

ESPRIT – UP ASI (Architecture des Systèmes d'Information)
Bureau E204

Module DevOps

Chapitre 2: Intégration Continue (CI)

Agents Jenkins (Utilisation distribuée de Jenkins)

Plan du cours

- Architecture distribuée de Jenkins (architecture Maître-Esclave)
- Agents Jenkins
- Gestion de builds distribués Jenkins

Atelier 3 : Agents Jenkins - Builds distribués

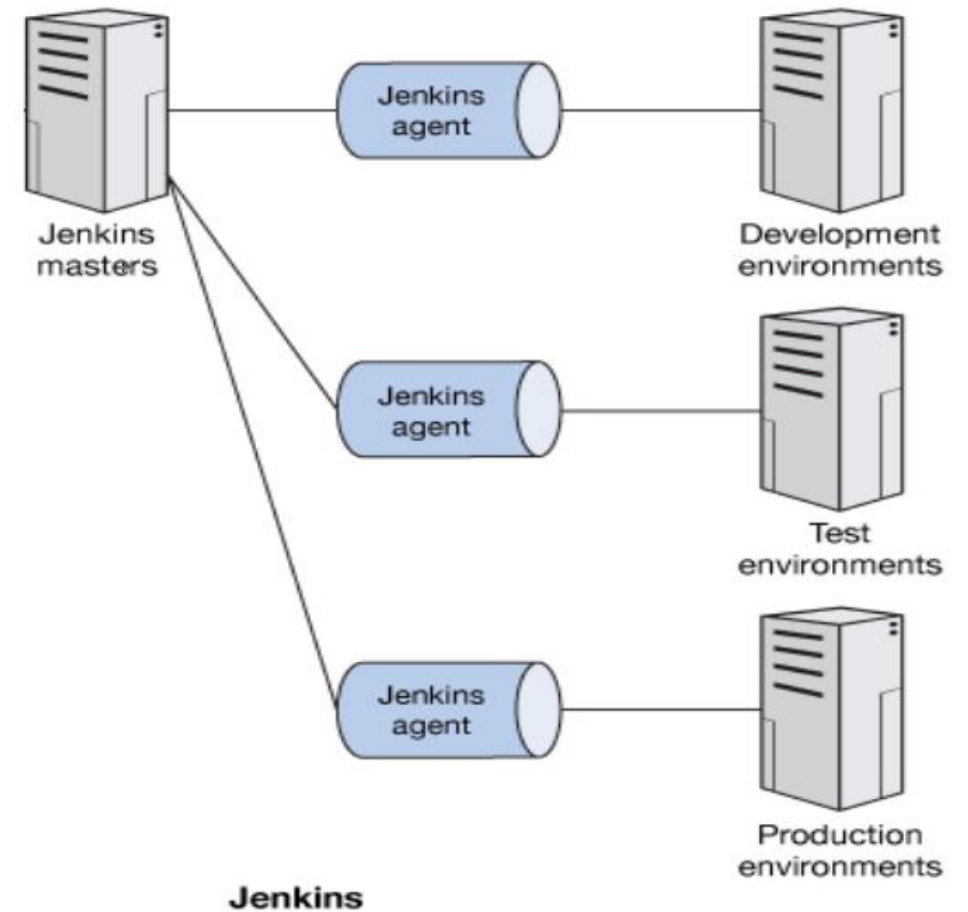
- Travail à faire !

Jenkins - Architecture distribuée

- Jenkins s'est imposé comme un acteur incontournable de l'Intégration Continue, de Livraison Continue et de Déploiement Continu.
- Jenkins est construit en suivant une architecture maître-esclave (**master-slave**) pour gérer les builds distribués.
- Il est infiniment scalable grâce à cet environnement **distribué** qui nous permet d'exécuter les futurs Jobs Jenkins sur des Agents (nœuds Agents) afin de soulager le maître (le serveur d'intégration continue Jenkins) qui est déjà très occupé à gérer l'interface Jenkins.

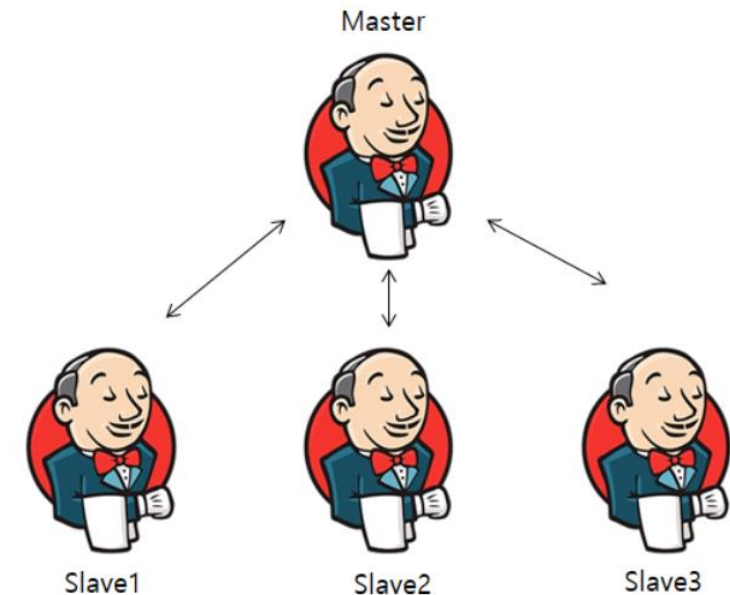
Jenkins - Architecture distribuée

- Cela peut particulièrement être utile pour différentier différents environnements :
Dev, Test, Recette, Préprod, Prod.
- Cette architecture est basée sur les AGENTS Jenkins.
- C'est une architecture Maître-Esclave.



Jenkins - Architecture Maître-Esclave

- L'architecture Maître-Esclave (master-slave) prend en charge la génération distribuée de builds pour réduire les charges sur le serveur Jenkins.
- **Environnement** : Une machine sur laquelle il y a un Jenkins 'master' et une autre machine sur laquelle il y a un agent 'slave':
 - **Master** : votre serveur Jenkins principal (celui que nous avons utilisé jusqu'à présent) est le maître.
 - **Slave** : un agent esclave est une instance légère de Jenkins qui fonctionne sur une machine distante et se met en écoute de requêtes de la part du maître Jenkins.

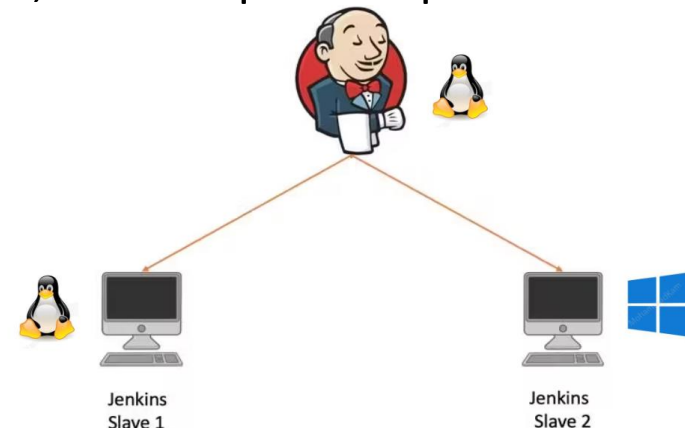


Jenkins - Missions du master et du slave

- **Master** : joue le rôle d'un serveur de contrôle. Il permet de :
 - gérer l'ordonnancement des tâches de build,
 - répartir les builds sur les esclaves pour leur exécution réelle,
 - surveiller les états des esclaves (en les mettant éventuellement hors-ligne si nécessaire),
 - enregistrer et présenter les résultats de build,
 - exécuter directement des tâches de build, même dans une architecture distribuée.
- **Slave** : le rôle des slaves est de faire ce qu'on leur dit, ce qui inclut :
 - l'exécution de tâches de build envoyées par le master.

Avantages de l'architecture distribuée de Jenkins

- **Soulager le serveur Jenkins** : répartir la charge de travail.
- **Parallélisme** : exécuter plusieurs builds simultanément sur différents agents.
- **Environnements divers** : différents agents peuvent être configurés dans des environnements variés.
- **Scalabilité** : une haute scalabilité permettant de s'adapter rapidement aux charges de travail changeantes de projets.
- **Optimisation des ressources** : en répartissant les tâches, l'utilisation des ressources est optimisée, car chaque agent gère une partie de la charge de travail.
- **Isolation des pannes** : si un agent rencontre un problème, cela n'a pas d'impact sur l'ensemble du processus CI ou sur d'autres builds en cours.



Jenkins - Terminologie

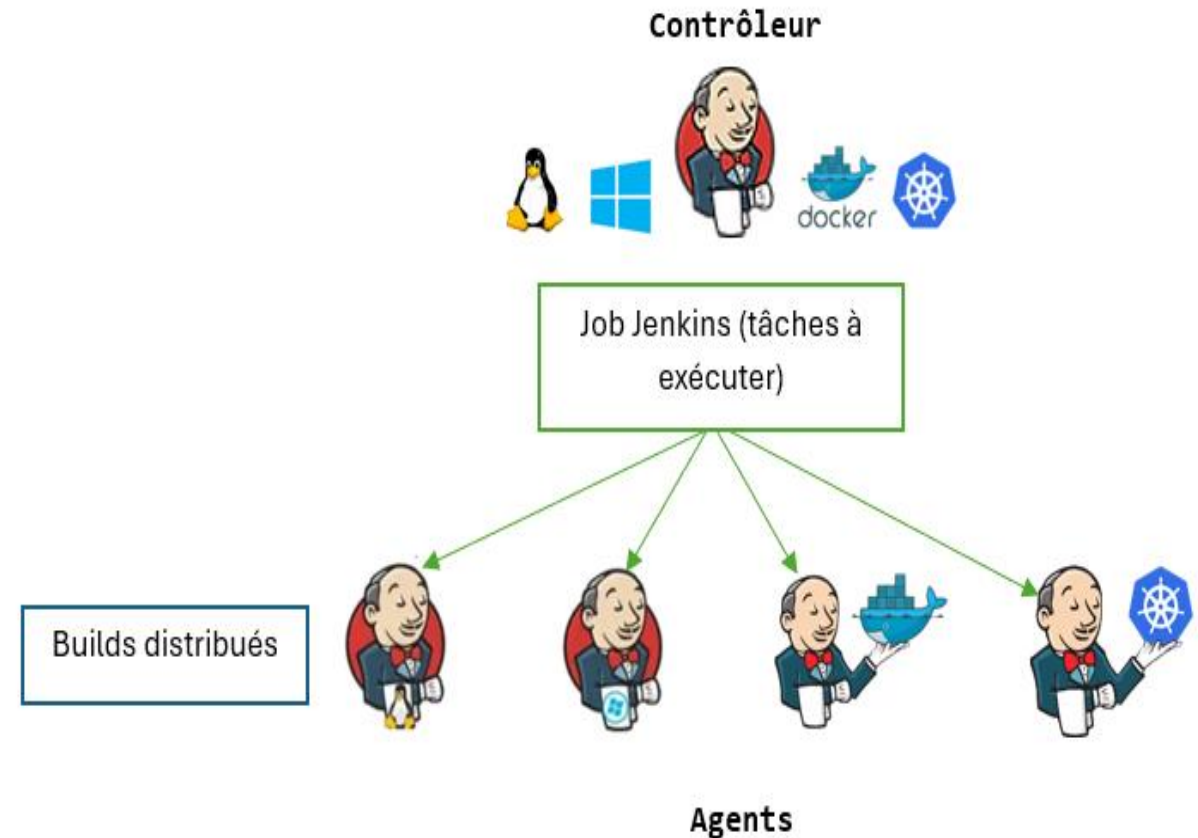
- **Master** (contrôleur) : serveur principal Jenkins qui gère et coordonne la chaîne d'intégration continue.
- **Slave** (agent) : machine ou conteneur utilisé par le « master » pour exécuter les étapes d'un projet.
- **Executors** : nombre de tâches pouvant être exécutées simultanément sur un agent (tâche d'un build, pipeline, projet).
- **Nœud** : fait référence à la machine ou au conteneur utilisé pour la configuration de l'environnement distribué Jenkins.
- **Job** (build) : ensemble des tâches définies par l'utilisateur pour construire un projet ou une partie d'un projet.

Jenkins - Différents types d'agents

Il existe deux types d'agents :

- les agents **statiques** : sont configurés et installés sur des machines (VM) directement.
- les agents **dynamiques** : sont provisionnés au besoin, sur des clusters Kubernetes ou dans des conteneurs Docker.

Le master peut aussi être installé sur un cluster Kubernetes, un conteneur docker ou une machine (en local, sur cloud).

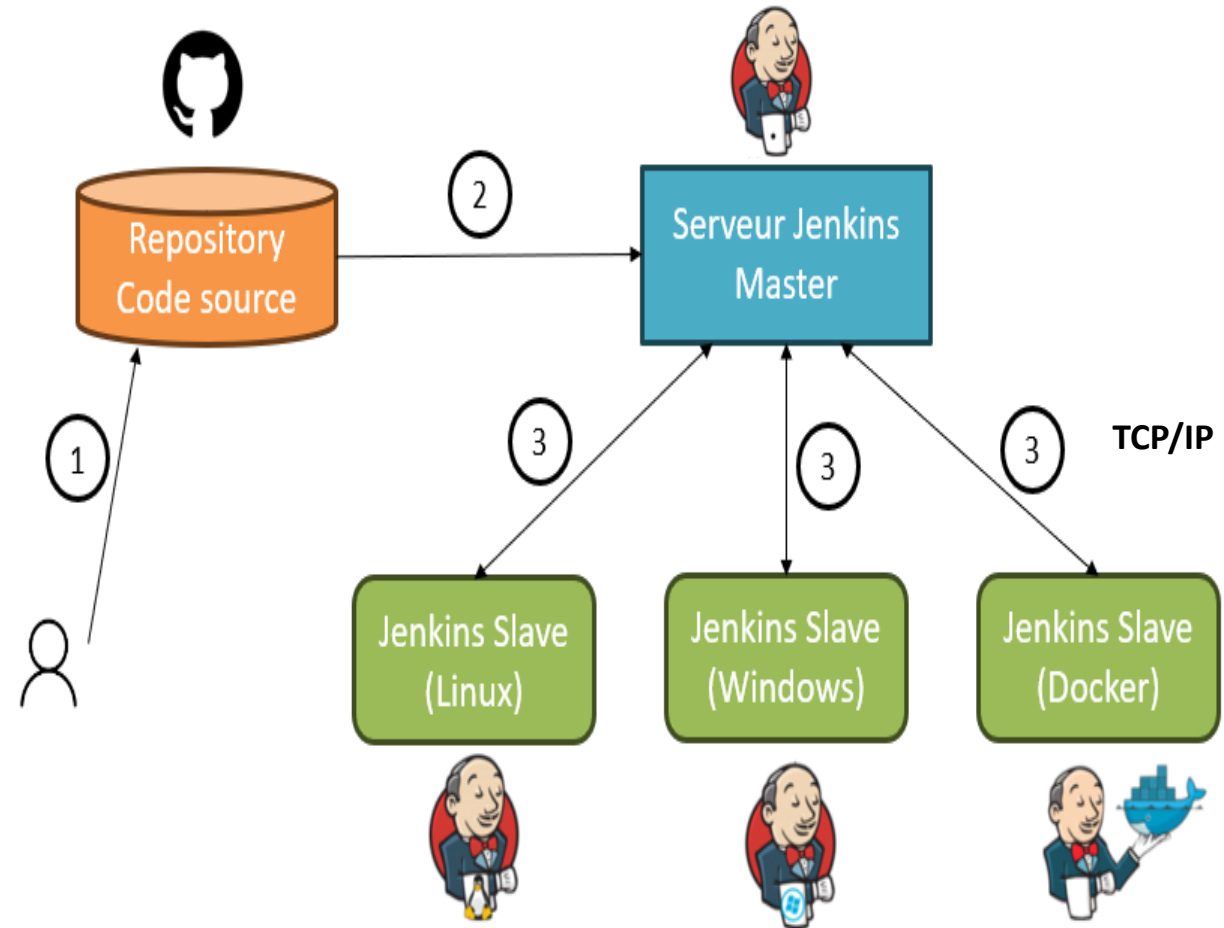


Jenkins - Stratégies de connexion master-slave

- Pour communiquer avec les agents, le master utilise soit le protocole SSH, soit le protocole JNLP ou soit avec des conteneurs Docker.
 1. Connexion **SSH** : Le maître démarre l'agent slave via le protocole SSH (la façon la plus pratique si vous travaillez dans un environnement Unix/Linux) → C'est la façon la plus commune de mettre en place une architecture de build distribuée.
 2. Connexion **JNLP** : Démarrer l'agent slave manuellement via Java Web Start (Cela fonctionne quel que soit le système d'exploitation de votre agent slave, toutefois c'est l'option la plus souvent utilisée pour les agents Windows). → Jenkins communique avec l'agent sur un port spécifique. Parfois un firewall bloque ce port. Vous pouvez forcer Jenkins à choisir un port aléatoire ou un port fixe que votre firewall autorise (voir Atelier 3).
 3. Connexion **Docker** : Les conteneurs Docker peuvent être utilisés comme des agents Jenkins. Chaque conteneur peut être lancé à la demande pour exécuter des tâches de build spécifiques, puis détruit une fois la tâche terminée (voir Chapitre 3 : Déploiement Continu - Docker).

Jenkins – CI dans un environnement distribué

1. Le développeur commit régulièrement les modifications dans le repository du code source,
2. Le serveur Jenkins récupère le code source pour gérer les différents **builds**. Jenkins ne peut pas gérer plusieurs builds simultanément,
3. Pour cela, le serveur Jenkins (master) distribue la charge de travail entre les agents Jenkins (slaves) pour exécuter les différents builds sur différents environnements.
→ les nœuds master et slave communiquent via le protocole **TCP/IP**.



Jenkins - Configuration de l'environnement distribué

- Installation et configuration de Jenkins (master).

JENKINS

(déjà fait, voir Atelier 1)

- Installation et configuration de l'agent (slave).

AGENTS

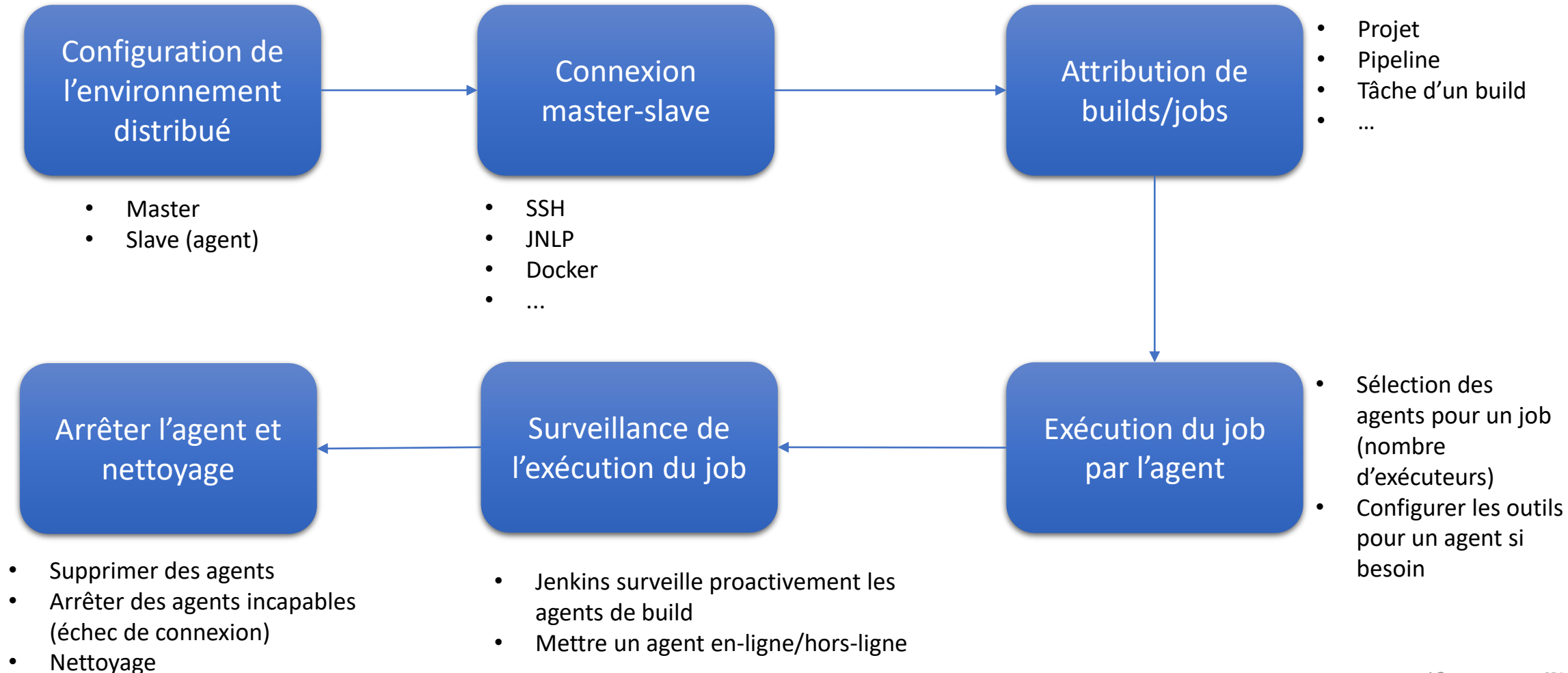
(différents environnements (machine ou conteneur) : Windows, Linux, Docker, ...)

- Configuration de la connexion entre Jenkins et l'agent (master et slave).

CONNEXION

Configuration d'une architecture master-slave

- La configuration d'une architecture maître-esclave comporte les éléments suivants :



Configuration de builds distribués - nouveau agent

- La première phase consiste à la mise en place de l'architecture maître-esclave : la création et la configuration des nœuds. Le nœud contrôleur (master) est déjà configuré, nous devons créer et configurer les agents :

Nom de l'agent

Nombre de job
en parallèle que
l'agent peut
exécuter

Nom logique pour grouper un
certain nombre d'agents en
fonction de de caractéristiques
telles que le système
d'exploitation

Tableau de bord > Nœuds > slave01 > Configurer

Nom ?

slave01

Description ?

C'est le premier Jenkins Slave

Texte brut [Prévisualisation](#)

Number of executors ?

2

Répertoire de travail du système distant ?

C:\agents

Étiquettes ?

slave_build

Configuration de builds distribués - connexion

- Cette phase permet de configurer la connexion entre le maître et les agents dans Jenkins (communication via JNLP) :

Stratégie de
connexion
maître-esclave
(JNLP)



Utilisation ?

Utiliser ce nœud autant que possible

Méthode de lancement ?

Launch agent by connecting it to the controller

Disponibilité ?

Maintenir l'agent activé le plus longtemps possible

Propriétés du nœud

- ☐ Disable deferred wipeout on this node ?
- ☐ Disk Space Monitoring Thresholds
- ☐ Emplacement des outils
- ☐ Variables d'environnement

Configuration des outils
pour un agent, par
exemple l'outil Git



Enregistrer

Configuration de builds distribués - connexion

- Cette phase permet de configurer la connexion entre le maître et les agents dans Jenkins (communication via SSH) :

Utilisation ?

Utiliser ce nœud autant que possible

Stratégie de connexion maître-esclave (SSH) →

Méthode de lancement ?

Launch agents via SSH

Configurer les identifiants (Credentials) →

Host ?

192.168.33.10

Credentials ?

vagrant (jenkins_key)

+ Ajouter ▾

Host Key Verification Strategy ?

Manually trusted key Verification Strategy

Stratégie de vérification de la clé SSH →

☐ Require manual verification of initial connection ?

Avancé ▾ Edited

Configuration de builds distribués - distribution

- Cette phase permet d'associer une tâche de build avec un agent ou un groupe d'agents. Par défaut, Jenkins utilisera simplement le premier agent slave disponible, ce qui offre généralement le meilleur temps de traitement global.

Pour attacher une tâche de build à agent particulier, vous devez cocher cette case et indiquer l'étiquette de l'agent



☒ Restreindre où le projet peut être exécuté ?

Expression ?

slave02

Label slave02 respecte 1 node. Les droits ou autres restrictions apportées par des plugins peuvent réduire cette liste.

Avancé ▾

Utiliser les opérateurs et (&&) et ou (!!) pour combiner les étiquettes d'un groupe d'agents

Résultat du build - Job/Build distribué Jenkins

Agent est connecté

Etiquette de l'agent

Job réussi

Agent slave01

C'est le premier Jenkins Slave

Agent is connected.

Monitoring Data ▾

Labels

slave_build

Projets rattachés à slave01

| S | M | Nom du projet ↓ | Dernier succès | Dernier échec | Dernière durée |
|---|---|-----------------|----------------|---------------|----------------|
| ✓ | ☁ | JobFree | 1 mn 45 s #29 | 5 mn 8 s #28 | 2.1 s |

Marquer ce nœud comme temporairement hors ligne

?

✎ modifier la description

Etats des nœuds
(contrôleur et agent)

Nodes

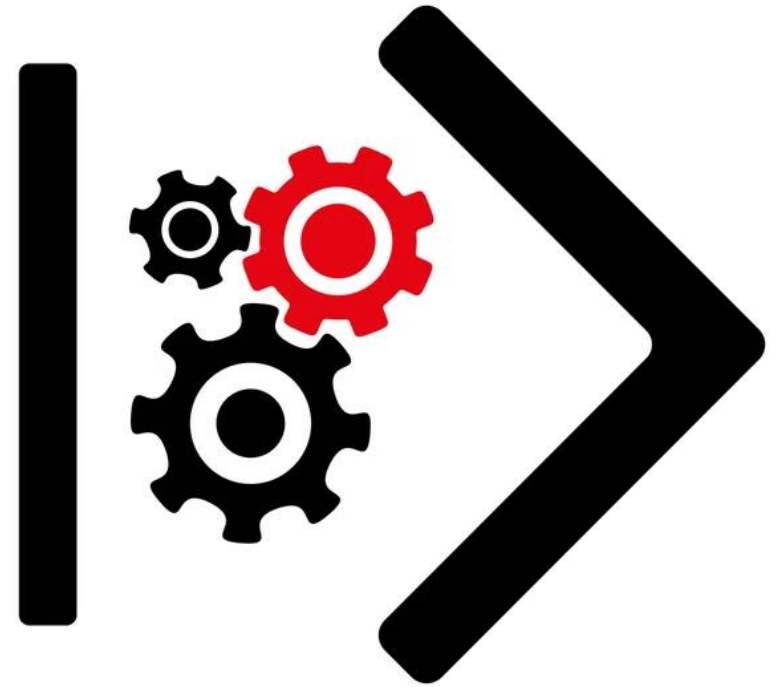
+ New Node

Configure Monitors

↺

| S | Nom ↓ | Architecture | Différence entre les horloges | Espace disque disponible | Espace de swap disponible | Free Temp Space | Temps de réponse |
|---|------------|--------------------|-------------------------------|--------------------------|---------------------------|-----------------|------------------|
| 🖥 | contrôleur | Linux (amd64) | Synchronisé | 22.37 GiB | 2.00 GiB | 22.37 GiB | 0ms ⚙ |
| 🖥 | slave01 | Windows 10 (amd64) | Synchronisé | 38.48 GiB | 4.27 GiB | 38.48 GiB | 84ms ⚙ |

Construction et configuration d'agents Jenkins - Builds distribués



Atelier -3-

Travail à faire !



Activité 3 : Construction et configuration de builds distribués : agents Jenkins.

"Apprendre par le projet, c'est découvrir
par l'action, créer par la compréhension, et
réussir par la persévérance."