Knative: Kubernetes Framework to manage Serverless Workloads

Nikhil Barthwal
Product Manager (Serverless), Google Cloud

@nikhilbarthwal
nikhilbarthwal@yahoo.com
www.nikhilbarthwal.com
Introduction to Knative
Knative

Kubernetes based open source building blocks for serverless
Serverless model

Operational Model
- No Infra Management
- Managed Security
- Pay only for usage

Programming Model
- Service-based
- Event-driven
- Portable
Kubernetes: The defacto platform

**Scheduling**
Decide where my containers should run

**Lifecycle and health**
Keep my containers running despite failures

**Scaling**
Make sets of containers bigger or smaller

**Naming and discovery**
Find where my containers are now

**Load balancing**
Distribute traffic across a set of containers

**Storage volumes**
Provide data to containers

**Logging and monitoring**
Track what’s happening with my containers

**Debugging and introspection**
Enter or attach to containers

**Identity and authorization**
Control who can do things to my containers
Kubernetes ecosystem

5,000+ Contributors
47k+ GitHub stars

*Sources: COCOMO Model, CNCF.io Certified Providers
Developers want serverless

... just want to run their code.

... want to use their favorite languages and dependencies.

... don't want to manage the infrastructure.

Operators want Kubernetes

Kubernetes is great orchestrating microservices

They love using GKE and not having to do operations for Kubernetes.

Kubernetes is not the right abstraction for their developers.
Containers

- Any Language
- Any Library
- Any Binary
- Ecosystem of base images
Containers: An industry standard

![Graph showing interest over time for docker container]

Note

- Apr 13, 2017
- Sep 24, 2017
Portability

**Kubernetes**
Offered by virtually **all**
Cloud Service Providers

**Knative**
Codifies serverless, **broad**
contributor/user community
Knative Community

v0.8
Predictable Releases

~450
Individual Contributors

55+
Contributing Companies

9
Working Groups

>6K
Pull Requests

source: knative.teststats.cncf.io/d/8/dashboards
**Knative project**

- Set of components (serving, eventing, build)
- Ingredients for Serverless
- Solves for modern development patterns
- Implements learnings from Google, partners
Knative stack

Products
- Google Cloud Run
- Red Hat OpenShift
- SAP Kyma
- Google Cloud Run on GKE
- IBM Cloud Kubernetes Service
- TriggerMesh

Primitives
- Serving
- Events
- Build Tekton

Service Mesh
- Istio
- Gloo
- Ambassador*  
  *No eventing

Platform
- Kubernetes
Knative for developer

Want to: Write code

Don’t have to
• Build docker image
• Upload image to registry
• Deploy service
• Expose to the internet
• Setup logging & monitoring
• Scale workload...
**Knative for operator**

**Abstracts** operational complexity, smooth infrastructure surface

**Universal** supported by all major Cloud providers, enables portability

**Extendable** platform with clear separation of concerns between operator and developer
Serverless on Google Cloud

Cloud Run
Fully managed, deploy your workloads and don’t see the cluster.

Cloud Run on GKE
Deploy into your GKE cluster, run serverless side-by-side with your existing workloads.

Knative Everywhere
Use the same APIs and tooling anywhere you run Kubernetes with Knative.
Knative Serving
Knative Serving

**What is it?**
- Rapid deployment of serverless containers
- Automatic (0-n) scaling
- Configuration and revision management
- Traffic splitting between revisions

**Pluggable**
- Connect to your own logging & monitoring platform, or use the built-in system
- Auto-scaler can be tuned or swapped out for custom code
Knative Serving Primitives

**Knative Service**
High level abstraction for the application

**Configuration**
Current/desired state of an application
Code & configuration separated (a la 12-factor)

**Revision**
Point in time snapshots for your code and configuration

**Route**
Maps traffic to revisions
**Knative Eventing**

**What is it?**

- For loosely coupled, event-driven services
- Declaratively bind between event producers and Knative services
- Scales from just few events to live streams
- Custom event pipelines to connect with your own existing systems

---

**Event types**

- KubernetesEventSource
- GitHubSource
- GcpPubSubSource
- AwsSqsSource
- ContainerSource
- CronJobSource

---

**Flow**

- Event source
- Event type
- Event consumer(s)
Knative Eventing

**Namespace**

- **Source**: Events
- **Trigger**: filter=
- **Broker**: ingress
- **Service**: (Callable)

**Flow**

1. Source Events → Broker
2. Broker ingress → Trigger filter= ✓ subscribe ✓
3. Trigger filter= ✓ subscribe ✓ Service (Callable)

**Events**

- Publish
- Ingress

---

**Source**

- Events

**Service**

- (Callable)
Knative Events

CloudEvent

```
{
    "specversion": "0.2",
    "type": "com.github.pull.create",
    "source": "https://github.com/cloudevents/spec/pull/123",
    "id": "A234-1234-1234",
    "time": "2019-04-08T17:31:00Z",
    "datacontenttype": "application/json",
    "data": "{ GitHub Payload... }"
}
```
Knative Eventing: Benefits

- Declaratively bind between event producers and deployed services
- Scales from just few events to live streams
- Custom event pipelines to connect with your own existing systems
## Knative Event Sources

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Apache Camel</strong></td>
<td>Allows to use Apache Camel components for pushing events into Knative</td>
</tr>
<tr>
<td><strong>Apache Kafka</strong></td>
<td>Brings Apache Kafka messages into Knative</td>
</tr>
<tr>
<td><strong>AWS SQS</strong></td>
<td>Brings AWS Simple Queue Service messages into Knative</td>
</tr>
<tr>
<td><strong>Cron Job</strong></td>
<td>Uses an in-memory timer to produce events on the specified Cron schedule.</td>
</tr>
<tr>
<td><strong>GCP PubSub</strong></td>
<td>Brings GCP PubSub messages into Knative</td>
</tr>
<tr>
<td><strong>GitHub</strong></td>
<td>Brings GitHub organization/repository events into Knative</td>
</tr>
<tr>
<td><strong>GitLab</strong></td>
<td>Brings GitLab repository events into Knative</td>
</tr>
<tr>
<td><strong>Google Cloud Scheduler</strong></td>
<td>Google Cloud Scheduler events in Knative when jobs are triggered</td>
</tr>
<tr>
<td><strong>Google Cloud Storage</strong></td>
<td>Brings Google Cloud Storage bucket/object events into Knative</td>
</tr>
<tr>
<td><strong>Kubernetes</strong></td>
<td>Brings Kubernetes cluster/infrastructure events into Knative</td>
</tr>
</tbody>
</table>

[https://github.com/knative/docs/tree/master/docs/eventing/sources](https://github.com/knative/docs/tree/master/docs/eventing/sources)
Building
Knative Build (Pre 0.8)

→

Tekton Pipelines (Post 0.8)
What is it?

- Go from source code to container images on repositories
- Build pipelines can have multiple steps and can push to different registries
- Builds run in containers in the cluster. No need for Docker locally

Primitives

- **Build**: Represents a single build job with 1 or more steps.
- **BuildTemplate**: A set of ordered and parameterized build steps.
- **ServiceAccount**: For auth with DockerHub etc
Tekton Pipelines

Primitives

- Task: Represents the work to be executed with 1 or more steps
- TaskRun: Runs the Task with supplied parameters
- Pipeline: A list of Tasks to execute in order
- ServiceAccount: For authentication with DockerHub etc.

What is it?

Kubernetes style resources for declaring CI/CD-style pipelines

Go from source code to container images on repositories

Build pipelines can have multiple steps and can push to different registries

Builds run in containers in the cluster. No need for Docker locally

github.com/tektoncd/pipeline
Knative: Ready for you!

Install, Samples, Docs
![GitHub](github.com/knative/docs)

Have questions?
[knative.slack.com](knative.slack.com)

Want to contribute?
[knative/docs/community](knative/docs/community)

Anything else?
[@nikhilbarthwal](@nikhilbarthwal)
Questions?
Rate today’s session

Cyberconflict: A new era of war, sabotage, and fear

We’re living in a new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you’re often the collateral damage in a growing conflict among states. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the dens of Chinese, Russian, North Korean, and Iranian hackers to the boardrooms of Silicon Valley, David reveals a world coming face-to-face with the perils of technological revolution—a conflict that the United States helped start when it began using cyberweapons against Iranian nuclear plants and North Korean missile launches. But now we find ourselves in a conflict we’re uncertain how to control, as our adversaries exploit vulnerabilities in our hyperconnected nation and we struggle to figure out how to deter these complex, short-of-war attacks.

David Sanger
The New York Times

David E. Sanger is the national security correspondent for the New York Times as well as a national security and political contributor for CNN and a frequent guest on CBS This Morning, Face the Nation, and many PBS shows.

O'Reilly Events App