

Knative vs Azure Functions

Ainul Habib

School of Enterprise Computing and Digital Transformation, TU Dublin, Ireland

X00159358@myTUDublin.ie

Introduction



The concept of **Serverless computing** has generated immense attention due to the promise of facilitating a **zero-administration** approach to application deployment, management, scalability and improve productivity by reducing delivery time. It intends to reduce infrastructure and operational cost of Cloud Solution, by scaling application instances to Zero when traffic is ideal or zero.



Microsoft's Azure Functions is a Serverless offering by Microsoft. It allows developers to deploy their Serverless application on Azure Cloud using a Pay-as-you-go model.



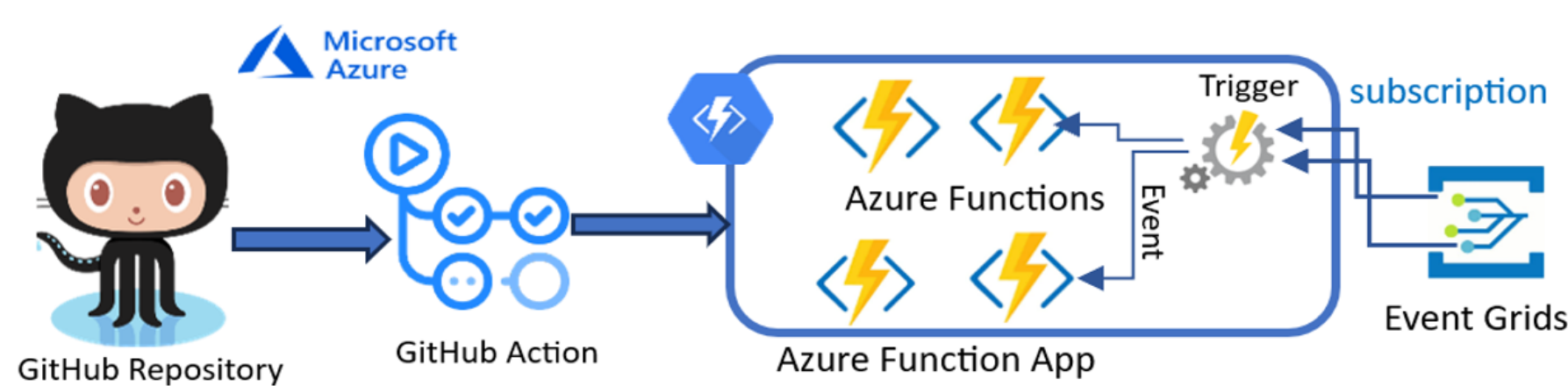
Knative, is an Enterprise level, Open-Source and Cloud-Native Solution framework, to build and run Serverless and Event Driven applications on a **Kubernetes Cluster**.



Proprietary Cloud Providers, like Microsoft Azure, Amazon AWS or Google Cloud, keeps the organization **Vendor locked by Design, Use and Licensing**, making them difficult to shift between cloud provider, reducing their opportunity to take advantages of new technological offering and discounts.

The study compares both **Knative** (a Cloud Native Serverless Platform) and **Azure Serverless** platforms. The study is based on several criteria, such as performance, Cold-Start, cost, adaptability, portability, development effort and maintenance.

Azure Function: Offering



Microsoft Azure provides a platform to build, deploy, manage and run Serverless Application in Azure Cloud.

Developers can use Azure DevOps Pipeline, GitHub Actions or Azure CLI to build and deploy their Serverless code to an **Azure Function App**, which groups Azure Functions as a logical manageable unit.

These Serverless functions are triggered by events from various Sources like Event-Grid, Event-Hub or by HTTP incoming requests.

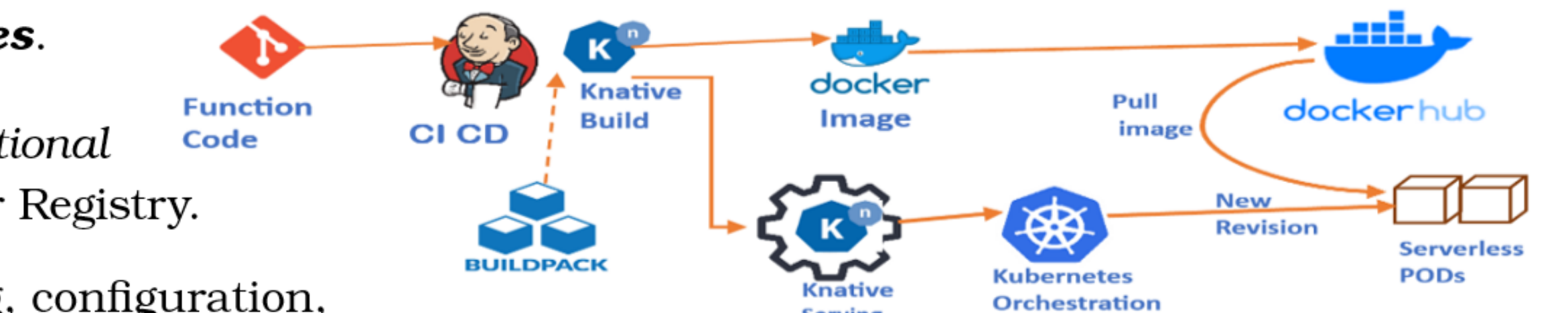
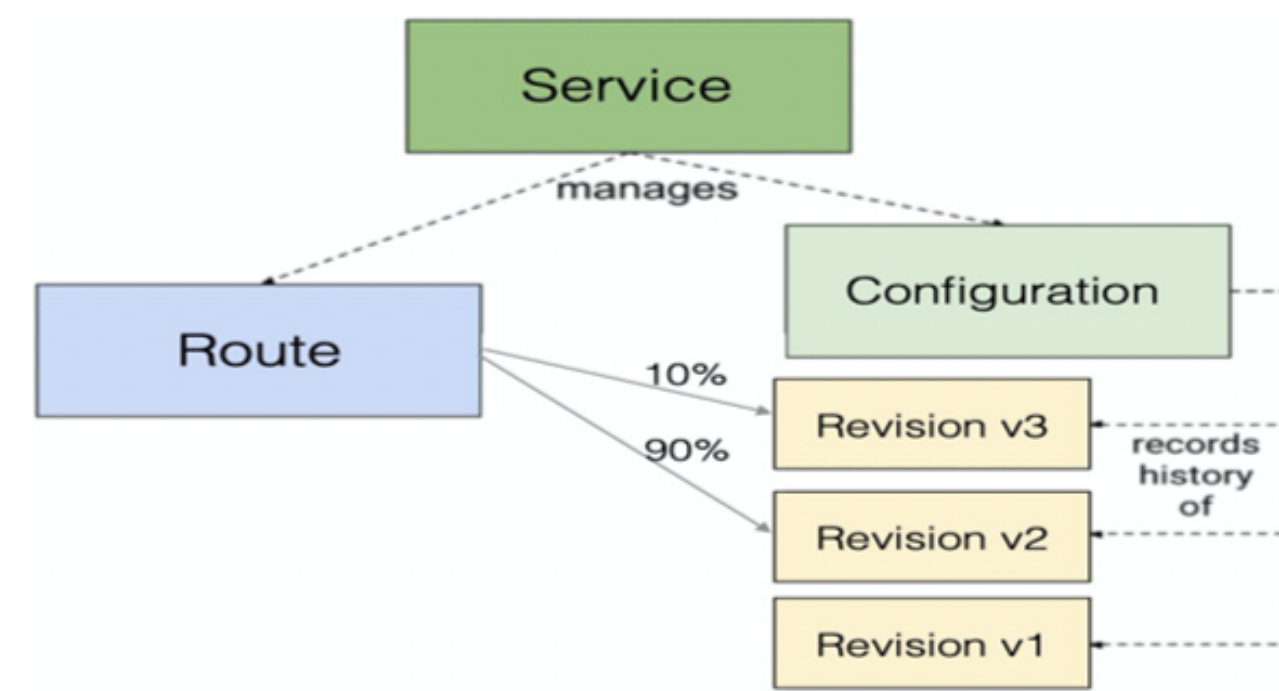
Knative: Cloud-Native Offering



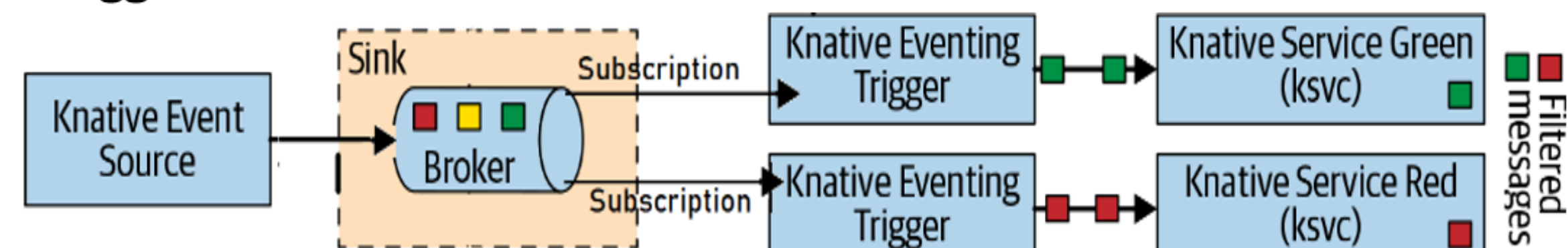
Knative provides a **Cloud-Native Platform** over Kubernetes, offering Build, Deployment, Orchestration and Management tools to create enterprise grade, auto-scalable, event-driven **Serverless** Application. It is offered as ready-to-use components built with Kubernetes **CRDs** (Custom Resource Definitions) and **Services**.

Knative Build is responsible in building "Functional Code" to a Docker Image, and push to a Docker Registry.

Knative Serving manages deployment, routing, configuration, revision and deployment of Serverless Function Containers as PODs



Knative Eventing is responsible for consuming events triggering the Serverless function. It consists of **Event-Source, Channel and Subscription**. Further abstraction is created by introducing **Brokers, Trigger and Filters**.



Topic Overview

- Cloud-Event Support:** Both Azure Function and Knative follows Serverless event-driven architecture, supporting "Cloud-Event" API. Developers only produce a functional code, containing business logic. It is invoked by Serverless framework, passing *Cloud-Event* object as parameter. "Cloud-Event" contain event payload and metadata. *No infrastructure code injection is needed.*
- Cold-Start:** All Serverless platform suffer from Cold-Start latency. It is delay in getting Serverless application scaled from **Zero** to **One**. **Application runtime** plays a major role in cold-start latency, e.g. Azure Function perform better using .Net runtime, while NodeJS produce lower start-up latency compared to Java. Application request latency may increase due to Cold-Start delay, effecting application performance. Azure Functions, under *Premium Plan*, keeps a "Warm instance" of Serverless, to reducing the cold-start latency. Knative also provide mitigation to cold-start latency by keeping at-least one instance alive.
- High Scalability:** Knative and Azure Functions scale horizontally, based on high volume of events metrics. These metrics includes "Concurrency", "TPS", "CPU" and "Memory", which can be configured for Serverless applications.
- Logging and Metrics:** Azure Functions and Knative emit logs to console-out, which must be streamed to external logging system. Knative emits many Kubernetes and custom metrics related to health and performance. Monitoring system Prometheus is deployed to capture and archive the required metrics. Application Insight is easily configured in Azure Platform to capture and present Azure Function's metrics.
- Build and Deployment:** Azure and Knative offer CLI to package and deploy the Serverless application to the Cloud. They also allow easy integration to DevOps pipeline and GitHub action. Knative CLI packages the function code and runtime to a docker image. Azure function offers many form of packaging and deployment like Zip, Docker Image, Cloud and Git Sync.

Conclusions and Future Work

Knative is a **Cloud-Native** alternative to traditional Serverless platform offered by public cloud providers like Azure and AWS. Customers can easily migrate their Knative Solutions from current public cloud Kubernetes platform to another.

Azure Function on other hand keeps the customers "vendor locked", making transition to different cloud provider difficult and costly. But Azure Functions lowers the maintenance cost by fully managing the platform, servers and infrastructures.

Future Works: A study into feasibility making the running cost of Knative applications to **Zero**.

QR Code for Recording

