

[Private Catalog](https://cloud.google.com/private-catalog/) (<https://cloud.google.com/private-catalog/>)

[Documentation](https://cloud.google.com/private-catalog/docs/) (<https://cloud.google.com/private-catalog/docs/>) [Guides](#)

Overview of Private Catalog

Private Catalog enables developers and cloud admins to make their solutions discoverable to their own organization's internal enterprise users.

While making solutions discoverable, cloud admins can also control the distribution of solutions and ensure compliance and governance.

Private Catalog includes two APIs, the Private Catalog Producer API and the Private Catalog API. The Private Catalog Producer API allows cloud admins to build and manage catalogs. The Private Catalog API allows users to discover catalogs and the products contained in them.

Features

Private Catalog includes the following features:

- A Private Catalog experience for users and cloud admins
- APIs for cloud admins
- Ability to create multiple catalogs and share them at different folder and project levels
- Ability to add additional Private Catalog cloud admins to create and curate content
- Support for reference links and Cloud Deployment Manager templates
- Single point of entry for cloud admins and end users
- Respect for organizational policies and Cloud Deployment Manager constraints to complement Private Catalog and provide controls

A cloud admin can use the Private Catalog Producer API to create a catalog under an organization. The catalog contains a cloud admin-curated list of trustworthy products. Cloud admins can then use the same API to share the catalog to the whole organization. All other users in the same organization can see the shared catalog and its products if they have permission.

Private Catalog can list solutions for internal enterprise users to discover and deploy. After cloud admins create a catalog, cloud admins can curate and update the content and define governance.

Why use Private Catalog?

This section describes a typical use case for Private Catalog.

The situation

A company called ACME Game Studios has over a thousand employees, half of whom are developers. They own a large share of the mobile game market and develop for multiple platforms. ACME's IT department creates and maintains images for their development environment.

- They have two virtual images for each platform that they maintain, one for development and one for testing.
- The development environment contains all the tools and connections to their internal source repository, along with build tools and the relevant platform SDKs.
- The test environment contains a virtualized environment of the targeted platform.
- While Acme's IT department has a process to build and update their images, they don't have a good way to distribute them.

Andrea

Andrea, the cloud admin, is the person who installs images and tools and makes them available. She manages about 20 company-owned virtual images. Once or twice a week, she updates about half of those images because of security issues, new SDK tools, or internal build tool updates.

Every time she updates an image, she sends out an email to all the engineers. Andrea generally receives 20-30 support tickets per week asking: "What's the latest image?"

Recently, Andrea launched a wiki, which she updates with a list of the latest tools and where to find them. This helped with the support tickets, but when Andrea went on vacation, the other admins forgot to update the wiki, and devs launched unpatched images for a couple of weeks.

To help fix that, Andrea added a freshness meter to the wiki and emailed the developers that if the wiki freshness meter isn't green, they should contact support. This caused the number of support tickets to return to 20-30 per week.

Darryl

Darryl's problem is that he doesn't know which tools to use because he receives numerous emails every week telling him about new updates, new tools, new images. He knows he's supposed to shut down his images when bugs or features are complete, but he just keeps them running instead of trying to find out which tools he needs. Sometimes he causes a build break and then he finds the newest tools, as needed.

How Private Catalog helps Andrea

Let's say Andrea wants her engineering team to use a known, free virtual image instead of any other paid software tools for building games. Recently, she's noticed several unauthorized instances being launched and some surprising charges on Acme's GCP bill.

So she wants to run Private Catalog specifically for her Eng team to control what they can launch from GCP.

From the left nav on the GCP Console, Andrea can go to Private Catalog and navigate to the admin console.

She can create a new catalog called "Dev Tools" with a description of "Tools for developing mobile games."

Inside the new catalog, she adds the first solution. She can see there are two supported solution types: one for [Cloud Deployment Manager templates](https://cloud.google.com/deployment-manager/templates) (<https://cloud.google.com/deployment-manager/docs/step-by-step-guide/create-a-template>) and the other for reference links.

Reference links are links to anything on the web that Andrea has verified and curated, links to help documentation, or anything else Andrea wants to link to.

Andrea adds a schema file to define where her engineers can deploy the solution (region) and the machine type to ensure her team doesn't exceed the budget.

Now that she's added her first solution, she shares it with her test project.

She knows that Darryl is not a fan of IT's current system, so recruiting him as a test subject is easy. Andrea shares her project with Darryl. He now has permission to use the new catalog.

When Andrea needs to update a solution, she can create a new one to replace the existing solution. This way, she can be sure all developers deploy the latest version.

How Private Catalog helps Darryl

Darryl logs in, navigates to the project, and launches Private Catalog.

He sees a page that looks like Google Cloud Marketplace. He can see the solution Andrea created.

Everything looks good, so he tries to launch it after selecting a region and CPU type.

Darryl can go to a central place to find the software he needs and deploy it all from there.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated December 4, 2019.