Google Cloud

A brief guide to containers

So far, you've learned that containers are software packages that include all the necessary elements you need to run in an environment. Containers enable you to share memory, network resources, and storage at the operating systems level. In this reading, you'll learn more about what containers do in cloud computing, and explore how they're made.

Containers

Containers are sets of elements that enable programs to run in a completely contained environment. They have several built-in elements, including:

- Libraries: Containing the prewritten, reusable code your application depends on
- Other dependencies: Including system tools and settings

Containers are important in cloud computing because they enable developers and development teams to deploy software quickly and efficiently. When an application is running in a container, it's running in its own isolated environment. You can also move the container across different operating systems. Containers can run on-premises in data centers, and in the cloud.

How containers are made

Here's a brief overview of how containers are made and deployed into a cloud environment:

1. A developer creates and names a directory. The developer names the directory after the app going into it.
2. The developer creates a module and a file with the code for the application.
3. The application is put together with the elements it needs into a container image, which is a static, unchangeable file with executable code. Container images are read-only templates including instructions for creating any desired number of containers.
4. When the container image is run through a container building engine, it creates a container.
5. The container is now ready, and the application is running in the container since it has everything it needs built into it.

Virtual machines (VMs) vs containers

The cloud uses different types of virtual machines (VMs), since users don't have access to the physical data center. A hypervisor is used to manage VMs, building an abstraction layer that

enables the VM to have virtual hardware and a full operating system. The hypervisor allocates physical resources to VMs. You can use VMs to put multiple operating systems on the same physical components. This allows data centers to handle a lot of VMs.

Containers and VMs take different approaches. Here are some examples:

- Containers share the host's OS system kernel instead of having their own complete operating systems.
- For VMs, there are isolated instances of operating systems on physical hosts. So, each VM is like a complete physical computer.
- You can use both containers and VMs together with the container running an application, while the VM provides the underlying infrastructure to the container.

Key takeaways

Containers work as isolated environments where applications have all the libraries and resources they need to run. Containers can be moved across different operating systems, and through the cloud. Containers are different from VMs because VMs include virtual hardware, an operating system, and a kernel.