BITS Pilani

Cloud Computing

Session 9-10

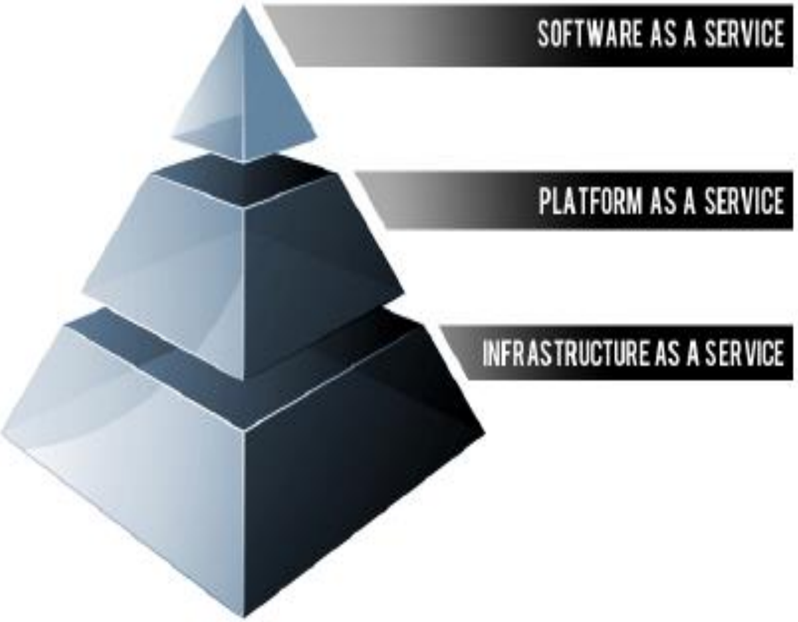
Serverless Computing & Cloud Scaling

Agenda

- Recap SaaS
- Serverless Computing
- Demonstration of AWS Lambda for Serverless computing
- Cloud Scaling

Software as a Service (SaaS)

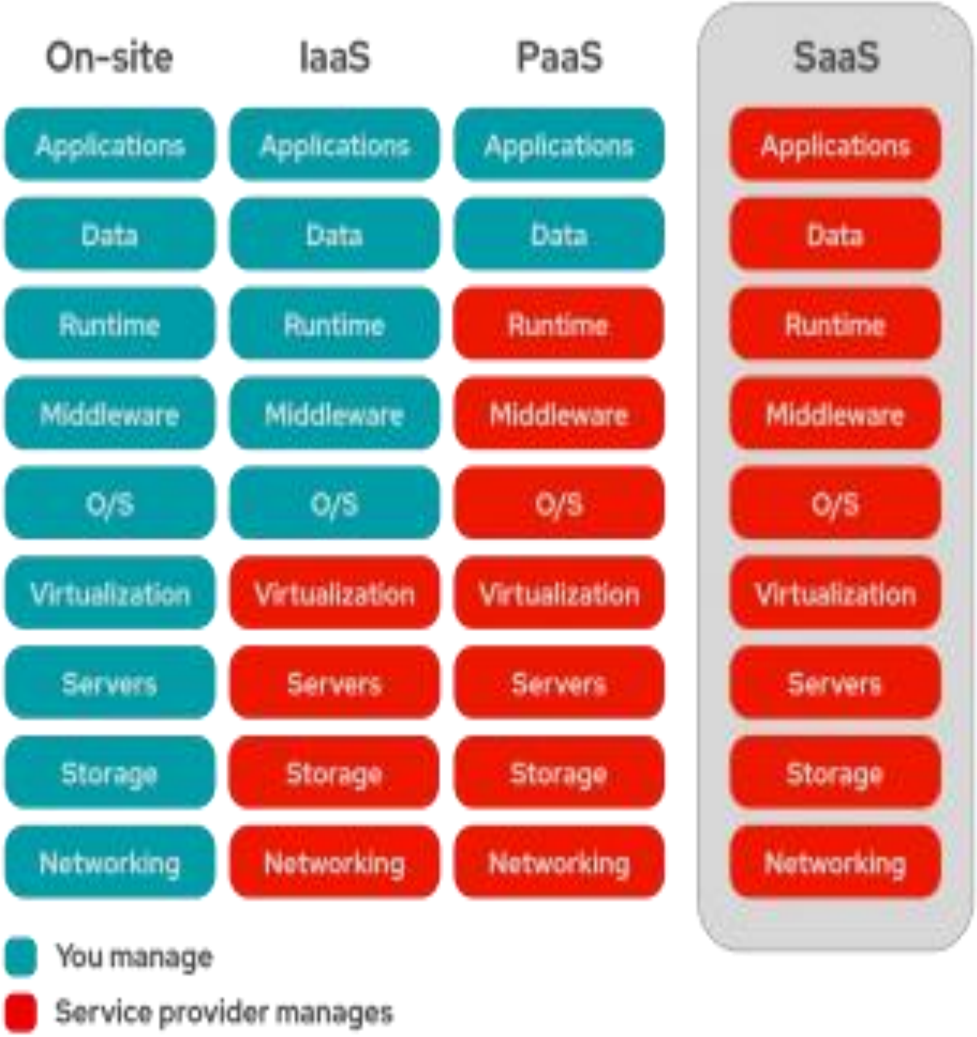
- Powerful way of consuming a software application as a service!
  - Do not develop, just use it
- Subscription-based
- All cloud features:
  - on-demand, hosted remotely, scalable.
- E.g: Project management Spreadsheets, docs, Netflix, Gmail, Office365



Pros & Cons of SaaS

- **Pros Of SaaS**
  - Cost Reduction – Lower the license costs
  - Scalability
  - Integration
  - Upgrades
  - Ease of Use – Best practices

- **Cons of SaaS**
  - Security
  - Limited Customization & Control



On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

🟦 You manage

🟥 Service provider manages

Serverless Computing

- Serverless computing is a cloud-computing model
- Cloud provider dynamically manages the allocation of machine resources.
- It allows developers to build and deploy applications without worrying about managing servers or infrastructure.

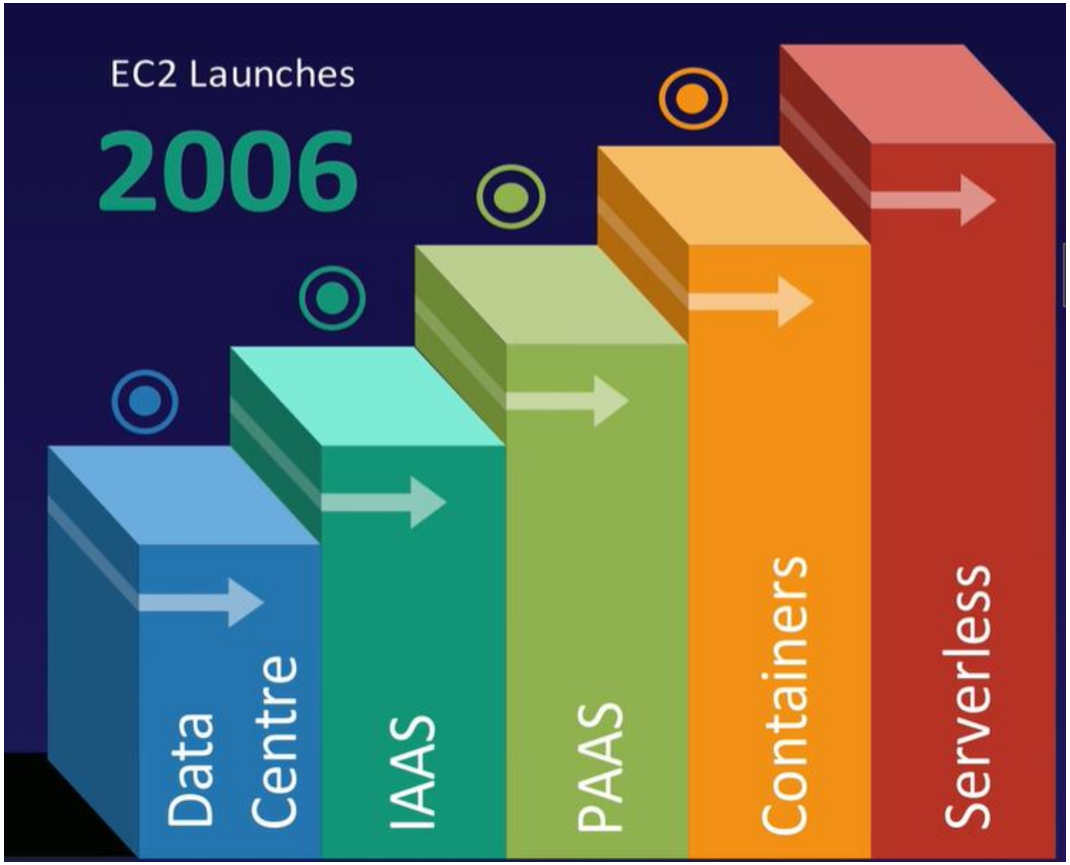


Image Courtesy: cloudguru.com

Key Features of Serverless Computing

- **No Server Management**:
  - Developers don’t need to worry about provisioning, scaling, or maintaining servers.
  - Everything is handled by the cloud provider.
- **Automatic Scaling**:
  - Serverless platforms automatically scale applications up and down based on demand.
  - Ensuring optimal performance and resource utilization.
- **Event-Driven Execution**:
  - Serverless functions are often triggered by events, such as HTTP requests, database changes, or message queue events. This allows for highly responsive and efficient applications.
- **Pay-as-You-Go Pricing**:
  - You only pay for the compute resources you use.
  - Billing is based on the number of requests and the execution time of your code.

Key Features of Serverless Computing

- **Reduced Operational Overhead**:
  - No need to manage and maintain server infrastructure, allowing developers to focus on writing code.
- **Cost Efficiency**:
  - You only pay for the compute time your code actually uses, potentially leading to significant cost savings.
- **Faster Development Cycles**:
  - Simplifies deployment and infrastructure management, leading to quicker iteration and delivery of features.

Cons of Serverless Computing

- **Cold Starts**
  - Initial invocation of serverless functions can be slow due to the time it takes to start up, which may impact performance for latency-sensitive applications.
- **Vendor Lock-In**
  - Relying on specific serverless platforms can make it difficult to switch providers or migrate applications.
- **Limited Execution Time**
  - Serverless functions often have execution time limits, which might not be suitable for long-running processes.

Cons of Serverless Computing

- **Complexity in Debugging**
  - Debugging serverless applications can be more challenging compared to traditional applications due to the distributed and event-driven nature.

- **Security Concerns**
  - The multi-tenant nature of serverless platforms can raise security issues, requiring robust measures to ensure data integrity and security.

Popular Serverless Platforms

- AWS Lambda
- Google Cloud Functions
- Azure Functions
- IBM Cloud Functions

AWS Lambda

- Abstraction Layer
  - Data Centers
  - Hardware
  - Assembly and High level Code
  - Operating Systems
  - Application Layer / APIs
- AWS Lambda
  - Compute service
  - Allows you to upload your code
  - Create a Lambda Function
  - Provisions and manages the servers that are used to run the code
  - Users need not worry about OS, scaling, patching.

Using Lambda

- Event Driven compute service
  - Changes in DB, S3 Bucket
- User request driven compute service
  - AWS API Calls using SDKs, API Gateway
- Languages
  - Node.js
  - Python
  - Go
  - Java

- Pricing
  - No of requests served
  - Duration
    - Time from start of execution of code till it returns or terminates

Why Lambda

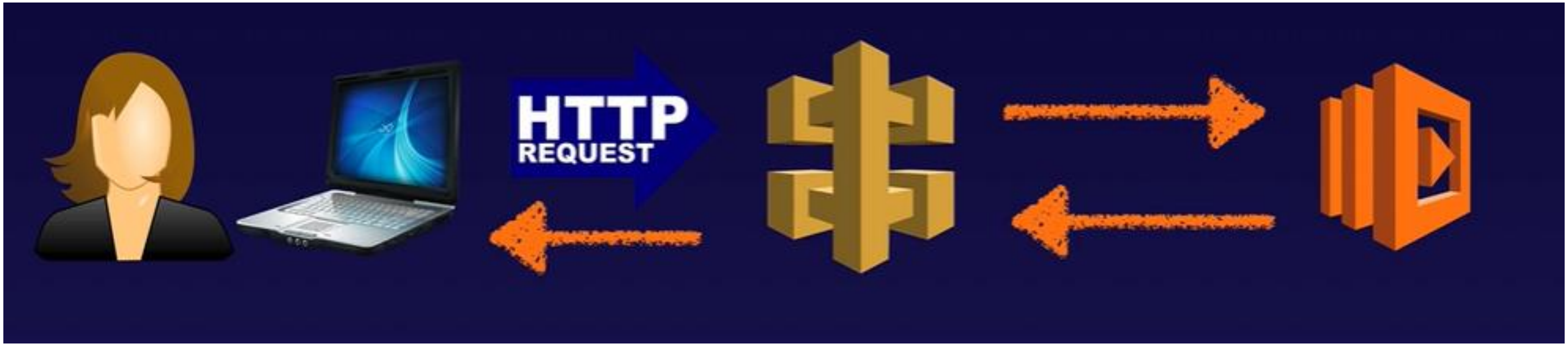
- No servers
- Continuous scaling
  - Automatic
  - Lambda functions are independent, one event = one lambda function
- Economic

AWS Serverless Services

- API Gateway
- Lambda
- S3
- Dynamo DB

EC2 is not serverless

Using Lambda

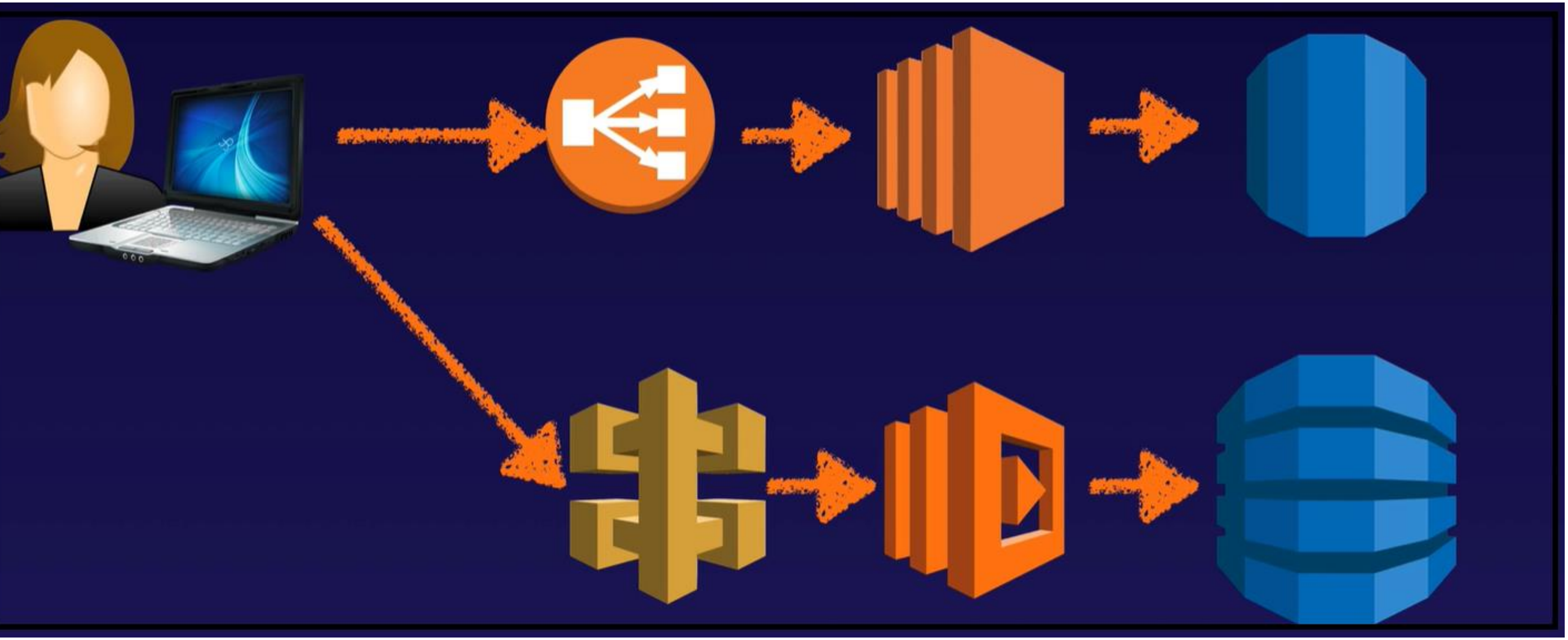


- API Gateway faces the user requests
- Invokes a lambda function in the backend
- Lambda function handles the request with specific action
- Returns response back to API Gateway – ultimately to the user

All Image Courtesy: cloudguru.com

Traditional Versus Serverless Architecture

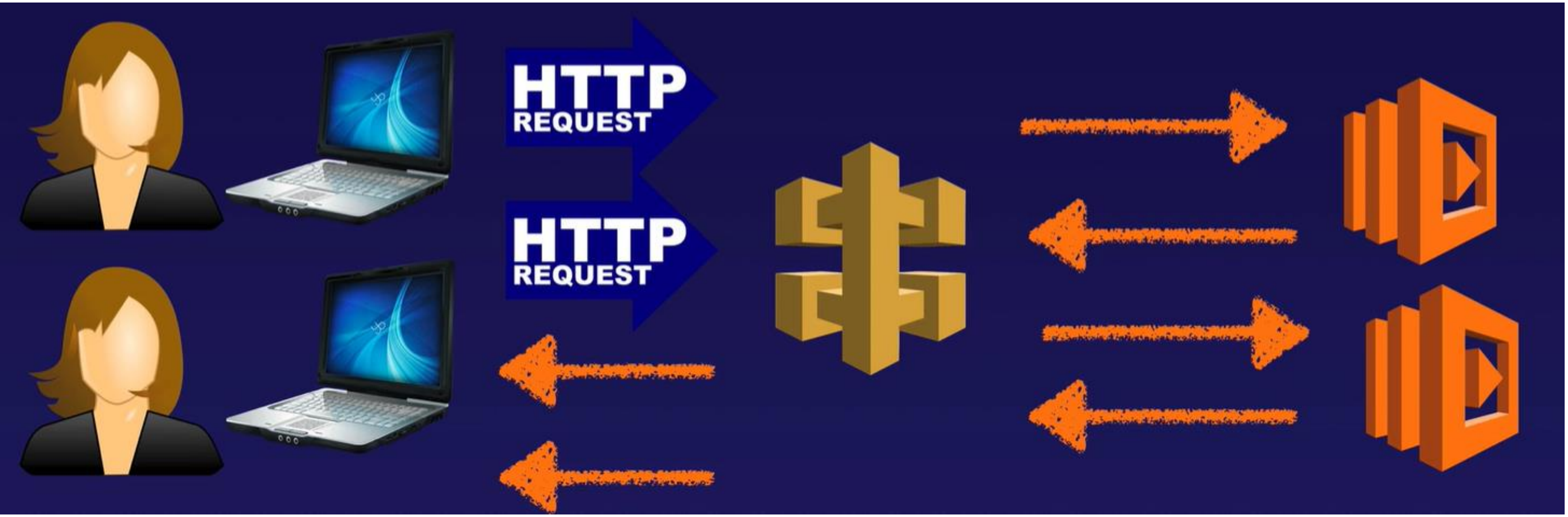
Traditional Architecture



Serverless Architecture

Image Courtesy: cloudguru.com

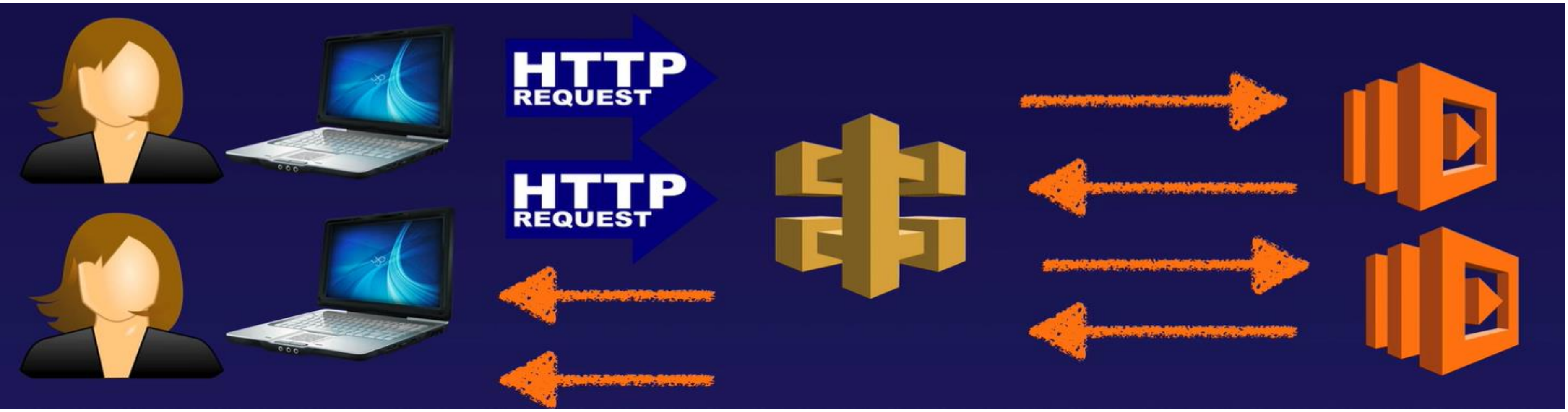
Lambda hands on



Let’s do some hands on

All Image Courtesy: cloudguru.com

Scaling



- Every request handled by API Gateway
- Redirects it to a Lambda function
- **HTTP Request handled by a separate Lambda function.**
  - **1-1 mapping**
- Returns response back to API Gateway
  - ultimately to the user

All Image Courtesy: cloudguru.com

What is Cloud Scaling

Image Credit: Scaling in Cloud Computing : A Comprehensive Guide?

- Increasing or decrease the resources - to meet the changing demand.
- Which resources ? – Compute, storage, networking
- How ? - Virtualization

Why Cloud Scaling ?

Image Credit: Scaling in Cloud Computing : A Comprehensive Guide?

- Meet consistent performance needs
- E.g: High Throughput and low latency with the spike in the arrived traffic

When to use ?

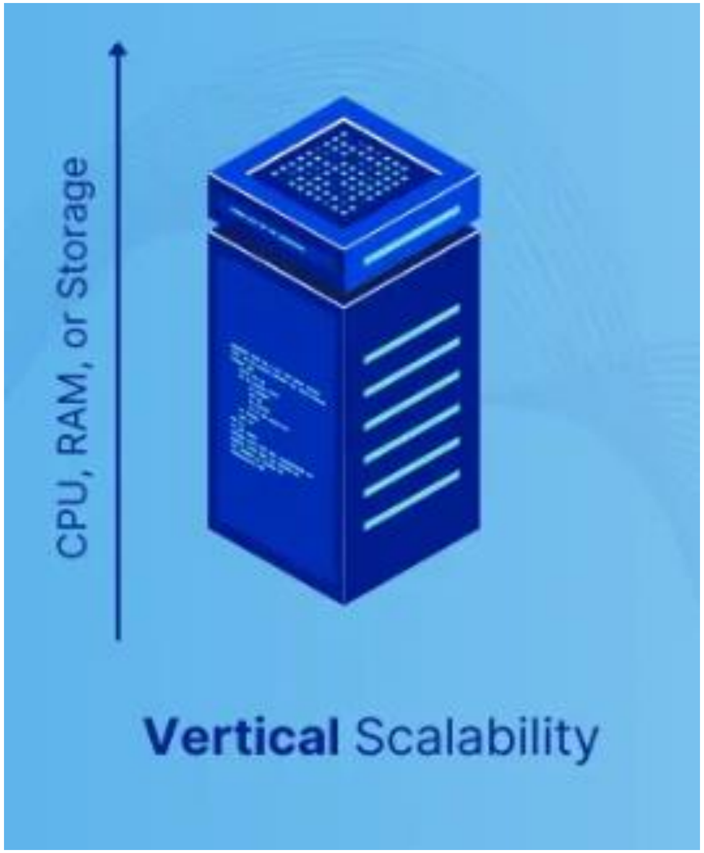
- Expand business
- Seamlessly adapt to changes without causing downtime

Scaling Types

Three Popular Types

- Vertical
- Horizontal
- Diagonal

Vertical Scaling



- Add additional CPU/memory/storage to the existing machine
- E.g: Add more memory for a database running out of memory. DB can handle more data

All Image Credit: What is Cloud Scalability: The Ultimate Guide in 2025

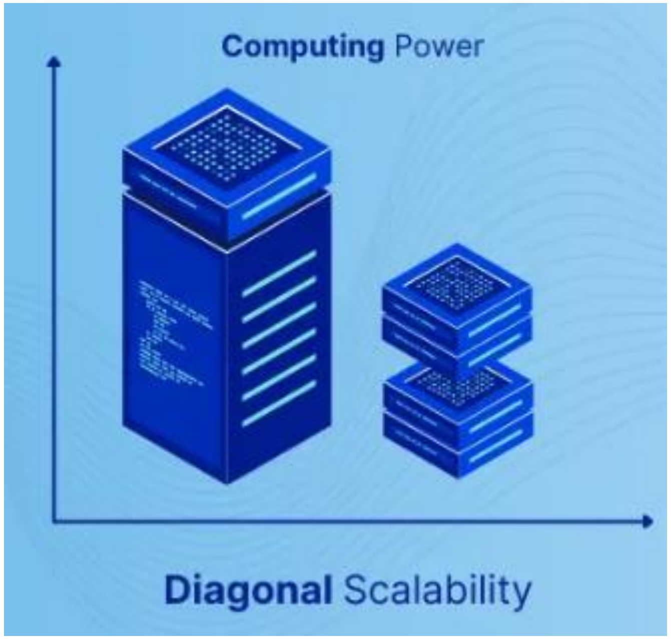
Horizontal Scaling

Computing Power
Horizontal Scalability

- Add additional machines/nodes/servers
- Workload gets distributed across nodes – including the new ones

All Image Credit: What is Cloud Scalability: The Ultimate Guide in 2025

Diagonal Scaling



- Combination of vertical and horizontal scaling – Hybrid

Image Credit: What is Cloud Scalability: The Ultimate Guide in 2025

More on Scaling…

Forms & Enablers of Scaling

- Auto Scaling
- Load Balancing
- Containerization
- Caching and Content Delivery Networks (CDNs)
- Cloud Monitoring

Benefits of Scaling

- Improved Performance
- Increased Reliability
- Cost Efficiency
- Easier Deployment

Summary

- SaaS
- Serverless Computing
- Demonstration of AWS Lambda for serverless computing
- What is Cloud Scaling ?
- Why Cloud Scaling ?
- Scaling Types
- Forms of Scaling
- Benefits of Scaling