Overview of IAM Conditions

Archived: 2026-04-06 00:34:56 UTC

This page describes the Conditions feature of Identity and Access Management (IAM). You can use IAM Conditions to define and enforce conditional, attribute-based access control for Google Cloud resources.

Conditions and policy types

You can use conditions in the following places:

- Allow policy role bindings, including role bindings managed by Privileged Access Manager entitlements
- Deny policy rules
- Policy bindings for principal access boundary policies

The following sections describe how you can use conditions in each of these places to enforce attribute-based access control.

Conditions in allow policies

You can use conditions in allow policies to choose to grant access to principals only if specified conditions are met. For example, you could grant temporary access to users so they can resolve a production issue, or you could grant access only to employees making requests from your corporate network.

Conditions are specified in the role bindings of a resource's allow policy. If a role binding has a condition, then the principals in the role are only granted the role if the condition expression evaluates to `true`.

To add a condition to a role binding, you define the `condition` field:

```
"bindings": [
  {
    "role": "ROLE",
    "members": [
      "MEMBER_1",
      "MEMBER_2"
    ],
    "condition": {
      "title": "TITLE",
      "description": "DESCRIPTION",
      "expression": "EXPRESSION"
    }
  }
]
```

If you're using Privileged Access Manager entitlements, you can also add conditions to the roles in that entitlement. When a user successfully requests a grant for that entitlement, they are granted the role with the specified condition.

To learn more about the fields in a condition, see Condition structure on this page.

Only some resource types accept conditions in role bindings. However, you can grant conditional access to other resource types by granting roles at the organization or project level.

As a best practice, don't add more than 100 conditional role bindings to a single allow policy. If you use a larger number of conditional role bindings, you might exceed the overall size limit for allow policies.

To learn how to add, modify, and remove conditional role bindings, see Managing conditional role bindings.

Conditions in deny policies

You can use conditions in deny policies to apply a deny rule only if a certain condition is met. For example, you could deny a permission only if the resource that the principal is trying to access is tagged as a part of the `prod` environment.

Conditions are specified in the deny rules in a resource's deny policies. If the condition evaluates to `true` or cannot be evaluated, the deny rule applies and the principals are unable to use the specified permissions. If the condition evaluates to `false`, the deny rule does not apply and the principals can use the specified permissions if they have them.

Conditions in deny policies have the same structure as conditions in allow policies; however, they only recognize resource tag functions.

To add a condition to a deny rule, you define the `denialCondition` field:

```
"rules": [
  {
    "denyRule": {
      "deniedPrincipals": [
        "PRINCIPAL_1",
        "PRINCIPAL_2"
      ],
      "exceptionPrincipals": [
        "EXCEPTION_PRINCIPAL_1",
        "EXCEPTION_PRINCIPAL_2"
      ],
      "deniedPermissions": [
        "DENIED_PERMISSION_1",
        "DENIED_PERMISSION_2"
      ],
      "denialCondition": {
        "title": "TITLE",
```

```
        "description": "DESCRIPTION",
        "expression": "EXPRESSION"
      }
    }
  }
]
```

To learn more about the fields in a condition, see Condition structure on this page.

To learn how to create and manage deny policies, see Deny access.

Conditions in principal access boundary policy bindings

You can use conditions in policy bindings for principal access boundary policies to refine the principal set that the principal access boundary policy applies to. For example, you could only enforce a policy for service accounts, or exempt `super-admin@example.com` from a policy.

Conditions are specified in each policy binding. If a policy binding has a condition, then the policy in the policy binding is enforced only if the condition evaluates to `true`.

To add a condition to a policy binding, you define the `condition` field in the policy binding:

```
{
  "displayName": "DISPLAY_NAME",
  "target": {
    "principalSet": "PRINCIPAL_SET"
  },
  "policyKind": "PRINCIPAL_ACCESS_BOUNDARY",
  "policy": "PAB_POLICY",
  "condition": {
    "title": "TITLE",
    "description": "DESCRIPTION",
    "expression": "EXPRESSION"
  }
}
```

To learn more about the fields in a condition, see Condition structure on this page.

To learn how to create policy bindings for principal access boundary policies, see Apply a principal access boundary policy to a principal set.

Condition structure

The `condition` object has the following structure:

```
"condition": {
    "title": ...,
    "description": ...,
    "expression": ...
}
```

The condition's `title` is required, but the `description` is optional. Both the title and description are purely informational fields to help you identify and describe the condition.

The `expression` field is required. It defines an attribute-based logic expression using a subset of the Common Expression Language (CEL). The condition expression can contain multiple statements; each statement evaluates one attribute. Statements are combined using logical operators, following the CEL language specification.

CEL for conditions

Common Expression Language, or CEL, is the expression language used to specify an expression in IAM Conditions. It is tailored to express attribute-based logic expressions. For more information, see the CEL spec and its language definition.

In IAM Conditions, a subset of CEL is used to make boolean authorization decisions based on attribute data. In general, a condition expression consists of one or more statements that are joined by logical operators (`&&` , `||` , or `!`).

Each statement expresses an attribute-based control rule, and ultimately determines whether the role binding, deny rule, or policy binding applies.

Conditions in IAM Conditions use the following CEL features:

- **Variables**: Conditions use *variables* to express a given attribute, such as `request.time` (of type Timestamp) or `resource.name` (of type String). These variables are populated with value based on the context at runtime.
- **Operators**: Every data type, such as Timestamp or String, supports a set of *operators* that can be used to create a logic expression. Most commonly, operators are used to compare the value contained in a variable with a literal value, such as `resource.service == 'compute.googleapis.com'` . In this example, if the input value of `resource.service` is `compute.googleapis.com` , then the expression evaluates to `true` .
- **Functions**: A function is a compound operator for data types that support more complex operations. In condition expressions, there are predefined functions that can be used with a given data type. For example, `request.path.startsWith('/finance')` uses a String prefix match function, and evaluates to `true` if the value of `request.path` contains a matching prefix, such as `/finance` .
- **Logical operators**: Conditions supports three logical operators that can be used to build complex logic expressions from basic expression statements: `&&` , `||` , and `!` . These logical operators make it possible to use multiple input variables in a condition expression. For example: `request.time.getFullYear() < 2020 && resource.service == 'compute.googleapis.com'` joins two basic statements, and requires both statements to be met in order to produce a `true` overall evaluation result.

For more information about supported variables, operators, and functions, see the attribute reference.

Condition attributes

Condition attributes are based on the requested resource—for example, its type or name—or on details about the request—for example, its timestamp or destination IP address.

The condition attributes that you can use in a condition expression depend on the policy type that you're writing conditions for. For a full list of condition attributes and more information about the attributes supported for each policy type, see the attribute reference.

The following sections show examples of some of the attributes that you can use in conditions.

Resource attributes

You can use resource attributes to write conditions that evaluate the resource in the access request. The attributes that you can evaluate include the following:

- The resource type
- The resource name
- The Google Cloud service being used
- The tags attached to the resource

You can use any of these attributes in allow policy role bindings. Additionally, you can use the resource tags attribute in deny policy deny rules.

For a complete list of resource attributes, see the resource attributes reference.

To learn how to use resource attributes to configure resource-based access, see Configuring resource-based access.

Example expressions

In a role binding, the following condition expression allows access to Compute Engine VM instances, but no other type of resource:

```
resource.type == 'compute.googleapis.com/Instance'
```

In a role binding, the following condition expression allows access to Cloud Storage resources, but no other service's resources:

```
resource.service == 'storage.googleapis.com'
```

In a role binding, the following condition expression allows access only to Cloud Storage objects inside a specific bucket:

```
 resource.type == 'storage.googleapis.com/Object' &&
resource.name.startsWith('projects/_/buckets/exampleco-site-assets/')
```

In a deny rule, the following condition expression denies access to Google Cloud resources that have the tag `env: prod`:

```
resource.matchTag('123456789012/env', 'prod')
```

Principal attributes

The principal attributes let you write conditions based on the principal that issued the request. The attributes that you can evaluate include the following:

- The type of principal in the request
- The identity of the principal in the request

You can use these attributes in policy bindings for principal access boundary policies.

For details, see the conditions attribute reference.

Example expressions

In a principal access boundary policy binding, the following condition expression ensures that the policy in the binding is only enforced for service accounts:

```
principal.type == 'iam.googleapis.com/ServiceAccount'
```

In a principal access boundary policy binding, the following condition expression ensures that the policy in the binding isn't enforced for `super-admin@example.com`:

```
principal.subject != 'super-admin@example.com'
```

Request attributes

You can use request attributes to write conditions that evaluate details about the request, such as the following:

- The access level

- The date and time
- The destination IP address and port (for IAP TCP tunneling)
- The expected URL host or path (for IAP)

You can use these attributes in allow policy role bindings.

Example access level expression (for IAP only)

In the following example, your organization defines an access level, `CorpNet`, that limits access to the range of IP addresses where traffic enters and exits a corporate network. Then, you add the following condition expression to a role binding to allow access only if the request meets the `CorpNet` access level:

```
 'accessPolicies/199923665455/accessLevels/CorpNet' in
request.auth.access_levels
```

Your organization defines access levels based on attributes of the request, such as origin IP address, device attributes, the time of day, and more. For more details, see the Access Context Manager documentation.

Example API attribute expression

In a role binding for a role with the `iam.projects.setIamPolicy` permission, the following condition expression allows a user to grant and revoke only the Billing Account Administrator (`roles/billing.admin`) role on the project:

```
 api.getAttribute('iam.googleapis.com/modifiedGrantsByRole', [])
                .hasOnly(['roles/billing.admin'])
```

To learn more about using API attributes to limit role granting, see Setting limits on granting roles.

Example date/time expressions

In a role binding, the following condition expression allows access until midnight on January 1st, 2021:

```
 request.time < timestamp('2021-01-01T00:00:00Z')
```

In a role binding, the following condition expression allows access only during specified working hours, based on the time zone for Berlin, Germany:

```
 request.time.getHours('Europe/Berlin') >= 9 &&
request.time.getHours('Europe/Berlin') <= 17 &&
// Days of the week range from 0 to 6, where 0 == Sunday and 6 == Saturday.
```

```
request.time.getDayOfWeek('Europe/Berlin') >= 1 &&
request.time.getDayOfWeek('Europe/Berlin') <= '
```

In a role binding, the following condition expression allows access only for June of 2020, based on the time zone for Berlin, Germany:

```
 request.time.getFullYear('Europe/Berlin') == 2020
request.time.getMonth('Europe/Berlin') < 6
```

To specify a timestamp, use RFC 3339 format. To specify a time zone, use the identifiers in the IANA Time Zone Database.

For more details about date/time expressions, see the CEL specification.

To learn how to use date/time expressions to configure temporary access, see Configuring temporary access.

Example destination IP and port expressions (for IAP TCP tunneling)

In a role binding, the following condition expression allows access to an internal destination IP address or port number:

```
 destination.ip == '14.0.0.1'
destination.ip != '127.0.0.1'
destination.port == 22
destination.port > 21 && destination.port <= 23
```

Example forwarding rule expressions

In a role binding, the following condition expression allows access for a principal if the request is not creating a forwarding rule, or if the request is creating a forwarding rule for an internal Google Cloud load balancer:

```
 !compute.isForwardingRuleCreationOperation() || (
  compute.isForwardingRuleCreationOperation() &&
  compute.matchLoadBalancingSchemes([
    'INTERNAL', 'INTERNAL_MANAGED', 'INTERNAL_SELF_MANAGED'
  ])
)
```

For details about load-balancing schemes, see Using IAM Conditions on Google Cloud load balancers.

Example URL host or path expressions (for IAP)

In a role binding, the following condition expression allows access only for certain subdomains or URL paths in the request:

```
 request.host == 'hr.example.com'
request.host.endsWith('.example.com')
request.path == '/admin/payroll.js'
request.path.startsWith('/admin')
```

Example expression with different types of attributes

In a role binding, the following condition expression allows access if the request is made during a specific time, matching a resource name prefix, with the chosen access level, and for a specific resource type:

```
 request.time > timestamp('2018-08-03T16:00:00-07:00') &&
request.time < timestamp('2018-08-03T16:05:00-07:00') &&
((resource.name.startsWith('projects/project-123/zones/us-east1-b/instances/dev') ||
 (resource.name.startsWith('projects/project-123/zones/us-east1-b/instances/prod') &&
  'accessPolicies/34569256/accessLevels/CorpNet' in request.auth.access_levels)) ||
 resource.type != 'compute.googleapis.com/Instance')
```

What's next

- Get details about the condition attributes that you can use to manage access.
- Learn more about allow policies.
- Find resource types that accept conditional role bindings.
