- `IAMPolicy` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.IAMPolicy) (interface)
- `Binding` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.Binding) (message)
- `GetIamPolicyRequest` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.GetIamPolicyRequest) (message)
- `Policy` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.Policy) (message)
- `SetIamPolicyRequest` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.SetIamPolicyRequest) (message)
- `TestIamPermissionsRequest` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.TestIamPermissionsRequest) (message)
- `TestIamPermissionsResponse` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.TestIamPermissionsResponse) (message)

Manages Identity and Access Management (IAM) policies.

Any implementation of an API that offers access control features implements the google.iam.v1.IAMPolicy interface.

Access control is applied when a principal (user or service account), takes some action on a resource exposed by a service. Resources, identified by URI-like names, are the unit of access control specification. Service implementations can choose the granularity of access control and the supported permissions for their resources. For example one database service may allow access control to be specified only at the Table level, whereas another might allow access control to also be specified at the Column level.

See google.iam.v1.Policy

This is intentionally not a CRUD style API because access control policies are created and deleted implicitly with the resources to which they are attached.

GetIamPolicy

`rpc GetIamPolicy(GetIamPolicyRequest` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.GetIamPolicyRequest)`)`
`returns (Policy` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.Policy)`)`

Gets the access control policy for a resource. Returns an empty policy if the resource exists and does not have a policy set.

Authorization Scopes

Requires the following OAuth scope:

- `https://www.googleapis.com/auth/cloud-platform`

For more information, see the Authentication Overview (https://cloud.google.com/docs/authentication/).

SetIamPolicy

SetIamPolicy

`rpc SetIamPolicy(SetIamPolicyRequest` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.SetIamPolicyRequest)`) returns (Policy` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.Policy)`)`

Sets the access control policy on the specified resource. Replaces any existing policy.

Authorization Scopes

Requires the following OAuth scope:

- `https://www.googleapis.com/auth/cloud-platform`

For more information, see the Authentication Overview (https://cloud.google.com/docs/authentication/).

TestIamPermissions

`rpc TestIamPermissions(TestIamPermissionsRequest` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.TestIamPermissionsRequest)`) returns (TestIamPermissionsResponse` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.TestIamPermissionsResponse)`)`

Returns permissions that a caller has on the specified resource. If the resource does not exist, this will return an empty set of permissions, not a NOT_FOUND error.

Note: This operation is designed to be used for building permission-aware UIs and command-line tools, not for authorization checking. This operation may "fail open" without warning.

Authorization Scopes

Requires the following OAuth scope:

- `https://www.googleapis.com/auth/cloud-platform`

For more information, see the Authentication Overview (https://cloud.google.com/docs/authentication/).

Associates `members` with a `role`.

Fields	
`role`	`string` Role that is assigned to **`members`**. For example, **`roles/viewer`**, **`roles/editor`**, or **`roles/owner`**.
`members[]`	`string` Specifies the identities requesting access for a Cloud Platform resource. **`members`** can have the following values: • **`allUsers`**: A special identifier that represents anyone who is on the internet; with or without a Google account. • **`allAuthenticatedUsers`**: A special identifier that represents anyone who is authenticated with a Google account or a service account. • **`user:{emailid}`**: An email address that represents a specific Google account. For example, **`alice@gmail.com`** . • **`serviceAccount:{emailid}`**: An email address that represents a service account. For example, **`my-other-app@appspot.gserviceaccount.com`**. • **`group:{emailid}`**: An email address that represents a Google group. For example, **`admins@example.com`**. • **`domain:{domain}`**: The G Suite domain (primary) that represents all the users of that domain. For example, **`google.com`** or **`example.com`**.
`condition`	Expr (/video-intelligence/automl/docs/reference/rpc/google.type#google.type.Expr) The condition that is associated with this binding. NOTE: an unsatisfied condition will not allow user access via current binding. Different bindings, including their conditions, are examined independently.

Request message for `GetIamPolicy` method.

Fields	
`resource`	`string` REQUIRED: The resource for which the policy is being requested. See the operation documentation for the appropriate value for this field.

Defines an Identity and Access Management (IAM) policy. It is used to specify access control policies for Cloud Platform resources.

A `Policy` consists of a list of `bindings`. A `binding` binds a list of `members` to a `role`, where the members can be user accounts, Google groups, Google domains, and service accounts. A `role` is a named list of permissions defined by IAM.

JSON Example

YAML Example

For a description of IAM and its features, see the IAM developer's guide (https://cloud.google.com/iam/docs).

Fields

`version (deprecated)`	`int32` ⚠ This item is deprecated! Deprecated.
`bindings[]`	`Binding` (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.Binding) Associates a list of **`members`** to a **`role`**. **`bindings`** with no members will result in an error.

Fields	
`etag`	`bytes` **`etag`** is used for optimistic concurrency control as a way to help prevent simultaneous updates of a policy from overwriting each other. It is strongly suggested that systems make use of the **`etag`** in the read-modify-write cycle to perform policy updates in order to avoid race conditions: An **`etag`** is returned in the response to **`getIamPolicy`**, and systems are expected to put that etag in the request to **`setIamPolicy`** to ensure that their change will be applied to the same version of the policy. If no **`etag`** is provided in the call to **`setIamPolicy`**, then the existing policy is overwritten blindly.

Request message for `SetIamPolicy` method.

Fields	
`resource`	`string` REQUIRED: The resource for which the policy is being specified. See the operation documentation for the appropriate value for this field.
`policy`	Policy (/video-intelligence/automl/docs/reference/rpc/google.iam.v1#google.iam.v1.Policy) REQUIRED: The complete policy to be applied to the **`resource`**. The size of the policy is limited to a few 10s of KB. An empty policy is a valid policy but certain Cloud Platform services (such as Projects) might reject them.

Request message for `TestIamPermissions` method.

Fields	
`resource`	`string` REQUIRED: The resource for which the policy detail is being requested. See the operation documentation for the appropriate value for this field.
`permissions[]`	`string` The set of permissions to check for the `resource`. Permissions with wildcards (such as '*' or 'storage.*') are not allowed. For more information see IAM Overview (https://cloud.google.com/iam/docs/overview#permissions).

Response message for `TestIamPermissions` method.

Fields	
`permissions[]`	`string` A subset of `TestPermissionsRequest.permissions` that the caller is allowed.