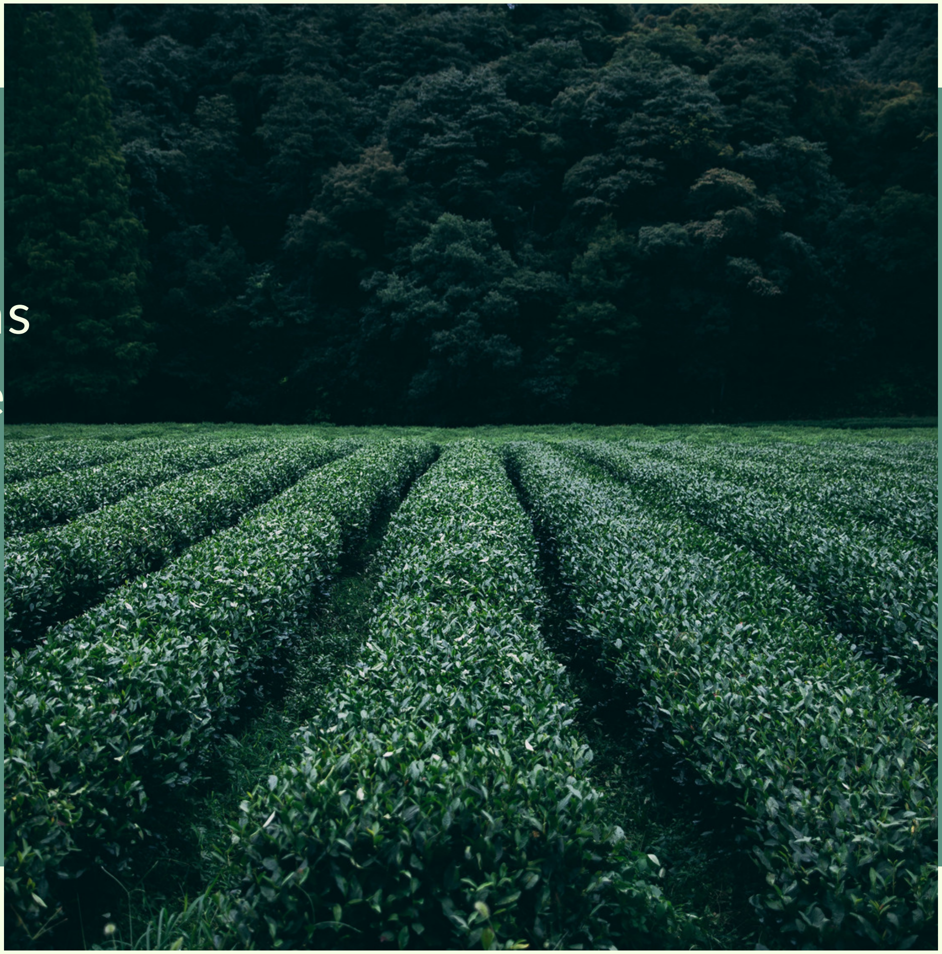
Build Triggers

JENKINS

BY ALOK SRIVASTAVA - NETWORK NUTS

BUILD TRIGGERS



While working with Jenkins, there are different ways we can tell Jenkins to run our projects, Manually. While working on real projects, we need more automated ways to run our builds. Build triggers help to make this possible.

There are different ways we can achieve this:

- Starting a build job when another job has been completed using upstream/downstream projects
- Running builds at periodic intervals
- Polling source code management for changes
- Triggering builds remotely

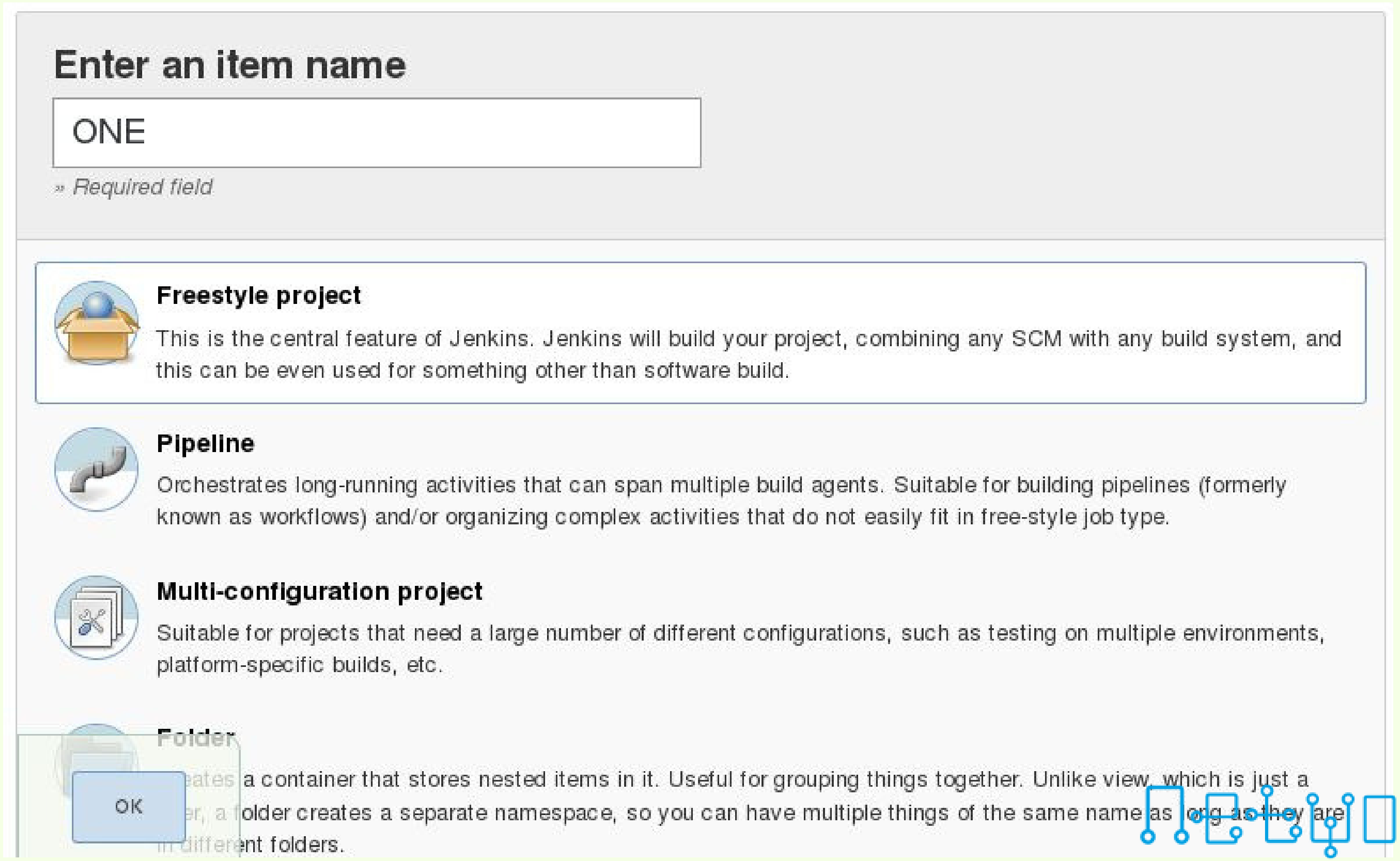
Creating Upstream / Downstream Projects

The first build trigger mechanism mentioned above involves creating two inter-linked build tasks.

We will create two projects and using one to trigger the other.

Two projects, project ONE and project TWO, if project TWO is configured to run once project ONE completes, we call project TWO the downstream project and project ONE the upstream project.

Create project ONE as a simple "parameterized" project with a variable "valone" having default value as "project one".



Enter an item name

ONE

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline

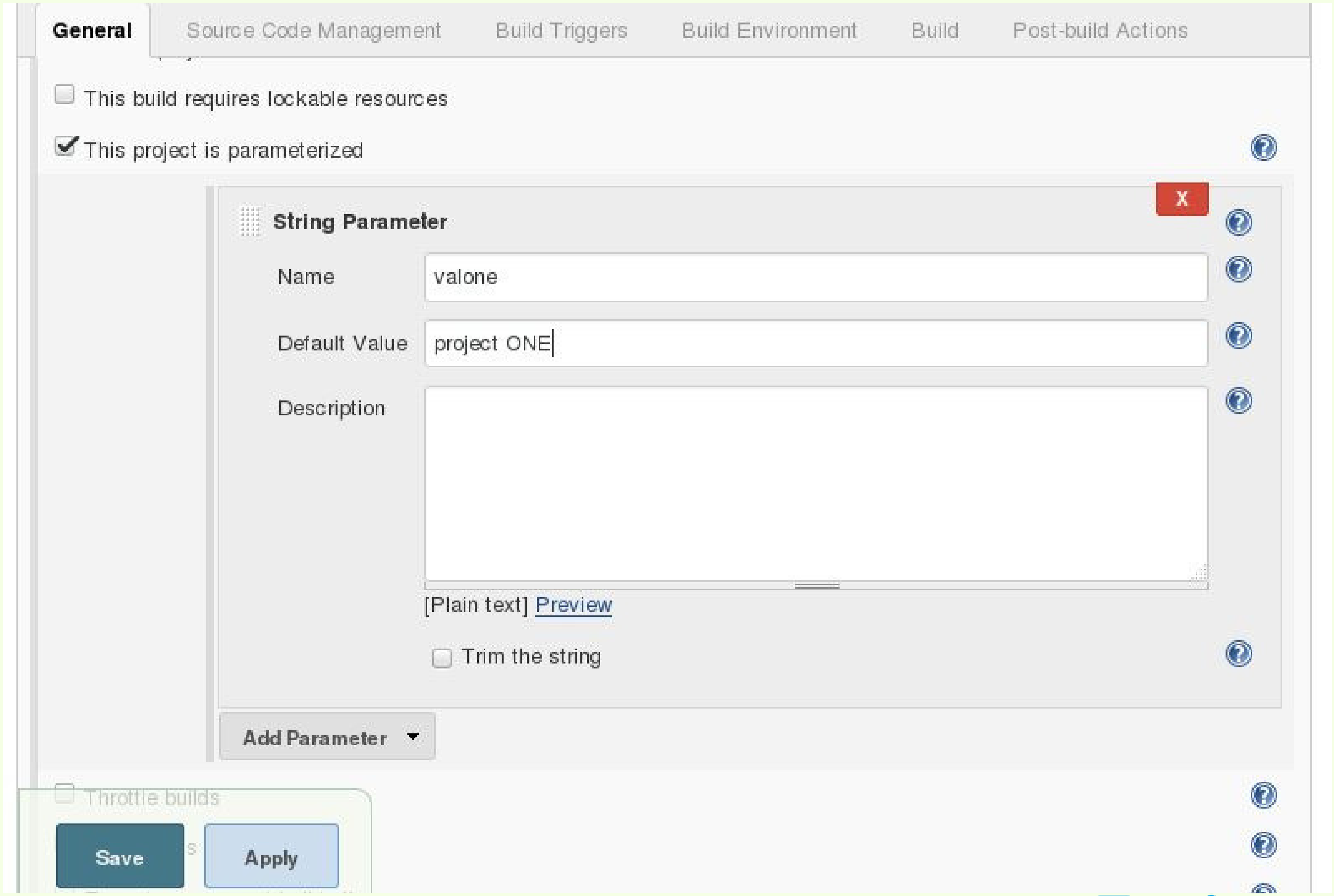
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder

eates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a er, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.


General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

☐ This build requires lockable resources

☑ This project is parameterized

String Parameter

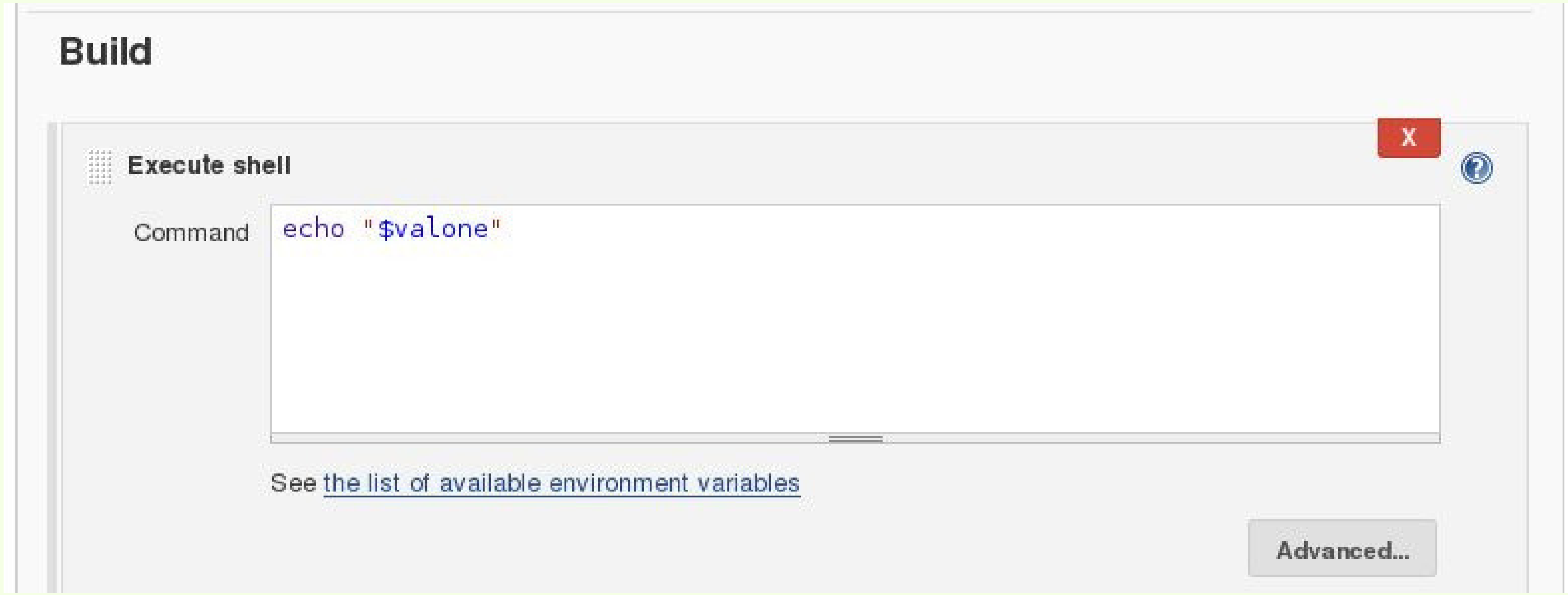
Name: valone

Default Value: project ONE

Description:

[Plain text] Preview

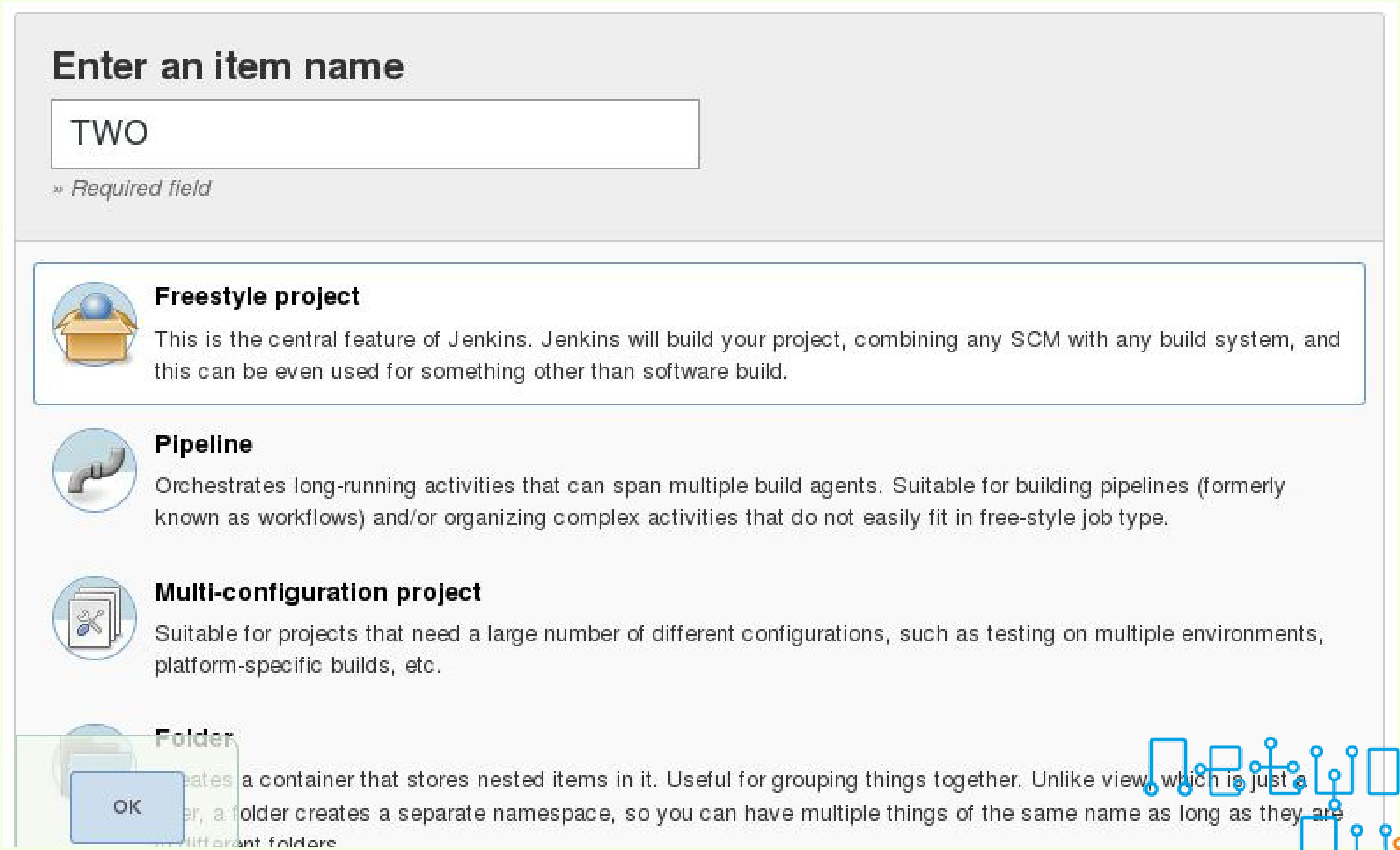
☐ Trim the string



```
echo "$valone"
```

Select Apply and Save at the end of the page.

Go back to the dashboard and create Project TWO the same way we set up Project ONE, but with different parameter identifier.


Enter an item name

TWO

» *Required field*

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline

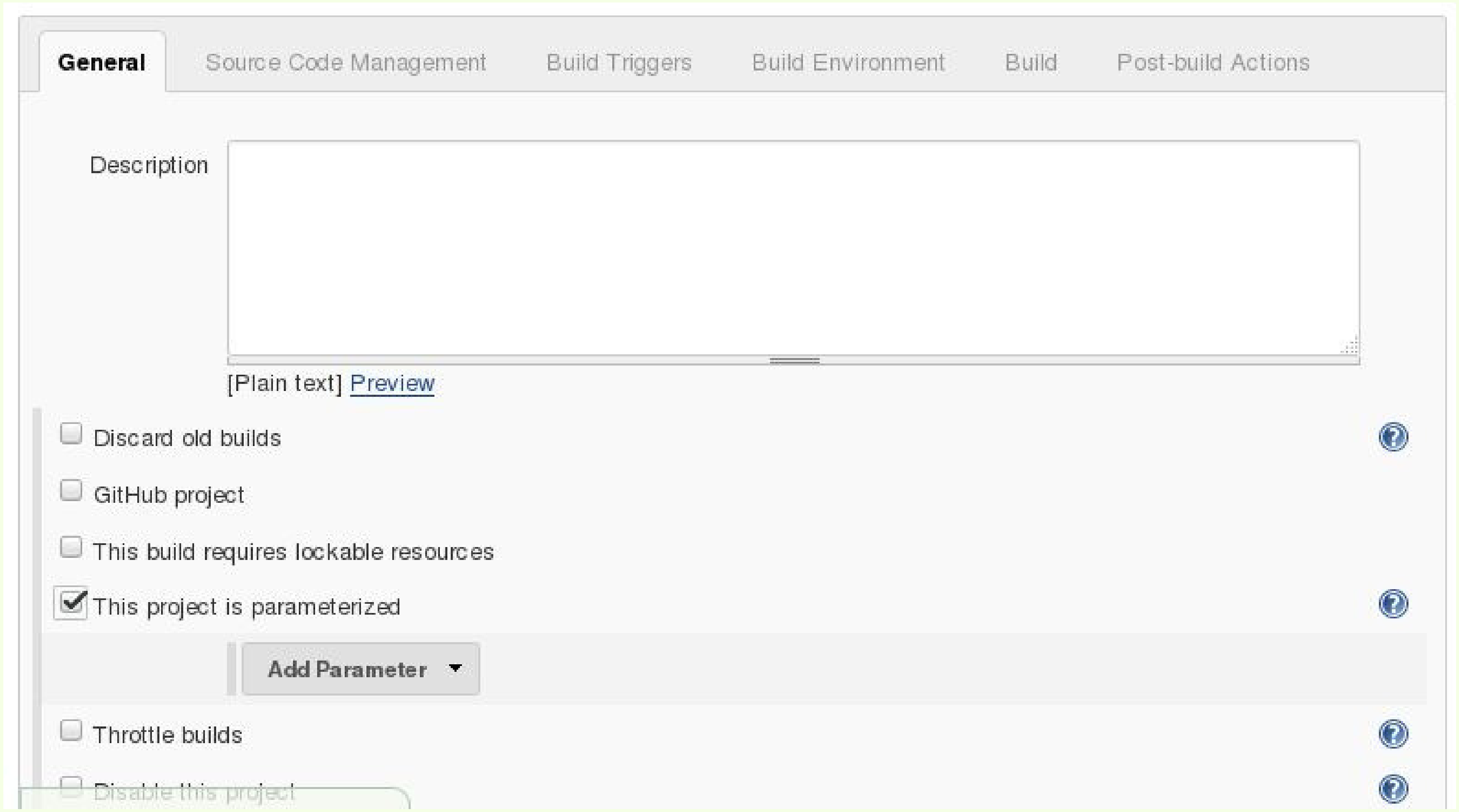
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder

eates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a er, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders



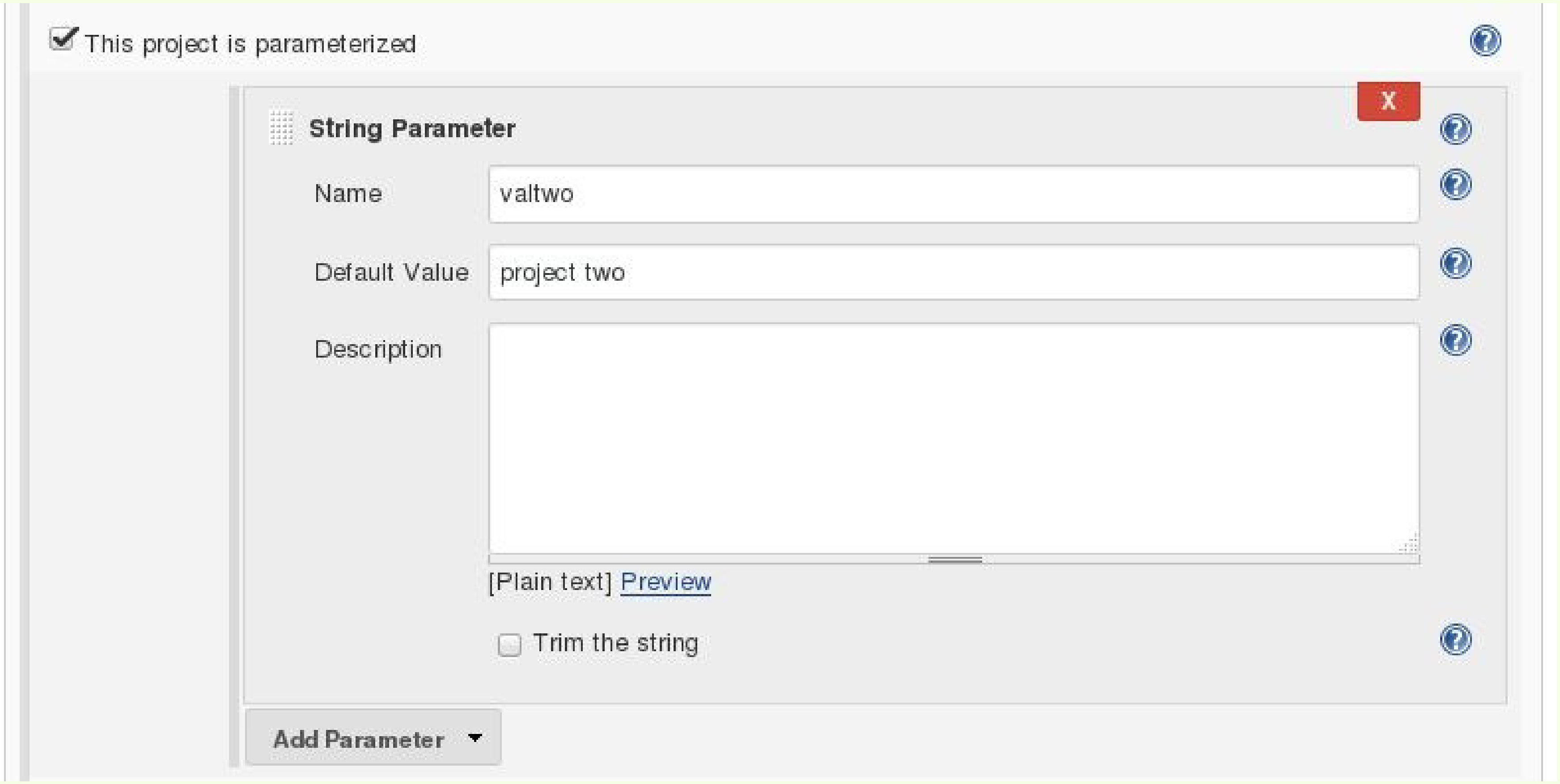
General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Description

[Plain text] Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] This build requires lockable resources
- [x] This project is parameterized

- [ ] Throttle builds
- [ ] Disable this project



☑ This project is parameterized

String Parameter

Name: valtwo

Default Value: project two

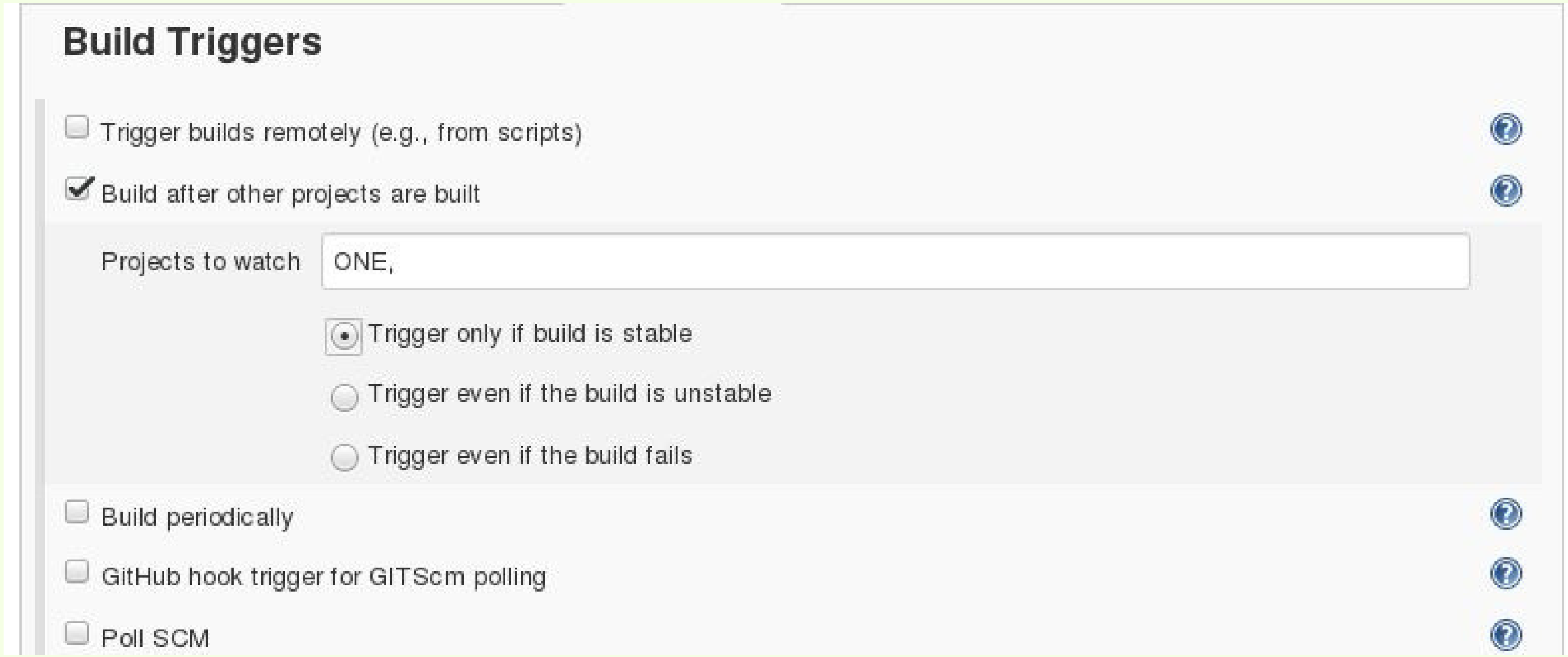
Description:

[Plain text] Preview

☐ Trim the string

Before we add our build step, we have to configure our project as a downstream project. Under Build Triggers in the project configuration, select Build after other projects are built.

After we select the **Build after other projects are built** option, we are presented with a text field in which we have to enter the upstream project, that is, Project ONE. Start typing the name of the upstream project and Jenkins will autocomplete and filter with the projects that match what you want to specify.



Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [✓] Build after other projects are built

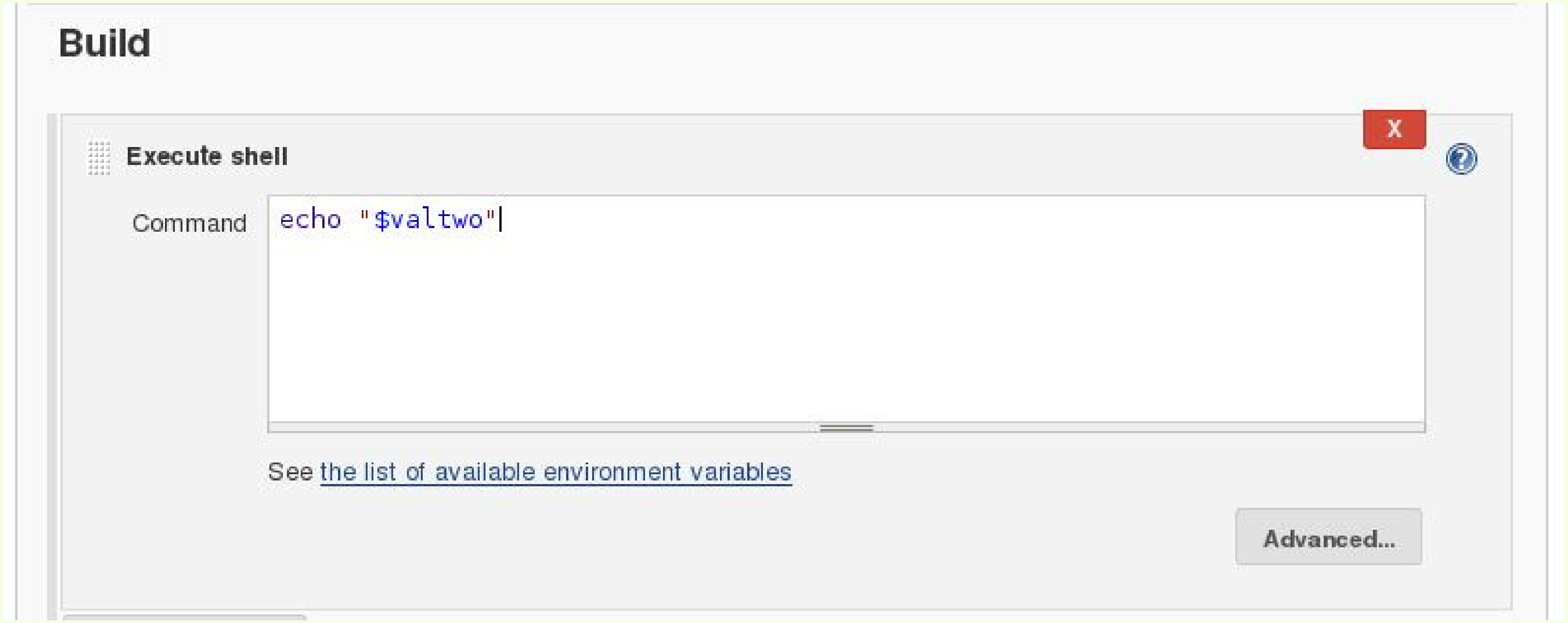
Projects to watch: ONE,

- ● Trigger only if build is stable
- ○ Trigger even if the build is unstable
- ○ Trigger even if the build fails

- [ ] Build periodically
- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

You can configure multiple upstream projects, thus you will notice a comma after Project ONE in the Projects to watch text field.

Finally, add a build step to execute a shell script that will output the parameter we added earlier.



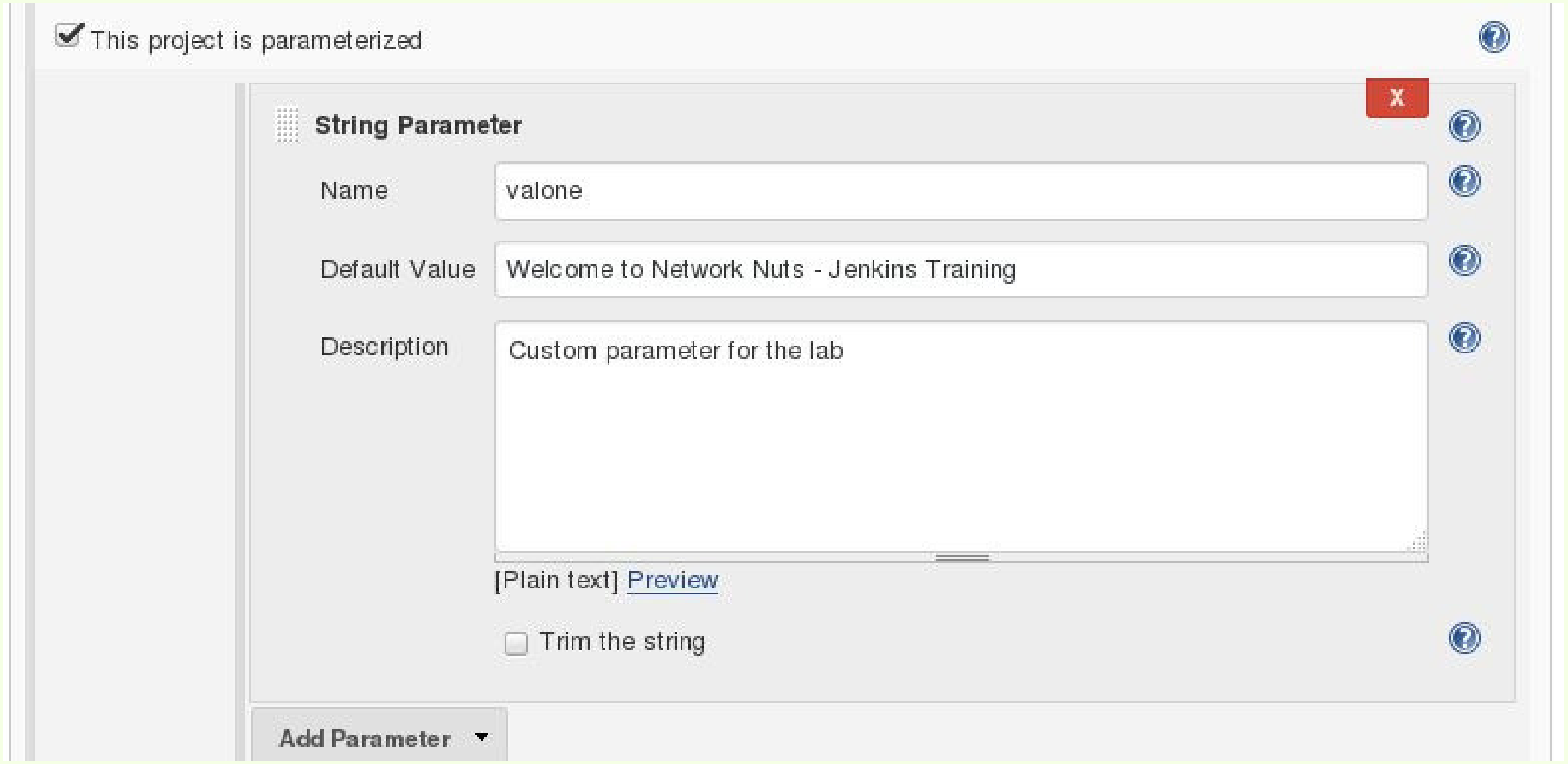
Command

```
echo "$valtwo"
```

See the list of available environment variables

Click **Apply and Save** to persist the build configuration you just made.

Before we run our projects, we need to make one more configuration change on our upstream project.



[✓] This project is parameterized

String Parameter

Name: valone

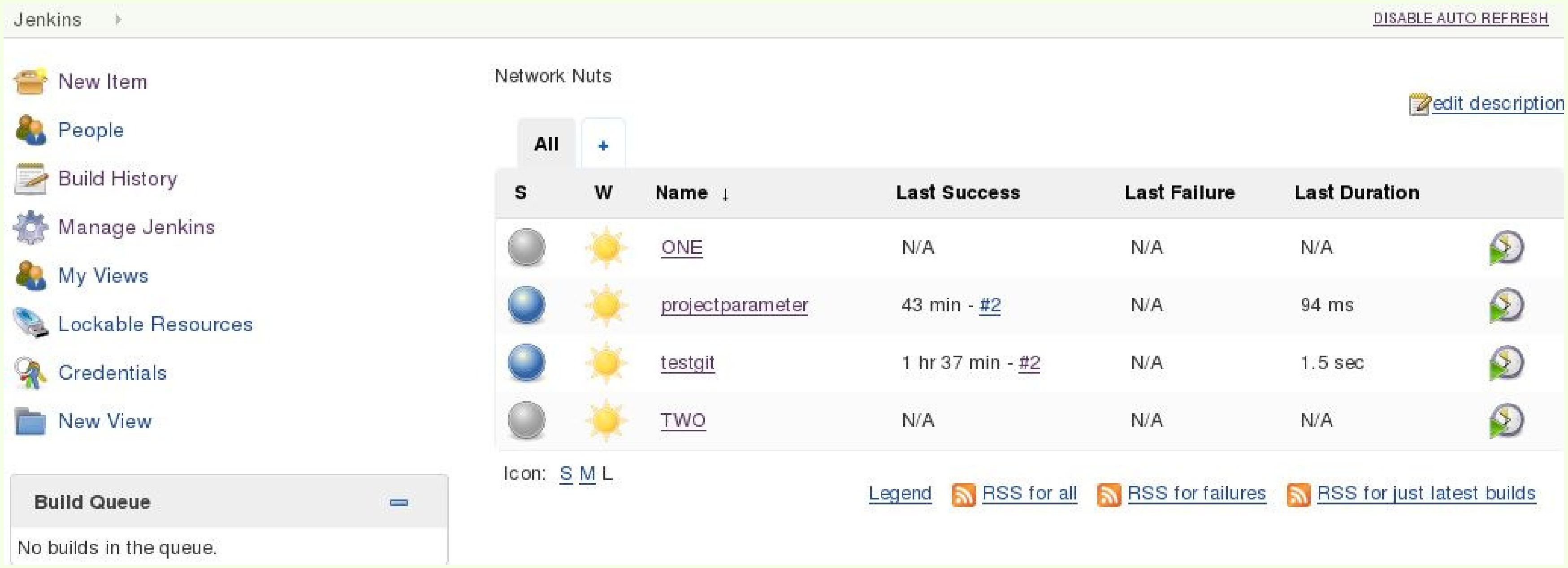
Default Value: Welcome to Network Nuts - Jenkins Training

Description: Custom parameter for the lab

[Plain text] Preview

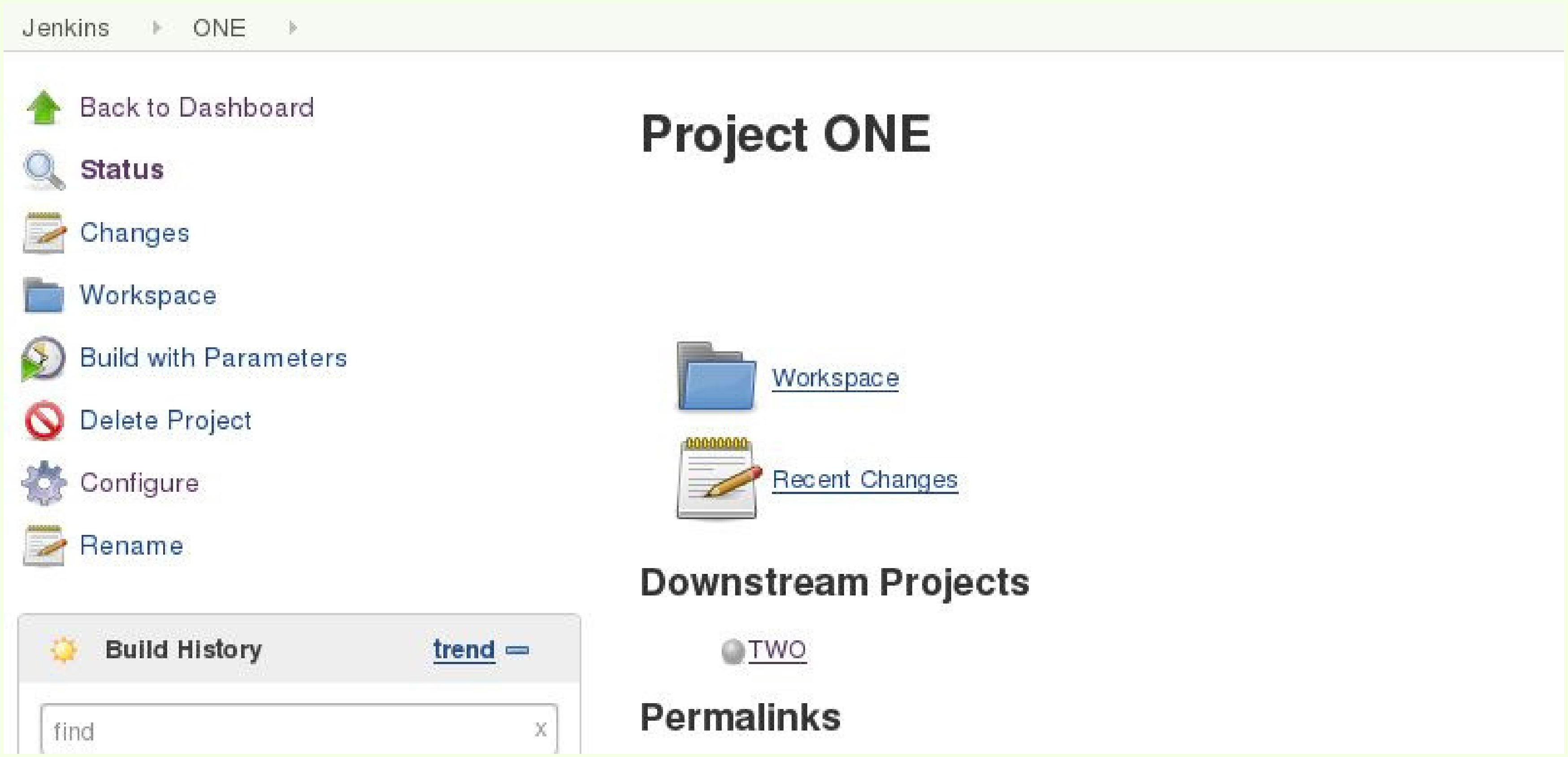
[ ] Trim the string

While on the Jenkins dashboard, select Project ONE. On the left configuration panel, click on **Configure** and this will open the now familiar project configuration page.



No builds in the queue.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		ONE	N/A	N/A	N/A
		projectparameter	43 min - #2	N/A	94 ms
		testgit	1 hr 37 min - #2	N/A	1.5 sec
		TWO	N/A	N/A	N/A


Jenkins ▸ ONE ▸

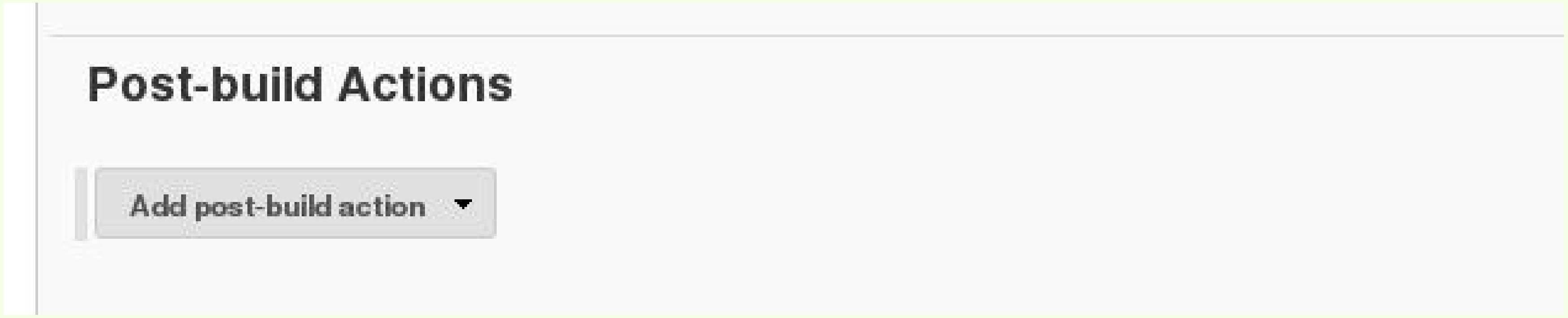
Project ONE

Downstream Projects

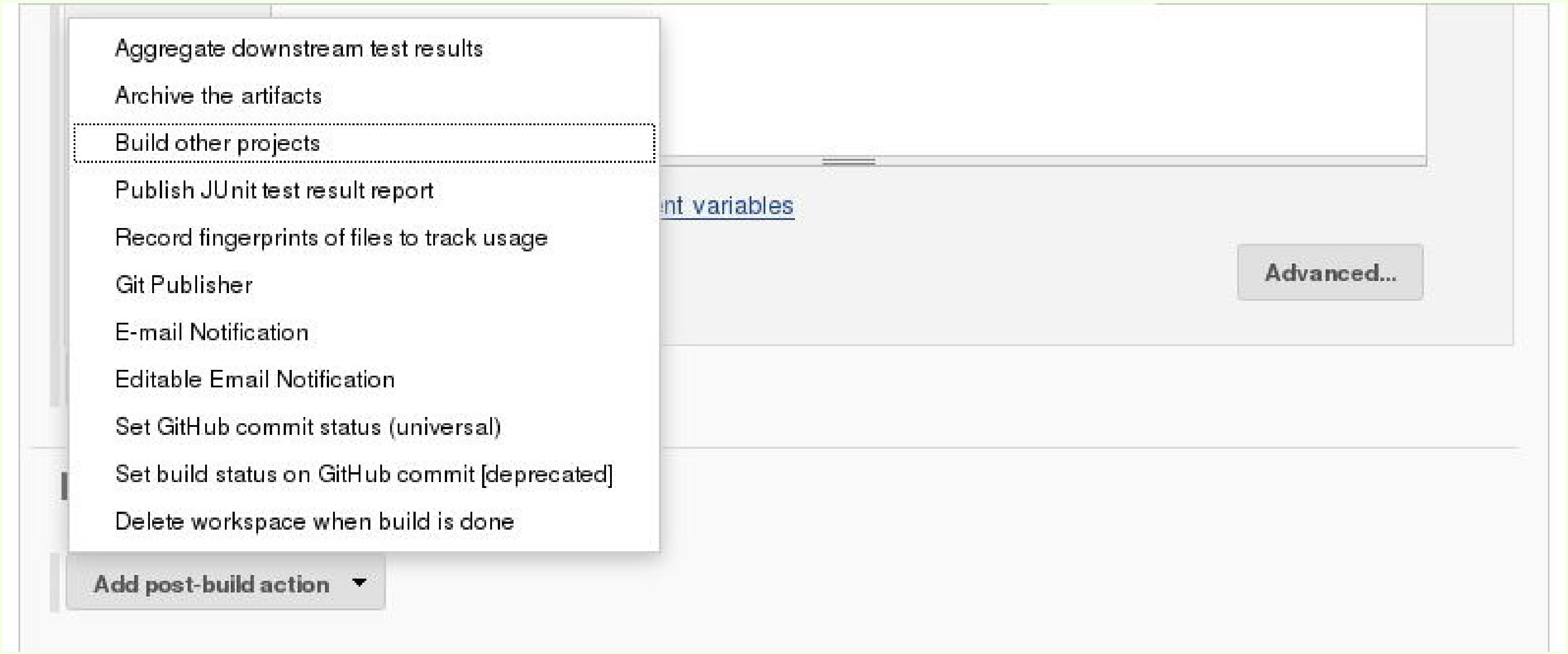
TWO

Permalinks

Scroll to the bottom to get to the **Post-build Actions** section. Click on Add post-build action and select **Build other projects**, as shown:

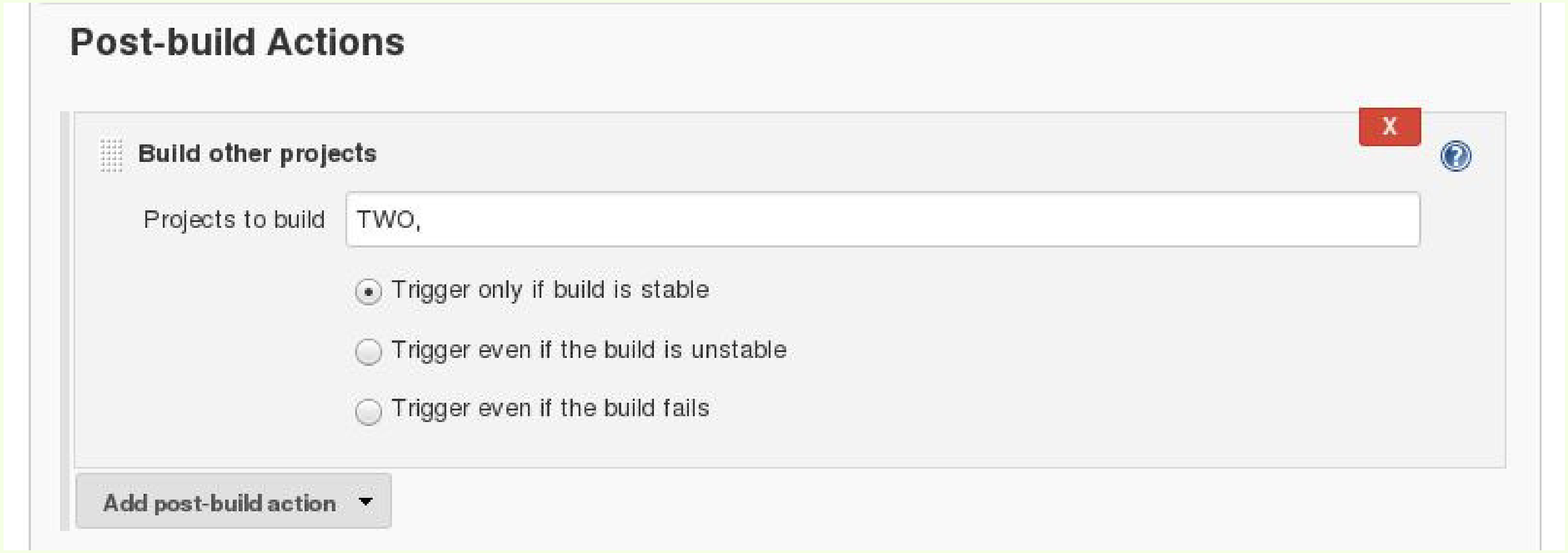

Post-build Actions

Add post-build action



Aggregate downstream test results
Archive the artifacts
Build other projects
Publish JUnit test result report
Record fingerprints of files to track usage
Git Publisher
E-mail Notification
Editable Email Notification
Set GitHub commit status (universal)
Set build status on GitHub commit [deprecated]
Delete workspace when build is done
nt variables
Advanced...
Add post-build action

Enter **Project TWO** in Projects to build and select the Trigger only if build is stable radio button.


Post-build Actions

Build other projects

Projects to build: TWO,

[● Trigger only if build is stable]
○ Trigger even if the build is unstable
○ Trigger even if the build fails

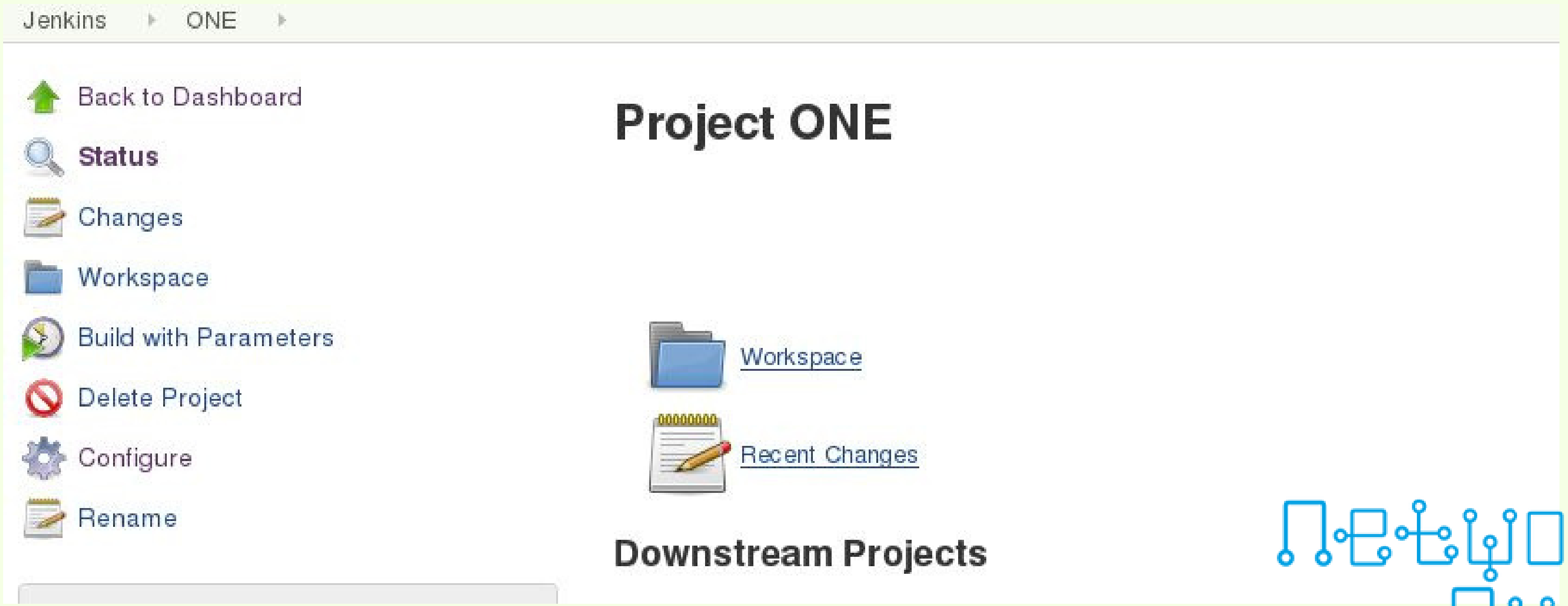
Select **Apply and Save** to save the configuration.

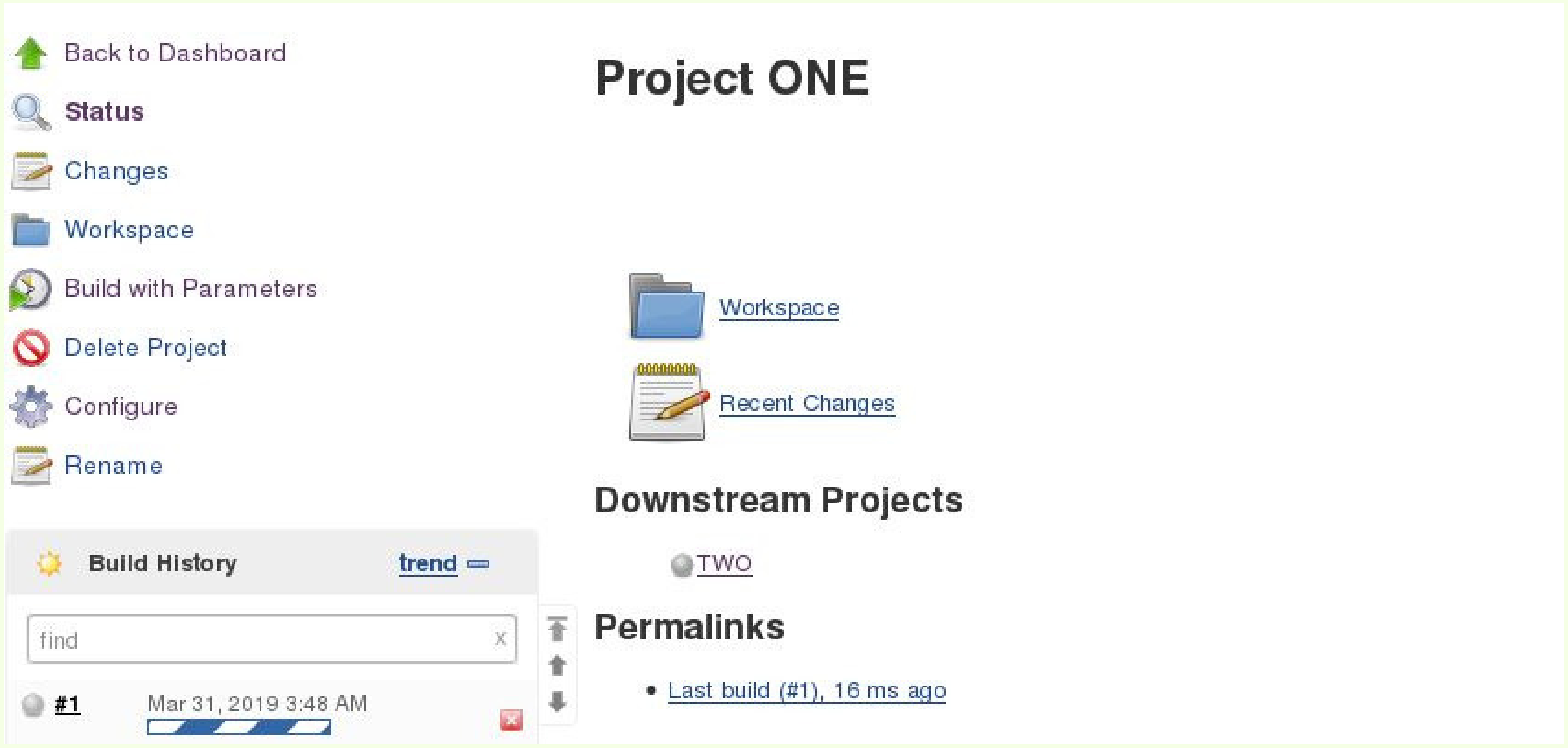
At this point, we have completed our upstream/downstream configuration.

Running an Upstream Project

Let's try to run an upstream project in Jenkins.

Jenkins dashboard, click on Project ONE, and, on the left-hand **configuration** panel, select **Build with parameters**. On the Build with Parameters page, click Build.





Back to Dashboard
Status
Changes
Workspace
Build with Parameters
Delete Project
Configure
Rename

Build History trend

find

#1 Mar 31, 2019 3:48 AM

Project ONE

Workspace

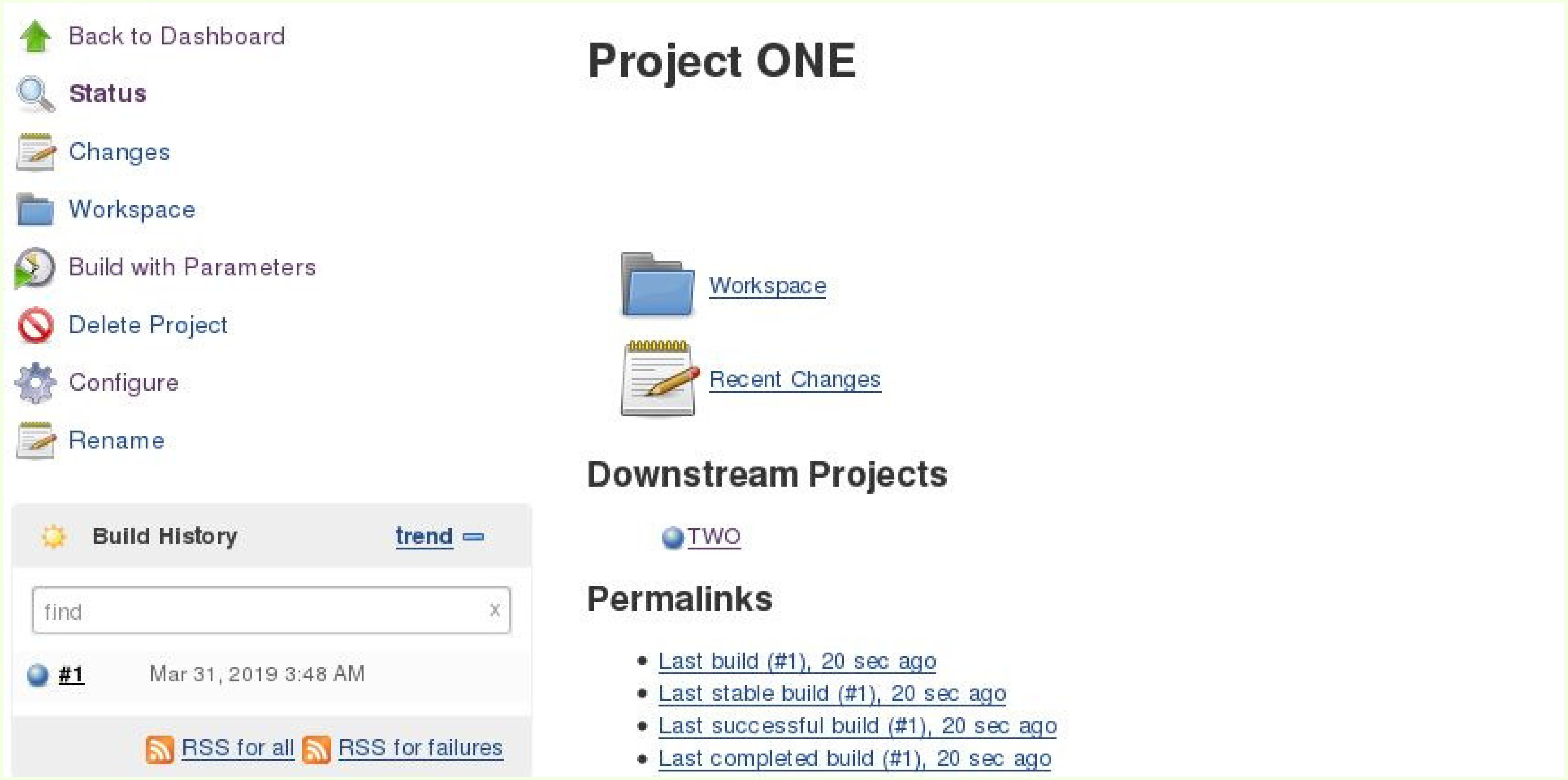
Recent Changes

Downstream Projects

TWO

Permalinks

- Last build (#1), 16 ms ago



Back to Dashboard
Status
Changes
Workspace
Build with Parameters
Delete Project
Configure
Rename

Build History trend

find

#1 Mar 31, 2019 3:48 AM

RSS for all RSS for failures

Project ONE

Workspace

Recent Changes

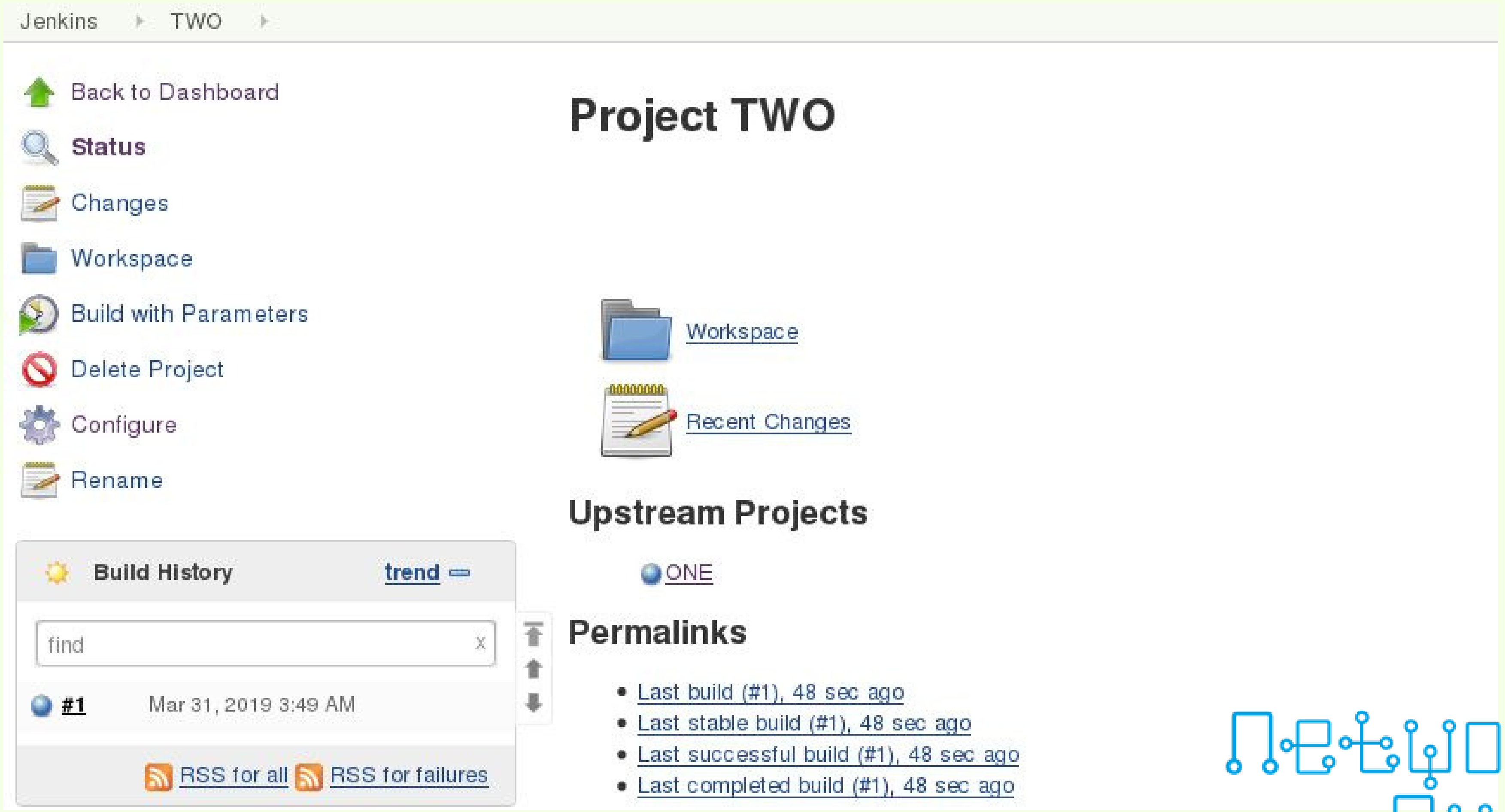
Downstream Projects

TWO

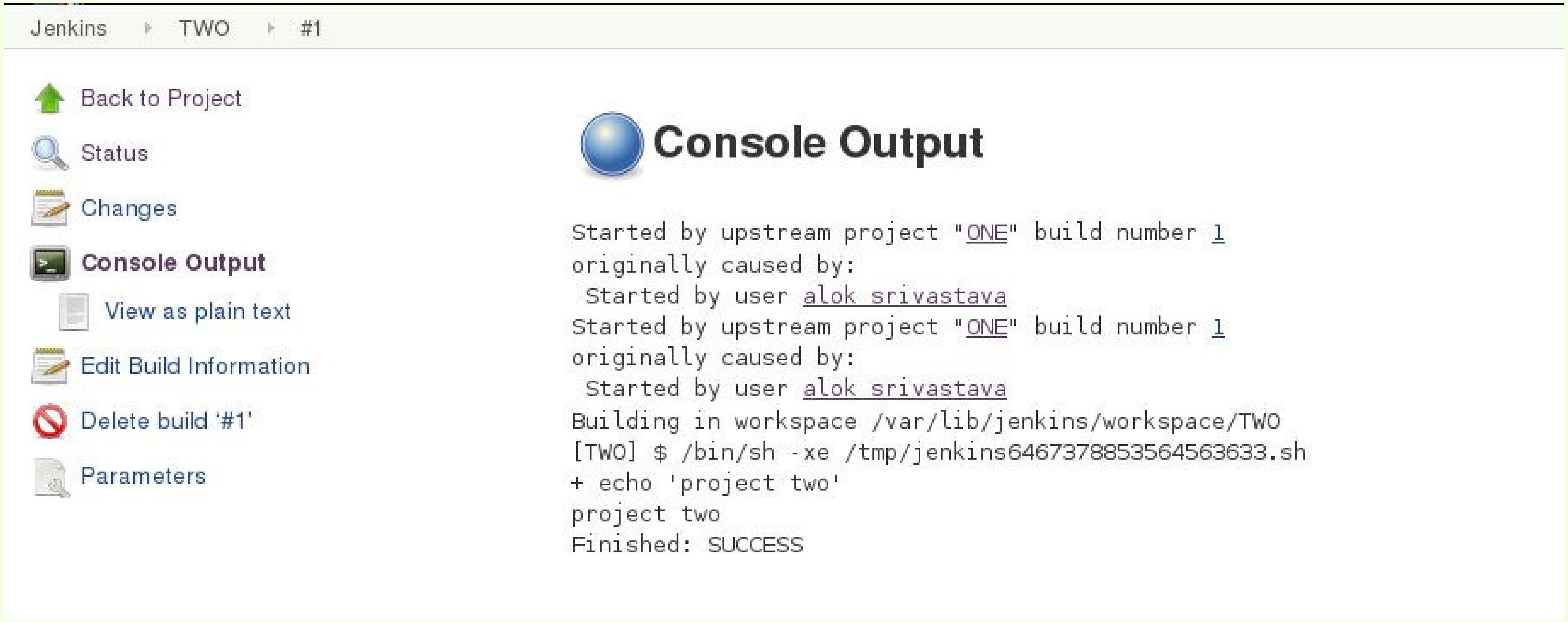
Permalinks

- Last build (#1), 20 sec ago
- Last stable build (#1), 20 sec ago
- Last successful build (#1), 20 sec ago
- Last completed build (#1), 20 sec ago

Select Project TWO under the Downstream Projects section. While on Project TWO, go the Build History section on the left and hover over the latest build, click the down arrow and select Console Output.



Looking at the output, we can see that the build of Project TWO was triggered by its upstream project. The output also informs us that Project ONE was run by the user.



Console Output

```
Started by upstream project "ONE" build number 1
originally caused by:
 Started by user alok srivastava
Started by upstream project "ONE" build number 1
originally caused by:
 Started by user alok srivastava
Building in workspace /var/lib/jenkins/workspace/TWO
[TWO] $ /bin/sh -xe /tmp/jenkins6467378853564563633.sh
+ echo 'project two'
project two
Finished: SUCCESS
```