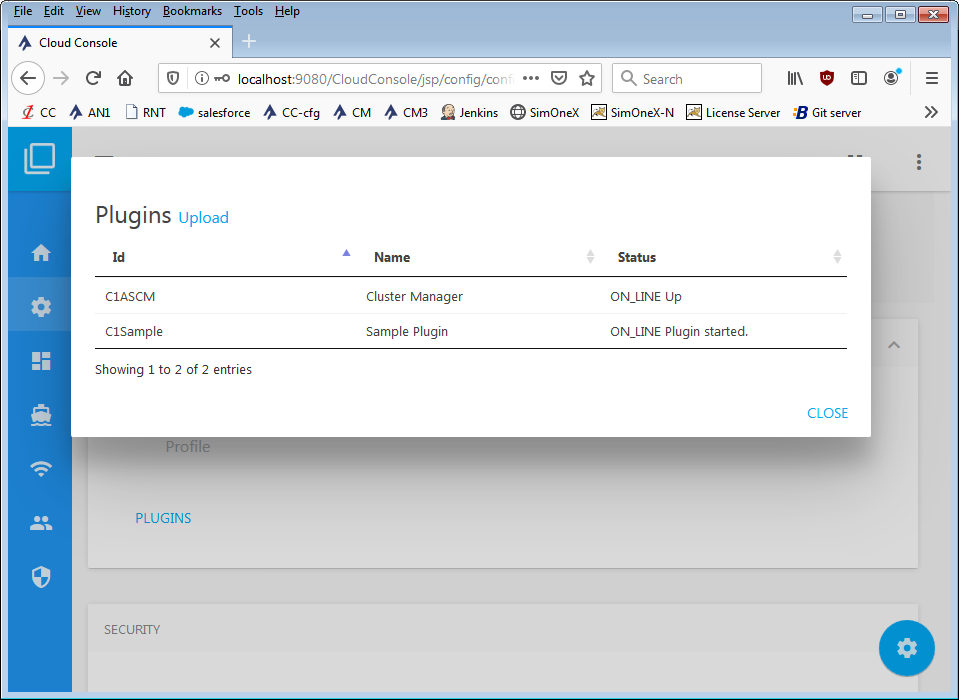
Cloud Console Plugin System

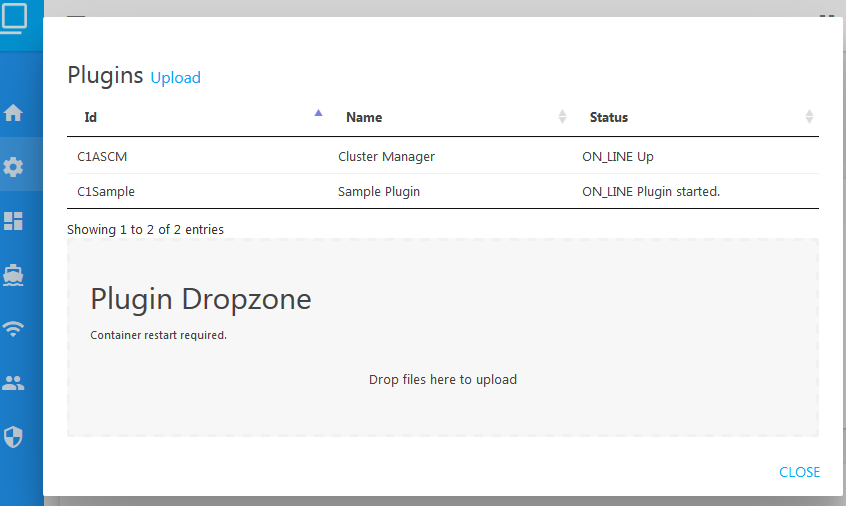
The cloud console features a plugin system that can be used to create

* File or configuration editors
* Dashboards for metrics
* Other useful features or service

To access the plugin system click Configure > Node > Plugins

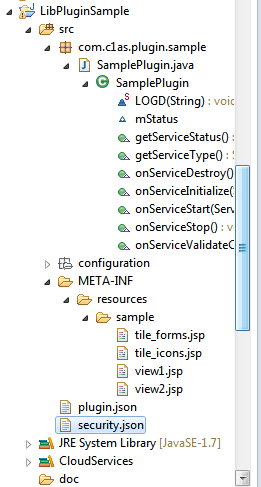


Plugins can be manually uploaded by drag and drop the plugin JAR via the upload link **(Note: Container restart is required to load the plugin code)**.



# Plugin Architecture

A plugin is a regular Java project that has a **plugin.json** and an optional **security.json** descriptors plus JSP views, java code or controller servlets as any common web app.



## Plugin Descriptor

It describes items such as load class, menu items, etc. See the following example.

{

"service.class" : "com.c1as.plugin.sample.SamplePlugin",

"service.config" : "sample.xml",

"service.vendor" : "Sample Plugin",

"service.vendorId" : "C1Sample",

"ui.sideBarMenus": [

{

"id": "00\_sample\_menu",

"label" : "Sample Menu",

"parent" : "/",

"url" : "",

"iconCss" : "people"

},

{

"id": "01\_sample\_submenu1",

"label" : "Sample SubMenu1",

"parent" : "Sample Menu",

"url" : "sample/view1.jsp",

"iconCss" : ""

},

{

"id": "02\_sample\_submenu2",

"label" : "Sample SubMenu1",

"parent" : "Sample Menu",

"url" : "sample/view2.jsp",

"iconCss" : ""

}

]

}

|  |  |
| --- | --- |
| Key | Description |
| service.class | Java class that controls the Plugin lifecycle. **It must implement the interface (IServiceLifeCycle)** |
| service.config | Plugin configuration XML. To be displayed in the Configure menu of the console. |
| service.vendor | Plugin Name |
| service.vendorId | **Unique Id of the plugin.** |
| ui.sideBarMenus | Array of left nav-bar menus:  {  "id": "00\_sample\_menu",  "label" : "Sample Menu",  "parent" : "/",  "url" : "",  "iconCss" : "people"  } |

## Navigation Bar Menus

|  |  |
| --- | --- |
| Key | Description |
| Id | Id of the menu. **It is used to sort all menus alphabetically.** |
| Label | Menu label |
| Parent | Name of the menu parent or / if located at the root level. |
| url | Url of the menu view (JSP). |
| iconCSS | Icon style: css name or id. |

## Optional OWASP Modular Security

If your plugin config has external URLs (any other type of external URL/access), the console will reject the request with **Security Access Violation.** To allow validations in the HTTP request params use a security descriptor as described in the Cloud Services document. For example, the following JSON validates the parameter foo to allow URL from bar.com.

{

"validators": [

{ "param": "SAMPLE\_HTTP\_PARAM", "regexp": "((?:https?\\:\\/\\/|(foo|bar).comm))"}

]

}

## Plugin Views

A JSP template is provided in the sample plugin that includes the default styles used by the cloud console. The basic skeleton is simple:

<%@page import=*"com.c1as.cloud.console.ThemeManager"*%>

<%@page import=*"com.c1as.cloud.console.SkinTools"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<%

**final** String contextPath = getServletContext().getContextPath();

String theme = (String)session.getAttribute("theme"); // console theme (should not be null)

//String title = (String)session.getAttribute("title"); // Top Left title (should not be null)

**if** ( theme == **null**) theme = ThemeManager.DEFAULT\_THEME;

String statusMessage = "NULL";

String statusType = **null**;

**final** String basePath = "../";

%>

<!DOCTYPE html>

<html>

<head>

<jsp:include page=*"*<%=SkinTools.buildHeadTilePath(basePath) %>*"*>

<jsp:param value=*"*<%=SkinTools.buildBasePath(basePath) %>*"* name=*"basePath"*/>

<jsp:param value=*"*<%=basePath%>*"* name=*"commonPath"*/>

<jsp:param value=*"*<%=theme%>*"* name=*"theme"*/>

<jsp:param value=*"View Title"* name=*"title"*/>

</jsp:include>

</head>

<body>

<jsp:include page=*"*<%=SkinTools.buildPageStartTilePath(basePath) %>*"*>

<jsp:param value=*"*<%=SkinTools.buildBasePath(basePath) %>*"* name=*"basePath"*/>

<jsp:param value=*"*<%=basePath%>*"* name=*"commonPath"*/>

<jsp:param value=*"Sample Heading"* name=*"pageTitle"*/>

<jsp:param value=*"Home,Pages,Sample View"* name=*"crumbLabels"*/>

<jsp:param value=*"../index.jsp,#,class\_active"* name=*"crumbLinks"*/>

</jsp:include>

<!-- STATUS MESSAGE -->

<jsp:include page=*"*<%=SkinTools.buildStatusMessagePath(basePath)%>*"*>

<jsp:param value=*"*<%=statusMessage%>*"* name=*"statusMsg"*/>

<jsp:param value=*"*<%=statusType%>*"* name=*"statusType"*/>

</jsp:include>

<!-- CONTENT START -->

<jsp:include page=*"tile\_forms.jsp"* />

<!-- CONTENT END -->

<jsp:include page=*"*<%=SkinTools.buildPageEndTilePath(basePath) %>*"*>

<jsp:param value=*"*<%=SkinTools.buildBasePath(basePath) %>*"* name=*"basePath"*/>

<jsp:param value=*"*<%=basePath%>*"* name=*"commonPath"*/>

</jsp:include>

<script type=*"text/javascript"*>

$(document).ready(**function**() {

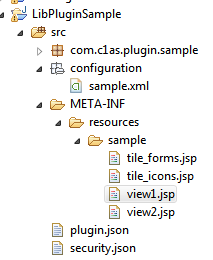
});

</script>

</body>

</html>

**Note: all views must be stored in META-INF/resources**.



## Servlets, Code and External Dependencies (JARS)

These are handled like any other Java project: put your code in the **src** folder; put all your external libraries (JAR) in the root **lib** folder. **Note: When uploading via drag-drop in the console upload all required JARS.**

# Sample Plugin

**A sample plugin (Java project) is available in the Cloud Services repo** as well as this document. Use it to create your own custom editor/service. Here is a look at the sample plugin once deployed in the console.

