Deploying to Cloud Run

This page describes how to deploy container images to a new Cloud Run service or to a new revision of an existing Cloud Run service.

Before you start

If you are under a domain restriction organization policy restricting (/run/docs/authenticating/public#domain-restricted-sharing) unauthenticated invocations for your project, you will need to access your deployed service as described under Testing private services (/run/docs/triggering/https-request#testijng-private).

Permissions required to deploy

You must have ONE of the following:

- *Owner*
- *Editor*
- Both the *Cloud Run Admin* and *Service Account User* roles
- Any custom role that includes this specific list of permissions (/run/docs/reference/iam/roles#additional-configuration)

Supported container registries and images

You can use container images stored in Artifact Registry (/artifact-registry/docs/overview), Container Registry (/container-registry), or Docker Hub (https://hub.docker.com/). Google recommends the use of Artifact Registry.

You can use only the following types of container images:

- Container images stored in the same project as the one you are creating the job or service in.

- Container images from other Google Cloud projects (provided that the correct IAM permissions are set (/run/docs/deploying#other-projects)).
- Public container images (/container-registry/docs/access-control#serving_images_publicly) from Artifact Registry, Container Registry, or Docker Hub.

If you are storing container images in another type of container registry, follow the instructions under deploying images from unsupported registries (/run/docs/deploying#other-registries).

Deploying a new service

You can specify a container image with a tag (for example, `us-docker.pkg.dev/my-project/container/my-image:latest`) or with an exact digest (for example, `us-docker.pkg.dev/my-project/container/my-image@sha256:41f34ab970ee...`).

Deploying to a service for the first time creates its first revision. Note that revisions are immutable. If you deploy from a container image tag, it will be resolved to a digest and the revision will always serve this particular digest.

You can deploy a container using the Google Cloud console, the `gcloud` command line or from a YAML configuration file.

Click the tab for instructions using the tool of your choice.

Console (#console) Command line (#command-line) YAML (#yaml) Cloud Code (#cloud-code) Terraform (#terraform) Client lib

1. In one of the following development environments, set up the gcloud CLI:
   - **Cloud Shell**: to use an online terminal with the gcloud CLI already set up, activate Cloud Shell.

     Activate Cloud Shell on this page

     At the bottom of this page, a Cloud Shell session starts and displays a command-line prompt. It can take a few seconds for the session to initialize.
   - **Local shell**: to use a local development environment, install (/sdk/docs/install) and initialize (/sdk/docs/initializing) the gcloud CLI.
2. To deploy a container image:

a. Run the command:

```
gcloud run deploy SERVICE --image IMAGE_URL
```

- Replace ***SERVICE*** with the name of the service you want to deploy to. Service names must be 49 characters or less and must be unique per region and project. If the service does not exist yet, this command creates the service during the deployment. You can omit this parameter entirely, but you will be prompted for the service name if you omit it.
- Replace ***IMAGE_URL*** with a reference to the container image, for example, `us-docker.pkg.dev/cloudrun/container/hello:latest`. Note that if you don't supply the `--image` flag, the deploy command will attempt to deploy from source code (/run/docs/deploying-source-code).

If you are creating a public API or website, you can allow unauthenticated invocations of your service using the `--allow-unauthenticated` flag. This assigns the **Cloud Run Invoker** IAM role (/run/docs/securing/managing-access#making_a_service_public) to `allUsers`. You can also specify `--no-allow-unauthenticated` to not allow unauthenticated invocations. If you omit either of these flags, you are prompted to confirm when the `deploy` command runs.

b. Wait for the deployment to finish. Upon successful completion, a success message is displayed along with the URL of the deployed service.

Note that to deploy to a different location from the one you set via the `run/region` `gcloud` properties, use:

- `gcloud run deploy SERVICE --region REGION`

When deploying, the Cloud Run service agent (/iam/docs/service-agents) needs to be able to access the deployed container, which is the case by default.

Each service has a unique and permanent URL that will not change over time as you deploy new revisions to it.

Deploying a new revision of an existing service

You can deploy a new revision using the Google Cloud console, the `gcloud` command line, or a YAML configuration file.

Note that changing any configuration settings results in the creation of a new revision, even if there is no change to the container image. Each revision created is immutable.

Click the tab for instructions using the tool of your choice.

Console (#console) Command line (#command-line) YAML (#yaml) Cloud Code (#cloud-code) Terraform (#terraform)

1. In one of the following development environments, set up the gcloud CLI:
    - **Cloud Shell**: to use an online terminal with the gcloud CLI already set up, activate Cloud Shell.

      Activate Cloud Shell on this page

      At the bottom of this page, a Cloud Shell session starts and displays a command-line prompt. It can take a few seconds for the session to initialize.
    - **Local shell**: to use a local development environment, install (/sdk/docs/install) and initialize (/sdk/docs/initializing) the gcloud CLI.
2. To deploy a container image:

    a. Run the command:

    ```
    gcloud run deploy SERVICE --image IMAGE_URL
    ```

    - Replace ***SERVICE*** with the name of the service you are deploying to. You can omit this parameter entirely, but you will be prompted for the service name if you omit it.
    - Replace ***IMAGE_URL*** with a reference to the container image, for example, `us-docker.pkg.dev/cloudrun/container/hello:latest`.

The revision suffix is assigned automatically for new revisions. If you want to supply your own revision suffix, use the gcloud CLI parameter --revision-suffix (/sdk/gcloud/reference/run/deploy#--revision-suffix).

b. Wait for the deployment to finish. Upon successful completion, a success message is displayed along with the URL of the deployed service.

Deploying images from other Google Cloud projects

You can deploy container images from other Google Cloud projects if you set the correct IAM permissions:

1. In the Google Cloud console, open the project for your Cloud Run service.

   Go to the IAM page (https://console.cloud.google.com/iam-admin/iam)

2. Select **Include Google-provided role grants**.
3. Copy the email of the Cloud Run service agent (/iam/docs/service-agents). It has the suffix **@serverless-robot-prod.iam.gserviceaccount.com**
4. Open the project that owns the container registry you want to use.

   Go to the IAM page (https://console.cloud.google.com/iam-admin/iam)

5. Click **Add** to add a new principal.
6. In the **New principals** text box, paste in the email of the service account that you copied earlier.
7. In the *Select a role* dropdown list, if you are using Container Registry, select the role **Storage -> Storage Object Viewer**. If you are using Artifact Registry, select the role **Artifact Registry -> Artifact Registry Reader**.
8. Deploy the container image (#deploying_a_new_service) to the project that contains your Cloud Run service.

★ **Note:** For stronger security, grant access to only the Cloud Storage bucket or Artifact Registry repository that contains your container images (/artifact-registry/docs/access-control#gcp).

Deploying images from unsupported registries

If you are storing container images in an unsupported public or private container registry, you can temporarily push them to Artifact Registry (/artifact-registry/docs/overview) using `docker push` in order to deploy them to Cloud Run. The container image is imported by Cloud Run when deployed, so after the deployment, you can delete the image from Artifact Registry (/artifact-registry/docs/docker/manage-images#deleting_images).

What's next

After you deploy a new service, you can do the following:

- Gradual rollouts, rollback revisions, traffic migration (/run/docs/rollouts-rollbacks-traffic-migration)
- View service logs (/run/docs/logging)
- Monitor service performances (/run/docs/monitoring)
- Set memory limits (/run/docs/configuring/memory-limits)
- Set environment variables (/run/docs/configuring/environment-variables)
- Change service concurrency (/run/docs/configuring/concurrency)
- Manage the service (/run/docs/managing/services)
- Manage service revisions (/run/docs/managing/revisions)
- Deploy only trusted images with Binary Authorization (/binary-authorization/docs/run/enabling-binauthz-cloud-run) (Preview (/products#product-launch-stages))

You can automate the builds and deployments of your Cloud Run services using Cloud Build Triggers:

- Set up Continuous Deployment (/run/docs/continuous-deployment)

You can also use Google Cloud Deploy to set up a continuous-delivery pipeline to deploy Cloud Run services to multiple environments:

- Deploy an app to Cloud Run using Google Cloud Deploy (/deploy/docs/deploy-app-run)

Last updated 2023-04-05 UTC.