Module 15: Déploiement d'applications Web ASP.NET MVC 5

# **Contenu:**

|  |  |
| --- | --- |
|  | [Aperçu du module](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C15.html#P1) |
| **Leçon 1:** | [Déployer une application Web](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C15.html#P2) |
| **Leçon 2:** | [Déploiement d'une application Web ASP.NET MVC 5](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C15.html#P3) |
| **Laboratoire:** | [Déploiement d'applications Web ASP.NET MVC 5](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C15.html#P4) |
|  | [Examen du module et points à retenir](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C15.html#P5) |

# **Aperçu du module**

Après avoir développé une application, vous devez déployer l'application dans l'environnement de production pour la mettre à la disposition des utilisateurs. La méthode de configuration ou de déploiement de chaque application varie en fonction de la configuration de l'environnement de production. Les applications ASP.NET MVC ont des considérations spéciales telles que la configuration nécessaire pour IIS.

### **Objectifs**

Après avoir terminé ce module, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Configurez et déployez une application Web ASP.NET. |
| • | Déployez une application Web ASP.NET MVC 5. |

# Leçon 1: Déployer une application Web

Les considérations relatives au déploiement d'une application Web sur une batterie de serveurs Web sont différentes pour les configurations à un ou plusieurs serveurs. Par exemple, dans les batteries de serveurs Web multi-serveurs, vous devez vous assurer que les informations d'état sont disponibles sur tous les serveurs Web. Vous devez également vous assurer que les dépendances nécessaires à l'hébergement des applications Web ASP.NET MVC 5 sont présentes dans la batterie de serveurs Web. Si l'une des dépendances d'application manque sur les serveurs Web de production, certaines ou toutes vos applications peuvent ne pas fonctionner et les utilisateurs peuvent rencontrer des erreurs. Vous devez appliquer des considérations spéciales lors du déploiement d'applications Web sur Microsoft Azure. Le déploiement d'une application Web sur Microsoft Azure offre des avantages tels que la haute disponibilité et la flexibilité.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Assurez-vous que les dépendances sont présentes pour l'hébergement des applications ASP.NET MVC4. |
| • | Déployez une application Web ASP.NET MVC 5 sur une batterie de serveurs Web à serveur unique. |
| • | Déployez une application Web ASP.NET MVC 5 sur une batterie de serveurs Web multi-serveurs. |
| • | Déployez une application Web ASP.NET MVC 5 sur Microsoft Azure. |

## **Dépendances d'ASP.NET MVC 5**

ASP.NET MVC 5 repose sur ASP.NET Framework. Chaque application Web nécessite la présence d'une gamme de composants différents sur le serveur Web pour fonctionner correctement. Voici des exemples d'exigences communes:

|  |  |
| --- | --- |
| • | Le Framework ASP.NETCommon Language Runtime (CLR). Le CLR exécute tout code managé tel que les classes C #. Toutes les applications Web ASP.NET nécessitent le CLR et vous pouvez installer le CLR avec Internet Information Services (IIS). |
| • | ASP.NET MVC 5Durée. Le runtime ASP.NET MVC 5 localise les contrôleurs et les actions pour gérer chaque demande reçue par l'application Web et renvoie les pages Web compilées aux utilisateurs. Vous pouvez installer le runtime MVC dans votre application Web en installant le package MVC 5 NuGet. Toutes les applications Web MVC nécessitent le runtime MVC. |
| • | Un serveur de base de données.La plupart des applications Web utilisent une base de données pour stocker des informations telles que les détails du produit, les détails du client, des images et d'autres entités. Vous devez vous assurer que l'environnement de production prend en charge la base de données que vous avez utilisée pendant le développement. Par exemple, si vous utilisez SQL Compact pendant le développement, vous devez vous assurer que Microsoft SQL Compact est installé sur le serveur de production. Vous pouvez également migrer la base de données vers Microsoft SQL Server. |
| • | Cadre d'entité.Si vous utilisez Entity Framework pour modéliser des données dans votre application Web, vous devez vous assurer que vous déployez l'infrastructure avec votre application. |
| • | Fournisseurs d'adhésion.Si votre application utilise des fournisseurs d'appartenance pour stocker les comptes utilisateur et les rôles, vous devez vous assurer que ces fournisseurs sont disponibles sur le serveur. Le .NET Framework inclut le fournisseur d'appartenance à SQL Server. Si vous utilisez les fournisseurs d'appartenance universelle en installant le package NuGet, vous devez vous assurer que vous incluez le package NuGet avec votre application déployée. |

**Question:**Laquelle des exigences courantes de la liste ci-dessus est requise pour prendre en charge l'authentification par formulaire?

## **Déploiement d'applications Web sur des serveurs Web**

Avant de déployer une application Web sur un serveur Web IIS de production, vous devez créer et configurer l'application IIS et le dossier qui héberge l'application.

**Création d'une application Web IIS**

Vous devez héberger les applications Web ASP.NET MVC 5 dans un dossier Web IIS marqué comme point de départ de l'application Web. Vous pouvez utiliser l'outil Gestionnaire des services Internet pour configurer ce paramètre dans le dossier de niveau supérieur de l'application Web. Lorsque vous marquez un dossier Web en tant qu'application Web, vous activez IIS pour reconnaître que le contenu du dossier est ASP.NET et nécessite le moteur d'exécution ASP.NET pour traiter chaque demande.

Après avoir créé une application, vous devez configurer l'application pour utiliser .NET 4.0 ou supérieur. Cette pratique permet à ASP.NET MVC 5 de s'exécuter, car les composants d'exécution MVC font partie du framework .NET.

**Pools d'applications IIS**

Dans IIS, les applications Web sont exécutées dans un contexte appelé pool d'applications. Vous pouvez exécuter de nombreuses applications dans le même pool d'applications ou installer une application dans son propre pool isolé. Les applications qui s'exécutent dans le même pool d'applications partagent les ressources suivantes:

|  |  |
| --- | --- |
| • | Processus des travailleurs.Le processus de travail utilise un processus d'exécution unique pour exécuter toutes les applications d'un pool. |
| • | Mémoire. Les applications partagent une seule plage de mémoire, car les applications d'un pool partagent un seul processus de travail. |
| • | Identité.Les applications d'un même pool partagent un compte utilisateur. |

Vous pouvez placer deux ou plusieurs applications dans un pool d'applications et réduire ainsi l'utilisation totale de la mémoire, car les applications partagent la mémoire. Cependant, cette disposition peut réduire la fiabilité de chaque application, car une application défectueuse peut affecter toutes les applications du pool. Pour maximiser la fiabilité d'une application, vous devez installer l'application dans un pool isolé qui n'exécute qu'une seule application spécifique.

L'identité partagée au sein d'un pool d'applications affecte la façon dont les applications authentifient les ressources telles que les serveurs de base de données lors de l'utilisation de l'authentification intégrée. Chaque application d'un pool partage un seul compte utilisateur. Si chaque application possède une base de données distincte, vous devez définir des autorisations pour permettre aux comptes d'utilisateurs d'accéder à toutes les bases de données. Si vous séparez les applications en pools isolés, vous pouvez utiliser un compte d'utilisateur différent pour chaque application. Cette pratique peut améliorer la sécurité de l'application, car chaque compte ne peut accéder qu'à une seule base de données.

**Déployer l'application**

Lorsque l'application Web IIS est prête, vous devez copier les fichiers de l'application Web dans IIS. Ces fichiers incluent toutes les classes de modèle, contrôleurs, vues, fichiers de script, feuilles de style, images et tout autre contenu du projet Microsoft Visual Studio ASP.NET MVC 5. Vous pouvez utiliser les outils de déploiement disponibles dans Microsoft Visual Studio, ou d'autres technologies telles que FTP, pour copier ces fichiers. Une fois l'opération de copie terminée, l'application Web est prête pour le déploiement.

**Question:**À quoi sert la configuration de pools d'applications supplémentaires?

## **Déploiement d'applications Web sur des fermes multi-serveurs**

Vous pouvez déployer une application Web sur une batterie de serveurs Web multi-serveurs pour augmenter les performances, la résilience et la fiabilité. Dans une batterie de serveurs Web multi-serveurs, un groupe d'au moins deux serveurs Web héberge une seule application Web. Tous les serveurs de la batterie partagent un seul nom de serveur, tel que www.contoso.com. L'équilibrage de la charge réseau Windows ou des équilibreurs de charge matériels dédiés distribuent les requêtes du navigateur aux serveurs.

Une batterie de serveurs multi-serveurs a généralement une capacité supérieure à un serveur Web unique, car la batterie de serveurs multi-serveurs partage presque également la charge entre plusieurs serveurs. Les fermes multi-serveurs augmentent également la fiabilité. Lorsqu'un serveur tombe en panne pour une raison quelconque, le mécanisme d'équilibrage de charge dirige automatiquement les demandes vers un autre serveur qui reste en ligne.

Si vous décidez de déployer votre application Web sur une batterie de serveurs multi-serveurs, vous devez effectuer les étapes suivantes. Pour effectuer ces étapes, vous devrez peut-être travailler avec l'administrateur du serveur Web responsable de la batterie de serveurs:

|  |  |
| --- | --- |
| 1. | Créez des applications IIS et des pools d'applications sur chaque serveur. Cette étape est identique à la configuration des applications et des pools d'applications dans un scénario à serveur unique. Cependant, vous devez effectuer cette étape sur chaque serveur de la batterie. Vous devez également vous assurer que les applications et les pools d'applications ont la même configuration, dans toute la batterie. |
| 2. | Créez une configuration IIS correspondante sur chaque serveur. Les options de configuration IIS, telles que les certificats de chiffrement, les extensions de fichier et les composants facultatifs, doivent être identiques sur chaque serveur de la batterie. |
| 3. | Utilisez l'état de session hébergé externe ou l'affinité de session. Vous pouvez configurer les applications Web ASP.NET pour stocker l'état de session dans le service d'état Windows ou une base de données. Vous devez utiliser cette technique dans une batterie de serveurs multi-serveurs, car chaque requête d'une session utilisateur unique peut être dirigée vers un serveur Web différent. Sinon, en configurant l'affinité de session, vous pouvez vous assurer que toutes les demandes d'une session utilisateur sont toujours envoyées au même serveur Web. Si vous n'utilisez pas l'état de session hébergée externe ou l'affinité de session, la préférence utilisateur sur un serveur Web peut être perdue lorsqu'une demande est envoyée à un autre serveur Web. L'affinité de session est prise en charge par certaines solutions d'équilibrage de charge, mais pas toutes. Collaborez avec votre administrateur de serveur Web pour déterminer si l'affinité est disponible sur votre batterie de serveurs. |
| 4. | Configurez l'élément machineKey dans le fichier Web.config. Si vous utilisez un état de session hébergé externe, vous pouvez crypter la connexion entre les serveurs Web et le serveur State Service ou le serveur de base de données. Cette technique améliore la sécurité en protégeant l'état de session pendant les transmissions réseau. Vous pouvez configurer ce chiffrement en configurant l'élément machineKey dans le registre Windows. |

**Lecture supplémentaire:**Pour plus d'informations sur la configuration de l'élément machineKey, accédez à:<http://go.microsoft.com/fwlink/?LinkID=288995&clcid=0x424>

**Question:**À quoi sert la configuration de l'élément machineKey dans le fichier Web.config?

## **Déployer des applications Web sur Microsoft Azure**

Vous pouvez déployer n'importe quelle application Web ASP.NET, y compris les applications Web ASP.NET MVC 5, sur Microsoft Azure. En choisissant Microsoft Azure comme plate-forme, vous pouvez utiliser une infrastructure hautement disponible et flexible que vous pouvez optimiser pour faire face à une charge intense. Vous avez le choix entre les abonnements, qui peuvent vous aider à ne payer que le trafic utilisé par votre application Web.

Le déploiement d'une application Web sur Microsoft Azure est différent du déploiement de l'application sur des versions locales d'IIS. Pour déployer une application sur Microsoft Azure, vous effectuez les étapes suivantes: To deploy an application on Microsoft Azure, you perform the following steps:

**Publication sur une nouvelle application Web:**

|  |  |
| --- | --- |
| 1. | Dans Microsoft Visual Studio, ouvrez votre solution d'application Web. Ensuite, démarrez l'assistant de publication et choisissez Azure Web App. |
| 2. | Dans l'assistant de publication, attribuez un nom unique à la nouvelle application Web. |
| 3. | Terminez l'assistant de publication. |
| 4. | Observez que Microsoft Visual Studio publie l'application Web sur Microsoft Azure. |

**Publication sur une application Web existante**

|  |  |
| --- | --- |
| 1. | Créez une nouvelle application Web dans le portail de gestion Microsoft Azure. |
| 2. | Dans Microsoft Visual Studio, ouvrez votre solution d'application Web. Ensuite, démarrez l'assistant de publication. |
| 3. | Sélectionnez Azure Web App et assurez-vous de sélectionner «Utiliser l'existant» |
| 4. | Assurez-vous de sélectionner les chaînes de connexion appropriées pour les connexions Entity Framework, les connexions à la base de données de service et toutes les autres connexions à la base de données. |
| 5. | Terminez l'assistant de publication. |
| 6. | Observez que Microsoft Visual Studio publie l'application Web sur Microsoft Azure. |

Une fois le processus terminé, vous pouvez accéder à l'application Web hébergée par Microsoft Azure à partir de n'importe quel navigateur connecté à Internet.

**Noter:**Au moment de la rédaction de cet article, les applications Web Microsoft Azure sont toujours en préversion, au sein de la plateforme Microsoft Azure.

## **Démonstration: Comment créer une application Web Microsoft Azure**

Dans cette démonstration, vous verrez comment créer une nouvelle base de données et une nouvelle application Web dans Microsoft Azure.

### **Étapes de démonstration**

|  |  |
| --- | --- |
| • | Vous trouverez les étapes dans la section «Leçon 1: Déployer une application Web» sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD15_DEMO.md>. |

# Leçon 2: Déploiement d'une application Web ASP.NET MVC 5

Le déploiement d'applications ASP.NET MVC est similaire au déploiement d'autres applications ASP.NET. Ce processus est simple et direct. Cependant, vous devez mettre à jour la configuration de l'application pour une utilisation dans l'environnement de production. Microsoft Visual Studio inclut des fonctionnalités qui simplifient la mise à jour de la configuration de l'application. Vous pouvez également automatiser le processus de déploiement à l'aide des outils de déploiement disponibles dans Microsoft Visual Studio.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Reconfigurez une application ASP.NET MVC pour la déployer sur un serveur Web de production. |
| • | Utilisez la méthode Bin Deploy pour déployer les bibliothèques .NET Framework avec votre application Web. |
| • | Utilisez les outils de déploiement dans Visual Studio. |

## **Vérification de la configuration pour la production**

Chaque application Web ASP.NET MVC comprend un fichier Web.config. En plus de ce fichier, Microsoft Visual Studio crée deux fichiers de variance pour spécifier les transformations de déploiement dans le fichier Web.config. Vous pouvez utiliser ces fichiers de variance pour différentes configurations de construction. Les fichiers de variance comprennent:

|  |  |
| --- | --- |
| • | La toileFichier .release.config. Ce fichier stocke les modifications que Microsoft Visual Studio applique au fichier Web.config lorsque vous compilez l'application en mode Release. |
| • | La toileFichier .debug.config. Ce fichier stocke les modifications que Microsoft Visual Studio applique au fichier Web.config, lorsque vous compilez l'application en mode débogage. |

Microsoft Visual Studio peut automatiquement transformer le fichier Web.config pour une utilisation dans différents environnements. Avant de publier l'application, vous pouvez supprimer l'attribut de débogage du fichier Web.config.

**Suppression de l'attribut de débogage pour le déploiement**

<system.web>

<compilation xdt: Transform = "RemoveAttributes (debug)" />

</system.web>

**Suppression de l'attribut de débogage à l'aide d'une transformation de déploiement**

<system.web>

<compilation xdt: Transform = "RemoveAttributes (debug)" />

</system.web>

L'exemple de code précédent utilise l'attribut xdt: Transform pour modifier le fichier Web.config existant et produire le fichier Web.config publié. Vous pouvez également utiliser la valeur Insérer pour ajouter des éléments supplémentaires au fichier Web.config résultant.

**Insertion d'une chaîne de connexion**

<connectionStrings>

<add name = "DemoConnStr" connectionString = "Data Source = | DataDirectory | demo.sdf"

providerName = "System.Data.SqlServerCe.4.0" xdt: Transform = "Insérer" />

</connectionStrings>

**Question:**Comment configurer le fichier Web.config pour la publication dans un environnement de production sans utiliser le fichier de configuration Web.release.config?

## **Utilisation de Bin Deploy**

Les applications ASP.NET MVC dépendent d'une plage d'assemblys .NET; ces assemblys garantissent que l'application fonctionne selon les besoins. Ces assemblys sont des fichiers de bibliothèque de liens dynamiques (DLL) disponibles dans les références de votre projet. Les fichiers d'assemblage sont généralement déjà présents sur le serveur Web car ils sont inclus dans le cadre de l'installation de la bibliothèque MVC. Toutefois, les administrateurs de serveur Web n'installent parfois pas tous les assemblys .NET Framework. Microsoft Visual Studio fournit la fonctionnalité Bin Deploy pour copier tous les fichiers d'assemblage dépendants dans un dossier de l'application Web déployée. Cette fonctionnalité permet de déployer les dépendances sur le serveur.

Vous pouvez utiliser la fonctionnalité Ajouter des dépendances déployables pour générer le dossier \_bin\_deployableAssemblies. Ce dossier comprend les bibliothèques suivantes:

|  |  |
| --- | --- |
| • | Microsoft.Web.Infrastructure |
| • | System.Web.Helpers |
| • | System.Web.Mvc |
| • | System.Web.Razor |
| • | System.Web.WebPages |
| • | System.Web.WebPages.Deployment |
| • | System.Web.WebPages.Razor |

Vous pouvez choisir un ensemble spécifique de bibliothèques à inclure dans Microsoft Visual Studio. Cette pratique vous permet de vous assurer que Microsoft Visual Studio déploie:

|  |  |
| --- | --- |
| • | Bons assemblages. |
| • | Bonnes versions des assemblages. |

Vous pouvez ensuite copier la bibliothèque dans le dossier bin du serveur de production, après l'avoir ajoutée à la bibliothèque de votre projet.

**Question:**Pourquoi devriez-vous utiliser l'assembly déployable dans le cadre du plan de déploiement?

## **Utilisation des outils de déploiement de Visual Studio 2017**

La fonction de publication permet de générer une copie de l'application Web lorsque l'application est prête à être déployée dans l'environnement de production. La fonctionnalité comprend trois méthodes pour déployer l'application:

|  |  |
| --- | --- |
| • | Partage de fichiers |
| • | FTP |
| • | Déploiement Web |

Sur la page Publier l'application Web, vous pouvez publier l'application à l'aide de Partage de fichiers ou FTP. Vous pouvez utiliser le partage de fichiers pour les serveurs auxquels vous avez un accès direct au réseau. Vous pouvez également utiliser le partage de fichiers pour publier d'abord une application Web dans un dossier local. Ensuite, vous pouvez copier manuellement les fichiers sur le serveur Web. Le partage de fichiers vous permet également de spécifier un profil pour stocker toutes les options de déploiement. Cette pratique élimine le besoin de spécifier les options de déploiement chaque fois que vous publiez l'application.

Si vous sélectionnez Déploiement Web, vous pouvez spécifier des paramètres de base de données, tels que des chaînes de connexion, dans l'assistant de publication. Ces paramètres remplaceront les paramètres du fichier Web.config dans l'environnement cible. Web Deploy inclut également des outils pour mettre à jour automatiquement le schéma de la base de données. Ces outils éliminent le besoin de mettre à jour manuellement le schéma. L'outil de publication détecte toutes les modifications de schéma et génère des scripts pour appliquer ces modifications à la base de données publiée. Si vos serveurs Web de développement et de production sont isolés les uns des autres, vous pouvez exécuter ces scripts générés sur l'environnement de production. Cette pratique réplique les modifications de schéma sur un serveur sur lequel Microsoft Visual Studio n'est pas installé.

L'image suivante montre la page de publication de l'application Web.

**FIGURE 15.1: LA PAGE D'APPLICATION WEB PUBLIÉE**

**Question:**Quel est l'avantage d'utiliser l'outil de publication Web Deploy?

## **Démonstration: comment déployer un site Web sur Microsoft Azure**

Dans cette démonstration, vous verrez comment obtenir un profil de publication à partir de Microsoft Azure et l'utiliser pour publier un site Web à partir de Visual Studio.

### **Étapes de démonstration**

|  |  |
| --- | --- |
| • | Vous trouverez les étapes de la section «Leçon 2: Déploiement d'une application Web ASP.NET MVC 5» sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD15_DEMO.md>. |

# **Atelier: Déploiement d'applications Web ASP.NET MVC 5**

### **Scénario**

Vous avez terminé le développement et les tests de l'application de partage de photos. Vos responsables et développeurs seniors ont approuvé le projet et vous ont demandé de déployer l'application sur le compte Microsoft Azure Adventure Works.

### **Objectifs**

Après avoir terminé cet atelier, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Préparez une application Web MVC pour le déploiement. |
| • | Déployez une application Web MVC sur Microsoft Azure. |

##### **Configuration du laboratoire**

Durée estimée: 45 minutes

Vous trouverez les étapes de haut niveau sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD15_LAB_MANUAL.md>.

Vous trouverez les étapes détaillées sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD15_LAK.md>.

### **Exercice 1: déploiement d'une application Web sur Microsoft Azure**

##### **Scénario**

Dans cet exercice, vous allez:

|  |  |
| --- | --- |
| • | Reconfigurez l'application de partage de photos pour le déploiement de la version. |
| • | Configurez la classe d'initialisation Entity Framework, qui remplit la base de données avec les données initiales, et assurez-vous que la configuration de génération et les chaînes de connexion sont correctes. |
| • | Créez une nouvelle application Web dans Microsoft Azure et déployez l'application de partage de photos sur le nouveau site. |

### **Exercice 2: Test de la demande terminée**

##### **Scénario**

Vous avez terminé et entièrement déployé l'application Web de partage de photos dans Microsoft Azure. Maintenant, vous souhaitez effectuer quelques tests de fonctionnalité finaux avant de confirmer l'achèvement de l'application à votre responsable.

**Vérifiez vos connaissances**

**Découverte**

**Pourquoi est-il inutile d'utiliser le déploiement de bac dans ce laboratoire?**

Afficher la solution Réinitialiser

**Vérifiez vos connaissances**

**Découverte**

**Dans les ateliers de ce cours, vous avez utilisé la même base de données SQL Microsoft Azure pour le développement et la production. Si vous vouliez utiliser des bases de données distinctes pour le développement et la production, mais que vous ne vouliez pas reconfigurer l'application Web à chaque fois que vous déployez sur les serveurs Web de développement et de production, comment configureriez-vous l'application Web?**

Afficher la solution Réinitialiser

# **Examen du module et points à retenir**

Le déploiement est généralement la dernière tâche exécutée par les développeurs; cependant, c'est la tâche à laquelle les développeurs ne passent pas beaucoup de temps. L'environnement cible a généralement un impact sur la procédure de déploiement. Microsoft Web Apps fournit un environnement convivial pour les développeurs dans lequel le développeur peut laisser la gestion de la plate-forme à Microsoft Azure et se concentrer sur le développement d'applications.

Visual Studio 2017 dispose d'un outil de déploiement simple mais puissant qui permet au développeur de déployer rapidement et en toute sécurité des applications sur Microsoft Azure Web Apps.

### **Question (s) de révision**