SERVERLESS ARCHITECTURE

A NEW FRONTIER IN CLOUD COMPUTING

AMITH PULLA, TECHNICAL PROGRAM MANAGER

SERVER PRE-SILICON PROGRAMS AND VIRTUAL PLATFORMS, INTEL CORP.

DEVELOPERWEEK AUSTIN, NOVEMBER 2019

SERVERLESS ARCHITECTURE OVERVIEW

- Its not about running code without servers (compute, storage and network is required)
- No purchasing, renting or provisioning servers or virtual machines
- No need to worry about scaling, capacity planning and maintenance operations
- Compute Stack: (Function as a Service or "FaaS") for custom code run in containers and (Backend as a Service or "BaaS") for database and API Gateway
- Pay-per-execution model
- Lower TCO (Total cost of ownership), engineering complexity and TTM (Time to market)
- New paradigm in software architecture and development

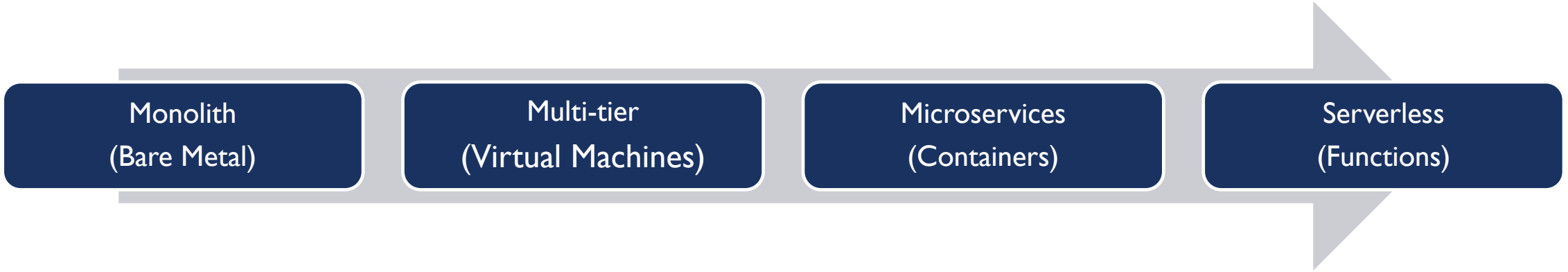
KEY MARKET PLAYERS AND INNOVATORS

- AWS Lambda, introduced in 2014 (abstract serverless computing), Firecracker 2018
- Google Cloud Functions and Google Cloud Datastore by Google Cloud Platform, 2016
- IBM Cloud Functions and OpenWhisk in the public IBM Cloud
- Azure Functions and Azure Data Lake by Microsoft (Azure public cloud or on-premises via Azure Stack)
- Fn Project by Oracle
- FaunaDB Cloud

FAAS (FUNCTION AS A SERVICE)

- FaaS is evolution from IaaS and PaaS
- Stateless, Ephemeral (short-lived), Event-triggered, continuous scaling
- With FaaS, you compose your application into individual, autonomous functions.
- Each function is hosted by the FaaS provider
- Scaled automatically as function call frequency increases/decreases
- Cost effective way of paying for compute resources, no "always on" applications on multiple instances
- Ideal for small number of functions to be hosted
- Reduce operational cost, complexity, and engineering lead time

ARCHITECTURAL CONSIDERATIONS



- **Implement single-purpose stateless functions**
- **Design for push-based, event-driven patterns**
- **More thicker and richer client/frontends (business logic)**
- **Apply security mechanisms across the software stack**
- **Leverage third-party services as much as possible**
- **Use BaaS for Authentication**

FRAMEWORKS

- Amplify_Framework
- Claudia_js
- Serverless
- Auth0
- Up
- Pulumi
- Sigma
- Riff
- Architect
- Jets

CHALLENGES

- Limited control for the user
- Third-party APIs dependencies and security
- Cold starts/startup delays
- Vendor lock-in
- Hard timeouts
- Performance with relational databases and indexing
- Language support by vendor
- Multitenancy (shared resources)
- Limited Debugging
- No custom server optimizations

POTENTIAL SECURITY CHALLENGES

- **Increased attack surface**
  - Serverless functions consume data from a wide range of event sources
  - Inadequate security testing
- **Function event data injection**
- **Broken Authentication**
- **Insecure third-party dependencies**
- **Insecure application secrets storage**
- **DDoS attacks, resources pushed to the limit**
- **Flow manipulation for the functions**
- **Obsolete functions, resources and event triggers**
- **Over-privileged access permissions and roles**
- **Exception handling and error messages**

SECURITY TESTING TOOLS AND SOLUTIONS

- PureSec
  - End-to-end application security solution for serverless
- Protego
  - Automated solution, Web-based UI application with security-focused visualizations
- Snyk
  - Open Source, Scan, find and fix the vulnerabilities
- DivvyCloud
  - Comprehensive set of tools for Identity (IAM), Security, and Compliance
  - Helps with Classification, Monitoring and Analytics

BALANCING COST AND PERFORMANCE

- **Performance Advantages:** Continuous auto-scaling, high availability and low latency
- **Cost Advantages:** No idle cost, no operational costs and pay per execution
- Use Kubernetes-native Serverless Framework like Fission (Open-Source), Kubeless

- **Dealing with Cold Starts**
  - resource reusing, runtime pooling, prefetching, prewarming
  - Hardware optimized for faster prefetching from memory and warm storage

BEST APPLICATIONS

- Microservices
- Mobile Backends
- Bots, ML Inferencing
- IoT
- Service integration

Not great for:

- Deep Learning Training
- Spark/Hadoop Analytics
- Simulation
- Video Streaming

REFERENCES

- https://martinfowler.com/articles/serverless.html
- https://www.eweek.com/innovation/predictions-2018-why-serverless-processing-may-be-wave-of-the-future
- https://learntocodewith.me/posts/serverless-architecture/
- https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology-media-telecommunications/Serverless%20Computing.pdf
- https://opensourceforu.com/2017/10/serverless-architectures-demystifying-serverless-computing/
- Serverless Security, What are we up against? (PNSQC 2019)

THANK YOU

AMITH.PULLA@INTEL.COM

TWITTER @PAMITH