Serverless functions and security

So far, you've learned that containers are useful for cloud computing, because they're standalone packages that include everything you need to run a fully containerized application. Containers partition hosts into user space environments. Certain types of applications can also be broken up into functions and hosted using serverless functions. For these functions, server resources are granted access to the server space by a service provider as needed by the application. In this reading, you'll learn about serverless functions, how they relate to containers, and how to keep them secure.

What are serverless functions and who is responsible?

A serverless function is a bundle of code you can upload to a service provider. The service provider offers a fully managed environment to execute this code in. You can configure this function to respond to requests coming through a URL. Or you can configure this function to be called from other services or serverless functions, or run on a schedule. You can use them to add functionality to frontend applications, or you can use them to build whole applications.

Serverless functions and host servers

Serverless functions run on a host server, but only use resources as needed. The service provider provisions resources as the application needs them. Serverless functions are not assigned specific machines.

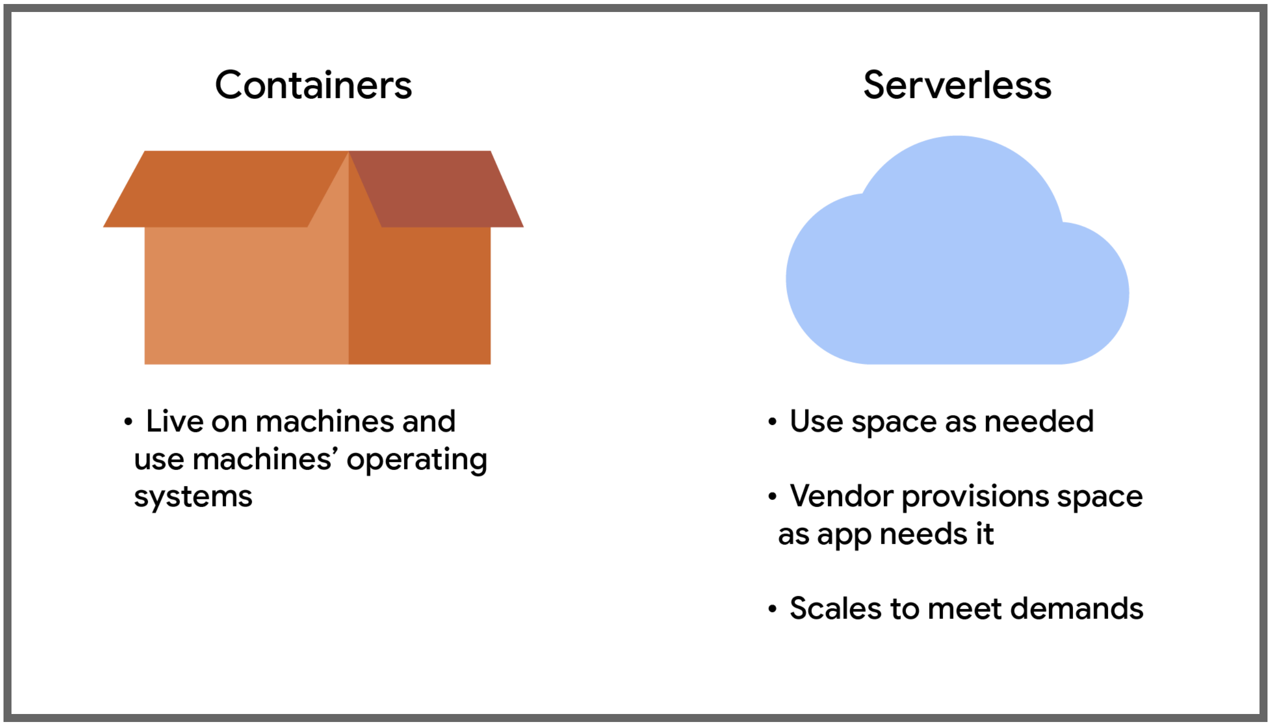
Shared responsibility

Serverless functions follow the shared responsibility model in the cloud. The cloud provider is responsible for the hardware, while the cloud user is responsible for the code of the functions, including keeping the code secure.

Differences between serverless functions and standard containers

There are several differences between serverless functions and standard containers, including:

- Resources are provisioned for serverless functions based on needed resources.
- Containers live on machines and use the machines' operating systems.
- Containers require manual or automated scaling through orchestration platforms like Kubernetes or Docker Swarm.
- Serverless functions automatically and dynamically provision resources based on demand and scale.



Containers

- Live on machines and use machines’ operating systems

Serverless

- Use space as needed
- Vendor provisions space as app needs it
- Scales to meet demands

Serverless functions and security

Like containers and VMs, security is vital in serverless functions, so it’s important to address security concerns like:

- Authentication and authorization by retaining user data
- Privileges and roles
- Network security

Privileges and roles

The concept of least privilege applies to serverless functions, since serverless functions perform tasks in response to actions or events. Best practices to follow include:

- Do not have two service accounts on the same resource defining their actions.
- Do not share service accounts across multiple serverless functions. This will violate the principle of least privilege by associating all permissions multiple functions require, with the same role.

Network security

Network security is a vital component of serverless security, encompassing the management of incoming (ingress) and outgoing (egress) traffic. Effective network security measures restrict unauthorized access and protect the function's environment.

Key takeaways

Serverless functions are granted resources by service providers, and only use as much as they need. The shared responsibility model is also applicable to serverless functions. The cloud provider is responsible for the physical infrastructure, while the user is responsible for security. You need to use authentication and assign service accounts to keep your serverless functions secure.