

Azure DevOps for Teenagers

**CONVIENT AUSSI AUX
ADULTES**



Introduction



Azure DevOps est le fruit de l'évolution de plusieurs générations d'outils de développement logiciel de Microsoft, offrant aujourd'hui une plateforme robuste qui intègre les pratiques DevOps avec l'écosystème cloud de Microsoft.

- **Visual Studio Team System (VSTS)** : Lancé en 2005, c'était la première offre de Microsoft pour le développement logiciel intégré, précurseur direct de TFS et Azure DevOps.
- **Team Foundation Server (TFS)** : Issu de VSTS, TFS a fourni une solution complète On Premise.
- **Visual Studio Online (VSO)** : Avec l'avènement du cloud, TFS a évolué en VSO en 2013, proposant les mêmes fonctionnalités dans une version accessible en ligne, facilitant l'accès et la flexibilité.
- **Azure DevOps** : En 2018, VSO a été renommé Azure DevOps, intégrant de manière plus profonde avec les services Azure et renforçant l'engagement de Microsoft envers les pratiques DevOps. Cette version marque une offre plus cohérente et intégrée pour le développement logiciel moderne.
- **Azure DevOps Server** : Parallèlement, la dernière version de TFS a été renommée Azure DevOps Server, offrant une option sur site pour ceux qui le préfèrent, tout en restant en synchronisation avec les services cloud d'Azure DevOps.

Azure DevOps se décline en plusieurs services principaux :

1. **Azure Boards** : Suivi des travaux et gestion de projets.
2. **Azure Repos** : Gestion de versions avec Git.
3. **Azure Pipelines** : CI/CD pour tous les langages et plateformes.
4. **Azure Test Plans** : Tests manuels et automatisés.
5. **Azure Artifacts** : Partage de packages Maven, npm, et NuGet.

Obtenir Azure DevOps



Pour démarrer votre voyage avec Azure DevOps, la première étape consiste à créer un compte et à configurer votre environnement de travail.



dev.azure.com

Visitez le site d'Azure DevOps et cliquez sur "Start free".

Connectez-vous ou créez un compte Microsoft : Si vous possédez déjà un compte Microsoft (Outlook, Office 365, etc.), connectez-vous avec celui-ci. Sinon, créez un nouveau compte.

Créez votre organisation Azure DevOps : Une fois connecté, vous serez invité à créer une nouvelle organisation. Donnez-lui un nom unique qui représente votre équipe, projet ou même votre entreprise.

Configurez votre premier projet : Après avoir créé votre organisation, créez votre premier projet en lui attribuant un nom et en choisissant sa visibilité (privée ou publique).

Organisation & Projets



Dans Azure DevOps, une Organisation constitue le niveau principal de structuration pour gérer vos projets, utilisateurs, et ressources de développement logiciel dans un environnement partagé. Elle sert de conteneur pour vos projets Azure DevOps et offre un cadre pour la collaboration d'équipe, la gestion de la sécurité, et l'intégration avec d'autres services Azure.

The screenshot shows the 'ODYCD' organization dashboard. At the top, there are links for 'Projects', 'My work items', and 'My pull requests'. A blue button on the right says '+ New project'. Below these, there's a search bar with the placeholder 'Filter projects'. Three project cards are displayed: 'Mire Hub' (green icon), 'Sides' (dark green icon with the text 'Other projets. NO MONEY HERE.'), and 'ForBabies' (purple icon).

Vous pouvez avoir plusieurs projets dans une organisation, chacun configuré avec ses propres paramètres, permissions, et membres d'équipe.

The screenshot shows the 'Turtle' project page. At the top, there are buttons for 'Private', 'Invite', and a star icon. On the left, the 'About this project' section includes a description 'Project for legal tech', a 'Languages' section with 'JavaScript' and 'CSS' selected, and a 'Wiki / Avocachet' section with the note 'I've no idea what i'm doing here.'. On the right, the 'Project stats' section shows 'Period: Last 30 days' with data for 'Repos' (0 pull requests opened, 54 commits by 1 author) and 'Pipelines' (93% builds succeeded).

Wiki



Le Wiki dans Azure DevOps est un outil collaboratif intégré qui permet aux équipes de créer, partager et gérer de la documentation de projet directement au sein de leur espace de travail Azure DevOps. Il est conçu pour capturer et organiser les connaissances du projet, faciliter la communication entre les membres de l'équipe, et servir de référentiel central pour les informations importantes du projet.

A screenshot of the Azure DevOps Wiki interface. On the left, there's a sidebar with a search bar and a list of recent pages: 'Enter page title', 'Avocachet', and 'Stamp'. The main area shows a page titled 'Avocachet' with a timestamp '4m ago' and the content 'I've no idea what i'm doing here.' Below it, it says '0 visits in last 30 days' and has a placeholder 'Add a comment...'. There are also 'Unfollow' and 'Edit' buttons.

Azure DevOps possède l'avantage d'avoir un wiki global à un projet mais également de rassembler toute la documentation contenue dans vos projets directement dans la partie Wiki.

Project wiki

□ Turtle.wiki

Code wikis

✓ □ Stamping

↑ Publish code as wiki

Azure Boards



The screenshot shows the left sidebar of the Azure Boards interface. The 'Boards' option is selected, indicated by a blue background. Other visible options include 'Work items', 'Boards', 'Backlogs', 'Sprints', 'Queries', 'Delivery Plans', and 'Analytics views'.

Azure Boards est un service d'Azure DevOps qui offre des outils complets pour la planification agile, le suivi des travaux et la gestion de projet. Il permet aux équipes de suivre le progrès de leurs projets à travers des tableaux Kanban, des backlogs, des sprints, et des rapports.

Hiérarchie : Azure Boards permet d'organiser les work items en une hiérarchie, aidant à visualiser les relations entre Epics, Features, User Stories, et Tasks. Cette structure facilite la planification et le suivi à différents niveaux d'abstraction.

Statuts : Chaque work item passe par différents statuts qui représentent son avancement, tels que "New", "Active", "Resolved", et "Closed". Vous pouvez personnaliser le workflow des statuts pour l'adapter à votre processus de développement.

Tags et Filtres : Utilisez des tags pour catégoriser les work items et filtrez-les pour faciliter la recherche et l'organisation. Les filtres peuvent être basés sur des critères tels que le type de work item, l'état, l'assigné, et plus encore.

Process Boards



Azure DevOps supporte différents types de processus de travail pour s'adapter aux méthodologies spécifiques d'une équipe ou d'un projet. Ces processus définissent le cadre pour les work items, les boards, et les workflows dans Azure Boards, offrant un ensemble de modèles adaptés à diverses approches de gestion de projet. Les principaux types de processus disponibles dans Azure DevOps sont : Agile, Scrum, CMMI, et Basic.

Agile

Conçu pour les équipes qui suivent la méthodologie agile, mettant l'accent sur des cycles de développement itératifs et incrémentiels, et une capacité de réponse élevée aux changements.

Work Items Principaux : Epics, Features, User Stories, Tasks, Bugs.

Scrum

Adapté aux équipes qui utilisent le cadre de travail Scrum, une sous-catégorie de la méthodologie agile axée sur des sprints fixes, des revues de sprint, et des stand-ups quotidiens.

Work Items Principaux : Epics, Features, Backlog Items, Tasks, Bugs.

CMMI

Orienté vers les équipes qui appliquent des approches de gestion de projet plus formelles et structurées, avec un accent sur l'amélioration continue des processus.

Work Items Principaux : Requirements, Change Requests, Defects, Tasks.

Basic

Un cadre simplifié, avec un ensemble minimal de types de work items. C'est une bonne option pour les petites équipes ou les projets qui n'ont pas besoin des formalités des cadres Agile, Scrum, ou CMMI.

Work Items Principaux : Epics, Tasks, Bugs.

Work Items



+ New Work Item ▾

- 🐞 Bug
- 👑 Epic
- 🏆 Feature
- ⚠️ Issue
- 📝 Task
- ⚖️ Test Case
- 📘 User Story

Azure Boards propose différents types de work items pour s'adapter à divers besoins et méthodologies de projet. Les types de base comprennent :

- Bug : Utilisé pour suivre les problèmes, erreurs ou bugs dans le logiciel. Les bugs peuvent être liés à des User Stories spécifiques pour un suivi facile.

- Epic : Représente un grand bloc de travail qui peut être décomposé en plusieurs features. Les epics sont souvent utilisés pour suivre le travail sur plusieurs sprints ou même des versions du produit.
- Feature : Utilisée pour définir une fonctionnalité ou un ensemble de fonctionnalités sous une Epic. Une feature est généralement livrée sur plusieurs sprints.
- Task : Représente une unité de travail spécifique nécessaire pour compléter une User Story ou une Feature. Les tâches sont souvent assignées à des individus pour exécution.
- User Story (ou simplement "Story") : Décrit une fonctionnalité du point de vue de l'utilisateur final. Les stories sont des éléments de travail plus petits que les features et visent à fournir de la valeur à l'utilisateur.

Work Items



BUG 1

1 Bug - Doesn't work on my machine

Antony Kervazo-Canut 1 Comment Add Tag

Save Follow Details

Updated by Antony Kervazo-Canut: Just now

| | |
|-------------|---------------------|
| State: New | Area: Mire Hub |
| Reason: New | Iteration: Mire Hub |

Repro Steps
Click to add Repro Steps.

System Info
Click to add System Info.

Discussion
Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

Antony Kervazo-Canut commented 7m ago (edited)
RTFM !

Planning
Resolved Reason: Story Points
Priority: 2
Severity: 3 - Medium
Activity: Development

Deployment
To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development
Add link
Link an Azure Repos commit, pull request or branch to see the status of your development. You can also [create a branch](#) to get started.

Effort (Hours)
Original Estimate
Remaining
Completed

Related Work
Add link Add an existing work item as a parent

System Info
Found in Build
Integrated in Build

- **Commentaires et Pièces Jointes :** Les membres de l'équipe peuvent laisser des commentaires sur les work items et y attacher des fichiers ou des images pour fournir plus de contexte ou partager des ressources.
- **Liens et Intégrations :** Azure Boards permet de lier des work items entre eux (par exemple, lier une Task à une User Story) ou à d'autres éléments comme des commits de code, des pull requests, et des builds dans Azure Repos et Azure Pipelines. Cela crée une trace d'audit complète et intégrée de l'activité du projet.

Kanban



Le Kanban est une méthode de gestion de projet et de flux de travail qui vise à améliorer l'efficacité et la flexibilité des processus de travail. Originaire du Japon et initialement développée par Toyota dans les années 1940 pour optimiser ses processus de fabrication, la méthode Kanban a depuis été adoptée dans de nombreux autres domaines, notamment le développement logiciel et la gestion de projet IT, grâce à sa simplicité et son efficacité. Azure Boards intègre le système Kanban dans ses fonctionnalités, permettant aux équipes de développement logiciel de gérer leurs projets de manière agile et visuelle.

| New | Active | Resolved | Closed |
|------------|---|----------|--------|
| + New item | <p>2 Create THE feature</p> <p>Active</p> <p>Antony Kervazo-Canut</p> <p>Activated By Antony Kervazo-Canut</p> | | |

L'implémentation du Kanban dans Azure Boards permet aux équipes de développement de maximiser leur productivité, d'optimiser leurs processus et d'atteindre une livraison de valeur plus rapide et plus fiable pour leurs clients.

Dans cette idée, il est possible de relier un commit dans un repository à un ticket.

Sprints



Les sprints dans Azure Boards font partie intégrante de la gestion agile de projet, permettant aux équipes de planifier, exécuter et suivre le travail dans des intervalles de temps définis, appelés sprints. Cette fonctionnalité est essentielle pour les équipes qui adoptent des méthodologies agiles telles que Scrum, car elle aide à organiser le travail en lots gérables, à maintenir l'équipe concentrée sur les objectifs à court terme, et à fournir des améliorations continues et incrémentielles du produit.

[New Sprint](#)

Team [?](#)
Mire Hub Team

Name
Sprint 1

Start [3/11/2024](#) End [3/22/2024](#)

Location
Mire Hub

Selecting existing iteration

[Create](#) [Cancel](#)

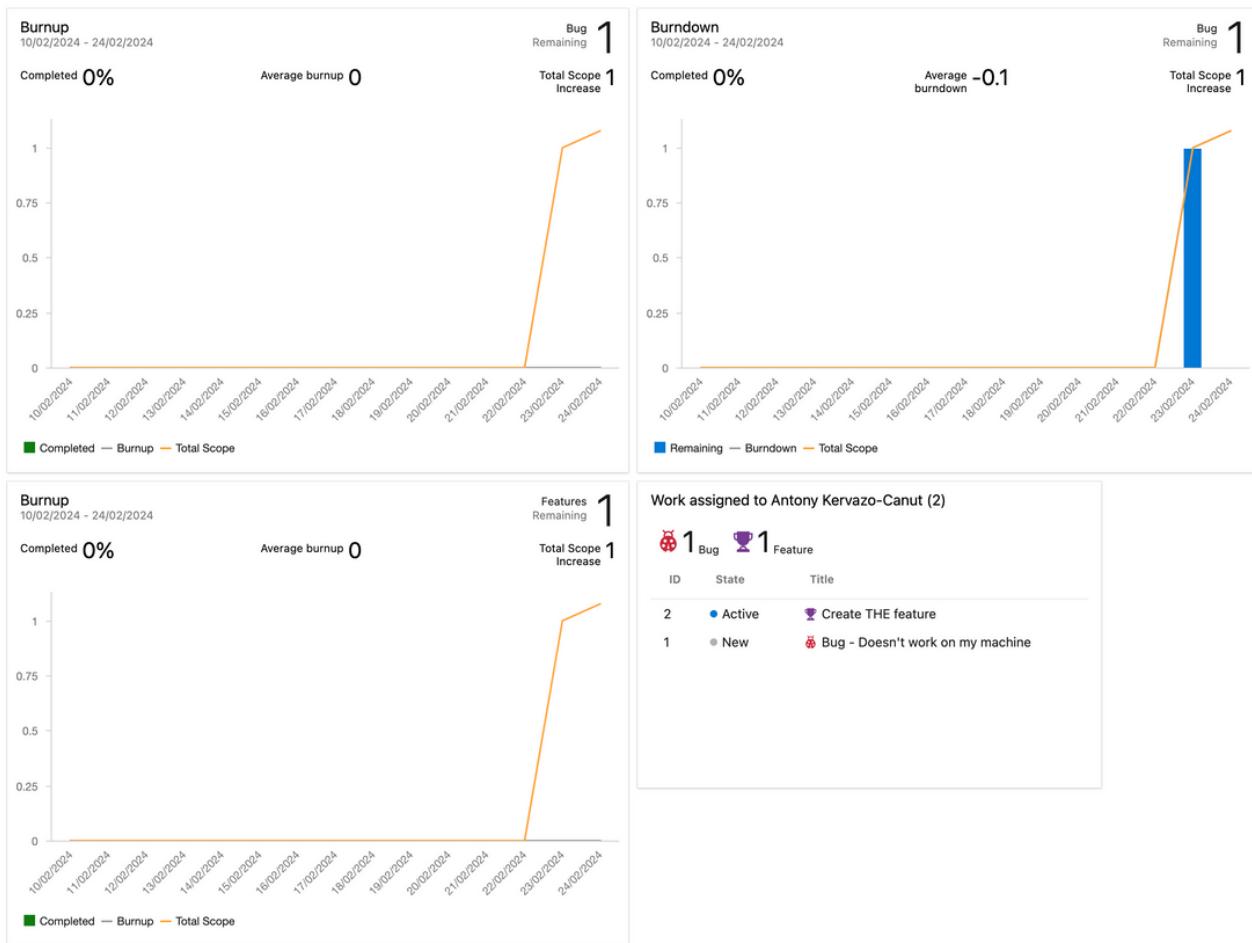
Définir vos itérations : Dans les paramètres de votre projet Azure DevOps, définissez la structure de vos itérations. Vous pouvez créer une hiérarchie d'itérations (par exemple, en les divisant par année, trimestre, etc.) et spécifier la durée de chaque sprint.

Planifier le travail : Affectez des work items à des sprints spécifiques en définissant l'itération pour chaque work item. Vous pouvez le faire en glissant-déposant des éléments dans le backlog d'itération ou en modifiant l'itération dans les détails du work item.

Dashboard & Widgets



Les tableaux de bord dans Azure Boards offrent une vue personnalisable et interactive de divers aspects de votre projet, grâce à l'utilisation de widgets. Ces widgets peuvent afficher un large éventail d'informations allant de l'état d'avancement des travaux, le suivi des bugs, les résultats des derniers builds ou déploiements, jusqu'à des métriques personnalisées importantes pour l'équipe. Voici un aperçu plus détaillé de certains widgets clés disponibles dans Azure Boards et comment ils peuvent être utilisés pour améliorer la visibilité et le suivi de vos projets.



Delivery Plan



Le Delivery Plan est une extension puissante pour Azure Boards qui fournit une vue de calendrier pour visualiser le travail à travers plusieurs équipes et projets. Il aide les organisations à comprendre comment le travail s'aligne sur les calendriers de livraison, facilitant la planification et le suivi des dépendances entre les équipes. Cette vue permet aux managers et aux équipes de rester synchronisés sur les objectifs et les délais de livraison, tout en identifiant les risques potentiels et les goulets d'étranglement à un stade précoce.

The screenshot shows the 'Release 1.0' interface of the Delivery Plan extension. At the top, there are navigation icons and a 'Settings' button. Below that is a header bar with 'Teams' and 'Features' dropdowns, and date navigation buttons for 'fevrier' (February) and 'mars' (March). The main area displays a calendar grid where each cell represents a day. In the 'Iteration 1' column for February, there is a box labeled 'Iteration 1' with the dates '2/24 - 3/9'. A purple button at the bottom right of this box says 'Create THE feature'. On the far left, under 'Teams', the 'Mire Hub Team' is selected, showing its 'Features' backlog.

Le Delivery Plan est un outil essentiel pour les gestionnaires de projet, les chefs d'équipe, et les parties prenantes qui cherchent à avoir une compréhension globale de la livraison des projets dans un environnement Agile. Il fournit les insights nécessaires pour prendre des décisions éclairées, gérer efficacement les ressources, et atteindre les objectifs de livraison dans les délais prévus.

Azure Repos



The screenshot shows the Azure DevOps interface for managing repositories. The left sidebar is titled 'Repos' and contains the following items:

- Files (selected)
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security

Azure Repos est un ensemble de services d'hébergement de contrôle de version qui vous permet de gérer votre code source dans le cloud. Il fait partie de la suite Azure DevOps de Microsoft et offre deux types principaux de contrôle de version : Git, un système de contrôle de version distribué, et Team Foundation Version Control (TFVC), un système de contrôle de version centralisé. Voici un aperçu simplifié de ce que vous devez savoir sur Azure Repos et comment l'utiliser pour gérer votre code.

The screenshot shows the 'Create a repository' dialog box. The 'Repository type' dropdown is set to 'Git'. Other options shown are 'Git' (selected) and 'TFVC'.

Git Repositories : Azure Repos fournit un hébergement Git illimité, supportant des workflows de développement collaboratif. Cela inclut des fonctionnalités telles que les revues de code, les branches de fonctionnalités, et les pull requests pour faciliter la collaboration et la gestion du code.

TFVC (Team Foundation Version Control) : Pour ceux qui préfèrent un modèle de contrôle de version centralisé, TFVC permet de suivre les changements dans le code, de maintenir des historiques de versions et de travailler sur de grands projets avec de nombreuses dépendances.

Sécurité et Permissions



Azure Repos utilise un système de permissions basé sur des rôles qui vous permet de contrôler finement l'accès à vos dépôts. Les permissions peuvent être attribuées à différents niveaux, y compris :

- Niveau Projet** : Vous pouvez définir des permissions qui s'appliquent à tous les repos dans un projet Azure DevOps. Cela est utile pour configurer des rôles à l'échelle du projet, comme les administrateurs de projet qui ont besoin d'accéder à tous les repos.
- Niveau Dépôt** : Les permissions peuvent également être définies spécifiquement pour chaque dépôt, permettant de personnaliser l'accès en fonction des besoins de chaque équipe ou projet. Cela inclut le contrôle de qui peut lire, écrire, ou gérer le dépôt.

Settings Policies Security Approvals and checks

User permissions Download detailed report

Inheritance

Search for users or groups

Azure DevOps Groups

- Build Administrators
- Contributors
- Project Administrators
- Readers
- Project Collection Administrators
- Project Collection Build Service Accounts
- Project Collection Service Accounts

Users

- Antony Kervazo-Canut
- Mire Hub Build Service (ODYCD)

[Mire Hub] | Contributors

| | |
|--|-------------------|
| Bypass policies when completing pull requests | Not set |
| Bypass policies when pushing | Not set |
| Contribute | Allow (inherited) |
| Contribute to pull requests | Allow (inherited) |
| Create branch | Allow (inherited) |
| Create tag | Allow (inherited) |
| Delete or disable repository | Not set |
| Edit policies | Not set |
| Force push (rewrite history, delete branches and tags) | Not set |
| Manage notes | Allow (inherited) |
| Manage permissions | Not set |
| Read | Allow (inherited) |
| Remove others' locks | Not set |
| Rename repository | Not set |

Sécurité et Permissions



Niveau Branches : Azure Repos permet de définir des politiques de branche qui peuvent restreindre les modifications à certaines branches, exiger des revues de code pour les pull requests, ou imposer des builds de validation. Cela aide à protéger les branches importantes, comme la branche principale ou de release.

All Branches

Settings Policies Security Approvals and checks

Branch Policies

Note: If any required policy is enabled, this branch cannot be deleted and changes must be made via pull request.

| | |
|---------------------------|---|
| <input type="radio"/> Off | Require a minimum number of reviewers Require approval from a specified number of reviewers on pull requests. |
| <input type="radio"/> Off | Check for linked work items Encourage traceability by checking for linked work items on pull requests. |
| <input type="radio"/> Off | Check for comment resolution Check to see that all comments have been resolved on pull requests. |
| <input type="radio"/> Off | Limit merge types Control branch history by limiting the available types of merge when pull requests are completed. |

✓ **Build Validation** 0

Validate code by pre-merging and building pull request changes.

No build policies found, but you can use the add button to create one!

Filter +

✓ **Status Checks** 0

Require other services to post successful status to complete pull requests.

No status checks found, but you can use the add button to create one!

Filter +

✓ **Automatically included reviewers** 0

Designate code reviewers to automatically include when pull requests change certain areas of code.

No automatic reviewer policies found, but you can use the add button to create one!

Filter +

Pull Request



Les pull requests dans Azure Repos sont un mécanisme central pour faciliter la collaboration et la révision du code au sein des équipes de développement utilisant Git. Elles permettent aux développeurs de notifier les membres de l'équipe qu'ils ont complété des changements dans une branche et de demander que ces changements soient intégrés (ou "mergés") dans la branche principale ou toute autre branche cible. Voici un aperçu détaillé des pull requests et de leur utilisation dans Azure Repos.

New pull request

client/mathilde ▾ into main ↗

Overview Files 96 Commits 56

Title
Merge feature clients into Main

Description
Set website as template

23/4000

Markdown supported. Drag & drop, paste, or select files to insert.

Add commit messages

Link work items.

Set website as template

Reviewers Add required reviewers

Search users and groups to add as reviewers

Work items to link 1 Clear all

Search work items by ID or title

Feature 2: Create THE feature Updated 1h ago, Active

Tags

Create

Pull Request



Révisions de Code : Les pull requests offrent un cadre pour les révisions de code, où les pairs peuvent commenter et suggérer des modifications avant que le code soit intégré. Cela contribue à maintenir la qualité du code et à partager les connaissances au sein de l'équipe.

Intégration Continue : Avec l'intégration de Azure Pipelines, vous pouvez configurer des builds automatiques pour les pull requests, s'assurant que le code passe tous les tests et vérifications avant la fusion.

Politiques de Branche : Vous pouvez appliquer des politiques de branche aux pull requests pour exiger des revues de code, des builds réussis, ou d'autres critères avant que le code puisse être mergé, renforçant les standards de qualité.

Automatisation : Les pull requests peuvent déclencher des workflows automatisés, comme la fermeture des tâches associées dans Azure Boards une fois que la pull request est complétée.

Merge feature clients into Main

Active 13 Antony Kervazo-Canut proposes to merge [client/mathilde](#) into [main](#)

Overview Files Updates Commits

No merge conflicts Last checked Just now

Description

Set website as template

Show everything (1) ▾

Reviewers Add ▾

Required No required reviewers

Optional No optional reviewers

Tags +

No tags

Work items +

Just now

Add a comment...

Antony Kervazo-Canut created the pull request

Feature 2: Create THE feature Updated Just now, Active

Pull Request



Astuce :

Si un endroit spécifique du code a été touché, vous pouvez inclure automatiquement un reviewer ou une équipe en particulier pour valider la Pull Request.

The screenshot shows a configuration page for "Automatically included reviewers". It includes a header with a dropdown for "Automatically included reviewers" (set to 1), a "Filter" button, and a "+" button. Below this is a table with columns: "Enabled", "Reviewers ↑", "Path filter", and "Inheritance". A row is selected with the status "On" (blue toggle), the reviewer "Critical space touched" (with a person icon and "Required" status), and the path filter "/Settings/*".

Cette technique peut être utilisée pour le déclenchement de pipelines. Ainsi, certains tests, certains builds peuvent être réalisés spécifiquement seulement si certaines parties des sources sont touchées.

Une plus grande séparation qui peut faire gagner du temps sur des builds, ou améliorer la sécurité et les processus pour n'inclure que les personnes réellement utiles à une revue.

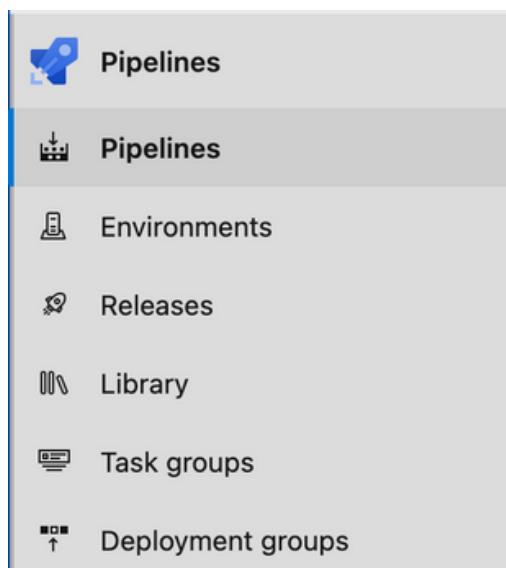
Azure Pipelines



Azure DevOps a connu plusieurs évolutions en matière de pipelines, reflétant les changements dans les pratiques de développement logiciel et l'intégration continue (CI) / livraison continue (CD). Voici un bref historique des différents types de pipelines utilisés au fil du temps dans Azure DevOps (anciennement connu sous le nom de Visual Studio Team Services (VSTS) et Team Foundation Server (TFS)).

- **Pipelines en XAML**, première génération, utilisant XAML pour la configuration. Complexes et difficiles à gérer.
- **Pipelines Classiques**, configuration via interface utilisateur sans scripts. Accessibles, faciles à utiliser.
- **Pipelines YAML**, configuration "as code" dans des fichiers YAML. Flexibles, versionnables, et favorisent la collaboration.

Azure DevOps supporte toujours les pipelines classiques et les pipelines en XAML (bien que ce dernier soit en fin de vie), mais encourage fortement l'utilisation des pipelines YAML pour la plupart des nouveaux projets.



Sources externes



La création d'un pipeline dans Azure Pipelines est un processus flexible qui permet non seulement d'utiliser du code hébergé sur Azure Repos, mais aussi de se connecter à des dépôts de code situés sur d'autres plateformes comme GitHub, GitLab, ou Bitbucket.

New pipeline

Where is your code?

-  Azure Repos Git YAML
Free private Git repositories, pull requests, and code search
-  Bitbucket Cloud YAML
Hosted by Atlassian
-  GitHub YAML
Home to the world's largest community of developers
-  GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise
-  Other Git
Any generic Git repository
-  Subversion
Centralized version control by Apache

Si vous choisissez une source de code externe, vous devrez autoriser Azure Pipelines à accéder à votre dépôt. Cela implique généralement de se connecter à votre compte sur la plateforme externe et d'accorder les permissions nécessaires pour que Azure Pipelines puisse lire le code et configurer des webhooks pour les déclencheurs de CI/CD.

Pipeline Classic



Les pipelines Classic dans Azure DevOps offrent une approche graphique et conviviale pour configurer et gérer l'intégration continue (CI) et la livraison continue (CD) sans écrire de code YAML.

⋯ > Mire Hub-Docker container-CI

The screenshot shows the Azure DevOps Pipeline Classic interface. On the left, there's a sidebar with 'Tasks', 'Variables', 'Triggers', 'Options', and 'History'. The main area is titled 'Pipeline' and shows a 'Build pipeline' configuration. It includes a 'Get sources' step (using 'BaseWebTemplate' and 'main' branch), an 'Agent job 1' step with two tasks: 'Build an image' (using 'Docker') and 'Push an image' (also using 'Docker'). The 'Build an image' task is currently selected. On the right, there's a detailed configuration panel for the 'Build an image' task, which includes fields for 'Task version' (set to '0.*'), 'Display name' ('Build an image'), 'Container Registry Type' ('Azure Container Registry'), 'Azure subscription' ('Azure Antony'), 'Azure Container Registry' ('mirehubregistry'), 'Action' ('Build an image'), 'Docker File' ('**/Dockerfile'), and 'Build Arguments' (empty). There are also 'Link settings', 'View YAML', and 'Remove' buttons at the top of this panel.

Les pipelines Classic possèdent de nombreux écrans pour la configuration des variables, triggers, et la gestion des sources. C'est un modèle historique, facile à comprendre et très flexible.

Astuce :

Azure DevOps propose toutefois la possibilité d'exporter un pipeline Classic existant vers un format YAML pour commencer à migrer vos anciens pipelines. Cependant, vous risquez d'éprouver des difficultés si vous utilisez des fonctionnalités telles que les « Task Groups ».

Pipeline YAML



Les pipelines YAML dans Azure DevOps représentent une approche moderne et code-centric pour définir les processus d'intégration continue (CI) et de livraison continue (CD). En utilisant le langage YAML (YAML Ain't Markup Language), ces pipelines permettent de décrire de manière déclarative comment construire, tester et déployer votre code directement dans un fichier texte qui peut être versionné avec votre code source.

La construction de pipelines au sein d'Azure DevOps permet toutefois d'avoir un assistant.

New pipeline

Review your pipeline YAML

Save and run

```
◆ BaseWebTemplate / azure-pipelines.yml * ⓘ
1 # Starter pipeline
2 # Start with a minimal pipeline that you can customize to build and deploy your code.
3 # Add steps that build, run tests, deploy, and more:
4 # https://aka.ms/yaml
5
6 trigger:
7 - main
8
9 pool:
10 -> vmiImage: ubuntu-latest
11
12 steps:
13   - task: Docker@2
14     inputs:
15       containerRegistry: 'Proget Mirehub'
16       command: 'buildAndPush'
17       Dockerfile: '**/Dockerfile'
18
19
```

← Docker ⓘ

Container Repository

Container registry ⓘ

Proget Mirehub

Container repository ⓘ

Commands

Command ⓘ

buildAndPush

Dockerfile ⓘ

**/Dockerfile

Les modifications apportées à la configuration du pipeline sont traçables via le système de contrôle de version, offrant une visibilité complète sur l'évolution du processus CI/CD.

Astuce :

En utilisant les politiques de branche, vous pouvez faire en sorte d'inclure systématiquement des personnes qui valideront la modification du pipeline.

Agent Pool



Les Agent Pools dans Azure DevOps sont des collections d'agents qui exécutent les tâches définies dans vos pipelines CI/CD. Un agent est une application installée sur une machine physique ou virtuelle qui exécute des builds, des déploiements, ou d'autres tâches automatisées définies dans vos pipelines. Les pools d'agents permettent de gérer et d'organiser ces agents en groupes, facilitant l'allocation des ressources nécessaires pour l'exécution des pipelines.

- **Agents Hébergés** : Azure DevOps fournit des agents hébergés, qui sont des machines virtuelles gérées par Microsoft avec des environnements de développement préinstallés. Ces agents sont prêts à l'emploi et ne nécessitent pas de maintenance. Il existe plusieurs types d'agents hébergés pour cibler différents systèmes d'exploitation ou configurations.
- **Agents Auto-Hébergés** : Vous pouvez également configurer vos propres agents sur vos machines. Cela est utile pour des scénarios où vous avez besoin de configurations spécifiques, d'accéder à des ressources internes, ou d'utiliser des logiciels qui ne sont pas disponibles sur les agents hébergés.

La configuration d'un agent auto-hébergé se fait dans la configuration de projets.

Releases



Les Releases dans Azure DevOps sont une partie cruciale du processus de livraison continue (CD), permettant de gérer le déploiement d'applications à travers différents environnements, comme le développement, le test, et la production. La gestion des releases dans Azure DevOps est conçue pour automatiser, planifier et suivre le déploiement de votre code, tout en assurant la qualité et la stabilité de vos applications.

Les pipelines de release sont principalement configurés via une interface utilisateur graphique, sans option directe pour écrire la configuration en YAML.

The screenshot illustrates the Azure DevOps Pipeline interface for managing releases. At the top, the pipeline for "My App" is shown with three stages: Development, Integration, and Production. Each stage contains one job with four tasks. A tooltip indicates that the schedule for artifact generation is not set. Below the main pipeline view, a detailed configuration pane is open for the "Development" stage. It shows tasks for archiving files and signing APKs. A large callout box highlights the "Pre-deployment conditions" section, which includes settings for triggers, approvals (with a user named "Antony Kervazo-Canut" listed), and gates. Other sections visible include "Archive files", "Task version", "Display name", "Root folder or file to archive", "Archive type", "Archive file to create", "Control Options", "Output Variables", and "Approval policies".

Pipelines Multi-Stages en YAML



Avec l'introduction et l'évolution des pipelines multi-stages en YAML, Azure DevOps a commencé à offrir une expérience unifiée pour la définition de l'intégration continue (CI) et de la livraison continue (CD) dans un seul fichier YAML. Cette approche étend la flexibilité et la puissance du YAML aux processus de déploiement, permettant aux utilisateurs de définir des stages de déploiement, des jobs, des étapes, des environnements, des stratégies de déploiement, et d'autres aspects de la livraison dans le même fichier YAML utilisé pour la CI.

```
● ○ ●

trigger:
- main

stages:
- stage: Build
  jobs:
    - job: BuildJob
      pool:
        vmImage: 'ubuntu-latest'
      steps:
        - script: echo Building ...
        - script: dotnet build --configuration Release

- stage: Deploy
  dependsOn: Test
  condition: and(succeeded(),
eq(variables['Build.SourceBranch'], 'refs/heads/main'))
  jobs:
    - deployment: DeployJob
      environment: production
      pool:
        vmImage: 'ubuntu-latest'
      strategy:
        runOnce:
          deploy:
            steps:
              - script: echo Deploying to production ...
```

Library



La Library dans Azure DevOps est une fonctionnalité qui permet de stocker, gérer et partager des variables et des secrets entre plusieurs pipelines, facilitant la réutilisation des configurations et la centralisation de la gestion des données sensibles.

Library

| Variable groups | Secure files | + Variable group | Security | Help |
|-------------------|--------------|------------------|----------|----------------------|
| Name | | Date modified | | Modified by |
| fx Azure-Registry | | 11/2/2023 | | Antony Kervazo-Canut |

Bonne Pratiques :

- **Principe du Moindre Privilège** : Limitez l'accès aux variables et aux groupes de variables selon le besoin. Ne donnez l'accès qu'aux membres de l'équipe qui nécessitent ces informations pour leur travail.
- **Révision Régulière** : Passez régulièrement en revue vos variables et groupes de variables pour vous assurer qu'ils sont à jour et que seules les personnes appropriées ont accès.
- **Utilisation de Groupes de Variables par Environnement** : Organisez vos variables en groupes spécifiques à chaque environnement pour simplifier la gestion et réduire les risques de déploiement accidentel avec la mauvaise configuration.

Variable Groups



Library > Registry

Variable group | Save | Clone | Security | Pipeline permissions | Approvals and checks

Properties

Variable group name

Registry

Description

 Link secrets from an Azure key vault as variables ⓘ

Variables

| Name ↑ | Value |
|---------|-------------|
| Account | AccountName |
| Token | ***** |

```
trigger:
- main

variables:
- group: Registry

jobs:
- job: ExampleJob
  pool:
    vmImage: 'ubuntu-latest'
  steps:
- script: echo Using account $(account)
  displayName: 'Display Account Variable'
```



Secure Files

La section **Secure Files** dans Azure DevOps est une fonctionnalité de la Library qui permet de stocker et de gérer des fichiers sécurisés, tels que des certificats, des fichiers de configuration contenant des secrets, ou tout autre type de fichier sensible, que vous pouvez ensuite utiliser dans vos pipelines CI/CD. Ces fichiers sont stockés de manière sécurisée et peuvent être référencés par les pipelines pour automatiser les déploiements et les builds tout en gardant les informations sensibles protégées.

Library

| Variable groups | Secure files | + Secure file | Security | Help |
|------------------------|--------------|---------------|----------------------|------|
| Name | | Date modified | Modified by | |
| kubernetes-config.yaml | | 9/28/2022 | Antony Kervazo-Canut | |

Le fichier ne pourra pas être lu depuis son interface mais possède une gestion de permissions et de conditions d'accès.

Library > kubernetes-config.yaml

Secure file

Secure file name

kubernetes-config.yaml

Properties
Optionally define properties for the secure file.

| Name | Value |
|------|-------|
|------|-------|

+ Add

Approvals and Checks



Les groupes de variables ou les secure files peuvent obtenir des conditions d'utilisations sur les pipelines. Celles-ci peuvent s'appliquer sur des branches en particulier, des personnes pouvant lancer des pipelines ayant comme utilisation des éléments secrets ou même l'horaire d'utilisation pour éviter par exemple un déploiement maladroit sur une plage horaire ou un jour de la semaine.

D'autres conditions peuvent même être écrite directement via des Azure Functions, ou faisant appel à des API REST pour obtenir la validation d'utilisation du secret.

Add check ×

Search check types

- Approvals**
Approvers should grant approval for deployment
- Branch control**
Allow deployments based on branches linked to the run
- Business Hours**
Ensure the deployment is started in a specific time window
- Evaluate artifact (preview)**
Ensure artifacts adhere to custom policies (container images only)
- Exclusive Lock**
Limit access to this resource to only a single stage at a time
- Invoke Azure Function**
Invoke an Azure Function
- Invoke REST API**
Invoke a REST API as a part of your pipeline.
- Query Azure Monitor alerts**
Observe the configured Azure Monitor rules for active alerts
- Required template**
Ensure the pipeline extends one or more YAML templates

Azure Test Plans



Azure Test Plans, une composante d'Azure DevOps, est un ensemble d'outils puissants conçus pour la planification des tests, l'exécution manuelle et l'automatisation des tests, ainsi que pour le suivi de la qualité globale du logiciel au sein de vos projets de développement. Azure Test Plans soutient une approche intégrée à la gestion de la qualité, facilitant la collaboration entre les développeurs, les testeurs et les gestionnaires de projet pour améliorer la fiabilité et la performance des applications.

First test (ID: 5) ?Help

Define Execute Chart □ ■ ↗ =

| Test Points (2 items) | | | | Run for web application |
|--|---------|-------|--------------|-------------------------|
| | Outcome | Order | Test Case Id | |
| <input type="checkbox"/> Title | | | | |
| <input type="checkbox"/> First test case | Failed | 1 | 6 | |
| <input type="checkbox"/> Other | Active | 2 | 7 | |

La version gratuite d'Azure DevOps inclut l'accès à Azure Boards, Azure Repos, Azure Pipelines, et une capacité limitée pour Azure Artifacts. Cependant, pour Azure Test Plans, l'accès est plus restreint.

Azure Artifacts



Azure Artifacts représente une fonctionnalité puissante pour gérer et partager des packages au sein de votre organisation ou avec la communauté. Un "feed" est un dépôt de packages où vous pouvez publier, consommer et partager des packages de différents types. Cela facilite la collaboration entre les équipes de développement et la réutilisation du code, tout en soutenant les pratiques de DevOps en permettant une gestion fluide des dépendances de projet.

Create new feed

Feeds host your packages and let you control permissions.

Name *

Visibility

- Members of ODYCD
Any member of your organization can view the packages in this feed
- Specific people
Only users you grant access to can view the packages in this feed

Upstream sources

- Include packages from common public sources

For example: nuget.org, npmjs.com

Scope

- Project: Mire Hub (Recommended)
The feed will be scoped to the Mire Hub project.
- Organization

| Type | Source | Location |
|--------------------------|---|---|
| <input type="checkbox"/> | Type | Source |
| <input type="checkbox"/> |  npmjs | https://registry.npmjs.org/ |
| <input type="checkbox"/> |  NuGet Gallery | https://api.nuget.org/v3/index.json |
| <input type="checkbox"/> |  PowerShell Gallery | https://www.powershellgallery.com/api/v2/ |
| <input type="checkbox"/> |  PyPI | https://pypi.org/ |
| <input type="checkbox"/> |  Maven Central | https://repo.maven.apache.org/maven2/ |
| <input type="checkbox"/> |  Google Maven Repository | https://dl.google.com/android/maven2/ |
| <input type="checkbox"/> |  JitPack | https://jitpack.io/ |
| <input type="checkbox"/> |  Gradle Plugins | https://plugins.gradle.org/m2/ |
| <input type="checkbox"/> |  crates.io | https://index.crates.io/ |

Visual Studio Marketplace



La Visual Studio Marketplace est une plateforme en ligne où les utilisateurs peuvent découvrir, acquérir et installer des extensions pour améliorer et personnaliser leur expérience avec les produits Visual Studio, Azure DevOps, et GitHub. Elle propose une large gamme d'extensions, allant des intégrations avec d'autres services et outils, aux améliorations de productivité pour les développeurs, en passant par des widgets de tableau de bord et des tâches personnalisées pour les pipelines Azure DevOps.

Extensions for Azure DevOps

🔍

Featured

Test & Feedback
Microsoft 131K
★★★★★ FREE

Retrospectives
Microsoft DevLabs [microsoft.com](#) 71K
★★★★★ FREE

Timetracker
7pace (an Appfire c... [7pace.com](#) 26.9K
★★★★★ FREE TRIAL

Portfolio++
iTrellis [itrellis.com](#) 15.6K
★★★★★ FREE

Jira to Azure DevOps/
Solidify AB [solidify.dev](#) 3.2K
★★★★★ FREE

Code Quality NDepend
ndepend 2.9K
★★★★★ FREE TRIAL

Most Popular

Azure DevOps Open i...
Microsoft DevLabs 245K
★★★★★ FREE

Code Search
Microsoft 206K
★★★★★ FREE

SARIF SAST Scans Tab
Microsoft DevLabs 186K
★★★★★ FREE

SonarQube
SonarSource [sonarsource.com](#) 123K
★★★★★ FREE

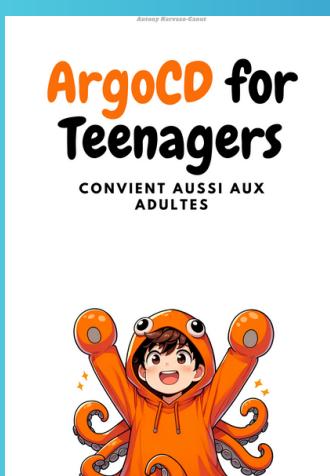
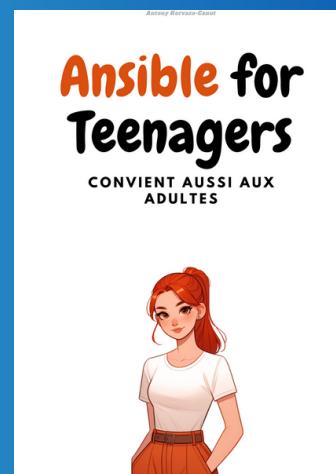
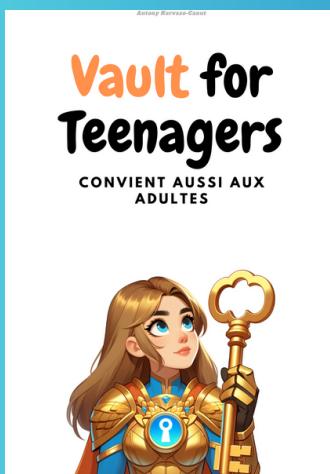
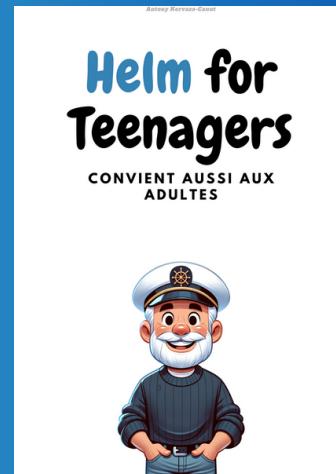
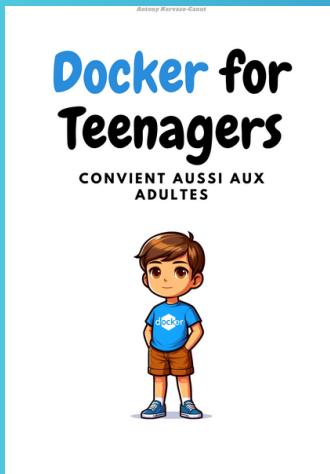
Replace Tokens
Guillaume Rouchon 113K
★★★★★ FREE

Terraform
Microsoft DevLabs [microsoft.com](#) 92.3K
★★★★★ FREE

[See more ↗](#)

Visual Studio Marketplace offre des extensions qui peuvent étendre et personnaliser les fonctionnalités d'Azure Boards, Azure Repos, Azure Pipelines, et plus encore.

Dans la même collection



ANTONYCANUT



ANTONY KERVAZO-CANUT

