

OSF Builder Suite For Salesforce Commerce Cloud Deploy Plugin

OSF Builder Suite For Salesforce Commerce Cloud :: Deploy is an alternative to **Salesforce Commerce Cloud Build Suite** and is a very easy-to-use and highly configurable Jenkins plugin that is used to deploy your builds to your Salesforce Commerce Cloud continuous integration sandbox, development or staging instance. Being a native plugin for Jenkins and not a set of scripts it tightly integrates with it providing the best and easiest to use experience. You can make use of any existing Jenkins plugin (source code management, email notification, reports, etc.), configure it accordingly and use it.

OSF Builder Suite For Salesforce Commerce Cloud :: Deploy wants to be the first in a set of plugins that will make your life easier in regards to building your code and deploying it. We have a few ideas for what to build next, but we also take requests into account, so if you think of something you can let us know and we'll take it under consideration for a future plugin.

If you have a bug to report or maybe a feature that you wish to request, please do so [on GitHub](#), [on the project's issues page](#).

Features

- Simple. It does one thing, and it does it well.
- Easy to install, use and keep updated.
- Easy to configure. The plugin can be configured from the Jenkins web interface.
- Support for classical mode, Jenkins [Pipelines](#) and also the new modern [Blue Ocean](#) interface.
- Super flexible. Every little thing is configurable so that the plugin can be easily adjusted to your workflow.
- Integrated with the Jenkins [credentials plugin](#) so that your credentials are safely stored encrypted.
- Support for two factor authentication
- Build info cartridge. A `inf_build` cartridge will be created as part of the build process that can be used to display information about the build number, version, when the build was made and who made it, in the storefront. This information can be displayed in the page source, title or by using the [OSF Builder Suite For Salesforce Commerce Cloud :: Deploy](#) plugin for Chrome.
- Support for builds bigger than 100MB. The build is split into multiple parts that are deployed one by one so you don't hit the 100MB upload quota limit that Salesforce Commerce Cloud has.
- Good documentation. Every option is documented both here on this page but also inline in Jenkins's UI by clicking the question mark icon next to the item for which you wish to display the help information.
- Support for HTTP proxy with basic or [NTLM](#) authentication.
- Free
- Open source

Installation

Just go to "*Manage Jenkins > Manage Plugins > Available*", search for "*OSF Builder Suite*", select "*OSF Builder Suite For Salesforce Commerce Cloud :: Deploy*" and click "*Download now and install after restart*" button.

Configuration

Instance Hostname	<input type="text"/>
-------------------	----------------------

Hostname of the SFCC instance where this build should be deployed. Examples:

<code>cert.staging.<i>realm.customer</i>.demandware.net</code>	For deployments to a staging instance that has two factor auth enabled.
<code>staging-<i>realm-customer</i>.demandware.net</code>	For deployments to a staging instance that does not have two factor auth enabled.
<code>development-<i>realm-customer</i>.demandware.net</code>	For deployments to a development instance.

devNN-*realm-customer*.demandware.net

For deployments to a sandbox instance.

Business Manager Credentials

- none - ▾



Business Manager credentials of type "*OSF Builder Suite :: Business Manager Credentials*" for the SFCC instance where this build should be deployed.

Two Factor Auth Credentials

- none - ▾



Two Factor Auth credentials of type "*OSF Builder Suite :: Two Factor Auth Credentials*" for the SFCC instance where this build should be deployed. Select - *none* - if you deploy to a instance that does not require two factor auth.

Open Commerce API Credentials

- none - ▾



Open Commerce API credentials of type "*OSF Builder Suite :: Open Commerce API Credentials*" for the SFCC instance where this build should be deployed.

Open Commerce API Version

The version to be used by the calls made to OCAPI. The Open Commerce API Version starts with the character "v" (lowercase) followed by the actual version number, separated by an underscore.

For example: "v18_3"

Build Version

Name of the code version that is being deployed. This will be added as a suffix to the one generated by the builder.

Example: If you set it to dev then the code version will be *bn_YYYYMMDD_dev* with N the number of the current build, *YYYY* four digits year, *MM* two digits month, *DD* two digits day and dev at the end.

Create Build Info Cartridge

When this option is checked, a new cartridge named inf_build will be created. This cartridge can be used to display information about the build number, version, when the build was made and who made it, in the storefront.

This information can be displayed in the page source, title or by using the [OSF Builder Suite For Salesforce Commerce Cloud :: Deploy](#) plugin for Chrome.

Activate Build

When this option is checked, the build will also be activated after it was deployed. If left unchecked the build will only be deployed to the target instance without being activated.

Sources

X

Source Path

Exclude Patterns

List of sources where the builder will look for cartridges in the form of a path (relative to the workspace) to a directory where the builder will look for cartridges.

Example: *scm/my-git-repo/cartridges*

You can also define a list of patterns to be ignored. If a path matches any of the patterns in this list then it will be ignored and not added to the build. The pattern needs to be relative to the source path defined above.

When a path is matched against a pattern, the following special characters can be used:

?	Matches one character (any character except path separators)
*	Matches zero or more characters (not including path separators)
**	Matches zero or more path segments

Examples:

**/*.js	Matches all .js files/dirs in a directory tree
node_modules/**	Matches the node_modules folder and all its contents
test/a?? .js	Matches all files/dirs which start with an a, then two more characters and then .js, in a directory called test
**	Matches everything in a directory tree
/test//XYZ*	Matches all files/dirs which start with XYZ and where there is a parent directory called test (e.g. abc\test\defghi\XYZ123)

Temp Build Directory

Path (relative to the workspace) to a temp directory, that will be used during the build. If the directory does not exist, it will be created by the builder and it will also be automatically cleaned up before each build.

Example: *tmp/build*

Open Commerce API Settings

Go to "Administration > Site Development > Open Commerce API Settings", select type "Data", select context "Global" and add following configuration:

```
1  {
2  |   "_v": "18.3",
3  |   "clients": [
4  |     {
5  |       |   "client_id": "YOUR_CLIENT_ID",
6  |       |   "resources": [
7  |         |   {
8  |         |     |   "resource_id": "/code_versions/*",
9  |         |     |   "methods": ["put", "patch"],
10 |         |     |   "read_attributes": "(**)",
11 |         |     |   "write_attributes": "(**)"
12 |         |   }
13 |       | ]
14 |     }
15 |   ]
16 | }
17
```

HTTP Proxy Host

If your Jenkins server sits behind a firewall and does not have direct access to the internet, you can specify the HTTP proxy host in this field to allow Jenkins to connect to the internet through it.

HTTP Proxy Port

This field works in conjunction with the proxy host field to specify the HTTP proxy port.

HTTP Proxy Username

This field works in conjunction with the proxy host field to specify the username used to authenticate with the proxy.

If this proxy requires Microsoft's [NTLM](#) authentication scheme then the domain name can be encoded within the username by prefixing the domain name followed by a back-slash \ before the username, e.g "ACME\John Doe".

HTTP Proxy Password

This field works in conjunction with the proxy host field to specify the HTTP proxy password.

Disable SSL validation

When this option is checked, the builder will no longer validate the SSL certificate and hostname of the target instance.

This has potential security implications so make sure you know what you are doing before enabling this option!

Jenkins Pipeline Configuration

Here's a sample pipeline configuration to get you started:

```

1  node {
2      stage("Cleanup") {
3          // do stuff...
4      }
5
6      stage("Git") {
7          dir("scm/my-git-repo") {
8              git(
9                  url: "git@server.tld:account/my-git-repo.git",
10                 branch: "develop",
11                 credentialsId: "0b45d834-c982-4026-ad6b-6117276d4b03"
12             )
13         }
14     }
15
16     stage("Build") {
17         nodejs("NodeJS (v6.11.2)") {
18             dir("scm/my-git-repo") {
19                 // do stuff...
20             }
21         }
22     }
23
24     stage("Deploy") {
25         osfBuilderSuiteForSFCCDeploy(
26             hostname: "cert.staging.realm.customer.demandware.net",
27             bmCredentialsId: "e4b5af2c-f1ce-4b82-ae59-43bbba37a889",
28             tfCredentialsId: "cf9e0a60-ed85-41cb-b5ff-e2be51fcc8ab",
29             buildVersion: "dev",
30             createBuildInfoCartridge: true,
31             sourcePaths: [
32                 [sourcePath: "scm/my-git-repo"]
33             ],
34             activateBuild: true,
35             tempDirectory: 'tmp/build'
36         )
37     }
38 }

```

You can also always consult the pipelines documentation available at <https://jenkins.io/doc/book/pipeline/> or check the pipeline syntax link right inside Jenkins on the left navigation menu.

 Back to Dashboard

 **Status**

 Changes

 Build Now

 Delete Pipeline

 Configure

 Open Blue Ocean

 Full Stage View

 Pipeline Syntax 

 Git Polling Log

Version history

<https://github.com/jenkinsci/osf-builder-suite-for-sfcc-deploy-plugin/releases>