Serverless Architecture Security Challenges

by Vinay Gujar

Google Cloud Functions

IBM Cloud Functions

Apache OpenWhisk

Some of the top FaaS providers are :

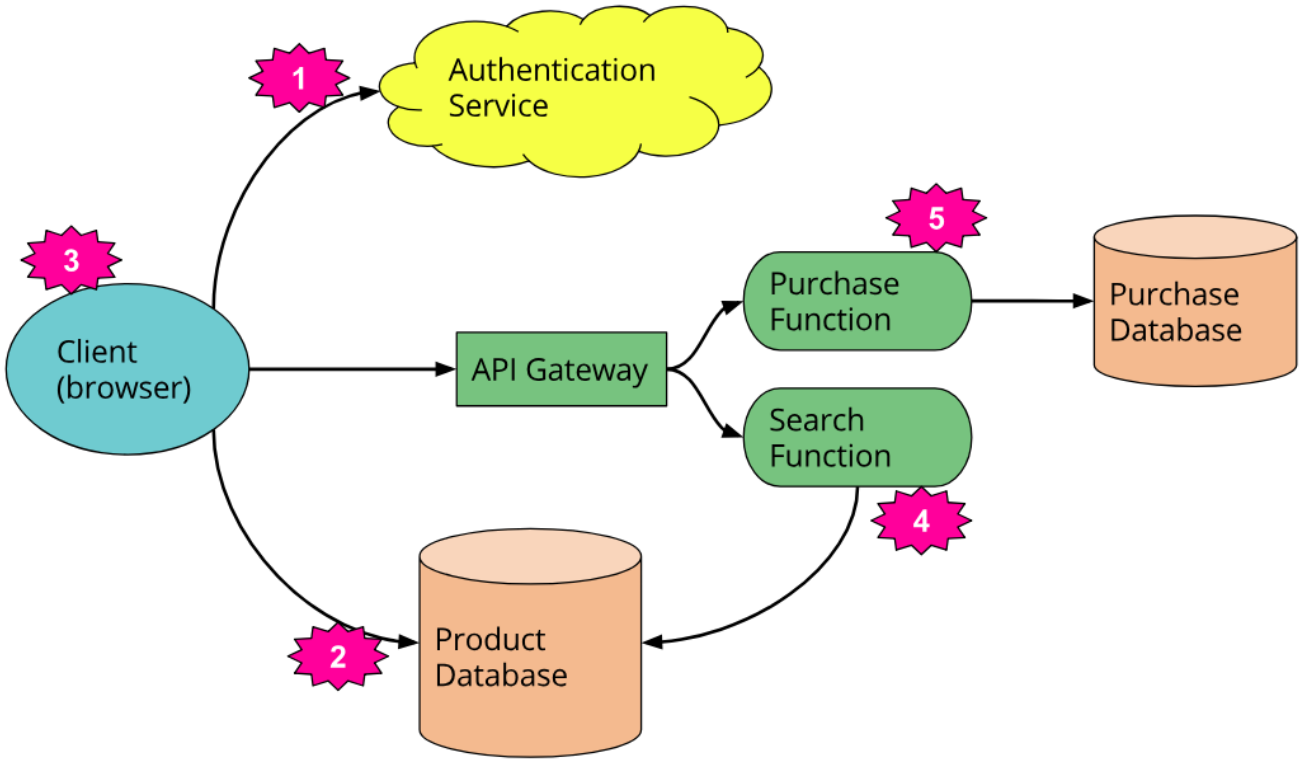
1. AWS Lambda
2. Google Functions
3. Microsoft Azure Functions
4. IBM Apache OpenWhisk Functions

What is serverless architecture?

- A serverless architecture is a way to build and run applications and services without having to manage infrastructure.
- Also known as Function as a Service or FaaS
- Your application still runs on servers, but all the server management is done by the FaaS provider.

How a traditional server works?

How FaaS works?



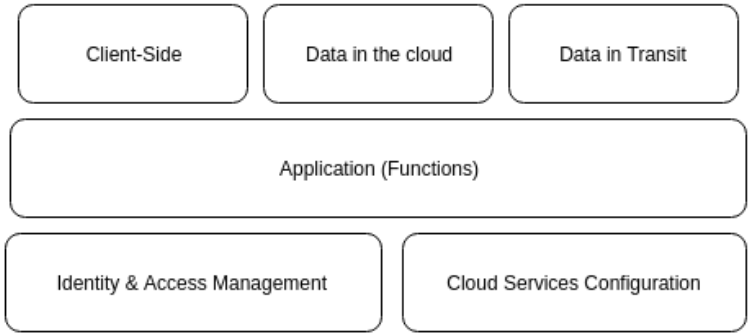
Why go serverless?

- Zero administration — Infra Operations aren't to be bothered.
- Auto-scaling — the serverless application can be scaled dynamically according to cloud workloads.
- Pay-per-use — Function-as-a-service (FaaS) computes and manages services that are charged upon usage.

Security responsibilities

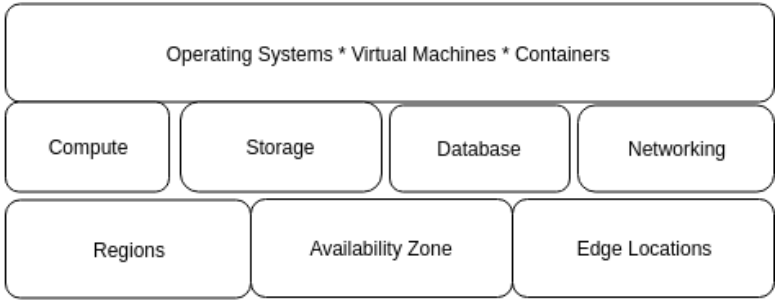
Application Owner

Responsible for security "in" the cloud & client.



FaaS Provider

Responsible for security "of" the cloud



The Shared Security Responsibilities Model for Serverless Architectures

Serverless architecture common flaws

Increased attack surface

Serverless functions consume data from a wide range of event sources such as

- HTTP APIs
- message queues
- cloud storage
- IoT device communications and so forth.

Attack surface complexity

The attack surface in serverless architectures can be difficult
for some to understand given that such architectures are still rather new

Overall system complexity

Visualizing and monitoring serverless architectures is still more complex than standard software environments

Inadequate security testing

Performing security testing for serverless architectures is more complex than testing standard applications

- DAST (dynamic application security testing)
- SAST (static application security testing)
- IAST (interactive application security testing)
- Traditional security protections (Firewall, WAF, IPS/IDS)

Critical Risks

- (SAS-1) — Function Event Data Injection
- (SAS-2) — Broken Authentication
- (SAS-3) — Insecure Serverless Deployment Configuration
- (SAS-4) — Over-Privileged Function Permissions and Roles
- (SAS-5) — Inadequate Function Monitoring and Logging

Critical Risks (cont)

- (SAS-6) — Insecure 3rd Party Dependencies
- (SAS-7) — Insecure Application Secrets Storage
- (SAS-8) — Denial of Service and Financial Resource Exhaustion
- (SAS-9) — Serverless Function Execution Flow Manipulation
- (SAS-10) — Improper Exception Handling and Verbose Error Messages


Questions?