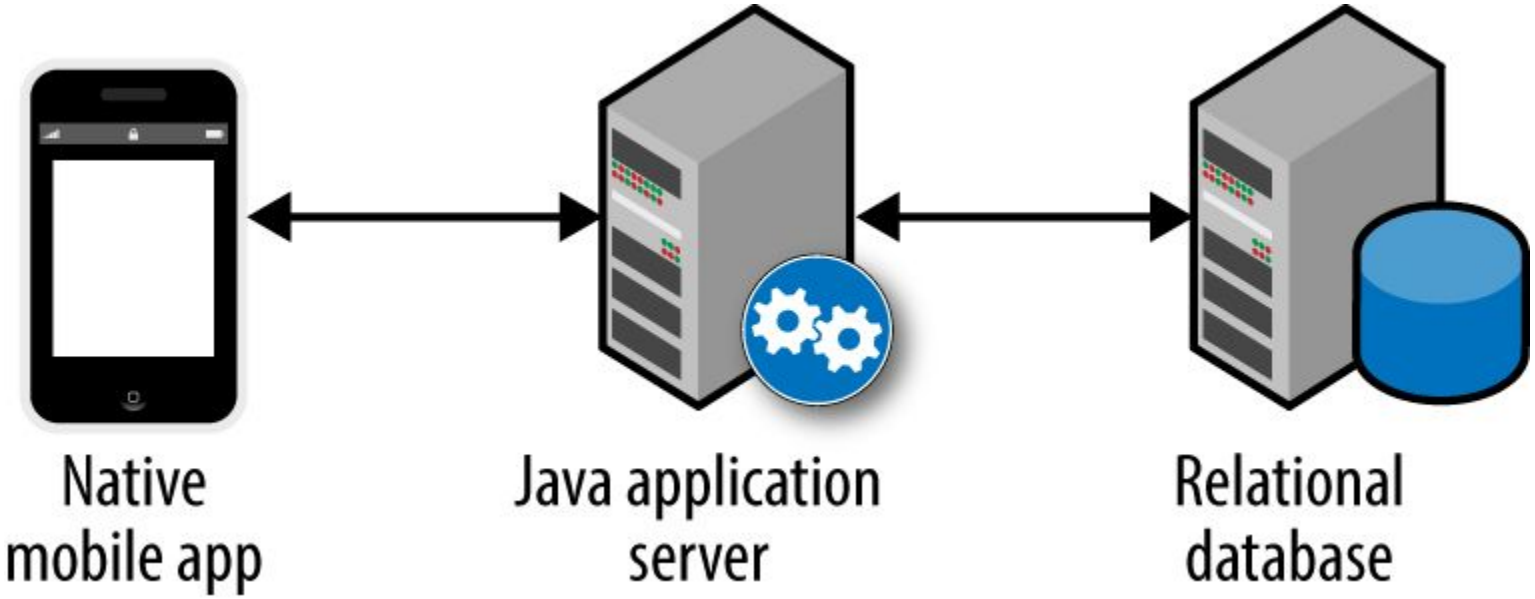
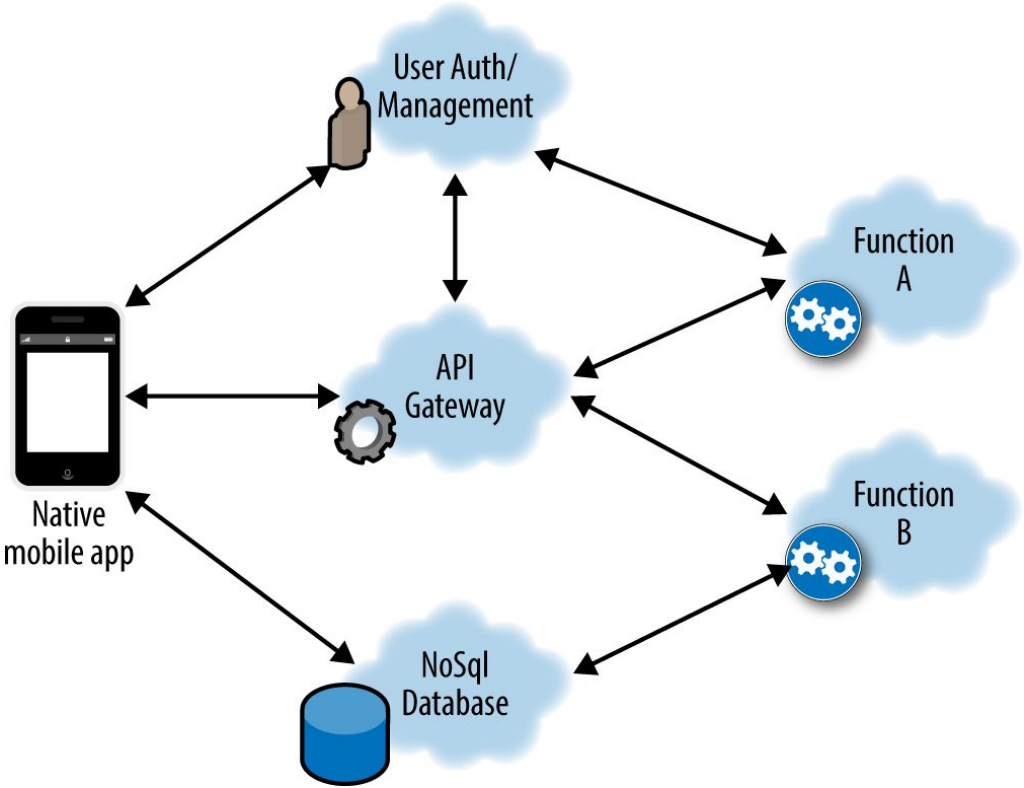
Integrating Serverless Computing (FaaS) to k8s Cluster using Terraform

Ilham – SysOps Engineer
ilham@btech.id

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

So what is Serverless?


Native mobile app
Java application server
Relational database


User Auth/
Management
Function
A
API
Gateway
Native
mobile app
Function
B
NoSql
Database

What are the strengths or benefit of Serverless?

Benefits Serverless

- Reduced labor cost
- Reduced resource cost
- Increased flexibility of scaling
- Shorter lead time

Limitations Serverless

- State
- Latency
- Local Testing
- Loss Of Control
- Vendor Lock-In

<>

Serverless Use Cases Recommendation

- You need apps that are all set to perform tasks but don’t need to be running all the time
- You have IoT-based applications that get activated only in certain situations
- Development speed and cost optimization are of prime importance
- There is a need for auto-scaling
- There is a requirement for tying into legacy systems that handle specific business logic
- You need to handle high-volume backend tasks for websites and mobile apps

Mobile phone takes picture and uploads it to object storage in the cloud.

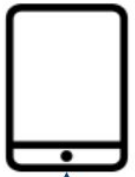
Various side photos are uploaded to an S3 bucket on AWS

S3 Event triggers an Amazon Lambda function.

Amazon Lambda function resizes image.

Thumbnails are written to another S3 bucket on AWS.

Thumbnails are server to a mobile application from a webserver and s3.



What is FaaS - Function as a Service ?

Fully Managed Compute

Provisioning, patching, scaling, monitoring, logging are provided out-of-the- box

Deploy You Code

Just package and upload the code

Pay for actual usage

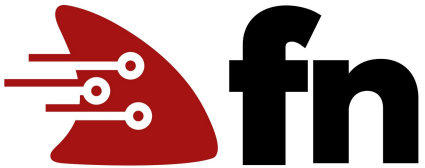
Getting charged only upon code execution,

FaaS On Public Cloud & Open Sources




QINLING
an OpenStack Community Project

OPENFAAS



fn

fission

APACHE
OpenWhisk

Biznet GioCloud
Biznet
Mellanox TECHNOLOGIES
BANK BRI
OSF OpenStack Foundation

CNCF Serverless Landscape

2019-10-31T04:02:55Z ecd3da9

See the serverless interactive display at s.cncf.io

Greyed logos are not open source

Tools

CLOUD NATIVE Landscape, CLOUDZERO, dashbird, epsagon, HASURA GraphQL Engine, IO|pipe, Iron.io, lumigo, Node Lambda, SCAR, SIGMA, STACKERY, THUNDRA

Security

Protego, threat stack

Framework

APEX, Architect, AWS SAM, CHALICE, Claudia.js, dapr, FLOGO, serverless, SPARTA, Spring Cloud Function

Hosted

Installable

Platform

ALGORITHMIA, Azure Functions, BINARIS, CLOUDFLARE Workers, Google Cloud Functions, HUAWEI FunctionStage, IBM Cloud Functions, netlify Functions, nimbella, AWS Lambda, Nuweba, PubNub FUNCTIONS, spotinst, Standard Library, Tencent Cloud, twilio Functions, ZEIT

Apache Camel, Kyma, Apache OpenWhisk, nuclio, AppScale, OPENFAAS, fission, PipelineAI, fn, riff, K Knative, Virtual Kubelet, Kubeless

s.cncf.io

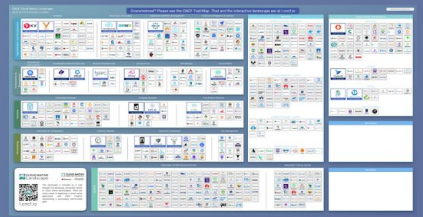
Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. This landscape illustrates a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment

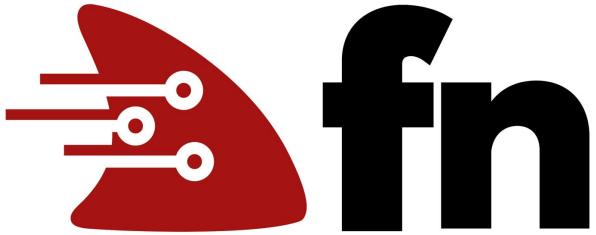
CLOUD NATIVE Landscape

CLOUD NATIVE COMPUTING FOUNDATION

Redpoint

Cloud Native Landscape





The **Fn project** is an open-source container-native serverless platform that you can run anywhere -- any cloud or on-premise.

It’s easy to use, supports every programming language, and is extensible and performant.

Why Choose

- Open Source
- Multi Cloud
- Container Native
- Orchestrator Agnostic
- There is a requirement for tying into legacy systems that handle specific business logic
- You need to handle high-volume backend tasks for websites and mobile apps

How To Use ?

```
sudo apt install docker.io
sudo systemctl enable --now docker

sudo wget https://github.com/fnproject/cli/releases/download/0.5.91/fn_linux -O
/usr/local/bin/fn

sudo chmod +x /usr/local/bin/fn

sudo fn -h
sudo fn start &


sudo fn init --runtime python --trigger http pythonfn

sudo fn create app pythonapp


## TESTING

sudo fn invoke pythonapp pythonfn

curl -H "Content-Type: application/json" http://localhost:8080/t/pythonapp/pythonfn
```

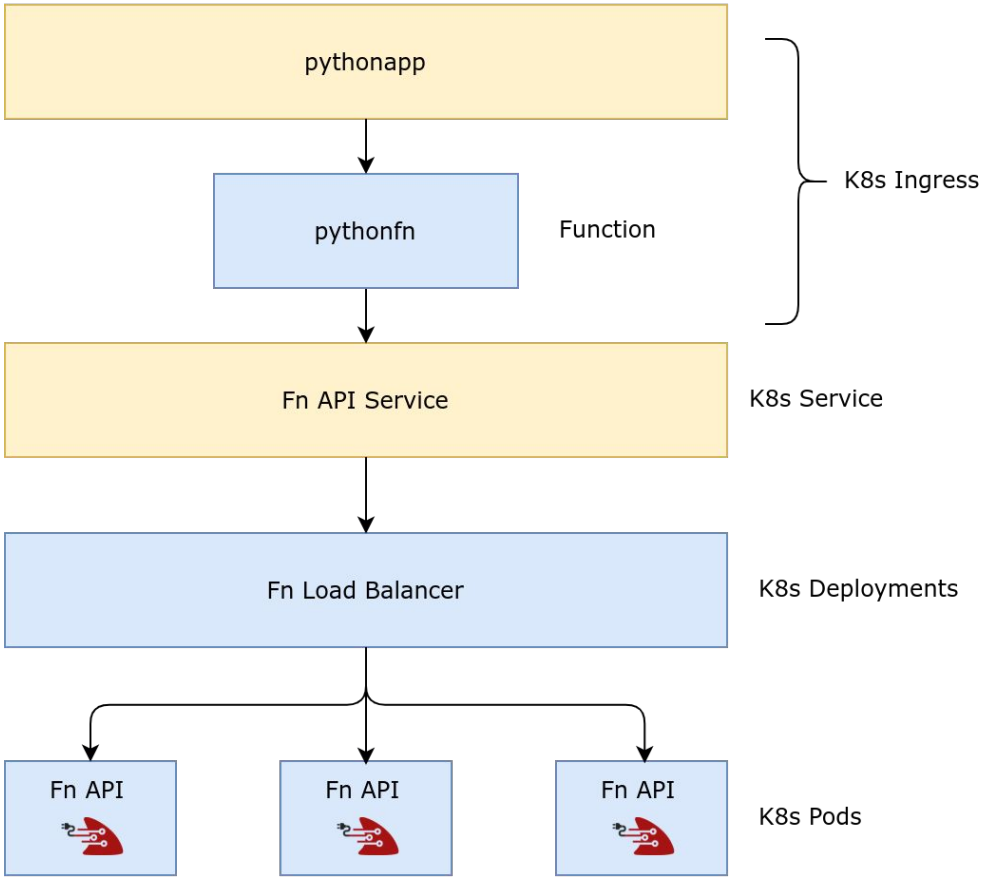
Integrate with k8s Cluster ?

```
sudo snap install helm --classic
git clone git@github.com:fnproject/fn-helm.git && cd fn-helm
helm init --service-account tiller --history-max 200
helm dep build fn
helm install --name my-release fn

kubectl apply -f
https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/mand
atory.yaml
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/prov
ider/baremetal/service-nodeport.yaml

kubectl get pods && kubectl get deployments
kubectl get svc && kubectl get ingress
```

```
ubuntu@k8s-master:~$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
my-release-fn-5f9d75f5d5-66rfz          2/2     Running   0          21h
my-release-fn-5f9d75f5d5-hnmfv          2/2     Running   1          3d4h
my-release-fn-flow-depl-864f675f75-zcbtc   1/1     Running   4          3d4h
my-release-fn-runner-79c4d6b6f5-7rnft   1/1     Running   0          3d4h
my-release-fn-runner-79c4d6b6f5-ch6z9   1/1     Running   0          3d4h
my-release-fn-runner-79c4d6b6f5-dkv6n   1/1     Running   0          3d4h
my-release-fn-ui-78bc5444d7-487ph       1/1     Running   0          3d4h
my-release-mysql-69fdc44cc6-5tv7s       1/1     Running   0          3d4h
my-release-redis-fc4d9c757-52gxk        1/1     Running   0          3d4h
ubuntu@k8s-master:~$ kubectl get deployments
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
my-release-fn          2/2     2            2           3d4h
my-release-fn-flow-depl   1/1     1            1           3d4h
my-release-fn-runner   3/3     3            3           3d4h
my-release-fn-ui       1/1     1            1           3d4h
my-release-mysql       1/1     1            1           3d4h
my-release-redis       1/1     1            1           3d4h
ubuntu@k8s-master:~$
```

```
ubuntu@k8s-master:~$ kubectl get svc
NAME                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes            ClusterIP   10.96.0.1        <none>        443/TCP          5d1h
my-release-fn         ClusterIP   10.99.15.45      <none>        80/TCP,90/TCP    3d4h
my-release-fn-flow    ClusterIP   10.109.197.165   <none>        80/TCP           3d4h
my-release-fn-runner  ClusterIP   10.101.76.79     <none>        9191/TCP         3d4h
my-release-fn-ui      ClusterIP   10.108.151.53    <none>        3000/TCP         3d4h
my-release-mysql      ClusterIP   10.96.218.4      <none>        3306/TCP         3d4h
my-release-redis      ClusterIP   10.100.34.135    <none>        6379/TCP         3d4h
ubuntu@k8s-master:~$
```

http://my-release.lb.fn.internal/t/pythonapp/pythonfn



```
ubuntu@k8s-master:~$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
my-release-fn-5f9d75f5d5-66rfz          2/2     Running   0          21h
my-release-fn-5f9d75f5d5-hnmfv          2/2     Running   1          3d5h
my-release-fn-flow-depl-864f675f75-zcbtc   1/1     Running   4          3d5h
my-release-fn-runner-79c4d6b6f5-7rnft   1/1     Running   0          3d5h
my-release-fn-runner-79c4d6b6f5-ch6z9   1/1     Running   0          3d5h
my-release-fn-runner-79c4d6b6f5-dkv6n   1/1     Running   0          3d5h
my-release-fn-ui-78bc5444d7-487ph       1/1     Running   0          3d5h
my-release-mysql-69fdc44cc6-5tv7s       1/1     Running   0          3d5h
my-release-redis-fc4d9c757-52gxk        1/1     Running   0          3d5h
ubuntu@k8s-master:~$ kubectl get hpa
NAME     REFERENCE                  TARGETS          MINPODS   MAXPODS   REPLICAS   AGE
hpa-fn   Deployment/my-release-fn   4%/10%, 3%/10%   2         10        2          22h
ubuntu@k8s-master:~$ kubectl get hpa
NAME     REFERENCE                  TARGETS           MINPODS   MAXPODS   REPLICAS   AGE
hpa-fn   Deployment/my-release-fn   8%/10%, 19%/10%   2         10        4          22h
ubuntu@k8s-master:~$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
my-release-fn-5f9d75f5d5-66rfz          2/2     Running   0          21h
my-release-fn-5f9d75f5d5-dx5lg          2/2     Running   0          20s
my-release-fn-5f9d75f5d5-hnmfv          2/2     Running   1          3d5h
my-release-fn-5f9d75f5d5-vwtxj          2/2     Running   0          20s
my-release-fn-flow-depl-864f675f75-zcbtc   1/1     Running   4          3d5h
my-release-fn-runner-79c4d6b6f5-7rnft   1/1     Running   0          3d5h
my-release-fn-runner-79c4d6b6f5-ch6z9   1/1     Running   0          3d5h
my-release-fn-runner-79c4d6b6f5-dkv6n   1/1     Running   0          3d5h
my-release-fn-ui-78bc5444d7-487ph       1/1     Running   0          3d5h
my-release-mysql-69fdc44cc6-5tv7s       1/1     Running   0          3d5h
my-release-redis-fc4d9c757-52gxk        1/1     Running   0          3d5h
ubuntu@k8s-master:~$
```

Automate Integrate to k8s Cluster ?

```
ubuntu@inputmhs:~/k8s$ tree
.
├── file
│   ├── config
│   └── id_rsa
├── fn-helm
│   ├── CONTRIBUTING.md
│   ├── LICENSE
│   ├── README.md
│   └── fn
│       ├── Chart.yaml
│       ├── charts
│       │   ├── mysql-0.3.0.tgz
│       │   └── redis-1.0.4.tgz
│       ├── requirements.lock
│       ├── requirements.yaml
│       ├── templates
│       │   ├── NOTES.txt
│       │   ├── _helpers.tpl
│       │   ├── cluster-issuer.yaml
│       │   ├── flow-deployment.yaml
│       │   ├── flow-service.yaml
│       │   ├── fn-deployment.yaml
│       │   ├── fn-ingress.yaml
│       │   ├── fn-runner-node-deployment.yaml
│       │   ├── fn-runner-node-service.yaml
│       │   ├── fn-service.yaml
│       │   ├── ui-deployment.yaml
│       │   └── ui-service.yaml
│       └── values.yaml
│   └── test.sh
├── main.tf
├── providers.tf
├── terraform.tfstate
├── terraform.tfstate.backup
└── variables.tf

5 directories, 29 files
ubuntu@inputmhs:~/k8s$
```

```
ubuntu@inputmhs:~/k8s$ cat providers.tf
provider "kubernetes" {
   version     = "1.8"
   config_path = "${var.k8s_config_path}"
}


provider "helm" {
    kubernetes {
        config_path = "${var.k8s_config_path}"
    }
}
ubuntu@inputmhs:~/k8s$ cat variables.tf
variable "k8s_config_path" {
    default = "/home/ubuntu/k8s/file/config"
}

variable "private_keys" {
    default = "/home/ubuntu/k8s/file/id_rsa"
}


variable fn_cluster_name {
    default = "kubernetes"
}

variable dns_zone {
   default = "openinfra"
}

variable namespace_cert_manager {
    default = "cert-manager"
}

variable namespace_external_dns {
    default = "ingress"
}

variable namespace_fn {
    default = "fn-openinfra"
}
```

```
resource "helm_release" "fn" {
    name       = "fn"
    chart      = "fn-helm/fn"
    namespace  = "${var.namespace_fn}"
    set {
        name  = "nameOverride"
        value = "${var.fn_cluster_name}"
    }
    set {
        name  = "fn_lb_runner.service.port"
        value = "90"
    }
    set {
        name  = "fn_lb_runner.service.ingress_hostname"
        value = "lb.${var.dns_zone}"
    }
    set {
        name  = "fn_api.service.ingress_hostname"
        value = "api.${var.dns_zone}"
    }
    set {
        name  = "ui.service.ingress_hostname"
        value = "ui.${var.dns_zone}"
    }
    set {
        name  = "fn_runner.resources.requests.cpu"
        value = "200m"
    }
    set {
        name  = "fn_runner.resources.requests.memory"
        value = "5Gi"
    }
    set {
        name  = "fn_runner.resources.limits.cpu"
        value = "200m"
    }
    set {
        name  = "fn_runner.resources.limits.memory"
        value = "5Gi"
    }
}
```

```
ubuntu@k8s-master:~$ kubectl get deployment -n fn-openinfra
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
openinfra-fn            1/1     1            1           2m39s
openinfra-fn-flow-depl  1/1     1            1           2m39s
openinfra-fn-runner     3/3     3            3           2m39s
openinfra-fn-ui         1/1     1            1           2m39s
openinfra-mysql         1/1     1            1           2m39s
openinfra-redis         1/1     1            1           2m39s
ubuntu@k8s-master:~$ kubectl get pods -n fn-openinfra
NAME                                    READY   STATUS    RESTARTS   AGE
openinfra-fn-59ddf5d695-89n6c           2/2     Running   0          2m44s
openinfra-fn-flow-depl-6b94b985b9-l2b6v 1/1     Running   2          2m44s
openinfra-fn-runner-854fc67dc4-5tbpm    1/1     Running   0          2m44s
openinfra-fn-runner-854fc67dc4-j6hg8    1/1     Running   0          2m44s
openinfra-fn-runner-854fc67dc4-sm7jr    1/1     Running   0          2m44s
openinfra-fn-ui-6c6cd84c4f-tc5n7        1/1     Running   0          2m44s
openinfra-mysql-7c6ff775b6-gg9rw        1/1     Running   0          2m44s
openinfra-redis-5558c9d9-sjq7m          1/1     Running   0          2m44s
ubuntu@k8s-master:~$ kubectl get svc -n fn-openinfra
NAME                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
openinfra-fn        ClusterIP   10.99.61.49      <none>        80/TCP,90/TCP    2m55s
openinfra-fn-flow   ClusterIP   10.109.130.199   <none>        80/TCP           2m55s
openinfra-fn-runner ClusterIP   10.105.28.22     <none>        9191/TCP         2m55s
openinfra-fn-ui     ClusterIP   10.99.195.170    <none>        3000/TCP         2m55s
openinfra-mysql     ClusterIP   10.102.203.241   <none>        3306/TCP         2m55s
openinfra-redis     ClusterIP   10.99.136.65     <none>        6379/TCP         2m55s
ubuntu@k8s-master:~$ kubectl get ingress -n fn-openinfra
NAME                        HOSTS                                                                    ADDRESS        PORTS   AGE
openinfra-fn-ingress-rules  openinfra.api.fn.internal,openinfra.lb.fn.internal,openinfra.ui.fn.internal + 1 more...   10.109.35.49   80      2m59s
ubuntu@k8s-master:~$ kubectl describe ingress -n fn-openinfra
Name:             openinfra-fn-ingress-rules
Namespace:        fn-openinfra
Address:          10.109.35.49
Default backend:  default-http-backend:80 (<none>)
Rules:
  Host                      Path  Backends
  ----                      ----  --------
  openinfra.api.fn.internal
                            /   openinfra-fn:80 (10.244.3.17:80)
  openinfra.lb.fn.internal
                            /   openinfra-fn:90 (10.244.3.17:90)
  openinfra.ui.fn.internal
                            /   openinfra-fn-ui:3000 (10.244.2.29:3000)
  openinfra.flow.fn.internal
                            /   openinfra-fn-flow:80 (10.244.3.18:8081)
Annotations:
```

Jadi?

Serverless adalah tentang tidak diperlukannya manage server, semua tentang pemanfaatan service, Kita tidak lagi dipusingkan oleh limitasi server, sehingga proses bisnis berjalan dengan baik berapapun traffic yang datang.

INDONESIA OpenInfra Days

Thank You!

https://github.com/ilosaurus

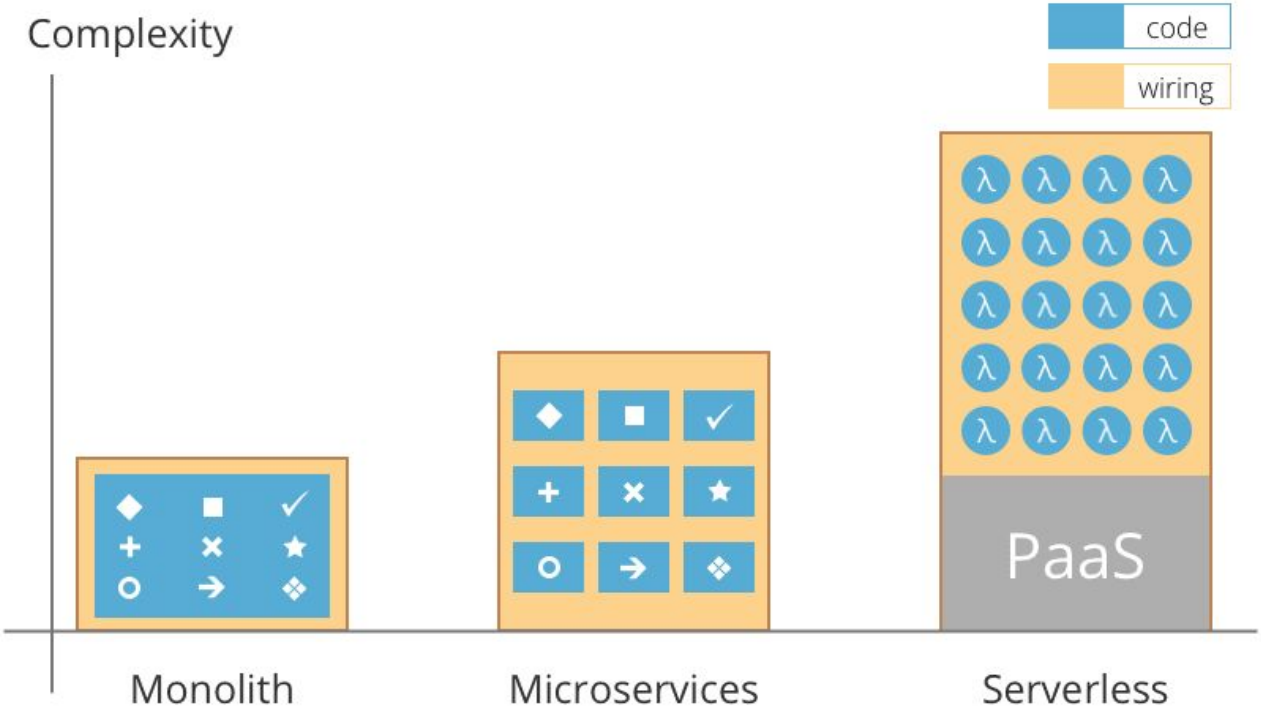
Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

What Is Serverless?
by Michael Roberts and John Chapin
Published by O'Reilly Media, Inc.

Serverless Computing vs. Containers
by Ankit Kumar (dzone.com)

Fn Project
by Peter Jausovec (hackernoon.com)

Fn Project
by fnproject team (medium.com)



https://dev.to/jignesh_simform