Contents

[Serverless 2](#_Toc16091563)

[Compute 2](#_Toc16091564)

[Storage 2](#_Toc16091565)

[Data stores 3](#_Toc16091566)

[API Proxy 3](#_Toc16091567)

[Application integration 3](#_Toc16091568)

[Orchestration 3](#_Toc16091569)

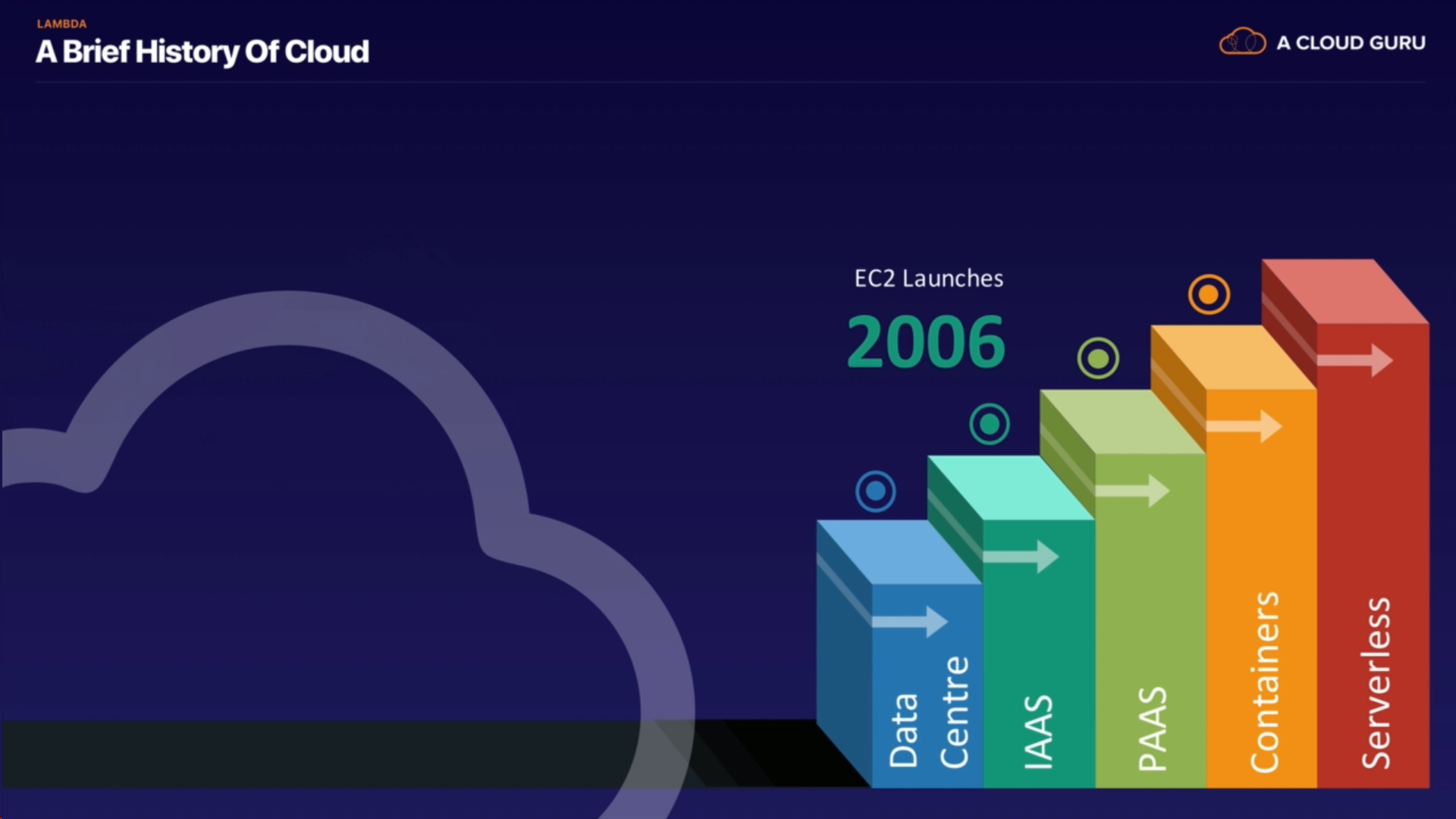
[Analytics 4](#_Toc16091570)

[Developer tooling 4](#_Toc16091571)

[Serverless application use cases 4](#_Toc16091572)

[1. AWS Lambda 6](#_Toc16091573)

# Serverless



<https://aws.amazon.com/serverless/>

## Compute

[AWS Lambda](https://aws.amazon.com/lambda/) lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code is not running.

[Lambda@Edge](https://aws.amazon.com/lambda/edge/) allows you to run Lambda functions at AWS Edge locations in response to Amazon CloudFront events.

[AWS Fargate](https://aws.amazon.com/fargate/) is a purpose-built serverless compute engine for containers. Fargate scales and manages the infrastructure required to run your containers.

## Storage

[Amazon Simple Storage Service](https://aws.amazon.com/s3/) (Amazon S3) provides developers and IT teams with secure, durable, highly-scalable object storage. Amazon S3 is easy to use, with a simple web service interface to store and retrieve any amount of data from anywhere on the web.

[Amazon Elastic File System](https://aws.amazon.com/efs/) (Amazon EFS) provides simple, scalable, elastic file storage. It is built to elastically scale on demand, growing and shrinking automatically as you add and remove files.

## Data stores

[Amazon DynamoDB](https://aws.amazon.com/dynamodb/) is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale.

[Amazon Aurora Serverless](https://aws.amazon.com/rds/aurora/serverless/) is an on-demand, auto-scaling configuration for [Amazon Aurora](https://aws.amazon.com/rds/aurora/) (MySQL-compatible edition), where the database will automatically start up, shut down, and scale capacity up or down based on your application's needs.

## API Proxy

[Amazon API Gateway](https://aws.amazon.com/api-gateway/) is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. It offers a comprehensive platform for [API management](https://aws.amazon.com/api-gateway/api-management/). API Gateway allows you to process hundreds of thousands of concurrent API calls and handles traffic management, authorization and access control, monitoring, and API version management.

## Application integration

[Amazon SNS](https://aws.amazon.com/sns/) is a fully managed pub/sub messaging service that makes it easy to decouple and scale microservices, distributed systems, and serverless applications.

[Amazon SQS](https://aws.amazon.com/sqs/) is a fully managed message queuing service that makes it easy to decouple and scale microservices, distributed systems, and serverless applications.

[AWS AppSync](https://aws.amazon.com/appsync/) simplifies application development by letting you create a flexible GraphQL API to securely access, manipulate, and combine data from one or more data sources.

[Amazon EventBridge](https://aws.amazon.com/eventbridge/) is a serverless event bus service that makes it easy to access application data from a variety of sources and send it into your AWS environment.

## Orchestration

[AWS Step Functions](https://aws.amazon.com/step-functions/) makes it easy to coordinate the components of distributed applications and microservices using visual workflows. Building applications from individual components that each perform a discrete function lets you scale and change applications quickly. Step Functions is a reliable way to coordinate components and step through the functions of your application.

## Analytics

[Amazon Kinesis](https://aws.amazon.com/kinesis/) is a platform for streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data, and also providing the ability for you to build custom streaming data applications for specialized needs.

[Amazon Athena](https://aws.amazon.com/athena/) is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.

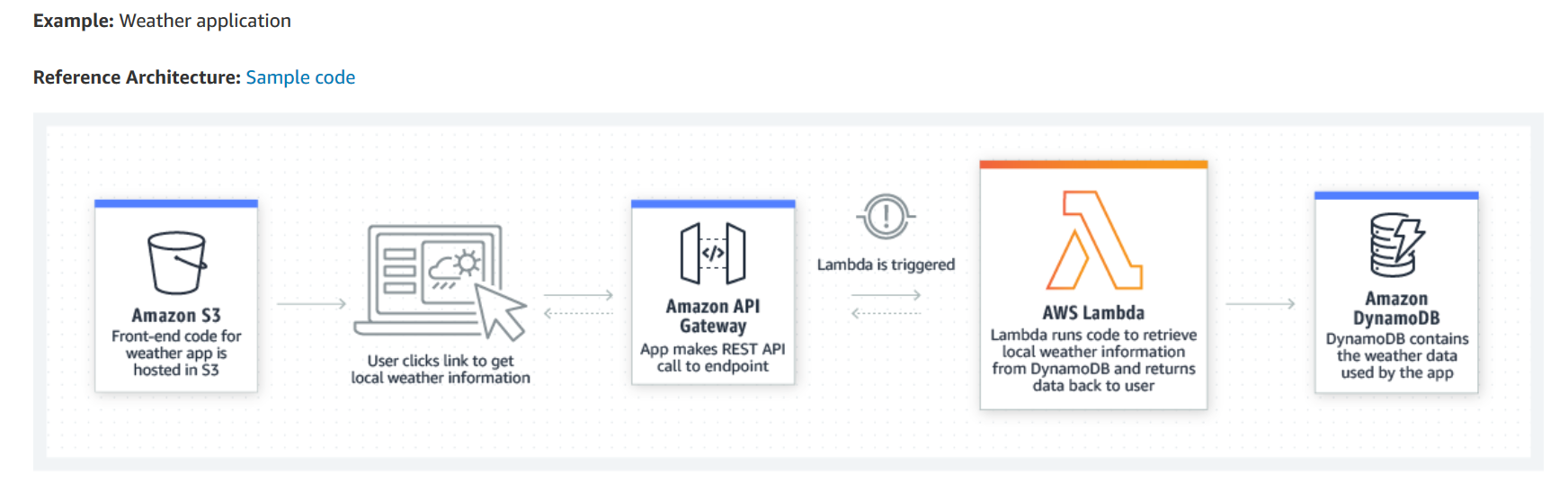
## Developer tooling

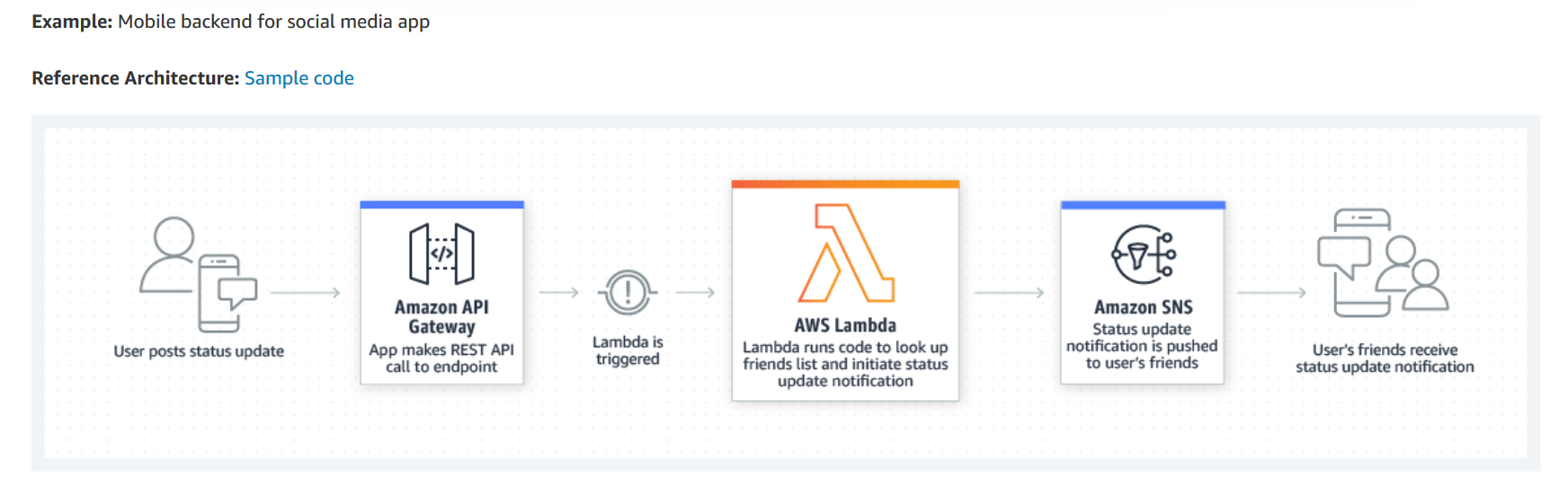
AWS provides [tools and services](https://aws.amazon.com/serverless/developer-tools/) that aid developers in the serverless application development process. AWS and its partner ecosystem offer tools for continuous integration and delivery, testing, deployments, monitoring and diagnostics, SDKs, frameworks, and integrated development environment (IDE) plugins.

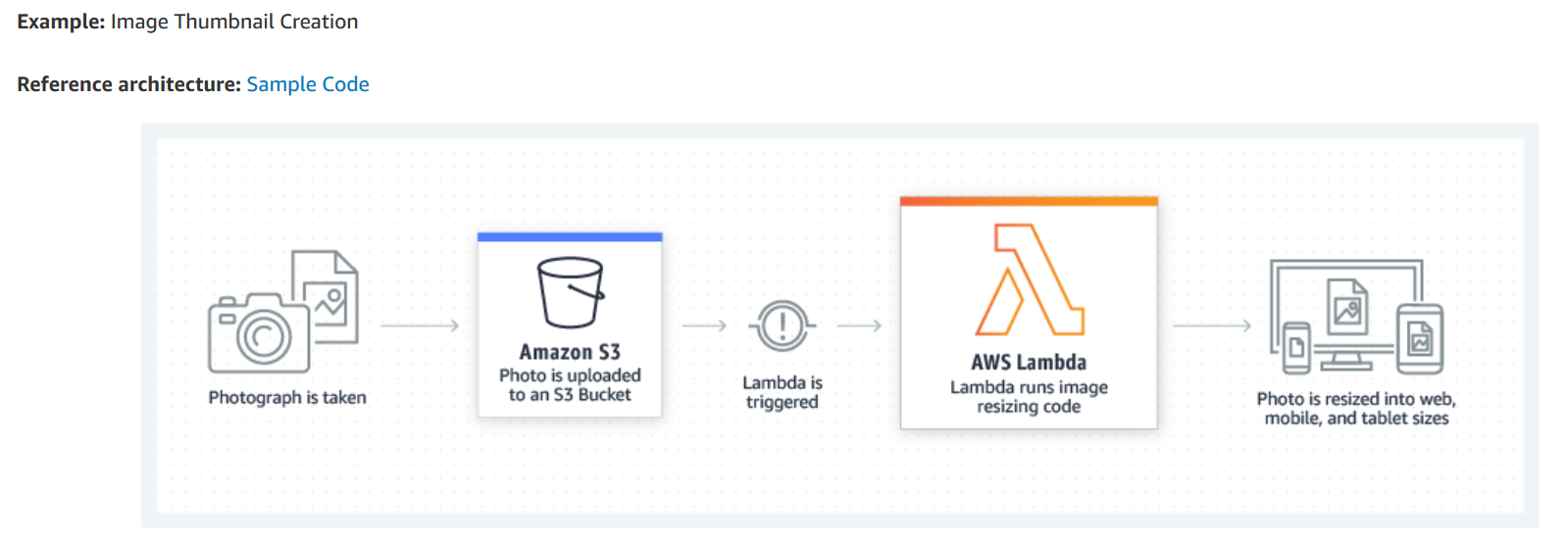
To Check ?

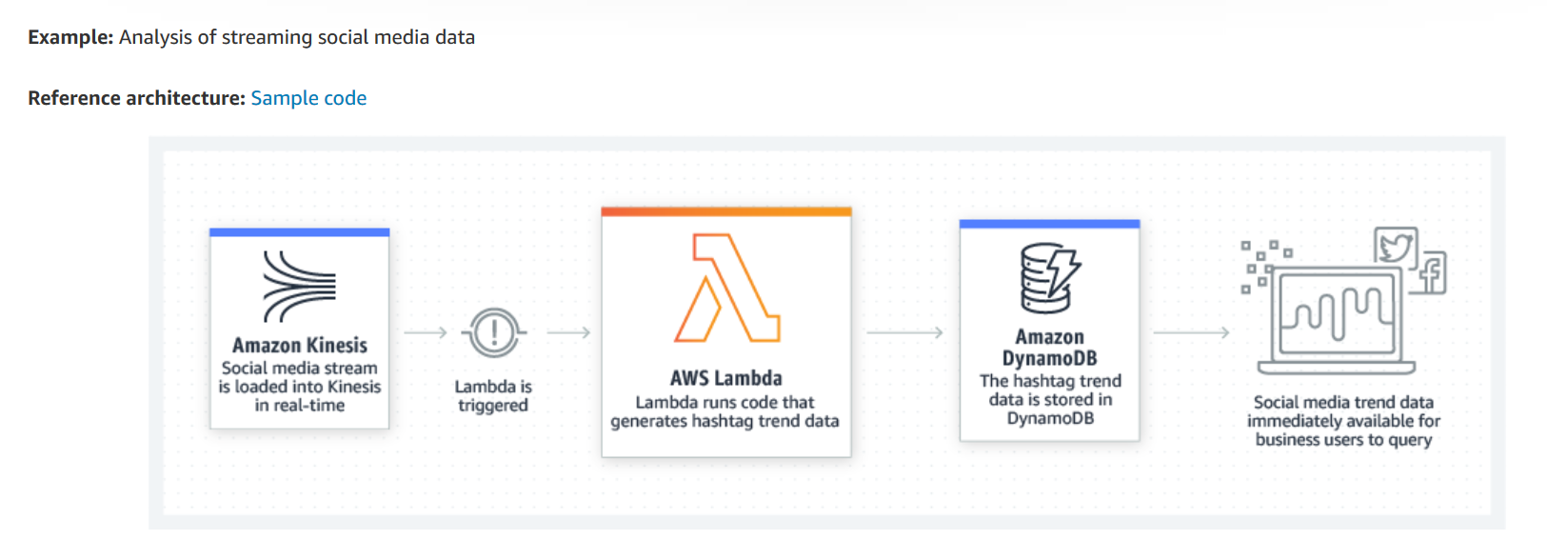
serverless Spark : Glue

# Serverless application use cases

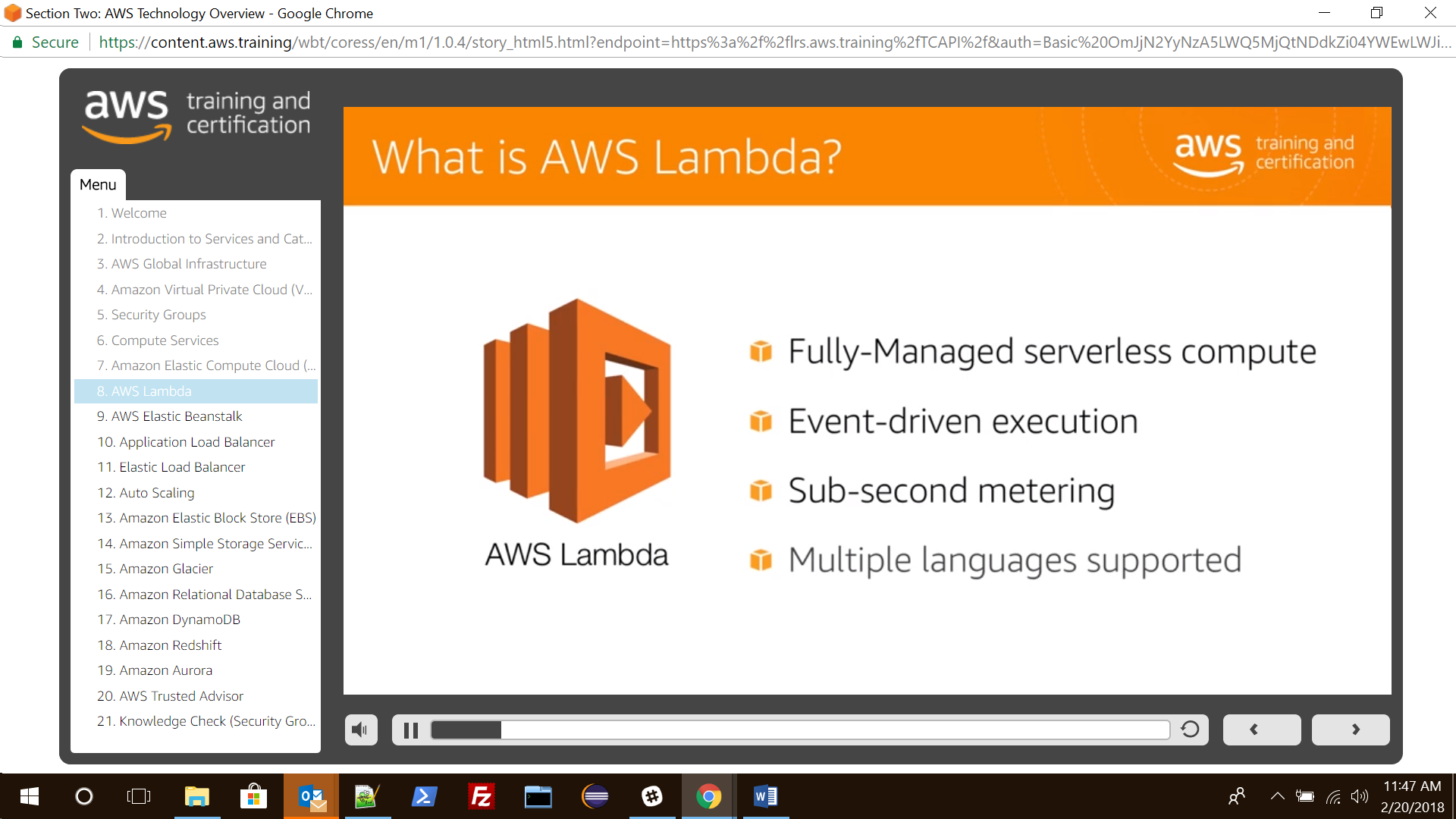


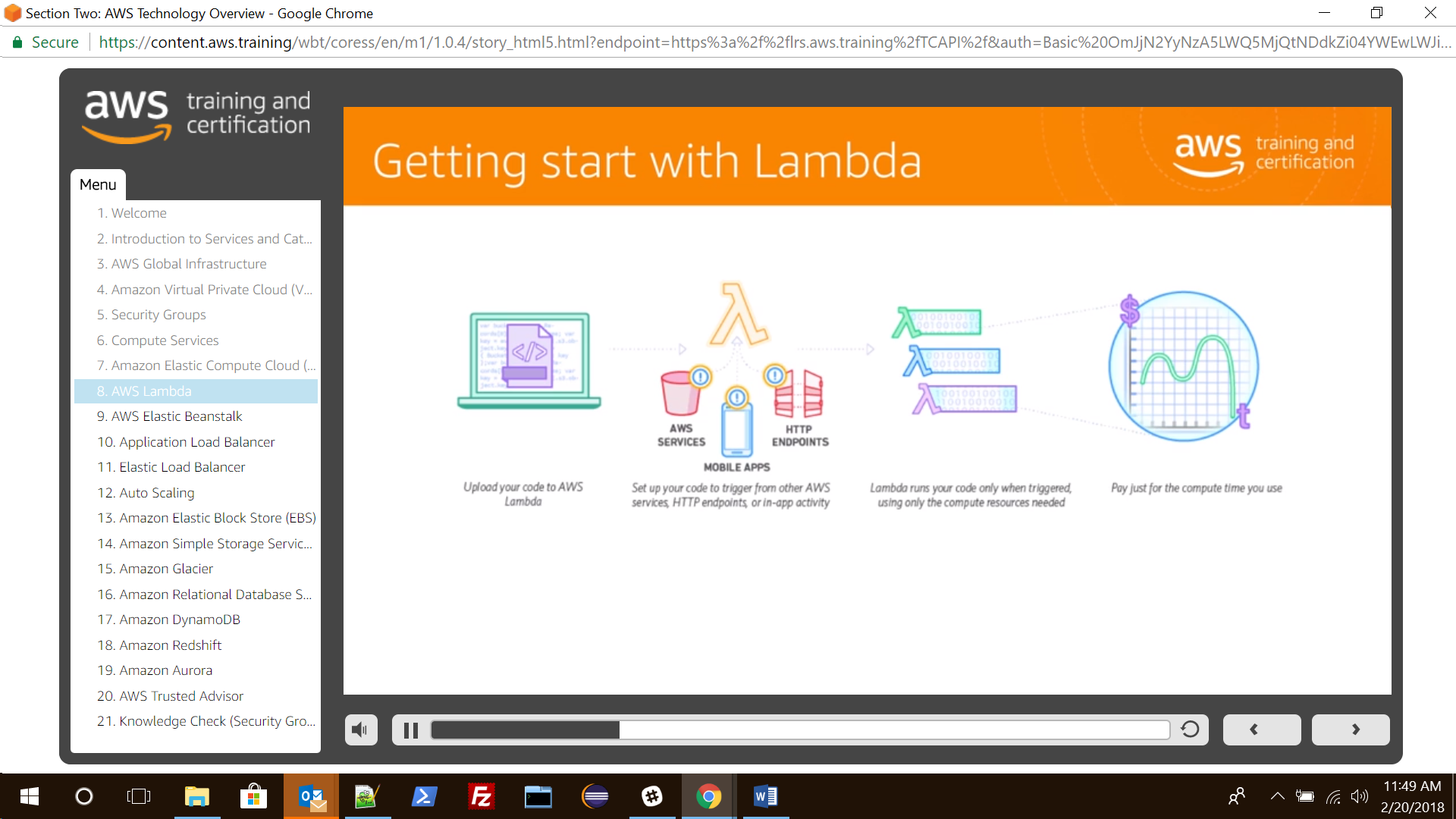






# AWS Lambda

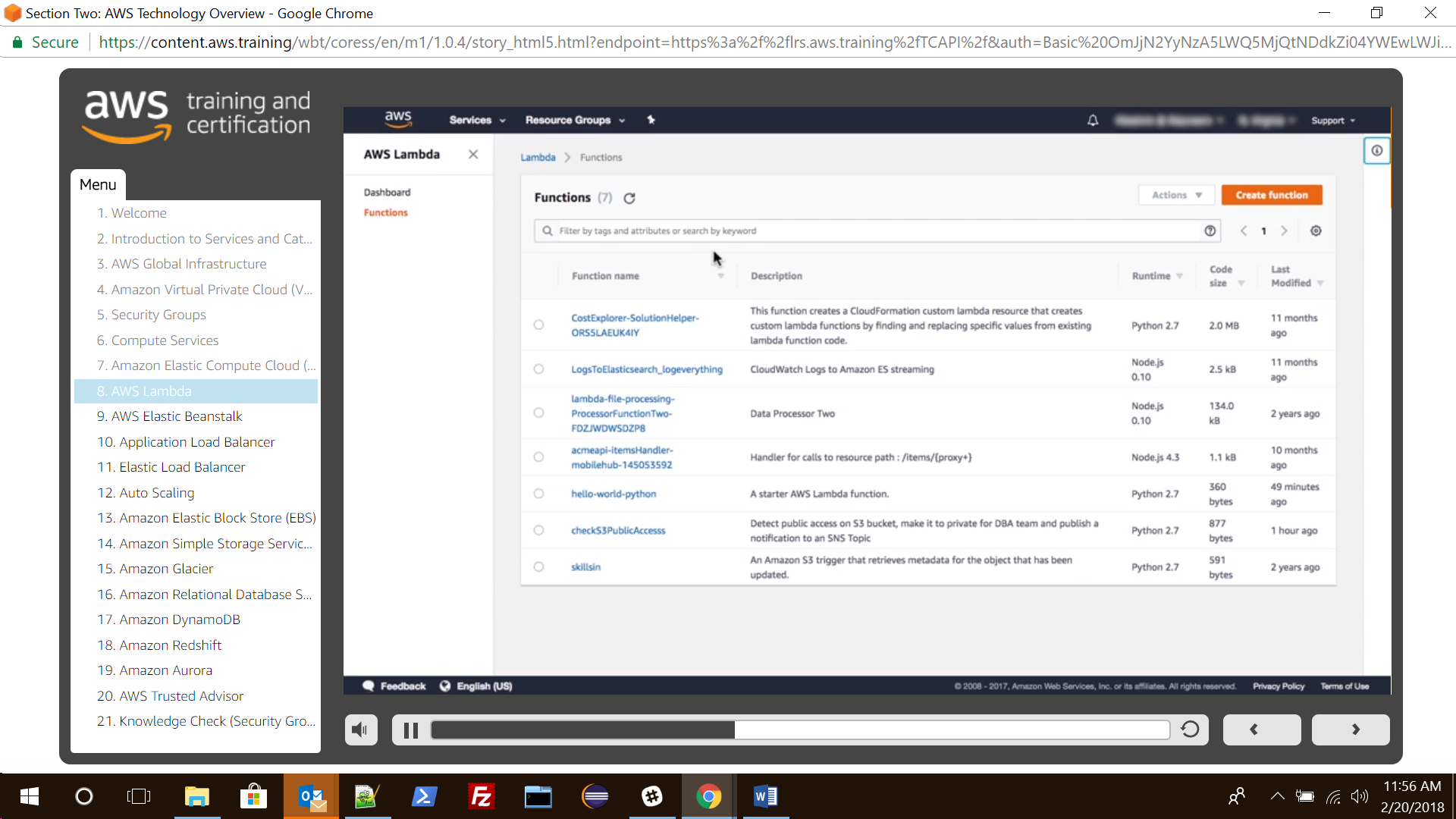


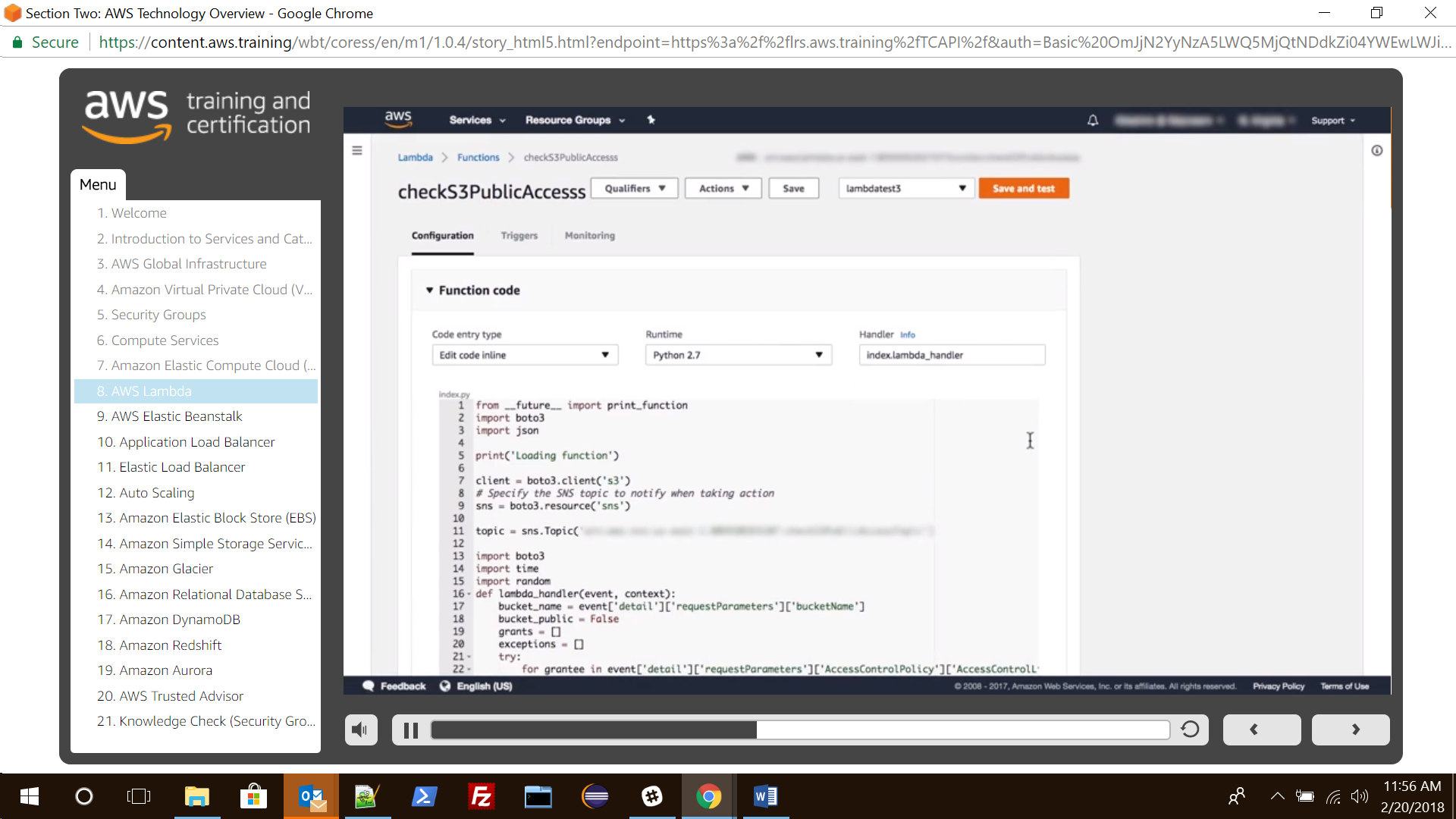


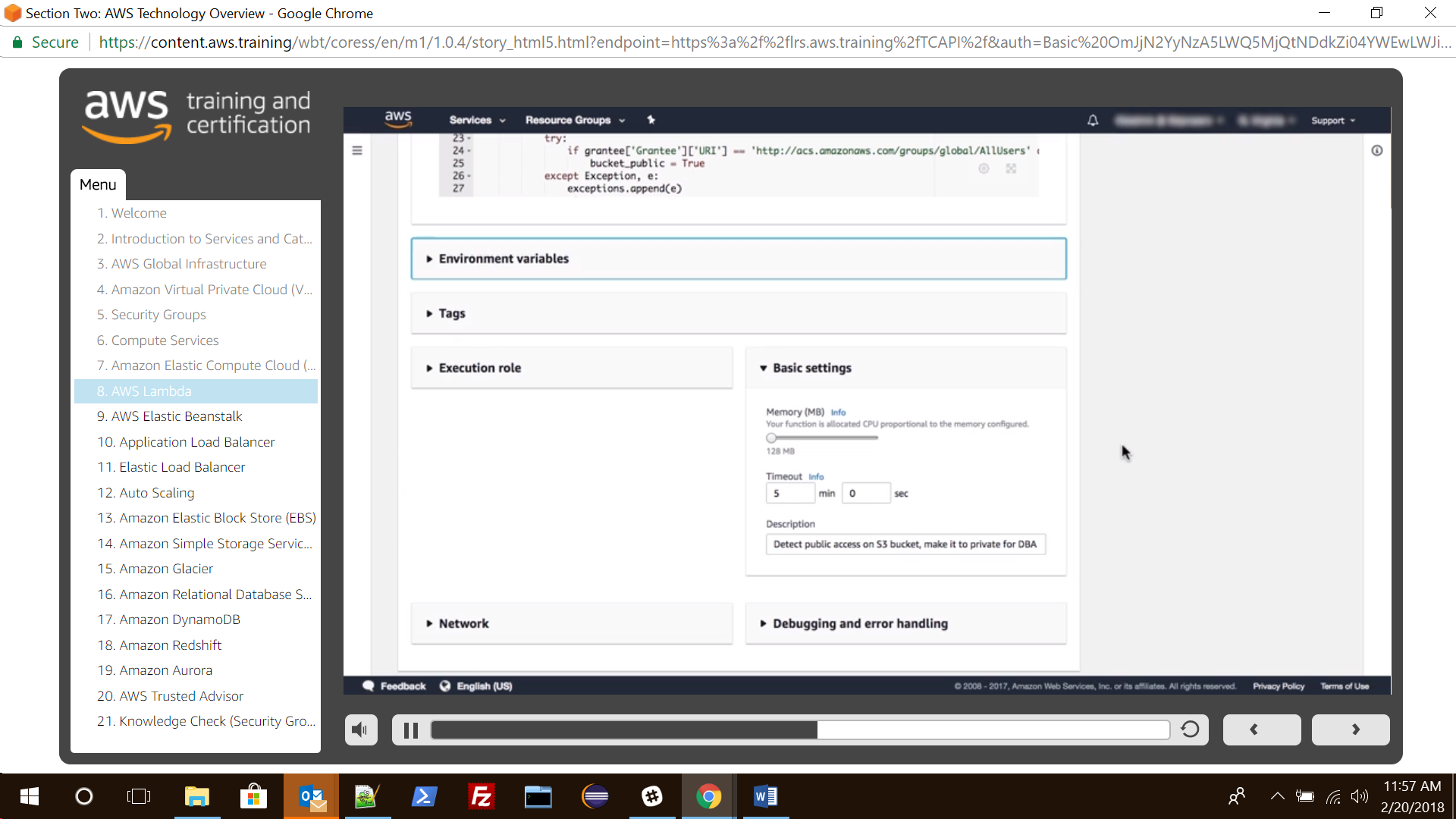
There are limitations based on

1. Memory size
2. package size
3. payload size
4. concurrent execution
5. Timeout
6. Etc
7. etc

Create a Lambda function







Set trigger to launch Lambda function. Here it is using **Cloud Watch Events** to trigger the events.

