

## Ubuntu

Linux tabanlı bir işletim sistemidir. GNU lisansı olan açık kaynak yazılımdır.

Linux sistemindeki her şey dosyadır. Yazılımlar \*.deb ya da .snap uzantılı dosyalar halinde sunulur.

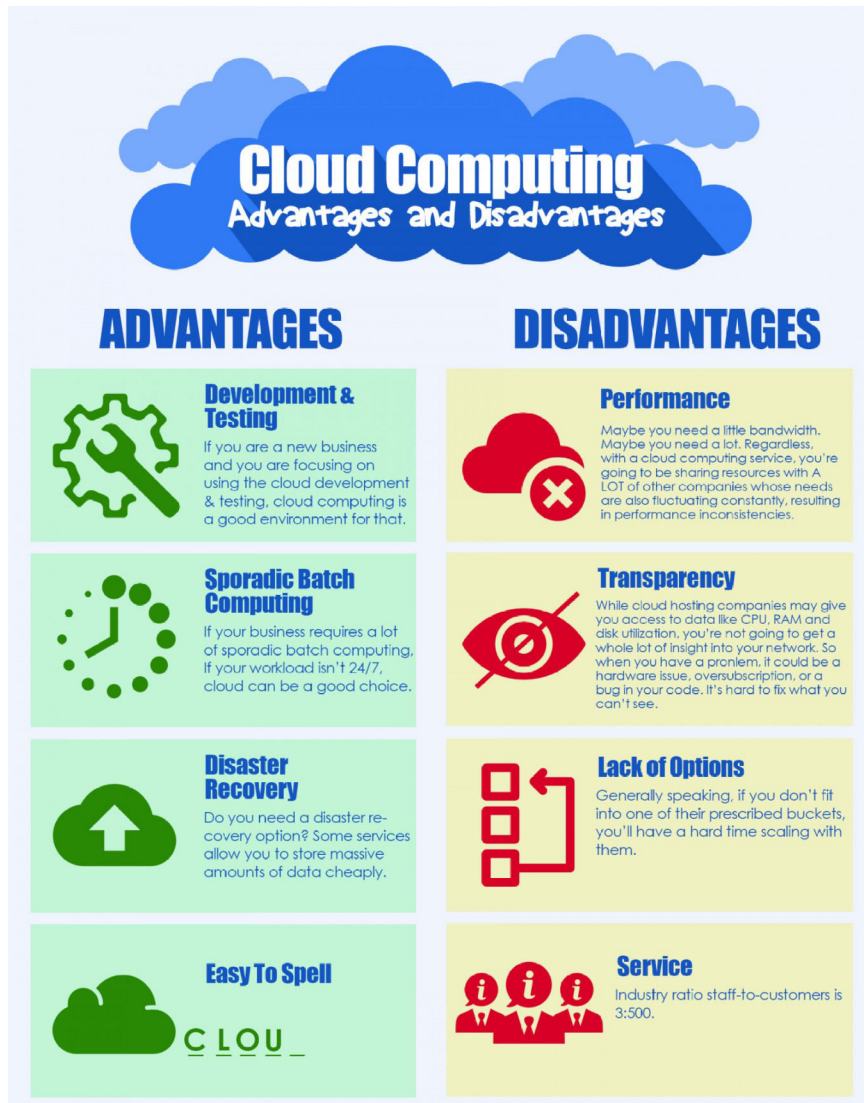
Paket yönetim sistemi APT'dir. APT yazılım paketlerinin alınma, yapılandırma ve kurulum sürecini otomatikleştirir.

## Cloud Computing(Bulut Bilişim)

Uygulamaların sanal olarak kullanılıp çalıştırılmasıdır. Programlara kurulum yapılmadan İnternet üzerinden erişilebilmektedir. Bulut bilişim sistemleri genellikle REST tabanlı API'leri kullanır.

Google Docs, Microsoft Office 365, Zoom ve AWS Lambda bulut sistemlerine örnektir.

Client cihazların uzak sunuculardan, veritabanlarından ve bilgisayarlardan internet üzerinden verilere erişmesine izin vererek çalışır.



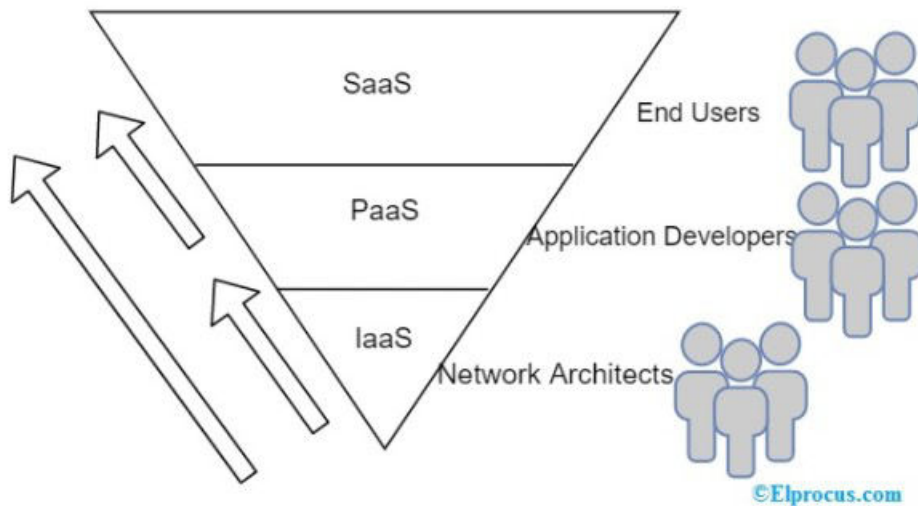
## Cloud Computing Tipleri

**Özel Bulut(Public Cloud):** Bulut servisleri kullanıcıya özeldir. Kaynaklar diğer kullanıcılar ile paylaşılmaz. Sunulan servislerin ölçeklenebilirliğini ve uyarlanabilirliğini kullanıcı kontrol eder. IaaS ve PaaS modeli ile sağlanabilir.

**Genel Bulut(Public Cloud):** Herkese açıktır. Yönetim ve bakım yapmak zorunluluğu yoktur.

### Genel Bulut Servis Modelleri

1. *Infrastructure as a Service:* Uygulamalar için bilgisayarlara, ağlara ve depolamaya erişim sağlar ama sistem ve ağı kurmak uygulamayı yapanların görevidir. AWS ve Microsoft Azure IaaS'a örnektir. Kullanıcı CPU, RAM ve depolama miktarını belirler ve servis sağlayıcıdan alır. Servis konfigürasyonu kullanıcıya aittir.
2. *Platform as a Service:* IaaS'a ek olarak veritabanı yönetim sistemi de sağlar. SaaS'dan daha düşük seviyede çalışır. Donanım ve altyapı bakımı PaaS tarafından karşılanır.
3. *Software as a Service:* Yazılım bulutta bulunur ve internet üzerinden erişilebilirdir. Web tabanlı posta ve müzik uygulamaları SaaS'a örnektir.

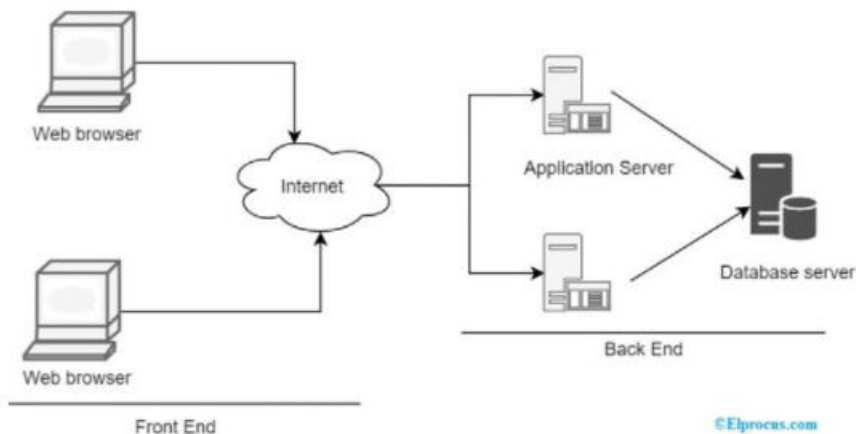


**Melez Bulut(Hybrid Cloud):** Private ve Public bulutun kombinasyonudur.

### Cloud Computing Mimarisi

Ön taraf ve arka taraf olarak iki bileşenden oluştuğu düşünülebilir. Ön taraf kullanıcıyı arka taraf da sistemin bulut kısmını ifade eder.

Ön taraf clientın bilgisayarı ya da bilgisayar ağı ve sisteme erişim için gerekli olan uygulamadır. Arka taraf ise verilen hizmet için olan bilgisayarlar, sunucular ve veri depolama sistemleridir.



## FaaS(Function as a Service)

CPU, Memory, I/O kaynaklarının Cloud tarafından sağlanması ve bize sadece kodu yazmanın kalması demektir. Kullanıcıyı karmaşık alt yapı sorunlarından uzaklaştırır.

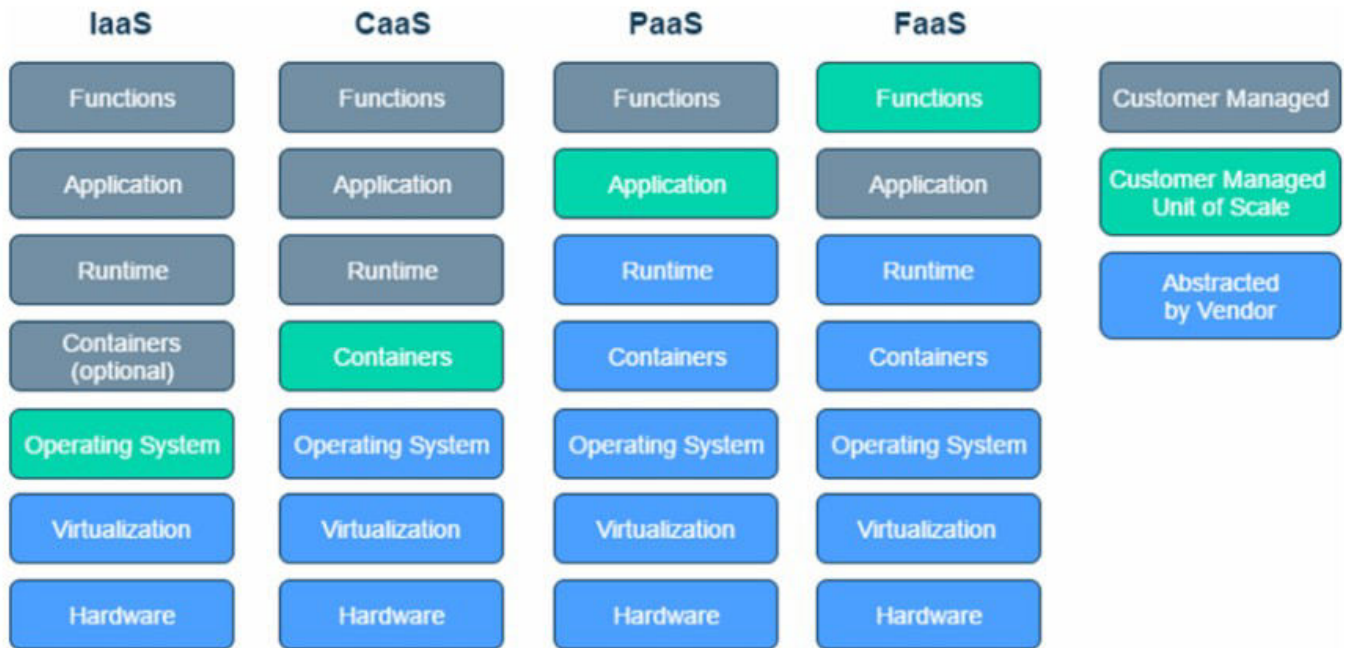
FaaS ile fiziksel donanım, sanal makine işletim sistemi ve web sunucusu yazılım yönetimi Cloud hizmeti sağlayıcısı tarafından gerçekleştirilir. Bunun sayesinde uygulamada sadece koda odaklanma sağlanır.

FaaS mikroservisler şeklinde bir uygulama geliştirme ortamı sağlar.

FaaS uygulamaları buluta dağıtma konusunda kolaylık sağlar. Sunucusuz olma özelliği ile bulut platformuna fonksiyon yüklenir. Sunuculara bakım yapmak geliştiricinin görevi değildir.

FaaS ile uygulamanın tamamını bir veya daha fazla sunucuya dağıtmak yerine fonksiyonlarını yüklemektir.

FaaS Kubernetes ve Knative ile kullanılabilir.

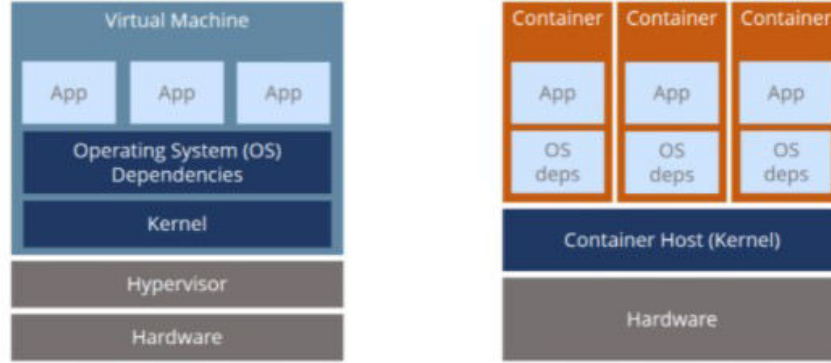


Serverless olarak da tanımlanmaktadır. Serverless uygulamaları server'a gerek kalmadan geliştirebilmek ve çalıştırabilmektir.

## Container

Aynı işletim sistemi üzerinde birbirlerinden izole edilmiş ortamlara denilir. Containerler işletim sistemi bağımsız çalışır. 2008 yılından beri Linux sistemlerinde(LXC) bulunmaktadır ancak Docker ile popülerleşmiştir.

## Virtual Machine vs. Container



Uygulama kurulumu yaparken sanal makine kullanılırsa kaynak kullanımı çok olacaktır ve bu işlemleri yavaşlatır. Başka bir makineye taşınmak istenirse uyumsuzluklar meydana gelebilir. Ama container kullanılırsa kaynak tüketimi daha verimli ve makinelerden bağımsız çalışma imkanı verir.

### Docker

Kendi engine'i sayesinde sanallaştırma yapan bir platformdur. Containerlar kullanarak uygulamaları daha kolay dağıtılabılır ve çalıştırılabilir hale getirir.

Docker client-server mimarisi kullanır. Docker da client CLI(commend line interface), server ise dockerdır.

### Docker komutları

**kullanım:** docker [seçenekler] komut

#### seçenekler:

--config string: client config dosyalarının konumu  
-c, --context string: daemon'a bağlanmak için kullanılacak içerik adı  
-D, --debug: debug modu etkinleştirme  
-H, --host list: daemon soketlerine bağlanma  
-l, --log-level string: loglama seviyesi kurma, seviyeler: debug, info, warn, error, fatal .default olarak infodur.  
tls komutları: --tls, --tlscacert string, --tlscert string, --tlskey string, --tlsverify  
-V, --version: versiyon bilgilerini verme

#### komutlar:

attach: Çalışan containera yerel standart input, output ve hata streamleri ekler.  
build: Dockerfile'dan image oluşturur.  
commit: Container değişimlerinden yeni bir image oluşturur.  
cp: Container ve yerel dosyayı kopyalar.  
create: Yeni container oluşturur.  
diff: Container dosyalarındaki yapılan değişiklikleri gösterir.  
events: Sunucudan gerçek zamanlı olaylar alınır.  
exec: Çalışan containerda komut çalıştırır.  
export: Containerı tar dosyası olarak dışarı aktarır.  
history: Image'in geçmişini gösterir.

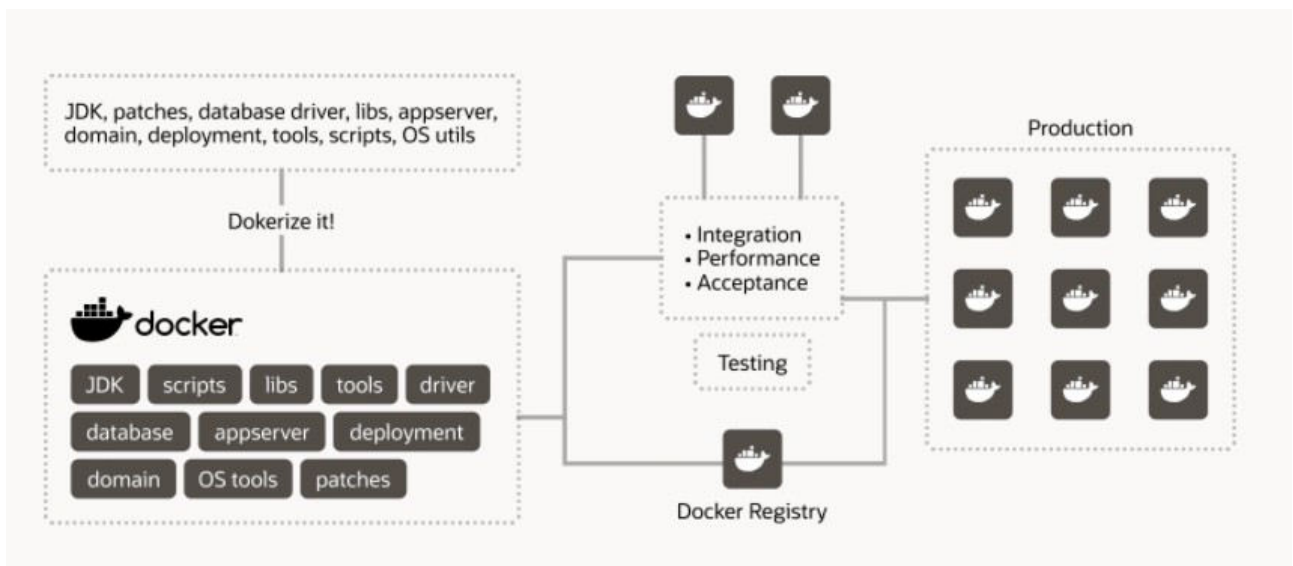
*images*: Imagerleri listeler.  
*import*: Image oluşturmak için içeriği tarballdan içeri aktarır.  
*info*: Docker Daemon ile ilgili bilgi verir.  
*inspect*: Docker nesnelerinin bilgisini verir.(low level)  
*kill*: Containerı sonlandırır.  
*load*: Tar arşivinden ya da STDIN'den image yükler.  
*login*: Docker registry'de oturum açar.  
*logout*: Docker registry'den çıkış yapar.  
*logs*: Container kayıtlarını gösterir.  
*pause*: Container içindeki işlemleri duraklatır.  
*port*: Maplenen portlarını gösterir.  
*ps*: Containerları listeler.  
*pull*: Registryden image ya da repo çeker.  
*push*: Registry'e image ya da repo gönderir.  
*rename*: Containerı yeniden adlandırır.  
*restart*: Containerı yeniden başlatır.  
*rm*: Container'ı kaldırır. Bir veya daha fazla olabilir.  
*rmi*: Image'i kaldırır. Bir veya daha fazla olabilir.  
*run*: Yeni container'ı çalıştırır.  
*save*: Image'i tar arşivine kaydeder. Bir veya daha fazla olabilir.  
*search*: Docker Hub'dan image arar.  
*start*: Durmuş olan container'ı çalıştırır.  
*stats*: Kaynak kullanım istatistiklerini gösterir.  
*stop*: Containerı durdurur.  
*tag*: Image'e etiket verir.  
*top*: Containerin çalışan işlemlerini gösterir.  
*unpause*: Duraklatmayı kaldırır.  
*update*: Container konfigürasyonunu günceller.  
*version*: Docker versiyon bilgisi verir.  
*wait*: Container durana kadar engellenmesini ve sonradan çıkış kodlarını yazmayı sağlar.

**Docker Daemon(Docker Engine)**: Hypervisor'ün dockerdaki karşılığıdır. CPU ve RAM gibi işletim sistemine ait işleri Docker Daemon yapar.  
Hypervisor sanal makineleri oluşturan ve çalıştıran yazılımdır.

**Docker Compose**: Birden fazla containerı tek dosyada kullanmayı sağlar.

### Docker Registry

Depolanan imagerlerin erişim kontrolü ve güvenliğini iyileştirmeye yardımcı olur. Image dağıtımını yönetir.





## Örnekler:

**docker search:** docker hub'daki containerleri getirir.

```
elif@ubuntu:~$ sudo docker search wordpress
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
wordpress	The WordPress rich content management system...	4310	[OK]	
appcontainers/wordpress	Centos/Debian Based Customizable Wordpress C...	34		[OK]
aveltens/wordpress-backup	Easily backup and restore your WordPress blo...	23		[OK]
conetix/wordpress-with-wp-cli	WordPress with wp-cli integration	17		[OK]
arm32v7/wordpress	The WordPress rich content management system...	15		
centurylink/wordpress	Wordpress image with MySQL removed.	14		[OK]
appsvcorp/wordpress-alpine-php	This is a WordPress Docker image which can ...	12		
wordpressdevelop/php	PHP images for the WordPress local developme...	9		
wordpressdevelop/cli	WP-CLI images for the WordPress local develo...	7		
wodby/wordpress-php	PHP for WordPress	6		[OK]
wodby/wordpress-nginx	Nginx for WordPress	6		[OK]
dalareo/wordpress-ldap	Wordpress images with LDAP support automatic...	6		[OK]
arm64v8/wordpress	The WordPress rich content management system...	5		
wordpressdevelop/phpunit	PHPUnit images for the WordPress local devel...	4		
ansibleplaybookbundle/wordpress-ha-apb	An APB which deploys Wordpress HA	4		[OK]
gwee/wordpress-sendmail-docker	wordpress with sendmail	4		[OK]
s390x/wordpress	The WordPress rich content management system...	3		
wodby/wordpress	Vanilla WordPress container image	2		
demyx/wordpress	Non-root Docker image running Alpine Linux, ...	2		
dsteinkopf/wordpress	wordpress clone plus some php extensions	1		[OK]
ppc64le/wordpress	The WordPress rich content management system...	1		
bakudankun/wordpress-ja	WordPress Japanese Edition forked from offic...	1		[OK]
i386/wordpress	The WordPress rich content management system...	1		
amd64/wordpress	The WordPress rich content management system...	0		
arm32v5/wordpress	The WordPress rich content management system...	0		

**docker pull <name>:** image'i kendi ortamımıza indirmeyi sağlar.

```
root@ubuntu:/home/elif# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
e1acdbe380c: Pull complete
bed879327370: Pull complete
03285f80bafd: Pull complete
ccc17412a00a: Pull complete
1f556ecc09d1: Pull complete
adc5528e468d: Pull complete
1afc286d5d53: Pull complete
6c724a59adff: Pull complete
0f2345f8b0a3: Pull complete
c8461a25b23b: Pull complete
3adb49279bed: Pull complete
77f22cd6c363: Pull complete
Digest: sha256:d45561a65aba6edac77be36e0a53f0c1fba67b951cb728348522b671ad63f926
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

**docker run:** Image localde yoksa indirir, sonrasında imageden containir oluşturur ve başlatır. Her run komutu çalıştırıldığında yeni bir container oluşturulur.

```
root@ubuntu:/home/elif# docker run -d --name docker-sql -e MYSQL_ROOT_PASSWORD=0000 mysql
71b4ff158df97f1da730402e16b3f57c64b9aa088e78fb86ce6f3091d5bf786b
```

`-d --name docker-sql -e MYSQL_ROOT_PASSWORD=0000 mysql`

`-d` : detach mode -> arka planda çalışması

`--name`: containeri isimlendirmek

`-e` : environment variable

**docker ps:** çalışan containerlar listelenir.

```
root@ubuntu:/home/elif# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c9393d93aa84	wordpress	"docker-entrypoint.s..."	37 seconds ago	Up 35 seconds	0.0.0.0:8080->80/tcp, :::8080->80/tcp	sharp_bell
71b4ff158df9	mysql	"docker-entrypoint.s..."	6 minutes ago	Up 6 minutes	3306/tcp, 33060/tcp	docker-sql

**docker ps -a:** daha önce çalışmış ve durmuş olan containerları da gösterir

## Alpine indirip kullanmak

Alpine linux tabanlı işletim sistemidir.

```
sudo docker pull alpine
```

```
elif@worker01:~$ sudo docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
29291e31a76a: Pull complete
Digest: sha256:eb3e4e175ba6d212ba1d6e04fc0782916c08e1c9d7b45892e9796141b1d379ae
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
elif@worker01:~$
```

```
sudo docker run alpine ls -l
```

```
elif@worker01:~$ sudo docker run alpine ls -l
total 56
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 bin
drwxr-xr-x  5 root    root      340 Aug 26 17:11 dev
drwxr-xr-x  1 root    root      4096 Aug 26 17:11 etc
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 home
drwxr-xr-x  7 root    root      4096 Aug  5 12:25 lib
drwxr-xr-x  5 root    root      4096 Aug  5 12:25 media
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 mnt
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 opt
dr-xr-xr-x 375 root    root        0 Aug 26 17:11 proc
drwx----- 2 root    root      4096 Aug  5 12:25 root
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 run
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 sbin
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 srv
dr-xr-xr-x 13 root    root        0 Aug 26 17:11 sys
drwxrwxrwt  2 root    root      4096 Aug  5 12:25 tmp
drwxr-xr-x  7 root    root      4096 Aug  5 12:25 usr
drwxr-xr-x 12 root    root      4096 Aug  5 12:25 var
```

Alpine üzerinden terminalde echo komutunu kullanmak.

```
sudo docker run alpine echo "alpine in docker"
```

```
elif@worker01:~$ sudo docker run alpine echo "alpine in docker"
alpine in docker
elif@worker01:~$
```

```
sudo docker run -it alpine /bin/sh / # ls -l
```

```
elif@worker01:~$ sudo docker run -it alpine /bin/sh
/ # ls -l
total 56
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 bin
drwxr-xr-x  5 root    root      360 Aug 26 17:16 dev
drwxr-xr-x  1 root    root      4096 Aug 26 17:16 etc
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 home
drwxr-xr-x  7 root    root      4096 Aug  5 12:25 lib
drwxr-xr-x  5 root    root      4096 Aug  5 12:25 media
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 mnt
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 opt
dr-xr-xr-x 374 root    root        0 Aug 26 17:16 proc
drwx----- 1 root    root      4096 Aug 26 17:16 root
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 run
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 sbin
drwxr-xr-x  2 root    root      4096 Aug  5 12:25 srv
dr-xr-xr-x 13 root    root        0 Aug 26 17:16 sys
drwxrwxrwt  2 root    root      4096 Aug  5 12:25 tmp
drwxr-xr-x  7 root    root      4096 Aug  5 12:25 usr
drwxr-xr-x 12 root    root      4096 Aug  5 12:25 var
/ # uname -a
Linux cd6012bb3bb1 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed Aug 11 15:58:17 UTC 2021 x86_64
Linux
/ #
```

## Kubernetes(cluster management system)

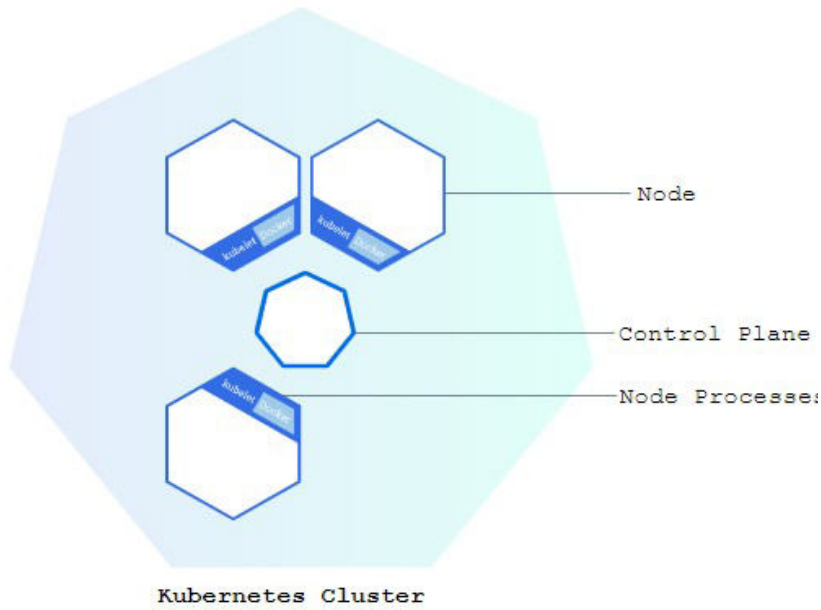
Uygulamaların dağıtım, ölçeklendirme, çalıştırma gibi yönetim süreçlerini otomatikleştiren bir uygulamadır.

Kubernetes containerları production ortamında orkestrasyon yapabilmeyi sağlar. Orkestrasyon, bilgisayar sistemleri ve yazılımlarının otomasyon yapılandırması, koordinasyonu ve yönetimidir.

### Nasıl çalışır?

Kubernetes clusterlardan oluşur ve bu clusterların için de temel yönetim ve kontrol için master node'u ve birden çok worker node bulunur.

Cluster aynı amaç üzerinden toplanmış bir ya da birden fazla bilgisayarın oluşturduğu ortamdır. Node clusterdaki bilgisayardır. Fiziksel olma zorunluluğu yoktur.



Uygulama kubernetes clusterine gönderilmesi için yapılması gerekenler; Önce geliştirilen uygulama containera dönüştürülmelidir. Dönüştürme aşamasından sonra YAML ya da JSON dosyası olarak deployment objesi yaratılır. Deployment, rest objeleridir ve bir kere hazırlanıp birden fazla kez deploy edilebilir. Api-server üzeinden oluşturulur. Son olarak da YAML ya da JSON dosyası kubectl ile master node'da bulunan api-server'a gönderilir. Api-server ile worker nodeların da çalışması sağlanır.

### YAML bileşenleri

**kind:** Kaynak tipini tanımlar. Pod, deployment, service kaynak tiplerine örnektir.

**Pod:** En küçük ve en basit kubernetes nesnesidir. Clusterda çalışan containerları temsil eder. Pod dockerdaki containerlara benzer. 1 pod 1 node üzerinde çalışabilir.

**Deployment:** rest objeleridir. Yaml ya da json dosyaları içinde tanımlanabilir.

**Service:** Load balancing görevini yerine getirir. Reverse proxy kullanılır ve default olarak TCP protokolü üzerinde çalışır. Servisler arası kesinti yaşanmamasını sağlar.

**image:** Kullanılacak image belirtilir.

**label:** tanımlama için kullanılır.



Type: kullanılacak bileşeni belirtmek için kullanılır.

Örneğin load-balancer ile external api kullanılarak dışardan erişim sağlanır.

## Helm

Kubernetes üzerinde uygulamaları yönetmeyi sağlar. Uygulama deploy etme, upgrade etme ve sürümleri kontrol etme işlemleri yapılabilir.

## Kubernetes interaktif dökümanından örnek:

### *-kubernetes clusteri oluşturmak*

minikube başlatılır. minikube test ve geliştirme yapmak için kullanılan kubernetes kümesidir.

```
minikube start --wait=false
```

```
* Reconfiguring existing host ...
* Using the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.4 LTS
* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Launching Kubernetes ...
* Enabling addons: default-storageclass, storage-provisioner
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
```

### *-cluster hakkında bilgi almak*

```
kubectl cluster-info
```

```
$ kubectl cluster-info
Kubernetes master is running at https://172.17.0.94:8443
KubeDNS is running at https://172.17.0.94:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

### *-mevcut nodeları ve durumlarını öğrenmek*

```
kubectl get nodes
```

```
$ kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready     master   4m8s  v1.17.3
```

### ***-containerı deploy etmek***

```
kubectl create deployment minikube-deployment --image=katacoda/docker-http-server
```

```
$ kubectl create deployment minikube-deployment --image=katacoda/docker-http-server
deployment.apps/minikube-deployment created
```

### ***-podları ve durumlarını öğrenmek***

```
kubectl get pods
```

```
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
first-deployment-666c48b44-9z8l4    1/1     Running   0           90s
minikube-deployment-8464845dfb-zkgnb 1/1     Running   0           65s
```

### ***-dinamik bağlantı sağlamak***

```
kubectl expose deployment first-deployment --port=80 --type=NodePort
```

```
$ kubectl expose deployment minikube-deployment --port=80 --type=NodePort
service/minikube-deployment exposed
```

### ***-bağlantı noktasını bulup, HTTP isteği göndermek***

```
export PORT=$(kubectl get svc minikube-deployment -o go-
template='{{range.spec.ports}}{{if .nodePort}}{{.nodePort}}{{"\n"}}{{end}}{{end}}')
echo "Accessing host01:$PORT"
curl host01:$PORT
```

```
$ export PORT=$(kubectl get svc minikube-deployment -o go-template='{{range.spec.ports}}{{if .nodePort}}{{.nodePort}}{{"\n"}}{{end}}{{end}}')
$ echo "Accessing host01:$PORT"
Accessing host01:31604
$ curl host01:$PORT
<h1>This request was processed by host: minikube-deployment-8464845dfb-zkgnb</h1>
```

### ***-dashboardu kullanmak***

```
minikube addons enable dashboard
kubectl apply -f /opt/kubernetes-dashboard.yaml
kubectl get pods -n kubernetes-dashboard -w
```

```
$ kubectl apply -f /opt/kubernetes-dashboard.yaml
namespace/kubernetes-dashboard configured
service/kubernetes-dashboard-katacoda created
$ kubectl get pods -n kubernetes-dashboard -w
NAME                                READY   STATUS    RESTARTS   AGE
dashboard-metrics-scraper-7b64584c5c-9t8zh 1/1     Running   0           2m14s
kubernetes-dashboard-79d9cd965-bghvp      1/1     Running   0           2m13s
```

Deployments						
Name	Namespace	Labels	Pods	Age	Images	
minikube-deployment2	default	app: minikube-deployment2	1 / 1	11.minutes	katacoda/docker-http-server	
minikube-deployment	default	app: minikube-deployment	1 / 1	13.minutes	katacoda/docker-http-server	
first-deployment	default	app: first-deployment	1 / 1	14.minutes	katacoda/docker-http-server	

Pods								
Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Age
minikube-deployment2-6f944f54fd-7i9v	default	app: minikube-deployment2 pod-template-hash: 6f944f54fd	minikube	Running	0	-	-	11.minutes
minikube-deployment-8464845dfb-zkgnb	default	app: minikube-deployment pod-template-hash: 8464845dfb	minikube	Running	0	-	-	13.minutes
first-deployment-666c48b44-9z8l4	default	app: first-deployment pod-template-hash: 666c48b44	minikube	Running	0	-	-	14.minutes

## Ubuntu kubernetes kurulumu

### -kubectl kurulumu

Kubectl kubernetes ortamına bağlanmak ve yönetmek için kullanılır.

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
elif@ubuntu:~$ curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://
/storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 44.7M 100 44.7M 0 0 2442k 0 0:00:18 0:00:18 --:--:-- 2538k
```

```
elif@ubuntu:~$ sudo apt-get install kubectl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
kubectl
0 upgraded, 1 newly installed, 0 to remove and 277 not upgraded.
Need to get 9,036 kB of archives.
After this operation, 46.9 MB of additional disk space will be used.
Get:1 https://packages.cloud.google.com/apt cloud-sdk/main amd64 kubectl amd64 1.22.1-00 [9,036 kB]
Fetched 9,036 kB in 9s (1,017 kB/s)
Selecting previously unselected package kubectl.
(Reading database ... 216403 files and directories currently installed.)
Preparing to unpack ../kubectl_1.22.1-00_amd64.deb ...
Unpacking kubectl (1.22.1-00) ...
Setting up kubectl (1.22.1-00) ...
```

```
elif@ubuntu:~$ chmod +x ./kubectl
elif@ubuntu:~$ sudo mv ./kubectl /usr/local/bin/kubectl
elif@ubuntu:~$
```

## **-docker'ı başlatmak ve etkinleştirmek**

sudo systemctl enable docker

```
elif@ubuntu:~$ sudo systemctl enable docker
elif@ubuntu:~$
```

sudo systemctl status docker

```
elif@ubuntu:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-08-25 23:37:58 PDT; 1min 10s ago
 TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 52599 (dockerd)
     Tasks: 10
    Memory: 41.8M
    CGroup: /system.slice/docker.service
            └─52599 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.128823505-07:00" level=warning msg>
Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.129094792-07:00" level=warning msg>
Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.129307984-07:00" level=warning msg>
Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.129560346-07:00" level=info msg="L>
Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.342946489-07:00" level=info msg="D>
Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.495328994-07:00" level=info msg="L>
Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.543604164-07:00" level=info msg="D>
Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.543713004-07:00" level=info msg="D>
Aug 25 23:37:58 ubuntu systemd[1]: Started Docker Application Container Engine.
Aug 25 23:37:58 ubuntu dockerd[52599]: time="2021-08-25T23:37:58.587203866-07:00" level=info msg="A>
lines 1-21/21 (END)
```

docker çalışmıyorsa:

sudo systemctl start docker

## **-Signing Key eklemek**

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add

```
elif@ubuntu:~$ curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key --keyring /
usr/share/keyrings/cloud.google.gpg add -
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload    Total   Spent    Left   Speed
100 2537 100 2537    0     0  4554      0 --:--:-- --:--:-- --:--:-- 4546
OK
```

## **-Dahil olmayan repoları eklemek**

sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

```
elif@ubuntu:~$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 https://packages.cloud.google.com/apt cloud-sdk InRelease [6,739 B]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:3 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [9,383 B]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [49.4 kB]
Fetched 394 kB in 1s (275 kB/s)
Reading package lists... Done
```

**Not: Bu işlemler her server node'u için tekrarlanır.**

**- kubernetes kurulum toollarını indirmek**

**kubeadm(kubernetes admin):** clusterı başlatmaya yardımcı olan araçtır.

```
sudo apt-get install kubeadm kubelet kubectl
```

```
elif@ubuntu:~$ sudo apt-get install kubeadm kubelet kubectl
Reading package lists... Done
Building dependency tree
Reading state information... Done
kubectl is already the newest version (1.22.1-00).
The following additional packages will be installed:
  conntrack cri-tools ebtables ethtool kubernetes-cni socat
Suggested packages:
  nftables
The following NEW packages will be installed:
  conntrack cri-tools ebtables ethtool kubeadm kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 277 not upgraded.
Need to get 64.8 MB of archives.
After this operation, 300 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 conntrack amd64 1:1.4.5-2 [30.3 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal/main amd64 ebtables amd64 2.0.11-3build1 [80.3 kB]
Get:1 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.13.0-01 [8,775 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal/main amd64 ethtool amd64 1:5.4-1 [134 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal/main amd64 socat amd64 1.7.3.3-2 [323 kB]
```

```
sudo apt-mark hold kubeadm kubelet kubectl
```

```
elif@ubuntu:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
```

**-kurulumu onaylamak**

```
kubeadm version
```

```
elif@ubuntu:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.1", GitCommit:"632ed300f2c34f6d6d15ca4cef3d3c7073412212", GitTreeState:"clean", BuildDate:"2021-08-19T15:44:22Z", GoVersion:"go1.16.7", Compiler:"gc", Platform:"linux/amd64"}
```

**Not: Bu işlemler her server node'u için tekrarlanır.**



### *-Serverlara hostname vermek*

#### **master node:**

```
sudo swapoff -a  
sudo hostnamectl set-hostname master-node
```

```
elif@ubuntu:~$ sudo swapoff -a  
elif@ubuntu:~$ sudo hostnamectl set-hostname master-node
```

#### **worker node:**

```
sudo swapoff -a  
sudo hostnamectl set-hostname master-node
```

```
elif@ubuntu:~$ sudo swapoff -a  
elif@ubuntu:~$ sudo hostnamectl set-hostname worker01
```

### *-master node üzerinde kubernetes'i başlatmak*

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --v=5 --ignore-preflight-errors=all
```

```
elif@master-node:~/etc/kubernetes/manifests$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16  
You should now deploy a pod network to the cluster.  
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:  
https://kubernetes.io/docs/concepts/cluster-administration/addons/  
  
Then you can join any number of worker nodes by running the following on each as root:  
  
kubeadm join 192.168.222.132:6443 --token y1b0pv.rnz19b31nm5lxu6k \
```

```
--discovery-token-ca-cert-hash sha256:4e81c27db0adcdcc9e4b562fbe2ccca3264f6838a3c9d6c29e21764b55da6a3b  
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

```
elif@master-node:~$ mkdir -p $HOME/.kube  
elif@master-node:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
kubernetes-master:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
elif@master-node:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

### *-node durumunu değiştirmek*

```
elif@master-node:~$ kubectl get nodes  
NAME           STATUS    ROLES          AGE   VERSION  
master-node    NotReady  control-plane, 30m   v1.22.1
```

```
sudo kubectl apply -f kube-flannel.yml --validate=false
```

kube-flannel.yml: <https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml>

```
elif@master-node:~/Desktop$ kubectl apply -f kube-flannel.yml --validate=false
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/psp.flannel.unprivileged created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel unchanged
configmap/kube-flannel-cfg configured
daemonset.apps/kube-flannel-ds created
elif@master-node:~/Desktop$ kubectl get nodes
NAME          STATUS    ROLES          AGE   VERSION
master-node   NotReady  control-plane, 114m  v1.22.1
elif@master-node:~/Desktop$ kubectl get nodes
NAME          STATUS    ROLES          AGE   VERSION
master-node   Ready     control-plane, 115m  v1.22.1
```

connection refused çözümü:

```
sudo -i
```

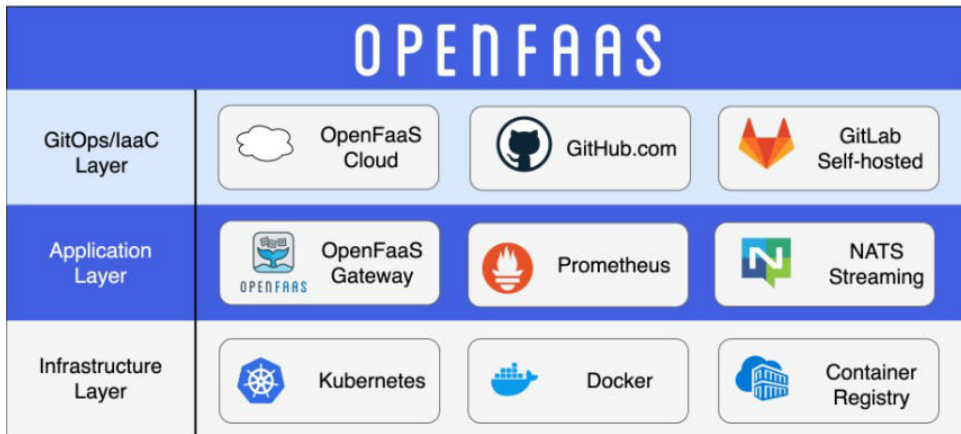
```
swapoff -a
```

```
exit
```

```
strace -eopenat kubectl version
```

## OPENFAAS

Açık kaynaklı FaaS platformudur. MIT lisansı altında dağıtılır. Docker ve Kubernetes ile kullanılabilir.



### OPENFAAS Mimarisi

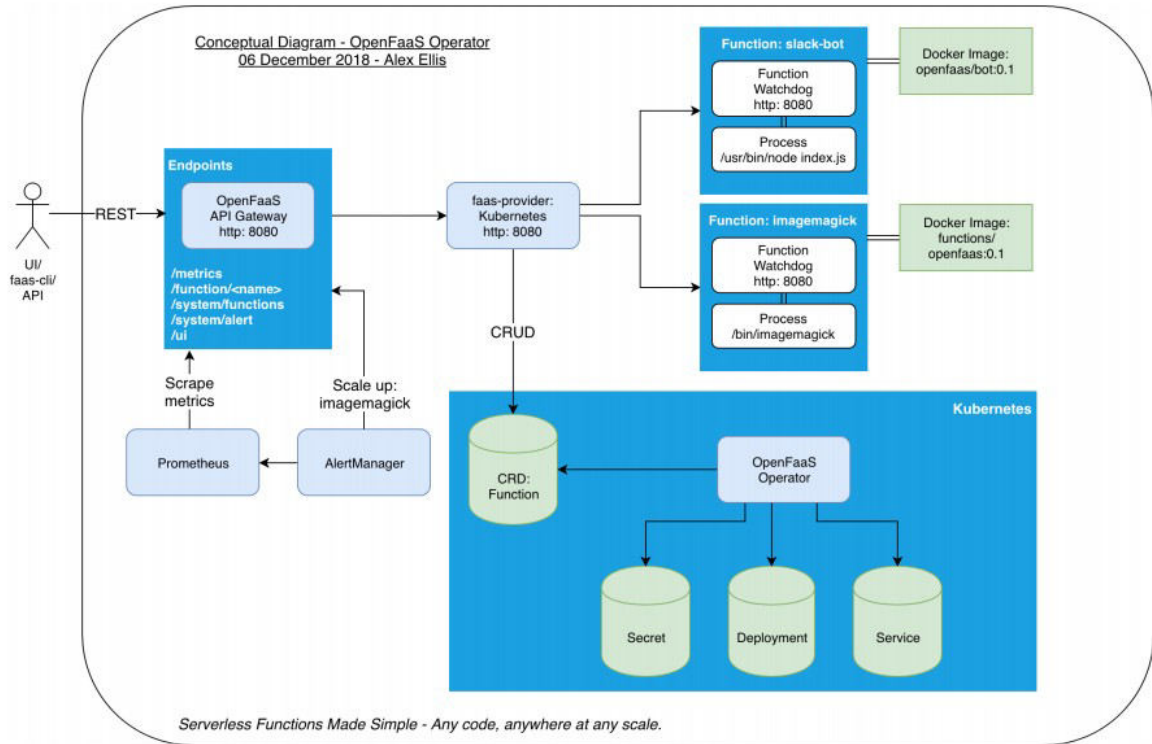
OpenFaaS'ın API Gateway'i faas-provider'a dışarıdan erişim verir. Faas-provider docker imajlarının oluşturulması, fonksiyonların deploy edilmesi ve sistemden kaldırılması işlemlerini yapar. Gateway bir REST API olarak gerçekleştirilir ve faas-cli UI ve http isteklerini sağlar. Gateway'in diğer görevi kullanıcının belirttiğine göre otomatik ölçeklendirme sağlamaktır. Fonksiyonların ölçeklendirme gibi konfigürasyon dosyaları her fonksiyonun yaml dosyasında ayarlanır.

OpenFaaS Kubernetes Cluster'ında kuruluysa Horizontal Pod Autoscaler kullanılır.

*Horizontal Pod Autoscaler:* CPU kullanımı ya da belirlenen bazı metriklere göre pod sayısını

otomatik olarak ölçeklendirir.

Prometheus, fonksiyonların metriklerini almayı sağlayan izleme uygulamasıdır. Bir metrik fonksiyonun çağırılma oranı olabilir.



## Kurulum

```
curl -sL https://cli.openfaas.com | sudo sh
```

```
elif@master-node:~/Desktop/faas-netes$ curl -sL https://cli.openfaas.com | sudo sh

Finding latest version from GitHub
0.13.13
Downloading package https://github.com/openfaas/faas-cli/releases/download/0.13.13/faas-cli as /tmp/faas-cli
Download complete.

Running with sufficient permissions to attempt to move faas-cli to /usr/local/bin
New version of faas-cli installed to /usr/local/bin
Creating alias 'faas' for 'faas-cli'.

OpenFaaS

CLI:
commit: 72816d486cf76c3089b915dfb0b66b85cf096634
version: 0.13.13
```

```
faas template pull
```

```
elif@master-node:~/Desktop/faas-netes/faas-cli$ faas template pull
Fetch templates from repository: https://github.com/openfaas/templates.git at
2021/08/26 02:01:09 Attempting to expand templates from https://github.com/openfaas/templates.git
2021/08/26 02:01:11 Fetched 13 template(s) : [csharp dockerfile go java11 java11-vert-x node node12 node14 php7 python python3 python3-debian ruby] from https://github.com/openfaas/templates.git
```

arkade openfaas için CLI tooludur.

```
curl -Ssf https://dl.get-arkade.dev/ | sudo sh
```

```
elif@master-node:~$ curl -Ssf https://dl.get-arkade.dev/ | sudo sh
x86_64
Downloading package https://github.com/alexellis/arkade/releases/download/0.7.29/
arkade as /tmp/arkade
Download complete.

Running with sufficient permissions to attempt to move arkade to /usr/local/bin
New version of arkade installed to /usr/local/bin
Creating alias 'ark' for 'arkade'.

  _ _ _ _ _
 / _ _ _ _ < _ _ _ _ \
( _ _ _ _ )
 \ _ _ _ _ / _ _ _ _ \

Get Kubernetes apps the easy way

Version: 0.7.29
Git Commit: def87cc54f6440729e8e737f74e74ecc19e57293
```

arkade get faas-cli

```
elif@master-node:~$ arkade get faas-cli
Downloading: faas-cli
https://github.com/openfaas/faas-cli/releases/download/0.13.13/faas-cli
8.04 MiB / 8.04 MiB [-----] 100.00%
Tool written to: /home/elif/.arkade/bin/faas-cli

# Add (faas-cli) to your PATH variable
export PATH=$PATH:$HOME/.arkade/bin/

# Test the binary:
/home/elif/.arkade/bin/faas-cli

# Or install with:
sudo mv /home/elif/.arkade/bin/faas-cli /usr/local/bin/
```

## Kaynakça:

- <https://www.ibm.com/cloud/learn/faas>
- <https://searchcloudcomputing.techtarget.com/definition/cloud-computing>
- <https://docs.docker.com/engine/install/ubuntu/>
- <https://www.endustri40.com/bulut-bilisim-cloud-computing-nedir/>
- <https://serverless.zone/abstracting-the-back-end-with-faas-e5e80e837362>
- <https://solus.com.tr/tr/container-konteyner-teknolojisi-nedir/>
- <https://www.ringcentral.co.uk/gb/en/blog/definitions/function-as-a-service-faas/>
- <https://www.investopedia.com/terms/c/cloud-computing.asp>
- <https://www.ibm.com/cloud/learn/cloud-computing-gbl>
- <https://www.ibm.com/tr-tr/cloud/learn/kubernetes>
- <https://martinfowler.com/articles/serverless.html#unpacking-faas>
- <https://martinfowler.com/bliki/Serverless.html>
- <https://martinfowler.com/articles/serverless.html>
- <https://www.sistas.com.tr/yeni-bir-teknoloji-kubernetes-nedir/>
- <https://www.erkanceran.com/cloud-sistemler-uzerinde-iaas-nedir-paas-nedir-caas-nedir-faas-nedir-saas-nedir-kaas-nedir/>
- [https://www.upnxtblog.com/index.php/2018/10/19/openfaas-tutorial-build-and-deploy-serverless-java-functions/#Step\\_1\\_Install\\_OpenFaaS](https://www.upnxtblog.com/index.php/2018/10/19/openfaas-tutorial-build-and-deploy-serverless-java-functions/#Step_1_Install_OpenFaaS)
- <https://kubernetes.io/docs/tasks/tools/install-kubectllinux/>
- <https://cloud.google.com/sdk/docs/install#deb>
- <https://www.katacoda.com/courses/serverless/openfaas>
- [https://www.upnxtblog.com/index.php/2018/10/19/openfaas-tutorial-build-and-deploy-serverless-java-functions/#Step\\_1\\_Install\\_OpenFaaS](https://www.upnxtblog.com/index.php/2018/10/19/openfaas-tutorial-build-and-deploy-serverless-java-functions/#Step_1_Install_OpenFaaS)
- <https://acloudguru.com/hands-on-labs/building-a-kubernetes-cluster-with-kubeadm>
- <https://www.vmwaremine.com/2021/02/11/no-matches-for-kind-daemonset-error-while-deploying-flannel-on-kubernetes/>
- <https://acloudguru.com/hands-on-labs/building-a-kubernetes-cluster-with-kubeadm>
- <https://discuss.kubernetes.io/t/the-connection-to-the-server-host-6443-was-refused-did-you-specify-the-right-host-or-port/552/5>
- <https://kubernetes.io/docs/reference/glossary/?fundamental=true>
- <https://kubernetes.io/docs/tasks/tools/install-kubectllinux/>
- <https://www.portainer.io/>
- <https://phoenixnap.com/kb/install-kubernetes-on-ubuntu>
- <https://stackoverflow.com/questions/52119985/kubeadm-init-shows-kubelet-isnt-running-or-healthy>
- <https://docs.docker.com/engine/reference/commandline/docker/>
- <https://www.oracle.com/tr/cloud-native/container-registry/what-is-docker/>
- <https://www.elprocus.com/cloud-computing-technology/>
- <https://gokhansengun.com/docker-nedir-nasil-calisir-nerede-kullanilir/>
- <https://medium.com/batech/docker-nedir-docker-kavramlar%C4%B1-avantajlar%C4%B1-901b37742ee0>