

Lab 4 : Automatisation du flux de travail: publication d'artefacts dans le référentiel Nexus à l'aide des pipelines Jenkins

Dans ce guide, nous utiliserons Jenkins comme serveur d'intégration continue et Nexus Repository comme référentiel de build.

L'objectif de ce guide est de créer un flux de travail dans lequel nous pouvons créer, stocker, organiser et surveiller les artefacts compilés par Maven via notre serveur CI.

Avant de commencer, assurez-vous que Jenkins est déjà opérationnel. Si vous n'avez pas encore configuré Jenkins, copiez la commande ci-dessous et exécutez-la sur votre hôte/machine compatible Docker.

```
# docker status jenkins
```

Installation du référentiel Nexus

Nexus est un gestionnaire de référentiels qui vous permet de stocker et de récupérer des artefacts. Il vous permet d'héberger vos artefacts créés dans un dépôt privé et sécurisé.

Vous pouvez toujours extraire l'image Nexus Docker à l'aide de la commande suivante:

```
# docker pull sonatype/nexus3
Using default tag: latest
latest: Pulling from sonatype/nexus3
cb3c77f9bdd8: Pull complete
fd8daf2668d1: Pull complete
fd1ff82b00e8: Pull complete
2a05f7b573af: Pull complete
Digest:
sha256:6570855dfbc3eb094fe5cbbacec87aa8b91d16394dab627177e1deeebb5ac8ee
Status: Downloaded newer image for sonatype/nexus3:latest
docker.io/sonatype/nexus3:latest
```

Il est maintenant temps d'exécuter sonatype / nexus téléchargé sur le port par défaut 8081. Suivez ces commandes:

```
$ docker run -d --name nexus_repo -p 8081:8081 sonatype/nexus3
```

Le lancement du service Nexus dans votre conteneur Docker nouvellement créé prend généralement entre 1 et 2 minutes. Si vous souhaitez suivre le journal pour voir si Nexus est opérationnel et prêt, exécutez la commande suivante:

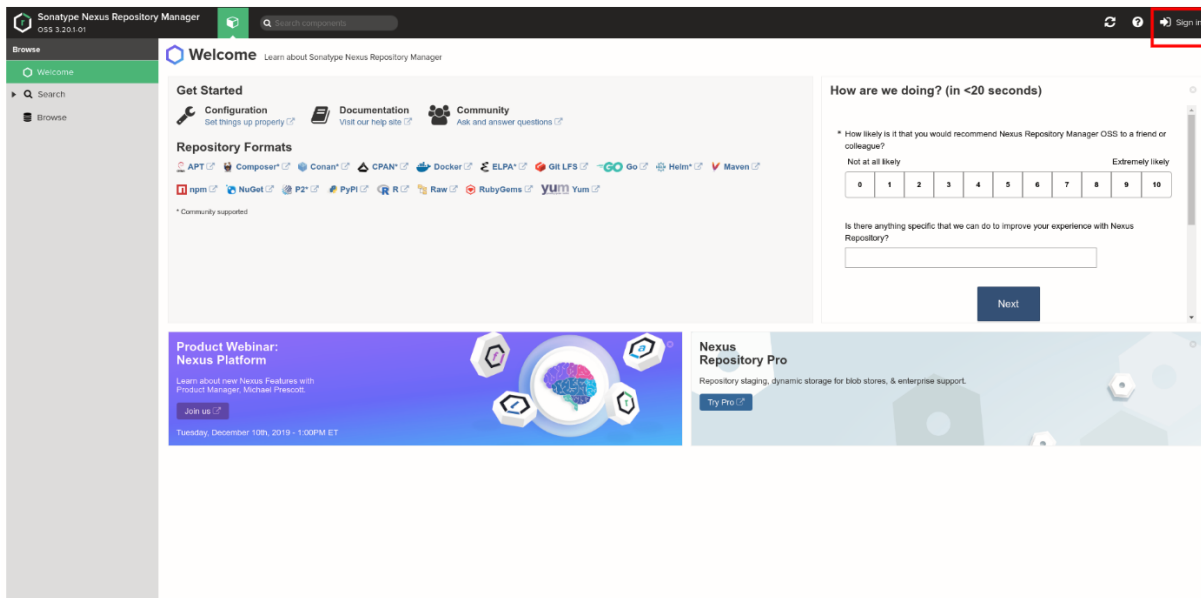
```
$ docker logs -f nexus_repo
```

Dans les journaux, vous verrez un message indiquant Started Sonatype Nexus OSS 3.20.1-01

Cela signifie que votre Nexus Repository Manager est prêt à être utilisé. Maintenant, allez dans votre navigateur et ouvrez

<http://your-ip-addr:8081>

Trouvez l'option *Connexion* comme indiqué ci - dessous:



Le nom d'utilisateur par défaut est **admin**, alors que pour récupérer le mot de passe, vous devez exécuter la commande suivante:

```
$ docker exec -i nexus_repo cat /nexus-data/admin.password
502ace93-5450-4f0d-97d2-9b3b3a88d149
```

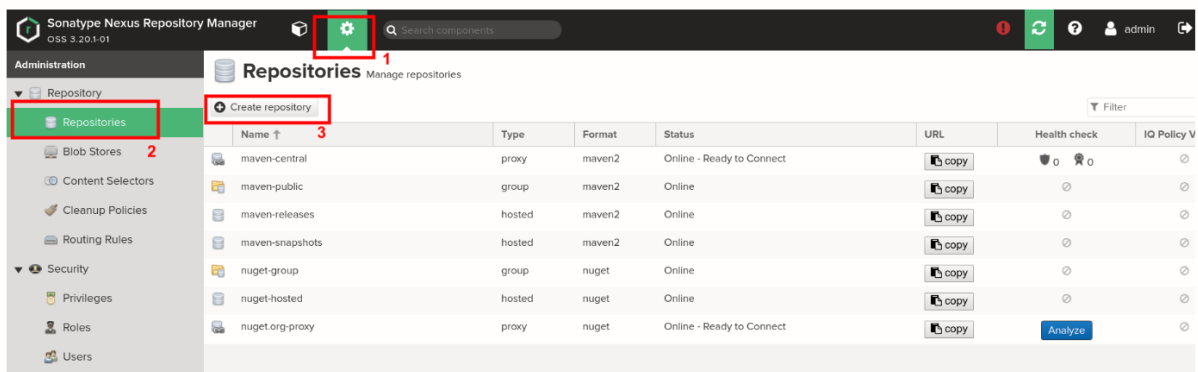
Nexus Repository Manager est prêt à être utilisé. L'étape suivante consiste à créer un nouveau référentiel.

Créer un référentiel dans Nexus

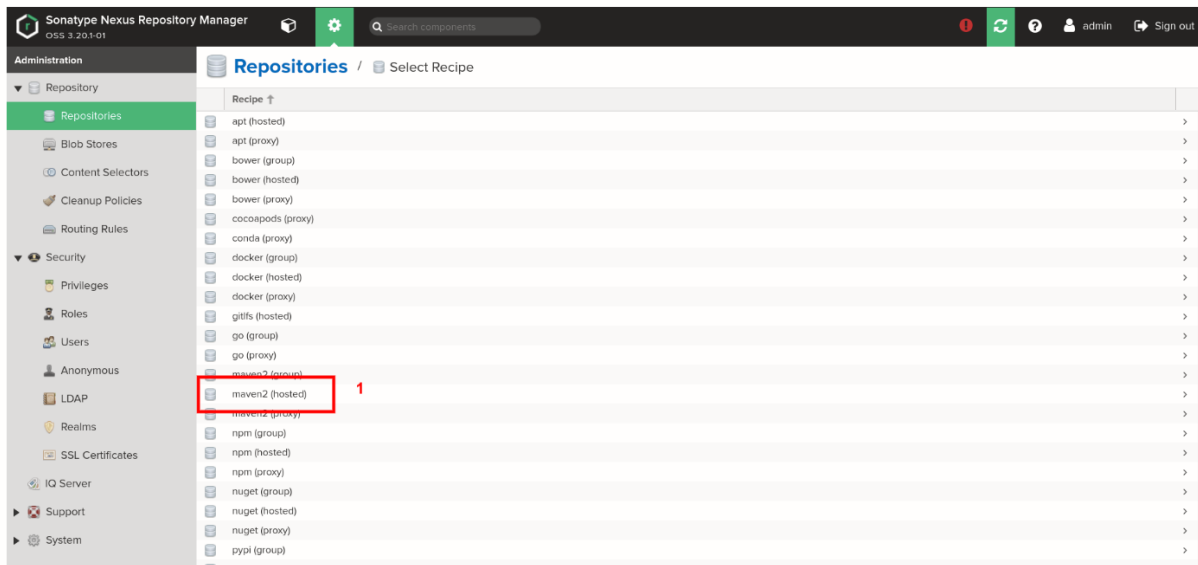
Dans cette étape, vous allez créer un référentiel hébergé Maven dans Nexus, où votre Jenkins va télécharger des artefacts de «construction».

Étape 1:

Suivez les étapes ci-dessous pour créer un référentiel hébergé, nommez-le `maven-nexus-repo`, que vous allez utiliser tout au long de ce Lab.



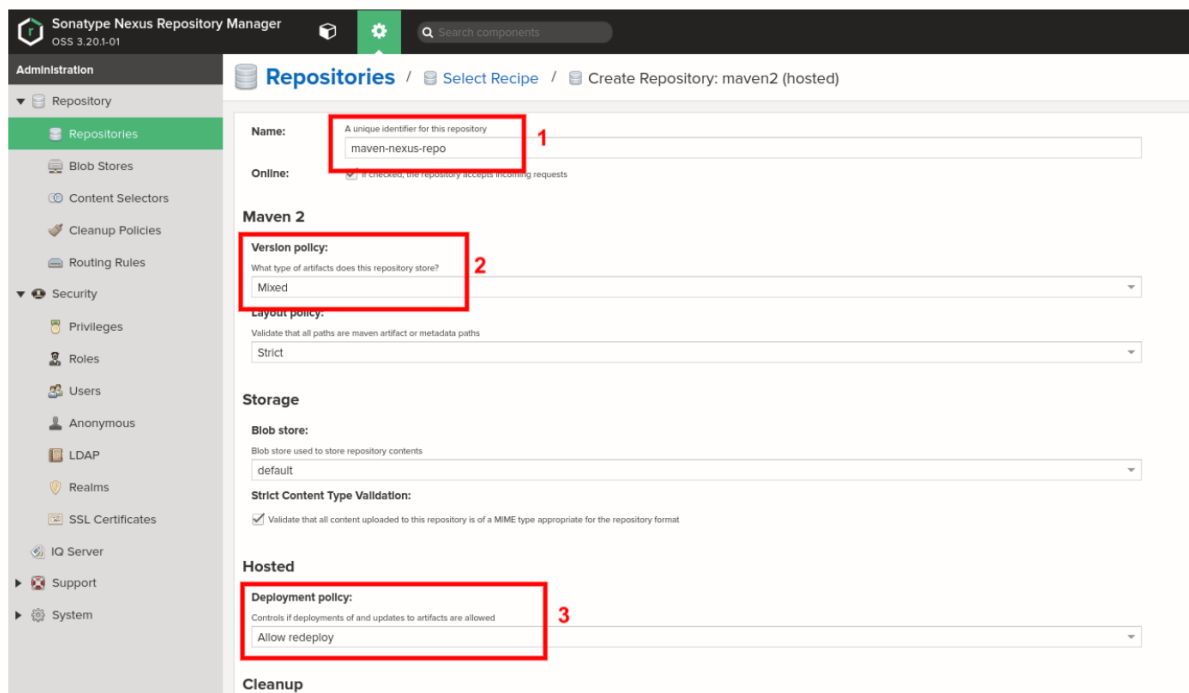
Sélectionnez la *recette maven2(hosted)* dans la liste comme indiqué ci-dessous:



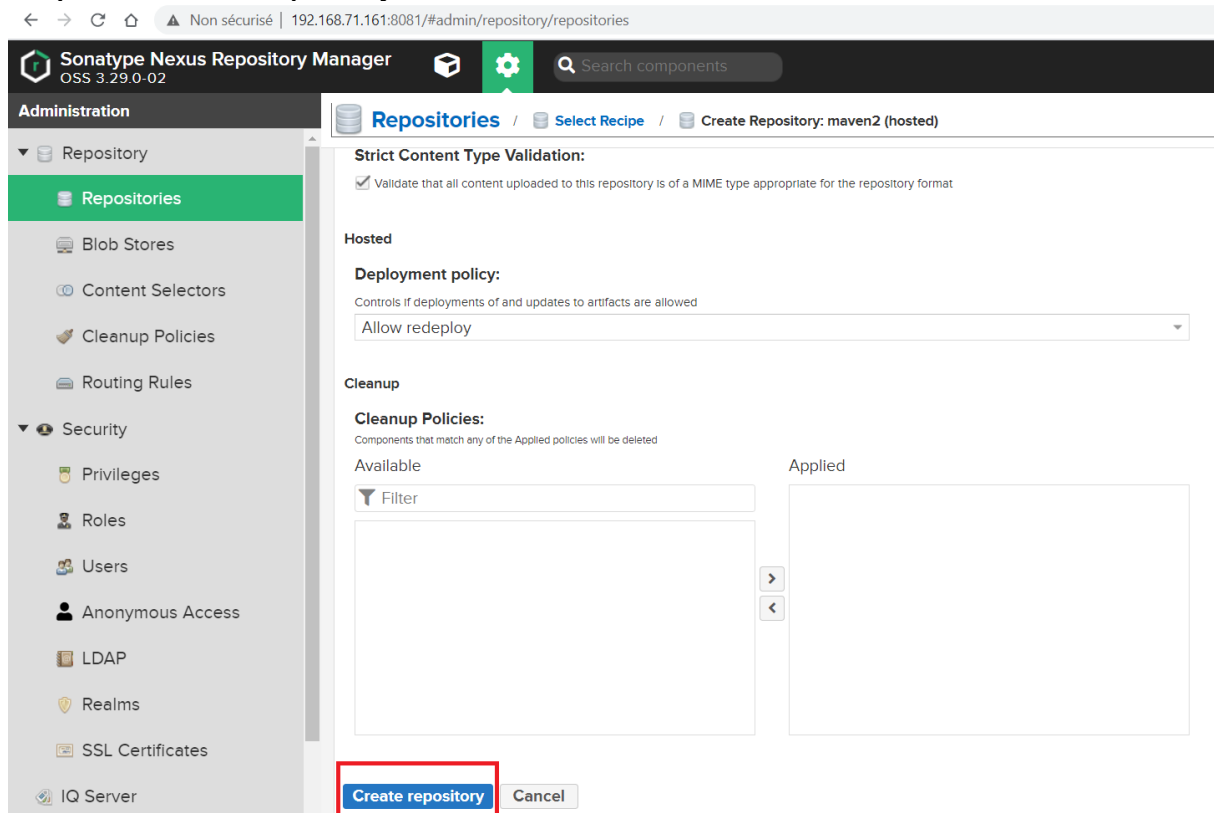
Étape 2:

Sur la page *Créer un référentiel*

- Entrez le nom comme *maven-nexus-repo*
- Dans la stratégie de version, sélectionnez le type d'artefacts *Mixed*.
- Dans la section *Hosted*, dans la stratégie de déploiement, sélectionnez *Allow Redeploy*. Cela vous permettra de déployer une application plusieurs fois.

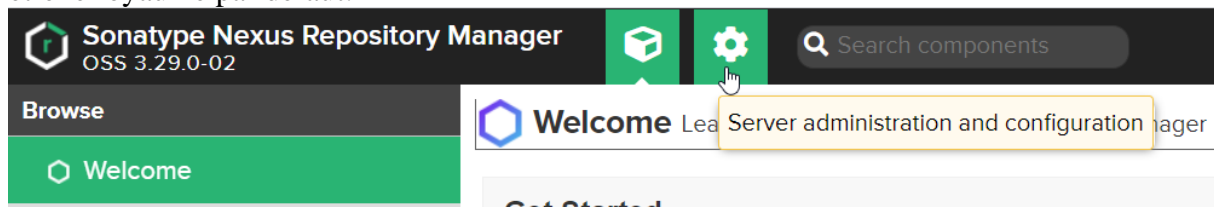


Cliquez sur Create Repository.



Étape 3:

Pour créer un nouvel utilisateur, accédez à *Dashboard > Server Administrator and Configuration > Users > Create user*. Sélectionnez le type d'utilisateur *local* qui se trouve être le royaume par défaut:



Sonatype Nexus Repository Manager
OSS 3.20.1-01

Administration

- Repository
 - Repositories
 - Blob Stores
 - Content Selectors
 - Cleanup Policies
 - Routing Rules
- Security
 - Privileges
 - Roles
 - Users**
 - Anonymous
 - LDAP
 - Realms
 - SSL Certificates
- Support
- System

Users / Create User

ID:
This will be used as the username
jenkins-user

First name:
Jenkins

Last name:
User

Email:
Used for notifications
username@example.com

Password:

Confirm password:

Status:
Active

Roles:

Available

Filter

nx-anonymous

Granted

nx-admin

Create local user Cancel

Dans la page *Créer un utilisateur*

1. **ID** : saisissez l'ID souhaité; dans notre cas, il s'agit de *jenkins-user*.
2. **Prénom** : Saisissez le prénom souhaité; dans notre cas, c'est *Jenkins*.
3. **Nom de famille** : entrez le deuxième nom souhaité; dans notre cas, il s'agit de *User*.
4. **E-mail** : saisissez votre adresse e-mail.
5. **Statut** : sélectionnez *Actif* dans votre menu déroulant.
6. **Rôles** : assurez-vous que vous accordez le rôle *nx-admin* à votre utilisateur.

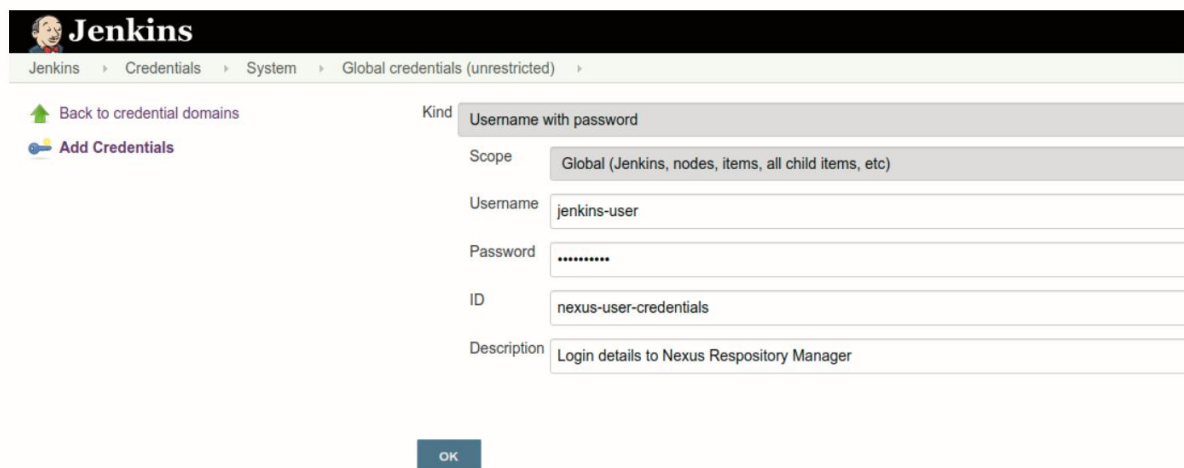
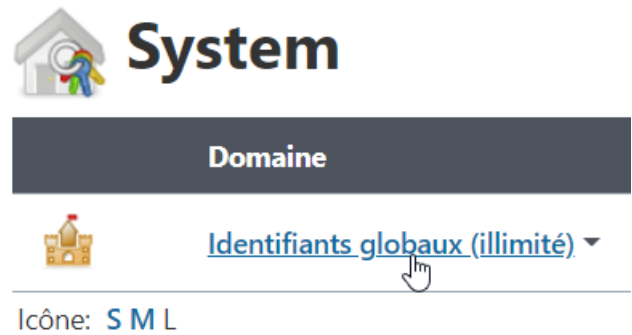
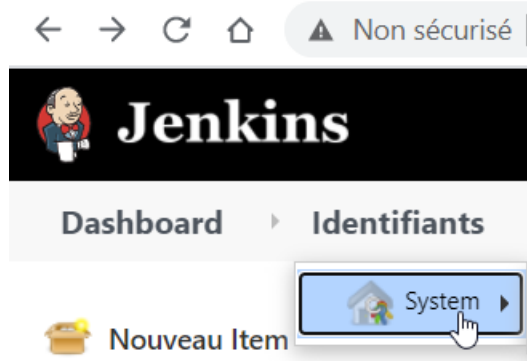
Avec cela, nous en avons terminé avec la partie configuration de Nexus Repository Manager. Passons à Jenkins pour y installer Nexus.

Installer et configurer les plugins Nexus dans Jenkins

Ici, vous allez installer et configurer quelques plugins pour Nexus dans Jenkins. Pour cela, accédez à Jenkins, puis à *Tableau de bord* > *Gérer Jenkins* > *Gérer les plugins* > *Disponible* et recherchez et installez **Nexus Artifact Uploader** et **Pipeline Utility Steps**.

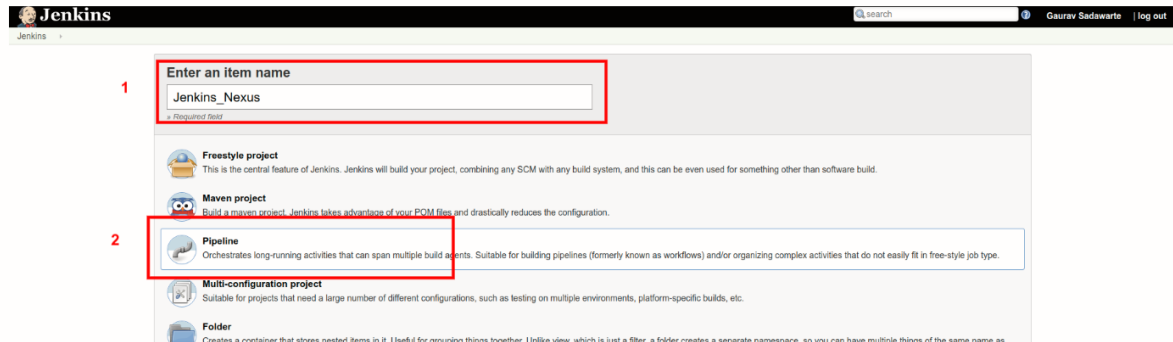
Ajoutez les informations d'identification de l'utilisateur de Nexus Repository Manager dans Jenkins.

Accédez à *Administrer Jenkins* > *Manage Credentials* > *Système* > *Identifiants globaux (illimité)*, comme indiqué ci-dessous:



Créer un pipeline Jenkins

Il est temps de créer un Job Jenkins. Ici, vous allez utiliser le type de travail Pipeline, nommé *JenkinsNexus*, comme indiqué ci-dessous:



Dans la page suivante, recherchez la section Pipeline et copiez le script ci-dessous dans la zone de texte:

```
pipeline {
  agent {
    label "master"
  }
  tools {
    maven "Maven 3.6.3"
  }
  environment {
    NEXUS_VERSION = "nexus3"
    NEXUS_PROTOCOL = "http"
    NEXUS_URL = "192.168.71.161:8081"
    NEXUS_REPOSITORY = "maven-nexus-repo"
    NEXUS_CREDENTIAL_ID = "nexus-user-credentials"
  }
  stages {
    stage("Clone code from VCS") {
      steps {
        script {
          git 'https://github.com/eliesjebri/demo';
        }
      }
    }
    stage("Maven Build") {
      steps {
        script {
          sh "mvn package -DskipTests=true"
        }
      }
    }
    stage("Publish to Nexus Repository Manager") {
      steps {
        script {
          pom = readMavenPom file: "pom.xml";
          filesByGlob = findFiles(glob: "target/*.${pom.packaging}");
          echo "${filesByGlob[0].name} ${filesByGlob[0].path}
${filesByGlob[0].directory} ${filesByGlob[0].length} ${filesByGlob[0].lastModified}"
          artifactPath = filesByGlob[0].path;
          artifactExists = fileExists artifactPath;
          if(artifactExists) {
            echo "*** File: ${artifactPath}, group: ${pom.groupId},
packaging: ${pom.packaging}, version ${pom.version}";
            nexusArtifactUploader(
              nexusVersion: NEXUS_VERSION,
```



```

Repository:maven-nexus-repo
Downloading: http://192.168.71.161:8081/repository/maven-nexus-repo/CMADSession/CMADSession/0.0.21-SNAPSHOT/maven-metadata.xml
Uploading: http://192.168.71.161:8081/repository/maven-nexus-repo/CMADSession/CMADSession/0.0.21-SNAPSHOT/CMADSession-0.0.21-20201222.165039-1.war
100 % completed (3.0 kB / 3.0 kB).
Uploaded: http://192.168.71.161:8081/repository/maven-nexus-repo/CMADSession/CMADSession/0.0.21-SNAPSHOT/CMADSession-0.0.21-20201222.165039-1.war (3.0 kB at 7.6
kB/s)
Uploading: http://192.168.71.161:8081/repository/maven-nexus-repo/CMADSession/CMADSession/0.0.21-SNAPSHOT/CMADSession-0.0.21-20201222.165039-1.pom
Uploaded: http://192.168.71.161:8081/repository/maven-nexus-repo/CMADSession/CMADSession/0.0.21-SNAPSHOT/CMADSession-0.0.21-20201222.165039-1.pom (956 B at 6.4 kB/s)
Uploading: http://192.168.71.161:8081/repository/maven-nexus-repo/CMADSession/CMADSession/0.0.21-SNAPSHOT/maven-metadata.xml
Uploaded: http://192.168.71.161:8081/repository/maven-nexus-repo/CMADSession/CMADSession/0.0.21-SNAPSHOT/maven-metadata.xml (775 B at 8.0 kB/s)
Uploading artifact CMADSession-0.0.23.war completed.
Uploading artifact pom.xml completed.
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Alors que, dans votre Nexus Repository Manager, vous verriez quelque chose de similaire à ceci:

The screenshot displays the Sonatype Nexus Repository Manager web interface. The left sidebar shows navigation options: Welcome, Search, Browse, and Upload. The main content area is titled 'Browse' and shows a tree view of the repository structure. The path is: `net` > `java` > `cargo-tracker` > `1.0-SNAPSHOT`. Under `1.0-SNAPSHOT`, there are several files listed, including `cargo-tracker-1.0-20200129.100842-1.war`, `cargo-tracker-1.0-20200129.100842-1.war.md5`, `cargo-tracker-1.0-20200129.100842-1.war.sha1`, `cargo-tracker-1.0-20200129.100844-2`, `cargo-tracker-1.0-20200129.100844-2.pom`, `cargo-tracker-1.0-20200129.100844-2.pom.md5`, `cargo-tracker-1.0-20200129.100844-2.pom.sha1`, `maven-metadata.xml`, `maven-metadata.xml.md5`, and `maven-metadata.xml.sha1`. The `maven-metadata.xml` file is selected, and its details are shown on the right. The details include a 'Summary' section with fields like Repository, Format, Path, Content type, File size, Blob created, Blob updated, Last downloaded, Locally cached, Blob reference, Containing repo, Uploader, and Uploader's IP Address. Below the summary is an 'Attributes' section with fields like Checksum, sha1, md5, Content, last_modified, and Maven2.

Conclusion

Une manière systématique de distribuer les artefacts du projet est essentielle pour toute organisation. Avec l'aide de Jenkins Pipeline et Nexus Repository Manager, vous pouvez centraliser le dépôt d'artefacts, ce qui réduit en fin de compte les efforts de reproduction du temps de construction et de changement d'outils CI sans vous soucier de la migration de nos artefacts.

Nexus peut également être configuré avec des services de stockage dans le cloud tels que AWS S3 et Google Cloud Storage, ce qui vous offre une liberté supplémentaire et une application de livraison sans aucun problème de stockage.