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).

JSON representation

Fields	

`version`
`(deprecated)`
	`number`
	⚠ This item is deprecated!
	Deprecated.

`bindings[]`	`object(Binding (/source-repositories/docs/reference/rest/v1/Policy#Binding))` Associates a list of `members` to a `role`. `bindings` with no members will result in an error.
`auditConfigs[]`	`object(AuditConfig (/source-repositories/docs/reference/rest/v1/Policy#AuditConfig))` Specifies cloud audit logging configuration for this policy.

Fields	
`etag`	`string (bytes` (https://developers.google.com/discovery/v1/type-format) `format)` `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. A base64-encoded string.

Associates `members` with a `role`.

JSON representation

Fields	
`role`	`string` Role that is assigned to `members`. For example, `roles/viewer`, `roles/editor`, or `roles/owner`.

Fields	
`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}`**: A Google Apps domain name that represents all the users of that domain. For example, **`google.com`** or **`example.com`**.
`condition`	`object(Expr` (/source-repositories/docs/reference/rest/v1/Policy#Expr) `)` Unimplemented. 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.

Represents an expression text. Example:

JSON representation

Fields	
`expression`	`string` Textual representation of an expression in Common Expression Language syntax. The application context of the containing message determines which well-known feature set of CEL is supported.
`title`	`string` An optional title for the expression, i.e. a short string describing its purpose. This can be used e.g. in UIs which allow to enter the expression.
`description`	`string` An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.
`location`	`string` An optional string indicating the location of the expression for error reporting, e.g. a file name and a position in the file.

Specifies the audit configuration for a service. The configuration determines which permission types are logged, and what identities, if any, are exempted from logging. An AuditConfig must have one or more AuditLogConfigs.

If there are AuditConfigs for both `allServices` and a specific service, the union of the two AuditConfigs is used for that service: the log_types specified in each AuditConfig are enabled, and the exemptedMembers in each AuditLogConfig are exempted.

Example Policy with multiple AuditConfigs:

For fooservice, this policy enables DATA_READ, DATA_WRITE and ADMIN_READ logging. It also exempts foo@gmail.com (mailto:foo@gmail.com) from DATA_READ logging, and bar@gmail.com (mailto:bar@gmail.com) from DATA_WRITE logging.

JSON representation

Fields	
`service`	`string` Specifies a service that will be enabled for audit logging. For example, `storage.googleapis.com`, `cloudsql.googleapis.com`. `allServices` is a special value that covers all services.
`auditLogConfigs[]`	`object(`AuditLogConfig (/source-repositories/docs/reference/rest/v1/Policy#AuditLogConfig)`)` The configuration for logging of each type of permission.

Provides the configuration for logging a type of permissions. Example:

This enables 'DATA_READ' and 'DATA_WRITE' logging, while exempting foo@gmail.com (mailto:foo@gmail.com) from DATA_READ logging.

JSON representation

Fields	
`logType`	`enum(LogType (/source-repositories/docs/reference/rest/v1/Policy#LogType))` The log type that this config enables.
`exemptedMembers[]`	`string` Specifies the identities that do not cause logging for this type of permission. Follows the same format of `Binding.members` (/source-repositories/docs/reference/rest/v1/Policy#Binding.FIELDS.members) .

The list of valid permission types for which logging can be configured. Admin writes are always logged, and are not configurable.

Enums

Enums	
`LOG_TYPE_UNSPECIFIED`	Default case. Should never be this.
`ADMIN_READ`	Admin reads. Example: CloudIAM getIamPolicy
`DATA_WRITE`	Data writes. Example: CloudSQL Users create
`DATA_READ`	Data reads. Example: CloudSQL Users list