

# DevOps - Pipeline

# Objectifs

- Les Pipeline
  - Théorie
  - Démo

# Intégration continue / déploiement continu

- L'intégration continue / Continuous integration (CI) :
  - Générer des artefacts à partir d'un gestionnaire de source / source control management (SCM)
  - Exécuter les tests
- Le déploiement continu / continuous delivery (CD) :
  - Prendre les artefacts
  - Les déploiements dans les environnements ciblés
  - Souvent, il y a un flux de travail pour migrer sur plusieurs environnements avec des validations à chaque niveau

On parle souvent de CI/CD



## Pipeline : Agent

Quand votre build ou déploiement s'exécute, le système démarre un ou plusieurs travaux. Un agent calcule l'infrastructure avec un logiciel d'agent installé qui exécute un travail à la fois. Par exemple, votre travail peut s'exécuter sur un agent Ubuntu hébergé par Microsoft.

Image	Spécification de l'agent de l'éditeur classique	Étiquette de l'image de machine virtuelle YAML
Windows Server 2022 avec Visual Studio 2022	<i>Windows-2022</i>	<code>windows-2022</code>
Windows Server 2019 avec Visual Studio 2019	<i>Windows-2019</i>	<code>windows-latest</code> OU <code>windows-2019</code>
Ubuntu 20.04	<i>Ubuntu-20,04</i>	<code>ubuntu-latest</code> OU <code>ubuntu-20.04</code>
Ubuntu 18.04	<i>Ubuntu-18,04</i>	<code>ubuntu-18.04</code>
macOS 11 Big sur	<i>macOS-11</i>	<code>macOS-latest</code> OU <code>macOS-11</code>
macOS X Catalina 10,15	<i>macOS-10,15</i>	<code>macOS-10.15</code>

## Déploiement

Pour les pipelines classiques, un déploiement est l'action d'exécuter les tâches pour une étape, qui peut inclure l'exécution de tests automatisés, le déploiement d'artefacts de build et toute autre action spécifiée pour cette étape.

## Environnement

Un [environnement](#) est une collection de ressources, dans laquelle vous déployez votre application. Il peut contenir une ou plusieurs machines virtuelles, conteneurs, applications Web ou tout autre service utilisé pour héberger l'application en cours de développement. Un pipeline peut déployer l'application dans un ou plusieurs environnements une fois la génération terminée et les tests exécutés.

Dev, test, QA, Staging et production sont des exemples typiques de noms d'environnement.

- 1 Repos
- 2 Pipelines
- 2 Pipelines
- 2 Environments
- Releases
- Library
- Task groups
- Deployment groups
- Test Plans
- Artifacts
- Compliance



## Create your first environment

Manage deployments, view resource status and get end to end traceability

Create environment

# Définition des environnements : Dev ou Develop

- **Development** : il s'agit de la **version de travail** (working copy) du logiciel ou de l'application web se trouvant soit sur un serveur sécurisé soit sur un ordinateur local.
- Les développeurs peuvent procéder à des **modifications du code source** et télécharger des fonctionnalités individuelles. Étant donné que la version de travail est souvent exploitée avec des compilateurs, des outils de test et de débogage, cette dernière ne fonctionne uniquement qu'au sein de cet environnement.
- Les développeurs disposent souvent d'une **base de données réduite** pour vérifier des fonctions en particulier. Pour le développement d'une boutique en ligne, une partie du catalogue de produits peut servir à pouvoir transférer la méta-structure des données comme les champs pour le prix, la disponibilité ou encore différentes variantes de produits.
- Lorsque plusieurs développeurs travaillent sur des gros projets, ils utilisent souvent un système de contrôle de version comme **Git**, Mercurial ou Subversion pour synchroniser les modifications et les mises à jour.

Source : [Ryte Wiki - Wiki du marketing digital](#)



# QA (Quality and Assurance)

- Cet environnement optionnel sert à vérifier si les applications contiennent des bugs ou si des erreurs se retrouvent dans le code, sans que cette étape de travail ait un quelconque impact sur l'environnement de développement.
- Si une personne est par exemple en train de tester une fonction et qu'un autre procède à des modifications sur cette même fonction, le système ne serait alors plus fonctionnel. C'est pourquoi les deux étapes de travail sont séparées l'une de l'autre.
- La vérification qualité (QA) sert cependant aussi à assurer la sécurité du système. Des équipes entières sont souvent utilisées, ces dernières reçoivent un accès au système – comme pour un test bêta, seulement sans utilisateurs réels.

# Staging

- C'est l'environnement de simulation.
- Cette **version est le « candidat »** pour le lancement de l'application.
- Généralement, l'environnement de simulation est identique à celui de production, si bien que le matériel et les logiciels ne présentent pas de grandes différences lors de l'utilisation de l'application.
- Pour finir, cela a aussi pour objectif de vérifier la connectivité au sein de l'ensemble du système : par exemple l'accès à la base de données et l'interaction avec la périphérie. Étant donné que la performance dépend beaucoup de l'environnement, les temps de chargement et autres critères de performance sont aussi examinés lors du staging. Lorsque l'assurance qualité est supprimée – et cela est le cas dans la plupart des projets – les tests se font avec le staging. Dans ce cas-là, le staging est en quelque sorte la répétition générale, avant que l'application migre vers l'environnement de production.

# Production :

- l'environnement de production est idéalement une copie conforme de l'environnement staging. Dans la pratique, cela n'est certes pas toujours possible, mais le hardware et le software sont censés autant l'un que l'autre être très semblables aux composants utilisés dans le staging. Ceci est la seule manière d'exclure les conflits. À cette étape, plus aucune modification n'est faite par les développeurs car l'ancienne copie de travail du code source se trouve déjà sur le serveur en ligne. Le code source a pu être publié car ce dernier a été largement testé et amélioré au cours des différents environnements.

# Où sont c'est environnement ?

- Azure
- AWS
- Autres
- Privé.

# Démo

- Les sources : git remote -v
- Azure devops, Setting dépôt et **Service connections**
- Pipeline Nouveau pipeline

# Références

- [Azure Pipelines nouveau Guide de l'utilisateur-concepts clés - Azure Pipelines | Microsoft Docs](#)
- [Documentation Azure Pipelines | Microsoft Docs](#)