

# SpoonScript Plugin

## Plugin Information

View TurboScript [on the plugin site](#) for more information.

The SpoonScript plugin for Jenkins allows to automatically build Spoon images and push them to the Spoon.net Hub or local repository.

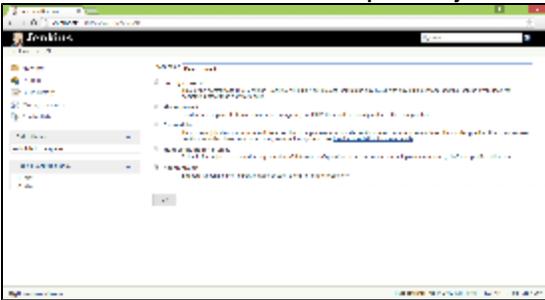
The SpoonScript Jenkins plugin includes the following capabilities:

- Build an image with SpoonScript
- Push the resulting image to the Spoon.net Hub
- Export an image to the local repository
- Trigger a build job using webhooks

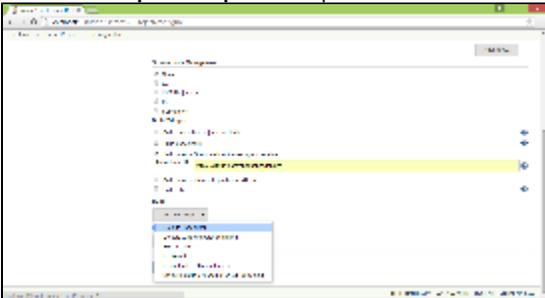
## Quick Tour

### Build an image with SpoonScript

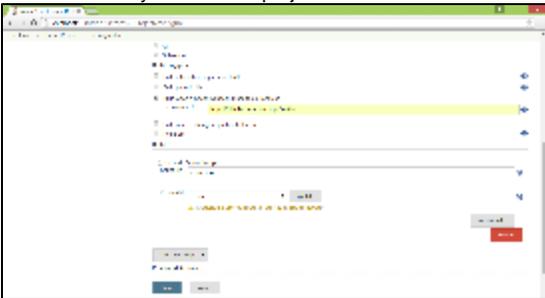
1. Create a new Jenkins item and select a **Spoon Project** as its type.



2. Add **Execute SpoonScript** build step.

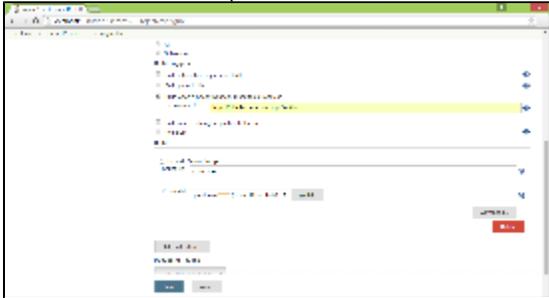


3. By convention, the default name for a SpoonScript file is **spoon.me**. It should be located in the Jenkins project workspace which corresponds to the root directory of a GitHub project.

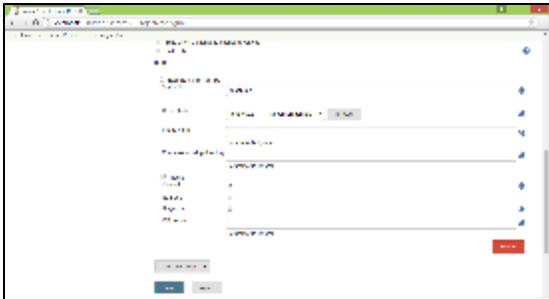


4. Write a SpoonScript to define how a new image should be created. For more information about SpoonScript refer to the [language reference](#) or take a look at sample scripts in [GitHub](#).

5. Provide credentials to a Spoon.net account. If no credentials are provided, the SpoonScript will be run as the currently logged on user.



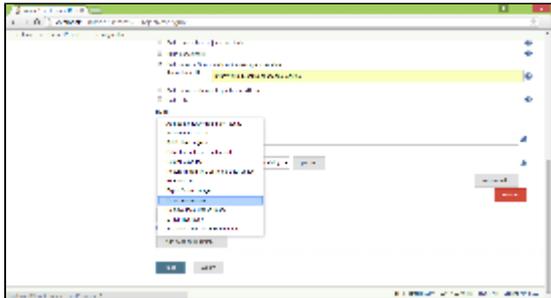
6. For enterprise deployments you may be interested in building images that target a specific version of the Spoon VM. To gain access to this and other optional settings, click **Advanced** button. For more information about specific options, click buttons with help content or refer to the [online documentation](#).



7. Save your changes and test the script by triggering the job manually. In the next sections you will learn how to push new images to the Spoon.net Hub.

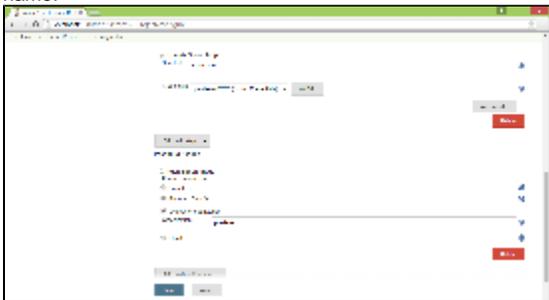
### Push an image to the Spoon.net Hub

Pushing images to the Spoon.net Hub is available as post-build action:

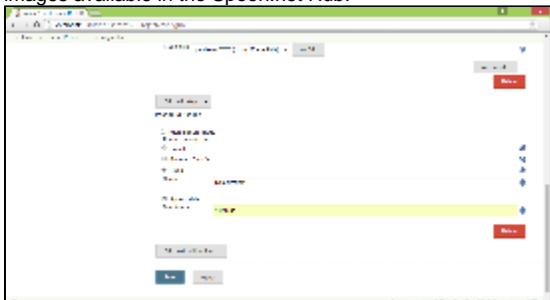


There are three options for naming the new remote image:

1. Use the same name as the local image.
2. Generate the remote image name from the Git information for your project (organization, project, branch, and current head version). Note that you cannot publish images from an organization to which you do not belong. For example, if your Spoon.net account name is **JimmyKimmel** then you won't be allowed to push images as **OprahWinfrey**. If this is required, then overwrite the name of the GitHub organization with your account name.

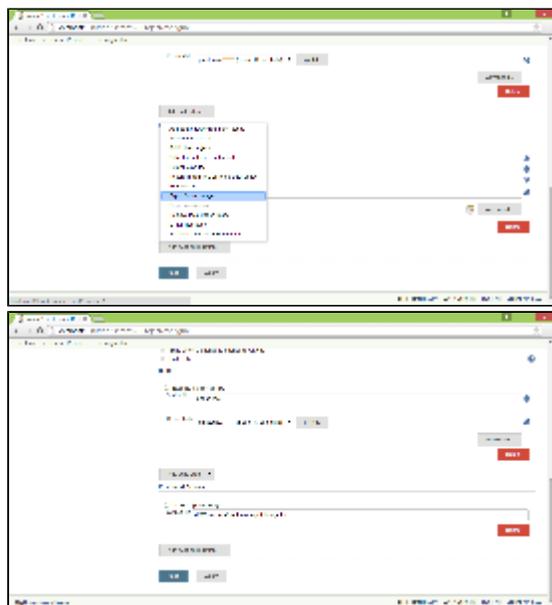


3. Use a fixed remote image name hardcoded in the job definition. Optionally, you may append a suffix from the date and time in order to keep old images available in the Spoon.net Hub.



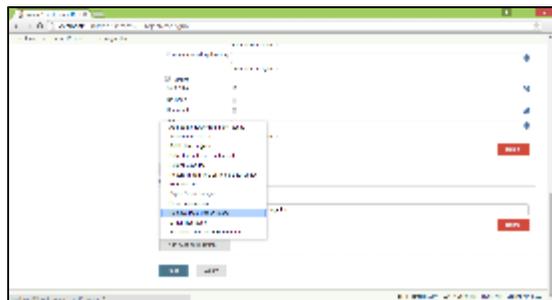
### Export an image to the local repository

Exporting images to the local repository is available as post-build action:



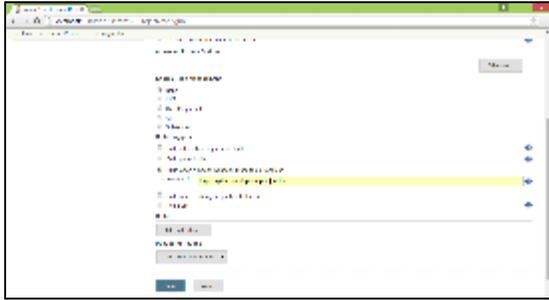
### Remove an image from the local repository

New images will remain in the local repository unless they are deleted using the `spoon rmi <image_name>` command. This step can be included in the job definition using the post-build action **Remove local Spoon image**.

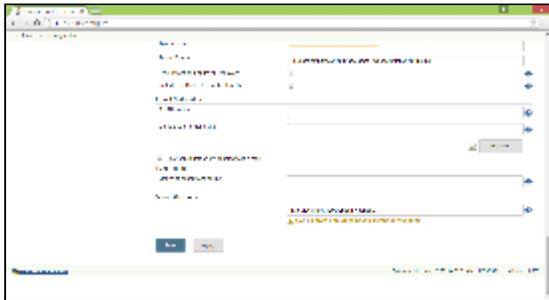


### Trigger a build job using webhooks

You may decide to build a Spoon container image whenever the source code changes for your project. To facilitate this, a Spoon Jenkins job can be triggered using an HTTP POST request. The request content is required to be compatible with the structure of [GitHub webhook events](#).



Webhook messages are sent to Jenkins\_URL/spoon-webhook endpoint as POST with Content-Type set to application/x-www-form-urlencoded.



Enjoy and please send any feedback or questions to [@spoonapps!](#)

## Change Log

Version 1.0 (Apr 21, 2014)

- Build an image with SpoonScript
- Push the resulting image to the Spoon.net Hub
- Export an image to the local repository
- Trigger a build job using webhooks
- Integration with Credentials plugin
- Integration with Git plugin