Hewlett Packard Enterprise

Codar

Software version: 1.70

For Microsoft Windows® and Linux operating systems

Codar Plugin Automation for Continuous Integration Tool

Document release date: November 2016

Software release date: July 2016

Contents

Introduction

Codar is a continuous delivery automation tool which supports deployment steps and pipeline process automation. It is the best choice for the customer to integrate the Continuous Integration (CI) tool with Continuous Delivery (CD) tool like Codar to automate the process of CI-CD.

Any code which is built by the build tool should be first deployed and verified on the dedicated environment before being consumed by other lifecycle stages. This will help the team to regress the code further and finally roll out to production.

This white paper explains how to use the APIs which are required to connect or integrate CI with Codar tool.

Also the steps to automate the integration process are documented in this whitepaper. For more details on the product features, refer the product guides.

Types of designs

Codar supports two types of designs, namely Topology and Sequenced. This section will explain the list of APIs that are to be used to automate the integration of CI tool with Codar's topology design and Codar's sequenced design.

Codar Topology Design

Following are the details and actions required to automate CI tool with Codar,

- Application design ID
- Component ID
- Application JSON to fetch the component property names which are modifiable by CI tools
- Create package API or flow
- Promote package API or flow with continuous promote yes or no option

There are multiple ways to get the above information. One of the easiest ways is to export the topology design as JSON and then fetch the ID from it. Another option is to access swagger API portal to get the list of designs which will have all the design ID with JSON body. In this topology example, we are going to get the information from topology JSON. For sequence design, we will see how to fetch the same information through an API in the section 'Codar Sequenced Design'

Steps to fetch the information required to integrate topology application design with CI tool:

1. Log in to Codar
2. Go to Designs→ Topology → Designer → Search and type the application name you want to deploy through CI-CD integration.
3. Click the specific version of the application design.
4. Click the gear icon on the right  and select Export.

   This will download a generated JSON file.
5. Open the downloaded JSON file which is in the format as shown in the image below.

   The highlighted string is the ID of this design.

```
{
  "@self" : "/csa/api/topology-model/topology/b81bab25-c2ef-4fc3-88f3-cb8cb6e916d1",
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "c805f5de5e5447d9b1a262a8f445dd29",
  "version" : "5.0.0",
  "displayName" : "PetClinic Application -  Partial Design - Release",
  "description" : "A two-tier PetClinic Application with Database component installed on MySQL
```


6. In order to get the component ID which contains the parameters which are modifiable during deploy or re-deploy , scroll down to look for the component names which will give the detailed information about the component and ID.

   In the sample JSON, the component name “PetClinic Application 4 Partial Design’ has the modifiable property which is “artifacturl”. Search for this component name and fetch the ID highlighted as given in the image below:


```
  requirements" : [ ]
}, {
  "id" : "9b4da342-8b49-8c5d-1db4-8062778c9e3b",
  "name" : "PetClinic Application 4 Partial Design",
  "component" : {
    "@self" : "/csa/api/topology-model/component-type/a22516ee-ff82-4686-ad10-33056421f08f",
    "groupId" : "com.hp.csa.type.HPOO",
    "artifactId" : "PetClinicApplication4PartialDesign_bfccdc9dfdf14de788131fd7fdb80f0f",
    "version" : "1.50.0000"
  },
```


   The “property” which is modifiable during deploy or re-deploy is present as part of this component. The value of this property is set by the Jenkins variables and by the build admin.


```
    }
  }, {
    "propertyKey" : "artifacturl",
    "propertyValue" : {
```


7. By following the above steps you should have the application design ID, application component ID, and the property which are required to be modified from the CI tool.

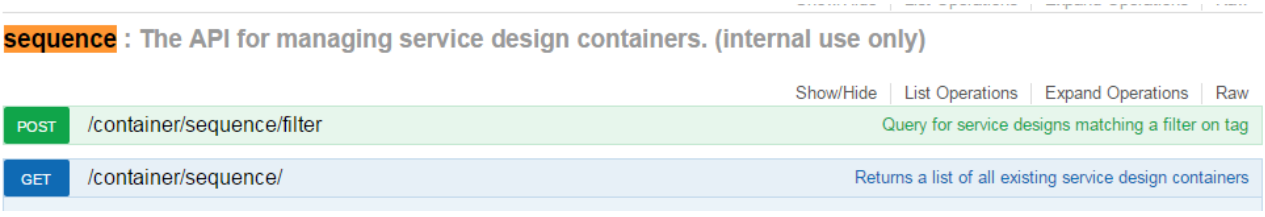
Codar Sequenced Design

As mentioned earlier, through APIs, we will fetch the information required to automate the integration between CI tools with Codar’s sequenced design.

Steps to fetch the information which required to integrate sequenced design application with CI tool:

1. Log in to Codar
2. Attach the following URI with the logged-in URL ’apidocs.jsp‘ so the URL is as below:

   https://Codarmc:8444/csa/apidocs.jsp#!/



3. Click on the ’Try it out‘ button in the subsection. This will list all the sequenced designs along with the JSON body.

GET /container/sequence/ Returns a list of all existing service design containers

Implementation Notes

The API response returns the collection of containers for service designs existing in the system.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
start-index		Specifies the offset of the first entry to be included in the page.	query	integer
page-size		Specifies the page size.	query	integer
sort		Name of field to be used in ordering optionally followed by colon and 'ascending' or 'descending'	query	string
after		Filter members to those modified at or after this timestamp. Uses SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'") in UTC	query	string
before		Filter members to those modified before this timestamp. Uses SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'") in UTC	query	string

Error Status Codes

HTTP Status Code	Reason
404	No containers found.
403	Authorization failure

Try it out!

4. The below JSON body will contain the information about the application container ID as well as the versions under this container. The members section will give the version ID of the sequenced design. In the image below, the IDs inside the box are the application version design IDs.

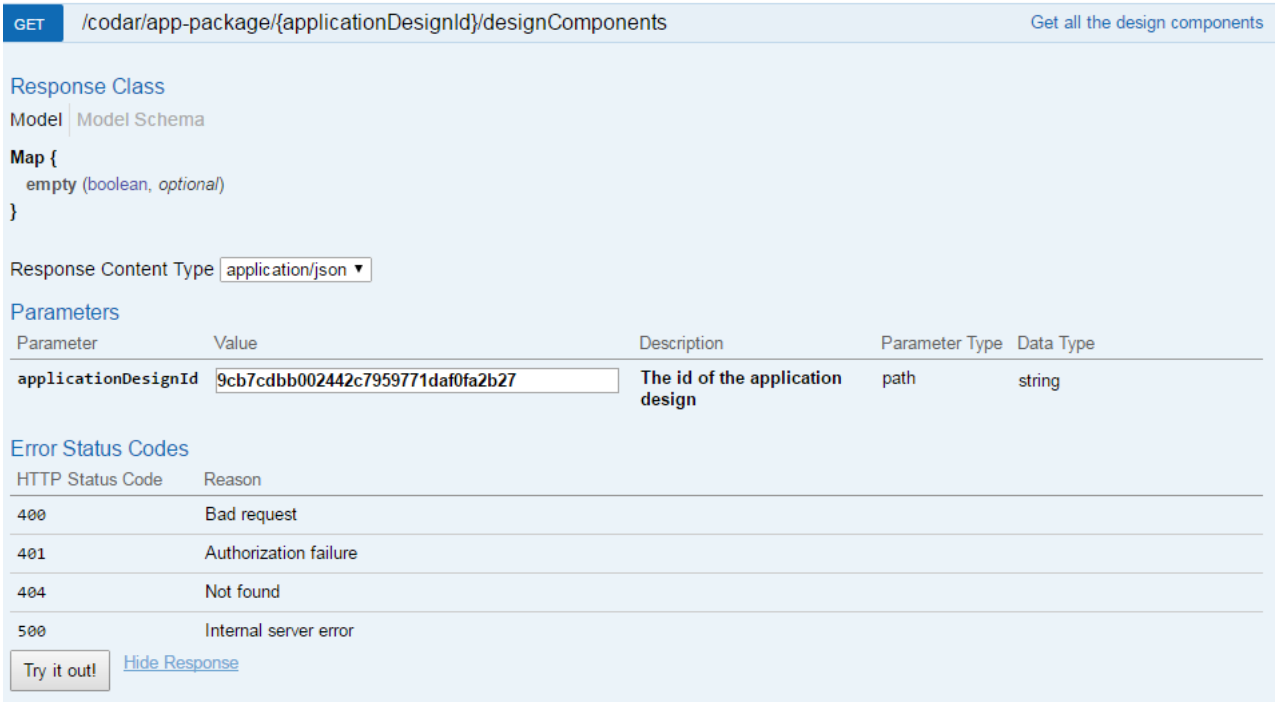

```
{
  "@self": "/csa/api/container/sequence/f5310bb1135845c7b0374f85980e62a4",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "@created": "2016-09-23T13:41:37.397Z",
  "@modified": "2016-09-23T13:52:36.581Z",
  "global_id": "f5310bb1135845c7b0374f85980e62a4",
  "name": "vCenter Compute with Basic Options",
```

```
 "members": [
   {
     "@self": "/csa/api/service/design/9cb7cdbb002442c7959771daf0fa2b27",
     "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
     "@created": "2016-09-23T13:41:38.198Z",
     "published": false,
     "version": "16.07b",
     "upgrade_available": false,
     "designId": "9cb7cdbb002442c7959771daf0fa2b27"
   },
   {
     "@self": "/csa/api/service/design/3afb4361e79f4caca092e90aa1c18cfa",
     "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
     "@created": "2016-09-23T13:52:36.581Z",
     "published": false,
     "version": "16.07",
     "upgrade_available": false,
     "designId": "3afb4361e79f4caca092e90aa1c18cfa"
   }
```


5. Now, the next step is to fetch the component IDs present inside the application version JSON body. This information also can be fetched from the swagger API available as part of Codar.

6. Go to the section from 'https://Codarmc:8444/csa/apidocs.jsp#!/'

7. Open the API "GET /codar/app-package/{applicationDesignId}/designComponents" and provide the application version design ID to fetch the component information by clicking "Try it out" button.



8. The component ID can be fetched from the response body (JSON output) which is displayed by the above API after clicking 'Try it out!'.

   The highlighted ID is the component ID which should be used to pass any input to the property which is present as part of the component. The 'displayName' is the property name which is highlighted in the second box in the image below:

Note: In sequenced-based designs, Codar does support all properties as modifiable but re-deploy is not supported.

Integration Automation

Here is the master piece which is going to integrate CI tool with Codar. Codar has an HPE Operation Orchestration (HPE OO) flow which can automatically create a package and do a continuous promote release pipeline or mere deployment on first lifecycle stage.

This OO flow name and the path is 'Library/Integrations/Hewlett-Packard/Cloud Service Automation/Components/CODAR/Devops/Continuous Deployment Flow.xml'.

The OO flow can be triggered remotely from a command line tool called "RSFlowInvoke.exe" with the required options.

On **Linux** environment, **JRSFlowInvoke.jar** can be used to trigger the workflow with the same option.

RSFlowInvoke.exe and JRSFlowInvoke.jar is supported only for HPE OO Central 9.x version but this tools still works with 10.x version.

HPE OO 10.x comes with the tool called "OOSH.bat/OOSH.sh" to accomplish the same task of invoking or triggering the OO flows remotely from a CI tool. This tool may be further enhanced in the next version which can be invoked as standalone tool.

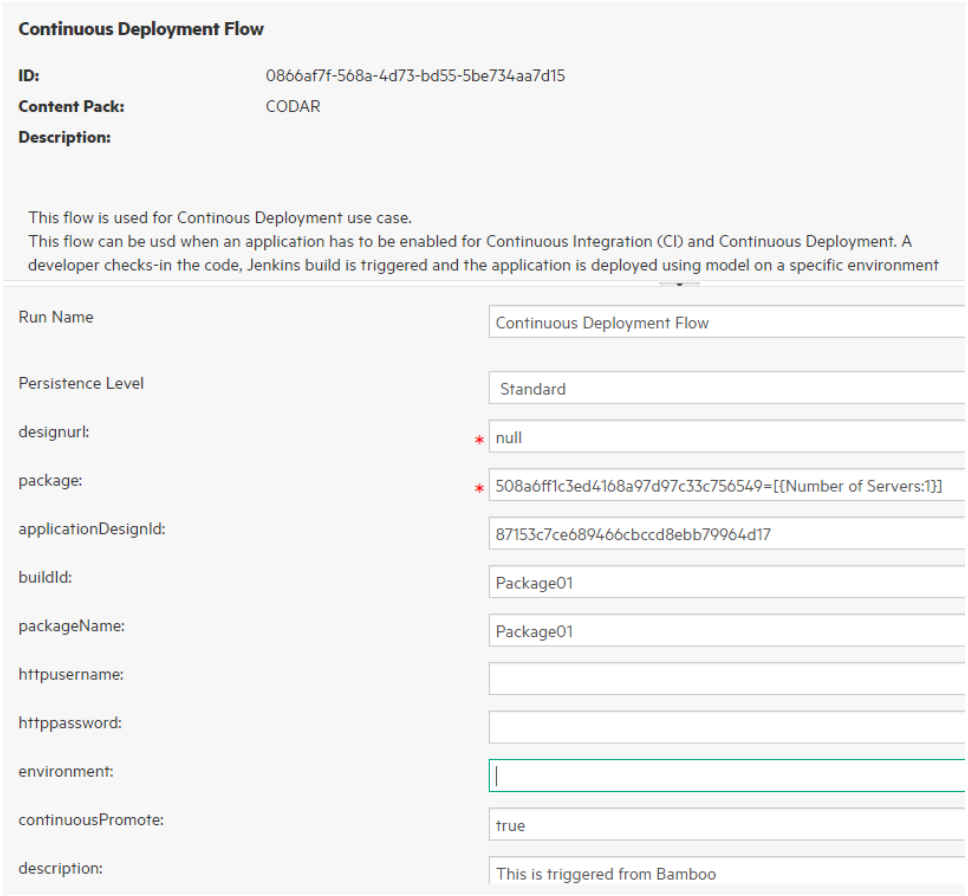
This flow will take the following input in order to remotely trigger an OO flow:

RSFlowInvoke.exe -host <CODARHOST>:<CODARPORT> -flow 0866af7f-568a-4d73-bd55-5be734aa7d15 -a Basic -u <OO Central Server Username> -p <OO Central Server Password> -t <Flow Timeout> -inputs "<Input to the application flow>"

Sample Command

RSFlowInvoke.exe -host 16.103.31.119:8445 -flow 0866af7f-568a-4d73-bd55-5be734aa7d15 -a Basic -u admin -p cloud -t 600 -inputs "codarusername=codaruser&codarpassword=admin@123&designurl=null&package=508a6ff1c3ed4168a97d97c33c756549=[{Number of Servers:1}]&applicationDesignId=87153c7ce689466cbccd8ebb79964d17&buildId=Package1&packageName=Package1&continuousPromote=true&description=This is triggered from bamboo"

In the above command, the option input has the following values:



The password can be encrypted using the same tool and passed with an option “-ep”

Refer to OO SDK guide for more details about this tool (https://softwaresupport.hpe.com/group/softwaresupport/search-result/-/facetsearch/document/KM00214112).

This flow can be invoked from any CI tool with any of the native plugins.

```
java -jar JRSFlowInvoke.jar  "https://172.16.165.21:8445/PAS/services/rest/run/0866af7f-568a-4d73-bd55-5be734aa7d15?codarusername=codaruser&codarpassword=codarpassword&designurl=null&package=508a6ff1c3ed4168a97d97c33c756549=[{Number of Servers:1}]&applicationDesignId=87153c7ce689466cbccd8ebb79964d17&buildId=Package1&packageName=Package1&continuousPromote=true&description=This is triggered from bamboo"  -u admin -p cloud
```

As mentioned above, there are many ways to invoke an OO flow remotely, for example through “wget”. For more information, refer to OO Software Development Kit (https://softwaresupport.hpe.com/group/softwaresupport/search-result/-/facetsearch/document/KM00214112)

Sample Implementation

In this section, we will see how a Codar plugin can be created for a CI tool using the native features of the CI tool.

The sample tool taken is Bamboo. Note that Codar already has a Bamboo plugin to trigger a topology based design with the Continuous Promote ‘Yes or No’ option.

In this sample implementation, we will see:

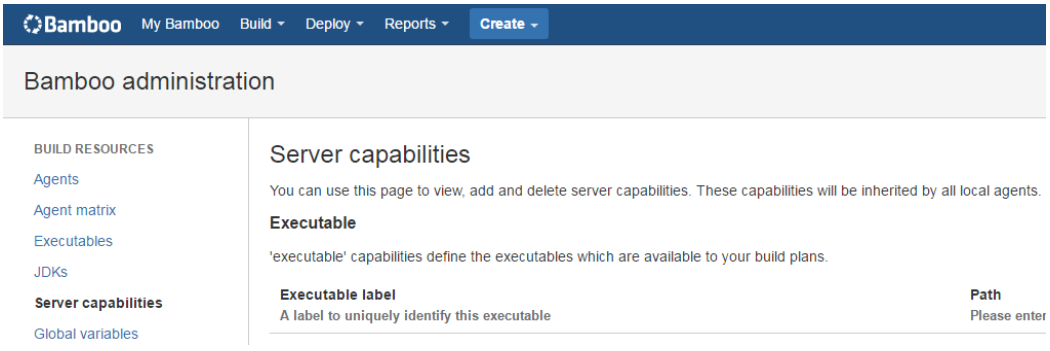
- How a sequenced design can be used in a pipeline to stand up an infrastructure
- How to install or configure the software
- How to deploy the application remotely
- How a pipeline can be kick started from Bamboo.

For details on the Bamboo CI tool, refer the Bamboo documentation.

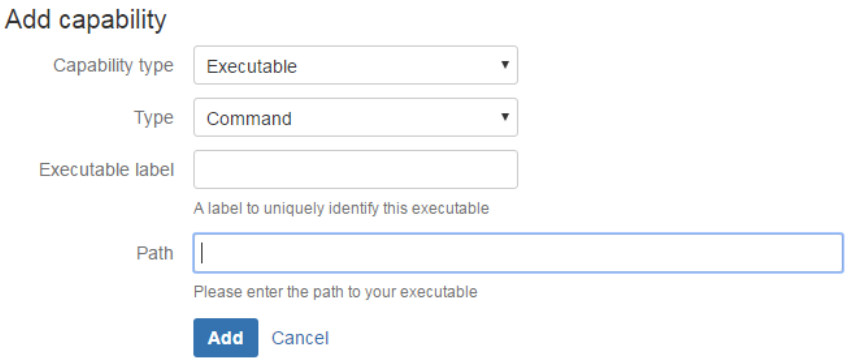
We will now see how this tool can be integrated with Bamboo and start an application release pipeline by passing appropriate inputs.

Steps to add an executable within Bamboo tool.

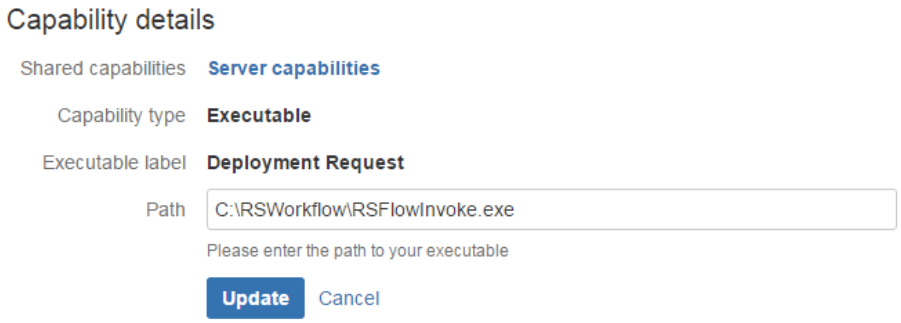
1. Create a new executable in the server capabilities as given in the image below.



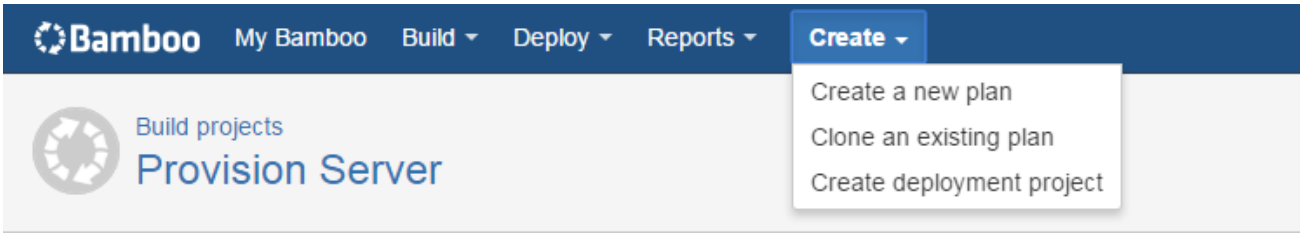
2. Create a new capability.



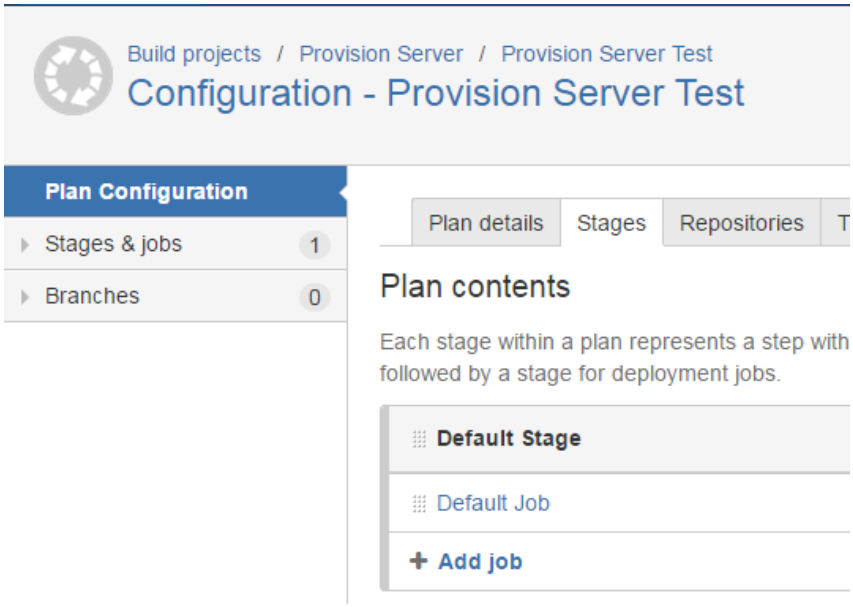
3. Copy the “RSFlowInvoke.exe” tool on the Bamboo system and place it under a directory. The “Path” should have the value as given below, which is the actual path of the RSFlowInvoke.exe present in the Bamboo system.



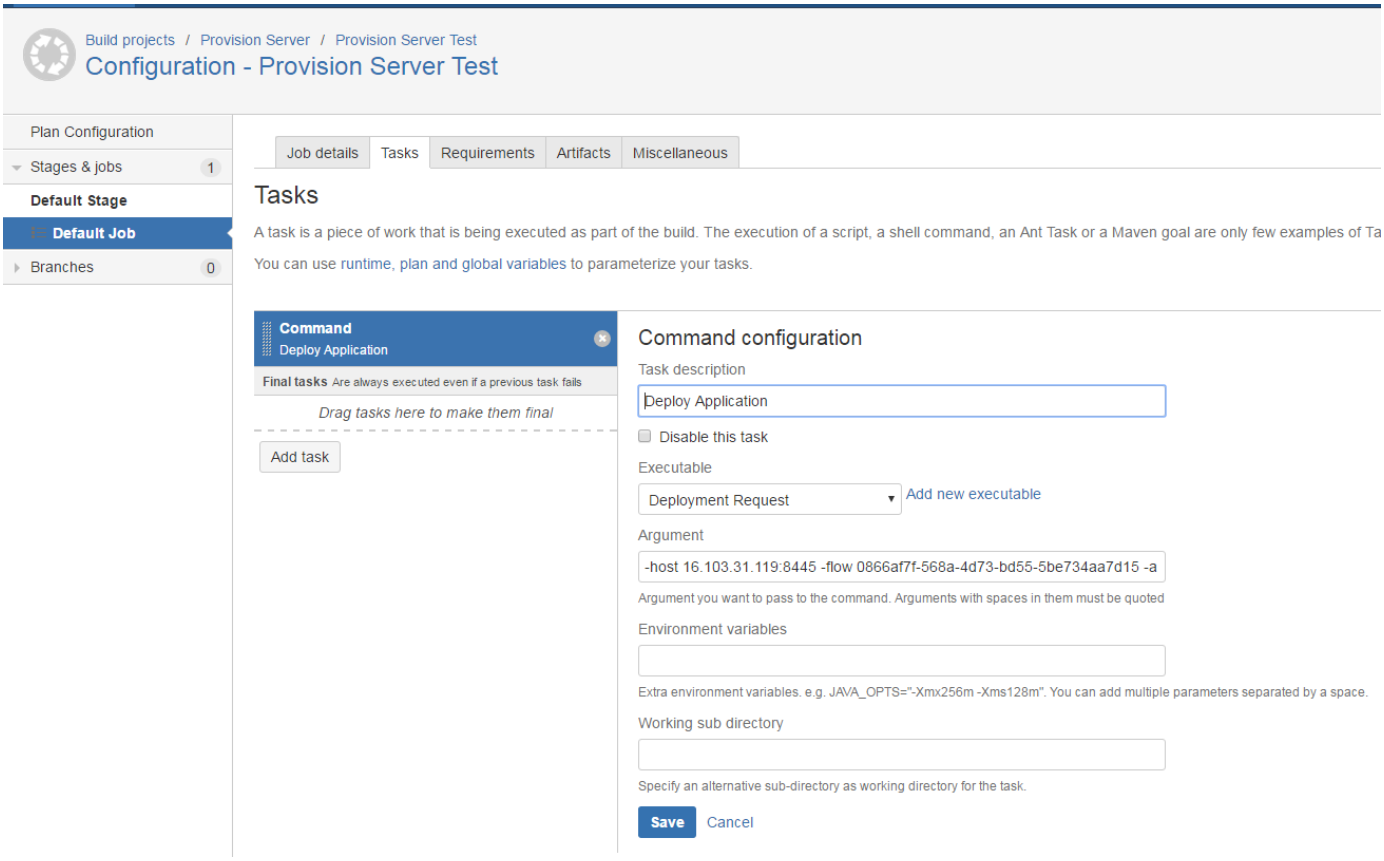
4. Create a “new build plan”



5. Click on the “default job”



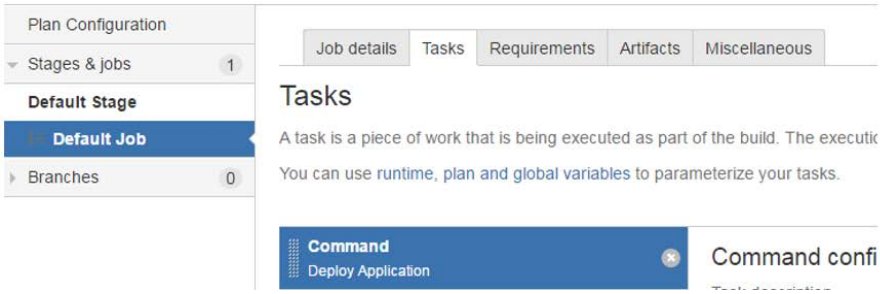
6. Add Task to the default job

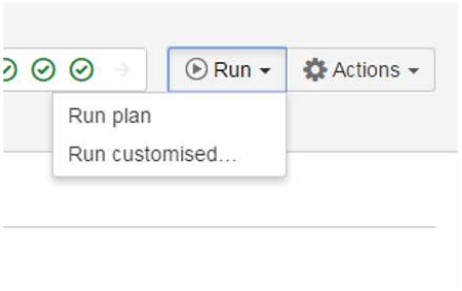


7. In the above image, the executable selected is already a configured executable for “RSFlowInvoke.exe”

   The arguments should be the RSFlowInvoke.exe arguments which is given below and explained in the ’Integration Automation‘ section.

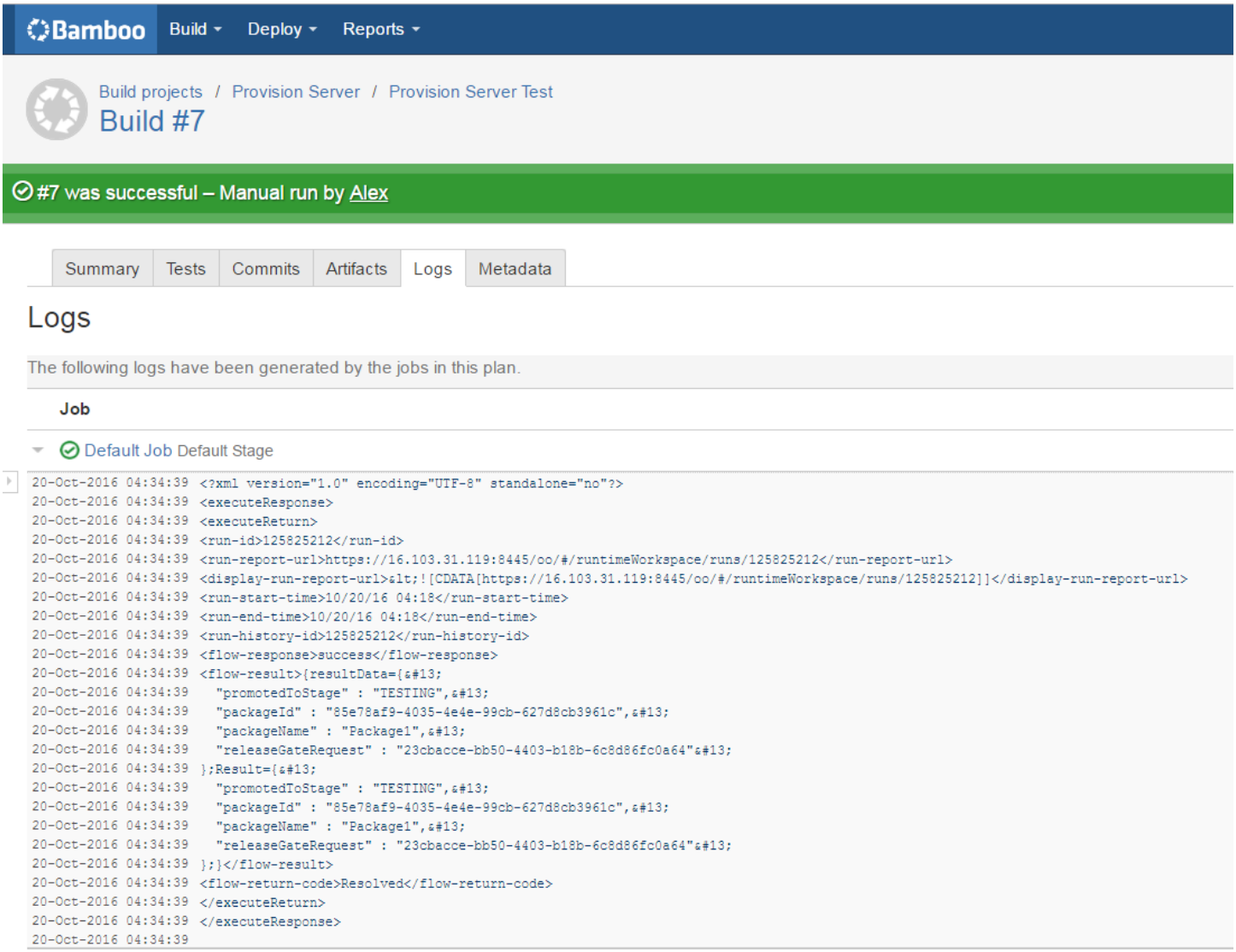
8. Once it is configured, trigger a test run to test the configuration by clicking the highlighted item in the box.





This is currently configured as part of the build task but this certainly can be configured as part deployment job as well

Below is the output of the run triggered from Bamboo plugin which invokes the OO flow to start the pipeline after creating the package in the design configured.



Conclusion

The instructions given in this whitepaper can be used to develop a quick and simple plugin for any CI tool to integrate with Codar in order to initiate the continuous pipeline after creating the package with required inputs. The APIs referenced in this whitepaper may get enhanced. Refer the Codar 'API Reference Guide' for details about the APIs.

Send documentation feedback

If you have comments about this document, you can send them to clouddocs@hpe.com.

Legal notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted rights legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright notice

Trademark notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

The OpenStack word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation in the United States and other countries, and are used with the OpenStack Foundation's permission.

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to the following URL and sign-in or register: https://softwaresupport.hpe.com.

Select Manuals from the Dashboard menu to view all available documentation. Use the search and filter functions to find documentation, whitepapers, and other information sources.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your Hewlett Packard Enterprise sales representative for details.

Support

Visit the Hewlett Packard Enterprise Software Support Online web site at https://softwaresupport.hpe.com.