



Windows PowerShell Get-Help on Cmdlet 'Set-AzStorageAccount'

PS:\>Get-HELP Set-AzStorageAccount -Full

NAME

Set-AzStorageAccount

SYNOPSIS

Modifies a Storage account.

SYNTAX

```
Set-AzStorageAccount [-ResourceGroupName] <System.String> [-Name] <System.String> [-AccessTier {Hot | Cool}]
[-ActiveDirectoryAccountType <System.String>]
[-ActiveDirectoryAzureStorageSid <System.String>] [-ActiveDirectoryDomainGuid <System.String>]
[-ActiveDirectoryDomainName <System.String>] [-ActiveDirectoryDomainSid
<System.String>] [-ActiveDirectoryForestName <System.String>] [-ActiveDirectoryNetBiosDomainName
<System.String>] [-ActiveDirectorySamAccountName <System.String>]
[-AllowBlobPublicAccess <System.Boolean>] [-AllowCrossTenantReplication <System.Boolean>] [-AllowedCopyScope
<System.String>] [-AllowSharedKeyAccess <System.Boolean>]
[-AsJob] [-AssignIdentity] [-CustomDomainName <System.String>] [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>]
[-DefaultSharePermission {None | StorageFileDataSmbShareContributor |
StorageFileDataSmbShareReader | StorageFileDataSmbShareElevatedContributor}]
```

-EnableActiveDirectoryDomainServicesForFile <System.Boolean> [-EnableHttpsTrafficOnly
 <System.Boolean>] [-EnableLargeFileShare] [-EnableLocalUser <System.Boolean>] [-EnableSftp <System.Boolean>]
 [-Force] [-IdentityType {SystemAssigned | UserAssigned |
 SystemAssignedUserAssigned | None}] [-ImmutabilityPeriod <System.Int32>] [-ImmutabilityPolicyState <System.String>]
 [-KeyExpirationPeriodInDay <System.Int32>]
 [-KeyVaultFederatedClientId <System.String>] [-KeyVaultUserAssignedIdentityId <System.String>] [-MinimumTlsVersion
 {TLS1_0 | TLS1_1 | TLS1_2}] [-NetworkRuleSet
 <Microsoft.Azure.Commands.Management.Storage.Models.PSNetworkRuleSet>] [-PublicNetworkAccess
 <System.String>] [-PublishInternetEndpoint <System.Boolean>]
 [-PublishMicrosoftEndpoint <System.Boolean>] [-RoutingChoice {MicrosoftRouting | InternetRouting}]
 [-SasExpirationPeriod <System.TimeSpan>] [-SkuName {Standard_LRS |
 Standard_ZRS | Standard_GRS | Standard_RAGRS | Premium_LRS | Standard_GZRS | Standard_RAGZRS}] [-Tag
 <System.Collections.Hashtable>] [-UpgradeToStorageV2]
 [-UserAssignedIdentityId <System.String>] [-UseSubDomain <System.Nullable`1[System.Boolean]>] [-Confirm] [-WhatIf]
 [<CommonParameters>]

Set-AzStorageAccount [-ResourceGroupName] <System.String> [-Name] <System.String> [-AccessTier {Hot | Cool}]
 [-ActiveDirectoryDomainGuid <System.String>]
 [-ActiveDirectoryDomainName <System.String>] [-AllowBlobPublicAccess <System.Boolean>]
 [-AllowCrossTenantReplication <System.Boolean>] [-AllowedCopyScope
 <System.String>] [-AllowSharedKeyAccess <System.Boolean>] [-AsJob] [-AssignIdentity] [-CustomDomainName
 <System.String>] [-DefaultProfile
 <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>]
 [-DefaultSharePermission {None | StorageFileDataSmbShareContributor |
 StorageFileDataSmbShareReader | StorageFileDataSmbShareElevatedContributor}]
 [-EnableAzureActiveDirectoryKerberosForFile <System.Boolean>] [-EnableHttpsTrafficOnly
 <System.Boolean>] [-EnableLargeFileShare] [-EnableLocalUser <System.Boolean>] [-EnableSftp <System.Boolean>]
 [-Force] [-IdentityType {SystemAssigned | UserAssigned |
 SystemAssignedUserAssigned | None}] [-ImmutabilityPeriod <System.Int32>] [-ImmutabilityPolicyState <System.String>]
 [-KeyExpirationPeriodInDay <System.Int32>]
 [-KeyVaultFederatedClientId <System.String>] [-KeyVaultUserAssignedIdentityId <System.String>] [-MinimumTlsVersion
 {TLS1_0 | TLS1_1 | TLS1_2}] [-NetworkRuleSet
 <Microsoft.Azure.Commands.Management.Storage.Models.PSNetworkRuleSet>] [-PublicNetworkAccess

```

<System.String>] [-PublishInternetEndpoint <System.Boolean>]
    [-PublishMicrosoftEndpoint <System.Boolean>] [-RoutingChoice {MicrosoftRouting | InternetRouting}]
[-SasExpirationPeriod <System.TimeSpan>] [-SkuName {Standard_LRS |
    Standard_ZRS | Standard_GRS | Standard_RAGRS | Premium_LRS | Standard_GZRS | Standard_RAGZRS}] [-Tag
<System.Collections.Hashtable>] [-UpgradeToStorageV2]
    [-UserAssignedIdentityId <System.String>] [-UseSubDomain <System.Nullable`1[System.Boolean]>] [-Confirm] [-WhatIf]
[<CommonParameters>]

Set-AzStorageAccount [-ResourceGroupName] <System.String> [-Name] <System.String> [-AccessTier {Hot | Cool}]
[-AllowBlobPublicAccess <System.Boolean>]
    [-AllowCrossTenantReplication <System.Boolean>] [-AllowedCopyScope <System.String>] [-AllowSharedKeyAccess
<System.Boolean>] [-AsJob] [-AssignIdentity]
    [-CustomDomainName <System.String>] [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>]
    [-DefaultSharePermission {None | StorageFileDataSmbShareContributor | StorageFileDataSmbShareReader |
StorageFileDataSmbShareElevatedContributor}]
    [-EnableAzureActiveDirectoryDomainServicesForFile <System.Boolean>] [-EnableHttpsTrafficOnly <System.Boolean>]
[-EnableLargeFileShare] [-EnableLocalUser
    <System.Boolean>] [-EnableSftp <System.Boolean>] [-Force] [-IdentityType {SystemAssigned | UserAssigned |
SystemAssignedUserAssigned | None}] [-ImmutabilityPeriod
    <System.Int32>] [-ImmutabilityPolicyState <System.String>] [-KeyExpirationPeriodInDay <System.Int32>]
[-KeyVaultFederatedClientId <System.String>]
    [-KeyVaultUserAssignedIdentityId <System.String>] [-MinimumTlsVersion {TLS1_0 | TLS1_1 | TLS1_2}]
[-NetworkRuleSet
    <Microsoft.Azure.Commands.Management.Storage.Models.PSNetworkRuleSet>] [-PublicNetworkAccess
<System.String>] [-PublishInternetEndpoint <System.Boolean>]
    [-PublishMicrosoftEndpoint <System.Boolean>] [-RoutingChoice {MicrosoftRouting | InternetRouting}]
[-SasExpirationPeriod <System.TimeSpan>] [-SkuName {Standard_LRS |
    Standard_ZRS | Standard_GRS | Standard_RAGRS | Premium_LRS | Standard_GZRS | Standard_RAGZRS}]
[-StorageEncryption] [-Tag <System.Collections.Hashtable>]
    [-UpgradeToStorageV2] [-UserAssignedIdentityId <System.String>] [-UseSubDomain
<System.Nullable`1[System.Boolean]>] [-Confirm] [-WhatIf] [<CommonParameters>]

```

```

Set-AzStorageAccount [-ResourceGroupName] <System.String> [-Name] <System.String> [-AccessTier {Hot | Cool}]
[-AllowBlobPublicAccess <System.Boolean>]

    [-AllowCrossTenantReplication <System.Boolean>] [-AllowedCopyScope <System.String>] [-AllowSharedKeyAccess
<System.Boolean>] [-AsJob] [-AssignIdentity]

                                [-CustomDomainName <System.String>]                                [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>]

    [-DefaultSharePermission {None | StorageFileDataSmbShareContributor | StorageFileDataSmbShareReader |
StorageFileDataSmbShareElevatedContributor}]

    [-EnableAzureActiveDirectoryDomainServicesForFile <System.Boolean>] [-EnableHttpsTrafficOnly <System.Boolean>]
[-EnableLargeFileShare] [-EnableLocalUser
    <System.Boolean>] [-EnableSftp <System.Boolean>] [-Force] [-IdentityType {SystemAssigned | UserAssigned |
SystemAssignedUserAssigned | None}] [-ImmutabilityPeriod
    <System.Int32>] [-ImmutabilityPolicyState <System.String>] [-KeyExpirationPeriodInDay <System.Int32>] -KeyName
<System.String> [-KeyvaultEncryption]

    [-KeyVaultFederatedClientId <System.String>] -KeyVaultUri <System.String> [-KeyVaultUserAssignedIdentityId
<System.String>] [-KeyVersion <System.String>]

                                [-MinimumTlsVersion {TLS1_0 | TLS1_1 | TLS1_2}]                                [-NetworkRuleSet
<Microsoft.Azure.Commands.Management.Storage.Models.PSNetworkRuleSet>] [-PublicNetworkAccess
    <System.String>] [-PublishInternetEndpoint <System.Boolean>] [-PublishMicrosoftEndpoint <System.Boolean>]
[-RoutingChoice {MicrosoftRouting | InternetRouting}]

    [-SasExpirationPeriod <System.TimeSpan>] [-SkuName {Standard_LRS | Standard_ZRS | Standard_GRS |
Standard_RAGRS | Premium_LRS | Standard_GZRS | Standard_RAGZRS}]

    [-Tag <System.Collections.Hashtable>] [-UpgradeToStorageV2] [-UserAssignedIdentityId <System.String>]
[-UseSubDomain <System.Nullable`1[System.Boolean]>] [-Confirm]

    [-WhatIf] [<CommonParameters>]

```

DESCRIPTION

The Set-AzStorageAccount cmdlet modifies an Azure Storage account. You can use this cmdlet to modify the account type, update a customer domain, or set tags on a Storage account.

PARAMETERS

`-AccessTier <System.String>`

Specifies the access tier of the Storage account that this cmdlet modifies. The acceptable values for this parameter are: Hot and Cool. If you change the access

tier, it may result in additional charges. For more information, see Azure Blob Storage: Hot and cool storage tiers

(<http://go.microsoft.com/fwlink/?LinkId=786482>). If the Storage account has Kind as StorageV2 or BlobStorage, you can specify the AccessTier parameter. If the

Storage account has Kind as Storage, do not specify the AccessTier parameter.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

`-ActiveDirectoryAccountType <System.String>`

Specifies the Active Directory account type for Azure Storage. Possible values include: 'User', 'Computer'.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

`-ActiveDirectoryAzureStorageSid <System.String>`

Specifies the security identifier (SID) for Azure Storage. This parameter must be set when

`-EnableActiveDirectoryDomainServicesForFile` is set to true.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

`-ActiveDirectoryDomainGuid <System.String>`

Specifies the domain GUID. This parameter must be set when `-EnableActiveDirectoryDomainServicesForFile` is set to `true`.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

`-ActiveDirectoryDomainName <System.String>`

Specifies the primary domain that the AD DNS server is authoritative for. This parameter must be set when `-EnableActiveDirectoryDomainServicesForFile` is set to `true`.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

`-ActiveDirectoryDomainSid <System.String>`

Specifies the security identifier (SID). This parameter must be set when `-EnableActiveDirectoryDomainServicesForFile` is set to `true`.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

`-ActiveDirectoryForestName <System.String>`

Specifies the Active Directory forest to get. This parameter must be set when -EnableActiveDirectoryDomainServicesForFile is set to true.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-ActiveDirectoryNetBiosDomainName <System.String>

Specifies the NetBIOS domain name. This parameter must be set when -EnableActiveDirectoryDomainServicesForFile is set to true.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-ActiveDirectorySamAccountName <System.String>

Specifies the Active Directory SAMAccountName for Azure Storage.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-AllowBlobPublicAccess <System.Boolean>

Allow or disallow anonymous access to all blobs or containers in the storage account.

Required?	false
Position?	named

Default value None
Accept pipeline input? False
Accept wildcard characters? false

-AllowCrossTenantReplication <System.Boolean>

Gets or sets allow or disallow cross Microsoft Entra tenant object replication. The default interpretation is true for this property.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-AllowedCopyScope <System.String>

Set restrict copy to and from Storage Accounts within a Microsoft Entra tenant or with Private Links to the same VNet. Possible values include: 'PrivateLink', 'AAD'

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-AllowSharedKeyAccess <System.Boolean>

Indicates whether the storage account permits requests to be authorized with the account access key via Shared Key. If false, then all requests, including shared access signatures, must be authorized with Microsoft Entra ID. The default value is null, which is equivalent to true.

Required? false
Position? named
Default value None

Accept pipeline input? False

Accept wildcard characters? false

-AsJob <System.Management.Automation.SwitchParameter>

Run cmdlet in the background

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-AssignIdentity <System.Management.Automation.SwitchParameter>

Generate and assign a new Storage account Identity for this Storage account for use with key management services like Azure KeyVault.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-CustomDomainName <System.String>

Specifies the name of the custom domain.

Required? false

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

-DefaultSharePermission <System.String>

Default share permission for users using Kerberos authentication if RBAC role is not assigned.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-EnableActiveDirectoryDomainServicesForFile <System.Boolean>

Enable Azure Files Active Directory Domain Service Authentication for the storage account.

Required? true
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-EnableAzureActiveDirectoryDomainServicesForFile <System.Boolean>

Enable Azure Files Active Directory Domain Service Authentication for the storage account.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-EnableAzureActiveDirectoryKerberosForFile <System.Boolean>

Enable Azure Files Active Directory Domain Service Kerberos Authentication for the storage account.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-EnableHttpsTrafficOnly <System.Boolean>

Indicates whether or not the Storage account only enables HTTPS traffic.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-EnableLargeFileShare <System.Management.Automation.SwitchParameter>

Indicates whether or not the storage account can support large file shares with more than 5 TiB capacity. Once the account is enabled, the feature cannot be

disabled. Currently only supported for LRS and ZRS replication types, hence account conversions to geo-redundant accounts would not be possible. Learn more in

<https://go.microsoft.com/fwlink/?linkid=2086047>

Required? false
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-EnableLocalUser <System.Boolean>

Enable local users feature for the Storage account.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-EnableSftp <System.Boolean>

Enable Secure File Transfer Protocol for the Storage account.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-Force <System.Management.Automation.SwitchParameter>

Forces the change to be written to the Storage account.

Required? false
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-IdentityType <System.String>

Set the new Storage Account Identity type, the identity is for use with key management services like Azure KeyVault.

Required? false
Position? named
Default value None
Accept pipeline input? False

Accept wildcard characters? false

-ImmutabilityPeriod <System.Int32>

The immutability period for the blobs in the container since the policy creation in days. This property can only be changed when account is created with

'-EnableAccountLevelImmutability'.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ImmutabilityPolicyState <System.String>

The mode of the policy. Possible values include: 'Unlocked', 'Locked', 'Disabled. Disabled state disable the policy. Unlocked state allows increase and decrease

of immutability retention time and also allows toggling allowProtectedAppendWrites property. Locked state only allows the increase of the immutability retention

time. A policy can only be created in a Disabled or Unlocked state and can be toggled between the two states. Only a policy in an Unlocked state can transition

to a Locked state which cannot be reverted. This property can only be changed when account is created with '-EnableAccountLevelImmutability'.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-KeyExpirationPeriodInDay <System.Int32>

The Key expiration period of this account, it is accurate to days.

Required? false

Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-KeyName <System.String>

If using -KeyvaultEncryption to enable encryption with Key Vault, specify the Keyname property with this option.

Required? true
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-KeyvaultEncryption <System.Management.Automation.SwitchParameter>

Indicates whether or not to use Microsoft KeyVault for the encryption keys when using Storage Service Encryption. If KeyName, KeyVersion, and KeyVaultUri are all set, KeySource will be set to Microsoft.Keyvault whether this parameter is set or not.

Required? false
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-KeyVaultFederatedClientId <System.String>

Set ClientId of the multi-tenant application to be used in conjunction with the user-assigned identity for cross-tenant customer-managed-keys server-side encryption on the storage account.

Required? false
Position? named
Default value None

Accept pipeline input? False

Accept wildcard characters? false

-KeyVaultUri <System.String>

When using Key Vault Encryption by specifying the -KeyvaultEncryption parameter, use this option to specify the URI to the Key Vault.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-KeyVaultUserAssignedIdentityId <System.String>

Set resource id for user assigned Identity used to access Azure KeyVault of Storage Account Encryption, the id must in the storage account's UserAssignIdentityId.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-KeyVersion <System.String>

When using Key Vault Encryption by specifying the -KeyvaultEncryption parameter, use this option to specify the URI to the Key Version.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-MinimumTlsVersion <System.String>

The minimum TLS version to be permitted on requests to storage.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Name <System.String>

Specifies the name of the Storage account to modify.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-NetworkRuleSet <Microsoft.Azure.Commands.Management.Storage.Models.PSNetworkRuleSet>

NetworkRuleSet is used to define a set of configuration rules for firewalls and virtual networks, as well as to set values for network properties such as services

allowed to bypass the rules and how to handle requests that don't match any of the defined rules.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-PublicNetworkAccess <System.String>

Allow or disallow public network access to Storage Account. Possible values include: 'Enabled', 'Disabled'.

Required? false

Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-PublishInternetEndpoint <System.Boolean>

Indicates whether internet routing storage endpoints are to be published

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-PublishMicrosoftEndpoint <System.Boolean>

Indicates whether microsoft routing storage endpoints are to be published

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-ResourceGroupName <System.String>

Specifies the name of the resource group in which to modify the Storage account.

Required? true
Position? 0
Default value None
Accept pipeline input? True (ByPropertyName)
Accept wildcard characters? false

-RoutingChoice <System.String>

Routing Choice defines the kind of network routing opted by the user. Possible values include: 'MicrosoftRouting', 'InternetRouting'

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-SasExpirationPeriod <System.TimeSpan>

The SAS expiration period of this account, it is a timespan and accurate to seconds.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-SkuName <System.String>

Specifies the SKU name of the Storage account. The acceptable values for this parameter are: - Standard_LRS - Locally-redundant storage.

- Standard_ZRS - Zone-redundant storage.

- Standard_GRS - Geo-redundant storage.

- Standard_RAGRS - Read access geo-redundant storage.

- Premium_LRS - Premium locally-redundant storage.

- Standard_GZRS - Geo-redundant zone-redundant storage.

- Standard_RAGZRS - Read access geo-redundant zone-redundant storage.

You cannot change Standard_ZRS and Premium_LRS types to other account types. You cannot change other account types to Standard_ZRS or Premium_LRS.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-StorageEncryption <System.Management.Automation.SwitchParameter>

Indicates whether or not to set the Storage account encryption to use Microsoft-managed keys.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

-Tag <System.Collections.Hashtable>

Key-value pairs in the form of a hash table set as tags on the server. For example:

@{key0="value0";key1=\$null;key2="value2"}

Required?	false
Position?	named
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-UpgradeToStorageV2 <System.Management.Automation.SwitchParameter>

Upgrade Storage account Kind from Storage or BlobStorage to StorageV2.

Required?	false
-----------	-------

Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-UserAssignedIdentityId <System.String>

Set resource ids for the the new Storage Account user assignedd Identity, the identity will be used with key management services like Azure KeyVault.

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-UseSubDomain <System.Nullable`1[System.Boolean]>

Indicates whether to enable indirect CName validation.

Required? false
Position? named
Default value None
Accept pipeline input? False
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

System.Collections.Hashtable

OUTPUTS

Microsoft.Azure.Commands.Management.Storage.Models.PSSStorageAccount

NOTES

----- Example 1: Set the Storage account type -----

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -SkuName  
"Standard_RAGRS"
```

This command sets the Storage account type to Standard_RAGRS.

----- Example 2: Set a custom domain for a Storage account -----

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -CustomDomainName  
"www.contoso.com" -UseSubDomain $true
```

This command sets a custom domain for a Storage account.

----- Example 3: Set the access tier value -----

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -AccessTier Cool
```

The command sets the Access Tier value to be cool.

----- Example 4: Set the custom domain and tags -----

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -CustomDomainName  
"www.domainname.com" -UseSubDomain $true -Tag  
@{tag0="value0";tag1="value1";tag2="value2"}
```

The command sets the custom domain and tags for a Storage account.

----- Example 5: Set Encryption KeySource to Keyvault -----

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -AssignIdentity
```

```
$account = Get-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount"
```

```
$keyVault = New-AzKeyVault -VaultName "MyKeyVault" -ResourceGroupName "MyResourceGroup" -Location  
"EastUS2"
```

```
$key = Add-AzKeyVaultKey -VaultName "MyKeyVault" -Name "MyKey" -Destination 'Software'
```

```
Set-AzKeyVaultAccessPolicy -VaultName "MyKeyVault" -ObjectId $account.Identity.PrincipalId -PermissionsToKeys  
wrapkey,unwrapkey,get
```

```
# In case to enable key auto rotation, don't set KeyVersion
```

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -KeyvaultEncryption  
-KeyName $key.Name -KeyVersion $key.Version -KeyVaultUri  
$keyVault.VaultUri
```

```
# In case to enable key auto rotation after set keyvault proeprites with KeyVersion, can update account by set KeyVersion  
to empty
```

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -KeyvaultEncryption  
-KeyName $key.Name -KeyVersion "" -KeyVaultUri  
$keyVault.VaultUri
```

This command set Encryption KeySource with a new created Keyvault. If want to enable key auto rotation, don't set keyversion when set Keyvault properties for the

first time, or clean up it by set keyvault properties again with keyversion as empty.

```
-- Example 6: Set Encryption KeySource to "Microsoft.Storage" --
```

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -StorageEncryption
```

This command set Encryption KeySource to "Microsoft.Storage"

Example 7: Set NetworkRuleSet property of a Storage account with JSON

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -NetworkRuleSet
(@{bypass="Logging,Metrics";
  ipRules=(@{IPAddressOrRange="20.11.0.0/16";Action="allow"},
    @{IPAddressOrRange="10.0.0.0/7";Action="allow"});
  virtualNetworkRules=(@{VirtualNetworkResourceId="/subscriptions/s1/resourceGroups/g1/providers/Microsoft.Network/virtualNetworks/vnet1/subnets/subnet1";Action="allow"},
    @{VirtualNetworkResourceId="/subscriptions/s1/resourceGroups/g1/providers/Microsoft.Network/virtualNetworks/vnet2/subnets/subnet2";Action="allow"});
  defaultAction="allow"})
```

This command sets NetworkRuleSet property of a Storage account with JSON

Example 8: Get NetworkRuleSet property from a Storage account, and set it to another Storage account

```
$networkRuleSet = (Get-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name
"mystorageaccount").NetworkRuleSet

Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount2" -NetworkRuleSet
$networkRuleSet
```

This first command gets NetworkRuleSet property from a Storage account, and the second command sets it to another Storage account

Example 9: Upgrade a Storage account with Kind "Storage" or "BlobStorage" to "StorageV2" kind Storage account

```
Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount" -UpgradeToStorageV2
```

The command upgrade a Storage account with Kind "Storage" or "BlobStorage" to "StorageV2" kind Storage account.

Example 10: Update a Storage account by enable Azure Files Microsoft Entra Domain Services Authentication and set DefaultSharePermission.

```
$account = Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount"
-EnableAzureActiveDirectoryDomainServicesForFile $true
-DefaultSharePermission StorageFileDataSmbShareContributor
```

```
$account.AzureFilesIdentityBasedAuth
```

```
DirectoryServiceOptions ActiveDirectoryProperties
```

```
DefaultSharePermission
```

```
-----
```

```
-----
```

```
AADDS
```

```
Microsoft.Azure.Commands.Management.Storage.Models.PSActiveDirectoryProperties
```

```
StorageFileDataSmbShareContributor
```

The command update a Storage account by enable Azure Files Microsoft Entra Domain Services Authentication.

Example 11: Update a Storage account by enable Files Active Directory Domain Service Authentication, and then show the File Identity Based authentication setting

```
$account = Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount"
-EnableActiveDirectoryDomainServicesForFile $true `
-ActiveDirectoryDomainName "mydomain.com" `
-ActiveDirectoryNetBiosDomainName "mydomain.com" `
-ActiveDirectoryForestName "mydomain.com" `
-ActiveDirectoryDomainGuid "12345678-1234-1234-1234-123456789012" `
-ActiveDirectoryDomainSid "S-1-5-21-1234567890-1234567890-1234567890" `
-ActiveDirectoryAzureStorageSid "S-1-5-21-1234567890-1234567890-1234567890-1234" `
-ActiveDirectorySamAccountName "samaccountname" `
```

-ActiveDirectoryAccountType Computer

\$account.AzureFilesIdentityBasedAuth.DirectoryServiceOptions

AD

\$account.AzureFilesIdentityBasedAuth.ActiveDirectoryProperties

DomainName : mydomain.com

NetBiosDomainName : mydomain.com

ForestName : mydomain.com

DomainGuid : 12345678-1234-1234-1234-123456789012

DomainSid : S-1-5-21-1234567890-1234567890-1234567890

AzureStorageSid : S-1-5-21-1234567890-1234567890-1234567890-1234

SamAccountName : samaccountname

AccountType : Computer

The command updates a Storage account by enable Azure Files Active Directory Domain Service Authentication, and then shows the File Identity Based authentication

setting

Example 12: Set MinimumTlsVersion, AllowBlobPublicAccess and AllowSharedKeyAccess

```
$account = Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount"
-MinimumTlsVersion TLS1_1 -AllowBlobPublicAccess $false
-AllowSharedKeyAccess $true
```

\$account.MinimumTlsVersion

TLS1_1

\$account.AllowBlobPublicAccess

False

\$a.AllowSharedKeyAccess

True

The command sets MinimumTlsVersion, AllowBlobPublicAccess and AllowSharedKeyAccess, and then show the the 3 properties of the account

Example 13: Update a Storage account with RoutingPreference setting

```
$account = Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount"
-PublishMicrosoftEndpoint $false -PublishInternetEndpoint $true
-RoutingChoice InternetRouting
```

\$account.RoutingPreference

RoutingChoice PublishMicrosoftEndpoints PublishInternetEndpoints

InternetRouting	False	True

\$account.PrimaryEndpoints

Blob : https://mystorageaccount.blob.core.windows.net/

Queue : https://mystorageaccount.queue.core.windows.net/

Table : https://mystorageaccount.table.core.windows.net/

File : https://mystorageaccount.file.core.windows.net/

Web : https://mystorageaccount.z2.web.core.windows.net/

Dfs : https://mystorageaccount.dfs.core.windows.net/

MicrosoftEndpoints :

InternetEndpoints :

{"Blob":"https://mystorageaccount-internetrouting.blob.core.windows.net/","File":"https://mystorageaccount-internetrouting.fil
e.core.windows.net/"

,"Web":"https://mystorageaccount-internetrouting.z2.web.core.windows.net/","Dfs":"https://w

eirp3-internetrouting.dfs.core.windows.net/"}

This command updates a Storage account with RoutingPreference setting: PublishMicrosoftEndpoint as false, PublishInternetEndpoint as true, and RoutingChoice as MicrosoftRouting.

Example 14: Update a Storage account with KeyExpirationPeriod and SasExpirationPeriod

```
$account = Set-AzStorageAccount -ResourceGroupName "myresourcegroup" -Name "mystorageaccount"
-KeyExpirationPeriodInDay 5 -SasExpirationPeriod "1.12:05:06"
-EnableHttpsTrafficOnly $true

$account.KeyPolicy.KeyExpirationPeriodInDays
5

$account.SasPolicy.SasExpirationPeriod
1.12:05:06
```

This command updates a Storage account with KeyExpirationPeriod and SasExpirationPeriod, then show the updated account related properties.

Example 15: Update a Storage account to Keyvault encryption, and access Keyvault with user assigned identity

```
# Create KeyVault (no need if using exist keyvault)

$keyVault = New-AzKeyVault -VaultName $keyvaultName -ResourceGroupName $resourceGroupName -Location
eastus2euap -EnablePurgeProtection

$key = Add-AzKeyVaultKey -VaultName $keyvaultName -Name $keyname -Destination 'Software'

# create user assigned identity and grant access to keyvault (no need if using exist user assigned identity)

$userId = New-AzUserAssignedIdentity -ResourceGroupName $resourceGroupName -Name $userIdName

Set-AzKeyVaultAccessPolicy -VaultName $keyvaultName -ResourceGroupName $resourceGroupName -ObjectId
$userId.PrincipalId -PermissionsToKeys get,wrapkey,unwrapkey
```

-BypassObjectIdValidation

\$useridentityId= \$userId.Id

Update Storage account with Keyvault encryption and access Keyvault with user assigned identity, then show properties

```
$account = Set-AzStorageAccount -ResourceGroupName $resourceGroupName -Name $storageAccountName `
    -IdentityType UserAssigned -UserAssignedIdentityId $useridentityId `
    -KeyVaultUri $keyVault.VaultUri -KeyName $keyname -KeyVaultUserAssignedIdentityId $useridentityId
```

\$account.Encryption.EncryptionIdentity.EncryptionUserAssignedIdentity

/subscriptions/{subscription-id}/resourceGroups/myresourcegroup/providers/Microsoft.ManagedIdentity/userAssignedIdentities/myuserid

\$account.Encryption.KeyVaultProperties

KeyName : wrappingKey

KeyVersion :

KeyVaultUri : https://mykeyvault.vault.azure.net:443

CurrentVersionedKeyIdentifier :

https://mykeyvault.vault.azure.net/keys/wrappingKey/8e74036e0d534e58b3bd84b319e31d8f

LastKeyRotationTimestamp : 4/12/2021 8:17:57 AM

This command first creates a keyvault and a user assigned identity, then updates a storage account with keyvault encryption, the storage access access keyvault with the user assigned identity.

Example 16: Update a Keyvault encrypted Storage account, from access Keyvault with user assigned identity, to access Keyvault with system assigned identity

Assign System identity to the account, and give the system assigned identity acces to the keyvault

\$account = Set-AzStorageAccount -ResourceGroupName \$resourceGroupName -Name \$storageAccountName

```
-IdentityType SystemAssignedUserAssigned
```

```
Set-AzKeyVaultAccessPolicy -VaultName $keyvaultName -ResourceGroupName $resourceGroupName -ObjectId  
$account.Identity.PrincipalId -PermissionsToKeys  
get,wrapkey,unwrapkey -BypassObjectIdValidation
```

```
# Update account from access Keyvault with user assigned identity to access Keyvault with system assigned identity
```

```
$account = Set-AzStorageAccount -ResourceGroupName $resourceGroupName -Name $storageAccountName  
-IdentityType SystemAssignedUserAssigned -KeyName $keyname  
-KeyVaultUri $keyvaultUri -KeyVaultUserAssignedIdentityId ""
```

```
# EncryptionUserAssignedIdentity is empty, so the account access keyvault with system assigned identity
```

```
$account.Encryption.EncryptionIdentity
```

```
EncryptionUserAssignedIdentity
```

```
-----
```

```
$account.Encryption.KeyVaultProperties
```

```
KeyName : wrappingKey
```

```
KeyVersion :
```

```
KeyVaultUri : https://mykeyvault.vault.azure.net:443
```

```
CurrentVersionedKeyIdentifier :
```

```
https://mykeyvault.vault.azure.net/keys/wrappingKey/8e74036e0d534e58b3bd84b319e31d8f
```

```
LastKeyRotationTimestamp : 4/12/2021 8:17:57 AM
```

This command first assigns System identity to the account, and give the system assigned identity access to the keyvault;
then updates the Storage account to access
Keyvault with system assigned identity.

Example 17: Update both Keyvault and the user assigned identity to access keyvault

```
# Update to another user assigned identity
```

```
$account = Set-AzStorageAccount -ResourceGroupName $resourceGroupName -Name $storageAccountName
-IdentityType SystemAssignedUserAssigned -UserAssignedIdentityId
$useridentity2 -KeyVaultUserAssignedIdentityId $useridentity2
```

Update to encrypt with another keyvault

```
$account = Set-AzStorageAccount -ResourceGroupName $resourceGroupName -Name $storageAccountName
-KeyVaultUri $keyvaultUri2 -KeyName $keyname2 -KeyVersion $keyversion2
```

This command first update the user assigned identity to access keyvault, then update the keyvault for encryption. To update both both Keyvault and the user assigned identity, we need update with the above 2 steps.

Example 18: Update a Storage account with AllowCrossTenantReplication

```
$account = Set-AzStorageAccount -ResourceGroupName "myresourcegroup" -Name "mystorageaccount"
-AllowCrossTenantReplication $false -EnableHttpsTrafficOnly $true
```

```
$account.AllowCrossTenantReplication
```

```
False
```

This command updates a Storage account by set AllowCrossTenantReplication to false, then show the updated account related properties.

Example 18: Update a Storage account by enable PublicNetworkAccess

```
$account = Set-AzStorageAccount -ResourceGroupName "myresourcegroup" -Name "mystorageaccount"
-PublicNetworkAccess Enabled
```

```
$account.PublicNetworkAccess
```

Enabled

This command updates a Storage account by set PublicNetworkAccess as enabled.

----- Example 19: Update account level immutability policy -----

```
$account = Set-AzStorageAccount -ResourceGroupName "MyResourceGroup" -Name "mystorageaccount"
-ImmutabilityPeriod 2 -ImmutabilityPolicyState Unlocked
```

```
$account.ImmutableStorageWithVersioning.Enabled
```

```
True
```

```
$account.ImmutableStorageWithVersioning.ImmutabilityPolicy
```

```
ImmutabilityPeriodSinceCreationInDays State
```

```
-----
```

```
2 Unlocked
```

The command updates account-level immutability policy properties on an existing storage account, and show the result.

The storage account must be created with enable

account level immutability with versioning. The account-level immutability policy will be inherited and applied to objects that do not possess an explicit

immutability policy at the object level.

Example 20: Update a Storage account by enable Sftp and localuser

```
$account = Set-AzStorageAccount -ResourceGroupName "myresourcegroup" -AccountName "mystorageaccount"
-EnableSftp $true -EnableLocalUser $true
```

```
$account.EnableSftp
```

```
True
```

```
$account.EnableLocalUser
```

```
True
```

This command updates a Storage account by enable Sftp and localuser. To run the command succssfully, the Storage account should already enable Hierarchical Namespace.

Example 21: Update a Storage account with Keyvault from another tenant (access Keyvault with FederatedClientId)

```
# create Storage account with Keyvault encryption (access Keyvault with FederatedClientId), then show properties
```

```
$account = Set-AzStorageAccount -ResourceGroupName $resourceGroupName -Name $storageAccountName `
    -KeyVaultUri $keyVault.VaultUri -KeyName $keyname -KeyVaultUserAssignedIdentityId $useridentityId
-KeyVaultFederatedClientId $federatedClientId
```

```
$account.Encryption.EncryptionIdentity
```

```
EncryptionUserAssignedIdentity
```

```
EncryptionFederatedIdentityClientId
```

```
-----
```

```
-----
```

```
/subscriptions/{subscription-id}/resourceGroups/myresourcegroup/providers/Microsoft.ManagedIdentity/userAssignedIdentities/myuserid
```

```
*****_****_****_****_*****
```

```
$account.Encryption.KeyVaultProperties
```

```
KeyName : wrappingKey
```

```
KeyVersion :
```

```
KeyVaultUri : https://mykeyvault.vault.azure.net:443
```

```
CurrentVersionedKeyIdentifier :
```

```
https://mykeyvault.vault.azure.net/keys/wrappingKey/8e74036e0d534e58b3bd84b319e31d8f
```

LastKeyRotationTimestamp : 3/3/2022 2:07:34 AM

This command updates a storage account with Keyvault from another tenant (access Keyvault with FederatedClientId).

RELATED LINKS

Online Version: <https://learn.microsoft.com/powershell/module/az.storage/set-azstorageaccount>

Get-AzStorageAccount

New-AzStorageAccount

Remove-AzStorageAccount