

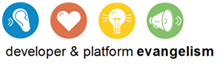
Demo Script

Deploying Windows Azure Services

Lab version: 2.0.0

Last updated: 2/4/2011

Prepared by: Ryan Dunn

* 1. 
  2. Contents

[Overview 3](#_Toc284422743)

[Key Messages 3](#_Toc284422744)

[Key Technologies 3](#_Toc284422745)

[Time Estimates 3](#_Toc284422746)

[Setup and Configuration 4](#_Toc284422747)

[Demo Flow 4](#_Toc284422748)

[Opening Statement 5](#_Toc284422749)

[Step-by-Step Walkthrough 6](#_Toc284422750)

[Deploying the Application 6](#_Toc284422751)

[Summary 14](#_Toc284422752)

[Known Issues 14](#_Toc284422753)



Overview

* 1. This document provides setup documentation, step-by-step instructions, and a written script for showing a demo of Windows Azure. This document can also serve as a tutorial or walkthrough of the technology. In this demo, you will deploy a simple To Do list application to Windows Azure. For additional demos of the Azure Services Platform, please visit <http://www.azure.com>.
     1. **Note:** In order to run through this complete demo, you must have network connectivity and a Live Id. You will also need a token if you wish to show token redemption.

### Key Messages

In this demo you will see two key things:

* 1. How you can easily deploy applications and services to Windows Azure
  2. How you can easily scale applications via the Windows Azure portal

### Key Technologies

This demo uses the following technologies:

* 1. Windows Azure Portal

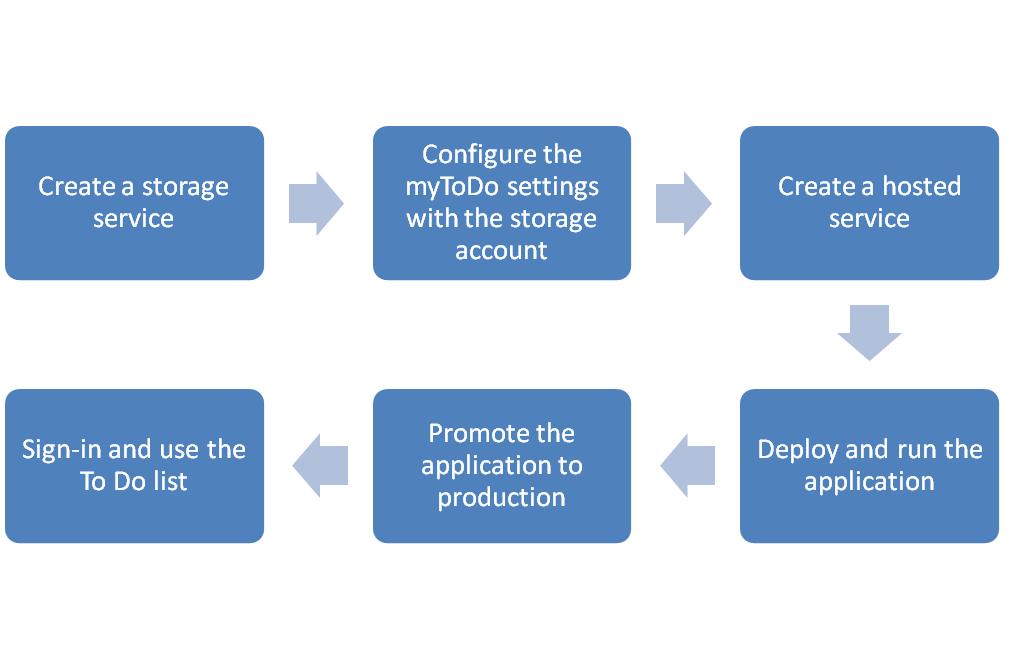
### Time Estimates

* + Estimated time for setting up and configuring the demo: 5 min
  + Estimated time to complete the demo: 10 min

Setup and Configuration

* 1. This demo does not have any advanced configuration requirements. You simply need a web browser and a live id account. You only need to do these steps once.
  2. Find the MyTodo.cspkg and ServiceConfiguration.cscfg files in the **Assets** directory. By default this would be found at **C:\WAPTK\Demos\DeployWindowsAzure\Assets**
  3. We recommend navigating to the Windows Azure portal and save this link on the IE favorites
  4. When demoing this, you should deploy your own copy of myTODO. Due to time constraints or any failure during provisioning, it is best to have a copy ready to show them while the attendees wait

Demo Flow

1. The following diagram illustrates the high-level flow for this demo and the steps involved:
   1. 
   2. Figure 1
   3. Diagram

Opening Statement

In the next 5 minutes, we will show you how to deploy a simple TODO list creation and sharing application into Windows Azure. We will provision servers and network equipment without leaving our chair.

In this simple, Deployment demo you will specifically see two key things:

* 1. How you can easily deploy applications to the cloud without need to provisioning servers, load balances and other data center equipment.
  2. How you can change the number of instances running the service with a few clicks.

Step-by-Step Walkthrough

1. This demo is composed of the following segments:
   * Deploying the application
   1. **Note:** It is best to let the customer drive this demo, you can guide them.

### Deploying the Application

|  |  |  |
| --- | --- | --- |
| Action | Script | Screenshot |
| 1. Open the Web Browser and navigate to the Windows Azure Portal at <http://windows.azure.com> 2. Sign in with the customers live id. | * Let us first navigate to the Windows Azure portal and get you signed in. * We use live id for authentication on the portal – so let’s get you logged in |  |
| 1. Click on the **New Storage Account** button from the Common Tasks ribbon. | * The application we will deploy requires both compute and storage. * First, we will create the storage account. * The storage account will give the application access to queues, blobs and tables. These are not SQL Azure tables, but Windows Azure Tables. |  |
| 1. Choose a subscription. 2. Enter a service label (such as jonitododb) and click **Next**. 3. Make sure the name is available. 4. Select the region. 5. Click **Create**. | * We need to select our subscription where we want to create the hosted service. * We have to give the storage account a name that will also be used as the URL prefix. * It’s good to make the project and dns names for storage the same. However if the name is already taken you will need to think of another name. * We’d like to choose the Anywhere US region. |  |
| 1. Once the storage account is created, select it from the items list. 2. On the properties pane click **View** on the **Primary access key** field. 3. Click **Copy to Clipboard** for the **Primary Access Key** and click **OK**. 4. Paste the key in a notepad. | * The dns name and the primary access key are the 2 pieces of information used when accessing storage. Without the access key, you cannot access storage. * We need these values when we deploy the application. The values can be found in the properties pane after selecting the storage account from the items list. * Retrieve the access key clicking on the View button. Copy the value to the clipboard and paste it on a notepad. |  |
| 1. Use Notepad to edit the **ServiceConfiguration.cscfg** file, located on **Assets** folder. 2. Replace the prompts with the key you copied to notepad earlier and the name of the storage account you created earlier. 3. Increase the **Instances count** number to **2**. 4. Click Save | * Before deploying the hosted service, we have to update the settings for the package to use the storage we have just created. * The configuration file contains 3 things we need to change. * First we copy the account dns name. * Then we copy the shared key. There are placeholders in the configuration file for these. * Next, find the number of instance and change the value 1 to 2. |  |
| 1. Click the **New Hosted Service** button from the **Common Tasks** ribbon. | * Now that we have the storage project, we can create the Hosted service project and deploy the application. * To create the hosted service we click the “New Hosted Service” button from the “Common Tasks” ribbon. |  |
| 1. Choose a subscription. 2. Enter the service name. 3. Enter the URL prefix. 4. Choose a region. | * We need to select the subscription where we want to create the hosted service. * Next, let’s give our service a name. * Now we pick a URL prefix and the dialog will check the availability. * For this demo, we will just choose any region for the deployments, but in practice we can choose the geography for our app as well or use an affinity group. |  |
| 1. Leave the “Deploy to stage environment” option selected and the “Start after successful deployment” check enabled. 2. Set a name for the deployment. 3. For the package location use the Browse Locally button and browse for the **MyToDo.cspkg** file generated by Visual Studio. 4. For the configuration file use the Browse Locally button and browse for the **ServiceConfiguration.cscfg** file generated by Visual Studio. 5. Click **OK**. | * We’ll leave the deployment options by default to let our service deploy to stage environment and start after the deployment. * Now, let’s give the deployment a name and select the package and configuration files location. * To specify the package from our local filesystem we use the Browse Locally button and browse to the service package that we published from Visual Studio. * We also need to browse to the ServiceConfiguration file that was also published from Visual Studio. You might recall that this file defines the roles and number of instances per role. * In this case, we don’t need to add any certificate for our service, so we click OK to start creating and deploying the hosted service. |  |
| 1. Wait for the package to upload and initialize. | * Now, we have to wait a few minutes to upload the package. * Then, the web role will be initialized. * At this point Windows Azure will start up virtual machine instances for each of the roles. * Once the instances started, our application – the assemblies, ASPX pages, etc. that we uploaded earlier – will be deployed into each of the instances. * We need to wait a while until the service becomes ready. |  |
| 1. Click the **Swap VIP** button in the Deployments ribbon. 2. You will be prompted to confirm that you want to switch to the production environment. Press the **OK** button when prompted. 3. **NOTE:** It can take a few minutes to start the production environment. | * Now the service is running in “Staging” this is actually a virtual place as there is no physical difference between staging and production – it depends on where the load balancer is connected. * We can swap from staging to production with a mouse click. |  |
| 1. Once it’s swapped wait a minute before clicking on the URL. 2. Select the node with the deployment name. 3. On the properties pane click on the DNS name link. | * Once the service is in production, give it a few seconds for the dns to propagate. Then you can click the link and be taken to the deployed application.. * By selecting the deployment in the items list we can see its properties in the right pane, clicking on the DNS name we can navigate to our myToDo application. * If you get a 404 error, simply try again in a minute. |  |
| 1. Click Start | * When the service runs for the first time, you have to create an account. |  |
| 1. Have the customer enter their details | * Complete the account details. These are stored in the storage account created earlier. Data is not shared between to do lists. |  |

Summary

* 1. In this demo, you saw how easy it was to create storage and compute accounts, deploy to the cloud, and run applications on Windows Azure.

Known Issues

* 1. Please note the following known issues with this demo:
  + It can sometimes take several minutes to start a Windows Azure application on the cloud.