Indus Production Line

Configuration Guide

Version 1.18-SNAPSHOT

2018-08-17

Table of Contents

Table 1. History

Date	Author	Detail Unresolved directive in subdocs/_init.adoc - include::D:\workspaceJava\cg-wm\target/generated-docs/history/ci-pf-configuration.adoc.psv[]

1. Introduction

This document describes, from scratch, how to configure CI Platform on BPM Factory.

This should work for any project using the Production Line from CG Indus team.

2. Prerequisites

2.1. CI PF Initialization

This has to be done only for a new Production Line

Browse and ask for a CI platform :

```
Service Catalog -> iProd -> Production Line - PLaaS -> Production Line Initialization
```

Later, every help tickets will be raised from here :

```
Service Catalog -> iProd -> Production Line - PLaaS -> Production Line Coaching and Support
```

When CI PF is up and running, test the access : https://bpmfactory.s2-eu.nvx.com

2.2. Technical account

This has to be done only for a new Production Line

Using a fake account created in LAM won't work : you can't login to Gerrit with it, hence no HTTP nor SSH key handling is possible.

2.2.1. CORP account

Ask for the creation of a technical account (for communication between Jenkins, SonarQube, Gerrit and Nexus) : raise a ticket in

```
Service Catalog -> iPlan -> Engagement Repository - ERaaS -> CTF: Service account creation
```

Add it to LAM (see below).

The user will have to be re-conducted every 6 months, a mail will be sent to administrator. The password never expires.

2.2.2. Administration

You can administer it here https://corporatedirectory.nvx.com in section **My Service Accounts**

2.3. Visio account with PC

If you have a TV where you want to display your devOps KPI, you have to order a PC with a specific account.

Information :

- Price is 220€ + 78€/month.
- 5-10 days estimated delivery.
- The "Tiny PC" cannot display more than HD.
- As of early 2018, it seems that only Dells can display 4K.

Steps :

- Prepare the list of hosts and ports the PC should access
- Browse https://nvxiticsprod.service-now.com/itics_ess/
- Go to **Order Services → Physical Computing → Communication → Project Skype Conferencing**
- Put the order, giving as much detail as possible (hosts/ports/HD/4K...). Then wait for the validation of the EM.
- Go buy the TV while waiting for the account.

2.3.1. Add hosts/ports

- Browse http://help.nvx.com/
- **IT Infrastructure Services → Network Connectivity → Request Now**
- Category → **Modify Network Security Rule**
  - Describe the new host/ports to access
  - Give detailed information
    - Login : SVC-FR-VCRNSBPMFACT
    - PC : LFR018678 / 10.69.240.109

3. Deploy asciidoc to github

```
mkdir docs
cd docs
git init
git checkout --orphan gh-pages
```

Copy a first version of the site in the directory, then :

```
git add *
git commit -m "initial site content"
git remote add origin "https://github.com/NeVraX182/docs.git"
git push --set-upstream origin gh-pages
```

4. LDAP Account Manager (lam)

Choose the service **lam** on your production line :

https://bpmfactory.s2-eu.nvx.com/lam/templates/lists/list.php?type=user

Add every user of your project to grant him access to your CI PF :

- **New user**
  - **First name**
  - **Last name** in uppercase
  - **Email adress**
  - **Unix**
    - **User name** : 8 char max, confirm with the person that it is his CORP login, else change
    - **Primary group** : users
  - **Save**

4.1. Mailing list

Follow these steps if and when you need to create a mailing list :

- Browser here : http://help.nvx.com/
- Under section **Popular Requests** click on **(Form) Request for new distribution list**.
- Expand the **Instructions** section and fill all fields
  - the name you provide will have .fr@nvx.com appended
  - for example, if you want a webMethods list for ADM, you can ask a list named "webmethods.adm" and the mailing list will be webmethods.adm.fr@nvx.com
- Click **Submit**
- When created, you can administer it here : https://corporatedirectory.nvx.com
  - Section **My Distribution Lists**

5. FusionForge

Official documentation is here : https://km3.nvx.com/book/1071769

Choose the service "FusionForge" on your production line : https://bpmfactory.s2-eu.nvx.com/fusionforge/

5.1. Add admin rights

By default, only the PL owner has admin rights (create project, create trackers).

Select **Site Admin**, section **Global roles and permissions → Forge administrators → Edit Role**

Add CORP login next to **Add User** and click it.

The user should have admin rights right away, thus being able to see the **Site Admin** section.

5.2. Create project

In the **Site Admin** tab, section **Project Maintenance**, click on the link **Register New Project**.

Fill the required fields.

Now that the project is created, you can access it by going on the **Projects** tab, then **Project List** and finally clicking on the project name.

5.3. Create tracker

On the project homepage, click on **Tracker → Administration**.

Set name "TASKS" and a description, then hit **Submit**.

5.4. Add users

The user must have connected at least once on FusionForge to be available.

To add access to users, on the project homepage, click on **Admin**, then **Users and permissions**.

Two options :

- You know the CORP login, then you can put it in the box
- You don’t, then click on **Add Users From List**, then the first letter of the family name, then the user

5.5. Change your name

Your name can be "Forge Admin" at PL reception, here is how to change it :

- Click on **My Page** → Tab **My Account**
- Change your **First Name** and **Last Name**
- Hit save

6. Gerrit

Choose the service **Gerrit** on your production line : https://bpmfactory.s2-eu.nvx.com/gerrit

6.1. Initialization

This has to be done only for a new Production Line

6.1.1. Permissions

Jenkins user push

- Click on **People** → **List Groups** → **Non-interactive Users**
- Add Jenkins (your technical account) in the list

Deleting tags

- Click on **Projects** → **List** → **[All-projects]** → section **Access** → **Edit**
- Under **[Reference: refs/tags/*]**
  - Click on **[Add Permission...]** and select **Push**
  - Select group **Administrator** and click **Force Push**
  - Save Changes

Now you can delete tags from your projects, for ex :

```
git push --force --delete origin cg-wm-1.17.6
```

6.1.2. Verified status

- Click on **Projects** → **List** → **[All-Projects]** → section **General** → **Edit Config**
- Add this

```
[label "Verified"]
   function = MaxWithBlock
   value = -1 Fails
   value =  0 No score
   value = +1 Verified
```

- Click on **Save**, then **Close**
- Click on **Publish Edit**, then **Publish**, **[Code-Review+2]**, **Submit**
- Click on **Projects** → **List** → **[All-Projects]** → **Access** → **Edit**
- Under **[Reference: refs/heads/*]**
  - Click on **[Add Permission...]** and select **Label Verified**
  - Select group **Administrator**

- Select group **[Non-Interactive Users]**
- Save Changes

6.1.3. Fast Forward

By default, when projet submissions are not fast forward, final submitting a change will create a merge commit. The history is potentially doubled.

- Click on **Projects** → **List** → **[All-Projects]** → **General**
- Under **Submit Type**, select **Rebase if Necessary**

6.2. User preferences

Click on **YourName** → **Settings** → **Diff Preferences** and set **columns** = **120** (you will probably have to paste it due to a GUI bug)

6.3. Project creation

Create your GIT project by clicking on **Projects** → **Create New Project**

- Project Name = **cg-wm**
- Rights Inherit From = **All-Projects**
- Check that it has inherited correctly "Rebase if necessary", else change and save

6.3.1. Project git address

The git is visible in **Projects** → **[(gitweb)]**.

Something like :

- https://cric.pl.s2-eu.nvx.com/gerrit/cg-wm.git
- https://cric.pl.s2-eu.nvx.com/gerrit/cg-tibco.git
- https://cric.pl.s2-eu.nvx.com/gerrit/cg-talend.git

6.3.2. Users groups creation

For each project, create a reviewer list and a validator list.

- Go to **Projects** → **Create New Group**
- Reviewers list
  - Give a name, for example **[dge-reviewers]**
  - Add every developers / primary reviewers on the project
  - Click on **General**
  - Description = Reviewers (first level : +1)

    - Click **Save Description**
    - Check **[Make group visible to all registered users.]**
    - Click **Save Group Options**
- Validators list
    - Give a name for example **[dge-validators]**
    - Add technical responsible and a backup
    - Click on **General**
    - Description = "Validators (level 2 : +2)"
    - Click **Save Description**
    - check **[Make group visible to all registered users.]**
    - Click **Save Group Options**

7. Jenkins

7.1. Plugins installation

This has to be done only for a new Production Line

- Go to **Jenkins → Administration Jenkins → Gestion des plugins**
- Update all plugins which have an update available
- Select **Disponibles** (=available) and install :
  - Pipeline Maven Integration
  - Throttle Concurrent Builds Plug-in
    - To be able to force non concurrent builds
  - Xvnc
    - To have a virtual screen if needed in tests
  - Naginator
    - For retry on failure
  - Gerrit Trigger
    - To launch job on gerrit update
  - HTML Publisher plugin
    - To have the **Maven Reporting** link when "maven site" is launched
  - Monitoring
    - To see nice health data of Jenkins on https://bpmfactory.s2-eu.nvx.com/jenkins/monitoring
  - JUnit Attachments
    - for enhanced job reporting
  - Logstash
    - To send jenkins jobs output to logstash then elastic
  - diskcheck
    - Check filesystem space on slave before a build
  - disk-usage
    - Show disk usage per build, configuration in **Administrer Jenkins → [Configurer le système] → Utilisation du disque**
  - AnsiColor
    - To allow colors in build logs
  - Simple Theme Plugin
    - to change Jenkins basic theme

7.2. Change Theme

This has to be done only for a new Production Line

- Have the Simple Theme Plugin installed
- Navigate **Administrer Jenkins** → **[Configurer le système]** → **Theme** section
  - URL of theme CSS = https://cdn.rawgit.com/afonsof/jenkins-material-theme/gh-pages/dist/material-cyan.css
    - see the author's page for other colors : http://afonsof.com/jenkins-material-theme/
  - Save

7.3. Gerrit Trigger Configuration

This has to be done only for a new Production Line

On Jenkins :

- Create the console-master job if not already existing

  Create a new freestyle job.

  Name it console-master

  General

  ☑ **[Restreindre où le projet peut être exécuté]**
    - master

  Put this **Build** → **[Ajouter une étape au build]** → **[Exécuter un script shell]** → paste this and save :

```
ssh-keygen -y -f /root/.ssh/id_rsa > /root/.ssh/id_rsa.pub
ls -lart /root/.ssh/
more /root/.ssh/id_rsa.pub
```

- Add 1 executor on the master node
  - **Home** → **[État du lanceur de compilations]** → **[maître]** → **Configurer**
- Execute the console-master
- Keep track of what the execution gave for later Gerrit configuration, example :

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDKGER5oLwkNhcCYtTzmUQooA+1mdrjIGi84AVsOHyNpsMqFBhkpxfImvopvKlYiztXUA15dwwDsPWq1tUcy/4N
WqKnMTQA57xxxT2r8suF/DVlH6fNn8T73mGz9+kT77FXHuaMfmDTqrwPngUYQMm2Y9kTjGhIcH/jseq6jCUawITAOs/6EUbs7jtJ/S+jMb6Ed60S7S/n
R3IzQwVrXMiQjDdFsL8RWEBQ54T4cNia/HMI8MK7mEEF5K008g4Ru3Bidk+VSisPUYFPmNc/tE12RyAjvkcwWxrYqFEB5h6RlS0yWXAjCUzjv8T0ov4W
us+ZqNgqUMYtBBf+zQvQC1ub
```

- When finished, remove the executor from master node
- Create a local trigger server

- **Home → Administrer Jenkins → Gerrit Trigger → Add New Server**
  - Gerrit Connection Setting
    - Name = local_server
    - Hostname = gerrit
    - Frontend URL = http://gerrit/
    - SSH Port = 29418
    - Username = svc-fr-bpmfact
    - SSH Keyfile = /root/.ssh/id_rsa
  - Gerrit Reporting Values
    - Verify = <vide>, 1, -1, -1, -1
    - Code Review = <vide>, 1, -1, -1, -1
  - Gerrit Verified Commandes
    - Started = vide
    - Successful =

```
gerrit review <CHANGE>,<PATCHSET> --message 'Build Successful (      ) <BUILDS_STATS>' --verified
<VERIFIED>
```

    - Failed =

```
gerrit review <CHANGE>,<PATCHSET> --message 'Build Failed (  _  ) <BUILDS_STATS>' --verified
<VERIFIED>
```

    - Unstable =

```
gerrit review <CHANGE>,<PATCHSET> --message 'Build Unstable (   °       °)  <BUILDS_STATS>' --verified
<VERIFIED>
```

    - Not Built =

```
gerrit review <CHANGE>,<PATCHSET> --message 'No Builds Executed (      ,)  <BUILDS_STATS>' --verified
<VERIFIED>
```

  - Save

On Gerrit :

- Connect with the technical user (svc-fr-bpmfact / Bpm-fact0ry)
  - You may have to use a secondary browser, since authentication is very persistent on Gerrit
- Click on the user top right → **Settings** → **SSH Public Keys** → **[Add Key…]**
- Add the public key content from Jenkins server (the one asked to be kept track earlier), starting with **ssh-rsa**

On Jenkins :

- Test the earlier configured connection of the trigger with **Test Connection** while editing local_server
- Restart jenkins with : https://bpmfactory.s2-eu.nvx.com/jenkins/safeRestart
- The Gerrit trigger should be up and running

7.4. Administration

This has to be done only for a new Production Line

7.4.1. Overall configuration

Connect to Jenkins configuration page : https://bpmfactory.s2-eu.nvx.com/jenkins/configure

Propriétés globales

- JAVA_HOME = /usr/

Jenkins Location

- Adresse email de l'administrateur système = xxxxx@nvx.com

Extended E-mail Notification

- SMTP server = smtp.nvx.fr
- Default user E-mail suffix = @nvx.com

Notification par email

- Serveur SMTP = smtp.nvx.fr
- Suffixe par défaut des emails des utilisateurs = @nvx.com

Save.

7.4.2. Tools configuration

Connect to Jenkins tools configuration page : https://bpmfactory.s2-eu.nvx.com/jenkins/configureTools/

Maven

- Nom = Maven 3.5
- Version = 3.5.2

Logstash Plugin

- Indexer type = ELASTICSEARCH
- Host name = http://frpardge.corp.nvx.com
- Port = 9200
- Key = /jenkins/builds

Save.

7.5. Allow CSS on published HTML

This has to be done only for a new Production Line

- Create a pipeline "css-support"
- Build Triggers
  - Construire périodiquement
    - Planning = 0 10,15,20 * * *
- Pipeline

```
println(System.getProperty("hudson.model.DirectoryBrowserSupport.CSP"))
System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "")
println(System.getProperty("hudson.model.DirectoryBrowserSupport.CSP"))
```

  - Uncheck **Use Groovy Sandbox** and save

7.6. SonarQube token

To be able to upload quality results to SonarQube, you have to create a token.

Go to SonarQube application on the PL → **YourName** → **My Account** → **Security** → Name = Jenkins → **Generate**

Now maven can upload results to SonarQube with something like :

```
mvn sonar:sonar -Dsonar.login=ab7451586619e21d0e2bb50389899ce3595e3 -Dsonar.host.url=http://sonarqube:9000/sonarqube
```

7.7. ssh key on remote server

This has to be done only for a new remote server

If you have a remote server where you deploy your artifacts for further developments or tests : * note the result of the slavePrep.sh script under **Here is this server's ssh public key**. Here is an example

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDZRLfTsI+cTRjbhYhnDvIOI3lsexMiJpwcBmeuJrISnEdh1LRPlviQjtI1h7NCihejVIPgvzyMVn3tMLsvABBXLTbV
FIetOudpJn+8isnYAWWaaqX2fce/BqjLC26ygR4n25sqTO/GE9AhV5uBPbYTr4HCrH9Wzd8nU13DXm8COhxUKh1+Uwm47KB11fVH/boIUygocIRu1FXS9TJy
MU0qFf3GGmDXs56VTe4ZQtPBHJ1klRXQQc6UIhTbdLpedo4Khvzr7TpdVZg13qXZt35/t7Gu4lbImHSlN64TKhaxAYgCPjYKgl9tAWJpEkk3WzXghohLivIQ
PInu5h3uvckH jenkins@b43496a2520e
```

- Connect on the remote server via SSH
- add the key to **~/.ssh/authorized_keys** file

7.8. Pipelines creations

7.8.1. The Review pipeline

This will be the review pipeline with steps from checkout to quality check. This pipeline is a "pipeline as code".

Go to Jenkins home page :

https://bpmfactory.s2-eu.nvx.com/jenkins

- Click **New Item**
- Choose a name : CG-WM_P1_Review
- Choose Pipeline type

General

- Description = This is the review pipeline fired by Gerrit on non yet validated push
- Check **Supprimer les anciens builds**
  - Strategy = Log Rotation
  - Nombre de builds à conserver = 10

Build Triggers

In the field **Choose a Server**, **Any Server** won't work

- Choose **Gerrit event**

Gerrit Trigger

- Choose a Server = local_server
- Trigger on = Patchset Created
- Gerrit Project
  - Type = Plain
  - Pattern = cg-wm
  - Branches
    - Type = Plain
    - Pattern = master

Advanced Project Options

None.

Pipeline

- Definition = Pipeline script

Pipeline content to copy/paste

```
#!groovy

properties([
    buildDiscarder(logRotator(artifactDaysToKeepStr: '', artifactNumToKeepStr: '', daysToKeepStr: '', numToKeepStr:
'7')),
    [$class: 'ThrottleJobProperty',
        categories: [],
        limitOneJobWithMatchingParams: false,
        maxConcurrentPerNode: 0,
        maxConcurrentTotal: 0,
        paramsToUseForLimit: '',
        throttleEnabled: false,
        throttleOption: 'project'],
    pipelineTriggers([
        gerrit(customUrl: '',
            gerritProjects: [[branches: [[compareType: 'PLAIN', pattern: 'master']],
            compareType: 'PLAIN', disableStrictForbiddenFileVerification: false, pattern: 'cg-wm']],
            serverName: 'local_server',
            triggerOnEvents: [patchsetCreated(excludeDrafts: false, excludeNoCodeChange: false, excludeTrivialRebase:
false)]
        )
    ])
])

node {
    timeout(30) {
        try {
            stage('Checkout') {
                cleanWs()   // requires workspace cleanup plugin to be installed
                echo "**** Starting checkout of patchset ${GERRIT_PATCHSET_NUMBER} on change number
${GERRIT_CHANGE_NUMBER}"
                git username: 'svc-fr-cric', password: 'Bocibo15', url: 'https://cric.pl.s2-eu.nvx.com/gerrit/cg-wm.git'
                def changeBranch = "change-${GERRIT_CHANGE_NUMBER}-${GERRIT_PATCHSET_NUMBER}"
                sh "git fetch origin ${GERRIT_REFSPEC}:${changeBranch}"
                sh "git checkout ${changeBranch}"

                def v = version(readFile('pom.xml'))
                echo "Building version ${v}"
            }
            stage('Compilation') {
                //slaves are wiped out randomly, so we prepare them on each execution
                sh '$WORKSPACE/src/scripts/slavePrep.sh'

                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    //clean to deploy libs to local maven repository
                    sh "mvn clean dependency:purge-local-repository"
                    //The assembly is postponed : it needs some further generated PDF
                    sh "mvn install verify -DskipTests -Dassembly.skipAssembly=true"
                }
            }
            stage('Verification'){
                parallel (
                    "Unit Tests" : {
                        wrap([$class: 'Xvnc', takeScreenshot: false, useXauthority: true]) {
                            withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled:
true)]) {

                                sh "mvn test -s cg-settings.xml -Dcheckstyle.skip=true"
                                //Maven auto reports JUnit surefire results
                            }
                        }
                     },
                    "Documentation" : {
                        sh '$WORKSPACE/src/scripts/asciidocOnlyModified.sh'
```

```
                        //get history from git to asciidoc documentation
                        sh '$WORKSPACE/src/scripts/asciidocHistory.sh $WORKSPACE'
                        withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled:
true)]) {
                            //validate produces the date for PDF
                            sh "mvn validate asciidoctor:process-asciidoc antrun:run@pdfsAddVersion -s cg-settings.xml
-Dcheckstyle.skip=true"
                        }
                        archiveArtifacts artifacts: '**/*.pdf', excludes: '**/test*.pdf', allowEmptyArchive: true
                    }
                )
            }
            stage('Integration Tests'){
                //integration tests have to be after documentation for the tracker zip to include the user manual
                wrap([$class: 'Xvnc', takeScreenshot: false, useXauthority: true]) {
                    withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)])
{
                        try{
                            //we do not install, since these suspicious jars could be misused by other projects
                            sh "mvn verify failsafe:verify -Dcg.ut.skip=true -Dcheckstyle.skip=true"
                        } finally {
                            //Maven does not auto report JUnit failsafe results
                            junit '**/target/failsafe-reports/*.xml'
                        }
                    }
                }
            }
            stage('Quality Gate') {
                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    sh "mvn sonar:sonar -Dsonar.login=0d1356516289799b179c6c7f851c9d4464ab04e2
-Dsonar.host.url=http://sonarqube:9000/sonarqube"
                }
                sh '$WORKSPACE/src/scripts/sonarStatus.sh'
            }
            stage('Assembly') {
                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    sh "mvn install -DskipTests -Dcheckstyle.skip=true"
                    sh "mvn dependency:purge-local-repository"
                }
                archiveArtifacts artifacts: '**/target/*.zip'
            }
        } catch (any) {
            step([
                $class: 'Mailer', notifyEveryUnstableBuild: true,
                recipients: emailextrecipients([[$class: 'CulpritsRecipientProvider'],
                [$class: 'RequesterRecipientProvider']])
                ])
            currentBuild.result = 'FAILURE'
        }
    }//timeout
    logstashSend failBuild: false, maxLines: 1000
}//node

@NonCPS
def version(text) {
    def matcher = text =~ '<version>(.+)</version>'
    matcher ? matcher[0][1] : null
}
```

7.8.2. The Deploy pipeline

This will be the main pipeline with everything from checkout to deployment. This pipeline is a “pipeline as code”.

Go to Jenkins home page :

https://bpmfactory.s2-eu.nvx.com/jenkins

- Click **New Item**
- Choose a name : CG-WM_P2_Deploy
- Choose **Pipeline** type

General

- Check **Supprimer les anciens builds**
  - Strategy = Log Rotation
  - Nombre de builds à conserver = 10

Build Triggers

- Choose « Scrutation de l'outil de gestion de version »
- Planning = H * * * *

Advanced Project Options

None.

Pipeline

Definition = Pipeline script from SCM

SCM = Git

- Repositories
  - Repository URL = http://bpmfactory.s2-eu.nvx.com/gerrit/p/cg-wm.git
  - Credentials = svc-fr-bpmfact / Bpm-fact0ry
- Branches to build : */master

Script Path = Jenkinsfile-2-deploy-to-dev

☑ Lightweight checkout

Pipeline content (for information)

```
#!groovy
node {
    timeout(60) {
        try {
            stage('Checkout') {
                cleanWs()   // requires workspace cleanup plugin to be installed
                retry(3) {
                    checkout scm
                }
                def v = version(readFile('pom.xml'))
                echo "Building version ${v}"
            }
            stage('Compilation') {
                //slaves are wiped out randomly, so we prepare them on each execution
                sh '$WORKSPACE/src/scripts/slavePrep.sh'
                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    //used to deploy libs to local maven repository
```

```
                    sh "mvn clean"
                    //The assembly is postponed : it needs some further generated PDF
                    sh "mvn install -DskipTests -Dassembly.skipAssembly=true"
                }
            }
            stage('Unit Tests') {
                wrap([$class: 'Xvnc', takeScreenshot: false, useXauthority: true]) {
                    withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)])
{
                        sh "mvn test -Dcheckstyle.skip=true"
                        //Maven auto reports JUnit surefire results
                    }
                }
            }
            stage('Documentation') {
                //get history from git to asciidoc documentation
                sh '$WORKSPACE/src/scripts/asciidocHistory.sh $WORKSPACE'
                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    //validate produces the date for PDF
                    //javadoc:aggregate is CPU intensive, we don't parallelize for now
                    sh "mvn validate asciidoctor:process-asciidoc antrun:run@pdfsAddVersion javadoc:aggregate
-Dcheckstyle.skip=true"
                    sh "mvn javadoc:jar -pl cg-utils -Dcheckstyle.skip=true"
                }
                step([$class: 'JavadocArchiver', javadocDir: 'target/site/javadoc', keepAll: true])
                archiveArtifacts artifacts: '**/*.pdf,**/*-javadoc.jar', excludes: '**/test*.pdf'
            }
            stage('Integration Tests') {
                wrap([$class: 'Xvnc', takeScreenshot: false, useXauthority: true]) {
                    withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)])
{
                        try{
                            sh "mvn verify failsafe:verify -Dcg.ut.skip=true -Dcheckstyle.skip=true"
                        } finally {
                            //Maven does not auto report JUnit failsafe results
                            junit '**/target/failsafe-reports/*.xml'
                        }
                    }
                }
            }
            stage('Quality Check') {
                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    sh "mvn sonar:sonar -Dsonar.login=0d1356516289799b179c6c7f851c9d4464ab04e2
-Dsonar.host.url=http://sonarqube:9000/sonarqube"
                }
                sh '$WORKSPACE/src/scripts/sonarStatus.sh'
            }
            stage('Assembly') {
                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: false)]) {
                    sh "mvn install -DskipTests -Dcheckstyle.skip=true"
                }
                //archiveArtifacts is now in "Deployment" phase since we download packages
            }
            stage('Publication'){
                parallel (
                    "Deployment to Nexus and IS": {
                        withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled:
true)]) {
                            //sh 'mvn wagon:update-maven-3'
                            sh 'mvn deploy -DskipTests -Dassembly.skipAssembly=true -Dcheckstyle.skip=true -s cg-
settings.xml'
                        }
                        sh "ssh devops@frpardge.corp.nvx.com 'cd /opt/sagis/profiles/IS_default/bin;./restart.sh'"
                        sh '$WORKSPACE/src/scripts/deployJavadoc.sh'
                        sh '$WORKSPACE/src/scripts/getPackages.sh'

                        //SchemaSpy must not fail the deployment so we put it after deployment
                        sh '$WORKSPACE/src/scripts/schemaspy.sh'
                        publishHTML([
                            allowMissing         : false,
```

```
                            alwaysLinkToLastBuild: false,
                            keepAll              : true,
                            reportDir            : 'target/schemaspy',
                            reportFiles          : 'index.html',
                            reportName           : 'DB Schema'])

                        archiveArtifacts artifacts: '**/target/*.zip'
                    },
                    "Reporting" : {

                        //Git Inspector
                        sh 'mkdir target/gitinspector'
                        sh 'export PYTHONIOENCODING=utf-8 ; gitinspector --format=html -rTw >
target/gitinspector/index.html'
                        publishHTML([
                            allowMissing         : false,
                            alwaysLinkToLastBuild: false,
                            keepAll              : true,
                            reportDir            : 'target/gitinspector',
                            reportFiles          : 'index.html',
                            reportName           : 'Git Inspector'])

                        //Maven Site
                        withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled:
true)]) {
                            sh 'mvn site site:stage -DskipTests -Dcheckstyle.skip=true -s cg-settings.xml'
                        }
                        publishHTML([
                            allowMissing         : false,
                            alwaysLinkToLastBuild: false,
                            keepAll              : true,
                            reportDir            : 'target/staging',
                            reportFiles          : 'index.html',
                            reportName           : 'Maven Reporting'])
                    }
                )
            }
        } catch (any) {
            step([
                $class: 'Mailer', notifyEveryUnstableBuild: true,
                recipients: emailextrecipients([[$class: 'CulpritsRecipientProvider'],
                [$class: 'RequesterRecipientProvider']])
                ])
            currentBuild.result = 'FAILURE'
        }
    }//timeout
    logstashSend failBuild: false, maxLines: 1000
}//node

@NonCPS
def version(text) {
    def matcher = text =~ '<version>(.+)</version>'
    matcher ? matcher[0][1] : null
}
```

7.8.3. The Deploy Int pipeline

Describe this pipeline when stable

7.8.4. The Release pipeline

This is the release pipeline launched manually at will when an external release is needed. This pipeline is a "pipeline as code".

Go to Jenkins home page :

https://bpmfactory.s2-eu.nvx.com/jenkins

- Click **New Item**
- Choose a name : CG-WM_P3_Release
- Choose **Pipeline** type

General

- Check **Ce build a des paramètres**
  - Paramètre texte
    - RELEASE_VERSION
    - the release version, with pattern 1.YY.MM[.increment] (ex : 1.17.5.9)
- Check **Supprimer les anciens builds**
  - Strategy = Log Rotation
  - Nombre de builds à conserver = 10

Build Triggers

No trigger (manual launch).

Advanced Project Options

None.

Pipeline

Definition = Pipeline script from SCM

SCM = Git

- Repositories
  - Repository URL = http://bpmfactory.s2-eu.nvx.com/gerrit/p/cg-wm.git
  - Credentials = svc-fr-bpmfact / Bpm-fact0ry
- Branches to build : */master

Script Path = Jenkinsfile-4-release

☑ Lightweight checkout

Pipeline content (for information)

```
#!groovy

//Release is a manual firing (and should always be)
//No need to do the whole process, trunk is always trustworthy with our setup
//Just check that the merge pipeline (DeployToDev) is successful

node {
    timeout(30) {
```

```
        try {
            stage('Checkout') {
                cleanWs()   // requires workspace cleanup plugin to be installed
                retry(3) {
                    checkout scm
                }
                echo "Releasing version $RELEASE_VERSION"
            }
            stage('Documentation') {
                //get history from git to asciidoc documentation
                sh '$WORKSPACE/src/scripts/asciidocHistory.sh $WORKSPACE'

                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    //to put jars in local maven repository if needed
                    sh "mvn clean"
                    sh "mvn versions:set -DnewVersion=$RELEASE_VERSION"
                    //without this local installation, modules are searched on internet on mvn validate
                    sh "mvn install -DskipTests -Dassembly.skipAssembly=true"
                    //we launch some (quick) tests that contains the generation of service list for the cg-utils doc
                    sh "mvn test -pl cg-utils"
                    //time to launch the actual doc generation
                    //validate produces the date for PDF
                    sh "mvn validate asciidoctor:process-asciidoc antrun:run@pdfsAddVersion javadoc:aggregate
-Dcheckstyle.skip=true"
                    sh "mvn javadoc:jar -pl cg-utils -Dcheckstyle.skip=true"
                }

                step([$class: 'JavadocArchiver', javadocDir: 'target/site/javadoc', keepAll: true])
                archiveArtifacts artifacts: '**/*.pdf,**/*-javadoc.jar', excludes: '**/test*.pdf'
            }
            stage('Deployment') {
                //Deployment is after documentation because a pdf must be in the zip

                //Delete tag if this is a replayed-on-error build...

                //...locally
                sh "git tag -d cg-wm-$RELEASE_VERSION || true"

                //...remotely
                //Special characters have to be URL encoded : https://stackoverflow.com/questions/6172719/escape-
character-in-git-proxy-password
                sh "git push --force --delete https://svc-fr-cric:ptTpilL5FS47RHDFV8541owV4zkbZOtVrxyqRsmGhw@cric.pl.s2-
eu.nvx.com/gerrit/p/cg-wm.git cg-wm-$RELEASE_VERSION || true"

                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    sh "mvn deploy scm:tag -s cg-settings.xml -DskipTests -Dcheckstyle.skip=true"
                }
                sh "$WORKSPACE/src/scripts/deployJavadoc.sh"
                sh "$WORKSPACE/src/scripts/getPackages.sh"
                archiveArtifacts artifacts: '**/target/*.zip'
            }
            stage('Reporting') {
                withMaven(maven: 'Maven 3.5', mavenOpts: '-Xmx1024M', options: [artifactsPublisher(disabled: true)]) {
                    sh "mvn site site:stage -DskipTests -Dcheckstyle.skip=true -s cg-settings.xml"
                }
                publishHTML([
                    allowMissing         : false,
                    alwaysLinkToLastBuild: false,
                    keepAll              : true,
                    reportDir            : 'target/staging',
                    reportFiles          : 'index.html',
                    reportName           : 'Maven Reporting'])
            }
        } catch (any) {
            step([
                $class: 'Mailer', notifyEveryUnstableBuild: true,
                recipients: emailextrecipients([[$class: 'CulpritsRecipientProvider'],
                [$class: 'RequesterRecipientProvider']])
                ])
            currentBuild.result = 'FAILURE'
```

```
        }
    }//timeout
    logstashSend failBuild: false, maxLines: 1000
}//node

@NonCPS
def version(text) {
    def matcher = text =~ '<version>(.+)</version>'
    matcher ? matcher[0][1] : null
}
```

7.9. Troobleshooting

7.9.1. Disk space usage > 90 %

If the disk space usage is too high and your build fails a the start for this reason, you can purge some folders with the below actions.

- Edit the **console** job.
- Put these lines and save :

```
du --max-depth=1 /home/jenkins/workspace/ | sort -n -r | head -n 30
find /home/jenkins/workspace/ -maxdepth 1 -mtime +90 -type d -depth -print
```

- Launch the job
- Following the results, do the necessary deletions
- If there are some ws-cleanup directory, you can delete them safely :

```
rm -rf /home/jenkins/workspace/\*ws-cleanup*/ ???
```

8. SonarQube

8.1. Quality Gates

Click on **Quality Gate**

Create a new one for your project and select your quality gate rules :

- Comments is less than 30 then 25
- Coverage is less than 86 then 80
- Maintainability Raiting is worse than A then C
- Reliability Raiting is worse than A then C
- Security Raiting is worse than A then C
- Unit Test Success (%) is worse than A then C

8.2. Update

This has to be done only for a new Production Line

Administration → System → Update Center

- SonarJava : install or update to latest
- Findbugs : install
- SoftVis3D : install

8.3. Rules / quality profile

8.3.1. Export

- Navigate to **Quality Profiles → BCT** (or your profile) → **Actions → Back Up**
- This will download a XML
- On CG-WM, the common java XML quality profile is saved here :

```
src\docs\SonarQube.qualityProfile.BCT.xml
```

8.3.2. Import

for an import to be successful, SonarQube must know the rules, so SonarJava and Findbugs must be up to date

- Navigate to **Quality Profiles** → dropdown menu next to **Create → Restore Profile**

- Provide the saved XML

8.3.3. Create

This has to be done only if you want to define a custom set of rules

- Choose **Quality Profiles → Java → Sonar way → Copy**
- Select the newly created profile
- Now you can :
  - Click on the number of active rules to deactivate some
  - Click on **Activate More**
  - Go to **Rules**, search one and change its value

8.4. Technical Debt

This has to be done only for a new Production Line

- Select **Administration → Configuration → General Settings → Technical Debt**
- Set Maintainability rating grid = 0.01,0.02,0.03,0.05
  - *This means that a tech debt < 1% is rated A, a tech debt > 5% is rated E.*
- Hit **save technical debt**

9. Nexus3

This has to be done only for a new Production Line

The technical account has to be created in CORP and defined in LAM as an administrator

9.1. URL

Repositories, release, snapshot and group, should already be created.

The public repository address can be found there :

- Click on the wheel
- **Repository** → **Repositories** → **maven-public**
- Get the URL

9.2. Set SNAPSHOT retention

This has to be done only for a new Production Line

- Go to **Administration** (wheel) → **System** → **Tasks** → **Create task**
- Type = Remove Snapshots From Maven Repository
- Fields
  - Name = Delete old snapshots
  - Repository = (All Repositories)
  - Minimum snapshot count = 5
  - Snapshot retention (days) = 7
  - Delete immediately = [x]
  - Task frequency = Daily
  - Time to run this task = 8:00

9.3. Add a proxy (=remote) repository

- Connect to Nexus
- Click on **Administration** (wheel) → **Repositories** → **Repository** → **Create repository**
- Choose **maven2 (proxy)**
- Fill repository data
  - Name = Wakaleo

    - Remote Storage = http://www.wakaleo.com/maven/repos
    - Download Remote Indexes = False
- Save
- On the corresponding line in the list, click on **Analyze**

Now artifacts should be accessible via maven if the settings.xml is well configured to reach the Nexus :

Settings.xml using Nexus

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">

    <servers>
    <!-- tag::server[] -->
        <server>
            <id>nexus-local</id>
            <username>svc-fr-cric</username>
            <password>ESBpwd2018</password>
        </server>
    <!-- end::server[] -->
        <server>
            <id>iserver-dev</id>
            <username>devops</username>
            <password>devopswm</password>
        </server>
        <server>
            <!-- Gerrit credentials for maven-scm-plugin, which search with the hostname
                as server id -->
            <id>cric.pl.s2-eu.nvx.com</id>
            <username>svc-fr-cric</username>
            <!-- Gerrit does not use the CORP password, we use the UI generated one -->
            <password>ptTpilL5FS47RHDFV8541owV4zkbZOtVrxyqRsmGhw</password>
        </server>
    </servers>

    <!-- tag::mirrors[] -->
    <mirrors>
        <mirror>
            <id>nexus-local</id>
            <mirrorOf>*</mirrorOf>
            <url>https://cric.pl.s2-eu.nvx.com/nexus3/repository/maven-public/</url>
        </mirror>
    </mirrors>
    <!-- end::mirrors[] -->

    <!-- tag::profiles[] -->
    <profiles>
        <profile>
            <id>nexus-local</id>
            <repositories>
                <repository>
                    <id>central</id>
                    <url>http://central</url>
                    <releases>
                        <enabled>true</enabled>
                    </releases>
                    <snapshots>
                        <enabled>true</enabled>
                    </snapshots>
                </repository>
            </repositories>

            <pluginRepositories>
```

```
                <pluginRepository>
                    <id>central</id>
                    <url>http://central</url>
                    <releases>
                        <enabled>true</enabled>
                    </releases>
                    <snapshots>
                        <enabled>true</enabled>
                    </snapshots>
                </pluginRepository>
            </pluginRepositories>

            <properties>
                <snapshotUrl>https://cric.pl.s2-eu.nvx.com/nexus3/repository/maven-snapshots/</snapshotUrl>
                <releaseUrl>https://cric.pl.s2-eu.nvx.com/nexus3/repository/maven-releases/</releaseUrl>
            </properties>

        </profile>
    </profiles>
    <!-- end::profiles[] -->

    <!-- tag::activeProfiles[] -->
    <activeProfiles>
        <activeProfile>nexus-local</activeProfile>
    </activeProfiles>
    <!-- end::activeProfiles[] -->


</settings>
```

Repositories to add

- For SchemaSpy : http://www.wakaleo.com/maven/repos

10. JIRA

Jira is used here in association with BugPicture, to have a Gantt chart.

10.1. JIRA Access restriction

To restrict the access to stay under the 10 members for the cheapest license ($12 with tax once and for all) :

- in LAM, create the jira-users group, and put in all users you want to give access to JIRA
- browse user management in JIRA : https://dge.pl.s2-eu.nvx.com/jira/secure/admin/user/UserBrowser.jspa
  - Give you the **jira-administrators** right and remove **pladmin** from this group
  - define **jira-users** as default group
  - remove all groups EXCEPT **jira-administrators** and **jira-users**

10.2. Project

10.2.1. Add Time Estimated

To add **time estimated** *in Kanban*

- Edit an issue
- Where is my field ? : "Time Tracking"
- Click on the link to add it

Now it is added. To define to days as default unit :

- **Wheel** → **Issues** → Left pannel **Time Tracking** → **Deactivate**
- Change configuration
  - **Time format** = days
  - **Default Unit** = day
- Click **Activate**

10.2.2. Configure emails notification

To disable emails notification :

- **Projects** → **MyProject** → Bottom left corner **Project Settings** → **Notifications** → **Actions** → **Select a scheme** → **None**

10.3. Version

Perform these actions for each new version.

10.3.1. Add new members

If you have new members in this version, add them in LAM and especially in the **jira** group.

Update JIRA rights :

- Click on the wheel → **User management** → left section **User Directories**
- Click **Synchronize**

10.3.2. Create the version

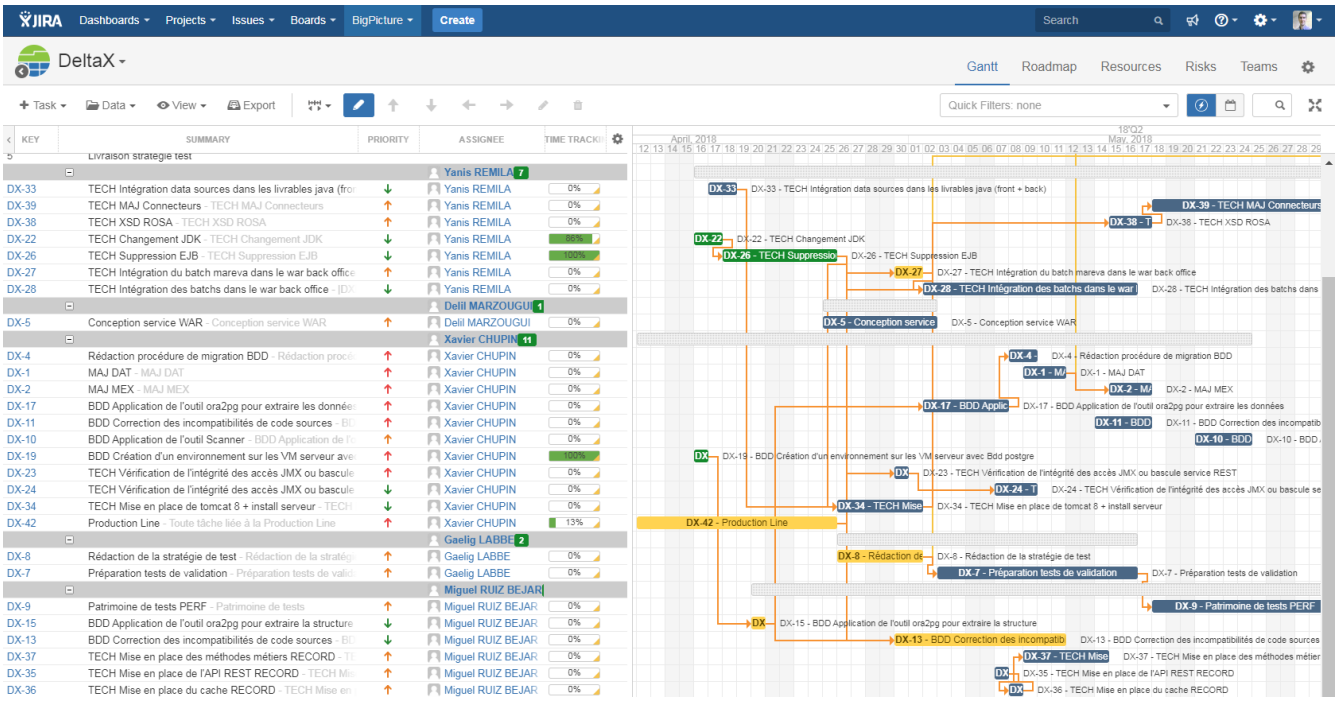
- **Projects** → **MyProject** → left section **Releases**
- Fill The data name/dates/description
- Click **Add**

10.3.3. Add tickets

- Add all tickets in JIRA
- Import all in BigPicture

10.4. BigPicture

BigPictures is a 10$ plugin that adds Gantt capabilities to JIRA. Install this plugin if you have a license or want to try it for a month.



10.4.1. Gantt

To see Gantt chart, go to **BigPicture → YourProject → Gantt**.

You can edit a ticket using "E" shortcuts when in GANTT view

10.4.2. Configure layout

To obtain the layout of the picture

- **BigPicture → Gantt**
- **View → Layout → Compact**
- **View → Perspective → Team**
- **View → Perspective → Manager Perspectives**
- using drag and drops, keep only these fields
  - Key, Summary, Priority, Assignee, Time Tracking
- save
- **BigPicture → Gantt**
- **Data → Sort tasks A-Z → Assignee**

10.4.3. Regroup tasks

To regroup tasks under bigger subjects in Gantt view

- **Task → Create task → Artificial Task**
- Put it before all wanted subtasks with **up arrow** or **down arrow**
- Select each task and make them subtask by clicking on →

10.4.4. Add deadlines

To add a deadline in Gantt view

- **Task → Create task → Artificial task**

11. Nexus2

Deprecated, use Nexus3

11.1. Define rights of technical account

This has to be done only for a new Production Line

The technical account has to be created in CORP and defined in LAM as an administrator

- Click on **Security** → **Users**
- Select **LDAP** (next to **Delete**) and search for your technical account name. It should appear in list below.
- In **Config** section associated, there is only **admins** Role.
- Add these roles :
  - **Nexus Deployment Role**
  - **Nexus Developer Role**
  - **Repo: All Repositories (Full Control)**

11.2. Set SNAPSHOT retention

This has to be done only for a new Production Line

- Go to **Administration** → **Scheduled Tasks** → **Add**
- Define the attributes :
  - Name = Delete old snapshots
  - Task Type = Remove Snapshots From Repository
  - Repository/Group = All Repositories
  - Minimum snapshot count = 5
  - Snapshot retention (days) = 7
  - Remove if released = [ ]
    - Else, you can end up loosing all SNAPSHOTS without notice !
  - Delete immediately = [x]
  - Recurrence = Daily
  - Recurring Time = 8:00

11.3. Add a proxy (=remote) repository

- Connect to Nexus
- Click on **Repositories** → **Add...** → **Proxy**
- Fill, example sonatype snapshot
  - Repository ID = sonatype-snapshots
  - Repository Name = Sonatype Snapshots
  - Repository Type = proxy
  - Repository Policy = Snapshot
  - Remote Storage Location = https://oss.sonatype.org/content/repositories/snapshots/
  - Download Remote Indexes = False
- Hit save
- Click on bottom section **Routing** → **Update now**
- Go back to **Repositories** and select **Public Repositories**
- In bottom section **Configuration** add your new repository to the right
- Hit save

Now artifacts should be accessible via maven if the settings.xml is well configured to reach the Nexus.

Repositories to add

- For SchemaSpy : http://www.wakaleo.com/maven/repos

12. Additionnal nvx VM : webMethods & Elastic servers

- VM host = **frpardge.corp.nvx.com**
- VM user = **devops**

12.1. Initialize VM

- Adding a user

```
adduser devops
```

- Granting him root privileges

```
visudo
```

```
devops ALL=(ALL:ALL) ALL
```

- Checking FS size

```
parted
print free
```

  - Example

```
Number  Start   End     Size    Type     File system  Flags
        32.3kB  1049kB  1016kB           Free Space
1       1049kB  500MB   499MB   primary  ext2         boot
2       500MB   53.7GB  53.2GB  primary               lvm
        53.7GB  53.7GB  1049kB           Free Space
```

Below instructions are for Ubuntu only. You can check your Linux distribution with this command : cat /etc/*-release

- Add some server for apt-get

```
sudo vi /etc/apt/sources.list
```

```
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty-security main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty-updates main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty-proposed main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty-backports main restricted universe multiverse
```

12.2. Install Docker

Below instructions are for Ubuntu 14 only. You can check your Linux distribution with this command : cat /etc/*-release

```
apt-get install apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable"
apt-get update
apt-cache search docker-ce
apt-get install docker-ce
```

- May work on Jenkins slave

don’t use on managed PL, we don’t have enough rights

```
sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-release; echo "$ID") \
  $(lsb_release -cs) \
  stable"
sudo apt-get update
sudo apt-cache search docker-ce
sudo apt-get install --assume-yes docker-ce
sudo dockerd
```

- Allow Docker remote API

Solution found here https://forums.docker.com/t/enable-remote-api-on-docker-hosts-in-ubuntu-14/11583/2

```
vi /etc/default/docker
```

```
DOCKER_OPTS="-H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock"
```

- Start Docker Daemon

```
sudo dockerd
```

  - To restart (as root)

Don’t forget the docker.sock chmod if you use metricbeat

```
service docker restart
```

- To check FS size

```
root@frpardge:/var/lib/docker
$ du -sh -- * .*
92K     aufs
44K     containers
116K    image
52K     network
20K     plugins
4.0K    swarm
4.0K    tmp
4.0K    trust
28K     volumes
4.0K    .
61M     ..
```

- Get rid of sudo for devops user

```
sudo groupadd docker
sudo gpasswd -a devops docker
newgrp docker
docker run hello-world
```

- Install **Portainer** to ease administration

```
sudo docker pull portainer/portainer
```

```
sudo docker run -d --name portainer --restart=always -p 19000:9000 -v /var/run/docker.sock:/var/run/docker.sock
portainer/portainer
```

- To use, go to http://frpardge.corp.nvx.com:19000
  - login/password = **admin** / **administrator**
    - Install docker-compose

```
curl -L https://github.com/docker/compose/releases/download/1.19.0/docker-compose-`uname -s`-`uname -m` -o
/usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

```
docker-compose --version
```

12.2.1. Define Nexus3 as the Docker registry

- Raise a ticket in INSERE to ask a port opening for Nexus3 as a Docker registry
  - They will provide this kind of response, which indicates how to login before 'docker push' :

```
docker login docker-registry-bpmfactory.s2-eu.nvx.com
User name: docker
User Password: dockerPWdbpmfactory
```

- Use the information to add the registry in docker configuration

```
vi /etc/docker/daemon.json
```

```
{
  "storage-driver": "devicemapper",
  "insecure-registries": [
    "docker-registry-bpmfactory.s2-eu.nvx.com"
  ],
  "disable-legacy-registry": true
}
```

- be carefull not to have INSECURE_REGISTRY here, it would not start :

```
vim /etc/sysconfig/docker
```

```
#INSECURE_REGISTRY='--insecure-registry userbxxy05.socle:8444'
```

  - Redémarrer docker

```
service docker restart
```

12.3. Setup a dockerized Oracle12c database

Database found here : https://hub.docker.com/r/sath89/oracle-12c/

```
docker pull sath89/oracle-12c
```

```
docker run --restart=always --name dbdev -d -p 18080:8080 -p 1521:1521 sath89/oracle-12c
```

```
docker logs -f feef20144fdc124d7b19d22aaf7bd63cbb837df667cc9764e7bdb5bcafa1af46
```

```
Database not initialized. Initializing database.
Starting tnslsnr
Copying database files
1% complete
3% complete
Import finished
Database ready to use. Enjoy! ;)
```

Connect to Oracle Application Express web management console with following settings :

- host = http://frpardge:18080/apex
- workspace = **INTERNAL**
- user = **ADMIN**
- password **0raclE!**

12.4. Install Elastic items

Configuration files are given in next associated sections below. For some of them, some chmod change is needed :

```
cd ~/elastic
chmod go-w ./*.yml
```

12.4.1. Migration prerequisites

If you are upgrading from a previous version of Elastic, you have to do this before anything :

- Close data senders using Portainer for containers
  - Shutdown the IS, or just disable CgElastic & WmMediator packages
  - Stop Heartbeat, Filebeat, Metricbeat containers
  - No need to stop Logstash if Filebeat is closed
- Check that nothing is coming in Elasticsearch with Kibana, then stop Kibana container
- Stop Elasticsearch container

For now, no data migration has been tried, so no support on it. This will be a fresh new Elasticsearch, and a Kibana with imported dashboards (hoping they still work).

Rename all stopped container, to be able to get the initial name on new containers.

12.4.2. Elasticsearch

If you are new to the Elastic Stack, learn with the excellent official Kibana tutorial : https://www.elastic.co/guide/en/kibana/current/getting-started.html

- Install with docker without x-pack

```
docker pull docker.elastic.co/elasticsearch/elasticsearch-oss:6.0.0
```

To start it

```
docker run --restart=always -d --name elastic -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node"
docker.elastic.co/elasticsearch/elasticsearch-oss:6.0.0
```

- if elastic stops directly after start with this error

```
max virtual memory areas vm.max_map_count [65530] likely too low, increase to at least [262144]
```

- Then type before retry

```
sudo sysctl -w vm.max_map_count=262144
```

12.5. Kibana

- Install with docker without x-pack

```
docker pull docker.elastic.co/kibana/kibana-oss:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/kibana.yml
```

- Start the container

```
docker run --restart=always -d --name kibana -p 5601:5601 -v
~/elastic/kibana.yml:/usr/share/kibana/config/kibana.yml docker.elastic.co/kibana/kibana-oss:6.0.0
```

Check that it is up and running : http://frpardge:5601/

Once every application is up, you will be able to declare patterns :

- cgwmbeat-*
- heartbeat-*
- jenkins
- logstash-*
- metricbeat-*
- webmethodsmediator

And to apply some Elasticsearch default index configuration :

- the limit of 1000 fields by index is a bit low, updated to 2000
- default is 5 shards per index, too many for dev
- default is 1 replica, for a single node ES it's 0

```
PUT _template/all
{
  "index_patterns" : ["*"],
  "settings": {
    "index.mapping.total_fields.limit": 2000,
    "index.max_docvalue_fields_search": 400,
    "number_of_shards": 1,
    "number_of_replicas": 0
  }
}
```

Here is something to try, inside the "PUT _template/all", someday, to not have keyword (fixed word) + text (searchable) but only keyword :

```
"dynamic_templates": [
  {
    "match_mapping_type": "string",
    "mapping": {
      "type": "keyword"
    }
  }
]
```

For Elasticsearch monitoring :

```
GET /_cat/indices?v
GET _cluster/health
```

~/elastic/kibana.yml

```
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "0.0.0.0"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy. This only affects
# the URLs generated by Kibana, your proxy is expected to remove the basePath value before forwarding requests
# to Kibana. This setting cannot end in a slash.
#server.basePath: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576

# The Kibana server's name.  This is used for display purposes.
#server.name: "your-hostname"

# The URL of the Elasticsearch instance to use for all your queries.
elasticsearch.url: "http://frpardge.corp.nvx.com:9200"

# When this setting's value is true Kibana uses the hostname specified in the server.host
# setting. When the value of this setting is false, Kibana uses the hostname of the host
# that connects to this Kibana instance.
#elasticsearch.preserveHost: true

# Kibana uses an index in Elasticsearch to store saved searches, visualizations and
# dashboards. Kibana creates a new index if the index doesn't already exist.
#kibana.index: ".kibana"

# The default application to load.
#kibana.defaultAppId: "discover"

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
#elasticsearch.username: "user"
#elasticsearch.password: "pass"

# Enables SSL and paths to the PEM-format SSL certificate and SSL key files, respectively.
# These settings enable SSL for outgoing requests from the Kibana server to the browser.
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# Optional settings that provide the paths to the PEM-format SSL certificate and key files.
# These files validate that your Elasticsearch backend uses the same key files.
#elasticsearch.ssl.certificate: /path/to/your/client.crt
#elasticsearch.ssl.key: /path/to/your/client.key
```

```
# Optional setting that enables you to specify a path to the PEM file for the certificate
# authority for your Elasticsearch instance.
#elasticsearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]

# To disregard the validity of SSL certificates, change this setting's value to 'none'.
#elasticsearch.ssl.verificationMode: full

# Time in milliseconds to wait for Elasticsearch to respond to pings. Defaults to the value of
# the elasticsearch.requestTimeout setting.
#elasticsearch.pingTimeout: 1500

# Time in milliseconds to wait for responses from the back end or Elasticsearch. This value
# must be a positive integer.
#elasticsearch.requestTimeout: 30000

# List of Kibana client-side headers to send to Elasticsearch. To send *no* client-side
# headers, set this value to [] (an empty list).
#elasticsearch.requestHeadersWhitelist: [ authorization ]

# Header names and values that are sent to Elasticsearch. Any custom headers cannot be overwritten
# by client-side headers, regardless of the elasticsearch.requestHeadersWhitelist configuration.
#elasticsearch.customHeaders: {}

# Time in milliseconds for Elasticsearch to wait for responses from shards. Set to 0 to disable.
#elasticsearch.shardTimeout: 0

# Time in milliseconds to wait for Elasticsearch at Kibana startup before retrying.
#elasticsearch.startupTimeout: 5000

# Specifies the path where Kibana creates the process ID file.
#pid.file: /var/run/kibana.pid

# Enables you specify a file where Kibana stores log output.
#logging.dest: stdout

# Set the value of this setting to true to suppress all logging output.
#logging.silent: false

# Set the value of this setting to true to suppress all logging output other than error messages.
#logging.quiet: false

# Set the value of this setting to true to log all events, including system usage information
# and all requests.
#logging.verbose: false

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000.
#ops.interval: 5000
```

12.5.1. Troubleshoot

Here is a list of problems and solutions.

Kibana cannot connect to Elasticsearch

If Kibana cannot connect to Elasticsearch with this message :

```
blocked by: [FORBIDDEN/12/index read-only / allow delete (api)];: [cluster_block_exception] blocked by:
[FORBIDDEN/12/index read-only / allow delete (api)];
```

Then apply these settings :

```
PUT _settings
   {
       "index": {
       "blocks": {
       "read_only_allow_delete": "false"
   }
```

```
PUT cgwmbeat-2018.02.16/_settings
   {
       "index": {
       "blocks": {
       "read_only_allow_delete": "false"
   }
```

12.6. Curator

```
wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

```
sudo vi /etc/apt/sources.list
```

```
deb [arch=amd64] http://packages.elastic.co/curator/5/debian stable main
```

```
sudo apt-get update && sudo apt-get install elasticsearch-curator
```

To start it

```
curator --config ~/elastic/curator.config.yml --dry-run ~/elastic/curator.delete_indices.yml
curator --config ~/elastic/curator.config.yml ~/elastic/curator.delete_indices.yml
```

12.6.1. Automation

- Create below script

~/elastic/curator.sh

```
#!/bin/sh

curator --config ~/elastic/curator.config.yml ~/elastic/curator.delete_indices.yml
```

- Open crontab

  ```
  crontab -e
  ```

- Add this line to launch it at 8:00 everyday

  ```
  0 8 * * * ~/elastic/curator.sh
  ```

- Exit and save with **Ctrl+X**, **Y**, **Enter**

12.6.2. Configuration

~/elastic/curator.config.yml

```
---
# Remember, leave a key empty if there is no value.  None will be a string,
# not a Python "NoneType"
client:
  hosts:
    - 127.0.0.1
  port: 9200
  url_prefix:
  use_ssl: False
  certificate:
  client_cert:
  client_key:
  ssl_no_validate: False
  http_auth:
  timeout: 30
  master_only: False

logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']
```

~/elastic/curator.delete_indices.yml

```
---
# Remember, leave a key empty if there is no value.  None will be a string,
# not a Python "NoneType"
#
# Also remember that all examples have 'disable_action' set to True.  If you
# want to use this action as a template, be sure to set this to False after
# copying it.
# # # #
# curator --config ~/elastic/curator.config.yml --dry-run ~/elastic/curator.delete_indices.yml
# curator --config ~/elastic/curator.config.yml ~/elastic/curator.delete_indices.yml
# # # #
actions:
  1:
    action: delete_indices
    description: Delete indices older than 30 days. No error when no actual deletion.
    options:
      ignore_empty_list: True
    filters:
    - filtertype: age
      source: name
      direction: older
      timestring: '%Y.%m.%d'
      unit: days
      unit_count: 30
```

12.7. Heartbeat

- Pull the image

```
docker pull docker.elastic.co/beats/heartbeat:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/heartbeat.yml
```

- Start the container

```
docker run --name heartbeat -d -v ~/elastic/heartbeat.yml:/usr/share/heartbeat/heartbeat.yml
docker.elastic.co/beats/heartbeat:6.0.0
```

~/elastic/heartbeat.yml

```
#
# wget --user=svc-fr-pldouane --password=Na9Is4Aw0! https://cdsdouane.pl.s2-eu.nvx.com/jenkins/job/DTXE_P1_CodeReview/
#
heartbeat.monitors:
- name: Jenkins
  type: http
  schedule: '@every 30s'
  urls: ["https://bpmfactory.s2-eu.nvx.com/jenkins/job/CNAV-DGE_P1_Review/"]
  username: svc-fr-bpmfact
  password: Bpm-fact0ry
  check.request.method: GET
  check.response.status: 200
- name: 'Jenkins Douane'
  type: http
  schedule: '@every 30s'
  urls: ["https://cdsdouane.pl.s2-eu.nvx.com/jenkins/job/DTXE_P1_CodeReview/"]
  username: svc-fr-pldouane
  password: Na9Is4Aw0!
  check.request.method: GET
  check.response.status: 200
- name: 'Gerrit home'
  type: http
  schedule: '@every 30s'
  urls: ["https://bpmfactory.s2-eu.nvx.com/gerrit/changes/?n=25&O=81"]
  username: svc-fr-bpmfact
  password: Bpm-fact0ry
  check.response.status: 200
- name: 'Gerrit viewFile'
  type: http
  schedule: '@every 30s'
  urls: ["https://bpmfactory.s2-
eu.nvx.com/gerrit/changes/421/revisions/5ab9d4c5cab6a087b936748f2df6550666a502dd/files/Jenkinsfile-2-deploy-to-
dev/diff?context=ALL"]
  username: svc-fr-bpmfact
  password: Bpm-fact0ry
  check.response.status: 200
- name: 'IS Dev'
  type: http
  schedule: '@every 30s'
  urls: ["http://frpardge:5555"]
  username: Administrator
  password: manage
  check.response.status: 200
- name: Kibana
  type: http
  schedule: '@every 30s'
  urls: ["http://frpardge:5601/app/kibana#/management?_g=()"]
  check.response.status: 200
- name: 'UM Dev'
  type: tcp
  schedule: '@every 30s'
  hosts: ["frpardge:9000"]

heartbeat.scheduler:
  limit: 10

output.elasticsearch:
  hosts: ["frpardge.corp.nvx.com:9200"]

dashboards.enabled: true
```

12.8. Logstash

Install this only if you have files to be parsed and sent to Elasticsearch

- Pull the image

```
docker pull docker.elastic.co/logstash/logstash-oss:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/logstash-pipelines/logstash.conf
```

- Start the container

```
docker run --restart=always --name logstash -d -p 5043:5043 -v ~/elastic/logstash-
pipelines/:/usr/share/logstash/pipeline/ docker.elastic.co/logstash/logstash-oss:6.0.0
```

~/elastic/logstash-pipelines/logstash.conf

```
input {
    beats {
        port => "5043"
    }
}

filter {
  if [fields][log_type] == "perflog" {
    grok {
        match => { "message" => "%{TIMESTAMP_ISO8601:timestamp} INFO  PERFORMANCES - \[%{GREEDYDATA:package}\]
%{WORD:method}\(\) completed successfully in %{NUMBER:duration:int} ms" }
    }
  }
  else {
    grok {
        match => { "message" => "\[%{TIMESTAMP_ISO8601:timestamp}\] \[%{NOTSPACE:wMCode}\] %{GREEDYDATA:textMsg}" }
    }
  }
  date {
    match => [ "timestamp", ISO8601 ]
    timezone => "Europe/Paris"
    target => "@timestamp"
  }
}

output {
    elasticsearch {
        hosts => [ "frpardge.corp.nvx.com:9200" ]
    }
    #stdout { codec => rubydebug }
}
```

12.9. Filebeat

Install this only if you have files to be parsed and sent to Elasticsearch

- Pull the image

```
docker pull docker.elastic.co/beats/filebeat:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/filebeat.yml
```

- Start the container

```
docker run --name filebeat -d -v /opt/sagis/IntegrationServer/instances/default/logs/:/islogs/ -v
~/elastic/filebeat.yml:/usr/share/filebeat/filebeat.yml docker.elastic.co/beats/filebeat:6.0.0
```

~/elastic/filebeat.yml

```
filebeat.prospectors:

- type: log
  paths:
    - /islogs/log4j2/perfs.log
  fields: {log_type: perflog}

- type: log
  paths:
    - /islogs/server.log
  multiline.pattern: '^\[20'
  multiline.negate: true
  multiline.match: after
  fields: {log_type: serverlog}

output.logstash:
  hosts: ["frpardge.corp.nvx.com:5043"]
```

12.10. Metricbeat

This chmod has to be done again after each VM reboot before starting Metricbeat :

```
sudo chmod 777 /var/run/docker.sock
```

- Pull the image

```
docker pull docker.elastic.co/beats/metricbeat:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/metricbeat.yml
```

- Start the container

```
docker run --name metricbeat -d -v /var/run/docker.sock:/var/run/docker.sock -v
~/elastic/metricbeat.yml:/usr/share/metricbeat/metricbeat.yml --volume=/proc:/hostfs/proc:ro
--volume=/sys/fs/cgroup:/hostfs/sys/fs/cgroup:ro --volume=/:/hostfs:ro --net=host
docker.elastic.co/beats/metricbeat:6.0.0 metricbeat -e -system.hostfs=/hostfs
```

To test you CPU graphs, with the proper handling of the cores, you can use stress application to load one or multiple cores :

```
sudo apt-get install stress
stress --cpu 2
```

~/elastic/metricbeat.yml

```
metricbeat.modules:
- module: system
  period: 10s
  metricsets:
    - cpu
    #- load
    - memory
    #- network
    - process
    - process_summary
    #- core
    #- diskio
    #- socket
  processes: ['.*']
  process.include_top_n:
    by_cpu: 10      # include top processes by CPU
    by_memory: 10   # include top processes by memory

- module: system
  period: 1m
  metricsets:
    - filesystem
    - fsstat
  processors:
  - drop_event.when.regexp:
      system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|hostfs|run|var)($|/)'

- module: docker
  metricsets:
    #- container
    - cpu
    #- diskio
    #- healthcheck
    #- image
    #- info
    - memory
    #- network
  hosts: ["unix:///var/run/docker.sock"]
  period: 10s

output.elasticsearch:
  hosts: ["frpardge.corp.nvx.com:9200"]

metricbeat.config.modules:
  path: /usr/share/metricbeat/metricbeat.yml
  reload.enabled: true
  reload.period: 60s
```

12.11. Grafana

```
wget https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_4.4.3_amd64.deb
```

```
sudo apt-get install -y adduser libfontconfig
```

```
sudo dpkg -i grafana_4.4.3_amd64.deb
```

To start it

```
sudo service grafana-server start
```

To auto start it at boot time

```
sudo update-rc.d grafana-server defaults
```

12.12. Jaeger Tracing (OpenZipkin-like)

To start it

```
docker run --restart=always --name jaeger -d -p5775:5775/udp -p6831:6831/udp -p5778:5778 -p16686:16686
jaegertracing/all-in-one:latest
```

12.13. Install WM servers

12.13.1. Prerequisites

Enable X11, ssh graphical

- On Windows machine
  - install Xming
  - configure putty session for X11 :
  - Connection > SSH > X11
  - check "ENable X11"
  - X display location = localhost:0.0
- On linux VM

```
apt-get install libfontconfig1 libxrender1
apt-get install libxtst6
apt-get install libxi6
```

Install Java

- https://tecadmin.net/install-oracle-java-8-ubuntu-via-ppa/

Setup hosts

```
vi /etc/hosts
```

```
127.0.0.1 dbhost umhost ishost
```

Prepare database

```
./dbConfigurator.sh -i -file wm-db-script.xml -dir . --password=WMUSER --admin_password=oracle
```

12.13.2. Install IS & UM manually

TODO