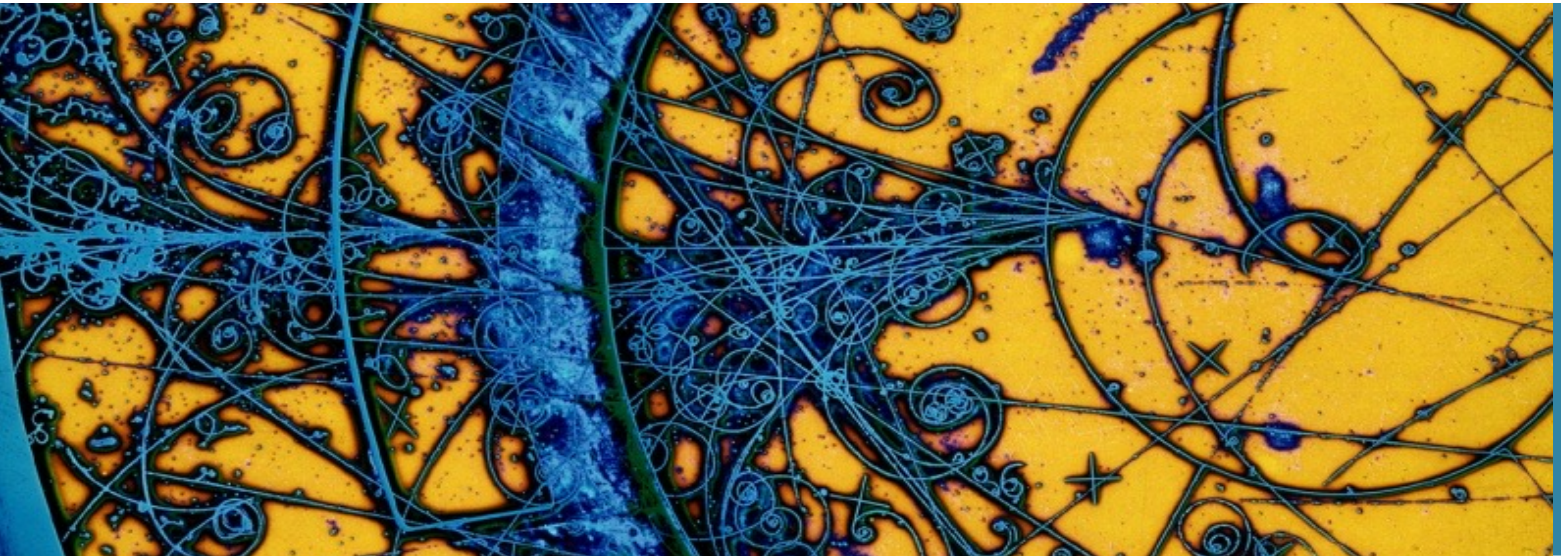
Porting the IRCCS Sant'Orsola Computational Genomic platform on INFN Cloud: a first proof of concept



Jacopo Gasparetto

International Symposium on Grid & Clouds (ISGC)
24 – 29 March 2024

SERVIZIO SANITARIO REGIONALE
EMILIA-ROMAGNA
Azienda Ospedaliero - Universitaria di Bologna
IRCCS Istituto di Ricovero e Cura a Carattere Scientifico

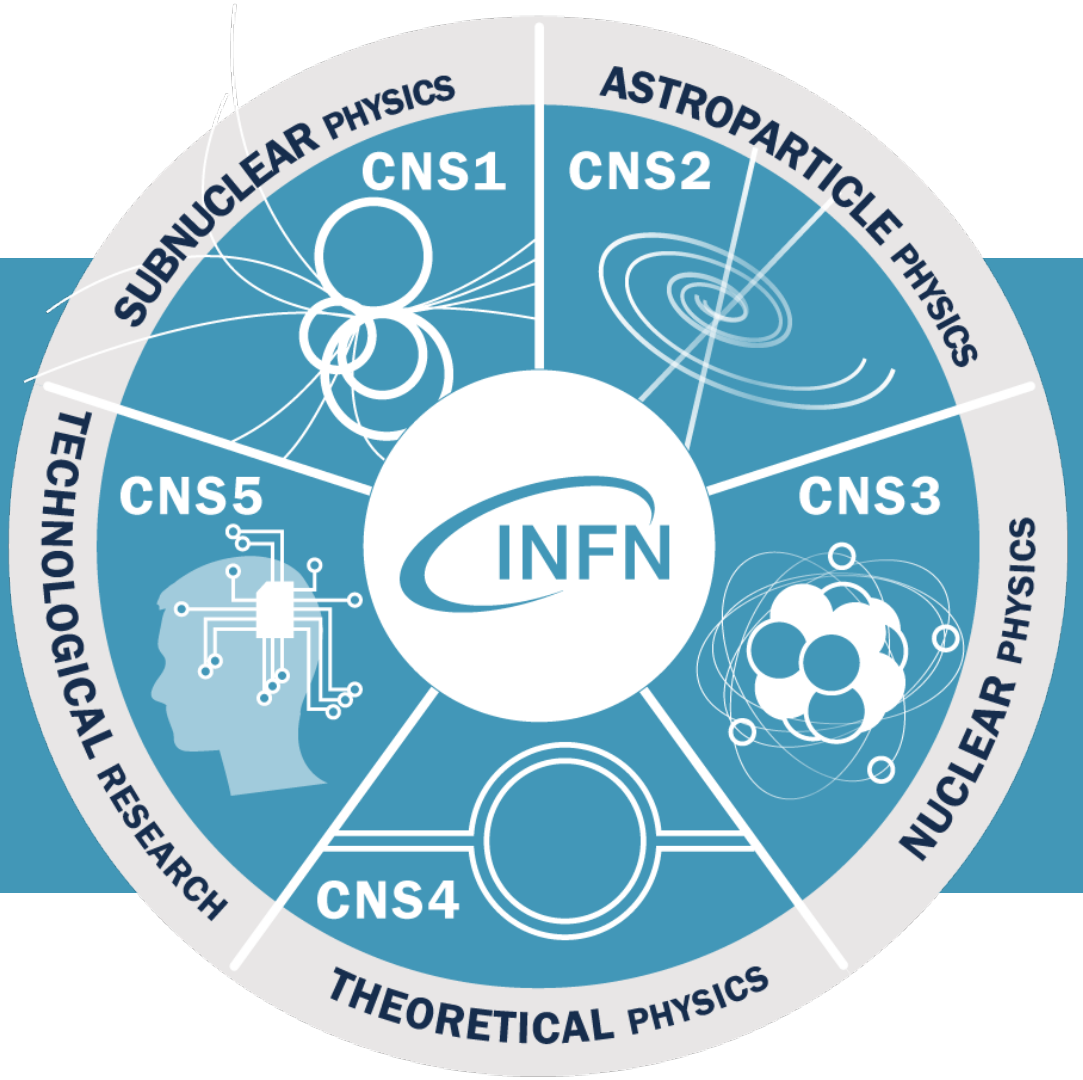
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

POLICLINICO DI SANT'ORSOLA

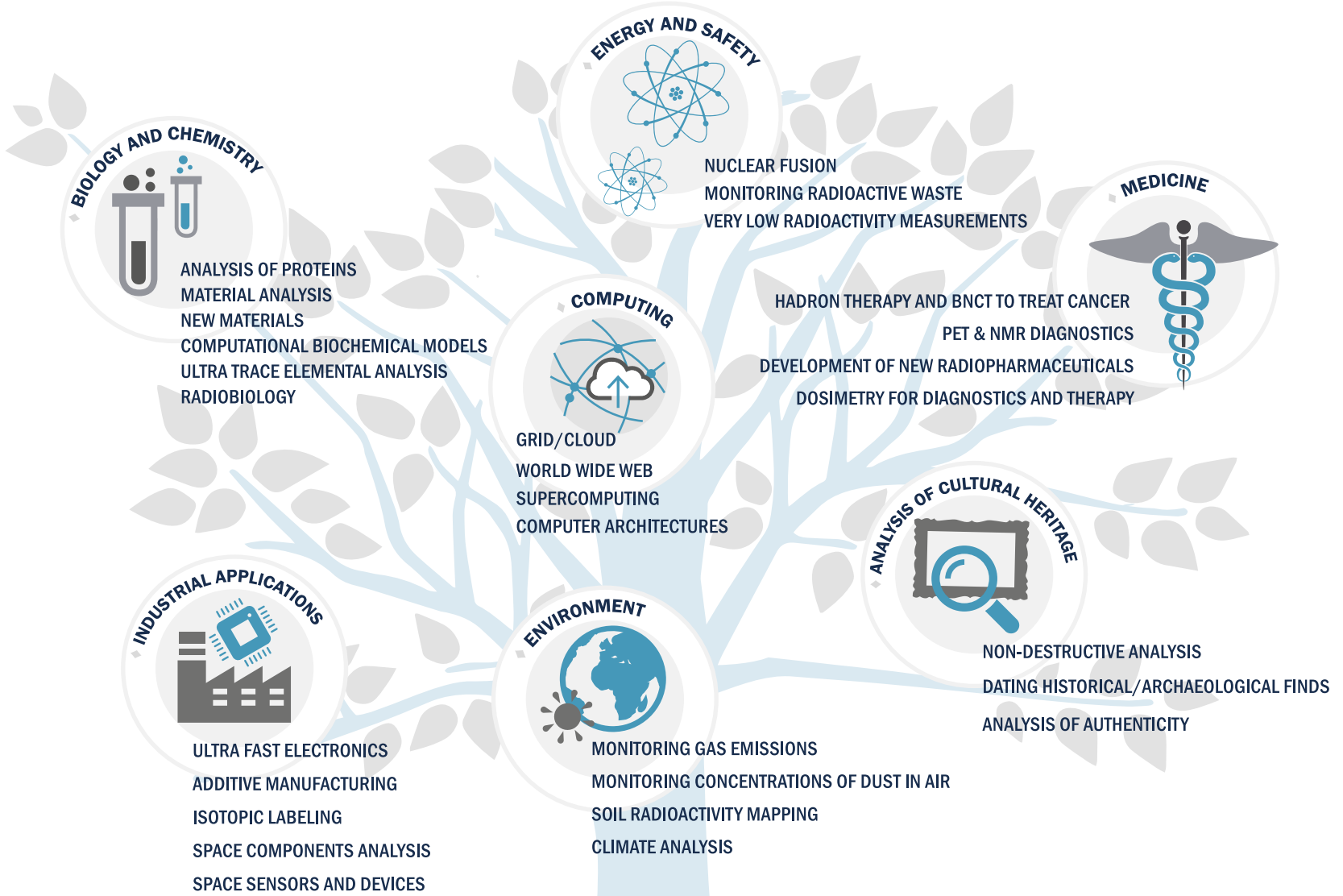
ICSC
Centro Nazionale di Ricerca in HPC,
Big Data and Quantum Computing

INFN

The **5 research lines**
and the National Scientific Committee



Knowledge Transfer



SERVIZIO SANITARIO REGIONALE
EMILIA-ROMAGNA
Azienda Ospedaliero - Universitaria di Bologna
IRCCS Istituto di Ricovero e Cura a Carattere Scientifico

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

POLICLINICO DI [illegible]

ICSC
Centro Nazionale di Ricerca in HPC,
Big Data and Quantum Computing

INFN

Enhanced PrIvacy and Compliance (EPIC) Cloud

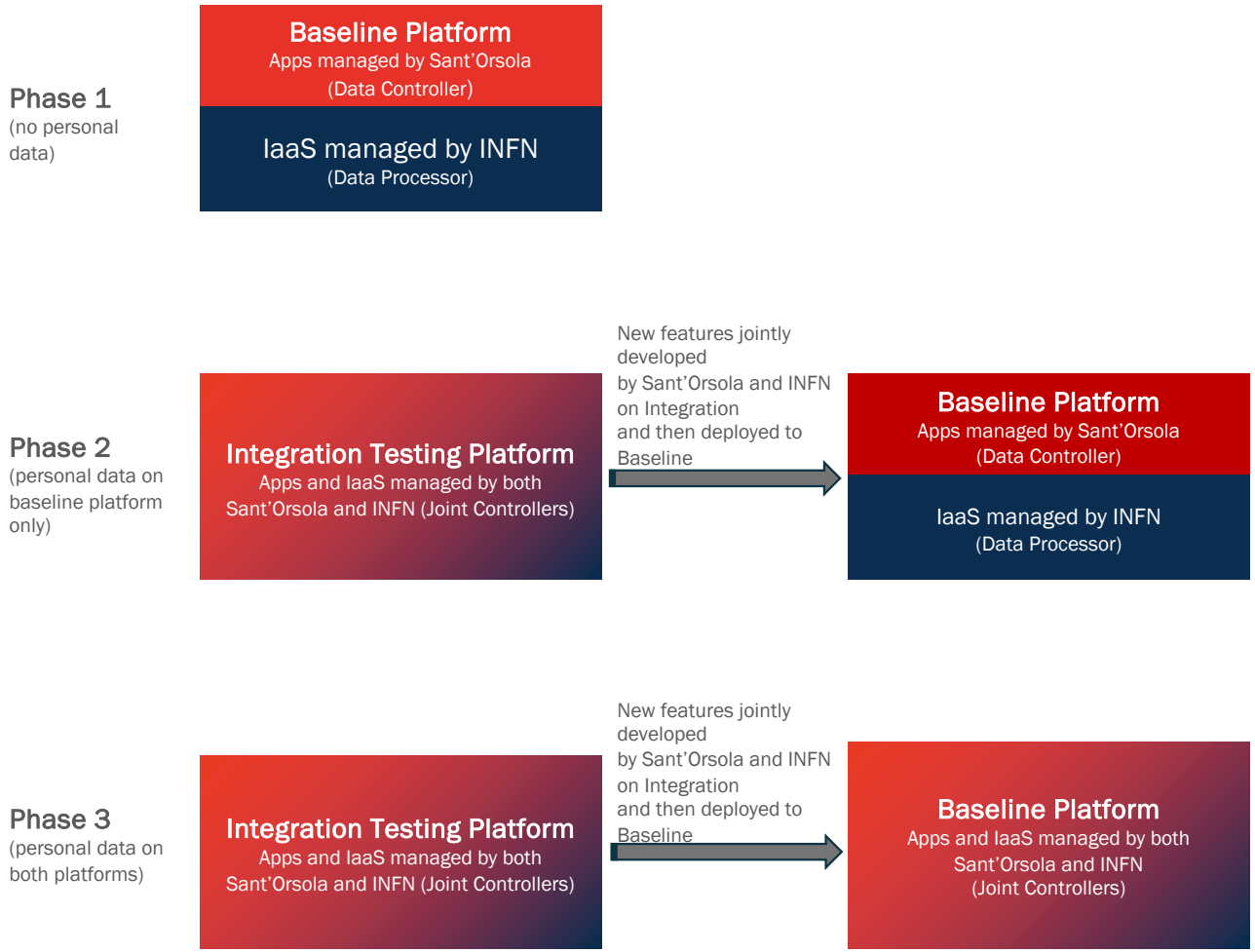
- The GDPR states that **Clinical and medical data** (for instance, genomic) **is personal data.** Thus, it fits in the Art.9 special categories of personal data.
  - **Genomic data is mostly impossible to be anonymized** → GDPR shall always be applied
  - ISO/IEC 27001 is the main certification mechanism compliant with GDPR requirements (Art. 43, 58, 63)
- To comply with the requirements of health research projects INFN is involved in, we created a portion of the INFN Data Cloud infrastructure, applied specific organizational and technical security measures, and certified it ISO/IEC 27001, 27017, 27018.
  - This is the EPIC Cloud: a reference Cloud implementation for the treatment of sensitive data at INFN.

From the Data Controller side, the fact that EPIC Cloud is ISO-certified is a way to demonstrate that processing is performed in accordance with the GDPR.

INFN - IRCCS Sant'Orsola Collaboration

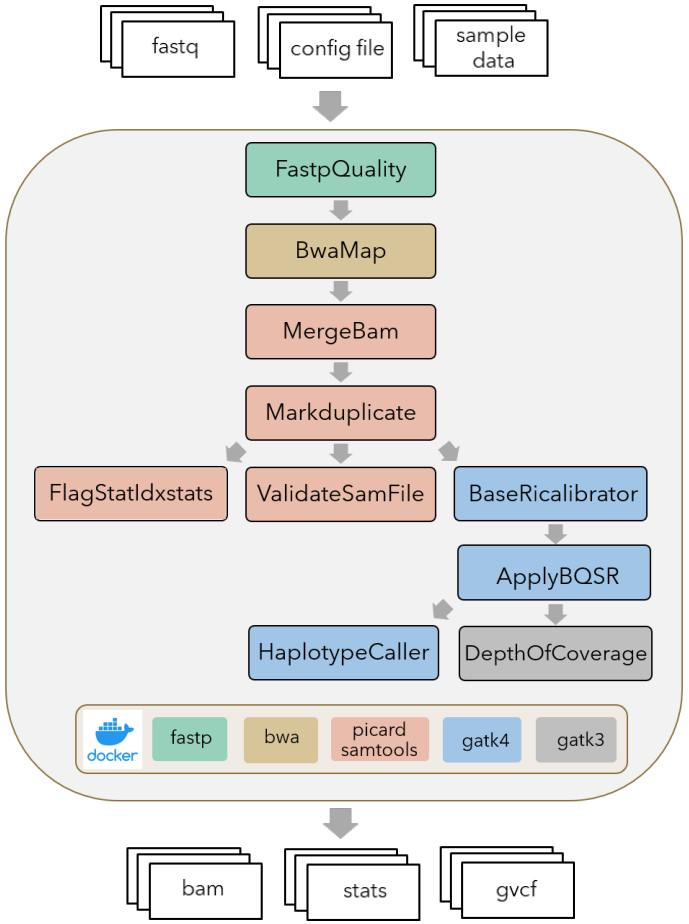
Joint research agreement with the following objectives

- secure applications for genomic data
- GPU - based solutions for genomic analysis methods
- federated and integrated cloud platforms for omics data
- adaptation of genomic pipelines to cloud and data lake architectures based on microservices
- Integration of omics data and other clinical data like Electronic Medical Records (EMR)



Benchmark workflow

- Workflow designed to detect germline variants on whole genome sequencing data
- File types: .bam, .fastq, .gvcf
- Software containerized using custom Docker image
  - Burrow-Wheeler Aligner
  - Samtools
  - Genome Analysis Toolkit
- ~100GB of input genome per sample
- ~100GB of output files per sample
  - BAM file (Binary Alignment Map format)
  - Quality metrics
  - GVCF file (Genomic Variant Call Format)
- ~ 0.5 TB of temp files



Current Computing Architecture

Architecture

- Monolithic Infrastructure based on big Virtual Machines
- Snakemake as workflow manager
- SLURM as batch system
- Conda as package and environment management system

Issues

- Low scalability
- Low availability
- Not “elastic”
- Security concerns

Where we are going

- OpenStack IaaS
- Microservices approach (containers)
- Kubernetes/RKE2 cluster
- Nextflow workflow manager
- Prometheus/Grafana
- CI/CD and private container registry
- Automation tools (Puppet, Ansible, etc...)

Kubernetes – RKE2

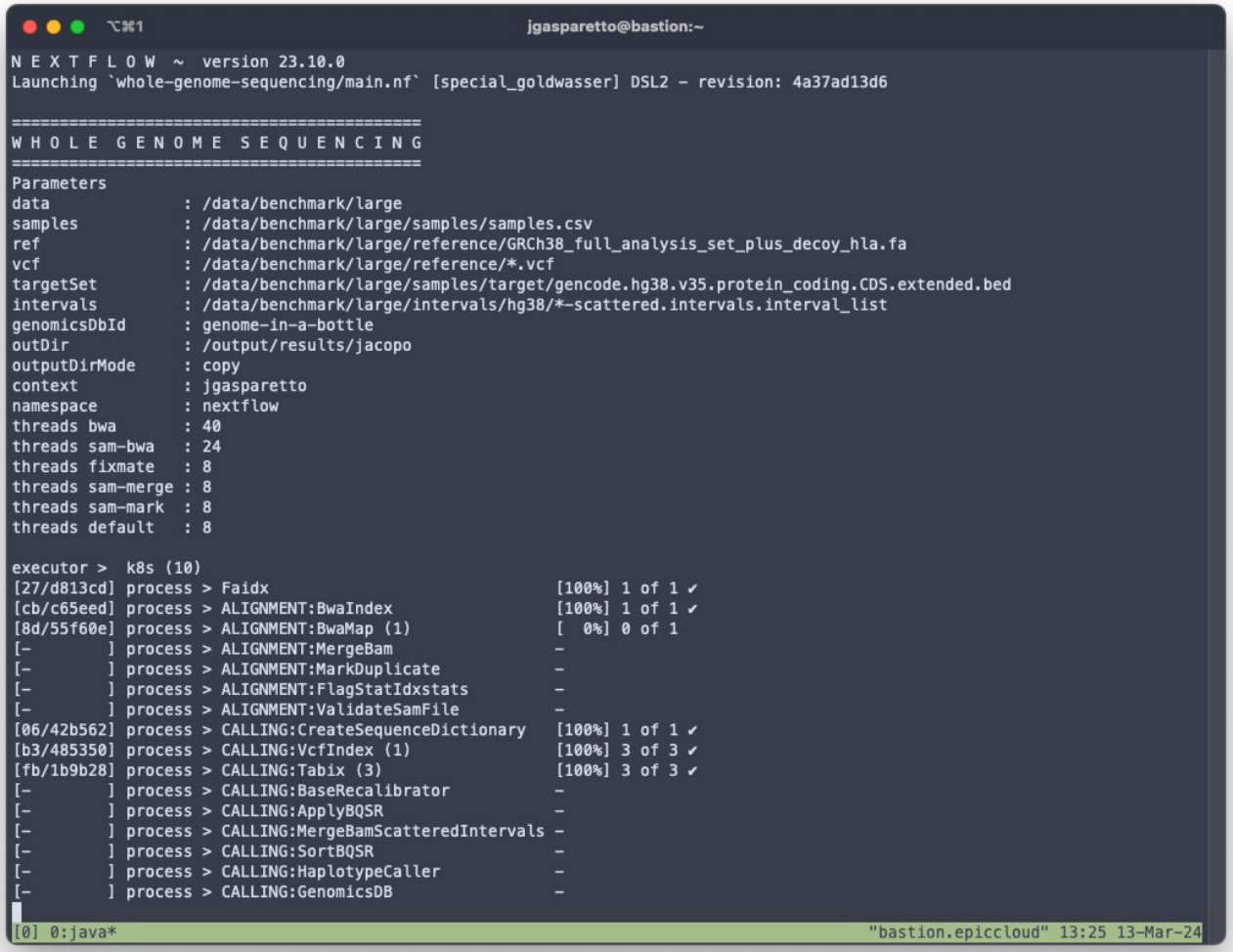
- Distributed Environment
- High Availability and Resilience
- Elasticity
- Wide community
- Well and continuously maintained
- Focused on security
- Hardening presets
- CVEs and interventions regularly published

The Cluster

- 3 Master Nodes 4 CPU, 8GB RAM
- 3 Worker Nodes 8 CPU, 16GB RAM
- 1 Worker Node 40CPU, 80GB RAM
- 3 Persistent Volume Claims (1 TB each)
- 1 Bastion 8 CPU, 16GB RAM

Nextflow + K8s

- General purposes Workflow Manager
- Well supported and widely adopted by the bioinformatics community
- Native support of containers
- Native support of Kubernetes (also Snakemake)
- Native support of K8s Persistent Volume Claims (PVC)
- Node selectors (CPU, GPU, ”small”, “medium”, ”large” nodes etc)



Nextflow vs. Snakemake

nextflow

- Pipelines written in Groovy
- Full support of containers
- The exact same workflow can run both locally or on the cloud
- Native support of Kubernetes with high customization
- K8s images cache friendly
- Native support of K8s Persistent Volume Claims (PVC)
- Data produced by previous steps is already visible to the next pods without additional transfers

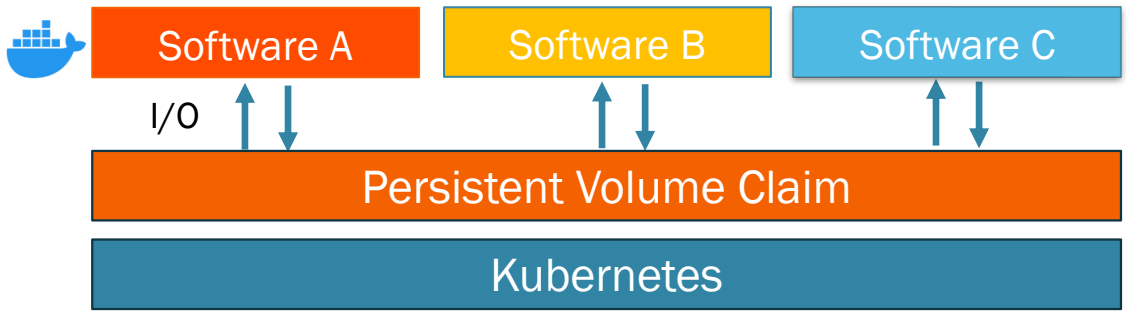
snakemake

- Pipelines written in Python
- Heavily focused on Conda environments
- Container support limited to docker-in-docker
- Images cannot be cashed by K8s
- Persistent Volume Claims not supported
- Storage I/O tight to S3/SFTP transfers
- Each pod must download/upload hundreds of GBs of data per each step

Nextflow vs. Snakemake

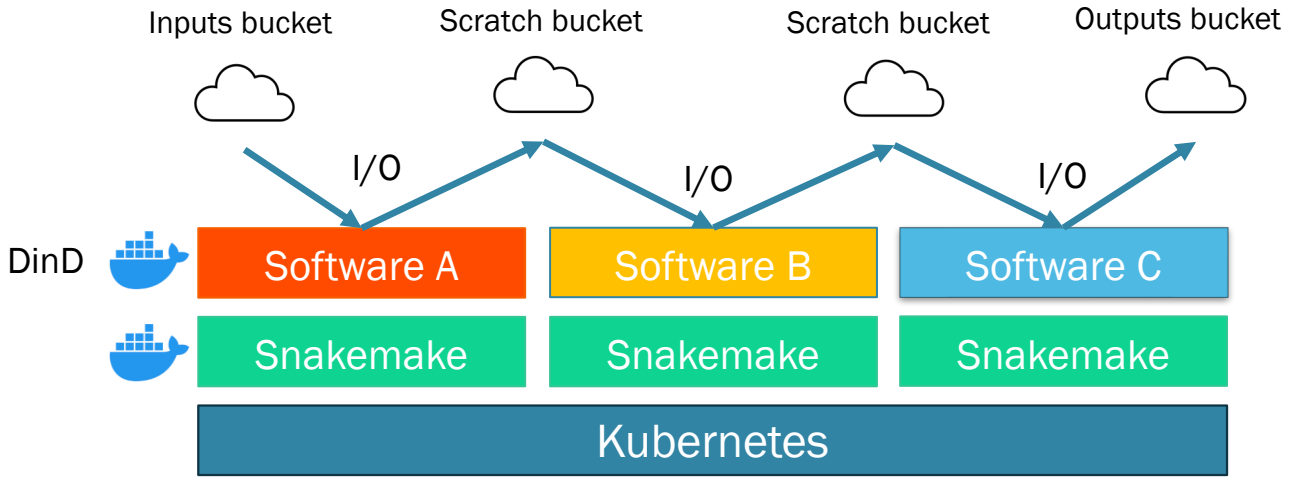
nextflow

- PVC configured as NFS volumes mounted on worker nodes
- PVC mounted directly to the pods
- Pods spawned as “native” docker images



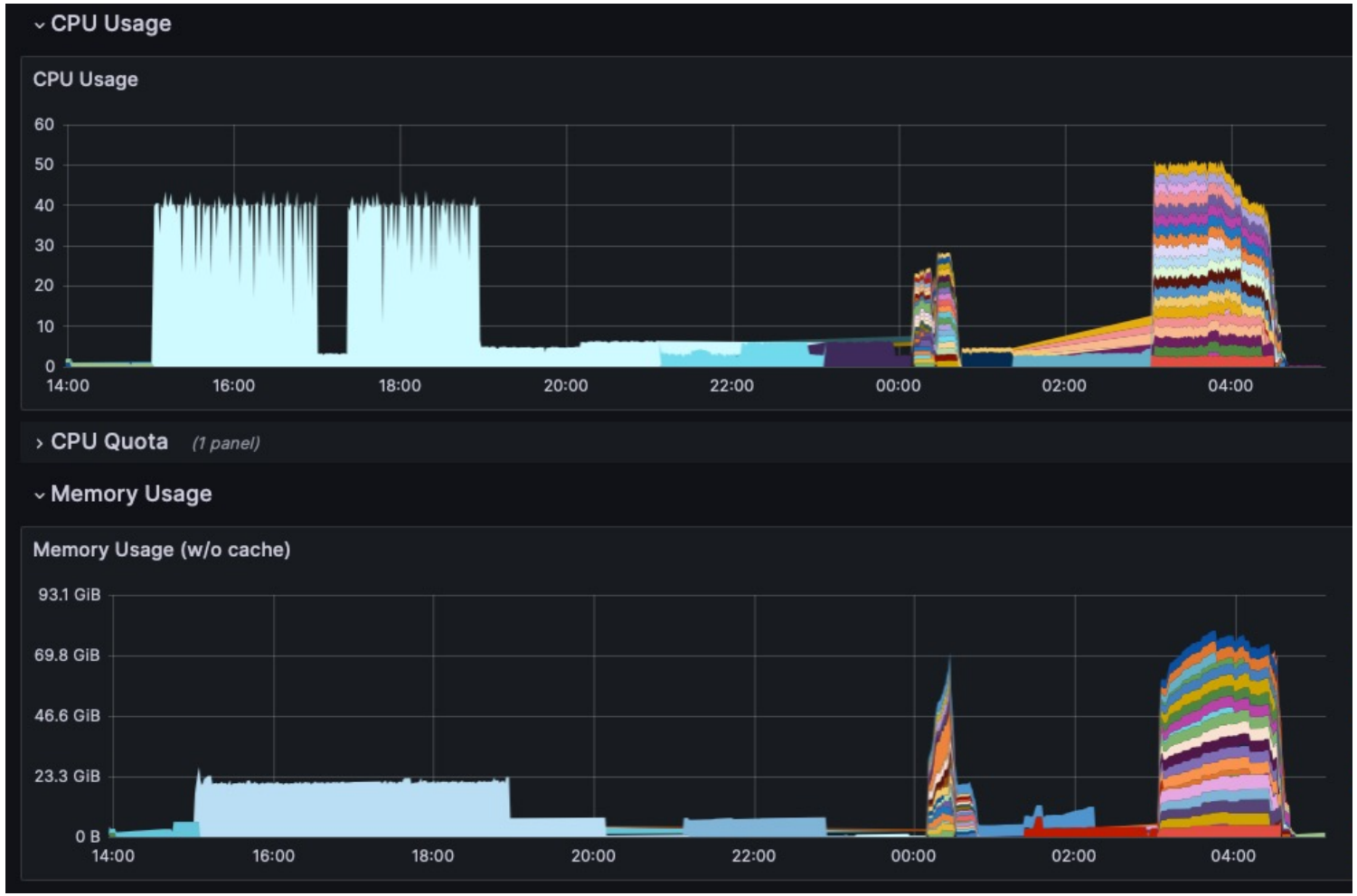
snakemake

- Files downloaded/uploaded to S3-like storage
- All pods spawned as Snakemake Docker image
- Software is run as docker-in-docker on top of it



Monitoring (Prometheus & Grafana)

- Deep insights of the workflows
- Enables accurate debugging of the single steps
- Enables the optimization of resource utilization by each step
- Discovery of bottlenecks
- Comparable and consistent benchmarks and results



Monitoring (Prometheus & Grafana)

Conclusions

- The migration from a “classical” monolithic architecture to a more flexible, even though more complex, cloud architecture reveled to be promising
- Initial overhead to build the infrastructure
- Complex infrastructure but it offers high availability, scalability and elasticity
- The adopted “microservices” paradigm offers portability since the same workflow can be run on from a small laptop up to full-size computing cluster
- User friendly for the operator/researcher who submits the jobs
- Full-fledged monitoring platform to enable deep insights about job runs, performance and optimizations
- Better security due to a better control of the employed software and Vulnerability Scans of the Docker images

Future Outlooks

- Integration of and Identity Provider (IdP) tool such as Keycloak, FreeIPA or INDIGO IAM
- Integration of Galaxy as user friendly frontend to launch jobs
- Integration with object storage solutions to store the output files
- Stress tests
- Exploration of queue managers
- Mirroring of architecture to the Baseline tenant

Many Thanks to all contributors

jacopo.gasparetto@cnaf.infn.it

Alessandro Costantini, Andrea Chierici, Daniele Cesini, Francesco Sinisi, Diego Michelotto, Giacinto Donvito, Giusy Sergi, Letizia Magenta, Lorenzo Chiarelli, Luca dell'Agnello, Luigi Scarponi, Stefano Zani, Tania Giangregorio, Federica Isidori, Emanuela Iovino, Tommaso Pippucci, Vincenzo Ciaschini, Barbara Martelli