Setting up Azure and Deploying Your First Application

Learn how to use Windows Azure to setup and deploy your first application hosted in the "cloud". You do not need detailed infrastructure, platform, server, software or business application knowledge to accomplish this user friendly set up.

Windows Azure is Microsoft's entry into cloud computing. If you are unfamiliar with cloud computing, it was explained pretty well at a Microsoft road show event with the following analogy.

Everyone is familiar with electricity. You flip a switch; your lights go on or off. You pay based on that switch and if electricity is being used or not. Electricity is something expected to just be there. You probably don't wonder about the power plant, their operations, transmission lines, sub stations, etc.

Microsoft explains Azure by saying "The Windows® Azure™ Platform (Azure) is an internet-scale cloud services platform hosted in Microsoft data centers, which provides an operating system and a set of developer services that can be used individually or together".

At a basic level, Azure can save costs as users pay a provider for only what they use. You need it, it exists and you pay for it. If you don't need it, turn it off and you don't pay for it. Cloud computing can really offer the most value to a company by taking 'commodity' applications out of the IT burden, letting your resources (infrastructure and personnel) focus on the company's critical line of business applications. The "electric" analogy works at a very high level but is not intended to give a complete picture. It is important to know what cloud computing is not. This analogy might imply that cloud computing is a magical solution where companies can place all their IT infrastructure, applications and trust. Companies will always maintain the need to have applications and infrastructure in house. Cloud computing will not always contain IT elements like company policy, privacy concerns, industry regulations, etc. Some people have already confused the cloud computing concept with other concepts including grid computing (system cluster combined for large functions), utility computing (a resource package similar to a metered electricity service) and autonomic computing (self managing systems).

The following is a walkthrough of getting your Azure CTP account setup so you can get a preview experience in this next evolution in architecture.

First, you must access the website (http://www.microsoft.com/azure/register.mspx) and complete the following steps:

- Register for Azure services
- Download the SDKs and add-ins for Visual Studio
    - Note: The current version is the Microsoft VS July 2009 CTP. It requires Windows Vista SP1 or later operating system with IIS7 to install.
- Training and tutorials

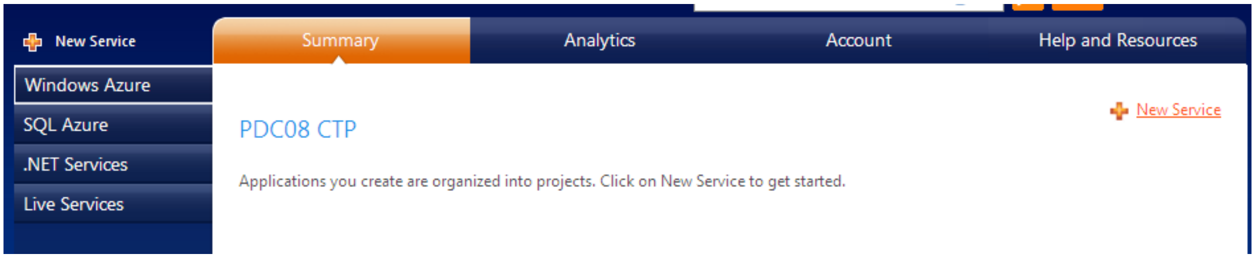
Registering requires that you have a Microsoft Live ID. When you click on the Register for Azure Services link you are brought to connect.microsoft.com to complete your registration. After completing a series of questions and agreeing to user agreements you are brought to the Windows Azure Platform page that talks about invitation codes. This is important because during the CTP you need to get invitation tokens to use the Azure services.

From the Windows Azure Platform page you have the option to request/get a Windows Azure invitation token. As a tip, get this token first before requesting the .NET Services Portal and Live Services Portal tokens. To get those tokens you are navigated away from the Microsoft Connect page and there is no real logical way to get back. When the Windows Azure invitation token is created, copy to Notepad or somewhere you can get it later. This is all you'll need from the Microsoft Connect site.

Please note, as of writing this, the Windows Azure service, .NET Services and Live Services are all available instantaneously after signing up. The SQL Azure service still requires you to wait until it is available. For more information on the Azure services visit: https://www.microsoft.com/azure/services.mspx

Once you have your Windows Azure token you can sign-in Azure site (www.azure.com) and claim your token. Claiming your token is done by clicking on the Account menu option and then the Manage My Tokens link. Copy and paste your token into the provided box and click Claim Token. Once activated you will be taken to the My Projects page under the Summary menu tab where you can create your first project.

Consider projects as the overall solution container for your Azure applications. You can organize your applications and manage services within a project. Your first project will be empty when you click on it in the project's grid of the My Projects page.

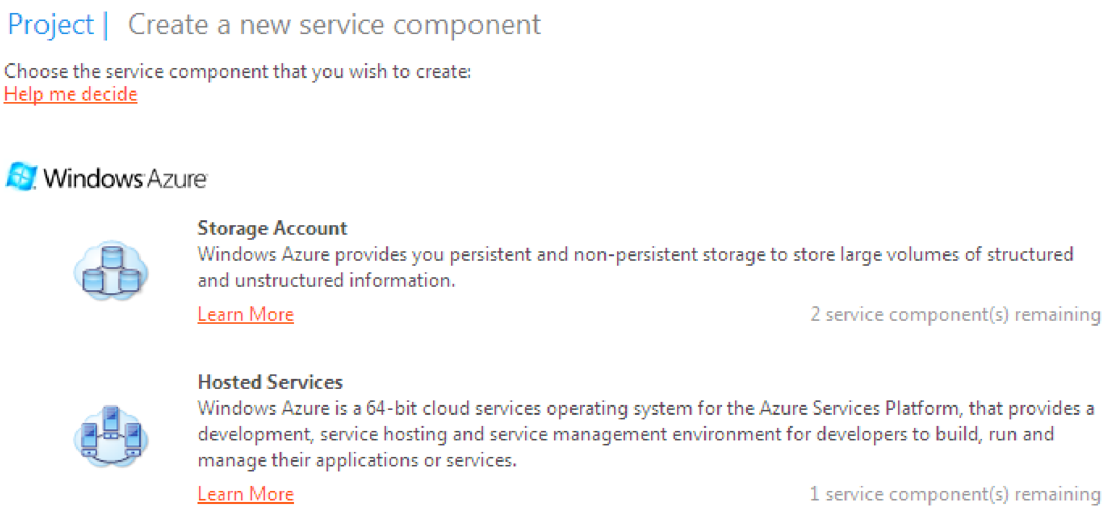


PDC08 CTP

Applications you create are organized into projects. Click on New Service to get started.

Figure 1.1 New Service Selection

The first step will be to add a new service. Click the New Service link in the top right corner of the page.



Project | Create a new service component

Choose the service component that you wish to create:
Help me decide

Windows Azure

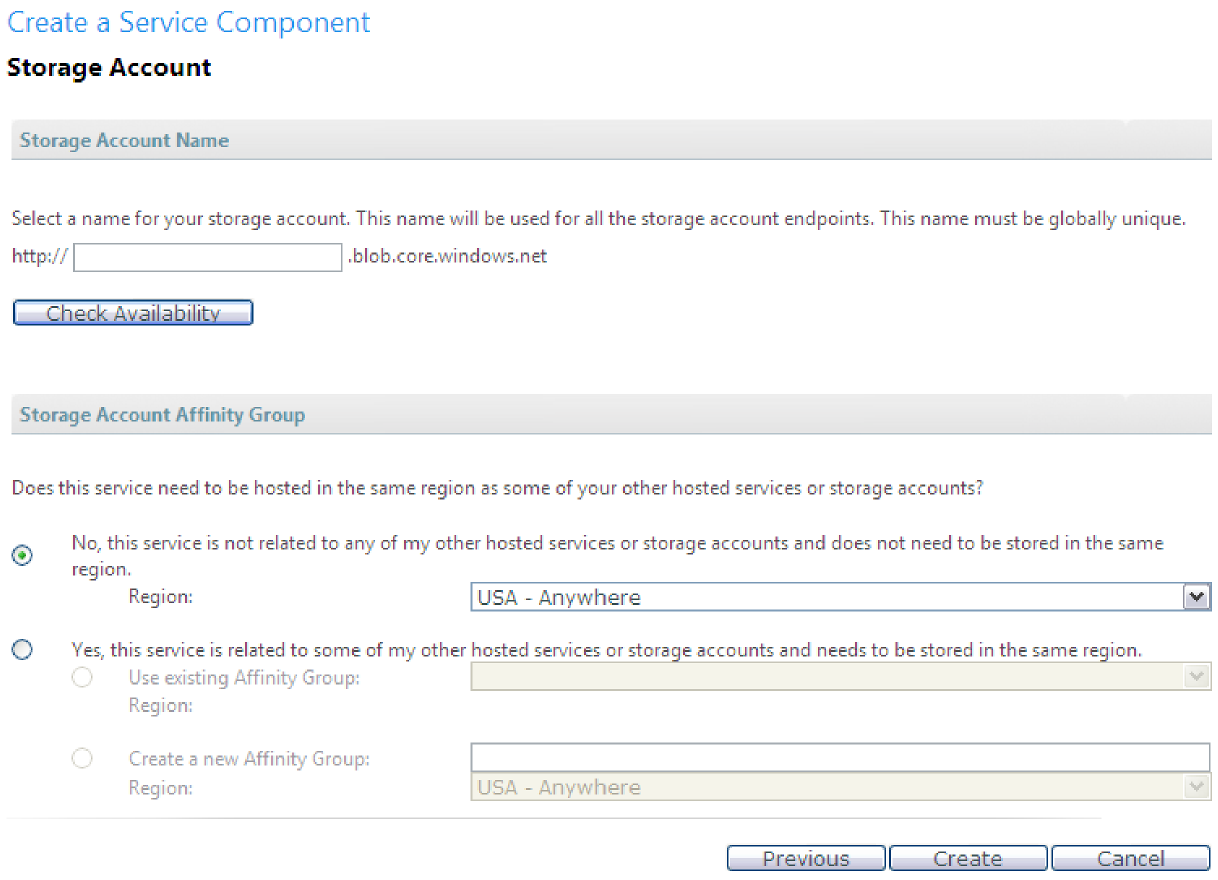
Storage Account
Windows Azure provides you persistent and non-persistent storage to store large volumes of structured and unstructured information.
Learn More 2 service component(s) remaining

Hosted Services
Windows Azure is a 64-bit cloud services operating system for the Azure Services Platform, that provides a development, service hosting and service management environment for developers to build, run and manage their applications or services.
Learn More 1 service component(s) remaining

Figure 1.2 Storage Account/Hosted Services

You will be taken to a page that allows you to create a Storage Account or add Hosted Services. For any application you will typically need some form of persistence or storage. Windows Azure offers storage of large volumes (up to 100TB per account if you were wondering) of structured and unstructured data as part of the Storage Account service. If you plan on storing data in the cloud you will need two key pieces of information from the Storage Account service for your application to access the storage.

Click on the Storage Account and give it a name and description. On the next page you will be asked to provide what is going to become your storage account's URL, or the end point where you access your data in Azure.



Create a Service Component

Storage Account

Storage Account Name

Select a name for your storage account. This name will be used for all the storage account endpoints. This name must be globally unique.

http:// .blob.core.windows.net

Storage Account Affinity Group

Does this service need to be hosted in the same region as some of your other hosted services or storage accounts?

● No, this service is not related to any of my other hosted services or storage accounts and does not need to be stored in the same region.
Region: USA - Anywhere

○ Yes, this service is related to some of my other hosted services or storage accounts and needs to be stored in the same region.
○ Use existing Affinity Group:
Region:
○ Create a new Affinity Group:
Region: USA - Anywhere

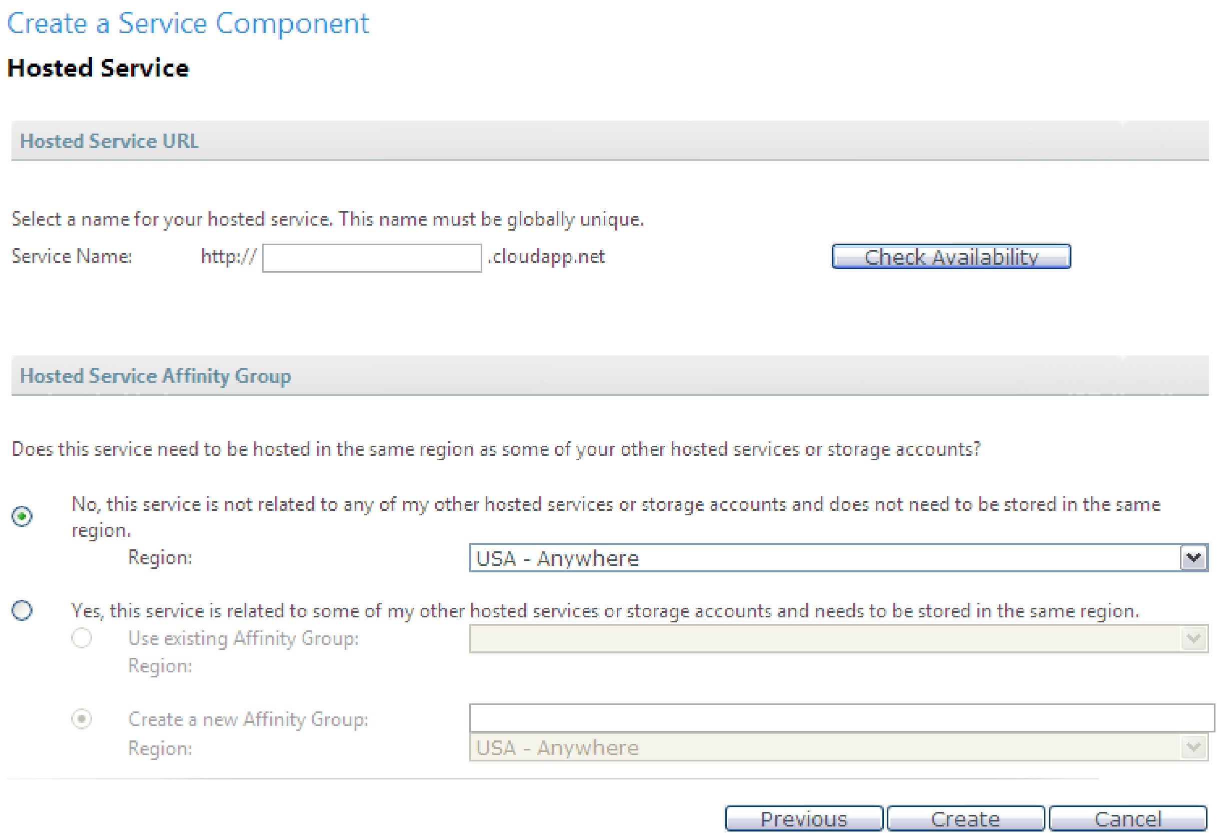
Figure 1.3 Account Creation

Click the Create button to generate the Storage Account. The Storage Account summary page that is generated has two key pieces of information on it that your application will need to know in order to access the data. Your application will need to know the end point URL and the Access Key. To help you get started, check the page links to download the SDK, tools for VS and learning material.

Now that you have a storage account setup, create the Hosted Service, which can be thought of the front end to a web application or web service. Back on the new service page, pick Hosted Services and you will be asked to enter the services name and a description. On the next screen you will be able to setup the application's URL and affinity.

Currently your cloud service's URL is limited to being a sub domain of cloudapp.net (e.g. myappname.cloudapp.net). This condition might leave you wondering if and how you can use your own domain name, like myappname.net. The quick answer is that DNS hosting is not part of the Azure service. Instead, use your existing NDS hosting service and add a CNAME record for your domain to point to your cloudapp.net domain.

Selecting the region/affinity is fairly limited in the CTP with only two main regions, but as the rest of the MS datacenters come online, the options would increase. As you saw in the setup of the Storage Account, both that account and the Hosted Service can be assigned independently to the same or different regions. Keep them in the same region for performance reasons. For assistance, create an Affinity Group which helps you keep storage and hosting services in the same region.



Create a Service Component

Hosted Service

Hosted Service URL

Select a name for your hosted service. This name must be globally unique.

Service Name: http:// .cloudapp.net

Hosted Service Affinity Group

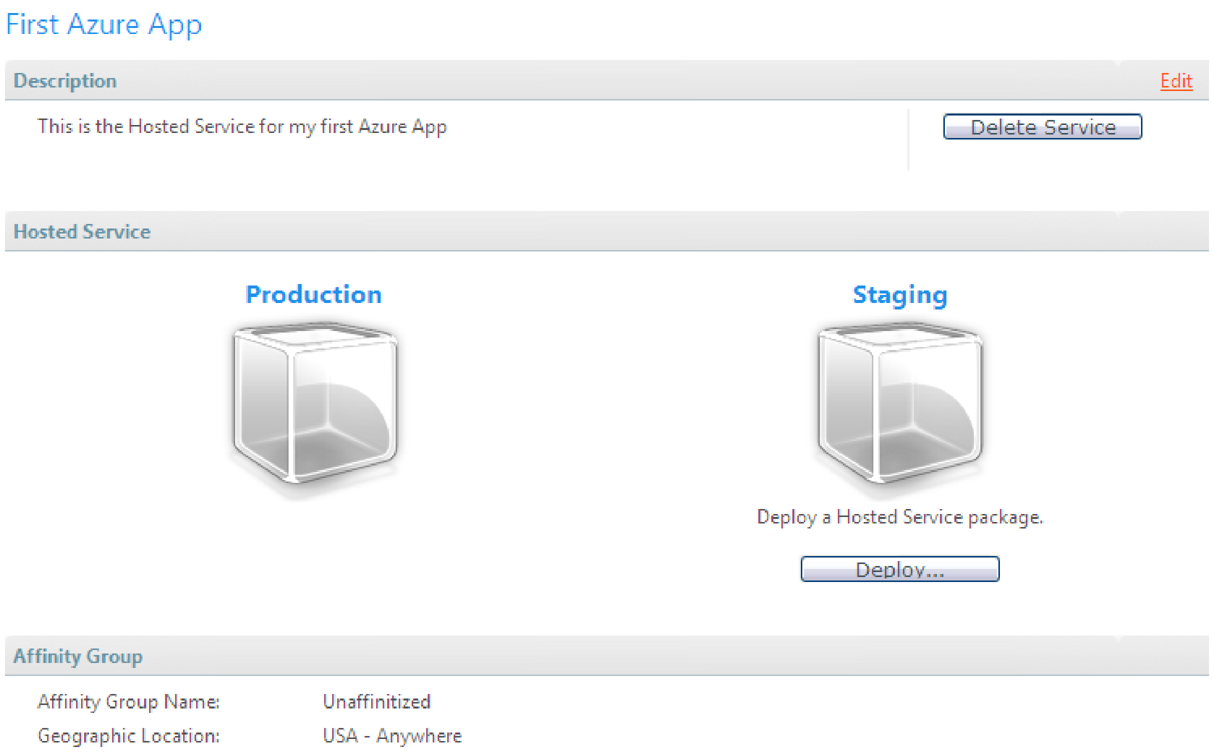
Does this service need to be hosted in the same region as some of your other hosted services or storage accounts?

● No, this service is not related to any of my other hosted services or storage accounts and does not need to be stored in the same region.
Region: USA - Anywhere

○ Yes, this service is related to some of my other hosted services or storage accounts and needs to be stored in the same region.
○ Use existing Affinity Group:
Region:
● Create a new Affinity Group:
Region: USA - Anywhere

Figure 1.4 Service Component Creation

Once your URL and region/affinity is setup, clicking the Create button finalizes the creation process of the hosted service and takes you to the service's summary page. Deployments and migrations from staging to production are initiated from the service's summary page.



First Azure App

Description

This is the Hosted Service for my first Azure App

Hosted Service

Production

Staging

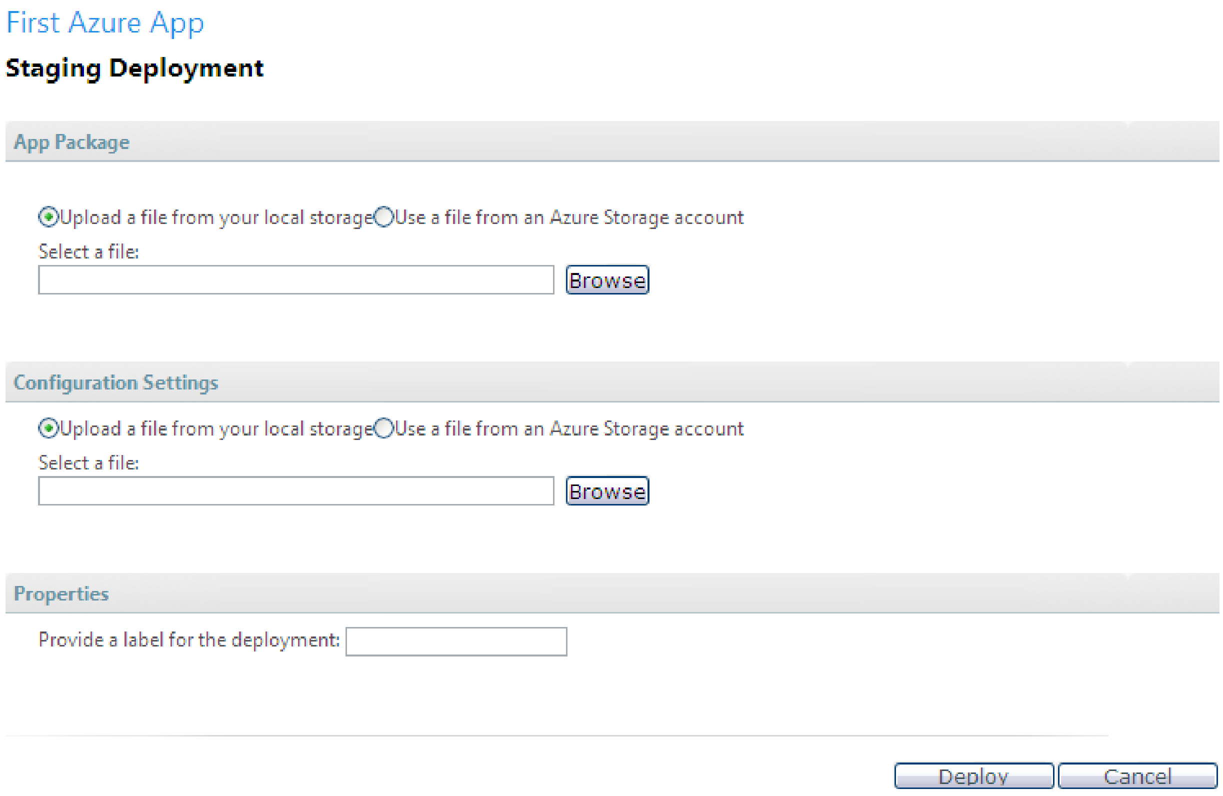
Deploy a Hosted Service package.

Affinity Group

Affinity Group Name:	Unaffinitized
Geographic Location:	USA - Anywhere

Figure 1.5 First Azure Application Deployment

Below staging click on the Deploy button to select the application package and configuration settings for your application.



First Azure App

Staging Deployment

App Package

● Upload a file from your local storage ○ Use a file from an Azure Storage account

Select a file:

Configuration Settings

● Upload a file from your local storage ○ Use a file from an Azure Storage account

Select a file:

Properties

Provide a label for the deployment:

Figure 1.6 First Azure Application Staging Deployment

The package extension for the application will be '.cspkg' and the configuration settings '.cscfg'. Don't worry too much if you are unfamiliar with these extensions since the Visual Studio add-in for Azure does almost all the lifting. This walkthrough uses an app package and configuration settings already built using VS 2008.

The last thing to note on the deployment page is property to "Provide a label for the deployment", which is an easy way for you to keep track of the deployment version. Enter "1.0" and click Deploy to deploy your first application up to the cloud. The initial deployment should only take one to two minutes. During deployment, the uploaded configuration file is used by Azure to copy the application to the correct number of instances (machines).

Immediately following a successful deployment you'll notice that your staging environment is 'lit' now that it contains an application. Under the staging environment you will see detail about how many instance are setup with your application and the private test URL, using a private deployment ID (GUID), you can use to test your application before moving it to production.

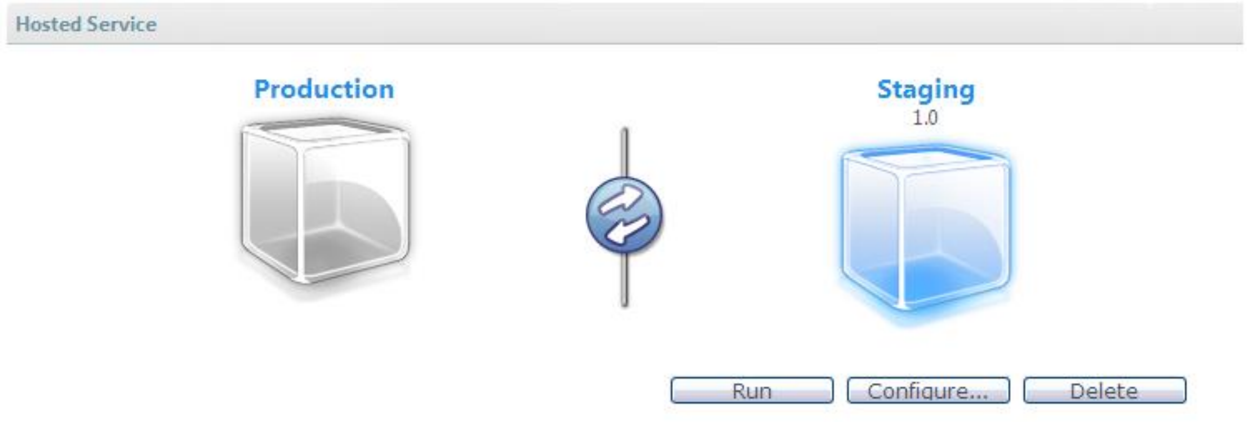


Figure 1.7 Hosted Service - Staging

The Configure button allows you to update your configuration settings after uploading your application. From the configuration editor you are able to change the number of instances the application will use, update the storage account end point URL and access key, and many more things. In the future Microsoft has plans to extend the uses and options in the configuration file.

Once the application is deployed and configured, click the Run button to have Azure start up the configured number of instances (machines). While you wait on Azure to start the servers, it is nice to appreciate at this point that with a few clicks of a mouse button you (really Azure, but you can take credit) have setup a staging environment, deployed your application, provisioned load balancers, networking devices, handled patching of the OS, ensured redundancy against hardware and made sure that plenty of storage was allocated for your needs.

After the application is running and tested you have the ability to move it into your production environment from staging. Since these environments are actually virtual, Microsoft plans in the future to let you have multiple environments like "Dev", "Test", "QA", etc. Clicking the 'sync' button in the middle, between the two environments, moves the application from staging to production.

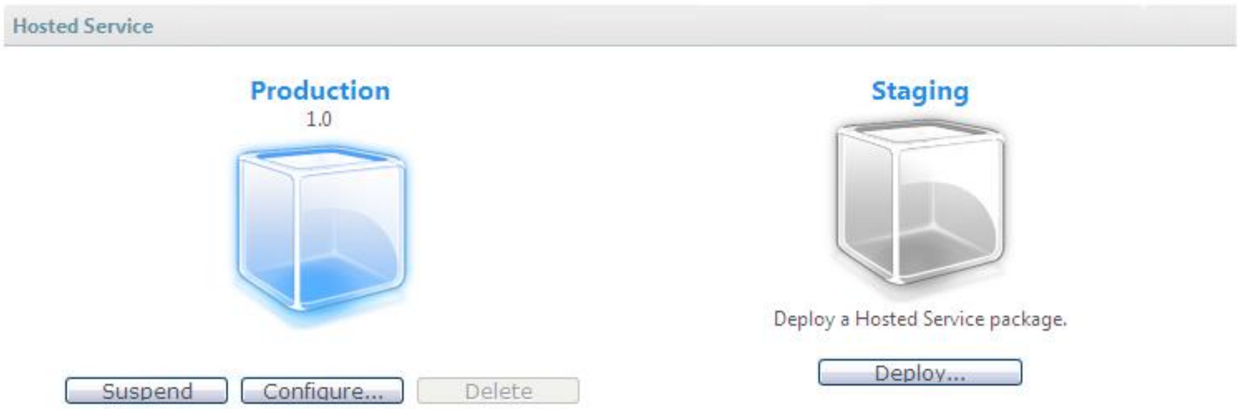


Figure 1.7 Hosted Service - Production

Wait a few minutes for the DNS to propagate before navigating to the web site URL. If you get a 404 page error, wait a few minutes and try again. The time it takes to propagate for the first time varies.

As you can see from the above walkthrough Azure makes it extremely easy to setup scalable environments, deploy your application to those environments and configure it on the fly.

Microsoft Azure creates a unique, comprehensive system ideal for simplified managements and resource organization, which can greatly decrease unnecessary costs. The “pick and choose” service can save time, money and resources while simplifying complex systems into a manageable application.