DĚLAT
DOBRÝ SOFTWARE
NÁS BAVÍ

PROFINIT

B0M33BDT

Serverless

Marek Sušický

7. 12. 2022

Agenda

- Intro
- Comparison
- How to use it
- AWS
- Azure
- Summary

PROFINIT

Serverless architecture - definition

- Serverless architecture is an approach to software design that allows developers to build and run services without having to manage the underlying infrastructure. Developers can write and deploy code, while a cloud provider provisions servers to run their applications, databases, and storage systems at any scale.

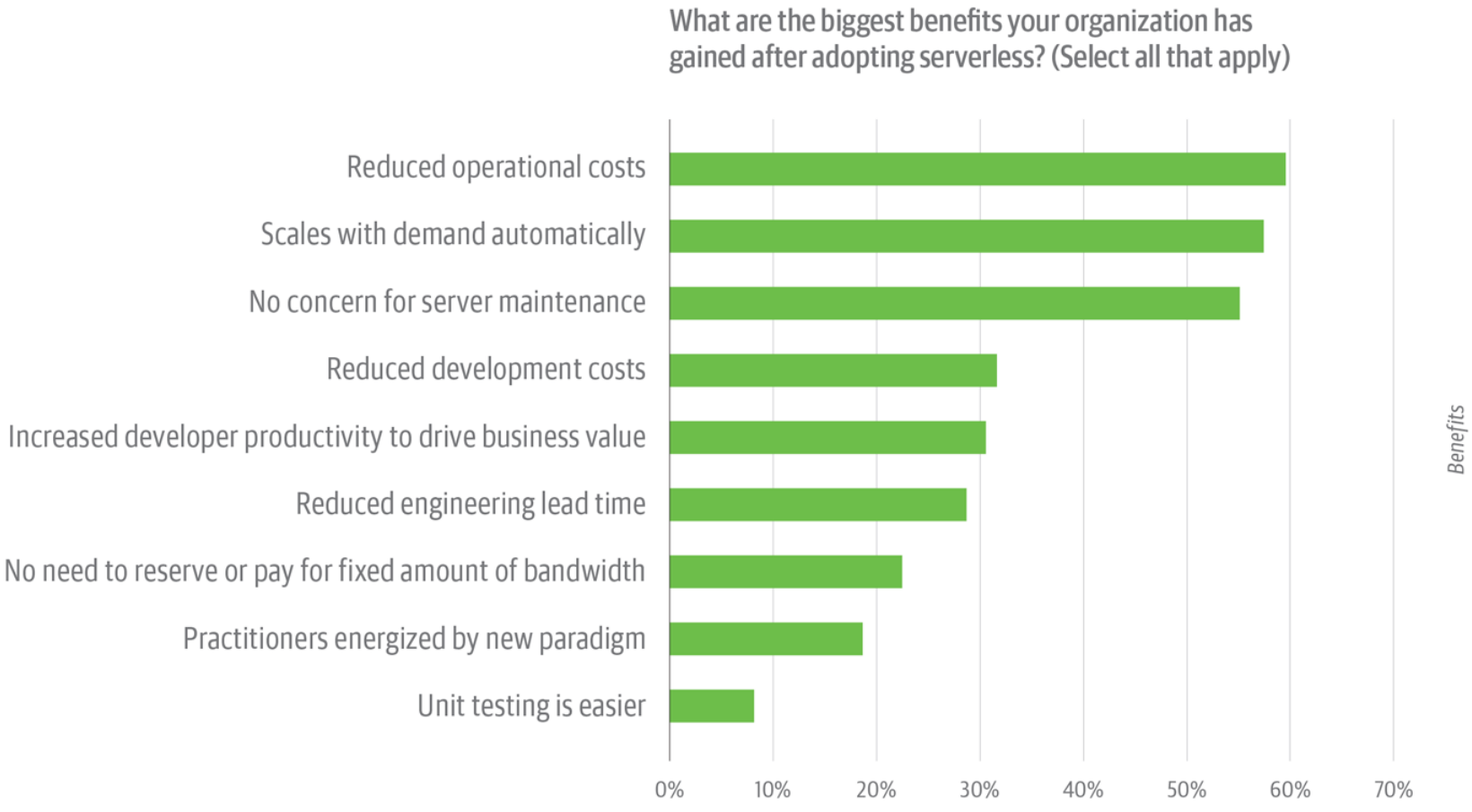
- Servers allow users to communicate with an application and access its business logic, but managing servers takes considerable time and resources. Teams have to maintain the server hardware, take care of software and security updates, and create backups in case of failure. By adopting serverless architecture, developers can offload these responsibilities to a third-party provider, enabling them to focus on writing application code.

Serverless architecture

- One of the most popular serverless architectures is Function as a Service (FaaS), where developers write their application code as a set of discrete functions.
- When a function is invoked, the cloud provider either executes the function on a running server, or, if there is no server currently running, it spins up a new server to execute the function.

Why?

Survey – Biggest benefits of serverless



PROFINIT

Main benefits - costs

- Imagine Hadoop cluster
  - Dozens of servers
  - HW issues
  - Hosting issues
  - OS issues
    - Updates/patches
  - SW issues
    - Updates/patches
  - Application issues
    - Firewall
    - Backups
    - HA
    - Deployment
- Pay OPS team… (and what about 24/7…)

PROFINIT

Main benefits

- With serverless you can be really focused on your business
- No traffic – no costs
- Scalability
- Productivity - simple deployment

- But you have to pay for it (someone else should do it for you)
- It is not under your control – risk
- Long running jobs can be expensive or impossible
- Security – shared HW
- Complicated testing – integration tests
- You have to study the vendor conditions and limits
- Vendor lock-in

PROFINIT

Serverless vs Containers

- Containers should be updated
- Containers should be scaled with 3rd party app like Kubernetes

- Containers – better control of the environment, good for high traffic applications
- Serverless – better for triggered based events

PROFINIT

Use cases

- Triggered base actions
  - When user sign in
  - Filled application for a loan - scoring
- RESTful API
  - With Amazon API Gateway – will scale up automatically
- Asynchronous processing
  - Processing video on the background etc.
- Security checks
  - When new container is started – check for the vulnerabilities
- CI/CD
  - Push will trigger build
  - Pull request can start tests etc.

PROFINIT

Lifecycle and history

- 1. Upload function
- 2. Define trigger
- 3. Run your function

- 2014 - AWS Lambda
- 2016 - Google Cloud Functions
- 2016 - Azure Functions

PROFINIT

Crucial terms

- **Invocation**
- A single function execution.
- **Duration**
- The time it takes for a serverless function to execute.
- **Cold Start**
- The latency that occurs when a function is triggered for the first time or after a period of inactivity.
- **Concurrency Limit**
- The number of function instances that can run simultaneously in one region, as determined by the cloud provider. A function will be throttled if it exceeds this limit.
- **Timeout**
- The amount of time that a cloud provider allows a function to run before terminating it. Most providers set a default timeout and a maximum timeout.

Cold start

- Initialization of a Lambda function will either be a “**warm start**”—reusing an instance of a Lambda function and its host container from a previous event—or a “**cold start**” —creating a new container instance
- Latency depends on many factors
- Pinging the service to stay in cache – cleaned after some minutes of inactivity

Serverless databases

Serverless databases

- In the last years – easy way how to fetch data from your functions

- Nutanix – switching RDBMS to serverless – Oracle, MariaDB, PostgreSQL, MSSQL…
- Amazon Aurora
- Azure Data Lake
- Google Firebase

PROFINIT

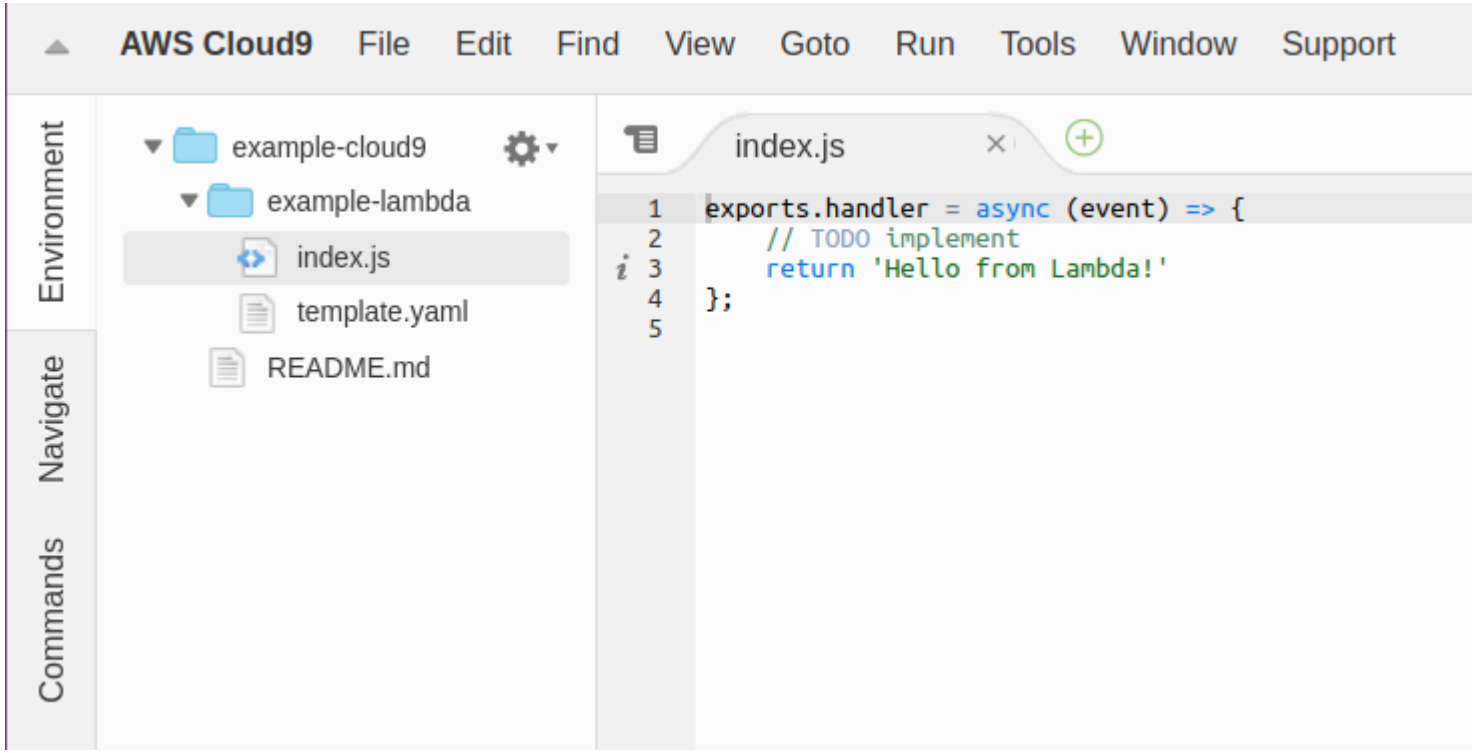
AWS

AWS Lambda

- Event-driven, serverless computing platform provided by Amazon
- Computing service that runs code in response to events and automatically manages the computing resources required by that code
- Introduced 2014

- Node.js, Python, Java, Go, Ruby, and C#

AWS Lambda



PROFINIT

AWS Step function

- AWS Step Functions is a serverless orchestration service that lets you integrate with AWS Lambda functions

- Step Functions is based on state machines and tasks
- A state machine is a workflow
- A task is a state in a workflow that represents a single unit of work that another AWS service performs
- Each step in a workflow is a state

- Not the only solution

PROFINIT

AWS Step function – workflow types

- Standard
  - 2,000 per second execution rate
  - 4,000 per second state transition rate
  - Priced by state transition
  - Show execution history and visual debugging
  - Support all service integrations and patterns

- Express
  - 100,000 per second execution rate
  - Nearly unlimited state transition rate
  - Priced by number and duration of executions
  - Send execution history to Cloud Watch

PROFINIT

AWS Step function – pricing

- Free Tier
  - 4,000 state transitions
- $0.025 per 1,000 state transitions

PROFINIT

AWS Glue

- Serverless data integration service

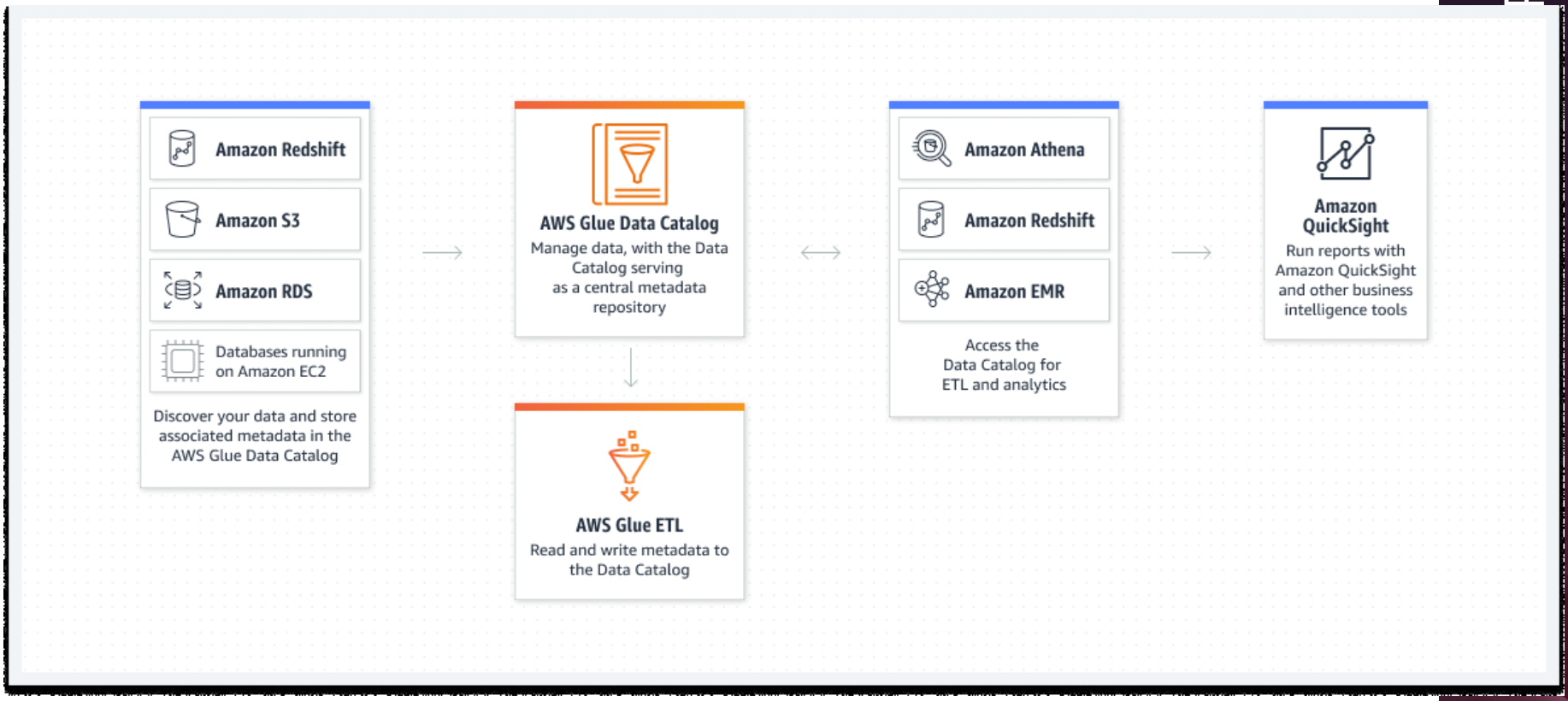
- Multiple use cases (not all)
  - Event-driven ETL
  - Data catalog
  - No-code ETL
  - Data quality

- Various pricing – depends on components
  - Monthly fee
  - Pay per seconds
  - Pay per session
  - Pay per minute

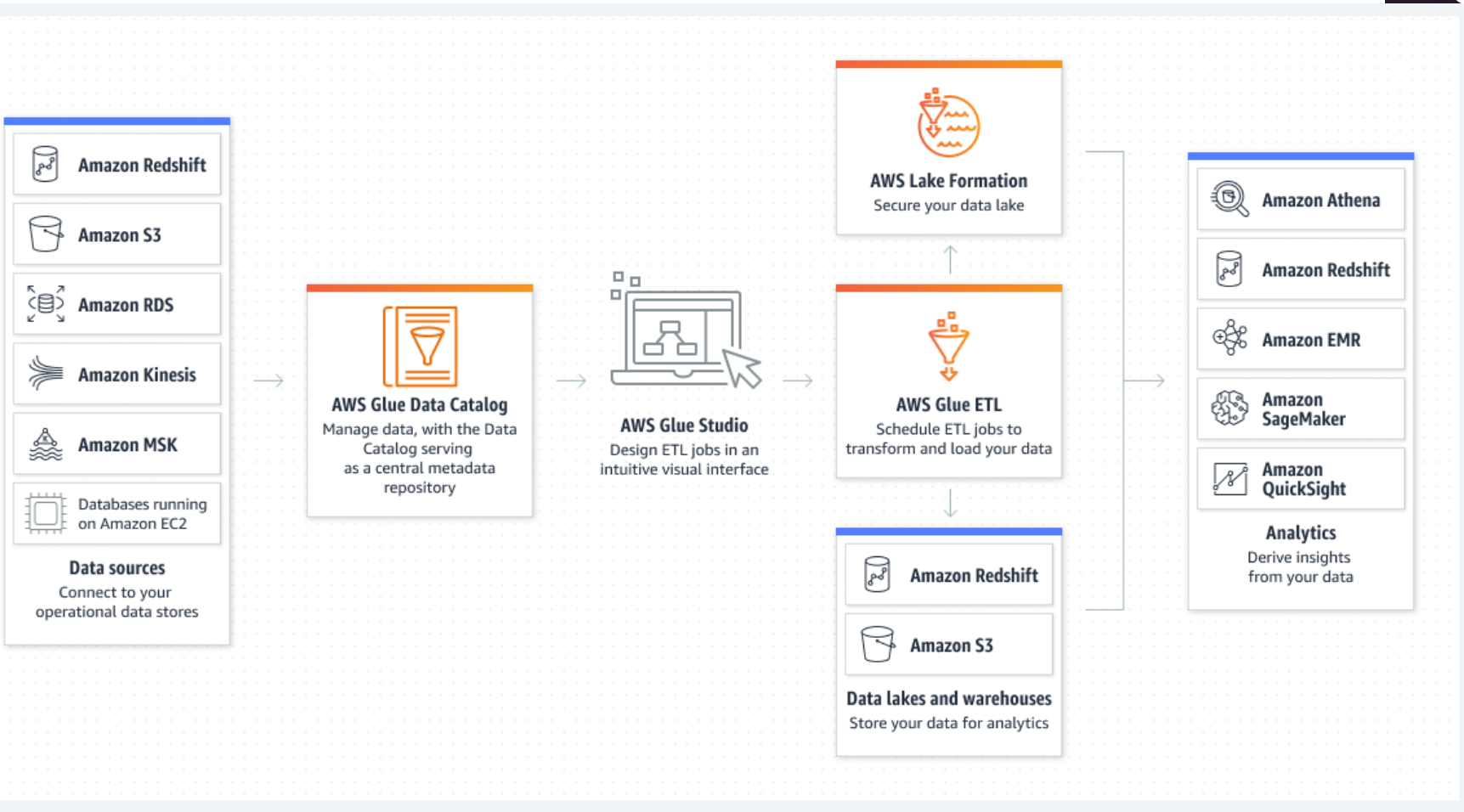
PROFINIT

AWS Glue – event driven ETL

AWS Glue – data catalog



AWS Glue – no-code ETL



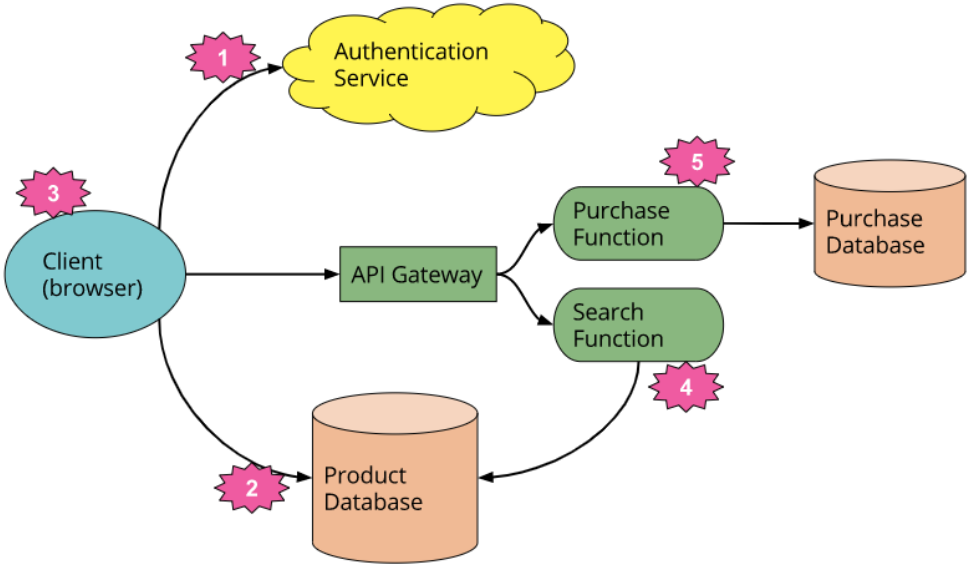
AWS Glue – data quality

School Example

Pet store

- Standard approach

- Serverless approach



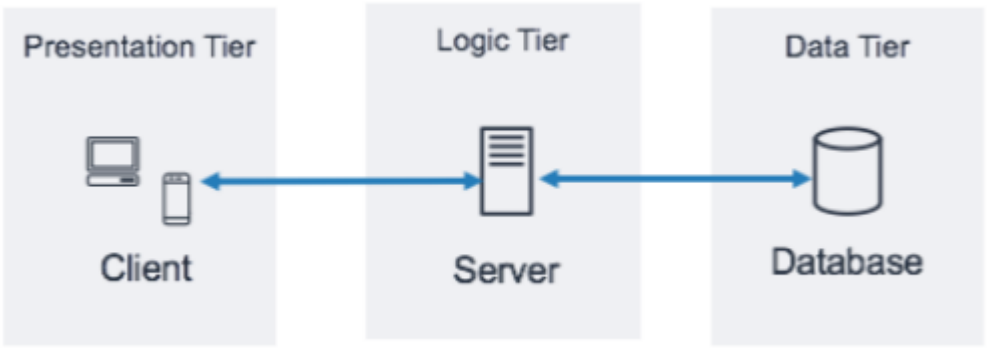
PROFINIT

Pet store

- Splitted to more modules
- External authentication service
- API Gateway
- Database splitted

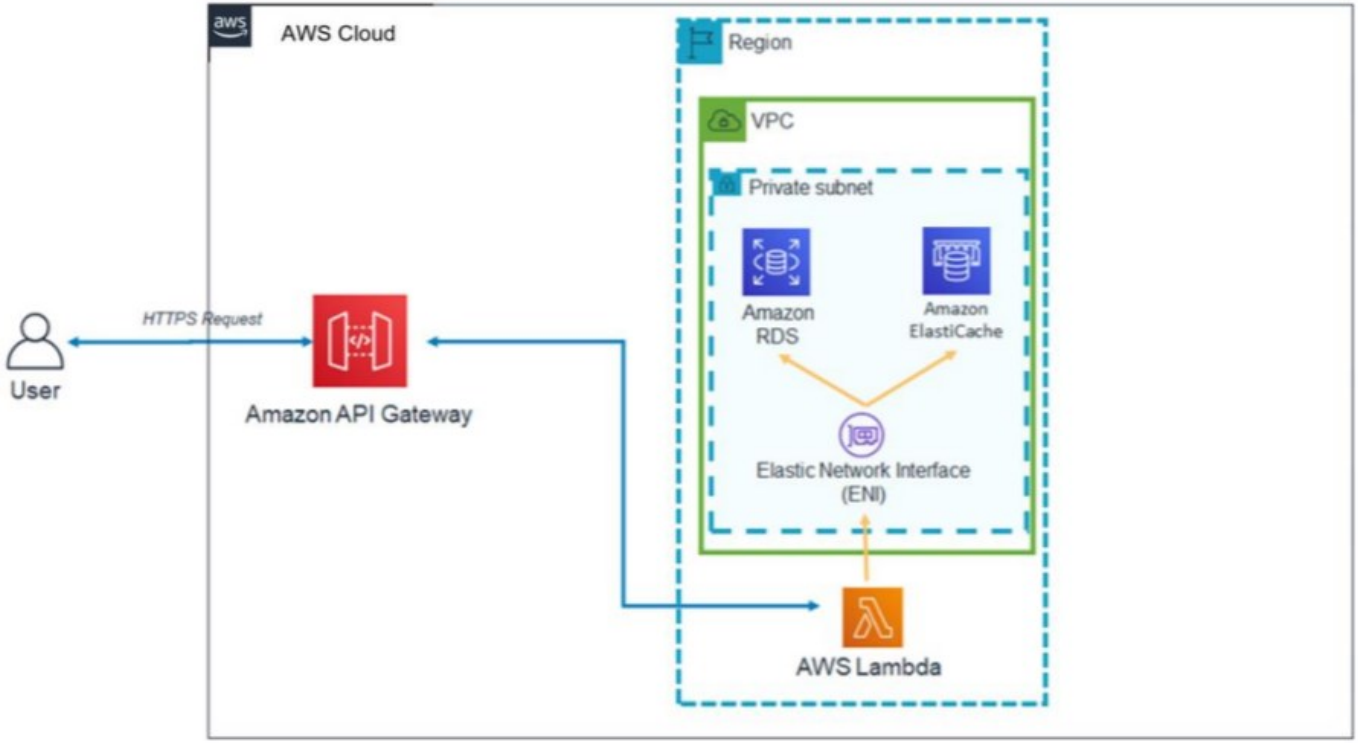
PROFINIT

AWS three tier architecture



PROFINIT

AWS three tier architecture

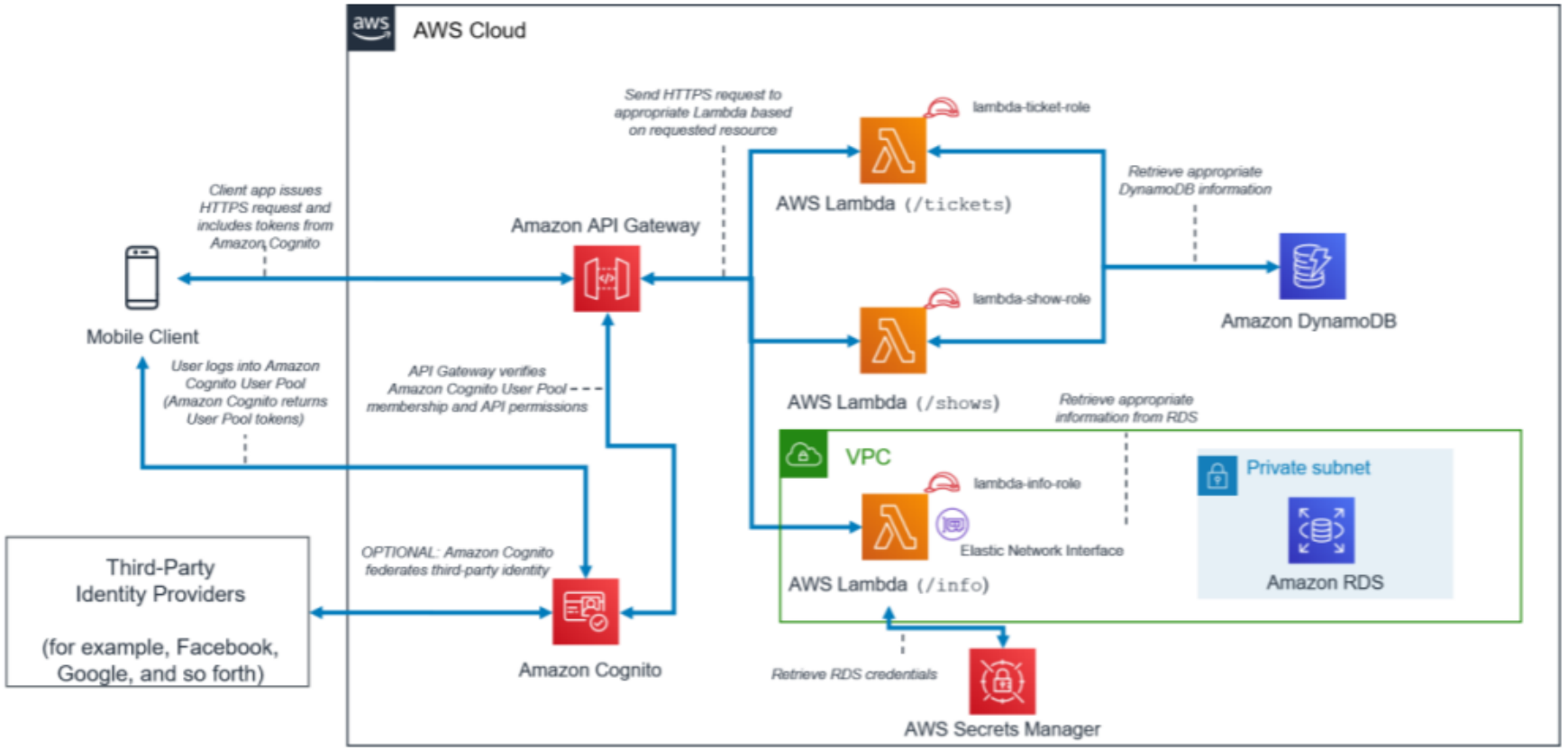


PROFINIT

AWS use cases

- • Mobile backend – A mobile application communicates with API Gateway and Lambda to access application data. This pattern can be extended to generic HTTPS clients that don't use serverless AWS resources to host presentation tier resources (such as desktop clients, web server running on EC2, and so forth).
- • Single-page application – A single-page application hosted in Amazon S3 and CloudFront communicates with API Gateway and AWS Lambda to access application data.
- • Web application – The web application is a general-purpose, event-driven, web application back-end that uses AWS Lambda with API Gateway for its business logic. It also uses DynamoDB as its database and Amazon Cognito for user management. All static content is hosted using Amplify.

AWS use cases – Mobile backend

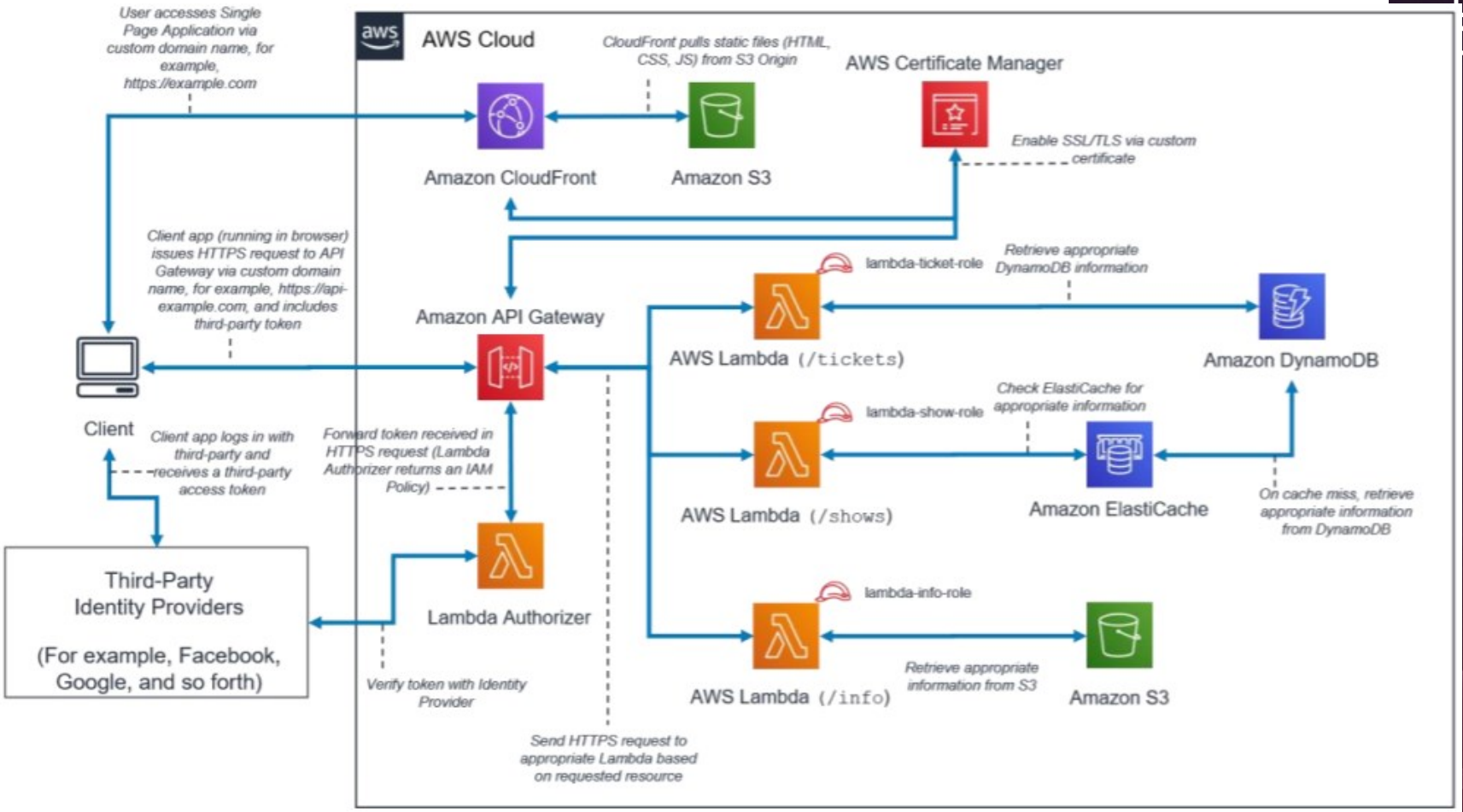


AWS use cases

- Presentation
  - Application running on the backend
- Logic
  - API Gateway with AWS Lambda.
- Data
  - DynamoDB is used

PROFINIT

AWS use cases – Single page application



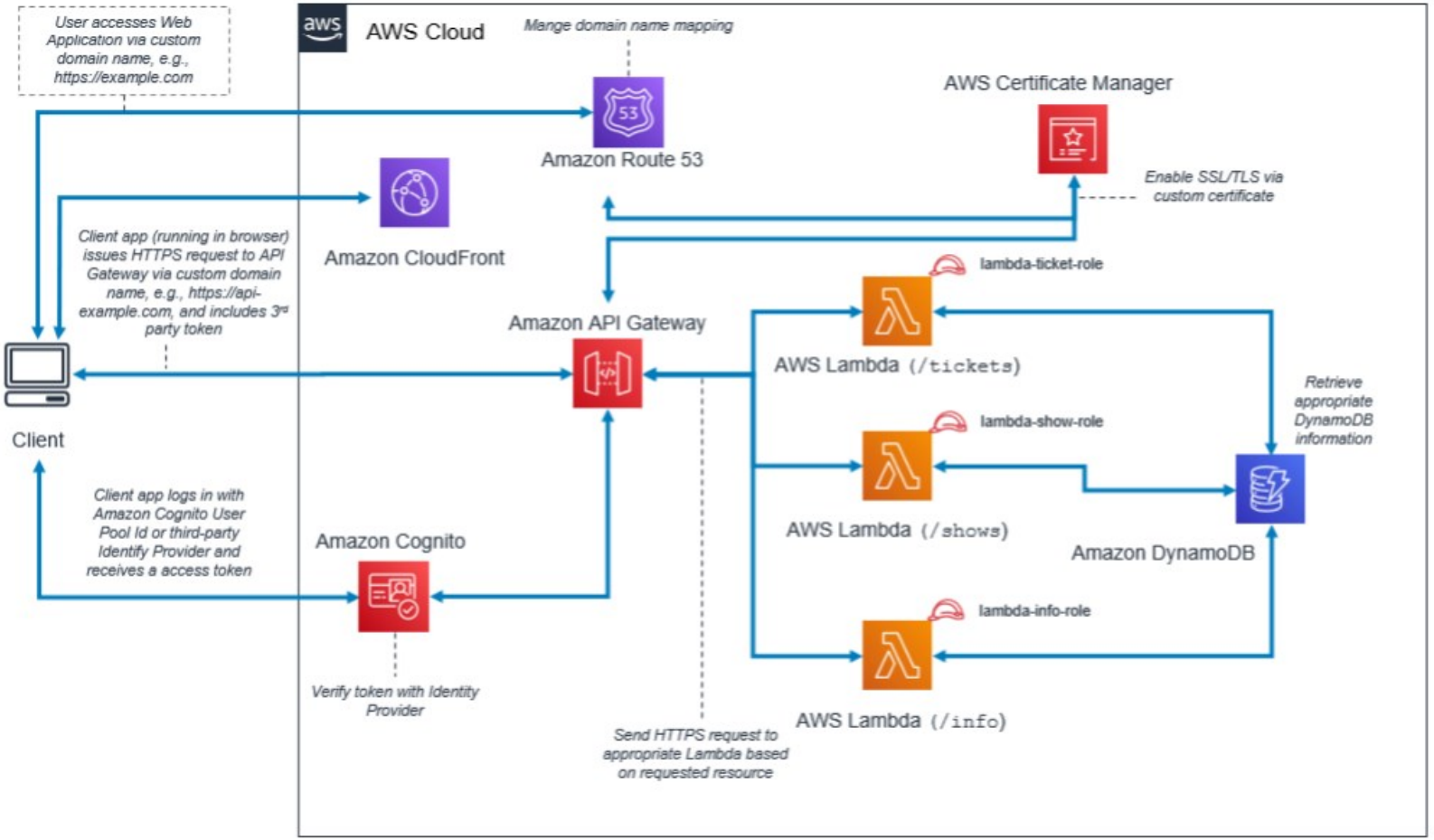
PROFINIT

AWS use cases

- Presentation
  - Static website content is hosted in Amazon S3 and distributed by CloudFront. AWS Certificate Manager allows a custom SSL/TLS certificate to be used.
- Logic
  - API Gateway with AWS Lambda.
- Data
  - DynamoDB is used

PROFINIT

AWS use cases – Mobile backend



AWS use cases

- Presentation
  - The front-end application is all static content (HTML, CSS, JavaScript and images) which are generated by React utilities like create-react-app. Amazon CloudFront hosts all these objects.
- Logic
  - Logic layer is built using Lambda functions fronted by API Gateway REST APIs.
- Data
  - DynamoDB is used

PROFINIT

AWS Lambda triggers

Pricing

- The AWS Lambda free tier includes one million free requests per month and 400,000 GB-seconds of compute time per month
- You can allocate any amount of memory to your function between 128 MB and 10,240 MB, in 1 MB increments.

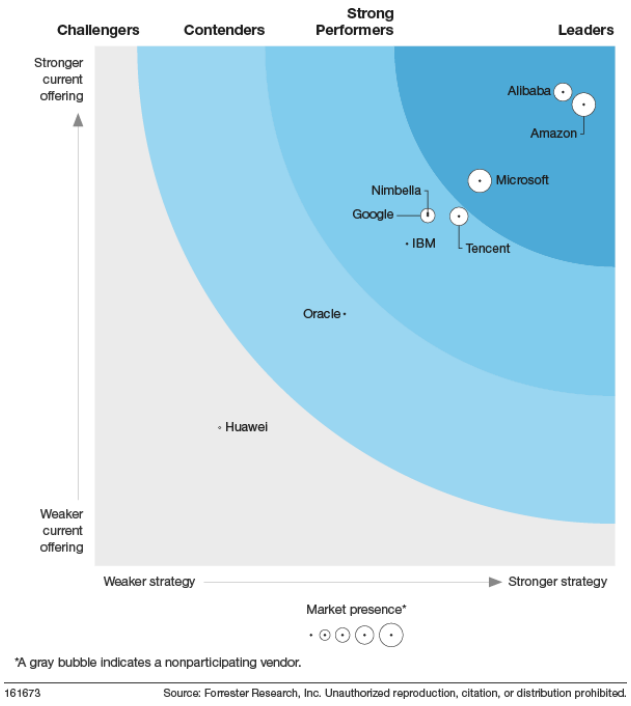
Architecture	Duration	Requests
x86 Price	$0.0000166667 for every GB-second	$0.20 per 1M requests
Arm Price	$0.0000133334 for every GB-second	$0.20 per 1M requests

PROFINIT

Azure functions

Azure Functions

- Supported languages
  - C#, Node.js, Python, JavaScript, Java, and PowerShell
- Forrester Wave
  - https://reprints2.forrester.com/#/assets/2/108/RES161673/report
- Detailed documentation
  - https://docs.microsoft.com/cs-cz/azure/azure-f



PROFINIT

Azure Functions

- Similar usage like Lambda
- Whitepaper with example
  - https://azure.microsoft.com/mediahandler/files/resourcefiles/create-event-driven-serverless-apps-with-azure-event-grid-and-azure-functions/Event-driven-serverless-apps-with-Event-Grid.pdf

Pricing

- Consumption
  - You're only charged for the time that your function app runs. This plan includes a free grant on a per subscription basis.
- Premium
  - Provides you with the same features and scaling mechanism as the Consumption plan, but with enhanced performance and VNET access. Cost is based on your chosen pricing tier. To learn more, see Azure Functions Premium plan.
- Dedicated (App Service) (basic tier or higher)
  - When you need to run in dedicated VMs or in isolation, use custom images, or want to use your excess App

Pricing – consumption

- Computed on the GB-seconds
- Example:
  - 2 GB for 5 seconds = **10GB-seconds**
- Memory usage is rounded up to the nearest 128-MB bucket
- West-central Germany

Meter	Price	Free Grant (Per Month)
Execution Time*	**$0.000016**/GB-s	400,000 GB-s
Total Executions*	**$0.20** per million executions	1 million executions

PROFINIT

Summary

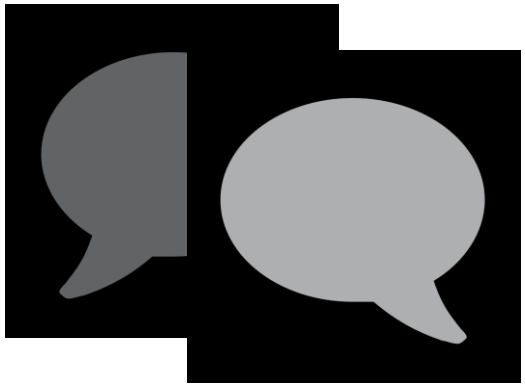
Summary

- **Serverless is not the correct approach for every problem**
- Serverless can save a lot of OPS effort
- Every cloud provider has this feature
- Vendor lock-in

PROFINIT

Sources

- https://www.datadoghq.com/knowledge-center/serverless-architecture/#:~:text=Serverless%20architecture%20is%20an%20approach,storage%20systems%20at%20any%20scale.
- https://d0.awsstatic.com/whitepapers/AWS_Serverless_Multi-Tier_Architectures.pdf
- https://martinfowler.com/articles/serverless.html
- https://en.wikipedia.org/wiki/Serverless_computing



Q&A

PROFINIT