Get compliance data of Azure resources

07/15/2020 • 15 minutes to read •

In this article

One of the largest benefits of Azure Policy is the insight and controls it provides over resources in a subscription or management group of subscriptions. This control can be exercised in many different ways, such as preventing resources being created in the wrong location, enforcing common and consistent tag usage, or auditing existing resources for appropriate configurations and settings. In all cases, data is generated by Azure Policy to enable you to understand the compliance state of your environment.

There are several ways to access the compliance information generated by your policy and initiative assignments:

- Using the Azure portal
- Through command line scripting

Before looking at the methods to report on compliance, let's look at when compliance information is updated and the frequency and events that trigger an evaluation cycle.

> ⚠ **Warning**
>
> If compliance state is being reported as **Not registered**, verify that the **Microsoft.PolicyInsights** Resource Provider is registered and that the user has the appropriate role-based access control (RBAC) permissions as described in RBAC in Azure Policy.

Evaluation triggers

The results of a completed evaluation cycle are available in the `Microsoft.PolicyInsights` Resource Provider through `PolicyStates` and `PolicyEvents` operations. For more information about the operations of the Azure Policy Insights REST API, see Azure Policy Insights.

Evaluations of assigned policies and initiatives happen as the result of various events:

- A policy or initiative is newly assigned to a scope. It takes around 30 minutes for the assignment to be applied to the defined scope. Once it's applied, the evaluation cycle begins for resources within that scope against the newly assigned policy or initiative and depending on the effects used by the policy or initiative, resources are marked as compliant or non-compliant. A large policy or initiative evaluated against a large scope of resources can take time. As such, there's no pre-defined expectation of when the evaluation cycle completes. Once it completes, updated compliance results are available in the portal and SDKs.
- A policy or initiative already assigned to a scope is updated. The evaluation cycle and timing for this scenario is the same as for a new assignment to a scope.
- A resource is deployed to a scope with an assignment via Azure Resource Manager, REST, Azure CLI, or Azure PowerShell. In this scenario, the effect event (append, audit, deny, deploy) and compliant status information for the individual resource becomes available in the portal and SDKs around 15 minutes later. This event doesn't cause an evaluation of other resources.
- Standard compliance evaluation cycle. Once every 24 hours, assignments are automatically reevaluated. A large policy or initiative of many resources can take time, so there's no pre-defined expectation of when the evaluation cycle completes. Once it completes, updated compliance results are available in the portal and SDKs.
- The Guest Configuration resource provider is updated with compliance details by a managed resource.
- On-demand scan

On-demand evaluation scan

An evaluation scan for a subscription or a resource group can be started with Azure CLI, Azure PowerShell, or a call to the REST API. This scan is an asynchronous process.

On-demand evaluation scan - Azure CLI

The compliance scan is started with the az policy state trigger-scan command.

By default, `az policy state trigger-scan` starts an evaluation for all resources in the current subscription. To start an evaluation on a specific resource group, use the

resource-group parameter. The following example starts a compliance scan in the current subscription for the *MyRG* resource group:

Azure CLI Copy Try It

```
az policy state trigger-scan --resource-group "MyRG"
```

You can chose not to wait for the asynchronous process to complete before continuing with the **no-wait** parameter.

On-demand evaluation scan - Azure PowerShell

The compliance scan is started with the Start-AzPolicyComplianceScan cmdlet.

By default, `Start-AzPolicyComplianceScan` starts an evaluation for all resources in the current subscription. To start an evaluation on a specific resource group, use the **ResourceGroupName** parameter. The following example starts a compliance scan in the current subscription for the *MyRG* resource group:

Azure PowerShell Copy Try It

```
Start-AzPolicyComplianceScan -ResourceGroupName 'MyRG'
```

You can have PowerShell wait for the asynchronous call to complete before providing the results output or have it run in the background as a job. To use a PowerShell job to run the compliance scan in the background, use the **AsJob** parameter and set the value to an object, such as `$job` in this example:

Azure PowerShell Copy Try It

```
$job = Start-AzPolicyComplianceScan -AsJob
```

You can check on the status of the job by checking on the `$job` object. The job is of the type `Microsoft.Azure.Commands.Common.AzureLongRunningJob`. Use `Get-Member` on the `$job` object to see available properties and methods.

While the compliance scan is running, checking the `$job` object outputs results such as these:

Azure PowerShell Copy Try It

```
$job

Id     Name             PSJobTypeName   State         HasMoreData
```

```
Location             Command
--     ----            -------------   -----         -----------     -------
-             -------
2      Long Running O… AzureLongRunni… Running       True
localhost            Start-AzPolicyCompliance…
```

When the compliance scan completes, the **State** property changes to *Completed*.

On-demand evaluation scan - REST

As an asynchronous process, the REST endpoint to start the scan doesn't wait until the scan is complete to respond. Instead, it provides a URI to query the status of the requested evaluation.

In each REST API URI, there are variables that are used that you need to replace with your own values:

- `{YourRG}` - Replace with the name of your resource group
- `{subscriptionId}` - Replace with your subscription ID

The scan supports evaluation of resources in a subscription or in a resource group. Start a scan by scope with a REST API **POST** command using the following URI structures:

- Subscription

  HTTP Copy

  ```
  POST
  https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/policyStates/latest/triggerEvaluation?api-version=2019-10-01
  ```

- Resource group

  HTTP Copy

  ```
  POST
  https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{YourRG}/providers/Microsoft.PolicyInsights/policyStates/latest/triggerEvaluation?api-version=2019-10-01
  ```

The call returns a **202 Accepted** status. Included in the response header is a **Location** property with the following format:

HTTP Copy

```
https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/asyncOperationResults/{ResourceContainerGUID}?api-version=2019-10-01
```

`{ResourceContainerGUID}` is statically generated for the scope requested. If a scope is already running an on-demand scan, a new scan isn't started. Instead, the new request is provided the same `{ResourceContainerGUID}` **location** URI for status. A REST API **GET** command to the **Location** URI returns a **202 Accepted** while the evaluation is ongoing. When the evaluation scan has completed, it returns a **200 OK** status. The body of a completed scan is a JSON response with the status:

```json
{
    "status": "Succeeded"
}
```

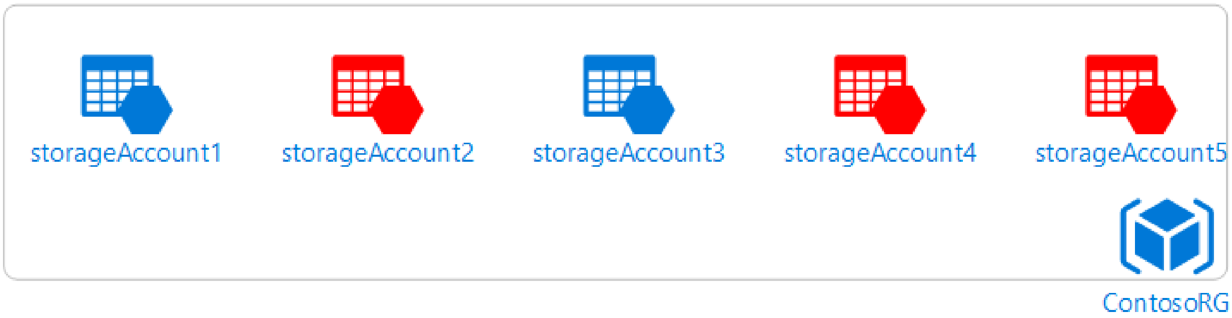
How compliance works

In an assignment, a resource is **Non-compliant** if it doesn't follow policy or initiative rules. The following table shows how different policy effects work with the condition evaluation for the resulting compliance state:

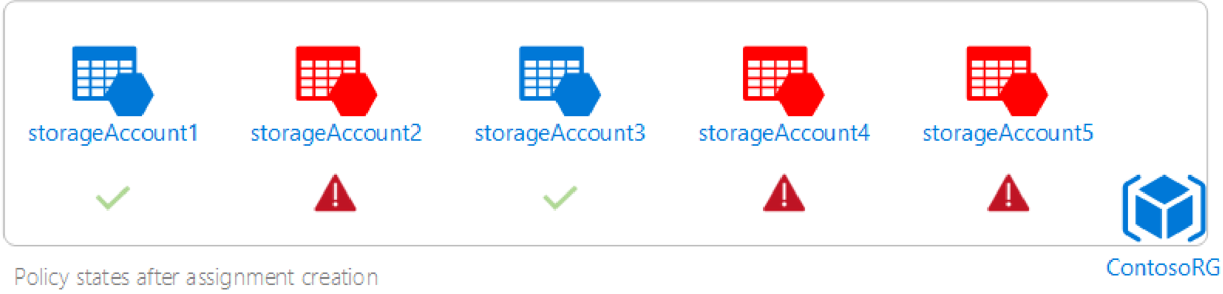
Resource state	Effect	Policy evaluation	Compliance state
Exists	Deny, Audit, Append*, DeployIfNotExist*, AuditIfNotExist*	True	Non-compliant
Exists	Deny, Audit, Append*, DeployIfNotExist*, AuditIfNotExist*	False	Compliant
New	Audit, AuditIfNotExist*	True	Non-compliant
New	Audit, AuditIfNotExist*	False	Compliant

* The Append, DeployIfNotExist, and AuditIfNotExist effects require the IF statement to be TRUE. The effects also require the existence condition to be FALSE to be non-compliant. When TRUE, the IF condition triggers evaluation of the existence condition for the related resources.

For example, assume that you have a resource group – ContsoRG, with some storage accounts (highlighted in red) that are exposed to public networks.



In this example, you need to be wary of security risks. Now that you've created a policy assignment, it's evaluated for all storage accounts in the ContosoRG resource group. It audits the three non-compliant storage accounts, consequently changing their states to **Non-compliant.**



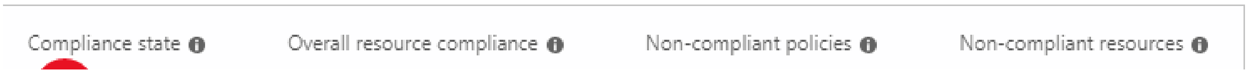
Policy states after assignment creation

Besides **Compliant** and **Non-compliant**, policies and resources have three other states:

- **Conflicting**: Two or more policies exist with conflicting rules. For example, two policies appending the same tag with different values.
- **Not started**: The evaluation cycle hasn't started for the policy or resource.
- **Not registered**: The Azure Policy Resource Provider hasn't been registered or the account logged in doesn't have permission to read compliance data.

Azure Policy uses the **type** and **name** fields in the definition to determine if a resource is a match. When the resource matches, it's considered applicable and has a status of either **Compliant** or **Non-compliant**. If either **type** or **name** is the only property in the definition, then all resources are considered applicable and are evaluated.

The compliance percentage is determined by dividing **Compliant** resources by *total resources*. *Total resources* is defined as the sum of the **Compliant**, **Non-compliant**, and **Conflicting** resources. The overall compliance numbers are the sum of distinct resources that are **Compliant** divided by the sum of all distinct resources. In the image below, there are 20 distinct resources that are applicable and only one is **Non-compliant**. The overall resource compliance is 95% (19 out of 20).



Non-compliant

95%
19 out of 20

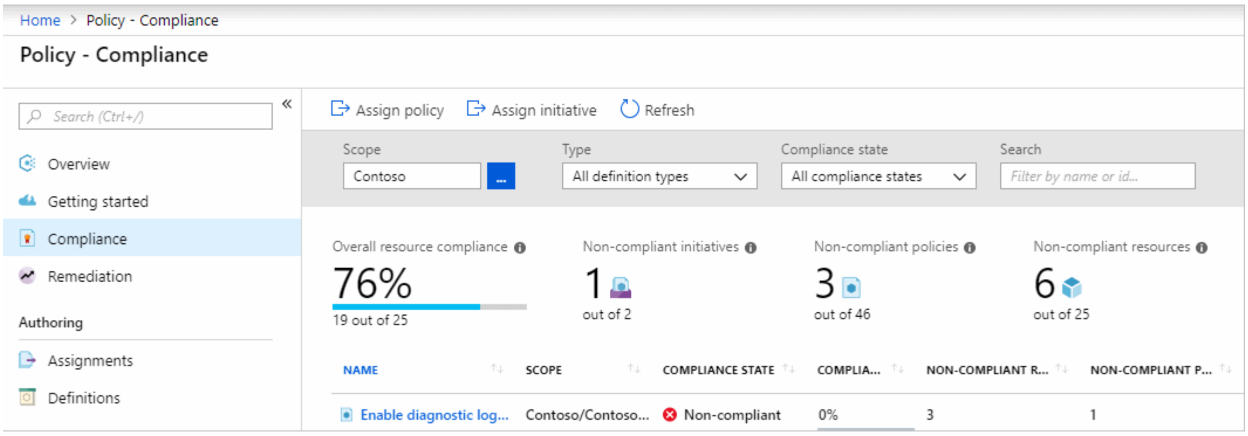
1
out of 38

1
out of 20

> ⓘ **Note**
>
> Regulatory Compliance in Azure Policy is a Preview feature. Compliance properties from SDK and pages in portal are different for enabled initiatives. For more information, see **Regulatory Compliance**

Portal

The Azure portal showcases a graphical experience of visualizing and understanding the state of compliance in your environment. On the **Policy** page, the **Overview** option provides details for available scopes on the compliance of both policies and initiatives. Along with the compliance state and count per assignment, it contains a chart showing compliance over the last seven days. The **Compliance** page contains much of this same information (except the chart), but provide additional filtering and sorting options.



Home > Policy - Compliance

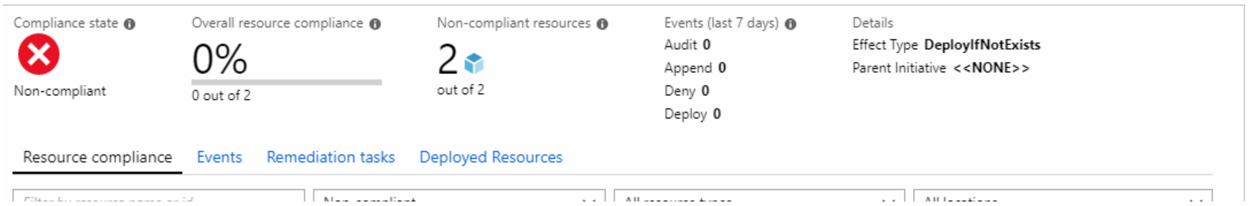
Policy - Compliance

Scope: Contoso
Type: All definition types
Compliance state: All compliance states
Search: Filter by name or id...

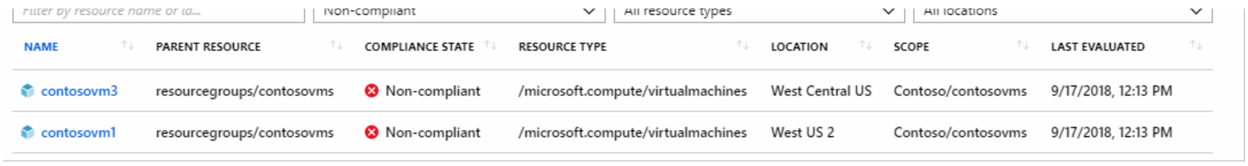
Overall resource compliance: 76% — 19 out of 25
Non-compliant initiatives: 1 — out of 2
Non-compliant policies: 3 — out of 46
Non-compliant resources: 6 — out of 25

NAME	SCOPE	COMPLIANCE STATE	COMPLIA...	NON-COMPLIANT R...	NON-COMPLIANT P...
Enable diagnostic log...	Contoso/Contoso...	☒ Non-compliant	0%	3	1

Since a policy or initiative can be assigned to different scopes, the table includes the scope for each assignment and the type of definition that was assigned. The number of non-compliant resources and non-compliant policies for each assignment are also provided. Clicking on a policy or initiative in the table provides a deeper look at the compliance for that particular assignment.



Compliance state: Non-compliant
Overall resource compliance: 0% — 0 out of 2
Non-compliant resources: 2 — out of 2
Events (last 7 days): Audit 0, Append 0, Deny 0, Deploy 0
Details: Effect Type **DeployIfNotExists**, Parent Initiative **<<NONE>>**

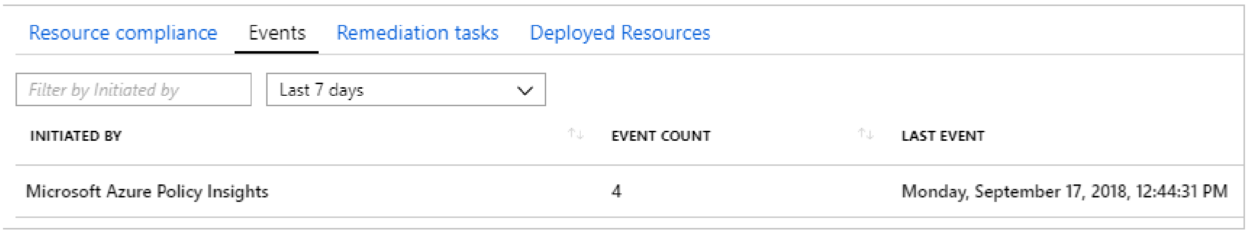


Filter by resource name or id... | Non-compliant | All resource types | All locations

NAME	PARENT RESOURCE	COMPLIANCE STATE	RESOURCE TYPE	LOCATION	SCOPE	LAST EVALUATED
contosovm3	resourcegroups/contosovms	☒ Non-compliant	/microsoft.compute/virtualmachines	West Central US	Contoso/contosovms	9/17/2018, 12:13 PM
contosovm1	resourcegroups/contosovms	☒ Non-compliant	/microsoft.compute/virtualmachines	West US 2	Contoso/contosovms	9/17/2018, 12:13 PM

The list of resources on the **Resource compliance** tab shows the evaluation status of existing resources for the current assignment. The tab defaults to **Non-compliant**, but can be filtered. Events (append, audit, deny, deploy) triggered by the request to create a resource are shown under the **Events** tab.

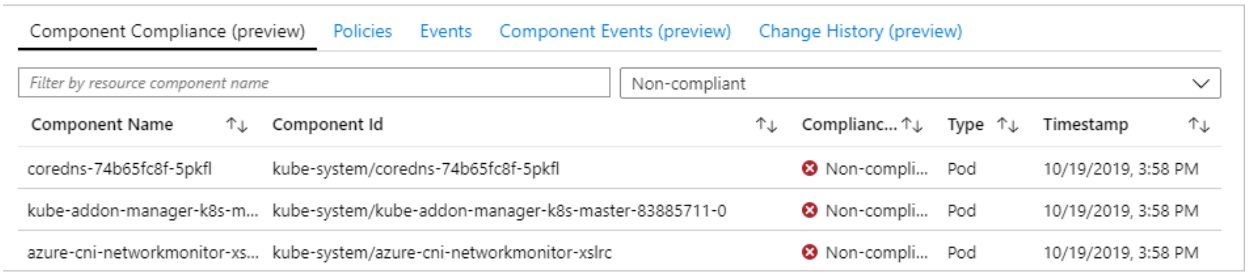
> ⓘ **Note**
>
> For an AKS Engine policy, the resource shown is the resource group.



Filter by Initiated by | Last 7 days

INITIATED BY	EVENT COUNT	LAST EVENT
Microsoft Azure Policy Insights	4	Monday, September 17, 2018, 12:44:31 PM

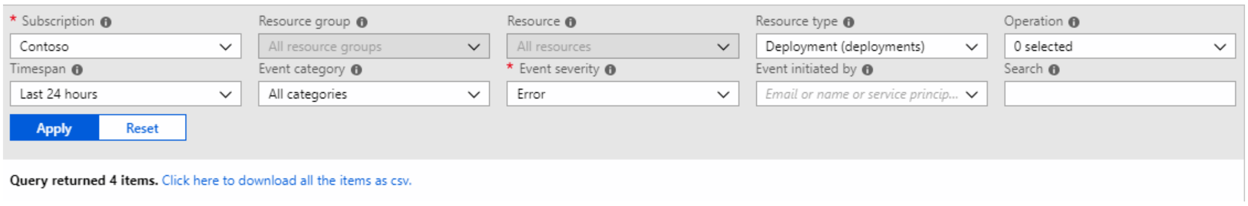
For Resource Provider mode resources, on the **Resource compliance** tab, selecting the resource or right-clicking on the row and selecting **View compliance details** opens the component compliance details. This page also offers tabs to see the policies that are assigned to this resource, events, component events, and change history.



Filter by resource component name | Non-compliant

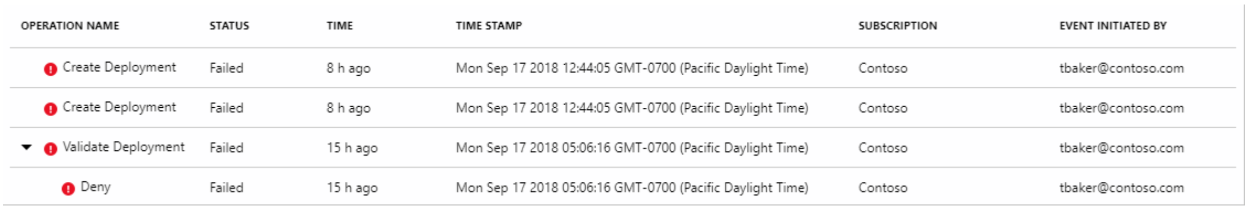
Component Name	Component Id	Complianc...	Type	Timestamp
coredns-74b65fc8f-5pkfl	kube-system/coredns-74b65fc8f-5pkfl	☒ Non-compli...	Pod	10/19/2019, 3:58 PM
kube-addon-manager-k8s-m...	kube-system/kube-addon-manager-k8s-master-83885711-0	☒ Non-compli...	Pod	10/19/2019, 3:58 PM
azure-cni-networkmonitor-xs...	kube-system/azure-cni-networkmonitor-xslrc	☒ Non-compli...	Pod	10/19/2019, 3:58 PM

Back on the resource compliance page, right-click on the row of the event you would like to gather more details on and select **Show activity logs**. The activity log page opens and is pre-filtered to the search showing details for the assignment and the events. The activity log provides additional context and information about those events.



* Subscription	Resource group	Resource	Resource type	Operation
Contoso	All resource groups	All resources	Deployment (deployments)	0 selected
Timespan	Event category	* Event severity	Event initiated by	Search
Last 24 hours	All categories	Error	Email or name or service princip...	

Query returned 4 items. Click here to download all the items as csv.



OPERATION NAME	STATUS	TIME	TIME STAMP	SUBSCRIPTION	EVENT INITIATED BY
Create Deployment	Failed	8 h ago	Mon Sep 17 2018 12:44:05 GMT-0700 (Pacific Daylight Time)	Contoso	tbaker@contoso.com
Create Deployment	Failed	8 h ago	Mon Sep 17 2018 12:44:05 GMT-0700 (Pacific Daylight Time)	Contoso	tbaker@contoso.com
▾ Validate Deployment	Failed	15 h ago	Mon Sep 17 2018 05:06:16 GMT-0700 (Pacific Daylight Time)	Contoso	tbaker@contoso.com
Deny	Failed	15 h ago	Mon Sep 17 2018 05:06:16 GMT-0700 (Pacific Daylight Time)	Contoso	tbaker@contoso.com

Understand non-compliance

When a resource is determined to be **non-compliant**, there are many possible reasons. To determine the reason a resource is **non-compliant** or to find the change responsible, see Determine non-compliance.

Command line

The same information available in the portal can be retrieved with the REST API (including with ARMClient), Azure PowerShell, and Azure CLI. For full details on the REST API, see the Azure Policy Insights reference. The REST API reference pages have a green 'Try It' button on each operation that allows you to try it right in the browser.

Use ARMClient or a similar tool to handle authentication to Azure for the REST API examples.

Summarize results

With the REST API, summarization can be performed by container, definition, or assignment. Here is an example of summarization at the subscription level using Azure Policy Insight's Summarize For Subscription:

HTTP

```http
POST
https://management.azure.com/subscriptions/{subscriptionId}/providers/Micros
oft.PolicyInsights/policyStates/latest/summarize?api-version=2019-10-01
```

The output summarizes the subscription. In the example output below, the summarized compliance are under **value.results.nonCompliantResources** and **value.results.nonCompliantPolicies**. This request provides further details, including each assignment that made up the non-compliant numbers and the definition information for each assignment. Each policy object in the hierarchy provides a **queryResultsUri** that can be used to get additional detail at that level.

JSON

```
{
    "@odata.context":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Micro
soft.PolicyInsights/policyStates/$metadata#summary",
    "@odata.count": 1,
    "value": [{
        "@odata.id": null,
        "@odata.context":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Micro
soft.PolicyInsights/policyStates/$metadata#summary/$entity",
        "results": {
            "queryResultsUri":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Micro
soft.PolicyInsights/policyStates/latest/queryResults?api-version=2019-10-
01&$from=2018-05-18 04:28:22Z&$to=2018-05-19
04:28:22Z&$filter=ComplianceState eq 'NonCompliant'",
            "nonCompliantResources": 15,
            "nonCompliantPolicies": 1
        },
        "policyAssignments": [{
            "policyAssignmentId":
"/subscriptions/{subscriptionId}/resourcegroups/rg-
tags/providers/microsoft.authorization/policyassignments/37ce239ae4304622914
f0c77",
            "policySetDefinitionId": "",
            "results": {
                "queryResultsUri":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Micro
soft.PolicyInsights/policyStates/latest/queryResults?api-version=2019-10-
01&$from=2018-05-18 04:28:22Z&$to=2018-05-19
04:28:22Z&$filter=ComplianceState eq 'NonCompliant' and PolicyAssignmentId
eq '/subscriptions/{subscriptionId}/resourcegroups/rg-
tags/providers/microsoft.authorization/policyassignments/37ce239ae4304622914
f0c77'",
                "nonCompliantResources": 15,
                "nonCompliantPolicies": 1
            },
            "policyDefinitions": [{
                "policyDefinitionReferenceId": "",
                "policyDefinitionId":
"/providers/microsoft.authorization/policydefinitions/1e30110a-5ceb-460c-
a204-c1c3969c6d62",
                "effect": "deny",
                "results": {
                    "queryResultsUri":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Micro
soft.PolicyInsights/policyStates/latest/queryResults?api-version=2019-10-
01&$from=2018-05-18 04:28:22Z&$to=2018-05-19
04:28:22Z&$filter=ComplianceState eq 'NonCompliant' and PolicyAssignmentId
eq '/subscriptions/{subscriptionId}/resourcegroups/rg-
tags/providers/microsoft.authorization/policyassignments/37ce239ae4304622914
f0c77' and PolicyDefinitionId eq
'/providers/microsoft.authorization/policydefinitions/1e30110a-5ceb-460c-
a204-c1c3969c6d62'",
                    "nonCompliantResources": 15
                }
            }]
        }]
    }]
}
```

Query for resources

In the example above, **value.policyAssignments.policyDefinitions.results.queryResultsUri** provides a sample Uri for all non-compliant resources for a specific policy definition. Looking at the **$filter** value, ComplianceState is equal (eq) to 'NonCompliant', PolicyAssignmentId is specified for the policy definition, and then the PolicyDefinitionId itself. The reason for including the PolicyAssignmentId in the filter is because the PolicyDefinitionId could exist in several policy or initiative assignments with different scopes. By specifying both the PolicyAssignmentId and the PolicyDefinitionId, we can be explicit in the results we're looking for. Previously, for PolicyStates we used **latest**, which automatically sets a **from** and **to** time window of the last 24-hours.

HTTP Copy

```
https://management.azure.com/subscriptions/{subscriptionId}/providers/Micros
oft.PolicyInsights/policyStates/latest/queryResults?api-version=2019-10-
01&$from=2018-05-18 04:28:22Z&$to=2018-05-19
04:28:22Z&$filter=ComplianceState eq 'NonCompliant' and PolicyAssignmentId
eq '/subscriptions/{subscriptionId}/resourcegroups/rg-
tags/providers/microsoft.authorization/policyassignments/37ce239ae4304622914
f0c77' and PolicyDefinitionId eq
'/providers/microsoft.authorization/policydefinitions/1e30110a-5ceb-460c-
a204-c1c3969c6d62'
```

The example response below has been trimmed to a single non-compliant resource for brevity. The detailed response has several pieces of data about the resource, the policy or initiative, and the assignment. Notice that you can also see what assignment parameters were passed to the policy definition.

JSON Copy

```
{
    "@odata.context":

"https://management.azure.com/subscriptions/{subscriptionId}/providers/Micro
soft.PolicyInsights/policyStates/$metadata#latest",
    "@odata.count": 15,
    "value": [{
        "@odata.id": null,
        "@odata.context": "https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/policyStates/$metadata#latest/$entity",
        "timestamp": "2018-05-19T04:41:09Z",
        "resourceId": "/subscriptions/{subscriptionId}/resourceGroups/rg-tags/providers/Microsoft.Compute/virtualMachines/linux",
        "policyAssignmentId": "/subscriptions/{subscriptionId}/resourceGroups/rg-tags/providers/Microsoft.Authorization/policyAssignments/37ce239ae4304622914f0c77",
        "policyDefinitionId": "/providers/Microsoft.Authorization/policyDefinitions/1e30110a-5ceb-460c-a204-c1c3969c6d62",
        "effectiveParameters": "",
        "ComplianceState": "NonCompliant",
        "subscriptionId": "{subscriptionId}",
        "resourceType": "/Microsoft.Compute/virtualMachines",
        "resourceLocation": "westus2",
        "resourceGroup": "RG-Tags",
        "resourceTags": "tbd",
        "policyAssignmentName": "37ce239ae4304622914f0c77",
        "policyAssignmentOwner": "tbd",
        "policyAssignmentParameters": "{\"tagName\": {\"value\":\"costCenter\"},\"tagValue\":{\"value\":\"Contoso-Test\"}}",
        "policyAssignmentScope": "/subscriptions/{subscriptionId}/resourceGroups/RG-Tags",
        "policyDefinitionName": "1e30110a-5ceb-460c-a204-c1c3969c6d62",
        "policyDefinitionAction": "deny",
        "policyDefinitionCategory": "tbd",
        "policySetDefinitionId": "",
        "policySetDefinitionName": "",
        "policySetDefinitionOwner": "",
        "policySetDefinitionCategory": "",
        "policySetDefinitionParameters": "",
        "managementGroupIds": "",
        "policyDefinitionReferenceId": ""
    }]
}
```

View events

When a resource is created or updated, a policy evaluation result is generated. Results are called *policy events*. Use the following Uri to view recent policy events associated with the subscription.

HTTP Copy

```
https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/policyEvents/default/queryResults?api-version=2019-10-01
```

Your results resemble the following example:

Your results resemble the following example.

JSON

Copy

```json
{
    "@odata.context":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/policyEvents/$metadata#default",
    "@odata.count": 1,
    "value": [{
        "@odata.id": null,
        "@odata.context":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/policyEvents/$metadata#default/$entity",
        "NumAuditEvents": 16
    }]
}
```

For more information about querying policy events, see the Azure Policy Events reference article.

Azure CLI

The Azure CLI command group for Azure Policy covers most operations that are available in REST or Azure PowerShell. For the full list of available commands, see Azure CLI - Azure Policy Overview.

Example: Getting the state summary for the topmost assigned policy with the highest number of non-compliant resources.

Azure CLI

Copy Try It

```
az policy state summarize --top 1
```

The top portion of the response looks like this example:

JSON

Copy

```json
{
    "odatacontext":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/policyStates/$metadata#summary/$entity",
    "odataid": null,
    "policyAssignments": [{
            "policyAssignmentId":
"/subscriptions/{subscriptionId}/providers/microsoft.authorization/policyassignments/e0704696df5e4c3c81c873e8",
            "policyDefinitions": [{
                "effect": "audit",
                "policyDefinitionGroupNames": [
                    ""
                ],
                "policyDefinitionId":
"/subscriptions/{subscriptionId}/providers/microsoft.authorization/policydef
initions/2e3197b6-1f5b-4b01-920c-b2f0a7e9b18a",
                "policyDefinitionReferenceId": "",
                "results": {
                    "nonCompliantPolicies": null,
                    "nonCompliantResources": 398,
                    "policyDetails": [{
                        "complianceState": "noncompliant",
                        "count": 1
                    }],
                    "policyGroupDetails": [{
                        "complianceState": "noncompliant",
                        "count": 1
                    }],
                    "queryResultsUri":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Micro
soft.PolicyInsights/policyStates/latest/queryResults?api-version=2019-10-
01&$from=2020-07-14 14:01:22Z&$to=2020-07-15 14:01:22Z and
PolicyAssignmentId eq
'/subscriptions/{subscriptionId}/providers/microsoft.authorization/policyass
ignments/e0704696df5e4c3c81c873e8' and PolicyDefinitionId eq
'/subscriptions/{subscriptionId}/providers/microsoft.authorization/policydef
initions/2e3197b6-1f5b-4b01-920c-b2f0a7e9b18a'",
                    "resourceDetails": [{
                            "complianceState": "noncompliant",
                            "count": 398
                        },
                        {
                            "complianceState": "compliant",
                            "count": 4
                        }
                    ]
                }
            }],
    ...
```

Example: Getting the state record for the most recently evaluated resource (default is by timestamp in descending order).

Azure CLI Copy Try It

```
az policy state list --top 1
```

JSON Copy

```
[
  {
    "complianceReasonCode": "",
    "complianceState": "Compliant",
    "effectiveParameters": "",
    "isCompliant": true,
    "managementGroupIds": "{managementgroupId}",
    "odatacontext":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Micro
soft.PolicyInsights/policyStates/$metadata#latest/$entity",
    "odataid": null,
    "policyAssignmentId":
"/subscriptions/{subscriptionId}/providers/microsoft.authorization/policyass
ignments/securitycenterbuiltin",
    "policyAssignmentName": "SecurityCenterBuiltIn",
    "policyAssignmentOwner": "tbd",
    "policyAssignmentParameters": "",
    "policyAssignmentScope": "/subscriptions/{subscriptionId}",
    "policyAssignmentVersion": "",
    "policyDefinitionAction": "auditifnotexists",
    "policyDefinitionCategory": "tbd",
    "policyDefinitionGroupNames": [
      ""
    ],
    "policyDefinitionId":
"/providers/microsoft.authorization/policydefinitions/aa633080-8b72-40c4-
a2d7-d00c03e80bed",
    "policyDefinitionName": "aa633080-8b72-40c4-a2d7-d00c03e80bed",
    "policyDefinitionReferenceId":
"identityenablemfaforownerpermissionsmonitoring",
    "policyDefinitionVersion": "",
    "policyEvaluationDetails": null,
    "policySetDefinitionCategory": "security center",
    "policySetDefinitionId":
"/providers/Microsoft.Authorization/policySetDefinitions/1f3afdf9-d0c9-4c3d-
847f-89da613e70a8",
    "policySetDefinitionName": "1f3afdf9-d0c9-4c3d-847f-89da613e70a8",
    "policySetDefinitionOwner": "",
    "policySetDefinitionParameters": "",
    "policySetDefinitionVersion": "",
    "resourceGroup": "",
    "resourceId": "/subscriptions/{subscriptionId}",
    "resourceLocation": "",
    "resourceTags": "tbd",
    "resourceType": "Microsoft.Resources/subscriptions",
    "subscriptionId": "{subscriptionId}",
    "timestamp": "2020-07-15T08:37:07.903433+00:00"
  }
]
```

Example: Getting the details for all non-compliant virtual network resources.

Azure CLI Copy Try It

```
az policy state list --filter "ResourceType eq
'Microsoft.Network/virtualNetworks'"
```

JSON

Copy

```json
[
  {
    "complianceReasonCode": "",
    "complianceState": "NonCompliant",
    "effectiveParameters": "",
    "isCompliant": false,
    "managementGroupIds": "{managementgroupId}",
    "odatacontext":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/policyStates/$metadata#latest/$entity",
    "odataid": null,
    "policyAssignmentId":
"/subscriptions/{subscriptionId}/providers/microsoft.authorization/policyassignments/e0704696df5e4c3c81c873e8",
    "policyAssignmentName": "e0704696df5e4c3c81c873e8",
    "policyAssignmentOwner": "tbd",
    "policyAssignmentParameters": "",
    "policyAssignmentScope": "/subscriptions/{subscriptionId}",
    "policyAssignmentVersion": "",
    "policyDefinitionAction": "audit",
    "policyDefinitionCategory": "tbd",
    "policyDefinitionGroupNames": [
      ""
    ],
    "policyDefinitionId":
"/subscriptions/{subscriptionId}/providers/microsoft.authorization/policydefinitions/2e3197b6-1f5b-4b01-920c-b2f0a7e9b18a",
    "policyDefinitionName": "2e3197b6-1f5b-4b01-920c-b2f0a7e9b18a",
    "policyDefinitionReferenceId": "",
    "policyDefinitionVersion": "",
    "policyEvaluationDetails": null,
    "policySetDefinitionCategory": "",
    "policySetDefinitionId": "",
    "policySetDefinitionName": "",
    "policySetDefinitionOwner": "",
    "policySetDefinitionParameters": "",
    "policySetDefinitionVersion": "",
    "resourceGroup": "RG-Tags",
    "resourceId": "/subscriptions/{subscriptionId}/resourceGroups/RG-Tags/providers/Microsoft.Network/virtualNetworks/RG-Tags-vnet",
    "resourceLocation": "westus2",
    "resourceTags": "tbd",
    "resourceType": "Microsoft.Network/virtualNetworks",
    "subscriptionId": "{subscriptionId}",
    "timestamp": "2020-07-15T08:37:07.901911+00:00"


  }
]
```

Example: Getting events related to non-compliant virtual network resources that occurred after a specific date.

Azure CLI Copy Try It

```
az policy state list --filter "ResourceType eq 'Microsoft.Network/virtualNetworks'" --from '2020-07-14T00:00:00Z'
```

JSON Copy

```
[
  {
    "complianceReasonCode": "",
    "complianceState": "NonCompliant",
    "effectiveParameters": "",
    "isCompliant": false,
    "managementGroupIds": "{managementgroupId}",
    "odatacontext":
"https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.PolicyInsights/policyStates/$metadata#latest/$entity",
    "odataid": null,
    "policyAssignmentId":
"/subscriptions/{subscriptionId}/providers/microsoft.authorization/policyassignments/e0704696df5e4c3c81c873e8",
    "policyAssignmentName": "e0704696df5e4c3c81c873e8",
    "policyAssignmentOwner": "tbd",
    "policyAssignmentParameters": "",
    "policyAssignmentScope": "/subscriptions/{subscriptionId}",
    "policyAssignmentVersion": "",
    "policyDefinitionAction": "audit",
    "policyDefinitionCategory": "tbd",
    "policyDefinitionGroupNames": [
      ""
    ],
    "policyDefinitionId":
"/subscriptions/{subscriptionId}/providers/microsoft.authorization/policydefinitions/2e3197b6-1f5b-4b01-920c-b2f0a7e9b18a",
    "policyDefinitionName": "2e3197b6-1f5b-4b01-920c-b2f0a7e9b18a",
    "policyDefinitionReferenceId": "",
    "policyDefinitionVersion": "",
    "policyEvaluationDetails": null,
    "policySetDefinitionCategory": "",
    "policySetDefinitionId": "",
    "policySetDefinitionName": "",
    "policySetDefinitionOwner": "",
    "policySetDefinitionParameters": "",
    "policySetDefinitionVersion": "",
    "resourceGroup": "RG-Tags",
    "resourceId": "/subscriptions/{subscriptionId}/resourceGroups/RG-Tags/providers/Microsoft.Network/virtualNetworks/RG-Tags-vnet",
    "resourceLocation": "westus2",
    "resourceTags": "tbd",
    "resourceType": "Microsoft.Network/virtualNetworks",
    "subscriptionId": "{subscriptionId}",
    "timestamp": "2020-07-15T08:37:07.901911+00:00"
  }
]
```

Azure PowerShell

The Azure PowerShell module for Azure Policy is available on the PowerShell Gallery as Az.PolicyInsights. Using PowerShellGet, you can install the module using `Install-Module -Name Az.PolicyInsights` (make sure you have the latest Azure PowerShell installed):

Azure PowerShell Copy Try It

```
# Install from PowerShell Gallery via PowerShellGet
Install-Module -Name Az.PolicyInsights

# Import the downloaded module
Import-Module Az.PolicyInsights

# Login with Connect-AzAccount if not using Cloud Shell
Connect-AzAccount
```

The module has the following cmdlets:

- `Get-AzPolicyStateSummary`
- `Get-AzPolicyState`
- `Get-AzPolicyEvent`
- `Get-AzPolicyRemediation`
- `Remove-AzPolicyRemediation`
- `Start-AzPolicyRemediation`
- `Stop-AzPolicyRemediation`

Example: Getting the state summary for the topmost assigned policy with the highest number of non-compliant resources.

Azure PowerShell Copy Try It

```
PS> Get-AzPolicyStateSummary -Top 1

NonCompliantResources : 15
NonCompliantPolicies  : 1
PolicyAssignments     : {/subscriptions/{subscriptionId}/resourcegroups/RG-
Tags/providers/micros

oft.authorization/policyassignments/37ce239ae4304622914f0c77}
```

Example: Getting the state record for the most recently evaluated resource (default is by timestamp in descending order).

Azure PowerShell Copy Try It

```
PS> Get-AzPolicyState -Top 1

Timestamp                   : 5/22/2018 3:47:34 PM
ResourceId                  :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags/providers/Mi
                              crosoft.Network/networkInterfaces/linux316
PolicyAssignmentId          :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags/providers/Mi

crosoft.Authorization/policyAssignments/37ce239ae4304622914f0c77
PolicyDefinitionId          :
/providers/Microsoft.Authorization/policyDefinitions/1e30110a-5ceb-460c-
a204-c1c3969c6d62
ComplianceState             : NonCompliant
SubscriptionId              : {subscriptionId}
ResourceType                : /Microsoft.Network/networkInterfaces
ResourceLocation            : westus2
ResourceGroup               : RG-Tags
ResourceTags                : tbd
PolicyAssignmentName        : 37ce239ae4304622914f0c77
PolicyAssignmentOwner       : tbd
PolicyAssignmentParameters  : {"tagName":{"value":"costCenter"},"tagValue":
{"value":"Contoso-Test"}}
PolicyAssignmentScope       :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags
PolicyDefinitionName        : 1e30110a-5ceb-460c-a204-c1c3969c6d62
PolicyDefinitionAction      : deny
PolicyDefinitionCategory    : tbd
```

Example: Getting the details for all non-compliant virtual network resources.

Azure PowerShell Copy Try It

```
PS> Get-AzPolicyState -Filter "ResourceType eq
'/Microsoft.Network/virtualNetworks'"

Timestamp                   : 5/22/2018 4:02:20 PM
ResourceId                  :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags/providers/Mi
                              crosoft.Network/virtualNetworks/RG-Tags-vnet
PolicyAssignmentId          :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags/providers/Mi


crosoft.Authorization/policyAssignments/37ce239ae4304622914f0c77
PolicyDefinitionId          :
/providers/Microsoft.Authorization/policyDefinitions/1e30110a-5ceb-460c-
a204-c1c3969c6d62
```

```
ComplianceState            : NonCompliant
SubscriptionId             : {subscriptionId}
ResourceType               : /Microsoft.Network/virtualNetworks
ResourceLocation           : westus2
ResourceGroup              : RG-Tags
ResourceTags               : tbd
PolicyAssignmentName       : 37ce239ae4304622914f0c77
PolicyAssignmentOwner      : tbd
PolicyAssignmentParameters : {"tagName":{"value":"costCenter"},"tagValue":
{"value":"Contoso-Test"}}
PolicyAssignmentScope      :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags
PolicyDefinitionName       : 1e30110a-5ceb-460c-a204-c1c3969c6d62
PolicyDefinitionAction     : deny
PolicyDefinitionCategory   : tbd
```

Example: Getting events related to non-compliant virtual network resources that occurred after a specific date.

Azure PowerShell

Copy Try It

```
PS> Get-AzPolicyEvent -Filter "ResourceType eq
'/Microsoft.Network/virtualNetworks'" -From '2018-05-19'

Timestamp                  : 5/19/2018 5:18:53 AM
ResourceId                 :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags/providers/Mi
                             crosoft.Network/virtualNetworks/RG-Tags-vnet
PolicyAssignmentId         :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags/providers/Mi

crosoft.Authorization/policyAssignments/37ce239ae4304622914f0c77
PolicyDefinitionId         :
/providers/Microsoft.Authorization/policyDefinitions/1e30110a-5ceb-460c-
a204-c1c3969c6d62
ComplianceState            : NonCompliant
SubscriptionId             : {subscriptionId}
ResourceType               : /Microsoft.Network/virtualNetworks
ResourceLocation           : eastus
ResourceGroup              : RG-Tags
ResourceTags               : tbd
PolicyAssignmentName       : 37ce239ae4304622914f0c77
PolicyAssignmentOwner      : tbd
PolicyAssignmentParameters : {"tagName":{"value":"costCenter"},"tagValue":
{"value":"Contoso-Test"}}
PolicyAssignmentScope      :
/subscriptions/{subscriptionId}/resourceGroups/RG-Tags
PolicyDefinitionName       : 1e30110a-5ceb-460c-a204-c1c3969c6d62
PolicyDefinitionAction     : deny
PolicyDefinitionCategory   : tbd
TenantId                   : {tenantId}
PrincipalOid               : {principalOid}
```

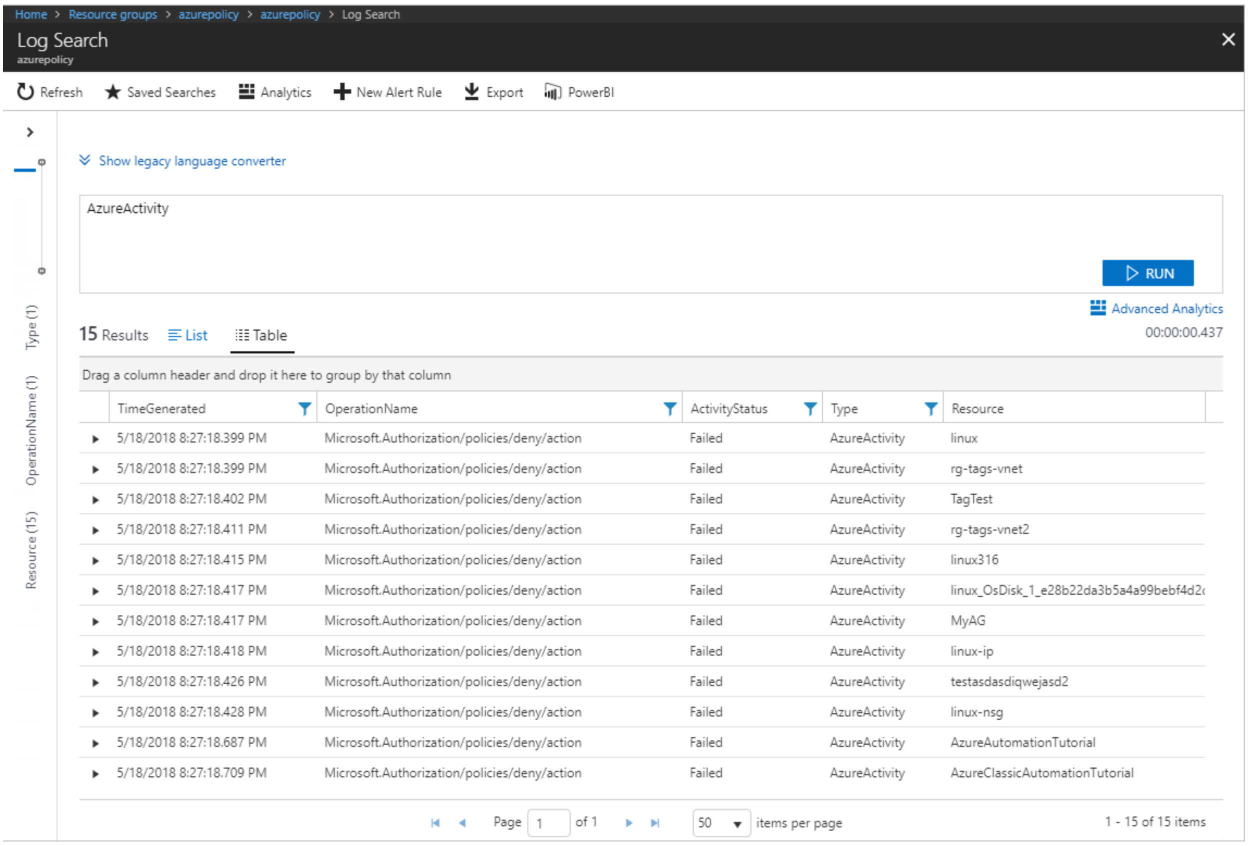
The **PrincipalOid** field can be used to get a specific user with the Azure PowerShell cmdlet `Get-AzADUser`. Replace **{principalOid}** with the response you get from the previous example.

Azure PowerShell Copy Try It

```
PS> (Get-AzADUser -ObjectId {principalOid}).DisplayName
Trent Baker
```

Azure Monitor logs

If you have a Log Analytics workspace with `AzureActivity` from the Activity Log Analytics solution tied to your subscription, you can also view non-compliance results from the evaluation cycle using simple Kusto queries and the `AzureActivity` table. With details in Azure Monitor logs, alerts can be configured to watch for non-compliance.



AzureActivity

15 Results

00:00:00.437

TimeGenerated	OperationName	ActivityStatus	Type	Resource
5/18/2018 8:27:18.399 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	linux
5/18/2018 8:27:18.399 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	rg-tags-vnet
5/18/2018 8:27:18.402 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	TagTest
5/18/2018 8:27:18.411 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	rg-tags-vnet2
5/18/2018 8:27:18.415 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	linux316
5/18/2018 8:27:18.417 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	linux_OsDisk_1_e28b22da3b5a4a99bebf4d2
5/18/2018 8:27:18.417 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	MyAG
5/18/2018 8:27:18.418 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	linux-ip
5/18/2018 8:27:18.426 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	testasdasdiqwejasd2
5/18/2018 8:27:18.428 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	linux-nsg
5/18/2018 8:27:18.687 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	AzureAutomationTutorial
5/18/2018 8:27:18.709 PM	Microsoft.Authorization/policies/deny/action	Failed	AzureActivity	AzureClassicAutomationTutorial

Next steps

- Review examples at Azure Policy samples.
- Review the Azure Policy definition structure.
- Review Understanding policy effects.
- Understand how to programmatically create policies.

- Learn how to remediate non-compliant resources.
- Review what a management group is with Organize your resources with Azure management groups.

Is this page helpful?

👍 Yes 👎 No