#04

súbete a la nube de Microsoft

Diseño y herramientas

Ibón Landa Martín
Unai Zorrilla Castro

campus MVP .com
Plain Concepts
The Microsoft Technologies Company

Súbete a la nube de Microsoft

Parte 4: Diseño y herramientas

Ibón Landa Martín
Unai Zorrilla Castro

SÚBETE A LA NUBE DE MICROSOFT PARTE 4: DISEÑO Y HERRAMIENTAS

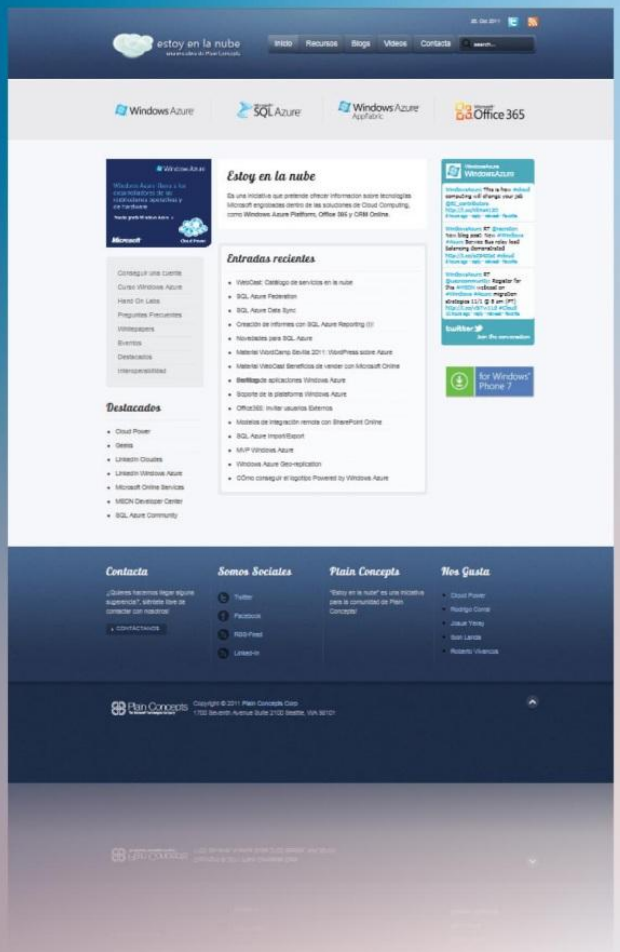
Diciembre de 2011

Contenido

estoy en la nube

INICIATIVA DE PLAIN CONCEPTS

www.estoyenlanube.com



Windows Azure

www.plainconcepts.com

Plain Concepts is a company specialized in Microsoft technologies, agile methodologies, Application Lifecycle Management, performance tuning, advanced debugging, software architecture and User Experience.

Plain Concepts focuses on delivering high quality consulting, mentoring and training as well as in being an effective and reliable team resolving all type of software development issues.

Plain Concepts

¿Aún quieres más?

campus MVP

Formación online especializada en tecnologías Microsoft.

krasis PRESS

Los libros que lo saben todo sobre tecnologías Microsoft.

Síguenos y descubrirás los mejores trucos y recursos:

facebook.com/campusmvp twitter.com/campusmvp

feed your brain®

- Sin tener que desplazarse
- Sin romper el ritmo de trabajo
- Preguntándole a los que más saben

infórmate ya:
902 876 475
www.campusmvp.com

http://www.krasis.com

krasis

Microsoft Partner
Silver Learning
Silver Software Development

CAPÍTULO

1

Consideraciones de diseño

1.- WINDOWS AZURE NO HACE MILAGROS

Uno de los aspectos clave que nos da la plataforma es la escalabilidad sin límites. Desplegando una aplicación en Windows Azure ésta podrá escalar hasta límites insospechados e inimaginables, y esto sin exagerar.

Claro está que la plataforma Windows Azure permite realizar aplicaciones que pueden ser altamente escalables, pero el límite está en la aplicación; si ésta no está preparada para escalar, Windows Azure no va a hacer milagros para que escale.

Por ejemplo, si se dispone de una aplicación web que no es capaz de desplegar en una batería de servidores web on-premise, difícilmente Windows Azure va a poder hacerlo.

Puede parecer una cosa sin sentido, pero más de una vez se olvida este punto y se piensa que por el hecho de emplear la plataforma o migrar una aplicación a Windows Azure se van a conseguir todos los beneficios que ésta puede llegar a ofrecer.

Pero hay un tema más básico. Windows Azure es una nueva plataforma dónde se pueden desplegar aplicaciones y que ofrece indudables beneficios sobre otras soluciones, pero esto no significa que los conocimientos anteriores dejen de ser válidos.

¿Cómo sé sabe cuántas instancias se tienen que levantar? ¿Cómo se hace que sea escalable? ¿Dónde se guarda la sesión? ¿Cómo se asegura que sea escalable?¿Qué arquitectura se debe usar?

Como plataforma nueva tiene sus peculiaridades y hay ciertos conocimientos que son diferentes, pero la esencia sobre cómo desarrollar una aplicación altamente escalable y qué conceptos arquitectónicos se deben tener en cuenta son iguales a los que se necesitarían para desarrollar una aplicación que vaya a desplegarse en un entorno on-premise.

Por ejemplo, a la pregunta ¿Cómo sé sabe cuántas instancias se tiene que desplegar? se puede responder con ¿Cómo se sabe ahora? ¿Cómo dimensionas una solución on-premise?

Con esto lo que se quiere comentar es que no debe olvidarse que muchos de los retos a los que los desarrolladores se enfrentan en Windows Azure, no dejan de ser los mismos a los que se enfrentaban antes de existir esta plataforma y que por tanto no se deeb hacer tabla rasa, ni pensar que por usar Windows Azure se vaya a tener por arte de magia aplicaciones escalables sin ningún tipo de esfuerzo.

Si no se ha dedicado el tiempo a hacer una buena arquitectura, si no se han aplicados los patrones y técnicas de desarrollo adecuadas, si no se ha pensado en ciertos aspectos clave sobre el desarrollo de aplicaciones.

2.- DISEÑO DE ARQUITECTURAS ESCALABLES CON AZURE

2.1.- Particionado de datos

Prácticamente cualquier aplicación hoy en día necesita disponer de algún sitio de almacenamiento. Dentro de la plataforma Azure existen dos alternativas, perfectamente compatibles, que son Windows Azure Storage y SQL Azure.

En este punto se abordará el concepto de particionado de datos, particionado que muchas veces es necesario del diseño de aplicaciones con un requisito de escalabilidad y cómo es posible dicha acción en los dos sistemas de almacenamiento.

Windows Azure Storage

El servicio de tablas de Windows Azure proporciona almacenamiento estructurado no relacional basado en tablas.

Una entidad dentro de las tablas de Windows Azure se define de manera única con dos propiedades.

La primera propiedad es la clave de la partición, PartitionKey, que identifica a que partición pertenece una entidad. La única garantía que existe sobre las particiones, es que, acceder a entidades almacenadas en la misma partición va a tener, típicamente, un menor coste que acceder a entidades en particiones diferentes.

La segunda propiedad es la clave de la entidad, EntityKey, que identifica de manera univoca una entidad dentro de una partición.

Figura 1.1 Particionado

Por tanto, definir correctamente el PartitionKey puede influir enormemente en el rendimiento de la aplicación, ya que este valor determina la ubicación física de una entidad. Es importante poner cuidado a la hora de elegir la clave de partición de las entidades para asegurar que entidades que se acceden típicamente al mismo tiempo comparten la misma clave de partición. También es importante que las particiones sean homogéneas en tamaño.

Por ejemplo, si se almacenan clientes, una posible clave de partición podría ser el código postal, si típicamente la aplicación trabaja con los clientes de un determinado código postal, por ejemplo para realizar estadísticas.

SQL Azure

El particionado es una técnica por la cuál se divide la información de una base de datos en subconjuntos más pequeños de información que distribuyen entre múltiples base de datos para mejorar aspectos como la manejabilidad, la disponibilidad o la escalabilidad.

Una de las ventajas del particionado es el posible aumento de rendimiento que puede obtenerse paralelizando tareas que usan diferentes servidores de datos, aunque también dispone de desventajas, como que aumenta el coste de mantenimiento y la complejidad de la aplicación, ya que en lugar de existir una base de datos existen varias.

Por ejemplo, en aplicaciones dónde pueda haber procesos de carga masiva el particionado puede ser una alternativa, ya que podrían paralelizarse las tareas a realizar.

Otro escenario posible puede ser en aquellas aplicaciones que puedan necesitar más del almacenamiento permitido por SQL Azure, más de 50 Gb. En este caso, si se desea utilizar SQL Azure como sistema de almacenamiento, sería necesario utilizar un sistema de particionado.

Existen diversas técnicas de particionado pero básicamente el particionado puede realizarse de forma horizontal o vertical.La elección a elegir depende de la aplicación y de la forma en que ésta consultará la información.

En cualquier de los dos casos, el proceso de particionado no es un proceso transparente para la aplicación y éste debe conocer el particionado para poder realizar las consultas necesarias.

El particionado vertical lo que hace es dividir una tabla en función de sus columnas. Una tabla con múltiples columnas se divide y se llevan unas determinas columnas a una tabla y otras a otra, relacionado toda la dos a través de la clave primaria.

Otra alternativa de particionado vertical es dividir la base de datos por las tablas, llevando unas tablas a una determinada base de datos y otras tablas a otra.

Por ejemplo, si un cliente tiene varias direcciones de correo, la tabla se puede dividir poniendo la dirección principal de una tabla y el resto de direcciones en otra tabla. La mayoría de las consultas emplearán la tabla de direcciones principal y otras pocas consultas requerirán del resto de direcciones.

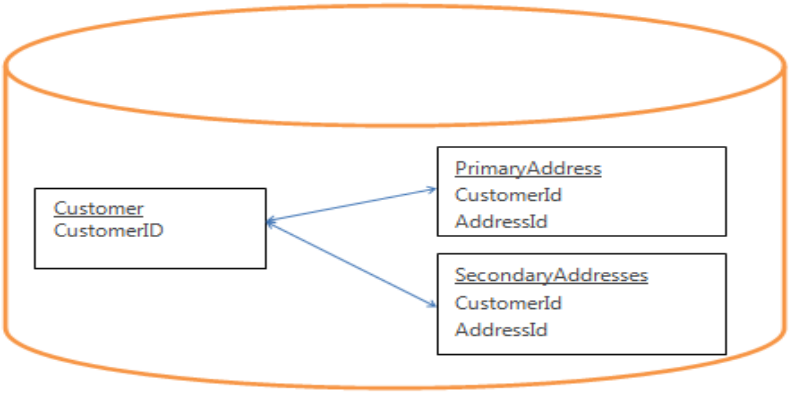


Figura 1.2 Particionado vertical

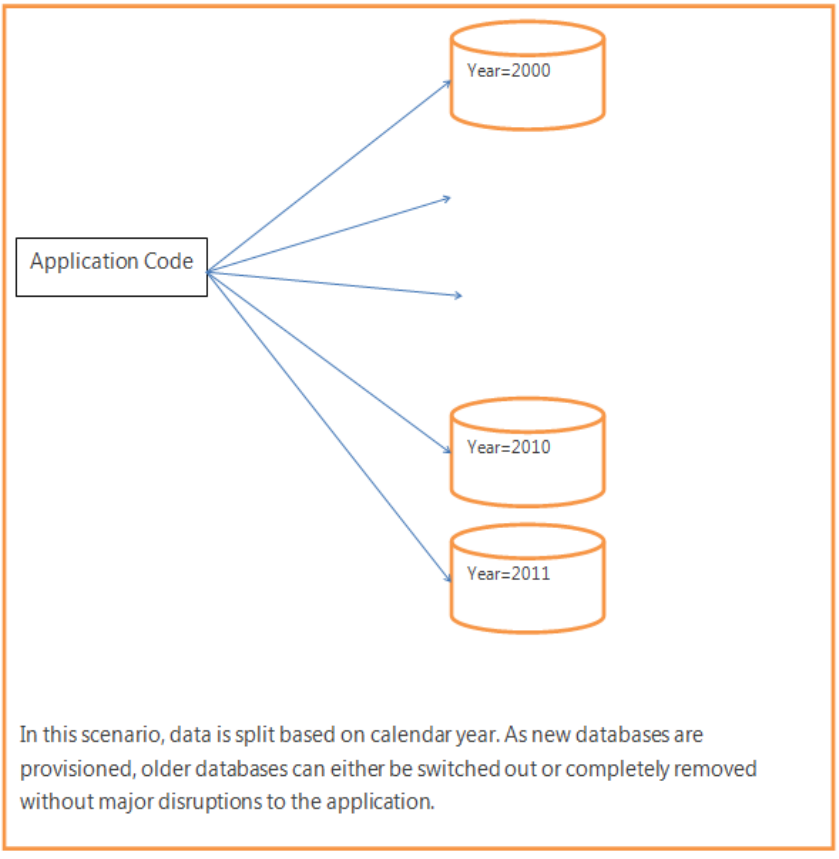


Figura 1.3 Particionado vertical

El particionado horizontal divide las tablas a través de una determinada clave. Existirá una tabla maestra para poder conocer dónde se encuentra la información asociada a una determina clave.

Habitualmente este es el sistema empleado cuando quiere dividirse la información entre múltiples instancias de base de datos SQL Azure. En este caso el schema de la base de datos es el mismo en todas las instancias existentes y lo que varía es la información que residen en ella.

A continuación se muestra un ejemplo de un escenario dónde la información se divide en múltiples base de de datos en función del año. Debe existir una tabla maestra que indique para cada año dónde está su información.

2.2.- Estado de las aplicaciones

Guardar el estado de las aplicaciones y la forma en que éste se almacena es uno de las decisiones habituales que es necesario tomar a la hora de definir la arquitectura de la aplicación.

En escenario de escalabilidad existiría múltiples instancias por cada rol desplegado, lo que implica tener que valor cómo se va a guardar esta información, dónde se almacenará y cómo se pude conseguir que se comparta entre las diferentes instancias o roles.

Una de las alternativas es almacenar la información de forma centraliza empleando las tablas de Windows Azure Storage.

Para poder disponer de esta funcionalidad dentro las aplicaciones es necesario realizar modificaciones en el fichero de configuración, modificando la configuración para emplear un proveedor personalizado que permita guardar la información de la sesión en storage.

```
<system.web>

<sessionState mode="Custom" customProvider="TableStorageSessionStateProvider">

<providers>

<clear/>

<add name="TableStorageSessionStateProvider"
     type="Microsoft.Samples.ServiceHosting.AspProviders.TableStorageSessionStateProvider"
     applicationName="myWebAppName"/>

</providers>

</sessionState>
```

En el training kit de Windows Azure se encuentra disponible el código fuente para el proveedor que posibilidad guardar la información de la sesión en el storage.

Es importante tener en cuenta que usar el storage con este fin implica un coste adicional para la aplicación, por lo que no sólo en términos de rendimiento sino también en términos de costes es importante reducir al máximo la información a almacenar en la sesión.

2.3.- Distribución de cargas

Windows Azure provee de una serie de servicios que hace que los usuarios de la plataforma puedan olvidarse de algunos concepts básicos de los cuáles deberían preocuparse en soluciones desplegadas sobre servidores propios.

- Gestión del Sistema Operativo sobre el que se ejecutan las aplicaciones y servicios.
- Configuración de servicios "base": S.O., servidor de datos, Servidor de aplicaciones...
- Gestión de actualizaciones y parches de los servicios "base"
- Diagnóstico de fallos de servicios "base"

- Respuesta a fallos de hardware
- Disponibilidad de capacidad de almacenamiento adecuada, lo que puede llegar a ser muy complejo.
- Monitorización
- Respuesta ante desastres
- Configurar balanceado de carga
- Gestionar incrementos súbitos de tráfico y, por ende, potencia de proceso.
- Etc...

Desde el punto de vista del diseño de una aplicación escalable, interesa conocer en profundidad las opciones que la plataforma ofrece para poder conseguir este objetivo.

Uno de los principales componentes de la plataforma es el Windows Azure Fabric. "El Fabric", como se conoce familiarmente, es el componente de la arquitectura que se encarga de proporcionar los servicios base de Windows Azure de manera transparente respecto a la infraestructura IT subyacente. El desarrollador no sabrá nunca en que máquina concreta del centro de datos de Microsoft se está ejecutando la aplicación Azure.

El Fabric se encarga de asegurar que la aplicación recibe tiempo de ejecución, ancho de banda y recursos en general para su ejecución, balanceando la carga a las máquinas virtuales que considere necesario de manera trasparente para la aplicación. Además orquesta para la aplicación el acceso a los recursos de almacenamiento y colas de la plaforma Windows Azure, siendo todo ello algo de lo que no se debe preocupar el desarrollador.

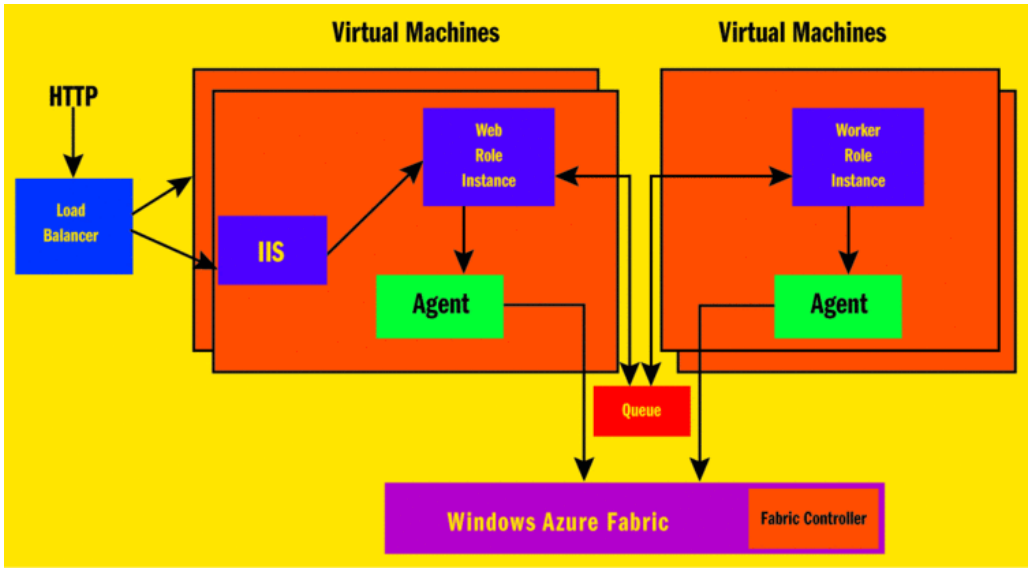


Figura 1.4 Diagrama básico de la arquitectura de Windows Azure

Para sistema de alta escalabilidad Windows Azure permite la posibilidad de añadir el número de instancias de una aplicación casi de forma inmediata, sin que este cambio implique realizar un nuevo despliegue. Añadir una instancia es tan sencillo como cambiar un fichero de configuración.

En el momento de desplegar la aplicación es posible configurar el número de instancias a desplegar, estableciendo dicho valor en el fichero de configuración. El Fabric es capaz de interpretar la configuración y desplegar tantas instancias como se hayan configurado.

Del mismo modo será responsabilidad del Fabric distribuir la carga entre las diferentes instancias de la aplicación.

Una vez desplegada la aplicación, ya sea desde el portal de Windows Azure o desde herramientas que utilicen el API de administración, es posible cambiar el número de instancias de forma casi inmediata.

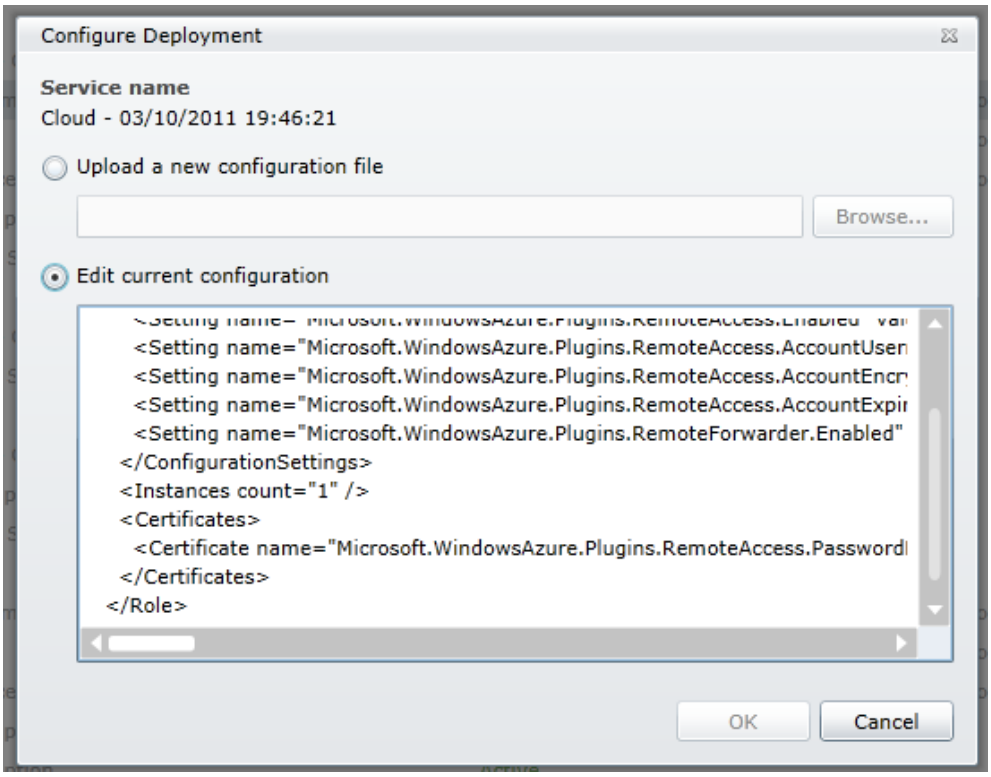


Figura 1.5 - Actualización de la configuración

El AppFabric también tiene la responsabilidad de asegurarse de que las instancias de la aplicación estén siempre accesibles y disponibles para atender peticiones.

AppFabric dispone de una serie de agentes que se encargan de monitorizar el estado de las diferentes instancias de la aplicación. Si detecta que una instancia no está funcionando correctamente elimina la instancia y añade una nueva que pueda atender las peticiones.

Por este motivo, es conveniente también disponer de más de una instancia por cada rol, para que en caso de que falle una instancia siempre haya otra que esté disponible mientras el Fabric "soluciona" el problema.

Para detectar el estado de las aplicaciones AppFabric dispone de una serie de procedimientos y reglas que debe cumplir una aplicación, información que está disponible dentro de la documentación de la plataforma. Adicionalmente, también puede configurarse la aplicación desplegada para que ofrezca información sobre su estado al Fabric y que éste pueda actuar teniendo en cuenta el estado reportado por la aplicación.

```
public override RoleStatus GetHealthStatus()
{
      // return the health status of worker role.
      return RoleStatus.Healthy;
 }
```

2.4.- Procesamiento asíncrono de mensajes

Un escenario típico en el desarrollo de servicio de alta disponibilidad es la inclusión de sistemas asíncronos de mensajes que puedan ayudar a la aplicación a poder atender siempre las peticiones de los clientes.

Si dentro de Windows Azure queremos disponer de esta funcionalidad será necesario emplear el sistema de colas que ofrece Windows Azure Storage. El servicio de colas de Windows Azure proporciona un mecanismo fiable y persistente para la comunicación entre aplicaciones de Windows Azure.

Para poder interactuar con los clientes que residen fuera de la nube será necesario disponer de un Web Role capaz de atender las peticiones. Éste componente atenderá las peticiones y las incluirá dentro del sistema de colas de Azure.

Un Worker Role será en encargado de revisar de forma periódica los mensajes de la cola, leerlos y procesarlos.

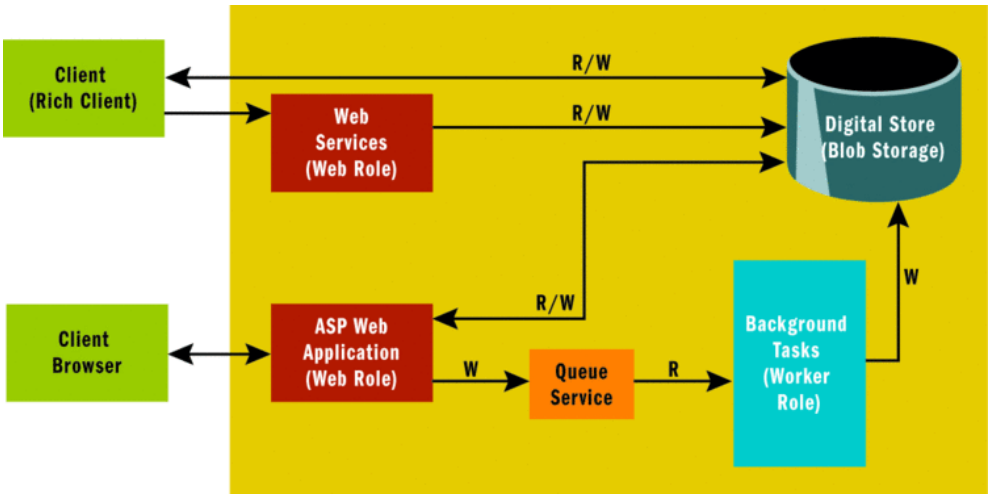


Figura 1.6 - Arquitectura de una aplicación Web basada en Windows Azure

Cada cuenta de almacenamiento puede contener un número ilimitado de colas de mensajes y cada cola puede contener un número de mensajes ilimitado. La principal limitación en las colas de Windows Azure viene marcada por el hecho de que el tamaño máximo de mensaje es 8Kb.

Cuando un mensaje se lee desde una cola, el consumidor del mensaje es responsable de, tras procesarlo, eliminarlo. Una vez el mensaje es leído, durante un periodo de tiempo no estará disponible para otros consumidores. Si el mensaje no es borrado en ese intervalo de tiempo, su visibilidad es restablecida de manera que otros consumidores pueden procesarlo

3.- CONSEJOS PRACTICOS PARA CREAR APLICACIONES SEGURAS EN AZURE

Cuando se trata de soluciones en la nube, es muy importante para los arquitectos y desarrolladores de software poder anticiparse a las amenazas en tiempo de diseño, más aún que en el caso de desarrollar aplicaciones que vayan a ser desplegadas en servidores corporativos. En este punto se ofrecerán algunas consideraciones a tener en cuenta para intentar mitigar algunas posibles amenazas para las aplicaciones que de despliegen en la nube.

El abanico de posibles amenazas de un servicio basado en la nube es sustancialmente diferente de los tradicionales servicios desplegados en un servidor corporativo en términos de técnicas de mitigación y tecnologías. Las amenazas también varían dramáticamente entre los proveedores de nube.

Los puntos clave a tener en cuenta son:

- Los desarrolladores deben relacionar los tradicionales requisitos de seguridad de las soluciones tradicionales con los requisitos de las aplicaciones desplegadas en Windows Azure. Cualquier amenaza deberá ser mitigada por la aplicación o servicio.
- Comprender los requisitos de seguridad del servicio que se diseña o se migra, sobre todo en el contexto de la autenticación, autorización y auditoría. Lo servicios de la plataforma que ofrecen estas características (Windows Identity Foundation, Windows Azure AppFabric Servicio de Control de Acceso y diagnóstico de Windows Azure) y la forma de utilizarlo por parte de los desarrolladores es sustancialmente diferente de las previstas en una implementación empresarial on-premise (Kerberos, Active Directory, los registros de sucesos de Windows).
- Aprovechar los servicios de Windows Azure plataforma para construir aplicaciones más seguras.

Es necesario resaltar, que aunque la plataforma ofrezca servicios para poder mejorar la seguridad de las aplicaciones desplegadas en la nube, el desarrollador sigue siendo el responsable de crear código de buena calidad y de ser responsable de seguir las recomendaciones de seguridad para el desarrollo de aplicaciones y protegiéndose de las posibles amenazas; validar las entradas, incluir restricciones en la entrada de valores etc...

3.1.1.- Consideraciones en la configuración del espacio de nombres

En este punto se detallan algunas consideraciones a tener en cuenta en la configuración de aplicaciones a desplegar en la nube.

Evitar usar nombres como <servicename>.cloudapp.net. Es conveniente utilizar un nombre personalizado para el nombre del dominio en lugar del nombre del servicio.

El espacio de nombres cloudapp.net se comparte entre todos los clientes de Windows Azure Platform mientras que un espacio tradicional, como microsoft.com, es controlado únicamente por la empresa propietario del dominio. Esto significa que el espacio de nombre cloudapp.net ofrece menos nivel de confianza que es un espacio de nombres tradicional debido a que un cliente de Windows Azure no tiene por qué confiar automáticamente en el resto de clientes de ese espacio de nombres. No debe crearse código que requiere añadir el espacio de nombres "cloudapp.net" en la lista de sitios de confianza del navegador web.

Nunca utilizar cookies con el espacio de nombres cloudapp.net. En su lugar es mejor utilizar un subdominio, por ejemplo miaplicacion.cloudapp.net. El espacio de nombres de común para todos los clientes de Windows Azure.

Para la mayoría se contenidos, sólo se permite la interacción con contenido dentro de mismo dominio. Desde un página que está contenida dentro de un dominio puede accederse sin problemas al contenido de otra página que está dentro del mismo dominio, empleando scripting.

El modelo de objetos DHTML permite a través de document.domain chequear y configurar dicha restricción. Sólo páginas con la misma configuración de las propiedades de dominio pueden interaccionar entre ellas. Además de la configuración, el protocolo empleado debe ser el mismo. Por ejemplo, desde HTTP no puede acceder a contenido que está en HTTPS.

Este funcionamiento es importante tenerlo en cuenta si en algún momento se modifica y se amplia el acceso a document.domain, ya que una mala configuración puede dejar abierta la posibilidad de que cualquier aplicación de Windows Azure pueda acceder al contenido de la aplicación. IIS 7 debe acceso por defecto a todo el subdominio.

3.1.2.- Seguridad de datos

Cuando se diseña la relación entre un Web Role y Windows Azure Storage el uso de firmas de acceso compartido (Shared Access Signatures) es una herramienta potente que permite securizar el acceso al contenido del storage y de ofrecer acceso sólo a aquellas aplicaciones que deban tenerlo. Si el contenido de Windows Azure Storage no se securiza convenientemente éste será de acceso público. Cualquier aplicación o cliente, ya esté en la nube o en otra ubicación.

Las firmas de acceso compartido representan un token de seguridad que ofrece permisos de acceso a contenedores y blobs de Windows Azure Storage que son de acceso privado. Los contenedores y blobs de acceso público son accesibles por cualquier aplicación o cliente, ya esté en la nube o en cualquier otra ubicación.

Si la información no debe ser pública, es conveniente establecer correctamente la seguridad de acceso a los datos empleando estas claves que actúan como token de seguridad. Es necesario poder conocer la firma para poder acceder al contenido.

La aplicación que crea el contenido es la responsable de establecer la información de seguridad y de distribuir convenientemente el token de seguridad a los clientes que necesiten acceso a la información. Si un token se ve comprometido es posible cambiar dicho token e invalidar el existente.

Por este motivo, es conveniente diseñar correctamente la política de seguridad de acceso a los datos albergados en Windows Azure Storage y de proteger aquel contenido que sea de especial relevancia.

Es conveniente generar token de seguridad (Shared Access Signatures) lo más restrictivas posibles y si es posible, que dispongan de un período corto de vida.

Es conveniente emplear siempre comunicación HTTPS para intentar evitar que las firmas se vean comprometidas.

Y por último, es importante recordar que estos tokens de seguridad deben usarse únicamente para acceder de manera temporal al contenido privado de los blobs y que es una práctica poco recomendada emplear siempre los mismos tokens.

Windows Azure Table ofrece un entorno para almacenar información estructura no-SQL, por lo que no es vulnerable a las amenazas de SQL Injection típicas de los sistemas SQL. La aplicaciones que empleen SQL Azure como sistema de almacenamiento sí que deberán tener en cuenta las mismas amenazas que podría haber en una aplicación que emplee SQL Server.

Las peticiones a las tablas del storage puede hacer por HTTP GET o HTTP POST. Es conveniente tener en cuenta que estas peticiones podrían llegar a ser modificadas de manera malintencionada, haciendo modificaciones en las información enviada en la petición HTTP. Por este motivo es conveniente no incluir información sobre los nombres de contenedores, blogs, identificativos o nombre de tablas.

3.1.3.- Almacenamiento de información secreta

Durante la realización de este contenido la plataforma Windows Azure no dispone de un soporte para Data Protection API (DPAPI) para poder encriptar la información almacenada en Windows Azure Storage.

Si por ejemplo se desea encriptar un blob, debe ser la aplicación que suba el blob al contenedor la encargada de encriptar la información antes de subir la información a Windows Azure Storage, empleando algunas de las técnicas que puedan existir actualmente.

Los desarrolladores debe preocuparse de no almacenar las claves privadas en Windows Azure Storage, ya que éstas podrían ser accesibles por terceros.

3.1.4.- Auditoría y registro de sucesos

En Windows Azure no se dispone de acceso a recursos locales del sistema de archivos ni al log de eventos de Windows.

El acceso local al disco de una instancia de Windows Azure debe considerarse siempre como un acceso temporal y no persistente y no debería emplearse nada más que para almacenar ficheros temporales o a modo de caché.

Esta situación plantea la necesidad de tener un mecanismo diferente a los tradicionales para escribir las trazas de la aplicación. Algo que permita detectar y diagnosticar comportamientos anómalos de la aplicación cuando ésta se encuentra desplegada en la plataforma Windows Azure.

El SDK de Windows Azure proporcionar un 'trace listener' especialmente diseñado para ser utilizado en aplicaciones en la nube. Este listener está implementado en la clase DiagnosticsMonitorTraceListener del namespace Microsoft.WindowsAzure.Diagnostics y deriva de la clase TraceListener del framework de .Net. Trabajar con este listener es, por tanto, idéntico a trabajar con cualquier otro trace listener de .Net.

Como el almacenamiento local no es posible existe la posibilidad de almacenar las trazas de la aplicación en el almacenamiento de Window Azure.

Windows Azure no soporta actualmente la opción de encriptar el contenido de las tablas de Windows Azure, por lo que es importante que el desarrollador no escriba información sensible en la información de diagnóstico, ya que esta se almacenará en el Windows Azure Storage.

El desarrollador podrá elegir entre HTTP o HTTPS. La recomendación por defecto es emplear HTTPS, aunque debe tenerse en cuenta que el rendimiento ofrecido por este protocolo es menor que el ofrecido por HTTP, consideración que debe tenerse en cuenta si la cantidad de información que se escribe es alta.

3.1.5.- Patrón de diseño seguro “GateKeeper”

Gatekeeper es un patrón de diseño que define la forma de acceder a un sistema de almacenamiento con el objetivo de minimizar el área de ataque. El acceso al almacenamiento sólo puede accederse de forma privada y a través de canales privados, no pudiéndose hacer desde aplicaciones externas.

Si una aplicación externa desea acceder a la información debe acceder a través de un rol que actúa como "broker". El punto de entrada a la información es rol que actúa como broker y que residen en otra máquina virtual.

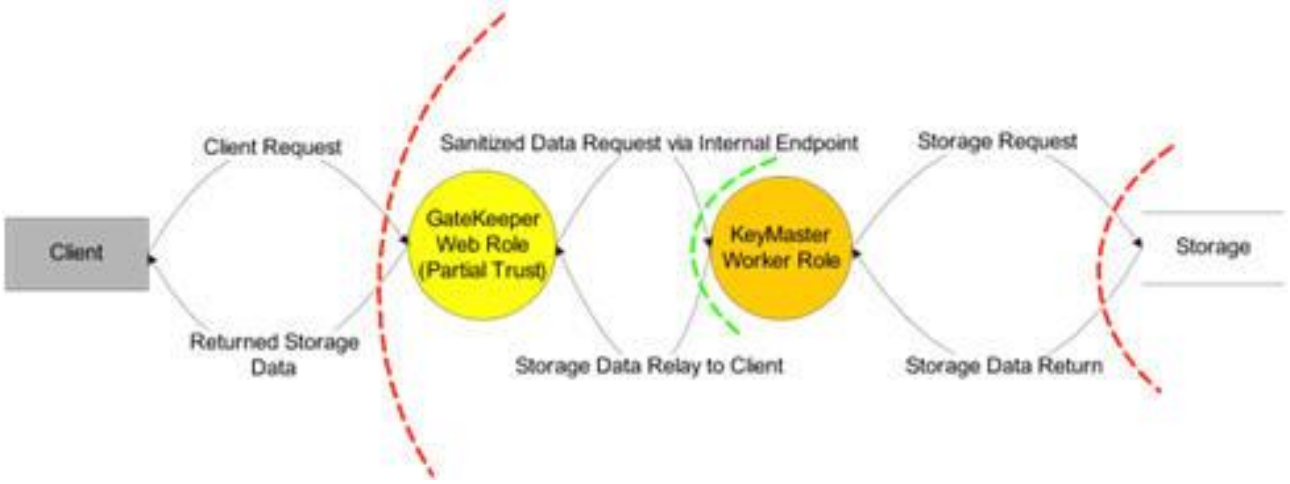


Figura 1.7.- Patrón GateKeeper

El GateKeeper es un web role que recibe las peticiones que vienen desde Internet, peticiones que son potencialmente peligrosas y en las cuáles el GateKeeper no puede confiar. El GateKeeper se implementa en código manejado y funciona con Partial Trust. La configuración de este rol no debe contener ninguna información secreta ni clave de acceso a ningún recurso.

El KeyMaster es un worker rol que sólo permite comunicaciones internas dentro de Windows Azure, que por tanto, no puede recibir peticiones desde Internet. El KeyMaster recibe las peticiones que el GateKeeper le realiza a través de HTTPS y es capaz de interactuar con el sistema de almacenamiento para atender las peticiones del GateKeeper.

El KeyMaster confía en las peticiones del GateKeeper y en su configuración dispone de la información necesaria para conectarse al sistema de almacenamiento.

3.2.- Partial Trust y full trust: Qué puede y que no puede hacer una aplicación Windows Azure

Aunque la experiencia de desarrollo de una aplicación de Windows Azure es muy similar a desarrollar una aplicación web en la plataforma .NET, la naturaleza compartida de los recursos en la nube exige establecer ciertas limitaciones al comportamiento de las aplicaciones desplegadas en Azure. Dichas limitaciones están orientadas especialmente a evitar que, intencionadamente o por error, las aplicaciones de Windows Azure tengan comportamientos anómalos, traten de interferir las operaciones de otras aplicaciones, realicen ataques a maquinas en Internet, etc...

La plataforma .NET cuenta con un mecanismo llamado Code Access Security (CAS) que permite delimitar perfectamente que acciones puede llevar a cabo una aplicación .NET. Este mecanismo es aprovechado por Window Azure para establecer límites a lo que pueden hacer las aplicaciones.

Aprovechando las capacidades de CAS, Windows Azure define dos niveles de confianza para las aplicaciones: 'partial trust' y 'full trust.

El nivel de confianza 'Partial Trust'

Este es el nivel de confianza que por defecto tendrá todo rol que creemos. Este es el nivel de confianza recomendado para la mayoría de aplicaciones. En este nivel de confianza se tienen las siguientes limitaciones:

- Se pueden realizar casi todas las acciones en ASP.NET que se pueden realizar en ASP.NET con 'medium trust'. Este es el nivel habitual en todos los hosting compartidos de ASP.NET. La mayor dificultad es que algunos controles de terceros mal diseñados exigen 'full trust' para funcionar correctamente.
- Se pueden realizar consultas DNS.
- Solo se puede leer y escribir las variables de entorno TMP y TEMP, pero ninguna otra.
- No se tiene acceso al log de eventos.

- Solo se puede escribir y leer archivo en el almacenamiento local de Windows Azure, pero no en ninguna otra ubicación del disco. Típicamente sólo se utiliza el almacenamiento local de Windows Azure como cache.
- No se dispone de acceso de ningún tipo a la característica de almacenamiento aislado de .NET.
- No se pueden realizar peticiones por OleDb.
- Solo algunas de las características de reflexión están disponibles.
- Solo se pueden abrir sockets de salida, sobre TCP y hacia sitios externos.
- Solo se pueden realizar conexiones a fuentes SQL externas.
- El nivel de confianza 'Full Trust'
- El nivel de confianza 'full trust' confiere a la aplicación de Windows Azure privilegios adicionales. De estos privilegios adicionales los mas notables son:
- La capacidad para poder llamar a código nativo mediante P/Invoke.
- La capacidad para poder usar ensamblados .NET que requieran 'full trust', esto es, todos aquellos que no estén marcados con el atributo AllowPartiallyTrustedCallers.
- La capacidad para leer, que no escribir, en el registro de Windows de la máquina sobre la que se está ejecutando el Fabric de Windows Azure.

Para configurar un rol de Azure para que se ejecute con 'full trust' en el archivo de definición del servicio (.csdef) se debe establecer el atributo enableNativeCodeExecution del rol correspondiente a true, tal y como se puede ver en el siguiente código:

```
<WorkerRole name="Role" enableNativeCodeExecution="true">
<ConfigurationSettings />
</WorkerRole>
```

CAPÍTULO 2

Herramientas

1.- WINDOWS AZURE PLATFORM HEALTH DASHBOARD

A través de portal de la plataforma Windows Azure es posible ver un cuadro de mando que ofrece información sobre el estado de los diferentes componentes de la plataforma.

La herramienta permite subscribirse a un feed RSS para poder recibir información sobre el estado de los componentes y/o servidores de la plataforma Windows Azure y de esta manera poder estar informados de cualquier contratiempo.

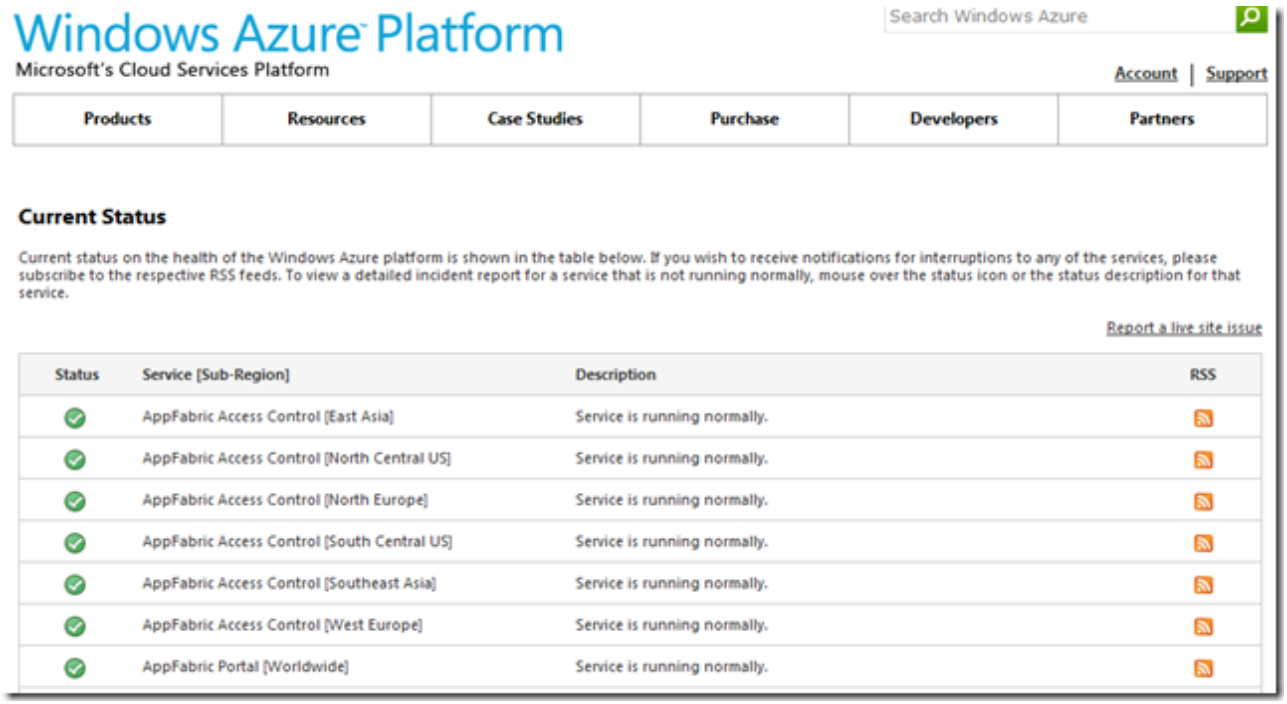


Figura 2.1. - Windows Azure Dashboard

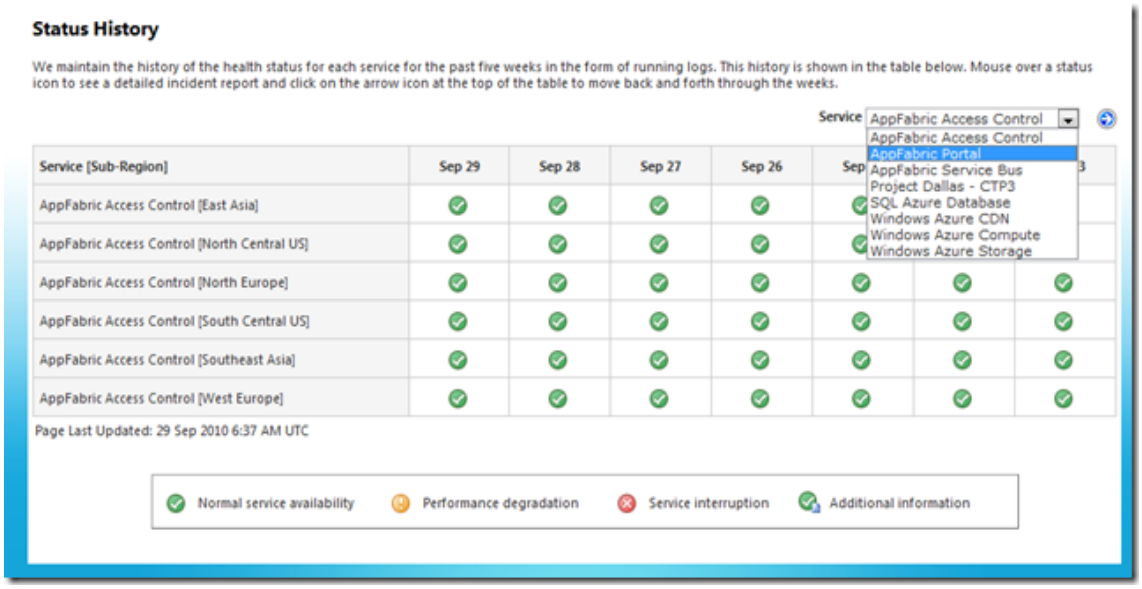


Figura 2.2. - Windows Azure Dashboard

2.- AZURE THROUGHPUT ANALYZER

Azure Throughput Analyzer es una herramienta de Microsoft Research, una aplicación de escritorio, cuyo objetivo es monitorizar y medir el rendimiento de subida y bajada entre una máquina de un cliente y el almacenamiento de Windows Azure (blobs, tablas o colas).

Esta utilidad puede ser muy útil para medir la velocidad de acceso a los datos y ver las diferencias entre los diferentes Data Centers de Windows Azure. La aplicación realiza diferentes subidas y bajadas con datos de ejemplo para poder medir el rendimiento que ofrece el Data Center.

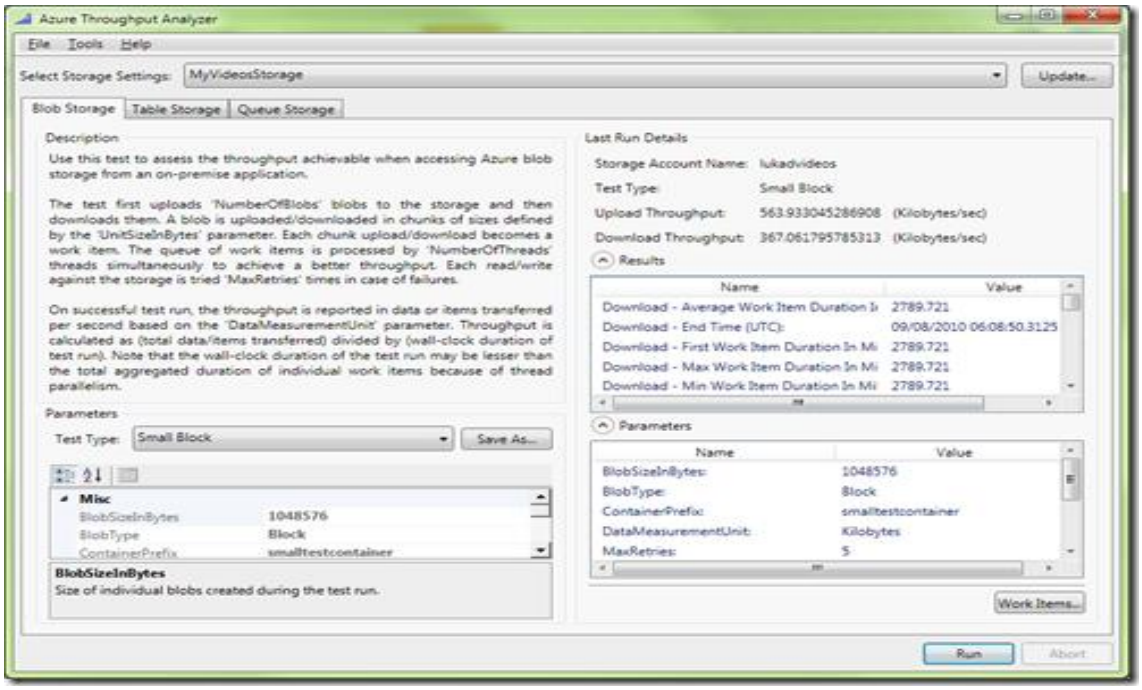


Figura 2.3.- Azure Throughput Analyzer

3.- WINDOWS AZURE MMC

Windows Azure Management Tool es una herramienta gratuita diseñada y creada para poder administrar los servicios y el sistema de almacenamiento de una cuenta de Windows Azure.

Esta herramienta provee de interesantes características que pueden simplificar enormemente el trabajo con Windows Azure:

- Permite realizar despliegues de servicios, actualizarlos y administrarlos.
- Permite configurar el sistema de diagnóstico de servicios hosteados en Windows Azure.
- Permite administrar los certificados disponibles en los servicios.
- Permite configurar las cuentas de almacenamiento.
- Permite la gestión de Blobs de Windows Azure; crear y eliminar contenedores, subir, eliminar o descargar blobs.
- Permite la gestión de Tablas de Windows Azure; consultar y borrar colas.
- Permite la gestión de las colas de Windows Azure; crear colas, purgarlas y borrar mensajes.

A continuación se muestran algunas imágenes de cómo es la herramienta.

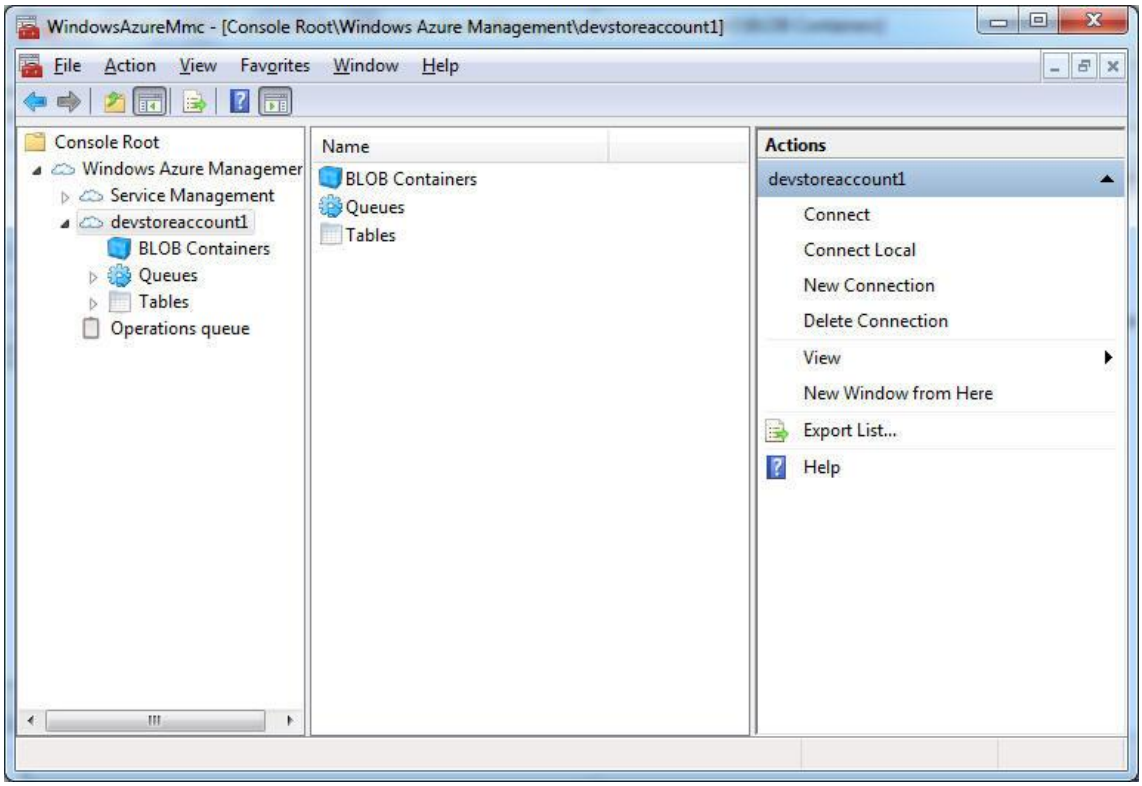


Figura 2.4.- Windows Azure MMC

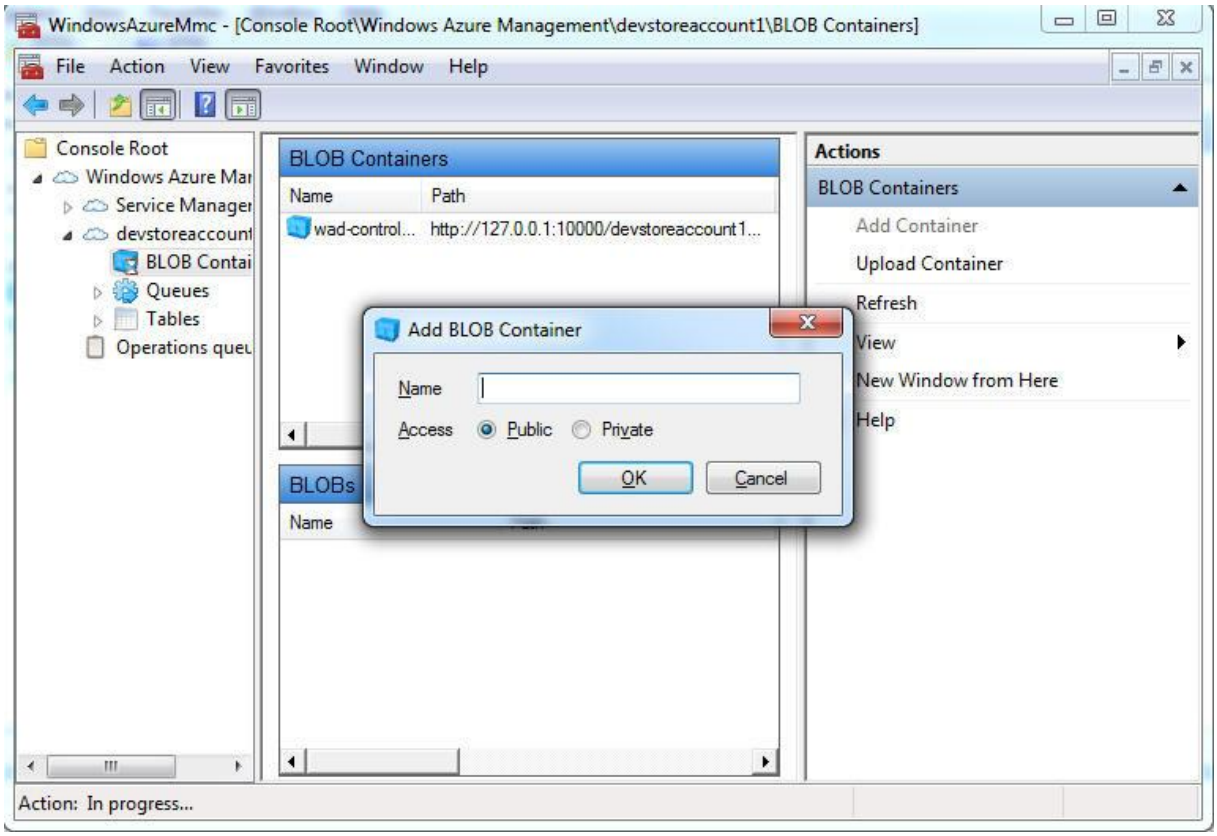


Figura 2.5.- Crear un contenedor

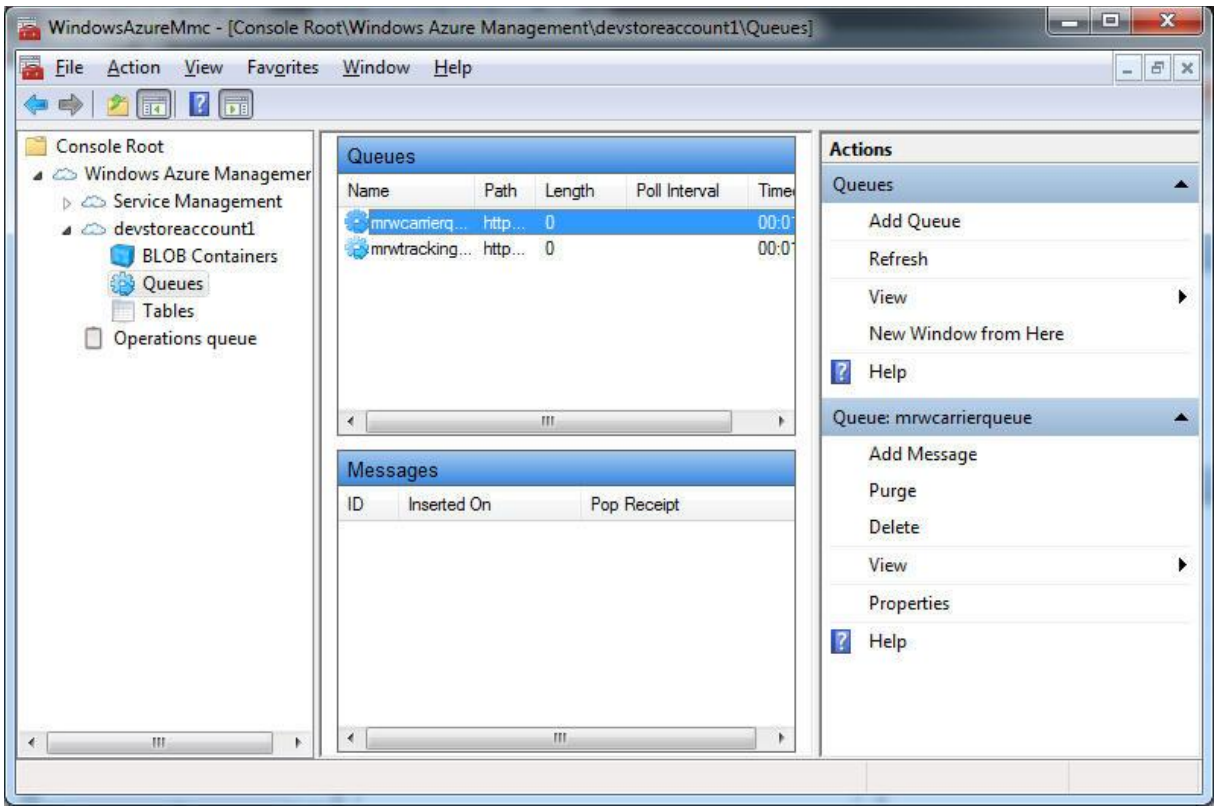


Figura 2.6.- Visualizar el contenido de las colas

4.- CLOUD STORAGE STUDIO

Cloud Storage Studio es una herramienta desarrollada por la empresa Cerebrata que permite trabajar de forma cómodo tanto con Windows Azure Storage como con las aplicaciones hosteadas en Windows Azure.

Cloud Storaga Studio es una herramienta que requiere disponer de una licencia y que ofrece una variedad de funcionalidad relacionada con Windows Azure:

- Permite conectarse a una cuenta de Windows Azure Storage y administrar las tablas, blobs y colas.
- Permite conectarse y administrar el Development Storage.
- Permite crear tablas, eliminarlas, crear entidades, consultarlas…
- Permite gestionar contenedores de blobs, crear blobs; crear, eliminar, modificar, copiar, renombrar etc…
- Administrar colas de Azure; crear, actualizar, borrar etc…
- Administrar aplicaciones hosteadas;
- Hacer despliegues, actualizarlos, borrarlos, ver el estado de un servicio etc...
- Desplegar en producción, preproducción, hacer paso de preproducción a producción.
- Hacer actualizaciones de los servicios; swaping, upgrades manuales, upgrades automáticos…

A continuación se muestran algunas imágenes de cómo es la herramienta.

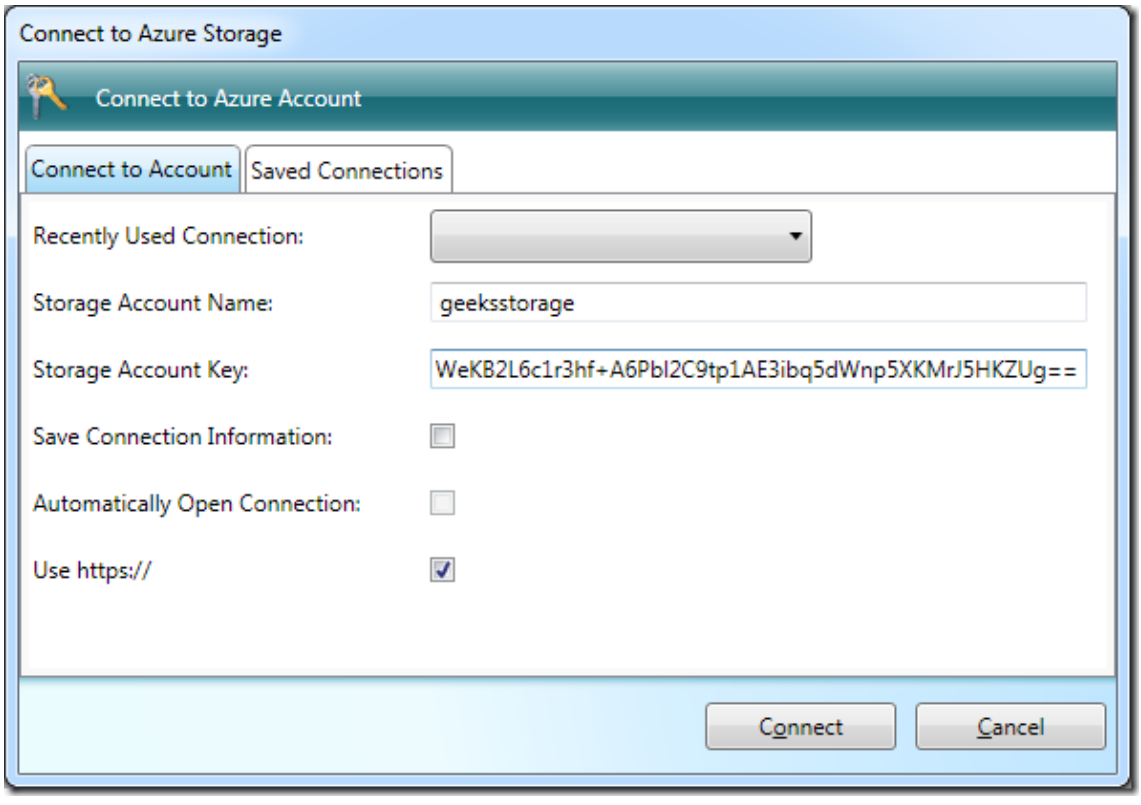


Figura 2.7- Pantalla de conexión al servicio de almacenamiento Azure

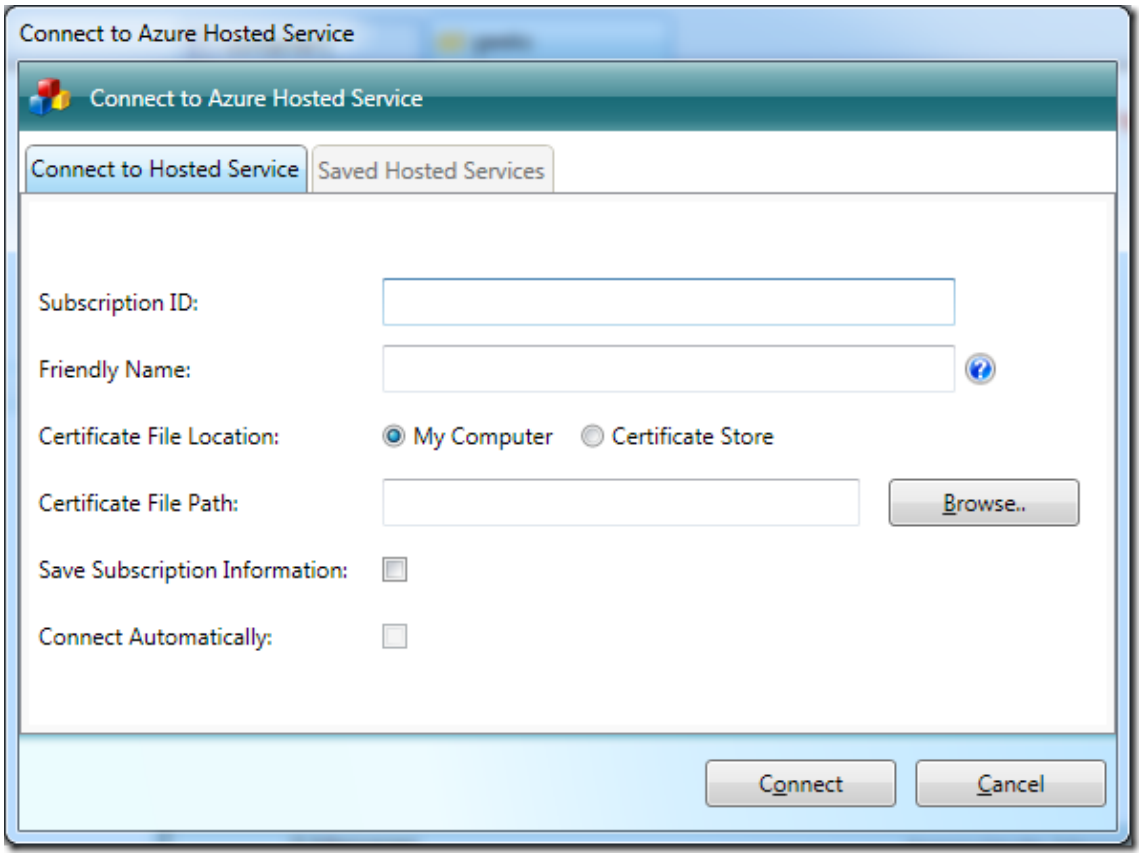


Figura 2.8.- Pantalla de conexión al sistema de ejecución de Windiws Azure

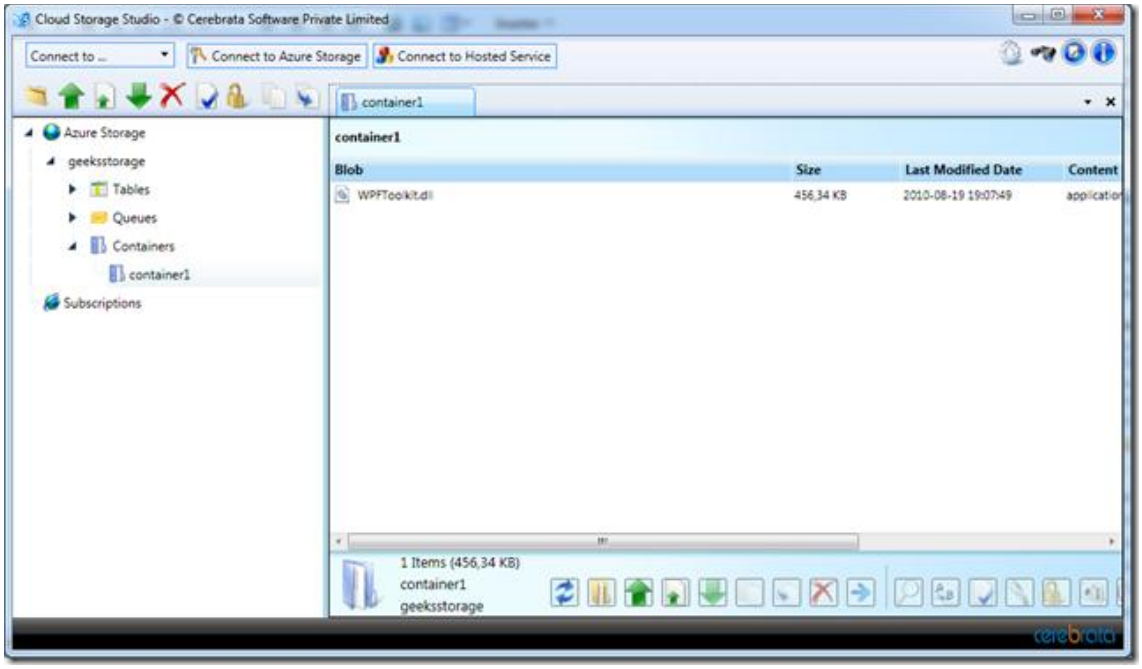


Figura 2.9.- Pantalla de visualización de los componentes de Windows Azure Storage.

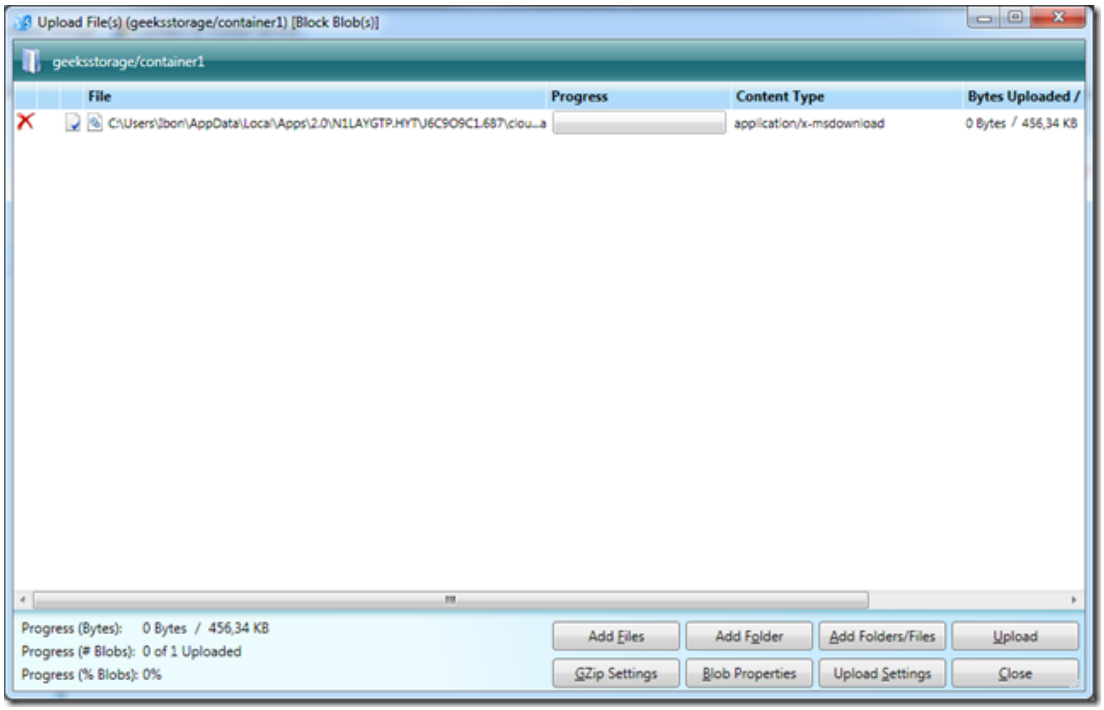


Figura 2.10.- Pantalla de subir un Blob.

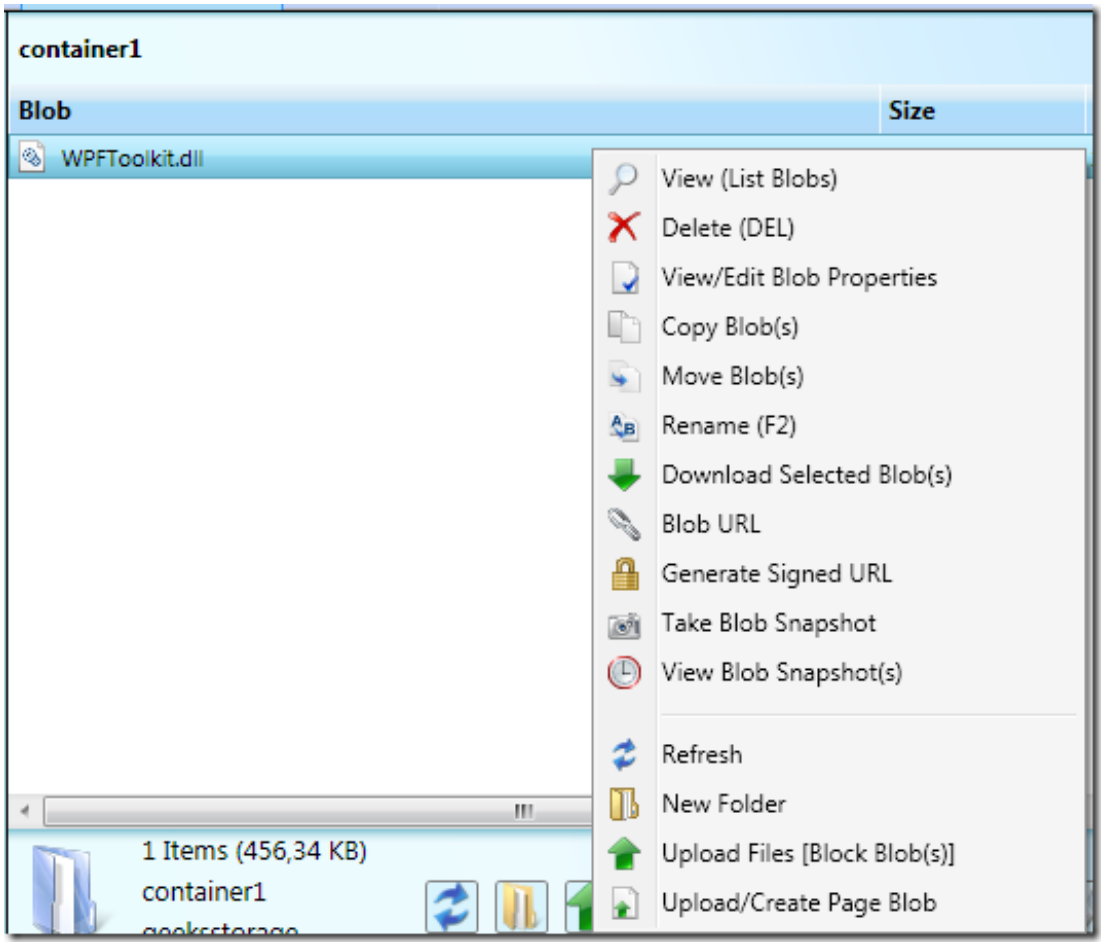


Figura 2.11.- Acciones posibles a realizar sobre un Blob.

Por ejemplo, también se podrían crear tablas y subir entidades a esas tablas.

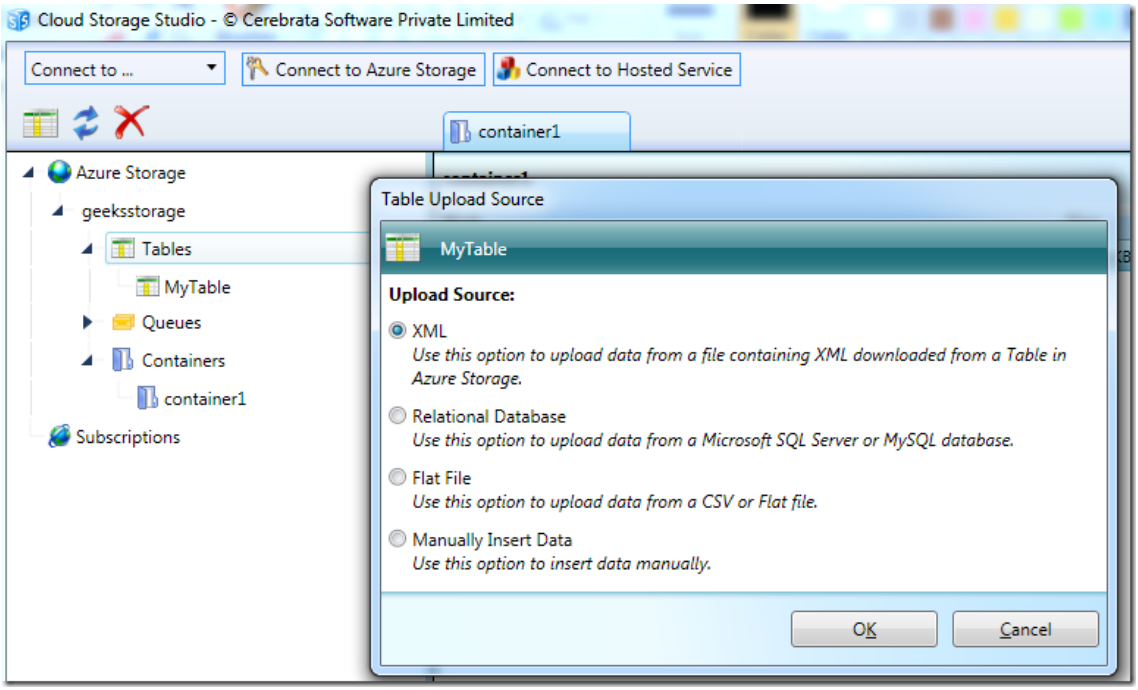


Figura 2.12.- Creación de tablas y subida de entidades

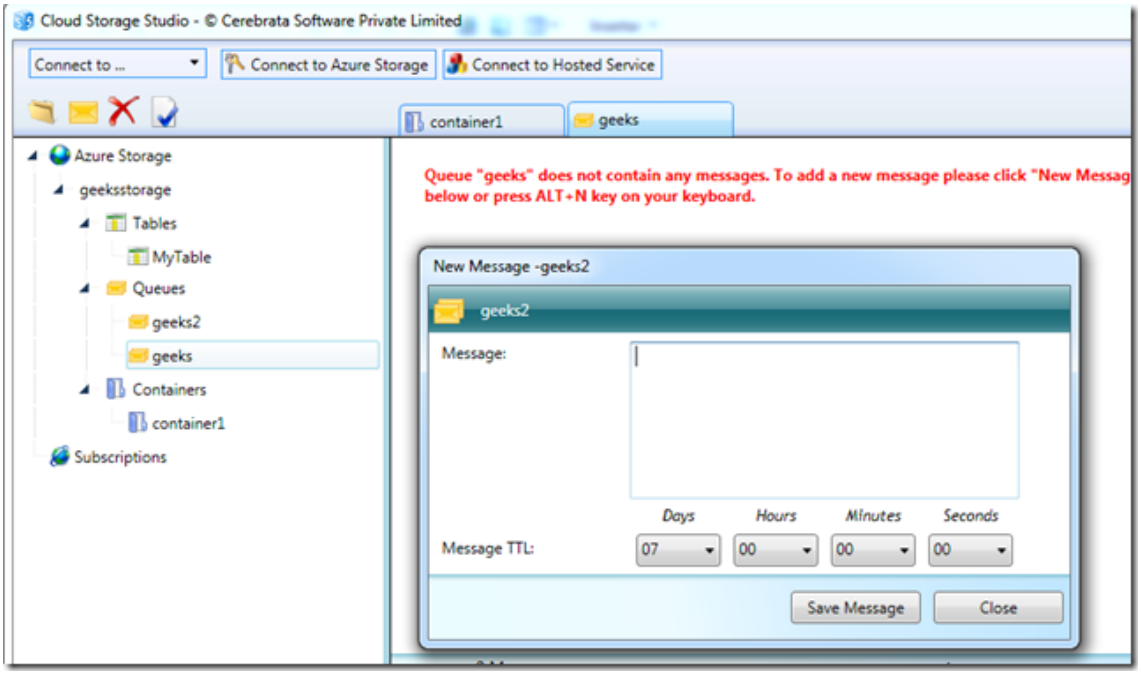


Figura 2.13.- Pantalla de administración de colas

5.- POWERSHELL

Trabajando con aplicaciones reales en Windows Azure te das cuenta rápidamente de que necesitas automatizar ciertas tareas, ya que hacerlas de manera manual puede resultar muy pesado y poco productivo; despliegues, actualizaciones, monitorizaciones etc…

Para esta fin los CmdLets de PowerShell para Windows Azure ha sido desde su aparición un gran aliado, aliado que ahora se actualiza a la versión 2.0 que incluye nueva funcionalidad gracias a que Windows Azure ofrece cada vez más APIs de Administración que los desarrolladores pueden utilizar.

Tanto la descarga como la documentación sobre su utilización se pueden descargar desde CodePlex.

Por ejemplo, algunos comandos nuevos son:

Tabla 2.1. - Windows Azure Storage Analytics

Comando	Descripción
Get-StorageAnalyticsLogs	Downloads the analytics logs for the specified service.
Get-StorageAnalyticsMetrics	Downloads the Windows Azure Storage Analytics metrics for the specified service
Get-StorageServicePropertiesForAnalytics	Analytics properties for a storage account.
Set-StorageServicePropertiesForAnalytics	Sets Windows Azure Storage Analytics properties for a storage account.

Tabla 2.2. - SQL Azure Servers

Comando	Descripción
Get-SqlAzureServer	Enumerates SQL Azure servers that are provisioned for a subscription
New-SqlAzureServer	Adds a new SQL Azure server to a subscription
Remove-SqlAzureServer	Deletes a SQL Azure server from a subscription
Set-SqlAzurePassword	Sets the administrative password of a SQL Azure server.

Tabla 2.3. - SQL Azure Firewall

Comando	Descripción
Get-SqlAzureFirewallRules	Retrieves a list of all the firewall rules for a SQL Azure server
New-SqlAzureFirewallRule	Updates an existing firewall rule or adds a new firewall rule for a SQL Azure server
Remove-SqlAzureFirewallRule	Deletes a firewall rule from a SQL Azure server.

5.1.- Ejemplos

Crear un servicio hosteado:

```
    $cert = Get-Item cert:\CurrentUser\My\<Thumbprint>
    $subscriptionId = <subscriptionId>
    $servicename = "myservice"


    New-HostedService -ServiceName $servicename -Label "v2.0" -Location "West Europe" -
SubscriptionId $subscriptionId -Certificate $cert
```

Desplegar una aplicación en el entorno de staging y arrancarla una vez desplegada:

```
    $cert = Get-Item cert:\CurrentUser\My\<Thumbprint>
    $subscriptionId = <subscriptionId>
    $servicename = "myservice"
    $storagename = "mystorage"
    $Package = "azuresample.cspkg"
    $configuration = "ServiceConfiguration.Cloud.cscfg"
    $Label = "mysampledeployment"

    New-Deployment -Slot staging -Package $package -Configuration $configuration -ServiceName
$servicename -StorageAccountName $storagename -SubscriptionId $subscriptionId -Certificate $cert
-Label $Label |
        Get-OperationStatus -WaitToComplete |
        Get-Deployment staging -serviceName $servicename -subscriptionId $subscriptionId -
certificate $cert |
        Set-DeploymentStatus -Status running |
        Get-OperationStatus –WaitToComplete
```

Pasar una aplicación del entorno de staging a producción:

```
    $cert = Get-Item cert:\CurrentUser\My\<Thumbprint>
    $subscriptionId = <subscriptionId>
    $servicename = "myservice"

    Get-HostedServices -subscriptionId $subscriptionId -certificate $cert |
        where {$_.ServiceName –eq $servicename} |
        Get-Deployment staging |
        Move-Deployment |
        Get-OperationStatus –WaitToComplete
```

Aumentar el número de instancias de un servicio dsplegado en producción:

```
    $cert = Get-Item cert:\CurrentUser\My\<Thumbprint>
    $subscriptionId = <subscriptionId>
    $servicename = "myservice"

    Get-HostedService $servicename -Certificate $cert -SubscriptionId $subscriptionId  |
    Get-Deployment -Slot Production |
    Set-DeploymentConfiguration {$_.RolesConfiguration["WebRole1"].InstanceCount += 1}
```

6.- CEREBRATA DIAGNOSTICS

Cerebrata Diagnostics es una herramienta muy útil que ofrece la posibilidad de interpretar toda la información de diagnóstico que las aplicaciones desplegadas en Windows Azure puede generar; logs de la aplicación, log de eventos de Windows, contadores de rendimiento etc…

Esta aplicación resulta muy útil para la monitorización y diagnóstico de las aplicaciones Windows Azure.

La aplicación se conecta al storage dónde la aplicación generar la información y la muestra de una forma gráfica para que sea fácil de entender por el usuario que realiza las labores de monitorización.

También desde la herramienta es posible modificar la configuración de diagnóstico de la aplicación que se desea monitorizar.

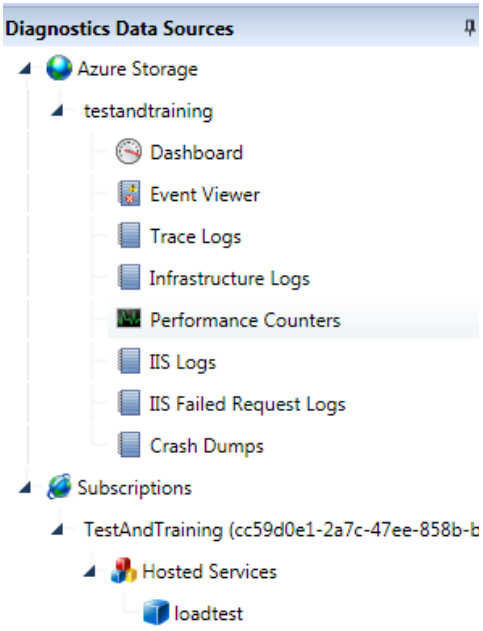


Figura 2.14.- Cerebrata Diagnostics

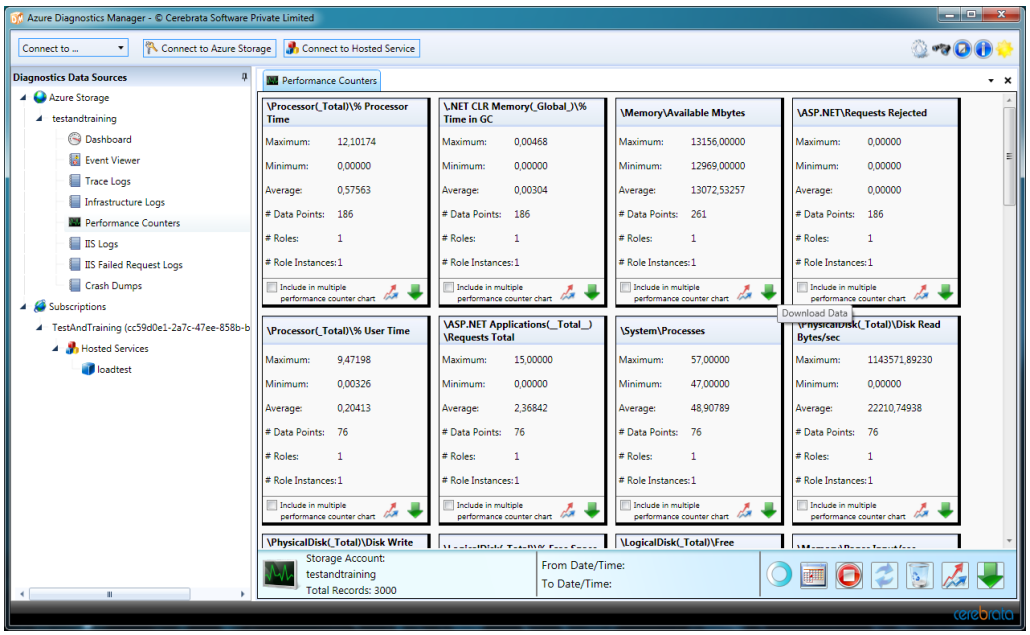


Figura 2.15.- Performance Counters

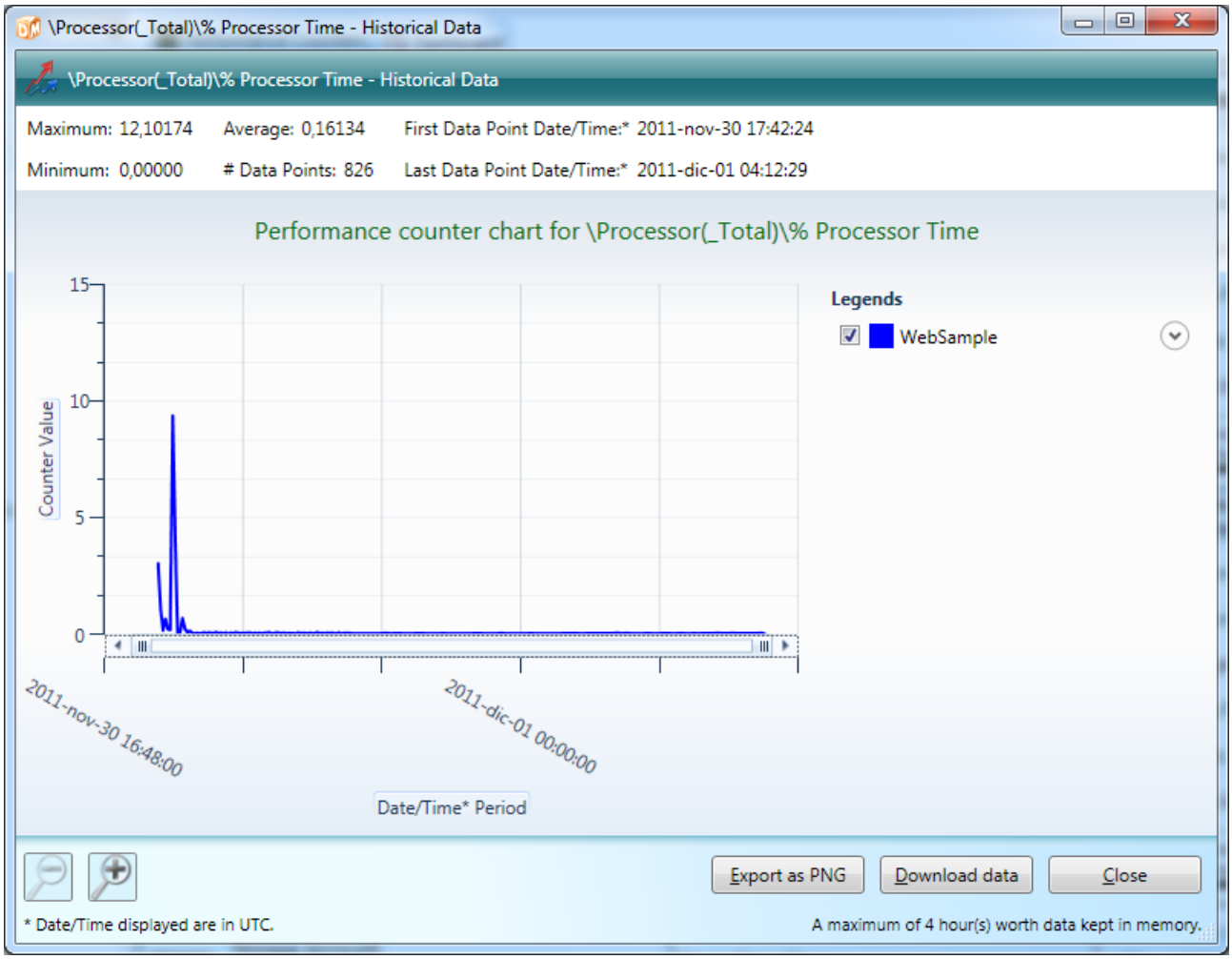


Figura 2.16.- Performance Counters

7.- SPOTLIGHT

SpotLight es una herramienta de Quest que ofrece la posibilidad de monitorizar aplicaciones desplegadas en Windows Azure.

La aplicación se conecta al storage dónde la aplicación generar la información y la muestra de una forma gráfica para que sea fácil de entender por el usuario que realiza las labores de monitorización.

También desde la herramienta es posible modificar la configuración de diagnóstico de la aplicación que se desea monitorizar.

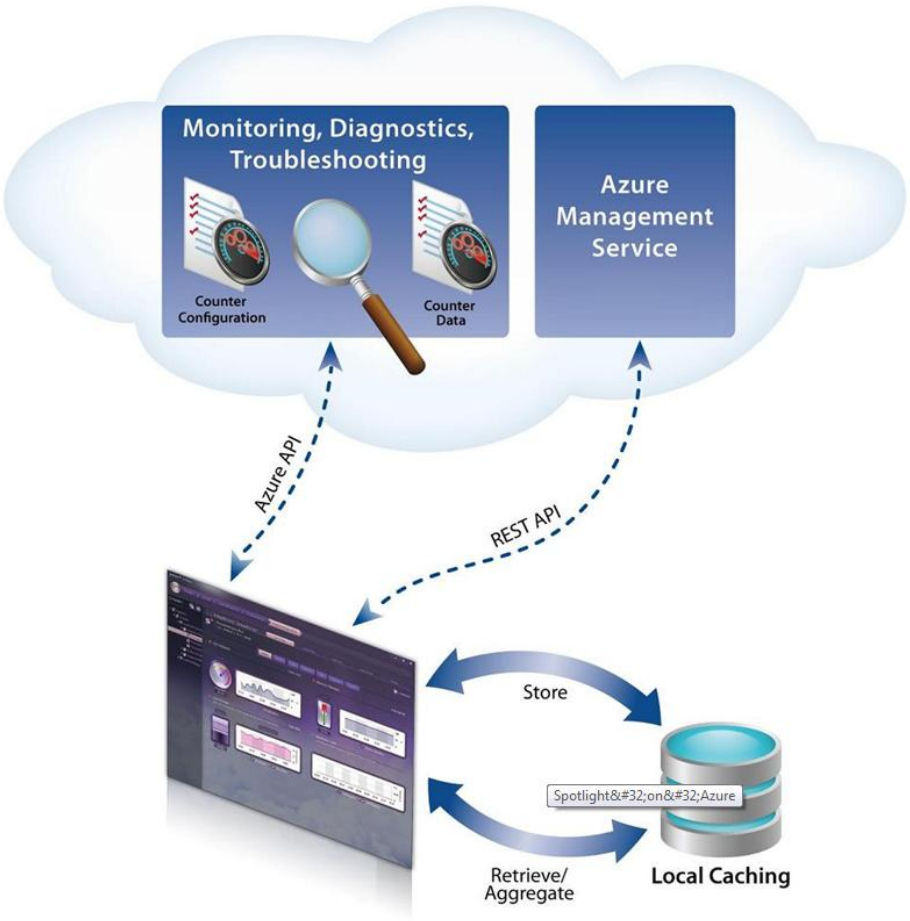


Figura 2.17.- SpotLight

Figura 2.18.- SpotLight Dashboard

8.- DMVS

Desde su aparición en SQL Server 2005 las DMVs han sido una herramienta muy necesaria para poder detectar y solucionar problemas de rendimiento en SQL Server.

SQL Azure está basado en SQL Server, pero desgraciadamente muchas de las DMVs no está disponibles en esta primera versión de Azure….aunque sí que quieren ir añadiéndolas en sucesivas versiones.

El principal problema por el cuál no están soportadas todas las DMVs es porque en SQL Server éstas funcionan a nivel de instancia, cosa que en SQL Azure no es posible, tendrían que funcionar a nivel de base de datos. Este trabajo es el que deben hacer para que todas las DMVs estén accesibles también en SQL Azure.

Por ejemplo, el permiso que hay que tener para trabajar con DMVs en Azure es VIEW DATABASE STATE, mientras que en SQL Server es VIEW SERVER STATE).

Microsoft ha publicado un documento bastante interesante dónde se comentan las diferentes DMVs disponibles en esta primera versión de SQL

Identificar los planes de ejecución que provocan excesivas recompilaciones.

```
select top 25
    sql_text.text,
    sql_handle,
    plan_generation_num,
    execution_count,
    dbid,
    objectid
from
    sys.dm_exec_query_stats a
    cross apply sys.dm_exec_sql_text(sql_handle) as sql_text
where
    plan_generation_num >1
order by plan_generation_num desc
```

Identificar planes de ejecución ineficientes:

```
select
    highest_cpu_queries.plan_handle,
    highest_cpu_queries.total_worker_time,
    q.dbid,
    q.objectid,
    q.number,
    q.encrypted,
    q.[text]
from
    (select top 50
        qs.plan_handle,
        qs.total_worker_time
    from
        sys.dm_exec_query_stats qs
    order by qs.total_worker_time desc) as highest_cpu_queries
    cross apply sys.dm_exec_sql_text(plan_handle) as q
order by highest_cpu_queries.total_worker_time desc
```

Búsqueda de indices más beneficiosos

```
SELECT TOP (20)

    CAST(REPLACE(CAST(qp.query_plan AS NVARCHAR(MAX)),

    'xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan"','') AS XML),

    qp.query_plan.value(

        'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/showplan";
```

```
(/ShowPlanXML/BatchSequence/Batch/Statements/StmtSimple/QueryPlan/MissingIndexes/MissingIndexGr
oup/@Impact)[1]' , 'decimal(18,4)') * execution_count AS TotalImpact

   FROM

       sys.dm_exec_query_stats qs

       cross apply sys.dm_exec_sql_text(sql_handle) st

       cross apply sys.dm_exec_query_plan(plan_handle) qp

   WHERE

       qp.query_plan.exist(

           'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/showplan";/ShowPlanXML/BatchSequence/Batch/Stat
ements/StmtSimple/QueryPlan/MissingIndexes/MissingIndexGroup/MissingIndex[@Database!="m"]') = 1

   ORDER BY TotalImpact DESC
```

Planes de ejecución más costosos

```
   SELECT TOP (20)
       q.text, s.total_elapsed_time, s.max_elapsed_time, s.min_elapsed_time,
       s.last_elapsed_time, s.execution_count, last_execution_time, *
   FROM sys.dm_exec_query_stats as s
         cross apply sys.dm_exec_sql_text(plan_handle) AS q
   WHERE s.last_execution_time > DateAdd(mi , -1500 , GetDate()) -- solo las que se han
ejecutado recientemente
        AND text not like '%sys.%' -- eliminar esta propia consulta
        ORDER BY s.total_elapsed_time DESC
```

Tamaño de las tablas e índices

```
   SELECT
         sys.objects.name AS Name, SUM(reserved_page_count) * 8.0 / 1024 AS [Reserved in MB],
SUM(used_page_count) * 8.0 / 1024 AS [Used in MB], MAX(row_count) AS [Number of rows]
   FROM
         sys.dm_db_partition_stats, sys.objects
   WHERE
         sys.dm_db_partition_stats.object_id = sys.objects.object_id

   GROUP BY sys.objects.name

   UNION ALL

   SELECT
       sys.indexes.name AS Name, SUM(reserved_page_count) * 8.0 /1024 AS [Reserved in MB],
SUM(used_page_count) * 8.0 / 1024 AS [Used in MB], MAX(row_count) AS [Number of rows]
   FROM
       sys.dm_db_partition_stats, sys.indexes
   WHERE
       sys.dm_db_partition_stats.object_id = sys.indexes.object_id
       AND sys.dm_db_partition_stats.index_id = sys.indexes.index_id
       AND sys.dm_db_partition_stats.index_id > 0
   GROUP BY sys.indexes.name
   ORDER BY 2 DESC
```

9.- RED GATE BACKUP

Red Gate dispone de una herramienta completamente gratuita que permite realizar backups de base de datos SQL Azure.

Dispone tanto de una interfaz Visual como la posibilidad de realizar las acciones a través de línea de comandos.

Permite seleccionar una base de datos e indicar si se quiere hacer una copia de base de datos o una acción exportar sobre Windows Azure Storage. En este segunda opción también permite configurar si se quiere disponer de consistencia transaccional, lo que fuerza a que antes de que se haga la exportación se haga una copia de la base de datos.

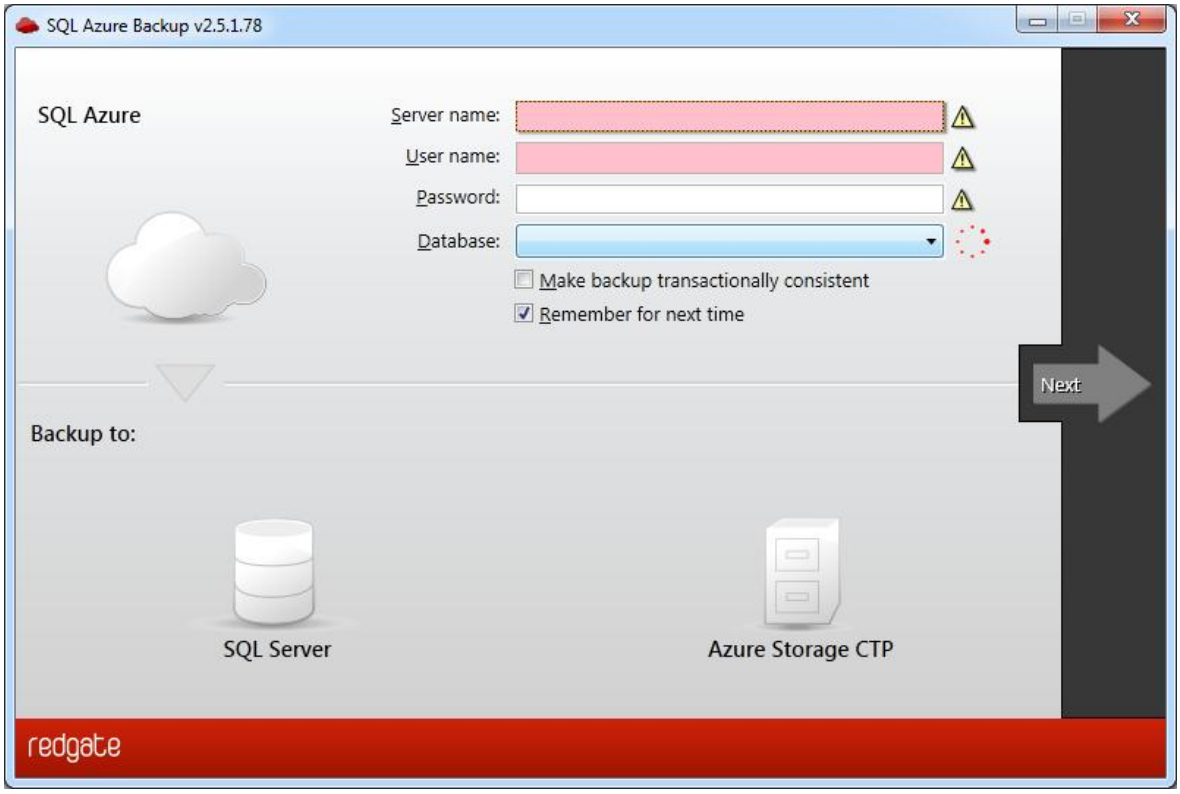


Figura 2.19.- Red Gate Backup

```
RedGate.SQLAzureBackupCommandLine.exe [/Help]

Copy to local sql server:

                  RedGate.SQLAzureBackupCommandLine.exe
                           /AzureServer:server /AzureDatabase:database
                           /AzureUserName:user /AzurePassword:password
                           [/CreateCopy] [/Skip] /LocalServer:server
                           /LocalDatabase:database [/LocalUserName:user]
                           [/LocalPassword:password] [/DropLocal] [/Verbose]
                           [/Help]

Create BACPAC file with Import/Export service CTP (Access Key):

                  RedGate.SQLAzureBackupCommandLine.exe
                           /AzureServer:server /AzureDatabase:database
                           /AzureUserName:user /AzurePassword:password
                           [/CreateCopy] /StorageAccount:name /AccessKey:key
                           /Container:container [/Filename:filename]

Create BACPAC file with Import/Export service CTP (Shared Access Signature):

                  RedGate.SQLAzureBackupCommandLine.exe
```

```
                              /AzureServer:server /AzureDatabase:database
                              /AzureUserName:user /AzurePassword:password
                              [/CreateCopy] /SignatureUrl:url [/Filename:filename]

   /sk:key,        /AccessKey:key           Primary or secondary access key
   /ad:database,   /AzureDatabase:database  Database name to backup
   /ap:password,   /AzurePassword:password  Azure password
   /as:server,     /AzureServer:server      Azure server
   /au:user,       /AzureUserName:user      Azure username
   /sc:container,  /Container:container     Container where file will be placed
   /cc,            /CreateCopy              Ensure db transactional consistentency
   /dl,            /DropLocal               Drop database if it already exists
   /sf:filename,   /Filename:filename       Filename to be created
   /?,             /Help                    Show usage
   /ld:database,   /LocalDatabase:database  Database to be created
   /lp:password,   /LocalPassword:password  Local password
   /ls:server,     /LocalServer:server      Local server
   /lu:user,       /LocalUserName:user      Local username for sql authentication
   /su:url,        /SignatureUrl:url        Shared access signature url
   /s,             /Skip                    Skip objects that fail to be copied
   /sa:name,       /StorageAccount:name     Backup to azure blob storage account
   /v,             /Verbose                 Show verbose output
```

10.- CSS SQL AZURE DIAGNOSTICS

En un punto anterior se comentaba el soporte que SQL Azure tiene de las DMVs y cómo sólo se pueden usar algunas de ellas.

En este apartado se habla una herramienta muy interesante que permite simplificar el trabajo con éstas, ya que simplifica su utilización e interpretación; CSS SQL Azure Diagnostics.

Es una herramienta pensada para simplificar el trabajo con SQL Azure y ayudar a encontrar y solucionar problemas. La herramienta se basa única y exclusivamente en las DMVs soportadas por SQL Azure, no hace ninguna otra cosa, pero simplifica enormemente su uso.

Top 10 CPU

Total Worker Time (µs)	Creation Time (UTC)	Last Execution Time (UTC)	Execution Count	Last Physical Reads	Min Physical Reads	Statement
33223633	4/2 03:32:49	4/5 03:27:36	200425	0	0	(@1 varchar(8000),@2 varchar(8000),@3 varchar (8000),@4 varchar(8000),@5 varchar(8000),@6 varchar(800
8648437	4/21 01:15:42	4/23 09:56:17	81	0	0	SELECT TOP 10 qs.creation_time, qs.last_execution_time, qs.plan_generation_num, qs.execution_count,
2705078	4/21 01:15:42	4/23 09:54:34	80	0	0	SELECT TOP 10 (total_physical_reads/execution_count) as avg_physical_reads, min_physical_reads, last
2581054	4/21 01:15:41	4/23 09:43:25	76	0	0	SELECT TOP 10 (total_logical_reads/execution_count) as avg_logical_reads, min_logical_reads, last_lo
2473632	4/21 01:15:41	4/23 12:37:33	73	0	0	SELECT TOP 10 highest_cpu_queries.total_worker_time, highest_cpu_queries.plan_generation_num, highes
701172	4/23 09:45:35	4/25 07:32:51	19	0	0	SELECT TOP 10 highest_cpu_queries.total_worker_time, highest_cpu_queries.plan_generation_num, highes
692383	4/23 09:55:44	4/25 01:11:33	12	0	0	SELECT TOP 10 qs.creation_time, qs.last_execution_time, qs.plan_generation_num, qs.execution_count,
507812	4/23 09:54:34	4/25 01:11:32	14	0	0	SELECT TOP 10 (total_logical_reads/execution_count) as avg_logical_reads, min_logical_reads, last_lo
469726	4/23 09:55:44	4/25 01:11:32	13	0	0	SELECT TOP 10 (total_physical_reads/execution_count) as avg_physical_reads, min_physical_reads, last
144531	4/23 09:48:10	4/23 09:52:06	4	0	0	SELECT TOP 10 (total_logical_reads/execution_count) as avg_logical_reads, min_logical_reads, last_lo

Figura 2.20.- Top 10 CPU

Top 10 Logical I/O (dates in UTC)						
Avg. Reads	Min. Reads	Avg. Writes	Min. Writes	Count	Last	Statement
4	4	0	0	200425	4/5 03:27:36	(@1 varchar(8000),@2 varchar(8000),@3 varchar (8000),@4 varchar(8000),@5 varchar(8000),@6 varchar(800
651	651	0	0	3	4/4 07:48:31	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @
651	651	0	0	3	4/4 07:48:32	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @
651	651	0	0	3	4/4 07:48:32	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @
651	651	0	0	3	4/4 07:48:32	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @
651	651	0	0	3	4/4 07:48:32	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @
651	651	0	0	3	4/4 07:48:32	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @
651	651	0	0	3	4/4 07:48:32	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @
651	651	0	0	3	4/4 07:48:32	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @
651	651	0	0	3	4/4 07:48:32	declare @maxopen int declare @dblocation nvarchar (50) declare @fieldname nvarchar(100) declare @

Figura 2.21.- Top 10 Logical I/O

11.- WINDOWS AZURE BOOTSTRAPPER

Windows Azure Bootstrapper es una utilidad que está disponible en CodePlex, cuyo objetivo es simplificar las tareas de instalación en el arranque de los web y worker roles.

Se trata de una utilidad de línea de comandos que puede ser de gran utilidad cuando se necesitan instalar componentes extras en los despliegues, como por ejemplo instalar la runtime de Crystal Report, las tools de una determinada tecnología que no está disponible en Azure por defecto etc…

Permite conectar al storage para descargse instaladores, descargar algo de una determinada URL, descomprimir, ejecutar, definir el número de instalaciones simultáneas etc…

En su página de CodePlex se comentan estos ejemplos, que pueden servir para ilustrar más claramente el objetivo de la misma:

```
   bootstrapper.exe -get bootstrap/Installer.zip -lr $lr(temp) -unzip $lr(temp)\extract -sc
$config(ConnectionString) -run $lr(temp)\extract\installer.msi -args /qn –block
```

El ejemplo anterior se descarga un fichero llamado installer.zip de un blob container llamado bootstrap al directorio temporal local, lo descomprime, y ejecuta el fichero installer.msi pasándole los argumentos /qn.

Otro ejemplo podría ser el siguiente, que lo que hace es descargarse desde la página de Microsoft las herramientas de ASP.MVC2 y luego instalarlas.

```
   bootstrapper.exe -get http://download.microsoft.com/download/F/3/1/F31EF055-3C46-4E35-AB7B-
3261A303A3B6/AspNetMVC3ToolsUpdateSetup.exe -lr $lr(temp) -run
$lr(temp)\AspNetMVC3ToolsUpdateSetup.exe -args /q
```

12.- WINDOWS AZURE MULTI APPLICATION

Windows Azure Multi Application es una aplicación de ejemplo disponible en CodePlex que permite desplegar de forma dinámica aplicaciones sobre una instancia de Windows Azure desplegada anteriormente.

Cuando se despliega la aplicación en Windows Azure únicamente se despliega la herramienta de administración, la cuál permite posteriormente desplegar nuevas aplicaciones o actualizar o eliminar las que ya estuvieran desplegadas.

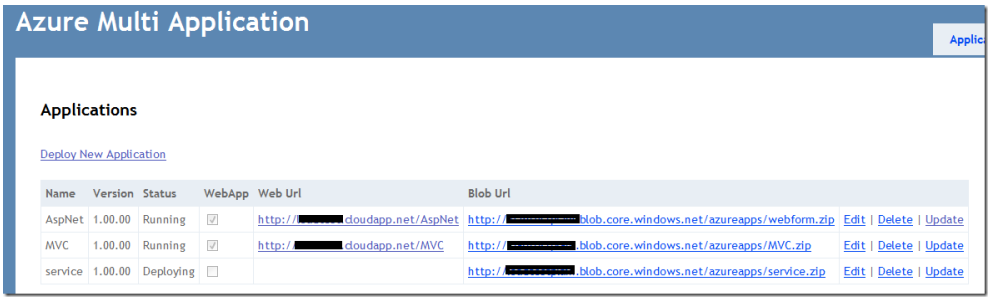


Figura 2.22- Lista de aplicaciones desplegadas

El objetivo de la aplicación es optimizar el uso de recursos de la instancia y por tanto, reducir el coste de uso de la plataforma.

Sobre la misma instancia pueden desplegarse instancias de diferente naturaleza; ASP.NET Web Forms, ASP.NET MVC, Servicios Windows etc…

La aplicación soporta disponer de múltiples instancias y modificar el número de las desplegadas de forma dinámica sin tener que hacer acciones adicionales.

Si os interesa podéis coger el código fuente de la misma aquí y si la probáis, no dudéis en proponer mejoras!

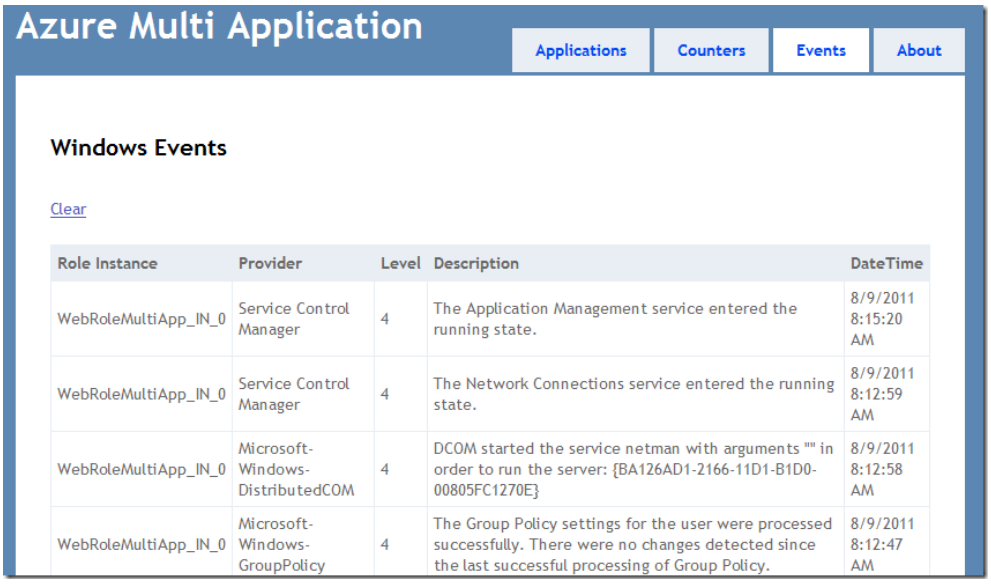


Figura 2.23.- Windows Events

13.- AZURE MONITOR

Se trata de una aplicación Windows Phone 7 cuyo objetivo es permitir la realización de labores administrativas desde el dispositivo móvil.

Básicamente, la funcionalidad que realiza actualmente es la siguiente:

- Visualizar los servicios hosteados.
- Conocer su información, tanto la del servicio como la de los despliegues de producción y/o staging.
- Poder eliminar un despliegue.
- Poder parar o arrancar un despliegue existente.
- Realizar un paso de staging a producción.
- Reiniciar una instancia.
- Modificar el número de instancias desplegadas de un determinado rol.
- Visualizar los servicios de almacenamiento existentes en la subscripción junto con su información.
- Visualizar los servidores SQL Azure; su información, las reglas del firewall etc…
- Monitorizar el estado de los diferentes servicios de Windows Azure.

A continuación se muestran unas capturas sacadas desde el emulador para que se vea el aspecto de la aplicación.

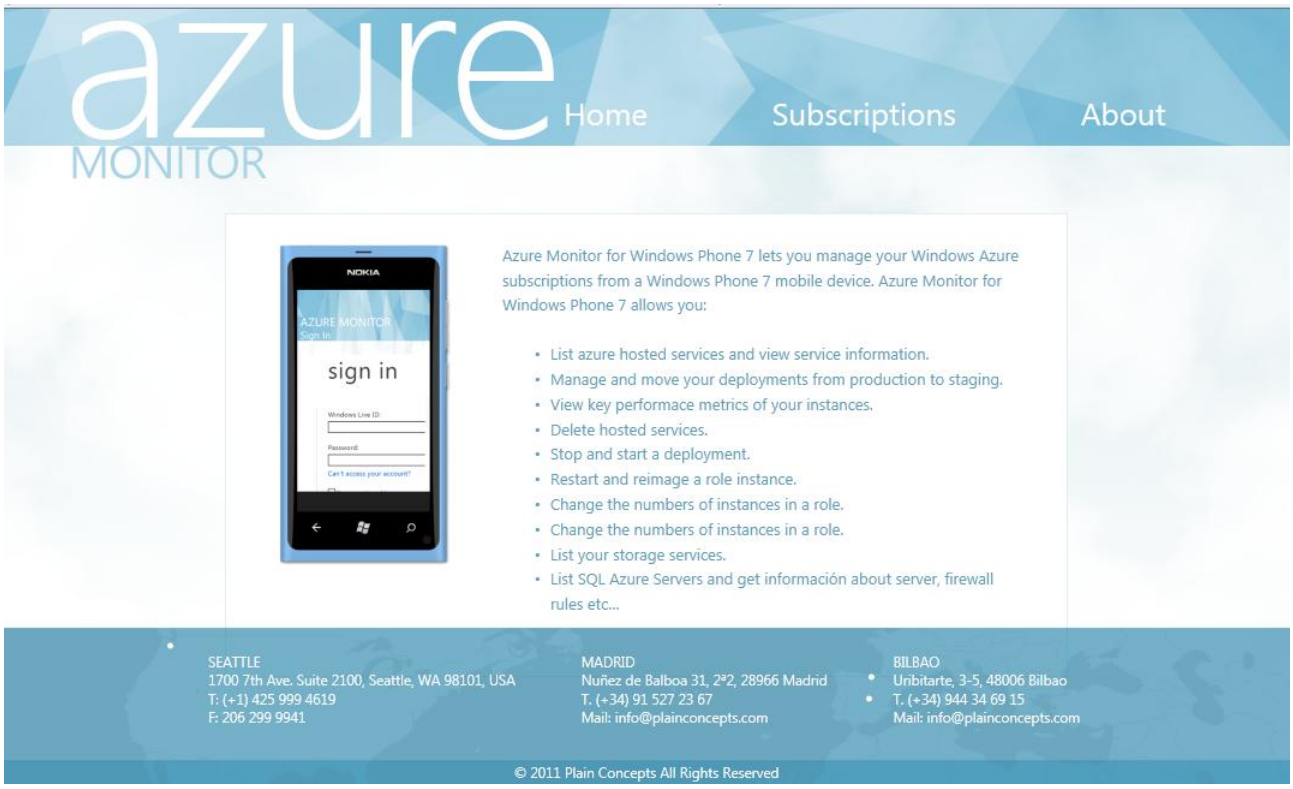


Figura 2.24.- Azure Monitor

Figura 2.25.- Azure Monitor

14.- AUTOESCALADO DE INSTANCIAS

Otra de las grandes características que hace de Windows Azure una plataforma de indudable interés es la elasticidad y flexibilidad que nos ofrece a la hora de desplegar nuestra aplicación en múltiples instancias.

Desplegar una aplicación en 1 o 100 máquinas es simplemente cambiar un fichero de configuración, cosa que se puede establecer en el momento del despliegue o a posteriori. Lógicamente, recordad que la facturación va en función del número de instancias, pago por uso.

De esta manera, un usuario de Azure puede modificar fácilmente el número de instancias de la aplicación en función de las necesidades de ésta, ya sea añadiendo o eliminando instancias.

Aumentar una instancia en un rol ya desplegado podría ser tan sencillo como estos comandos de powershell.

```
    $cert = Get-Item cert:\CurrentUser\My\<Certificate ThumbPrint>
    $sub = <Azure Subcription Id>
    $servicename = <Service Name>

    Get-HostedService $servicename -Certificate $cert -SubscriptionId
$sub |
    Get-Deployment -Slot Production |
    Set-DeploymentConfiguration
{$_.RolesConfiguration["WebSample"].InstanceCount += 1}
```

Cuando necesita más potencia la pone y cuando necesita menos la quita, consiguiendo entre otras cosas dos objetivos; pagar por aquello que realmente está usando y sobre todo, poder conseguir que la aplicación se pueda adaptar a la demanda y evitar que ésta pueda dejar de dar servicio.

Es en este momento dónde casi siempre sale la misma pregunta; ¿Se puede hacer que el añadir o eliminar instancia sea un proceso automático? Que no tenga que ser una persona la encargada de detectar las necesidades de la aplicación para aumentar o disminuir instancias, sino poder disponer de un sistema automático que detecte cuando es necesario modificar el número de instancias.

Generalmente esta necesidad viene de situaciones de picos impredecibles; tienes desplegada tu aplicación con N instancias y por cualquier motivo existe un punto en el tiempo, inesperado lógicamente, dónde debe aumentarse la capacidad de proceso. Si es un proceso manual, puede ocurrir que cuando el administrador se entere ya sea demasiado tarde.

A día de hoy no existe ninguna herramienta Microsoft que permita realizar auto escalado dinámico de instancias; No existe una herramienta que disponga de un sistema de reglas en las cuáles nosotros podamos configurar la lógica de escalado.

La primera alternativa es la de siempre; Hacerse uno mismos una herramienta que disponga de la funcionalidad que se necesite.

Windows Azure permite obtener diferente y muy variada información de diagnóstico y monitorización de las instancias desplegadas en Windows Azure. A partir de esta información, se puede construir una aplicación capaz de interpretar dicha información e implementar la lógica que se considere adecuada para detectar cuando es necesario añadir o eliminar instancias de un determinado rol; Si el procesador está el 80% entonces añadir una instancia más.

Si esta opción puede resultaros válida la mejor opción actualmente es utilizar un Application Block de Enterprise Libray pensado para este tipo de tareas, el cual se puede encontrar en CodePlex.

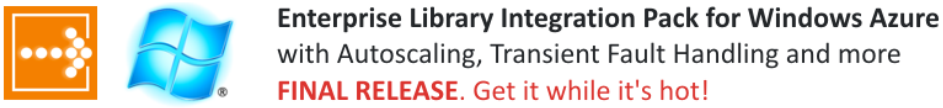


Figura 2.26.- Enterprise Library

Existe también la posibilidad de usar herramientas de terceros que contengan la funcionalidad que se necesite.

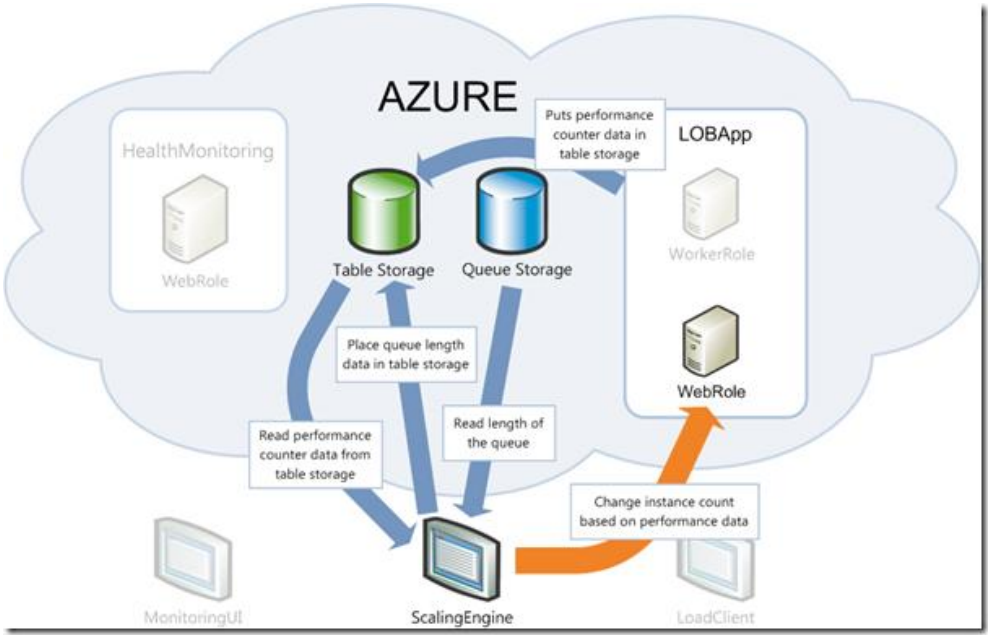


Figura 2.27. – Azure Watch

Una de las herramientas que disponen de la funcionalidad de auto escalado es **AzureWatch**. Es una herramienta comercial (pago por uso), que ofrece al usuario la posibilidad de configurar un sistema de reglas tan complejo como desee, para poder modificar el número de instancias desplegadas de un rol. Como es esperar, hace uso de la información que Windows Azure ofrece sobre diagnóstico y monitorización, lo mismo que se comentaba en el caso anterior.

Además de la funcionalidad de auto escalado, esta herramienta ofrece funcionalidad muy útil para monitorizar el estado de las aplicaciones Windows Azure, como se puede ver en los pantallazos que se muestran a continuación:

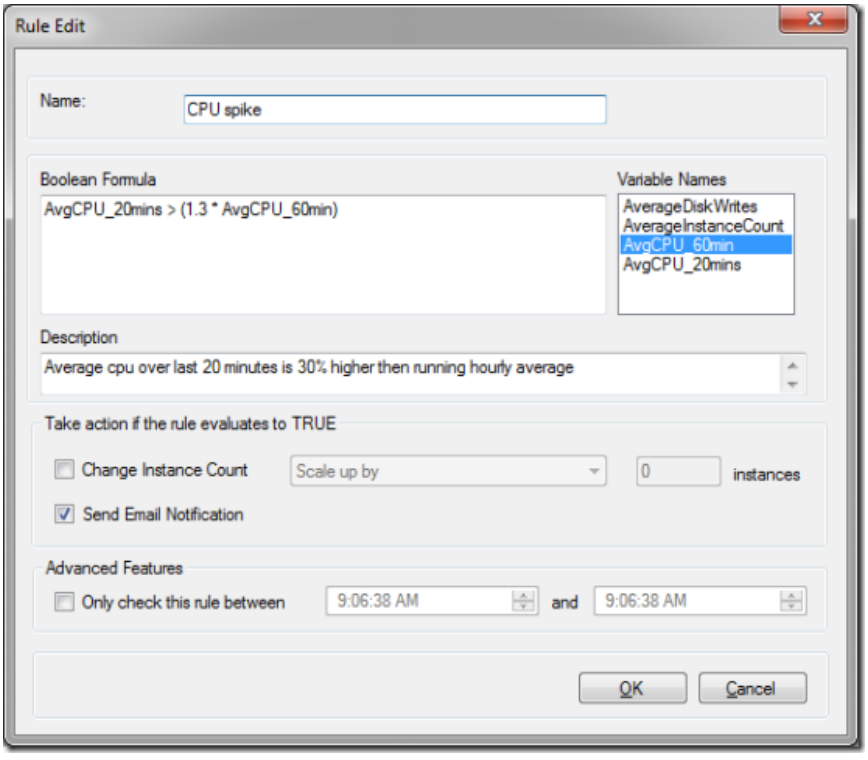


Figura 2.28.- Azure Watch

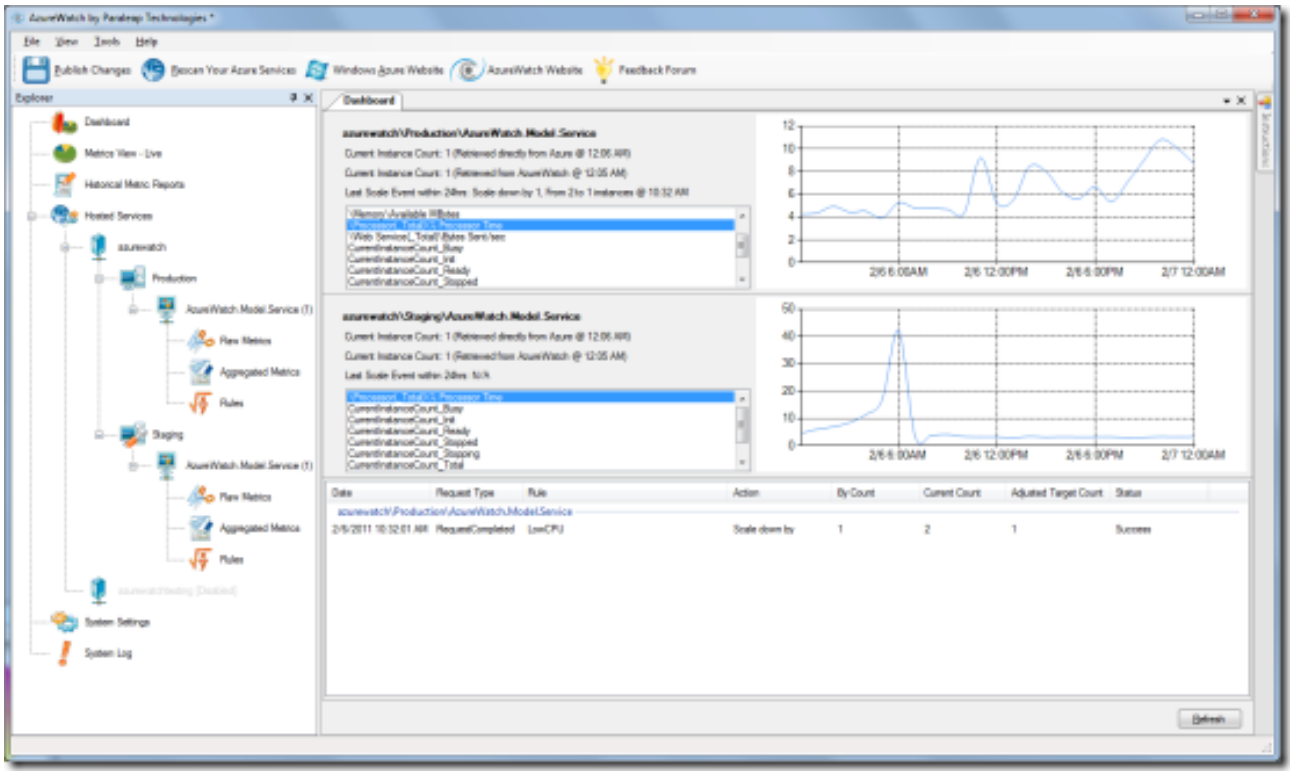


Figura 2.29. – Azure Watch

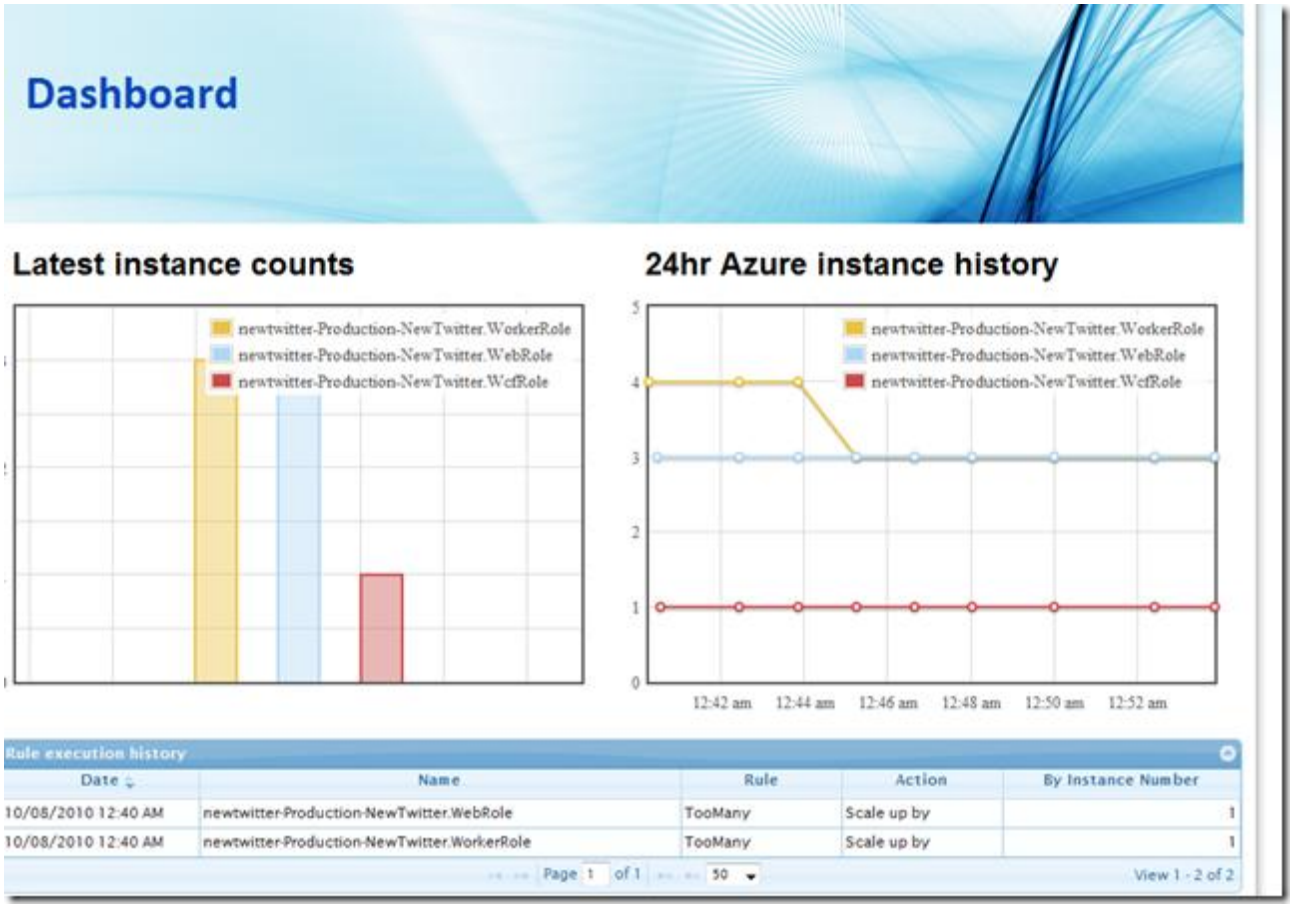
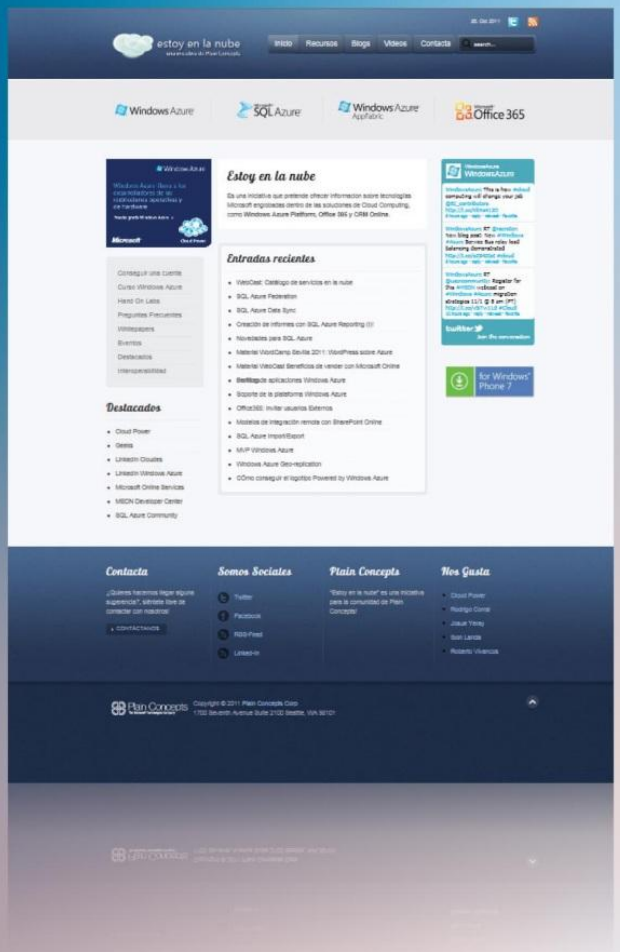


Figura 2.30. – Azure Watch

estoy en la nube

INICIATIVA DE PLAIN CONCEPTS

www.estoyenlanube.com



Windows Azure

www.plainconcepts.com

Plain Concepts is a company specialized in Microsoft technologies, agile methodologies, Application Lifecycle Management, performance tuning, advanced debugging, software architecture and User Experience.

Plain Concepts focuses on delivering high quality consulting, mentoring and training as well as in being an effective and reliable team resolving all type of software development issues.

Plain Concepts

¿Aún quieres más?

campus MVP

Formación online especializada en tecnologías Microsoft.

krasis PRESS

Los libros que lo saben todo sobre tecnologías Microsoft.

Síguenos y descubrirás los mejores trucos y recursos:

facebook.com/campusmvp twitter.com/campusmvp

feed your brain®

- Sin tener que desplazarse
- Sin romper el ritmo de trabajo
- Preguntándole a los que más saben

infórmate ya:
902 876 475
www.campusmvp.com

http://www.krasis.com

krasis

Microsoft Partner
Silver Learning
Silver Software Development