

[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](#)

Using Environment Variables

When you set environment variables, they are injected into the container and are accessible to your code.

Like any configuration change, setting an environment variable leads to the creation of a new revision. Subsequent revisions will also automatically get this environment variable unless you make explicit updates to change it.

Valid variable name characters

You can use only alphanumeric characters and underscores in environment variable names.

Reserved names

The environment variables defined in the [container runtime contract](https://cloud.google.com/run/docs/reference/container-contract#env-vars) (https://cloud.google.com/run/docs/reference/container-contract#env-vars) are reserved and cannot be set. In particular, the `PORT` environment variable is injected inside your container by Cloud Run. You should not set it yourself.

Setting environment variables on a service

You can set environment variables using the Cloud Console or the `gcloud` command line when you [create a new service](https://cloud.google.com/run/docs/deploying#service) (https://cloud.google.com/run/docs/deploying#service) or [deploy a new revision](https://cloud.google.com/run/docs/deploying#revision) (https://cloud.google.com/run/docs/deploying#revision):

CONSOLE

COMMAND LINE

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

2. Click **CREATE SERVICE** if you are setting environment variables on a new service you are deploying to. If you are setting variables on an existing service, then click on the service, then click **DEPLOY NEW REVISION**.
3. Click **SHOW OPTIONAL SETTINGS**:

Name	Value
e.g. ENV	e.g. prod

+ ADD VARIABLE

- If you are adding a variable, click **Add Variable**, and specify the name you want for the variable and its value in the **Name** and **Value** text boxes.
 - If you are changing a value for a variable, replace the current value in the **Value** text box with the one you want.
 - If you are removing one or more environment variables, hover your cursor to the left of the **Value** textbox of the variable you are removing to display the Trash icon, and click it.
4. Click **Create** or **Deploy**.

Setting default environment variables in the container

You can use the **ENV** (<https://docs.docker.com/engine/reference/builder/#env>) statement in a Dockerfile to set default values for environment variables:

```
ENV KEY1=VALUE1, KEY2=VALUE2
```



Order of precedence: container vs service variables

If you set a default environment variable (`#default`) in the container and also set an environment variable (`#setting`) with the same name on the Cloud Run service, the value set on the service takes precedence.

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>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated December 4, 2019.