How-to use Datazen to generate Azure Billing Rich Dashboards



Example documentation

Prepared for

**Technical Community**

21-May-15

Version 1.0 Draft

Prepared by

**Marcus Milhomem - CSA**

**Rober Torres – TSP Data Plat.**

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
| 05/19/2015 | Rober Torres | 1.0 | Development of the initial solution |
| 05/20/2015 | Marcus Milhomem | 1.0 | Documentation, Draft Creation and Tests |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Position | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 6](#_Toc420012983)

[2 Requirements 7](#_Toc420012984)

[2.1 Naming Conventions 7](#_Toc420012985)

[3 Setting up The Environment 8](#_Toc420012986)

[3.1 Preparing The Azure SQL DB 8](#_Toc420012987)

[3.1.1 Create the Azure DB 8](#_Toc420012988)

[3.1.2 Pin the Azure Resource Group 11](#_Toc420012989)

[3.1.3 Configuring the Database Access. 12](#_Toc420012990)

[3.2 Creating the Data Base Schema 15](#_Toc420012991)

[3.3 Creating a VM for Installing Datazen 22](#_Toc420012992)

[3.3.1 Creating a VNET 22](#_Toc420012993)

[3.3.2 Creating a VM to hold Datazen Server. 25](#_Toc420012994)

[3.4 The Web App / Web job 31](#_Toc420012995)

[3.4.1 Creating the Web App 32](#_Toc420012996)

[3.4.2 Configure DB access on Web App 34](#_Toc420012997)

[3.4.3 Create The AzureWebJobsDashboard and AzureWebJobsStorage Connections 36](#_Toc420012998)

[3.4.4 Create the Web Job 39](#_Toc420012999)

[4 Installing and Configuring Datazen 44](#_Toc420013000)

[4.1 Installing Datazen. 44](#_Toc420013001)

[4.2 Configuring Datazen Access 56](#_Toc420013002)

[5 Installing and Configuring Datazen Clients 66](#_Toc420013003)

[5.1 Installing the Clients 66](#_Toc420013004)

[5.2 Configuring the Client and Creating the First Dashboard. 68](#_Toc420013005)

[5.3 Creating your very first Dashboard. 72](#_Toc420013006)

[5.4 Publishing a Dashboard 81](#_Toc420013007)

[5.5 Viewing Dashboards 83](#_Toc420013008)

[6 References 86](#_Toc420013009)

# Introduction

The Idea of this document is to be a high level how-to guide to use the recent acquire by Microsoft Datazen software as a dashboard generator for the Billing of Azure. This idea was formulated after a customer complained about the poor graphical visualization tools he had to understand how he was spending his Azure EA.

As this is a how-to guide to proof a functionality, the possibility of using our own tools to solve an issue that maybe more than one customer is facing, we will not worry about assembling a high availability environment. But if you pretend to use this guide to assemble a production environment to one of your customers, it´s strongly recommended that you read the Datazen and Azure documentation on how to assemble high available environemnts.

To be able to use the Datazen the customer needs to have a valid Microsoft SA with at least one SQL Server 2008 or newer on it. If this condition is met, he is eligible to use the Datazen Software.

# Requirements

You should be familiar with Windows Azure concepts in general. You should also be familiar with SQL Server and Visual Studio.

For this example, you will need the following:

1. One Desktop with Visual studio Installed
2. One Valid Azure subscription
3. One Windows 2012 R2 Datacenter Server on Azure
4. One Web App created on Azure
5. One Azure SQL DB
6. One Storage Account for the disc of the Windows Server on Azure
7. Optionally you could create a VNET to hold the Windows Server VM. But as we are going to use only one VM for this example, doing so it´s optional
8. Download the Datazen Installer from the Datazen website - <http://www.datazen.com/>

## Naming Conventions

We recommend establishing a naming convention to uniquely identify the Azure Elements described. For example DEV-Datazen could be used on the beginning of every server and service for this environment. If you are using the new Azure Portal (<http://portal.azure.com>), its recommended that you create a Resource Group to group all the servers and services used on this environment.

# Setting up The Environment

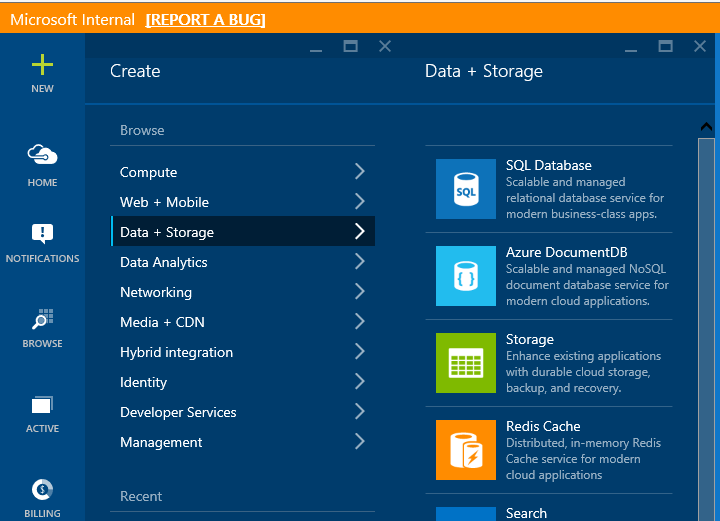
Now we will follow a step by step guide to prepare the environment that will connect to the Azure API, collect the billing data and store it in a SQL DB on Azure SQL.

## Preparing The Azure SQL DB

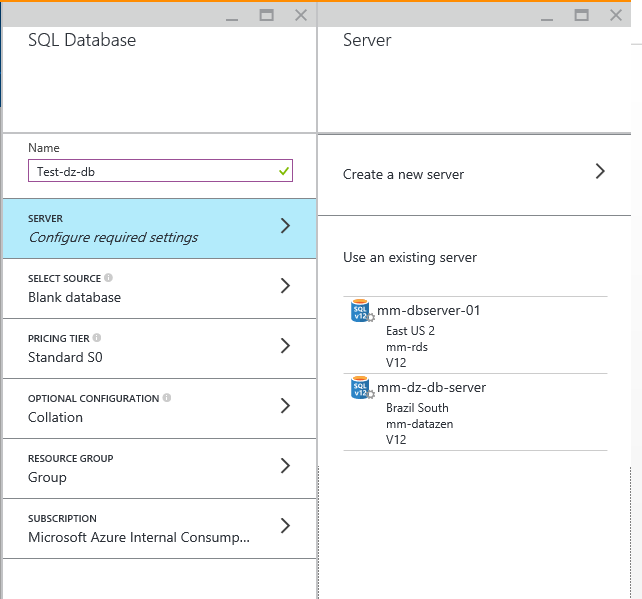
The first thing we need to do is to create a Azure SQL DB and create our database on it. Please follow the steps below to create the infrastructure and database needed. Optionally you could also use a database already created.

### Create the Azure DB

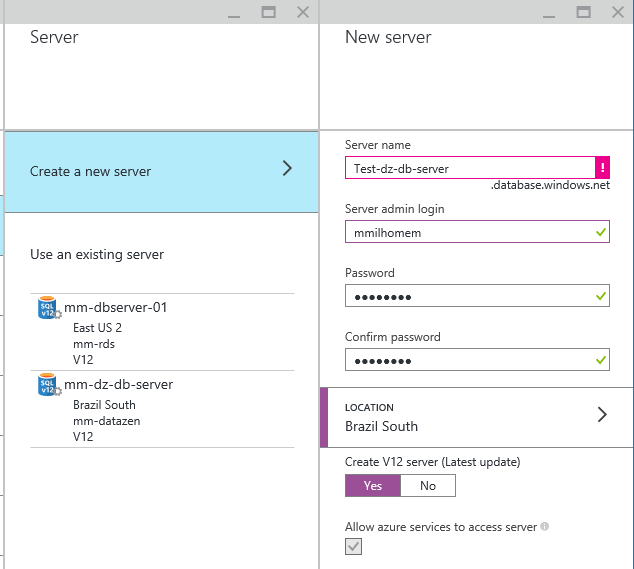
First login into Azure and create a new Azure SQL Server.



After you select the SQL Database option, enter a name and select the server option to open the server configuration tile.



In this case we will use the name Test-dz-db as our database and also we will choose to create a new database server. To do this, select create a new server and enter the information below:

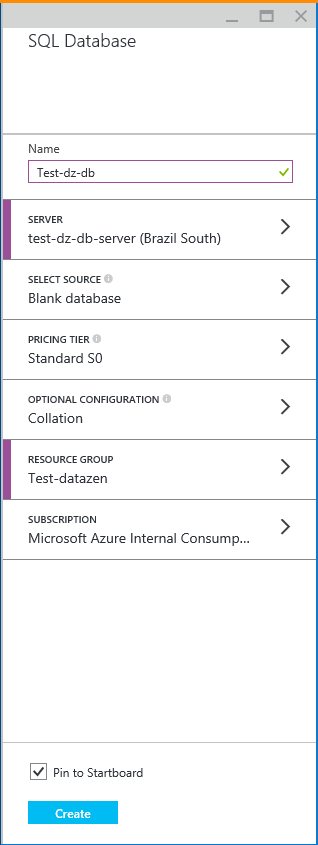


On this example we used the name Test-dz-db-server as our Azure SQL database server. We also entered the name of the administrator user and it´s password. You can enter any username and password you want as long as it comply with the Azure security policies. It´s important to leave the option “Allow azure services to access server” checked as we will create a web job that will access that data base. After you configure this tile click the OK button.

Next, click on the Resource Group Option to create a new Resource Group to group all of our environment into one Resource Group. This step is optional, but recommended to assemble every resource together making it easy to administrate later. In this case we are going to use the name Test-datazen for the Resource Group.



After filling the name, click on the OK Button. And Finally check to see if every information that you entered is right and click on the create button.

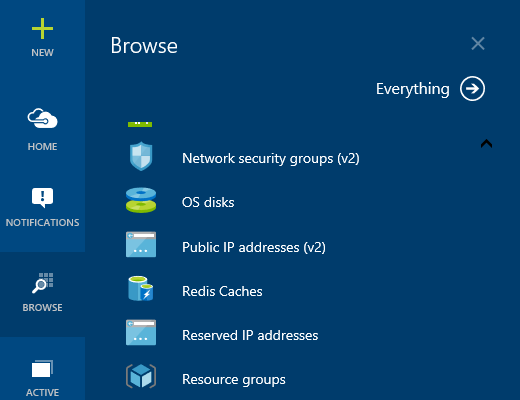


It can take up to 5 minutes to create the database. Please wait it finishes to proceed to the next step.

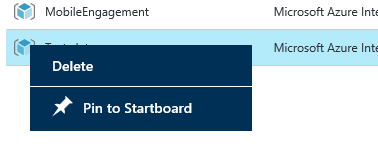
### Pin the Azure Resource Group

First to make our lives easy, as an optional step you can pin the Test-datazen Resource Group on the Azure dashboard, as we are assembling all of our resource on it.

To do so, go to browse and select resource groups



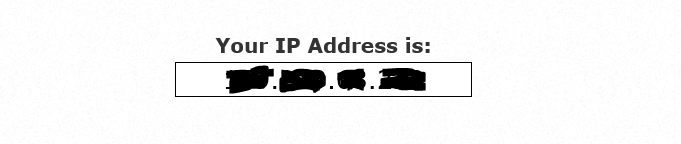
After that, right click on the Resource Group Test-Datazen and select the option Pin to start board



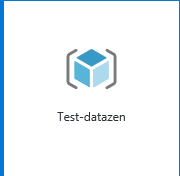
### Configuring the Database Access.

The Next step that we need to do is to configure a rule so we can access the database created on the above steps so that our desktop with Visual studio can access it and create our SQL schema.

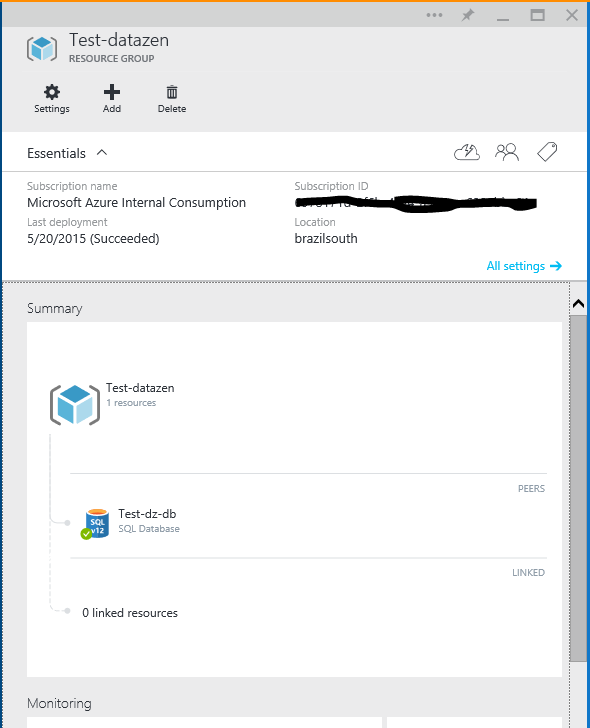
The first thing you need to do is to discovery what public IP are your desktop using. To do this you can use the show my IP web site at <http://showip.net/>. This web site will show your current public IP, and you are going to use it to create a rule allowing it to access the data base we just created.



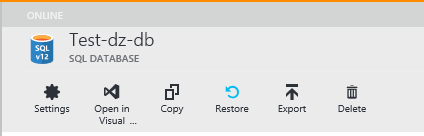
With the IP address go to the database that you just created. To do so, you can click on the Test-datazen Resource Group create on the above steps. Go to your start board (you can do this by clicking on the home button on the left toolbar on Azure and then on the Test-datazen resource group.



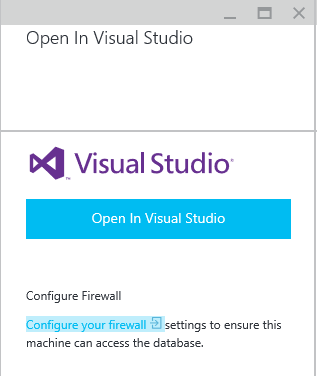
After that, the Resource Group tile will open. You can then click on the Test-dz-db to open the data base tile.



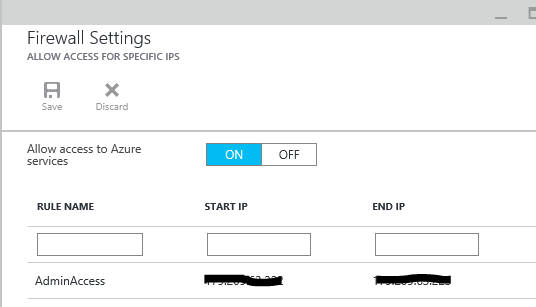
After the Data base tile opens, you will click on the option open in visual studio button.



This will open a new tile with a link called: Configure your Firewall Rules. Click on it to open the firewall rules tile.



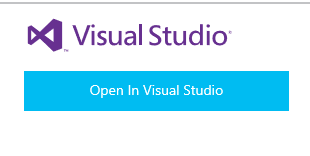
After the firewall rules tile opens, Create a rule allowing your public IP (the one we discover on the above steps) and click on the save button. Wait for the confirmation that your rule was saved successfully.



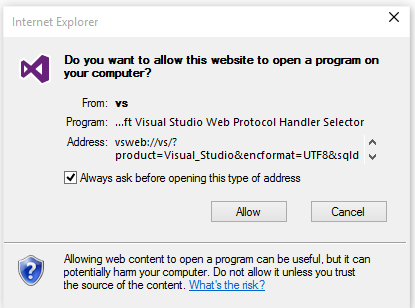
You will then return to the Open In Visual studio Tile.

## Creating the Data Base Schema

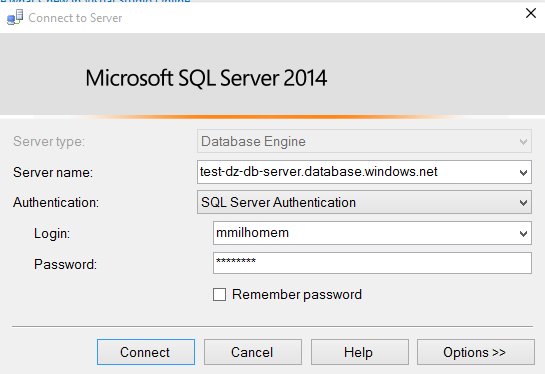
On the Open In Visual studio Tile (please refer to the above step to see how to get to this tile) click on the link open In Visual studio



This will open a dialog box requesting to access Visual studio. Click on the allow button



After Visual Studio Opens, it will show a dialog requesting the username and password that you used when creating the data base server. Fill in the information and click connect.



After you successfully connects, right click on the data base Test-dz-db and select the option to do a New Query. On the new query screen you will enter the following script. Optionally, you can also download the script from GitHub at <https://github.com/mmilhomem/AzureBillingDZ>

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[BillingDetail](

[BillingID] [int] IDENTITY(1,1) NOT NULL,

[EANumber] [int] NOT NULL,

[AccountOwnerId] [nvarchar](255) NULL,

[AccountName] [nvarchar](255) NULL,

[ServiceAdministratorId] [nvarchar](255) NULL,

[SubscriptionId] [float] NULL,

[SubscriptionGuid] [nvarchar](255) NULL,

[SubscriptionName] [nvarchar](255) NULL,

[Date] [datetime] NOT NULL,

[Month] [float] NULL,

[Day] [float] NULL,

[Year] [float] NULL,

[Product] [nvarchar](255) NULL,

[ResourceGUID] [nvarchar](255) NOT NULL,

[Service] [nvarchar](255) NULL,

[ServiceType] [nvarchar](255) NULL,

[ServiceRegion] [nvarchar](255) NULL,

[ServiceResource] [nvarchar](255) NULL,

[ResourceQtyConsumed] [float] NULL,

[ResourceRate] [float] NULL,

[ExtendedCost] [float] NULL,

[ServiceSubRegion] [nvarchar](255) NULL,

[ServiceInfo] [nvarchar](255) NULL,

[Component] [nvarchar](255) NULL,

[ServiceInfo1] [nvarchar](255) NULL,

[ServiceInfo2] [nvarchar](255) NULL,

[AdditionalInfo] [nvarchar](255) NULL,

[Tags] [nvarchar](255) NULL,

[StoreServiceIdentifier] [nvarchar](255) NULL,

[DepartmentName] [nvarchar](255) NULL,

[CostCenter] [nvarchar](255) NULL,

CONSTRAINT [PK\_BillingDetail\_1] PRIMARY KEY CLUSTERED

(

[BillingID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON)

)

GO

/\*\*\*\*\*\* Object: Table [dbo].[BillingSummary] Script Date: 21/05/2015 14:11:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[BillingSummary](

[BillingID] [int] IDENTITY(1,1) NOT NULL,

[EANumber] [int] NULL,

[CurrencyCode] [varchar](3) NULL,

[Month] [int] NULL,

[MonthName] [nvarchar](30) NULL,

[BeginningBalance] [float] NULL,

[NewPurchases] [float] NULL,

[Adjustments] [float] NULL,

[UtilizedSubtractedFromBalance] [float] NULL,

[EndingBalance] [float] NULL,

[Overage] [float] NULL,

[ServiceOverage] [float] NULL,

[ChargesBilledSeparately] [float] NULL,

[TotalOverage] [float] NULL,

[TotalUsage] [float] NULL,

[AzureMarketplaceServiceCharges] [float] NULL,

[ReportGenerationDate] [datetime] NULL,

PRIMARY KEY CLUSTERED

(

[BillingID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON)

)

GO

SET ANSI\_PADDING OFF

GO

/\*\*\*\*\*\* Object: Table [dbo].[EAContracts] Script Date: 21/05/2015 14:11:04 \*\*\*\*\*\*/

SET ANSI\_NULLS OFF

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[EAContracts](

[EANumber] [int] NOT NULL,

[EAKey] [text] NOT NULL,

[Customer] [nvarchar](100) NULL,

[LastReportDate] [smalldatetime] NULL,

[IsActive] bit null,

CONSTRAINT [PrimaryKey\_a4ea1bf0-df44-4398-ac16-fe7fc4ee36ac] PRIMARY KEY CLUSTERED

(

[EANumber] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON)

)

GO

ALTER TABLE [dbo].[BillingDetail] WITH CHECK ADD CONSTRAINT [FK\_BillingDetail\_EAContracts] FOREIGN KEY([EANumber])

REFERENCES [dbo].[EAContracts] ([EANumber])

GO

ALTER TABLE [dbo].[BillingDetail] CHECK CONSTRAINT [FK\_BillingDetail\_EAContracts]

GO

ALTER TABLE [dbo].[BillingSummary] WITH CHECK ADD CONSTRAINT [FK\_BillingSummary\_EAContracts] FOREIGN KEY([EANumber])

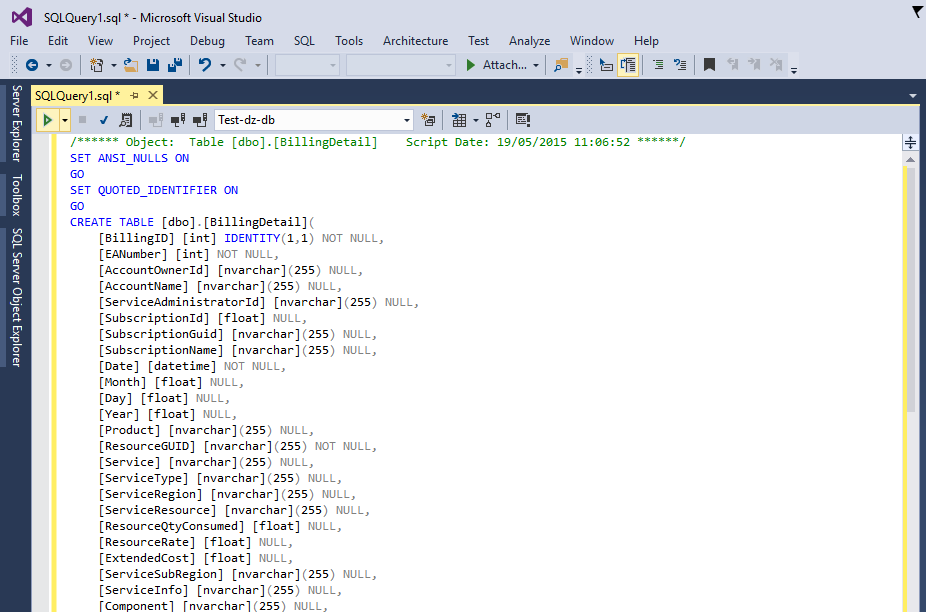
REFERENCES [dbo].[EAContracts] ([EANumber])

GO

ALTER TABLE [dbo].[BillingSummary] CHECK CONSTRAINT [FK\_BillingSummary\_EAContracts]

GO

After you entered the SQL script, click on the execute button.



Wait for the confirmation that the command ran successfully.

Now we need to add the EA Number and key of the customer you want to access billing information. You can add more than one EA. To do so you will need to ask your customer to access it´s Account Portal, get his EA number and generate a temporary access key to give to you.

Microsoft have a test EA number and Key. We will use those on this guide, but when deploying this get the right EA number and key from your customer.

EA Number = 100

EA Key = ‘eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Il92ZHZieDJYRUkyb2l5N09abVpVelZGelM2TSJ9.eyJpc3MiOiJlYS5taWNyb3NvZnRhenVyZS5jb20iLCJhdWQiOiJjbGllbnQuZWEubWljcm9zb2Z0YXp1cmUuY29tIiwibmJmIjoxNDI4NDYwOTgxLCJleHAiOjE0NDQyNzIxODEsIkVucm9sbG1lbnROdW1iZXIiOiIxMDAiLCJJZCI6Ijk3OGQ5MGRjLTEyZjUtNDVmZC1hZWY2LWNiZTQ0ZDE0NmFmNiIsIlJlcG9ydFZpZXciOiJTeXN0ZW0iLCJQYXJ0bmVySWQiOiIifQ.QEHdbHmUz4TXO8Ljhn8ktnt5m7rHO0hsRrDLTvoQP4luL-8gy98CywFi2OByhfPL6UY5Vx8MYtwTQhMbzefS6Cd8hNopP74AL1ENLT-WGn42JG2mWesizIuN1QHV-obZR5rKDT57leurgEWy3LscWlwM0v5vhtk-9e5p8\_9YgYqnF3rC05l1kUBQYzEIeYeMxcp8YxsqlnQNoEdCIA4UN08Py7zjr0ohCYbkm76a0-XdquqOTdSKGrnFkmwPxzjWMkuX3f2zpck\_Ps8x8TVBpfvywly0QjUbN0ssmLV20zrS8FqkKSdx13uQVzM3MXCBylu5WvhTKE\_ogRJdxPpfiA’

To do so, right click on the Test-dz-db and select the option new query. Paste the following command on it and click the execute button. Do not forget to replace the test EA number and key with the ones from your customer.

insert into EAContracts values ('100', 'eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Il92ZHZieDJYRUkyb2l5N09abVpVelZGelM2TSJ9.eyJpc3MiOiJlYS5taWNyb3NvZnRhenVyZS5jb20iLCJhdWQiOiJjbGllbnQuZWEubWljcm9zb2Z0YXp1cmUuY29tIiwibmJmIjoxNDI4NDYwOTgxLCJleHAiOjE0NDQyNzIxODEsIkVucm9sbG1lbnROdW1iZXIiOiIxMDAiLCJJZCI6Ijk3OGQ5MGRjLTEyZjUtNDVmZC1hZWY2LWNiZTQ0ZDE0NmFmNiIsIlJlcG9ydFZpZXciOiJTeXN0ZW0iLCJQYXJ0bmVySWQiOiIifQ.QEHdbHmUz4TXO8Ljhn8ktnt5m7rHO0hsRrDLTvoQP4luL-8gy98CywFi2OByhfPL6UY5Vx8MYtwTQhMbzefS6Cd8hNopP74AL1ENLT-WGn42JG2mWesizIuN1QHV-obZR5rKDT57leurgEWy3LscWlwM0v5vhtk-9e5p8\_9YgYqnF3rC05l1kUBQYzEIeYeMxcp8YxsqlnQNoEdCIA4UN08Py7zjr0ohCYbkm76a0-XdquqOTdSKGrnFkmwPxzjWMkuX3f2zpck\_Ps8x8TVBpfvywly0QjUbN0ssmLV20zrS8FqkKSdx13uQVzM3MXCBylu5WvhTKE\_ogRJdxPpfiA', 'Client01', null, '1');

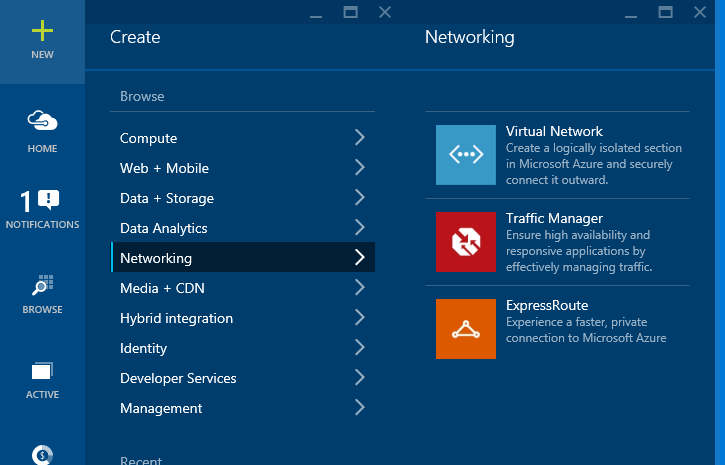
Now we are ready to deploy the Web Job

## Creating a VM for Installing Datazen

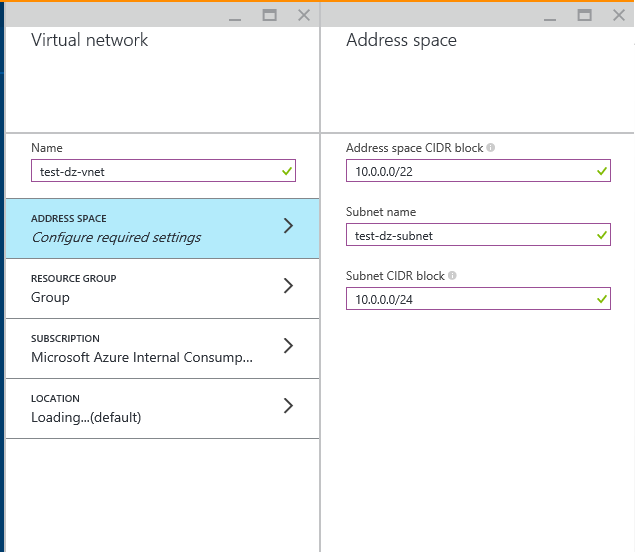
Now we will create a Windows VM to install Datazen Later on it. This VM will host Datazen and it will be our dashboard server.

### Creating a VNET

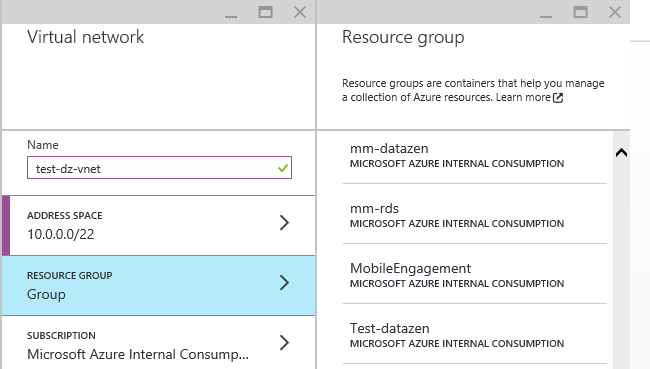
This is an optional step, but is recommended if in the future you would like to grow the Datazen environment into more than one machine. To create a VNET you need to go into the home screen and click on the new button on the top of the left toolbar. Then select VNET



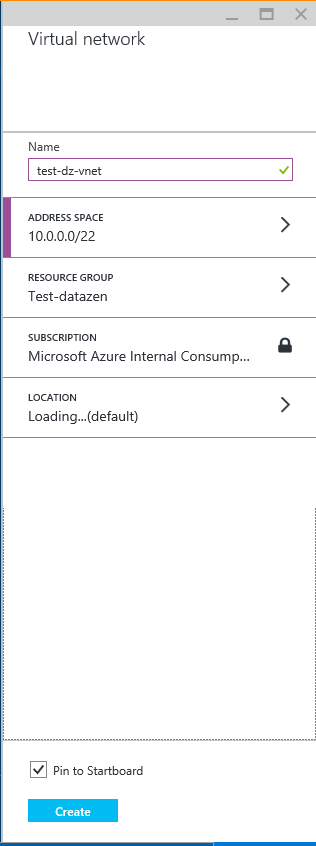
After you select VNET, a new tile will open where you will setup the VNET. Please choose a name for it and click on the address space button. And fill the network information as follows and click ok.



After it returns to the Virtual Network tile, click on the Resource Group button and select the Test-datazen Resource Group



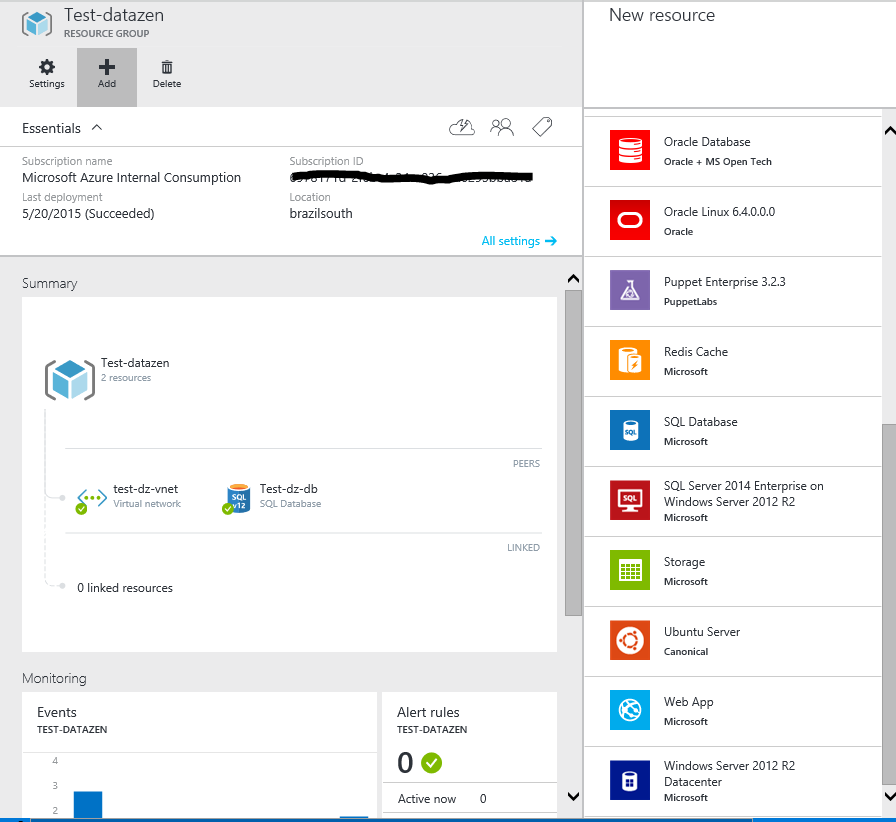
After you return to the Virtual Network tile, check to see if the information you entered are ok and click on the create button.



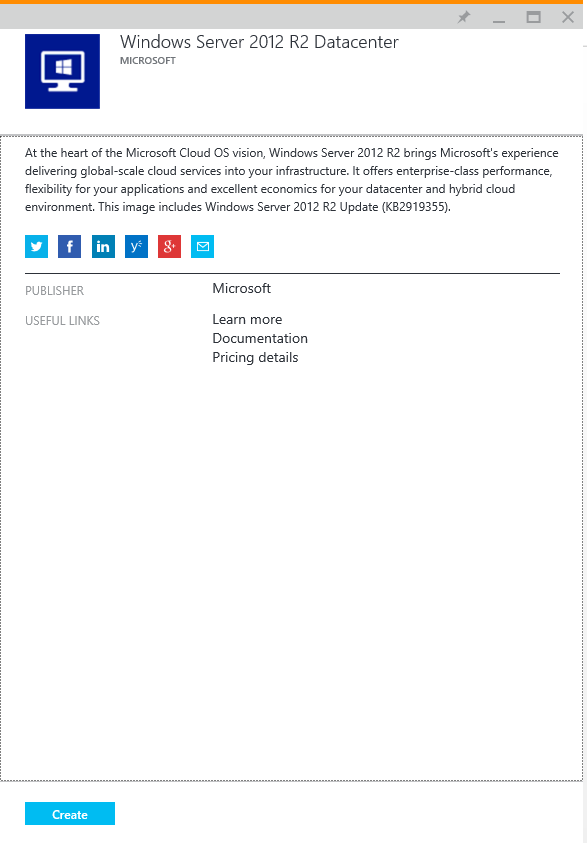
Now you will have to wait for the VNET to be created before proceeding to the next step.

### Creating a VM to hold Datazen Server.

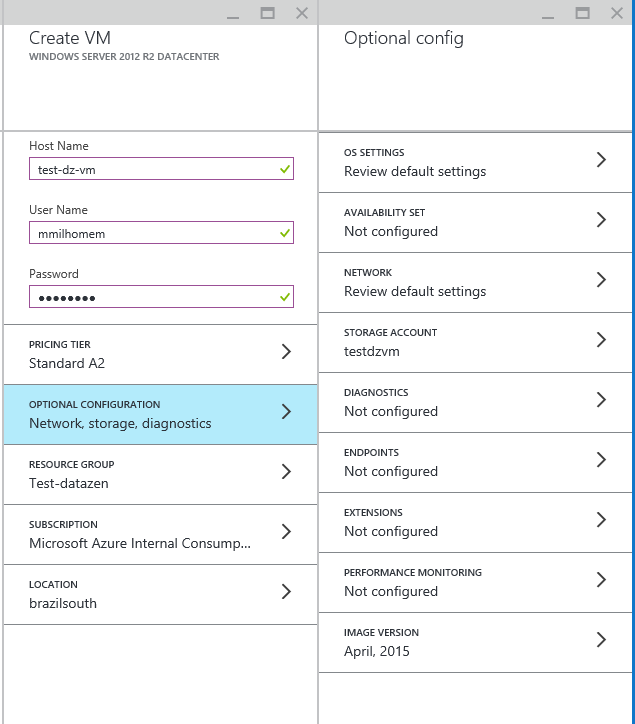
Now that we have our VNET created the next step is to create a VM with Windows Server. To do that, go to our Resource Group and select Add button on top of the toolbar. After that select Windows 2012 R2 VM option.



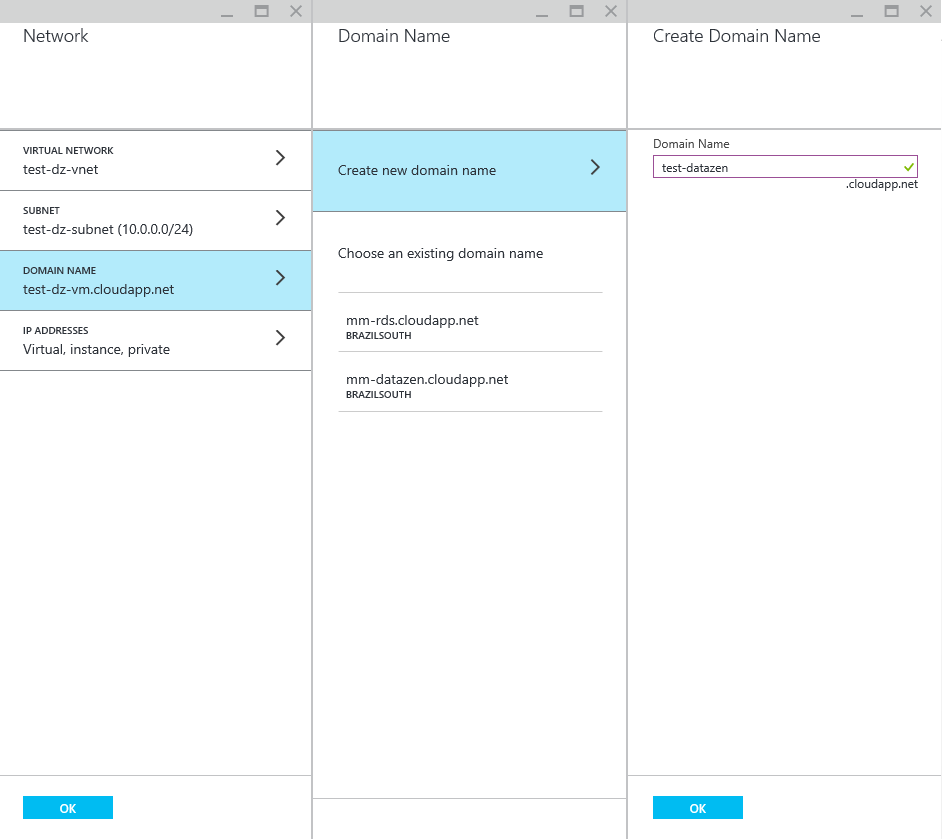
Once you click on the Windows Server Option, it will open a new tile with some information about VMs. Just click on create.



A New tile will open with information regarding the Creation of the VM. Please fill up the hostname, username and password. Choose the size that best suits you, for this example we are using an A2 and click on the Network, storage, diagnostics button.



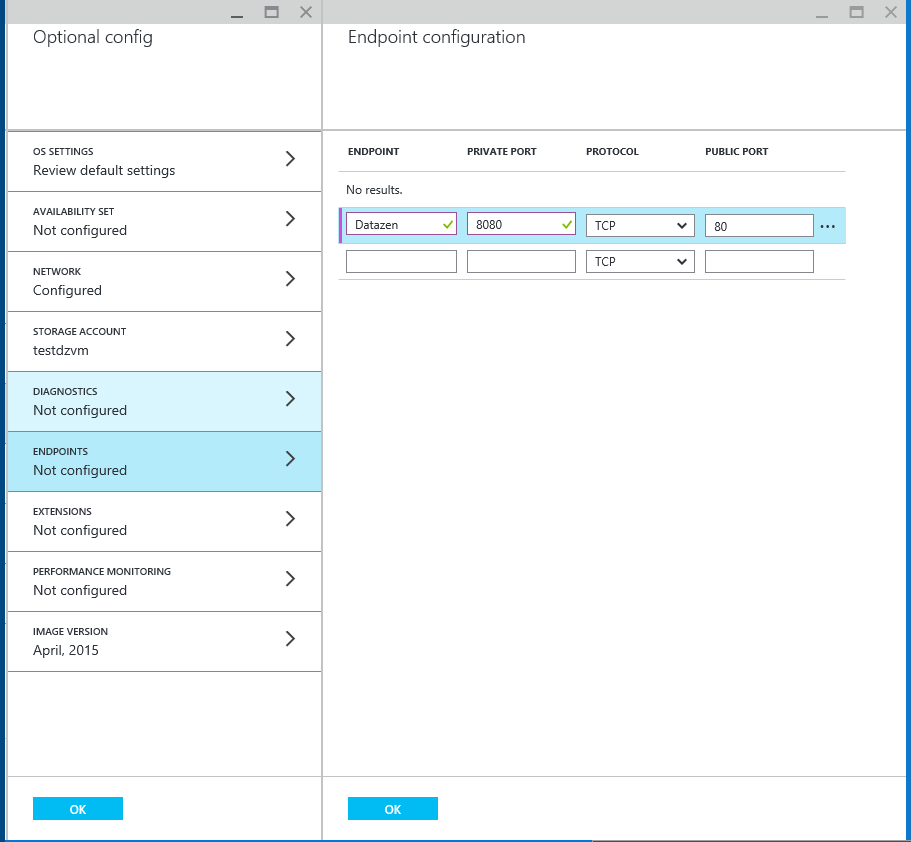
Now the first thing we need to do is to Select the Network button and them the Domain name button. When a the tile Create Domain Name appears, simply type test-datazen and press ok.



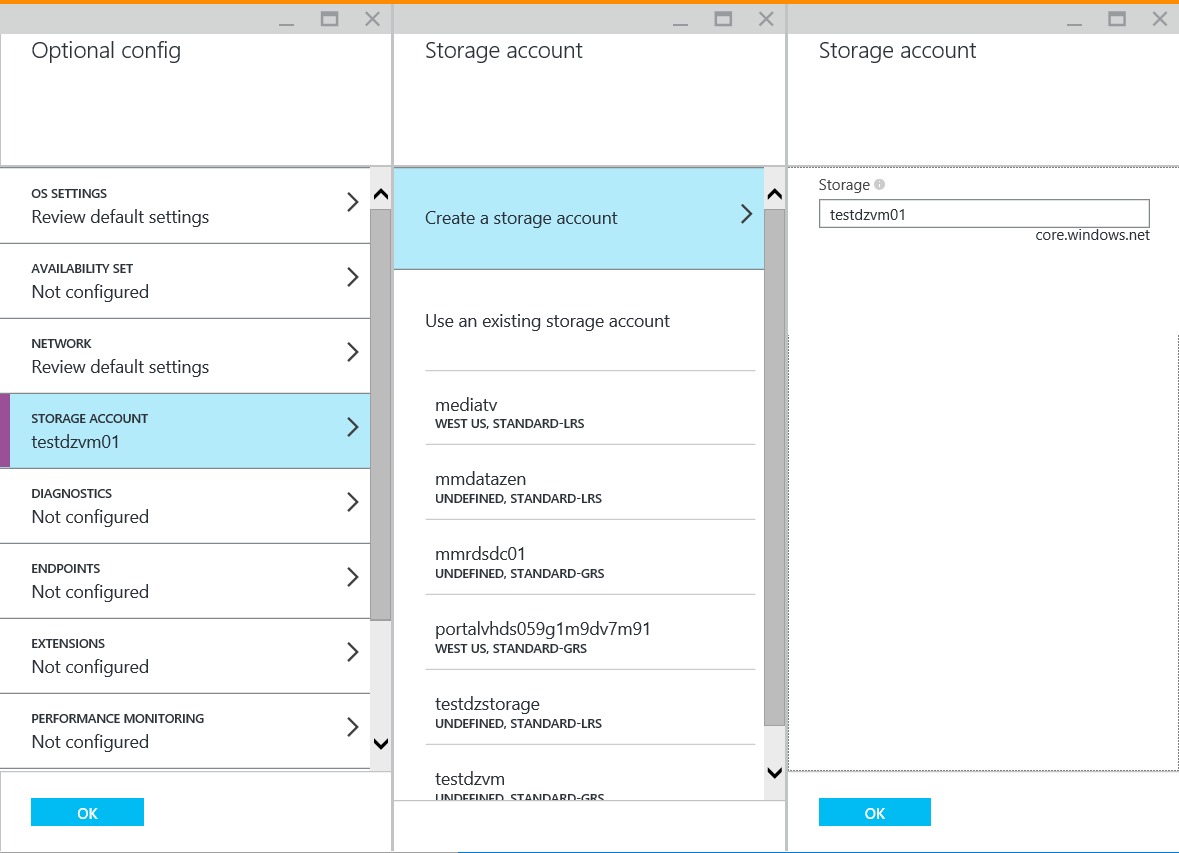
Once It returns to the Network Tile, click on the IP addresses button and fill the information as follows. Remember to change the ip address assignment to static and fill an ip address and hit ok.



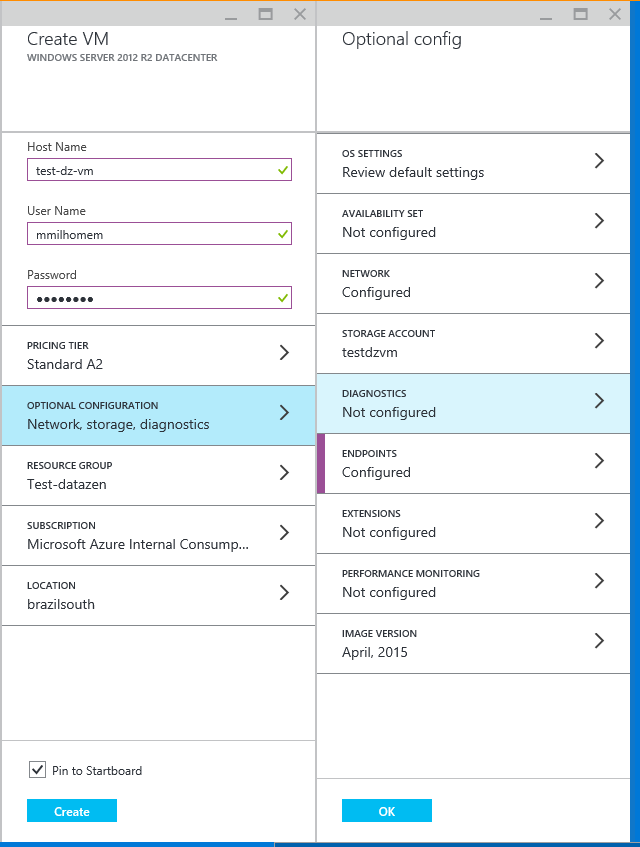
Once it returns to the Network tile, hit ok again. Once it returns to the Optional Config tile, select endpoint. Once the endpoint tile appears, fill with the following information. We will need this NAT rule later to access the Datazen server from the internet. Make note of the internal and public ports that you are using. We will need them later. When done, click the ok button.



When you return to the Optional config tile, Click on the storage Account button. We will now create a storage account to hold the VM disks. This account will also be used by the Web Job later. When the New tile opens, select “Create a storage account”, name it and click ok. After you hit ok you will return to the Option Config tile.



When you return to the Optional config tile, hit ok and them create. You will need to wait for the VM to get created before continue with this guide.

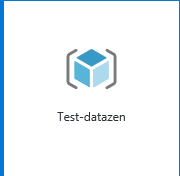


## The Web App / Web job

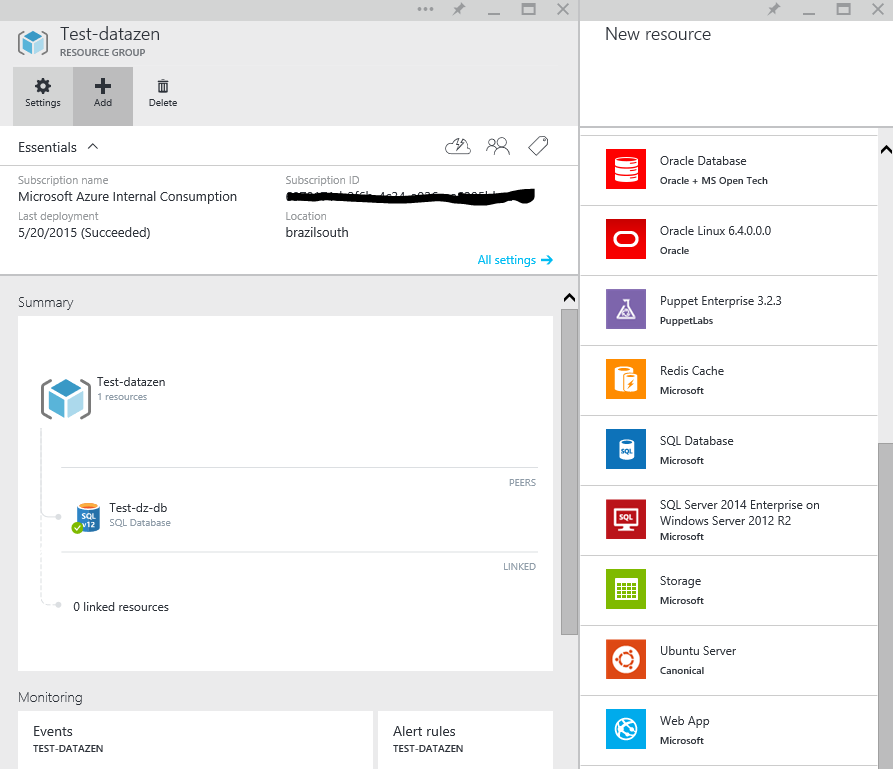
Now that the database piece of the solution is ready, the next step is to create a Web App and within that Web App, create a web job that will run periodically and will collect billing information from the Azure API and it will store it into the data base that we just created on the steps above.

### Creating the Web App

To create a web app return to the Azure Portal (<http://portal.azure.com>) and click on the Test-datazen Resource Group.



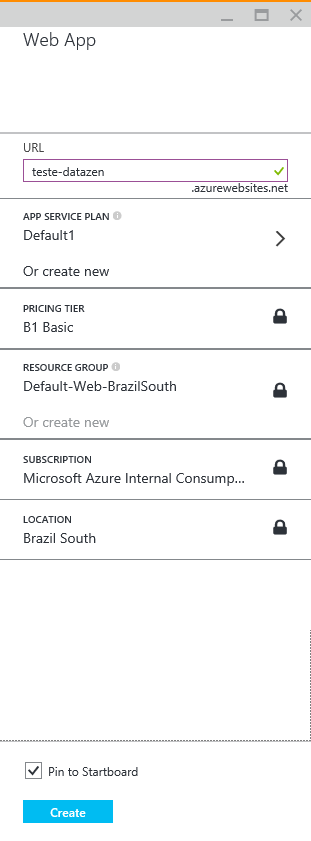
After the Resource Group opens, click on the Add button to create a new resource. This will open a new tile called new resource. Select the option Web App.



After you selected the option Web App a new tile with some information about what is a Web App will show. On this tile there is a button called create, click on it.



After you clicked on the create button, another tile will open where you can configure your Web App. Fill the information needed as Name, Resource Group, App Service Plan, Tier, location and click on the button create.



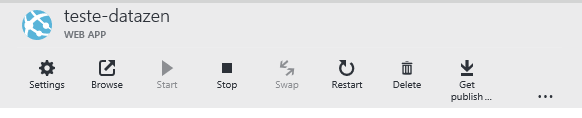
You will need to wait for the Web App to be created before proceeding to the next step.

### Configure DB access on Web App

After you created the Web App, you will need to access its configurations to configure a DB connection with the SQL Azure data base that we created on the previous steps. To do this, access the Web App you just created.



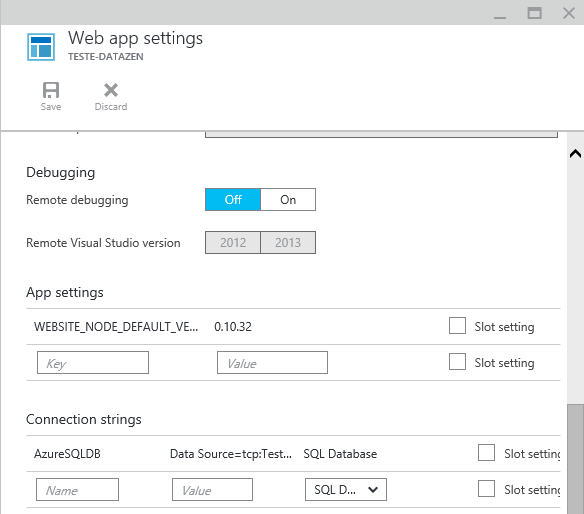
After you click on the Web App, The Web App tile will open, after it opens, click on the Settings button. This will open a new tile showing all the settings.



After the Settings tile opens, select the application settings option. This will open a new tile with all the application settings. Scroll to the Connection Settings section. We will need to create 1 connection with the following information:

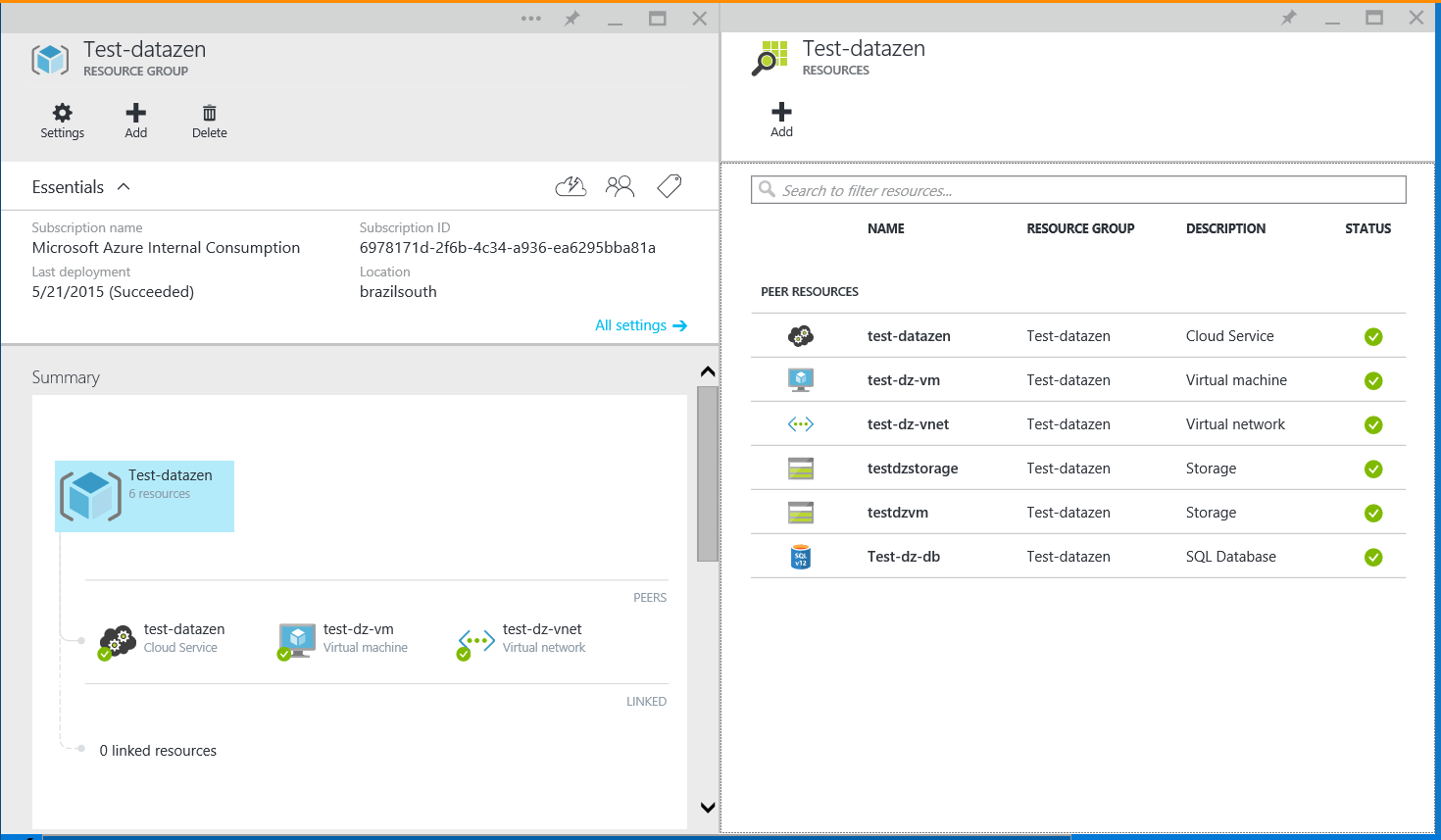
* Name: AzureSQLDB
* Value: Data Source=tcp:Test-dz-db-server.database.windows.net,1433;Initial Catalog=test-dz-db;User ID=mmilhomem@test-dz-db-server;Password=Passw0rd
* Type: SQL database

And then click on the save button.

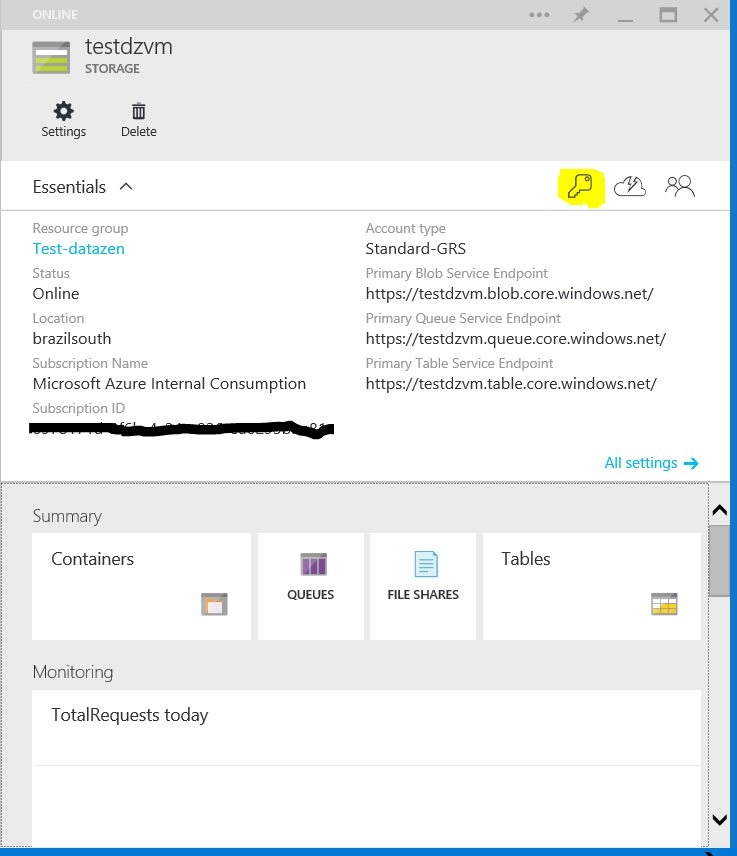
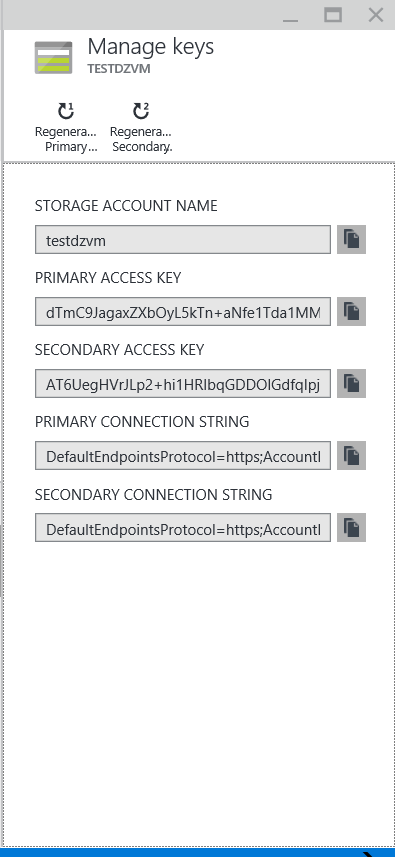


### Create The AzureWebJobsDashboard and AzureWebJobsStorage Connections

The first thing we need to do is to access the storage account that we just created and annotate its name and key. To do this, open the Resource Group Test-datazen and click on the Test-datazen icon under the summary section (highlighted in blue on the screenshot below) . This will open a tile with the list of resources belonging to the resource group. Click on the testdzvm storage account (or the storage account that you are using if you choose a different name).



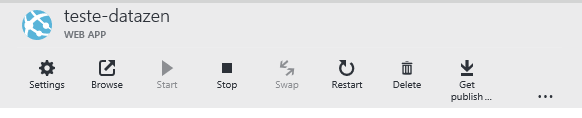
After the Storage tile opens and load completely. Click on the key icon on the top right of the screen (highlighted in yellow on the screen below). This will open a new tile named Manage Keys. Copy the Primary Connection string and also the name of the Storage Account.

Now you will need to access again the configurations of the web app to configure 2 storage connections with the storage account. To do this, access the Web App you just created from the Resource Group.



After you click on the Web App, The Web App tile will open, after it opens, click on the Settings button. This will open a new tile showing all the settings.



After the Settings tile opens, select the application settings option. This will open a new tile with all the application settings. Scroll to the Connection Settings section. We will need to create 2 connections with the following information. Remember to use the name and keys that you copied on the steps above.

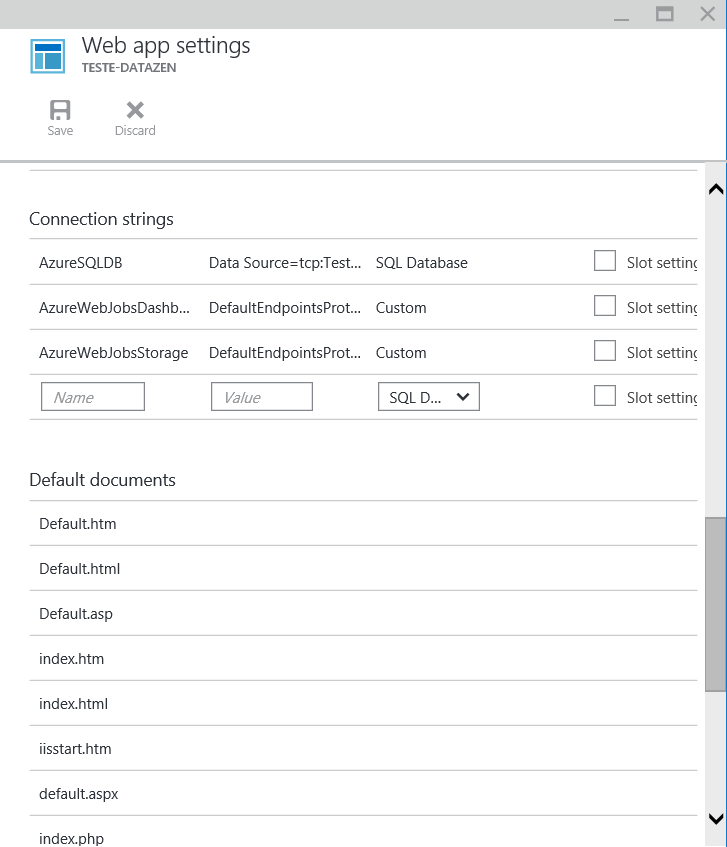
1. One Connection for the Azure Web Job Dashboard

* + Name: AzureWebJobsDashboard
  + Value:DefaultEndpointsProtocol=https;AccountName=testdzstorage;AccountKey=+Xc7HUbjEZski8zB244cARJQSjU5sxxMLdAxGEadluFy4kgqmpsIF6P1DJZIlVscMMNyi69xlv5xEHtQQ6yjgw==
  + Type: Custom

2. Once Connection for the Azure Web Jobs Storage

* + Name: AzureWebJobsDashboard
  + Value:DefaultEndpointsProtocol=https;AccountName=testdzstorage;AccountKey=+Xc7HUbjEZski8zB244cARJQSjU5sxxMLdAxGEadluFy4kgqmpsIF6P1DJZIlVscMMNyi69xlv5xEHtQQ6yjgw==
  + Type: Custom

And then click on the save button.



### Create the Web Job

The web job will be the “engine” that will daily pull the date from Azure billing APIs and store it in our Azure SQL so that the Datazen server can consume and create dashboards.

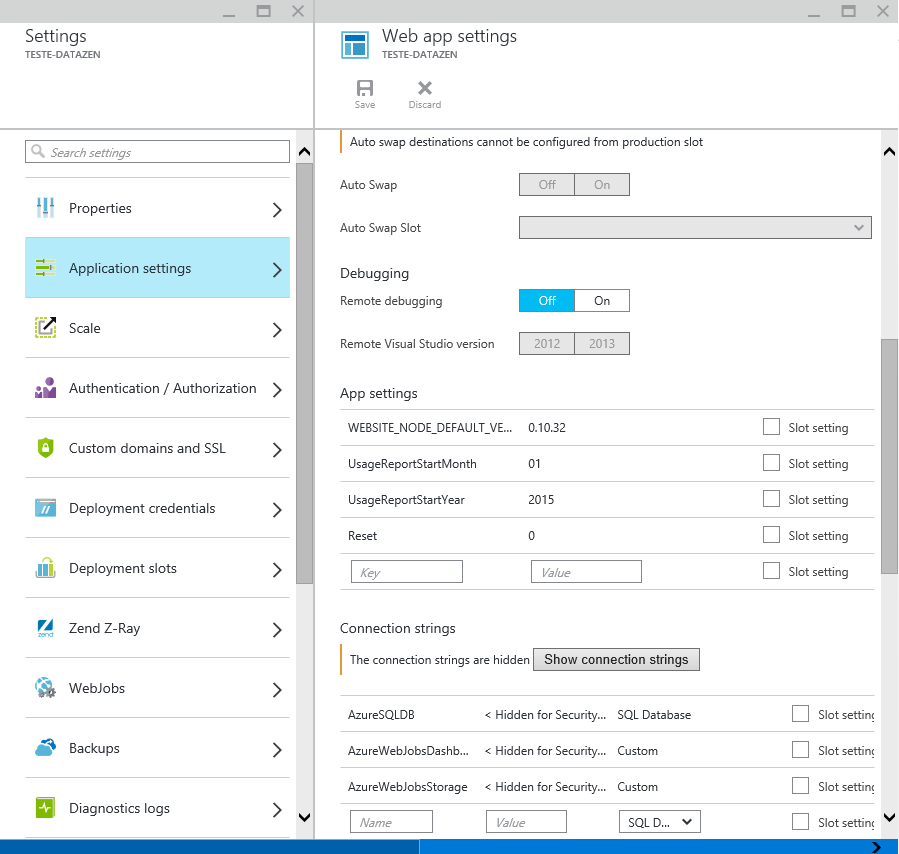
Before we can create the actual web job there are 3 application parameters that we have to add to our web app. This parameters are:

UsageReportStartMonth -> This parameter will tell the web job to start collecting data from this month on.

UsageReportStartYear - > This parameter will tell the web job to start collecting data from this year on.

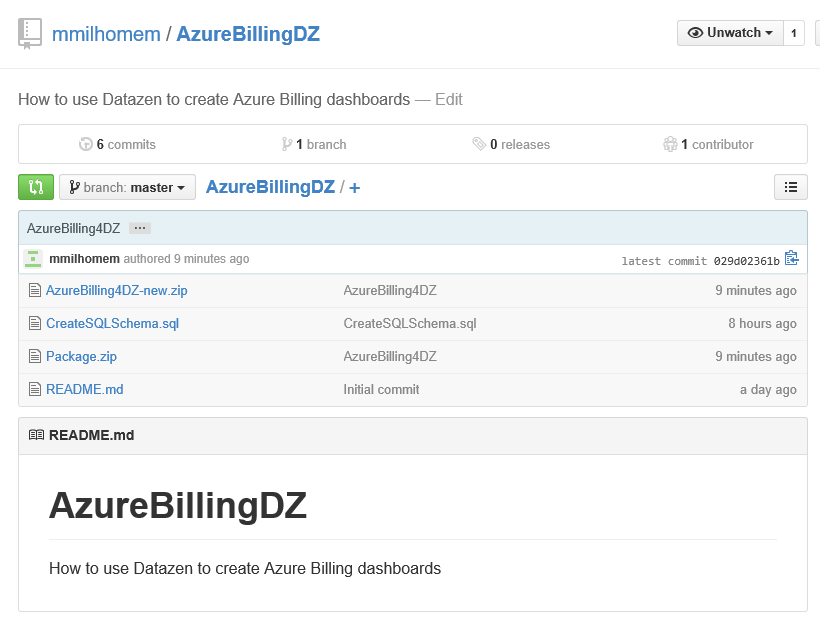
Reset -> If set to 1, this parameter will tell the web job not to do an incremental pull since the last time it ran of the Azure billing API and instead of this, will pull everything again starting from the values inserted on the UsageReportStartYear and UsageReportStartMonth parameters.

To do this we will open once more the settings of our web app (If you are in doubts on how to perform this step, please see the 2 chapters above) and select the option application settings. Once the Tile Web App settings opens, scroll until the App settings section and insert the following parameters and its values on it. After you finish, do not forget to press save.

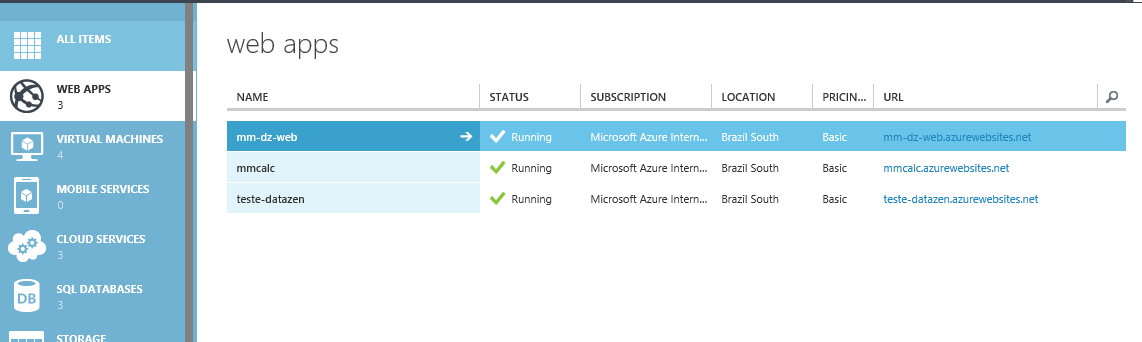


Now we are going to create a Web Job. To do so the first thing you need to do is get the the software from Git hub at <https://github.com/mmilhomem/AzureBillingDZ>. Download a file called Package.zip and save it on your desktop. This is a compiled version of the web job ready to be deployed into Azure.

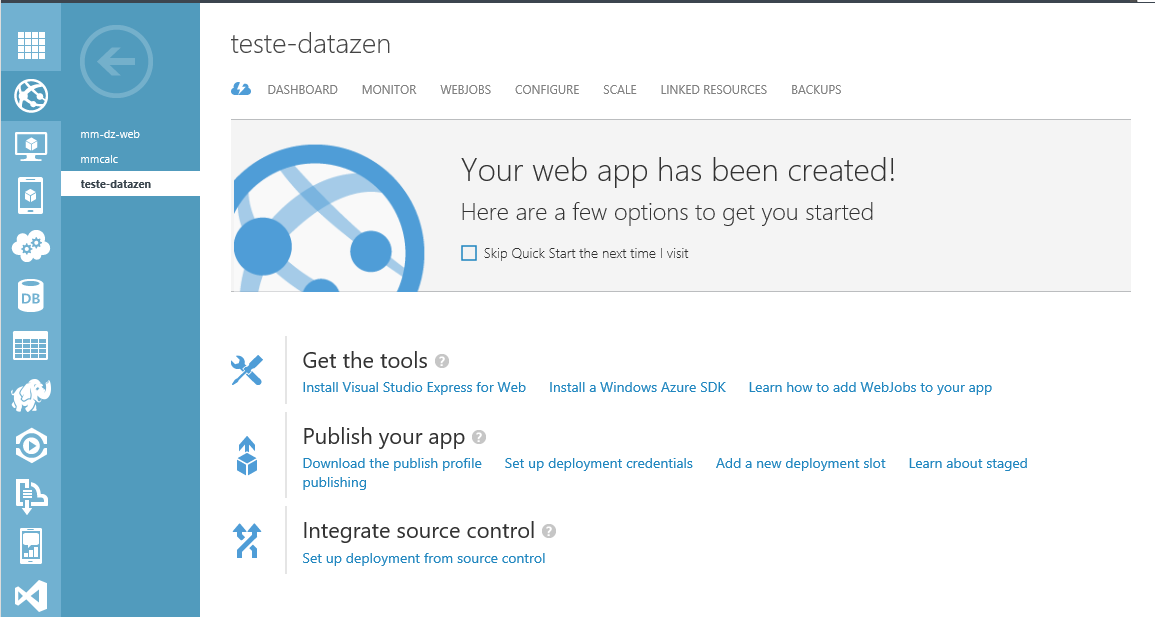
Optionally (this steps will not be covered in this guide) you can also download the source code from the same address, compile it and deploy yourself. The source code is also at <https://github.com/mmilhomem/AzureBillingDZ> under the name of AzureBilling4DZ.zip.



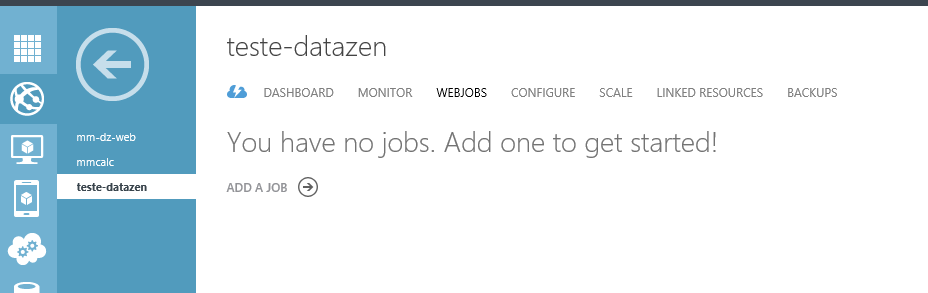
To do this last step on the web job creation, we will need to go to the old Azure portal as we need to create a scheduled web job and this feature is not available on the new portal yet. To do this, open a browser go to <https://manage.windowsazure.com> and login to the same subscription you are using. After you have logged in, click on the Web Apps icon on the left toolbar.



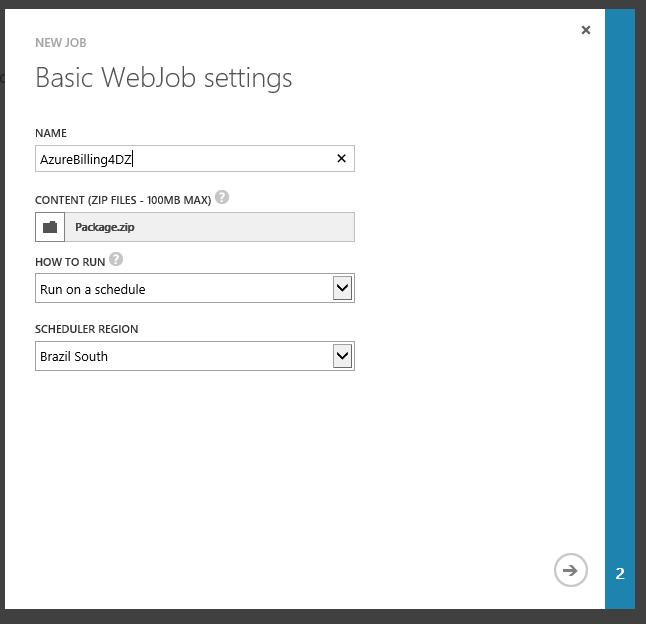
Once your web Apps appear on the right of the page, select the web app that we created earlier on the new portal.



Once the Teste-datazen opens, click on the web job menu option.



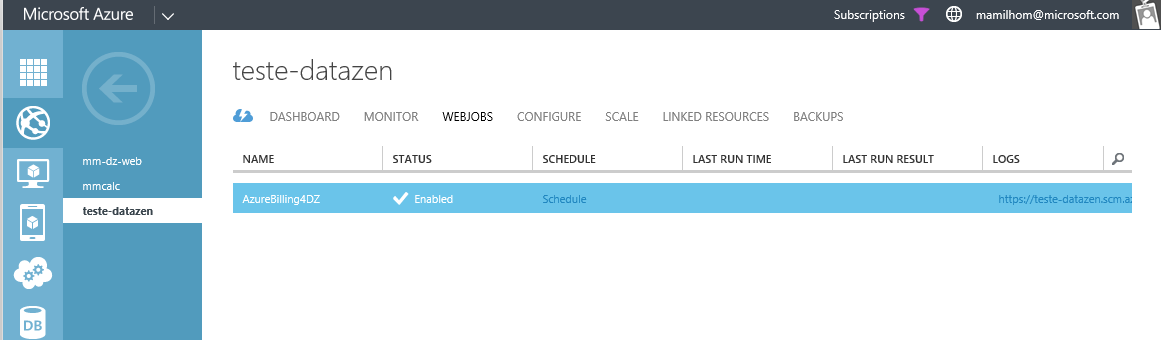
You will find on the middle of the page a link to add a new web job. Click on it and a dialog will open. On this dialog you will fill a name to the web job, select the package.zip file that you just downloaded on the earlier chapter and select to run on a schedule. After you have filled all the information needed, click on the right arrow on the bottom right of the dialog.



Now you will mark the web job as a recurring job, select it to run once a day, choose a start and end schedule. After you have filled all the information needed, click on the check mark on the bottom right of the dialog



Before proceeding to the next step you will have to wait for the web job to run and have it´s status marked as enabled



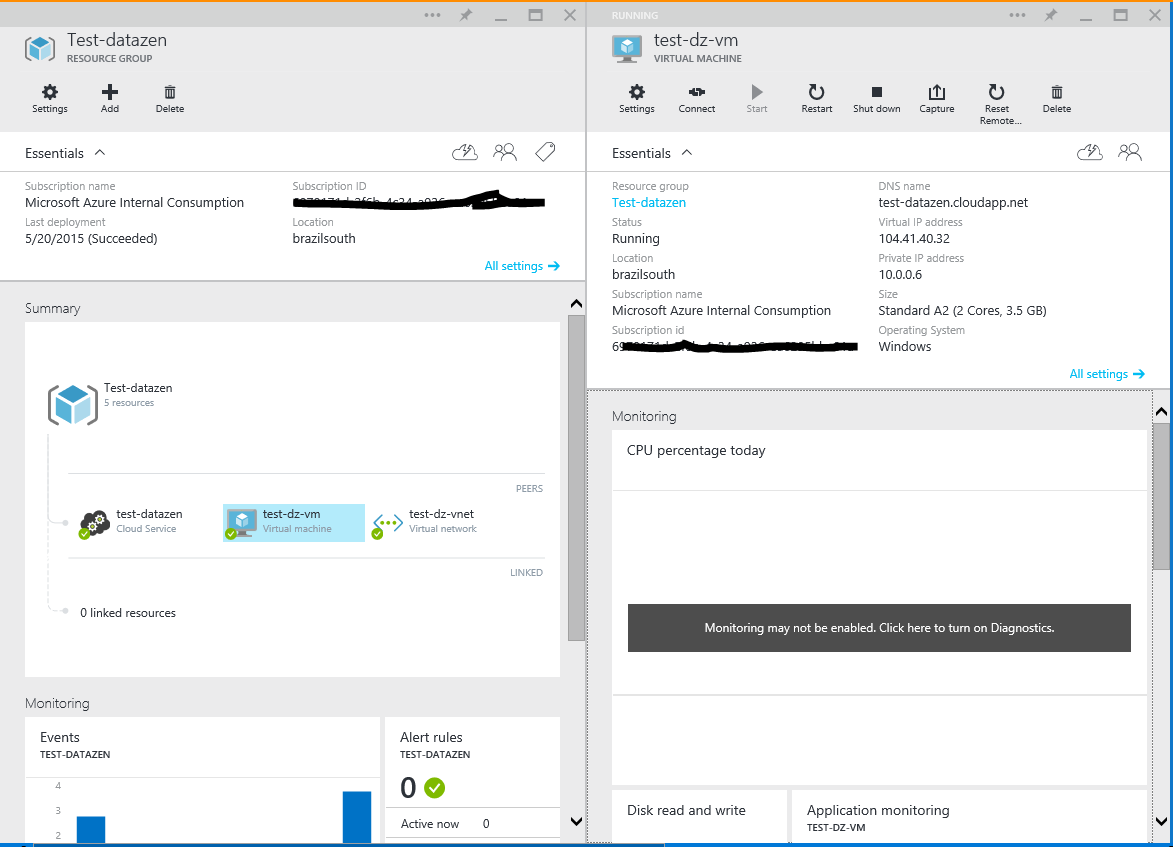
# Installing and Configuring Datazen

By now you should already have everything you need to install and configure the Datazen server. This chapter of the guide we will go through this process. Please remember that everything you did so far was to be able to pull data from Azure billing API and pump it into a Azure SQL. Now we will install the Datazen server into our VM and configure it to pull data from the Azure SQL data base that is already populated.

## Installing Datazen.

The first thing we need to do is to remote desktop into our VM and install the datazen server.

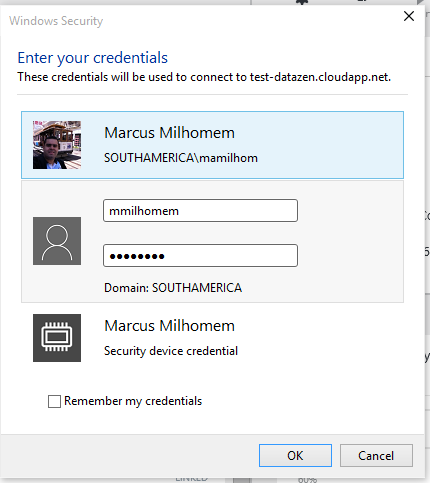
To remote desktop go to the Resource Group on the Azure Portal, click on the test-dz-vm and select connect from the top menu.



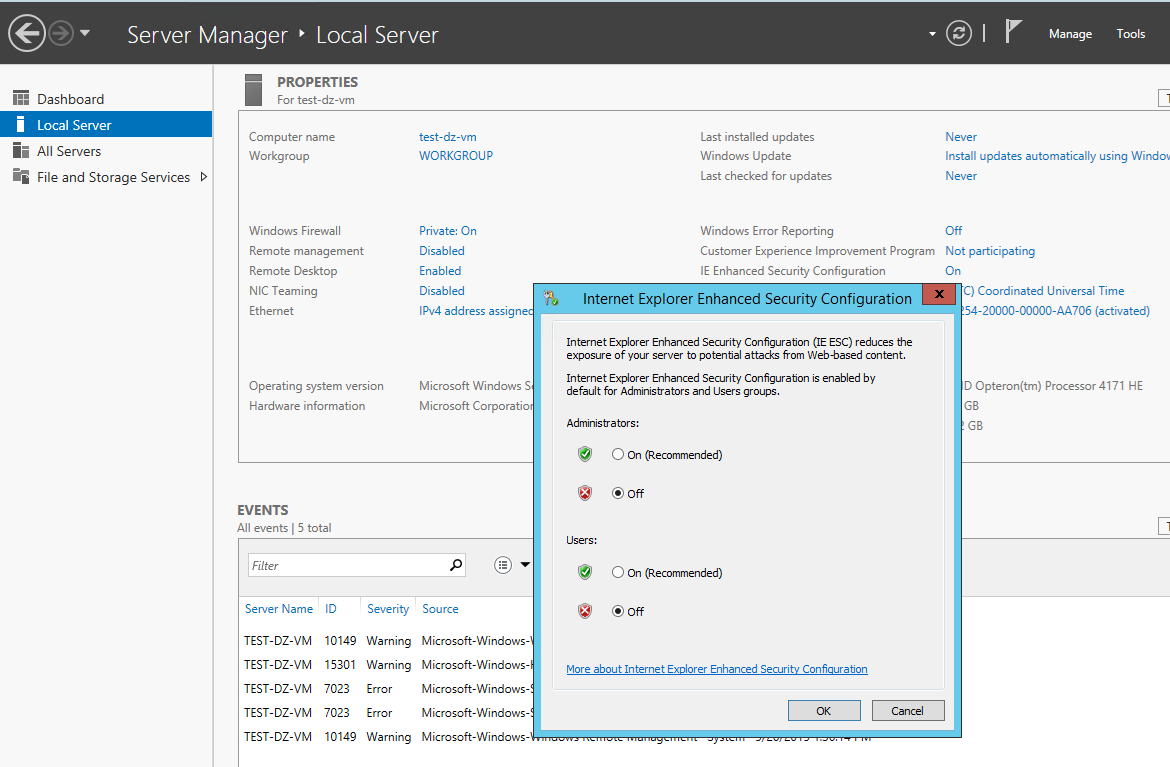
Once you click connect select to open the rdp file on your browser.



When to login screen appears, fill with the information that you entered on the creation of the VM and hit ok



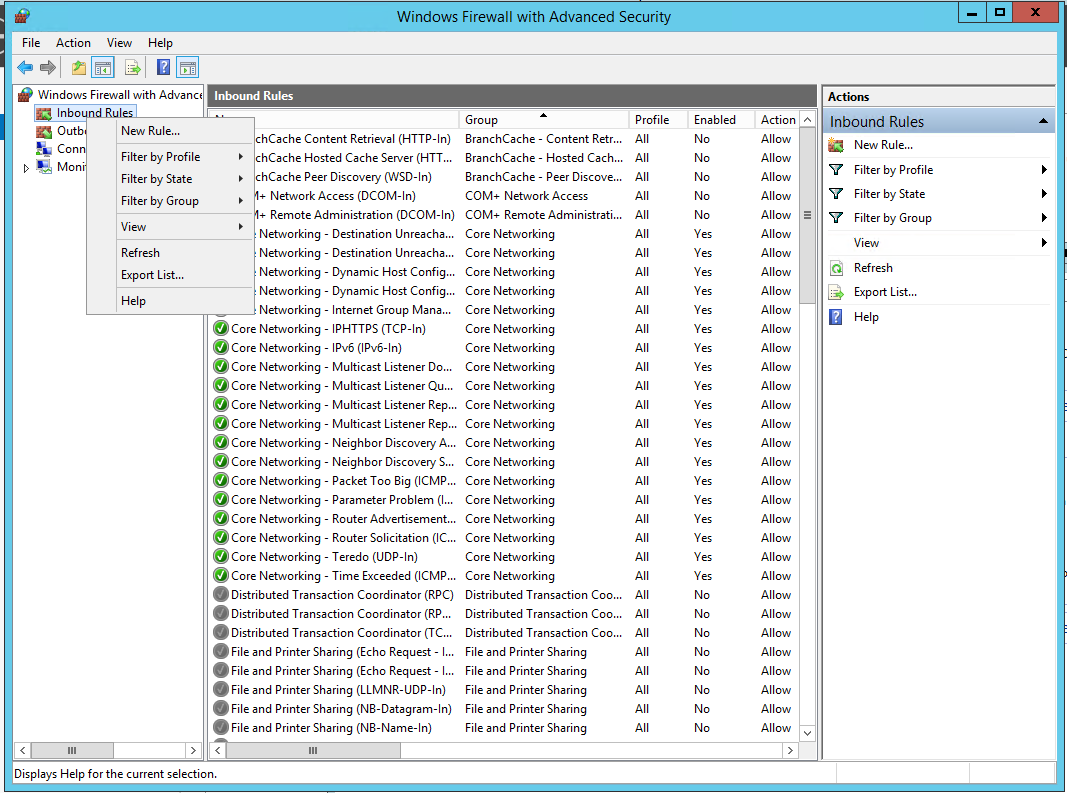
When The remote desktop session appears, the first thing you need to do is to disable the IE Security browser and create a firewall rule opening the inbound traffic for port 8080. To do so, as soon as the rdp session opens, click on the Local server option and then on the IE Enhanced Security Configuration link. After that a new screen will show. Change everything to off and click the ok button.



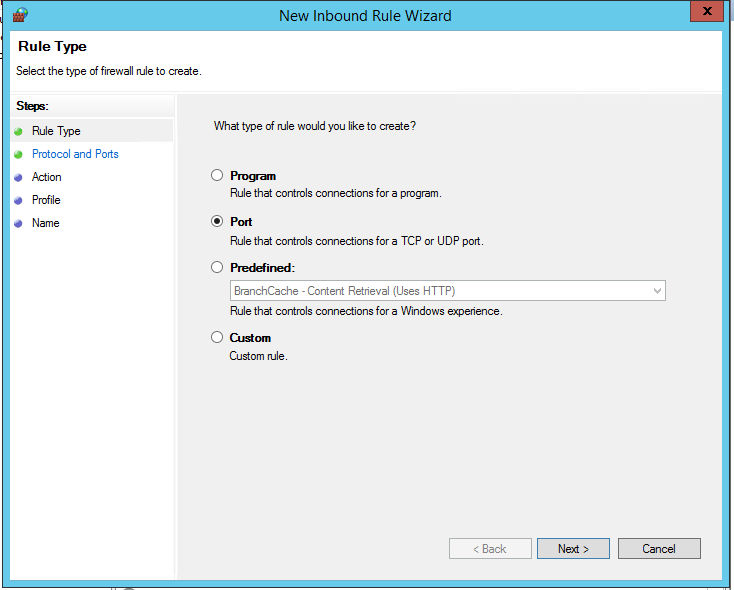
After you return to the Server Manager screen, click on the Windows Firewall link. A new screen will appear. Click on the advanced settings link on the right of the screen.



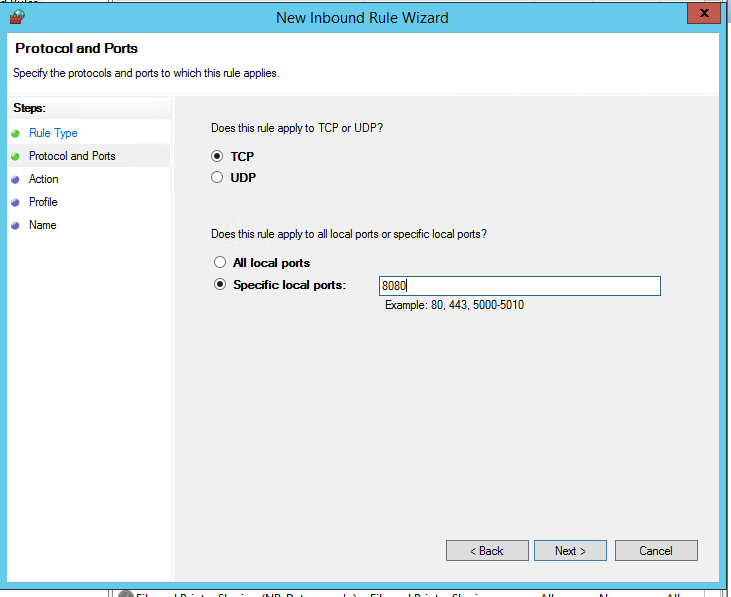
A new windows will appear. Right click on the Inbound Rules and select the New Rule option. We will create a rule to allow inbound traffic to the port 8080.



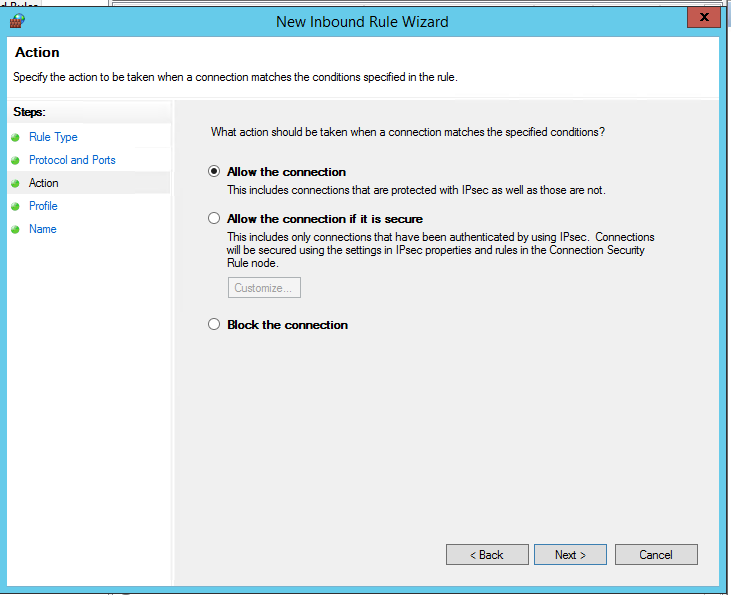
Now select the option named Port and click next.



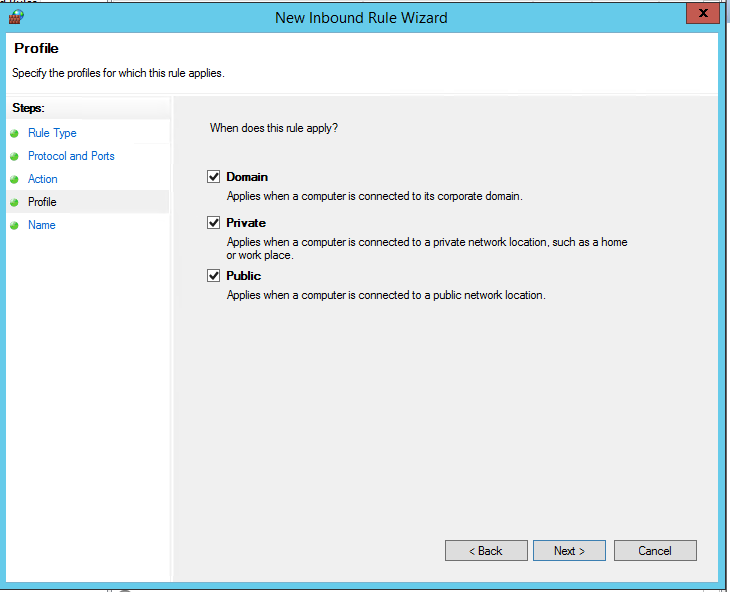
Fill in the desired port and do not forget to select TCP as the protocol. When don, click the next button.



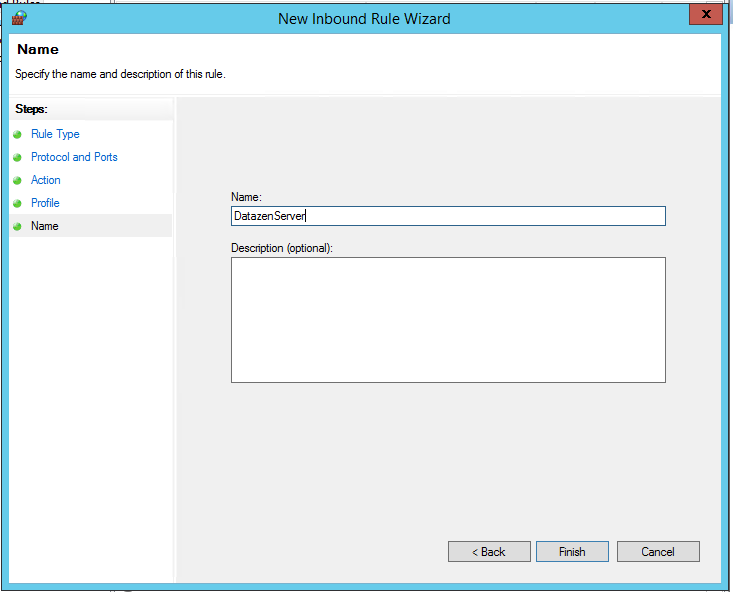
Now we will leave the option allow the connection and click next once more.



Once again we will not change anything and click next.



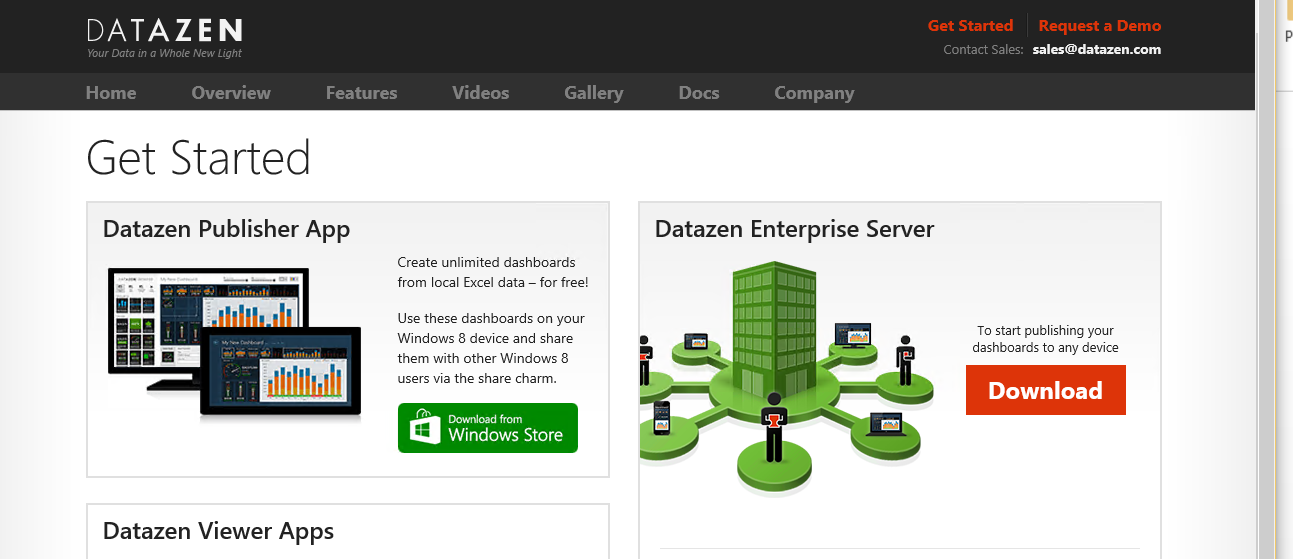
Now we will give the rule a name and click finish.



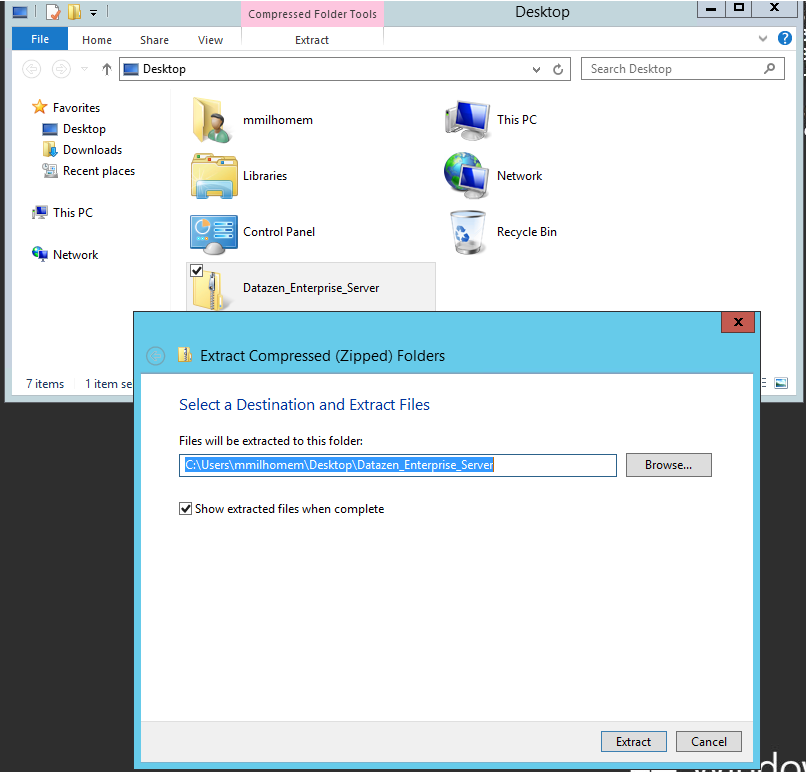
The next thing we need to do is open the IE and download the datazen installer from the site <http://www.datazen.com/>. To do this, open the IE, go the datazen web site and click on Get started on the top right of the page.



This will lead you to the download page where you will find a download button. Click on it and save the datazen file.



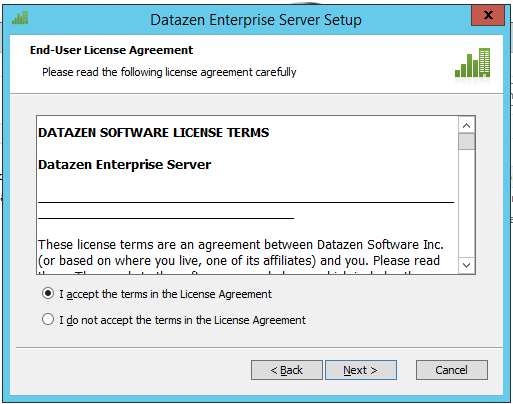
Once the file finishes to download, unzip it and run the executable named Datazen.Enterprise.Server.3.0.2962.



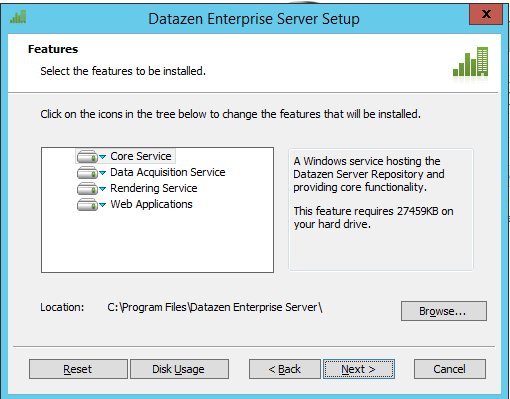
Once the Datazen installer shows up. Click next.



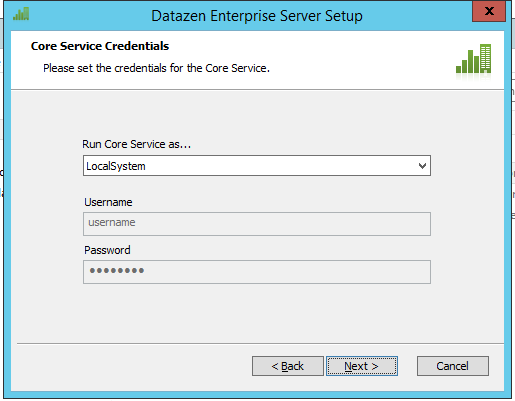
Accept the terms and click next.



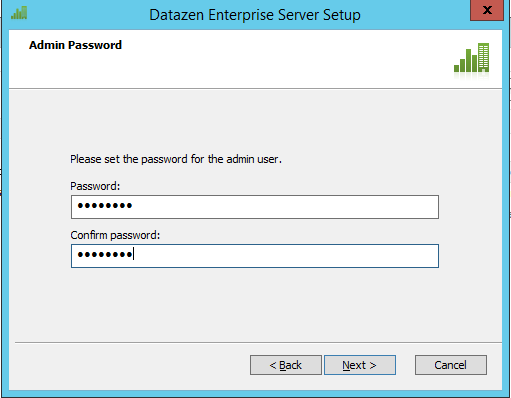
In this example we will install everything into one server. Do not make any changes and click next.



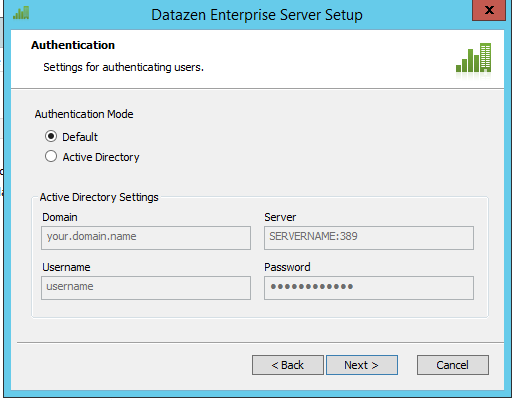
On the Run core screen, again just click next.



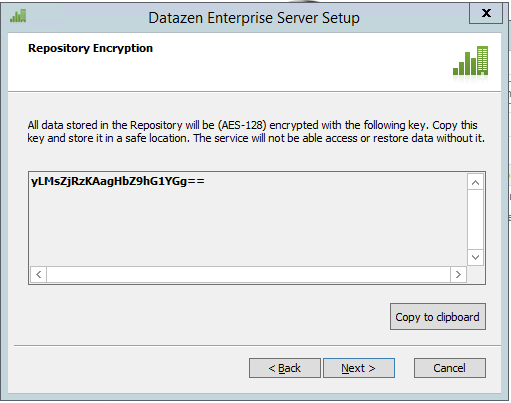
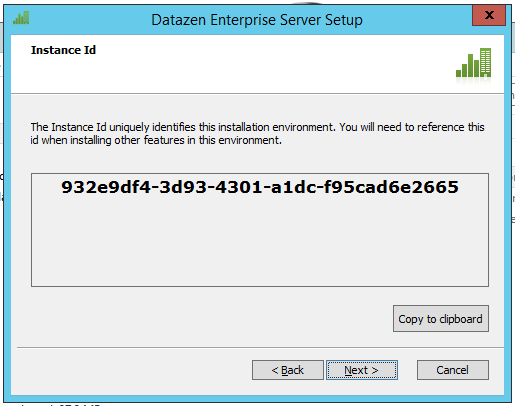
On the next screen, fill in the admin password. You will need this on your first login into the Datazen Config Portal.



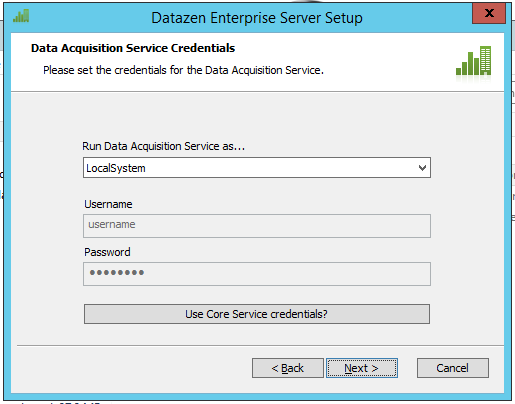
Leave the authentication mode as default and click next.



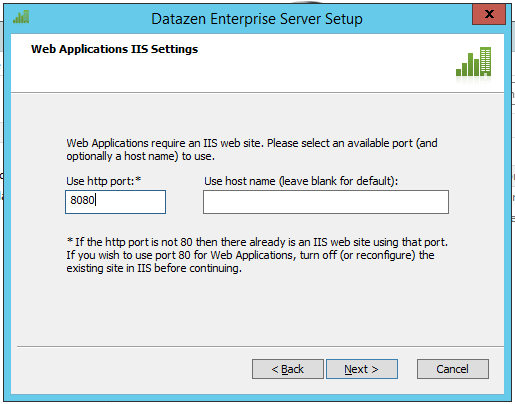
On the next two pages, save the 2 keys on a safe place and click next (you will do this two times). This keys are needed in case you lose access to the system.

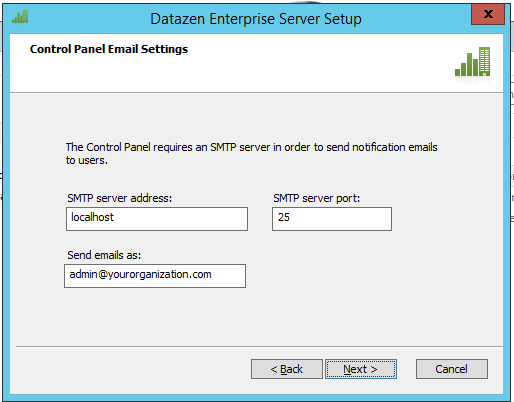
After saving the keys and clicking next, leave the te server to run as local system and click next.



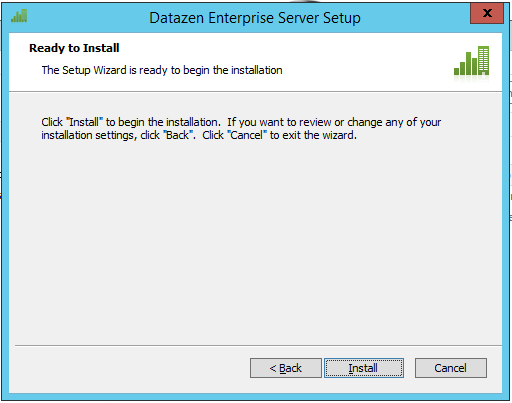
On the next screen, change the port to use the one that you opened on the firewall and endpoint. In our case 8080 and click next.



On the email settings, as we are not configuring any email server, just leave as it is and click next.

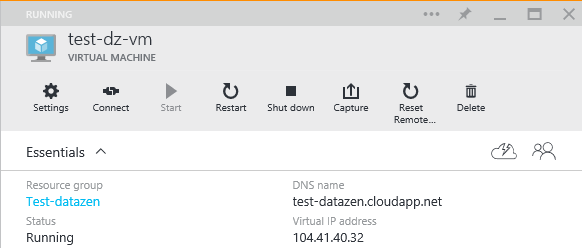


Now finally, click install and wait for it to finish.

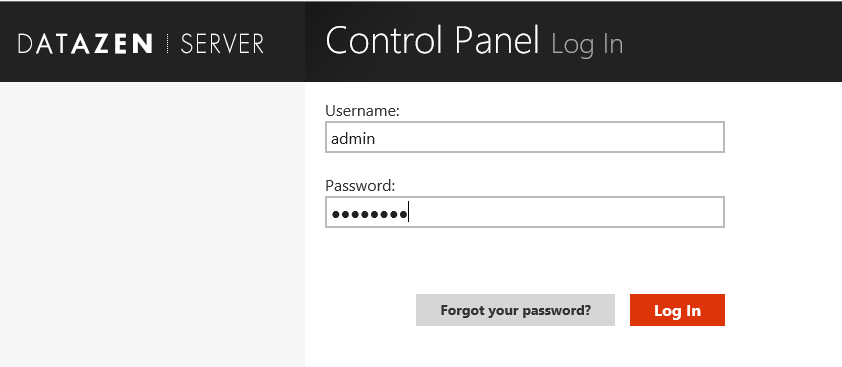


## Configuring Datazen Access

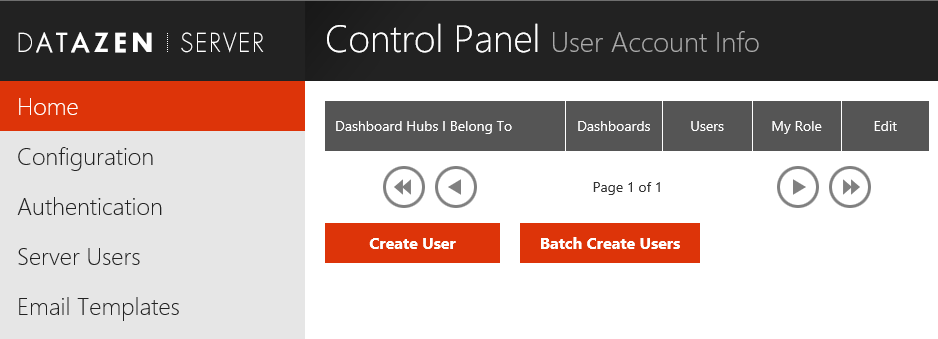
After you installed the Datazen server you need to configure the access to it. To do so, open a browser and go to the following url: <http://test-datazen.cloudapp.net/cp/admin>. You can verify if this is the correct URL by going on your Virtual Machine on the Azure Portal and checking the DNS name.



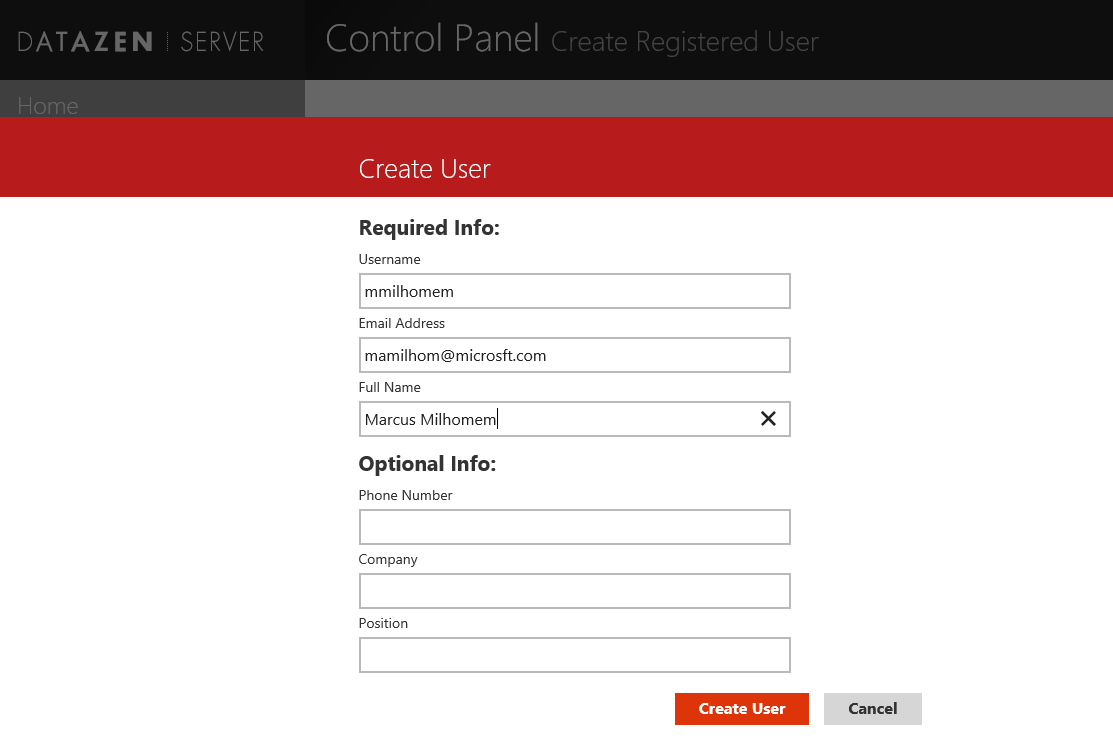
After you opened the browser and accessed the url, enter the admin as the user and the password that you filled during the installation and click login.



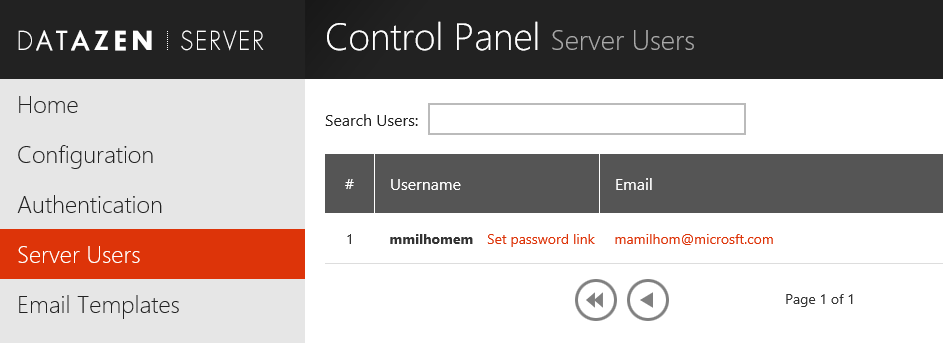
After you login, the first thing you need to do is to create a new user. This will be the user that we are going to use to configure dashboards. Click on the Create User button.



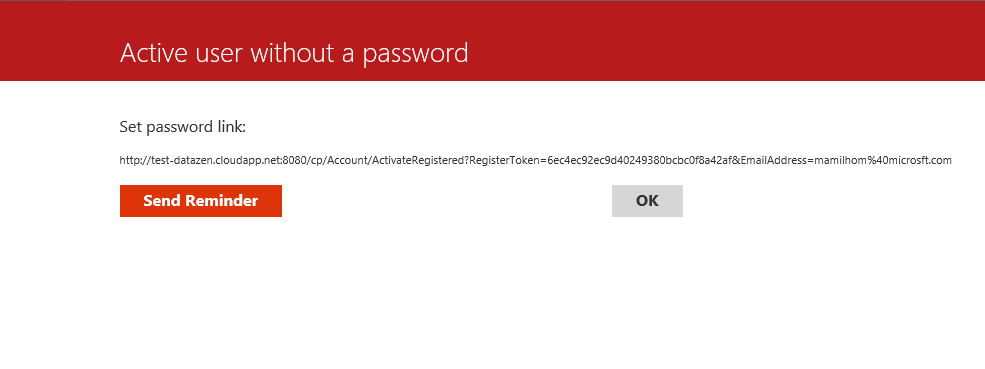
Fill in the user information and click on create user.



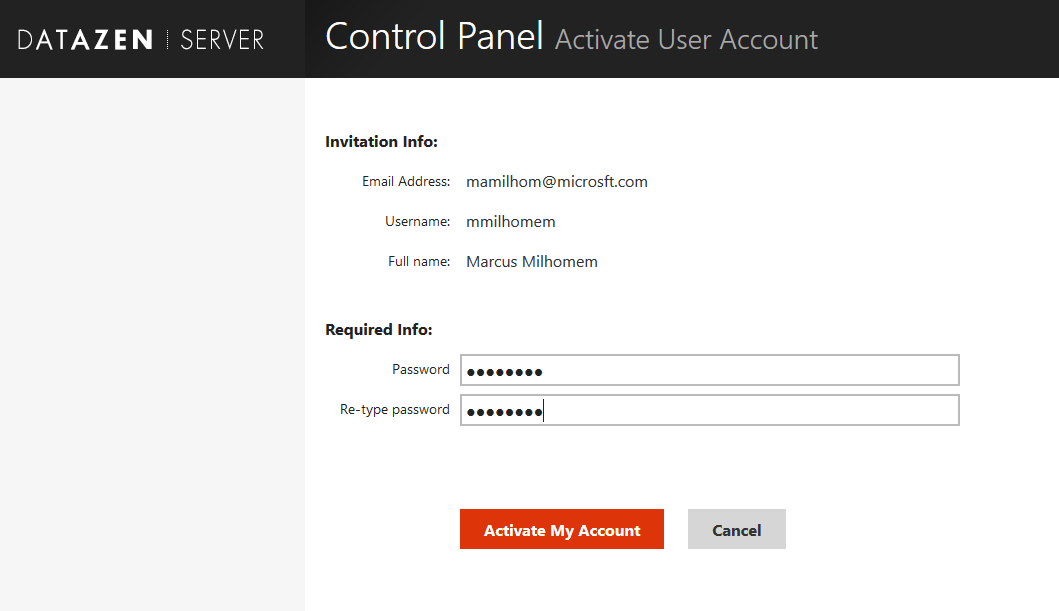
After you receive the message that the user was added, click on the done button. After you return to the home page, click in server Users and on the Set password link of the user you just created.



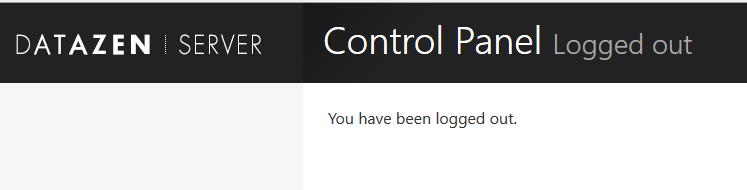
Copy the URL that appears and paste it on a browser. If you are accessing this url from the internet, do not forget to adjust the url ports (in our example change it from 8080 to 80) due to our NAT configuration of the endpoint.



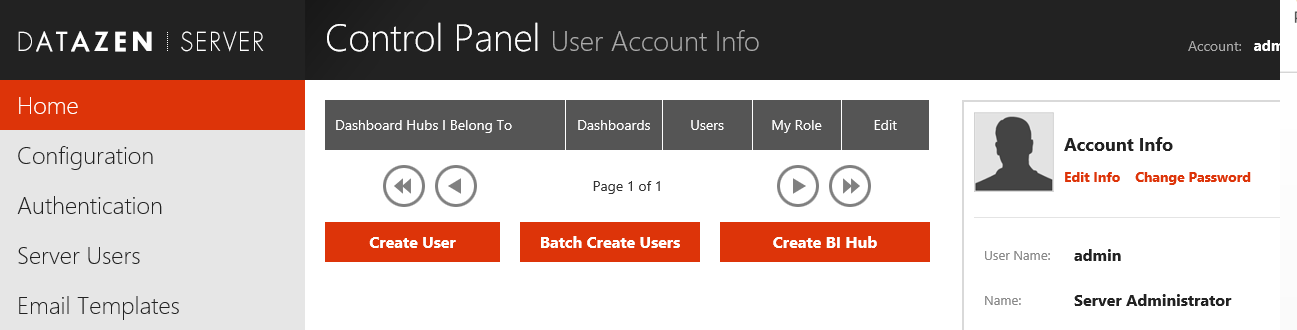
Please fill the password and click Activate My Account. Sometimes, if takes too long from de generation of the URL until you actually uses, you may need to re generate the link.



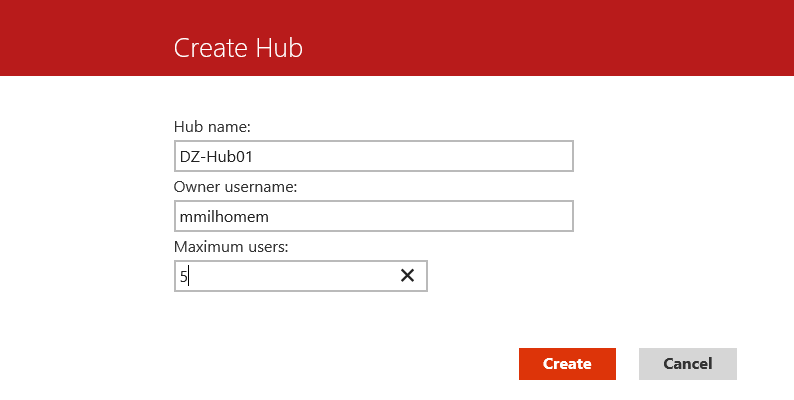
If everything worked, you will receive the following screen



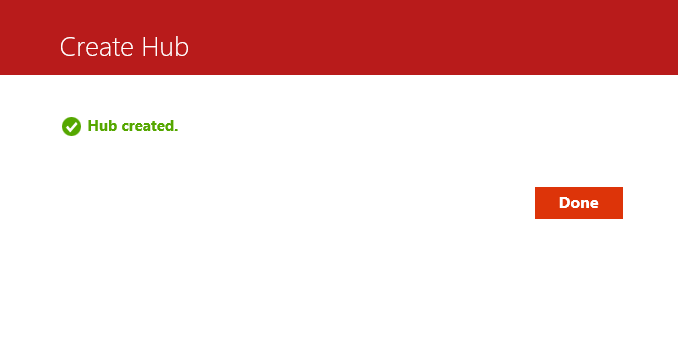
Return to the admin console, still logged as admin and on the home screen select the option create BI Hub.



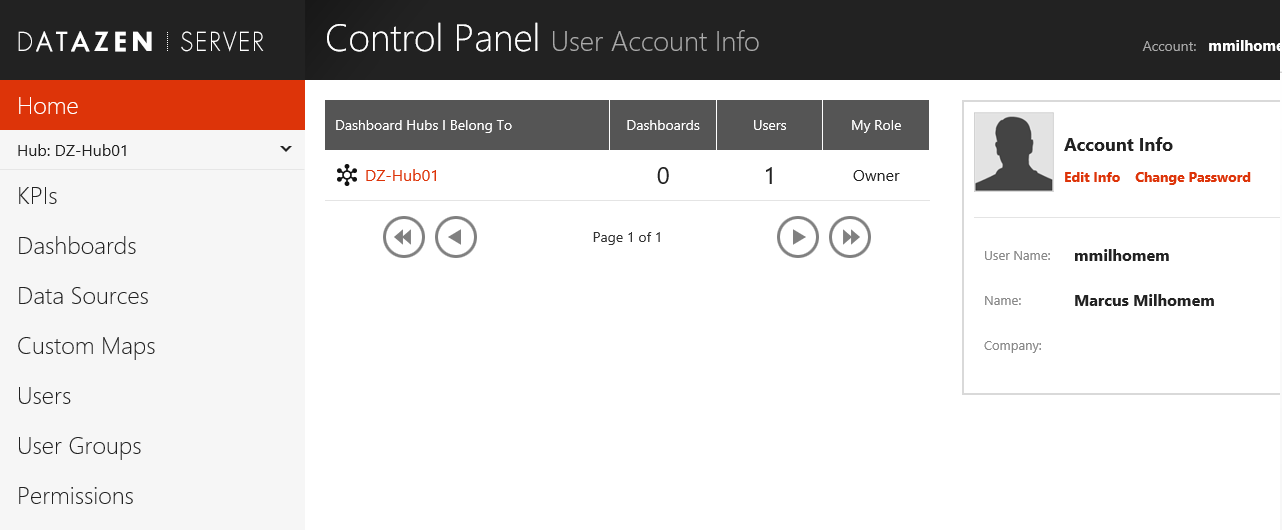
On the Create hub screen, give the hub a name, select the user we just created as the owner and select the maximum amount of concurrent users.



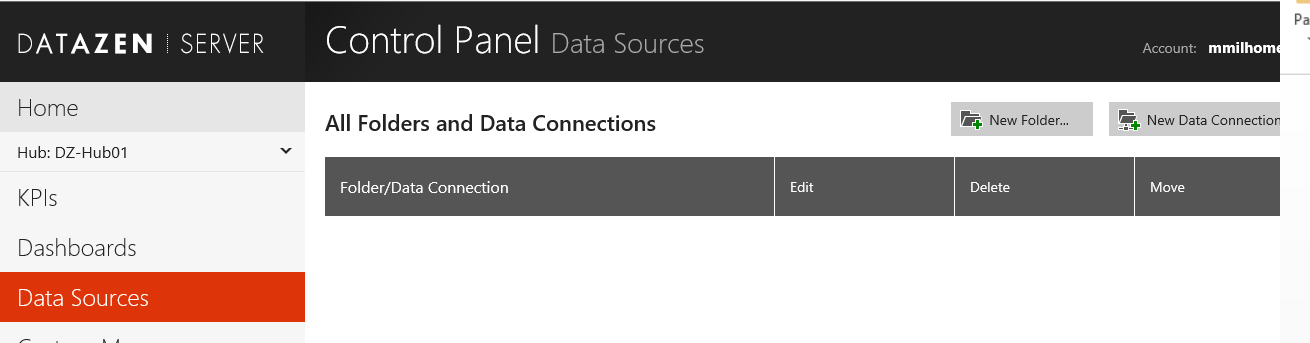
After you received the message saying that the hub was created, click done.



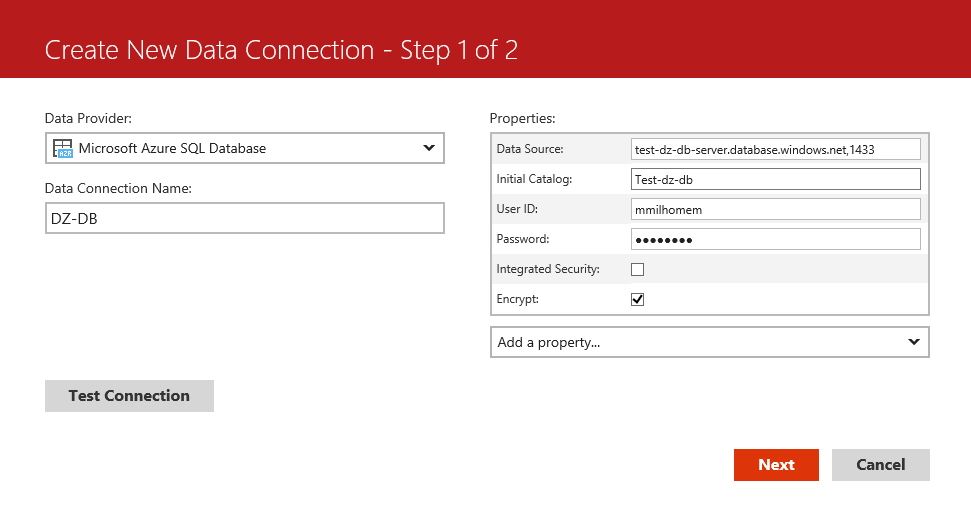
Now you need to logout and log back in with the user we just created and assigned as Hub owner. Once you do that, you will see a dashboard that looks like this.



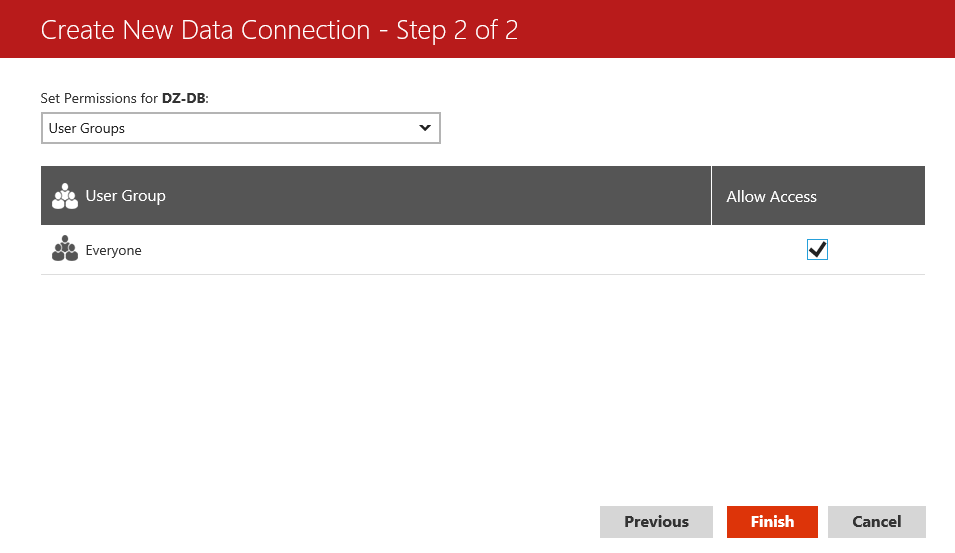
The Next step is to click on the Data Sources link on your left menu. Once you do that, you will see a page with the option to add a New Data Collection. Click on it.



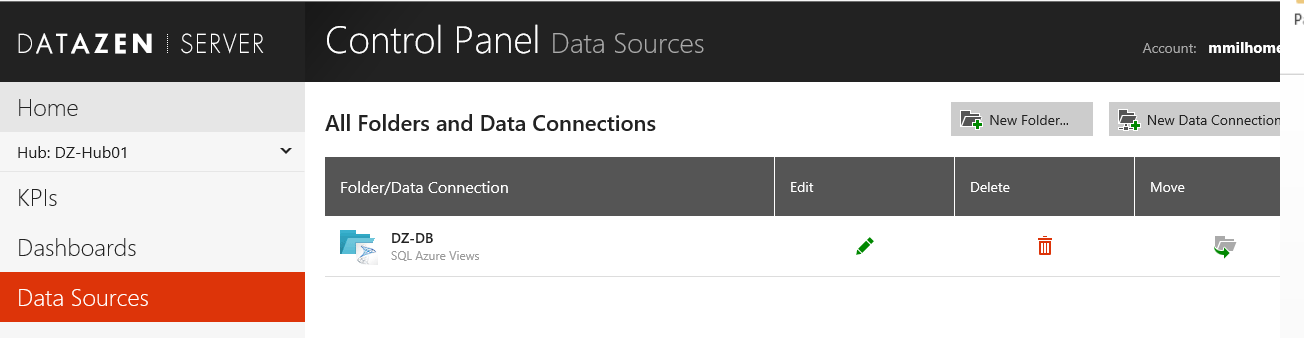
After you clicked on the New Data Collection button, the following screen will show. You will need to fill it up with the information of the Azure SQL data base that we created. If you do not remember, you can go back to the Azure Portal, select the database and hit the link show connection. Remember to change the data provider to Microsoft Azure SQL Database. After you fill the information up, click on the test connection button.



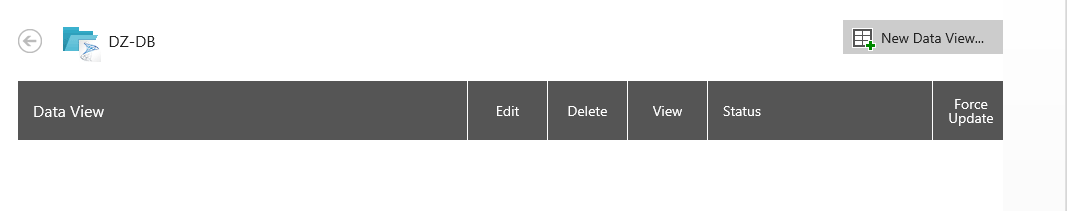
If the connection succeed, press next. To make things easy for this guide we will allow everybody access to this Data source and click finish. You can change this later with you want.



After you click finish, you will return to the screen that shows your newly created data connection. Now we need to click on it (in our example click on DZ-DB).

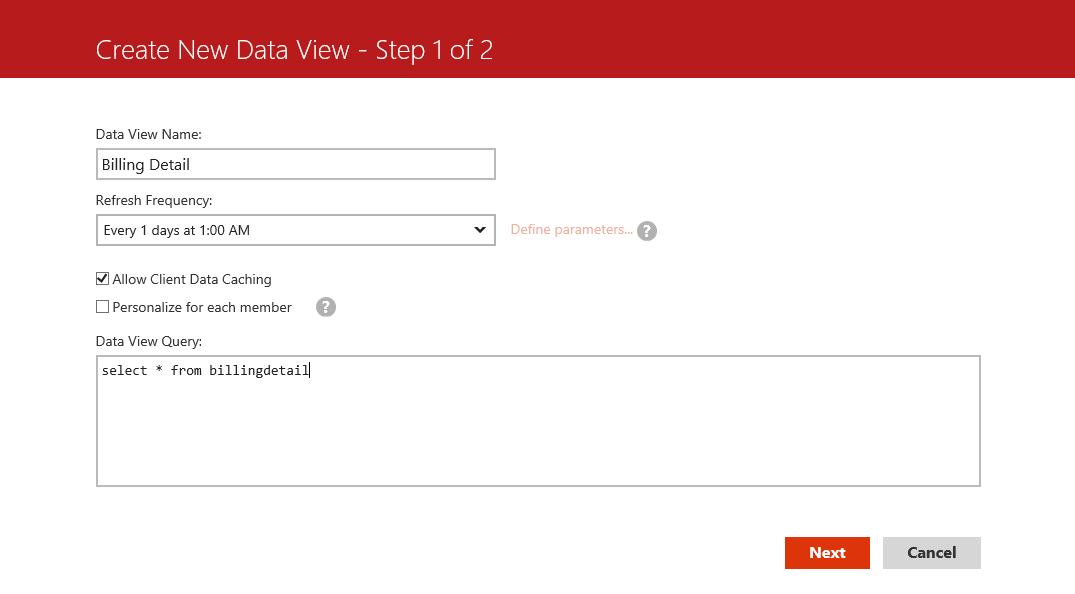


After you click on it a new screen will appear. On this screen we will need to configure a new Data View. We will do this by clicking on the New data view button on the top right of the page.

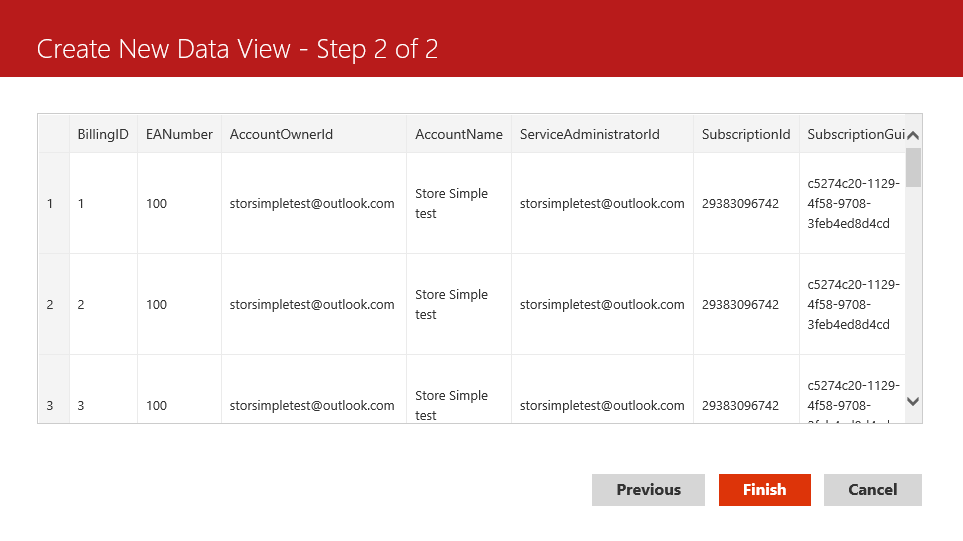


On the Create New Data View screen, fill the information as follows. We will configure Datazen to pull data from the Billing Detail table from our Azure SQL data base. When done, click next.

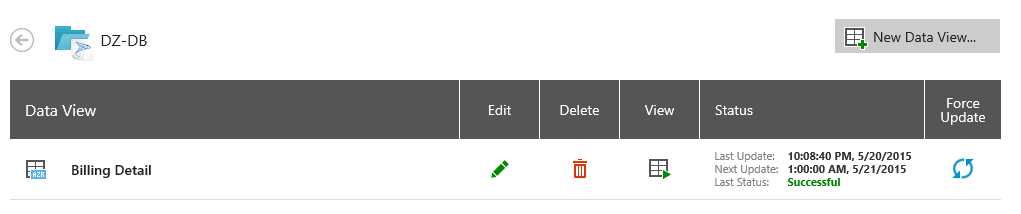
Query: select \* from billingdetail



Datazen will show a sample of the data, click Finish.

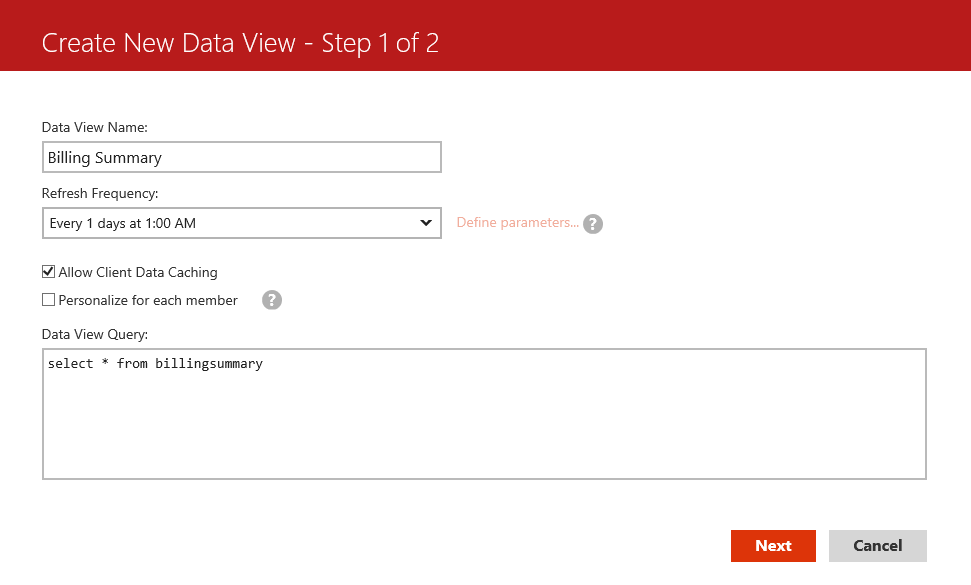


After you clicked finish, you will return to the Data View page. Where We will do the same process to add a Billing Summary Query. To do that, click once again in the New Data View Button.

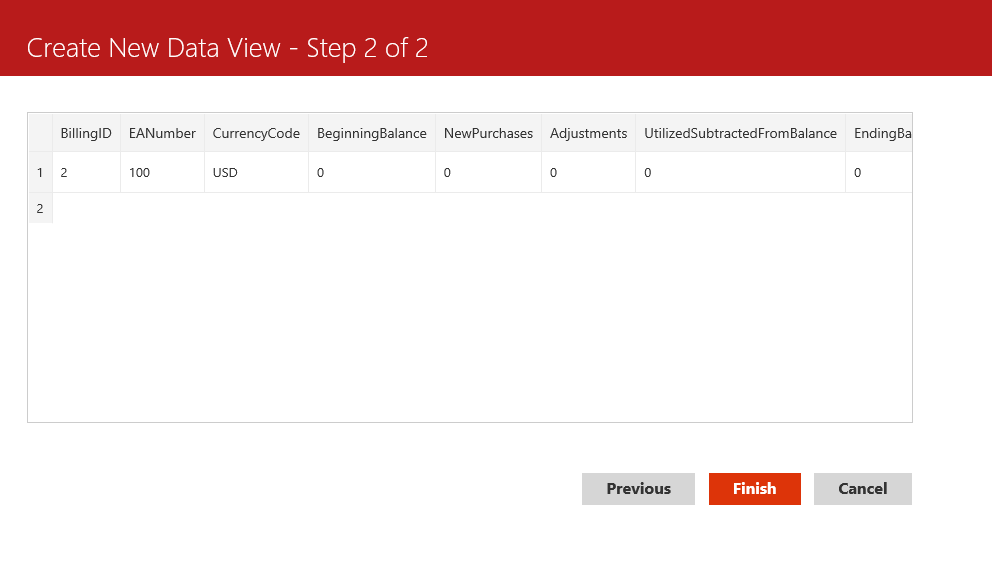


Now we will fill the information of this new Data View with the Billing Summary Query. Once done, click next.

Query: select \* from billingsummary



Once again Datazen will show a sample of the data and you should click finish.



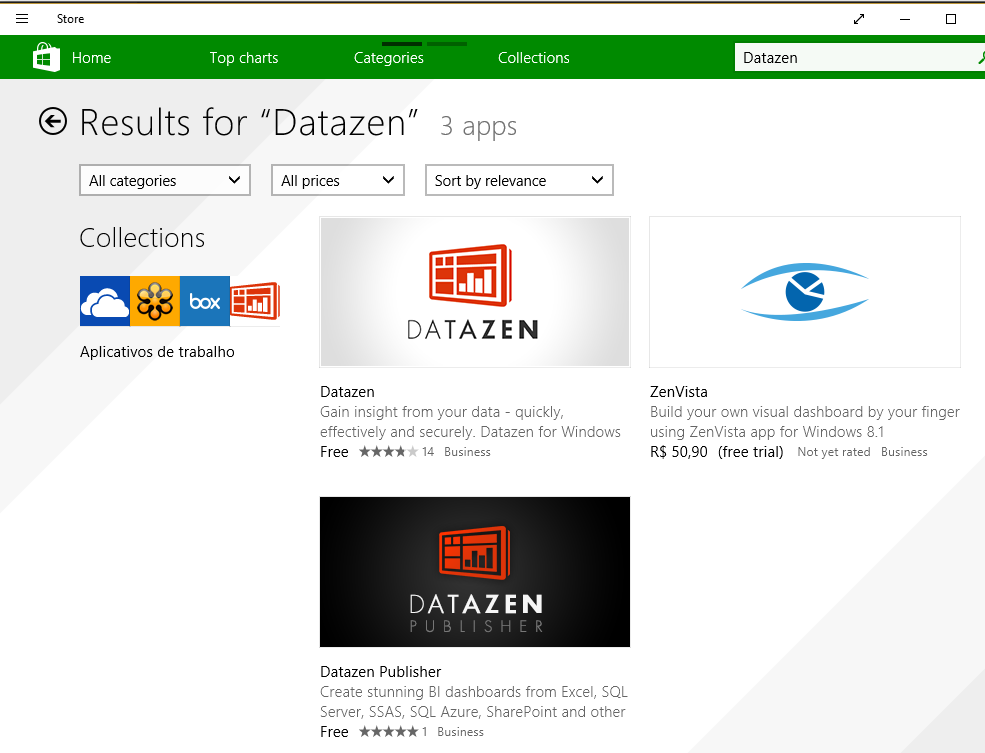
Now the Datazen is ready for us to start building our dashboards.

# Installing and Configuring Datazen Clients

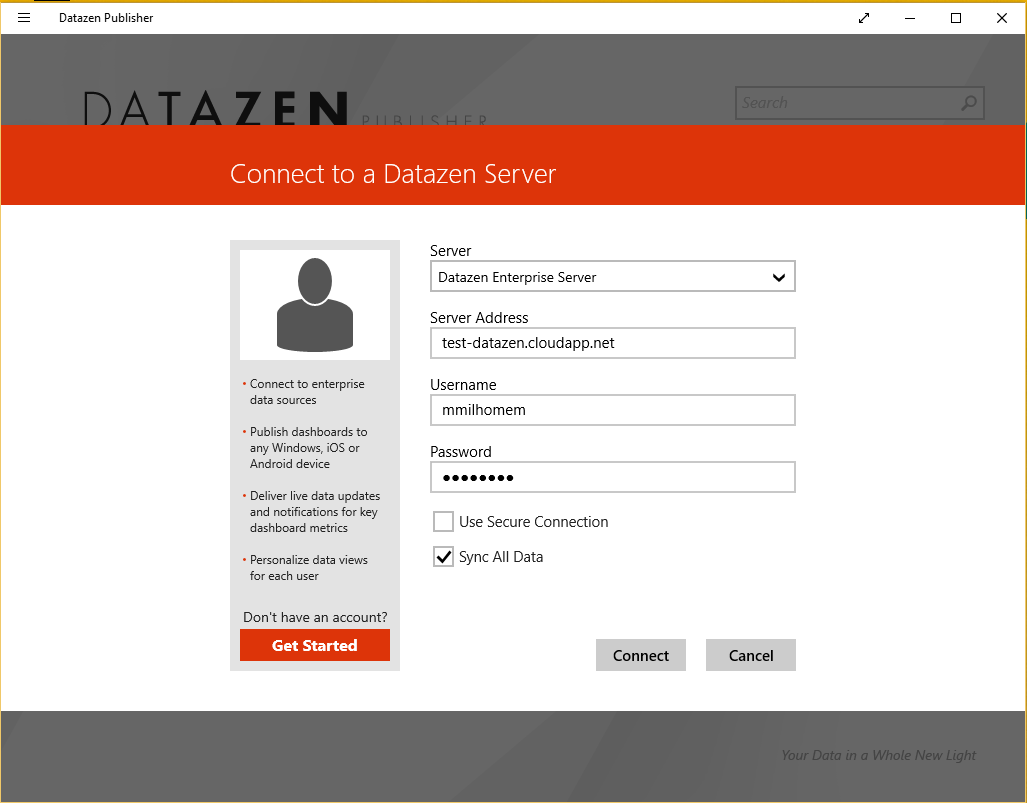
Now that we have all the server parts working, we need to install the clients and start to build the dashboards. One important thing to notice is that we can install the client on phones and tablets as well, including window phone.

## Installing the Clients

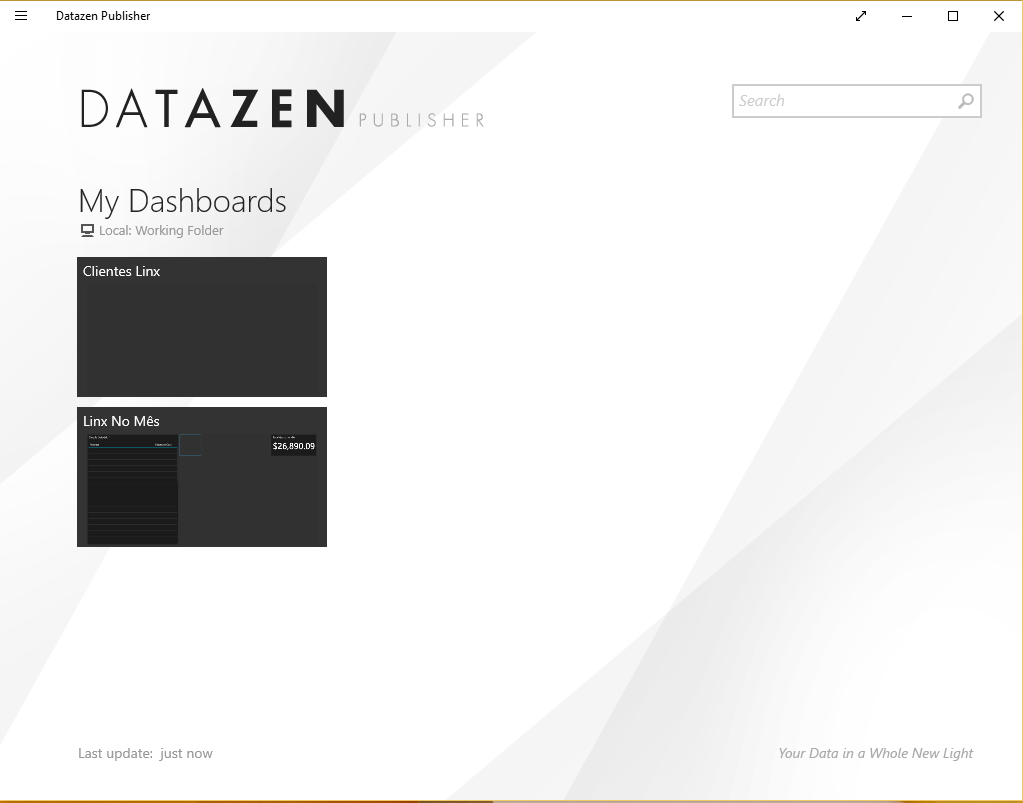
The first thing you need to do is to go to the Microsoft store, search and download the Datazen and Datazen Publisher Apps.



Install both apps into your desktop. After you installed both apps, launch the Datazen publisher app. You will see a screen that looks like the following. Fill the information needed to connect to your Datazen server. Do not forget to uncheck the Use Security Connection option as we are not using it for our example.

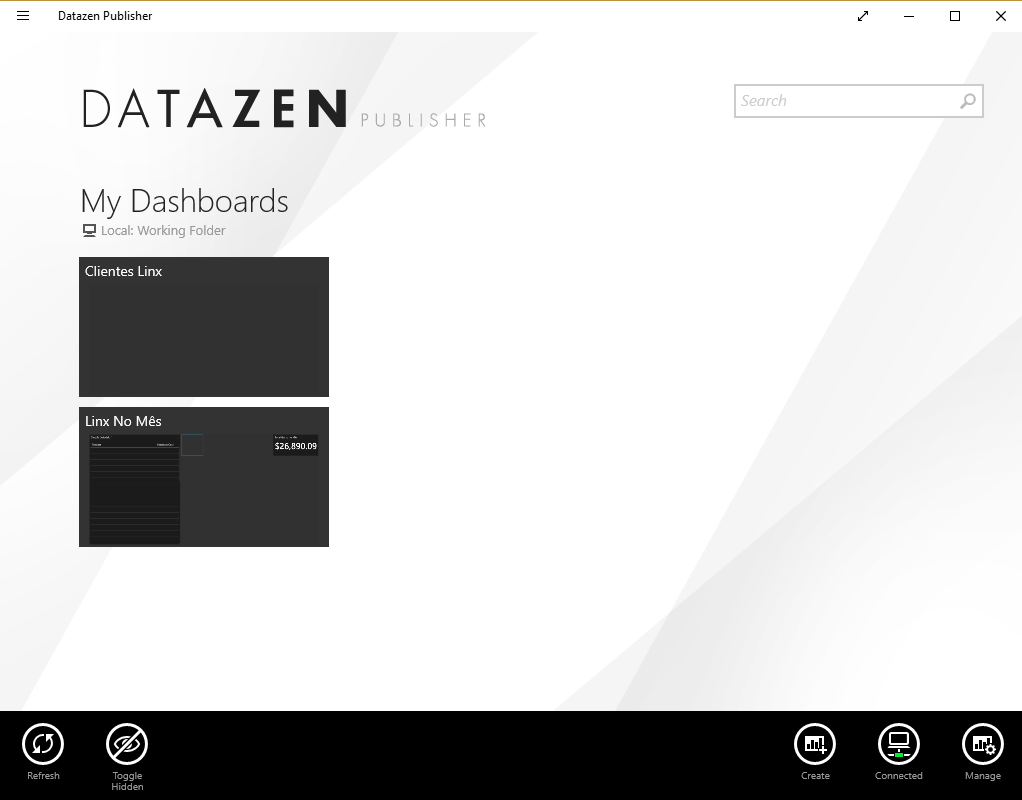


After you login, you will see a console that looks like the following. You probably will not have any dashboard. I already have 2 created.

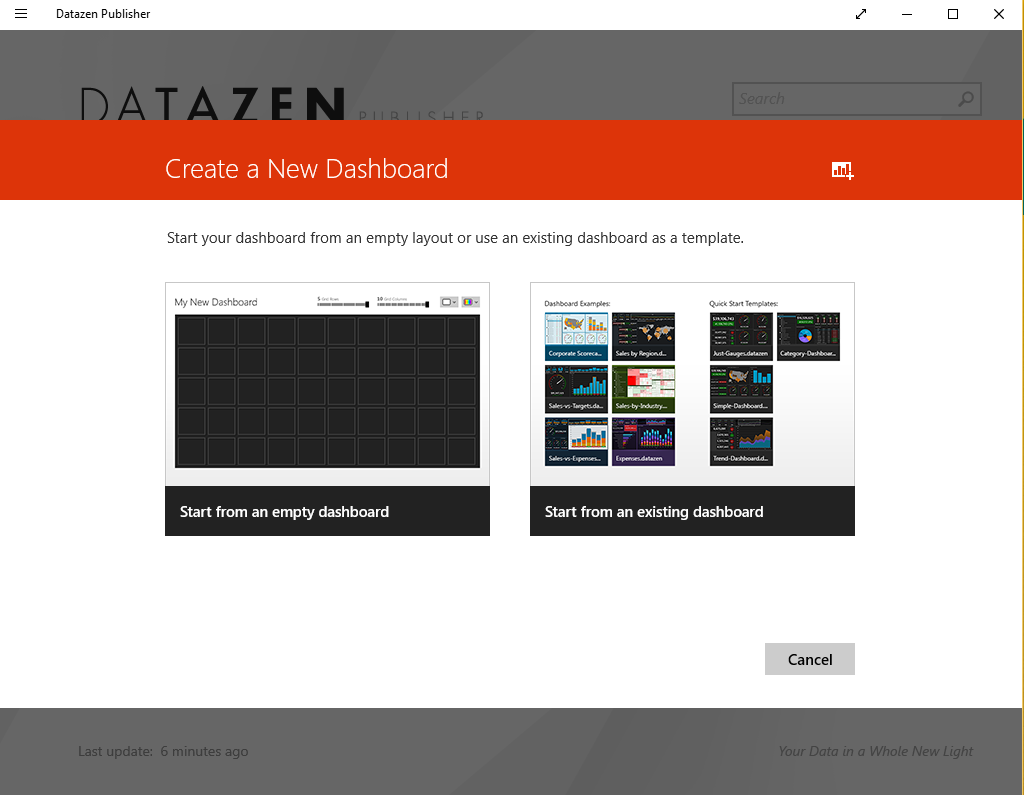


## Configuring the Client and Creating the First Dashboard.

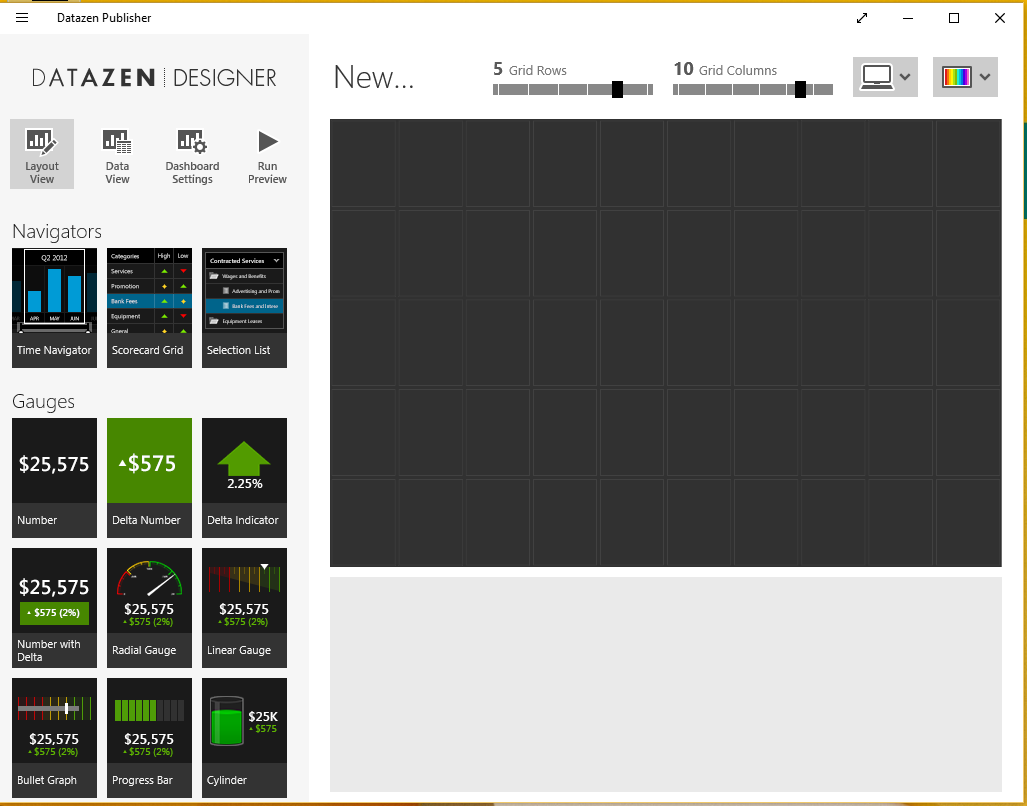
Now all we need to do is to start the creation of a dashboard. To do so right click on any wear of the console to open a button black menu. On this black menu click on the option create.



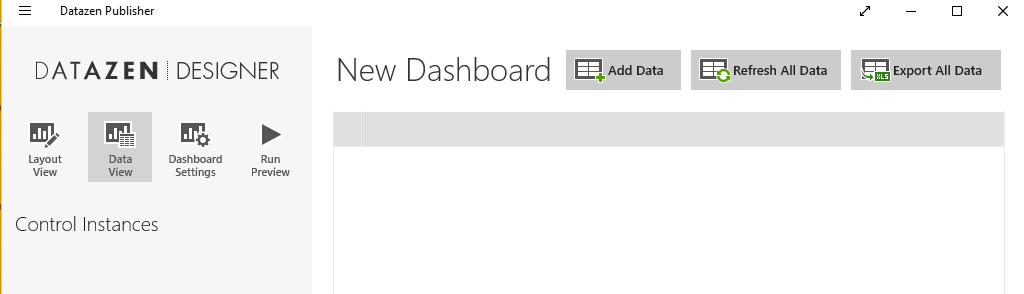
Now we will select the option Start from an empty dashboard.



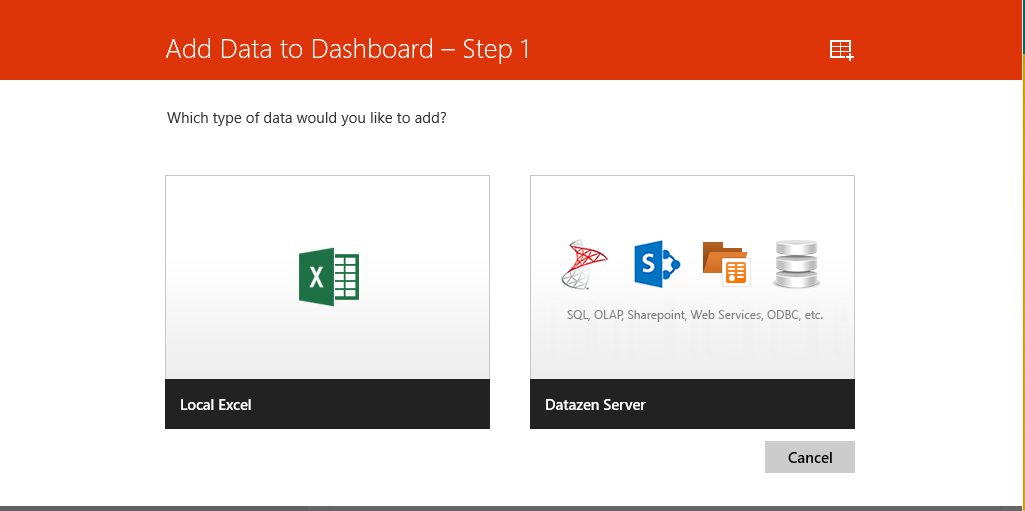
This will open the Datazen designer. The first thing we need to do is to click on the Data View button on the top left of the screen.



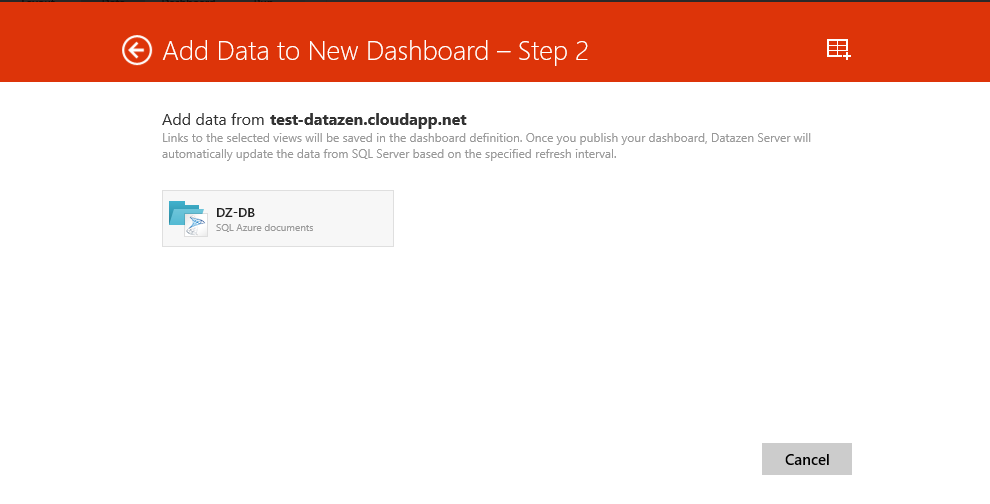
After we switch to the Data View, we will click on the Add Data button



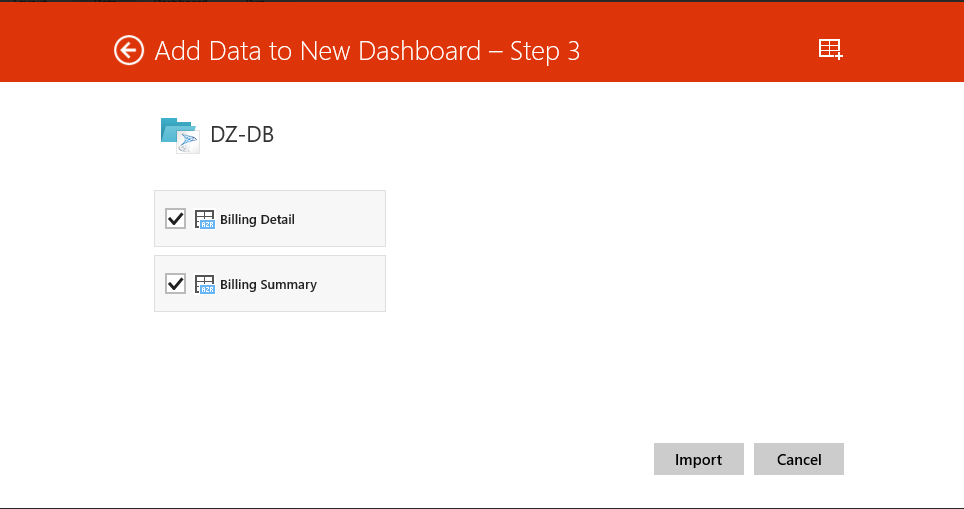
Then we will select the option Datazen Server.



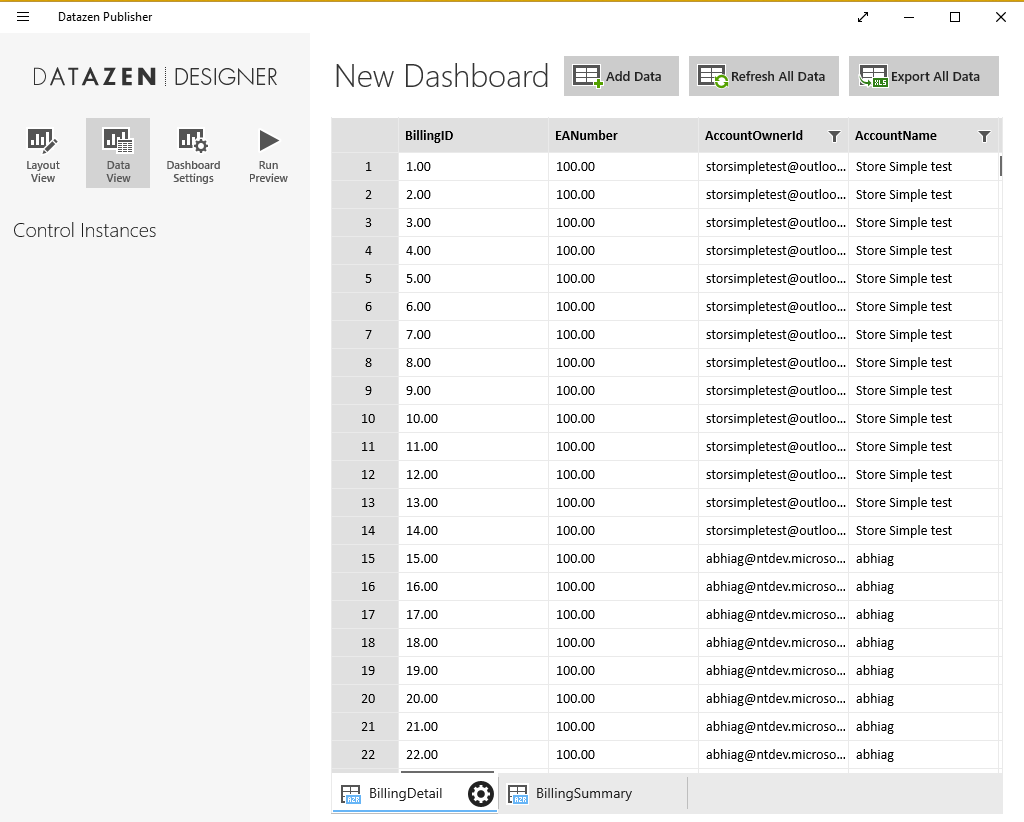
Now we will select the data source DZ-DB that we configured earlier on the Datazen server.



Finally we will select both views and click on the import button.

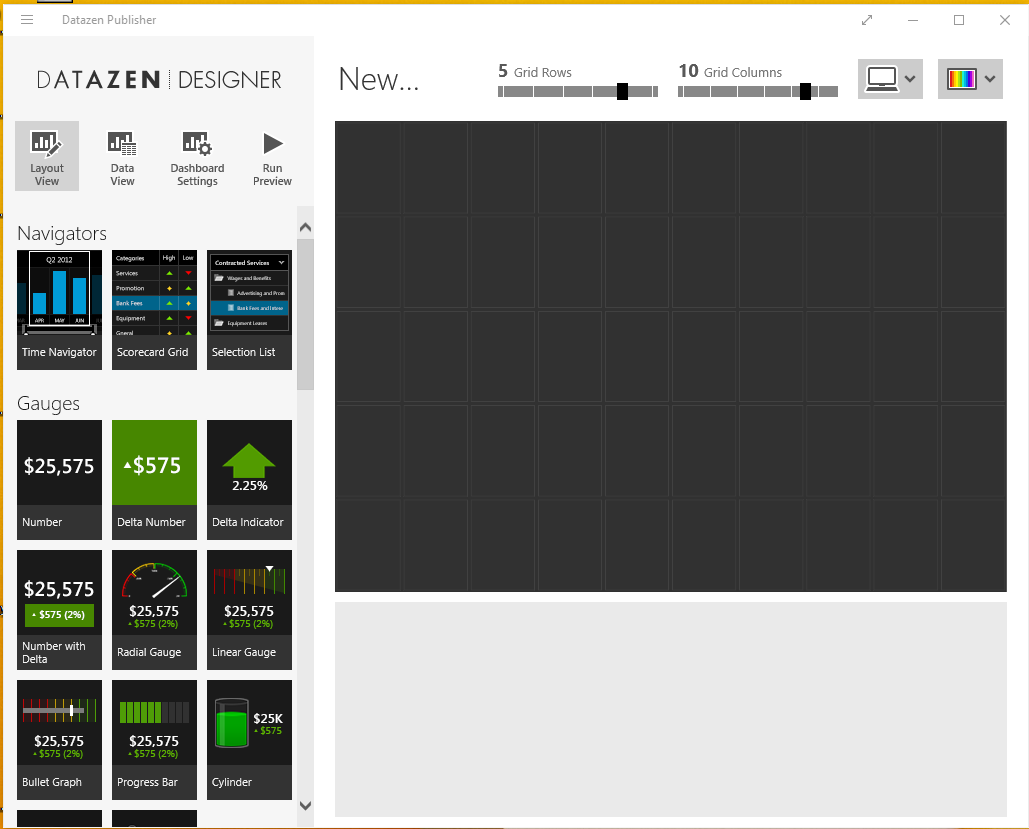


Now we are able to see that our client already sees all the billing information from our customer. Now we just need to start creating dashboards.

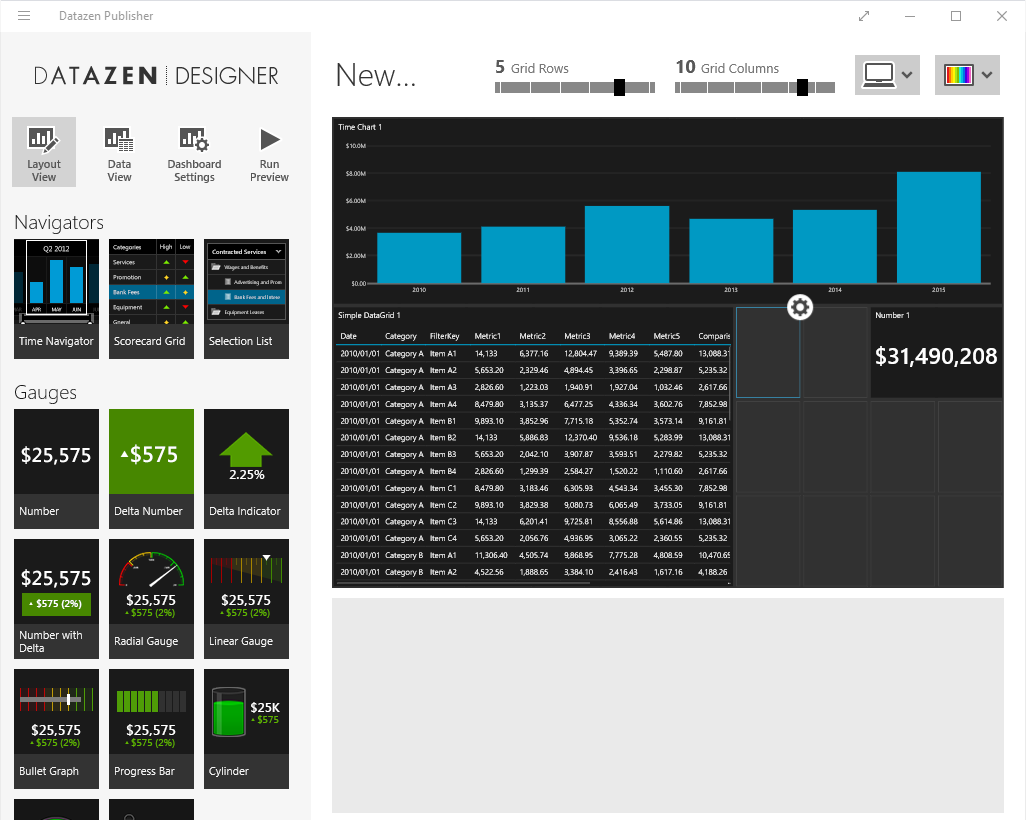


## Creating your very first Dashboard.

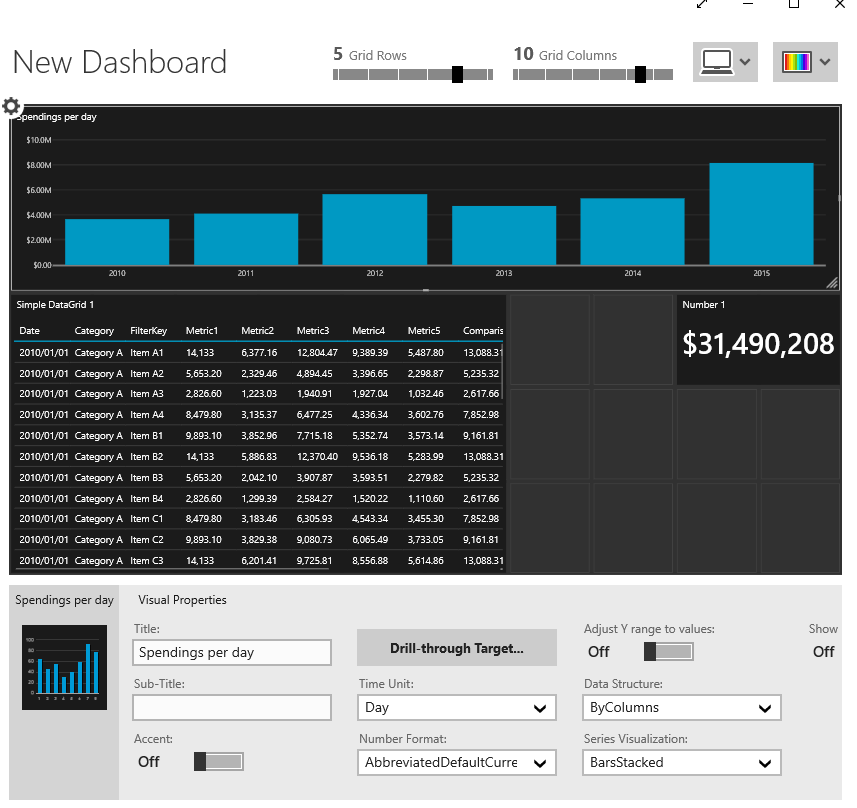
To create a very simple first dashboard, switch back to the Layout view of the Datazen Client. On this view you have on your left the types of graphs available to create a dashboard. On the right you have an empty dash board.



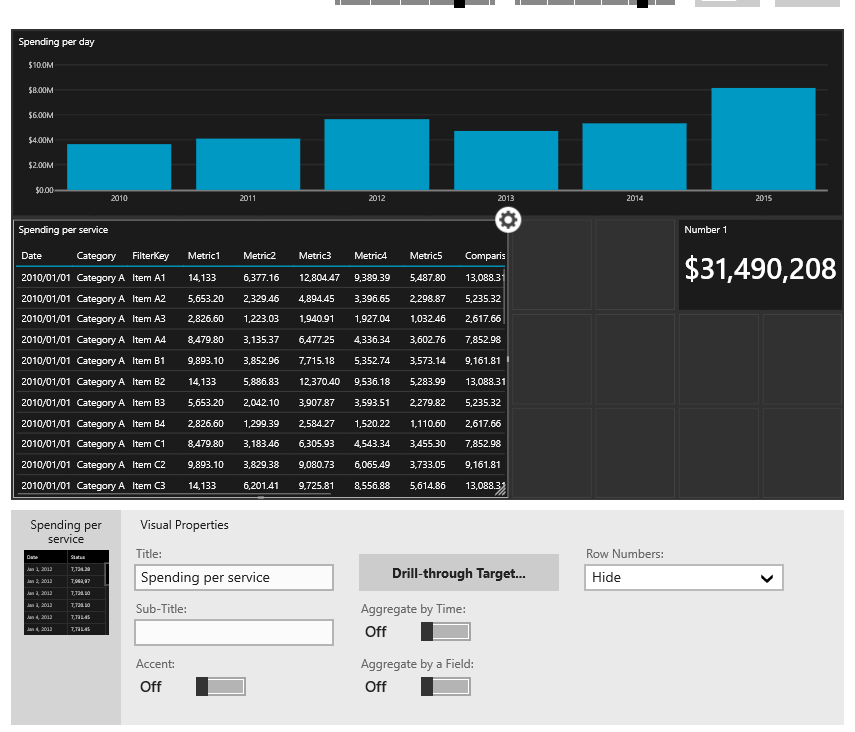
The first thing we will do is drag the time Chart from the right of the screen to the left on our blank dashboard and resize accordingly. Then we will get a Simple Datagrid and do the same thing. Last but not least we will also drag and resize a number object. The end result will look like this.



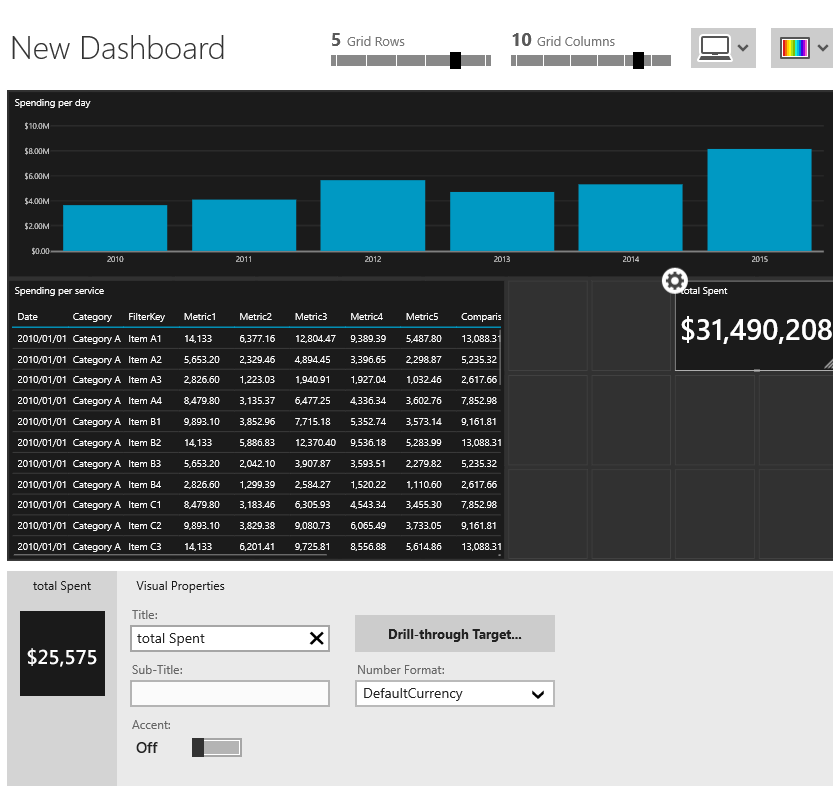
Now we will click on the Time Chart 1 and customize it with the button options that will appear. We will choose a name for it, in our case spending per day and change the time unit to days.



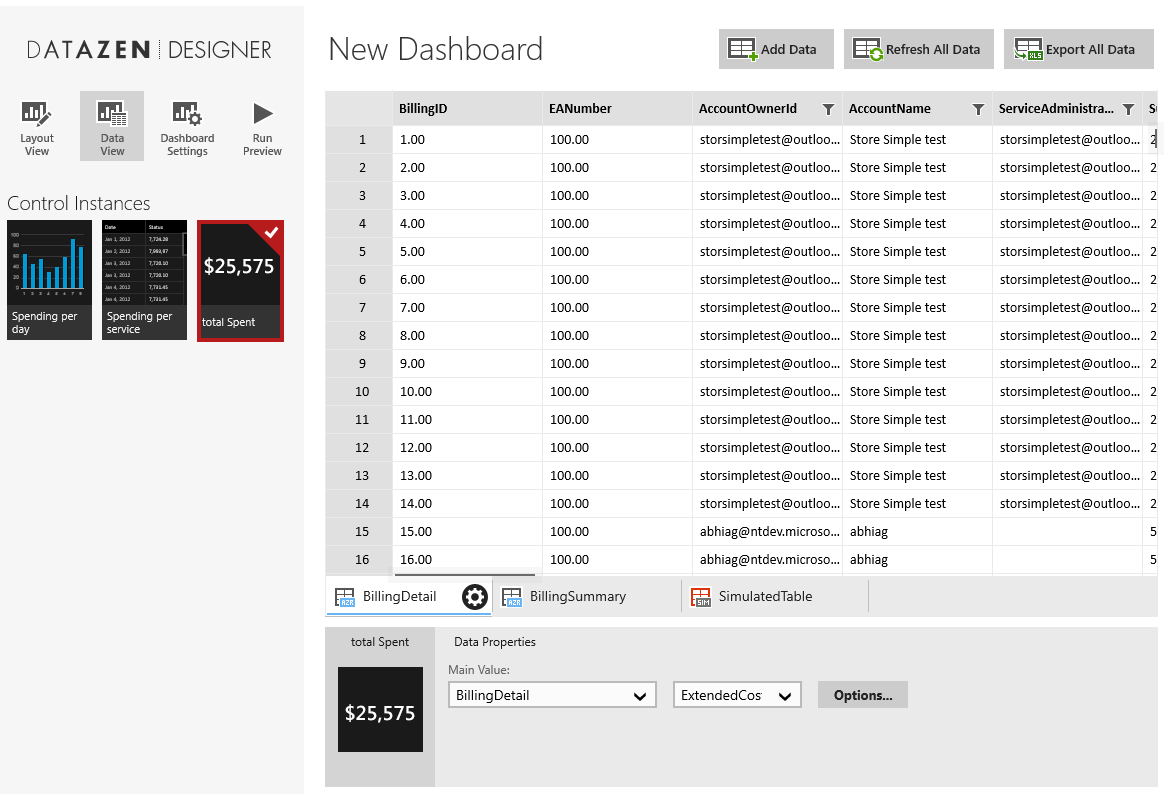
Now we will select the Simple table and we will do the same. As name we will put spending per service. And on the Row numbers we will choose to hide them



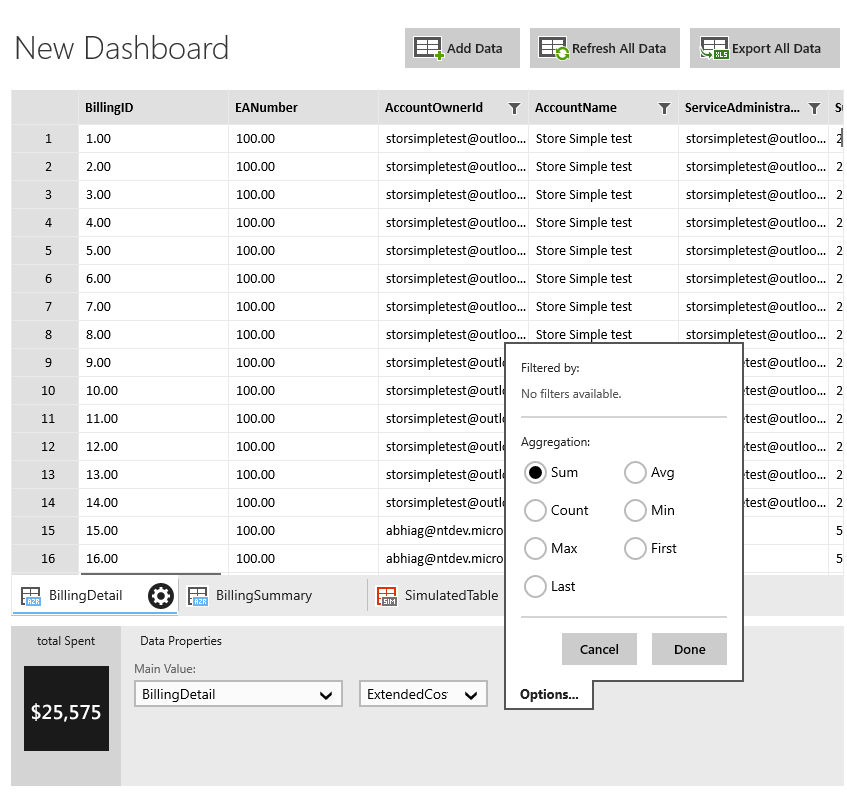
Finally we will select the Number object and change its name to Total spent



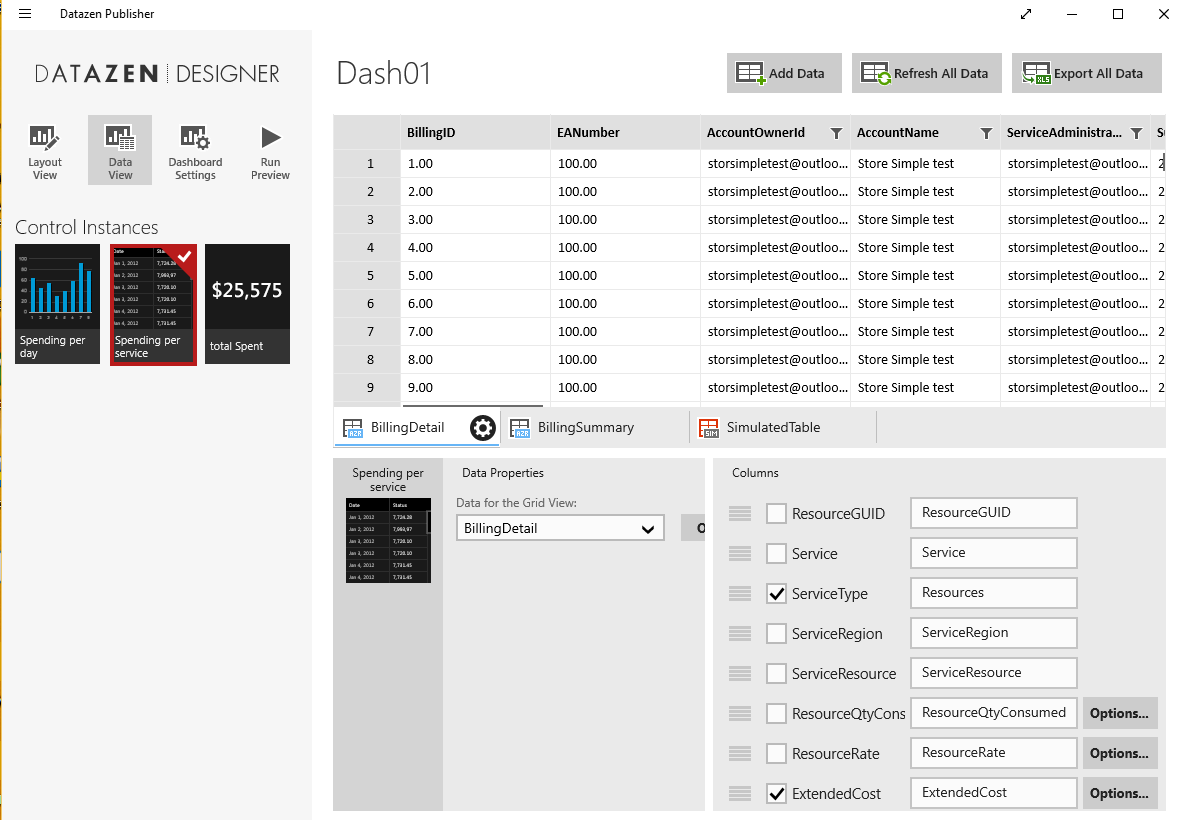
Now we need to switch to the Data View and select the information we want to show. After selecting the data view, we will select the number object, select the Billing detail data view and choose Extended Cost as the row. The screen should look like this



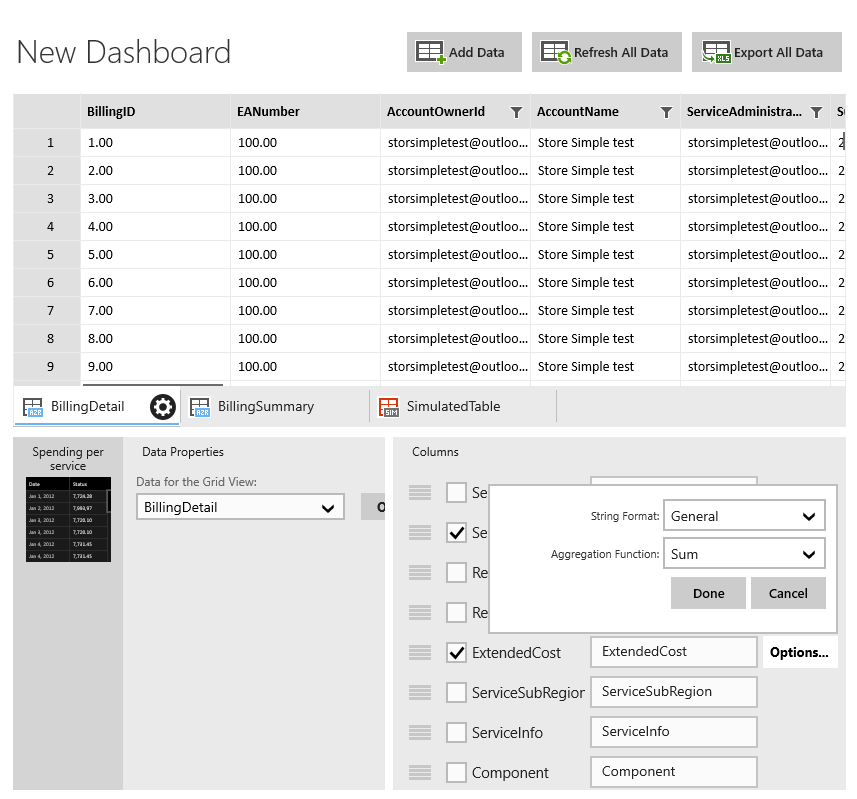
Now we will click on the Options button and select sum and click done.



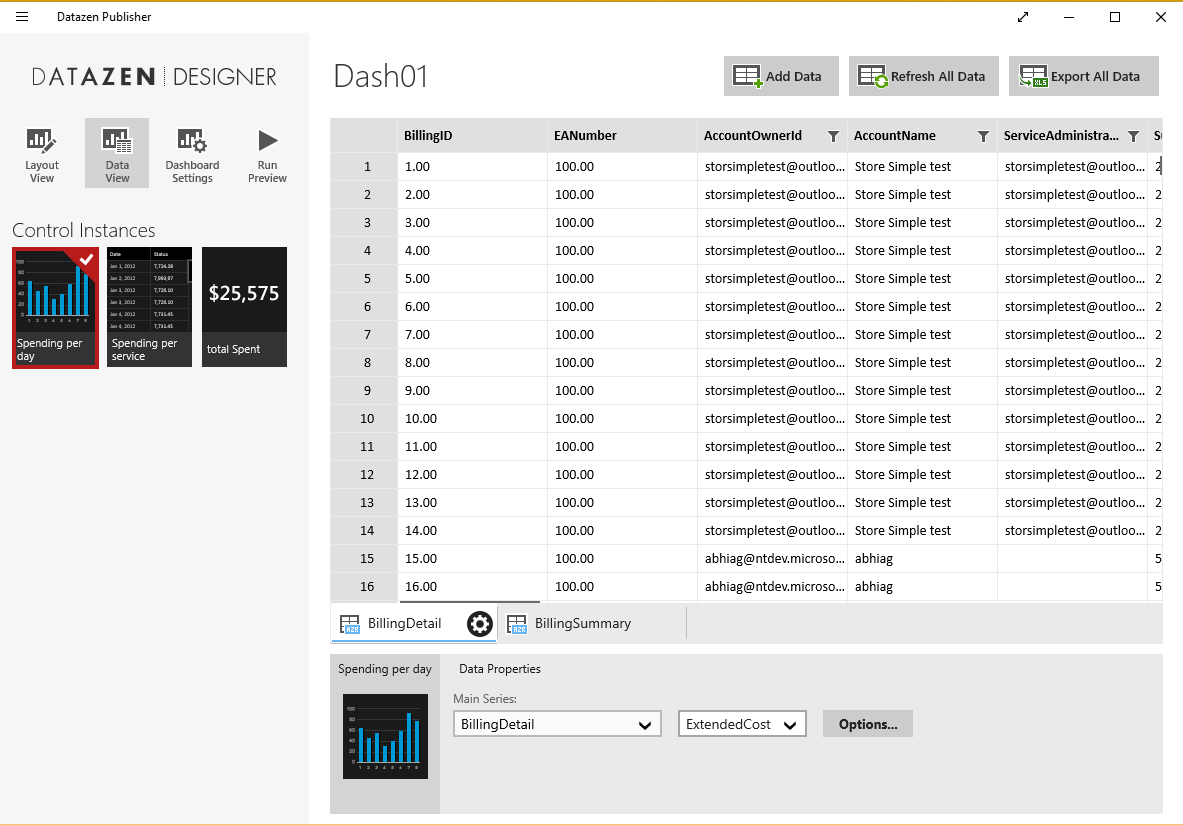
Second step, we will select the simple table object and on the right select the Billing Detail data view. On the Columns configuration we will just select Service Type and Extended Cost. We will rename the Service Type to Resources and Extended Cost to Cost.



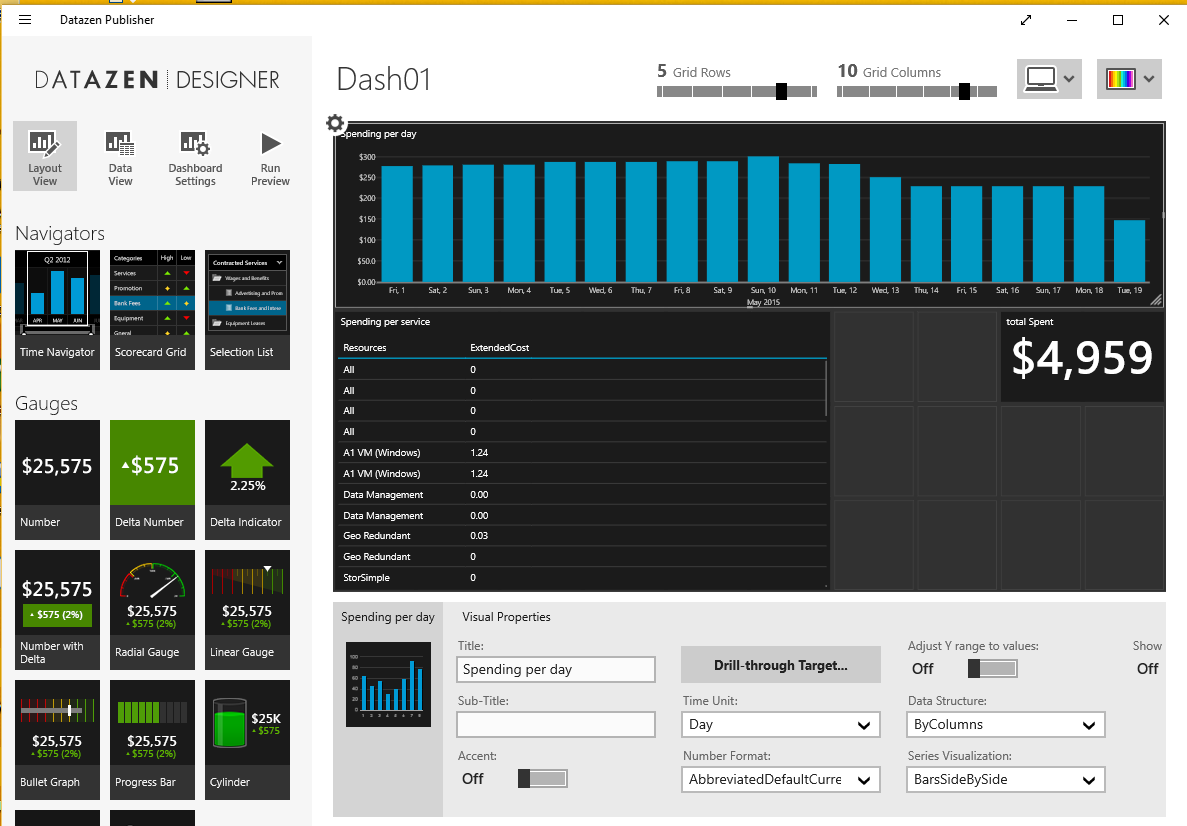
On the ExtendedCost option button we will change the Aggregation Function to Sum.



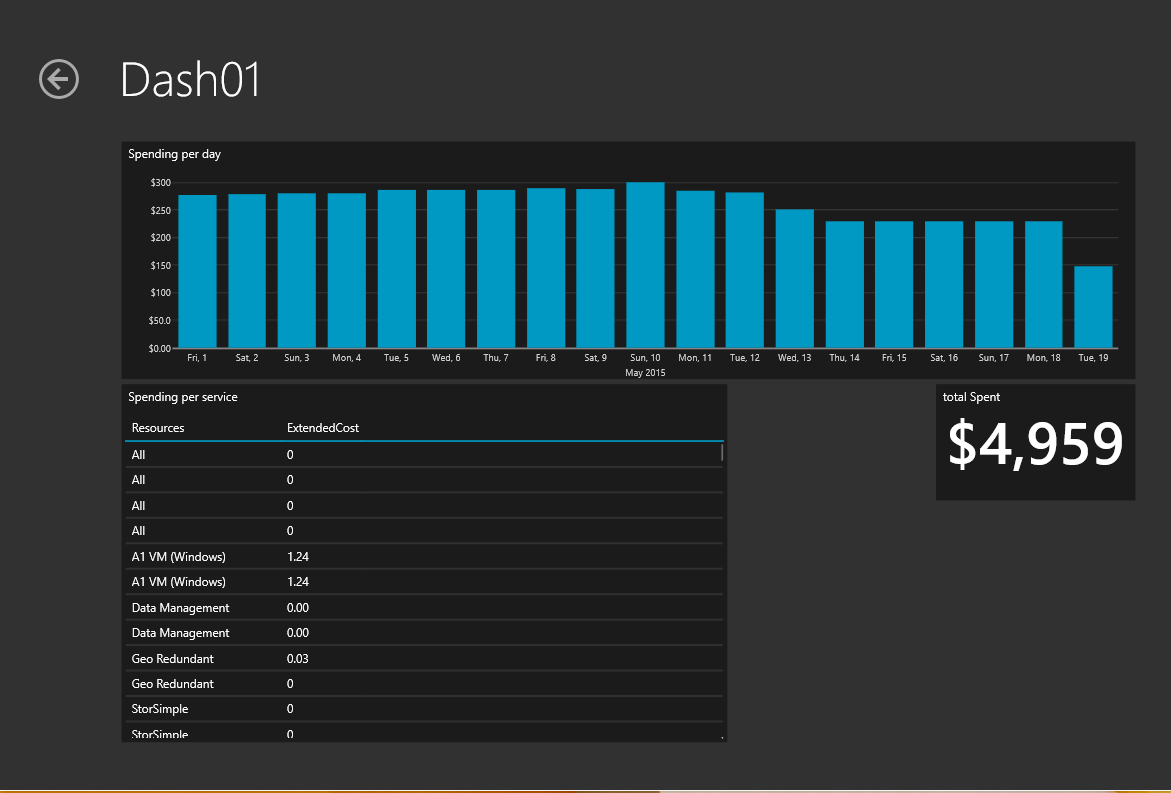
Finally we will select the Time Graph object, and select the Billing detail as our data view and the ExtendedCost as our Row.



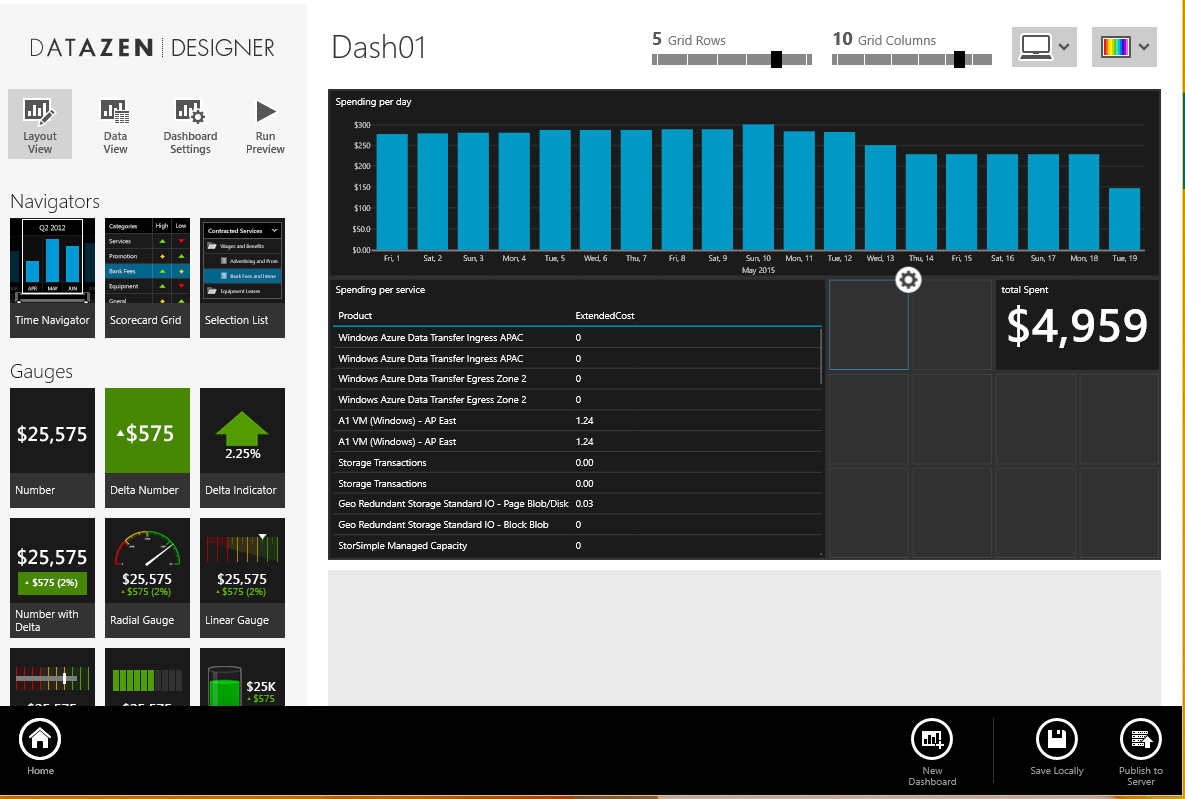
Now if we switch back to the Layout view we will be able to see how our first dashboard looks like. After you switch back to layout view, you will see the following dashboard. Press the button Run Preview to see a more accurate view of how your dashboard looks like.



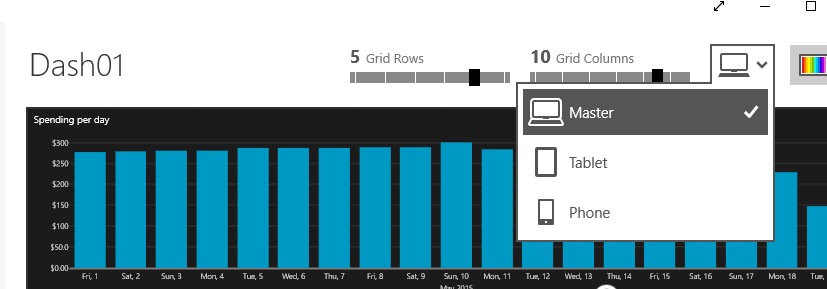
This is your Dash board on preview mode. To return just click on the back arrow. Here you can see the amount spent so far, the amount spent per day and also the amount spent by Azure feature.



Now after you click on the left arrow to return to the layout view. Just Click on the name to give it a more meaningful name. After naming it, just right click and hit save locally to save your dashboard.

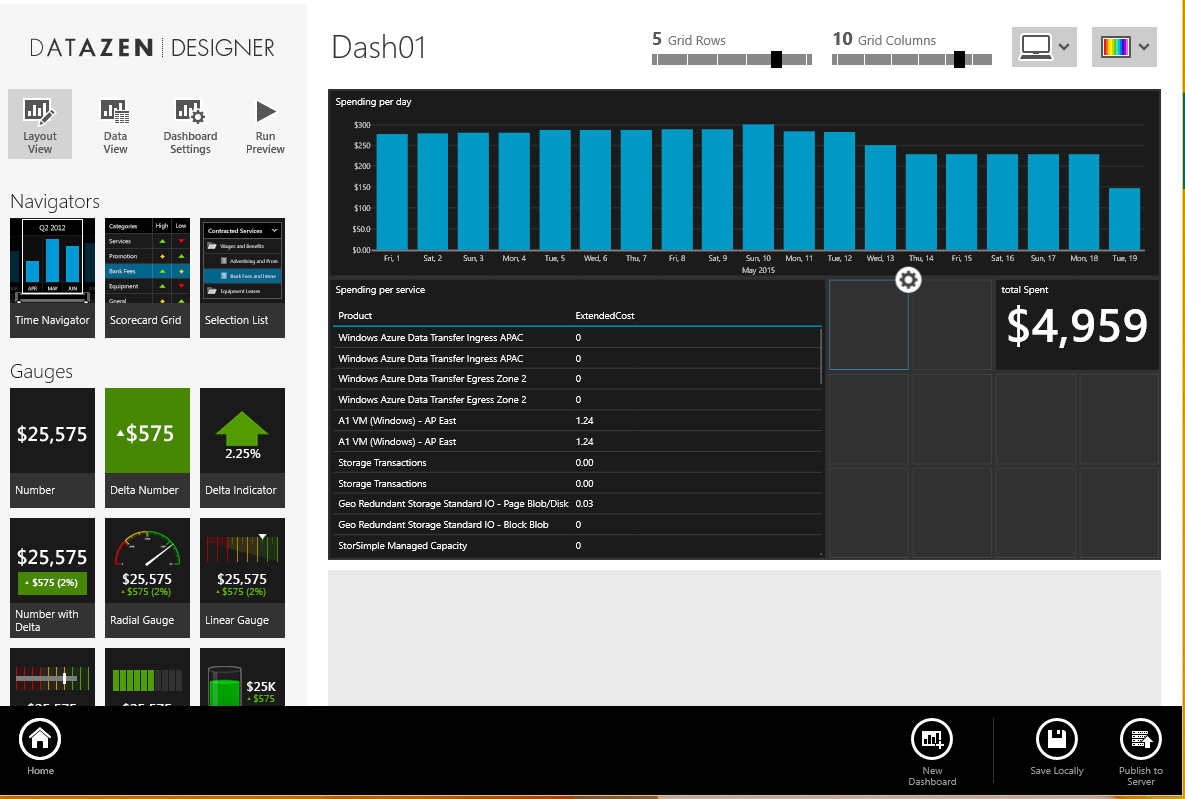


You also have the option to create dashboards for mobile devices such as phones and tablets. To do so, just select the option desired on the top menu and follow the same steps to create such dashboards.

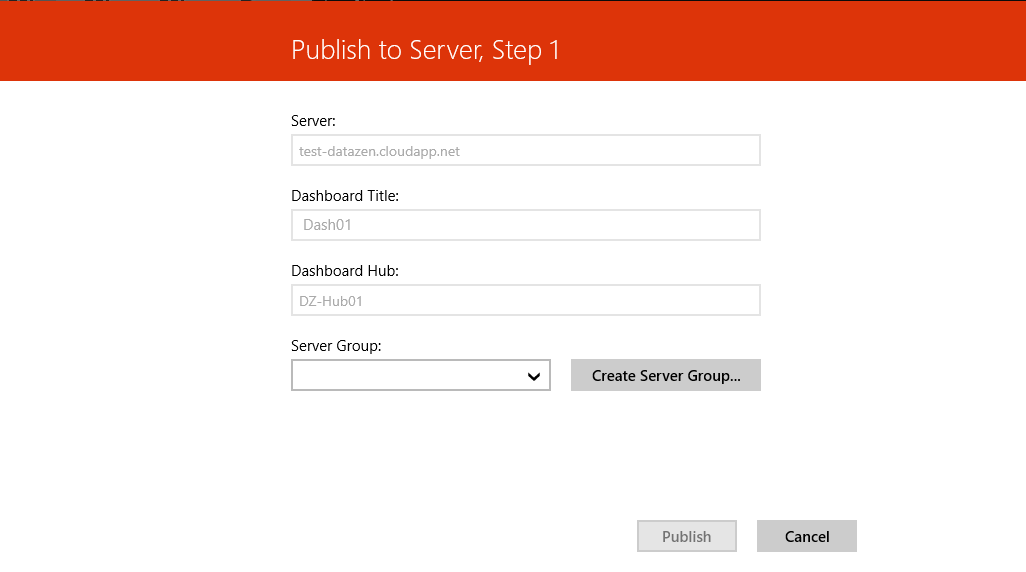


## Publishing a Dashboard

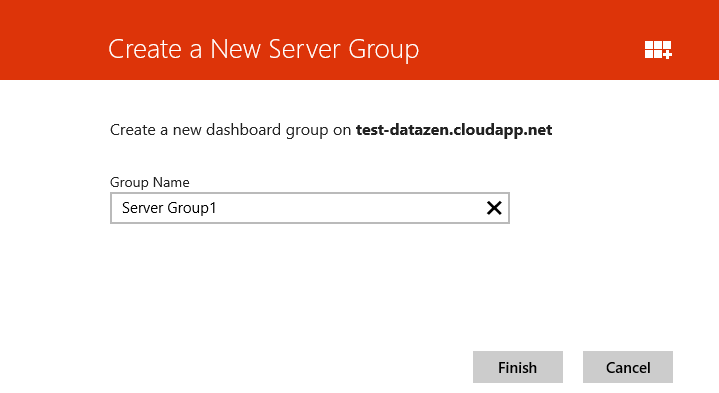
To publish a dashboard, on the Layout view, right click the screen and select publish to server.



A new dialog will open where you will have to choose a Server Group to publish your dashboard. If is the first time you are doing this. Select the option to create a server group.



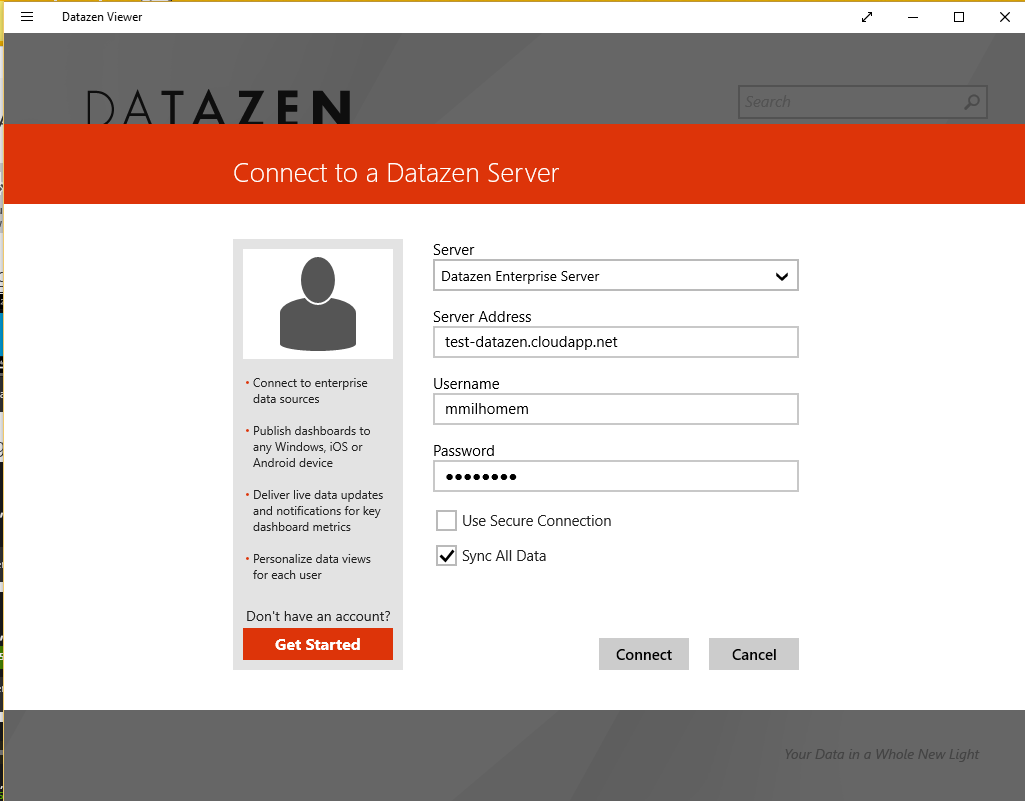
Give a name to your server group and click finish



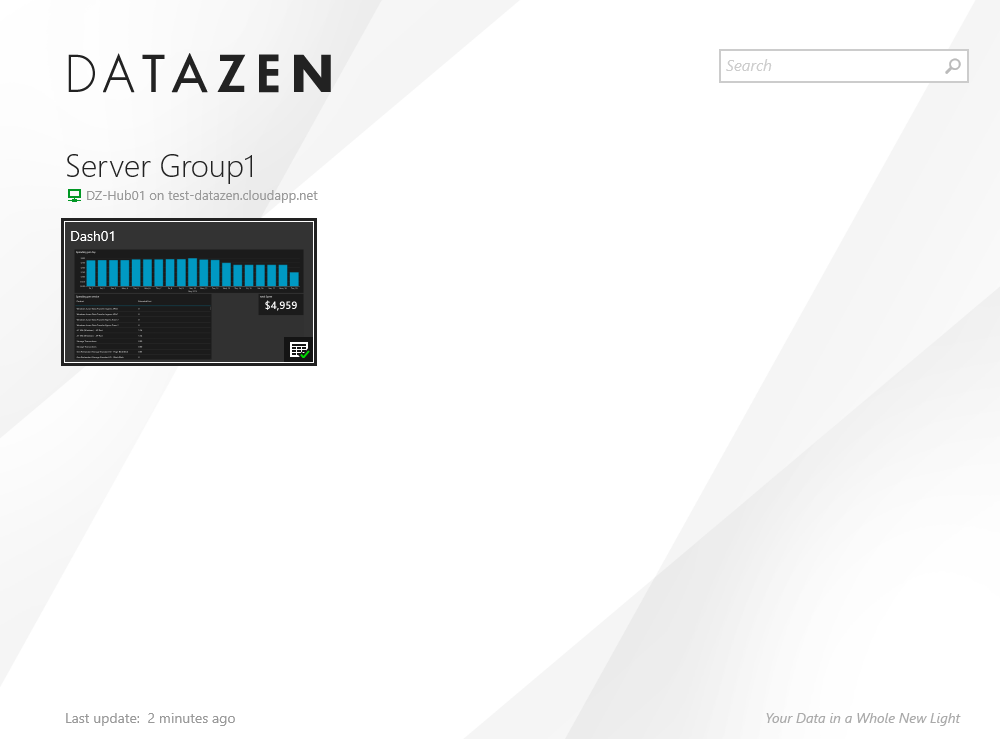
After that click on the publish button. Once your dashboard gets published, you can open the Datazen Viewer (the other app that you downloaded) to view the dashboards. Actually this is the app intended for the final users to use.

## Viewing Dashboards

Once the dashboard is published, you just need to launch the Datazen Viewer app that was already installed from the Windows Store and log in.



Once you legged in, all the dashboards published will appear and to see it you just have to click on it.



If you click on one dashboard, you will see it with the live data that we are collecting.



# References

* [What Is Azure?](https://msdn.microsoft.com/en-us/library/azure/dd163896.aspx)
* [About Azure Storage Accounts](http://azure.microsoft.com/en-us/documentation/articles/storage-create-storage-account/)
* [Azure Subscription and Service Limits, Quotas, and Constraints](http://azure.microsoft.com/en-us/documentation/articles/azure-subscription-service-limits/)
* [Manage Accounts, Subscriptions, and Administrative Roles](https://msdn.microsoft.com/en-us/library/azure/hh531793.aspx)
* [Azure Implementation Guidelines](http://blogs.msdn.com/b/thecolorofazure/archive/2014/05/13/azure-implementation-guidelines.aspx)
* [Create a Virtual Network for Site-to-Site Cross-Premises Connectivity](http://www.windowsazure.com/en-us/manage/services/networking/cross-premises-connectivity/)
* [Windows Azure Network Security Whitepaper Available](http://blogs.msdn.com/b/msftashwin/archive/2013/12/17/new-windows-azure-network-security-whitepaper-released.aspx)
* [High Availability & Availability Sets](http://msdn.microsoft.com/en-us/library/windowsazure/3d15682e-6c44-4967-88dc-0150543e4d83#bkmk_HighAvailability)
* [Step-by-Step: Build a FREE SharePoint 2013 Dev/Test Lab in the Cloud with Windows Azure Infrastructure Services](http://blogs.technet.com/b/keithmayer/archive/2013/01/07/step-by-step-build-a-free-sharepoint-2013-lab-in-the-cloud-with-windows-azure-31-days-of-servers-in-the-cloud-part-7-of-31.aspx)
* [Datazen Get started](http://www.datazen.com/start/)
* [Datazen Blogs](http://www.datazen.com/blogs/)
* [Datazen Features](http://www.datazen.com/features/)
* [End to end Datazen dashboard example](http://www.skylinetechnologies.com/Blog/Article/2527/Creating-Dashboards-with-Datazen-An-end-to-end-example.aspx)
* [Designing Datazen dashboards](https://www.youtube.com/watch?v=pgUjqVGnqFk)
* [Creating Mobile dashboards with Dataen](https://msevents.microsoft.com/CUI/EventDetail.aspx?culture=en-US&EventID=1032552946&CountryCode=US)