

EXTREME COMPUTING GROUP

Microsoft

Defining the future.

MapReduce on Windows Azure

Daytona – Deployment Guide

Table of Contents

1	INTRODUCTION	2
2	SETUP DETAILS	2
2.1	CREATE STORAGE ACCOUNT	2
2.2	UPDATE CONFIGURATION SETTINGS	5
2.2.1	<i>Service Configuration</i>	5
2.2.2	<i>ServiceDefinition.csdef</i>	7
2.3	CREATE HOSTED SERVICE	8
2.4	PUBLISH AND DEPLOY SERVICE	9
2.4.1	<i>Deploy using Windows Azure management portal</i>	9
2.4.2	<i>Publish package from Visual Studio</i>	10
2.5	VERIFYING DEPLOYMENT	11
2.5.1	<i>Verify Role Instances</i>	11
2.5.2	<i>Run a sample application</i>	11
2.6	STOP AND DELETE DEPLOYMENT	11

Note: While your Azure deployment is stopped or running unused, it can continue to accrue charges. Please delete a stopped or unused deployment if you no longer require its services and want to ensure you will not be charged for it. For more details please refer to [Windows Azure pricing FAQs](#).

1 Introduction

This document details the steps involved in deploying and running the Daytona MapReduce framework on Windows Azure. It is assumed that the reader has a basic understanding of the architecture and design of Daytona. Please refer to the Daytona Developer Guide for background on the architecture and design of Daytona.

2 Setup Details

2.1 Create storage account

Create a new storage account which will be used by the Daytona service to store information related to applications such as inputs, outputs, results etc. The following steps show how to create a new storage account:

1. Login to [windows azure developer portal](#) and you will be redirected to the landing page.

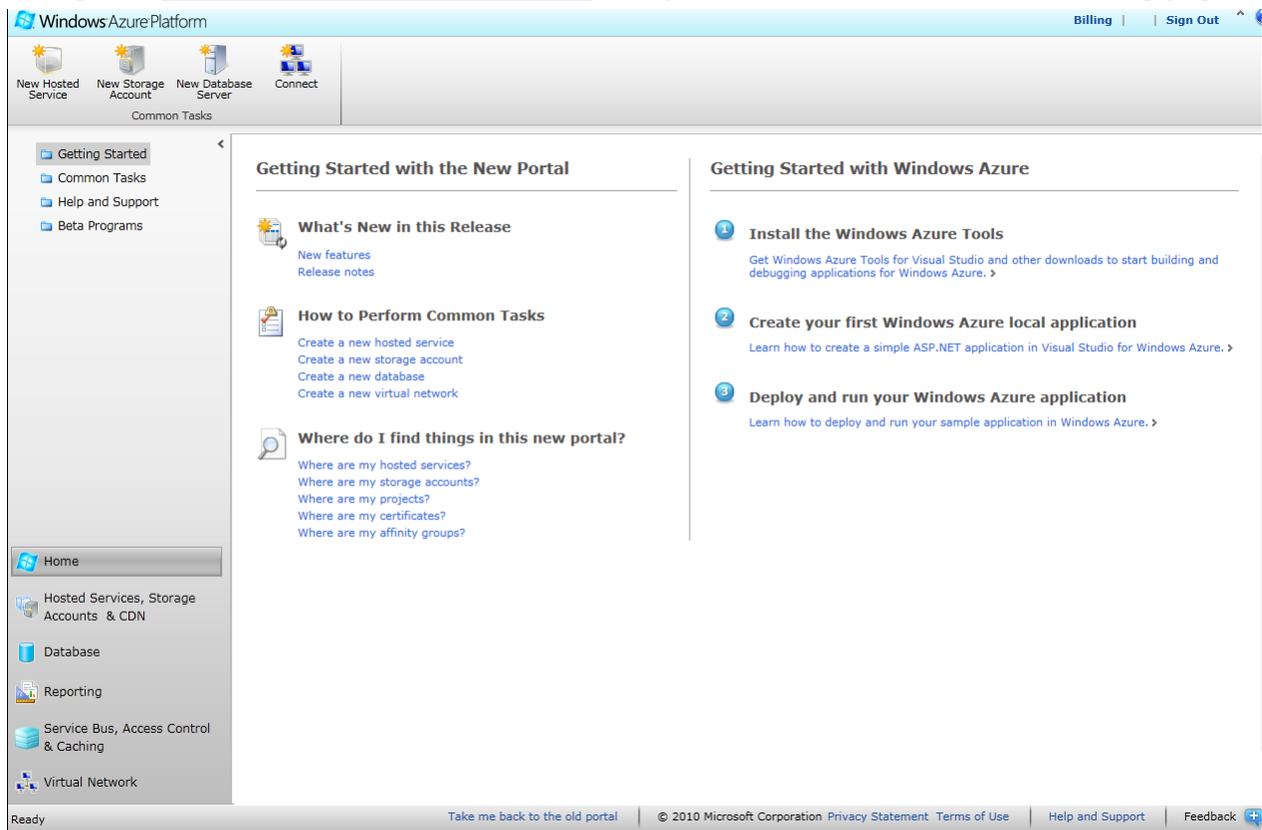


Figure 1 Landing page after login.

2. Click on “New Storage Account” button on the Ribbon.

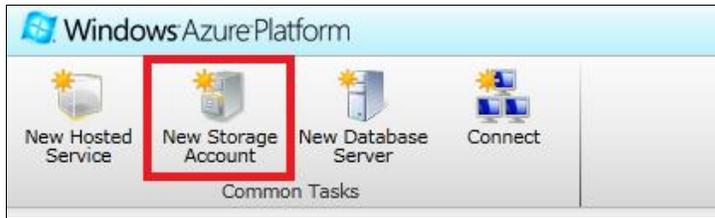


Figure 2 New Storage Account - 1

3. Choose an appropriate subscription and provide a **unique name** for the storage with which it will be identified globally.

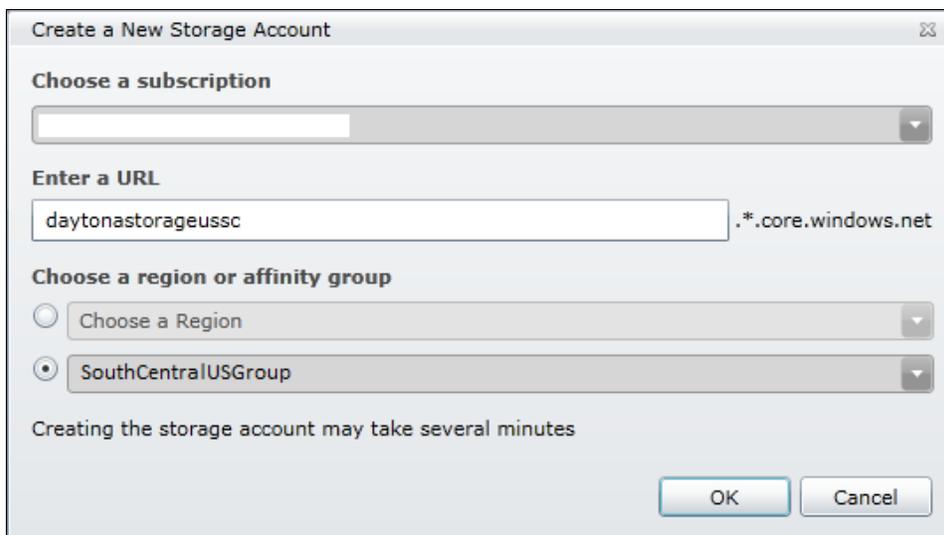


Figure 3 New Storage Account - 2

4. Create an affinity group so that the hosted service can be created in the same geographical location as the storage account.

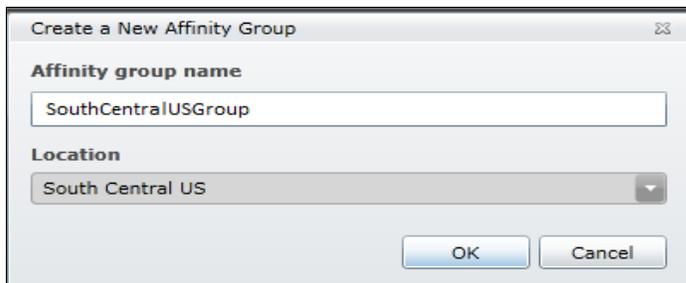


Figure 4 Create Affinity Group

5. Select the storage account from the list. Details can be seen in right pane of the page.

The screenshot displays the Azure portal interface. On the left, a table lists storage accounts under a 'Subscription' header. The table has columns for 'Type', 'Status', and 'Last updated'. One row is highlighted. On the right, the 'Properties' pane shows details for the selected storage account, including access keys, URLs, and other metadata.

Type	Status	Last updated
Subscription	Active	
Storage account	Created	6/30/2011 5:06:36 AM UTC
Storage account	Created	6/30/2011 5:06:36 AM UTC
Storage account	Created	6/30/2011 5:06:45 AM UTC
Storage account	Created	6/30/2011 5:06:53 AM UTC
Storage account	Created	6/30/2011 5:06:53 AM UTC
Storage account	Created	6/30/2011 5:06:54 AM UTC
Storage account	Created	6/30/2011 5:06:54 AM UTC
Storage account	Created	6/30/2011 5:06:54 AM UTC
Storage account	Created	6/30/2011 5:06:54 AM UTC
Storage account	Created	6/30/2011 5:06:55 AM UTC
Storage account	Created	6/30/2011 5:06:55 AM UTC
Storage account	Created	6/30/2011 5:06:55 AM UTC
Storage account	Created	6/30/2011 5:06:55 AM UTC
Storage account	Created	6/30/2011 5:06:55 AM UTC
Storage account	Created	6/30/2011 5:06:55 AM UTC
Storage account	Created	6/30/2011 5:06:57 AM UTC
Storage account	Created	6/30/2011 5:06:55 AM UTC
Storage account	Created	6/30/2011 5:06:55 AM UTC
Storage account	Created	6/30/2011 5:06:56 AM UTC
Storage account	Created	6/30/2011 5:06:56 AM UTC
Storage account	Created	6/30/2011 5:06:56 AM UTC

Properties

- Primary access key**: <Hidden> View
- Secondary access key**: <Hidden> View
- Blob URL**: daytonastorageussc.blob.core.windows.net
- Table URL**: daytonastorageussc.table.core.windows.net
- Queue URL**: daytonastorageussc.queue.core.windows.net
- Name**: daytonastorageussc
- Affinity group name**: SouthCentralUSGroup
- Last updated**: 6/30/2011 5:06:57 AM UTC
- Country/Region**: South Central US
- Status**: Created
- Type**: Storage account

Figure 5 Storage Account Details

6. To view the access keys for the account either click on 'View' button on the right pane under 'Primary Access Key' or click on 'View Access Keys' button on the Ribbon.

The screenshot shows a dialog box titled 'View Storage Access Keys'. It contains two text input fields: 'Primary access key' and 'Secondary access key'. Each field has a copy icon to its right. A 'Close' button is located at the bottom right of the dialog.

Figure 6 View Access Keys

2.2 Update configuration settings

This section describes the changes needs to be done in service configuration and service definition settings. These changes are done in two files `ServiceConfiguration.cscfg` and `ServiceDefinition.csdef`.

Before updating the configuration settings, let's have a brief discussion about Master and Slave roles, it will help to understand the setting parameters.

1. Master – Master role is responsible for picking up new applications from the storage, handling communication with all the slaves, assigning Map and Reduce tasks to available slaves, monitoring task execution etc.
2. Slave – Slave role is responsible for executing assigned map and reduce tasks, handling communication with master as well as other slaves, reporting master about the task execution etc.

2.2.1 Service Configuration

The following configuration settings must be updated in the file 'ServiceConfiguration.cscfg'.

<i>Name</i>	<i>Description</i>
<i>DiagnosticConnectionString</i>	Windows azure storage account connection string which will be used for logging by the worker roles.
<i>StorageConnectionString</i>	Windows azure storage account connection string which will be used for storing the input and output data.
<i>MapTaskSlotSize</i>	The maximum number of map tasks that can be executed at a slave in parallel.
<i>ReduceTaskSlotSize</i>	The maximum number of reduce tasks that can be executed at a slave in parallel.
<i>MapTaskOutputBufferSizeInMB</i>	The maximum size of map output (per task) that can be kept in by a slave. If this size is exceeded, then the output is spilled to the local disk.

<i>ReduceTaskInputBufferSizeInMB</i>	The maximum size of reduce input (per task) that can be kept in by a slave. If this size is exceeded, then the output is spilled to the local disk.
---	---

Table 1 Service Configuration Settings

Also update the instance count of the ‘Slave’ role as per the anticipated load and the number of cores allocated to your azure project. The number of instances for ‘Master’ role must be 1.

The illustration below shows a sample service configuration.

```
<?xml version="1.0" encoding="utf-8"?>
<ServiceConfiguration serviceName="Research.MapReduce.CloudHost"
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConf
iguration" osFamily="1" osVersion="*">
  <Role name="Research.MapReduce.CloudHost.Master">
    <!--Number of master instances must always be kept as 1-->
    <Instances count="1" />
    <ConfigurationSettings>
      <Setting
name="DiagnosticsConnectionString" value="DefaultEndpointsProtocol=http
s;AccountName=XXXXXXXX;AccountKey= XXXXXXXXXXXX"/>
      <Setting name="StorageConnectionString"
value=="DefaultEndpointsProtocol=https;AccountName=XXXXXXXX;AccountKey
= XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX " />
    </ConfigurationSettings>
  </Role>
  <Role name="Research.MapReduce.CloudHost.Slave">
    <Instances count="2" />
    <ConfigurationSettings>
      <Setting name="DiagnosticsConnectionString"
value="DefaultEndpointsProtocol=https;AccountName=XXXXXXXX;AccountKey=
```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX "/>
    <Setting name="StorageConnectionString"
value=="DefaultEndpointsProtocol=https;AccountName=XXXXXXXX;AccountKey
=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"/>
    <Setting name="MapTaskSlotSize" value="4" />
    <Setting name="ReduceTaskSlotSize" value="1" />
    <Setting name="MapTaskOutputBufferSizeInMB" value="128" />
    <Setting name="ReduceTaskInputBufferSizeInMB" value="300" />
</ConfigurationSettings>
</Role>
</ServiceConfiguration>

```

Listing 1 Sample Service Configuration

2.2.2 ServiceDefinition.csdef

The following configuration settings can be updated in the file 'ServiceDefinition.csdef'.

<i>Name</i>	<i>Description</i>
<i>VMSize</i>	VMSize can be separately configured for Master and Slave roles. To update the VMSize of Master role navigate to WorkerRole node with Name attribute is " Research.MapReduce.CloudHost.Master ". For the same node change value of 'VMSize' attribute. For more details on VMSize please refer here . Similar steps can be followed for Slave role instance.
<i>LocalStorage</i>	LocalStorage defines the maximum disk space role can have. To update the local storage value, follow the path – "Worker Role node -> LocalResources -> LocalStorage".

Table 22 Service Definition Settings

Note: Changes to ServiceDefinition file are part of package, hence these changes needs to be done before publishing the package.

2.3 Create Hosted Service

Create a new hosted service onto which the Daytona service will be deployed. The following steps show how to create a new hosted service using [windows azure developer portal](#):

1. Click on 'New Hosted Service' from the Ribbon.

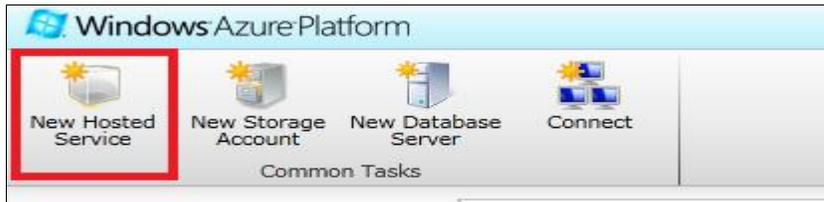
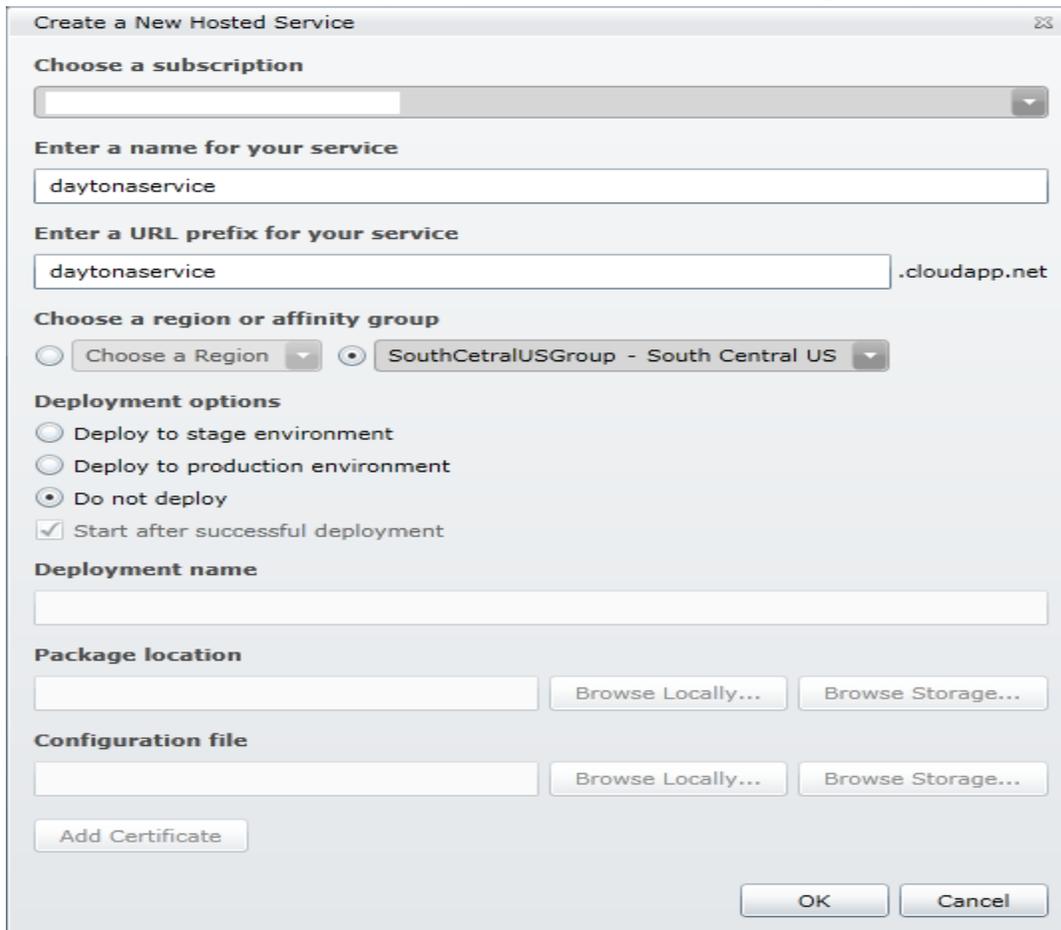


Figure 7 New Hosted Service -1

2. Choose an appropriate subscription. Provide a name which will be used to identify the service within the developer portal (name does not have to be globally unique).

A screenshot of the "Create a New Hosted Service" dialog box. The dialog has a title bar with the text "Create a New Hosted Service" and a close button. The main content area is divided into several sections:

- Choose a subscription:** A dropdown menu with a downward arrow.
- Enter a name for your service:** A text input field containing "daytonaservice".
- Enter a URL prefix for your service:** A text input field containing "daytonaservice" followed by ".cloudapp.net".
- Choose a region or affinity group:** Two radio buttons. The first is "Choose a Region" with a dropdown arrow. The second is "SouthCetralUSGroup - South Central US" with a dropdown arrow.
- Deployment options:** Three radio buttons: "Deploy to stage environment", "Deploy to production environment", and "Do not deploy" (which is selected). A checked checkbox "Start after successful deployment".
- Deployment name:** An empty text input field.
- Package location:** An empty text input field, a "Browse Locally..." button, and a "Browse Storage..." button.
- Configuration file:** An empty text input field, a "Browse Locally..." button, and a "Browse Storage..." button.
- Add Certificate:** A button.

At the bottom right, there are "OK" and "Cancel" buttons.

Figure 8 New Hosted Service - 2

3. Select 'Do not deploy' from 'Deployment options' It will disable the deployment options.
4. Click on 'OK' button. It will create a hosted service.

2.4 Publish and Deploy Service

There are two options to deploy Windows Azure cloud service.

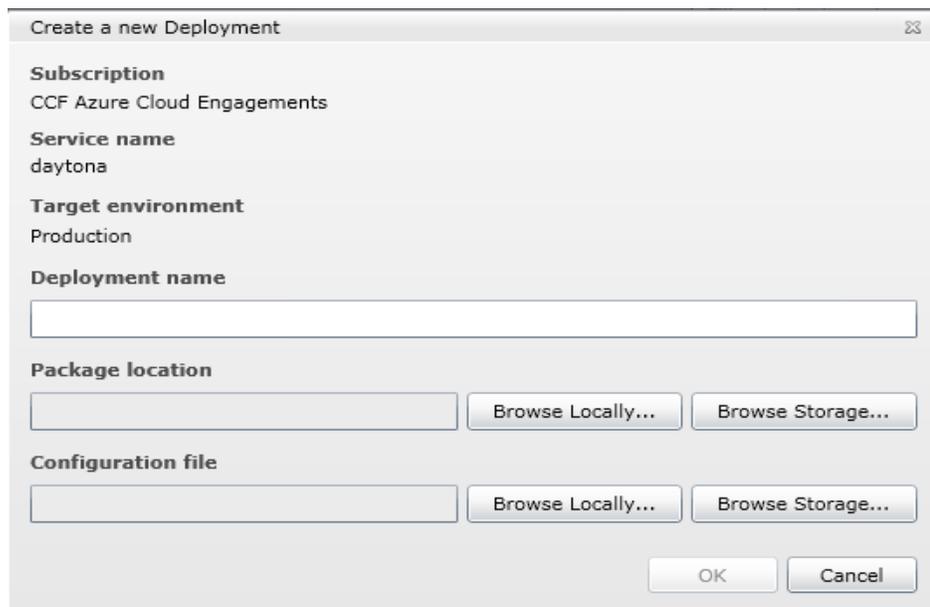
1. Using Windows Azure management portal
2. Using Visual Studio IDE

2.4.1 Deploy using Windows Azure management portal

Daytona release provides **default "Medium" VM size precompiled package** which can be directly uploaded through management portal.

Choose the earlier created hosted service.

Browse release package "Research.MapReduce.CloudHost.cspkg" for the "Package Location" and browse "ServiceConfiguration.cscfg" for "Configuration file". Give name to deployment and click "OK".



The screenshot shows a dialog box titled "Create a new Deployment". It contains the following fields and controls:

- Subscription:** CCF Azure Cloud Engagements
- Service name:** daytona
- Target environment:** Production
- Deployment name:** An empty text input field.
- Package location:** An empty text input field with "Browse Locally..." and "Browse Storage..." buttons to its right.
- Configuration file:** An empty text input field with "Browse Locally..." and "Browse Storage..." buttons to its right.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

This will deploy the cloud service under hosted service as per the configuration settings.

2.4.2 Publish package from Visual Studio

To deploy the service, package and service configuration file is required. It can be generated by publishing the service. To publish and deploy the service, open “Research.MapReduce.CloudHost” project in visual studio (2010) and follow the below steps –

1. Right click on cloud service project node i.e. “Research.MapReduce.CloudHost” and select “Publish” menu.



Figure 9 Publish Service

2. It will open a popup window shown below.

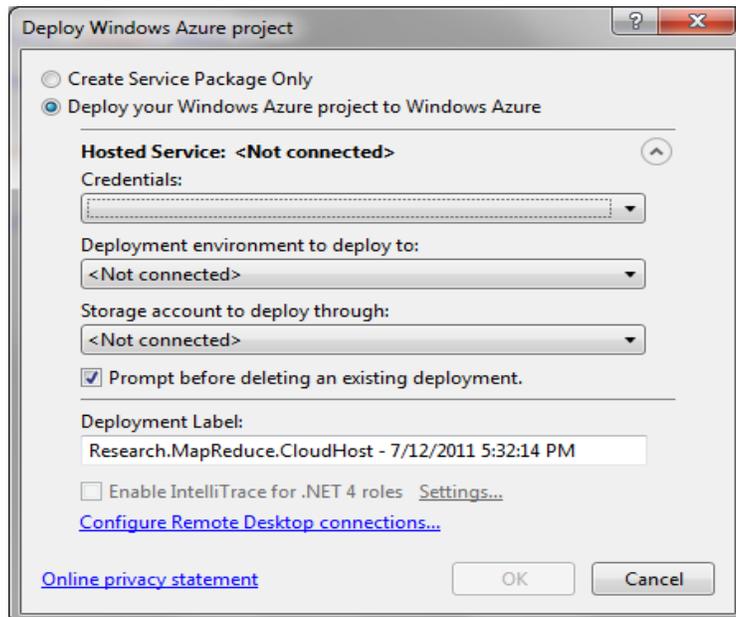


Figure 10 Publish and deploy service

3. “Credentials” are required to deploy the service. Click the Credentials dropdown and select the appropriate credential. To create a new one click on ‘<Add>’ option. For more details refer [here](#).
4. Select “Deployment environment to deploy to” as the <Service Name> - Production. In our case it is “daytonaservice-production”.
5. Select “Storage account to deploy through” as the earlier created storage account. In our case it is “daytonastorageussc”.
6. Change the “Deployment Label” if required.

- Click on “OK” button. It will publish the service and deploy it on the hosted service.

2.5 Verifying Deployment

2.5.1 Verify Role Instances

To verify the status of deployed role instances, expand ‘daytonaservice’ from hosted services list. To check details of any particular instance, select it from the list of instances. Details can be seen in the right pane.



daytonaservice		Hosted Service	Created	
Certificates				
Research.MapReduce.CloudHost - 7/1/2011 3:50:0		Deployment	Ready	Production
Research.MapReduce.CloudHost.Slave				
Research.MapReduce.CloudHost.Slave_IN_0		Instance	Ready	Production
Research.MapReduce.CloudHost.Slave_IN_1		Instance	Ready	Production
Research.MapReduce.CloudHost.Slave_IN_2		Instance	Ready	Production
Research.MapReduce.CloudHost.Slave_IN_3		Instance	Ready	Production
Research.MapReduce.CloudHost.Slave_IN_4		Instance	Ready	Production
Research.MapReduce.CloudHost.Master				
Research.MapReduce.CloudHost.Master_IN_0		Instance	Ready	Production

Figure 11 Verify Deployment

2.5.2 Run a sample application

To verify if the service is functioning properly, go to the WordCount sample in Samples folder. Update App.Config file, to set the ‘MasterConnectionString’ and ‘SlaveConnectionString’ configuration values to those in Daytona’s ServiceConfiguration file. Once the App.Config is updated, run the sample. It will create and submit an application. Verify the progress.

2.6 Stop and Delete Deployment

To stop the running Daytona service, click on the ‘Stop’ button on the Ribbon and wait for the roles to get ‘Stopped’ and then delete the deployment by clicking on the ‘Delete’ button. Please note that if the deployment is left in the ‘Stopped’ state, it will continue to accrue charges.



Figure 12 Stop and Delete Deployment