# Présentation Azure DevOps

Table des matières

[Présentation Azure DevOps 1](#_Toc20986951)

[Introduction 3](#_Toc20986952)

[CI/CD 3](#_Toc20986953)

[DevOps 3](#_Toc20986954)

[Introduction à Azure DevOps 4](#_Toc20986955)

[Tableau de bord 4](#_Toc20986956)

[Compilation 4](#_Toc20986957)

[Requête de tirage 5](#_Toc20986958)

[Déploiement 5](#_Toc20986959)

[Azure DevOps versus tfs2012 5](#_Toc20986960)

[Formation Azure DevOps 5](#_Toc20986961)

[Formation Git 6](#_Toc20986962)

[Pour demeurer à jour 6](#_Toc20986963)

# 

# Introduction

* Ce n’est pas un cours magistral, je m’attends à ce que l’on philosophe ensemble
* Si vous avez des interventions à faire, seulement à lever la main pour avoir la parole
* Si vous avez des questions que je n’ai pas la réponse, je vais la prendre en note et m’assurer de vous revenir avec une réponse

# CI/CD

* Un des grands principes est l’automatisation des processus
* Définition de l’intégration continue : un processus d'automatisation pour les développeurs. L'intégration continue consiste, pour les développeurs, à apporter régulièrement des modifications au code de leur application, à les tester, puis à les fusionner dans un référentiel partagé. Cette solution permet d'éviter de travailler en même temps sur un trop grand nombre d'éléments d'une application, qui pourraient entrer en conflit les uns avec les autres.
* Déploiement continu : peut désigner le transfert automatique des modifications du développeur depuis le référentiel vers l'environnement de production, où elles peuvent être utilisées par les clients. Ce processus permet de soulager les équipes d'exploitation surchargées par les tâches manuelles qui ralentissent la distribution des applications. Il repose sur la distribution continue et automatise l'étape suivante du pipeline.
* Il suffit de retenir que l'approche CI/CD se rapporte à un processus, souvent représenté sous forme de pipeline, qui consiste à introduire un haut degré d'automatisation et de surveillance continues dans le processus de développement des applications. La signification réelle de ces termes varie au cas par cas, selon le niveau d'automatisation du pipeline CI/CD. De nombreuses entreprises commencent par l'intégration continue, puis se mettent peu à peu à automatiser la distribution et le déploiement, par exemple dans le cadre du développement d'applications natives pour le Cloud.

# DevOps

* Précision sur la terminologie, DevOps provient de l’anglais, « Development and Operations »
* Dans un modèle DevOps, les équipes de développement et d'opérations ne sont plus isolées. Il arrive qu'elles soient fusionnées en une seule et même équipe. Les ingénieurs qui la composent travaillent alors sur tout le cycle de vie d'une application, de la création à l’exploitation, en passant par les tests et le déploiement, et développent toute une gamme de compétences liées à différentes fonctions. Dans certains modèles DevOps, les équipes d'assurance qualité et de sécurité peuvent également s'intégrer étroitement au développement et aux opérations, ainsi qu’à l’ensemble du cycle de vie des applications. Lorsque la sécurité est au cœur de l’activité d’une équipe DevOps, on parle parfois de DevSecOps. Ces équipes utilisent des pratiques pour automatiser des processus qui étaient autrefois manuels et lents. Elles exploitent une pile technologique et des outils qui les aident à faire fonctionner et à faire évoluer les applications de façon rapide et fiable. Ces outils aident également les ingénieurs à accomplir de façon autonome des tâches (par exemple, le déploiement de code ou la mise en service d'infrastructure) qui nécessiteraient normalement l'aide d'autres équipes, ce qui augmente encore davantage la productivité de l’équipe d’ingénieurs.

# Introduction à Azure DevOps

* Software as a service - Saas : Est un modèle d'exploitation commerciale des logiciels dans lequel ceux-ci sont installés sur des serveurs distants plutôt que sur la machine de l'utilisateur
* Application Lifecycle Management - ALM : Couvre l'ensemble des moyens nécessaires au développement ou à la maintenance d'une application
* Fonctionnalités possibles à même l’outil :
  + Azure Boards : <https://azure.microsoft.com/en-us/services/devops/boards/>
  + Azure Repos : <https://azure.microsoft.com/en-us/services/devops/repos/>
  + Azure Pipeline : <https://azure.microsoft.com/en-us/services/devops/pipelines/>
  + Azure Tests Plans : <https://azure.microsoft.com/fr-ca/services/devops/test-plans/>
  + Azure Artifacts : <https://azure.microsoft.com/en-us/services/devops/artifacts/>
* Facilite la collaboration : Outil facilement utilisable par tous les secteurs nécessaires à la livraison d’une application. Que ce soit par l’équipe de développement, les clients, les gens des opérations ou même les dirigeants tout le monde ont accès!

# Tableau de bord

* Est 100% configurable, et ce par équipe
* Plusieurs modules existent et dans le cas où vous n’avez pas ce que vous recherchez, vous pouvez toujours aller jeter un coup d’œil sur le marketplace. L’écosystème est très grand, donc la chance de trouver ce dont vous avez besoin dans le marketplace est vraiment grande.
* L’information sur le tableau de bord peut servir à plus que l’équipe de développement, celle-ci peut être utile pour les gestionnaires et les clients pour suivre l’évolution d’un sprint.

# Compilation

* Dans le sommaire d’une compilation, on retrouve plein d’information comme : le temps d’exécution pour chacune des étapes de la compilation, la couverture de code, le déploiement associé à cette compilation, le type d’action dans lequel la compilation a été lancée (automatique ou manuelle et par qui), les changements associés à la modification
* Les compilations sont par « produits »
* Pourquoi que c’est nécessaire d’avoir une compilation pour l’assurance qualité? Un des gains c’est le temps de compilation, l’activité de publier le code produit sur le serveur distant est l’activité la plus longue et cette étape n’est pas nécessaire dans le processus de l’assurance qualité. De plus, on peut ajouter des étapes pour refuser automatiquement une modification si celle-ci ne respecte pas les standards de l’entreprise. Par exemple, si la couverture de code est sous les normes définies ou bien que le code n’est pas suffisamment de qualité à l’aide d’outil d’analyse de code comme [SonarCube](https://www.sonarqube.org/).
* Ce qui est bien c’est que l’on peut vraiment compiler n’importe quoi, même une application IOS qui demande un mac pour compiler!

# Requête de tirage

* Personnellement, je changerais n’importe quand de produit pour utiliser Azure DevOps seulement pour le processus d’assurance qualité tellement que c’est bien fait!
* Liste des règles possibles lors de la soumission de code : un nombre minimal de réviseurs, s’assurer qu’un élément de travail soit associé au changement, s’assurer que tous les commentaires soient résolus, préciser le type de merge que l’on veut enforcer dans l’équipe (squash merge par exemple), définir une compilation pour l’assurance qualité, utiliser un autre produit pour l’autorisation comme ([ServiceNow](https://www.servicenow.fr/) – service d’autorisation) et inclure automatiquement des réviseurs à une demande de changement au code
* L’interface est en direct, donc lors de la rédaction d’un commentaire, la personne le voit directement à son écran sans devoir faire un rafraichissement de la page. ([SignalR](https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr) est utilisé)
* La syntaxe prise en charge est le Markdown, donc il est facile d’inclure des images, des liens, des Gifs et du code
* Il est facile et convivial d’avoir des discussions sur la plateforme à fin d’avoir une historique de nos changements et de laisser une trace sur les décisions prises en équipe
* Lors d’un merge vers master c’est le commit de l’assurance qualité qui est inscrit, il est donc très facile de retrouver une requête de tirage dans le temps.

# Déploiement

* Chaque « environnement » correspond à un stage
* On peut mettre des checks en amonts et après chaque stage
* Un déploiement en différé c’est par exemple de dire, S.V.P est-ce que tu pourrais me déployer telle modification ce soir à 18h?
* Chaque stage peut contenir ses propres variables pour le déploiement ce qui fait en sorte qu’il est facile de paramétrer les environnements et favorise grandement la réutilisation entre les différents stages
* Le déploiement est souvent fait par [Robocopy](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/robocopy) (tâche PowerShell) dans les environnements Windows (même linux avec [PowerShell Core](https://github.com/PowerShell/PowerShell) qui est multiplateforme)
* Déployer sur AWS Amazon, Azure ou autre Cloud sans problème
* Si on veut déployer sur son infrastructure à l’interne, l’organisation se doit d’avoir son serveur de compilation avec des agents qui eut pourront faire le travail.

# Azure DevOps versus tfs2012

* Qu’est-ce que TFS 2012 : TFS veut dire Team Foundation Server. C’est comme Azure DevOps, mais que l’application est hébergée chez les clients. La dernière version du produit s’appelle maintenant Azure DevOps server 2019.
* Il y a beaucoup de pression de la communauté chinoise pour avoir une version chinoise de l’application, mais c’est actuellement un problème au niveau de la localisation BD qui freine Microsoft à aller de l’avant pour le moment.

# Formation Azure DevOps

* Je ne m’attends pas à ce que tout le monde écoute toutes ces vidéos, mais les deux vidéos sur la chaîne « Bracket Show » sont vraiment bien.
* Prendre note que le produit évolue très rapidement et il est fort possible de retrouver des exemples qui ne correspondent plus à la réalité d’aujourd’hui.

# Formation Git

* Je conseille également de suivre la chaîne « [The Coding Train](https://www.youtube.com/channel/UCvjgXvBlbQiydffZU7m1_aw) »
* Sourcetree : il est gratuit et nécessite seulement un compte Atlassian
* GitKraken : La version « pro » est gratuite pour les étudiants
* Git Lens : Très simple à utiliser!

# Pour demeurer à jour

* Prendre note que le service évolue rapidement et un bon moyen de se garder au courant des nouvelles fonctionnalités c’est de lire le blogue et de suivre le roadmap du produit.