UNIVERSITY OF COLOGNE

University of Cologne
Parallel and Distributed Systems
https://pds.uni-koeln.de

NEXUS: MALLEABLE INFRASTRUCTURE FOR DYNAMIC WORKLOAD CONVERGENCE

Simon Volpert, Robert Keßler, Marlon Engels, Max Stempel, Paul Suhr, Stefan Wesner

MOTIVATION: STATIC INFRASTRUCTURE LIMITATIONS

Modern research environments demand heterogeneous computing models—spanning High-Performance Computing, Cloud Computing, and Big Data Analytics. However, these are often deployed on statically partitioned clusters.

1. Static clusters lead to severe resource underutilization when demand shifts.
2. Re-provisioning physical servers manually is slow and error-prone.
3. Testing complex, multi-paradigm scheduling is difficult without a dynamic physical testbed.

To address these limitations, we introduce **Nexus**, an incubator research project and proof of concept for deploying complex, malleable physical infrastructure. Nexus acts as a foundational enabler for researching the dynamic orchestration and convergence of different computing models.

This physical testbed empowers researchers to investigate scheduling, dynamic resource allocation, and the convergence of previously isolated ecosystems.

CORE PRINCIPLES 1

The Nexus architecture bridges declarative infrastructure and hardware provisioning by adhering to four rigorous design principles:

Immutability: Leveraging NixOS[1] to enforce strictly reproducible node configurations, establishing a deterministic software environment aligned with computational reproducibility best practices [1].

Idempotency: Utilizing GitOps reconciliation (e.g., Flux[2]) to continuously converge the cluster toward a declared target state, satisfying strict artifact evaluation guidelines [2].

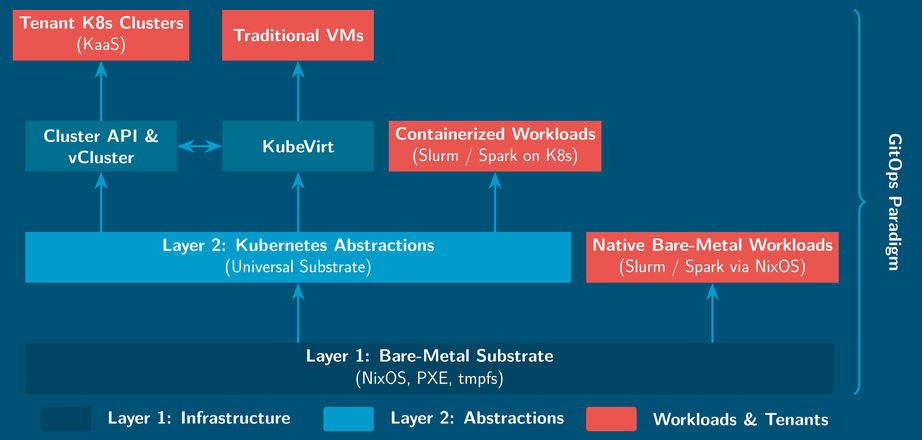
Impermanence: Employing ephemeral PXE boot sequences with in-memory (`tmpfs`) root filesystems. This strict impermanence minimizes accumulated state drift across re-provisioning cycles, ensuring experiments execute within clean and fresh environments [3].

Declarative Delivery: Defining the entire stack as code, from base operating systems to user-facing applications. This eliminates ad-hoc manual interventions, yielding a fully auditable provisioning pipeline.

[3] https://nixos.org/
[3] https://fluxcd.io/
[3] https://kubernetes.io/

ARCHITECTURE: CLOUD-NATIVE MALLEABILITY 2

Nexus facilitates the declarative definition of complex, heterogeneous topologies. By enforcing immutability and impermanence at the node level, researchers can reconfigure shared physical hardware across mutually exclusive paradigms without lingering state artifacts. As illustrated below, the architecture clearly separates the immutable bare-metal foundation (Layer 1) from the dynamic cloud-native orchestration abstractions (Layer 2).



- **Native Physical Workloads:** Provisioning conventional HPC schedulers (Slurm) and analytics frameworks (Spark) natively via NixOS.
- **Universal Kubernetes Substrate:** Deploying a foundational Kubernetes[3] [4] substrate to orchestrate containerized workloads alongside standard microservices.
- **Virtualization & Tenant Clusters:** Leveraging KubeVirt, Cluster API (CAPI), and vCluster to dynamically provision isolated tenant environments, forming the basis for Reproducibility-as-a-Service (RaaS) [2].
- **Unified GitOps Workflow:** Decoupling the bare-metal substrate (Layer 1) from the orchestration abstraction (Layer 2) via a unified GitOps pipeline to ensure auditable, extensible workflows [1].

By bridging bare-metal immutability with cloud-native orchestration, Nexus enables the automated convergence of disjoint environments onto unified, reproducible hardware [5].

PLANNED RESEARCH

Nexus is the foundational physical testbed for our ongoing research:

- **Isolation of Co-Located HPC Workloads (Simon Volpert):** Investigating strict isolation mechanisms, resource contention, and performance interference when executing native HPC tasks on shared infrastructure [6].
- **Malleability of HPC Systems (Robert Keßler):** Exploring the dynamic elasticity and malleability of HPC clusters when augmented with cloud-native ecosystems (Kubernetes) and big data analytics (Spark) [7].
- **HPC Research & Education:** Providing a reproducible sandbox for students to gain hands-on experience with HPC research and the embedding of low-level supercomputing workflows into modern cloud-native practices.

RAPID EXPERIMENTATION & VISION 3

The Nexus cluster functions primarily as a scientific instrument and an educational testbed rather than a conventional production environment. It facilitates empirical observation of system interactions, isolation boundaries, and scheduling behaviors for both researchers and students.

Vision: Holistic Infrastructure Capabilities

The capability grid below delineates the four foundational dimensions of our infrastructure design.

State Consistency
Eliminates drift and purges remnants of previous workloads via ephemeral boots.

Structural Malleability
Dynamically reshapes physical nodes to serve disjoint paradigms (HPC or K8s).

Tenancy Isolation
Achieves strict multi-tenancy through nested virtualization instead of shared OS.

Fully Declarative
Defines 100% of the stack as code, from bare-metal firmware to applications.

OPEN SOURCE AVAILABILITY

Nexus is fully open-source. The entire stack, including our NixOS configurations, Kubernetes manifests, and installation instructions, are publicly released.

nexus.pds.uni-koeln.de

REFERENCES

[1] V. Stodden and S. Miguez, "Best practices for computational science: Software infrastructure and environments for reproducible and extensible research," *Journal of Open Research Software*, 2014.

[2] J. A. Hernández and M. Colom, "Reproducible research policies and software/data management in scientific computing journals: A survey, discussion, and perspectives," *Frontiers in Computer Science*, 2025.

[3] M. Baker, "1,500 scientists lift the lid on reproducibility," *Nature*, 2016.

[4] S. Volpert, G. Eisenhart, and J. Domaschka, "Holistic life-cycle management of cloud-to-thing compute entities," in *2023 IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PerCom)*, 2023.

[5] T. Trappen, R. Keßler, R. Pabel, *et al.*, "Automated dynamic ai inference scaling on hpc-infrastructure: Integrating kubernetes, slurm and vllm," in *Next-Gen Middleware for MLOps in Distributed Systems (MIND '25)*, 2025.

[6] S. Volpert, B. Erb, G. Eisenhart, *et al.*, "A methodology and framework to determine the isolation capabilities of virtualisation technologies," in *Proc. of the 2023 ACM/SPEC Int. Conf. on Performance Engineering (ICPE)*, 2023.

[7] R. Keßler, S. Volpert, and S. Wesner, "Towards improving resource allocation for multi-tenant hpc systems: An exploratory hpc cluster utilization case study," in *2024 IEEE Int. Conf. on Cluster Computing Workshops*, 2024.