AZURE MIGRATION COMPATIBILITY

EXPONENTIAL-E

DOCUMENT DATED: 28/01/2022
PREPARED BY: Dominic.Fradley@exponential-e.com
DOCUMENT VERSION: v1.0

Table of Contents

Copyright© 2022 Exponential-e Ltd

Distribution

This document may be distributed internally within Exponential-e to all of its employees. Any non-disclosure agreements in place between Exponential-e and Exponential-e automatically include all of Exponential-e's employees. Exponential-e will take all possible precautions to ensure the information contained in this document remains confidential.

This document may be distributed internally within Exponential-e however it may not be disclosed to any other party or reproduced in whole or in part in any format without the prior written consent of a director of Exponential-e.

Revision History

REVISION	CHANGES MADE	MADE BY	DATE
V0.1	Document draft	Dominic Fradley	27/01/22
V0.2	Document proof	Archie Christopher	27/01/22
V1.0	Document Release	Archie Christopher	28/01/22

Copyright© 2022 Exponential-e Ltd

1. Summary

1.1 Background

When moving Azure resources its important to understand that not all resources can be moved without some form of manual intervention. For example, Child resources cannot be moved independently from its Parent resource, additionally third-party resources do not currently support move operations. Identifying these resources can be time consuming and confusing if attempting to gather all the pre-requisites before attempting a move, especially in expansive environments.
With this in mind, Exponential-e have created a script that can be used to identify resources that may not be moved in a pending operation and therefore must be considered for manual moving or recreation.

1.2 Objectives

The purpose of this script is to allow the user to identify which items may or may not be moved from an ever-growing list of resources. It is easy to run with little configuration required, has no dependency on third party tools and can be initiated from any PC with an internet connection and the correct credentials to connect to the desired tenancy.
The script calls upon a list of data that is consistently updated with new resources that is hosted in GitHub to ensure that data output in the script logs is current and correct.

1.3 Deliverables

Once the script is complete, it will allow the end user to easily digest the output and discuss with Exponential-e any upcoming migration projects that require Resource Group, Regional or Subscription moves for their Azure resources. It will provide an accurate snapshot of the environment without the need to cross reference Microsoft knowledge base articles to ensure accuracy of results.

Copyright© 2022 Exponential-e Ltd

2. Pre-requisites

2.1 PowerShell

To run this script the latest version of PowerShell core should be used to ensure that compatibility with the latest Azure PowerShell modules is achieved. At the time of writing, the latest version available is 7.2.1 and is available directly from Microsoft at the following link…

https://docs.microsoft.com/en-gb/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.2

The correct .msi should be chosen depending on your machines architecture (x86 or x64) and can be installed by simply running the installer. PowerShell 7.2 will run side-by-side with the inbuilt PowerShell 5.1 and does not require any further configuration.

2.2 Azure PowerShell Module

The script relies upon the Azure PowerShell Module to execute correctly. To ensure you have the latest version of the module please follow the steps outline here.

1. PowerShell script execution policy must be set to remote signed or less restrictive. Get-ExecutionPolicy -List can be used to determine the current execution policy.

   Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

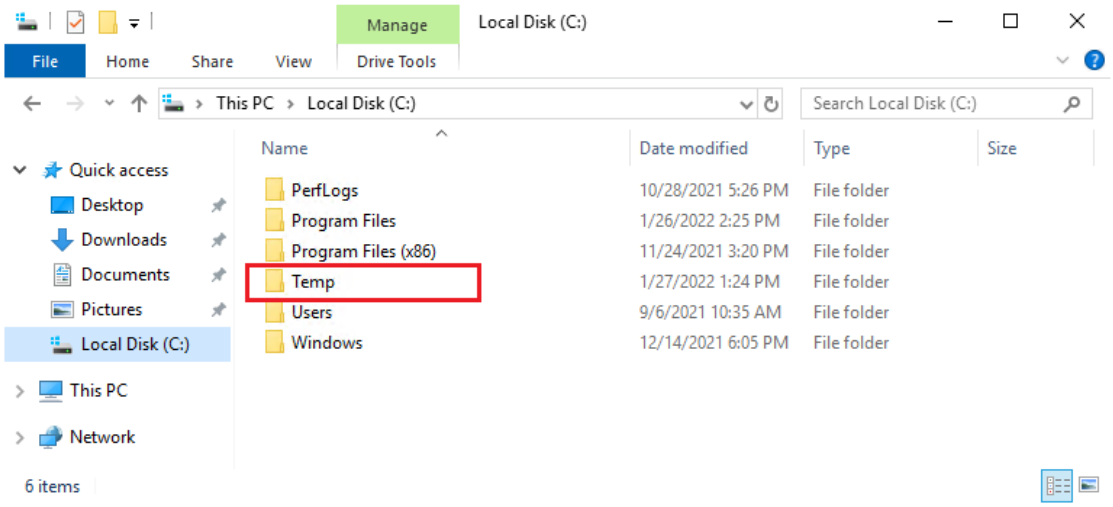
2. Using the Install-Module cmdlet is the preferred installation method for the Az PowerShell module. Install the Az module for the current user only. This is the recommended installation scope. This method works the same on Windows, macOS, and Linux platforms. Run the following command from a PowerShell session:

   Install-Module -Name Az -Scope CurrentUser -Repository PSGallery -Force

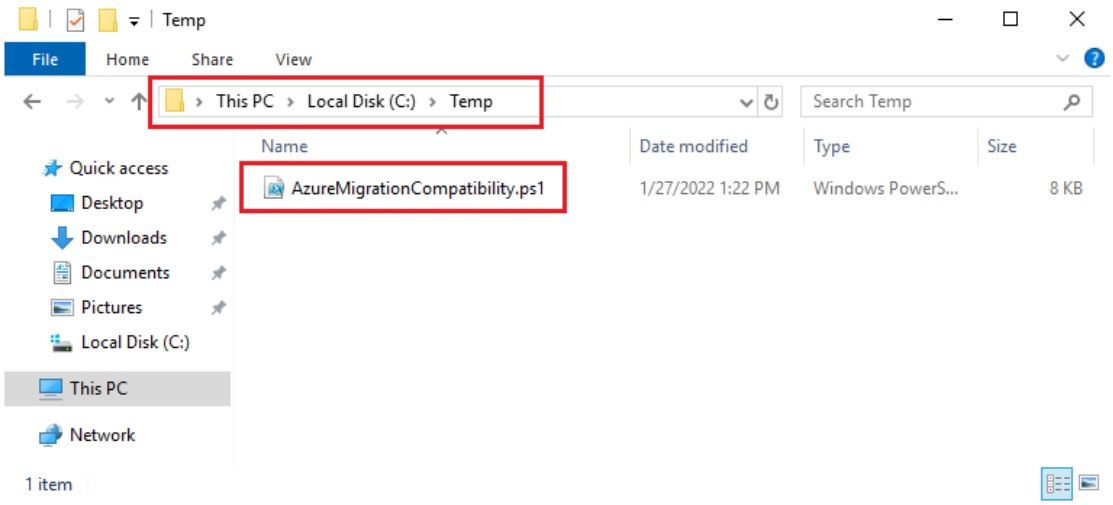
Copyright© 2022 Exponential-e Ltd

2.3 Script Placement

Create a folder in the root of your local drive named “Temp”. When the script is executed it will create an additional nested folder under this location to write the required logs to.



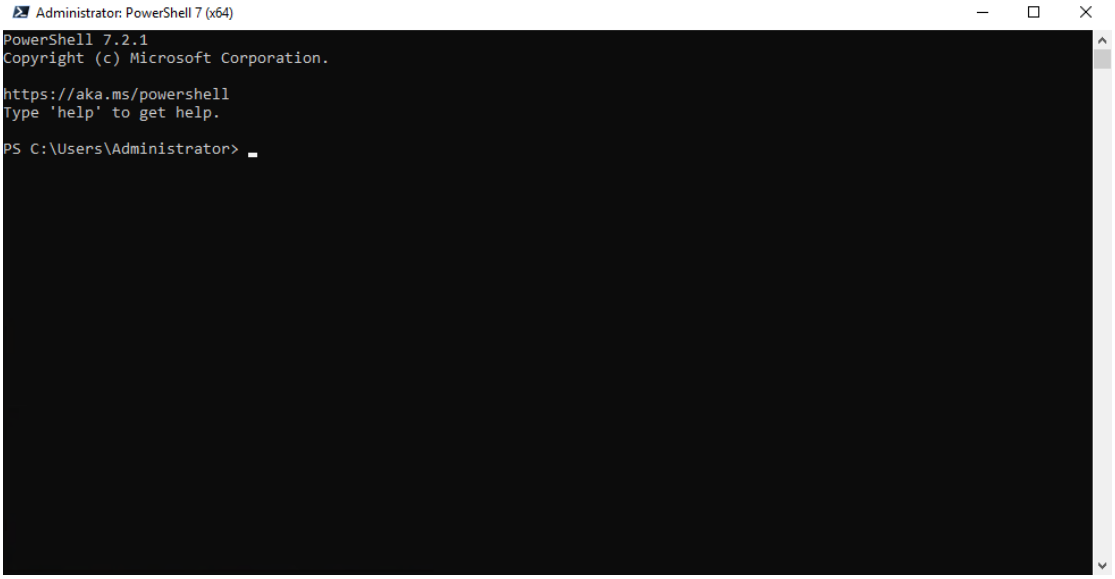
Copy the AzureMigrationCompatibility.ps1 script into his folder.



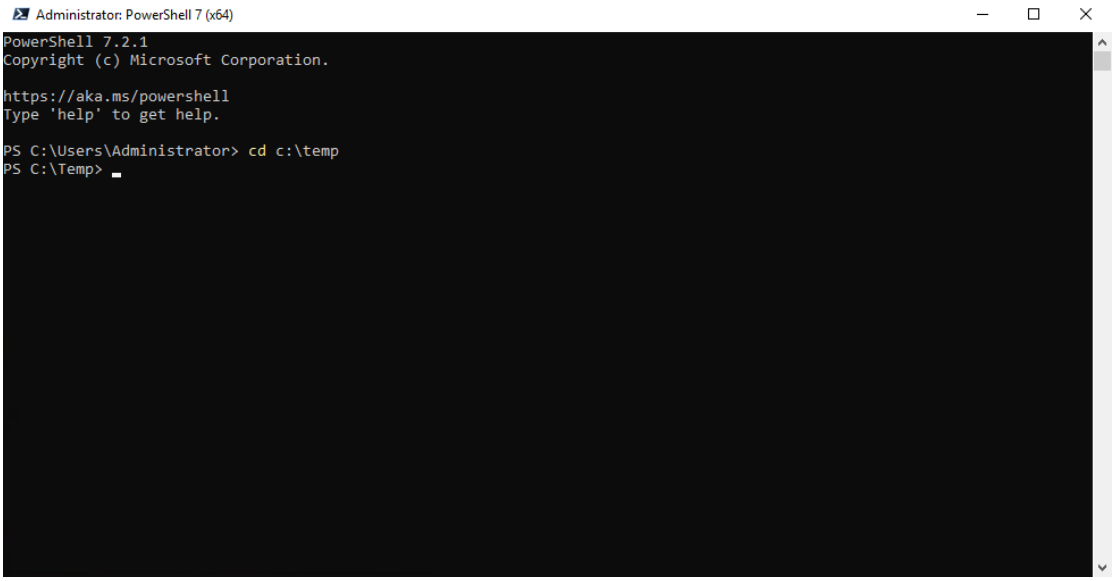
2.4 Script Execution

Now the prerequisites have been met the script can now be executed from the PowerShell console and data gathering may begin. To do so, open the PowerShell 7.2 application.

Copyright© 2022 Exponential-e Ltd

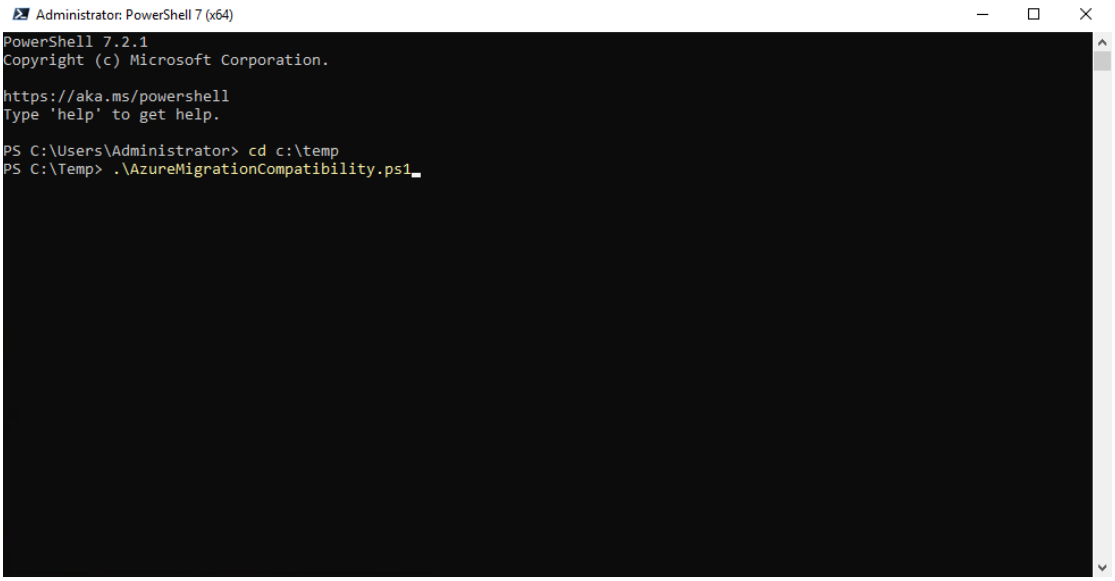


We will now set our working directory to the Temp folder we created earlier.

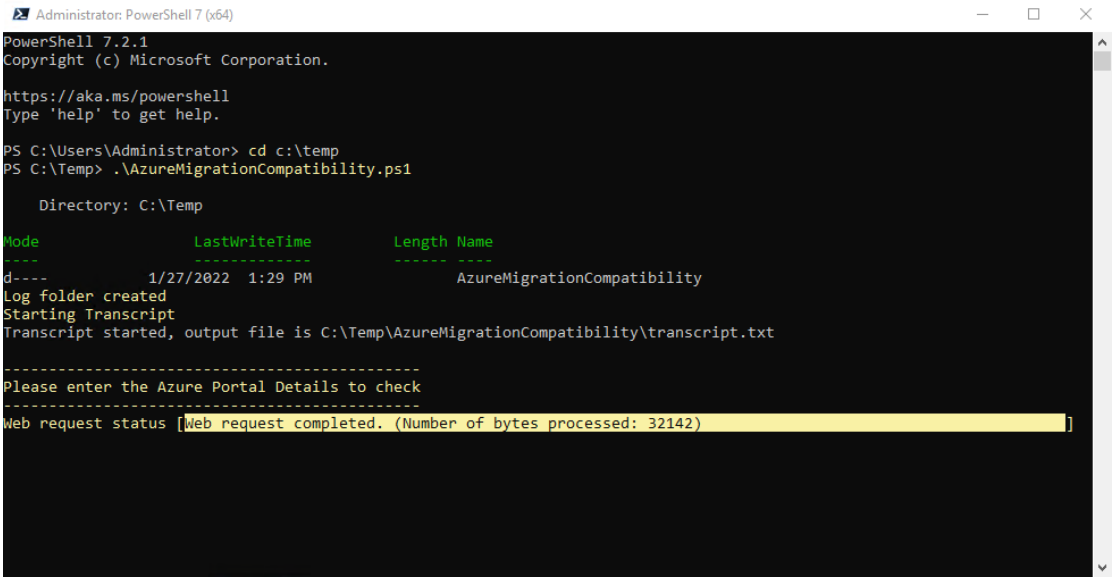


Copyright© 2022 Exponential-e Ltd

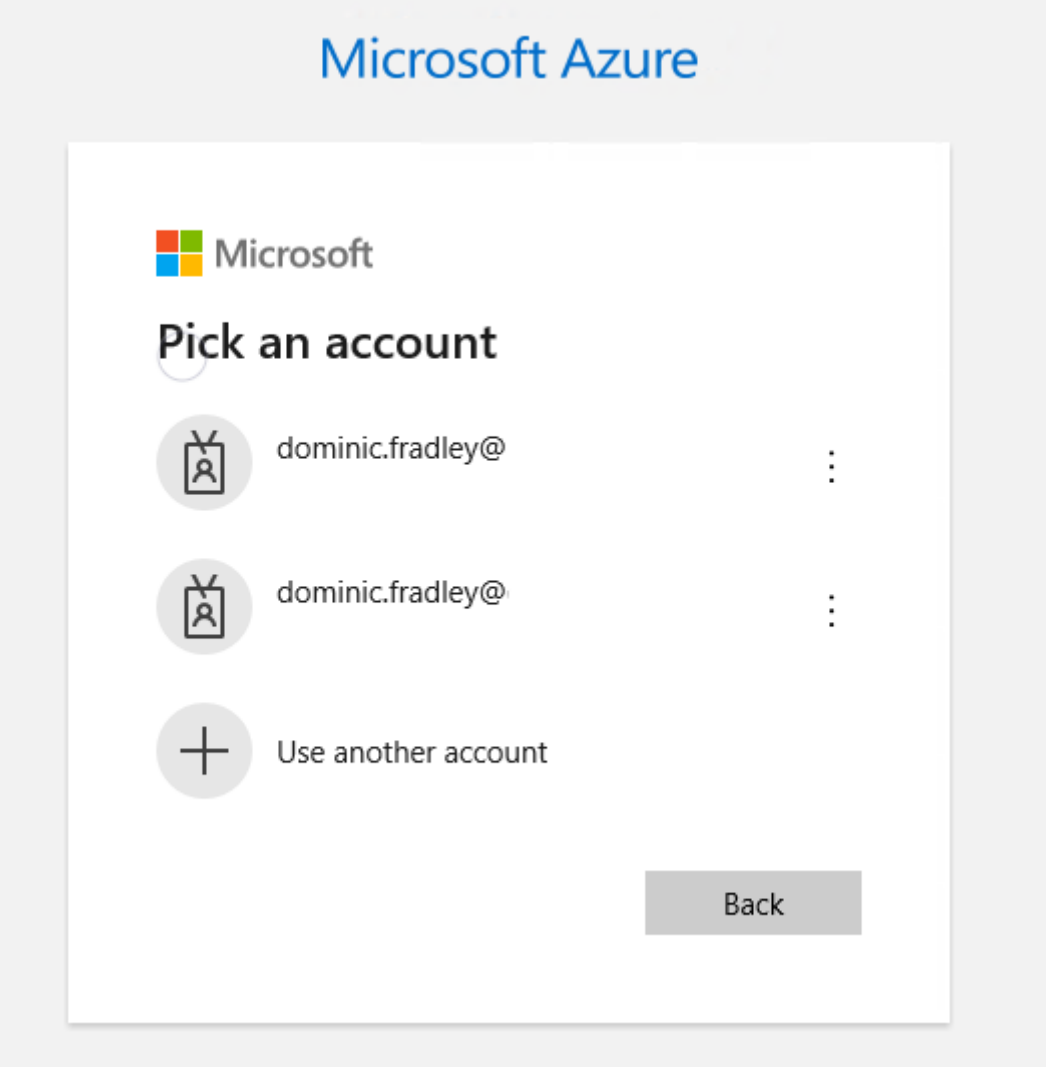
We may now run the script to begin login to Azure.



You will be prompted to login to the Azure portal with your credentials via a webpage pop up. Please note, if you have MFA (Mult-factor Authentication) you will also be required to enter those details as well. Follow the onscreen prompts.

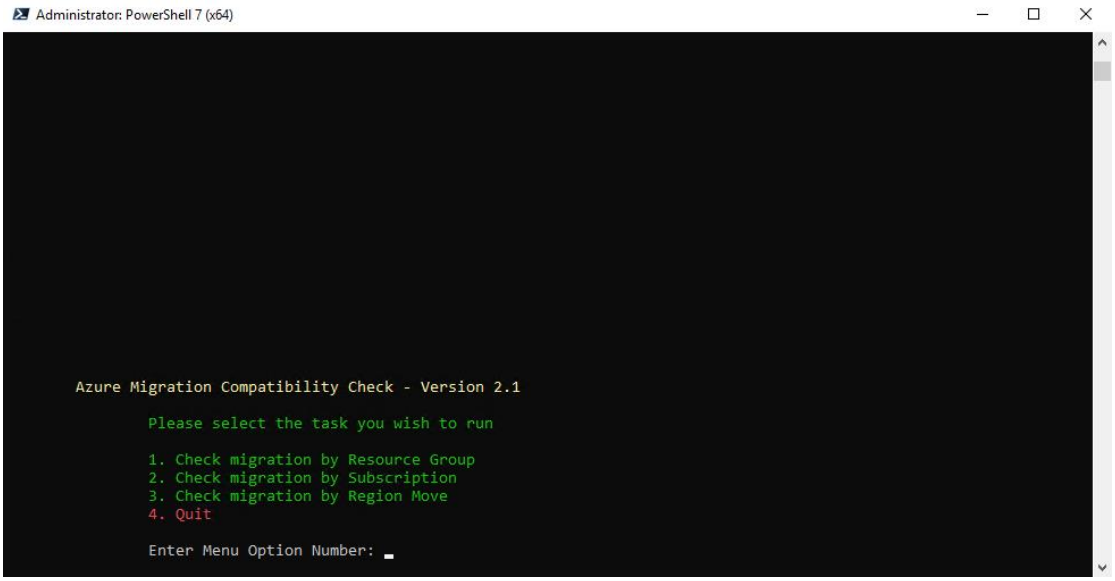


Copyright© 2022 Exponential-e Ltd

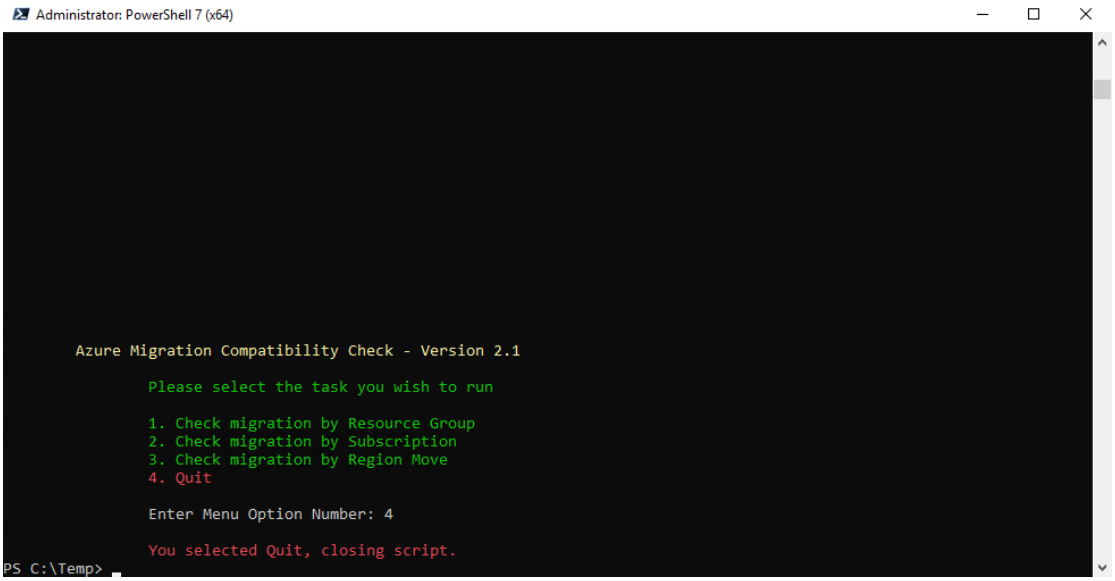


Copyright© 2022 Exponential-e Ltd

Once we are signed in you will be presented with four options. It is recommended to run all sections to gather as much data as possible, however, depending on the project this is not required if you know the type of move you are preparing to undertake.



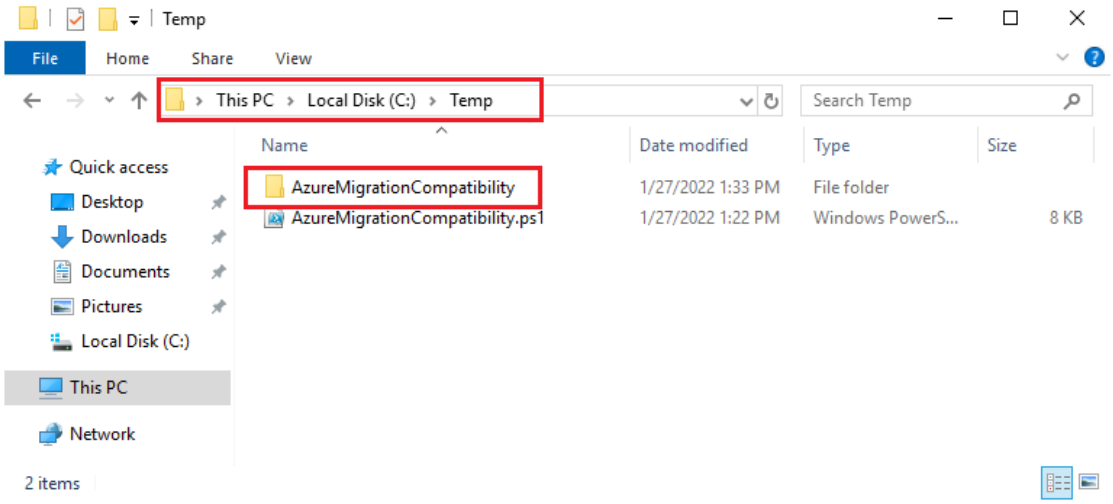
Once you have gathered the required data you may quit the script.



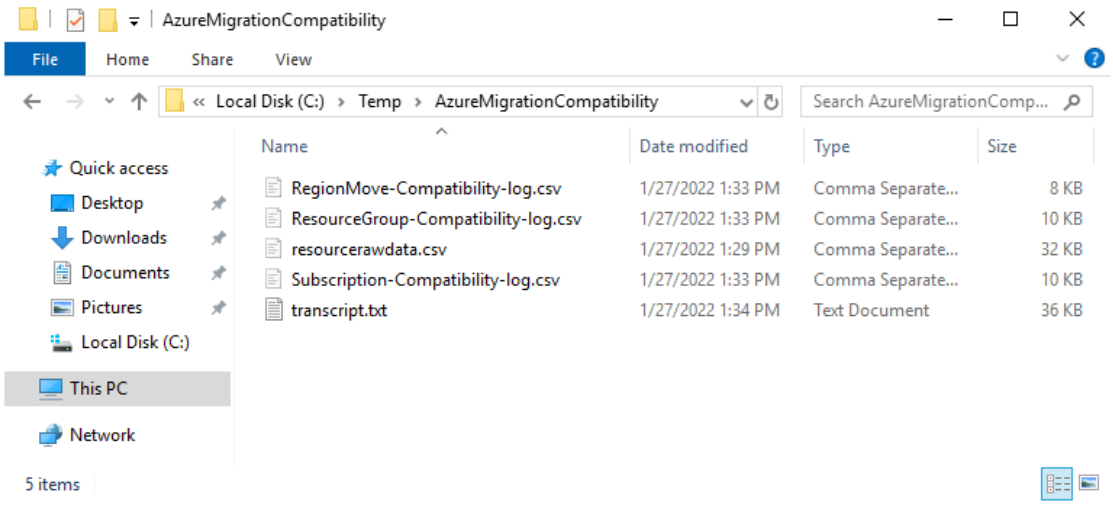
Copyright© 2022 Exponential-e Ltd

2.5 Log Inspection

The script has now completed collecting the required data and you may inspect this before sharing with Exponential-e. To do so, navigate to the Temp folder that was created and open the newly created folder “AzureMigrationCompatibility”.



Inside this directory the script has created some CSV files and a transcript of the running script to assist with troubleshooting.

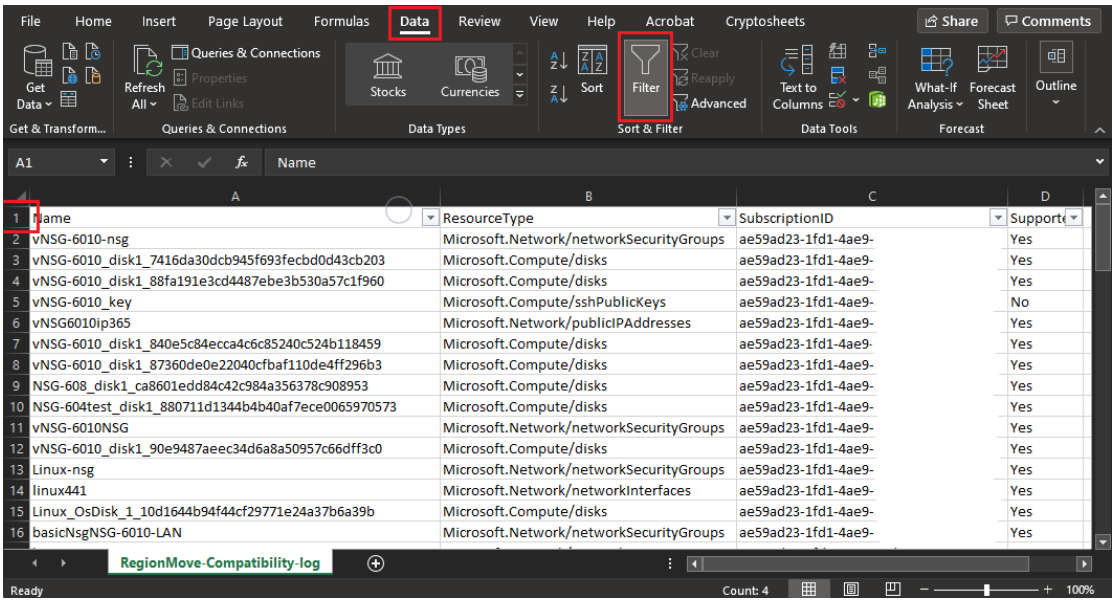


The *-Compatibility-log.csv files contain the necessary data, the transcript.txt contains the script log for troubleshooting, the resourcerawdata.csv can safely be ignored.

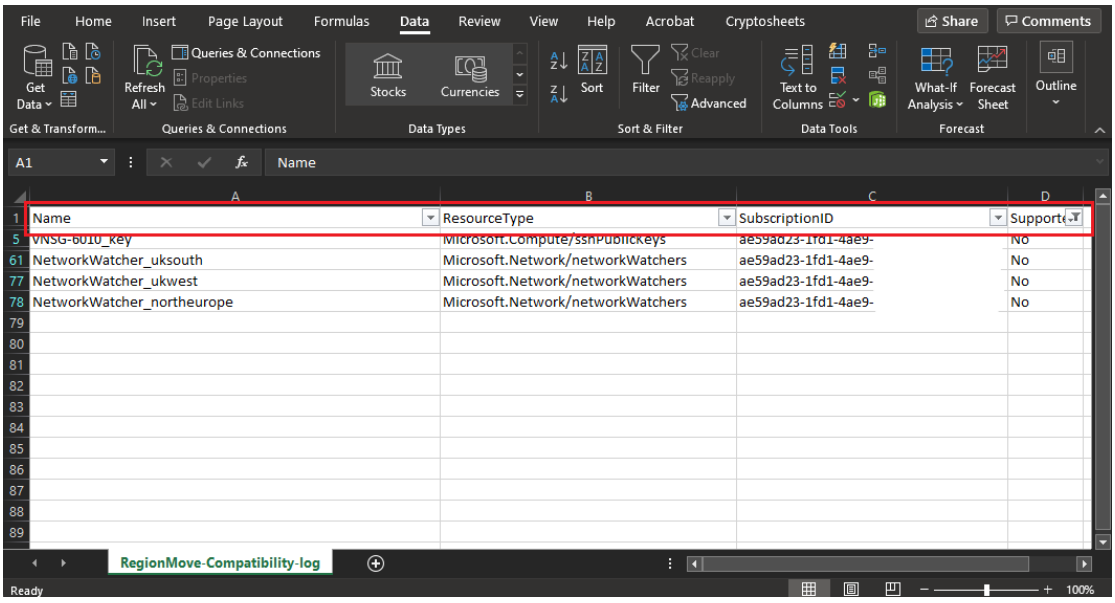
These files can now be opened in Excel to see the output. It is suggested that the data is filtered by the top row, especially if the Azure environment is expansive.

Copyright© 2022 Exponential-e Ltd

1. First, select the “Data” tab from your top ribbon in Excel.
2. Second, select the first row of data from the left.
3. Finally, select the “Filter” option from your top ribbon.



You can now filter out the data to show only the resources that may not be moved, resources by type, resources by SubscriptionId or by name etc. by selecting the appropriate value from the top row and selecting your filter from the dropdown box.



You may now share this information with either your Account Manager or Professional Services resource at Exponential-e for discussion. If any troubleshooting is required, we are also happy to discuss this with you and help with successful completion of the script.

Copyright© 2022 Exponential-e Ltd