

[Serverless Computing](https://cloud.google.com/products/serverless/) (https://cloud.google.com/products/serverless/)

[Cloud Run: Serverless Computing](https://cloud.google.com/run/) (https://cloud.google.com/run/)

[Documentation](https://cloud.google.com/run/docs/) (https://cloud.google.com/run/docs/) [Guides](#)

Running services on a schedule

You can use Cloud Scheduler to securely trigger a Cloud Run service on a schedule. This is similar to using cron jobs.

Possible use cases include:

- Performing backups on a timed basis
- Performing recurrent administration tasks, such as:
 - Re-generating a sitemap
 - Deleting old data, content, configuration, or revisions
 - Synchronizing content between data systems
 - Processing daily email notifications
 - Verifying and reporting on access to downstream services
- Generating documents, such as bills

This page shows how to securely use Cloud Scheduler with Cloud Run in the same Google Cloud project.

Before you start

ENABLE THE CLOUD SCHEDULER API ([HTTP://CONSOLE.CLOUD.GOOGLE.COM/APIS/LIBRARY/CLOUDSCH](http://console.cloud.google.com/apis/library/cloudscheduler)) on the project you are using.

Creating and deploying your service

To create and deploy:

1. In your service, implement the job that you want to run on a schedule.

2. Note which request type your service is expecting to receive the job requests from, for example, GET or POST. When you create the scheduled job that invokes your service, you'll need to specify the HTTP method that matches this.
3. If you are using Cloud Run for Anthos on Google Cloud, you must verify the identity within the container. See the [IAP sample code](https://cloud.google.com/iap/docs/authentication-howto#authenticating_from_a_service_account) (https://cloud.google.com/iap/docs/authentication-howto#authenticating_from_a_service_account) that demonstrates this.
4. When you [deploy the service](https://cloud.google.com/run/docs/deploying) (<https://cloud.google.com/run/docs/deploying>) you are using with Cloud Scheduler, make sure you do NOT allow unauthenticated invocations.

Creating a service account for Cloud Scheduler

You need to create a service account to associate with Cloud Scheduler, and give that service account the permission to invoke your Cloud Run service. You can use an existing service account to represent Cloud Scheduler, or you can create a new one.

To create a service account and give it permission to invoke the Cloud Run service:

CONSOLE

COMMAND LINE

1. Visit the *Create service account key* page in the Cloud Console.

[CREATE SERVICE ACCOUNT PAGE](https://console.cloud.google.com/apis/credentials/) ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/APIS/CREDENTIALS/](https://console.cloud.google.com/apis/credentials/))

2. From the *Service account list*, select **New service account**.
3. In the *Service account name* field, enter the name you want to use for the service account.
4. Click **Create**.
5. Copy the service account email to use in the following steps.
6. Click **Continue** if prompted to specify permissions.
7. Visit the *Cloud Run Services* page in the Cloud Console.

[GO TO THE SERVICES PAGE](https://console.cloud.google.com/run/) ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/RUN](https://console.cloud.google.com/run/))

8. Select your service in the displayed list.
9. If necessary, click the **Show Info Panel/Hide Info Panel** toggle in the far right of the page to show information.
10. Locate the *Permissions* tab, and in that tab, click **Add Member**.


11. Paste your service account email into the **New members** field.
12. From the Role dropdown menu, select **Cloud Run > Cloud Run Invoker**.
13. Click **Save**.

Creating a Cloud Scheduler job

You'll need to create a job that invokes your service at specified times. You can use either the console or the command line:

To create a job you can use either the console or the gcloud command line. Click on the appropriate tab:

CONSOLE GCLOUD

1. Visit the Cloud Scheduler console page.
[GO TO CLOUD SCHEDULER \(HTTPS://CONSOLE.CLOUD.GOOGLE.COM/CLOUDSCHEDULER\)](https://console.cloud.google.com/cloudscheduler)
2. Click **Create job**.

3. Supply a name for the job.
4. Specify the frequency, or job interval, at which the job is to run, using a configuration string. For example, the string `0 */3 * * *` runs the job every 3 hours. The string you supply here can be any [crontab](http://man7.org/linux/man-pages/man5/crontab.5.html) (<http://man7.org/linux/man-pages/man5/crontab.5.html>) compatible string.
For more information, see [Configuring Job Schedules](https://cloud.google.com/scheduler/docs/configuring/cron-job-schedules) (<https://cloud.google.com/scheduler/docs/configuring/cron-job-schedules>).
5. From the dropdown list, choose the timezone to be used for the job frequency.
6. Specify HTTP as the target:
 - a. Specify the fully qualified URL of your service, for example `https://myservice-abcdef-uc.a.run.app`. The job will send requests to this URL.
 - b. Specify the HTTP method: the method must match what your previously deployed Cloud Run service is expecting. The default is `POST`.
 - c. Optionally, specify the data to be sent to the target. This data is sent in the body of the request when either the `POST` or `PUT` HTTP method is selected.

- d. Click **More** to show the auth settings.
 - e. From the dropdown menu, select **Add OIDC token**.
 - f. In the *Service account* field, copy the service account email of the service account you created previously (#create-service-account).
 - g. In the **Audience** field, copy the full URL of your service.
7. Click **Create** to create and save the job.

Your Cloud Run service will be triggered via a request by the Cloud Scheduler job at the frequency you defined. You can confirm and monitor this by examining the logs for this service (https://cloud.google.com/run/docs/logging#viewing_logs_in).

What's next

- [Logging and viewing logs](https://cloud.google.com/run/docs/logging) (<https://cloud.google.com/run/docs/logging>)
- [Monitoring health and performance](https://cloud.google.com/run/docs/monitoring) (<https://cloud.google.com/run/docs/monitoring>)
- [Triggering from Pub/Sub](https://cloud.google.com/run/docs/events/pubsub-push) (<https://cloud.google.com/run/docs/events/pubsub-push>)
- [Invoking with HTTPS](https://cloud.google.com/run/docs/events/https-request) (<https://cloud.google.com/run/docs/events/https-request>)

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.