Windows PowerShell Get-Help on Cmdlet 'New-AzDisk'

PS:\>Get-HELP New-AzDisk -Full

NAME

New-AzDisk

SYNOPSIS

Creates a managed disk.

SYNTAX

New-AzDisk [-ResourceGroupName] <System.String> [-DiskName] <System.String> [-Disk] <Microsoft.Azure.Commands.Compute.Automation.Models.PSDisk> [-AsJob] [-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The New-AzDisk cmdlet creates a managed disk.

PARAMETERS

-AsJob <System.Management.Automation.SwitchParameter>

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

Required? false

Position? named

Default value False

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

-Disk <Microsoft.Azure.Commands.Compute.Automation.Models.PSDisk>

Specifies a local disk object.

Required? true

Position? 2

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-DiskName <System.String>

Specifies the name of a disk.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ResourceGroupName <System.String>

Specifies the name of a resource group.

Required? true

Position? 0

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

Microsoft.Azure.Commands.Compute.Automation.Models.PSDisk

OUTPUTS

Microsoft.Azure.Commands.Compute.Automation.Models.PSDisk

NOTES

---------- Example 1 ----------

$diskconfig = New-AzDiskConfig -Location 'Central US' -DiskSizeGB 5 -SkuName Standard_LRS -OsType Windows -CreateOption Empty -EncryptionSettingsEnabled $true;

$secretUrl = 'https://myvault.vault-int.azure-int.net/secrets/123/';

$secretId = '/subscriptions/0000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup01/providers/Microsoft.KeyVault/vaults/TestVault123';

$keyUrl = 'https://myvault.vault-int.azure-int.net/keys/456';

$keyId = '/subscriptions/0000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup01/providers/Microsoft.KeyVault/vaults/TestVault456';

$diskconfig = Set-AzDiskDiskEncryptionKey -Disk $diskconfig -SecretUrl $secretUrl -SourceVaultId $secretId;

$diskconfig = Set-AzDiskKeyEncryptionKey -Disk $diskconfig -KeyUrl $keyUrl -SourceVaultId $keyId;

New-AzDisk -ResourceGroupName 'ResourceGroup01' -DiskName 'Disk01' -Disk $diskconfig;

The first command creates a local empty disk object with size 5GB in Standard_LRS storage account type. It also sets Windows OS type and enables encryption settings.

The second and third commands set the disk encryption key and key encryption key settings for the disk object. The last command takes the disk object and creates a

disk with name 'Disk01' in resource group 'ResourceGroup01'.

---------- Example 2 ----------

$diskconfig = New-AzDiskConfig -Location 'Central US' -DiskSizeGB 5 -SkuName Standard_LRS -OsType Windows -CreateOption Empty -EncryptionSettingsEnabled $true;

$diskConfig.EncryptionSettingsCollection = New-Object Microsoft.Azure.Management.Compute.Models.EncryptionSettingsCollection

$encryptionSettingsElement1 = New-Object Microsoft.Azure.Management.Compute.Models.EncryptionSettingsElement

$encryptionSettingsElement1.DiskEncryptionKey = New-Object Microsoft.Azure.Management.Compute.Models.KeyVaultAndSecretReference

$encryptionSettingsElement1.DiskEncryptionKey.SourceVault = New-Object Microsoft.Azure.Management.Compute.Models.SourceVault

$encryptionSettingsElement1.DiskEncryptionKey.SourceVault.Id = $disk_encryption_key_id_1

$encryptionSettingsElement1.DiskEncryptionKey.SecretUrl = $disk_encryption_secret_url_1

$encryptionSettingsElement1.KeyEncryptionKey = New-Object Microsoft.Azure.Management.Compute.Models.KeyVaultAndKeyReference

$encryptionSettingsElement1.KeyEncryptionKey.SourceVault = New-Object Microsoft.Azure.Management.Compute.Models.SourceVault

$encryptionSettingsElement1.KeyEncryptionKey.SourceVault.Id = $key_encryption_key_id_1

$encryptionSettingsElement1.KeyEncryptionKey.KeyUrl = $key_encryption_key_url_1

$encryptionSettingsElement2 = New-Object Microsoft.Azure.Management.Compute.Models.EncryptionSettingsElement

$encryptionSettingsElement2.DiskEncryptionKey = New-Object

Microsoft.Azure.Management.Compute.Models.KeyVaultAndSecretReference

$encryptionSettingsElement2.DiskEncryptionKey.SourceVault = New-Object Microsoft.Azure.Management.Compute.Models.SourceVault

$encryptionSettingsElement2.DiskEncryptionKey.SourceVault.Id = $disk_encryption_key_id_2

$encryptionSettingsElement2.DiskEncryptionKey.SecretUrl = $disk_encryption_secret_url_2

$encryptionSettingsElement2.KeyEncryptionKey = New-Object Microsoft.Azure.Management.Compute.Models.KeyVaultAndKeyReference

$encryptionSettingsElement2.KeyEncryptionKey.SourceVault = New-Object Microsoft.Azure.Management.Compute.Models.SourceVault

$encryptionSettingsElement2.KeyEncryptionKey.SourceVault.Id = $key_encryption_key_id_2

$encryptionSettingsElement2.KeyEncryptionKey.KeyUrl = $key_encryption_key_url_2

$diskConfig.EncryptionSettingsCollection.EncryptionSettings += $encryptionSettingsElement1

$diskConfig.EncryptionSettingsCollection.EncryptionSettings += $encryptionSettingsElement2

New-AzDisk -ResourceGroupName 'ResourceGroup01' -DiskName 'Disk01' -Disk $diskconfig;

The above command creates a disk with two encryption settings.

------ Example 3: Export a gallery image version to disk. ------

$galleryImageVersionID =

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myImageRG/providers/Microsoft.Compute/galleries/myGallery/images/myImage/versions/1.0.0"

$location = "eastus"

$rgName = "eastus"

$region = "eastus"

Export the OS disk

$myDiskName = "myOSDisk"

$imageOSDisk = @{Id = $galleryImageVersionID}

$OSDiskConfig = New-AzDiskConfig -Location $location -CreateOption "FromImage" -GalleryImageReference

```
$imageOSDisk
    New-AzDisk -ResourceGroupName $rgName -DiskName $myDiskName -Disk $OSDiskConfig

     # Export any data disk from the image version
    $myDiskName = "myDataDisk"
    $imageDataDisk = @{Id = $galleryImageVersionID; Lun=1}
        $dataDiskConfig = New-AzDiskConfig -Location $location -CreateOption "FromImage" -GalleryImageReference
$imageDataDisk
    New-AzDisk -ResourceGroupName $rgName -DiskName $myDiskName -Disk $dataDiskConfig
```

This example exports a disk from the image version. To export a data disk from the image version, include the LUN number of the data disk to export from the image

version.

Example 4: Create a disk with HyperVGeneration V2 and TrustedLaunch enabled by default.

```
    $rgname = <Resource Group Name>;
    $loc = <Azure Region>;
    New-AzResourceGroup -Name $rgname -Location $loc -Force;

    $diskname = "d" + $rgname;

            $image = Get-AzVMImage -Skus 2022-datacenter-azure-edition -Offer WindowsServer -PublisherName
MicrosoftWindowsServer -Location $loc -Version latest;
        $diskconfig = New-AzDiskConfig -DiskSizeGB 127 -AccountType Premium_LRS -OsType Windows -CreateOption
FromImage -Location $loc;

    $diskconfig = Set-AzDiskImageReference -Disk $diskconfig -Id $image.Id;

    $disk = New-AzDisk -ResourceGroupName $rgname -DiskName $diskname -Disk $diskconfig;
    # Validate $disk.SecurityProfile.securityType is TrustedLaunch.
    # Validate $disk.HyperVGeneration is V2.
```

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.compute/new-azdisk