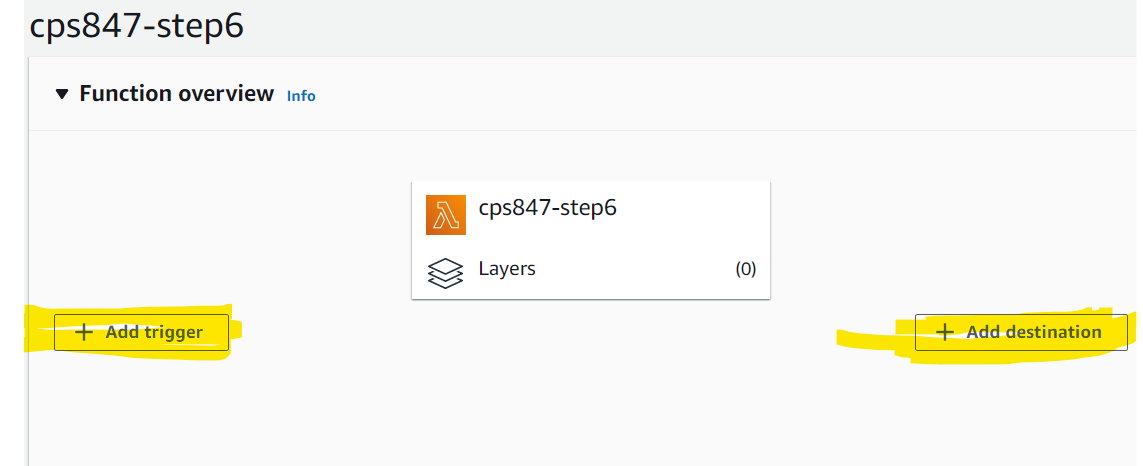
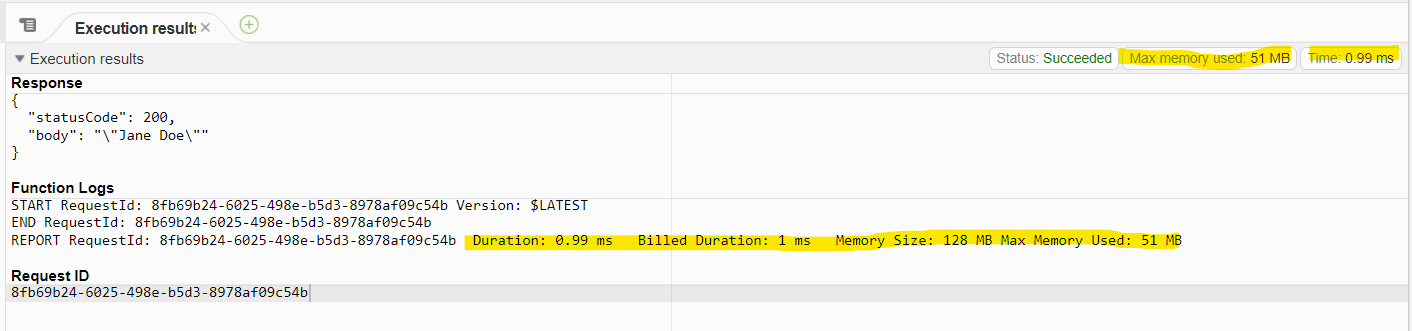
Amazon lambda is described as a event-driven, serverless computing platform.

Event-driven meaning that it can be triggered by outside events such as other requests from other platforms such as Amazon Alexa, or from database changes such as from DynamoDB, to name a few out of a large assortment. This event-driven-ness also allows the lambda to pipe its output after it is done its operations to another process elsewhere such as a git deployment, Amazon S3, or to even another lambda function to do further operations on data.

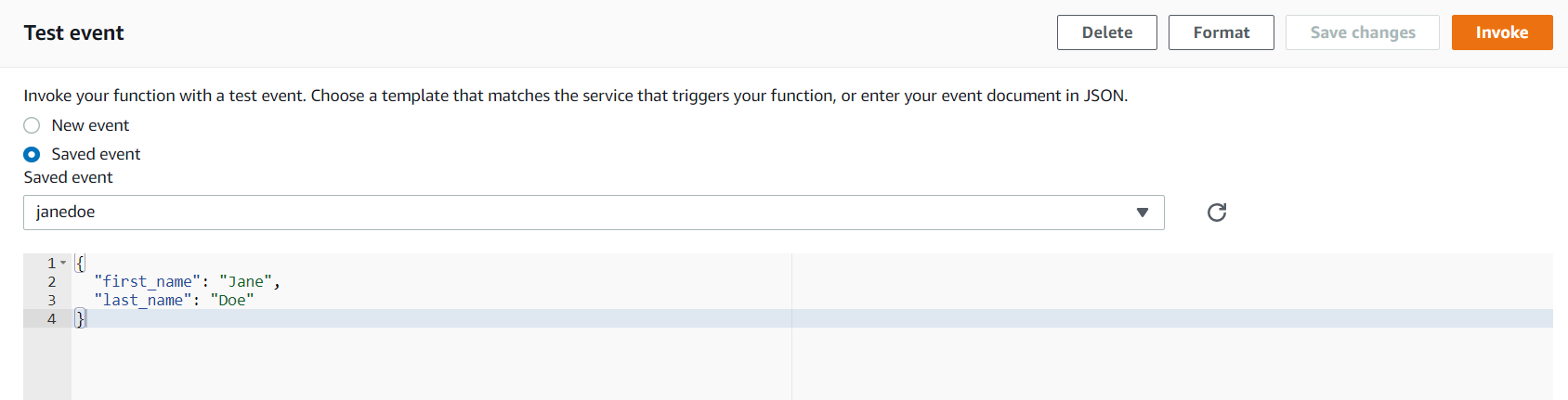


Trigger and destination are the event-driven parts of the lambda function for its input and output and tr

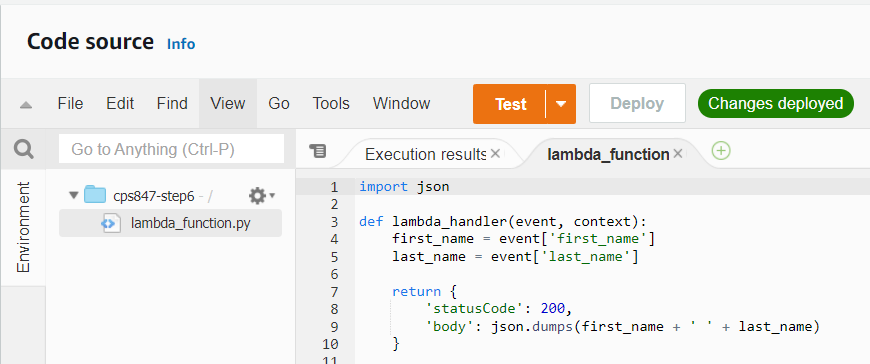
Serverless meaning that the person running the lambda function does not have to utilise their own resources to running the operation which can be costly at times. All AWS Lambda code is run on Amazon’s own servers (therefore serverless to the end-user) which dynamically allocates the resources needed to run the code within the Lambda function.



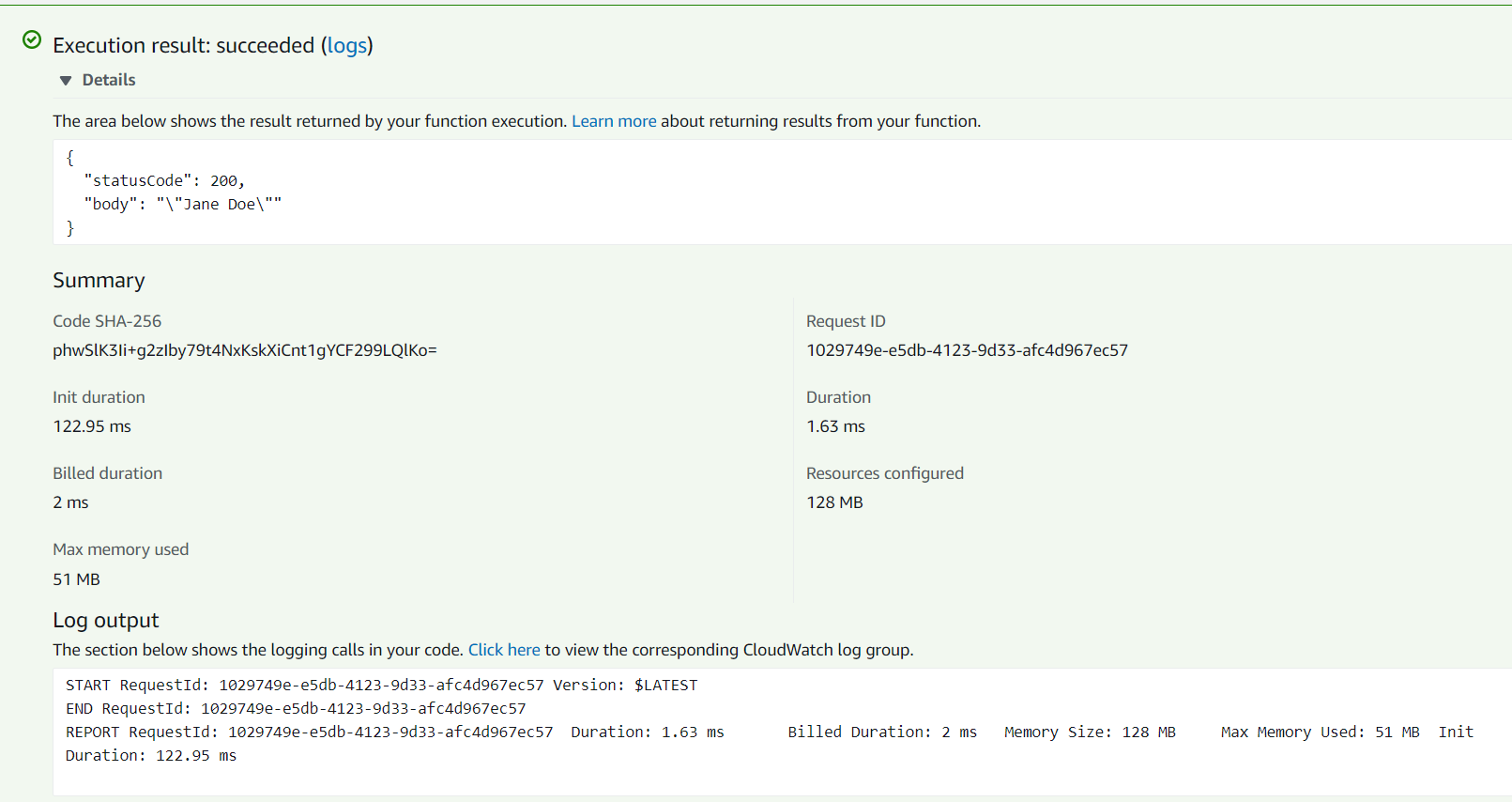
Execution is run on AWS’ platform, their resources such as memory and CPU execution time without need for the end-user to set it.



The test event with the json containing first\_name and last\_name and its respective values. Clicking invoke runs this test case against the created Lambda function.



The Lambda function takes in two parameters, event and context. Event is the json inputted, already turned into a python dict (hashmap in other languages such as Java). Context is a custom LambdaContext class which can be used to determine things such as the function name, the memory limit, the context from the client, while the Lambda is running. Code must be saved by clicking Deploy at the top before it can be tested.



The execution result shown after Deploying the Lambda and clicking Invoke on the test event. The top part shows the response given by the lambda function, 200 as in the HTTP OK response code meaning that there were no errors. The body is the result returned by the Lambda after processing the input using its logic. It also shows other information regarding the execution of the Lambda such as memory and duration as well as any output from the lambda piped to stdout. This output can now be passed onto other functions if need be.