Alex Phillips

Kenyatte Simuel

SVAD 260

06/08/2024

PowerShell Best Practices

PowerShell is a powerful tool built into every Windows Operating System since its release in 2006. PowerShell is a cross-platform task automation solution made up of a command-line shell, a scripting language, and a configuration management framework. PowerShell is also capable of running on Linux and macOS. The versatility and availability of PowerShell make it an excellent candidate for automating repetitive tasks in any enterprise.

PowerShell is a command shell that includes many features from other popular shells. Something that makes PowerShell different is that instead of only accept and return text, PowerShell accepts and returns .NET objects. Some features included in the shell are a robust command-line history, tab completion and command prediction, parameter aliases, pipeline for chaining commands, and an in-console help system. With tab completion and command prediction, PowerShell reduces the amount of time it takes to write scripts.

PowerShell is typically used for automating the management of systems. PowerShell can also be used to build, test, and deploy solutions. PowerShell uses the .NET Common Language Runtime, where all inputs and outputs are .NET objects. When creating scripts for PowerShell, a best practice is to create a script for a repetitive task with a high margin for human error. People can make mistakes, which lead to lost time and productivity that could be used doing other parts of their job.

PowerShell is optimized to deploy and manage with other Microsoft products such as Azure, Windows, and SQL. PowerShell is also capable of working with third party systems like AWS, VMWare, and Google Cloud. PowerShell Desired Station Configuration is a management framework in PowerShell that enables organizations to manage infrastructure with configuration as code. This means that you can spin up virtual machines, servers, ect and link them to a database all within the same command line interface.

Microsoft has specifically recommended best practices for PowerShell. One best practice is to use a specific noun for a cmdlet name. Nouns used in cmdlet naming need to be very specific so that the user can discover your cmdlets. If a noun references a server that is running an instance of SQL, use a noun such as “SQLServer”. Combining specific nouns and the short list of approved verbs allow the user to discover and anticipate functionality and avoid duplication among cmdlet names.

Another recommendation is to use standard parameter names. Cmdlets need to use standard parameter names so that the user can determine what a parameter means. If a specific parameter name is required, use a standard parameter name, and then specify a specific name as an alias. An example of this practice in action is the Get-Service cmdlet having a parameter that has a generic name (Name) and a more specific alias (ServiceName).

PowerShell is a versatile tool and an asset for any organization to take advantage of. PowerShell can be used to spin up VM’s in Azure, interact with SQL databases, and make changes and modify active directory profiles. PowerShell is a cross-platform task automation solution made up of a command-line shell, a scripting language, and a configuration management framework that can be used to effectively automate repetitive tasks.

Works Cited:

Microsoft. "Overview of PowerShell." Microsoft Learn, https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.4. Accessed 8 June 2024.

"PowerShell." Wikipedia, 28 May 2024, https://en.wikipedia.org/wiki/PowerShell. Accessed 8 June 2024.

Microsoft. "Strongly Encouraged Development Guidelines for PowerShell Cmdlets." Microsoft Learn, https://learn.microsoft.com/en-us/powershell/scripting/developer/cmdlet/strongly-encouraged-development-guidelines?view=powershell-7.4. Accessed 8 June 2024.