What Is Knative? Overview, Working, Features, and Importance

SEO title: What Is Knative and How Does it Work? | Spiceworks

Excerpt:

Knative is a serverless environment built by Google (made open source in 2021) for easy code deployment to Kubernetes.

Meta-description:

Knative is a serverless environment built by Google (made open source in 2021) that enables code deployment to Kubernetes.

Introduction:

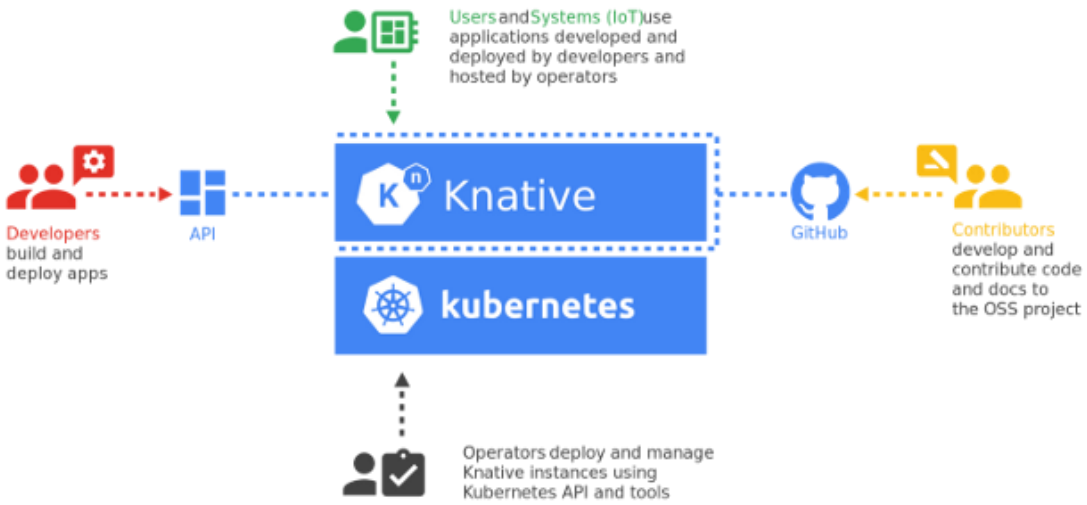
Knative is defined as a serverless environment initially developed by Google and then made open source in 2021, which enables code deployment to a Kubernetes platform with the explicit purpose of supporting microservices. This article explains how Knative works, its key features, and its importance for enterprises today.

Table of Contents

What Is Knative?

Knative is defined as a serverless environment initially developed by Google and then made open source in 2021, which enables code deployment to a Kubernetes platform with the explicit purpose of supporting microservices.

Reference image:



How Knative and Kubernetes Work Together

Image source: https://betterprogramming.pub/go-serverless-on-kubernetes-with-knative-b3aff3dbdffa

Knative is a publicly accessible project built based on the Kubernetes service and used for deploying and managing workloads on serverless platforms. To fully understand Knative, you must first understand the working and meaning of serverless architectural platforms and Istio Kubernetes.

Traditionally, every organization that ran a software or website application had at least one or a group of servers stored in a room that accepted and responded to requests made on the application or website. Maintaining physical servers required a lot of time, effort, and human resources and was capital intensive. This led to the development of what is now called serverless architecture.

Serverless architecture is a means of software development that allows developers to build, deploy and run applications or software on a cloud without needing physical servers. In reality, these programs are still running on servers, but the developer is not directly concerned with that. A third-party, the cloud provider is responsible for maintaining the server hardware while he ‘rents’ out the infrastructure to developers.

Kubernetes, on the other hand, is also an open source system like Knative, which is used for the general management of containerized applications across more than one platform. Kubernetes was initially built by Google and was based on several years of experience in running software and production. Kubernetes is a software, an automated platform that can manage applications and programs made up of multiple microservices.

In essence, Kubernetes manages, maintains, and scales workloads made up of several containers while keeping them in the desired state, fully functional, and interacting with each other to present a whole application.

See More: **What Is Cloud Migration? Definition, Process, Benefits, and Trends**

Understanding the use of Knative

With a proper understanding, we can now go back to what Knative is and how it works. Knative, pronounced as “Kay-nay-tive,” is a Kubernetes extension that increases the ease and functionality of running workload on Kubernetes with cloud architecture and provides the tools for easy deployment and building and managing microservice applications.

Knative stands for Kubernetes plus native – providing a smooth deployment and cohesive experience for containerized apps native to the Kubernetes platform.

Knative makes it possible and easy for serverless functions to be managed on Kubernetes. With Knative, you can build containers easily. These containers or microservices are packets of codes or mini software that possess all the necessary elements to be executed in any setting.

Over the years, there has been a sharp increase in the rate at which developers use containers in software development and management. It makes deploying changes, updates and features easier as the focus is narrowed down to a unique set of codes. Knative helps you deploy code, packaging it as a container and managing how it runs on the Kubernetes environment.

How Does Knative Work?

Knative works by being an event steering segment that connects Kubernetes and Istio. Kubernetes acts as the orchestrator for all containers and microservices. At the same time, Istio is an open-source mesh technology that brings the different components together to interact with themselves and the user. Knative provides users with several components targeted at carrying out basic day-to-day tasks that might be too bulky and mundane for developers to keep up with.

These components can be used over and over again in different applications. This is because they do not control application-specific processes – e.g., routing, load balancing during deployments, scaling, etc. Knative allows developers to use any programming language. Thus, it does not require knowledge of a specific language as what it recognizes are container images.

There are three main components of Knative that are key to its functioning. These components are:

1. Build

Knative has a build component that is responsible for the building of new containers. The build component can convert public accessible source codes to a container. In this function, Knative is not only flexible but can be configured to meet specific requirements. To build, Knative first pulls out the chosen source code from a code library like GitHub.

After that, underlying dependencies, including software libraries, are added so the code can run effectively. Next, container images are constructed and placed in files that the Kubernetes platform can access. The container is also made available to other developers using Knative and Kubernetes. Thus, with Knative, containers are built automatically as long as the source code location is known.

2. Serving

Knative consists of a serving component that is fundamental to the running of the platform. This entails running containers as a part of Knative services, and it involves:

- **Configuration:** Configuration is specific to managing different versions of a service. Every time there is the deployment of a new feature or modification to the existing features of a container, Knative saves the previous version and creates a new version with the latest features and changes. In addition, Knative maintains and defines the state of a service,
- **Auto-scaling:** For serverless containers to work well, one must be able to autoscale them either up or down. Knative can autoscale services to thousands if needed or scale down services to one or even zero.
- **Intelligent service routing:** Intelligent service routing is an essential part of the working mechanism of Knative. It helps developers direct the amount and flow of traffic to various existing versions of a microservice. It can be used when introducing new features and mimics the canary and blue-green deployment strategies. Intelligent service routing allows you to expose only a fraction of users to a recent version, testing how they respond and then gradually routing more traffic to the new version.

3. Eventing

Eventing describes the function of Knative that allows it to define the running of a container based on specific events. In other words, different events can trigger specific container-based functions. Developers simply define event triggers and their associated containers, and Knative does the work. Knative also handles the list of events a user can pick from (channels) and the delivery of events to respective containers.

Top 5 Features of Knative

The five features of Knative that make it a must-have solution for cloud engineers and developers are:

1. Autoscaling

Automatic scaling is a highly critical feature offered by Knative. It is necessary so that applications can meet up with the peripheral subscriber demands. It has a default Knative pod autoscaling (KPA) that performs the job of auto-scaling.

Here is how Knative auto-scaling works – if an application or service is not getting any traffic from users, the Knative pod autoscaler scales down to zero, which means it reduces the replicas of the application on the cloud to zero. If there is a small amount of traffic, the replicas are scaled down to the minimum number needed to handle requests. In the same way, one can scale replicas of an application to meet up with any surge in traffic and requests. This is done automatically without the developer having to monitor the process.

Knative, however, allows the user to enable or disable the scale functions for a cluster once the user has cluster-admin permissions. Knative has its default KPA but can also work with Kubernetes horizontal pod autoscaler (HPA). However, the Knative serving must be installed for HPA to run as it is not part of the Knative serving core.

KPA allows applications to be scaled down to zero, while HPA does not. Lastly, KPA (while part of their serving core and runs by default) does not support CPU-based auto-scaling. HPA, on the other hand, promotes CPU-based auto-scaling.

2. Traffic management and progressive rollout

Knative also offers a feature that is used to manage traffic routing to different versions of an application. This feature is responsible for the intelligent service routing in the service component of Knative. The administrator can then manage incoming traffic and route them to different service versions by changing the traffic specification of the service.

At initial creation, Knative microservices do not have default traffic specifications. The admin then sets the spec and shares traffic among various revisions. Traffic can also be sent solely to the latest revisions by selecting the latest ones. Progressive rollout allows you to control the speed with which new versions handle the incoming load.

3. Load balancing

Knative also has load balancing, which one can turn in by instructing the activator service to act as a load balancer. This requires that each pod’s addressability is turned on. The activator pods are scaled horizontally, and there could be several activators in every deployment. The best way to run the Knative platform is by having a whole number of multiple revision pods compared to the number of activator pods.

The algorithm behind activator load balancing works dependent on the number of concurrencies. When it's unlimited, requests are directed to the more preferred option out of two random choices. If the concurrency is set to a value of three or less, the activator pod then directs the requests to the initial pod that can handle it.

4. Developer-friendly software

One outstanding feature of Knative, perhaps one that has added to its increasing popularity, is its developer-friendly characteristics. This is not just because it is used to manage serverless applications on Kubernetes. It gives the developer reusable components and sections of codes that one uses to solve regular, boring, and bulky tasks.

Some of these – like load balancing, traffic management, containment building using source codes, auto-scaling, etc. – are just ways it helps lessen the developers' burden. Lastly, developers can also access all the features on the platform in a language they are most comfortable with as it is not language specific.

5. Flexibility and control

Knative is designed to be flexible and offer a good amount of control and regulation of its functions. Knative can plug in easily to already built continuous integration and deployment (CI/CD) toolchains. Developers can quickly move their workload to any cloud or infrastructure that best suits their application as long as that infrastructure is supported by Kubernetes and thus by association Knative.

This gives flexibility and control necessary to adapt any application as demand evolves. For example, you can manage your workload on Google Kubernetes Engine and later move to cloud run. With Knative as the underlying platform, switching is straightforward and far less expensive.

Importance of Knative

Knative has proven to be a technology that is indispensable to many software developers and programmers. It has also been applied to many use cases that further delineate its importance. Thus, the importance of Knative includes the following:

1. Provides support for the management of the Kubernetes serverless platform

Knative offers services like route management, gradual and progressive release, and microservice connections essential for serverless operations. It supports Kubernetes serverless orchestration and provides services such as route management, service connection, and phased release.

2. Simplifies Kubernetes adoption for businesses

One of the most significant benefits of Knative is its suitability for business users embracing serverless architecture. To begin with, it has an extensive community backing it, which makes it easier to troubleshoot issues and build innovations. Also, despite not being a platform as a service (PaaS), it allows you to create your own PaaS systems through serverless orchestration. Finally, it is a flexible framework., which means that there is no risk of vendor lock-in as businesses can entirely leverage open source resources.

3. Enables widespread compatibility

Knative is known for its widespread compatibility with other function as a service (FaaS) solutions. This is because it utilizes the standard CloudEvent framework, and has standard events, making it compatible with other FaaS systems. Also, it has built-in cross-platform support, which is helpful for enterprises operating in a hybrid cloud environment. Its universal standards are interoperable across multiple cloud providers, and there is no need to tether a single Knative instance to a particular cloud.

4. Covers the end-to-end code deployment cycle

Knative is a valuable solution for building a cloud app across the entire software development lifecycle (SDLC). Thanks to its mature serverless design, you can transform every aspect of building a microservices-based application from source code to deploying images. Its inherent design is tailored for proportional phased releases, characteristic of agile delivery. Using Knative, you can recreate this serverless deployment experience in a containerized environment.

5. Supports IoT data

Knative is especially important for the Internet of Things (IoT) implementations. Apps built using Knative can efficiently process IoT data, and its event-based model is suitable for use cases like registrations, subscriptions, and connecting to external systems. It also validates the configuration of IoT device security groups.

6. Reduces developer effort

Knative makes Kubernetes developers more productive by removing repetitive configuration and build tasks. Knative is a good fit for any development team having trouble keeping track of increasing Kubernetes clusters. If you use Kubernetes, it’s easy to switch to other cloud providers that work with Knative. Also, you can use Knative to manage the workflow as a service and have it run automatically. This frees developers from focusing on writing code and easily deploying new containers with little effort.

7. Accelerates the journey to serverless

Serverless setups may be challenging to establish and administer manually. Knative helps enterprises to build up serverless workloads swiftly. Regarding developers, they're simply constructing a container — it is Knative executing it as a serverless operation.

8. Empowers Agile and DevOps teams

Knative makes it simpler to deploy containerized apps in short, iterative stages as part of a DevOps or Agile process. It achieves this by allowing developers to generate new containers and container versions rapidly. In addition, Knative services are simple to include in automated CI/CD pipelines and do not need specialized tools or custom scripting.

9. Enables smooth rollouts for new features

Rolling out new versions to users might uncover software vulnerabilities that have the potential to impact business operations. The configuration and routing capabilities of Knative enable developers to reveal new container upgrades to a portion of the user base, then progressively enlarge that audience as they resolve difficulties - or swiftly revert to prior versions if required.

10. Maintains the focus on innovation and coding

When it comes down to it, software developers and DevOps engineers would rather concentrate on creating bug-free software and cutting-edge new features than setting message bus queues for triggering events or maintaining the scalability of container environments. Knative enables developers to spend more time focusing on the tasks they excel at by removing the need for environmental governance.

Takeaway

With serverless adoption continually rising, Knative is an essential tool for cloud app developers. According to Datadog's 2022 report, 50% of customers across Amazon Web Services (AWS), Azure, and Google Cloud Platform use serverless technology. That is why it is so important to know about Knative, its features, and benefits – so you can maximize the power of your Kubernetes ecosystem in your cloud workflows.

Did this article help you understand how Knative works? Tell us on Facebook, Twitter, and LinkedIn. We'd love to hear from you!