



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

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

NAME

Set-AzStorageLocalUser

SYNOPSIS

Creates or updates a specified local user in a storage account.

SYNTAX

```
Set-AzStorageLocalUser [-ResourceGroupName] <System.String> [-StorageAccountName] <System.String>
[-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-HasSharedKey
<System.Boolean>] [-HasSshKey <System.Boolean>]
[-HasSshPassword <System.Boolean>] [-HomeDirectory <System.String>] [-PermissionScope
<Microsoft.Azure.Commands.Management.Storage.Models.PSPermissionScope[]>]
[-SshAuthorizedKey <Microsoft.Azure.Commands.Management.Storage.Models.PSSshPublicKey[]>] -UserName
<System.String> [-Confirm] [-WhatIf] [<CommonParameters>]
```

Set-AzStorageLocalUser	[-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>	[-HasSharedKey
<System.Boolean>]	

```

    [-HasSshKey <System.Boolean>] [-HasSshPassword <System.Boolean>] [-HomeDirectory <System.String>]
[-PermissionScope
    <Microsoft.Azure.Commands.Management.Storage.Models.PSPermissionScope[]>] [-SshAuthorizedKey
<Microsoft.Azure.Commands.Management.Storage.Models.PSSshPublicKey[]>]
    -StorageAccount <Microsoft.Azure.Commands.Management.Storage.Models.PSStorageAccount> -UserName
<System.String> [-Confirm] [-WhatIf] [<CommonParameters>]

```

DESCRIPTION

The Set-AzStorageLocalUser cmdlet creates or updates a specified local user in a storage account. To run this cmdlet, the storage account must have already set

EnableLocalUser as true.

PARAMETERS

-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

-HasSharedKey <System.Boolean>

Whether shared key exists. Set it to false to remove existing shared key.

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

-HasSshKey <System.Boolean>

Whether SSH key exists. Set it to false to remove existing SSH key.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-HasSshPassword <System.Boolean>

Whether SSH password exists. Set it to false to remove existing SSH password.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-HomeDirectory <System.String>

Local user home directory

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-PermissionScope <Microsoft.Azure.Commands.Management.Storage.Models.PSPermissionScope[]>

The permission scopes of the local user. Get the object with New-AzStorageLocalUserPermissionScope cmdlet.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ResourceGroupName <System.String>

Resource Group Name.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SshAuthorizedKey <Microsoft.Azure.Commands.Management.Storage.Models.PSSshPublicKey[]>

Local user ssh authorized keys for SFTP. Get the object with New-AzStorageLocalUserSshPublicKey cmdlet.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-StorageAccount <Microsoft.Azure.Commands.Management.Storage.Models.PSStorageAccount>

Storage account object

Required? true

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-StorageAccountName <System.String>

Storage Account Name.

Required?	true
Position?	1
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-UserName <System.String>

The name of local user. The username must contain lowercase letters and numbers only. It must be unique only within the storage account.

Required?	true
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

Microsoft.Azure.Commands.Management.Storage.Models.PSStorageAccount

OUTPUTS

Microsoft.Azure.Commands.Management.Storage.Models.PSLocalUser

NOTES

----- Example 1: Create or update a local user -----

```
$sshkey1 = New-AzStorageLocalUserSshPublicKey -Key "ssh-rsa base64encodedkey=" -Description "sshpublickey  
name1"
```

```
$permissionScope1 = New-AzStorageLocalUserPermissionScope -Permission rw -Service blob -ResourceName  
container1
```

```
$localuser = Set-AzStorageLocalUser -ResourceGroupName "myresourcegroup" -AccountName "mystorageaccount"  
-UserName testuser1 -HomeDirectory "/" -SshAuthorizedKey
```

```
$sshkey1 -PermissionScope $permissionScope1 -HasSharedKey $true -HasSshKey $true -HasSshPassword $true
```

\$localuser

ResourceGroupName: myresourcegroup, StorageAccountName: mystorageaccount

Name	Sid	HomeDirectory	HasSharedKey	HasSshKey	HasSshPassword	PermissionScopes
testuser1	S-1-2-0-0000000000-000000000-0000000000-0000	/	True	True	True	[container1]

\$localuser.SshAuthorizedKeys

Description	Key
sshpublickey name1	ssh-rsa base64encodedkey=

\$localuser.PermissionScopes

Permissions	Service	ResourceName
rw	blob	container1

The first command creates a local SSH public key object. Note that the key follows the format of ``<algorithm> <data>`` where data is the base64 encoded contents of the public key. The second command creates a local permission scope object that defines the container level access for the local user. The third command creates or updates the local user, using the local objects from the first 2 commands. The final command shows the local user properties.

Example 2: Create or update a local user by input permission scope and ssh key with json

```
testuser1 -HomeDirectory "/" -HasSharedKey $true -HasSshKey
$true -HasSshPassword $true `
    -SshAuthorizedKey (@{
        Description="sshpullickey name1";
        Key="ssh-rsa base64encodedkey=";
    },
    @{
        Description="sshpullickey name2";
        Key="ssh-rsa otherbase64encodedkey=";
    }) `
    -PermissionScope (@{
        Permissions="rw";
        Service="blob";
        ResourceName="container1";
    },
    @{
        Permissions="rwd";
        Service="share";
        ResourceName="share1";
    })
```

ResourceGroupName: weitry, StorageAccountName: weisftp3

Name	Sid	HomeDirectory	HasSharedKey	HasSshKey	HasSshPassword	PermissionScopes
testuser1	S-1-2-0-00000000000-0000000000-0000000000-0000	/	True	True	True	[container1,...]

This command creates or updates a local user by input permission scope and ssh key with json.

