

Source: C# Corner (www.c-sharpcorner.com)

PRINT

Article



Azure Automation - Adding Work Loads For Azure Runbook And Publishing It

By [Abdul Rasheed Feroz Khan](#) on **Dec 06 2016**

Introduction

This article will help you to add work loads for Azure Runbook. To get more knowledge about Azure Automation and what Runbook is, surf for my previous articles.

Note

Before you work on this article, you should have an Azure Runbook, which you have already created.

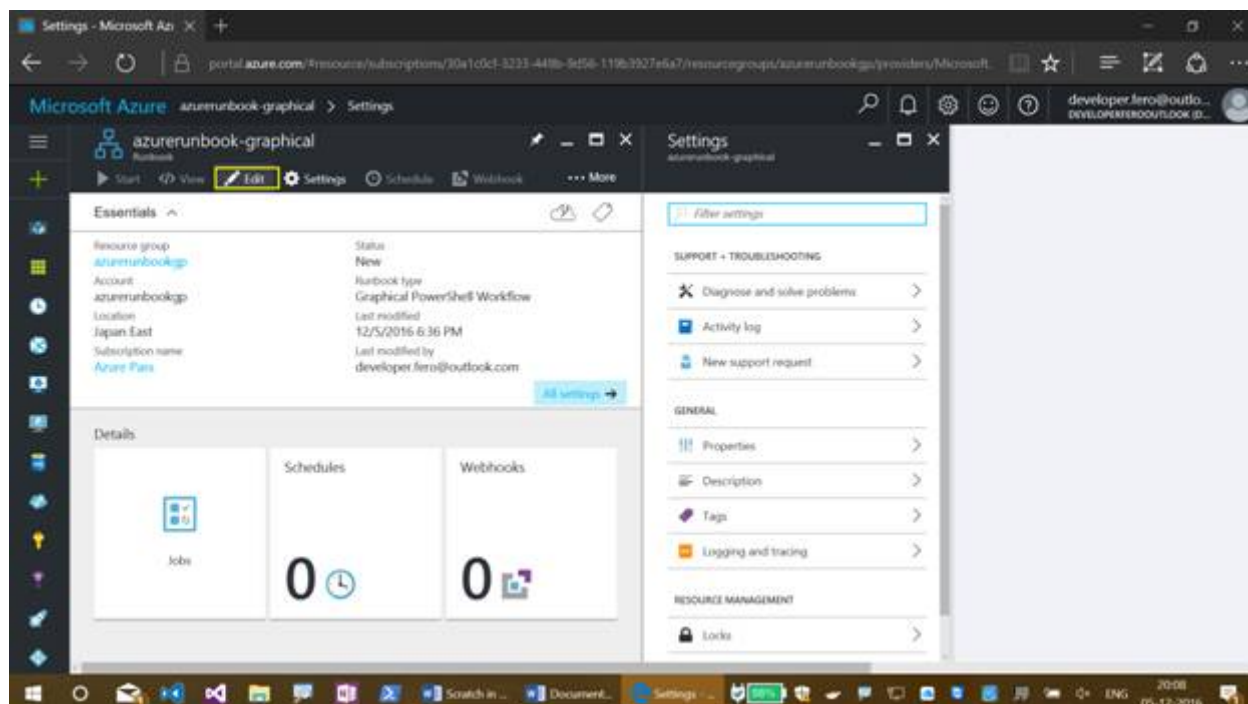
Follow the below steps now

Step 1

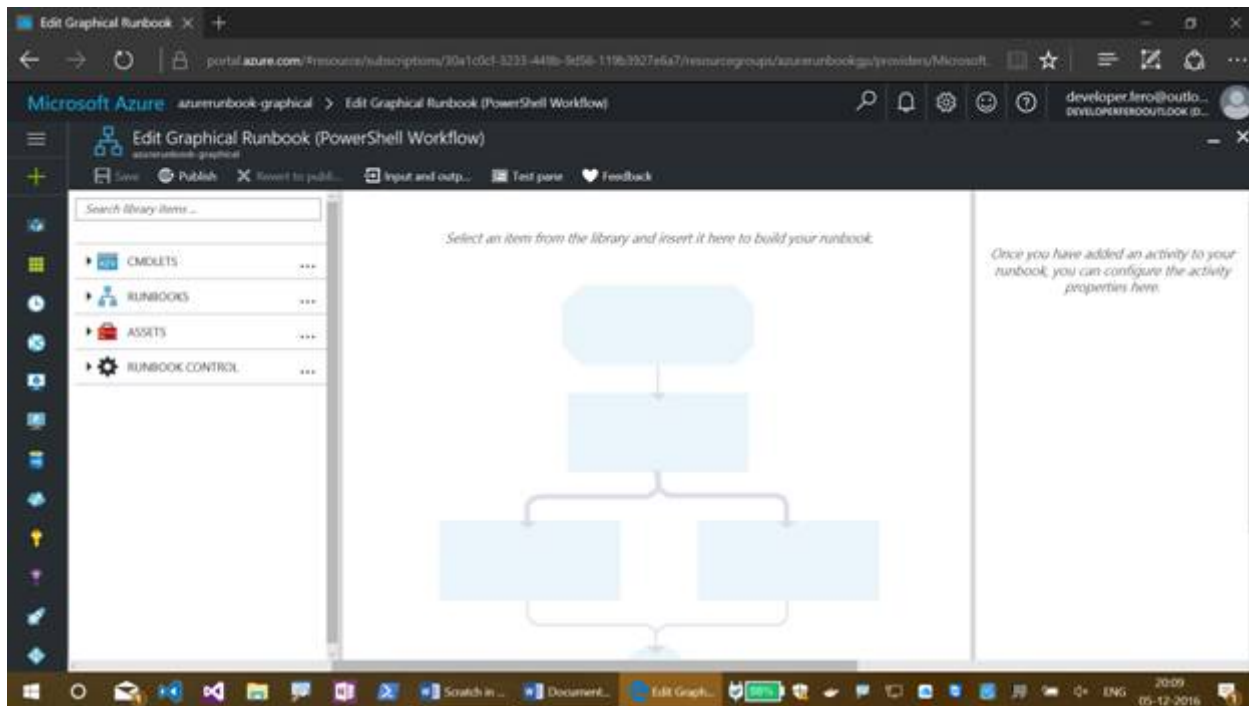
Login to your Azure account and start with Azure graphical runbook. In my case I have already created it, if you have to create a new one, check the previous article of mine.

Step 2

Move for the Library control of the Azure runbook, which has been created. This will help you to add work functions for Azure runbook.

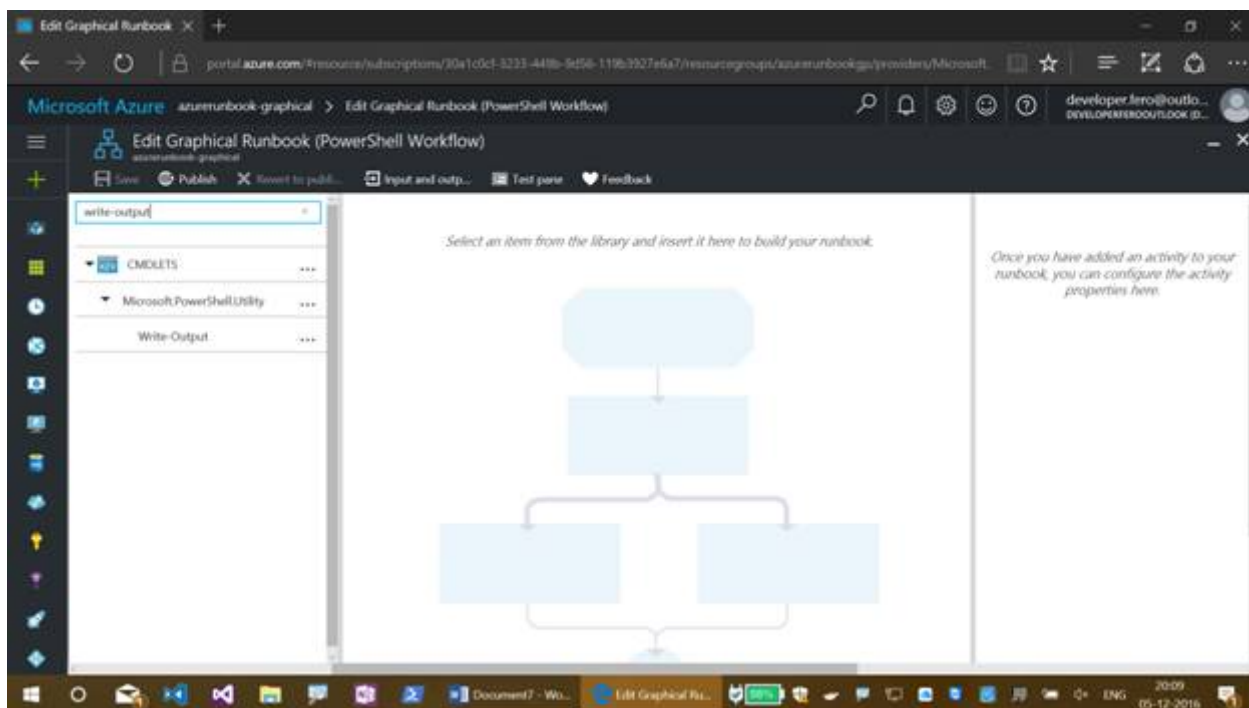


You will be getting your library items here.

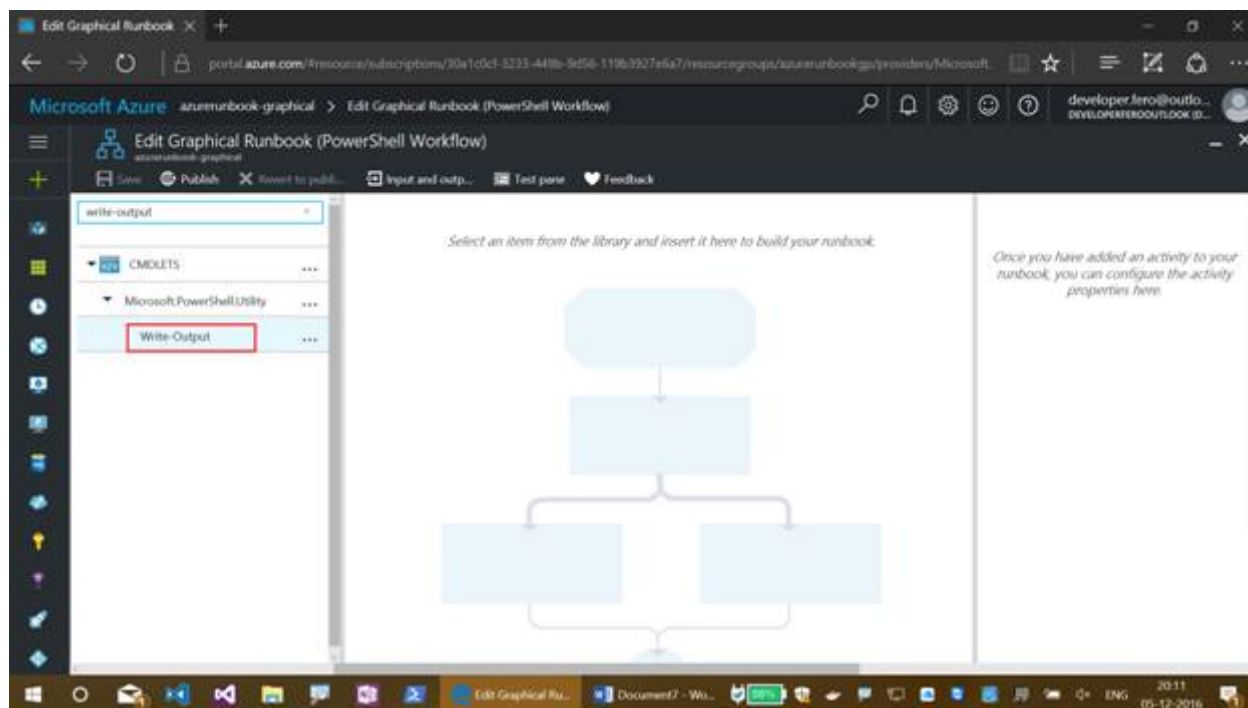


Step 3

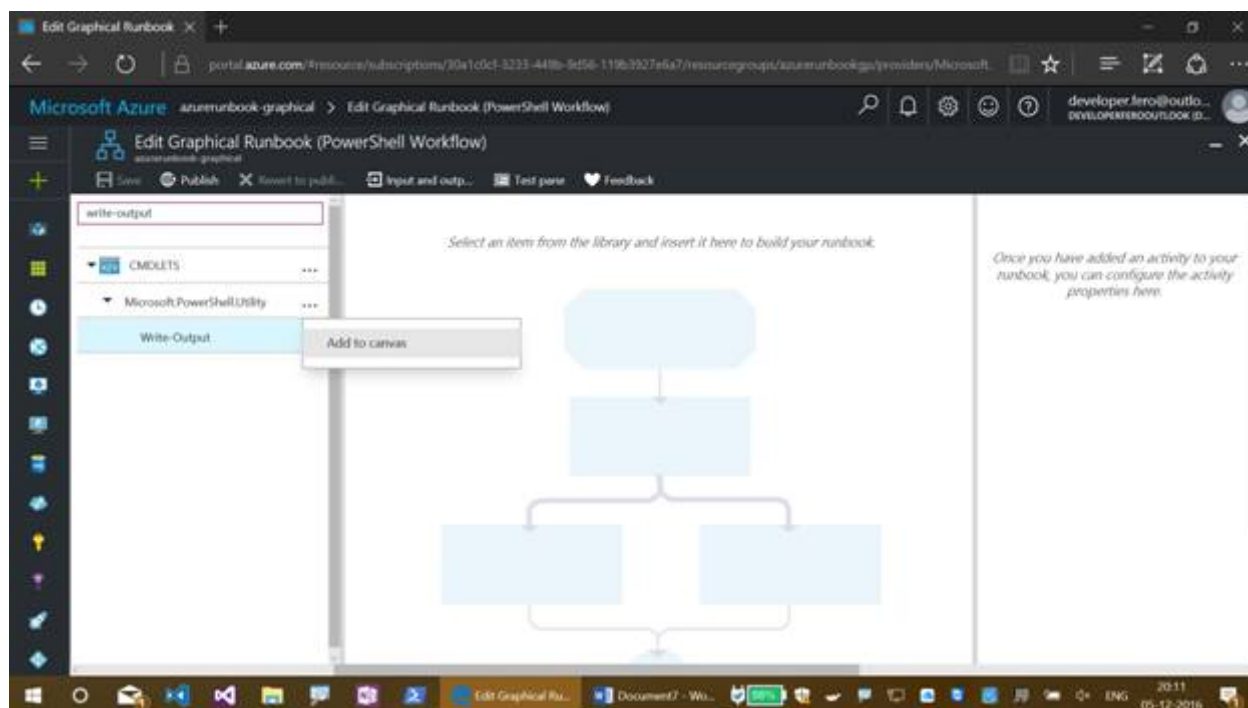
Now, search for write-output in this library items pane.



Click on the Write-Output item here.



Click on Add to Canvas now.

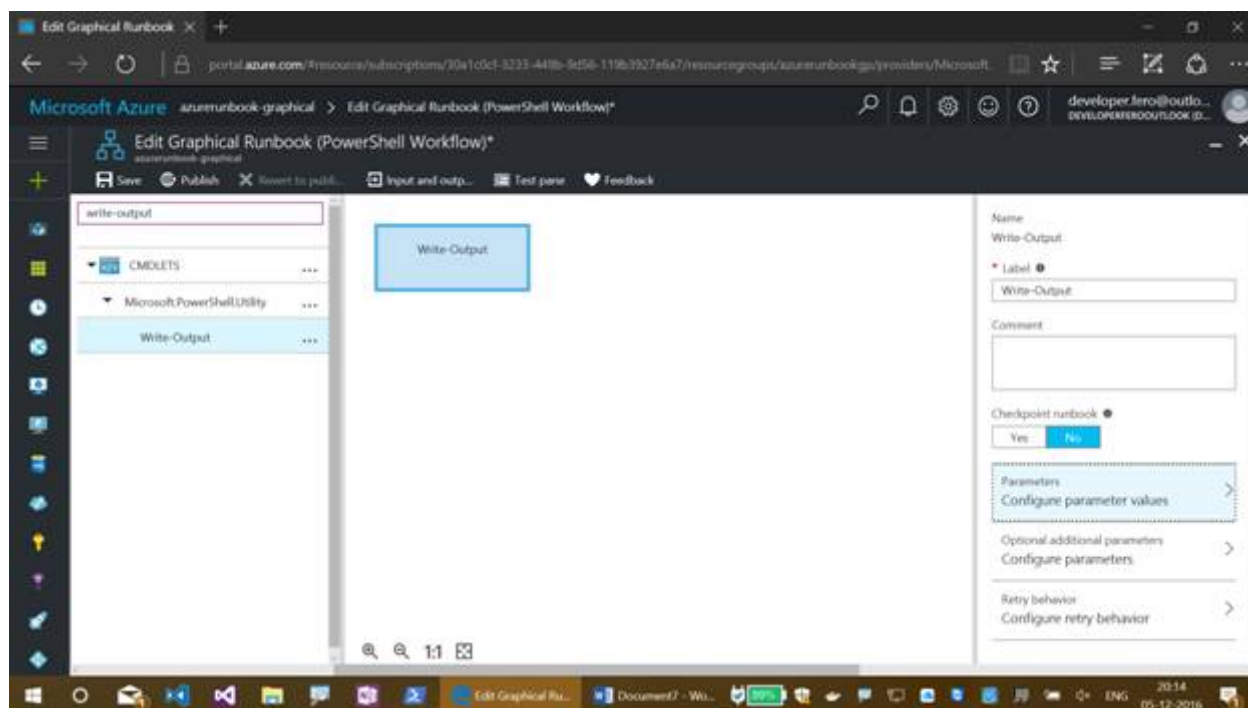


Step 4

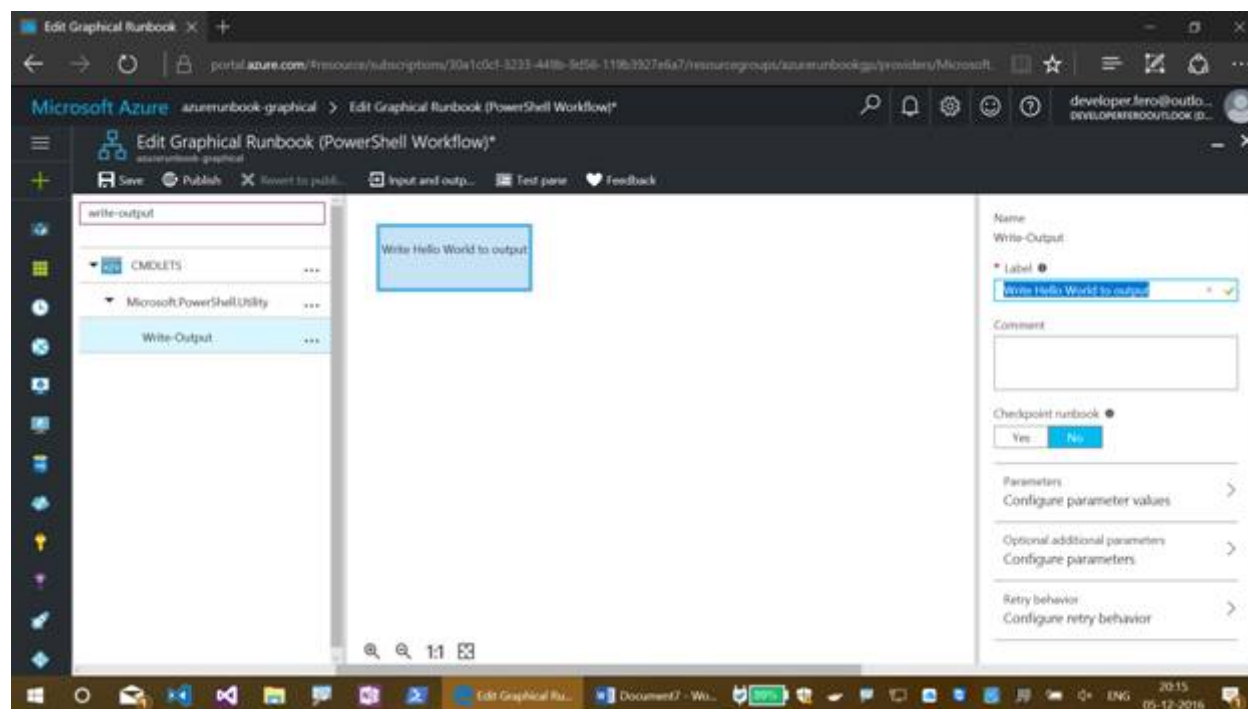
Here, it has been added. Let's work on editing the configuration. You should be configuring for the following details here.

- Name
- Label
- Comment
- Checkpoint for runbook
- Parameters
- Optional additional parameters

- Retry behavior

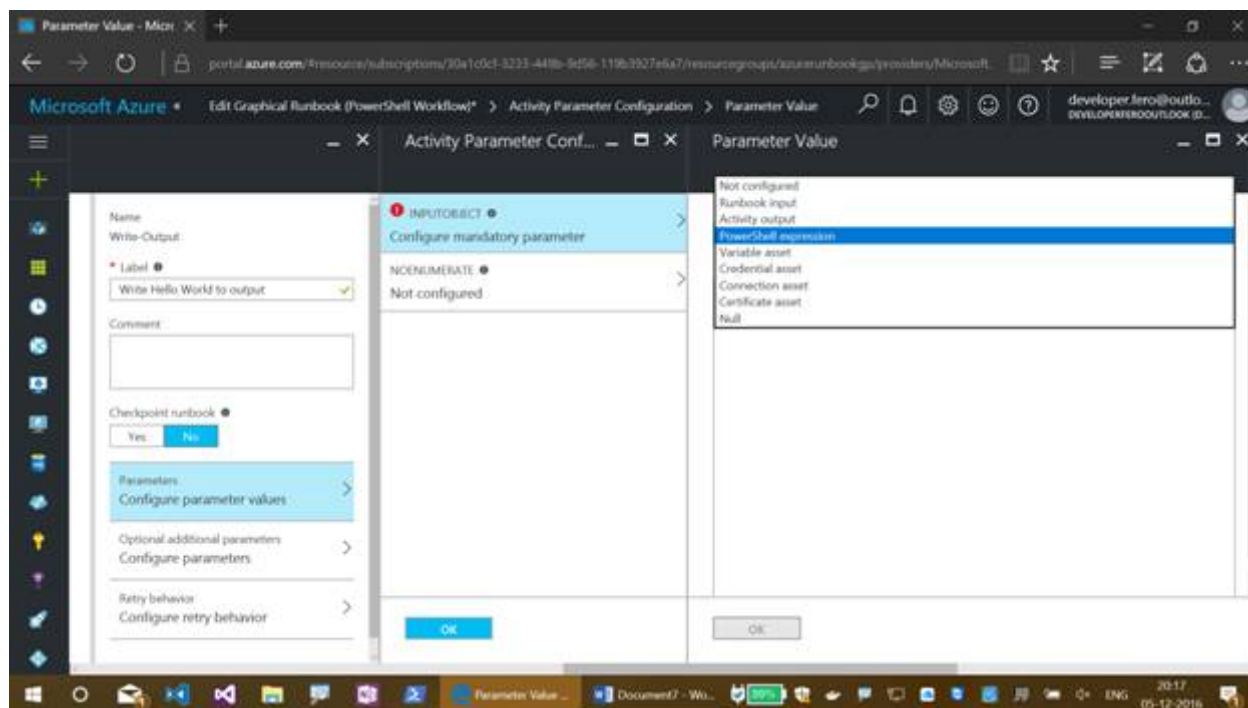
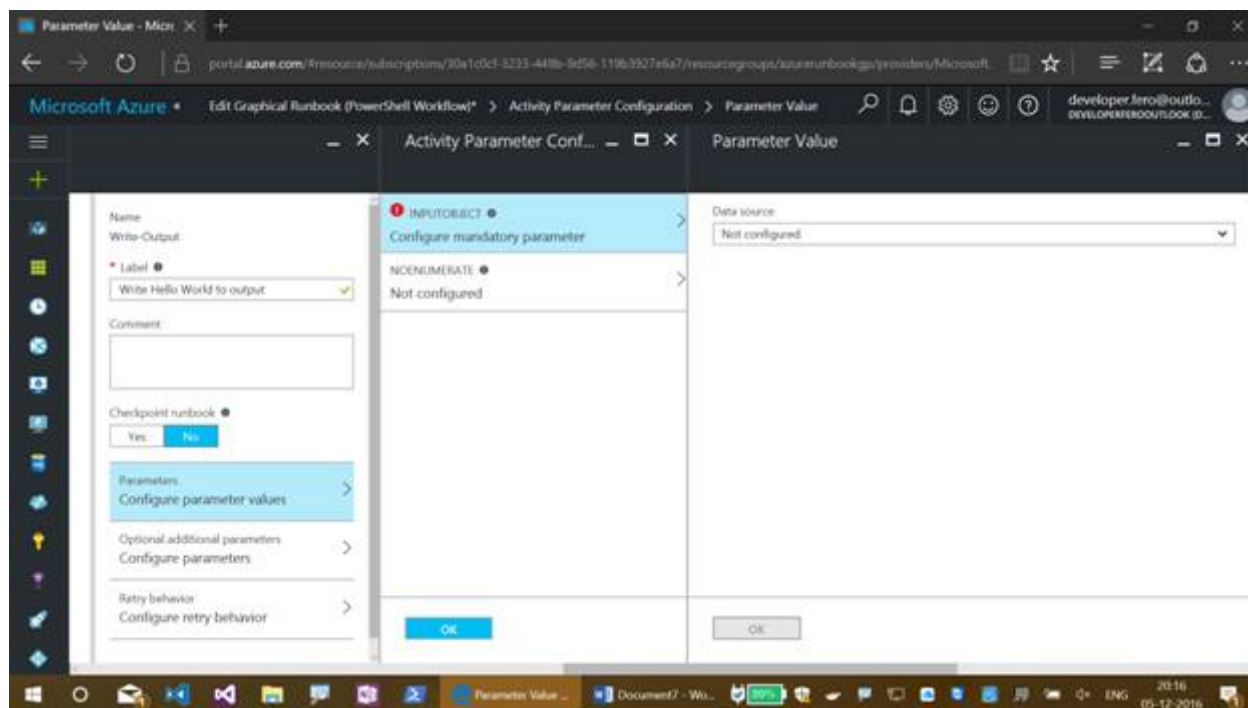


Let's label the name as "Write Hello World to output".



Parameters

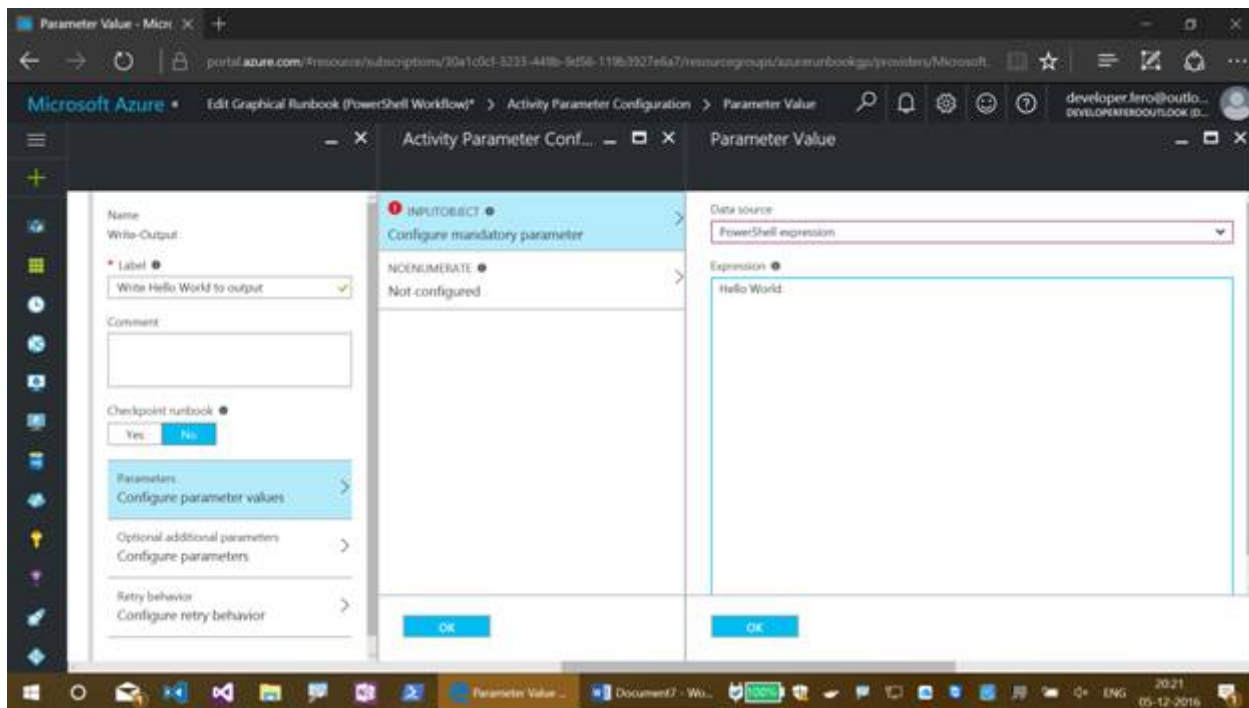
We should configure the parameter values here. Click on parameters, go for inputobject and select PowerShell expression as DataSource.



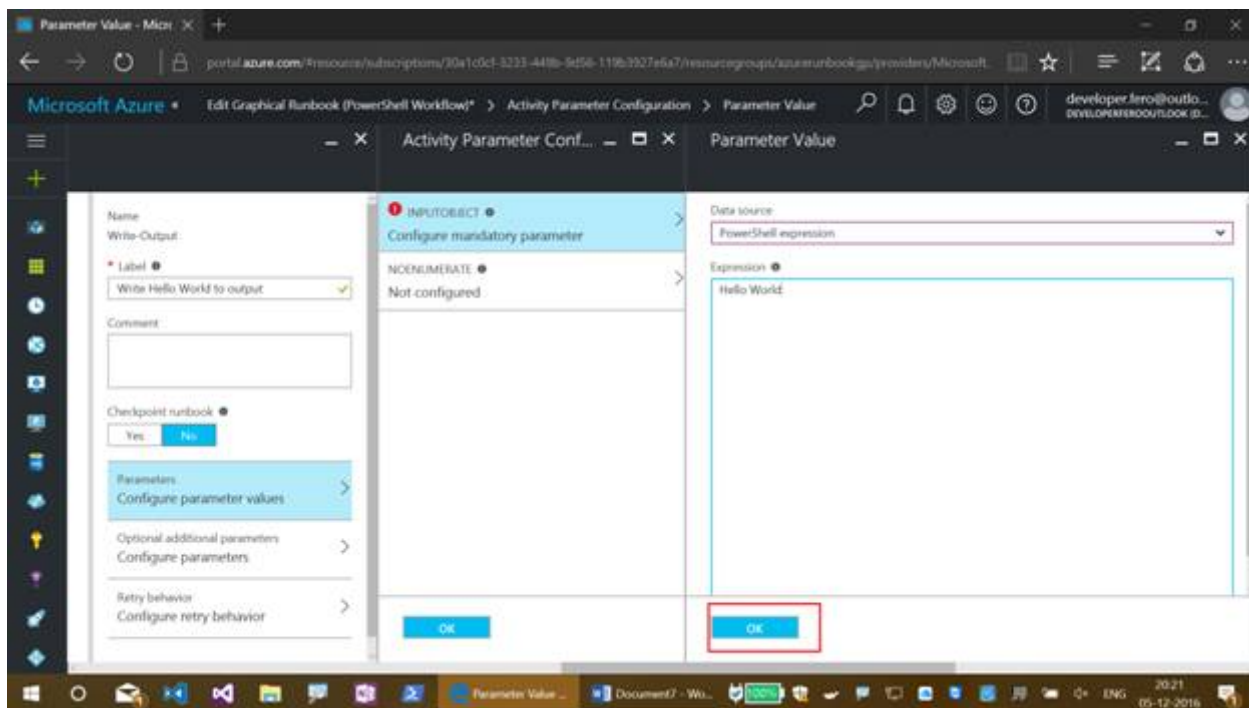
Step 5

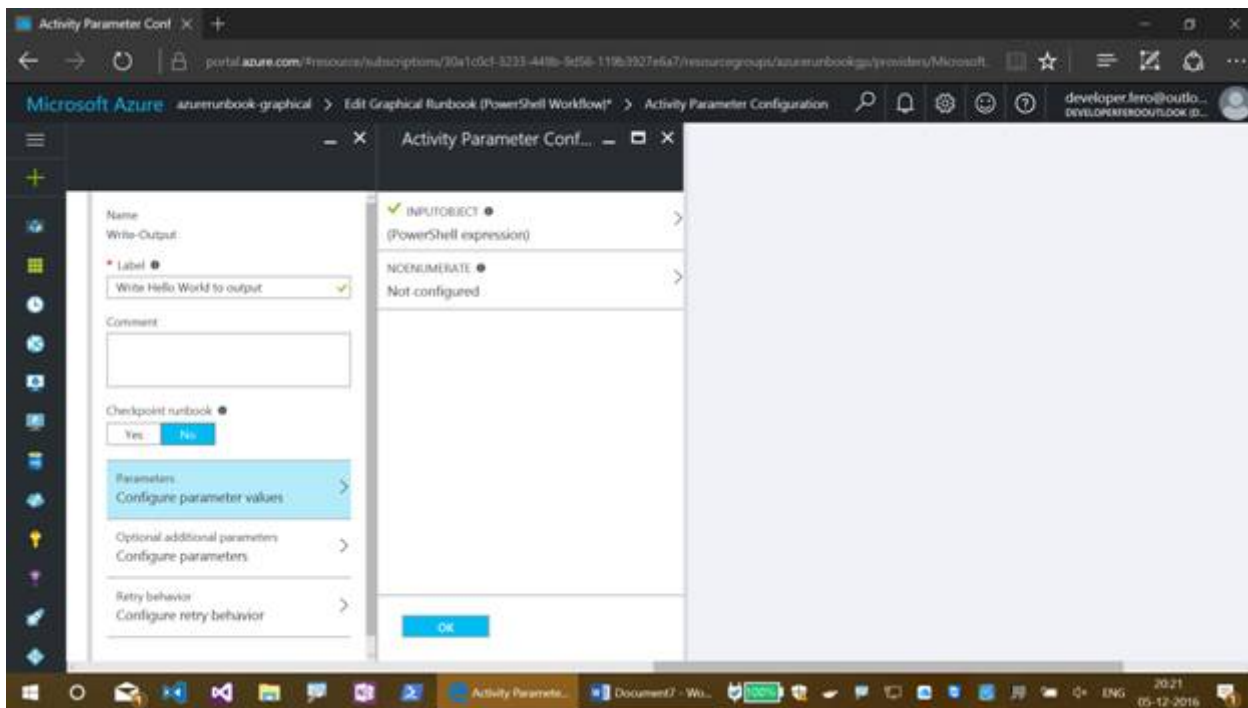
Here, we need an output text as Hello World and hence we are using PowerShell expression over here is added. We will be specifying a string too.

Type the expression as Hello World now.

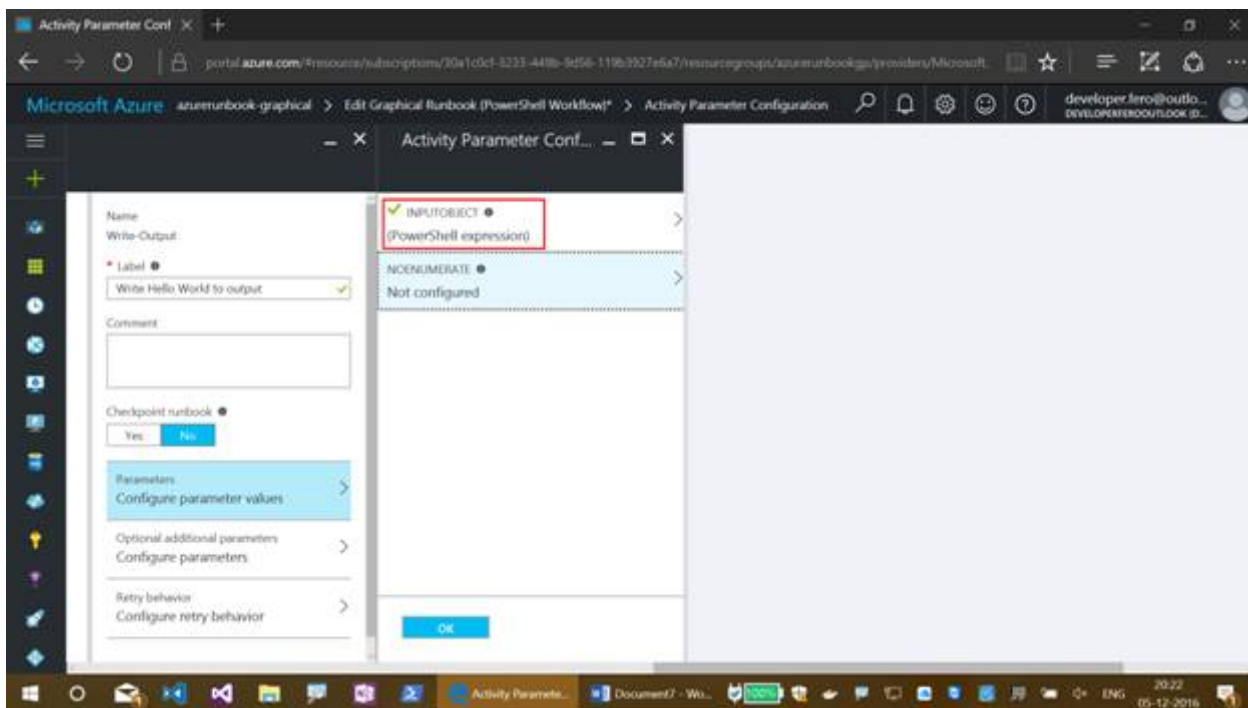


Click OK after adding the expression.



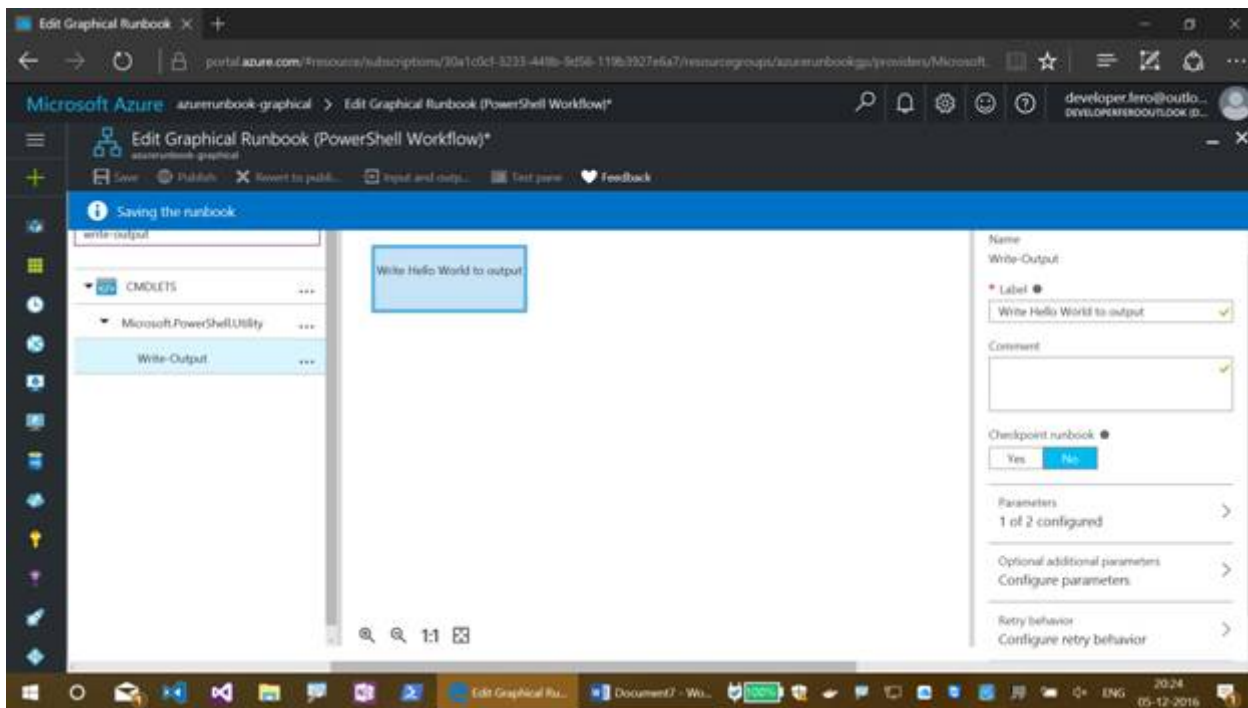
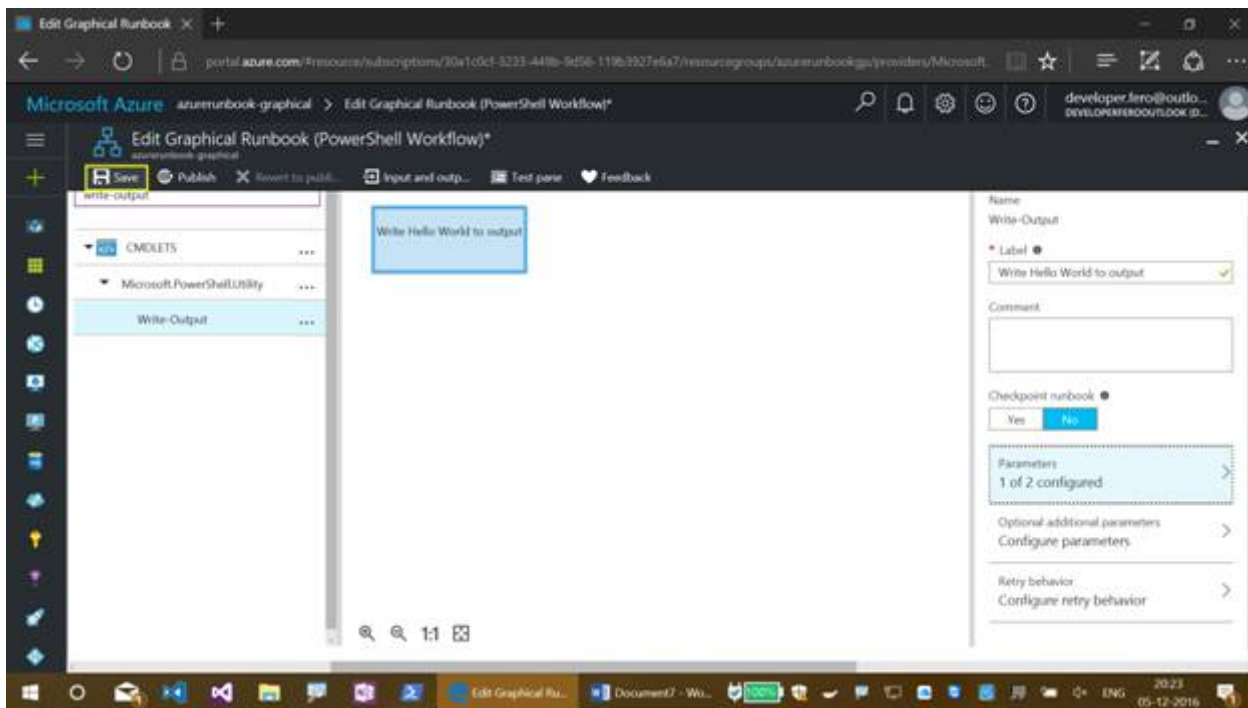


You can find the input object with a Green color and tick mark now with PowerShell expression.



Click OK in this blade.

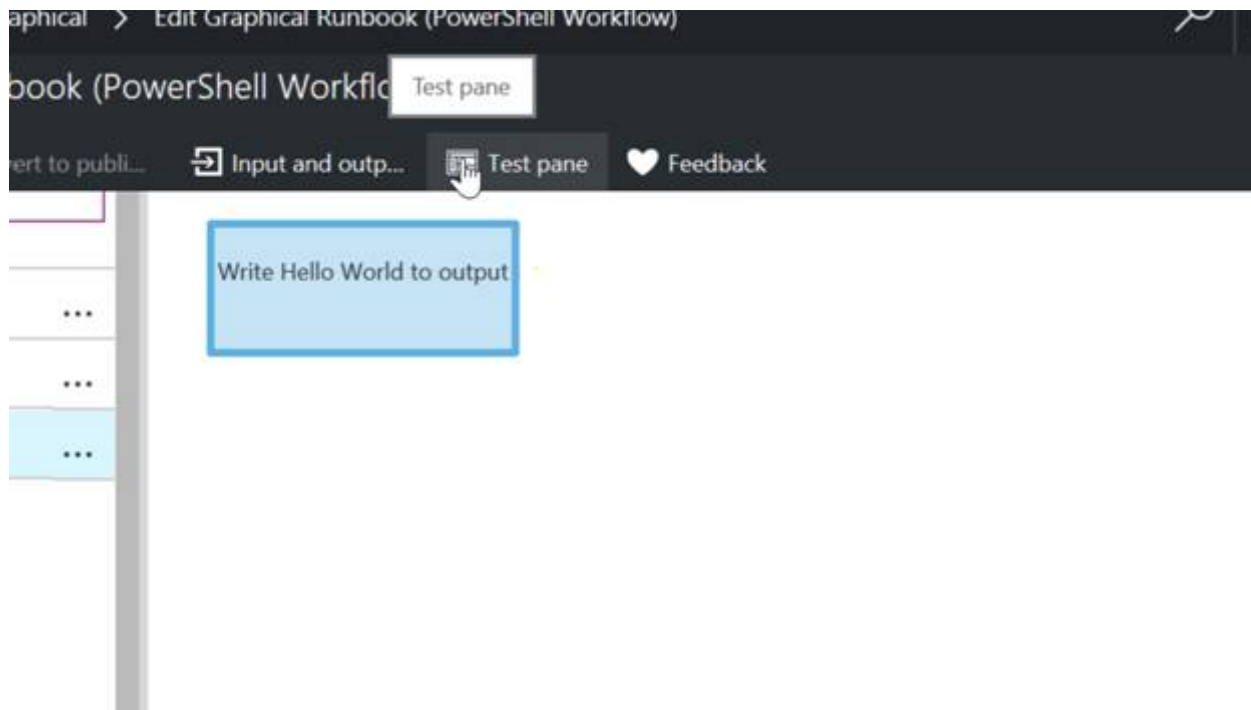
Click Save to save the Graphical runbook of the Azure Automation process.



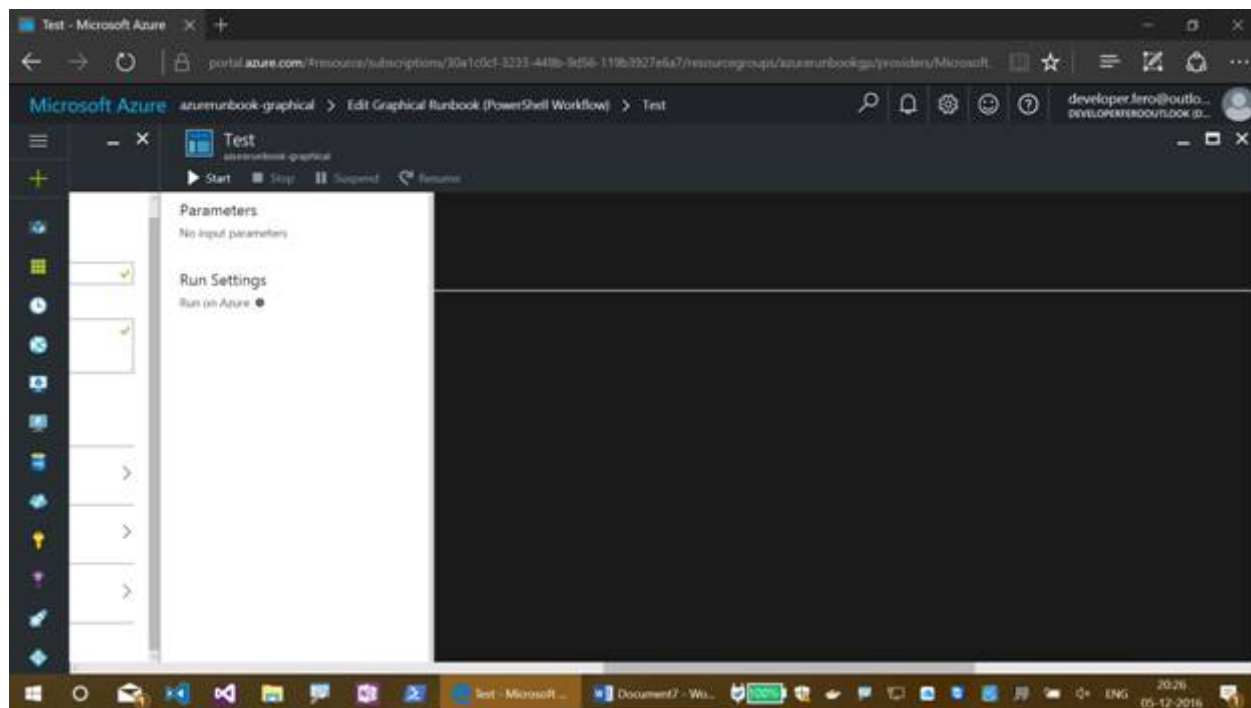
Once your Runbook is saved let's move on for testing.

Testing the Runbook

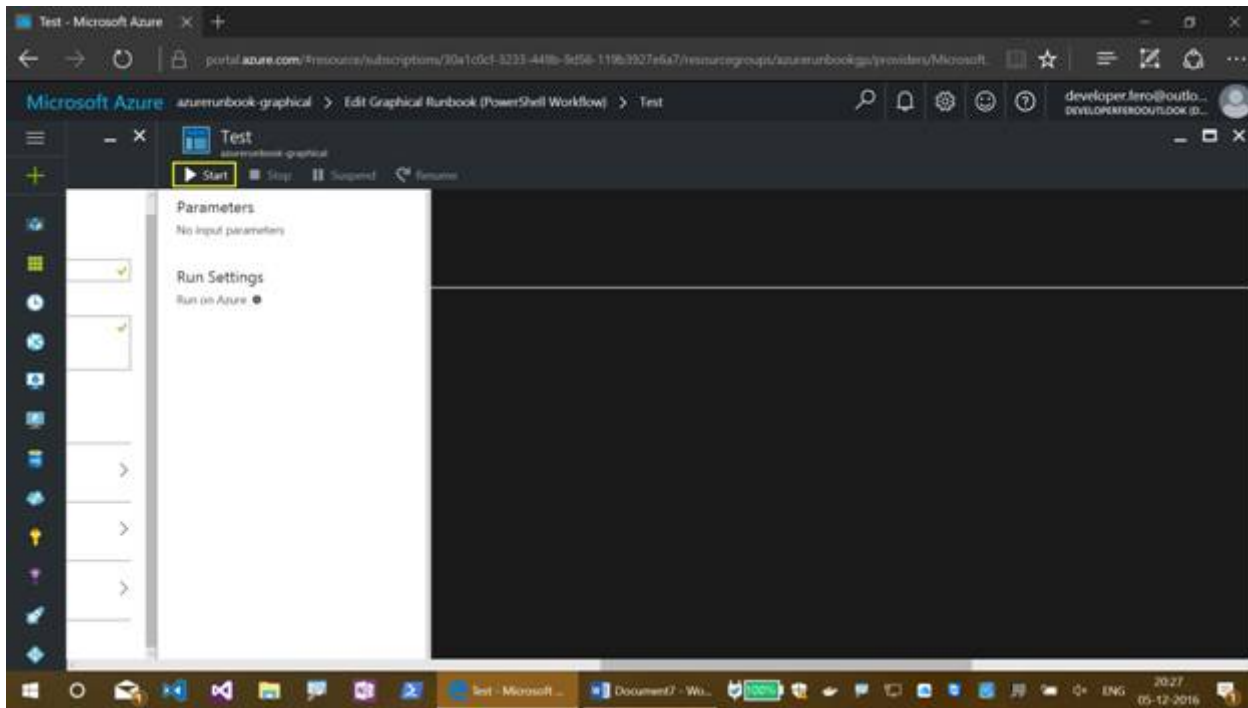
Click Test pane now in your graphical runbook of Azure automation.



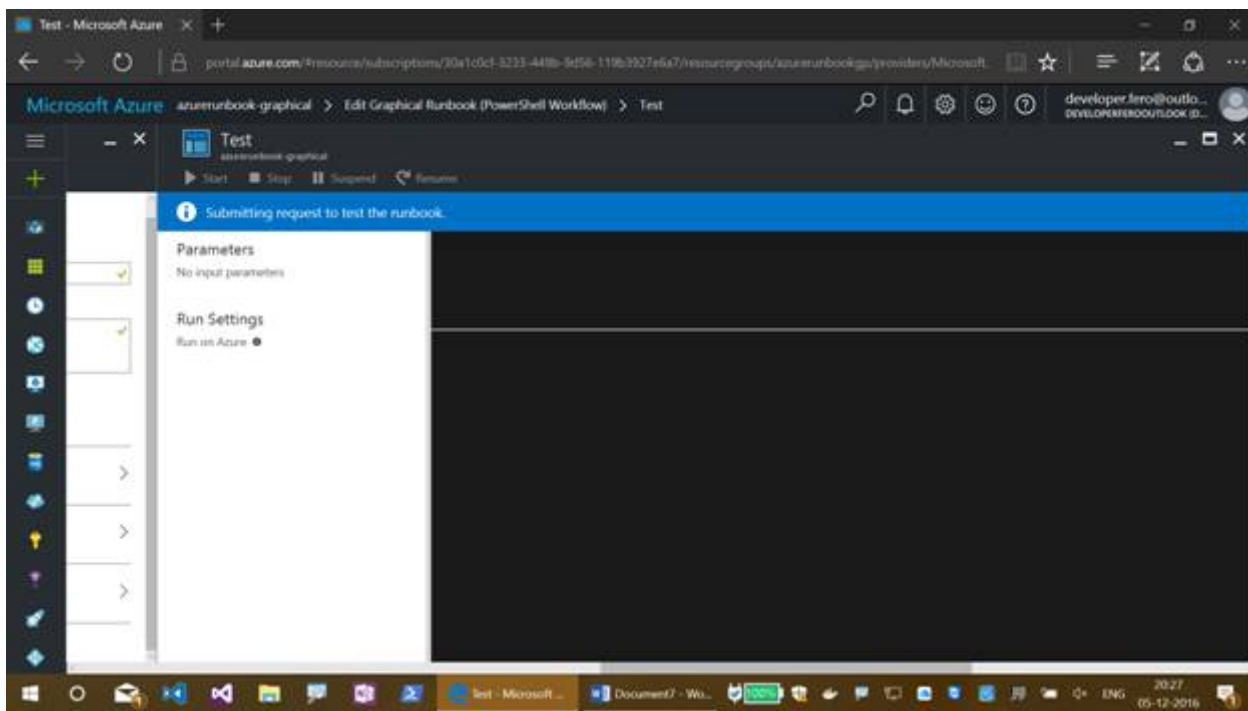
You will be getting the test screen, mentioned below in your Azure graphical runbook.



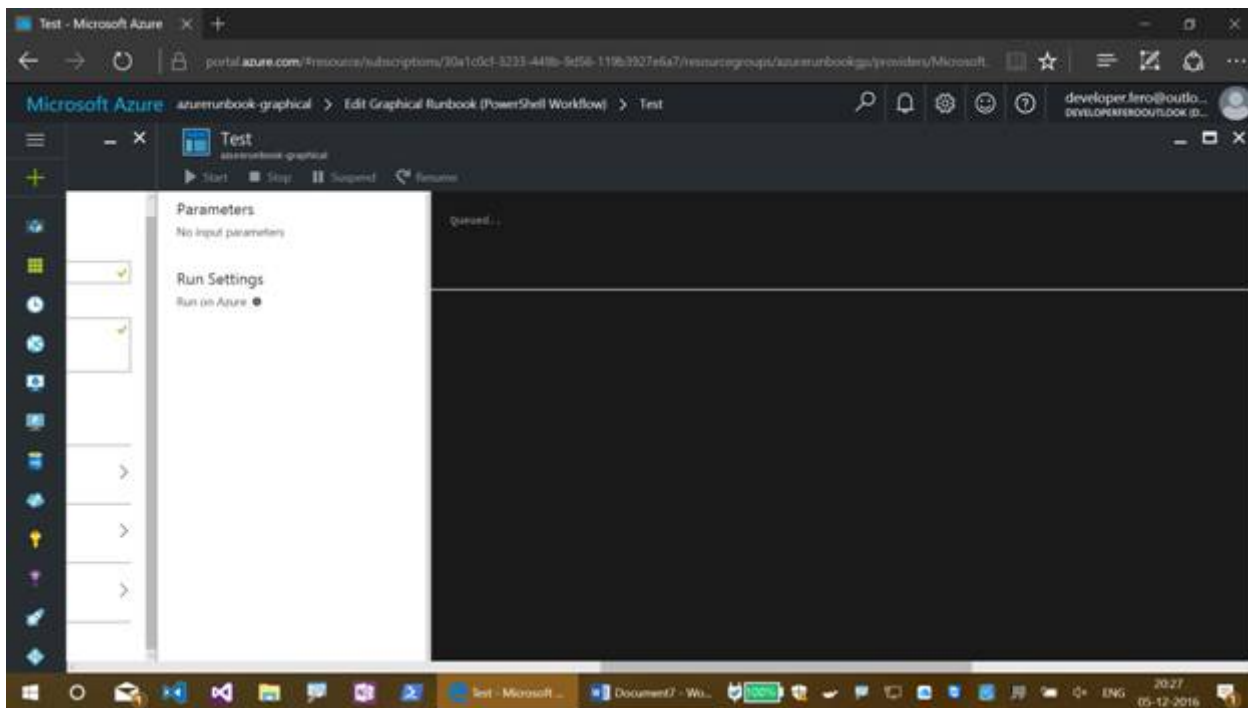
Click Start now.



Afterwards, start process is given for runbook and it will submit the request to test the Runbook.



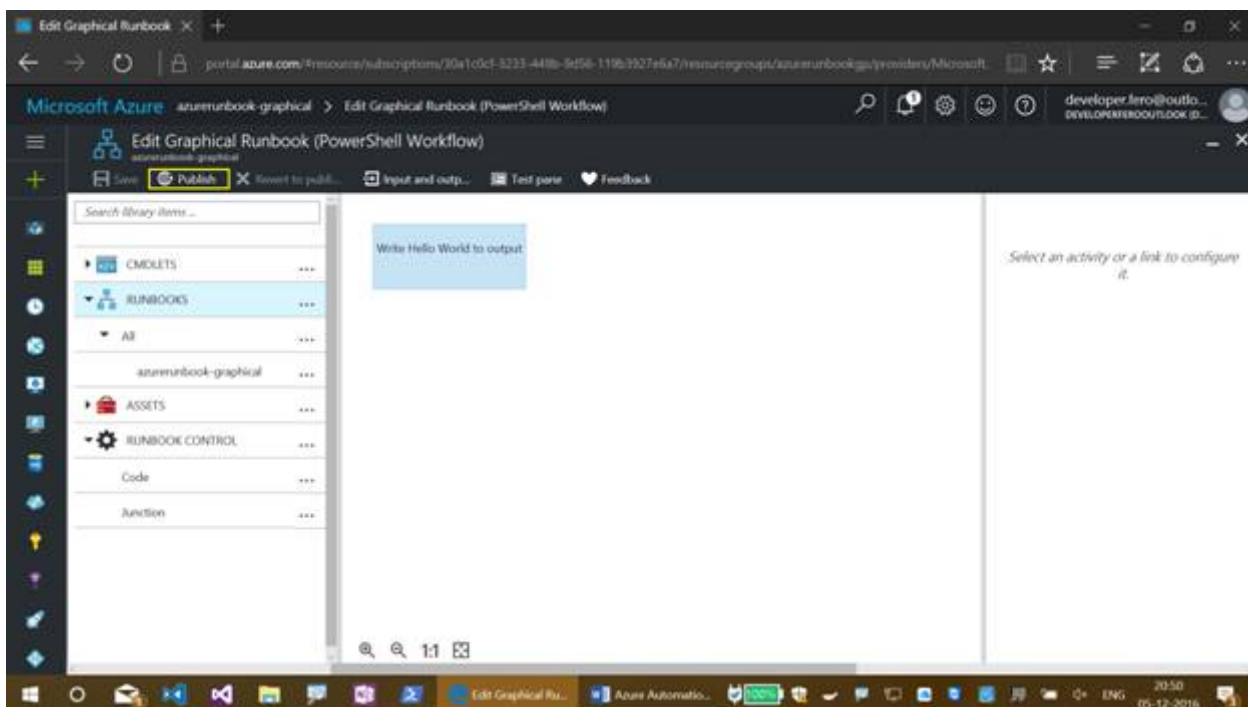
This will display the output of Hello World to your runbook.



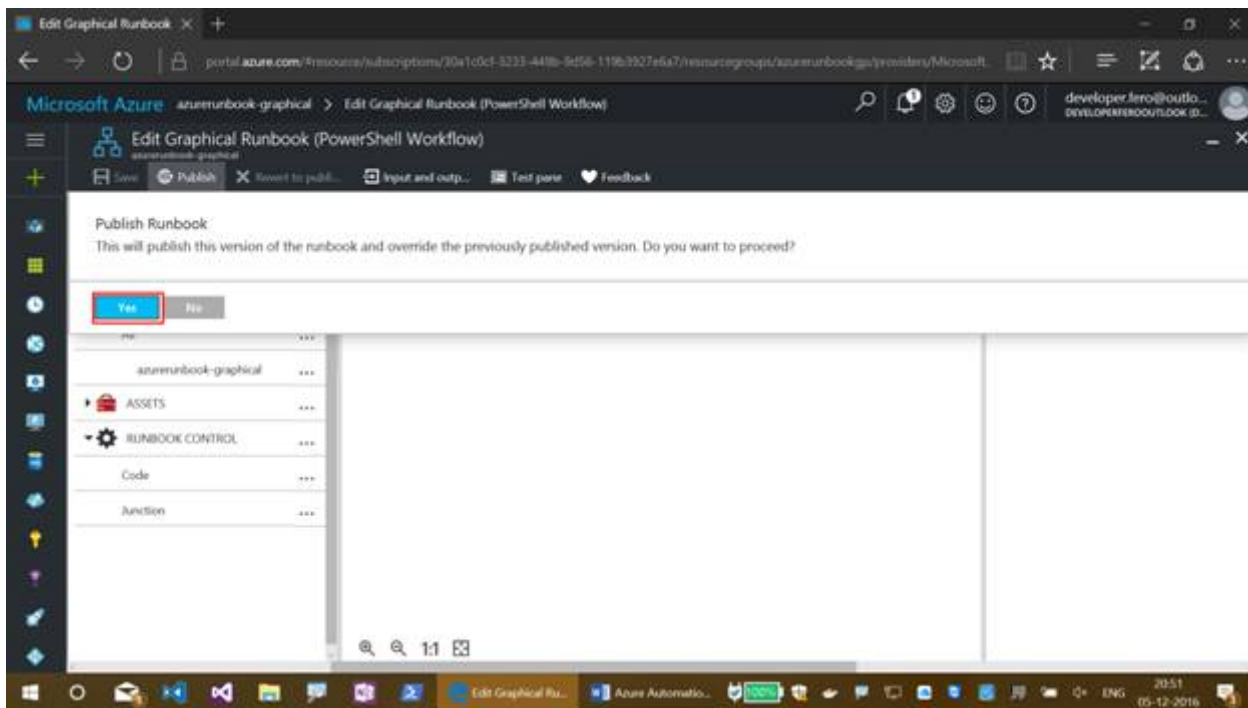
You will be getting your Hello World text here.

Publishing the runbook

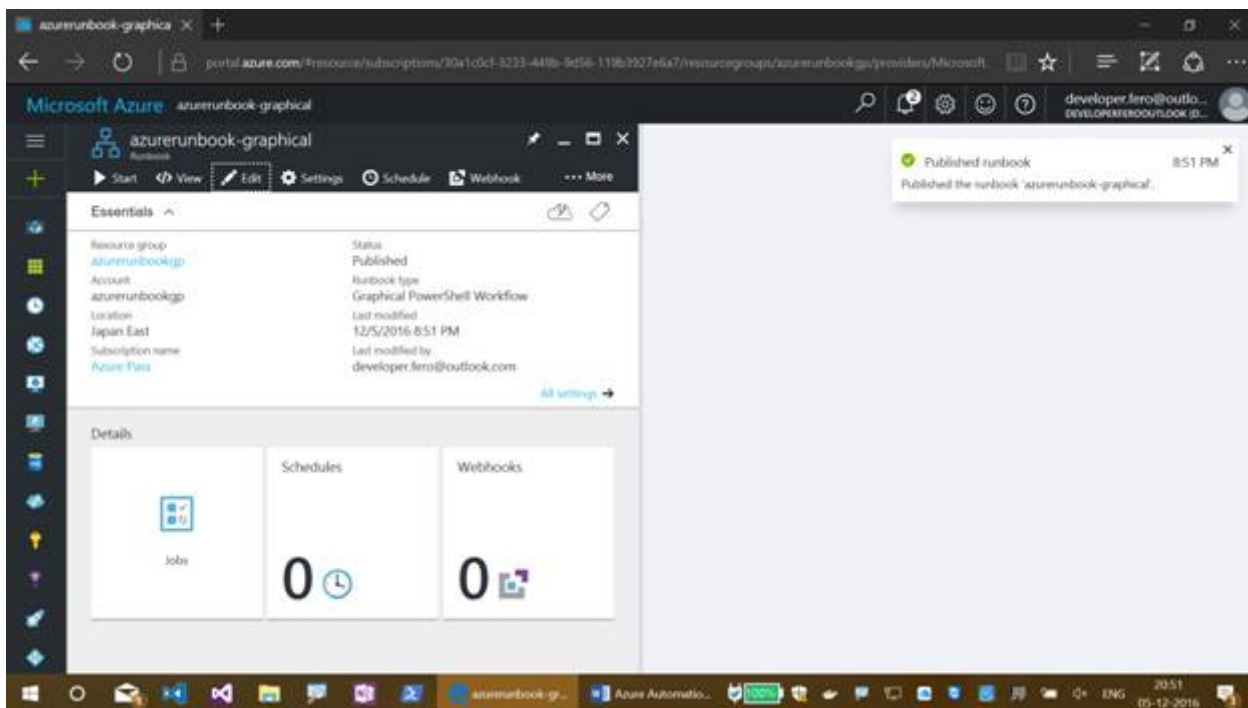
Click Publish in the runbook here.

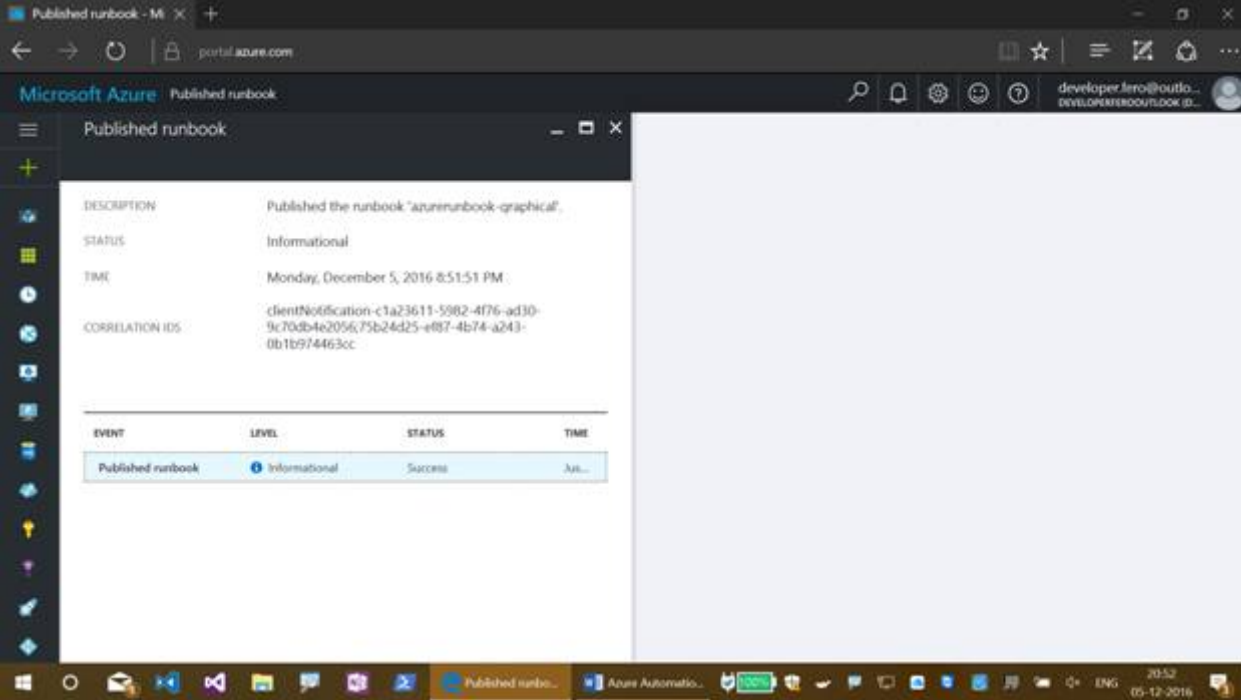


Click Yes to give confirmation to publish runbook.



Your runbook is published now.





The screenshot shows the Microsoft Azure portal interface. The main content area displays the details for a published runbook named 'azurerunbook-graphical'. The details include:

- DESCRIPTION:** Published the runbook "azurerunbook-graphical".
- STATUS:** Informational
- TIME:** Monday, December 5, 2016 8:51:51 PM
- CORRELATION IDS:** clientNotification-c1a23611-5982-4f76-ad30-9c70db4e2056;75b24d25-e087-4b74-a243-0b1b974463cc

Below the details, there is a table showing the event log:

| EVENT | LEVEL | STATUS | TIME |
|-------------------|---------------|---------|--------|
| Published runbook | Informational | Success | Jul... |

The taskbar at the bottom shows several open applications, including 'Published runbook...', 'Azure Automatio...', and a file explorer. The system clock in the bottom right corner indicates the time is 20:52 on 05-12-2016.

Thank you for using C# Corner