

WORKSHOP:

Jenkins



www.smart-it-partner.com



www.sip-academy.com

Plan de Workshop





CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Nous allons débuter notre workshop en créant un repo Bitbucket et push de notre projet dans ce repo, suivant les étapes décrites dans le workshop sur Git.

The screenshot shows the Bitbucket interface for a repository named 'springdata-master'. The repository was created by 'dhia eddine saria' and has one branch, 'master'. The repository settings menu is open on the left. The main area displays the repository's contents, which include several files and folders: '.mvn', '.settings', 'src', '.classpath', and '.gitignore'. All files were committed 3 days ago, and the commit message is 'first commit'.

Name	Size	Last commit	Message
.mvn		3 days ago	first commit
.settings		3 days ago	first commit
src		3 days ago	first commit
.classpath	2.05 KB	3 days ago	first commit
.gitignore	624 B	3 days ago	first commit



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de app password

Créer un app password dans Bitbucket confère un rôle de gestionnaire d'accès et de sécurité sur votre propre compte Bitbucket. Cela permet de générer des mots de passe d'application pour des intégrations tierces ou des services qui nécessitent un accès sécurisé à votre compte Bitbucket sans exposer votre mot de passe principal.



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de app password

Ce rôle permet essentiellement à l'utilisateur de gérer les permissions pour des applications tierces ou des scripts automatisés, en créant des mots de passe spécifiques à ces cas d'utilisation sans divulguer le mot de passe principal du compte Bitbucket. Cela garantit un niveau de sécurité supplémentaire en restreignant l'accès aux fonctionnalités et aux données du compte Bitbucket, limité à l'application tierce concernée.



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de app password

1

Accédez aux paramètres du compte ou du profil : Recherchez l'option "Settings" liée à votre compte ou votre profil Bitbucket et ensuite "personal Bitbucket settings".

The screenshot shows the Bitbucket Settings page. At the top, there is a navigation bar with a 'Create' button, a search bar, and a gear icon. The main area is titled 'Settings' and contains two sections: 'BITBUCKET ADMINISTRATION' and 'PERSONAL SETTINGS'. Under 'PERSONAL SETTINGS', there are three options: 'Atlassian account settings', 'Personal Bitbucket settings', and 'Labs'. The 'Personal Bitbucket settings' option is highlighted with a blue border.

- BITBUCKET ADMINISTRATION
 - Workspace settings
Manage your workspace settings and integrations.
- PERSONAL SETTINGS
 - Atlassian account settings
Manage your language, time zone, and other profile information.
 - Personal Bitbucket settings
Manage your email notification and other Bitbucket settings.
 - Labs
Help us test these new or experimental features before they are released. ...



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de app password

2

Sélectionnez "App passwords"

Personal settings

GENERAL

Account settings

Email aliases

Notifications

ACCESS MANAGEMENT

App authorizations

App passwords

SECURITY

SSH keys

Two-step verification

Sessions

Audit log

FEATURES

Labs

Account settings

Avatar



Name **dhia eddine saria**

Update your avatar, name, email address, and password.

Bitbucket profile settings

Username **dhiasaria5 (change)**

Website

Language

English

Help translate Bitbucket into your language.

[Delete account](#)





CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de app password

3

Choisissez "Create app password" : Vous serez invité à donner un nom à ce mot de passe d'application pour une meilleure identification.

App passwords

App passwords allow users to access their Bitbucket account through apps such as Sourcetree. We'll generate the app passwords for you, and you won't need to remember them.

[Create app password](#)





CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de app password

4

Définissez les autorisations nécessaires :
Bitbucket vous permet généralement de définir les autorisations que ce mot de passe aura. Assurez-vous de sélectionner les autorisations appropriées pour votre cas d'utilisation, telles que l'accès en lecture, en écriture, etc.

Add app password

Details

Label* app-pass

Permissions

Account	Issues	Wikis	Snippets	Webhooks	Pipelines	Runners
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> Read					
<input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> Write					
<input checked="" type="checkbox"/> Write						
<input checked="" type="checkbox"/> Workspace membership	<input checked="" type="checkbox"/> Read					
	<input checked="" type="checkbox"/> Write					
<input checked="" type="checkbox"/> Projects	<input checked="" type="checkbox"/> Read					
	<input checked="" type="checkbox"/> Write					
	<input checked="" type="checkbox"/> Admin					
<input checked="" type="checkbox"/> Repositories	<input checked="" type="checkbox"/> Read					
	<input checked="" type="checkbox"/> Write					
	<input checked="" type="checkbox"/> Admin					
	<input checked="" type="checkbox"/> Delete					
<input checked="" type="checkbox"/> Pull requests	<input checked="" type="checkbox"/> Read					
	<input checked="" type="checkbox"/> Write					

Create Cancel





CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de app password

5

Générez le mot de passe : Une fois que vous avez spécifié les autorisations, Bitbucket générera un mot de passe unique.

New app password

Here is your app password for **app-pass**. You will not be able to view this password again once you close this window, so be sure to record it.

ATBBaM39mww9yRsMY3PzXkqZEGSw561ECFC9

Close





CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de app password

6

Sauvegardez soigneusement le mot de passe généré : Ce mot de passe ne sera affiché qu'une seule fois. Assurez-vous de le copier et de le sauvegarder dans un endroit sûr, car Bitbucket ne le montrera plus par la suite.



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de Credentials jenkins

Le rôle de création de credentials dans Jenkins est généralement attribué aux utilisateurs disposant de privilèges d'administration ou de gestion des configurations. Ce rôle permet à ces utilisateurs de créer, de gérer et de stocker en toute sécurité des informations d'identification sensibles utilisées par Jenkins pour interagir avec des services externes ou exécuter des tâches.



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de Credentials jenkins

Pour créer un credentials dans Jenkins :

- 1. Connectez-vous à Jenkins :** Accédez à votre tableau de bord Jenkins en utilisant vos identifiants d'administration ou d'un compte ayant les autorisations nécessaires pour la gestion des configurations.
- 2. Accédez à la page de gestion des credentials :** Cela peut varier selon la version de Jenkins, mais généralement, vous trouverez cette option dans le panneau de configuration ou sous la section "**Manage Jenkins**" > "**Manage Credentials**".
- 3. Sélectionnez "Add Credentials" :** Cliquez sur le bouton pour ajouter de nouveaux credentials.



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de Credentials jenkins

4. **Choisissez le type de credentials à créer :** Jenkins offre différents types d'identifiants tels que Username with Password, Secret text, SSH Username with private key, etc. Sélectionnez le type approprié en fonction de l'utilisation prévue. Dans notre cas **Username with Password**.
5. **Renseignez les informations demandées :** Selon le type de credentials choisi, remplissez les champs requis. Par exemple, pour un Username with Password, entrez le nom d'utilisateur et le mot de passe associé. (**app password de bitbucket**)
6. **Donnez un nom descriptif à l'identifiant :** Pour une meilleure gestion, donnez un nom significatif à vos credentials pour les reconnaître facilement.
7. **Sauvegardez de credentials :** Une fois que tous les champs nécessaires sont remplis, sauvegardez le credentials nouvellement créé.



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création de Credentials jenkins

New credentials

Kind

Username with password

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Username ?
sariadhiaeddine@gmail.com

Treat username as secret ?

Password ?
.....

ID ?
bitbucket_cred

Description ?
bitbucket_cred

Create



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création d'un job jenkins

Le rôle d'un "job" Jenkins concerne la définition et la gestion des tâches ou flux de travail au sein de Jenkins. La création d'un "job", spécifiquement un "Pipeline job", implique les étapes suivantes :

- 1. Accéder au tableau de bord Jenkins** : Connectez-vous à votre instance Jenkins.
- 2. Créer un nouvel élément ("New Item")** : Cliquez sur "**New Item**" depuis le tableau de bord Jenkins.
- 3. Entrer le nom du job** : Fournissez un nom significatif pour votre "job".



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création d'un job jenkins

4. **Sélectionner Pipeline** : Choisissez l'option "Pipeline" pour le type de "job" que vous souhaitez créer.

The screenshot shows the Jenkins dashboard with a search bar at the top. Below it, there's a section titled 'Enter an item name' containing the text 'demospringd'. A note below the input field says '» Required field'. To the right, there are three project types listed: 'Freestyle project', 'Maven project', and 'Pipeline'. The 'Pipeline' option is highlighted with a blue border, indicating it has been selected. A tooltip for 'Pipeline' states: 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.'





CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création d'un job jenkins

5. Configurer les paramètres du Pipeline :

Configurez le "Pipeline job" en définissant ses propriétés, telles que l'emplacement du code source, les détails de gestion de version.

The screenshot shows the Jenkins Pipeline configuration interface. In the 'Definition' dropdown, 'Pipeline script from SCM' is selected. Under the 'SCM' section, 'Git' is chosen. The 'Repositories' section contains a single repository configuration with the URL `https://dhiasaria5@bitbucket.org/dhiasaria5/gestbank.git` and a credential entry for `sariadhiaeddine@gmail.com/***** (cred_bitbucket)`. At the bottom, there are 'Save' and 'Apply' buttons.



CRÉATION D'UN REPO BITBUCKET ET JOB JENKINS

Création d'un job jenkins

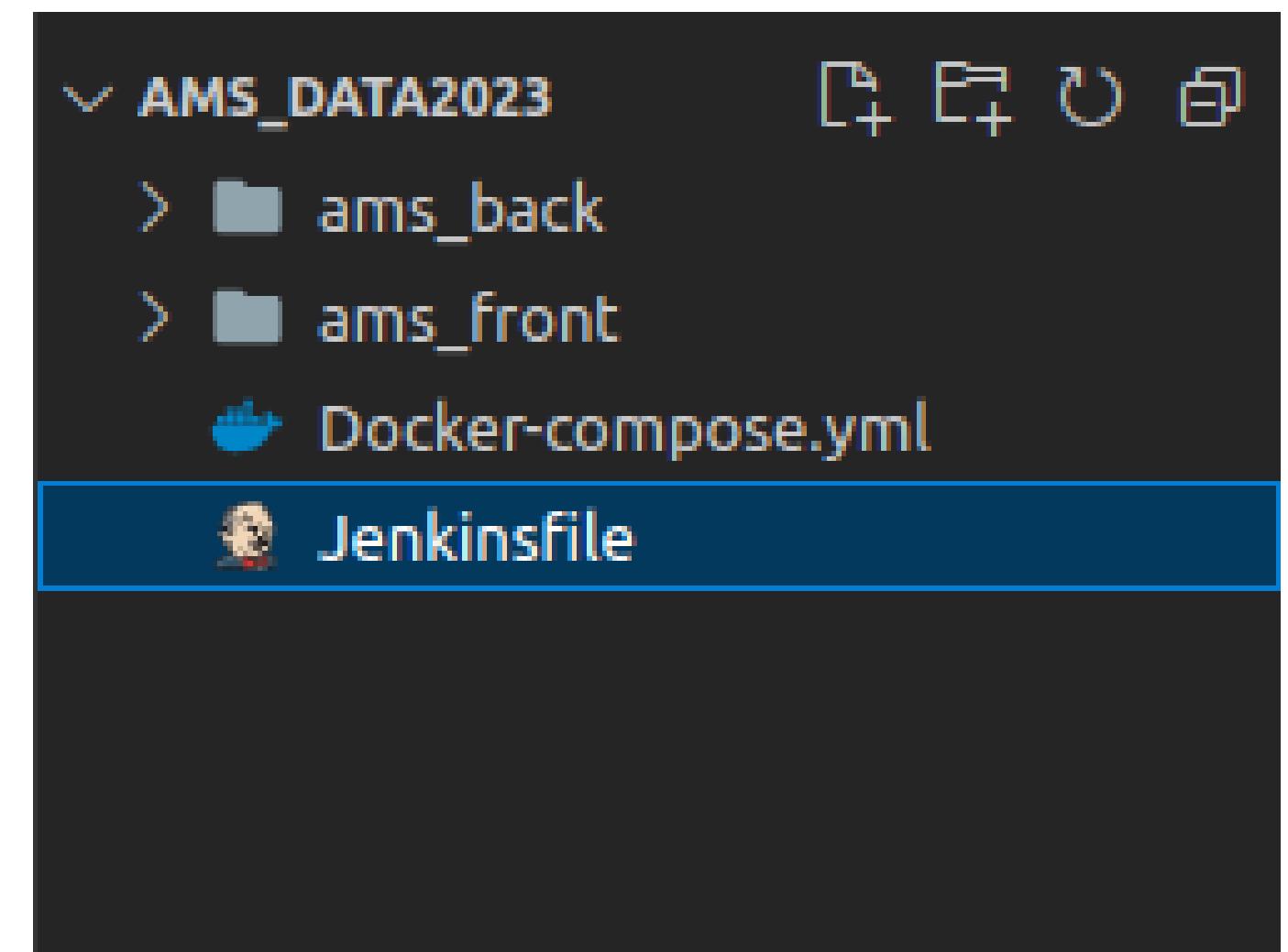
6. **Sauvegarder la configuration du job** : Enregistrez la configuration du job pour appliquer les changements.

Ces étapes permettent aux utilisateurs de créer un "Pipeline job" dans Jenkins, permettant l'automatisation et l'orchestration de diverses tâches, de la construction du code au déploiement, en utilisant une approche de pipeline scriptée définie dans un Jenkinsfile.



JENKINSFILE

Nous allons maintenant créer un fichier nommé « Jenkinsfile » dans la racine de notre projet, à l'emplacement où se trouvent les dossiers "ams_front" et "ams_back".





JENKINSFILE

Dans la section "script_path", nous allons indiquer le chemin de notre fichier Jenkinsfile.

The screenshot shows the Jenkins Pipeline configuration screen. At the top, there is a label 'Script Path' with a question mark icon. Below it is a text input field containing the text 'Jenkinsfile'. Underneath this, there is a checkbox labeled 'Lightweight checkout' which is checked, indicated by a blue checkmark icon. At the bottom left, there is a link labeled 'Pipeline Syntax'.



JENKINSFILE

Dans la section "script_path", nous allons indiquer le chemin de notre fichier Jenkinsfile.

The screenshot shows the Jenkins Pipeline configuration screen. At the top, there is a label 'Script Path' with a question mark icon. Below it is a text input field containing the text 'Jenkinsfile'. Underneath this, there is a checkbox labeled 'Lightweight checkout' which is checked, indicated by a blue checkmark icon. At the bottom of the configuration area, there is a link labeled 'Pipeline Syntax'.



JENKINSFILE

Structure de Jenkinsfile

Ce script représente un pipeline Jenkins organisé en étapes distinctes, chacune étant une phase spécifique du processus de déploiement ou de build d'une application.

```
pipeline {  
    agent any  
  
    stages{  
        stage('#####') {  
            steps {  
                sh '#####'  
                sh '#####'  
            }  
        }  
        stage('#####') {  
            steps {  
                sh '#####'  
                sh '#####'  
            }  
        }  
    }  
}
```



JENKINSFILE

Structure de Jenkinsfile

1. **pipeline { }** : Cela marque le début du script de pipeline Jenkins.
2. **agent any** : Cette ligne spécifie l'agent sur lequel les étapes du pipeline seront exécutées. En utilisant "any", le pipeline peut être exécuté sur n'importe quel agent disponible.
3. **stages { }** : Cette section déclare l'ensemble des étapes du pipeline.
4. **stage('#####') { }** : Chaque "stage" représente une phase spécifique du pipeline. Les "#####" doivent être remplacés par des noms descriptifs pour chaque étape, par exemple : "Compilation", "Tests unitaires", "Déploiement", etc.



JENKINSFILE

Structure de Jenkinsfile

5. **steps { }** : À l'intérieur de chaque "stage", il y a une section "steps" où les actions à effectuer pour cette étape sont définies.
6. **sh #####** : Cette instruction "sh" représente l'exécution d'une commande shell spécifique dans cette étape. Les "#####" doivent être remplacés par les commandes réelles que vous souhaitez exécuter, telles que des commandes de compilation, d'exécution de tests, ou de déploiement.
7. **Répétition des stages et des steps** : Vous pouvez ajouter autant de stages et de steps que nécessaire pour refléter les différentes phases du processus de déploiement ou de build de votre application.



JENKINSFILE

Structure de Jenkinsfile

En résumé, ce script de pipeline Jenkins organise le flux de travail en étapes distinctes, permettant une exécution séquentielle des tâches spécifiques à chaque phase du processus de développement, test ou déploiement d'une application.



JENKINSFILE

Jenkinsfile de notre projet

Ce script représente un pipeline Jenkins organisé en différentes étapes pour la création d'images Docker et le déploiement via Docker Compose.

```
Jenkinsfile x
Jenkinsfile
1 pipeline {
2
3     environment {
4         springF = "ams_back"
5         angularF = "ams_front"
6     }
7
8     agent any
9
10    stages{
11        stage('Création d"une image docker-spring') {
12            steps {
13                sh 'docker build -t ams-2023 ${springF}'
14            }
15        }
16
17        stage('Création d"une image docker-angular') {
18            steps {
19                sh 'docker build -t ams-2023-front ${angularF}'
20            }
21        }
22
23    }
24
25
26
27
28
29
30 }
```



JENKINSFILE

Jenkinsfile de notre projet

Voici une explication détaillée de chaque partie :

- **environment { ... }** : Cette section définit des variables d'environnement pour le pipeline. Ici, "springF" est défini comme "ams_back" et "angularF" comme "ams_front". Ces variables seront utilisées pour référencer les noms des dossiers des projets Spring et Angular.
- **agent any** : Cela spécifie que le pipeline peut être exécuté sur n'importe quel agent disponible.
- **stages { ... }** : Cette partie déclare les différentes étapes du pipeline.



JENKINSFILE

Jenkinsfile de notre projet

- **stage('Création d'une image docker-spring') { ... } :** Cette étape crée une image Docker pour le projet Spring.
 - **steps { ... } :** Cette section définit les actions à effectuer dans cette étape. Ici, une commande shell est utilisée pour construire une image Docker nommée "ams-2023" à partir du dossier référencé par la variable "springF".
- **stage('Création d'une image docker-angular') { ... } :** Cette étape crée une image Docker pour le projet Angular.
 - **steps { ... } :** Comme précédemment, cette section contient une commande pour construire une image Docker nommée "ams-2023-front" à partir du dossier référencé par la variable "angularF".



JENKINSFILE

Jenkinsfile de notre projet

```
34 |         stage('Docker-compose') {  
35 |             steps {  
36 |                 sh 'docker-compose -f Docker-compose.yml up -d'  
37 |             }  
38 |         }  
39 |     }  
40 | }  
41 | }  
42 | }  
43 | }  
44 | }  
45 | }  
46 | }
```

- **stage('Docker-compose') { ... }** : Cette étape exécute le déploiement via Docker Compose.
 - **steps { ... }** : Une commande shell est utilisée pour démarrer les services décrits dans le fichier "Docker-compose.yml" à l'aide de la commande "docker-compose up -d".



JENKINSFILE

Jenkinsfile de notre projet

En résumé, ce script de pipeline Jenkins automate les étapes de construction des images Docker pour les projets Spring et Angular, puis lance le déploiement à l'aide de Docker Compose pour exécuter les services correspondants.



JENKINSFILE

Jenkinsfile de notre projet

Ensuite, dans notre job Jenkins, nous allons déclencher un processus de construction en cliquant sur "Build Now"

The screenshot shows the Jenkins Pipeline interface for the 'AMS-DATA2023' project. The top navigation bar includes links for 'Dashboard' and 'AMS-DATA2023'. Below the header is a sidebar with the following options: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, and Pipeline Syntax. At the bottom of the sidebar is a 'Build History' section with a 'trend' dropdown set to 'trend'. A search bar at the bottom allows filtering of builds.





JENKINSFILE

Jenkinsfile de notre projet

Puis examiner les résultats de cette exécution.

The screenshot shows the Jenkins dashboard for the project "AMS-DATA2023". The top navigation bar includes the Jenkins logo, a search bar, and user information for "sip-academy". The main content area displays the project name "AMS-DATA2023" with a status indicator (green circle with a white minus sign). On the left, a sidebar provides options like "Status", "Changes", "Build Now", "Configure", "Delete Pipeline", "Full Stage View", "Rename", and "Pipeline Syntax". Below this is the "Stage View" section, which shows four stages: "Declarative: Checkout SCM" (2s), "Création d'une image docker-spring" (2s), "Création d'une image docker-angular" (5s), and "Docker-compose" (1s). The "Build History" section shows a single build (#5) from December 28 at 18:20 with "No Changes". Average stage times are listed as 2s, 2s, 5s, and 1s respectively.



NGROK / WEBHOOK-TRIGGERS

ngrok

ngrok peut être considéré comme une application multiplateforme qui peut exposer notre hôte local à Internet. Il héberge le serveur de développement local sur son sous-domaine ngrok.com. Nous n'avons donc pas besoin d'adresse IP publique à ce sujet.



NGROK / WEBHOOK-TRIGGERS

ngrok

- Pour télécharger ngrok, vous pouvez visiter le lien **<https://ngrok.com/download>**.
- Après le téléchargement, suivez les étapes ci-dessous pour configurer le tunnel.

\$ ngrok http 8080

```
dhiasaria@dhiasaria-HP-Notebook: ~ (ctrl+c to quit)
ngrok
Add OAuth and webhook security to your ngrok (its free!): https://ngrok.com/free

Session Status
Account
Update
Version
Region
Latency
Web Interface
Forwarding
Connections

online
dhiasaria (Plan: Free)
update available (version 3.2.1, Ctrl-U to update)
3.1.0
Europe (eu)
75ms
http://127.0.0.1:4040
https://d710-196-235-60-38.eu.ngrok.io -> http://l

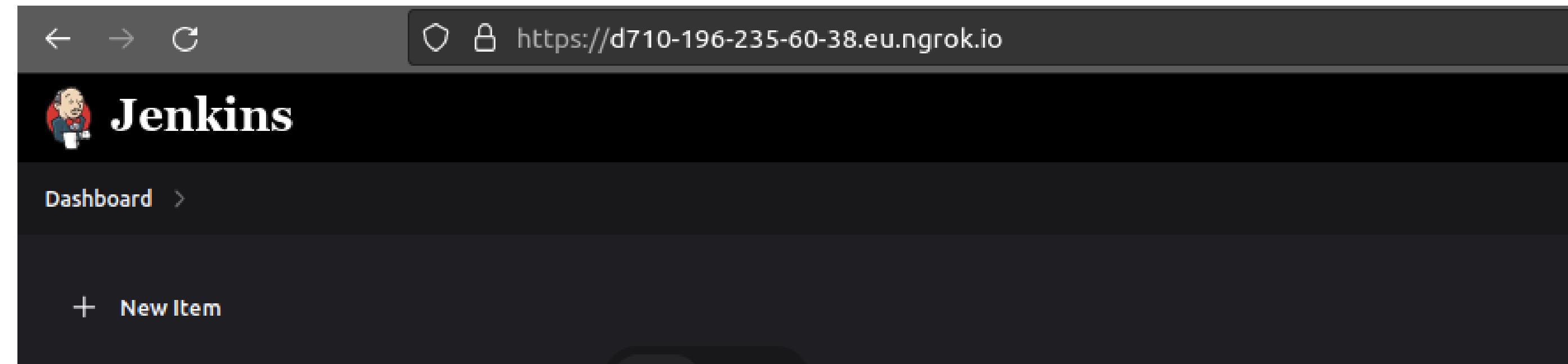
ttl     opn      rt1      rt5      p50      p90
0       0        0.00     0.00     0.00     0.00
```



NGROK / WEBHOOK-TRIGGERS

ngrok

On peut maintenant accéder à notre serveur jenkins avec ce url





NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Le rôle des Webhook Triggers est de permettre à des événements externes, tels que des mises à jour de code source dans un référentiel (comme GitHub, Bitbucket, etc.), de déclencher automatiquement des builds ou des workflows dans Jenkins. Ces déclencheurs webhook sont configurés pour surveiller des événements spécifiques, et lorsque l' événement correspondant se produit, Jenkins est notifié pour lancer des actions préconfigurées, telles que lancer un build, déclencher un pipeline, etc. Cela permet une intégration et une automatisation continues, réduisant ainsi la nécessité de déclencher manuellement des builds ou des workflows dans Jenkins.



NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau bitbucket

Pour configurer un webhook dans Bitbucket pour déclencher des builds dans Jenkins à chaque modification du code source, suivez ces étapes :

1. Allez sur votre repository Bitbucket.
2. Cliquez sur "Repository Settings"

The screenshot shows the Bitbucket repository settings interface for the 'springdata-master' repository. The left sidebar lists various repository management options: Source, Commits, Branches, Pull requests, Pipelines, Deployments, Jira issues, Security, Downloads, and Repository settings. The 'Source' option is currently selected, indicated by a highlighted background. The main content area displays basic repository information: owner 'dhia eddine saria', repository name 'springdata-master', and a brief description: 'Here's where you'll find this repository's source files. To give your users an idea of what's in your repository.' Below this, there are dropdown menus for 'master' and 'Files', a 'Filter files' input field, and a search icon. The file list table has columns for Name, Size, Last commit, and Message.

Name	Size	Last commit	Message
.mvn		3 days ago	first commit
.settings		3 days ago	first commit
src		3 days ago	first commit
.classpath	2.05 KB	3 days ago	first commit
.gitignore	624 B	3 days ago	first commit



NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau bitbucket

3. Dans le menu latéral gauche, sélectionnez "Webhooks".

WORKFLOW

Branch restrictions

Branching model

Merge strategies

Webhooks

Links





NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau bitbucket

4. Cliquez sur "Add webhook"

dhia eddine saria / springdata-master / springdata-master / Repository settings
Webhooks

Webhooks allow you to extend what Bitbucket does when the repository changes (for example, merged).

To learn more about how webhooks work, check out the documentation.

Add webhook





NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau bitbucket

5. Remplissez les détails du webhook :

- **Title:** Entrer le titre de webhook.
- **URL :** Entrez l'URL du job Jenkins que vous souhaitez déclencher. Habituellement, c'est quelque chose comme : `http://votre-ip-ou-domaine/bitbucket-hook/`. (On va mettre l'url fournie par ngrok)
- **Méthode :** Choisissez le type de méthode (souvent Push).
- **6. Enregistrez le webhook.**

To learn more about how webhooks work, check out the [documentation](#).

Title *

webhook

URL *

`https://d710-196-235-60-38.eu.ngrok.io/bitbucket-hook/`

Status

Active

Inactive webhooks don't trigger requests.

SSL/TLS

Skip certificate verification

Untrusted or self-signed certificates may not be secure. [Learn more](#)

Triggers

Repository

Push

Issue

Created

Pull request

Created



NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau bitbucket

- 6. Enregistrez le webhook.

To learn more about how webhooks work, check out the [documentation](#).

Title *

webhook

URL *

<https://d710-196-235-60-38.eu.ngrok.io/bitbucket-hook/>

Status

Active

Inactive webhooks don't trigger requests.

SSL/TLS

Skip certificate verification

Untrusted or self-signed certificates may not be secure. [Learn more](#)

Triggers

Repository

Push

Issue

Created

Pull request

Created





NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau jenkins

Plugins

bitbucket

Install	Name ↓	Released
<input type="checkbox"/>	Bitbucket Branch Source 805.v7f97d29dc0f5 bitbucket Source Code Management Allows to use Bitbucket Cloud and Bitbucket Server as sources for multi-branch projects. It also provides the required connectors for Bitbucket Cloud Team and Bitbucket Server Project folder (also known as repositories auto-discovering).	6 mo 29 days ago
<input type="checkbox"/>	Bitbucket Pipeline for Blue Ocean 1.27.5.1 BlueOcean Bitbucket pipeline creator	4 mo 27 days ago
<input type="checkbox"/>	Generic Webhook Trigger 1.88.2 notification github webhook Build Parameters gitlab Build Triggers bitbucket bitbucket-server jira Can receive any HTTP request, extract any values from JSON or XML and trigger a job with those values available as variables. Works with GitHub, GitLab, Bitbucket, Jira and many more.	1 mo 0 days ago





NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau jenkins

- 1. Accès au projet Jenkins :** Assurez-vous que vous avez les autorisations nécessaires dans Jenkins pour créer et gérer des jobs.
- 2. Installation du plugin Jenkins Bitbucket :** Si ce n'est pas déjà fait, installez le plugin Bitbucket sur Jenkins. Allez dans "Manage Jenkins" > "Manage Plugins", puis recherchez et installez le plugin Bitbucket.



NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau jenkins

3. Configuration du job Jenkins : Assurez-vous que le job Jenkins que vous souhaitez déclencher via le webhook est configuré correctement pour être déclenché par Bitbucket. Cela implique d'utiliser les plugins « Bitbucket Branch Source » et « Generic Webhook Trigger ».

Build Triggers

Build when a change is pushed to BitBucket
Override Repository URL or Path
<https://dhiasaria5@bitbucket.org/dhiasaria5/springdata-master-local.git>

Build on branch created

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Quiet period ?

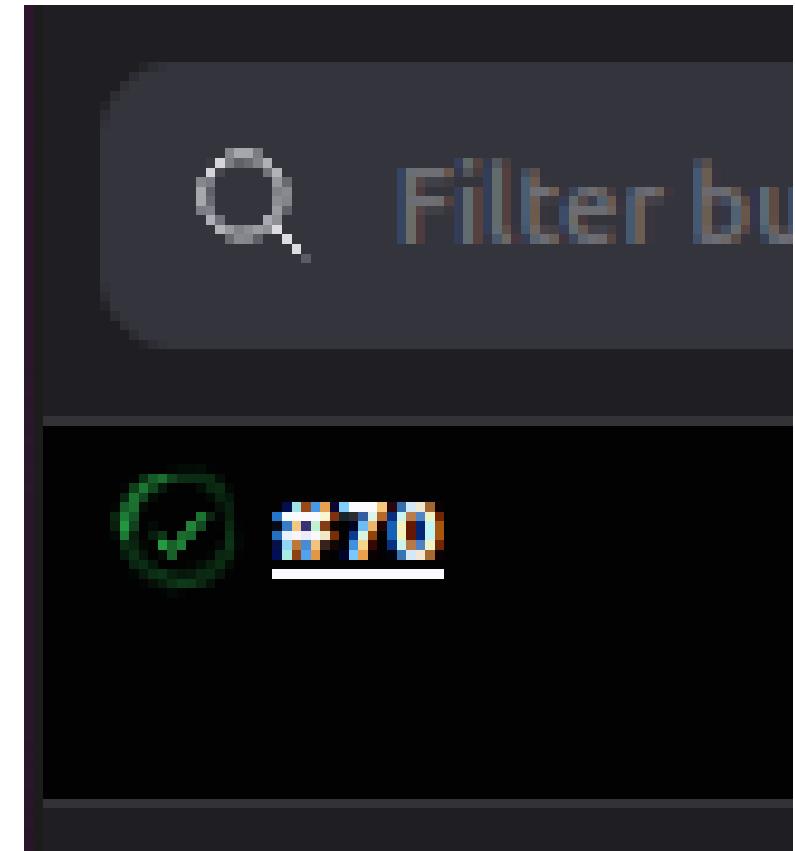


NGROK / WEBHOOK-TRIGGERS

webhook-triggers

Configuration au niveau jenkins

4. Test du webhook : Apportez une modification à votre référentiel Bitbucket et observez si cela déclenche un build dans Jenkins.





Smart IT Partner, Editeur de logiciels

Nous proposons des solutions sur mesure pour nos partenaires avec différentes technologies et frameworks web et mobile principalement Spring Boot, Symfony et Angular/React aussi bien des solutions CMS tel que Prestashop et wordpress.

Nos principaux domaines de développement touchent principalement tout ce qui est E-Commerce, immobilier ainsi que le secteur transport et logistique.



SIP Academy, centre de formation

Notre centre de formations propose des sessions de formations en différentes disciplines tel que Agile Scrum, Java OCP, Spring Boot, Angular, Devops....

Récemment, SIP academy s'est lancé dans des cursus de formations d'une durée allant d'1 mois à 3 mois (Bootcamp sur mesure) pour le compte de ses clients en Tunisie et en France



A black and white photograph of four people in an office environment. A man in a light-colored shirt is laughing heartily on the left. In the center, a man's back is to the camera. To his right, a woman in a polka-dot blouse is smiling. On the far right, another woman wearing glasses and a striped shirt is laughing. They appear to be in a meeting or a break room.

Merci pour votre attestation

www.smart-it-partner.com

www.sip-academy.com

