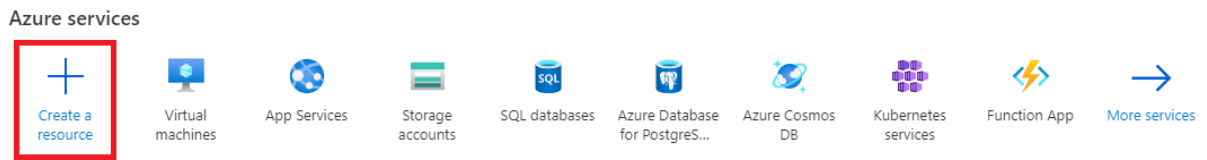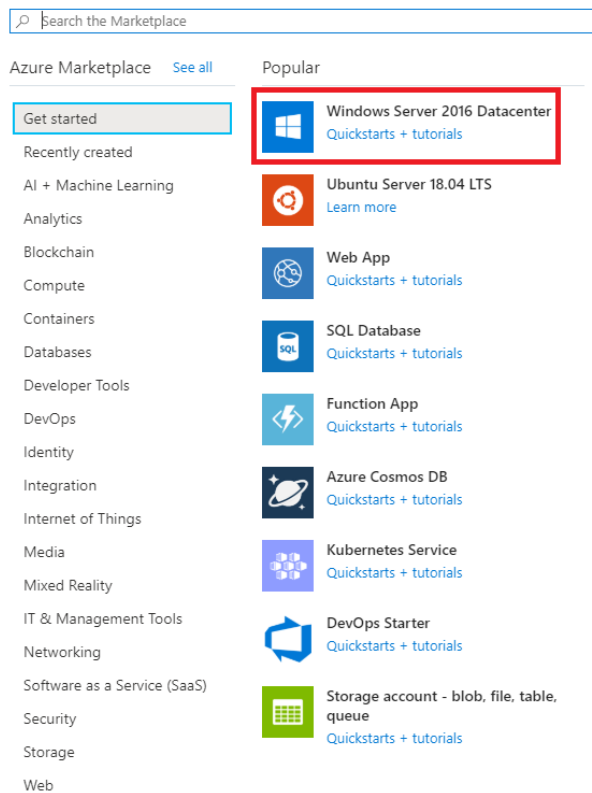**Create an Azure VM**

1. Navigate to the Azure portal at *portal.azure.com.* On the home page, select *Create a resource*

2. Select *Windows Server 2016 Datacenter*

3. Choose your Azure subscription—I am using a free trial, so I chose *Azure subscription 1.* Create a new *Resource group* or use an existing one*.* I created a group called *GatewayRG.* You can do this by selecting the *Create new* button under the dropdown menu. A resource group houses metadata for related resources

4. Provide a name for your virtual machine and specify its region. I am keeping it in the same location as the Power Platform tenant's default region (West US, according to the admin center). Click one of the VM offerings given, or click the *Select size* button to view other VM options

Instance details

Virtual machine name * ⓘ   CloudGateway ✓

Region * ⓘ   (US) West US ⌄

Availability options ⓘ   No infrastructure redundancy required ⌄

Image * ⓘ   Windows Server 2016 Datacenter ⌄

Browse all public and private images

Size * ⓘ   Standard_E2s_v3 - 2 vcpus, 16 GiB memory ($175.20/month) ⌄

Select size

5. Set an administrator username and password. Allow inbound port 3389 through since it is used to connect to the remote machine. Then, click *Review + create*.

Administrator account

Username * ⓘ   AzureUser ✓

Password * ⓘ   •••••••••••••••• ✓

Confirm password * ⓘ   •••••••••••••••• ✓

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ   ◯ None   ⦿ Allow selected ports

Select inbound ports *   RDP (3389) ⌄

⚠ **This will allow all IP addresses to access your virtual machine.** This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

6. Ensure that validation passes. I have shown a basic configuration here, but take note of advanced features, including VM backup. Then, select *Create.* When the deployment finishes, navigate to the VM

7. Click *Connect* on the VM's overview page. Then, select *RDP.* Download the file. Click the RDP file to initiate the connection, and enter your administrator credentials

     🖋 Connect   ▷ Start   ↻ Restart   ☐ Stop   🖾 Capture   🗑 Delete   ⟳ Refresh

8. Since Server Manager opens automatically, I recommend disabling Internet Explorer's Enhanced Security Configuration. To do this, press *Local Server* on the side navigator of the administration
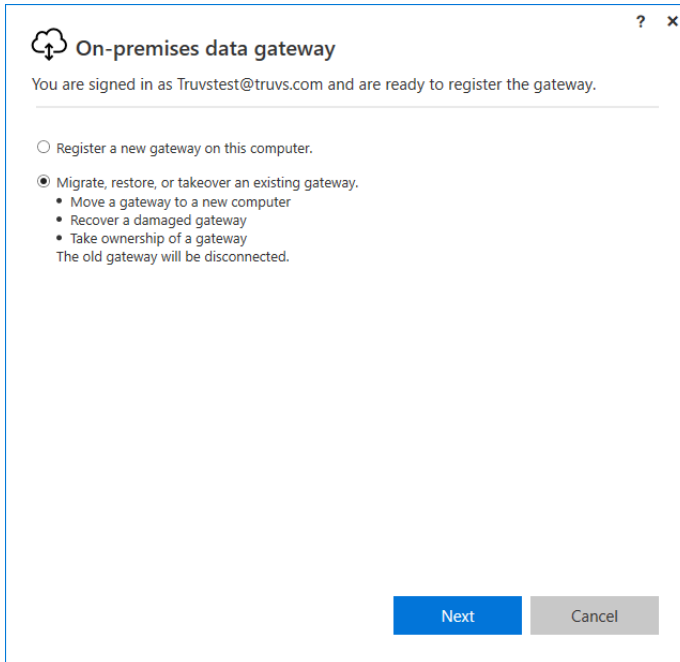
application, click *IE Enhanced Security Configuration*, and disable the feature for both administrators and users
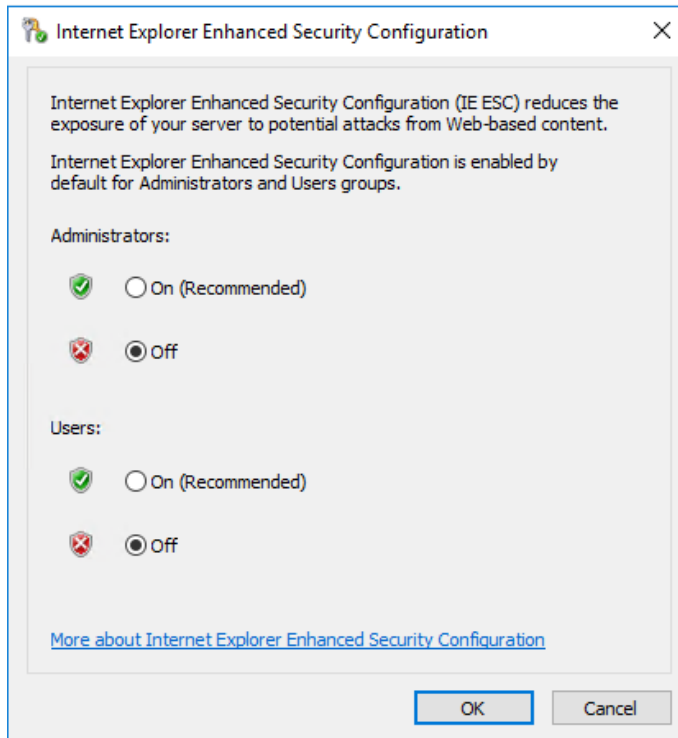
Gateway Installation (you will only need to install the gateway if you wish to run UI flows on a system *different* from the one used to develop the flows)
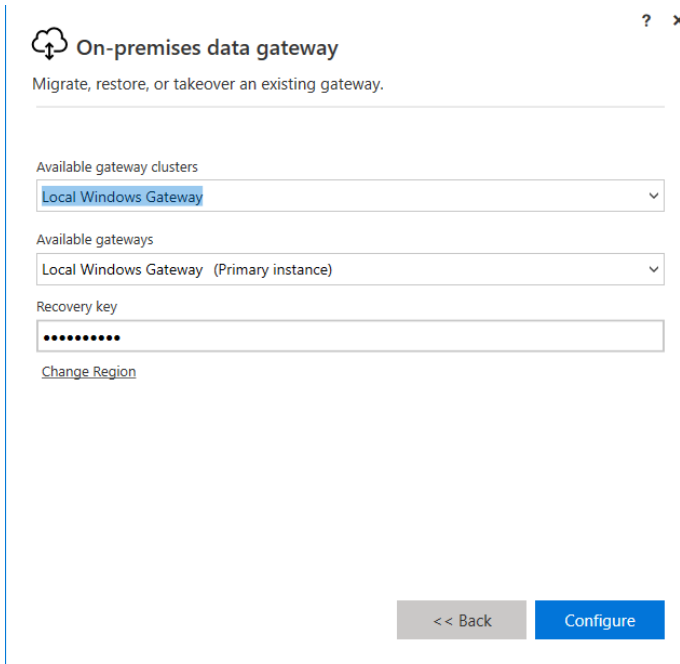
9. Run the local gateway installer, available from this link. You will be asked to enter an email address during the installation. This user will become the gateway admin. You will also need to sign into Power Automate and create flows with this account, unless you grant others permission to use the gateway (more on this later)

10. You will be asked to choose between creating a new gateway or migrating from an existing gateway. Since this guide indicates how to perform a cloud migration, I will be migrating from an existing gateway


11. The old gateway, known as *Local Windows Gateway*, ran on my laptop. Since I installed it, I know the recovery key needed to perform the migration. By default, the gateway will be created in the default region of the Power Platform tenant. The primary instance designation relates to the fact that if the gateway is part of a cluster, it will receive all requests (if it is online). Finally, select *Configure*

Installing Extensions

12. Install Microsoft Edge or Google Chrome on the VM
13. Navigate to this link to obtain and run the UI flows installer. Accept the given defaults
14. As part of the installation process, the UI flows browser extension is installed, which you will need to enable. In the image below, Chrome notifies me of the addition and the actions the extension performs
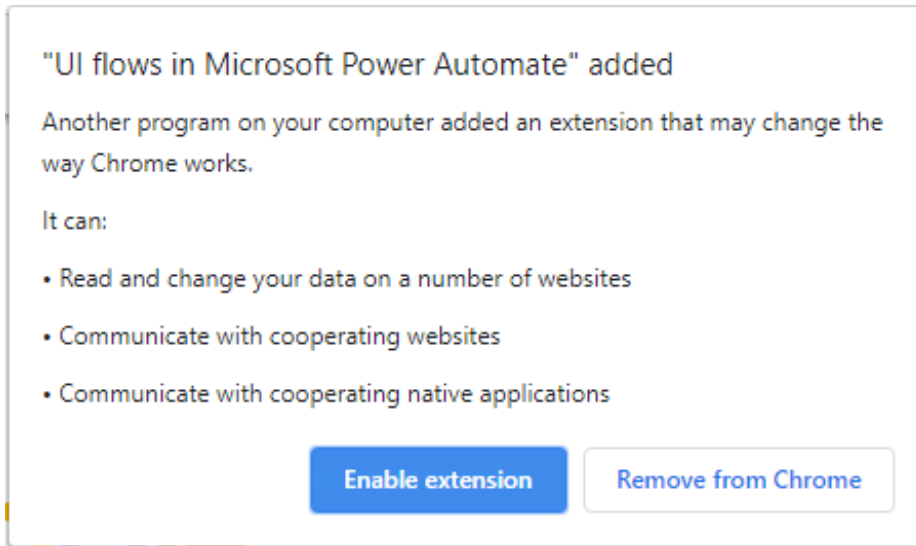
**Changing Configuration in Power Automate & Running the Flow on the VM**

15. Our server configuration is complete. Navigate to Power Automate. Sign in with the account used to configure the gateway and select *My flows*. Choose a flow from the selection that calls a UI flow that can be executed on the VM. Then, select *Edit*
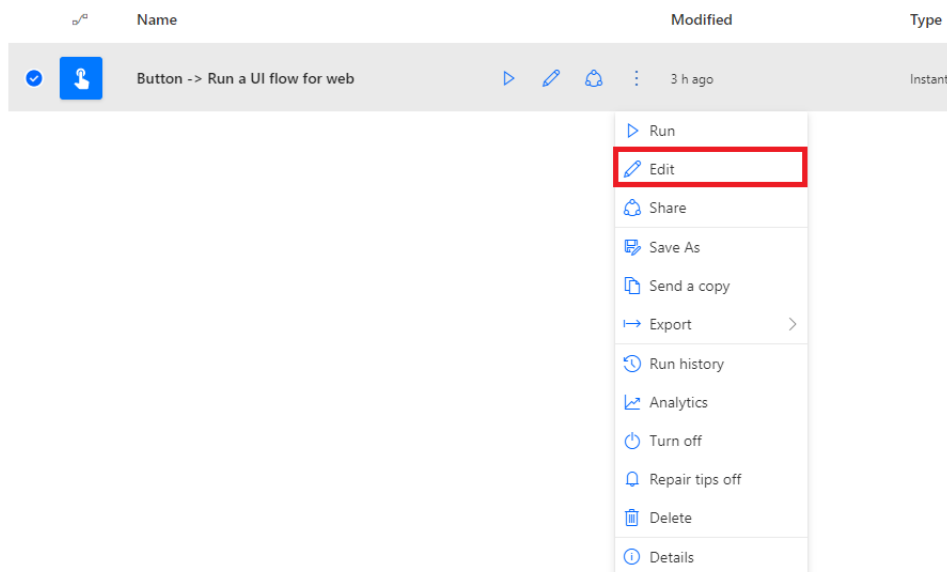
16. Select the call to the UI flow. Then, under *My connections,* select *+Add new connection*
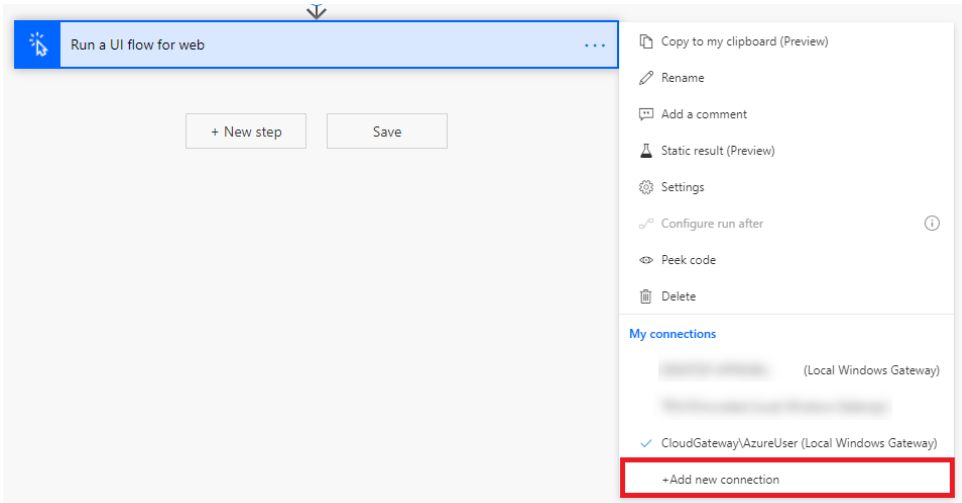
17. Choose the gateway that you created or migrated as the *Gateway name*

18. To find your domain and username, navigate to your VM. Here, you can issue the *echo %userdomain%* and *echo %user%* commands to obtain your domain and username, respectively

19. Input your admin user password into the *Password* field. Then, select *Create*
20. Configuration is now complete. Click *Save* at the bottom of the flow to persist your changes

21. Select *Test* from the top right corner of the page. Select *I'll perform the trigger action* and then select *Test.* Then, select *Run flow.* Verify that the actions documented in the UI flow are executing on the VM. As you can see in the image below, my flow ran correctly

By now, it is easy to see the capability of this system. However, running a VM 24/7 when it spends much of its time at idle is wasteful. So, I will show you how to create flow elements to start and stop your VM when an action must be performed. Then, I will finish my discussion by exploring the possibility of giving other users access to the gateway

Start and Stop VM On-Demand

22. Navigate back to Azure Portal's *+ Create a resource* and locate *Automation* under *IT & Management Tools*
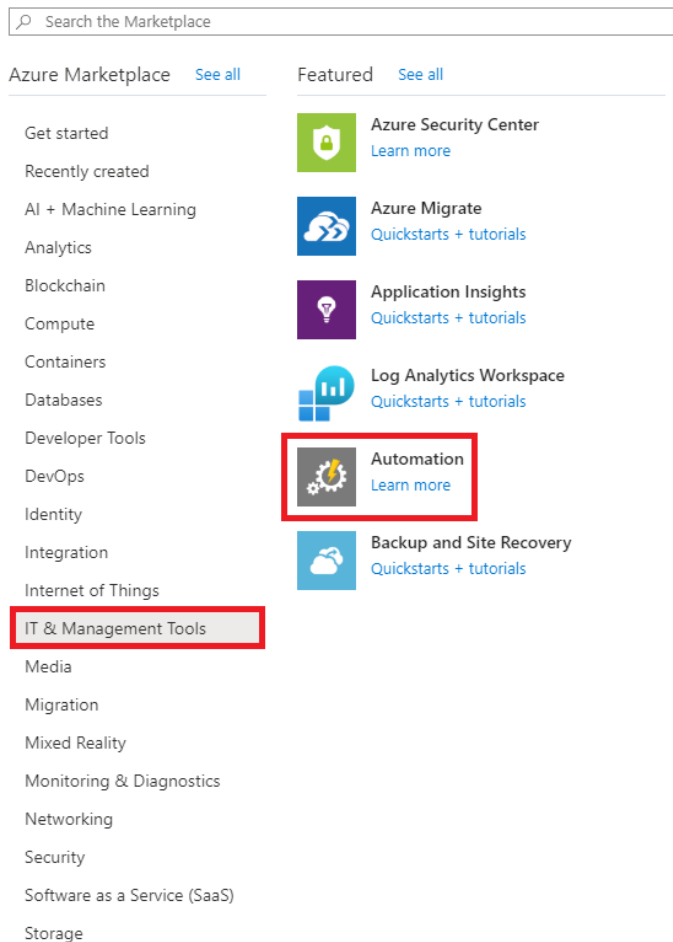
23. *Name* your Automation Account. Ensure that it is created in the same subscription and resource group in which your VM resides. Also, verify that an *Azure Run As* account will be created. Then, select *Create*

Add Automation Account

Name * ⓘ

ControlFlowVM [!]

Subscription *

Azure subscription 1 ⌄

Resource group *

GatewayRG ⌄

Create new

Location *

West US ⌄

Create Azure Run As account * ⓘ

[**Yes** | No]

ⓘ This will create Azure Run As account in the Automation account which are useful for authenticating with Azure to manage Azure resources from Automation runbooks. Note that the creation of Azure Run As account may affect the security of the subscription.Learn more
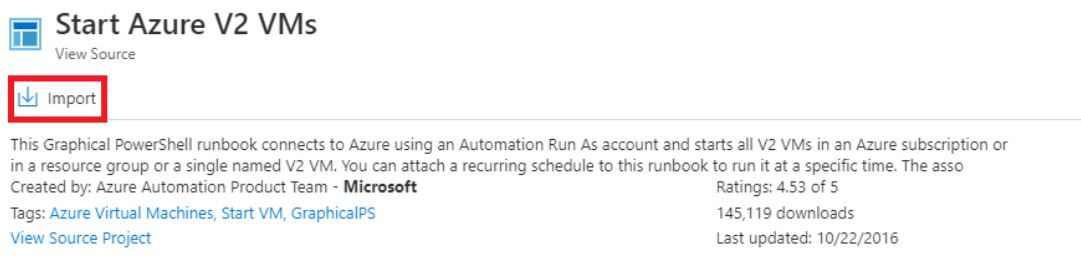
ⓘ Learn more about Automation pricing. ↗

Create

24. Once the deployment finishes, navigate to the Account. Navigate to *Runbooks* under *Process Automation*

Process Automation

⁂ Runbooks

▣ Jobs

🛍 Runbooks gallery

🐍 Hybrid worker groups

🔍 Watcher tasks
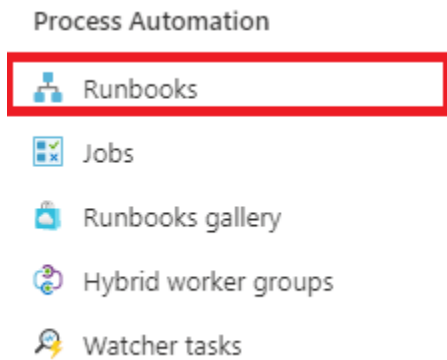
25. Select *Browse gallery* at the top of the page

＋ Create a runbook   ↓ Import a runbook   🛍 Browse gallery   ☐ Learn more   ⟳ Refresh

26. Select *Start Azure V2 VMs,* which should be the first entry on the page. Then, select *Import*

🖼 **Start Azure V2 VMs**
View Source

↓ Import

This Graphical PowerShell runbook connects to Azure using an Automation Run As account and starts all V2 VMs in an Azure subscription or in a resource group or a single named V2 VM. You can attach a recurring schedule to this runbook to run it at a specific time. The asso
Created by: Azure Automation Product Team - **Microsoft**   Ratings: 4.53 of 5
Tags: Azure Virtual Machines, Start VM, GraphicalPS   145,119 downloads
View Source Project   Last updated: 10/22/2016

27. Keep the default *Name* and *Description* the same, and select *OK*

Name * ⓘ

StartAzureV2Vm
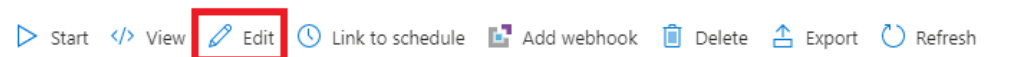
Runbook type ⓘ

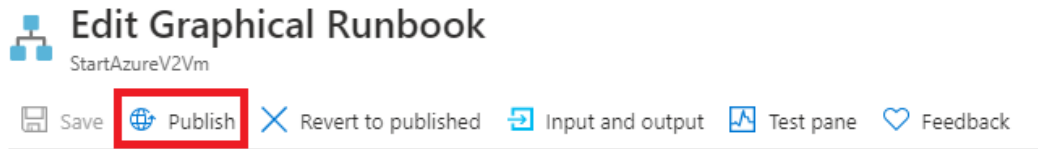Graphical   ⌄

Description

This Graphical PowerShell runbook
connects to Azure using an Automation
Run As account and starts all V2 VMs in

28. At the *Runbooks* page, select the *StartAzureV2Vm* runbook. Select *Edit* from the toolbar

▷ Start   </> View   ✎ Edit   ⟲ Link to schedule   🖼 Add webhook   🗑 Delete   ⬆ Export   ⟳ Refresh
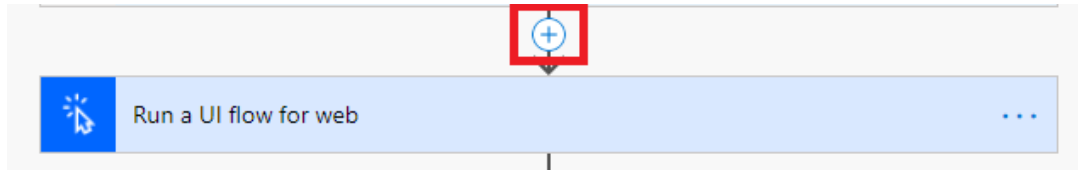
29. Select *Publish* from the toolbar of the *Edit Graphical Runbook* page

30. Repeat steps 25-29, but publish *Stop Azure V2 VMs* instead
31. Again, navigate to Power Automate and sign in with the account used to configure the gateway. Select the same flow that you edited previously. Before the call to the UI flow, select the *+* button that appears on the arrow

32. Select *Add an action* and search for *Create job.* Select *Create job* by *Azure Automation*
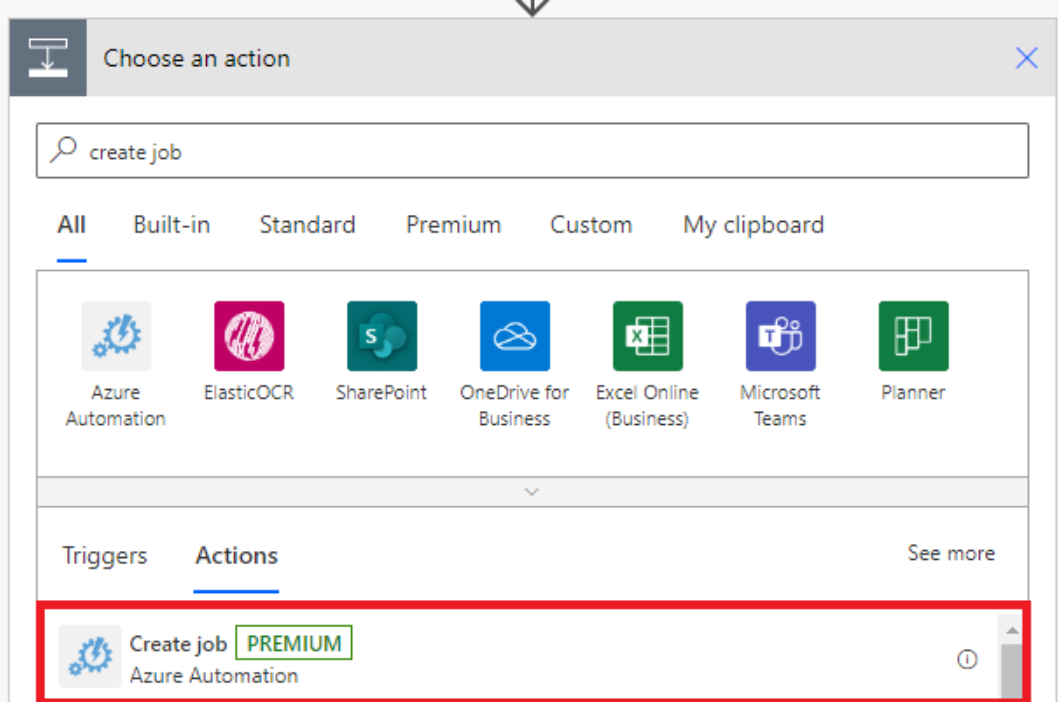
33. Enter your Azure subscription, the name of the resource group in which you created your automation account, and your automation account itself into the job parameter box. Then, enter *StartAzureV2Vm* as the *Runbook Name*, and verify that *Wait for Job* is set to *Yes.* Set the *ResourceGroupName* runbook parameter to the resource group of your VM. Finally, populate the *VMName* parameter with the name of your VM. Below is my completed configuration

34. Repeat this procedure following the call to the UI flow. However, use *StopAzureV2Vm* as the runbook name

Give Others Gateway Access

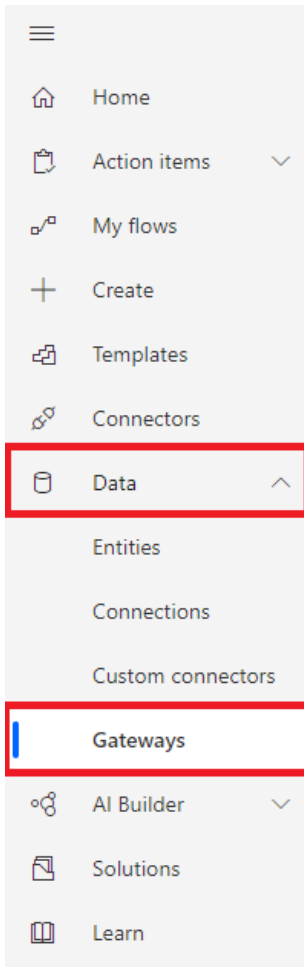35. Again, navigate to [Power Automate](#) and sign in with the account used to configure the gateway. Under *Data*, select *Gateways*

36. Select the gateway that you would like to give others permission to use. Select *Share*
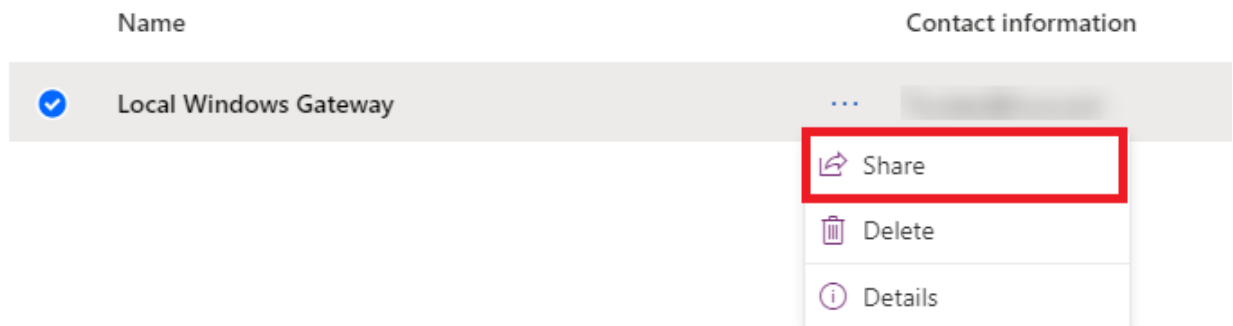
37. I can enter the email address of the individual who needs to access the gateway

38. Note that when I enter an email address, I can assign a gateway permission to the user. The three possible permissions are:
   a. *Can use* – can run flows on the system identified by the gateway, but cannot provide this ability to others
   b. *Can use + share* – can run flows on the system identified by the gateway, and can share the gateway with others when sharing flows
   c. *Admin* – can modify gateway users, can alter the local resources which they can access, and can delete the gateway

39. Note that if a user is given either the *Can use* or *Can use + share* designation, then a gateway admin can manually choose which local resources they can access

- ☑ DB2
- ☑ File System
- ☑ Apache Impala
- ☑ Informix
- ☑ MySQL
- ☑ Oracle Database
- ☑ PostgreSQL
- ☑ SAP ERP
- ☑ SharePoint
- ☑ SQL Server
- ☑ Teradata
- ☑ UI flows
- ☑ HTTP with Azure AD