



Architecture & Deployment

2025-2026 v0.1.0 on branch main Rev: bf5a3ed8baf85ebafdf2c8031e836d37fa6b3121

Platform-as-a-Service (PaaS)

Learn to deploy web applications on Platform-as-a-Service (PaaS) cloud application platforms such as [GitHub Pages](#), [Netlify](#), and [Render](#).

You will need

- [Git](#)
- A free [GitHub](#) account

Recommended reading

- [Command line](#)
- [Git](#), [Git branching](#), [Collaborating with Git](#), [Git hooks](#)

What is PaaS?

Platform-as-a-Service (PaaS) is a **cloud deployment** service model where the developer deploys applications without the complexity of building and maintaining the infrastructure.

[Heroku](#), one of the first cloud platforms, has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, **Node.js**, Scala, Clojure, Python, PHP, and Go. Today there are many alternatives such as [Netlify](#), [Fly](#), or [Render](#).

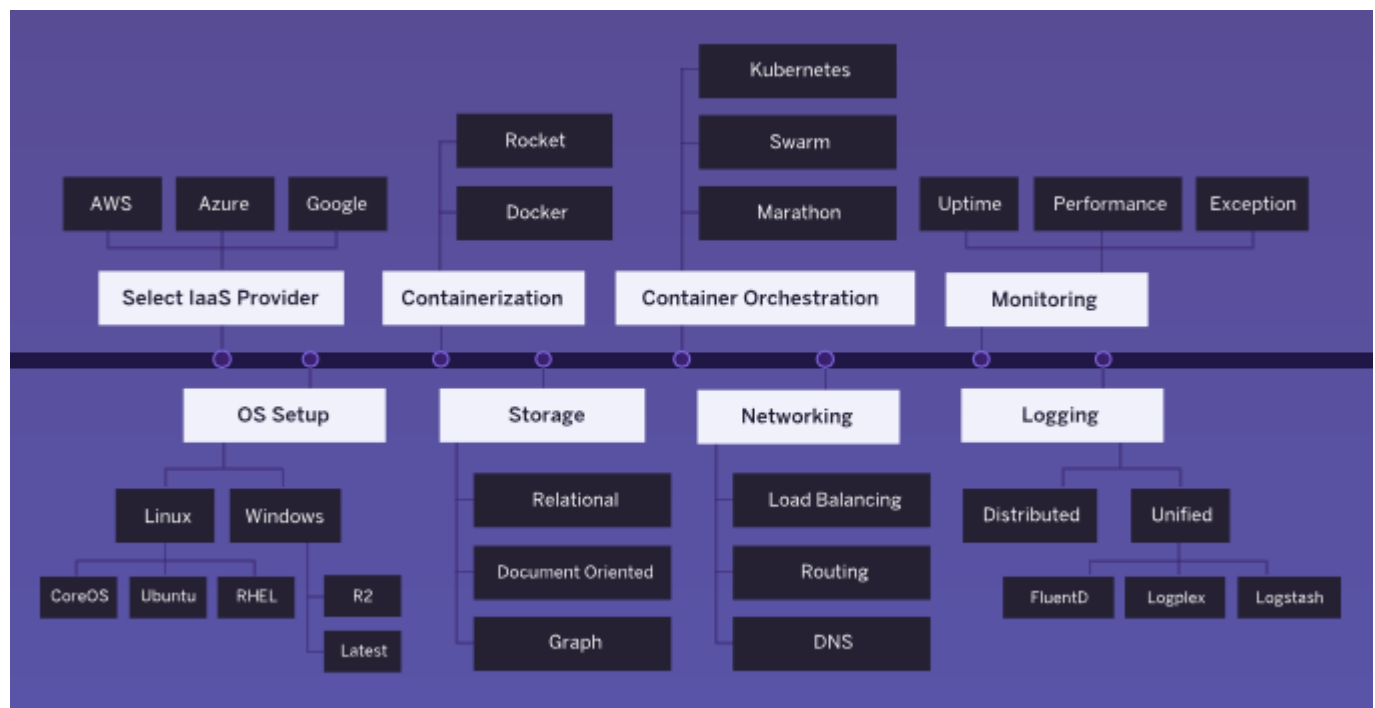
Cloud service models

Cloud-computing providers offer their services according to **different models**, some of which are listed below:

Service models	What they provide	Examples
<u>Infrastructure as a Service</u>	Servers, virtual machines	Amazon EC2, Azure (Microsoft), Rackspace
<u>Platform as a Service</u>	Runtime environments	Heroku, OpenShift, Netlify, Fly, Render
<u>Software as a Service</u>	Online services	Gmail
<u>Functions as a Service</u>	Serverless environments	Amazon Lambda, OpenWhisk (IBM)

Infrastructure as a Service (IaaS)

With traditional cloud providers, you have to **set up, maintain and operate** the **infrastructure** on which your applications are run:



You'll often need a professional **system administrator** to do that for sizable projects.

Platform as a Service (PaaS)

The goal of PaaS platforms is to get **straight to building applications**.



- Higher-level programming
- Reduced complexity
- Effective deployment with built-in infrastructure
- Easier maintenance
- Scaling

It's also a part of the DevOps movement where software **developers** increasingly step into the world of **operations** and vice-versa.

How does a PaaS platform work?

PaaS platforms usually run your applications inside **containers** on a fully **managed runtime environment**.

As a developer, you will deploy your **code** written with your favorite programming language and framework to a **build system** which automatically performs the necessary steps to build and deploy your application.

The various other components required to run and deploy your applications, such as the **database** and **reverse proxy** are installed and configured for you. The **system and language** stacks are **monitored, patched, and upgraded**, so it's always ready and up-to-date.

Is it magic?

A PaaS platform will often assume you are following the **conventions** of the language or framework you're using. For example, when deploying a Node.js application, many PaaS platforms assume:

- You are using npm, the Node.js Package Manager, and have a `package.json` file.
- The dependencies for your application can be installed by executing `npm install`.
- Your application can be run by executing `npm start`.

Similar conventions exist for each language supported by Heroku (e.g. using Maven for Java, Composer for PHP, Bundler for Ruby).

As for automation, it is usually achieved through Git hooks. New versions of your application will be **automatically deployed every time you push your latest commits**.

How do I use a PaaS platform?

You will usually **deploy by pushing your commits to a remote Git repository**, hosted either directly on the PaaS platform you are using, or on GitHub (or your favorite hosting service) if the PaaS platform supports it.

Many PaaS platforms offer **free plans with restrictions**. For example, Render lets you use one free PostgreSQL database for 90 days. You can of course whip out your credit card and pay to use it longer, or to have more storage/power/etc.

When you need to **configure your application**, PaaS platforms will usually allow you to define environment variables through a web interface. These variables will be provided to your application when it is deployed.

Some PaaS platforms provide **command line tools** to help make and manage your deployments. For example, there is the Heroku CLI and the Netlify CLI.

Table of contents

- [What is PaaS?](#)
 - [Cloud service models](#)
 - [Infrastructure as a Service \(IaaS\)](#)
 - [Platform as a Service \(PaaS\)](#)
 - [How does a PaaS platform work?](#)
 - [Is it magic?](#)
 - [How do I use a PaaS platform?](#)

[↑](#) Back to top