Воспроизводимые окружения на bare-metal при помощи Talos Linux и Cozystack

ГЕОРГ ГААЛ

CTO & Founder of Ænix

DevOops
2024

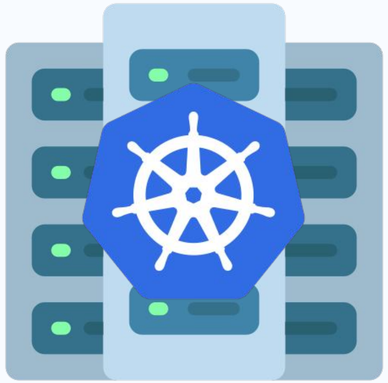
Кто я?

Георг Гаал

CTO & Founder of Ænix

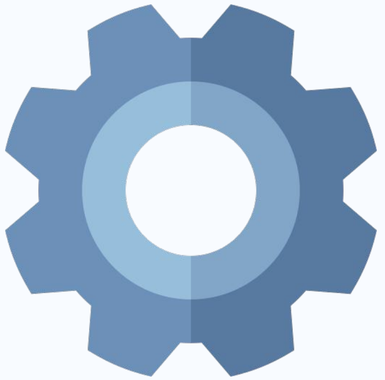
george.gaal@aenix.io

- 10+ лет в IT
- Из них не менее 7 - в крупных enterprise / fintech
- Начинал карьеру с низов
- Спикер крупнейших международных конференций и образовательных проектов
- Стараюсь дружить dev, ops и sec

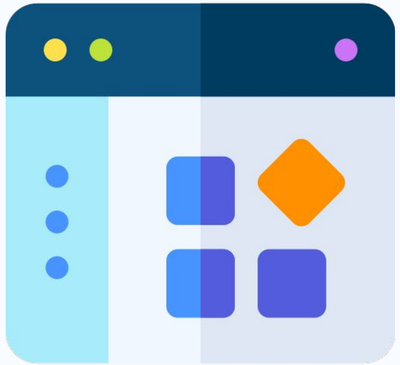


K8s на железе

Железо и ОС

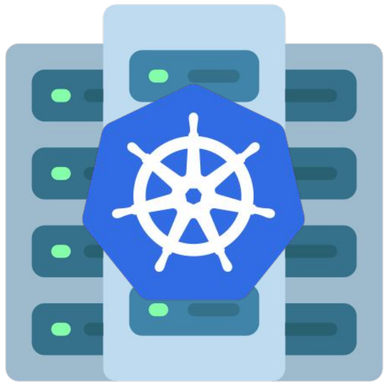


Системные компоненты



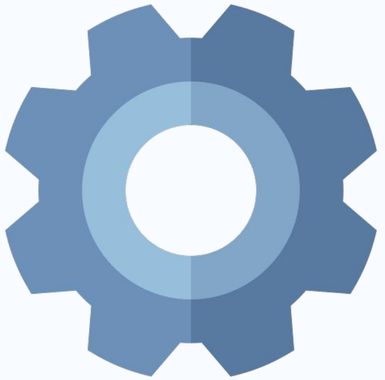
Пользовательские приложения

ПЛАТФОРМА

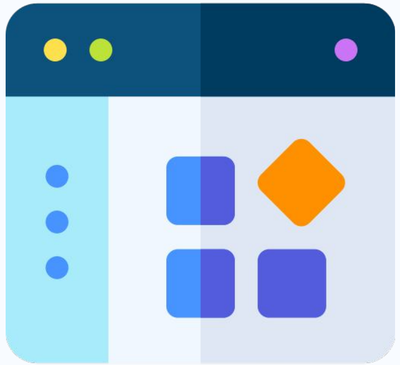


K8s на железе

Железо и ОС



Системные компоненты



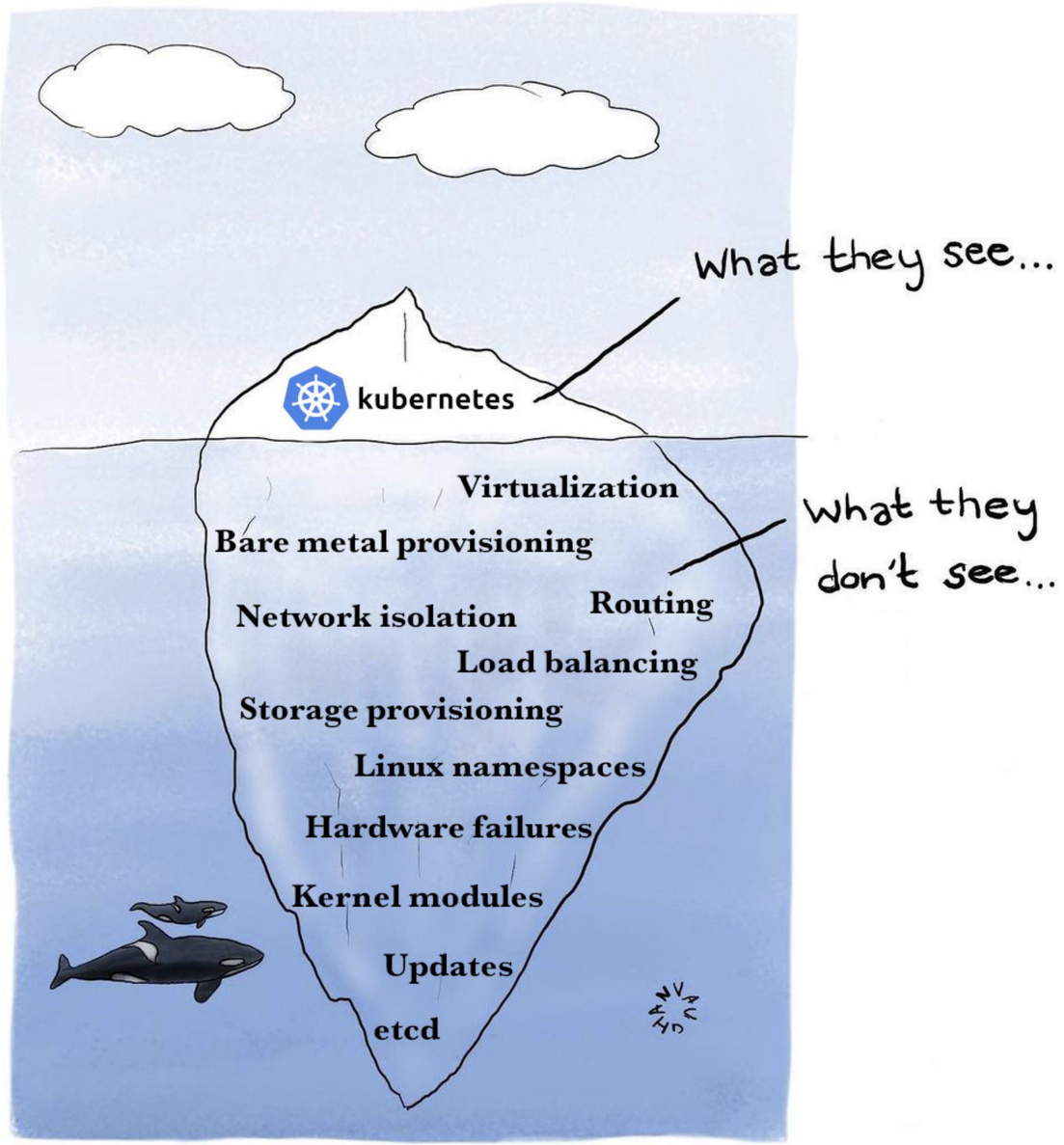
Пользовательские приложения

ПЛАТФОРМА

Kubernetes в облаке?

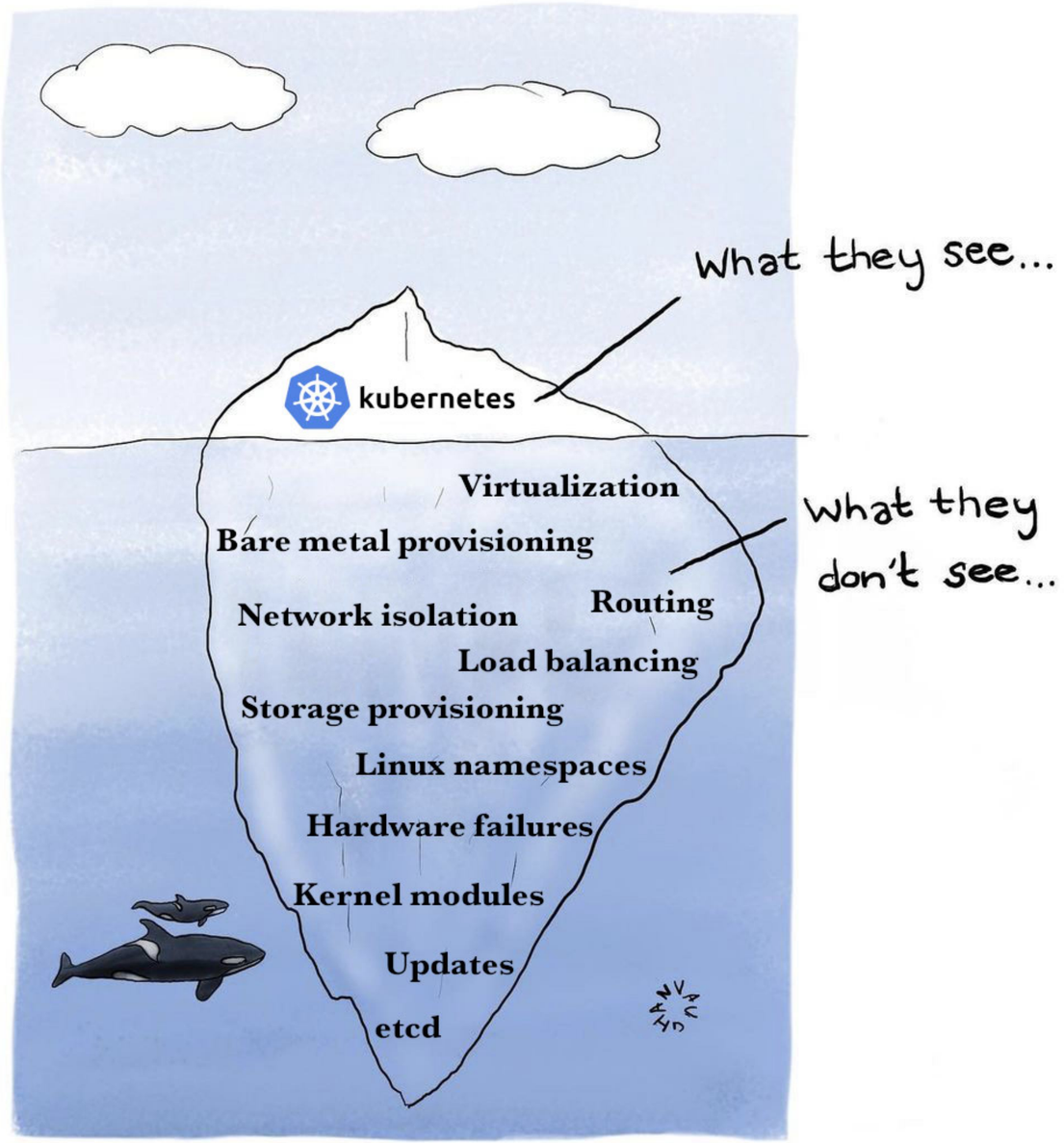

There is no cloud
It’s just someone else’s computer

Введение


What they see...
kubernetes
What they don't see...
Virtualization
Bare metal provisioning
Network isolation
Routing
Load balancing
Storage provisioning
Linux namespaces
Hardware failures
Kernel modules
Updates
etcd

Введение

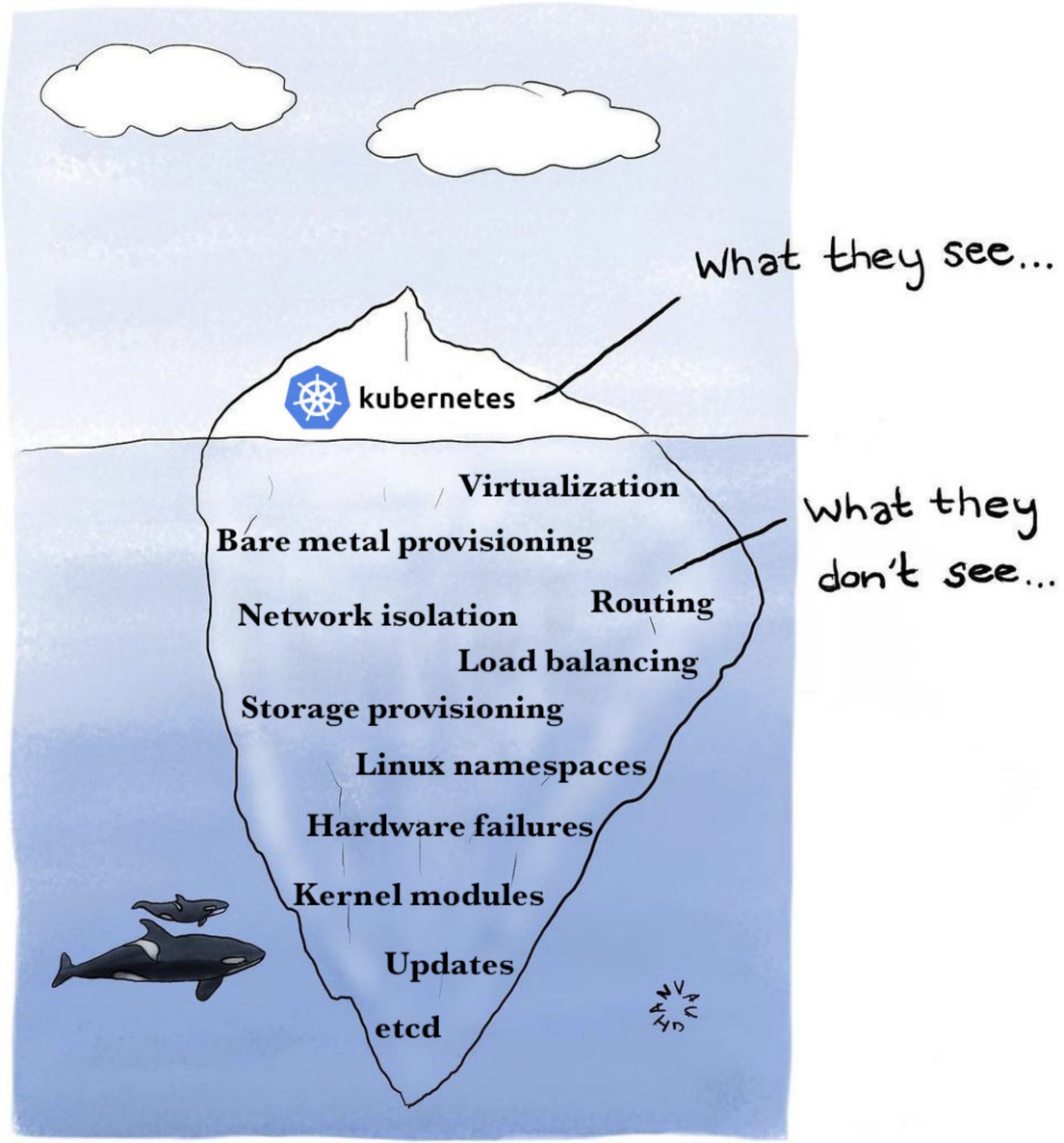
kubernetes



Введение

kubernetes

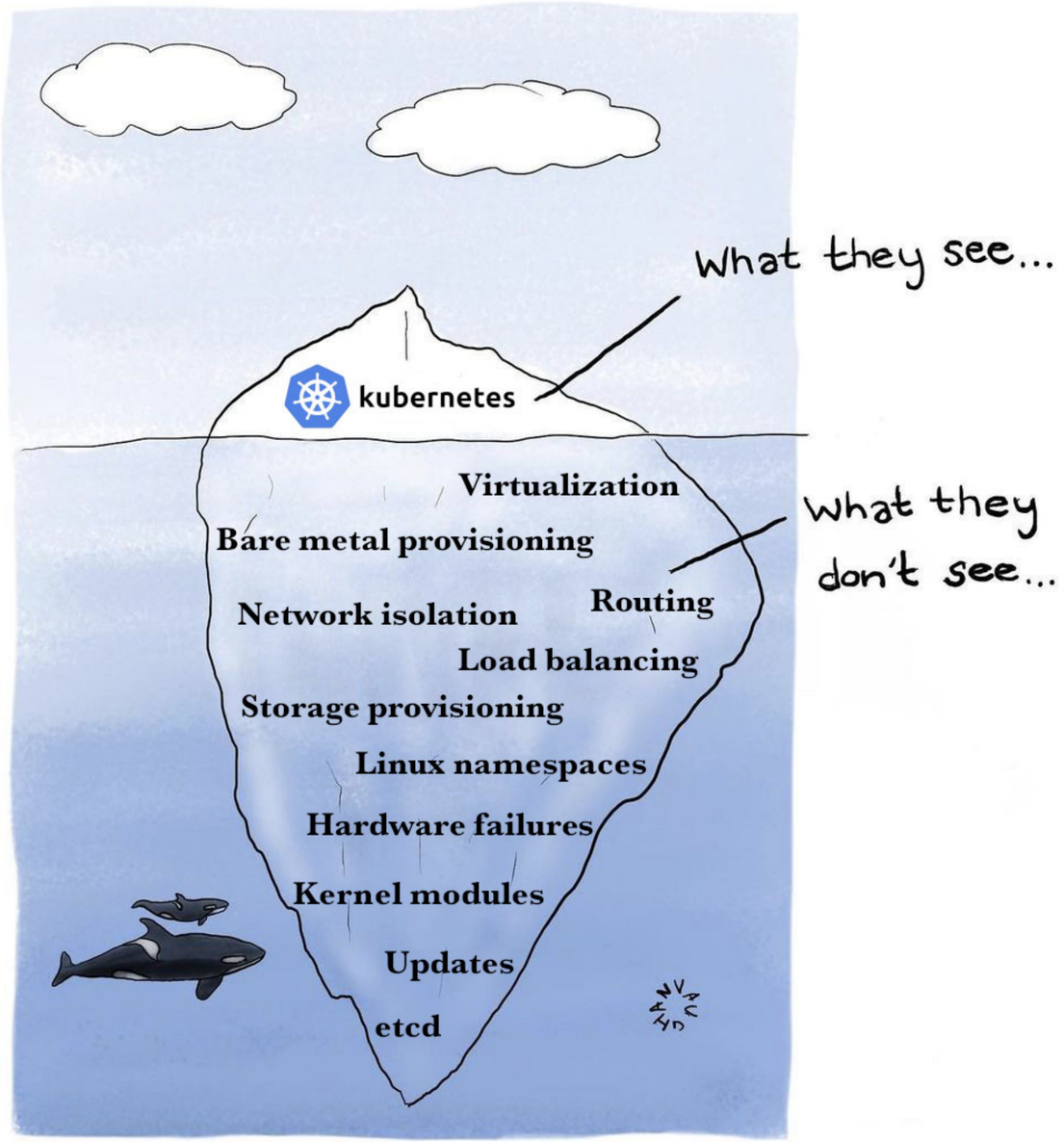
bare-metal



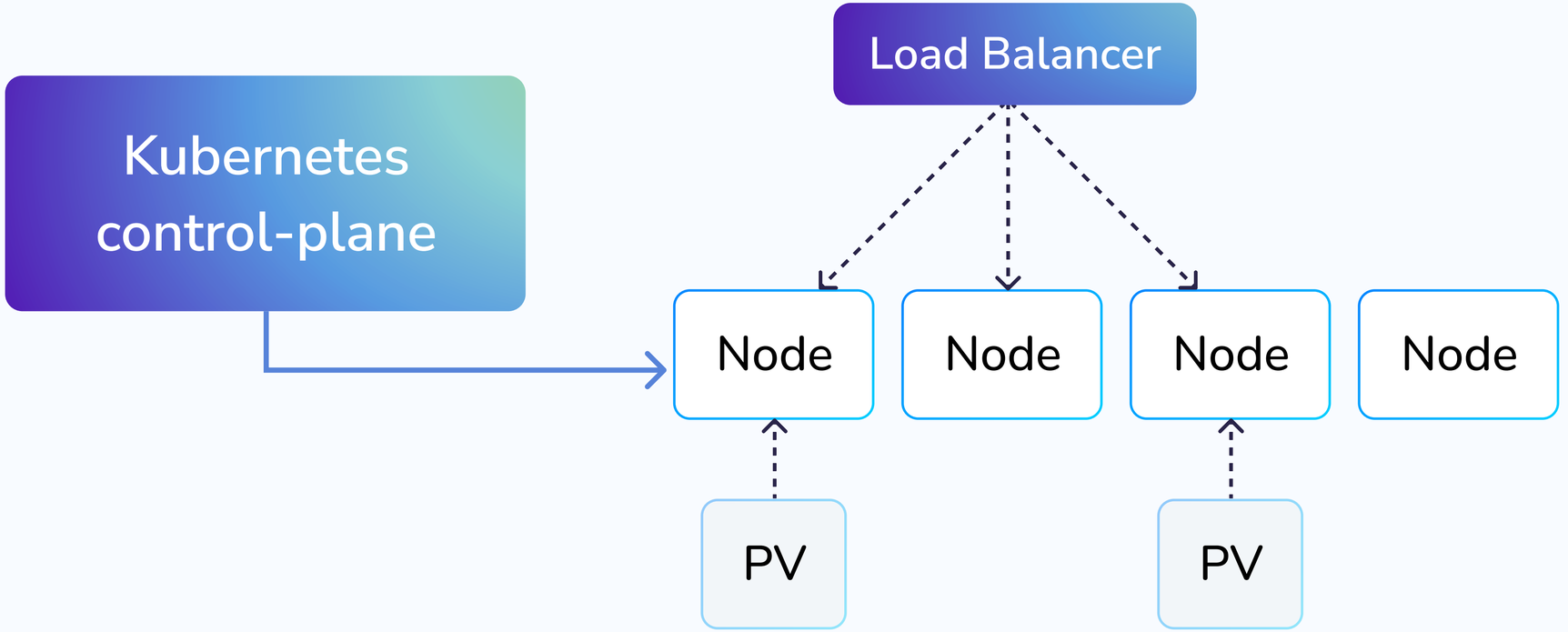
Введение

kubernetes

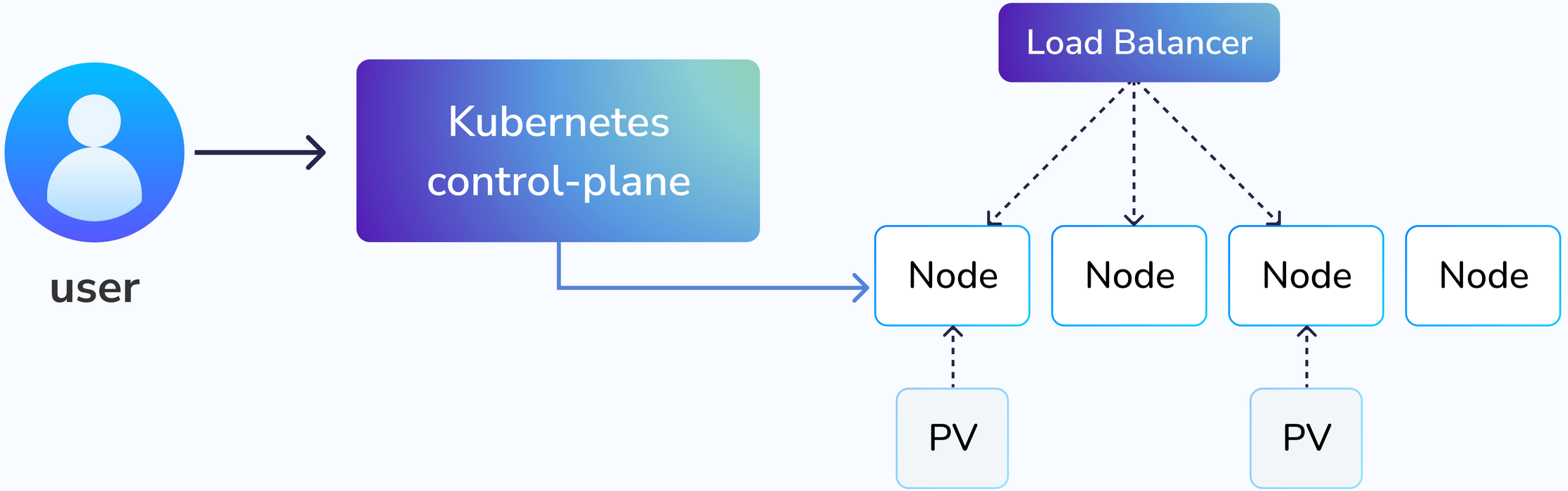
bare-metal



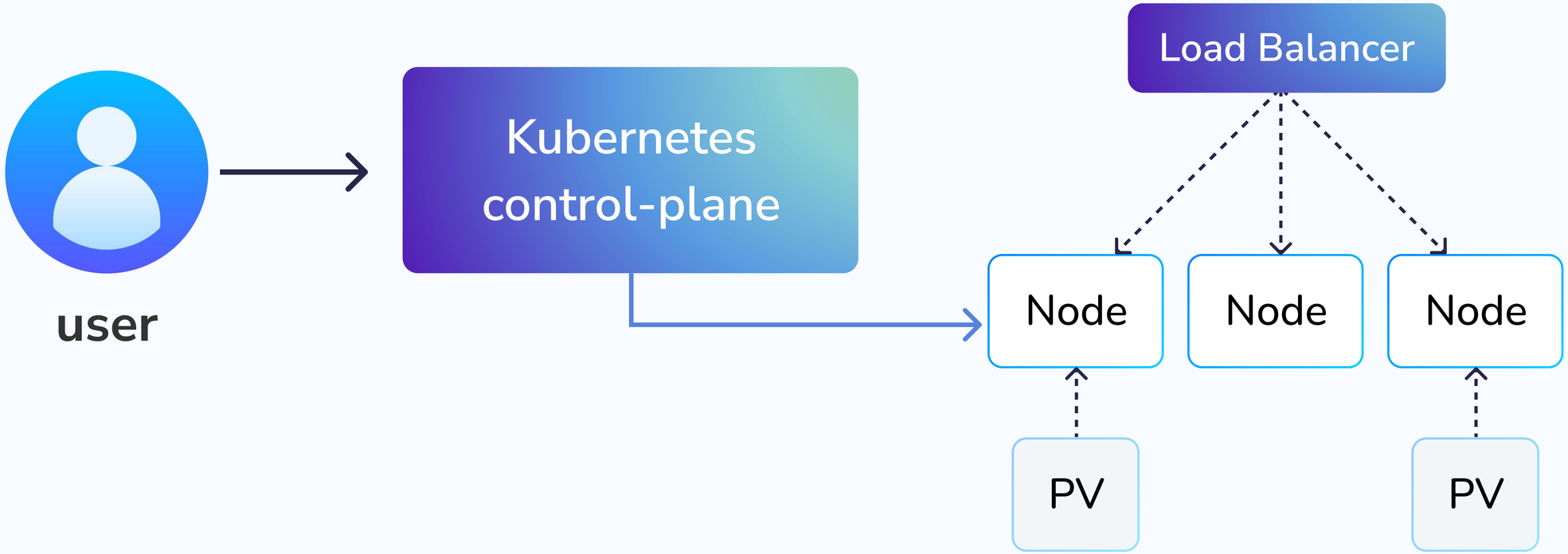
K8s в облаке


Load Balancer
Kubernetes control-plane
Node
Node
Node
Node
PV
PV

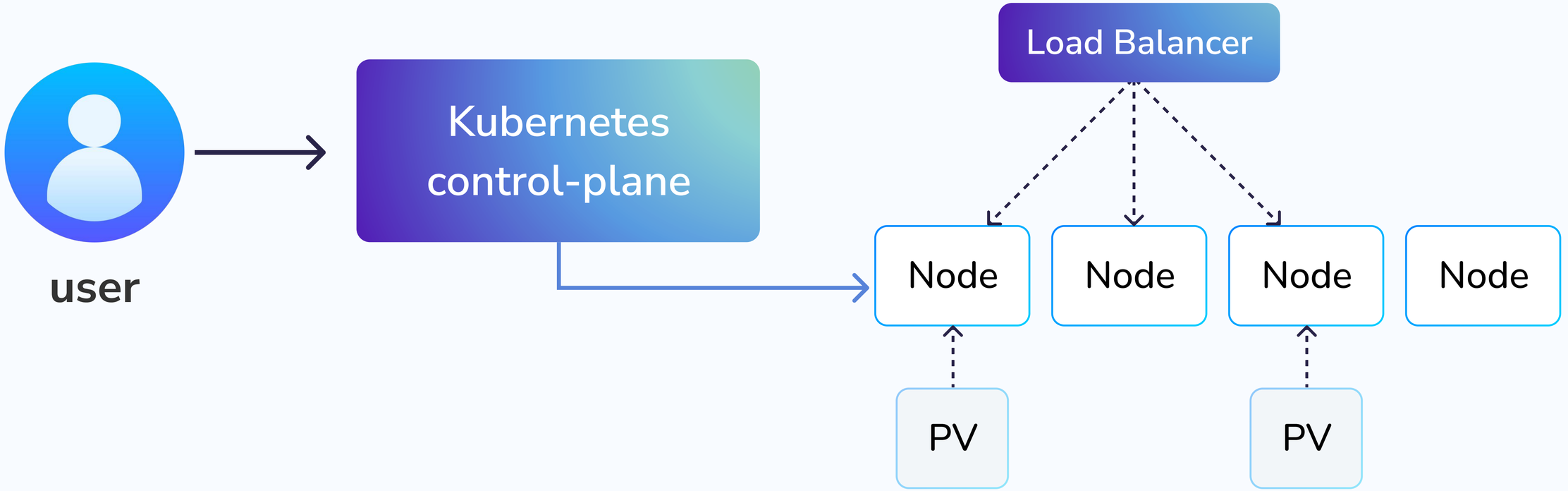
K8s в облаке


user
Kubernetes
control-plane
Load Balancer
Node
Node
Node
Node
PV
PV

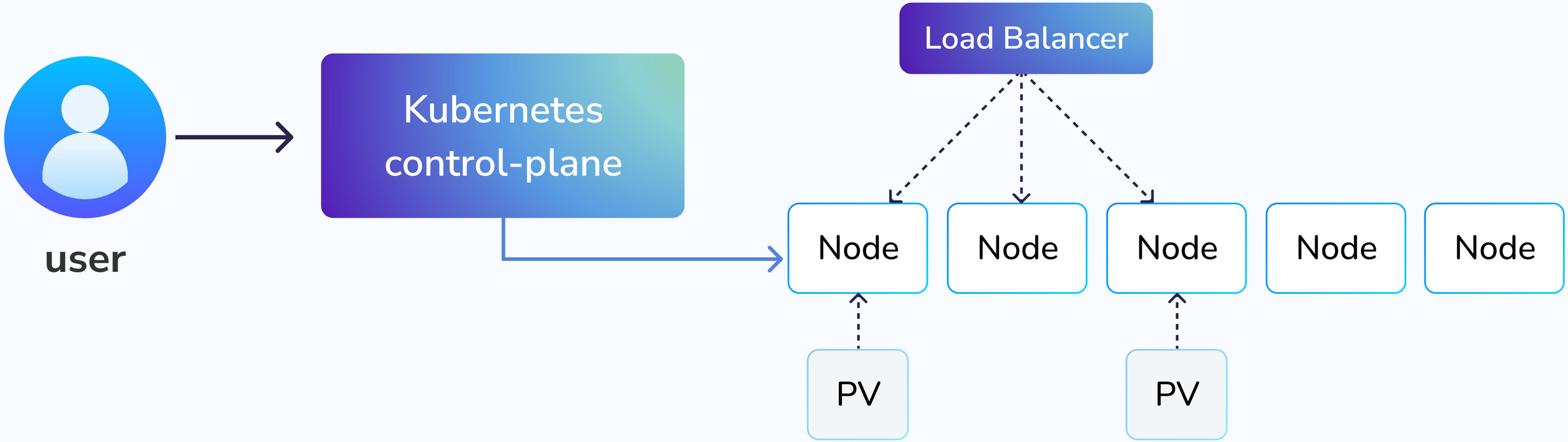
K8s в облаке


user
Kubernetes
control-plane
Load Balancer
Node
Node
Node
PV
PV

K8s в облаке


Load Balancer
Kubernetes
control-plane
user
Node
Node
Node
Node
PV
PV

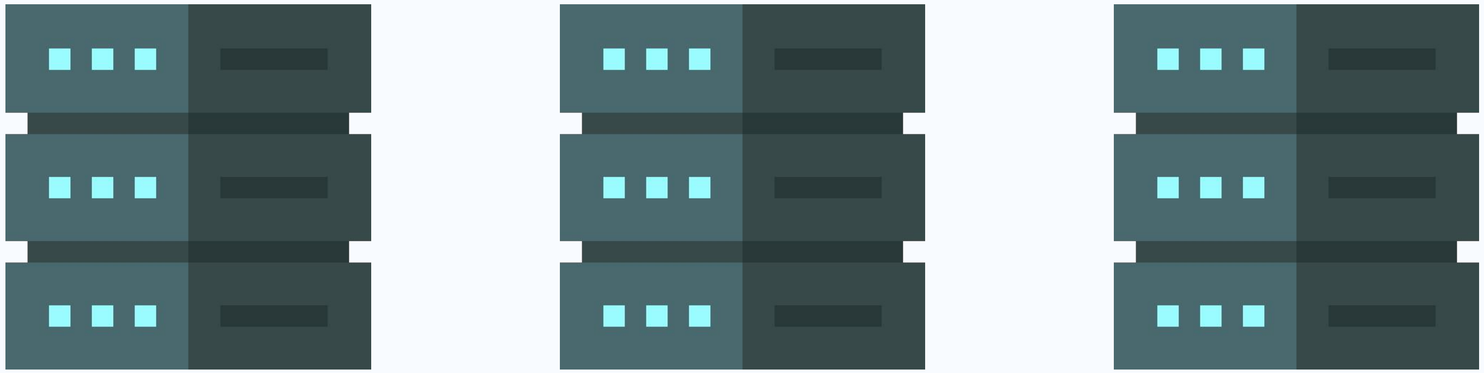
K8s в облаке

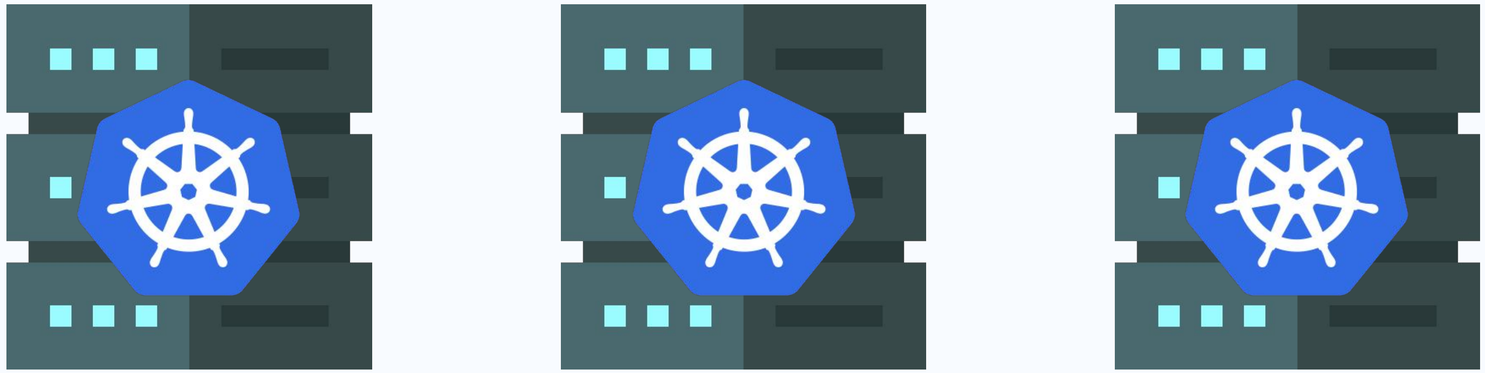

user
Kubernetes
control-plane
Load Balancer
Node
Node
Node
Node
Node
PV
PV

Как приготовить свое облако?

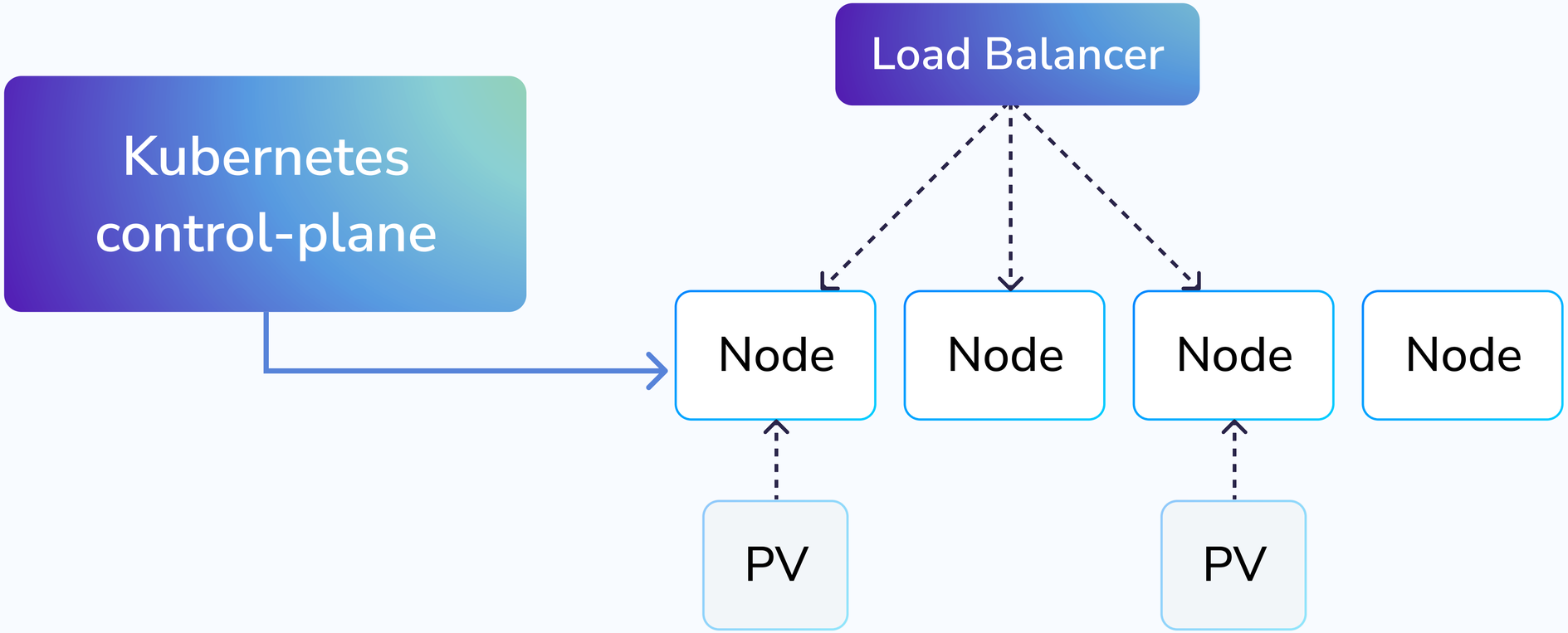
Вариант 1:

Kubernetes на bare-metal

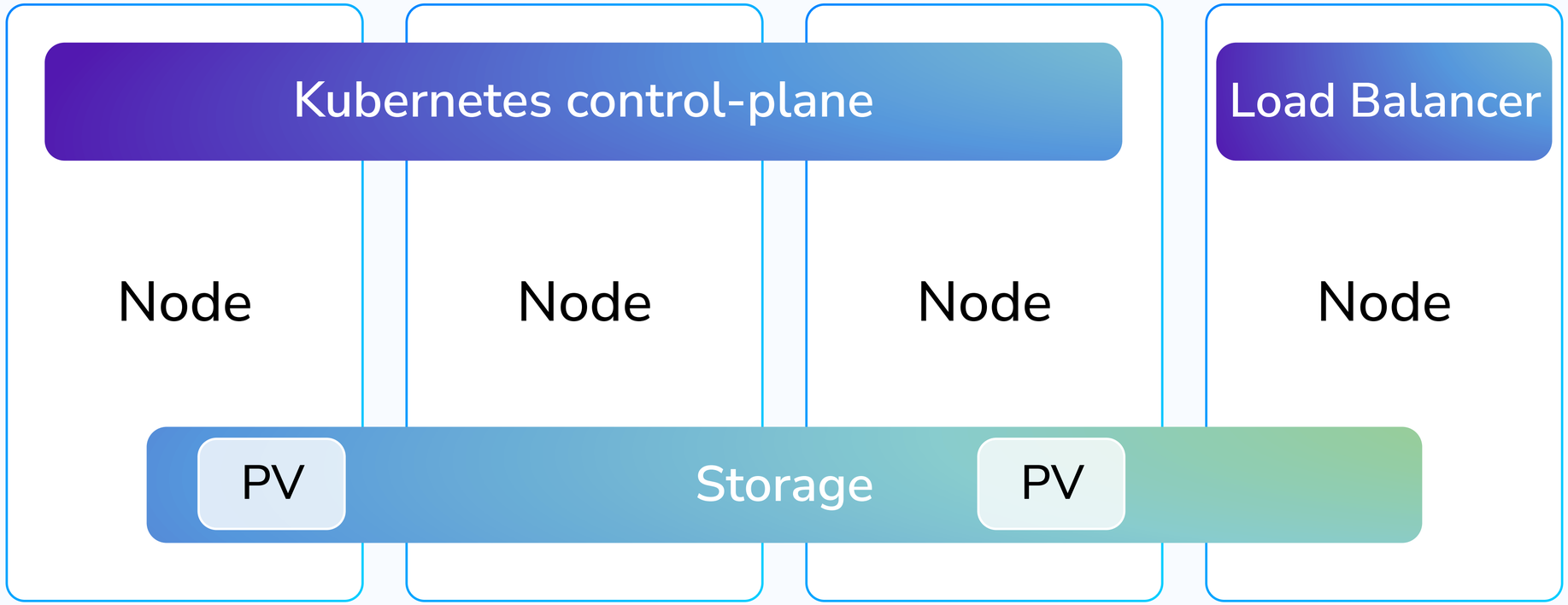




Ожидания:

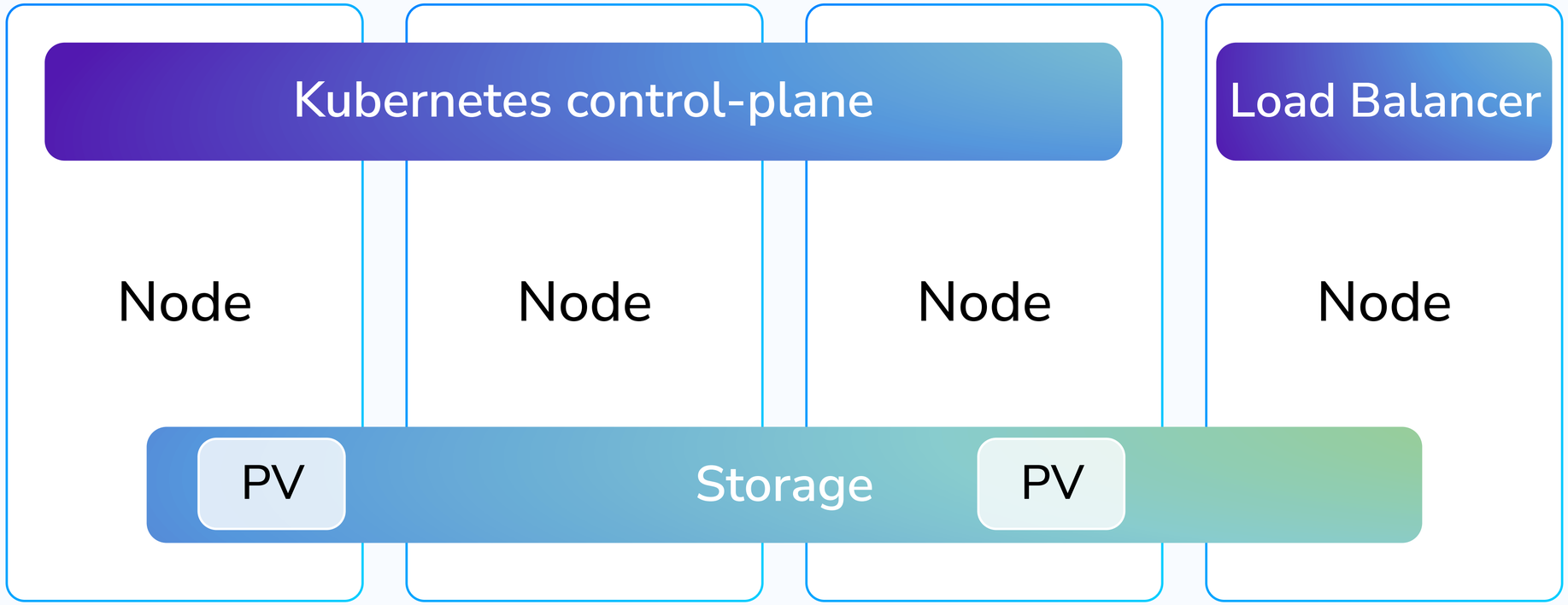

Load Balancer
Kubernetes control-plane
Node
Node
Node
Node
PV
PV

Действительность:



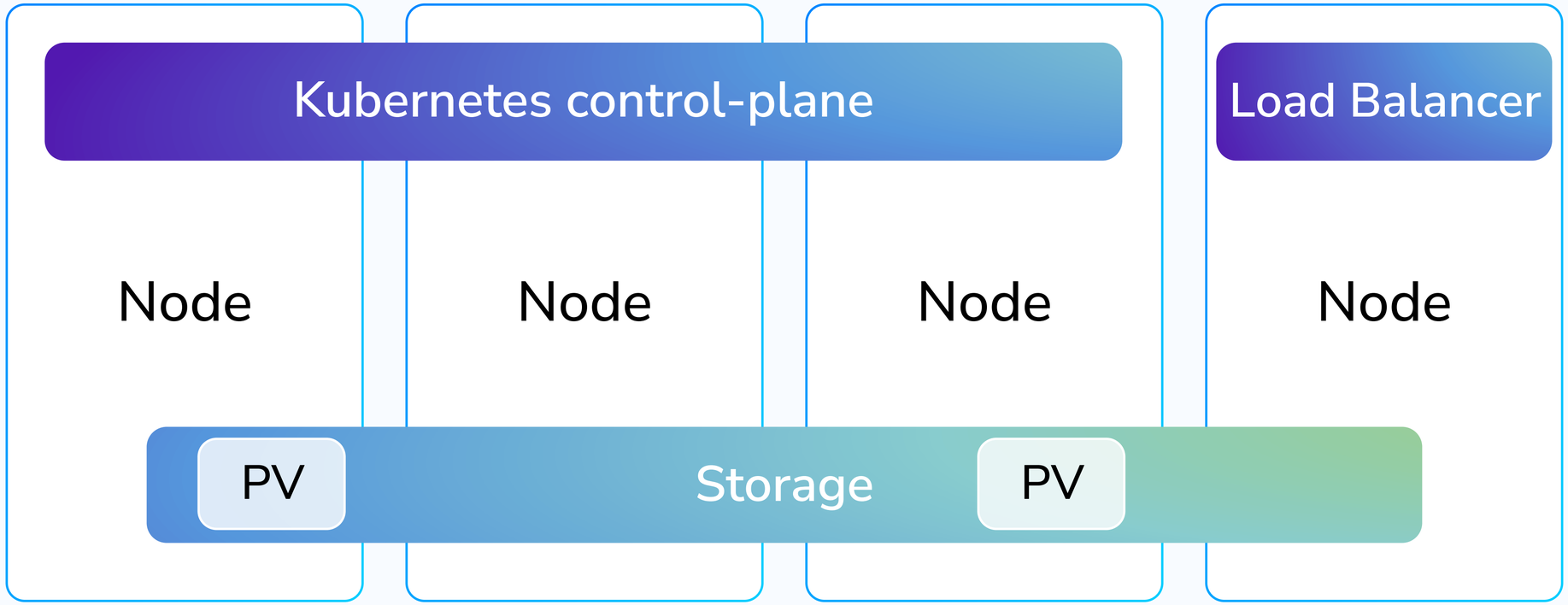
Действительность:

- удобное управление



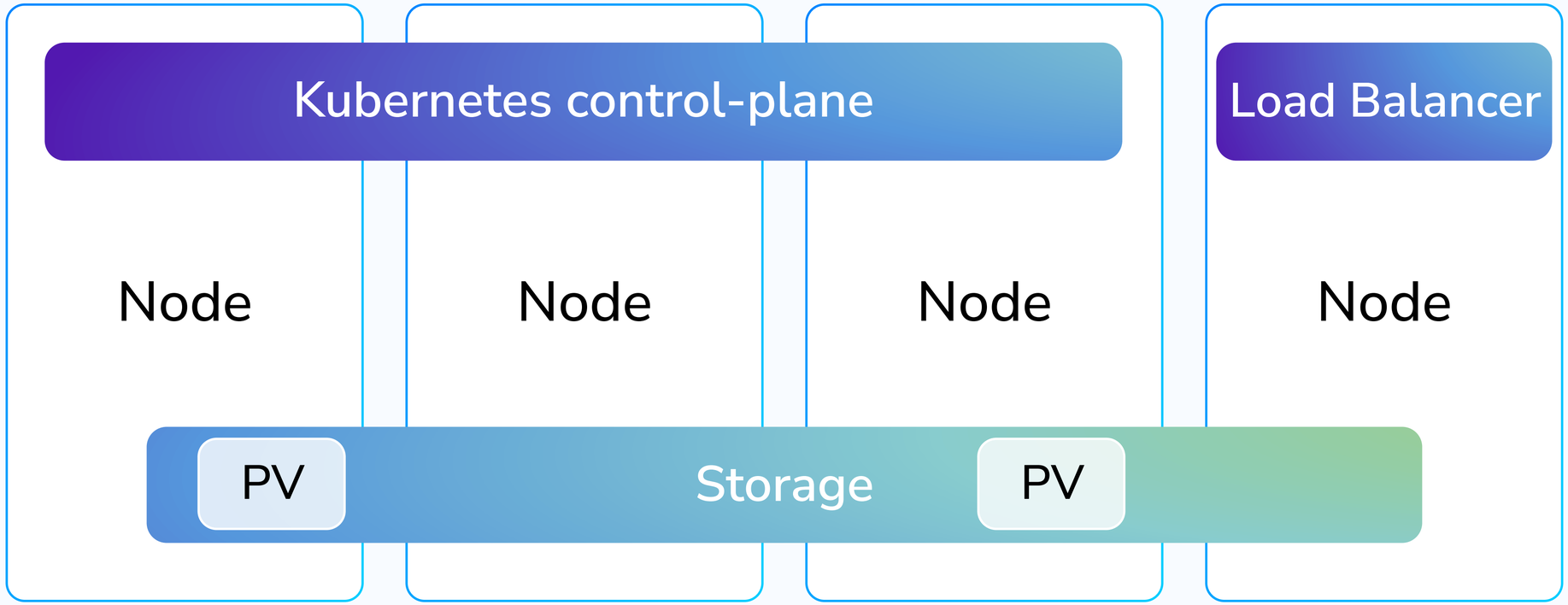
Действительность:

- ~~удобное управление~~
- безболезненные обновления



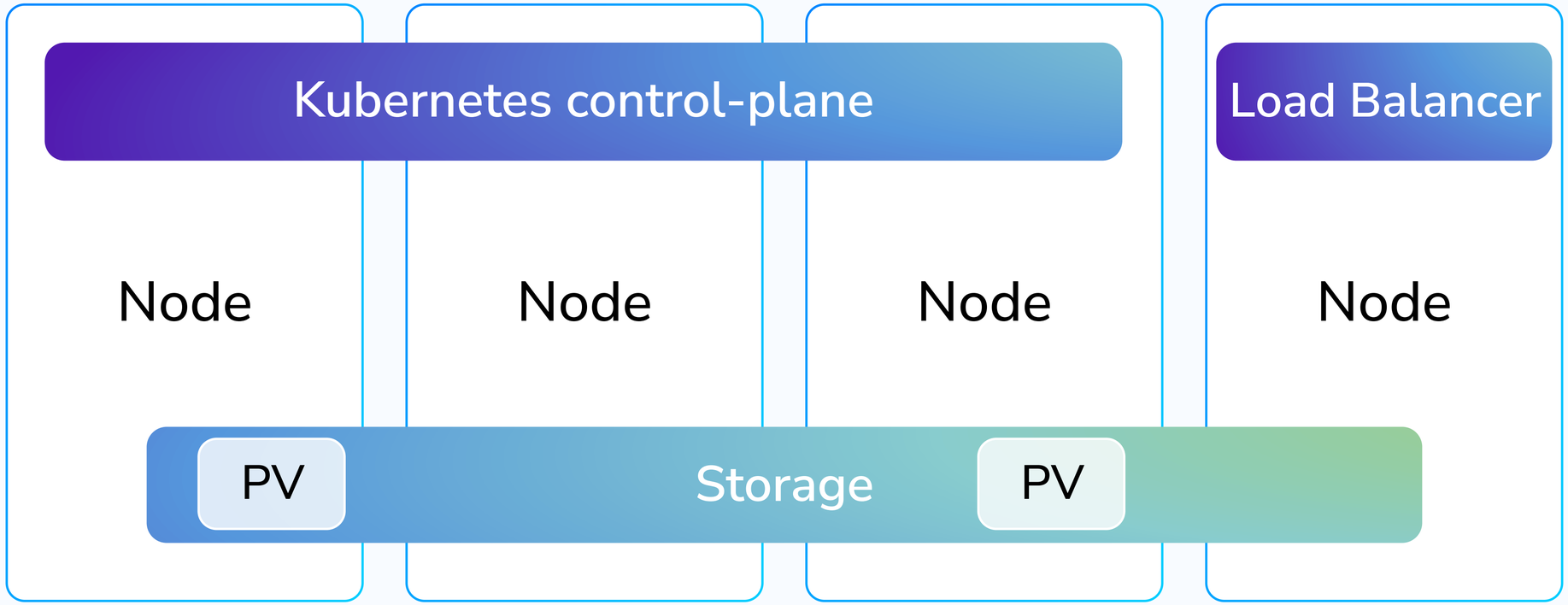
Действительность:

- ~~удобное управление~~
- ~~безболезненные обновления~~
- автомасштабирование



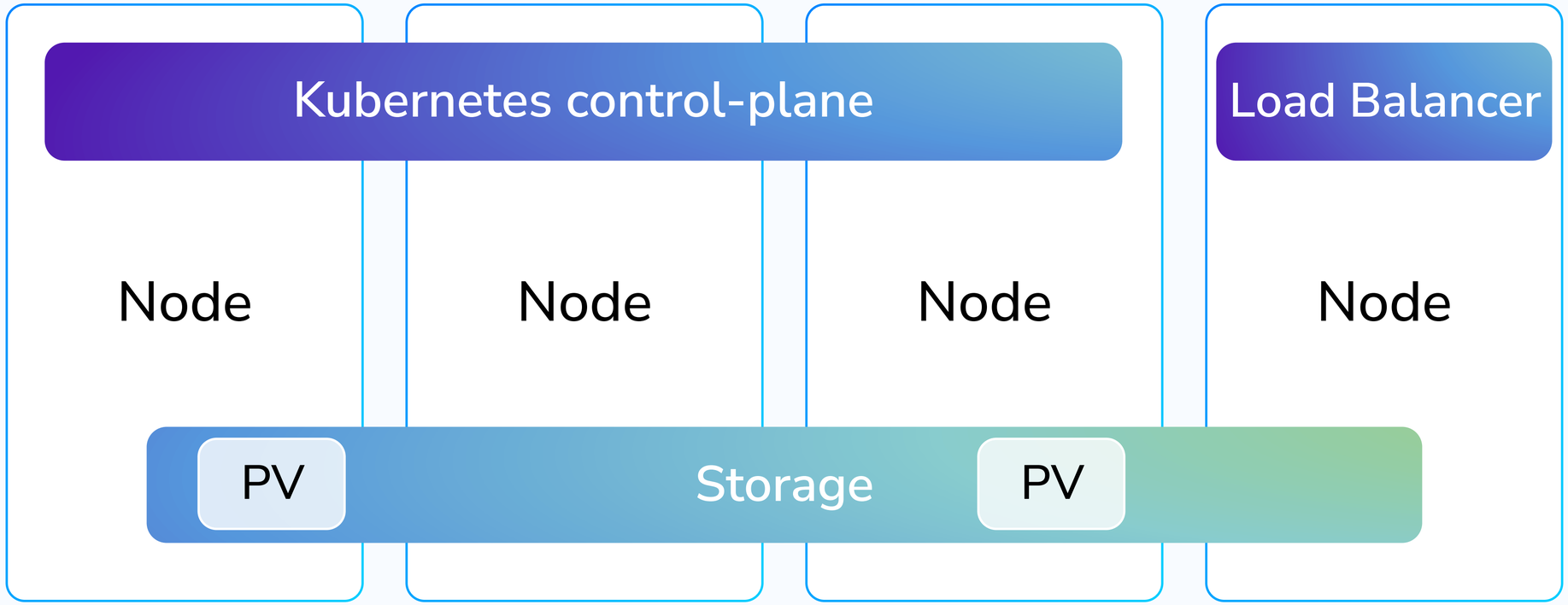
Действительность:

- ~~удобное управление~~
- ~~безболезненные обновления~~
- ~~автомасштабирование~~
- мульти-тенантность



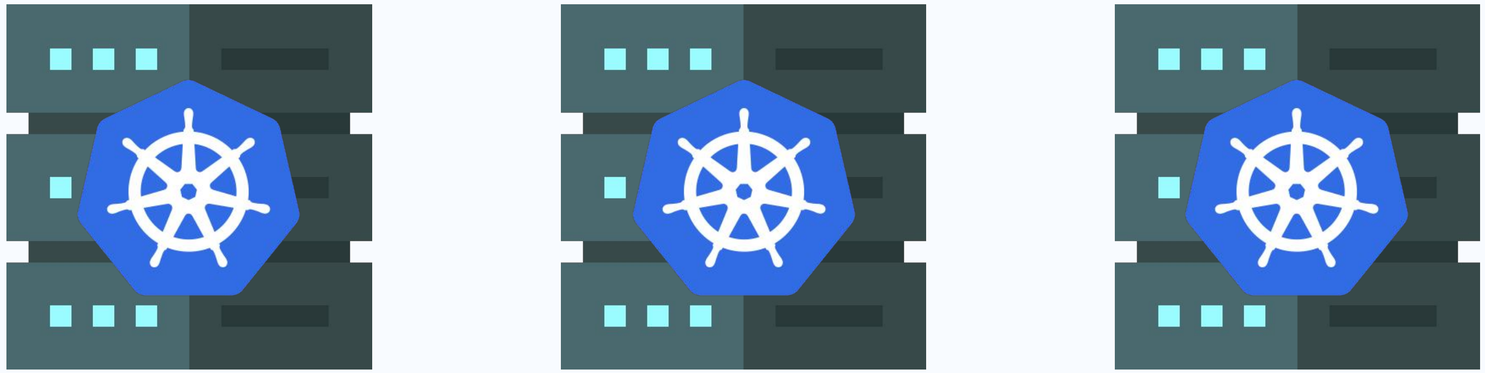
Действительность:

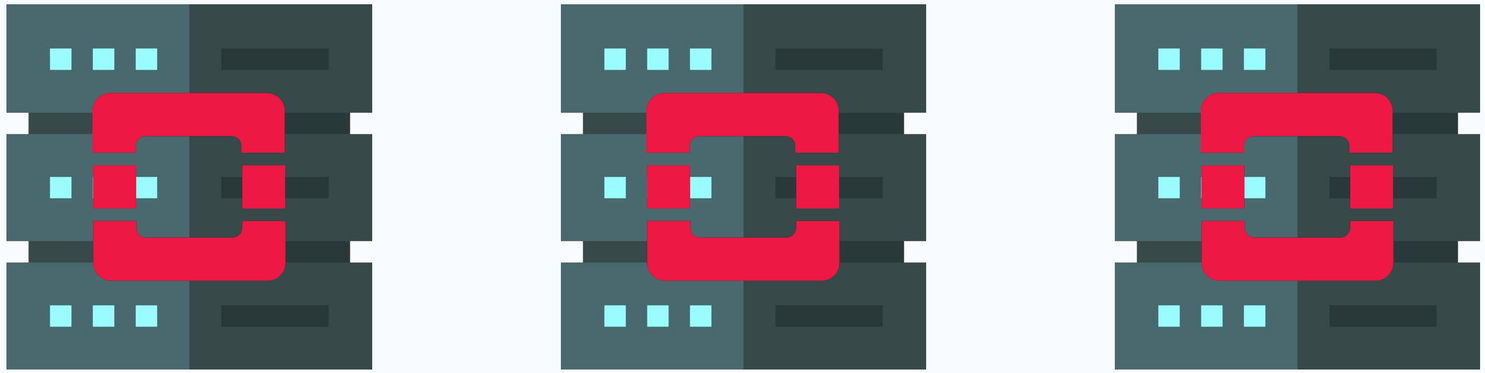
- ~~удобное управление~~
- ~~безболезненные обновления~~
- ~~автомасштабирование~~
- ~~мульти-тенантность~~

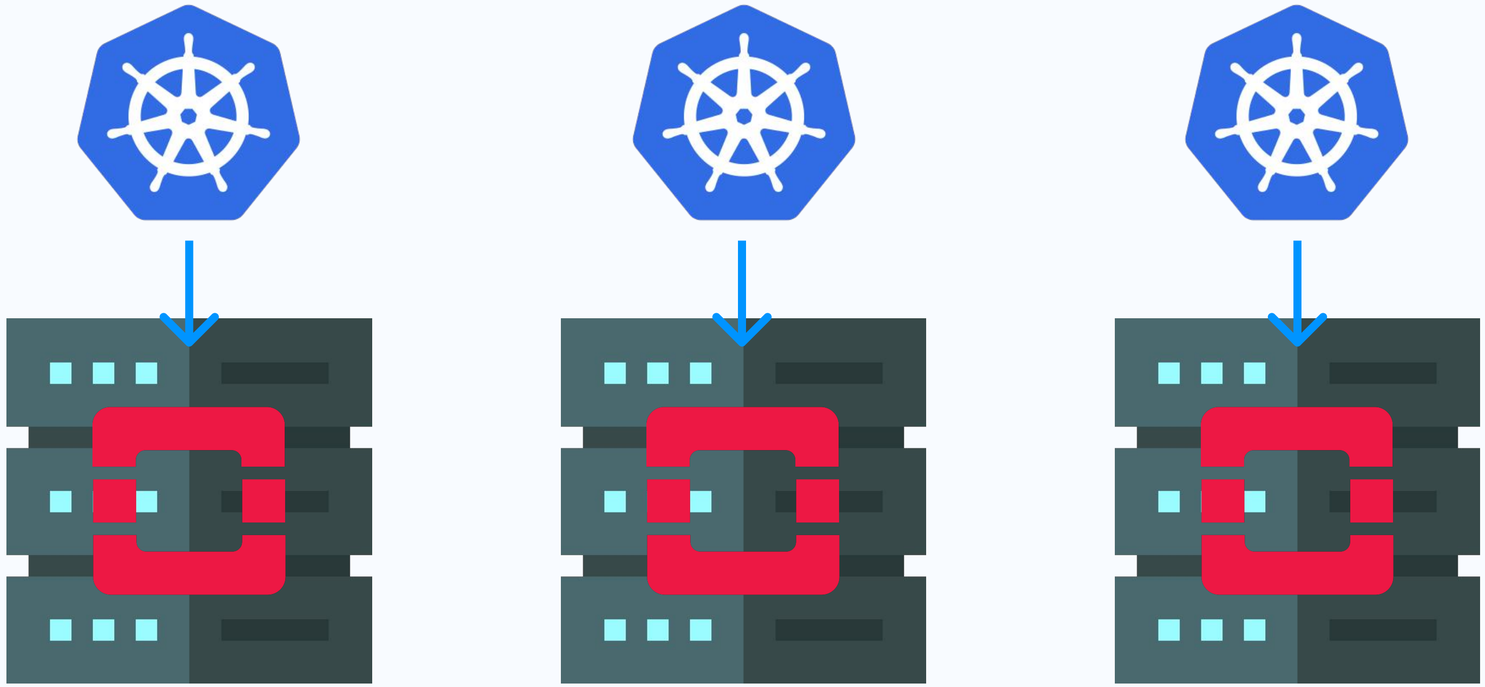


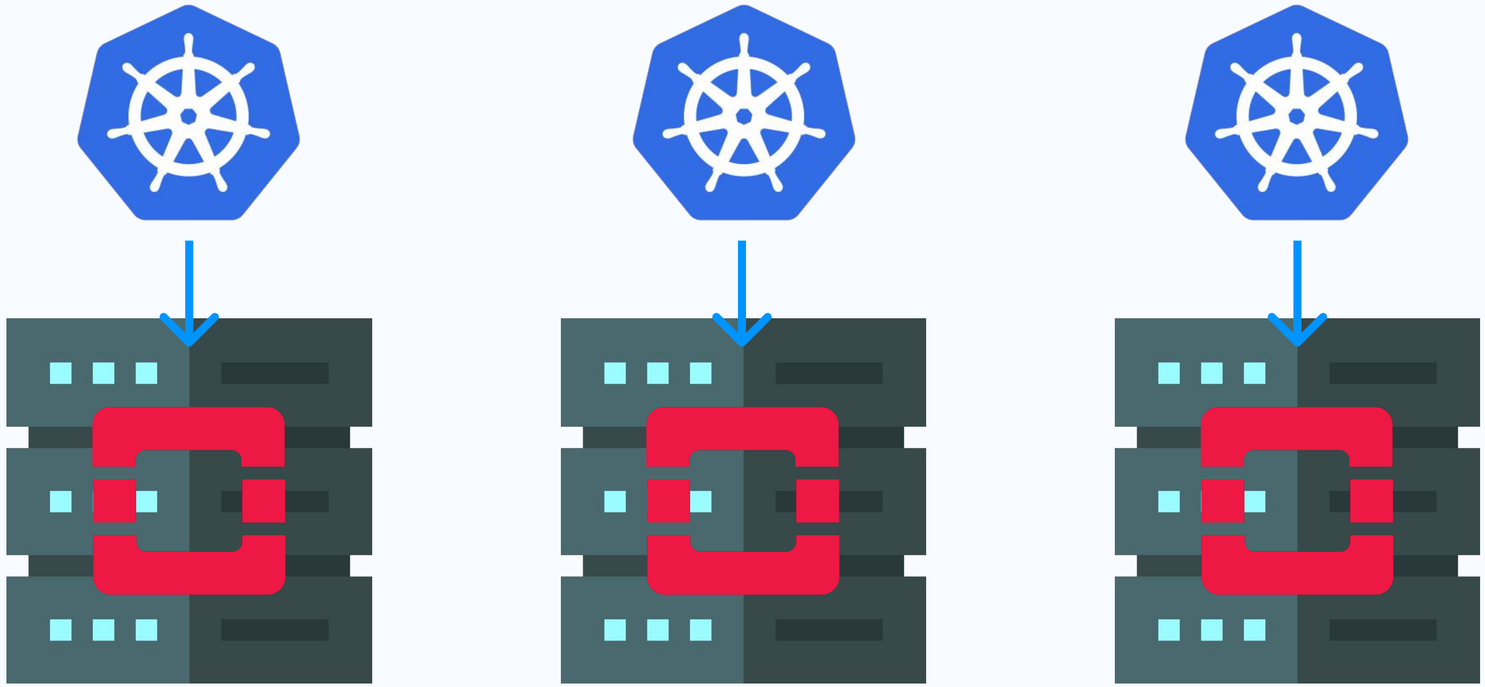
Вариант 2:

Kubernetes в частном облаке

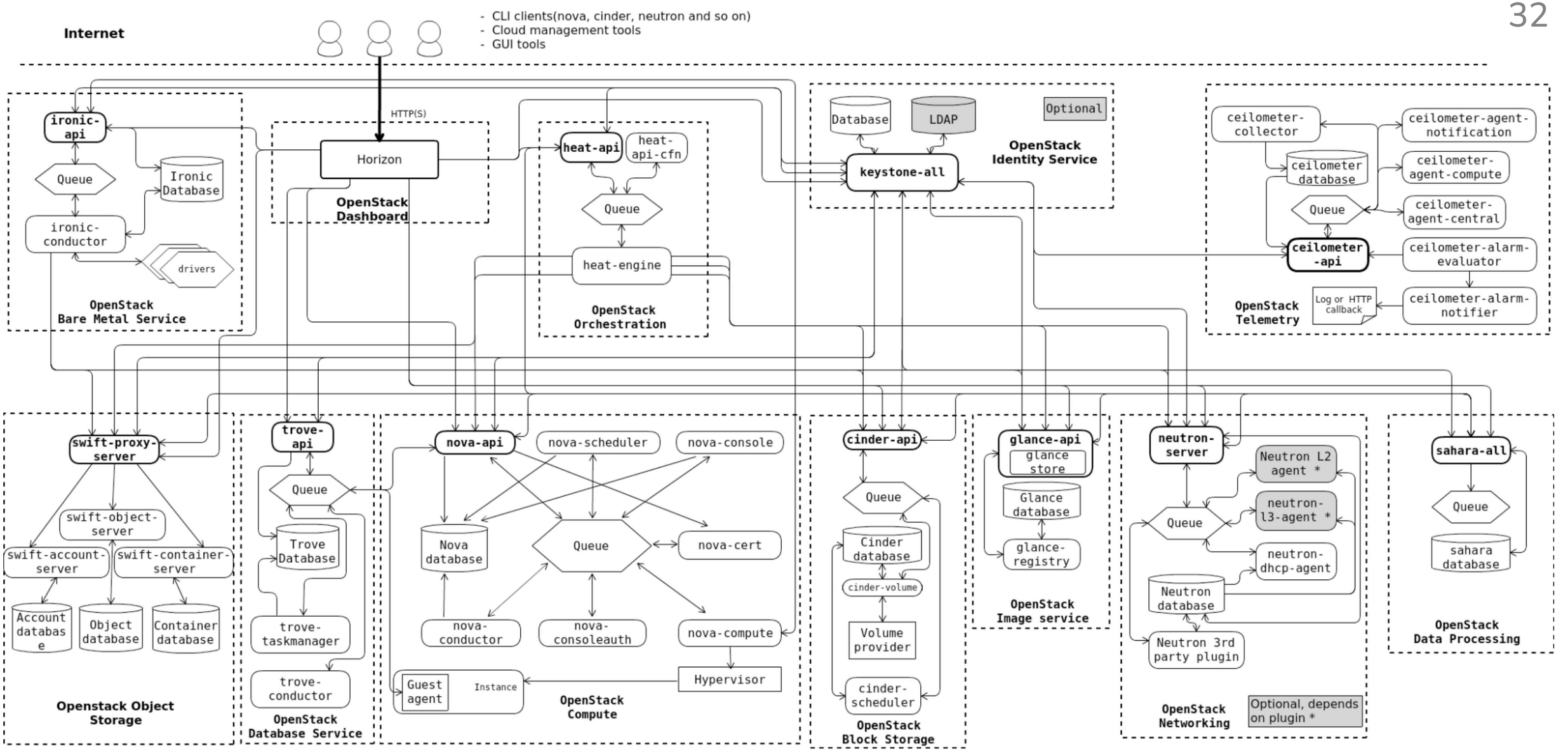






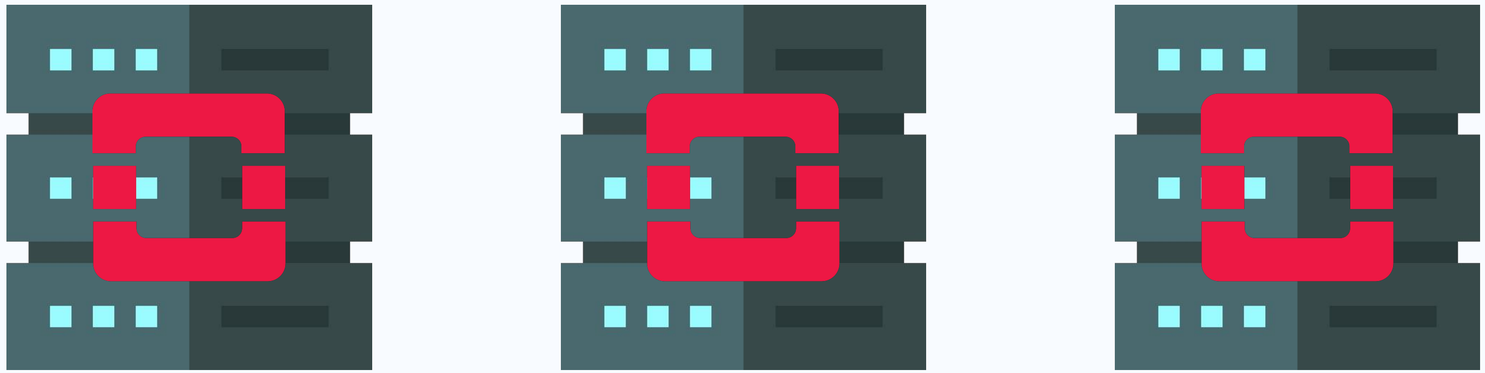


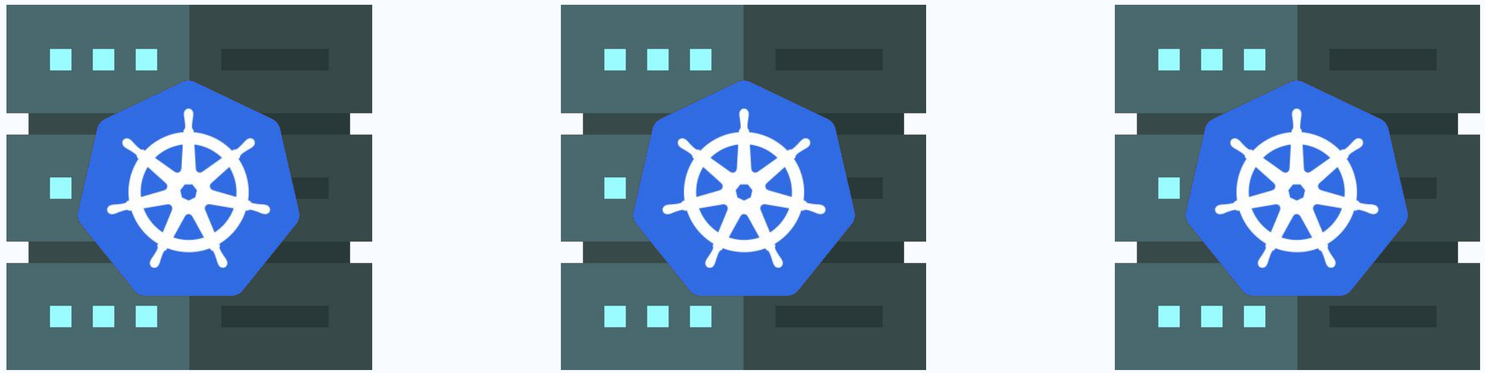
Кто отвечает за это ?

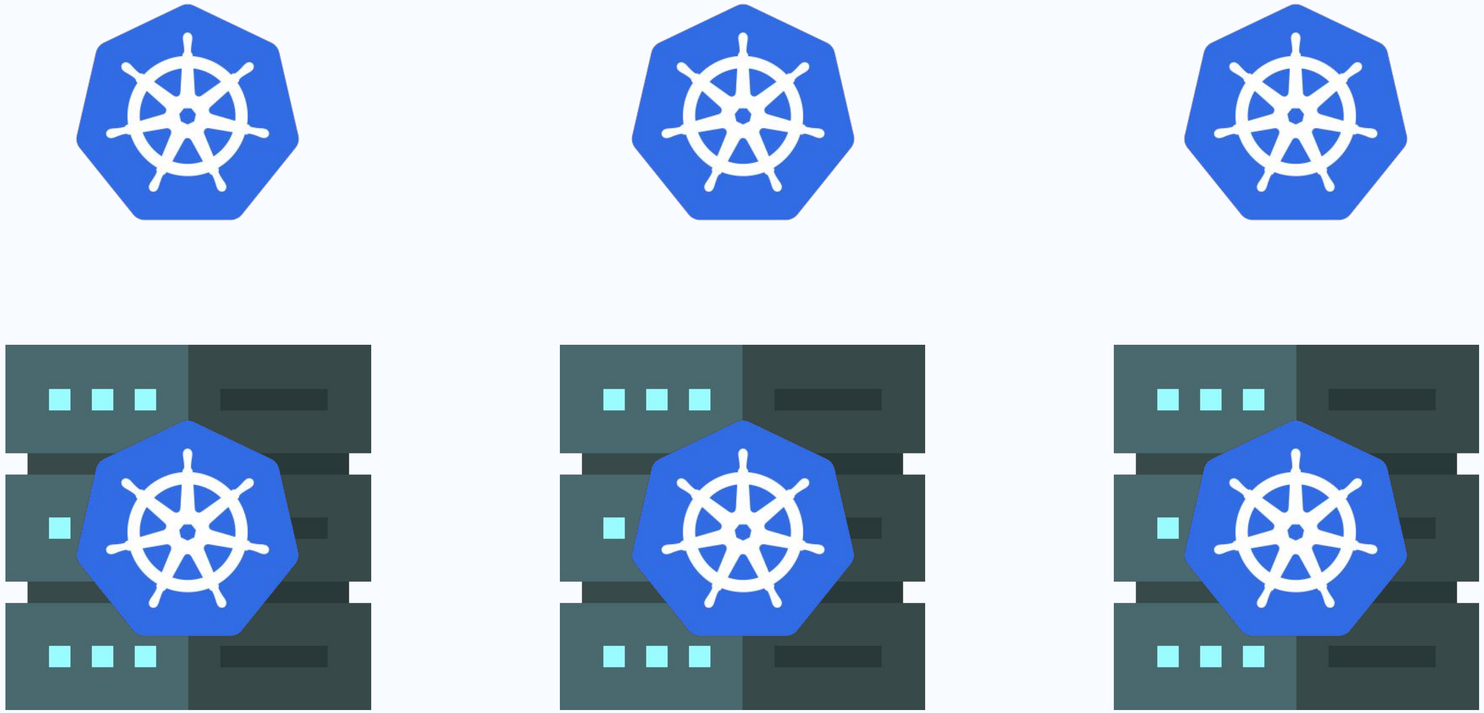

Internet
- CLI clients(nova, cinder, neutron and so on)
- Cloud management tools
- GUI tools
HTTP(S)
ironic-api
Queue
Ironic Database
ironic-conductor
drivers
OpenStack Bare Metal Service
Horizon
OpenStack Dashboard
heat-api
heat-api-cfn
Queue
heat-engine
OpenStack Orchestration
Database
LDAP
Optional
keystone-all
OpenStack Identity Service
ceilometer-collector
ceilometer-agent-notification
ceilometer database
ceilometer-agent-compute
Queue
ceilometer-agent-central
ceilometer-api
ceilometer-alarm-evaluator
Log or HTTP callback
ceilometer-alarm-notifier
OpenStack Telemetry
swift-proxy-server
swift-object-server
swift-account-server
swift-container-server
Account database
Object database
Container database
Openstack Object Storage
trove-api
Queue
Trove Database
trove-taskmanager
trove-conductor
OpenStack Database Service
nova-api
nova-scheduler
nova-console
Nova database
Queue
nova-cert
nova-conductor
nova-consoleauth
nova-compute
Hypervisor
Guest agent
Instance
OpenStack Compute
cinder-api
Queue
Cinder database
cinder-volume
Volume provider
cinder-scheduler
OpenStack Block Storage
glance-api
glance store
Glance database
glance-registry
OpenStack Image service
neutron-server
Neutron L2 agent *
Queue
neutron-l3-agent *
neutron-dhcp-agent
Neutron database
Neutron 3rd party plugin
OpenStack Networking
Optional, depends on plugin *
sahara-all
Queue
sahara database
OpenStack Data Processing

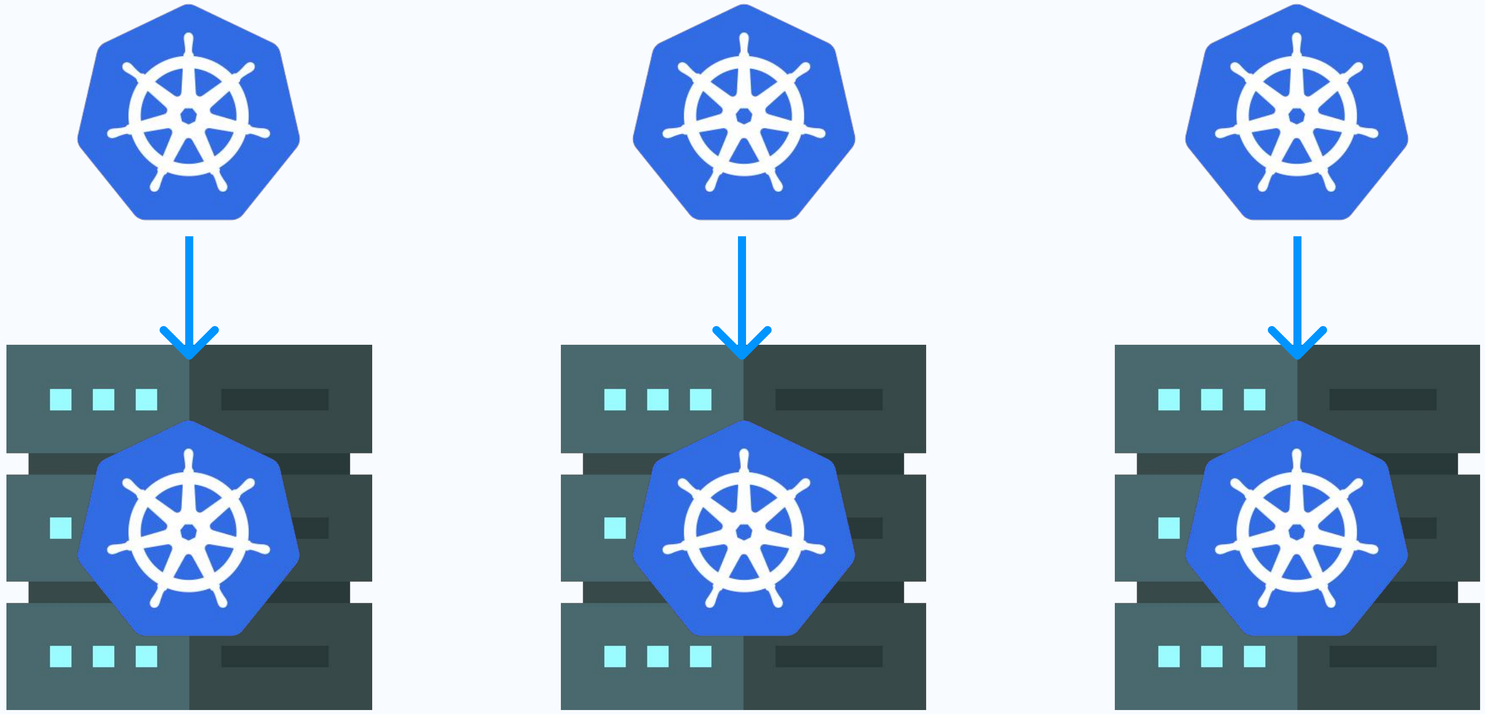
Вариант 3:

Матрешка - Kubernetes in Kubernetes!

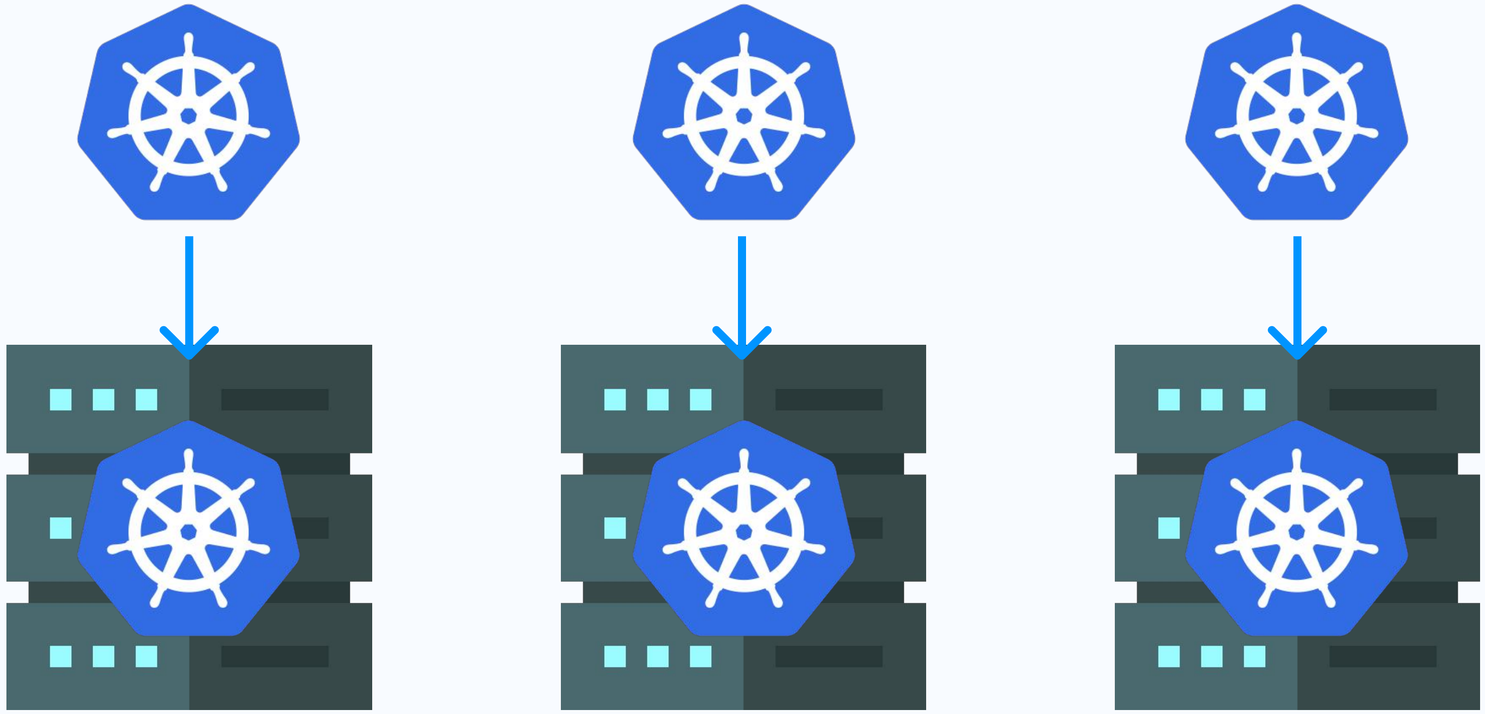








Управляющий кластер

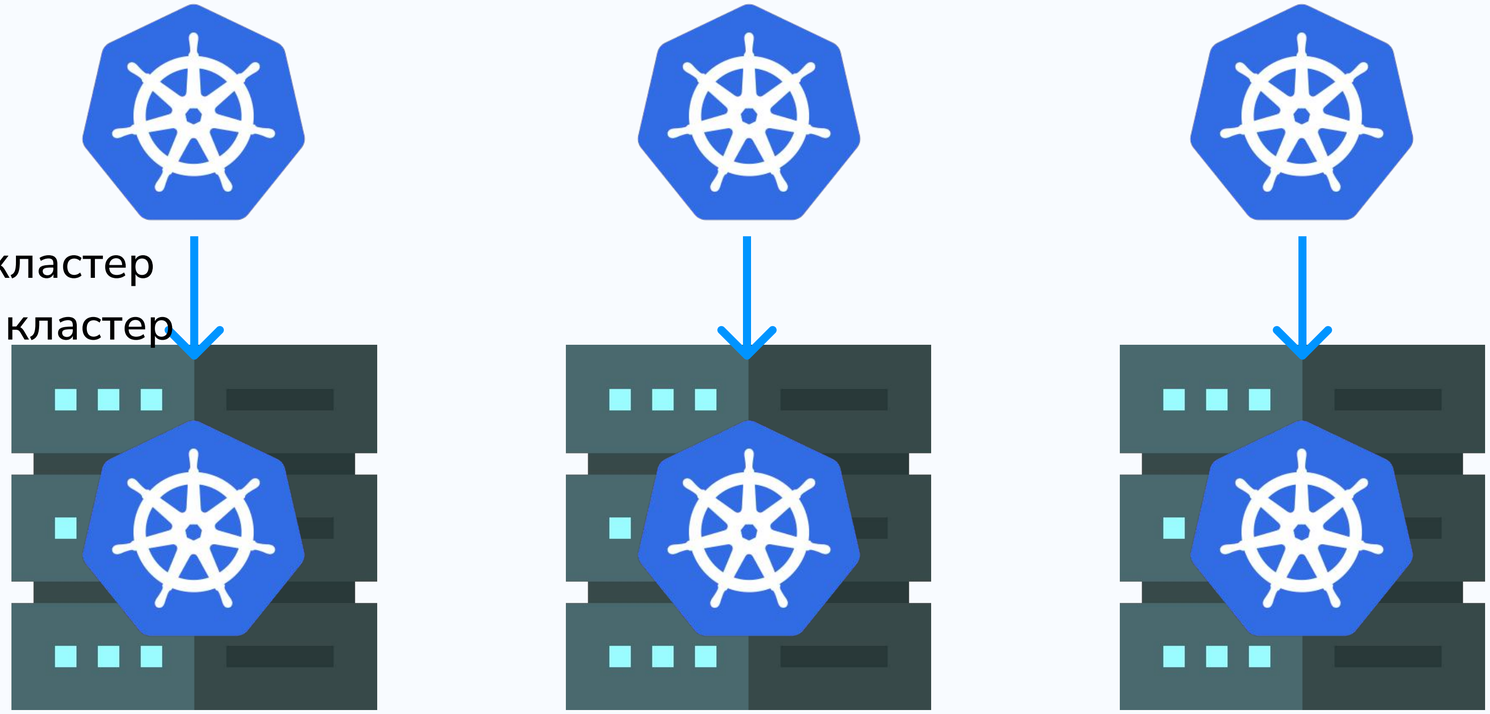


Тенант кластер

Еще можно называть:

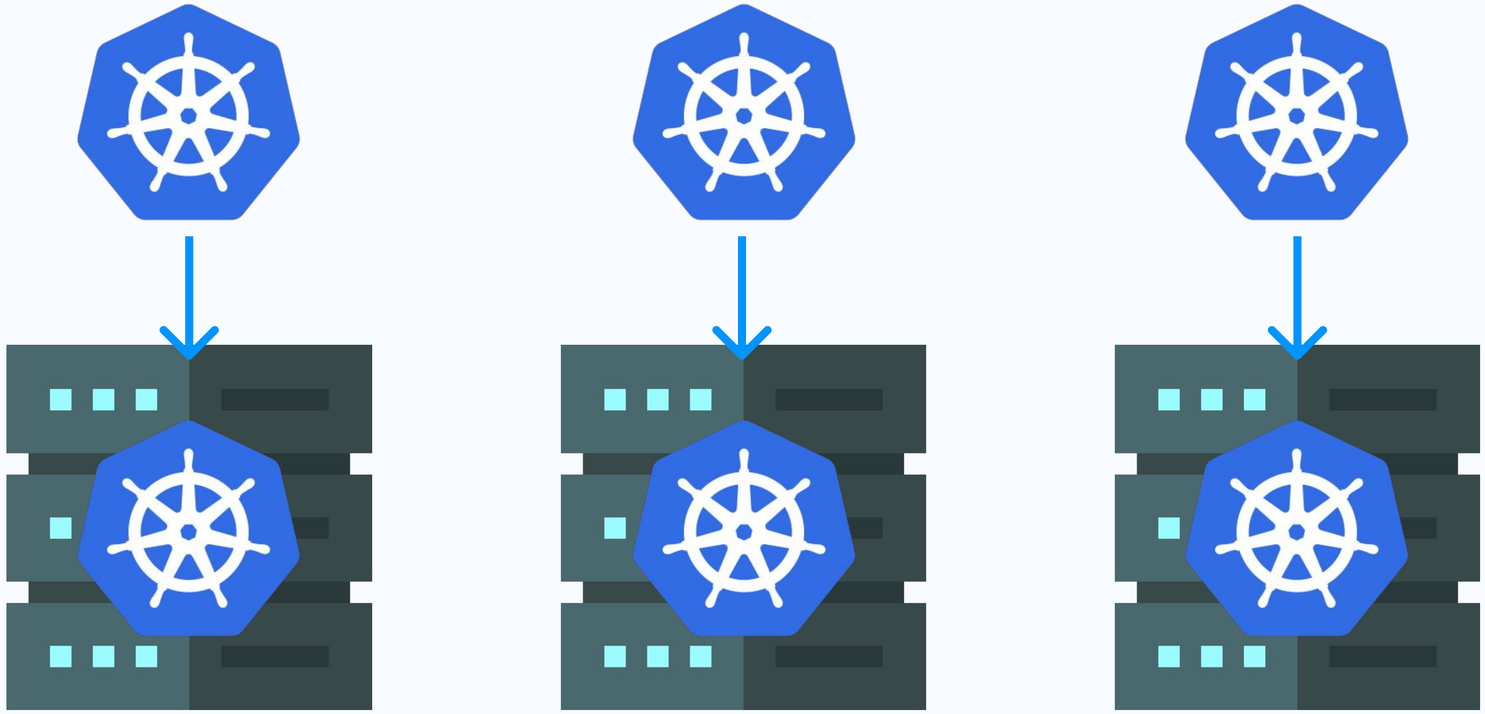
- пользовательский кластер
- листовой (узловой) кластер

Управляющий кластер



Тенант кластер

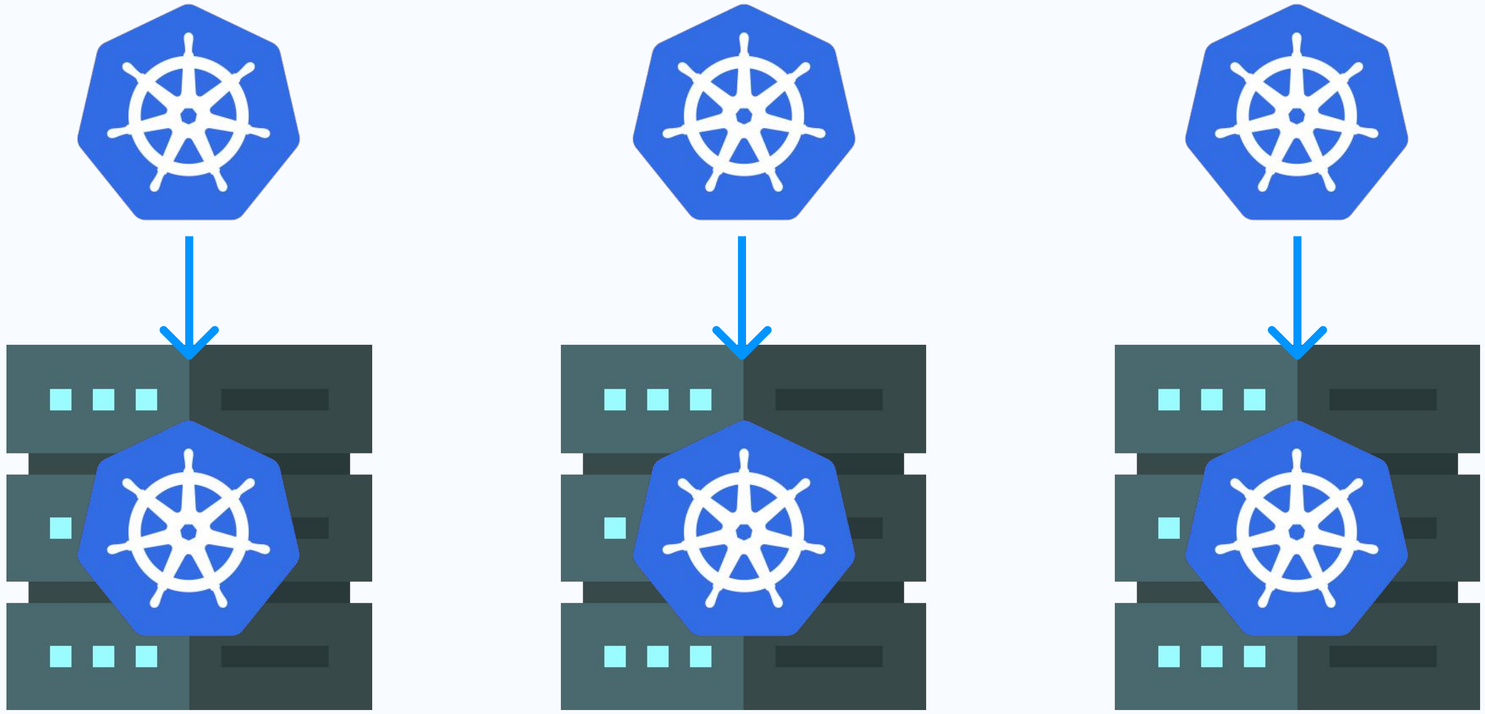
Управляющий кластер



- **единое окружение**

Тенант кластер

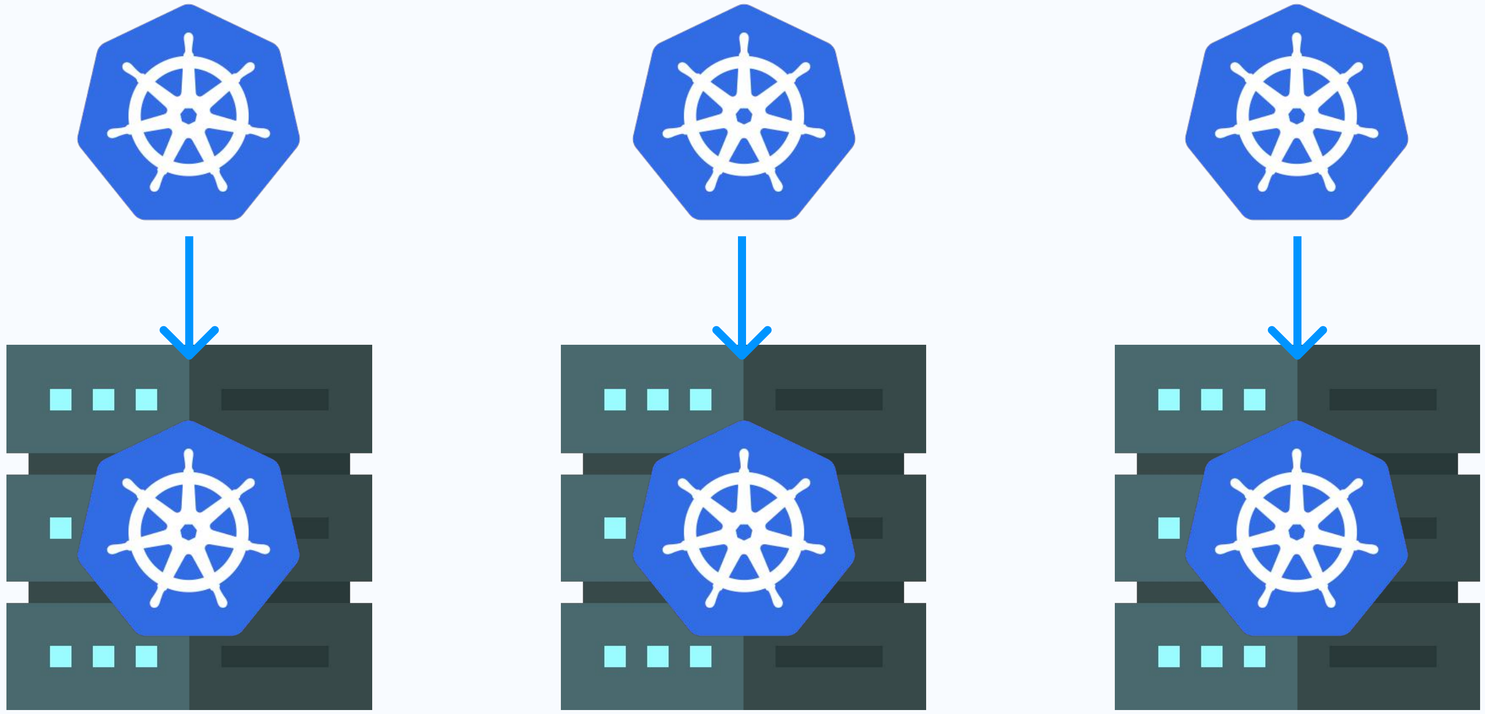
Управляющий кластер



- **единое окружение**
- **единый способ управления**

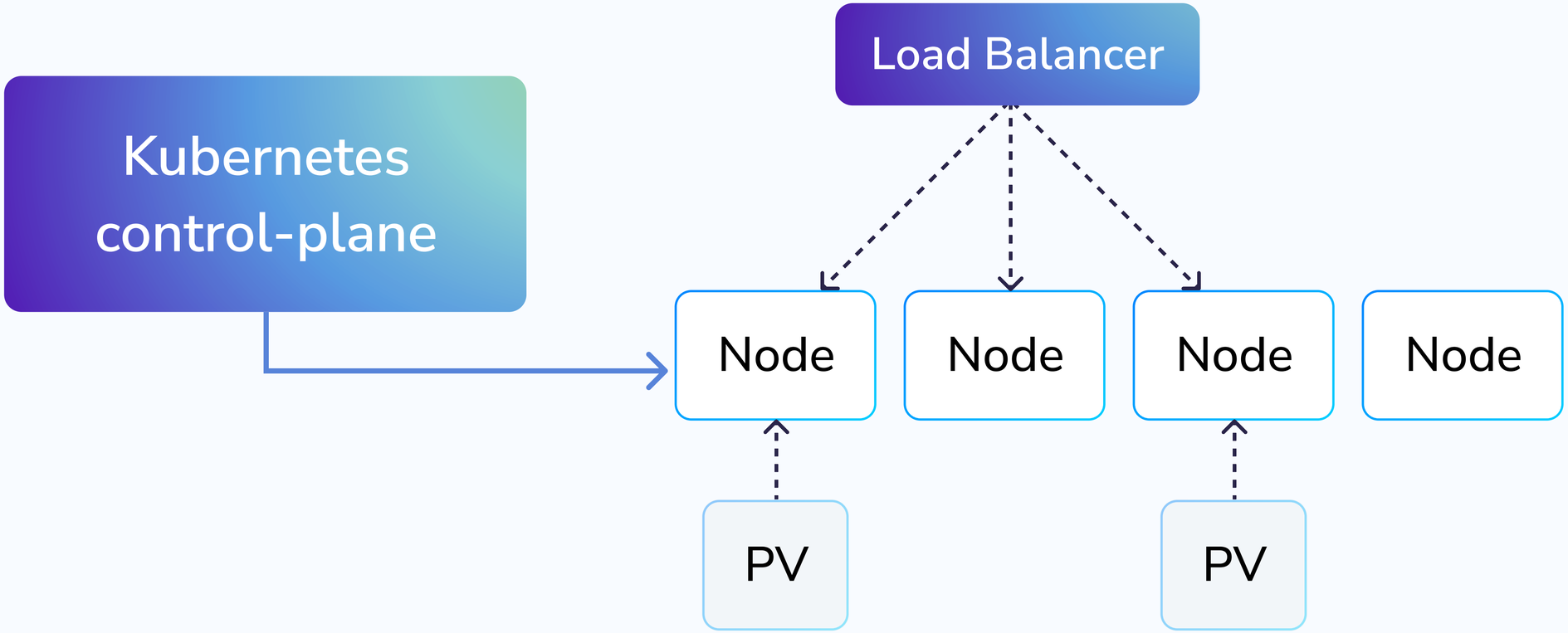
Tenant clusters

Management cluster

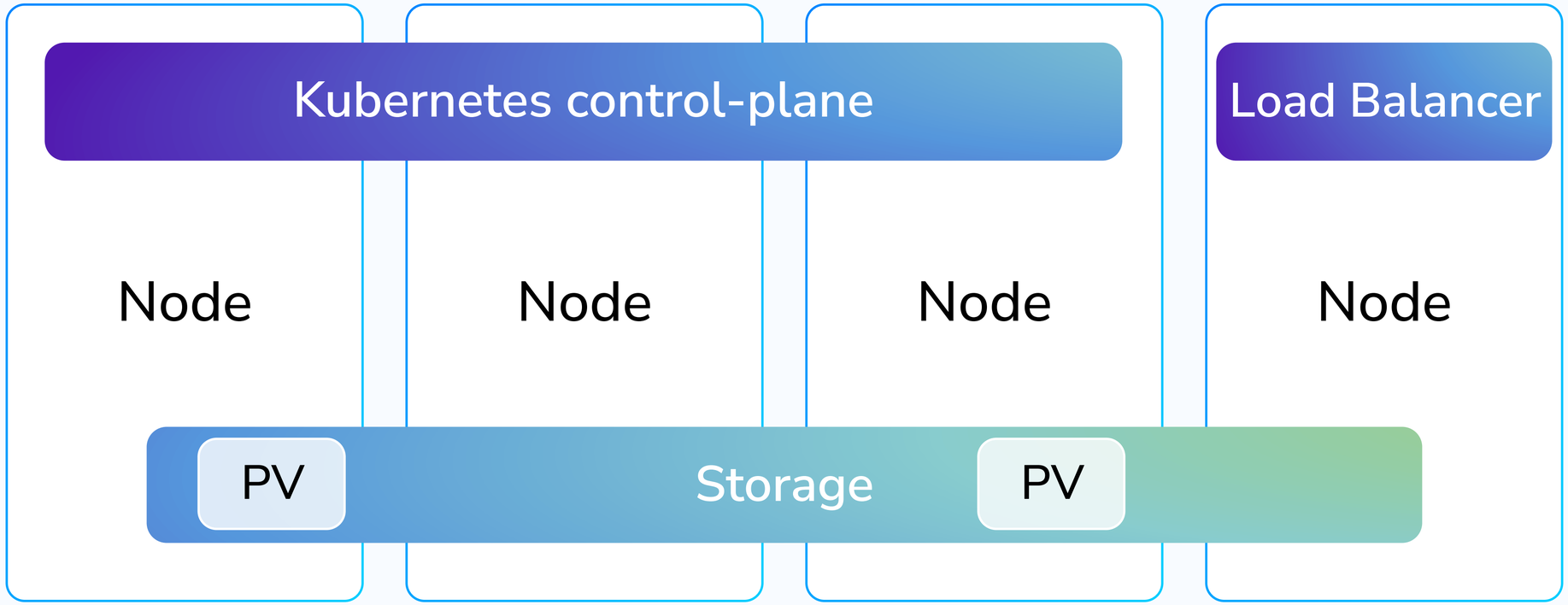


- **единое окружение**
- **единый способ управления**
- **мульти-тенантность**

Тенант кластер:

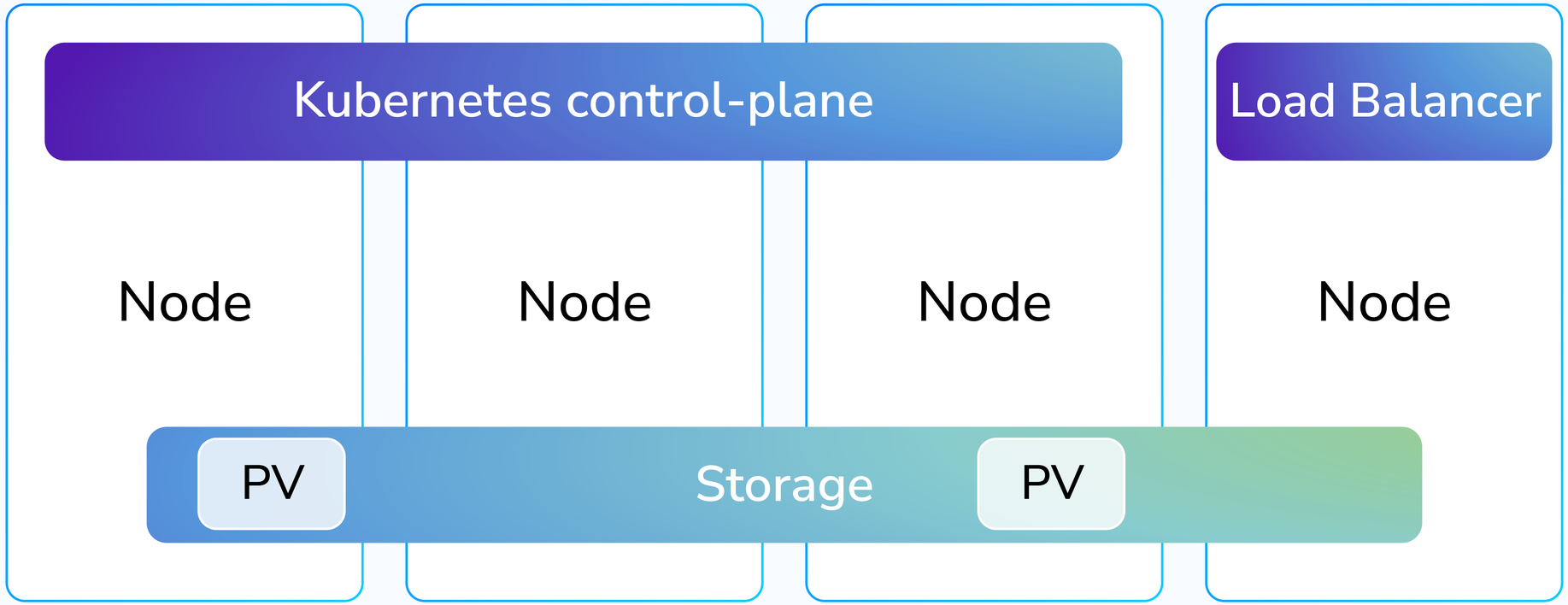


Управляющий кластер:



Управляющий кластер:

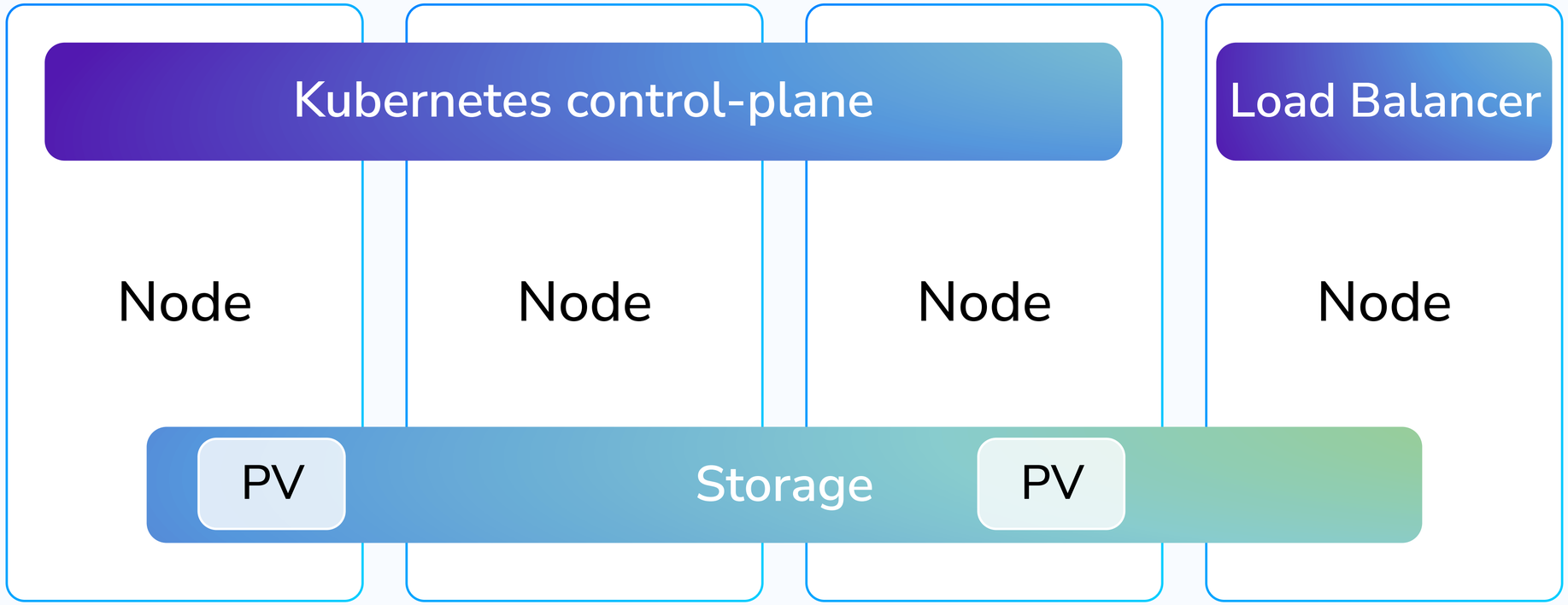
Предоставляет инфраструктурные сервисы :



Управляющий кластер:

Предоставляет инфраструктурные сервисы :

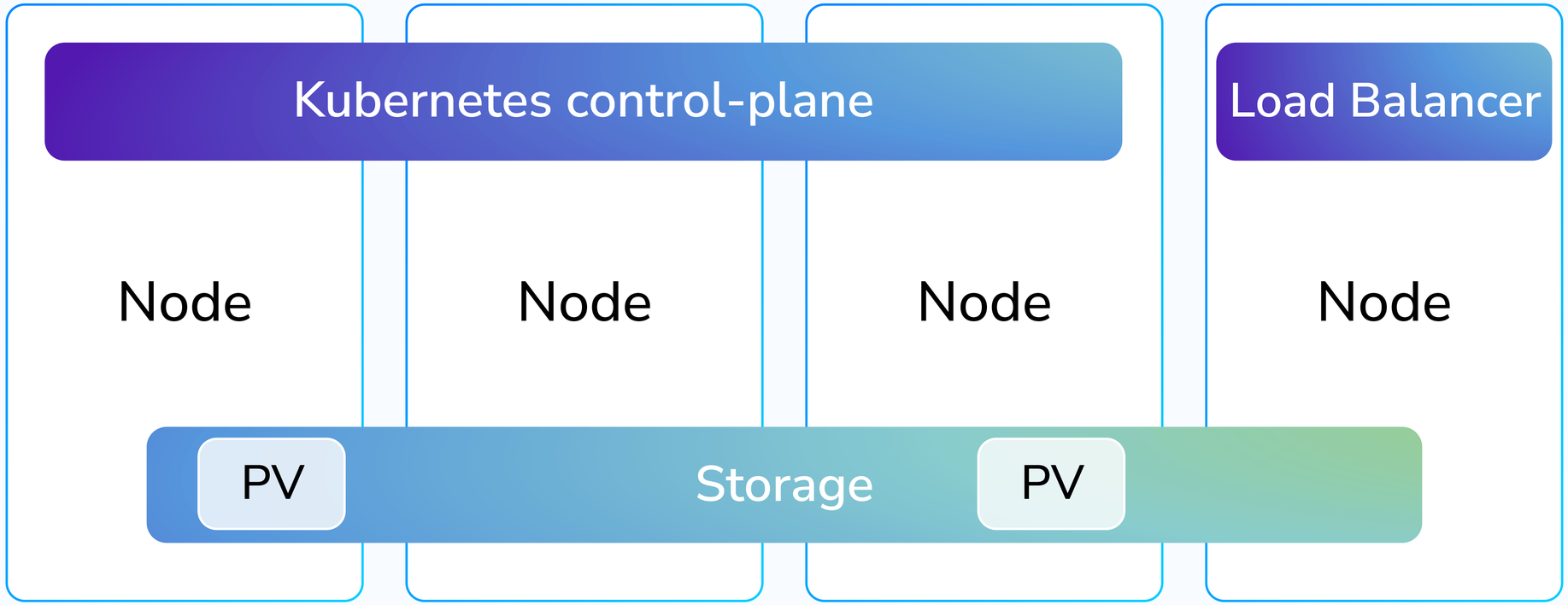
- **виртуальные машины**



Управляющий кластер:

Предоставляет инфраструктурные сервисы :

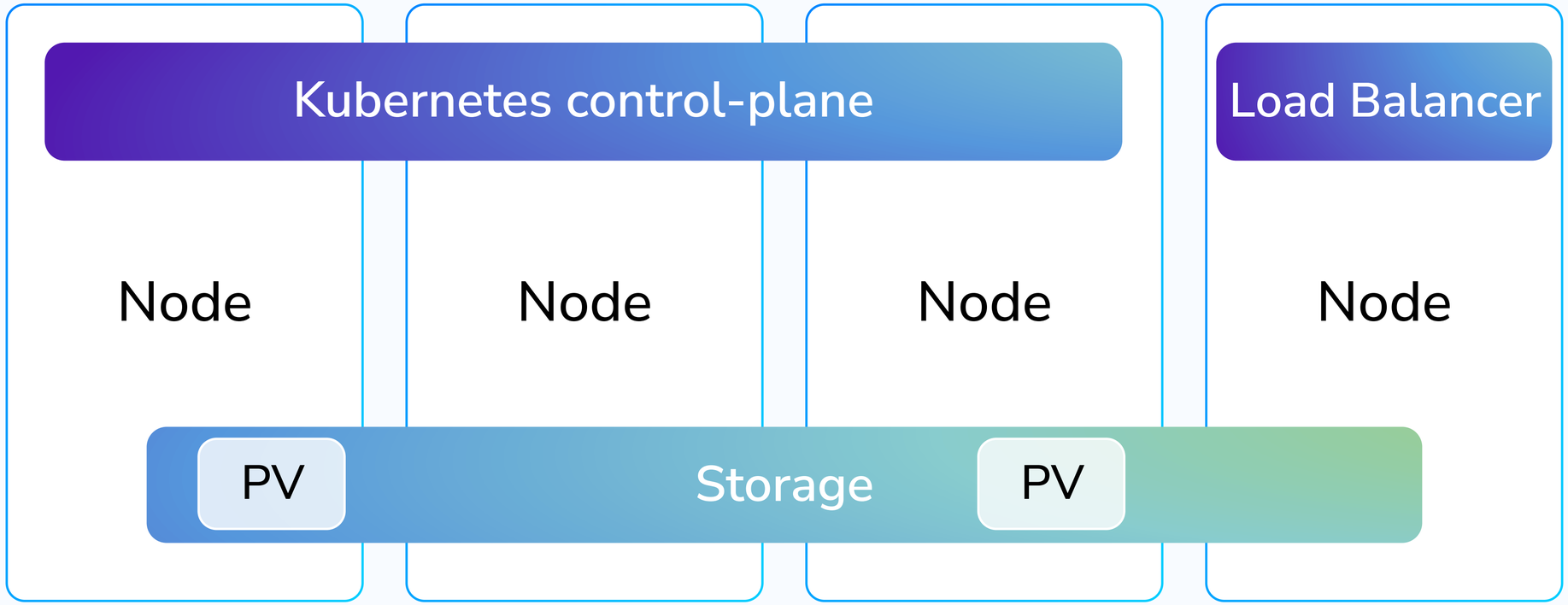
- **виртуальные машины**
- **кластера kubernetes**



Управляющий кластер:

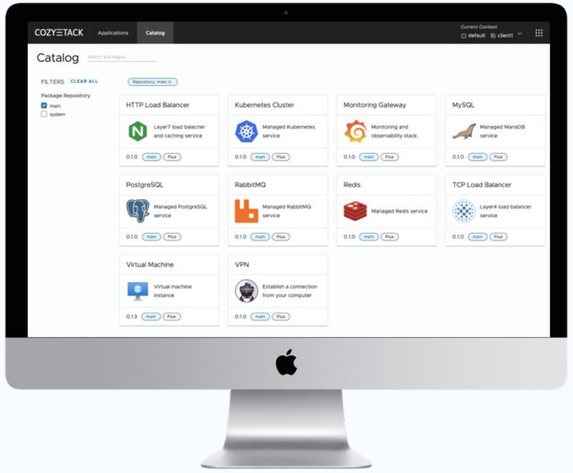
Предоставляет инфраструктурные сервисы :

- **виртуальные машины**
- **кластера kubernetes**
- **СУБД**
- **прочие сервисы (кэши, S3...)**



Платформа

Cozystack с легкостью превращает груду серверов в умную систему с простым REST API для запуска кластеров Kubernetes, баз данных (DBaaS), виртуальных машин, балансировщиков нагрузки, HTTP-кэшей и других сервисов.



Monitoring API Kubernetes

COZYSTACK | Applications | Catalog

Current Context: default / tenant-root

Catalog

search packages...

FILTERS

Package Type

- [ ] Helm Charts via Flux

clickhouse

Managed ClickHouse service

0.2.0 cozystack-apps Flux

http-cache

Layer7 load balacner and caching service

0.1.0 cozystack-apps Flux

kafka

Managed Kafka service

0.1.0 cozystack-apps Flux

kubernetes

Managed Kubernetes service

0.2.0 cozystack-apps Flux

mysql

Managed MariaDB service

0.3.0 cozystack-apps Flux

postgres

Managed PostgreSQL service

0.2.0 cozystack-apps Flux

rabbitmq

Managed RabbitMQ service

0.2.0 cozystack-apps Flux

redis

Managed Redis service

0.2.0 cozystack-apps Flux

tcp-balancer

Layer4 load balancer service

0.2.0 cozystack-apps Flux

tenant

Additional tenant namespace

1.0.0 cozystack-apps Flux

virtual-machine

Virtual machine instance

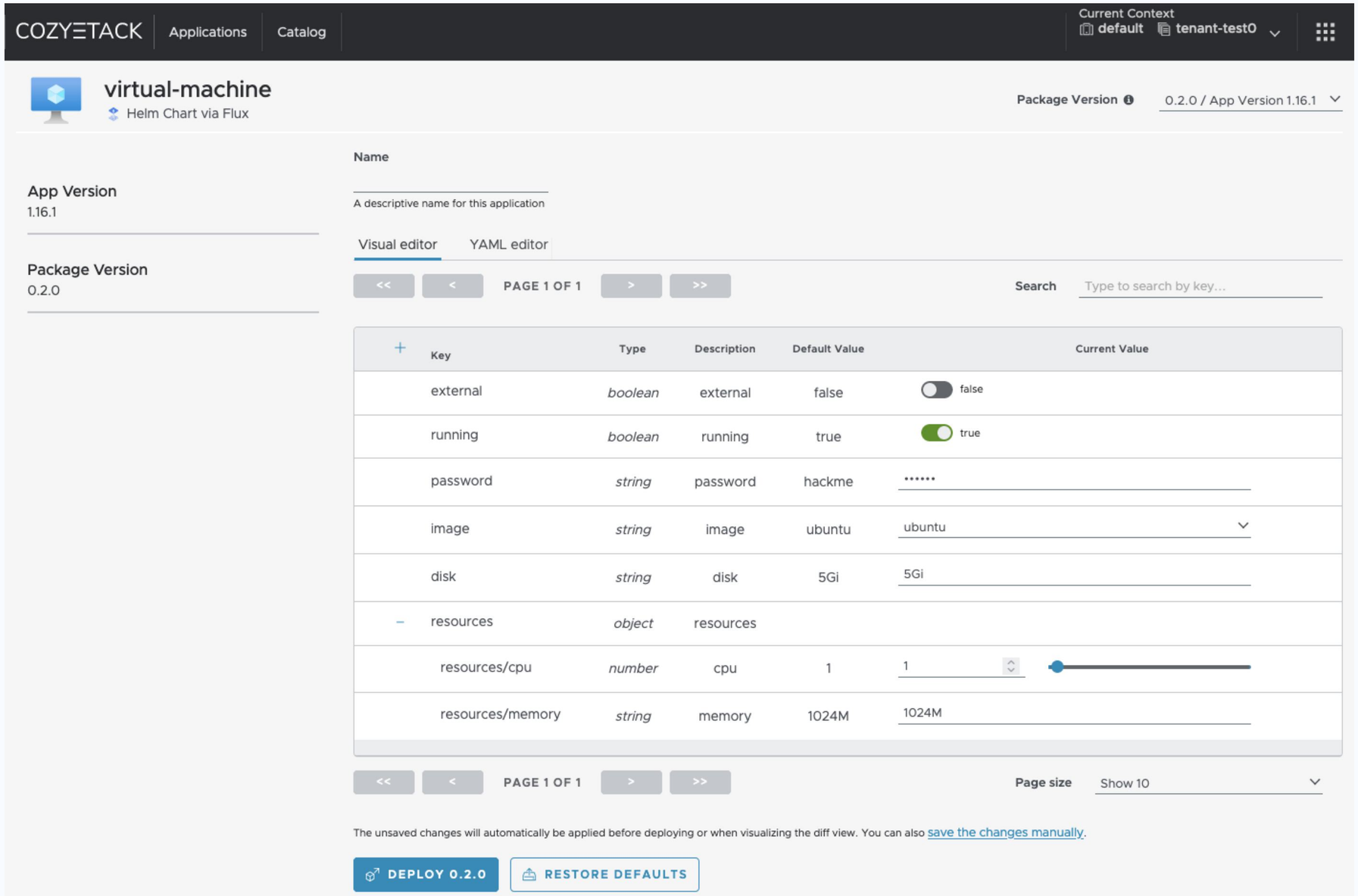
0.1.5 cozystack-apps Flux

vpn

Managed VPN service

0.2.0 cozystack-apps Flux

Развертывание приложения через дашборд



Параметры VM

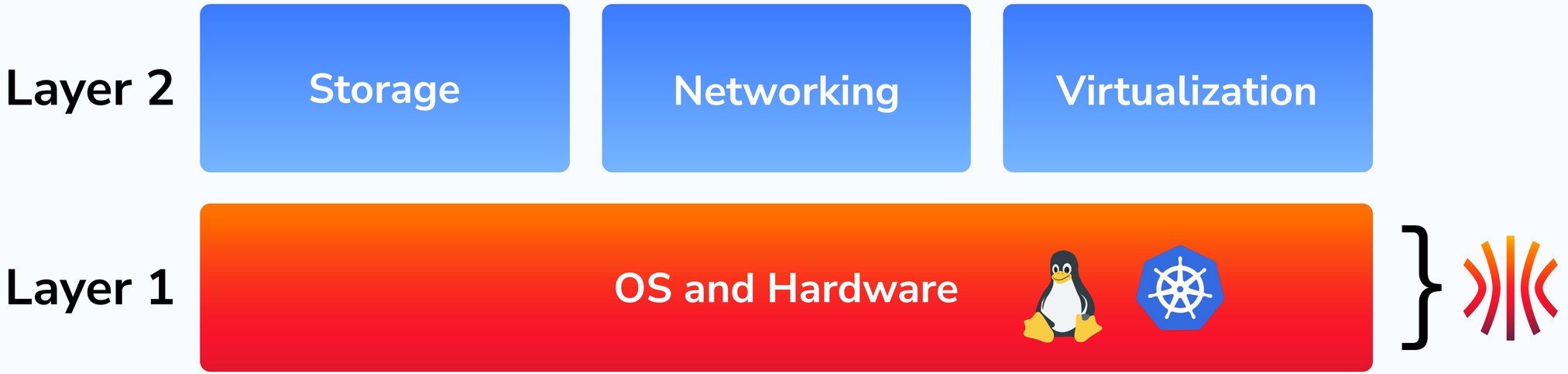

```
external: false
running: true
password: hackme
image: ubuntu
disk: 5Gi
resources:
  cpu: 1
  memory: 1024M
```

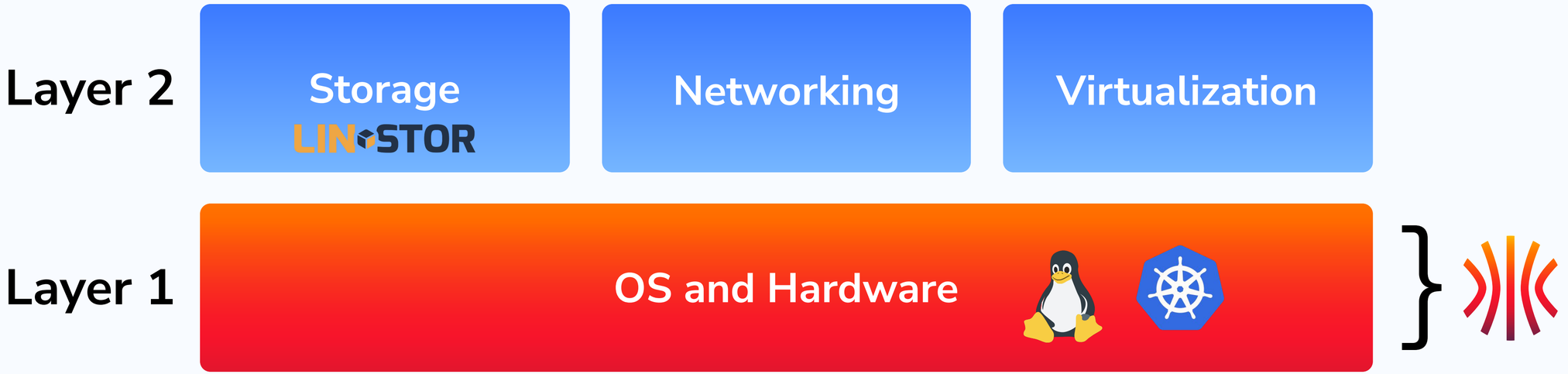
Layer 1

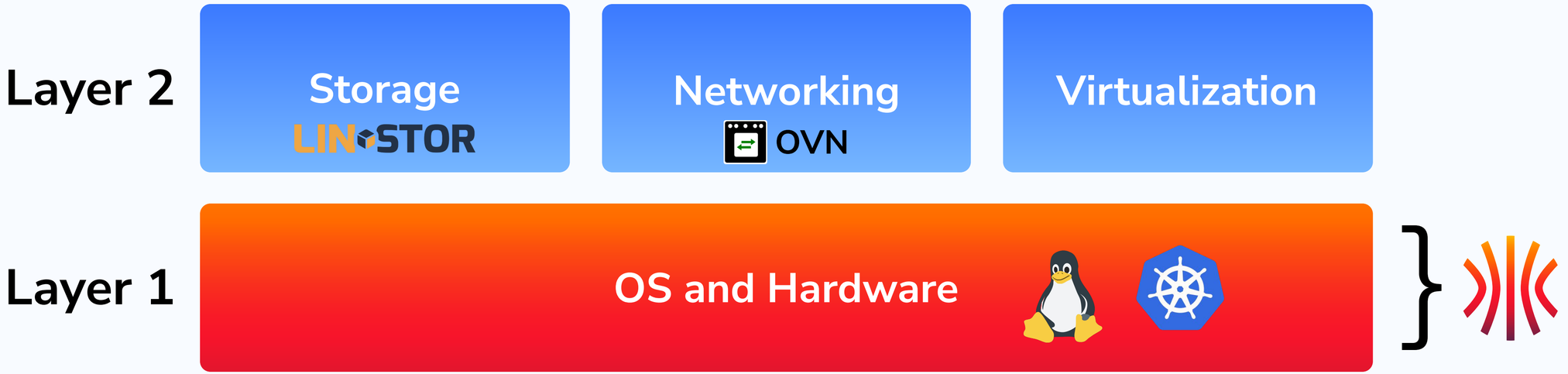
OS and Hardware

Layer 1

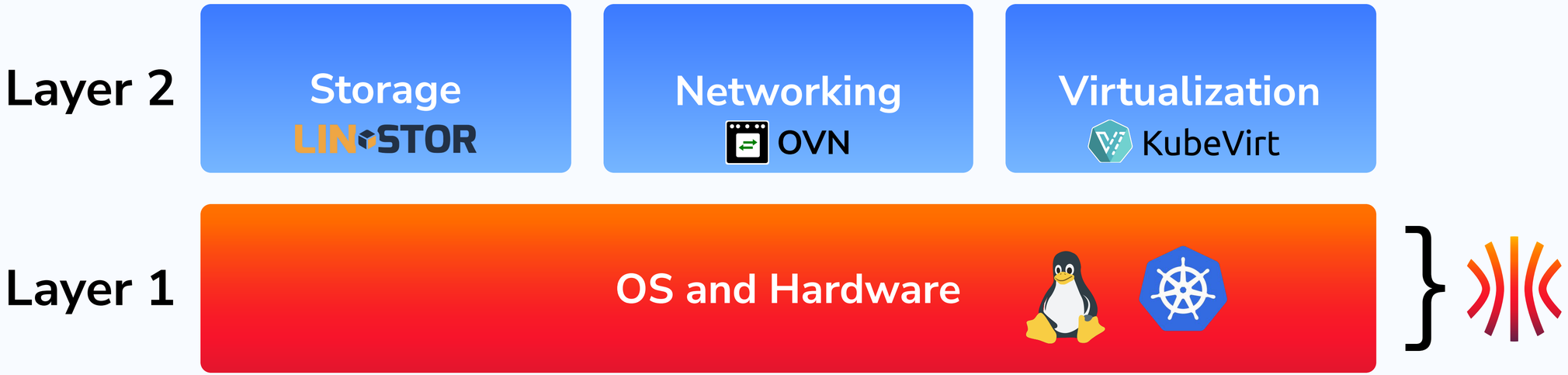
Layer 1

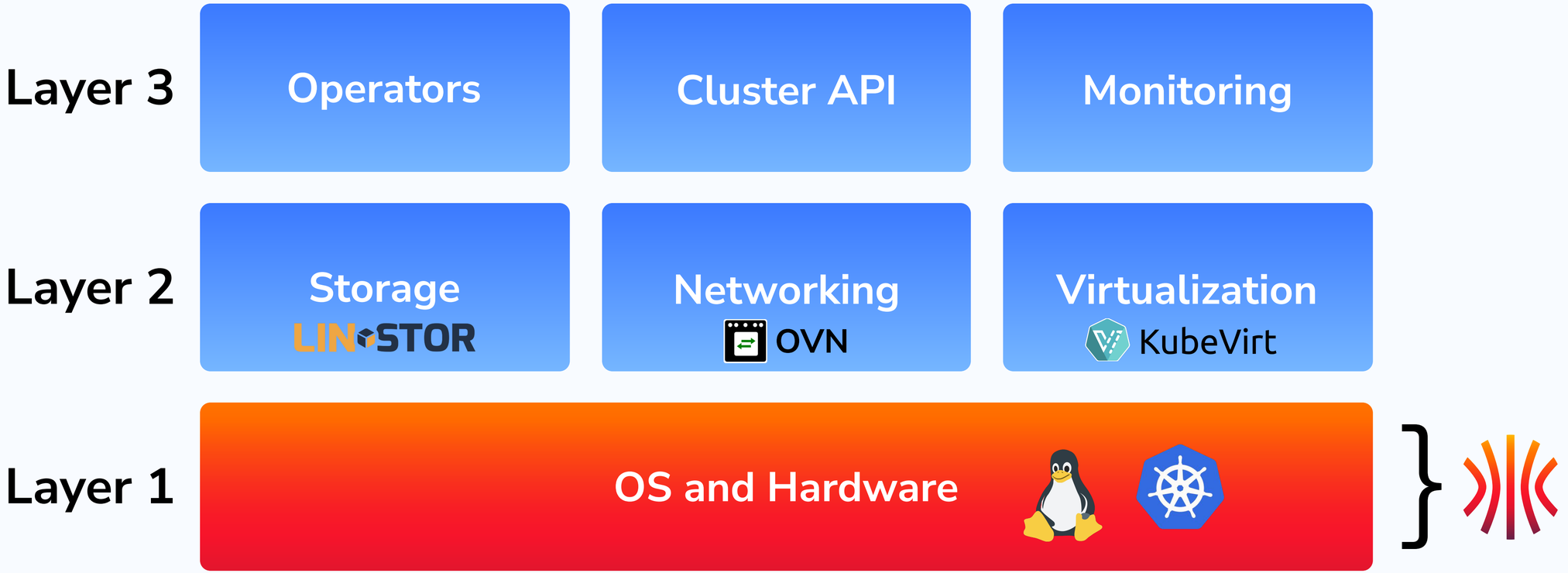

Layer 2
Storage
Networking
Virtualization
Layer 1
OS and Hardware


Layer 2
Storage
LINSTOR
Networking
Virtualization
Layer 1
OS and Hardware

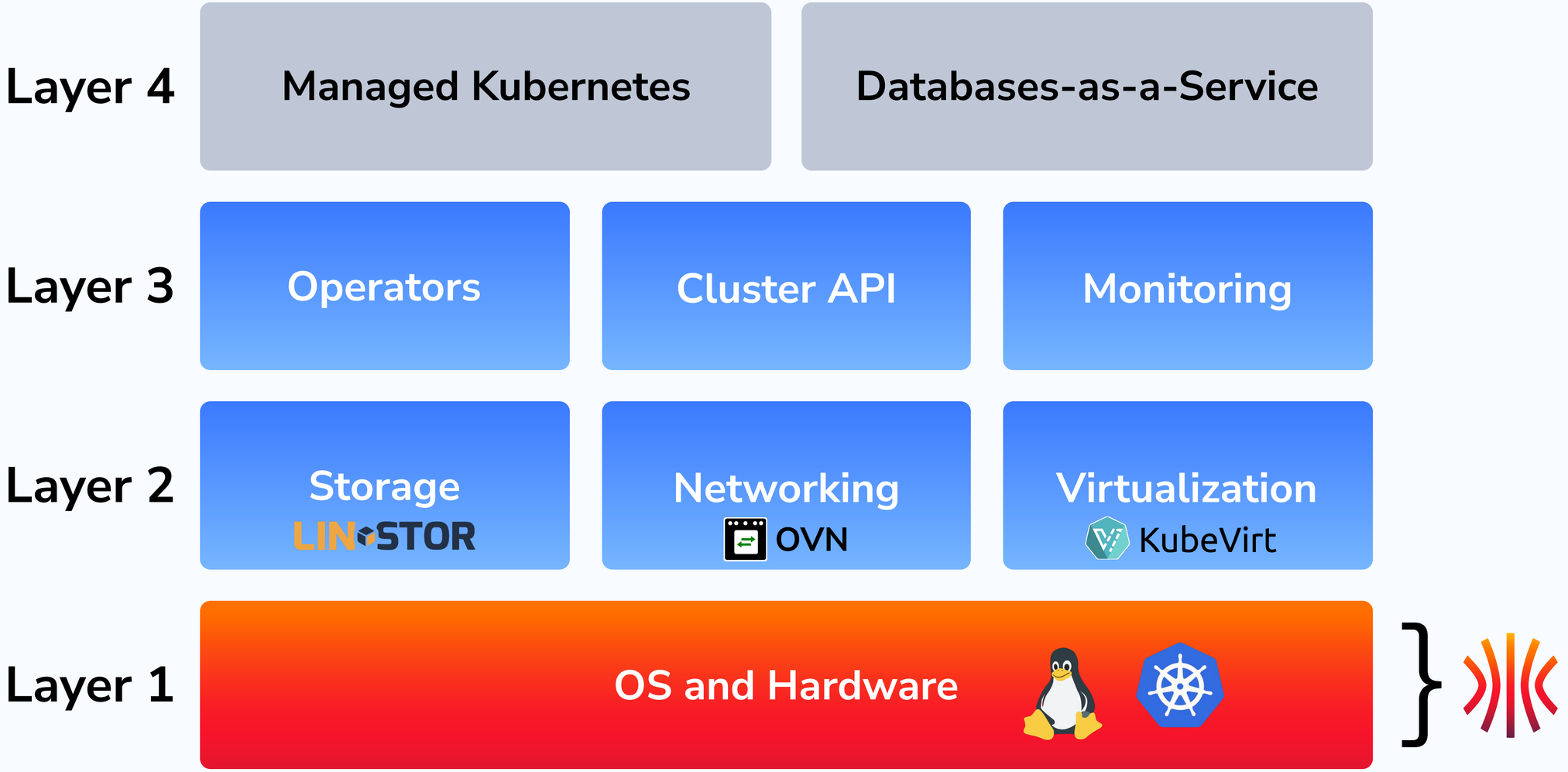

Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
Layer 1
OS and Hardware

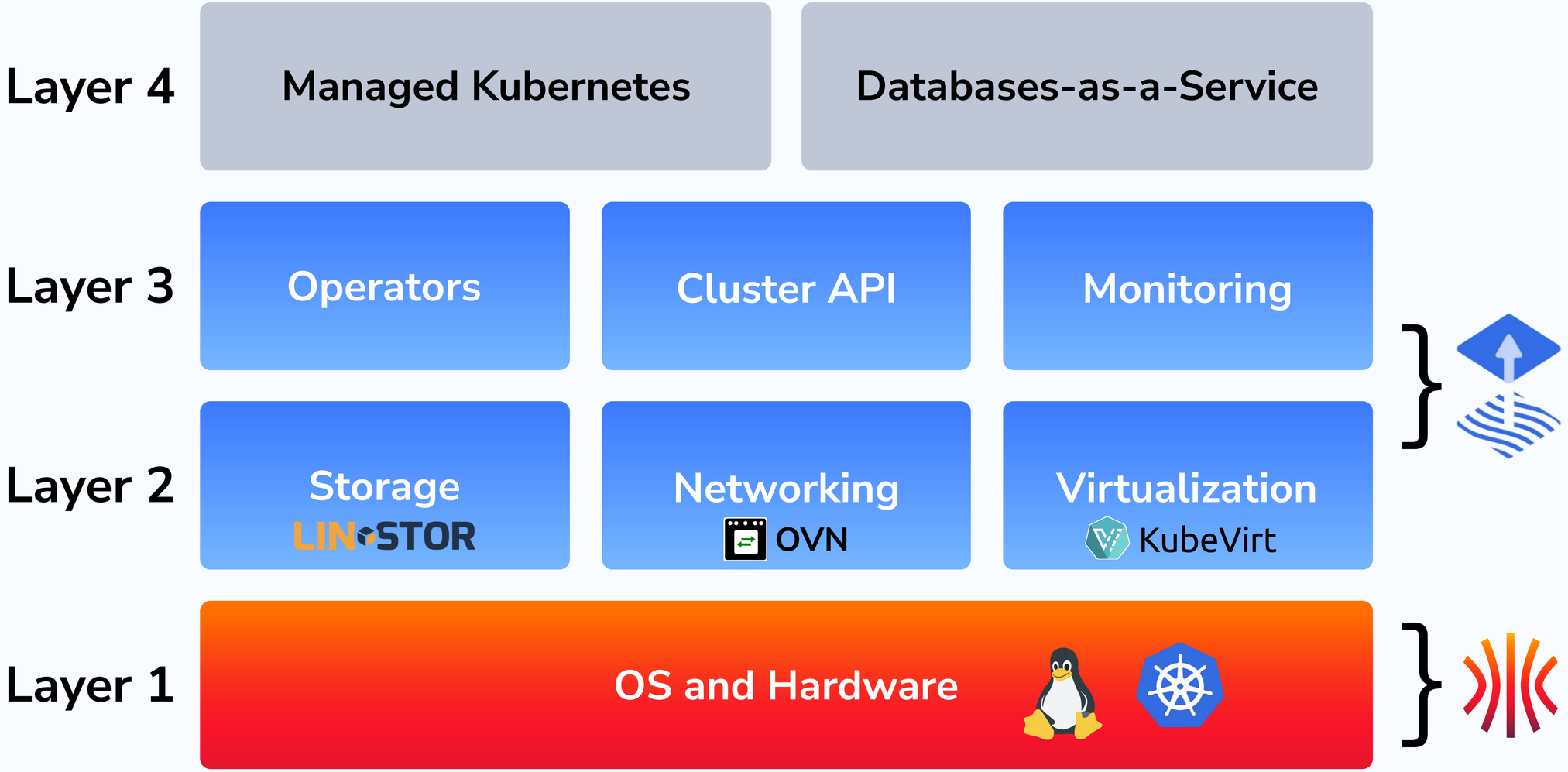
COZYSTACK


Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

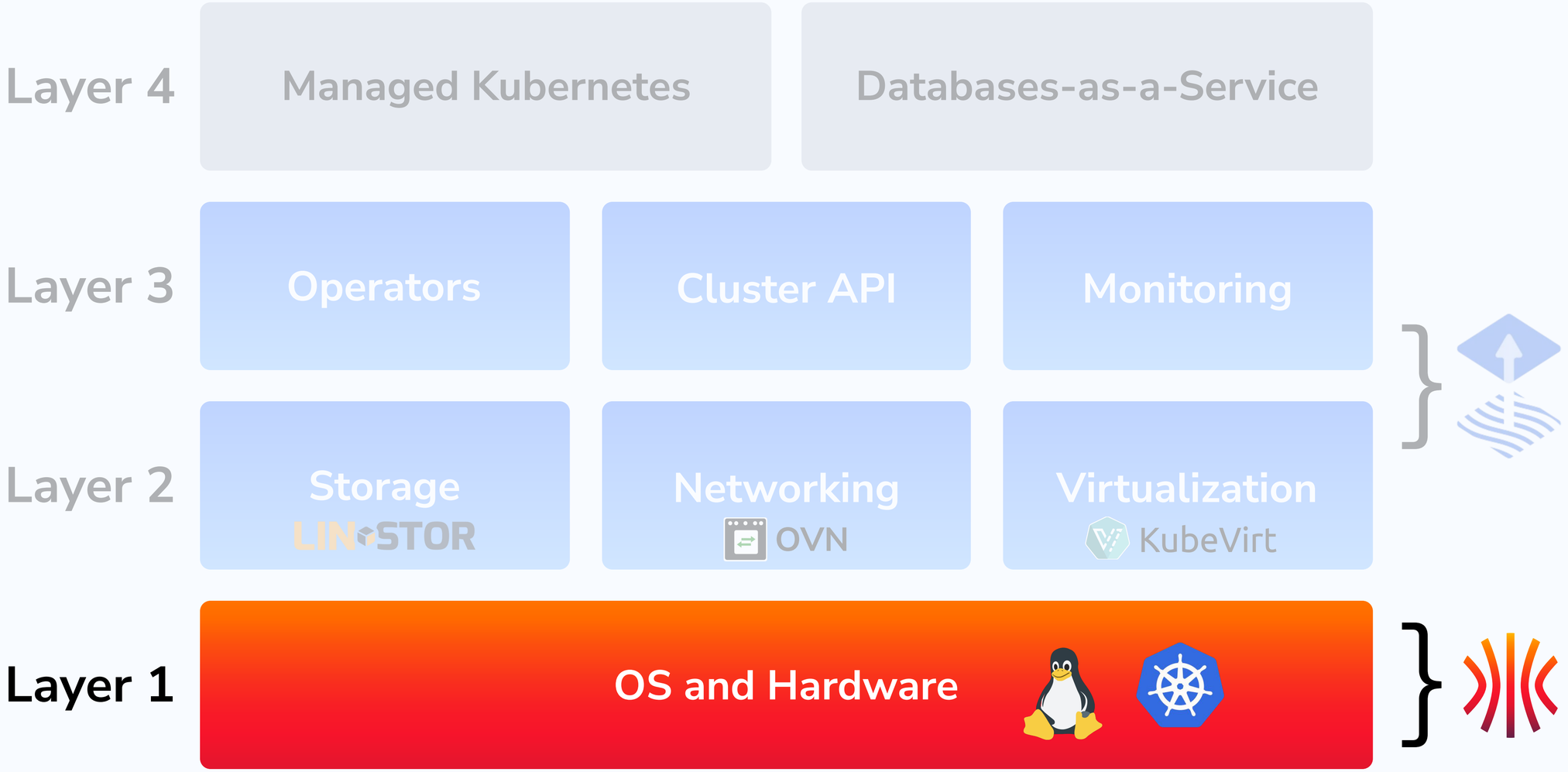

Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

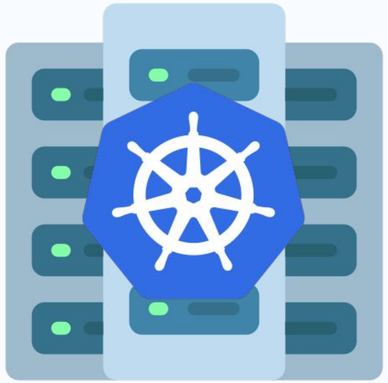
COZYSTACK


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

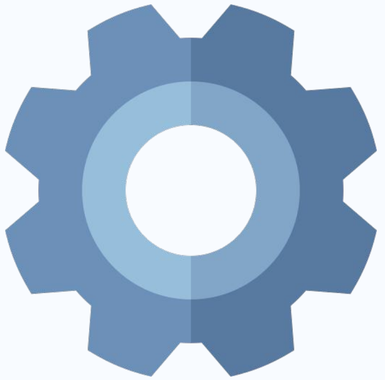
COZYSTACK


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

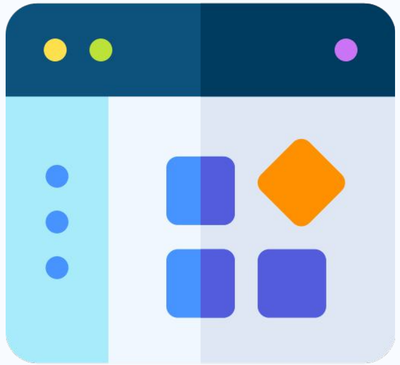


K8s на железе

Железо и ОС

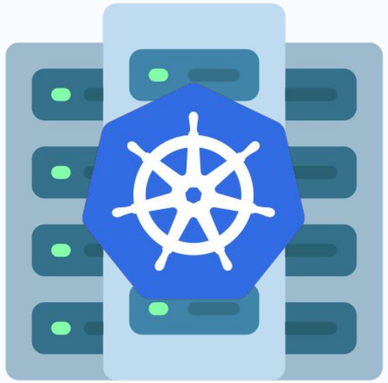


Системные компоненты



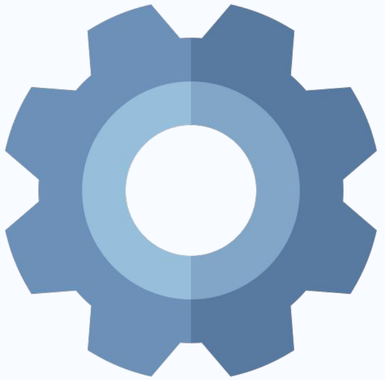
Пользовательские приложения

ПЛАТФОРМА

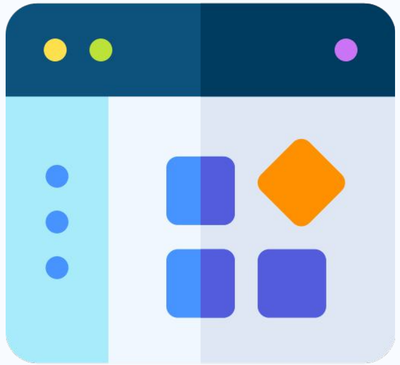


K8s на железе

Железо и ОС



Системные компоненты



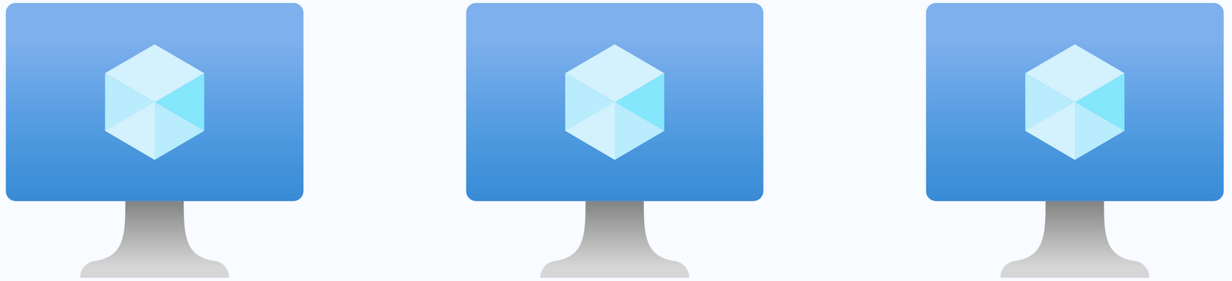
Пользовательские приложения

ПЛАТФОРМА

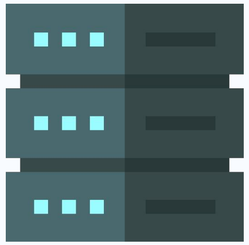
Уже есть инфраструктура

Уже есть инфраструктура

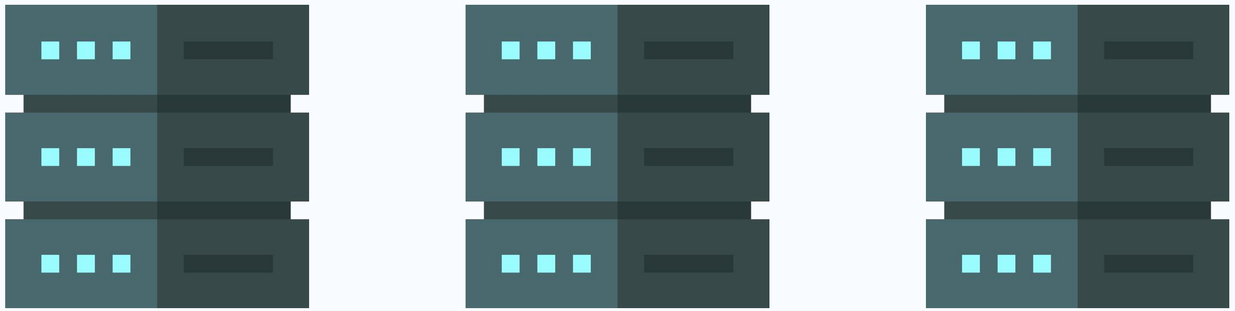
Уже есть инфраструктура



Уже есть инфраструктура



Уже есть инфраструктура



Предсобранные образа Talos Linux:

Предсобранные образа Talos Linux:

- Bare Metal Platforms
  - Digital Rebar
  - Equinix Metal
  - ISO
  - Matchbox
  - Network Configuration
  - PXE
  - SecureBoot

Предсобранные образа Talos Linux:

- **Bare Metal Platforms**
  - Digital Rebar
  - Equinix Metal
  - ISO
  - Matchbox
  - Network Configuration
  - PXE
  - SecureBoot
- **Virtualized Platforms**
  - Hyper-V
  - KVM
  - Proxmox
  - Vagrant & Libvirt
  - VMware
  - Xen

Предсобранные образа Talos Linux:

- **Bare Metal Platforms**
  - Digital Rebar
  - Equinix Metal
  - ISO
  - Matchbox
  - Network Configuration
  - PXE
  - SecureBoot
- **Virtualized Platforms**
  - Hyper-V
  - KVM
  - Proxmox
  - Vagrant & Libvirt
  - VMware
  - Xen
- **Cloud Platforms**
  - AWS
  - Azure
  - DigitalOcean
  - Exoscale
  - GCP
  - Hetzner
  - Nocloud
  - Openstack
  - Oracle
  - Scaleway
  - UpCloud
  - Vultr

Предсобранные образа Talos Linux:

- **Bare Metal Platforms**
  - Digital Rebar
  - Equinix Metal
  - ISO
  - Matchbox
  - Network Configuration
  - PXE
  - SecureBoot
- **Virtualized Platforms**
  - Hyper-V
  - KVM
  - Proxmox
  - Vagrant & Libvirt
  - VMware
  - Xen
- **Cloud Platforms**
  - AWS
  - Azure
  - DigitalOcean
  - Exoscale
  - GCP
  - Hetzner
  - Nocloud
  - Openstack
  - Oracle
  - Scaleway
  - UpCloud
  - Vultr
- **Local Platforms**
  - Docker

Предсобранные образа Talos Linux:

- Bare Metal Platforms
  - Digital Rebar
  - Equinix Metal
  - ISO
  - Matchbox
  - Network Configuration
  - PXE
  - SecureBoot
- Virtualized Platforms
  - Hyper-V
  - KVM
  - Proxmox
  - Vagrant & Libvirt
  - VMware
  - Xen
- Cloud Platforms
  - AWS
  - Azure
  - DigitalOcean
  - Exoscale
  - GCP
  - Hetzner
  - Nocloud
  - Openstack
  - Oracle
  - Scaleway
  - UpCloud
  - Vultr
- Local Platforms
  - Docker

Предсобранные образа Talos Linux:

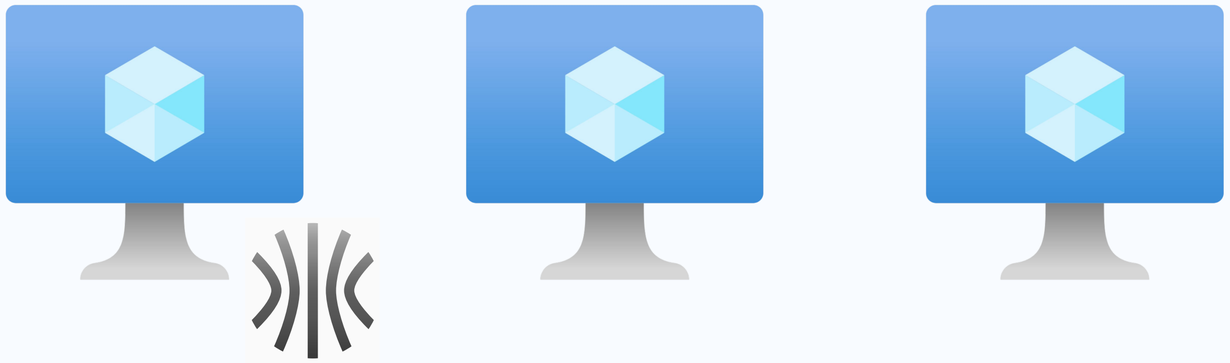
- Bare Metal Platforms
  - Digital Rebar
  - Equinix Metal
  - ISO
  - Matchbox
  - Network Configuration
  - PXE
  - SecureBoot
- Virtualized Platforms
  - Hyper-V
  - KVM
  - Proxmox
  - Vagrant & Libvirt
  - VMware
  - Xen
- Cloud Platforms
  - AWS
  - Azure
  - DigitalOcean
  - Exoscale
  - GCP
  - Hetzner
  - Nocloud
  - Openstack
  - Oracle
  - Scaleway
  - UpCloud
  - Vultr
- Local Platforms
  - Docker

Предсобранные образа Talos Linux:

- Bare Metal Platforms
  - Digital Rebar
  - Equinix Metal
  - ISO
  - Matchbox
  - Network Configuration
  - PXE
  - SecureBoot
- Virtualized Platforms
  - Hyper-V
  - KVM
  - Proxmox
  - Vagrant & Libvirt
  - VMware
  - Xen
- Cloud Platforms
  - AWS
  - Azure
  - DigitalOcean
  - Exoscale
  - GCP
  - Hetzner
  - Nocloud
  - Openstack
  - Oracle
  - Scaleway
  - UpCloud
  - Vultr
- Local Platforms
  - Docker

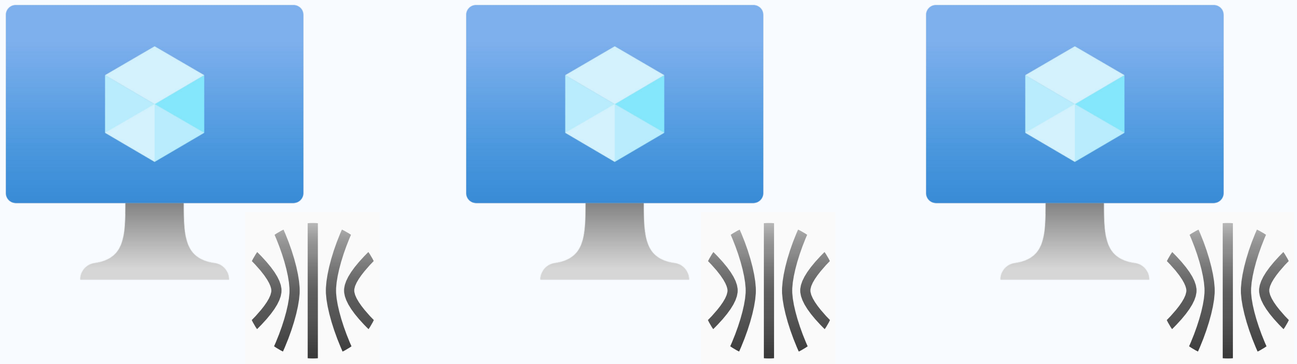
Для первоначального запуска нужно просто загрузить образ Talos

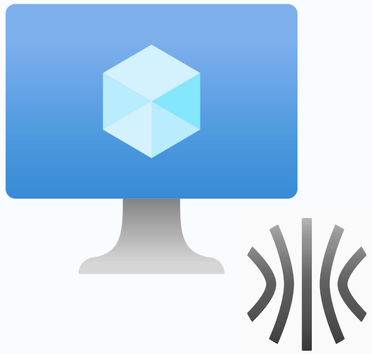
Для первоначального запуска нужно просто загрузить образ Talos

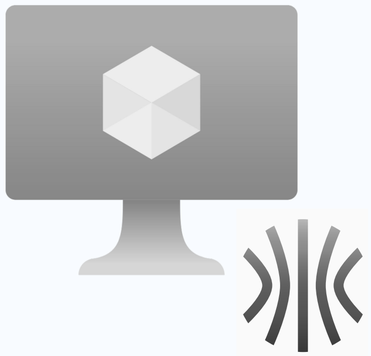


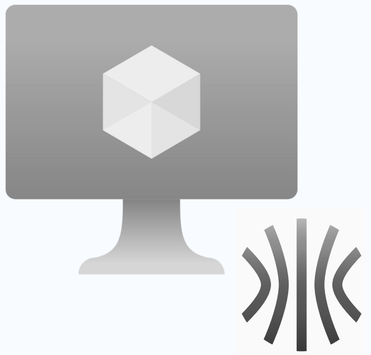
Для первоначального запуска нужно просто загрузить образ Talos

Для первоначального запуска нужно просто загрузить образ Talos

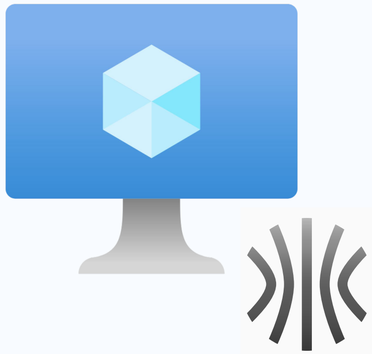


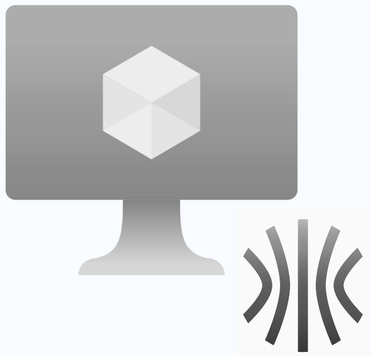


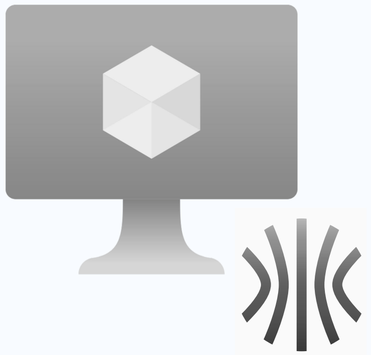




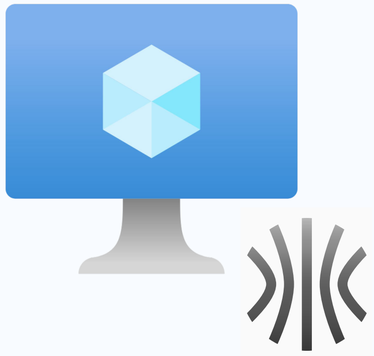
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```



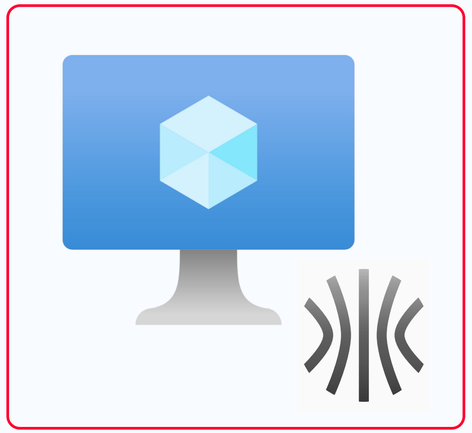




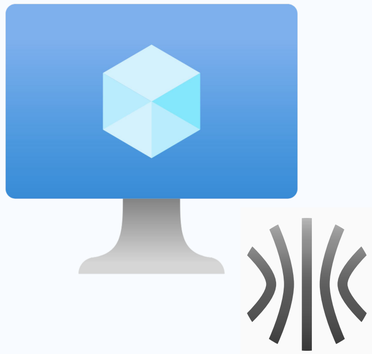
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

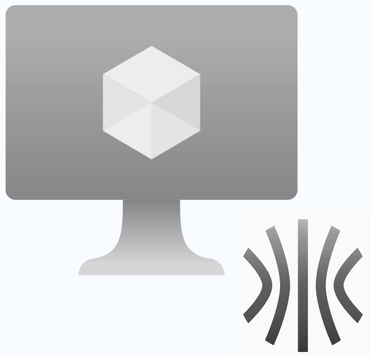


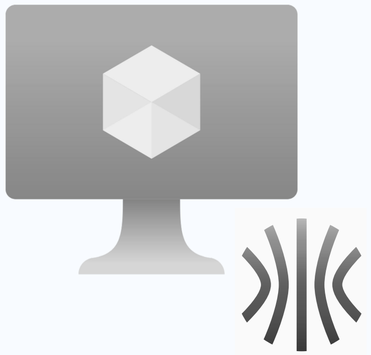

```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```



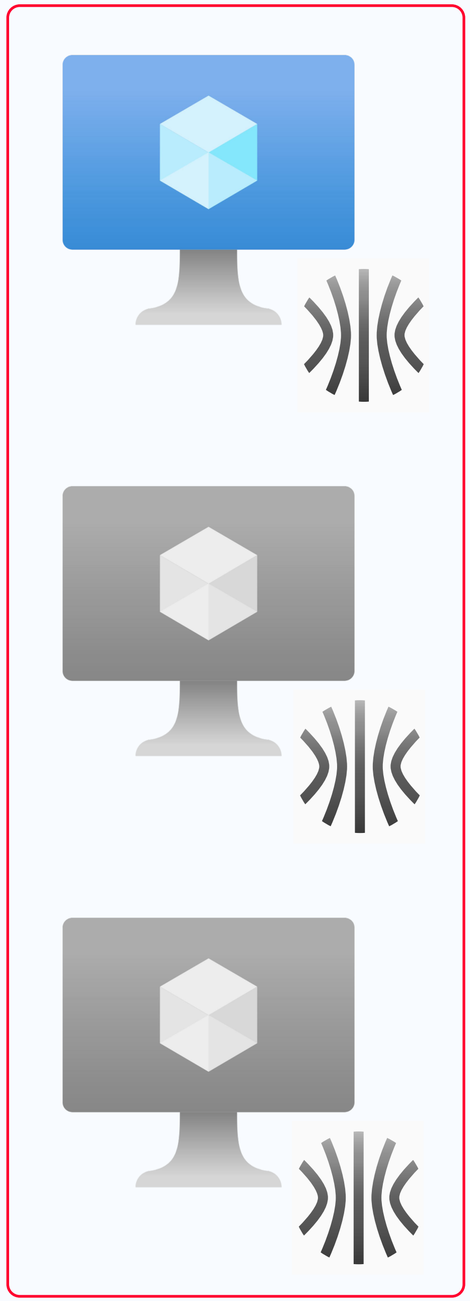

```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```



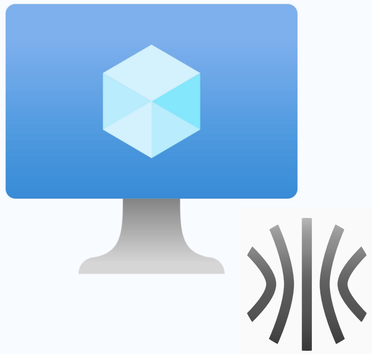


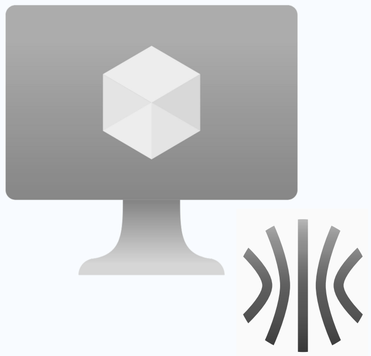


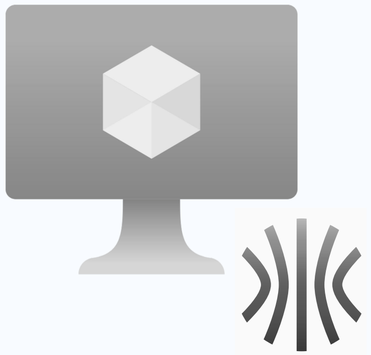
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```



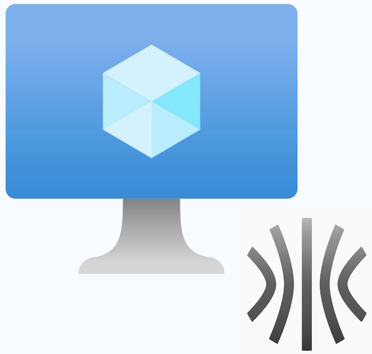
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

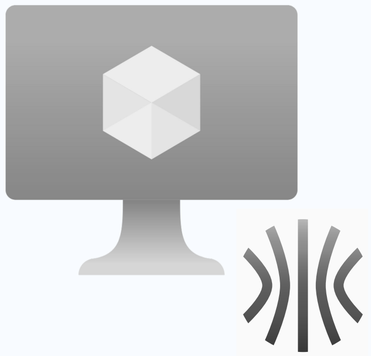


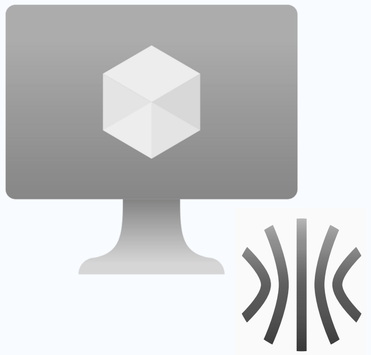




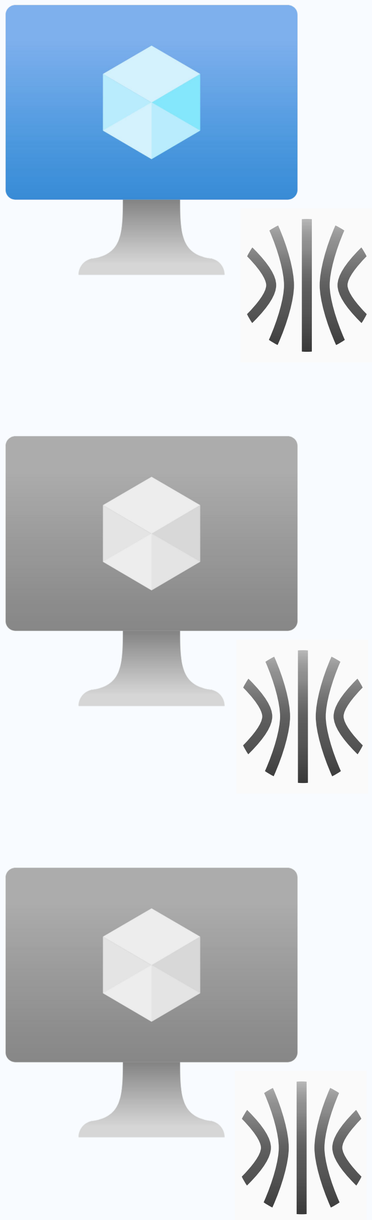
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```



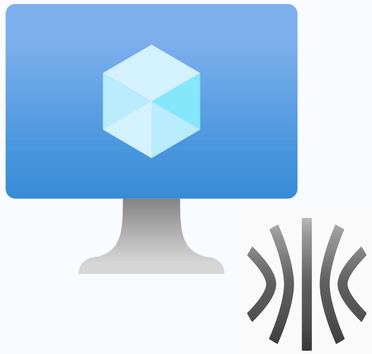


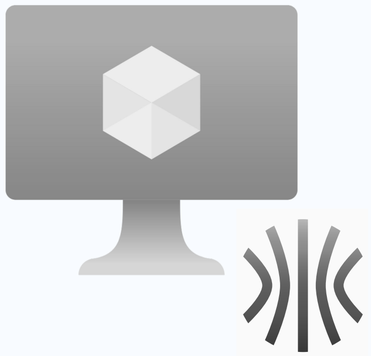


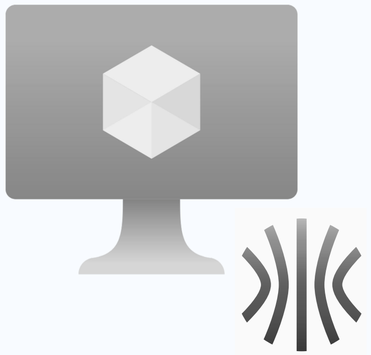
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```



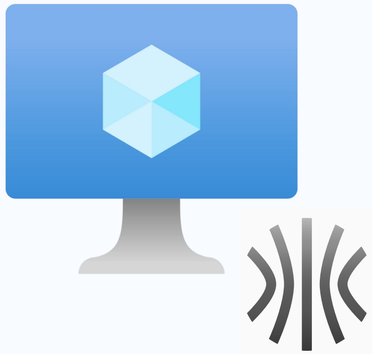
```
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

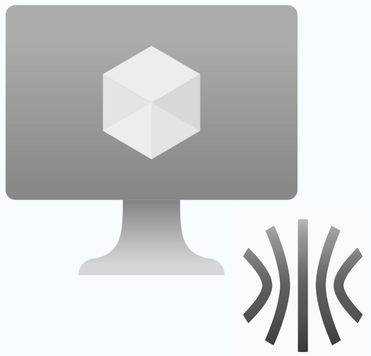


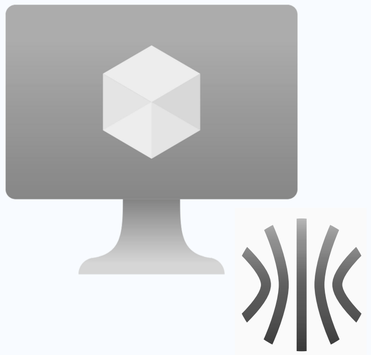




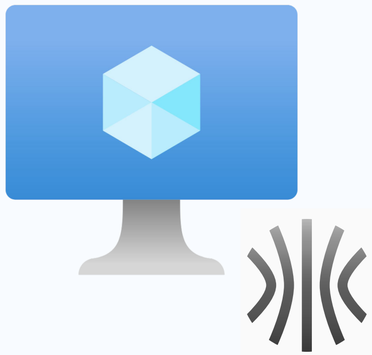
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

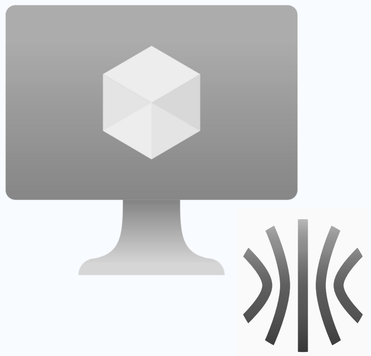


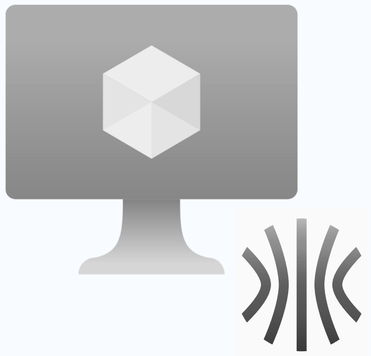




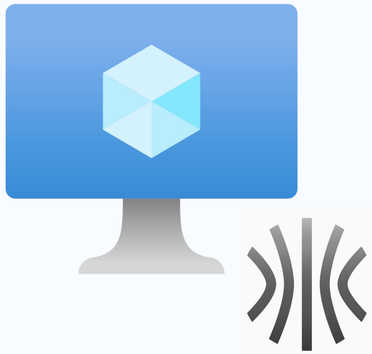
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

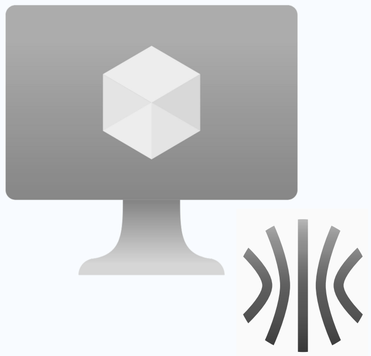


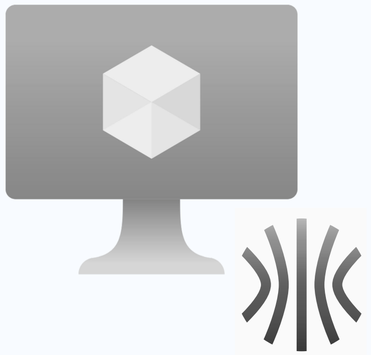




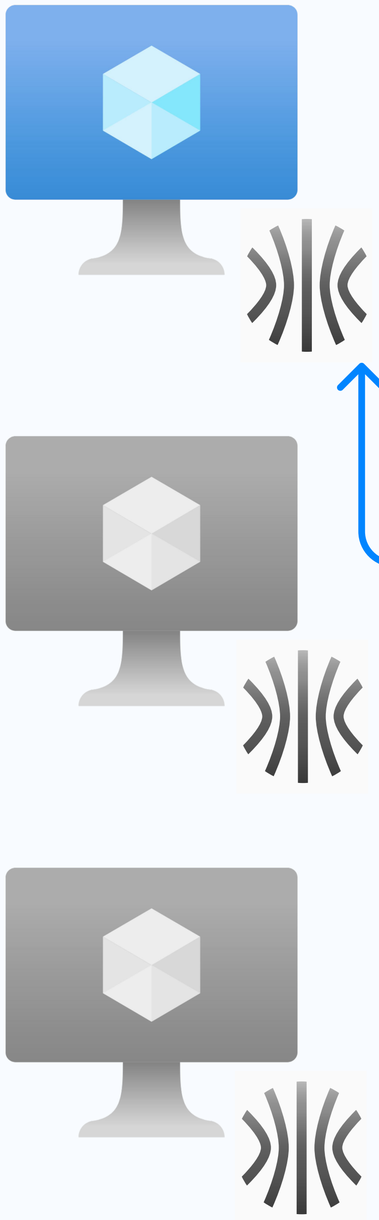
```
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```





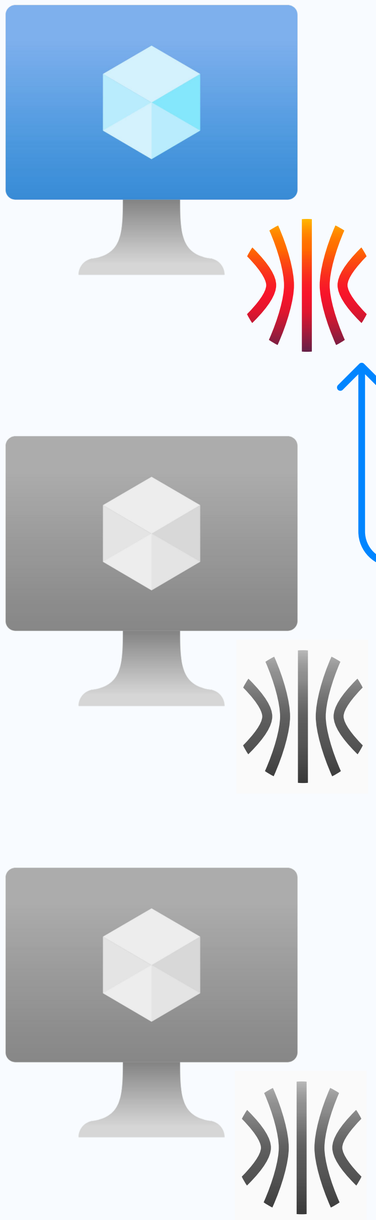


```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```



talosctl apply -f

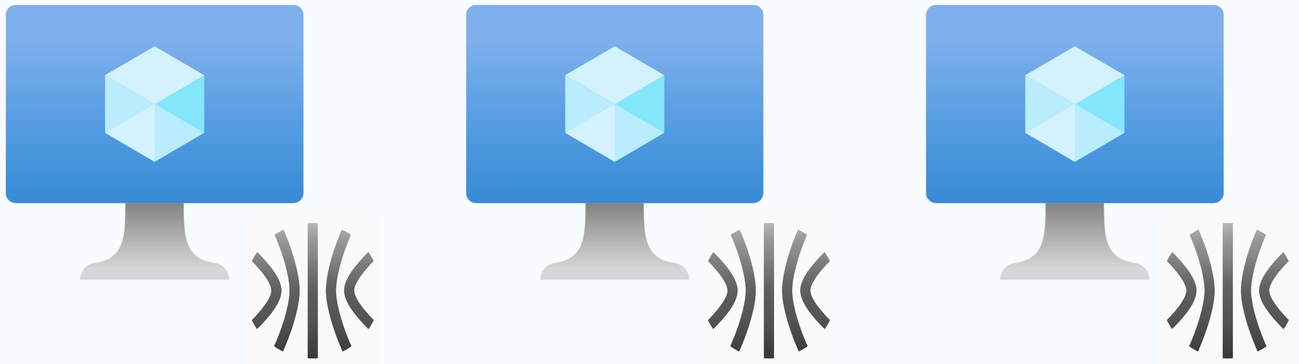
```
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```



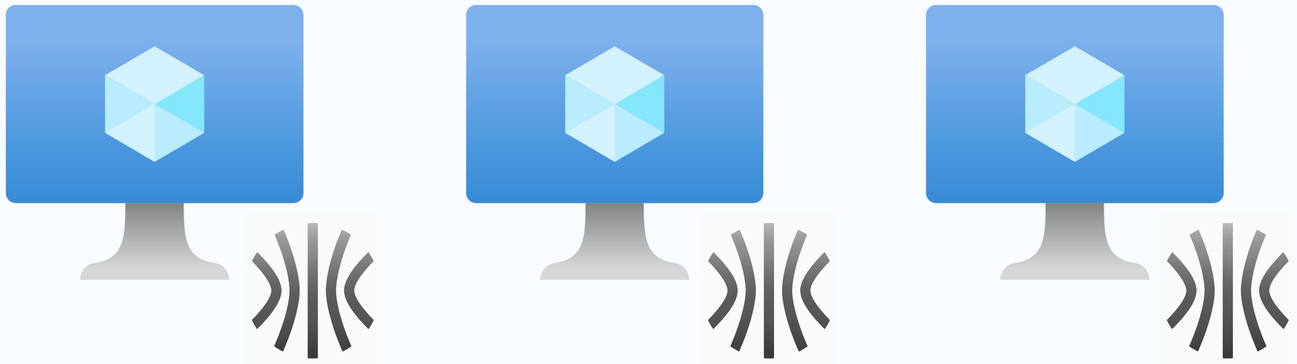
talosctl apply -f

```
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

Первоначальный запуск кластера

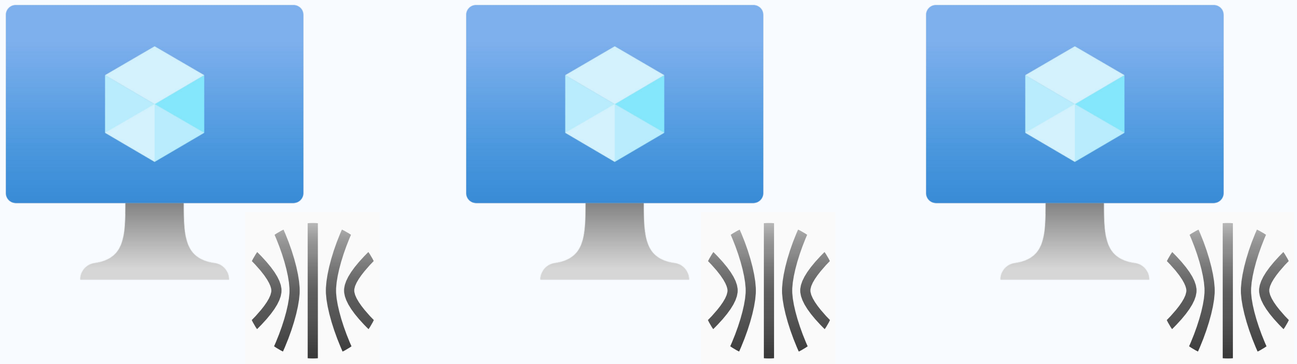


Первоначальный запуск кластера



talosctl gen secrets -o secrets.yaml

Первоначальный запуск кластера



talosctl gen secrets -o secrets.yaml

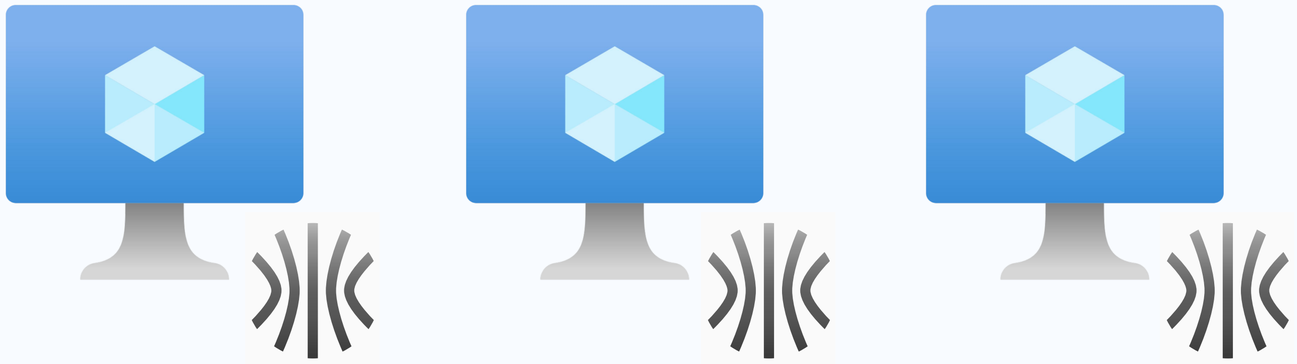
secrets.yaml

Первоначальный запуск кластера

talosctl gen secrets -o secrets.yaml

secrets.yaml

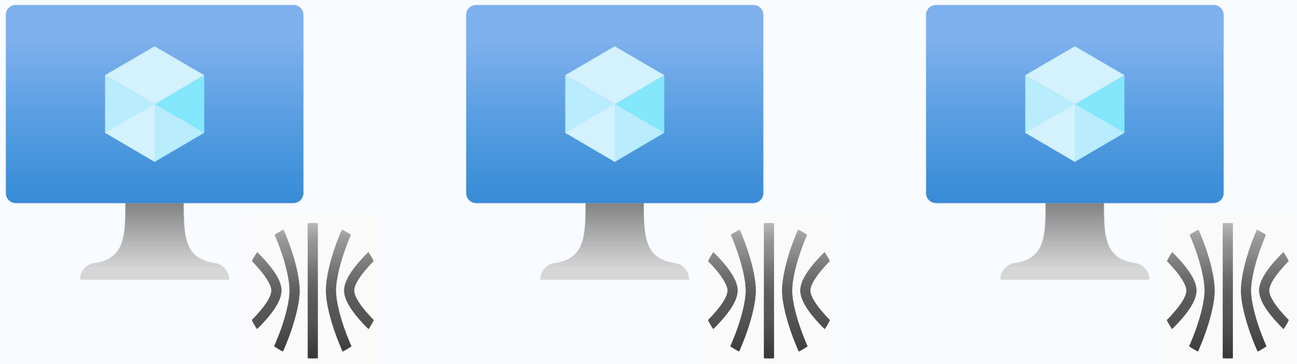
Первоначальный запуск кластера



talosctl gen secrets -o secrets.yaml

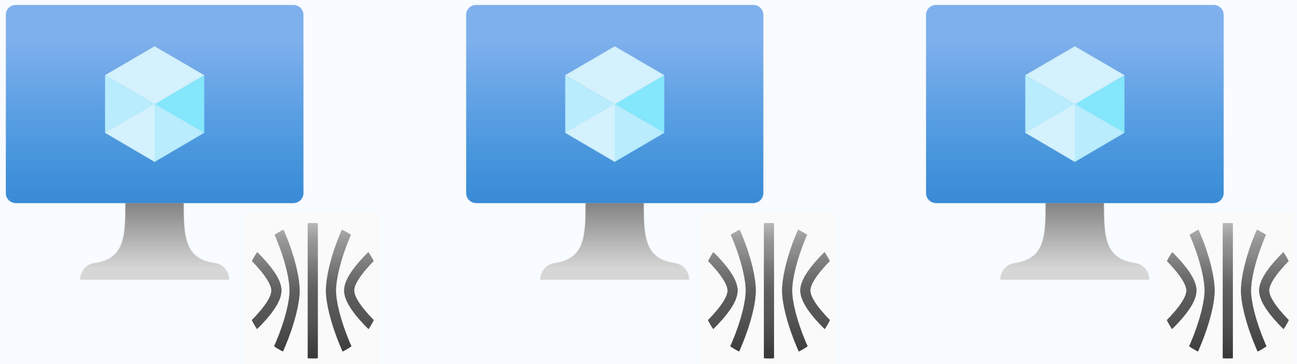
secrets.yaml

Первоначальный запуск кластера



secrets.yaml

Первоначальный запуск кластера



talosctl gen config --with-secrets secrets.yaml

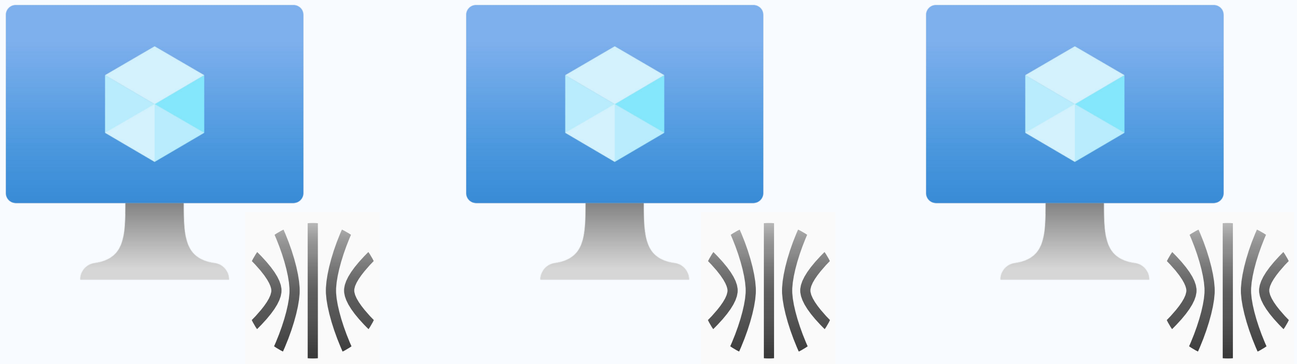
secrets.yaml

Первоначальный запуск кластера

talosctl gen config --with-secrets secrets.yaml

secrets.yaml

Первоначальный запуск кластера



talosctl gen config --with-secrets secrets.yaml

secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Первоначальный запуск кластера

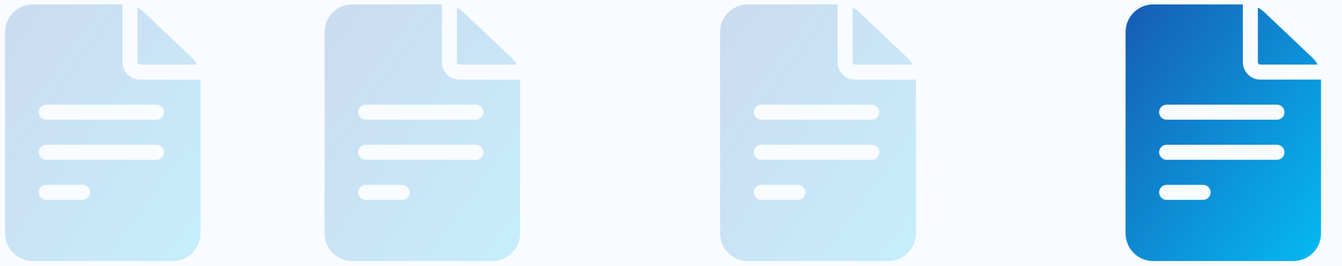
talosctl gen config --with-secrets secrets.yaml

Первоначальный запуск кластера

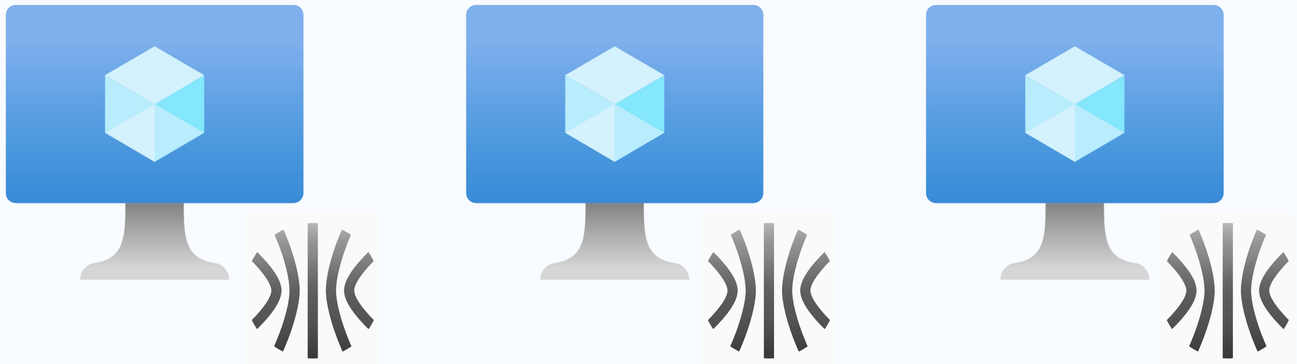
talosctl gen config --with-secrets secrets.yaml

Первоначальный запуск кластера

talosctl gen config --with-secrets secrets.yaml



Первоначальный запуск кластера



talosctl gen config --with-secrets secrets.yaml

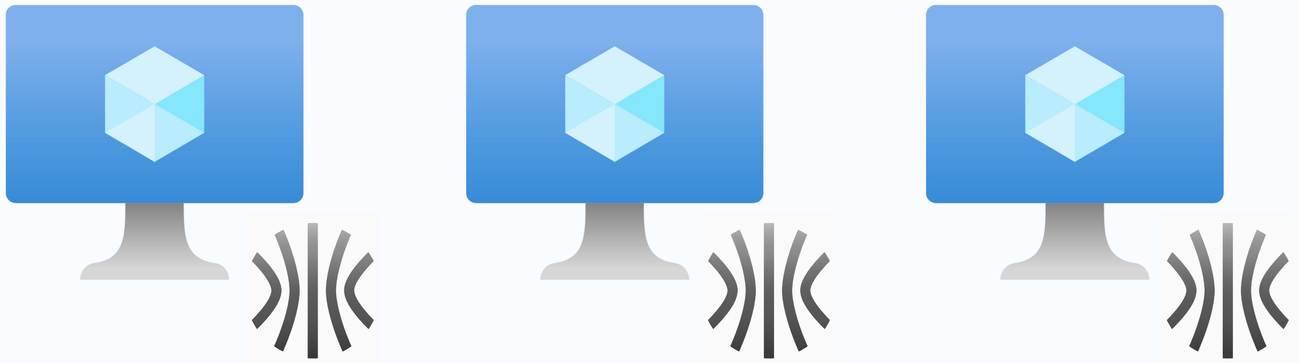
secrets.yaml

talosconfig

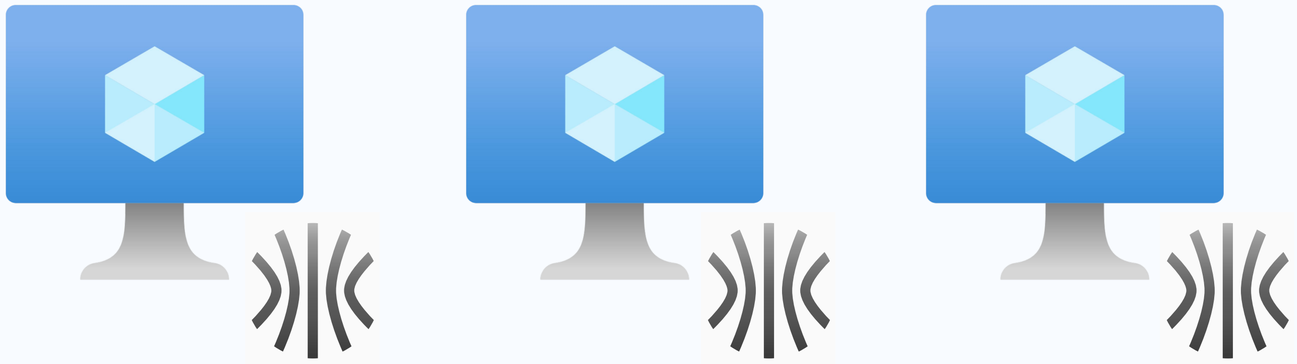
control-plane.yaml

worker.yaml

Первоначальный запуск кластера



Первоначальный запуск кластера



talosctl apply -f control-plane.yaml -n vm1

secrets.yaml

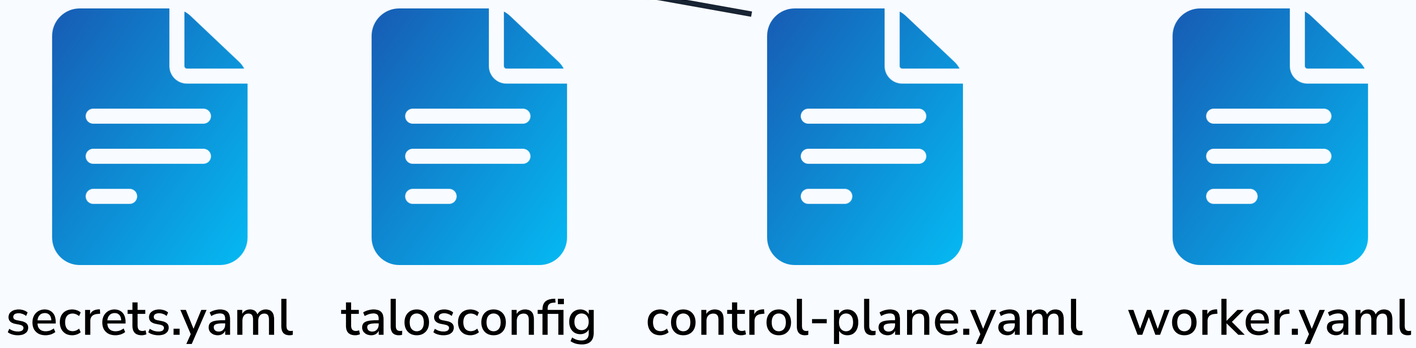
talosconfig

control-plane.yaml

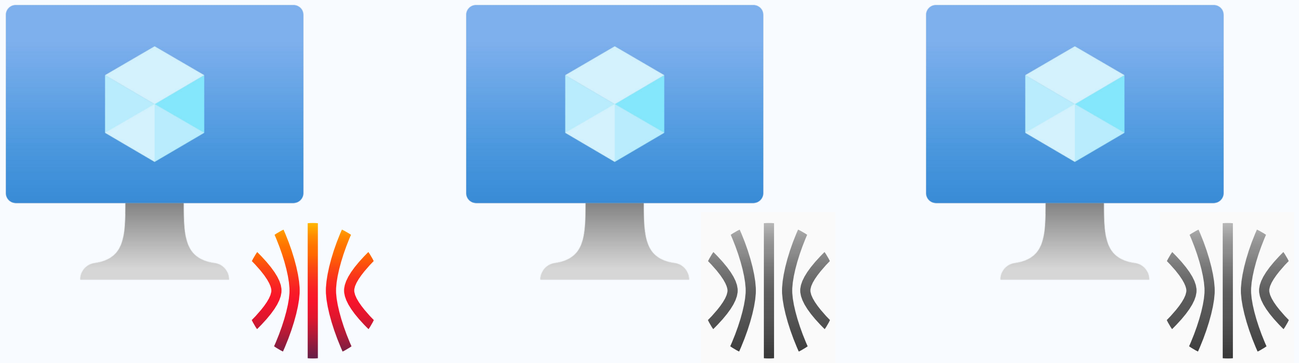
worker.yaml

Первоначальный запуск кластера

talosctl apply -f control-plane.yaml -n vm1

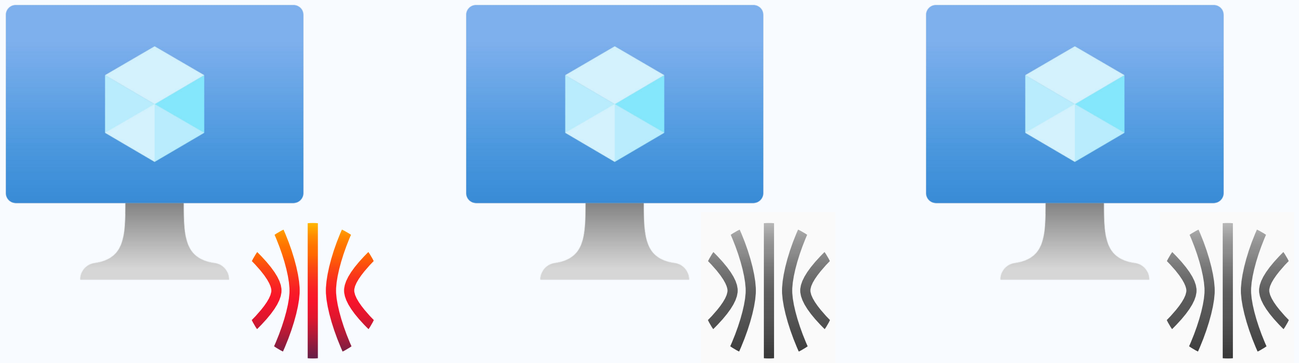


Первоначальный запуск кластера

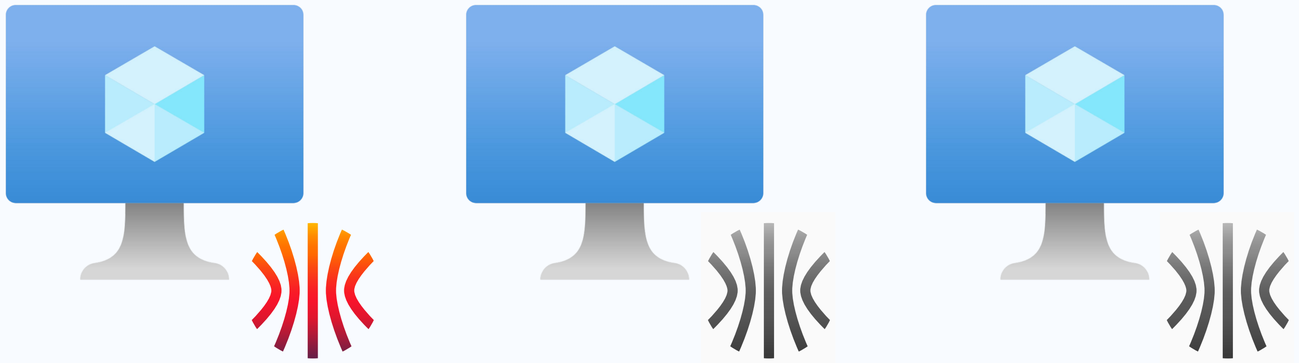


talosctl apply -f control-plane.yaml -n vm1

Первоначальный запуск кластера



Первоначальный запуск кластера



talosctl apply -f control-plane.yaml -n vm2

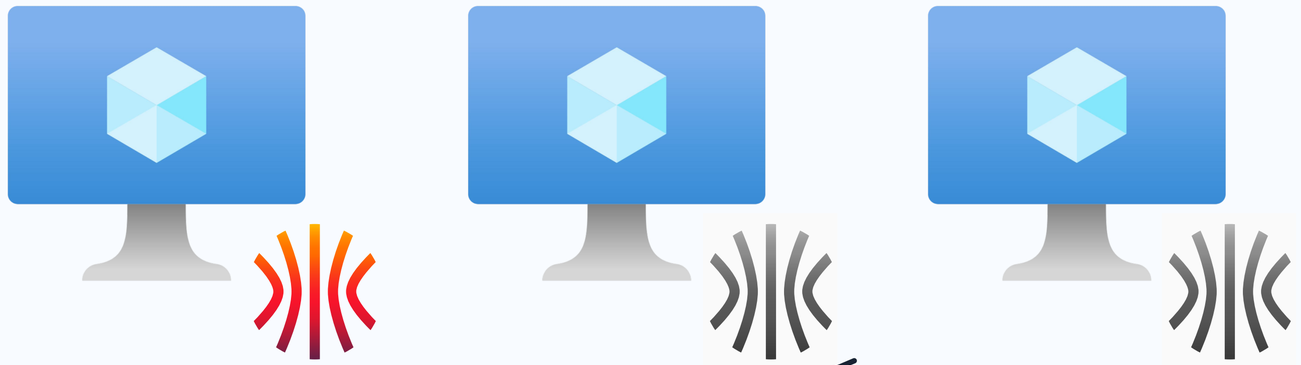
secrets.yaml

talosconfig

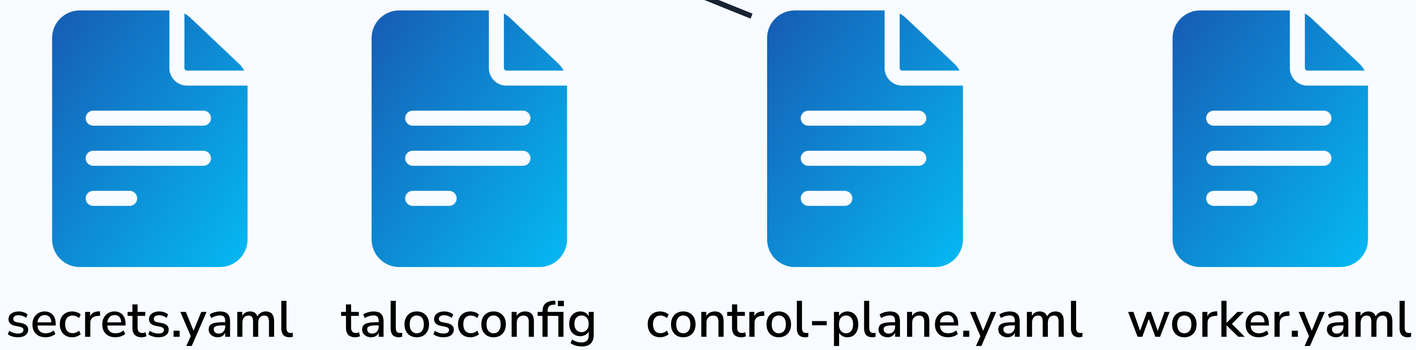
control-plane.yaml

worker.yaml

Первоначальный запуск кластера



talosctl apply -f control-plane.yaml -n vm2



Первоначальный запуск кластера

talosctl apply -f control-plane.yaml -n vm2

secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Первоначальный запуск кластера

Первоначальный запуск кластера

talosctl apply -f control-plane.yaml -n vm3

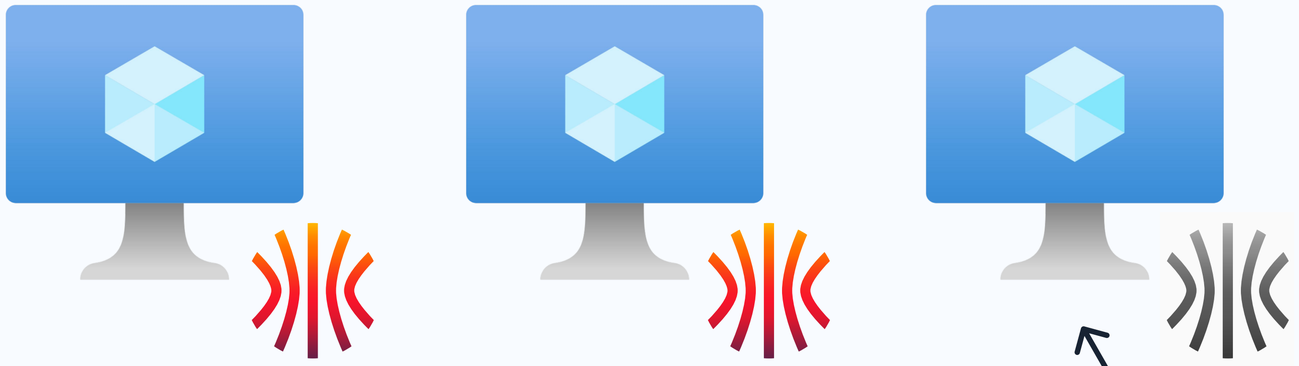
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Первоначальный запуск кластера



talosctl apply -f control-plane.yaml -n vm3

Первоначальный запуск кластера

talosctl apply -f control-plane.yaml -n vm3

secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Первоначальный запуск кластера

Первоначальный запуск кластера

talosctl bootstrap -n vm1

secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Первоначальный запуск кластера

talosctl bootstrap -n vm1

secrets.yaml talosconfig control-plane.yaml worker.yaml

Первоначальный запуск кластера

talosctl bootstrap -n vm1

secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Первоначальный запуск кластера

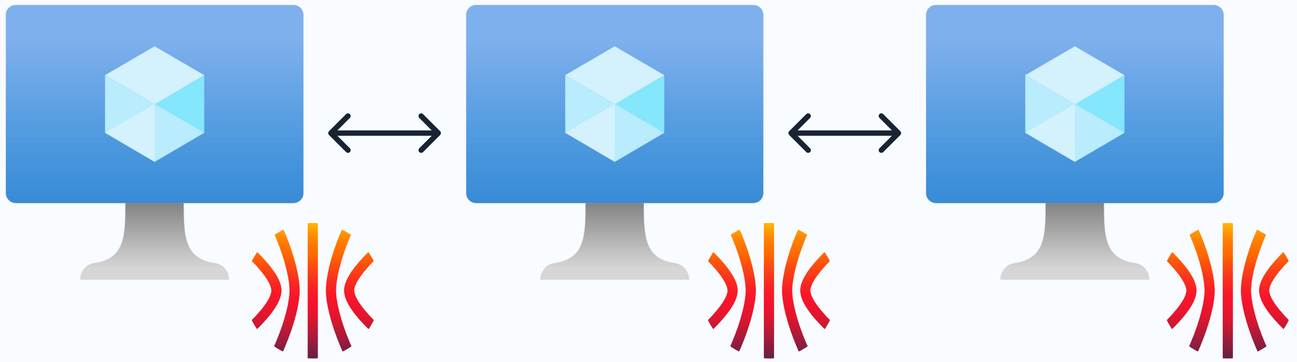
secrets.yaml

talosconfig

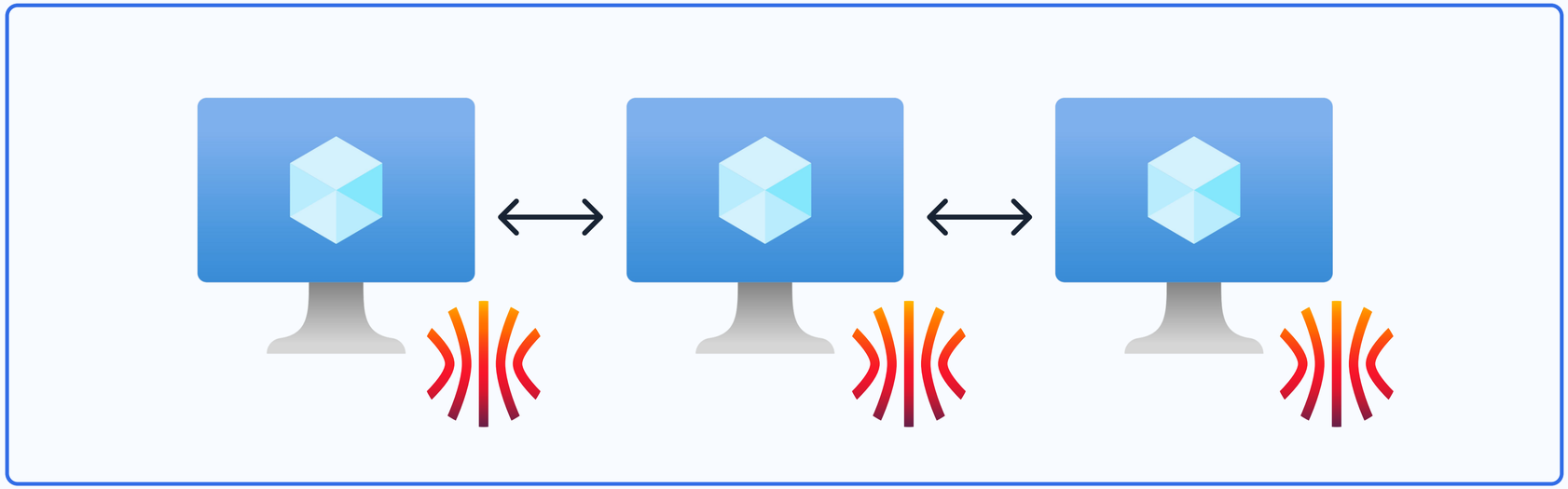
control-plane.yaml

worker.yaml

Первоначальный запуск кластера



Первоначальный запуск кластера



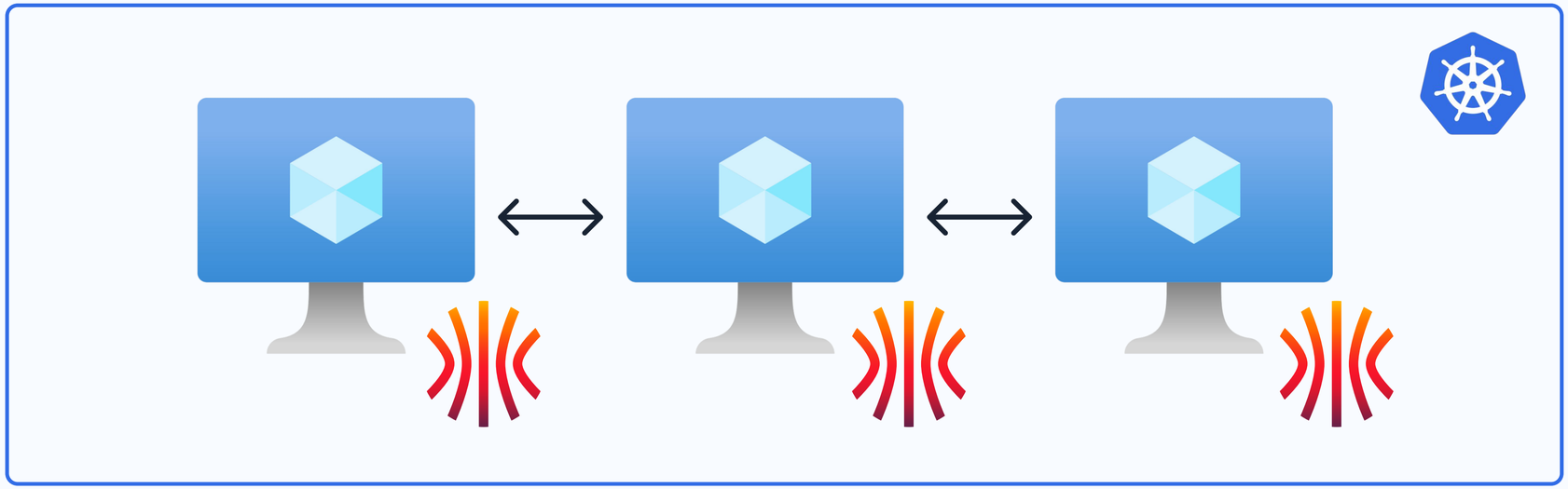
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Первоначальный запуск кластера



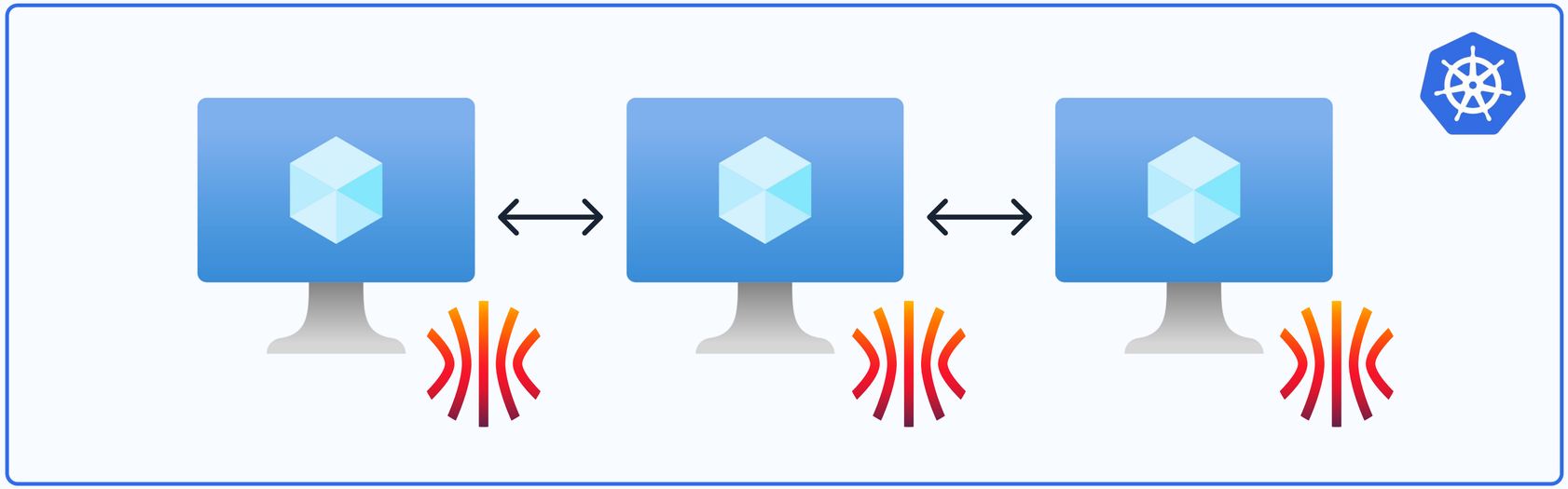
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



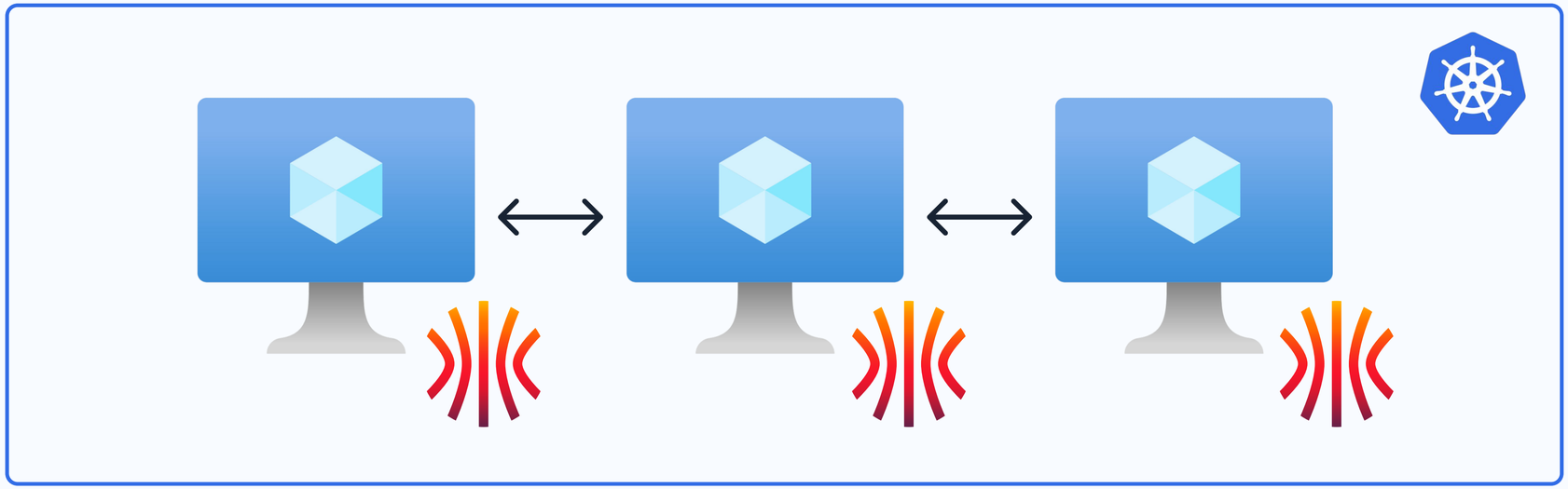
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



talosctl apply -f control-plane.yaml -n vm1

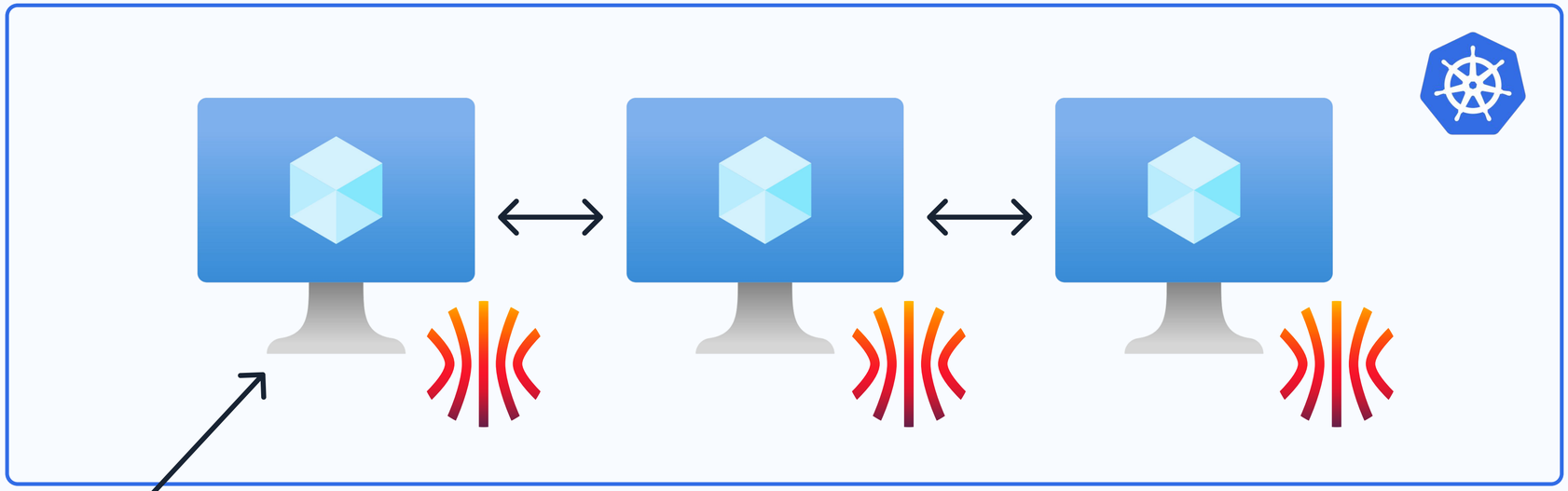
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



talosctl apply -f control-plane.yaml -n vm1

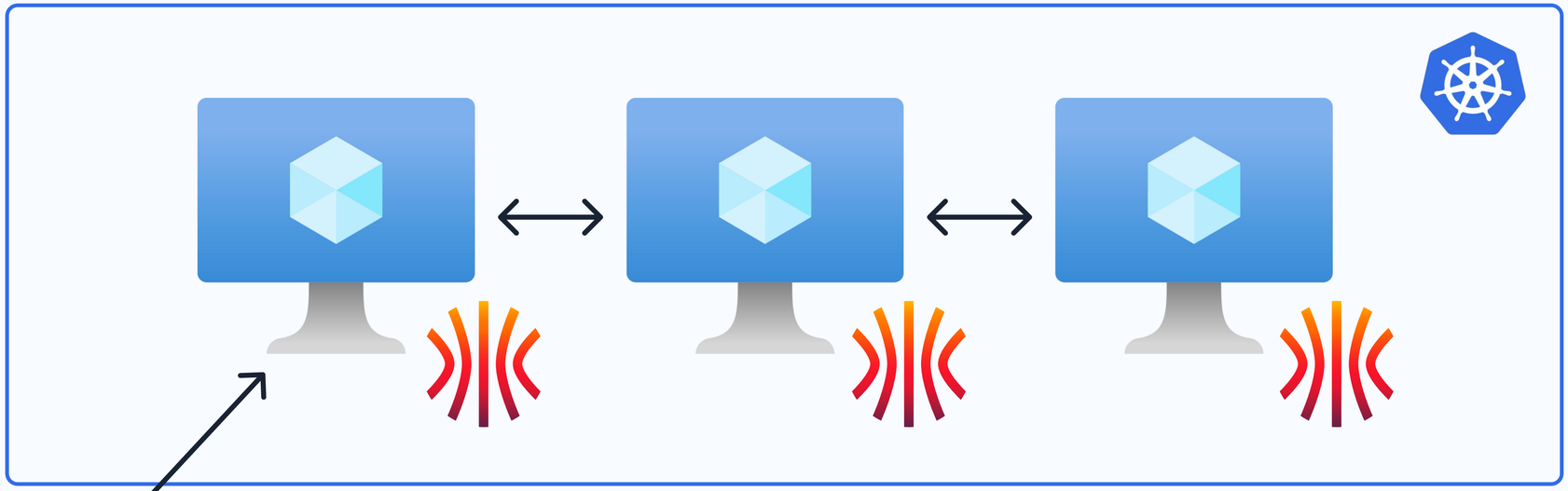
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



talosctl upgrade -n vm1

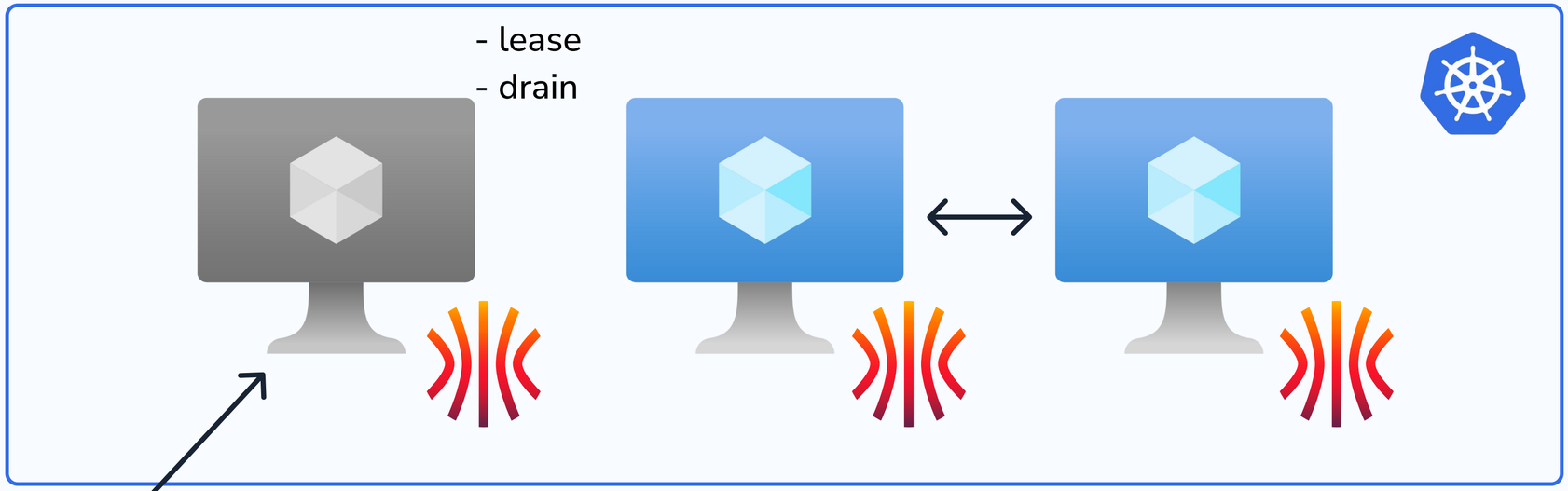
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



talosctl upgrade -n vm1

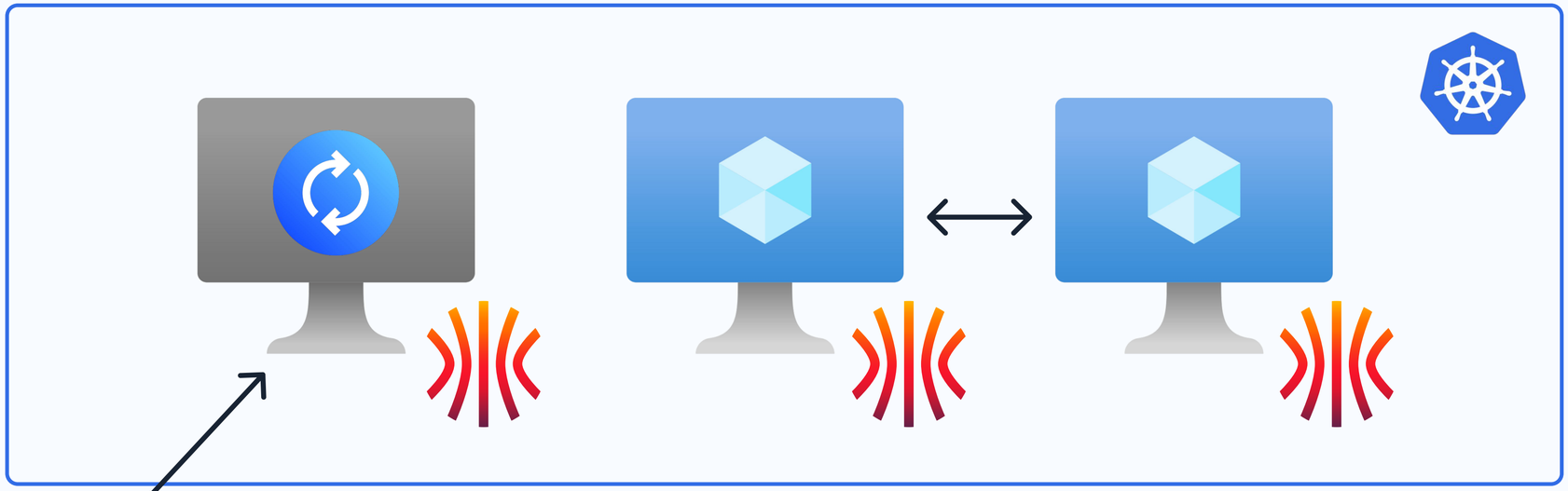

secrets.yaml


talosconfig


control-plane.yaml


worker.yaml

Обновление узлов кластера



talosctl upgrade -n vm1

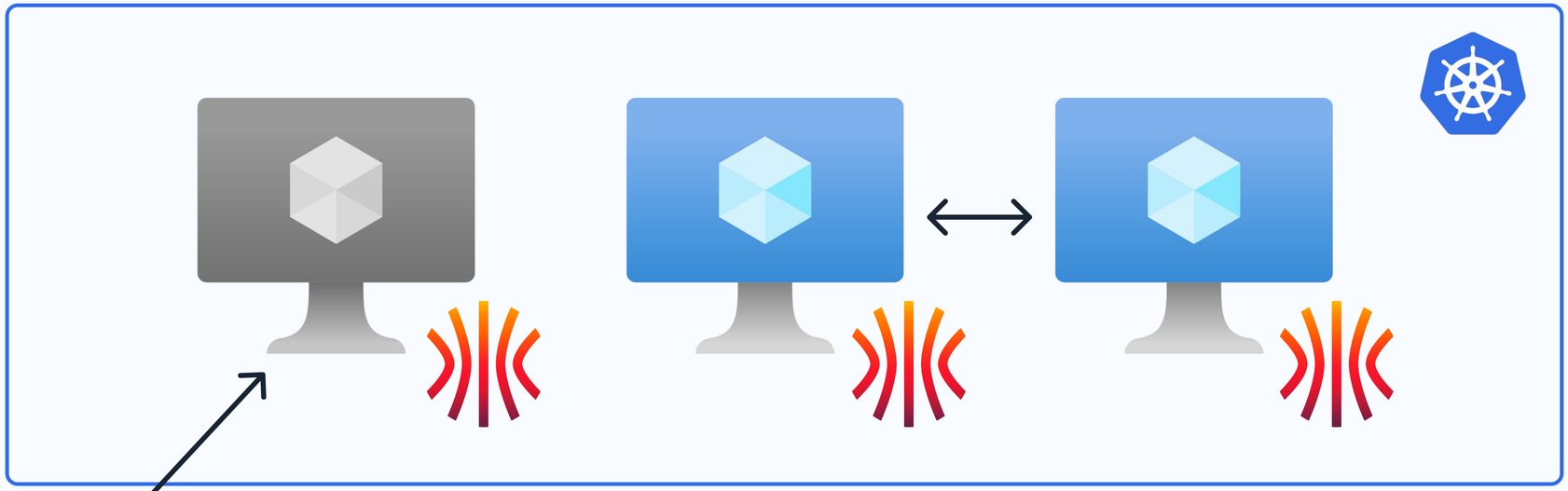
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



talosctl upgrade -n vm1

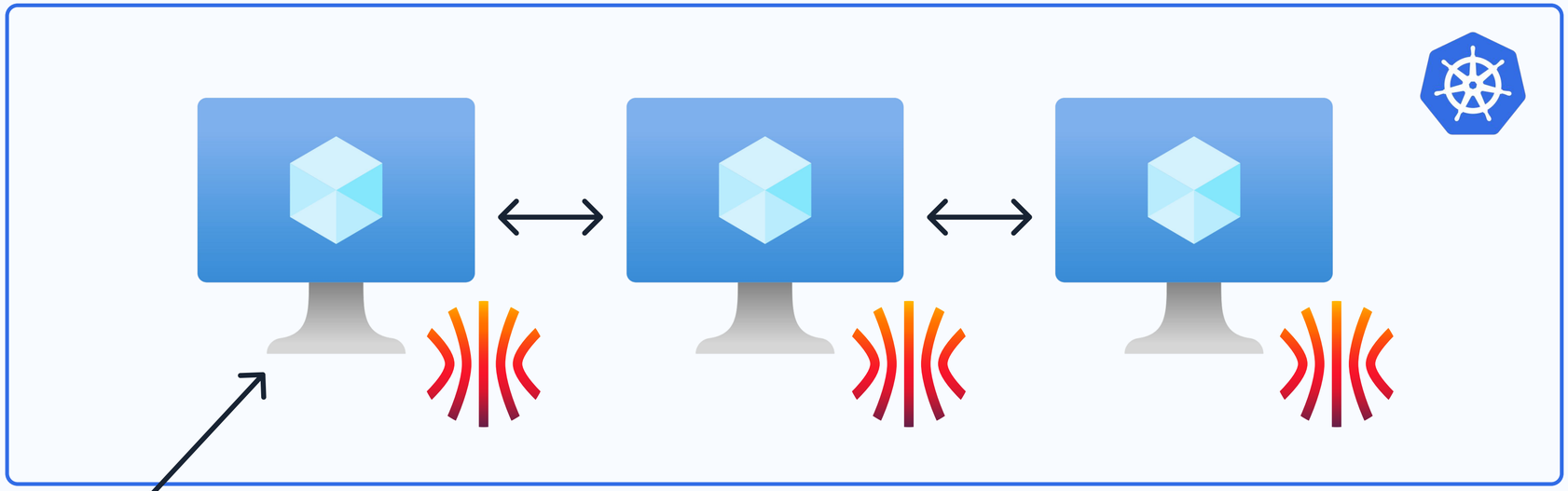
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



talosctl upgrade -n vm1

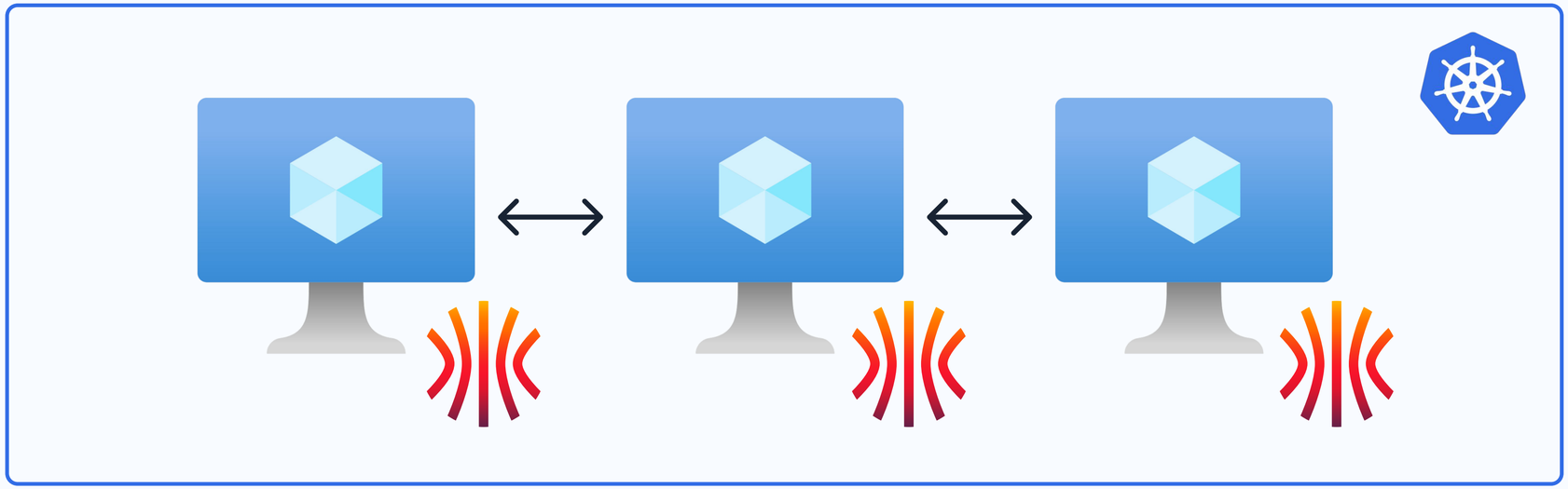
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



talosctl upgrade -n vm1

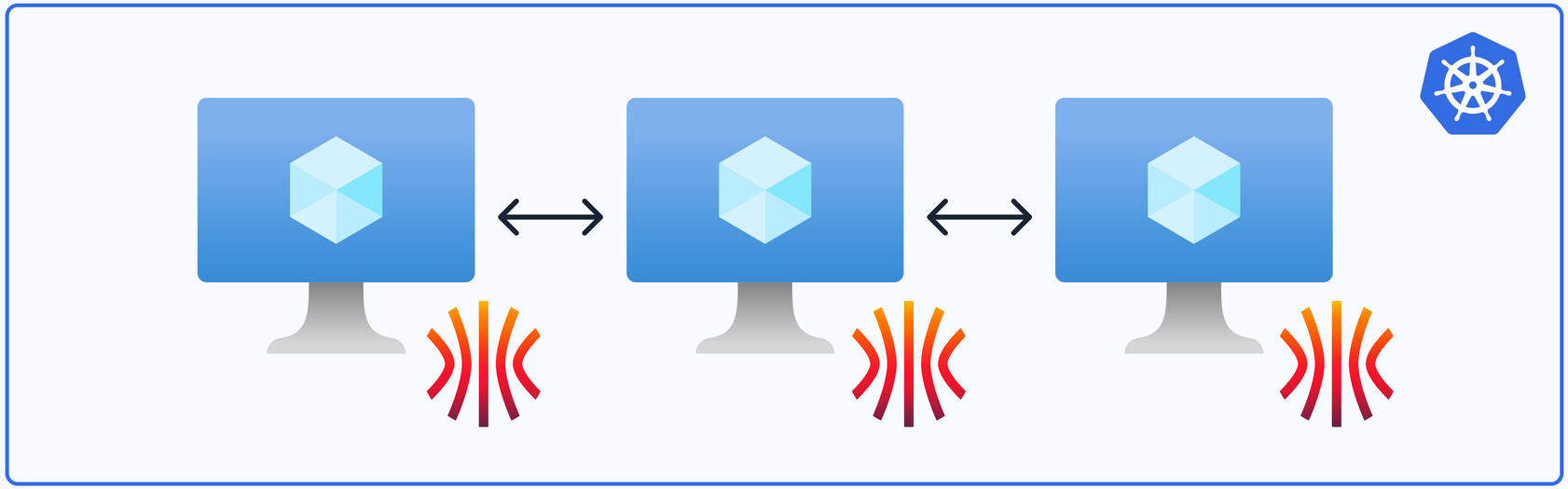
secrets.yaml

talosconfig

control-plane.yaml

worker.yaml

Обновление узлов кластера



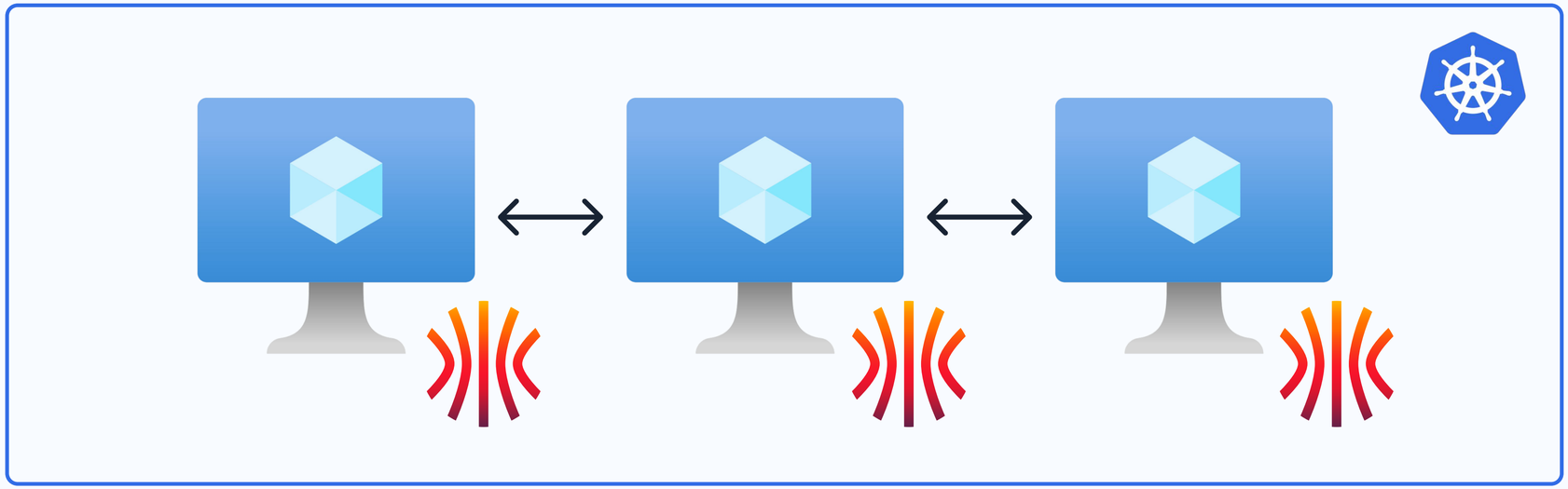
secrets.yaml

talosconfig

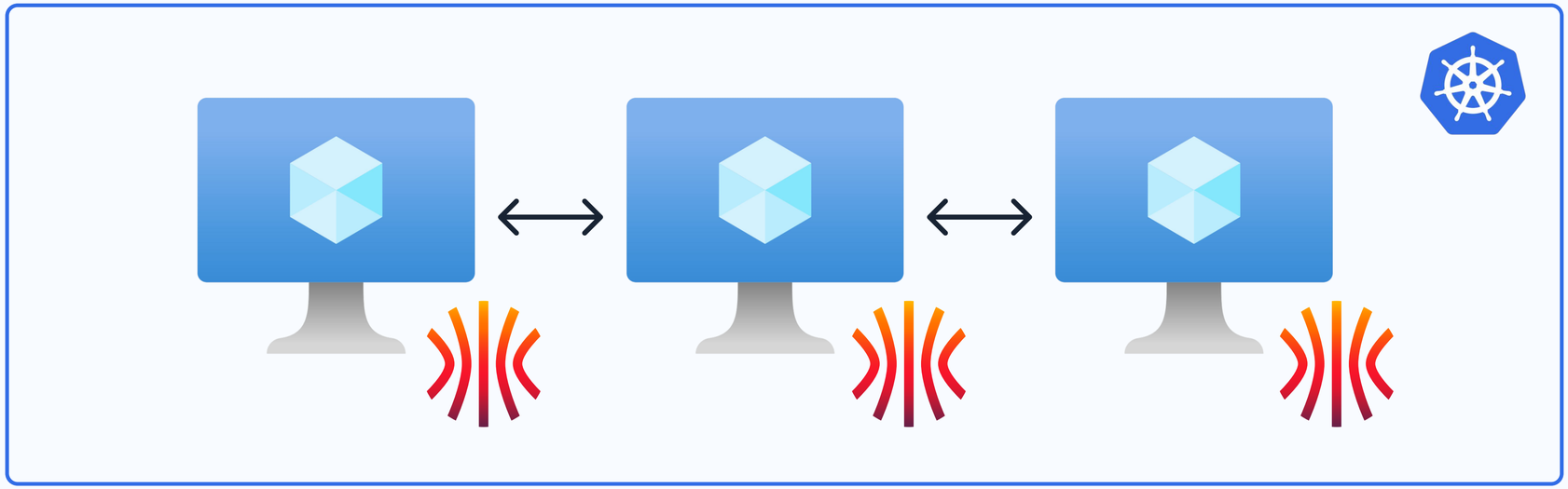
control-plane.yaml

worker.yaml

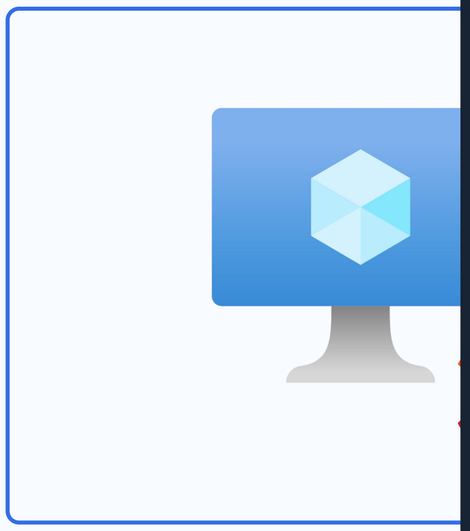
Обновление узлов кластера



Доступ в кластер

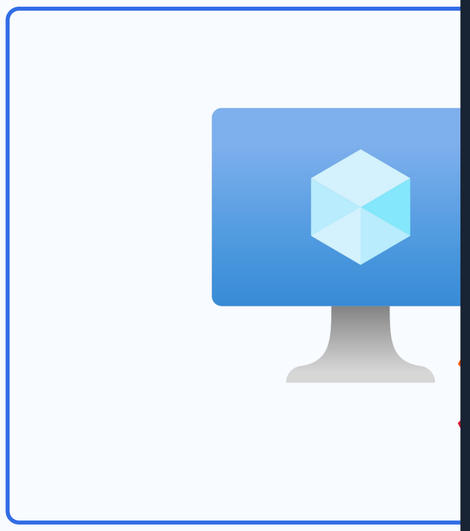


Доступ в кластер



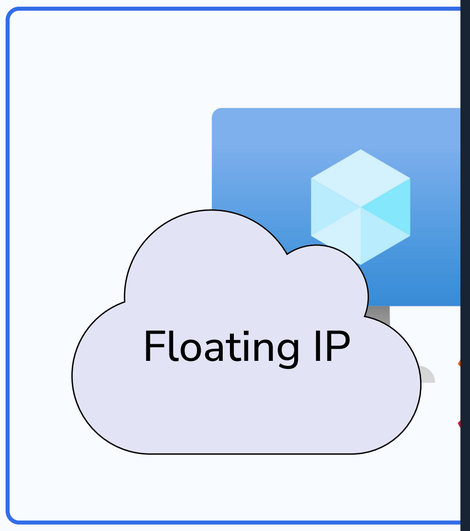
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

Доступ в кластер



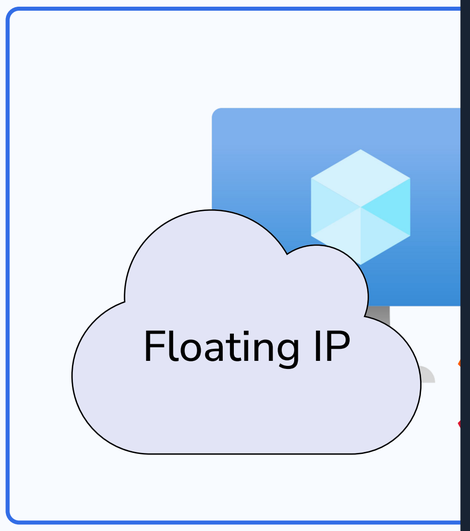
```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

Доступ в кластер




```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

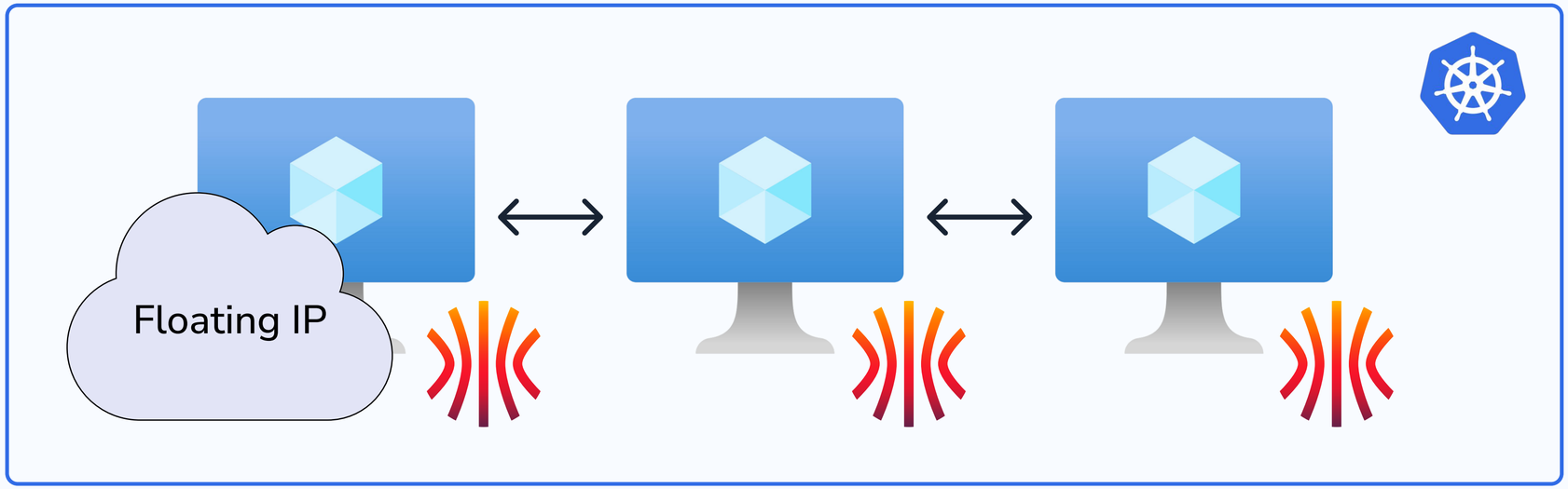
Доступ в кластер



kubeconfig

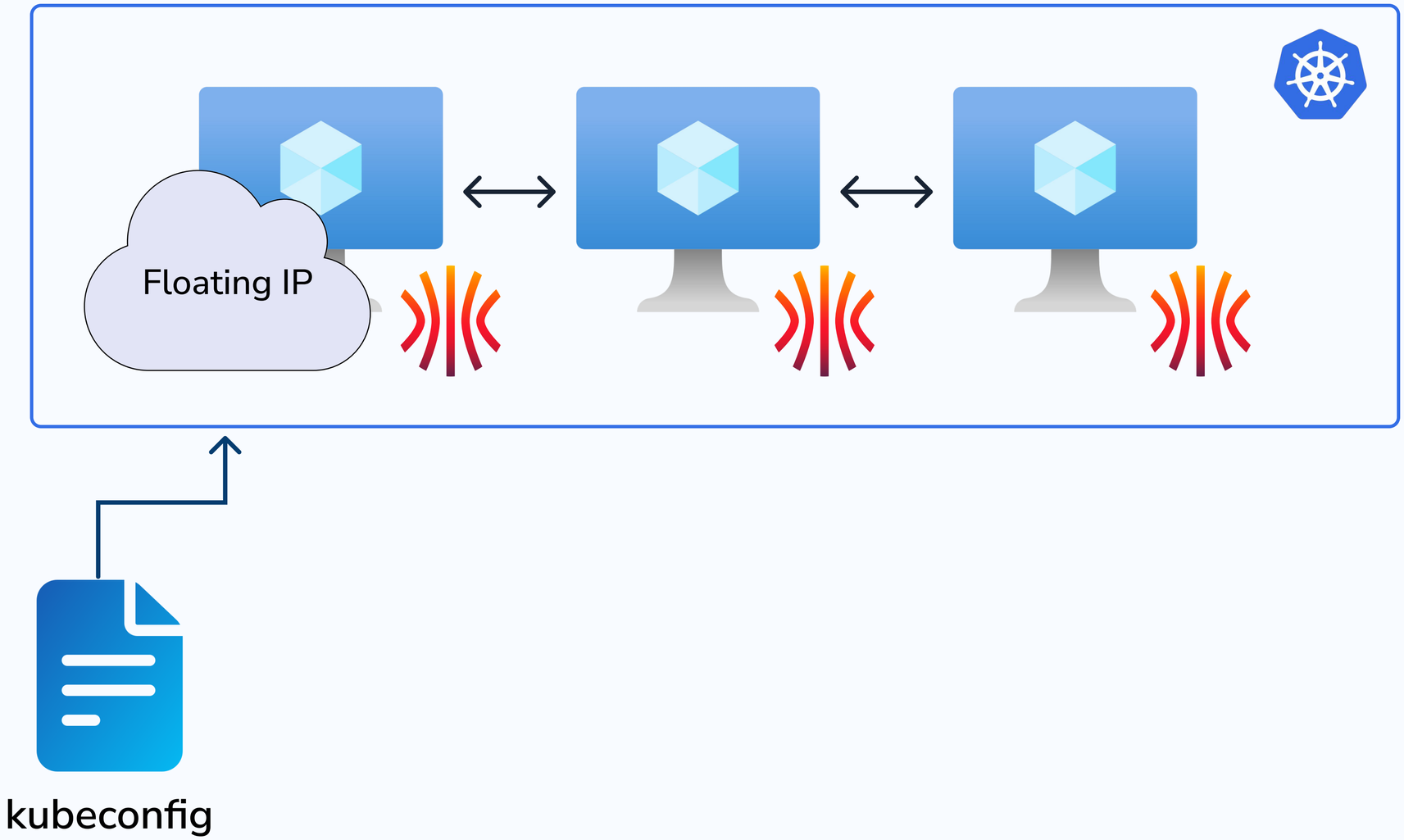
```
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: {}
```

Доступ в кластер

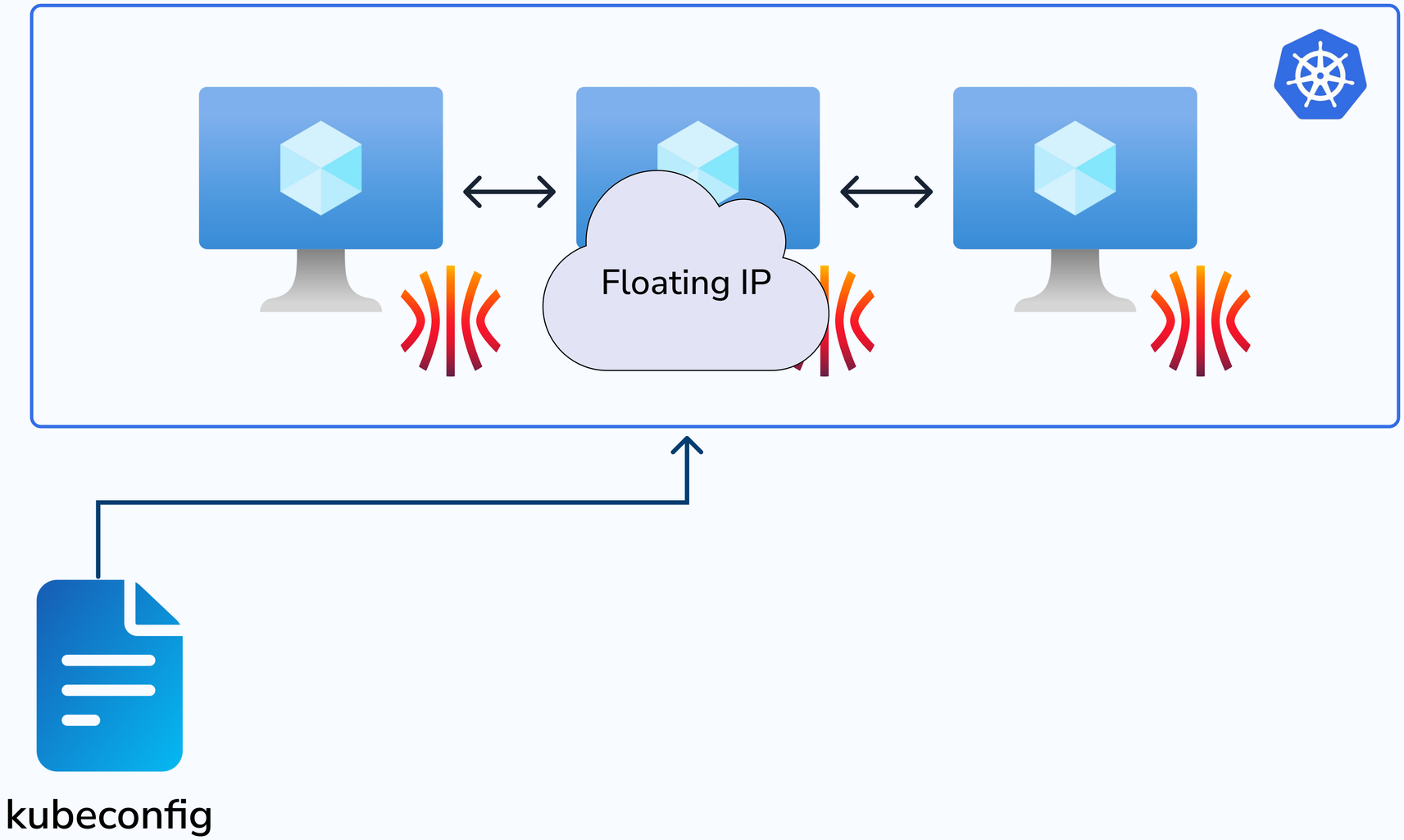


kubeconfig

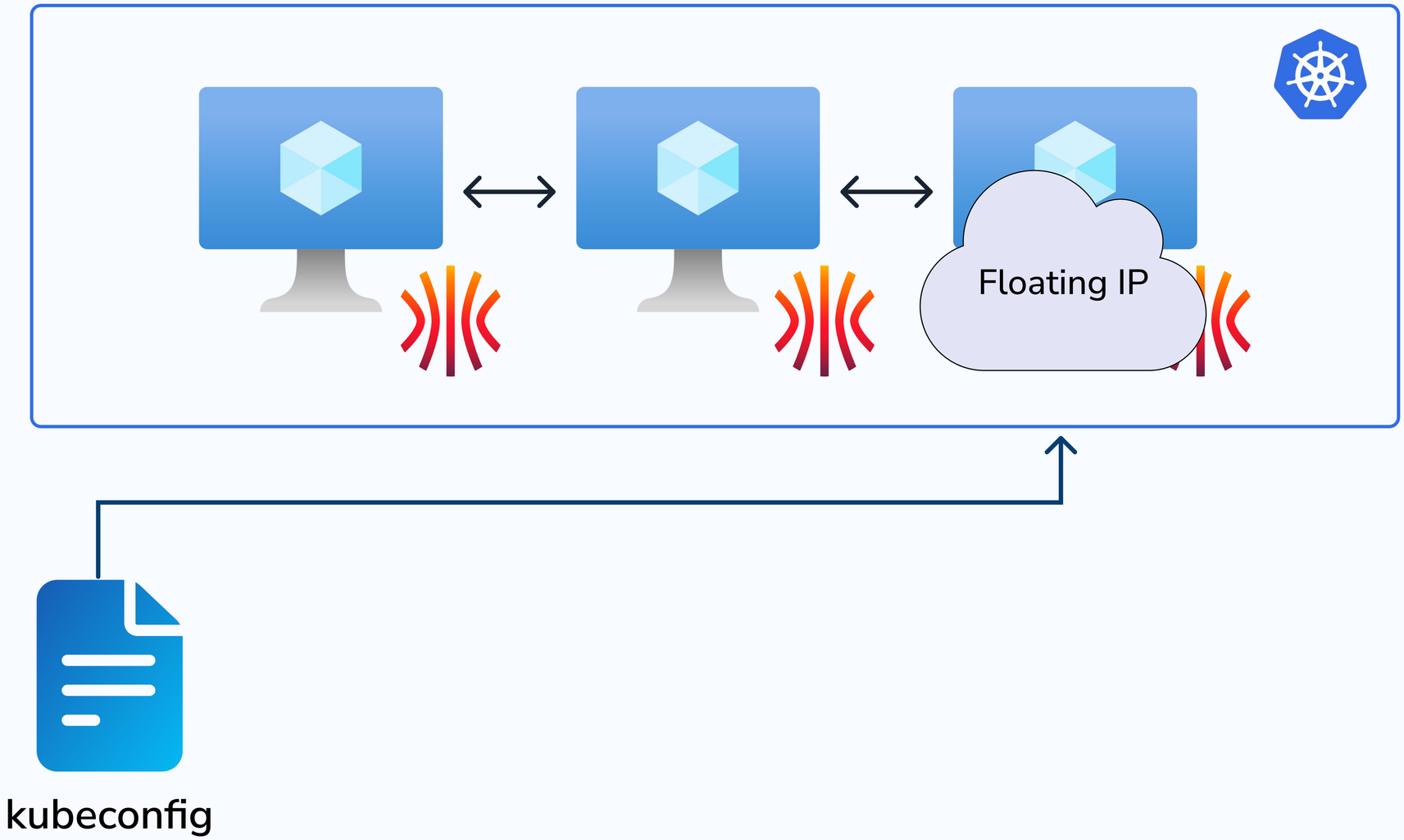
Доступ в кластер



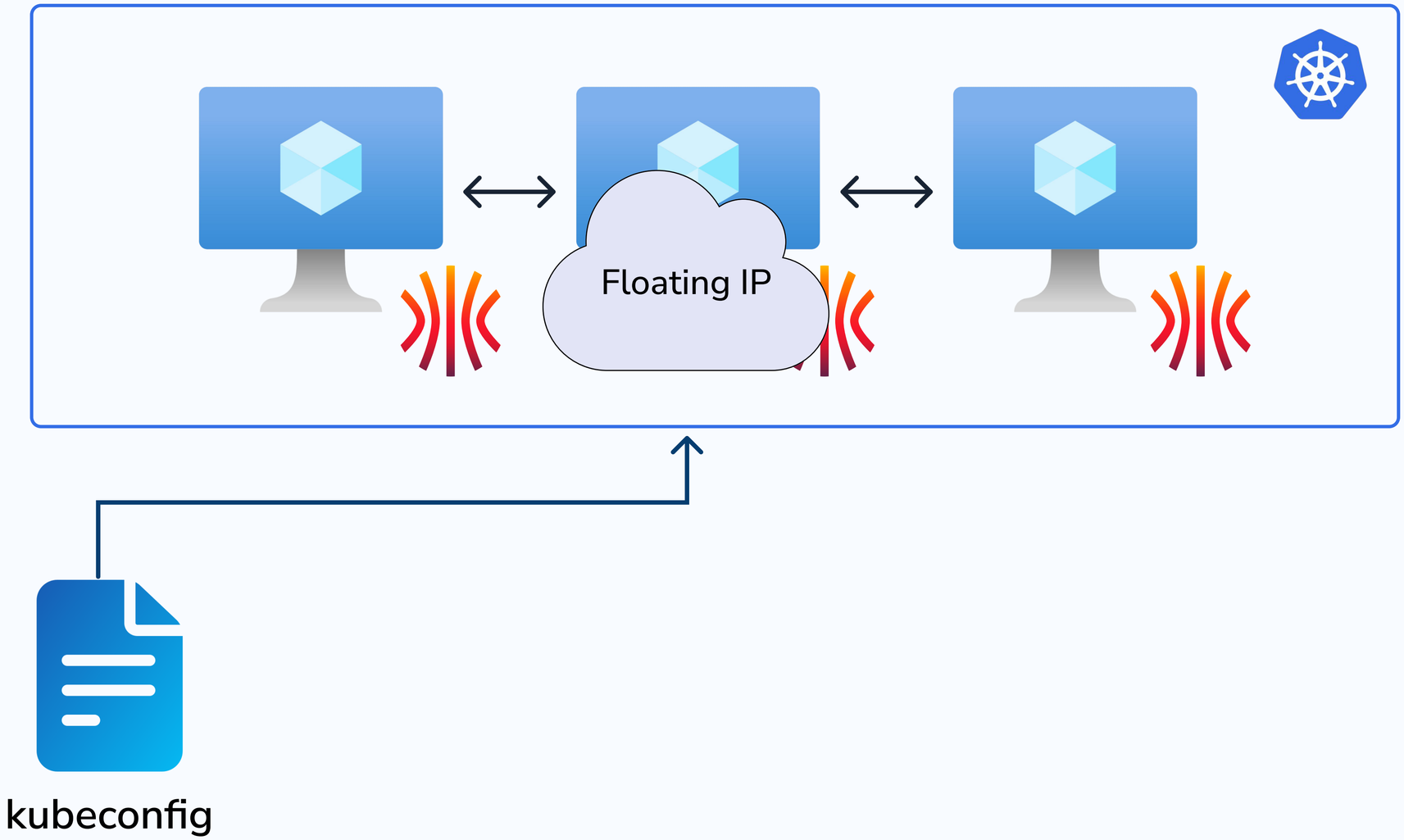
Доступ в кластер

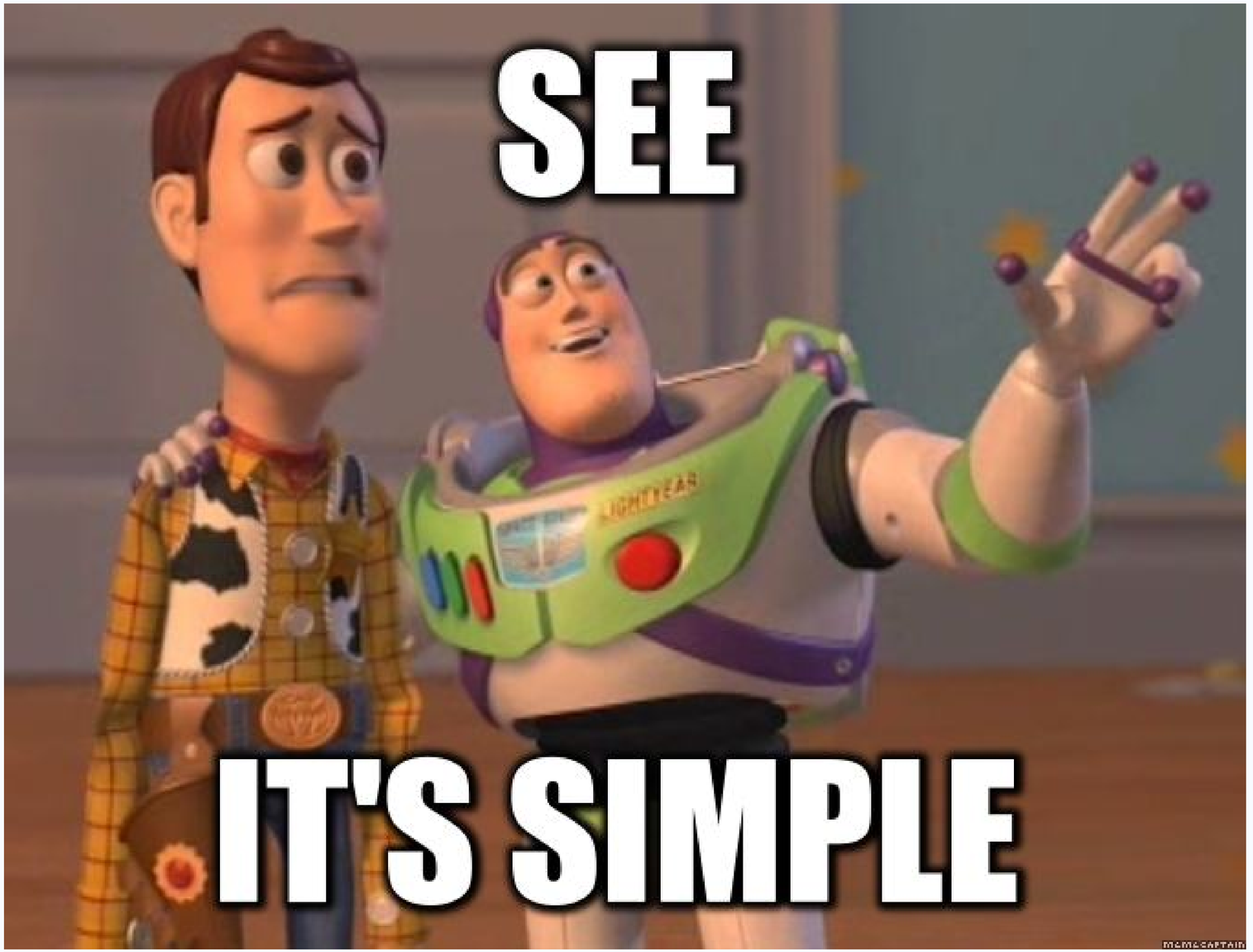


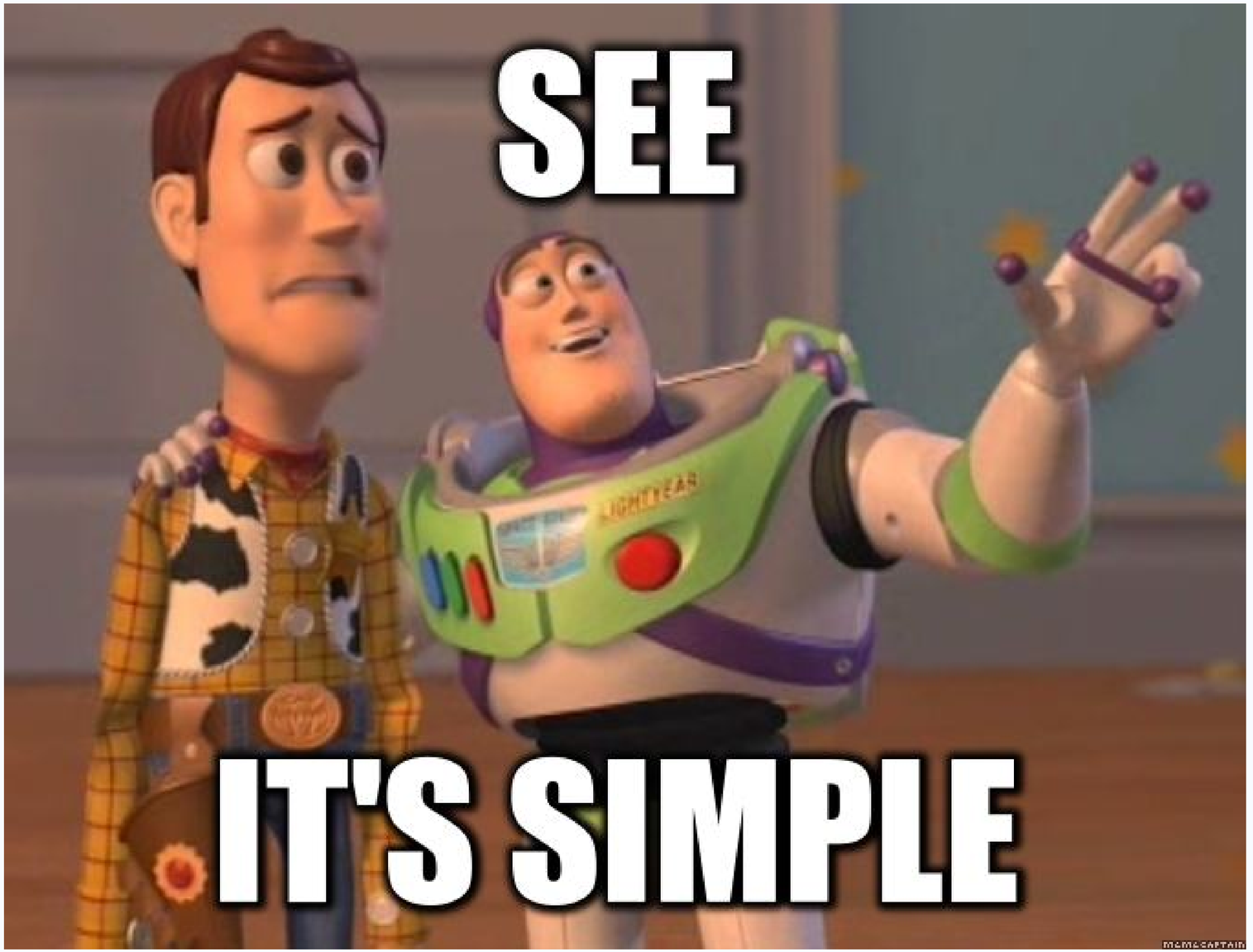
Доступ в кластер



Доступ в кластер




SEE
IT'S SIMPLE

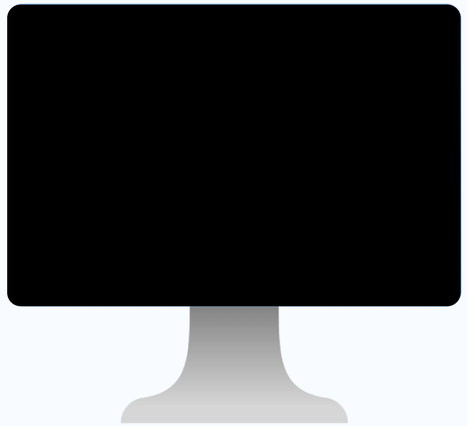


Talos-bootstrap: наша автоматизация

talos-bootstrap

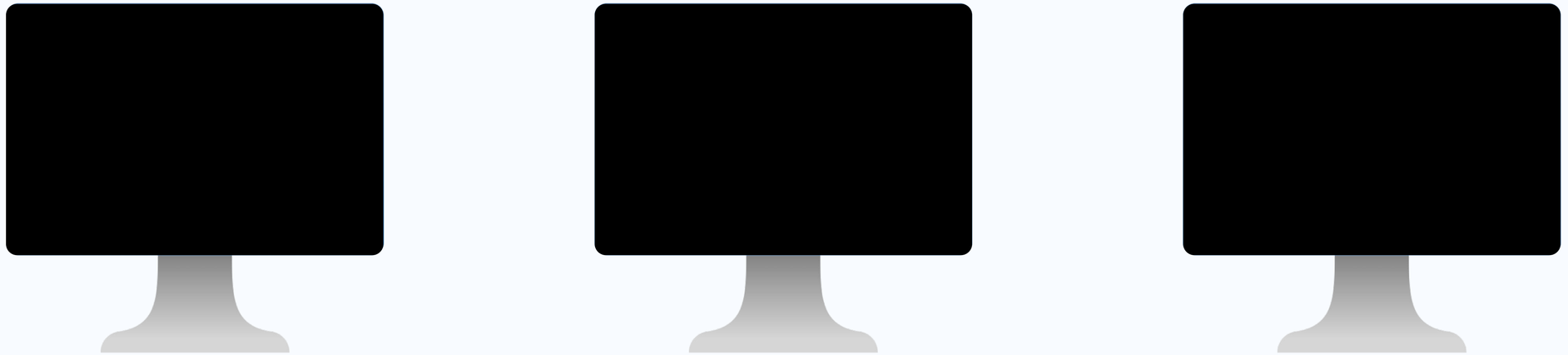
developed by Ænix

talos-bootstrap

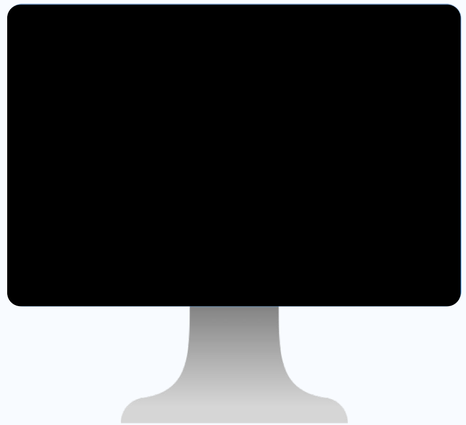
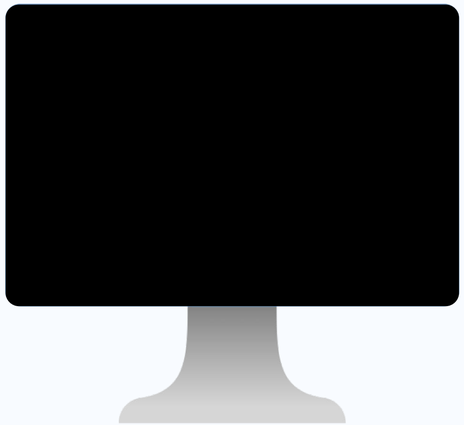
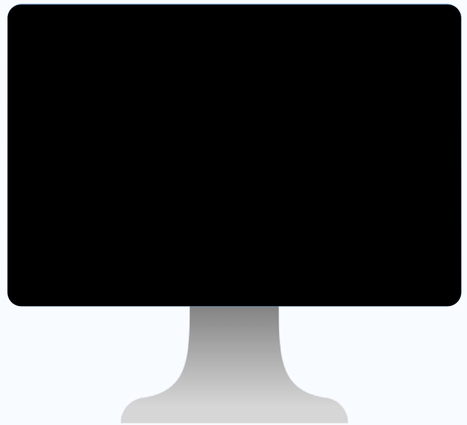


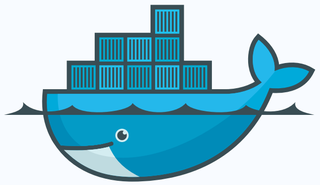
talos-bootstrap

talos-bootstrap



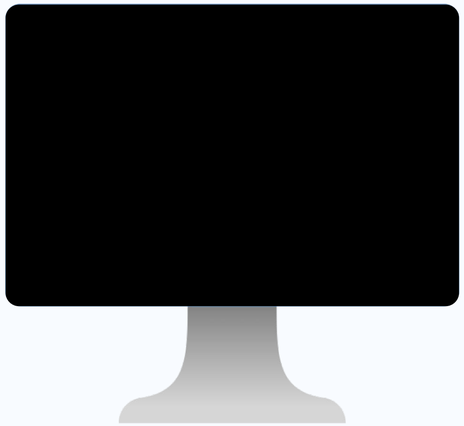
talos-bootstrap

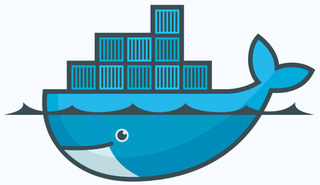






DHCP-server
PXE-server

talos-bootstrap

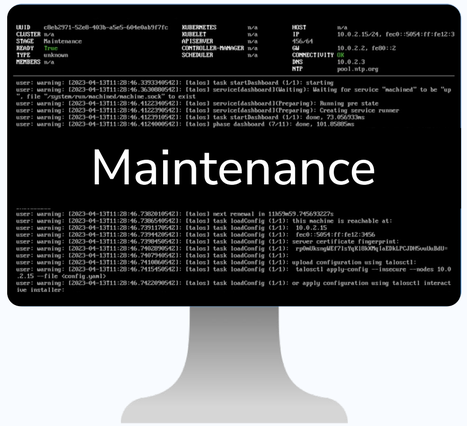



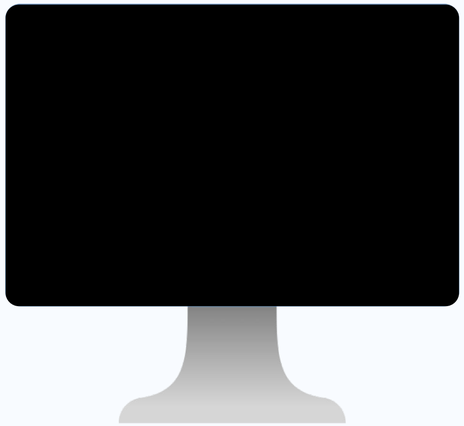



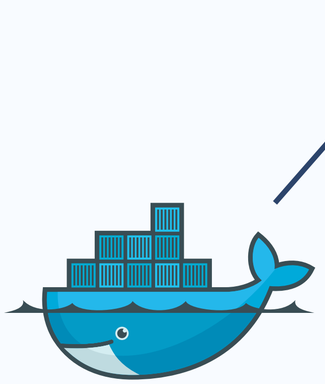
DHCP-server
PXE-server

talos-bootstrap

talos-bootstrap



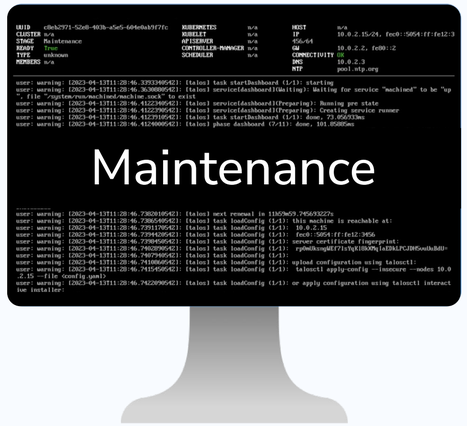


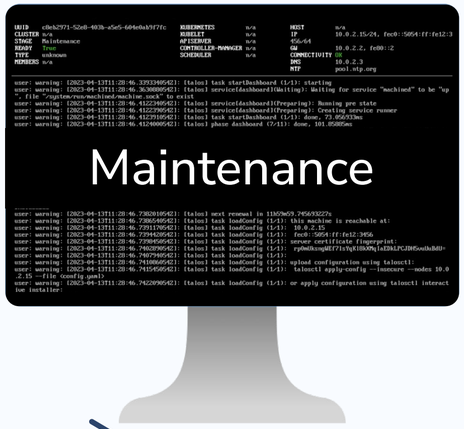


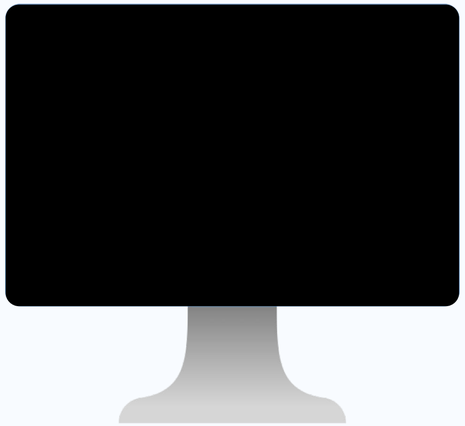
DHCP-server
PXE-server

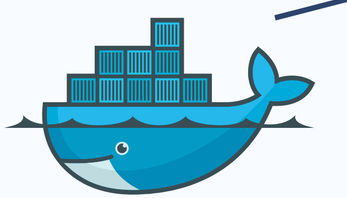
talos-bootstrap

talos-bootstrap





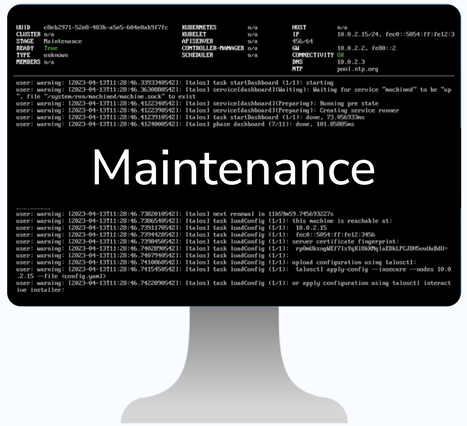


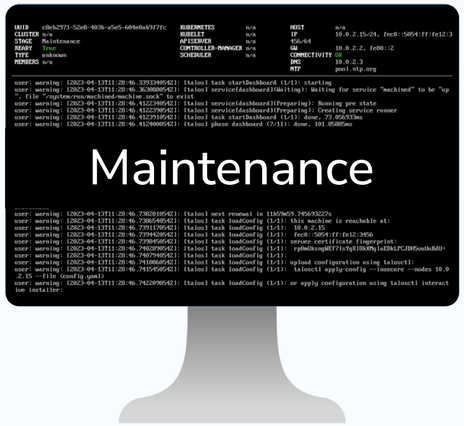


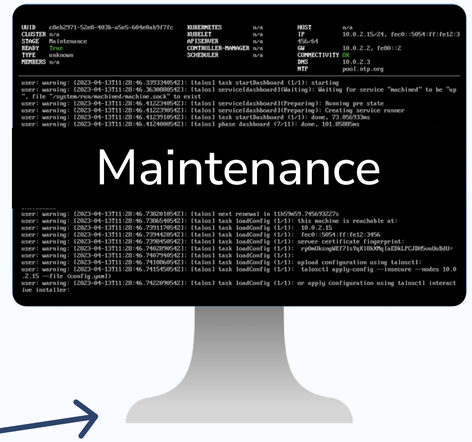
DHCP-server
PXE-server

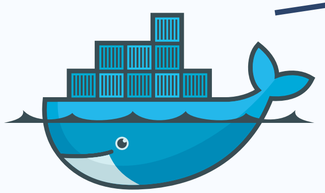
talos-bootstrap

talos-bootstrap





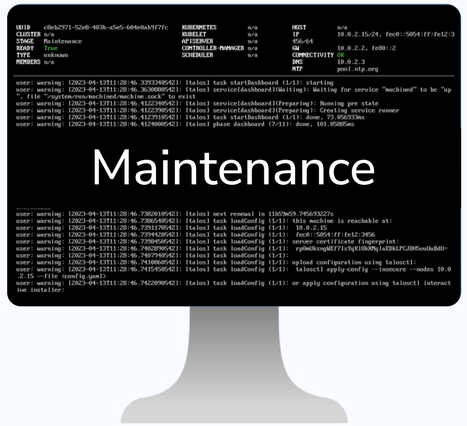


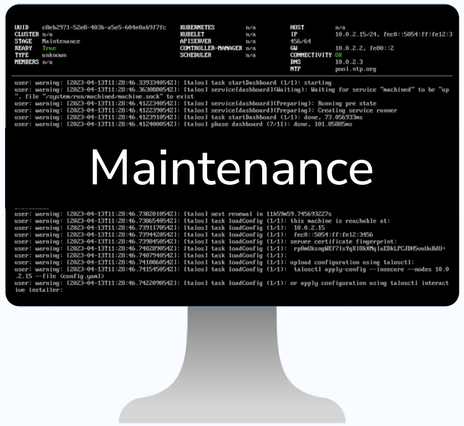


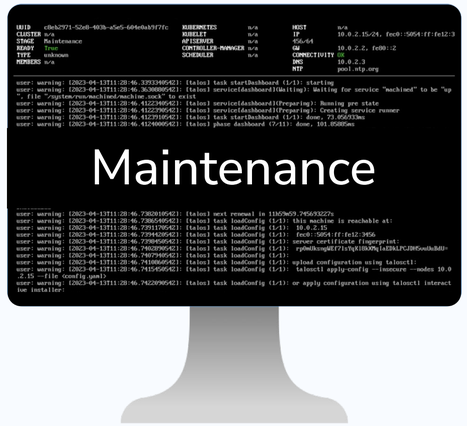
DHCP-server
PXE-server

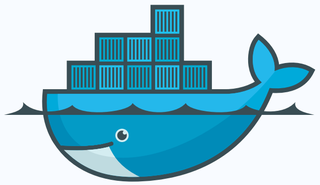
talos-bootstrap

talos-bootstrap





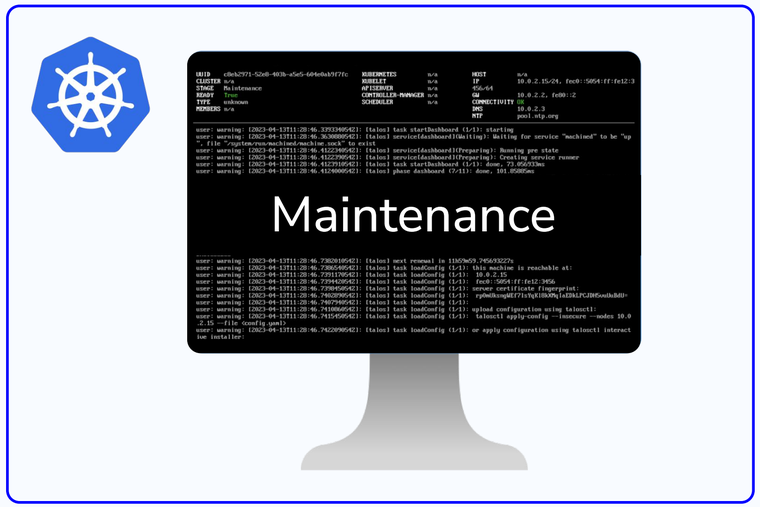


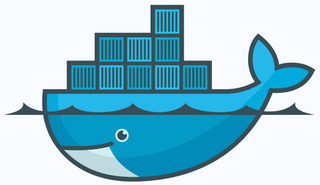


DHCP-server
PXE-server

talos-bootstrap

talos-bootstrap

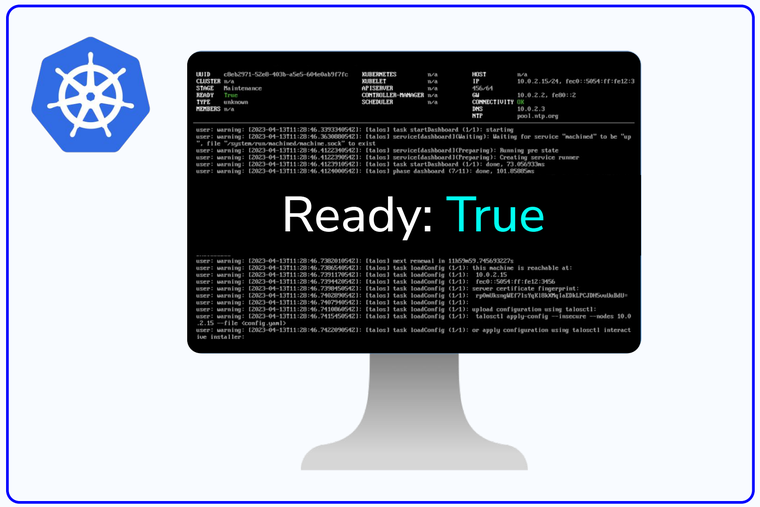


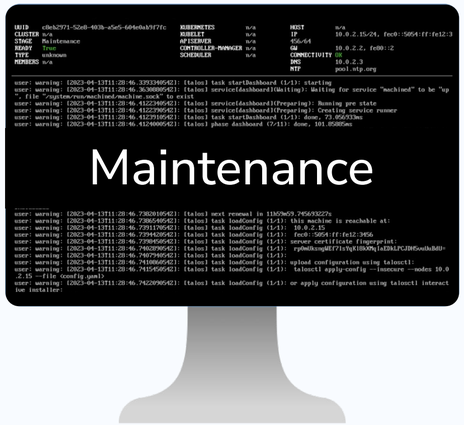


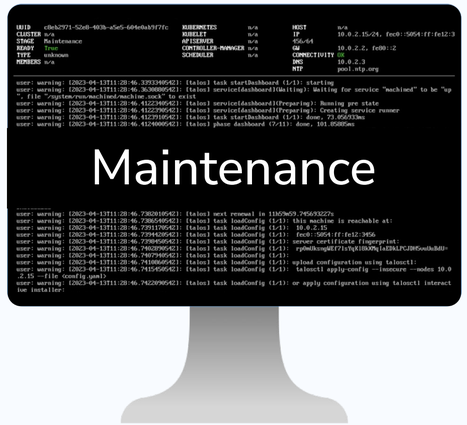
DHCP-server
PXE-server

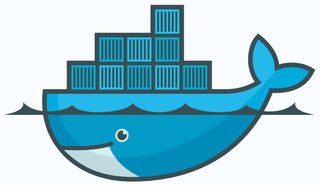
talos-bootstrap

talos-bootstrap





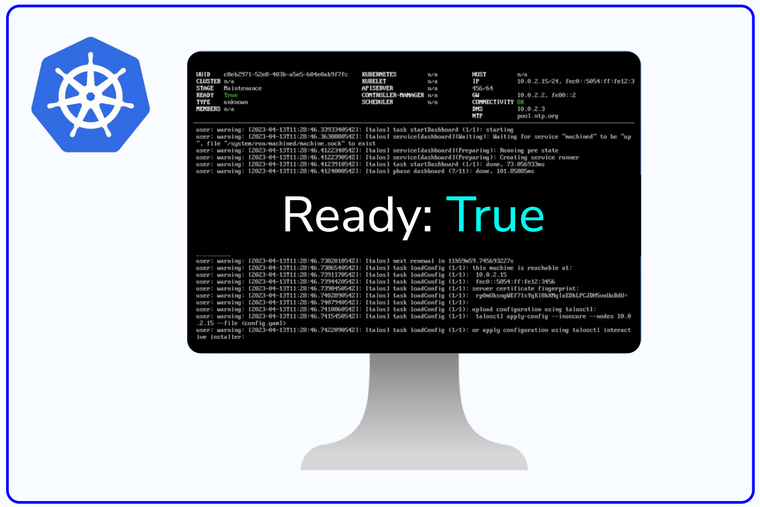


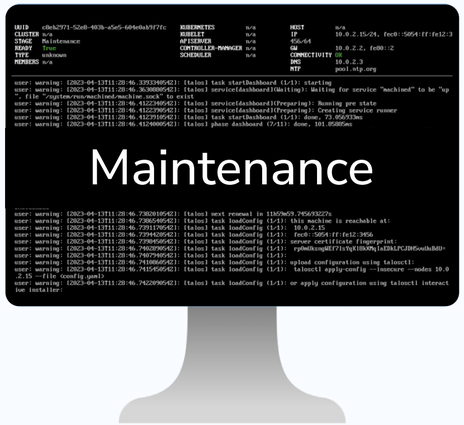


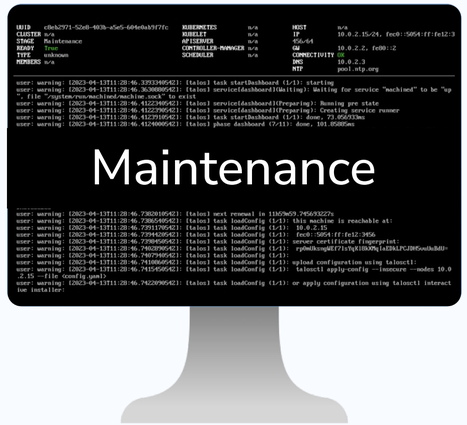
DHCP-server
PXE-server

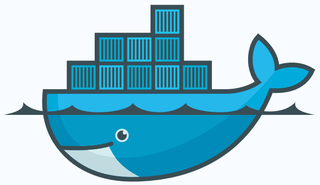
talos-bootstrap

talos-bootstrap





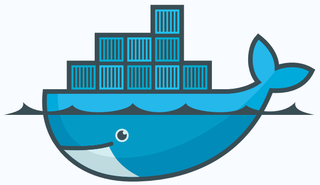




DHCP-server
PXE-server

talos-bootstrap

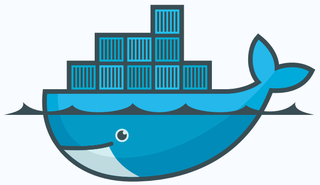
talos-bootstrap



DHCP-server
PXE-server

talos-bootstrap

talos-bootstrap



DHCP-server
PXE-server

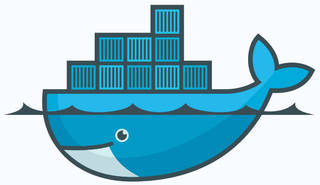
talos-bootstrap

talos-bootstrap

DHCP-server
PXE-server

talos-bootstrap

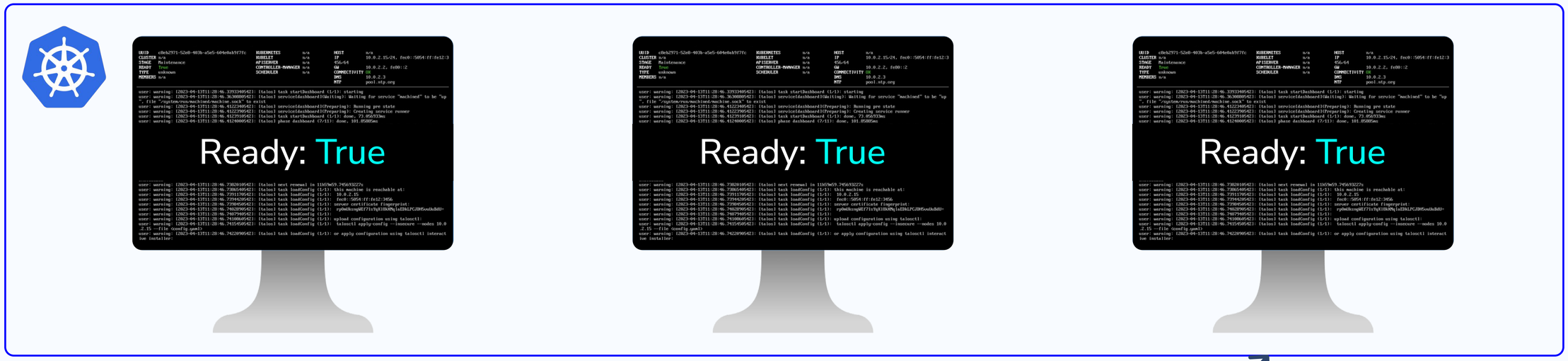
talos-bootstrap



DHCP-server
PXE-server

talos-bootstrap

talos-bootstrap

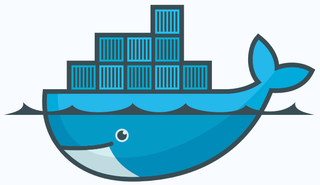


DHCP-server
PXE-server

talos-bootstrap

talos-bootstrap

How does it work?

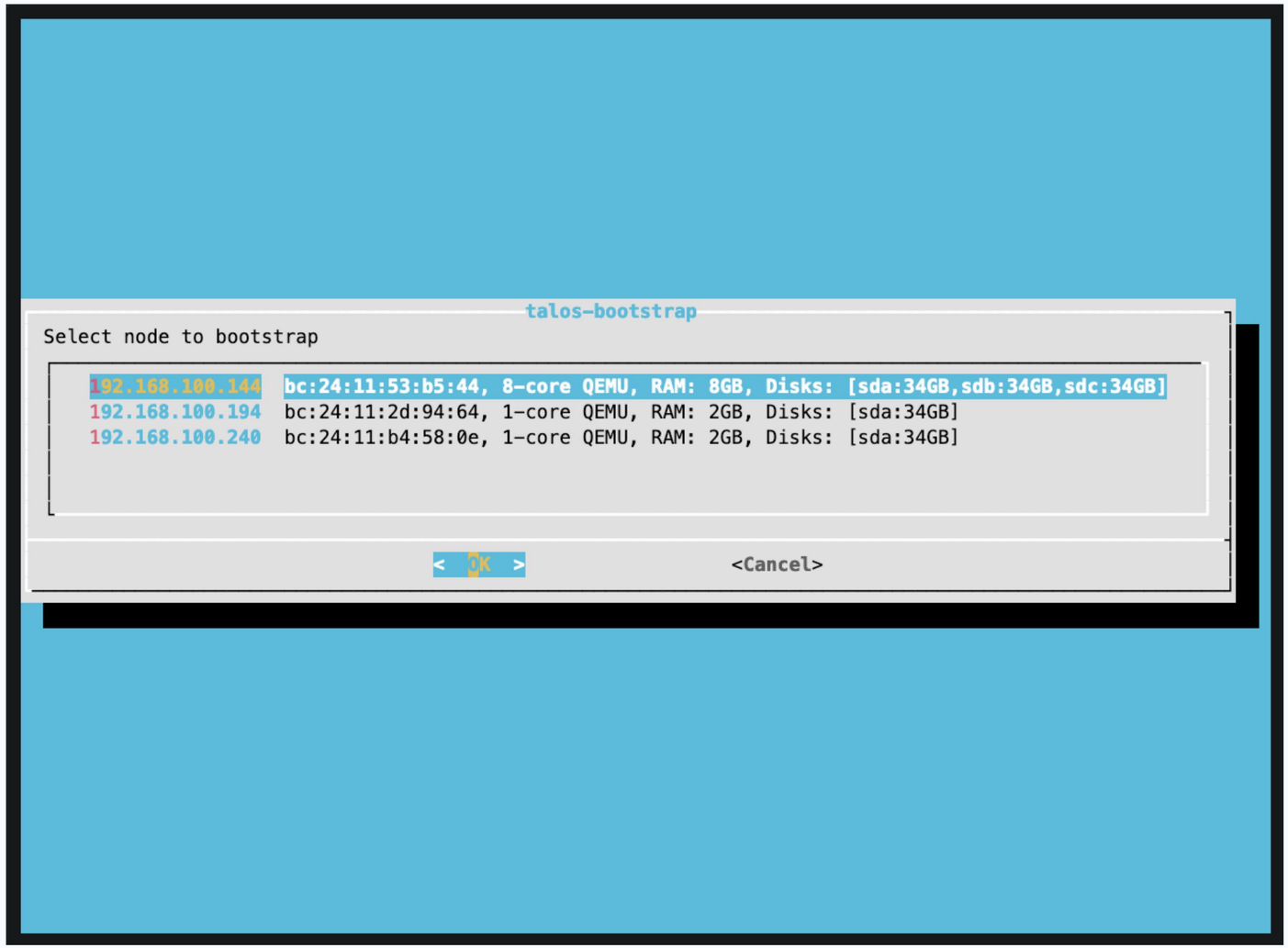


DHCP-server
PXE-server

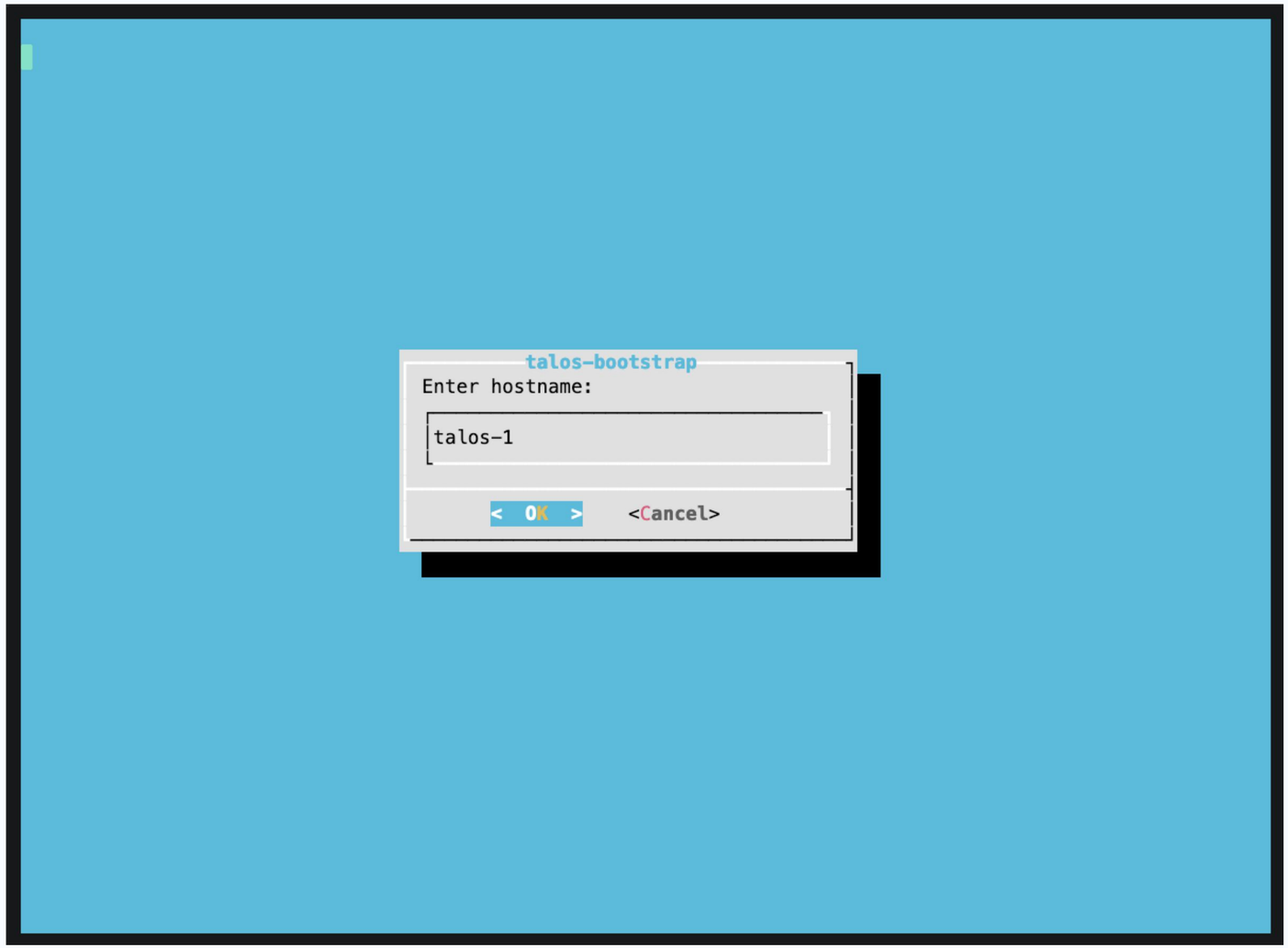
talos-bootstrap

talos-bootstrap

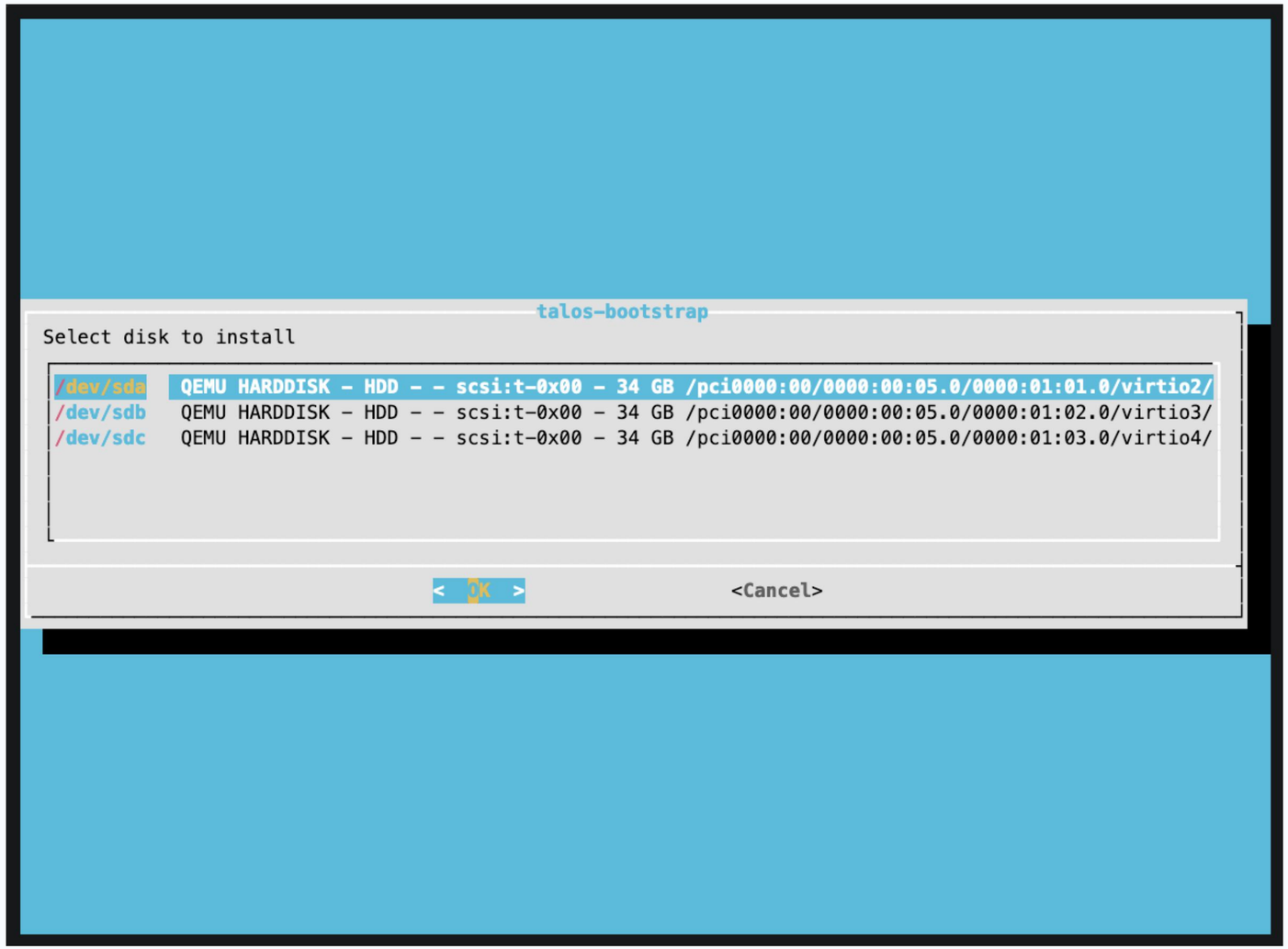
talos-bootstrap



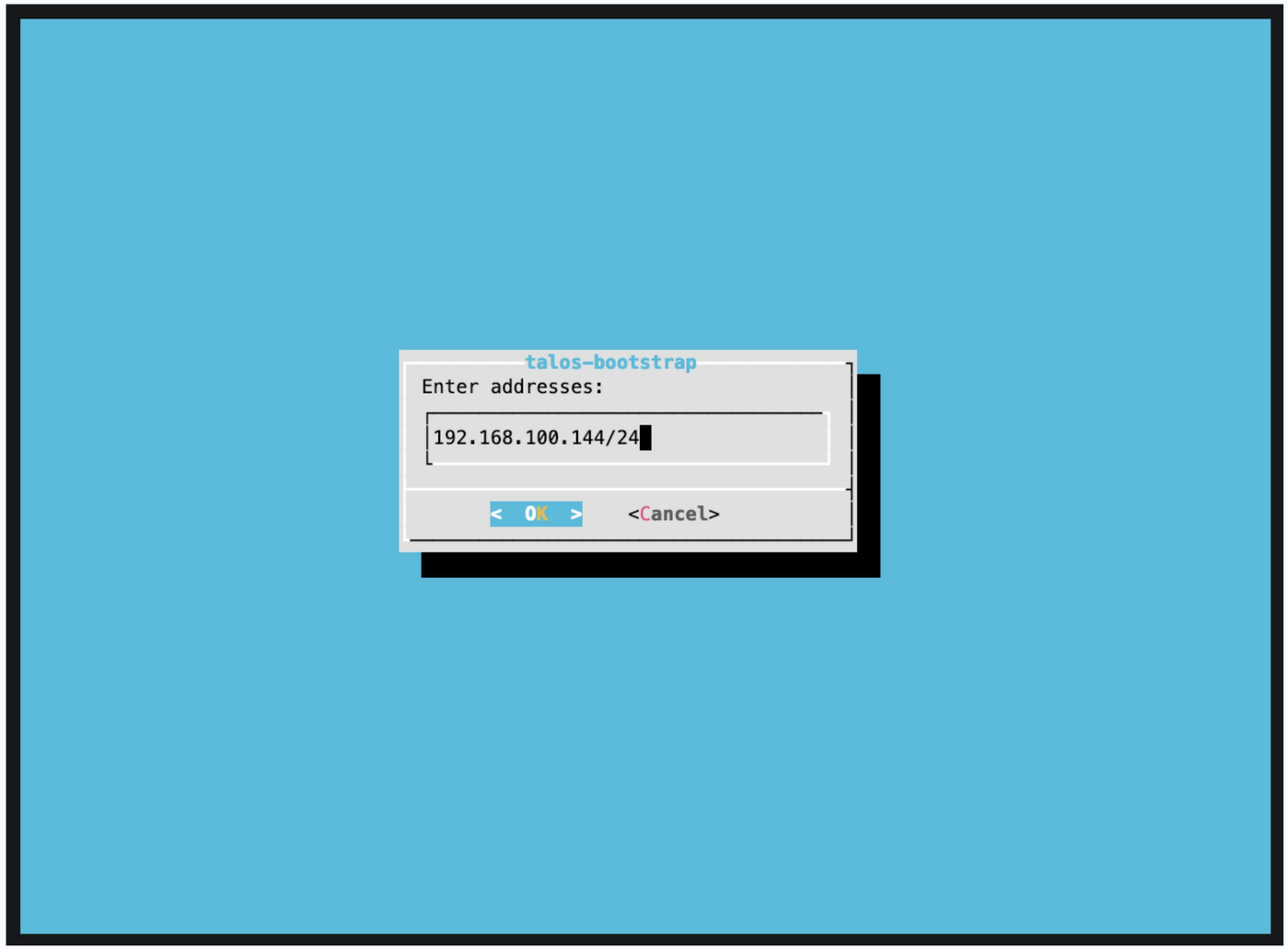
talos-bootstrap



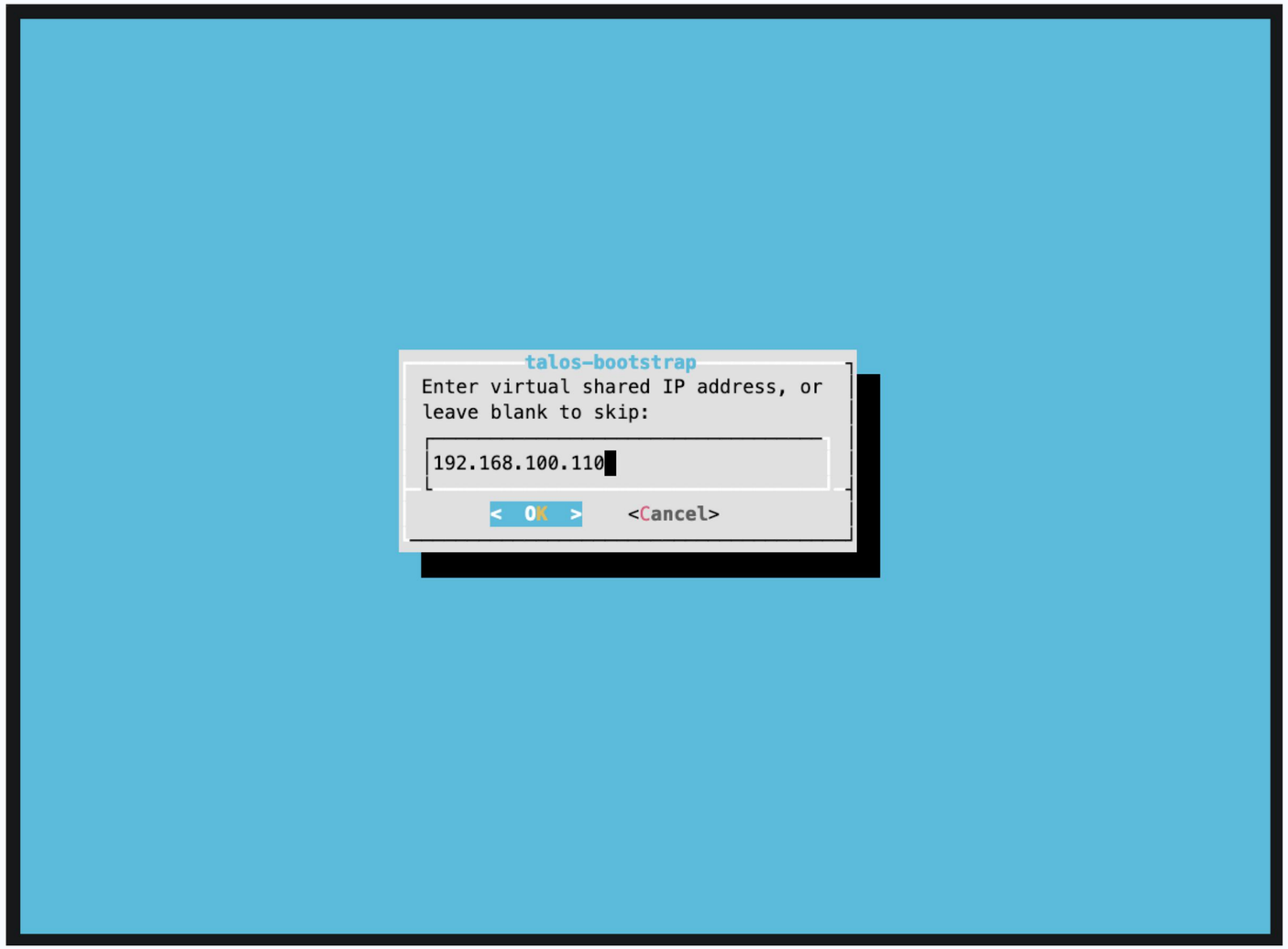
talos-bootstrap



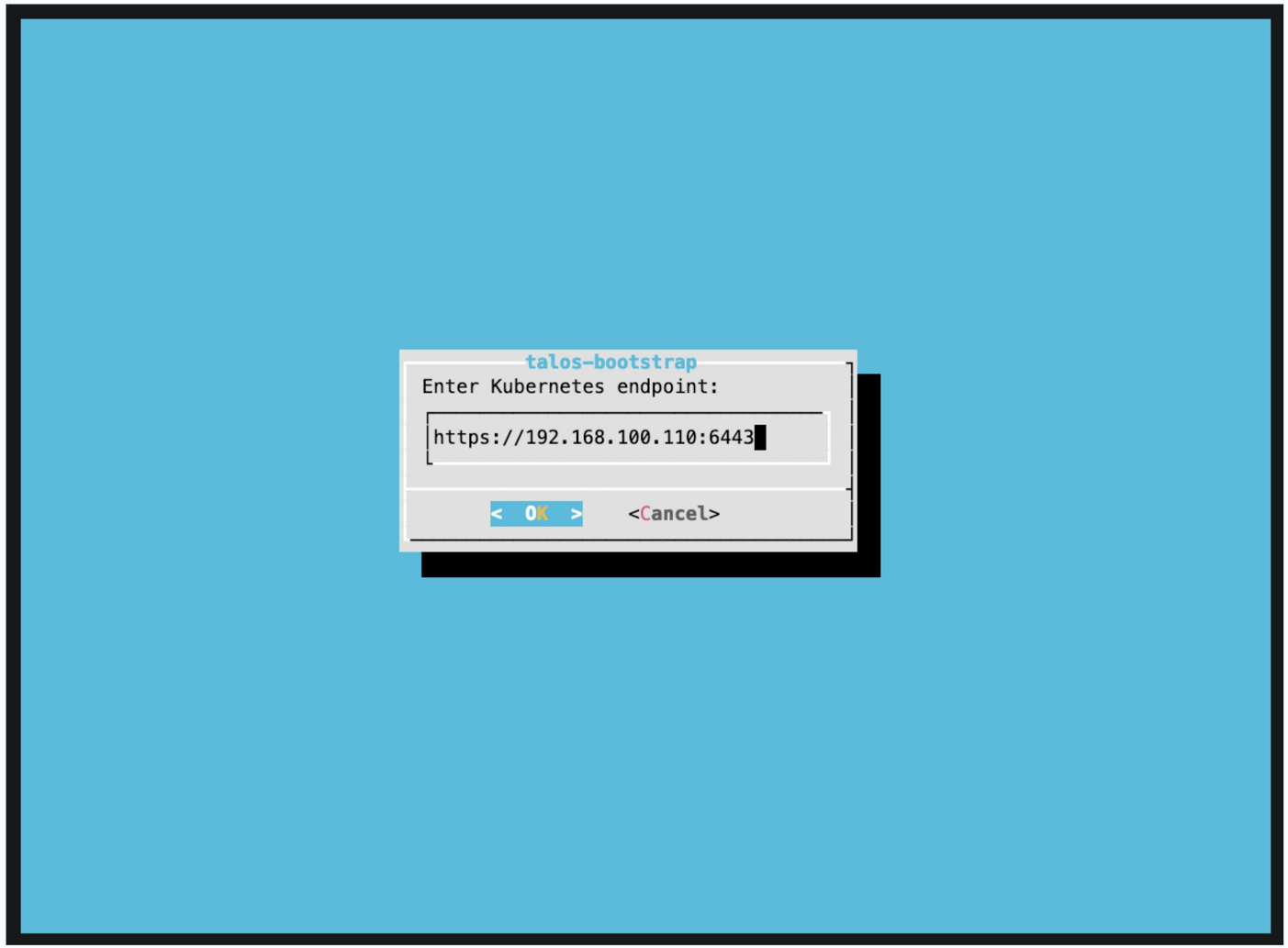
talos-bootstrap



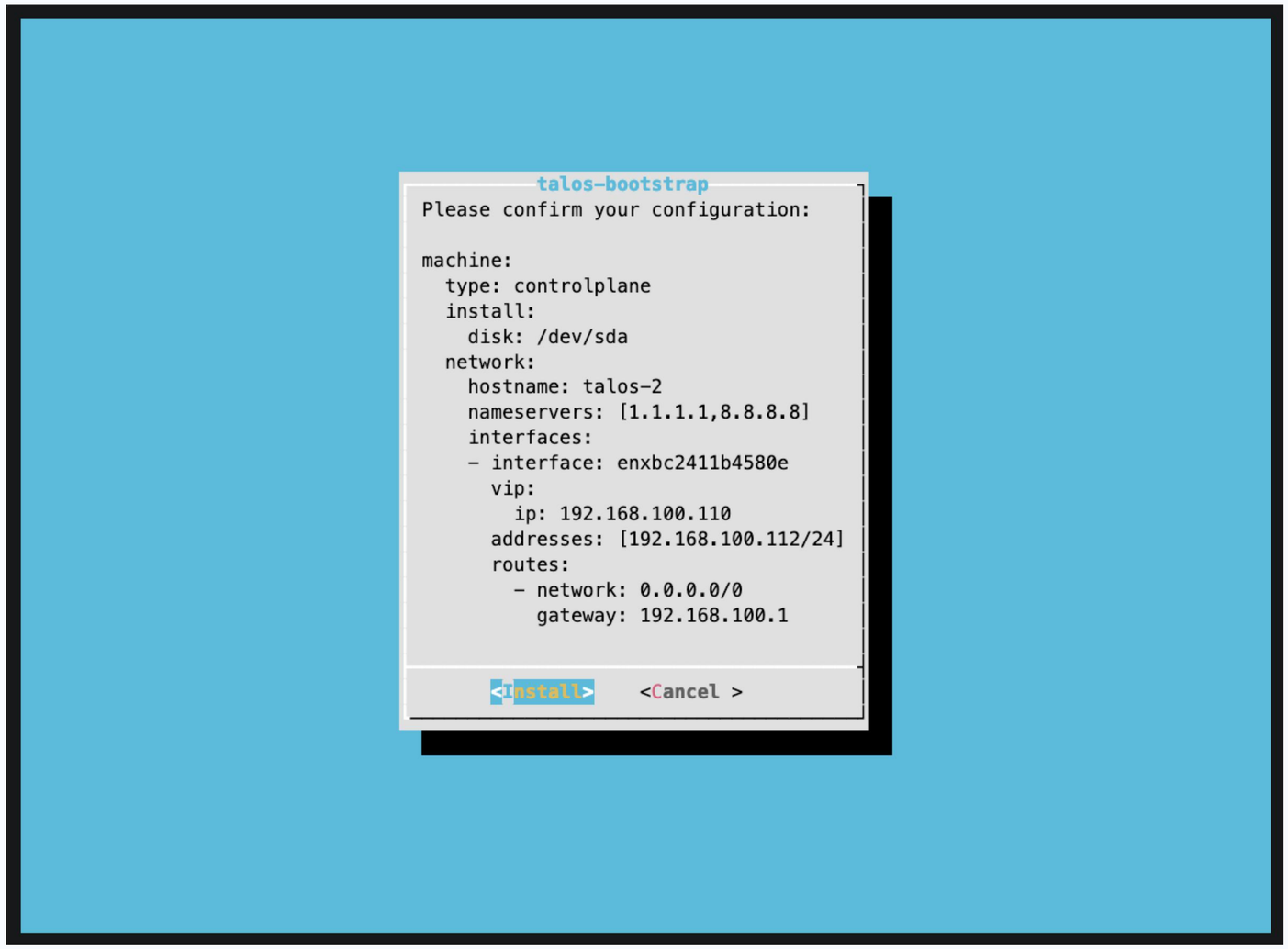
talos-bootstrap



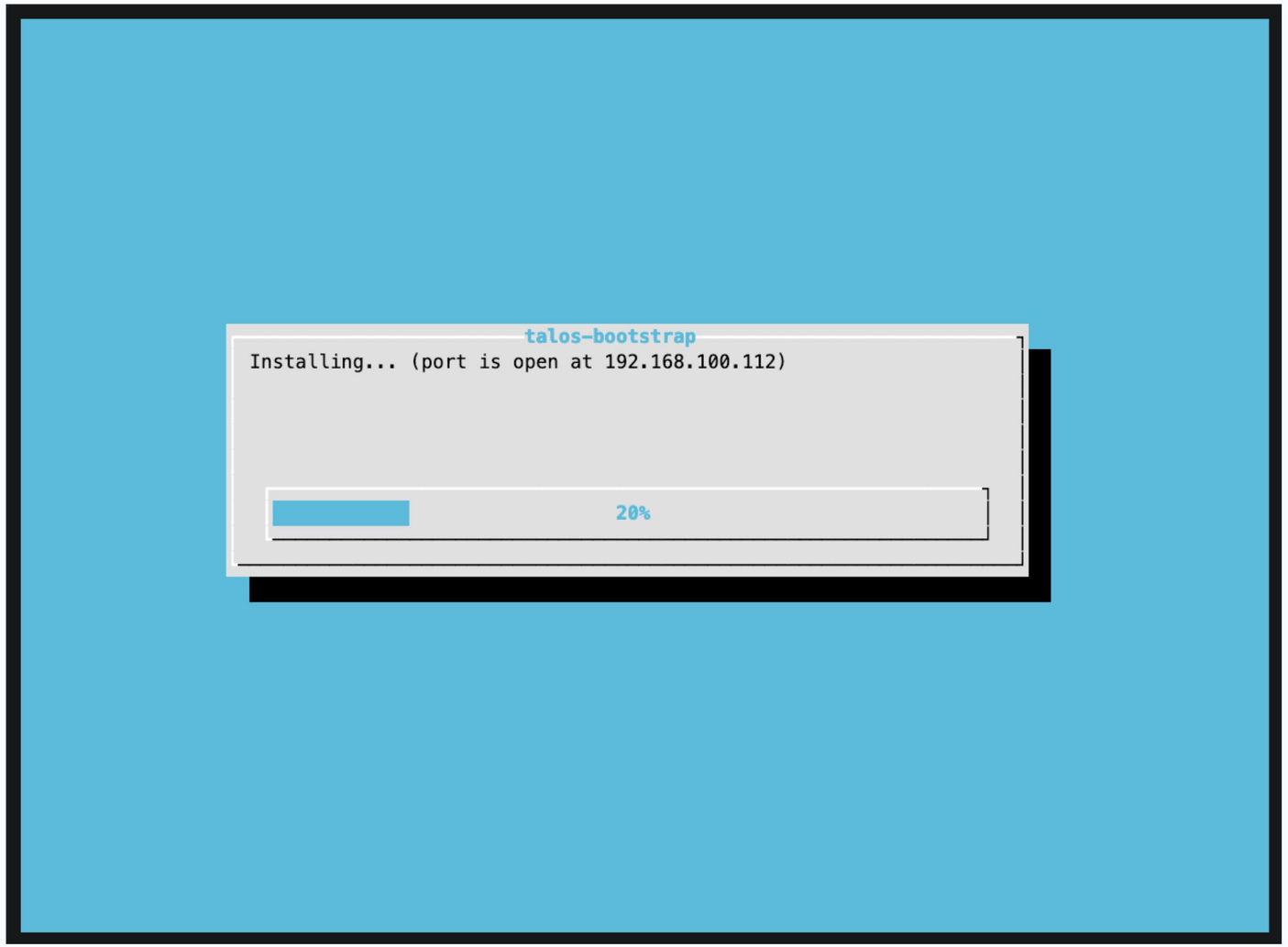
talos-bootstrap



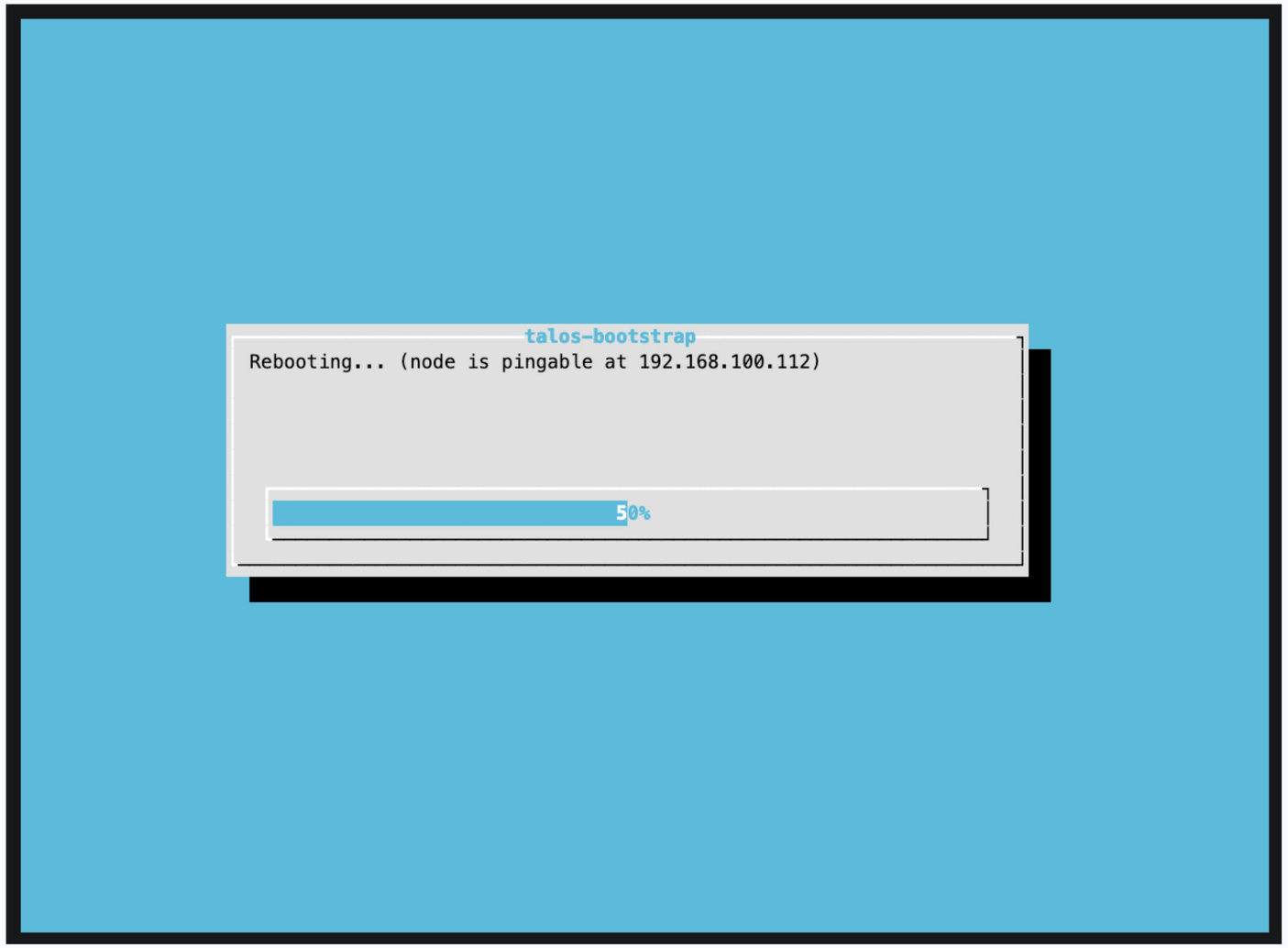
talos-bootstrap



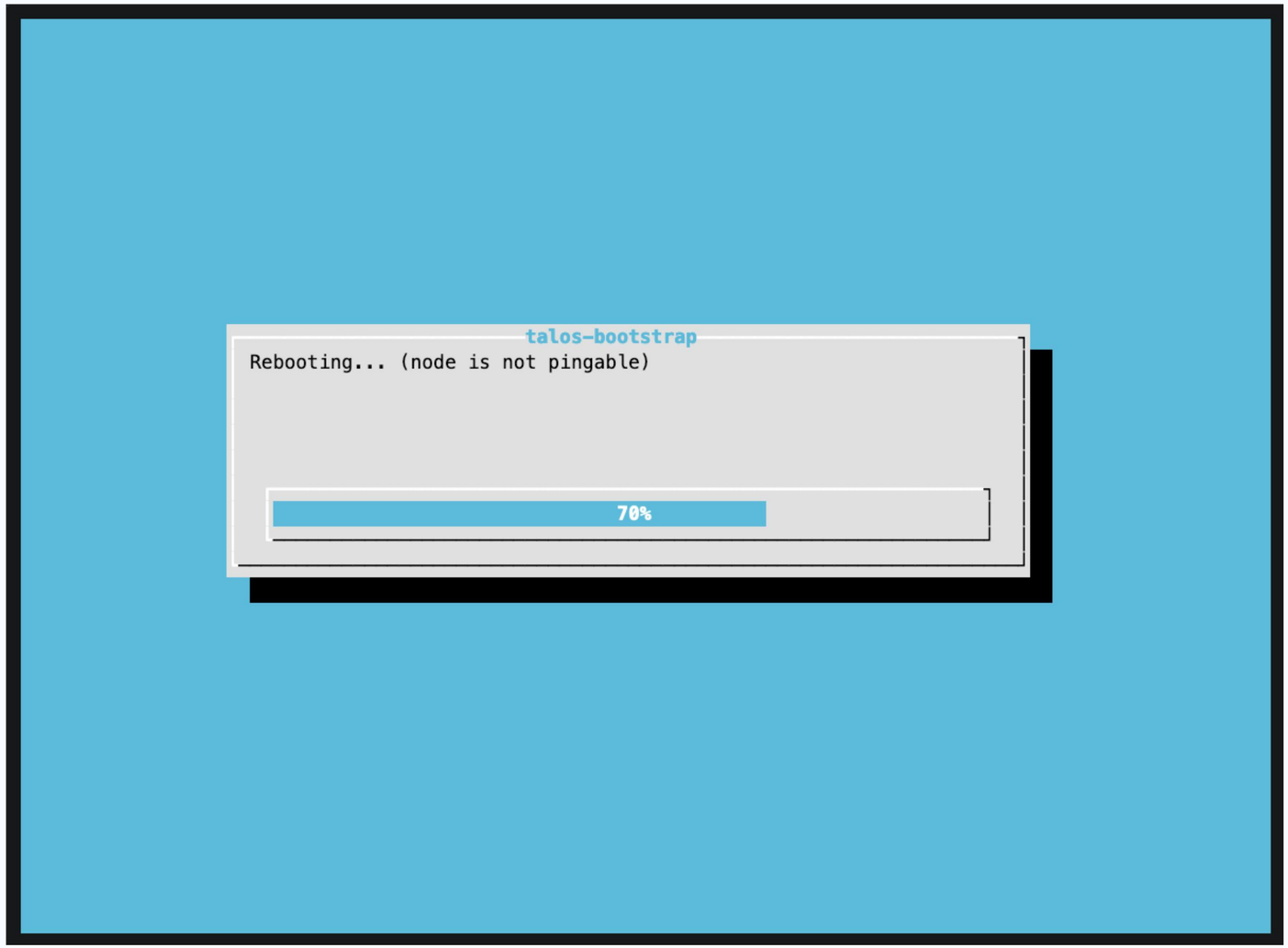
talos-bootstrap



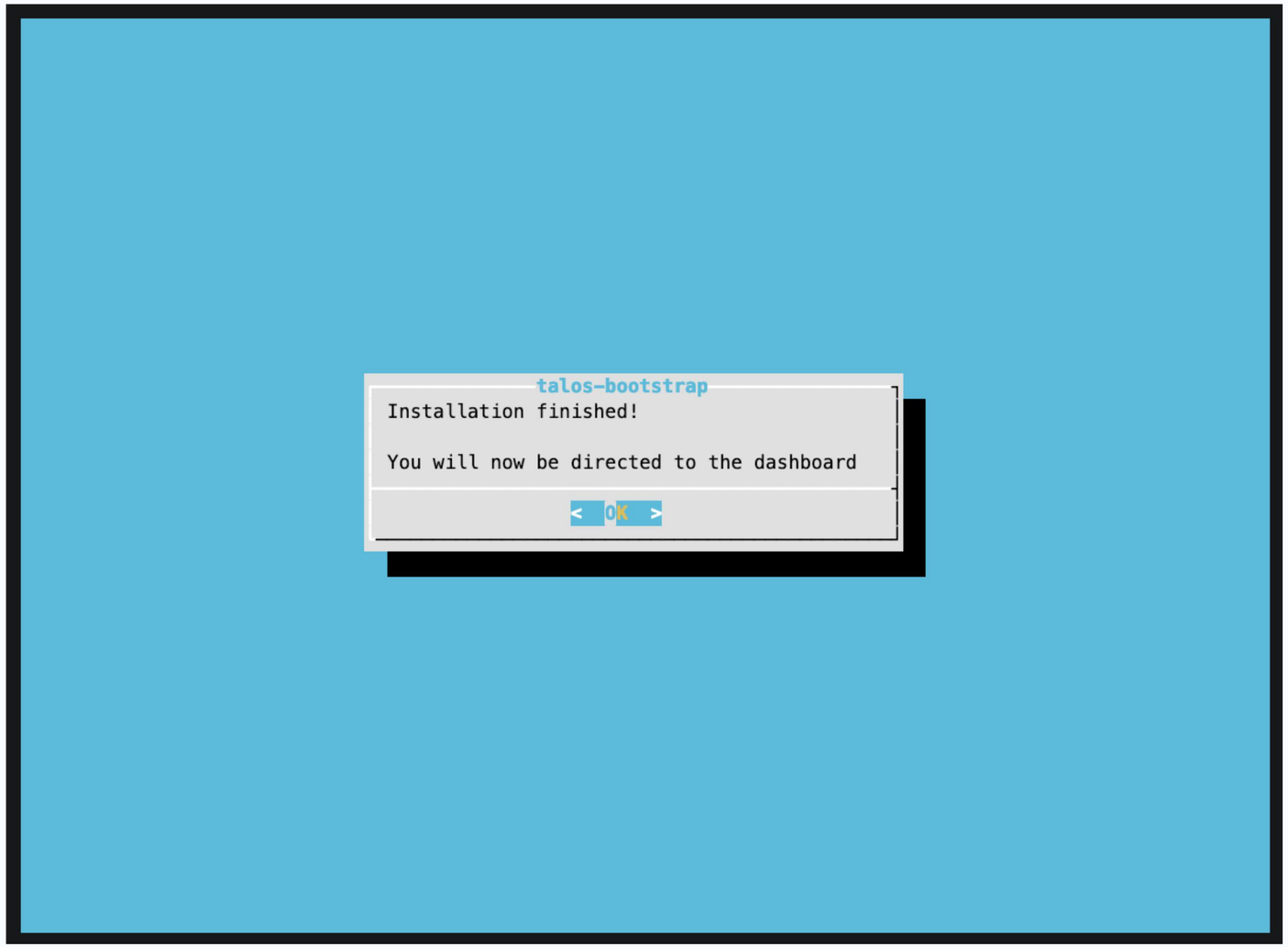
talos-bootstrap



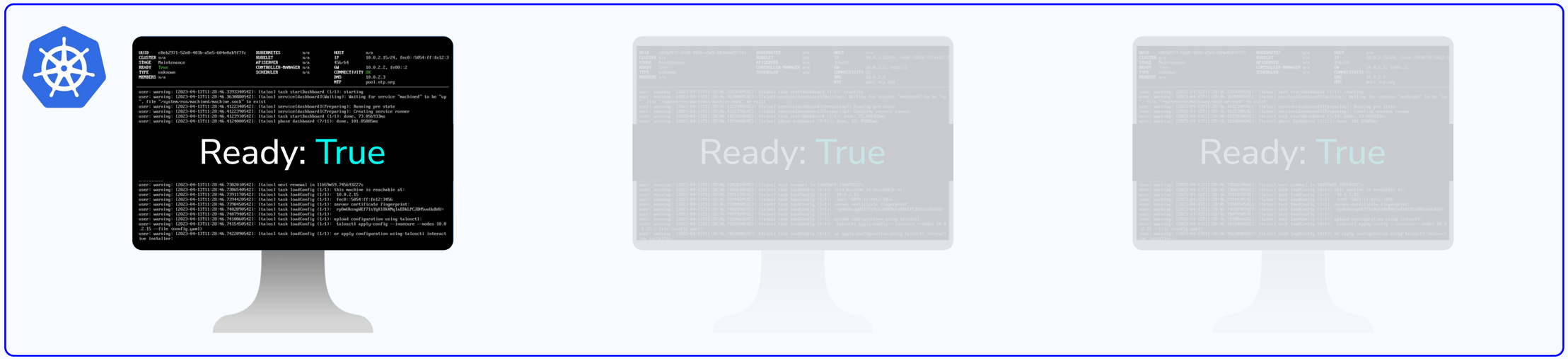
talos-bootstrap

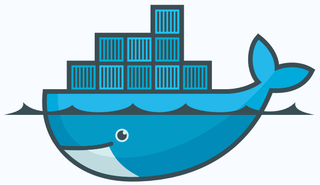


talos-bootstrap



Готовый кластер за 5 минут

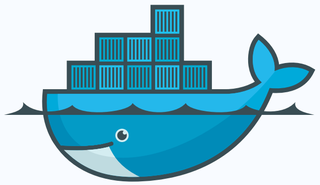




DHCP-server
PXE-server

talos-bootstrap

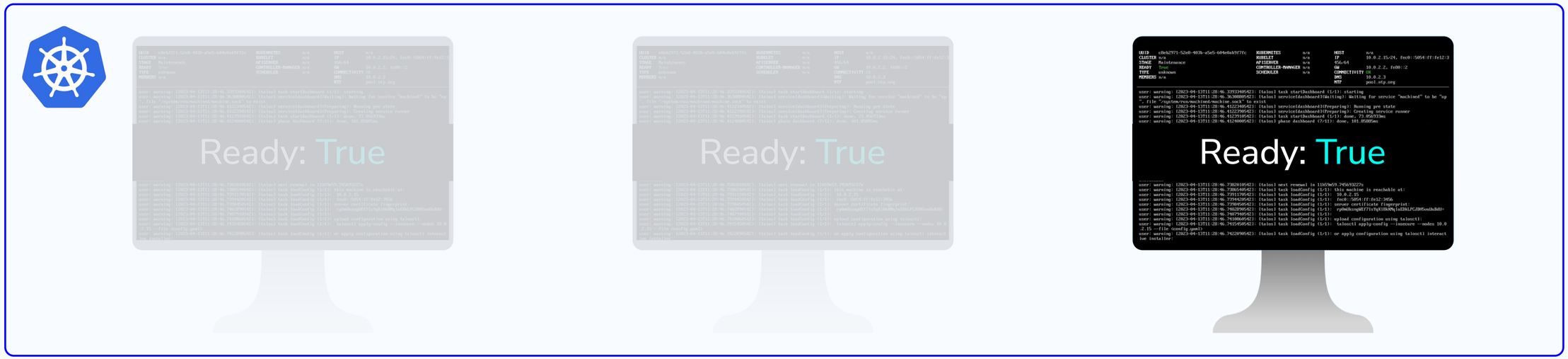
Готовый кластер за 5 минут

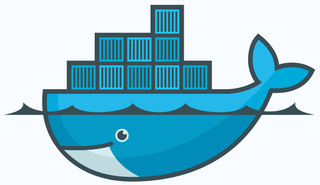


DHCP-server
PXE-server

talos-bootstrap

Готовый кластер за 5 минут

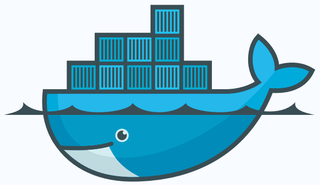




DHCP-server
PXE-server

talos-bootstrap

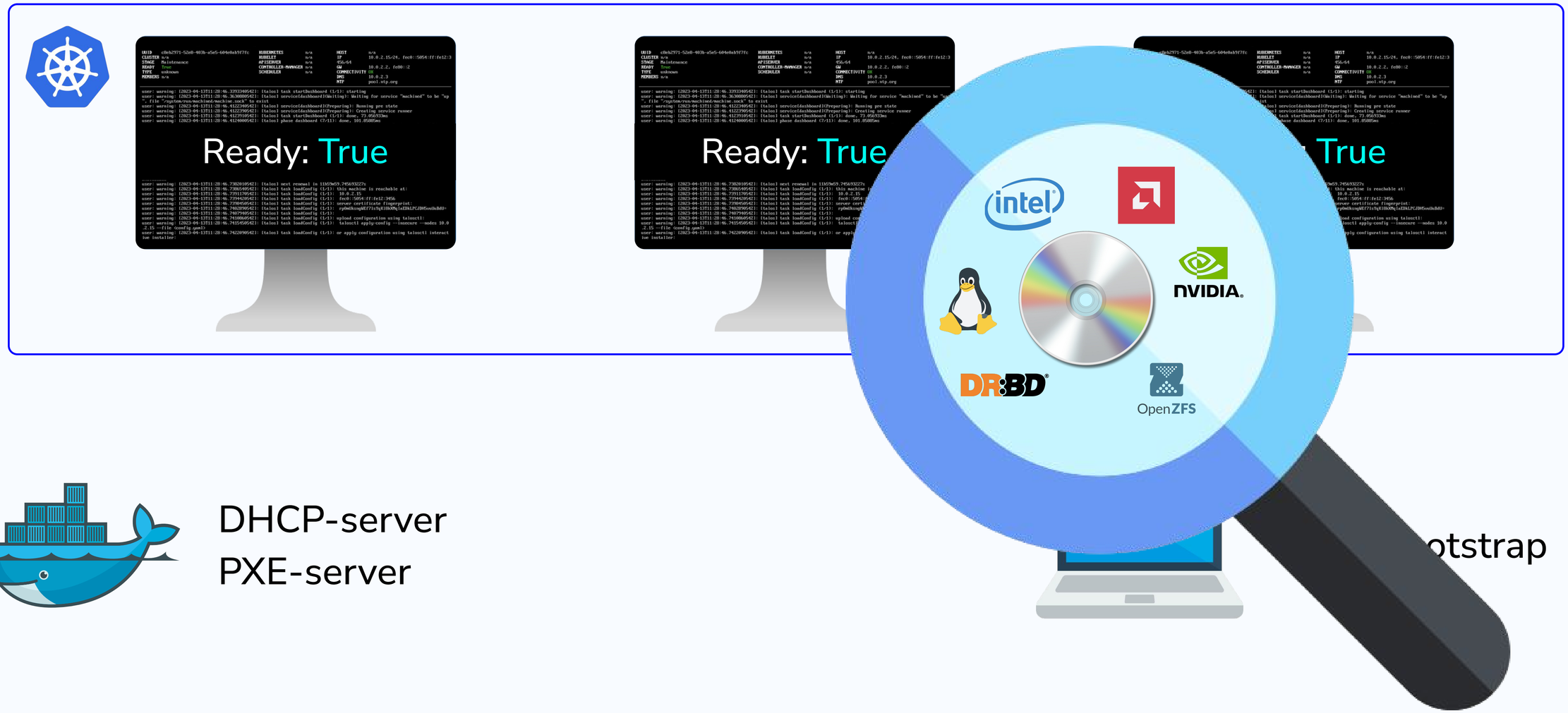
Готовый кластер за 5 минут



DHCP-server
PXE-server

talos-bootstrap

Готовый кластер за 5 минут



DHCP-server
PXE-server

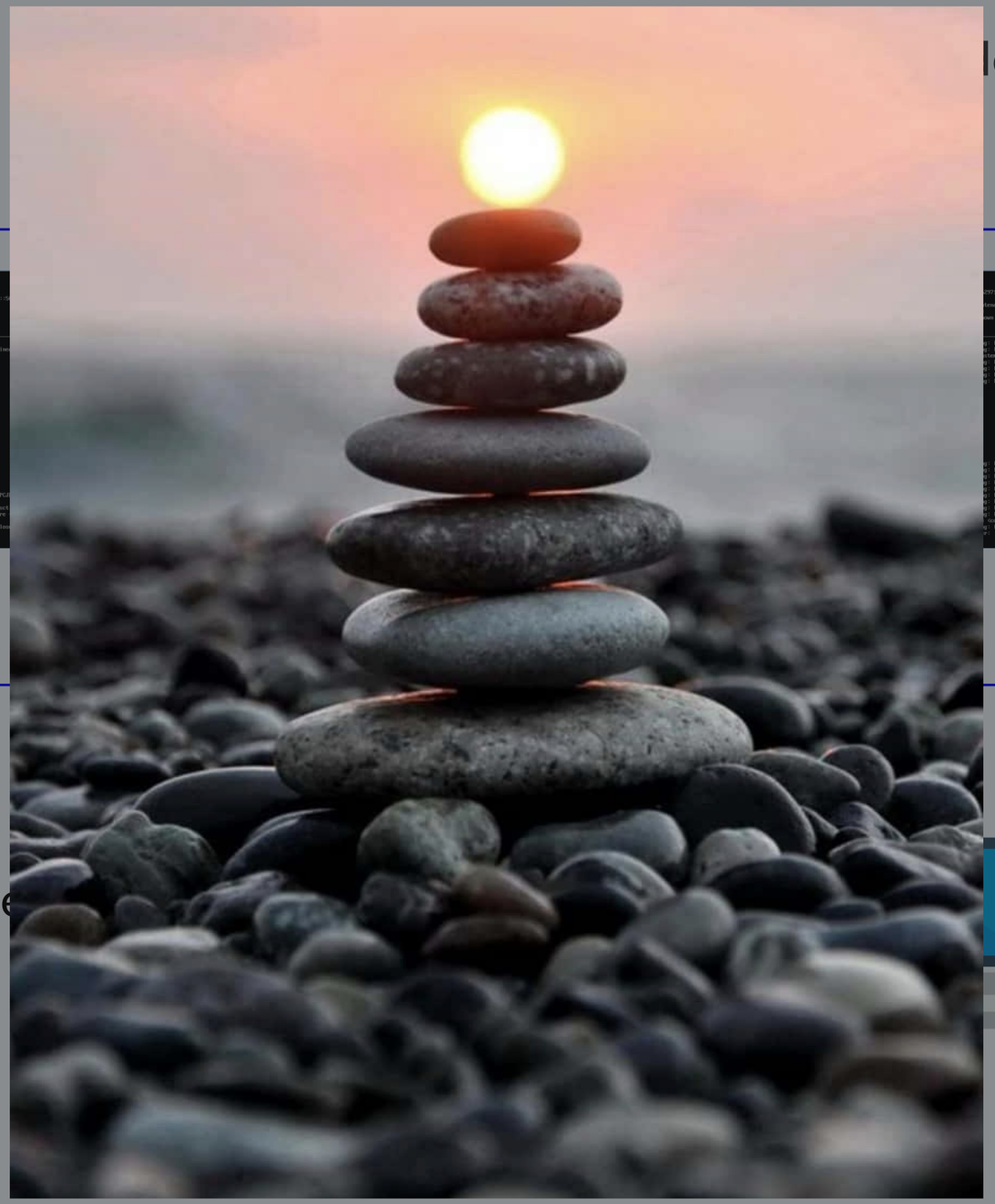
otstrap

Готовый кластер за 5 минут

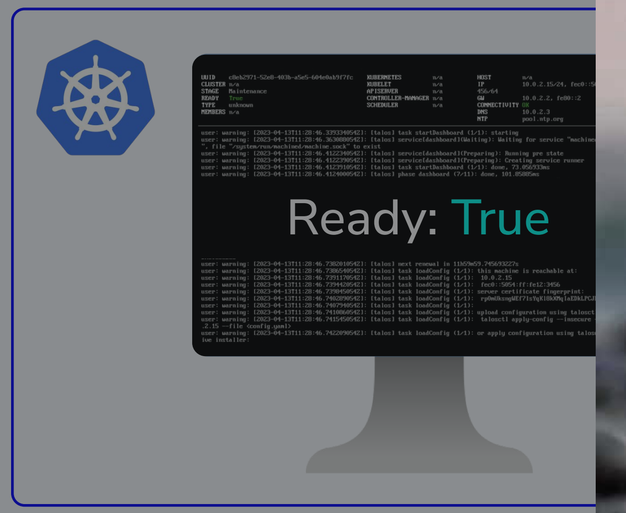
DHCP-server
PXE-server

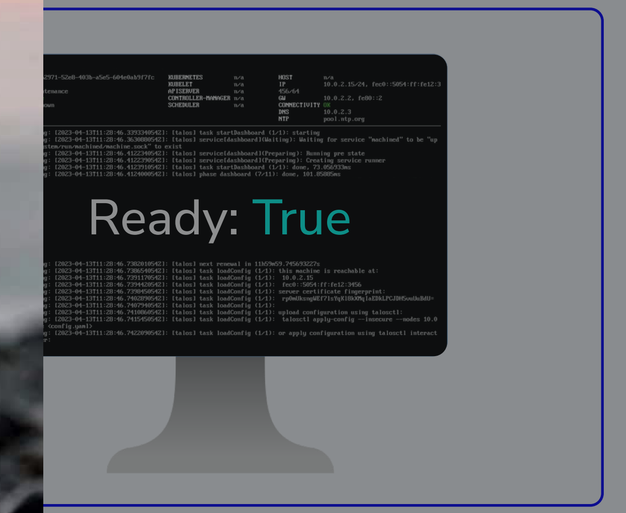
talos-bootstrap

Готовый кластер за



developed by Ænix

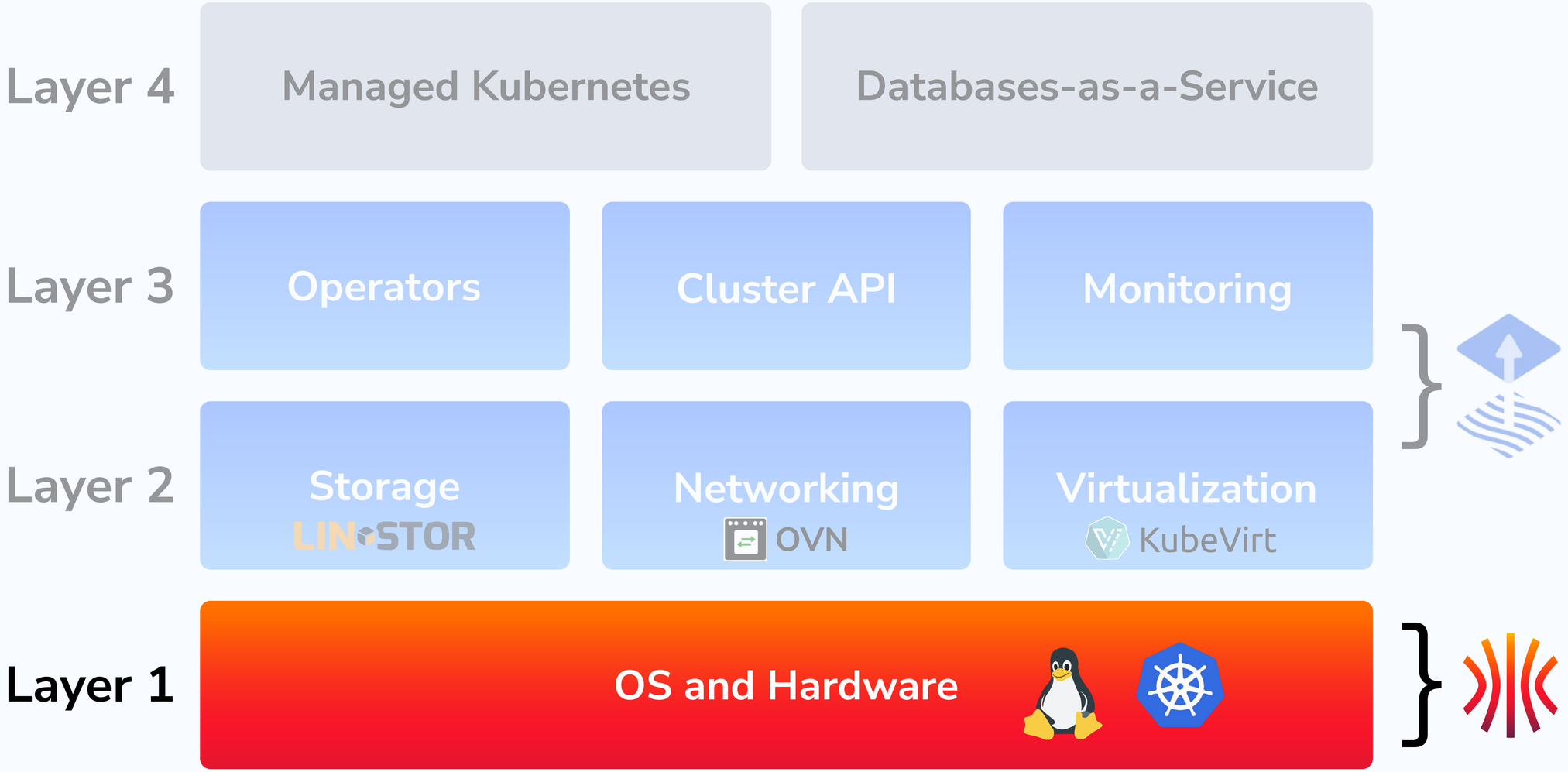


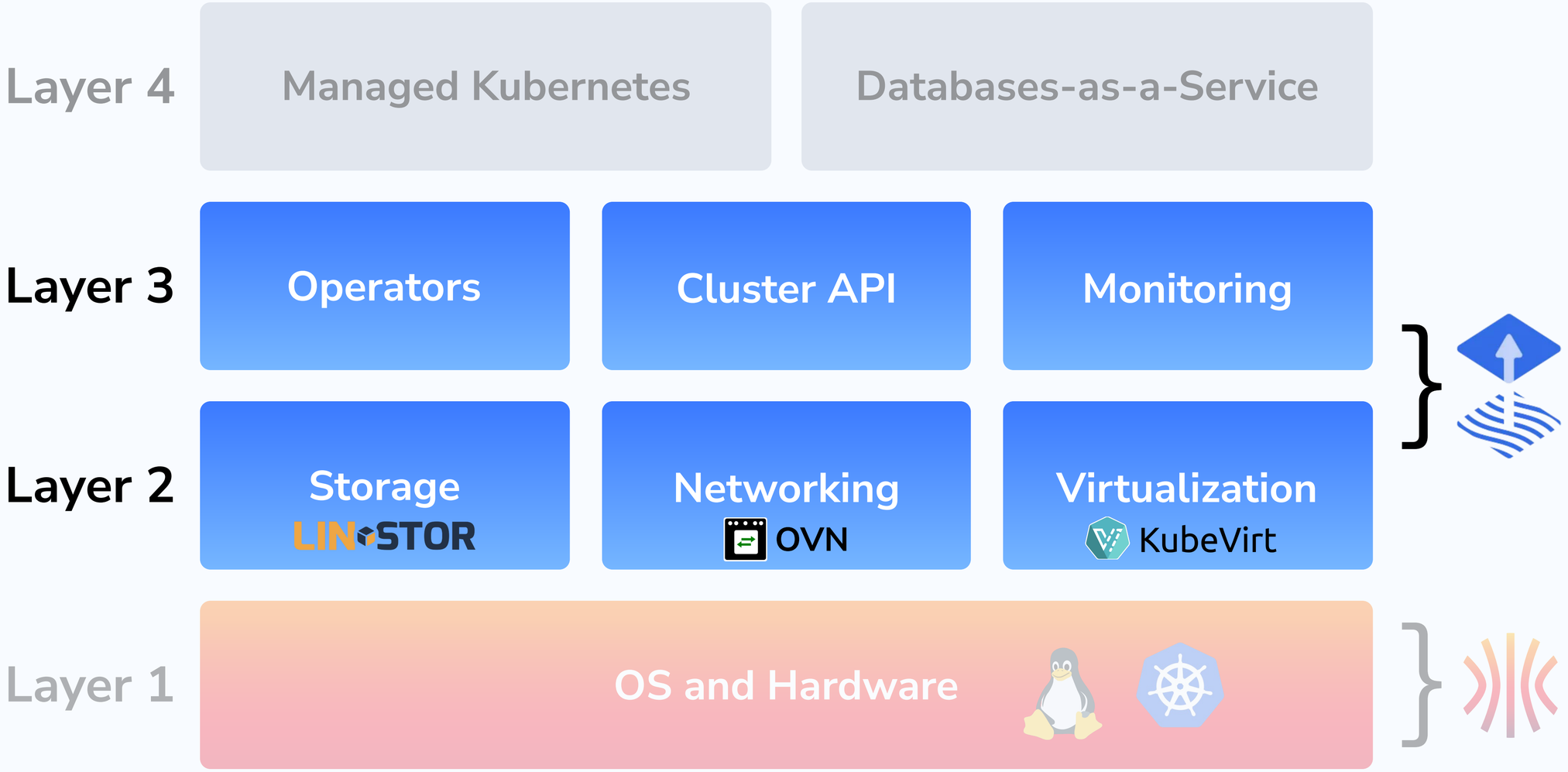


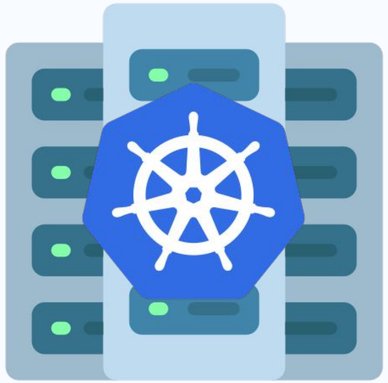
DHCP-serve
PXE-server

talos-bootstrap

COZYSTACK

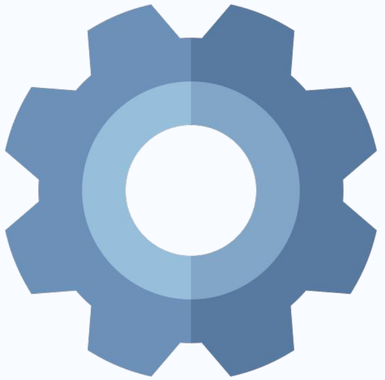

Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

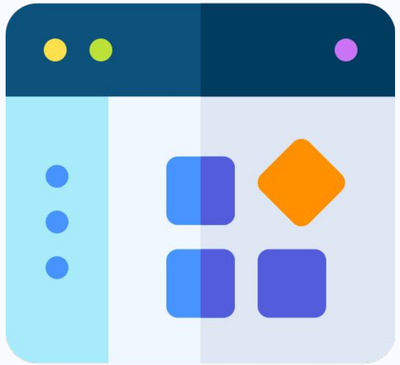


K8s на железе

Железо и ОС

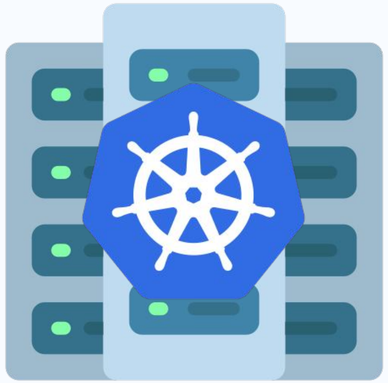


Системные компоненты



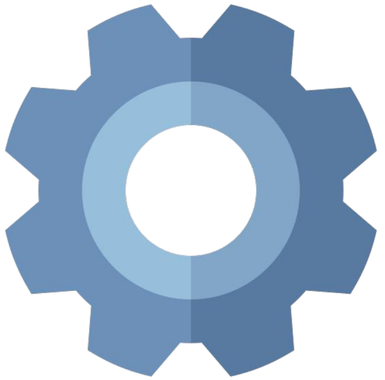
Пользовательские приложения

ПЛАТФОРМА

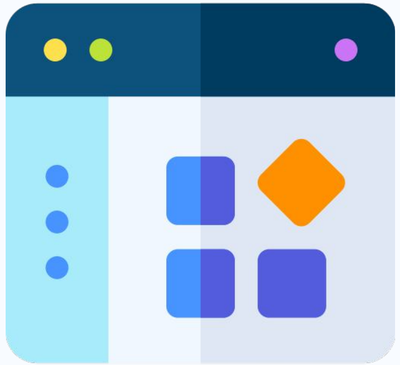


K8s на железе

Железо и ОС



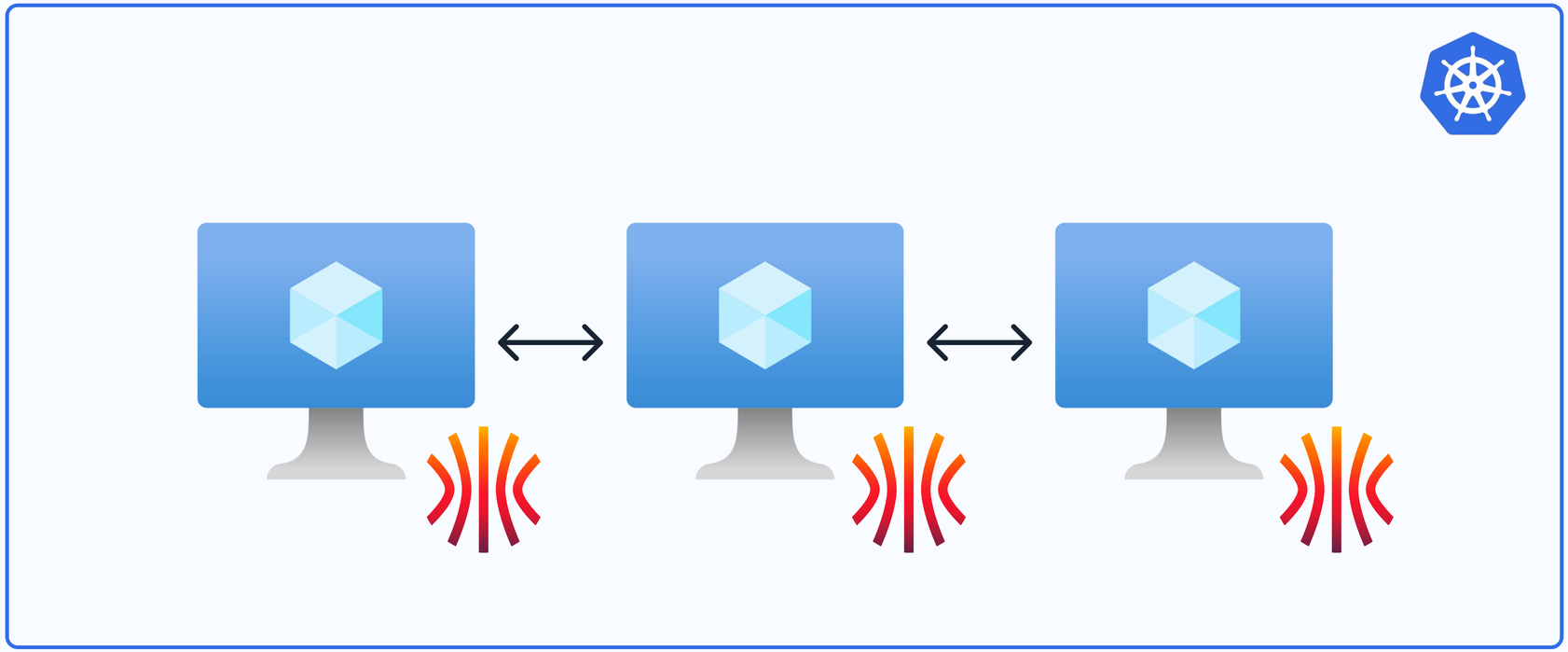
Системные компоненты



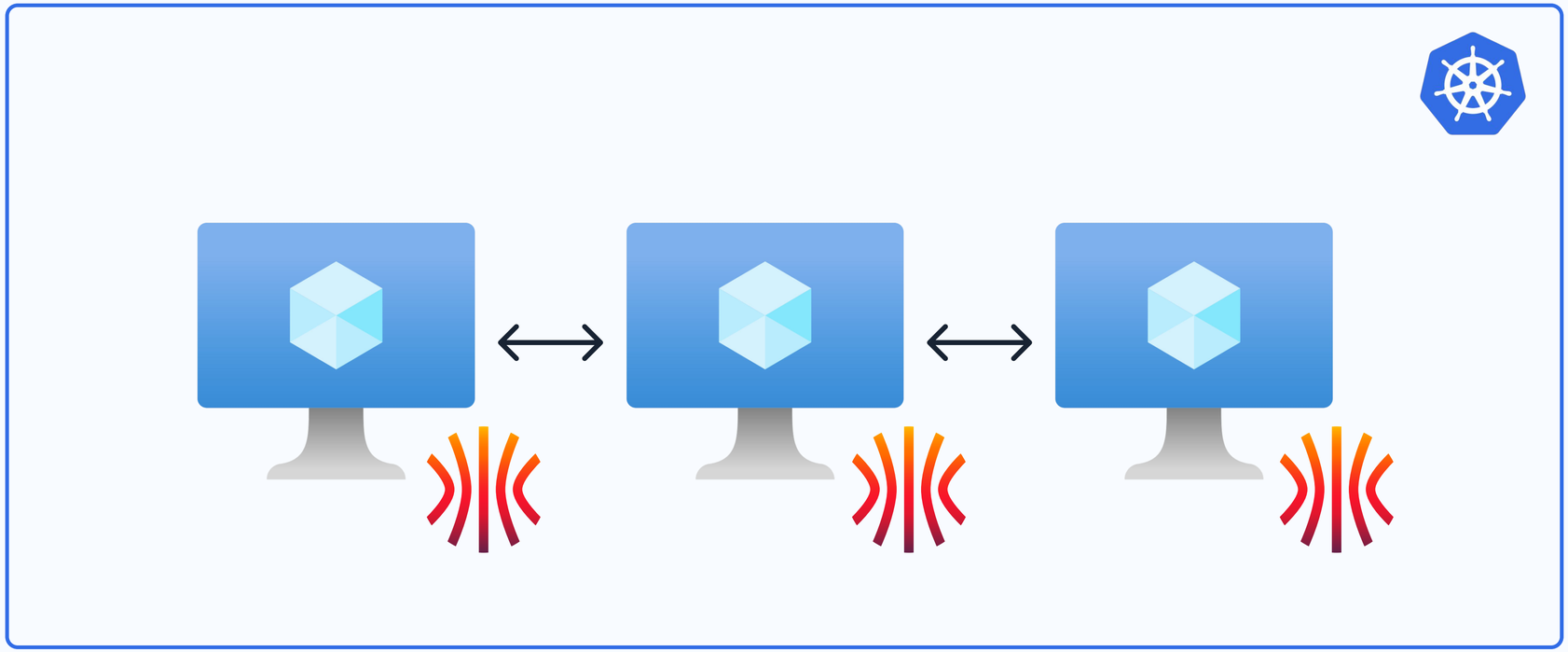
Пользовательские приложения

ПЛАТФОРМА

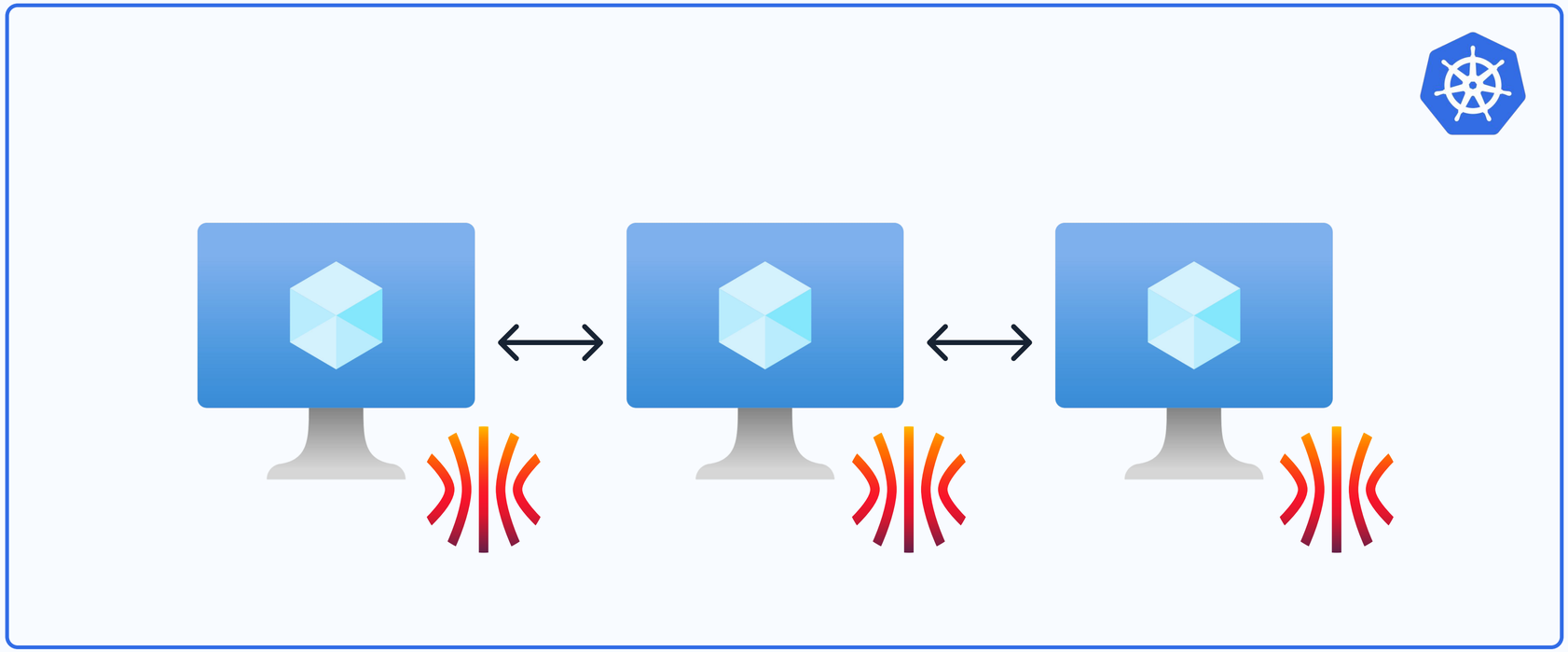
Установка инфраструктурных приложений



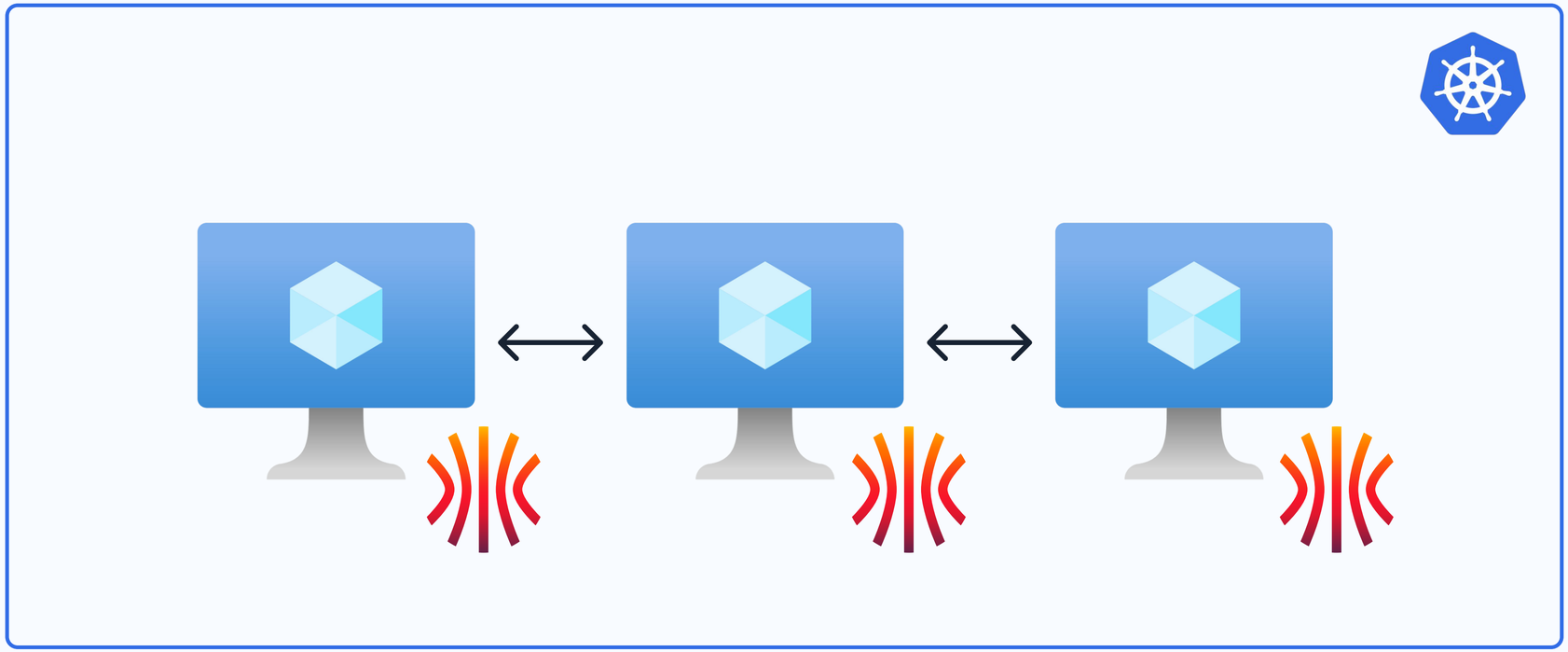
Установка инфраструктурных приложений



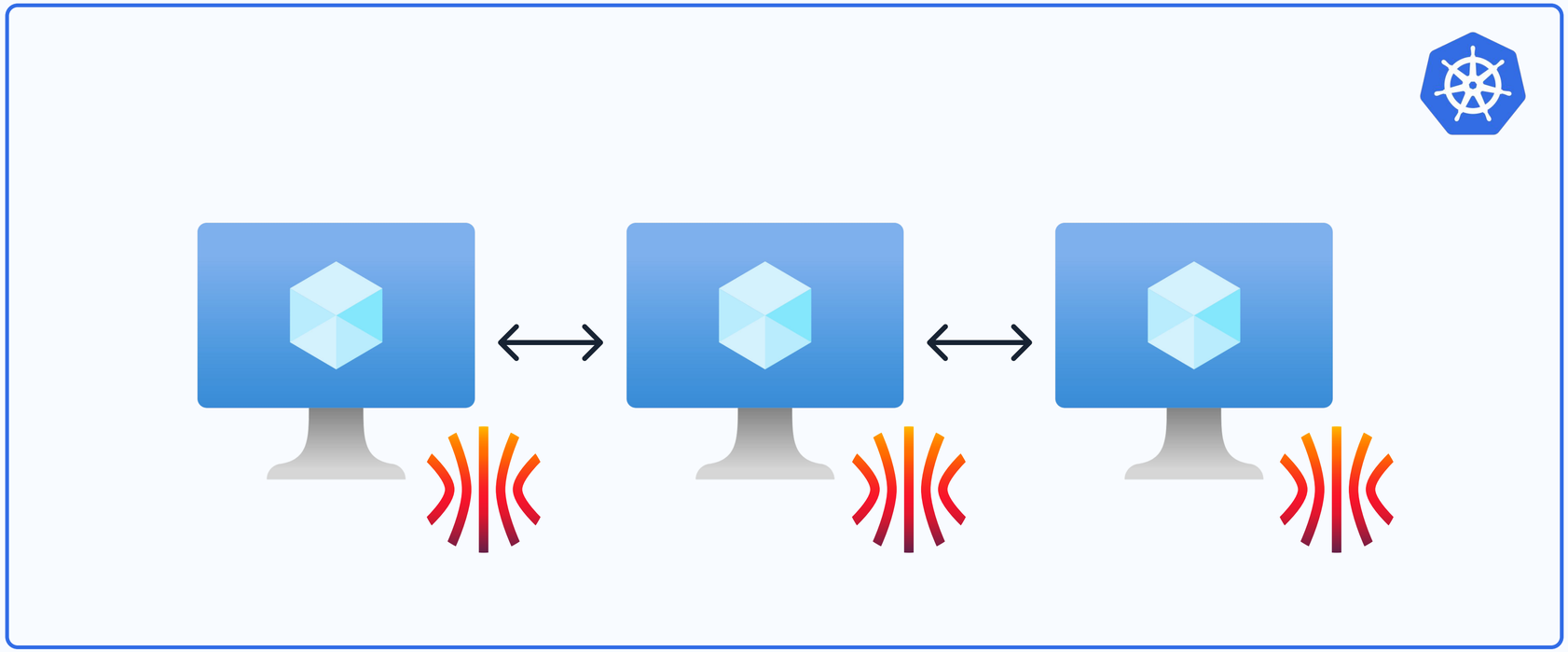
Установка инфраструктурных приложений



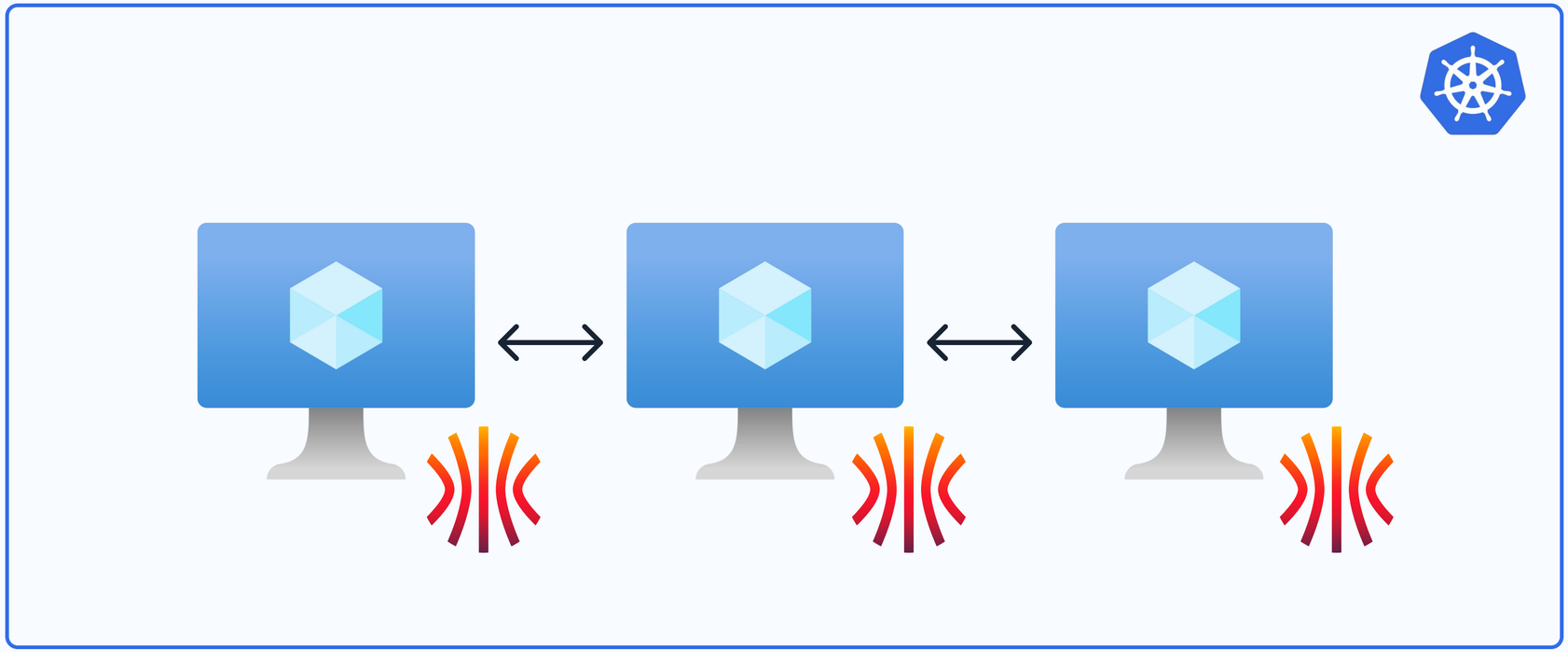
Установка инфраструктурных приложений



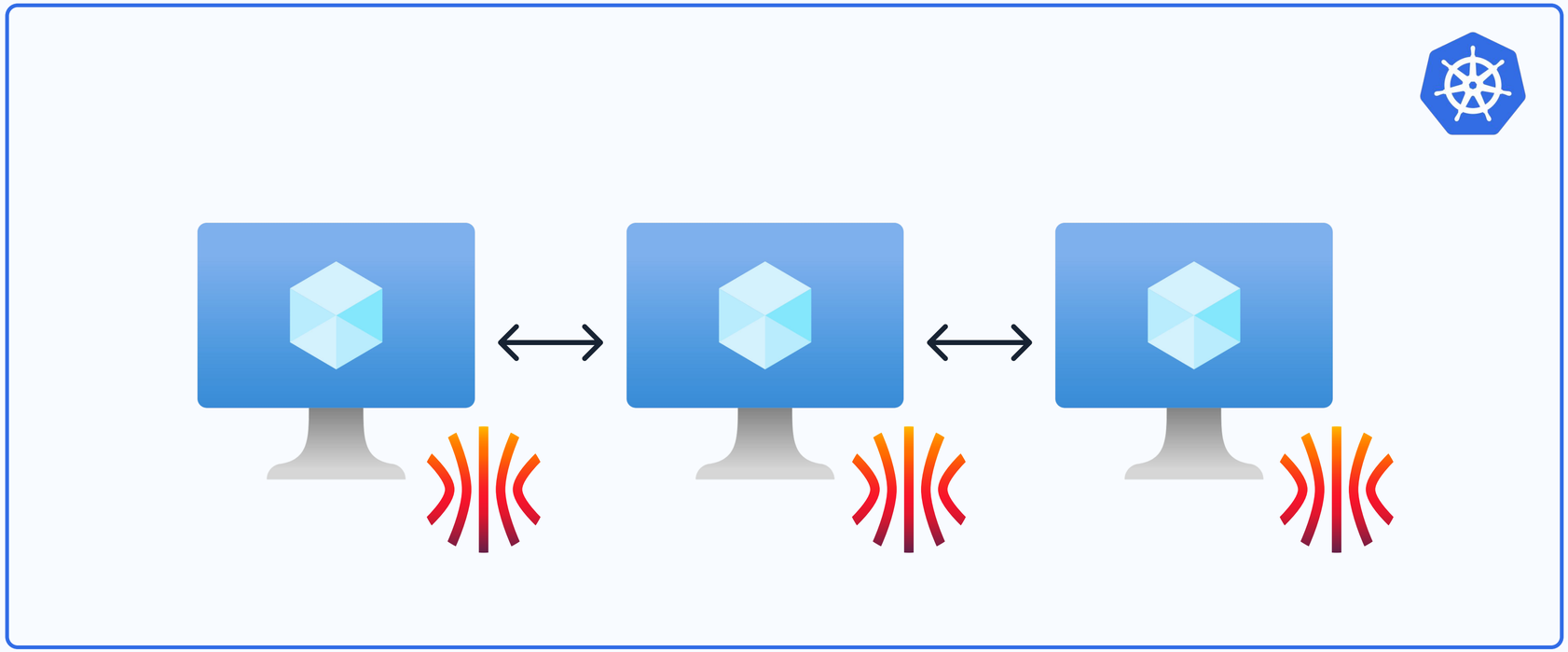
Установка инфраструктурных приложений



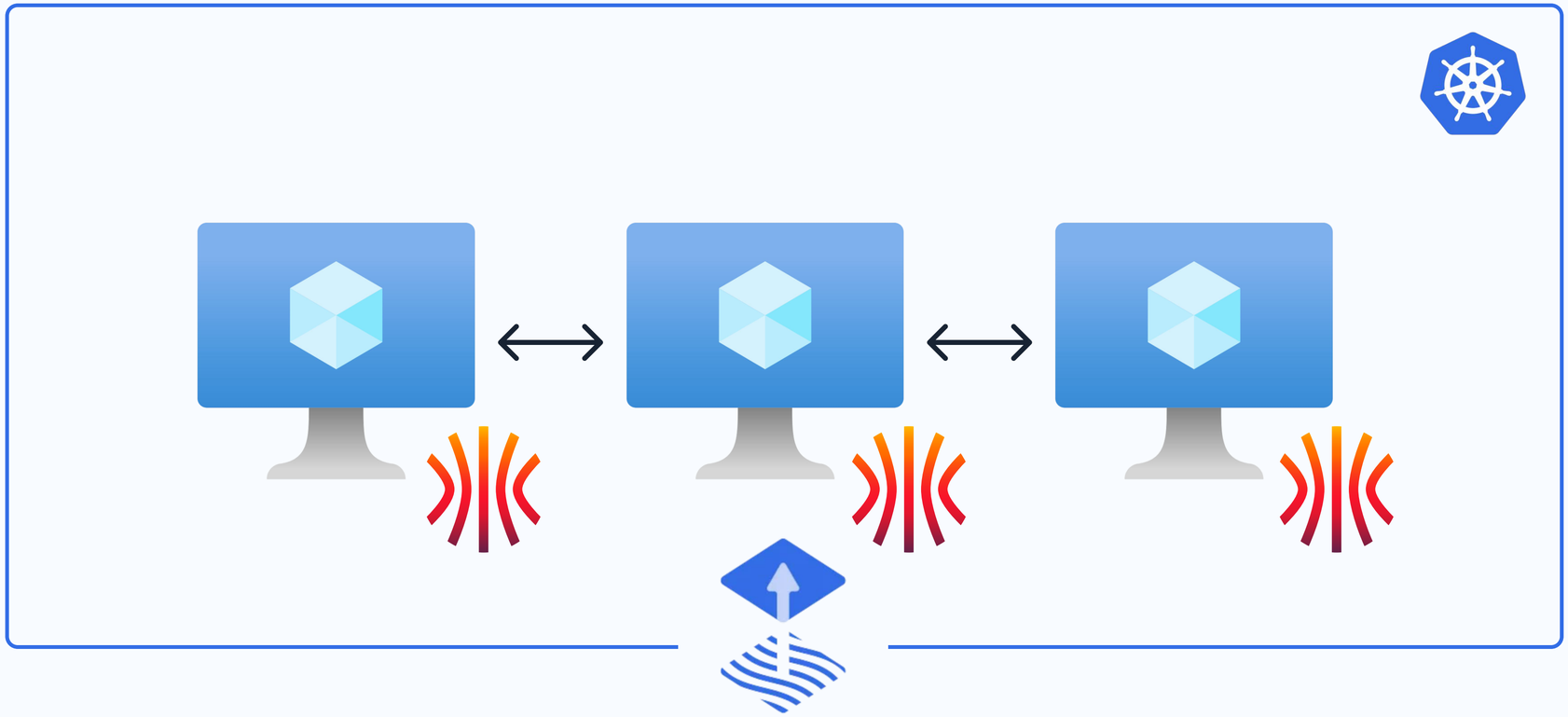
Установка инфраструктурных приложений



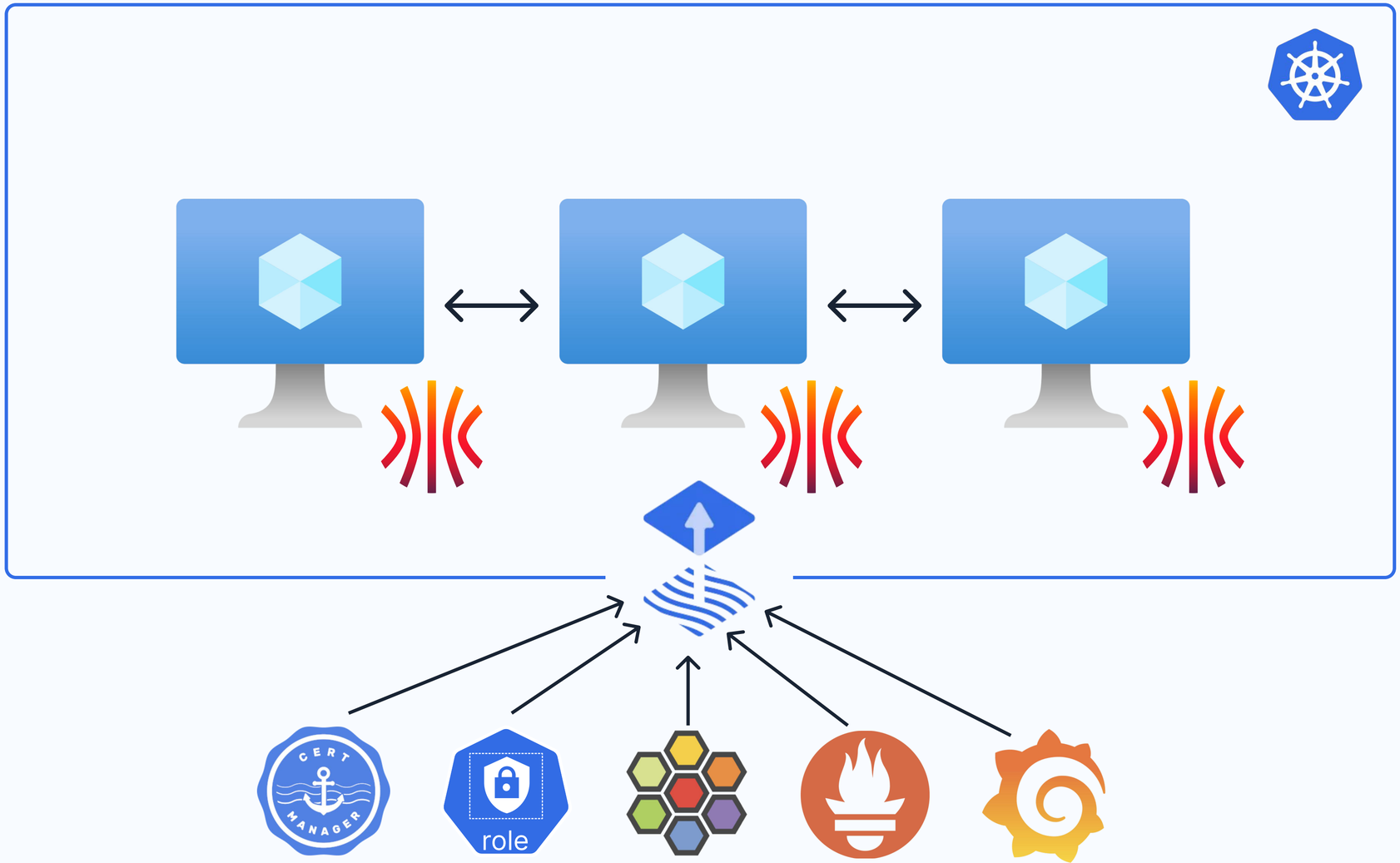
Talos Linux ❤️❤️❤️ FluxCD



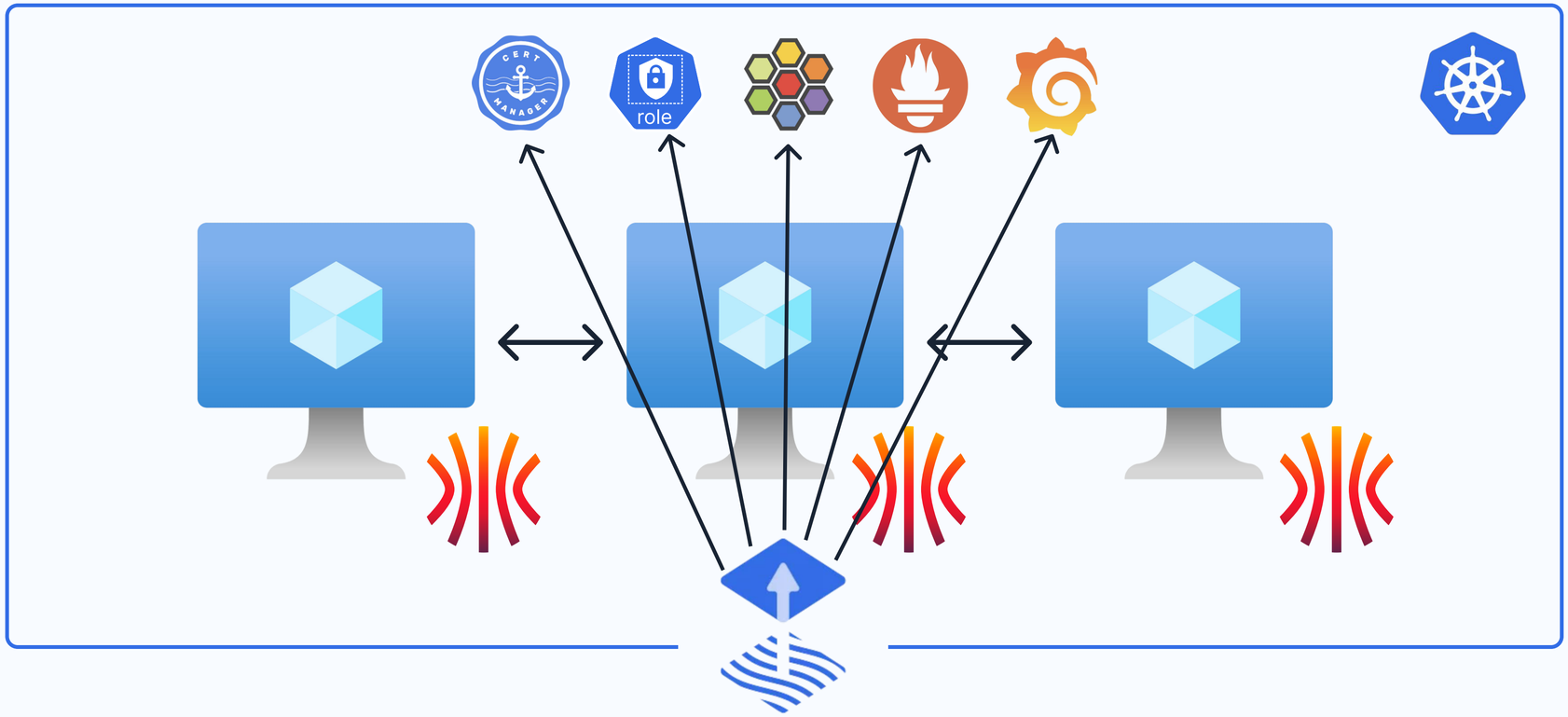
Talos Linux ❤️❤️❤️ FluxCD



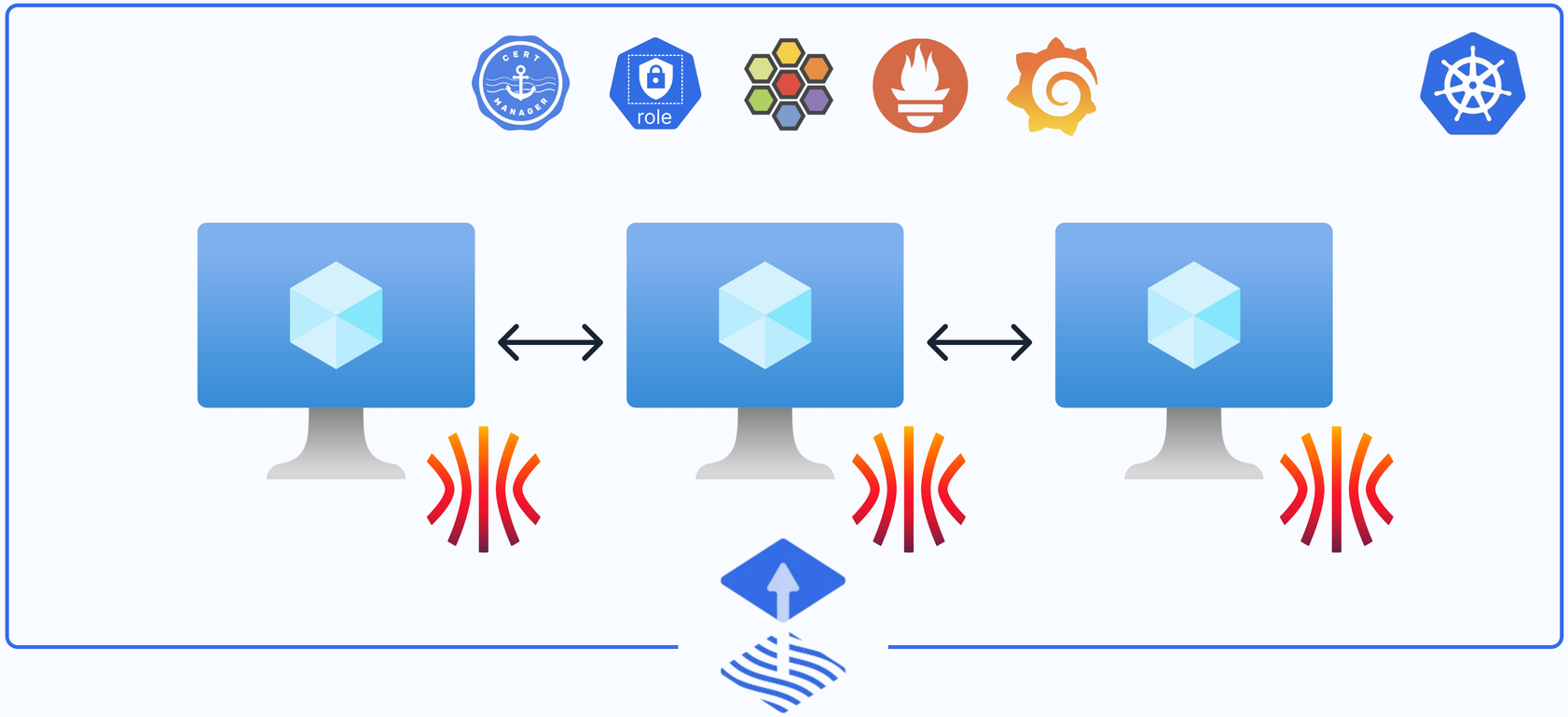
Talos Linux ❤️❤️❤️ FluxCD



Talos Linux ❤️❤️❤️ FluxCD



Talos Linux ❤️❤️❤️ FluxCD



```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: { }
```

```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: { }
```

```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests: { }
```

```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests:
  - https://example.org/fluxcd.yaml
```

```yaml
machine:
  type: control-plane
  ca: { crt: ""  key: "" }
  certSANs:
    - 127.0.0.1
    - cluster1.example.org
  network:
    interfaces:
      - interface: eth0
        vip: 192.168.100.10
  install:
    disk: /dev/sda
cluster:
  controlPlane:
    endpoint: https://192.168.100.10:6443
  ca: { crt: ""  key: "" }
  apiServer: {}
  etcd: {}
  extraManifests:
  - https://example.org/fluxcd.yaml
  - https://example.org/my-config.yaml
```

Метод наш другой немного,
не используем git мы и вместе все
компоненты устанавливаем.

Установка платформы

COZYSTACK

Установка платформы

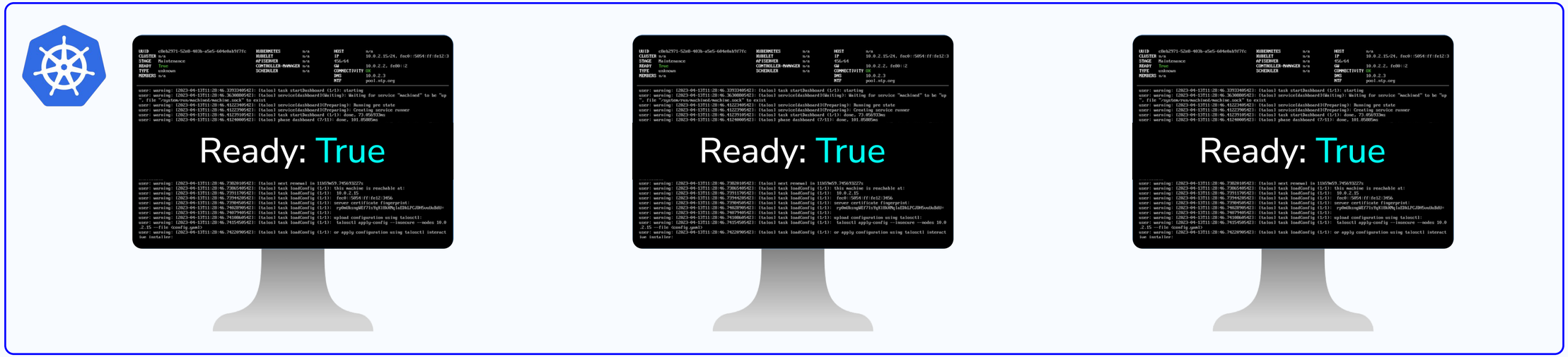
COZYSTACK

Установка платформы

COZYSTACK

Установка платформы

COZYSTACK



Установка платформы

COZYSTACK

kubectl apply -f cozystack.yaml

Установка платформы

COZYSTACK

kubectl apply -f cozystack.yaml

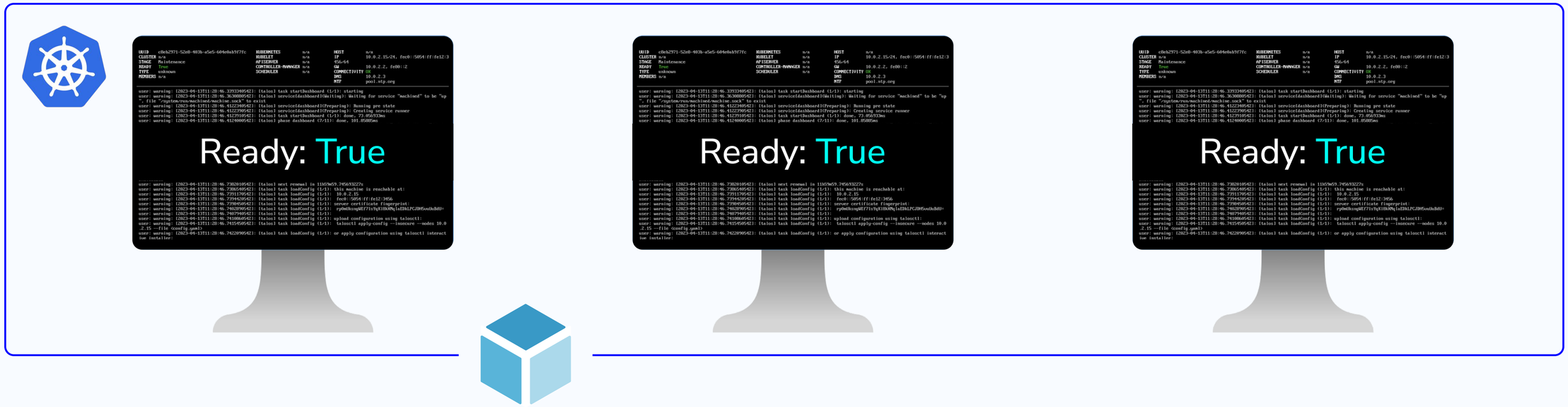
Установка платформы

COZYSTACK

kubectl apply -f cozystack.yaml

Установка платформы

COZYSTACK



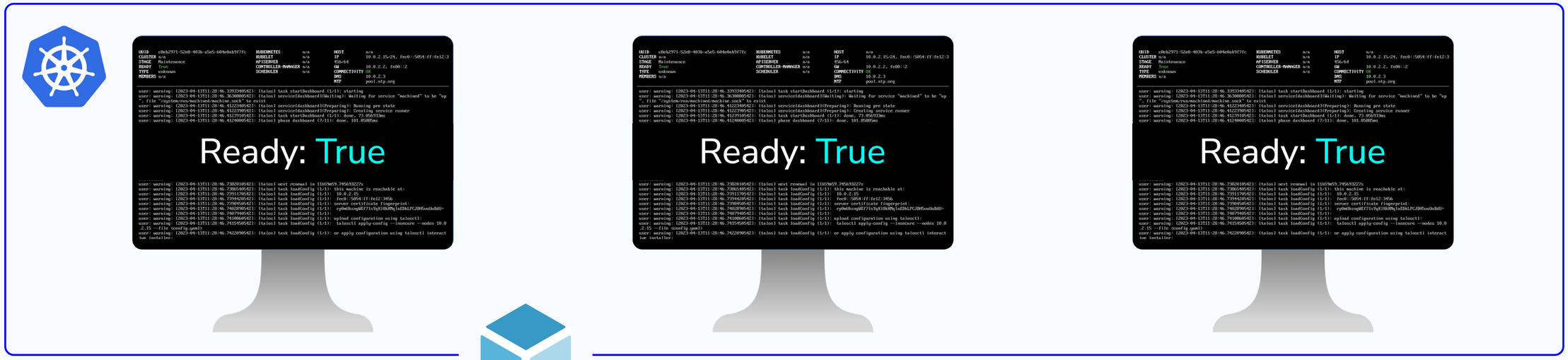
kubectl apply -f cozystack.yaml

Установка платформы

COZYSTACK

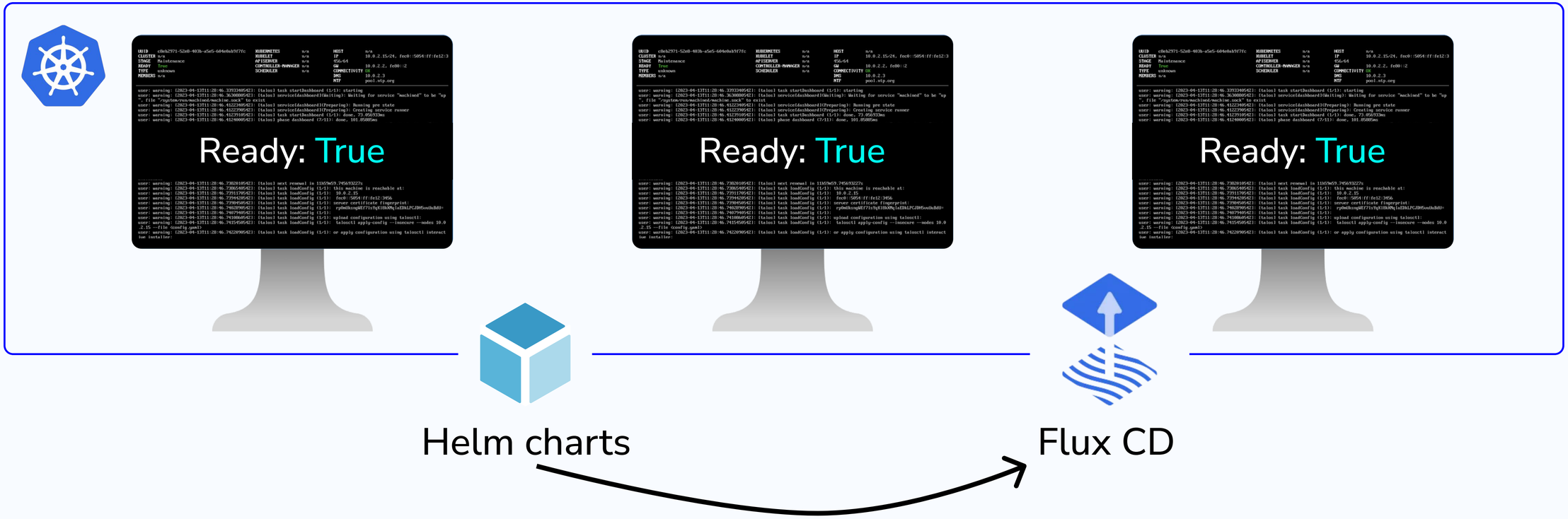
Установка платформы

COZYSTACK



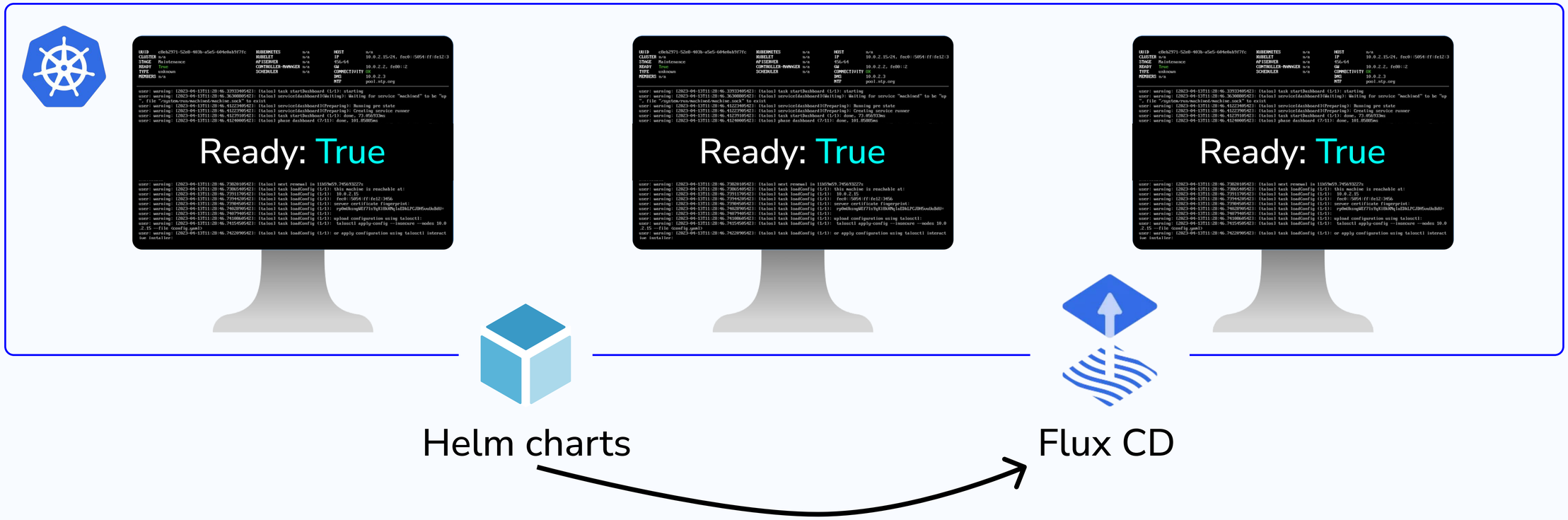
Установка платформы

COZYSTACK



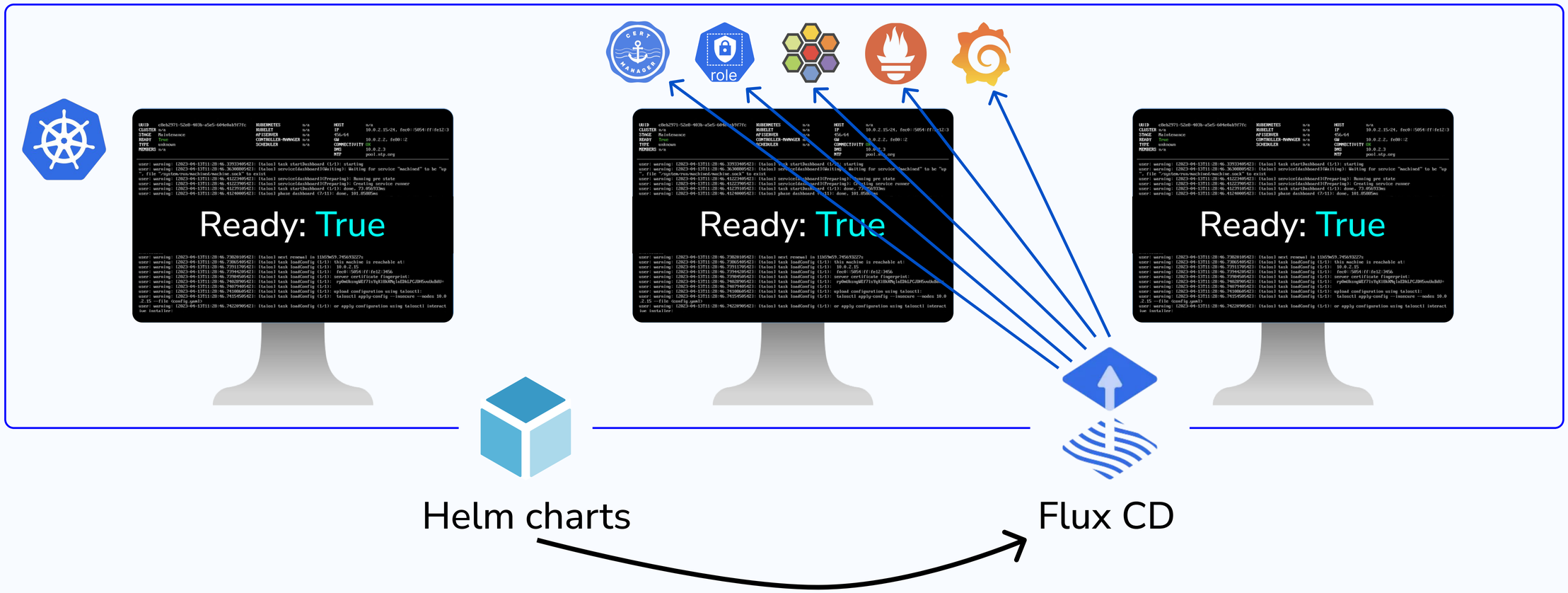
Установка платформы

COZYSTACK



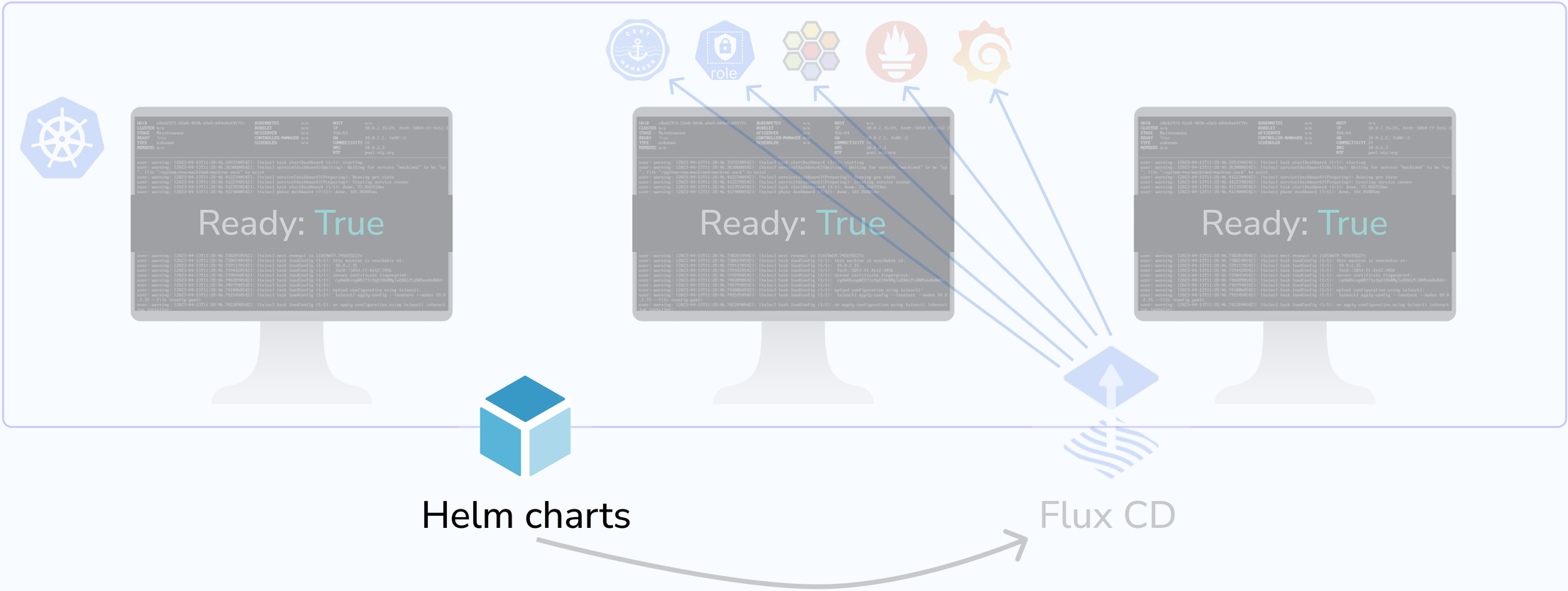
Установка платформы

COZYSTACK



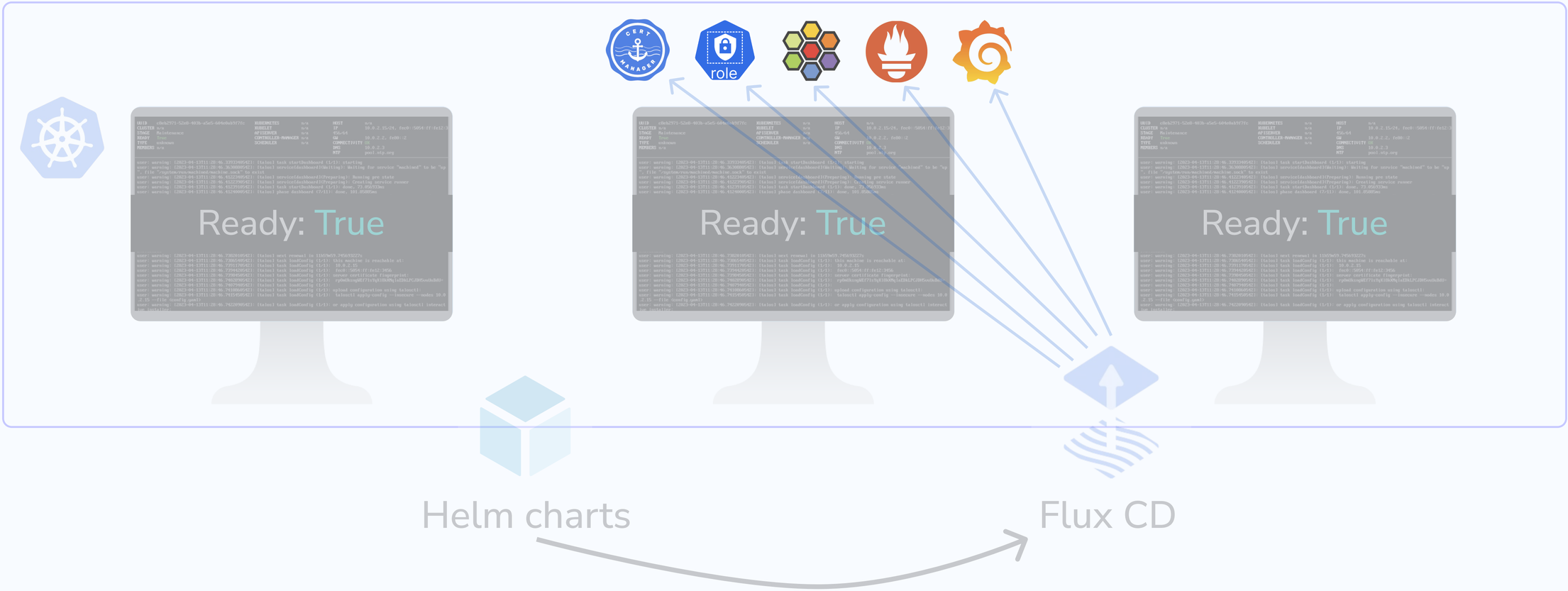
Установка платформы

COZYSTACK



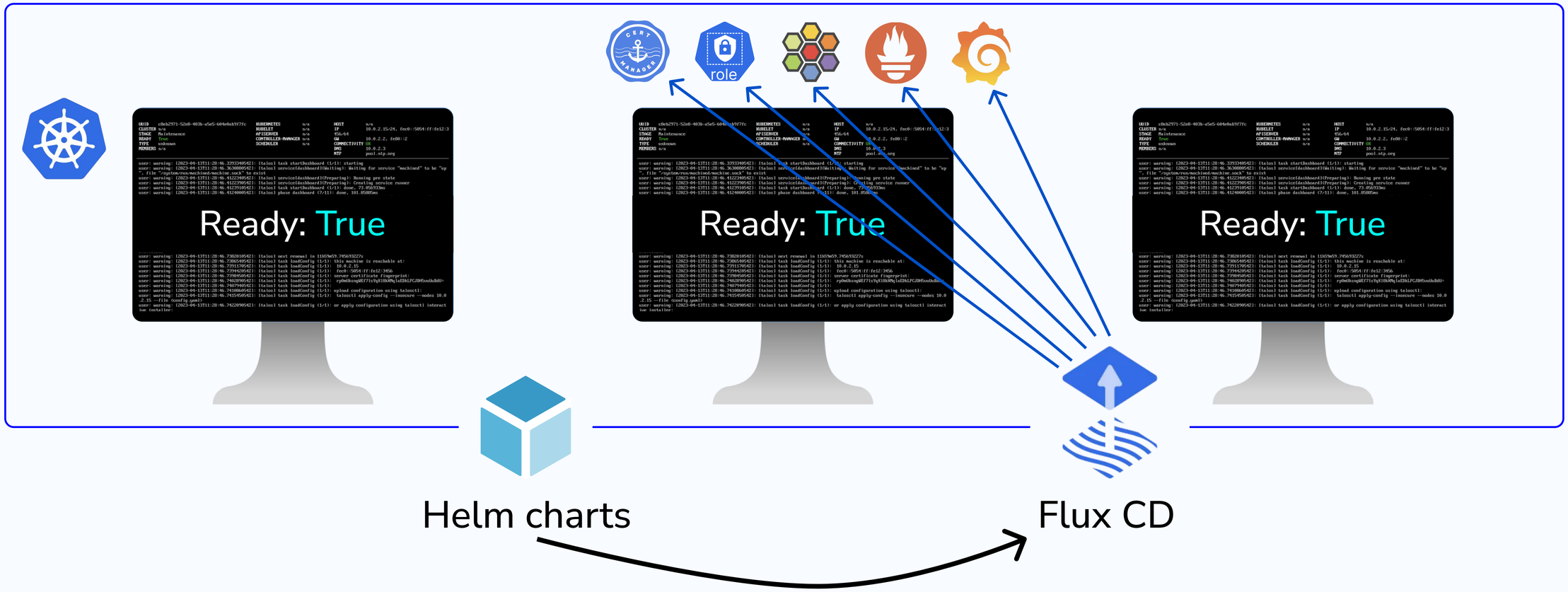
Установка платформы

COZYSTACK



Установка платформы

COZYSTACK

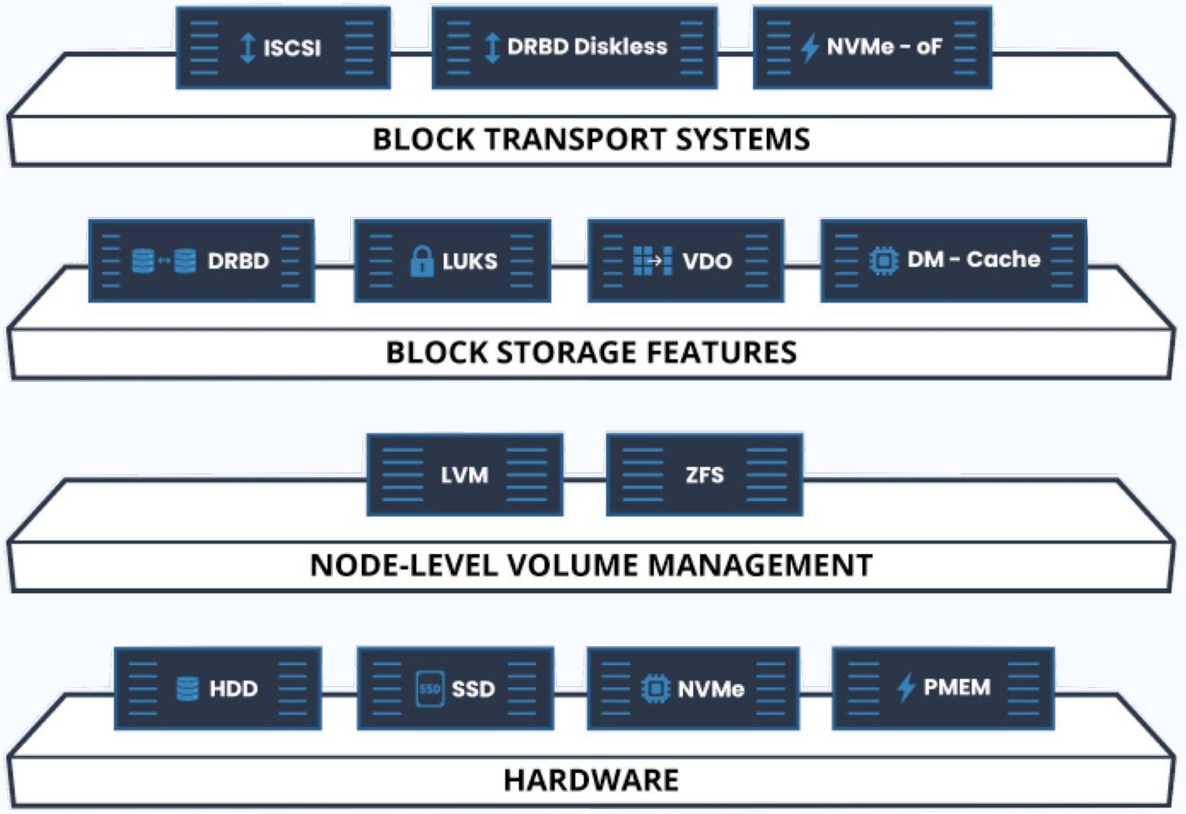


Быстрое и надежное хранилище

COZYSTACK

- Репликация в самом ядре linux
- Простой дизайн
- Проверенные временем технологии
- Готовые к использованию конфигурации

LINSTOR

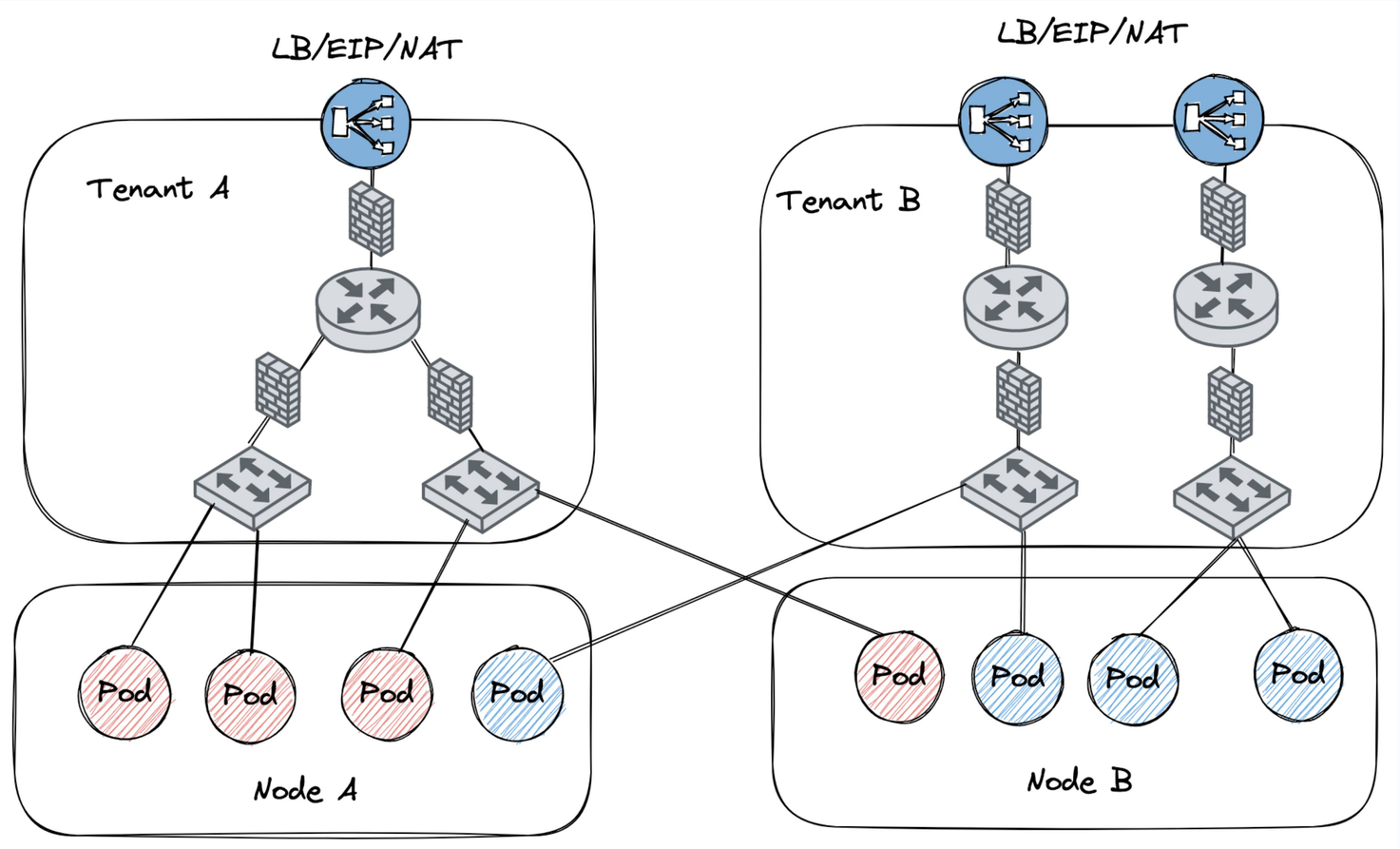


powered by LINSTOR

Современная сетевая фабрика

COZYSTACK

- Мульти-тенантность и VPCs (Virtual Personal Networks)
- Простая и бесшовная интеграция с существующей сетью
- Ускорение сети за счет eBPF и SR-IOV

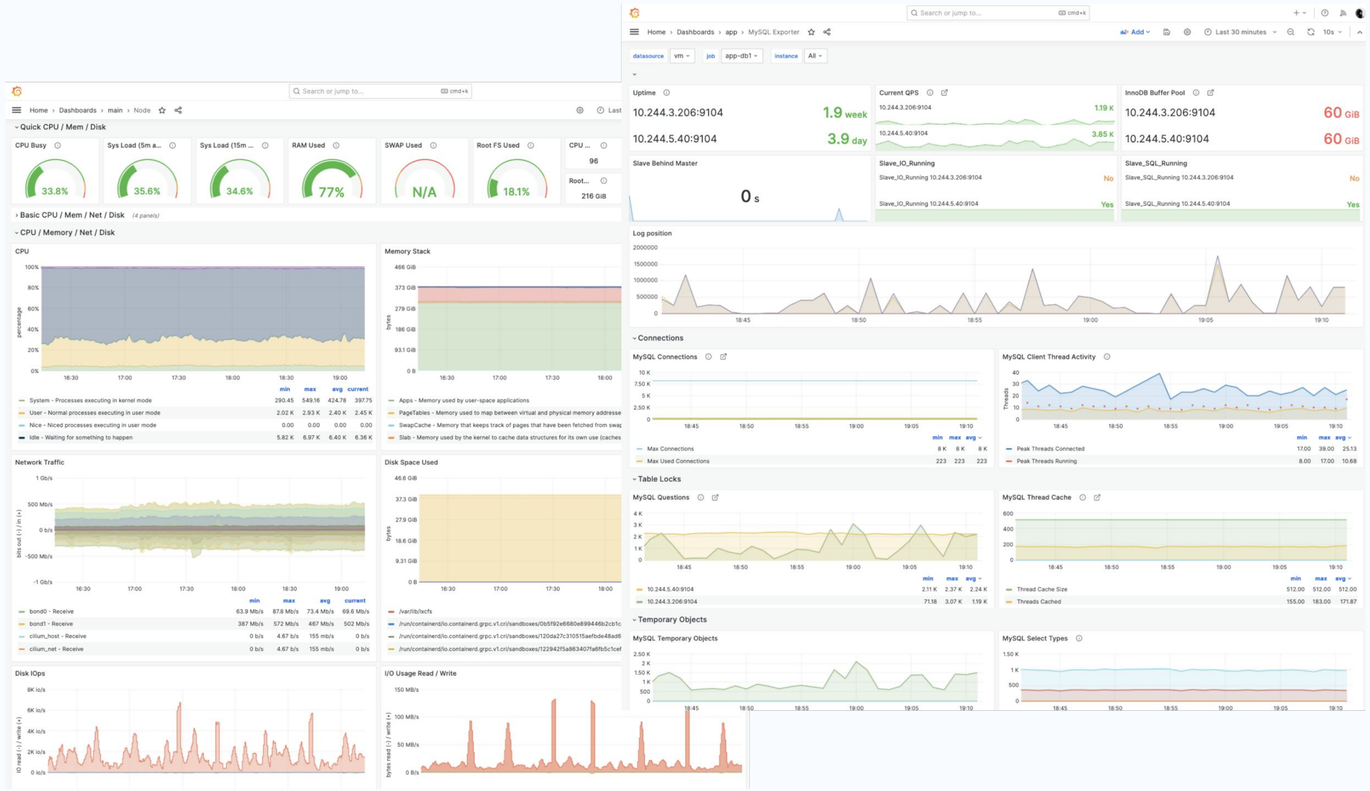


powered by cilium and Kube-OVN

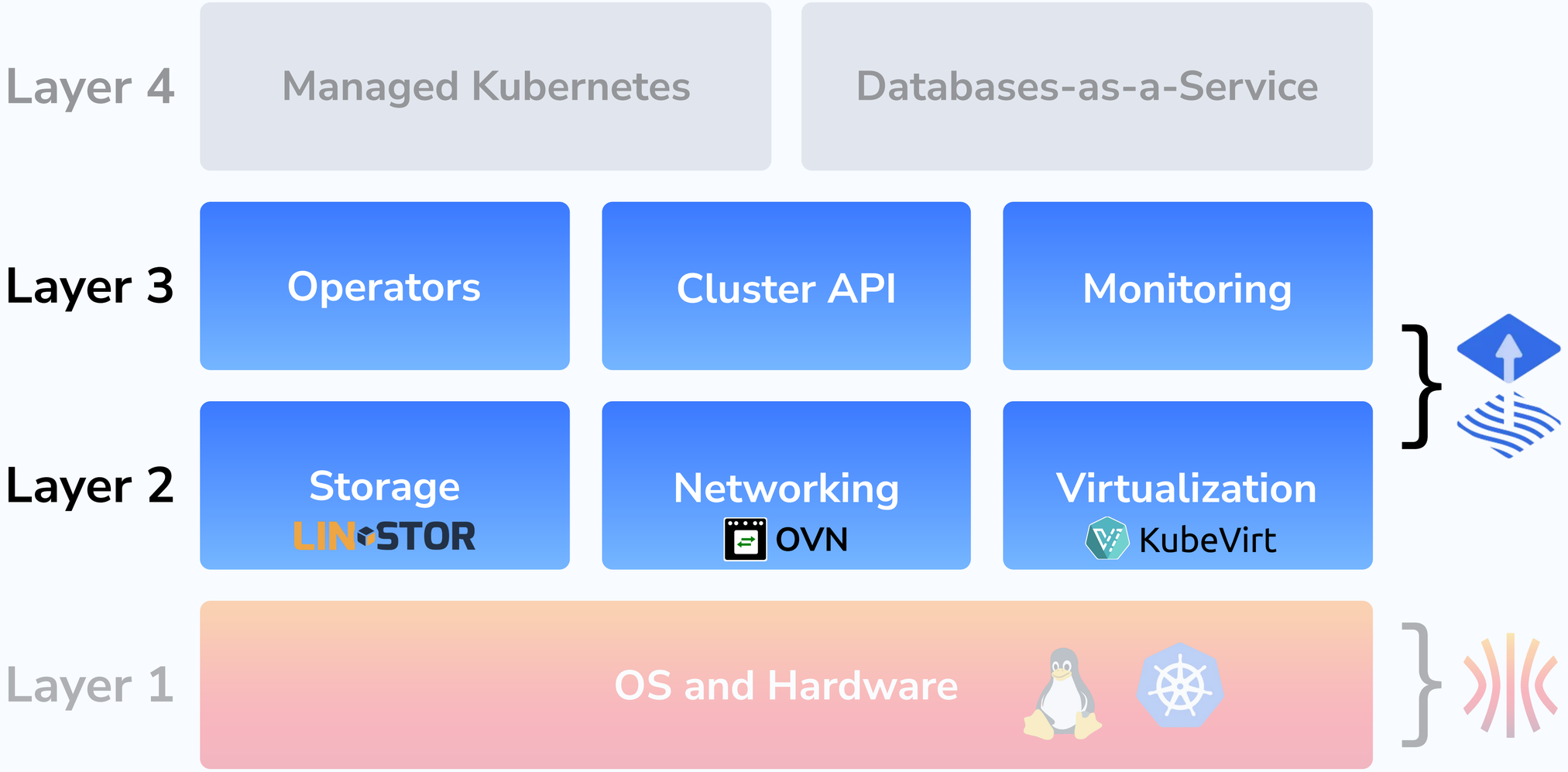
Мониторинг

COZYSTACK

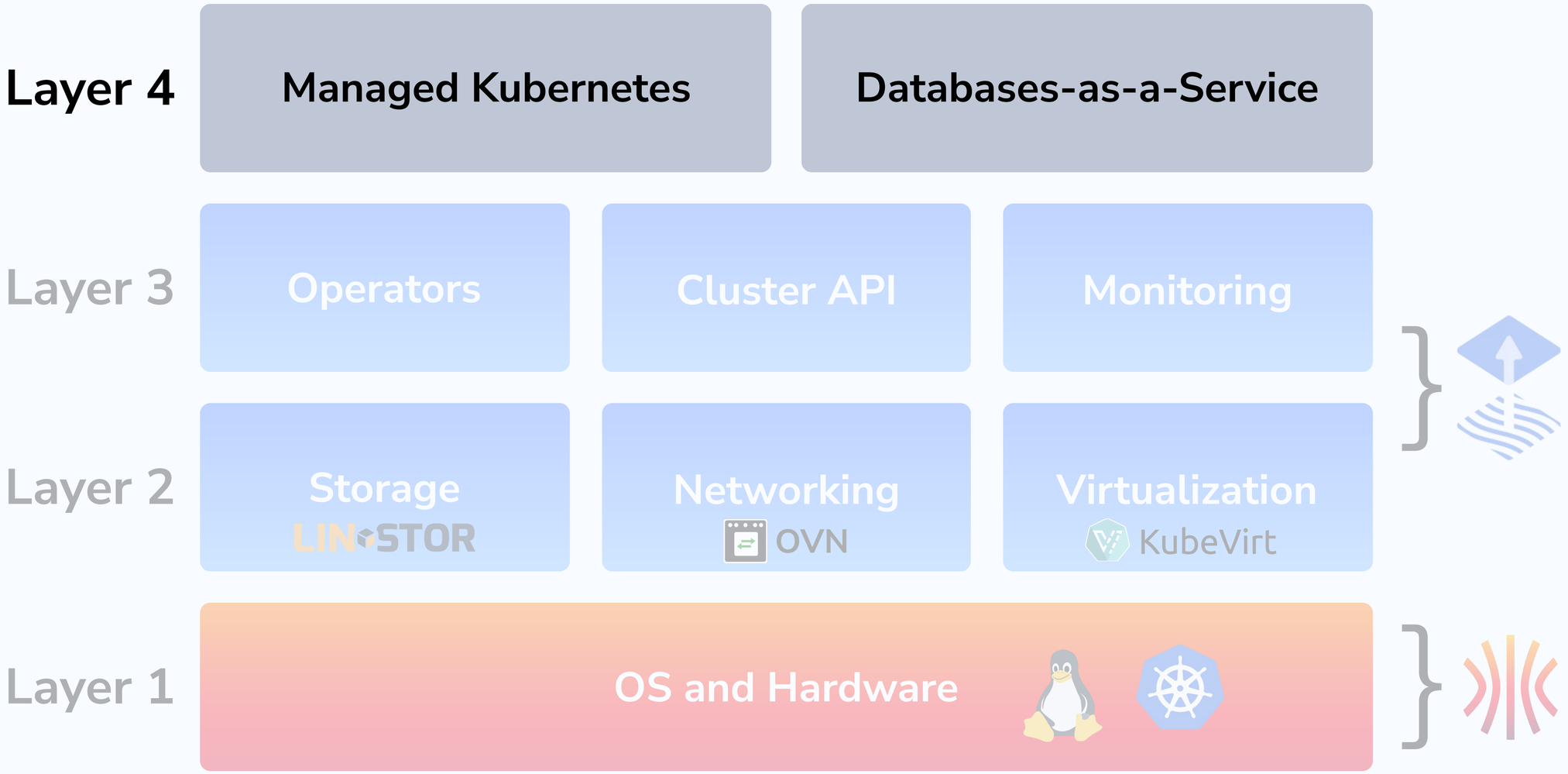
- Преднастроенные дашборды Grafana
- Все сервисы автоматически подключаются
- Встроенная система реакции на инциденты (IRM)

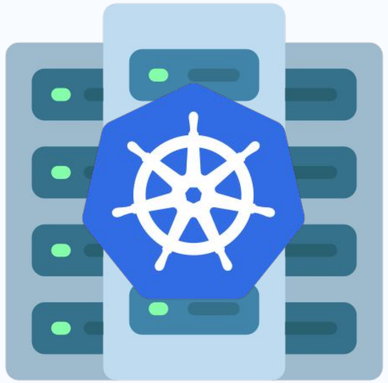


powered by VICTORIA METRICS


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

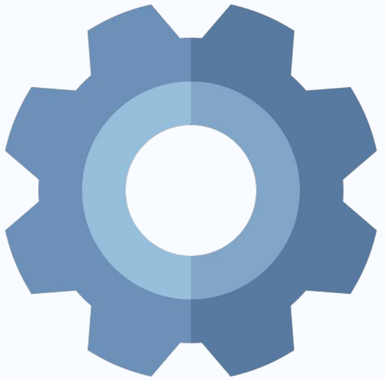
COZYSTACK


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

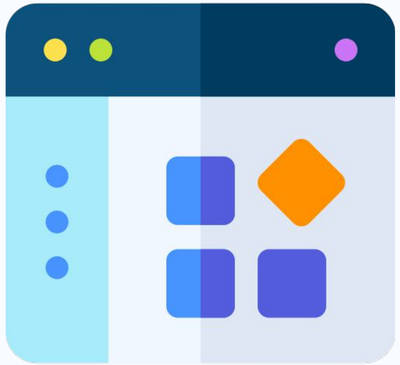


K8s на железе

Железо и ОС

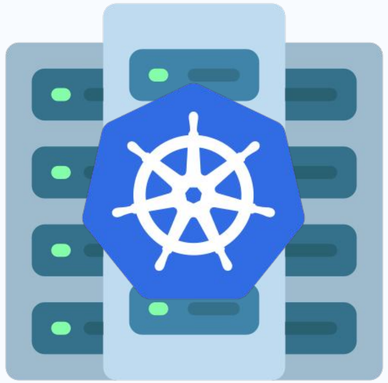


Системные компоненты



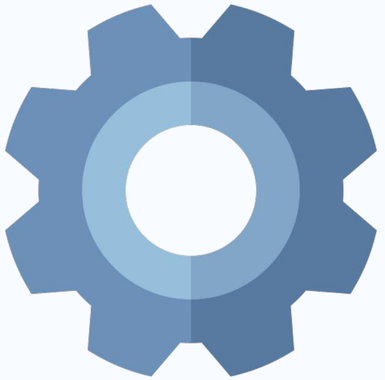
Пользовательские приложения

ПЛАТФОРМА

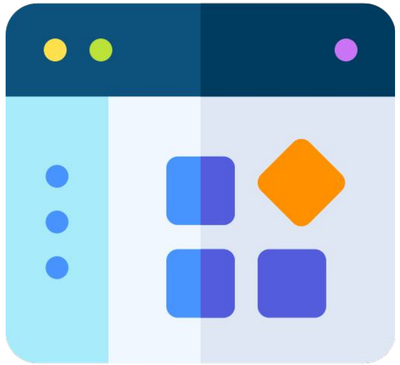


K8s на железе

Железо и ОС



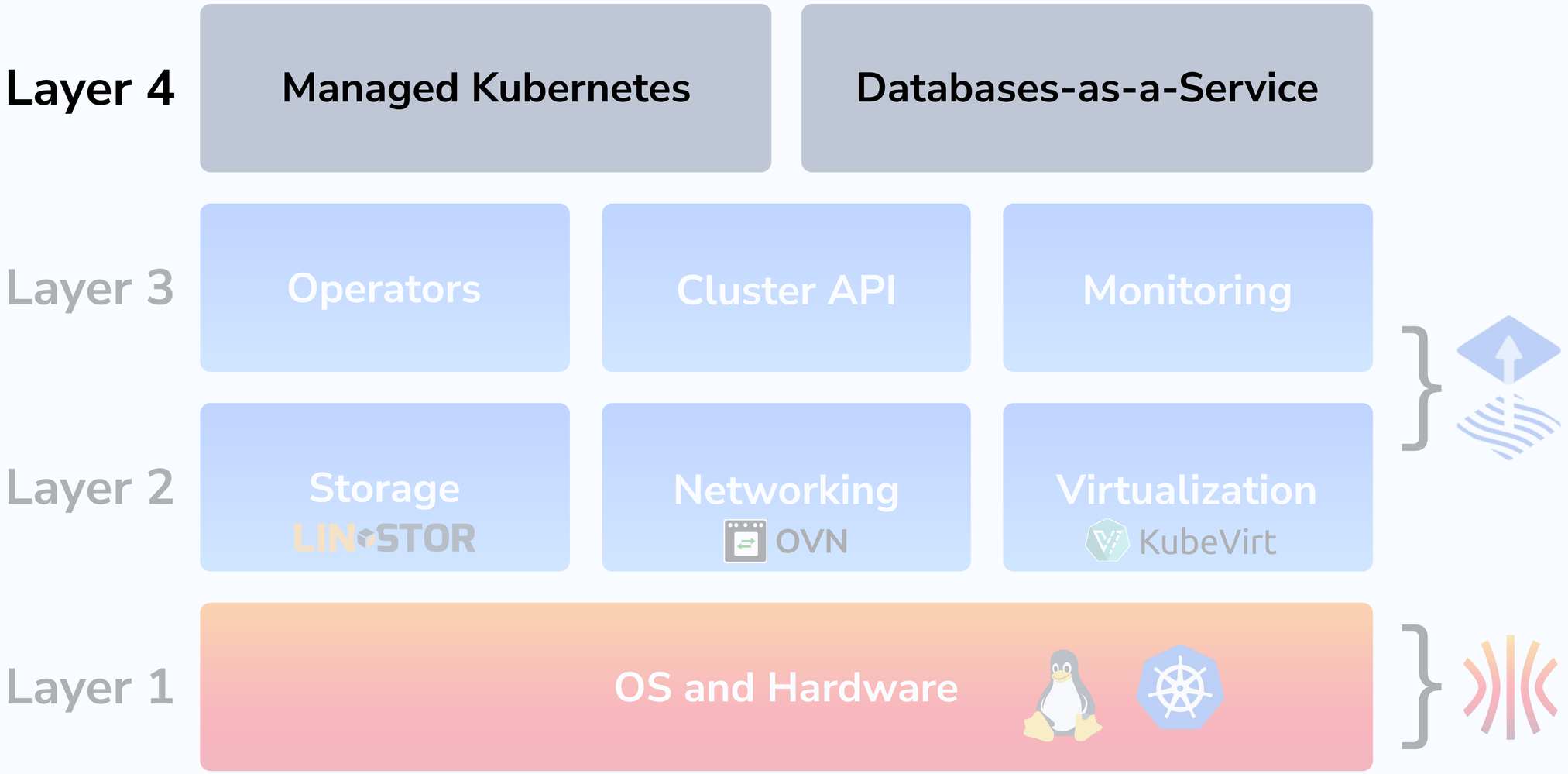
Системные компоненты



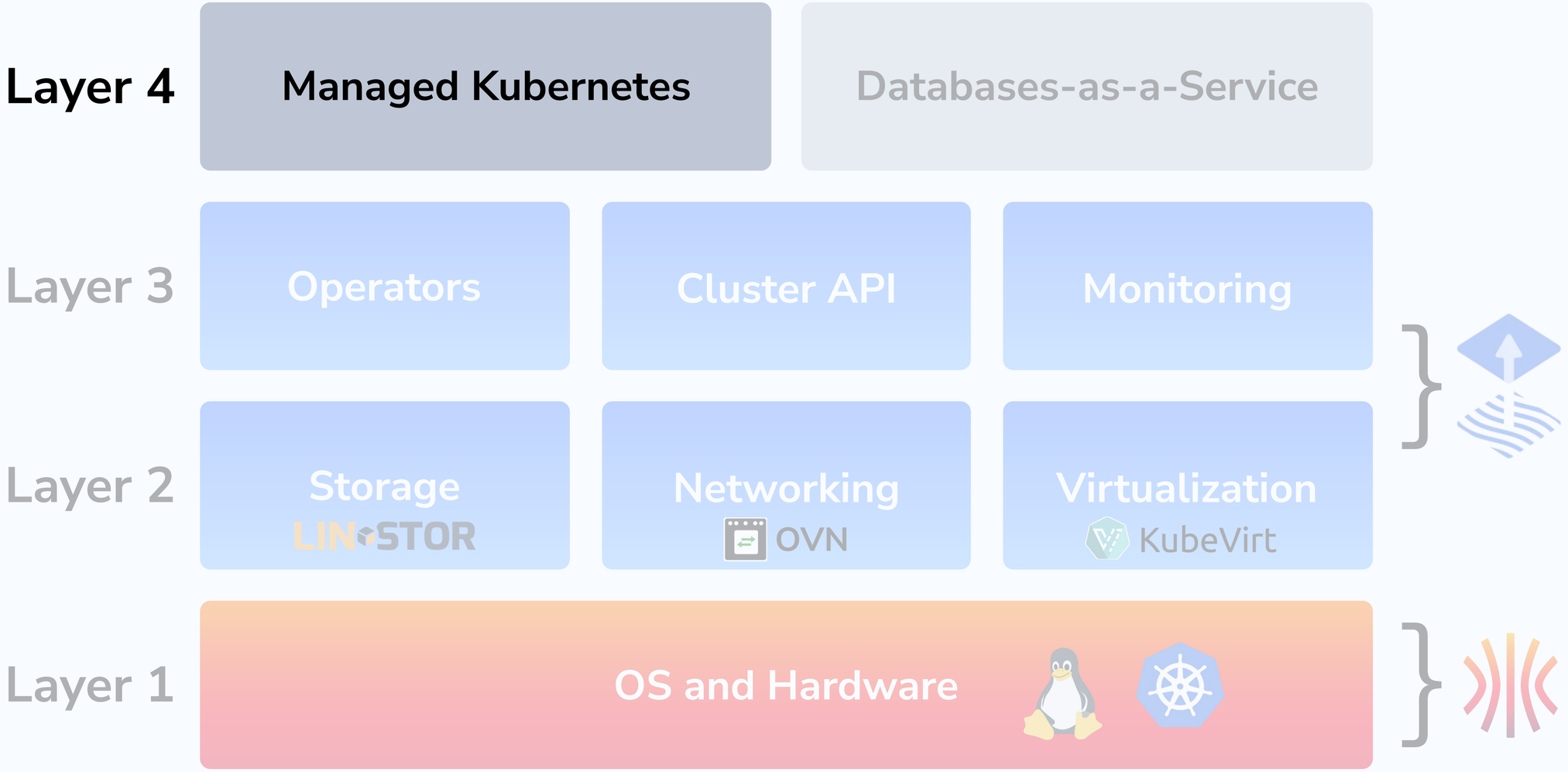
Пользовательские приложения

ПЛАТФОРМА

COZYSTACK


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

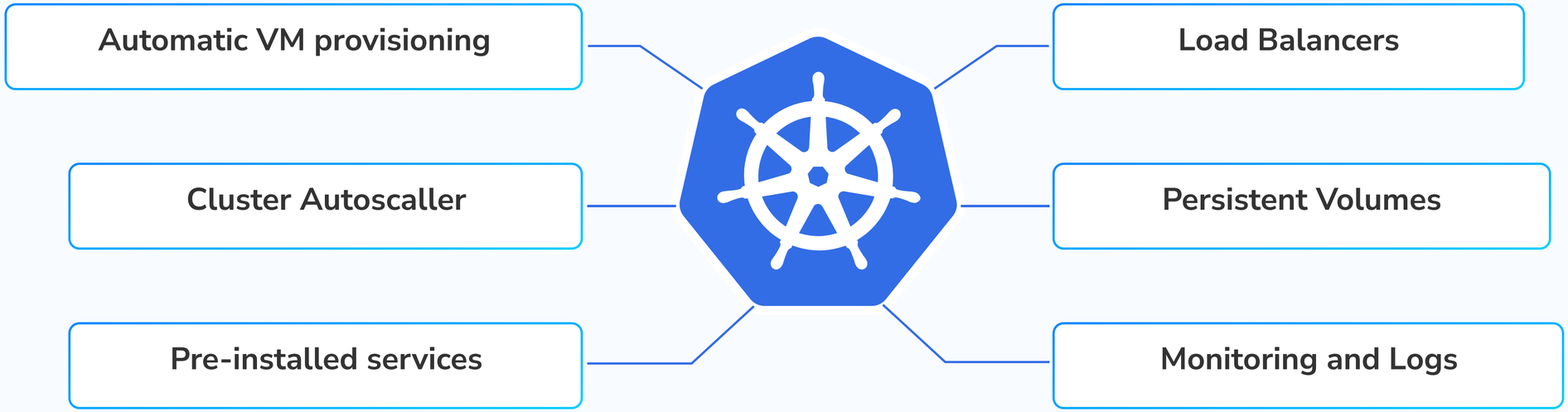
COZYSTACK


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

По-настоящему управляемый Kubernetes

COZYSTACK

Построить по-настоящему управляемый Kubernetes непросто.
Мы предлагаем простое и абсолютно полноценное решение на базе Kubernetes,
которое работает везде - и в облаках, и на своем железе

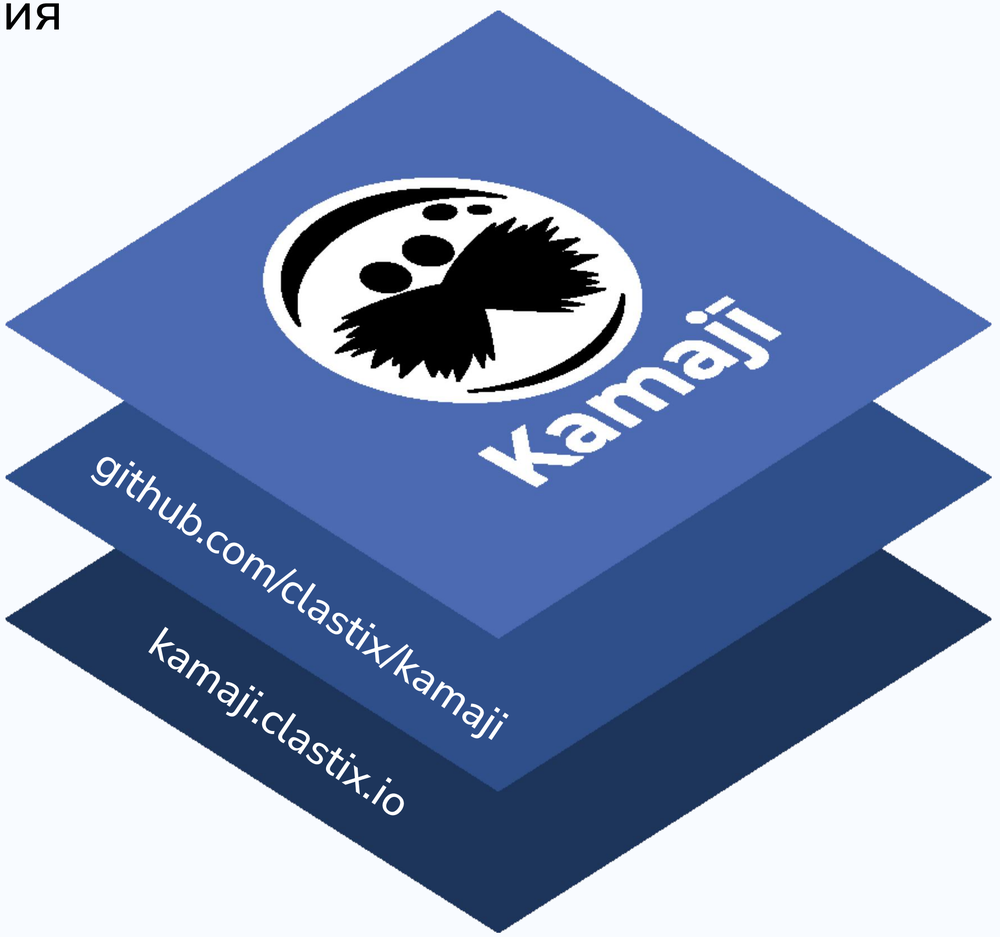


Знакомьтесь, Kamaji

Kubernetes оператор, предоставляющий Custom Resource Definitions для декларативного управления Tenant Control Plane.

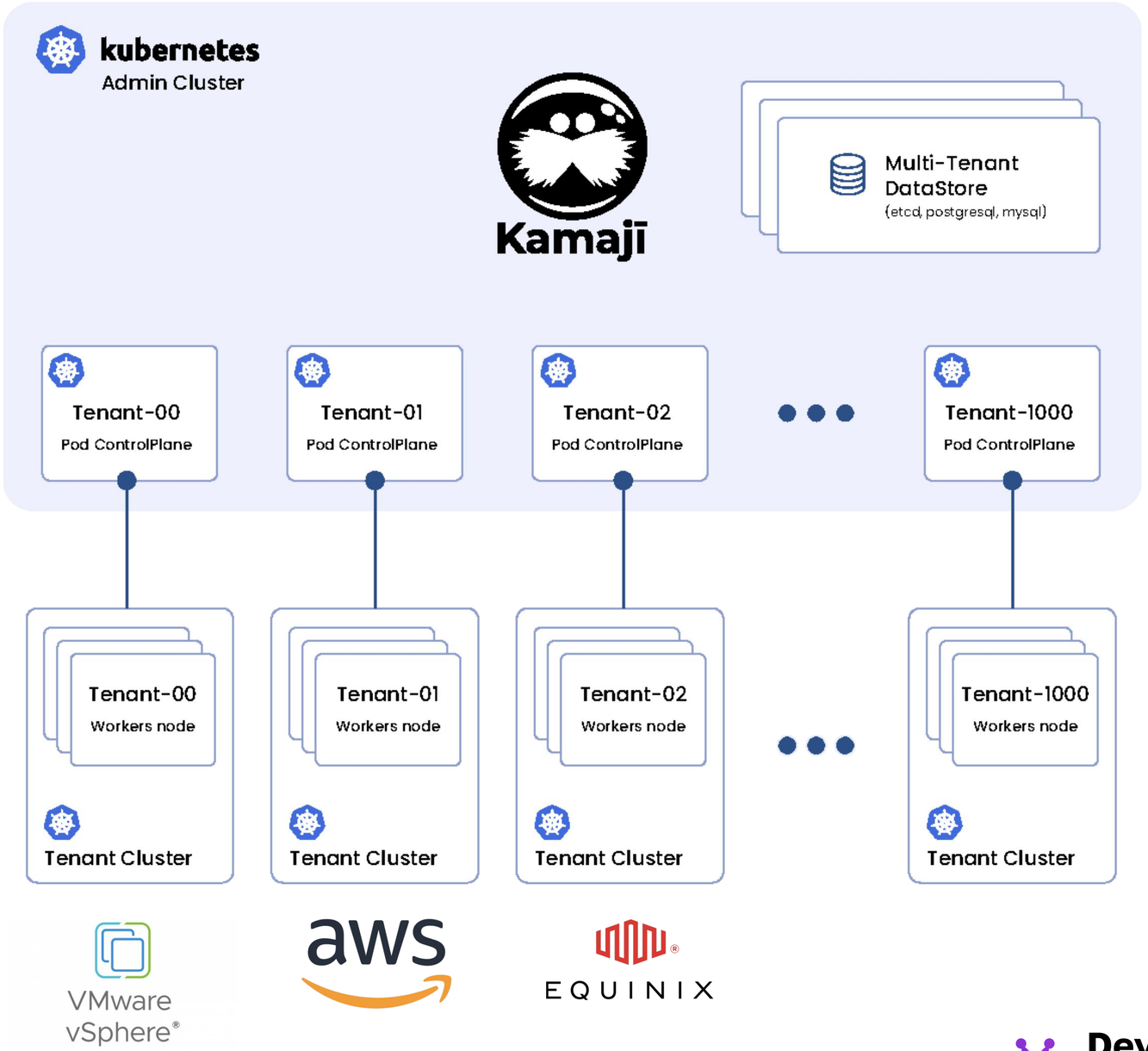
- Реализует day 2 operations
- Полностью декларативный (GitOps)
- Надежный и безопасный
- Дешевый

...и не забудьте поставить звездочку на GitHub! 🌟




HONEST REACTION
WHEN EXPLAINING THAT K8S CONTROL PLANES CAN RUN AS PODS
imgflip.com

- **Kubernetes as a Service**
  - Можно подключать свои узлы
  - Разные провайдеры
- **Обработка событий**
  - Высокая доступность
  - Обновление
  - Миграция на более емкое хранилище
  - Реконсилиация
- **Быстрый запуск**
  - запуск подов vs. виртуальные машины
- **Безопасный**
  - KataContainers
  - Выделенное хранилище
- **Дешевый**
  - потребление ресурсов (поды vs VM)



Cluster API

Clastix реализует интеграции через Cluster API с основными инфраструктурными провайдерами, чтобы предоставить пользователям полную свободу выбора (Bring Your Own Infrastructure - BYOI). И в будущем интеграций будет еще больше!

KubeVirt

EQUINIX

VMware vSphere®

Архитектура

Будучи оператором Kubernetes предоставляет Custom Resource Definition для декларативного описания control plane.

CRDs

- **TenantControlPlane (tcp)**

  Управляет control plane

- **DataStore**

  Для надежности и оптимизации стоимости

Kubernetes app


```
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```

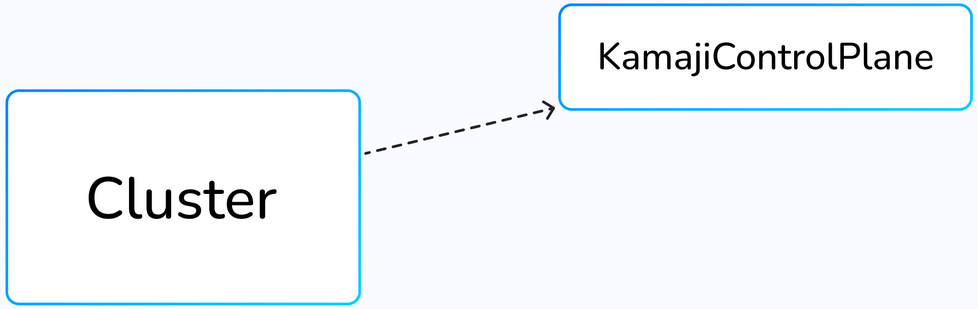
Kubernetes app


```
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```

Cluster

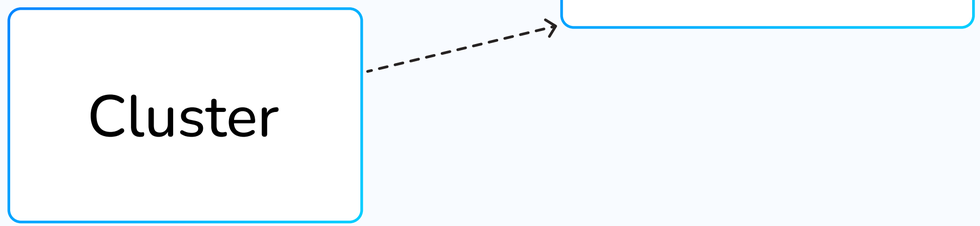
Kubernetes app


```
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



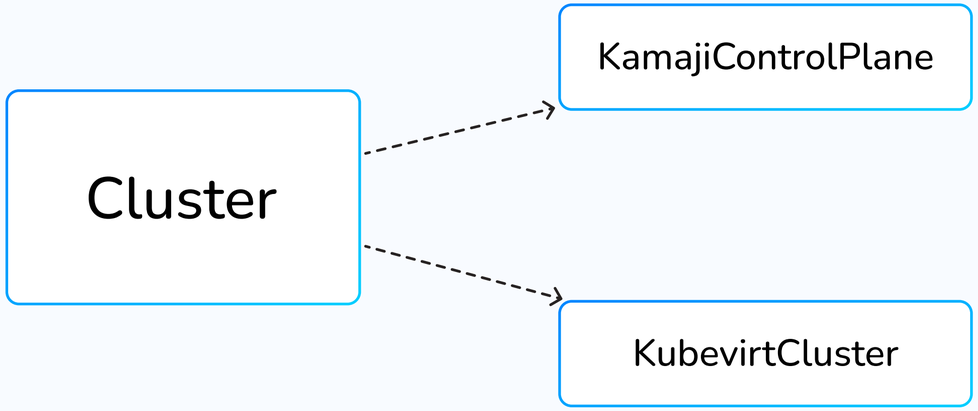
Kubernetes app


```yaml
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



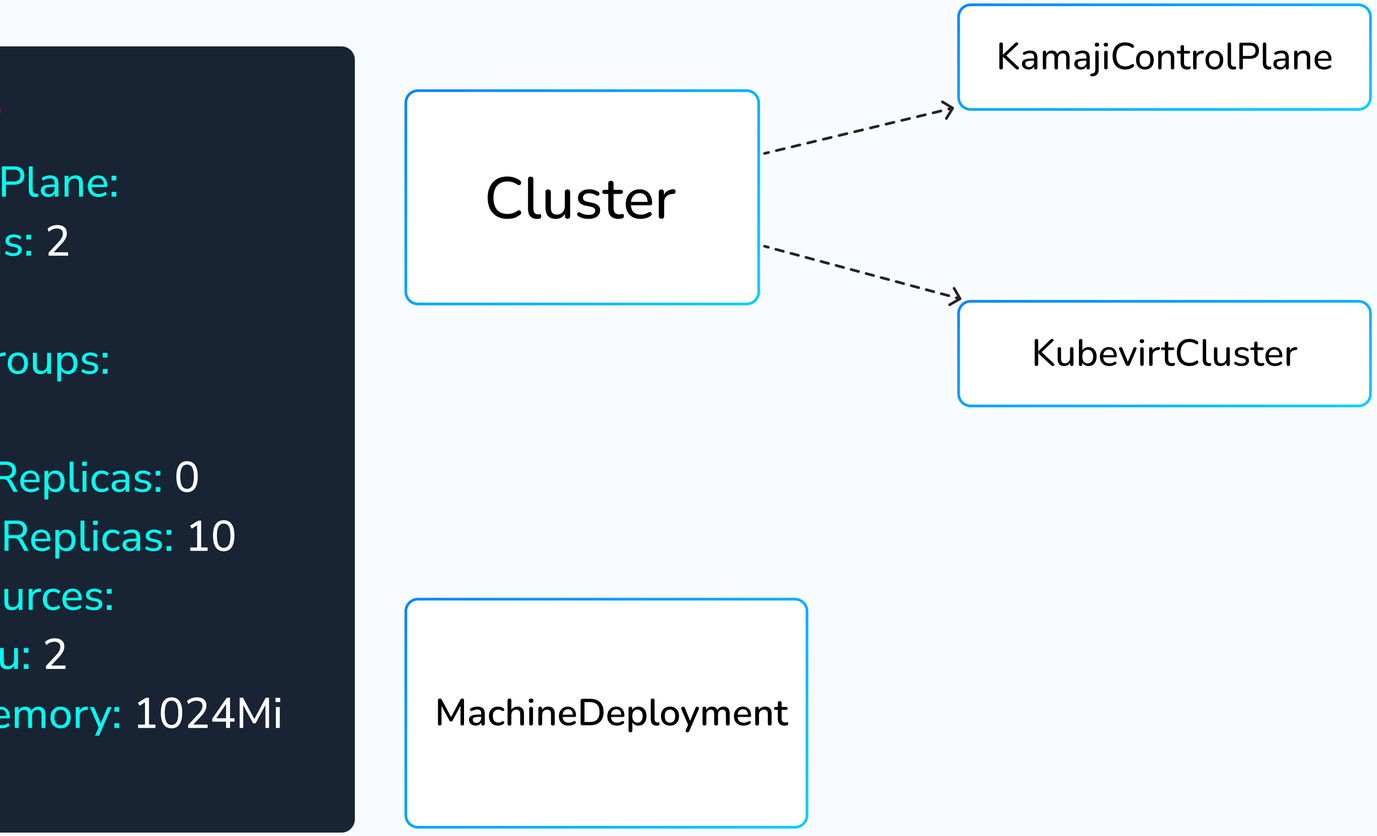
Kubernetes app


```yaml
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



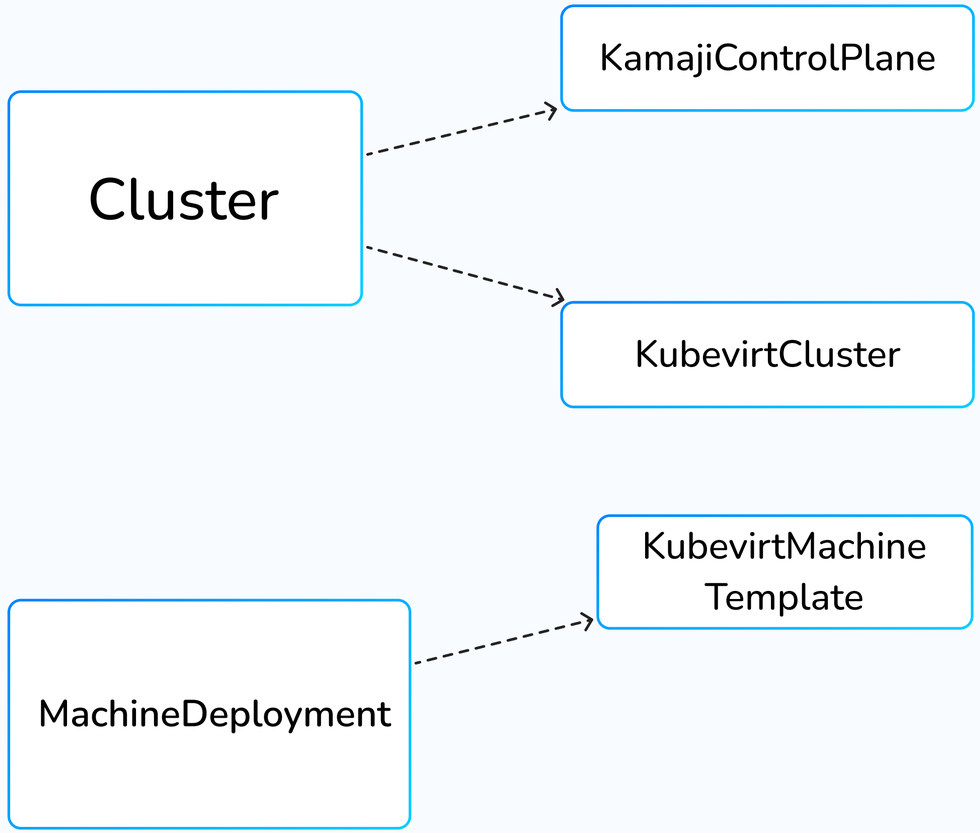
Kubernetes app


```
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



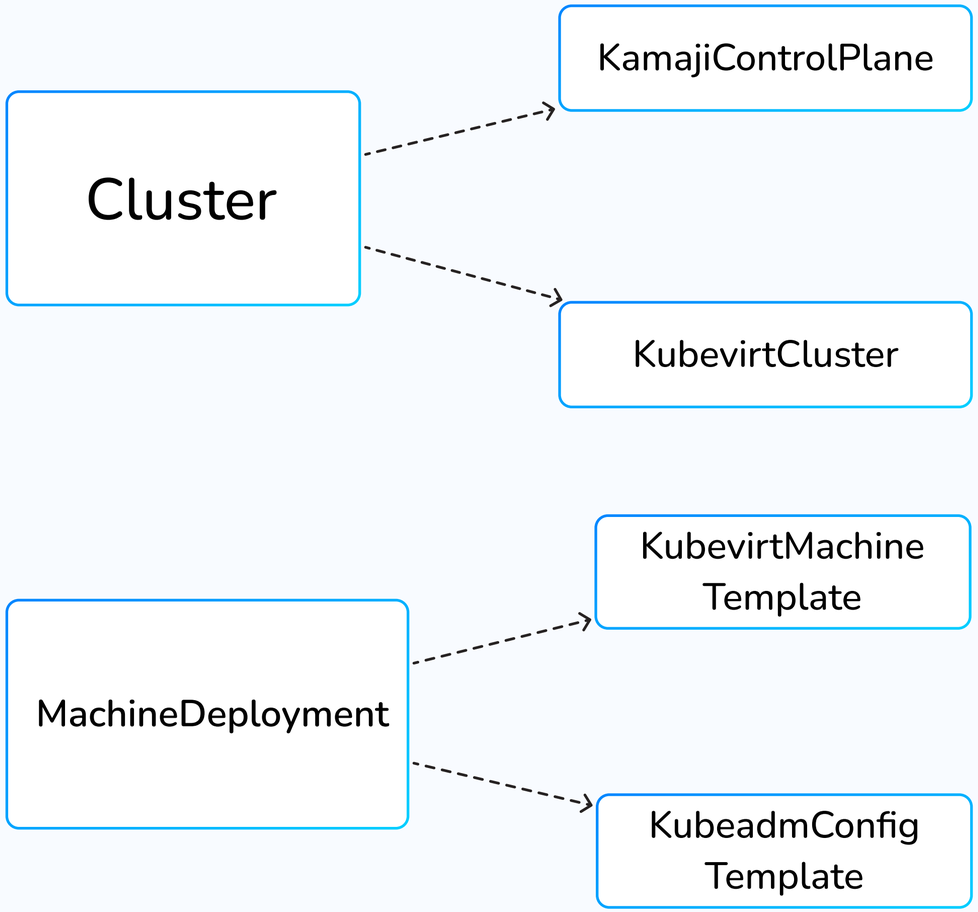
Kubernetes app


```
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



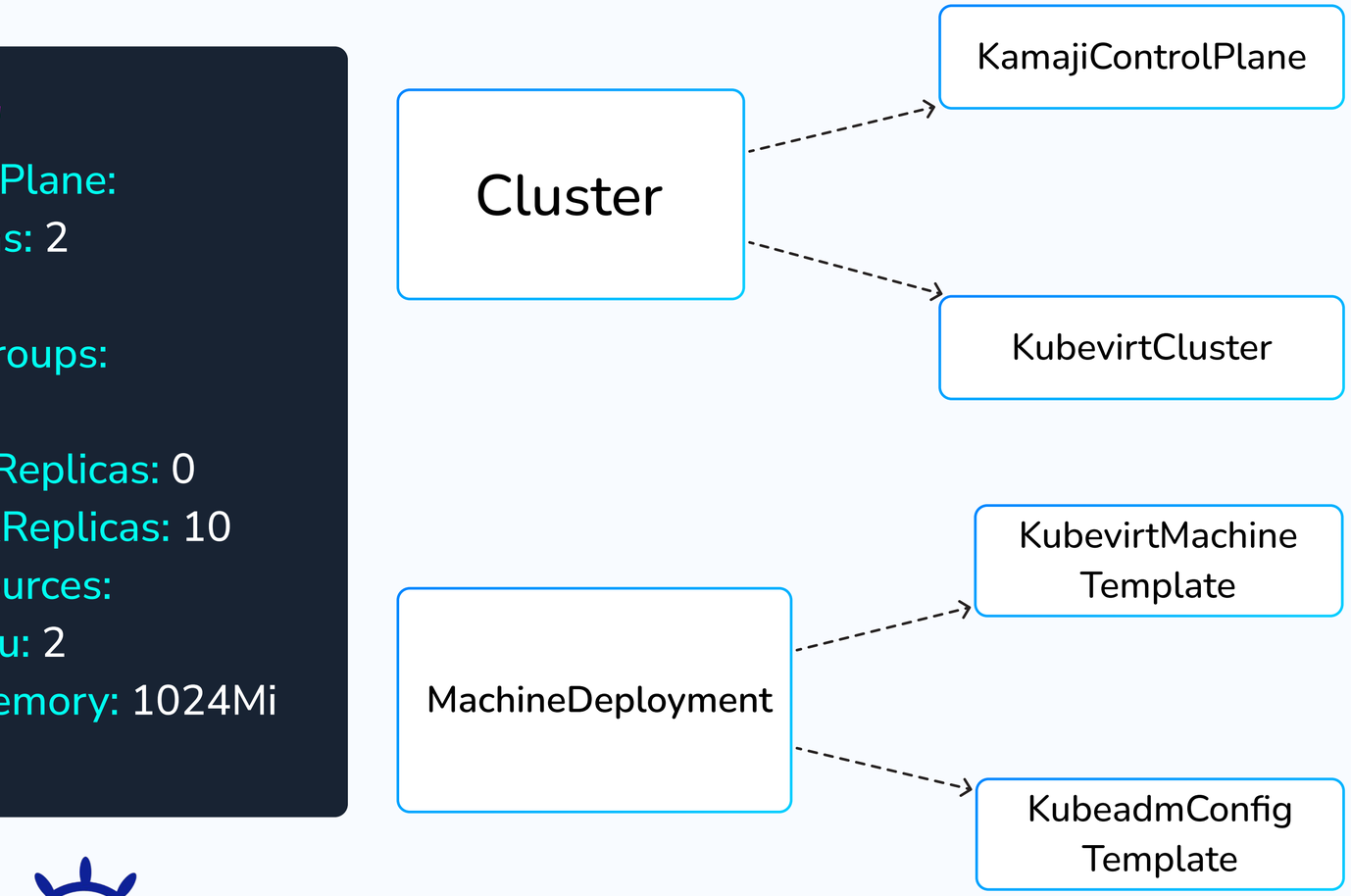
Kubernetes app


```yaml
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



Kubernetes app

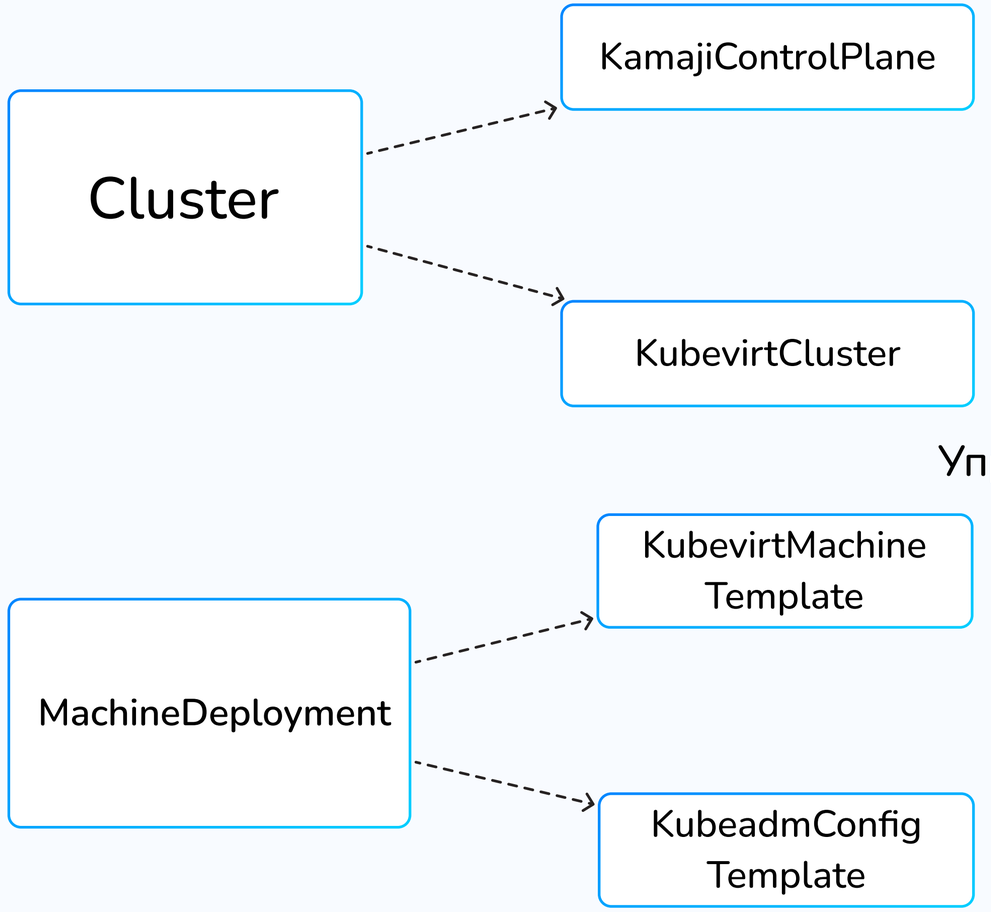

```
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



Управляется control-plane провайдером

Kubernetes app


```yaml
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```

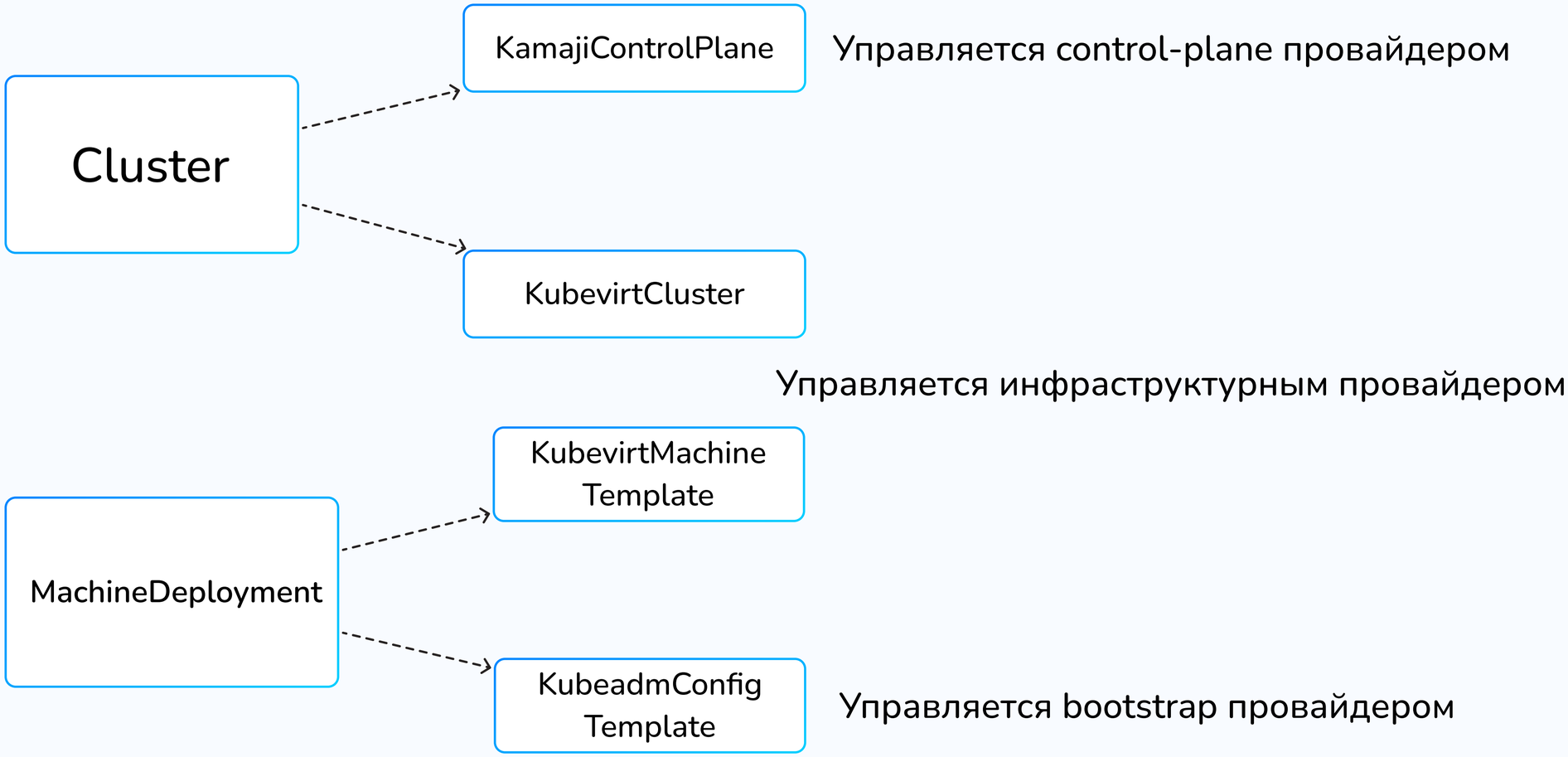


Управляется control-plane провайдером

Управляется инфраструктурным провайдером

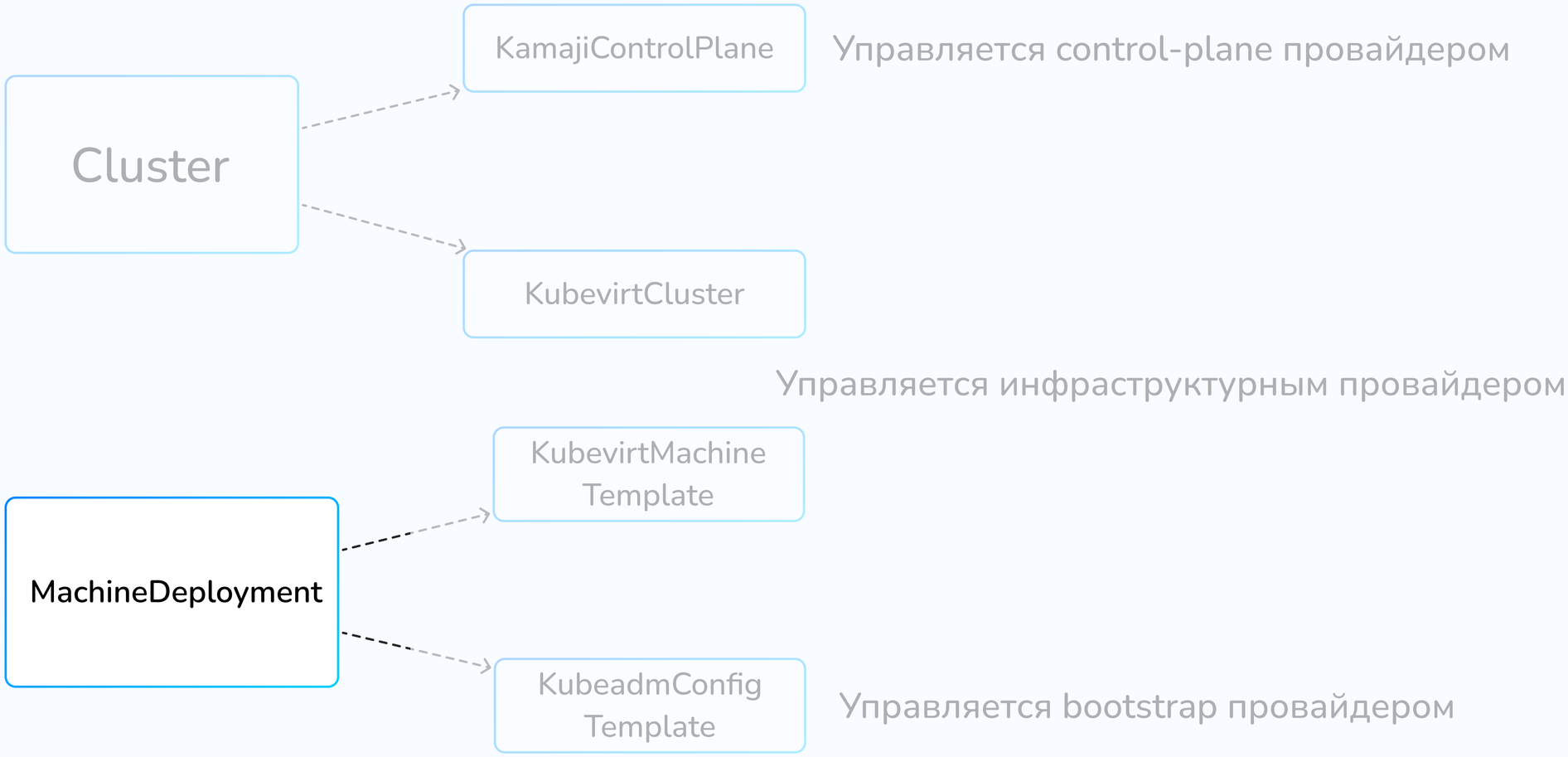
Kubernetes app


```
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



Kubernetes app

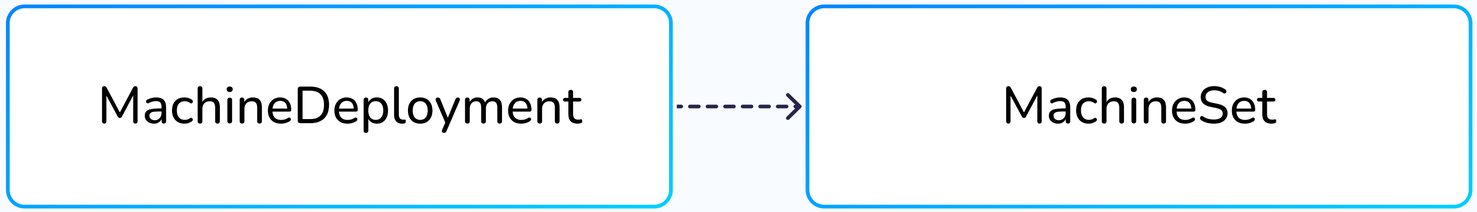

```
host: ""
controlPlane:
 replicas: 2

nodeGroups:
  md0:
    minReplicas: 0
    maxReplicas: 10
    resources:
      cpu: 2
      memory: 1024Mi
```



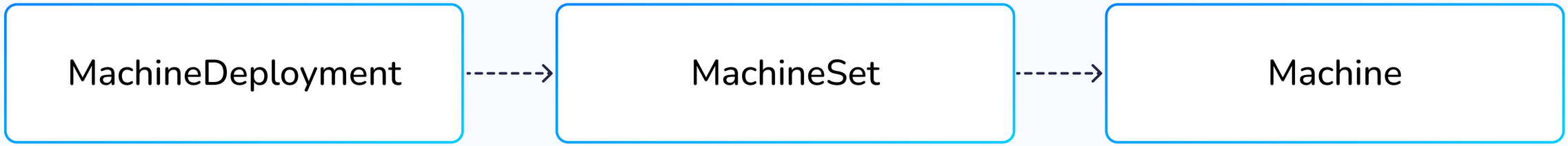
Kubernetes app

MachineDeployment

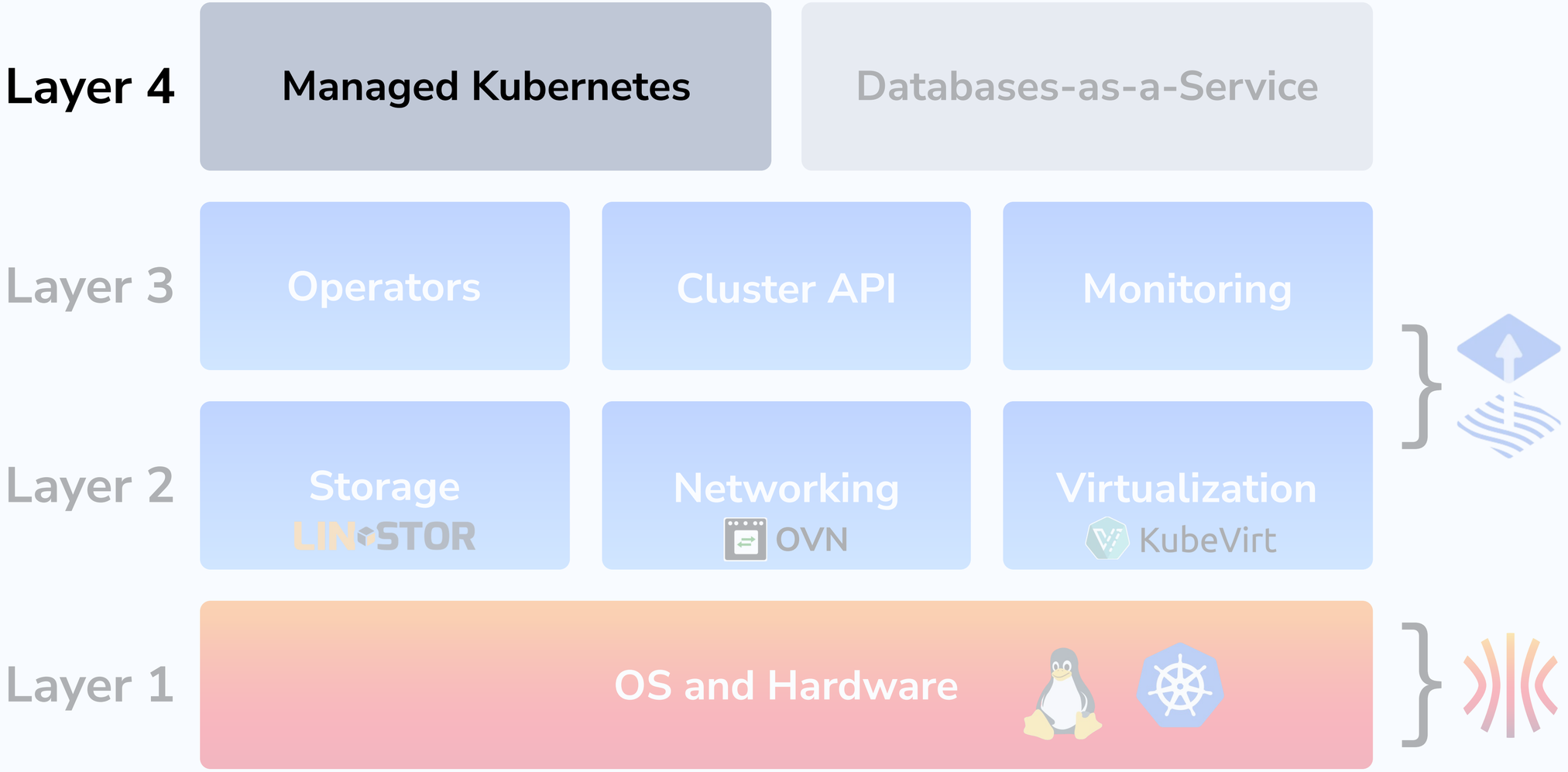
Kubernetes app

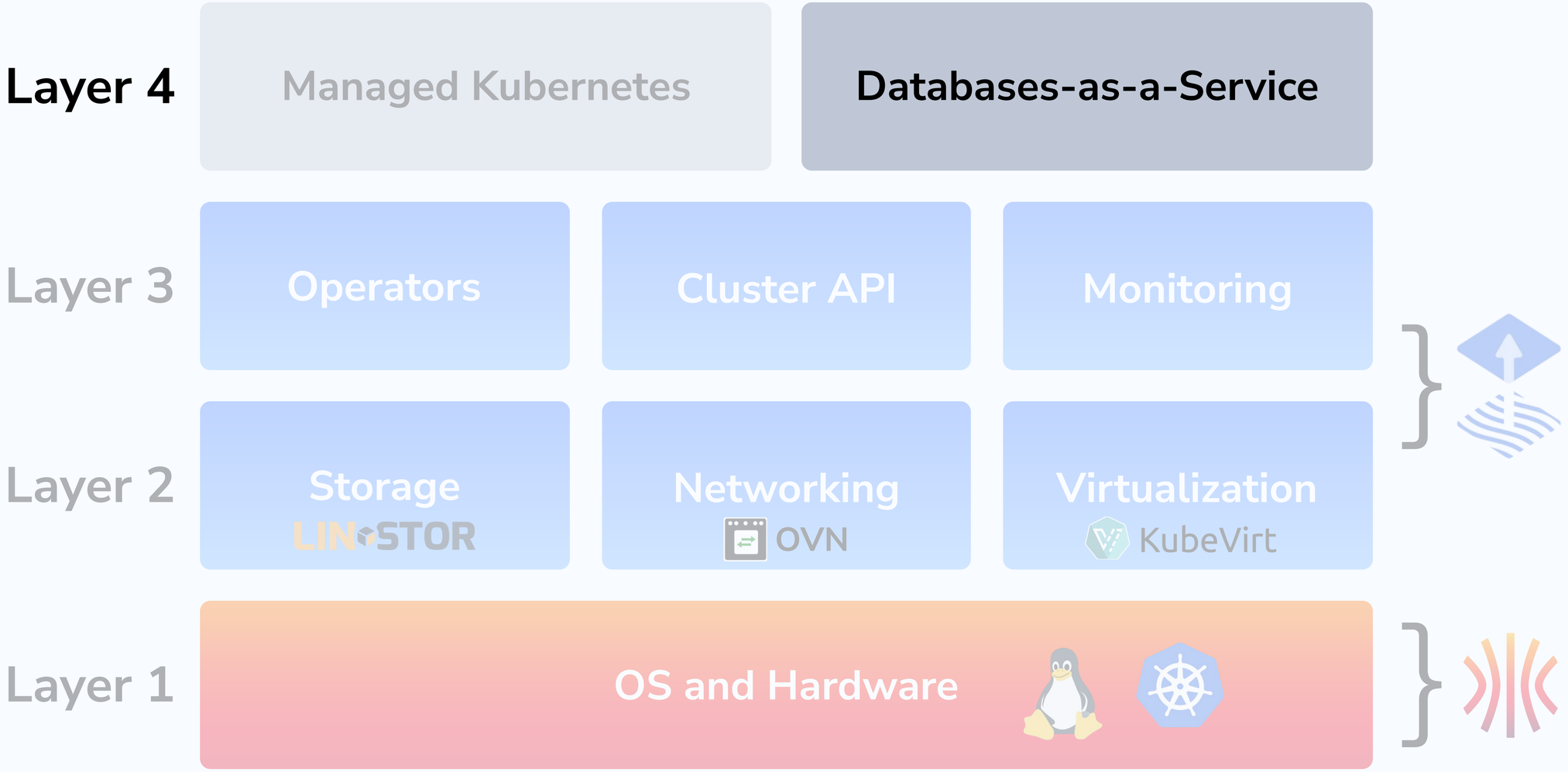


Kubernetes app



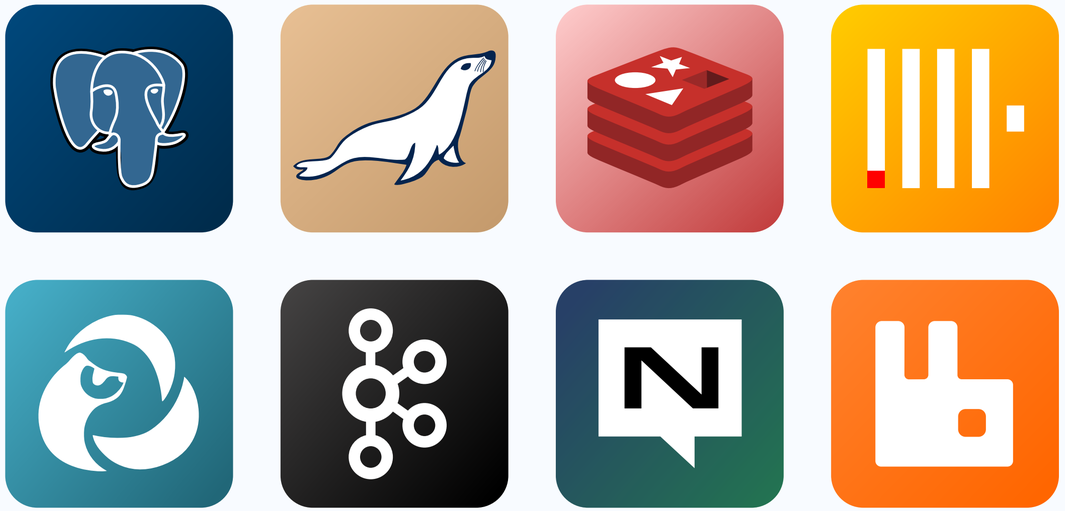
COZYSTACK


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware


Layer 4
Managed Kubernetes
Databases-as-a-Service
Layer 3
Operators
Cluster API
Monitoring
Layer 2
Storage
LINSTOR
Networking
OVN
Virtualization
KubeVirt
Layer 1
OS and Hardware

Управляемые базы данных

Управляются и настраиваются операторами.
Высокодоступны с самого старта.



Возможности:

- Создание баз данных
- Управление пользователями и доступами
- Настройка автоматического резервного копирования
- Мониторинг (дашборды Grafana в комплекте)
- Оповещения о важных событиях и сбоях

cloudnative-pg/
cloudnative-pg

CloudNativePG is a comprehensive platform designed to seamlessly manage PostgreSQL databases within Kubernetes environments, covering the entire operational lifecycle from...

strimzi/**strimzi-kafka-operator**

spotahome/**redis-operator**

Redis Operator creates/configures/manages high availability redis with sentinel automatic failover atop Kubernetes.

Altinity/**clickhouse-operator**

Altinity Kubernetes Operator for ClickHouse creates, configures and manages ClickHouse clusters running on Kubernetes

mariadb-operator/
mariadb-operator

🦭 Run and operate MariaDB in a cloud native way

rabbitmq/**cluster-operator**

RabbitMQ Cluster Kubernetes Operator

Catalog

search packages...

263

FILTERS

Package Type

- [ ] Helm Charts via Flux

bucket

S3 compatible storage

0.1.0 cozystack-apps Flux

clickhouse

Managed ClickHouse service

0.3.0 cozystack-apps Flux

ferretdb

Managed FerretDB service

0.2.0 cozystack-apps Flux

http-cache

Layer7 load balacner and caching service

0.3.0 cozystack-apps Flux

kafka

Managed Kafka service

0.3.0 cozystack-apps Flux

kubernetes

Managed Kubernetes service

0.9.0 cozystack-apps Flux

mysql

Managed MariaDB service

0.4.0 cozystack-apps Flux

nats

Managed NATS service

0.2.0 cozystack-apps Flux

postgres

Managed PostgreSQL service

0.5.0 cozystack-apps Flux

rabbitmq

Managed RabbitMQ service

0.3.0 cozystack-apps Flux

redis

Managed Redis service

0.3.0 cozystack-apps Flux

tcp-balancer

Layer4 load balancer service

0.2.0 cozystack-apps Flux

tenant

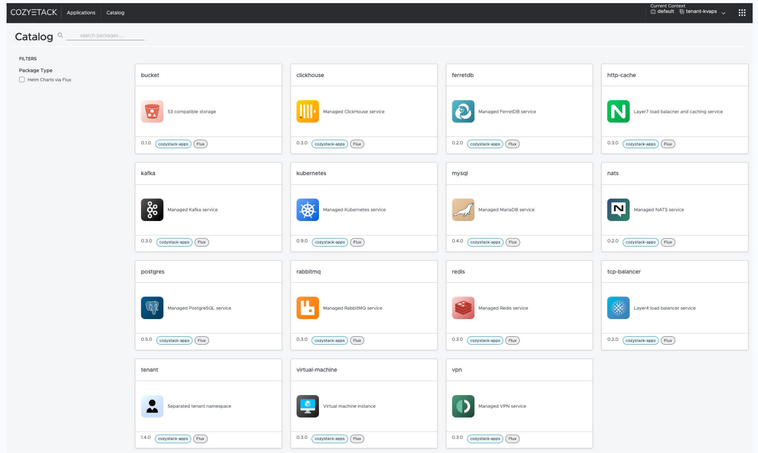
Separated tenant namespace

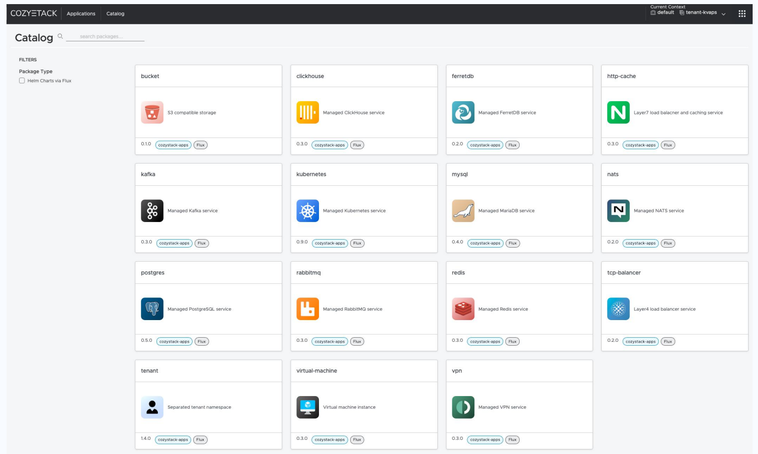
virtual-machine

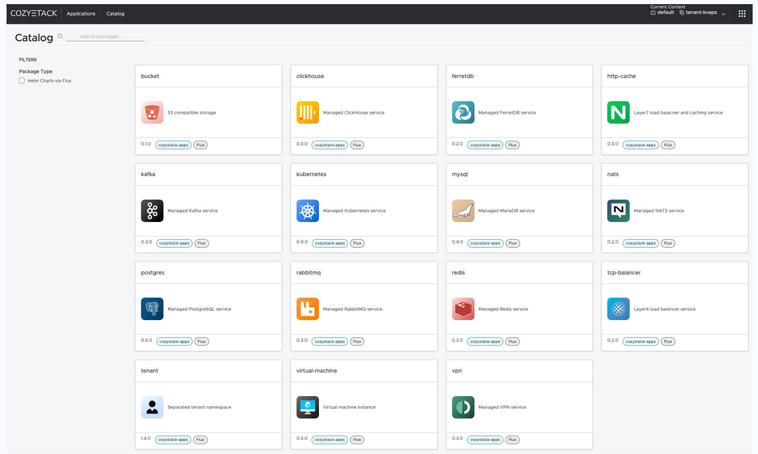
Virtual machine instance

vpn

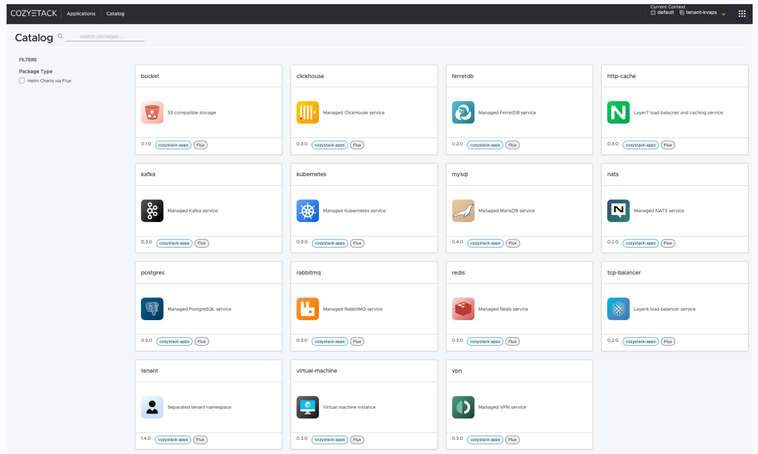
Managed VPN service

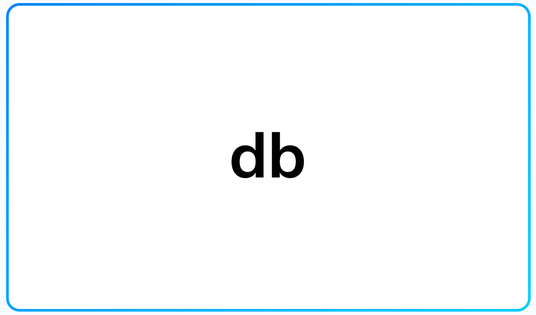

COZYSTACK
Applications
Catalog
Current Context
default
tenant-kvaps
Catalog
search packages...
FILTERS
Package Type
Helm Charts via Flux
bucket
S3 compatible storage
0.1.0
cozystack-apps
Flux
clickhouse
Managed ClickHouse service
0.3.0
cozystack-apps
Flux
ferretdb
Managed FerretDB service
0.2.0
cozystack-apps
Flux
http-cache
Layer7 load balacner and caching service
0.3.0
cozystack-apps
Flux
kafka
Managed Kafka service
0.3.0
cozystack-apps
Flux
kubernetes
Managed Kubernetes service
0.9.0
cozystack-apps
Flux
mysql
Managed MariaDB service
0.4.0
cozystack-apps
Flux
nats
Managed NATS service
0.2.0
cozystack-apps
Flux
postgres
Managed PostgreSQL service
0.5.0
cozystack-apps
Flux
rabbitmq
Managed RabbitMQ service
0.3.0
cozystack-apps
Flux
redis
Managed Redis service
0.3.0
cozystack-apps
Flux
tcp-balancer
Layer4 load balancer service
0.2.0
cozystack-apps
Flux
tenant
Separated tenant namespace
1.4.0
cozystack-apps
Flux
virtual-machine
Virtual machine instance
0.3.0
cozystack-apps
Flux
vpn
Managed VPN service
0.3.0
cozystack-apps
Flux


COZYSTACK
Applications
Catalog
Current Context
default
tenant-kvaps
Catalog
search packages...
FILTERS
Package Type
Helm Charts via Flux
bucket
S3 compatible storage
0.1.0
cozystack-apps
Flux
clickhouse
Managed ClickHouse service
0.3.0
cozystack-apps
Flux
ferretdb
Managed FerretDB service
0.2.0
cozystack-apps
Flux
http-cache
Layer7 load balacner and caching service
0.3.0
cozystack-apps
Flux
kafka
Managed Kafka service
0.3.0
cozystack-apps
Flux
kubernetes
Managed Kubernetes service
0.9.0
cozystack-apps
Flux
mysql
Managed MariaDB service
0.4.0
cozystack-apps
Flux
nats
Managed NATS service
0.2.0
cozystack-apps
Flux
postgres
Managed PostgreSQL service
0.5.0
cozystack-apps
Flux
rabbitmq
Managed RabbitMQ service
0.3.0
cozystack-apps
Flux
redis
Managed Redis service
0.3.0
cozystack-apps
Flux
tcp-balancer
Layer4 load balancer service
0.2.0
cozystack-apps
Flux
tenant
Separated tenant namespace
1.4.0
cozystack-apps
Flux
virtual-machine
Virtual machine instance
0.3.0
cozystack-apps
Flux
vpn
Managed VPN service
0.3.0
cozystack-apps
Flux

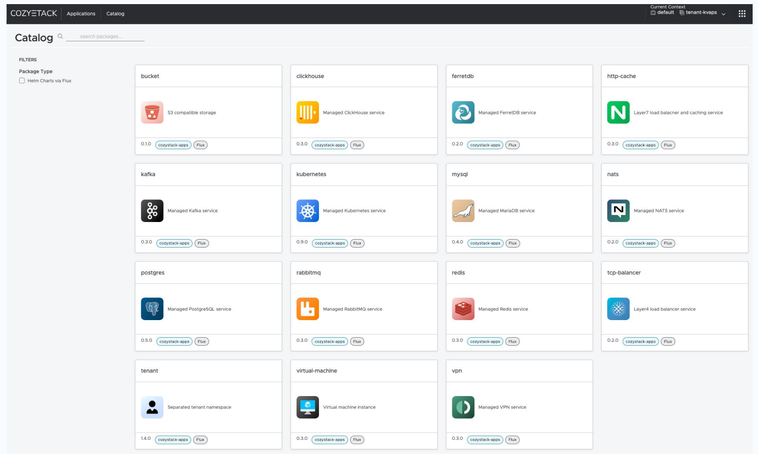

COZYSTACK
Applications
Catalog
Current Context
default
tenant-kvaps
Catalog
search packages...
FILTERS
Package Type
Helm Charts via Flux
bucket
S3 compatible storage
0.1.0
cozystack-apps
Flux
clickhouse
Managed ClickHouse service
0.3.0
ferretdb
Managed FerretDB service
0.2.0
http-cache
Layer7 load balancer and caching service
0.3.0
kafka
Managed Kafka service
0.3.0
kubernetes
Managed Kubernetes service
0.9.0
mysql
Managed MariaDB service
0.4.0
nats
Managed NATS service
0.2.0
postgres
Managed PostgreSQL service
0.5.0
rabbitmq
Managed RabbitMQ service
0.3.0
redis
Managed Redis service
0.3.0
tcp-balancer
Layer4 load balancer service
0.2.0
tenant
Separated tenant namespace
1.4.0
virtual-machine
Virtual machine instance
0.3.0
vpn
Managed VPN service
0.3.0


db


COZYSTACK
Applications
Catalog
Current Context
default
tenant-kvaps
Catalog
search packages...
FILTERS
Package Type
Helm Charts via Flux
bucket
S3 compatible storage
0.1.0
cozystack-apps
Flux
clickhouse
Managed ClickHouse service
0.3.0
cozystack-apps
Flux
ferretdb
Managed FerretDB service
0.2.0
cozystack-apps
Flux
http-cache
Layer7 load balancer and caching service
0.3.0
cozystack-apps
Flux
kafka
Managed Kafka service
0.3.0
cozystack-apps
Flux
kubernetes
Managed Kubernetes service
0.9.0
cozystack-apps
Flux
mysql
Managed MariaDB service
0.4.0
cozystack-apps
Flux
nats
Managed NATS service
0.2.0
cozystack-apps
Flux
postgres
Managed PostgreSQL service
0.5.0
cozystack-apps
Flux
rabbitmq
Managed RabbitMQ service
0.3.0
cozystack-apps
Flux
redis
Managed Redis service
0.3.0
cozystack-apps
Flux
tcp-balancer
Layer4 load balancer service
0.2.0
cozystack-apps
Flux
tenant
Separated tenant namespace
1.4.0
cozystack-apps
Flux
virtual-machine
Virtual machine instance
0.3.0
cozystack-apps
Flux
vpn
Managed VPN service
0.3.0
cozystack-apps
Flux

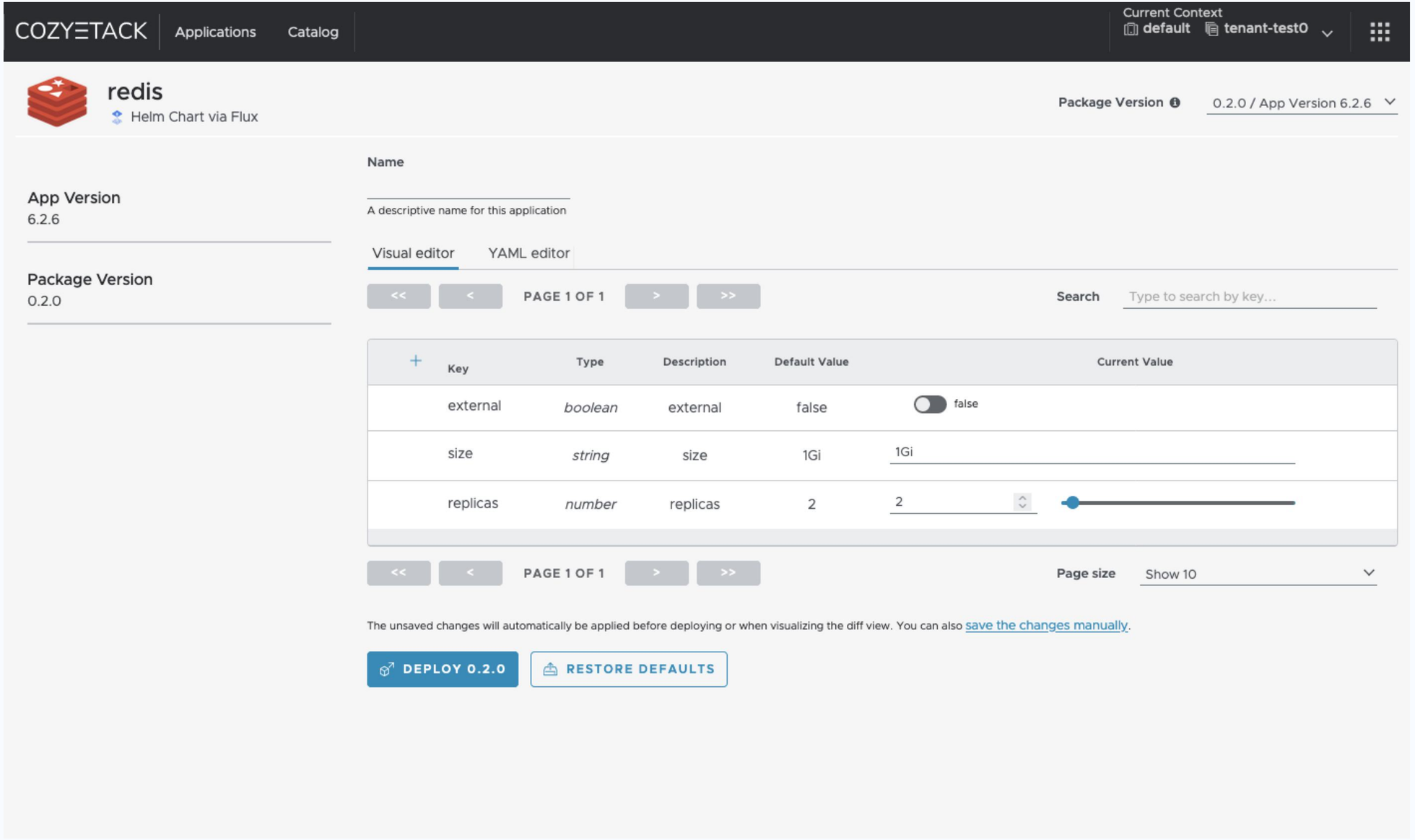

db


HELM


COZYSTACK
Applications
Catalog
Current Context
default
tenant-kvaps
Catalog
search packages...
FILTERS
Package Type
Helm Charts via Flux
bucket
S3 compatible storage
0.1.0
cozystack-apps
Flux
clickhouse
Managed ClickHouse service
0.3.0
ferretdb
Managed FerretDB service
0.2.0
http-cache
Layer7 load balancer and caching service
0.3.0
kafka
Managed Kafka service
0.3.0
kubernetes
Managed Kubernetes service
0.9.0
mysql
Managed MariaDB service
0.4.0
nats
Managed NATS service
0.2.0
postgres
Managed PostgreSQL service
0.5.0
rabbitmq
Managed RabbitMQ service
0.3.0
redis
Managed Redis service
0.3.0
tcp-balancer
Layer4 load balancer service
0.2.0
tenant
Separated tenant namespace
1.4.0
virtual-machine
Virtual machine instance
0.3.0
vpn
Managed VPN service
0.3.0


db


HELM


COZYSTACK
Applications
Catalog
Current Context
default
tenant-test0
redis
Helm Chart via Flux
Package Version
0.2.0 / App Version 6.2.6
App Version
6.2.6
Package Version
0.2.0
Name
A descriptive name for this application
Visual editor
YAML editor
<<
<
PAGE 1 OF 1
>
>>
Search
Type to search by key...
Key
Type
Description
Default Value
Current Value
external
boolean
external
false
false
size
string
size
1Gi
1Gi
replicas
number
replicas
2
2
Page size
Show 10
The unsaved changes will automatically be applied before deploying or when visualizing the diff view. You can also save the changes manually.
DEPLOY 0.2.0
RESTORE DEFAULTS

Redis app


```
external: false
size: 1Gi
replicas: 2
```

Redis app


```
external: false
size: 1Gi
replicas: 2
```

RedisFailover

Service

VMServiceScrape

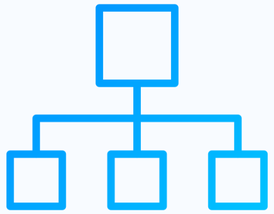
COZYSTACK

Установка в одно нажатие

Managed Kubernetes

Managed Databases

Virtual Machines



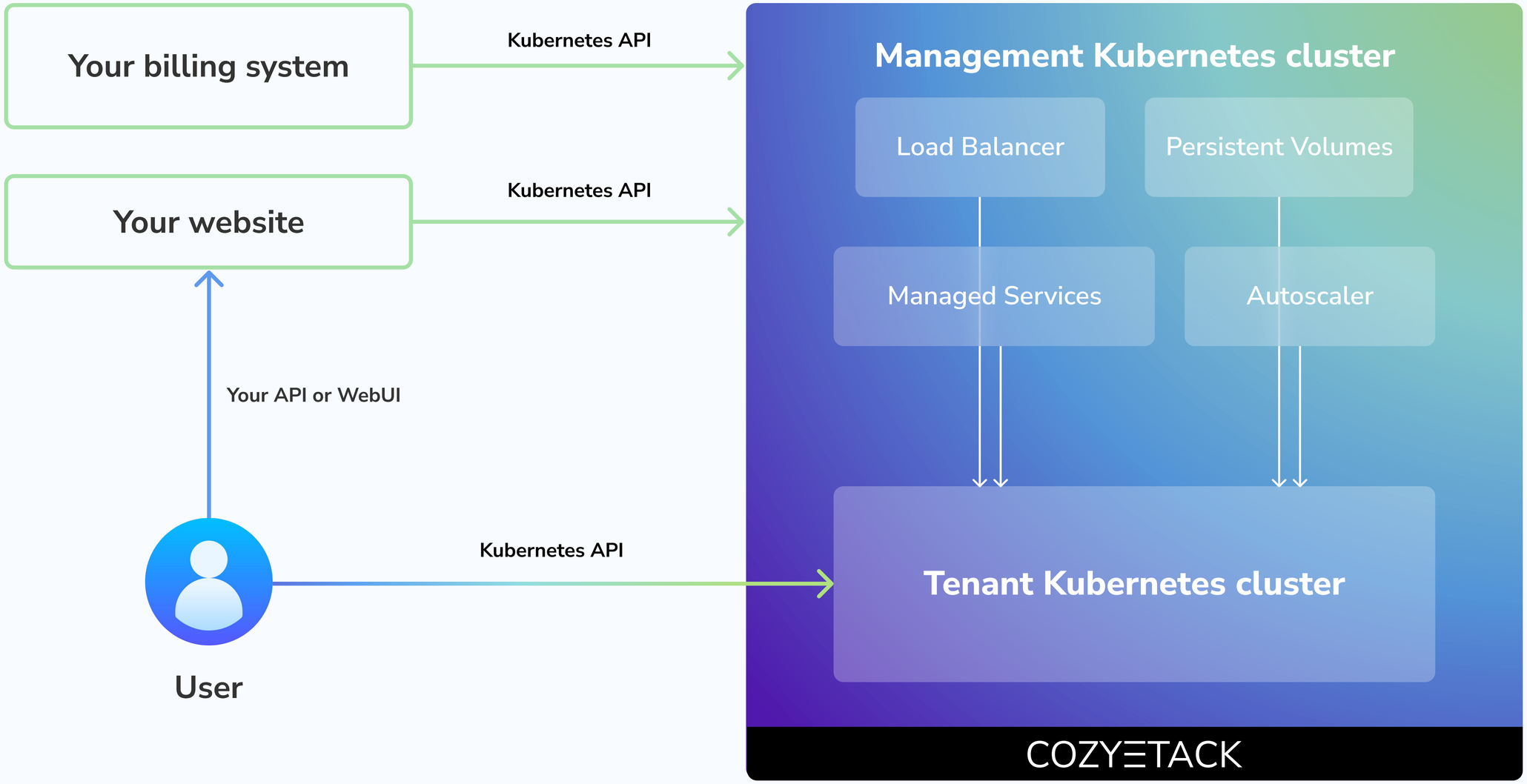
Load balancers

Managed VPN service

HTTP-caching service

S3 Storage

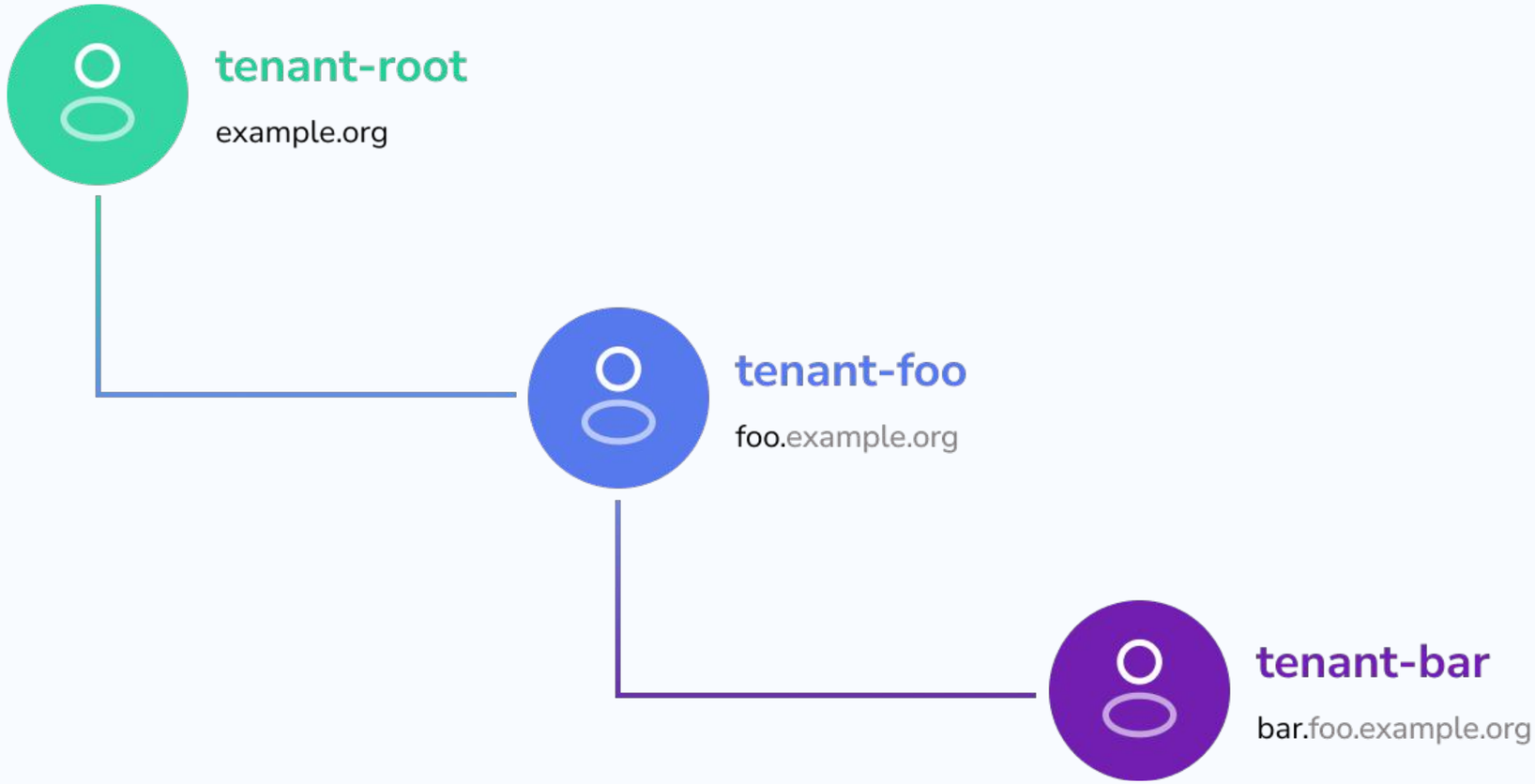
Что насчет мульти-тенантности?


Your billing system
Kubernetes API
Your website
Kubernetes API
Your API or WebUI
User
Kubernetes API
Management Kubernetes cluster
Load Balancer
Persistent Volumes
Managed Services
Autoscaler
Tenant Kubernetes cluster
COZYSTACK

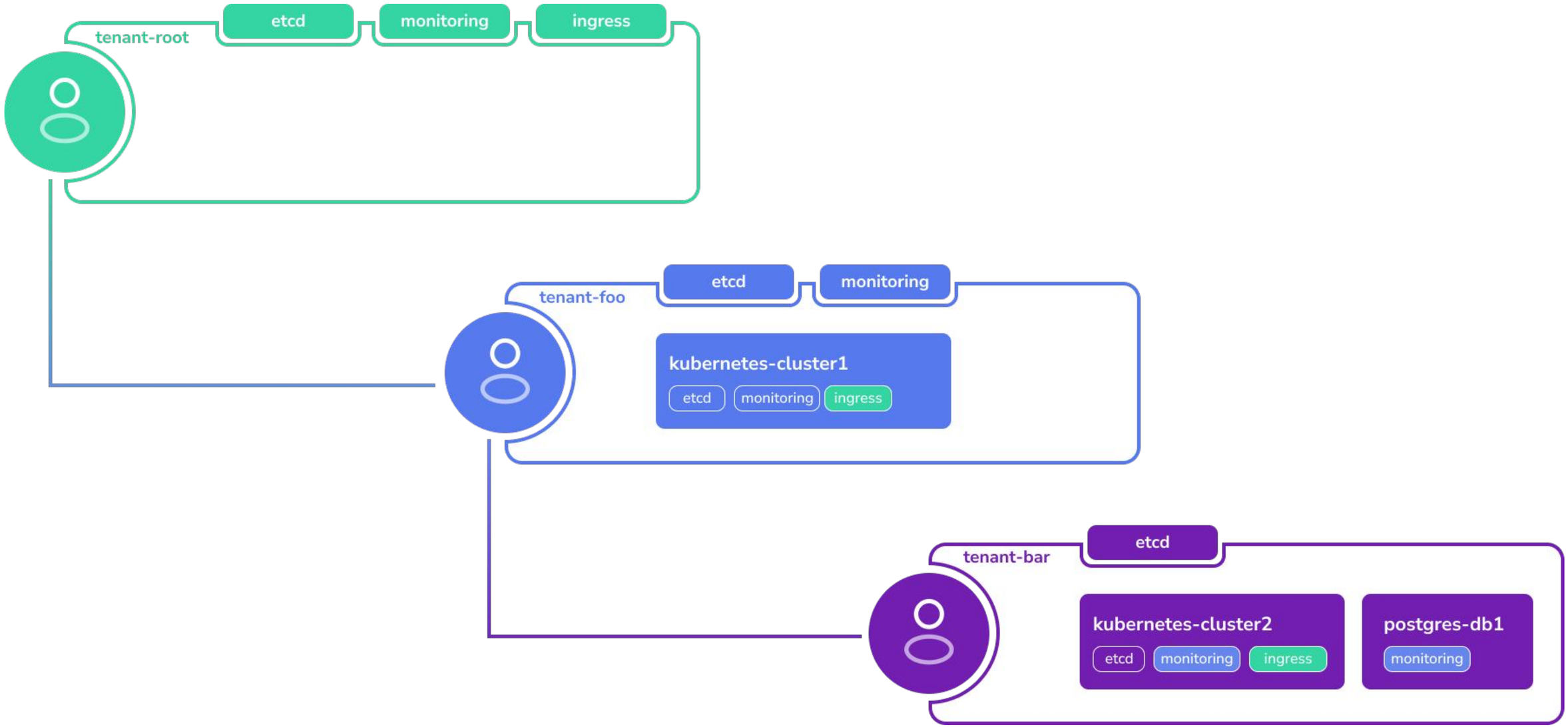
COZYSTACK

Экономно и безопасно

Наша уникальная модель тенантов обеспечивает эффективное выделение ресурсов и необходимый уровень безопасности.



COZYSTACK


tenant-root
etcd
monitoring
ingress
tenant-foo
etcd
monitoring
kubernetes-cluster1
etcd
monitoring
ingress
tenant-bar
etcd
kubernetes-cluster2
etcd
monitoring
ingress
postgres-db1
monitoring

Выводы

Все окружения единообразны и повторяемы

Выводы

Все окружения единообразны и повторяемы

Простой интерфейс установки

Выводы

Все окружения единообразны и повторяемы

Простой интерфейс установки

Платформа закрывает весь стек для построения облака

Выводы

Все окружения единообразны и повторяемы

Простой интерфейс установки

Платформа закрывает весь стек для построения облака

Легко расширяемая

Выводы

Все окружения единообразны и повторяемы

Простой интерфейс установки

Платформа закрывает весь стек для построения облака

Легко расширяемая

Открытый код!

Ссылки

Ссылки

Cozystack

See more

Ссылки

Cozystack

See more

Talos Linux

See more

Ссылки

Cozystack

See more

Talos Linux

See more

Kamaji

See more

Ссылки

Cozystack

See more

Talos Linux

See more

Kamaji

See more

DIY: Create Your Own Cloud with Kubernetes

See more

Благодарности

Dario Tranchitella

Timur Tukaev

Viktoriia Kvapil

Andrei Kvapil

Спасибо!
Вопросы?

Георг Гаал

CTO & основатель of Ænix