# ESGI - Conteneurs et Devops

https://github.com/TiboJhjj/art-explorer.git

CI/CD pour \*Chicago Art Explorer\* – Guide d'installation (utilisateur \*\*tibo\*\*)

## 1. Vue d’ensemble

1. \*\*Installer Docker\*\* sur Debian 11

2. \*\*Lancer Jenkins\*\* dans un conteneur Docker

3. Effectuer la \*\*configuration initiale\*\* de Jenkins (déverrouillage, plugins, utilisateur)

4. \*\*Ajouter un agent distant\*\* via SSH (clé, credentials, nœud)

5. Mettre en place un \*\*pipeline type\*\* pour :

- exécuter les \*tests unitaires\*

- construire l’\*image Docker\* de production

- \*déployer\* automatiquement sur une VM distante

- envoyer une \*notification\* en cas d’échec de build

Cet environnement assure une CI/CD fiable pour l’application \*\*Chicago Art Explorer\*\* (Flask dockerisée).

---

## 2. Prérequis

- Machine hôte \*\*Debian 11\*\* (kernel ≥ 4.x)

- Accès \*\*sudo\*\* avec le compte \*tibo\*

- Ports \*\*8080\*\* (HTTP Jenkins) et \*\*50000\*\* (JNLP) ouverts

- \*\*VM distante\*\* accessible en SSH (agent & déploiement)

---

## 3. Installation de Docker sur Debian 11

Mettre à jour les paquets

sudo apt update && sudo apt upgrade -y

Installer les dépendances

sudo apt install -y apt-transport-https ca-certificates curl gnupg lsb-release

Ajouter la clé GPG Docker + dépôt officiel

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \

https://download.docker.com/linux/debian $(lsb\_release -cs) stable" \

| sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Installer Docker Engine

sudo apt update

sudo apt install -y docker-ce docker-ce-cli containerd.io

Ajouter l'utilisateur tibo au groupe docker

sudo usermod -aG docker tibo

Se déconnecter / reconnecter pour prise en compte

Vérifier

docker run hello-world

---

## 4. Lancement de Jenkins dans Docker

(Optionnel) Créer un réseau dédié

docker network create jenkins

Volume persistant

docker volume create jenkins\_home

Conteneur Jenkins LTS

docker run -d \

--name jenkins \

--restart unless-stopped \

--network jenkins \

-p 8080:8080 -p 50000:50000 \

-v jenkins\_home:/var/jenkins\_home \

jenkins/jenkins:lts

Mot de passe initial

docker exec jenkins cat /var/jenkins\_home/secrets/initialAdminPassword

Accédez ensuite à `http://192.168.45.130:8080` et collez le mot de passe.

---

## 5. Configuration initiale de Jenkins

1. \*\*Déverrouillage\*\* : saisir la clé initiale.

2. \*\*Plugins\*\* : \*Install suggested plugins\*.

3. \*\*Compte admin\*\* : créer un utilisateur administrateur.

4. \*\*Git\*\* :

- Installer \*\*Git Plugin\*\* si nécessaire

- \*Manage Jenkins → Global Tool Configuration\* → Git executable : `/usr/bin/git`

5. \*\*Plugins clés\*\* :

- Docker Pipeline

- SSH Agent

- Publish Over SSH (ou Slack / Telegram)

- Email Extension (notifications mail)

---

## 6. Ajout d’un agent SSH distant (utilisateur \*\*tibo\*\*)

6.1 Sur la VM distante

Installer OpenSSH

sudo apt update

sudo apt install -y openssh-server

sudo systemctl enable ssh --now

Préparer l'utilisateur tibo

sudo mkdir -p /home/tibo/.ssh

sudo chown tibo:tibo /home/tibo/.ssh

sudo chmod 700 /home/tibo/.ssh

6.2 Sur le serveur Jenkins

Générer la paire de clés (sans passphrase)

ssh-keygen -t ed25519 -f ~/.ssh/tibo\_agent -N ""

Copier la clé publique sur l’agent

ssh-copy-id -i ~/.ssh/tibo\_agent.pub tibo@192.168.45.130

6.3 Ajouter les identifiants SSH dans Jenkins

\*Manage Jenkins → Credentials → System → Global credentials → Add Credentials\*

| Champ | Valeur |

|-------|--------|

| \*\*Kind\*\* | SSH Username with private key |

| \*\*Username\*\* | `tibo` |

| \*\*Private Key\*\* | \*Enter directly\* — collez le contenu de `~/.ssh/tibo\_agent` |

| \*\*ID\*\* | `ssh-agent-key` |

| \*\*Description\*\* | Clé SSH pour agent distant |

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

Le contenu généré par l’IA peut être incorrect.

6.4 Déclarer le nœud

\*Manage Jenkins → Manage Nodes and Clouds → New Node\*

| Paramètre | Valeur |

|-----------|--------|

| \*\*Name\*\* | `agent-ssh` |

| \*\*Type\*\* | \*Permanent Agent\* |

| \*\*Remote root dir\*\* | `/home/tibo` |

| \*\*Launch method\*\* | \*Launch agents via SSH\* |

| \*\*Host\*\* | `<IP\_AGENT>` |

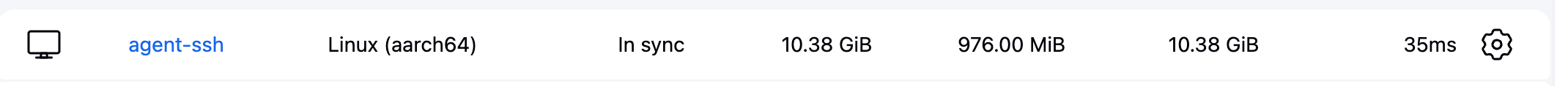
| \*\*Credentials\*\* | `ssh-agent-key` |

| \*\*Host Key Verification\*\* | \*Non verifying\* (tests) ou \*Manually trusted key\* |

Une image contenant texte, reçu, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Sauvegardez puis lancez ; le statut passe à \*\*online\*\*.



---

7. Pipeline type (aperçu rapide)

Le `Jenkinsfile` du dépôt automatise :

1. \*\*Tests unitaires\*\* (`pytest`)

2. \*\*Build & push\*\* d’une image Docker

3. \*\*Déploiement\*\* sur la VM via SSH

4. \*\*Notification\*\* (Discord / Slack / mail) en cas d’échec de build

Cela boucle l’intégration et la livraison continues pour \*Chicago Art Explorer\*.

---

> Document rédigé pour le compte \*tibo\* – dernière mise à jour : 2025.