***Гольдін М.І.и Ярослав***

1. **Windows Asure для Java додатків. Огляд можливостей та технологій.**

Windows Azure SDK для Java надає клієнтські бібліотеки та інструменти для Java, дозволяє розробникам швидко і легко створювати додатки, що працюють на Windows Azure або важелі Windows Azure послуг, такі як Windows Azure Storage або Service Bus. Така сторінка збирає інформацію про деякі з ресурсів, які доступні для розробників Java.

Microsoft розглядає мову програмування Java як першокласну на своїй платформі Windows Azure. У цілому користувачі згодні з цією заявою, бо вже існуючі пропозиції для розміщення Java-додатків в хмарі Microsoft дозволяють розмістити будь-який додаток. Розгляд "за" і "проти" розміщення Java-додатки в Windows Azure.

Чому взагалі необхідна хмара для потреб Java-додатків? Вже розроблені програми можуть бути досить складними і мати розподілену архітектуру, яка використовує безліч серверів, відповідно, те, що пропонує хмарна платформа, підходить для цих потреб і, можливо, обіцяє економічні вигоди.

Зазвичай веб-додатки на Java пишуться з використанням сервлетів, JSP, JSF і так далі. На потужностях Tomcat, JBoss, GlassFish. Радісно бачити, що навіть в плагіні Windows Azure для Eclipse, є конфігураційні файли, підготовлені окремо для кожного з цих серверів додатків і контейнерів. Крім цього, у Microsoft вже заготовлені корисні готові проекти з мануалами, так звані Solution Accelerator - Windows Azure Tomcat Solution Accelerator і Windows Azure Jetty Solution Accelerator.

Поширене питання про збереження сесій і липкі сесії актуальниі і з Java-додатками. Іншим підходом є використання Atomus Tomcat Azure Session Manager (зберігаючого сесію в сховище таблиць Windows Azure) і Windows Azure Sticky HTTP сесій маршрутизатора. Розробнику дозволяється використовувати модуль додатків IIS маршрутизації запитів.

**Що з сховищем в Java-додатку?**

Де зберігати дані, якщо є Java-додаток? Є кілька варіантів - це SQL Azure, реляційна СУБД на основі SQL Server 2008 R2 в якій є більшість функціональності (і, все ж, не вся). Нещодавно драйвер JDBC від Microsoft був вбудований в 3,0 версію і став остаточно офіційно підтримувати SQL Azure. В цілому, робота з драйвером JDBC, SQL Azure і Java-додатком досить приємна і нескладна. Якщо ж є вже готова база даних та її потрібно перенести в хмару, у Microsoft є для цього засіб - SQL Azure Migration Wizard. Дозволяє переносити готову базу даних в хмару, будь вона Oracle, MySQL або Access.

Другим варіантом розміщення даних є сервіси сховища Windows Azure. Windows Azure SDK для Java повністю підтримує взаємодію сховища Windows Azure і Java, тому й тут не виникне жодних проблем. Нагадування: сервіси сховищ складаються з трьох основних сервісів - Блоб (дешеве і масштабоване сховище великих документів, яке може використовувати Windows Azure Content Delivery мережа для доставки контенту ближче до користувачів), таблиці (масштабоване і дешеве зберігання нереляційних даних, аналог реляційних баз даних, але не дозволяє зберігати скільки-небудь зв'язаних даних) і черги (сховище повідомлень для здійснення комунікації між ролями).

Таким чином, якщо ви не використовуєте додаткову функціональність платформи, ви можете легко розмістити ваш додаток і управляти його масштабованістю.

**До недоліків необхідно віднести:**

**Максимальний розмір бази даних в SQL Azure - 150 гігабайт.** Для певної частини додатків такого обсягу буде достатньо, але, якщо його стане мало, доведеться придумувати рішення щодо масштабування. У SQL Azure вже є федерації, можна реалізовувати шардінг - в цілому, недоліком це є тільки тому, що доведеться робити це самому, а не дати на відкуп платформі.

**Відсутність підтримки для Java сервісу кешування.**

**Кількість зовнішніх портів-кінцевих точок входу для кожного екземпляра.** Зараз одно 25, раніше було всього 5 і це було дійсно погано, враховуючи використання деякими функціональностями Java портів, цього може і не вистачити (JMX та ін.) Наприклад, для Glassfish необхідно за замовчуванням відкрити 4 порти відразу ж.

**Java! = Microsoft.** Незважаючи на те, що Java – першокласна мова на платформі, якоїсь виразної підтримки від вендорів зі світу Java можна і не дочекатися - все це тільки входить в моду.

**Відсутність шаблонів для Java Runtime і серверів додатків.** На платформі є стандартні образи віртуальних машин, які побудовані на основі Windows Server 2008 або Windows Server 2008 R2 різних білдів. Природно, за замовчуванням там немає Java Runtime і тим більше якихось серверів додатків. У цілому ж розгортання програми з Tomcat не займає особливо багато часу і не викликає труднощів. З Glassfish і JBoss трохи складніше, але знову ж ненабагато. Цей недолік знову ж дуже складно назвати недоліком - неможливо "осягнути неосяжне", неможливо мати підтримку всіх версій Tomcat і Glassfish, і при цьому просто освоювати платформу. Крім цього, рольова модель Windows Azure і Startup-Task дозволяють виконувати будь-які налаштування і установки при запуску екземпляра, в тому числі виробляти будь-які налаштування ваших серверів додатків. Microsoft не ставить ніяких обмежень у цьому контексті - Java-розробнику дозволено вибирати, встановлювати і налаштовувати будь-які версії Java Runtime і серверів додатків. А з плагіном для Windows Azure Eclipse все взагалі стає завданням внесення змін в текстовий файл конфігурації.

Як бачите, недоліків не так багато. Треба обов'язково згадати, що Microsoft активно працює над питаннями сумісності Windows Azure, ймовірно, яка є найбільш відкритою до всього світу ІТ-платформою.

**З цього можна зробити наступні висновки:**

Як вже було сказано вище, недоліків при розробці на Java на платформі Windows Azure не так багато, як може здатися на перший погляд. Так, необхідно переглядати архітектуру проекту, передбачати вартість, масштабованість, недоліки, переваги, специфічні для налаштування Windows Azure (брандмауери SQL Azure, балансувальників навантаження і так далі). Необхідно також чітко розуміти, чи потрібно взагалі переносити в хмару свій додаток - але це вже універсальне завдання, що не специфічне для Java-додатків, тому що, незважаючи на переваги та економічну ефективність, для деяких ситуацій хмара може і не підійти. У цілому ж платформа Microsoft - ще одна можливість для виконання ваших додатків. А, враховуючи явний інтерес корпорації до сумісності з Java, Python ... так, насправді, з усім, що зараз реально популярно і не-.NET, ця можливість повинна бути розглянута всерйоз.

**Огляд можливостей:**

Windows Azure повністю реалізує дві хмарні моделі - платформи як сервісу (платформа як послуга, PaaS) та інфраструктури як сервісу (інфраструктура як послуга, IaaS). Працездатність платформи Windows Azure забезпечують 8 глобальних дата центрів Microsoft. В пропозиції публічної хмари клієнт оплачує тільки ресурси і потужності, які задіяні в додаток і тільки за фактичний час використання цих ресурсів. Основні особливості даної моделі:

*оплата тільки спожитих ресурсів;*

*загальна, багатопотокова структура обчислень;*

*абстракція від інфраструктури.*

**Платформа як сервіс**

Модель надання платформи як сервісу реалізує можливість оренди платформи, яка зазвичай включає операційну систему і прикладні сервіси. Платформа як сервіс полегшує розробку, тестування, розгортання і супровід додатків без необхідності інвестицій в інфраструктуру і програмну середу. Платформа як сервіс також включає і інфраструктуру як сервіс.

В основі роботи Windows Azure лежить запуск віртуальної машини для кожного екземпляра додатку. Розробник визначає необхідний обсяг для зберігання даних і необхідні обчислювальні потужності (кількість віртуальних машин), після чого платформа надає відповідні ресурси. Коли початкові потреби в ресурсах змінюються, відповідно до нового запиту замовника платформа виділяє під додаток додаткові або скорочує невикористовувані ресурси дата-центру.

Особливістю Windows Azure як моделі PaaS є поділ програми та інфраструктури: розробнику достатньо лише визначити необхідний для роботи програми обсяг ресурсів, а всі дії щодо надання запитуваних ресурсів, управління ними, динамічному розподілу, моніторингу, масштабированию виконуються автоматично платформою Windows Azure.

Windows Azure як PaaS забезпечить не тільки всі базові функції операційної системи, але й додаткові: виділення ресурсів на вимогу для необмеженого масштабування, автоматичну синхронну реплікацію даних для підвищення відмовостійкості, обробку відмов інфраструктури для забезпечення постійної доступності та багато іншого. Крім того, надаватиметься середовище для виконання додатків, сервіси по зберіганню даних і ряд додаткових сервісів. Наприклад, інтеграційні та комунікаційні сервіси. Угода про надання послуг і сервісів (SLA) зазвичай покриває такі характеристики системи, як доступність середовища виконання додатків і її продуктивність.

Модель надання платформи як сервісу (платформа як послуга, PaaS) реалізує можливість оренди платформи, яка зазвичай включає операційну систему і прикладні сервіси. Платформа як сервіс полегшує розробку, тестування, розгортання і супровід додатків без необхідності інвестицій в інфраструктуру і програмну середу. Платформа як сервіс також включає інфраструктуру як сервіс (інфраструктура як сервіс, IaaS). Windows Azure повною мірою реалізує модель PaaS, коли платформа надається як сервіс. Споживачами PaaS зазвичай є самі компанії, які розробили програми.

Інфраструктура як сервіс Windows Azure так само реалізує інший тип сервісу - це інфраструктура як сервіс. Модель надання інфраструктури (апаратних ресурсів) реалізує можливість оренди таких інфраструктурних ресурсів, як сервери, пристрої зберігання даних та мережеве обладнання. Управління всією інфраструктурою здійснюється постачальником сервісів, а споживач управляє тільки операційною системою і встановленими додатками. Такі сервіси зазвичай оплачуються за фактичним використанням і дозволяють збільшувати або зменшувати обсяг інфраструктури через спеціальні портали, надані постачальниками сервісів. Хмарна платформа надає сервіси для запуску віртуальних машин і сервіси зберігання даних. У даній сервісній моделі можуть бути запущені практично будь-які додатки, встановлені на стандартні образи ОС. У Windows Azure ви можете легко використовувати власні образи Windows Server або Linux, а також вибрати образи із колекції. Залиште повний контроль над образами і підтримуйте їх відповідно до бізнес-вимог. Windows Azure також допомагає переносити ваші програми та інфраструктуру, не змінюючи існуючий код, що прискорює перехід.

1. **WA сервіси. Стандартні функції API.**

Windows Azure надає широкий набір сервісів: **Applications** — Середа виконання додатків - це високодоступні обчислювальні потужності, що дозволяють виконувати код додатків в хмарі і швидко масштабувати додатки в залежності від потреб клієнтів. Середа виконання додатків Windows Azure оперує такими основними поняттями:

**Cloud Services**:

**Web-роль** — веб-ролі в Windows Azure мають особливе призначення: надання виділеного веб-сервера служб IIS для розміщення інтерфейсних веб-додатків. Веб-ролі дозволяють легко і швидко розгортати веб-додатки з наступним масштабуванням обчислювальних ресурсів відповідно до потреб.

**Worker-роль** — додатки, розміщені в робочих ролях, можуть виконувати асинхронні, тривалі або безперервні завдання незалежно від дій користувачів. Ізоляція фонових процесів додатків в робочій ролі та розміщення інтерфейсної частини у веб-ролі дозволяє ефективніше розподілити логіку програми і більш точно керувати масштабуванням додатків.

**Web Sites**: Веб-сайти можна розробляти на ASP.NET, Node.js або PHP (Або CMS - WebMatrix, Joomla, Drupal, Wordpress, DotNetNuke, Umbraco та ін.) і розгортати за секунди з використанням FTP, Git, TFS, Mercurial і Dropbox. При цьому первинне використання безкоштовне, далі ж веб-сайти вертикально масштабуються згідно збільшенню трафіку. Типово веб-сайти знаходяться в стані загальному або безкоштовному, тобто потужності діляться між веб-сайтами, але при необхідності можна збільшити кількість примірників і перевести веб-сайт в стан захисту. З червня 2013 сервіс веб-сайтів офіційно підтримує користувальницькі сертифікати SSL (раніше підтримувалися тільки сертифікати, пропоновані Microsoft) як за IP-адресою, так і на базі SNI.

**Data Management** — нереляційні сховища даних: таблиці, диски, черги, зберігання двійкових об'єктів + реляційне сховище даних у вигляді бази даних SQL.

**Таблиці** — сховище таблиць використовується додатками, які зберігають великі обсяги даних з додатковими вимогами до структурування. У таблиці зберігаються структуровані дані, між якими не встановлюються відносини.

* **Черги** — черги забезпечують надійний і безперервний обмін повідомленнями між додатками.
* **Блоби** — сховище BLOB-об'єктів - це найпростіший спосіб зберігання великих обсягів неструктурованих текстових або двійкових даних, таких як відео, музичний файл та зображення.
* **SQL Database** — реляційна база даних - це високодоступна, швидка служба Бази даних, побудована на основі технологій SQL Server.
* **SQL DataSync** — хмарна служба синхронізації даних, що забезпечує як односпрямовану, так і двонаправлену синхронізацію. Служба синхронізації даних дозволяє легко обмінюватися даними між SQL Azure і локальними базами даних SQL Server, а також між кількома базами даних SQL баз даних (SQL Azure);
* **SQL Reporting** — служба звітів Microsoft SQL дозволяє легко вбудувати в додаток Windows Azure можливості роботи зі звітами. Доступ до звітів можна отримати через портал Windows Azure, веб-браузер або безпосередньо з програми. Завдяки можливостям хмари відпадає необхідність у створенні та підтримці власної інфраструктури звітів;
* **SQL Federations** — федерація SQL Azure значно спрощує масштабування безлічі баз даних, розміщених на сотнях вузлів, що дозволяє клієнтам платити тільки за реально використовувані ресурси.
* **Azure Backup** — цей сервіс пропонує можливість організації захищеної інфраструктури збереження резервних копій Windows Server в хмарі. На 16.04.2013 сервіс Windows Azure резервного копіювання здійснює підтримку бекапів інформації із систем на базі Windows Server 2008 R2 SP1 і Windows Server 2012, Windows Server 2012 Essentials, і System Center Data Protection Manager 2012 SP1 в Windows Azure. Підтримуються інкрементальні бекапи, що дозволяє економити трафік і місце зберігання і мати можливість відновлення до обраної точки з безлічі версій. Підтримуються політики і всі стандартні засоби бекапів Windows Server для організації стиснення даних і шифрування.

**Performance**:

* **Content Delivery Network** — мережа кешуючих серверів (мережа CDN) підвищує продуктивність додатків шляхом кешування контенту якнайближче до клієнтів і користувачів, забезпечуючи максимально ефективну взаємодію з ними, наприклад, мережа CDN дозволяє доставляти фрагменти мультимедійних файлів для динамічного адаптивного відтворення мультимедіа поверх HTTP-контенту.
* **Caching** — розподілений кеш в пам'яті, за допомогою якого ви замість повільного дискового сховища додатка отримуєте високошвидкісний доступ до даних, що зберігаються в оперативній пам'яті, з можливістю масштабування;
* **Media Services** — служби мультимедіа забезпечують гнучкість, масштабованість і надійність хмарної платформи для обробки об'єктів мультимедіа високої якості для користувачів у всьому світі. Служби мультимедіа включають в себе хмарні версії багатьох існуючих технологій платформи мультимедіа Microsoft і багатьох партнерів, в тому числі для перегляду, кодування, перетворення формату та захисту контенту, а також потокової передачі за запитом і в реальному часі. Покращуючи існуючі рішення або створюючи нові, користувач можете легко об'єднувати служби мультимедіа і керувати ними для формування робочих процесів для будь-якої потреби.
* **Mobile Services** — Windows Azure Mobile Services пропонує хмарну інфраструктуру для всіх популярних мобільних платформ: Windows 8, Windows Phone, IOS і Android. На основі сервісу можна побудувати хмарний бекенд, на який можна перенести завдання по зберіганню даних, аутентифікації і Push-повідомлень.

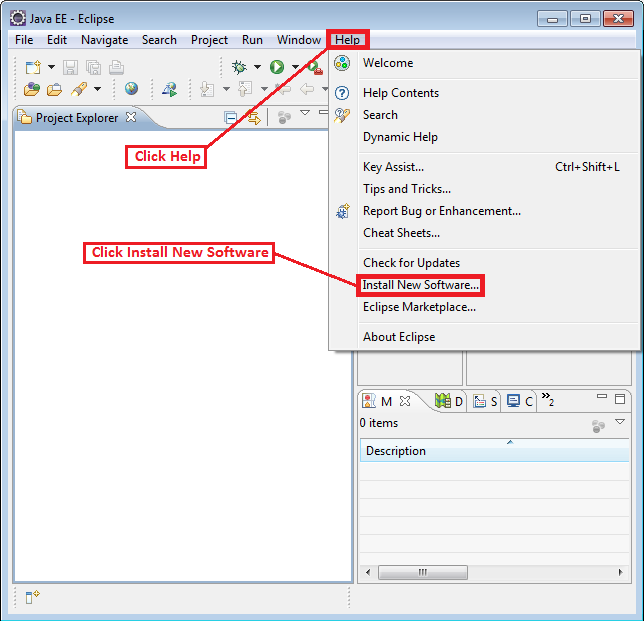
1. **Встановлення плагіну Windows Asure для Eclipse IDE.**

Плагін Windows Azure для Eclipse Java (по Microsoft Open Technologies) надає шаблони і функції, які дозволяють вам легко створювати, розробки, тестування і розгортання Windows Azure додатків за допомогою середовища Eclipse Development. Він є проектом з відкритим вихідним кодом, чий вихідний код доступний під ліцензією Apache License 2.0 з сайту проекту на <https://github.com/MSOpenTech/WindowsAzureToolkitForEclipseWithJava>.

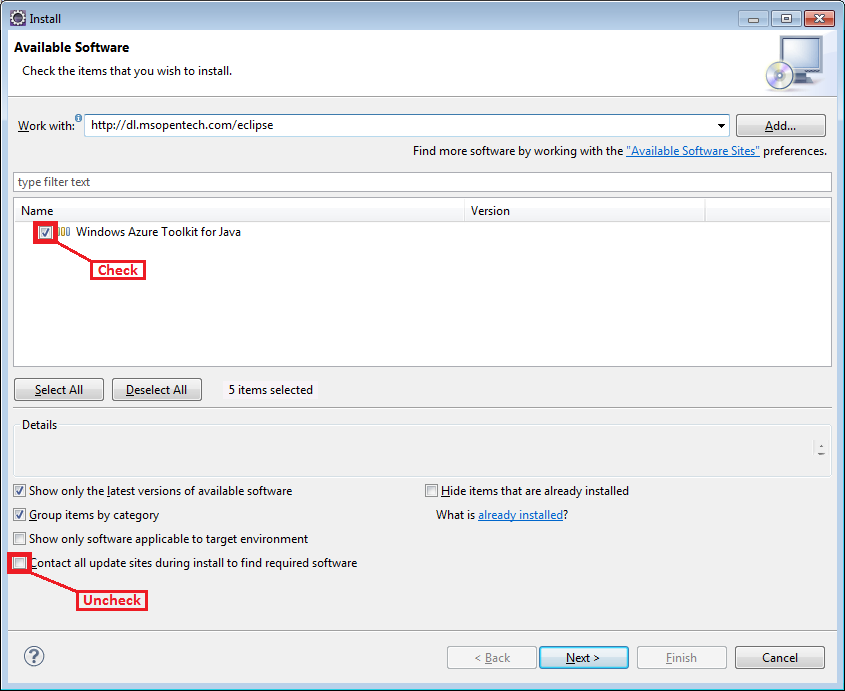
Цей модуль вимагає Windows Azure SDK 2.0. Це може бути завантажено за допомогою інсталятора веб-платформи (WebPI) на http://go.microsoft.com/fwlink/?LinkID=252838. Однак, якщо ви не маєте його встановленого​​, при створені вашого першого Windows Azure проекту розгортання, плагін автоматично встановить Windows Azure SDK 2.0.

Для встановлення Windows Azure Toolkit для Java:

* 1. Запустіть Eclipse;
  2. У Eclipse, в меню виберіть Довідка, а потім натисніть встановити нове програмне забезпечення, як показано на наступному малюнку.



* 1. У діалоговому вікні Доступного Програмного забезпечення, в роботі з текстовим полем, типу http://dl.msopentech.com/eclipse введіть ключ.
  2. В імені панелі, перевірте Windows Azure Toolkit для Java, і зніміть прапорець Контакти всіх сайтів оновлення в процесі установки, щоб знайти необхідне програмне забезпечення. Ваш екран повинен виглядати приблизно наступним чином:



Якщо розгорнути Windows Azure Toolkit для Java, можна побачити наступне:

* **Microsoft JDBC Driver 4.0 for SQL Server**: Цей компонент спрощує розробку додатків баз даних за допомогою SQL.
* **Package for Apache Qpid Client Libraries for JMS (by MS Open Tech)**: Цей компонент надає бібліотеку JMS клієнт з проекту Apache Qpid, щоб дозволити вашому додатку використовувати Advanced Queuing повідомлення протоколу (AMQP) обміну повідомленнями в Windows Azure.
* **Package for Windows Azure Libraries for Java (by MS Open Tech)**: Цей компонент дозволяє створювати додатки Windows Azure в Java, які дозволяють скористатися перевагами Windows Azure масштабованих обчислювальних ресурсів хмари.
* **Windows Azure Access Control Services Filter (by MS Open Tech)**: Цей компонент забезпечує підтримку аутентифікації користувачів додатків з Windows Azure ACS.
* **Windows Azure Common Plugin (by MS Open Tech)**: Цей компонент містить загальні функції, на які посилаються інші компоненти.
* **Windows Azure Plugin for Eclipse with Java (by MS Open Tech)**: Цей компонент містить логіку конфігурації проекту, публікація-в-хмарі майстра, і користувальницький інтерфейс.
  1. Натисніть кнопку Далі. (Якщо у вас виникли незвичайні затримки під час установки плагіна, переконайтеся, що є зв'язок з усіма сайтами оновлення в процесі установки, щоб знайти необхідне не встановлене програмне забезпечення).
  2. У вікні Встановити подробиці, натисніть кнопку Далі.
  3. У діалоговому вікні ліцензії, ознайомтеся з умовами ліцензійних угод. Якщо ви згодні з умовами ліцензійної угоди, виберіть Я приймаю умови ліцензійної угоди та натисніть кнопку Готово. (Решта кроків передбачає, що Ви робите прийняття умови ліцензійної угоди. Якщо ви не згодні з умовами ліцензійної угоди, вийдіть з процесу установки.)
  4. Якщо буде запропоновано перезавантажити Eclipse, для завершення установки, натисніть кнопку Перезавантажити зараз.

1. **Реєстрація додатку у WA, тестування за допомогою WA емулятора та запуск додатку.**

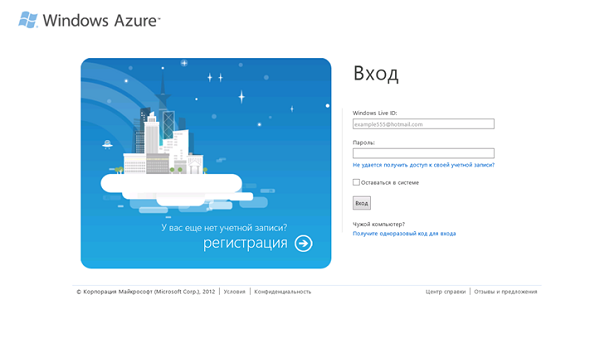
[***http://habrahabr.ru/post/147204/***](http://habrahabr.ru/post/147204/)

***http://habrahabr.ru/post/166159/***

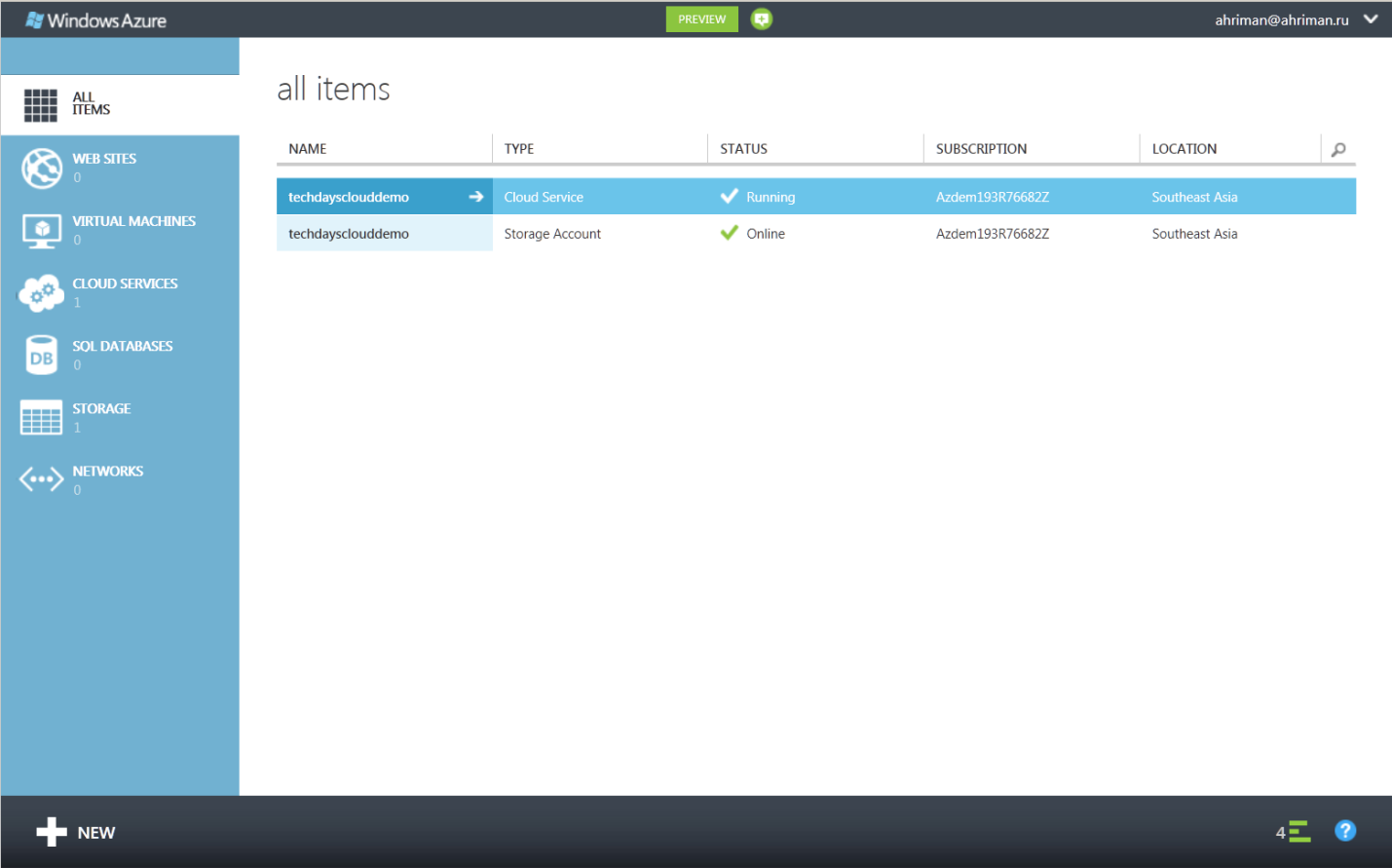
[***http://msdn.microsoft.com/ru-ru/jj569864.aspx***](http://msdn.microsoft.com/ru-ru/jj569864.aspx)

**Додавання заявки на доступ до функціональності Preview на порталі адміністрування.**

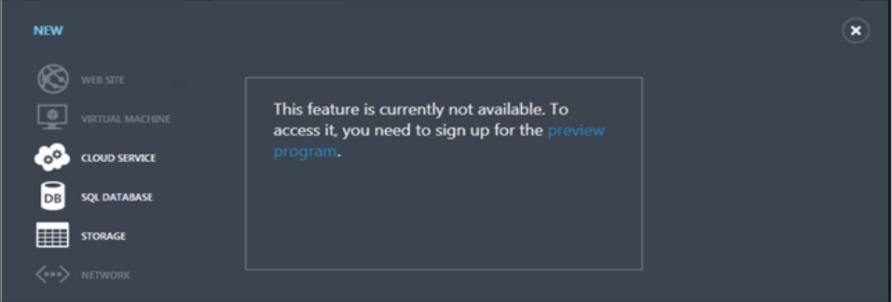
Windows Azure Web Sites знаходиться в стадії "Preview". Отримати доступ можна, подавши заявку на безкоштовний тримісячний період використання Windows Azure і додавши необхідну функціональність на порталі управління чи запросивши доступ до функціональності Preview на сайті Windows Azure.



Увійдіть на портал керування Windows Azure, використовуючи облікові дані Windows Live ID.



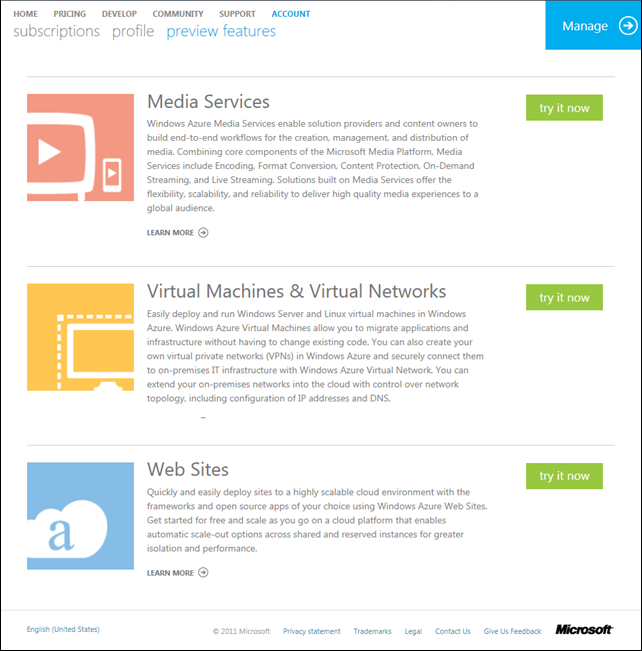
Увійшовши на портал керування, натисніть кнопку New, розташовану в нижньому лівому кутку сторінки.



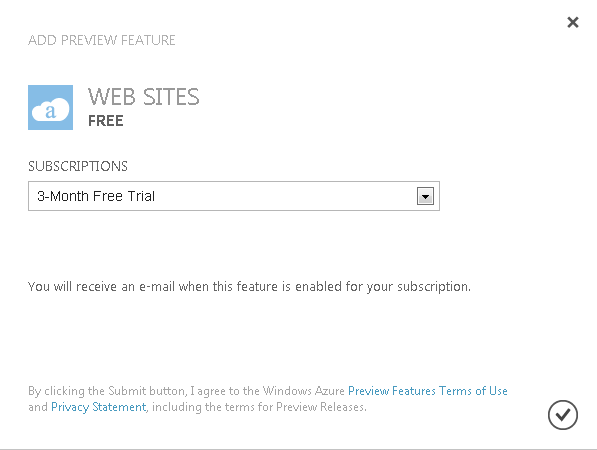
для відкриття діалогового вікна New form.



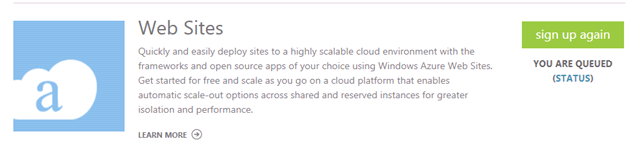
У діалоговому вікні New form виберіть опцію Web Site. Натисніть на ссилку preview program. На сторінці, натисніть Preview features, щоб перейти на сторінку додавання заявки на функціональність Preview.



На сторінці preview features натисніть try it now поруч з Web Sites.



На формі, що з'явилася Add Preview Feature виберіть свою підписку (якщо у вас кілька підписок) і натисніть на кнопку, щоб запросити доступ до функціональності Preview.



Після натискання поруч з назвою функціональності з'являється напис You Are Queued.

**The Windows Azure Emulator**

Windows Azure Compute Emulator програмної емуляції Windows Azure Compute - це сервіс, який працює в місцевому середовищі, так що ви можете тестувати і налагоджувати додатки Windows Azure, перш ніж розгорнути їх на Windows Azure. Емулятори обчислень обробляються так званим життєвим циклом примірників ролей (lifecycle of your role instances), а також забезпечують доступ до штучних ресурсів, таких як місцеві ресурси зберігання. При налагодженні або запуску послуги від Visual Studio, Visual Studio автоматично запускає емулятор обчислювання в якості фону додатка, а потім розгортає ваш сервіс для емуляторів обчислень. Ви можете використовувати обчислювальні емулятори для перегляду ваших послуг, коли вони працюють в місцевому середовищі.

***Примітка:***

При публікації хмарного сервісу від Visual Studio, ви можете включити IntelliTrace для будь-яких ролей у цьому сервісі, які призначені для. NET Framework 4 або. NET Framework 4.5. З IntelliTrace можна переглянути події, які відбулися, наприклад, в минулому, і відтворити контекст того часу.

***Налагодження хмарного сервісу:***

- Якщо ви не хочете налагоджувати, ви можете вибрати Запуск без налагодження для запуску програми в локальному середовищі розробки без налагодження. Якщо запустити без налагодження, розгортання додатків продовжує працювати, поки ви самі це не зупините.

- Для відображення користувачеві обчислювального емулятора інтерфейсу, відкрийте контекстне меню для Windows Azure значок в області повідомлень і виберіть Показати Обчислення емулятора користувачу інтерфейсу.

У лівій частині інтерфейсу Compute емулятор покаже ті служби, які в даний час розгорнуто на емуляторі обчислень і кожну службу, яка працює. Ви можете вибрати послугу або роль для відображення життєвого циклу, реєстрацію та діагностичну інформацію в правій панелі. Якщо ви помістите фокус у верхнє поле включеного вікна, воно розширеться, щоб заповнити необхідні панелі.

- Для налагодження додатка, ви можете покроково виконати додаток, за допомогою команд у меню Налагодження, а також встановити точки зупину в коді. Коли ви виходите через програми в відладчик, панелі оновлюються з урахуванням поточного стану програми. При зупинці налагодження, розгортання додатків видаляється.

Якщо ваш додаток включає в себе веб-ролі, виберіть для нього тип запуску і властивість дії, щоб запустити веб-браузер, Visual Studio запустить ваш веб-додаток в браузері.

Якщо змінити кількість примірників ролей у конфігурації служби, то необхідно зупинити ваш Windows Azure додаток, а потім перезапустити налагодження, так щоб ви могли налагоджувати ці нові екземпляри ролей.

1. **Збереження даних додатку у БД та створення прототипу додатку.**

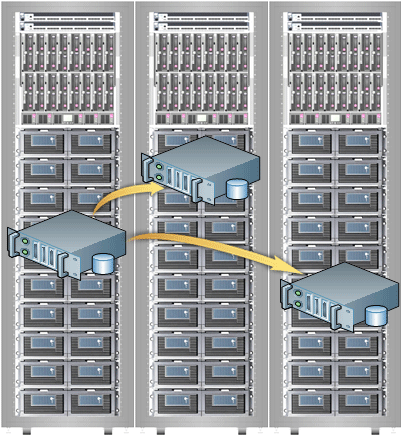
Служби баз даних SQL в Windows Azure володіють вбудованими можливостями щодо забезпечення відмовостійкості. Для захисту бази даних від небажаного видалення або зміни даних або від масштабних руйнувань в центрі обробки даних потрібна розробка стратегії забезпечення безперервності бізнесу.

**Як захистити бази даних від збою окремих серверів і пристроїв**

Використання База даних SQL Windows Azure для зберігання даних надає доступ до безлічі елементів відмовостійкій і безпечної інфраструктури, які в іншому разі довелося б проектувати, купувати, реалізовувати і адмініструвати. У цьому розділі описано, що База даних SQL Windows Azure може зробити для вас без додаткових витрат.

**Надмірність інфраструктури**

База даних SQL Windows Azure дозволяє пом'якшити наслідки простоїв через збої окремих пристроїв, таких як жорсткі диски, адаптери мережного інтерфейсу або навіть цілі сервери. Надійність і відмовостійкість даних підвищуються за рахунок використання декількох копій всіх даних у різних фізичних вузлах, розташованих в повністю незалежних фізичних підсистемах, таких як стійки серверів і мережеві маршрутизатори. У будь-який момент часу База даних SQL Windows Azure зберігає три репліки використовуваних даних - одну первинну репліку і дві вторинні репліки. База даних SQL Windows Azure використовує схему фіксації на основі кворуму, в якій дані спочатку записуються на первинну і одну вторинну репліку, і тільки після цього транзакція вважається зафіксованої. При відмові обладнання первинної репліки База даних SQL Windows Azure виявляє збій і переходить на вторинну репліку. У випадку фізичної втрати репліки База даних SQL Windows Azure створює нову репліку автоматично. Таким чином, у центрі даних є як мінімум дві фізичні копії ваших даних, узгоджені на рівні транзакцій. На наступній діаграмі показано, як База даних SQL Windows Azure зберігає три репліки у фізичних стійках серверів в центрі даних.



Крім надлишкових реплік База даних SQL Windows Azure створює внутрішні копії даних за останні 14 днів для всіх баз даних, наявних в центрі даних. Ці копії забезпечують захист від одночасних або катастрофічних апаратних або системних збоїв, але вони недоступні для клієнтів. Рекомендується реалізувати власні рішення для резервного копіювання і відновлення.

**Важливі примітки**

1. Мета полягає в тому, щоб збої, що відбуваються в центрі даних, не приводили до втрати даних, а викликали б тимчасові відключення і збої транзакцій.
2. Ваш додаток повинен бути стійким до тимчасових відключень. Рекомендується реалізувати логіку повтору в додатку для запобігання втрат сполук.
3. В даний час ми не надаємо SLA для типової цільової точки відновлення (RPO) і цільового часу відновлення (RTO).

**Як захистити свою базу даних від небажаного видалення або змін**

Користувачі або додатки можуть вносити до дані небажані зміни. Для цього можуть знадобитися зворотні операції. Наприклад, користувач може змінити якісь дані, що відносяться не до того клієнту і так далі. Можливість відновити дані програми у разі пошкодження або небажаної зміни або видалення - це основоположне вимога для програмних додатків. Рекомендується реалізувати такі підходи для створення резервних копій примірників База даних SQL Windows Azure.

Копії бази даних можна використовувати в якості варіанту резервного копіювання і відновлення для захисту бази даних, як описано в наведених далі розділах

**Важливі примітки**

1. Копіювання баз даних може тривати довго, тривалість може відрізнятися.
2. В рахунок буде включатися плата за кожну копію бази даних. Але плата за нову копію бази даних стягується тільки після успішного завершення процесу копіювання. Плата за кожну копію бази даних береться за тим самим тарифом, що й за базу даних-джерело.
3. Ви відповідаєте за керування копіями та їх видалення в міру необхідності.
4. Показник використання бази даних обчислюється щодня, так що якщо копія бази даних використовується тільки один день, то рахунок за використання копії бази даних також виставляється за один день.
5. Відновлення на іншому логічному сервері вимагає внесення зміни до рядка підключення.
6. Відновлення на іншому логічному сервері не гарантує кращий захист від втрати даних і підвищення рівня продуктивності системи.
7. Під час виконання процесу копіювання всі зміни у вихідній базі даних реплікуються в копію. Це слід враховувати при прийнятті рішення про те, яку з копій бази даних слід відновлювати.
8. У версії База даних SQL, випущеної в листопаді 2011 року, вперше реалізовані Федерації в базі даних SQL Windows Azure (колишня назва - SQL Azure). У даний момент не можна виконати операцію копіювання бази даних, яка містить федерації. Вірно і зворотне, створення федерації завершиться збоєм, якщо виконується операція копіювання бази даних. Копіювання бази даних заблоковано також і для членів федерації.
9. Цільовий час відновлення (RTO) повинен дорівнювати часу для розпізнавання помилки плюс час на перейменування бази даних плюс інтервал часу між виконанням двох інструкцій ALTER.

**Захист бази даних від масштабних руйнувань в центрі обробки даних**

Для захисту від втрати даних у разі катастрофи в центрі обробки даних слід створити сховище резервних копій бази даних поза межами того центру обробки даних, на якому розміщена база даних програми. У подібному випадку рекомендується використовувати копіювання бази даних

**Для поліпшення захисту даних від можливих аварій в центрі обробки даних можна реалізувати такі підходи.**

1. Експорт файлу BACPAC у великий двійковий об'єкт з використанням облікового запису сховища в іншому центрі обробки даних.
2. Експортувати BACPAC-файл у великий двійковий об'єкт з використанням облікового запису сховища в тому ж центрі обробки даних і сподіватися, що географічна реплікація сховища Windows Azure скопіює BACPAC-файл в інший центр обробки даних.
3. Імпортувати BACPAC-файл в локальний екземпляр SQL Server.

**Служба синхронізації даних Data Sync і аварійне відновлення**

Служба Microsoft База даних SQL Windows Azure Data Sync надає можливості синхронізації даних для примірників База даних SQL Windows Azure. В даний час служба забезпечує дві основні функції:

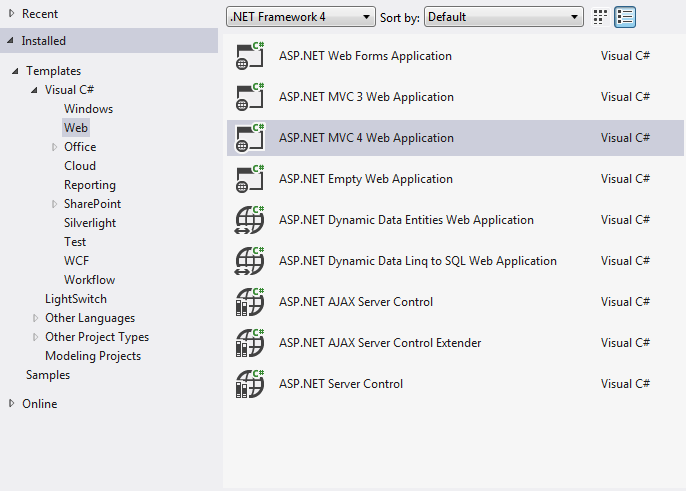
1. Синхронізацію даних між локальними базами даних SQL Server та примірниками База даних SQL Windows Azure, що забезпечує можливість використання одних і тих же даних як локальними, так і хмарними додатками.
2. Синхронізацію даних між двома і більше базами даних База даних SQL, які можуть перебувати в одному центрі обробки даних, в різних центрах обробки даних або в різних регіонах. Ця функція надає механізм масштабування додатків шляхом багаторазового копіювання даних. Вона також дозволяє розгортати безліч екземплярів додатків і баз даних по всьому світу ближче до кінцевих користувачів; при цьому всі бази даних завжди синхронізовані. До відома, служба синхронізації даних SQL часто використовується разом з Windows Azure Traffic Manager.

**Windows Azure SQL Databases для Windows Phone 7**

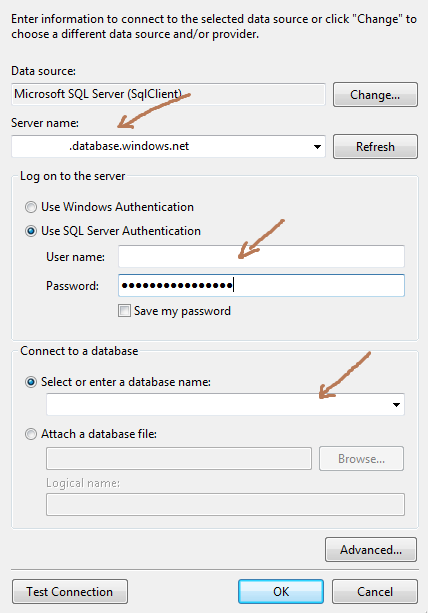
Ідея одного з проектів полягала у створенні бази даних продуктів харчування: штрих-код, склад, шкідливі добавки і т.д. Оскільки цю базу даних передбачалося використовувати на мобільних пристроях, зокрема Windows Phone 7, рішення про те, які технології зберігання даних використовувати при розробки проекту схилялося до однієї з технологій: Windows Azure SQL Databases (WASQLDB).

Передачу даних з Windows Azure SQL Databases на клієнт досить просто здійснити з використанням OData WCF Services. OData представляє дані в якості джерел, до яких можна звертатися за URL, змінювати дані можна за допомогою стандартних HTTP-заголовків. Для представлення джерел в якості наборів сутностей, пов'язаних з допомогою асоціацій OData використовує зв'язки і сутності моделі EDM. У результаті доступ до цих служб можна отримати з будь-якого клієнта, який підтримує OData. Власне, сам процес не займає багато часу і являє собою «розробку на одному диханні». Я розповім, Вам, про те, як створити клієнт для прийому даних з Windows Azure, а саме Windows Azure SQL Databases.

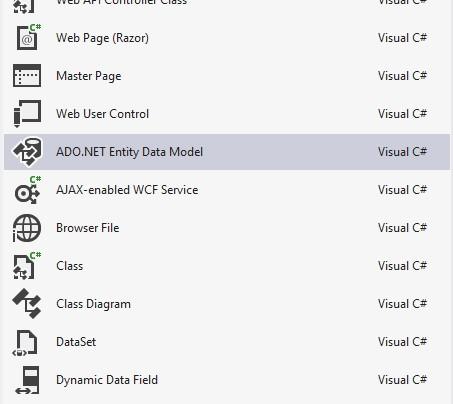
 Створюємо новий проект ASP.Net Web Application.



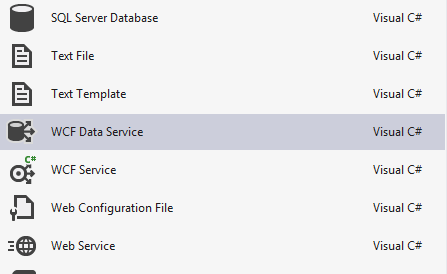
Далі створюємо підключення до WASQLDB. Windows Azure SQL Data Bases - проекція традиційного SQL Server на хмару, що надає можливості для роботи з базою даних за допомогою інтернет-сервісів. Ця технологія дозволяє виконувати реляційні запити, а також надає функціонал для здійснення пошуку, створення аналітичних звітів, інтеграції і синхронізації даних

. 

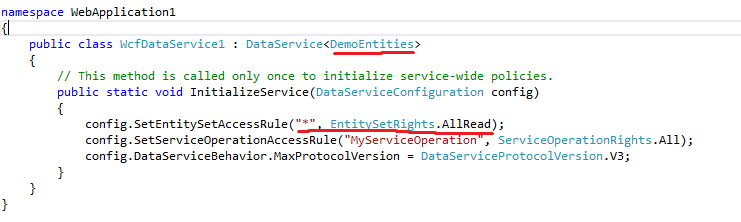
Перш ніж приступати до створення WCF Data Services, знадобляться дані, з якими доведеться працювати. Для надання цих даних використовується Entity Framework.  Після того, як настроювання джерела даних закінчені, додаємо підтримку ADO.NET Entity Framework Model в проект.



Далі додаємо служби WCF Service в проект і змінюємо код.



Тепер ми повинні додати певний код для щойно створеного Data Service, і відредагувати першу сходинку, вказавши джерело даних, який створювали раніше. Встановимо також правила доступу для таблиць в БД.



На даному етапі можна перевірити правильність виконання роботи, запустивши проект. Для того, щоб перейти і побачити структуру БД, необхідно перейти за посиланням: http://localhost:47274/wcfdataservice1.svc/



Отже, створення WCF-сервісу закінчено. Тепер же залишається створити проксі-клас для підключення до сервісу на клієнті. Запустимо cmd.exe і скористаємося нехитрої командою для створення проксі:



В результаті отримуємо файл Proxy.cs, який і містить проксі-клас. Далі ми можемо використовувати цей клас де завгодно - від консольних до мобільних додатків. При цьому нам залишиться тільки вказати адресу, за якою розташований сервіс.

1. **Керування поштою (SendGrid) та виконання задач за графіком*.***

**Як відправити листа використовуючи SendGrid з Windows Azure**

**Що таке SendGrid?**

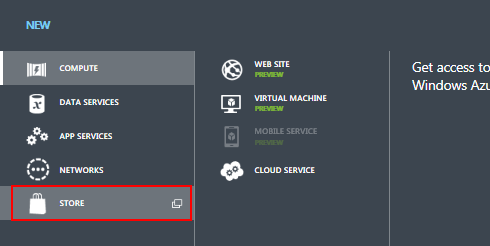
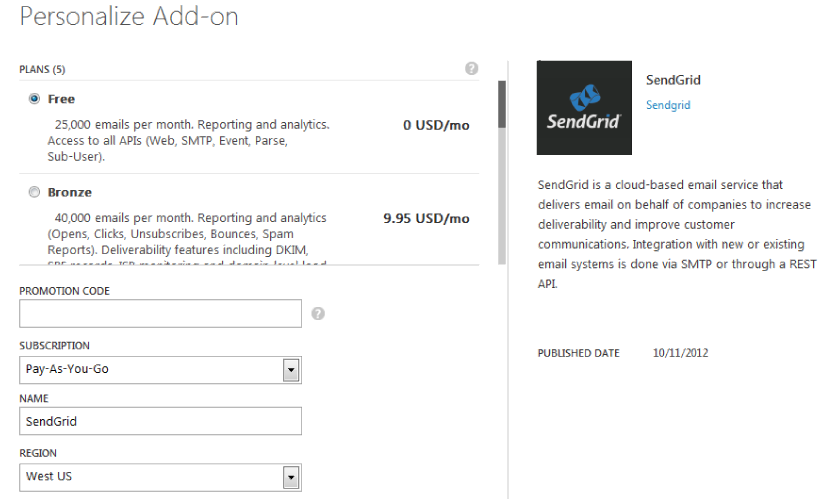
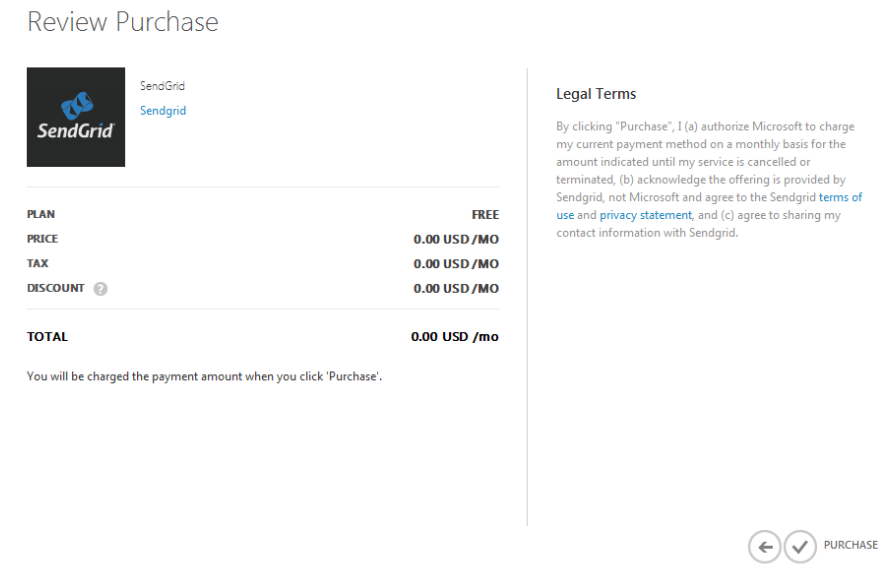
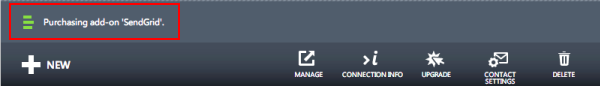
SendGrid це хмарна служба електронної пошти, що забезпечує надійну доставку електронної пошти, масштабованість, аналіз в режимі реального часу поряд з гнучким API-інтерфейсом, який робить легку інтеграцію користувача. Загальне використання SendGrid сценаріїв включають в себе:

1. Автоматичне відправлення квитанцій клієнтам
2. Адміністрування списків розсилки для відправки клієнтам щомісячних електронних флаєрів і спеціальних пропозицій.
3. Збір у режимі реального часу показників блокованих мейлів, і клієнтів.
4. Створення звітів для виявлення тенденцій.
5. Переадресація запитів клієнтів.

**Створення SendGrid акаунту**

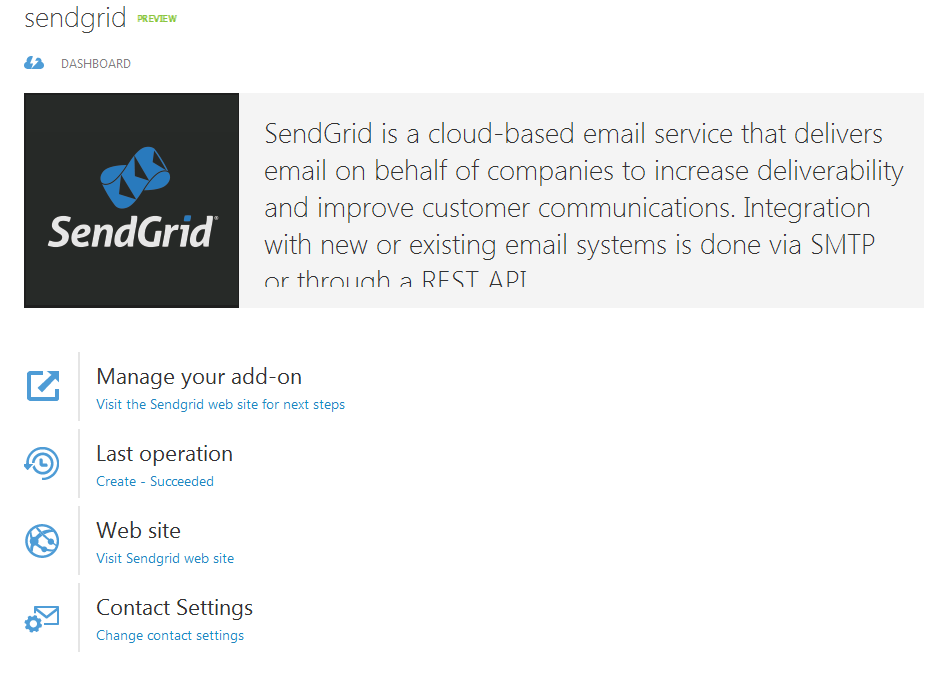
Windows Azure замовники можуть розблокувати 25000 безкоштовних повідомлень електронної пошти кожен місяць. Ці 25000 безкоштовні щомісячні електронні листи дадуть вам доступ до розширених звітів та аналітики, і всіх інтерфейсів (Web, SMTP, подія, Parse Sub-User).

**Щоб підписатися на SendGrid акаунт**

1. Увійти в Windows Azure Management Portal.
2. У нижній панелі управління порталу, натисніть кнопку Створити. 
3. Натисніть Store. 
4. У Choose an Add-on dialog діалоговому вікні виберіть SendGrid і натисніть стрілку вправо.
5. У Personalize Add-on диалоговому вікні виберіть SendGrid, який ви хочете зареєструвати.
6. Введіть ім'я для ідентифікації SendGrid послуг в вашому параметрах Windows Azure, або використайте значення за замовчуванням SendGrid. Імена повинні бути в межах від 1 до 100 символів в довжину і містити тільки буквено-цифрові символи, тире, крапки та підкреслення. Ім'я повинно бути унікальним в вашому списку підписок Windows Azure Store Items. 
7. Виберіть значення для регіону, наприклад, Захід США.
8. Натисніть стрілку вправо.
9. На вкладці Review Purchase, перегляньте план та інформацію про ціни і перегляньте юридичні терміни. Якщо ви згодні з умовами, натисніть на прапорець. Після натискання на галочку, ваш SendGrid акаунт почне процес підготовки. 
10. Після підтвердження покупки ви будете перенаправлені панель доповнень, і ви побачите повідомлення Purchasing SendGrid. 

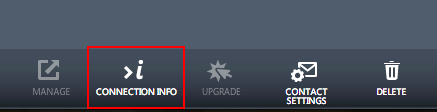
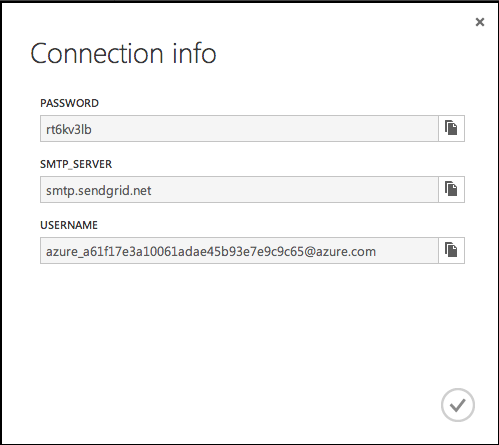
Ваш SendGrid акаунт буде доступний для вас негайно, і ви побачите повідомлення Successfully purchased Add-On SendGrid. Ваш акаунт і повноваження створені. Ви готові для відправки листів в цій точці.

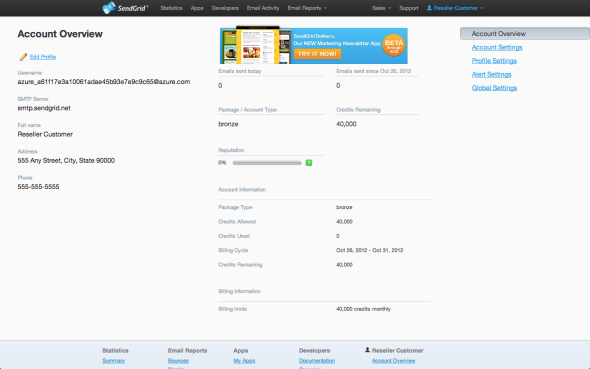
Щоб змінити тарифний план або побачити SendGrid налаштування контакту, клацніть ім'я вашої SendGrid сервісу і відкриється SendGrid приладова панель.



Щоб відправити електронний лист за допомогою SendGrid, необхідно вказати ваші облікові дані (ім'я користувача і пароль).

**Щоб знайти SendGrid повноваження**

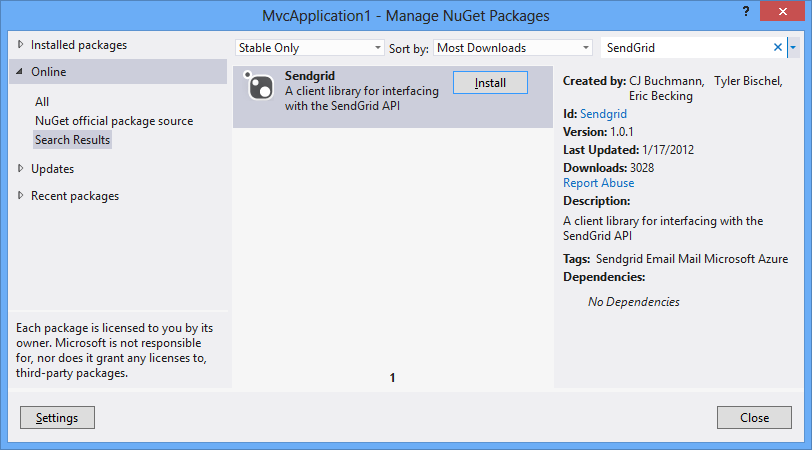
1. Натисніть Connection Info. 
2. У Connection info діалоговому вікні, скопіюйте Password і Username, щоб використовувати пізніше в цієї інструкції. 

Щоб налаштувати параметри електронної пошти, натисніть кнопку Manage. Це відкриє Sendgrid.com веб-інтерфейс, де ви можете увійти і відкрити панель управління SendGrid. 

**Reference the SendGrid .NET class library**

Пакет SendGrid NuGet це найпростіший спосіб отримати SendGrid API і налаштування програми з усіма залежностями. NuGet є Visual Studio розширенням, що входить до Microsoft Visual Studio 2012, що дозволяє легко встановлювати та оновлювати бібліотеки та інструменти.

**Для установки пакета NuGet SendGrid у вашому додатку, виконайте такі дії:**

1. У Solution Explorer клацніть правою кнопкою миші References, виберіть пункт Manage NuGet Packages.
2. У лівій частині Manage NuGet Packages діалогового вікна натисніть Online.
3. Знайдіть SendGrid і виберіть SendGrid елемент у списку результатів. 
4. Натисніть кнопку Install, щоб завершити установку, а потім закрийте це діалогове вікно.

SendGrid's .NET class library називають SendGridMail. Він містить наступний простір імен:

* SendGridMail для створення та роботи з елементами електронної пошти.
* SendGridMail.Transport для відправки електронної пошти з використанням або протоколу SMTP, або HTTP 1.1 протоколу з Web / REST.

Додайте наступні коди оголошення простору імен у верхній частині С# файлу, в якому ви хочете отримати програмний доступ к SendGrid email сервісу. System.Net і System.Net.Mail, це простір імен .NET Framework включених в нього, тому що вони включають типи, які ви звичайно використовуєте с SendGrid APIs.

using System.Net;

using System.Net.Mail;

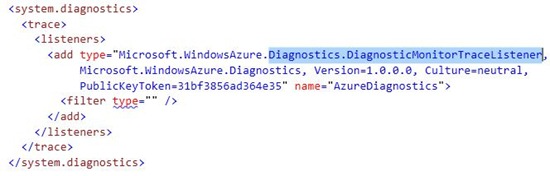
using SendGridMail;

using SendGridMail.Transport;

1. **Використання журналів подій та консолі адміністратора.**

**Журнали Windows Azure**

Журнали Windows Azure містять специфічну для програми діагностичну інформацію, створювану на основі трасувань подій у додатку. Для того, щоб інфраструктура Windows Azure могла отримувати дані від цих подій, програма має використовувати відповідні механізми трасування (слухачі трасування). Клас DiagnosticMonitorTraceListener в ​​просторі імен Microsoft.WindowsAzure.Diagnostics повинен бути доданий до списку механізмів трасування у файлі web.config (для веб-ролі) або app.config (для прикладної ролі). Нижче показано вміст файлу web.config з використанням DiagnosticMonitorTraceListener:



Другий спосіб додавання джерел даних до списку механізмів трасування – це використання методу System.Diagnostics.Trace.Listeners.Add.

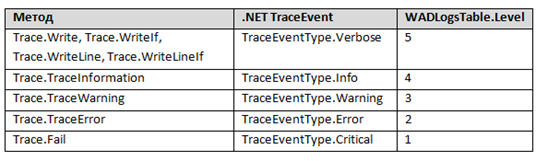
Коли додаток генерує трассіровочние дані, вони обробляються відповідним механізмом трасування і потрапляють в буфер. Типово, монітор діагностики збирає дані з журналів Windows Azure журнали. Проте, для збереження даних потрібно додати до додатка відповідний код (про це - нижче). Коли дані передаються з буфера, вони зберігаються в таблиці Windows Azure Storage, яка має назву WADLogsTable. Ця таблиця має такі характеристики:

* PartitionKey
* RowKey
* Timestamp
* EventTickCount
* DeploymentId
* Role
* RoleInstance
* Level
* EventId
* Pid
* Tid
* Message

Клас трасування з простору імен System.Diagnostics містить ряд методів, що дозволяють записувати в механізм трасування різні типи повідомлень. До таких методів належать методи Write, WriteIf, WriteLine і WriteLineIf. Значення, що передається цим методам, заноситься в таблицю WADLogsTable у властивість Message. При записі, значення комбінується із категорією повідомлення. наприклад,

Trace.WriteLine(“”“Трассировочное сообщение”, “General”);

буде збережено в таблиці WADLogsTable як «Загальні: трассіровочние повідомлення». Група трасувань методів також включає методи TraceInformation, TraceWarning, TraceError і Невдача. Ці методи дозволяють вказувати тип події - інформація, попередження, помилка збій, а також рівень повідомлення для кожного типу. У наступній таблиці показано відповідність методів і типів подій.



**Журнали IIS 7.0**

Журнали IIS 7.0 представляють собою запити до IIS, оброблювані відповідним процесом. Файли журналів зберігаються в каталозі% SystemDrive% \ Inetpub \ журнали \ LogFiles на локальному диску кожного екземпляра ролі. Для веб-ролей ведення журналу для IIS включено за замовчуванням. Тим не менш, необхідно переконатися в тому, що монітор діагностики налаштований для того, щоб збирати інформацію в каталогах. Для того, щоб це зробити, необхідно задати відповідний інтервал через властивість TimeSpan методу ScheduledTransferPeriod. Якщо інтервал не заданий, інформацію можна передавати за запитом. Нижче показаний приклад установки значення властивості TimeSpan.

*var conf = DiagnosticMonitor.GetDefaultInitialConfiguration();*

*conf.Directories.ScheduledTransferPeriod = TimeSpan.FromSeconds(10);*

Коли Windows Azure передає журнали IIS, файли завантажуються в контейнер для зберігання бінарних об'єктів пиж-IIS-лог-файлів. Для перегляду самих файлів слід використовувати утиліти, що дозволяють завантажувати і відображати бінарні об'єкти зі сховища Windows Azure Storage. Однією з таких утиліт є утиліта перегляду Контейнер Blob, що входить до складу Windows Azure Tools для Microsoft Visual Studio.

**Інфраструктурні журнали Windows Azure**

Інфраструктурні журнали Windows Azure містять діагностичні дані, специфічні для Windows Azure, які збираються автоматично. Ці дані зберігаються в таблиці WADDiagnosticInfrastructureLogsTable і являють собою події (інформація і помилки), пов'язані з розгортання додатків. Дані, збережені в цьому журналі, слід використовувати при вирішенні проблем, пов'язаних з розгортанням додатків і знаходження помилок, що виникають на рівні інфраструктури.

**Журнали запитів зі збоями**

Журнали запитів зі збоями (Не вдалося Журнали Request) не заповнюються за замовчуванням і повинні бути включені шляхом зазначення відповідних опцій трасування у файлі web.config додатки. Журнали запитів зі збоями зберігаються в контейнері для зберігання бінарних об'єктів пиж-IIS-failedreqlogfiles. Створення трасування для запитів зі збоями є функцією IIS 7, що дозволяє котушка трассировочні події для запитів. Якщо запит завершився з помилкою, інформація про це зберігається в журналі невиконаних запитів. Підтримується можливість зміни даних, що зберігаються. Наприклад, можна задати діапазон кодів статусу HTTP (HTTP Status Code), які означають запит зі збоєм, наприклад, 400-599. Будь-який запит, що відповідає заданому критерію, вважатиметься запитом зі збоєм і інформація про нього буде занесена до відповідного журналу. Також можна задати тривалість часу, після закінчення якої необроблений запит буде вважатися збоєм. Нижче показаний приклад конфігурації, активізує збір інформації про запити зі збоями - в даному прикладі збоями вважаються запити, не оброблені протягом 10 секунд, а також запити з кодами статусу в діапазоні від 400 до 599.



**Журнал подій Windows**

У Windows Azure підтримується можливість створення і ведення журналу подій Windows,. Події Windows, містять важливу інформацію про стан програми та самої системи. Це означає, що можна конфігурувати додаток таким чином, що будуть збиратися дані не тільки про самому додатку, а й про екземпляр, в якому воно працює. Події Windows, повинні використовуватися для запису важливих подій на рівні додатку. Якщо потрібно збір інформації про виконання якихось етапів програми, методів і т.п. слід використовувати події групи і журнал трасування Windows Azure. Для того, щоб активізувати журнал подій Вікна, слід відповідним чином конфігурувати монітор діагностики. Нижче показано, як це зробити.

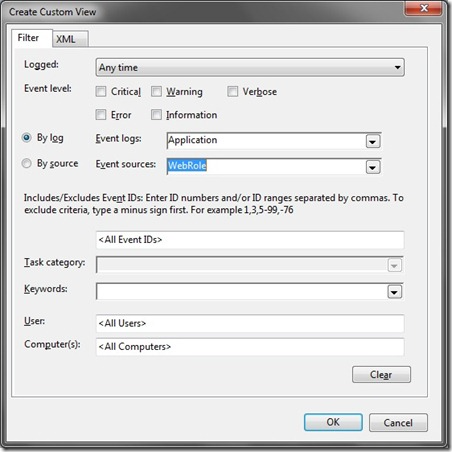
*var conf = DiagnosticMonitor.GetDefaultInitialConfiguration();*

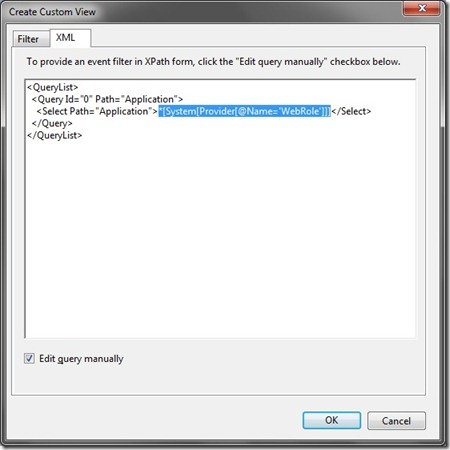
*conf.WindowsEventLog.DataSources.Add(“Application!\*[System[Provider [@Name=’WebRole’]]] ”);*

*conf.WindowsEventLog.ScheduledTransferPeriod = TimeSpan.FromSeconds(10);*

*conf.WindowsEventLog.ScheduledTransferLogLevelFilter = LogLevel.Undefined;*

У наведеному вище прикладі монітор діагностики конфігурується таким чином, щоб збирати дані з каналу Застосування для провайдера WebRole. Найбільш простий спосіб для створення рядка, необхідної для методу Add (із зазначенням каналу, провайдера, XPath-запиту і т.п.) - це використання програми перегляду подій Windows (% windir% \ system32 \ eventvwr.msc / с). У цій системній утиліті ми вибираємо режим Створити користувальницький вигляд (панель Actions) і копіюємо рядок з XML-подання (вкладка XML). Як це зробити показано на двох наступних ілюстраціях.





**Журнал лічильників продуктивності**

Windows Azure підтримуються лічильники продуктивності. Вони не активовані за умовчанням і потрібно вказати, які лічильники слід використовувати для збору даних. Лічильники слід використовувати для збору інформації про продуктивність програми, використанні ресурсів і т.п. Можна конфігурувати монітор діагностики для збору даних як від стандартних лічильників для Windows, так і від власних лічильників продуктивності. Нижче показано, як конфігурувати монітор продуктивності, щоб збирати дані від лічильника Процес% часу.

*configuration.PerformanceCounters.DataSources.Add(new PerformanceCounterConfiguration*

*{*

*CounterSpecifier = @”\Processor (\_Total)\% Processor Time”,*

*SampleRate = TimeSpace.FromSeconds(30)*

*}*

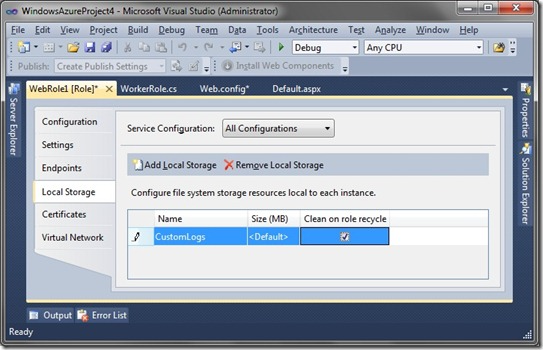
*);*

conf.PerformanceCounters.ScheduledTransferPeriod = TimeSpan.FromSeconds(60);

Для створення власного лічильника необхідно написати додаток (або скрипт на PowerShell), що створює такий лічильник і використовувати стартові завдання для завдання додатком необхідних привілеїв для реєстрації лічильника.

**Користувальницькі журнали**

Користувальницькі журнали можуть створюватися на основі будь-яких файлів в будь-якому заданому каталозі. Каталог повинен бути налаштований як локальне сховище для розгорнутого додатки. Типово, використовується локальне сховище з ім'ям DiagnosticStore. Для збереження вмісту користувальницьких журналів потрібно вказати ім'я контейнера для бінарних файлів. Конфігурація сховища може бути виконана в середовищі Visual Studio - нижче показано, як це зробити.



У наступному прикладі показано, як конфігурувати файли всередині локального сховища CustomLogs таким чином, щоб вони включалися до списку місць, з яких Windows Azure збирає і вивантажує дані.

*var res = RoleEnvironment.GetLocalResource(“CustomLogs”);*

*var dirConf = new DirectoryConfiguration*

*{*

*Container = “wad-customlogs-container”,*

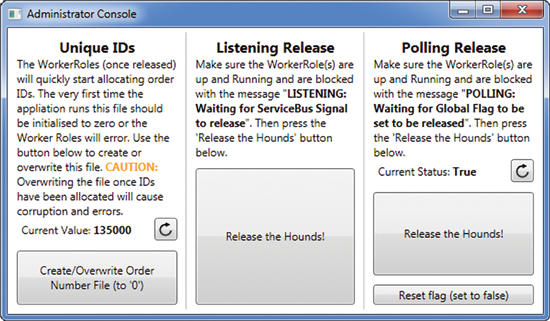
*DirectoryQuotaInMB = res.MaximumSizeInMegabytes,*

*Path = res.RootPath*

*};*

*conf.Directories.DataSources.Add(dirConf);*

**Консоль адміністратора**



Хоча підхід з ListeningRelease дійсно позбавляє від затримок, властивих класу PollingRelease, є й інші затримки, з якими теж треба якось боротися. Однак головний недолік підходу з прослуховуванням полягає в тому, що він не підтримує стану, через що будь-які підготовлені вузли після передачі оповіщення про запуск не побачили б це подія і залишилися б в загальмованому стані. Звичайно, очевидним рішенням могло б бути комбіноване використання як Service Bus, так і глобального прапора в сховище великих двійкових об'єктів.