**JENKINS PLUGINS**

* Jenkins uses plugins to provide much of its functionality
* Many "classical" Jenkins functions are implemented as plugins
* More than 1700 plugins are available
* This "modular" architecture means that your Jenkins installation includes the features and functionality you need without being bloated with functionality you do not need
* Managing plugins is a critical part of managing Jenkins
* Many features are implemented with multiple plugins
* Many plugins have dependencies on other plugins

**FEATURES IMPLEMENTED WITH PLUGINS**

* Source code management tools
* Build Tools
* Reporting tools
* Code coverage, static code analysis
* Online source code browsers
* Issue tracker
* Notification tools
* Views and UI customizations
* Distributed builds

## WHAT ARE PLUGINS ?

* A plugin is a JAR file with some special conventions (e.g. no web.xml)
* Contained in a file with an hpi or jpi extension
* Stored in the ${JENKINS\_HOME}/plugins directory
* unless you use --pluginroot to change where the binary of the plugins are "exploded"
* May have dependencies (mandatory or optional)
* Plugins are versioned artifacts that can be upgraded
* If necessary, they can be downgraded but this is not generally recommended

## UNINSTALLING A PLUGIN

* Removes the plugin binary (jpi or hpi file) from the disk
* Plugin continues to function until you restart Jenkins
* After a restart, plugin does not appear in the UI and all extensions it contributed disappear
* Does not remove the configuration that the plugin may have created
* If the plugin contributed extensions to any jobs/agents/views/builds/etc, Jenkins reports unrecognized fragments in configuration files then ignores such fragments
* Until you remove those configuration files, you can re-install the plugin and restart Jenkins to restore the configurations
* Use the "Manage Old Data" screen to scrap configurations left by an uninstalled plugin

## DISABLING A PLUGIN

* Disabling a plugin is a softer way to retire a plugin
* The extension for the plugin binary is changed to jpi.disabled or hpi.disabled
* Jenkins does not start the plugin
* Extensions contributed by this plugin are not visible
* The plugin still appears in the "Installed" list
* The box under the **Enabled** column is not checked
* Fragments contributed to configuration files by a disabled plugin are handled like those for uninstalled plugins
* Reenable the plugin by checking the box in the **Enabled** column
* You may need to restart the instance to make the plugin operational

**MANAGE OLD DATA**

* Use the "Manage Old Data" page to clean up configurations left by uninstalled and disabled plugins:

**UPDATE PLUGINS REGULARLY**

* Many plugins are updated frequently
* Enhanced functionality, bug fixes, security fixes
* These should be applied as soon as is feasible
* Always read the changelog information about new plugins before installing
* Use a staging environment to validate updated plugins before they are applied to the production environment
* Take a backup of your instance before updating a plugin
* Many updates take effect only after Jenkins is restarted

## JENKINS UPDATE CENTER

* The full set of open source plugins are available at the [Plugins Center](https://plugins.jenkins.io/)
* Also called the "Update Center"
* When you download a plugin:
* Jenkins downloads the plugin index from the configured Update Center
* Jenkins filters the index by the core version
* If you install the plugin, the proper binary is downloaded

**INSTALL NEW PLUGIN**

* To illustrate how to install a new plugin, we are going to install the beer plugin
* This is a harmless plugin that adds a page on the Jenkins root to print some jokes about beer
* It has no dependencies and is lightweight.
* You MUST have Internet access (HTTP proxy supported) to download a new plugin from this page

**CONFIGURE HTTP PROXY ACCESS**

* If you are running under an HTTP proxy, select the **Advanced** tab to configure HTTP proxy access:
* Click the **Advanced** button on the bottom right to enable the **Validate Proxy** configuration fields before clicking "Submit" to submit the configuration

**Manual Plugin Installation**

Method 1:

* Download hpi file
* Manage Plugins >> Advanced >> upload hpi file >> restart Jenkins

Method 2:

* Assuming a .hpi file has been explicitly downloaded by a systems administrator, the administrator can manually place the .hpi file in a specific location on the file system.
* Copy the downloaded .hpi` file into the JENKINS\_HOME/plugins directory on the Jenkins master (for example, on Debian systems JENKINS\_HOME is generally /var/lib/jenkins).
* The master will need to be restarted before the plugin is loaded and made available in the Jenkins environment.

But *outside* the **Update Center** method, the dependency management is **YOUR** responsibility. You have to manage your upgrade policy carefully, in order to have **stability** across versions!