

# Contents

Documentation Azure DevOps Projects

Vue d'ensemble

Présentation d'Azure DevOps Project

Démarrages rapides

.NET

Node.js

Java

Python

PHP

Ruby

Go

Tutoriels

1 - GitHub et DevOps Projects

2 - Déployer sur des machines virtuelles

3 - Déployer sur Azure SQL Database

4 - Déployer sur AKS

5 - Déployer sur Azure Service Fabric

6 - Déployer sur Azure Cosmos DB

7 - Déployer sur Azure Functions

# Vue d'ensemble d'Azure DevOps Projects

17/09/2019 • 5 minutes to read • [Edit Online](#)

Azure DevOps Projects facilite la prise en main d'Azure. Il vous permet de lancer votre application favorite sur le service Azure de votre choix en seulement quelques étapes rapides à partir du portail Azure.

DevOps Projects configure tout ce dont vous avez besoin pour développer, déployer et superviser votre application. Vous pouvez utiliser le tableau de bord DevOps Projects pour superviser les validations de code, les builds et les déploiements dans une même vue du portail Azure.

## Pourquoi utiliser DevOps Projects ?

DevOps Projects automatise la configuration d'un pipeline complet d'intégration continue (CI) et de livraison continue (CD) sur Azure. Vous pouvez démarrer avec du code existant ou utiliser un des exemples d'application fournis. Vous pouvez ensuite déployer rapidement cette application sur différents services Azure, comme Machines virtuelles, App Service, Azure Kubernetes Service (AKS), Azure SQL Database et Azure Service Fabric.

Azure DevOps Projects s'occupe de toute la configuration initiale d'un pipeline DevOps, notamment configurer le dépôt Git initial, configurer le pipeline CI/CD, créer une ressource Application Insights pour la supervision et fournir une même vue de l'ensemble de la solution avec la création d'un tableau de bord DevOps Projects dans le portail Azure.

Vous pouvez utiliser DevOps Projects pour :

- Déployer rapidement votre application dans Azure
- Automatiser la configuration d'un pipeline CI/CD
- Voir et comprendre comment configurer correctement un pipeline CI/CD
- Personnaliser davantage les pipelines de mise en production selon vos scénarios spécifiques

## Comment utiliser DevOps Projects ?

DevOps Project est disponible à partir du portail Azure. Vous créez une ressource DevOps Projects comme toute autre ressource Azure à partir du portail. DevOps Projects fournit une expérience de type Assistant pas à pas pour les différentes options de configuration.

Vous choisissez plusieurs options de configuration dans le cadre de la configuration initiale. Ces options sont les suivantes :

- Utiliser l'exemple d'application fourni ou utiliser votre propre code
- Sélectionner un langage pour l'application
- Choisir un framework d'application en fonction du langage
- Sélectionner un service Azure (cible de déploiement)
- Créer une organisation Azure DevOps ou utiliser une organisation existante
- Choisir votre abonnement Azure
- Choisir l'emplacement des services Azure
- Choisir parmi différents niveaux tarifaires pour les services Azure

Après avoir utilisé DevOps Projects, vous pouvez également supprimer toutes les ressources à partir d'un même emplacement dans le tableau de bord DevOps Projects sur le portail Azure.

# Intégration de DevOps Projects et d'Azure DevOps

DevOps Projects s'appuie sur Azure DevOps. DevOps Projects automatise tout le travail nécessaire dans Azure Pipelines pour configurer un pipeline CI/CD. Il crée un dépôt Git dans une organisation Azure DevOps nouvelle ou existante, puis valide un exemple d'application ou votre code existant dans un nouveau dépôt Git.

L'automatisation établit également un déclencheur CI pour la build de sorte que chaque nouvelle validation de code lance une build. DevOps Projects crée un déclencheur CD et déploie chaque nouvelle build réussie sur le service Azure de votre choix.

Les pipelines de build et de mise en production peuvent être personnalisés pour d'autres scénarios. Vous pouvez également cloner les pipelines de build et de mise en production pour les utiliser dans d'autres projets.

Après avoir créé votre projet DevOps Project, vous pouvez :

- Personnaliser votre build et pipeline de mise en production
- Utiliser des requêtes Pull pour gérer votre flux de code et conserver une qualité élevée
- Tester et générer chaque validation avant de fusionner votre code pour optimiser la qualité
- Effectuer le suivi de votre backlog et de vos problèmes en même temps que votre application

## Comment commencer à utiliser DevOps Projects ?

- [Bien démarrer avec DevOps Projects](#)

## Vidéos sur DevOps Projects

- [Create CI/CD with Azure DevOps Projects](#)

# Didacticiel : Créer un pipeline CI/CD pour votre code existant avec Azure DevOps Projects

17/09/2019 • 14 minutes to read • [Edit Online](#)

Azure DevOps Projects offre une expérience simplifiée dans laquelle vous pouvez apporter vos code et dépôt Git existants ou choisir un exemple d'application pour créer un pipeline d'intégration continue (CI) et de livraison continue (CD) dans Azure.

Vous allez effectuer les étapes suivantes :

- Utiliser DevOps Projects pour créer un pipeline CI/CD
- Configurer l'accès à votre dépôt GitHub et choisir un framework
- Configurer Azure DevOps et un abonnement Azure
- Valider les modifications apportées à GitHub et les déployer automatiquement dans Azure
- Examiner le pipeline CI/CD Azure Pipelines
- Configurer la surveillance d'Azure Application Insights
- Supprimer des ressources

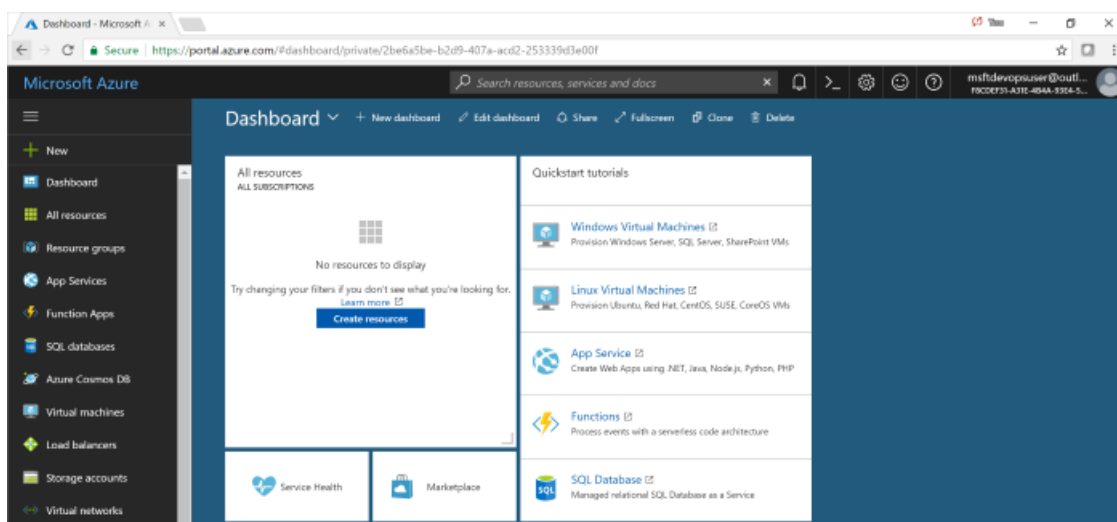
## Prérequis

- Un abonnement Azure. Vous pouvez en obtenir un gratuit via [Visual Studio Dev Essentials](#).
- Accédez à un dépôt GitHub ou Git externe qui contient .NET, Java, PHP, Node, Python ou du code web statique.

## Connectez-vous au portail Azure.

Azure DevOps Projects crée un pipeline CI/CD dans Azure Pipelines. Vous pouvez créer une organisation Azure DevOps ou utiliser une organisation existante. Azure DevOps Projects crée également des ressources Azure dans l'abonnement Azure de votre choix.

1. Connectez-vous au [Portail Azure](#).
2. Dans le volet gauche, sélectionnez **Nouveau...**
3. Dans la zone de recherche, tapez **DevOps Projects**, puis sélectionnez **Créer**.



4. Sélectionnez **Apportez votre propre code**, puis sélectionnez **Suivant**.

## Configurer l'accès à votre dépôt GitHub et choisir un framework

1. Sélectionnez **GitHub** ou un dépôt Git externe, puis sélectionnez votre dépôt et la branche qui contient votre application.
2. Sélectionnez votre framework web, puis sélectionnez **Suivant**.

The screenshot shows a four-step progress bar at the top: 1. Code Repository (green checkmark), 2. Application/Framework (blue circle with '2'), 3. Service (grey circle with '3'), and 4. Create (grey circle with '4'). Below the progress bar, the text 'Tell us more about the code' is displayed. Underneath, there is a label '\* Web Application framework' followed by an information icon. A dropdown menu is open, showing a list of frameworks: ASP.NET (selected), ASP.NET Core, Node.js, PHP, Python, and Static Webapp. The dropdown menu has a blue border and a downward arrow on the right side.

Le framework d'application, que vous avez choisi précédemment, détermine le type de cible de déploiement de service Azure disponible ici.

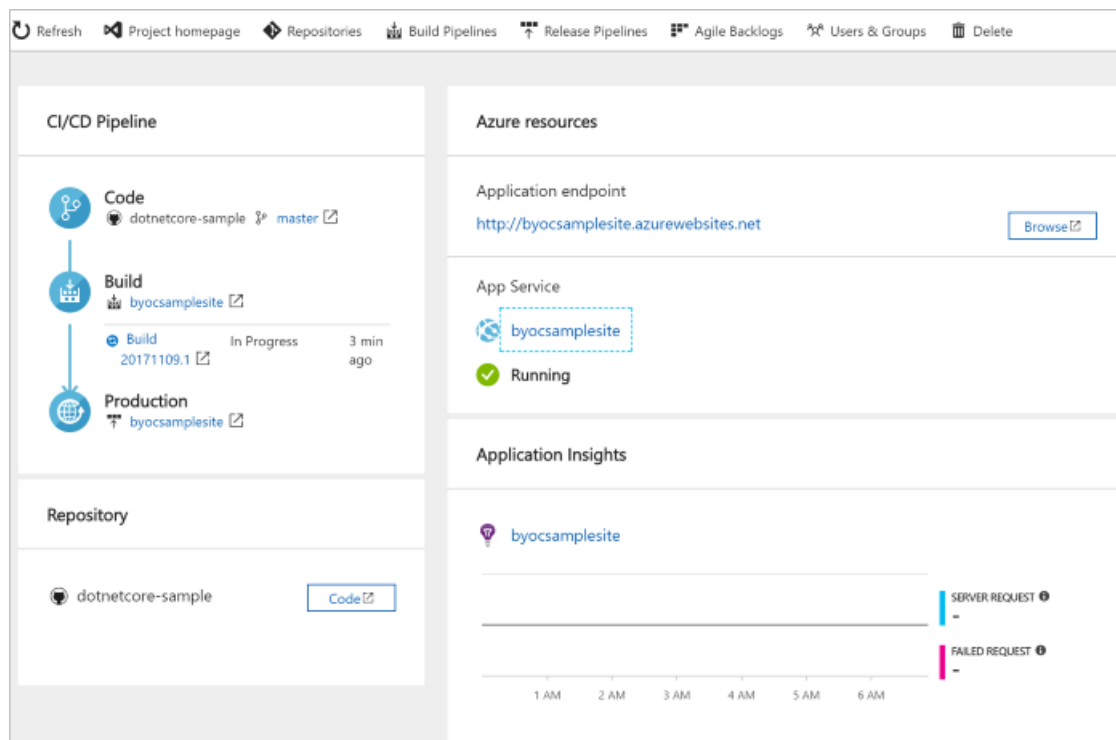
3. Sélectionnez le service cible, puis sélectionnez **Suivant**.

## Configurer Azure DevOps et un abonnement Azure

1. Créez une organisation Azure DevOps ou sélectionnez une organisation existante.
  - a. Indiquez un nom pour votre projet Azure DevOps.
  - b. Sélectionnez votre abonnement et un emplacement Azure, entrez un nom pour votre application, puis sélectionnez **Terminé**.

Après quelques minutes, le tableau de bord DevOps Projects s'affiche dans le portail Azure. Un exemple d'application est configuré dans un dépôt de votre organisation Azure DevOps, une build est exécutée et votre application est déployée sur Azure. Ce tableau de bord donne une visibilité sur votre dépôt de code GitHub, le pipeline CI/CD et votre application dans Azure.

2. Sélectionnez **Parcourir** pour voir votre application en cours d'exécution.



Azure DevOps Projects configure automatiquement un déclencheur de build et de mise en production CI. Votre code reste dans votre dépôt GitHub ou un autre dépôt Git externe.

## Valider les modifications apportées à GitHub et les déployer automatiquement dans Azure

Vous êtes maintenant prêt à collaborer avec une équipe sur votre application, par le biais d'un processus d'intégration et de déploiement continu (CI/CD) qui déploie automatiquement votre dernier travail sur votre site web. Chaque modification apportée au dépôt GitHub démarre une build dans Azure DevOps tandis qu'un pipeline CD exécute un déploiement dans Azure.

1. Effectuez une modification sur votre application, puis validez-la dans votre dépôt GitHub.  
Après quelques instants, une build démarre dans Azure Pipelines. Vous pouvez superviser l'état de la build dans le tableau de bord DevOps Projects, ou dans le navigateur avec votre organisation Azure DevOps.
2. Une fois la build terminée, actualisez votre application pour vérifier vos modifications.

## Examiner le pipeline CI/CD Azure Pipelines

Azure DevOps Projects configure automatiquement un pipeline CI/CD dans Azure Pipelines. Explorez et personnalisez le pipeline selon vos besoins. Pour vous familiariser avec les pipelines de build et de mise en production, effectuez les étapes suivantes :

1. En haut du tableau de bord DevOps Projects, sélectionnez **Pipelines de build**.  
Un onglet du navigateur affiche le pipeline de build de votre nouveau projet.
2. Pointez sur le champ **État**, puis sélectionnez les points de suspension (...).  
Un menu affiche plusieurs options, telles que de la mise en file d'attente d'une nouvelle build, la mise en pause d'une build et la modification du pipeline de build.
3. Sélectionnez **Modifier**.
4. Dans ce volet, vous pouvez examiner les différentes tâches de votre pipeline de build.  
La build effectue diverses opérations, comme la récupération (fetch) de sources dans le dépôt Git, la restauration de dépendances et la publication de sorties utilisées pour les déploiements.

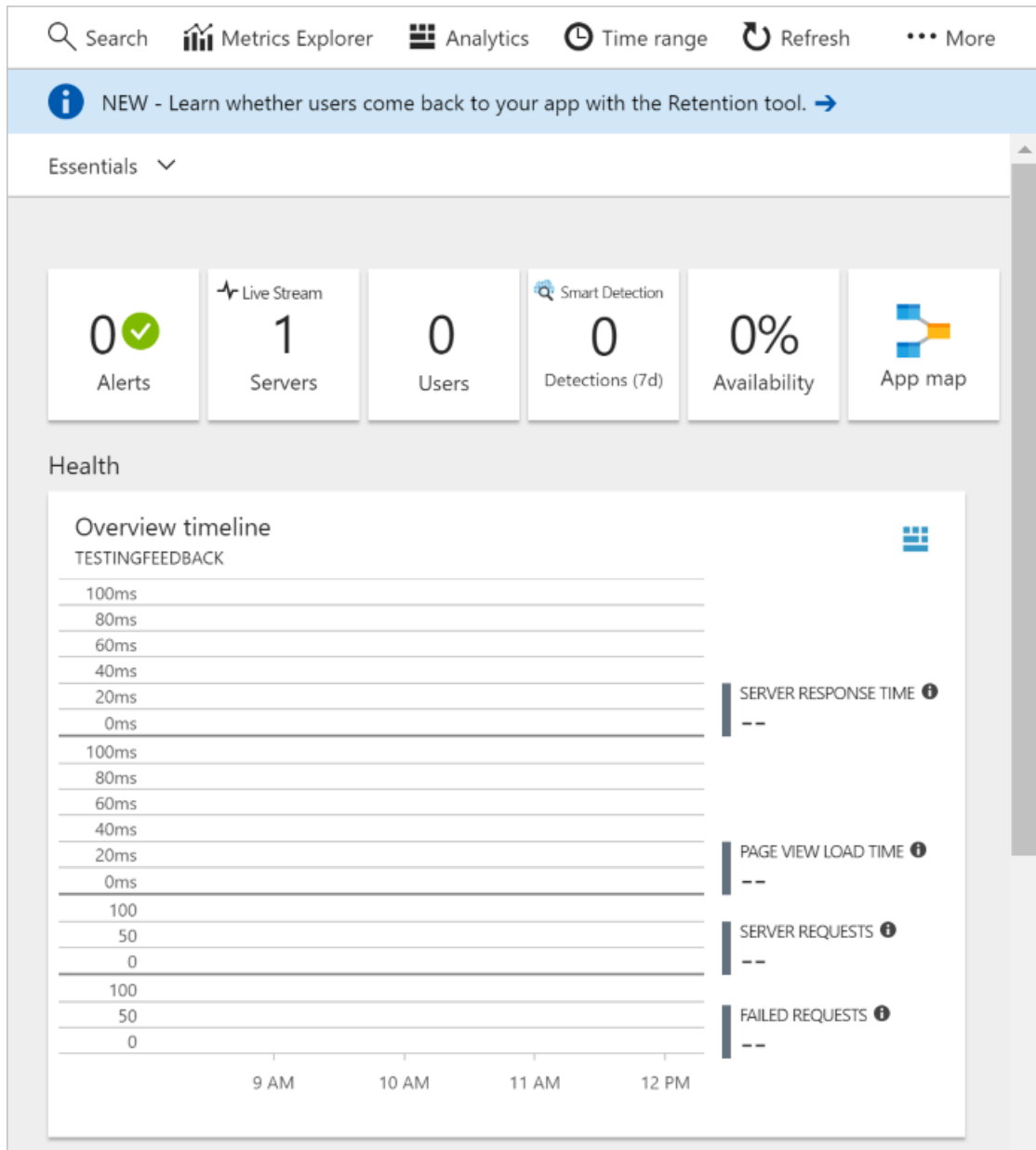
5. En haut du pipeline de build, sélectionnez le nom du pipeline de build.
6. Remplacez le nom de votre pipeline de build par un nom plus descriptif, sélectionnez **Enregistrer et mettre en file d'attente**, puis sélectionnez **Enregistrer**.
7. Sous le nom de votre pipeline de build, sélectionnez **Historique**.  
Vous pouvez observer une piste d'audit des modifications que vous avez apportées récemment à la build. Azure DevOps suit les modifications apportées au pipeline de build et permet de comparer les versions.
8. Sélectionnez **Déclencheurs**.  
Azure DevOps Projects crée automatiquement un déclencheur CI tandis que chaque validation du dépôt lance une nouvelle build. Si vous le souhaitez, vous pouvez choisir d'inclure ou d'exclure des branches dans le processus CI.
9. Sélectionnez **Rétention**.  
En fonction de votre scénario, vous pouvez spécifier des stratégies pour conserver ou supprimer un certain nombre de builds.
10. Sélectionnez **Build et mise en production**, puis **Mises en production**.  
Azure DevOps Projects crée un pipeline de mise en production pour gérer les déploiements sur Azure.
11. Sélectionnez les points de suspension (...) en regard de votre pipeline de mise en production, puis sélectionnez **Modifier**.  
Le pipeline de mise en production contient un *pipeline* qui définit le processus de mise en production.
12. Sous **Artefacts**, sélectionnez **Déposer**.  
Le pipeline de build que vous avez examiné aux étapes précédentes produit la sortie qui est utilisée pour l'artefact.
13. En regard de l'icône **Déposer**, sélectionnez **Déclencheur de déploiement continu**.  
Ce pipeline de mise en production possède un déclencheur CD actif, qui exécute un déploiement à chaque fois qu'un nouvel artefact de build est disponible. Si vous le souhaitez, vous pouvez désactiver le déclencheur afin que vos déploiements nécessitent une exécution manuelle.
14. À gauche, sélectionnez **Tâches**.  
Les tâches correspondent aux activités exécutées par votre processus de déploiement. Dans cet exemple, une tâche a été créée pour déployer sur le service Azure App.
15. À droite, sélectionnez **Afficher les mises en production** pour voir l'historique des mises en production.
16. Sélectionnez les points de suspension (...) en regard d'une mise en production, puis sélectionnez **Ouvrir**.  
Il y a plusieurs menus à explorer, comme un résumé des mises en production, les éléments de travail associés et les tests.
17. Sélectionnez **Validations**.  
Cette vue montre les validations de code qui sont associées à ce déploiement.
18. Sélectionnez **Journaux d'activité**.  
Les journaux d'activité contiennent des informations utiles sur le processus de déploiement. Vous pouvez les voir à la fois pendant et après les déploiements.

## Configurer la surveillance d'Azure Application Insights

Azure Application Insights vous permet d'analyser facilement les performances et l'utilisation de votre application. Azure DevOps Projects configure automatiquement une ressource Application Insights pour votre application. Vous pouvez configurer différentes capacités d'alertes et de surveillance, si besoin.

1. Dans le portail Azure, accédez au tableau de bord DevOps Projects.

2. En bas à droite, sélectionnez le lien **Application Insights** pour votre application.
- Le volet **Application Insights** s'ouvre. Cette vue contient les informations sur la surveillance de l'utilisation, des performances et de la disponibilité de votre application.



3. Sélectionnez **Intervalle de temps**, puis **Dernière heure**. Pour filtrer les résultats, sélectionnez **Mettre à jour**.
- Vous pouvez voir à présent toutes les activités des 60 dernières minutes. Pour quitter l'intervalle de temps, sélectionnez **x**.
4. Sélectionnez **Alertes**, puis **Ajouter une alerte Métrique**.
5. Entrez un nom pour l'alerte.
6. Dans la liste déroulante **Alerte pour**, sélectionnez votre **ressource App Service**.
7. Dans la liste déroulante **Métrique**, examinez les différents métriques d'alerte.
- L'alerte par défaut est pour un **temps de réponse serveur supérieur à 1 seconde**. Vous pouvez aisément configurer une variété d'alertes pour améliorer les capacités de surveillance de votre application.
8. Cochez la case **Envoyer un e-mail aux propriétaires, aux contributeurs et aux lecteurs**.
- Si vous le souhaitez, lorsqu'une alerte est affichée, vous pouvez effectuer des actions supplémentaires en exécutant une application logique Azure.



9. Sélectionnez **OK** pour créer l'alerte.  
Au bout de quelques instants, l'alerte apparaît comme étant active sur le tableau de bord.
10. Quittez la zone **Alertes** et revenez au volet **Application Insights**.
11. Sélectionnez **Disponibilité** puis **Ajouter un test**.
12. Entrez un nom de test, puis sélectionnez **Créer**.  
Un test Ping basique est créé pour vérifier la disponibilité de votre application. Au bout de quelques instants, les résultats de test sont disponibles, et le tableau de bord Application Insights affiche le statut de disponibilité.

## Supprimer des ressources

Quand ils ne sont plus nécessaires, vous pouvez supprimer le service Azure App Service et les ressources associées que vous avez créés dans ce tutoriel. Pour ce faire, utilisez la fonctionnalité **Supprimer** du tableau de bord DevOps Projects.

## Étapes suivantes

Lorsque vous avez configuré votre processus CI/CD dans ce tutoriel, des pipelines de mise en production et de build ont été automatiquement créés dans Azure DevOps Projects. Vous pouvez le modifier afin qu'il réponde aux besoins de votre équipe. Vous avez appris à effectuer les actions suivantes :

- Utiliser DevOps Projects pour créer un pipeline CI/CD
- Configurer l'accès à votre dépôt GitHub et choisir un framework
- Configurer Azure DevOps et un abonnement Azure
- Valider les modifications apportées à GitHub et les déployer automatiquement dans Azure
- Examiner le pipeline CI/CD Azure Pipelines
- Configurer la surveillance d'Azure Application Insights
- Supprimer des ressources

Pour en savoir plus sur le pipeline CI/CD, consultez :

[Définir votre pipeline de déploiement continu \(CD\) multiétape](#)

# Tutoriel : Déployer votre application ASP.NET sur des machines virtuelles Azure avec Azure DevOps Projects

17/09/2019 • 18 minutes to read • [Edit Online](#)

Azure DevOps Projects offre une expérience simplifiée dans laquelle vous pouvez apporter vos code et dépôt Git existants ou choisir un exemple d'application pour créer un pipeline d'intégration continue (CI) et de livraison continue (CD) dans Azure.

De plus, DevOps Projects :

- Crée automatiquement des ressources Azure, telles qu'une nouvelle machine virtuelle Azure
- Crée et configure un pipeline de mise en production dans Azure DevOps qui inclut un pipeline de build pour CI
- Configurer un pipeline de mise en production pour CD
- Crée une ressource Azure Application Insights pour la supervision

Dans ce tutoriel, vous allez :

- Utiliser DevOps Projects pour déployer votre application ASP.NET
- Configurer Azure DevOps et un abonnement Azure
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications apportées à Azure Repos et les déployer automatiquement dans Azure
- Configurer la surveillance d'Azure Application Insights
- Supprimer des ressources

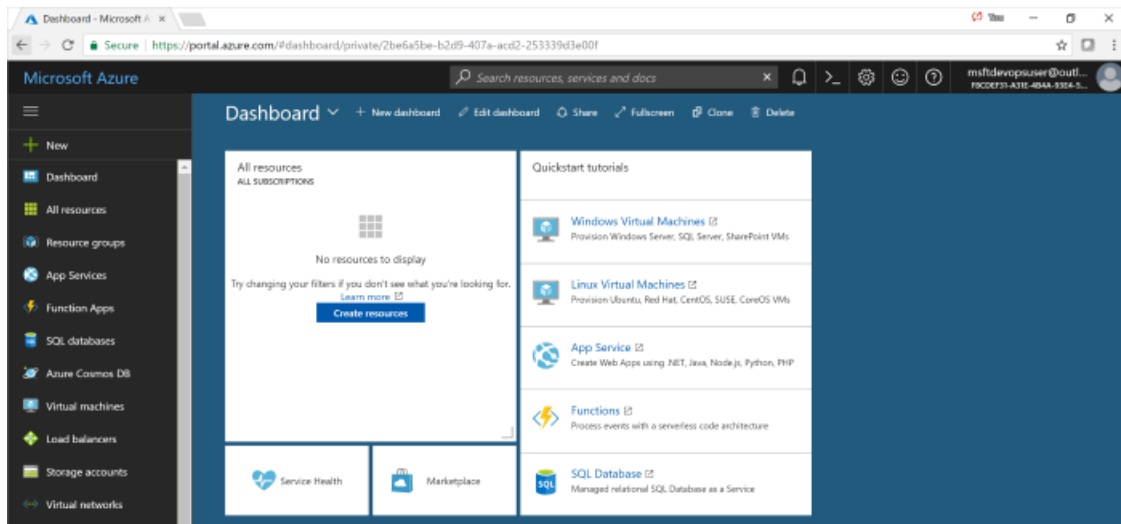
## Prérequis

- Un abonnement Azure. Vous pouvez en obtenir un gratuit via [Visual Studio Dev Essentials](#).

## Utiliser DevOps Projects pour déployer votre application ASP.NET

DevOps Projects crée un pipeline CI/CD dans Azure Pipelines. Vous pouvez créer une organisation Azure DevOps ou utiliser une organisation existante. Azure DevOps Project crée également des ressources Azure, telles que des machines virtuelles dans l'abonnement Azure de votre choix.

1. Connectez-vous au [Portail Azure](#).
2. Dans le volet gauche, sélectionnez **Nouveau...**
3. Dans la zone de recherche, tapez **DevOps Projects**, puis sélectionnez **Créer**.



4. Sélectionnez **.NET**, puis sélectionnez **Suivant**.
5. Sous **Choisir un framework d'application**, sélectionnez **ASP.NET**, puis **Suivant**.  
Le framework d'application, que vous avez choisi à l'étape précédente, détermine le type de cible de déploiement de service Azure disponible ici.
6. Sélectionnez la machine virtuelle, puis sélectionnez **Suivant**.

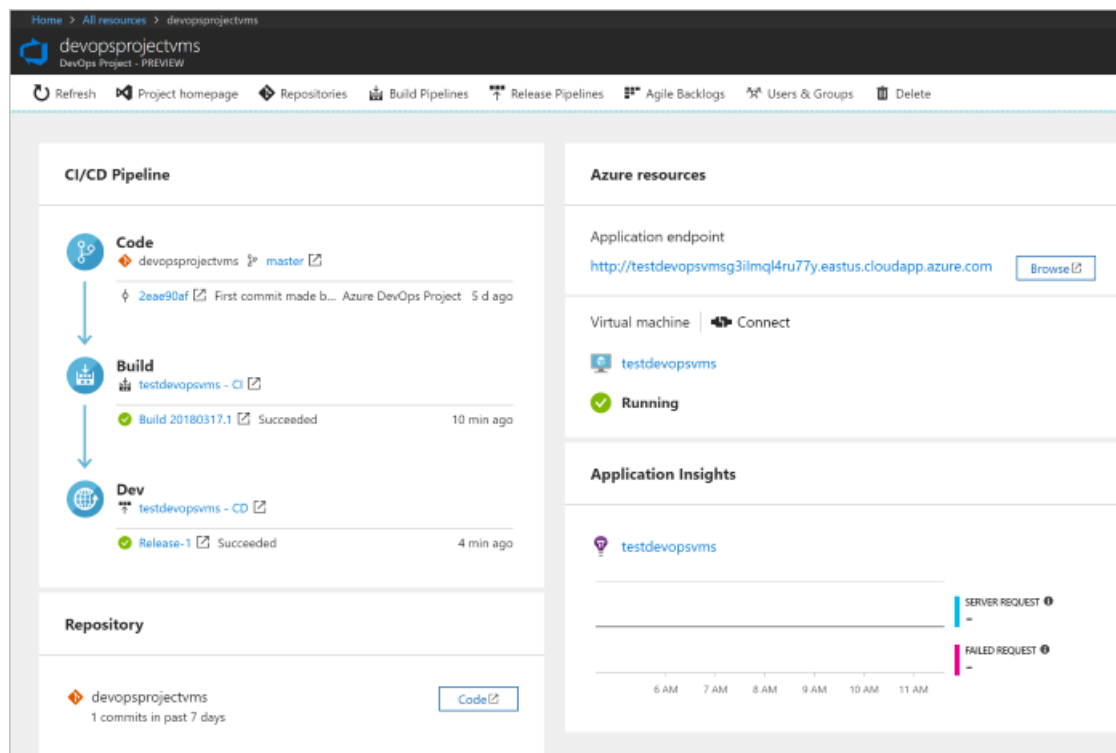
## Configurer Azure DevOps et un abonnement Azure

1. Créez une organisation Azure DevOps ou sélectionnez une organisation existante.
2. Indiquez un nom pour votre projet Azure DevOps.
3. Sélectionnez les services de votre abonnement Azure.  
Si vous le souhaitez, vous pouvez sélectionner **Modifier**, puis entrer plus de détails de configuration, comme l'emplacement des ressources Azure.
4. Entrez un nom de machine virtuelle, un nom d'utilisateur et un mot de passe pour votre nouvelle ressource de machine virtuelle Azure, puis sélectionnez **Terminé**.

La machine virtuelle Azure est mise en service au bout de quelques minutes. Un exemple d'application ASP.NET est configuré dans un dépôt de votre organisation Azure DevOps, une build et une mise en production sont exécutées, et votre application est déployée sur la nouvelle machine virtuelle Azure créée.

Une fois ces opérations terminées, le tableau de bord DevOps Projects s'affiche dans le portail Azure. Vous pouvez également accéder au tableau de bord directement à partir de **Toutes les ressources** dans le portail Azure.

Ce tableau de bord permet de visualiser votre dépôt de code Azure DevOps, le pipeline CI/CD et votre application en cours d'exécution dans Azure.



DevOps Projects configure automatiquement un déclencheur de build et de mise en production CI qui déploie les modification de code dans votre dépôt. Vous pouvez configurer des options supplémentaires dans Azure DevOps. Pour voir votre application en cours d'exécution, sélectionnez **Parcourir**.

## Examiner le pipeline CI

DevOps Projects a configuré automatiquement un pipeline CI/CD dans Azure Pipelines. Vous pouvez explorer et personnaliser le pipeline. Pour vous familiariser avec le pipeline de build, effectuez les étapes suivantes :

1. En haut du tableau de bord DevOps Projects, sélectionnez **Pipelines de build**.  
Un onglet du navigateur affiche le pipeline de build de votre nouveau projet.
2. Pointez sur le champ **État**, puis sélectionnez les points de suspension (...).  
Un menu affiche plusieurs options, telles que de la mise en file d'attente d'une nouvelle build, la mise en pause d'une build et la modification du pipeline de build.
3. Sélectionnez **Modifier**.
4. Dans ce volet, vous pouvez examiner les différentes tâches de votre pipeline de build.  
La build effectue diverses opérations, comme la récupération (fetch) de sources dans le dépôt Git, la restauration de dépendances et la publication de sorties utilisées pour les déploiements.
5. En haut du pipeline de build, sélectionnez le nom du pipeline de build.
6. Remplacez le nom de votre pipeline de build par un nom plus descriptif, sélectionnez **Enregistrer et mettre en file d'attente**, puis sélectionnez **Enregistrer**.
7. Sous le nom de votre pipeline de build, sélectionnez **Historique**.  
Ce volet montre une piste d'audit des modifications que vous avez apportées récemment à la build. Azure DevOps suit les modifications apportées au pipeline de build et permet de comparer les versions.
8. Sélectionnez **Déclencheurs**.  
DevOps Projects crée automatiquement un déclencheur CI tandis que chaque validation dans le dépôt lance une nouvelle build. Si vous le souhaitez, vous pouvez choisir d'inclure ou d'exclure des branches dans le processus CI.

9. Sélectionnez **Rétention**.

En fonction de votre scénario, vous pouvez spécifier des stratégies pour conserver ou supprimer un certain nombre de builds.

## Examiner le pipeline CD

DevOps Projects crée et configure automatiquement les étapes nécessaires au déploiement entre votre organisation Azure DevOps et votre abonnement Azure. Ces étapes comprennent la configuration d'une connexion au service Azure pour authentifier Azure DevOps auprès de votre abonnement Azure. L'automatisation crée également un pipeline CD qui fournit le déploiement continu à la machine virtuelle Azure. Pour en savoir plus sur le pipeline CD Azure DevOps, effectuez les étapes suivantes :

1. Sélectionnez **Build et mise en production**, puis **Mises en production**.

DevOps Projects crée un pipeline de mise en production pour gérer les déploiements sur Azure.

2. Sélectionnez les points de suspension (...) en regard de votre pipeline de mise en production, puis sélectionnez **Modifier**.

Le pipeline de mise en production contient un *pipeline* qui définit le processus de mise en production.

3. Sous **Artefacts**, sélectionnez **Déposer**.

Le pipeline de build que vous avez examiné aux étapes précédentes produit la sortie qui est utilisée pour l'artefact.

4. En regard de l'icône **Déposer**, sélectionnez **Déclencheur de déploiement continu**.

Ce pipeline de mise en production est doté d'un déclencheur de déploiement continu activé qui exécute un déploiement chaque fois qu'un nouvel artefact de build est disponible. Si vous le souhaitez, vous pouvez désactiver le déclencheur afin que vos déploiements nécessitent une exécution manuelle.

5. À gauche, sélectionnez **Tâches**, puis sélectionnez votre environnement.

Les tâches correspondent aux activités exécutées par votre processus de déploiement ; elles sont regroupées par phases. Ce pipeline de mise en production se déroule en deux phases :

- La première phase contient une tâche de déploiement de groupe de ressources Azure qui effectue deux opérations :
  - Configurer la machine virtuelle pour le déploiement
  - Ajouter la nouvelle machine virtuelle à un groupe de déploiement Azure DevOps Le groupe de déploiement de la machine virtuelle dans Azure DevOps gère des groupes logiques de machines cibles de déploiement.
- Dans la deuxième phase, une tâche de gestion d'application web IIS crée un site web IIS sur la machine virtuelle. Une seconde tâche de déploiement d'application web IIS est créée pour déployer le site.

6. À droite, sélectionnez **Afficher les mises en production** pour voir l'historique des mises en production.

7. Sélectionnez les points de suspension (...) en regard d'une mise en production, puis sélectionnez **Ouvrir**.

Vous pouvez explorer plusieurs menus, comme un résumé des mises en production, les éléments de travail associés et les tests.

8. Sélectionnez **Validations**.

Cette vue montre les validations de code qui sont associées à ce déploiement. Comparez les mises en production pour afficher les différences de validation entre les déploiements.

9. Sélectionnez **Journaux d'activité**.

Les journaux d'activité contiennent des informations utiles sur le processus de déploiement. Vous pouvez les voir à la fois pendant et après les déploiements.

## Valider les modifications apportées à Azure Repos et les déployer

# automatiquement dans Azure

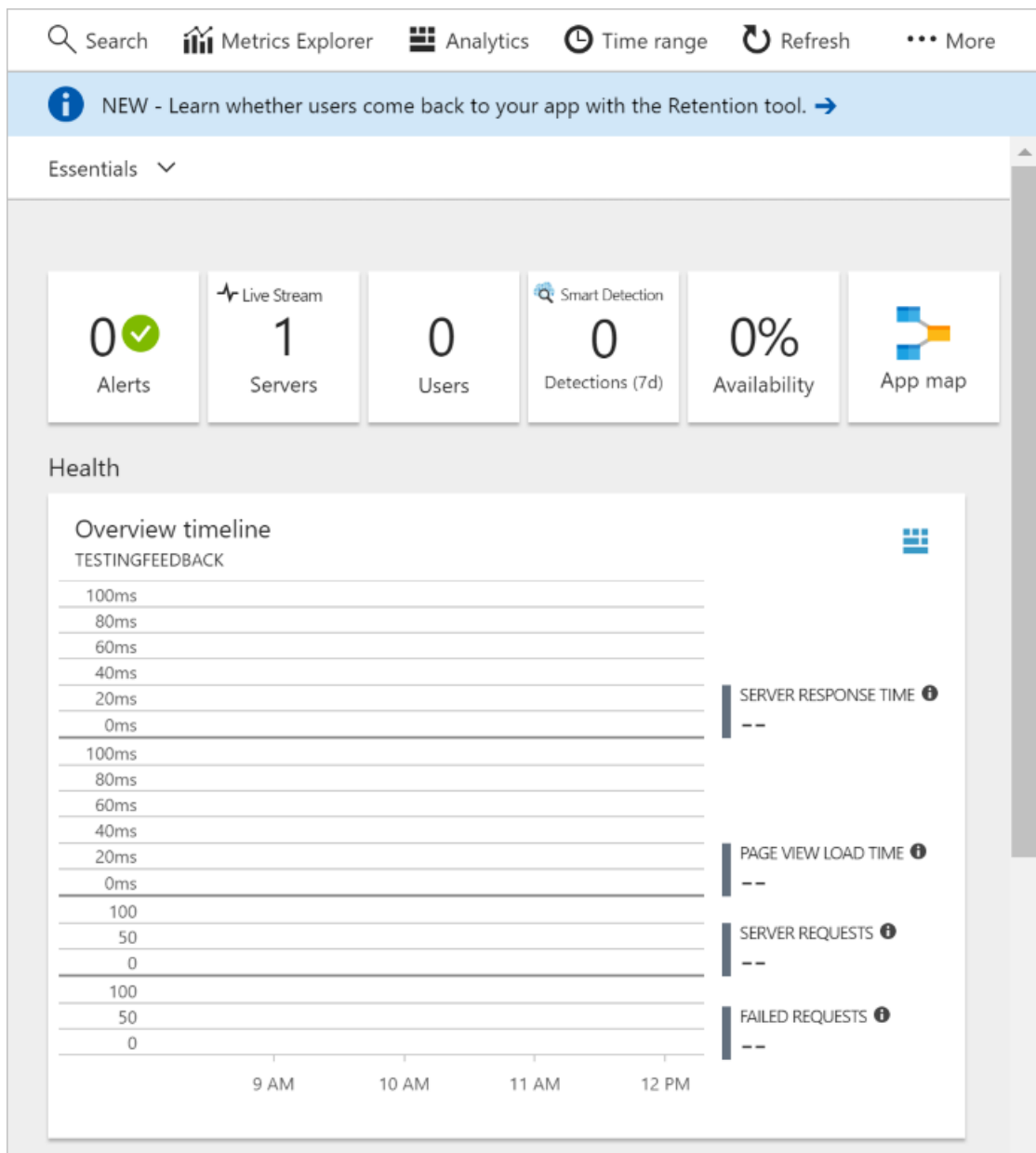
Vous êtes maintenant prêt à collaborer avec une équipe sur votre application, par le biais d'un processus d'intégration et de déploiement continu (CI/CD) qui déploie automatiquement votre dernier travail sur votre site web. Chaque modification apportée au dépôt Git démarre une build dans Azure DevOps tandis qu'un pipeline CD exécute un déploiement dans Azure. Suivez la procédure décrite dans cette section, ou utilisez une autre technique pour valider les modifications dans votre dépôt. Les modifications de code lancent le processus d'intégration et de déploiement continu (CI/CD) et déploient automatiquement vos modifications pour le site web IIS sur la machine virtuelle Azure.

1. Dans le volet gauche, sélectionnez **Code**, puis accédez à votre dépôt.
2. Accédez au répertoire *Views\Home*, sélectionnez les points de suspension (...) situés en regard du fichier *Index.cshtml*, puis sélectionnez **Modifier**.
3. Modifiez le fichier, par exemple en ajoutant du texte dans une des balises `div`.
4. En haut à droite, sélectionnez **Valider**, puis **Valider** à nouveau pour transmettre votre modification. Au bout de quelques instants, une build est démarrée dans Azure DevOps tandis qu'une mise en production est exécutée pour déployer les modifications. Surveillez l'état de la build dans le tableau de bord DevOps Projects, ou dans le navigateur avec votre organisation Azure DevOps.
5. Une fois la mise en production terminée, actualisez votre application pour vérifier vos modifications.

## Configurer la surveillance d'Azure Application Insights

Azure Application Insights vous permet d'analyser facilement les performances et l'utilisation de votre application. DevOps Projects configure automatiquement une ressource Application Insights pour votre application. Vous pouvez configurer différentes capacités d'alertes et de surveillance, si besoin.

1. Dans le portail Azure, accédez au tableau de bord DevOps Projects.
2. En bas à droite, sélectionnez le lien **Application Insights** pour votre application. Le volet **Application Insights** s'ouvre. Cette vue contient les informations sur la surveillance de l'utilisation, des performances et de la disponibilité de votre application.



3. Sélectionnez **Intervalle de temps**, puis **Dernière heure**. Pour filtrer les résultats, sélectionnez **Mettre à jour**.  
Vous pouvez voir à présent toutes les activités des 60 dernières minutes.
4. Pour quitter l'intervalle de temps, sélectionnez **x**.
5. Sélectionnez **Alertes**, puis **Ajouter une alerte Métrique**.
6. Entrez un nom pour l'alerte.
7. Dans la liste déroulante **Métrique**, examinez les différents métriques d'alerte.  
L'alerte par défaut est pour un **temps de réponse serveur supérieur à 1 seconde**. Vous pouvez aisément configurer une variété d'alertes pour améliorer les capacités de surveillance de votre application.
8. Cochez la case **Envoyer un e-mail aux propriétaires, aux contributeurs et aux lecteurs**.  
Si vous le souhaitez, lorsqu'une alerte est affichée, vous pouvez effectuer des actions supplémentaires en exécutant une application logique Azure.
9. Sélectionnez **OK** pour créer l'alerte.  
Au bout de quelques instants, l'alerte apparaît comme étant active sur le tableau de bord.
10. Quittez la zone **Alertes** et revenez au volet **Application Insights**.
11. Sélectionnez **Disponibilité** puis **Ajouter un test**.

12. Entrez un nom de test, puis sélectionnez **Créer**.

Un test Ping basique est créé pour vérifier la disponibilité de votre application. Au bout de quelques instants, les résultats de test sont disponibles, et le tableau de bord Application Insights affiche le statut de disponibilité.

## Supprimer des ressources

Si vous êtes en phase de test, vous pouvez éviter une hausse des frais de facturation en nettoyant vos ressources. Quand elles ne sont plus nécessaires, vous pouvez supprimer la machine virtuelle Azure et les ressources associées que vous avez créées dans ce tutoriel. Pour ce faire, utilisez la fonctionnalité **Supprimer** du tableau de bord DevOps Projects.

### IMPORTANT

La procédure suivante supprime définitivement les ressources. La fonctionnalité *Supprimer* détruit les données qui sont créées par le projet dans DevOps Projects, à la fois dans Azure et dans Azure DevOps, et vous ne pourrez plus les récupérer. Utilisez cette procédure uniquement après avoir lu attentivement les invites.

1. Dans le portail Azure, accédez au tableau de bord DevOps Projects.
2. En haut à droite, sélectionnez **Supprimer**.
3. À l'invite, sélectionnez **Oui** pour *supprimer définitivement* les ressources.

Si vous le souhaitez, vous pouvez modifier ces pipelines de build et de mise en production afin qu'ils répondent aux besoins de votre équipe. Vous pouvez également utiliser ce modèle CI/CD comme modèle pour vos autres pipelines.

## Étapes suivantes

Dans ce tutoriel, vous avez appris à :

- Utiliser DevOps Projects pour déployer votre application ASP.NET
- Configurer Azure DevOps et un abonnement Azure
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications apportées à Azure Repos et les déployer automatiquement dans Azure
- Configurer la surveillance d'Azure Application Insights
- Supprimer des ressources

Pour en savoir plus sur le pipeline CI/CD, consultez :

[Définir votre pipeline de déploiement continu \(CD\) multiétape](#)



# Tutoriel : Déployer votre application ASP.NET et votre code Azure SQL Database à l'aide d'Azure DevOps Projects

17/09/2019 • 18 minutes to read • [Edit Online](#)

Azure DevOps Projects offre une expérience simplifiée dans laquelle vous pouvez apporter vos code et dépôt Git existants ou choisir un exemple d'application pour créer un pipeline d'intégration continue (CI) et de livraison continue (CD) dans Azure.

De plus, DevOps Projects :

- Crée automatiquement des ressources Azure, telles qu'une base de données Azure SQL.
- Crée et configure un pipeline de mise en production dans Azure Pipelines qui inclut un pipeline de build pour CI
- Configurer un pipeline de mise en production pour CD
- Crée une ressource Azure Application Insights pour la supervision

Ce didacticiel présente les procédures suivantes :

- Utiliser Azure DevOps Projects pour déployer votre application ASP.NET et votre code Azure SQL Database
- Configurer Azure DevOps et un abonnement Azure
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications apportées à Azure Repos et les déployer automatiquement dans Azure
- Se connecter à la base de données Azure SQL
- Supprimer des ressources

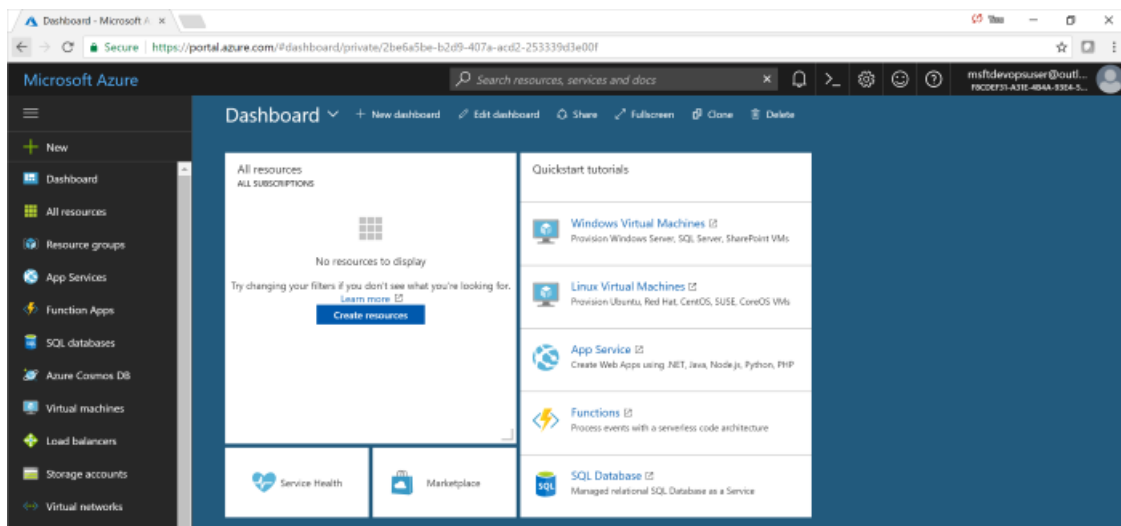
## Prérequis

- Un abonnement Azure. Vous pouvez en obtenir un gratuit via [Visual Studio Dev Essentials](#).

## Créer un projet dans DevOps Projects pour une application ASP.NET et une base de données Azure SQL

DevOps Projects crée un pipeline CI/CD dans Azure Pipelines. Vous pouvez créer une organisation Azure DevOps ou utiliser une organisation existante. DevOps Projects crée également des ressources Azure, telles qu'une base de données Azure SQL, dans l'abonnement Azure de votre choix.

1. Connectez-vous au [Portail Azure](#).
2. Dans le volet de gauche, sélectionnez **Créer une ressource**.
3. Dans la zone de recherche, tapez **DevOps Projects**, puis sélectionnez **Créer**.



4. Sélectionnez **.NET**, puis sélectionnez **Suivant**.
5. Sous **Choisir un framework d'application**, sélectionnez **ASP.NET**.
6. Sélectionnez **Ajouter une base de données**, puis **Suivant**.  
Le framework d'application, que vous avez choisi à l'étape précédente, détermine le type de cible de déploiement de service Azure disponible ici.
7. Sélectionnez **Suivant**.

## Configurer Azure DevOps et un abonnement Azure

1. Créez une organisation Azure DevOps ou sélectionnez une organisation existante.
2. Indiquez un nom pour votre projet Azure DevOps.
3. Sélectionnez les services de votre abonnement Azure.  
Si vous le souhaitez, pour afficher d'autres paramètres de configuration Azure et identifier le nom d'utilisateur dans la section **Détails de connexion au serveur de base de données**, vous pouvez sélectionner **Modifier**. Enregistrez le nom d'utilisateur pour le réutiliser dans les étapes suivantes de ce tutoriel. Si vous effectuez cette étape facultative, quittez la zone de configuration Azure avant de sélectionner **Terminé**.
4. Sélectionnez **Terminé**.  
Après quelques minutes, le tableau de bord DevOps Projects s'ouvre dans le portail Azure. Vous pouvez également accéder au tableau de bord directement à partir de **Toutes les ressources** dans le portail Azure. À droite, sélectionnez **Parcourir** pour voir votre application en cours d'exécution.

## Examiner le pipeline CI

DevOps Projects configure automatiquement un pipeline CI/CD complet dans Azure Repos. Vous pouvez explorer et personnaliser le pipeline. Pour vous familiariser avec le pipeline de build d'Azure DevOps, effectuez les étapes suivantes :

1. En haut du tableau de bord DevOps Projects, sélectionnez **Pipelines de build**.  
Un onglet du navigateur affiche le pipeline de build de votre nouveau projet.
2. Pointez sur le champ **État**, puis sélectionnez les points de suspension (...).  
Un menu affiche plusieurs options, telles que de la mise en file d'attente d'une nouvelle build, la mise en pause d'une build et la modification du pipeline de build.
3. Sélectionnez **Modifier**.

4. Dans ce volet, vous pouvez examiner les différentes tâches de votre pipeline de build.  
La build effectue diverses opérations, comme la récupération (fetch) de sources dans le dépôt Git, la restauration de dépendances et la publication de sorties utilisées pour les déploiements.
5. En haut du pipeline de build, sélectionnez le nom du pipeline de build.
6. Remplacez le nom de votre pipeline de build par un nom plus descriptif, sélectionnez **Enregistrer et mettre en file d'attente**, puis sélectionnez **Enregistrer**.
7. Sous le nom de votre pipeline de build, sélectionnez **Historique**.  
Ce volet montre une piste d'audit des modifications que vous avez apportées récemment à la build. Azure Pipelines suit les modifications apportées au pipeline de build et permet de comparer les versions.
8. Sélectionnez **Déclencheurs**.  
DevOps Projects crée automatiquement un déclencheur CI tandis que chaque validation dans le dépôt lance une nouvelle build. Si vous le souhaitez, vous pouvez choisir d'inclure ou d'exclure des branches dans le processus CI.
9. Sélectionnez **Rétention**.  
En fonction de votre scénario, vous pouvez spécifier des stratégies pour conserver ou supprimer un certain nombre de builds.

## Examiner le pipeline CD

DevOps Projects crée et configure automatiquement les étapes nécessaires au déploiement entre votre organisation Azure DevOps et votre abonnement Azure. Ces étapes comprennent la configuration d'une connexion au service Azure pour authentifier Azure DevOps auprès de votre abonnement Azure. L'automatisation crée également un pipeline CD qui fournit le déploiement continu à la machine virtuelle Azure. Pour en savoir plus sur le pipeline CD Azure DevOps, effectuez les étapes suivantes :

1. Sélectionnez **Build et mise en production**, puis **Mises en production**.  
DevOps Projects crée un pipeline de mise en production pour gérer les déploiements sur Azure.
2. Sélectionnez les points de suspension (...) en regard de votre pipeline de mise en production, puis sélectionnez **Modifier**.  
Le pipeline de mise en production contient un *pipeline* qui définit le processus de mise en production.
3. Sous **Artefacts**, sélectionnez **Déposer**.  
Le pipeline de build que vous avez examiné aux étapes précédentes produit la sortie qui est utilisée pour l'artefact.
4. À droite de l'icône **Déposer**, sélectionnez **Déclencheur de déploiement continu**.  
Ce pipeline de mise en production est doté d'un déclencheur de déploiement continu activé qui exécute un déploiement chaque fois qu'un nouvel artefact de build est disponible. Si vous le souhaitez, vous pouvez désactiver le déclencheur afin que vos déploiements nécessitent une exécution manuelle.  
  
DevOps Projects configure un mot de passe SQL aléatoire et l'utilise pour le pipeline de mise en production.
5. À gauche, sélectionnez **Variables**.

### NOTE

Effectuez l'étape suivante uniquement si vous avez modifié le mot de passe SQL Server. Il existe une seule variable de mot de passe.

6. En regard de la zone **Valeur**, sélectionnez l'icône de cadenas, entrez le nouveau mot de passe, puis sélectionnez **Enregistrer**.

7. À gauche, sélectionnez **Tâches**, puis sélectionnez votre environnement.  
Les tâches correspondent aux activités exécutées par votre processus de déploiement ; elles sont regroupées par phases. Ce pipeline de mise en production est constitué d'une seule phase contenant une tâche de *déploiement d'Azure App Service* et de *déploiement d'Azure SQL Database*.
8. Sélectionnez la tâche *Exécuter Azure SQL* et examinez les diverses propriétés qui sont utilisées pour le déploiement SQL.  
Sous **Package de déploiement**, la tâche utilise un fichier *SQL DACPAC*.
9. À droite, sélectionnez **Afficher les mises en production** pour voir l'historique des mises en production.
10. Sélectionnez les points de suspension (...) en regard d'une mise en production, puis sélectionnez **Ouvrir**.  
Vous pouvez explorer plusieurs menus, comme un résumé des mises en production, les éléments de travail associés et les tests.
11. Sélectionnez **Validations**.  
Cette vue montre les validations de code qui sont associées à ce déploiement. Comparez les mises en production pour afficher les différences de validation entre les déploiements.
12. Sélectionnez **Journaux d'activité**.  
Les journaux d'activité contiennent des informations utiles sur le processus de déploiement. Vous pouvez les voir à la fois pendant et après les déploiements.

## Valider les modifications apportées à Azure Repos et les déployer automatiquement dans Azure

### NOTE

La procédure suivante teste le pipeline CI/CD avec une modification de texte simple. Si vous le souhaitez, pour tester le processus de déploiement SQL, vous pouvez apporter une modification du schéma SQL Server à la table.

Vous êtes maintenant prêt à collaborer avec une équipe sur votre application, par le biais d'un processus d'intégration et de déploiement continu (CI/CD) qui déploie automatiquement votre dernier travail sur votre site web. Chaque modification apportée au dépôt Git démarre une build dans Azure DevOps tandis qu'un pipeline CD exécute un déploiement dans Azure. Suivez la procédure décrite dans cette section, ou utilisez une autre technique pour valider les modifications dans votre dépôt. Les modifications de code lancent le processus CI/CD et déploient automatiquement vos modifications dans Azure.

1. Dans le volet gauche, sélectionnez **Code**, puis accédez à votre dépôt.
2. Accédez au répertoire *SampleWebApplication\Views\Home*, sélectionnez les points de suspension (...) situés en regard du fichier *Index.cshtml*, puis sélectionnez **Modifier**.
3. Modifiez le fichier, par exemple en ajoutant du texte dans une des balises div.
4. En haut à droite, sélectionnez **Valider**, puis **Valider** à nouveau pour transmettre votre modification.  
Au bout de quelques instants, une build est démarrée dans Azure DevOps tandis qu'une mise en production est exécutée pour déployer les modifications. Surveillez l'état de la build dans le tableau de bord DevOps Projects, ou dans le navigateur avec votre organisation Azure DevOps.
5. Une fois la mise en production terminée, actualisez votre application pour vérifier vos modifications.

## Se connecter à la base de données Azure SQL

Vous devez disposer des autorisations appropriées pour vous connecter à la base de données Azure SQL.

1. Dans le tableau de bord DevOps Projects, sélectionnez **SQL Database** pour accéder à la page de gestion de la base de données SQL.
2. Sélectionnez **Définir le pare-feu du serveur**, puis **Ajouter une adresse IP de client**.
3. Sélectionnez **Enregistrer**.  
Votre adresse IP de client est désormais autorisée à accéder à la ressource Azure SQL Server.
4. Revenez au volet **SQL Database**.
5. À droite, sélectionnez le nom du serveur pour accéder à la page de configuration pour **SQL Server**.
6. Sélectionnez **Réinitialiser le mot de passe**, entrez un mot de passe pour la connexion d'administrateur SQL Server, puis sélectionnez **Enregistrer**.  
Veillez à conserver ce mot de passe pour l'utiliser ultérieurement dans ce tutoriel.

Si vous le souhaitez, vous pouvez désormais utiliser les outils clients, tels que SQL Server Management Studio ou Visual Studio, pour vous connecter à SQL Server et à la base de données Azure SQL. Utilisez la propriété **Nom du serveur** pour vous connecter.

Si vous n'avez pas modifié le nom d'utilisateur de la base de données lors de la configuration initiale du projet dans DevOps Projects, votre nom d'utilisateur correspond à la partie locale de votre adresse e-mail. Par exemple, si votre adresse e-mail est *john.doe@microsoft.com*, votre nom d'utilisateur est *john.doe*.

#### NOTE

Si vous modifiez votre mot de passe pour la connexion SQL, vous devez modifier le mot de passe dans la variable du pipeline de mise en production, comme décrit à la section « Examiner le pipeline CD ».

## Supprimer des ressources

Si vous êtes en phase de test, vous pouvez éviter une hausse des frais de facturation en nettoyant vos ressources. Quand elles ne sont plus nécessaires, vous pouvez supprimer la base de données Azure SQL et les ressources associées que vous avez créées dans ce tutoriel. Pour ce faire, utilisez la fonctionnalité **Supprimer** du tableau de bord DevOps Projects.

#### IMPORTANT

La procédure suivante supprime définitivement les ressources. La fonctionnalité *Supprimer* détruit les données qui sont créées par le projet dans DevOps Projects, à la fois dans Azure et dans Azure DevOps, et vous ne pourrez plus les récupérer. Utilisez cette procédure uniquement après avoir lu attentivement les invites.

1. Dans le portail Azure, accédez au tableau de bord DevOps Projects.
2. En haut à droite, sélectionnez **Supprimer**.
3. À l'invite, sélectionnez **Oui** pour *supprimer définitivement* les ressources.

## Étapes suivantes

Si vous le souhaitez, vous pouvez modifier ces pipelines de build et de mise en production afin qu'ils répondent aux besoins de votre équipe. Vous pouvez également utiliser ce modèle CI/CD comme modèle pour vos autres pipelines. Dans ce tutoriel, vous avez appris à :

- Utiliser Azure DevOps Projects pour déployer votre application ASP.NET et votre code Azure SQL Database
- Configurer Azure DevOps et un abonnement Azure
- Examiner le pipeline CI

- Examiner le pipeline CD
- Valider les modifications apportées à Azure Repos et les déployer automatiquement dans Azure
- Se connecter à la base de données Azure SQL
- Supprimer des ressources

Pour en savoir plus sur le pipeline CI/CD, consultez :

[Définir votre pipeline de déploiement continu \(CD\) multiétape](#)

## Videos

# Tutoriel : Déployer des applications ASP.NET Core dans Azure Kubernetes Service avec Azure DevOps Projects

17/09/2019 • 15 minutes to read • [Edit Online](#)

Azure DevOps Projects offre une expérience simplifiée dans laquelle vous pouvez apporter vos code et dépôt Git existants ou choisir un exemple d'application pour créer un pipeline d'intégration continue (CI) et de livraison continue (CD) dans Azure.

De plus, DevOps Projects :

- Crée automatiquement des ressources Azure, telles qu'Azure Kubernetes Service (AKS)
- Crée et configure un pipeline de mise en production dans Azure DevOps qui définit un pipeline de build et de mise en production pour CI/CD.
- Crée une ressource Azure Application Insights pour la supervision
- Permet à [Azure Monitor pour conteneurs](#) de superviser les performances des charges de travail de conteneur sur le cluster AKS

Ce didacticiel présente les procédures suivantes :

- Utiliser DevOps Projects pour déployer une application ASP.NET Core sur AKS
- Configurer Azure DevOps et un abonnement Azure
- Examiner le cluster AKS
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications dans Git et les déployer automatiquement dans Azure
- Supprimer des ressources

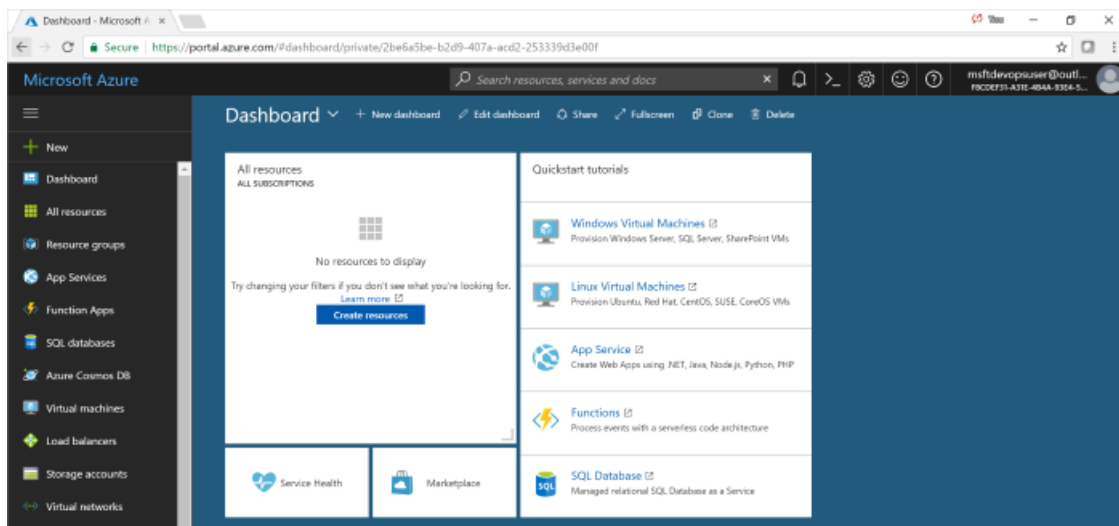
## Prérequis

- Un abonnement Azure. Vous pouvez en obtenir un gratuit via [Visual Studio Dev Essentials](#).

## Utiliser DevOps Projects pour déployer une application ASP.NET Core sur AKS

DevOps Projects crée un pipeline CI/CD dans Azure Pipelines. Vous pouvez créer une organisation Azure DevOps ou utiliser une organisation existante. DevOps Projects crée également des ressources Azure, telles qu'un cluster AKS, dans l'abonnement Azure de votre choix.

1. Connectez-vous au [Portail Azure](#).
2. Dans le volet de gauche, sélectionnez **Créer une ressource**.
3. Dans la zone de recherche, tapez **DevOps Projects**, puis sélectionnez **Créer**.



4. Sélectionnez **.NET**, puis sélectionnez **Suivant**.
5. Sous **Choisir un framework d'application**, sélectionnez **ASP.NET Core**.
6. Sélectionnez **Kubernetes Service**, puis **Suivant**.

## Configurer Azure DevOps et un abonnement Azure

1. Créez une organisation Azure DevOps ou sélectionnez une organisation existante.
2. Indiquez un nom pour votre projet Azure DevOps.
3. Sélectionnez votre abonnement Azure.
4. Pour afficher des paramètres de configuration Azure supplémentaires et déterminer le nombre de nœuds du cluster AKS, sélectionnez **Modifier**.  
Ce volet regroupe différentes options pour la configuration du type et de l'emplacement des services Azure.
5. Quittez la zone de configuration Azure, puis sélectionnez **Terminé**.  
Le processus est achevé au bout de quelques minutes. Un exemple d'application ASP.NET Core est configuré dans un dépôt Git de votre organisation Azure DevOps, un cluster AKS est créé, un pipeline CI/CD est exécuté et votre application est déployée sur Azure.

Une fois toutes ces opérations terminées, le tableau de bord Azure DevOps Projects s'affiche dans le portail Azure. Vous pouvez également accéder au tableau de bord DevOps Projects directement à partir de **Toutes les ressources** dans le portail Azure.

Ce tableau de bord permet de visualiser votre dépôt de code Azure DevOps, le pipeline CI/CD et votre cluster AKS. Vous pouvez configurer d'autres options de CI/CD dans votre pipeline Azure DevOps. À droite, sélectionnez **Parcourir** pour voir votre application en cours d'exécution.

## Examiner le cluster AKS

DevOps Projects configure automatiquement un cluster AKS que vous pouvez explorer et personnaliser. Pour vous familiariser avec le cluster AKS, effectuez les étapes suivantes :

1. Accédez au tableau de bord DevOps Projects.
2. À droite, sélectionnez le service AKS.  
Un volet s'ouvre pour le cluster AKS. À partir de cette vue, vous pouvez effectuer diverses actions, comme superviser l'intégrité des conteneurs, effectuer des recherches dans les journaux d'activité et ouvrir le tableau de bord Kubernetes.



3. À droite, sélectionnez **Voir le tableau de bord Kubernetes**.  
Si vous le souhaitez, suivez ces étapes pour ouvrir le tableau de bord Kubernetes.

## Examiner le pipeline CI

DevOps Projects configure automatiquement un pipeline CI/CD dans votre organisation Azure DevOps. Vous pouvez explorer et personnaliser le pipeline. Pour vous familiariser avec celui-ci, procédez comme suit :

1. Accédez au tableau de bord DevOps Projects.
2. En haut du tableau de bord DevOps Projects, sélectionnez **Pipelines de build**.  
Un onglet du navigateur affiche le pipeline de build de votre nouveau projet.
3. Pointez sur le champ **État**, puis sélectionnez les points de suspension (...).  
Un menu affiche plusieurs options, telles que de la mise en file d'attente d'une nouvelle build, la mise en pause d'une build et la modification du pipeline de build.
4. Sélectionnez **Modifier**.
5. Dans ce volet, vous pouvez examiner les différentes tâches de votre pipeline de build.  
La build effectue diverses opérations, comme la récupération (fetch) de sources dans le dépôt Git, la restauration de dépendances et la publication de sorties utilisées pour les déploiements.
6. En haut du pipeline de build, sélectionnez le nom du pipeline de build.
7. Remplacez le nom de votre pipeline de build par un nom plus descriptif, sélectionnez **Enregistrer et mettre en file d'attente**, puis sélectionnez **Enregistrer**.
8. Sous le nom de votre pipeline de build, sélectionnez **Historique**.  
Ce volet montre une piste d'audit des modifications que vous avez apportées récemment à la build. Azure DevOps suit les modifications apportées au pipeline de build et permet de comparer les versions.
9. Sélectionnez **Déclencheurs**.  
DevOps Projects crée automatiquement un déclencheur CI tandis que chaque validation dans le dépôt lance une nouvelle build. Si vous le souhaitez, vous pouvez choisir d'inclure ou d'exclure des branches dans le processus CI.
10. Sélectionnez **Rétention**.  
En fonction de votre scénario, vous pouvez spécifier des stratégies pour conserver ou supprimer un certain nombre de builds.

## Examiner le pipeline de mise en production CD

DevOps Projects crée et configure automatiquement les étapes nécessaires au déploiement entre votre organisation Azure DevOps et votre abonnement Azure. Ces étapes comprennent la configuration d'une connexion au service Azure pour authentifier Azure DevOps auprès de votre abonnement Azure. L'automatisation crée également un pipeline de mise en production, qui fournit le déploiement continu vers Azure. Pour en savoir plus sur le pipeline de mise en production, suivez ces étapes :

1. Sélectionnez **Build et mise en production**, puis **Mises en production**.  
DevOps Projects crée un pipeline de mise en production pour gérer les déploiements sur Azure.
2. Sélectionnez les points de suspension (...) en regard de votre pipeline de mise en production, puis sélectionnez **Modifier**.  
Le pipeline de mise en production contient un *pipeline* qui définit le processus de mise en production.
3. Sous **Artefacts**, sélectionnez **Déposer**.  
Le pipeline de build que vous avez examiné aux étapes précédentes produit la sortie qui est utilisée pour

l'artefact.

4. À droite de l'icône **Déposer**, sélectionnez **Déclencheur de déploiement continu**.  
Ce pipeline de mise en production est doté d'un déclencheur de déploiement continu activé qui exécute un déploiement chaque fois qu'un nouvel artefact de build est disponible. Si vous le souhaitez, vous pouvez désactiver le déclencheur afin que vos déploiements nécessitent une exécution manuelle.
5. À droite, sélectionnez **Afficher les mises en production** pour voir l'historique des mises en production.
6. Sélectionnez les points de suspension (...) en regard d'une mise en production, puis sélectionnez **Ouvrir**.  
Vous pouvez explorer plusieurs menus, comme un résumé des mises en production, les éléments de travail associés et les tests.
7. Sélectionnez **Validations**.  
Cette vue montre les validations de code qui sont associées à ce déploiement. Comparez les mises en production pour afficher les différences de validation entre les déploiements.
8. Sélectionnez **Journaux d'activité**.  
Les journaux d'activité contiennent des informations utiles sur le processus de déploiement. Vous pouvez les voir à la fois pendant et après les déploiements.

## Valider les modifications apportées à Azure Repos et les déployer automatiquement dans Azure

### NOTE

La procédure suivante teste le pipeline CI/CD en apportant une modification de texte simple.

Vous êtes maintenant prêt à collaborer avec une équipe sur votre application, par le biais d'un processus d'intégration et de déploiement continu (CI/CD) qui déploie automatiquement votre dernier travail sur votre site web. Chaque modification apportée au dépôt Git démarre une build dans Azure DevOps tandis qu'un pipeline CD exécute un déploiement dans Azure. Suivez la procédure décrite dans cette section, ou utilisez une autre technique pour valider les modifications dans votre dépôt. Par exemple, vous pouvez cloner le dépôt Git dans votre outil ou environnement de développement intégré favori, puis envoyer (push) les modifications vers ce dépôt.

1. Dans le menu Azure DevOps, sélectionnez **Code** > **Fichiers**, puis accédez à votre dépôt.
2. Accédez au répertoire *Views\Home*, sélectionnez les points de suspension (...) situés en regard du fichier *Index.cshtml*, puis sélectionnez **Modifier**.
3. Modifiez le fichier, par exemple en ajoutant du texte dans une des balises div.
4. En haut à droite, sélectionnez **Valider**, puis **Valider** à nouveau pour transmettre votre modification.  
Au bout de quelques instants, une build est démarrée dans Azure DevOps tandis qu'une mise en production est exécutée pour déployer les modifications. Supervisez l'état de la build dans le tableau de bord DevOps Projects, ou dans le navigateur avec votre organisation Azure DevOps.
5. Une fois la mise en production terminée, actualisez votre application pour vérifier vos modifications.

## Supprimer des ressources

Si vous êtes en phase de test, vous pouvez éviter une hausse des frais de facturation en nettoyant vos ressources. Quand ils ne sont plus nécessaires, vous pouvez supprimer le cluster AKS et les ressources associées que vous avez créés dans ce tutoriel. Pour ce faire, utilisez la fonctionnalité **Supprimer** du tableau de bord DevOps Projects.

### IMPORTANT

La procédure suivante supprime définitivement les ressources. La fonctionnalité *Supprimer* détruit les données qui sont créées par le projet dans DevOps Projects, à la fois dans Azure et dans Azure DevOps, et vous ne pourrez plus les récupérer. Utilisez cette procédure uniquement après avoir lu attentivement les invites.

1. Dans le portail Azure, accédez au tableau de bord DevOps Projects.
2. En haut à droite, sélectionnez **Supprimer**.
3. À l'invite, sélectionnez **Oui** pour *supprimer définitivement* les ressources.

## Étapes suivantes

Si vous le souhaitez, vous pouvez modifier ces pipelines de build et de mise en production afin qu'ils répondent aux besoins de votre équipe. Vous pouvez également utiliser ce modèle CI/CD comme modèle pour vos autres pipelines. Dans ce tutoriel, vous avez appris à :

- Utiliser DevOps Projects pour déployer une application ASP.NET Core sur AKS
- Configurer Azure DevOps et un abonnement Azure
- Examiner le cluster AKS
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications dans Git et les déployer automatiquement dans Azure
- Supprimer des ressources

Pour en savoir plus sur l'utilisation du tableau de bord Kubernetes, consultez :

[Utiliser le tableau de bord Kubernetes](#)

# Tutoriel : Déployer votre application ASP.NET Core dans Azure Service Fabric avec Azure DevOps Projects

17/09/2019 • 14 minutes to read • [Edit Online](#)

Azure DevOps Projects offre une expérience simplifiée dans laquelle vous pouvez apporter vos code et dépôt Git existants ou choisir un exemple d'application pour créer un pipeline d'intégration continue (CI) et de livraison continue (CD) dans Azure.

De plus, DevOps Projects :

- Crée automatiquement des ressources Azure, telles qu'Azure Service Fabric
- Crée et configure un pipeline de mise en production dans Azure DevOps qui définit un pipeline CI/CD
- Crée une ressource Azure Application Insights pour la supervision

Ce didacticiel présente les procédures suivantes :

- Utiliser DevOps Projects pour créer une application ASP.NET Core et la déployer sur Service Fabric
- Configurer Azure DevOps et un abonnement Azure
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications dans git et les déployer automatiquement dans Azure
- Supprimer des ressources

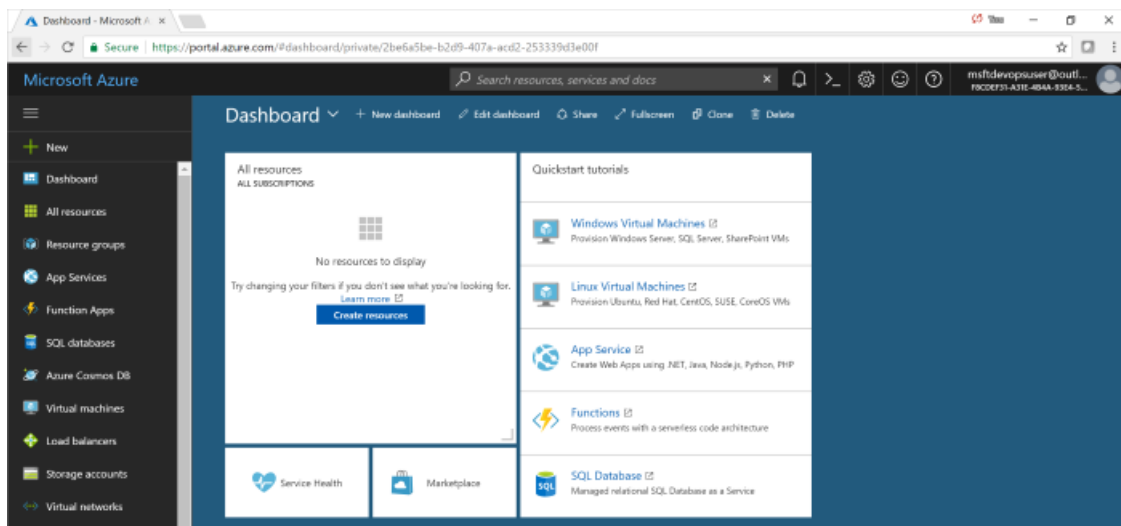
## Prérequis

- Un abonnement Azure. Vous pouvez en obtenir un gratuit via [Visual Studio Dev Essentials](#).

## Utiliser DevOps Projects pour créer une application ASP.NET Core et la déployer sur Service Fabric

DevOps Projects crée un pipeline CI/CD dans Azure Pipelines. Vous pouvez créer une organisation Azure DevOps ou utiliser une organisation existante. DevOps Projects crée également des ressources Azure, telles qu'un cluster Service Fabric, dans l'abonnement Azure de votre choix.

1. Connectez-vous au [Portail Azure](#).
2. Dans le volet de gauche, sélectionnez **Créer une ressource**.
3. Dans la zone de recherche, tapez **DevOps Projects**, puis sélectionnez **Créer**.



4. Sélectionnez **.NET**, puis **Suivant**.
5. Sous **Choisir un framework d'application**, sélectionnez **ASP.NET Core**, puis **Suivant**.
6. Sélectionnez **Cluster Service Fabric**, puis **Suivant**.

## Configurer Azure DevOps et un abonnement Azure

1. Créez une organisation Azure DevOps ou sélectionnez une organisation existante.
2. Indiquez un nom pour votre projet Azure DevOps.
3. Sélectionnez votre abonnement Azure.
4. Pour afficher d'autres paramètres de configuration Azure et identifier la taille de la machine virtuelle du nœud ainsi que le système d'exploitation du cluster Service Fabric, sélectionnez **Modifier**.  
Ce volet regroupe différentes options pour la configuration du type et de l'emplacement des services Azure.
5. Quittez la zone de configuration Azure, puis sélectionnez **Terminé**.  
Le processus est achevé au bout de quelques minutes. Un exemple d'application ASP.NET Core est configuré dans un dépôt Git de votre organisation Azure DevOps, un cluster Service Fabric est créé, un pipeline CI/CD est exécuté et votre application est déployée sur Azure.

Une fois toutes ces opérations terminées, le tableau de bord DevOps Projects s'affiche dans le portail Azure. Vous pouvez également accéder au tableau de bord DevOps Projects directement à partir de **Toutes les ressources** dans le portail Azure.

Ce tableau de bord permet de visualiser votre dépôt de code Azure DevOps, le pipeline CI/CD et le cluster Service Fabric. Vous pouvez configurer d'autres options pour votre pipeline CI/CD dans Azure Repos. À droite, sélectionnez **Parcourir** pour voir votre application en cours d'exécution.

## Examiner le pipeline CI

DevOps Projects configure automatiquement un pipeline CI/CD dans Azure Pipelines. Vous pouvez explorer et personnaliser le pipeline. Pour vous familiariser avec celui-ci, procédez comme suit :

1. Accédez au tableau de bord DevOps Projects.
2. En haut du tableau de bord DevOps Projects, sélectionnez **Pipelines de build**.  
Un onglet du navigateur affiche le pipeline de build de votre nouveau projet.
3. Pointez sur le champ **État**, puis sélectionnez les points de suspension (...).  
Un menu affiche plusieurs options, telles que de la mise en file d'attente d'une nouvelle build, la mise en

pause d'une build et la modification du pipeline de build.

4. Sélectionnez **Modifier**.

5. Dans ce volet, vous pouvez examiner les différentes tâches de votre pipeline de build.

La build effectue diverses opérations, comme la récupération (fetch) de sources dans le dépôt Git, la restauration de dépendances et la publication de sorties utilisées pour les déploiements.

6. En haut du pipeline de build, sélectionnez le nom du pipeline de build.

7. Sous le nom de votre pipeline de build, sélectionnez **Historique**.

Ce volet montre une piste d'audit des modifications que vous avez apportées récemment à la build. Azure DevOps suit les modifications apportées au pipeline de build et permet de comparer les versions.

8. Sélectionnez **Déclencheurs**.

DevOps Projects crée automatiquement un déclencheur CI tandis que chaque validation dans le dépôt lance une nouvelle build. Si vous le souhaitez, vous pouvez choisir d'inclure ou d'exclure des branches dans le processus CI.

9. Sélectionnez **Rétention**.

En fonction de votre scénario, vous pouvez spécifier des stratégies pour conserver ou supprimer un certain nombre de builds.

## Examiner le pipeline CD

DevOps Projects crée et configure automatiquement les étapes nécessaires au déploiement entre votre organisation Azure DevOps et votre abonnement Azure. Ces étapes comprennent la configuration d'une connexion au service Azure pour authentifier Azure DevOps auprès de votre abonnement Azure. L'automatisation crée également un pipeline de mise en production, qui fournit le déploiement continu vers Azure. Pour en savoir plus sur le pipeline de mise en production, suivez ces étapes :

1. Sélectionnez **Build et mise en production**, puis **Mises en production**.

DevOps Projects crée un pipeline de mise en production pour gérer les déploiements sur Azure.

2. Sélectionnez les points de suspension (...) en regard de votre pipeline de mise en production, puis sélectionnez **Modifier**.

Le pipeline de mise en production contient un *pipeline* qui définit le processus de mise en production.

3. Sous **Artefacts**, sélectionnez **Déposer**.

Le pipeline de build que vous avez examiné précédemment génère la sortie qui est utilisée pour l'artefact.

4. À droite de l'icône **Déposer**, sélectionnez **Déclencheur de déploiement continu**.

Ce pipeline de mise en production est doté d'un déclencheur de déploiement continu activé qui exécute un déploiement chaque fois qu'un nouvel artefact de build est disponible. Si vous le souhaitez, vous pouvez désactiver le déclencheur afin que vos déploiements nécessitent une exécution manuelle.

5. À droite, sélectionnez **Afficher les mises en production** pour voir l'historique des mises en production.

6. Sélectionnez les points de suspension (...) en regard d'une mise en production, puis sélectionnez **Ouvrir**.

Vous pouvez explorer plusieurs menus, comme un résumé des mises en production, les éléments de travail associés et les tests.

7. Sélectionnez **Validations**.

Cette vue montre les validations de code qui sont associées à ce déploiement. Comparez les mises en production pour afficher les différences de validation entre les déploiements.

8. Sélectionnez **Journaux d'activité**.

Les journaux d'activité contiennent des informations utiles sur le processus de déploiement. Vous pouvez les

voir à la fois pendant et après les déploiements.

## Valider les modifications dans Git et les déployer automatiquement dans Azure

### NOTE

La procédure suivante teste le pipeline CI/CD en apportant une modification de texte simple.

Vous êtes maintenant prêt à collaborer avec une équipe sur votre application, par le biais d'un processus d'intégration et de déploiement continu (CI/CD) qui déploie automatiquement votre dernier travail sur votre site web. Chaque modification apportée au dépôt Git démarre un build tandis qu'une mise en production déploie ces modifications sur Azure. Suivez la procédure décrite dans cette section, ou utilisez une autre technique pour valider les modifications dans votre dépôt. Par exemple, vous pouvez cloner le dépôt Git dans votre outil ou environnement de développement intégré favori, puis envoyer (push) les modifications vers ce dépôt.

1. Dans le menu Azure DevOps, sélectionnez **Code** > **Fichiers**, puis accédez à votre dépôt.
2. Accédez au répertoire *Views\Home*, sélectionnez les points de suspension (...) situés en regard du fichier *Index.cshtml*, puis sélectionnez **Modifier**.
3. Modifiez le fichier, par exemple en ajoutant du texte dans une des balises div.
4. En haut à droite, sélectionnez **Valider**, puis **Valider** à nouveau pour transmettre votre modification. Au bout de quelques instants, un build est démarré, puis une mise en production est exécutée pour déployer les modifications. Vous pouvez superviser l'état du build dans le tableau de bord DevOps Projects, ou dans le navigateur avec la journalisation en temps réel d'Azure DevOps.
5. Une fois la mise en production terminée, actualisez votre application pour vérifier vos modifications.

## Supprimer des ressources

Si vous êtes en phase de test, vous pouvez éviter une hausse des frais de facturation en nettoyant vos ressources. Quand ils ne sont plus nécessaires, vous pouvez supprimer le cluster Azure Service Fabric et les ressources associées que vous avez créés dans ce tutoriel. Pour ce faire, utilisez la fonctionnalité **Supprimer** du tableau de bord DevOps Projects.

### IMPORTANT

La procédure suivante supprime définitivement les ressources. La fonctionnalité *Supprimer* détruit les données qui sont créées par le projet dans DevOps Projects, à la fois dans Azure et dans Azure DevOps, et vous ne pourrez plus les récupérer. Utilisez cette procédure uniquement après avoir lu attentivement les invites.

1. Dans le portail Azure, accédez au tableau de bord DevOps Projects.
2. En haut à droite, sélectionnez **Supprimer**.
3. À l'invite, sélectionnez **Oui** pour *supprimer définitivement* les ressources.

## Étapes suivantes

Si vous le souhaitez, vous pouvez modifier le pipeline CI/CD Azure afin qu'il réponde aux besoins de votre équipe. Vous pouvez également utiliser ce modèle CI/CD comme modèle pour vos autres pipelines. Dans ce tutoriel, vous avez appris à :

- Utiliser DevOps Projects pour créer une application ASP.NET Core et la déployer sur Service Fabric

- Configurer Azure DevOps et un abonnement Azure
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications dans Git et les déployer automatiquement dans Azure
- Supprimer des ressources

Pour en savoir plus sur Service Fabric et les microservices, voir :

[Utiliser une approche de microservices pour la conception d'applications](#)



# Déployer des applications node.js alimentées par Azure Cosmos DB avec DevOps Projects

17/09/2019 • 14 minutes to read • [Edit Online](#)

Azure DevOps Projects offre une expérience rationalisée dans laquelle vous pouvez créer un pipeline d'intégration continue (CI) et de déploiement continu (CD) dans Azure. Pour ce faire, utilisez votre code et votre référentiel Git existants ou sélectionnez un exemple d'application.

De plus, DevOps Projects :

- Crée automatiquement des ressources Azure, telles qu'Azure Cosmos DB, Azure Application Insights, Azure App Service et des plans App Service
- Crée et configure un pipeline de mise en production CI/CD dans Azure DevOps

Ce didacticiel présente les procédures suivantes :

- Utiliser DevOps Projects pour déployer une application Node.js alimentée par Azure Cosmos DB
- Configurer Azure DevOps et un abonnement Azure
- Examiner Azure Cosmos DB
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications dans Git et les déployer automatiquement dans Azure
- Nettoyer les ressources

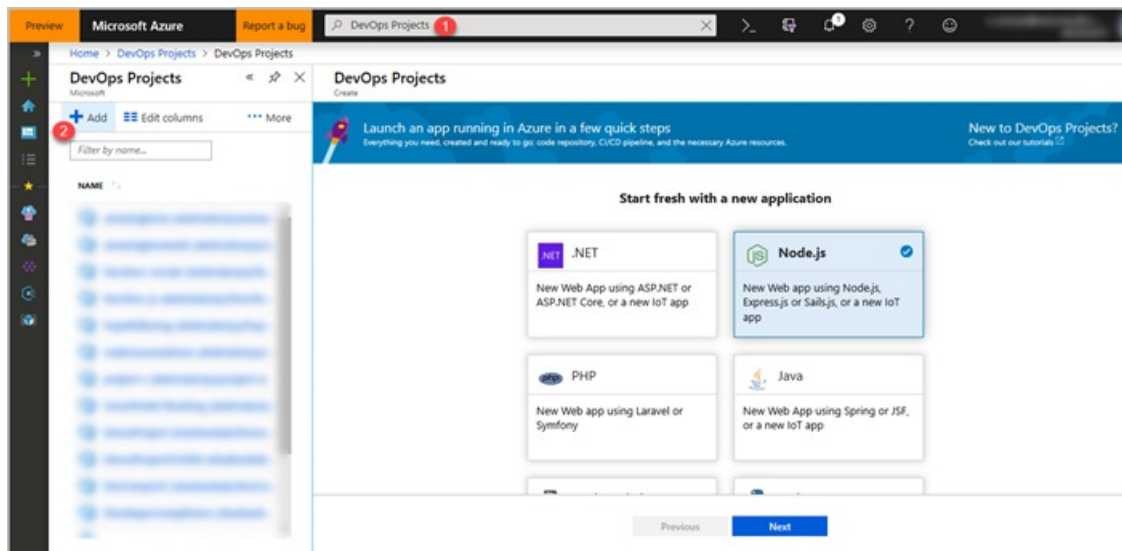
## Prérequis

Vous avez besoin d'un abonnement Azure. Vous pouvez en obtenir un gratuitement via [Visual Studio Dev Essentials](#).

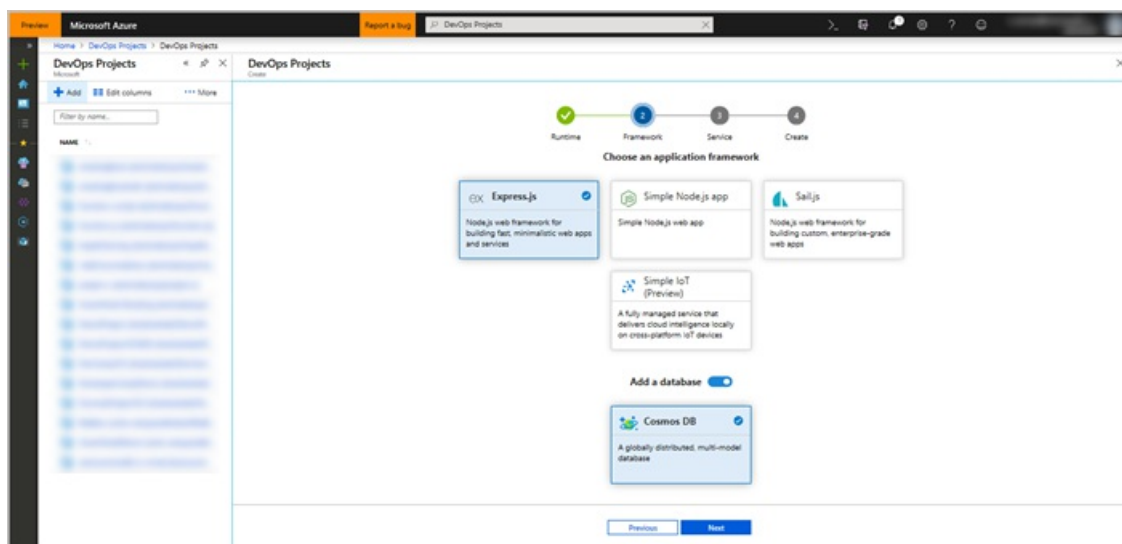
## Utiliser DevOps Projects pour déployer l'application Node.js

DevOps Projects crée un pipeline CI/CD dans Azure Pipelines. Vous pouvez créer une organisation Azure DevOps ou utiliser une organisation existante. DevOps Projects crée également des ressources Azure, telles qu'Azure Cosmos DB, Application Insights, App Service et des plans App Service, dans l'abonnement Azure de votre choix.

1. Connectez-vous au [Portail Azure](#).
2. Dans le volet de gauche, sélectionnez **Créer une ressource**.
3. Dans la zone de recherche, saisissez **DevOps Projects**, puis sélectionnez **Ajouter**.



4. Sélectionnez **Node.js** comme runtime, puis cliquez sur **Suivant**. Sous **Choisir un framework d'application**, sélectionnez **Express.js**.
5. Activez la section **Ajouter une base de données** pour **Cosmos DB**, puis cliquez sur **Suivant**.



Azure DevOps Projects prend en charge différentes infrastructures d'application, telles qu'**Express.js**, **Sample Node.js app** et **Sail.js**. Dans ce tutoriel, nous allons utiliser **Express.js**.

6. Sélectionnez un service Azure pour déployer l'application, puis cliquez sur **Suivant**. Vos options incluent Windows Web App, Azure Kubernetes Service et Azure Web App pour conteneurs. Dans ce tutoriel, nous allons utiliser **Windows Web App**.

## Configurer Azure DevOps et l'abonnement Azure

1. Indiquez un nom pour votre projet Azure DevOps.
2. Créez une organisation Azure DevOps ou sélectionnez une organisation existante.
3. Sélectionnez votre abonnement Azure.
4. Pour voir d'autres paramètres de configuration Azure ou pour identifier le niveau tarifaire et l'emplacement, cliquez sur **Paramètres supplémentaires**. Ce volet affiche différentes options pour la configuration du niveau tarifaire et de l'emplacement des services Azure.
5. Quittez la zone de configuration Azure, puis sélectionnez **Terminé**.
6. Le processus se termine au bout de quelques minutes. Un exemple d'application Node.js est configuré dans

un référentiel Git dans votre organisation Azure DevOps. Ensuite, Azure Cosmos DB, App Service, un plan App Service et des ressources Application Insights sont créés, ainsi qu'un pipeline CI/CD. Votre application est ensuite déployée sur Azure.

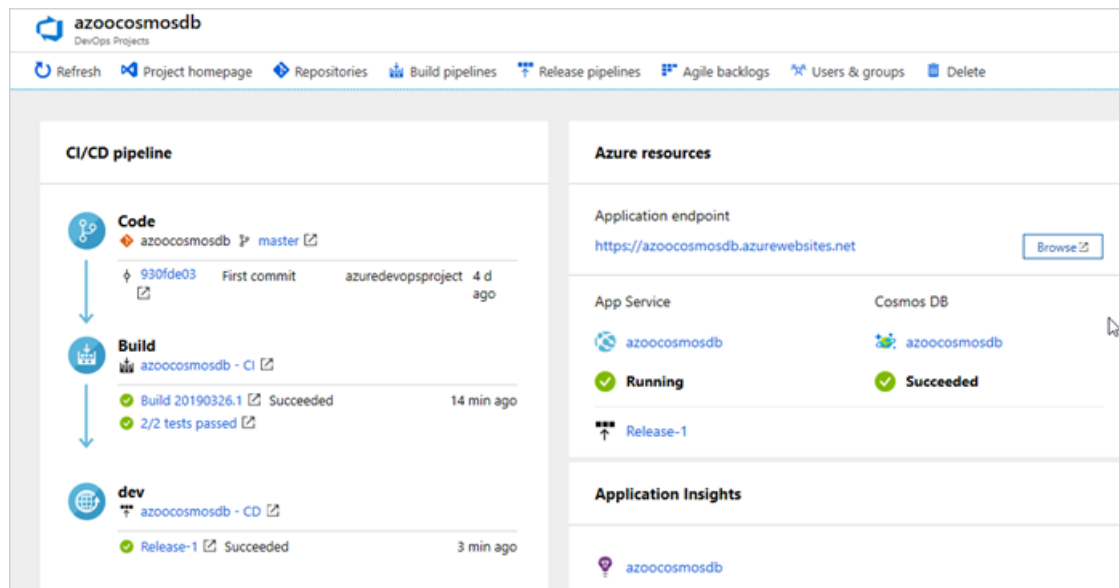
Une fois tous ces processus terminés, le tableau de bord Azure DevOps Projects s'affiche dans le Portail Azure. Vous pouvez également accéder au tableau de bord DevOps Projects directement à partir de **Toutes les ressources** dans le portail Azure.

Ce tableau de bord permet de visualiser votre référentiel de code Azure DevOps, votre pipeline CI/CD et votre base de données Azure Cosmos DB. Vous pouvez configurer d'autres options de CI/CD dans votre pipeline Azure DevOps. Sur la droite du tableau de bord, sélectionnez **Azure Cosmos DB** pour afficher ces options.

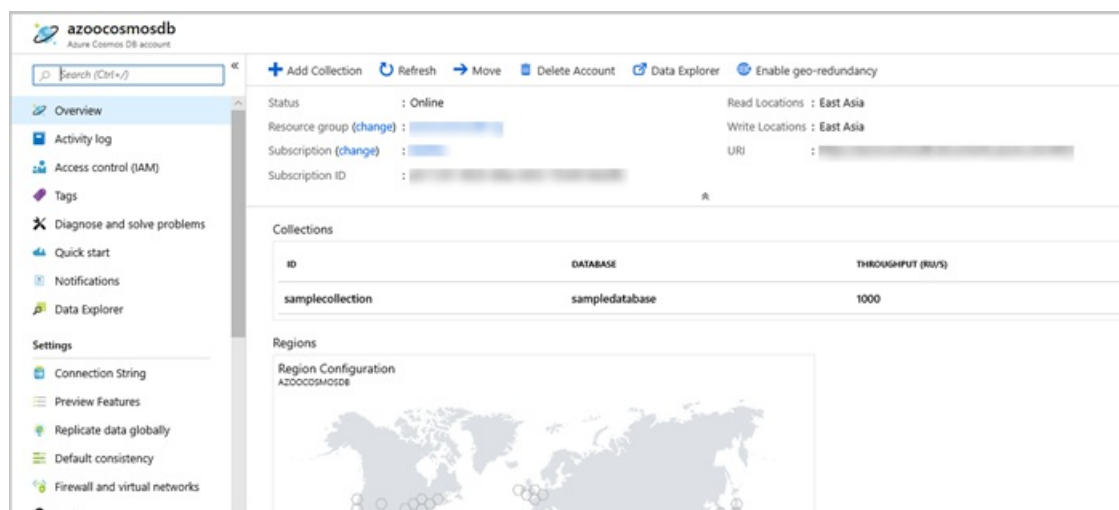
## Examiner Azure Cosmos DB

DevOps Projects configure automatiquement Azure Cosmos DB, que vous pouvez explorer et personnaliser. Pour vous familiariser avec la base de données Azure Cosmos DB, effectuez les étapes suivantes :

1. Accédez au tableau de bord DevOps Projects.



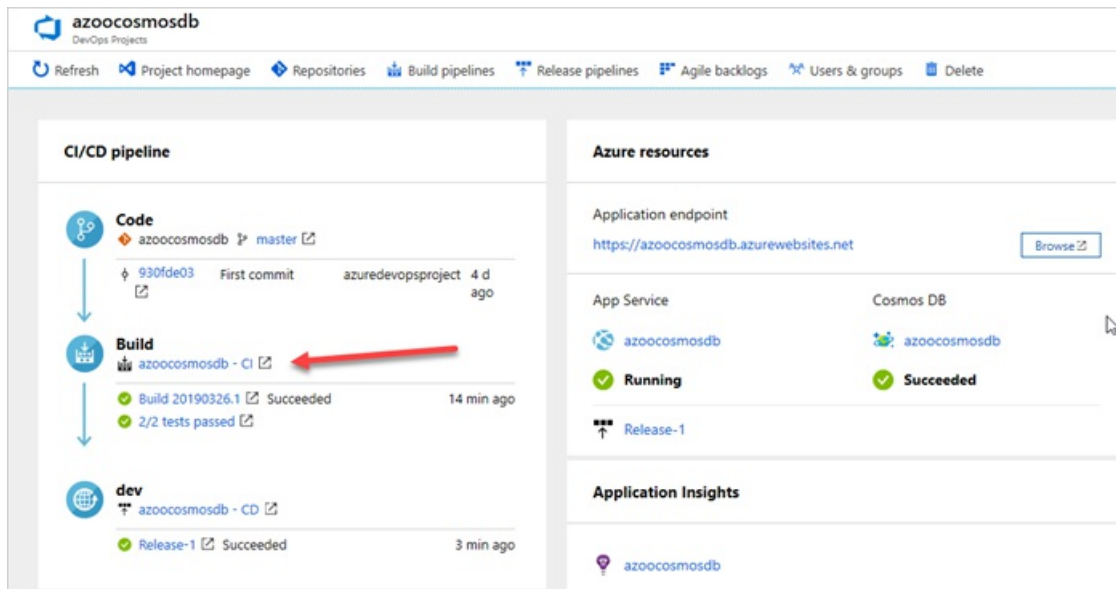
2. Sur la droite, sélectionnez Azure Cosmos DB. Un volet s'ouvre pour la base de données Azure Cosmos DB. À partir de cette vue, vous pouvez effectuer diverses actions, par exemple superviser les opérations et effectuer des recherches dans les journaux.



## Examiner le pipeline CI

DevOps Projects configure automatiquement un pipeline CI/CD dans votre organisation Azure DevOps. Vous pouvez explorer et personnaliser le pipeline. Pour vous familiariser avec celui-ci, procédez comme suit :

1. Accédez au tableau de bord DevOps Projects.
2. Cliquez sur le lien hypertexte sous **Build**. Un onglet du navigateur affiche le pipeline de build de votre nouveau projet.



3. Sélectionnez **Modifier**. Dans ce volet, vous pouvez examiner les différentes tâches de votre pipeline de build. La build effectue différentes tâches, telles que l'extraction du code source à partir du dépôt git, la génération de l'application, l'exécution de tests unitaires et la publication des sorties utilisées pour les déploiements.
4. Sélectionnez **Déclencheurs**. DevOps Projects crée automatiquement un déclencheur CI tandis que chaque validation dans le dépôt lance une nouvelle build. Vous pouvez choisir d'inclure ou d'exclure des branches dans le processus CI.
5. Sélectionnez **Rétention**. En fonction de votre scénario, vous pouvez spécifier des stratégies pour conserver ou supprimer un certain nombre de builds.
6. En haut du pipeline de build, sélectionnez le nom du pipeline de build.
7. Remplacez le nom de votre pipeline de build par un nom plus descriptif, puis sélectionnez **Enregistrer** dans le menu déroulant **Enregistrer et mettre en file d'attente**.
8. Sous le nom de votre pipeline de build, sélectionnez **Historique**. Ce volet montre une piste d'audit des modifications que vous avez apportées récemment à la build. Azure DevOps suit les modifications apportées au pipeline de build et permet de comparer les versions.

## Examiner le pipeline de mise en production CD

DevOps Projects crée et configure automatiquement les étapes nécessaires au déploiement entre votre organisation Azure DevOps et votre abonnement Azure. Ces étapes comprennent la configuration d'une connexion au service Azure pour authentifier Azure DevOps auprès de votre abonnement Azure. L'automatisation crée également un pipeline de mise en production, qui fournit le déploiement continu vers Azure. Pour en savoir plus sur le pipeline de mise en production, suivez ces étapes :

1. Accédez à **Pipelines**, puis sélectionnez **Mises en production**.
2. Sélectionnez **Modifier**.

3. Sous **Artefacts**, sélectionnez **Déposer**. Le pipeline de build que vous avez examiné aux étapes précédentes produit la sortie qui est utilisée pour l'artefact.
4. À droite de l'icône **Déposer**, sélectionnez **Déclencheur de déploiement continu**. Ce pipeline de mise en production est doté d'un déclencheur de déploiement continu activé, qui exécute un déploiement chaque fois qu'un nouvel artefact de build est disponible. Vous pouvez désactiver le déclencheur afin que vos déploiements s'exécutent manuellement.
5. À droite, sélectionnez la section **Afficher les mises en production** pour voir l'historique des mises en production.
6. Sélectionnez la mise en production pour afficher le pipeline. Sélectionnez n'importe quel environnement pour vérifier le résumé, les validations de la mise en production et les éléments de travail associés.
7. Sélectionnez **Validations**. Cette vue montre les validations de code qui sont associées à ce déploiement. Comparez les mises en production pour afficher les différences de validation entre les déploiements.
8. Sélectionnez **Afficher les journaux**. Les journaux d'activité contiennent des informations utiles sur le processus de déploiement. Vous pouvez les voir à la fois pendant et après les déploiements.

## Valider les modifications de code et exécuter le pipeline CI/CD

### NOTE

La procédure suivante teste le pipeline CI/CD en apportant une modification de texte simple.

Vous êtes maintenant prêt à collaborer avec une équipe sur votre application, par le biais d'un processus d'intégration et de déploiement continus (CI/CD) qui déploie votre dernier travail sur votre App Service. Chaque modification apportée au dépôt Git démarre une build dans Azure DevOps tandis qu'un pipeline CD exécute un déploiement dans Azure. Suivez la procédure décrite dans cette section, ou utilisez une autre technique pour valider les modifications dans votre dépôt. Par exemple, vous pouvez cloner le dépôt Git dans votre outil ou environnement de développement intégré favori, puis envoyer (push) les modifications vers ce dépôt.

1. Dans le menu Azure DevOps, sélectionnez **Référentiels**, puis **Fichiers**. Accédez ensuite à votre référentiel.
2. Le référentiel contient déjà du code basé sur le langage de l'application que vous avez sélectionné lors du processus de création. Ouvrez le fichier **Application/views/index.pug**.
3. Sélectionnez **Modifier**, puis apportez une modification à la **ligne numéro 15**. Par exemple, vous pouvez la modifier avec le texte « Mon premier déploiement sur Azure App Service avec Azure Cosmos DB ».
4. En haut à droite, sélectionnez **Valider**, puis **Valider** à nouveau pour envoyer votre modification.

Au bout de quelques secondes, une build est démarrée dans Azure DevOps tandis qu'une mise en production est exécutée pour déployer les modifications. Supervisez l'état de la build dans le tableau de bord DevOps Projects, ou dans le navigateur avec votre organisation Azure DevOps.

## Supprimer des ressources

Vous pouvez supprimer les ressources associées que vous avez créées lorsque vous n'en avez plus besoin. Utilisez la fonctionnalité **Supprimer** du tableau de bord DevOps Projects.

## Étapes suivantes

Vous pouvez le modifier afin qu'il réponde aux besoins de votre équipe. Vous pouvez également utiliser ce modèle CI/CD comme modèle pour vos autres pipelines. Dans ce tutoriel, vous avez appris à :

- Utiliser DevOps Projects pour déployer une application Node.js alimentée par Azure Cosmos DB
- Configurer Azure DevOps et un abonnement Azure
- Examiner Azure Cosmos DB
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications dans Git et les déployer automatiquement dans Azure
- Supprimer des ressources

Consultez [Définir votre pipeline de déploiement continu \(CD\) multiétape](#) pour plus d'informations et pour connaître les étapes suivantes.

# Déployer en continu sur Azure Functions avec DevOps Projects

17/09/2019 • 13 minutes to read • [Edit Online](#)

Azure DevOps Projects offre une expérience simplifiée dans laquelle vous pouvez apporter vos code et dépôt Git existants ou choisir un exemple d'application pour créer un pipeline d'intégration continue (CI) et de livraison continue (CD) dans Azure.

De plus, DevOps Projects :

- Crée automatiquement des ressources Azure comme des fonctions Azure.
- Crée et configure un pipeline de mise en production dans Azure DevOps pour CI/CD.

Ce tutoriel présente les procédures suivantes :

- Utiliser DevOps Projects pour déployer une application ASP.NET sur Azure Functions
- Configurer Azure DevOps et un abonnement Azure
- Créer la fonction Azure
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications dans Git et les déployer automatiquement dans Azure
- Supprimer des ressources

Actuellement, les runtimes pris en charge pour les fonctions sont **.NET** et **Node.js**. Nous utilisons le runtime .NET dans le cadre de ce tutoriel pour le déploiement sur Azure Functions.

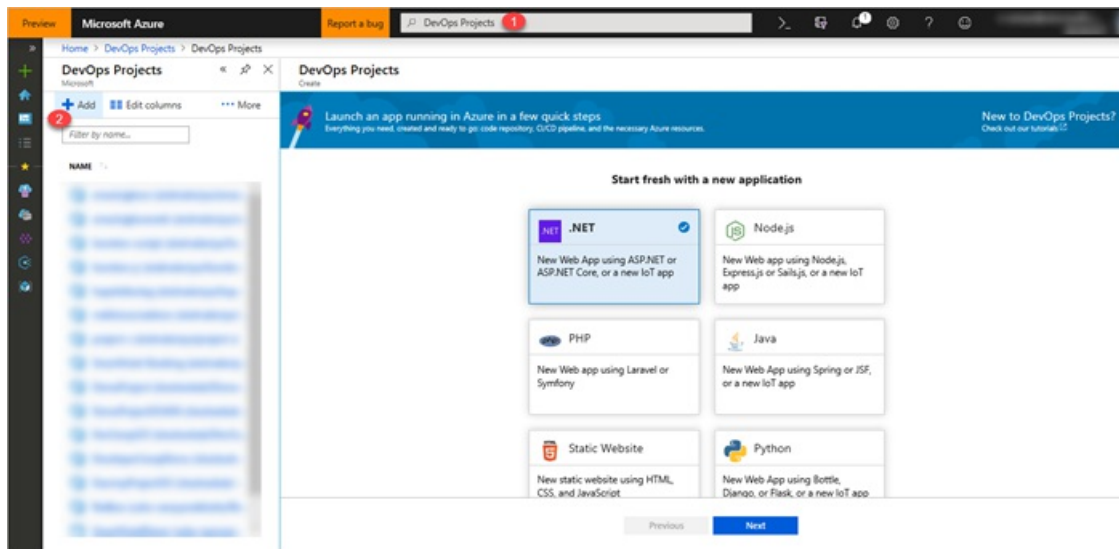
## Prérequis

- Un abonnement Azure. Vous pouvez en obtenir un gratuit par le biais de [Visual Studio Dev Essentials](#)

## Utiliser DevOps Projects pour déployer une application ASP.NET sur Azure Functions

DevOps Projects crée un pipeline CI/CD dans Azure Pipelines. Vous pouvez créer une organisation Azure DevOps ou utiliser une organisation existante. DevOps Projects crée également des ressources Azure, telles qu'un hub IoT, dans l'abonnement Azure de votre choix.

1. Connectez-vous au [portail Azure](#)
2. Dans le volet de gauche, sélectionnez **Créer une ressource**.
3. Dans la zone de recherche, tapez **DevOps Projects**, puis cliquez sur **Ajouter**.



4. Sélectionnez **.NET**, puis sélectionnez **Suivant**. Sous **Choisir un framework d'application**, sélectionnez **ASP.NET** et cliquez sur **Suivant**.
5. Sélectionnez **Application de fonction**, puis **Suivant**.

## Configurer Azure DevOps et l'abonnement Azure

1. Indiquez un nom pour votre projet Azure DevOps.
2. Créez une organisation Azure DevOps ou sélectionnez une organisation existante.
3. Sélectionnez votre abonnement Azure.
4. Pour voir d'autres paramètres de configuration Azure et pour identifier le niveau tarifaire et l'emplacement, cliquez sur Paramètres supplémentaires. Ce volet regroupe différentes options pour la configuration du niveau tarifaire et de l'emplacement des services Azure.
5. Quittez la zone de configuration Azure, puis sélectionnez Terminé.
6. Le processus est achevé au bout de quelques minutes. Un exemple d'application ASP.NET est configuré dans un dépôt Git de votre organisation Azure DevOps, une application de fonction, Application Insights est créé, un pipeline CI/CD est exécuté, et votre application est déployée sur Azure.

Une fois toutes ces opérations terminées, le tableau de bord Azure DevOps Projects s'affiche dans le portail Azure. Vous pouvez également accéder au tableau de bord DevOps Projects directement à partir de **Toutes les ressources** dans le portail Azure.

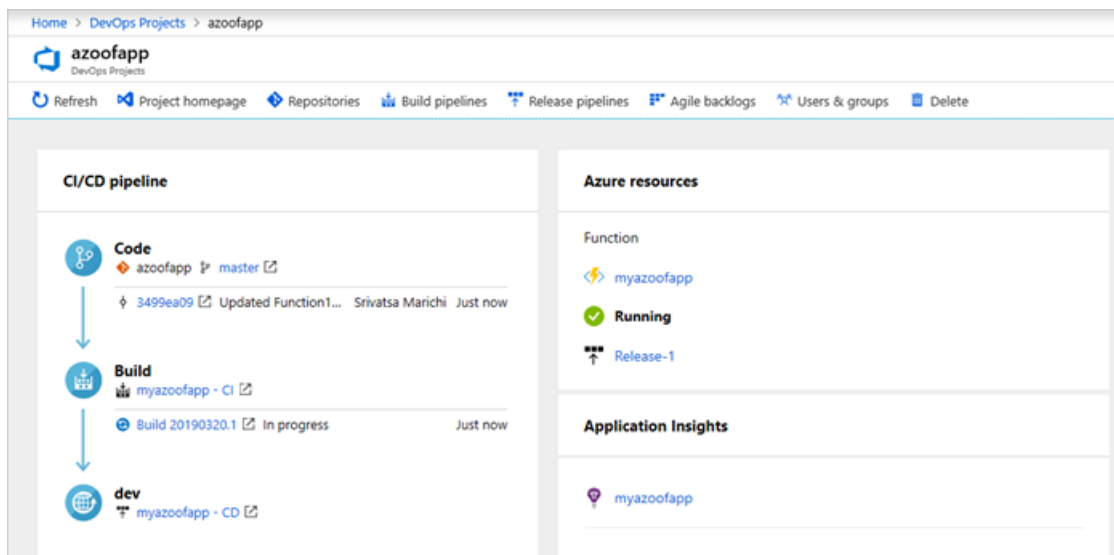
Ce tableau de bord permet de visualiser votre dépôt de code Azure DevOps, le pipeline CI/CD et votre fonction Azure. Vous pouvez configurer d'autres options de CI/CD dans votre pipeline Azure DevOps. Sur la droite, sélectionnez **Application de fonction** pour la voir.

## Examiner l'application de fonction

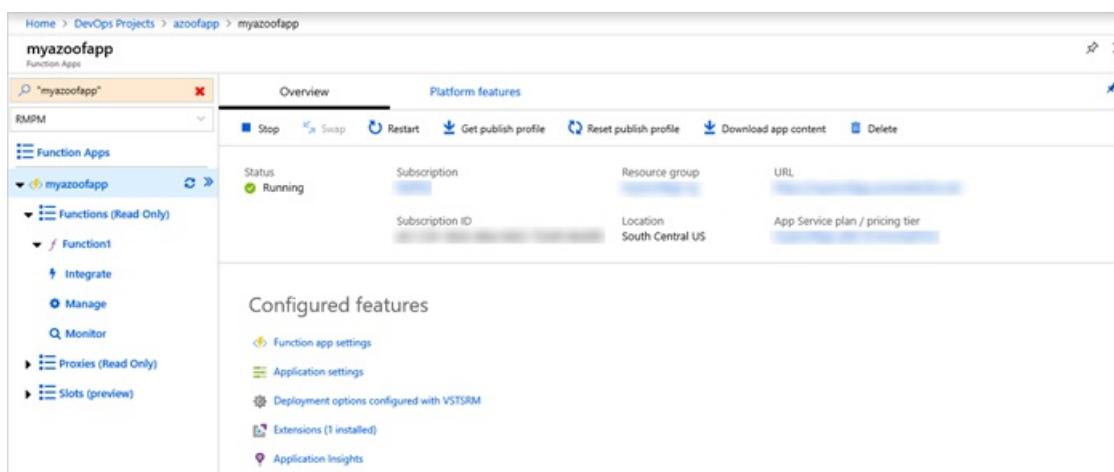
DevOps Projects configure automatiquement une application de fonction, que vous pouvez explorer et personnaliser. Pour vous familiariser avec l'application de fonction, effectuez les étapes suivantes :

1. Accédez au tableau de bord DevOps Projects.





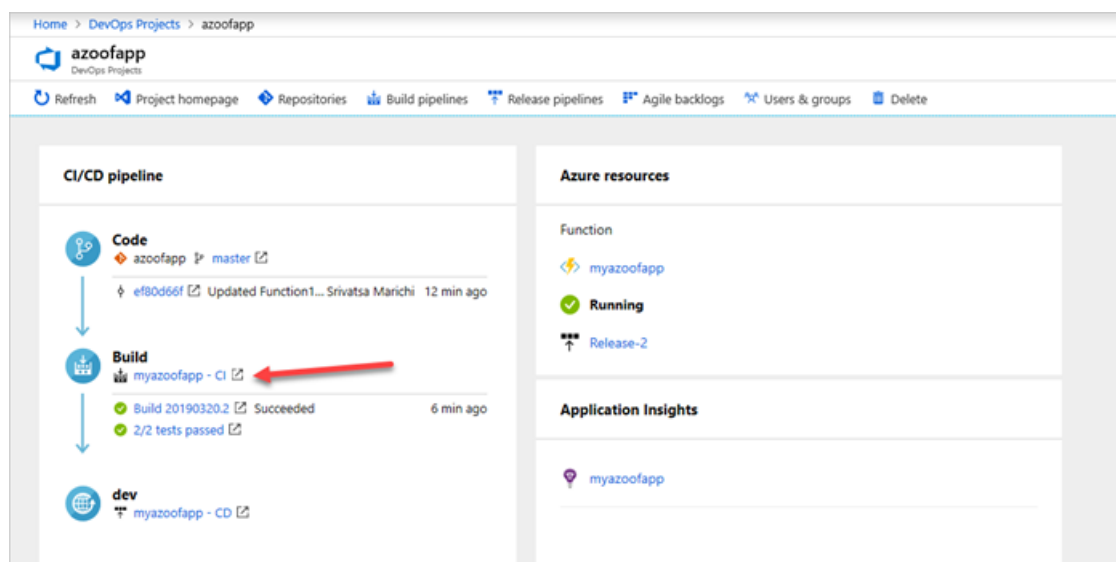
2. Sur la droite, sélectionnez l'application de fonction. Un volet s'ouvre pour l'application de fonction. À partir de cette vue, vous pouvez effectuer diverses actions, par exemple superviser les opérations et effectuer des recherches dans les journaux.



## Examiner le pipeline CI

DevOps Projects configure automatiquement un pipeline CI/CD dans votre organisation Azure DevOps. Vous pouvez explorer et personnaliser le pipeline. Pour vous familiariser avec celui-ci, procédez comme suit :

1. Accédez au tableau de bord DevOps Projects.
2. Cliquez sur le lien hypertexte sous **Build**. Un onglet du navigateur affiche le pipeline de build de votre nouveau projet.



3. Sélectionnez **Modifier**. Dans ce volet, vous pouvez examiner les différentes tâches de votre pipeline de build. La build effectue différentes tâches, telles que l'extraction du code source à partir du dépôt git, la génération de l'application, l'exécution de tests unitaires et la publication des sorties utilisées pour les déploiements.
4. Sélectionnez **Déclencheurs**. DevOps Projects crée automatiquement un déclencheur CI, et chaque validation dans le dépôt lance une nouvelle build. Si vous le souhaitez, vous pouvez choisir d'inclure ou d'exclure des branches dans le processus CI.
5. Sélectionnez **Rétention**. En fonction de votre scénario, vous pouvez spécifier des stratégies pour conserver ou supprimer un certain nombre de builds.
6. En haut du pipeline de build, sélectionnez le nom du pipeline de build.
7. Remplacez le nom de votre pipeline de build par un nom plus descriptif, puis sélectionnez **Enregistrer** dans le menu déroulant **Enregistrer et mettre en file d'attente**.
8. Sous le nom de votre pipeline de build, sélectionnez **Historique**. Ce volet montre une piste d'audit des modifications que vous avez apportées récemment à la build. Azure DevOps suit les modifications apportées au pipeline de build et permet de comparer les versions.

## Examiner le pipeline de mise en production CD

DevOps Projects crée et configure automatiquement les étapes nécessaires au déploiement entre votre organisation Azure DevOps et votre abonnement Azure. Ces étapes comprennent la configuration d'une connexion au service Azure pour authentifier Azure DevOps auprès de votre abonnement Azure. L'automatisation crée également un pipeline de mise en production, qui fournit le déploiement continu vers Azure. Pour en savoir plus sur le pipeline de mise en production, suivez ces étapes :

1. Accédez à **Pipelines | Mises en production**.
2. Cliquez sur **Modifier**.
3. Sous **Artefacts**, sélectionnez **Déposer**. Le pipeline de build que vous avez examiné aux étapes précédentes produit la sortie qui est utilisée pour l'artefact.
4. À droite de l'icône **Déposer**, sélectionnez **Déclencheur de déploiement continu**. Ce pipeline de mise en production est doté d'un déclencheur de déploiement continu activé qui exécute un déploiement chaque fois qu'un nouvel artefact de build est disponible. Si vous le souhaitez, vous pouvez désactiver le déclencheur afin que vos déploiements nécessitent une exécution manuelle.
5. À droite, sélectionnez **Afficher les mises en production** pour voir l'historique des mises en production.

6. Cliquez sur la mise en production pour afficher le pipeline. Cliquez sur n'importe quel environnement pour vérifier **la Synthèse, les Validations** de la mise en production et les **Éléments de travail associés**.
7. Sélectionnez **Validations**. Cette vue montre les validations de code qui sont associées à ce déploiement. Comparez les mises en production pour afficher les différences de validation entre les déploiements.
8. Sélectionnez **Afficher les journaux**. Les journaux d'activité contiennent des informations utiles sur le processus de déploiement. Vous pouvez les voir à la fois pendant et après les déploiements.

## Valider les modifications de code et exécuter CI/CD

### NOTE

La procédure suivante teste le pipeline CI/CD en apportant une modification de texte simple.

Vous êtes maintenant prêt à collaborer avec une équipe sur votre application, par le biais d'un processus d'intégration et de livraison continus (CI/CD) qui déploie automatiquement votre dernier travail sur votre fonction Azure. Chaque modification apportée au dépôt Git démarre une build dans Azure DevOps tandis qu'un pipeline CD exécute un déploiement dans Azure. Suivez la procédure décrite dans cette section, ou utilisez une autre technique pour valider les modifications dans votre dépôt. Par exemple, vous pouvez cloner le dépôt Git dans votre outil ou environnement de développement intégré favori, puis envoyer (push) les modifications vers ce dépôt.

1. Dans le menu Azure DevOps, sélectionnez **Dépôts | Fichiers**, puis accédez à votre dépôt.
2. Le dépôt contient déjà du code appelé **SampleFunctionApp** basé sur le langage de l'application que vous avez sélectionné lors du processus de création. Ouvrez le fichier **application/SampleFunctionApp/fonction1.cs**.
3. Sélectionnez **Modifier**, puis apportez une modification à la **ligne numéro 31**. Par exemple, vous pouvez la mettre pour qu'elle indique **Bonjour. Bienvenue dans Azure Functions à l'aide de DevOps Projects**.
4. En haut à droite, sélectionnez **Valider**, puis **Valider** à nouveau pour transmettre votre modification.
5. Ouvrez le fichier **Application/SampleFunctionApp.Test/Function1TestRunner.cs**.
6. Sélectionnez **Modifier**, puis apportez une modification à la **ligne numéro 21**. Par exemple, vous pouvez la mettre à jour pour qu'elle indique **Bonjour. Bienvenue dans Azure Functions via Azure DevOps Projects**.

Au bout de quelques instants, une build est démarrée dans Azure DevOps tandis qu'une mise en production est exécutée pour déployer les modifications. Supervisez l'état de la build dans le tableau de bord DevOps Projects, ou dans le navigateur avec votre organisation Azure DevOps.

## Supprimer des ressources

Quand vous n'en avez plus besoin, vous pouvez supprimer les ressources associées que vous avez créées. Utilisez la fonctionnalité **Supprimer** du tableau de bord DevOps Projects.

## Étapes suivantes

Si vous le souhaitez, vous pouvez modifier ces pipelines de build et de mise en production afin qu'ils répondent aux besoins de votre équipe. Vous pouvez également utiliser ce modèle CI/CD comme modèle pour vos autres pipelines. Dans ce tutoriel, vous avez découvert comment effectuer les tâches suivantes :

- Utiliser DevOps Projects pour déployer une application ASP.NET Core sur Azure Functions
- Configurer Azure DevOps et un abonnement Azure

- Examiner la fonction Azure
- Examiner le pipeline CI
- Examiner le pipeline CD
- Valider les modifications dans Git et les déployer automatiquement dans Azure
- Supprimer des ressources