Deploying and running services via OpenStack and Kubernetes at Chalmers e-Commons

Network, Hardware, Storage and Software setup

CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

We operate Data Center(s) and provide Services to Researchers

- eInfra Group at Chalmers e-Commons

Presentation overview

Introducing Infrastructure:

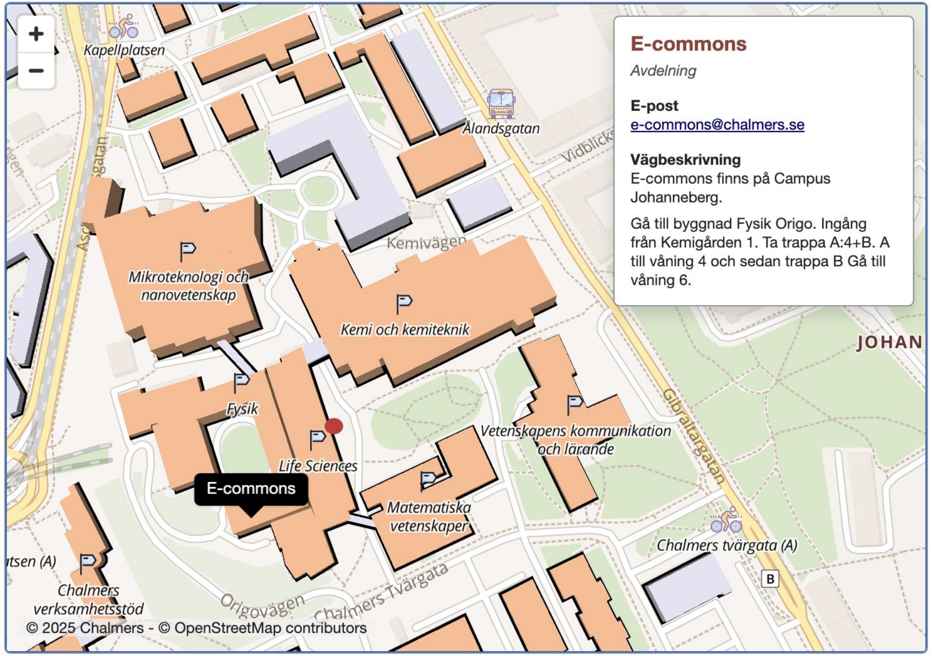
- Data Center
- Network resources
- Computing resources
- Storage resources

Introducing Services:

- Ceph Cluster
- HPC Cluster
- Openstack Cluster
- Kubernetes Cluster

Discussion and follow ups

Data Center(s) Overview

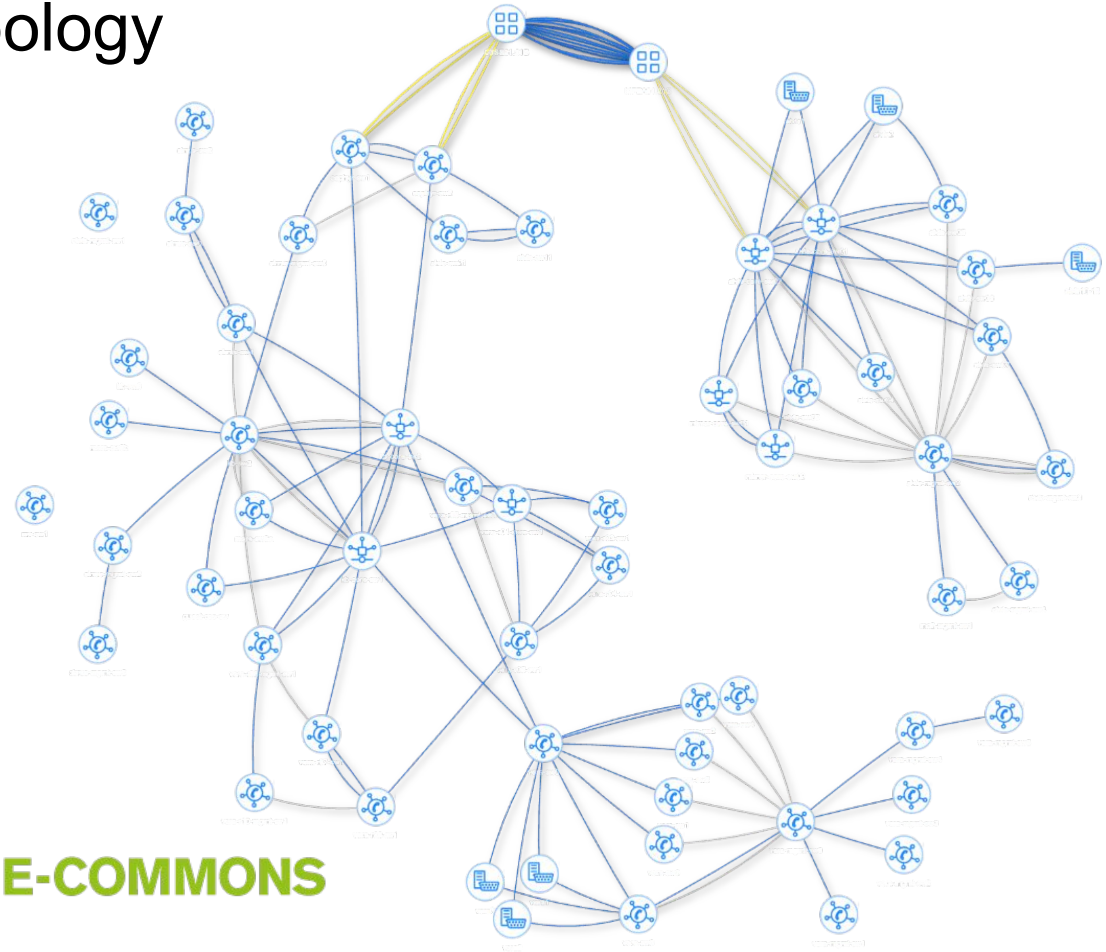


- MC2 (Alvis)
- KB (Vera - Cloud)
- MV (extra)
- HPC2N (backup)

- 24/7
- power, dual-power, UPS
- cooling and environmental controls
- floor space and cable management
- physical security - access controls
- redundancy and disaster recovery

CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

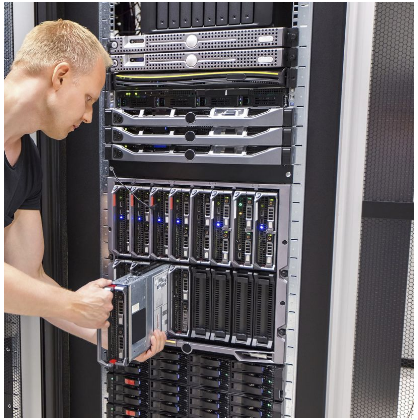
Network Topology



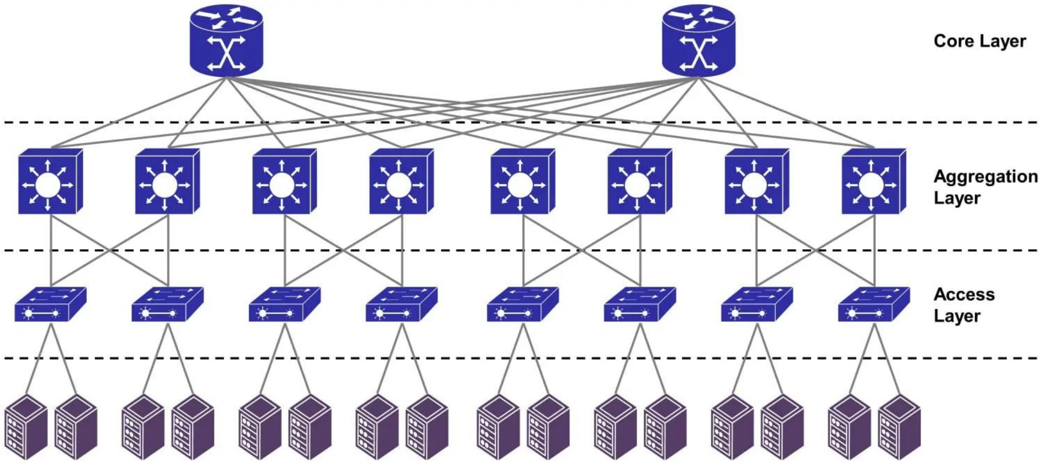
CHALMERS
UNIVERSITY OF TECHNOLOGY
E-COMMONS

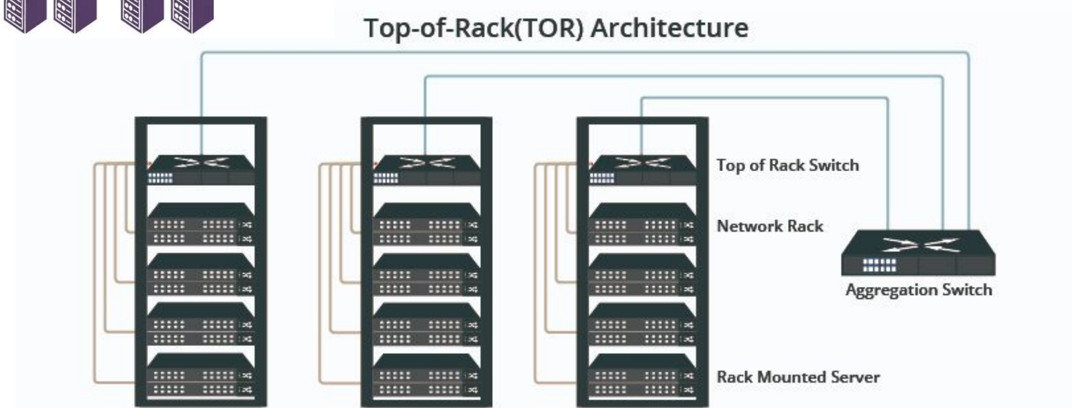
Data Center Servers

- Servers (Compute Resource)
  - Rack-mount servers

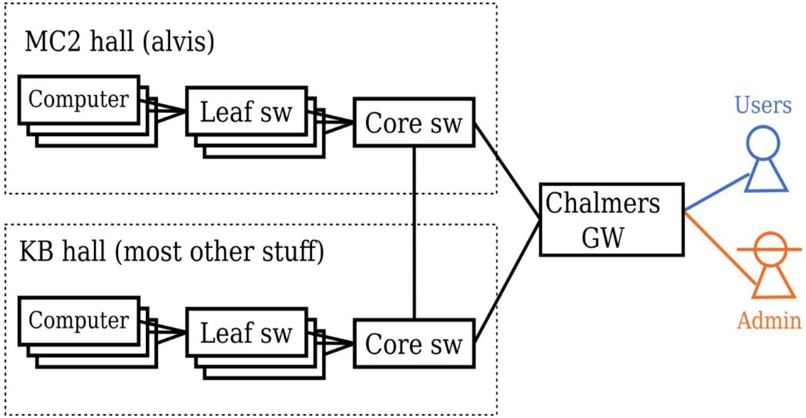


3-Tier Data Center Network Architecture

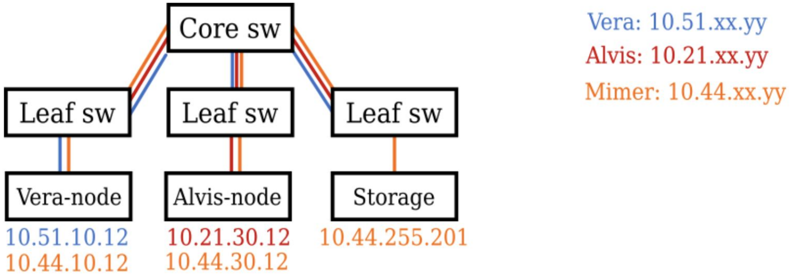




Data Center(s) networking



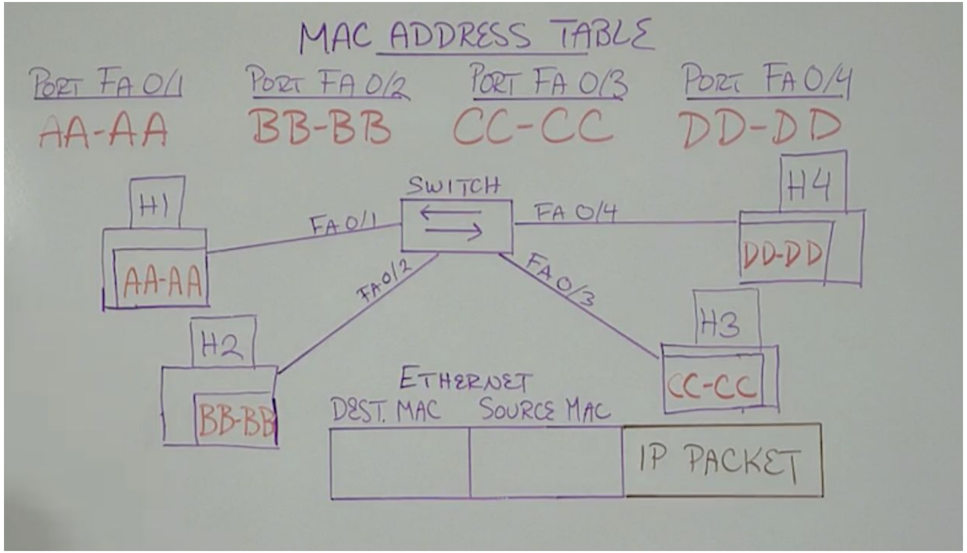
- Layer 1/2/3 networking;
- We have rather simple setup (mostly layer 2);

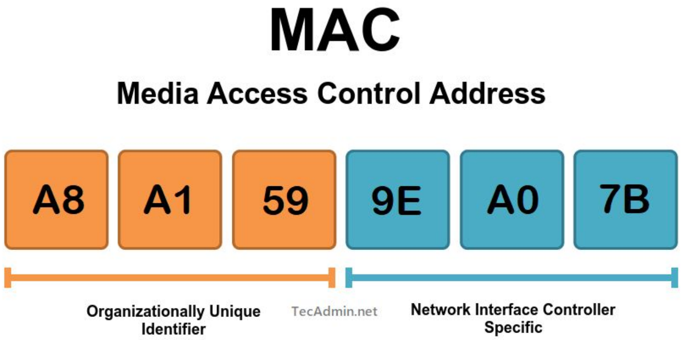


- generally no router – only level-2 switches;
- separation with VLAN;
- convention: consistent naming and IP assignment;

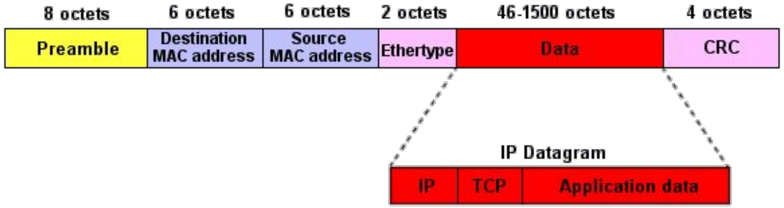
Credit: **Yunqi**

Layer 2: Data Link Layer

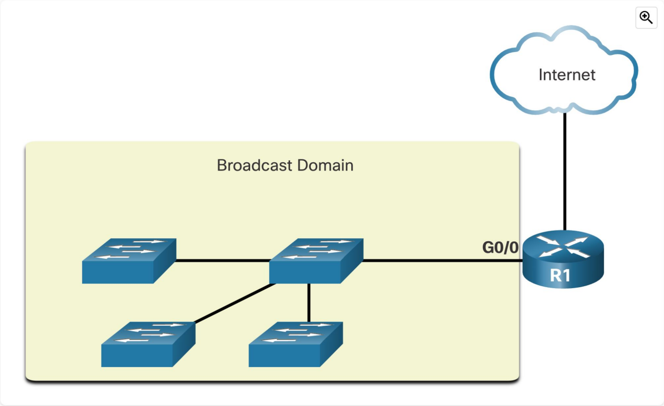


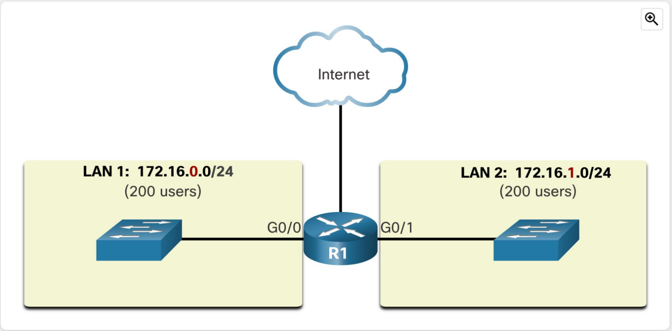


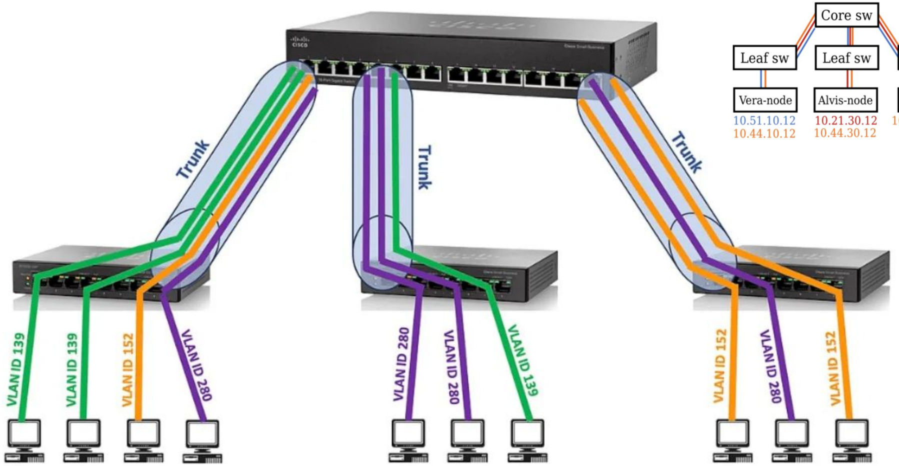
- **Ethernet** is the protocol of choice in LAN
- Each **Ethernet NIC** has **MAC address**
- **Ethernet Frame** has source and destination MAC address
- **Ethernet Switch** operates on L2 level
-

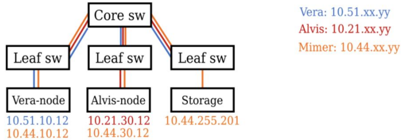


VLANs

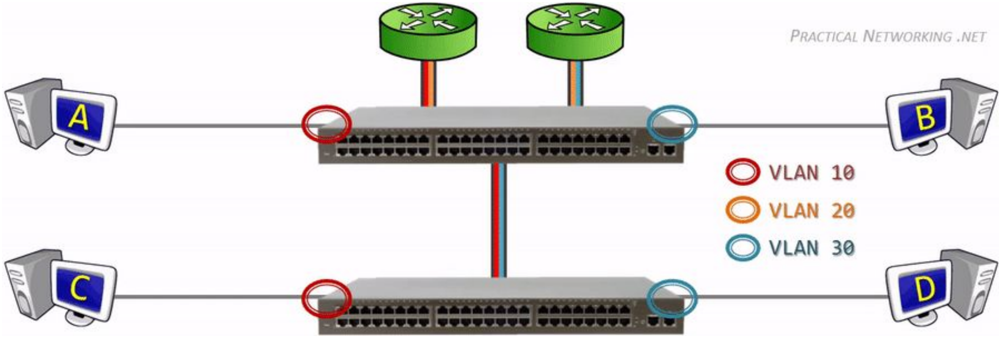








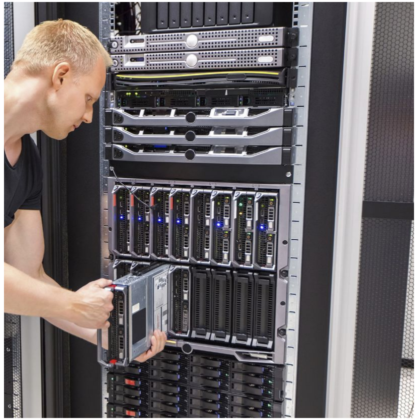
Traffic arriving on a switch port assigned to one VLAN will only ever be forwarded out another switch port that belongs to the same VLAN – **a switch will never allow traffic to cross a VLAN boundary**.



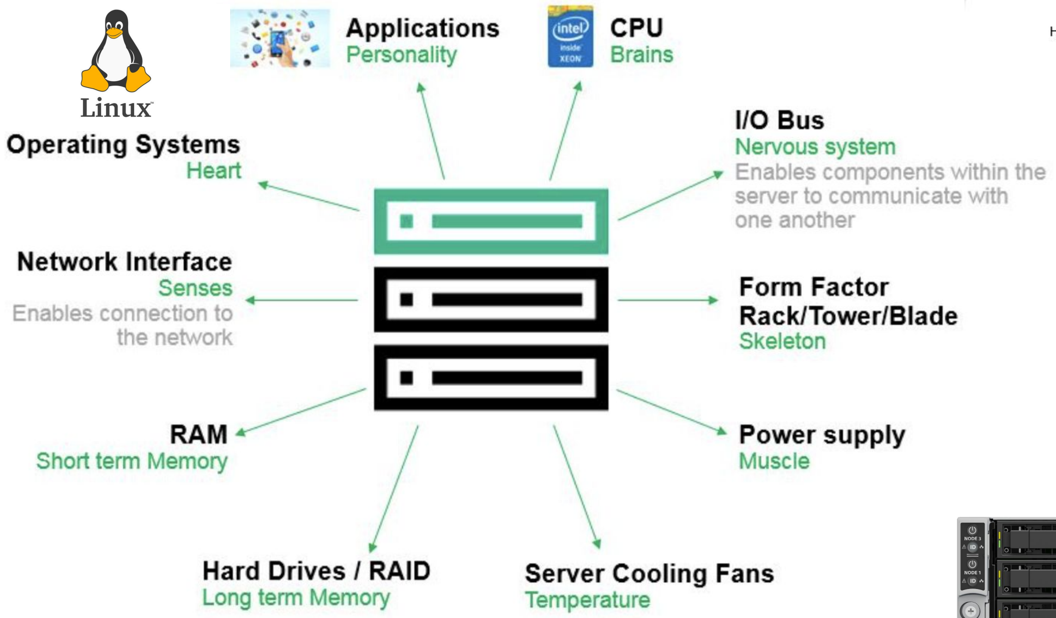
CHALMERS
UNIVERSITY OF TECHNOLOGY
E-COMMONS

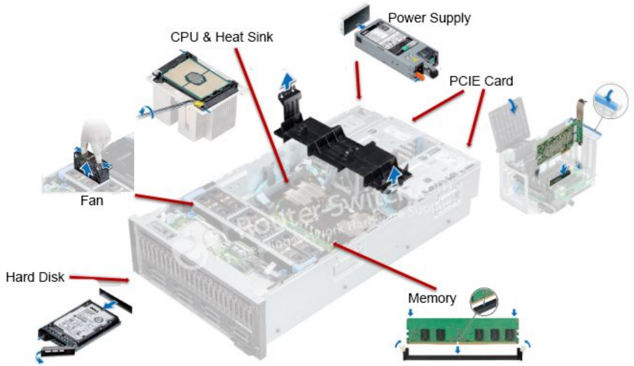
Data Center Servers

- Servers (Compute Resource)
  - Rack-mount servers
  - Blade servers
  - Tower server
  - Mainframes



Data Center Servers





ALVIS

CHALMERS
UNIVERSITY OF TECHNOLOGY
E-COMMONS

Data Center Servers

Data Center CPU + GPU servers

Most **scientific computing environments** use a **combination of CPU + GPU** servers.
CPU handles orchestration, I/O, and non-parallel logic while **GPU** accelerates compute-heavy sections like linear algebra, simulations, or ML.

	CPU Servers	**GPU Servers**
Architecture & Parallelism	**General-purpose** processors with a few (typically 8–64) powerful cores.	Designed with **thousands of smaller, simpler cores** (e.g., 7,000+ cores per GPU).
	Optimized for **sequential tasks** and **complex logic.**	Excellent at **massive parallelism**—performing the same operation across large data sets (SIMD: Single Instruction, Multiple Data).
	Handle a wide range of workloads, from file systems to databases to simulation logic.	
Best-Suited Tasks	Works best for tasks that are **control-heavy, branching-intensive, or sequential.** **Examples**: Data management, Control flow in simulations, Pre/post-processing of data.	Excels in **highly parallelizable computations.** **Examples**: Matrix algebra, Molecular dynamics, Climate modeling, Finite element analysis,Machine learning and AI.

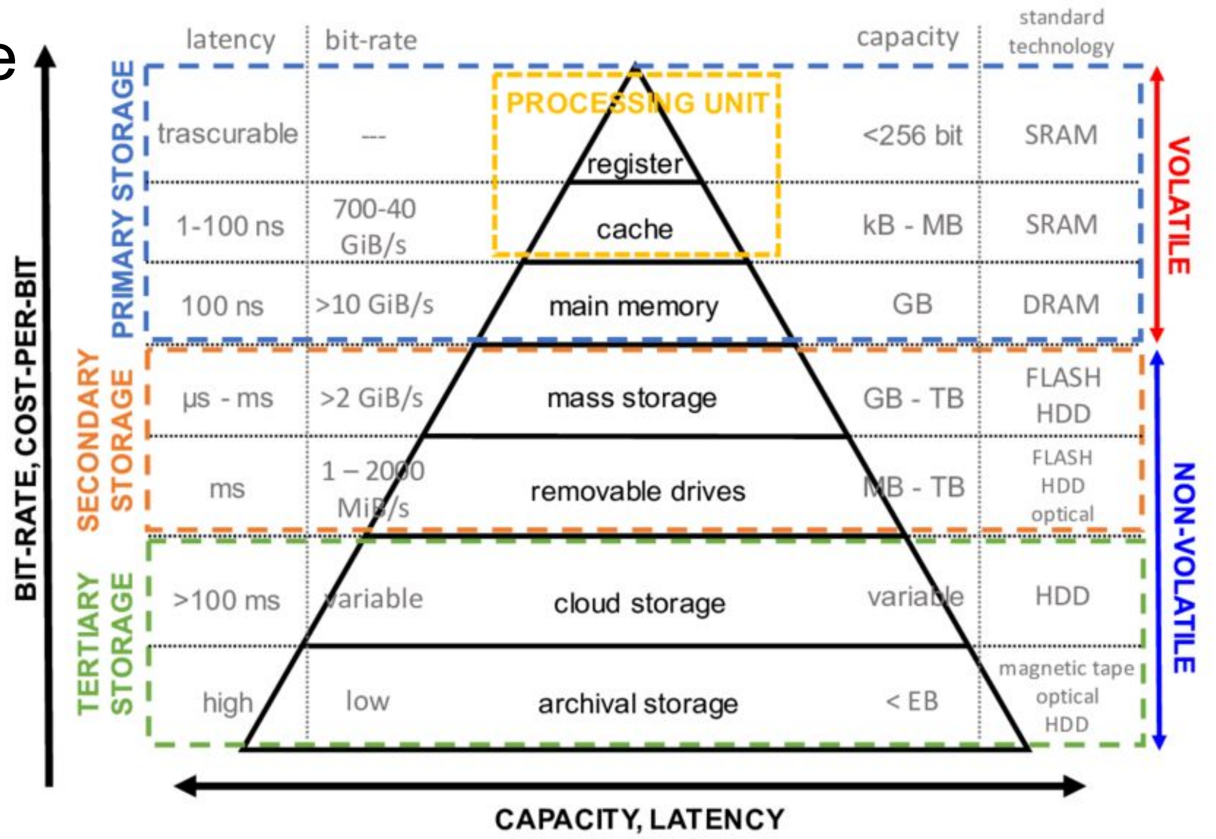
Data Center CPU + GPU servers (2)

Most **scientific computing environments** use a **combination of CPU + GPU** servers:

- **CPU** handles orchestration, I/O, and non-parallel logic.
- **GPU** accelerates compute-heavy sections like linear algebra, simulations, or ML.

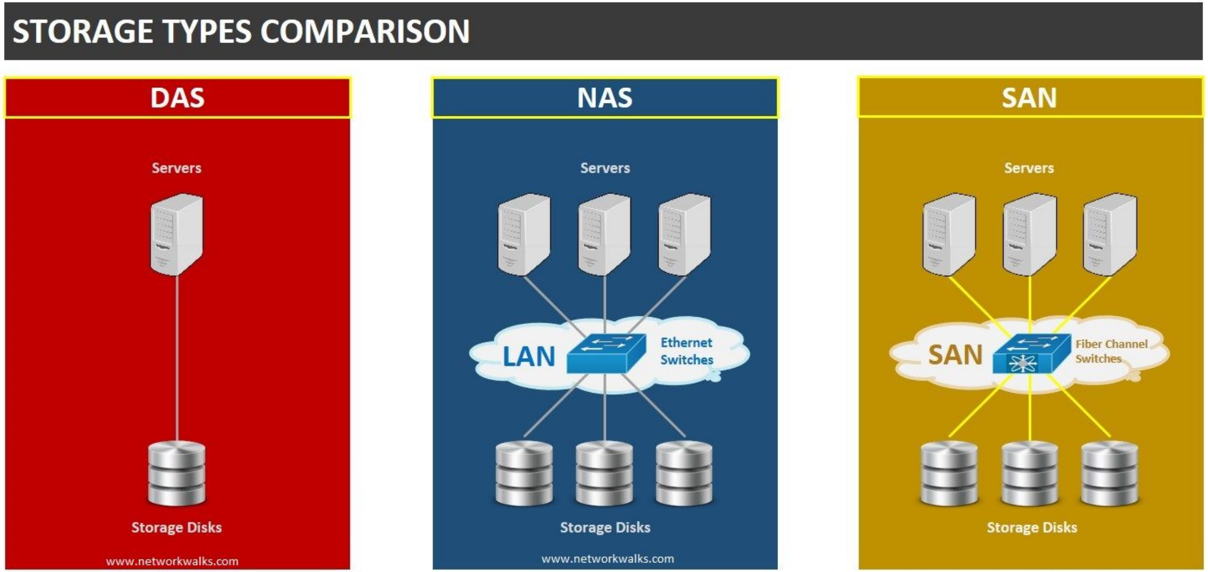
	CPU Servers	**GPU Servers**
Performance & Efficiency	Slower for massively parallel tasks but **more versatile**	Offers **order-of-magnitude speedups** (10×, 100×+) on suitable tasks
	Often **bottlenecked** when handling workloads like deep learning or large-scale numerical simulation	More energy-efficient for parallel workloads but needs well-optimized code to fully utilize.
Programming & Ecosystem	Easier to program using traditional languages (C/C++, Fortran, Python).	Requires specialized programming (CUDA, OpenCL, HIP, etc.).
		Increasingly supported in scientific libraries (TensorFlow, PyTorch, cuBLAS, etc.).

Data Center Storage



Data Center Storage

- Storage Configurations
  - DAS
  - NAS
  - SAN
  - Tape Libraries



CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

Block Storage …

- came first, in the 1960s
- are HDDs are SSDs that are physically attached to servers
- presents the raw blocks to the server as a volume
- it is presented as a raw device (a block device)
- can be used :
  - with filesystem
    - raw block storage is first **partitioned**
    - partitioning defines logical sections of the storage
    - partition is then **formatted** (Linux: ext4, xfs, btrfs Windows: NTFS, FAT32, exFAT)
  - without filesystem
    - Some applications use raw block storage directly (e.g., databases or virtual machine disk images, high-performance applications) for performance reasons (raw device mapping or block device access)

File Storage …

- is built on top of block storage
- is a file level storage
- stores data as files organized in a hierarchical structure (directories, subdirectories)
- is not to be confused with filesystem
    - It **organizes data blocks** into files and directories **on a disk or partition.**
- is a storage service or technology that allows you to store and retrieve files over a network or locally
- simplicity makes it a great solution for sharing a large number of files and folders within an organization
- provides access to files (with metadata) and handles file-sharing, permissions, locking, etc
- in most cases, especially in network-based file storage (like NFS, SMB, NAS), file storage is associated with a file server
- when accessed locally the OS itself plays the role of managing file storage

Think of file storage as the full system that offers file access and storage capabilities, often across a network.

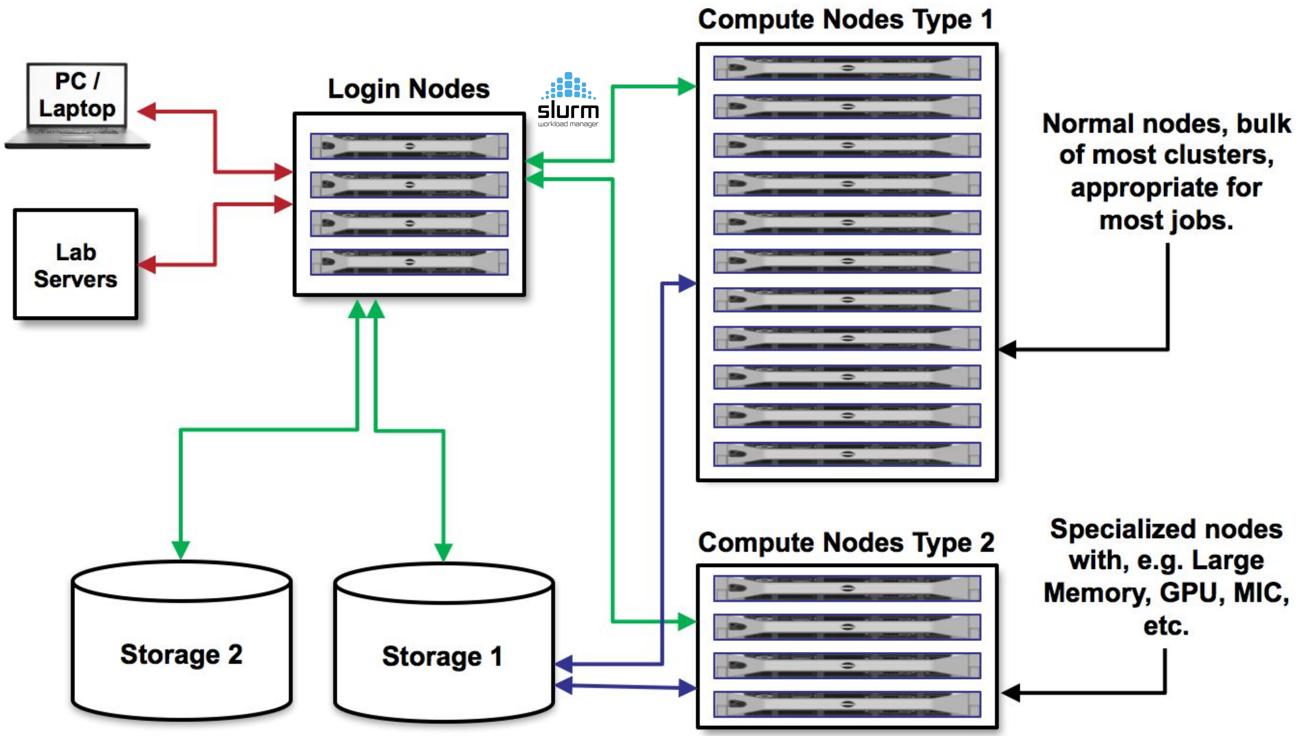
Object Storage …

- came last, New Kid On The Block
- no hierarchical directory structure
- stores all data as objects in a flat structure
- designed for **unstructured data** such as media, documents,
- logs, backups, application binaries and VM images
- does deliberate tradeoff to sacrifice performance for high durability, vast scale, and low cost
- relatively “cold” data (warm?) and is mainly used for archival and backup
- Data access is normally provided via a RESTful API, relatively slow compared to other storage types
- When a portion of the file is updated, an entire object needs to be updated, unlike in block storage, where only the corresponding block is updated.
- Hence Object store is well suited for the write-once and read many applications (static content, photo or video repository).

High Performance Computing Overview

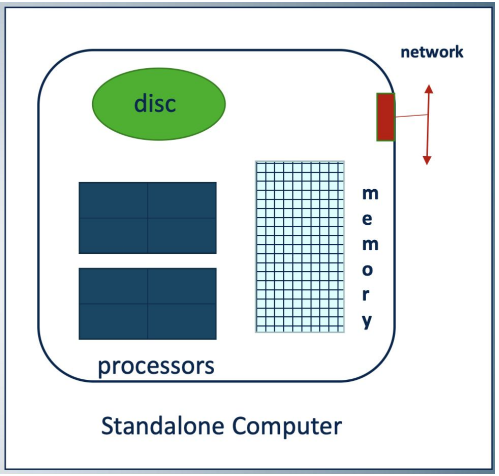
High-performance computing (**HPC**) is the use of supercomputers and computer clusters (+ fast networks + massive fast storage) to solve advanced and large computation problems.

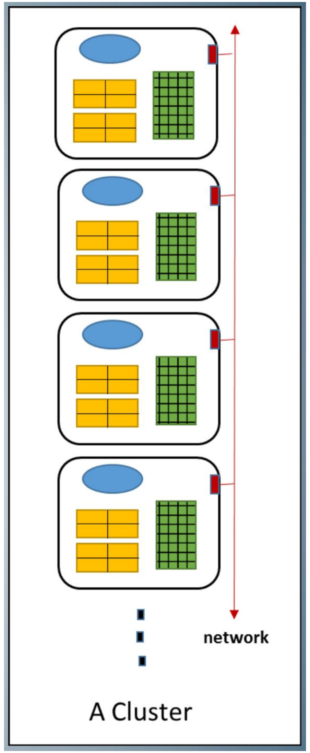
HPC Cluster

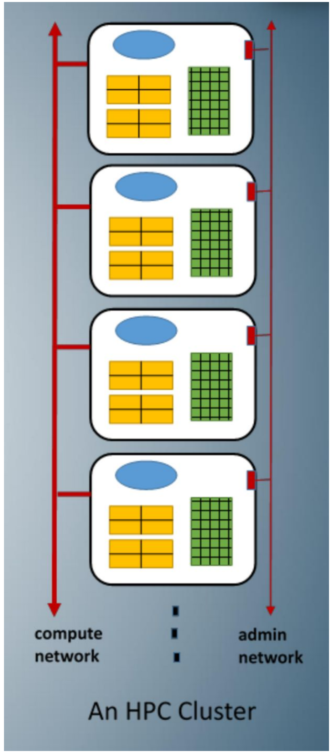


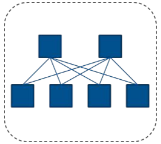
CHALMERS
UNIVERSITY OF TECHNOLOGY
E-COMMONS

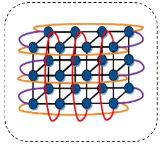
HPC and Network

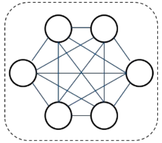


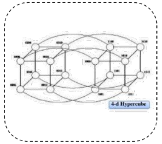


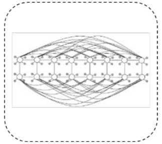



Fat Tree


Torus


Dragonfly


Hypercube


HyperX

HPC and Storage

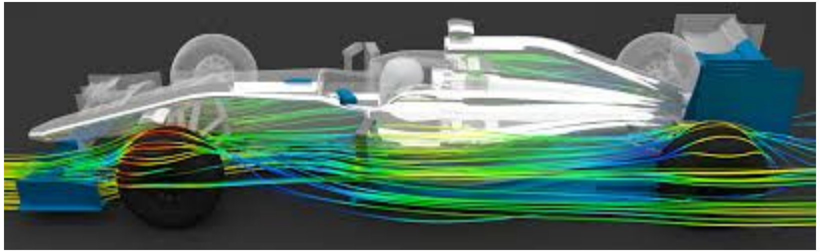
	Path	Intended use	Hardware partition used
User home	`/users/<username>`	User home directory for personal and configuration files	LUMI-P
Project space	`/project/<project>`	Project home directory for shared project files	LUMI-P
Project scratch	`/scratch/<project>`	Temporary storage for input, output or checkpoint data	LUMI-P
Project flash	`/flash/<project>`	High performance temporary storage for input and output data	LUMI-F

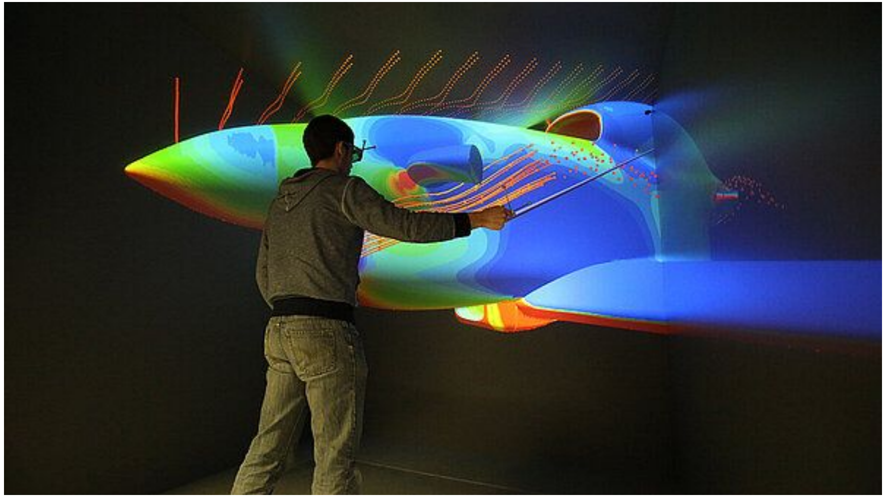
	Quota	Max files	Expandable	Retention	Billing rate
User home	20 GB	100k	No	User lifetime	NA
Project space	50 GB	100k	Yes, up to 500GB	Project lifetime	1x
Project scratch	50 TB	2000k	Yes, up to 500TB	Project lifetime*	1x
Project flash	2 TB	1000k	Yes, up to 100TB	Project lifetime*	3x

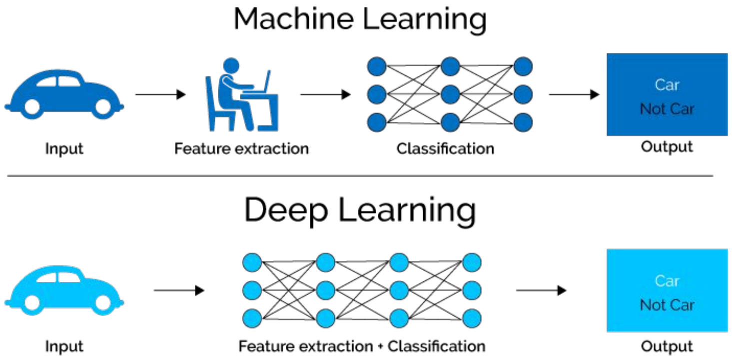
	Quota	Max objects	Expandable	Retention	Billing rate
Object storage	150 TB	500M (500k/bucket)	Yes, up to 2.1 PB	project lifetime	0.25x

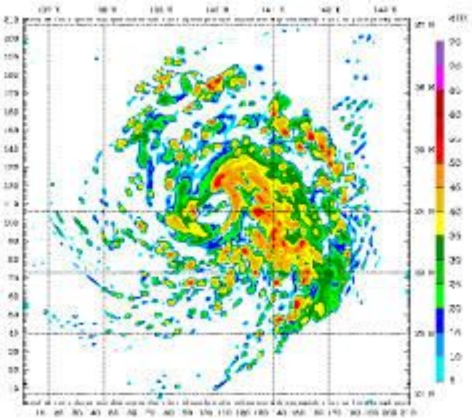
CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

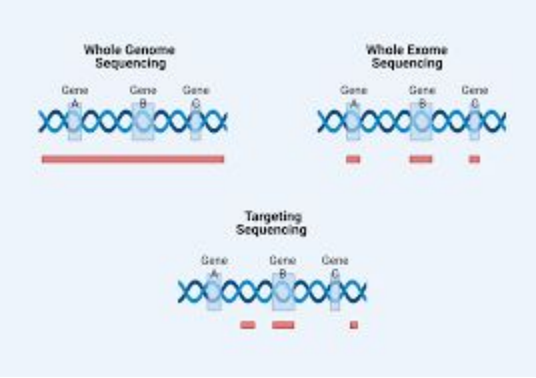
HPC Applications











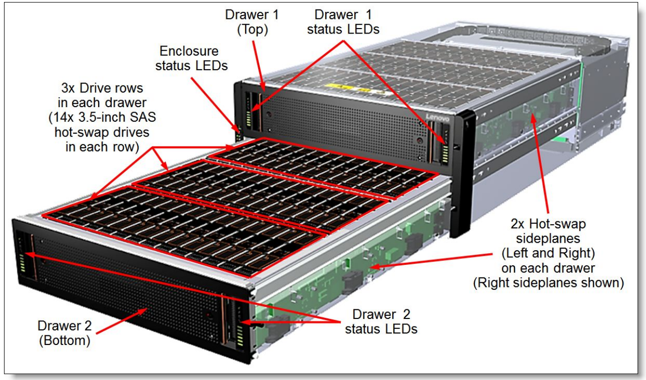
CHALMERS
UNIVERSITY OF TECHNOLOGY
E-COMMONS

Ceph

is a **distributed storage system** designed to provide excellent performance, reliability, and scalability. Ceph is often used for **object storage, block storage, and file systems**.

Mimer-Ceph

/cephyr file-system



14 Lenovo SR630v2 servers

2 x **Intel Xeon Silver** 4314 16c 2.3GHz Processors, 256 GB memory
2 x M.2 5300 **480GB SSD** (Mirrored for OS)
3 x **800GB NVMe** PCIe 4.0 (for Ceph journal and database)
1 x Mellanox ConnectX-6 **100 Gb** 2-port Ethernet adapter
1 x Mellanox ConnectX-6 **10/25 Gb** 2-Port Ethernet adapter

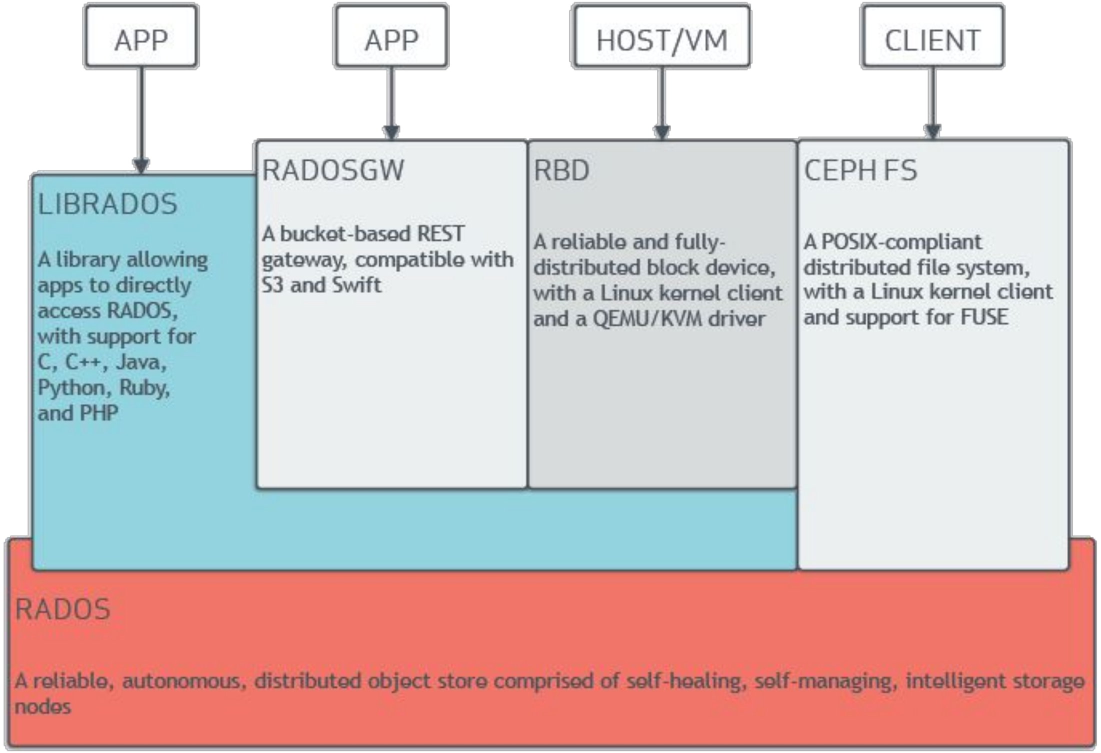
A total of 8232 TB raw
6860 TB usable capacity

7 Lenovo D3284 JBOD

84x 14TB SAS HDD Drives

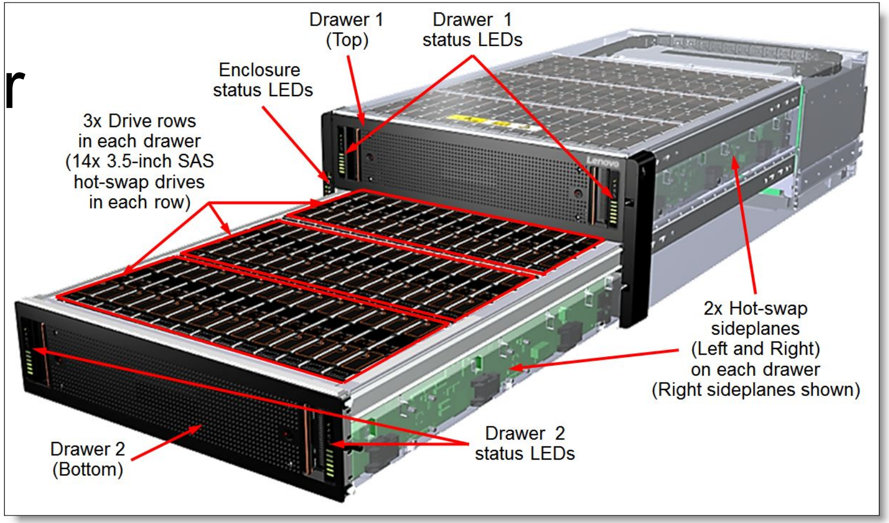
CHALMERS
UNIVERSITY OF TECHNOLOGY
E-COMMONS

Ceph as Software Defined Storage



Mimer - Ceph as part of Mimer

All-flash tier; 14 SR630v2 servers with:
2 x Intel Xeon Gold 6338 32c 2.0GHz Processor, 384 GB memory
2 x M.2 5300 480GB SSD (Mirrored for OS)
10 x Intel P5500 7.68TB NVMe PCIe 4.0
2 x Mellanox ConnectX-6 HDR Infiniband adapters
1 x Mellanox ConnectX-6 Lx 10/25GbE Ethernet adapter
A total of 1075 TB raw / 740 TB usable capacity

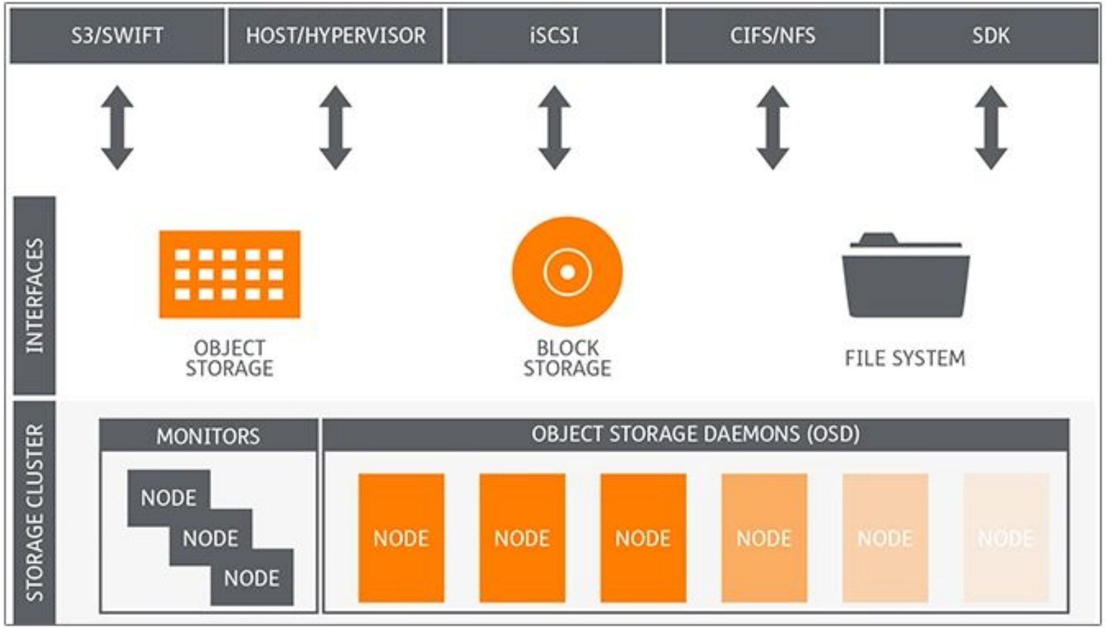


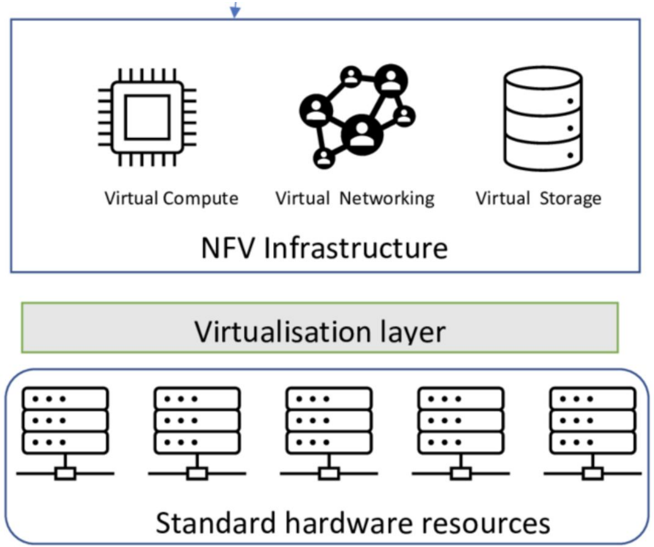
Bulk tier; 14 SR630v2 servers:
2 x Intel Xeon Silver 4314 16c 2.3GHz Processors, 256 GB memory
2 x M.2 5300 480GB SSD (Mirrored for OS)
3 x 800GB NVMe PCIe 4.0 (for Ceph journal and database)
1 x Mellanox ConnectX-6 100 Gb 2-port Ethernet adapter
1 x Mellanox ConnectX-6 10/25 Gb 2-Port Ethernet adapter
Connected (2 servers to one JBOD) to a
D3284 JBOD with 84x 14TB SAS HDD Drives

A total of 8232 TB raw / 6860 TB usable capacity

CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

Virtualisation of resources

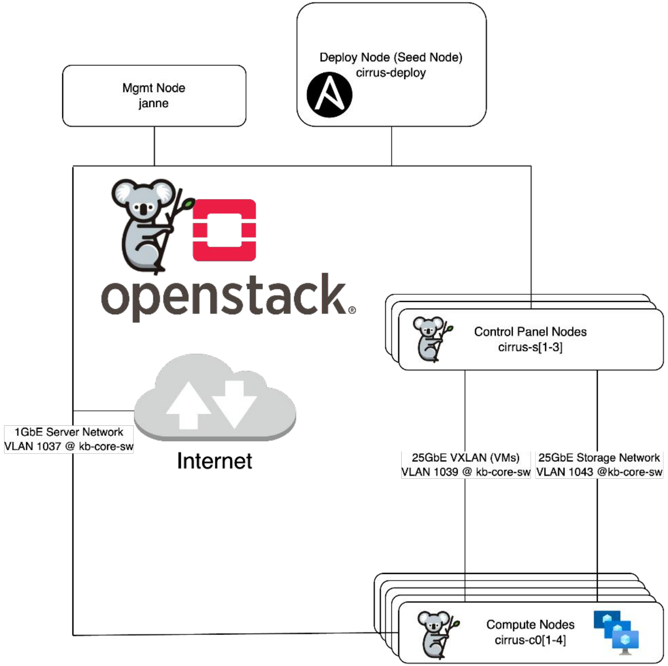


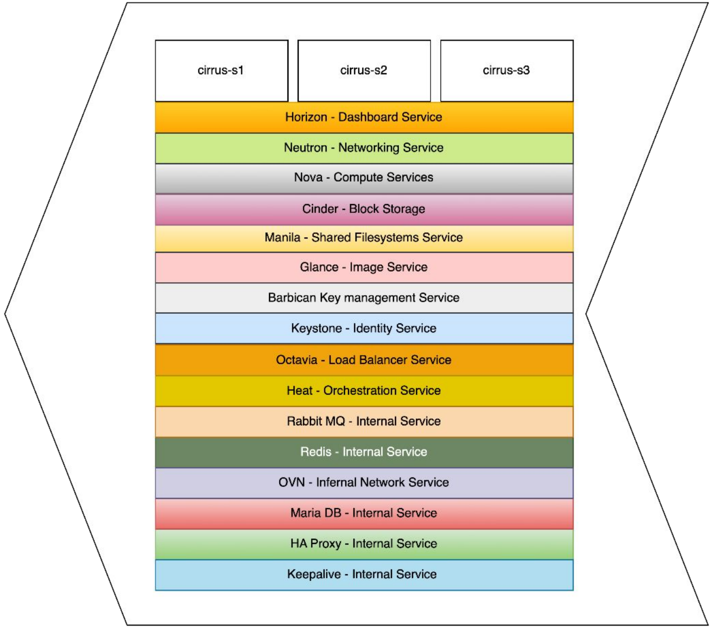


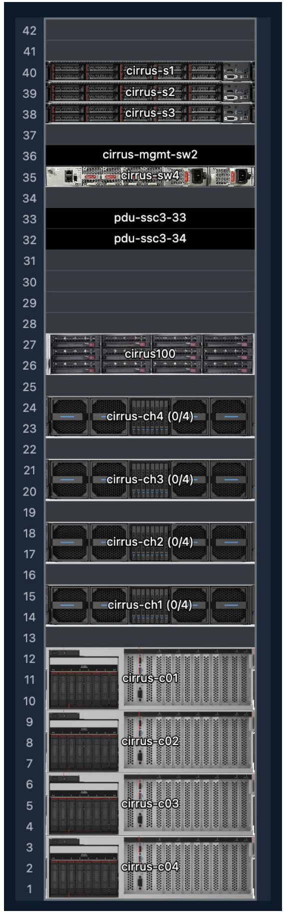
OpenStack Overview

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.

OpenStack Cluster







CHALMERS
UNIVERSITY OF TECHNOLOGY
E-COMMONS

OpenStack Components (or OpenStack Services)

Compute

NOVA	Compute Service
ZUN	Containers Service

Storage

SWIFT	Object store
CINDER	Block Storage
MANILA	Shared filesystems

Networking

NEUTRON	Networking
OCTAVIA	Load balancer
DESIGNATE	DNS service

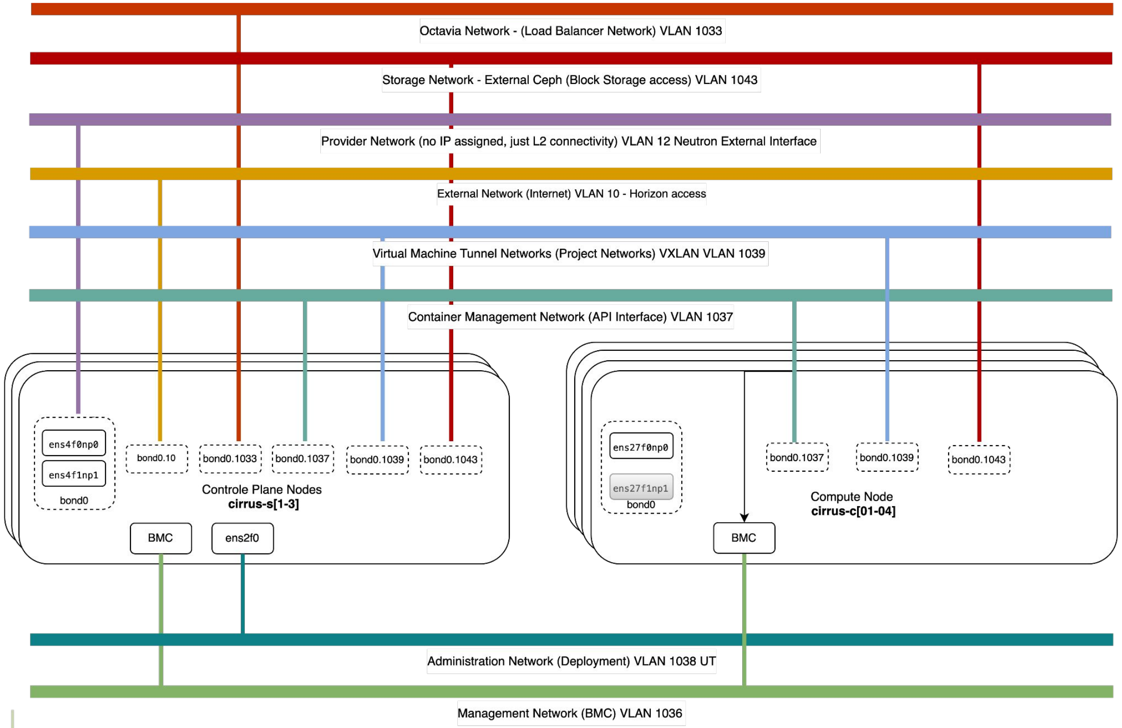
Shared Services

KEYSTONE	Identity service
PLACEMENT	Placement service
GLANCE	Image service
BARBICAN	Key management

Web frontends

HORIZON	Dashboard
SKYLINE	Next generation dashboard

OpenStack Network (to “rule” them all)



CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

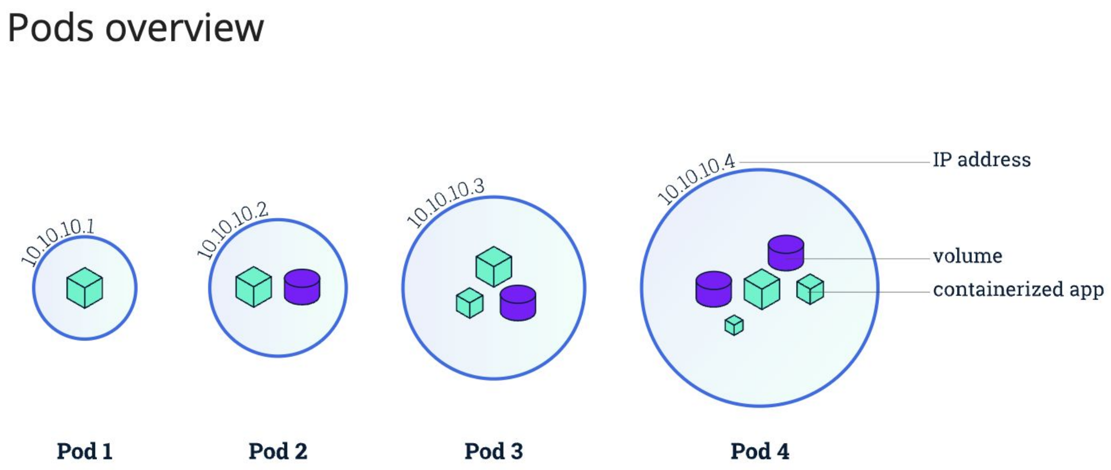
OpenStack Project

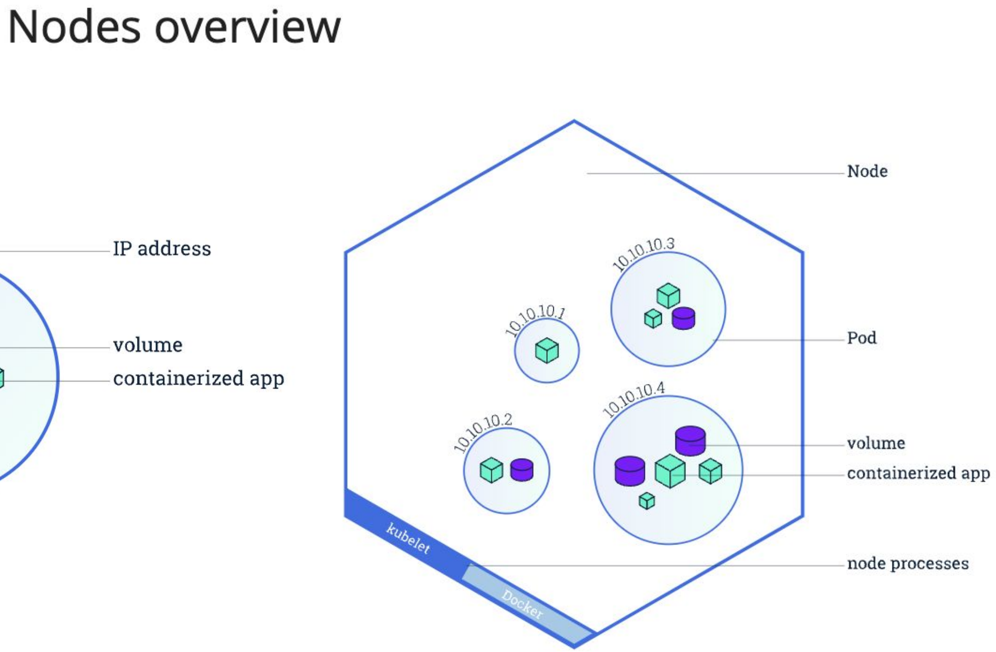
Horizon walk-through

Kubernetes (K8s) Overview

Kubernetes is a production-grade, open-source platform that orchestrates the placement (scheduling) and execution of **application containers** within and across computer clusters.

Kubernetes Components

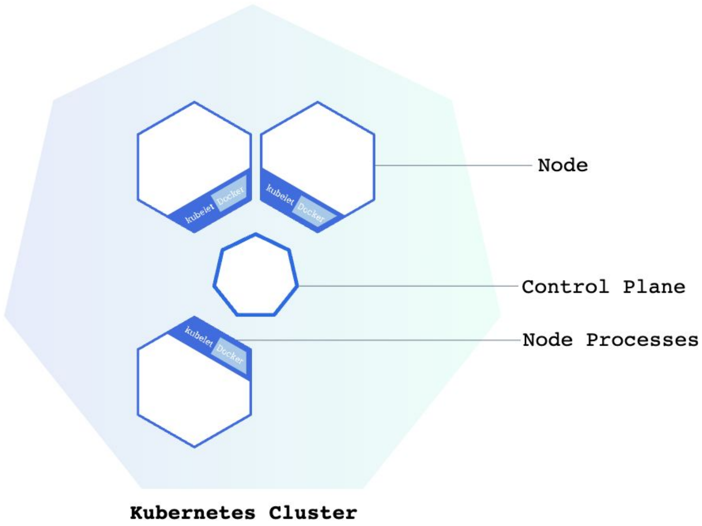


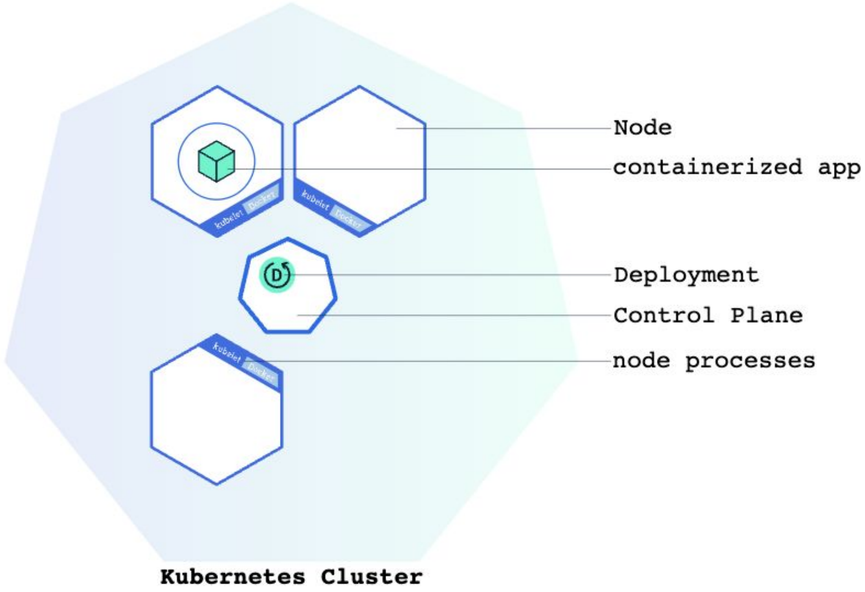


CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

Kubernetes Cluster

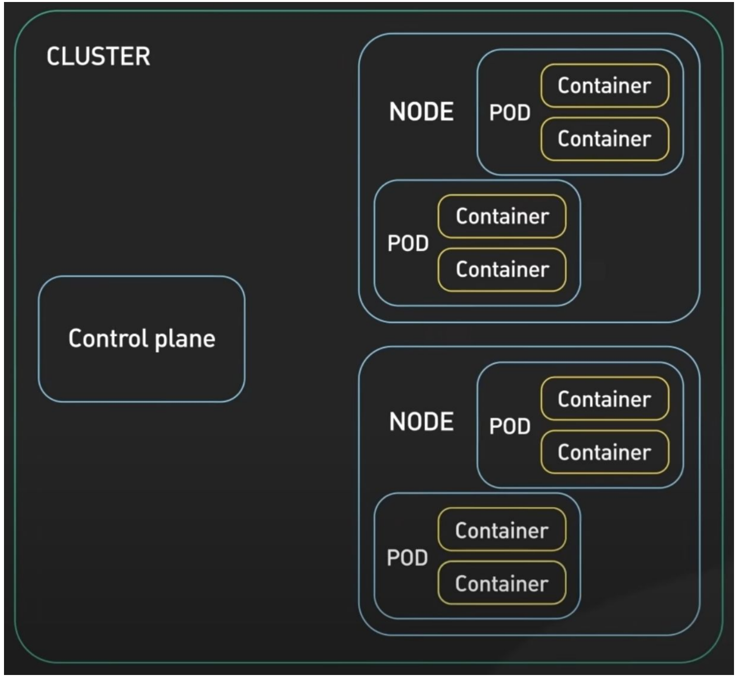
Cluster Diagram

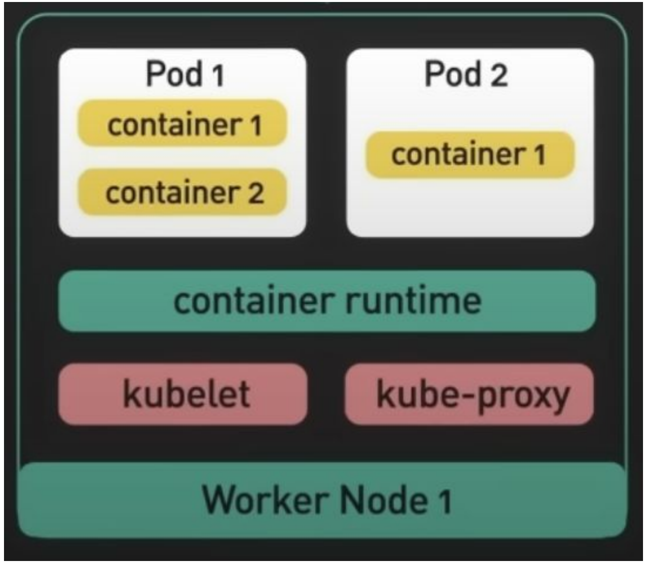




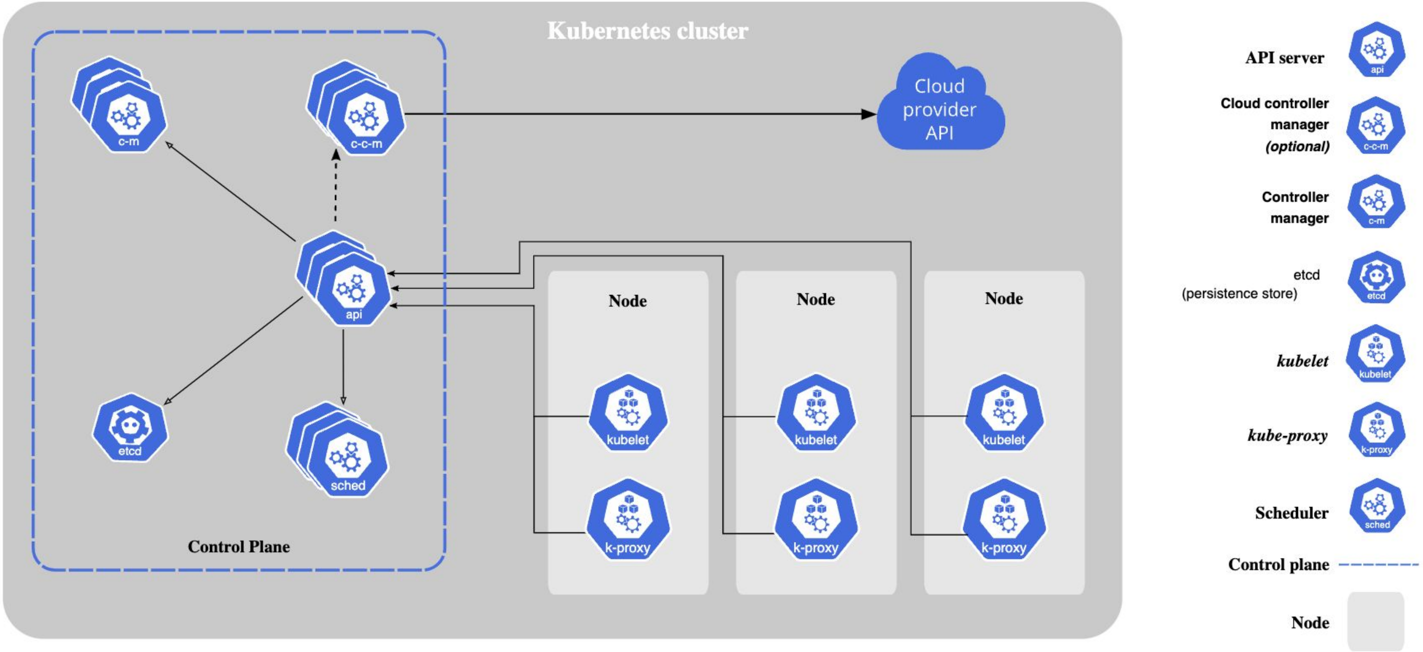
CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

Kubernetes Cluster

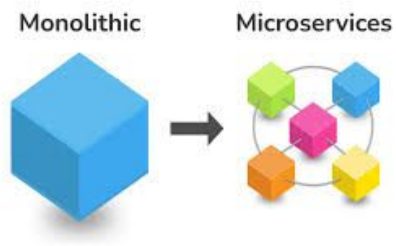


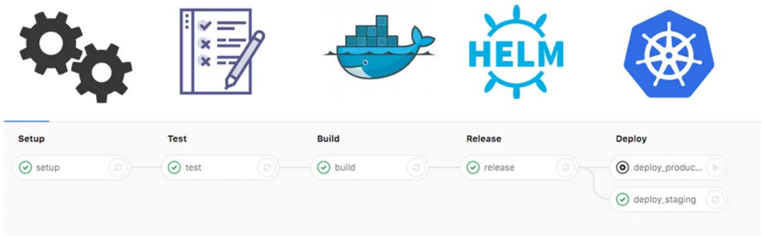


Kubernetes Components Overview (1)



Kubernetes

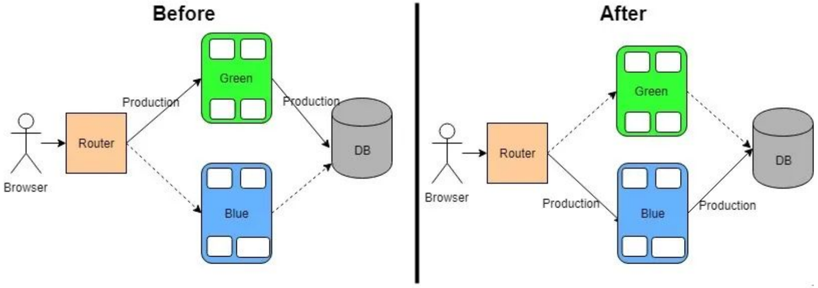




jupyterhub

Kubeflow

mlflow™



Summary

HPC CPU Cluster
Good for:
Scientific simulations (e.g., climate models, physics)
Large-scale mathematical computations
Code with complex branching and low parallelism
Strengths: Precise, flexible, scales well with many CPU cores

HPC GPU Cluster
Good for:
Deep learning / AI training
Image and video processing
Highly parallel workloads (e.g., molecular dynamics)
Strengths: Massive parallelism,
faster than CPUs for data-heavy tasks

OpenStack
Good for:
Building private or hybrid clouds
Managing virtual machines, storage, and networking
Large-scale, enterprise cloud deployments
Strengths: Full IaaS control

Kubernetes
Good for:
Running and managing containerized applications (e.g., Docker)
Microservices architecture, DevOps, CI/CD
Scaling, self-healing, and automating app deployment
Strengths: Portability, resilience, automation at scale


THE
END