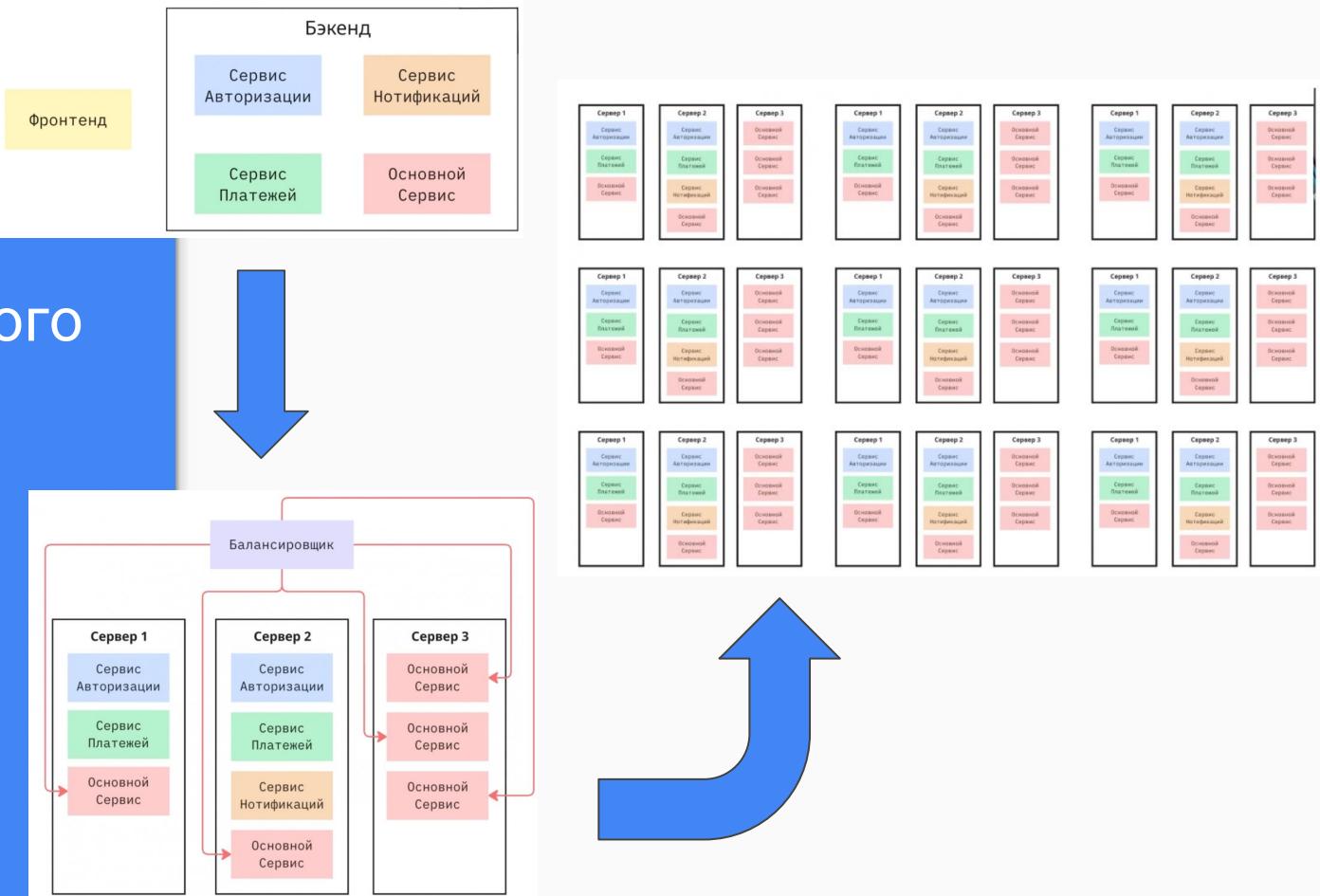


# Kubernetes внутреннее устройство

Подготовили студенты:  
Соков Д.А, Соломатов А.Д

# Проблема одного сайта.



# Что такое Kubernetes

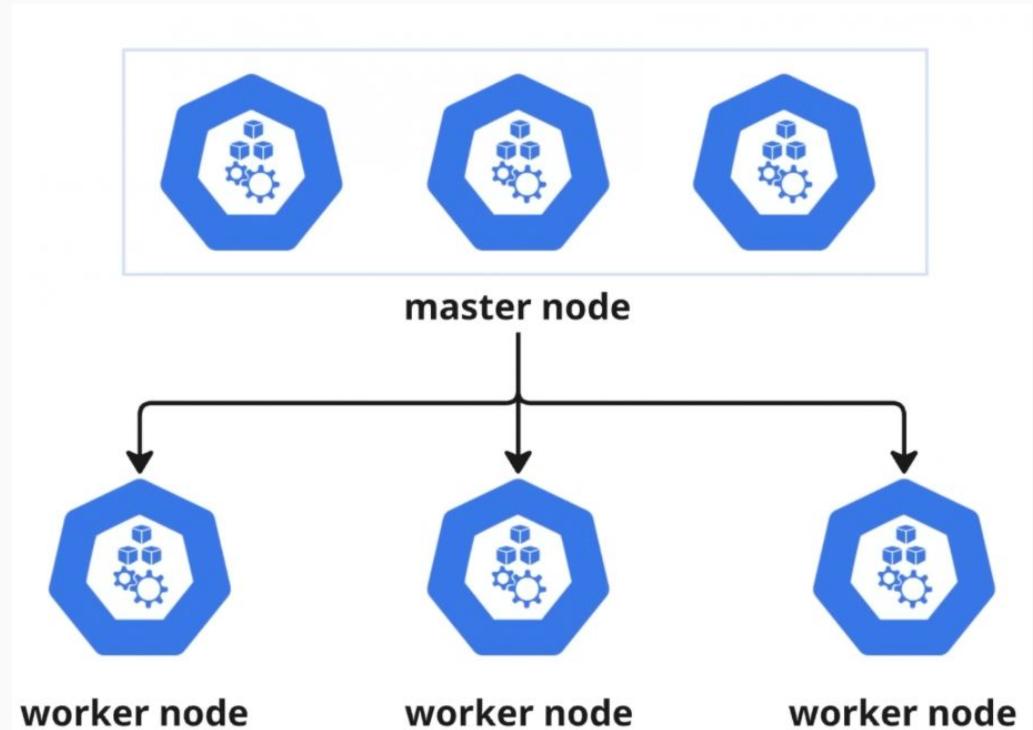
- это система для автоматизации **развертывания, масштабирования и управления** контейнеризированными приложениями.



**kubernetes**

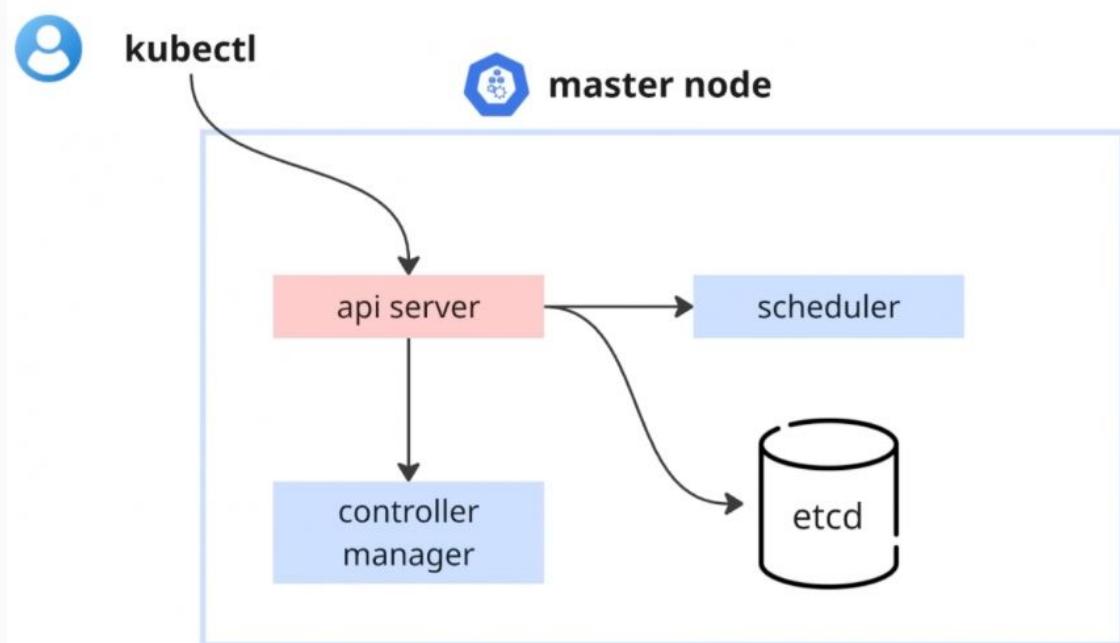
Но Kubernetes не хранит данные и образы приложений, а также не собирает Docker-образы.

# Архитектура Kubernetes: Высокоуровневый взгляд



# Master Node Components

- **kubectl:** Инструмент управления (клиент).
- **API Server:** Единая точка входа, аутентификация, валидация.
- **etcd:** Хранилище состояния (Key-Value, Raft Consensus).
- **Scheduler:** Планирование размещения подов.
- **Controller Manager:** Обеспечение желаемого состояния.



# Установка k8s

Для macOS (через Homebrew):

```
[● ● ●] brew install kubectl
```

Для Windows (через PowerShell/Winget):

```
[● ● ●] winget install -e --id Kubernetes.kubectl  
# Или если есть Chocolatey: choco install kubernetes-cli
```

Для Linux:

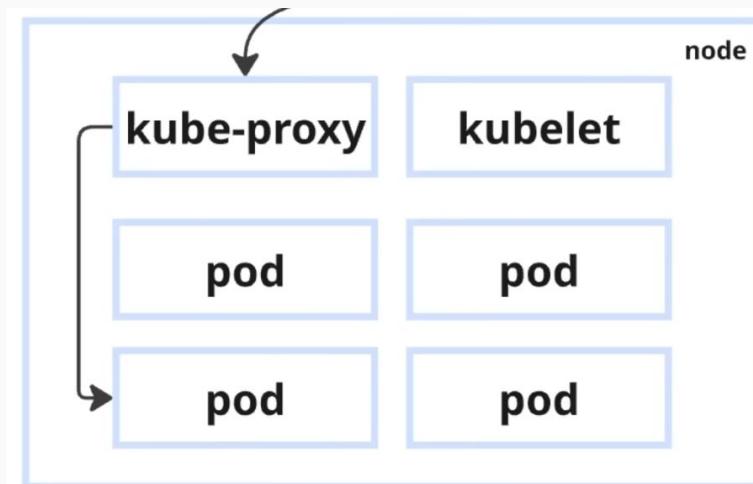
```
[● ● ●] curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"  
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

# Worker Node Components

**Kubelet:** Агент узла. Общается с API Server.

**Container Runtime (CRI):** containerd / CRI-O  
(работа с контейнерами).

**Kube-proxy:** Сетевые правила (Service abstraction).



# Пример DEMO-1 START

Config: kind-config.yaml

```
● ● ●

kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
- role: worker
- role: worker
```

# Пример созданного кластера

```
PS C:\Users\Hp\OneDrive\Рабочий стол\Kubernetes\DEMO 1> kind create cluster --  
config kind-config.yaml --name demo-cluster
```

```
PS C:\Users\Hp\OneDrive\Рабочий стол\Kubernetes\DEMO 1> kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
demo-cluster-control-plane	Ready	control-plane	2m4s	v1.34.0
demo-cluster-worker	Ready	<none>	113s	v1.34.0
demo-cluster-worker2	Ready	<none>	113s	v1.34.0

# Вид в Docker

	Name	Image	Status	CPU (%)	Port(s)	Actions
<input type="checkbox"/>	<a href="#"> demo-cluster-worker</a> de8155b88840 	<a href="#">kindest/no</a>	Running	1.37%		  
<input type="checkbox"/>	<a href="#"> demo-cluster-worker2</a> be0f47c545a0 	<a href="#">kindest/no</a>	Running	1.8%		  
<input type="checkbox"/>	<a href="#"> demo-cluster-control-plane</a> 5b1637e2c66a 	<a href="#">kindest/no</a>	Running	14.4%	<a href="#">52301:6443</a>	  

# Вид в Docker внутри Control Plane

CONTAINER ATTEMPT	IMAGE	CREATED	STATE	NAME	NAMESPACE	RESTARTS
8bd29be9b1617	52546a367cc9e	2 minutes ago	Running	coredns	kube-system	0
6728430f143bf	cddc146a35aed	coredns-66bc5c9577-jh2gf	2 minutes ago	Running	local-path-provisioner	0
	bbb6209cc873b				local-path-storage	
385a5bdcc8f99	bd8501aeba996	local-path-provisioner-7b8c8ddbd6-zc5bl	2 minutes ago	Running	coredns	0
52546a367cc9e	5f3d6a5530b1e	coredns-66bc5c9577-228wp	2 minutes ago	Running	kube-system	0
2d047b101bab3	409467f978b4a	coredns-66bc5c9577-228wp	3 minutes ago	Running	kindnet-cni	0
	681cde2da594d	kindnet-5kn9k			kube-system	
e1d03e65d5886	df0860106674d	kindnet-5kn9k	3 minutes ago	Running	kube-proxy	0
	13a236d11e2b0	kube-proxy-4vt7c			kube-system	
507f68d711del	5f1f5298c888d	kube-proxy-4vt7c	3 minutes ago	Running	etcd	0
	2a27d0ed6f36e	etcd-demo-cluster-control-plane			kube-system	
1c7287c82ef10	90550c43ad2bc	etcd-demo-cluster-control-plane	3 minutes ago	Running	kube-apiserver	0
	010809b452cae	kube-apiserver-demo-cluster-control-plane			kube-system	
02f799ebff412	a0af72f2ec6d6	kube-apiserver-demo-cluster-control-plane	3 minutes ago	Running	kube-controller-manager	0
	13aacb528d108	kube-controller-manager-demo-cluster-control-plane			kube-system	
6db962b8bee5	46169d968e920	kube-controller-manager-demo-cluster-control-plane	3 minutes ago	Running	kube-scheduler	0
	8ea2ff738f2e6	kube-scheduler				

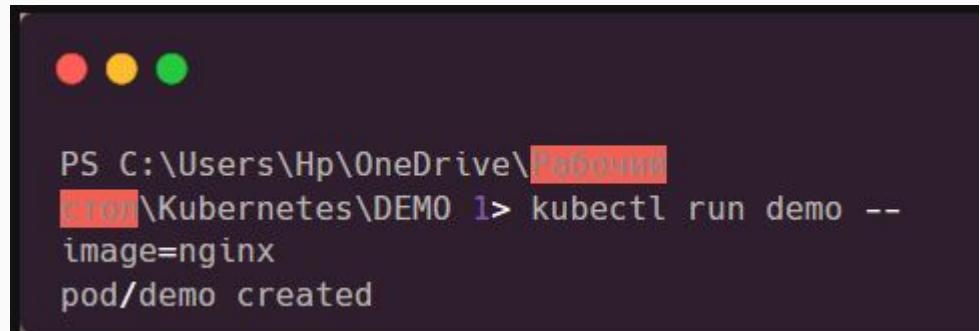
# Внутри Worker Node

```
# crictl ps
CONTAINER          IMAGE               CREATED             STATE              NAME                ATTEMPT
POD ID            POD                 3 minutes ago     Running            kindnet-cni           0
e523ca4651c88    409467f978b4a      kindnet-fh4g7       kube-system
687807bcd8cb     df0860106674d      kube-proxy         kube-system
32651913e67b5    2ef1b10b8c3bc      kube-proxy-xv9nl   kube-system
```

# Жизненный цикл

# Pod – Атомарная единица

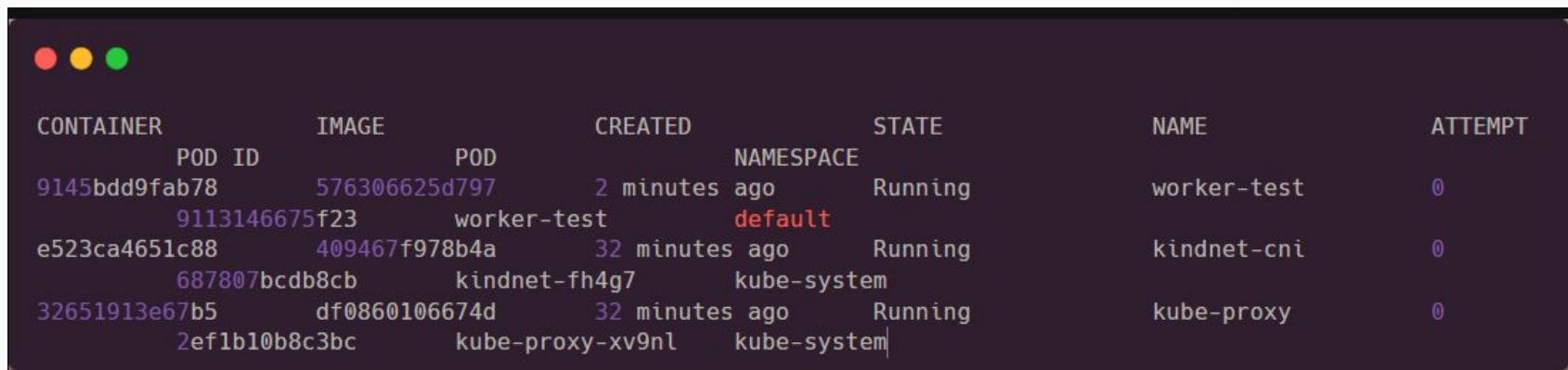
- Минимальная единица деплоя.
- Общий сетевой стек (Network Namespace, IP).
- Общие тома (Volumes).
- Эфемерность (может умереть и пересоздаться в другом месте).



```
PS C:\Users\Hp\OneDrive\Рабочий
STOP\Kubernetes\DEMO 1> kubectl run demo --
image=nginx
pod/demo created
```

A screenshot of a Windows terminal window. The title bar shows three colored dots (red, yellow, green). The command prompt is PS C:\Users\Hp\OneDrive\Рабочий. The user has typed 'STOP\Kubernetes\DEMO 1> kubectl run demo --image=nginx' and has hit enter. The output shows 'pod/demo created'. The word 'STOP' is highlighted in red, and 'Рабочий' is also highlighted in red.

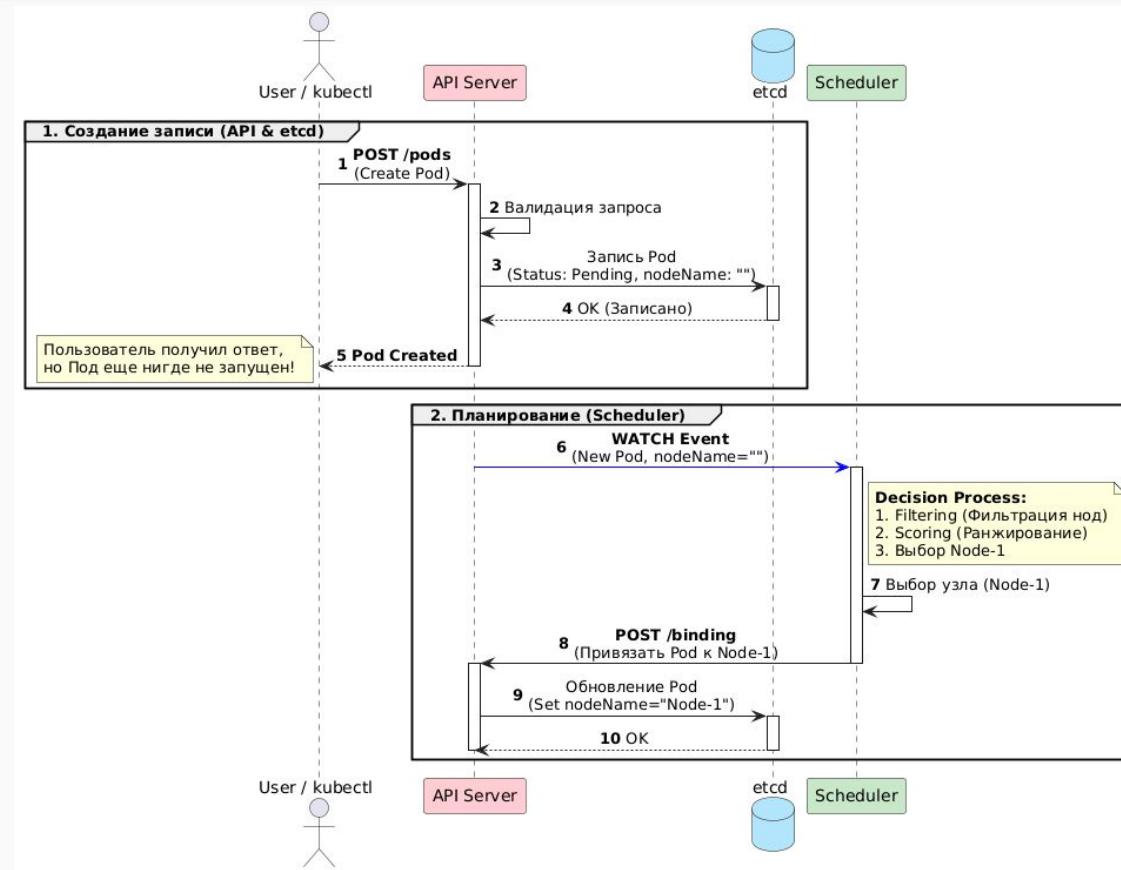
# Внутри Worker



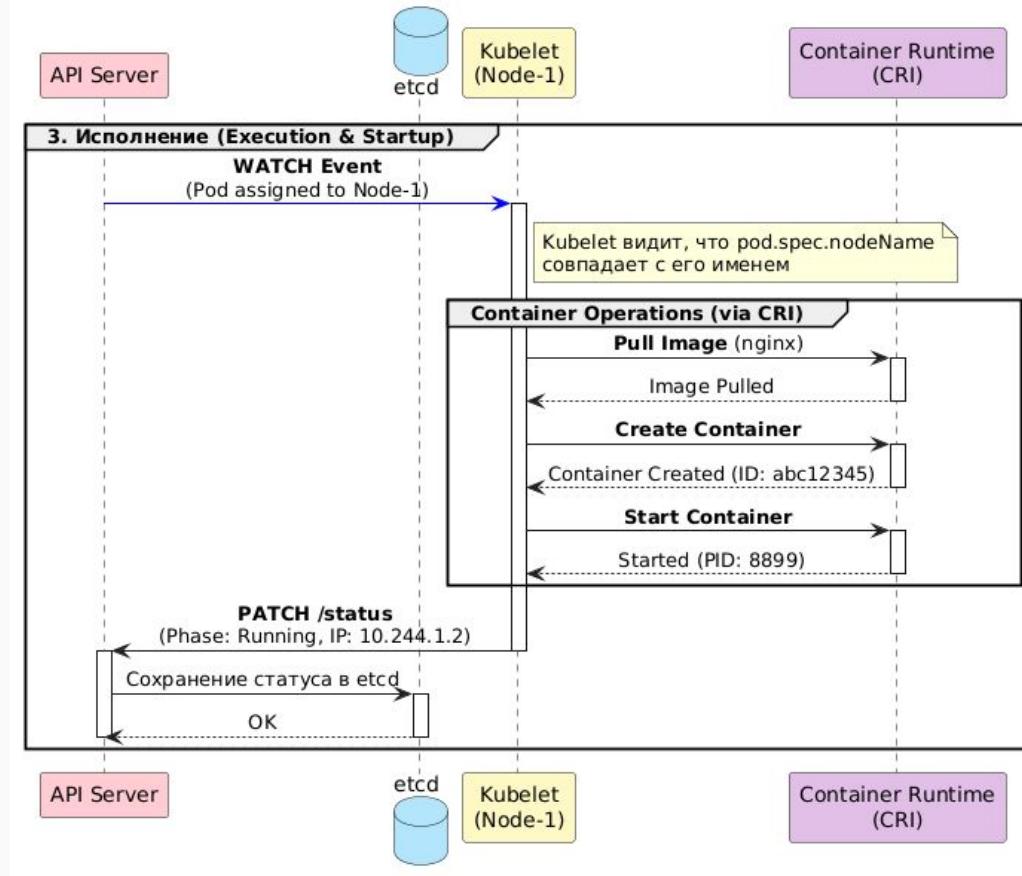
A terminal window displaying a table of running pods in a Kubernetes cluster. The table has columns for CONTAINER ID, IMAGE, POD ID, POD, CREATED, STATE, NAME, and ATTEMPT. The data shows four pods: worker-test, kindnet-cni, kube-proxy, and two system pods.

CONTAINER ID	IMAGE	POD ID	POD	CREATED	STATE	NAME	ATTEMPT
9145bdd9fab78	576306625d797	9113146675f23	worker-test	2 minutes ago	Running	worker-test	0
e523ca4651c88	409467f978b4a	687807bcdb8cb	kindnet-fh4g7	32 minutes ago	Running	kindnet-cni	0
32651913e67b5	df0860106674d	2ef1b10b8c3bc	kube-proxy-xv9nl	32 minutes ago	Running	kube-proxy	0

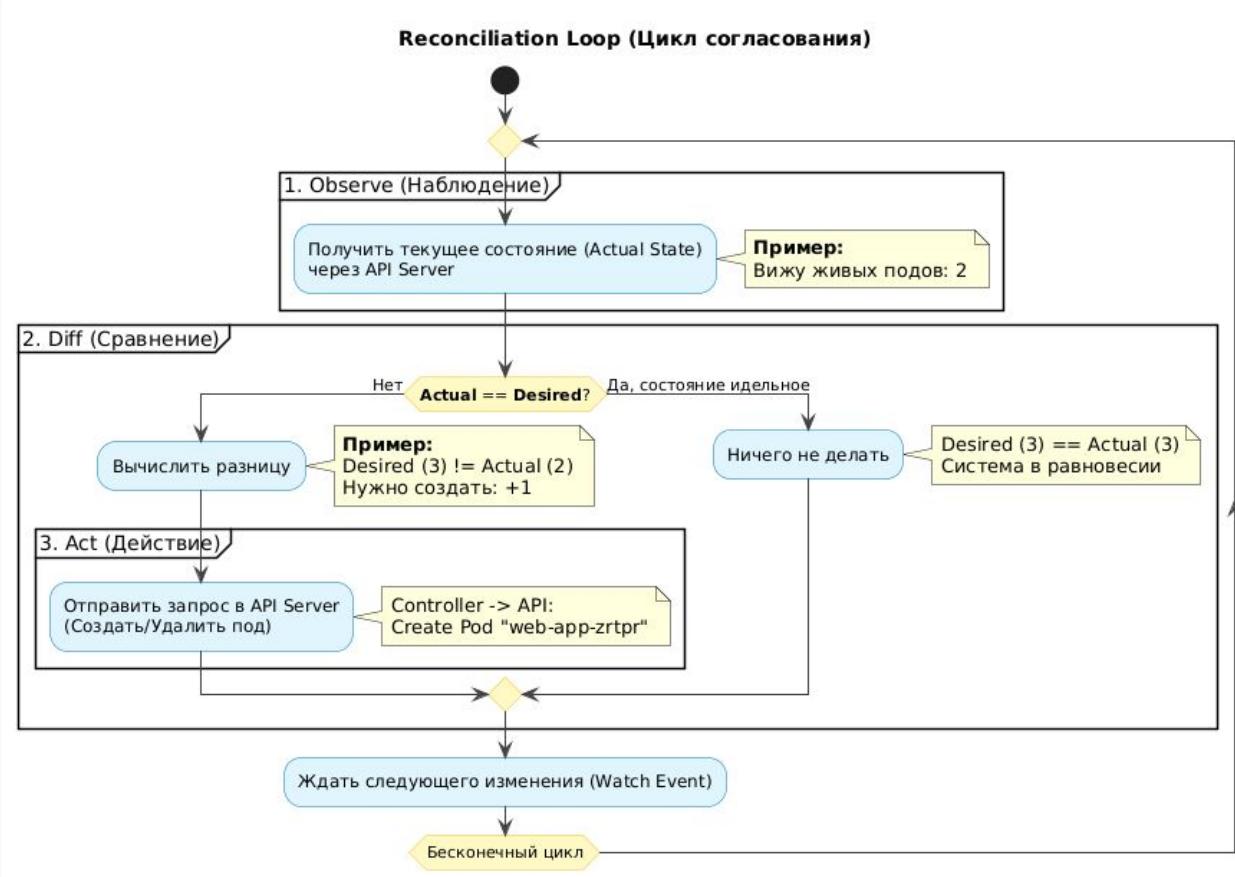
# Scheduling



# Execution



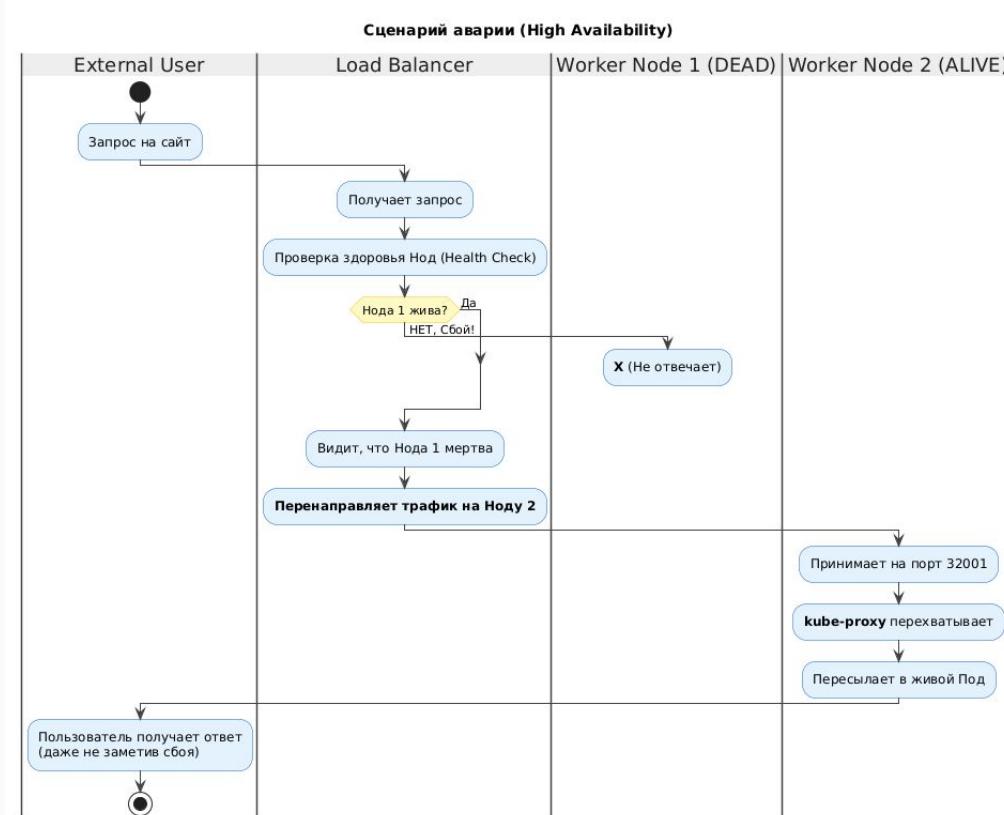
# Reconciliation Loop



# Сетевая модель (Внутреннее взаимодействие)

- **Проблема:** Поды "смертны", их IP меняются. Нужен постоянный адрес.
- **Решение:** Объект **Service** (ClusterIP).
- **Секрет:** Service IP — это **виртуальный IP**. Он не висит на сетевой карте.
- **Механика:**
  1. **kube-proxy** слушает API и обновляет правила ядра Linux (iptables/IPVS).
  2. Ядро перехватывает пакеты на Service IP.
  3. **DNAT:** Адрес назначения подменяется на IP реального Пода.

# Сетевая модель (Внешнее взаимодействие)



# ИТОГИ

«Подводя итог. Kubernetes — это сложная распределенная система, построенная на простых принципах:

1. Централизованное API и хранение состояния (etcd).
2. Разделение ответственности (Scheduler планирует, Kubelet исполняет).
3. Непрерывные циклы согласования (Controllers). Понимание этих процессов помогает не просто вводить команды, а эффективно диагностировать и проектировать отказоустойчивые системы»

# Список источников

<https://kubernetes.io/docs/reference/kubectl/>

<https://kind.sigs.k8s.io/>

<https://habr.com/ru/companies/slurm/articles/803739/>

<https://habr.com/ru/companies/selectel/articles/956266/>