Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Technology
(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Department of Information Technology

IT-Bulletin

May-2022

'Kubernetes Serverless simply visually explained'

13th Apr 2022

HIGHLIGHTS

We explore ways of making Kubernetes serverless in a straight-forward and tool-agnostic way.

What this article is about

- Will your hard-learned Kubernetes knowledge become obsolete through serverless and did you waste 3 years of your life?
- How to use serverless without becoming cloud provider locked-in?
  This isn't a comparison of specific tools, but rather general ideas.

TL;DR

Serverless on Kubernetes reduces repetitive configuration in a cloud provider independent way. It's just the result of continuously auto-mating away manual work. When we're talking about serverless on Kubernetes we need to consider two different areas:

1. **Deploying applications serverless in the cluster** (reduce the number of YAML files needed for every app + auto-building containers).
2. **Running pods' containers serverless** without managing nodes/ VMs

Serverless

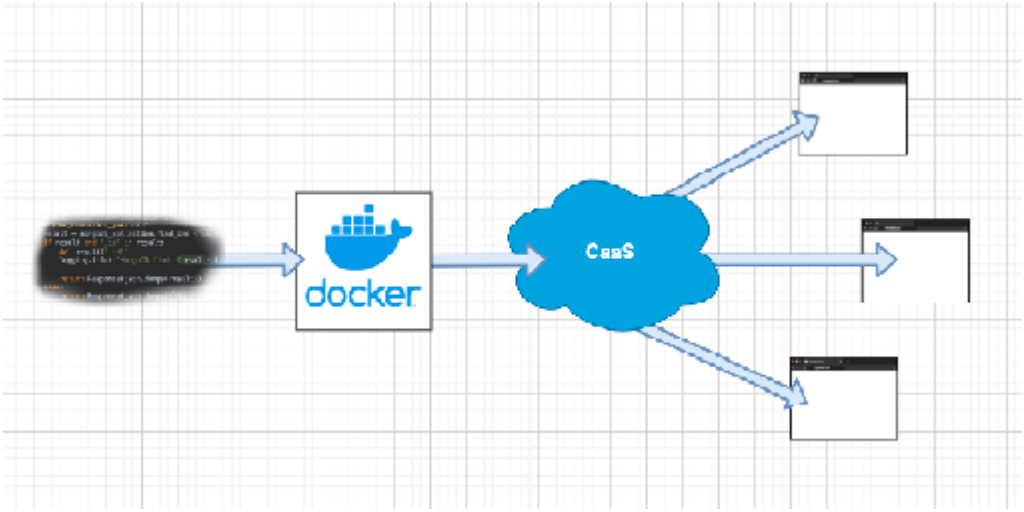
In short: you don't or interact much less with the servers and

'Kubernetes Serverless simply visually explained'

infrastructure necessary to run your applications. When using the word "serverless" today it can point to two different things: CaaS and FaaS.

CaaS — Container as a Service

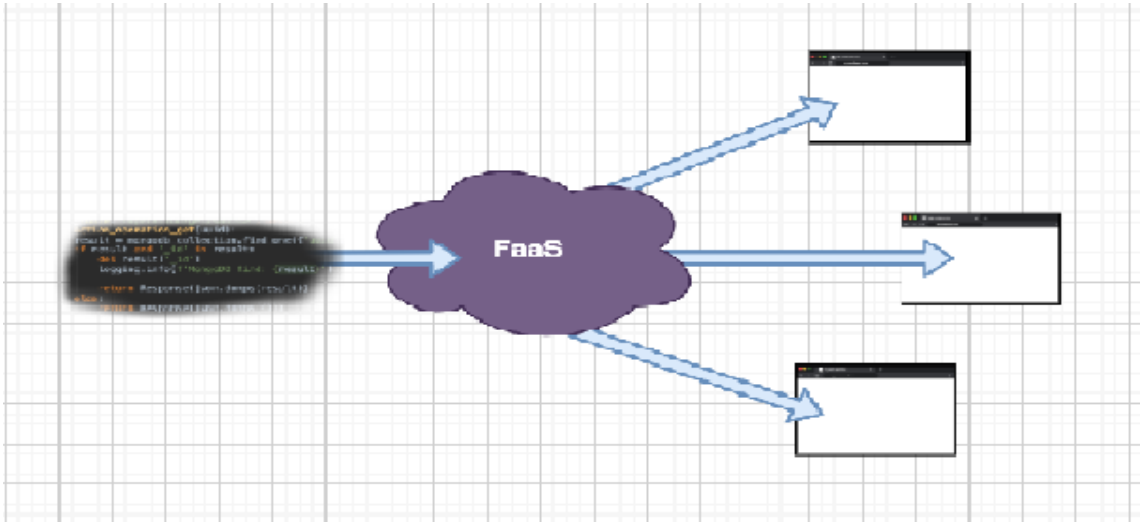
You create the (Docker) container, throw it at the CaaS and it runs, serves and scales it automatically.



Managed examples are Azure Container Instances, Google Cloud Run or AWS Fargate.

FaaS — Function as a Service

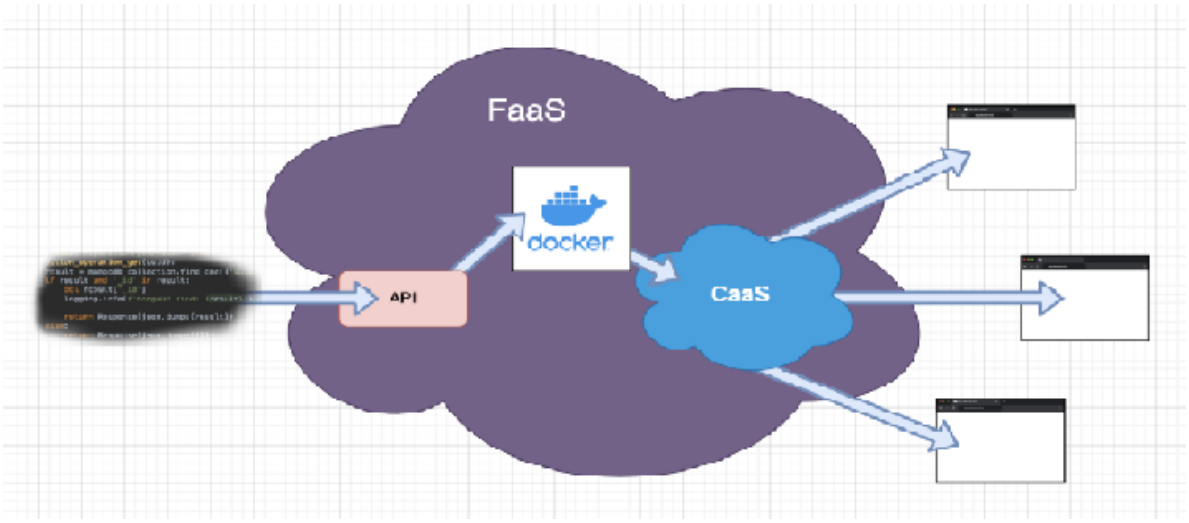
You write code, throw it at the FaaS and it runs, serves and scales it automatically.



‘Kubernetes Serverless simply visually explained’

FaaS implementations

How the FaaS runs your code can happen in different ways. **One way** could be that the FaaS actually builds a container for every code change and then uses a CaaS like this:

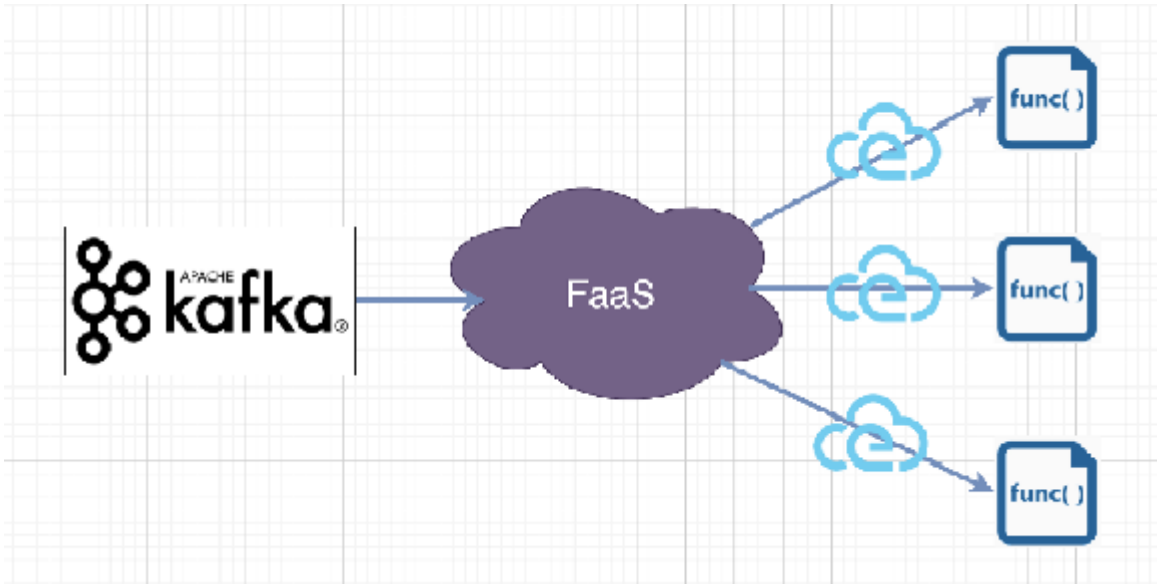


Another way could be that the FaaS pulls the function’s source code into a pre-defined environment (container) dynamically during boot. Environments would be available for different languages. When using a language which has to be compiled like Go, then compilation also has to be done during boot.

Events / Scaling

FaaS are most of the time used in conjunction with eventing systems which trigger the instantiation of the functions. Events can originate from API-Gateways, Github, Kafka, RabbitMQ, CronJobs etc.

'Kubernetes Serverless simply visually explained'



For every event, a new function will be created to handle it. If there are multiple events happening at the same time multiple instances will be created to handle these. This way we have automatic scaling.

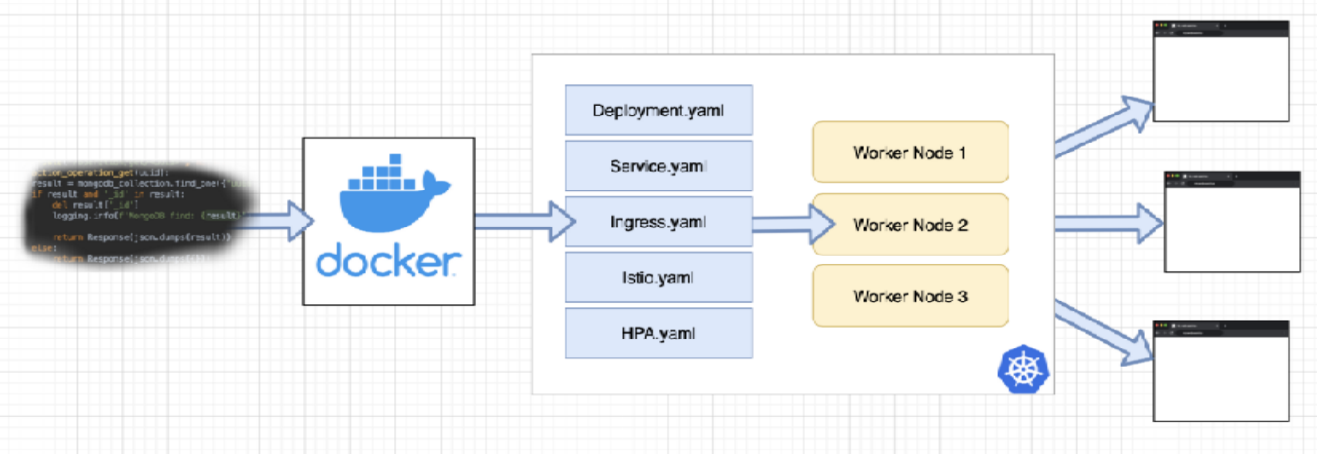
The FaaS communicates with the various event sources, so the functions themselves don't have to. They only have to work with one event format the FaaS uses, like CloudEvents or transport via HTTP.

There is the CloudEvents project which describes the structure and metadata of events as a "standard". It also includes data and a schema which describes that data. Cloud events are envelopes that wrap around the event data. This could be great if adopted by many vendors to gain interoperability.

‘Kubernetes Serverless simply visually explained’

Kubernetes Applications

Let’s have a look at steps necessary to develop a traditional **non-serverless** application running on Kubernetes:



quite a bit of manual “server” interactions necessary

We need to build a container, create various Kubernetes resources (YAML files) an then decide how many worker nodes we need to run our app.

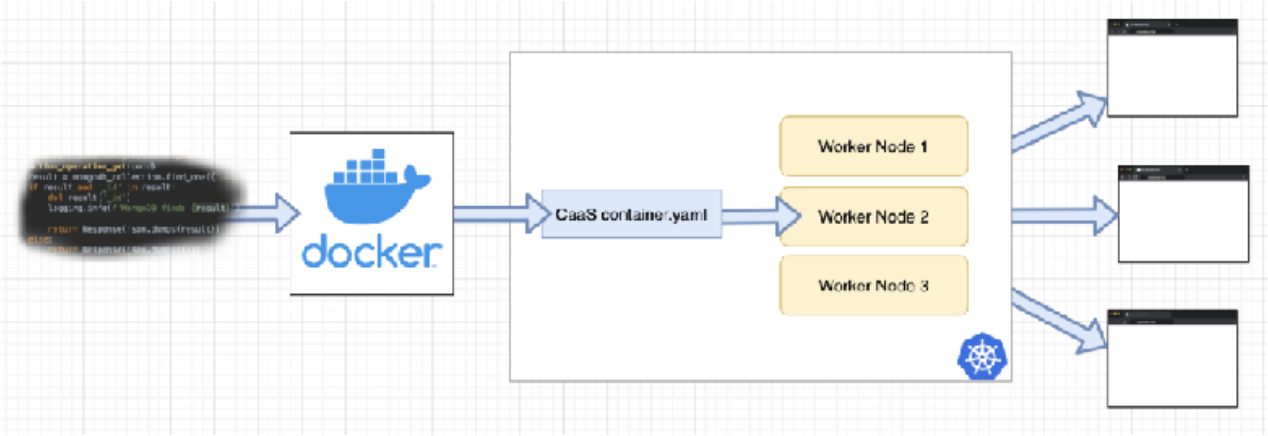
The decision of how many worker nodes we need could be handled more dynamically by configuring a Cluster/Node autoscaler. Though even then we still have to configure it and need to set a min+max amount of nodes.

We interact with “servers” quite a bit with this traditional approach. First creating/ building a container, then writing YAML files and also defining the amount and resources of the nodes.

Kubernetes Serverless Applications

Now let’s explore serverless approaches when developing apps for Kubernetes.

‘Kubernetes Serverless simply visually explained’



we reduced the creation of a lot of YAML files

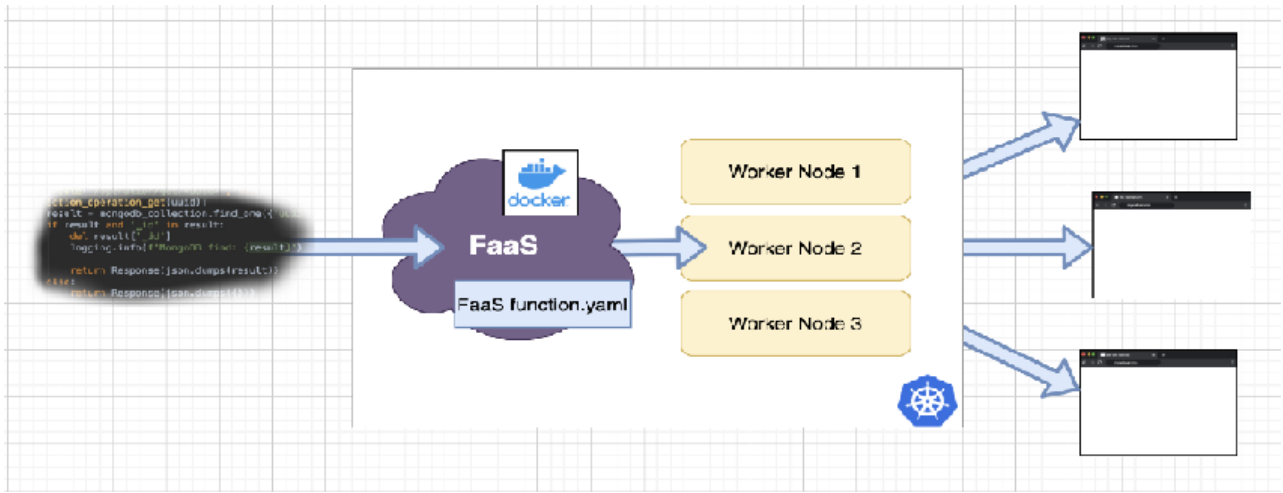
Here we reduce the amount of Kubernetes resources (YAML files) to be created significantly. The CaaS will create all necessary subresources for us, like autoscaler, Ingress or Istio routing.

All we do is provide a (Docker) container and create one single k8s resource, the CaaS-container resource introduced via a CRD. The CaaS decides when to start instances of our app and how many, maybe based on events or on our definitions.

We have to make sure that the container that we build can receive and handle the events coming from the CaaS which can be through HTTP or CloudEvents for example. This may require certain libraries inside the container.

CaaS Example: Knative (Knative provides flexible building blocks which other solutions can use and depend on).

FaaS — Function as a Service



we now also automate the build process and maybe have a nice FaaS webinterface

'Kubernetes Serverless simply visually explained'

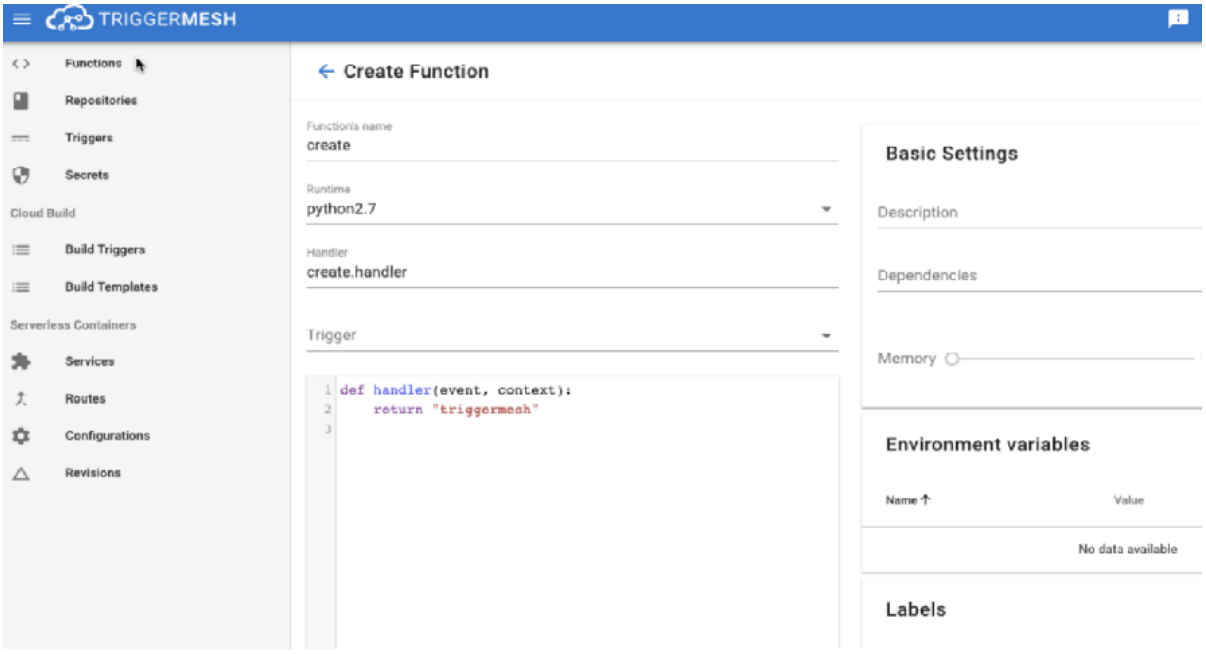
The FaaS function.yml will contain one K8s resource from the FaaS system, introduced via a CRD. In that resource, we set things like the function name, location of source code, language runtime and trigger events.

If we upload code via a web interface then creating the FaaS function.yaml wouldn't be necessary. But keeping functions as code should be good practice. Webinterfaces are good for prototyping or testing modifications.

With a FaaS we also have everything that a CaaS solution provides. But now we reduce the work even further because we have tools running in our Kubernetes cluster which can execute/build our application source code directly.

The container that is built for us already includes the necessary libraries, like HTTP or CloudEvents, to receive events from the FaaS. We don't have to worry about that.

The source might be stored in a Git repo, is uploaded via a web interface or is available someplace else. The FaaS will access the code, listen to changes, build the container and then pass it to the CaaS for serving end eventing.



example web interface of Trigger Mesh to upload code and deploy as a function

'Kubernetes Serverless simply visually explained'

One K8s FaaS example, Fission, doesn't actually build immutable containers for every function's code change but uses the idea of mutable environment containers ("Generic pods") which pull code in dynamically to then convert these into "Specific Function pods". I think this is also they way AWS Lambda works utilising AWS Firecracker.

Observability

Moving from containerized microservices to functions could result in having to manage even more and smaller services than before. This is caused by the ease to create small functions which just listen to and handle one single event. I noticed this myself when I created the same demo app as Container Microservices in Part 1 and AWS Serverless in Part 2.

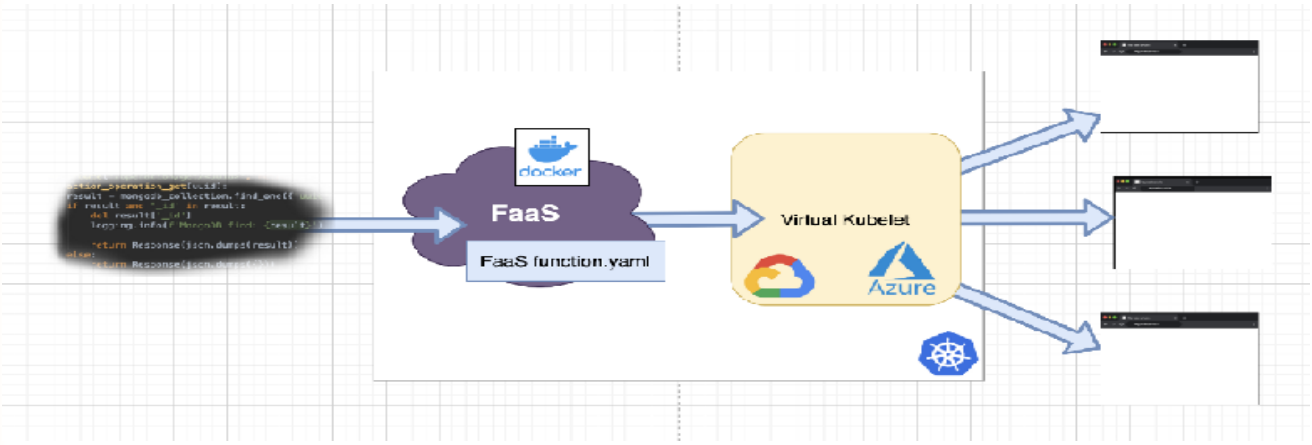
To manage the higher amount of services or functions its necessary to keep observability (metrics, logging, tracing). This is why most Kubernetes FaaS and CaaS integrate already with for example Prometheus, Jaeger and service mesh like Istio or Linkerd.

Kubernetes Serverless Nodes

In the previous section we talked about K8s serverless applications and we saw workflows when using CaaS or FaaS. These simplify processes and reduce repetitive work by a lot.

But the developers or operators are still interacting with servers: the VMs used as worker nodes in the cluster. They still need to specify how many nodes to have and their resources (CPU/memory).

Now we go one step further and make the actual underlying Kubernetes nodes serverless using the Virtual Kubelet.



FaaS on top of Kubernetes with Virtual Kubelet as nodes. The ultimate setup?

'Kubernetes Serverless simply visually explained'

FaaS Examples:

1.TriggerMesh (uses Knative as CaaS)
2.OpenFaaS (can use Knative)

3.Kubeless

4. Fission (works with environments rather than immutable function containers, more further down)

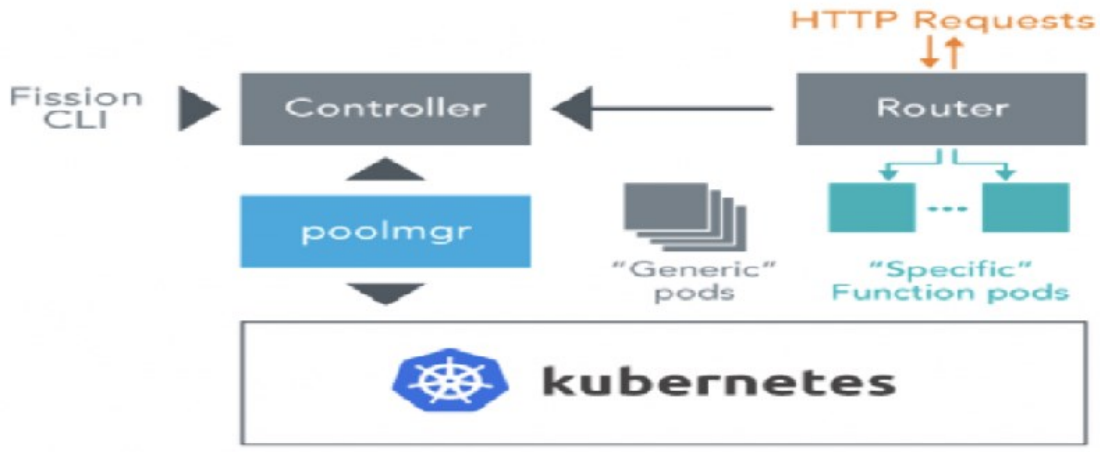
5. Open Whisk

6. overview and comparison of various K8s FaaS

Cold and warm starts

A **cold start** would mean that no pod is already running to handle an event, so it'll take some time creating it. Usually, these pods will stay alive for a bit after they were used last and can be reused. Invocations during the "already-running" period will be called a **warm start**. Warm starts are faster but also cost resources.

Fission / Environment based FaaS



'Kubernetes Serverless simply visually explained'

The Virtual Kubelet simulates a worker node to Kubernetes which can then be used to schedule pods on, like with any other normal node. Though the pods' containers aren't running on a VM, but in the Serverless Container offerings of cloud providers like AWS Fargate, Google Cloud Run or Azure Container Instances. The combination of K8s serverless applications and K8s serverless nodes for Kubernetes could be a powerful one. But why then still use K8s at all if we serverless everything away?

Why still use Kubernetes?

Kubernetes provides great and flexible building blocks rather than being made for easy interaction and end-users. This makes K8s complex and requires a lot of repetitive work when using directly.

Kubernetes became a cloud provider independent standard. Using serverless frameworks on top of it makes sense to keep this independence when going serverless. We can always define our applications in more detail if necessary because it's still just running K8s under the hood.

By using serverless on K8s we can reduce repetitive work by a very high percentage, which then allows us to spend more time on building the actual applications.

Conclusion

I think modern serverless event-driven architecture has already proven itself and will become more and more prevalent in the coming years. Let's work on using it on top of open standards like K8s to ensure greater interoperability.

Serverless on Kubernetes is just the result of continuously automating away manual work.

Sources

1.The internet.

2.A very quick comparison of Kubernetes Serverless Frameworks

3. Gitlab uses Knative and Triggermesh to offer Serverless applications

4. Google Cloud Run uses Knative under the hood

Scientific advance leads to a new tool in the fight against hackers

28 April 2022

A new form of security identification could soon see the light of day and help us protect our data from hackers and cybercriminals. Quantum mathematicians at the University of Copenhagen have solved a mathematical riddle that allows for a person's geographical location to be used as a personal ID that is secure against even the most advanced cyber attacks.

People have used codes and encryption to protect information from falling into the wrong hands for thousands of years. Today, encryption is widely used to protect our digital activity from hackers and cybercriminals who assume false identities and exploit the internet and our increasing number of digital devices to steal from us.

As such, there is an ever-growing need for new security measures to detect hackers posing as our banks or other trusted institutions. Within this realm, researchers from the University of Copenhagen's Department of Mathematical Sciences have just made a giant leap.

"There is a constant battle in cryptography between those who want to protect information and those seeking to crack it. New security keys are being developed and later broken and so the cycle continues. Until, that is, a completely different type of key has been found." , says Professor Matthias Christandl.

For nearly twenty years, researchers around the world have been trying to solve the riddle of how to securely determine a person's geographical location and use it as a secure ID. Until now, this had not been possible by way of normal methods like GPS tracking.

"Today, there are no traditional ways, whether by internet or radio signals for example, to determine where another person is situated geographically with one hundred percent accuracy. Current methods are not unbreakable, and hackers can impersonate someone you trust even when they are far far away. However, quantum physics opens up a few entirely different possibilities," says Matthias Christandl.

Quantum physics makes hacking impossible

Using the laws of quantum physics, the researchers developed a new security protocol that uses a person's geographical location to guarantee that they are communicating with the right person. Position-based quantum encryption, as it is called, can be used to ensure that a person is speaking with an actual bank representative when the bank calls and asks a customer to make changes to their account.

Scientific advance leads to a new tool in the fight against hackers

28 April 2022

"Ask yourself, why do I trust an employee at the bank counter? Because they're in a bank. Their location creates trust. This explains the principle behind pposition-based cryptography, where physical location is used to identify oneself," explains postdoc Andreas Bluhm.

The researchers' recipe for securing a person's location combines the information in a single quantum bit -- a qubit -- followed by classical bits, consisting of the ones and zeroes that we are familiar with from ordinary computers.

Both types of bits are needed to send a message that is impossible for cybercriminals to read, hack or manipulate, and which can confirm whether a person is in your bank's office or in some far-off country.

The quantum bit serves as a kind of lock on the message, due to the role of Heisenberg's Uncertainty Principle in quantum physics, which causes quantum information to be disrupted and impossible to decode when trying to measure it. It is also due to what is known as the "no-cloning theorem," which makes quantum information impossible to intercept and secretly copy. This will remain the case for quite some time.

"Until a full-fledged quantum computer is built and hackers gain access to one, our method is completely secure and impossible to hack," says Andreas Bluhm.

Could soon be a reality

The researchers highlight the fact that the new method is particularly handy because only a single quantum bit is needed for position verification. So, unlike many other quantum technologies that require further development, this new discovery can be put to use today. Suitable quantum sources that can send a quantum bit of light already exist.

"The particular strength of our technique is that it is relatively straightforward to implement. We're already able to send individual quantum bits, which is all this technique requires," says Matthias Christandl.

The security ID needs to be developed commercially, by a company for example, before it can be widely adopted. However, its quantum foundation is in place.

The new research result is particularly useful in contexts where communications between two parties need to be extremely secure. This could be payments on the internet or transmission of sensitive personal data.

Scientific advance leads to a new tool in the fight against hackers

28 April 2022

"Secure communication is a key element of our daily lives. Whenever we communicate with public authorities, our banks or any party that manages our personal data and information, we need to know that the people we're dealing with are those who we expect them to be -- and not criminals," says Andreas Bluhm.

Story Source:

Materials provided by **University of Copenhagen - Faculty of Science**. *Note: Content may be edited for style and length.*

Journal Reference:

Andreas Bluhm, Matthias Christandl, Florian Speelman. **A single-qubit position verification protocol that is secure against multi-qubit attacks**. *Nature Physics*, 2022; DOI: 10.1038/s41567-022-01577-0

University of Copenhagen - Faculty of Science. "Scientific advance leads to a new tool in the fight against hackers." ScienceDaily. ScienceDaily, 28 April 2022. <www.sciencedaily.com/releases/2022/04/220428125430.htm>.

Scientific advance leads to a new tool in the fight against hackers

TOP