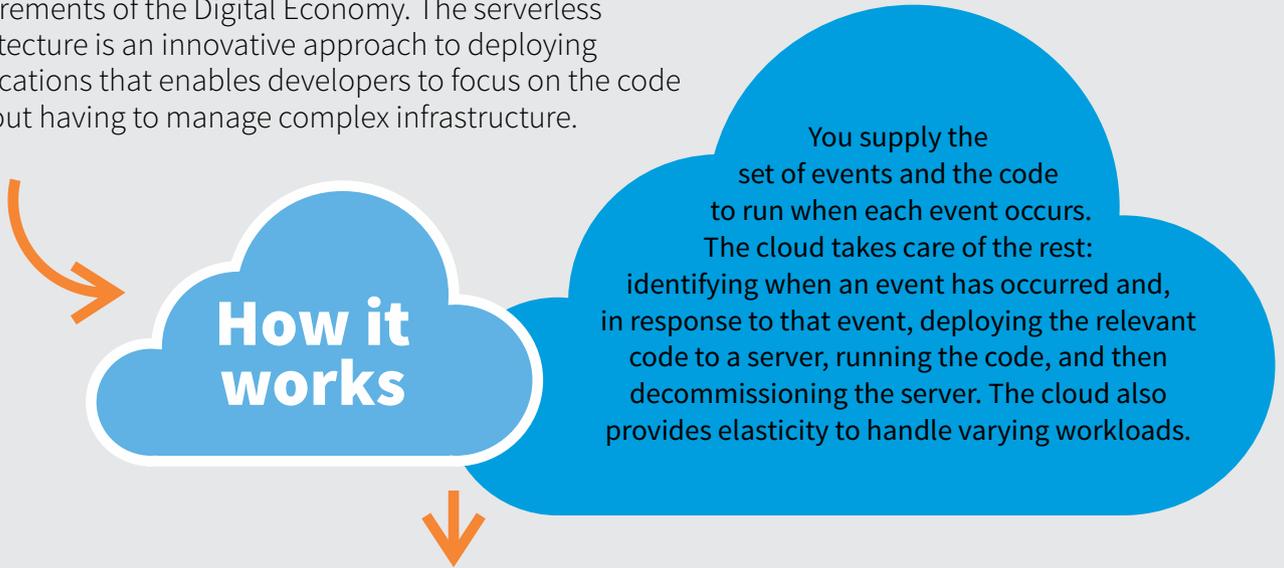# Using Serverless Computing to Modernize Data Applications

Companies often need to modernize their technology platforms to accommodate time-to-market and data analytics requirements of the Digital Economy. The serverless architecture is an innovative approach to deploying applications that enables developers to focus on the code without having to manage complex infrastructure.

## How it works

You supply the set of events and the code to run when each event occurs. The cloud takes care of the rest: identifying when an event has occurred and, in response to that event, deploying the relevant code to a server, running the code, and then decommissioning the server. The cloud also provides elasticity to handle varying workloads.

## Benefits

**Cost**: You will be charged as per your actual usage of server resources, and you do not have to pay for a dedicated server.

**Flexibility**: Changing server needs for applications can be handled automatically on the cloud.

## Ideal Use Cases

For applications in which:

- The flow of the application can be expressed as responses to a series of events.
- Events occur sporadically. If your application is going to be constantly bombarded with events, it will be cheaper to rent an entire dedicated server rather than paying per event.
- Event processing is not resource intensive, e.g. does not require a lot of money or time.
- High latency (having to wait several seconds before an event is processed) is acceptable.

## Limitations

**High Latency:** With serverless computing, you must wait until the cloud platform allocates a server to your code. Hence, it can be a poor choice for applications that require quick, single-digit millisecond response to events.

**Resource Limits**: Because different cloud platforms place limits on serverless resource size, serverless can be a poor choice for applications that are memory intensive.

**Development Challenges**: Many of the events in a serverless program can only be generated in the cloud, and there are currently no good tools to emulate cloud events in a local development environment. This can present a challenge for developers.

**Testing Challenges**: To implement a real-world use case that accomplishes useful work, you have to simulate the flow of events in the correct order, as well as all other feasible orders. This requires a new set of test tools that is still evolving.

For more ideas on how you can modernize your technology platforms for digital business, contact us at **ness.com/contact** or **read our Blog.**