Revoking Access to Google Cloud Platform

This document will cover best practices for revoking a person's access to a Google Cloud Platform project. There are two sections to this document. The first covers setting up a project in a way that makes access revocation easy. The second contains step-by-step procedures for revoking a person's access to various types of resources.

Background

A critical time in the life of your Google Cloud Platform resources is when you need to remove someone’s access to them. When an employee leaves your company, your engagement with a contractor ends, or a collaborator moves on to other projects, there are a few things you should do to thoroughly revoke unneeded access to your resources.

Some of these processes are optional. You should determine which of these steps to execute depending on your security needs, products in use, and trust in the person whose access is being revoked. Use this document to come up with policies and procedures that make sense for you and your use of GCP.

Setting up your project

You can improve your project's ability to efficiently and securely revoke user access by making thoughtful choices at setup time.

Limit access to VMs

Virtual machines, like those used by Google Compute Engine (https://cloud.google.com/compute/), Google Kubernetes Engine (https://cloud.google.com/kubernetes-engine/), and App Engine Flexible Environment (https://cloud.google.com/appengine/docs/flexible/), are large potential attack surfaces. If someone has ever had access to a VM, especially root or Administrator access, it’s extremely difficult to guarantee that they haven’t modified the VM and left a backdoor to allow themselves future access. Limit VM access to those people who have a clear and specific need for it. Note that by default, project editors and owners have administrative access to all VMs in the project.

Before granting login access to an individual, think about what they need to do that requires it, and figure out other ways to meet those needs if possible. Instead of giving every developer login access in order to deploy code, consider using tools like Chef (https://www.chef.io/), Puppet (https://puppet.com/), and Salt (https://saltstack.com/) to manage your deployments.

Prepare for credential rotation

You should design your projects and resources to allow for easy and non-disruptive rotation of project-level credentials. These are secrets tied to the project itself, like service account keys, OAuth client secrets, and application-specific secrets, like database root passwords. Plan for this now, and make it easy to deploy new credentials in any applications that need them.

Restrict API keys

When creating and managing API keys, restrict the set of web sites, IP addresses, and apps that can use them. API keys are visible to all project members, so any unrestricted keys need to be rotated or deleted in order to revoke billing access. Read more about best practices for securely using API keys (https://support.google.com/cloud/answer/6310037?hl=en) and plan your usage accordingly.

Revoking access

If you've made good choices in project setup, the following processes will be an efficient and secure way to revoke a person's access.

Remove the account from project membership

1. In the Google Cloud Platform console, go to the IAM permissions page.

   IAM PERMISSIONS (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/IAM-ADMIN)

2. Select the project you want to remove an account from.
3. Click the checkbox next to the row containing the account you want removed from the member list, then click **Remove**. Alternatively, click the trash can icon next to the account you want to remove.

Rotate project credentials

Note: If you have applications and services depending on project-level secrets, be sure to update them after rotating credentials.

Service account keys

Service accounts are non-user accounts that are, by default, considered editors of their associated project. Someone with the ability to act as a service account can do anything a project editor can do.

Only project owners can create new service accounts or keys for existing service accounts. If the person whose access is being revoked was a project owner, you need to rotate any existing service account keys. If the person was not a project owner, you may skip this step. However, consider if the person may have had access to service account keys somewhere outside of secure Google Cloud Platform tools, such as your source code repository or application configurations.

1. In the Google Cloud Platform console, go to the API credentials page.

   API CREDENTIALS (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/APIS/CREDENTIALS)
2. Click **Create credentials**, then select **Service account key**.
3. Select the target account from the **Service account** menu.
4. Choose the **Key type** you want to create. In most situations, **JSON** is recommended, but **P12** is available for backwards compatibility with code that depends on it.
5. Click **Create**. A file containing the new key will be automatically downloaded through your browser. Deploy this key to any applications that need it.
6. After confirming the new key works as expected, return to the credentials page and delete the old key associated with that service account.

OAuth client ID secrets

OAuth client ID secrets don't provide any direct access to your project. However, if an attacker is able to steal the OAuth refresh tokens provided by Google on behalf of your application's users, possession of the client ID secret would allow the attacker to access your users' Google accounts within the same scopes originally requested by your application.

OAuth client ID secrets are viewable by all project owners and editors, but not readers. If the person whose access is being revoked was not an owner or editor, you may skip this step. However, consider if the person may have had access to client ID secrets somewhere outside of secure Google Cloud Platform tools, such as your source code repository or application configurations.

1. In the Google Cloud Platform console, go to the API credentials page.

   API CREDENTIALS (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/APIS/CREDENTIALS)

2. Click the name of the OAuth 2.0 client ID you want to modify. This will open a Client ID page with details on the selected ID.
3. On the Client ID page, click **Reset secret**.
4. Click **Reset** in the confirmation dialog to immediately revoke the old secret and set a new one. Note that any active users will need to reauthenticate upon their next request.
5. Deploy the new secret to any applications that need it.

API keys

API keys don't provide access to your project or your users' data, but they control who Google bills for API requests. Any project member can see your project's API keys. If you have any unrestricted keys, you need to delete or regenerate them when revoking someone's access to your project.

1. In the Google Cloud Platform console, go to the API credentials page.

   API CREDENTIALS (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/APIS/CREDENTIALS)

2. Click **Create credentials**, then select **API key**.
3. A dialog will display the newly created key. Deploy this key to any applications using the key you wish to replace.
4. After confirming that your applications are working as expected with the new key, return to the credentials page and delete the old unrestricted key.

Revoke access to VMs

If the person whose access you are revoking did not have login access to any of your project VMs, you may skip this step.

Note: If someone has ever had root access to a VM, it is not possible to guarantee that their access has been terminated by removing their associated user account. That person may be able to log in as a different user, use personal public keys to access other accounts, or run processes as other users. The most secure procedure is to recreate VMs from a known good image.

1. Remove all project-level SSH keys (https://cloud.google.com/compute/docs/instances/adding-removing-ssh-keys#project-wide) that the person had access to.
2. On each VM where the person had SSH access, remove any instance-level keys (https://cloud.google.com/compute/docs/instances/adding-removing-ssh-keys#instance-only).
3. Remove the person's account from any VMs they had login access to.
4. Check for suspicious applications the person may have installed to provide backdoor access to the VM. If you are uncertain about the security of any code running on the VM, recreate it and redeploy the applications you need from source.
5. Verify that the VM firewall settings have not been changed from your planned or expected configuration.
6. If you create new VMs from custom base images, verify that the base images have not been modified in a way that would compromise the security of new VMs.

Revoke access to Cloud SQL databases

If your project does not use any Cloud SQL resources, you may skip this step.

1. In the Google Cloud Platform console, go to the SQL instances page.

   SQL INSTANCES (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/SQL/INSTANCES)
2. Click the instance ID of the database you want to revoke access to.
3. Click **Access Control**. In this tab, confirm that the list of IP addresses under **Authorized networks** and list of apps under **App Engine authorization** match what you expect. If the person whose access you're trying to revoke has access to networks or applications listed here, they can access this database.
4. Click **Users**. In this tab, delete or change the password for any user accounts the person had access to. Be sure to update any applications that depend on those user accounts.

Redeploy App Engine

App Engine apps have access by default to a service account that is an editor on the associated project. App Engine request handlers can do things like create new VMs, and read or modify data in Cloud Storage. Someone with the ability to deploy code to App Engine could use this service account to open a backdoor into your project. If you're concerned about the code integrity of your deployed apps, you may want to redeploy them (including any modules) with a known-good checkout from your version control system.

Verify permissions on other resources

Consider other resources in your project that the person may have had access to, and verify that the permissions for those objects are secure. Resources to check include:

- Cloud Storage bucket Access Control Lists (https://cloud.google.com/storage/docs/access-control/create-manage-lists)
- BigQuery dataset permissions (https://cloud.google.com/bigquery/docs/access-control)
- Cloud Pub/Sub topic permissions (https://cloud.google.com/pubsub/docs/access_control)

Next steps

Read the Google Cloud Security Overview (https://cloud.google.com/security/).

Last updated December 4, 2019.