Serverless Architectures: The future of Software Development

. . .

For those working in the IT industry, Cloud computing is a term that, over the past 15 years, has been increasingly in the spotlight!

Cloud computing is at the center of the modern world. Briefly, it is the on-demand access to a shared pool of IT configurable resources and services through the internet where customers pay for what they use. In other words, it allows companies to rent hardware instead of having huge up-front costs on building their own infrastructure.

It brings several advantages such as agility, elasticity, cost savings, and go global in minutes.

What is Cloud Computing

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go ...

aws.amazon.com

What Is Cloud Computing? A Beginner's Guide | Microsoft Azure

Simply put, cloud computing is the delivery of computing services-including servers, storage, ...

azure.microsoft.com

What is Cloud Computing? | Google Cloud

Understanding the types of cloud computing resources can be time-consuming and costly. ...

cloud.google.com

If you are new to Cloud world, check the following introduction video from Amazon Web Services, which is the leading platform on the market.

"What is Cloud Computing?" by Amazon Web Services.

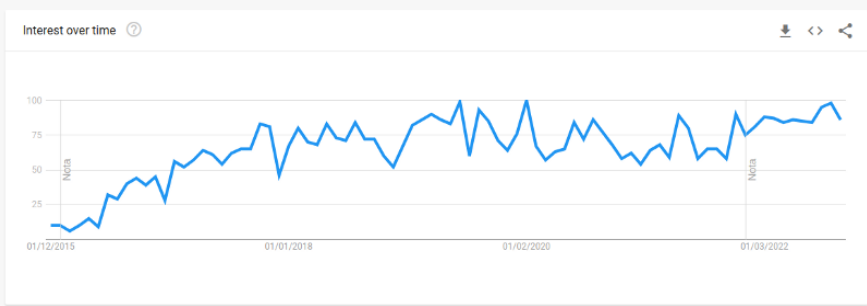
. . .

The Serverless architecture usage trend!

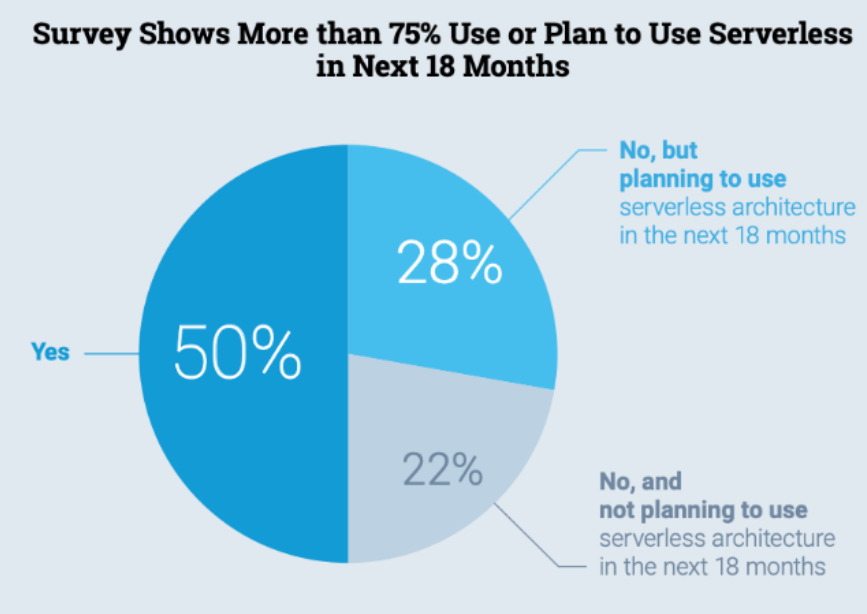
This new computing paradigm brought new ideas and trends to the market being Serverless computing one of the major ones. And for those who are already thinking in giving up on this article, let me give you some numbers.

According to this study from Fact.Mr, Cloud computing is expected to grow 15% annually. The market around Serverless architectures is forecasted to grow from $7.6 billion in 2020 to $21.1 billion in 2025, meaning that more and more companies are expected to surf on this trend in the upcoming years! Have I got your attention now?

Cloud computing has already become the dominant paradigm worldwide and Serverless architecture is its natural evolution.



Serverless Architecture term interest over time in Google Trends.



Source: The New Stack Serverless Survey 2018.

. . .

So, what is Serverless computing?

When you think about Serverless computing you might ask yourself: “Does this mean that there is no servers?”. No, you’re wrong! I will explain to you why.

Serverless architectures run on top of Cloud technologies and allow users to focus only on developing their own applications without the burden of worrying about the underlying infrastructure. All the provisioning, configuration, update, management, and scalability of the infrastructure are delegated to a third-party entity, the cloud provider.

> *“Serverless is a cloud-native development model that allows developers to build and run applications without having to manage servers.”*

> *Source: Red Hat*

"What is Serverless?! (in 3 minutes)" by Coding With Lewis.

Therefore, Serverless doesn't mean that there are no servers, it means that you don't have to manage those servers!

. . .

FaaS as an extension of Serverless computing

Function as a Service (FaaS) is a concept that goes hand in hand with Serverless computing since it is implemented in a Serverless environment and therefore it is a sub-component of Serverless architectures.

FaaS is a category of Serverless computing services that provide a way to build applications that are based on event-driven behavior. Events, such as a request on a specific API endpoint, will trigger functions that are called on-demand and are responsible for handling those events. Applications are composed as a set of separate functions, each performing a specific action.

The infrastructure and all issues related to it such as scaling the functions according to the demand is cloud provider's responsibility.

Some of the most popular FaaS examples are:

- AWS Lambda
- Microsoft Azure Functions

- Google Cloud Functions
- Red Hat Openshift Cloud Functions
- Cloudflare Workers

If you want to know more about FaaS, check the following introduction video on AWS Lambda, the leading service on the market.

"Introduction to AWS Lambda" by Amazon Web Services.

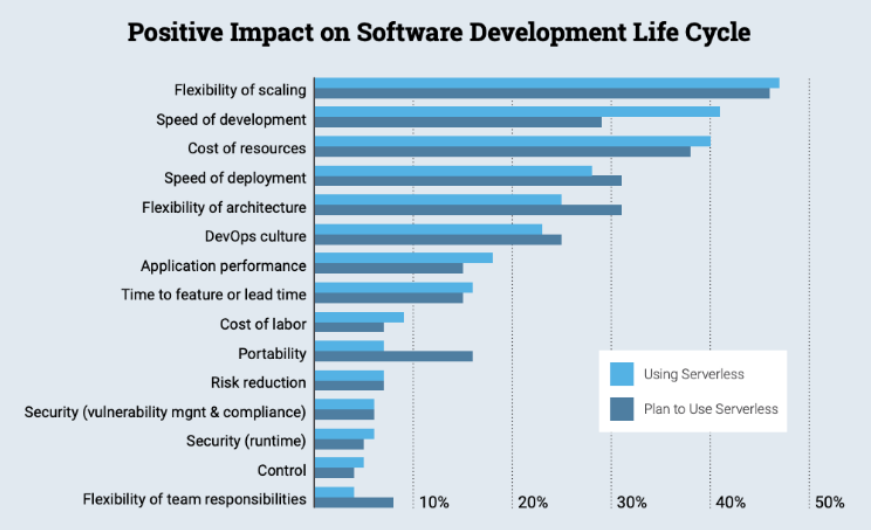
. . .

What are the advantages of Serverless architectures and FaaS?

Well, the list of perks is endless, however, let's highlight the main ones:

- Increase developers' productivity. This leads to a minimization of the time needed to develop an app.
- Cost-effective approach. Often users will pay less than with traditional architectures since they are not paying for unused reserved servers.
- Quick updates and deployments.
- Require less human resources since you don't have responsibility for the infrastructure.
- Simplified code since developers create independent functions.

- Infinite scaling. FaaS functions are automatically monitored and scaled according to the load.
- Lower latency. Code is not running on an origin server so it can run anywhere, thus closer to users.
- Ideal for microservice-based architectures.



Source: The New Stack Serverless Survey 2018.

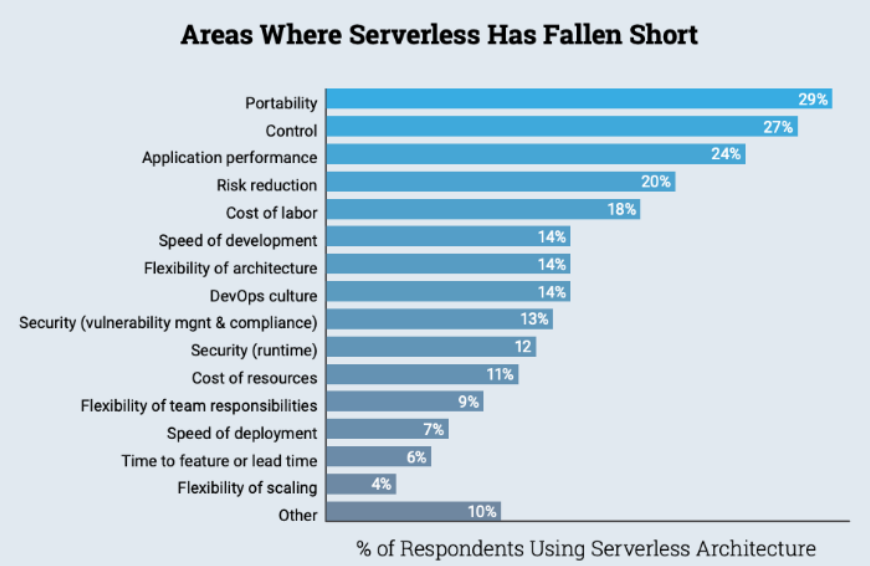
. . .

What are the downsides of this paradise?

Okay, so we can agree that Serverless architectures seem like bulletproof solutions at first sight. However, they also have drawbacks:

- No access to underlying infrastructure: If for some reason, you want to access the infrastructure that is running your code, you cannot!
- Cold Starts: When your FaaS function is not called for a while, handling the first request might take a while since functions need to be internally activated by the cloud platform resulting in a lack of performance.
- Vendor lock-in issue: Companies start developing their products focused on Serverless services provided by a specific cloud provider and become hostages of that provider. Porting the application to a new, cheaper or faster, provider might be difficult and require considerable effort and costs.

- Security: You are trusting your data to a third-party entity since your cloud provider will have access to it.
- Privacy: Your applications will run in environments that are shared with other cloud provider customers.
- It is not ideal for long-running workloads and high-computing operations due to resource limitations from providers.



Source: The New Stack Serverless Survey 2018.

. . .

Who is using Serverless architectures and FaaS to power up their own businesses?

Major companies worldwide are migrating their traditional architectures to serverless architectures due to the flexibility, greater scalability, faster and easier deploys at a reduced cost. Some examples are listed below:

- Netflix
- Codepen
- Nordstrom
- Figma
- Zalora
- Coca-Cola

Netflix Gains New Efficiencies using AWS Lambda by Amazon Web Services.

. . .

Who and Which Serverless services are available?

Serverless services are provided by several Cloud providers such as Amazon Web Services (AWS), Microsoft Azure (Azure), and Google Cloud Platform (GCP) which are the leaders of the current market.

We have already seen a set of Serverless services, the FaaS category. However, a wider range of services based on the Serverless concept quickly emerged and are increasingly being used. From compute, storage, database, application integration to even monitoring services.

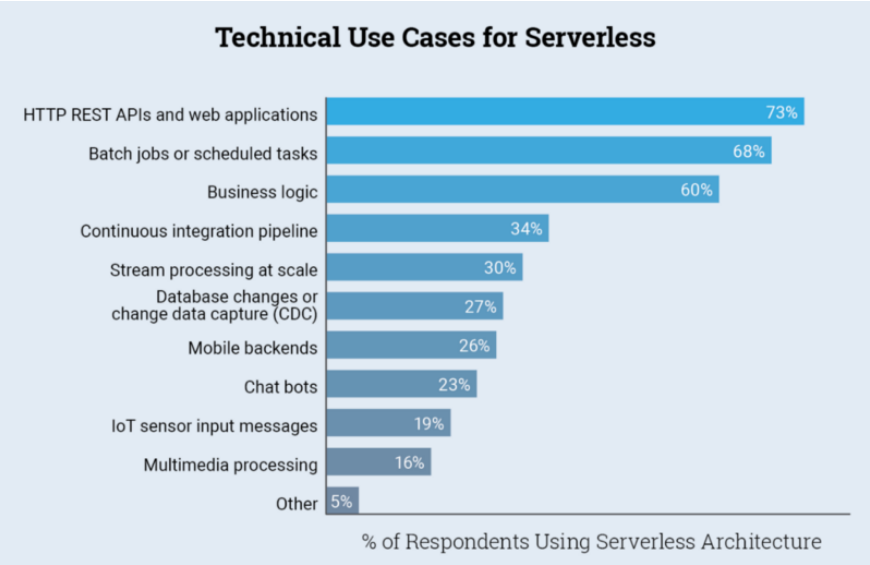
List of Serverless Services from Amazon Web Services. Source: itemis.

. . .

Where can Serverless architectures shine?

After this overview on Serverless architectures we must identify use cases where Serverless architectures are the most suitable to be used:

- Build APIs: RESTful APIs that scale with demand using a set of Serverless functions that handle the requests.
- Batch and Cron Jobs: Serverless managed services (ex: AWS Batch) allow users to run Batch jobs without provisioning and maintaining compute instances. This can be integrated with other services (ex: AWS CloudWatch) to periodically trigger these functions implementing cron jobs.
- Internet of Things: IoT devices perform tasks when triggered by an event.
- Multimedia processing: FaaS can be triggered to process media files on upload events.
- CI/CD Pipeline: This kind of architectures can automate the steps of the pipeline. Functions can be triggered when pipeline events occur such as a code commit.
- ChatBots: building chatbots has become much easier using a Serverless approach. Amazon Lex is being widely used by well-known services such as DropBox to build this kind of technology.



Source: The New Stack Serverless Survey 2018.

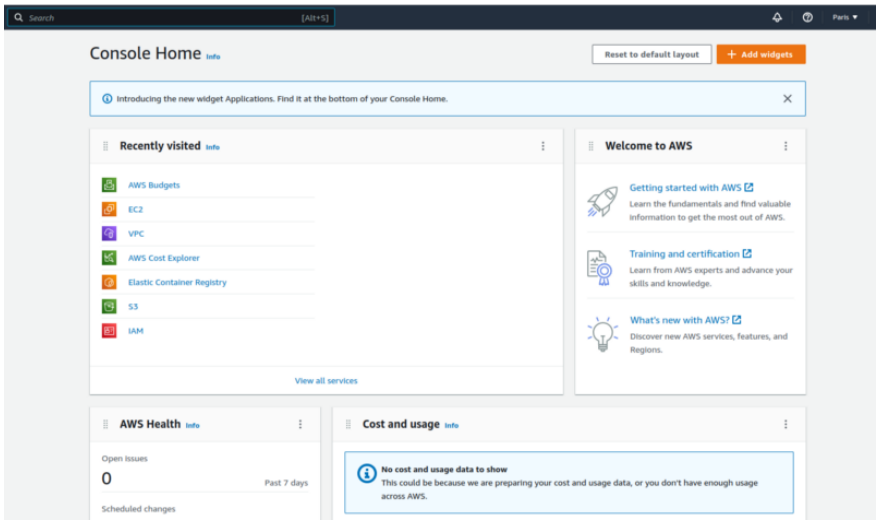
Let's build our first Serverless Function! Hands-On!

After all this theory, we want to get our hands dirty and enter in this Serverless cool world!

I will guide you through how you can build your first serverless function on AWS Lambda. It is required an AWS Account. If you don't have one, you can follow the following tutorial to create one:

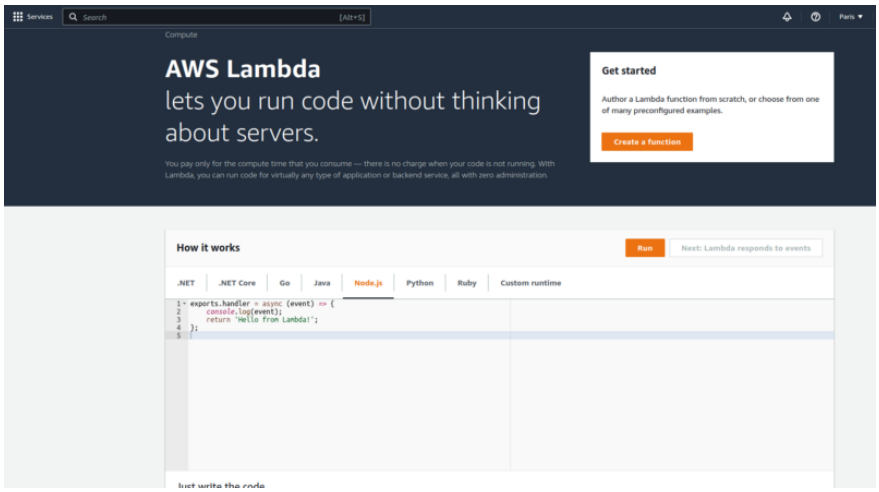
"How to Create AWS Account Free in 5 minutes" by LearnITGuide Tutorials.

Now that you already have an AWS account, log into your account and you should see the following initial dashboard:



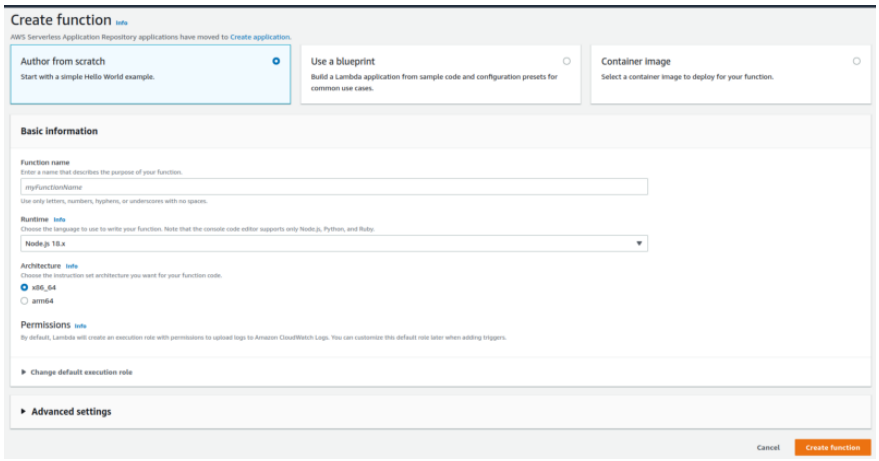
Amazon Web Services initial dashboard.

On the top left corner, search for "Lambda" and select Lambda Service. Now you are on AWS Lambda introduction page:



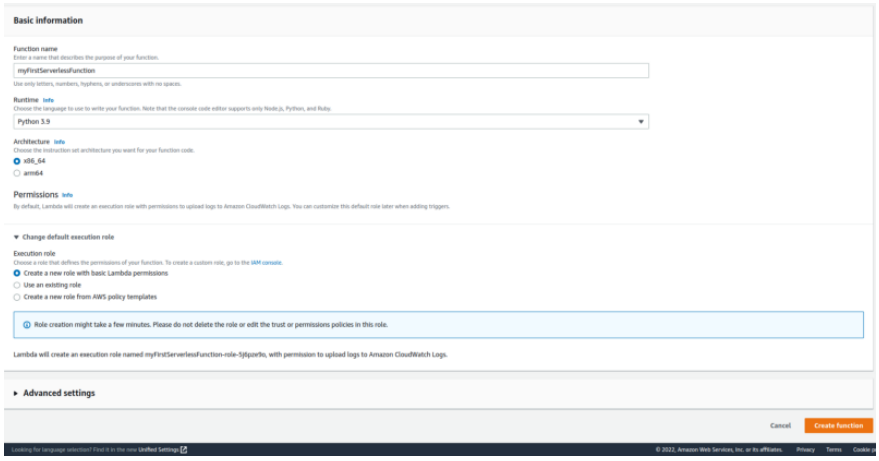
AWS Lambda introduction page.

To create your first function, click on “Create a Function” button and you will be redirected to the following function’s creation page:



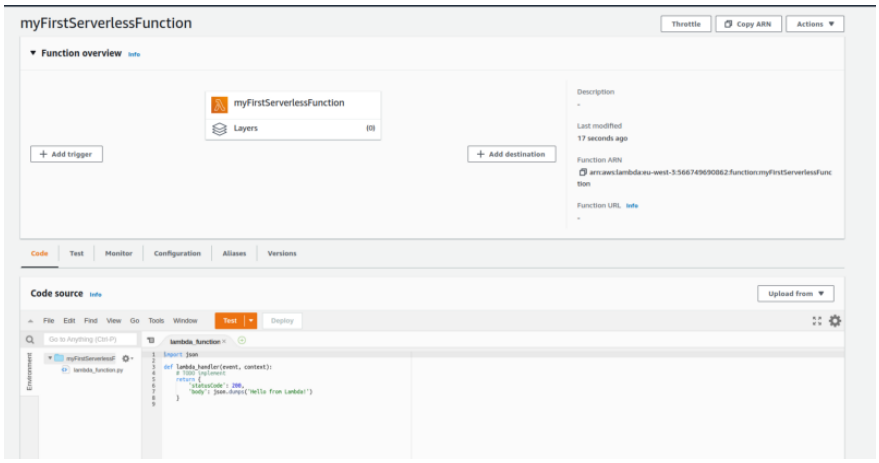
AWS Lambda Function creation page.

Here we are going to specify our function’s name, programming language, and permissions. You can give the name you want. We will be selecting python as our language and we will leave permission as default since we will only test the function inside AWS environment.



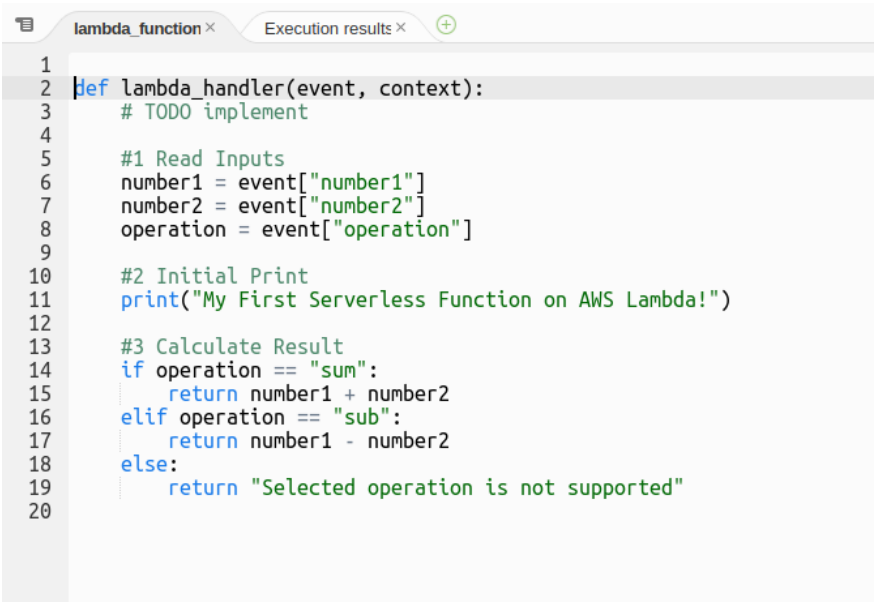
AWS Lambda Function creation parameters.

When you are ready, click “Create function” button. You will be redirected to the following page where you can change your function:



AWS Lambda Function page.

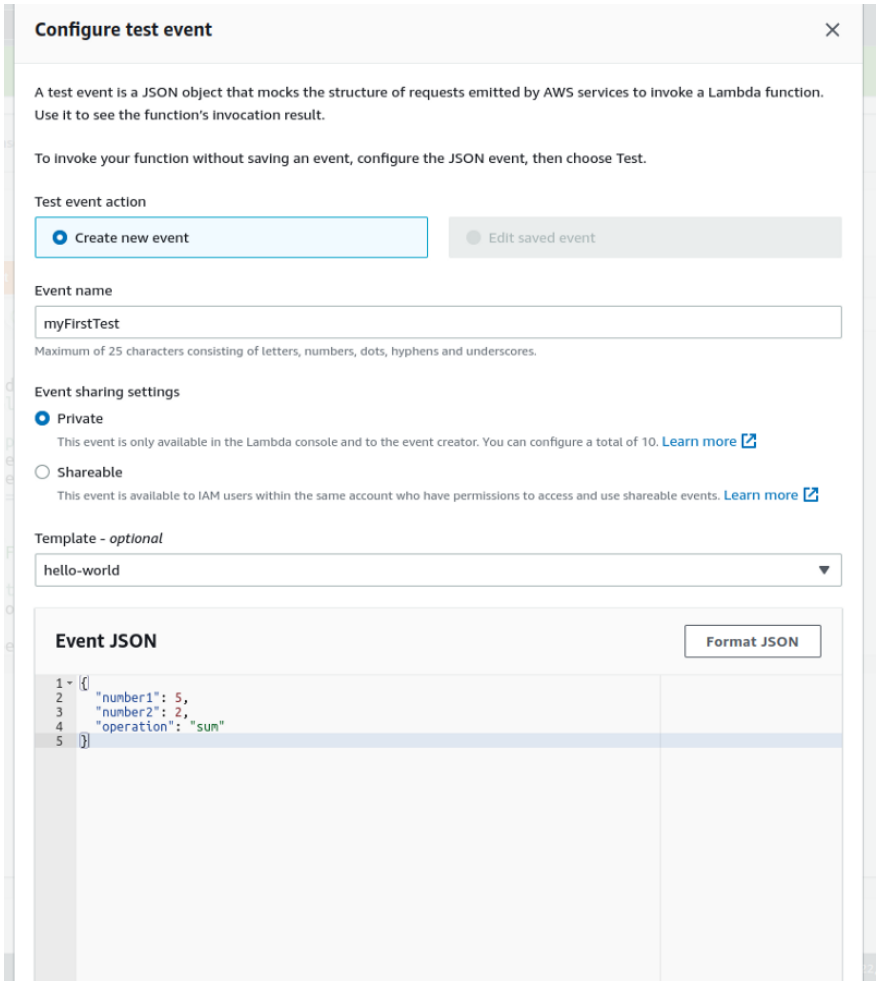
Let’s change the function code to work as a simple calculator with only 2 available operations:



AWS Lambda Function code changes.

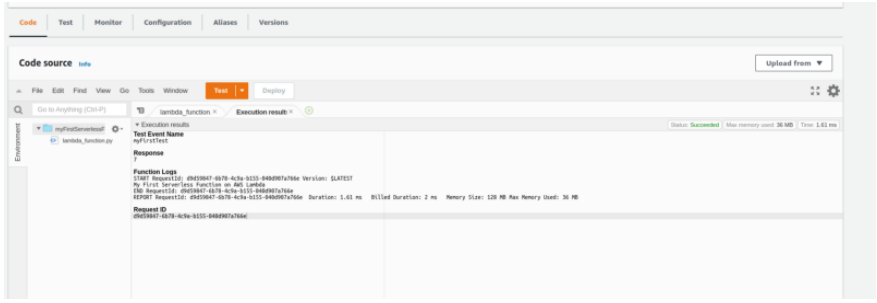
Click “Deploy” button to deploy your function with the code changes.

Next, let’s test our function. Click “Test” button to create a test event. Test events allow developers to create a sample event that can be used to invoke a function to test its code. Add an event name and change Event JSON body, as shown in the figure:



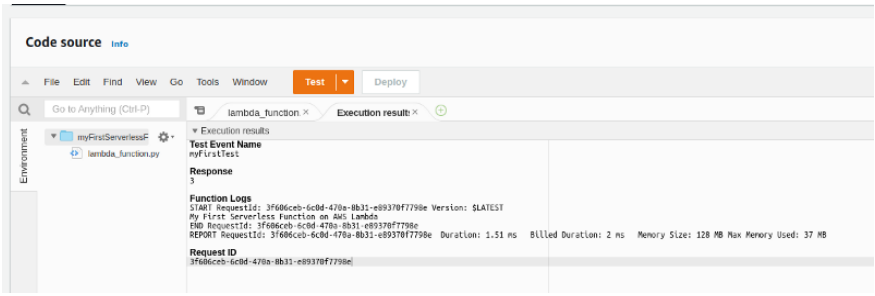
Create a trigger event to test our function.

Now it's time to test it! Click on "Test" button again. Check that the response is 7, as we were expecting. Also, in the function logs, you can see the print that we created.



Function test with trigger event created.

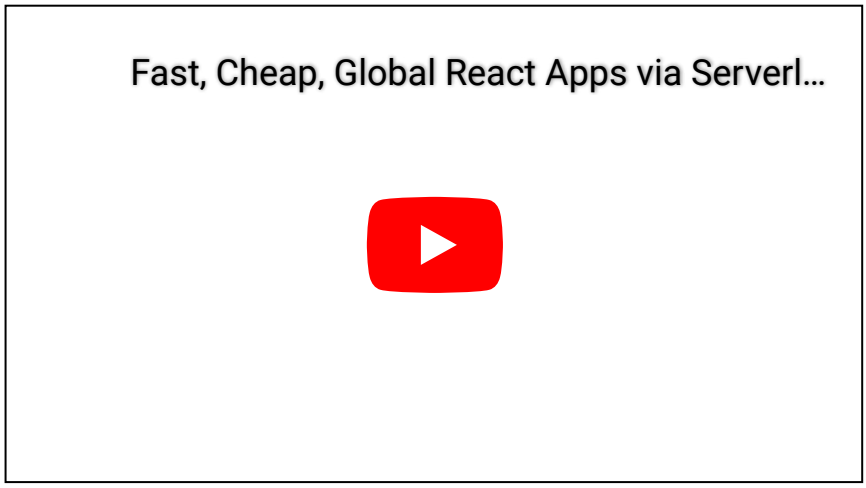
If you change the "operation" parameter in the test event to "sub", you will see that the result will change to 3, as expected.



Second test with "operation" parameter changed to "sub".

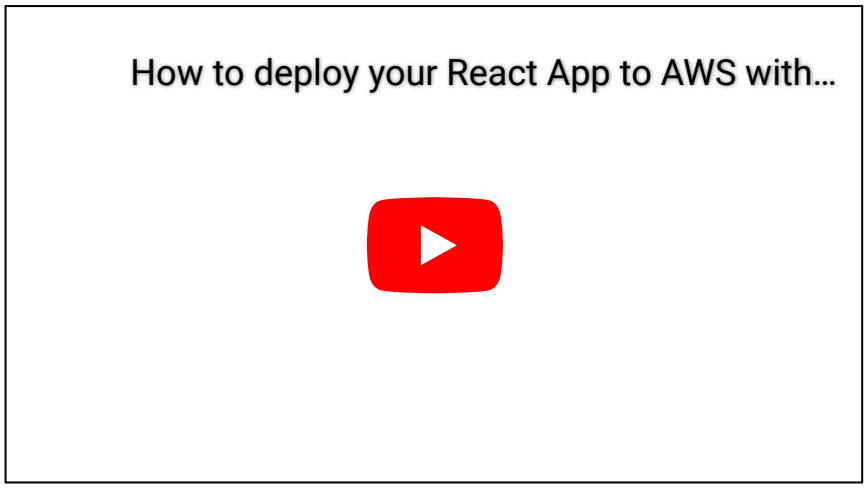
Congratulations! You have created, deployed, and tested your first Serverless function on AWS Lambda.

To dive deeper into Serverless architectures, you can check out the following tutorial on how to deploy a React application through Serverless Framework.



"Fast, Cheap, Global React Apps via Serverless Components" by Serverless.

Finally, a more detailed tutorial on how to deploy a React single page application using Serverless Framework with AWS CloudFront is available.



"How to deploy your React App to AWS with the Serverless Framework" by Complete Coding.

Thanks for reading, I hope you enjoyed it!

Diogo Carvalho