



Aujourd'hui, je vous propose d'aborder la solution **Azure DevOps (ADO)**.

Ah super, en plus, c'est un outil très largement utilisé.

Oui c'est vrai.

ADO est une plateforme composée de **différents services**, proposée par Microsoft pour simplifier le **cycle de vie de développement**.

Elle est utilisée uniquement par des développeurs ?

Non justement.

Elle permet d'améliorer la collaboration entre les équipes de Dev d'un côté, et d'Exploitation de l'autre.

D'où le terme DevOps ?!

Exactement, c'est la contraction de **Dev** et **Ops**.

ADO propose un ensemble d'outils et de services complètement intégrés.

Si je comprend bien, cela évite de faire appel à une multitude d'outils ?!

Tu as tout compris.

D'ailleurs, ADO est disponible en tant que solution SAAS, mais tu peux aussi l'installer sur un serveur.

C'est génial pour ceux qui rencontrent des contraintes techniques ou de sécurité.

En effet.

Essayons de parler des différents outils disponibles dans ADO.

Allez on t'écoute attentivement.



Débutons avec **Azure Boards**, qui permet aux équipes, de planifier et suivre des actions sur l'ensemble du processus de Développement.

Est il possible d'y gérer les méthodologies agile ?

Tout à fait comme Scrum et Kanban.

Tu peux gérer différents **Product Backlog Item (PBI)** comme les User Story, Task ou encore Bug.

Génial, en plus on peut les suivre via différents Boards.

Exactement, ce qui permet en un coup d'oeil, d'avoir un état d'avancement.

Et pour la gestion du code ?

C'est **Azure Repos** qui permet de versionner ton code.

Tu peux suivre les modifications apportées au code au fil du temps, avec la gestion des branches et tout ce qui va avec.

Cela me fait penser à **GitHub** ?

Oui, tout à fait.

D'ailleurs, Azure Repos offre deux types de gestion de version, que sont Git et Team Foundation version Control (TFVC).

Et concernant les déploiements ?



C'est géré au travers de **Azure Pipelines**.

Ce dernier permet de gérer les workflows **CI/CD**.

CI/CD ?

L'**intégration continue** ou **CI**, permet d'intégrer les modifications de ton code de manière automatisée.

La **livraison continue** ou **CD**, permet de générer ton code, le tester et le déployer sur un ton environnement de test ou de production.

Merci pour les explications !



Azure Test Plans quant à lui, permet de planifier des tests, manuels ou automatiques, de ton application, via un navigateur.

Comme pour simuler l'activité d'un utilisateur ?

Absolument.

Et enfin, **Azure Artefacts** permet de consommer ou de publier des packages.

Un peu à la manière de registres publics comme NuGet.org ou npmjs.com

Oui, décidément, tu as une bonne culture DevOps !!

Un peu, je l'avoue.

Et en terme de sécurité, qu'est ce qui est proposé ?

L'une des dernières fonctionnalités se nomme **GitHub Advanced Security for Azure DevOps**.

Elle permet de vérifier qu'il n'y a pas de secrets lors des push ou dans Azure Repos.

Elle recherche d'éventuelles vulnérabilités dans les dépendances open source.

Et enfin, elle permet d'identifier des vulnérabilités au niveau application comme l'**injection SQL**, par exemple.

Génial !

Azure DevOps est vraiment une solution fantastique.

Il permet de **collaborer**, de **gérer le cycle de vie** de ton code durant **toutes les étapes de développement**, c'est vraiment cool.





Si vous souhaitez continuer à **apprendre**, de façon ludique, sur **l'écosystème Azure**, et ne rater aucune de nos illustrations ...

... N'hésitez pas à vous abonner sur :



<https://aka.ms/grow-una>



<https://www.youtube.com/@grow-una>

Et si le contenu vous plaît, partagez-le ;o)

A très vite !

