**1. High Availability**

* **Built-In Fault Tolerance**: Serverless platforms such as AWS Lambda, Google Cloud Functions, and Azure Functions are designed to be highly available by default. They automatically distribute your function execution across multiple physical locations to ensure continuous operation even if a specific location experiences an outage.
* **Redundant Infrastructure**: Serverless services leverage cloud provider infrastructure that includes redundancy and data replication across multiple regions and availability zones, ensuring minimal downtime and consistent availability.

**2. Resilience**

* **Automatic Failover**: Serverless architectures can automatically route traffic to healthy instances or functions if a failure is detected in a particular instance or region. This built-in failover mechanism enhances the resilience of your application.
* **Stateless Functions**: Serverless functions are inherently stateless, meaning they do not store any user or application data locally. This design principle allows functions to be quickly restarted or replaced without losing critical state information, contributing to overall system resilience.

**3. Security**

* **Managed Security Updates**: Cloud providers handle the patching and updating of the underlying infrastructure, reducing the risk of security vulnerabilities. This ensures that your functions are running on the latest, most secure infrastructure.
* **Fine-Grained Permissions**: Serverless architectures often leverage fine-grained access controls, such as AWS IAM roles for Lambda functions, to ensure that functions have the minimum necessary permissions to perform their tasks, enhancing overall security.
* **Isolation**: Each serverless function runs in its own isolated environment, which can reduce the risk of security vulnerabilities spreading across the system.
* **Managed Security**: Cloud providers often offer built-in security features and best practices, reducing the burden on developers to secure the infrastructure.

**4. Scalability**

* **Automatic Scaling**: Serverless platforms automatically scale your functions in response to incoming requests. This ensures that your application can handle any level of traffic without manual intervention or provisioning.
* **Global Distribution**: Functions can be deployed across multiple regions globally, enabling low-latency responses and distributing the load evenly. This allows your application to scale horizontally and handle a large number of concurrent users efficiently.

**5. Cost Efficiency**

* **Pay-as-you-go Pricing**: With serverless, you only pay for the compute time you consume. There are no costs associated with idle time, unlike traditional server-based architectures.
* **Reduced Operational Costs**: Since serverless platforms handle server management, scaling, and maintenance, you save on costs associated with these tasks.