

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

**Filière d'ingénieur :**

Ingénierie Informatique Big Data Cloud Computing  
IIBDCC

**Ingénierie des Infrastructures Big Data  
et Cloud**

**TP4 : Déploiement d'une application  
web sur le AWS**



amazon EC2



**Jenkins**

**Réalisé par :**

TSEH Kokou Benoit

**Encadré par :**

Pr. EL GUEMMAT

## INTRODUCTION

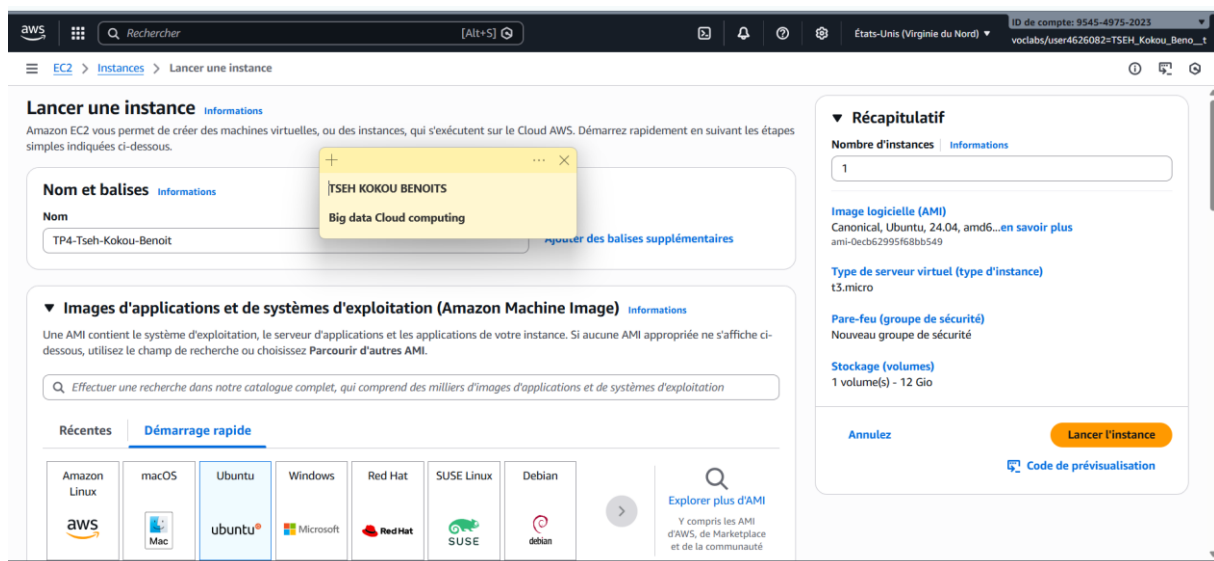
Dans le cadre de ce travail pratique, nous allons découvrir et mettre en œuvre les principes fondamentaux de l'intégration continue (CI) et du déploiement continu (CD) à l'aide de Jenkins et Docker. Ce TP a pour objectif principal de montrer comment automatiser les différentes étapes du cycle de vie d'une application web — depuis la création du code source jusqu'à son déploiement sur un environnement d'exécution. Dans un premier temps, nous procéderons à l'installation et à la configuration de Jenkins, à la création d'un projet web simple et à la mise en place d'un Job freestyle permettant de construire et de publier une image Docker sur Docker Hub. Dans la seconde partie, nous approfondirons la démarche DevOps en mettant en place une véritable chaîne CI/CD, avec des Jobs freestyle et pipeline automatisant les processus de build, test, publication et déploiement de l'application. Ce TP permettra ainsi d'acquérir une compréhension pratique du rôle de Jenkins dans l'automatisation des pipelines de développement, ainsi que de la collaboration entre les outils Git, Docker et Jenkins dans un environnement DevOps moderne.

## TECHNOLOGIES UTILISEES

### 1. Configuration de l'infrastructure sur AWS

#### 1. Création d'une Instance EC2 (Ubuntu) :

- J'ai créé une instance EC2 (système d'exploitation Ubuntu) avec un nom qui comprend nom et prénom : TP3\_Prenom\_Nom.
- J'ai configuré cette instance avec les ressources suffisantes (vCPU et RAM) pour supporter le déploiement de l'application.



EC2 > Instances > Lancer une instance

Type d'instance

t3.micro

Famille: t3 2 vCPU 1 Gio Mémoire Génération actuelle: true  
À la demande Ubuntu Pro base tarification: 0.0139 USD par heure  
À la demande SUSE base tarification: 0.0104 USD par heure  
À la demande Linux base tarification: 0.0104 USD par heure  
À la demande RHEL base tarification: 0.0392 USD par heure  
À la demande Windows base tarification: 0.0196 USD par heure

Éligible à l'offre gratuite

Toutes les générations

Comparer les types d'instance

Des frais supplémentaires s'appliquent pour les AMI avec un logiciel préinstallé

Paire de clés (connexion)

Vous pouvez utiliser une paire de clés pour vous connecter en toute sécurité à votre instance. Assurez-vous d'avoir accès à la paire de clés sélectionnée avant de lancer l'instance.

Nom de la paire de clés - obligatoire

tpdevops

Créer une paire de clés

Paramètres réseau

Réseau

vpc-0ae5c6789a0c92e85

Modifier

Récapitulatif

Nombre d'instances

1

Image logicielle (AMI)

Canonical, Ubuntu, 24.04, amd64...en savoir plus

Type de serveur virtuel (type d'instance)

t3.micro

Pare-feu (groupe de sécurité)

Nouveau groupe de sécurité

Stockage (volumes)

1 volume(s) - 12 Gio

Annulez

Lancer l'instance

Code de prévisualisation

aws

Rechercher

[Alt+S]

États-Unis (Virginie du Nord)

ID de compte: 9545-4975-2023

EC2 > Instances > Lancer une instance

Succès

Lancement de l'instance réussi (i-0abd2898d0e9acc68)

Journal de lancement

Étapes suivantes

Créer des alertes d'utilisation de la facturation

Pour gérer les coûts et éviter les factures surprises, configurez des notifications par e-mail pour la facturation et les seuils d'utilisation.

Créer des alertes de facturation

Connectez-vous à votre instance

Une fois que votre instance est en cours d'exécution, connectez-vous à celle-ci à partir de votre ordinateur local.

Connectez-vous à l'instance

En savoir plus

Connecter une base de données RDS

Configurez la connexion entre une instance EC2 et une base de données pour autoriser le flux de trafic entre elles.

Connecter une base de données RDS

Créer une nouvelle base de données RDS

En savoir plus

Afficher toutes les instances

## 2. Accès SSH et installation de Docker et Git

### 1. Connexion SSH à l'Instance :

Se connecter

Connectez-vous à une instance à l'aide du client basé sur un navigateur.

EC2 Instance Connect

Session Manager

Client SSH

EC2 Serial Console

ID d'instance

i-0abd2898d0e9acc68 (TP4-Tseh-Kokou-Benoit)

1. Ouvrez un client SSH.

2. Recherchez votre fichier de clé privée. La clé utilisée pour lancer cette instance est tdevops.pem

3. Exécuter, si nécessaire, cette commande pour vous assurer que votre clé n'est pas visible publiquement.

chmod 400 "tdevops.pem"

4. Connectez-vous à votre instance à l'aide de son DNS public :

ec2-3-228-13-51.compute-1.amazonaws.com

Exemple :

ssh -i "tdevops.pem" ubuntu@ec2-3-228-13-51.compute-1.amazonaws.com

Remarque :

Dans la plupart des cas, le nom d'utilisateur deviné est correct. Cependant, lisez les instructions d'utilisation de l'AMI pour vérifier si le propriétaire de l'AMI a modifié le nom d'utilisateur par défaut.

2025-2026

```
believer@DESKTOP-CFQ8VNV:/mnt/d/ENSET/SEMESTRES/INFRASTRUCTURE_BIG_DATA_CLOUD/TP4-Devops-Dans-Cloud$ cp tpdevops.pem ~/
cd ~
believer@DESKTOP-CFQ8VNV:~$ chmod 400 tpdevops.pem
believer@DESKTOP-CFQ8VNV:~$ ssh -i "tpdevops.pem" ubuntu@ec2-3-228-13-51.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Dec 25 16:30:45 UTC 2025

System load:  0.0           Temperature:   -273.1 C
Usage of /:   16.4% of 10.58GB Processes:      110
Memory usage: 22%          Users logged in: 0
Swap usage:   0%           IPv4 address for ens5: 172.31.6.48

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by

System information as of Thu Dec 25 16:30:45 UTC 2025

System load:  0.0           Temperature:   -273.1 C
Usage of /:   16.4% of 10.58GB Processes:      110
Memory usage: 22%          Users logged in: 0
Swap usage:   0%           IPv4 address for ens5: 172.31.6.48

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-6-48:~$
```

## 2. Installation de Docker :

```
$ sudo apt-get update
```

```
$ sudo curl -fsSL https://get.docker.com -o get-docker.sh
```

```
$ sudo sh get-docker.sh
```

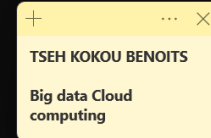
```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

```
ubuntu@ip-172-31-6-48:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
ubuntu@ip-172-31-6-48:~$ curl -fsSL https://get.docker.com -o get-docker.sh
ubuntu@ip-172-31-6-48:~$ sudo sh get-docker.sh
# Executing docker install script, commit: 8b33a64d28ec86a1121623f1d349801b48f2837b
Warning: the "docker" command appears to already exist on this system.
```

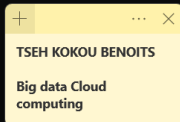
If you already have Docker installed, this script can cause trouble, which is why we're displaying this warning and provide the opportunity to cancel the installation.

If you installed the current Docker package using this script and are using it again to update Docker, you can ignore this message, but be aware that the script resets any custom changes in the deb and rpm repo configuration files to match the parameters passed to the script.

You may press Ctrl+C now to abort this script.  
+ sleep 20

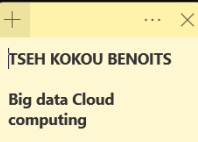


```
=====
ubuntu@ip-172-31-6-48:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-ce is already the newest version (5:29.1.3-1-ubuntu.24.04~noble).
docker-ce-cli is already the newest version (5:29.1.3-1-ubuntu.24.04~noble).
containerd.io is already the newest version (2.2.1-1-ubuntu.24.04~noble).
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
ubuntu@ip-172-31-6-48:~$
```



### 3. Installation de Git :

```
ubuntu@ip-172-31-6-48:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Fetched 126 kB in 1s (251 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
68 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-6-48:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
ubuntu@ip-172-31-6-48:~$ git --version
git version 2.43.0
ubuntu@ip-172-31-6-48:~$
```



### 3. Déploiement de l'Application

#### 1. Clonage du Projet GitHub :

o J'ai cloné le projet contenant la page index.html depuis GitHub :

Believer2001 / tp4-DevOps Public

Code Issues Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags

Go to file Code

About

No description, web.

Activity

0 stars

0 watching

0 forks

Report repository

Releases

Believer2001 q5	2cc03f7 · last month	20 Commits
Jenkinsfile	q5	last month
dockerfile	correction de dockerfile	last month
index.html	changement pour q5	last month

```
ubuntu@ip-172-31-6-48:~$ git clone https://github.com/Believer2001/tp4-DevOps.git
Cloning into 'tp4-DevOps'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 54 (delta 14), reused 49 (delta 9), pack-reused 0 (from 0)
Receiving objects: 100% (54/54), 7.13 KiB | 2.38 MiB/s, done.
Resolving deltas: 100% (14/14), done.
ubuntu@ip-172-31-6-48:~$
```

TSEH KOKOU BENOITS

Big data Cloud computing

## 2. Construction et Lancement de l'Image Docker :

o Construction de l'image Docker de l'application et démarrage du conteneur :

```
ubuntu@ip-172-31-6-48:~/tp4-DevOps$ ls
Jenkinsfile  dockerfile  index.html
ubuntu@ip-172-31-6-48:~/tp4-DevOps$ mv dockerfile Dockerfile
ubuntu@ip-172-31-6-48:~/tp4-DevOps$ docker build -t build .
ERROR: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping":
dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-6-48:~/tp4-DevOps$ sudo su
```

```
root@ip-172-31-6-48:/home/ubuntu/tp4-DevOps# docker build -t 3devopscloud .
[+] Building 4.1s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 98B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 326B
=> [1/2] FROM docker.io/library/nginx:latest@sha256:fb0117283ff38c2f9af91db1a7489459182a37c87cced5cb442d1d8fcc66d19
=> resolve docker.io/library/nginx:latest@sha256:fb0117283ff38c2f9af91db1a7489459182a37c87cced5cb442d1d8fcc66d19
=> sha256:5b5fa8b64d74b2ce9915f89c1be1b98a3480a7ca4fb4654ccc353db54342c2a9 1.40kB / 1.40kB
=> sha256:114e699da838b744a5dd7389723341d8f9a392ce236dd4bf2b6680d4f4065f0 1.21kB / 1.21kB
=> sha256:7382b41547b8fa58d4183ac9618d7e359eb989e675faf7e0d3e7445496bba94 953B / 953B
=> sha256:9ee66cc6c8558552f379f9868e31e578994118589e1f118a98396aeebcc9316b8d 482B / 482B
=> sha256:ee3a09d2248a2df379f9868e31e578994118589e1f118a98396aeebcc9316b8d 628B / 628B
=> sha256:5b219a92f92aeb674b0b29811b447d120ce69b41d21cfa56f90ad323aff91a2 29.99MB / 29.99MB
=> sha256:1733a4cd59840b3470ff7a90963bcddea5b543279dd6bdafe022d7883fdad221e5 29.78MB / 29.78MB
=> extracting sha256:1733a4cd59840b3470ff7a90963bcddea5b543279dd6bdafe022d7883fdad221e5
=> extracting sha256:5b219a92f92aeb674b0b29811b447d120ce69b41d21cfa56f90ad323aff91a2
=> extracting sha256:ee3a09d2248a2df379f9868e31e578994118589e1f118a98396aeebcc9316b8d
=> extracting sha256:7382b41547b8fa59d4183ac9618d7e359eb989e675faf7e0d3e7445496bba94
=> extracting sha256:9ee66cc6c8558552f379f9868e31e578994118589e1f118a98396aeebcc9316b8d
=> extracting sha256:114e699da838b744a5dd7389723341d8f9a392ce236dd4bf2b6680d4f4065f0
=> extracting sha256:5b5fa8b64d74b2ce9915f89c1be1b98a3480a7ca4fb4654ccc353db54342c2a9
=> [2/2] COPY index.html /usr/share/nginx/html
=> exporting image
=> exporting layers
=> exporting manifest sha256:1a540aa42d3842ae3174c49d5c6a2af65ccdd74cc598f397eff89fa06c1b658d4
=> exporting config sha256:bef3d6119d43ed483ec6421b4e9b4c5801b566044be9220200268cfd5b1ecf55
=> exporting attestation manifest sha256:3d75e46c452056e3bdf24099dd347fe6e310e645c73fffc3a36c16cb5a315c2d8
=> exporting manifest list sha256:8dde8559f825ef78fb3fcb407138532c7b69e938397b82c905f5e19025376f93
=> naming to docker.io/library/3devopscloud:latest
=> attaching to docker.io/library/3devopscloud:latest
root@ip-172-31-6-48:/home/ubuntu/tp4-DevOps#
```

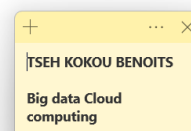
```

IMAGE          ID          DISK USAGE  CONTENT SIZE  EXTRA
3devopscld:latest 8dde8559f025 225MB       59.8MB
root@ip-172-31-6-48:/home/ubuntu# docker run -d --name tpdevopscld -p 80:80 3devopscld:latest
95b398bd7bbc42159e869ef8c13586b1e3d90b818b18b49bba39faa5991a7071
root@ip-172-31-6-48:/home/ubuntu# dockeer ps
Command 'dockeer' not found, did you mean:
  command 'docker' from deb docker.io (28.2.2-0ubuntu1~24.04.1)
  command 'docker' from deb podman-docker (4.9.3+ds1-1ubuntu0.2)
Try: apt install <deb name>
root@ip-172-31-6-48:/home/ubuntu# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
95b398bd7bbc   3devopscld:latest "/docker-entrypoint..." 23 seconds ago Up 22 seconds 0.0.0.0:80->80/tcp, [::]:80->80/tcp tpdevopscld
root@ip-172-31-6-48:/home/ubuntu#

```

### 3. Vérification de l'accès à l'application :

o Pour vérifier que l'application est accessible, je me suis rendu à l'adresse :



## Partie 2 : Automatisation CI/CD avec Jenkins

### 1. Installation de Jenkins sur l'instance EC2

#### 1. Installation de Java (Jenkins nécessite Java) :

```
root@ip-172-31-6-48:/home/ubuntu# sudo apt install openjdk-17-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-common
  at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service
  fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra
  fonts-dejavu-mono gsettings-desktop-schemas gtk-update-icon-cache
  hicolor-icon-theme humanity-icon-theme java-common libasound2-data
  libasound2t64 libatk-bridge2.0-0t64 libatk-wrapper-java
  libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3
  libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2
  libcups2t64 libdatrie1 libdconf1 libdeflate0 libdrm-amdgpu1
  libdrm-common libdrm-intel1 libdrm2 libfontconfig1 libgail18t64
  libgbm1 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-common libgif7 libgl1
  libgl1-mesa libglx-mesa0 libglx0 libgraphite2-3 libgtk2.0-0t64
  libgtk2.0-common libharfbuzz0b libice-dev libice6 libjpeg-turbo8
  libjpeg8 liblcms2-2 liblerc4 libllvmlibpangocairo-1.0-0
  libpangoft2-1.0-0 libpciaccess0 libpixman-1-0 libpthread-stubs0-dev
  librsvg2-2 librsvg2-common libsharpyuv0 libsm-dev libsm6 libthai-data
  libthai0 libtiff6 libvulkan1 libwayland-client0 libwayland-server0
  libwebp7 libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri3-0
  libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0
  libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev
  libxcomposite1 libxcursor1 libxdamage1 libxdmcp-dev libxfixes3
  libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2
  libxrender1 libxshmfence1 libxt-dev libxt6t64 libxtst6 libxv1
  libxxf86dga1 libxxf86vm1 mesa-libgallium mesa-vulkan-drivers
  openjdk-17-jdk-headless openjdk-17-jre openjdk-17-jre-headless
  session-migration ubuntu-mono x11-common x11-utils x11proto-dev
  xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre alsa-utils libasound2-plugins cups-common gvfs
  libice-doc liblcms2-utils libpangocairo-1.0-0 librsvg2-bin libsm-doc
  libx11-doc libxcb-doc
```

## 2. Installation de Jenkins :

```
root@ip-172-31-6-48:/home/ubuntu# curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee
curl: (22) The requested URL returned error: 404
root@ip-172-31-6-48:/home/ubuntu# "C
root@ip-172-31-6-48:/home/ubuntu# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2025-12-25 18:26:56-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)[146.75.34.133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/octet-stream]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyring 100%[=====] 3.10K --.-KB/s in 0s

2025-12-25 18:26:56 (33.9 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

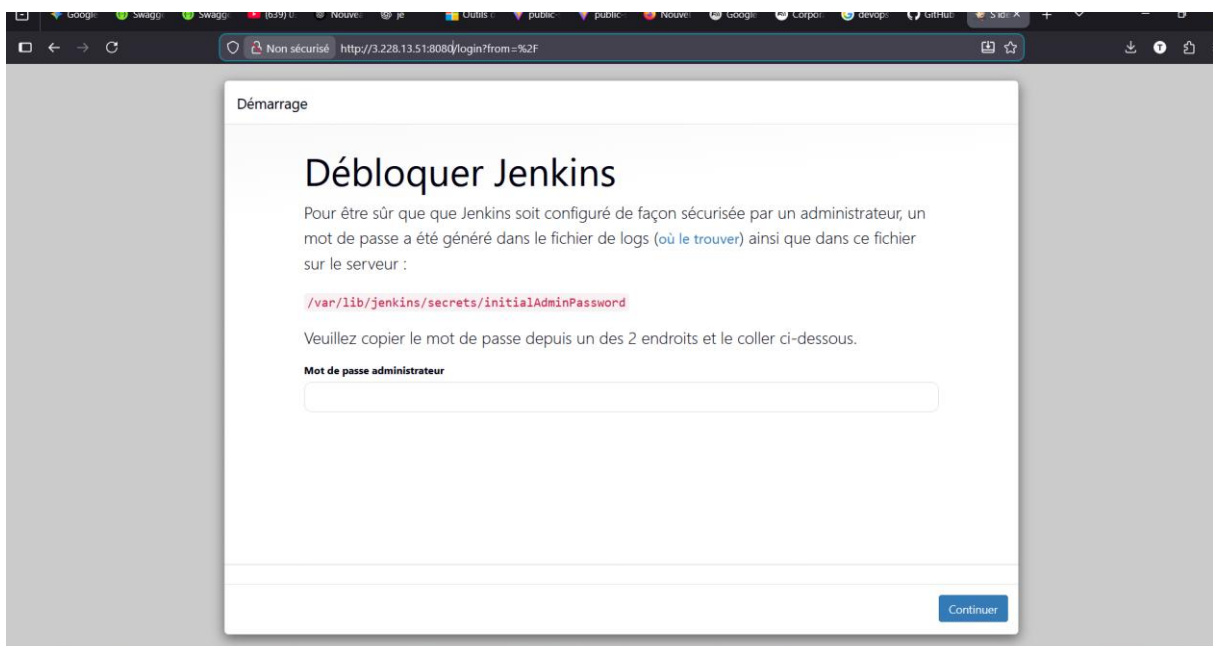
root@ip-172-31-6-48:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/" | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
root@ip-172-31-6-48:/home/ubuntu# sudo apt update
sudo apt install fontconfig openjdk-17-jre -y
sudo apt install jenkins -y;
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:8 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:9 https://pkg.jenkins.io/debian-stable binary/ Packages [30.3 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [71.4 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [288 B]
Fetched 253 kB in 1s (228 KB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
66 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
```

```
root@ip-172-31-6-48:/home/ubuntu# sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/li
b/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-6-48:/home/ubuntu# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; pres>
   Active: active (running) since Thu 2025-12-25 18:28:09 UTC; 4min 41s a>
   Main PID: 9227 (java)
   Tasks: 38 (limit: 1008)
   Memory: 303.4M (peak: 349.2M)
   CPU: 23.558s
   CGroup: /system.slice/jenkins.service
           └─9227 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/>

Dec 25 18:28:04 ip-172-31-6-48 jenkins[9227]: [LF]> This may also be found >
Dec 25 18:28:04 ip-172-31-6-48 jenkins[9227]: [LF]>
Dec 25 18:28:04 ip-172-31-6-48 jenkins[9227]: [LF]> *****>
Dec 25 18:28:04 ip-172-31-6-48 jenkins[9227]: [LF]> *****>
Dec 25 18:28:04 ip-172-31-6-48 jenkins[9227]: [LF]> *****>
Dec 25 18:28:09 ip-172-31-6-48 jenkins[9227]: 2025-12-25 18:28:09.375+0000 >
Dec 25 18:28:09 ip-172-31-6-48 jenkins[9227]: 2025-12-25 18:28:09.404+0000 >
Dec 25 18:28:09 ip-172-31-6-48 systemd[1]: Started jenkins.service - Jenkin>
Dec 25 18:28:09 ip-172-31-6-48 jenkins[9227]: 2025-12-25 18:28:09.585+0000 >
Dec 25 18:28:09 ip-172-31-6-48 jenkins[9227]: 2025-12-25 18:28:09.586+0000 >
lines 1-20/20 (END)
```

### 3. Configuration Initiale de Jenkins :

o J'ai accédé à l'interface Jenkins via le port 8080



```
root@ip-172-31-6-48:/home/ubuntu/tp4-DevOps# sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
a56b13cc2ae34e149211478e011dd15e  
root@ip-172-31-6-48:/home/ubuntu/tp4-DevOps# |
```

Non sécurisé http://3.228.13.51:8080

## Démarrage

# Personnaliser Jenkins

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

### Installer les plugins suggérés

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

### Sélectionner les plugins à installer

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

Jenkins 2.528.3

TSEH KOKOU BENOITS  
Big data Cloud computing

Démarrage

# Créer le 1er utilisateur Administrateur

Nom d'utilisateur

believer

Mot de passe

••••••••

Confirmation du mot de passe

••••••••

Nom complet

Tseh kokou benoit

Jenkins 2.528.3

Continuer en tant qu'Administrateur

Sauver et continuer



+ Nouveau Item

Historique des constructions

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

0/2



Ajouter une description

## Bienvenue sur Jenkins !

Vos jobs Jenkins seront affichés sur cette page. Pour commencer, vous pouvez mettre en place un build distribué ou commencer à créer un projet.

### Commencer à créer votre projet

Créer un job



### Configurer un build distribué

Mettre en place un agent



Configurer un cloud



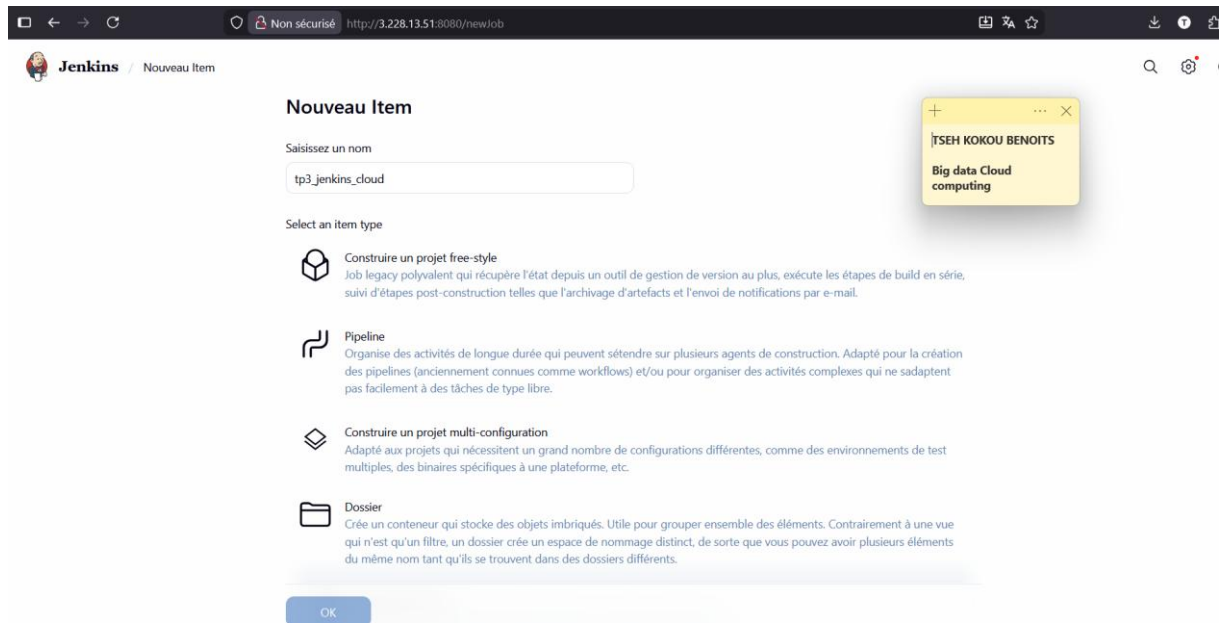
En apprendre plus sur les builds distribués



## 2. Création du Pipeline CI/CD dans Jenkins

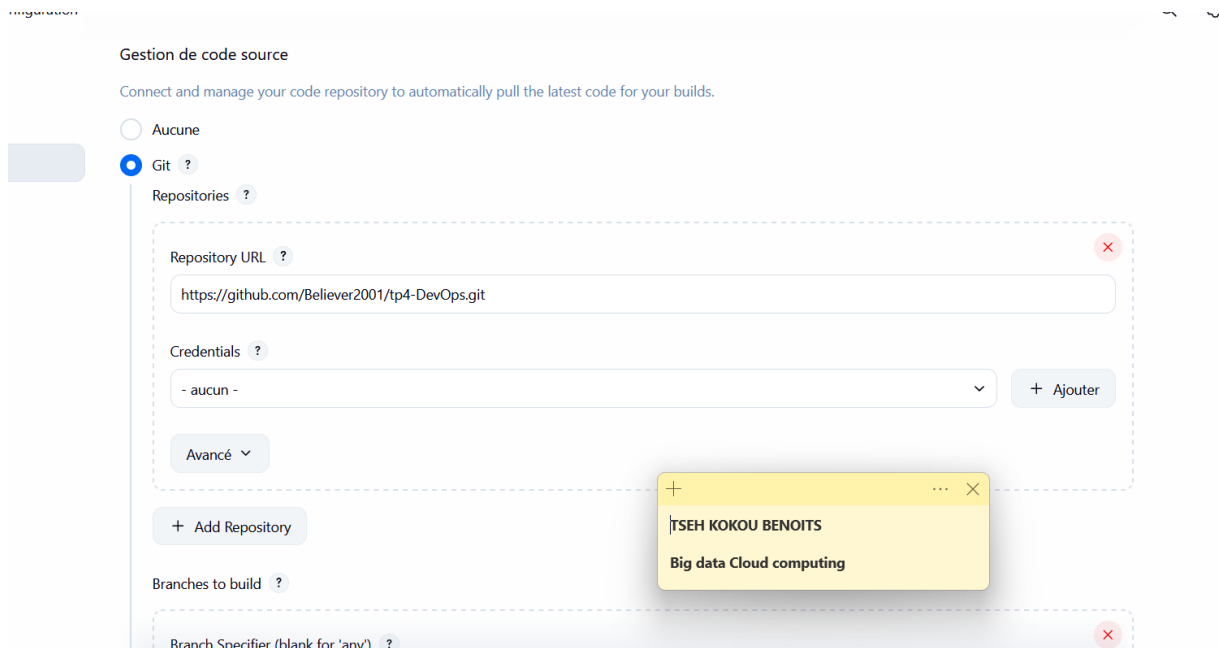
### 1. Création d'un Nouveau Projet :

o Dans Jenkins, j'ai créé un projet de type "Freestyle" pour configurer le pipeline.



## 2. Configuration de la Gestion du Code Source (Git) :

o J'ai configuré le dépôt GitHub dans la section "Source Code Management" du projet Jenkins.



## 3. Ajout d'Étapes de Construction et de Déploiement :

o Dans la section "Build", j'ai ajouté un script shell pour arrêter, supprimer, reconstruire et redémarrer le conteneur Docker :

VIEW FILE

Étapes du build

Automate your build process with ordered tasks like code compilation, testing, and deployment.

☰ Exécuter un script shell ?

✕

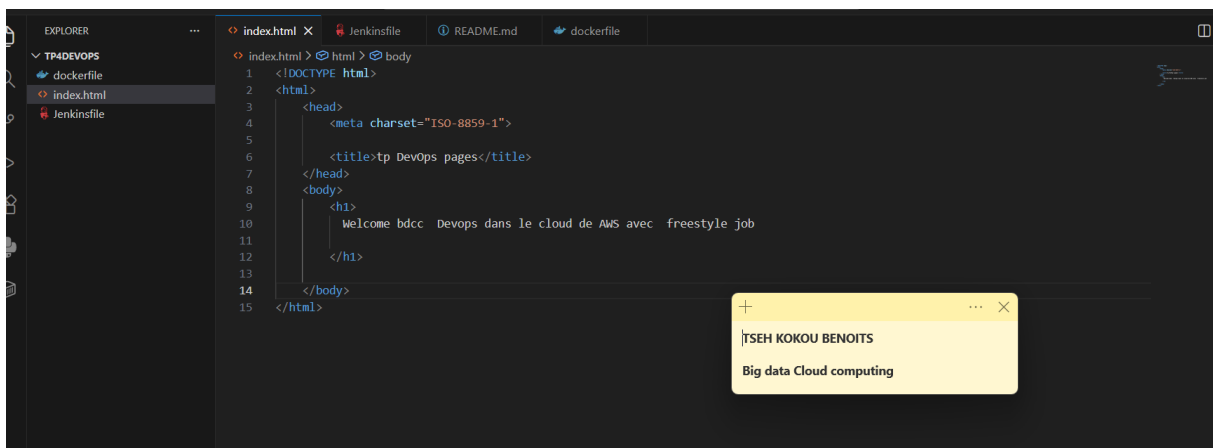
Commande

Voir la liste des variables d'environnement disponibles

```
docker stop tp3 || true
doker rm tp3 || true
doker build -t tp3 .
ocker run -d --name tp3 -p 9090:80 tp3
```

Avancé ▾

+ Ajouter une étape au build



On execute obligatoirement cette commande Pour ajouter Jenkins au groupe de docker pour donner l'autorisation Jenkins d'utiliser docker :

```
ubuntu@ip-172-31-6-48:~$ sudo usermod -aG docker jenkins
ubuntu@ip-172-31-6-48:~$ sudo systemctl restart jenkins
ubuntu@ip-172-31-6-48:~$
```

TSEH KOKOU BENOITS  
Big data Cloud computing

État

Modifications

Sortie de la console

Informations de la construction

Supprimer le build "#3"

Timings

Git Build Data

Build précédent

✓ #3 (1 janv. 2026, 11:12:22)



Lancé par l'utilisateur Tseh kokou Benoît



This run spent:

- 35 ms waiting;
- 7.7 s build duration;
- 7.7 s total from scheduled to completion.

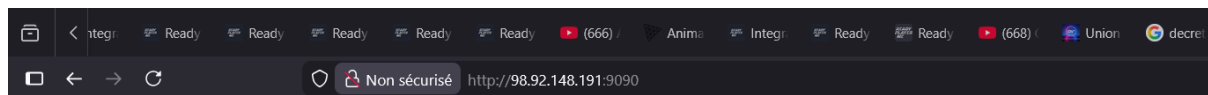


**Revision:** 71b940e996506926e10f9709e87844dc35f2055e  
**Repository:** <https://github.com/Believer2001/tp4-DevOps.git>

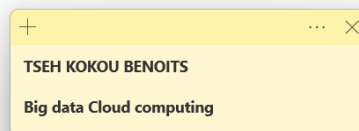
- refs/remotes/origin/main



Aucun changement.

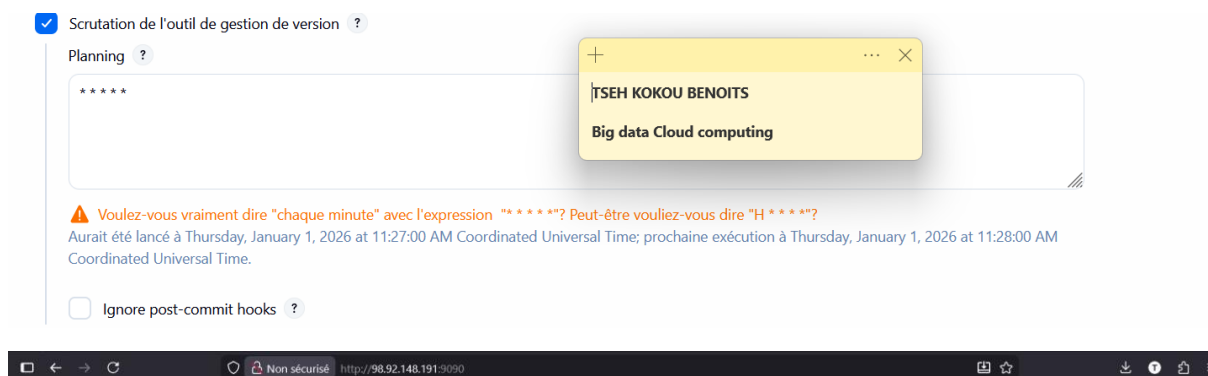


## Welcome bdcc Devops dans le cloud de AWS avec freestyle job



### 3. Déclenchement Automatique des Builds avec Webhooks GitHub

### 2. Configuration de Jenkins pour utiliser l'outil de scrutation :



## Welcome bdcc Devops dans le cloud de AWS avec freestyle job

J ai essayer de configurer le configurer l'outil de scrutation de git pour faire des build automatiquement



### **Conclusion :**

Ce TP m'a permis de comprendre et d'implémenter un pipeline CI/CD pour le déploiement d'une application web sur une instance AWS EC2. En suivant ces étapes, j'ai pu automatiser le processus de déploiement de l'application à chaque mise à jour du code source dans GitHub. Jenkins s'avère être un outil puissant pour gérer et automatiser les processus de déploiement, et Docker facilite grandement le déploiement des applications en environnement isolé.