



Leveraging PowerShell on AWS

Siavash Irani

Overview

- What is PowerShell
- Windows PowerShell and PowerShell Core
- AWS Toolkit for PowerShell
- Discoverability
- Get-Help
- Working with Credentials
- PowerShell DSC
- AWS Systems Manager

What is PowerShell?!

- Task automation and configuration management shell
- Exposes cmdlets
- Each cmdlet is a specialized .Net class implementing a particular operation
- Sets of cmdlets can form scripts
- All cmdlets are named using the same convention: verb-noun
- Can directly invoke other .Net classes
- Can be extended by developing new modules and cmdlets



Windows PowerShell and PowerShell Core



Windows PowerShell

- .Net Framework
- Windows
- PowerShell.exe



PowerShell Core

- .Net Core
- Cross-platform
- pwsh

You can run both PowerShell and PowerShell Core side-by-side on the same Windows instance

AWS APIs and SDK



Android



iOS



Java



JavaScript



.NET



Node.js



PHP



Python (boto)



Ruby



Xamarin



AWS CLI



AWS Toolkit for
Eclipse



AWS Toolkit for
Visual Studio



AWS Tools for
Windows
PowerShell


AWS Toolkit for PowerShell




- [AWS.Tools](#) - The modularized version of AWS Tools for PowerShell. Each AWS service is supported by its own individual, small module, with shared support modules.
- [AWSPowerShell.NetCore](#) - The single, large-module version of AWS Tools for PowerShell. All AWS services are supported by this single, large module.
- [AWSPowerShell](#) - The legacy Windows-specific, single, large-module version of AWS Tools for PowerShell. All AWS services are supported by this single, large module.

AWS Tools for PowerShell


The AWS Tools for PowerShell let developers and administrators manage their AWS services and resources in the PowerShell scripting environment. Now you can manage your AWS resources with the same PowerShell tools you use to manage your Windows, Linux, and MacOS environments.




Getting Started »



Cmdlet Reference »



Github Repository »



Developer Blog »

Install from Powershell Gallery


- [AWS Tools for PowerShell \(modular, recommended\)](#)
- [AWS Tools for PowerShell \(bundle\)](#)
- [AWS Tools for PowerShell \(bundle, legacy\)](#)

Download Zip

- [AWS Tools for PowerShell \(modular, recommended\)](#)
- [AWS Tools for PowerShell \(bundle\)](#)
- [AWS Tools for PowerShell \(bundle, legacy\)](#)

PowerShell Gallery

Home Items Publish Statistics Documentation Status Search



AWSPowerShell.NetCore 3.3.343.0

The AWS Tools for PowerShell Core lets developers and administrators manage the Core scripting environment.

Inspect

```
PS> Save-Module -Name AWSPowerShell.NetCore -Path <path>
```

Install

```
PS> Install-Module -Name AWSPowerShell.NetCore
```

37,449 Downloads

124

Discoverability

```
PS C:\Users\sepehrs> Get-Command -Module AWSPowerShell -Verb remove -Noun *wm*
```

CommandType	Name	Version	Source
Cmdlet	Remove-WMAlias	3.3.313.0	AWSPowerShell
Cmdlet	Remove-WMDelegateFromResource	3.3.313.0	AWSPowerShell
Cmdlet	Remove-WMFromWorkMail	3.3.313.0	AWSPowerShell
Cmdlet	Remove-WMGroup	3.3.313.0	AWSPowerShell
Cmdlet	Remove-WMMailboxPermission	3.3.313.0	AWSPowerShell
Cmdlet	Remove-WMMemberFromGroup	3.3.313.0	AWSPowerShell
Cmdlet	Remove-WMResource	3.3.313.0	AWSPowerShell
Cmdlet	Remove-WMUser	3.3.313.0	AWSPowerShell

Windows PowerShell

Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

```
PS C:\Users\sepehrs> Get-Command -Verb start -Noun *ec2*
```

CommandType	Name	Version	Source
Cmdlet	Start-EC2Instance	3.3.313.0	AWSPowerShell
Cmdlet	Start-EC2InstanceMonitoring	3.3.313.0	AWSPowerShell

Windows PowerShell

Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

```
PS C:\Users\sepehrs> Get-Command -Noun *orgaccount*
```

CommandType	Name	Version	Source
Cmdlet	Get-ORGAccount	3.3.313.0	AWSPowerShell
Cmdlet	Get-ORGAccountCreationStatus	3.3.313.0	AWSPowerShell
Cmdlet	Get-ORGAccountCreationStatusList	3.3.313.0	AWSPowerShell
Cmdlet	Get-ORGAccountForParent	3.3.313.0	AWSPowerShell
Cmdlet	Get-ORGAccountHandshakeList	3.3.313.0	AWSPowerShell
Cmdlet	Get-ORGAccountList	3.3.313.0	AWSPowerShell
Cmdlet	Move-ORGAccount	3.3.313.0	AWSPowerShell
Cmdlet	New-ORGAccount	3.3.313.0	AWSPowerShell
Cmdlet	New-ORGAccountInvitation	3.3.313.0	AWSPowerShell
Cmdlet	Remove-ORGAccountFromOrganization	3.3.313.0	AWSPowerShell

Get-Help

PowerShell-6.0.0

PowerShell v6.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

<https://aka.ms/pscore6-docs>
Type 'help' to get help.

PS C:\Program Files\PowerShell\6.0.0> **Get-Help** Start-EC2Instance

NAME
Start-EC2Instance

SYNOPSIS
Calls the Amazon Elastic Compute Cloud StartInstances API operation.

SYNTAX
Start-EC2Instance [[-InstanceId] <System.Object[]>] [-AdditionalInfo <System.String[]>] [-Automation <System.Management.Automation.SwitchParameter>] [<CommonParameters>]

DESCRIPTION
Starts an Amazon EBS-backed instance that you've previously stopped. Instances that use Amazon EBS volumes as their root devices can be quickly stopped is stopped, the compute resources are released and you are not billed for instance partition Amazon EBS volume remains and continues to persist your data, and you ar

PS C:\Program Files\PowerShell\6.0.0> **Get-Help** Start-EC2Instance -Examples

NAME
Start-EC2Instance

SYNOPSIS
Calls the Amazon Elastic Compute Cloud StartInstances API operation.

----- EXAMPLE 1 -----

PS C:\> Start-EC2Instance -InstanceId i-12345678

CurrentState	InstanceId	PreviousState
Amazon.EC2.Model.InstanceState	i-12345678	Amazon.EC2.Model.InstanceState

Description

This example starts the specified instance.

----- EXAMPLE 2 -----

PS C:\> @("i-12345678" "i-76543210") | Start-EC2Instance

Working with Credentials

1) Passed as Parameters (Don't do this)

- `PS C:\> Get-EC2Instance -AccessKey XXX -SecretKey YYY`

2) Environment Variables

- `PS C:\> set AWS_ACCESS_KEY_ID=XXX`
- `PS C:\> set AWS_SECRET_ACCESS_KEY=YYY`

3) Windows Credential Store (Encrypted)

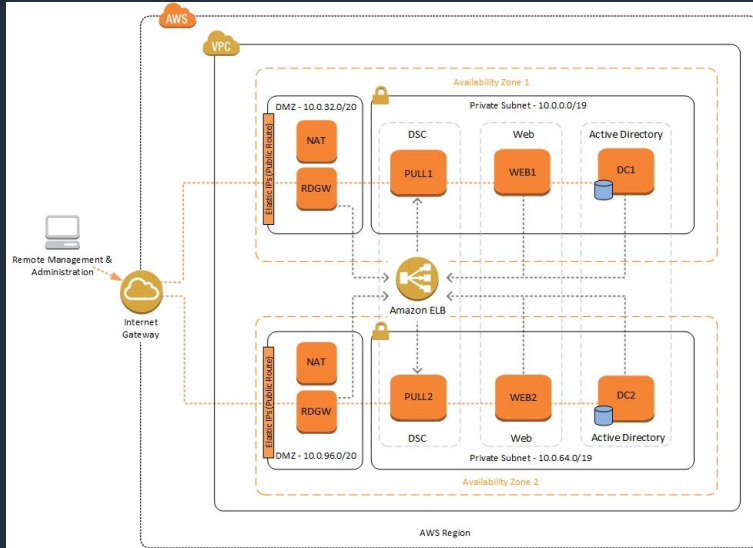
- `PS C:\> Set-AWSCredential -AccessKey XXX -SecretKey YYY`

4) AWS Config File (Shared by CLI)

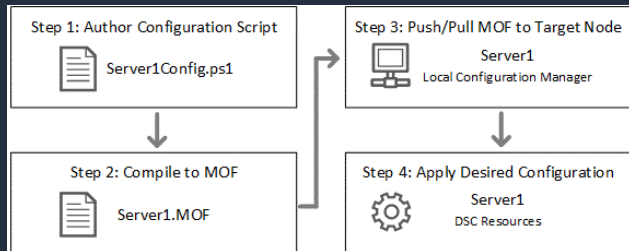
- `C:\Users\Brian\.aws\credentials`

5) EC2 Instance Profile

PowerShell DSC



- Desired state as declarative language syntax
- Distributed systems drift management
- Applicable on Windows and Linux
- Available in AWS as a Quick Start template
- Pull mode: nodes download configuration from pull servers
- Push mode: nodes push configuration to themselves



DSC Example

AWS Systems Manager Run Command



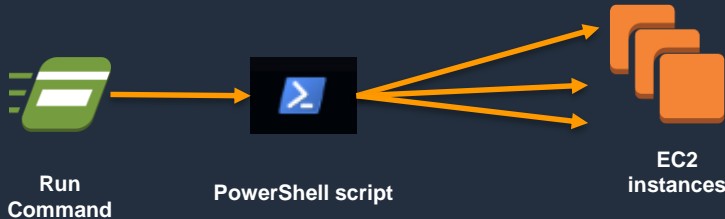
Amazon Systems Manager



AWS IAM



Amazon EC2



- IT tasks and operations at **scale**
- Improve **security** posture
- Share **best practices** across org
- **Operational excellence**

- Run PowerShell scripts on EC2 instances at scale
- Run PowerShell scripts securely using integration with AWS IAM
- No remote connection needed to EC2 instances
- See status reports from CloudWatch
- Use CloudWatch rules and events to automate tasks
- Integration with AWS Systems Manager Automation
- Integration with AWS Systems Manager State Manager
- Auditability through AWS CloudTrail
- Supports EC2 instances in AWS, VMs and devices on-premises or in other clouds

AWS Systems Manager Session Manager



Amazon Systems Manager



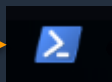
AWS IAM



Amazon EC2



Administrator



PowerShell script



EC2 instances

- Secure interactive shell
 - Access through browser or CLI
 - Access control through IAM
 - Auditability and time-bound
 - **No local or AD users needed**
-
- Possibility to eliminate need for AD infrastructure
 - Improve security posture by closing all SSH and RDP ports
 - Grant time-bound interactive session access using IAM's Date Condition Operators
 - More and easier automation: Include shell commands in infrastructure scripts (aws ssm ...)