IBM CP4I Jumpstart Kit
One-Click Deployment User’s Manual

07-JAN-2020

VERSION 0.1S

TATA

TATA CONSULTANCY SERVICES

DOCUMENT RELEASE NOTICE

Document Details:

Name	Version No.	Description
IBM CP4I Jumpstart Kit – One-click deployment User’s Manual	V0.1	

Revision Details

Action taken (Add / Del / Chg)	Preceding page no.	New page no.	Revision description

Review:

Reviewer	Role	Date

Target audience:

Group

Table of Contents

1. INTRODUCTION.. 4

1.1. PURPOSE OF DOCUMENT.. 4
1.2. ABOUT 'ONE-CLICK DEPLOYMENT TOOL' 4

2. SOLUTION ARCHITECTURE.. 5

1.1. SOLUTION ARCHITECTURE DIAGRAM.. 5
1.2. TOOLS USED 5
1.3. SOLUTION STEPS.. 5
1.3.1. Step 1: Checkout code from Source Code Manager... 5
1.3.2. Step 2: Build Code & Upload deployment package to Artifact Repository..................... 6
1.3.3. Step 3: Build Docker Image & Push to OpenShift Registry.. 6
1.3.4. Step 4: Deploy to OpenShift Cluster.. 6

3. USER'S MANUAL 7

3.1. CONFIGURE CICD TOOLS.. 7
3.2. EXECUTE.. 7
3.3. VALIDATE 8

1. INTRODUCTION

1.1. Purpose of Document

The One-Click Deployment Tool User’s Manual is targeted at IT teams involved in the development & deployment of integration services using IBM Cloudpak for Integration product.

The manual contains step by step instructions on how to configure & use the tool for automatic deployment of Integration message flows developed either on IIB or ACE to an IBM Cloudpak for Integration environment.

1.2. About ‘One-Click Deployment Tool’

‘One-Click Deployment Tool’ is a CICD solution to automatically deploy IBM IIB and ACE workloads to a containerized environment enabled by IBM Cloudpak for Integration or to an ACE server running on a virtual machine hosted on private or public cloud.

This tool will enable Integration team to deliver successful Integration projects in a fast & consistent manner.

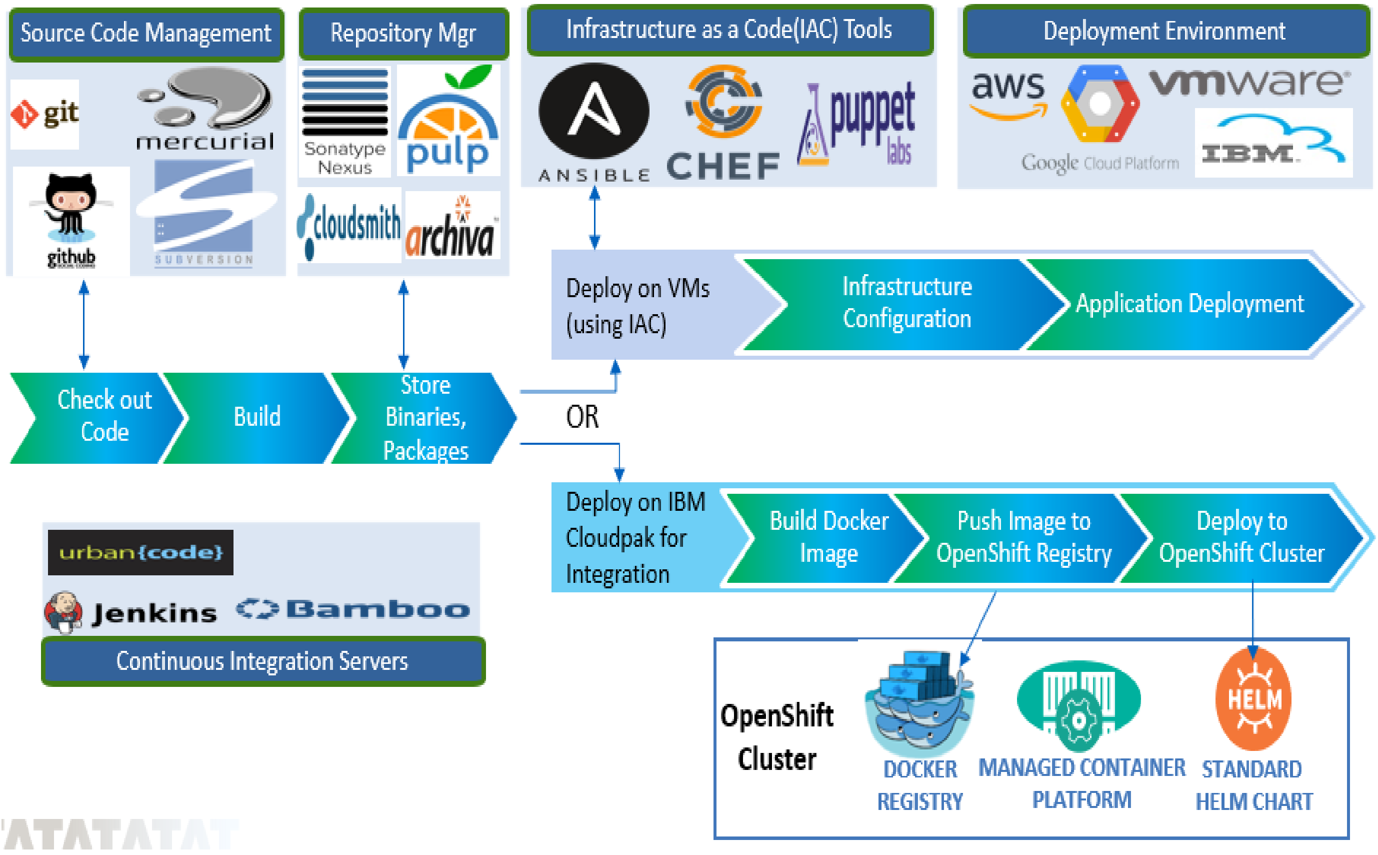
Features:

- Enables Agile delivery of Integration projects
- Provides end-to-end automation to IBM Cloudpak for Integration
- Easily customizable to leverage Client’s existing DevOps environment & tools
- Ability to selectively deploy Integration services as containers using Cloudpak for Integration or on VMs

2. SOLUTION ARCHITECTURE

This section describes the components and steps involved in the solution.

1.1.Solution Architecture Diagram



1.2.Tools Used

#	Tools Used		Description
1	Continuous Integration Server	Jenkins	
2	Source Code Management	GitHub	
3	Artifact Repository	Sonatype Nexus	
4	Kubernetes Cluster provider	OpenShift	
5	Infrastructure as a Code Tool	Ansible	
6	Cloud Deployment Environment	IBM Cloud	

1.3.Solution Steps

This section describes the steps followed in the end-to-end solution.

1.3.1.Step 1: Checkout code from Source Code Manager

This step checks out code from the Source Code Management system into Jenkins. The trigger to checkout the code can be configured to various events like – scheduled to run a specific time of a day or when a check-in to SCM happens or manually on-demand.

1.3.2.Step 2: Build Code & Upload deployment package to Artifact Repository

The checked-out code is compiled, BAR file is created, zipped and uploaded to Nexus repository.

1.3.3.Step 3: Build Docker Image & Push to OpenShift Registry

In this step the BAR file is checked-out from Nexus repository and BAROVERIDES arrived. A docker image then created with base image as ACE-only or ACE with MQ Client and BAR file added on top of it. The resulting image is then pushed to the OpenShift Registry.

1.3.4.Step 4: Deploy to OpenShift Cluster

This step deploys the image into containers in OpenShift Cluster. HELM install script is triggered to deploy the docker image to OpenShift cluster.

3. USER'S MANUAL

3.1.Configure CICD Tools

3.1.1.Configure Jenkins-Github Integration using SSH keys

Follow below instructions to configure Jenkins and GitHub integration using SSH.

1) Create a 'jenkins' user on Build server where Jenkins is installed

```
sudo -su jenkins
ssh-keygen
eval $(ssh-agent -s)
ssh-add ~/.ssh/id_rsa
```

2) Add SSH keys to GitHub

```
cat /var/lib/jenkins/.ssh/id_rsa.pub
```

Go to github repository -> settings ->add SSH and GPG keys and click on 'New SSH Key'.

Enter a name in 'Title' and past the content of id_rsa.pub key.

3) Add SSH private key in Jenkins.

Go to Credentials from left pane inside Jenkins console and then click global.

After this click on 'Add Credential' and select 'SSH username with private key'. Enter the value for fields and enter private key ie. the content of id_rsa.

4) Add GitHub to known hosts.

Let's do it by simply connecting to GitHub server.

```
whoami # make sure it says 'jenkins'
ssh -T git@github.com
```
Result should be a welcome message.

3.1.2.Configure Nexus

In this solution we have chosen Nexus repository to store ACE bar files.

1) Create a repository in Nexus

Create a repository with the name 'releases' in nexus. The generated BAR files will be uploaded to this repository, which will be pulled while building the image for application.

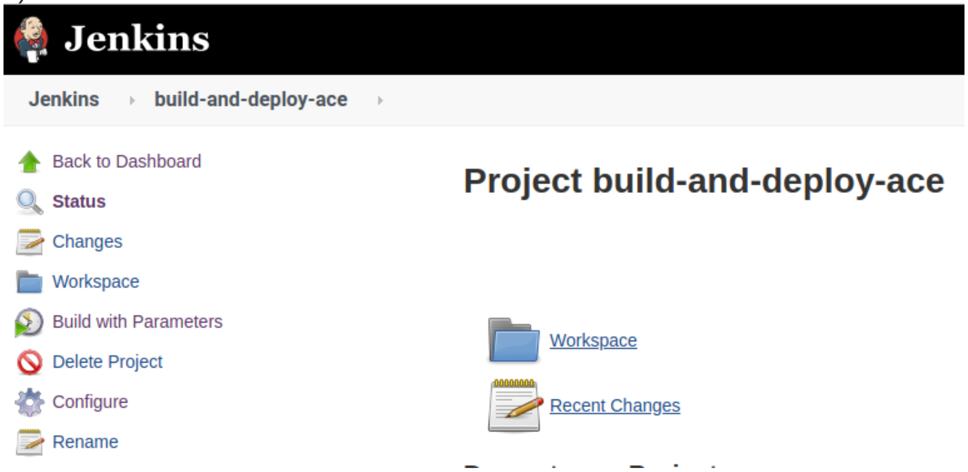
Login to nexus and go to 'Server administration and configuration'.

Go to Repositories –> Create repository and enter the information. Create it as 'maven2 (hosted)' repository.

3.2.Execute

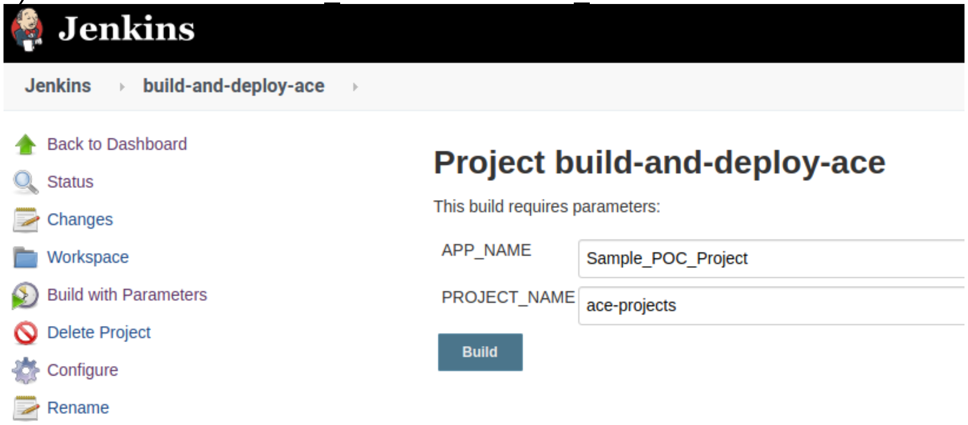
Steps to execute the migration using the One-click Deployment tool
1) Login to Jenkins console

2) Click on Project - build-and-deploy-ace
3) Click on 'Build with Parameters'



Project build-and-deploy-ace

4) Check the value for APP_NAME & PROJECT_NAME and click on BUILD



Project build-and-deploy-ace

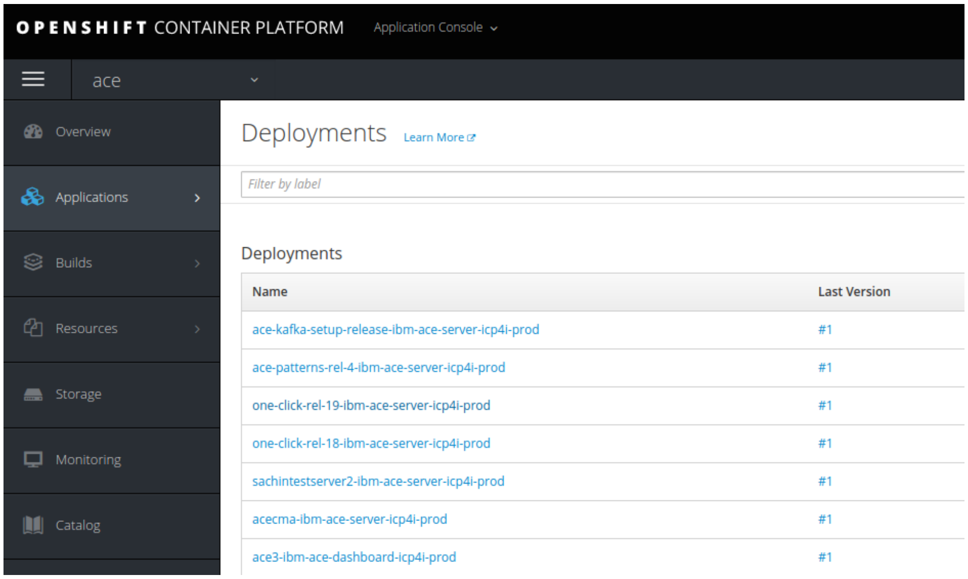
This build requires parameters:

APP_NAME: Sample_POC_Project
PROJECT_NAME: ace-projects

5) Monitor the console output for Projects - build-and-deploy-ace & build-image-and-deploy
6) The message 'deployment "rel-name-ibm-ace-server-icp4i-prod" successfully rolled out' at the end of console output denotes a successful deployment.

3.3.Validate

The successful deployments can be validated from the **OpenShift Web Console.**
1) On 'Application Console' select Project 'ace'
2) Go to 'Applicaitons > Deployments'. The deployment release name should be present in the list of deployments.



Deployments

Deployments

Name	Last Version
ace-kafka-setup-release-ibm-ace-server-icp4i-prod	#1
ace-patterns-rel-4-ibm-ace-server-icp4i-prod	#1
one-click-rel-19-ibm-ace-server-icp4i-prod	#1
one-click-rel-18-ibm-ace-server-icp4i-prod	#1
sachintestserver2-ibm-ace-server-icp4i-prod	#1
acecma-ibm-ace-server-icp4i-prod	#1
ace3-ibm-ace-dashboard-icp4i-prod	#1