

# Kubernetes

# Co to są kontenery

Kontenery to nowy sposób na separację aplikacji od systemu operacyjnego oraz fizycznej infrastruktury wykorzystywanej do połączeń z siecią. Są one instalowane poza jądrem systemu operacyjnego i wirtualizują środowisko określonej aplikacji.



# Co to są kontenery

Duże firmy jak Google, Amazon wykorzystują kontenery od wielu lat,  
Ze względu na korzyści jakie niosą

- Szybki start aplikacji
- Łatwe w użyciu
- Łatwa skalowalność
- Powtarzalność w uruchamianiu aplikacji
- Duża wydajność

# Co to są kontenery

- Wszystkie usługi w Google uruchamiane są w kontenerach
- - Gmail, Web Search, Maps
- - Nawet wirtualne maszyny w GCE są uruchamiane z kontenerów



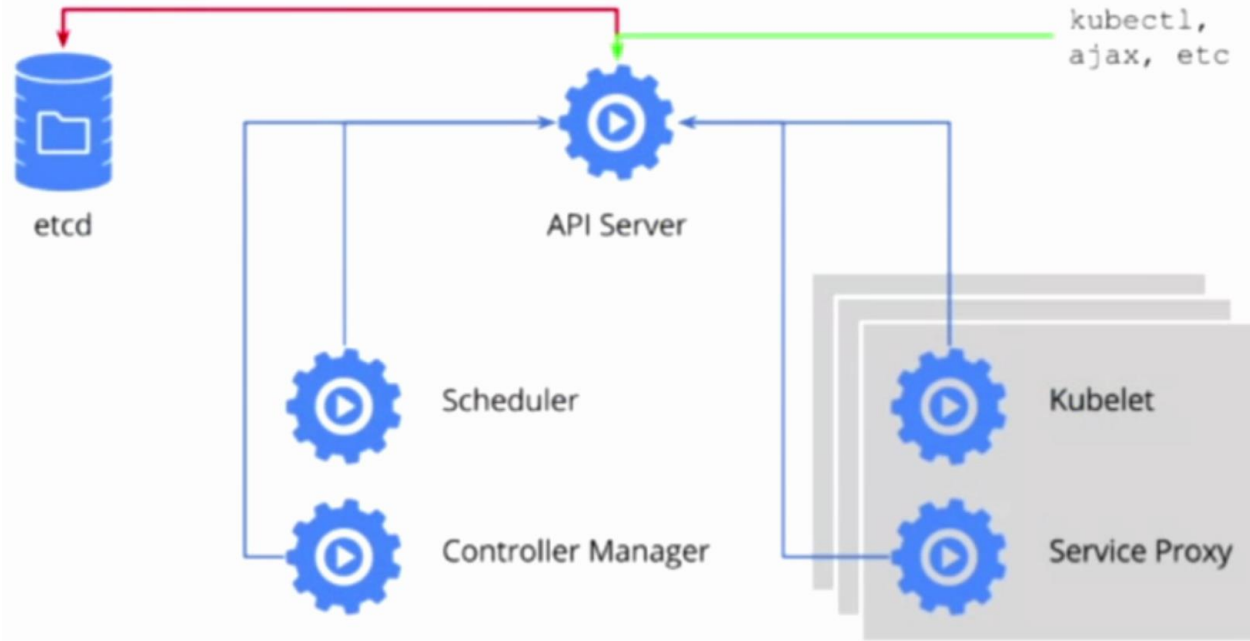
Tygodniowo uruchamianych jest 2 miliardy kontenerów

# Co to jest kubernetes

Jest to specjalistyczne narzędzie do zarządzania aplikacjami w kontenerach, rozmieszczonymi na wielu maszynach. Kubernetes potrafi planować nody, reorganizować klastry podczas dużego obciążenia lub awarii któryś nodów (samodzielne naprawianie się).

- Platforma do hostowania Dockerowych kontenerów
- Dostarcza takie rozwiązania jak: grupowanie kontenerów, load balancing, auto-naprawa, skalowanie
- Może współpracować z wieloma platformami chmurowymi lub być instalowany na fizycznej maszynie
- Umożliwia bardzo szybkie uruchamianie i update aplikacji

## Kubernetes Architecture



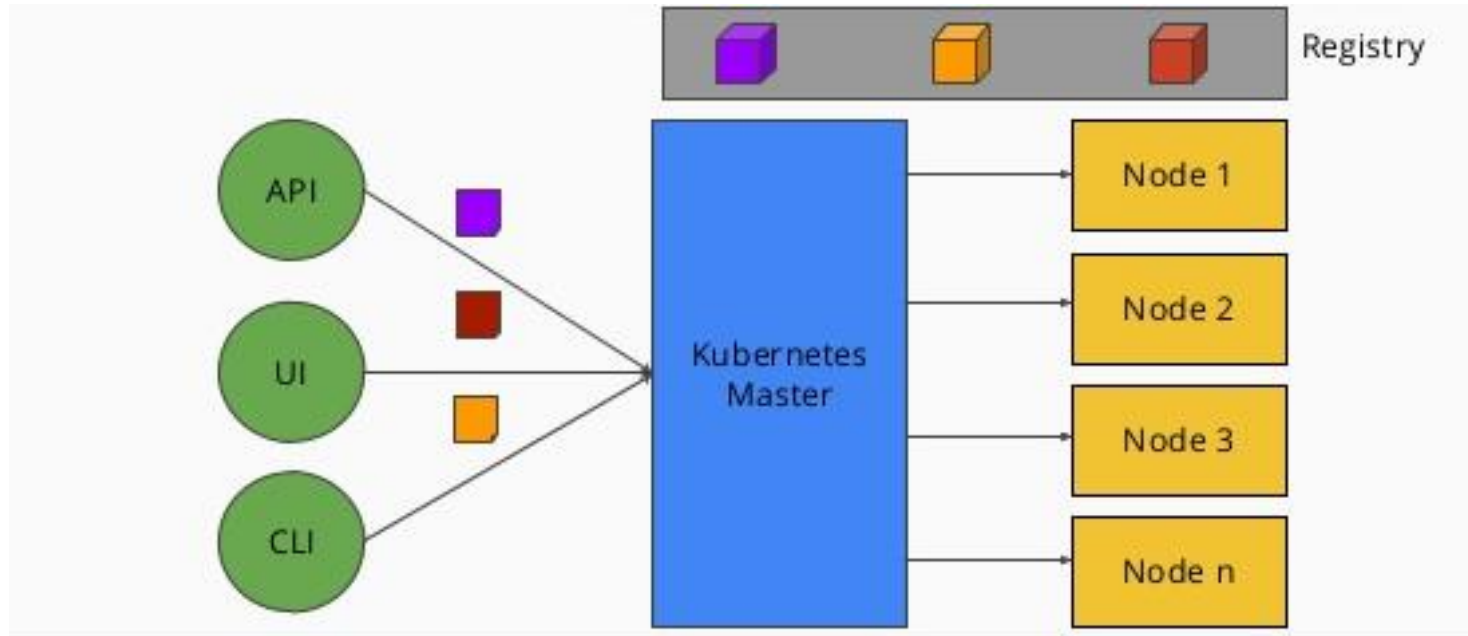
To grupa nodów, które mogą być fizycznymi serwerami lub wirtualnymi maszynami na których zainstalowany jest kubernetes.

Jednostkę zarządzającą poszczególnymi nodami nazywamy masterem.

Jest to główny punkt zarządzający oraz wyjście na świat poprzez wystawiony RESTowy endpoint.



# Klaster



Pod – grupa kontenerów

Labels – klucz/wartość opisująca pewną właściwość Poda

Kubelet – zarządca kontenerów


Proxy – load balancer dla podów

Replication Controller – zarządza replikacją podów

Scheduler – planuje pracę podów

API server – centralny punkt kontaktowy z masterem

W kubernetes opis Podów umieszczany jest w plikach yaml w obiektach: Deployment lub ReplicationController

nginx-deployment.yaml 

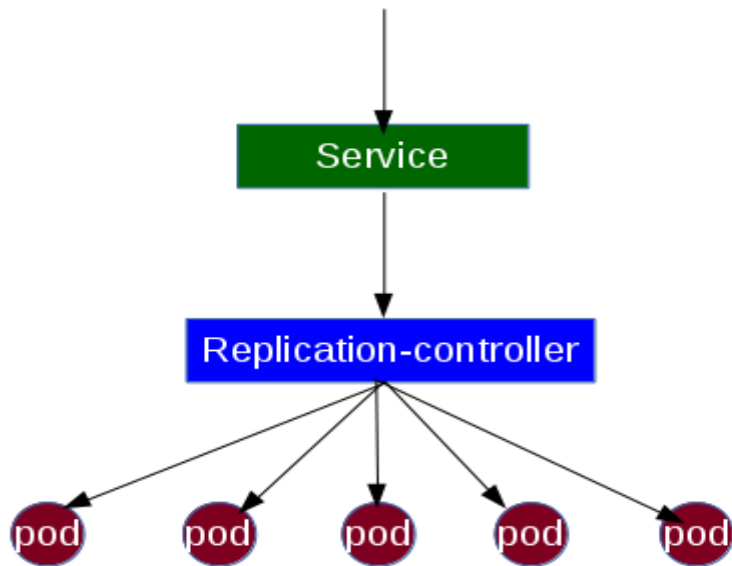
```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

Service jest abstrakcją, która definiuje logiczny zestaw podów i politykę, za pomocą której można je uzyskać.

Pody są grupowane i wybierane za pomocą label selektorów

```
{  
  "kind": "Service",  
  "apiVersion": "v1",  
  "metadata": {  
    "name": "my-service"  
  },  
  "spec": {  
    "selector": {  
      "app": "MyApp"  
    },  
    "ports": [  
      {  
        "protocol": "TCP",  
        "port": 80,  
        "targetPort": 9376  
      }  
    ]  
  }  
}
```

# Pods + Service



HTTP Port : 8000	MYSQL Port : 3306	WORDPRESS Port : 8001
Replicas = 2	Replicas = 1	Replicas = 2
Pod x 2	Pod x 1	Pod x 2

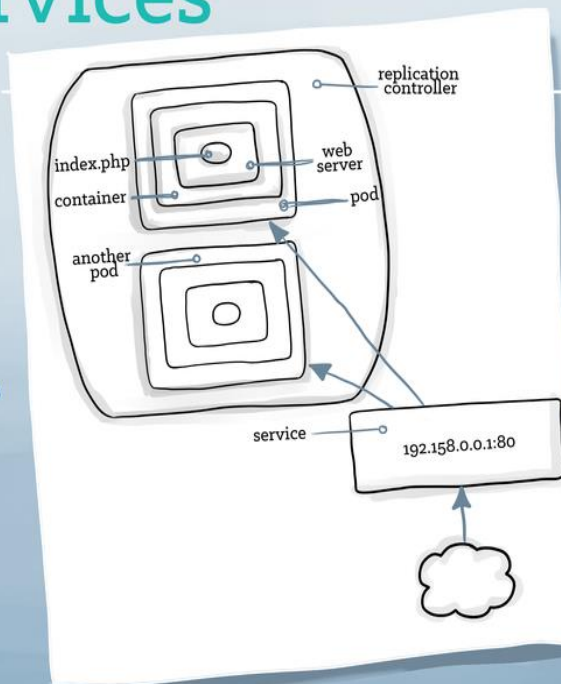
Docker host

Docker host

Docker host

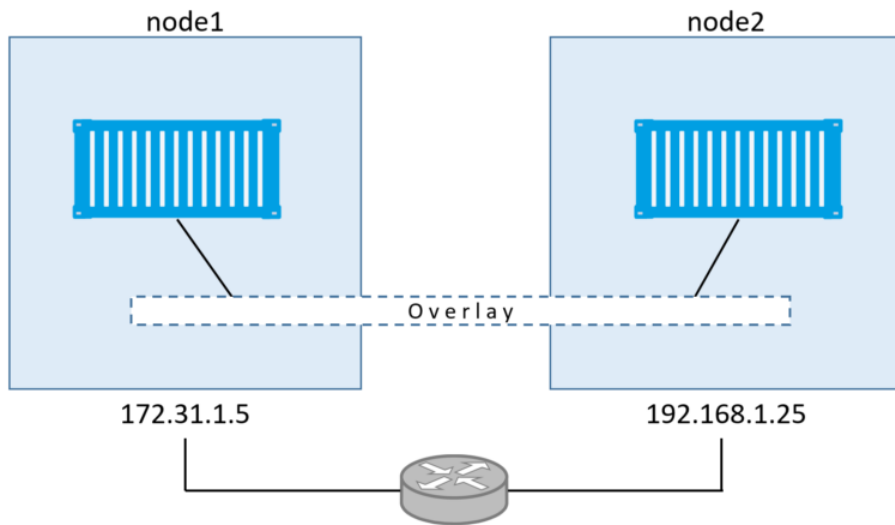
## Services

- Persistent
- Provide discovery
- Provide load balancing
- Provide stable service address
- Find pods by label selector



# Networking – overlay network

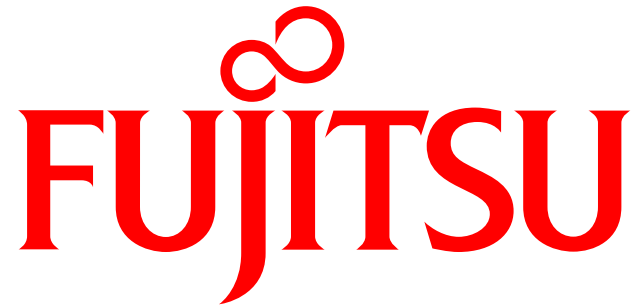
- Sieć logiczna budowana na bazie istniejącej fizycznej sieci
- Punkty tej sieci łączą się poprzez tunele





<https://www.youtube.com/watch?v=4ht22ReBjno>





shaping tomorrow with you