**Question 3- If you need to make your application serverless how it can be done.**

**Answer-**

We can build a serverless web application by using several AWS services together. Each service is fully managed and does not require provision or manage servers. It only need to configure together and upload our application code to AWS Lambda, a serverless compute service. To get started with AWS SAM, use the AWS SAM CLI to create a serverless application that you can package and deploy in the AWS Cloud. You can run the application both in the AWS Cloud or locally on your development host.

Serverless applications are event-driven cloud-based systems where application development rely solely on a combination of third-party services, client-side logic and cloud-hosted remote procedure calls. We can build a serverless web application by using several AWS services together. Each service is fully managed and does not require we to provision or manage servers. We only need to configure them together and upload our application code to AWS Lambda, a serverless compute service.

AWS Lambda, Microsoft Azure Functions, Google Cloud Functions and IBM OpenWhisk are all well-known examples of serverless services offered by the cloud providers.

A **serverless application** is a combination of Lambda functions, event sources, and other resources that work together to perform tasks. Note that a serverless application is more than just a Lambda function—it can include additional resources such as APIs, databases, and event source mappings.

We can use AWS SAM to define our serverless applications. AWS SAM consists of the following components:

* **AWS SAM template specification**. We use this specification to define your serverless application. It provides us with a simple and clean syntax to describe the functions, APIs, permissions, configurations, and events that make up a serverless application. We use an AWS SAM template file to operate on a single, deployable, versioned entity that's your serverless application. For the full AWS SAM template specification, see [AWS Serverless Application Model (AWS SAM) specification](https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/sam-specification.html).

* **AWS SAM command line interface (AWS SAM CLI)**. We use this tool to build serverless applications that are defined by AWS SAM templates. The CLI provides commands that enable us to verify that AWS SAM template files are written according to the specification, invoke Lambda functions locally, step-through debug Lambda functions, package and deploy serverless applications to the AWS Cloud, and so on. For details about how to use the AWS SAM CLI, including the full AWS SAM CLI Command Reference, see [AWS SAM CLI command reference](https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-reference.html#serverless-sam-cli).

In the Amazon Web Services cloud, the execution environment for serverless code is called *AWS Lambda*. It supports a wide array of potential triggers, including incoming HTTP requests, messages from a queue, customer emails, changes to database records, user authentication, messages coming to web sockets, client device synchronization, and much more.

Because application developers do not package or distribute the server code to control a network socket in AWS Lambda, their applications are *serverless*.

**Benefits of serverless Application**[#](https://www.educative.io/courses/running-serverless-applications-aws-lambda/mymvROxZRx9#benefits-of-serverless-architecture)

Compared with running applications in a container cluster or managing virtual machines directly, serverless deployments have two significant​ benefits:

* *Shorter time to market* for new features, leading to faster innovation and delivering value to customers sooner
* *Reduced operational costs* due to better resource utilization