

Desarrollando aplicaciones para Windows Azure

Para los desarrolladores, construir aplicaciones para Windows Azure requiere un proceso similar al que se usa para desarrollar aplicaciones ASP.NET. Para hacerles la tarea más fácil, Windows Azure provee unos "Project Templates" para crear Web Roles / Worker Roles o ambos. Una diferencia importante, es que las aplicaciones Windows Azure no corren localmente. Esta diferencia hace el desarrollo de estas aplicaciones más costosas ya que usar los recursos de Windows Azure no será gratuito.

Para mitigar este problema, Microsoft provee la fábrica de desarrollo (Development Fabric), que es una versión del entorno de Windows Azure que corre en la maquina del desarrollador.

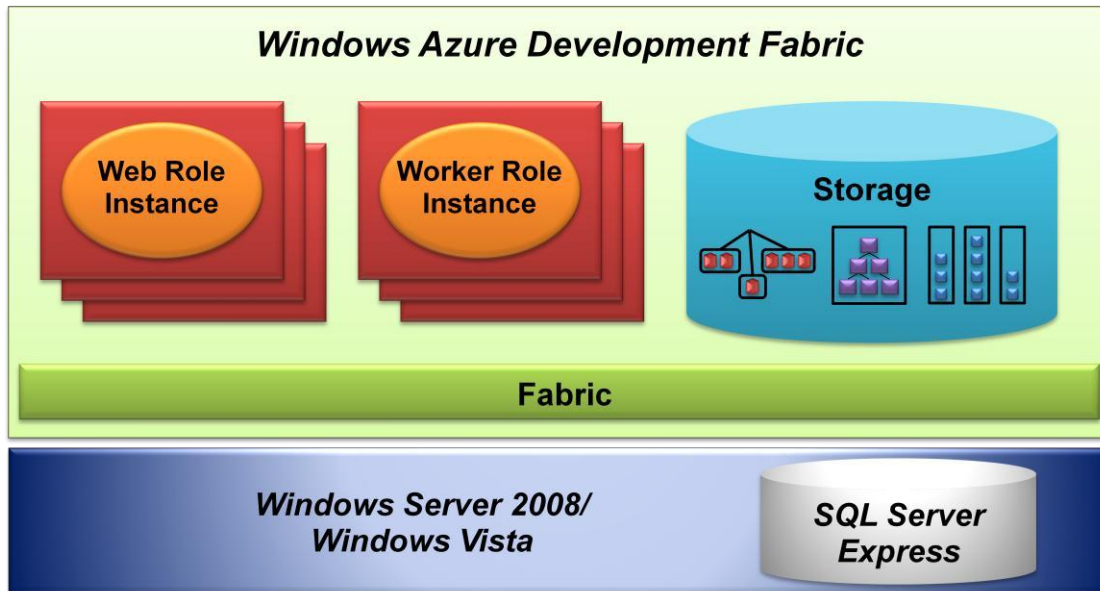


Figura 8: La fábrica de desarrollo provee una copia del entorno de Windows Azure para los desarrolladores

La fábrica de desarrollo corre en una maquina simple que tenga Server 2008 o Vista. La misma, emula la funcionalidad completa de Windows Azure en la nube, Web Roles, Worker Roles, y las tres opciones de almacenamiento. Un desarrollador puede construir una aplicación Windows Azure, hacer el despliegue (deploy) en la fábrica de desarrollo, y correrla de la misma manera que en el entorno real. Puede determinar cuántas instancias de cada rol deberían correr, de hecho, es posible crear una aplicación Windows Azure sin haber usado nunca Windows Azure en la nube. Una vez que la aplicación ha sido desarrollada y testeada localmente, el desarrollador puede subirla junto con su configuración a través del portal de Windows Azure y finalmente ponerla en producción. De todas maneras, hay algunas diferencias con la nube. No se puede adjuntar el depurador a la aplicación corriendo Windows Azure, por lo tanto, el desarrollador debe apoyarse mayormente en el sistema de logging. También se provee un servicio de alertas a través del agente de Windows Azure, y la plataforma alertará vía email, mensajes instantáneos, o algún otro mecanismo a los receptores que sean configurados. Si se desea, la fábrica de Windows Azure puede detectar si una aplicación falla y envía una alerta. La plataforma también provee información detallada acerca de los recursos que consume la aplicación, incluyendo tiempo de proceso, consumo de ancho de banda, y almacenamiento.

Diferencias entre almacenamiento de desarrollo y los servicios de almacenamiento de Windows Azure

Hay varias diferencias claves entre el almacenamiento de desarrollo local y los servicios de almacenamiento de Windows Azure

Diferencias generales

- El almacenamiento de desarrollo solo soporta una cuenta fija con una clave de autenticación conocida. Esta cuenta y esta clave están permitidas solo para uso con el almacenamiento de desarrollo. Las mismas son:

Account name: devstoreaccount1

Account key:

Eby8vdM02xNOcqFlqUwJPLlmEtlCDXJ1OUzFT50uSRZ6IFsuFq2UVErCz4I6tq/K1SZFPTOtr/KBHBeksoGMGw==

- El almacenamiento de desarrollo no es escalable y no soportara un gran número de usuarios concurrentes
- El esquema URI soportado por el almacenamiento de desarrollo difiere del esquema soportado por el almacenamiento en la nube. El esquema URI de desarrollo especifica la cuenta como parte jerárquica de la URI, en vez de cómo parte del nombre del dominio. La diferencia es debido al hecho de que la resolución del nombre del dominio está disponible en la nube pero no en la maquina local. Para más información de este tema, vea [Understanding Storage Service URIs](#).
- El almacenamiento de desarrollo no tiene soporte completo para mostrar información extendida de los errores

Diferencias para los servicios de almacenamiento de Blobs

- Los blobs locales soportan tamaños de hasta 2GB.

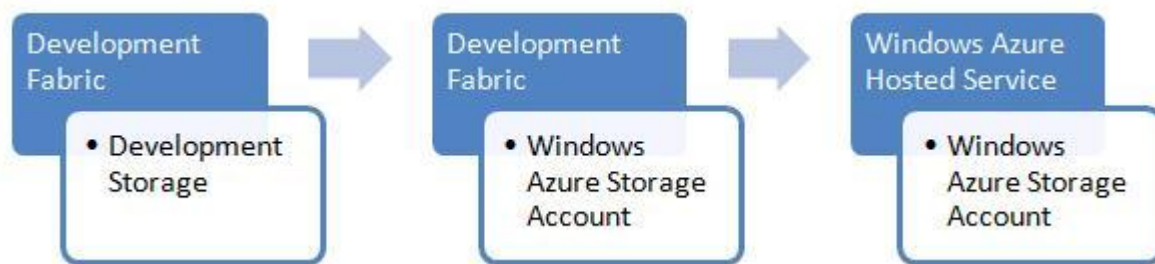
Diferencias para los servicios de almacenamiento de tablas

- El servicio local de tablas requiere la configuración local de un esquema fijo de tablas para poder usarlas.
- El servicio local de tablas soporta solo autenticación Shared Key Lite.
- El servicio local de tablas no soporta creación o borrado dinámico de tablas.
- El servicio local de tablas no soporta crear entidades con un numero variables de propiedades dentro de la misma tabla.
- El servicio local de tablas limita el tamaño las propiedades String a un máximo de 1000 caracteres.
- El servicio local de tablas soporta solo tipos Dates que tengan un rango soportado por SQL Server

Poner en producción una aplicación Windows Azure

Un buen proceso a seguir cuando movemos una aplicación local al ambiente de producción sería:

- Lograr que la aplicación funcione localmente
- Correr la aplicación localmente y usar el servicio de almacenamiento de Windows azure
- Correr el servicio en Windows Azure utilizando los servicios de almacenamiento de Windows Azure



Poner una aplicación Windows Azure en producción involucra dos pasos:

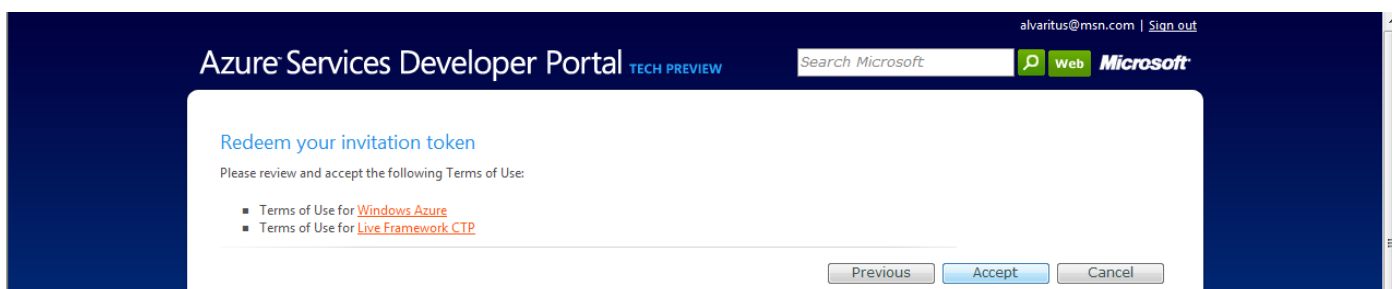
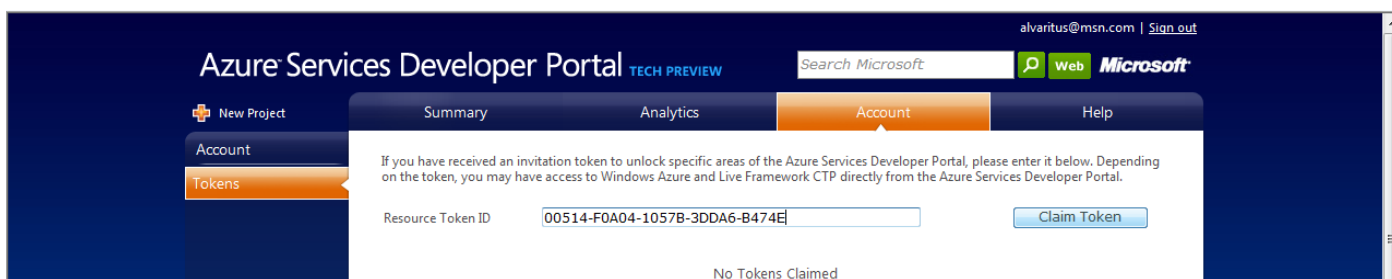
- Obtener una cuenta de almacenamiento.
- Hacer los cambios en la configuración para utilizar los servicios de almacenamiento de la nube

Crear una cuenta de almacenamiento

Ingresa a <http://www.microsoft.com/Azure> y registrarse. Recibirá un mail con el código de invitación.

Ingresa al portal de Azure Service Developer (<http://lx.azure.microsoft.com/fs>) y regístrate con su Live ID.

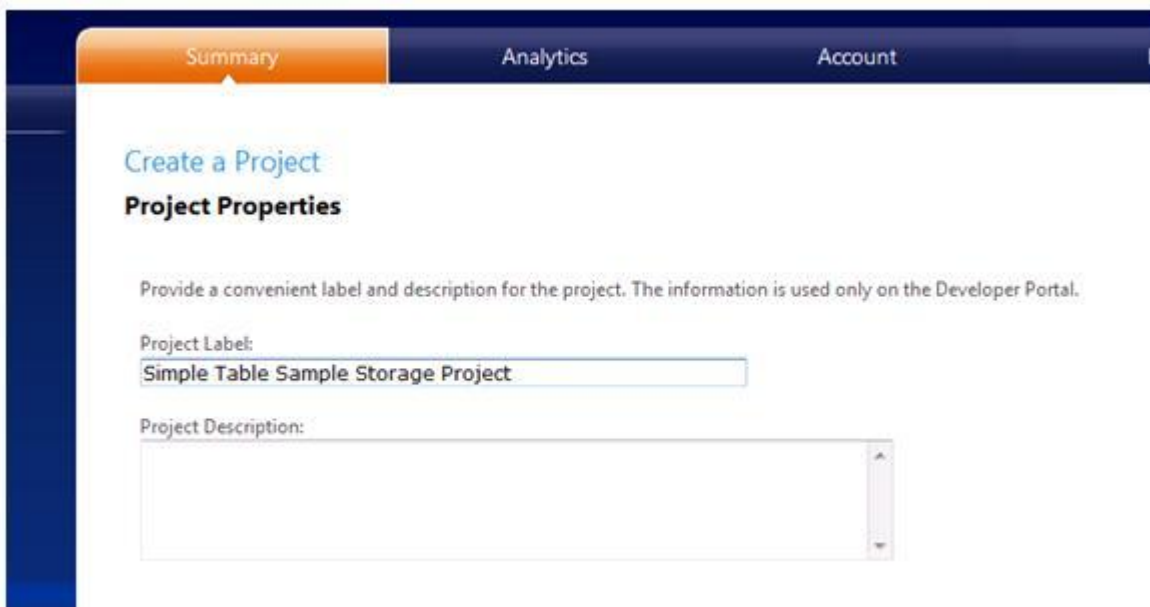
La primera vez deberá activar su cuenta ingresando a “Account”, luego haga un clic en “Manage My Tokens” y pegue su código de invitación obtenido en el primer paso. Presione “Claim Token” y finalmente “Accept”.



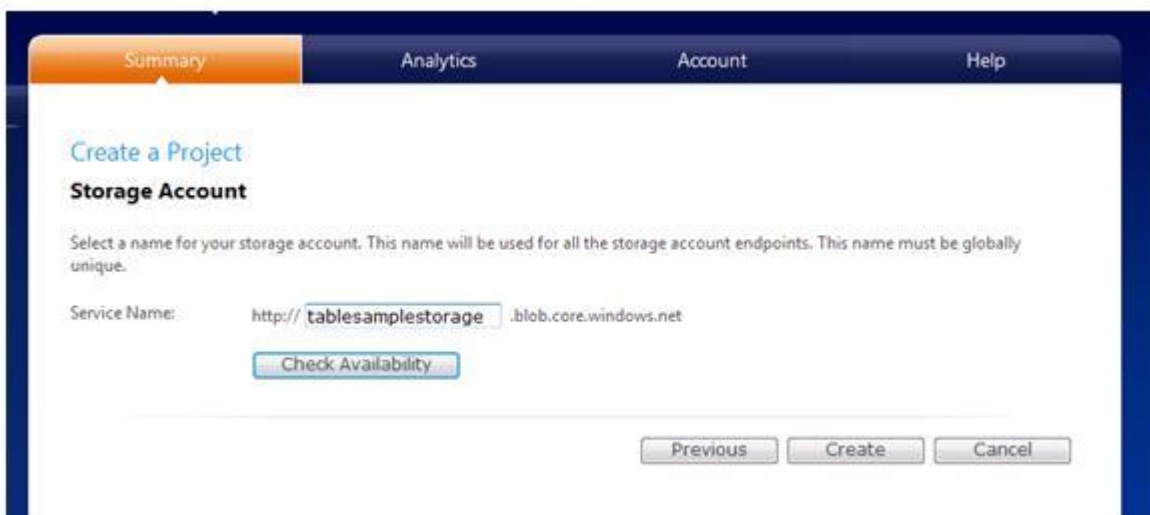
Desde la página del portal, seleccione “New Project” y haga clic en “Storage Account”.



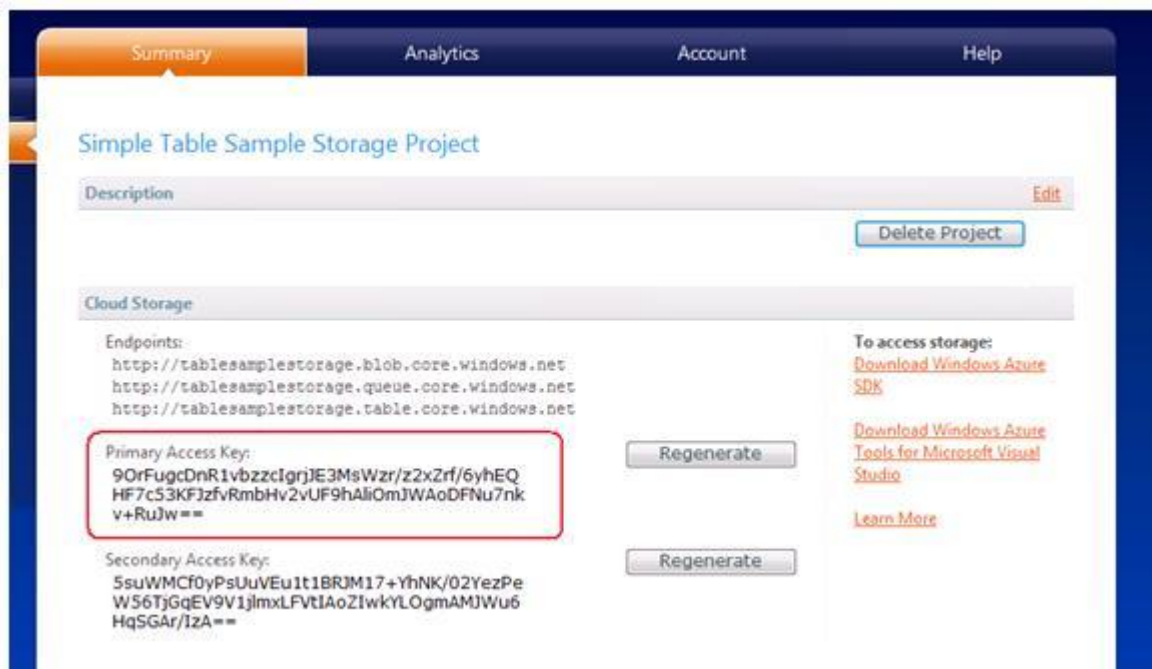
Seleccione un nombre para el proyecto, y opcionalmente una descripción breve del proyecto.



Luego llene el nombre del servicio con el formato servicio/dominio (es importante poner el nombre en minúsculas).



Finalmente aparecerá una pantalla de resumen en la que se muestran los endpoints y las claves para acceder a los mismos



Cambios en la configuración para usar el almacenamiento de Windows Azure

Cuando usamos el tipo `StorageAccountInfo` del proyecto `StorageClient` que se encuentra en los ejemplos del SDK de Windows Azure, tendremos que cambiar los seteos que se encuentran en los archivos de configuración para correr en la nube. Desde la página de resumen, que vimos anteriormente, se pueden sacar los endpoints y la clave de acceso. Actualice el nombre de cuenta (`AccountName`) al nombre del servicio ingresado (en la imagen superior `tablesamplestorage` – en minúsculas). Luego Actualice la clave de cuenta (`AccountSharedKey`) con la clave obtenida en la página de resumen. La segunda clave es generada como un respaldo. Se pueden regenerar las claves si es necesario. **Nota:** Los endpoint listados en la página de resumen no son las URL's que se usaran para acceder a los servicios de almacenamiento si estamos usando el tipo `StorageAccountInfo` del proyecto `StorageClient` mencionado anteriormente. A continuación se muestra un ejemplo de un archivo de configuración:

```
<?xml version="1.0"?>
<ServiceConfiguration serviceName="EjemploDeTableStorage" xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration" >
  <Role name="WebRole">
    <Instances count="1"/>
    <ConfigurationSettings>
      <!--<Setting name="AccountName" value="devstoreaccount1"/>
      <Setting name="AccountSharedKey" value="Eby8vdM02xNOcqFlqUwJPLlmEtlCDXJ1OUzFT50uSR26IFsuFq2UVErCz4I6tq/1" />
      <Setting name="TableStorageEndpoint" value="http://127.0.0.1:10002/" />-->
      <Setting name="AccountName" value="soluciondeejemplo"/>
      <Setting name="AccountSharedKey" value="ZX7jnJObn/IghebgDKefl1OLfr1VusJNKBO7isfS2XxkIEWuU+MVMqGCoPIIsvyX1" />
      <Setting name="TableStorageEndpoint" value="https://table.core.windows.net"/>
      <Setting name="BlobStorageEndpoint" value="https://blob.core.windows.net"/>
      <Setting name="QueueStorageEndpoint" value="https://queue.core.windows.net"/>
      <Setting name="allowInsecureRemoteEndpoints" value="false"/>
    </ConfigurationSettings>
  </Role>
</ServiceConfiguration>
```

Nota: allowInsecureRemoteEndpoints: Este seteo es usados por los proveedores ASP.NET. El valor por defecto es False u omitir la clave. Por defecto, los proveedores ASP.NET no se conectaran a endpoints HTTP remotos para los servicios de almacenamiento. Se creó ésta protección porque los proveedores podrían transmitir información de seguridad sensible sobre el canal. Setee este valor a True para habilitar a los proveedores a conectarse a endpoints HTTP remotos para los servicios de almacenamiento

La siguiente tabla muestra ejemplos de configuración de endpoints y servicios de almacenamiento de Windows Azure

Tipo	Almacenamiento local	Usando el almacenamiento de Windows Azure a través del tipo StorageAccountInfo	Usando la interface Rest de Windows Azure directamente
Blob	http://127.0.0.1:10000/	http://blob.core.windows.net	http://tablesamplestorage.blob.core.windows.net
Cola	http://127.0.0.1:10001/	http://queue.core.windows.net	http://tablesamplestorage.queue.core.windows.net
Tabla	http://127.0.0.1:10002/	http://table.core.windows.net	http://tablesamplestorage.table.core.windows.net

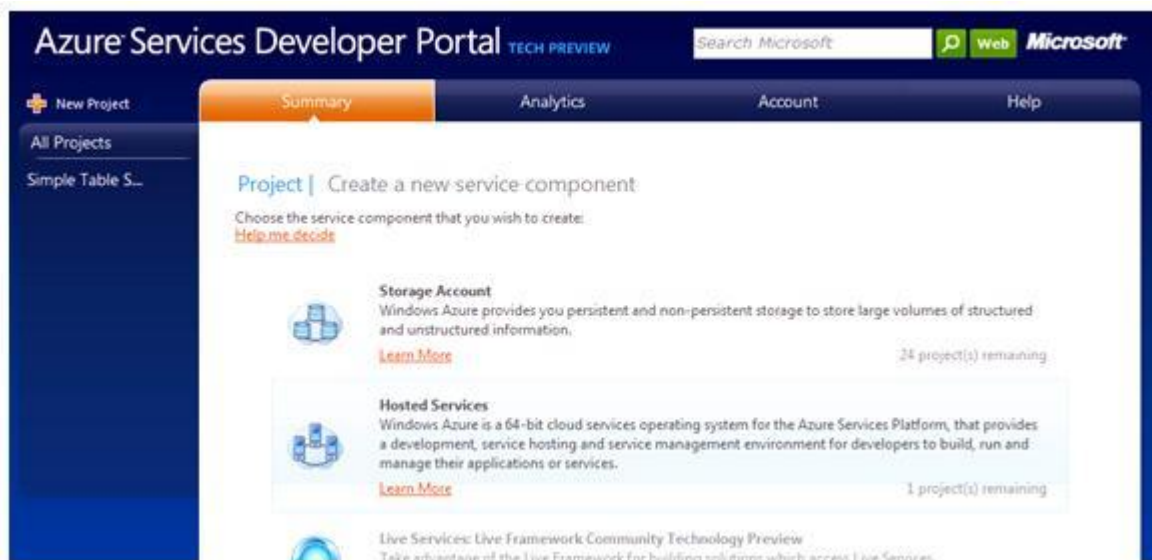
Depurando la aplicación localmente contra el almacenamiento de Windows Azure

Presione F5 para depurar la aplicación y asegúrese de que la misma funciona correctamente contra los servicios de almacenamiento de Windows Azure. Una vez que tiene todo trabajando correctamente con esa configuración, la aplicación esta lista para ser puesta en producción

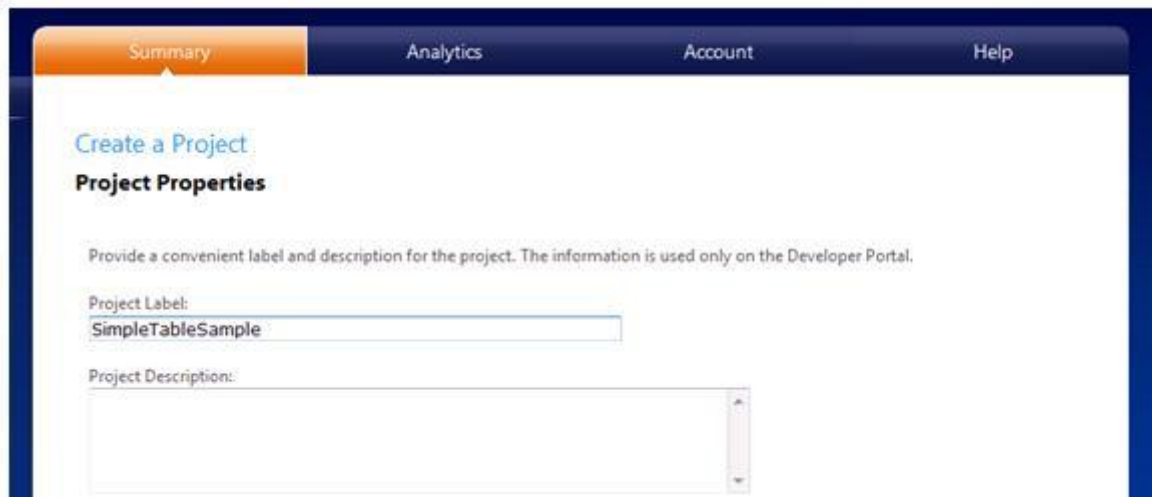
Crear un servicio en la nube

Ingrese al portal de Azure Service Developer (<http://lx.azure.microsoft.com/fs>) y regístrese con su Live ID.

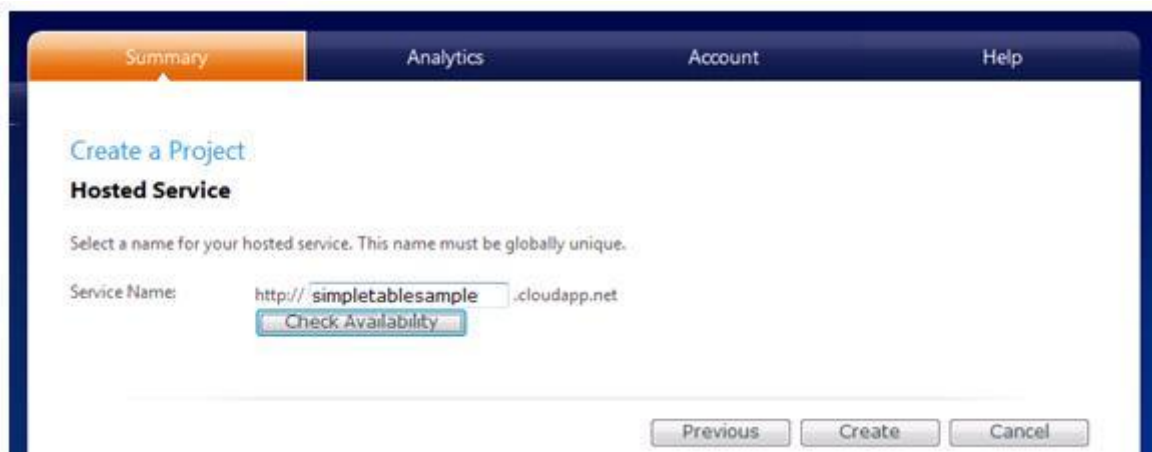
Desde la página del portal, seleccione “New Project” y haga clic en “Hosted Services”.



Esto nos llevara a una página en la que podremos especificar el nombre del proyecto y una descripción para el mismo. El nombre del proyecto es un nombre amigable para identificar al mismo en el portal



Después de hacer un clic en siguiente, deberá ingresar un nombre de dominio para el proyecto (en minúsculas)

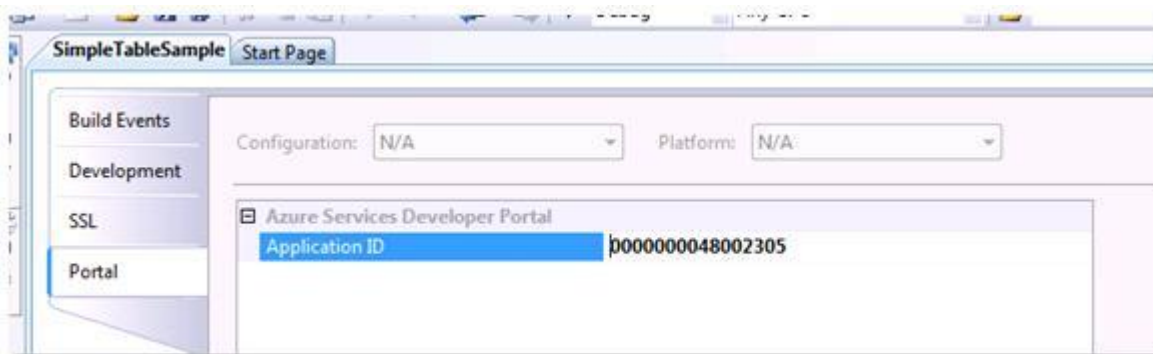
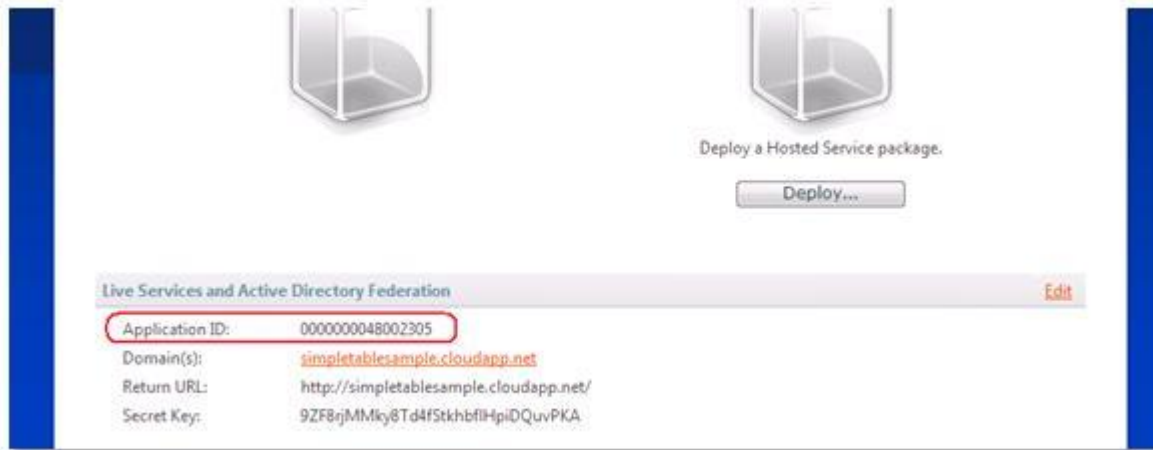


Después de presionar "create" y el proyecto es creado y nos mostrará una página resumen. Esta página muestra el ID de aplicación, el dominio, la URL de retorno, la clave, y botones para poner en producción (Deploy) y el estatus de su puesta en producción.



Configurando la clave en Visual Studio

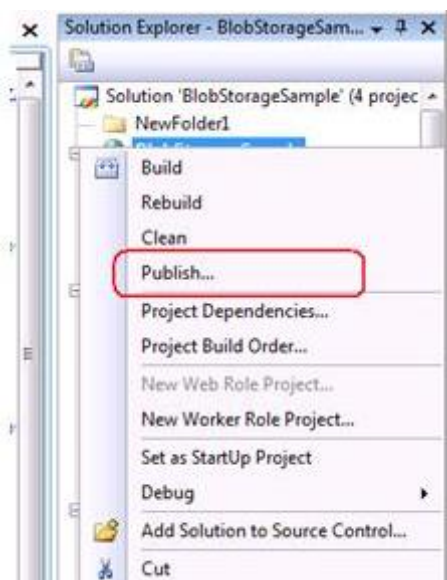
Copie el ID de la aplicación y péguela en el campo "Application ID" en las propiedades del proyecto



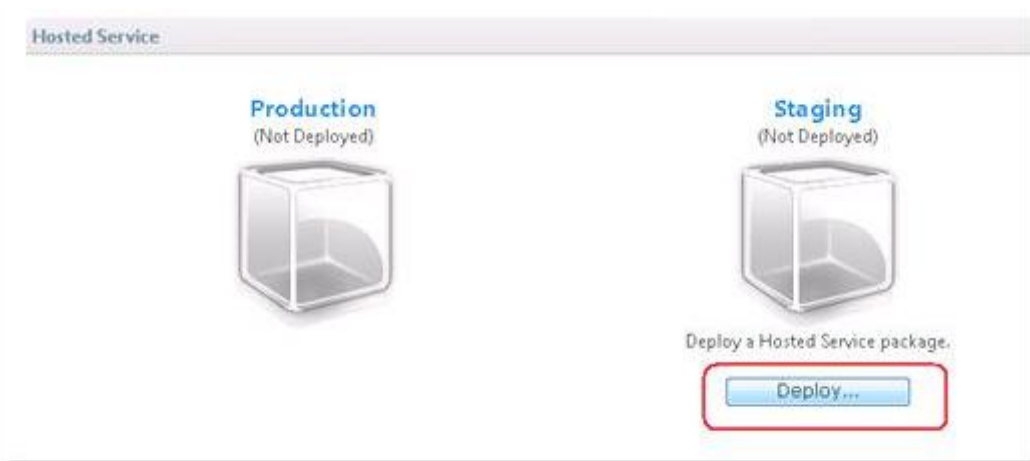
Si usamos el comando publicar (Publish con el botón derecho en el proyecto) en Visual Studio, teniendo el ID de aplicación seteado, tendremos como resultado que Visual Studio nos llevará al portal del desarrollador, y más específicamente a la página del proyecto. Seleccione "Save" para guardar el proyecto.

Puesta en producción desde Visual Studio (Deploy)

Haga clic con el botón derecho en el proyecto y seleccione "Publish...".

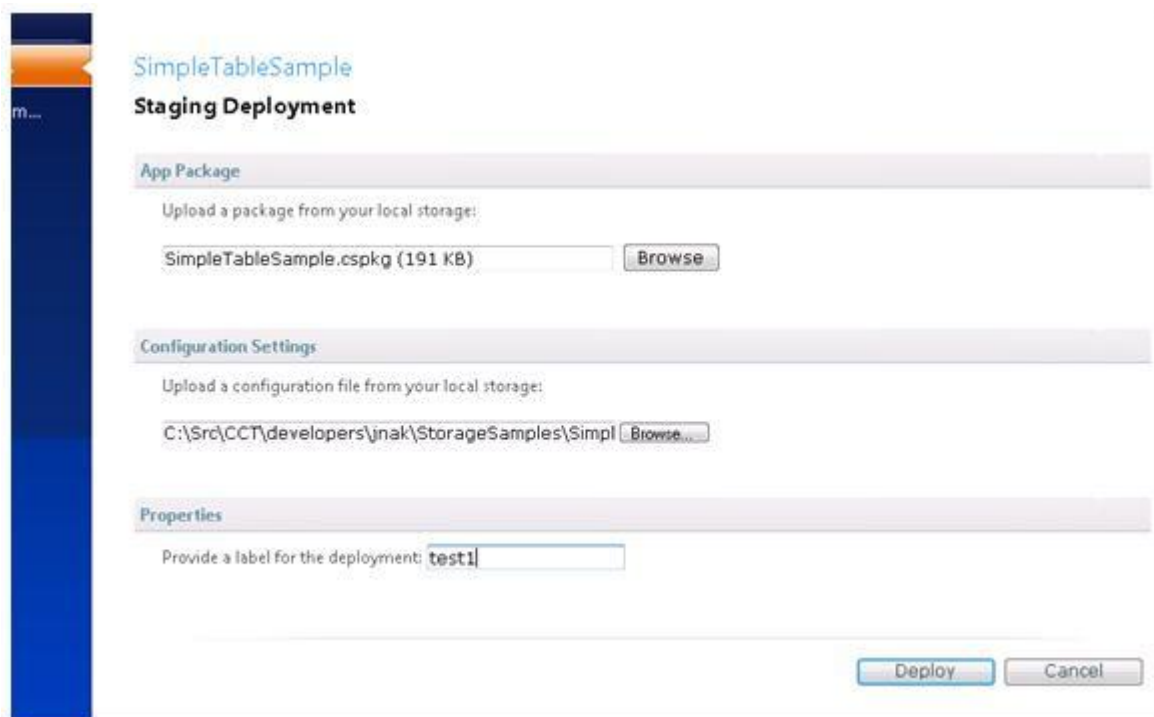


Este comando construirá el paquete, abrirá el explorador de Windows en la ubicación donde el paquete fue construido y además abrirá el portal del desarrollador. En esta página deberá presionar el botón “Deploy”.



Esto nos llevara a una página en donde podemos especificar el archivo del paquete y el archivo de configuración. Puede copiar el path del explorador de Windows que abrió Visual Studio

Nota: Hay una restricción existente para el número de instancias que pueden correr en Windows Azure CTP. El mismo es de 2 instancias de Web Roles y 2 instancias de Worker Roles.



El portal subirá el paquete y pondrá en producción el paquete en “Staging” Esto pone su aplicación en modo “Allocated”

Hosted Service

Production
(Not Deployed)



Staging
test1



Run Configure... Delete

WebRole:
i Allocated 1

Website URL:
754b5f87-99e2-4495-8b02-8d9c15d019ec.cloudapp.net

Private Deployment #:
5bf804a2a5894a13b68ab3efc4432e8e

Desde el modo "Staging", puede testear su aplicación en una URL de prueba (la misma comienza con un GUID) presionando "Run", que pondrá a la aplicación en estado "Initializing".

WebRole:
i Initializing 1

Cuando la aplicación esté lista, el estado cambiará a "Started":

WebRole:
✔ Started 1

Una vez que la aplicación se ha iniciado, puede testearla navegando a la URL de prueba

Después que está conforme con su aplicación, puede promoverla a producción presionando el botón "Promote"

Hosted Service

Production
(Not Deployed)



Staging
test1



Suspend Configure... Delete

WebRole:
✔ Started 1

Website URL:
754b5f87-99e2-4495-8b02-8d9c15d019ec.cloudapp.net

Private Deployment #:
5bf804a2a5894a13b68ab3efc4432e8e

Su aplicación estará en funcionamiento en la URL final.

Production
test1

Suspend... Configure... Delete...

WebRole:
Started 1

Website URL:
simpletable:sample.cloudapp.net

Staging
(Not Deployed)

Deploy a Hosted Service package.

Deploy...

También se puede hacer un clic en el botón “Configure” para copiar el log de eventos o actualizar el archivo de configuración

Service Tuning

Event Logs

Copy the event logs for this deployment to a storage account:

Storage Account:

Container Name:

Configuration Settings

Edit the configuration:

```
<?xml version="1.0" encoding="utf-16"?>
<ServiceConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" serviceName=""
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration">
  <Role name="WebRole">
    <ConfigurationSettings>
      <Setting name="AccountName" value="soluciondeejemplo" />
      <Setting name="AccountSharedKey"
value="ZX7jnJObn/IghebgDKef10LFr1VusJNKBO7isfS2:
WHfB4iz4Ceg==" />
      <Setting name="TableStorageEndpoint"
value="https://table.core.windows.net" />
      <Setting name="BlobStorageEndpoint" value="https://blob.core.windows.net" />
      <Setting name="QueueStorageEndpoint"
value="https://queue.core.windows.net" />
      <Setting name="allowInsecureRemoteEndpoints" value="false" />
    </ConfigurationSettings>
    <Instances count="1" />
  </Role>
</ServiceConfiguration>
```

Upload a new configuration file:

Referencias

Blog de Steve Marx

<http://blog.smarx.com>

Azure Home Page

<http://www.microsoft.com/azure>

Windows Azure Walkthrough: Table Storage

<http://blogs.msdn.com/jnak/archive/2008/10/28/walkthrough-simple-table-storage.aspx>

Deploying a Service on Windows Azure

<http://msdn.microsoft.com/en-us/library/dd203057.aspx>

The Easy Way to Install the Windows Azure Tools and SDK Pre-Requisites

<http://blogs.msdn.com/jnak/archive/2009/04/20/installing-the-windows-azure-tools-and-sdk-the-easy-way.aspx>

Differences Between Development Storage and Windows Azure Storage Services

<http://msdn.microsoft.com/en-us/library/dd320275.aspx>

Introducing the Azure Services Platform, David Chappell

http://download.microsoft.com/download/e/4/3/e43bb484-3b52-4fa8-a9f9-ec60a32954bc/Azure_Services_Platform.pdf

Windows Azure Blobs: Programming Blob Storage

<http://download.microsoft.com/download/D/6/E/D6E0290E-8919-4672-B3F7-56001BDC6BFA/Windows%20Azure%20Blob%20-%20Dec%202008.docx>

Windows Azure Tables: Programming Table Storage

<http://download.microsoft.com/download/3/B/1/3B170FF4-2354-4B2D-B4DC-8FED5F838F6A/Windows%20Azure%20Table%20-%20Dec%202008.docx>

Windows Azure Queues: Programming Queue Storage

<http://download.microsoft.com/download/5/2/D/52D36345-BB08-4518-A024-0AA24D47BD12/Windows%20Azure%20Queue%20-%20Dec%202008.docx>

Agradecimientos

MVP Fabian Imaz – www.siderys.com



Comunidad de Usuarios Microsoft Uruguay



2009

L&A SISTEMAS

www.lyasistemas.com

513-76-13