

# Power Week 2025

#pw2025

18 - 19 - 20 novembre 2025

IBM Innovation Studio Paris

S39 - Git, Jenkins, Nexus...entrent dans un IBM i

19 novembre 11:15 - 12:15

Sébastien JULLIAND

i and me – Training and Consulting GmbH

[sjulliand@mlitter.com](mailto:sjulliand@mlitter.com)

IBM

common  
FRANCE

# Qui suis-je?

- Sébastien “Seb” Julliand
- Membre de la *core team* Code for IBM i
- Créeur du plugin Jenkins IBM i Pipeline Steps
- IBM Champion 2024/2025
- R&D Tech Lead chez i and me - Training and consulting



# Point lexical - “*Pardon my french*”

- *Build*: le processus de construction de l'application
- *Artefact*: le livrable résultant du *build*
- *Repository*: le dépôt/référentiel de paquets logiciels
- *Versioning*: gestion des versions d'une application

# L'objectif

*Construire une fois, déployer plusieurs fois*

# L'objectif

- Builder l'application
  - Processus répétable
  - Dans un environnement contrôlé
  - En produisant un livrable (a.k.a. un artefact)
  - En un clic

# L'objectif

- Builder l'application
  - Processus répétable
  - Dans un environnement contrôlé
  - En produisant un livrable (a.k.a. un artefact)
  - En un clic
- Stocker l'artefact
  - Dans un repository unique
  - Traçabilité
  - Versioning (snapshot, release)
  - Sécurisé
  - Partagé par d'autres builds d'application?

# L'objectif

- Déployer les artefacts
  - Via un processus unique
  - Répétable
  - Traçable
  - Partout le même livrable

# Les outils

*Open-source, éprouvés et maintenus*

# Les outils

- Dépôt de code source
  - *GitLab*

# Les outils

- Dépôt de code source
  - *GitLab*
- Outil de build
  - *gmake*

# Les outils

- Dépôt de code source
  - *GitLab*
- Outil de build
  - *gmake*
- Serveur d'automatisation
  - *Jenkins (et quelques pipelines)*

# Les outils

- Dépôt de code source
  - *GitLab*
- Outil de build
  - *gmake*
- Serveur d'automatisation
  - *Jenkins (et quelques pipelines)*
- Dépôt d'artefacts
  - *Sonatype Nexus*

# Les outils

- Dépôt de code source
  - *GitLab*
- Outil de build
  - *gmake*
- Serveur d'automatisation
  - *Jenkins (et quelques pipelines)*
- Dépôt d'artefacts
  - *Sonatype Nexus*
- Et un peu de colle
  - *Maven*

# Les outils - quelques alternatives

- Dépôt de code source
  - *GitHub, GitHub, BitBucket, Azure, GitBucket (installable sur IBM i!)*
- Outil de build
  - *gmake, BOB for IBM i, script shell, CL, éditeurs tierces...*
- Serveur d'automatisation
  - *Jenkins, GitHub actions, GitLab CICD, ...*
- Dépôt d'artefacts
  - *Sonatype Nexus, JFrog Artifactory, dossier partagé (IFS, SSH, FTP...)*
- Et un peu de colle
  - *Maven, script shell*

# Jenkins

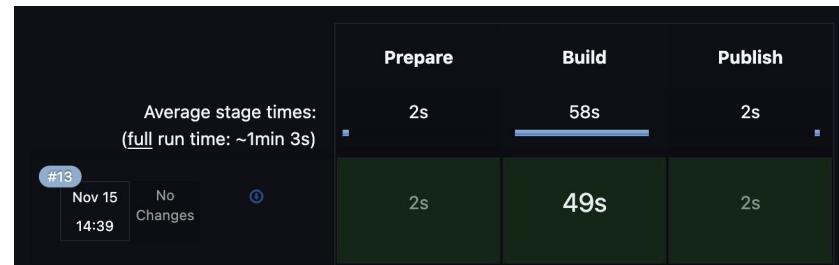
*Les pipelines et le plugin IBM i pipeline steps*

# Jenkins - les Pipelines

- Processus décrit *as code*
  - dans la configuration du job ou
  - dans le projet construit (Git, *Jenkinsfile*)

Pipeline script

```
1~ node {  
2~   stage('Prepare') {  
3~     dir('company_system') {  
4~       git branch: params.Branch, url: 'git@github.com:sebjulliand/company_system.git', credentialsId: 'github-credentials'  
5~     }  
6~   }  
7~  
8~   stage('Build') {  
9~     onIBMi(server:'PUB400') {  
10~       def buildLibrary = "SEBJUB";  
11~       def projectDirectory = "/home/sebju/builds/company_system";  
12~       ibmiCommand command: "RMDIR DIR('$projectDirectory') SUBTREE(*ALL)", failOnError: false  
13~       ibmiPutIFS from: 'company_system', to: projectDirectory  
14~       ibmiCommand command: "CLRLIB LIB($buildLibrary)"  
15~  
16~       def build = ibmiShellExec "cd $projectDirectory; chmod +x ./build.sh; ./build.sh $buildLibrary"  
17~       writeFile encoding: "UTF-8", file: "build.txt", text: build.output()  
18~     }  
19~   }  
20~ }
```



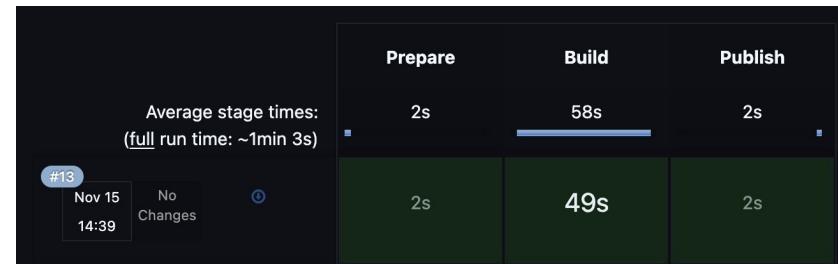
# Jenkins - les Pipelines

- Processus décrit *as code*
  - dans la configuration du job ou
  - dans le projet construit (Git, *Jenkinsfile*)

Pipeline script

```
1~ node {  
2~   stage('Prepare') {  
3~     dir('company_system') {  
4~       git branch: params.Branch, url: 'git@github.com:sebjulliand/company_system.git', credentialsId: 'github-credentials'  
5~     }  
6~   }  
7~  
8~   stage('Build') {  
9~     onIBMi(server:'PUB400') {  
10~       def buildLibrary = "SEBJUB";  
11~       def projectDirectory = "/home/sebju/builds/company_system";  
12~       ibmiCommand command: "RMDIR DIR('$projectDirectory') SUBTREE(*ALL)", failOnError: false  
13~       ibmiPutIFS from: 'company_system', to: projectDirectory  
14~       ibmiCommand command: "CLRLIB LIB($buildLibrary)"  
15~  
16~       def build = ibmiShellExec "cd $projectDirectory; chmod +x ./build.sh; ./build.sh $buildLibrary"  
17~       writeFile encoding: "UTF-8", file: "build.txt", text: build.output()  
18~     }  
19~   }  
20~ }
```

- Décrit la logique du build, des tests, du déploiement



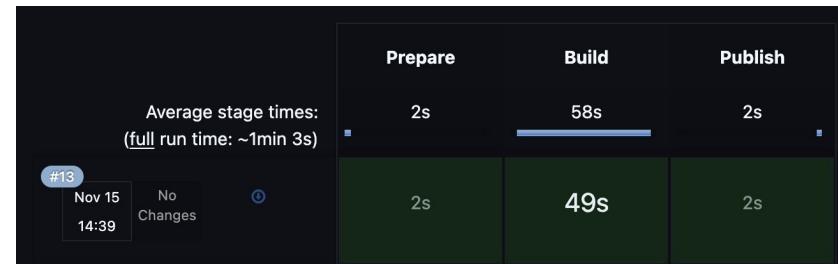
# Jenkins - les Pipelines

- Processus décrit *as code*
  - dans la configuration du job ou
  - dans le projet construit (Git, *Jenkinsfile*)

Pipeline script

```
1~ node {  
2~   stage('Prepare') {  
3~     dir('company_system') {  
4~       git branch: params.Branch, url: 'git@github.com:sebjulliand/company_system.git', credentialsId: 'github-credentials'  
5~     }  
6~   }  
7~  
8~   stage('Build') {  
9~     onIBMi(server:'PUB400') {  
10~       def buildLibrary = "SEBJUB";  
11~       def projectDirectory = "/home/sebju/builds/company_system";  
12~       ibmiCommand command: "RMDIR DIR('$projectDirectory') SUBTREE(*ALL)", failOnError: false  
13~       ibmiPutIFS from: 'company_system', to: projectDirectory  
14~       ibmiCommand command: "CLRLIB LIB($buildLibrary)"  
15~  
16~       def build = ibmiShellExec "cd $projectDirectory; chmod +x ./build.sh; ./build.sh $buildLibrary"  
17~       writeFile encoding: "UTF-8", file: "build.txt", text: build.output()  
18~     }  
19~   }  
20~ }
```

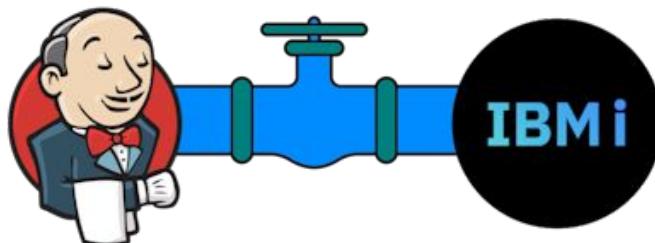
- Décrit la logique du build, des tests, du déploiement
- Découpage simple
  - stages
  - steps



# Jenkins - IBM i Pipeline Steps

- Plugin open source
  - Disponible sur le dépôt public jenkins.io
  - <https://plugins.jenkins.io/ibmi-steps>

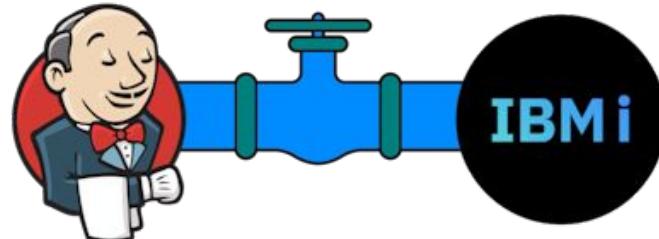
- Steps
  - onIBMi
  - ibmiCommand
  - ibmiShellExec
  - ibmiRunSQL
  - ibmiGetSAVF
  - ibmiPutSAVF
  - ibmiGetIFS
  - ibmiPutIFS
  - ibmiGetSPLF
  - ibmiWaitJob



# Jenkins - IBM i Pipeline Steps

- Plugin open source
  - Disponible sur le dépôt public jenkins.io
  - <https://plugins.jenkins.io/ibmi-steps>
- Fournit des steps dédiés à IBM i
  - exécution CL
  - exécution shell
  - exécution SQL
  - transferts IFS
  - transferts SAVF

- Steps
  - onIBMi
  - ibmiCommand
  - ibmiShellExec
  - ibmiRunSQL
  - ibmiGetSAVF
  - ibmiPutSAVF
  - ibmiGetIFS
  - ibmiPutIFS
  - ibmiGetSPLF
  - ibmiWaitJob



# Sonatype Nexus Repository

*Le dépôt d'artefacts*

# Sonatype Nexus Repository

- Gestionnaire de dépôts
  - artefacts
  - paquets
  - métadonnées
  - ...

# Sonatype Nexus Repository

- Gestionnaire de dépôts
  - artefacts
  - paquets
  - métadonnées
  - ...
- Version Community open-source et gratuite

# Apache Maven

*Mais pourquoi ?!* 

# Apache Maven

- Dédié à Java... à la base

# Apache Maven

- Dédié à Java... à la base
- Utilisable dans d'autres scénarios !

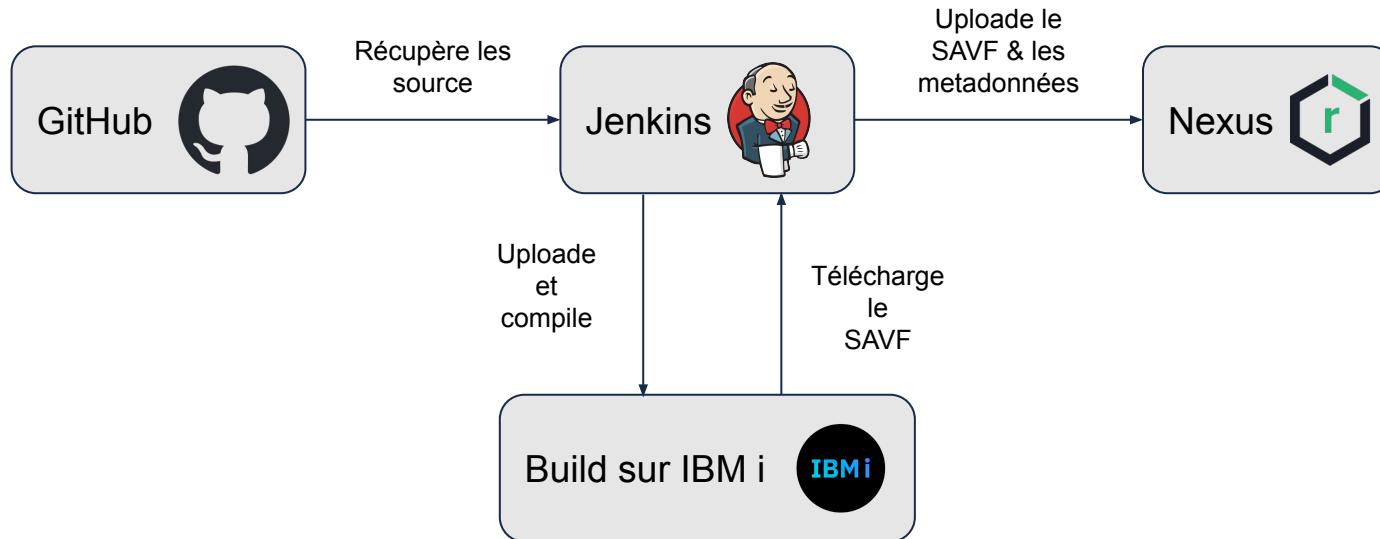
# Apache Maven

- Dédié à Java... à la base
- Utilisable dans d'autres scénarios !
  - Via de nombreux plugins
- Dans notre cas
  - Gestion agnostique des transferts d'artefacts
  - Versioning/estampillage
  - Possible gestion des dépendances inter-application

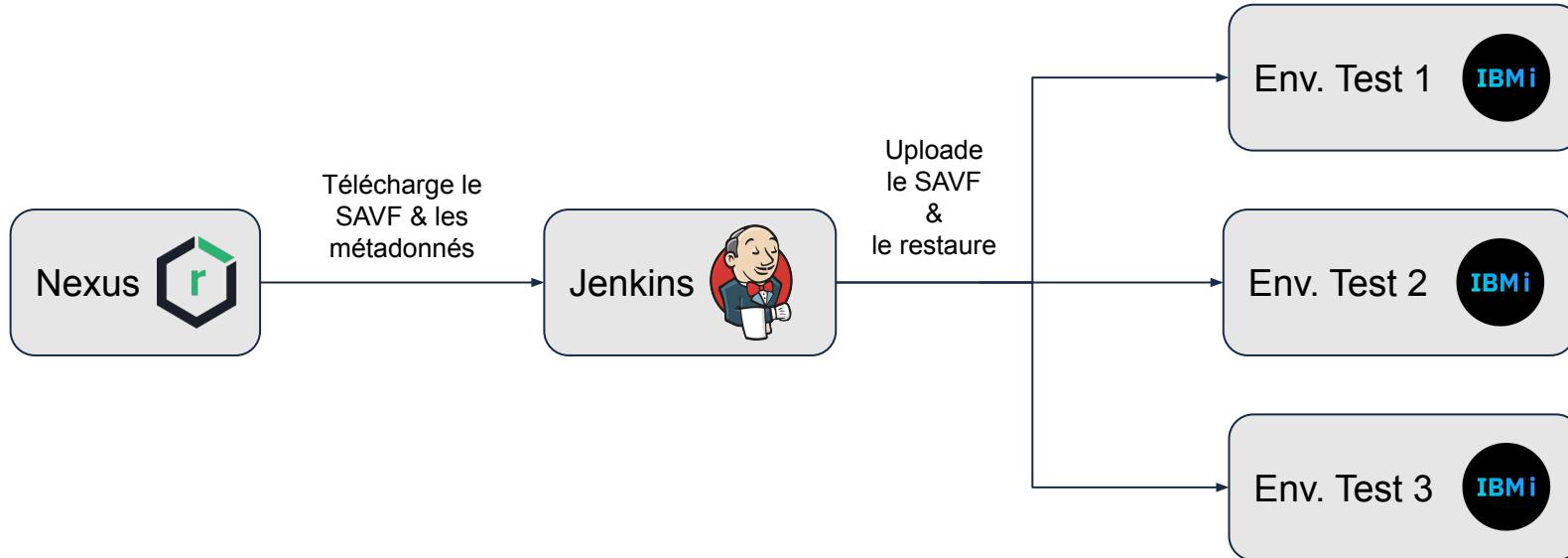
# Le processus

*En image* 

# Construire une fois...



# ...déployer plusieurs fois



# Le processus

*En action* 

# Des questions?

Ne soyez pas timide!



# En résumé

- 1 clic pour builder
- 1 clic pour déployer
- Facile à monitorer
- Flexible
  - C'est un grand Lego !
  - Briques remplaçables (excepté IBM i)

# Power Week

18 -19 - 20 novembre  
2025

[sebjulliand@gmail.com](mailto:sebjulliand@gmail.com)

[sjulliand@mlitters.com](mailto:sjulliand@mlitters.com)

**IBM**  
common  
FRANCE

IBM



Meet me for Coffee