Azure VMs Stop & Start PowerShell Script

Procedure

Table of Contents

[Revisions 3](#_Toc75951832)

[Initial Steps 4](#_Toc75951833)

[Initial Configuration 4](#_Toc75951834)

[Script run example 5](#_Toc75951835)

Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| June/30/2021 | 1.0 | Initial version | Walter Gonzalez Vega |

Initial Steps

Before any attempt to run the script there are a few steps needed to prepare your computer to run the script. There are 3 modules that needs to be installed, these are:

1. az.accounts
2. az.compute
3. az.resources

To install these components, you need Administrator level access to your computer. If you don’t have Administrator access to your computer, please contact Ivette Silva for assistant. One of her technicians could do the installation of these modules.

If you have Administrator access to your computer, follow the following steps to complete the installation of each module.

## Environment Configuration

This procedure starts from the assumption that the basic PowerShell scripting language is already installed. If not please follow this guideline to install the basic PowerShell scripting language from Microsoft.

You will need internet connection to do the installation for PowerShell and/or the required modules.

1. Open a CMD or terminal session with Administrator privilege and check if the PowerShell scripting language is already installed on your computer. To check for this run the following command, PWSH as illustrated in the following screenshot:

Graphical user interface

Description automatically generated with medium confidence

*Figure(a).*

If PowerShell is already installed, you’ll see a response like the above graphic stating the version that is installed on your system. See that your command prompt has changed, it starts with the letters PS (PowerShell) followed with the Path where you’re located.

1. Now run the following commands to install each module individually. You must wait for each module to finish its installation; these may take some time for each module. If you are prompted during the installation to accept the modules to be installed, please answer with a capital A to accept all changes:
   1. import-module -name az.accounts
   2. import-module -name az.compute
   3. import-module -name az.resources
2. Once the modules are installed, check each module with the following command. This will list all installed modules in your system. You’ll see the modules listed in the output of the command:
   1. Get-installedmodule

## Script run example

Now you’ll have a couple of options from where to run the script. If you have Microsoft Visual Studio Code, you can use this IDE to run the PowerShell Script. Also, you can run the script from the Windows PowerShell ISE (x86) application as Administrator (Run as Administrator) found on the Start Menu of your system, this is the environment that I’ll use for this example, Figure(b).

Graphical user interface, application

Description automatically generated

*Figure(b).*

1. Open the script named vm-start-crt.ps1 on the IDE of your selection and run the script.
2. When the script starts, you’ll be prompted to do a login in the same fashion when you do a login to the Azure Portal with the MFA proccess **if you’re not already logged in**. On this example I’m using a test subscription, you must use the Evertec account:
   1. Graphical user interface, text, application, chat or text message

      Description automatically generated

*Figure(c).*

* 1. **If you are** **not** **already logged in**, you’ll be prompted to provide your MFA information

1. After you logged in, a menu of subscriptions will be shown:

Graphical user interface, text, email

Description automatically generated

*Figure(d).*

These is the menu that you’ll see, the list on the above figure, *figure (d).*

**Note:** *For this example, as I’m using a test subscription, I see a different list of subscriptions (Pay-As-You-Go), Figure(e).*

1. Now you can choose the subscription that you want to use.
2. Once you select the subscription a process to connect to the subscription starts:

Graphical user interface, text

Description automatically generated

*Figure(e).*

1. Once connected, you’ll be prompted to enter the name of the virtual machine twice, Figure(f). For the script to continue you’ll need to provide the name of the VM a second time to make sure that we have choose the correct VM to work with, is like a confirmation process, Figure(g).



*Figure(f).*



*Figure(g).*

1. If both input values matched the script continues, if not you’ll need to enter the values again.



*Figure(h).*

**Note:** *During the script process, you could cancel the script by doing a Control/C on your keyboard, this will stop the script at any time.*

1. The script will look for the VM at each Resource Group and creates a list of all VMs occurrences:

Graphical user interface, text, email, website

Description automatically generated

*Figure(i).*

1. Select the VM that you are working with. The script will get the VM actual State, this is a brief information block where you can see the state of the VM (if it is deallocated or not, a deallocated stated for an Azure VM means that the VM is stopped), the name and Resource Group of the VM, the subscription name, and the Tenant (Main subscription) for the VM.

Text

Description automatically generated with medium confidence

*Figure(j).*

1. Is in here that you can select the desired action to performed, Figure(j).
2. In this case I choose to start the VM since the VM state was a deallocated state, Figure(j). Also, you need to confirm the Start action for the VM (this is also true for the Stop action for a VM, you’ll be prompted to confirm each action). If you don’t want to be prompted at several times choose the *“Yes to All”* option:

Graphical user interface, text, application

Description automatically generated

*Figure(k).*

1. The chosen action process begins, this might take a few minutes to complete, in some cases like starting a VM it might take up

to 3 minutes or more. After completion you’ll see a message on the screen stating the status of the action, succeed or failed and the start and end time of the action, Figure(l).

Graphical user interface, text

Description automatically generated

*Figure(l).*

At this point the script has finished its run. If you need to work again on the same or another VM you must run the script one more time for each VM.