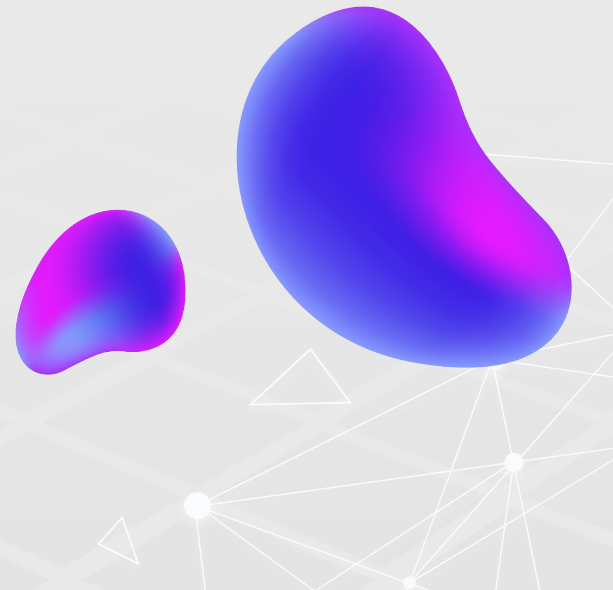


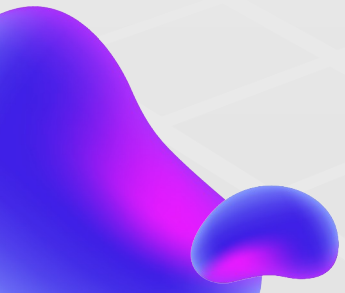
Введение в Kubernetes День 3

Летняя школа ВШЭ 2024.

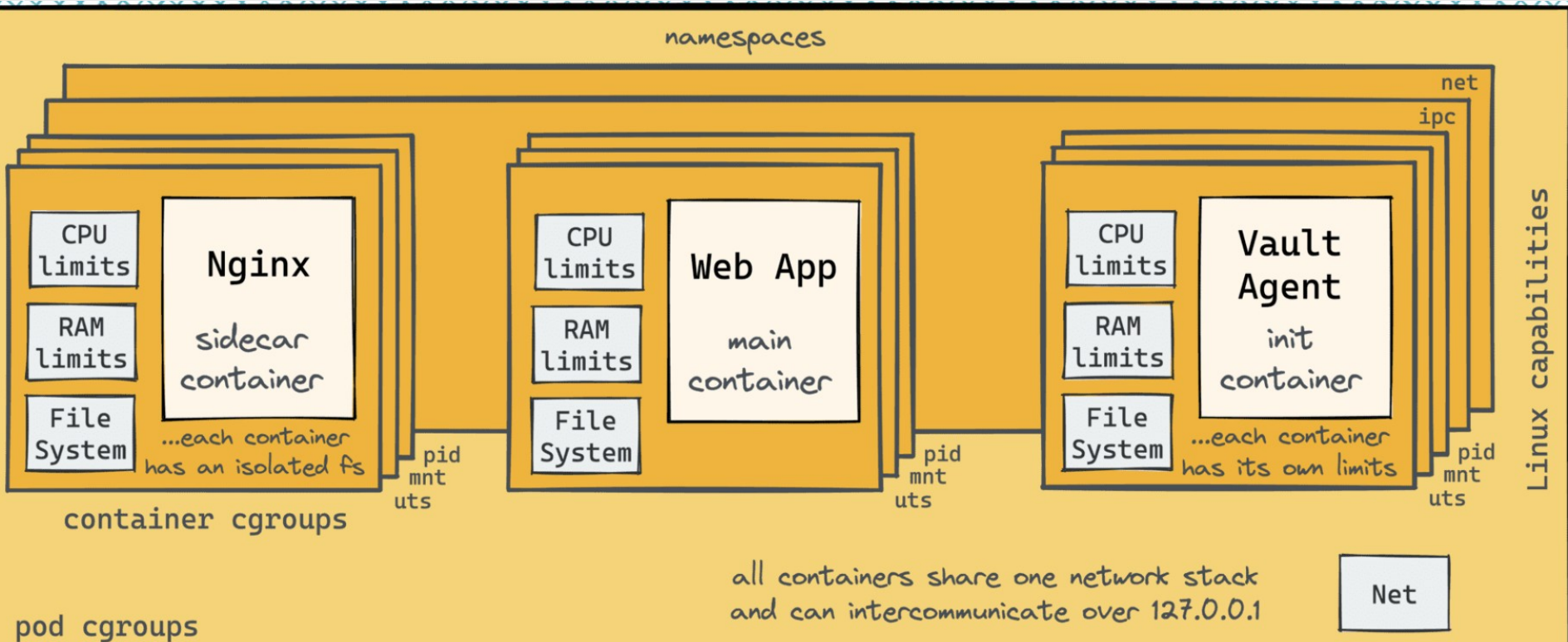




В предыдущей лекции

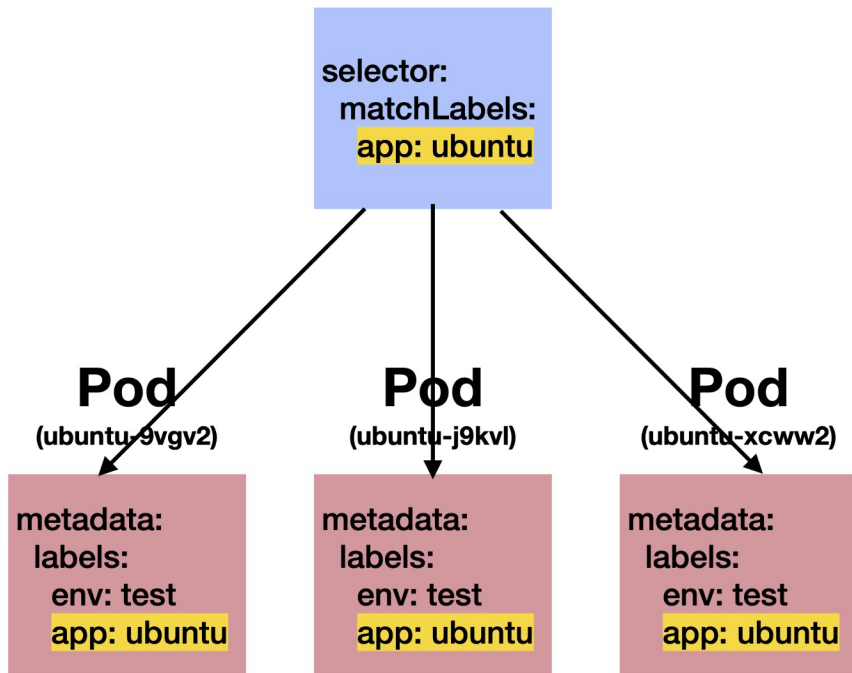
- При создании пода сначала запускается pause контейнер, для которого создаются неймспейсы: net, mnt, uts, ipc, pid
 - Затем контейнеры запускаются внутри пода и получают только 3 неймспейса: mnt, pid, cgroup
 - Оставшиеся 3 неймспейса (net, uts, ipc) контейнеры делят между собой.
- 

В предыдущей лекции



В предыдущей лекции

ReplicaSet



В предыдущей лекции

Deployment

Updates and Rollback

ReplicaSet

Self-healing, scalable, desired state

Pod

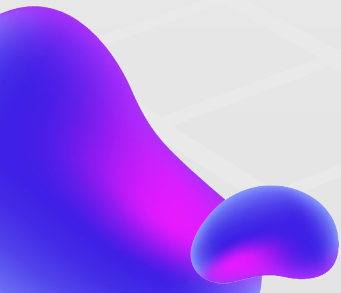


Pod

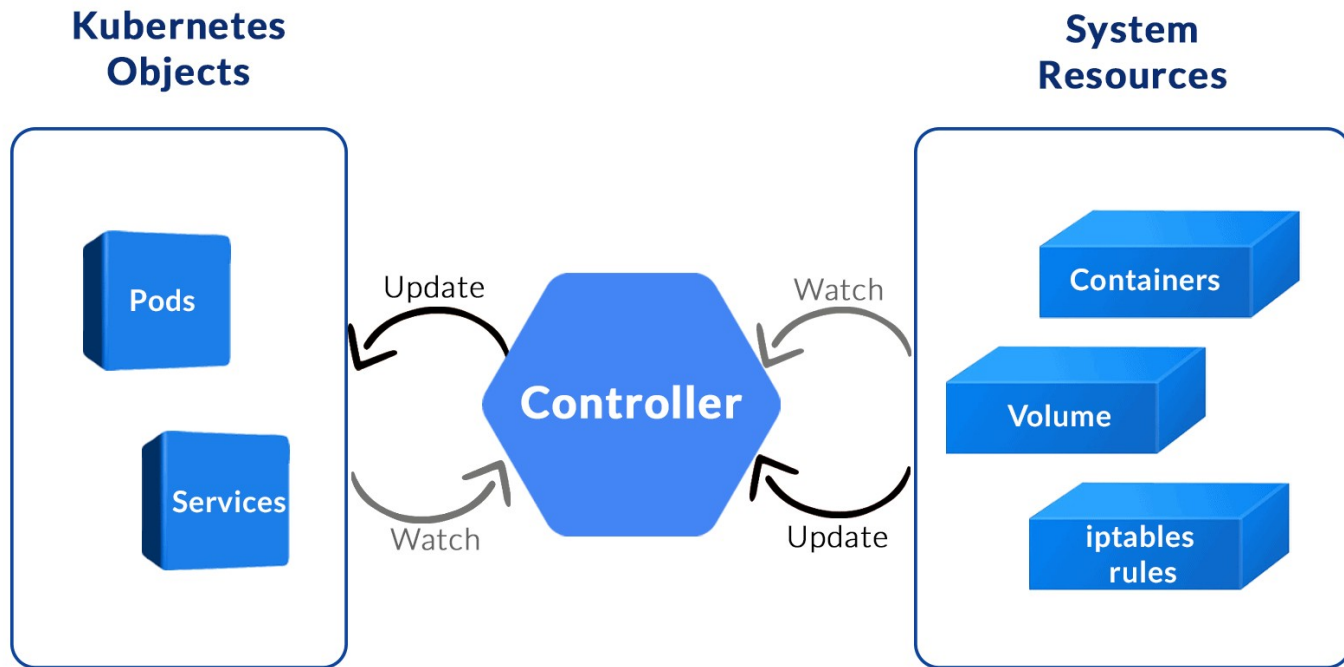


...

Pod



В предыдущей лекции





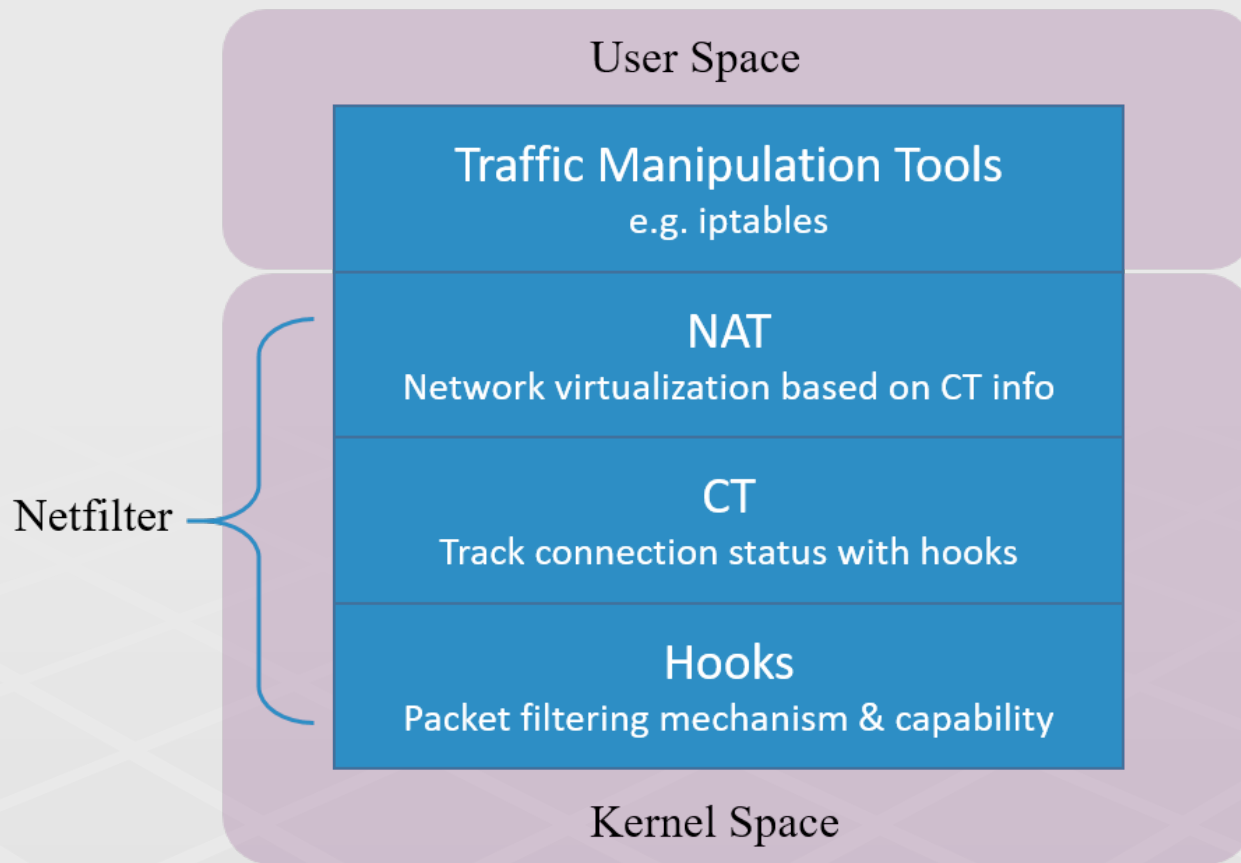
01

Сеть в ОС Linux

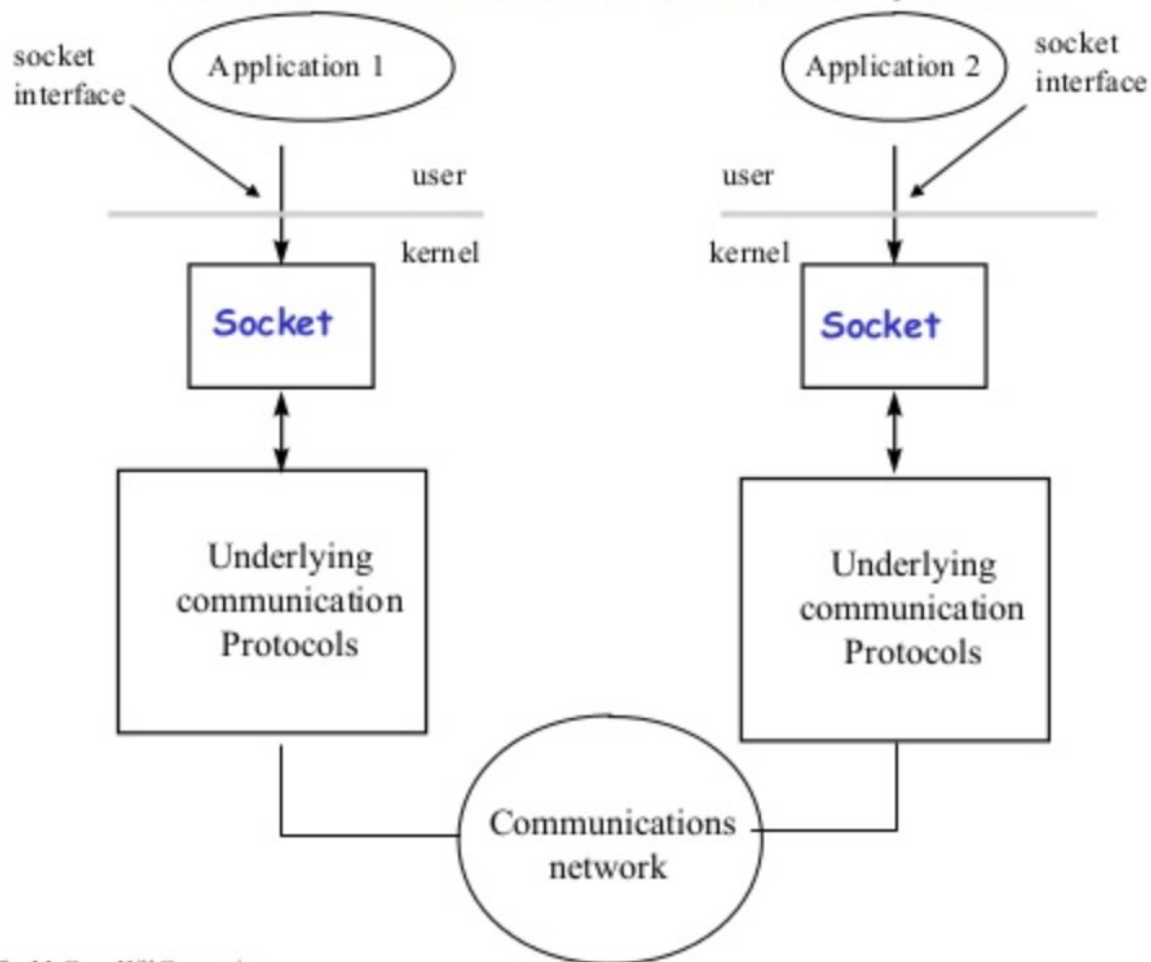
За что отвечает ядро Linux?

За взаимодействие между пакетами и согласованный поток данных для программ. Маршрутизация и брандмауэры - ключевые функции k8s базируются на процедурах пакетной обработки Linux.

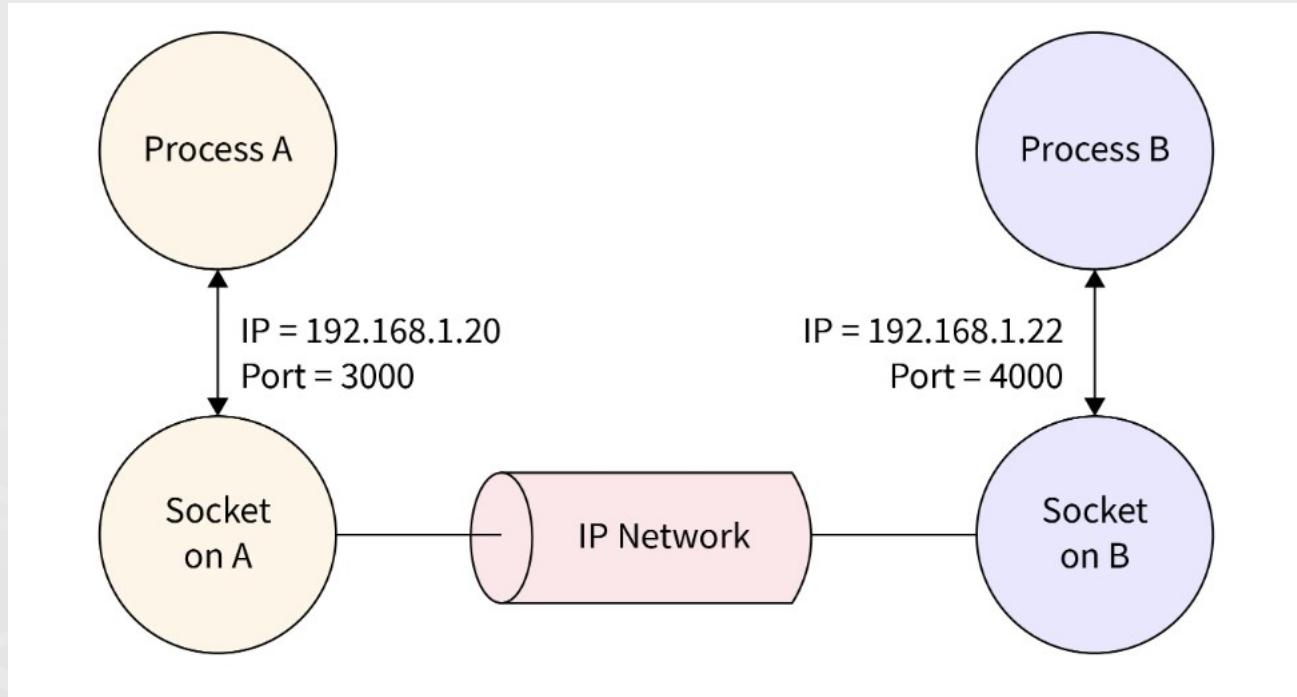
Linux Node



The Socket Interface



Ключевая абстракция пользовательского пространства - socket



Типы сокетов Linux

Datagram-Oriented Sockets (UDP)

Stream-Oriented Sockets (TCP)

Упорядоченная передача данных с проверкой на ошибки. Используется Веб-серверы, почтовые серверы, базы данных.

Ненадёжная, неупорядоченная передача данных, но высокая скорость. Используется для аудио/видео стриминга, онлайн-игры

Raw Sockets

Прямой доступ к сетевым протоколам, минуя TCP/UDP, для низкоуровневого анализа. Используется для Сетевые мониторы, фаерволы, анализаторы трафика

Sequenced Sockets (SCTP)

Надёжная, упорядоченная, мультистримовая передача данных, используется для телефонии, реальное время стриминг

Клиент:

Создаёт сокет (socket).

Привязывает сокет к порту (bind).

Иницирует соединение с сервером (connect).

Обменивается данными с сервером (send, recv).

Закрывает соединение (close).

Сервер:

Создаёт сокет (socket).

Привязывает сокет к порту (bind).

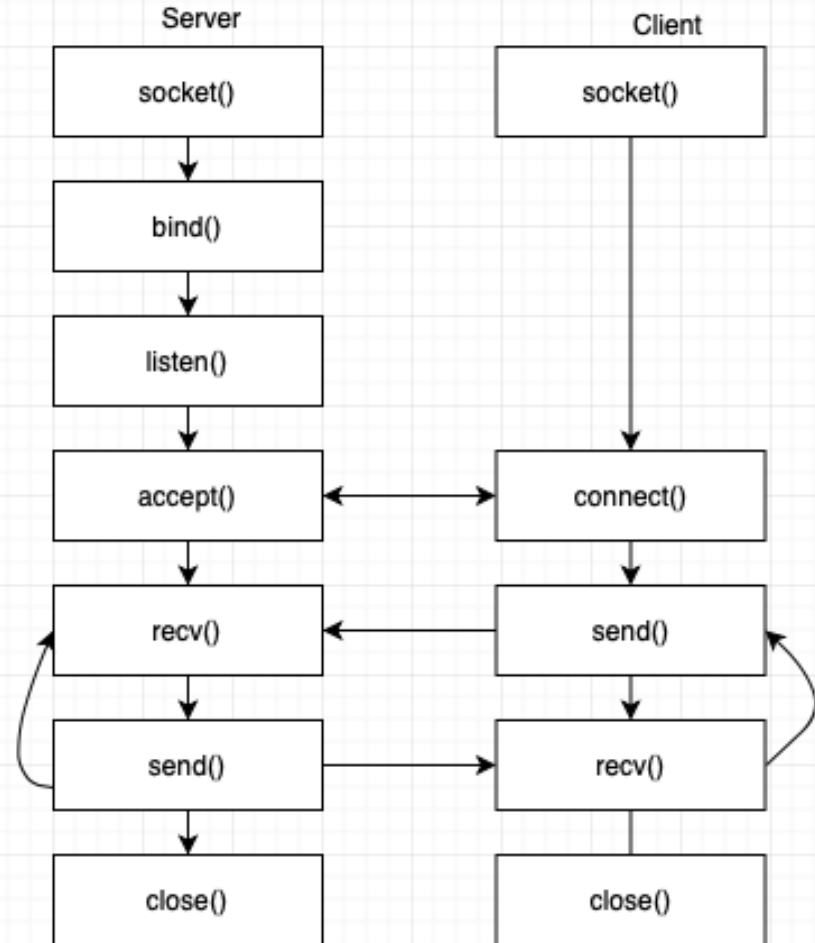
Начинает прослушивание на порту (listen).

Принимает соединение от клиента (accept).

Обменивается данными с клиентом (recv, send).

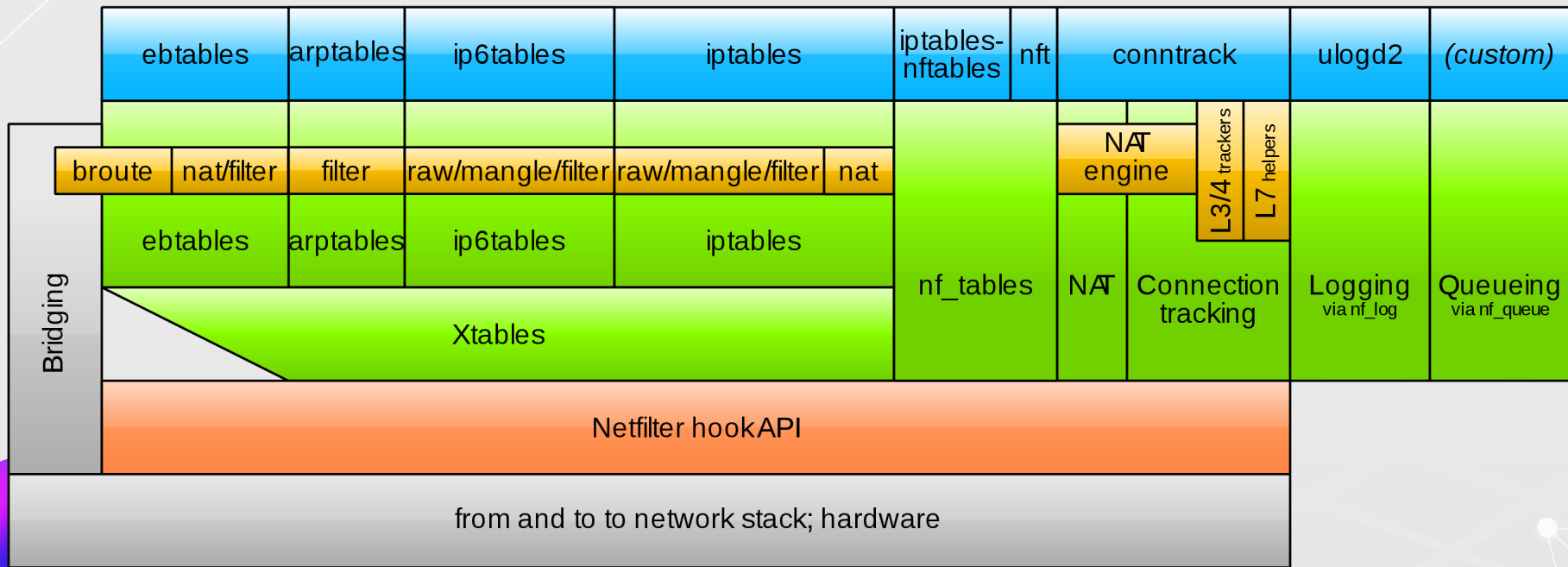
Закрывает соединение (close).

Stream-TCP socket Creation



Netfilter components

Jan Engelhardt, last updated 2014-02-28 (initial: 2008-06-17)



Userspace tools

Netfilter kernel components

other networking components

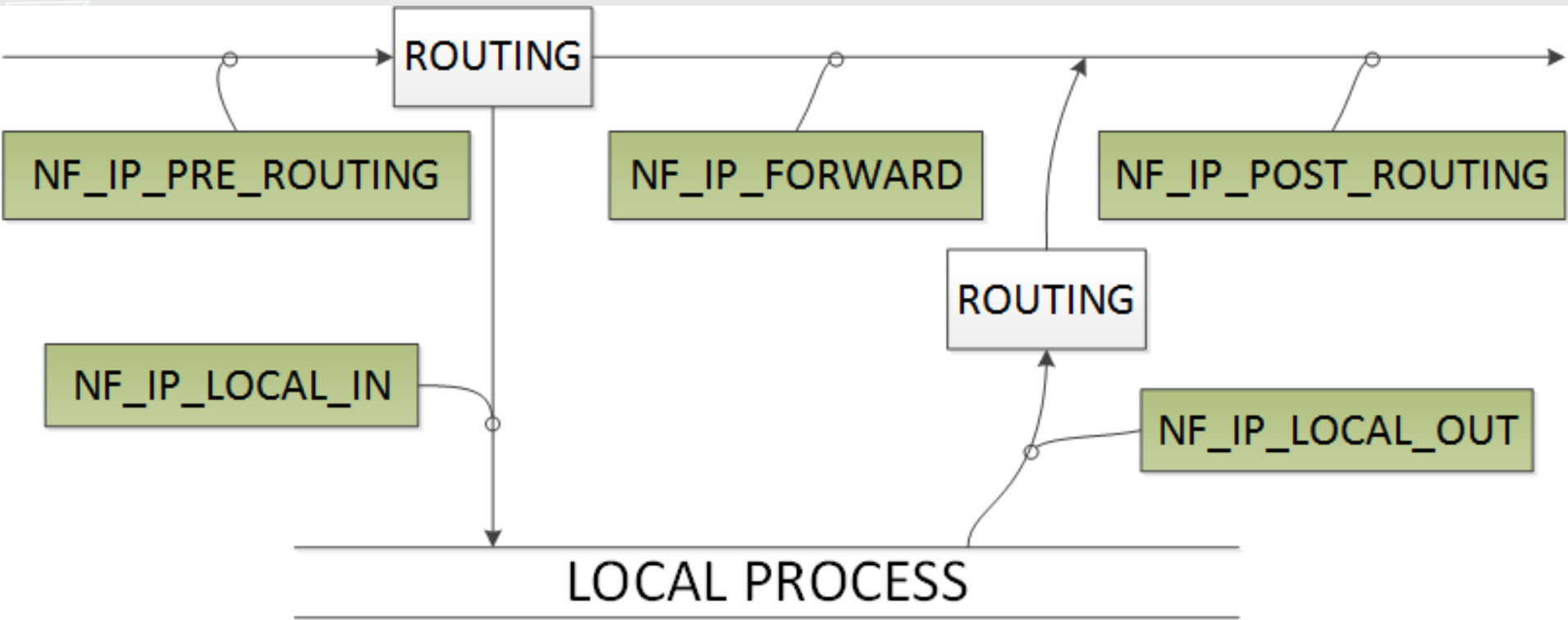
Netfilter

Фреймворк на уровне ядра Linux, который предоставляет различные возможности для работы с сетевыми пакетами. Он позволяет:

- фильтровать пакеты
- выполнять NAT (Network Address Translation)
- реализовывать маршрутизацию и отслеживать соединения

Netfilter функционирует на уровне ядра, и именно через него происходит большинство операций по обработке сетевых пакетов в Linux.

Хуки netfilter



Хуки netfilter

Исправленный маршрут пакетов:

1. Входящий пакет для локального процесса:

NF_IP_PRE_ROUTING → NF_IP_LOCAL_IN → LOCAL
PROCESS

2. Пакет для пересылки (форвардинга):

1. NF_IP_PRE_ROUTING → NF_IP_FORWARD → NF_IP_P
OST_ROUTING

3. Исходящий пакет, созданный локальным процессом:

1. NF_IP_LOCAL_OUT → NF_IP_POST_ROUTING

Действия netfilter

Какие действия может совершить перехват netfilter:

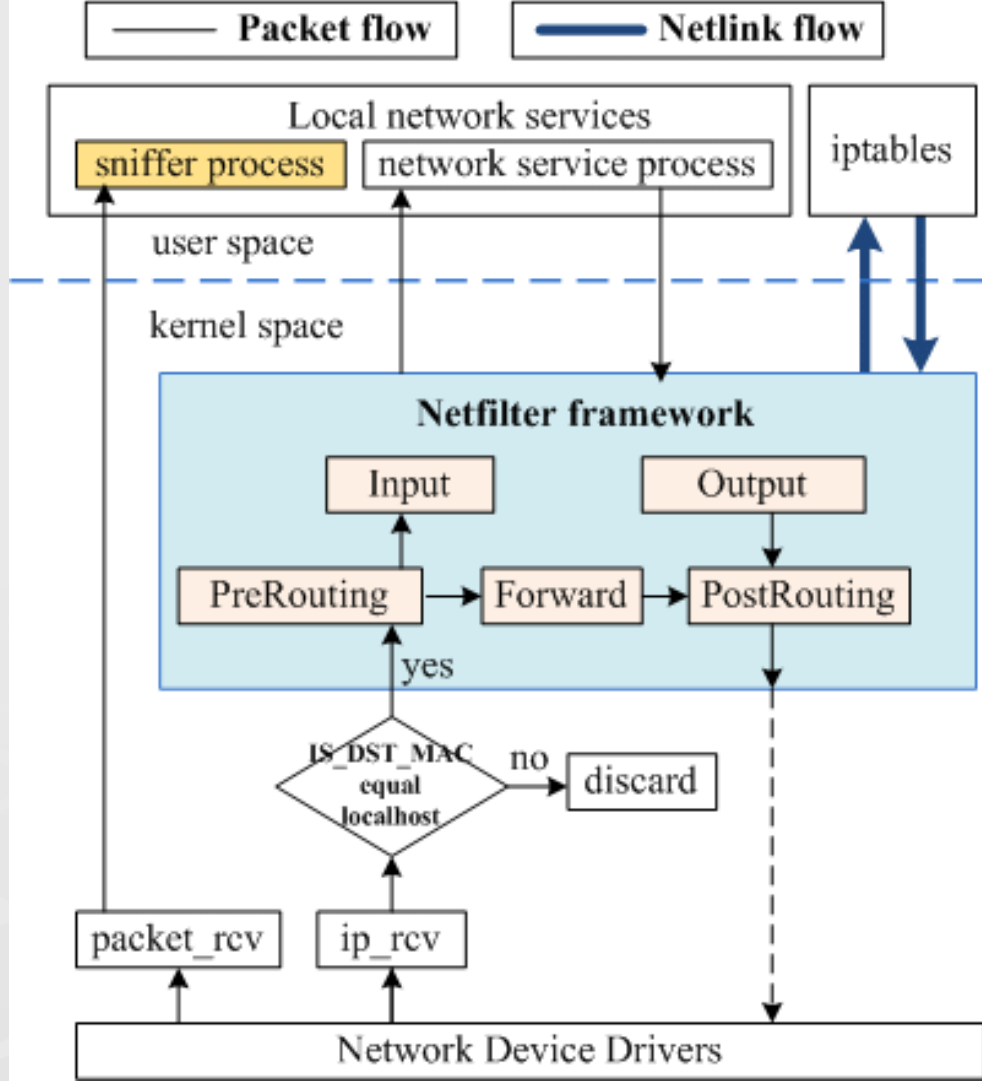
Accept - продолжить обработку пакета

Drop - сбросить пакет без дальнейшей обработки

Queue - Передать пакет программе пространства пользователя

Stolen - Дальнейшие перехваты не выполняются, пакет теперь принадлежит программе пространства пользователя

Repeat - Пакет снова попадает на перехват и переобрабатывается



Перехват Netfilter	Имя Цепочки iptables	Описание	Применение	Типы пакетов
NF_IP_PRE_ROUTING	PREROUTING	Перехват, выполняющийся до маршрутизации пакета	Используется для манипулирования пакетами перед тем, как будет определено их направление (маршрут).	Входящие пакеты (все интерфейсы).
NF_IP_LOCAL_IN	INPUT	Перехват, выполняющийся для пакетов, адресованных локальной системе.	Используется для фильтрации и обработки пакетов, предназначенных для локальной машины.	Входящие пакеты, направленные в локальную систему.
NF_IP_FORWARD	NAT	Перехват, выполняющийся для пакетов, которые будут пересланы через систему (маршрутизация).	Используется для фильтрации и обработки пакетов, которые должны быть пересланы с одного интерфейса на другой.	Пакеты, проходящие через систему (маршрутизируемые).
NF_IP_LOCAL_OUT	OUTPUT	Перехват, выполняющийся для пакетов, исходящих из локальной системы.	Используется для фильтрации и обработки пакетов, отправляемых локальной машиной.	Исходящие пакеты, созданные локальной системой.
NF_IP_POST_ROUTING	POSTROUTING	Перехват, выполняющийся после маршрутизации пакета, но перед его отправкой на интерфейс	Используется для манипулирования пакетами после того, как был определен их маршрут	Все исходящие пакеты (после маршрутизации).

Conntrack

Компонент системы Netfilter для контроля состояния соединений (внешнего и внутреннего) с компьютером. С помощью функций контроля соединения пакеты автоматически привязываются к своим соединениям и легко модифицируются через SNAT, DNAT.

Conntrack

Conntrack использует хуки Netfilter, чтобы отслеживать состояния соединений, каждый раз обновляя таблицу соединений. Сама таблица соединений является хэш таблицей, для которой используется следующий кортеж:

- IP источника
- IP назначения
- порт источника
- порт назначения
- протокол (tcp\udp)

Затем этот кортеж хэшируется и становится ключом таблицы, в дальнейшем поиск строки соединения осуществляется по хэшу и меняется состояние в зависимости этапа.

Conntrack состояния

Состояние	Описание	Пример
NEW	Пакет отослан или получен, ответ не получен.	Получен TCP SYN
ESTABLISHED	Пакеты пересылаются в обоих направлениях.	Получен TCP SYN, передан TCP SYN/ACK
RELATED	Открыто дополнительное соединение, метаданные которого указывают, что оно имеет отношение к основному соединению.	FTP программа со статусом соединения ESTABLISHED открывает дополнительные соединения для обмена данными.
INVALID	Пакет испорчен или не соответствует другим состояниям Conntrack	Получен TCP RST без предварительного соединения.

iptables

Используется для модификации и переадресации пакетов, iptables использует Netfilter, что позволяет перехватывать пакеты и изменять пакеты.

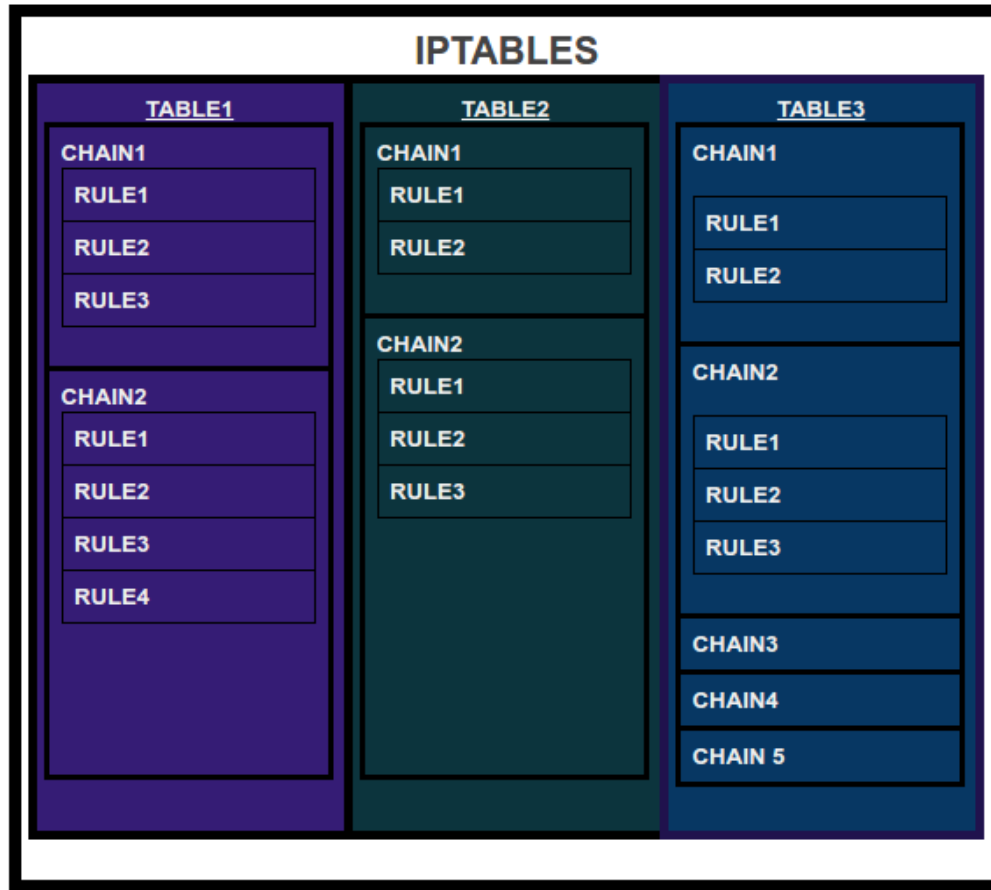
iptables состоит из трех основных элементов:

Таблицы

- Цепочки

- Правила

iptables



Таблицы iptables

Filter

Определяет, будет ли пакет принят или отфильтрован.

NAT

Используется для изменения IP-адресов источника или назначения.

Mangle

Может производить изменения в заголовках пакета, не относящихся к NAT. Также может отличать пакеты с метаданными, доступными только для iptables.

Raw

Позволяет модифицировать пакет перед тем, как будет произведен контроль состояния соединения и перед вычислениями на базе остальных таблиц. Обычно используется для исключения функции контроля состояния соединения для определенных пакетов.

Цепочки iptables

Цепочки IPtables - это списки правил. Когда пакет проходит через цепочку, то последовательно выполняются вычисления на основании правил, пока пакет не попадет на завершающее действие или не достигнет конца цепочки.

Функционируют на основании перехватов netfilter, каждая цепочка соответствует одному перехвату

Цепочки iptables

Встроенные цепочки верхнего уровня это:

Prerouting - NF_IP_PRE_ROUTING

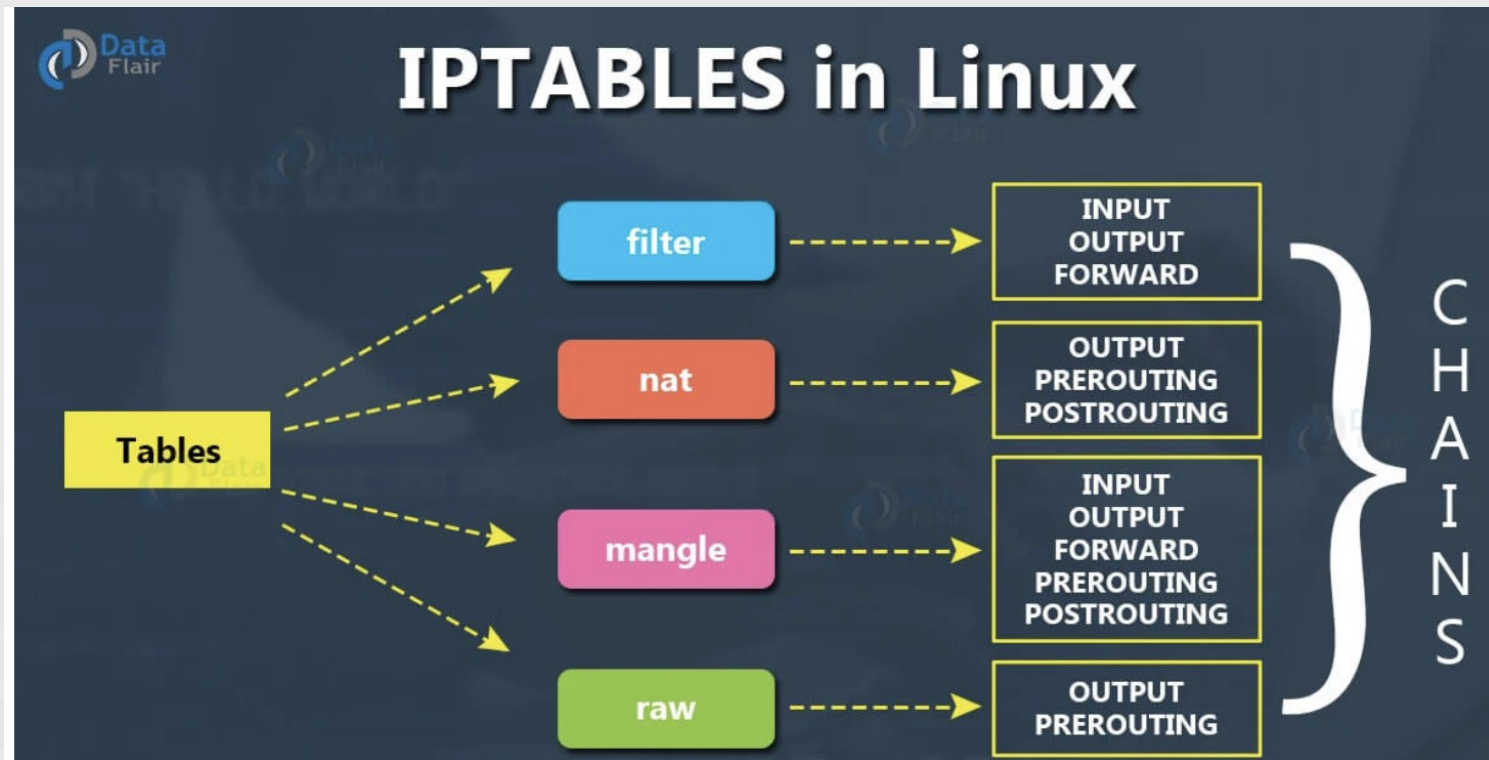
Input - NF_IP_LOCAL_IN

Nat - NF_IP_FORWARD

Output - NF_IP_LOCAL_OUT

Postrouting - NF_IP_POST_ROUTING

Таблицы - цепочки



Правила

Правила состоят из двух частей:

1. Условие
2. Действие

Существуют 2 типа действий:

1. Прекращающие
2. Непрекращающие.

Непрекращающие позволят продолжить действие по цепочке

Прекращающие действия заканчивают движение по цепочке: Accept, Drop, Reject, Return.

Правила пример

```
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -P INPUT DROP
```

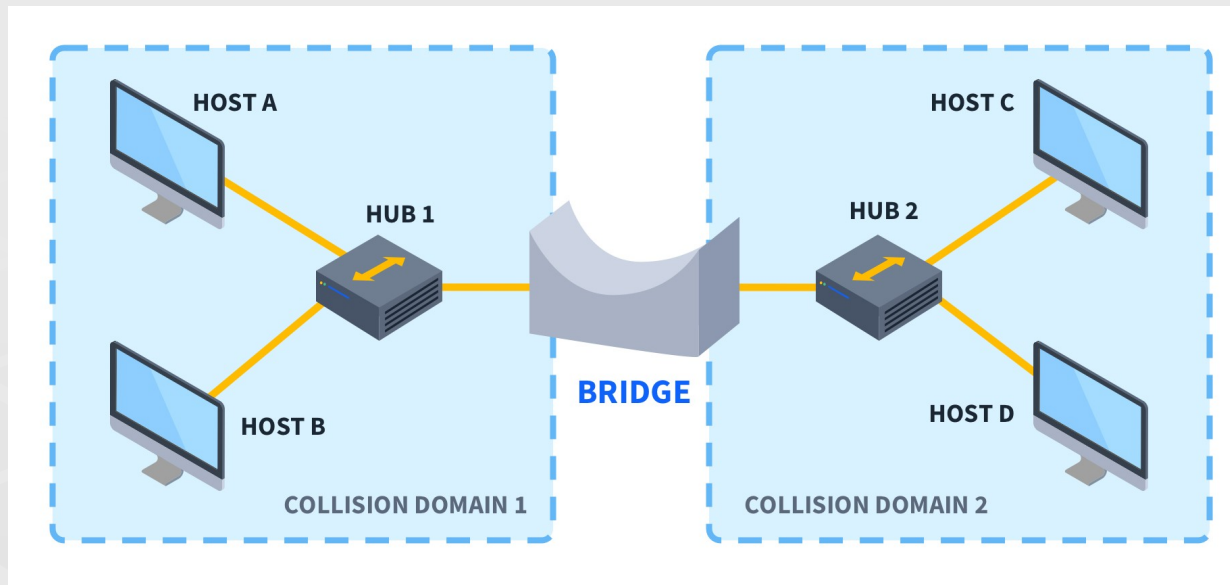
```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
iptables -A OUTPUT -d 192.168.1.100 -j REJECT
```

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

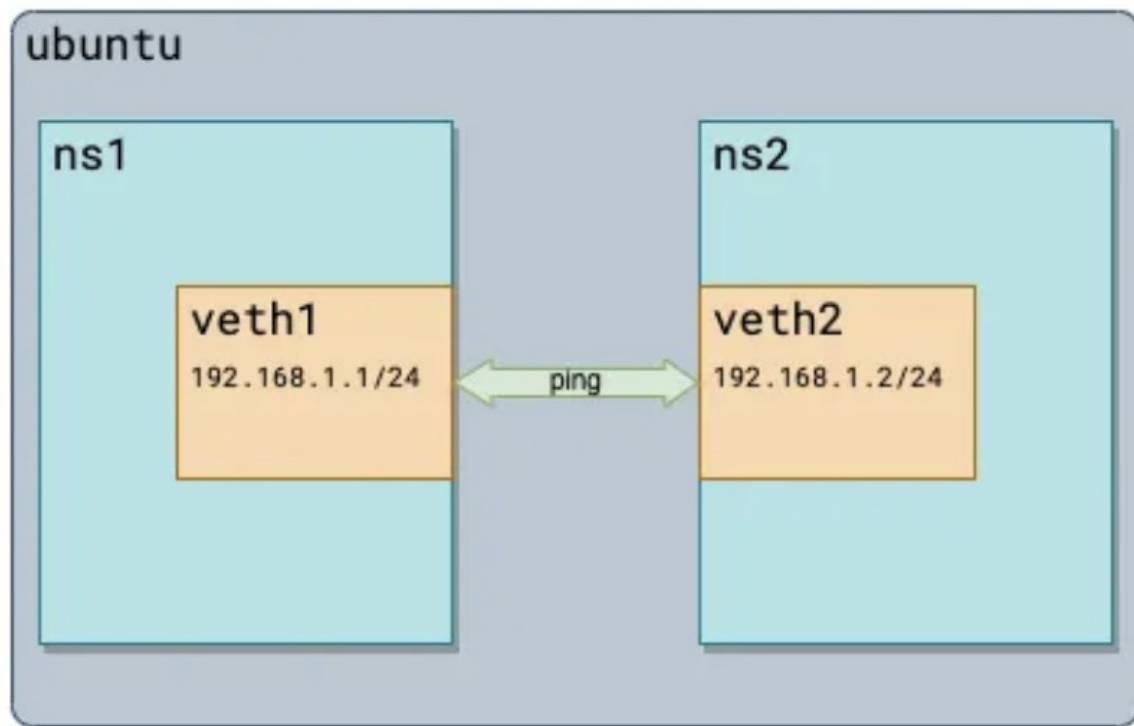
Интерфейс типа мост (bridge)

Это способ соединения двух сегментов Ethernet на канальном уровне (L2), без использования протоколов более высокого уровня.



Интерфейсы veth

Виртуальный сетевой интерфейс, который используется для соединения двух сетевых пространств.

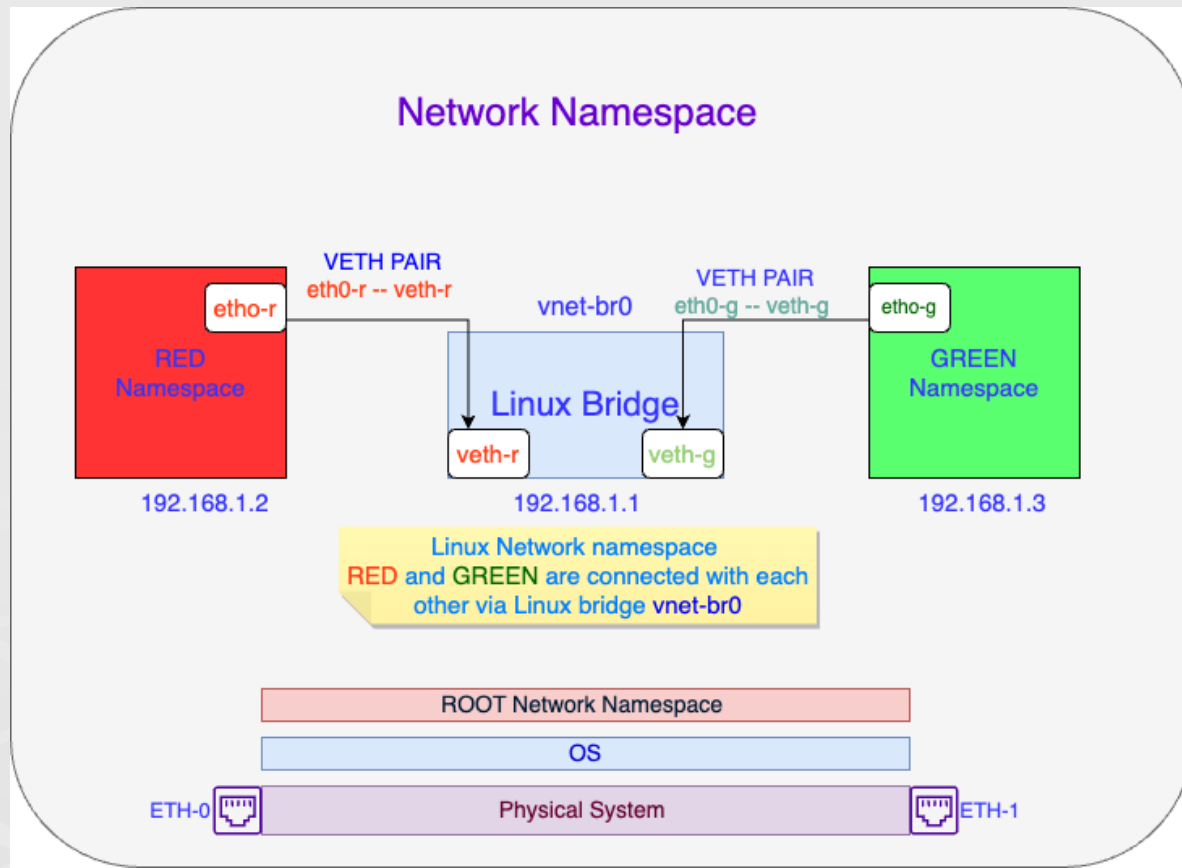




02

net namespace

Пространство имен сети



Пространство имен сети (veth)

Namespace
1

```
sudo ip netns add ns1
```

Namespace 2

```
sudo ip netns add ns2
```

```
eu@eu-VMware-Virtual-Platform:~$ sudo ip netns  
ns1  
ns2
```

eth0

Пространство имен сети (veth)

Namespace
1

Namespace 2

eth0

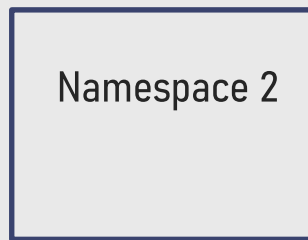
```
eu@eu-VMware-Virtual-Platform:~$ sudo ip netns exec ns1 ip link  
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
eu@eu-VMware-Virtual-Platform:~$ sudo ip netns exec ns2 ip link  
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

Пространство имен сети (veth)



veth-ns1

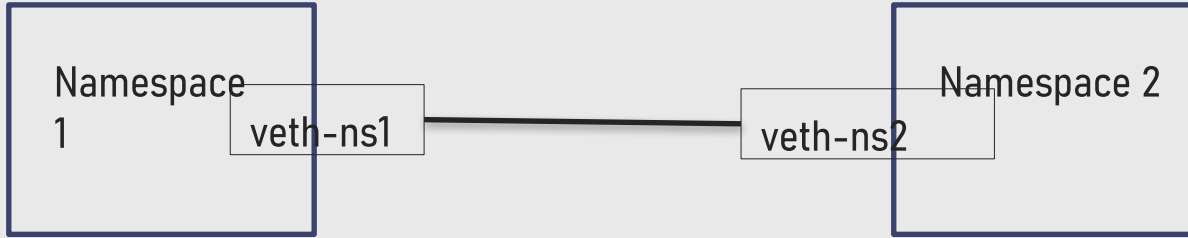


veth-ns2



```
sudo ip link add veth-ns1 type veth peer name veth-ns2
```

Пространство имен сети (veth)



```
sudo ip link set veth-ns1 netns ns1
```

```
sudo ip link set veth-ns2 netns ns2
```

eth0

Пространство имен сети (veth)



```
sudo ip -n ns1 addr add 192.168.1.1/24 dev veth-ns1
```

```
sudo ip -n ns1 link set veth-ns1 up
```

```
sudo ip -n ns2 addr add 192.168.1.2/24 dev veth-ns2
```

```
sudo ip -n ns2 link set veth-ns2 up
```

eth0



02

Сетевая модель Kubernetes

Принципы модели

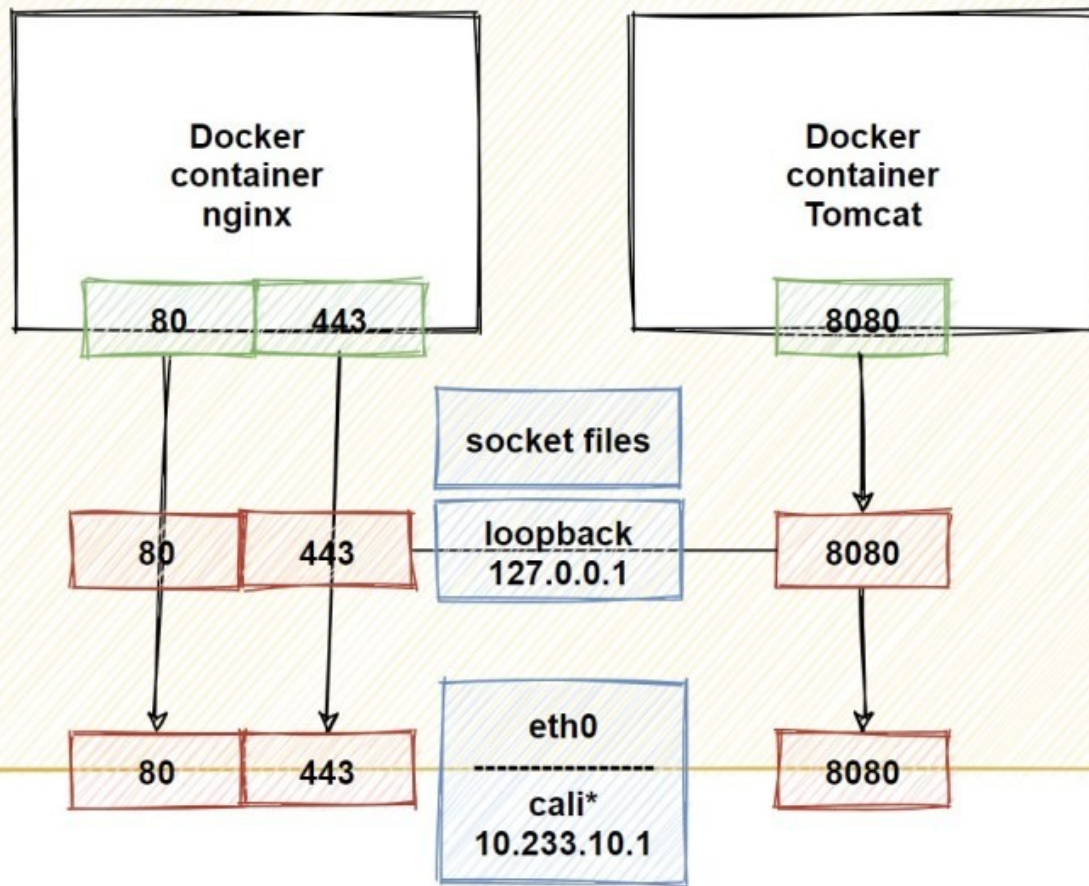
Каждый под имеет собственный IP адрес.

Контейнеры внутри пода имеют общий IP адрес и могут свободно обмениваться пакетами друг с другом.

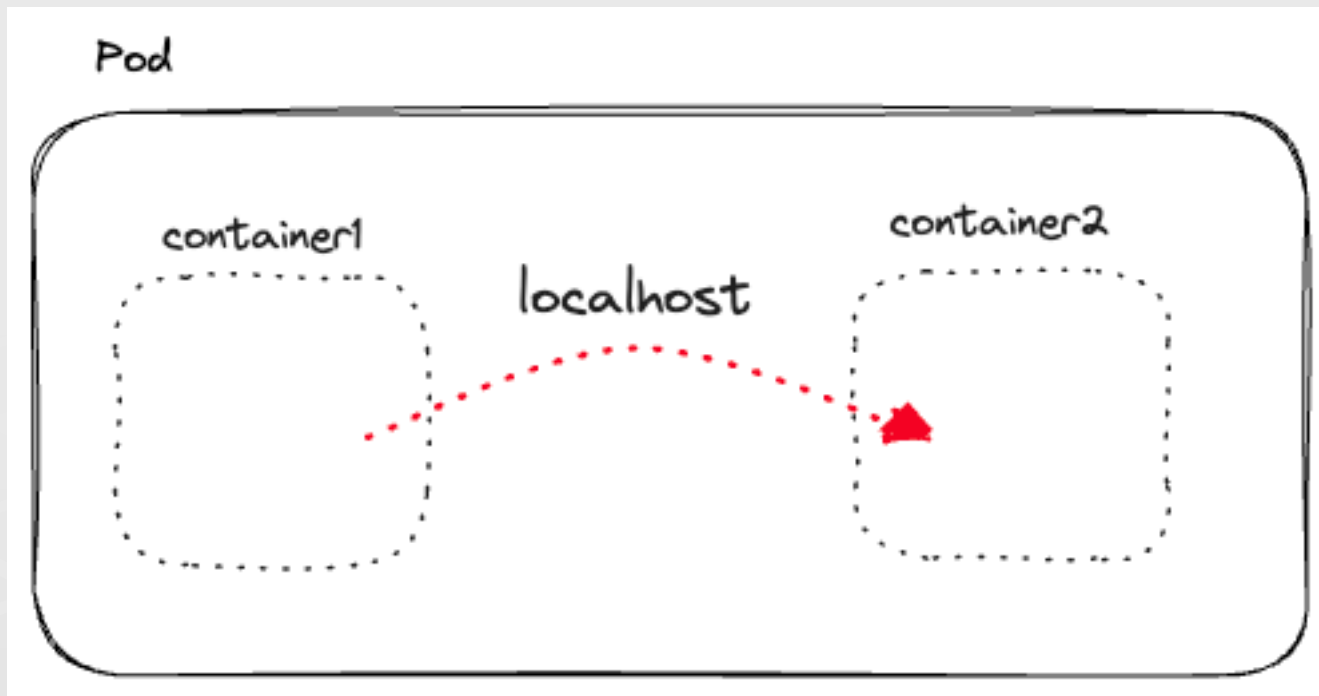
Поды могут посылать пакеты другими подами в кластере, используя их IP адреса. Без применения NAT.

Ограничения в хождении пакетов между подами определяется при помощи сетевых политик.

Pod



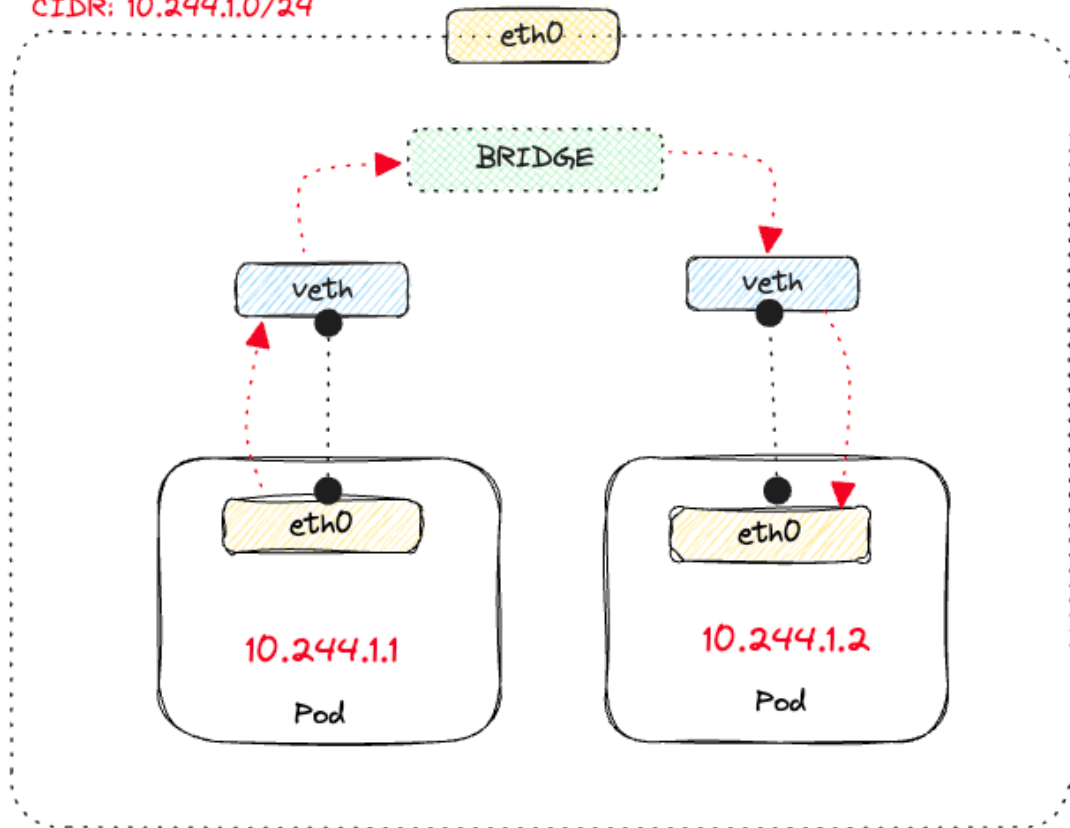
Контейнер-контейнер коммуникация



Под-под коммуникация

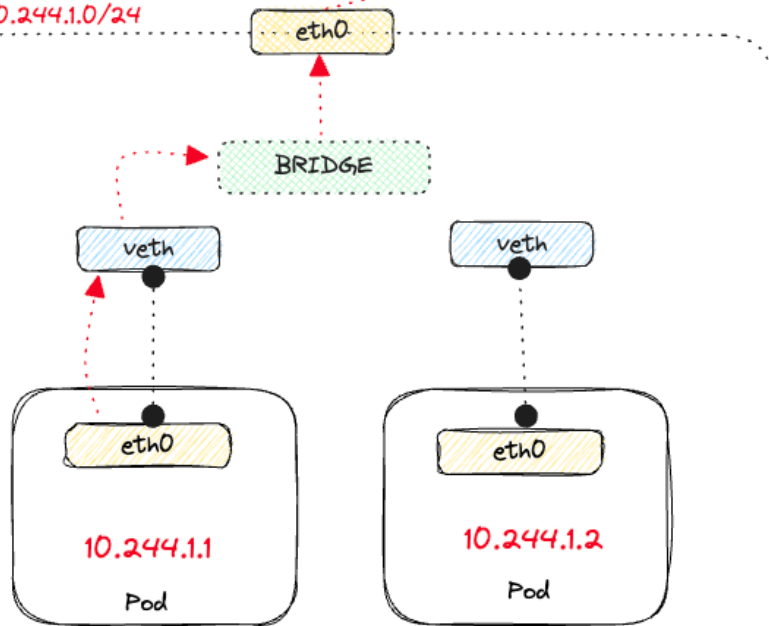
node1

CIDR: 10.244.1.0/24

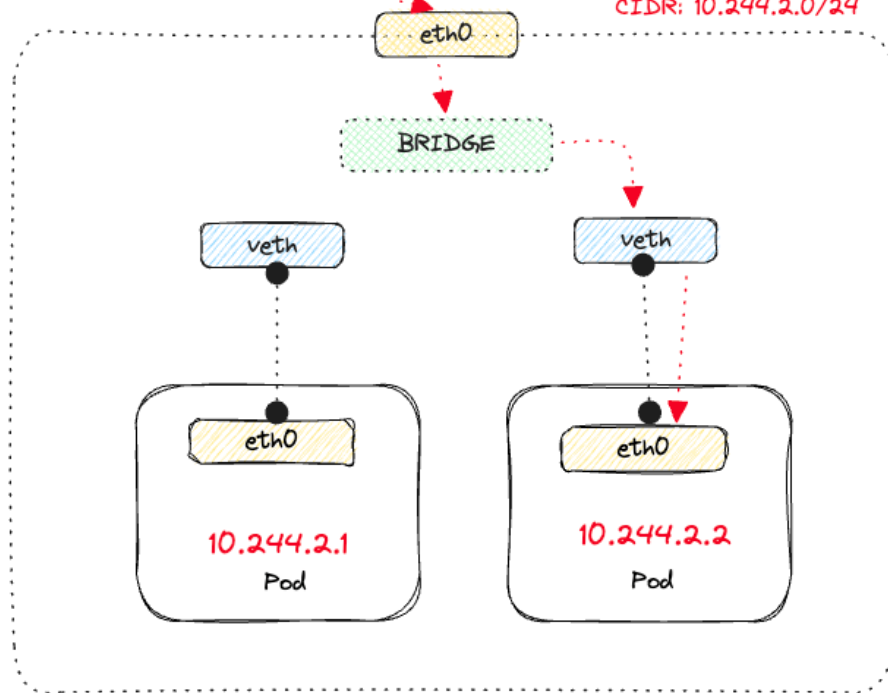


Под-под коммуникация на разных нодах

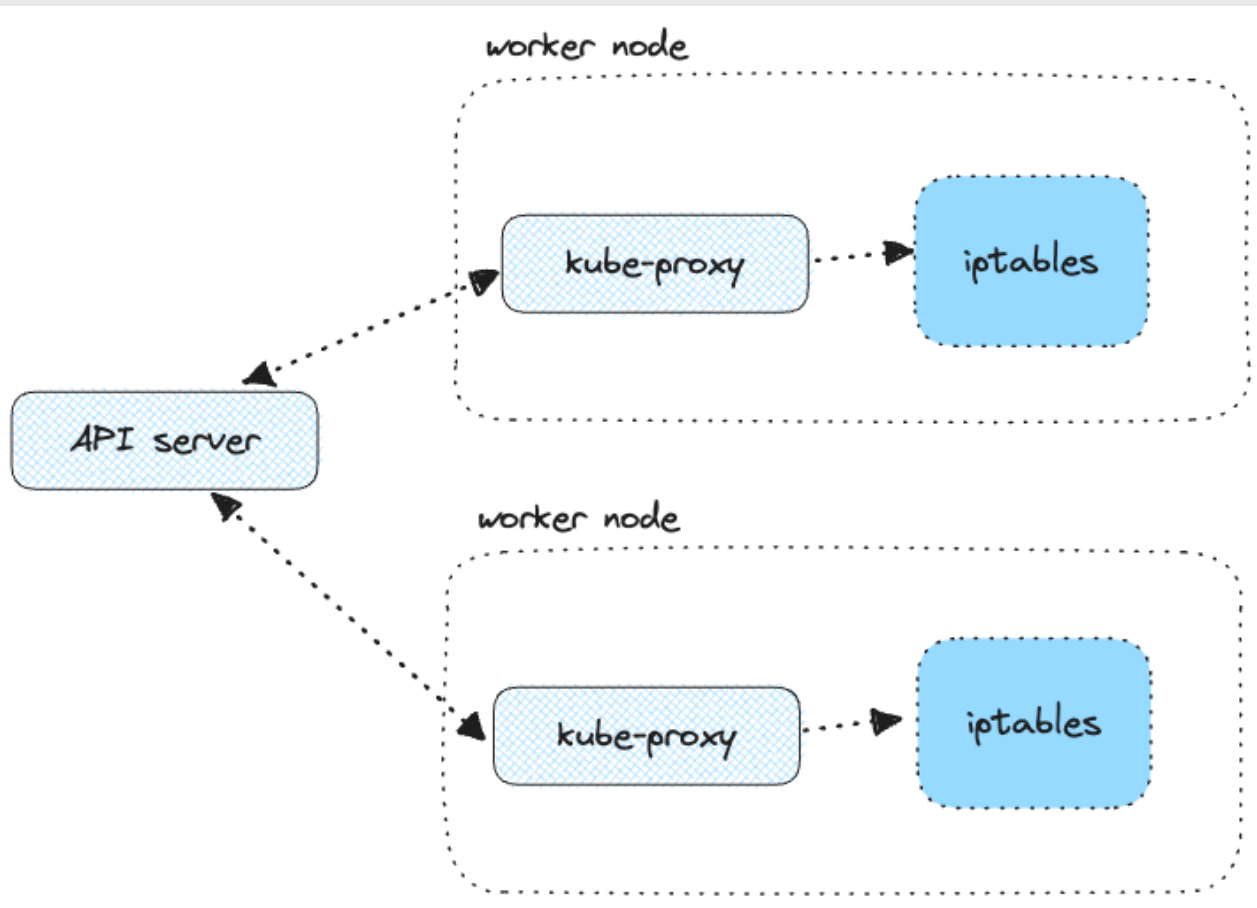
node1 (172.18.0.2)
CIDR: 10.244.1.0/24



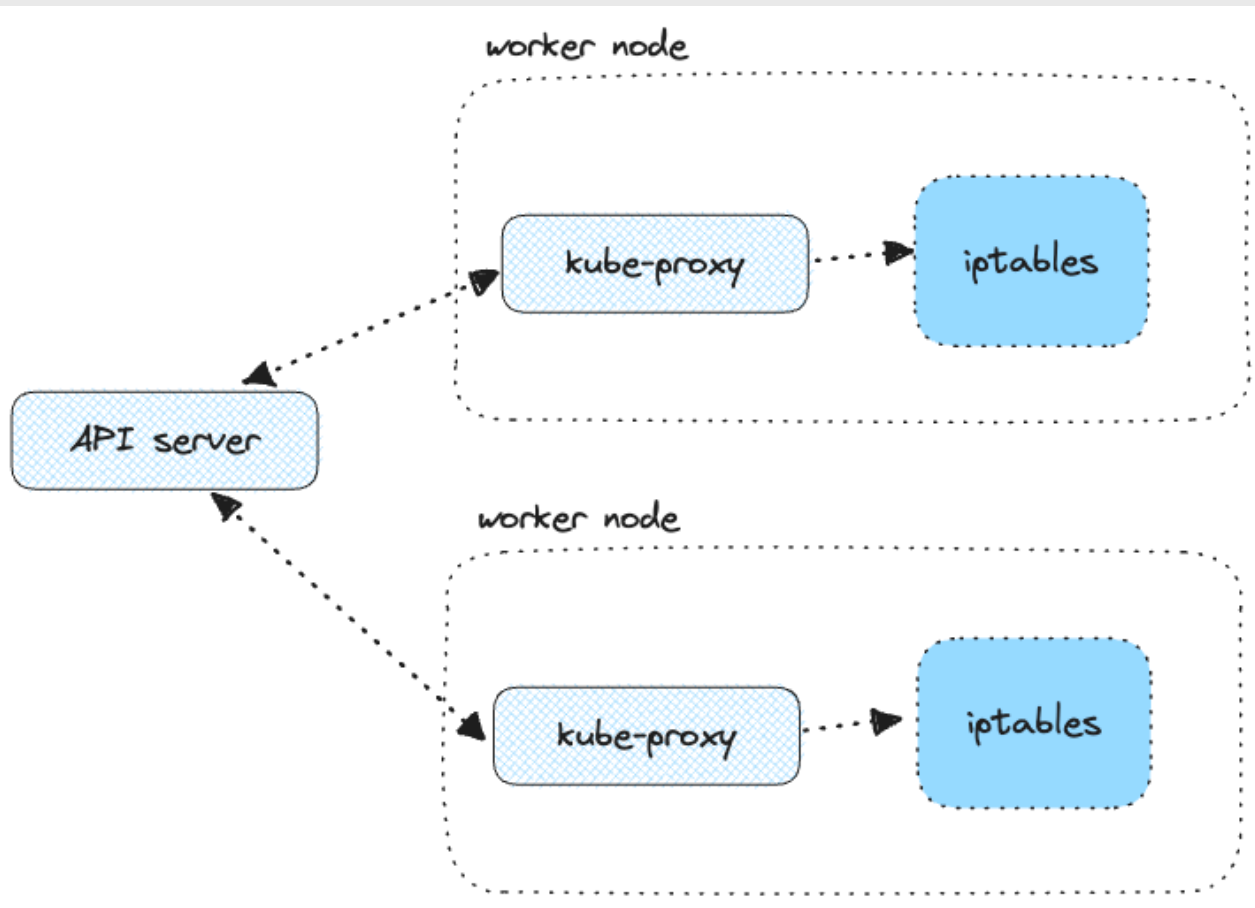
node2 (172.18.0.3)
CIDR: 10.244.2.0/24



Kube-proxy



Kube-proxy



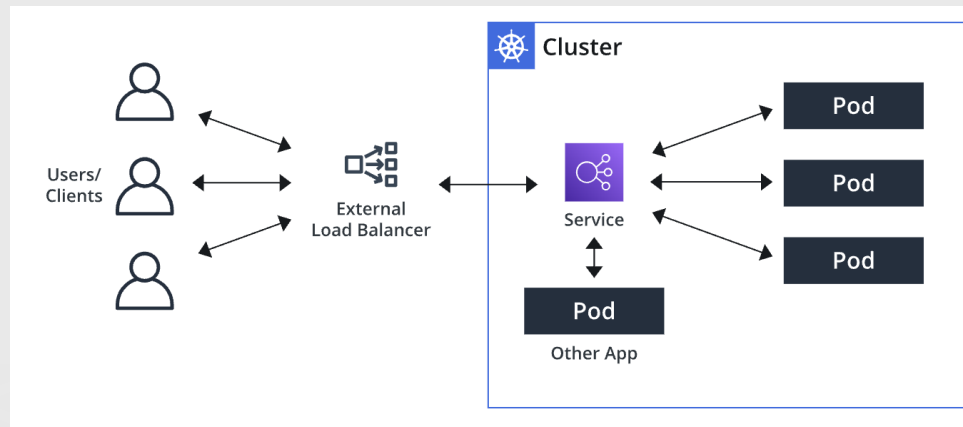


03

Сервисы Kubernetes

Сервисы

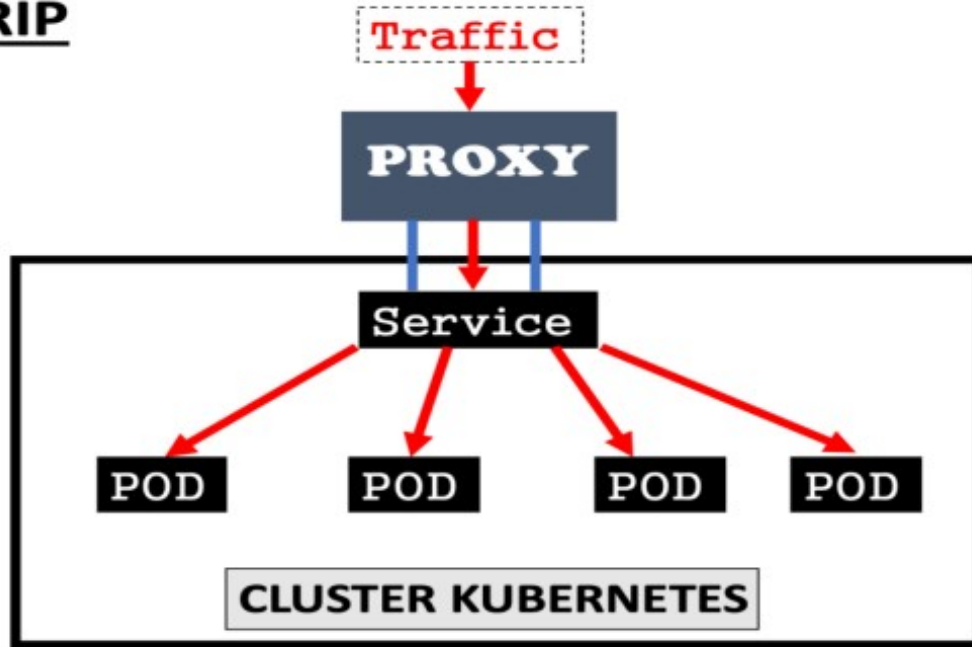
Абстракция, которая обеспечивает стабильный сетевой доступ к набору подов в кластере Kubernetes. Поды в Kubernetes являются временными — они могут быть перезапущены, удалены или пересозданы. Это приводит к изменению их IP-адресов. Services решают эту проблему, предоставляя постоянную точку доступа к подам, даже если их IP-адреса изменяются.



<https://kubernetes.io/docs/tutorials/kubernetes-basics/expose/expose-intro/>

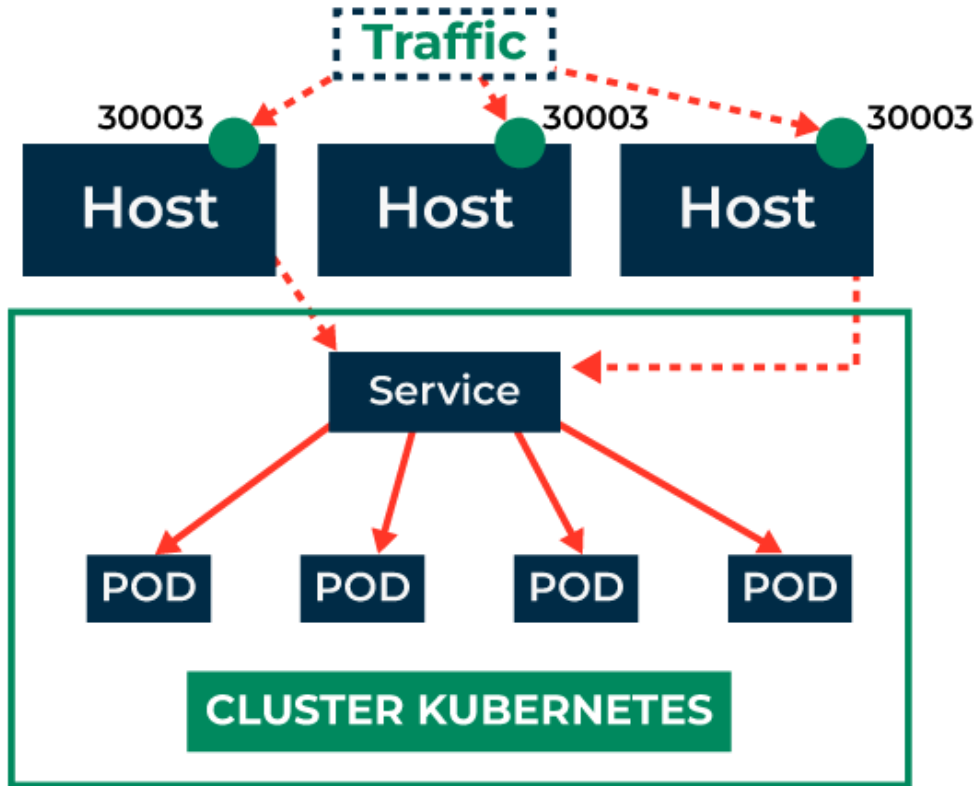
Cluster IP

CLUSTERIP

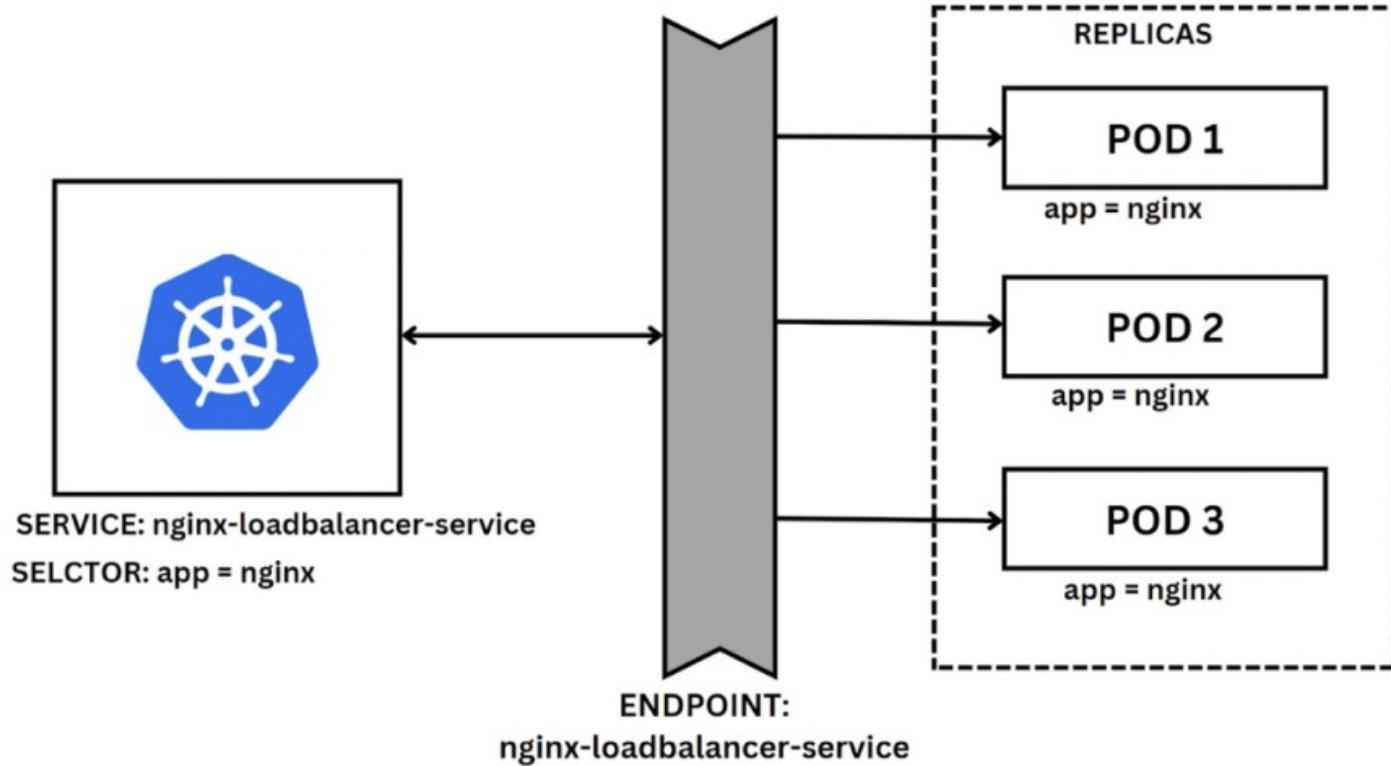


<https://kubernetes.io/docs/tutorials/kubernetes-basics/expose/expose-intro/>

Node port



Load balancer





04

Безопасность Kubernetes

Immutable OS

WHAT IS TALOS?

TALOS LINUX

API Managed,
declarative,
minimal Linux for
K8s

1

GETS KUBERNETES UP FAST

Talos installs vanilla K8s for you,
making clusters simple.

2

CONSISTENT: CLOUDS, BARE METAL, EDGE

Runs the same everywhere

3

TRUE MULTICLOUD BUILT IN

Built-in KubeSpan enables clusters
to span networks, clouds or edge.

4

SECURE BY DESIGN

Hardened Linux per KSPP, K8s
per STIG and CIS guidelines;

5

MINIMAL, IMAGE BASED OS

50MB image; Fast to boot,
minimal attack surface, no SSH.

Admission controllers

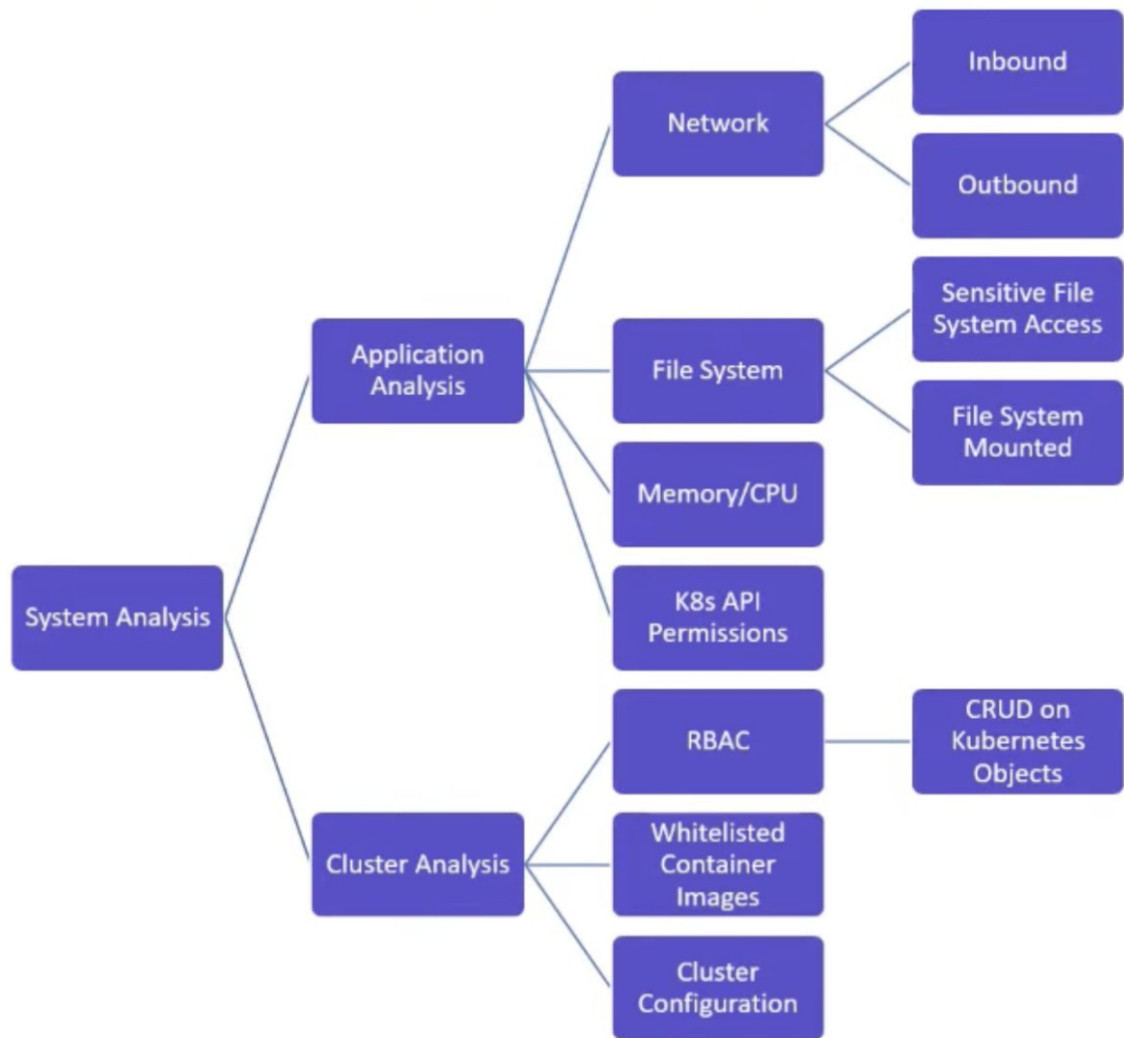


IDS



Falco





1) Scan ports and identify service with vulnerable remote code execution

2) Metasploit installation

3) Leverage kernel vulnerability to break out of the container

4) Replace running service with a malicious program to exfiltrate data

Reconnaissance

Weaponization

Delivery

Exploitation

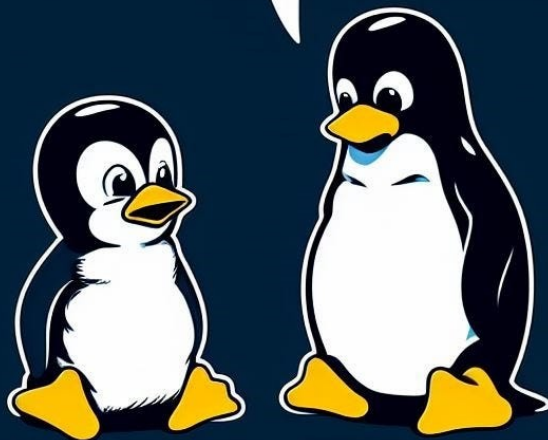
Installation

Command and Control

Action on Objectives



**DON'T LET HIM
FOOL, YOU, KID.**



**KUBERNETES JUST
LINUX IN A SUIT**