

greatlearning
Learning for Life



Post Graduate Program in Cloud Computing



Road to Serverless

About Me

- Sathyajith Bhat
- Senior DevOps Engineer – Adobe I/O
- Author, [Practical Docker With Python](#) (Apress)
- Organizer, AWS Users Group Bengaluru & Barcamp Bangalore
- [AWS Community Hero](#)
- Community Moderator: Stack Exchange (Super User, WebApps)

What is Cloud Computing?

- National Institute of Standards and Technology (aka NIST) defines Cloud Computing as Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
- This cloud model is composed of **five** essential characteristics, **three** service models, and **four** deployment models.

Essential Characteristics

- On-demand self service
 - Provision capabilities automatically without human intervention
- Broad network access
 - Provisioned capabilities are available over any network
- Resource pooling
 - Provisioned resources are selected from the service provider's pool of resources to serve multiple consumers and assigned/re-assigned per demand
- Rapid Elasticity
 - Capabilities are elastically provisioned and released automatically, per demand
- Measured service
 - Optimize resource usable by metering resources - via limits or per billing

Prior to Cloud

- Dedicated Machines
 - Located in Customer Premises (“On-Prem”)
 - Co-located in Data Centers (“Co-lo”)
- No scope for flexibility or elasticity
- Provision for peak 1%, pay for heavy underutilization

Prior to Cloud

- Dedicated Machines
 - Provisioning of new resources takes weeks
 - No option to scale down when not needed
 - Requires acquisition of expensive single use hardware
 - Upgrading new hardware is nontrivial



Prior to Cloud

- Virtualization
 - Use of Virtual Machines (VM) on dedicated machines in DC
- Why?
 - Better utilization of resources
 - Improved Hypervisor performance

Prior to Cloud

- Virtualization
 - Use of Virtual Machines (VM) on dedicated machines in DC
- But
 - Still limited by resources on the DC
 - Provision for peak 1%, pay for heavy underutilization
 - Provisioning is easy till it's not

Cloud

- Service Models
 - IaaS
 - Infrastructure as a Service
 - Servers, Networks, Storage
 - GCP, AWS, Azure
 - PaaS
 - Platform as a Service
 - You get an entire “platform” – OS + Programming Environment
 - Heroku, Google App Engine



Cloud

- Service Models
 - SaaS
 - Software as a Service
 - Pre-packaged Application as a service
 - Adobe Creative Cloud, Microsoft Office 365



Cloud

- Deployment Models
 - Private Cloud
 - Provisioned for exclusive use
 - Community Cloud
 - Provisioned for exclusive use by a specific community of consumers



Cloud

- Deployment Models
 - Public Cloud
 - Provisioned for use by general public
 - Hybrid Cloud
 - composition of two or more distinct cloud infrastructures



Presenting, Serverless

- Serverless?!
- Does it mean there are no servers?

Presenting, Serverless

- “Serverless computing is a cloud-computing execution model in which the cloud provider acts as the server, dynamically managing the allocation of machine resources”
- Translation:
 - You invoke a function
 - Cloud provider handles the rest of the things



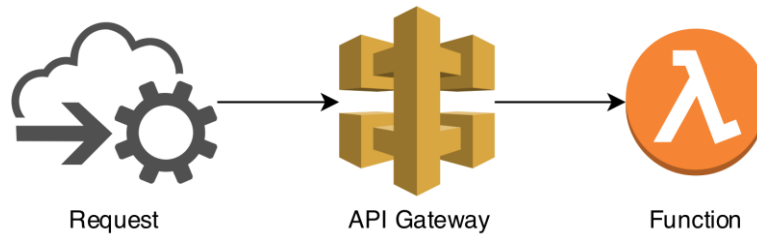
Presenting, Serverless

- Advantages of Serverless
 - No server management
 - Scaling on demand
 - True pay per use
 - High Availability and fault tolerance out of the box

Presenting, Serverless

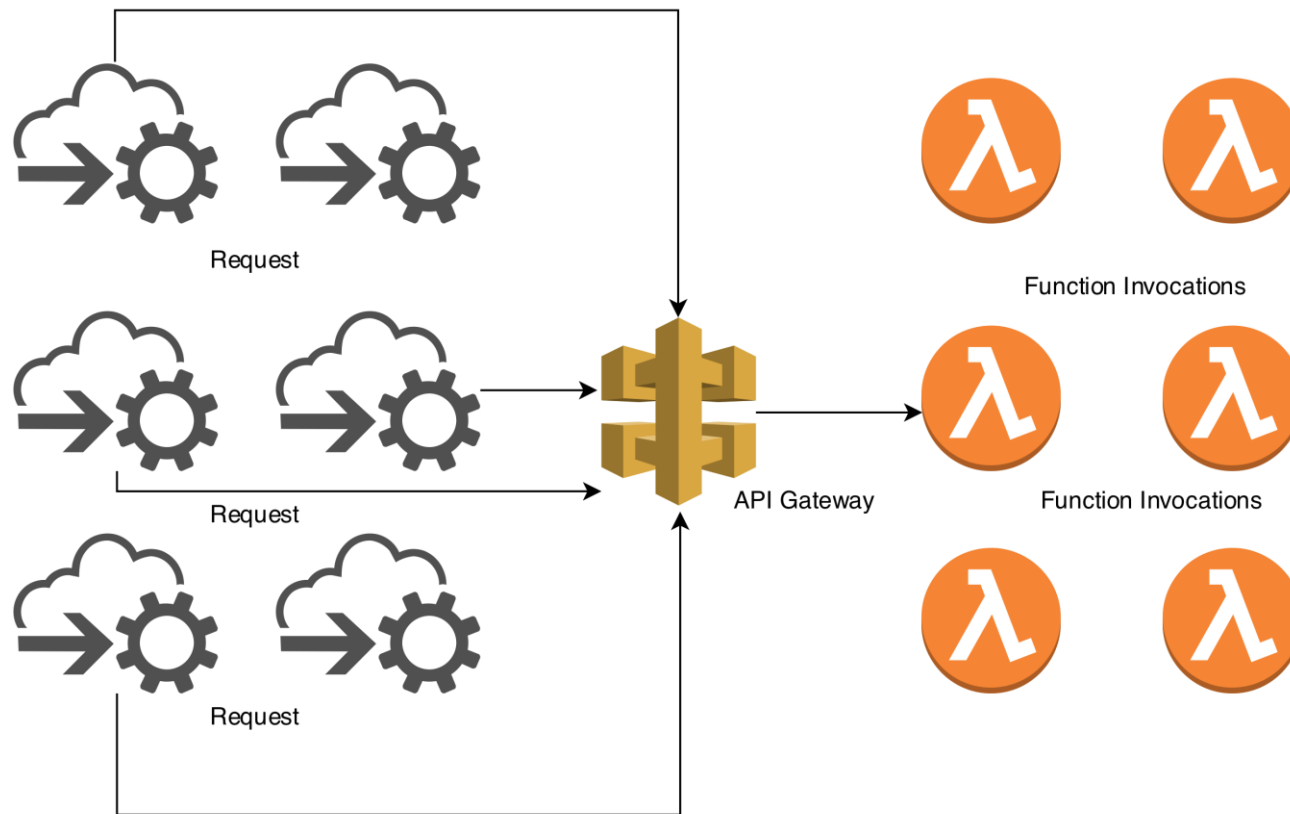
- What does serverless mean to you?
- You don't care about
 - Servers and server configurations
 - Scaling and balancing
 - Capacity planning
 - High running costs
 - Sudden spikes

Scaling up in Serverless



One request, one function invocation

Scaling up in Serverless





Containers and Serverless

- Containers: simplified versions of Virtual Machines
- Containers (tend to be) small, light weight and can be launched faster than VMs
- Ideal for Serverless workloads



Containers and Serverless

- A request comes in
- Function is invoked
- Application code is copied to the container and launched

Use cases for Serverless

- API Driven workloads
 - Have a function for endpoint
 - Independent scaling for each endpoint
- Scheduled job automation
 - Don't keep a server running for a job that runs once a day
 - Fire a function per timeframe
- Event driven model
 - Trigger functions based on events

You may have been using Serverless

- ???
- Why/how?
- Amazon S3
 - Use case: E-commerce portal
 - Fetch images from S3
 - S3 transparently downloads the images



You may have been using Serverless

- **Alexa**, **How many visitors on GreatLearning?**
 - **Alexa** – Recognized by Echo
 - **How many visitors on GreatLearning** – invokes a Lambda function

Serverless Provider Options



Azure Functions



AWS Lambda



GCP Cloud Functions



Apache OpenWhisk



Fn Project

Who's Using Serverless?

- [ABP News Network](#)
 - “ANN elected to use [AWS Lambda](#) to deliver an architecture that seamlessly updates content feeds served from Amazon S3 to mobile users without requiring the management and maintenance of a single server”
- [Saturn Systemwares](#)
 - “We’ve lowered the cost of running the Sabarimala Virtual Q Portal by around 30% by using AWS Lambda, despite a 42% increase in portal users.”
- Me
 - Replaced unused EC2 instances with Lambda for cleaning up unwanted data
 - Replaced always running MySQL database with Aurora serverless to scale up during tests and scale down when not needed

Resources

- [ACM Cloud Computing Definition](#)
- [Zappa Serverless Framework](#)
- [Introduction to Serverless & Functions as a Service](#)

Thank You!