|  |
| --- |
|  |

|  |
| --- |
| **Solution Approach** |
| **Deployment Automation via JENKINS** |
|  |
| This document’s purpose is to define the solution approach to carry out the PoC for deployment automation via Jenkins as agreed tool by OMW & HCL architects. |
|  |
| **Boominathan, Narayana Swamy (HCL)** |
| **11/22/2017** |
|  |

Table of Contents

[1. Background & Overview: 1](#_Toc499112831)

[1.1. Problem Statement: 1](#_Toc499112832)

[1.2. How this work is initiated: 1](#_Toc499112833)

[1.3. What this pack / document aims to achieve: 1](#_Toc499112834)

[2. Scope: 1](#_Toc499112835)

[2.1. In scope: 1](#_Toc499112836)

[2.2. Out of scope: 1](#_Toc499112837)

[3. Solution Overview : 1](#_Toc499112838)

[3.1. Jenkins Deployment Architecture: 1](#_Toc499112839)

[4. Technical Requirements: 1](#_Toc499112840)

[4.1. Servers required for Jenkins: 1](#_Toc499112841)

[4.2. Firewalls to be opened: 2](#_Toc499112842)

[4.3. Access Required: 2](#_Toc499112843)

[4.4. Environment used for PoC: 2](#_Toc499112844)

[5. Critical Success Factors: 2](#_Toc499112845)

[5.1. Factors using which the PoC success is measured 2](#_Toc499112846)

[5.2. Deployment is automated completely without any manual intervention 2](#_Toc499112847)

[5.3. Time taken for deployment (how much time save compared to manual deployment) 2](#_Toc499112848)

[5.4. Controlled deployment is possible from remote servers 2](#_Toc499112849)

# 

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision No.** | **Date of Release** | **Changes** | **Created/Updated By** |
| 0.1 | 22-Nov-2017 | Initial version |  |
|  |  |  |  |
|  |  |  |  |

# Background & Overview:

## Problem Statement:

BaNCS application is running on the servers which are having lot of micro service. While deployment of this application, manual changes required for the artifacts which is time consumption. Also, there is a possibility of manual errors.

## How this work is initiated:

Need to be discuss about this topic while in tomorrow call

## What this pack / document aims to achieve:

This document’s purpose is to define the solution approach to carry out the PoC for deployment automation via Jenkins as agreed tool by OMW & HCL architects.

# Scope:

## In scope:

In this PoC we are going to do the Deployment automation for the BaNCs hot fix deployment and TIBCO.

Below is the following covers in-scope

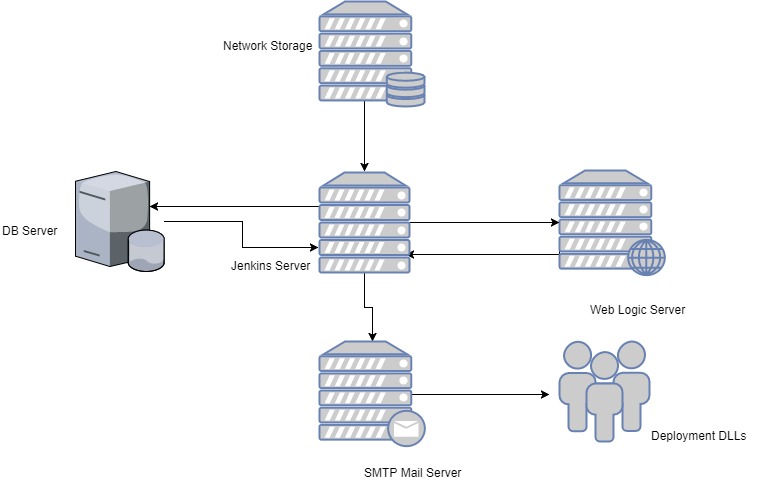
1. Scheduling deployment
2. Error logs and audit logs
3. Email Notification if any deployment fails
4. Handling the DB scripts
5. Deployment Validation checks through DB scripts

## Out of scope:

1. Products like Snr, Thunderhead, PX, Bods are not part of this POC/POV
2. BaNCS Full code deployment are not part of this POC
3. CI Build Creation(for BaNCS Application)
4. Only limited parts of BaNCS deployment features are included in this POC

# Solution Overview:

## Jenkins Deployment Architecture:



# Technical Requirements:

## Servers required for Jenkins:

One Jenkins server is enough for our PoC. In our future development we can use master and slave node architecture of Jenkins.

**About Master and Slave node main benfits:**

Sometimes we might need several different environments to test our builds. This cannot be done by a single Jenkins server.

If larger and heavier projects get built on a regular basis then a single Jenkins server cannot simply handle the entire load.

To address the above stated needs, Jenkins distributed (Master & Slave Node) architecture was introduced.

## Firewalls to be opened:

By default, we are going to use 9090 or 8080 ports for Jenkins while configuration.

## Access Required:

One service account with full admin access 1

The IP of the Jenkins server should be static

## Environment used for PoC:

|  |  |  |
| --- | --- | --- |
|  | IP Address | Server name |
| Jenkins | cc.aa.aad..ddd | Aaaaaa |
| Target Server | UAT IP Address(10.12.4.15) | UAT1 |

# Critical Success Factors:

## Factors using which the PoC success is measured

1. Reduce the Time taken of the deployment
2. Prevention of the Deployment error
3. Email Notification
4. Value

## Deployment is automated completely without any manual intervention

1. We can deploy automatically without manual intervention .But deployment job getting failed, the system will notify to the admin team through email. Then admin has to deduct the issues through logs and resolve it.
2. Error Notification

## Time taken for deployment (how much time save compared to manual deployment)

Right now it’s taking 45mins to 60mins for hot fix deployment. Through this automation we can reduce the time for deployment.

Note: Only we can able to reduce the file handling changes timings not the DB execution and file copying those things will be same time as manual.

## Controlled deployment is possible from remote servers