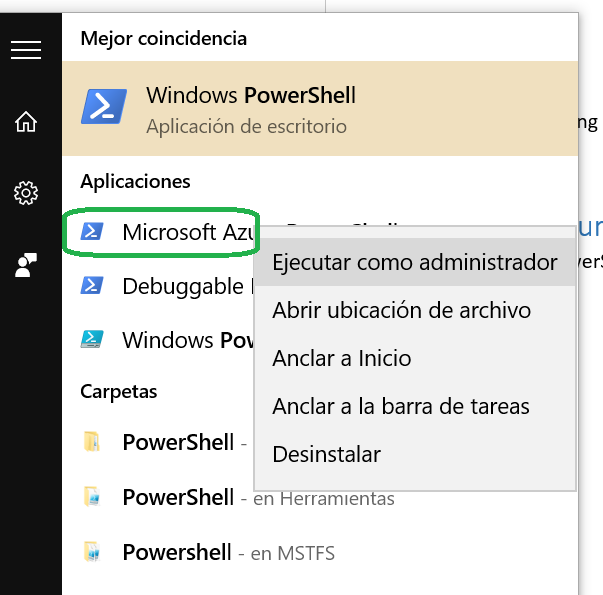
# Configuring Beacon42 – Media Services Solution VOD with Butler

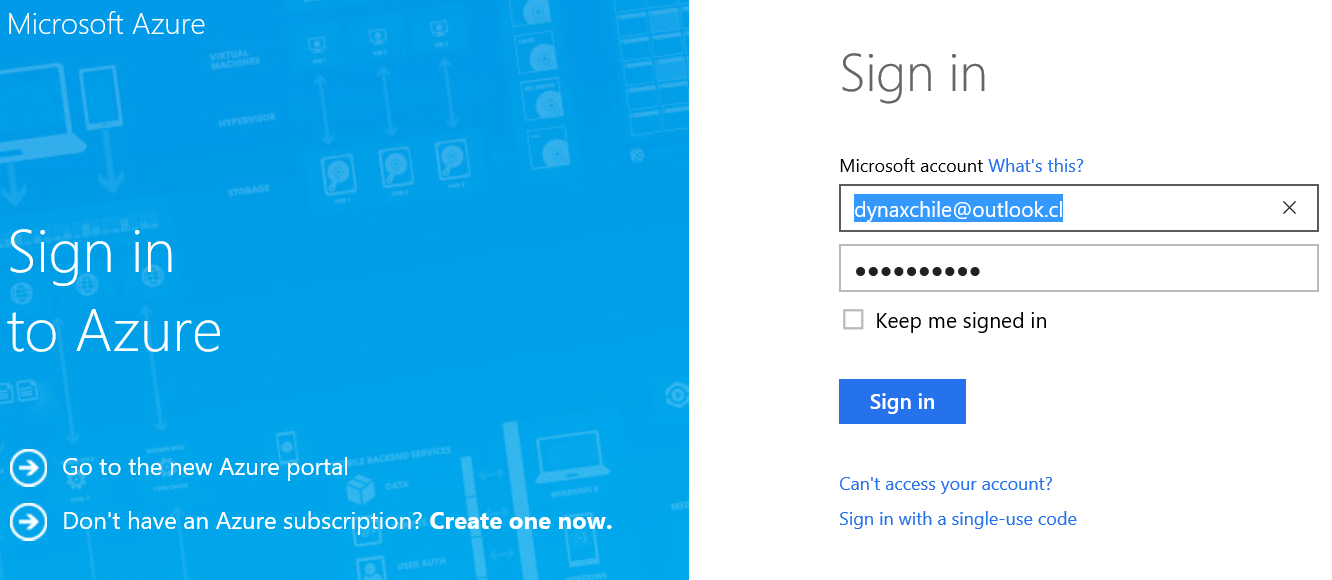
# Configure PowerShell with publishsettings file

Start Azure PowerShell with administrator



Run Get-AzurePublishSettingsFile (more info in <https://msdn.microsoft.com/en-us/library/dn385850(v=nav.70).aspx)>

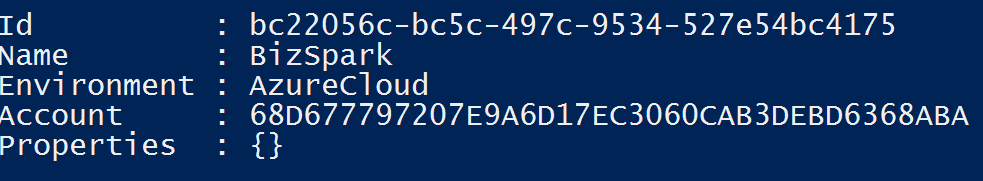
It should ask for the user (or make sure your logged in with the account which you want to configure)



Then it will download the settings file.

Import the settings with: Import-AzurePublishSettingsFile <downloaded file name>

As a result you will get the Subscription Name



In our example is “BizSpark”

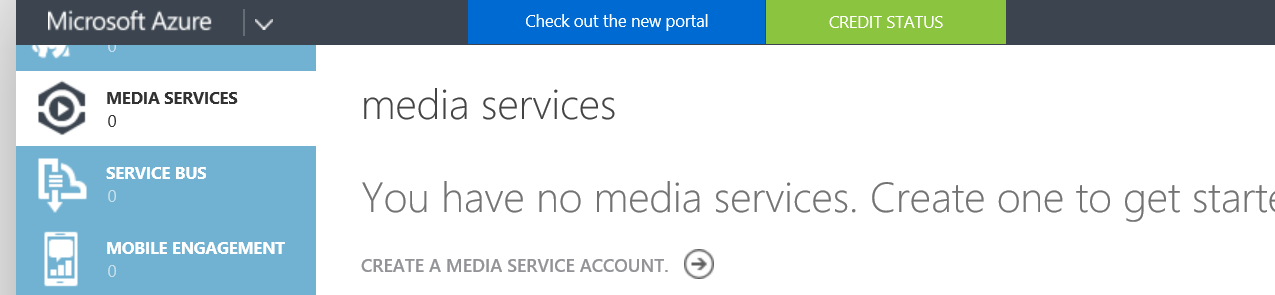
If you have more than one subscription imported, you’ll have to set the last one as the one you want to work with

Set-AzureSubscription –SubscriptionName xxxxx

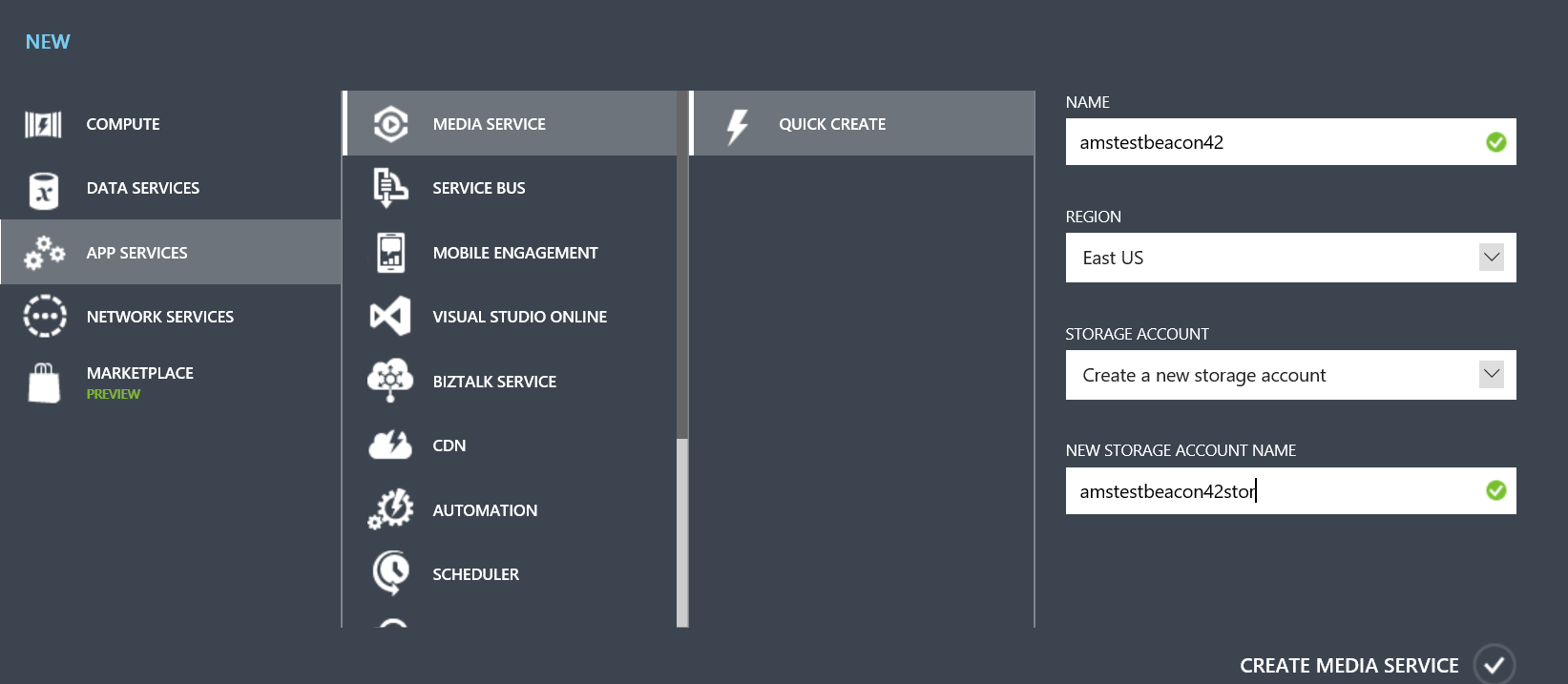
Select-AzureSubscription xxxxx

# Create a new MediaServices Account with the Portal

Go to manage.windowsazure.com select Media Services and Create a Media Service Account

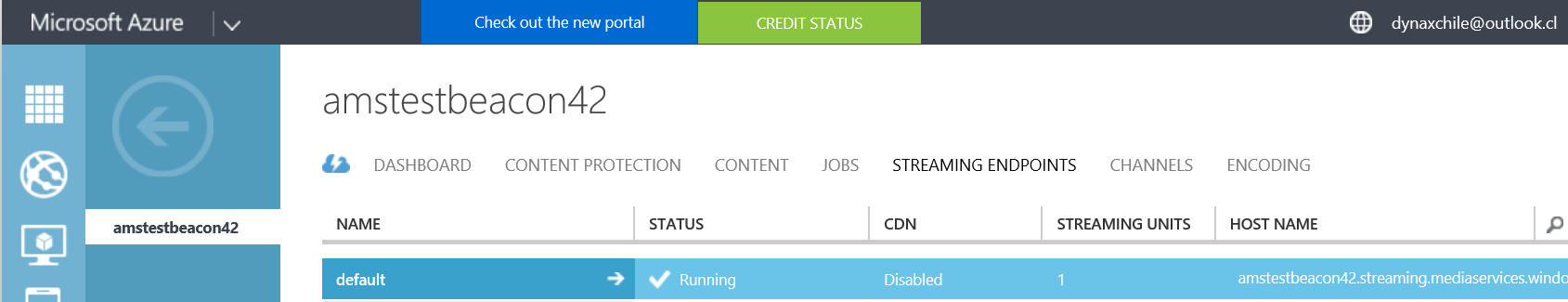


Fill the requested data as below and click on “Create Media Service”



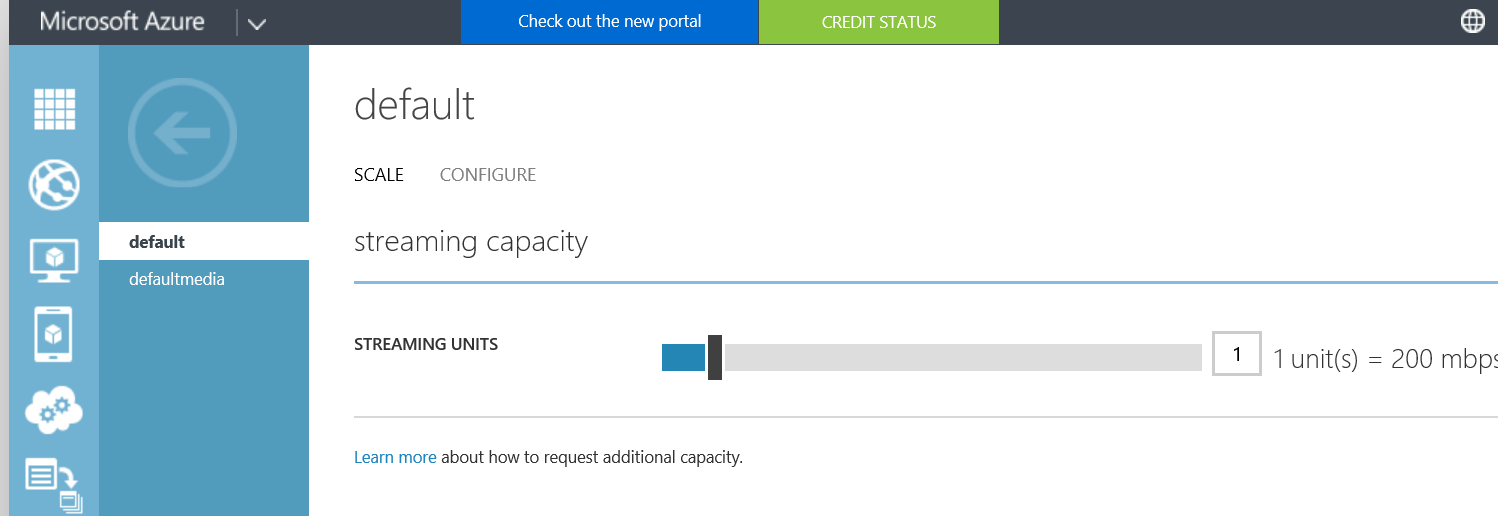
# Create an Streaming Endpoint and assign a Streaming Unit

In the portal, go to the Media Services account you have just created and select “Streaming EndPoints”



If none is created, create a new one with “default” name and do not enable Cdn.

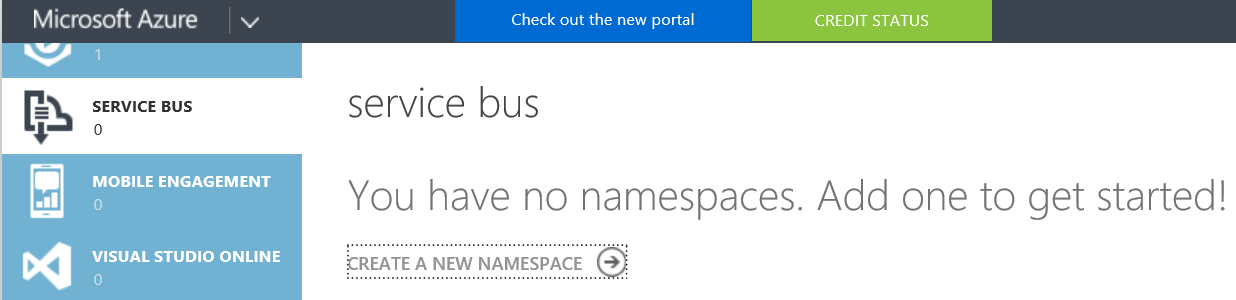
After it is created, click on the Streaming endpoint and configure 1 (one) streaming Unit



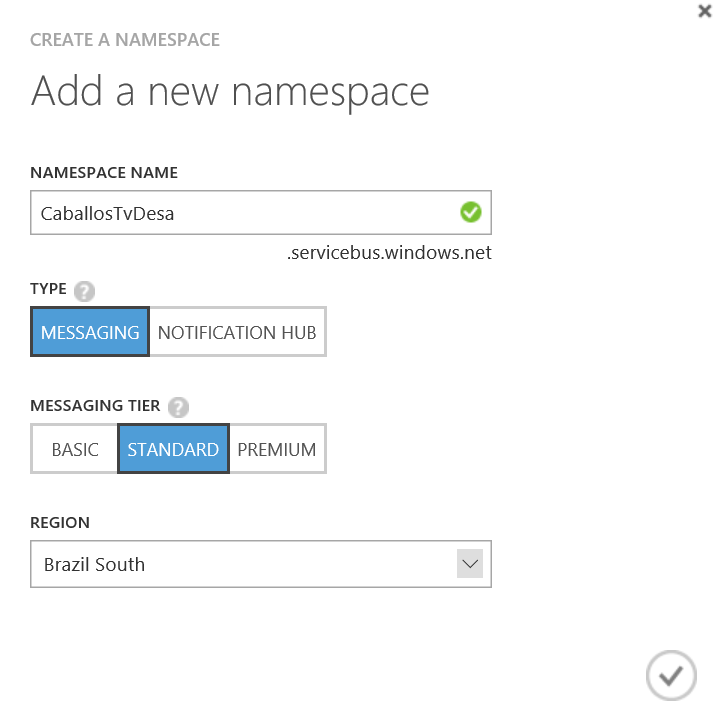
This is **mandatory** because as we will be using Dynamic Packaging if the streaming unit is not enabled, videos will not be delivered to the clients in correct format.

# Create a Service Bus Namespace

Go to Service Bus on Azure Portal and click on Create a New Namespace

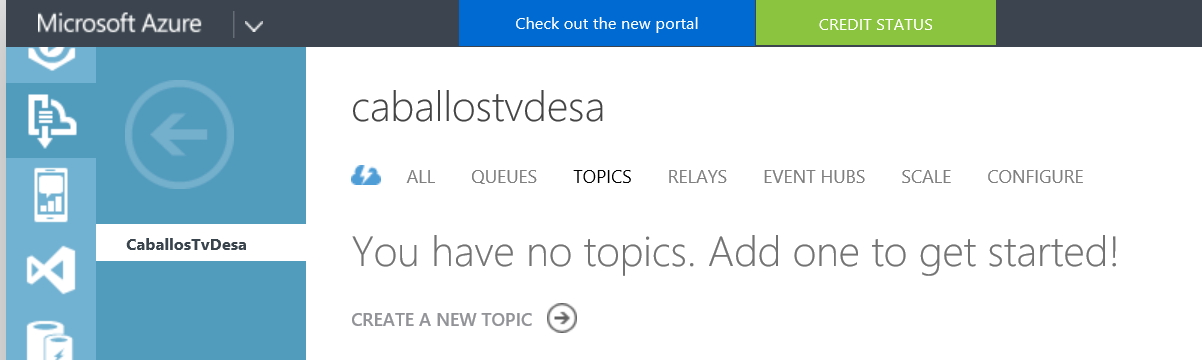


Create it with the parameters as shown below, please make sure you select “Standard” on Messaging Tier as we will be using topics.

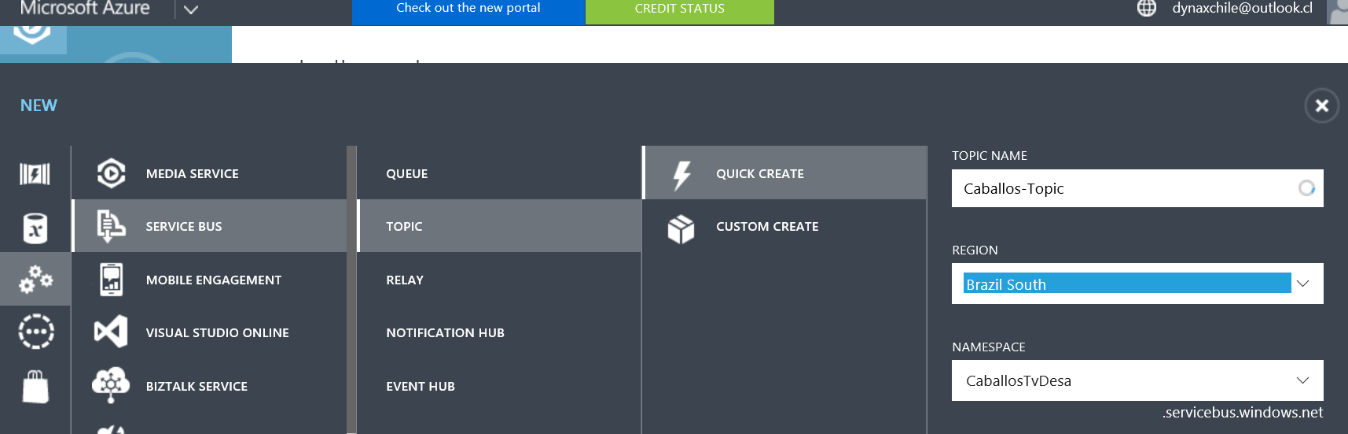


Once provisioned, click on Connection Information and copy the connection string (you will need it to configure the PowerShell Script.

Click on the Namespace and go to Topics.



Click to Create a New Topic. Select Quick Create



And fill with the Topic Name.

# Configure PowerShell Script (MediaButlerDeploy-Dec2015.ps1)

You have to configure the following parameters in PowerShell Script before running it.

$azureSubscriptionName: the name of the subscription (the one that was shown as a result of executing Import publish settings file). If you don’t have it you can get it running Get-AzureSubscription on PowerShell

$MediaServiceAccountName: Name of the recently previously created Media Services Account (amstestbeacon42)

$PrimaryMediaServiceAccessKey: Key for making operations on Media Services (you can obtain this in the portal, clicking on Manage Keys and copying the “Primary Media Services Access Key”

$butlerStorageAccountName: Name of the storage account where Configuration Tables of the solution will be created and where all blob operations will be performed. For simplicity, on Development environments, we will configure with the same storage account as Media Services (in this case amstestbeacon42stor) but on Qa/Production environments you should create a new storage account and assign this account.

$MediaStorageConn: Storage connection string in the form DefaultEndpointsProtocol=https;AccountName=amstestbeacon42stor;AccountKey=ngsdfasdf where AccountName is the name of the storage account for the Media Services Account. Beware of this if you use a different storage for butlerStorage.

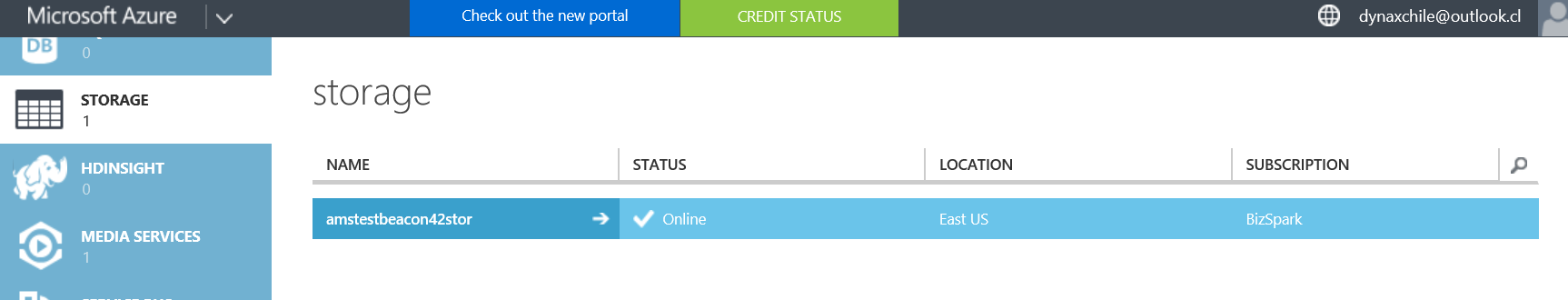
$ServiceBusConnection: Configuration to access Service Bus notifications in the sample. It’s in the form {""connectionString"": ""Endpoint=sb://caballostvdesa.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=ljlkjljlkjl4dlabfk+0lk0="",""topicText"": ""Caballos-Topic"", ""SubscriptionName"": ""CaballosSubscription""

Where connectionString must be filled with the Connection information for the ServiceBus Namespace and topicText with the Topic Name we created.

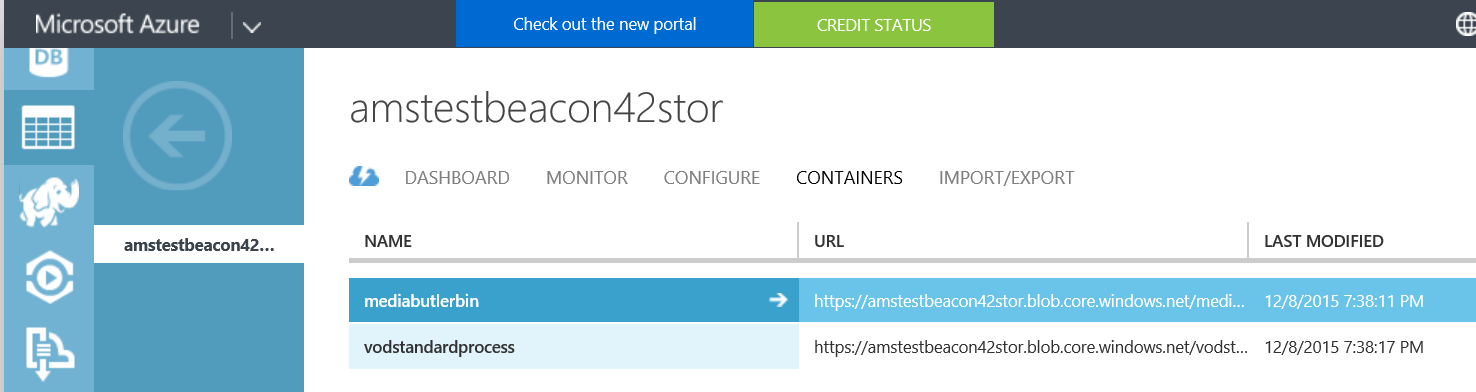
# Execute PowerShell Script (MediaButlerDeploy-Dec2015.ps1)

Open PowerShell Script in Windows PowerShell Ise or in PowerShell (always start as administrator) and Execute the script.

Everything should run Ok and you can verify in

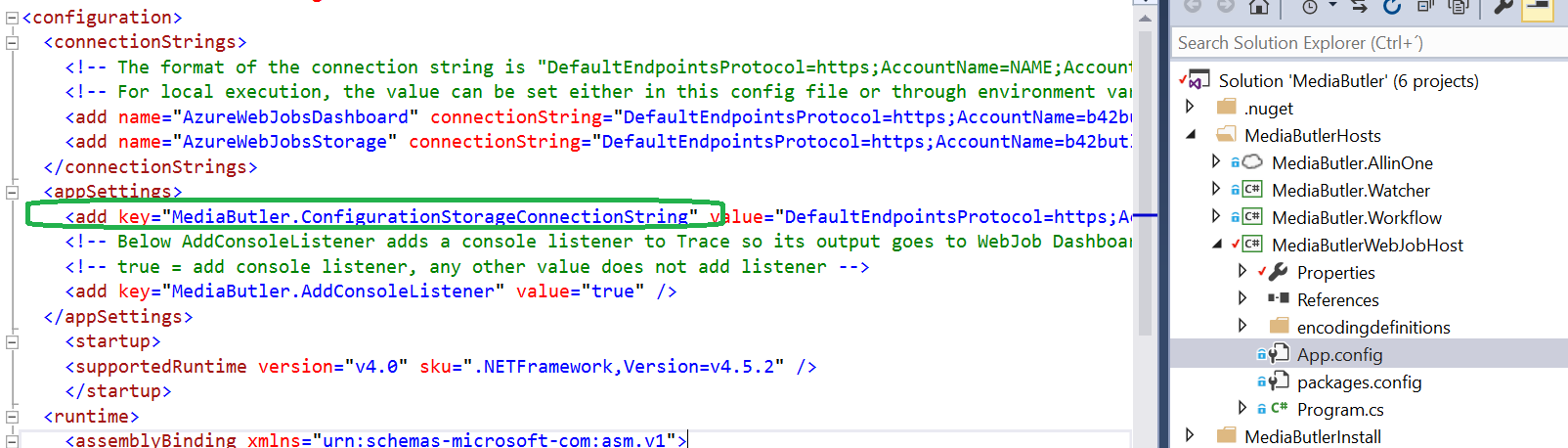


Your storage you can click on your storage and in containers you should see a mediabutlerbin container and a vodstandardprocess container



# Install WebJob

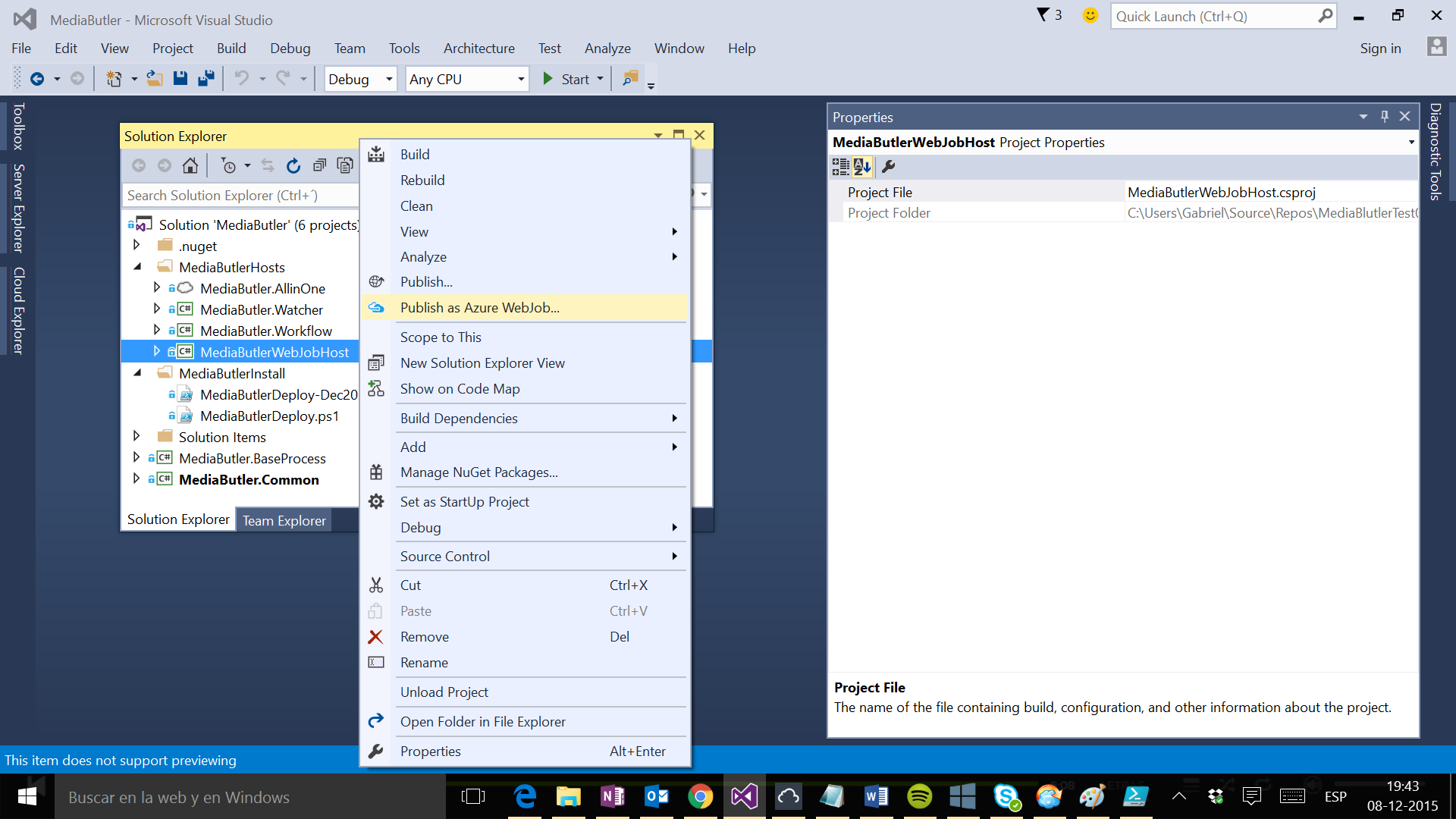
Modify the appconfig for MediButlerWebJobHost



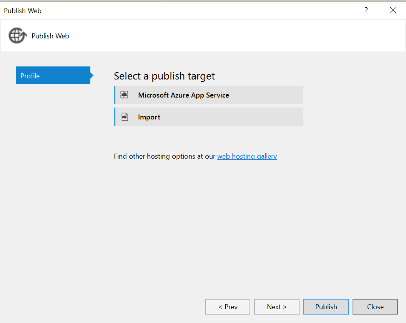
On MediaButler.ConfigurationStorageConnectionString you should configure the connection string to access the storage where the ConfigurationTable is located

DefaultEndpointsProtocol=https;AccountName=amstestbeacon42stor;AccountKey=AI2PlfovtOqFdEYzI/T2VFIEqqYSljljheCIggAZbNfGv6U6BqGHBsagH38p4wicfpsPomA==

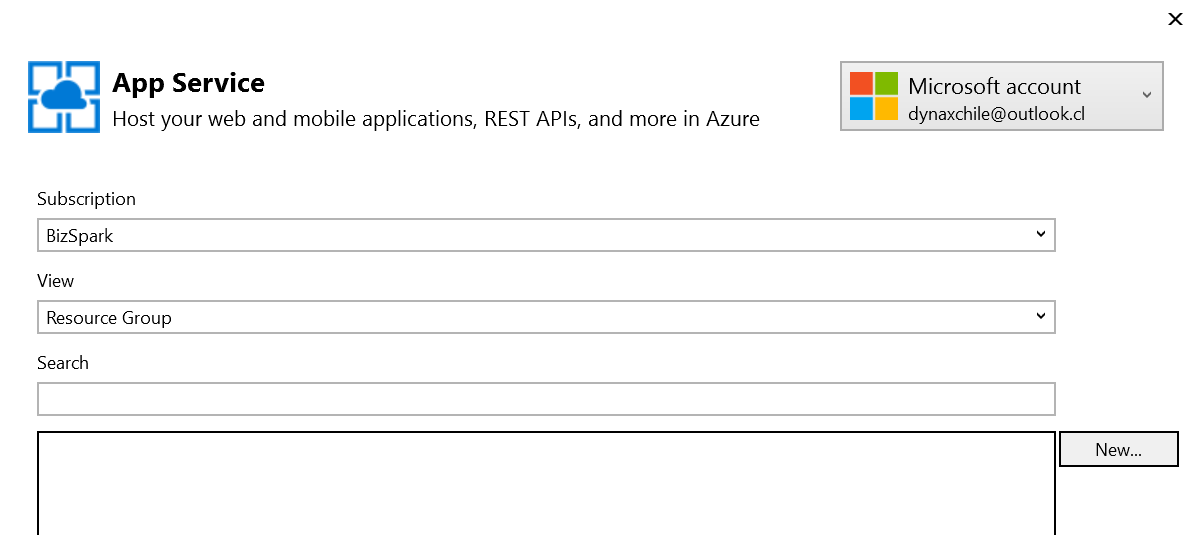
Go to Visual Studio and on the solution go to MediaButlerWebJobHost. Right Click and select Publish as Azure WebJob



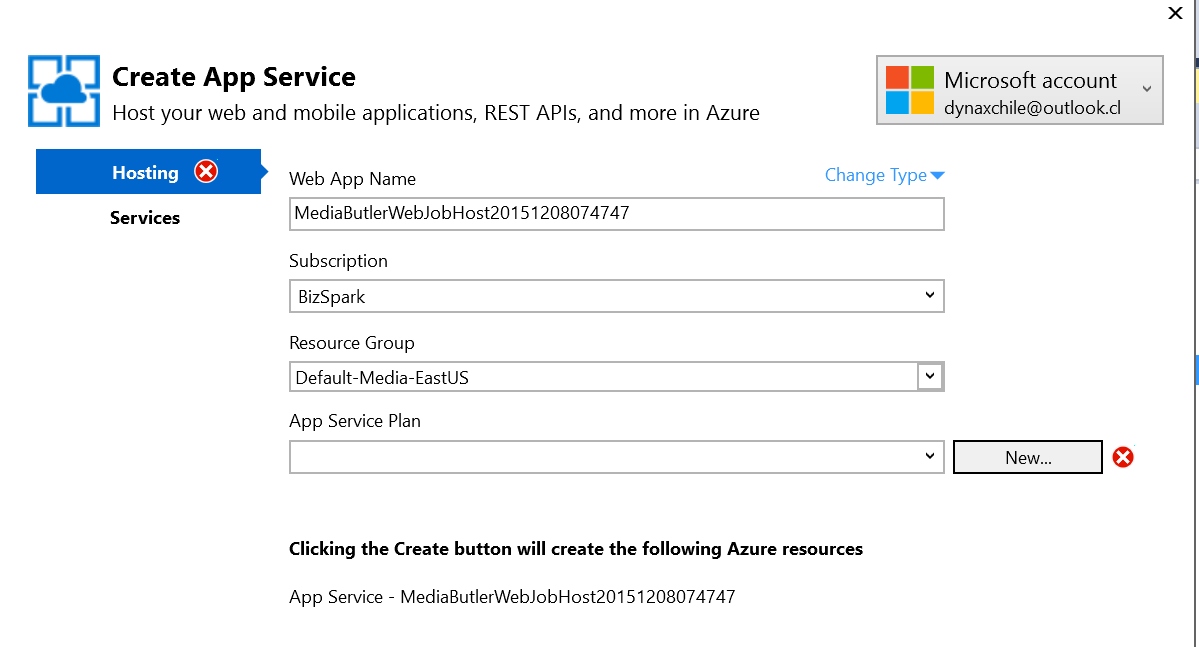
Select Microsoft Azure App Service



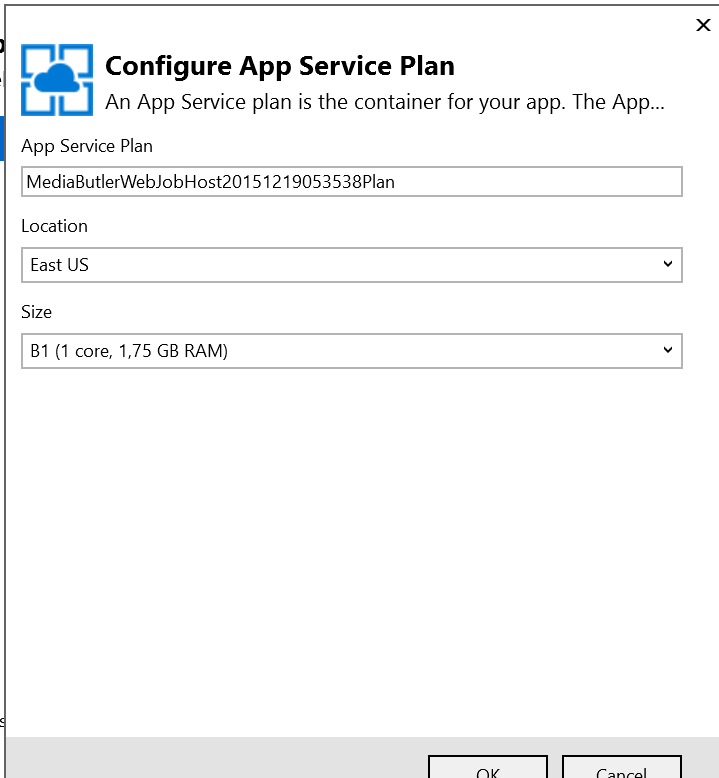
Select the account where Azure Media Services was created and then click on New



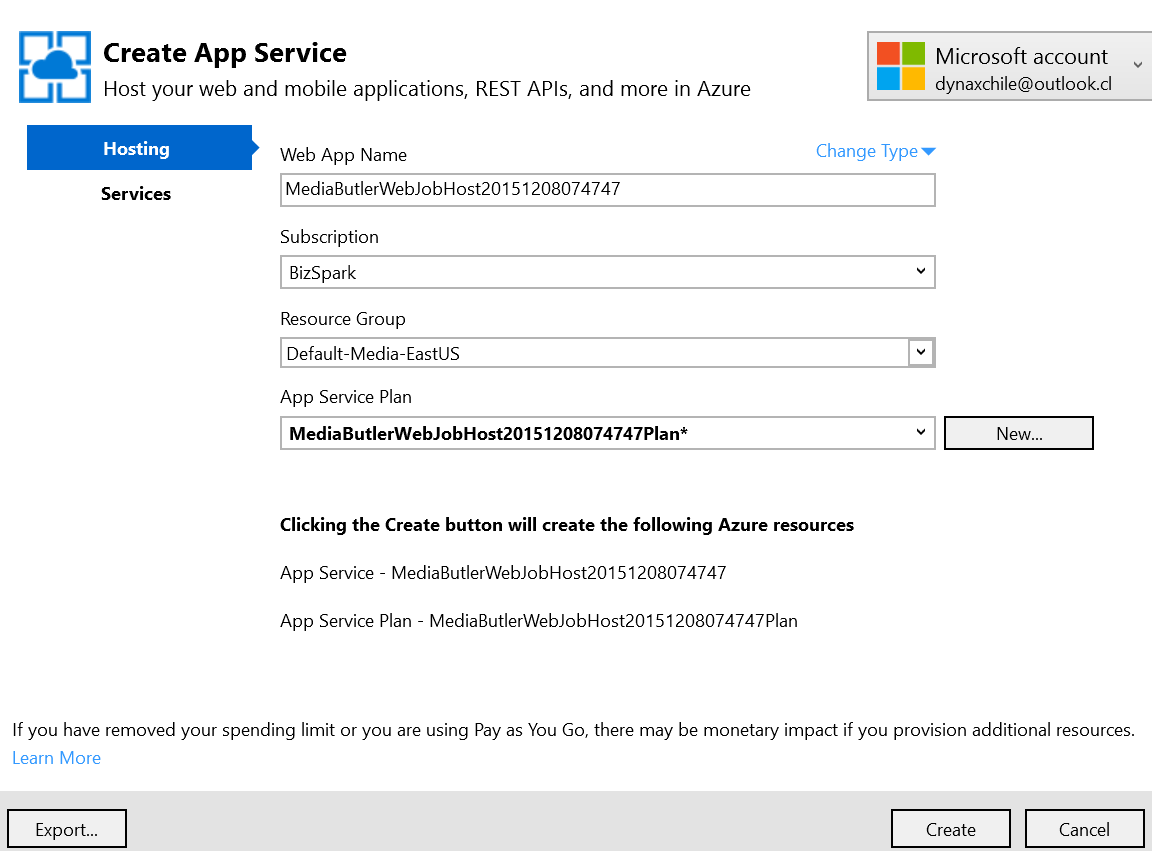
If it’s your first App Service you will have to create a new App Service Plan. Click on New



Select a Basic plan so you can later choose Continuous as execution mode for the webjob

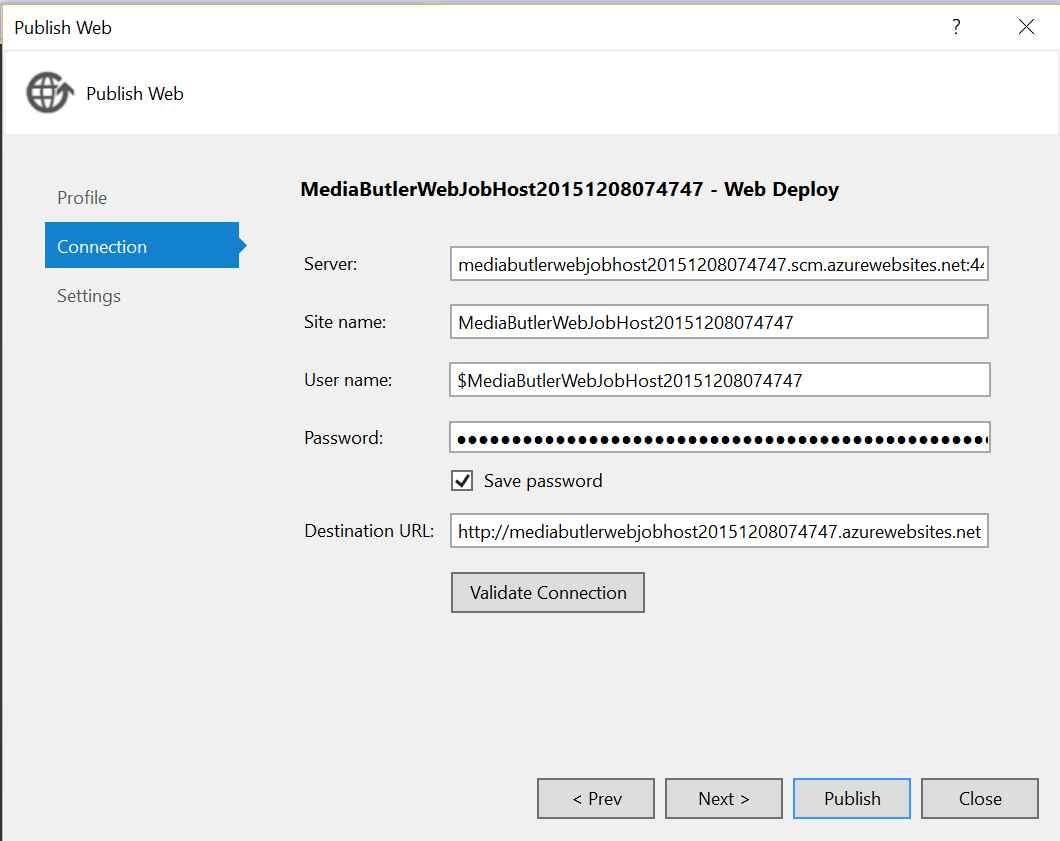


And then click on Create

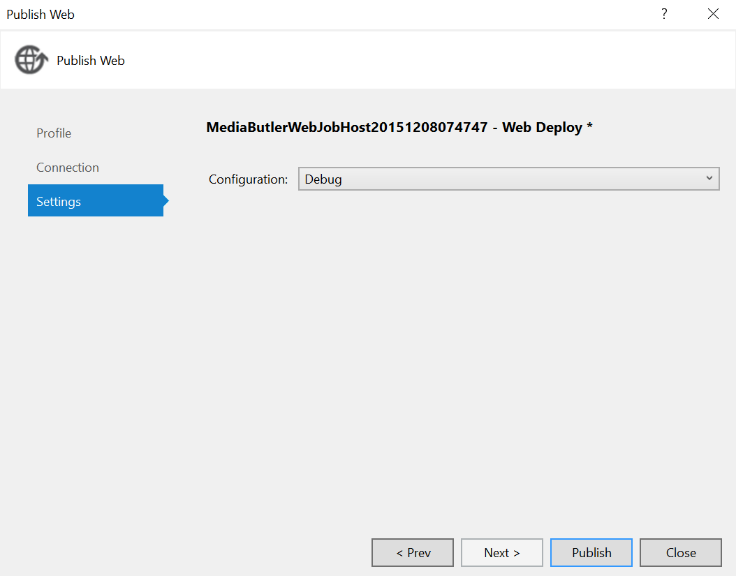


And wait for deployment to complete.

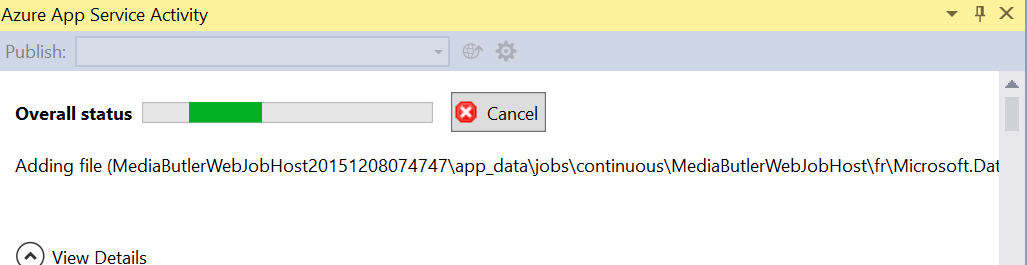
On the next screen click on validate Connection and then on Next



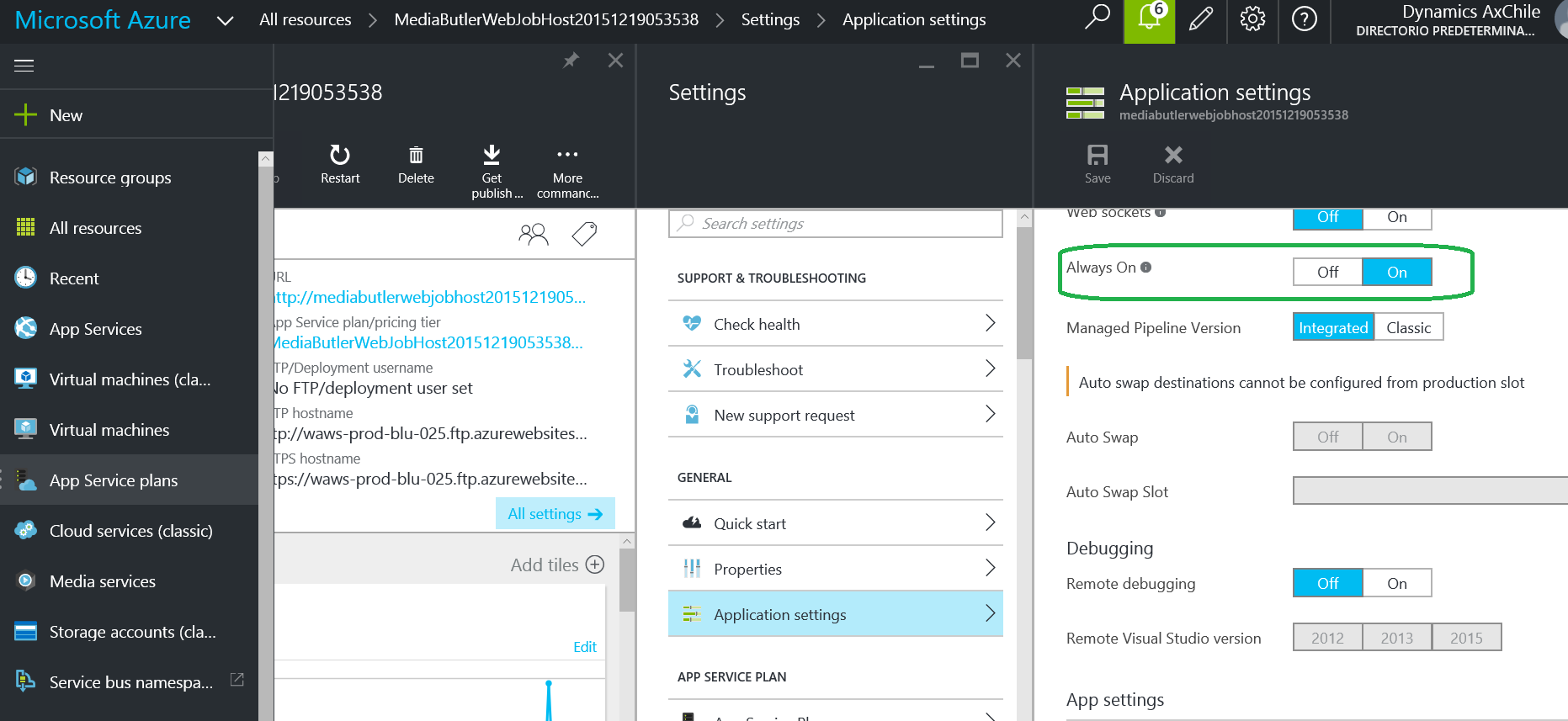
Select Debug Configuration and click on Publish



On Azure App Service Activity you should see the progress



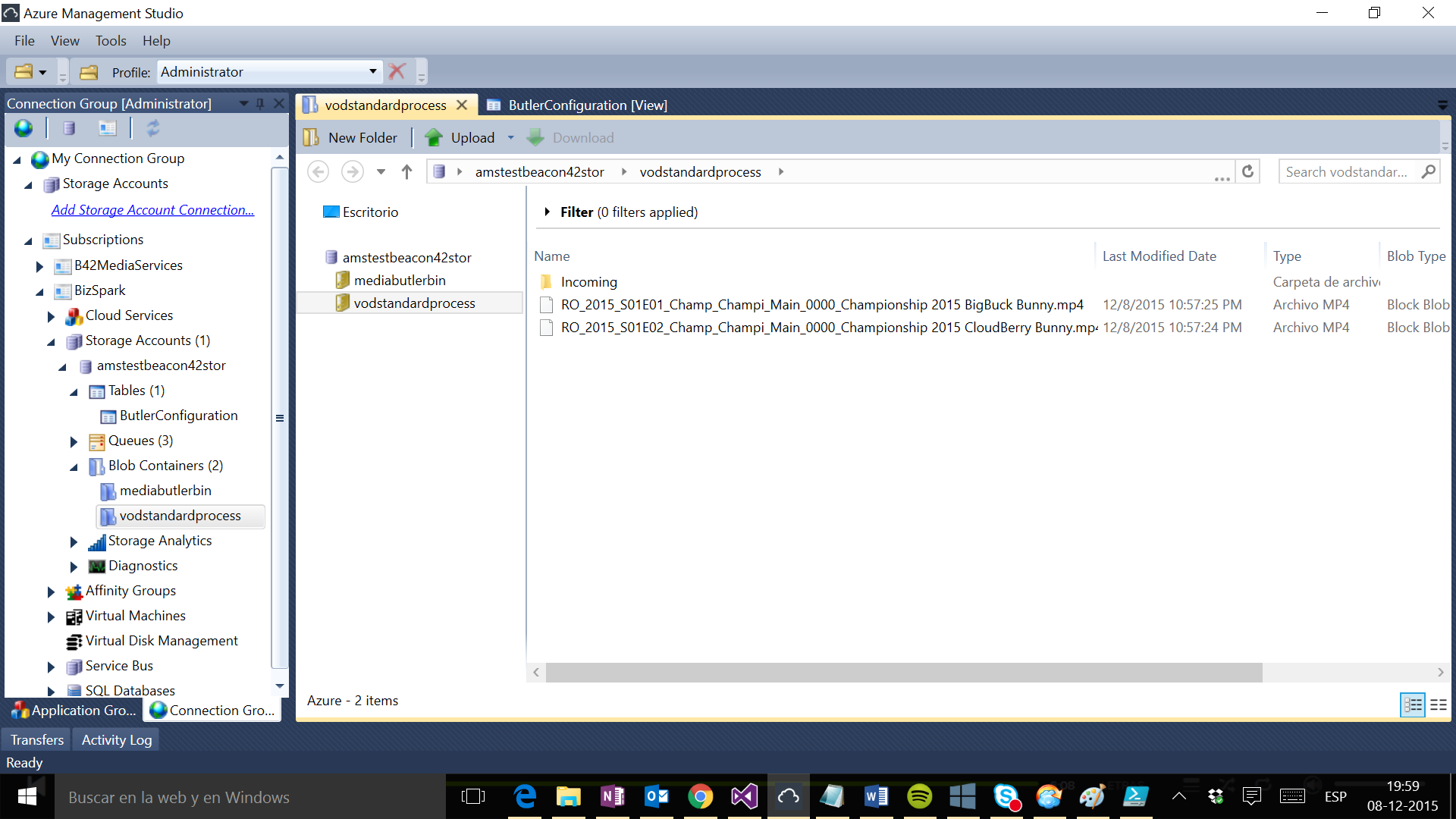
Find the webjob on your subscription and make sure “Always On” is selected for the application.



# Test Solution

Go your favorite Azure Storage Manager.

Connect to the Ams Test Storage and on the vodstandardprocess container create an Incoming folder.



Drop one of the provided files on MediaSamples directory to the Incoming folder.