



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

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

NAME

New-AzPolicyAttestation

SYNOPSIS

Creates a new policy attestation for a policy assignment.

SYNTAX

```
New-AzPolicyAttestation [-AssessmentDate <System.Nullable`1[System.DateTime]>] [-Comment <System.String>]
[-ComplianceState <System.String>] [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Evidence
<Microsoft.Azure.Commands.PolicyInsights.Models.Attestations.PSAttestationEvidence[]>] [-ExpiresOn
<System.Nullable`1[System.DateTime]>] [-Metadata <System.String>]
-Name <System.String> [-Owner <System.String>] -PolicyAssignmentId <System.String> [-PolicyDefinitionReferenceId
<System.String>] [-ResourceGroupName <System.String>]
[-Scope <System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]
```

```
New-AzPolicyAttestation [-AssessmentDate <System.Nullable`1[System.DateTime]>] [-Comment <System.String>]
[-ComplianceState <System.String>] [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Evidence
```

```

        <Microsoft.Azure.Commands.PolicyInsights.Models.Attestations.PSAttestationEvidence[]> [-ExpiresOn
<System.Nullable`1[System.DateTime]>] [-Metadata <System.String>]
        [-Owner <System.String>] -PolicyAssignmentId <System.String> [-PolicyDefinitionReferenceId <System.String>]
-ResourceId <System.String> [-Confirm] [-WhatIf]
        [<CommonParameters>]

```

DESCRIPTION

The New-AzPolicyAttestation cmdlet creates a policy attestation for a particular policy assignment. Attestations are used by Azure Policy to set compliance states of resources or scopes targeted by manual policies. They also allow users to provide additional metadata or link to evidence which accompanies the attested compliance state.

PARAMETERS

-AssessmentDate <System.Nullable`1[System.DateTime]>

The time the evidence of an attestation was assessed.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Comment <System.String>

Comments describing why this attestation was created.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ComplianceState <System.String>

The Compliance State of the resource. E.g. 'Compliant', 'NonCompliant', 'Unknown'

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

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

-Evidence <Microsoft.Azure.Commands.PolicyInsights.Models.Attestations.PSAttestationEvidence[]>

The evidence supporting the compliance state set in this attestation.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ExpiresOn <System.Nullable`1[System.DateTime]>

The time the compliance state set in the attestation should expire.

Required? false

Position? named

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

-Metadata <System.String>

Additional metadata for the attestation.

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

-Name <System.String>

Resource name.

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

-Owner <System.String>

The person responsible for setting the state of the resource. This value is typically a Microsoft Entra object ID.

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

-PolicyAssignmentId <System.String>

'/subscriptions/{subscriptionId}/providers/Microsoft.Authorization/policyAssignments/{assignmentName}'.

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

-PolicyDefinitionReferenceId <System.String>

The policy definition reference ID of the individual definition. Required when the policy assignment assigns a policy set definition.

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

-ResourceGroupName <System.String>

Resource group name.

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

-ResourceId <System.String>

Resource ID.

Required?	true
Position?	named
Default value	None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Scope <System.String>

Scope of the resource. E.g. '/subscriptions/{subscriptionId}/resourceGroups/{rgName}'.

Required? false

Position? named

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

System.Nullable`1[[System.DateTime, System.Private.CoreLib, Version=7.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e]]

Microsoft.Azure.Commands.PolicyInsights.Models.Attestations.PSAttestationEvidence[]

System.Object

OUTPUTS

Microsoft.Azure.Commands.PolicyInsights.Models.Attestations.PSAttestation

NOTES

---- Example 1: Create an attestation at subscription scope ----

Set-AzContext -Subscription "d1acb22b-c876-44f7-b08e-3fcf9f6767f4"

\$policyAssignmentId =

"/subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.authorization/policyassignments/psattestationsu

bassignment"

\$attestationName = "attestation-subscription"

New-AzPolicyAttestation -PolicyAssignmentId \$policyAssignmentId -Name \$attestationName -ComplianceState
"Compliant"

Id : /subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.policyinsights
/attestations/attestation-subscription

Name : attestation-subscription

Type : Microsoft.PolicyInsights/attestations

PolicyAssignmentId : /subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.authorization/
policyassignments/psattestationsubassignment

PolicyDefinitionReferenceId :

ComplianceState : Compliant

ExpiresOn :

Owner :

Comment :

Evidence :

ProvisioningState : Succeeded

LastComplianceStateChangeAt : 1/27/2023 2:26:24 AM

AssessmentDate :

Metadata :

SystemData :

This command creates a new policy attestation at subscription 'd1acb22b-c876-44f7-b08e-3fcf9f6767f4' for the given policy assignment.

> Note: >This command creates an attestation for the subscription and not the resources underneath it. For ease of management, manual policies should be designed to

target the scope which defines the boundary of resources whose compliance state needs to be attested. In this case, the manual policy should be targeting

`Microsoft.Resources/subscriptions`. For more information, go to <https://learn.microsoft.com/en-us/azure/governance/policy/concepts/attestation-structure> to

understand the best practices for creating attestations.

----- Example 2: Create an attestation at resource group -----

```

                                $policyAssignmentId
"/subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.authorization/policyassignments/psattestationrga
ssignment"

$attestationName = "attestation-RG"

$rgName = "ps-attestation-test-rg"

New-AzPolicyAttestation -ResourceGroupName $RGName -PolicyAssignmentId $policyAssignmentId -Name
$attestationName -ComplianceState "Compliant"

Id                : /subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/resourcegroups/ps-attestation-test
                  -rg/providers/microsoft.policyinsights/attestations/attestation-rg
Name              : attestation-RG
Type              : Microsoft.PolicyInsights/attestations
PolicyAssignmentId : /subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.authorization/
                  policyassignments/psattestationrgassignment
PolicyDefinitionReferenceId :
ComplianceState   : Compliant
ExpiresOn         :
Owner             :
Comment           :
Evidence          :
ProvisioningState  : Succeeded
LastComplianceStateChangeAt : 1/27/2023 2:35:28 AM
AssessmentDate    :
Metadata          :
SystemData        :
```

This command creates a new policy attestation at the resource group 'ps-attestation-test-rg' for the given policy assignment.

> Note: >This command creates an attestation for the resource group and not the resources underneath it. For ease of management, manual policies should be designed to

target the scope which defines the boundary of resources whose compliance state needs to be attested. In this case, the manual policy should be targeting

`Microsoft.Resources/subscriptions/resourceGroups`. For more information, go to

<https://learn.microsoft.com/en-us/azure/governance/policy/concepts/attestation-structure> to understand the best practices for creating attestations.

----- Example 3: Create an attestation at resource -----

```
                                $policyAssignmentId                =  
"/subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.authorization/policyassignments/psattestationresourceassignment"
```

```
$attestationName = "attestation-resource"  
  
                                $scope                            =  
"/subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/resourceGroups/ps-attestation-test-rg/providers/Microsoft.Network/networkSecurityGroups/pstests0"
```

New-AzPolicyAttestation `

```
-PolicyAssignmentId $policyAssignmentId `  
-Name $attestationName `  
-Scope $scope `  
-ComplianceState "NonCompliant"
```

```
Id                : /subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/resourcegroups/ps-attestation-test  
                  -rg/providers/microsoft.network/networksecuritygroups/pstests0/providers/microsoft.policyinsights/attestations/attestation-resource
```

```
Name              : attestation-resource
```

```
Type              : Microsoft.PolicyInsights/attestations
```

```
PolicyAssignmentId : /subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.authorization/  
                  policyassignments/psattestationresourceassignment
```

```
PolicyDefinitionReferenceId :
```

```
ComplianceState    : NonCompliant
```

ExpiresOn :
Owner :
Comment :
Evidence :
ProvisioningState : Succeeded
LastComplianceStateChangeAt : 1/27/2023 2:38:17 AM
AssessmentDate :
Metadata :
SystemData :

This command creates an attestation for the resource 'pstests0' for the given policy assignment.

Example 4: Create an attestation with all properties at resource group

```
$attestationName = "attestationRGAllProps"
```

```
$policyInitiativeAssignmentId =
```

```
"/subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.authorization/policyassignments/psattestationinitiativeassignment"
```

```
$policyDefinitionReferenceId = "PSTestAttestationRG_1"
```

```
$RGName = "ps-attestation-test-rg"
```

```
$description = "This is a test description"
```

```
$sourceURI = "https://contoso.org/test.pdf"
```

```
$evidence = @{
```

```
    "Description"=$description
```

```
    "SourceUri"=$sourceURI
```

```
}
```

```
$policyEvidence = @($evidence)
```

```
$owner = "Test Owner"
```

```
$expiresOn = [datetime]::UtcNow.AddYears(1)
```

```
$metadata = '{"TestKey":"TestValue"}'
```

New-AzPolicyAttestation `

-Name \$attestationName `
-ResourceGroupName \$RGName `
-PolicyAssignmentId \$policyInitiativeAssignmentId `
-PolicyDefinitionReferenceId \$policyDefinitionReferenceId `
-ComplianceState \$Compliant `
-Comment \$comment `
-Evidence \$policyEvidence `
-ExpiresOn \$expiresOn `
-AssessmentDate \$expiresOn.AddDays(-2) `
-Owner \$owner `
-Metadata \$metadata

Id : /subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/resourcegroups/ps-attestation-test
-rg/providers/microsoft.policyinsights/attestations/attestationrgallprops

Name : attestationRGAllProps

Type : Microsoft.PolicyInsights/attestations

PolicyAssignmentId : /subscriptions/d1acb22b-c876-44f7-b08e-3fcf9f6767f4/providers/microsoft.authorization/
policyassignments/psattestationinitiativergassignment

PolicyDefinitionReferenceId : pstestattestationrg_1

ComplianceState :

ExpiresOn : 1/27/2024 2:51:54 AM

Owner : Test Owner

Comment :

Evidence : {Microsoft.Azure.Commands.PolicyInsights.Models.Attestations.PSAttestationEvidence}

ProvisioningState : Succeeded

LastComplianceStateChangeAt : 1/27/2023 2:51:57 AM

AssessmentDate : 1/25/2024 2:51:54 AM

Metadata : {
"TestKey": "TestValue"
}

SystemData :

RELATED LINKS

Online Version: <https://learn.microsoft.com/powershell/module/az.policyinsights/new-azpolicyattestation>

Get-AzPolicyAttestation

Set-AzPolicyAttestation

Remove-AzPolicyAttestation