Serverless Computing

Cloud Computing

Brenden west

Contents

Learning Outcomes

- Overview of Serverless Computing
- Using AWS Lambda

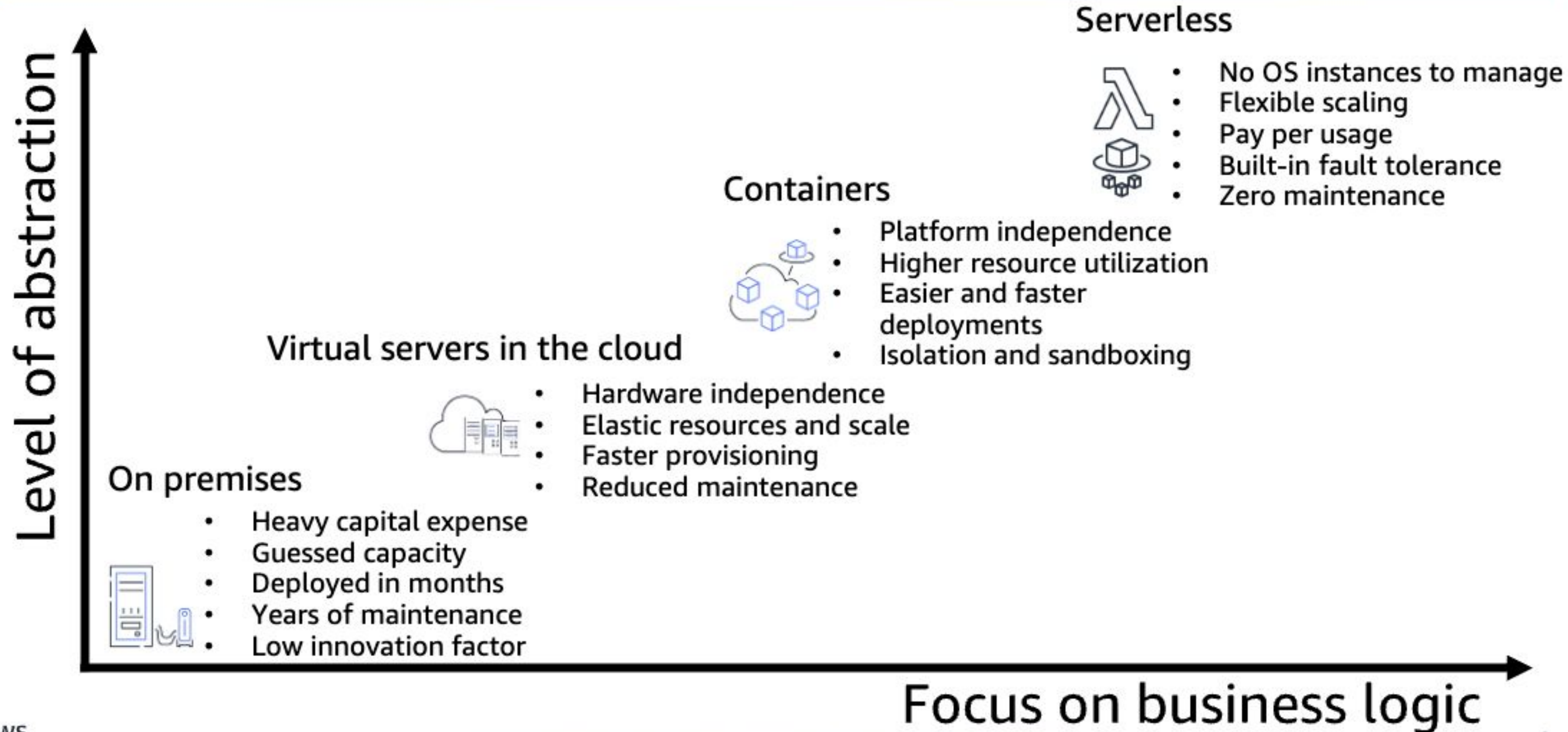
Reading

- AWS Cloud Foundations - Module 6, Sec. 7 - Intro to AWS Lambda
- AWS Cloud Developing - Module 7 - Serverless Computing
- https://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

Serverless computing



On premises

- Heavy capital expense
- Guessed capacity
- Deployed in months
- Years of maintenance
- Low innovation factor

Virtual servers in the cloud

- Hardware independence
- Elastic resources and scale
- Faster provisioning
- Reduced maintenance

Containers

- Platform independence
- Higher resource utilization
- Easier and faster deployments
- Isolation and sandboxing

Serverless

- No OS instances to manage
- Flexible scaling
- Pay per usage
- Built-in fault tolerance
- Zero maintenance

Intro to AWS Lambda

- Serverless compute service to run code without provisioning or managing services
- 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
- 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

AWS Lambda Pricing

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

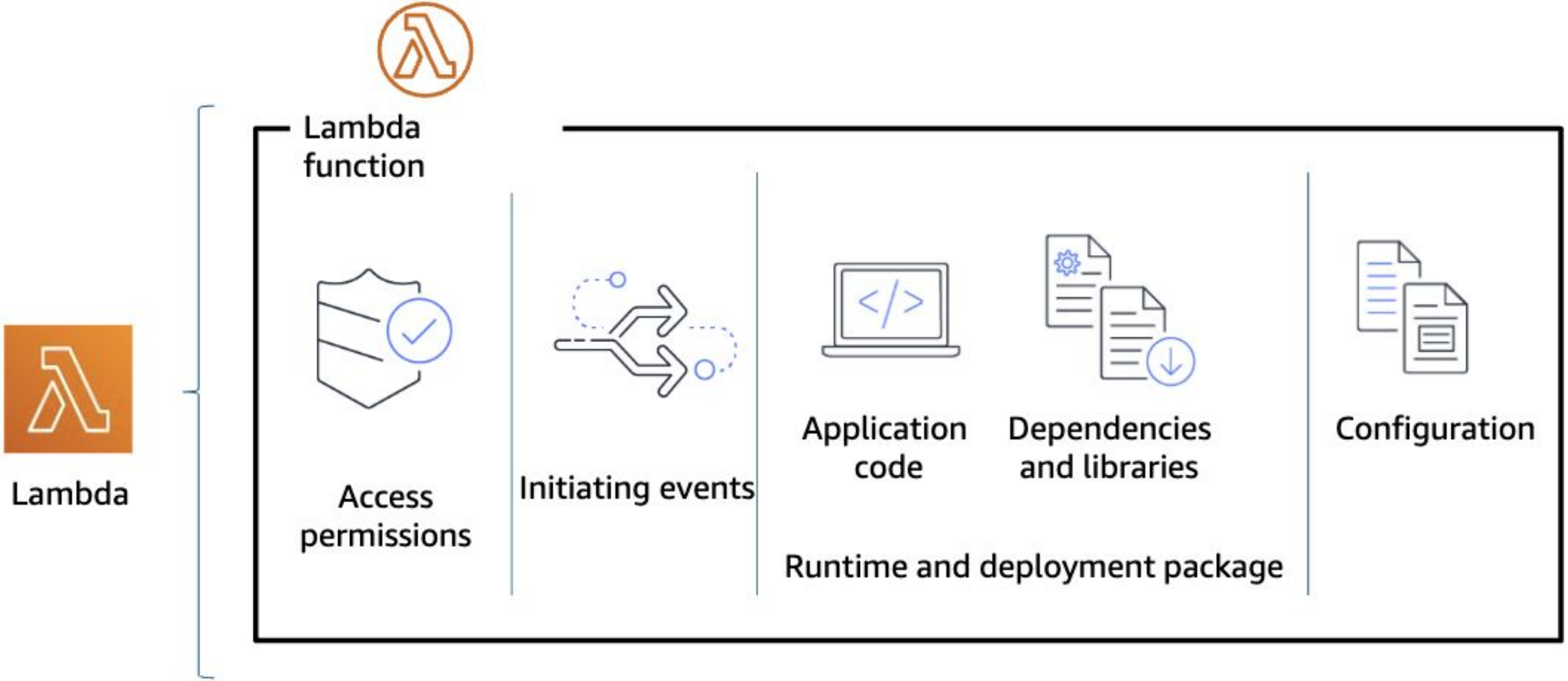
AWS Lambda Caveats

- 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

Other Cloud Function Options

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