

# Google Container Registry Auth Plugin

## Plugin Information

View Google Container Registry Auth [on the plugin site](#) for more information.

## Google Container Registry Auth Plugin

This plugin allows the credential provider to use Google Cloud Platform OAuth Credentials (provided by the [Google OAuth Credentials plugin](#)) to access Docker images from [Google Container Registry \(GCR\)](#).

## Introduction

This plugin supports both kinds of credentials provided by [Google OAuth Credentials plugin](#) : **Google Service Account from metadata** as well as **Google Service Account from private key**.

As of this writing, it has been verified that this plugin can be used together with [Docker Build Step Plugin](#) and [CloudBees Docker Custom Build Environment Plugin](#). These plugins will be able to retrieve the credential provided by this plugin, and then use it to authenticate against GCR to pull/push Docker images. Other plugins that rely on [credentials provider](#) or [Docker Commons Plugin](#) (preferred than Credentials Plugin) to provide credentials will also be able to utilize this credential provider plugin to pull images from or push images to GCR.

- The credential will need the scope [https://www.googleapis.com/auth/devstorage.read\\_write](https://www.googleapis.com/auth/devstorage.read_write) or [https://www.googleapis.com/auth/devstorage.full\\_control](https://www.googleapis.com/auth/devstorage.full_control). The full\_control scope is an overkill but it works with this plugin.
- Your service account will need to have access to the Google Cloud Platform project for which you want Google Container Registry to host Docker images for.

## Usage

The instruction below uses [Docker Build Step Plugin](#) as an example to show how this plugin can be used to provide credentials for the former plugin to authenticate against GCR to perform pull/push.

- Install [Docker Build Step Plugin](#), and configure it accordingly. Note that do not forget to configure its global configurations, which includes the Docker server's address.
- Configure your OAuth credentials per instructions from [Google OAuth Plugin](#), using the service account that has read/write access to your Google Container Registry.
- Note that the credentials provided by this plugin will not show up in the drop-down list when you add credentials. It merely wraps a Google Oauth credential.
- Install this plugin, then on Jenkins' global configuration page, under "Google Container Registry", set the correct Google Container Registry server address. By default, it is "gcr.io,\*gcr.io" (Do not include schemes such as "https://").
- In your Jenkins job, add a build step "Execute Docker Container", and choose either "pull image" or "Push image" as your docker command (other docker commands don't require credentials so they are not relevant to this plugin). Enter image name, tag and registry. In the "Docker registry URL" field, by default you should enter "https://gcr.io". The value in this field should match the value in "Google Container Registry" Server Address field in global configuration, but with the scheme (such as https://) added. Its exact value (the part after "https://") should be decided by the registry of your docker image.
- In the "Docker Credential" dropdown, select your account marked as "Google Container Registry Account".
- Save your configuration and run your job.

## Version History

### Version 0.1 (June 11, 2015)

- Initial release
- **Version 0.2 (July 14, 2015)**
- Hide this credential from "Add Credentials" drop-down list. This credentials merely wraps around Google Oauth credential, you should add the corresponding Oauth credential in "Add Credentials".
- Integrate with Docker Commons Plugin