# Introduction

## Why Azure Data Factory

Azure Data Factory is a service available on Microsoft Azure platform. This service allows you to do the following in the clouds:

* Manage data transfers from On-Premise and Azure cloud storage spaces.
  + Transfer data from a variety of sources (SQL Server, Oracle, file system, Azure blobs, etc.)
  + Monitor data transfers and easily recover from failure
* Coordinate Azure HDInsight computing resources like Pig and Hive.
* Simplify cloud based data ingestion like DocumentDB, Stream Analytics, Machine Learning, etc.
* Can manipulate large amount of data compared to SQL Server in Azure /in a Windows Azure virtual machine or APS

## Comparison with SSIS

### Two different paradigms

SSIS is an ETL (extract, transform and load) tool that comes with SQL Server Standard edition and higher. Its primary mission is to extract/transform and load on premise data. Since June 2015, an Azure feature pack is available for SQL Server 2012 and 2014. This feature pack allows SSIS to download and upload files from and to with Azure storage. It also allows us to create and drop HDInsight clusters as well as calling Pig or Hive script from there clusters.

But, as good as the tool is, SSIS remains a service that has to be installed on an on premise Windows server. And you have to purchase SQL Server. When a new version of SQL Server is released, you’ll mostly have to pay for an upgrade unless you have software insurance with Microsoft.

Azure Data Services is a service that is available with your Microsoft Azure subscription. Being a service means that you only pay for the time you use it. As opposed to on premise software, you don’t have to worry about updates. They’re applied automatically in Azure. Let’s say for example that they add a new feature or improve performance on specific modules, you don’t have to do anything. The new features will be available as soon as Microsoft make them available in Azure.

### ELT versus ETL

Azure data factory is also considered as an ELT (extract, load and then, transform) tool. This means that with ADF, since we have massive amount of data, it’s better to bring it into a cheaper storage store that is, Azure storage. Data can then be consumed from there by HDInsight computing resources and transformed to answer business prerogatives. For example, we can have a

Another benefit of using ELT is the fact that it separate transformations dependency from transformations. Azure Data Factory use pipeline schedules to execute its various processes. This way, data can be loaded at upon different schedules. For example, data can be loaded form an on premise data source every hours but can be aggregated every 3 hours in Hive.

Azure data factory is also more cloud aware than SSIS. As you’ll see in this book, it’s built to manage and orchestrate big data transfers. It doesn’t have all capabilities that SSIS has because it is intended to interact primarily in Azure and with Azure services. SSIS on the other hand has almost everything we need to interact with on premise data sources and services.

In conclusion, SSIS and Azure Data Factory share common grounds but are aimed at two different platforms:

* On premise ETL for SSIS and some Azure components interactions with the SSIS Azure Feature pack.
* In the clouds ELT for Azure Data Factory with on premise interaction using a data management gateway\*.

\* We’ll talk about the data management gateway in chapter 4 – Pipeline activities.

# Components

# Getting ready

## Azure account setup

Since Azure Data Factory relies on Azure cloud, you’ll need am Azure account to create, manage and interact with your factory (ies).

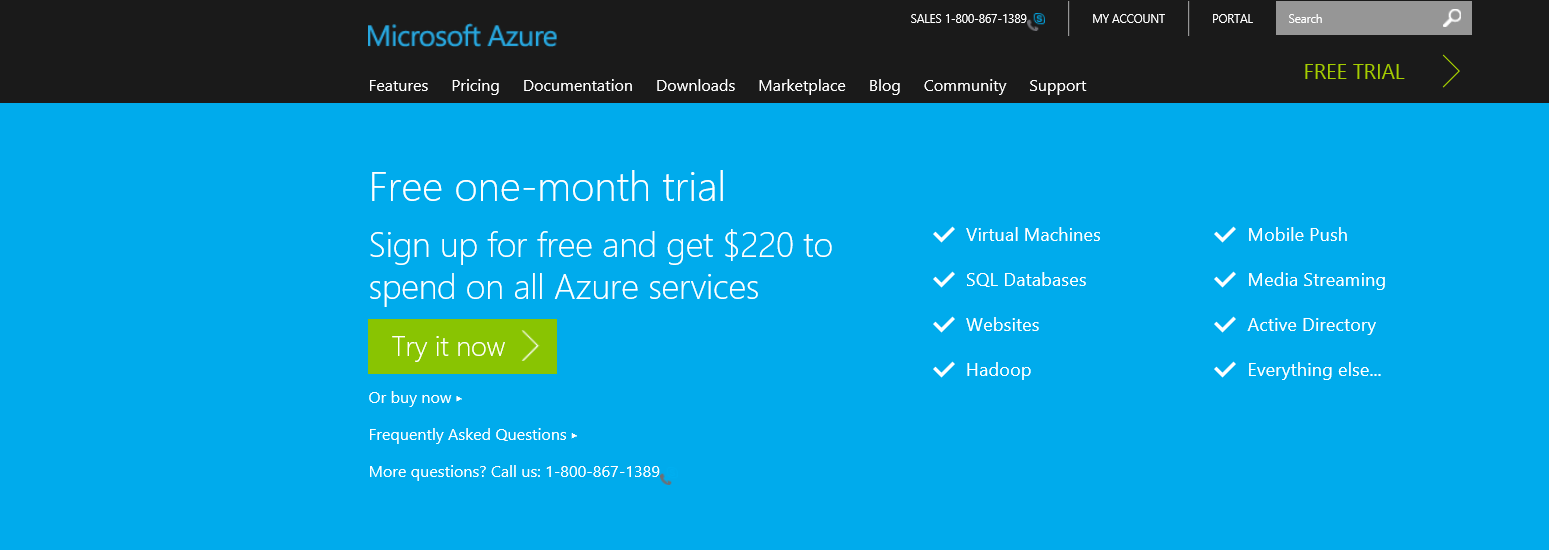
To do so you first go to the [Microsoft Azure web site](http://azure.microsoft.com/en-us/pricing/free-trial/?WT.mc_id=azurebg_US_sem_bing_BR_BRTop_Nontest_FreeTrial_azure&WT.srch=1) and sign in with your Microsoft account that is linked to Microsoft Azure. If you don’t have a Microsoft account, you have the option to get a one-month trial for free.

Figure 3‑1

If you select “Try it now”, you are redirected to the following scree. You now have two options: Sign-in

with your Microsoft account or create a new one by clicking “Sign up now” in the screen below.

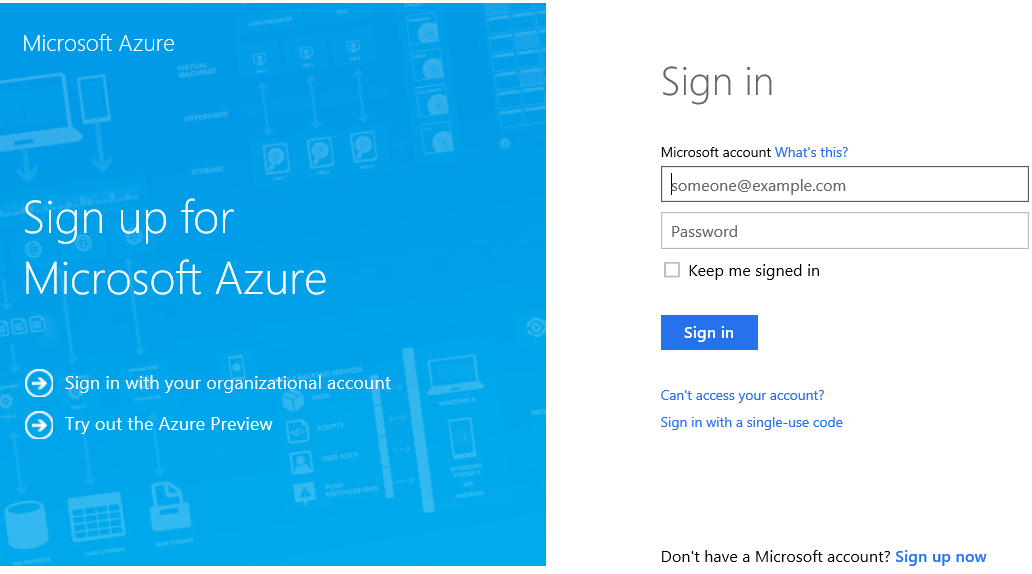


Figure 3‑2

Once you have setup your account, go the new [Azure portal](https://portal.azure.com/) and enter your credentials to log in. You should see a screen similar to the following one:

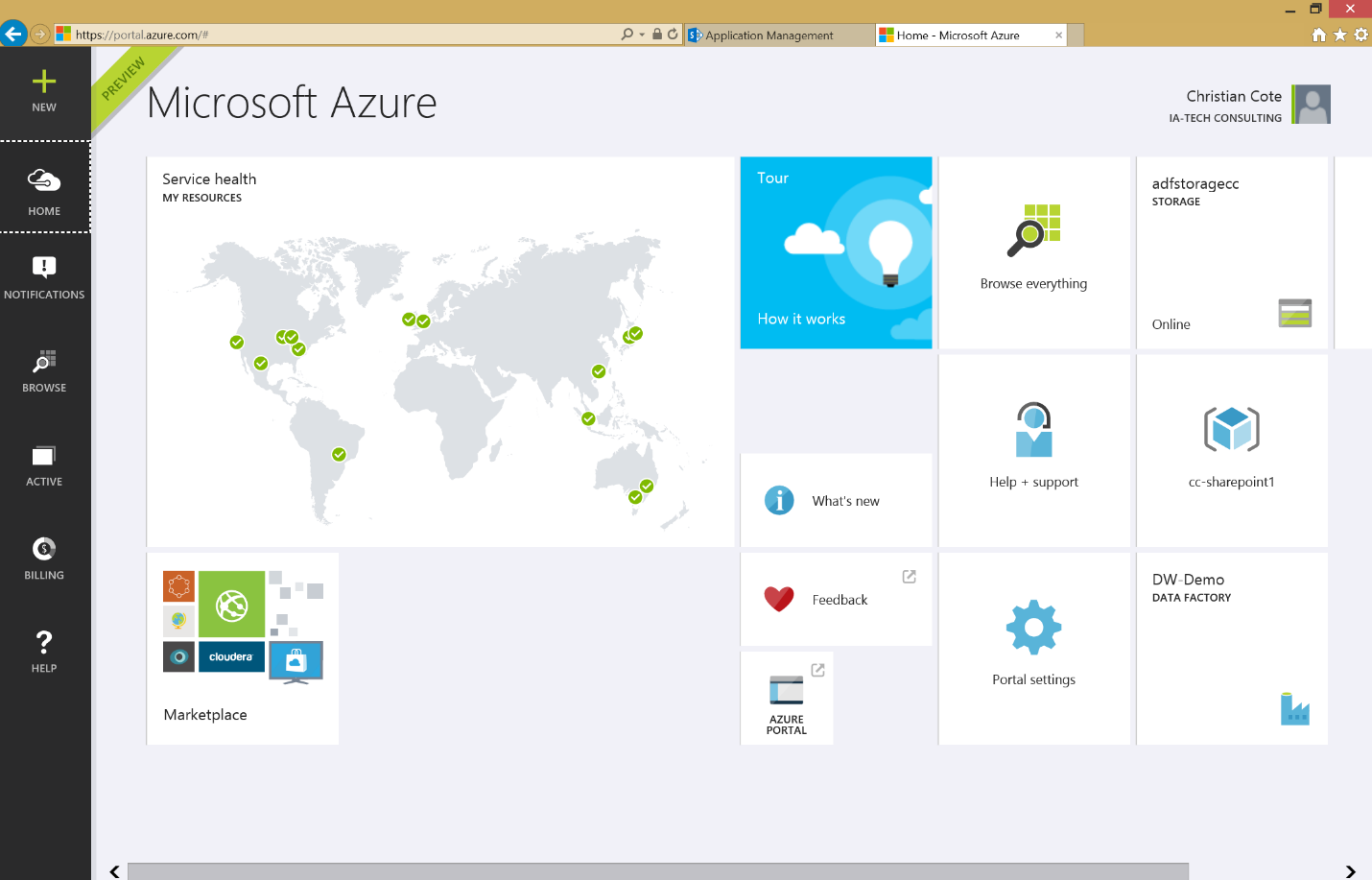




Figure 3‑3

In this book, we’ve changed the default theme to use the dark one. To do this, click on the portal settings (gear). The following screen appears:

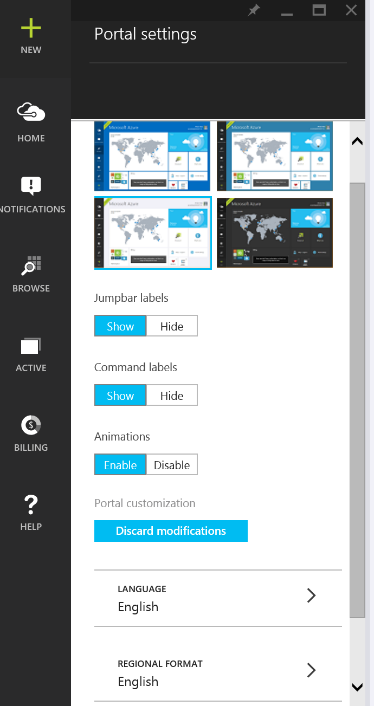


Figure 3‑4

This screen allows you to customize the portal as you like. But the only thing we want here is to use the dark theme. So we click on the dark theme to select it (Figure 3‑5)



Figure 3‑5

The background instantaneously turn black. Now, click on the X at the top of the Portal setting blade to close it as shown in Figure 3‑6 below.

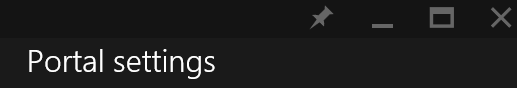


Figure 3‑6

## Creating a storage account

The next step is to configure the storage account. Almost every services in Azure are using storage. The counterpart of Azure storage on premise would be the hard drives or thumb drives on computers or servers. They contains files that are used by us or various services in Azure. Almost everything we do on Azure generate files. For example, when we execute a map reduce job on an HDInsight cluster, it generates a lot of log files that is the result of the job execution as well of the execution by itself.

### Create a storage from the portal

From the portal main page, click on the + sign, select “Data + Storage” and “Storage” at the right.

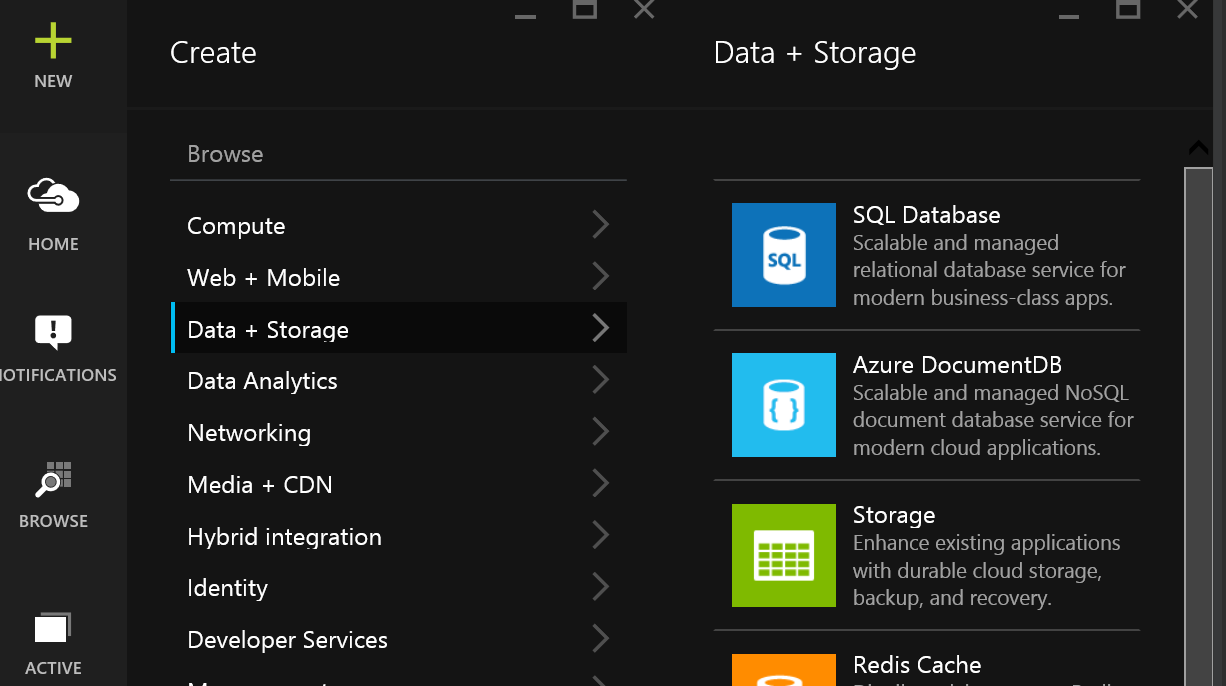


Figure 3‑7

The storage account blade appears.

#### Storage name

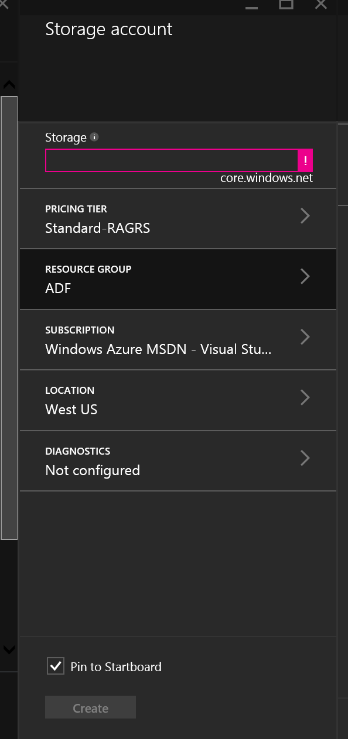


Figure 3‑8

The name of the storage has to be in lowercase, can only contains letters and digits and has to be unique. The reason for the uniqueness is that the storage has an address accessible via internet. For example, if I select MyAdfStorage, it will be known as being myadfstorage.core.windows.net and that’s how many tools or service will refer to it.

As you can see on Figure 3‑8 the name is validated as we enter it. In that case, the text box is magenta because we didn’t provided any name for the storage. Also if you hover you mouse over the exclamation mark at the right of the storage name text box (Figure 3‑9), you’ll get the validation error message that tells you what’s wrong with the name of your storage.



Figure 3‑9

Once the chosen name is valid, you’ll get a check green mark that tells you that the name is ok.

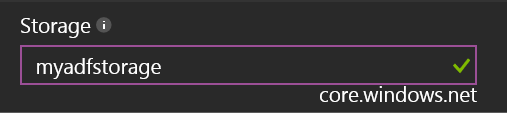


Figure 3‑10

#### Pricing tier

The pricing tier blade allows you to select the kind of storage you want to use. For the purpose of this book, we’ll use the basic (cheapest) one. But depending on your service level agreement (SLA) or requirements with your client or users, you might want to use a more expensive and performant storage.

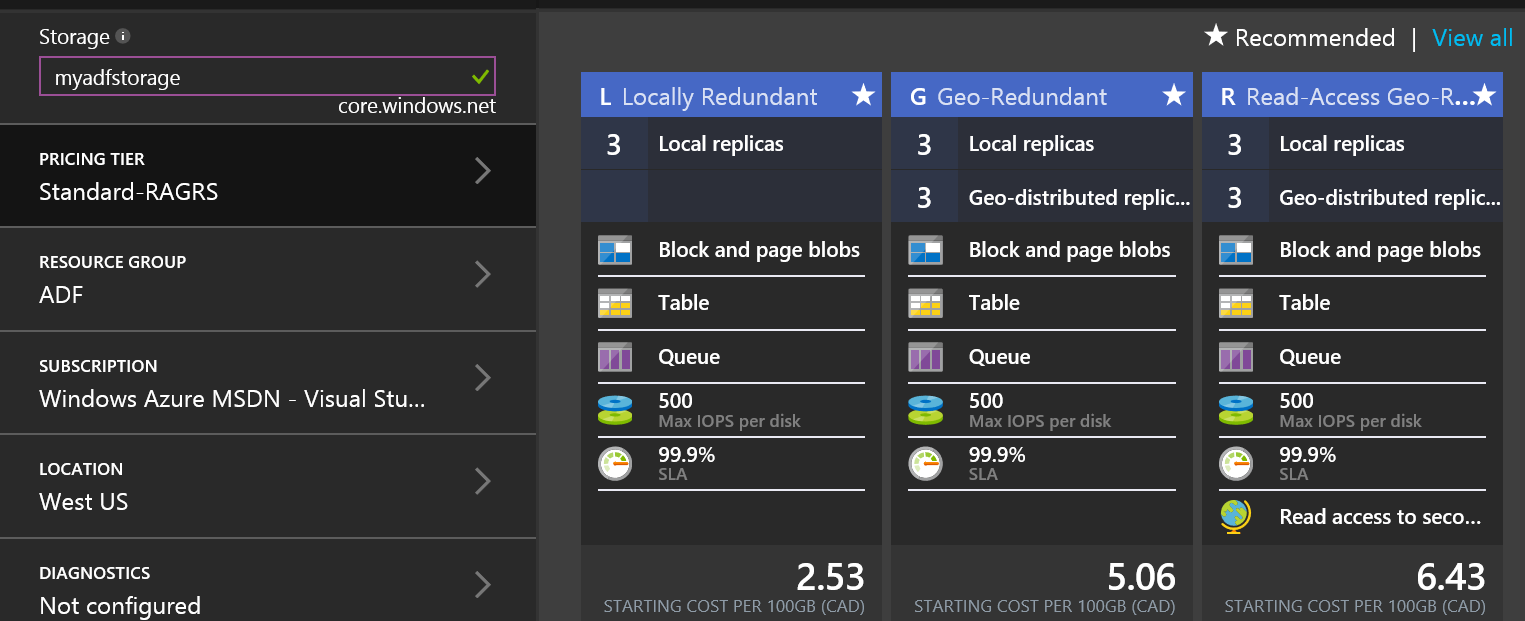


Figure 3‑11

If you click on “View all” as shown at the top right of Figure 3‑11, you’ll get more options in term of storage.

By default, all storage account types keep 3 copies of the files that you create or upload. This would allow file recovery if one file is damaged on a server for some reason. Some storage account options offer the possibility to geographically replicate and distribute your files at different data centers around the world. This gives more security and access to secondary storage via geographically distributed location.

With other storage options, you can have faster disk operations but at a higher cost. Prices varies upon time and are for each 100GB of monthly storage at time of writing.

#### Resource group

A resource group allows you to centralize resources into a common group in Azure. Think about them as folders in Windows/ A folder can have different types of files and / or folders. For example, let’s pretend you create a resource group named MyResourceGroup and you use it for factories, storage, HDInsight clusters and machines. When you decide to delete the resource group, all contained Azure resources contained into it will be deleted too.

You can access your resource groups in the Azure portal by clicking on “Browse” as shown on Figure 3‑12.

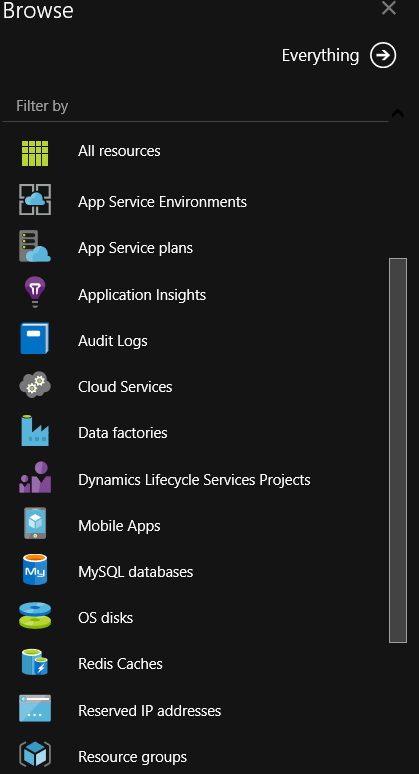


Figure 3‑12

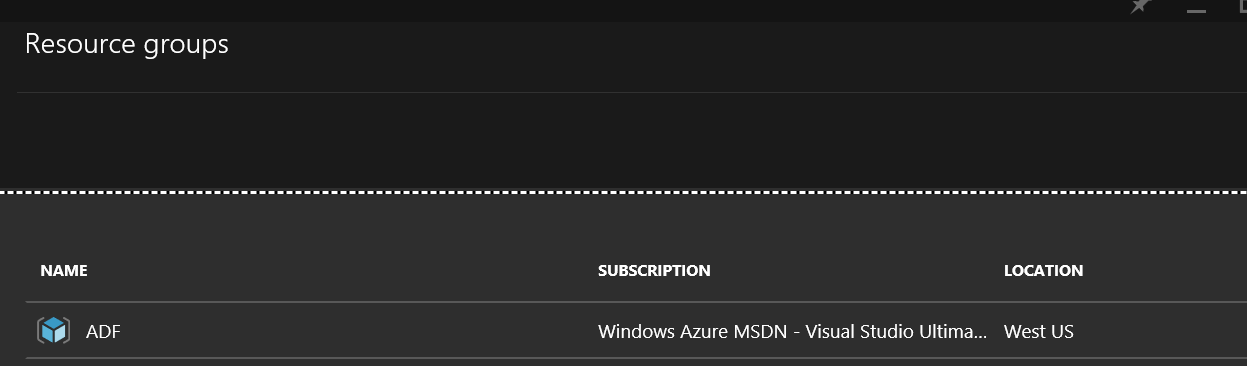
The Resource groups Azure Portal blade now appears.

Figure 3‑13

We can’t create resource groups individually, they have to be created when we create individual resources like storages, factories, etc. Every time theses individual resources are created, we have the option to specify a resource group.

#### Subscription

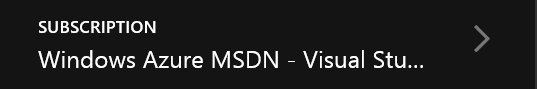


Figure 3‑14

The subscription blade allows you to select the Azure subscription you are using (and which will be invoiced) to interact with the storage. As we have talked about in Pricing tier section, depending on the type of storage you’ll be using and, as we’ll talk about in the Location section, storage is not free. You receive a monthly invoice for it as part of your Azure subscription.

#### Location

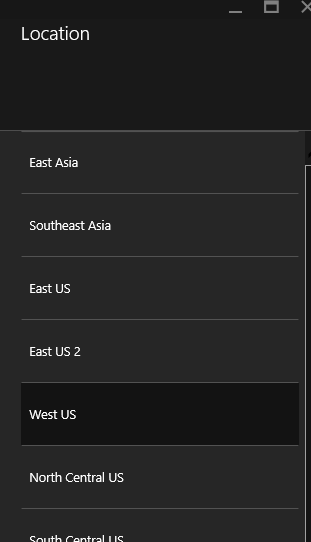


Figure 3‑15

When you create your storage resource, you have the option to create in different data centers in the world. The location that you choose has implications only if you plan to transfer data from one storage resource zone to another. And it depends if the data arrives in the zone (inbound) or goes out of it (outbound). For inbound traffic, there are no charges. But you will have to pay for outbound data or file movements.

Here are the various zones available at time of writing:

•Zone 1: US West, US East, US North Central, US South Central, US East 2, US Central, Europe West, Europe North

•Zone 2: Asia Pacific East, Asia Pacific Southeast, Japan East, Japan West, Australia East, Australia Southeast

•Zone 3: Brazil South

Notice that there are no charges for data transfer between locations that are in the same zone. For example, if you use Azure Data Factory to copy data from a blob storage that is located fin East-US to a SQL database located in Asia, depending on how much data we are transferring, charges may be applied. At time of writing, all monthly transfers below 5 gigabytes are free. You can refer to Microsoft Azure Data Transfers Pricing Details ( [http://azure.microsoft.com/en-us/pricing/details/data-transfers/#](http://azure.microsoft.com/en-us/pricing/details/data-transfers/)) if you want to see all available options related to data transfers between zones.

## Sample case used throughout this book

The case we’ll use in this book is based on AdventureWorks LT database. This database was available on [codeplex](http://msftdbprodsamples.codeplex.com/releases/view/55330) for all recent SQL Server edition up to SQL Server 2012. On the top of this database, we created a small data warehouse called AdventureWorksLTDW. This book’s support site will provide the necessary SQL Server data files for both databases.

### AdventureWorksLT database



Figure 3‑16

Figure 3‑17

### AdventureWorks LT DW

This database is a data warehouse that have been built on top of the operational database.

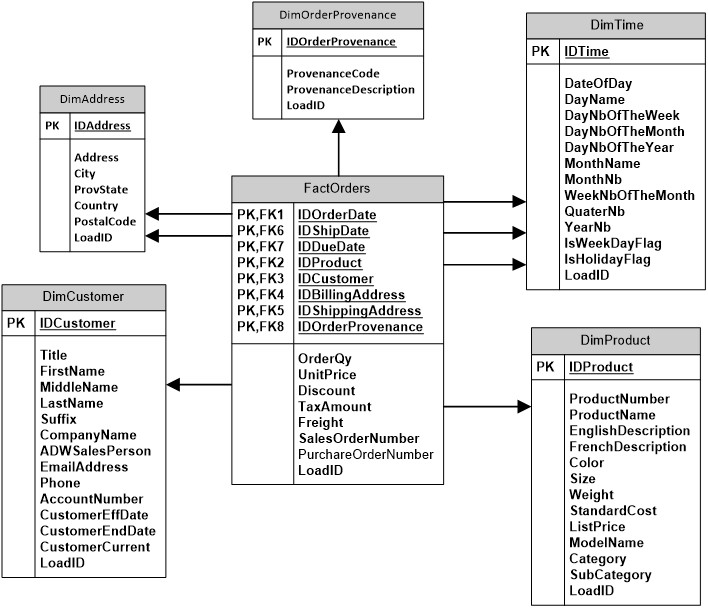


Figure 3‑18

The following list describes the main differences between AdventureWorksLT and the data warehouse.

* The tables “SalesOrderHeader” and “SalesOrderDetails” have been merged into a single fact table: FactOrders.
* We now have a single point of reference for all dates: DimDate. This dimension is referred three times in “FactOrders”. We call this type of links role playing dimension. The DimDate dimension can be used to constraint the fact table using the order, ship or due date of the order. This link provides more information’s on role playing dimensions.
* Product categories and sub categories have been denormalized in the “DimProduct” dimension table. This allow to easily navigate the fact table using the following hierarchy: Category🡪Sub category🡪products.
* We chose to keep only English and French product description. We denormalized them in the product dimension. This way, we can easily display the product in both languages.
* Customer’s addresses have their own dimension: DimAddress. Both billing and shipping addresses are linked to the fact table via the role playing dimension pattern. Since customer’s usually have several criteria that are related to sales that might change over the time, we modeled it as a [SCD type 2 dimension](https://en.wikipedia.org/wiki/Slowly_changing_dimension). This way, we’re able to compare sales for a customer before he/she was married and after.

## , Azure PowerShell

PowerShell is very useful when it comes to Azure. It allows Azure assets to be deployed from the command line and therefore automated. Once your Azure assets are scripted in PowerShell, they can be copied over to your source control software (e.g. Team Foundation Server (TFS), Subversion (SVN), Git, etc.).

### Installation

To setup Azure PowerShell, you need to download it from the following web site:

<https://azure.microsoft.com/en-us/documentation/articles/powershell-install-configure/>

The following screen appears:

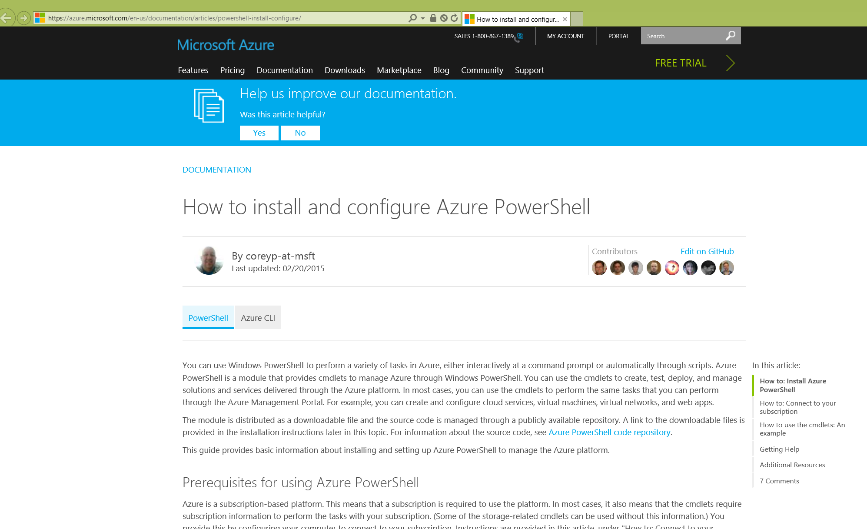


Figure 3‑19

In the upper section of the page you’ll see a section called: “How to: Install Azure PowerShell”:

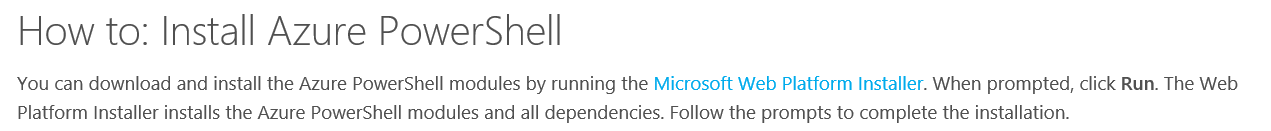


Figure 3‑20

Click on “Microsoft Web Platform Installer” to download and run the installer:



Figure 3‑21

The installer starts and the following screen appears when it’s ready to proceed at the installation:

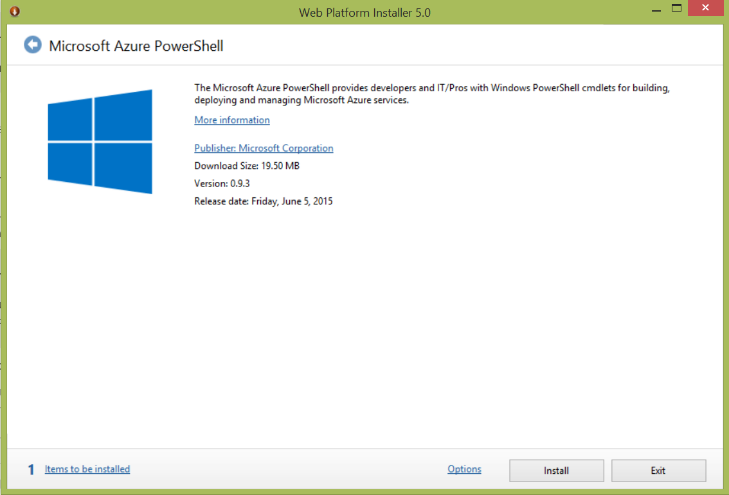


Figure 3‑22

You’ll notice that there’s at least one item to install as indicated at the bottom left of the screen:

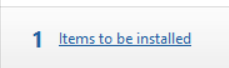


Figure 3‑23

You can click on the text “Items to be installed” to see what’s gonna be installed within this installation:

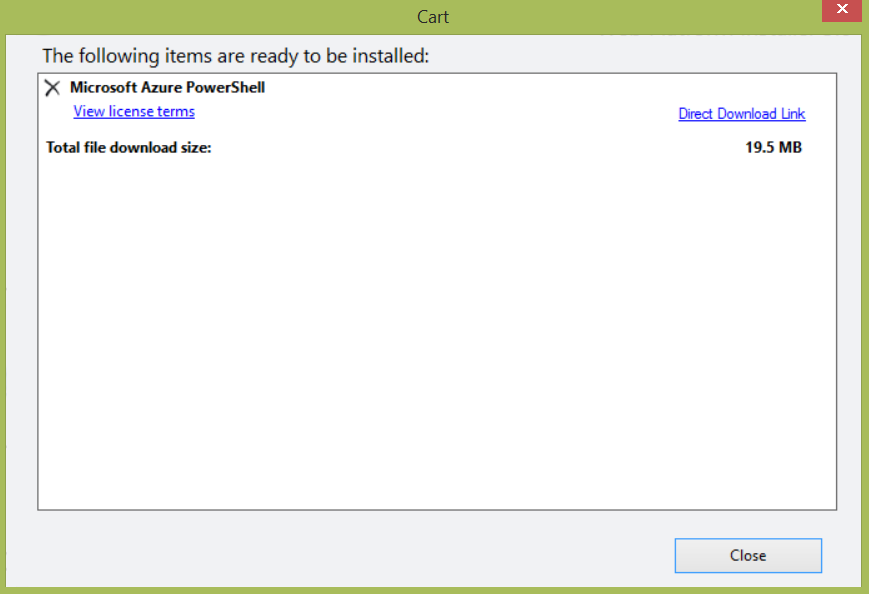


Figure 3‑24

Click close and click install in the main installer screen (Figure 3‑20). You’ll be directed to a prerequisites window telling you that you accept installation and licence terms:

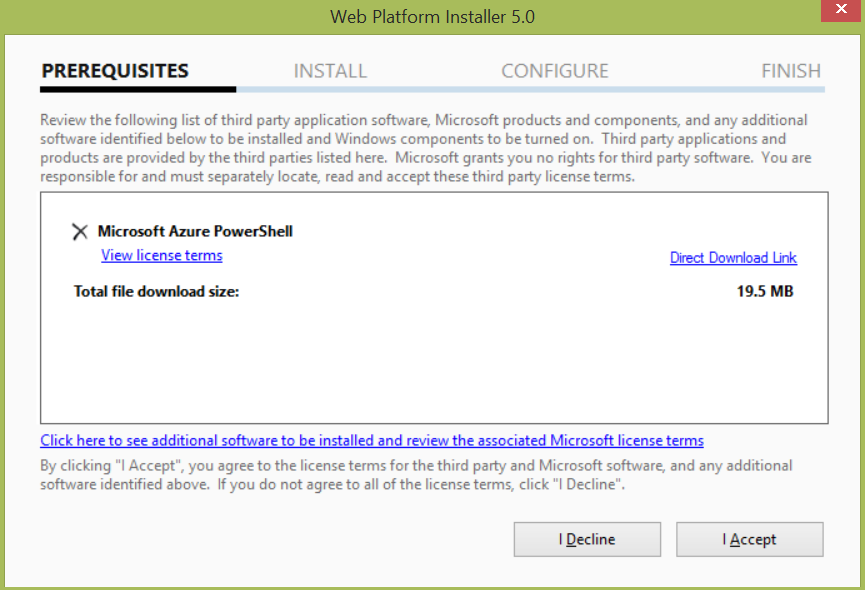


Figure 3‑25

Click I Accept to begin installation process:

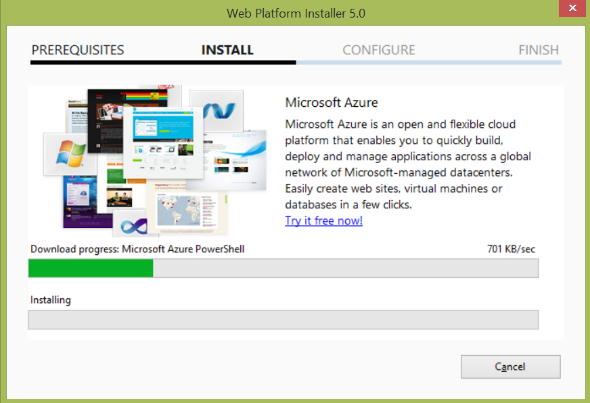


Figure 3‑26

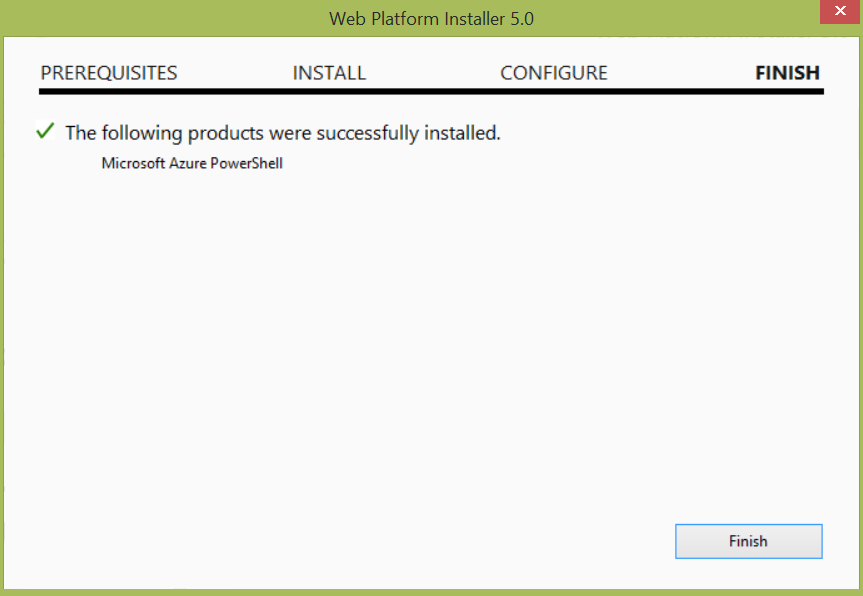
Once the installation finishes successfully, the following screen appears, click Finish to get rid of the screen. Installation of Azure PowerShell has now completed.  
  


Figure 3‑27

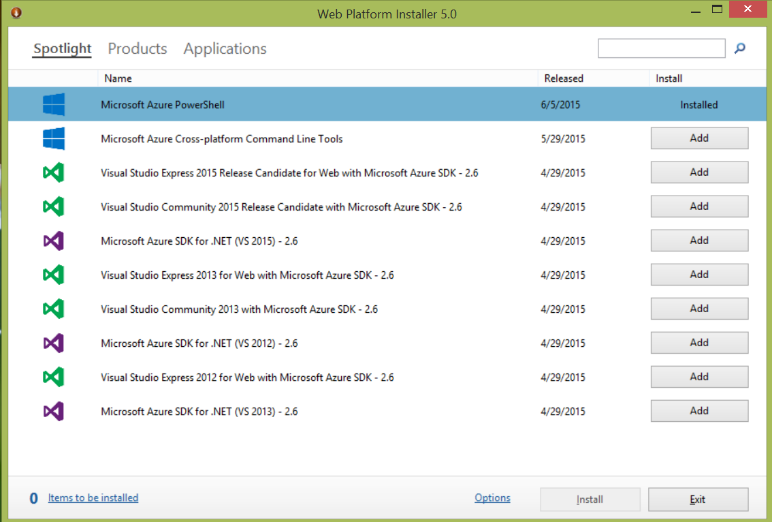
The installer allows you to install supplementary modules, click on the Exit button to close the screen:  
  


Figure 3‑28

### Verify Azure PowerShell installation

#### Windows 7

#### Windows 8.1

From the start screen, enter Azure PowerShell in the search text box and click Microsoft Azure PowerShell in the search results

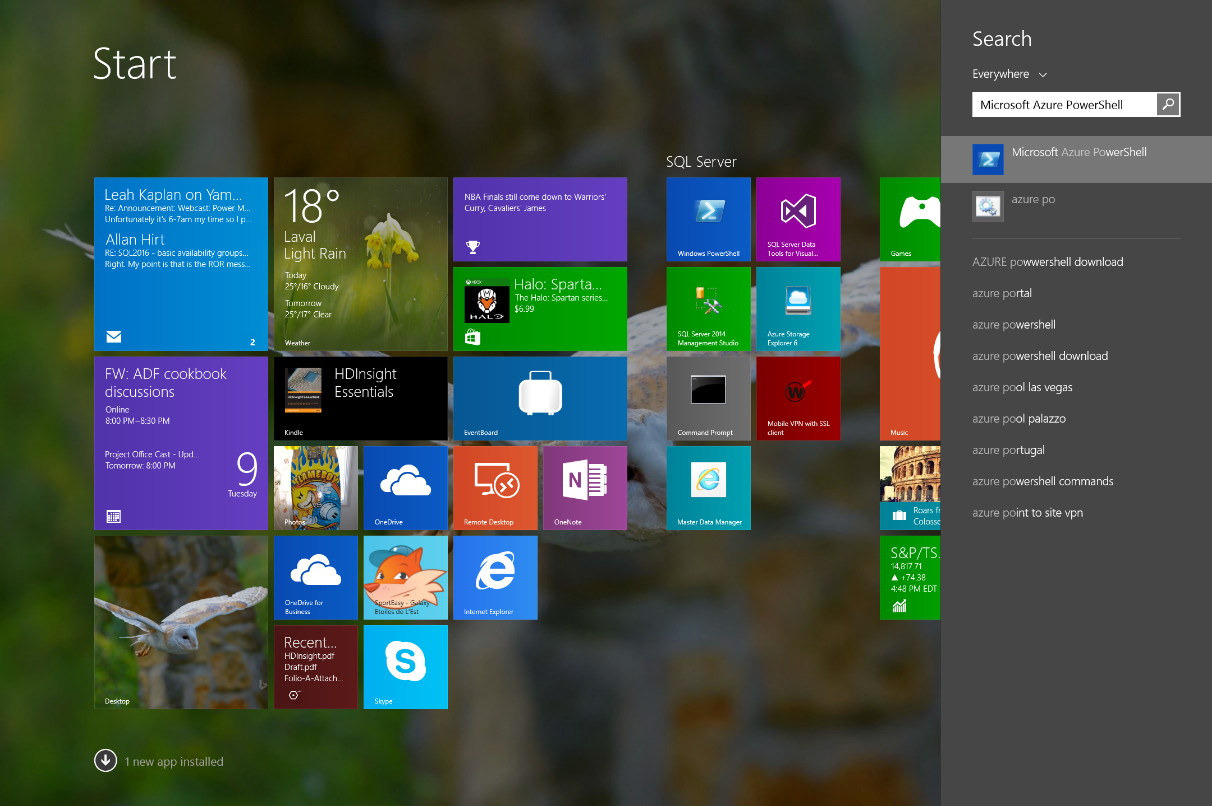


Figure 3‑29

### Windows 10

The following screen appears confirming that Azure PowerShell is correctly installed and works properly:

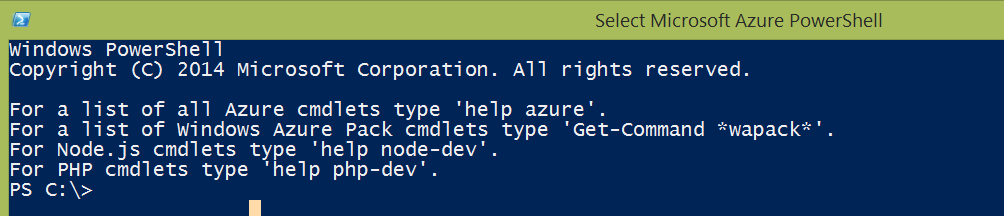


Figure 3‑30

To have a faster access to Azure PowerShell, you can also pin the program to your task bar. To do so, from the search result in Windows 8.1 or the all apps list in Windows 10, right-click on Microsoft Azure PowerShell and select the “Pin to taskbar” option from the drop down menu.

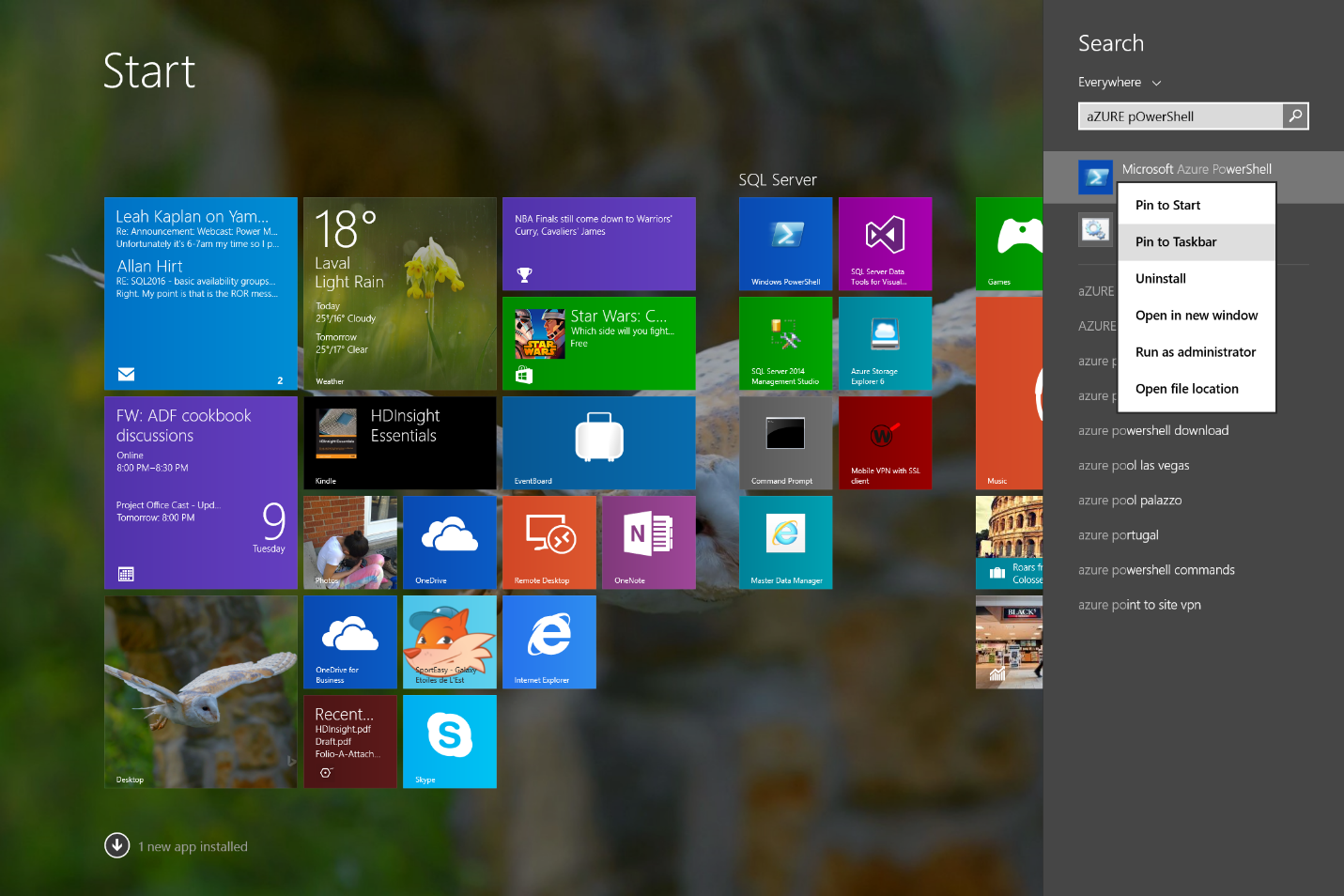


Figure 3‑31

Now, go to the desktop and you should see the following icon on your taskbar:



Figure 3‑32

Now, when you right-click on the icon in the taskbar, you can see these options:

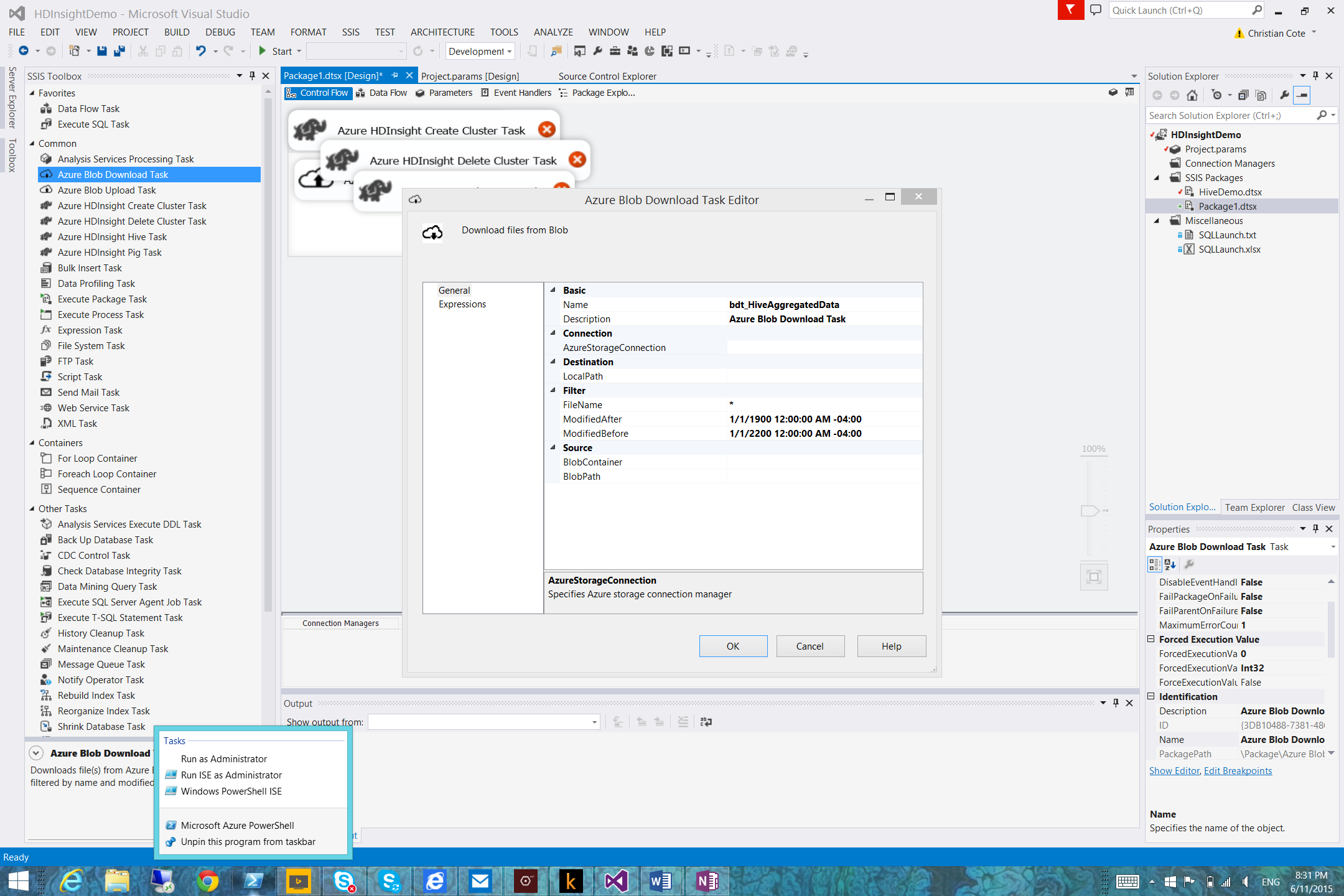


Figure 3‑33

You now have the option to open the following programs:

|  |  |
| --- | --- |
| Menu choice | Description |
| Run as Administrator | Opens regular Azure PowerShell like the one referred by Figure 3‑28 using local administrator privileges. |
| Run ISE as Administrator | Opens Windows PowerShell Integrated Scripting Environment (ISE) using local administrator privileges.(Figure 3‑32) |
| Windows PowerShell ISE | Opens Windows PowerShell Integrated Scripting Environment (ISE) using current user credentials. (Figure 3‑32) |

Here’s a screen capture of the Windows PowerShell Integrated Scripting Environment (ISE)



Figure 3‑34

This version of PowerShell has several capabilities compared to the “plain” version you’ve seen before (Figure 3‑28) such as multiline edition, debugging and possibility to run part of a script. You can learn more about PowerShell ISE here: <https://technet.microsoft.com/en-us/library/dd315244.aspx>.