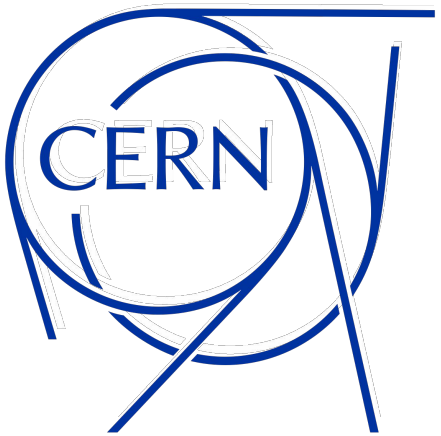


Computer Networks in HEP

Spyros Trigazis, IT-PW-PI, spyridon.trigazis@cern.ch

17 April 2024

Agenda

- **Networking basics**
  - Protocols, Concepts
- **CERN Networks**
  - DNS, DHCP, Firewall, LanDB
- **Cloud networking**
  - Data Centers, Virtual Networks, Security Groups, Load Balancers
- **Kubernetes networking**
  - Pods, Services, Ingress, CNI

Networking Basics

What is a Network?

A network is collection of computers, connected together so that they can exchange data.

Characteristics of a network:

- Bandwidth
- Latency
- Availability
- Scalability
- Security

Network Components

Network components include the end nodes and intermediate nodes.

- End nodes - compute, storage, and management nodes. These are used as the source or destination of the traffic carried in the network.

- Intermediate nodes - switches and routers receive the traffic generated by the end nodes and make decisions where to forward the traffic.

Forming a Network

Build a Simple Network

- Every end node has a network interface card, or NIC.
- The NIC is a hardware component that resides in a slot and has one or more network ports.
- Use a cable to connect the network interface card (NIC) in the end point to a switch port

Protocol Suite

Communication between nodes requires interaction between several protocols, where each protocol is responsible for a specific set of activities.

Each protocol defines a common format and set of rules for exchanging messages between devices.

- Setup and termination of data transfer sessions
- Message format
- Error and system messages

A protocol suite is a group of protocols that run concurrently to implement network communication.

OSI Model - Open Systems Interconnection

The OSI model breaks down the different components of network communication into layers, in order to make communication easier.

- Defines 7 layers
- ISO standard

OSI is a generic, protocol-independent standard

Why a layered network model?

- Standardizes interfaces
- Facilitates modular engineering
- Ensures interoperable technology
- Accelerates evolution

OSI Model	
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

TCP/IP Model

TCP/IP model is, in a way, an implementation of the OSI model.

- In the 1970s, the U.S. Department of Defense developed a network called ARPAnet that enabled data transport between computers at different locations by creating alternate routes.
- TCP/IP was used as the system protocol
- TCP/IP is the protocol for the Internet

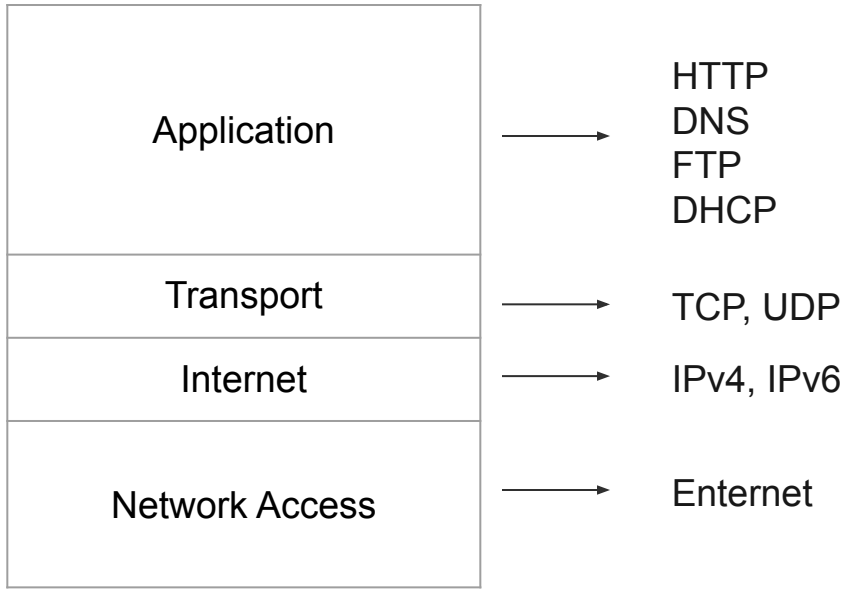
<table>
<tr><th>TCP/IP Model</th><th colspan="2">OSI Model</th></tr>
<tr><td rowspan="3">Application</td><td>7</td><td>Application</td></tr>
<tr><td>6</td><td>Presentation</td></tr>
<tr><td>5</td><td>Session</td></tr>
<tr><td>Transport</td><td>4</td><td>Transport</td></tr>
<tr><td>Internet</td><td>3</td><td>Network</td></tr>
<tr><td rowspan="2">Network Access</td><td>2</td><td>Data Link</td></tr>
<tr><td>1</td><td>Physical</td></tr>
</table>

TCP/IP Protocol Suite

TCP (Transmission Control Protocol)

IP (Internet Protocol)

- TCP/IP suite is a set of communications protocols used on the Internet.
- It includes protocols that specify how data is packetized, addressed, transmitted, routed, and received, thus providing end-to-end data communication.
  - Internet layer protocols - IPv4 and IPv6 (layer 3)
  - Transport layer protocols - TCP and UDP (layer 4)
  - Application layer protocols - HTTP, FTP, etc. (layer 7)



Encapsulation

Application messages created by the application layer are passed to a “production line” of the lower layers, where each layer adds its relevant information, encapsulating the data with the layer header.

By the end of the encapsulation process, the encapsulated data is transmitted on the wire.

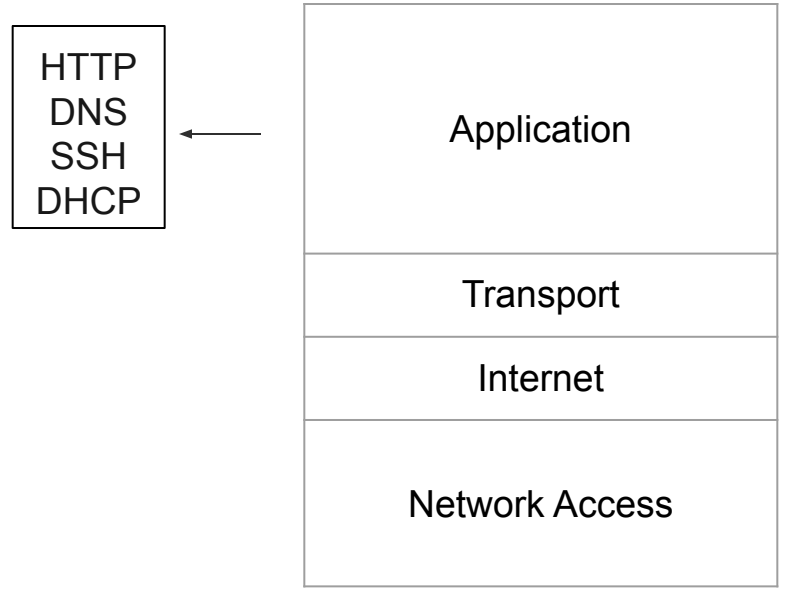
OSI Model	
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Application Layer Protocols

Remote applications implementing the same protocol exchange messages over the network (eg webserver).

The application layer provides network services to data generated by applications.

Whenever an application layer protocol wishes to communicate with a remote application, it hands over the data to the transport layer.



HTTP – Hyper Text Transfer Protocol

HTTP is an application layer protocol to to exchange text and logical links (hyperlinks) between nodes.

HTTPS is the secured variant of HTTP.

HTTP functions as a request–response protocol in the client–server computing model.

- The client submits an HTTP request message to the server.
- The server, which provides resources such as HTML files and other content types, returns a response message to the client.
- The response contains the request's completion status information and may also contain requested content in its message body.

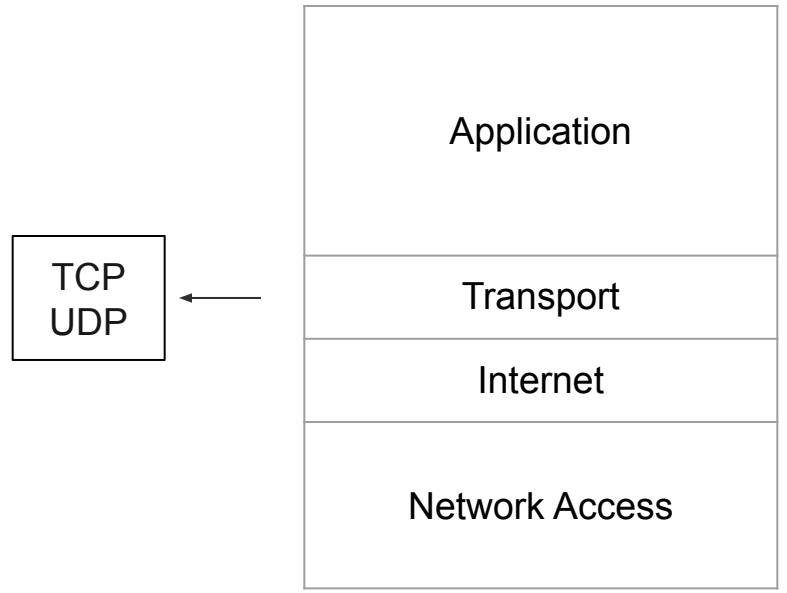
Read more about at the entrance of building 513.

Transport Layer Protocols

Transport layer protocols establish end-to-end logical communication channels between applications.

The transport layer functions are:

- Facilitating the communicating hosts to carry on a conversation
- Asking for layer 3 protocols to carry data to the destination
- Providing a reliable connection.
  It can also carry out error checking, flow control and verification
- TCP and UDP operate at the transport layer.



TCP – Transmission Control Protocol

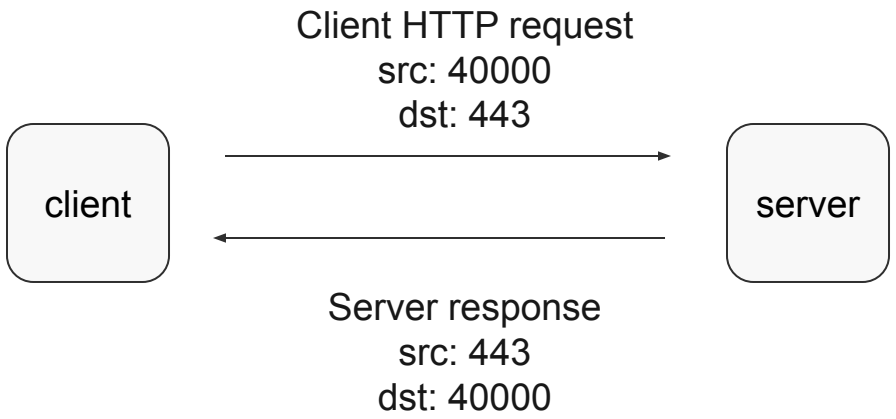
- TCP is a reliable, connection-oriented transport protocol.
  - **Connection oriented** - a connection is established between the peer entities prior to transmission.
  - **Reliable** - TCP uses sequence numbers and acknowledgments (ACKs) to recover from damaged or lost data.
  - **Flow control** - TCP allows the receiver to govern the amount of data sent by the sender.
    TCP uses a “window” size to indicate the allowed number of bytes the sender may transit before receiving further permission.
  - **Ordered delivery** - Sequence numbers are used to order segments.
- Applications that require reliability use TCP as their transport layer protocol.
- TCP reliability mechanisms create overhead that slows down data transfer.
- Applications that require fast light-weight transport service, TCP is not the best choice.

UDP – User Datagram Protocol

- UDP provides a procedure for applications to communicate with each other with a minimum protocol overhead.
- UDP is an unreliable, connectionless protocol.
  - Connectionless - no session establishment
  - Unreliable - no ACKs and no re-transmissions
  - No flow control
  - No ordered delivery
- UDP performance:
  - Less resources
  - Less bandwidth
  - Faster transfer

Port Numbers

- Transport layer protocols, such as TCP and UDP, use source and destination port numbers to identify the specific process within the local and remote nodes.
  - The destination port is a well-known number, identifying the service requested on the remote server
  - The Source port is a dynamically allocated number identifying the process that sends the data

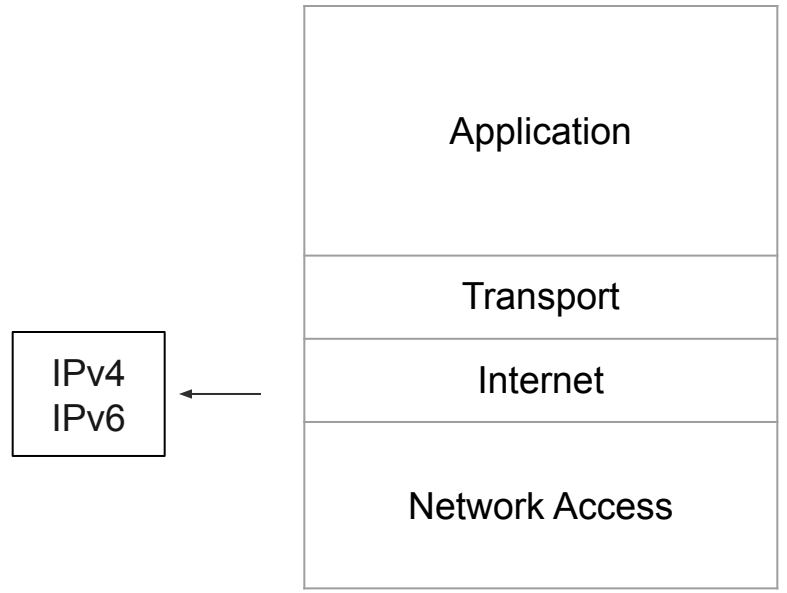


Well Known Port Numbers

- Port numbers are 16-bit, ranging 1-65,535 (0 is reserved)
- Port numbers in the range of 0-1023 are the well-known ports that are used by common TCP/IP applications.
- Port numbers 1024 to 49151 are registered to different applications.
  - Can be used without privileges unless the Operating System blocks them
  - eg installing packages on RHEL may block certain ports with firewallD or selinux

IP – Internet Protocol

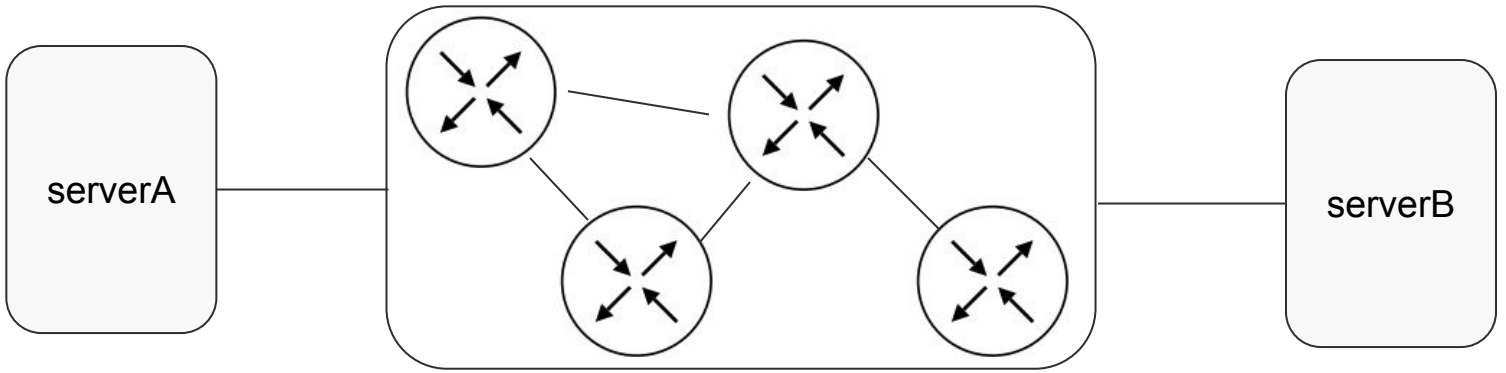
- IP operates in the network layer
- IP provides services to layer 4 protocols
- IP asks for layer 2 protocols to carry the IP packets
- IP is a best-effort protocol, as such, it DOES NOT include mechanisms for:
  - Reliability
  - Flow-control
  - Sequencing



IP Services

- IP provides the functions necessary to deliver a packet from a source to a destination over an interconnected system of networks.

- In order to fulfill these functions, IP provides the following:
  - Layer 3 addressing
  - Layer 3 packet format
  - Routing functionalities

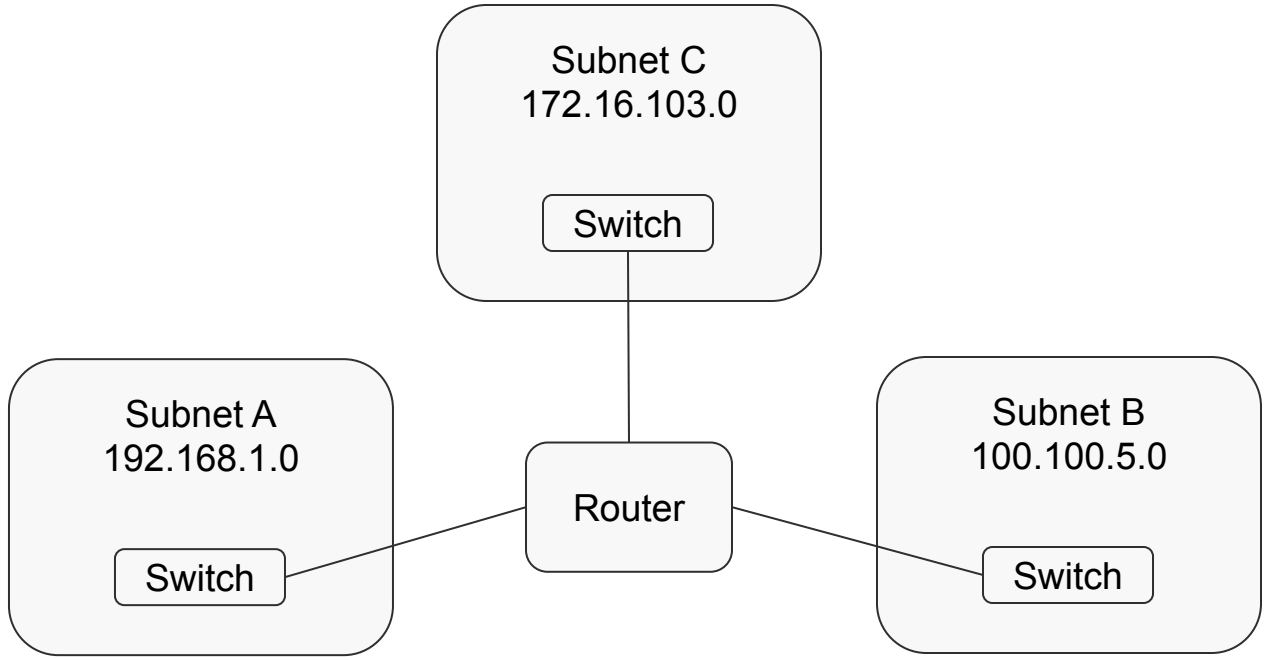


IP Subnets

- An IP address contain two parts:
  - Network address - identifies the network portion of the address
  - Host address - identifies a specific host within a network
- All hosts in the same subnet share a common subnet address
- Every router interface defines an IP subnet

10.254.0.22

Network Address | Host Address



IP Routers

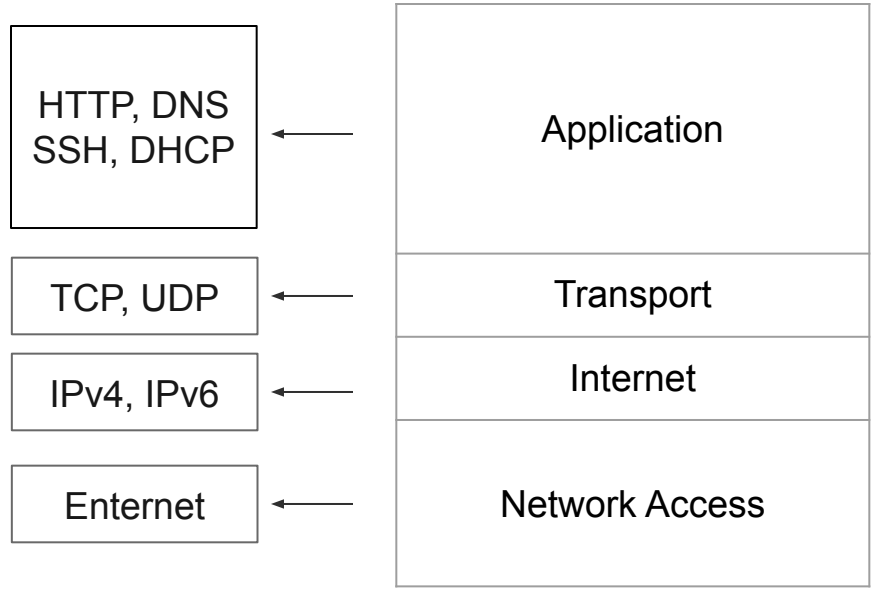
- Routing is the process of choosing the best path to reach a destination.

- A router is a computer that has been dedicated to the task of forwarding IP packets between networks.
  - Home router or even a Raspberry PI
  - Linux server on commodity hardware
  - Enterprise network appliance

- A router makes forwarding decisions based on its routing table.

TCP/IP

- The application layer protocol formats the message and requests transport services from the transport layer
- The transport layer (TCP/UDP) adds a layer 4 header and requests network services from network layer
- The network layer (IP) add a layer 3 header
  - The IP packet is ready to be carried over the physical medium
  - IP is a medium independent protocol, meaning it can be carried over different media types
- Ethernet is the most commonly used physical medium

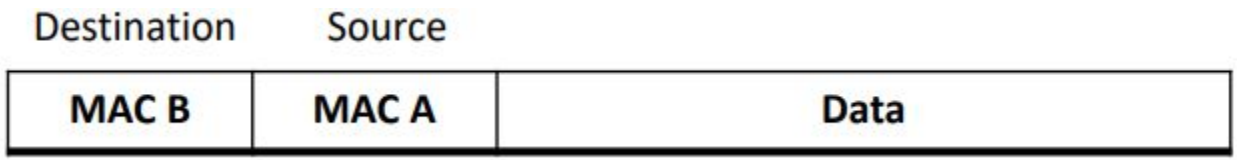


Network Access - L1/L2

Ethernet operates in the data link layer (L2) and the physical layer (L1) of the OSI model.

- Nodes send Ethernet frames to each other
- Ethernet nodes are identified with a unique address known as the **MAC address**
- The Ethernet frame header contains a source and a destination MAC address
- Forwarding decisions are based on the destination MAC address

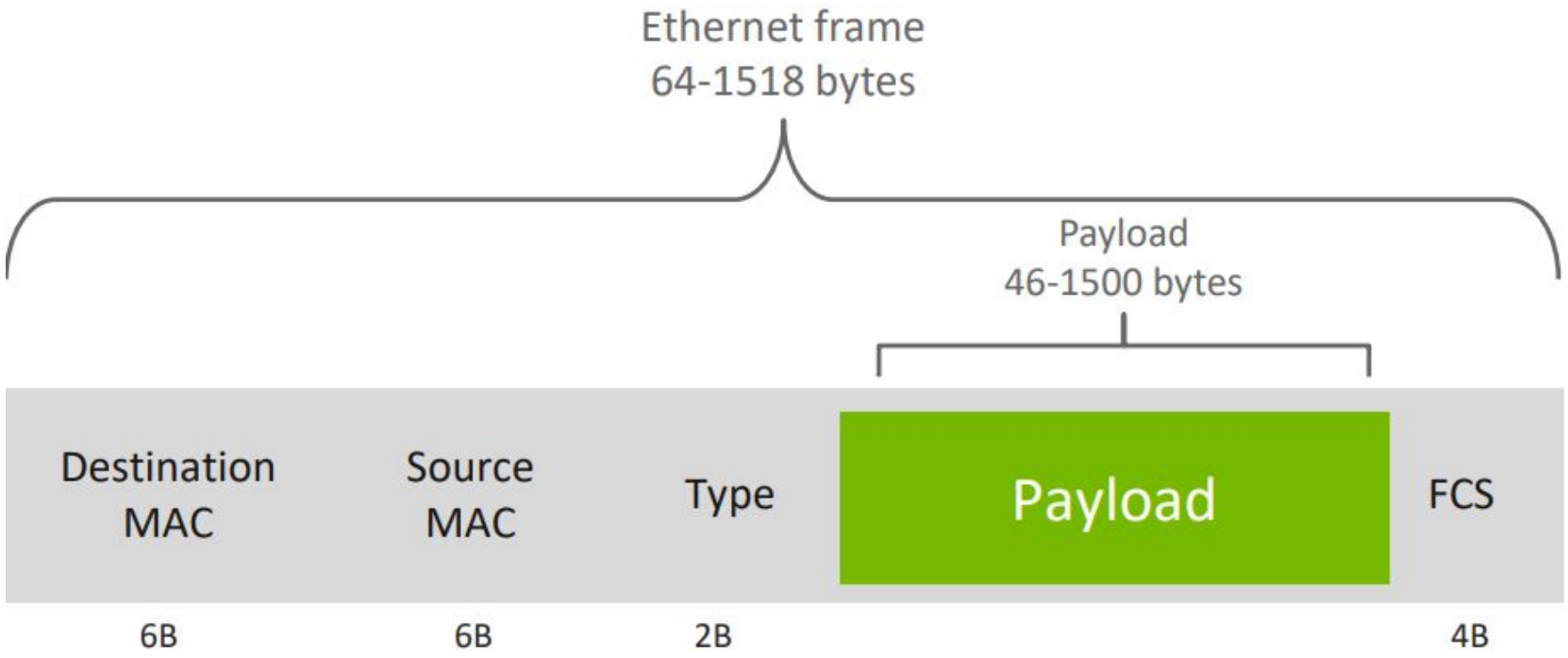
A media access control address (MAC) is a unique identifier assigned to a network interface for use as a network address.



Ethernet Frame Structure and MTU

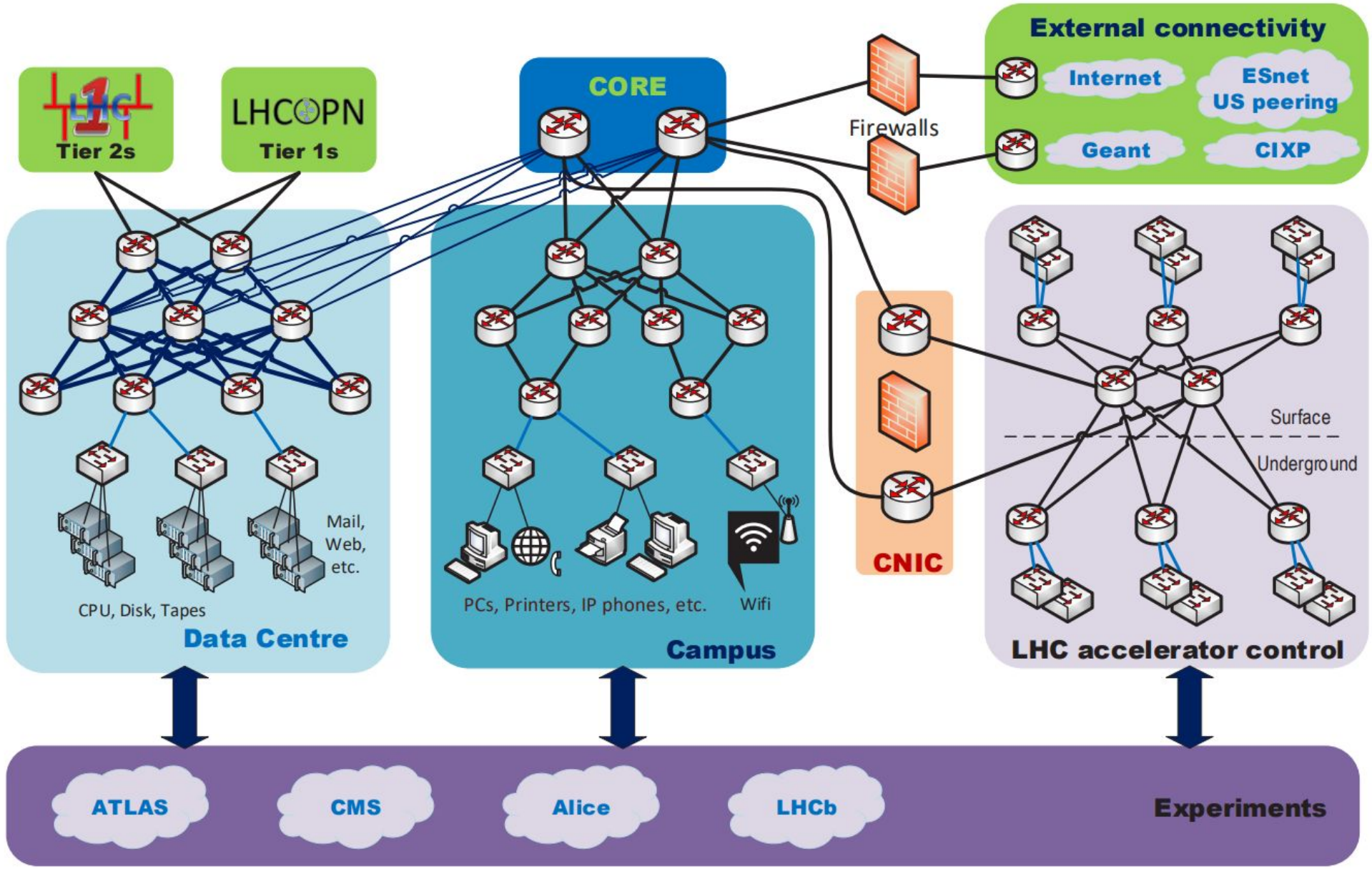
Maximum Transmit Unit (MTU) defines the maximum payload size that can be carried in a single Ethernet frame.

- Frames with more than 1500 bytes payload are considered “jumbo frames” and usually have a payload of 9000 bytes



CERN Networks

CERN Network Complex



CERN Network Complex

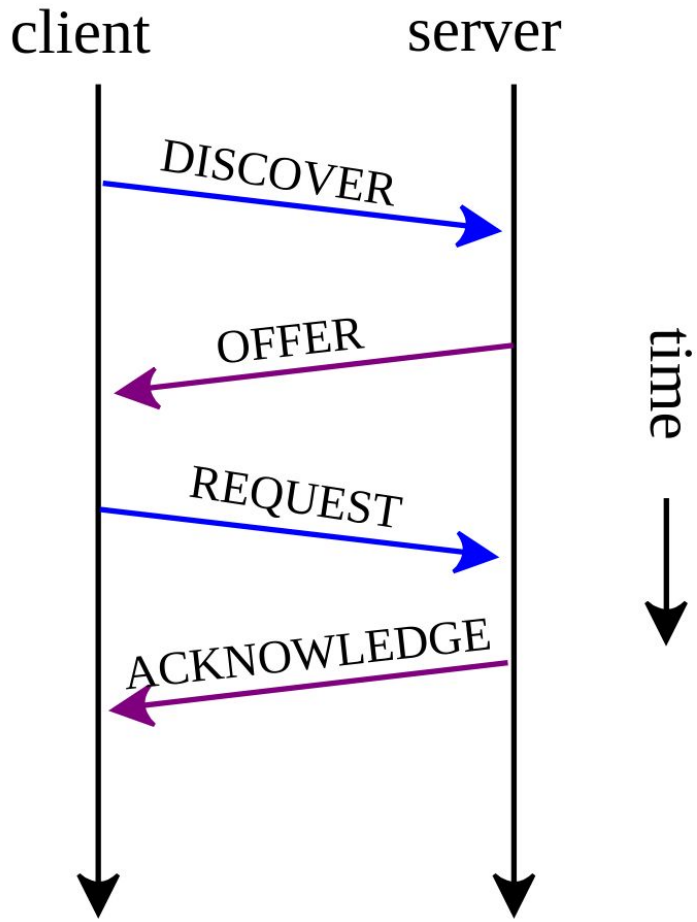
- General Purpose Network - GPN, the campus network
  - laptops, printers, phones
- IT service - ITS, servers in the data center
  - physical servers, storage and tape servers, virtual machines
- LHC Computing Grid - LCG, dedicated to experiments and WLCG, inside the data center
  - similar to ITS
- Technical Network - TN, air-gaped private network dedicates to accelerator controls
  - no internet access
  - no access from the campus or data center
  - in the data center, near the CERN control center, underground
- Experiments networks, isolated and near the corresponding detectors

Dynamic Host Configuration Protocol - DHCP

DHCP is a network protocol used to automate the configuration of devices on internet protocol (IP) networks.

- Server: issues the IP addresses and can tell the clients on its network apart.
- Client: initiates the handshake
- Relay: transport request from clients to server to scale out

DHCP is essential for Preboot eXecution Environment (PXE) to boot servers over network, as well as PCs/Laptops.



Domain Name System (DNS)

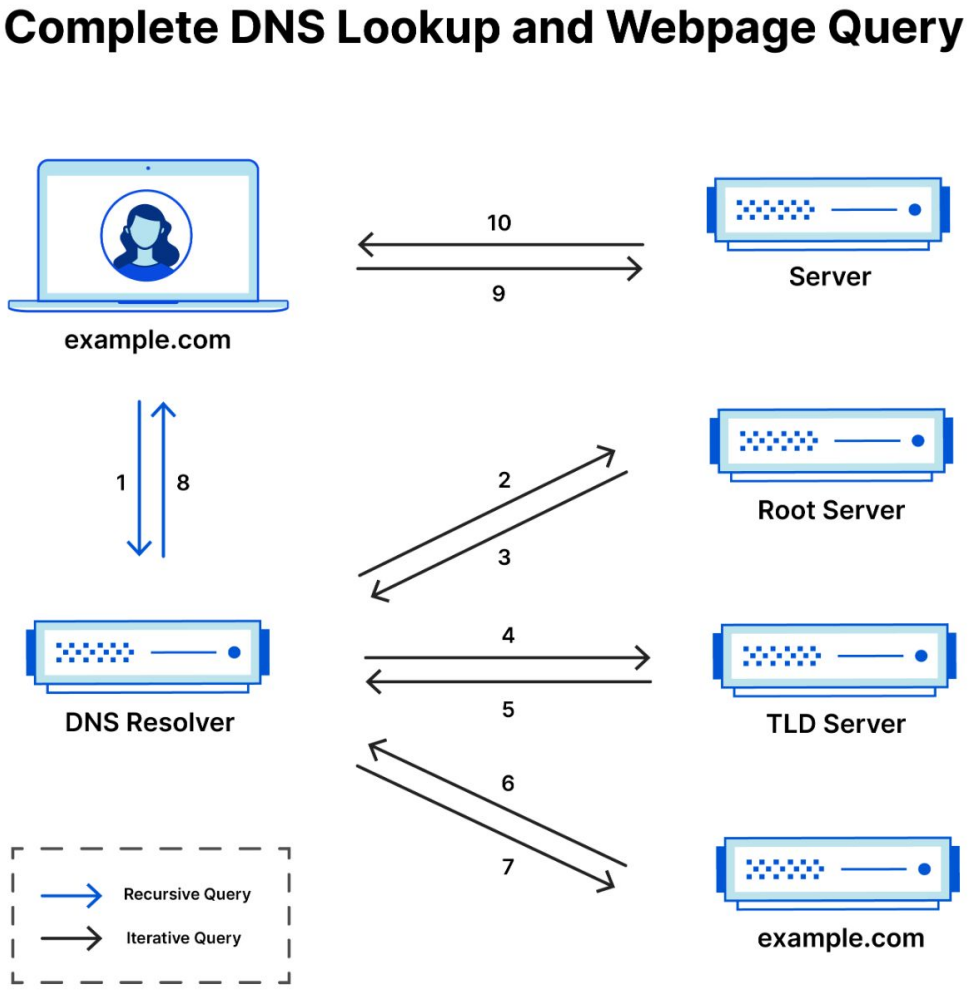
DNS is the phonebook of the Internet.

Humans access information through domain names like wikipedia.org using their web browsers.

Web browsers interact with web servers through the Internet Protocol.

DNS translates domain names into IP address.

By default, DNS is using the UDP protocol on port 53. Variants over TCP and HTTP also exist.

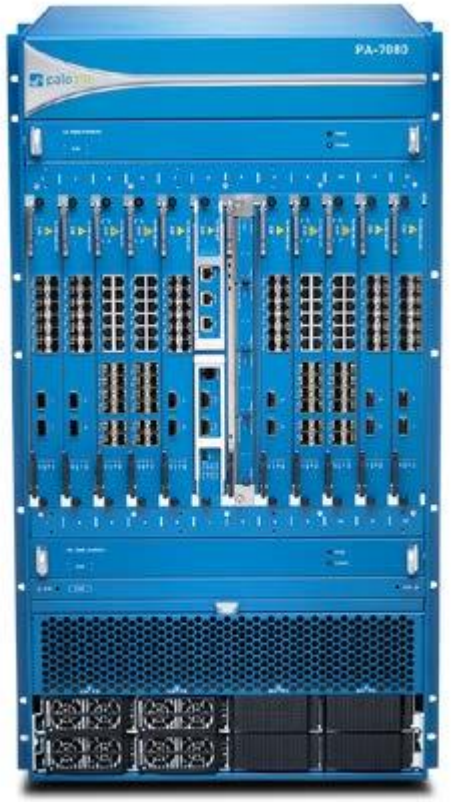


Perimeter Firewall

A firewall is a network security system that monitors and controls incomin and outgoing network traffic based on predetermined security rules.

- Packet filter, packets filtered by source and destination IP addresses, protocol, or source and destination ports
- Application layer, HTTP, DNS, FTP, etc traffic can be inspected and filtered

All devices can be accessed outside the CERN work, opening can be made on demand.



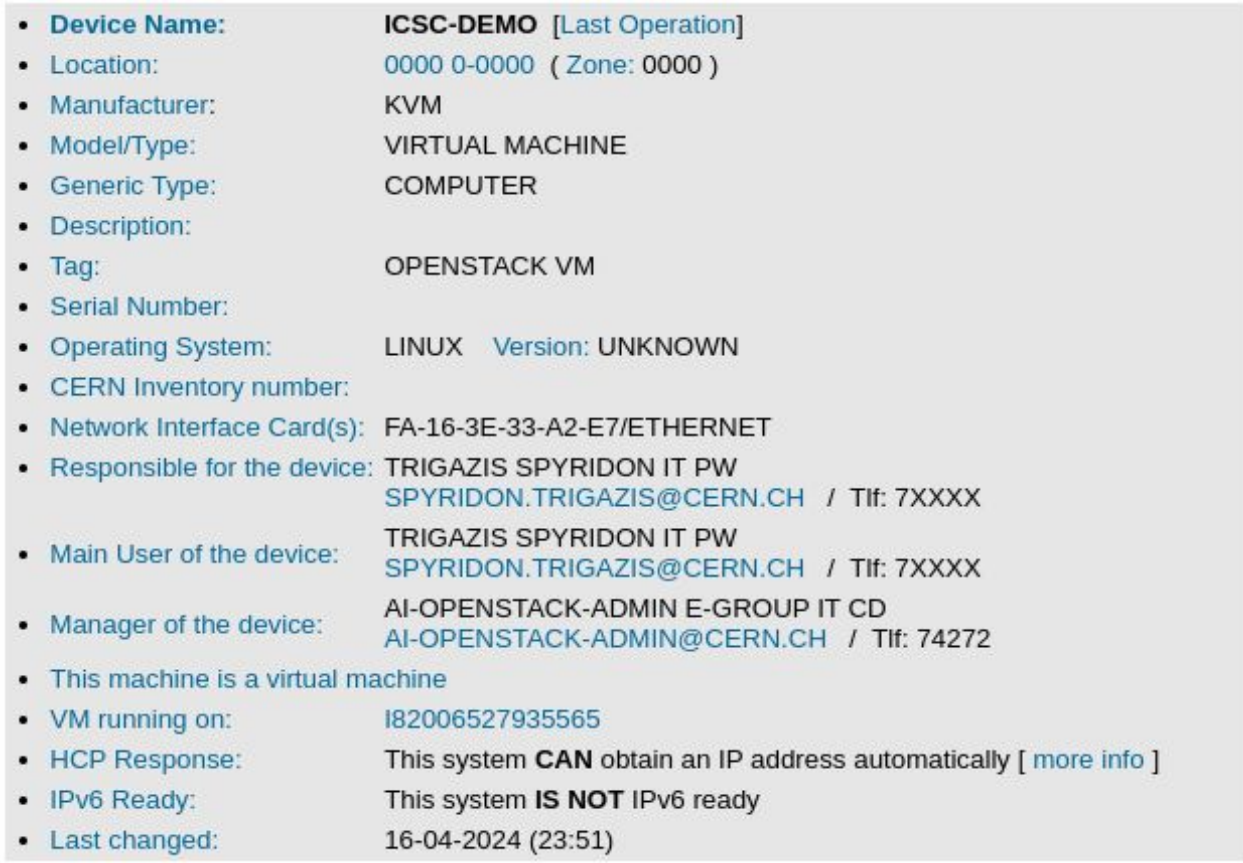
Network Database - LanDB

To track devices, their configuration and ownership, the networking group is maintain a network database.

Devices have:

- user
- responsible
- model, manufacturer, etc
- interfaces
- membership in landb sets

Landb sets, are entries in the database with sets of devices. Many other systems at CERN consume these sets to apply application logic, like internal firewall rules, DHCP, Certificates, etc



- **Device Name:** **ICSC-DEMO** [Last Operation]
- Location: 0000 0-0000 (Zone: 0000)
- Manufacturer: KVM
- Model/Type: VIRTUAL MACHINE
- Generic Type: COMPUTER
- Description:
- Tag: OPENSTACK VM
- Serial Number:
- Operating System: LINUX Version: UNKNOWN
- CERN Inventory number:
- Network Interface Card(s): FA-16-3E-33-A2-E7/ETHERNET
- Responsible for the device: TRIGAZIS SPYRIDON IT PW SPYRIDON.TRIGAZIS@CERN.CH / Tlf: 7XXXX
- Main User of the device: TRIGAZIS SPYRIDON IT PW SPYRIDON.TRIGAZIS@CERN.CH / Tlf: 7XXXX
- Manager of the device: AI-OPENSTACK-ADMIN E-GROUP IT CD AI-OPENSTACK-ADMIN@CERN.CH / Tlf: 74272
- This machine is a virtual machine
- VM running on: I82006527935565
- HCP Response: This system **CAN** obtain an IP address automatically [more info]
- IPv6 Ready: This system **IS NOT** IPv6 ready
- Last changed: 16-04-2024 (23:51)

Cloud Networking

What is cloud computing?

Cloud computing is the act of running workloads within clouds—which are IT environments that abstract, pool, and share scalable resources across a network. Neither cloud computing nor clouds are technologies themselves.

- Cloud computing is an act - the function of running a workload in a cloud.
- Clouds are environments - places where applications run.
- Technologies are things - software and hardware used to build and use clouds.

Cloud environments offer on-demand computing resources like:

- compute - physical or virtual servers
- **network - private and public networks, virtual IPs, firewalls, load balancers…**
- storage - block storage, filesystems, object storage, archives…
- services - websites, functions, container infrastructure, databases…

Virtual machines

A virtual machine (VM) is a software-based emulation of a physical computer system. It operates like an independent computer within a host server or a physical computer, running its own operating system (OS) and applications as if it were a separate physical machine.

- Hypervisor: A hypervisor is physical server which runs software that creates and manages virtual machines. It sits between the physical hardware and the virtual machines, allocating hardware resources to each VM and managing their execution.

- Guest Operating System: Each virtual machine runs its own guest operating system, which can be different from the host operating system.

Private and Provider networks

A private network is a computer network that uses a private address space of IP addresses. These addresses are commonly used for local area networks (LANs) in residential and corporate environments. Both the IPv4 and the IPv6 specifications define private IP address ranges.

In a cloud environment, private networks provide connectivity within a project or application.

Provider network, sometimes referred to as external or public network provides connectivity to the internet or a larger corporate network.

Floating and Virtual IPs

A virtual IP address (VIP) is an IP address that does not correspond to a physical network interface. In many cases floating IP (FIP) and virtual IP are used interchangeably. But in some cases, a FIP is a public and a VIP is private.

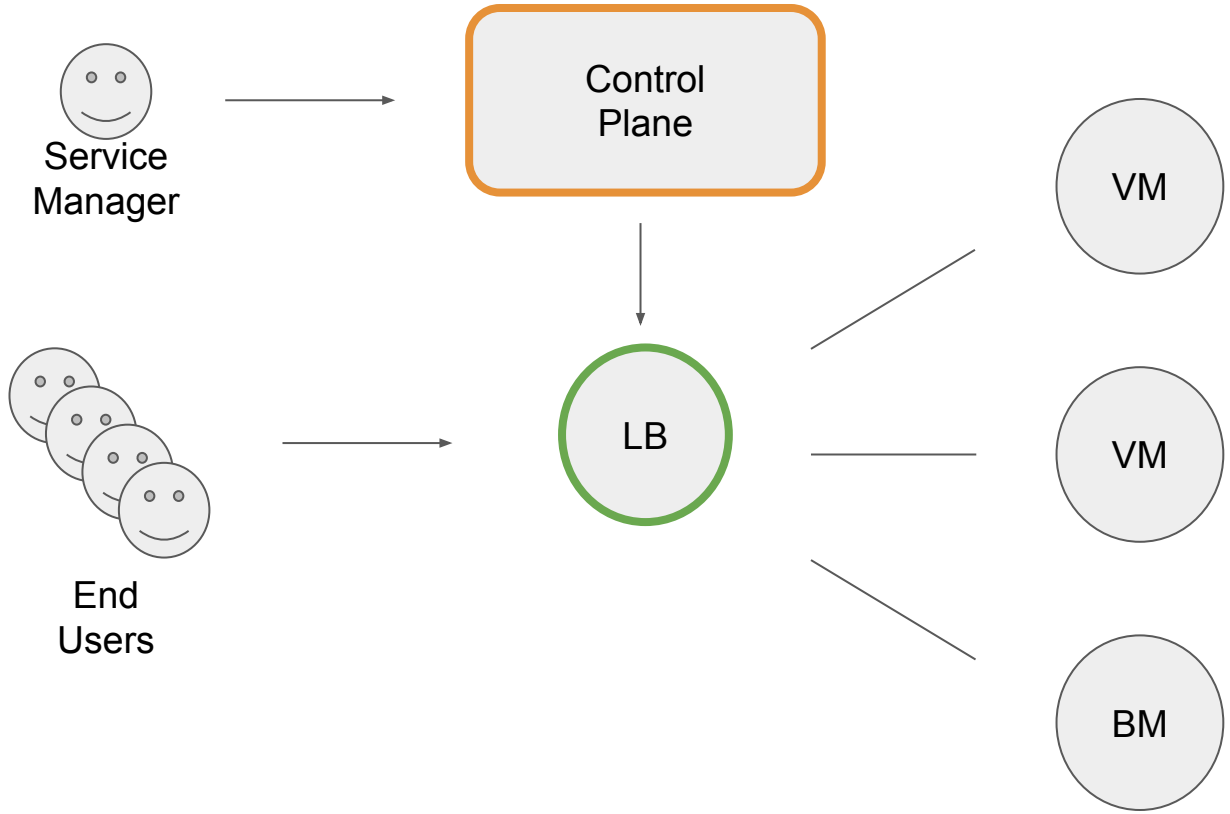
Use cases:

- Provider network connectivity
- IP mobility, move the IP from interface to interface
- Load balancing

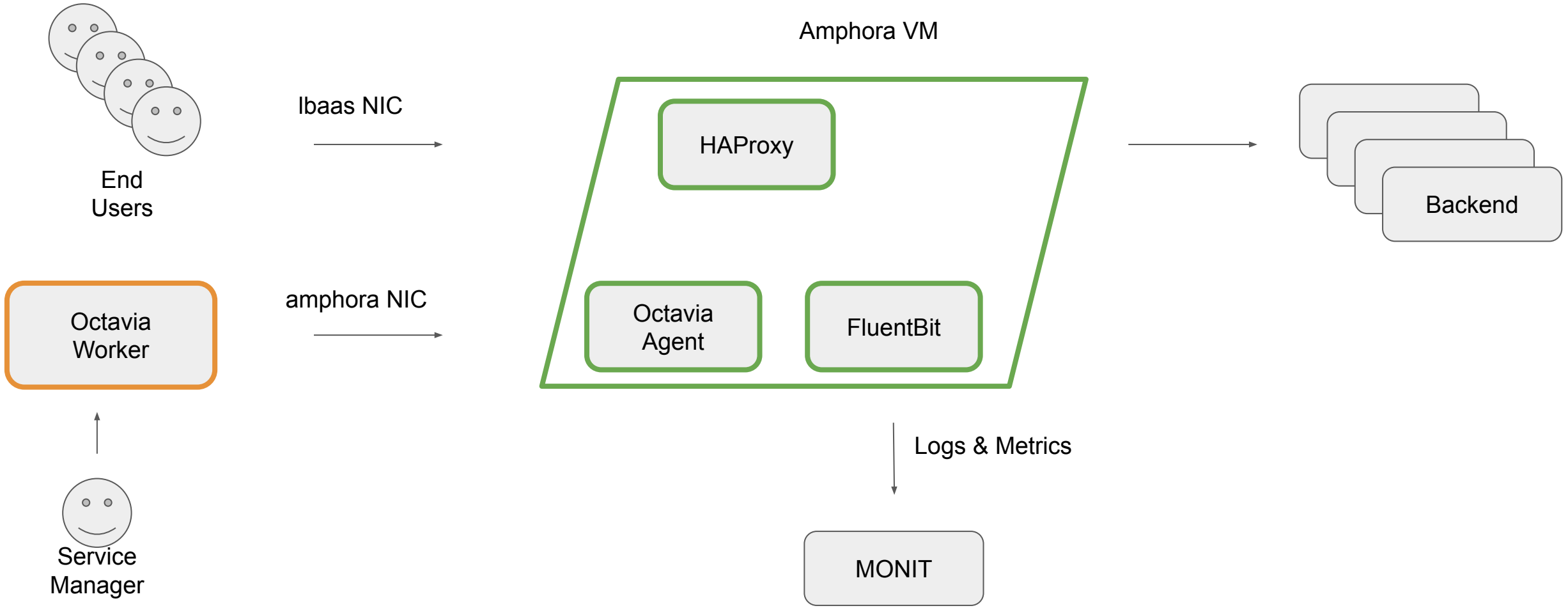
Load Balancing as a Service

REST API service to provision and configure load balancers.

- HAproxy
  - Fine tuning for applications, especially HTTP
- L3/L3 load balancer
  - native network load balancing
  - no SNAT



VM-based Load Balancers

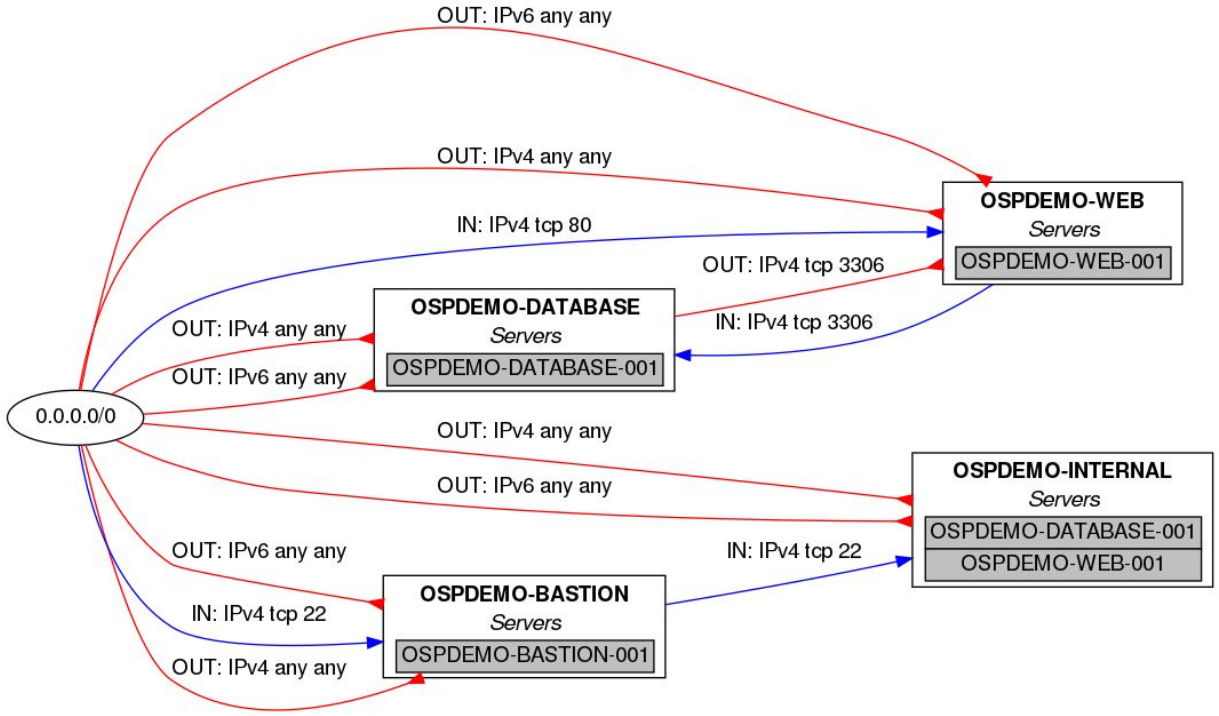


Security Groups

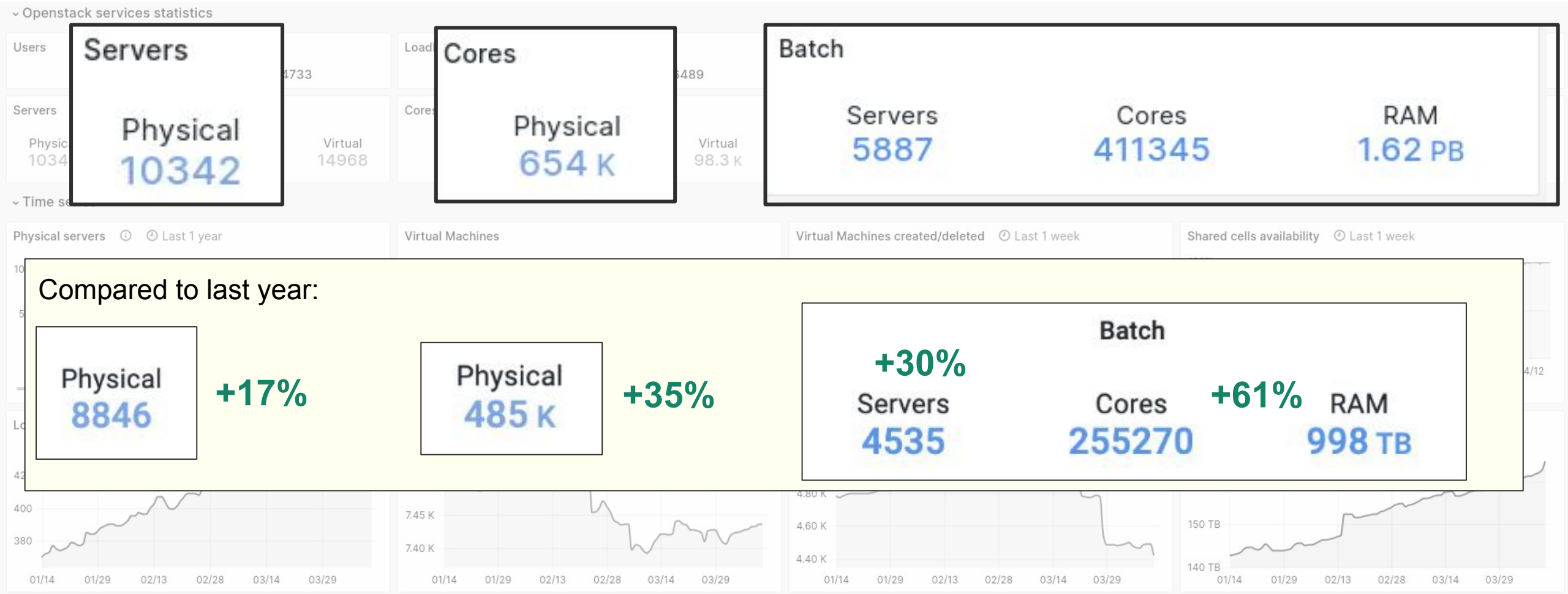
Security groups provide a container for virtual firewall rules that control **ingress** (inbound to instances) and **egress** (outbound from instances) network traffic at the port level.

Security groups use a default deny policy and only contain rules that allow specific traffic. Each port can reference one or more security groups in an additive fashion.

The firewall driver translates security group rules to a configuration for the underlying packet filtering technology such as iptables/nftables or a plugin-specific implementation.

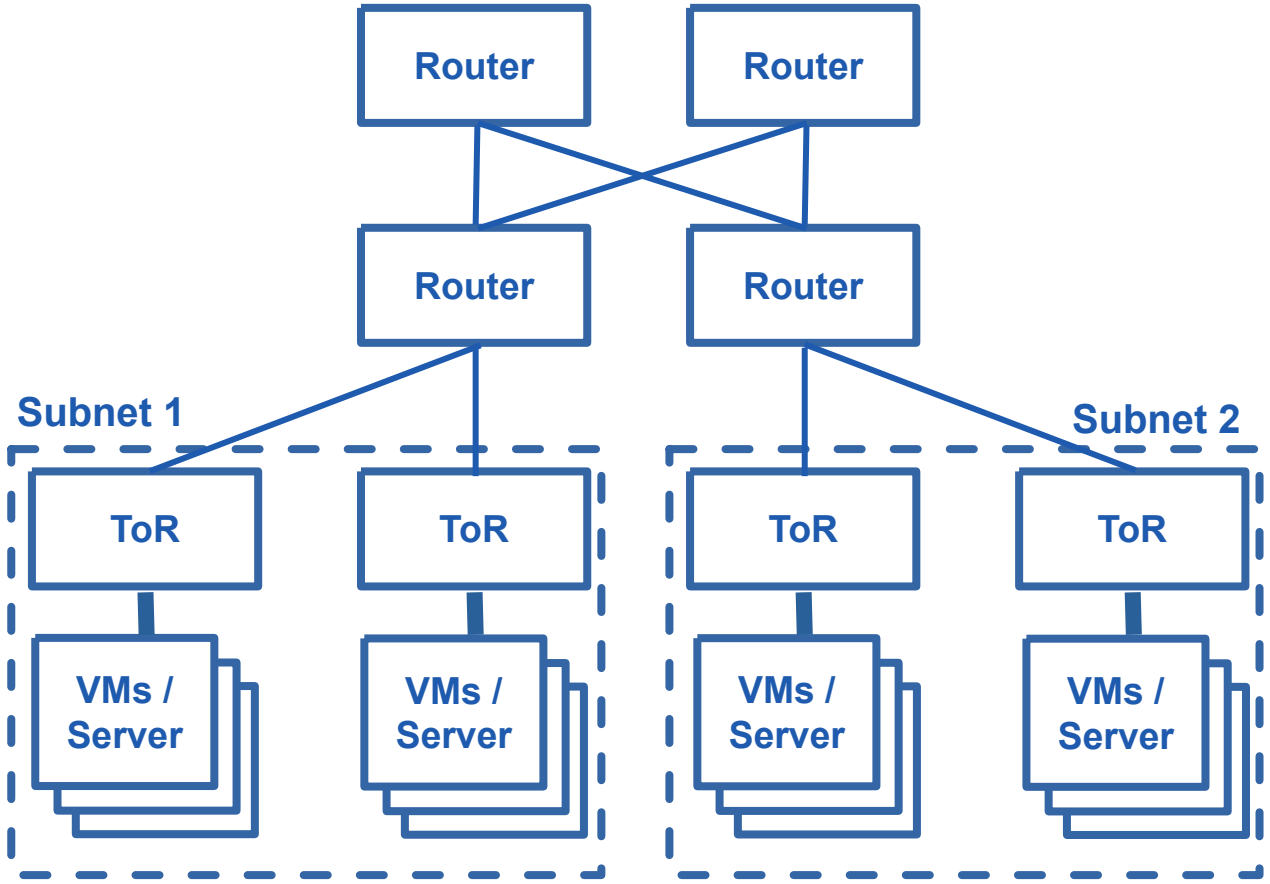


CERN Cloud Overview



Meyrin Data Center - MDC

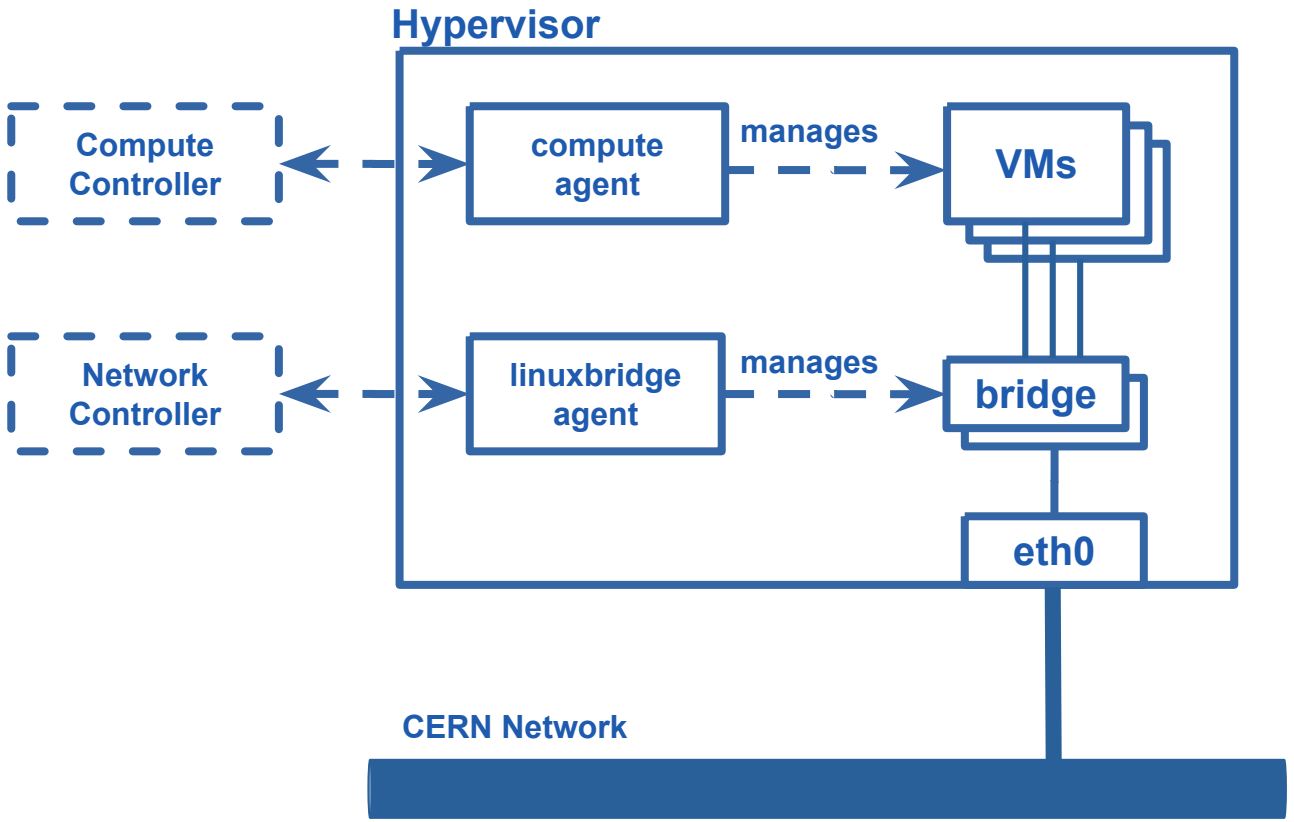
- L2 network segmentation
- multiple broadcast domains
- VMs connect via LinuxBridge
- All VMs in the same public network
    - Full Dual-Stack IPv4 / IPv6
- Mantra: “Everything in same network”
    - except e.g. tech./control networks



Meyrin Data Center

Hypervisor

- Libvirtd and compute controller
- linuxbridge agent creates tap devices for VMs
- firewall rule to access the metadata service

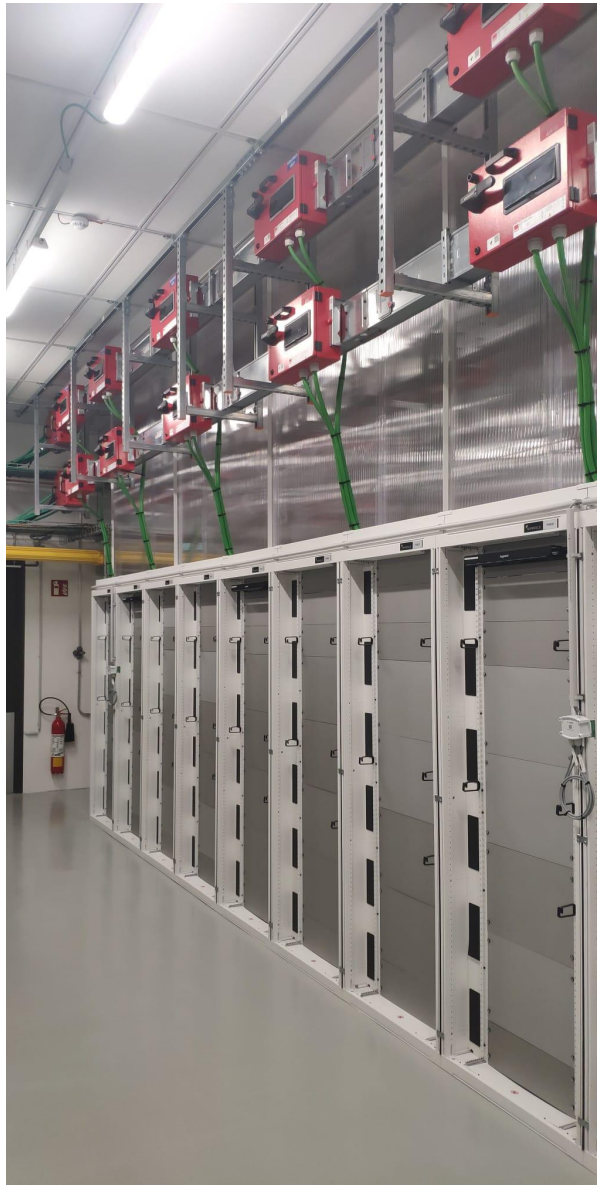


New requirements / Old Limitations

- Add Ability to migrate VMs between Hypervisor in different subnets
- New hypervisors can easily host over 100 VMs
    - some performance issues seen in current setup with higher VM count
- LinuxBridge implementation upstream marked experimental / unsupported
- Different Teams ask for
    - Private Networks
    - Security Groups (Firewall rules on HV level)
    - Floatings IPs

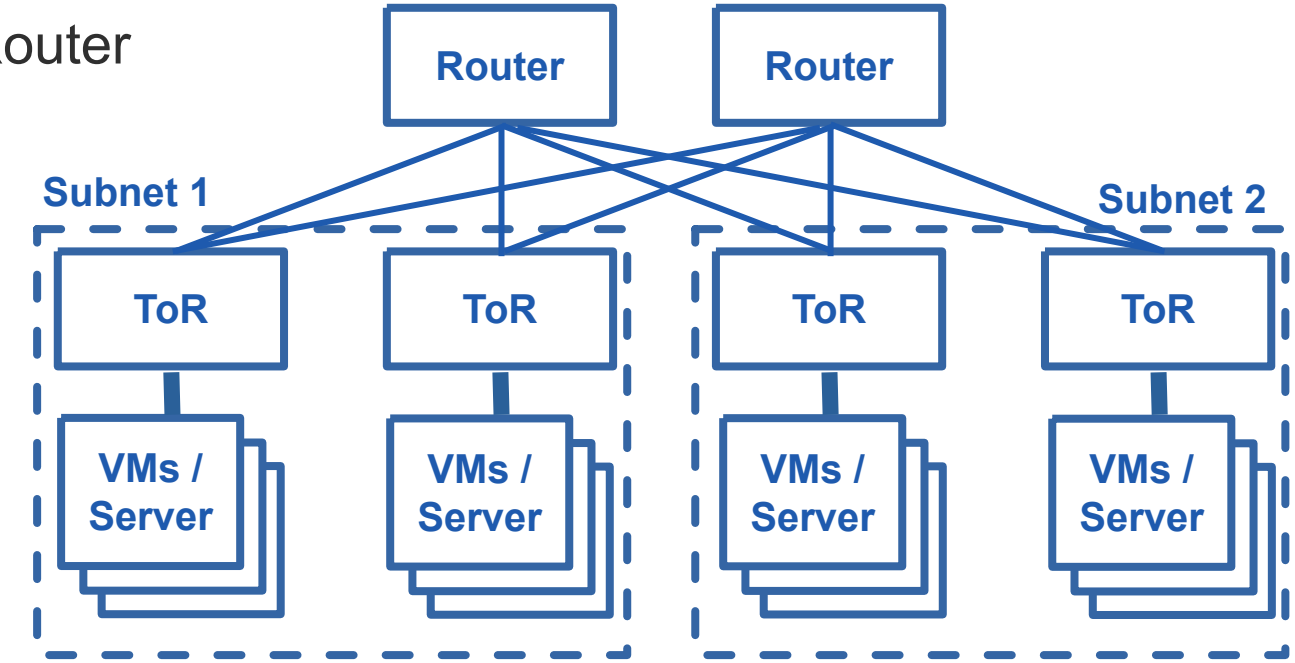
Prevessin Data Center

Prevessin Data Center



Prevessin Data Center - PDC

- For better scalability, network is divided into multiple subnets with up to 1000 IPv4s (for now) + IPv6
- ToR switch only L2
  - passing traffic outside subnet to Router
- Very similar architecture to other DC



Different network types in PDC

- **Public/Provider** network (from start)
  - Keep existing functionality to simplify on-boarding
  - one subnet per 16 servers (approx 1000 IPv4 + IPv6)

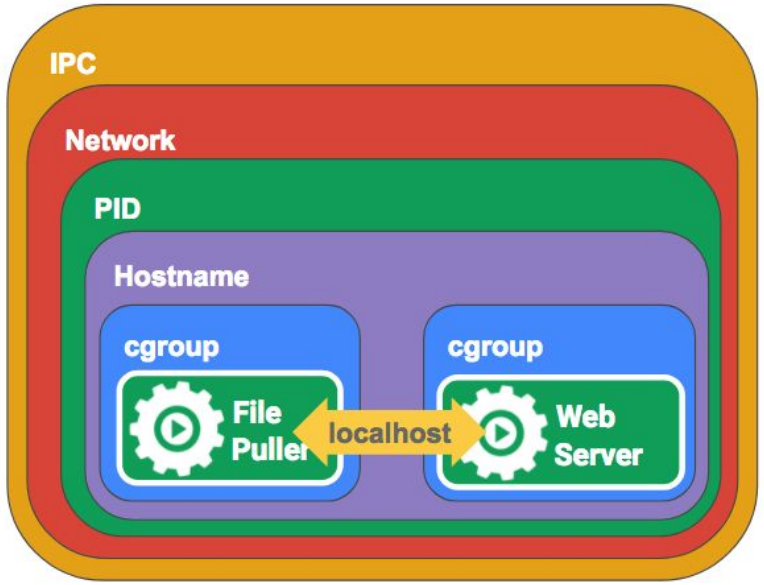
- **Private** networks (Q2/2024)
  - Overlay network with Open Virtual Network (OVN)
  - Geneve tunnel between hypervisors

Kubernetes Networking

Basic Concepts

- Container - regular process created by using two Linux Kernel features - namespaces and cgroups
  - Namespaces, process isolation - IPC, Network, PID, UTS, Users and Mount
  - Cgroups, manage/share resources - CPU, Memory

- Pod - one or more containers sharing:
  - network namespace, i.e. same IPs, localhost
  - users
  - storage, external volumes and shares directories



Kubernetes Network Concepts

Pods on a node can communicate with all pods on all nodes without NAT.

Agents on a node (e.g. system daemons, kubelet) can communicate with all pods on that node.

Services and internal DNS are used for discovery and load balancing.

Ingress and Gateway API are used for advanced traffic routing.

Network Policies are used to restrict access within the cluster and secure nodes.

Container Network Interface (CNI)

Kubernetes uses a pluggable support for network “drivers” via the Container Network Interface (CNI)

CNI plugin requirements:

- Expected to use no NAT for communication between the pods
- All nodes should communicate with all containers without NAT (and vice versa)
- IP that a container sees itself is the same IP that others see it as

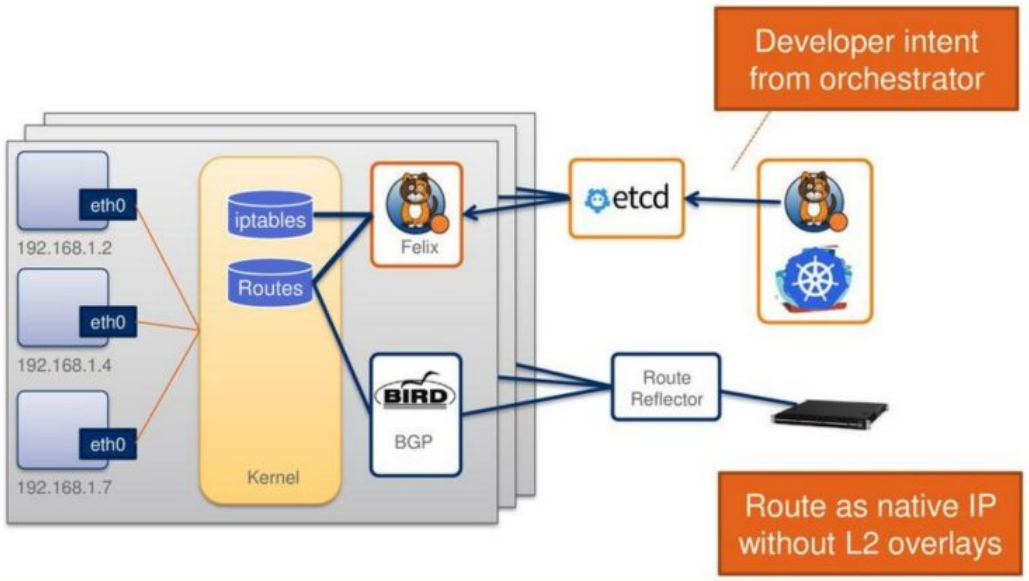
There are many existing CNI implementations, which are part of the Cloud-Native Networking landscape.

Calico

IP routed fabric that can work both on top layer-2 or layer-3 only networks and uses BGP peering for full-mesh between nodes and advertise routes to network appliances.

Four basic components:

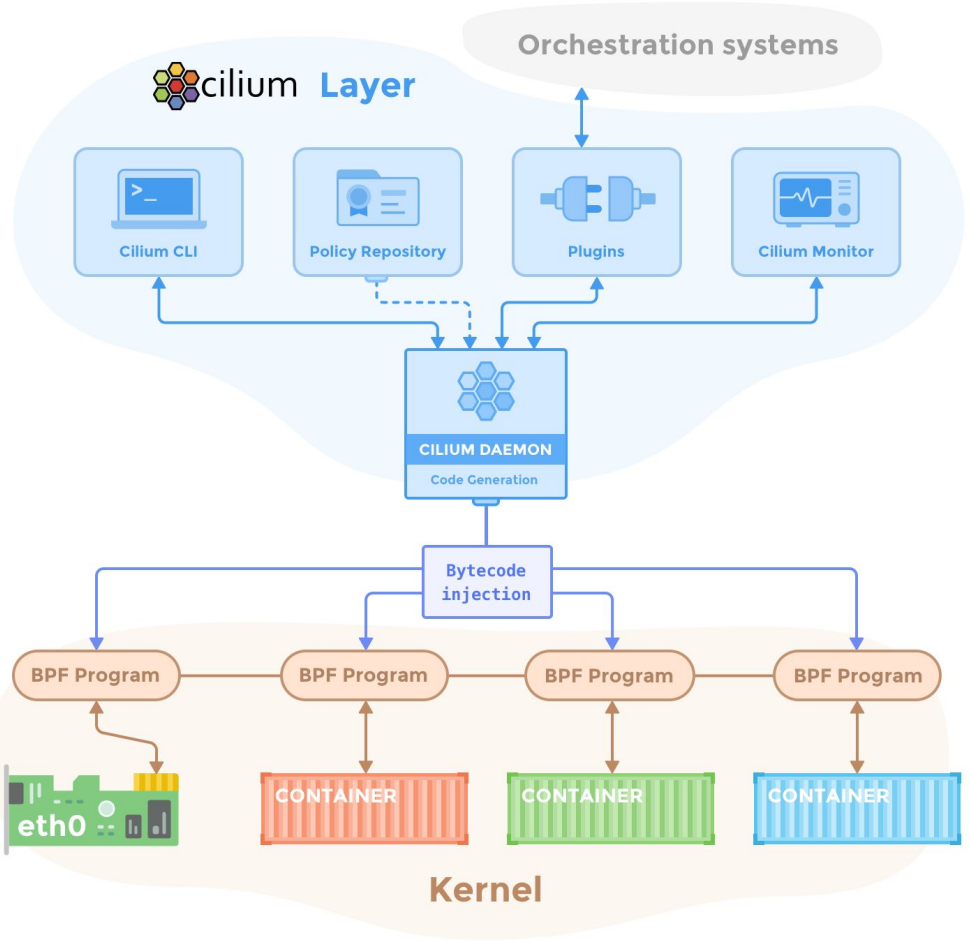
- Orchestrator plugin (CNI)
- Data store (etcd or k8s API)
- Felix - agent that runs on each node
- BIRD - BGP client that distributes route



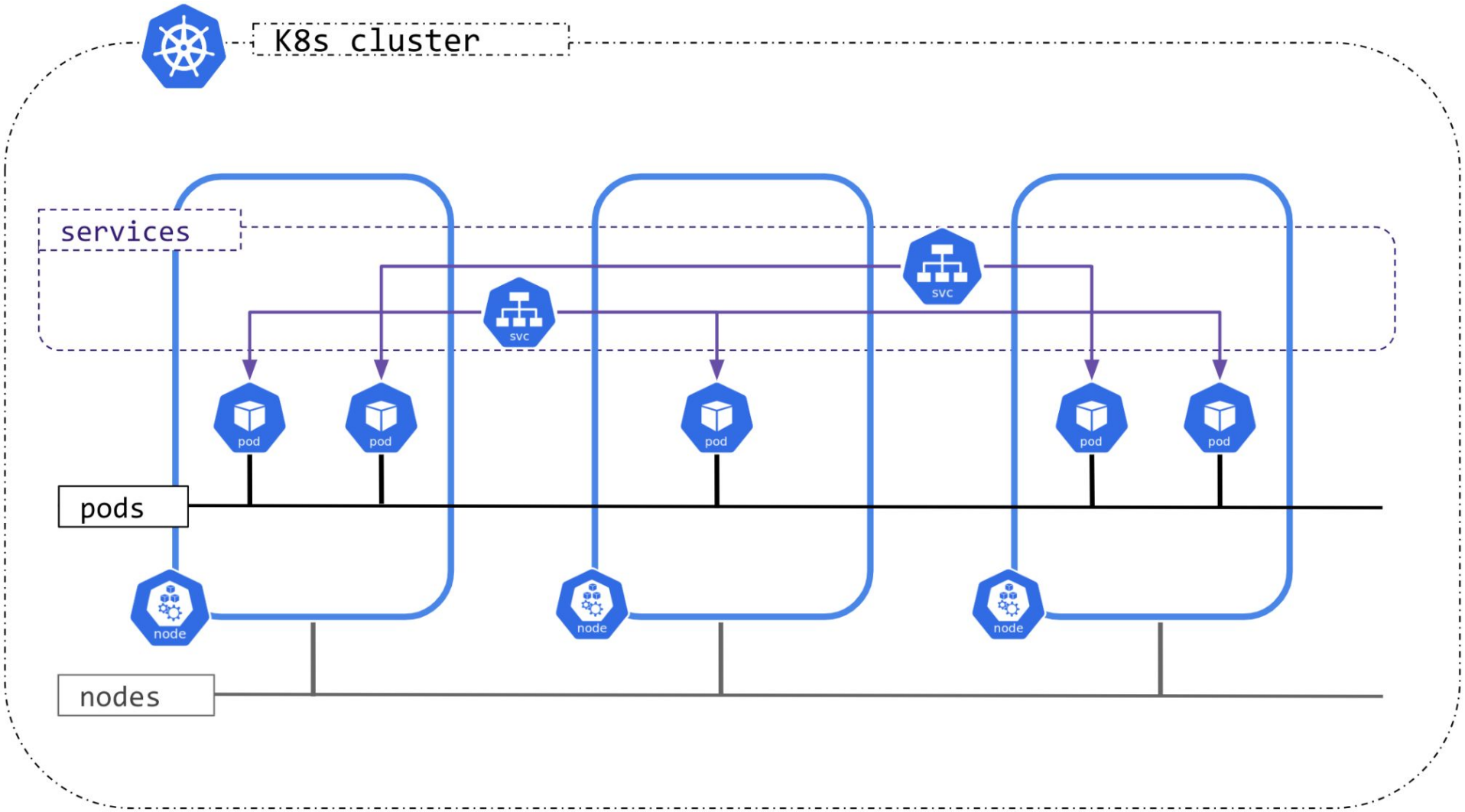
Cilium

Cilium operates at Layer 3/4 for networking and Layer 7 for applications.

- Uses vxlan or geneve for tunneling
- Uses eBPF programs
- Service LB with eBPF instead of kube-proxy
- Network and host policies
- Data encryption with IPSec or Wireguard
- Observability with hubble
- Cluster Mesh
- Cluster gateways



Standard layout of a cluster

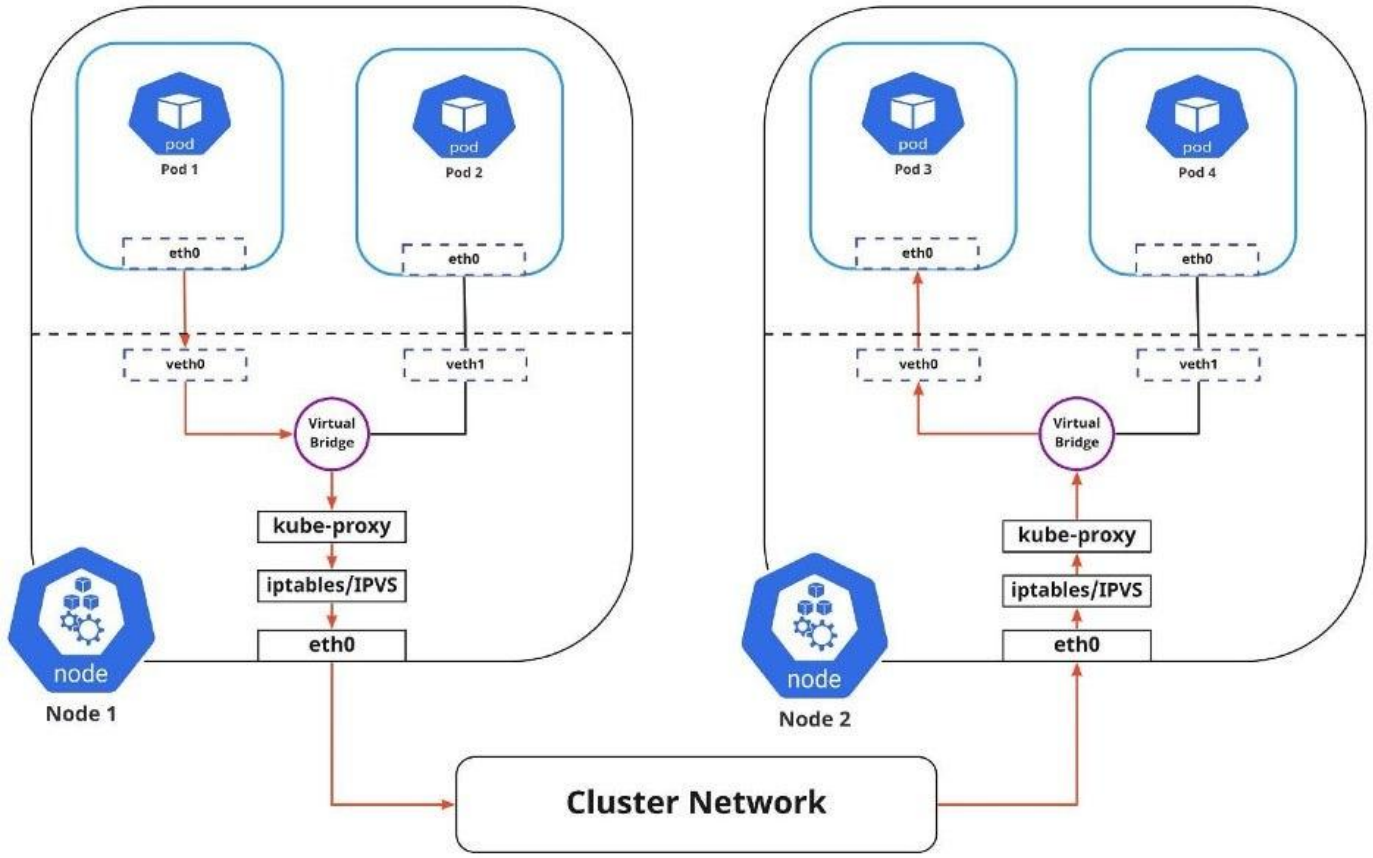


Kubernetes services

Services are used for discovery and load balancing. For example, for memcached in “prod” namespace can be reached at *memcached.prod* from any pod.

Standard implementation of services is done in kube-proxy with iptables/nftables or ipvs, both part of the Linux Kernel.

Network plugins like cilium can replace kube-proxy.

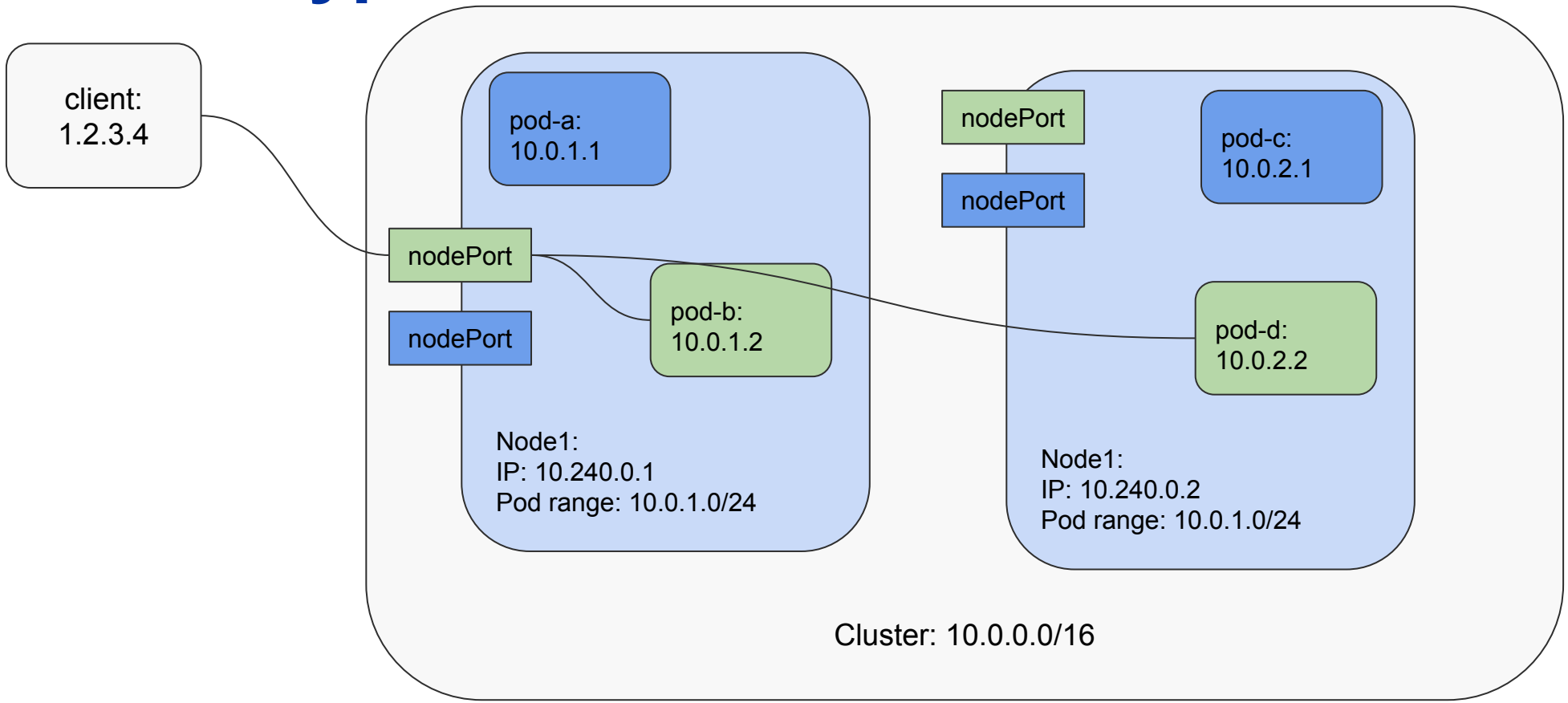


Getting traffic into the cluster

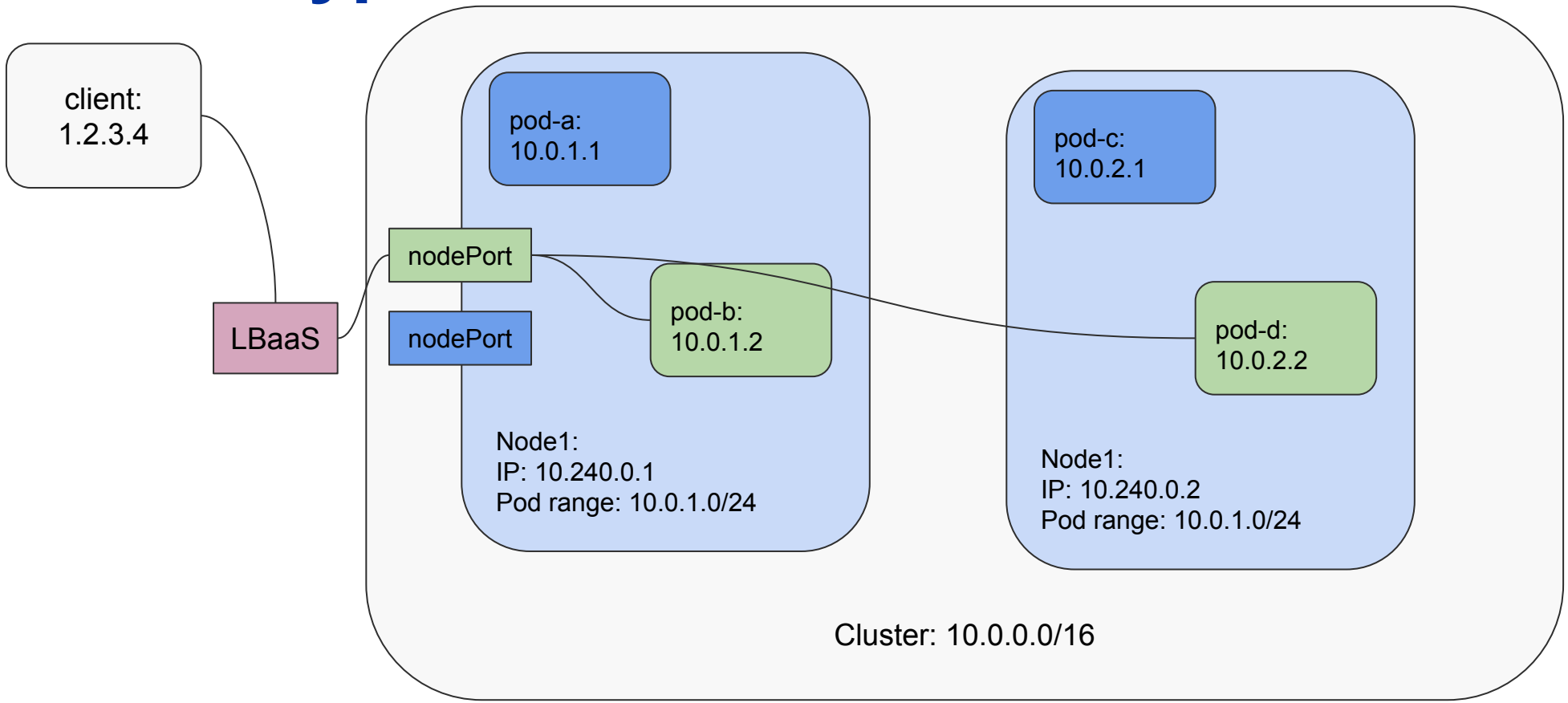
One of the hardest tasks to achieve on custom clusters.

1. serviceType: nodePort, all nodes accept traffic on high number ports and forward it to pods
2. pod hostPort, specific nodes are not running in the overlay and expose service port (eg 443)
3. serviceType: loadBalancer, a cloud-provider or with network appliance integration, an external load balancer is provisioned
4. Ingress
   a. nodePort or hostPort, same as 1 and 2 but with an internal nginx/haproxy/etc in the middle
   b. with loadBalancer, same as 3 with an internal nginx/haproxy/etc in the middle

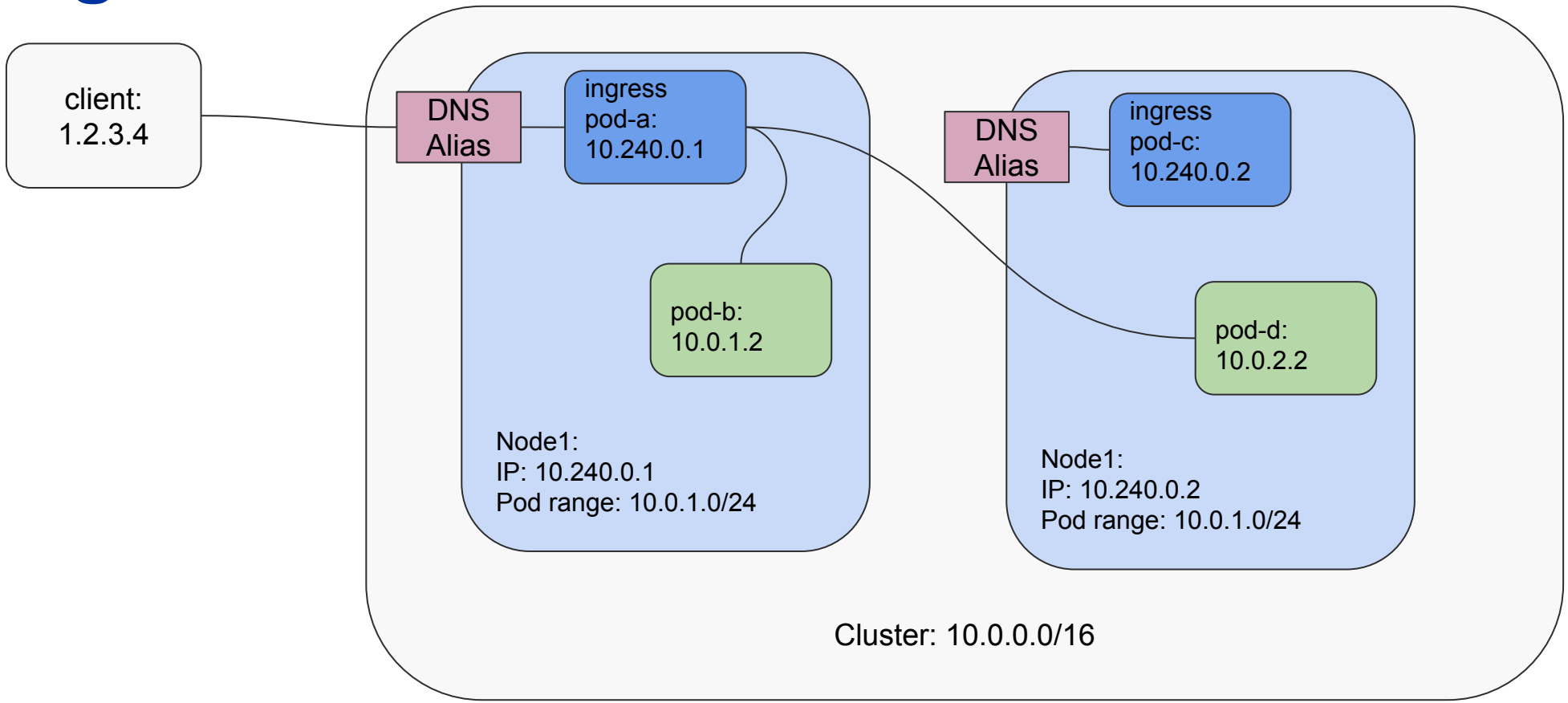
Getting traffic into the cluster with serviceType: nodePort



Getting traffic into the cluster with serviceType: loadBalancer



Getting traffic into the cluster with ingress

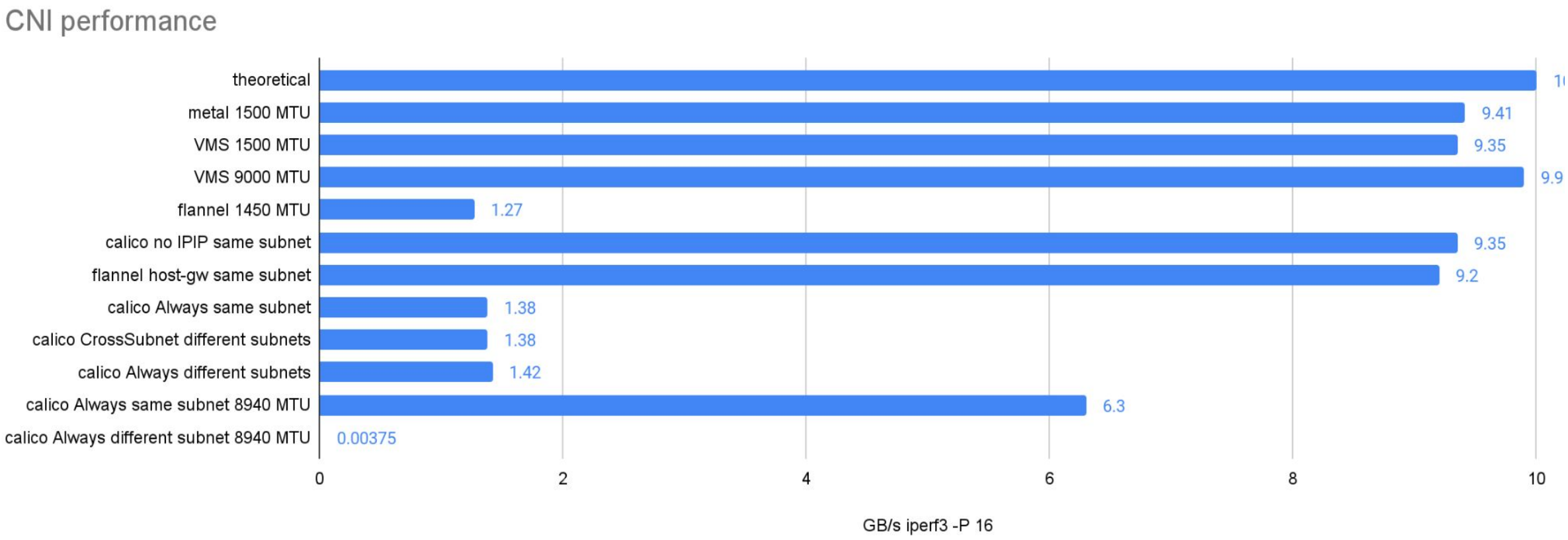


Performance considerations - specs

Motivation: Use cases like accessing storage systems (eos, cephfs) or DBs

- iperf tcp test between two nodes only
- Compare 10Gbit, baremetal, virtual machines, pods
- Baremetal nodes connected on the same switch
- Virtual Machines IP in same or different subnet
- Flannel with host-gw or vxlan
- Calico with no encapsulation and IP-in-IP
- Different MTU configurations

Performance considerations results



Spreading clusters across availability zones

Users may want to spread workloads and data across failures domains. In CERN's case, the failure domains are availability zones in the cloud, physical machines in other buildings, the new Prevessin Data Centre.

One cluster spanning as many availability zones as possible with a single overlay network, it's the easiest to configure with no additional complexity.

One cluster per availability zone offers isolation on rolling out new features/ configurations/ upgrades by reducing the blast radius of miss-configurations.

- Applications should be able to operate without east-west traffic
- Leverage a cluster-mesh solution taking care of scheduling workloads on separate clusters

Spreading clusters across clouds

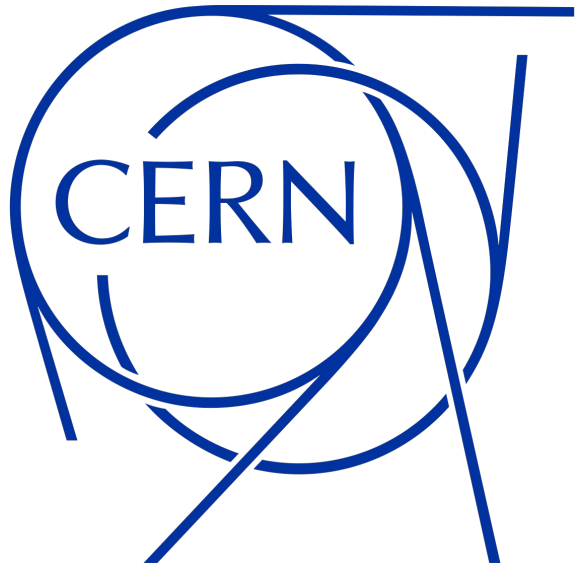
For maximum redundancy and to expand/bust capacity user can deploy cluster across clouds, private or public.

For applications that do not require east-west traffic, the setup is similar to clusters/availability zone.

For applications that require east-west traffic:

One cluster-mesh across clusters, across clouds by adding all nodes in a VPN.

By deploying one or more gateway nodes on each cloud.


CERN