

Orchestrator

lundi 23 janvier 2023 09:26

Formateur : Loup FORMENT

Orchestration informatique

15 langues ▾

[Article](#) [Discussion](#)

[Lire](#) [Modifier](#) [Modifier le code](#) [Voir l'historique](#)

L'**orchestration** décrit le processus automatique d'organisation, de coordination, et de gestion de systèmes informatiques complexes, de **middleware** et de services.

On en parle souvent comme ayant une intelligence inhérente ou même un contrôle implicitement autonome, mais il s'agit plutôt d'aspirations ou d'analogies plutôt que de descriptions techniques. En réalité, l'*orchestration* est largement le résultat de l'**automatisation** de systèmes qui déplient des éléments de **régulation**.

On parle souvent de cet usage de l'*orchestration* dans le contexte de la **virtualisation**, de la fourniture de services et des **centres de données dynamiques**.

Un usage quelque peu différent désigne le processus de coordination d'un échange d'information à travers l'interaction de **services web**.

Principaux outils -> **Jenkins/Git Action**

<https://www.jenkins.io/>

<https://github.com/features/actions>

Documentation :

<https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04-fr>

<https://doc.ubuntu-fr.org/jenkins>

Installer clé de dépôt :

```
vel@vel-Precision-3650-Tower:~$ wget -q sudo http://pkg.jenkins-ci.org/d
```

Rajouter le dépôt dans le fichier

2.2 Modifier le fichier des sources

Ouvrez le fichier `/etc/apt/sources.list` avec les [droits d'administration](#)

Un certain nombre de lignes sont déjà présentes, elles correspondent aux dépôts principaux d'Ubuntu.

Un exemple : pour ajouter le dépôt **deb Restricted**, vous rajouterez à la fin de votre fichier « `sources.list` » les lignes :

```
deb http://fr.archive.ubuntu.com/ubuntu/ xenial main restricted  
#deb-src http://fr.archive.ubuntu.com/ubuntu/ xenial main restricted
```

Ici, vous remarquez que les « # » placés devant les lignes des dépôts-sources commentent la ligne et donc désactivent ces dépôts..

```
vel@vel-Precision-3650-Tower:~$ cat /etc/apt/sources.list  
#deb cdrom:[Ubuntu 22.04.1 LTS _Jammy Jellyfish_ - Release amd64 (20220809.1)]/ jammy main restricted  
  
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to  
# newer versions of the distribution.  
deb http://fr.archive.ubuntu.com/ubuntu/ jammy main restricted  
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy main restricted  
  
## Major bug fix updates produced after the final release of the  
## distribution.  
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-updates main restricted  
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-updates main restricted  
  
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu  
## team. Also, please note that software in universe WILL NOT receive any  
## review or updates from the Ubuntu security team.  
deb http://fr.archive.ubuntu.com/ubuntu/ jammy universe  
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy universe  
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-updates universe  
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-updates universe  
  
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu  
## team, and may not be under a free licence. Please satisfy yourself as to  
## your rights to use the software. Also, please note that software in  
## multiverse WILL NOT receive any review or updates from the Ubuntu
```

```

## security team.
deb http://fr.archive.ubuntu.com/ubuntu/ jammy multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy multiverse
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-updates multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-backports main restricted universe multiverse

deb http://security.ubuntu.com/ubuntu jammy-security main restricted
# deb-src http://security.ubuntu.com/ubuntu jammy-security main restricted
deb http://security.ubuntu.com/ubuntu jammy-security universe
# deb-src http://security.ubuntu.com/ubuntu jammy-security universe
deb http://security.ubuntu.com/ubuntu jammy-security multiverse
# deb-src http://security.ubuntu.com/ubuntu jammy-security multiverse
deb http://pkg.jenkins-ci.org/debian binary/

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.

```

Etape suivante : installer Java

sudo apt install default-jdk

```

vel@vel-Precision-3650-Tower:~$ sudo apt install default-jdk
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  fonts-dejavu gir1.2-keybinder-3.0 kded5 keditbookmarks kio konsole-kpart kpackagelauncherqml kp
libflashrom1 libftdi1-2 libhfstspell11 libkeybinder-3.0-0 libkf5archive5 libkf5attica5 libkf5b
libkf5bookmarks5 libkf5completion-data libkf5completions5 libkf5crash5 libkf5declarative-data lib
libkf5doctools5 libkf5globalaccel-bin libkf5globalaccel-data libkf5globalaccelcs libkf5globalacc
libkf5iconthemes-bin libkf5iconthemes-data libkf5iconthemes5 libkf5itemviews-data libkf5itemview
libkf5jobwidgets-data libkf5jobwidgets5 libkf5kiocore5 libkf5kionlms5 libkf5kiow
libkf5kirigami2-5 libkf5newstuff-data libkf5newstuff5 libkf5newstuffcore5 libkf5notifyconfig-d
libkf5package-data libkf5package5 libkf5parts-data libkf5parts-plugins libkf5parts5 libkf5pty-d
libkf5quickaddons5 libkf5solids5 libkf5solid5-data libkf5sonnet5-data libkf5sonnetcore5 libkf5son
libkf5syndication5abi1 libkf5textwidgets-data libkf5textwidgets5 libkf5xmlgui-bin libkf5xmlgui-o
libptytty0 libqt5qmlworkerscript5 libqt5quickcontrols2-5 libqt5quicktemplates2-5 libutempter0 li
linux-headers-5.15.0-43 linux-headers-5.15.0-43-generic linux-image-5.15.0-43-generic linux-mod
linux-modules-extra-5.15.0-43-generic python3-pbr qml-module-org-kde-kirigami2
qml-module-org-kde-qquickcontrolsaddons qml-module-org-kde-newstuff qml-module-qtgraphicaleffect
qml-module-qtqml-models2 qml-module-qtquick-controls2 qml-module-qtquick-layouts qml-module-qtqu
qml-module-qtquick-window2 qml-module-qtquick2 sonnet-plugins
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
  ca-certificates-java default-jdk-headless default-jre default-jre-headless java-common libatk-w
libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb-
libxt-dev openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless x11proto-
xorg-sgml-doctools xtrans-dev

```

Restart le service une fois Java installé : il peut dire qu'une erreur est détecté avec Jenkins, mais c'est un **faux positif**

```

E: Sub-process /usr/bin/dpkg returned an error code (1)
vel@vel-Precision-3650-Tower:~$ sudo systemctl status jenkins.service
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
  Active: failed (Result: exit-code) since Mon 2023-01-23 09:52:59 CET; 15s ago
    Process: 12253 ExecStart=/usr/bin/jenkins (code=exited, status=1/FAILURE)
   Main PID: 12253 (code=exited, status=1/FAILURE)
     CPU: 4ms

janv. 23 09:52:59 vel-Precision-3650-Tower systemd[1]: jenkins.service: Scheduled restart job, restart cou
janv. 23 09:52:59 vel-Precision-3650-Tower systemd[1]: Stopped Jenkins Continuous Integration Server.
janv. 23 09:52:59 vel-Precision-3650-Tower systemd[1]: jenkins.service: Start request repeated too quickly
janv. 23 09:52:59 vel-Precision-3650-Tower systemd[1]: jenkins.service: Failed with result 'exit-code'.
janv. 23 09:52:59 vel-Precision-3650-Tower systemd[1]: Failed to start Jenkins Continuous Integration Serv
vel@vel-Precision-3650-Tower:~$ sudo systemctl restart jenkins.service
vel@vel-Precision-3650-Tower:~$ sudo systemctl status jenkins.service
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2023-01-23 09:53:32 CET; 2s ago
    Main PID: 13122 (java)
       Tasks: 62 (limit: 38057)
      Memory: 3.0G
         CPU: 40.414s

```

```

CGroup: /system.slice/jenkins.service
          └─13122 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/va

janv. 23 09:53:24 vel-Precision-3650-Tower jenkins[13122]: 5ca2e100cf8f43b68a9f49fbf8ee4845
janv. 23 09:53:24 vel-Precision-3650-Tower jenkins[13122]: This may also be found at: /var/lib/jenkins/sec
janv. 23 09:53:24 vel-Precision-3650-Tower jenkins[13122]: ****
janv. 23 09:53:24 vel-Precision-3650-Tower jenkins[13122]: ****
janv. 23 09:53:24 vel-Precision-3650-Tower jenkins[13122]: ****
janv. 23 09:53:24 vel-Precision-3650-Tower jenkins[13122]: 2023-01-23 08:53:32.478+0000 [id=55]      INF
janv. 23 09:53:24 vel-Precision-3650-Tower jenkins[13122]: 2023-01-23 08:53:32.505+0000 [id=32]      INF
janv. 23 09:53:32 vel-Precision-3650-Tower systemd[1]: Started Jenkins Continuous Integration Server.
janv. 23 09:53:33 vel-Precision-3650-Tower jenkins[13122]: 2023-01-23 08:53:33.702+0000 [id=86]      INF
janv. 23 09:53:33 vel-Precision-3650-Tower jenkins[13122]: 2023-01-23 08:53:33.703+0000 [id=86]      INF
lines 1-20/20 (END)

```

Une fois le service démarré, on peut accéder à la page localhost de Jenkins à l'adresse :
<http://localhost:8080>

Si besoin vérifier avec la commande **netstat -antp**



```

cat: /var/lib/jenkins/secrets/initialAdminPassword: Permissions non accordées
vel@vel-Precision-3650-Tower:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPa
[sudo] Mot de passe de vel :

```

On arrive ensuite sur la page de configuration des plugins

The screenshot shows the Jenkins "Personnaliser Jenkins" (Customize Jenkins) page. The title is "Personnaliser Jenkins". A sub-section says: "Les plugins étendent Jenkins avec des fonctionnalités additionnelles pour satisfaire différents besoins." Below this are two main sections: "Installer les plugins suggérés" (Install suggested plugins) and "Sélectionner les plugins à installer" (Select plugins to install). The "Installer les plugins suggérés" section contains the text: "Installer les plugins que la communauté Jenkins trouve les plus utiles." The "Sélectionner les plugins à installer" section contains the text: "Sélectionner et installer les plugins les plus utiles à vos besoins."

Une fois que tout est installé en sélectionnant : **Installer les plugins suggérés** : créer un nouveau projet

The screenshot shows the Jenkins General configuration page for a project named "First project... FREESTYLE". The "Enabled" toggle is turned on. In the "General" section, under "Description", the text "First project... FREESTYLE" is displayed. Under "GitHub project", the "Project url" is set to "git@github.com:VelRunner/test_jenkins_freestyle.git". There are several advanced options like "Ce build a des paramètres", "GitHub project", "Project url", "Avancé", "Supprimer les anciens builds", "Throttle builds", and "Exécuter des builds simultanément si nécessaire". Below the general section, there is a "Gestion de code source" section with a "Repository URL" field containing "git@github.com:VelRunner/test_jenkins_freestyle.git". A "Sauver" (Save) button is visible.

Il faut rentrer la **clé privée** qu'on utilisera du projet

The screenshot shows the Jenkins Git configuration page. The "Git" option is selected. In the "Repositories" section, there is a "Repository URL" field with the value "git@github.com:VelRunner/test_jenkins_freestyle.git". A red error message below the field states: "Failed to connect to repository : Command "git ls-remote -h --git@github.com:VelRunner/test_jenkins_freestyle.git HEAD" returned status code 128: stdout: stderr: No ED25519 host key is known for github.com and you have requested strict checking. Host key verification failed. fatal: Impossible de lire le dépôt distant." A note below says "Veuillez vérifier que vous avez les droits d'accès et que le dépôt existe." In the "Credentials" section, there is a text input field with "vel" and a "Ajouter" (Add) button. A note below says "Credential used to check out sources. (from Git)".

Pour ce TP, il faudra supprimer la host key vérification dans les paramètres de sécurités

Git Host Key Verification Configuration

Host Key Verification Strategy ?

No verification

| ⚠ This option is generally insecure. Host key will not be verified during SSH connection

Enregistrer

Appliquer

Le message d'erreur aura disparu

Git ?

Repositories ?

Repository URL ?

git@github.com:VelRunner/test_jenkins_freestyle.git

Credentials ?

vel

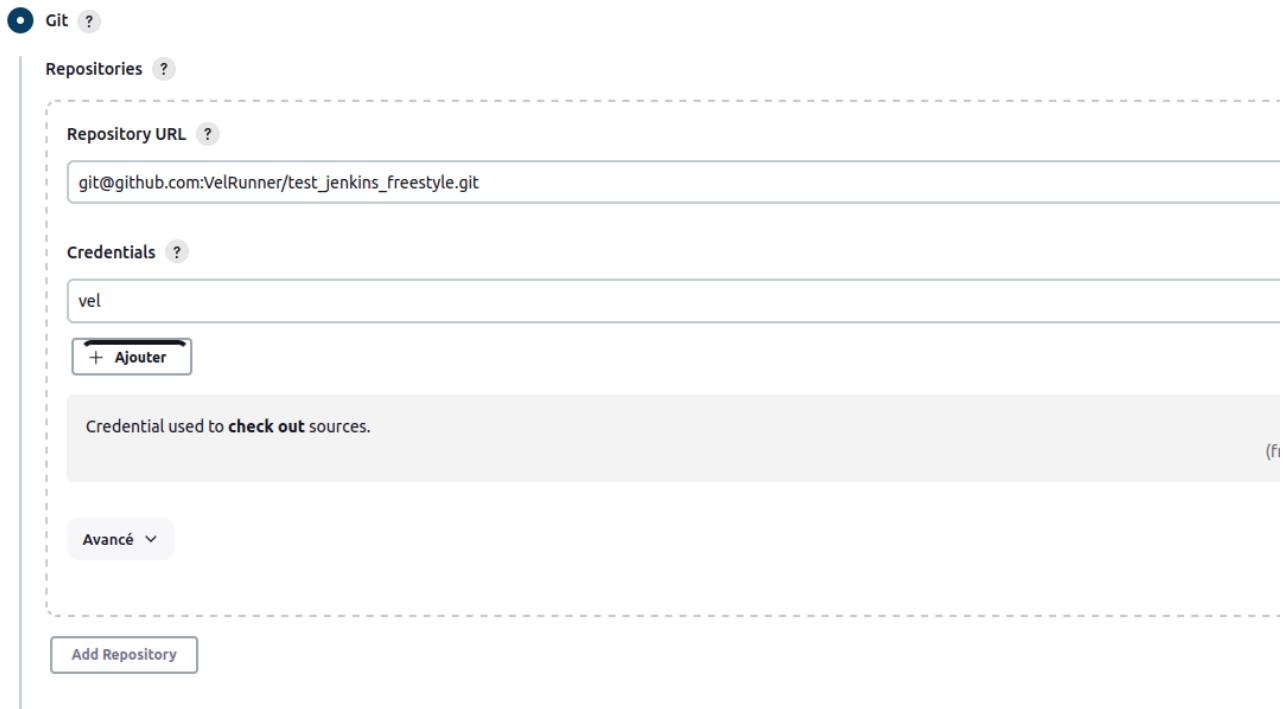
+ Ajouter

Credential used to **check out** sources.

(f)

Avancé ▾

Add Repository



Reste des options choisie pour ce TP

- Déclencher les builds à distance (Par exemple, à partir de scripts) ?
- Construire après le build sur d'autres projets ?
- Construire périodiquement ?
- GitHub hook trigger for GITScm polling ?
- Scrutation de l'outil de gestion de version ?

Environnements de Build

- Delete workspace before build starts
 - Avancé ▾
- Use secret text(s) or file(s) ?
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant ?

Build Steps

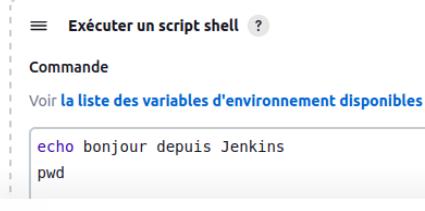
Exécuter un script shell ?

Commande

Voir [la liste des variables d'environnement disponibles](#)

```
echo bonjour depuis Jenkins
pwd
```

Sauver Apply



Une fois la configuration sauvegarder, il est possible de lancer un build sur cette page :

Projet test_freestyle

- État
- Modifications
- Répertoire de travail
- Lancer un build
- Configurer
- Supprimer Projet
- Favoris
- GitHub**
- Open Blue Ocean
- Renommer

Liens permanents

- Dernier build (#3), il y a 4 mn 24 s
- Dernier build en échec (#3), il y a 4 mn 24 s
- Dernier build non réussi (#3), il y a 4 mn 24 s
- Last completed build (#3), il y a 4 mn 24 s

Historique des builds [tendance](#)

Filter builds... /

#	Date
#4	23 janv. 2023 11:23
#3	23 janv. 2023 11:18
#2	23 janv. 2023 10:55
#1	23 janv. 2023 10:55

[Atom feed des builds](#) [Atom feed des échecs](#)

Résultat d'un build réussi :

✓ Sortie de la console

```
Démarré par l'utilisateur vel_seifer
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/test_freestyle
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] Done
The recommended git tool is: NONE
using credential vel_jenkins_ssh
Cloning the remote Git repository
Cloning repository git@github.com:VelRunner/test_jenkins_freestyle.git
> git init /var/lib/jenkins/workspace/test_freestyle # timeout=10
Fetching upstream changes from git@github.com:VelRunner/test_jenkins_freestyle.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT SSH to set credentials
> git fetch --tags --force --progress -- git@github.com:VelRunner/test_jenkins_freestyle.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url git@github.com:VelRunner/test_jenkins_freestyle.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision cead05bee3ce2b22d2d71a69f7771f9451ba94e (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f cead05bee3ce2b22d2d71a69f7771f9451ba94e # timeout=10
Commit message: "Create readme.txt"
> git rev-list --no-walk cead05bee3ce2b22d2d71a69f7771f9451ba94e # timeout=10
[test_freestyle] $ /bin/sh -xe /tmp/jenkins2523620363602591900.sh
+ echo bonjour depuis Jenkins
bonjour depuis Jenkins
+ pwd
/var/lib/jenkins/workspace/test_freestyle
+ touch file
Finished: SUCCESS
```

Note : il est impératif d'avoir au moins 1 fichier dans le dépôt **GIT** pour que le build fonctionne, sinon il passera en erreur

Le projet push sur Jenkins se trouve dans le fichier **/var/lib/jenkins/workspace/**

```
vel@vel-Precision-3650-Tower:~$ ls /var/lib/jenkins/workspace/test_freestyle
file  readme.txt
```

Test avec fichier python :

```
vim main.py
#!/usr/bin/python3
Aucun micro
Assurez-vous
ordinateur
```

```

print("Content-type: text/html")
print("")

def add(b):
    return b+b

a=2
print("<p>",add(a), '</p>')

for i in range(add(a)):
    print("le chiffre: ",a)

```

Redémarrer service : fcgiwrap et fcgiwrap.socket

```
[darksasuke@chocolat test_freestyle ]$ sudo systemctl restart fcgiwrap
[sudo] Mot de passe de darksasuke :
```

Pour remplacer automatiquement notre fichier avec Jenkins -> Nous allons exécuter un script shell :

Vérification du fichier de configuration

```

root@asus:/etc/nginx/conf.d# more test.conf
location ~ \.cgi$ {
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/fcgiwrap.socket;
    fastcgi_param DOCUMENT_ROOT /var/www/cgi-bin/;
    fastcgi_param SCRIPT_NAME script_python.py;
}

```

Pour remplacer automatiquement notre fichier avec Jenkins -> Nous allons exécuter un script shell :

Commande

Voir [la liste des variables d'environnement disponibles](#)

```

echo "test"
ls -la
chmod 755 main.py

```

```

echo "build"
echo "deploy"

cp main.py /usr/share/nginx/cgi-bin/

```

Il nous manque les droits pour pouvoir copier correctement : **chown** requis – l'utilisateur Jenkins doit devenir propriétaire du dossier **cgi-bin**

```

[darksasuke@chocolat test_freestyle ]$ ls -la /usr/share/nginx/cgi-bin/
total 16
drwxr-xr-x 2 http http 4096 16 janv. 09:42 .
drwxr-xr-x 8 http http 4096 13 janv. 16:46 ..
-rwxr-xr-x 1 root root 394 16 janv. 09:42 contacts.cgi
-rwxr-xr-x 1 http http 314 16 janv. 09:40 index.cgi
[darksasuke@chocolat test_freestyle ]$ chown jenkins /usr/share/nginx/cgi-bin/
chown: modification du propriétaire de '/usr/share/nginx/cgi-bin/': Opération non permise
[darksasuke@chocolat test_freestyle ]$ chown jenkins /usr/share/nginx/cgi-bin/

```

Une fois les modifications effectuées -> on relance un build, ce dernier devrait se dérouler avec succès

Prochain exercice : Apache Tomcat

https://wiki.archlinux.org/title/Apache_HTTP_Server

<https://doc.ubuntu-fr.org/apache2>

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04>

Maven :

Objectif de l'exercice : mettre en place Apache / TLS / redirection HTTP vers HTTPS

Avec Tomcat -> Réussir à allez sur la page managé

Redirection Apache vers Tomcat

Puis déployer une application ->

Dans un 1er temps -> stop nginx

```

vel@vel-Precision-3650-Tower:~$ systemctl stop nginx.service
vel@vel-Precision-3650-Tower:~$ sudo systemctl stop nginx
vel@vel-Precision-3650-Tower:~$ netstat -antp
(Tous les processus ne peuvent être identifiés, les infos sur les processus
non possédés ne seront pas affichées, vous devez être root pour les voir toutes.)
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale      Adresse distante      Etat      PID/Program name
tcp      0      0 127.0.0.53:53          0.0.0.0:*          LISTEN     -
tcp      0      0 127.0.0.1:631         0.0.0.0:*          LISTEN     -
tcp      0      0 0.0.0.0:21100        0.0.0.0:*          LISTEN     -
tcp      0      0 0.0.0.0:7070         0.0.0.0:*          LISTEN     -
tcp      0      0 0.0.0.0:22           0.0.0.0:*          LISTEN     -
tcp      0      0 10.125.25.59:35630   20.50.80.210:443  ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:43440   52.112.120.221:443  ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:53632   52.112.120.172:443  ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:51356   34.120.115.102:443 TIME_WAIT   -
tcp      0      0 10.125.25.59:53618   52.112.120.172:443  ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:56936   52.111.231.4:443   ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:47530   142.250.179.99:443 TIME_WAIT   -
tcp      0      0 10.125.25.59:49678   13.107.6.171:443   ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:37506   216.58.214.164:443 TIME_WAIT   -
tcp      0      0 10.125.25.59:45638   216.58.214.163:443 TIME_WAIT   -
tcp      0      0 10.125.25.59:60672   34.117.237.239:443 TIME_WAIT   -
tcp      0      0 10.125.25.59:48048   34.120.208.123:443 TIME_WAIT   -
tcp      0      0 10.125.25.59:59442   52.114.88.86:443   ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:37490   216.58.214.164:443 TIME_WAIT   -
tcp      0      0 10.125.25.59:38342   52.109.89.77:443   ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:40264   52.41.11.218:443   ESTABLISHED 3658/firefox
tcp      0      0 10.125.25.59:38630   138.199.36.116:443 ESTABLISHED -
tcp      0      0 10.125.25.59:56414   199.232.194.2:443  ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:48266   52.112.120.219:443 ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:49694   13.107.6.171:443   ESTABLISHED 5383/teams-for-linu
tcp6     0      0 ::1:631            ::*:*                LISTEN     -
tcp6     0      0 ::::8080          ::*:*                LISTEN     -
tcp6     0      0 ::::22           ::*:*                LISTEN     -
vel@vel-Precision-3650-Tower:~$ 

```

Je vois bien que mon port 80 est libre suite à l'arrêt du service

Installer Apache :

```

vel@vel-Precision-3650-Tower:~$ 
vel@vel-Precision-3650-Tower:~$ sudo apt install apache2

```

```

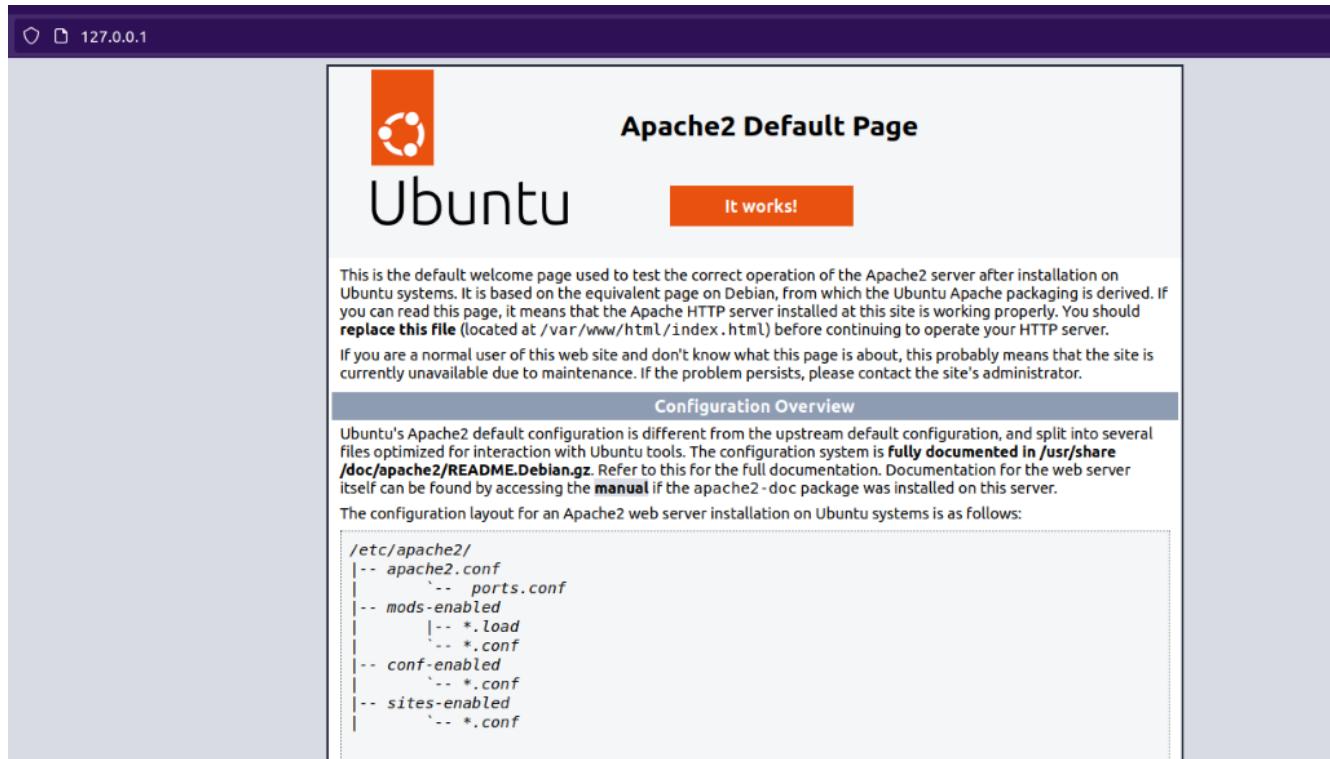
vel@vel-Precision-3650-Tower:~$ netstat -antp
(Tous les processus ne peuvent être identifiés, les infos sur les processus
non possédés ne seront pas affichées, vous devez être root pour les voir toutes.)
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale      Adresse distante      Etat      PID/Program name
tcp      0      0 127.0.0.53:53          0.0.0.0:*          LISTEN     -
tcp      0      0 127.0.0.1:631         0.0.0.0:*          LISTEN     -
tcp      0      0 0.0.0.0:21100        0.0.0.0:*          LISTEN     -
tcp      0      0 0.0.0.0:7070         0.0.0.0:*          LISTEN     -
tcp      0      0 0.0.0.0:22           0.0.0.0:*          LISTEN     -
tcp      0      0 10.125.25.59:56328   52.111.231.17:443  ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:43440   52.112.120.221:443  ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:49422   95.100.252.32:80   TIME_WAIT   -
tcp      0      0 10.125.25.59:50994   142.250.179.86:443 ESTABLISHED 3658/firefox
tcp      0      0 10.125.25.59:46584   142.250.201.163:443 ESTABLISHED 3658/firefox
tcp      0      0 10.125.25.59:56694   34.120.208.123:443 ESTABLISHED 3658/firefox
tcp      0      0 10.125.25.59:47596   13.69.116.104:443  ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:54326   151.101.64.233:443 ESTABLISHED 3658/firefox
tcp      0      0 10.125.25.59:50758   52.109.89.77:443   ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:49678   13.107.6.171:443   ESTABLISHED 5383/teams-for-linu
tcp      0      0 10.125.25.59:59642   34.200.15.139:443  ESTABLISHED 3658/firefox
tcp      0      0 10.125.25.59:50066   34.120.115.102:443 ESTABLISHED 3658/firefox
tcp      0      0 10.125.25.59:38914   216.58.215.42:443 ESTABLISHED 3658/firefox

```

tcp	0	0	10.125.25.59:37520	216.58.215.36:443	ESTABLISHED	3658/firefox
tcp	0	0	10.125.25.59:37942	104.16.182.15:443	ESTABLISHED	3658/firefox
tcp	0	0	10.125.25.59:33058	216.58.214.78:443	ESTABLISHED	3658/firefox
tcp	0	0	10.125.25.59:34256	104.17.72.206:443	ESTABLISHED	3658/firefox
tcp	0	0	10.125.25.59:34220	216.58.215.36:443	ESTABLISHED	3658/firefox
tcp	0	0	10.125.25.59:54292	104.16.181.15:443	ESTABLISHED	3658/firefox
tcp	0	0	10.125.25.59:59442	52.114.88.86:443	ESTABLISHED	5383/teams-for-linu
tcp	0	0	10.125.25.59:58202	216.58.215.46:443	ESTABLISHED	3658/firefox
tcp	0	0	10.125.25.59:40264	52.41.11.218:443	ESTABLISHED	3658/firefox
tcp	0	0	10.125.25.59:38630	138.199.36.116:443	ESTABLISHED	-
tcp	0	0	10.125.25.59:57880	93.184.220.29:80	TIME_WAIT	-
tcp	0	0	10.125.25.59:48266	52.112.120.219:443	ESTABLISHED	5383/teams-for-linu
tcp	0	0	10.125.25.59:49694	13.107.6.171:443	ESTABLISHED	5383/teams-for-linu
tcp6	0	0	::1:631	::*	LISTEN	-
tcp6	0	0	::8080	::*	LISTEN	-
tcp6	0	0	::22	::*	LISTEN	-
tcp6	0	0	::80	::*	LISTEN	-

On voit mon 80 utilisé par un service -> Installation OK

Tentative d'accès à la page web avec l'adresse 127.0.0.1 -> OK



Le fichier de configuration d'apache se trouve dans `/etc/apache2`

```
vel@vel-Precision-3650-Tower:~$ ls /etc/apache2/
apache2.conf      conf-enabled    magic          mods-enabled  sites-available
conf-available   envvars        mods-available  ports.conf    sites-enabled

```

HTTPS :

6.1 Activation du module SSL

Pour que le protocole **TLS** (successeur du **SSL**) puisse fonctionner avec Apache2, il faut activer le module `ssl` avec la commande :

```
sudo a2enmod ssl
```

puis recharger la configuration d'Apache 2 :

```
sudo systemctl reload apache2
```

```
vel@vel-Precision-3650-Tower:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb
```

```

Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificate
s.
To activate the new configuration, you need to run:
  systemctl restart apache2
vel@vel-Precision-3650-Tower:~$ systemctl restart apache2
vel@vel-Precision-3650-Tower:~$ 
```

A2ensite default-ssl -> activation ftps

Fichier **default-ssl.conf** généré : ce fichier va gérer toute les connexion FTPS d'apache :

```
vel@vel-Precision-3650-Tower:~$ cat /etc/apache2/sites-enabled/default-ssl.conf 
```

Si besoin, changer la clé et le certificat

```

# SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/nginx/ssl/www.vel.fr/nginx-selfsigned-vel.crt
SSLCertificateKeyFile   /etc/nginx/ssl/www.vel.fr/nginx-selfsigned-vel.key 
```

Exemple de fichier de configuration d'un site web : ici default.conf

```

<VirtualHost *:8070>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName vel.fr
    ServerAlias vel.fr

    Redirect permanent / https://www.vel.fr

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

<VirtualHost *:4437>
    ServerName vel.fr
    ServerAlias vel.fr
    ServerAdmin webmaster@example.com
    DocumentRoot /var/www/html
        <Directory /var/www/html>
```

TOMCAT :

<https://www.digitalocean.com/community/tutorials/install-tomcat-on-linux>

Proxy :

Un reverse proxy (proxy inversé), dans l'utilisation que je vais vous décrire ici, est une sorte passerelle entre un nom de domaines et un service (port). En gros, il nous sert à accéder à l'interface de certains services.

On peut nommer un service sans préciser de port, par exemple, nous devons accéder à l'interface de cups mais pas en via l'adresse <http://serveur.domaine.fr:631> mais via <http://print.domaine.fr> ou encore à l'interface de transmission via <http://torrent.domaine.fr> au lieu de <http://serveur.domaine.fr:9091>. Vous l'aurez compris, on utilise un nom de domaine pour accéder à l'interface d'un service sans avoir à préciser le port.

Cet article détaille comment configurer Apache et ne traite pas de l'attribution du nom de domaine à votre host.

Activation des modules utilisés

Pour utiliser Apache en mode proxy inverse, il nous faut activer les 2 modules `proxy` et `proxy_http` via la commande `a2enmod`.

```
# a2enmod proxy proxy_http
```

Le module `proxy` permet l'implémentation d'un serveur mandataire / passerelle.

Le module `proxy_http` apporte le support des requêtes HTTP et HTTPS au module `proxy`.

Pour activer ces services, il est nécessaire de redémarrer apache.

```
# service apache2 restart
```

```
vel@vel-Precision-3650-Tower:~$ a2enmod proxy proxy_http
Could not create /etc/apache2/mods-enabled/proxy.conf: Permission denied
vel@vel-Precision-3650-Tower:~$ sudo a2enmod proxy proxy_http
[sudo] Mot de passe de vel :
Enabling module proxy.
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

Ne pas oublier de restart le service

Mise en place du virtualhost

Créons maintenant notre virtualhost, le modèle ci-dessous est fait pour cups mais peut être personnalisé pour toutes les interfaces Web de vos services.

```
1 # vi /etc/apache2/sites-available/cups.domaine.fr.conf
2 <VirtualHost *:80>
3   ServerName cups.domaine.fr
4   ServerAdmin postmaster@domaine.fr
5
6   ProxyPass / http://127.0.0.1:631/
7   ProxyPassReverse / http://127.0.0.1:631/
8   ProxyRequests Off
9 </VirtualHost>
```

- `ServerName` permet de spécifier le nom de domaine voulu
- `ProxyPass` et `ProxyPassReverse` donnent la correspondance entre le path voulu et l'adresse du serveur destinataire et ajustent l'URL dans les en-têtes HTTP.
- `ProxyRequests` permet d'activer et de désactiver la fonctionnalité de serveur mandataire, doit être à `Off` pour des raisons de sécurité expliquée [ici](#).

Maintenant que notre fichier de configuration virtualhost est défini, nous devons l'activer puis recharger Apache.

Installer java8
<https://www.geofis.org/en/install/install-on-linux/install-openjdk-8-on-ubuntu-trusty/>

Correction :

1ère partie : installer apache + tomcat

```
[darksasuke@chocolat ~ ]$ sudo pacman -S tomcat9 apache2
[sudo] Mot de passe de darksasuke : [REDACTED]
```

Création des utilisateurs : modification du fichier `tomcat-users.xml`

```
[darksasuke@chocolat ~ ]$ sudo /etc/tomcat9/
Catalina/          jaspic-providers.xml  tomcat-users.xml
catalina.policy    jaspic-providers.xsd   tomcat-users.xsd
catalina.properties logging.properties    web.xml
context.xml        server.xml
[darksasuke@chocolat ~ ]$ sudo /etc/tomcat9/
```

Déclarer utilisateur + password + roles

```
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!... ...> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<role rolename="tomcat"/>
<user username="tomcat" password="manager" roles="tomcat,manager-gui,admin,manager-script"
/>
</tomcat-users>
-
-
        You will also need to set the passwords to something appropriate.
-->
<role rolename="tomcat,manager-gui,admin,manager-script"/>
<user username="manager" password="manager" roles="tomcat,manager-gui,admin,manager-s
t"/>
</tomcat-users>
  INSERTION
```

49 58

Connexion sur le site tomcat :



Gestionnaire d'applications WEB Tomcat

Message : OK

Gestionnaire					
Lister les applications		Aide HTML Gestionnaire		Aide Gestionnaire	
Etat du serveur					
Applications					
Chemin	Version	Nom d'affichage	Fonctionnelle	Sessions	Commandes
/	Aucun spécifié	Welcome to Tomcat	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
idoc	Aucun	Tomcat Documentation	true	0	Démarrer Arrêter Recharger Retirer

URL	Spécifié	Tomcat Documentation	True	False	Expirer les sessions inactives depuis ≥ 30 minutes
/examples	Aucun spécifié	Servlet and JSP Examples	true	false	Démarrer Arrêter Recharger Retirer
					Expirer les sessions inactives depuis ≥ 30 minutes
/host-manager	Aucun spécifié	Tomcat Host Manager Application	true	false	Démarrer Arrêter Recharger Retirer
					Expirer les sessions inactives depuis ≥ 30 minutes

```
[darksasuke@chocolat ~]$ ls /var/lib/tomcat9/webapps/
docs examples host-manager manager ROOT
```

```
[darksasuke@chocolat ~]$ ls /var/lib/tomcat9/webapps/
docs examples host-manager manager ROOT
[darksasuke@chocolat java_war_sparkexample ]$ ls
pom.xml README.md src target
```

Dans le fichier ou se trouve le **pom.xml** -> passer la commande **mvn clean install**

Vérifier notre version de java et passer en java 8 pour ce projet :

```
[darksasuke@chocolat java_war_sparkexample ]$ update-alternatives
bash: update-alternatives : commande introuvable
[darksasuke@chocolat java_war_sparkexample ]$ archlinux-java
archlinux-java <COMMAND>

COMMAND:
status          List installed Java environments and enabled one
get             Return the short name of the Java environment set as default
set <JAVA_ENV> Force <JAVA_ENV> as default
unset           Unset current default Java environment
fix             Fix an invalid/broken default Java environment configuration
[darksasuke@chocolat java_war_sparkexample ]$ archlinux-java status
Available Java environments:
  java-11-openjdk
  java-19-openjdk (default)
  java-8-openjdk
[darksasuke@chocolat java_war_sparkexample ]$ archlinux-java set java-8-openjdk
This script must be run as root
[darksasuke@chocolat java_war_sparkexample ]$ sudo !!
sudo archlinux-java set java-8-openjdk
[darksasuke@chocolat java_war_sparkexample ]$ ls|
```

Recommencer le **mvn clean install** si on été dans une mauvaise version, et importer le .war si le build est en success

```
vel@vel-Precision-3650-Tower:~/Atelier/git_project/sparkjava-war-example$ mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< sparkjava-hello-world:sparkjava-hello-world >-----
[INFO] Building sparkjava-hello-world 1.0
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ sparkjava-hello-world ---
[INFO] Deleting /home/vel/Atelier/git_project/sparkjava-war-example/target
[INFO]
[INFO] --- maven-enforcer-plugin:1.3.1:enforce (enforce-java) @ sparkjava-hello-world ---
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ sparkjava-hello-world ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /home/vel/Atelier/git_project/sparkjava-war-example/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ sparkjava-hello-world ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
```

```
[INFO] Compiling 1 source file to /home/vel/Atelier/git_project/sparkjava-war-example/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ sparkjava-hello-world ---
-
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform
dependent!
[INFO] skip non existing resourceDirectory /home/vel/Atelier/git_project/sparkjava-war-example/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ sparkjava-hello-world ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ sparkjava-hello-world ---
```

Applications					
Chemin	Version	Nom d'affichage	Fonctionnelle	Sessions	Commandes
/	Aucun spécifié	Welcome to Tomcat	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/docs	Aucun spécifié	Tomcat Documentation	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/examples	Aucun spécifié	Servlet and JSP Examples	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/host-manager	Aucun spécifié	Tomcat Host Manager Application	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/manager	Aucun spécifié	Tomcat Manager Application	true	1	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/sparkjava-hello-world-1.0	Aucun spécifié		false	0	Démarrer Arrêter Recharger Retirer

Utiliser notre site internet comme reverse proxy :

Sur Arch -> Vérifier les ports dans le fichier de configuration :

```
#Listen 12.34.56.78:80
Listen 80
Listen 443
```

Puis modifier le fichier de configuration du site :

```
sudo vim /etc/httpd/conf/extra/httpd-vhosts.conf
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:>http://httpd.apache.org/docs/2.4/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.
#
<VirtualHost *:80>
    ServerName loup.fr
        redirect permanent https:// %{HTTP_HOST} %{REQUEST_URI}
</VirtualHost>
```

Sur ubuntu :

Modification fichier default-ssl.conf

```
GNU nano 6.2                               /etc/apache2/sites-enabled/default-ssl.conf
>IfModule mod_ssl.c
    <VirtualHost *:443>
```

```

<VirtualHost *:443>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn
ProxyPass / http://localhost:8080/
ProxyPassReverse / http://localhost:8080/

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    # A self-signed (snakeoil) certificate can be created by installing
    # the ssl-cert package. See
    # /usr/share/doc/apache2/README.Debian.gz for more info.
    # If both key and certificate are stored in the same file, only the
    # SSLCertificateFile directive is needed.
    SSLCertificateFile      /etc/nginx/ssl/www.vel.fr/nginx-selfsigned-vel.crt
    SSLCertificateKeyFile   /etc/nginx/ssl/www.vel.fr/nginx-selfsigned-vel.key

    # Server Certificate Chain:

```

Modification

```

<VirtualHost *:8070>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName vel.fr
    ServerAlias vel.fr

    Redirect permanent / https://vel.fr:443

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn
    #ErrorLog ${APACHE_LOG_DIR}/error.log
    #CustomLog ${APACHE_LOG_DIR}/access.log combined
    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

    #Proxy

    #ProxyPreserveHost On
    #ProxyPass / http://localhost:8080/
    #ProxyPassReverse / http://localhost:8080/
    #ProxyRequests Off

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
```

Une fois la configuration effectué : redémarrer le service **apache**

Configuration sous Jenkins :

Saisissez un nom

spark-java

» Champ obligatoire

Construire un projet free-style
Ceci est la fonction principale de Jenkins qui sert à builder (construire) votre projet. Vous pouvez intégrer tous les outils de gestion de version avec tous les systèmes de build. Il est même possible d'utiliser Jenkins pour tout autre chose qu'un build logiciel.

Pipeline
Organise des activités de longue durée qui peuvent s'étendre sur plusieurs agents de construction. Adapté pour la création des pipelines (anciennement connues comme workflows) et/ou pour organiser des activités complexes qui ne s'adaptent pas facilement à des tâches de type libre.

Construire un projet multi-configuration
Adapté aux projets qui nécessitent un grand nombre de configurations différentes, comme des environnements de test multiples, des binaires spécifiques à une plateforme, etc.

Récupération d'un projet git :

```
remote: Total 29 (delta 0), reused 0 (delta 0), pack-reused 29
Réception d'objets: 100% (29/29), fait.
Résolution des deltas: 100% (10/10), fait.
[darksasuke@chocolat cloned ]$ git clone git@framagit.org:bkoj/spark_java.git
Clonage dans 'spark_java'...
avertissement : Vous semblez avoir cloné un dépôt vide.
[darksasuke@chocolat cloned ]$ cp sparkjava-war-example/. spark_java/
./          ../.git/      .gitignore
[darksasuke@chocolat cloned ]$ cp sparkjava-war-example/.git spark_java/
.git/       .gitignore
[darksasuke@chocolat cloned ]$ rm -r sparkjava-war-example/.git
rm : supprimer 'sparkjava-war-example/.git/objects/pack/pack-5b7251d6e32ddeda58782b148e6c
4bab2f42fd.idx' qui est protégé en écriture et est du type « fichier » ? ^C
[darksasuke@chocolat cloned ]$ sudo !!
sudo rm -r sparkjava-war-example/.git
[sudo] Mot de passe de darksasuke :
[darksasuke@chocolat cloned ]$ cp sparkjava-war-example/* -r spark_java/
[darksasuke@chocolat cloned ]$ cd spark_java/
[darksasuke@chocolat spark_java ]$ ls
pom.xml  README.md  src
[darksasuke@chocolat spark_java ]$ git add .
[darksasuke@chocolat spark_java ]$ git commit -m "vol"
[main (commit racine) aca3f2a] vol
 4 files changed, 105 insertions(+)
  create mode 100644 README.md
  create mode 100644 pom.xml
```

```
create mode 100644 pom.xml
create mode 100644 src/main/java/HelloWorld.java
create mode 100644 src/main/webapp/WEB-INF/web.xml
[darksasuke@chocolat spark_java ]$ git push
Énumération des objets: 11, fait.
Décompte des objets: 100% (11/11), fait.
Compression par delta en utilisant jusqu'à 16 fils d'exécution
Compression des objets: 100% (7/7), fait.
Écriture des objets: 100% (11/11), 1.83 Kio | 624.00 Kio/s, fait.
Total 11 (delta 0), réutilisés 0 (delta 0), réutilisés du pack 0
```

Pour récupérer un projet git et le push sur le nôtre : suppression des dossier/fichiers .git d'un projet cloné

Environnements de Build

Delete workspace before build starts

Avancé ▾

Use secret text(s) or file(s) ?

Add timestamps to the Console Output

Inspect build log for published build scans

Terminate a build if it's stuck

With Ant ?

Build Steps

≡ Exécuter un script shell ?



Commande

Voir [la liste des variables d'environnement disponibles](#)

```
mvn clean install
```

- Informations de la construction
- Supprimer le build "#1"
- Git Build Data

Construction #1 (25 janv. 2023 à 11:09:30)

Conserver cette construction sans limite de temps

Démarrée il y a 31 s.
A duré 8 s

Ajouter une description

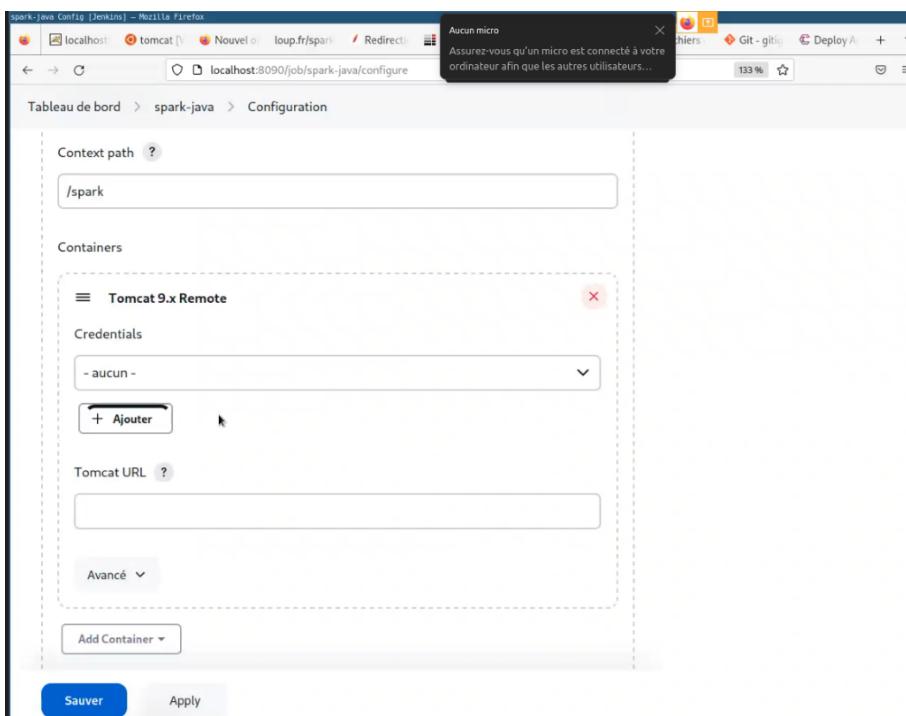
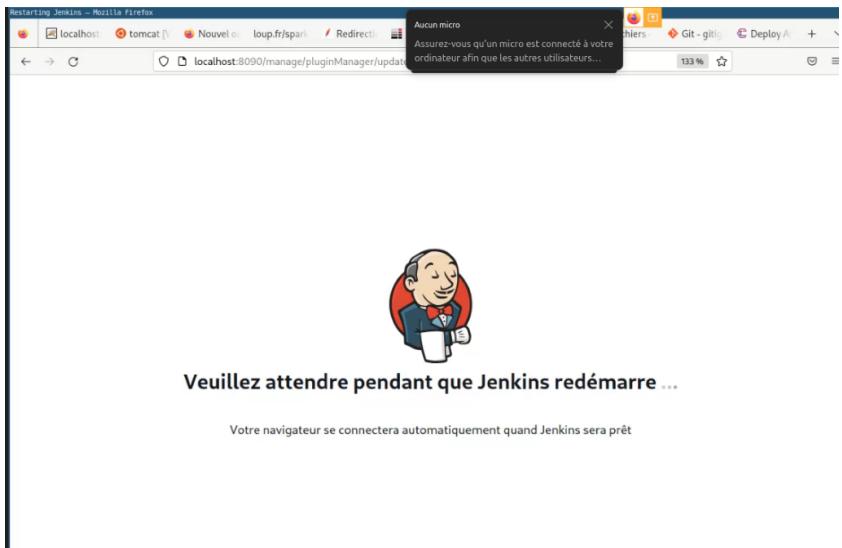
</> No changes.

Lancé par l'utilisateur **sasuke**

git Revision: e39b9112ddccdb86e629f02e9ac6eleaf95c4fc
Repository: git@framagit.org:ko/spark_java.git
refs/remotes/origin/main

REST API Jenkins 2.38

Télécharger jenkins Conteneur



Rajouter manager-script dans configuration user tomcat

```
<role rolename="tomcat,manager-gui,admin,manager-script"/>
<user username="manager" password="manager" roles="tomcat,manager-gui,admin,manager-script"/>
</tomcat-users>
```

Tableau de bord > spark-java > Configuration

Context path ?

/spark

Containers

Tomcat 9.x Remote



Credentials

manager/******/



+ Ajouter

Tomcat URL ?

http://localhost:8080/



Avancé ▾

Add Container ▾

Sauver

Apply

Si la configuration est ok -> Build fonctionne

Tester déploiement automatiser : github webhook



jenkins github webhook



<https://plugins.jenkins.io/github> · Traduire cette page

GitHub - Jenkins Plugins

4 janv. 2023 — The **Github** plugin decorates **Jenkins** "Changes" pages to create links to your **Github** commit and issue pages. It adds a sidebar link that links ...
Vous avez consulté cette page le 25/01/23.

<https://santoshk.dev/posts/how-to...> · Traduire cette page

How to Setup a GitHub to Jenkins Pipeline with WebHooks

20 févr. 2022 — This is one extension that we'll use with our **Jenkins** installation. To install this extension, head over your **Jenkins** instance and go to ...

<https://www.cprime.com/blog> · Traduire cette page

How to Integrate Jenkins with GitHub | Cprime Blogs

Git is an essential plugin that everyone working with Jenkins will need. You can get to the point of ... **Jenkins Github Webhook** Button Successful ...

Autres questions posées :

How to setup webhook in GitHub for Jenkins?



Can we connect Git with Jenkins?



How do I trigger a Jenkins build from GitHub?

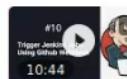


How do I trigger Jenkins pipeline from GitHub?



Commentaires

Vidéos :



Automatically Trigger Jenkins Jobs Using Github WebHook ...

YouTube · Mukesh otwani

5 mai 2021

On clone notre projet :

```

~/clones/java_war_sparkexample
[darksasuke@chocolat cloned ]$ git clone git@github.com:bkoj-arch/java_war_sparkexample.git
Clonage dans 'java_war_sparkexample'...
remote: Enumerating objects: 59, done.
remote: Counting objects: 100% (59/59), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 59 (delta 4), reused 59 (delta 4), pack-reused 0
Réception d'objets: 100% (59/59), 4.63 Mio | 1.27 Mio/s, fait.
Résolution des deltas: 100% (4/4), fait.
[darksasuke@chocolat cloned ]$ cd java_war_sparkexample/
[darksasuke@chocolat java_war_sparkexample ]$
```

Paramétrage webhook :

The screenshot shows the Jenkins configuration page for a job named "spark-java". The left sidebar lists several configuration sections: General, Source Code Management (Gestion de code source), Trigger (Ce qui déclenche le build), Environments (Environnements de Build), Build Steps, and Post-build Actions (Actions à la suite du build). The "Trigger" section is currently selected and expanded. It contains five checkboxes: "Déclencher les builds à distance (Par exemple, à partir de scripts)" (unchecked), "Construire après le build sur d'autres projets" (unchecked), "Construire périodiquement" (unchecked), "GitHub hook trigger for GITScm polling" (checked), and "Scrutation de l'outil de gestion de version" (unchecked). Below this, the "Environments de Build" section is also expanded, showing several optional checkboxes: "Delete workspace before build starts" (checked), "Use secret text(s) or file(s)" (unchecked), "Add timestamps to the Console Output" (unchecked), "Inspect build log for published build scans" (unchecked), "Terminate a build if it's stuck" (unchecked), and "With Ant" (unchecked).

The screenshot shows the deployment configuration for the "spark" container. The "Deploy war/ear to a container" section is open, showing the "Containers" list. A single entry, "Tomcat 9.x Remote", is selected and expanded. It contains a "Credentials" field with the value "manager/***" and a "Tomcat URL" field with the value "http://localhost:8080/". There is also an "Avancé" (Advanced) dropdown menu and a "Add Container" button. At the bottom of the page, there is a "Deploy on failure" checkbox.



On doit rajouter un webhook à notre github pour assurer la communication avec Jenkins s'il y a un push -> Il faut que ce soit un lien qui redirige vers Jenkins

The screenshot shows the GitHub repository settings for 'bkoj-arch/java_war_sparkexample'. The 'Webhooks' tab is selected. The left sidebar shows 'Code and automation' with 'Webhooks' highlighted. The main content area is titled 'Webhooks' and contains a brief description: 'Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#)'. There is a 'Add webhook' button at the top right of the main content area.

Nbrok va nous permettre d'encapsuler notre localhost:8090 dans une URL qui redirigera vers notre Jenkins.

```
[darksasuke@chocolat java_war_sparkexample ]$ ngrok http 8090
```

The screenshot shows the ngrok dashboard with the URL <https://7601-185-31-149-99.eu.ngrok.io> forwarded to <http://localhost:8090>. The dashboard also displays session status information including session status (online), session expiration (1 hour, 59 minutes), terms of service (<https://ngrok.com/tos>), version (3.1.1), region (Europe (eu)), latency (14ms), and web interface (http://127.0.0.1:4041).

The screenshot shows the Jenkins 'Add webhook' configuration page. It includes a note about sending POST requests to the URL with event details and a link to developer documentation. The 'Payload URL' field is populated with <https://7601-185-31-149-99.eu.ngrok.io/github-webhook/>. The 'Content type' dropdown is set to 'application/x-www-form-urlencoded'. A 'Secret' field is present at the bottom.

Obligatoire de rajouter `/github-webhook/` à la fin du lien

Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

● <https://7601-185-31-149-99.eu.ngrok.io/github-webhook/> (push)

Edit Delete

Ce lien permet de vérifier l'état de la connexion entre le webhook et Jenkins -> si réponse 200 -> OK

```
HTTP Requests
-----
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
GET /favicon.ico              200 OK
GET /                         403 Forbidden
```

Confirmation dans la fenêtre de nbrok

Pour que Jenkins comprenne que c'est nous, il faut générer un token dans github :
setting/developper settings

[Settings](#) / Developer settings

gg GitHub Apps
🔍 OAuth Apps
👤 Personal access tokens Beta
Fine-grained tokens
Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API](#) over Basic Authentication.

Note

What's this token for?

Expiration *

30 days The token will expire on Fri, Feb 24 2023

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry

[Settings](#) / Developer settings

gg GitHub Apps
🔍 OAuth Apps

Personal access tokens (classic)

Generate new token ▾

Revoke all

The screenshot shows the GitHub 'Personal access tokens' page under the 'Tokens (classic)' tab. A single token is listed with the value 'ghp_5Zvag4YvYf6wkK0mhz18K5KcXxV7b93iU3Nt'. A note at the top right says 'Make sure to copy your personal access token now. You won't be able to see it again!' and a 'Delete' button is visible.

Attention : il faut bien s'assurer d'avoir copié le token, il sera caché par github après avoir quitté la page

On rajoute notre token dans notre build jenkins

GitHub

GitHub Servers ?

GitHub Server ?

Pair of GitHub token and server URL. If no custom URL is specified, then the default `api.github.com` will be used. If your Jenkins uses multiple repositories that are spread across different user accounts, you can list them all here as separate configurations.

(from [GitHub plugin](#))

Name ?

github

API URL ?

`https://api.github.com`

Credentials ?

Secret text

- aucun -

Secret text

Secret text

test

Manage hooks

Avancé ▾

Add GitHub Server ▾

Enregistrer

Appliquer

Une fois la configuration sauvegardé, exécuter un push sur le dépôt git -> si le build s'effectue après un push -> OK

Plugin **Blue Ocean** :

<https://www.jenkins.io/doc/book/pipeline/jenkinsfile/>

Créer un **Pipeline** via blue ocean :

Jenkins demande directement un acces token -> redirection github, il génère les bon droit d'office

Une fois la configuration effectuée, on arrive sur cette page :

The screenshot shows the Jenkins pipeline configuration interface. At the top, it says "Jenkins" and "vel-sparkjava-war / vel-sparkjava-war". Below the header, there is a diagram of a pipeline stage. It starts with a grey circle labeled "Start", followed by a blue circle labeled "test", and ends with a grey circle labeled "+".

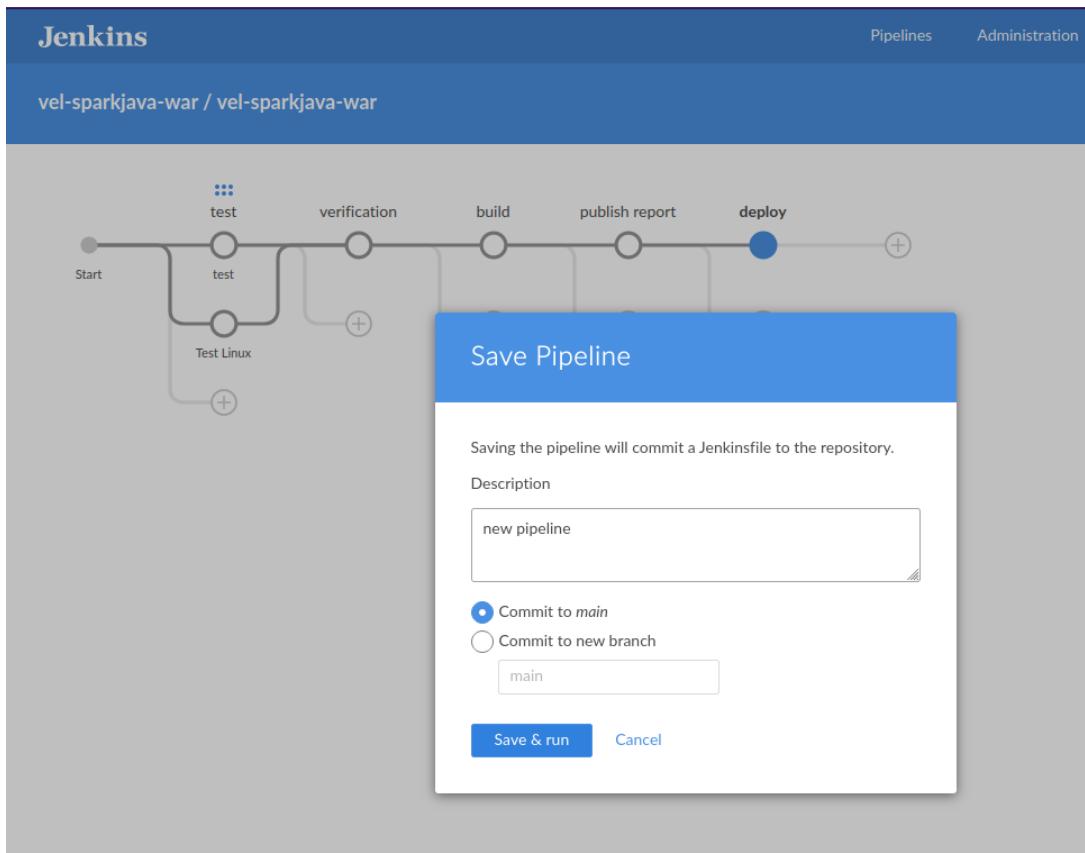
This screenshot shows the Jenkins pipeline configuration interface with a more complex pipeline. It includes a "Start" node, a "test" node (which is highlighted in blue), and a final "+". A context menu is open over the "test" node, with the option "test / Shell Script" visible. Other menu items like "Edit", "Delete", and "Test" are also present.

Ne pas enregistrer directement après avoir rajouté cette étape

Exemple de pipeline :

This screenshot shows a more complex Jenkins pipeline. It starts with a "Start" node, followed by a "test" stage (with a "test" step inside), a "verification" stage, and a "build" stage. The "test" stage has a sub-step "Test Linux" which loops back to the start of the "test" stage. The pipeline ends with a final "+".

Une fois terminé : **build and deploy**



Test création de pipeline avec interface de base de Jenkins :

Pipeline

Definition

Pipeline script

```
1> stage {
2>   steps {
3>     echo 'Hello Jenkins'
4>   }
5> }
```

try sample Pipeline... ▾

Use Groovy Sandbox ?

[Pipeline Syntax](#)

[Sauver](#)

[Apply](#)

Autre possibilité :

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?
git@github.com:VelRunner/vel-sparkjava-war.git

Credentials ?
vel
+ Ajouter

Avancé ▾

Add Repository

Status Pipeline pipeline_spark_java

</> Changes Démarrer la modification

▷ Lancer un build Configurer Supprimer Pipeline

GitHub Open Blue Ocean Renommer Pipeline Syntax GitHub Hook Log

Average stage times: (Average full run time: ~11s)

Declarative: Checkout SCM	test	test	Test Linux	verification	build	publish report	deploy
2s	302ms	0ms	461ms	447ms	1s	406ms	408ms

Stage View

Liens permanents

Historique des builds tendance Filter builds... #1 26 janv. 2023 11:10 Atom feed des builds Atom feed des échecs

Start test verification build publish report deploy End

build - 2s > mvn clean install — Shell Script

Normalement, Jenkins est censé fonctionner sur une VM, c'est pour ça qu'il nous dit qu'il doit utiliser un agent :

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#). [Set up agent](#) [Set up cloud](#) [Dismiss](#)

Des avertissements ont été publiés pour ces composants installés : [Script Security Plugin 1228.vd93135a_2fb_25](#): [Sandbox bypass vulnerability](#)

A fix for this issue is available. Go to the [plugin manager](#) to update the plugin.

Exécuter un script shell ?

Commande

Voir [la liste des variables d'environnement disponibles](#)

```
echo "test"  
ls -la  
chmod 755 main.py  
echo "build"  
echo "deploy"  
  
cp main.py /usr/share/nginx/cgi-bin/main.cgi
```

