

Intégration continue avec JENKINS

v1.0.0 | 02/09/2025 | Auteur : Bauer Baptiste

Chapitre | Durée de réalisation : 4 heures

Table des matières

1. Introduction	1
2. Objectifs	2
3. Activité 1 : Installation et configuration de Jenkins	3
3.1. Étape 1 : Téléchargement et exécution de l'image Docker Jenkins	3
3.1.1. Télécharger l'image Docker Jenkins	3
3.1.2. Démarrer le conteneur (Docker) Jenkins	4
3.2. Étape 2 : Configuration initiale de Jenkins	7

1. Introduction

Imaginez un monde où chaque modification de code est automatiquement testée, compilée et déployée sans que vous ayez besoin d'intervenir manuellement. C'est exactement ce que permet Jenkins. Jenkins est un outil open-source d'intégration continue et de livraison continue (CI/CD), utilisé par des millions de développeurs et d'équipes DevOps à travers le monde.

Sa force ? Automatiser vos tâches répétitives de build, de test et de déploiement, afin d'accélérer vos livraisons et de sécuriser vos projets logiciels. Dans cette vidéo, nous allons découvrir ce qu'est Jenkins, à quoi il sert, et pourquoi il est devenu un incontournable dans l'univers du DevOps et du développement moderne.

Jenkins est un logiciel open-source qui permet d'automatiser les étapes du cycle de vie d'une application : compiler le code, lancer les tests, détecter les erreurs et même déployer automatiquement sur un serveur.

Pourquoi est-ce important ?

Parce qu'au lieu de refaire manuellement les mêmes tâches à chaque modification, Jenkins s'occupe de tout. C'est un peu comme un chef d'orchestre qui coordonne tous les outils du développeur et de l'administrateur système.

Dans ce cours, on va voir ensemble : ce qu'est Jenkins et son rôle dans l'intégration continue (CI) et la livraison continue (CD), comment il s'installe et s'utilise, et quelques exemples concrets pour vos projets."

2. Objectifs

A chaque modification de code d'une application dans le gestionnaire de configuration, Jenkins se charge automatiquement de la recompiler et de la tester.

Cette séquence a pour objectif de mettre en place un processus d'intégration continue lors du développement d'une application grâce à Jenkins. Elle se compose de plusieurs parties permettant :

- d'installer et de configurer Jenkins ;
- de créer des builds et les lancer de manière automatisée ;
- de créer et de configurer des pipelines dans Jenkins ;
- d'écrire un JenkinsFile ;
- de créer des déclencheurs pour automatiser le lancement de jobs après une mise à jour d'une application ;
- de créer des scripts Jenkins pour répondre à des besoins spécifiques.

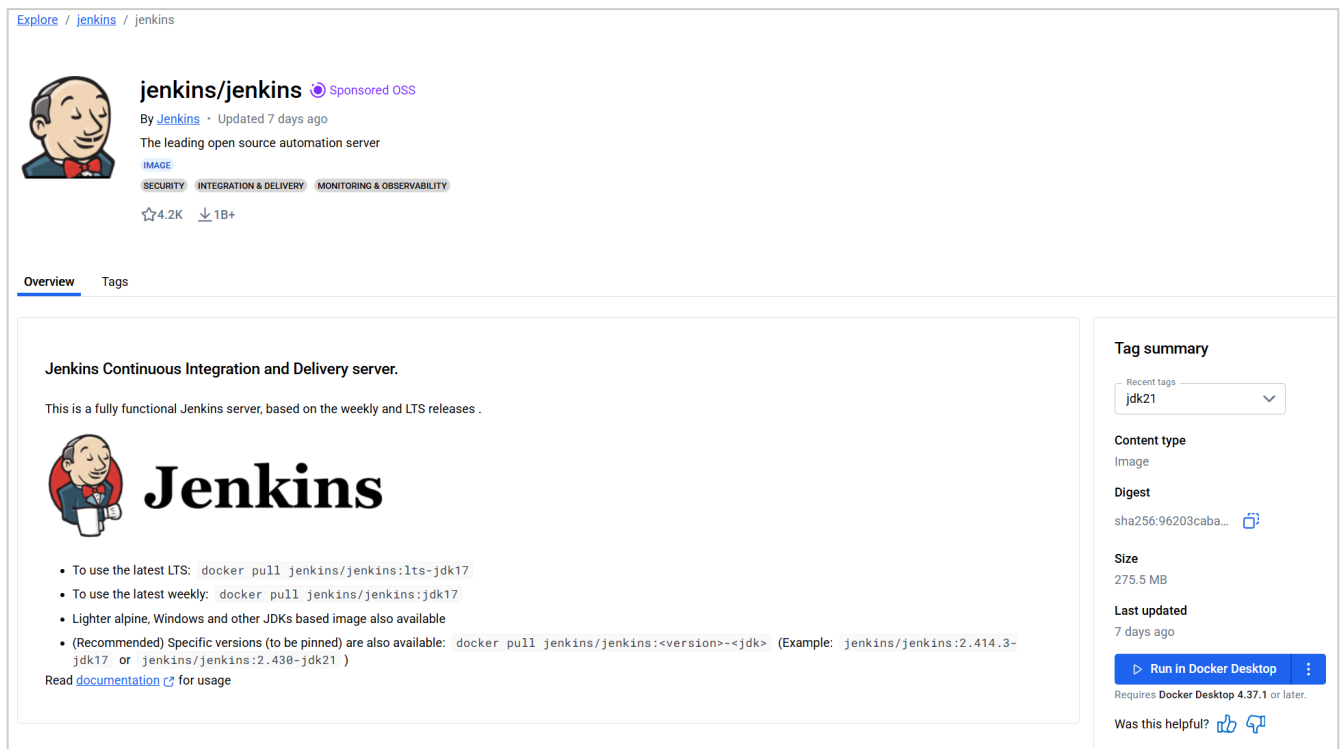
3. Activité 1 : Installation et configuration de Jenkins

3.1. Étape 1 : Téléchargement et exécution de l'image Docker Jenkins

3.1.1. Télécharger l'image Docker Jenkins

Dans cette étape, vous allez télécharger l'image Docker Jenkins. Vous allez ensuite démarrer une instance de l'image et vérifier que le serveur Jenkins est en cours d'exécution.

Déterminons dans un premier temps quelle sera l'image Docker que nous allons utiliser pour installer Jenkins. Nous allons utiliser l'image officielle de Jenkins disponible sur Docker Hub. Cette image est maintenue par la communauté Jenkins et est régulièrement mise à jour avec les dernières versions de Jenkins et des plugins.

The screenshot shows the Jenkins Docker Hub page. At the top, it says 'jenkins/jenkins' with a 'Sponsored OSS' badge. Below this, it says 'By Jenkins · Updated 7 days ago' and 'The leading open source automation server'. There are tags for 'IMAGE', 'SECURITY', 'INTEGRATION & DELIVERY', and 'MONITORING & OBSERVABILITY'. The page has 4.2K stars and 1B+ pulls. The 'Overview' tab is selected, showing the Jenkins logo and the text 'Jenkins Continuous Integration and Delivery server. This is a fully functional Jenkins server, based on the weekly and LTS releases.' Below this, there are instructions on how to pull the image: 'To use the latest LTS: docker pull jenkins/jenkins:lts-jdk17', 'To use the latest weekly: docker pull jenkins/jenkins:jdk17', 'Lighter alpine, Windows and other JDKs based image also available', and '(Recommended) Specific versions (to be pinned) are also available: docker pull jenkins/jenkins:<version>-<jdk> (Example: jenkins/jenkins:2.414.3-jdk17 or jenkins/jenkins:2.438-jdk21)'. A link to 'Read documentation for usage' is provided. On the right, the 'Tag summary' section shows the 'Recent tags' dropdown set to 'jdk21', the 'Content type' as 'Image', the 'Digest' as 'sha256:96203caba...', the 'Size' as '275.5 MB', and the 'Last updated' time as '7 days ago'. There is a button 'Run in Docker Desktop' and a note 'Requires Docker Desktop 4.37.1 or later.' At the bottom, it asks 'Was this helpful?' with thumbs up and down icons.

<https://hub.docker.com/r/jenkins/jenkins>

A ce jour le tag de version LTS la plus récente est la **lts-jdk17** :

```
docker pull jenkins/jenkins:lts-jdk17
```

Version LTS ou JDK ?



• Jenkins LTS

- LTS = Long-Term Support (support à long terme).

- C'est la **version stable** de Jenkins, mise à jour tous les 12 semaines environ.
- Elle est recommandée pour la **production** (entreprise, serveurs d'intégration continue).
- Objectif : fiabilité, sécurité, stabilité.

□ À l'inverse, il existe aussi la version **weekly**, publiée chaque semaine, qui contient les dernières nouveautés, mais qui peut être moins stable.

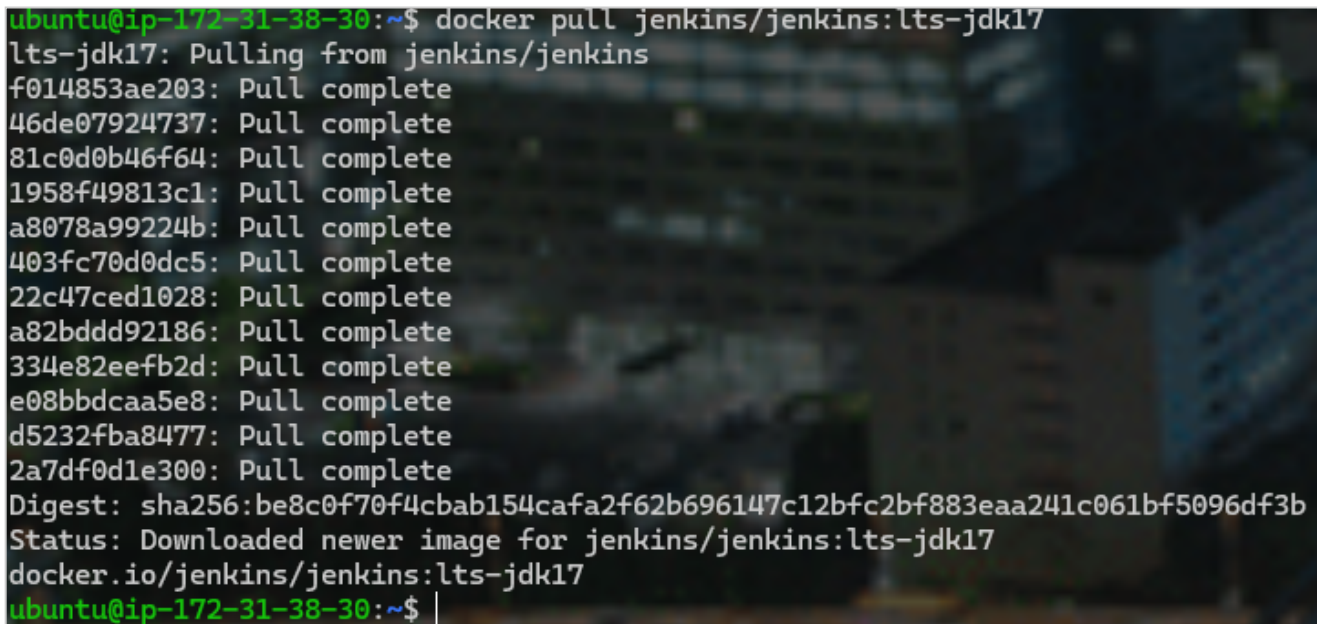
- **JDK (Java Development Kit)**

- Jenkins est écrit en Java, donc il a besoin d'un JDK pour tourner.
- Le JDK est le kit qui contient :
 - la JVM (Java Virtual Machine) : exécute le code Java,
 - les **bibliothèques Java**,
 - les **outils de compilation**.

Aujourd'hui, Jenkins fonctionne surtout avec les versions Java 11 et Java 17 (LTS également côté Java).

□ Sans JDK, Jenkins ne peut pas démarrer, un peu comme une voiture sans carburant.

En résumé ⇒ Jenkins LTS = version stable et recommandée du logiciel Jenkins



```
ubuntu@ip-172-31-38-30:~$ docker pull jenkins/jenkins:lts-jdk17
lts-jdk17: Pulling from jenkins/jenkins
f014853ae203: Pull complete
46de07924737: Pull complete
81c0d0b46f64: Pull complete
1958f49813c1: Pull complete
a8078a99224b: Pull complete
403fc70d0dc5: Pull complete
22c47ced1028: Pull complete
a82bddd92186: Pull complete
334e82eefb2d: Pull complete
e08bbdcaa5e8: Pull complete
d5232fba8477: Pull complete
2a7df0d1e300: Pull complete
Digest: sha256:be8c0f70f4cbab154cafa2f62b696147c12bfc2bf883eaa241c061bf5096df3b
Status: Downloaded newer image for jenkins/jenkins:lts-jdk17
docker.io/jenkins/jenkins:lts-jdk17
ubuntu@ip-172-31-38-30:~$ |
```

3.1.2. Démarrer le conteneur (Docker) Jenkins

Une fois l'image téléchargée, vous pouvez démarrer un conteneur Jenkins avec la commande suivante :

```
docker run -d --name jenkins --restart unless-stopped -p 8080:8080 -p 50000:50000 -v
jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock
```

`jenkins/jenkins:lts-jdk17`

Option	Explication
<code>--name jenkins</code>	Donne un nom au conteneur (<code>jenkins</code> ici), pratique pour le gérer plus facilement (<code>docker stop jenkins</code> , <code>docker logs jenkins</code>).
<code>-d</code>	Lance Jenkins en arrière-plan (il continue à tourner même si tu fermes ton terminal).
<code>--restart unless-stopped</code>	Redémarre automatiquement après un crash ou un redémarrage du serveur (sauf si tu arrêtes manuellement le conteneur).
<code>-p 8080:8080</code>	Expose le port web de Jenkins (accès à l'interface via http://localhost:8080).
<code>-p 50000:50000</code>	Expose le port utilisé par les agents Jenkins (utile si tu ajoutes des nœuds distants).
<code>-v jenkins_home:/var/jenkins_home</code>	Volume persistant pour sauvegarder la configuration Jenkins (jobs, plugins, utilisateurs, logs...).
<code>-v /var/run/docker.sock:/var/run/docker.sock</code>	Permet à Jenkins d'utiliser Docker de l'hôte (lancer des conteneurs, builder des images), sans installer Docker dans Jenkins.
<code>jenkins/jenkins:lts-jdk17</code>	Image officielle utilisée pour lancer Jenkins. Ici : version LTS (stable) avec Java 17 .

```
ubuntu@ip-172-31-38-50:~$ docker run -d --name jenkins --restart unless-stopped -p 8080:8080 -p 50000:50000 -v jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock jenkins/jenkins:lts-jdk17
47f3568113976881884383e63861b59119a07e724d68f9e9a68bccc875baa58
ubuntu@ip-172-31-38-50:~$
```



Un mot de passe de connection initial est généré lors du premier démarrage de Jenkins. Il est accessible via les logs du conteneur Docker.

`docker logs jenkins`


```

ubuntu@ip-172-31-38-30:~$ docker logs jenkins
Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2025-09-02 13:05:58.159+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war f
2025-09-02 13:05:59.479+0000 [id=1] WARNING o.e.j.ee9.nested.ContextHandler#setContextPath: Empty contex
2025-09-02 13:05:59.640+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-12.0.22; buil
2025-09-02 13:06:00.709+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP S
2025-09-02 13:06:00.832+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=
2025-09-02 13:06:01.833+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory
2025-09-02 13:06:02.201+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.Conte
{STARTED}}
2025-09-02 13:06:02.232+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnec
2025-09-02 13:06:02.265+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started oejs.Server
2025-09-02 13:06:02.267+0000 [id=24] INFO winstone.Logger#logInternal: Winstone Servlet Engine running
2025-09-02 13:06:02.577+0000 [id=23] INFO jenkins.model.Jenkins#<init>: Starting version 2.516.2
2025-09-02 13:06:02.808+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Started initializati
2025-09-02 13:06:02.857+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2025-09-02 13:06:04.612+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2025-09-02 13:06:04.622+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2025-09-02 13:06:04.644+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extens
2025-09-02 13:06:05.081+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2025-09-02 13:06:05.082+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: System config adapte
2025-09-02 13:06:05.082+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-09-02 13:06:05.084+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for al
2025-09-02 13:06:05.146+0000 [id=45] INFO hudson.util.Retrier#start: Attempt #1 to do the action check
2025-09-02 13:06:06.485+0000 [id=31] INFO jenkins.install.SetupWizard#init:

*****
*****
*****

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

31cd1a6b2c514252a8e924181e04486f

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*****
*****
*****

2025-09-02 13:06:12.930+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Completed initializa
2025-09-02 13:06:12.960+0000 [id=23] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and
2025-09-02 13:06:13.739+0000 [id=45] INFO h.m.DownloadService$Downloadable#load: Obtained the updated
2025-09-02 13:06:13.740+0000 [id=45] INFO hudson.util.Retrier#start: Performed the action check update
ubuntu@ip-172-31-38-30:~$ |

```



Copiez le mot de passe admin qui s'affiche dans le résultat de la commande, comme surligné sur la page précédente. Il sera nécessaire pour configurer le serveur Jenkins.

Mon code : **31cd1a6b2c514252a8e924181e04486f**

si vous perdez votre mot de passe, vous pourrez le retrouver en exécutant la commande suivante :

```
docker exec -it jenkins /bin/bash
```

Puis une fois dans le shell du conteneur, exécutez :

```
cat /var/jenkins_home/secrets/initialAdminPassword
```



```
ubuntu@ip-172-31-38-30:~$ docker exec -it jenkins /bin/bash
jenkins@47f356811397:/$ cat /var/jenkins_home/secrets/initialAdminPassword
31cd1a6b2c514252a8e924181e04486f
jenkins@47f356811397:/$ exit
exit
ubuntu@ip-172-31-38-30:~$
```

3.2. Étape 2 : Configuration initiale de Jenkins

- Ouvrir un onglet du navigateur web et accéder à :
- <http://localhost:8080/> si vous êtes sur une VM locale.
- http://<ip_public>; si vous êtes sur un serveur distant.

Se connecter à l'aide de votre mot de passe administrateur précédemment copié.

Il est probable que le port **8080** soit bloqué par les règles de votre **pare-feu**. Dans ce cas, vous devez ouvrir ce port pour accéder à l'interface web de Jenkins.

Dans les groupes de sécurité de votre instance EC2, ajoutez une règle entrante pour autoriser le trafic HTTP (port 8080) depuis votre adresse IP.



Modifier les règles entrantes

Les règles entrantes contrôlent le trafic entrant qui est autorisé à atteindre l'instance.

ID de règle de groupe de sécurité	Type	Protocole	Plage de ports	Source	Description - facultatif	Supprimer
sg-01a5dfe57ad5f8314	SSH	TCP	22	Personn...	Q. 0.0.0.0/X	Supprimer
sg-0967225d39bed7fa	HTTP	TCP	80	Personn...	Q. 0.0.0.0/X	Supprimer
sg-05e21f30616939dbf	HTTPS	TCP	443	Personn...	Q. 0.0.0.0/X	Supprimer
	TCP personnalisé	TCP	8080	N'impor...	Q. 0.0.0.0/X	Supprimer

Ajouter une règle

Les règles dont la source est 0.0.0.0/0 ou ::/0 permettent à toutes les adresses IP d'accéder à votre instance. Nous vous recommandons de paramétrer les règles du groupe de sécurité de sorte que les accès soient uniquement autorisés depuis des adresses IP connues.

Annuler Aperçu des modifications Enregistrer les règles

Vous devriez voir la page de configuration initiale de Jenkins.

⚠ Non sécurisé 52.47.185.227:8080/login?from=%2F

Démarrage

Débloquer Jenkins

Pour être sûr que Jenkins soit configuré de façon sécurisée par un administrateur, un mot de passe a été généré dans le fichier de logs ([où le trouver](#)) ainsi que dans ce fichier sur le serveur :

```
/var/jenkins_home/secrets/initialAdminPassword
```

Veuillez copier le mot de passe depuis un des 2 endroits et le coller ci-dessous.

Mot de passe administrateur

Continuer

Cliquez sur Installer les plugins suggérés et attendre que Jenkins télécharge et installe les plugins.

Personnaliser Jenkins

Les plugins étendent Jenkins avec des fonctionnalités additionnelles pour satisfaire différents besoins.

Installer les plugins suggérés

Installer les plugins que la communauté Jenkins trouve les plus utiles.

Sélectionner les plugins à installer

Sélectionner et installer les plugins les plus utiles à vos besoins.

Installation en cours...

Installation en cours...

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	** ASM API
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	⌚ Gradle	** JSON Path API
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline Graph View	** Structs
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ LDAP	** Pipeline: Step API
⌚ Email Extension	⌚ Mailer	⌚ Dark Theme		** commons-text API
				** Token Macro
				Build Timeout
				** bouncycastle API
				** Credentials
				** Plain Credentials
				** Variant
				** SSH Credentials
				Credentials Binding
				** SCM API
				** Pipeline: API



Dans **la fenêtre du terminal** depuis laquelle le serveur a été lancé, il est possible de suivre l'installation des plugins à travers les messages de journal affichés au fur et à mesure que les plugins s'installent. Assurez-vous que vous ne fermez pas cette fenêtre de terminal. Si vous avez besoin de lancer des commandes en ligne, ouvrez une autre fenêtre de terminal.

Une fois l'installation terminée, la fenêtre "**Créer le 1er utilisateur Administrateur**" s'affiche. Pour l'instant, cliquez sur **Skip and continue as admin** en bas.

Démarrage

Créer le 1er utilisateur Administrateur

Nom d'utilisateur


Mot de passe

Confirmation du mot de passe

Nom complet

Adresse courriel

Jenkins 2.516.2

 **Continuer en tant qu'Administrateur****Sauver et continuer**

Dans la fenêtre Configuration de l'Instance, ne modifiez rien. Cliquez sur Sauver et Terminer en bas.

Démarrage

Configuration de l'instance

URL de Jenkins :

L'URL de Jenkins est utilisée pour fournir l'URL de base pour les liens absolus vers les diverses ressources Jenkins. Cela signifie que cette valeur est nécessaire pour le bon fonctionnement de nombreuses fonctionnalités de Jenkins, notamment les notifications par mail, les mises à jour des statuts des pull requests, et la variable d'environnement BUILD_URL fournie pour les étapes de build.

La valeur par défaut affichée **n'est pas encore sauvegardée** et est générée à partir de la requête actuelle, lorsque c'est possible. Il est fortement recommandé d'utiliser comme valeur l'URL qui est censée être utilisée par les utilisateurs. Cela évitera des confusions lors du partage ou de la visualisation de liens.

Jenkins 2.516.2

[Passer cette étape et terminer](#)[Sauver et terminer](#)

Dans la fenêtre suivante, cliquez sur **Commencez à utiliser Jenkins**.

Vous devriez maintenant être sur le tableau de bord principal avec un message : "Bienvenue Sur Jenkins !".