**Configuration du jenkins en mode job**

Table des matières

[**1-** **Prérequis** 1](#_Toc71705341)

[**2-** **Connexion au serveur jenkins** 1](#_Toc71705342)

[**3-** **Préconfiguration du gradle** 1](#_Toc71705343)

[**4-** **Configuration d’un job** 3](#_Toc71705344)

[**5-** **Configuration de gitHUB pour la partie webhook** 12](#_Toc71705345)

1. **Prérequis**

* Le serveur Nexus 192.168.1.200 doit être installé et démarré.
* Le serveur jenkins 192.168.1.202 doit être installé et démarré
* Le repot github disponible pour le git clone

1. **Connexion au serveur jenkins**

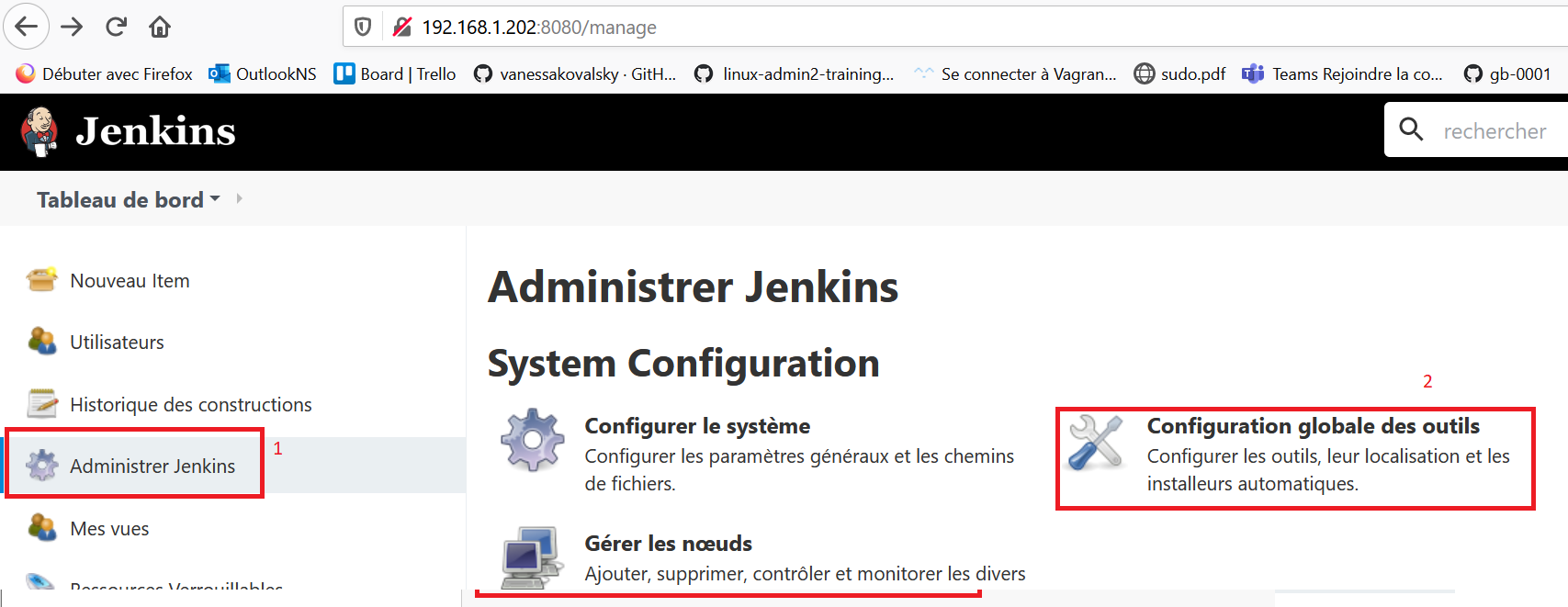
Se connecter au serveur jenkins par le navigateur web sur l’url **http://192.168.1.202:8080**

Le mot passe doit avoir été saisie à la page suivante :

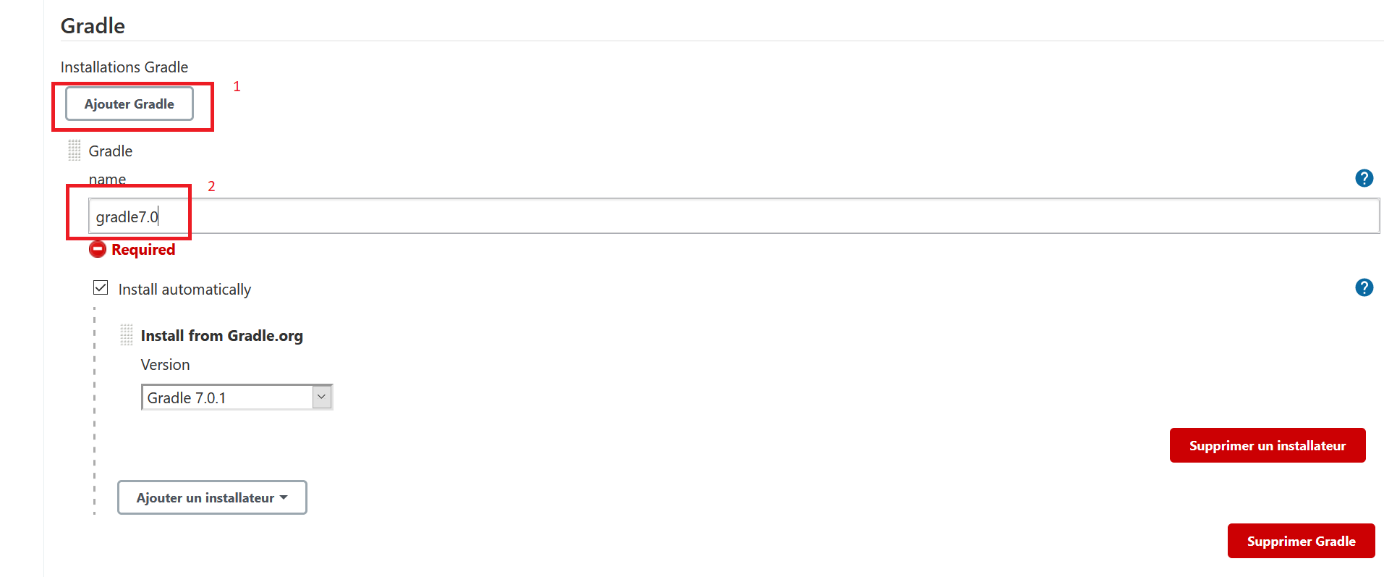
Choisir install jenkins ci-dessous :

1. **Pré configuration du gradle**

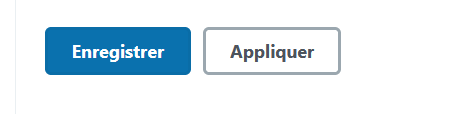
Configurer gradle cliquez sur **administrer jenkins** puis **configuration globale des outils**



Dans la section Gradle cliquez sur **ajouter Gradle** puis saisir **Gradle7.0**



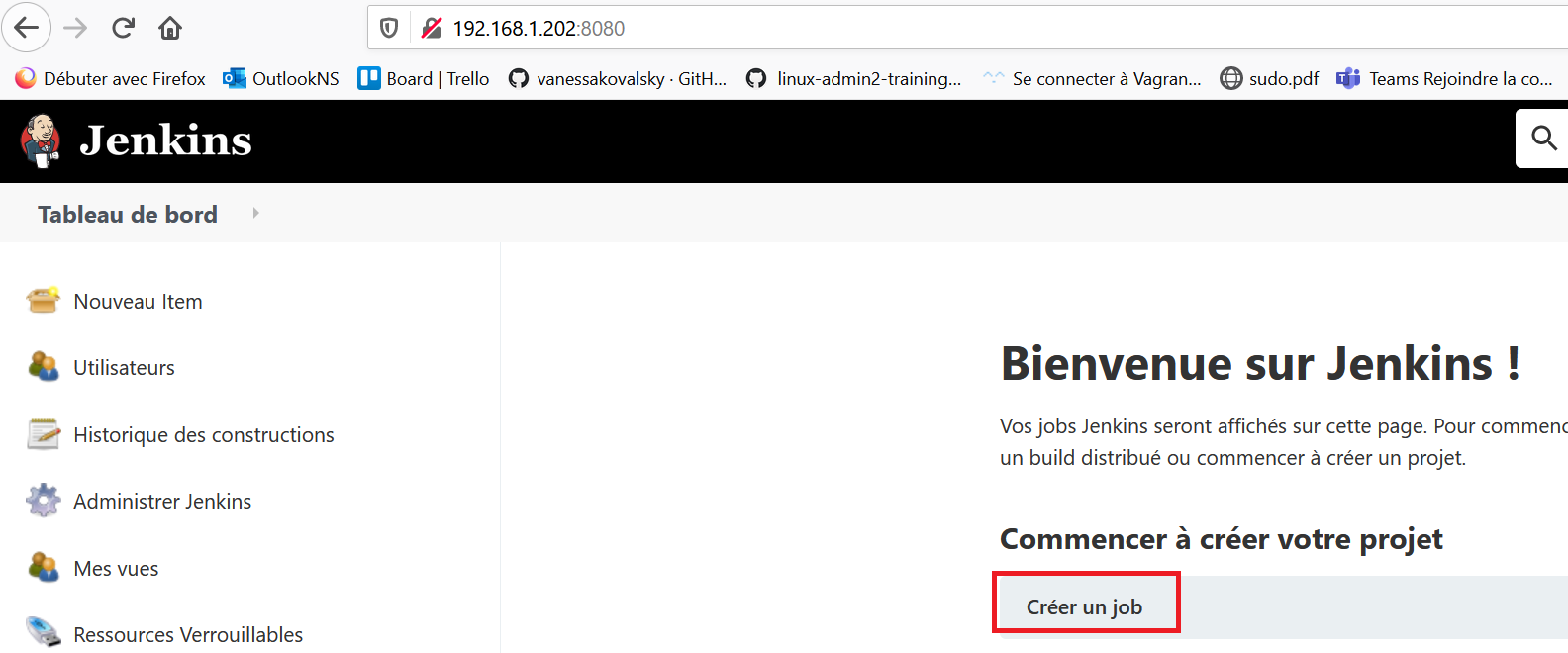
Puis **Appliquer** et **enregistrer**



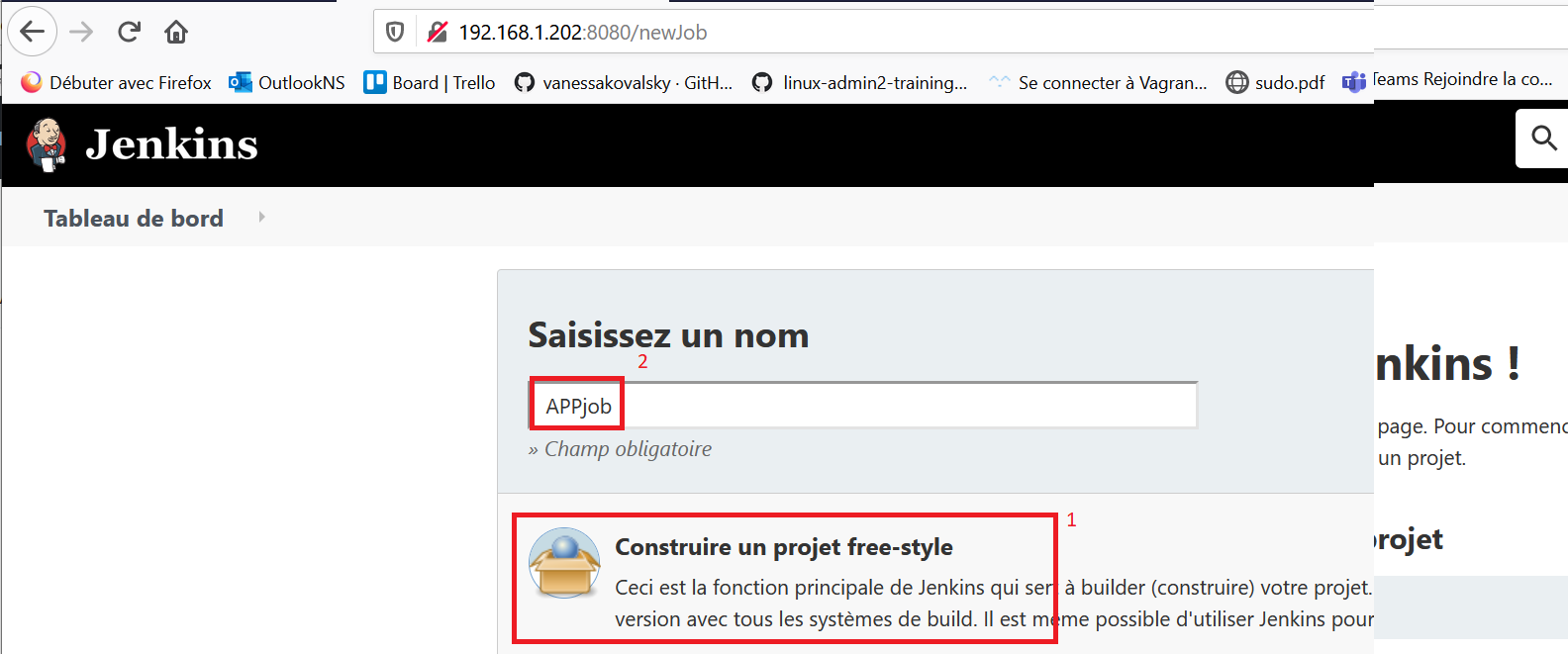
1. **Configuration d’un job**

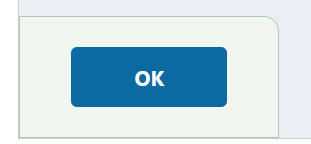
Cliquez sur **Nouveau Item**

Cliquez sur **créer un job**



Cliquez sur **construire un projet free-style** puis en saisir **APP\_job** :



Puis 

Détails de l’image ci-dessous avec en capture les éléments :

1 - Pour github project Saisir l’url pour le git clone ou se situe le package

2 – Pour git Saisir l’url pour le git clone ou se situe le package

3 – saisir \*/master pour prendre le repos principale

4 – cocher githubhook

5 – Ajouter une étape de build

6 – sélectionner executer un script shell

7 – saisir la commande de copie du fichier de version

8 – ajouter une nouvelle étape build

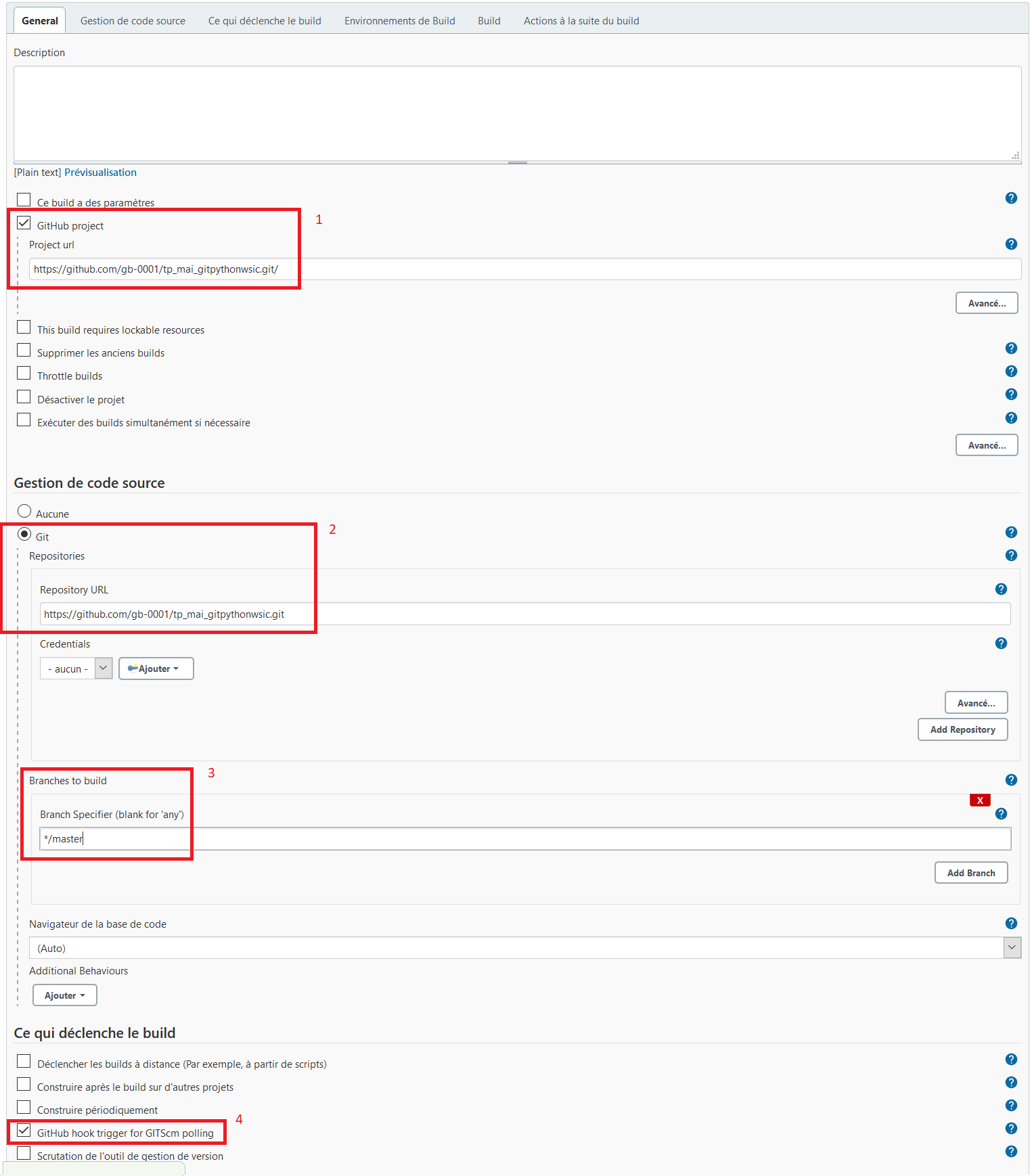
9 - sélectionner invoke gradle script

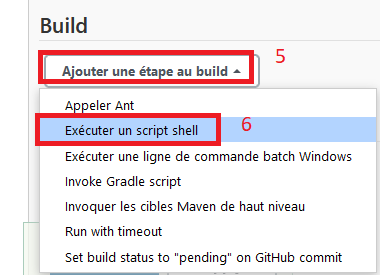
10 - sélectionner le modèle créer dans la section 3 préparation du gradle

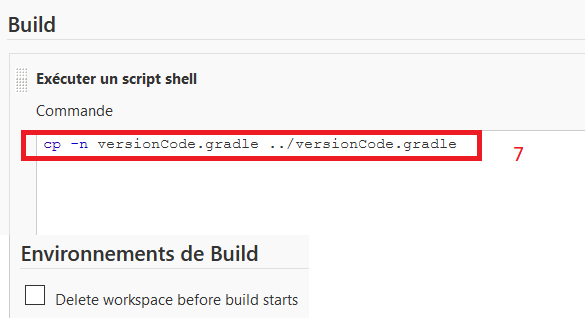
11 – saisir la task write

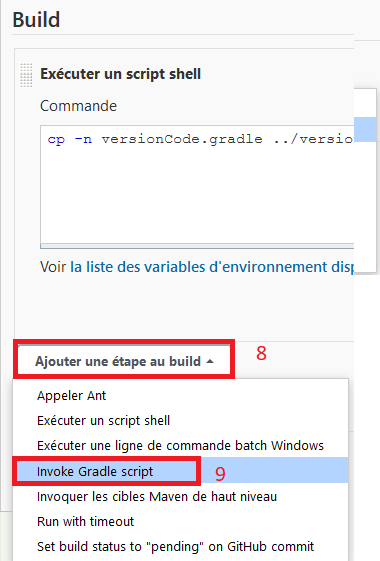
*Infos :*

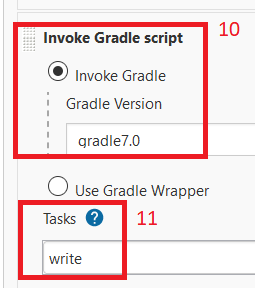
*Le build.gradle doit être a la racine du git clone pour être pris en compte*

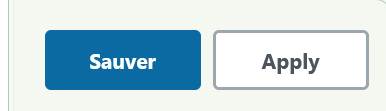


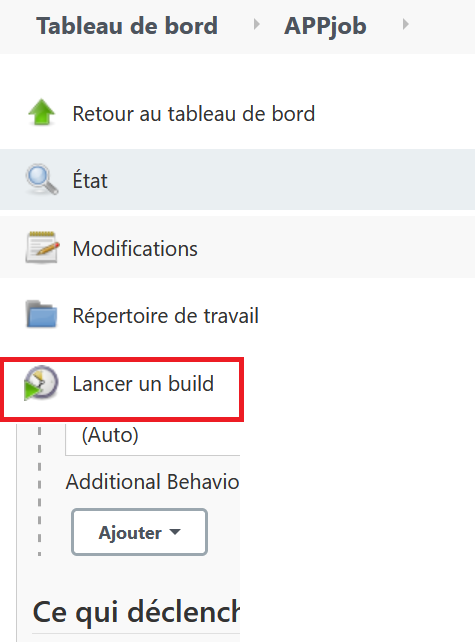






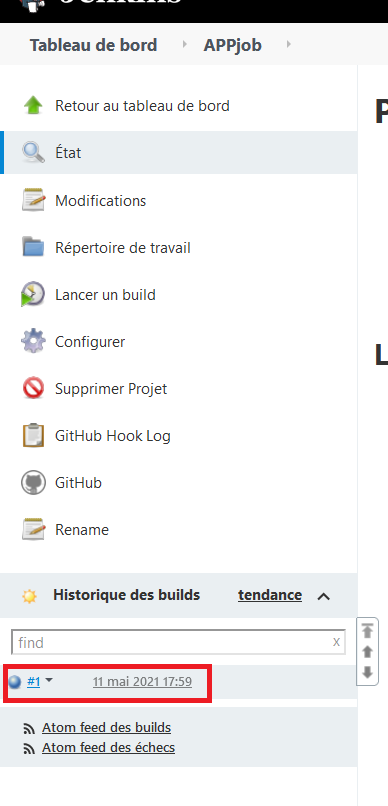


Puis **APPLY** et **sauver** 

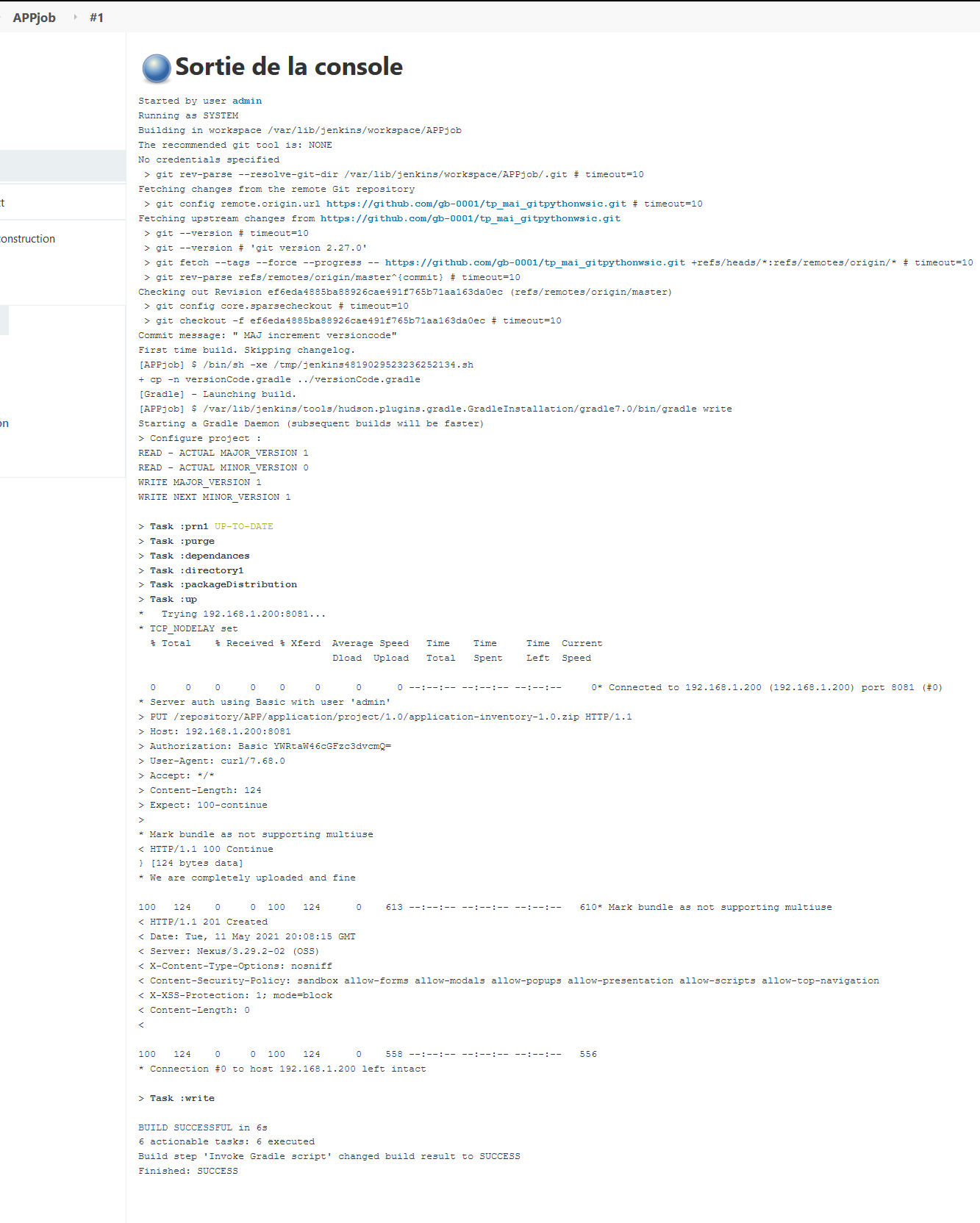
Lancer le build :

GENERATION DU 1er BUILD

Cliquez sur le build en cours **Console output** pour voir le retour ci-dessous :

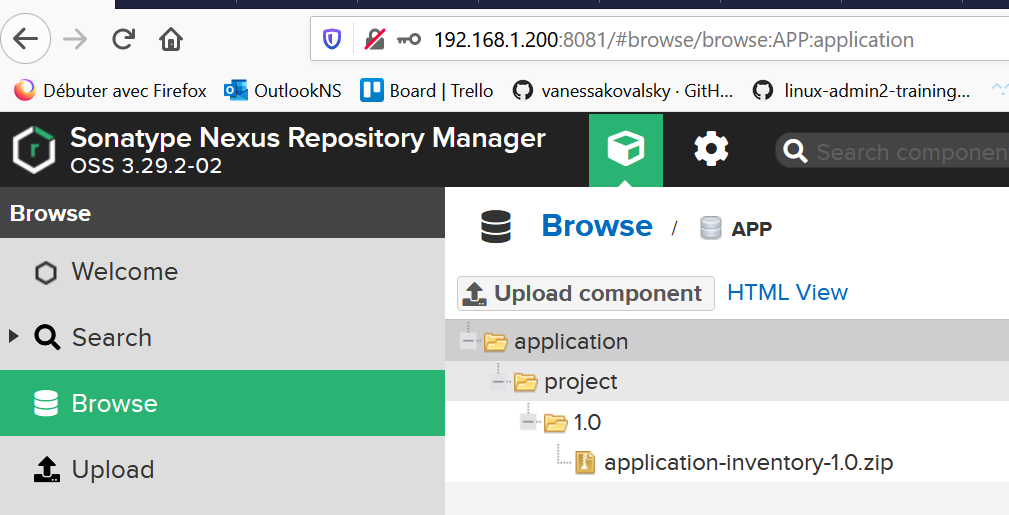


Le job doit être en success et sur le nexus le package est présent en version 1.0



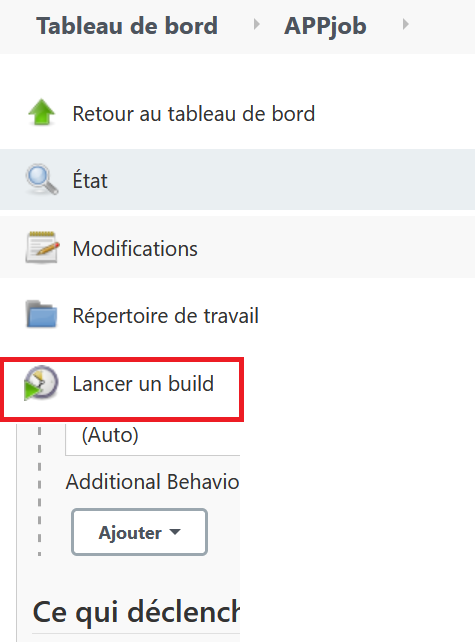
Vérifier sur la page web du nexus se connecté à l’url <http://192.168.1.200:8081>

Le retour ci-dessous de l’envoi vers le nexus

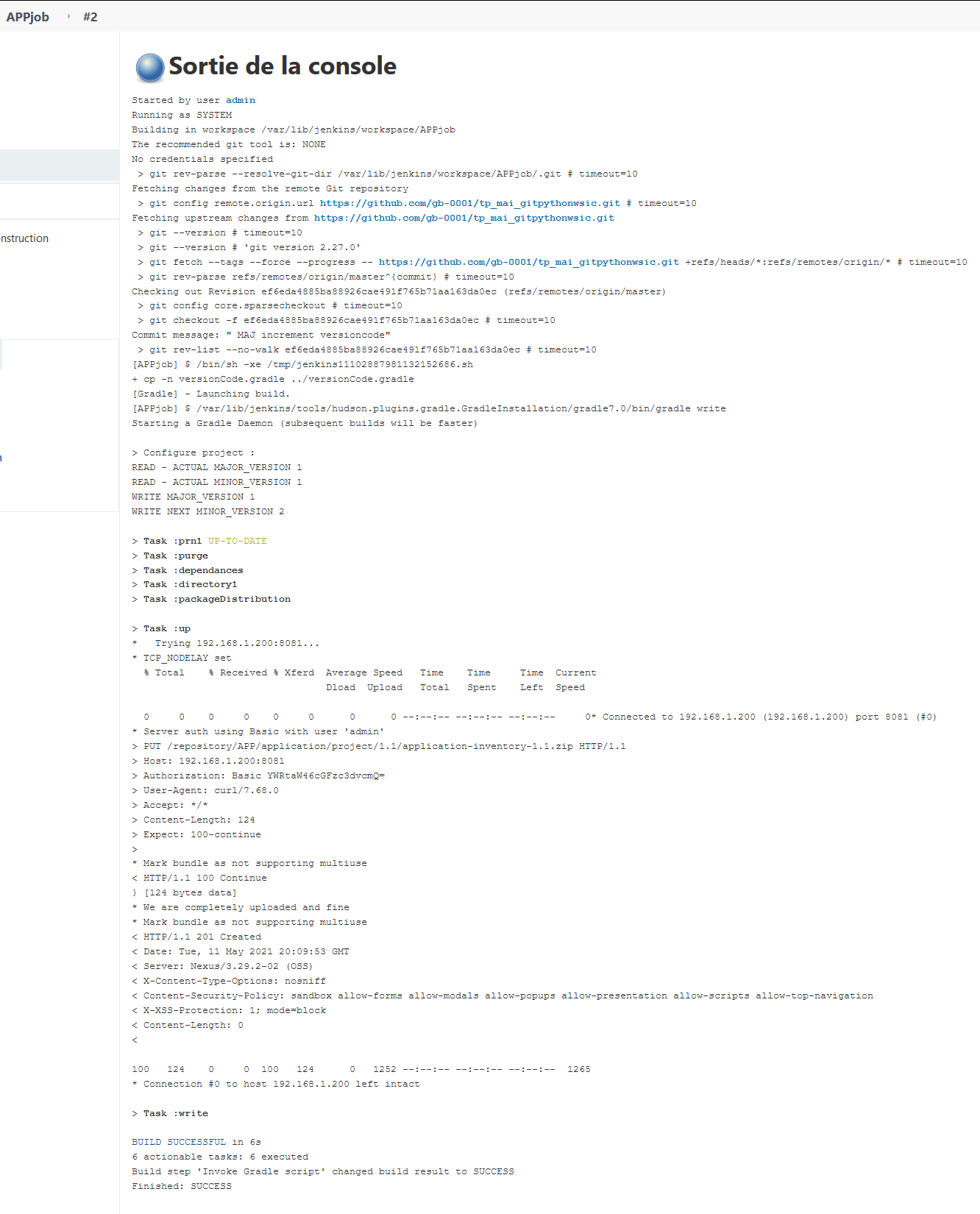


GENERATION DU 2ND BUILD POUR VERIFIER L’INCREMENTATION

Cliquez à nouveau sur Lancer un build

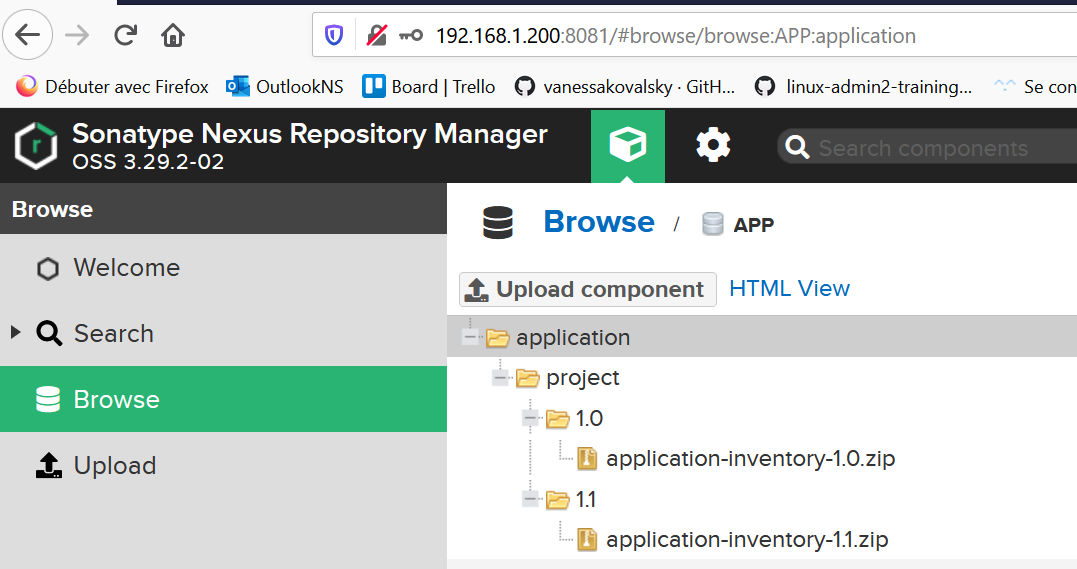


Retour de la generation du build



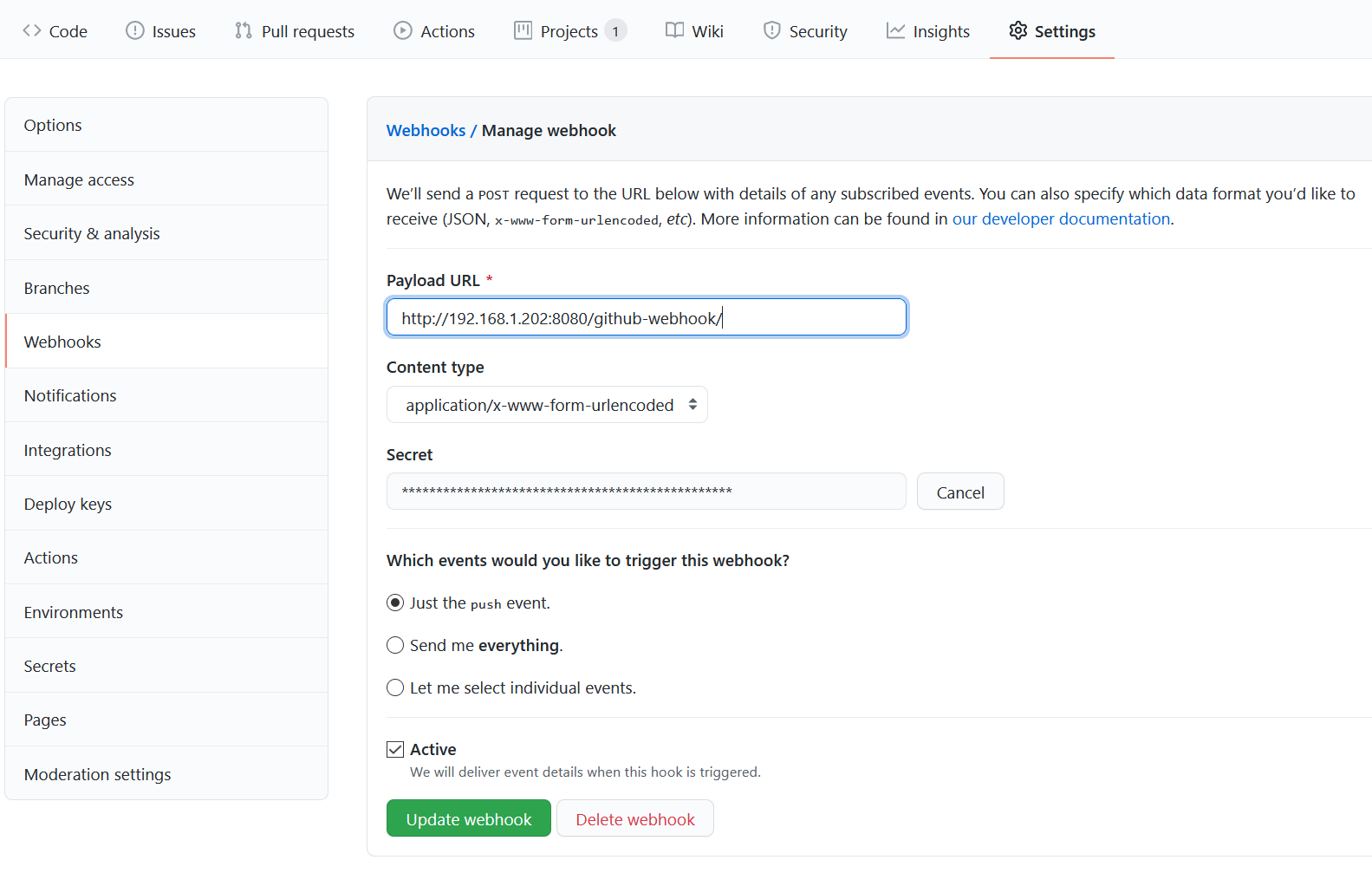
Vérifier sur la page web du nexus se connecté à l’url <http://192.168.1.200:8081>

Le retour ci-dessous de l’envoi vers le nexus



1. **Configuration de gitHUB pour la partie webhook**

* Se connecter au compte github et cliquer sur le repo tp\_mai\_gitpythonwsic
* Saisir dans un nouvelle onglet <https://github.com/settings/tokens> et générer un token à recopier dans le setting ci-dessous du webhook
* Cliquez sur setting puis webhook dans secret coller le token
* Dans payload url saisir l’adresse du serveur jenkins avec la chaine /github-webhook/ en fin d’url serveur <http://192.168.1.202:8080/github-webhook/>
* Laisser juste push event et cliquez sur **update webhook**



Non fonctionnel car pointe sur l’adresse local. Nécessite d’avoir une ouverture flux pour déclencher le job sur le serveur jenkins en 192.168.1.202 mais on voit les tentive de connexion