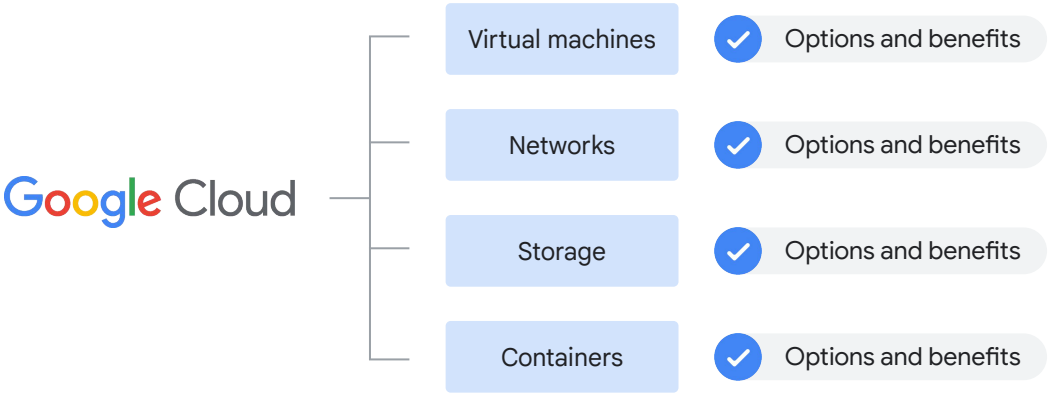
Applications in the Cloud

So far in this course ...



So far in this course, we've provided an introduction to Google Cloud and explored the options and benefits related to using virtual machines, networks, storage, and containers in the Cloud.

Applications in the Cloud

01 Cloud Run

02 Cloud Functions

In this final section of the course, we'll turn our attention to developing applications in the cloud by exploring Cloud Run and Cloud Functions.

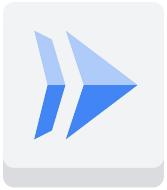
Applications in the Cloud

01 Cloud Run

02 Cloud Functions

Let’s begin with Cloud Run.

Cloud Run is managed serverless computing



- A managed compute platform that can run stateless containers
- Serverless, removing the need for infrastructure management
- Built on Knative, an open API and runtime environment built on Kubernetes
- Can automatically scale up and down from zero almost instantaneously, charging only for the resources used

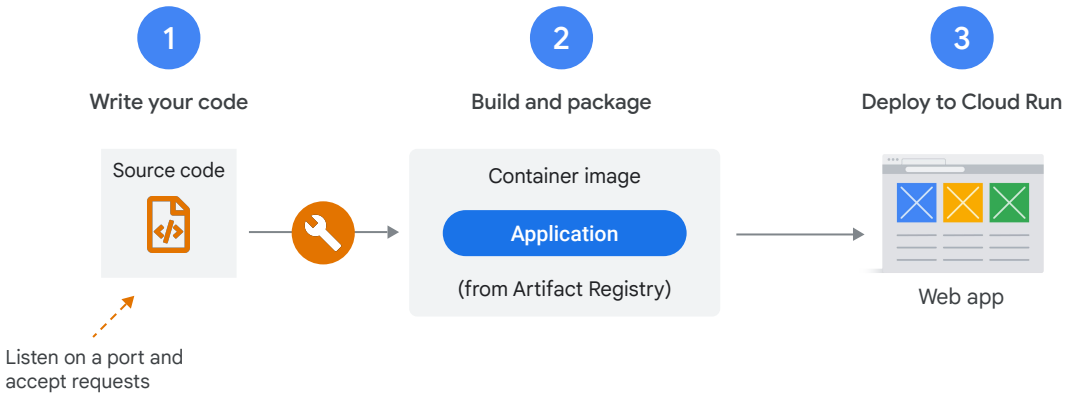
Cloud Run is a managed compute platform that lets you run stateless containers via web requests or Pub/Sub events.

Cloud Run is serverless. That means it removes all infrastructure management tasks so you can focus on developing applications.

It is built on Knative, an open API and runtime environment built on Kubernetes. It can be fully managed on Google Cloud, on Google Kubernetes Engine, or anywhere Knative runs.

Cloud Run is fast. It can automatically scale up and down from zero almost instantaneously, and it charges only for the resources used, calculated down to the nearest 100 milliseconds, so you'll never pay for over-provisioned resources.

Google Cloud Run workflow is a three-step process



The Cloud Run developer workflow is a straightforward three-step process.

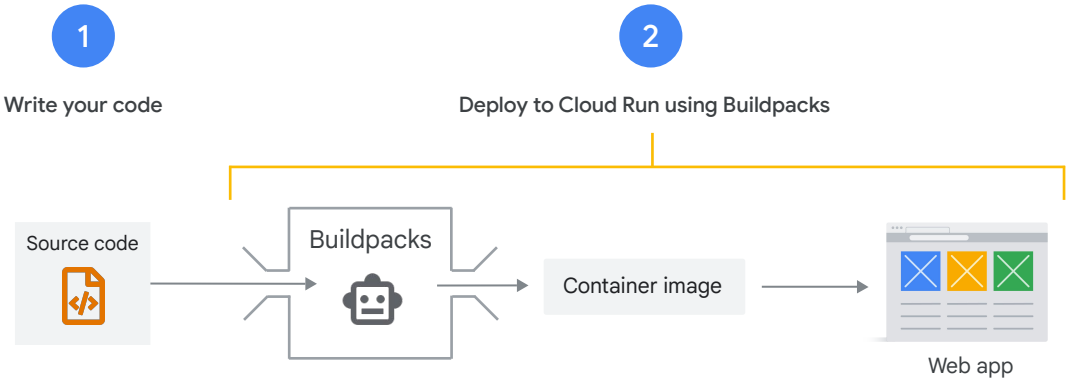
- First, you write your application using your favorite programming language. This application should start a server that listens for web requests.
- Second, you build and package your application into a container image.
- Third, the container image is pushed to Artifact Registry, where Cloud Run will deploy it.

Once you've deployed your container image, you'll get a unique HTTPS URL back.

Cloud Run then starts your container on demand to handle requests, and ensures that *all incoming requests* are handled by dynamically adding and removing containers.

Cloud Run is serverless, which means that you, as a developer, can focus on building your application and not on building and maintaining the infrastructure that powers it.

Cloud Run also has a source-based workflow



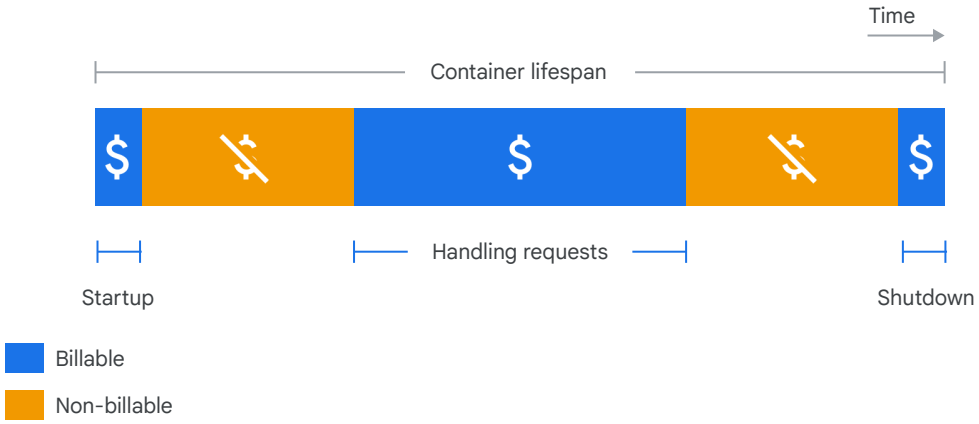
For some use cases, a container-based workflow is great, because it gives you a great amount of transparency and flexibility. Sometimes, you're just looking for a way to turn source code into an HTTPS endpoint, and you want your vendor to make sure it your container image is secure, well-configured and built in a consistent way.

With Cloud Run, you can do both. You can use a container-based workflow, as well as a source-based workflow.

The source-based approach will deploy source code, instead of a container image. Cloud Run then builds the source and packages the application into a container image.

Cloud Run does this using Buildpacks - an open source project.

You are only charged when your container handles requests



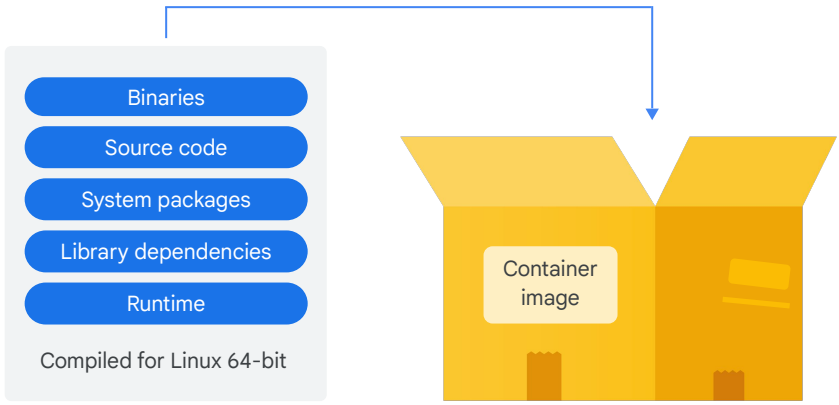
The pricing model on Cloud Run is unique; as you only pay for the system resources you use while a container is handling web requests, with a granularity of 100ms, and when it is starting or shutting down.

You do not pay for anything if your container does not handle requests. Additionally, there is a small fee for every one million requests you serve.

The price of container time increases with CPU and memory. A container with more vCPU and memory is more expensive. Today, Cloud run can allocate up to 4 vCPUs and 8GB of memory.

Most of the other compute products (such as Compute Engine), charge for servers as long as they are running, even if you are not using them. That means you're often paying for idle server capacity.

Cloud Run can run any binary



You can use Cloud Run to run any binary, as long as it is compiled for Linux sixty-four bit.

Now, this means you can use Cloud Run to run web applications written using popular languages, such as: Java, Python, Node.js, PHP, Go, and C++.

And you can also run code written in less popular languages: Cobol, Haskell, and Perl.

As long as your app handles web requests, you're good to go.

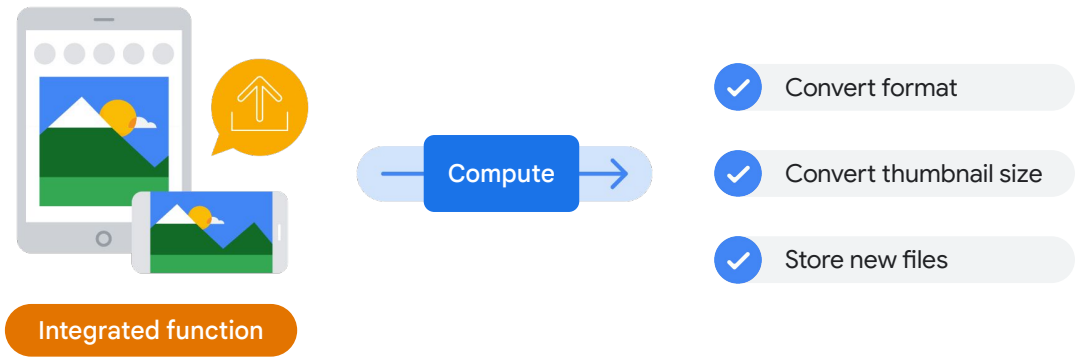
Applications in the Cloud

01 Cloud Run

02 Cloud Functions

And then there is **Cloud Functions**.

Integrated cloud functions handle application events



Many applications contain event-driven parts. For example, an application that lets users upload images. When that event takes place, the image might need to be processed in a few different ways, like converting the image to a standard format, converting a thumbnail into different sizes, and storing each new file in a repository.

This function could be integrated into an application, but then you'd have to provide compute resources for it–whether it happens once a millisecond or once a day.

Cloud Functions allows your code to respond to events

- Lightweight, event-based, asynchronous compute solution
- Allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment
- Use these functions to construct applications from bite-sized business logic and connect and extend cloud services
- Billed to the nearest 100 milliseconds, and only while your code is running
- Supports writing source code in a number of programming languages, including Node.js, Python, Go, Java, .Net Core, Ruby, and PHP
- Events from Cloud Storage and Pub/Sub can trigger Cloud Functions asynchronously, or use HTTP invocation for synchronous execution

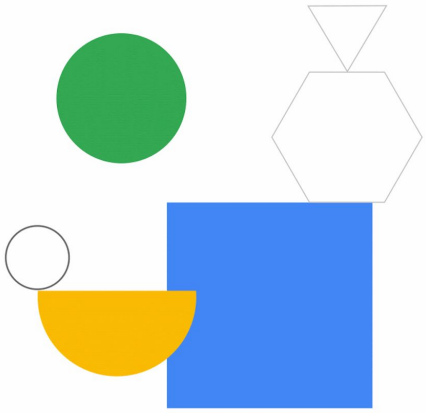
With **Cloud Functions**, you write a single-purpose function that completes the necessary image manipulations, and then arrange for it to automatically run whenever a new image is uploaded.

Cloud Functions is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment. These functions can be used to construct applications from bite-sized business logic. Cloud Functions can also be used to connect and extend cloud services. You are billed to the nearest 100 milliseconds, but only while your code is running.

Cloud Functions supports writing source code in a number of programming languages. These include Node.js, Python, Go, Java, .Net Core, Ruby, and PHP. For more information about the supported specific versions, refer to the runtimes documentation.

Events from Cloud Storage and Pub/Sub can trigger Cloud Functions asynchronously, or you can use HTTP invocation for synchronous execution.

Module Quiz



Quiz | Question 1

Question

Select the managed compute platform that lets you run stateless containers through web requests or Pub/Sub events.

A. Cloud Endpoints

B. Cloud Run

C. Apigee API Management

D. Cloud Source Repositories

Quiz | Question 1

Answer

Select the managed compute platform that lets you run stateless containers through web requests or Pub/Sub events.

A. Cloud Endpoints

B. Cloud Run 

C. Apigee API Management

D. Cloud Source Repositories

Select the managed compute platform that lets you run stateless containers through web requests or Pub/Sub events.

A: Cloud Endpoints
Feedback: Review the "Cloud Run" lecture.

B: Cloud Run
Feedback: Correct!

C: Apigee API Management
Feedback: Review the "Cloud Run" lecture.

D: Cloud Source Repositories
Feedback: Review the "Cloud Run" lecture.

Quiz | Question 2

Question

Cloud Run can only pull images from (Select 2):

A. Docker Hub

B. Self-hosted registries

C. Artifact Registry

D. GitHub

E. Cloud Storage

Quiz | Question 2

Answer

Cloud Run can only pull images from (Select 2):

A. Docker Hub 

B. Self-hosted registries

C. Artifact Registry 

D. GitHub

E. Cloud Storage

Cloud Run can only pull images from (Select 2):

A: Docker Hub
Feedback: Correct!

B: Self-hosted registries
Feedback: Review the "Cloud Run" lecture.

C: Artifact Registry
Feedback: Correct!

D: GitHub
Feedback: Review the "Cloud Run" lecture.

E: Cloud Storage
Feedback: Review the "Cloud Run" lecture.

Quiz | Question 3

Question

Why might a Google Cloud customer choose to use Cloud Functions?

A. Their application contains event-driven code that they don't want to provision compute resources for.

B. Cloud Functions is the primary way to run C++ applications in Google Cloud.

C. Cloud Functions is a free service for hosting compute operations.

D. Their application has a legacy monolithic structure that they want to separate into microservices.

Quiz | Question 3

Answer

Why might a Google Cloud customer choose to use Cloud Functions?

A. Their application contains event-driven code that they don't want to provision compute resources for.

B. Cloud Functions is the primary way to run C++ applications in Google Cloud.

C. Cloud Functions is a free service for hosting compute operations.

D. Their application has a legacy monolithic structure that they want to separate into microservices.

Why might a Google Cloud customer choose to use Cloud Functions?
***A: Their application contains event-driven code that they don't want to provision compute resources for.**
Feedback: Correct!
B: Cloud Functions is the primary way to run C++ applications in Google Cloud.
Feedback: Review the "Cloud Functions" lecture.
C: Cloud Functions is a free service for hosting compute operations.
Feedback: Review the "Cloud Functions" lecture.
D: Their application has a legacy monolithic structure that they want to separate into microservices.
Feedback: Review the "Cloud Functions" lecture.

Lab Intro

Hello Cloud Run

The goal of this lab is for you to build a simple containerized application image and deploy it to Cloud Run.