**How to Pause and Resume Azure SQL DW automatically by using Azure Automation**

**Or**

**Using Azure Automation to Pause and Resume Azure SQL DW automatically**

Azure SQL Data Warehouse is a SQL based, fully Microsoft managed (PaaS) and petabyte scalable data warehouse solution. The service can be provisioned in minutes and scale in seconds by working with up to hundreds of nodes in parallel to process your data. You can learn more about Azure SQL DW [here](https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-overview-what-is).

Azure SQL DW has many integration with Azure services like Azure SQL DB, Stream Analytics, Power BI and others but one of the greatest benefits of the service is in its design. Azure SQL DW has the processing part decoupled from storage. So, in that way, Azure allows you to pause the processing part of the service and keep data available in the storage part. By pausing the service you don´t need to pay for the time you have the service paused and you can resume the processing service in a few minutes and continue processing on your data warehouse. That is awesome!

The service can be paused by using Azure Portal or executing a Powershell cmdlet. But in both ways your Administrator will need to interact with the system during the execution of the pause and resume commands.

So, how to automate those tasks in any time during the day without Administrator has to be in front a desktop or laptop to “push” the button?

The answer is: **Azure Automation** with **Powershell** script!

We are going to use Powershell script in the form of a Runbook to automate those tasks by using Azure automation with a few lines of code and that is what we are going to see in this tutorial.

So, Let´s start it!

For this tutorial I am assuming that you already have:

* an Azure subscription
* the Azure SQL Data warehouse service running with at least one database

In a nutshell these will be the steps that we are going to follow:

1. create an Azure Automation account
2. Create a new Azure AD Directory (in case you are using a Microsoft account to log in to Azure)
3. Add the new user created to your subscription
4. import the Runbook to Automation account
5. edit the Runbook and publish it
6. add a Credential in Assets
7. configure the Schedule to run the Runbook to pause and resume the Azure SQL DW

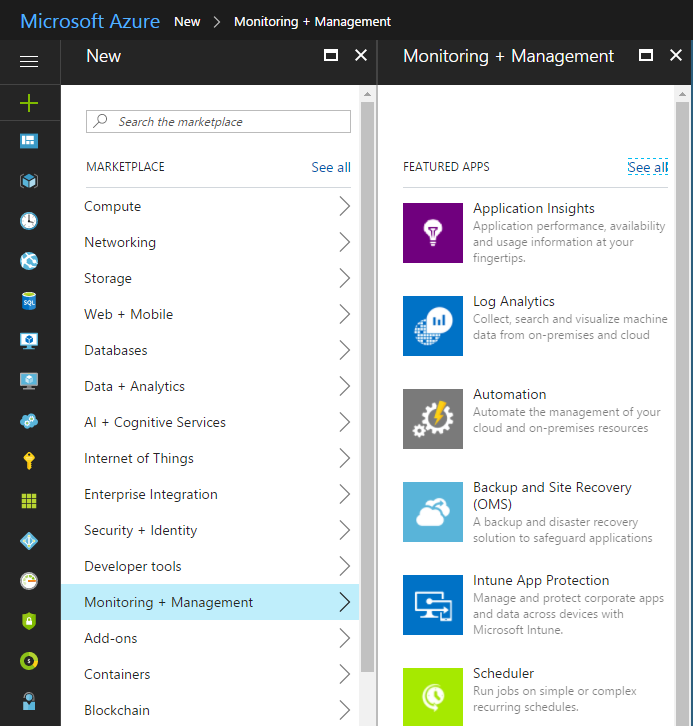
**Remind:** if you are using an user that is not a Microsoft account, you will not need to execute the steps 2 and 3.

**Step 1 - create an Azure Automation account**

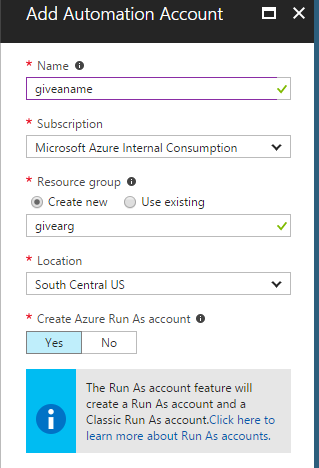
Azure Automation allows you to automate the manual, long-running and frequently repeated tasks that you usually have to perform in the cloud or even in your on-premises environment. As the service is a PaaS, you don´t need to worry if your code will run or not, like when you are using a Windows Task Scheduler in a single computer that may be out of service anytime, for example. More information about Azure Automation can be found [here](https://docs.microsoft.com/en-us/azure/automation/automation-intro).

The service is created very easily with the steps below:

1. in the Azure portal you might click in the “+” signal in the left upper corner and navigate as the picture below.



1. Fill out the fields as shown in the picture below.



Keep the “Create Azure Run As account” with the default option.

In a few minutes the service will be ready to use.

**Step 2 – Create a new Azure AD Directory**

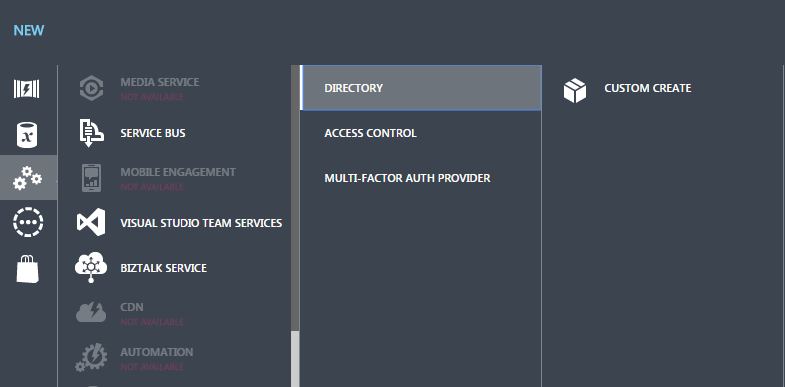
The reason to create a new Directory in Azure AD is because Azure will not allow the script to login in a non-interactvely way with a Microsoft account. So with the new Directory you will create a new user that is a non-Microsoft account to be used later with the automated script.

So **Remind** that if you are already using a non-Microsoft account user, YOU WILL NOT NEED to execute the Step 2 and 4.

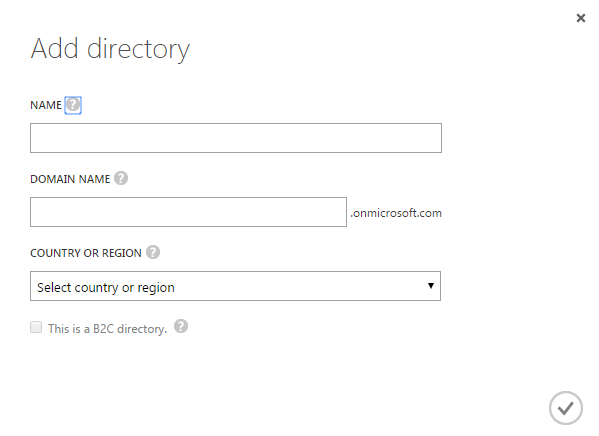
1. Go to the URL manage.windowsazure.com

Currently activities with Directories work better in the classic Portal.

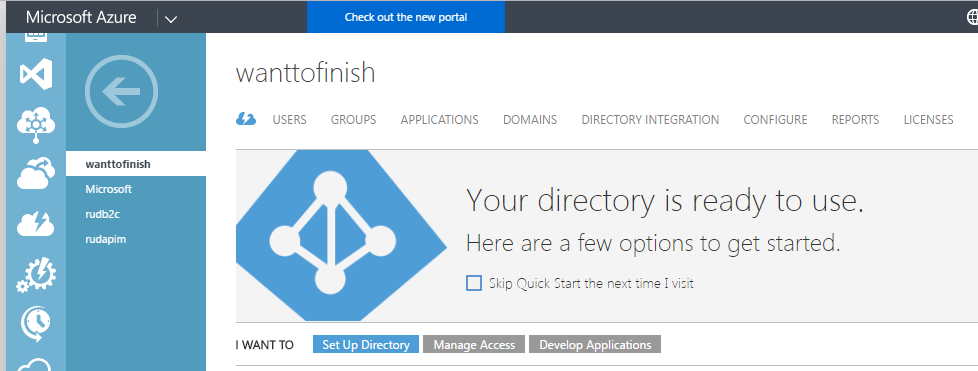
1. After login, scroll down until “Active Directory” in the left pane and click on it
2. Click in “New” in the left down corner, then in “directory” and then “custom directory”



1. In the pop up window fill out the fields as shown in the picture below and click in the green signal when you are done.



1. the new directory created will appear in the list. Click on it.
2. Then click in “users” to add a new user

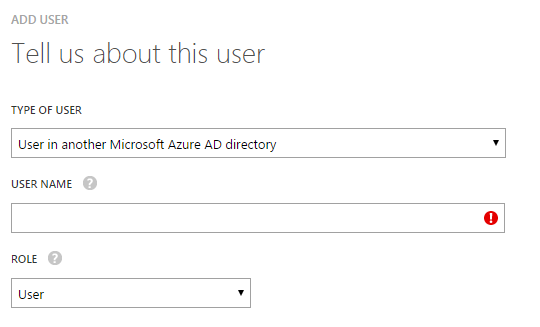


1. Then click in “add new user” button in the bottom of the screen.
2. Create a new user by filling out all fields. In the end take note of the temporary password provided.
3. Go back to the Directory screen, click in your “Default Directory” and then in “users”. Add the user you created in the other Directory. To do that you should use in the “type of user” the option “User in another Microsoft Azure AD directory”
4. In the field “user name” type the [user@domain.onmicrosoft.com](mailto:user@domain.onmicrosoft.com) (unless you configure a custom domain).

The field will not auto-complete your user name, you will have to type entirely.

**Remind** :

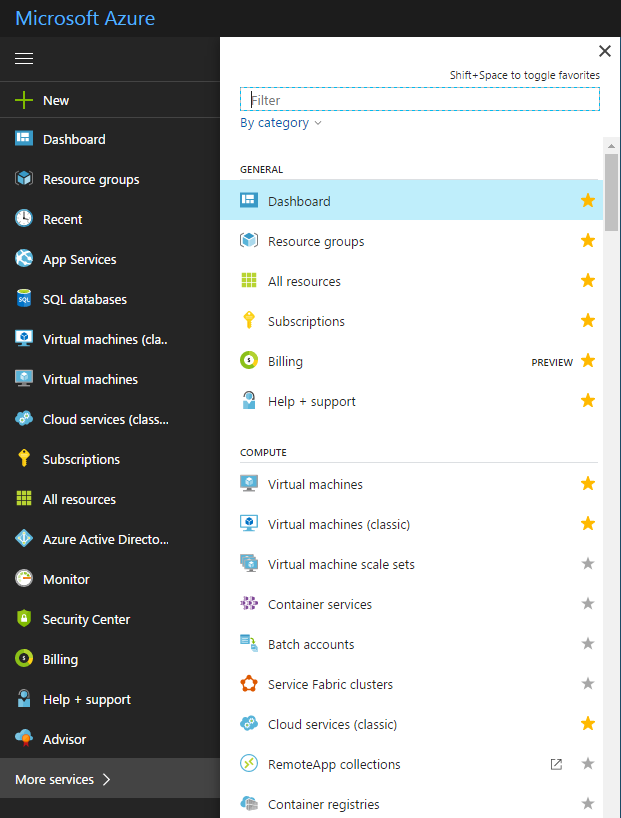
the field “Role” is regarding to the Azure Directory and not to the subscription that uses different roles.



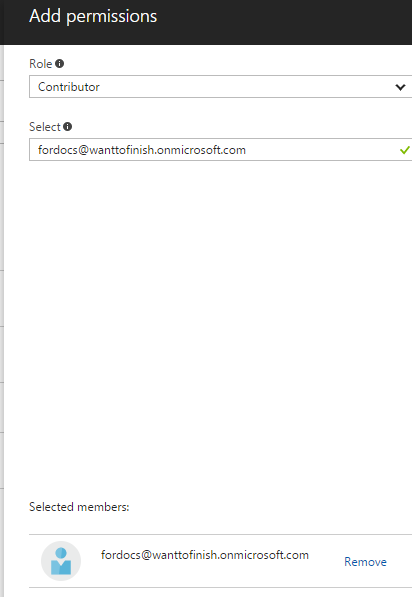
**Step 3 – Add the new user created to the subscription**

Remind that we created a new user in a new Azure Directory. This user doesn´t have any access to the subscription yet, because of that the next step is add the user to access the subscription that is running your Azure SQL DW.

1. Go to portal.azure.com
2. After login, go to “more services” and click on “subscriptions”



1. Click on the subscription that is going to appear in the list and then in “Access control (IAM)”
2. On the right side click on “Add” button and in the blade “Add permission” choose the role for you’re the user that you are going to select. Remind that this user is the one we created in the last step.



**Tip:**

It is important you login with this user at least once in the portal before you use it in the Automation account because this new user is still with a provisory password. When you login for the first time you will define your permanent password.

**Step 4 – import the Runbook to Automation Account**

Runbook is a structure of codes that uses Powershell cmdlets inside it to execute one or more tasks. It runs inside Azure Automation. There are some types of Runbooks and that will be important for you understand how each type works when you have to define the “runbook type” field in the blade configuration as you will see later.

The Runbook used in this tutorial is a “powershell workflow” type.

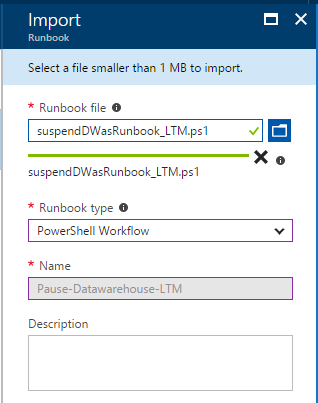
Check it out all types and their descriptions [here](https://docs.microsoft.com/en-us/azure/automation/automation-runbook-types).

Now, to import the Runbook, follow these steps:

1. Come back to the Automation account screen in the Azure Portal
2. In the main screen, click in “Runbooks” and then in “Add a runbook”.

You will notice that the service already comes with 4 runbooks. It is all about tutorial.

1. In the next blade called “Add Runbook” click in “Import an existing runbook”.
2. So in the blade called “Import” you will upload the first script that we are going to use to pause the Data warehouse and then click in the “create” button.



Before to import the runbook file you will create it by copying and pasting the code below in a notepad file and then save in your local folder with the extension .ps1.

**Important:**

For this Runbook the type might be a “PowerShell Workflow”.

**Pause Data warehoue**

workflow Pause-Datawarehouse

{

$CredentialName = "SQLDW-Cred"

#Get the credential with the above name from the Automation Asset store

$psCred = Get-AutomationPSCredential -Name $CredentialName

if(!$psCred) {

Throw "Could not find an Automation Credential Asset named '${CredentialName}'. Make sure you have created one in this Automation Account."

}

#Login using the above Credential

Login-AzureRmAccount -Credential $psCred

#Use those 2 lines below if your user has more than one subscription

#Get-AzureRmSubscription

#Select-AzureRmSubscription -SubscriptionName "<subscription\_name>"

#pausing datawarehouse

Suspend-AzureRmSqlDatabase –ResourceGroupName "<resource\_group\_name>" –ServerName "<server\_name\_without\_sufix>" –DatabaseName "<database\_name>"

}

Here is the second script that will be used to resume the service.

**Resume Data warehouse**

workflow Resume-Datawarehouse

{

$CredentialName = "SQLDW-Cred"

#Get the credential with the above name from the Automation Asset store

$psCred = Get-AutomationPSCredential -Name $CredentialName

if(!$psCred) {

Throw "Could not find an Automation Credential Asset named '${CredentialName}'. Make sure you have created one in this Automation Account."

}

#Login using the above Credential

Login-AzureRmAccount -Credential $psCred

#Use those 2 lines below if your user has more than one subscription

#Get-AzureRmSubscription

#Select-AzureRmSubscription -SubscriptionName "<subscription\_name>"

#reiniciando o datawarehouse

Resume-AzureRmSqlDatabase –ResourceGroupName "<resource\_group\_name>" –ServerName "<server\_name\_without\_sufix>" –DatabaseName "<database\_name>"

}

You can also find those scripts above on github:

Pausing the database:

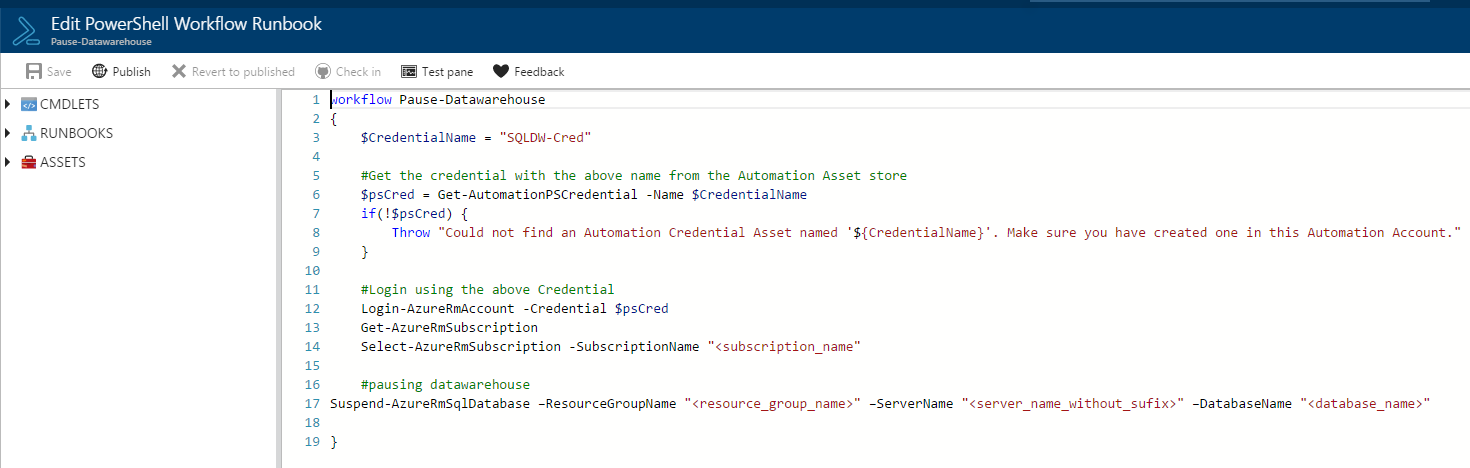
<https://github.com/ruolivei/pauseDWpsscript>

Resuming the database:

<https://github.com/ruolivei/resumeDWpsscript>

**Step 5 – edit the Runbook and publish it**

1. In the next screen after you have imported the Runbook you will see it in the list as “new”. Click on it.
2. Now click in “edit” and you will have your script to be edited.



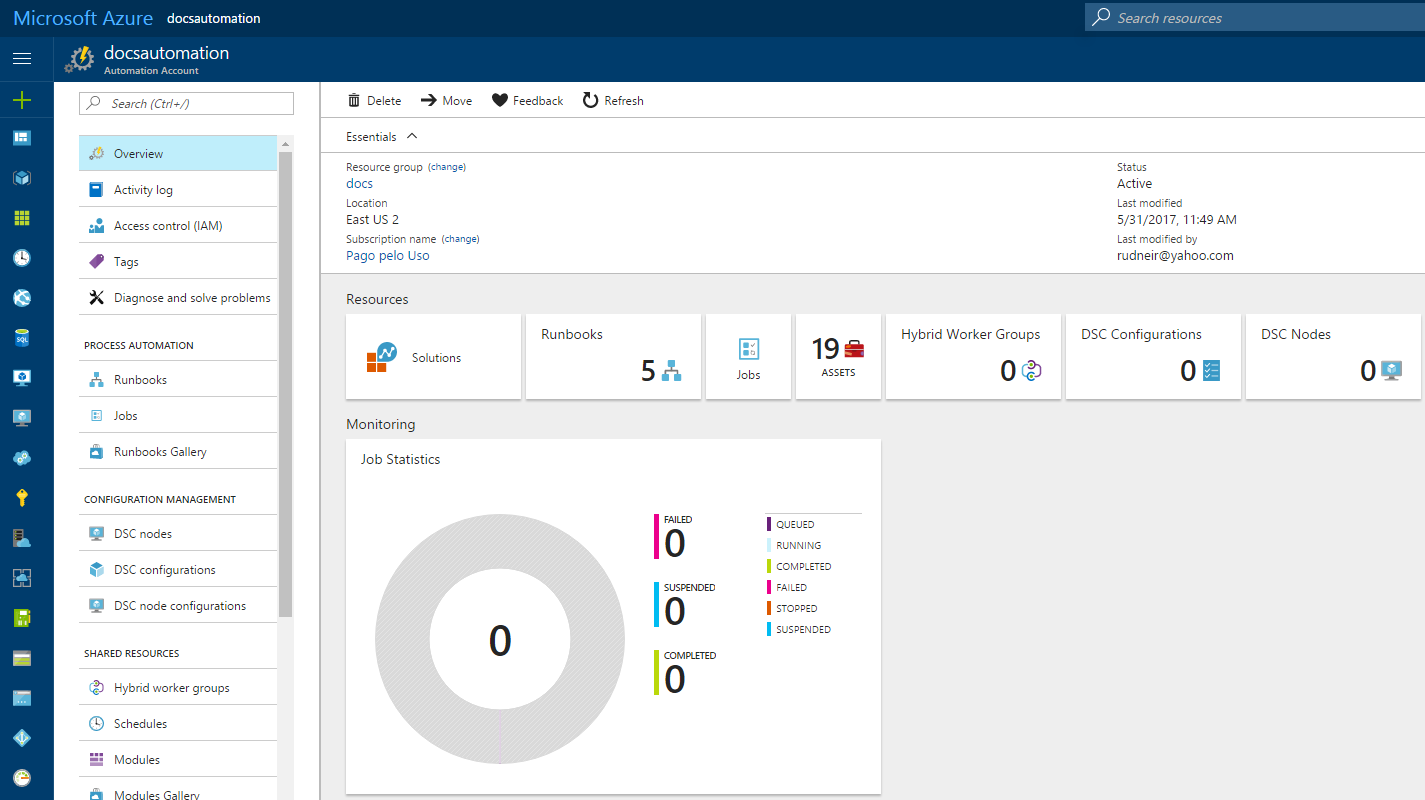
First take note of the value for $CredentialName. We will use it later to define the Credential. The name that you see here is just a suggestion, you can define your own name.

Edit the last line with the resource group, server name and database name that you are using.

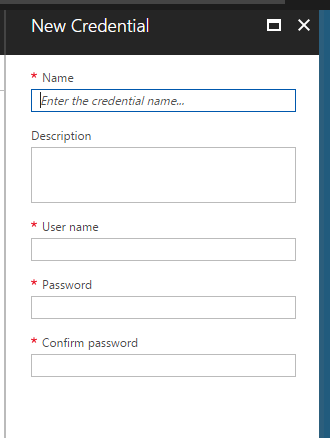
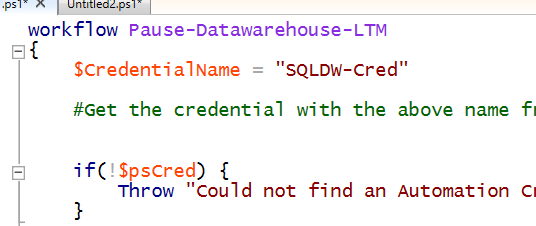
1. After editing the script click in “save” and then “publish”.

**Step 6 – add the Credentials**

* 1. Now go back to the Automation Account main screen and click in “Assets”



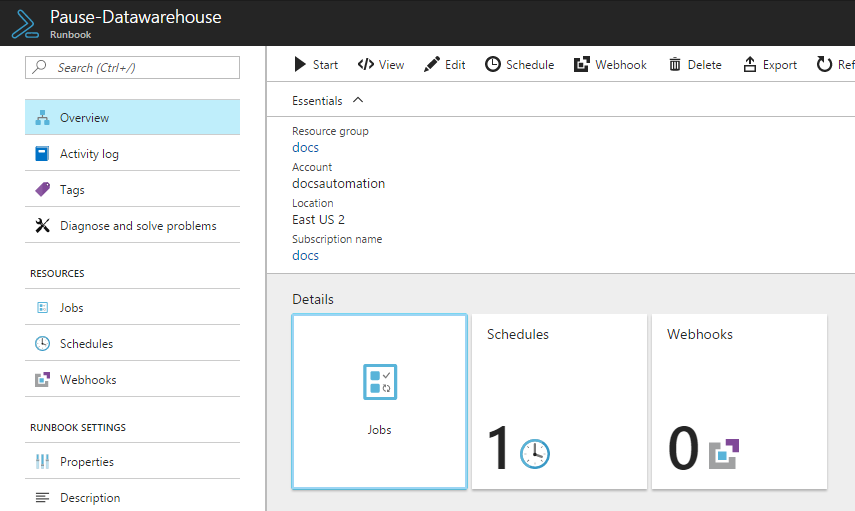
* 1. Then click in “Credentials” and “Add a credential”
  2. In the screen below the first field called “Name” is the most importat. It should match the variable name ($CredentialName) in the script.



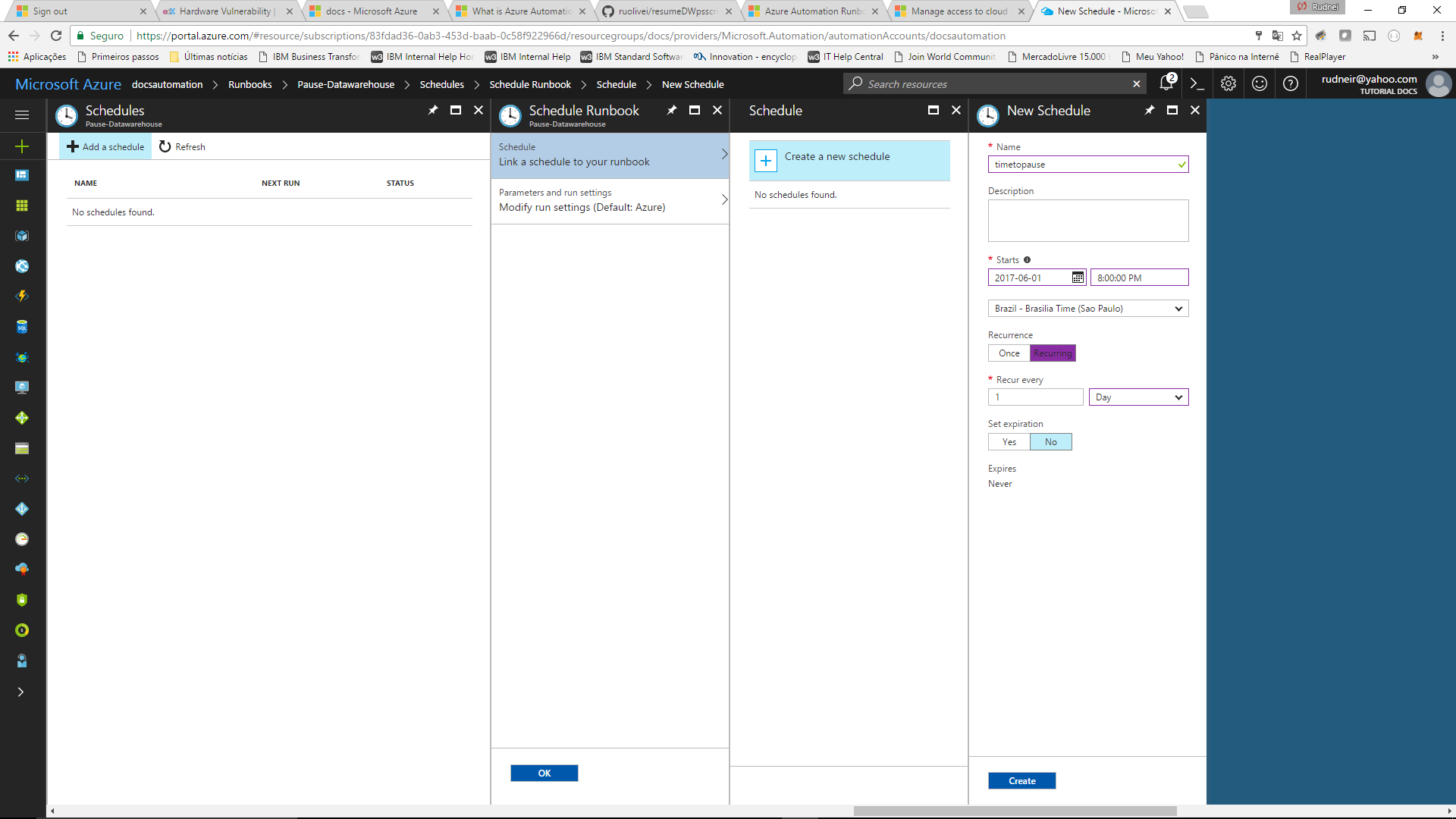
The fields “user name” and “Password” might contain the user that we created in the new Directory.

**Step 7- configure the Schedule to run the Runbook to pause and resume the Azure SQL DW**

* 1. Go to the Automation Account and click in “Runbook”
  2. Select the Runbook you want to schedule, in our example, we are going to click on the “pause…” script
  3. You will see a button called “Schedule”. Click on it.



* 1. In the Schedule screen you will navigate as the picture below. Then in the blade called “New Schedule” you will fill out all the fields according to the time and frequency you want your script run then click in the “Create” button.



Now your script is ready to run and pause your Azure SQL DW whenever you want with the frequency you need.

You will need to repeat the steps 4, 5 and 7 to add the script to resume your Azure SQL DW.

The step 6 is not necessary because we are going to use the same credential to login on Azure and run the script to resume the Azure SQL DW.

Now you will be able to sleep while Azure works for you ; )