Access control

This document describes the access control options available to you in Pub/Sub.

Overview

Pub/Sub uses Cloud Identity and Access Management (https://cloud.google.com/iam) (Cloud IAM) for access control.

In Pub/Sub, access control can be configured at the project level and at the individual resource level. For example:

- Grant access on a per-topic or per-subscription basis, rather than for the whole Cloud project.
- Grant access with limited capabilities, such as to only publish messages to a topic, or to only consume messages from a subscription, but not to delete the topic or subscription.
- Grant access to all Pub/Sub resources within a project to a group of developers.

For a detailed description of Cloud IAM and its features, see the Cloud IAM documentation (https://cloud.google.com/iam). In particular, see Granting, changing, and revoking access to resources (https://cloud.google.com/iam/docs/granting-changing-revoking-access).

Every Pub/Sub method requires the caller to have the necessary permissions. For a list of the permissions and roles Pub/Sub Cloud IAM supports, see the Roles (https://cloud.google.com/pubsub/docs/access-control#roles) section, below.

Note: Pub/Sub is not associated with any specific IP address. This is relevant if you rely on IP-based firewall rules.

Permissions and roles

This section summarizes the permissions and roles Pub/Sub Cloud IAM supports.

Required permissions

The following table lists the permissions that the caller must have to call each method:

Method
projects.snapshots.create (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.snapsh
projects.snapshots.delete (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.snapsh
projects.snapshots.getIamPolicy (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects
projects.snapshots.list (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.snapshot
projects.snapshots.setIamPolicy (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects
projects.snapshots.testIamPermissions (https://cloud.google.com/pubsub/docs/reference/rest/v1/
projects.subscriptions.acknowledge (https://cloud.google.com/pubsub/docs/reference/rest/v1/proj
projects.subscriptions.create (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.su

`projects.subscriptions.delete` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.su
`projects.subscriptions.get` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.subsc
`projects.subscriptions.getIamPolicy` (https://cloud.google.com/pubsub/docs/reference/rest/v1/pro
`projects.subscriptions.list` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.subs
`projects.subscriptions.modifyAckDeadline` (https://cloud.google.com/pubsub/docs/reference/rest/

`projects.subscriptions.modifyPushConfig` (https://cloud.google.com/pubsub/docs/reference/rest/v
`projects.subscriptions.pull` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.subs
`projects.subscriptions.seek` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.subs
`projects.subscriptions.setIamPolicy` (https://cloud.google.com/pubsub/docs/reference/rest/v1/pro
`projects.subscriptions.testIamPermissions` (https://cloud.google.com/pubsub/docs/reference/res
`projects.topics.create` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/crea
`projects.topics.delete` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/dele
`projects.topics.get` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/get)
`projects.topics.getIamPolicy` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.top
`projects.topics.list` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/list)

`projects.topics.publish` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/pu	
`projects.topics.setIamPolicy` (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.top	
`projects.topics.subscriptions.list` (https://cloud.google.com/pubsub/docs/reference/rest/v1/proj	
`projects.topics.testIamPermissions` (https://cloud.google.com/pubsub/docs/reference/rest/v1/proj	

Roles

The following table lists the Pub/Sub Cloud IAM roles with a corresponding list of all the permissions each role includes. Note that every permission is applicable to a particular resource type.

These preconfigured roles address many typical use cases. However, you might need a role that includes a custom set of permissions. For instance, you may wish to create a role that allows a user to create a subscription in a project, without letting them delete or update existing topics or subscriptions in the project. In those cases, you may be able to create an Cloud IAM custom role (https://cloud.google.com/iam/docs/understanding-custom-roles) that meets your needs.

Role	includes permission(s):	for resource type:
`roles/pubsub.publisher`	`pubsub.topics.publish`	Topic
`roles/pubsub.subscriber`	`pubsub.snapshots.seek`	Snapshot
	`pubsub.subscriptions.consume`	Subscription
	`pubsub.topics.attachSubscription`	Topic
`roles/pubsub.viewer` or `roles/viewer`	`pubsub.snapshots.get`	Snapshot
	`pubsub.snapshots.list`	Project
	`pubsub.subscriptions.get`	Subscription
	`pubsub.subscriptions.list`	Project
	`pubsub.topics.get`	Topic

	`pubsub.topics.list`	Project
	`resourcemanager.projects.get`	Project
	`servicemanagement.projectSettings.get`	Project
	`serviceusage.quotas.get`	Project
	`serviceusage.services.get`	Project
	`serviceusage.services.list`	Project
`roles/pubsub.editor` or `roles/editor`	All of the above, as well as:	
	`pubsub.snapshots.create`	Project
	`pubsub.snapshots.delete`	Snapshot
	`pubsub.snapshots.update`	Snapshot
	`pubsub.subscriptions.create`	Project
	`pubsub.subscriptions.delete`	Subscription
	`pubsub.subscriptions.update`	Subscription
	`pubsub.topics.create`	Project
	`pubsub.topics.delete`	Topic
	`pubsub.topics.update`	Topic
	`pubsub.topics.updateTag`	Topic
`roles/pubsub.admin` or `roles/owner`	All of the above, as well as:	
	`pubsub.snapshots.getIamPolicy`	Snapshot
	`pubsub.snapshots.setIamPolicy`	Snapshot
	`pubsub.subscriptions.getIamPolicy`	Subscription
	`pubsub.subscriptions.setIamPolicy`	Subscription
	`pubsub.topics.getIamPolicy`	Topic
	`pubsub.topics.setIamPolicy`	Topic

The roles roles/owner, roles/editor, and roles/viewer include also permissions for other Google Cloud services.

Controlling access via the Google Cloud Console

You can use the GCP Console to manage access control for your topics and projects.

To set access controls at the project level:

1. Open the Cloud IAM page (https://console.cloud.google.com/project/_/iam-admin/iam) in the Cloud Console.
2. Select your project, and click **Continue**.
3. Click **Add Member**.
4. Enter the email address of a new member to whom you have not granted any Cloud IAM role previously.
5. Select a role from the drop-down menu.
6. Click Add.
7. Verify that the member is listed under the role that you granted.

To set access controls for topics and subscriptions:

1. Navigate to the Pub/Sub topics page (https://console.cloud.google.com/project/_/cloudpubsub/topic/list) in the console.
2. Select your Pub/Sub-enabled project.
3. Select the topic or subscription.

   You can set permissions for multiple topics at one time. To set permissions for a topic's subscription, expand the topic and click the subscription to open it in its own page.
4. Click **Permissions**. In the pane that appears:
   a. Type in a member name or names.
   b. Select a role from the drop-down menu.
   c. Click **Add**.

Controlling access via the Cloud IAM API

The Pub/Sub Cloud IAM API lets you set and get policies on individual topics and subscriptions in a project, and test a user's permissions for a given resource. As with the regular Pub/Sub methods, you can invoke the Cloud IAM API methods via the client libraries, or the API Explorer, or directly over HTTP.

Note that you cannot use the Pub/Sub Cloud IAM API to manage policies at the Google Cloud project level.

The following sections give examples for how to set and get a policy, and how to test what permissions a caller has for a given resource.

Getting a policy

The `getIamPolicy()` method allows you to get an existing policy (https://cloud.google.com/iam/docs/managing-policies). This method returns a JSON object containing the policy associated with the resource.

Here is some sample code to get a policy for a subscription (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.subscriptions/getIamPolicy):

C# PYTHON MORE

Before trying this sample, follow the Python setup instructions in Quickstart: Using Client Libraries (https://cloud.google.com/pubsub/docs/quickstart-client-libraries). For more information, see the Pub/Sub Python API reference documentation (https://googleapis.github.io/google-cloud-python/latest/pubsub/).

GOOGLECLOUDPLATFORM/PYTHON-DOCS-SAMPLES/BLOB/MASTER/PUBSUB/CLOUD-CLIENT/IAM.PY)

FEEDBACK (#)

```
client = pubsub_v1.SubscriberClient()
subscription_path = client.subscription_path(project, subscription_name)

policy = client.get_iam_policy(subscription_path)

print("Policy for subscription {}:".format(subscription_path))
```

```
for binding in policy.bindings:
    print("Role: {}, Members: {}".format(binding.role, binding.members))
```

Here is some sample code to get a policy for a topic (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/getIamPolicy):

C# PYTHON MORE

Before trying this sample, follow the Python setup instructions in Quickstart: Using Client Libraries (https://cloud.google.com/pubsub/docs/quickstart-client-libraries). For more information, see the Pub/Sub Python API reference documentation (https://googleapis.github.io/google-cloud-python/latest/pubsub/).

GOOGLECLOUDPLATFORM/PYTHON-DOCS-SAMPLES/BLOB/MASTER/PUBSUB/CLOUD-CLIENT/IAM.PY)

FEEDBACK (#)

```
client = pubsub_v1.PublisherClient()
topic_path = client.topic_path(project, topic_name)

policy = client.get_iam_policy(topic_path)

print("Policy for topic {}:".format(topic_path))
for binding in policy.bindings:
    print("Role: {}, Members: {}".format(binding.role, binding.members))
```

Setting a policy

The `setIamPolicy()` method lets you attach a policy (https://cloud.google.com/iam/docs/managing-policies) to a resource. The `setIamPolicy()` method takes a `SetIamPolicyRequest`, which contains the policy to be set and the resource to which the policy is attached. It returns the resulting policy.

Here is some sample code to set a policy for a subscription (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.subscriptions/setIamPolicy):

C# PYTHON MORE

Before trying this sample, follow the Python setup instructions in Quickstart: Using Client Libraries (https://cloud.google.com/pubsub/docs/quickstart-client-libraries). For more information, see the Pub/Sub Python API reference documentation (https://googleapis.github.io/google-cloud-python/latest/pubsub/).

GOOGLECLOUDPLATFORM/PYTHON-DOCS-SAMPLES/BLOB/MASTER/PUBSUB/CLOUD-CLIENT/IAM.PY)

FEEDBACK (#)

```
client = pubsub_v1.SubscriberClient()
subscription_path = client.subscription_path(project, subscription_name)

policy = client.get_iam_policy(subscription_path)

# Add all users as viewers.
policy.bindings.add(role="roles/pubsub.viewer", members=["allUsers"])

# Add a group as an editor.
policy.bindings.add(
    role="roles/editor", members=["group:cloud-logs@google.com"]
)

# Set the policy
policy = client.set_iam_policy(subscription_path, policy)

print(
    "IAM policy for subscription {} set: {}".format(
        subscription_name, policy
    )
)
```

Here is some sample code to set a policy for a topic (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/setIamPolicy):

C# PYTHON MORE

Before trying this sample, follow the Python setup instructions in Quickstart: Using Client Libraries (https://cloud.google.com/pubsub/docs/quickstart-client-libraries). For more information, see the Pub/Sub Python API reference documentation (https://googleapis.github.io/google-cloud-python/latest/pubsub/).

GOOGLECLOUDPLATFORM/PYTHON-DOCS-SAMPLES/BLOB/MASTER/PUBSUB/CLOUD-CLIENT/IAM.PY)

FEEDBACK (#)

```
client = pubsub_v1.PublisherClient()
topic_path = client.topic_path(project, topic_name)

policy = client.get_iam_policy(topic_path)

# Add all users as viewers.
policy.bindings.add(role="roles/pubsub.viewer", members=["allUsers"])

# Add a group as a publisher.
policy.bindings.add(
    role="roles/pubsub.publisher", members=["group:cloud-logs@google.com"]
)

# Set the policy
policy = client.set_iam_policy(topic_path, policy)

print("IAM policy for topic {} set: {}".format(topic_name, policy))
```

Testing permissions

You can use the `testIamPermissions()` method to check which of the given permissions the caller has for the given resource. It takes as parameters a resource name and a set of permissions, and returns the caller's subset of permissions.

Here is some sample code to test permissions for a subscription (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/testIamPermissions):

C# PYTHON MORE

Before trying this sample, follow the Python setup instructions in Quickstart: Using Client Libraries (https://cloud.google.com/pubsub/docs/quickstart-client-libraries). For more information, see the Pub/Sub Python API reference documentation (https://googleapis.github.io/google-cloud-python/latest/pubsub/).

GOOGLECLOUDPLATFORM/PYTHON-DOCS-SAMPLES/BLOB/MASTER/PUBSUB/CLOUD-CLIENT/IAM.PY)

FEEDBACK (#)

```
client = pubsub_v1.SubscriberClient()
subscription_path = client.subscription_path(project, subscription_name)

permissions_to_check = [
    "pubsub.subscriptions.consume",
    "pubsub.subscriptions.update",
]

allowed_permissions = client.test_iam_permissions(
    subscription_path, permissions_to_check
)

print(
    "Allowed permissions for subscription {}: {}".format(
        subscription_path, allowed_permissions
    )
)
```

Here is some sample code to test permissions for a topic (https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.topics/testIamPermissions):

C# PYTHON MORE

Before trying this sample, follow the Python setup instructions in Quickstart: Using Client Libraries (https://cloud.google.com/pubsub/docs/quickstart-client-libraries). For more information, see the Pub/Sub Python API reference documentation (https://googleapis.github.io/google-cloud-python/latest/pubsub/).

GOOGLECLOUDPLATFORM/PYTHON-DOCS-SAMPLES/BLOB/MASTER/PUBSUB/CLOUD-CLIENT/IAM.PY)

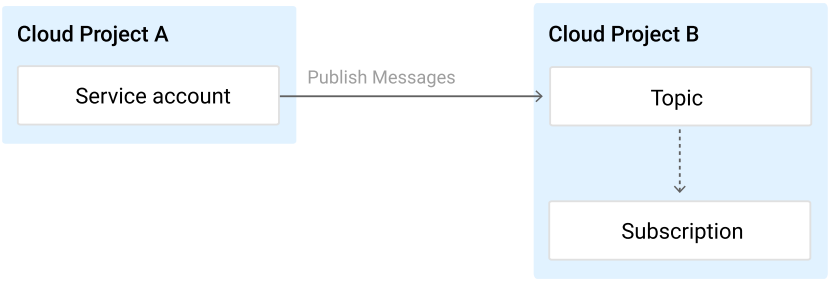
FEEDBACK (#)

```
client = pubsub_v1.PublisherClient()
topic_path = client.topic_path(project, topic_name)

permissions_to_check = ["pubsub.topics.publish", "pubsub.topics.update"]

allowed_permissions = client.test_iam_permissions(
```

```
    topic_path, permissions_to_check
)

print(
    "Allowed permissions for topic {}: {}".format(
        topic_path, allowed_permissions
    )
)
```

Sample use case: cross-project communication

Pub/Sub Cloud Identity and Access Management is useful for fine-tuning access in cross-project communication. For example, suppose a service account in Cloud Project A wants to publish messages to a topic in Cloud Project B. You could accomplish this by granting the service account Edit permission in Cloud Project B. However, this approach is often too coarse. You can use the Cloud IAM API to achieve a more fine-grained level of access.



For example, this snippet uses the `setIamPolicy()` method in ***project-b*** and a prepared `topic_policy.json` file to grant the service account `foobar@`***`project-a`***`.iam.gserviceaccount.com` of ***project-a*** the publisher role on the topic `projects/`***`project-b`***`/topics/`***`topic-b`***:

```
gcloud pubsub topics set-iam-policy \
    projects/project-b/topics/topic-b \
    topic_policy.json
```

Output:

```
Updated Cloud IAM policy for topic topic-b.
bindings:
- members:
  - serviceAccount:foobar@project-a.iam.gserviceaccount.com
  role: roles/pubsub.publisher
etag: BwWGrQYX6R4=
```

Partial availability behavior

Authorization checks depend on the Cloud IAM subsystem. In order to offer consistently low response latency for data operations (publishing and message consumption), the system may fall back on cached Cloud IAM policies. For information about when your changes will take effect, see the Cloud IAM documentation (https://cloud.google.com/iam).

Last updated January 13, 2020.