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

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

NAME

New-AzRoleDefinition

SYNOPSIS

Creates a custom role in Azure RBAC. Provide either a JSON role definition file or a PSRoleDefinition object as input. First, use the Get-AzRoleDefinition command to

generate a baseline role definition object. Then, modify its properties as required. Finally, use this command to create a custom role using role definition.

SYNTAX

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

[-SkipClientSideScopeValidation] [<CommonParameters>]

New-AzRoleDefinition [-Role] <Microsoft.Azure.Commands.Resources.Models.Authorization.PSRoleDefinition> [-DefaultProfile

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

DESCRIPTION

The New-AzRoleDefinition cmdlet creates a custom role in Azure Role-Based Access Control. Provide a role definition as an input to the command as a JSON file or a

PSRoleDefinition object. The input role definition MUST contain the following properties: 1) DisplayName: the name of the custom role 2) Description: a short

description of the role that summarizes the access that the role grants. 3) Actions: the set of operations to which the custom role grants access. Use

Get-AzProviderOperation to get the operation for Azure resource providers that can be secured using Azure RBAC. Following are some valid operation strings: -

"*/read" grants access to read operations of all Azure resource providers. - "Microsoft.Network/*/read" grants access to read operations for all resource types in

the Microsoft.Network resource provider of Azure. - "Microsoft.Compute/virtualMachines/*" grants access to all operations of virtual machines and its child resource

types. 4) AssignableScopes: the set of scopes (Azure subscriptions or resource groups) in which the custom role will be available for assignment. Using

AssignableScopes you can make the custom role available for assignment in only the subscriptions or resource groups that need it, and not clutter the user experience

for the rest of the subscriptions or resource groups. Following are some valid assignable scopes: - "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e",

"/subscriptions/e91d47c4-76f3-4271-a796-21b4ecfe3624": makes the role available for assignment in two subscriptions. -

"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e": makes the role available for assignment in a single subscription. -

"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network": makes the role available for assignment only in the Network resource group. The input

role definition MAY contain the following properties: 1) NotActions: the set of operations that must be excluded from the Actions to determine the effective actions

for the custom role. If there is a specific operation that you do not wish to grant access to in a custom role, it is convenient to use NotActions to exclude it,

rather than specifying all operations other than that specific operation in Actions. 2) DataActions: the set of data operations to which the custom role grants

access. 3) NotDataActions: the set of operations that must be excluded from the DataActions to determine the effective

data actions for the custom role. If there is a

specific data operation that you do not wish to grant access to in a custom role, it is convenient to use NotDataActions to exclude it, rather than specifying all

operations other than that specific operation in Actions. NOTE: If a user is assigned a role that specifies an operation in NotActions and also assigned another role

grants access to the same operation - the user will be able to perform that operation. NotActions is not a deny rule - it is simply a convenient way to create a set

of allowed operations when specific operations need to be excluded. Following is a sample json role definition that can be provided as input { "Name":

"Updated Role", "Description": "Can monitor all resources and start and restart virtual machines", "Actions": ["*/read",

"Microsoft.ClassicCompute/virtualmachines/restart/action",

"Microsoft.ClassicCompute/virtualmachines/start/action"], "NotActions":

["*/write"], "DataActions": [

"Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read"],

"NotDataActions": ["Microsoft.Storage/storageAccounts/blobServices/containers/blobs/write"], "AssignableScopes":

["/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"] }

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

-InputFile <System.String>

File name containing a single json role definition.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Role <Microsoft.Azure.Commands.Resources.Models.Authorization.PSRoleDefinition>

Role definition object.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SkipClientSideScopeValidation <System.Management.Automation.SwitchParameter>

If specified, skip client side scope validation.

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

None

OUTPUTS

Microsoft.Azure.Commands.Resources.Models.Authorization.PSRoleDefinition

NOTES

Keywords: azure, azurerm, arm, resource, management, manager, resource, group, template, deployment

-------- Example 1: Create using PSRoleDefinitionObject --------

```
$role = New-Object -TypeName Microsoft.Azure.Commands.Resources.Models.Authorization.PSRoleDefinition
$role.Name = 'Virtual Machine Operator'
$role.Description = 'Can monitor, start, and restart virtual machines.'
$role.IsCustom = $true
$role.AssignableScopes = @("/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx")
$role.Actions = @(
    "Microsoft.Compute/*/read"
    "Microsoft.Compute/virtualMachines/start/action"
    "Microsoft.Compute/virtualMachines/restart/action"
    "Microsoft.Compute/virtualMachines/downloadRemoteDesktopConnectionFile/action"
    "Microsoft.Network/*/read"
    "Microsoft.Storage/*/read"
    "Microsoft.Authorization/*/read"
    "Microsoft.Resources/subscriptions/resourceGroups/read"
    "Microsoft.Resources/subscriptions/resourceGroups/resources/read"
    "Microsoft.Insights/alertRules/*"
    "Microsoft.Support/*"
)
```

```
New-AzRoleDefinition -Role $role
```

---------- Example 2: Create using JSON file ----------

```
New-AzRoleDefinition -InputFile C:\Temp\roleDefinition.json
```

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.resources/new-azroledefinition

Get-AzProviderOperation

Get-AzRoleDefinition

Set-AzRoleDefinition

Remove-AzRoleDefinition