Serverless Computing (https://cloud.google.com/products/serverless/)
Cloud Run: Serverless Computing (https://cloud.google.com/run/)
Documentation (https://cloud.google.com/run/docs/) Guides

Continuous Deployment from git using Cloud Build

Beta

This feature is in a pre-release state and might change or have limited support. For more information, see the product launch stages (https://cloud.google.com/products/#product-launch-stages).

You can use Cloud Build to automate builds and deployments to Cloud Run.

You can accomplish this using a Cloud Build trigger (https://cloud.google.com/cloud-build/docs/running-builds/automate-builds) to automatically build and deploy your code whenever new commits are pushed to a given branch of a git repository. This includes Cloud Source Repositories, GitHub, or any other repository supported by Cloud Build.

The build trigger does the following:

- Builds the Docker Image
- Pushes the image to the Container Registry
- Deploys a new revision to the Cloud Run service

Setting up continuous deployment with Cloud Build

To automate deployment with Cloud Build:

1. In your repository root, add a file named `cloudbuild.yaml` that has these entries:

FULLY MANAGED   ANTHOS ON GOOGLE ...

```
steps:
# build the container image
- name: 'gcr.io/cloud-builders/docker'
  args: ['build', '-t', 'gcr.io/$PROJECT_ID/[SERVICE-NAME]:$COMMIT_SHA', '.']
```

```
# push the container image to Container Registry
- name: 'gcr.io/cloud-builders/docker'
  args: ['push', 'gcr.io/$PROJECT_ID/[SERVICE-NAME]:$COMMIT_SHA']
# Deploy container image to Cloud Run
- name: 'gcr.io/cloud-builders/gcloud'
  args:
  - 'run'
  - 'deploy'
  - '[SERVICE-NAME]'
  - '--image'
  - 'gcr.io/$PROJECT_ID/[SERVICE-NAME]:$COMMIT_SHA'
  - '--region'
  - '[REGION]'
  - '--platform'
  - 'managed'
images:
- 'gcr.io/$PROJECT_ID/[SERVICE-NAME]:$COMMIT_SHA'
```

Replace

- ***[SERVICE-NAME]*** with the name of the Cloud Run service.
- ***[REGION]*** with the region of the Cloud Run service you are deploying.

The use of the `$COMMIT_SHA` substitution variable is populated by Cloud Build when triggered from a git repository. To test this configuration manually override the variable:

```
gcloud builds submit --substitutions COMMIT_SHA=manual
```

2. Grant access for the Cloud Build service account to deploy the service:

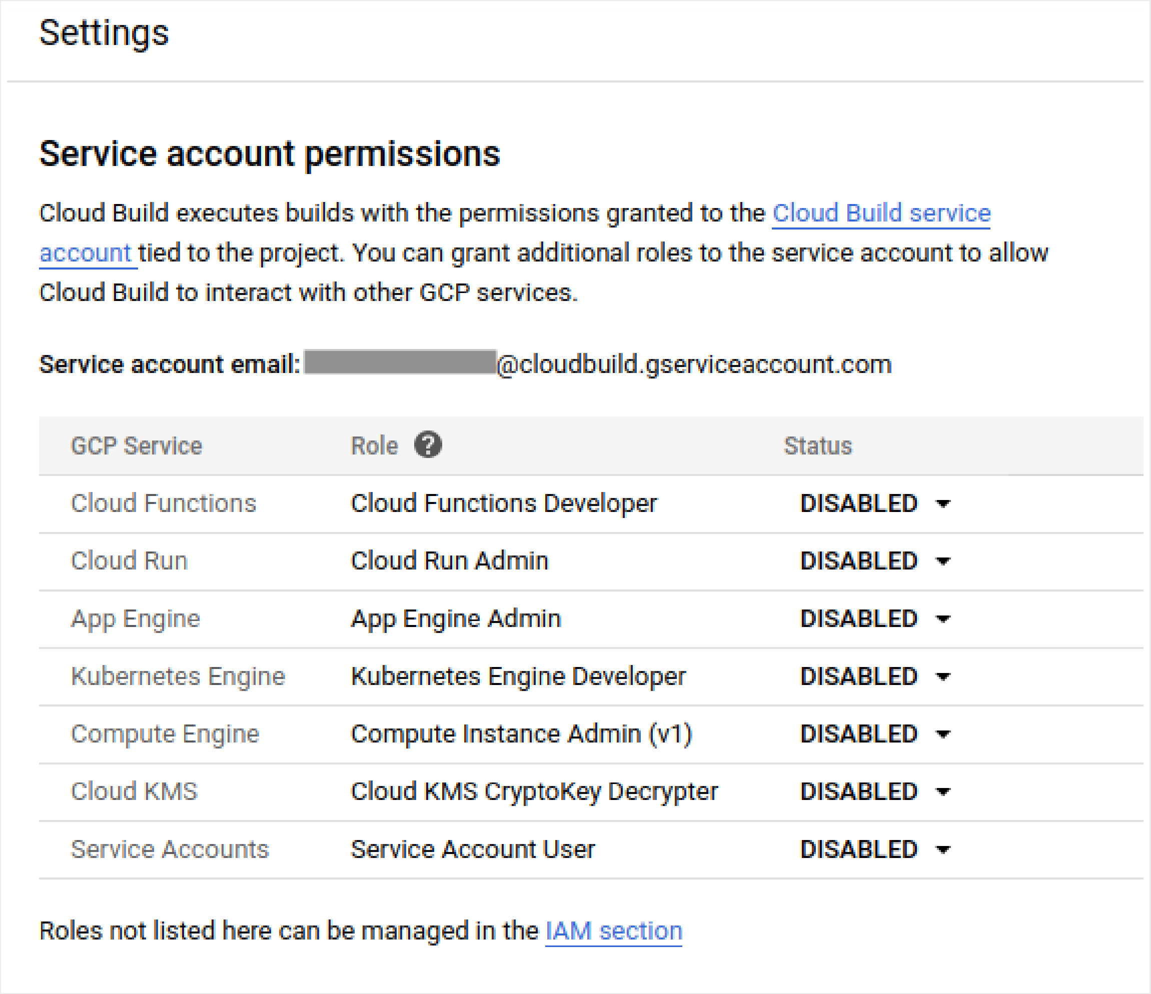
FULLY MANAGED ANTHOS ON GOOGLE ...

Grant the *Cloud Run Admin* and *Service Account User* roles to the Cloud Build service account:

a. Open the Cloud Build settings page in the Cloud Console:

VISIT THE CLOUD BUILD SETTINGS PAGE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/CLOUD-

b. In the Service account permissions panel, set the status of the *Cloud Run Admin* role to **Enable**:



Settings

Service account permissions

Cloud Build executes builds with the permissions granted to the Cloud Build service account tied to the project. You can grant additional roles to the service account to allow Cloud Build to interact with other GCP services.

Service account email: [redacted]@cloudbuild.gserviceaccount.com

GCP Service	Role	Status
Cloud Functions	Cloud Functions Developer	DISABLED ▾
Cloud Run	Cloud Run Admin	DISABLED ▾
App Engine	App Engine Admin	DISABLED ▾
Kubernetes Engine	Kubernetes Engine Developer	DISABLED ▾
Compute Engine	Compute Instance Admin (v1)	DISABLED ▾
Cloud KMS	Cloud KMS CryptoKey Decrypter	DISABLED ▾
Service Accounts	Service Account User	DISABLED ▾

Roles not listed here can be managed in the IAM section

c. Select **GRANT ACCESS TO ALL SERVICE ACCOUNTS** to grant the **Service Account User** role on all service accounts in the project on your page.

★ **Note:** For stronger security, only allow Cloud Build to act as a specific Cloud Run service (#continuous-iam).

3. Click **Triggers** in the left navigation panel to open the *Triggers* page:

VISIT THE TRIGGERS PAGE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/CLOUD-BUILD/TRIGGERS)

a. Click **Create Trigger**.

b. select your repository from the displayed repository list, and click **Continue**.

For more information on specifying which branches to autobuild, see Creating a build trigger (https://cloud.google.com/cloud-build/docs/running-builds/automate-builds#build_trigger).

c. Select `cloudbuild.yaml` in *Build Configuration*.

d. Click **Create**.

4. You are finished! From now on, whenever you push to your repository, a build and a deployment to your Cloud Run service is automatically triggered.

Note: To make the deployed service public, make a one-time change to the service IAM settings (https://cloud.google.com/run/docs/securing/managing-access#making_a_service_public).

Continuous deployment with minimal IAM permissions

When a container is deployed to a Cloud Run (fully managed) service, it runs with the identity of the Runtime Service Account of this Cloud Run (fully managed) service. Because Cloud Build can deploy new containers automatically, Cloud Build needs to be able to *act as* the Runtime Service Account of your Cloud Run service.

To grant limited access to Cloud Build to deploy to a Cloud Run service running as the default compute identity:

CONSOLE UI GCLOUD

1. Go to the **Service accounts** page of the Google Cloud Console:

   GO TO SERVICE ACCOUNTS (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/IAM-ADMIN/SERVICEACCO

2. Select the Runtime Service Account (`PROJECT_NUMBER-compute@developer.gserviceaccount.com`) from the table.
3. Click **Show Info Panel** in the top right corner to show the **Permissions** tab.
4. Click the **Add member** button.
5. Enter the Cloud Build Service Account (`PROJECT_NUMBER@cloudbuild.gserviceaccount.com`)
6. In the **Select a role** dropdown, select the **Service Accounts** > **Service Account User** role.
7. Click **Save**.

If using Cloud Run (fully managed) using a customized compute identity (https://cloud.google.com/run/docs/securing/service-identity#per-service-identity), replace `PROJECT_NUMBER-compute@developer.gserviceaccount.com` with your service account address.

See Deployment permissions (https://cloud.google.com/run/docs/reference/iam/roles#additional-configuration) for more information.

What's Next

- Learn how deploy or publish a container image to a private registry in another project in Setting service account permissions (https://cloud.google.com/cloud-build/docs/securing-builds/set-service-account-permissions)

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/)*, and code samples are licensed under the Apache 2.0 License* (https://www.apache.org/licenses/LICENSE-2.0)*. For details, see our Site Policies* (https://developers.google.com/terms/site-policies)*. Java is a registered trademark of Oracle and/or its affiliates.*

Last updated January 16, 2020.