GLOBAL INTEGRATION BOOTCAMP

Labs Setup Guide

TOMASSO GROENENDIJK
ELDERT GROOTENBOER
STEEF-JAN WIGGERS
ROB FOX
BILL CHESNUT
DAN TOOMEY

Contents

Prerequisites

The Hands-On Labs for this event have been carefully designed and prepared by the organisers to provide you with an opportunity for first-hand experience with the Microsoft integration toolset, using realistic scenarios.

However, it is impossible for us to provide all participants with an environment in which to do these activities. **It is therefore your responsibility to ensure that you have the necessary pre-requisites prepared before you arrive at the event.** This includes the steps in all sections of this document excluding the *Optional* Setups section (which are recommended but not mandatory). If you do not complete these steps in advance of the event, it will be impossible for you to complete the hands-on activities as there simply won't be enough time or bandwidth.

The minimum requirements to setup and run the labs for this Global Integration Bootcamp event are:

- An Azure Subscription (MSDN or Trial)
  - Instruction to create Trial Subscription below
- A developer machine with:
  - Visual Studio 2015 (recommended to have latest updates)
  - Azure .Net SDK (at least 2.9)
- (For Lab 1) A provisioned instance of API Management Service
- (For Lab 3) A developer machine with BizTalk Server 2016 installed and configured
  - A Hyper-V virtual machine has been pre-prepared and can be downloaded as described here; this will save you heaps of time
  - Alternatively, you may choose to setup your own VM, either running in a local virtual machine environment (i.e VMWare) or within Azure.

There are additional requirements for specific labs which are outlined on the next page.

NOTE:

You may use the BizTalk Server machine for all of the development activity required for all labs. However be aware that installing the Azure Logic Apps Enterprise Integration Tools will prevent any BizTalk applications from being developed on this machine (incompatible project types). In this case, the labs do not require BizTalk application development but be aware for future use.

Table of Requirements

The following table shows the requirements that are necessary to complete each of the labs. In case there are any labs that you prefer to skip, you can ignore the indicated requirements for that lab.

Requirement	LAB 1	LAB 2	LAB 3	LAB 4	LAB 5
Azure Subscription	✓	✓	✓	✓	✓
Visual Studio 2015 (Community Edition is fine)	✓	✓	✓		✓
Azure SDK for .NET*	✓	✓	✓		✓
SQL Server 2016		✓	✓		
BizTalk Server 2016			✓		
Microsoft Azure Logic Apps Enterprise Integration Tools for Visual Studio 2015 2.0*		✓			
Service Bus Explorer*	✓		✓		
Google Chrome Postman			✓	✓	
On-Premises Gateway* *(installed and configured as part of Lab 2)*			✓		
Legacy Order System database* *(imported as part of Lab 2)*			✓		
Azure Storage Explorer*				✓	
TableCustomerDiscount.csv*				✓	
PowerBI Account					✓
Device Explorer*					✓

*These installers may also be retrieved from within the **GIB17-SetupFiles.zip** archive that can be downloaded from here.

Naming Conventions

Because so many participants around the globe will be creating the same resources whilst completed the labs, the following naming conventions are suggested to prevent errors due to name clashes:

Resource	Suggested Name*	Sample
Resource Group	gib<loc>17-rgrp-<ini><##>	gibmel17-rgrp-pc01
Service Bus	gib<loc>17-sbus-<ini><##>	gibmel17-sbus-pc01
Storage Account	gib<loc>17st<ini><##>	gibmel17stpc01
API Management	gib<loc>17<ini><##>	gibmel17pc01
API App	gib<loc>17<ini><##>ordersapi	gibmel17pc01ordersapi
Function App	gib<loc>17-func-<ini><##>	gibmel17-func-pc01
Logic Apps Lab 1	gib<loc>17-logic-<ini><##>-validatemaporder	gibmel17-logic-pc01-validatemaporder
Logic Apps Lab 2	gib<loc>17-logic-<ini><##>- storeorderonprem	gibmel17-logic-pc01-storeorderonprem
Logic Apps Lab 3	gib<loc>17-logic-<ini><##>-procbusinesscustorder	gibmel17-logic-pc01-procbusinesscustorder
IoT Hub	gib<loc>17-ioth-<ini><##>	gibmel17-ioth-pc01
Document DB	gib<loc>17-docdb-<ini><##>	gibmel17-docdb-pc01

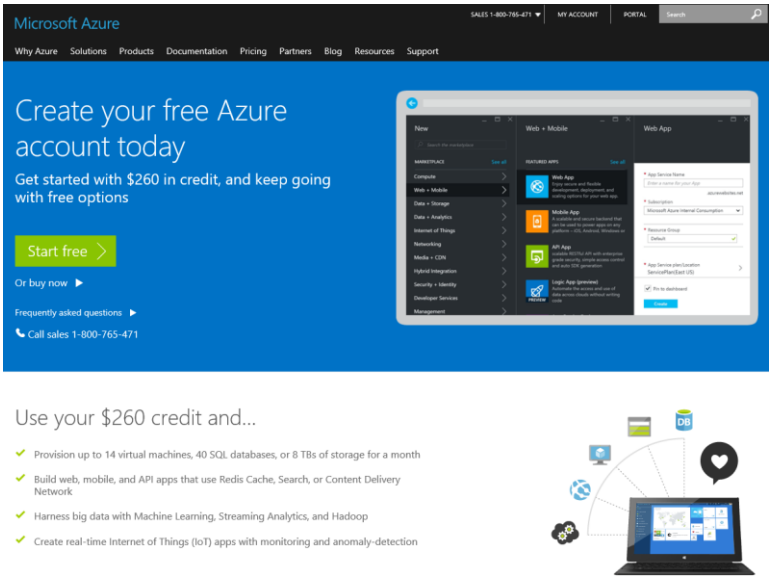
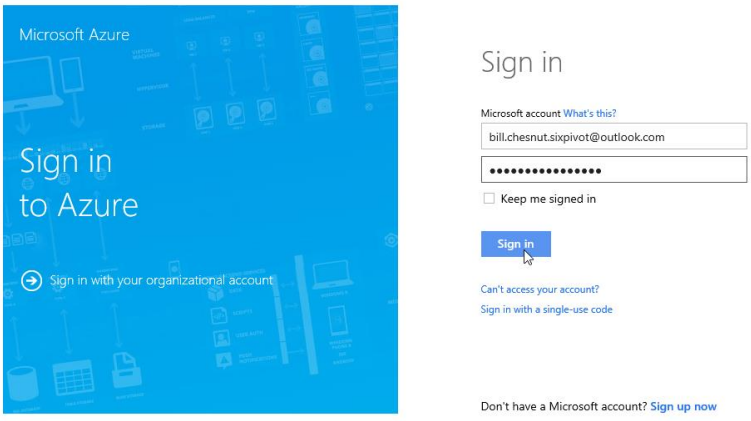
* Replace <loc> for a 3 char acronym of your location
* Replace <ini> for your 2 initials
* Replace <##> for 2 random numbers for uniqueness

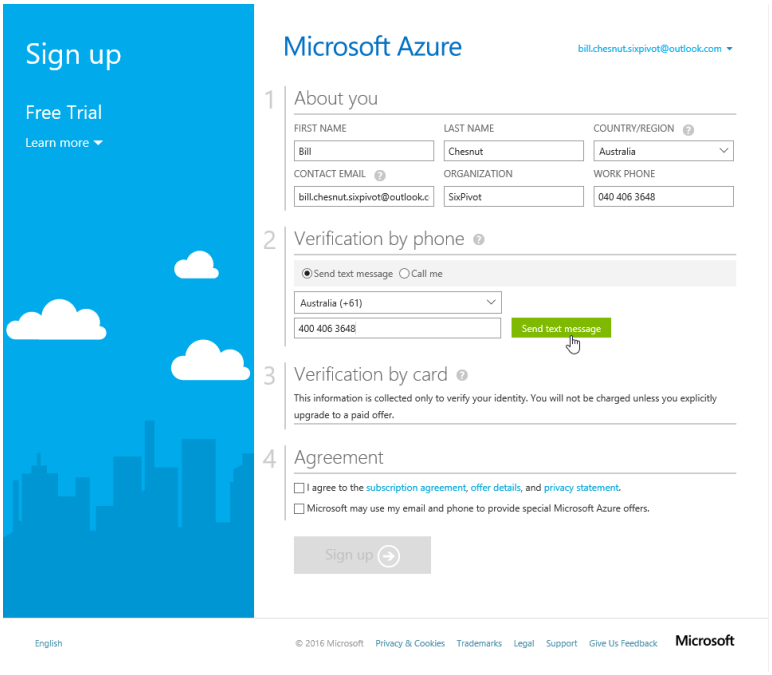
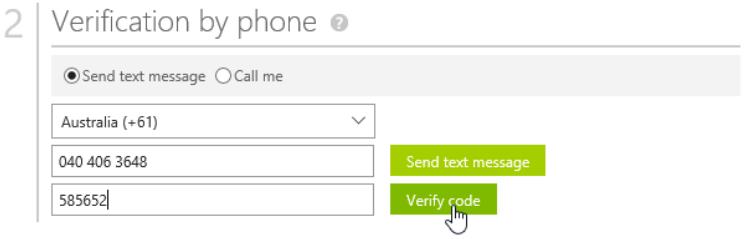
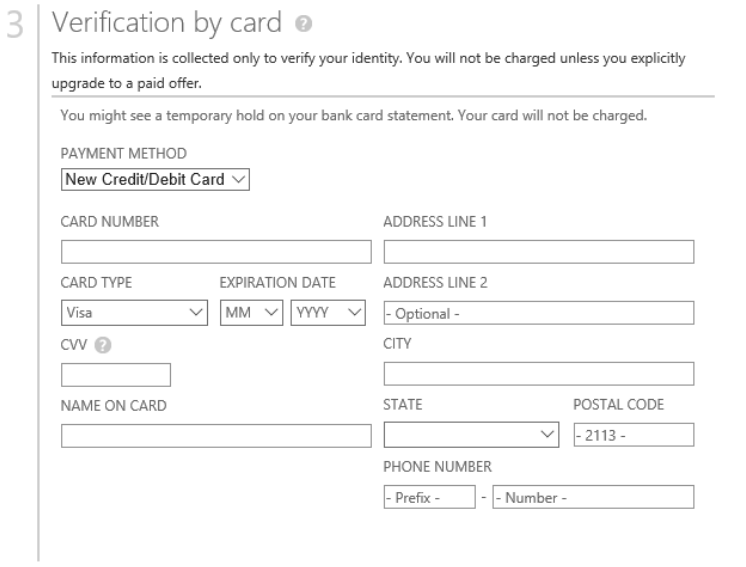
It is also suggested that everything be created under a single resource group (or perhaps on resource group for each lab). The reasoning behind this is that it will be far easier to "clean up" artefacts after the event to save unnecessary charges, as you can delete the entire resource group which deletes everything in it as well.

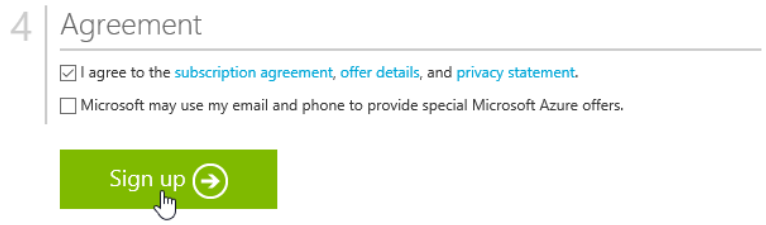
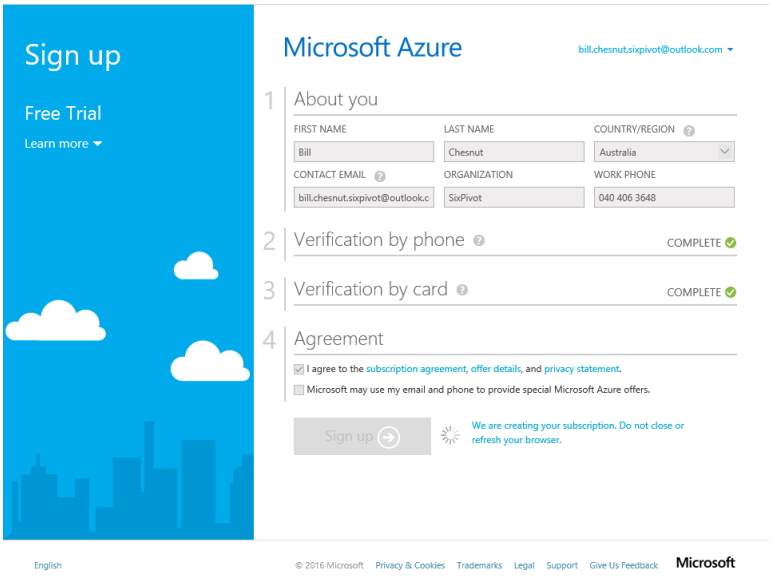
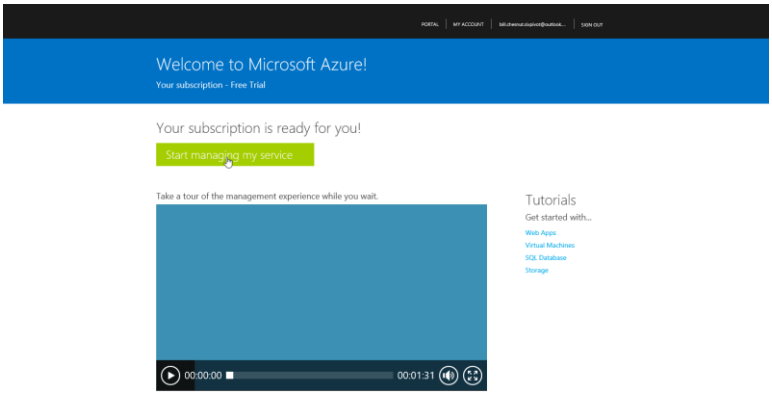
Creating an Azure Trial Subscription

If you already have an MSDN Azure subscription and do not prefer to use a trial subscription, you may skip this section.

In order to do the labs for this course it is necessary to have an Azure Subscription. Below are instructions for setting up a trial subscription.

Go to: https://azure.microsoft.com/en-us/free/ Click 'Start free'	
Sign In with a Microsoft Account that is not an owner of any Azure Subscriptions Click 'Sign in'	

Fill in your details Click ‘Send text message’	
Enter Verify Code received Click ‘Verify code’	
Enter Card Details	

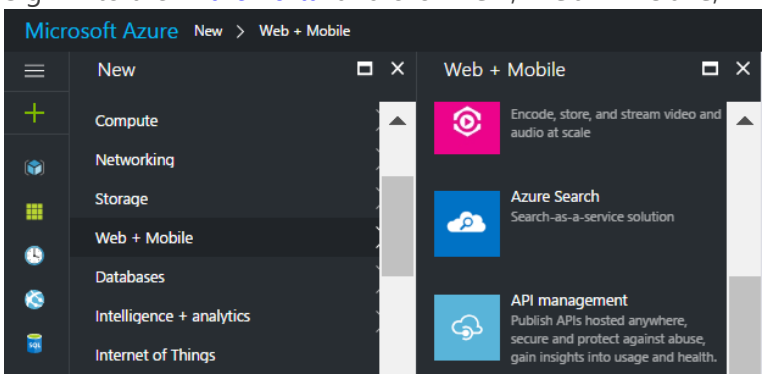
Click 'Sign up'	
Wait for this process to complete	
Complete Click 'Start managing my service'	

Creating an API Management Service Instance

Lab 1 requires the creation of an API Management Service.

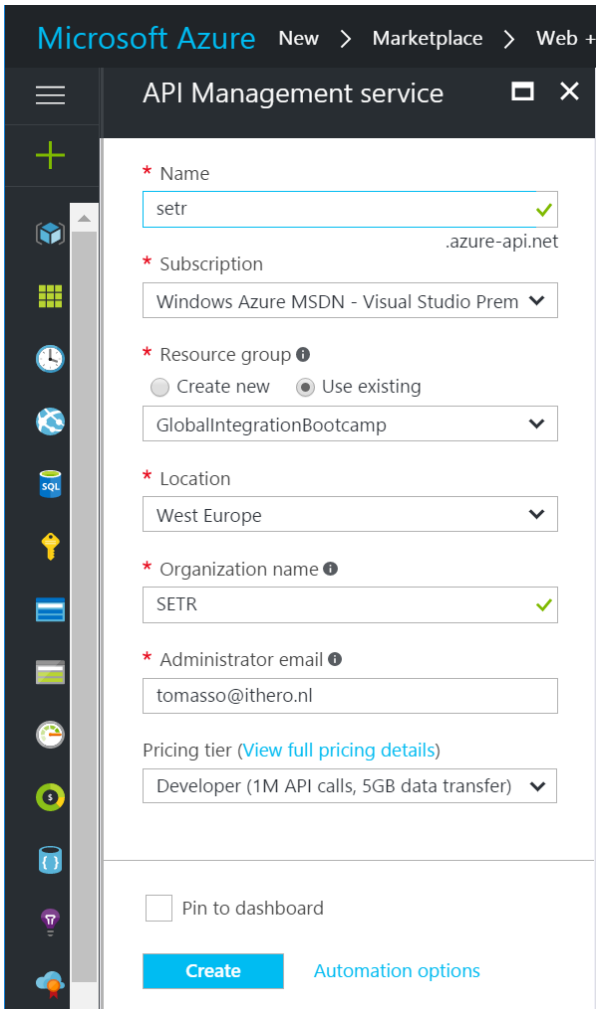
> **NOTE:**
>
> **It can take up to 45 minutes to provision an API Management instance! This is why you should not wait until the day of the event to set this up.**
>
> **That said, the Developer Pricing Tier is A$2.01 per day, so probably best to create the service 1 to 2 days before the event is scheduled to run.**

1. Sign in to the Azure Portal and click **New**, **Web + Mobile**, **API Management**.



2. For Name, specify a unique sub-domain name to use for the service URL. Choose the desired Subscription, Resource group and Location for your service instance.
   Enter SETR for the Organization Name, and enter your email address in the Administrator E-Mail field.

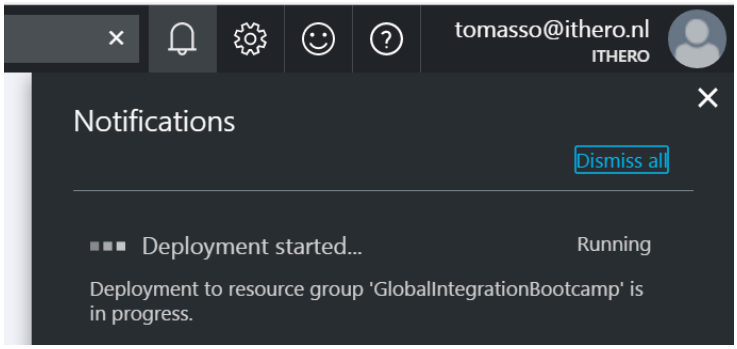
   Note
   The email address is used for notifications from the API Management system.



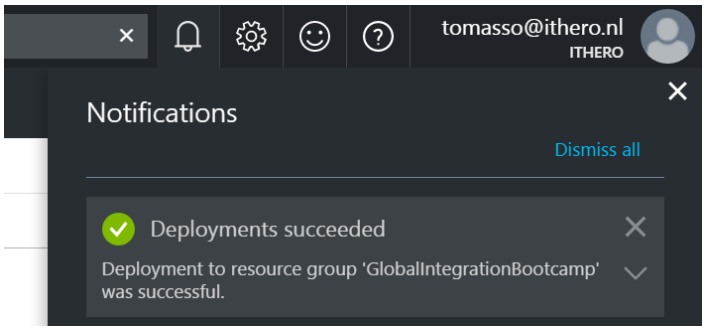
Note

API Management service instances are available in three tiers: Developer, Standard, and Premium. You can complete this lab by using the Developer tier.

3. Click **Create** to start provisioning your service instance.
4. The Deployment takes several minutes. Click in the Header toolbar on Notifications to follow the Deployment



5. Once the service instance is created, click on the notification to open API Management.



Preparing the BizTalk Server Virtual Machine

A BizTalk Server environment is required for Lab 3. Whilst you do not need any BizTalk developer skills (or a development environment) for this lab, you do need to be able to host a BizTalk application.

Setting up even a stand-alone single machine BizTalk environment is not a trivial task; it can take most of a day to build one from scratch.

Therefore, Bill Chesnut has provided a pre-built VM which you can download (~25GB) and import into a locally running Hyper-V instance on a Windows machine. This will save you heaps of time.

Using the Pre-Built BizTalk Server 2016 Virtual Machine

Download the Virtual Machine ZIP files

ZIP files can be downloaded via **azcopy** with the following command (download **azcopy** from here or use the **MicrosoftAzureStorageTools.msi** installer found in the GIB17-SetupFiles.zip archive):

```
AzCopy /Source:https://gibbtsdev16.blob.core.windows.net/bts16dev
/Dest:C:\myfolder
/SourceKey:JLP+t9OEjuLq3jxo+vp1MGGrAcAZugsTNUMAvRke+t3u+pZBFq/Wwvup1V4x951wJ4nwh
PXkWrPz5r8q2uwIzA== /S
```

Replace “myfolder” with the path on your local machine.

The logon account for the machine is **Adminstrator** and all passwords are **BizTalk2016** for the VM.

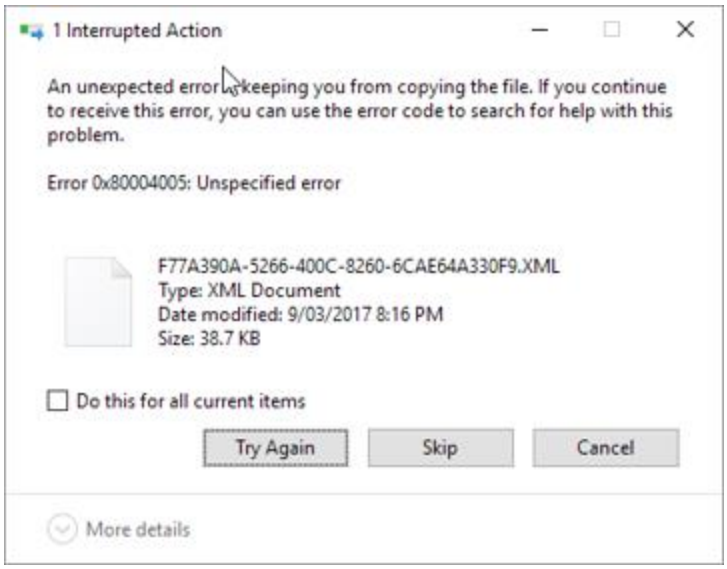
PLEASE ENSURE THAT YOU DO NOT DELETE THE FILES!!

Alternatively, you may access the same files from One Drive (although this seems to be a much slower download):

https://1drv.ms/f/s!Au9XCg12VXSijNpyfTRPWho6L0XJUg

It is reccomended to user the latest version of WinZip to unzip the files.

It is possible that you may get an error at the end of the extraction attempt:



In this case, do the following:

1. Just click **Skip** and it will finish
2. Now go into the ZIP archive and copy the file mentioned in the screenshot above, found here:

   BTS16DEV.zip\BTS16DEVEVAL\Virtual Machines*{GUID}*.xml

3. Paste it into your extracted version in the same location (withing the Virtual Machines folder).
4. Now you can import the VM into Hyper-V.

Importing the VM into Hyper-V

Follow the import instructions found here.

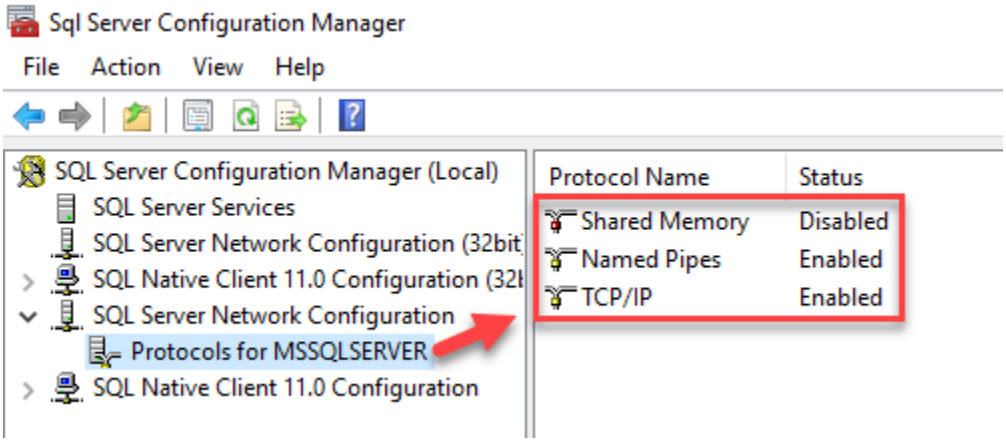
Final Configuration

There are two minor configuration changes that must be made to the VM before it will work in Lab 3.

Configure SQL Server Network Configuration Protocols

1. Press the **Windows** key to open the Start menu, type "SQL Server Configuration Manager" and click in "SQL Server 2016 Configuration Manager" option from the **Search** window.

   a. Or press the **Windows** key to open the **Start** menu, expand **All Apps > Microsoft SQL Server 2016** and select "SQL Server 2016 Configuration Manager"
2. In **SQL Server Configuration Manager** windows, from the left-hand pane expand "SQL Server Network Configuration" option and then click "Protocols for MSSQLSERVER"
3. Verify that both "TCP/IP" and "Named Pipes" are enabled;
   a. If not, right-click in the protocol, and then click "Enable"
   b. Repeat to enable the other protocol if necessary.
4. Verify that "Shared Memory" is disabled.
   a. If not, right-click Shared Memory, and then click "Disable"



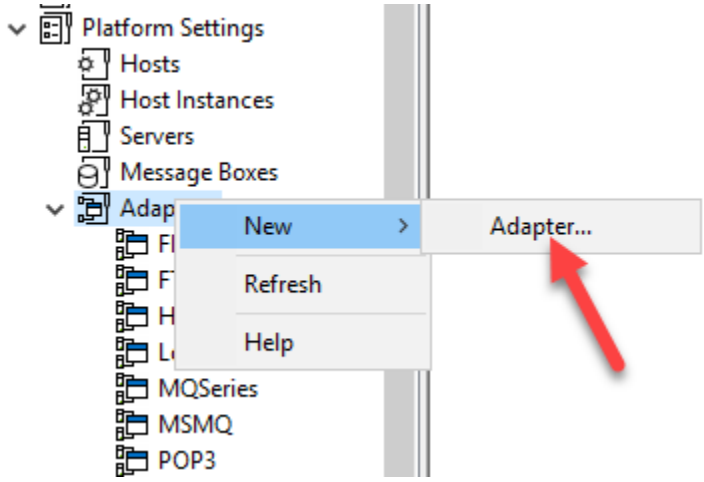
5. In the left-hand pane, click "SQL Server Services", right-click "SQL Server (MSSQLSERVER)", and then click "Restart". Or click "Stop" and when the service has stopped, right-click "SQL Server (MSSQLSERVER)" again, and then click "Start".
6. Close **SQL Server Configuration Manager**.

Enable the WCF-SQL Adapter

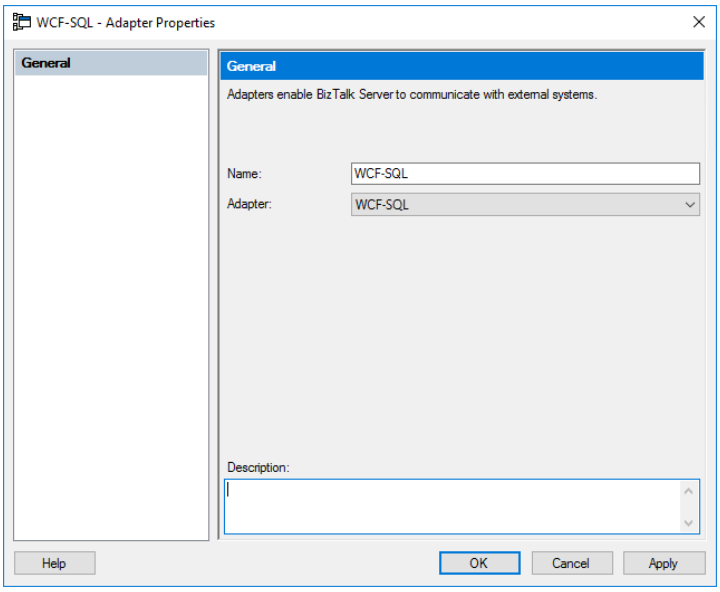
The WCF LOB Adapter Pack has been installed on the pre-configured VM, however the adapter has not been setup in the BizTalk Administration console. This is a very simple exercise:

1. Open **BizTalk Administration Console** by pressing the **Windows** key to switch to the Start menu, type "BizTalk Server Administration" or "BizTalk", click "BizTalk Server Administration" option from the Search window
2. In the console left tree, expand **BizTalk Server Administration –> BizTalk Group –> Platform Settings** and then **Adapters**

3. Right-click on "Adapters" and add a new adapter by selecting the option "New –> Adapter"



4. In the **Adapter Properties** window:
    a. In the **Name** box, type a descriptive name for this adapter (e.g. "WCF-SQL").
    b. In the **Adapter** combo box, select the "WCF-SQL" adapter from the drop-down.



5. Click **OK** to complete the process of adding the adapter.
6. NOTE: This configuration requires that you restart the host instance associated with the adapters (e.g. "BizTalkServerApplication").

Creating Your Own BizTalk Server 2016 Virtual Machine

If you are unable or unwilling to use the provided pre-built Virtual Machine (e.g. you are not running a Windows machine with Hyper-V enabled, etc), then your only alternative that will enable you to complete Lab 3 is to provide your own instance of BizTalk Server 2016. This can be a virtual machine running on your laptop or it can be running in Azure.

There is a new gallery image for BizTalk Server 2016 Developer Edition available in the Azure Portal. However, it is only a vanilla Window Server 2016 image with the installer files for BizTalk Server present on the C: drive. You will need to perform the complete install and configuration (including obtaining and installing a prerequisite copy of SQL Server 2016), which is a lengthy and non-trivial process.

For complete instructions on setting up a stand-alone BizTalk Server 2016 environment, download this excellent whitepaper by Sandro Pereira.

The following items are required for Lab 3:

- SQL Server 2016 (Developer, Standard or Enterprise - *not* Express)
- BizTalk Server 2016 (any edition)
- BizTalk Adapter Pack

For the purposes of this lab exercise, you do not need Visual Studio installed (unless you are using the same machine for the other labs). Just be aware that if you do install Visual Studio:

- It must be installed *before* BizTalk Server;
- If you install the Microsoft Azure Logic Apps Enterprise Integration Tools for Visual Studio 2015 2.0 then you will not be able to create and build BizTalk projects as the two cannot live on the same machine.

Optional Setups

Setup of the following items is built into the labs themselves. However, it is recommended to perform some or all of these steps in advance of the event so that you have the maximum amount of time to complete the remaining steps in the lab.

Lab 2: Installing the On-Premises Data Gateway

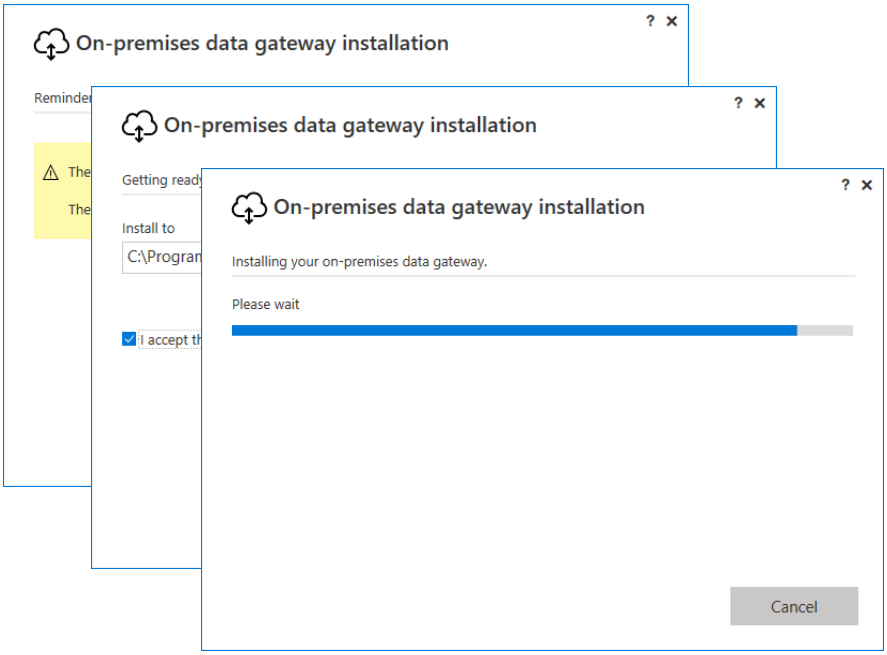
This lab requires that the On-Premises Data Gateway be installed and configured on the BizTalk Server (which should also be the server that will host a custom database).

Install the Gateway On-Premises

1. Download the On-Premises Data Gateway:

   https://www.microsoft.com/en-us/download/details.aspx?id=53127

2. Install the On-Premises Data Gateway



3. Configure the On-Premises Data Gateway

   After successful installation, it should be configured to be used with Azure.

The On-Premises Data Gateway does not work with a Microsoft account. You should use a work or school account.

If you are using a Microsoft account, follow the steps below (step I) to add a user to the active directory and setup the gateway.

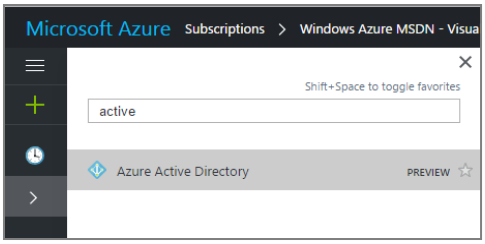
If you are using a work or school account, you can proceed to step II.

I. Adding a user to an active directory (for non-school or work account users)

The On-Premises Data Gateway needs a user to be setup in the Azure active directory to function properly.

a. Login to the Azure Portal by navigating to https://portal.azure.com.

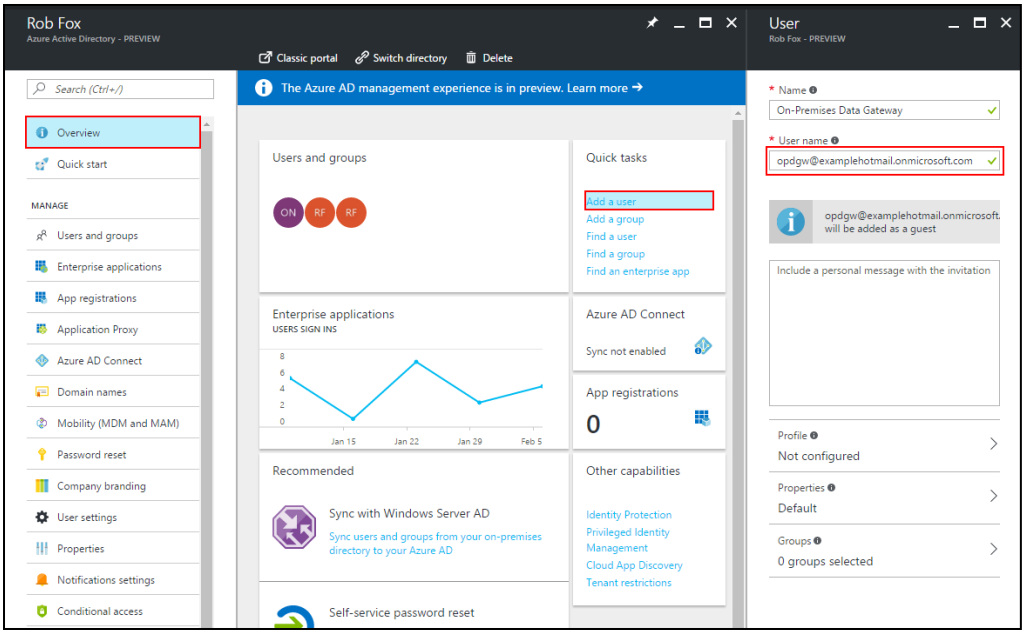
b. Navigate to the Azure Active Directory by searching for it in the resources directory. And open it.



c. In the overview click ***Add a user*** and create a new user for your On-Premises Data Gateway. This user **must have an extension of onmicosoft.com**.

Example

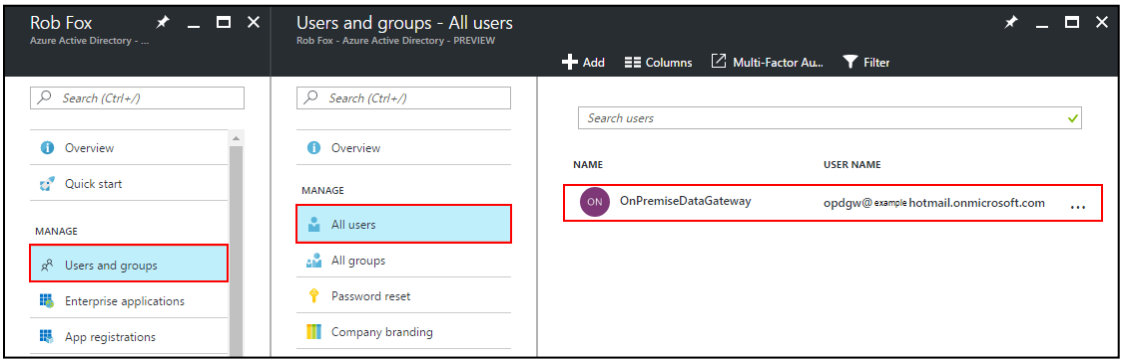
- If your Microsoft account is example@hotmail.com, the OPDGW user should be named <anything>@examplehotmail.onmicrosoft.com.
- If your Microsoft account is anotherexample@outlook.com, the OPDGW user should be name <anything>@anotherexample.onmicrosoft.com.

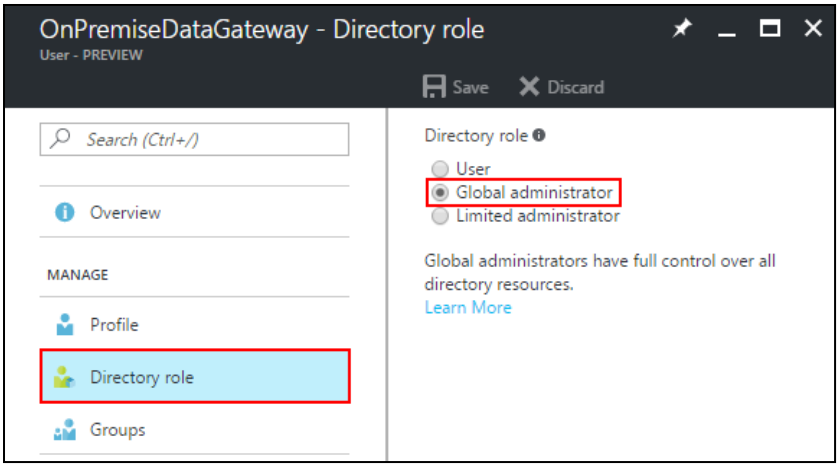


d. Set the appropriate rights for the user.

Select ***Users and groups*** and navigate to the user you just created. Now select the user and navigate to ***Directory Role***.

Make sure the user is set to ***Global Administrator*** and click Save (if applicable).

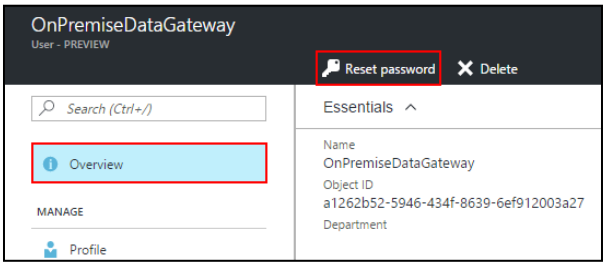




e. Reset the password of the user.

And now that we are here, the user's password should be reset. In ***Overview*** click ***Reset password***. This action will create a temporary password for the user.

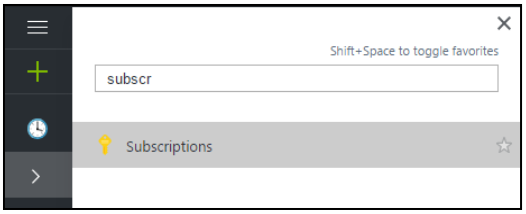
Make sure you copy it or save it for later!

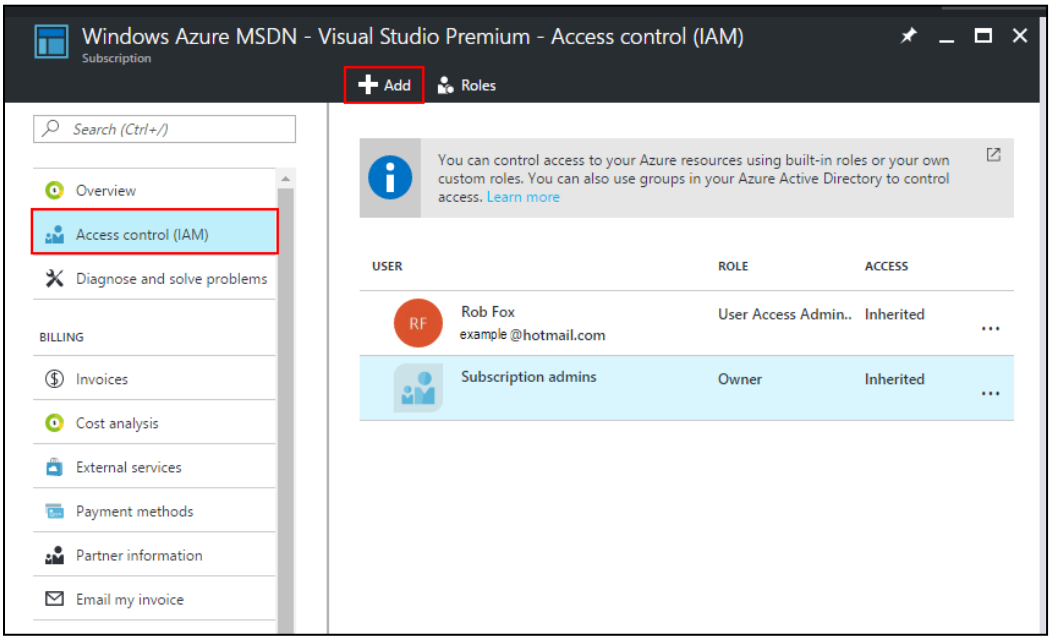


f. Make sure the new user has access to your subscription.

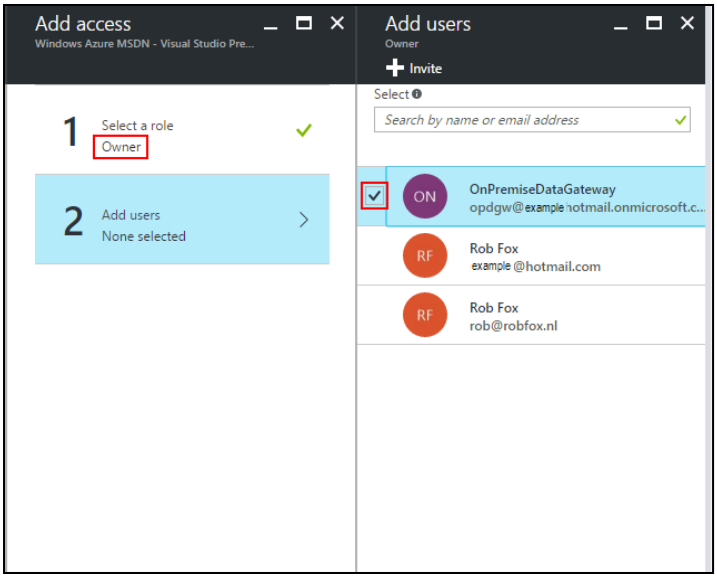
By enabling access, the user doesn't need his own subscription. Everything the user does, is part of your subscription. This way the data gateway can be added under your subscription and not under a new subscription of the user you just created.

First go to your ***Subscriptions*** and select the appropriate subscription. Now under ***Access Control*** select ***Add*** to add users to your subscription.





After clicking Add you'll end up at the following screen. First select ***Owner*** as the **role.** The second step adds the users. Select the user you just created for the On-Premises Data Gateway and apply by clicking ***Select***.



g. Login with the newly created user and reset the password

The password that was given after doing the password reset, was just a temporary password. This password must be reset by the user itself. The user will be asked to update the password on their next login.

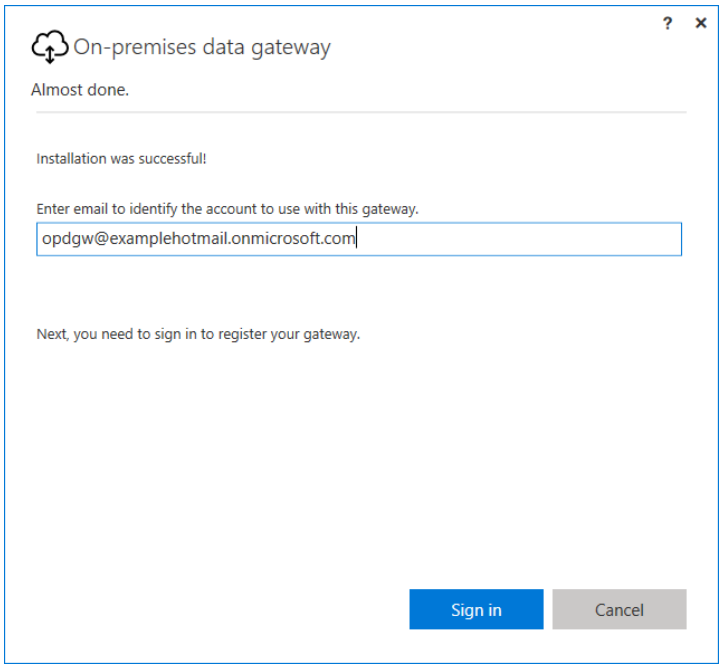
To change the password, open an in-cognito (or in private) browser window, or you can choose to sign out of the portal. After opening the new in-cognito browser window, or after signing out, navigate to https://portal.azure.com and login with the On-Premises Data Gateway user by supplying the e-mail address and the temporary password created during the reset a few steps back and set your own password.

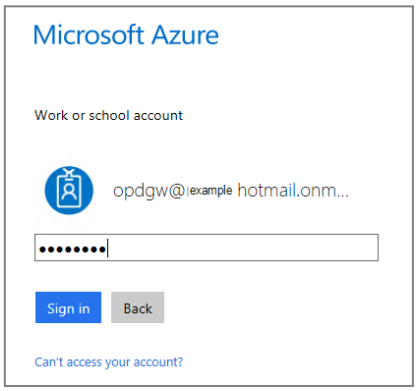
So, now we should finally be set to configure the On-Premises Data Gateway.

II. Configure the gateway

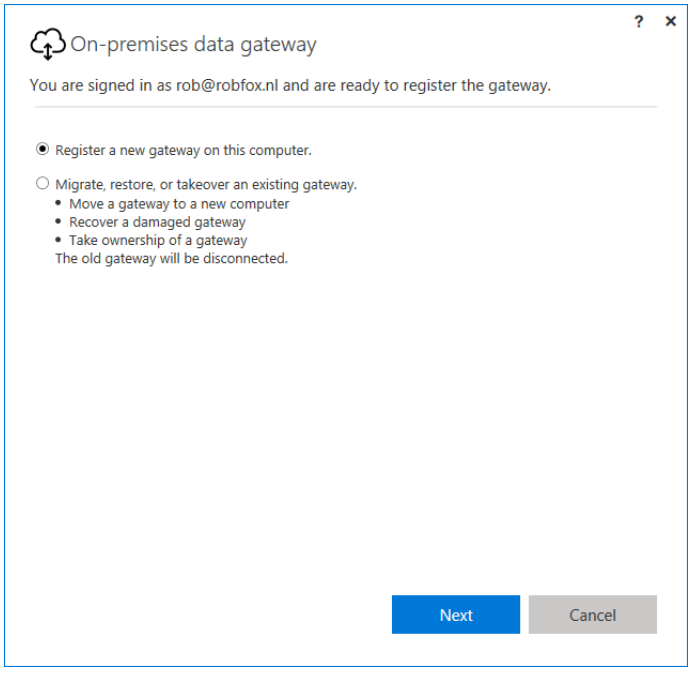
a. On the on-premises machine containing SQL server, start the On-Premises Data Gateway configuration. This should start automatically after installation.

Login with your work or school account.



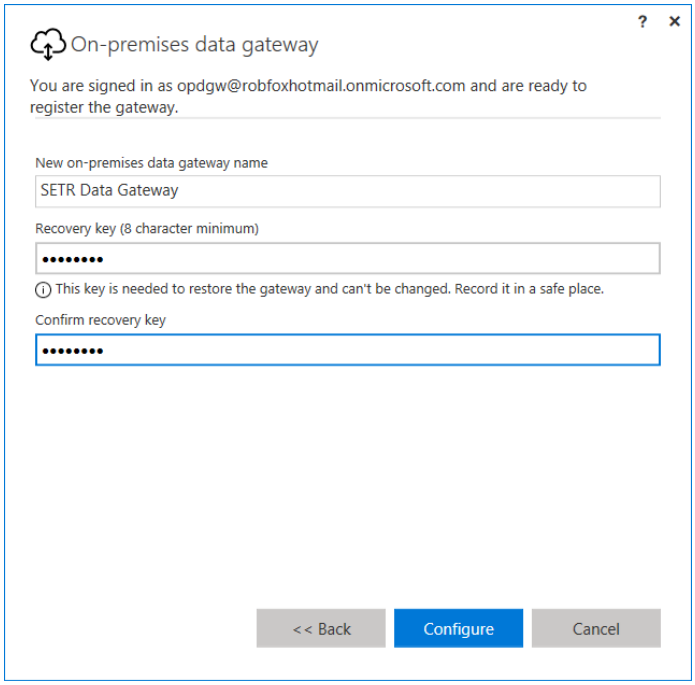


b. On the second screen choose ***Register a new Gateway on this computer***.

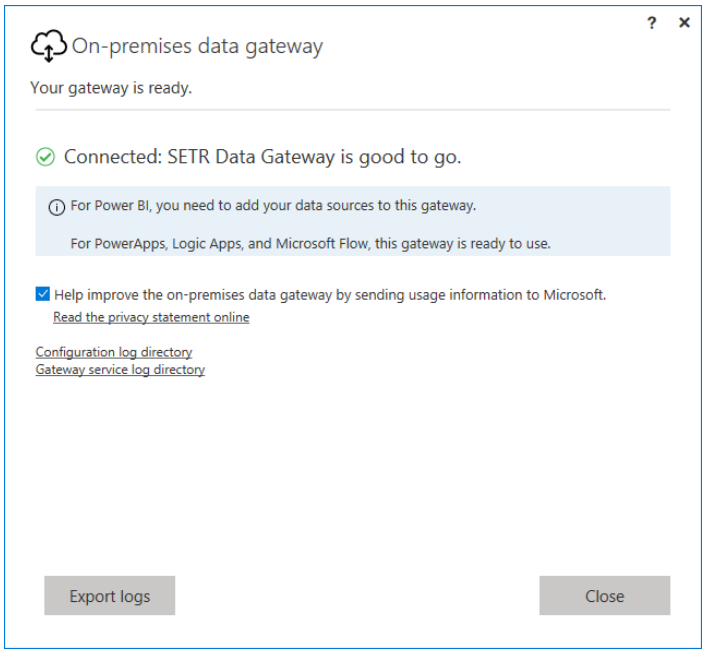


c. In the next screen enter the ***name of your gateway*** and enter a ***recovery key***. These values are totally up to you.

This step may take some time and does not always end successfully. If registration was not successful, repeat this step, until it works.



d. If registration has been successful, you will see the following screen. If registration was not successful, you'll need to repeat **step d** until it works.

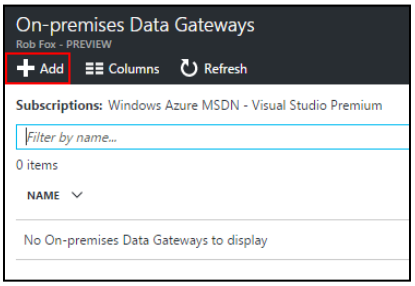


Register the Gateway in Azure

1. Go to the Azure portal and ***sign in*** with the **account you used to register the On-Premises Data Gateway with**. The example in this lab was opdgw@examplehotmail.onmicrosoft.com.

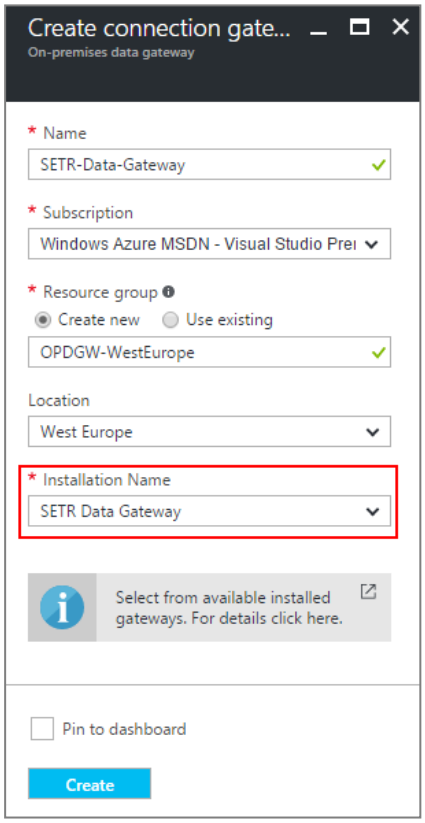
   https://portal.azure.com

2. After logging in, ***navigate to the On-Premises Data Gateway blade*** and click ***Add*** to register the gateway.



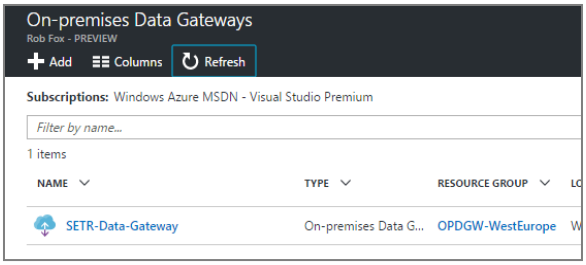
3. Now give your On-Premises Data Gateway a suitable ***name*** to identify it in Azure. Select your ***subscription*** and a ***Resource group*** (you can add a new one if you want). Select the desired ***Location***. I.e. don't select West-Europe if you are in Australia ☺. Please note that the Gateway should be in the same location as the Logic App and the Integration Account which will be created later on.

   Lastly ***select the On-Premises Data Gateway*** you just installed on your local machine. This should be visible in the dropdown box. Click ***Create*** to finish registration.



4. After successful registration, you should be able to see it in your list of gateways, which basically means we're done here.

   For everyone using a Microsoft account, you can logout with the Gateway account and login again with your Microsoft account.



Lab 2: Install the Enterprise Integration Tools

The **Microsoft Azure Logic Apps Enterprise Integration Tools for Visual Studio 2015** are fetured within this lab. You should install this toolset on a server with Visual Studio 2015 where you do not intend to build BizTalk application projects!

The Enterprise Integration Tools SDK is incompatible with the BizTalk Server project templates; installing the former will render the latter unusable. If this is an issue, there are two alternatives:

- Install the Enterprise Integration Tools on another machine with Visual Studio 2015 installed (this machine does not require any of the other resources or programs to be present as it will be used to create an export a schema only)
- Do not install the Enterprise Integration Tools at all, but instead use a BizTalk Server project to create the schema (final result is the same, but you lose the benefit of gaining familiarity with the Enterprise Integration Tools)

To download the Enterprise Integration tools, go here:

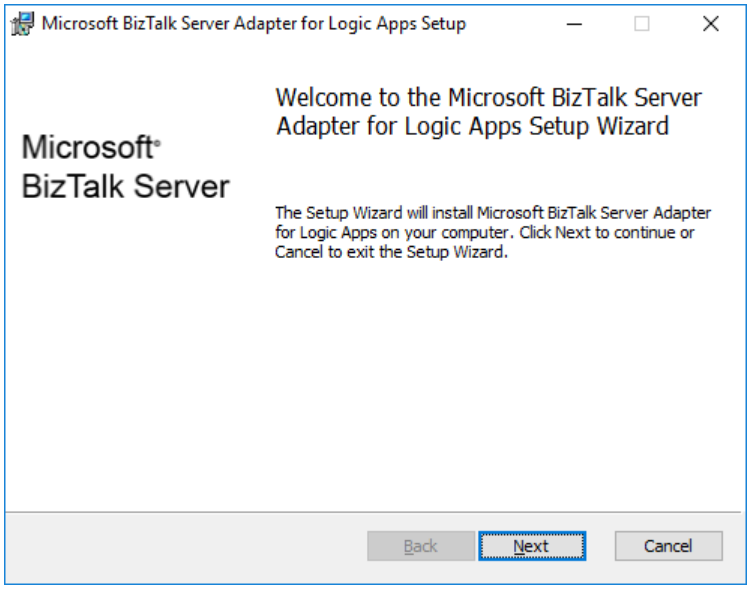
- https://www.microsoft.com/en-us/download/details.aspx?id=53016

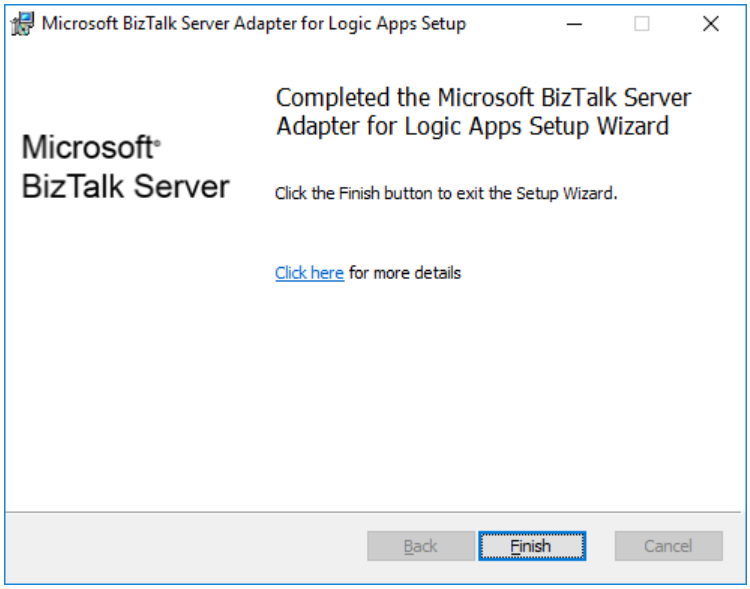
You will need to restart Visual Studio after performing the installation.

Lab 3: Installing the Logic Apps Adapter

Install the Logic Apps Adapter

This lab requires the new Logic Apps adapter. Download the adapter and start the installation. Follow the wizard, and install with default settings.



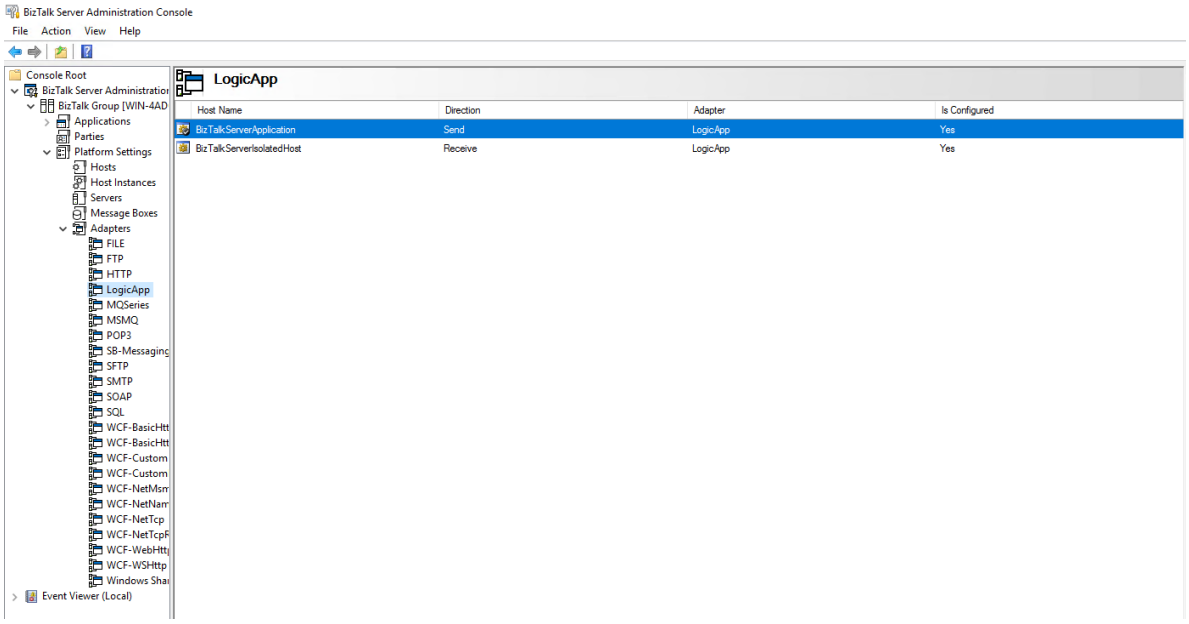


Once installed, the following will have been done automatically.

- The LogicApp adapter has been added to BizTalk
- The send handler has been created, and uses the default host (most probably BizTalkServerApplication)

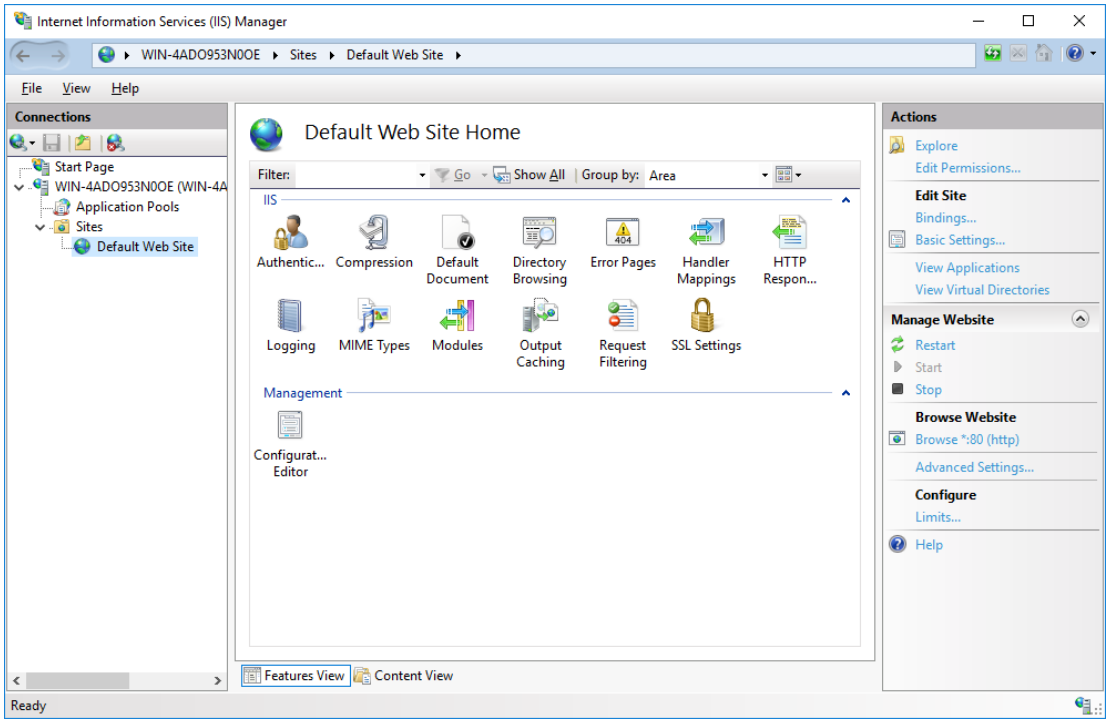
- The receive handler has been created as a WCF service, and uses the BizTalkServerIsolatedHost host
- The Program Files (x86)\Microsoft BizTalk Server 2016\LogicApp Adapter folder has been created, and includes two services: **Management** and **ReceiveService**

The Management service is used by the BizTalk Connector in a logic app to connect to BizTalk Server using the data gateway, to retrieve the ports and message types exposed by BizTalk. The ReceiveService is used by the BizTalk Connector in a logic app when you enter the receive location. This service is responsible for receiving the messages from the logic app in BizTalk. Open the BizTalk Administration Console to check if the LogicApp adapter is configured correctly.

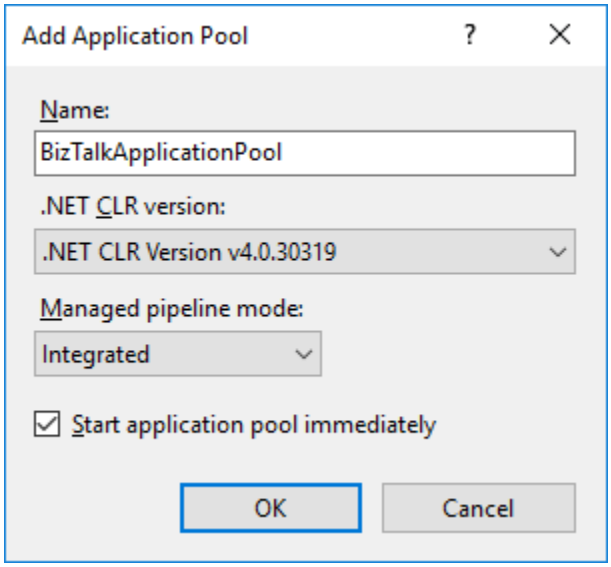


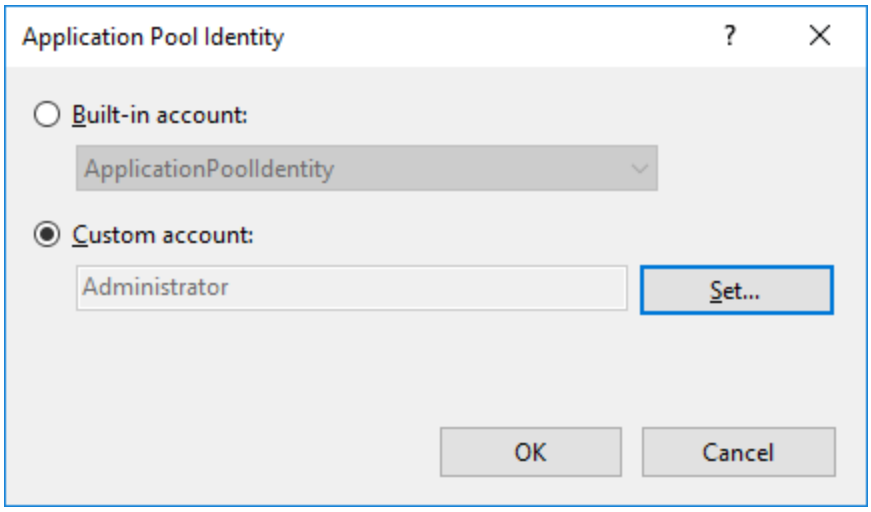
Create IIS applications for Logic App adapter

Now that the Logic App adapter has been installed, we need to configure the IIS applications which were installed. For both these applications, we need to create a WCF application in IIS. Start by opening the Internet Information Services Manager.



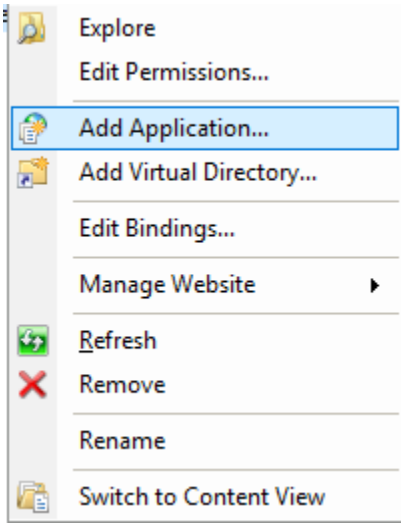
Create an application pool with same rights as BizTalk service users, which will host the IIS applications.

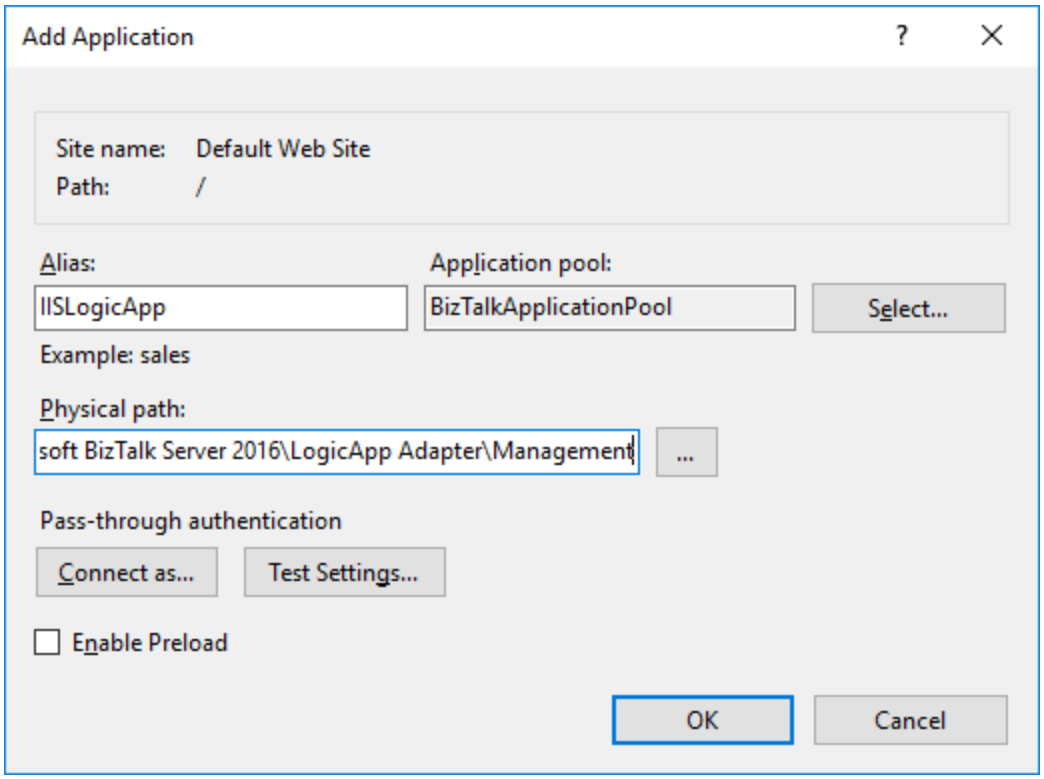




Add Management application

Now back in the IIS Administration console, add an application, and set the physical path set to C:\Program Files (x86)\Microsoft BizTalk Server 2016\LogicApp Adapter\Management. Make sure to select the application pool we created earlier.

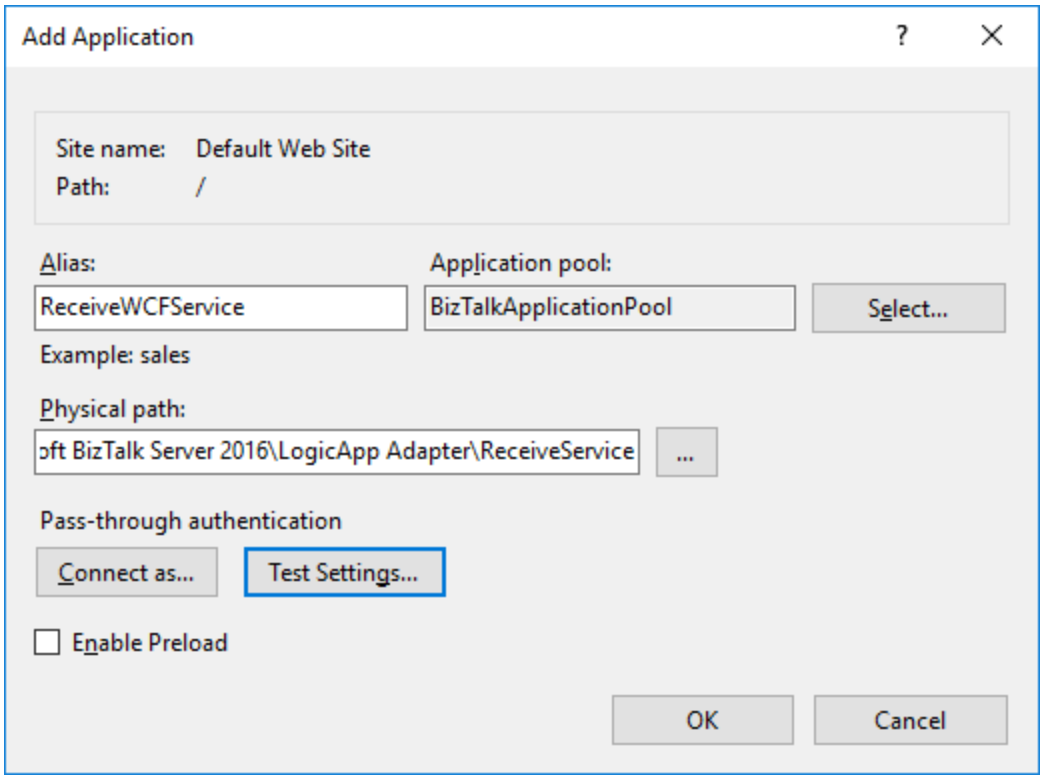




Test application by going to http://localhost/IISLogicApp/Schemas?api-version=2016-10-26, which should download a JSON file with the ports and message types from BizTalk.

Add BizTalk ReceiveService application

Once again in the IIS Administration console, create another application, this time with the physical path set to C:\Program Files (x86)\Microsoft BizTalk Server 2016\LogicApp Adapter\ReceiveService. Here we also need to make sure to select the application pool we created earlier.

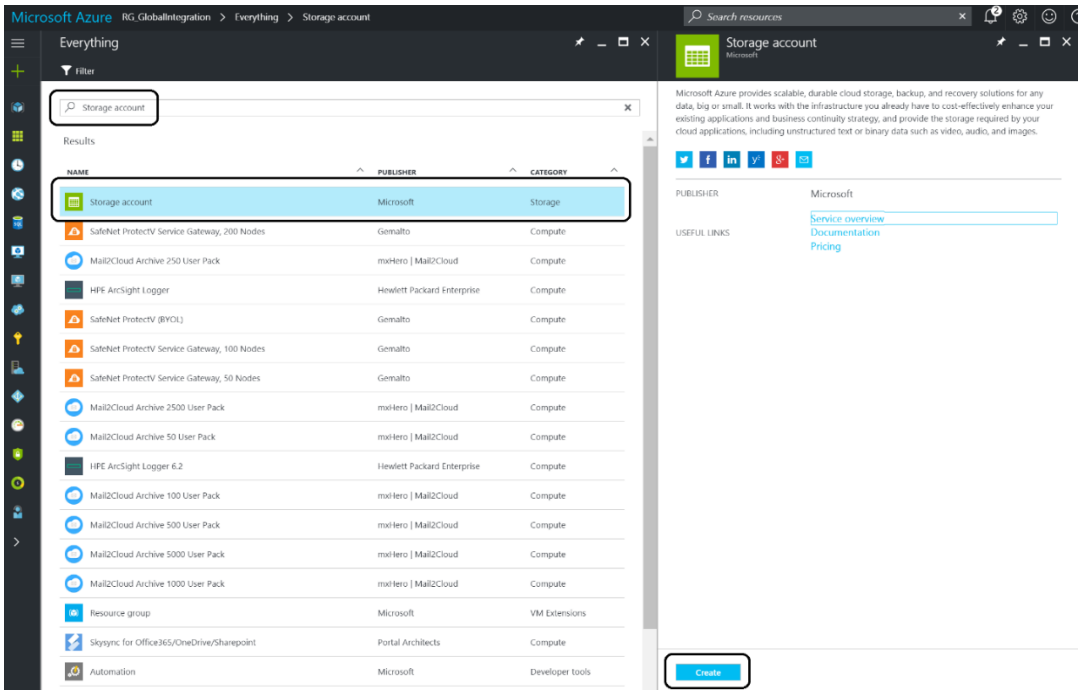


Lab 4: Creating a Storage Account

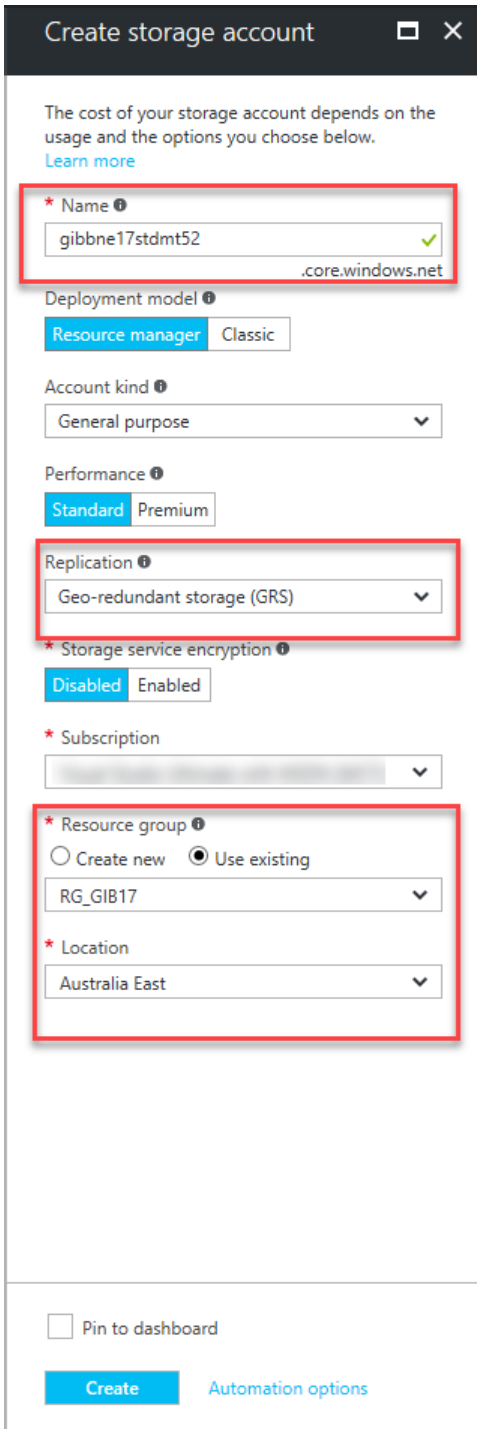
Create Storage Account

The first step in building the solution in this lab is to provision a storage account in Azure. We will be needing storage for setting up our reference table (Table Storage) and storing the order request message in Blob Storage.

1. Go to the Azure Portal: https://portal.azure.com/
2. Login into the Azure portal with your account.
3. In the Market Place enter storage account and select it from the list as shown below.



4. Click **Create**.

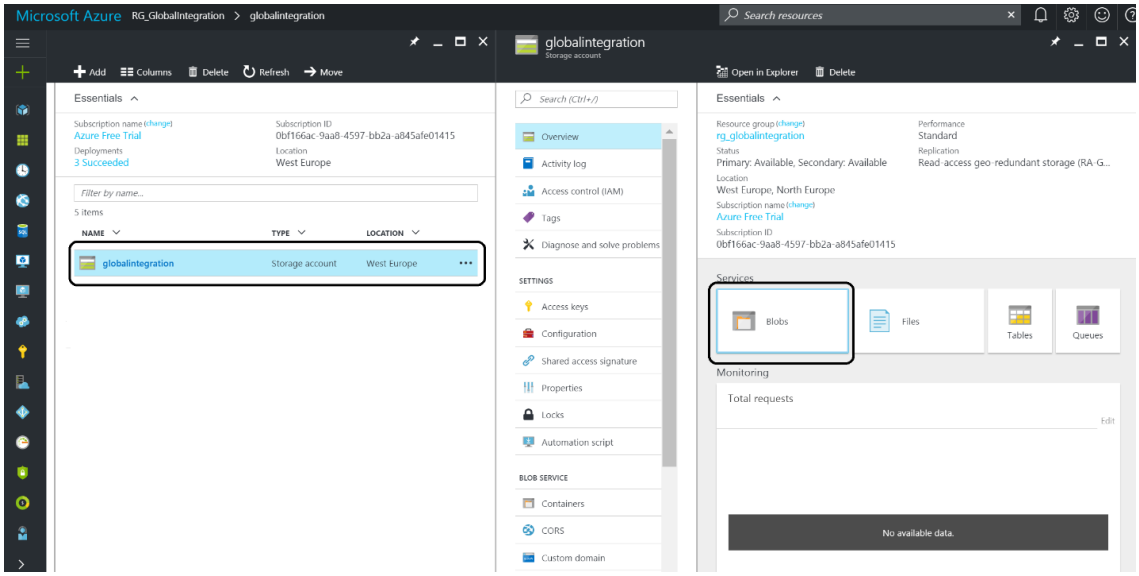


5. Specify a **name**, a **Resource Group** (you can create a new one here if you haven't created a resource group yet) and a **location**. Also you may want to choose GRS replication (cheaper than the default). Subsequently, click on **Create**.

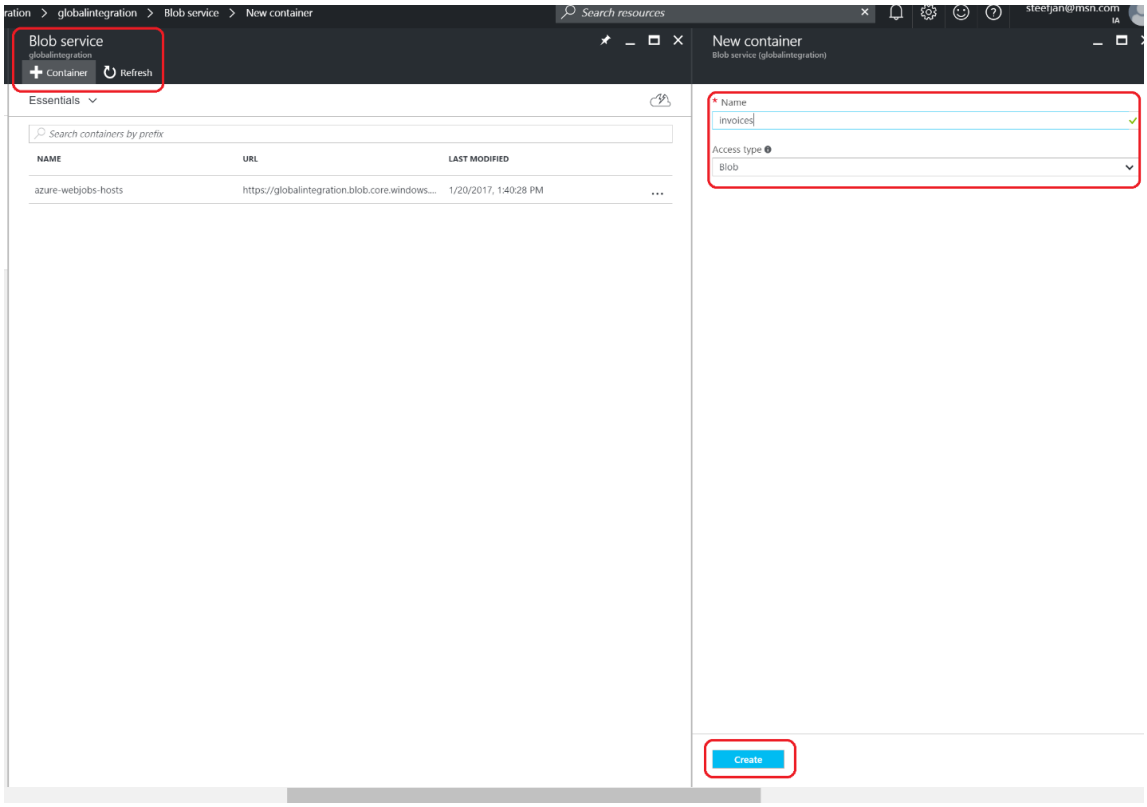
Create Storage Container

Once the storage account has been provisioned you can navigate to it and click on it.

1. In the storage account click on **Blobs**.



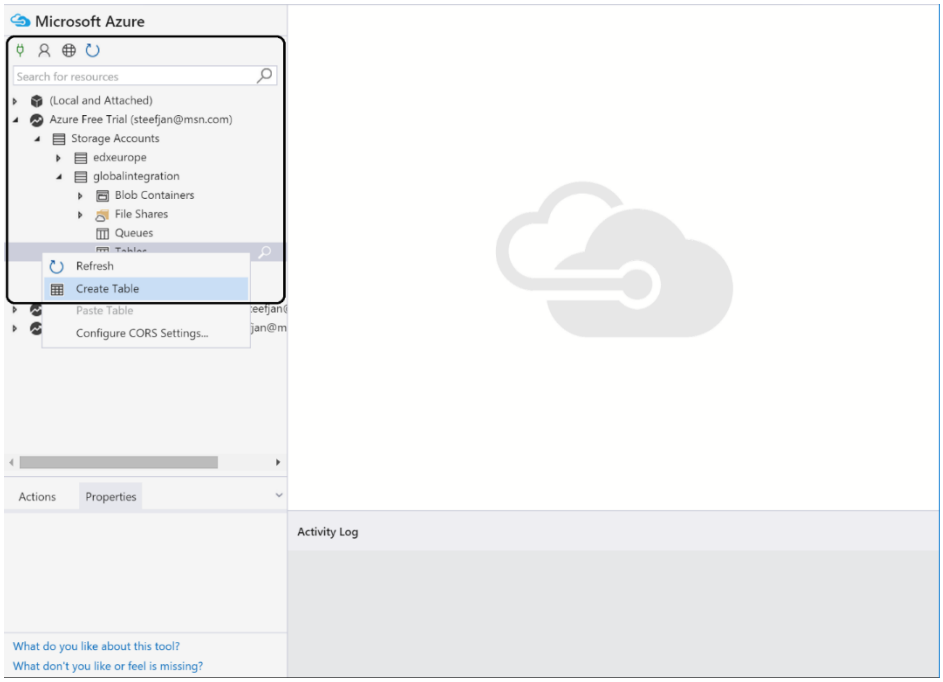
2. Click on **+ Container** and specify the name ("invoices") and Access Type: **Blob**.



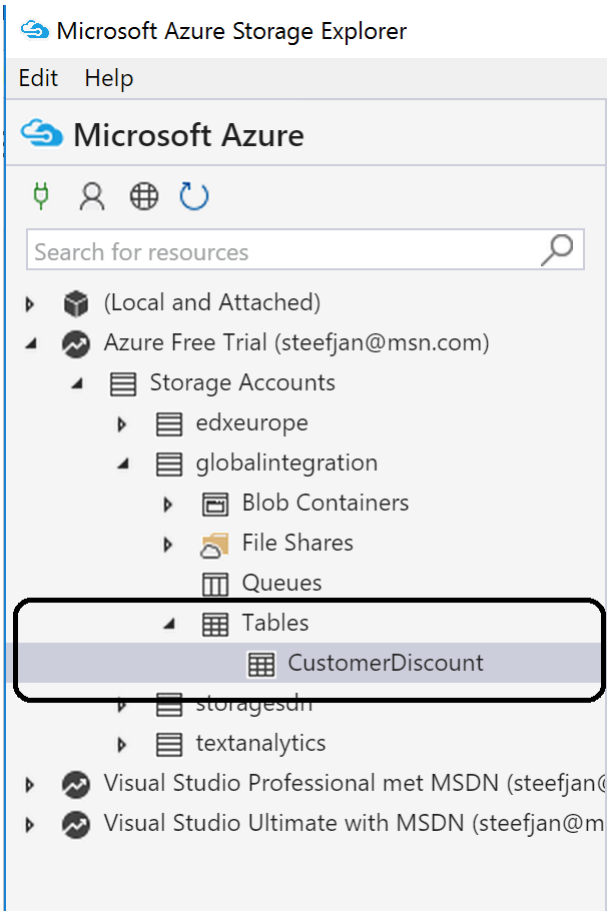
Create Storage Table

To create a storage table will use the **Azure Storage Explorer**, which can be downloaded from http://storageexplorer.com/.

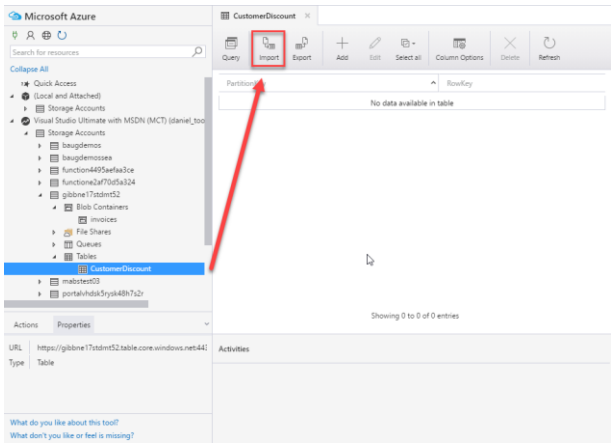
1. Install the tool, and login into your subscription.
2. Navigate to your storage account.
3. Select Tables
4. Right click Tables and click Create Table.



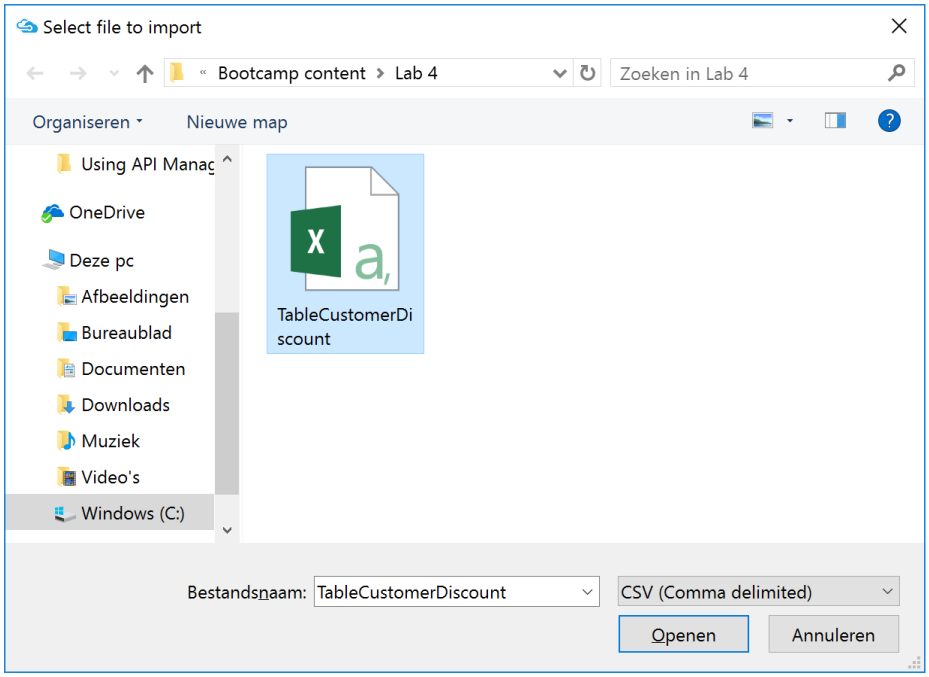
5. Specify a name for the table (“CustomerDiscount”).



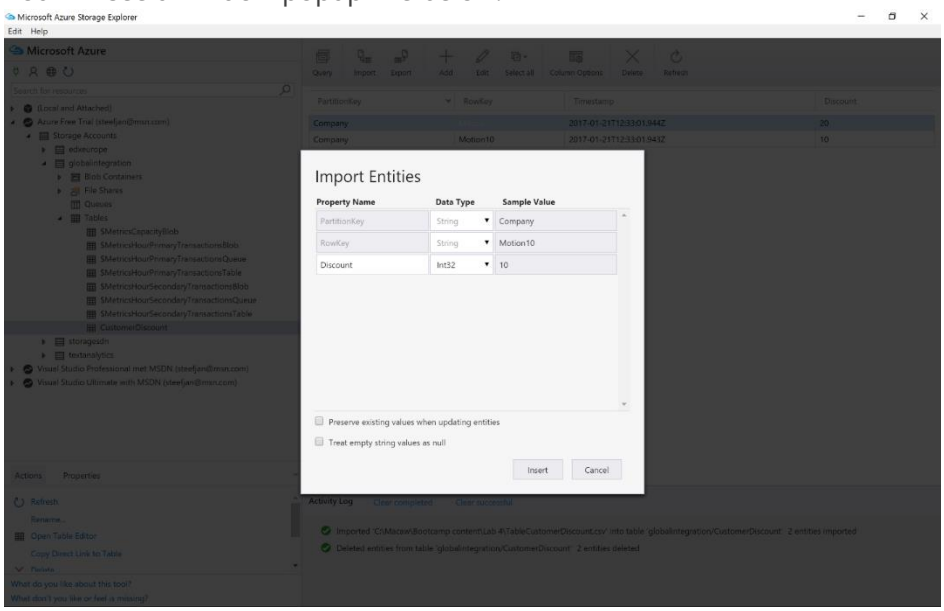
6. Select the **table**.
7. Click on **Import Entities** from file.



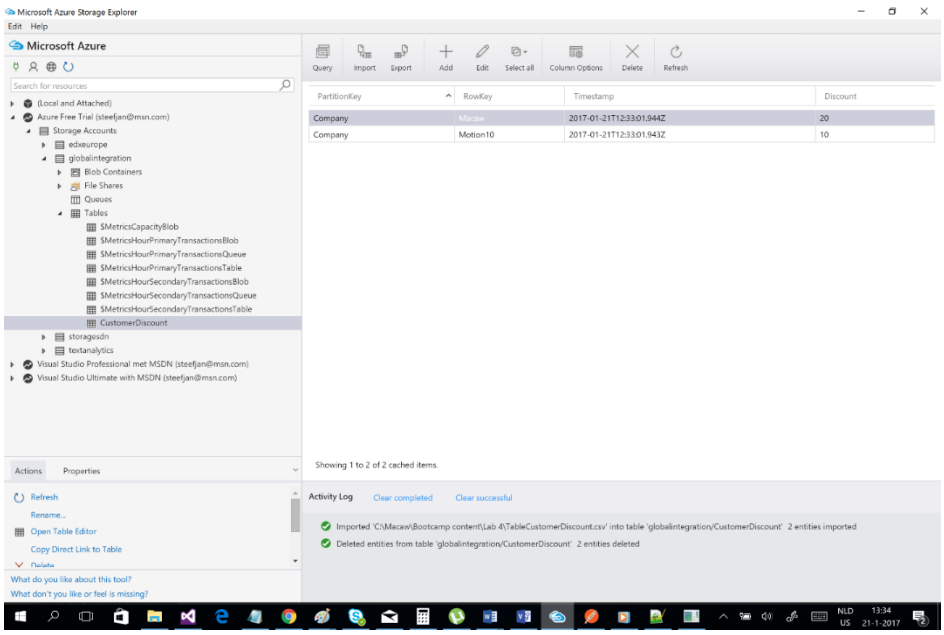
8. Select the **TableCustomerDiscount.csv** you can download from here.



9. You will see a window popup like below.

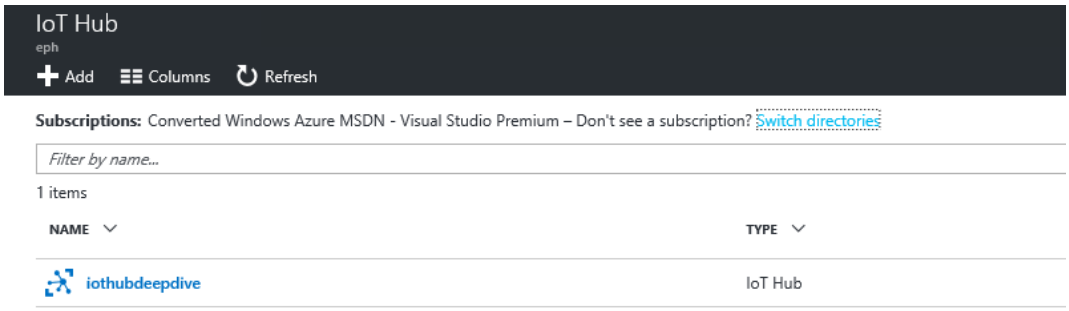


10. Click on **Insert**.
11. **Table** will be loaded with data.



Lab 5: Creating an IoT Hub

IoT Hub can be used for bi-directional communication between Azure and billions of devices. We will use IoT Hub here to send messages from our (simulated) device into Azure. Go to the IoT Hub blade in the Azure portal, and create a new IoT Hub.



You can create one free IoT Hub in your subscription which is great for testing, or you can use one of the paid priceplans if you want to be able to handle more events.