

# How to Set up Azure Automation

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2018 Microsoft. All rights reserved.

## Pre-Requisites

- Azure subscription
- Automation Account

## Step 1 - Create a new runbook

1. In the Azure portal, open your Automation account.

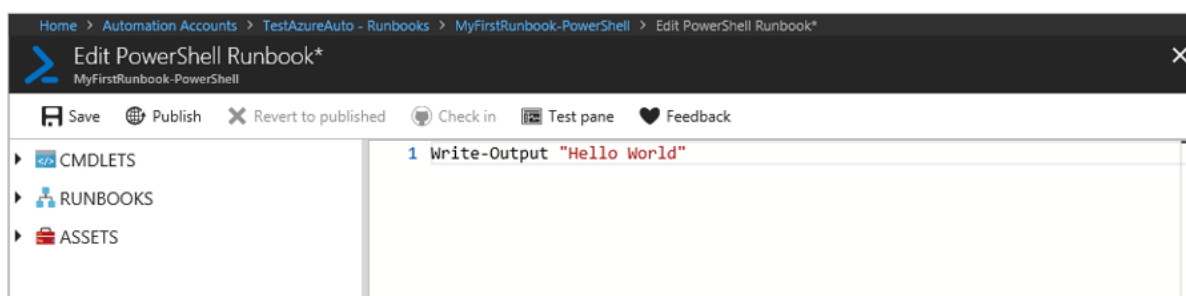
The Automation account page gives you a quick view of the resources in this account. You should already have some assets. Most of those are the modules that are automatically included in a new Automation account. You should also have the Credential asset that's mentioned in the [prerequisites](#).

2. Click **Runbooks** under **Process Automation** to open the list of runbooks.
3. Create a new runbook by clicking the **+ Add a runbook** button and then **Create a new runbook**.
4. Give the runbook the name *myrunbook-PS*
5. In this case, you're going to create a [PowerShell runbook](#) so select **Powershell** for **Runbook type**.
6. Click **Create** to create the runbook and open the textual editor

## Step 2 - Add code to the runbook

You can either type code directly into the runbook, or you can select cmdlets, runbooks, and assets from the Library control and have them added to the runbook with any related parameters. For this walkthrough, you type directly in the runbook.

- 2.1 Your runbook is currently empty, type *Write-Output "Hello World."* in the body of the script.



In our case, for the POC this document was generated by, the script was as following:

```
Invoke-ASCmd -InputFile small.json -Server "asazure://westus.asazure.windows.net/trdlpocaas"
```

```
Invoke-ASCmd -InputFile medium.json -Server "asazure://westus.asazure.windows.net/trdlpocaas"
```

```
Invoke-ASCmd -InputFile reports.json -Server "asazure://westus.asazure.windows.net/trdlpocaas"
```

```
Invoke-ASCmd -InputFile big.json -Server "asazure://westus.asazure.windows.net/trdlpocaas"
```

2.2 Save the runbook by clicking **Save**.

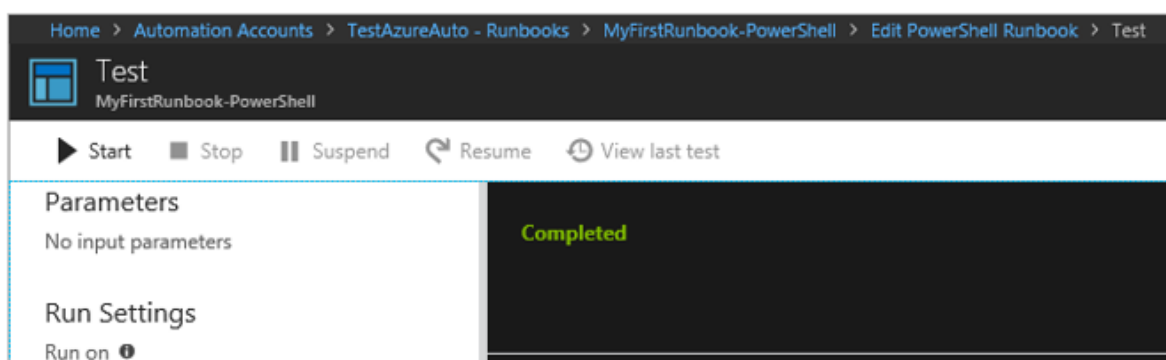
## Step 3 - Test the runbook

Before you publish the runbook to make it available in production, you want to test it to make sure that it works properly. When you test a runbook, you run its **Draft** version and view its output interactively.

1. Click **Test pane** to open the Test pane.
2. Click **Start** to start the test. This should be the only enabled option.
3. A [runbook job](#) is created and its status displayed.

The job status starts as *Queued* indicating that it is waiting for a runbook worker in the cloud to come available. It moves to *Starting* when a worker claims the job, and then *Running* when the runbook actually starts running.

4. When the runbook job completes, its output is displayed. In your case, you should see *myrunbook-PS*



5. Close the Test pane to return to the canvas.

## Step 4 - Publish and start the runbook

The runbook that you created is still in Draft mode. you need to publish it before you can run it in production. When you publish a runbook, you overwrite the existing Published version with the Draft version. In your case, you don't have a Published version yet because you just created the runbook.

1. Click Publish to publish the runbook and then Yes when prompted.
2. If you scroll left to view the runbook in the Runbooks pane now, it shows an Authoring Status of Published.
3. Scroll back to the right to view the pane for Myrunbook-PS. The options across the top allow us to start the runbook, view the runbook, schedule it to start at some time in the future, or create a webhook so it can be started through an HTTP call.
4. You want to start the runbook, so click Start and then click Ok when the Start Runbook page opens.
5. A job page is opened for the runbook job that you created. You can close this pane, but in this case you leave it open so you can watch the job's progress.
6. The job status is shown in Job Summary and matches the statuses that you saw when you tested the runbook.

## Step 5 - Add authentication to manage Azure resources

You've tested and published your runbook, but so far it doesn't do anything useful. You want to have it manage Azure resources. It is not able to do that though unless You have it authenticate using the credentials that are referred to in the [prerequisites](#). You do that with the **Add-AzureRmAccount** cmdlet.

1. Open the textual editor by clicking **Edit** on the Myrunbook-PS page.
2. You don't need the **Write-Output** line anymore, so go ahead and delete it.
3. Type or copy and paste the following code that handles the authentication with your Automation Run As account:

```
$Conn = Get-AutomationConnection -Name AzureRunAsConnection
Add-AzureRMAccount -ServicePrincipal -Tenant $Conn.TenantID `
-ApplicationId $Conn.ApplicationID -CertificateThumbprint
$Conn.CertificateThumbprint
```

4. Click **Test pane** so that you can test the runbook.
5. Click **Start** to start the test. Once it completes, you should receive output similar to the following, displaying basic information from your account. This confirms that the credential is valid.

## Feedback and suggestions

If you have feedback or suggestions for improving this data migration asset, please contact the Data Migration Jumpstart Team ([askdmjfordmtools@microsoft.com](mailto:askdmjfordmtools@microsoft.com)). Thanks for your support!

**Note:** For additional information about migrating various source databases to Azure, see the [Azure Database Migration Guide](#).