

МЕГА  
СЛЕРМ

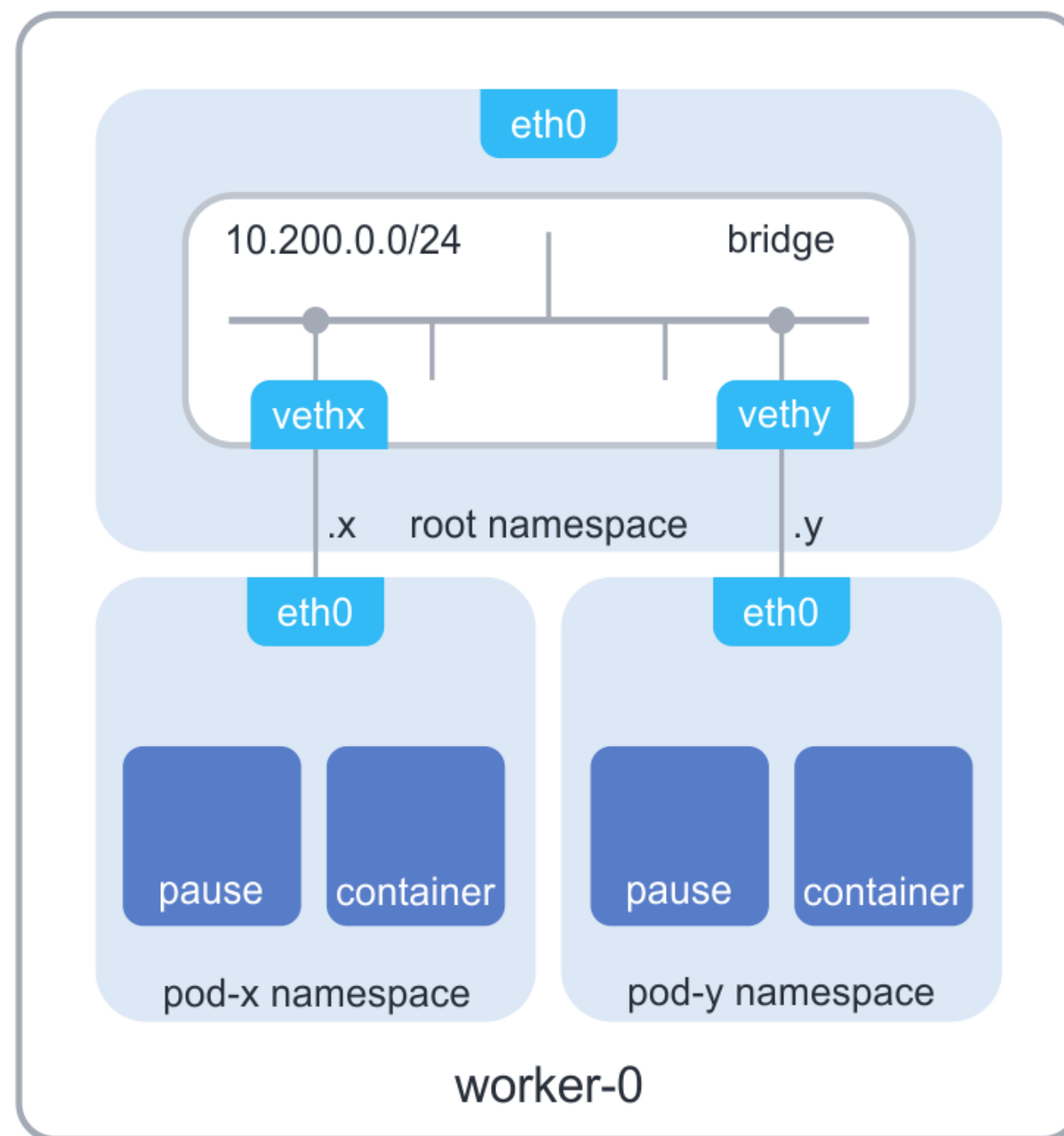
+



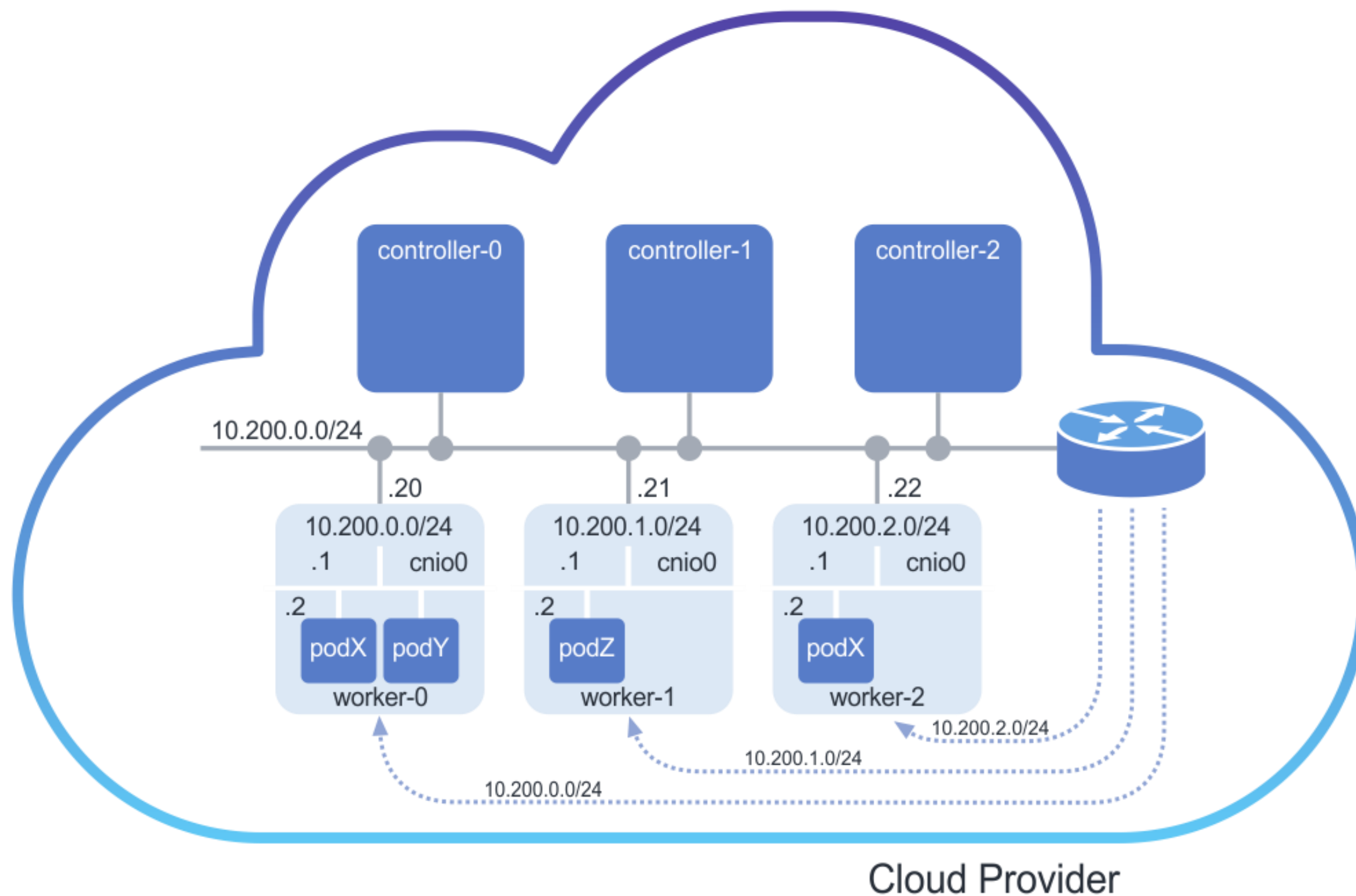
Southbridge

# Kubernetes Networking

# Структура сети на узле



# Структура сети кластера



# Container Network Interface

Исполняемая среда контейнера  
(rkt, Kubernetes и т. д.)

Container Network  
Interface (CNI)



плагины CNI

Loopback

IPvlan

etc.

# Container Network Interface

Опции kubelet:

```
--network-plugin=kubenet  
--network-plugin=cni
```

# Container Network Interface

Опции kubelet:

```
--cni-bin-dir=/opt/cni/bin  
--cni-conf-dir=/etc/cni/net.d
```



# Популярные CNI плагины

Provider	Network Model	Network Policies	Encryption
Calico	BGP layer2, IP, vxlan	Yes	No
Canal (flannel+calico)	layer2, IP, vxlan	Yes	No
flannel	layer2, IP, vxlan ipsec	No	Partial
Weave Net	vxlan	Yes	Yes

# Популярные CNI плагины

Provider	Network Model	Network Policies	Encryption
Calico	BGP layer2, IPIP, vxlan	Yes	No
Canal (flannel+calico)	layer2, IPIP, vxlan	Yes	No
flannel	layer2, IPIP, vxlan ipsec	No	Partial
Weave Net	vxlan	Yes	Yes



# Популярные CNI плагины

Provider	Network Model	Network Policies	Encryption
Calico	BGP layer2, IP, vxlan	Yes	No
Canal (flannel+calico)	layer2, IP, vxlan	Yes	No
flannel	layer2, IP, vxlan ipsec	No	Partial
Weave Net	vxlan	Yes	Yes

# Популярные CNI плагины

Provider	Network Model	Network Policies	Encryption
Calico	BGP layer2, IPIP, vxlan	Yes	No
Canal (flannel+calico)	layer2, IPIP, vxlan	Yes	No
flannel	layer2, IPIP, vxlan ipsec	No	Partial
Weave Net	vxlan	Yes	Yes

# Популярные CNI плагины

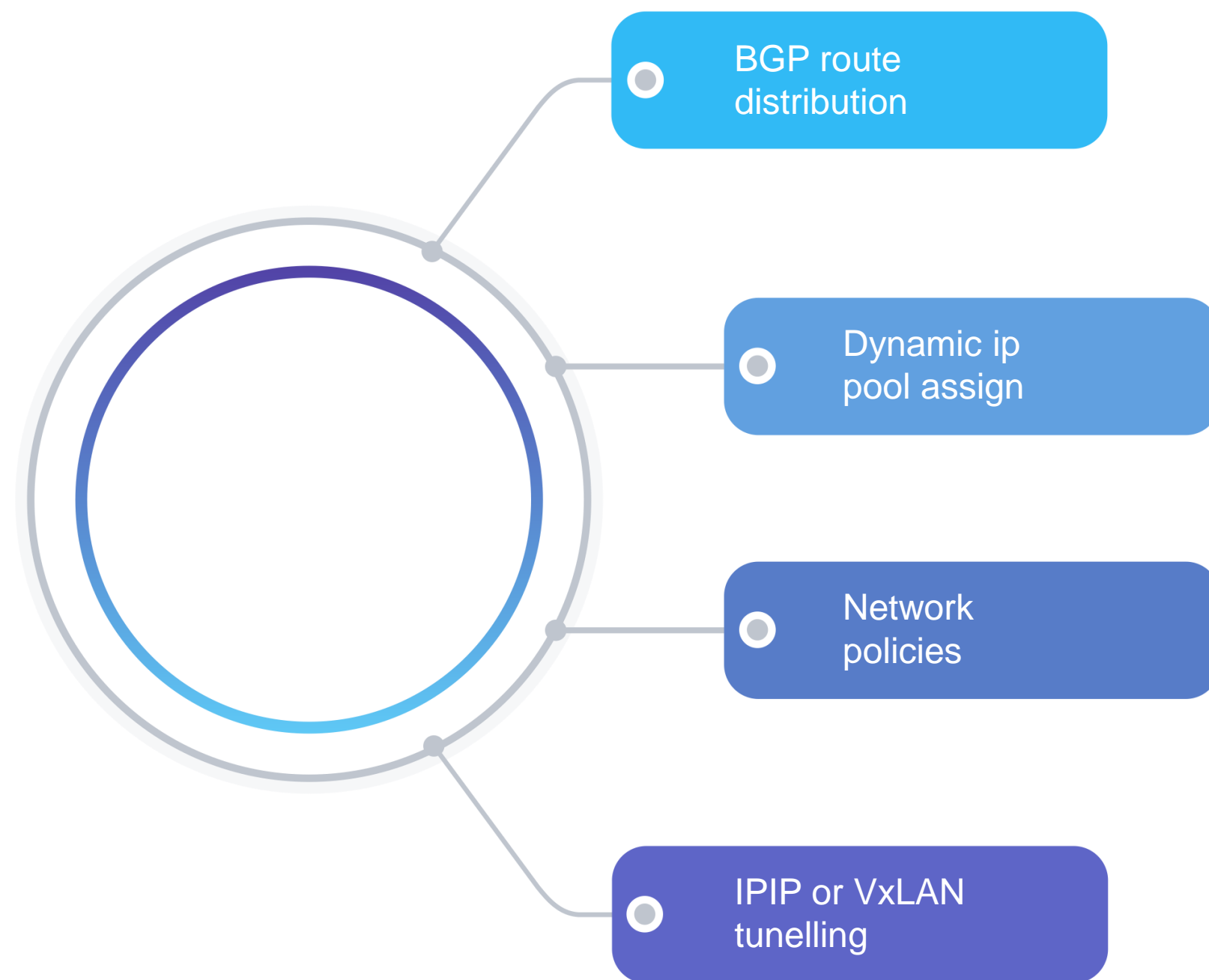
<https://kubernetes.io/docs/concepts/cluster-administration/networking/#how-to-implement-the-kubernetes-networking-model>

Provider	Network Model	Network Policies	Encryption
Calico	BGP layer2, IPIP, vxlan	Yes	No
Canal (flannel+calico)	layer2, IPIP, vxlan	Yes	No
flannel	layer2, IPIP, vxlan ipsec	No	Partial
Weave Net	vxlan	Yes	Yes

# Flannel backend



# Calico feature



PROJECT  
**CALICO**

# Calico IPPool

```
# calicoctl get ippool -o yaml
apiVersion: projectcalico.org/v3
kind: IPPool
metadata:
  name: default-pool
spec:
  cidr: 10.122.0.0/16
  ipipMode: CrossSubnet or Always or Never
  natOutgoing: true
```

```
# calicoctl get nodes -o wide
```

NAME	ASN	IPV4
master-1.mega.slurm.io	(64512)	172.21.1.2/24
node-1.mega.slurm.io	(64512)	172.21.1.6/24
node-2.mega.slurm.io	(64512)	172.21.1.7/32

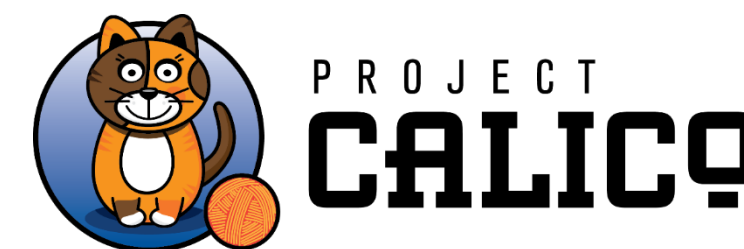


PROJECT  
**CALICO**

# Calico IPPool

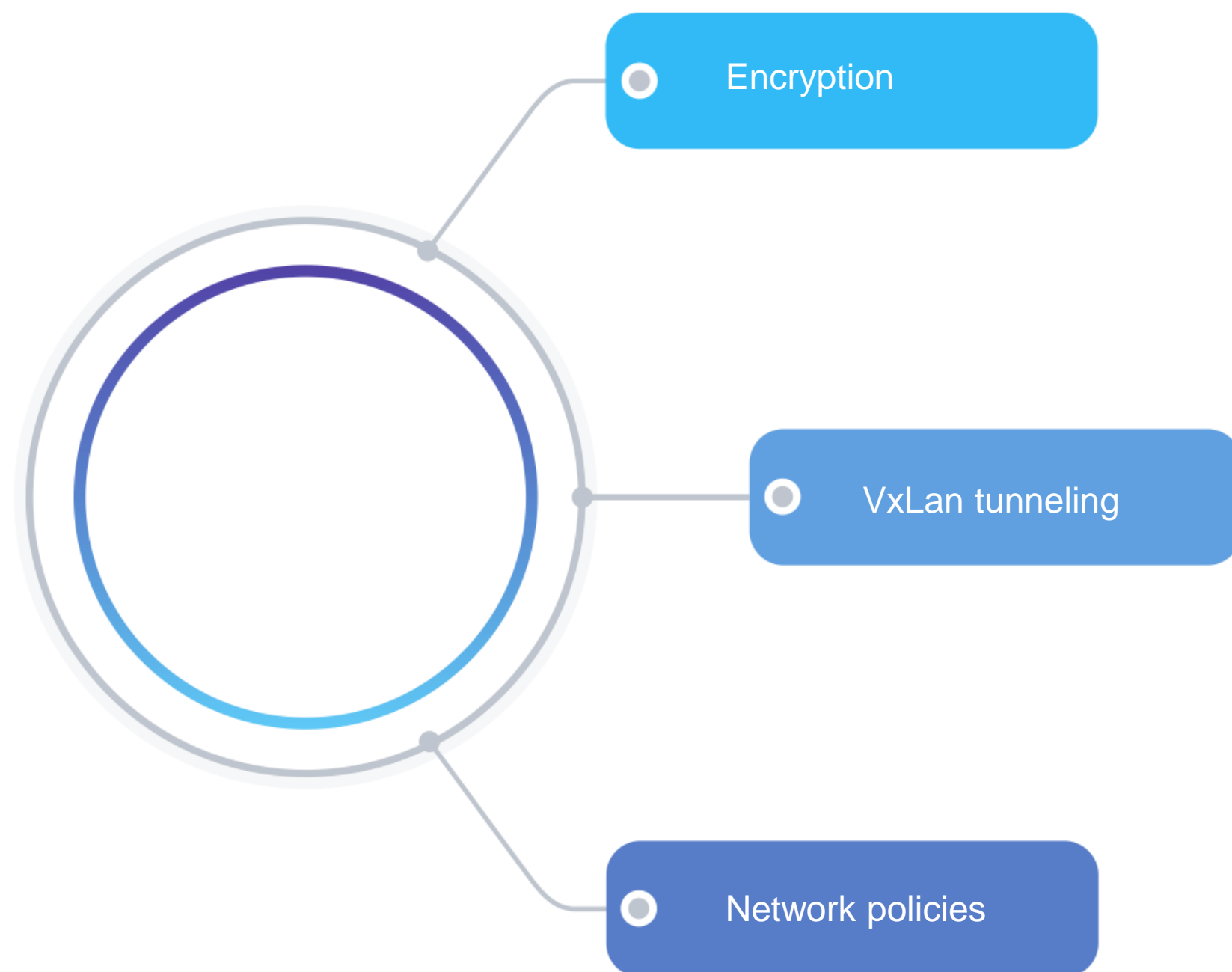
```
# kubectl get ds calico-node -o yaml

- name: IP_AUTODETECTION_METHOD
  value: interface=<int-regex> or
        can-reach=<dest> or
        first-found
- name: IP
  value: autodetect or none or <ip.add.re.ss>
```





# Weave Net feature



# Network Policy

Запрещено всё, что не разрешено



# Network Policy

Для примера:

Установка calico network policy controller

<https://docs.projectcalico.org/v3.10/manifests/calico-policy-only.yaml>

КАЧАЕМ

Изменено:

- выключен сервис typha (кеш обращений к etcd)
- установлен CALICO\_IPV4POOL\_CIDR = 10.244.0.0/16

kubectl apply -f calico-policy-only.yaml

ИЗМЕНЯЕМ

ПРИМЕНЯЕМ

# Network Policy

Запрещает все



```
kind: NetworkPolicy
apiVersion:
networking.k8s.io/v1
metadata:
  name: default-deny
  namespace: base
spec:
  podSelector:
    matchLabels: {}
```

# Network Policy

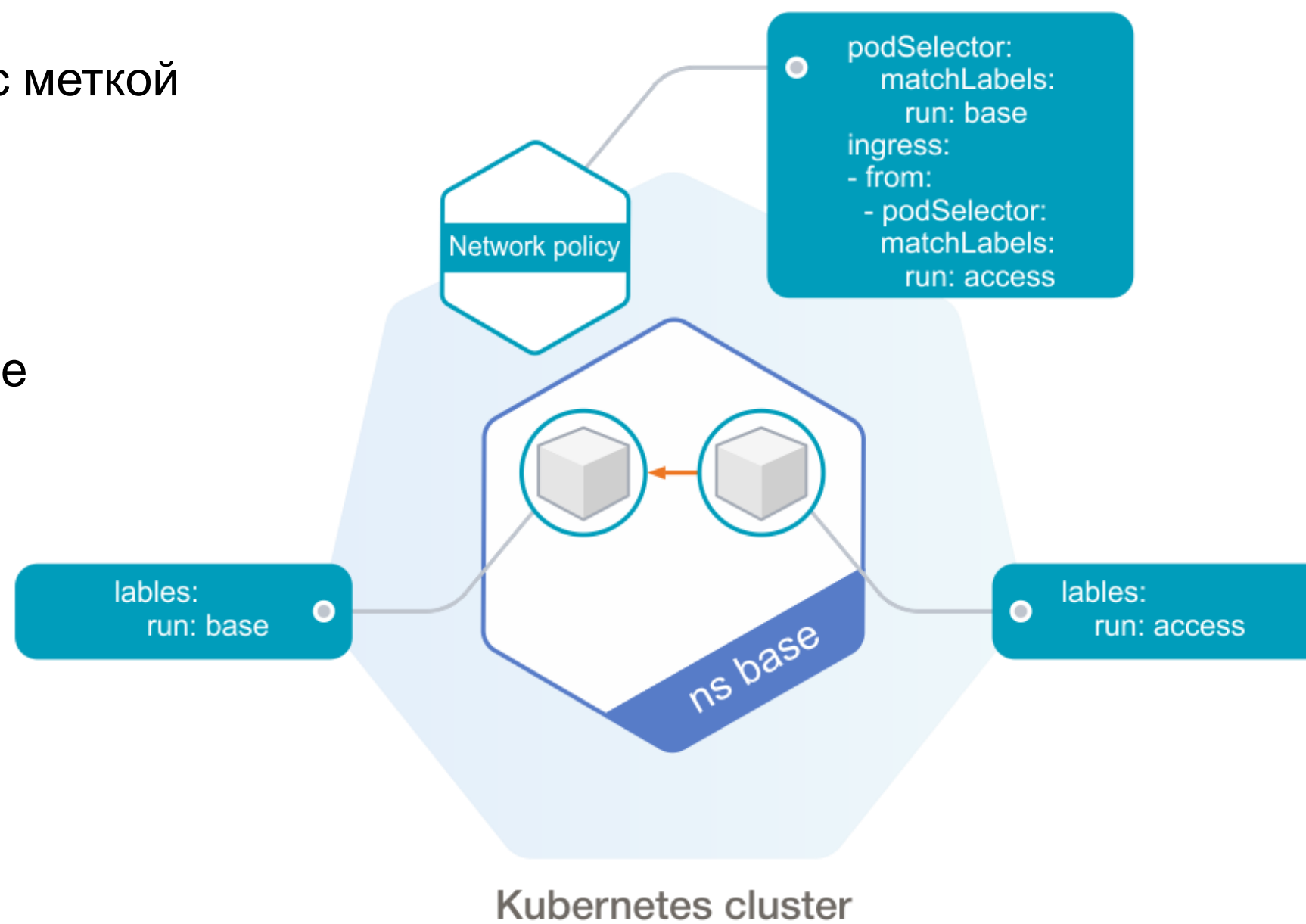
Разрешаем доступ к подам с меткой

`run=base`

с подов с меткой

`run=access`

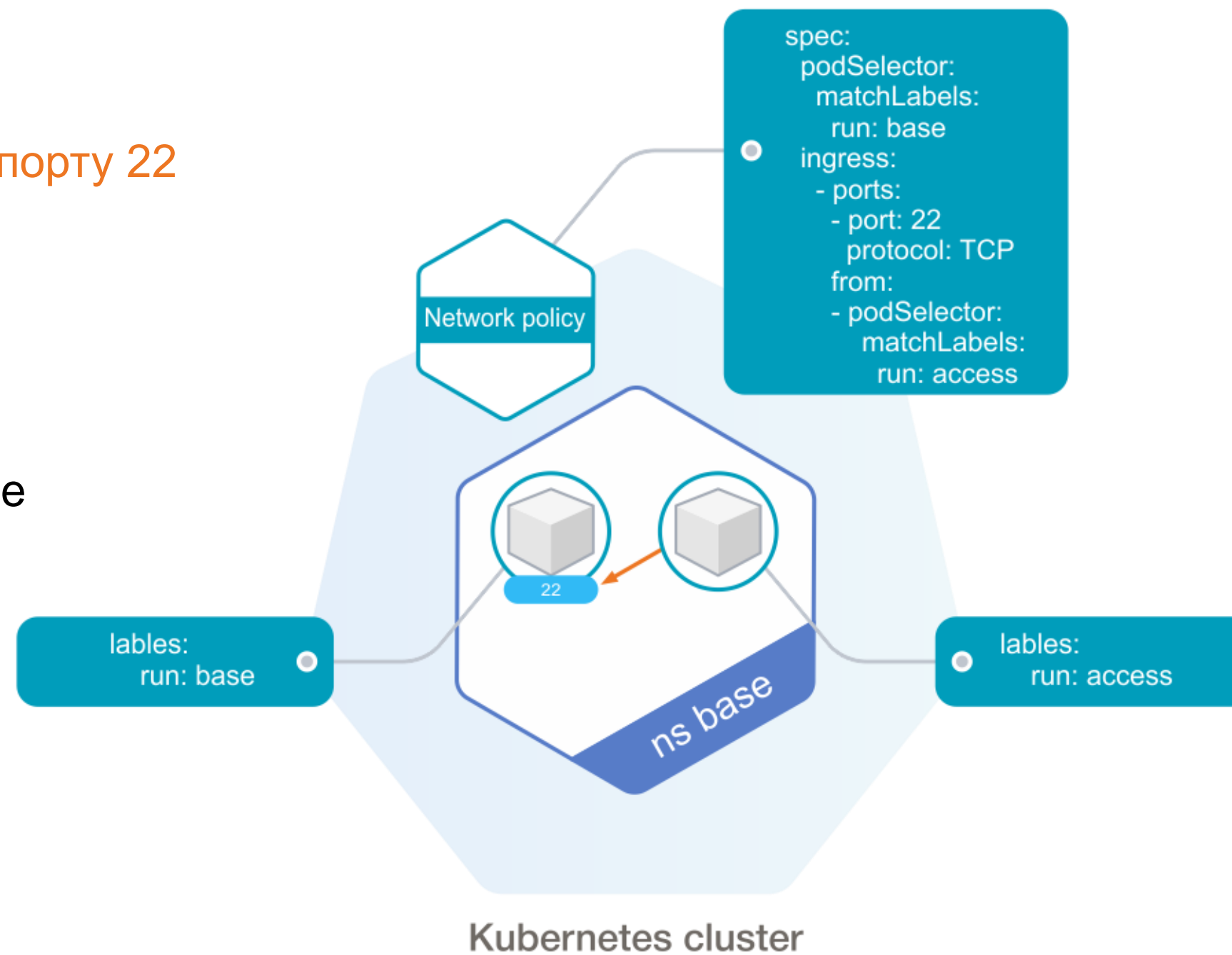
Все поды в одном namespace



# Network Policy

Разрешаем доступ только к **порту 22**  
подов с меткой  
**run=base**  
с подов с меткой  
**run=access**

Все поды в одном namespace



# Network Policy

Разрешаем доступ только к порту 22

подов с меткой

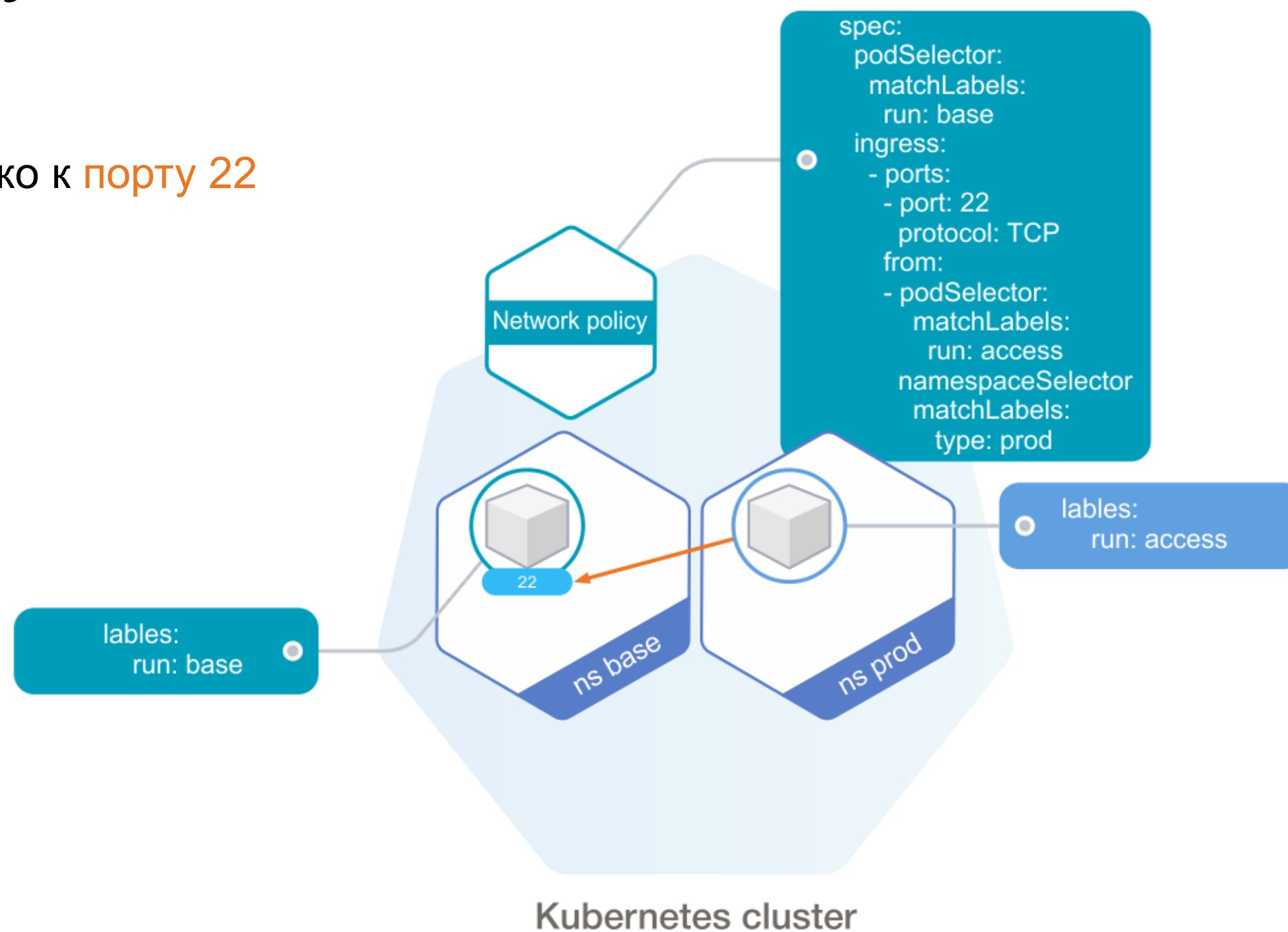
run=base

с подов с меткой

run=access

из namespaces с меткой

type=prod





# Network Policy

ИЛИ

```
- from:
  - podSelector:
      matchLabels:
        run: access
  - namespaceSelector:
      matchLabels:
        type: prod
```

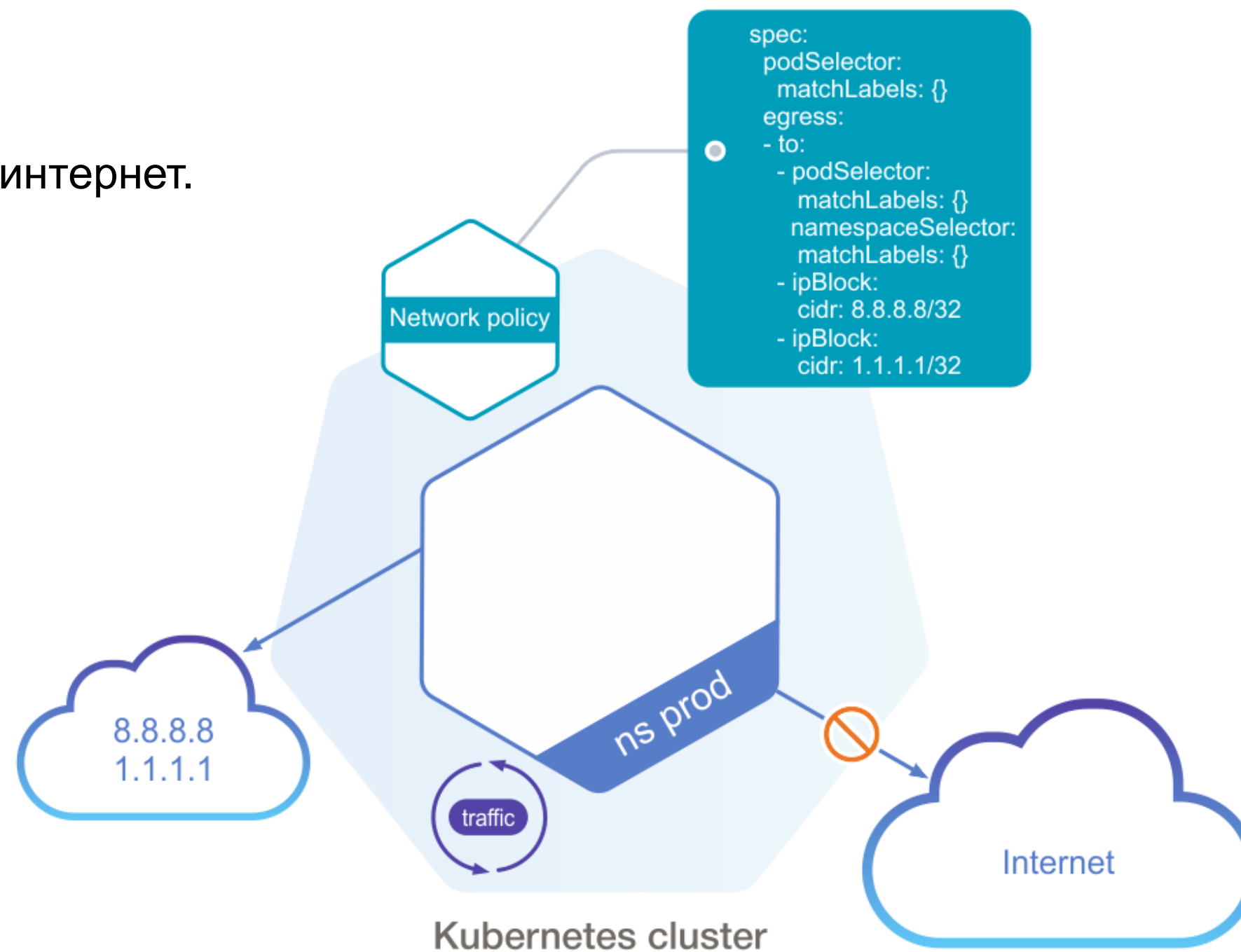
И

```
- from:
  - podSelector:
      matchLabels:
        run: access
    namespaceSelector:
      matchLabels:
        type: prod
```

# Network Policy

Запрещаем доступ с прода в интернет.  
Разрешаем:

- только внутри кластера
- 8.8.8.8
- 1.1.1.1



# Network Policy

Запрещаем доступ с прода в интернет.  
Разрешаем:

- только внутри кластера
- 8.8.8.8
- 1.1.1.1

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: deny-prod
  namespace: prod
spec:
  podSelector:
    matchLabels: {}
  egress:
  - to:
    - podSelector:
        matchLabels: {}
      namespaceSelector:
        matchLabels: {}
    - ipBlock:
        cidr: 8.8.8.8/32
    - ipBlock:
        cidr: 1.1.1.1/32
```

policyTypes:  
- Egress

# Network Policy

Запрещаем доступ с прода в i

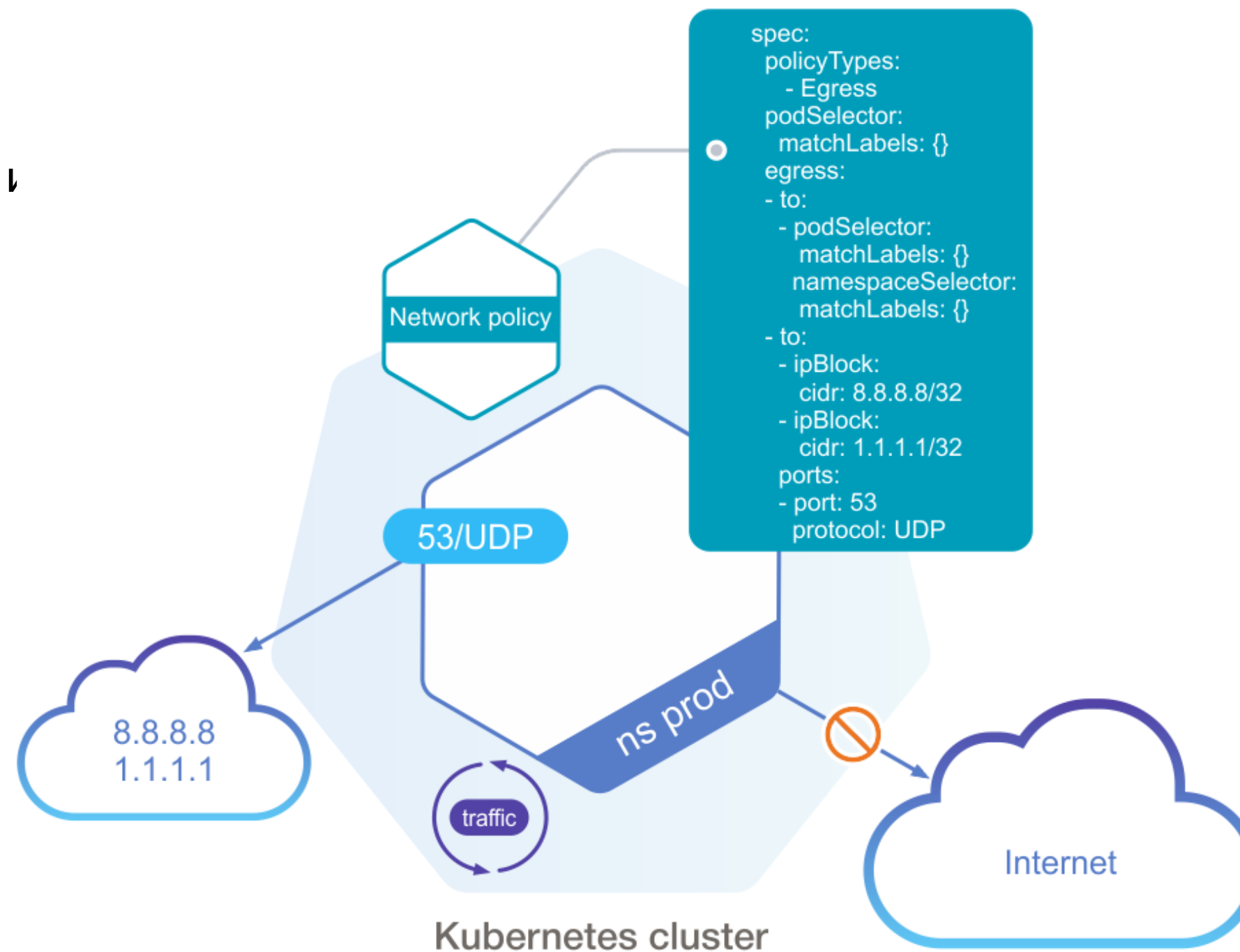
Разрешаем весь трафик:

- только внутри кластера

Разрешаем только 53/UDP:

- 8.8.8.8

- 1.1.1.1



# Network Policy

```
spec:
  policyTypes:
    - Ingress
    - Egress
    - Ingress,Egress
  ingress:
    - ports:
        - port: 53
          protocol: UDP
    - from:
  egress:
    - ports:
        to:
```

```
- from: или - to:
  ipBlock:
    cidr:
    except:
  namespaceSelector:
    matchLabels: {}
  podSelector:
    matchLabels:
      run: access
```

# Network Policy a lot of iptables

```
-A -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A -m conntrack --ctstate INVALID -j DROP
-A -j MARK --set-xmark 0x0/0x10000
-A --comment "Start of policies" -j MARK --set-xmark 0x0/0x20000
-A -m mark --mark 0x0/0x20000 -j cali-pi-SPiRZx_J8ehLJv19gp3
-A --comment "Return if policy accepted" -m mark --mark 0x10000/0x10000 -j RETURN
-A --comment "Drop if no policies passed packet" -m mark --mark 0x0/0x20000 -j DROP
-A -j cali-pri-kns.base
-A --comment "Return if profile accepted" -m mark --mark 0x10000/0x10000 -j RETURN
-A -j cali-pri-ksa.base.default
-A --comment "Return if profile accepted" -m mark --mark 0x10000/0x10000 -j RETURN
-A --comment "Drop if no profiles matched" -j DROP

-A cali-pi-SPiRZx_J8ehLJv19gp3 -p tcp -m set --match-set cali40s:WUSJuJOIzV8Yck9U8UDVSeH src -m multiport --dports
22 -j MARK --set-xmark 0x10000/0x10000

ipset save
create cali40s:WUSJuJOIzV8Yck9U8UDVSeH hash:net family inet hashsize 1024 maxelem 1048576
add cali40s:WUSJuJOIzV8Yck9U8UDVSeH 10.0.5.4
```

# Network Policy

Если под **попадает** под выборку Network Policy - то будет **запрещено все, кроме разрешенного**

Если под **не попадает** под Network Policy - то будет **разрешено все**

Если в NetworkPolicy **нет** namespaceSelector - то правило разрешает доступ только из подов в **том же** namespace

Если в NetworkPolicy **есть** namespaceSelector - то правило разрешает доступ только из подов в namespace, **попадающих под выборку**



МЕГА  
СЛЕРМ

+



Southbridge



slurm.io