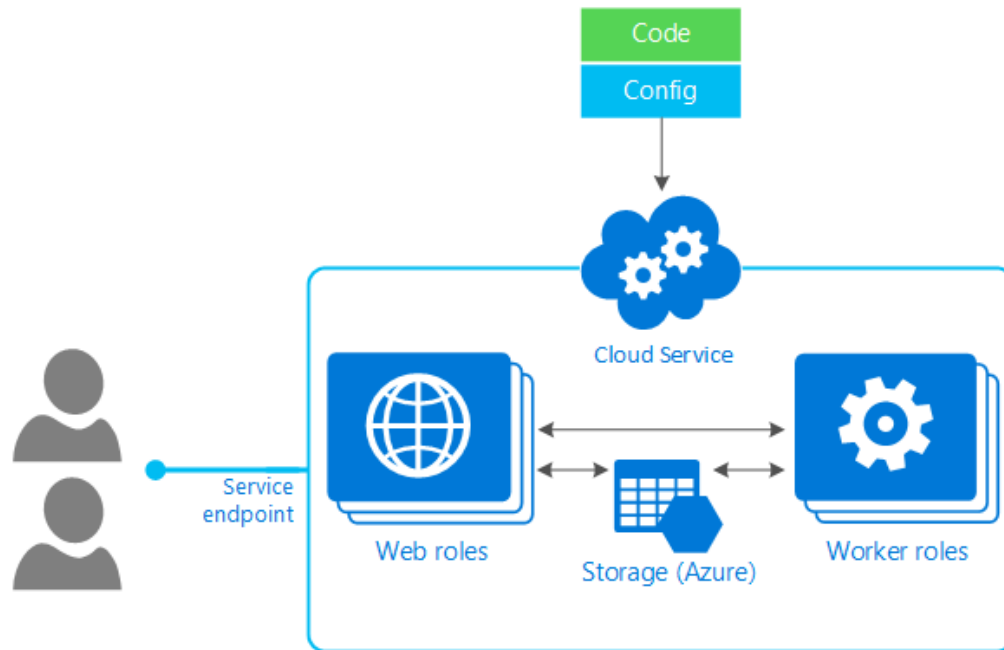


Azure Cloud Services is an example of a [platform as a service](#) (PaaS). Like [Azure App Service](#), this technology is designed to support applications that are scalable, reliable, and inexpensive to operate. In the same way that App Service is hosted on virtual machines (VMs), so too is Azure Cloud Services. However, you have more control over the VMs. You can install your own software on VMs that use Azure Cloud Services, and you can access them remotely.



More control also means less ease of use. Unless you need the additional control options, it's typically quicker and easier to get a web application up and running in the Web Apps feature of App Service compared to Azure Cloud Services.

There are two types of Azure Cloud Services roles. The only difference between the two is how your role is hosted on the VMs:

- **Web role:** Automatically deploys and hosts your app through IIS.
- **Worker role:** Does not use IIS, and runs your app standalone.

For example, a simple application might use just a single web role, serving a website. A more complex application might use a web role to handle incoming requests from users, and then pass those requests on to a worker role for processing. (This communication might use [Azure Service Bus](#) or [Azure Queue storage](#).)

As the preceding figure suggests, all the VMs in a single application run in the same cloud service. Users access the application through a single public IP address, with requests automatically load balanced across the application's VMs. The platform [scales and deploys](#) the VMs in an Azure Cloud Services application in a way that avoids a single point of hardware failure.

Even though applications run in VMs, it's important to understand that Azure Cloud Services provides PaaS, not infrastructure as a service (IaaS). Here's one way to think about it. With IaaS, such as Azure Virtual Machines, you first create and configure the environment your application runs in. Then you deploy your application into this environment. You're responsible for managing much of this world, by doing things such as deploying new patched versions of the operating system in each VM. In PaaS, by contrast, it's as if the environment already exists. All you have to do is deploy your application. Management of the platform it runs on, including deploying new versions of the operating system, is handled for you.

Scaling and management

With Azure Cloud Services, you don't create virtual machines. Instead, you provide a configuration file that tells Azure how many of each you'd like, such as "three web role instances" and "two worker role instances." The platform then creates them for you. You still choose [what size](#) those backing VMs should be, but you don't explicitly create them yourself. If your application needs to handle a greater load, you can ask for more VMs, and Azure creates those instances. If the load decreases, you can shut down those instances and stop paying for them.

An Azure Cloud Services application is typically made available to users via a two-step process. A developer first [uploads the application](#) to the platform's staging area. When the developer is ready to make the application live, they use the Azure portal to swap staging with production. This [switch between staging and production](#) can be done with no downtime, which lets a running application be upgraded to a new version without disturbing its users.

Monitoring

Azure Cloud Services also provides monitoring. Like Virtual Machines, it detects a failed physical server and restarts the VMs that were running on that server on a new machine. But Azure Cloud Services also detects failed VMs and applications, not just hardware failures. Unlike Virtual Machines, it has an agent inside each web and worker role, and so it's able to start new VMs and application instances when failures occur.

The PaaS nature of Azure Cloud Services has other implications, too. One of the most important is that applications built on this technology should be written to run correctly when any web or worker role instance fails. To achieve this, an Azure Cloud Services application shouldn't maintain state in the file system of its own VMs. Unlike VMs created with Virtual Machines, writes made to Azure Cloud Services VMs aren't persistent. There's nothing like a Virtual Machines data disk. Instead, an Azure Cloud Services application should explicitly write all state to Azure SQL Database, blobs, tables, or some other external storage. Building applications this way makes them easier to scale and more resistant to failure, which are both important goals of Azure Cloud Services.