

Nexus



Bureau E204

Plan du cours

- Introduction
- Nexus
- Release vs Snapshot
- Installation Nexus
- Configuration Nexus en standalone
- Configuration Nexus avec Jenkins
- Travail à faire

Introduction

- Une fois les différents modules de l'application implémentés et testés convenablement en respectant les bonnes pratiques de développement, il est temps de déployer l'application dans les environnement adéquats.
- L'objectif dans notre chaine DevOps est **d'automatiser la gestion des livrables et l'alimentation automatique des différents environnements** par les artéfacts correspondants.

Nexus - Définition

- Nexus est une plateforme de gestion de dépôts (repository management tool), permettant d'héberger des artefacts.
- Ces artefacts sont des composants (jar, war, ...), générés au build d'un projet, et déposés ensuite sur Nexus grâce à l'outil Maven.
- Cet outil a une forte dépendance envers Maven.

Nexus - Définition

- L'intérêt de Nexus est de pouvoir y partager des artéfacts avec les autres développeurs d'un projet, ou entre plusieurs environnements (Docker).
- Nexus est développé par Sonatype, en Java (il faut une JRE pour exécuter Nexus, c'est déjà installé sur votre VM).
- Il y a une version gratuite (Community) et une version payante.

Nexus - Dépôts



Browse Browse assets and components

Filter

	Name ↑	Type	Format	Status	URL	Health check	IQ Policy VL...
	maven-central	proxy	maven2	Online - Ready to Connect	copy	Analyze	Loading...
	maven-public	group	maven2	Online	copy	⊘	Loading...
	maven-releases	hosted	maven2	Online	copy	⊘	Loading...
	maven-snapshots	hosted	maven2	Online	copy	⊘	Loading...
	nuget-group	group	nuget	Online	copy	⊘	Loading...
	nuget-hosted	hosted	nuget	Online	copy	⊘	Loading...
	nuget.org-proxy	proxy	nuget	Online - Ready to Connect	copy	Analyze	Loading...

Nexus - Dépôts

Hosted : Les dépôts hébergés par Nexus contenant les dépôts créés par les utilisateurs.

Nexus est configuré par défaut par les dépôts hosted suivants :

- 3rd Party : des librairies non présentes dans les dépôts maven publics.
- Releases : les librairies (artéfacts et autres) stables de l'organisation.
- Snapshots : les librairies (artéfacts et autres) en cours de développement de l'organisation.

Nexus - Dépôts

Proxy : Dépôts dont le serveur Nexus est seulement un relais d'un repository distant. Le proxy permet d'accélérer le processus de build tout en évitant les téléchargements inutiles sur internet.

Group : Un regroupement de dépôts sous une même URL afin d'alléger la configuration.

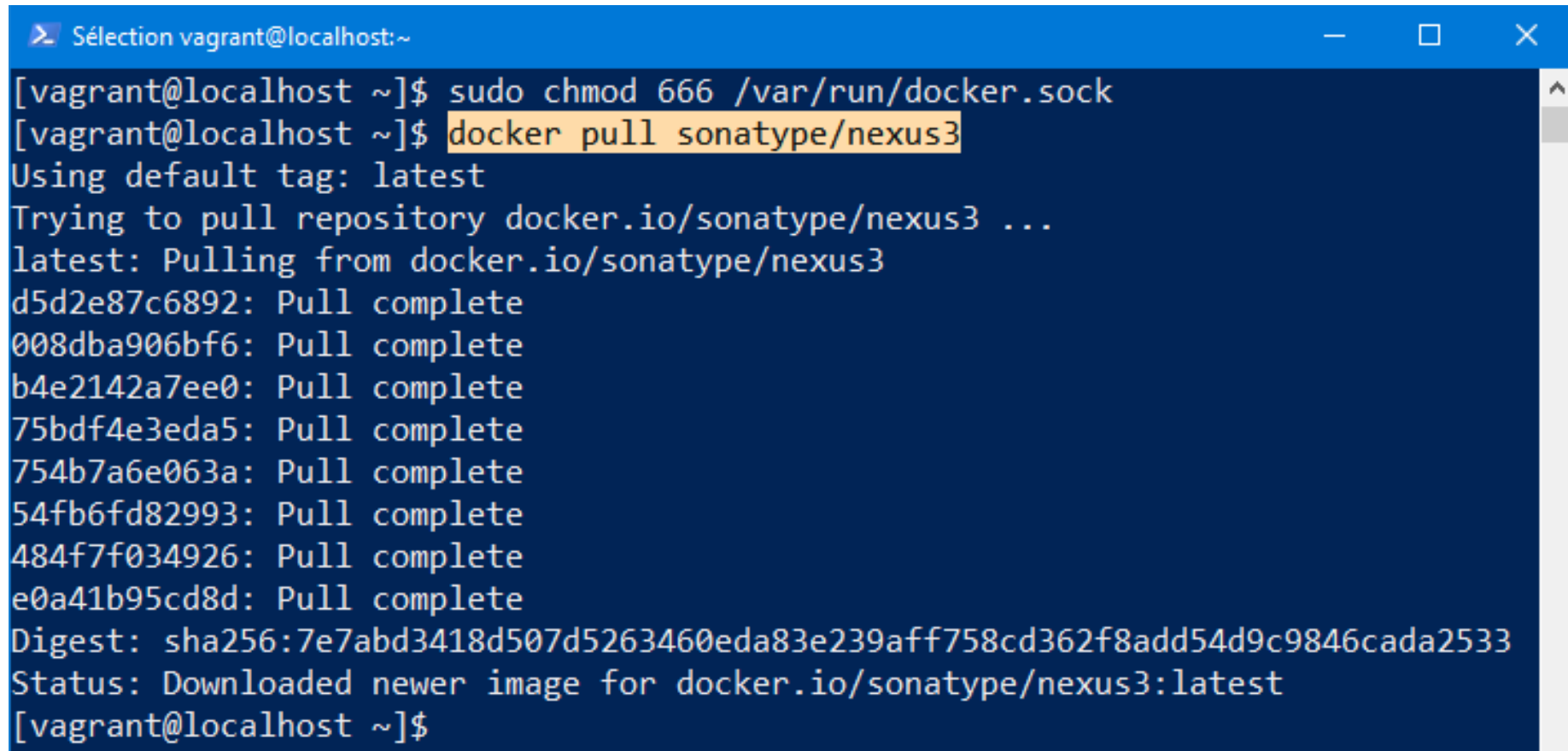
Nous allons utiliser les dépôts « hosted » pour nos projets.

Release vs Snapshot

- Une release est une **version fixe** d'un projet.
- Elle correspond à une version regroupant l'ensemble des fonctionnalités requises d'une itération (Sprint en cas de Scrum).
- Une version snapshot est une version **en cours de développement** avec une partie des fonctionnalités à implémenter.
- On peut avoir plusieurs snapshots pour la même version d'un projet pour de multiples utilisations (livraison à l'équipe de test, implémentation d'une solution en cours de validation , etc...).

Installation Nexus

- Nous allons utiliser une image Docker.
- Pour télécharger une image **nexus3**, vous devez exécuter la commande suivante: **docker pull sonatype/nexus3** (docker doit être up, faites un chmod avant) :

A terminal window titled 'Sélection vagrant@localhost:~' with standard window controls. It shows the execution of 'sudo chmod 666 /var/run/docker.sock' and 'docker pull sonatype/nexus3'. The output shows the image being pulled from docker.io, listing layer digests and their pull status, and finally confirming the download of the latest image.

```
[vagrant@localhost ~]$ sudo chmod 666 /var/run/docker.sock
[vagrant@localhost ~]$ docker pull sonatype/nexus3
Using default tag: latest
Trying to pull repository docker.io/sonatype/nexus3 ...
latest: Pulling from docker.io/sonatype/nexus3
d5d2e87c6892: Pull complete
008dba906bf6: Pull complete
b4e2142a7ee0: Pull complete
75bdf4e3eda5: Pull complete
754b7a6e063a: Pull complete
54fb6fd82993: Pull complete
484f7f034926: Pull complete
e0a41b95cd8d: Pull complete
Digest: sha256:7e7abd3418d507d5263460eda83e239aff758cd362f8add54d9c9846cada2533
Status: Downloaded newer image for docker.io/sonatype/nexus3:latest
[vagrant@localhost ~]$
```

Installation Nexus

- Pour créer le conteneur:

docker run -d -p 8081:8081 --name nexus sonatype/nexus3

- Pour vérifier que le conteneur est fonctionnel: **docker ps**

```
Sélection vagrant@localhost: ~
[vagrant@localhost ~]$ docker images
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
docker.io/sonatype/nexus3  latest      163907a855f2  11 days ago   556 MB
mouradhassini/alpine-5sim3  1.0.0       885ffcad896b  2 weeks ago   275 MB
mouradhassini/alpine      1.0.0       885ffcad896b  2 weeks ago   275 MB
docker.io/alpine        latest      9c6f07244728  8 weeks ago   5.54 MB
docker.io/sonarqube      8.9.7-community  8c30eec2357f  6 months ago  495 MB
docker.io/hello-world    latest      feb5d9fea6a5  12 months ago 13.3 kB
docker.io/centos         latest      5d0da3dc9764  12 months ago 231 MB
docker.io/nginx          1.15.12     53f3fd8007f7  3 years ago   109 MB
[vagrant@localhost ~]$ docker run -d -p 8081:8081 --name nexus sonatype/nexus3
18eefef10a943a58e02f314afc74042ae965e3f2f23eac3566bc726878d04b5b
[vagrant@localhost ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
18eefef10a94   sonatype/nexus3  "/opt/sonatype/nex..."  10 seconds ago  Up 8 seconds  0.0.0.0:8081->8081/tcp    nexus
[vagrant@localhost ~]$
```

Installation Nexus

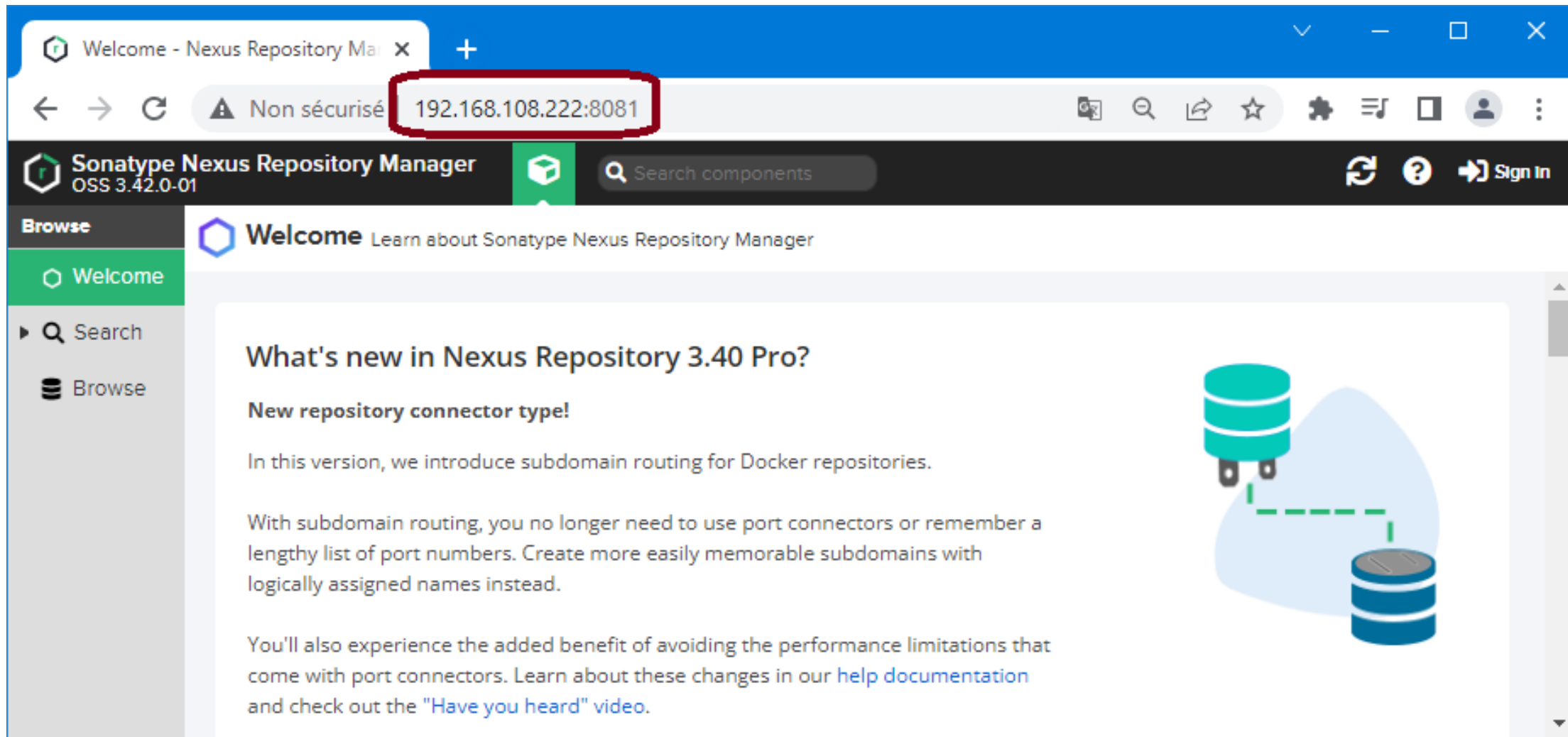
Pour accéder à Nexus, récupérer l'adresse ip de la VM : **ip addr show**

Sélection vagrant@localhost:~

```
[vagrant@localhost ~]$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:4d:77:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic eth0
        valid_lft 85644sec preferred_lft 85644sec
    inet6 fe80::5054:ff:fe4d:77d3/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:85:53:c8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.108.222/24 brd 192.168.108.255 scope global noprefixroute eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe85:53c8/64 scope link
        valid_lft forever preferred_lft forever
4: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:3a:2d:91:66 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:3aff:fe2d:9166/64 scope link
        valid_lft forever preferred_lft forever
6: vethb718d92@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether 22:66:5f:0c:3e:06 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::2066:5fff:fe0c:3e06/64 scope link
        valid_lft forever preferred_lft forever
[vagrant@localhost ~]$
```

Configuration Nexus

En local (Windows), sur un navigateur, aller à **http://<ip_vm>:8081**



Configuration Nexus

- Connecter avec les paramètres suivants:
 - Username : admin
 - Password : Accéder au conteneur (avec **docker exec -i idconteneur**), et lire le fichier mentionné dans l'interface d'authentification (avec **cat**) :

```
Sélection vagrant@localhost:~  
[vagrant@localhost ~]$ docker exec -i 18eefef10a94 cat /nexus-data/admin.password  
6f808d07-59f2-48fc-b35b-984bdcdbd6a3[vagrant@localhost ~]$
```

Sign In

Your **admin** user password is located in **/nexus-data/admin.password** on the server.

Sign In

Cancel

Configuration Nexus

- Changer le mot de passe à **nexus** par exemple, et accepter les accès anonymes.

Setup 1 of 4

This wizard will help you complete required setup tasks.

Next

Please choose a password for the admin user 2 of 4

New password:

! This field is required

Confirm password:

Back Next

Configuration Nexus

Configure Anonymous Access

3 of 4

Enable anonymous access means that by default, users can search, browse and download components from repositories without credentials. Please **consider the security implications for your organization**.

Disable anonymous access should be chosen with care, as it **will require credentials for all** users and/or build tools.

[More information](#)

- ☒ Enable anonymous access
- ☐ Disable anonymous access

Back

Next

Complete

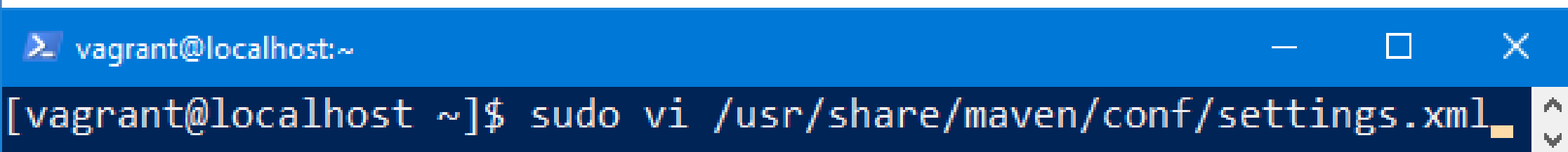
4 of 4

The setup tasks have been completed, enjoy using Nexus Repository Manager!

Finish

Configuration Nexus : settings.xml

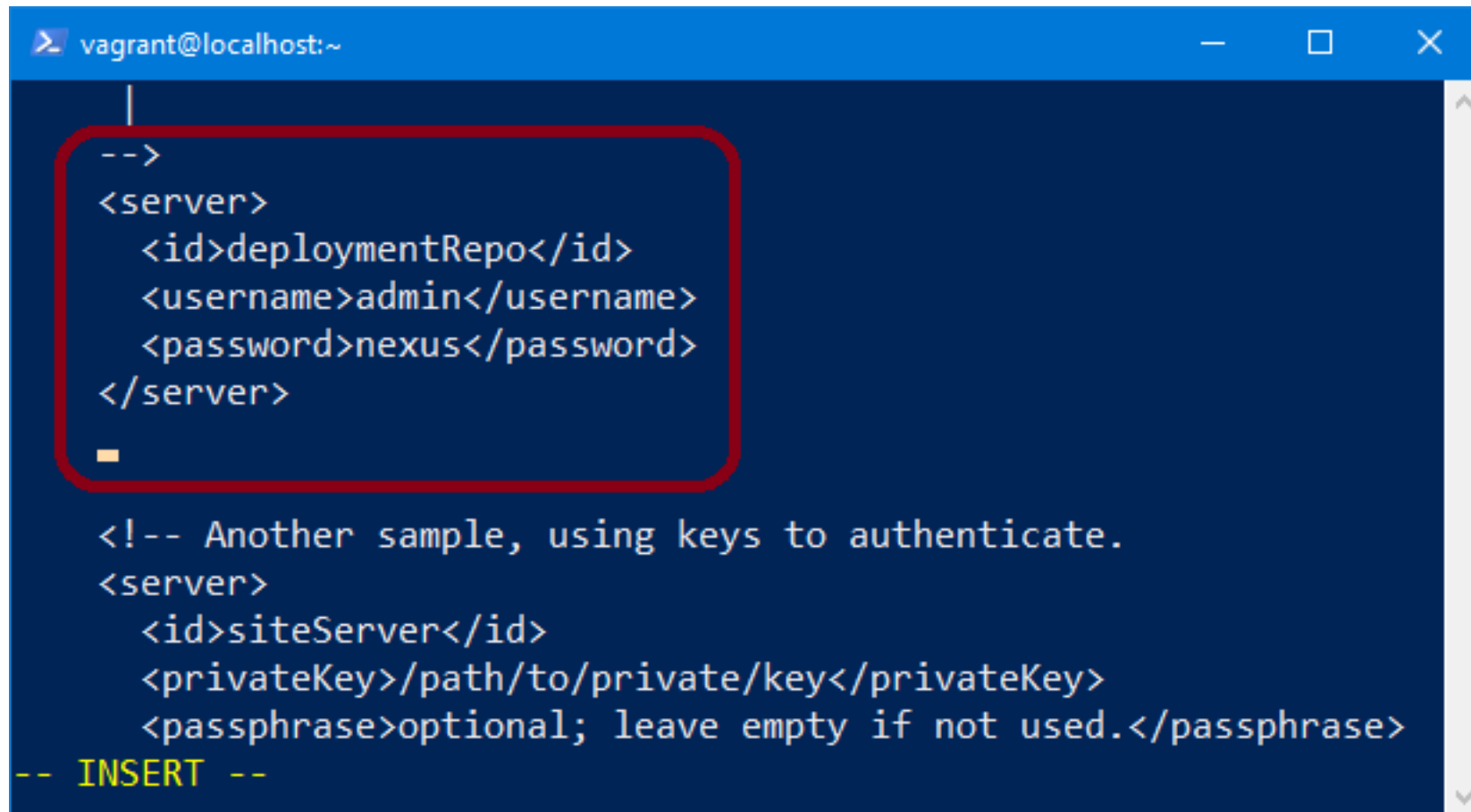
- Pour pouvoir déployer notre application sur nexus et vu que la construction du projet est basée sur Maven, il faut mettre à jour le fichier **settings.xml** sous le dossier conf de maven (installé auparavant).
- Editer ce fichier avec vi par exemple :

A terminal window with a blue title bar. The title bar contains a terminal icon, the text 'vagrant@localhost:~', and window control buttons (minimize, maximize, close). The terminal content shows a command prompt '[vagrant@localhost ~]\$' followed by the command 'sudo vi /usr/share/maven/conf/settings.xml'. The cursor is at the end of the command line. There are also up and down arrow icons on the right side of the terminal window.

```
vagrant@localhost:~  
[vagrant@localhost ~]$ sudo vi /usr/share/maven/conf/settings.xml
```

Configuration Nexus : settings.xml

- **Décommenter (-->)** la section associée au serveur nexus (modifier les champs username et password)
- (avec vi : «i» pour mode édition, «ESC» pour sortir du mode édition, «:wq» pour sauvegarder et sortir :



```
vagrant@localhost:~  
|  
|-->  
<server>  
  <id>deploymentRepo</id>  
  <username>admin</username>  
  <password>nexus</password>  
</server>  
  
<!-- Another sample, using keys to authenticate.  
<server>  
  <id>siteServer</id>  
  <privateKey>/path/to/private/key</privateKey>  
  <passphrase>optional; leave empty if not used.</passphrase>  
-- INSERT --
```

Configuration Nexus : pom.xml

- Configurer votre pom.xml pour le faire pointer vers le repository Nexus deploymentRep :

<!-- Deploy to Nexus -->

<distributionManagement>

<repository>

<id>deploymentRepo</id>

<name>Nexus Repository</name>

<url>http://localhost:8081/repository/maven-releases/</url>

</repository>

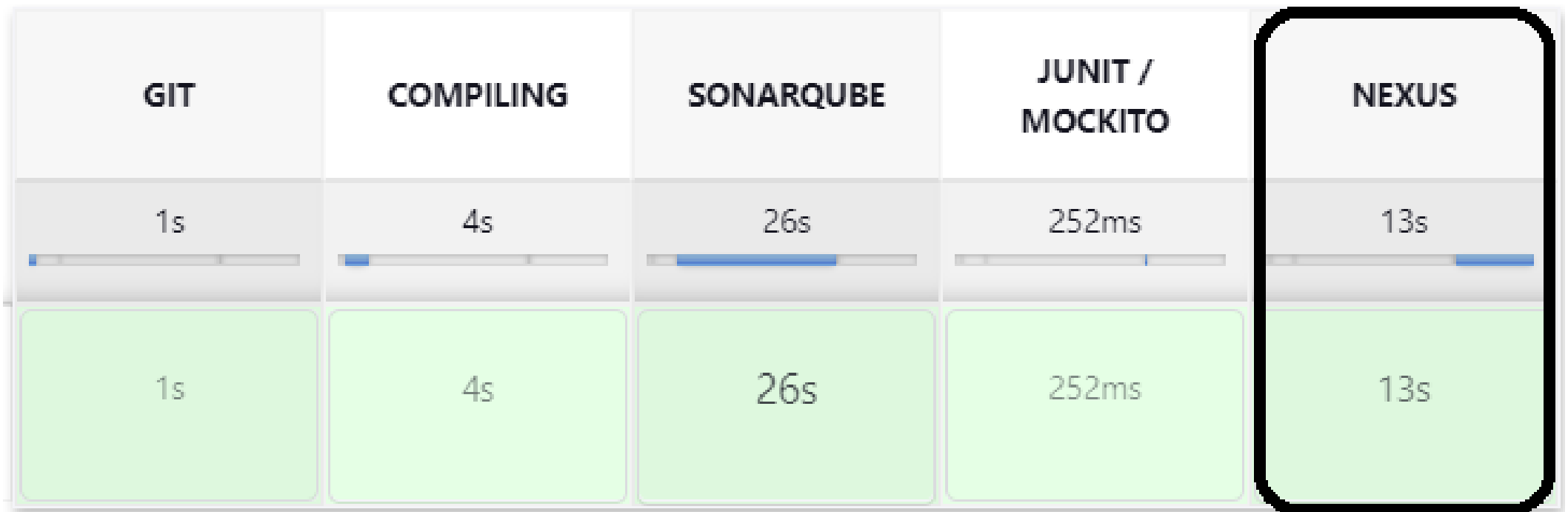
</distributionManagement>

Configuration Nexus avec Jenkins

Les étapes de construction du projet ainsi que le déploiement du livrable sous nexus seront automatiquement réalisés dans Jenkins (ajouter un stage Nexus : commande **mvn deploy** en «skippant» les tests).



Configuration Nexus avec Jenkins



Déploiement sur Nexus

The screenshot displays the Sonatype Nexus Repository Manager web interface. The browser tab is titled "Browse - Nexus Repository Mana" and the address bar shows the URL "192.168.108.222:8081/#browse/browse:maven-releases:tn%2F...". The page header includes the Sonatype logo, version "OSS 3.42.0-01", a search bar, and user information "admin" with a "Sign out" link.

The main content area is titled "Browse / maven-releases". It features a left sidebar with navigation options: "Welcome", "Search", "Browse" (highlighted), and "Upload". The main panel shows a tree view of the repository structure:

- tn
 - esprit
 - spring
 - services
 - timesheet-devops
 - 1.0
 - timesheet-devops-1.0.jar (highlighted)
 - timesheet-devops-1.0.jar.md5
 - timesheet-devops-1.0.jar.sha1
 - timesheet-devops-1.0.pom
 - timesheet-devops-1.0.pom.md5
 - timesheet-devops-1.0.pom.sha1
 - maven-metadata.xml
 - maven-metadata.xml.md5
 - maven-metadata.xml.sha1

On the right side, there is a "Delete asset" button and a "Summary" section with the following details:

Summary	
Repository	maven-releases
Format	maven2
Component Group	tn.esprit.spring.services
Component Name	timesheet-devops
Component Version	1.0
Path	tn/esprit/spring/services/timesheet-devops/1.0/timesheet-devops-1.0.jar
Content type	application/java-archive

Nexus : Indications

- **Attention** : Si vous arrêtez le conteneur Nexus (en arrêtant la VM par exemple), il ne faut pas lancer un `docker run` sur l'image nexus, car cela créera un nouveau conteneur. Il faut juste faire :

```
> Sélection vagrant@localhost:~  
[vagrant@localhost ~]$ sudo chmod 666 /var/run/docker.sock  
[vagrant@localhost ~]$ docker ps -a  
CONTAINER ID        IMAGE               COMMAND  
18eefef10a94        sonatype/nexus3    "/opt/sc
```

- Le `chmod` vous permet d'utiliser les commandes docker sans `sudo`
- Puis faire un `docker start id-conteneur-nexus` :

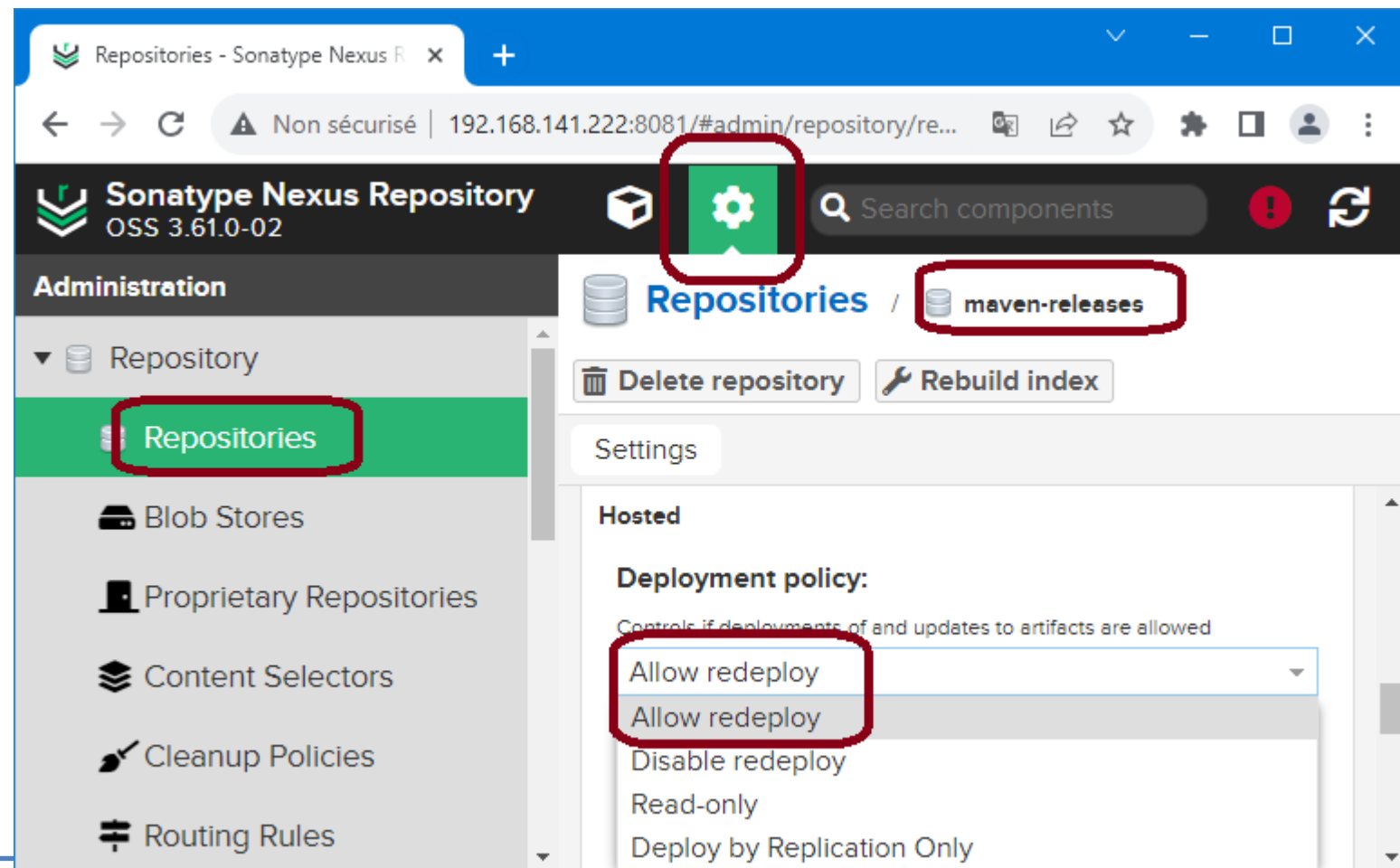
```
> Sélection vagrant@localhost:~  
[vagrant@localhost ~]$ docker start 18eefef10a94
```

Nexus : Indications

- Nous vous avons montré que les deux fichiers pom.xml et settings.xml servent à configurer l'accès à Nexus. Mais :
- Au lieu de modifier pom.xml (de votre projet) et settings.xml de Maven, vous pouvez aussi mettre toutes les propriétés (**id**, **username**, **password**, **url**) dans l'un ou l'autre des fichiers pom.xml ou settings.xml.

Nexus : Indications

- Si vous souhaitez redéployer le même artéfacts (achat-1.0.jar), alors vous avez le choix entre déployer sur le repo **snapshot** : <http://localhost:8081/repository/maven-snapshots/> Ou autoriser le redéploiement sur le repo **release** :



Nexus

 Sonatype
Nexus