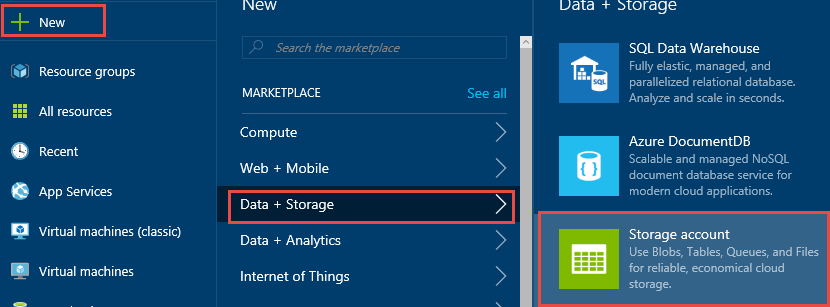
## Configuring Azure Cloud Services

### Demo Overview

In this demo, you will configure and deploy an Azure Cloud Service solution. As part of this exercise, you will create a new Azure Storage Account that will be the repository for Cloud Services diagnostics data. You will configure the application to use the storage account for diagnostics, and then deploy it to Azure. Finally, you will validate and analyze the captured diagnostic data.

### Demo: Create a Storage Account for Azure Diagnostics

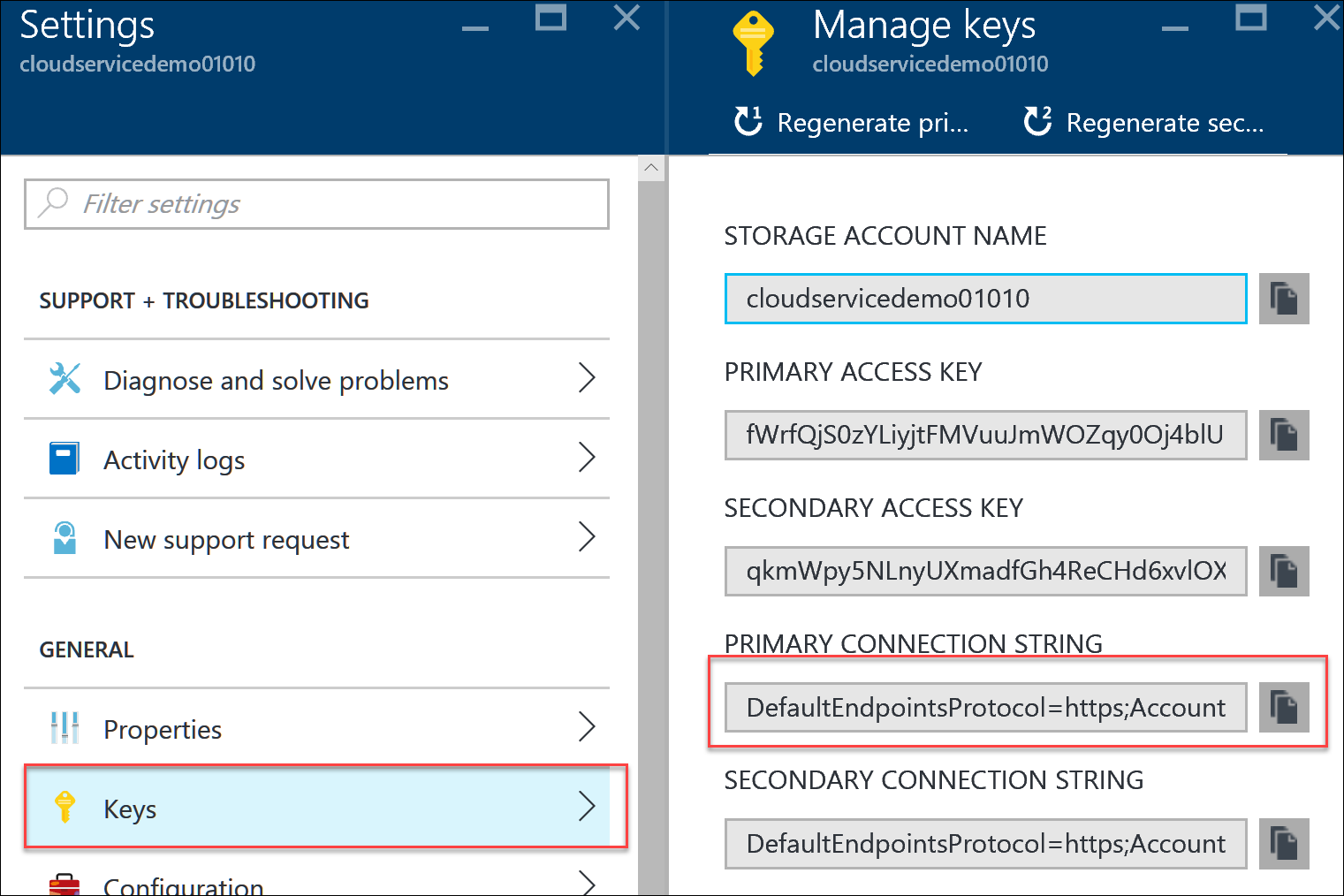
1. In the Azure Management Portal, click **New, Data + Storage**, and then click **Storage Account**.



1. Specify the following configuration on the new storage account blade and click **Create**.

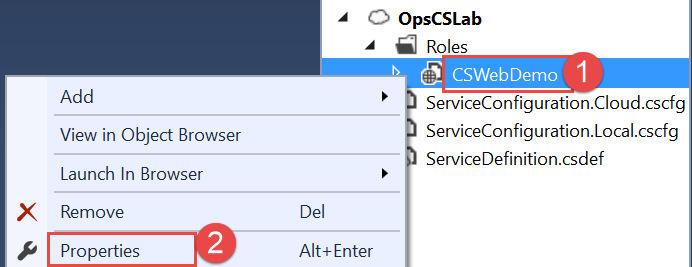
|  |  |
| --- | --- |
| * Name: **A unique name for the storage account.** * Deployment model: **Classic** * Performance: **Standard** * Replication: **Locally-redundant storage (LRS)** * Subscription: **Your subscription** * Resource Group: **Create new – CloudServiceLabRG** * Location: **Specify the closest region to you.** |  |

1. When the Settings blade opens for the storage account, click the **Keys** tile, and then copy the value for **PRIMARY CONNECTION STRING**. Save this value for later.



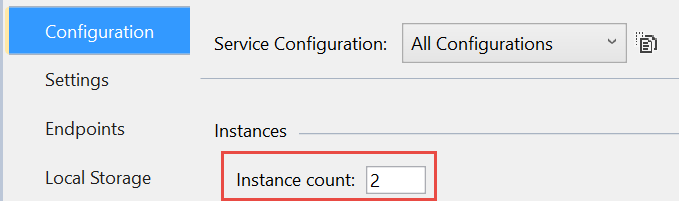
### Demo: Configure the Project for Diagnostics

1. Open the **OpsCSLab.sln** in the C:\OpsgilityTraining\OpsCSLab folder by double clicking the file.
2. Expand the **OpsCSLab** project, expand **Roles**, and then right click on the **CSWebDemo** role and click **Properties**.

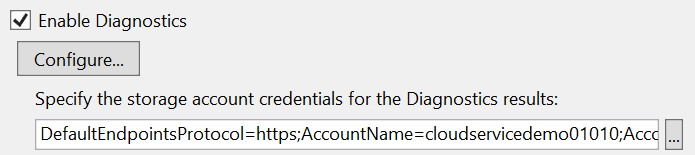


1. On the Configuration tab, change the **Instance count to 2**.

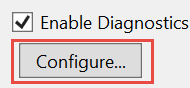
Note: to be eligible for the Microsoft Azure SLA each Cloud Services deployment must have at least 2 instances.



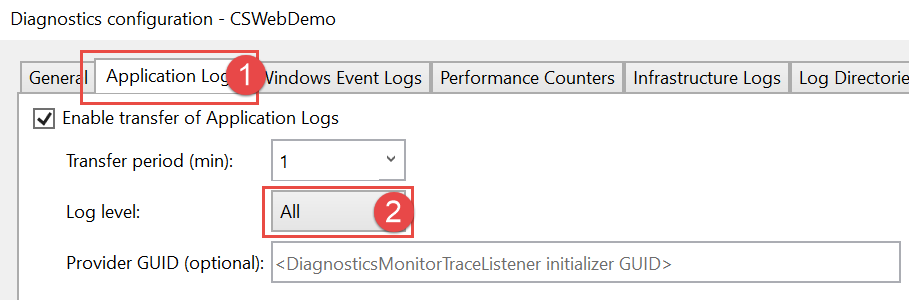
1. On the text box under the Enable Diagnostics checkbox, paste in the storage account connection string created earlier.



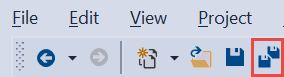
1. Under the Enable Diagnostics section, click the **Configure** button.



1. Click the **Application Logs** tab, and change Log level to **All**, and then click **OK**.



1. Save the modified configuration.



### Demo: Review and Update the Service Configuration

1. Open the **ServiceDefinition.csdef** and **ServiceConfiguration.cloud.cscfg** files and review their contents.
2. In the ServiceDefinition.csdef file, file the <ServiceDefinition element.



1. Go to the end of the line and right before the ending bracket > enter the following text:

upgradeDomainCount="10"

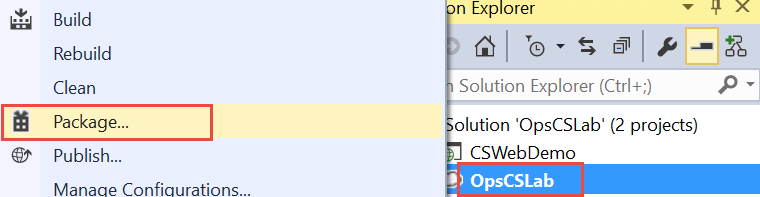
The element ending should look like this:



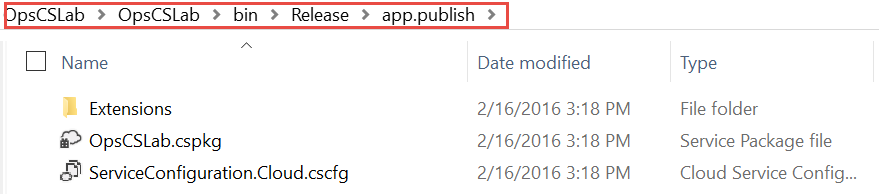
You can specify up to 20 upgrade domains. If not specified, the default number of upgrade domains is 5.

### Demo: Package and Deploy the Cloud Service

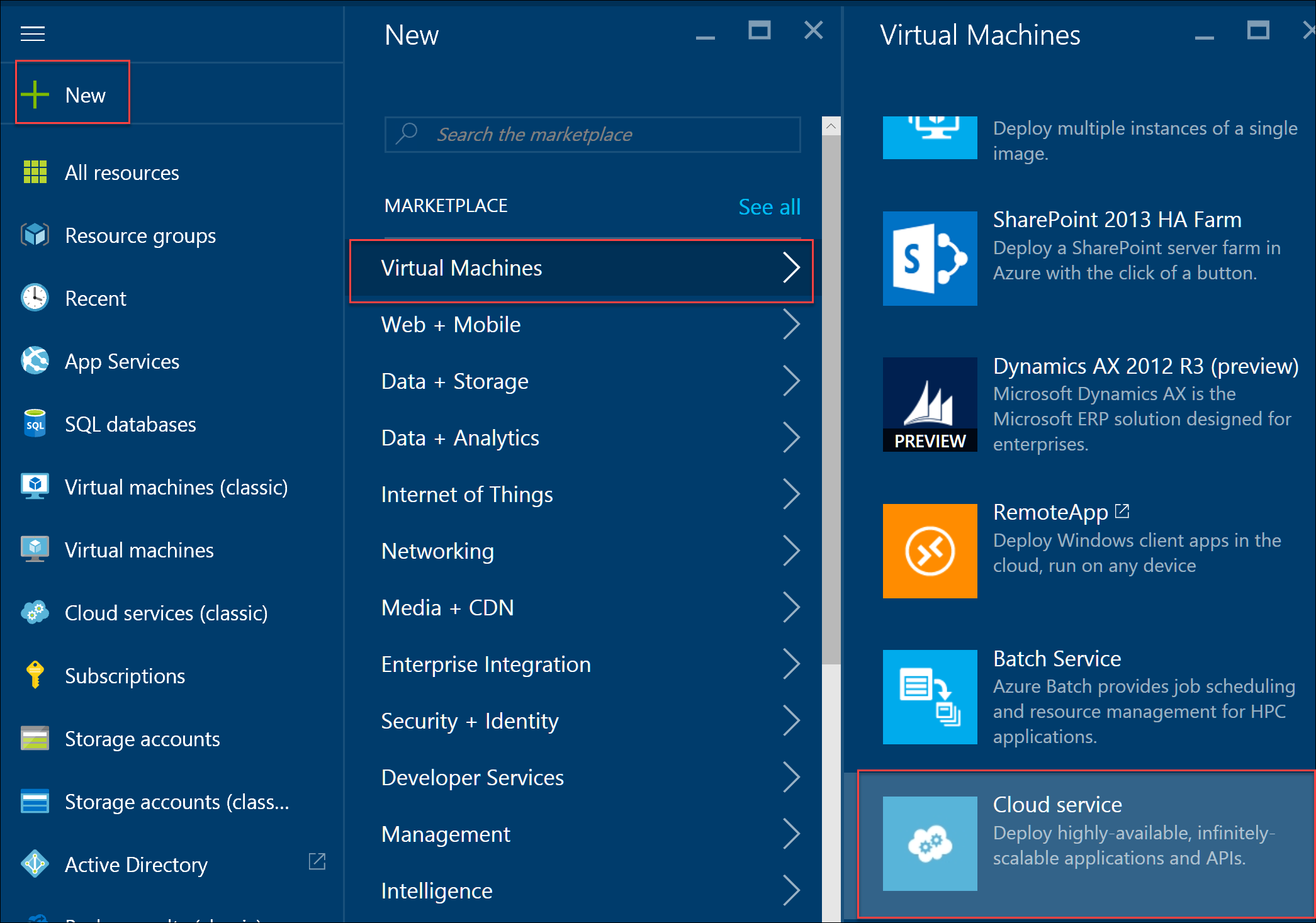
1. Right click the **OpsCSLab** project, and click **Package**.



1. Note the path of the packaged files.



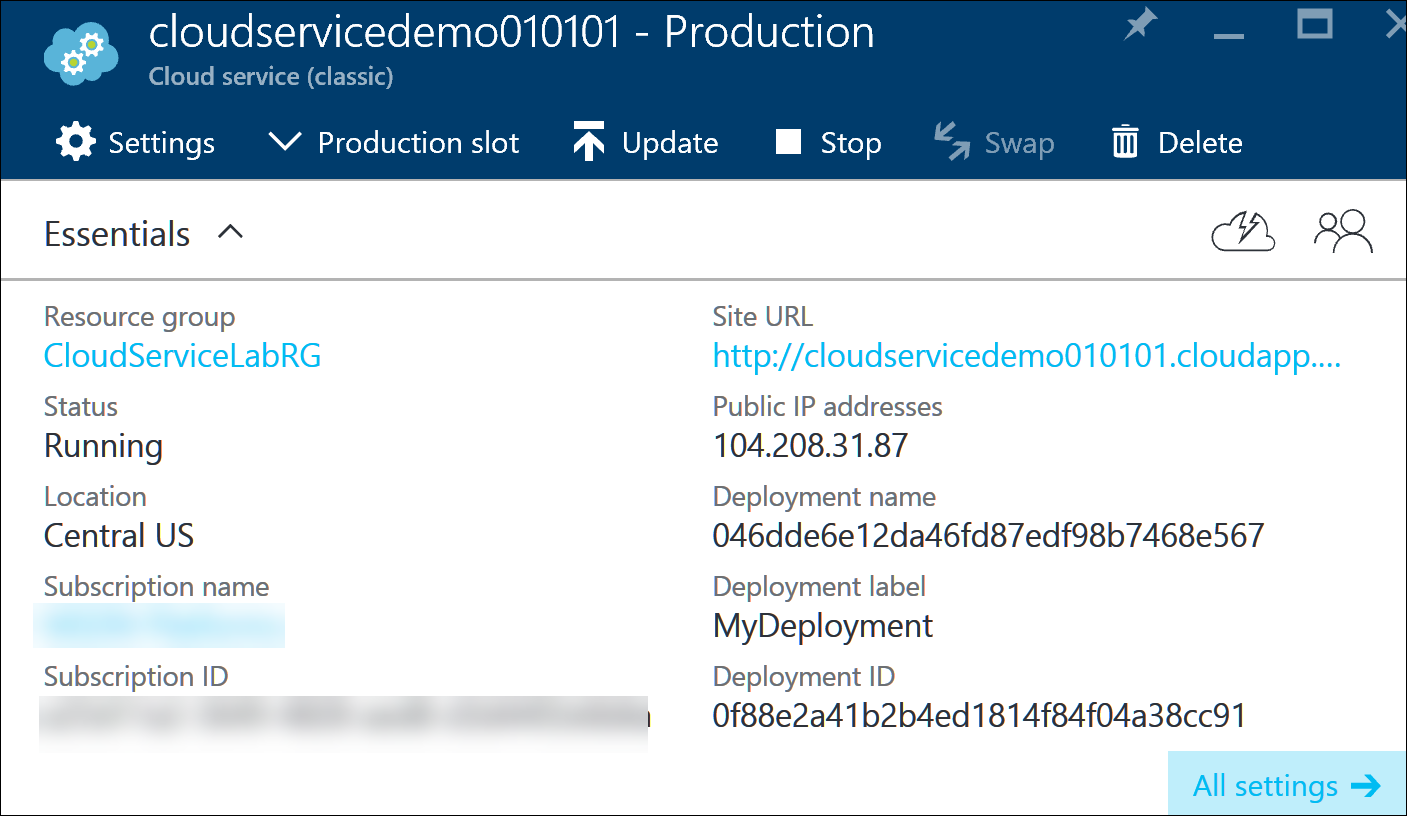
1. Switch to the Azure Management Portal and click **+ New**, **Virtual Machines >** and then click **Cloud Service**.



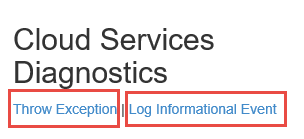
1. Specify the following configuration and then click the **Package** tile:

|  |  |
| --- | --- |
| * DNS Name: **Specify a unique name** * Subscription: **Choose your subscription** * Resource group: **Use existing – CloudServiceLabRG** * Location: **Specify the same region that you created the Azure Storage Account in.** * Click **Package**. * On the package blade, specify **MyDeployment** as the deployment label, and specify the path to the **OpsCSLab.cspkg** and **ServiceConfiguration.Cloud.cscfg** files. |  |

1. Click **OK**, and then click **Create** to start the deployment.
2. After the cloud service is deployed, click the Site URL link on the **Essentials** pane.

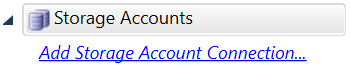


1. After the site is loaded, click the **Throw Exception** link, click back and then click the **Log Informational Event** link to generate some diagnostic data.

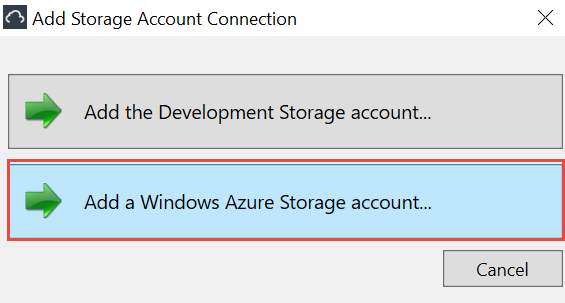


### Demo: Viewing Diagnostics Data

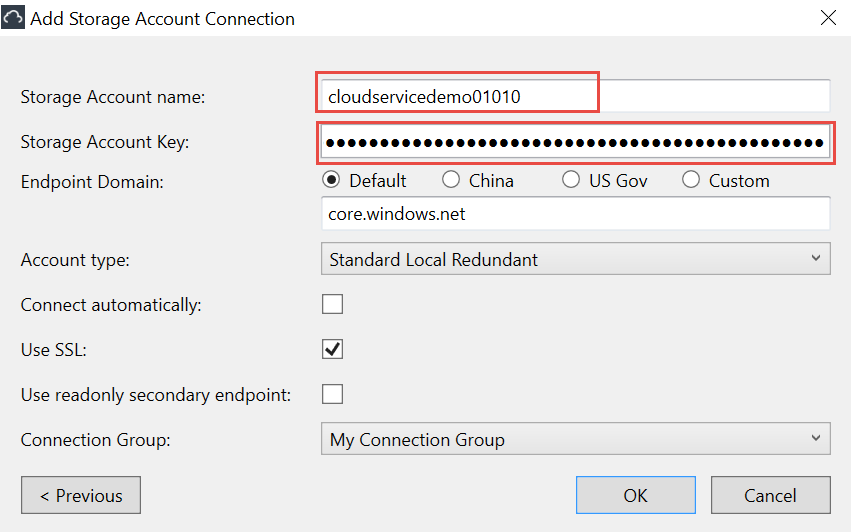
1. Launch Cerebrata’s Azure Management Studio.
2. Click the **Add Storage Account Connection** link



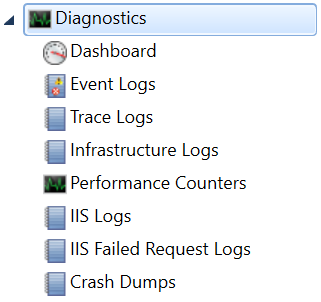
1. Click the Add a **Windows Azure Storage account** button.



1. Specify the **Storage Account Name** and **Storage Account Key** saved earlier and click **OK**.



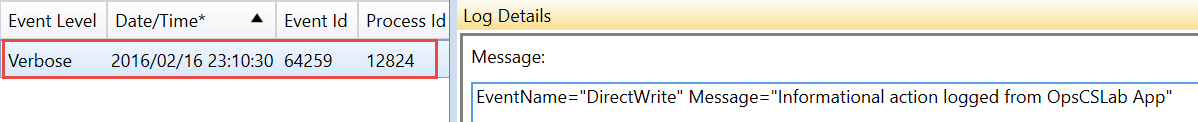
1. Click the **Diagnostics** node to display all of the diagnostics data that can be viewed.



1. Click the **Trace Logs** link to show the trace logs generated by the application.



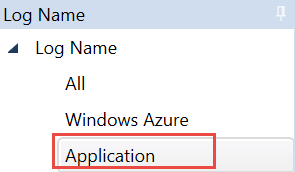
1. Double click the event to see the log details.



1. To find the exception thrown click **Event Logs**.



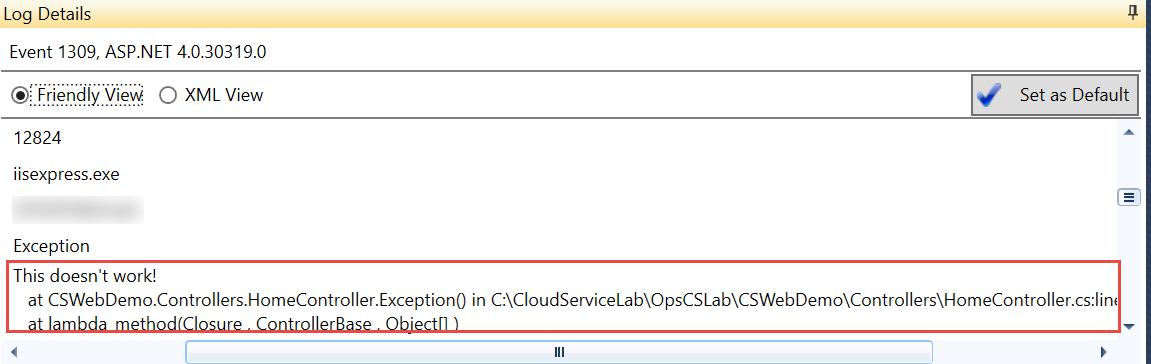
1. Double click **Application** under Log Name.



1. Scroll down and find events generated by ASP.NET and double click the event.

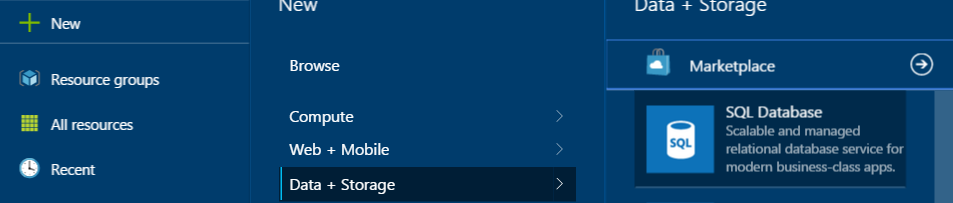


1. You should see the exception text generated by the sample application.

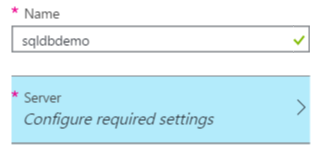


## Demo: Business Continuity and Management

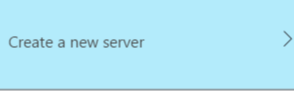
1. Click **New**, **Data + Storage**, **SQL Database**



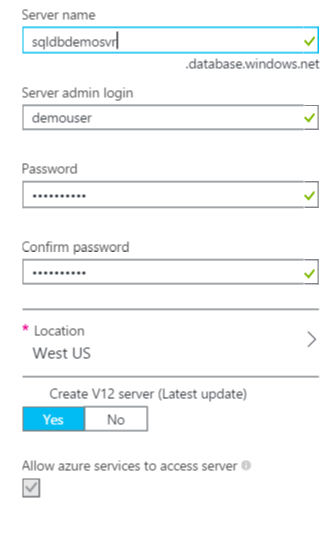
1. Name the database **sqldbdemo**, and click on **Server**.



1. Click **Create a new server**

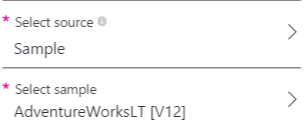


1. Give the server a globally unique name and specify the credentials: **demouser** and **demo@pass123** and select an Azure region to deploy the database server to and Press OK.

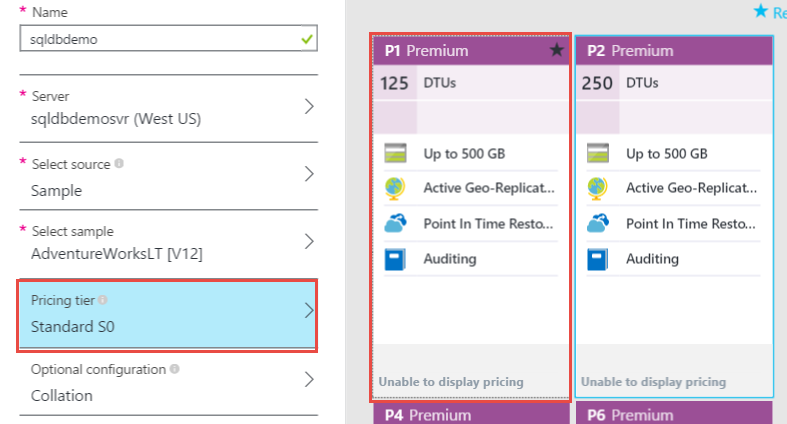


1. Change the Select source from blank database to Sample, the **AdventureWorksLT** database should automatically be selected.





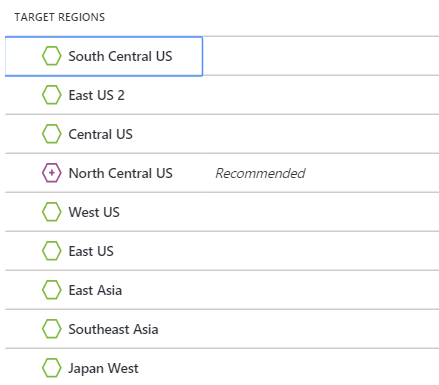
1. Change the pricing tier to **P1 Premium.** This is required for **Active Geo-Replication**.



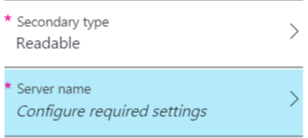
1. Click Resource Group and choose **OPSLABRG** and then click **Create**. This will start the deployment. You will need to wait for this to complete.
2. Navigate to the **OPSLABRG** and find the sqldbdemo database object and click **All Settings.** Then Click the **Geo-Replication role** link.



1. Then select a new region to replicate the database to another Azure Region.

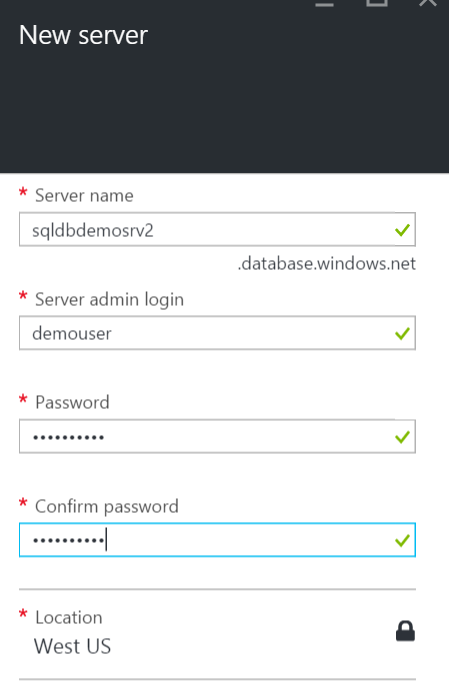


1. After selecting a region, click Server name to create a new SQL Database server in the region to host the database. On the new blade specify a new unique name and click OK.



*Note the Secondary type tile. With SQL DB Premium this secondary can be readable (it is read only) so applications can be designed to take advantage of the secondary. With standard geo-replication this can only be a write only secondary.*

1. Click Create Server and enter a name for the server, Username and Password. Click Select. Then click Ok.



1. At this point the database will initialize and start replicating. Highlight that with active geo-replication you can have up to 4 secondary replicas.

