Архитектура Kubernetes

Kubernetes

Kubernetes – это программная система, которая позволяет легко разворачивать контейнеризированные приложения и управлять ими

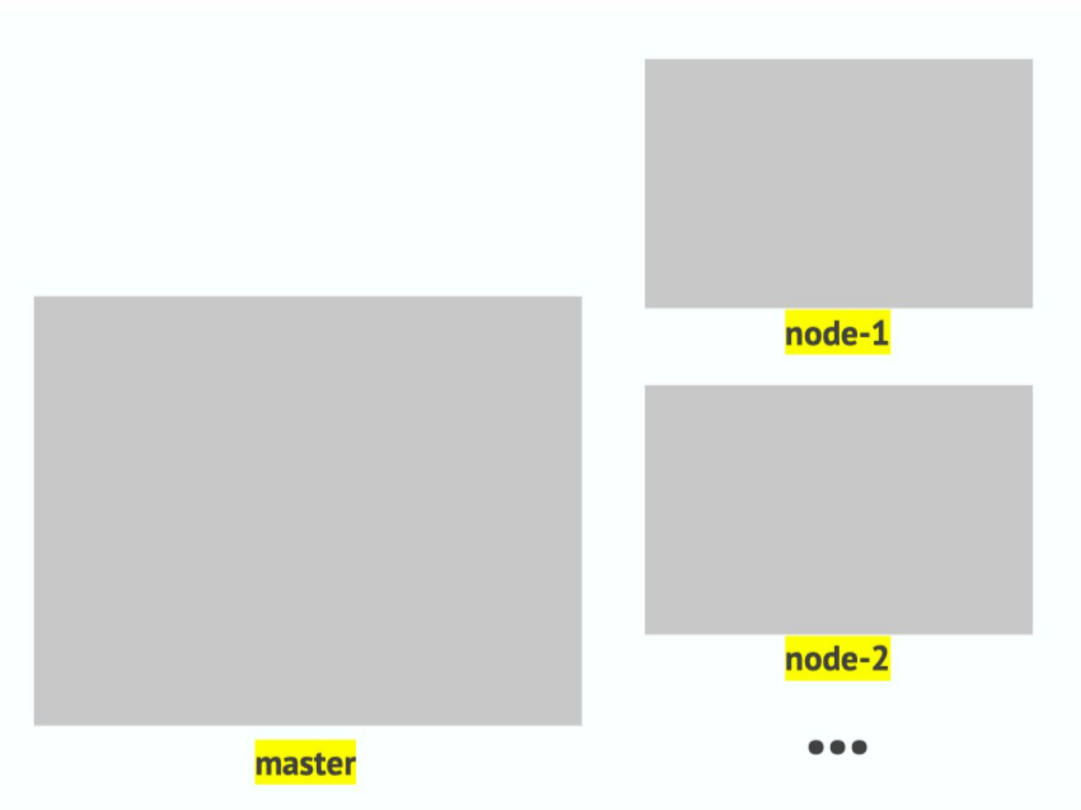
Она использует возможности контейнеров Linux для запуска разнородных приложений без необходимости знать какие-либо внутренние детали этих приложений и без необходимости вручную разворачивать эти приложения на каждом хосте Kubernetes позволяет выполнять ваши программные приложения на тысячах компьютерных узлов, как если бы все эти узлы были одним огромным компьютером

Kubernetes. Альтернативы.

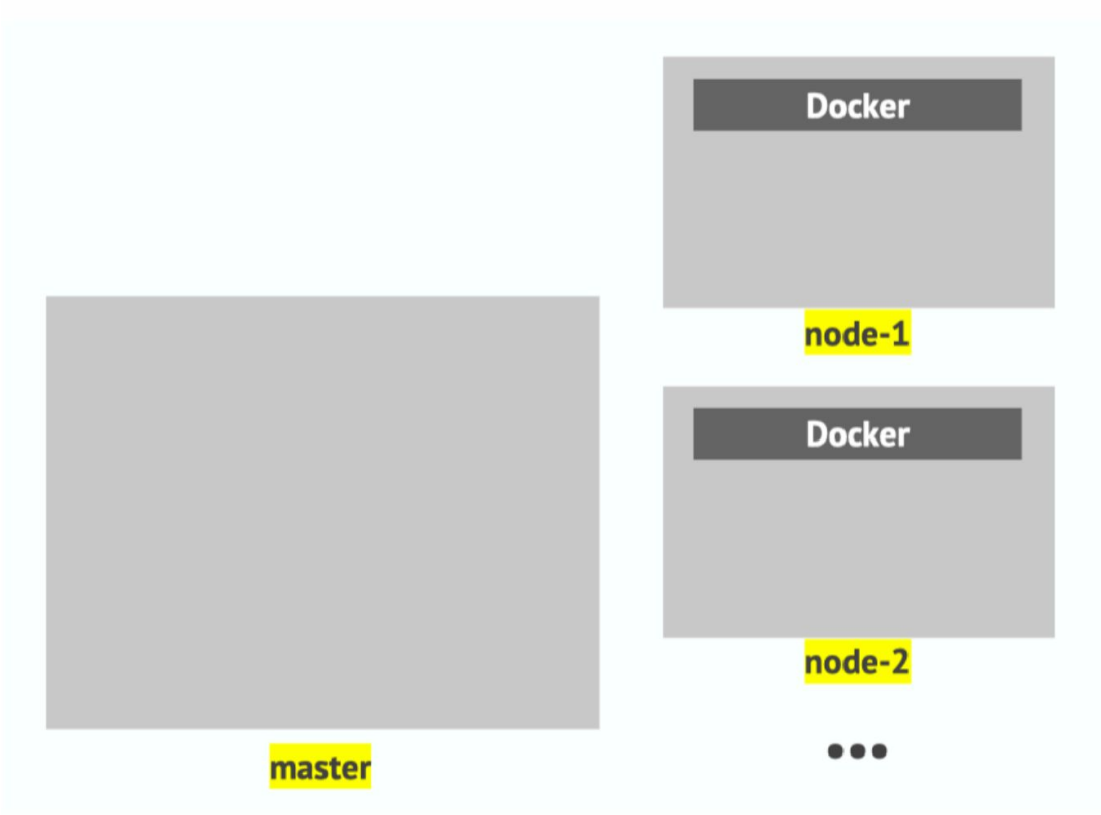
На текущий момент является самым популярным, самым развиваемым opensource проектом. На его основе создаются коммерческие продукты: openshift, tanzu, rancher и т.д, и managed кластера кубернетеса.

Nomad, mesos, dc/os, dockerswarm – развиваются похуже и зачастую уступают.

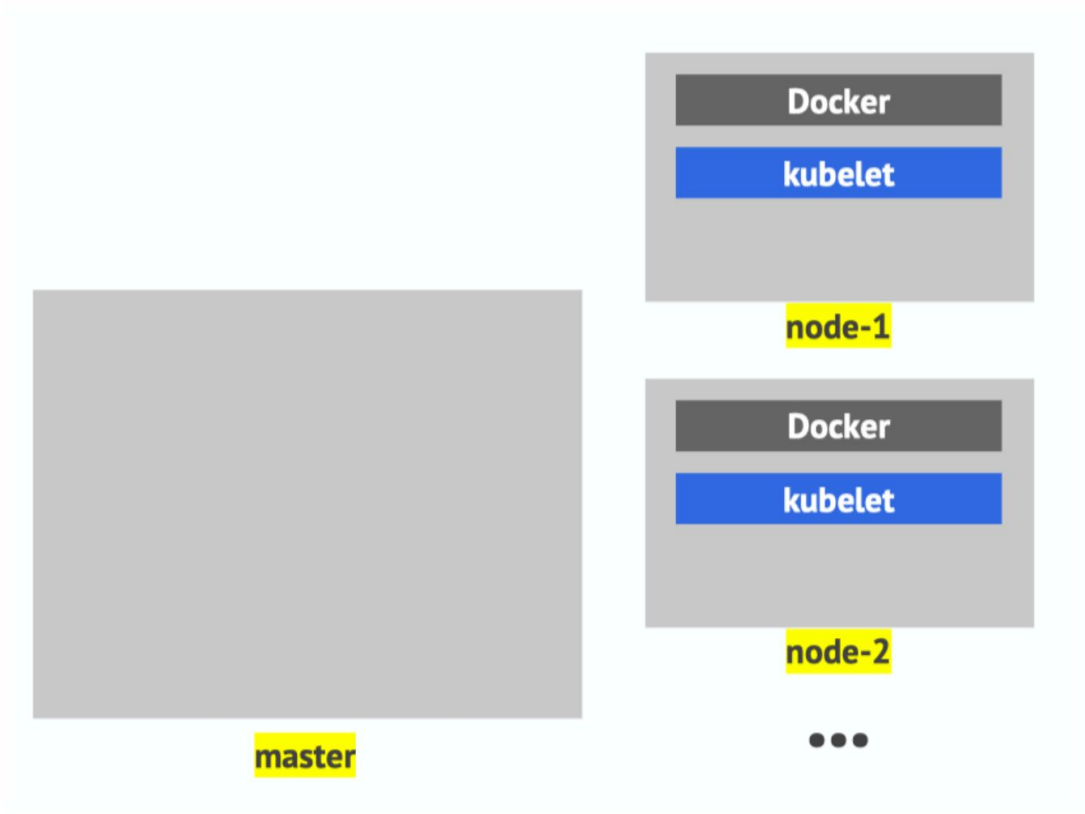
Архитектура Kubernetes



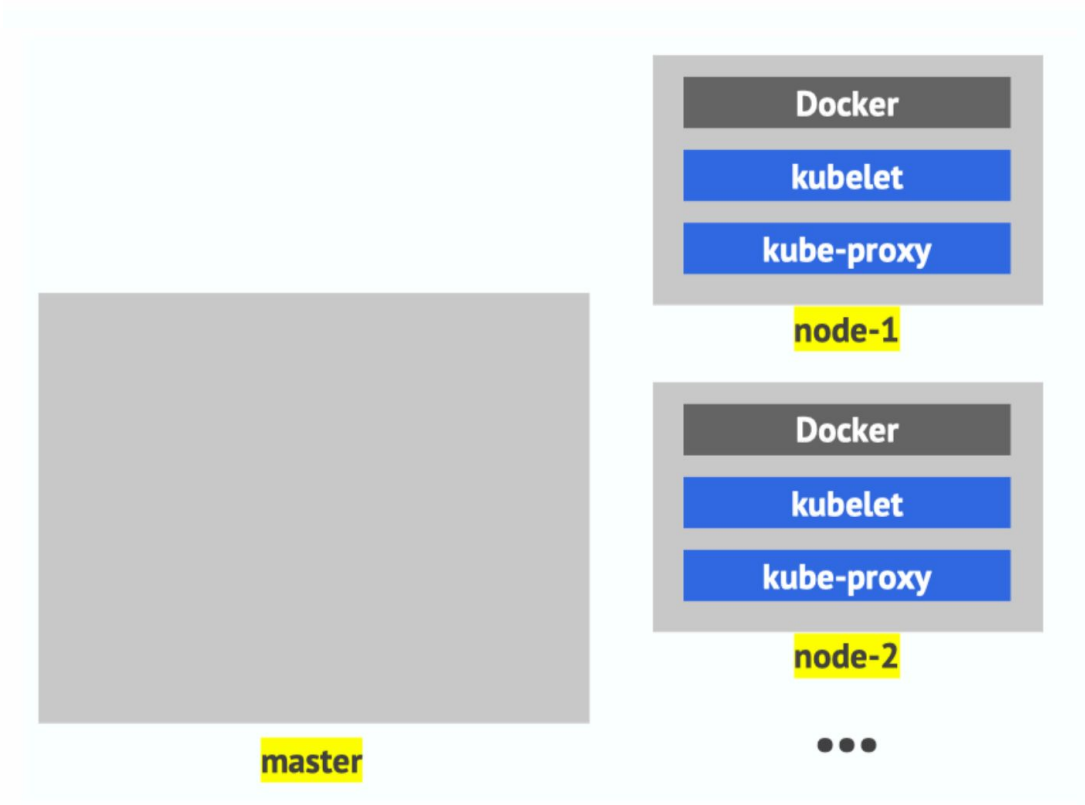
Архитектура Kubernetes



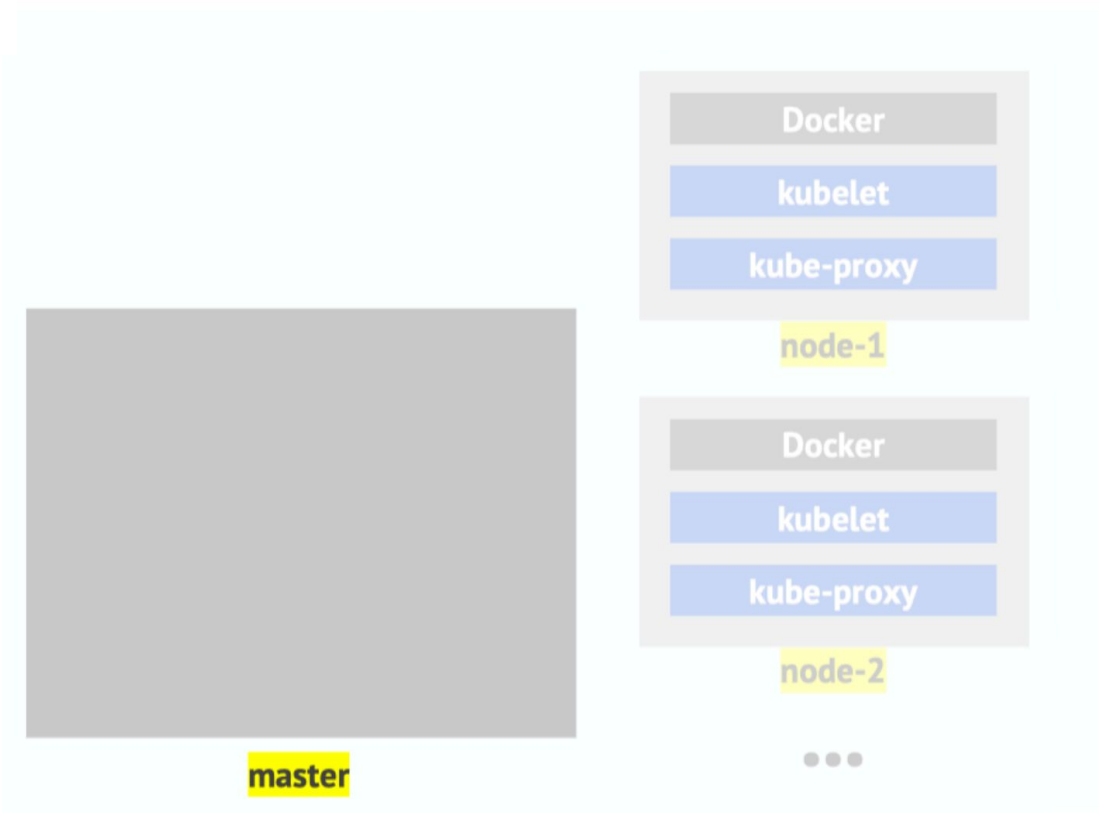
Архитектура Kubernetes



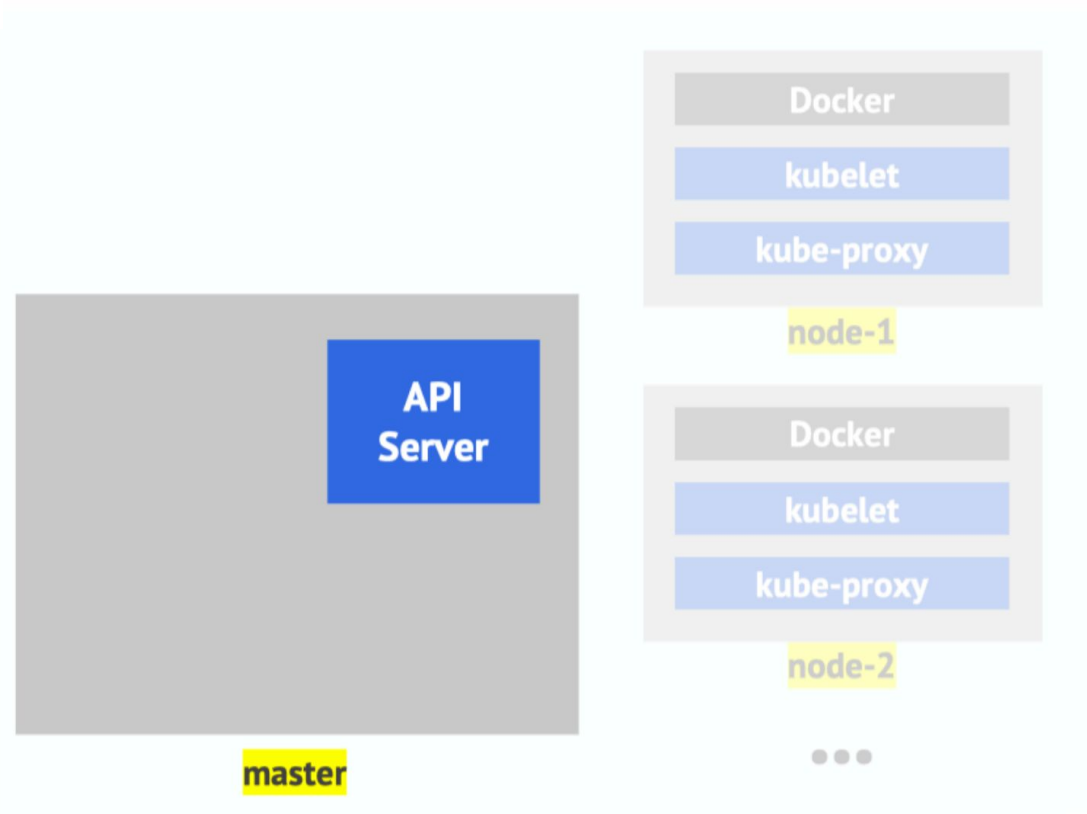
Архитектура Kubernetes



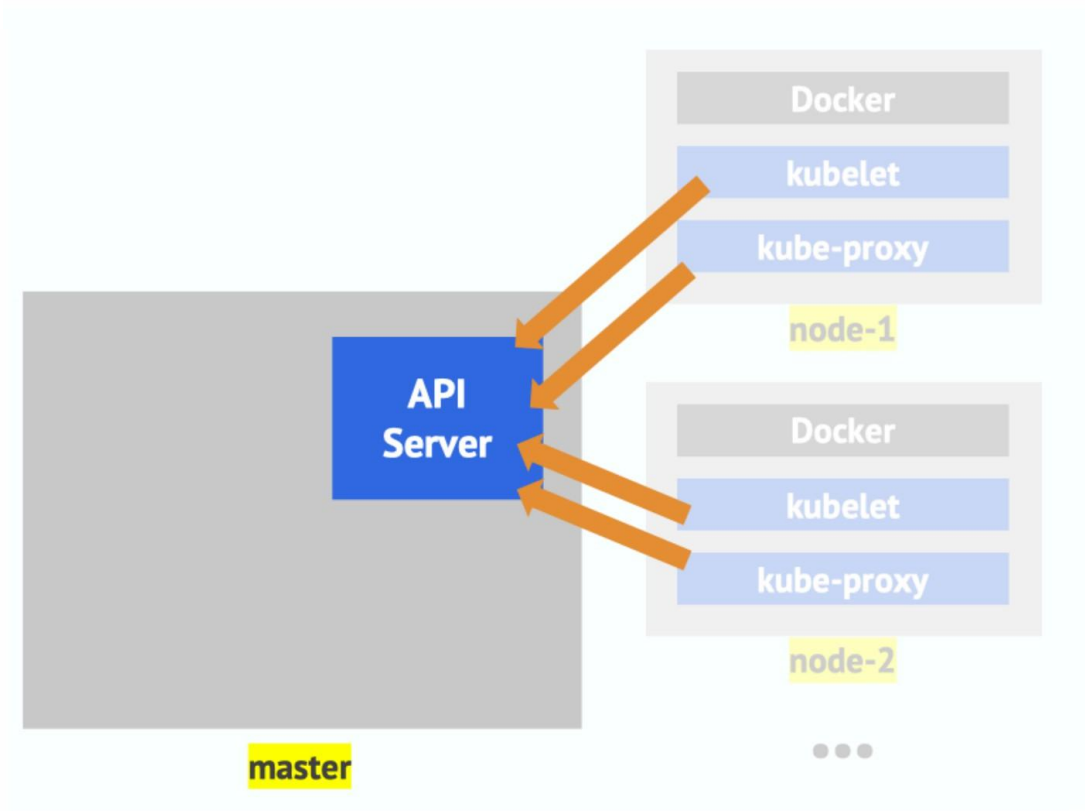
Архитектура Kubernetes



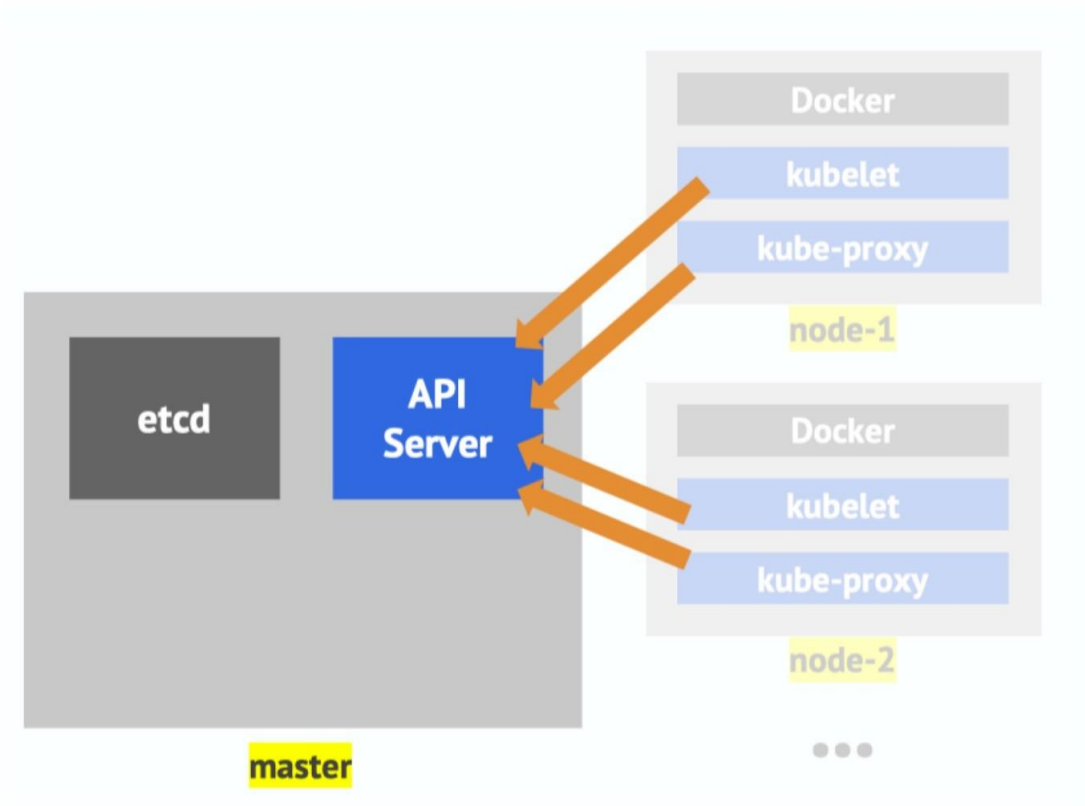
Архитектура Kubernetes



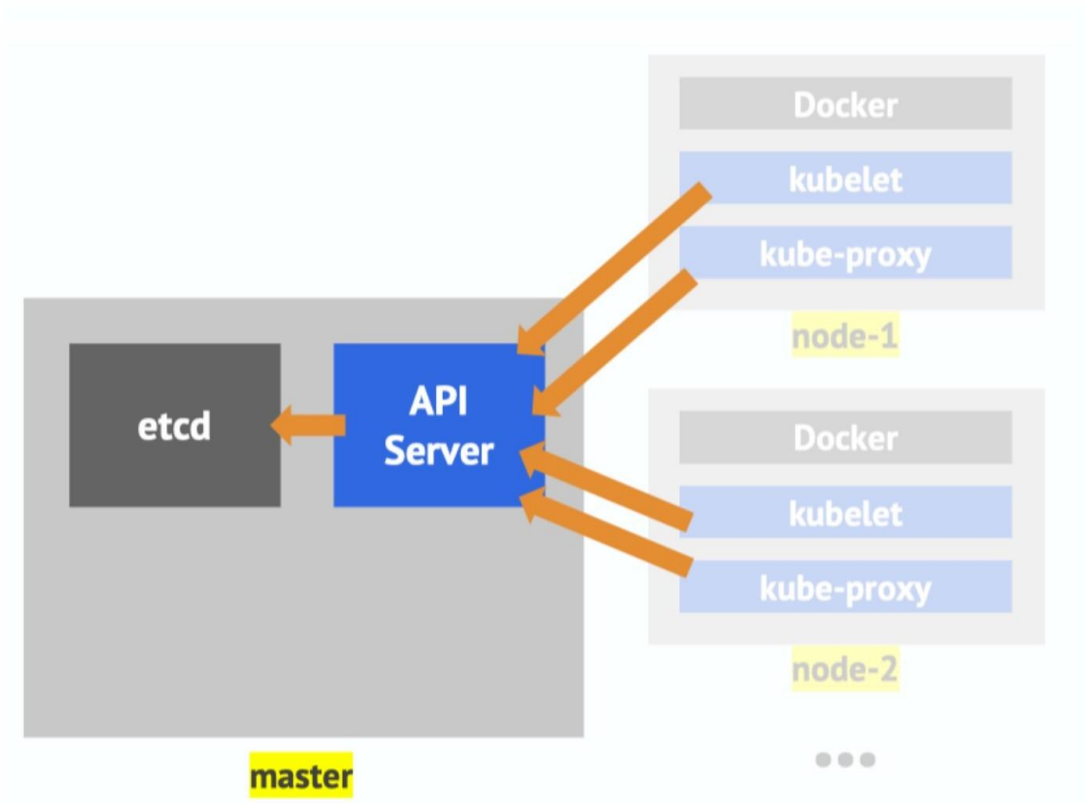
Архитектура Kubernetes



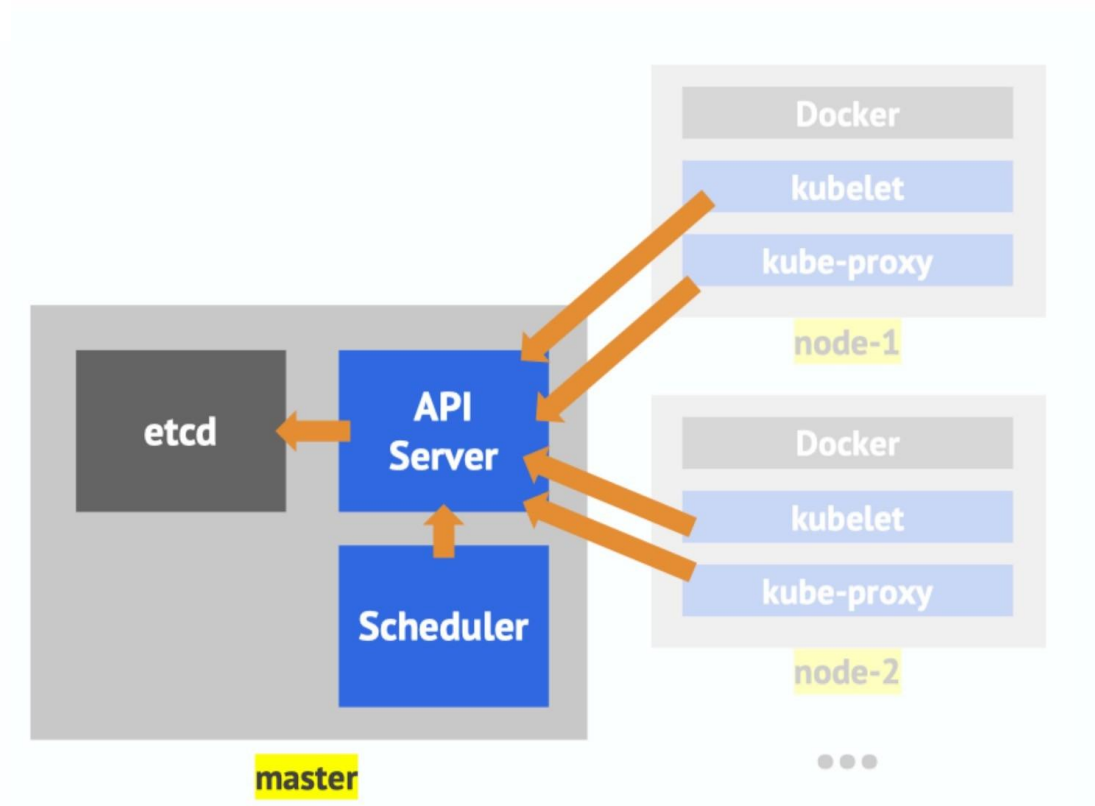
Архитектура Kubernetes



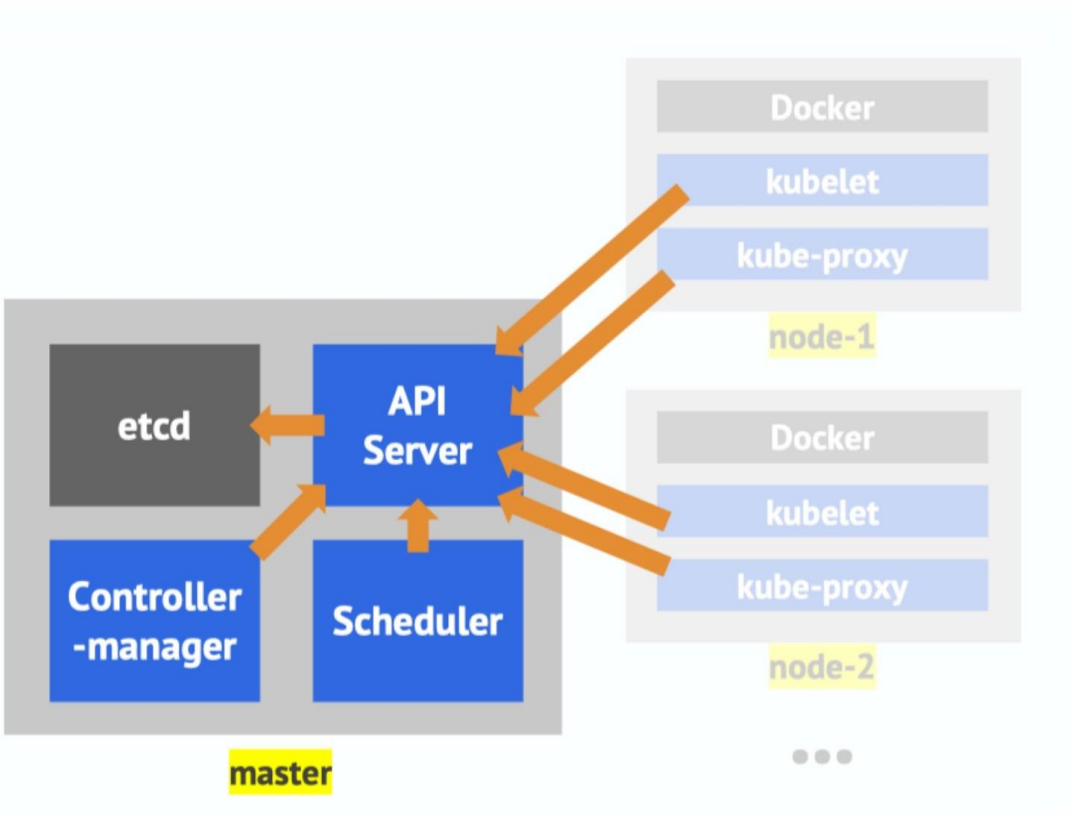
Архитектура Kubernetes



Архитектура Kubernetes



Архитектура Kubernetes



Инструменты для работы с кластером

Обращение напрямую к API

curl -X GET $APISERVER/api --header "Authorization: Bearer $TOKEN" --insecure

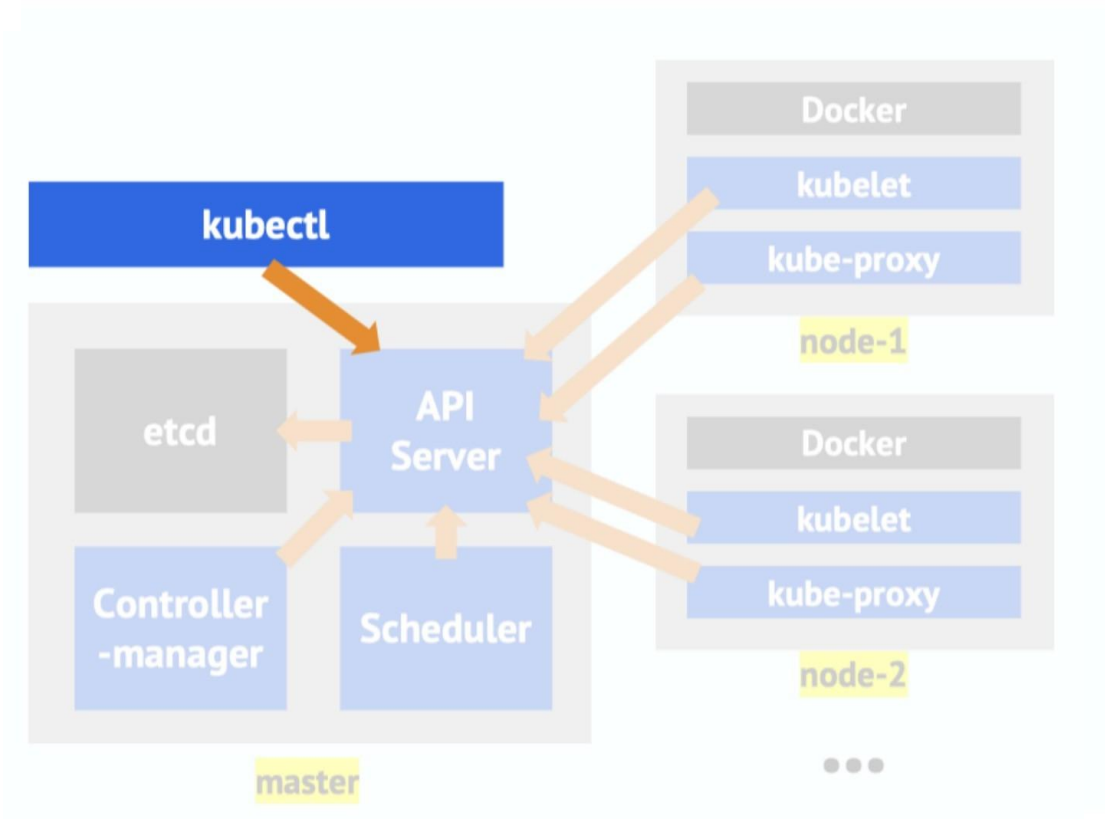

```
{
  "kind": "APIVersions",
  "versions": [
    "v1"
  ],
  "serverAddressByClientCIDRs": [
    {
      "clientCIDR": "0.0.0.0/0",
      "serverAddress": "10.0.1.149:443"
    }
  ]
}
```

Kubectl

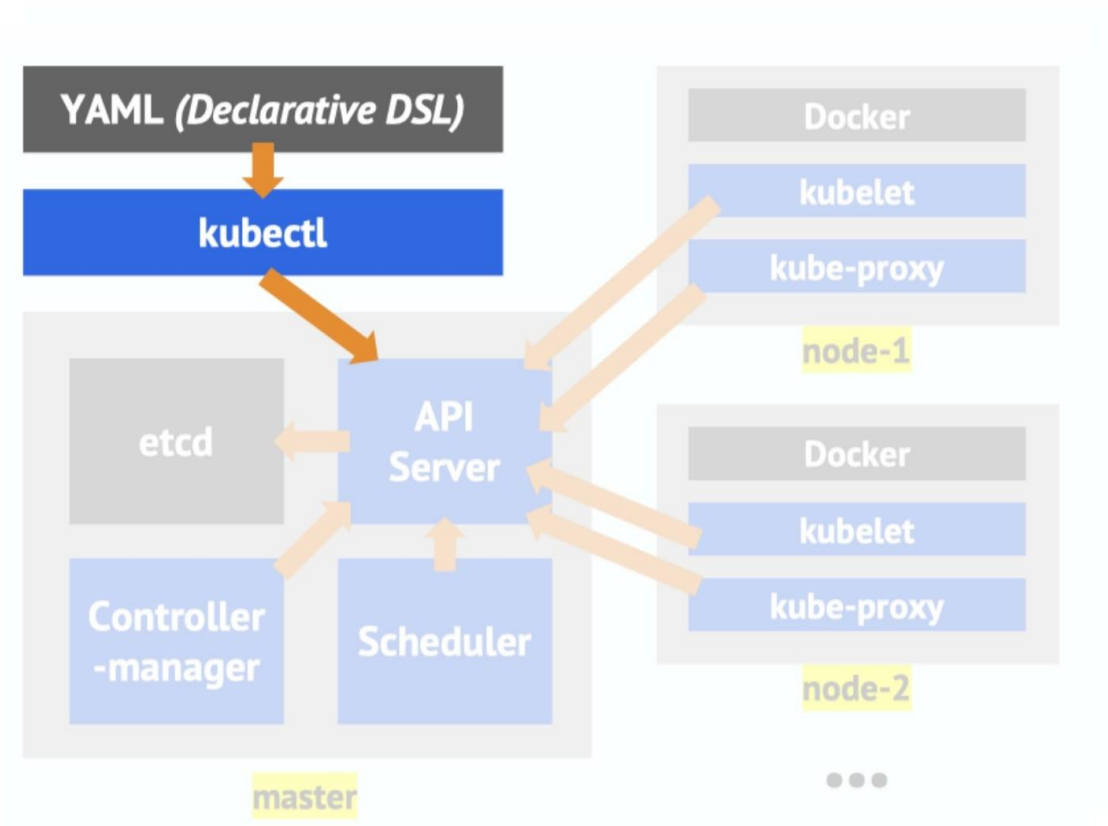
Kubectl – утилита коммандной строки, позволяющая отправлять запросы к Control plane кластера используя Kubernetes API

Инструкция по установке Kubectl

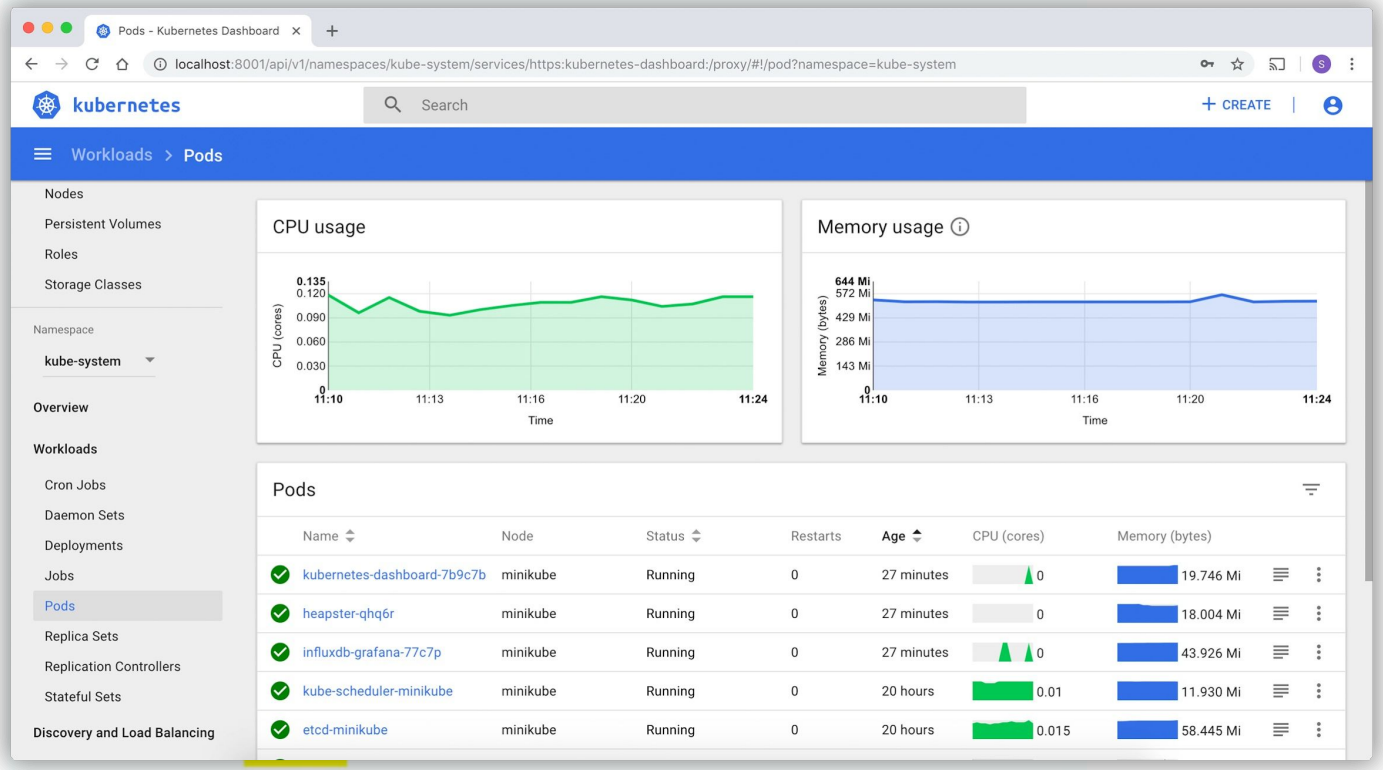
Работа с кластером. Kubectl.



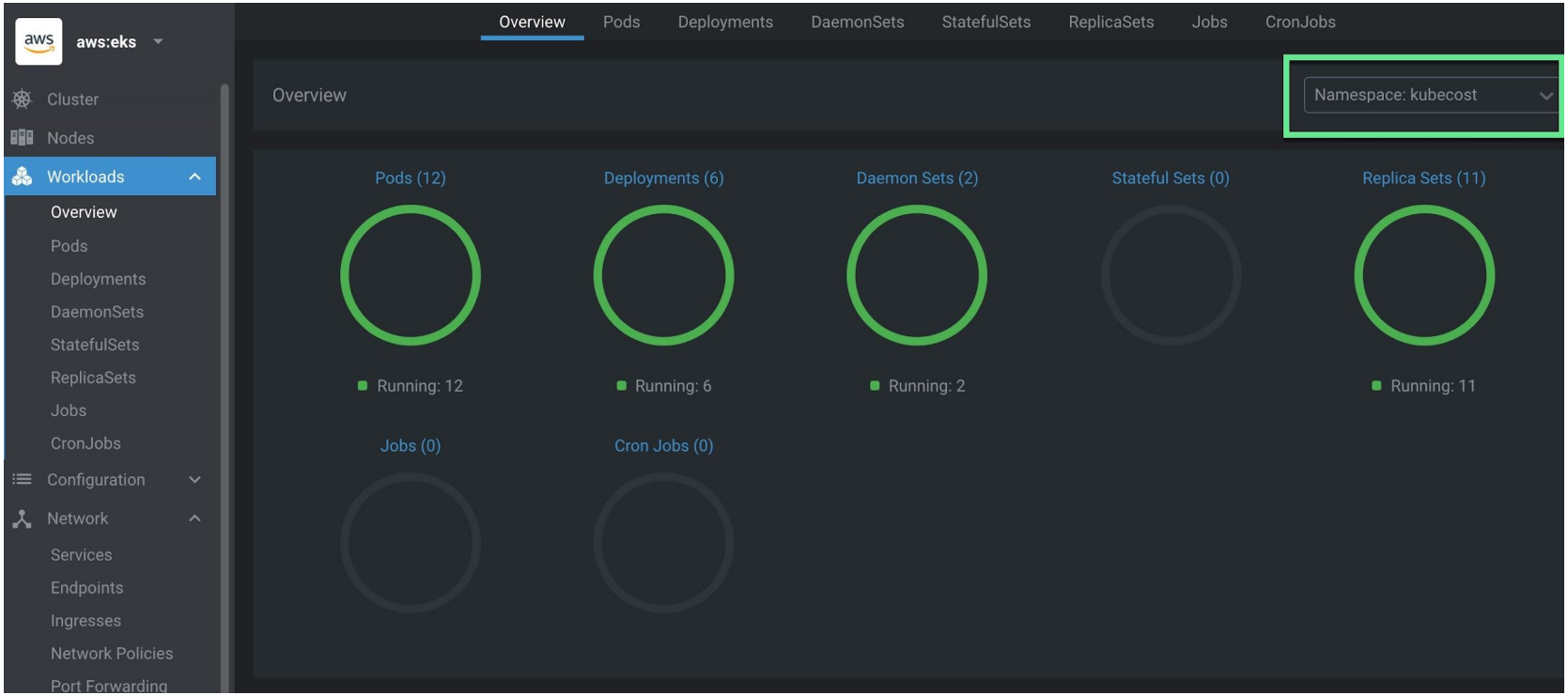
Работа с кластером. Kubectl.



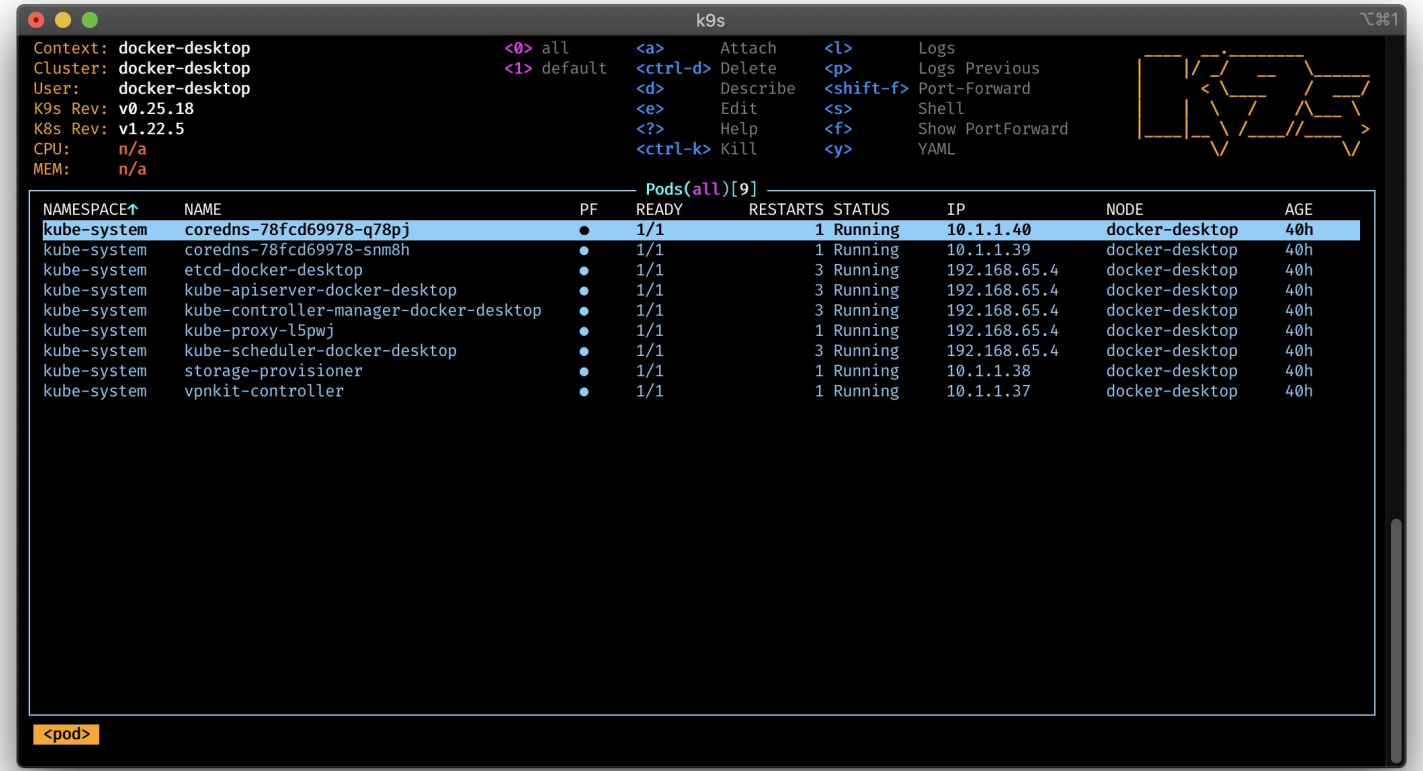
Kubernetes dashboard



Lens


aws:eks
Cluster
Nodes
Workloads
Overview
Pods
Deployments
DaemonSets
StatefulSets
ReplicaSets
Jobs
CronJobs
Configuration
Network
Services
Endpoints
Ingresses
Network Policies
Port Forwarding
Overview
Pods
Deployments
DaemonSets
StatefulSets
ReplicaSets
Jobs
CronJobs
Overview
Namespace: kubecost
Pods (12)
Running: 12
Deployments (6)
Running: 6
Daemon Sets (2)
Running: 2
Stateful Sets (0)
Replica Sets (11)
Running: 11
Jobs (0)
Cron Jobs (0)

k9s


k9s
⌥⌘1
Context: docker-desktop
Cluster: docker-desktop
User: docker-desktop
K9s Rev: v0.25.18
K8s Rev: v1.22.5
CPU: n/a
MEM: n/a
<0> all
<1> default
<a> Attach
<ctrl-d> Delete
<d> Describe
<e> Edit
<?> Help
<ctrl-k> Kill
<l> Logs
<p> Logs Previous
<shift-f> Port-Forward
<s> Shell
<f> Show PortForward
<y> YAML
Pods(all)[9]
NAMESPACE↑ NAME PF READY RESTARTS STATUS IP NODE AGE
kube-system coredns-78fcd69978-q78pj ● 1/1 1 Running 10.1.1.40 docker-desktop 40h
kube-system coredns-78fcd69978-snm8h ● 1/1 1 Running 10.1.1.39 docker-desktop 40h
kube-system etcd-docker-desktop ● 1/1 3 Running 192.168.65.4 docker-desktop 40h
kube-system kube-apiserver-docker-desktop ● 1/1 3 Running 192.168.65.4 docker-desktop 40h
kube-system kube-controller-manager-docker-desktop ● 1/1 3 Running 192.168.65.4 docker-desktop 40h
kube-system kube-proxy-l5pwj ● 1/1 1 Running 192.168.65.4 docker-desktop 40h
kube-system kube-scheduler-docker-desktop ● 1/1 3 Running 192.168.65.4 docker-desktop 40h
kube-system storage-provisioner ● 1/1 1 Running 10.1.1.38 docker-desktop 40h
kube-system vpnkit-controller ● 1/1 1 Running 10.1.1.37 docker-desktop 40h
<pod>

Базовые сущности Kubernetes

Объекты Kubernetes

Объекты в Kubernetes – это хранящиеся внутри Kubernetes сущности (ресурсы). И Kubernetes их использует для представления состояния кластера.

Объекты могут описывать:

- Какие приложения сейчас запущены и работают
- Какие ресурсы доступны этим приложениям
- Политики поведения, такие как политика рестартов, обновлений и т.д.

Для того, чтобы изменить желаемое состояние, например:

- увеличить количество инстансов сервиса с 2 до 3
- полностью потушить деплой
- убрать сервис из балансировки и т.д.

Необходимо просто удалить или изменить объекты

Объекты Kubernetes

Все объекты в Kubernetes имеют общий формат описания в yaml

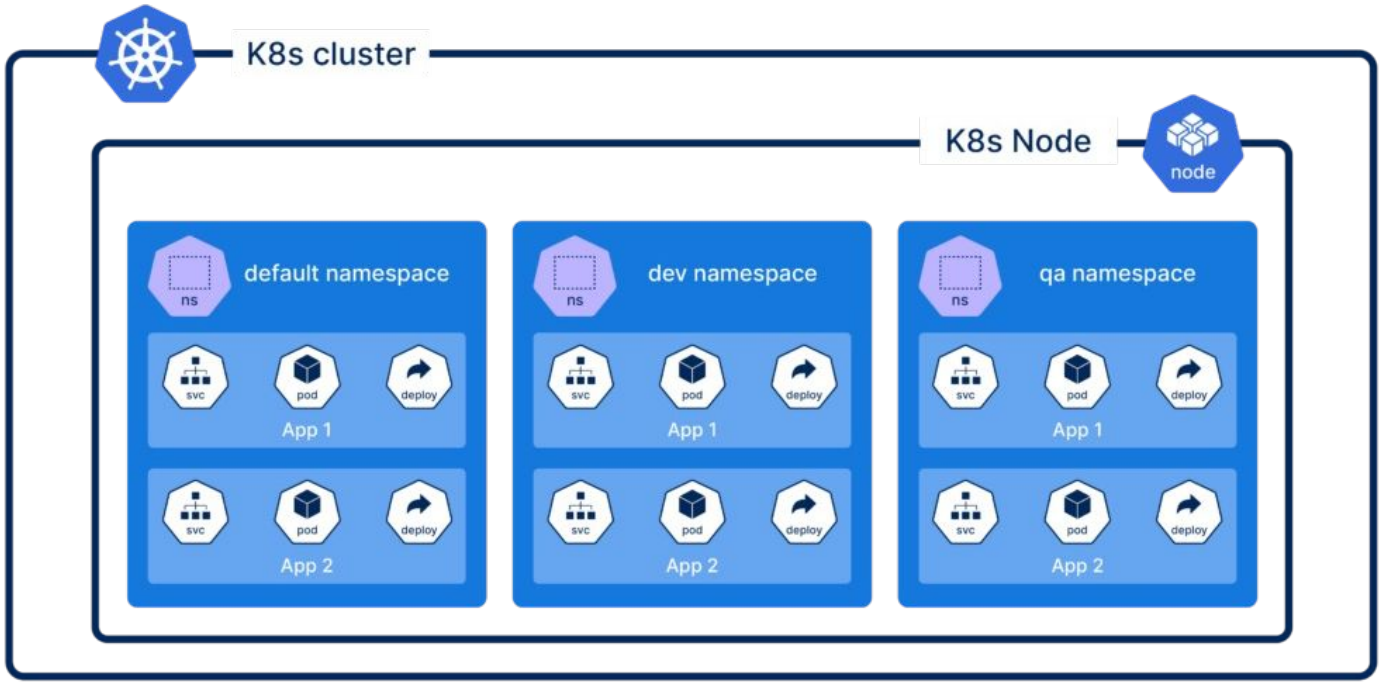
kind – тип (класс) объекта
apiVersion – группа api
metadata – метаданные: имя, метки, аннотации
spec – желаемое состояние объекта

Пространства имен

Namespace – способ изоляции объектов внутри одного кластера k8s.

- Имена объектов должны быть уникальны в рамках одного namespace (но могут повторяться в рамках разных)
- Применяются только к объектам
- Можно задавать права доступа, квоты, лимиты и т.д в рамках ns

Пространства имен



Метки и селекторы

Зачастую необходимо работать не с одним конкретным объектом, а сразу с несколькими. Для того чтобы находить необходимые объекты и ссылаться на них используются метки и селекторы

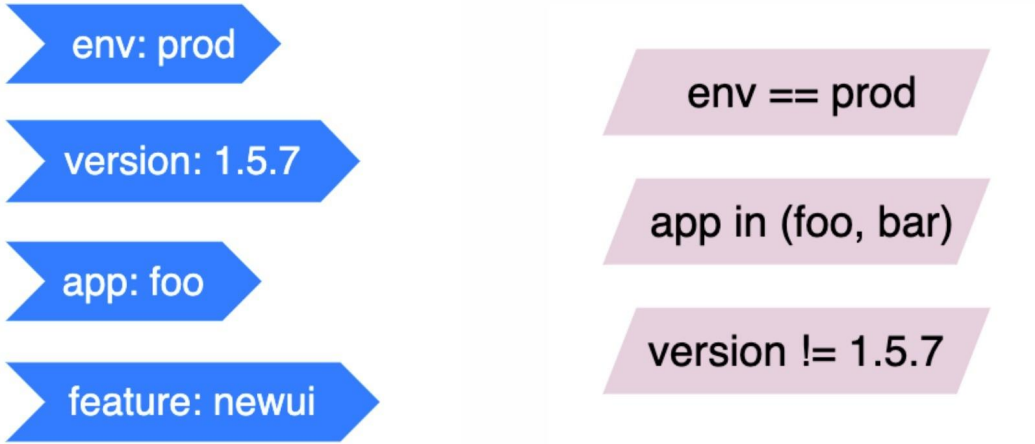
Метки – значения типа ключ : значение

Селекторы – выражения, позволяющие выбрать нужные объекты по меткам

Метки и селекторы

Селекторы бывают

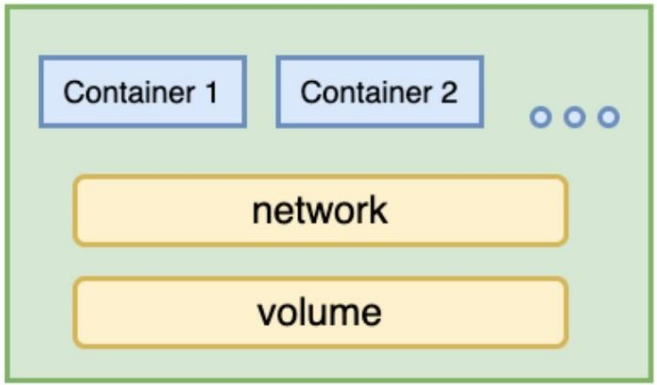
- Равенство (не равенство)
- set-based (in, notin, exist)



Pod

Pod

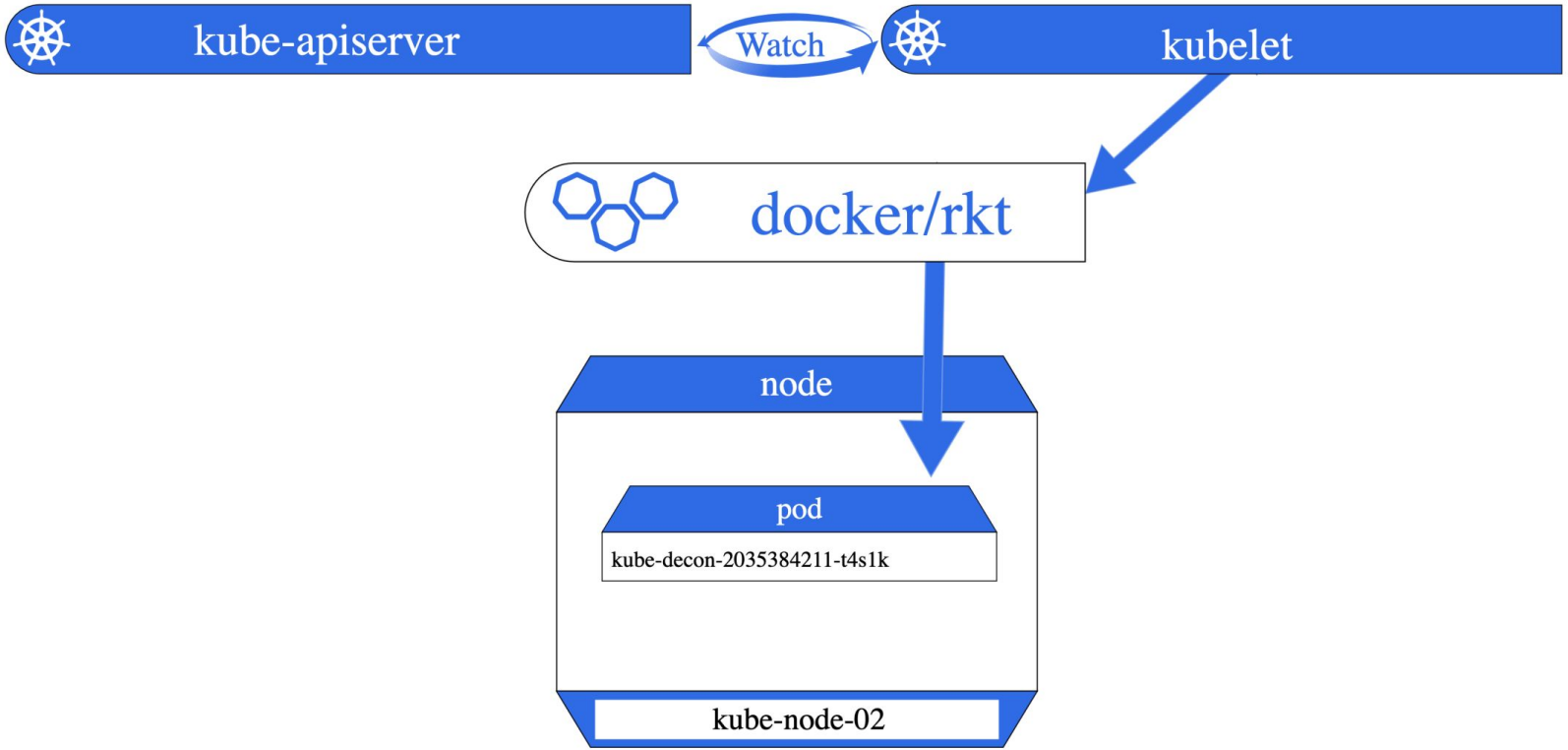
Pod – это объект Kubernetes, который является атомарной единицей рабочей нагрузки.
Pod можно воспринимать, как запущенный инстанс сервиса или приложения
При этом внутри пода может быть один или несколько контейнеров.
Внутри пода контейнеры разделяют сетевой интерфейс и имеют общее локальное хранилище
Kubernetes запускает, удаляет и обновляет контейнеры, описанные в Pod-е как единое целое.



Pod. Пример конфигурации

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

Pod. Kubectl


kube-apiserver
Watch
kubelet
docker/rkt
node
pod
kube-decon-2035384211-t4s1k
kube-node-02

Pod. Пробы

Виды проб:

- Liveness
- Readiness
- Startup

Способ

- Http get
- TCP
- Command exec

Pod. Пробы

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-exec
spec:
  containers:
  - name: liveness
    image: registry.k8s.io/busybox
    args:
    - /bin/sh
    - -c
    - touch /tmp/healthy; sleep 30; rm -f /tmp/healthy; sleep 600
    livenessProbe:
      exec:
        command:
        - cat
        - /tmp/healthy
      initialDelaySeconds: 5
      periodSeconds: 5
```

Pod. Volumes

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test-webserver
spec:
  os: { name: linux }
  nodeSelector:
    kubernetes.io/os: linux
  containers:
  - name: test-webserver
    image: registry.k8s.io/test-webserver:latest
    volumeMounts:
    - mountPath: /var/local/aaa
      name: mydir
    - mountPath: /var/local/aaa/1.txt
      name: myfile
  volumes:
  - name: mydir
    hostPath:
      # Ensure the file directory is created.
      path: /var/local/aaa
      type: DirectoryOrCreate
  - name: myfile
    hostPath:
      path: /var/local/aaa/1.txt
      type: FileOrCreate
```

Pod. Управление планированием на ноды

- Позволяют задать определенные ноды на которых может/должен быть запущен под
  - nodeSelector
  - nodeAffinity & nodeAntiAffinity
- Позволяют задать правила, какие поды могут/должны работать на одной ноде, а какие на различных
  - podAffinity & podAntiAffinity
- Позволяют запретить планирование/исполнение всех подов на определенных нодах
  - Taints & tolerations

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: disktype
            operator: In
            values:
            - ssd
```