

# Functions as a Service

Cloud Computing and SaaS



# Announcements

- Project
  - Checkpoint 4 Skeleton Code Released
    - Recommended to finish by Friday
  - Local Tests Released
  - Pushed Formatting fix for docker-compose.yml
  - Make sure to pull from source repo!
- We're getting close to AWS deployment -- Make sure your code passes the tests!

# Last Time

- Kubernetes
  - Overview
  - History
  - Applications
  - Architecture
  - Benefits
  - Drawbacks
- Guest Speaker: Usman Muzaffar

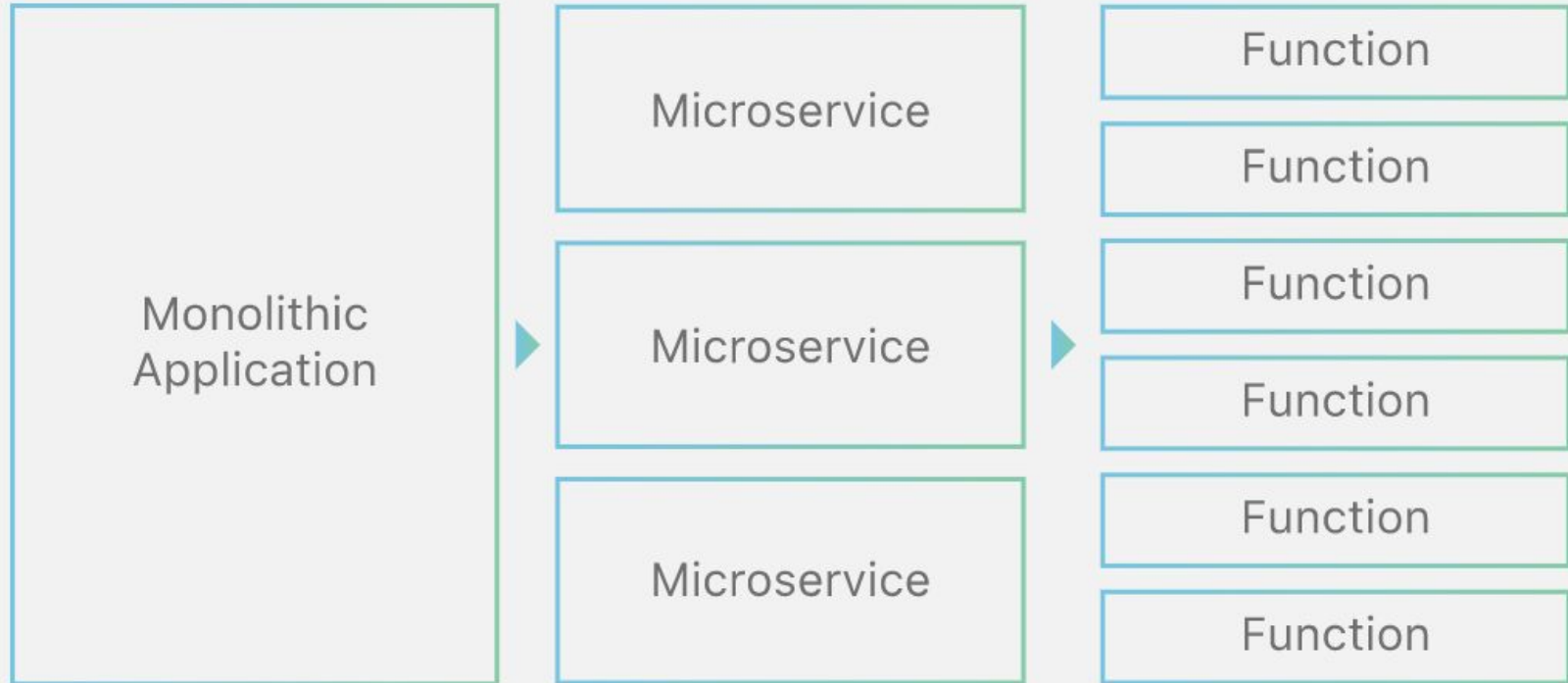
# Today

- Functions as a Service
  - Overview
  - vs. other as-a-Service models?
  - Benefits
  - Drawbacks
  - Demo

# FaaS Overview

- Key Idea - Serverless Computing
  - Manage functions, not servers
  - Abstract away server management
    - Developers only focus on writing code, let the cloud provider scale up and down as needed
    - Don't have to worry about Kubernetes!
  - Kind of a misnomer -> our apps are still deployed on servers

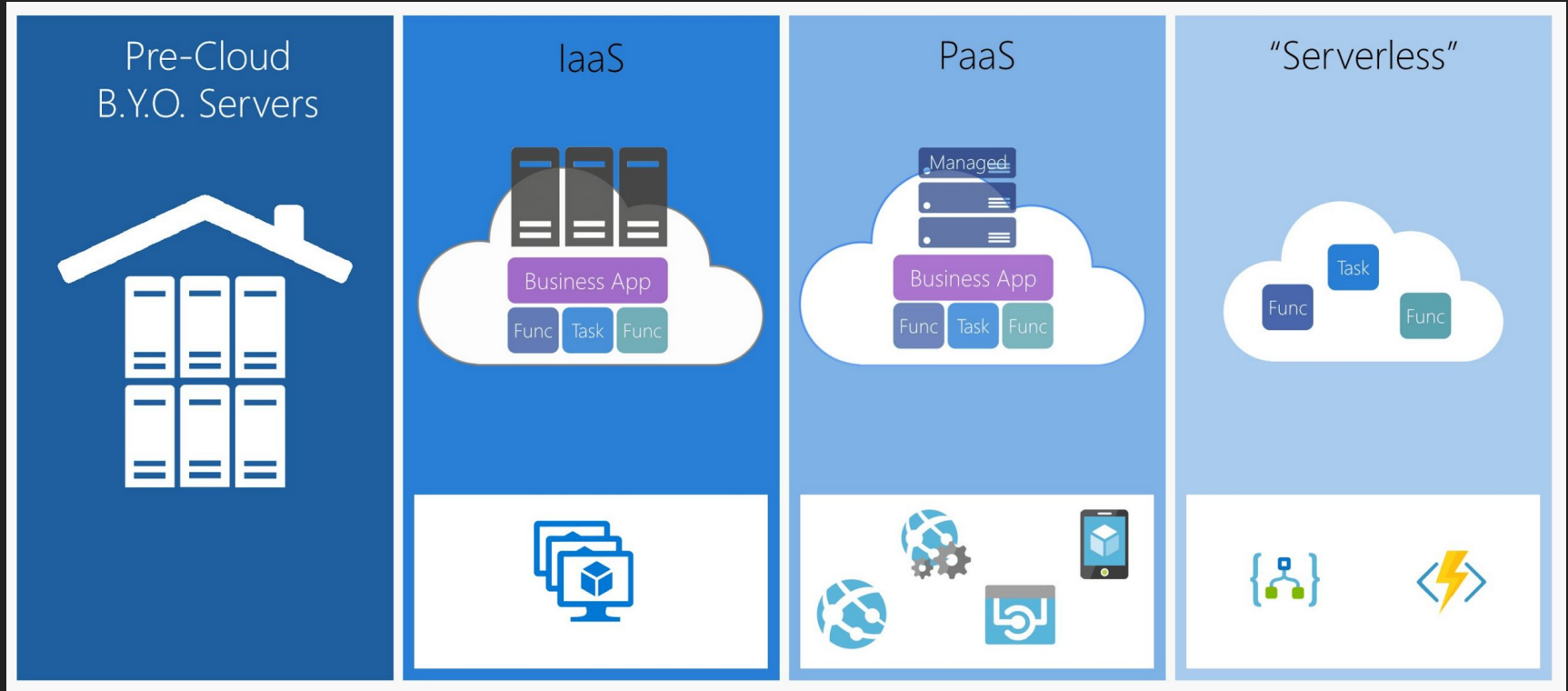
# FaaS Overview



# FaaS Overview

- Completely modular development flow
- Billed per execution, rather than server usage
- Each function is “listening”
  - In the background, the cloud provider manages servers that are constantly listening for requests
  - Functions are deployed on those servers
- AWS Lambda, Google Cloud Functions, Azure Serverless

# Recall - Other as-a-Service Models

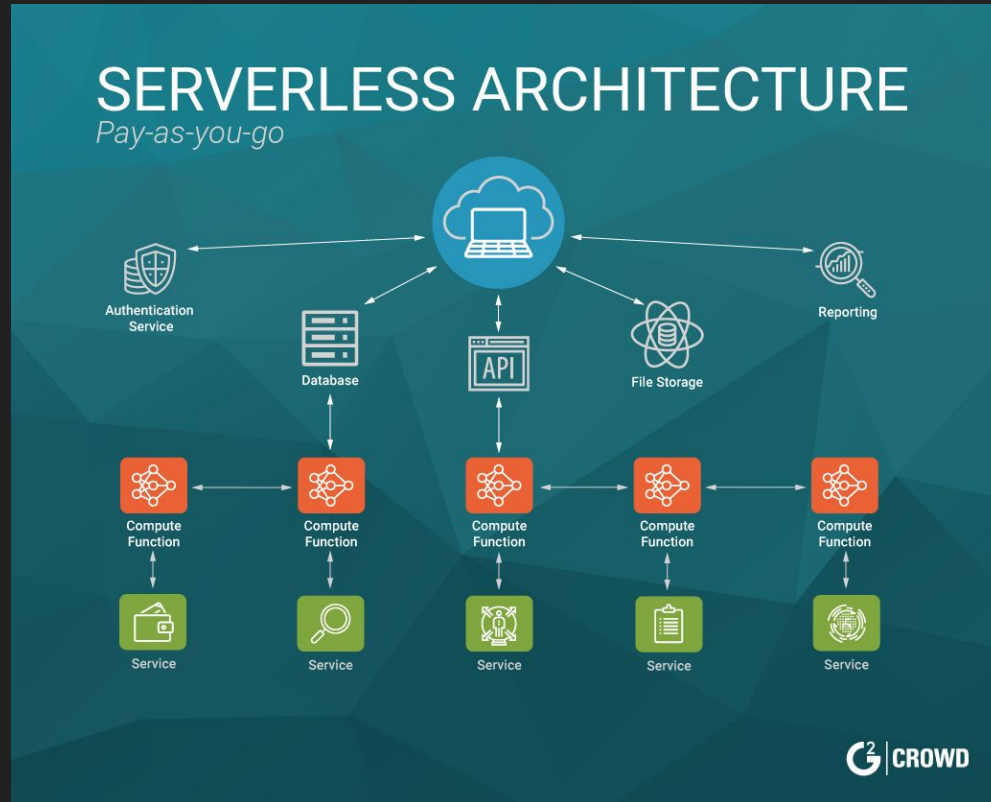




# FaaS Benefits

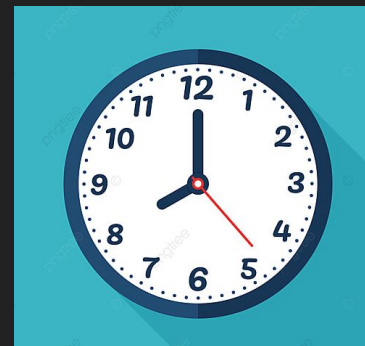
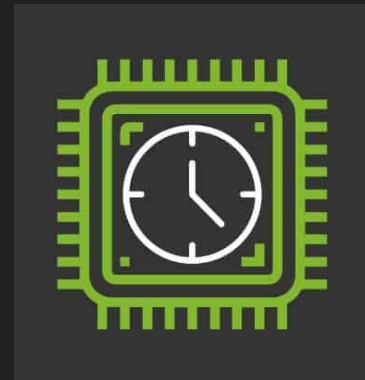
- Faster time to deployment
  - Servers management (digital and physical) managed by cloud service
- Built-in Scalability
  - Cloud Service will provision more/less resources depending on demand
- Cost-Efficient
  - Not charged for idle server time

# FaaS Benefits



# FaaS Drawbacks

- Functions must be stateless
  - Can't cache data between requests
  - Need to externalize state somewhere else
    - Databases, Unstructured Data Storage, etc.
- Hard Limits on Execution Time
  - Suitable for client-facing API endpoints
  - Unsuitable for long-running tasks
    - Data migration, Training ML Models, etc.



# FaaS Demo

- General Steps
  - Create Function
  - Deploy Code
  - Connect to API Endpoint
    - Allows our function to talk to the outside world
  - Test the function, HTTP Request