**Jenkins Documentation**

* Jenkins Installation
* Jenkins Installation in a Server
* Jenkins CI/CD/CT Pipeline Configuration
* Jenkins CI/CD/CT - Explanation
* Jenkins – CI/CD Scripts
  + Downloading the artifact from Nexus Repository
  + Deploying the artifact to glassfish Server
  + Other important scripts

**Jenkins Installation:**

* Download Jenkins (as a war file) for Windows/Linux System. War file name will be jenkins.war

**Link** :: <https://jenkins.io/download/>

* Go to the folder where Jenkins war is downloaded & open the command prompt
* Execute the below command,

**java -jar jenkins.war**

This will boot up the Jenkins Server. Once Jenkins server is up & running (status can be seen on the cmd or bash), hit the browser with the below URL.

<http://localhost:8080/jenkins>

Will be able to view the Jenkins Dashboard!

* To run Jenkins on a different or a customized port,

**java -jar jenkins.war --httpPort=<portNumber>**

**java -jar jenkins.war --httpPort=8454**

**Jenkins Installation on a Web Server:**

Jenkins can be run/booted up in any servers like Tomcat/Glassfish.

**Follow the below steps:**

1. Install Tomcat/Glassfish & bring up the server
2. Make sure that the Admin Console has sufficient rights to deploy any application
3. Go to the console of the Server & deploy the downloaded jenkins.war
4. Hit the browser with the below URL

<http://localhost:8080/jenkins>

**Jenkins CI/CD/CT Pipeline Configuration**

Jenkins CI/CD/CT Pipeline will have the following Jenkins Job & will be triggered depending on the result of the previous step in a sequential manner.

Let’s consider Application - Employee Management & this app will have the following Jenkins Job to build & deploy the App in DEV, QA & PROD with manual approval process while promoting to higher environments (i.e., from DEV to QA & from QA to PROD)

* empManagement-BUILD
* empManagement-DEV\_DEPLOY
* empManagement-EXECUTE\_SELENIUM
* empManagement-QA\_PROMOTION
* empManagement-QA\_DEPLOY
* empManagement-PROD\_PROMOTION
* empManagement-PROD\_DEPLOY

Developer will trigger the empManagement-BUILD Jenkins Job which in-turn will trigger the subsequent jobs upon Success & Deployment Jobs (except empManagement-DEV\_DEPLOY ) will be triggered only when the authorized person approves the Build Promotion

**Main Objective – BODA:**

CI/CD/CT Model’s main objective is BODA [Build Once, Deploy Anywhere] i.e., to build the application only once & then deploy it across the higher environments (in DEV, QA, UAT, Performance & PROD). So, the war generated in DEV Environment only will be deployed across all higher environments!

**Pipeline Configuration:**

On triggering, this job will check out the code from Git, Performs Sonar & Checkmarx Scan & Build the App

On Success, war file will be generated & triggers the DEV\_DEPLOY

On Success, deploys the app in DEV & triggers a mail to user & EXECUTE\_SELENIUM

Once the authorized person clicks the approval link sent in mail, triggers the PROD\_DEPLOY & deploys the app in PROD

On Success, triggers an approval mail to Authorized user for PROD Deployment

On Success, triggers an approval mail to Authorized user for QA Deployment

Once the authorized person clicks the approval link sent in mail, triggers the QA\_DEPLOY & deploys the app in QA & triggers the PROD\_PROMOTION

On Success, executes the Selenium test cases & triggers the QA\_PROMOTION

empManagement- PROD\_DEPLOY

empManagement- PROD\_PROMOTION

empManagement- QA\_DEPLOY

empManagement- QA\_PROMOTION

empManagement- EXECUTE\_SELENIUM

empManagement-DEV\_DEPLOY

empManagement-BUILD

**Jenkins CI/CD/CT - Explanation**

**Main Objective:**

* To build, test & deploy the Application in DEV, QA & PRODUCTION without manual intervention.
* To achieve BODA [Build Once Deploy Anywhere], i.e., Build the application only once in DEV Environment & deploy the same artifact (war/jar) across all higher environments

**Step by Step Process:**

* **empManagement-BUILD** – On Triggering this job,
  + Jenkins will **check out the application code from Git**.
  + Then it **executes Junit test cases and builds the Application & generate the Artifact** (war/jar).
  + Once the above step is successful, it then **executes Sonar** to identify code quality & code smells
  + Once the above step is successful, it **executes Checkmarx** to perform Security Scan of the application
  + Once this step is successful, it **uploads the Artifact (war/jar) to Nexus Repository Manager**.
  + Once the above step is successful, it triggers the **empManagement-DEV\_DEPLOY** & **deploys** the Artifact in **DEV Server** and sends a mail to the users
  + Once the above step is successful, it triggers the **empManagement-EXECUTE\_SELENIUM** & **executes the Selenium Test cases** and sends a mail to the users
  + Once the above step is successful, it triggers the **empManagement-QA\_PROMOTION** & sends a mail to the registered/authorized person **for approval for deploying the app in QA Environment**
  + Once the **authorized person clicks on the link** sent in the approval mail, he/she will be routed to Jenkins Dashboard & **Force Promotion** Button will be available. On clicking that, it triggers the **empManagement-QA\_DEPLOY** which in-turn will **deploy the Application in QA Environment**
  + Once the above step is successful, it triggers the **empManagement-PROD\_PROMOTION** & sends a mail to the registered/authorized person **for approval for** **deploying the app in PROD Environment**
  + Once the **authorized person clicks on the link** sent in the approval mail, he/she will be routed to Jenkins Dashboard & **Force Promotion** Button will be available. On clicking that, it triggers the **empManagement-PROD\_DEPLOY** which in-turn will **deploy the Application in PROD Environment**

**Jenkins – CI/CD Scripts**

**Script for Downloading the artifact (war file) from Nexus Repository:**

****

**Few things to be modified in the above script:**

1. In Line number – 9,10,11 – modify the URL Values of Nexus Repository

# Define Nexus Configuration

NEXUS\_BASE=http://10.10.176.183:8081/nexus

REST\_PATH=/service/local

ART\_REDIR=/artifact/maven/redirect

Above highlighted values need to be modified as per the Nexus URL

1. From Line number – 28 to 34, Maven Coordinates values needs to be modified

# Read in Complete Set of Coordinates from the Command Line

GROUP\_ID=MQMSystem-Web

ARTIFACT\_ID=MQMSystem-Web

VERSION=2.0.0

CLASSIFIER=""

PACKAGING=war

REPO=MQM-Release

VERBOSE=0

Above highlighted values need to be modified as per the Maven Coordinates values & **REPO** value refers to the value of Repository name created for the Application

1. In Line number – 103, download path needs to be modified

curl -sS -L -o /export/home/sacadmin/.jenkins/workspace/MQM-release\_2.0-QA\_DEPLOY/MQM-System.war -0 ${REDIRECT\_URL}

Above highlighted values need to be modified as per the downloading location.

**Deploying the artifact to glassfish Server**

**Script for Deploying the Artifact (war) to Glassfish Server:**

****

**Few things to be modified in the above script:**

1. In Line number -3, user name & password needs to be modified

curl --user admin:admin \

Highlighted in yellow refers to user name & in green refers to the password. This should be modified accordingly.

1. In Line number 9 & 10, war file location & context name needs to be modified

-F id=@/export/home/sacadmin/.jenkins/workspace/MQM-release\_2.0-QA\_DEPLOY/MQM-System.war \

-F contextroot=/MQM-System -F name=MQM-System \

Highlighted in Yellow refers to location of the Application war file & blue refers to the context name of the application & green refers to the name of the Application. This should be modified accordingly.

**Note** :: @ needs to be included in location of Application war file. Do not omit it!

1. In Line number 12, mention the application DNS name & port number

https://sacvl272.sac.flextronics.com:4848/management/domain/applications/application

Above highlighted in yellow refers to Server name & Port number. This should be modified accordingly.

**Note** :: Do not modify the rest of the URL values.

https://<Server DNS Name>:<Port Number>/management/domain/applications/application

**Other Important curl Scripts:**

**Uploading a file/artifact from one system to another securely (using SFTP)**

curl -k "sftp://sacvl842.sac.flex.com/export/home/sacadmin/glassfish-test/deployment/" --user "sacadmin:s.=F8E4q" -T "/export/home/sacadmin/.jenkins/workspace/testNexusRepoDownload/ElectronicPO4.war"

The above script will copy Artifact (ElectronicPO4.war) from one location & pastes in other location securely using Secure File Transfer Protocol (SFTP)

Highlighted in Pink refers to Source Location

Highlighted in Yellow refers to Destination Location

Highlighted in Blue refers to user name & green refers to password of Destination Location