

**BCEPHAL-SETUP-003**

**B-CEPHAL**



**Déploiement Jenkins**

Moriset & co

www.moriset.com

Table des matières

[Table des tableaux iii](#_Toc140908029)

[Table des figures iii](#_Toc140908030)

[1. introduction 1](#_Toc140908031)

[1.1. But du document 1](#_Toc140908032)

[1.2. Cible 1](#_Toc140908033)

[1.3. Acronymes et abréviations 1](#_Toc140908034)

[1.4. Documents de référence 1](#_Toc140908035)

[2. Généralités 1](#_Toc140908036)

[2.1. Définition 1](#_Toc140908037)

[2.2. Quelques concepts 1](#_Toc140908038)

[3. Implémentation 1](#_Toc140908039)

[3.1. Installation et configuration de Jenkins 1](#_Toc140908040)

[3.1.1 Installation 1](#_Toc140908041)

[$ sudo apt update 2](#_Toc140908042)

[3.1.2 Démarrage de Jenkins 2](#_Toc140908043)

[3.1.3 Ouvrir le pare-feu 2](#_Toc140908044)

[3.1.3 Configuration de Jenkins 3](#_Toc140908045)

[3.1.4 Configuration system 7](#_Toc140908046)

[3.1.5 Configuration des outils 8](#_Toc140908047)

[3.1.6 Configuration de sécurité 10](#_Toc140908048)

[3.2. Les jobs Jenkins 11](#_Toc140908049)

[3.2.1. Le job Bcephal-admin-Service 14](#_Toc140908050)

[3.2.2. Le job bcephal-archive-service 14](#_Toc140908051)

[3.2.3. Le job bcephal-billing-service 14](#_Toc140908052)

[3.2.4. Le job Bcephal-Configuration-Service 14](#_Toc140908053)

[3.2.5. Le job bcephal-dashboard-service 14](#_Toc140908054)

[3.2.6. Le job Bcephal-file-manager-Service 14](#_Toc140908055)

[3.2.7. Le job Bcephal-form-Service 14](#_Toc140908056)

[3.2.8. Le job bcephal-gateway-service 14](#_Toc140908057)

[3.2.9. Le job bcephal-initiation-service 14](#_Toc140908058)

[3.2.10. Le job bcephal-messenger-service 14](#_Toc140908059)

[3.2.11. Le job bcephal-planification-service 14](#_Toc140908060)

[3.2.12. Le job bcephal-project-service 14](#_Toc140908061)

[3.2.13. Le job bcephal-reconciliation-service 14](#_Toc140908062)

[3.2.14. Le job Bcephal-Registration-Service 14](#_Toc140908063)

[3.2.15. Le job bcephal-reporting-service 14](#_Toc140908064)

[3.2.16. Le job Bcephal-scheduler-Service 14](#_Toc140908065)

[3.2.17. Le job bcephal-security-service 14](#_Toc140908066)

[3.2.18. Le job Bcephal-settings-Service 14](#_Toc140908067)

[3.2.19. Le job Bcephal-sourcing-grid-Service 14](#_Toc140908068)

[3.2.20. Le job Bcephal-Web-ui-Service 14](#_Toc140908069)

[4. Deploiment 14](#_Toc140908070)

[4.1. Clonage 15](#_Toc140908071)

[4.2. Build 15](#_Toc140908072)

[4.3. Unit Tests 16](#_Toc140908073)

[4.4. Integration Tests 16](#_Toc140908074)

[4.5. Docker build image 16](#_Toc140908075)

[4.6. Docker tag image latest 17](#_Toc140908076)

[4.7. Docker tag image 17](#_Toc140908077)

[4.8. Docker push image latest 17](#_Toc140908078)

[4.9. Deploy app 17](#_Toc140908079)

[5. Conclusion 18](#_Toc140908080)

# Table des tableaux

[Tableau 1: Acronymes et abréviations 1](#_Toc45234097)

[Tableau 2: Document de référence 1](#_Toc45234098)

# Table des figures

Aucune entrée de table d'illustration n'a été trouvée.

# introduction

## But du document

B-CEPHAL peut être déployé sur des environnements Linux ou Windows via une procédure manuelle ou automatisée. Ce document décrit la procédure de déploiement automatique de B-CEPHAL à l’aide de Jenkins.

## Cible

Ce document s’adresse aux administrateurs B-CEPHAL.

## Acronymes et abréviations

|  |  |
| --- | --- |
| Acronyme/Abréviation | Définition |
|  |  |
|  |  |
|  |  |

Tableau 1: Acronymes et abréviations

## Documents de référence

|  |  |
| --- | --- |
| Référence | Titre |
|  |  |
|  |  |
|  |  |

Tableau 2: Document de référence

# Généralités

Cette section présente les concepts généraux liés au déploiement automatique B-CEPHAL.

## Définition

L’architecture micro-service est une approche de développement qui consiste à découper une application petits services, tournant chacun dans un processus autonome et qui communique entre eux via un protocole de communication léger et faiblement typé.

## Quelques concepts

# Implémentation

## Installation et configuration de Jenkins

### 3.1.1 Installation

Prérequis

* Une machine Ubuntu (version 18.04 ou plus récente)
* Accès root ou un utilisateur avec des privilèges sudo
* Connexion Internet

**Étape 1** : Mettre à jour le système

Avant d'installer Jenkins, il est recommandé de mettre à jour les packages existants.

sudo apt update

sudo apt upgrade -y

**Étape 2** : Installer Java

Jenkins nécessite Java pour fonctionner. Installez la version Java 11 (ou une version compatible).

sudo apt install openjdk-11-jdk

Vérifiez l'installation de Java :

java --version

**Étape 3 :** Ajouter le dépôt Jenkins

Ajoutez le dépôt Jenkins à votre système et importez la clé GPG.

Ajoutez la clé GPG Jenkins :

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

Ajoutez le dépôt Jenkins à la liste des sources apt :

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

**Étape 4 :** Installer Jenkins

Après avoir ajouté le dépôt, mettez à jour les sources apt et installez Jenkins.

sudo apt update

sudo apt install jenkins

**Étape 5 :** Démarrer et Activer Jenkins

Démarrez Jenkins et configurez-le pour qu'il se lance au démarrage du système.

sudo systemctl start Jenkins

sudo systemctl enable Jenkins

Pour confirmer l'installation, vérifiez l'état de s'assurer. Maintenant, exécutez la commande, et la sortie indiquée comme ci-dessous.

systemctl status jenkins



**Étape 6 :** Configurer le Pare-feu (si nécessaire)

Si vous utilisez ufw comme pare-feu, ouvrez le port par défaut de Jenkins (8080).

### sudo ufw allow 8080

### sudo ufw reload

|  |
| --- |
| **Remarque** |
| Si le pare-feu est inactif, les commandes suivantes autoriseront OpenSSH et activeront le pare-feu :  $ sudo ufw allow OpenSSH  $ sudo ufw enable |

Vérifiez l’état de ufw pour confirmer les nouvelles règles :

$ sudo ufw status

Vous remarquerez que le trafic vers le port 8080 est autorisé depuis n’importe où :

Output

Status: active

To Action From

-- ------ ----

OpenSSH ALLOW Anywhere

8080 ALLOW Anywhere

OpenSSH (v6) ALLOW Anywhere (v6)

8080 (v6) ALLOW Anywhere (v6)

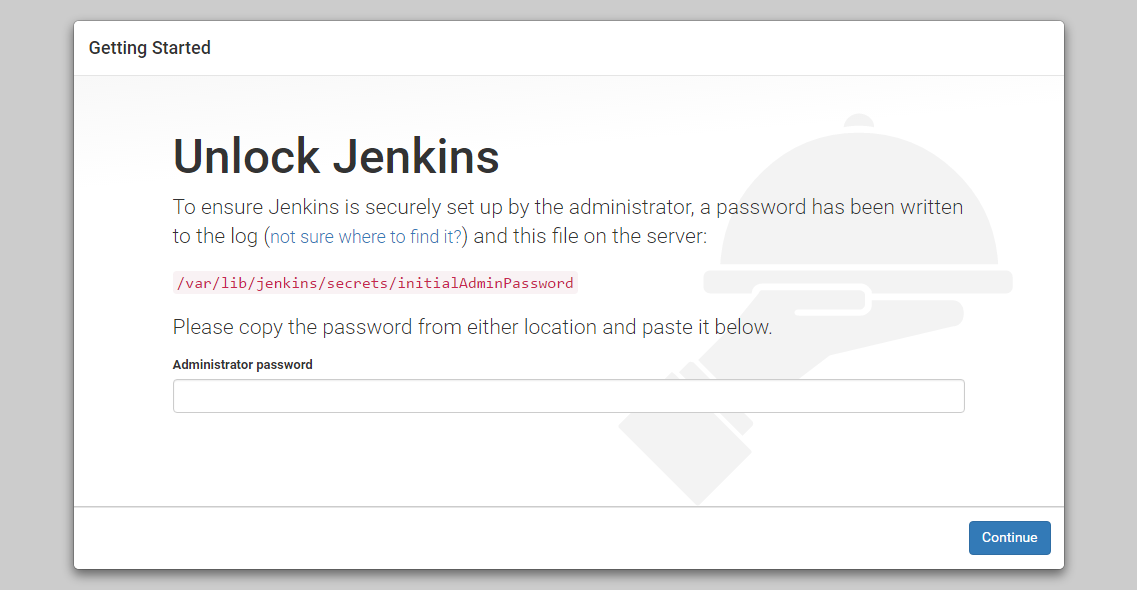
Une fois Jenkins installé et notre pare-feu configuré, nous pouvons achever l’étape d’installation et nous plonger dans la configuration de Jenkins.

### **Étape 7 :** Accéder à Jenkins

Pour configurer votre installation, consultez Jenkins sur son port par défaut, 8080 en utilisant votre nom de domaine ou l’adresse IP de votre serveur :

http://your\_server\_ip\_or\_domain:8080

Vous devriez voir apparaître l’écran **Unlock Jenkins** qui affichera l’emplacement du mot de passe initial :



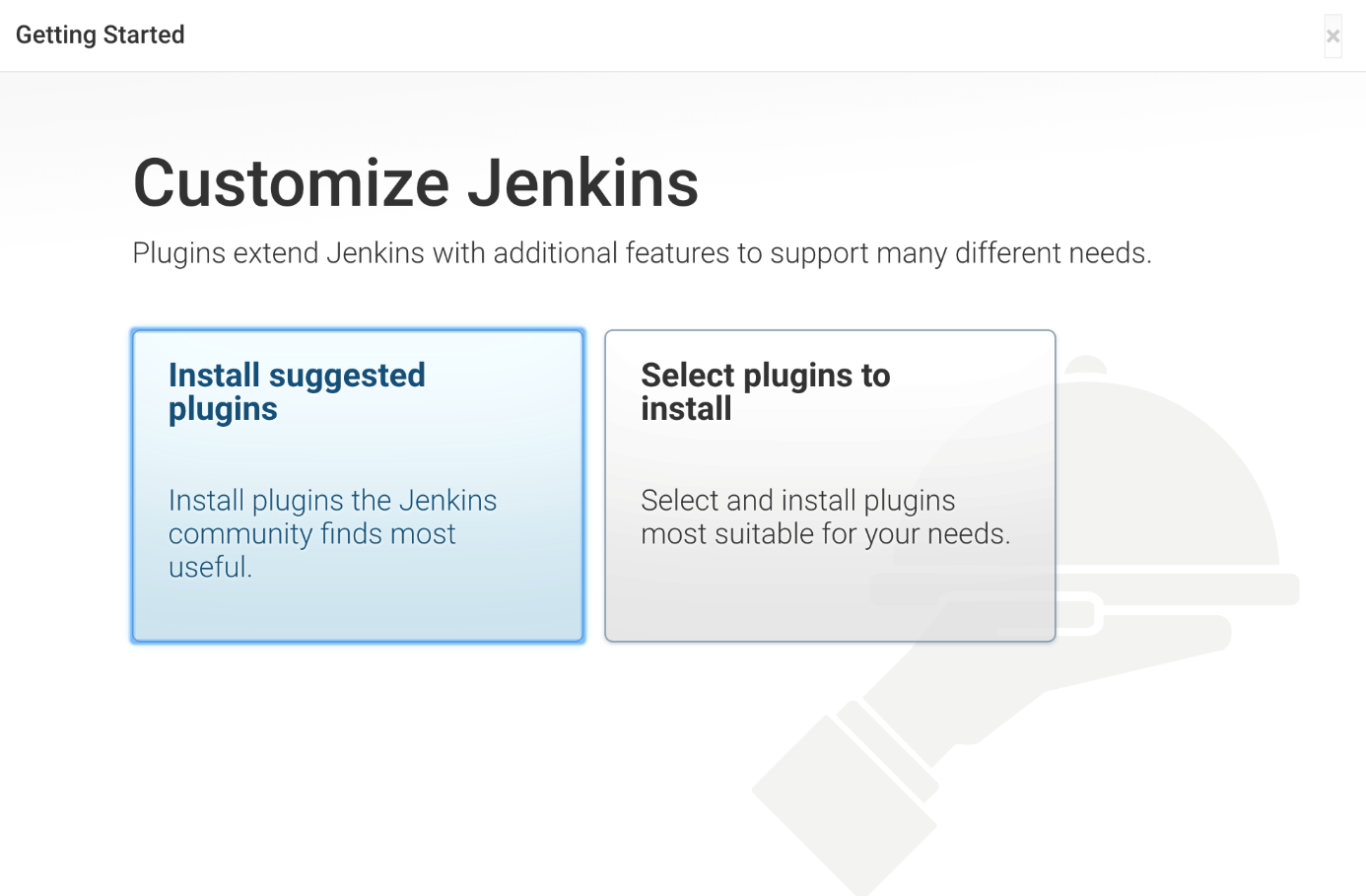
### Vous verrez une page demandant le mot de passe initial. Vous pouvez obtenir ce mot de passe en utilisant la commande suivante :

### sudo cat /var/lib/jenkins/secrets/initialAdminPassword

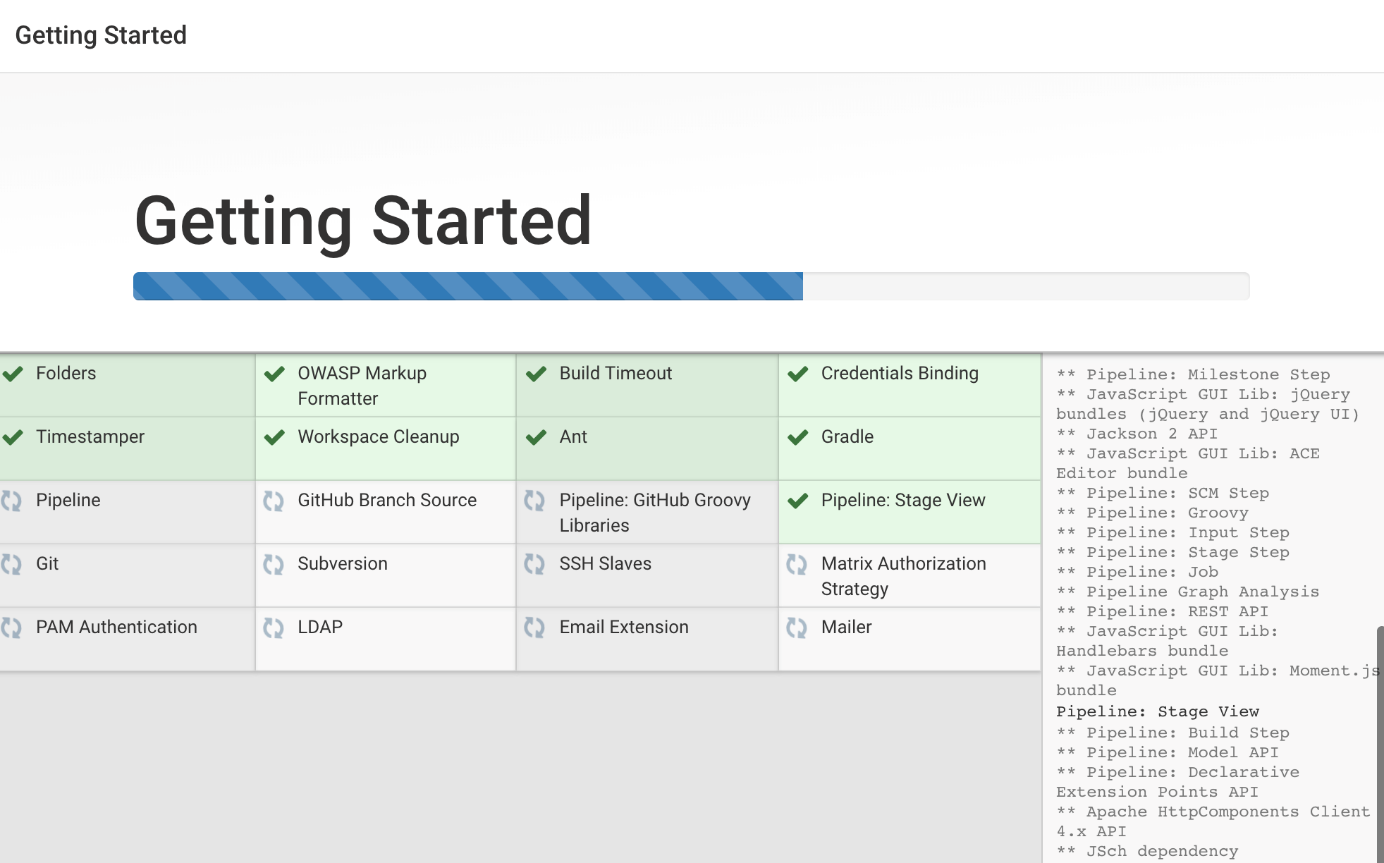
Copiez le mot de passe alphanumérique composé de 32 caractères du terminal et collez-le dans le champ **Administrator password**, puis cliquez sur **Continue**.

**Étape 8 :** Installer les Plugins Recommandés

L’écran suivant présente l’option d’installer des plugins suggérés ou de sélectionner des plugins spécifiques :

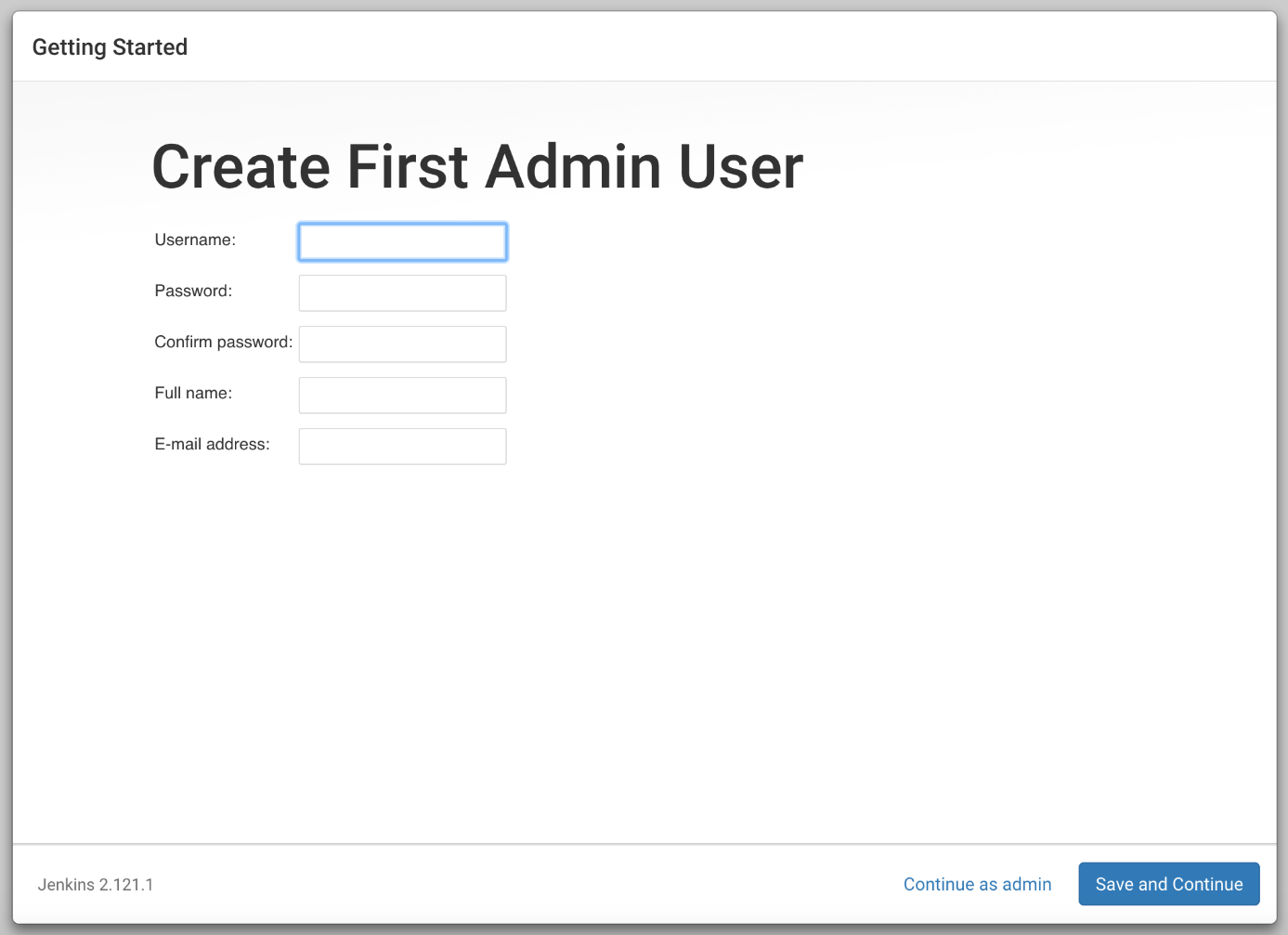


Nous allons cliquer sur l’option **Installer les plugins suggérés**, qui lancera immédiatement le processus d’installation.

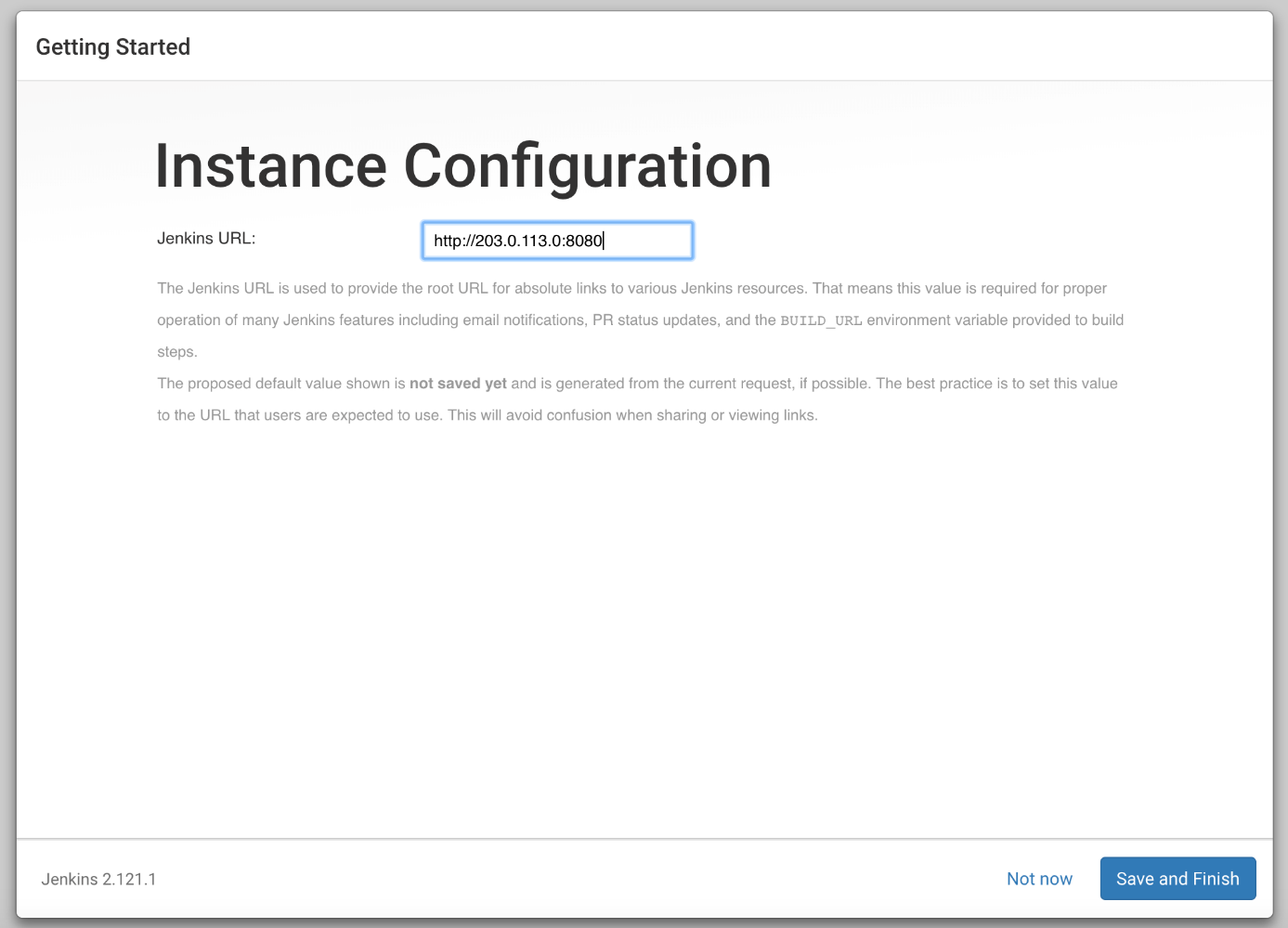


### **Étape 9 :** Créer un Premier Utilisateur Admin

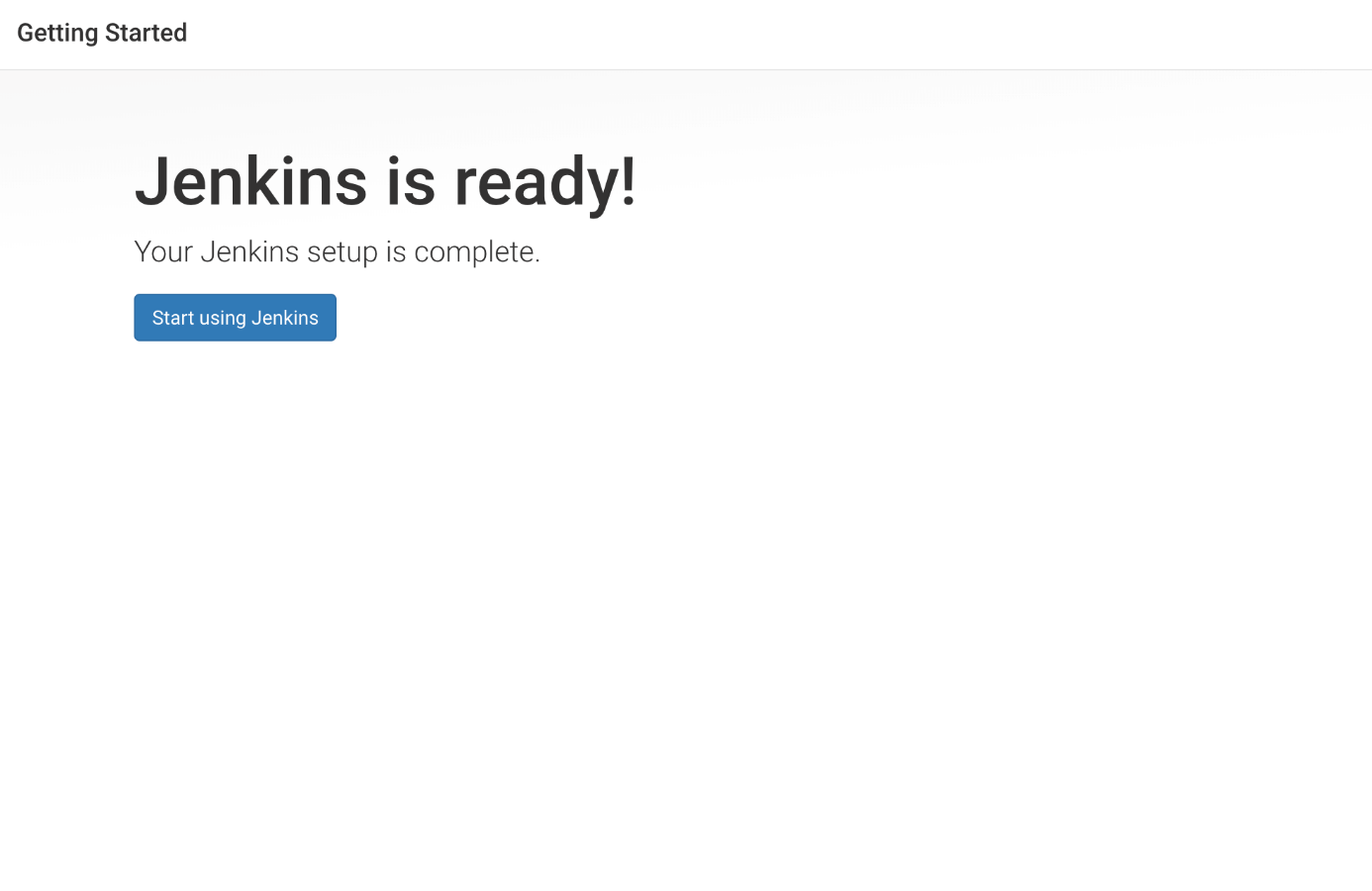
Une fois l’installation terminée, vous serez invité à configurer le premier utilisateur administratif. Vous pouvez sauter cette étape et continuer comme admin en utilisant le mot de passe initial utilisé ci-dessus, mais nous allons prendre quelques minutes pour créer l’utilisateur.



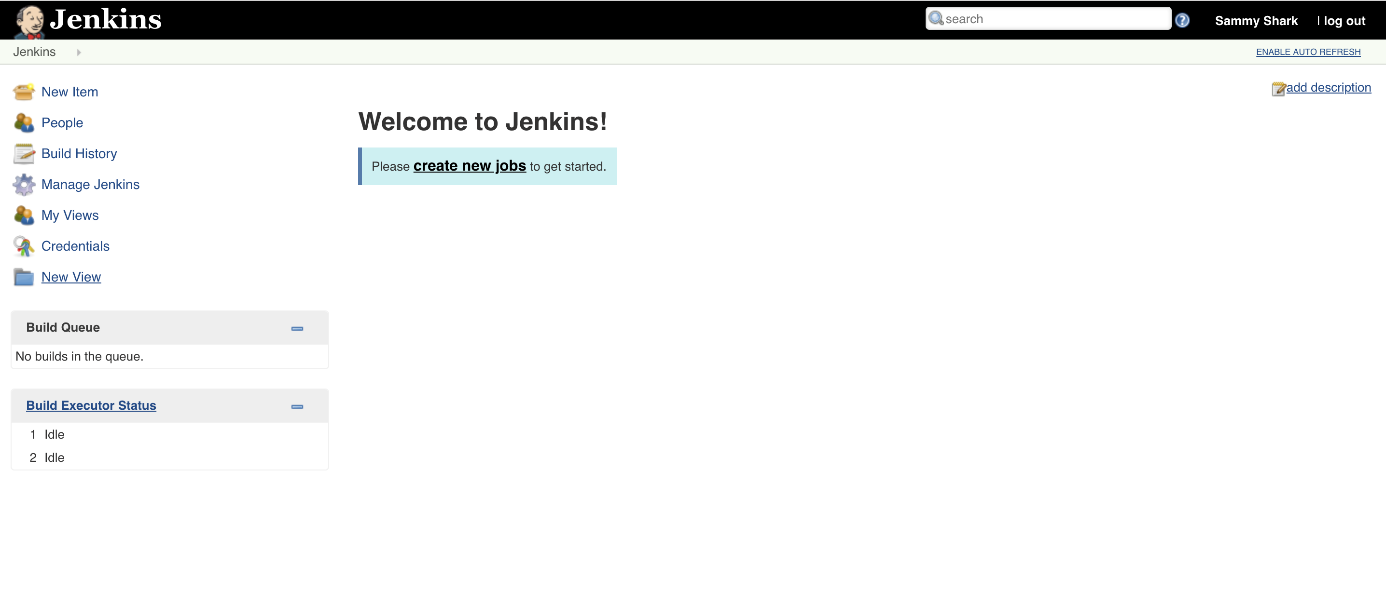
Vous verrez apparaître une page **Instance configuration** qui vous demandera de confirmer l’URL préférée de votre instance Jenkins. Vérifiez qu’il s’agit bien du nom de domaine ou de l’adresse IP de votre serveur :



Après avoir confirmé l’exactitude des informations, cliquez sur **Save and Finish** Vous verrez apparaître une page de confirmation confirmant que **« Jenkins est prêt ! »** :



Cliquez sur **Start using Jenkins** pour consulter le tableau de bord principal de Jenkins :



À ce stade, vous avez terminé d’installer correctement Jenkins.

### 3.1.2 Configuration system

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

### 3.1.3 Configuration des outils

1. JDK

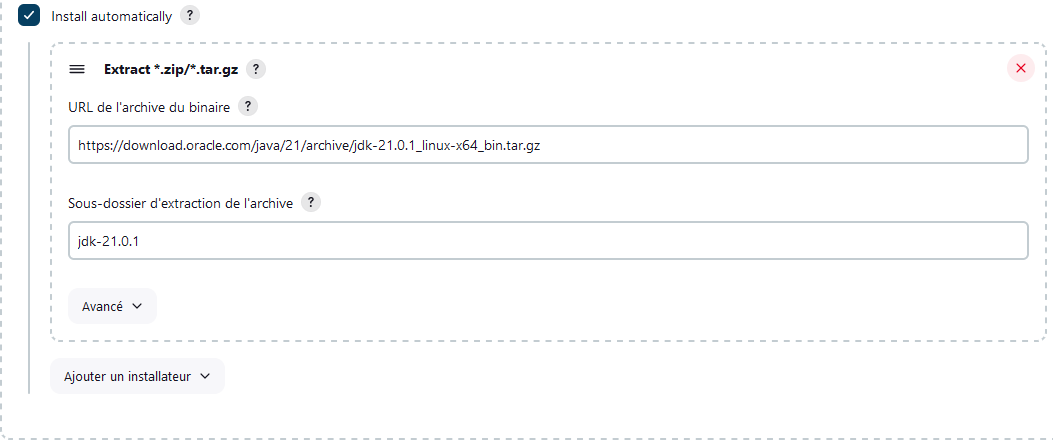
Une image contenant texte, capture d’écran, Police, blanc

Description générée automatiquement

Ouvrir et remplir comme suit :

Une image contenant texte, capture d’écran, ligne, nombre

Description générée automatiquement



1. GRADLE

Une image contenant texte, ligne, capture d’écran

Description générée automatiquement

Une image contenant texte, ligne, reçu, capture d’écran

Description générée automatiquement

1. DOCKER

Une image contenant texte, capture d’écran, ligne, nombre

Description générée automatiquement

### 3.1.4 Configuration de sécuritéw²

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

## Les jobs Jenkins

Description de la partie commune à tous les jobs.

Une image contenant texte, capture d’écran, nombre, ligne

Description générée automatiquement

Insérer ici l’url du dépôt distant du projet

Insérer ici le crédential pour récupérer le projet sur le dépôt distant

Une image contenant texte, capture d’écran, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Il correspond au fichier Jenkins défini sur dans chaque projet.

### Le job [Bcephal-admin-Service](http://192.168.100.93:8080/job/Bcephal-admin-Service/)

### Le job [bcephal-archive-service](http://192.168.100.93:8080/job/bcephal-archive-service/)

### Le job [bcephal-billing-service](http://192.168.100.93:8080/job/bcephal-billing-service/)

### Le job [Bcephal-Configuration-Service](http://192.168.100.93:8080/job/Bcephal-Configuration-Service/)

### Le job [bcephal-dashboard-service](http://192.168.100.93:8080/job/bcephal-dashboard-service/)

### Le job [Bcephal-file-manager-Service](http://192.168.100.93:8080/job/Bcephal-file-manager-Service/)

### Le job [Bcephal-form-Service](http://192.168.100.93:8080/job/Bcephal-form-Service/)

### Le job [bcephal-gateway-service](http://192.168.100.93:8080/job/bcephal-gateway-service/)

### Le job [bcephal-initiation-service](http://192.168.100.93:8080/job/bcephal-initiation-service/)

### Le job [bcephal-messenger-service](http://192.168.100.93:8080/job/bcephal-messenger-service/)

### Le job [bcephal-planification-service](http://192.168.100.93:8080/job/bcephal-planification-service/)

### Le job [bcephal-project-service](http://192.168.100.93:8080/job/bcephal-project-service/)

### Le job [bcephal-reconciliation-service](http://192.168.100.93:8080/job/bcephal-reconciliation-service/)

### Le job [Bcephal-Registration-Service](http://192.168.100.93:8080/job/Bcephal-Registration-Service/)

### Le job [bcephal-reporting-service](http://192.168.100.93:8080/job/bcephal-reporting-service/)

### Le job [Bcephal-scheduler-Service](http://192.168.100.93:8080/job/Bcephal-scheduler-Service/)

### Le job [bcephal-security-service](http://192.168.100.93:8080/job/bcephal-security-service/)

### Le job [Bcephal-settings-Service](http://192.168.100.93:8080/job/Bcephal-settings-Service/)

### Le job [Bcephal-sourcing-grid-Service](http://192.168.100.93:8080/job/Bcephal-sourcing-grid-Service/)

### Le job [Bcephal-Web-ui-Service](http://192.168.100.93:8080/job/Bcephal-Web-ui-Service/)

# Deploiment

Dans cette partie nous verrons en détail le contenu du Jenkinsfile en commençant par l’étape du clonage du projet jusqu’au déploiement sur kubernetes.

Nous travaillerons à cet effet sur le job Bcephal-Configuration-service.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Dans ces sections nous définissons les outils utilisés dans Jenkins et les variables d’environnement que nous utiliserons.

## Clonage

Cette étape consiste à récupérer sur un repository, tout le code du projet sur le quel on travail.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Il s’agit de l’id de notre crédential sur Jenkins qui nous donne l’autorisation de récupérer le code du projet sur le repository.

Remarque : dans cette section il faut bien se rassurer qu’on ait récupérer le code de toutes les dépendances de notre projet, ce qui pourrait rendre cette section d’avantage plus volumineuse en fonction des dépendances de notre projet et chacune d’elles devra avoir une section particulière comme celle-ci.

## Build

C’est dans cette section que nous compilons le code et nous produisons le .jar de notre application.

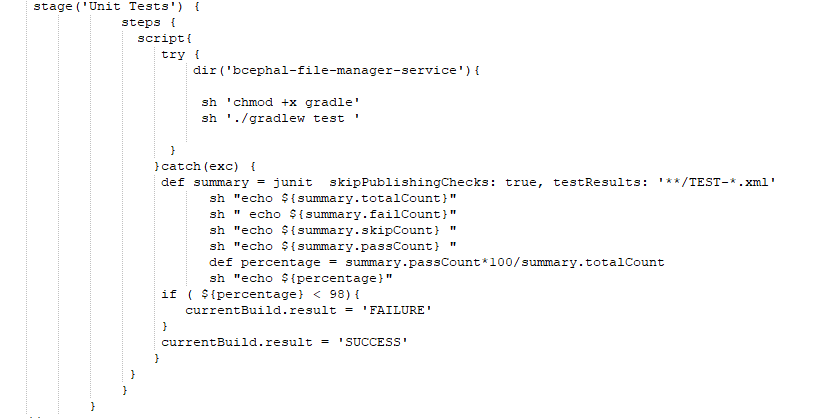
Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Cette commande est utilisée puisque nous travaillons avec un projet gradle ce qui signifie qu’en fonction du type de projet cela peut changer.

## Unit Tests

Ici nous exécutons les tests unitaires.



## Integration Tests

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

## Docker build image

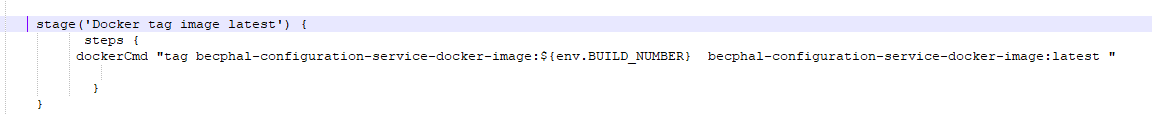
Cette partie est réservée à la production de notre image docker. Pour produire cette image nous devons fournir un dockerfile à notre projet, celui-ci contient l’essentiel de ce dont notre application aura besoin pour fonctionner.

Une image contenant texte, ligne, Police, capture d’écran

Description générée automatiquement

## Docker tag image latest

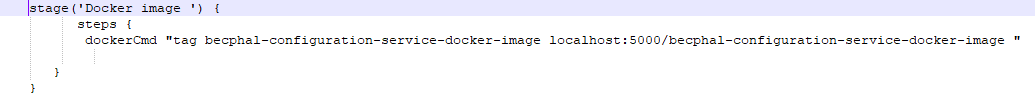
Cette partie nous permet de versionner notre image en : latest



## Docker tag image

Ici, nous renommons notre image en lui ajoutant l’addresse de notre registry. A cet effet nous devons garder à l’esprit qu’une registry devra être installée en locale en exécutant la commande suivante sur notre daemon Docker :

$ docker run -d -p 5000:5000 --restart=always --name registry registry:2



## Docker push image latest

Nous envoyons notre image docker sur notre registry en local.

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement

## Deploy app

Dans cette section, nous exécution le déploiement de notre application qui se fera sur Kuberntes. Celui-ci va récupérer notre image sur la registry locale si nous avons une installée ou sur une autre registry en fonction des configurations que nous aurons fait dans les sections réservées à Docker puis, il va démarrer un conteneur pour cette application en prenant en compte toutes les spécifications contenues dans le fichier yaml de notre application.

Une image contenant texte, capture d’écran, ligne

Description générée automatiquement

Il s’agit ici du crédential associé au fichier de configuration de kubernetes qui permet à Jenkins de se connecter au serveur de Kubernetes et de lui demander d’exécuter des requetes. A l’intérieur de ce fichier de configuration, nous avons l’adresse où se trouve notre serveur Kubernetes et le code pour établir la connexion avec celui-ci.

# Conclusion