

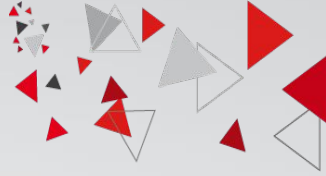


DevOps sonatype

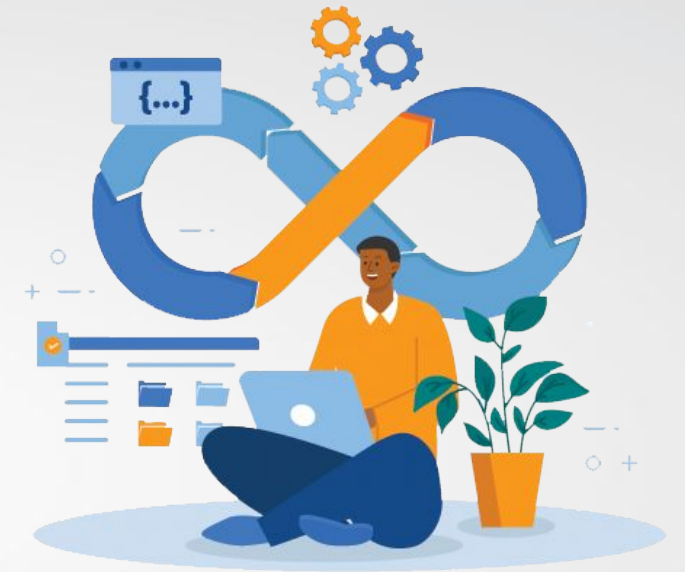
Chapitre 7 : Nexus

ESPRIT – UP ASI (Architecture des Systèmes d'Information)
Bureau E204

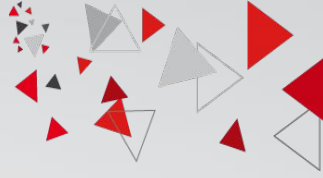
► Plan du cours



- Introduction
- Nexus : Définition et caractéristiques
- Release vs Snapshot
- Installation de Nexus
- Configuration de Nexus en standalone
- Configuration de Nexus avec Jenkins
- **Travail à faire** (Automatiser le déploiement des artéfacts en utilisant Jenkins)



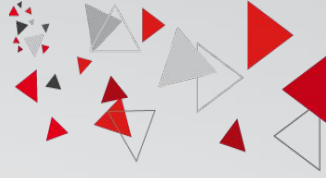
► Introduction



- 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 chaîne DevOps est **d'automatiser la gestion des livrables** et **l'alimentation automatique des différents environnements** par les *artéfacts* correspondants.



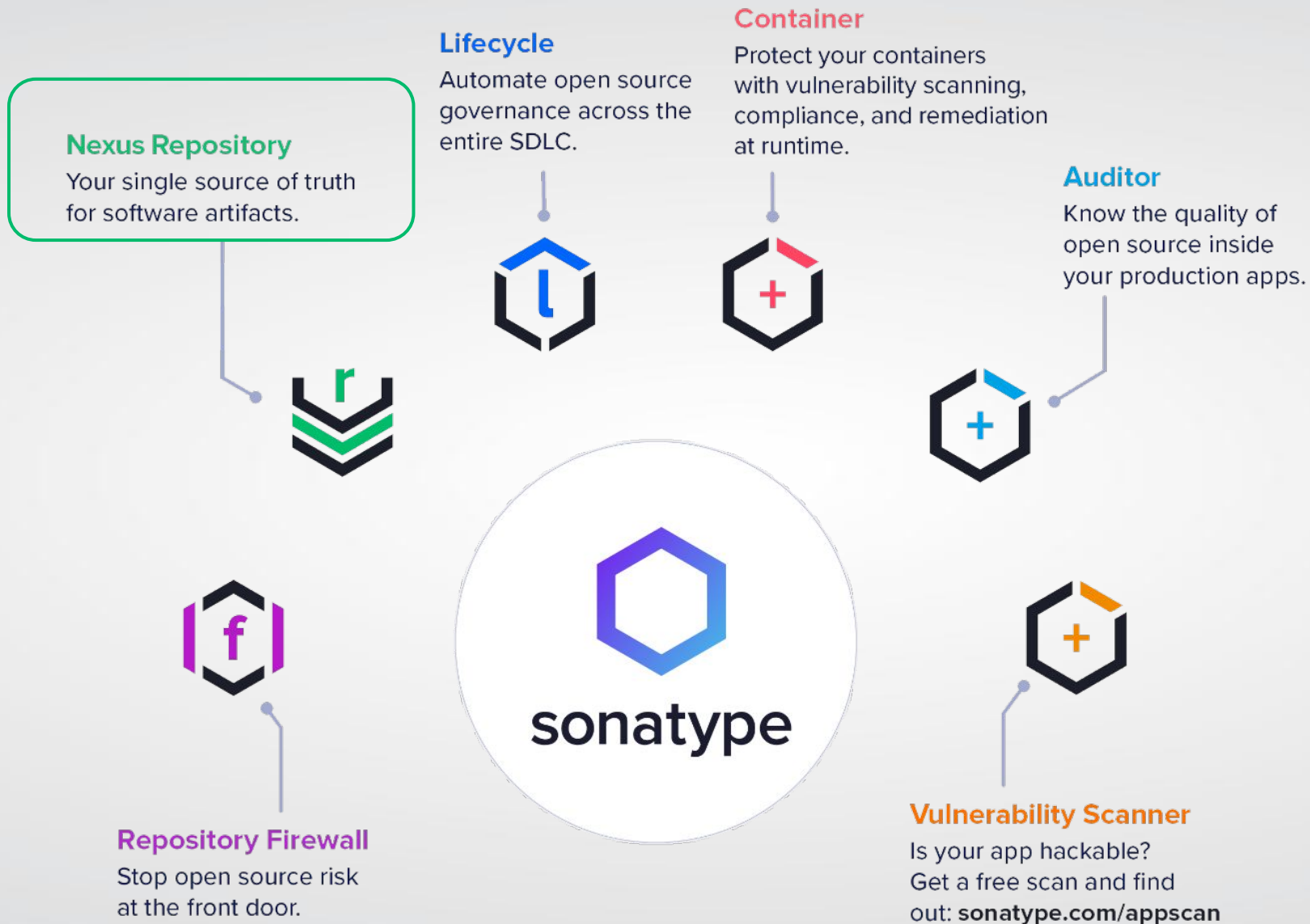
► Introduction



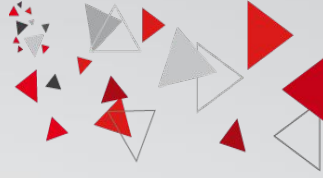
- Le développement de logiciels est un processus complexe qui implique de nombreux composants, dépendances et artefacts. Gérer efficacement ces éléments est essentiel pour garantir le succès d'un projet logiciel.
- C'est là qu'intervient le **Gestionnaire de Dépôts Nexus** de sonatype.



La plateforme Sonatype



Nexus - Définition



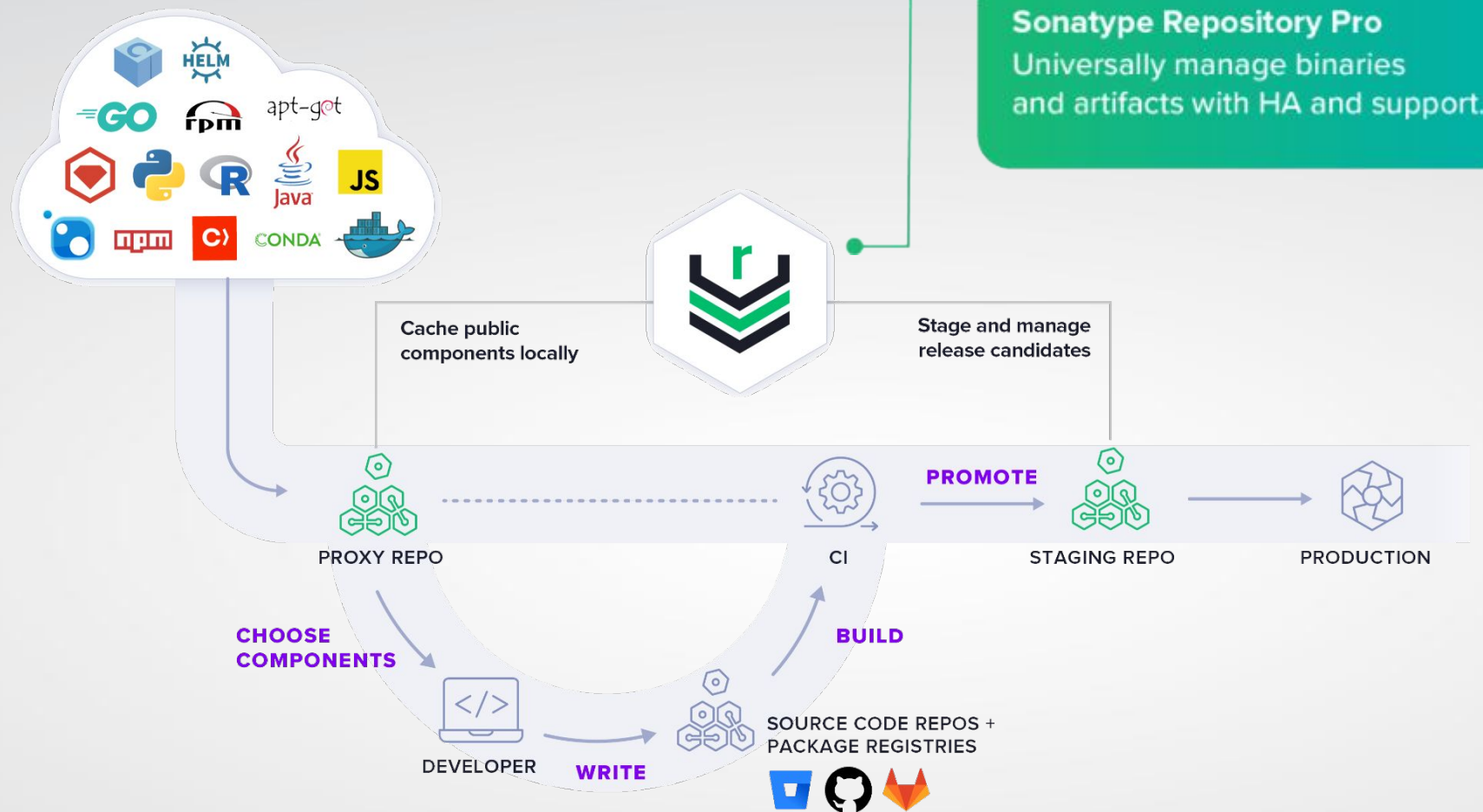
- › **Nexus** est une plateforme de **gestion de dépôts**, principalement utilisée pour héberger des artefacts.
- › Ces *artefacts* sont des composants générés lors de la construction d'un projet, tels que des fichiers JAR, WAR, et autres, qui sont ensuite stockés dans Nexus à l'aide de l'outil Maven.



- › Nexus est fortement intégré à Maven pour la gestion des dépendances et le stockage de ces artefacts.



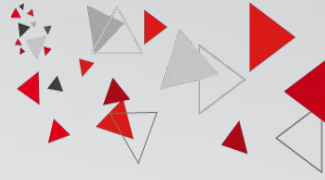
► Nexus - Définition



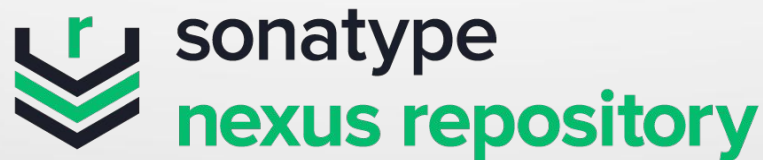
- › **Nexus** permet de partager facilement des fichiers entre les développeurs d'un projet ou entre différents environnements comme Docker.



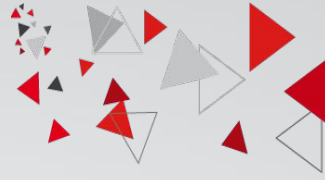
Nexus - Caractéristiques



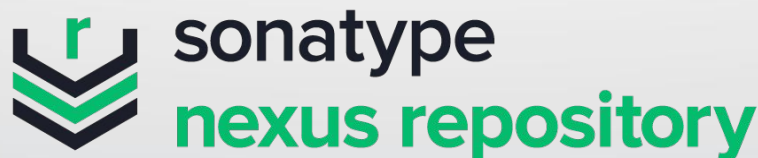
- › **Stockage centralisé** : Nexus permet de centraliser le stockage de tous les composants et artefacts nécessaires pour un projet logiciel. Cela évite d'avoir des dépendances dispersées sur différents systèmes.
- › **Gestion des dépendances** : Il facilite la gestion des dépendances en s'assurant que les versions correctes des bibliothèques et des composants sont disponibles pour les développeurs.



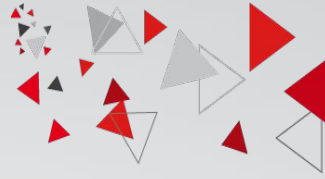
Nexus - Caractéristiques



- › **Distribution sécurisée** : Nexus assure la distribution sécurisée des dépendances aux équipes de développement. Cela garantit que les composants sont fiables et exempts de vulnérabilités.
- › **Intégration avec les outils DevOps** : Il s'intègre souvent dans les pipelines DevOps, ce qui permet d'automatiser le téléchargement, le déploiement et la gestion des dépendances.
- › **Amélioration des performances** : Il peut contribuer à accélérer les temps de construction et de déploiement des applications en offrant un accès rapide aux composants nécessaires.



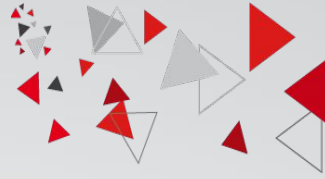
Nexus - Dépôts



- › Les dépôts dans Nexus se réfèrent à des emplacements de stockage centralisés où vous pouvez conserver des artefacts, des composants logiciels, des bibliothèques et d'autres éléments nécessaires à votre projet. **Nexus Repository Manager** est conçu pour gérer ces dépôts.



Nexus - Dépôts



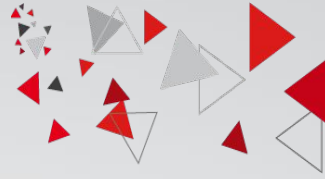
- › **Hosted** : Les dépôts hébergés dans Nexus sont des espaces de stockage qui contiennent les dépôts créés par les utilisateurs. Par défaut, Nexus est configuré avec les types de dépôts hébergés suivants :



- **3rd Party** : Pour stocker des bibliothèques externes qui ne se trouvent pas dans les dépôts Maven publics.
- **Releases** : Pour les bibliothèques stables de votre organisation.
- **Snapshots** : Pour les bibliothèques en cours de développement.



Nexus - Dépôts



- › **Proxy** : Les dépôts proxy agissent comme un relais pour accéder à des artefacts externes, tels que des bibliothèques open source provenant de dépôts publics sur Internet. Nexus peut les mettre en cache localement pour accélérer le téléchargement.

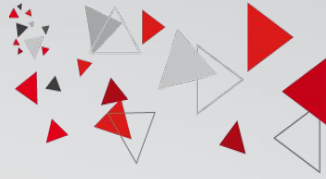


- › **Group** : Les dépôts de groupe permettent de regrouper plusieurs dépôts, ce qui facilite la recherche et l'accès à des artefacts à partir de sources multiples.

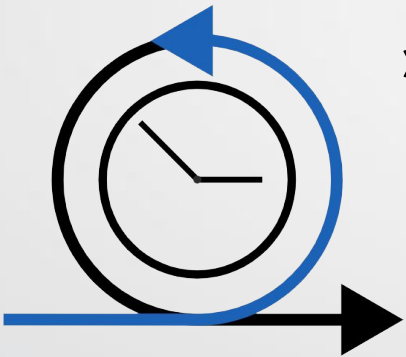
→ Nous allons opter pour l'utilisation des dépôts hébergés "**Hosted**" pour nos projets.



► Versions- Release vs Snapshot



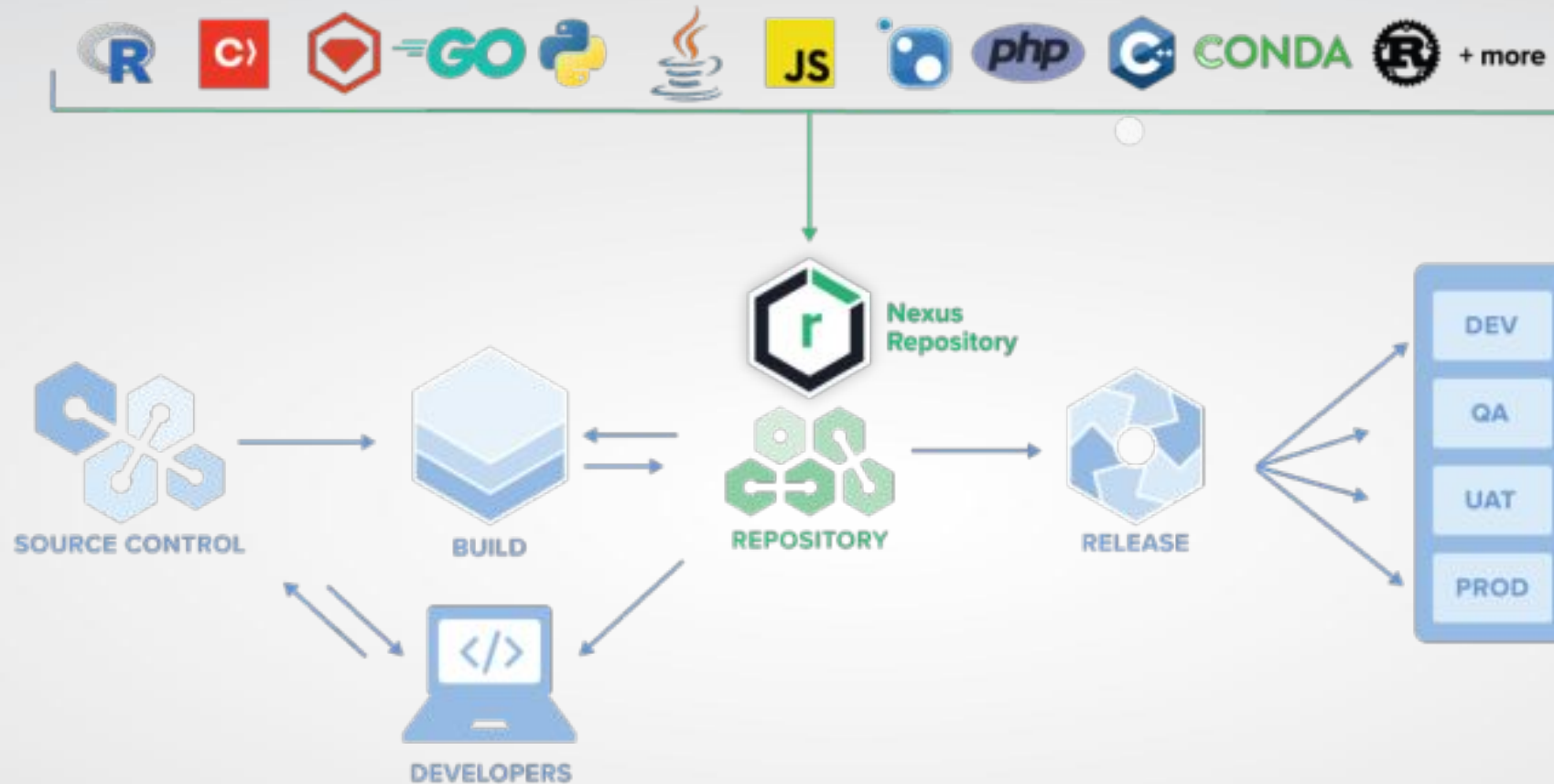
- › Une **release** est une version stable et figée d'un projet, englobant toutes les fonctionnalités requises pour une itération donnée (ou sprint, dans le contexte de Scrum).



- › Une **version snapshot** est une version en cours de développement, comprenant seulement certaines des fonctionnalités à implémenter.



► Versions- Release vs Snapshot



- › Il est possible d'avoir plusieurs versions snapshots pour le même projet, destinées à divers usages, tels que la livraison à l'équipe de test, la mise en place d'une solution en cours de validation, et plus encore.

Nexus - Installation



1. Téléchargez l'image Docker de nexus:

- › Connectez-vous à votre machine virtuelle Ubuntu en utilisant un client SSH. Pour ce faire, démarrez VirtualBox, ouvrez une fenêtre PowerShell et exécutez les commandes `vagrant up` et `vagrant ssh`.
- › Assurez-vous d'avoir effectué un `chmod` au préalable pour éviter les problèmes de droits d'accès.
- › Utilisez la commande `docker pull` pour télécharger l'image **nexus3** depuis le Docker Hub.

```
vagrant@vagrant:~$ sudo chmod 666 /var/run/docker.sock
vagrant@vagrant:~$ docker pull sonatype/nexus3
Using default tag: latest
latest: Pulling from sonatype/nexus3
967391cae45a: Pull complete
13ba478eb6c3: Pull complete
```



Nexus - Installation



2. Exécutez le conteneur nexus :

- › Vous pouvez maintenant lancer un conteneur nexus en utilisant la commande `docker run`.

```
vagrant@vagrant:~$ docker run -d -p 8081:8081 --name nexus sonatype/nexus3
ca69320e0467444545f820f6885c9c8a67a2c36e80d82eeff5fa90336e93c9a6
vagrant@vagrant:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ca69320e0467	sonatype/nexus3	"/opt/sonatype/nexus..."	20 seconds ago	Up 11 seconds	0.0.0.0:8081->8081/tcp, :::8081->8081/tcp	nexus

```
vagrant@vagrant:~$
```

`-d` : Lance le conteneur en en arrière-plan

`--name nexus` : Attribue un nom au conteneur, ici "*nexus*"

`-p 8081:8081` : Le port 8081 du conteneur est mappé sur le port 8081 de votre hôte.



ATTENTION : Choisissez un autre port, si vous utilisez déjà le port 8081 pour Jenkins.

Nexus - Installation



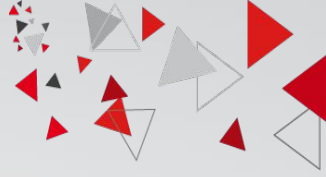
3. Attendez que le conteneur Nexus 3 soit entièrement démarré :

- › Vous pouvez vérifier les journaux du conteneur pour voir quand il est prêt. Vous pouvez utiliser la commande : `docker logs -f nexus`

```
-----  
Started Sonatype Nexus OSS 3.61.0-02  
-----
```

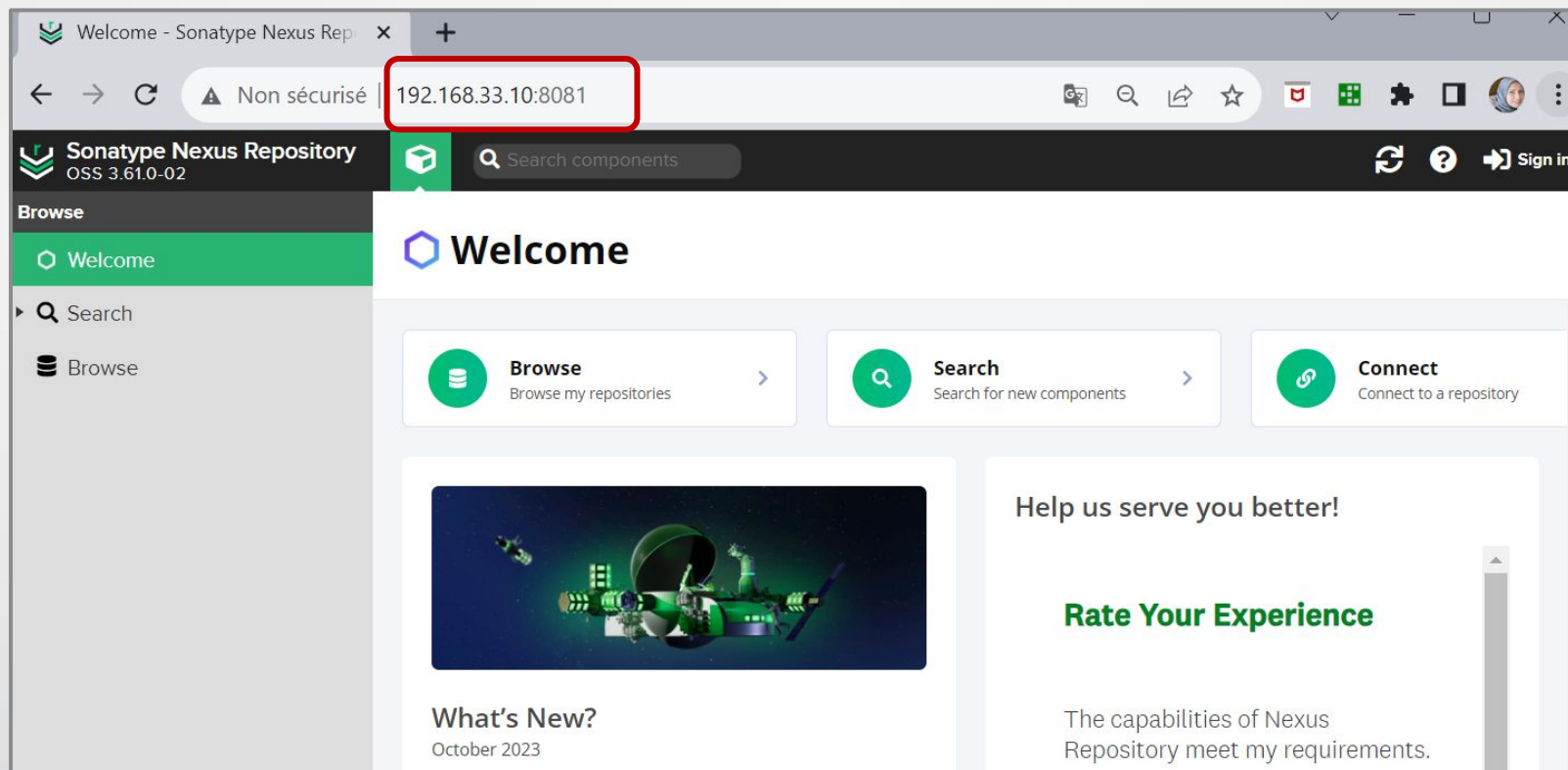


Nexus - Utilisation

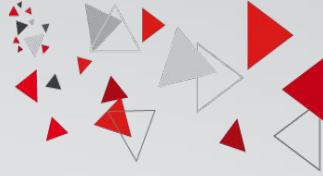


4. Accédez à l'interface web de Nexus Repository Manager :

4.1. Vous pouvez maintenant accéder à l'interface web de nexus repository manager en ouvrant votre navigateur et en visitant <http://<adresse-ip-vm>:8081>.



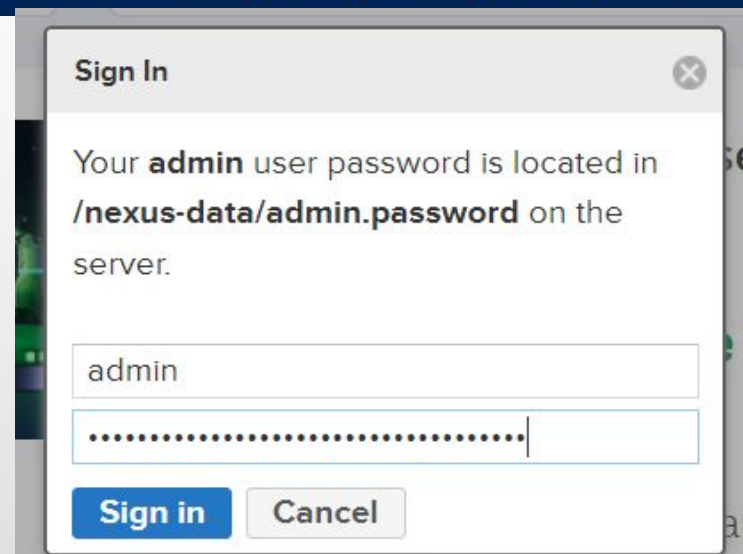
Nexus - Configuration



4.2. Vous devrez vous connecter avec les informations d'identification suivantes:

- › username : `admin`
- › password : Accédez au conteneur en utilisant la commande `docker exec -i <id_conteneur ou nom_conteneur>` et affichez le fichier indiqué dans l'interface d'authentification en utilisant la commande `cat`.

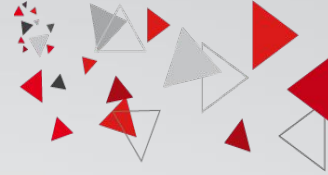
```
vagrant@vagrant:~$ docker exec -i nexus cat /nexus-data/admin.password  
0aaaae87-5167-40f9-9143-6f82b337d978vagrant@vagrant:~$
```



A screenshot of the Nexus Sign In dialog box. The title bar says "Sign In" with a close button. The main text reads: "Your **admin** user password is located in **/nexus-data/admin.password** on the server." Below this, there are two input fields: the first contains the text "admin", and the second is a password field with dots. At the bottom, there are two buttons: "Sign in" (blue) and "Cancel" (grey).



Nexus - Configuration



4.3. Modifiez le mot de passe pour **nexus**, par exemple, et autorisez 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:

Confirm password:

Back **Next**

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**

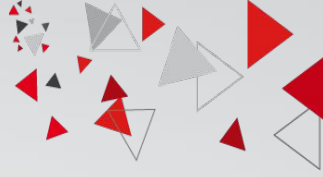
Complete 4 of 4

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

Finish



Nexus - Configuration

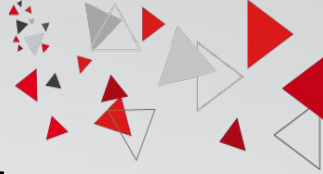


- › Pour déployer notre application sur Nexus, qui repose sur Maven pour la construction du projet, modifiez le fichier **settings.xml** situé dans le dossier `conf` de Maven (préalablement installé).
- › Vous pouvez le modifier en utilisant un éditeur tel que **vi**, par exemple :

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



► Nexus - setting.xml



- › Supprimez les commentaires de la section associée au serveur Nexus (et modifiez les champs `username` et `password` si nécessaire).

```
<servers>
  <!-- server
    Specifies the authentication information to use when connecting to a particular s
    a unique name within the system (referred to by the 'id' attribute below).

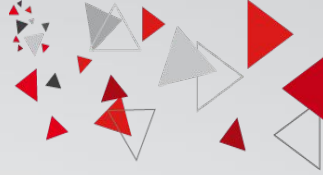
    NOTE: You should either specify username/password OR privateKey/passphrase, since
    used together.

  -->
  <server>
    <id>deploymentRepo</id>
    <username>admin</username>
    <password>nexus</password>
  </server>

  <!-- Another sample, using keys to authenticate.
  <server>
    <id>siteServer</id>
  </server>
-- INSERT --
```



Nexus - pom.xml



- › Modifiez votre fichier **pom.xml** afin de le configurer de manière à ce qu'il fasse référence au dépôt Nexus "deploymentRep".

```
<!-- Deploy to Nexus -->
```

```
<distributionManagement>
```

```
  <repository>
```

```
    <id>deploymentRepo</id>
```

```
    <name>Nexus Repository</name>
```

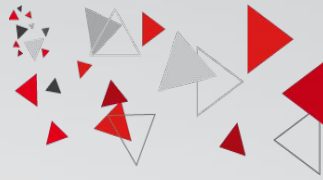
```
    <url>http://localhost:8081/repository/maven-releases/</url>
```

```
  </repository>
```

```
</distributionManagement>
```



► Configuration Nexus avec Jenkins



- › L'ensemble du processus de construction du projet, ainsi que le déploiement du résultat dans Nexus, sera automatisé grâce à Jenkins.
- › Cela peut inclure l'ajout d'une étape spécifique à Jenkins, où la commande **mvn deploy** sera exécutée, avec l'option de "**skip**" les tests.



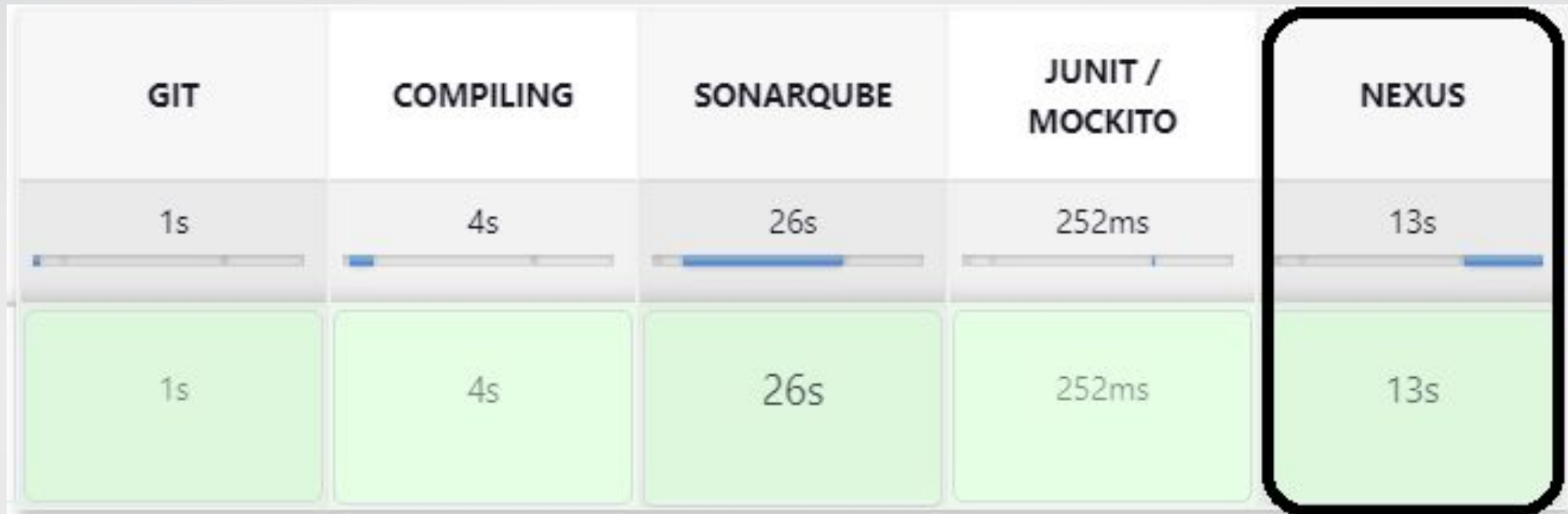
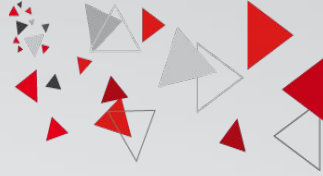
Jenkins



sonatype
nexus repository



► Configuration Nexus avec Jenkins



Jenkins




sonatype
nexus repository



Déploiement sur Nexus

The screenshot displays the Sonatype Nexus Repository web interface. The browser address bar shows the URL `192.168.33.10:8081/#browse/browse:maven-releases:tn%2Fesprit%2F...`. The interface includes a sidebar with navigation options: Welcome, Search, Browse (selected), and Upload. The main content area shows the breadcrumb `Browse / maven-releases` and a search bar. A tree view on the left shows the directory structure: `tn` > `esprit` > `spring` > `services` > `timesheet-devops` > `1.0`. The selected asset is `timesheet-devops-1.0.jar`. A summary table on the right provides details about the asset.

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
File size	37.2 MB



*"Apprendre par le projet, c'est découvrir
▶ par l'action, créer par la compréhension, et
réussir par la persévérance."*