
Serverless Computing

Cloud Computing
Brenden west

Contents

Learning Outcomes

- Overview of Serverless Computing
- Using AWS Lambda

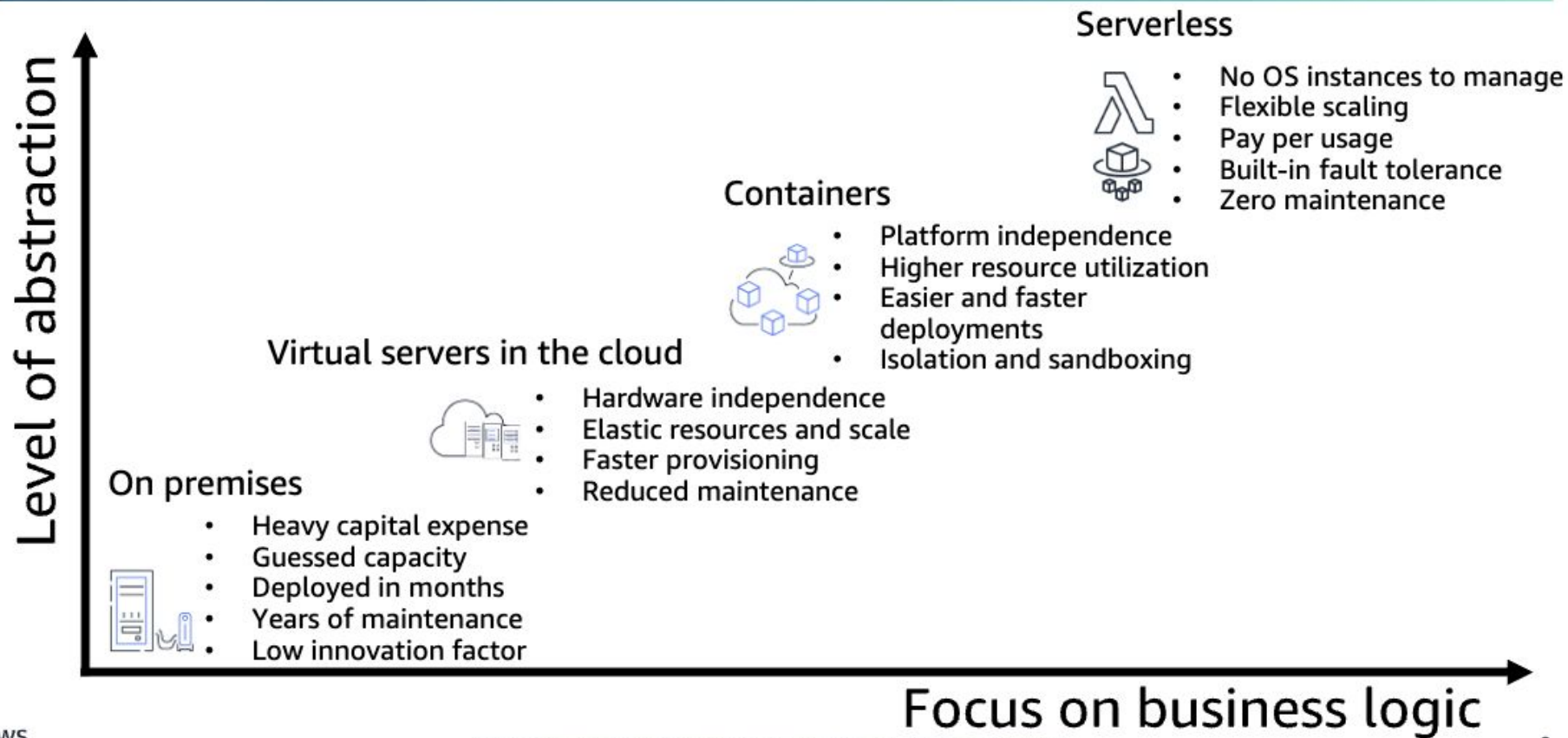
Reading

- AWS Cloud Foundations - Module 6, Sec. 7 - Intro to AWS Lambda
- AWS Cloud Developing - Module 7 - Serverless Computing
- <https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-concepts.html>
- <https://www.cloudzero.com/blog/aws-lambda/>

What is Serverless Computing

- Small, purpose-built services (aka **microservices**), loosely coupled
- Managed services where provider handles all infrastructure tasks
- **Pay-for-use** billing - costs incurred only when code is running
- Code only runs when needed
- Allows for **asynchronous, event-driven** system design patterns
- Individual services deploy & scale independently, allowing rapid development and flexibility
- eliminates infrastructure management tasks, such as server, cluster, or capacity provisioning, patching, operating system maintenance, etc.

Serverless computing



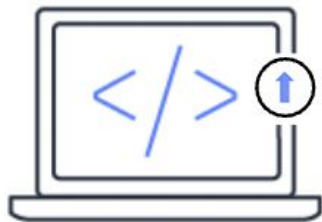
Microservices

- Small pieces, loosely coupled
- Purpose-built data stores
- Allow different system components to deploy & scale independently
- Allow use of different, optimal technologies for each component

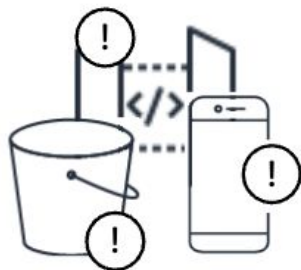
Intro to AWS Lambda

- Serverless compute service to run code without provisioning or managing infrastructure
- Invokes code (**functions**) in response to **events** or incoming HTTP requests, events in a stream or queue, or on a schedule.
- Supports a range of programming languages, custom runtimes & libraries
- Scales automatically based on demand
- Integrates with other AWS services for full application capabilities
- Uses AWS IAM for fine-grained access control
- Has built-in code monitoring & logging with Amazon CloudWatch
- Can be invoked synchronous or asynchronous

How Lambda works



Upload your code to Lambda or write code in the Lambda editor



Set up your code to run when events occur in other AWS services, at HTTP endpoints, or as part of in-app activity



Lambda runs your code only when it is activated by an event and uses only the compute resources that are needed



Pay only for the compute time that you use

AWS Lambda Concurrency

- Each invocation of a Lambda function runs in an **ephemeral** environment that is initialized on demand, lasts a brief time, and then is removed.
- Lambda manages creating and tearing down these environments. You have no direct control over them.
- **Concurrency** is the number of Lambda function invocations that are running at one time.
- Factors affecting concurrency include:
 - Reserved concurrency
 - Regional quota
 - Burst quota
 - Request rate & function duration
 - Event source

AWS Lambda Pricing

- Number of requests served
- Time required to execute code for those requests
- Amount of memory allocated to the functions
- Data transfer

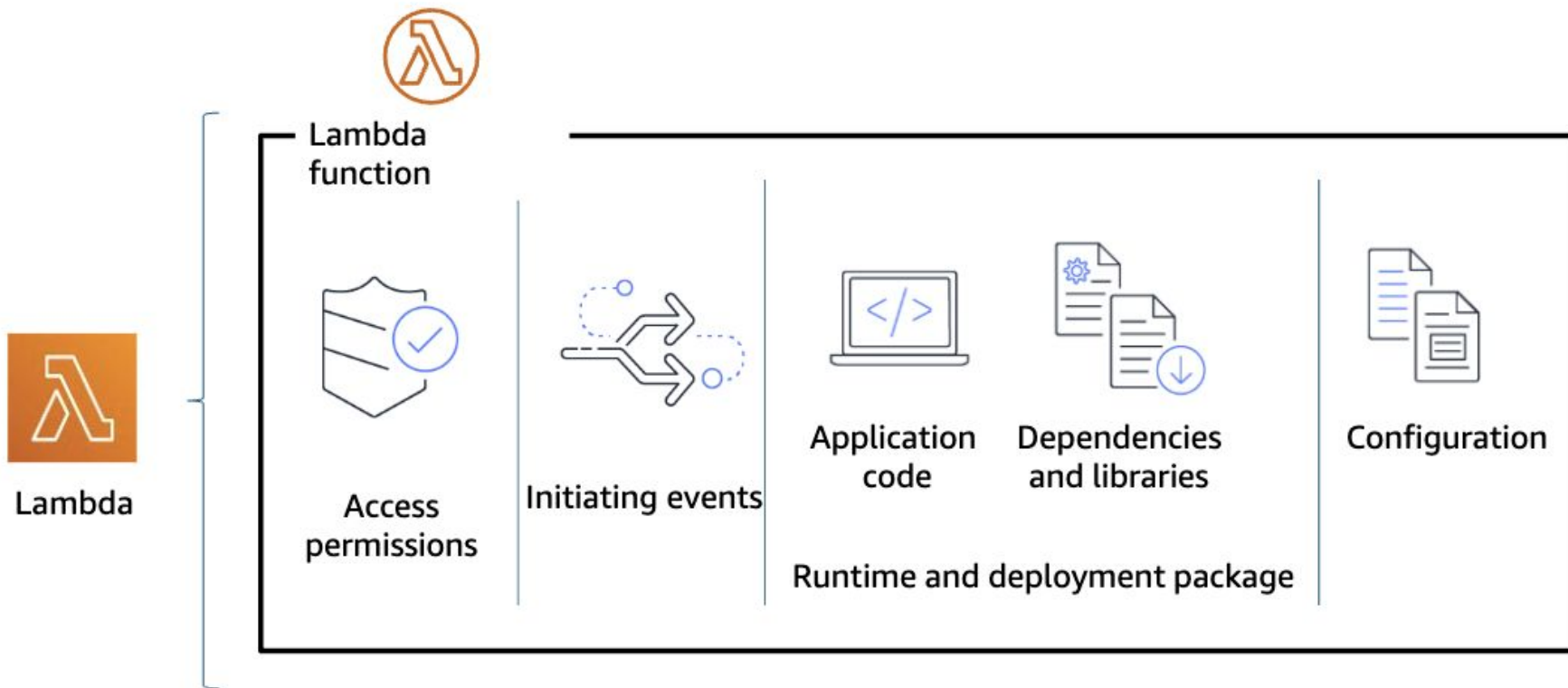
AWS Lambda Caveats

- AWS Lambda shuts down functions not used in > 15 minutes
- Functions have a **cold-start** latency that can vary based on price-tier, language and code size
- Long-running workloads can get shutdown after 15 minutes
- Per-region account limit on number of concurrent function executions
- Can get expensive quickly if costs are not optimized

Creating a Lambda Function

- Define permissions
- Specify events to invoke the function
- Specify invocation type (synchronous or asynchronous) if allowed
- Choose a runtime (language)
- Create deployment package (code & dependencies)
- Configure running parameters - memory, timeout, concurrency

Lambda functions



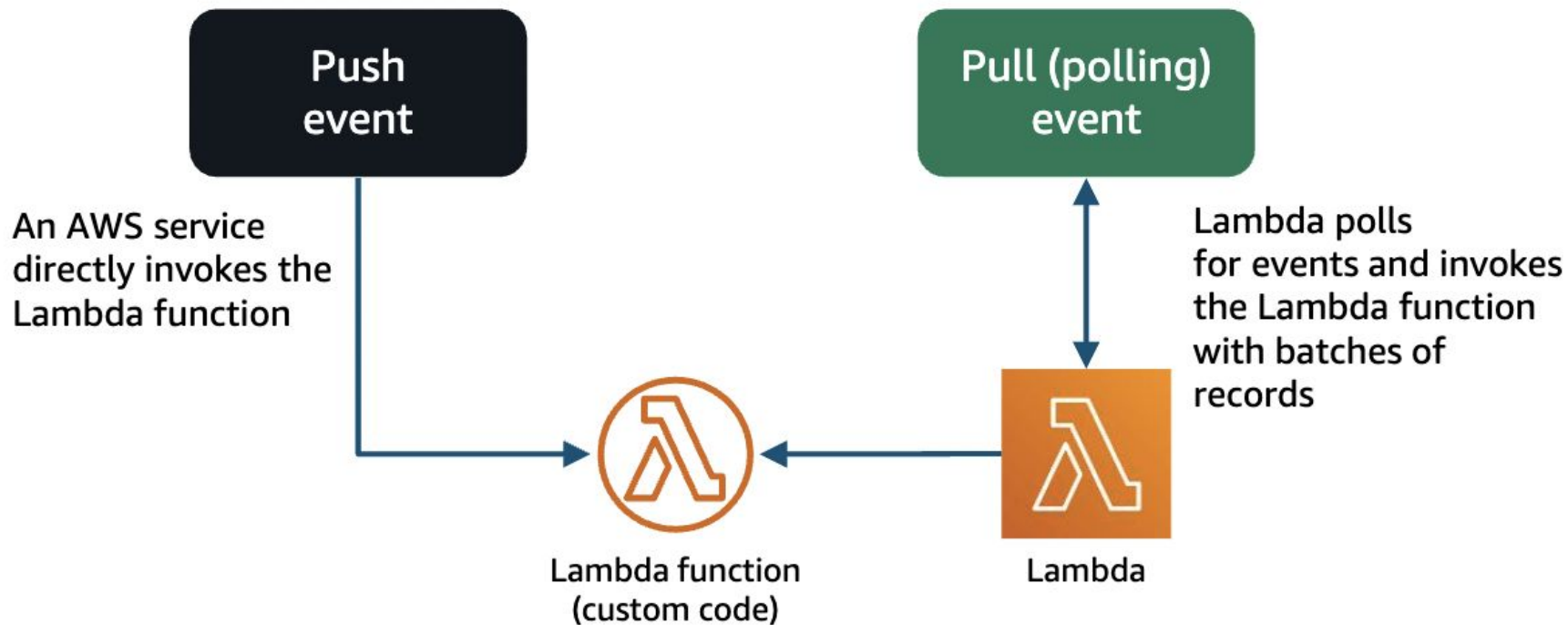
Invoking Lambda Functions

Model

- **Push model** - An event source directly invokes the Lambda function.
 - **Synchronous** - event source waits for a response
 - **Asynchronous** - Lambda queues event & responds to source independent of when function is invoked. Can retry & send failed events to a dedicated queue
- **Pull (or polling) model** - Lambda polls a stream or queue and invokes a Lambda function when it detects an event.

Invocation type is pre-determined by AWS service that triggers an event

Push and pull models



Other Cloud Function Options

- <https://cloud.google.com/functions>
- <https://www.cloudflare.com/developer-platform/workers/>