Windows PowerShell Get-Help on Cmdlet 'Invoke-AzVMRunCommand'

PS:\>Get-HELP Invoke-AzVMRunCommand -Full

NAME

Invoke-AzVMRunCommand

SYNOPSIS

Run a command on the VM.

SYNTAX

Invoke-AzVMRunCommand [-ResourceGroupName] <System.String> [-VMName] <System.String> [-AsJob] -CommandId <System.String> [-DefaultProfile

<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Parameter <System.Collections.Hashtable>] [-ScriptPath <System.String>]

[-ScriptString <System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Invoke-AzVMRunCommand [-ResourceId] <System.String> [-AsJob] -CommandId <System.String> [-DefaultProfile

<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Parameter <System.Collections.Hashtable>] [-ScriptPath <System.String>]

[-ScriptString <System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Invoke-AzVMRunCommand [-VM] <Microsoft.Azure.Commands.Compute.Models.PSVirtualMachine> [-AsJob]

-CommandId <System.String> [-DefaultProfile

<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Parameter

<System.Collections.Hashtable>] [-ScriptPath <System.String>]

[-ScriptString <System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

Invoke a run command on the VM.

PARAMETERS

-AsJob <System.Management.Automation.SwitchParameter>

Run cmdlet in the background and return a job object to track progress.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-CommandId <System.String>

The run command ID.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>

The credentials, account, tenant, and subscription used for communication with Azure.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Parameter <System.Collections.Hashtable>

The run command parameters. Specify parameters as key/value pairs to be passed at script execution.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ResourceGroupName <System.String>

The name of the resource group.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ResourceId <System.String>

The resource ID for the VM.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ScriptPath <System.String>

Path of the script to be executed. When this value is given, the given script will override the default script of the command. Path should point to a file from a

local file system. The command will load it and send it for execution.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ScriptString <System.String>

The script to be executed as a string.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-VM <Microsoft.Azure.Commands.Compute.Models.PSVirtualMachine>

The PS virtual machine object.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-VMName <System.String>

The name of the virtual machine.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.String

OUTPUTS

Microsoft.Azure.Commands.Compute.Automation.Models.PSRunCommandResult

NOTES

Example 1: Invoke a command on Windows - Using ScriptPath parameter when the script resides on the remote Windows VM

Invoke-AzVMRunCommand -ResourceGroupName 'rgname' -VMName 'vmname' -CommandId 'RunPowerShellScript' -ScriptPath 'sample.ps1' -Parameter @{param1 = "var1"; param2 =
"var2"}

Invoke a run command 'RunPowerShellScript' with overriding the script 'sample.ps1' on a Windows VM named 'vmname' in resource group 'rgname'. Var1 and var2 are
defined as parameters in the sample.ps1. Parameter value can be string type only and script is responsible for converting them to other types if needed.

Example 2: Invoke a command on Windows - Using ScriptString parameter to execute cmdlet on the Windows VM

Invoke-AzVMRunCommand -ResourceGroupName 'rgname' -VMName 'vmname' -CommandId 'RunPowerShellScript' -ScriptString "Set-TimeZone -Name 'Coordinated Universal Time'
-PassThru"

This command invokes a run command 'RunShellScript' that will execute the cmdlet Set-TimeZone with it's associated

parameters. This example is useful when you want to

execute short commands on Windows VM.

Example 3: Invoke a command on Windows - Using ScriptString parameter to run script blocks on the Windows VM

```
$ScriptBlock = {
    param(
        [string] $NewTimeZone,
        [string] $NewDate
        )
    Set-TimeZone -Id $NewTimeZone
    Set-Date -Date [DateTime]$NewDate
}

$Script = [scriptblock]::create($ScriptBlock)

 Invoke-AzVMRunCommand -ResourceGroupName 'rgname' -VMName 'vmname' -CommandId 'RunPowerShellScript'
-ScriptString $Script -Parameter @{'NewTimeZone' = "UTC";
'NewDate' = "Dec-8"}
```

This command invokes a run command 'RunShellScript' that executes a script block on a remote Windows VM named 'vmname'. The script block way allows you to execute

multiple cmdlets with parameters in a single invoke and it also saves time on invoking multiple run commands for different cmdlets. Parameter value(s) can be of

string type only.

---------- Example 4: Invoke a command on Linux ----------

```
export param1=var1 param2=var2
set -- var1 var2 var3
```

Invoke-AzVMRunCommand -ResourceGroupName 'rgname' -Name 'vmname' -CommandId 'RunShellScript' -ScriptPath 'sample.bash' -Parameter @{"param1" = "var1";"param2" =

"var2"}

echo This is a sample bash script

echo Usage 1: Ordered parameters: $0 $1

echo Usage 2: Named exports: $var1 $var2

This command invokes a run command 'RunShellScript' with overriding the script 'sample.bash' on a Linux VM named 'vmname'. Var1 and var2 are defined as parameters in

the sample.bash.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.compute/invoke-azvmruncommand