Resilient Hosting of Research Data Management Services

Halima Saker [1], Simon Pirkl [1], Jens Krüger [1], Holger Gauza [1], Suvasini Thangaraj [1], Ursula Eberhardt [1], Alexander Kirbis [1], Lucas Beuter [2]

[1] High Performance and Cloud Computing Group at IT Center, Eberhard Karls University Tübingen;
[2] IT Division, University Library, Eberhard Karls Universität Tübingen

High Availability (HA) environments achieve automatic failover to prevent data loss and minimize downtime while ensuring stability. The cost-effective and flexible HA setup avoids technology lock-in through its use of open-source components and virtual machine clusters. The system relies on two main components, which are the Pacemaker and Distributed Replicated Block Devices (DRBD). DRBD maintains real-time data replication, which results in backup servers obtaining exact copies of the primary server data. The Pacemaker system functions as a cluster manager to handle server cluster problems by performing automatic failover operations. The fencing mechanism of this system blocks simultaneous data access to prevent data corruption while maintaining data integrity. The HA setup provides scalability features that enable research environments to handle increasing data volumes, user numbers, and workflow demands through server additions or configuration modifications.

Keywords: Data integrity, Reproducibility and reuse, Digital sovereignty, Open source, High availability, RDM, RDM Service hosting

1 Introduction

The growing usage of digital platforms in the scientific research process (e.g., Navale, Kaeppler, and McAuliffe 2021; Gomes, Queiroz, and Ferreira 2020) requires the establishment of durable and available hosting systems for research data management services.

Published in: Vincent Heuveline, Philipp Kling, Florian Heuschkel, Sophie G. Habinger, and Cora F. Krömer (Hrsg.): E-Science-Tage 2025. Research Data Management: Challenges in a Changing World. Heidelberg: heiBOOKS, 2025. DOI: https://doi.org/10.11588/heibooks.1652.c23926 (CC BY-SA 4.0).

Users of online platforms for research data management consider access to research data to be a must-have feature (Reichenbach, Eberl, and Lindenmeier 2022). Therefore, institutions must ensure that data repositories, collaboration tools, and analytical platforms remain accessible at all times as research becomes more dependent on data. For research continuity and reproducibility purposes and to guarantee that data stays preserved over time, system integrity and availability serve as critical requirements rather than simple conveniences. Service disruptions can create severe research setbacks through data loss while also halting collaborations and damaging research results. Modern scientific workflows require an infrastructure that provides both reliability and resilience.

The continuity of RDM services at scientific research institutions, along with universities and data centers, encounters numerous challenges. The operational difficulties faced by scientific research institutions include system failures that occur without warning, together with network disruptions, hardware malfunctions, and routine maintenance activities, which all lead to service downtimes. Research fields including bioinformatics, climate science, and high-energy physics demand uninterrupted data processing; Thus, system downtime of even a short duration can critically affect experimental outcomes and data integrity. The growing implementation of open science practices requires research data to remain available for validation and replication studies as well as secondary analysis (Wilkinson et al. 2016). Research institutions that lack proper high-availability (HA) infrastructure may find themselves out of compliance with both funding agency expectations and established data stewardship standards, which refer to ongoing organizational and technical practices that ensure research data remains FAIR (Findable, Accessible, Interoperable, Reusable) throughout its lifecycle.

HA systems equip organizations with fault tolerance capabilities and data replication processes along with automated failover solutions, which work together to reduce downtime and prevent data loss. High-availability solutions play a critical role in maintaining the security and accessibility of research data across long durations. With proper implementation, an HA system automatically identifies failures and redistributes workloads to operational nodes without human assistance. Research organizations that face limited resources benefit significantly from this functionality because it eliminates unnecessary administrative tasks and preserves continuous operational performance.

Academic research environments encounter specific challenges because they handle multiple data types while dealing with changing computational requirements and restricted financial resources. These conditions require HA solutions which must be both flexible and affordable. The existing commercial HA systems prove too expensive and restrictive for public research applications because they do not meet long-term needs. This paper introduces an open-source HA architecture which addresses the requirements of research data management (RDM) at scale. Our tested solution maintains data integrity and availability for essential workflows through the implementation of Pacemaker and Distributed Replicated Block Device (DRBD). The selected tools meet academic requirements because they offer reliable performance alongside open-source licensing and affordable pric-

ing which supports digital sovereignty and reproducibility and sustainability in academic environments.

A promising approach to achieve HA in RDM services is to use Pacemaker and DRBD (LINBIT 2024), which are both open-source technologies that are used in enterprises and research networks. Pacemaker acts as a cluster resource manager that checks the health of the system and controls failover if there is a hardware or software failure (ClusterLabs 2023). DRBD offers data replication in real time between different machines, meaning that the research data will remain coherent and available in the case of a failure of the primary node. Altogether, these technologies create a strong infrastructure that prevents the risks of system failures and data corruption and, at the same time, allows for the necessary expansion of computational capacity to support growing research needs.

In addition to the availability of the data, its integrity is of great significance in scientific organizations (Lagoze 2014). Any alteration or degradation of the research data resulting from faulty failover mechanisms or unmonitored access can lead to wrong conclusions being made, thus compromising the validity of the scientific study. HA systems that employ fencing techniques such as quorum-based mechanisms and STONITH (Shoot The Other Node In The Head) prevent split-brain scenarios where two or more nodes try to write to the same portion of the data set at the same time. These guarantees are crucial to avoid the situation where there are two different versions of the data and to guarantee that there is only one latest version of the data at any point in time.

2 Related Work

Research data management systems receive multiple studies which examine their ability to maintain high availability (HA) and data integrity (Mesbahi, Rahmani, and Hosseinzadeh 2018). Research workflows require continuous platform access so researchers need robust infrastructure solutions to support this requirement (Navale and Bourne 2021; Wilkinson et al. 2016). Researchers have implemented multiple methods to achieve fault tolerance and minimize platform downtime in distributed platforms.

The distributed storage systems Ceph (Weil et al. 2006) and GlusterFS (Hedges, Hasan, and Powell 2013) implement data replication for HA but create performance issues that prevent their practical use in institutions with limited resources. DRBD (LINBIT 2024; Reisner and Ellenberg 2005) enables real-time block-level replication, which eliminates the requirement for distributed file systems.

The foundation of HA depends on automatic failover systems together with failure recovery functions. System resilience requires both load balancing and redundancy to function effectively (Ramy and Arafat 2019; Li, Zhang, and Wang 2020). The widely implemented enterprise solution Pacemaker (ClusterLabs 2023) lacks research-based exploration in aca-

demic environments. The research demonstrates Pacemaker's success as a scientific data hosting solution after its integration with DRBD.

Data integrity maintenance stands on the same level of importance as system availability. Node conflicts during split-brain events result in dangerous data version inconsistencies (Brown and Peters 2018; Santos and Lee 2021). STONITH fencing together with quorum-based decision-making provides protection against these risks (Zhao and Kumar 2017). The implementation of these techniques remains widespread in enterprise applications yet research IT infrastructure fails to adopt them. This work implements such safeguards to protect research data.

The current state of HA research for research data hosting shows limited development because most studies focus on general-purpose configurations. This research contribution fills this knowledge deficiency by creating an affordable and expandable HA system designed specifically for research institutions. A GitLab HA deployment case study within the de.NBI Cloud demonstrates both implementation difficulties and optimal implementation procedures.

Our open-source HA solution provides sustainable data availability with high data integrity in scientific environments.

3 Importance of Reliable Research Data Hosting

HA research data hosting functions as a solution to prevent interruptions in continuous research progress. Organizations that manage extensive datasets need to develop scalable redundant architectures to stop data loss and corruption, according to (Leitner, Huhn, and Scherer 2018).

The implementation of HA systems through failover mechanisms enables the system to recover from hardware, software, or network failures, which ensures continuous access to essential research tools, particularly in fields that require large-scale data processing, such as genomics and large-scale simulations (Gomes, Queiroz, and Ferreira 2020). Pacemaker, alongside redundancy functions, eliminates single points of failure by distributing the workload across multiple nodes while it automates failover procedures to minimize downtime (ClusterLabs 2023). The system also provides encryption and authentication features together with access restrictions to protect research data (Smith 2017). The strict data management policies from funding bodies and accreditation entities require institutions to implement robust HA strategies for maintaining their institutional credibility (Wilkinson et al. 2016). The research data availability and security needs of institutions are supported by HA solutions including Pacemaker and DRBD to maintain data integrity (LINBIT 2024).

4 Core Concepts of the HA System

The implementation of an HA system requires a clear framework of redundancy, failover, and data integrity (Endo et al. 2016). In scientific computing, irreproducible results are unpleasant, and data loss or corruption can lead to such results, costing time and damaging the credibility of research findings. To this end, an effective HA system design includes automated recovery strategies, real-time replication, and controlled access to prevent such risks.

4.1 High-Availability (HA) Clustering

HA clustering is the process of linking together several servers in order to make them work as one system such that if some components fail, the service will not stop. The main benefit of the clustering is failover for the purpose of redundancy to ensure that the workloads are redirected to the stand by nodes in case of failure. In an HA cluster, a monitoring system is supposed to be constantly running to check on the health of the active nodes and raise an alarm in case of any risk of failure and initiate the failover process accordingly.

Clusters can be classified as active-passive or active-active. The active-passive cluster is one where the role of the primary node is to manage the workload, and the standby nodes are passive and do not participate in the processing of the workload unless the primary node is unavailable. On the other hand, an active-active cluster shares the workload across multiple nodes at the same time for improved efficiency and load balancing. The difference between these two configurations is primarily governed by system needs, the existing infrastructure, and the nature of the research workflows that the system is expected to support.

4.2 Data Replication Strategies

HA systems require research data to be available and intact, which makes data replication critical. DRBD is a technology used to replicate data at the block level between multiple nodes, and DRBD is used in sync with Pacemaker to manage failover and to fail over control to a secondary node in case of a failure to avoid downtime. Geographic redundancy can enhance data resilience by spreading the replicas of the data across different locations to safeguard against failures that are localized. Asynchronous replication can be employed in low sensitive tasks to decrease the load, but synchronous replication with DRBD is preferable for research as it provides real time data consistency. In summary, DRBD and Pacemaker do a good job of replicating data to keep research data available, consistent, and secure.

4.3 Data Integrity Mechanisms

In addition to ensuring that HA systems remain available, strict data integrity controls must be implemented to prevent corruption and unauthorized modifications of data (Duggineni 2023). The split-brain scenario, where two nodes make different changes without coordination, calls for fencing to force a controlled, secure way of accessing critical data resources.

- **Quorum-based fencing**: This ensures that only a subset of nodes with sufficient voting power can change the data. This approach prevents two simultaneous write operations that can lead to inconsistencies if they happen at different times.
- **STONITH**: A last-resort mechanism that kills unresponsive nodes to prevent the cluster from becoming unstable. STONITH is most important for preventing bad processes from corrupting the synchronized datasets in some way.
- **Version Control and Checkpointing**: Some HA configurations also include version control systems that enable the researcher to go back to a previous version of the dataset in case of errors. This feature improves the reproducibility and the audit trail of scientific workflows.

4.4 Open-Source Architecture

A major strength of the proposed HA solution is that it is based on open source technologies. Proprietary systems bring in licensing expenses and possible vendor capture, whereas open-source frameworks are flexible, modular, and openly developed. Pacemaker, DRBD, and Corosync are integrated together to help research organizations develop their own HA configurations that suit their computing requirements. Thus, open source HA architectures can be further developed and improved, for example, in terms of development, security, and compatibility with other infrastructure components. The research computing environments must be flexible to grow with the dynamic nature of the scientific research, to support growing data sets, complex workflows and large collaboratory groups.

5 Core Technologies

Our setup includes DRBD for data replication, and Pacemaker for failover automation. The technologies were selected based on their maturity, open-source nature, and strong community support. Pacemaker and DRBD are widely adopted in high-availability clusters and integrate well into Linux environments. Corosync, used alongside Pacemaker, ensures reliable cluster communication and quorum decisions.

5.1 Pacemaker

Pacemaker functions as an open-source cluster resource manager which supports HA environments. The system tracks continuous health status and performs failovers through resource management and node failure detection and workload redistribution. Pacemaker functions as a core component to maintain service continuity by performing automatic application failovers to backup nodes which is essential for research environments that need accessible computational workflows and storage. The system operates with different cluster configurations including active-passive and active-active modes to maximize system reliability. Pacemaker implements complex decision-making through node weights and quorum calculations and fencing mechanisms to stop data corruption during failovers. The system uses DRBD integration to enforce strict data synchronization rules for maintaining data consistency while Corosync provides real-time cluster messaging to enhance failover efficiency.

5.2 Distributed Replicated Block Device (DRBD)

DRBD is a software-based networked storage replication solution that enables real-time data mirroring between nodes in an HA cluster. It operates at the block level, ensuring that changes made to one system are immediately replicated to another, reducing the risk of data loss during failures.

In research environments, DRBD is particularly valuable for preserving the integrity of scientific datasets. Unlike traditional backup solutions that rely on periodic snapshots, DRBD ensures that all write operations are immediately copied to a secondary node. This synchronization can be performed in synchronous, asynchronous, or dual-primary modes, each offering a balance between performance and consistency.

- **Synchronous Mode**: Guarantees that all data is written to both nodes before acknowledging a transaction. This mode ensures maximum data consistency but can introduce latency.
- **Asynchronous Mode**: Writes data to the primary node first and replicates it to the secondary node afterward. This approach reduces latency but carries a risk of minimal data loss in the event of failure.
- **Dual-Primary Mode**: Allows simultaneous read and write operations on both nodes, enabling load balancing. However, this mode requires advanced fencing mechanisms to prevent data corruption.

DRBD is crucial for maintaining uninterrupted access to research data, supporting applications that require real-time data consistency. When paired with Pacemaker, it enhances the overall resilience of scientific computing environments by ensuring seamless failover and disaster recovery capabilities.

5.3 Corosync

The cluster communication engine Corosync functions to synchronize messages between HA system nodes while maintaining node communication (Dake, Caulfield, and Beekhof 2008). The system effectively distributes state changes to maintain Pacemaker's current status. Corosync enables leader election and member management and failure detection in research computing environments. Pacemaker depends on reliable multicast messaging to achieve synchronized status updates because this functionality is essential for its failover process. The quorum-based consensus mechanism in Corosync prevents split-brain scenarios which prevents data inconsistencies. The system provides encryption for node-to-node communication which strengthens security in research environments with multiple nodes. The complete HA stack consists of Pacemaker, DRBD and Corosync which provides research data hosting with failover functionality and real-time replication and cluster coordination.

6 Architecture Overview & Failover Process

The architecture of an HA system is designed to prevent downtime and guarantee access to crucial research data. A good HA infrastructure is based on several parts, such as primary and secondary servers, network arrangements, and failover procedures. In this section, the elements of these architectures will be discussed in detail, and how they are interlinked to enhance the reliability and data integrity of the system in research environments.

6.1 System Components

A well designed HA system is a set of servers that are interconnected and work together in order to provide failover and data accessibility continuity. The main elements of the architecture are:

- **Primary Server**: The working node that acts as a point of contact for requests and data processing in real time. It acts as the primary channel through which users and other applications can access the research data.
- **Secondary Servers**: A hot standby node that keeps data in sync with the primary server using DRBD. In case of a failure, one of the secondary servers can easily take over control without any downtime.
- **Backup Server**: Extra sets of research data are stored on this server to increase the level of data redundancy in case of several failures. This server is a failback contingency plan in case of data loss.

- **Floating IP**: A method of using a single IP address that can be easily moved between servers in case of a failover. It enables users and applications to keep their connection intact without having to change settings manually.
- **Reverse Proxy**: It directs the network traffic and distributes the load among the available servers to increase the performance of the application.
- **Object Storage (S3 Ceph)**: Offered storage systems for big data sets employed in computational research and data mining purposes (MinIO, Inc 2023).

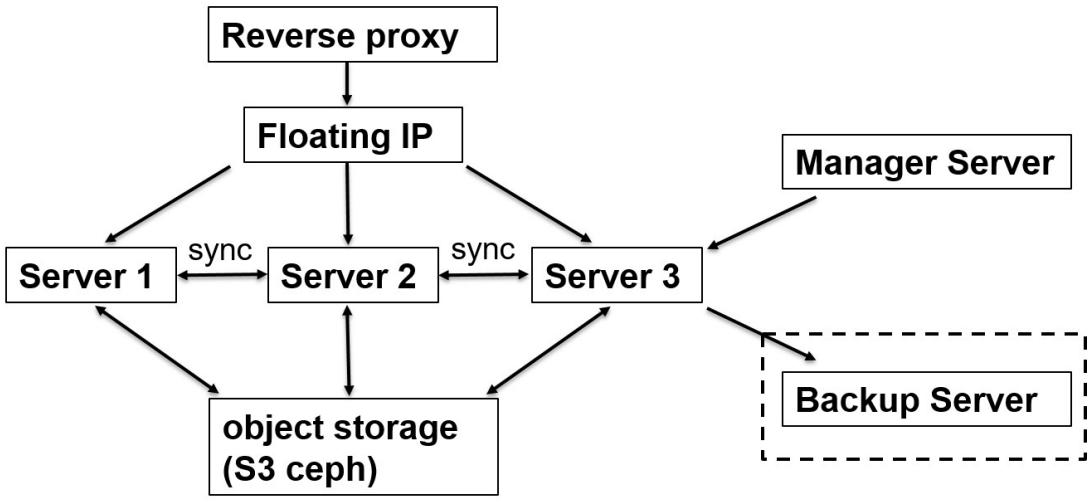

Figure 1: This diagram illustrates an HA system where multiple nodes are connected, ensuring redundancy and failover capabilities. A backup server is integrated to provide data protection and disaster recovery, enhancing system reliability and integrity.

6.2 Failover Mechanism

Failover is the crucial feature of an HA system that provides failure notification and subsequent failover of the primary server to a stand-by node without disrupting research. The failover process includes the following:

- **Failure Detection:** The Pacemaker is responsible for keeping a close eye on the primary server and its systems, including CPU load, memory consumption, and network connectivity. If the primary server is unavailable or dies, the Pacemaker initiates the failover process.
- **Resource Reallocation:** In case of failure, the Pacemaker shuts down the engaged node and makes the standby server the primary server. This ensures that all the data and applications are available to the users without any interruption.
- **Floating IP Reassignment:** The system reconfigures the floating IP address to the new primary server so that there is no break in communication for the researchers and other computational tools.

- **Data Synchronization:** First, DRBD checks that all the data on the new primary server is current and in sync with the last saved state to avoid data loss or corruption.
- **Fencing Activation:** To avoid the split-brain scenario, the Pacemaker starts the fencing mechanisms that deny the access to the critical storage resources to the active node.

Hence, the proposed model achieves a high level of availability by ensuring the continuous operation of the research environment through effective failover mechanisms.

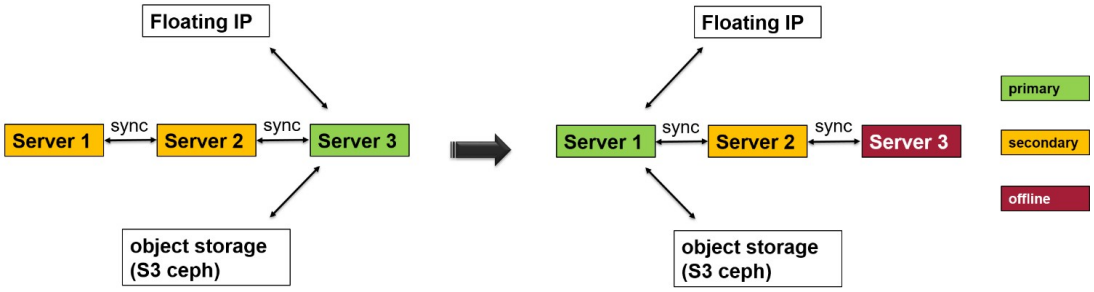

Figure 2: Failover Process.

6.3 Redundancy and Load Balancing

To increase the performance and robustness of the HA systems, the redundancy and load balancing mechanisms are incorporated. Data redundancy is ensured by DRBD, which is responsible for real-time replication of research data across multiple nodes to avoid a single point of failure. Load balancing is achieved as network requests are forwarded by the reverse proxy to active nodes, improving system performance and preventing the overload of a single server. Quorum-based decision-making is managed by Corosync, which makes failover decisions based on a majority consensus to prevent false positives and unnecessary transitions.

By integrating these architectural elements, research institutions can build a high-availability infrastructure capable of supporting intensive computational workloads while maintaining data integrity and security.

7 Network & Securing Integrity

Maintaining data integrity and network security is essential in HA research infrastructures. A good HA system not only provides failover but also protects research data from being corrupted, accessed inappropriately and cyber attacks. This section outlines the network

topology, fencing strategies, and security measures that are required to guarantee the data integrity of HA clusters.

Quorum-Based Fencing and Split-Brain Prevention: A major issue in HA designs is the avoidance of split-brain, where two nodes each take on the master role and try to push conflicting updates. Such incidents can lead to data errors, system failures, and possible degradation of research data. To address this issue, HA clusters use quorum-based fencing to prevent more than one node from accessing the shared storage at any given time (Jiménez-Peris et al. 2003).

Quorum mechanisms function by only allowing one node to take over control of the shared resources if it can get the support of the other nodes. When a node is isolated due to a network issue, quorum-based policies will stop it from making changes that could cause a mismatch. This ensures that only those nodes that are physically connected and actively participating can perform the read and write operations and thus avoid the situation of the cluster having two copies of data that are different.

Figure 3: Quorum-Based Fencing.

STONITH – Node Isolation: STONITH is a vital fencing mechanism that virtually shuts down failed or faulty nodes from the cluster to achieve data reliability (Robertson 2000). This approach entails powering cycle or forced shutdown of a node that is not responding properly within the cluster.

In this way, STONITH is used to prevent HA systems from being compromised by a node that is attempting to make changes to the shared storage while another node is operational. This ensures that even in case of software bugs, equipment failure, or network partitioning, the HA system remains functional and does not write data to the wrong location.

Secure Communication Among the Nodes: This paper highlights the importance of inter-node communication in HA clusters and the need to secure such communication against eavesdropping, tampering and unauthorized access. Today's HA systems encrypt their data traffic between the cluster nodes with TLS (Transport Layer Security).

Furthermore, role-based access control (RBAC) ensures that authentication is strictly enforced on who can change the configuration of the cluster. The use of multi-factor authentication (MFA) and public key infrastructure (PKI) improves the security by preventing

anyone from making changes to the HA environment without authenticating themselves properly.

Network Segmentation and Routine Auditing: HA clusters minimize attack risks because they operate through distinct networks which keep essential components in isolated subnets that block both external threats and lateral movement. Data integrity stands as an equally crucial factor in HA environments. The system uses checksum validation and automated integrity checks to synchronize data between primary and secondary nodes while scheduled integrity audits detect and correct potential errors or corruption.

When these security features are integrated into the RDM's HA infrastructure, it means that the infrastructure is secure from both internal and external threats that may cause damage to the integrity and confidentiality of research data.

8 Case Study: Implementation of GitLab HA Service in de.NBI Cloud

A highly available GitLab service by using HA clustering to achieve reliability, data integrity, and failover is offered on the de.NBI Cloud Tübingen. GitLab (GitLab B.V. 2024) is used extensively for research software development, version control, and collaboration, and so it becomes a critical platform in research data management that must be continuously available. To address these needs, the de.NBI GitLab HA setup incorporates several open-source technologies, such as Pacemaker, DRBD, Corosync, and S3-based storage replication.

High-Availability Architecture for GitLab: The GitLab HA cluster is running on several virtual machines (VMs), and the VMs are separated into different nodes of the architecture (Brandt et al. 2023). The main parts of the HA infrastructure are GitLab, DRBD, Corosync, and Pacemaker. GitLab provides repository hosting, CI/CD, and access control. DRBD guarantees the data synchronization between the primary and secondary nodes in real time to make sure that there is consistency. Corosync is used for failure detection and coordination between the cluster nodes and Pacemaker for managing the cluster resources, node failover, and fencing to achieve failover and availability in the system (e.g. GitLab B.V. 2023).

Data Replication and Backup Strategies: To prevent data loss and damage, the GitLab HA infrastructure has a strong multi-level backup and replication plan. The primary GitLab cluster is supposed to perform daily backups of the repositories, metadata,

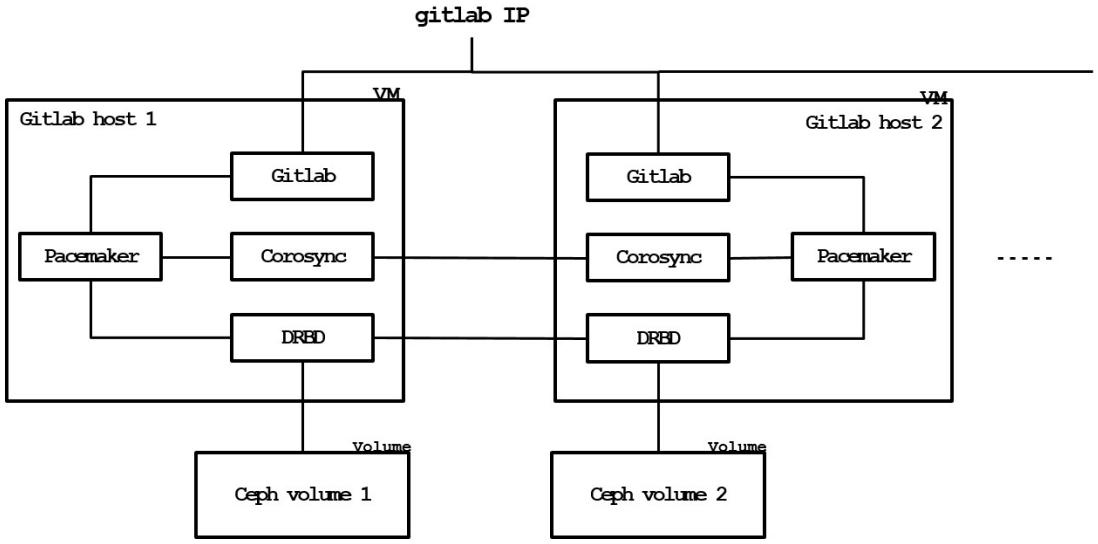

Figure 4: HA GitLab architecture designed with Pacemaker, Corosync, and DRBD for redundancy and failover, with Ceph storage for persistent data replication.

and application configurations. These backups are kept locally for a brief period of time to enable easy restoration. Furthermore, the backups are copied to an S3-based object storage system for long-term storage and a second S3 storage is used as a secondary copy. If the primary cluster fails, the last backup can easily be used to restore the secondary GitLab instance with minimal downtime and data loss.

Failover and Recovery Process: The failover and recovery process of the GitLab HA cluster is designed to avoid downtime and to maintain business process continuity. It is Corosync that keeps an eye on the health of each GitLab node, and in the event that the primary node becomes unresponsive, Pacemaker initiates the failover procedures. The GitLab secondary node is then able to take over the primary role, and the floating IP is moved to preserve the user's access. DRBD checks the data consistency and guarantees that the data is valid and ready to use on the new primary node. If required, the latest backup can be restored to the active node to include the most recent changes. When the failed node is returned online, it is first brought current with the active GitLab instance and then rejoined to the HA cluster.

Automation and Infrastructure-as-Code (IaC) Implementation: Manual configurations can lead to inconsistencies and involve a lot of administrative work. The use of Infrastructure-as-Code (IaC) tools like Ansible (Red Hat Inc. 2024) streamline the deployment processes, reproducibility, and reduce human error. Automated playbooks can be used to set up Pacemaker, Corosync, and DRBD based on predefined best practices. This enables the seamless configuration of GitLab HA instances and automatic integration of storage replication.

8.1 Real-World Application: GitLab and ARCmanager in NFDI4Plants

The resilient GitLab HA infrastructure described above is actively used to host the NFDI4Plants GitLab instance[1], which serves as the backend for the PLANTdataHUB. This instance enables researchers to store and manage their ARCs-structured, version-controlled collections of data and metadata.

In particular, the *ARCmanager* web service[2] connects seamlessly to this GitLab deployment. It allows users to create, browse, and edit ARCs directly in their browser while automatically syncing changes to the underlying Git repository. This setup offers a GUI-driven experience for researchers unfamiliar with Git workflows while maintaining the robustness and traceability of Git-based versioning. Importantly, thanks to the HA mechanisms (Pacemaker, DRBD, STONITH), the ARCmanager and its backing Git repositories remain continuously accessible and free from corruption even during server or network failures. As a result, plant science research projects that rely on these ARCs can proceed without disruption, ensuring transparency, traceability, and reproducibility.

9 Future Work & Next Steps

Although the application of HA systems with Pacemaker, DRBD, Corosync, and GitLab is very efficient in the preservation of service continuity in the de.NBI Cloud still has some possibilities for improvement and growth. As the research data management requirements are continuously developing, so does the infrastructure that supports them. This section highlights important aspects that can be improved further to increase the robustness, capacity, and security of HA deployments.

Expansion to Multi-Region High-Availability: Currently, the GitLab HA setup is currently running everything in a single cloud region. While intra-regional failover is good enough to cover most failures that are confined to a single region, adding multi-region replication would be another level of redundancy. This can be achieved by implementing geo-distributed DRBD clusters with asynchronous replication between regions, combined with multi-site Pacemaker clusters that include intelligent load balancing. Furthermore, deploying cross-region S3 storage replication will guarantee data consistency for backups, making the system even more reliable.

1 https://gitlab.nfdi4plants.de; *Visited by the editors on June 18, 2025.*
2 https://nfdi4plants.de/arcmanager/app/index.html; *Visited by the editors on June 18, 2025.*

Automated Deployment with Ansible Playbooks: To streamline the deployment and configuration of the HA cluster across various environments, we plan to develop and implement a comprehensive set of Ansible playbooks. This will reduce manual effort, ensure consistency, and enable rapid scaling and disaster recovery procedures.

Implementation of HA in Other Critical Services: Beyond GitLab, we aim to extend the high-availability setup to other essential research infrastructure services. This includes implementing HA for a reverse proxy (e.g., Nginx or HAProxy) to manage incoming traffic and distribute it across redundant service instances, as well as for identity and access management solutions like Keycloak. This will ensure end-to-end availability for the entire service stack.

Automated Scaling and Resource Optimization: Beyond failover, future work will investigate integrating dynamic scaling mechanisms with Pacemaker to automatically adjust computational resources based on real-time workload demands, particularly for bursty research computations. This would involve predictive analytics to anticipate resource needs and intelligent orchestration to optimize resource allocation, further enhancing efficiency and cost-effectiveness.

10 Conclusion

HA solutions are essential for ensuring the continuity of research data services. The growing dependence of research institutions on digital infrastructure makes protection against loss of data or corruption increasingly important. Pacemaker working with DRBD presents an economical and expandable open-source solution to address this problem. The HA setup provides research data accessibility by implementing node mirroring and fencing mechanisms to reduce corruption risks (LINBIT 2024; ClusterLabs 2023; Leitner, Huhn, and Scherer 2018). Research resources, including GitLab, maintain data integrity through this method, preventing simultaneous data access and ensuring data consistency (Wilkinson et al. 2016).

Research institutions can extend their infrastructure capacity by implementing this HA architecture through server additions or configuration modifications as their research needs expand. Extending this system is essential because it allows researchers to adapt their systems to the dynamic nature of their field. The utilization of open-source components eliminates vendor lock-in risks, which enables institutions to personalize their infrastructure needs while promoting collaborative development and community engagement (Sommestad and Karlsson 2014; Smith 2017).

The implementation of HA systems will likely spread throughout research institutions in the future. Research data management systems will gain increased resilience through automation tools for deployment and advanced monitoring and HA coverage extension to additional services (Bernstein and Smith 2019; Manerikar and Datta 2015). The continued growth of research data complexity requires sustained preservation of data integrity and availability to maintain scientific study reproducibility and trust in research outcomes. Research organizations can substantially benefit from HA solutions that use Pacemaker and DRBD to meet their present and future requirements while maintaining the dependability and accessibility of essential data (Navale and Bourne 2021; Gomes, Queiroz, and Ferreira 2020).

Acknowledgements

We acknowledge support for DataPLANT 442077441 through the German National Research Data Initiative (NFDI 7/1). We appreciate our co-fellows Dr.Adam Svahn, Mohamad Chehab, Amir Baleghi, and Fabian Paz for their valuable contributions to ensuring the High Availability of the Gitlab setup.

Authorship Contributions

Halima Saker drafted and wrote the manuscript. Simon Pirkl created and delivered the conference presentation. All authors reviewed and edited the manuscript.

Conflict of Interest

The authors declare that there are no competing interests or conflicts of interest related to the content or findings of this article.

Bibliography

Bernstein, David A., and Daniel J. Smith. 2019. "Automation in High-Availability Systems for Research Data". *Research Computing Systems Journal* 8:59–73. https://doi.org/10.1007/s00201-019-01168-7.

Brandt, Olaf, Holger Gauza, Jan Kaltenbach, Steve Kaminski, Jens Krüger, Fabian Paz, Saker Halima, Fabian Wannenmacher, Johannes Werner, and Thomas Zajac. 2023. *Distributed but Integrated: Forming a Science Gateway from Multiple Parts.* Zenodo. https://doi.org/10.5281/zenodo.7883191.

Brown, Alexander, and Daniel Peters. 2018. "Mitigating Split-Brain Scenarios in Distributed Systems: A Case Study". *IEEE Transactions on Distributed Systems* 29 (5): 1211–1223. https://doi.org/10.1109/TDS.2018.2814567.

ClusterLabs. 2023. *Pacemaker Documentation.* Visited on February 7, 2024. https://clusterlabs.org/pacemaker/doc/.

Dake, Steven C, Christine Caulfield, and Andrew Beekhof. 2008. "The corosync cluster engine". In *Linux Symposium,* 85:61–68. Citeseer.

Duggineni, Sasidhar. 2023. "Data integrity and risk". *Open Journal of Optimization* 12 (02). https://doi.org/10.4236/ojop.2023.122003.

Endo, Patricia T., Moisés Rodrigues, Glauco E. Gonçalves, Judith Kelner, Djamel H. Sadok, and Calin Curescu. 2016. "High availability in clouds: Systematic review and Research Challenges". *Journal of Cloud Computing* 5 (1). https://doi.org/10.1186/s13677-016-0066-8.

GitLab B.V.. 2023. *Reference architecture: up to 3,000 users | GitLab.* Https://docs.gitlab.com/ee/administration/reference_architectures/3k_users.html, Cited 22 Feb 2023.

———. 2024. *The most-comprehensive AI-powered DevSecOps platform | GitLab.* Visited on April 4, 2024. https://www.gitlab.com.

Gomes, Vitor C. F., Gilberto R. Queiroz, and Karine R. Ferreira. 2020. "An Overview of Platforms for Big Earth Observation Data Management and Analysis". *Remote Sensing* 12 (8). https://doi.org/10.3390/rs12081253.

Hedges, M. R., B. Hasan, and A. Powell. 2013. "A Performance Evaluation of the Gluster Parallel File System for Scientific Workloads". In *Proceedings of the International Conference on Data Management Technologies and Applications (DATA),* 82–92.

Jiménez-Peris, Ricardo, M. Patiño-Martínez, Gustavo Alonso, and Bettina Kemme. 2003. "Are quorums an alternative for data replication?" *ACM Transactions on Database Systems* (New York, NY, USA) 28 (3): 257–294. https://doi.org/10.1145/937598.937601.

Lagoze, Carl. 2014. "Big Data, data integrity, and the fracturing of the control zone". *Big Data & Society* 1 (2). https://doi.org/10.1177/2053951714558281.

Leitner, Philipp, Dietmar Huhn, and Klaus Scherer. 2018. "High-Availability Clusters in Cloud Computing Environments". *Journal of Cloud Computing: Advances, Systems and Applications* 7 (1): 1–16. https://doi.org/10.1186/s13677-018-0116-9.

Li, Yuan, Wei Zhang, and Hongbo Wang. 2020. “A Load Balancing Strategy for Cloud-Based High-Availability Systems”. *IEEE Transactions on Cloud Computing* 8 (3): 754–766. https://doi.org/10.1109/TCC.2019.2928371.

LINBIT. 2024. *DRBD.* Visited on February 7, 2024. https://linbit.com/drbd/.

Manerikar, Ashwin, and Rajesh Datta. 2015. “Improving Resilience in Research Data Management through Automation”. *Journal of Computational Research and Data Management* 6 (2): 45–60. https://doi.org/10.1145/2832578.

Mesbahi, Mohammad Reza, Amir Masoud Rahmani, and Mehdi Hosseinzadeh. 2018. “Reliability and high availability in cloud computing environments: a reference roadmap”. *Human-centric Computing and Information Sciences* 8 (1): 20. https://doi.org/10.1186/s13673-018-0143-8.

MinIO, Inc. 2023. *MinIO | High Performance, Kubernetes Native Object Storage.* Visited on March 19, 2023. https://min.io.

Navale, Vivek, and Philip E. Bourne. 2021. “Cloud computing applications for biomedical research”. *Methods in Molecular Biology* 2190:3–26. https://doi.org/10.1007/978-1-0716-0826-5_1.

Navale, Vivek, Denis von Kaeppler, and Matthew McAuliffe. 2021. “An overview of biomedical platforms for managing research data”. *Journal of Data, Information and Management* 3 (1): 21–27. https://doi.org/10.1007/s42488-020-00040-0.

Ramy, Mohamed, and Ahmed Arafat. 2019. “High-Availability and Load Balancing in Cloud Computing: Challenges and Solutions”. *International Journal of Cloud Computing* 9 (2): 112–127.

Red Hat Inc. 2024. *Ansible. Automation for everyone.* Visited on February 8, 2024. https://www.ansible.com/.

Reichenbach, Rebecca, Christoph Eberl, and Jörg Lindenmeier. 2022. “Online platforms for research data: A requirements and cost analysis”. *Science and Public Policy* 49 (4): 598–608. https://doi.org/10.1093/scipol/scac011.

Reisner, Philipp, and Lars Ellenberg. 2005. “DRBD: Distributed Replicated Block Device”. *Linux Journal* 2005 (145): 18–23.

Robertson, Alan. 2000. “Resource Fencing Using STONITH”. In *Proceedings of the High-Availability Linux Project.* Broomfield, Colorado: IBM Linux Technology Center.

Santos, Roberto, and Kevin Lee. 2021. “Ensuring Data Integrity in High-Availability Cloud Systems”. *Journal of Cloud Security* 6 (3): 221–237.

Smith, George R. 2017. “Benefits of Open-Source Software in High-Availability Research Data Management”. *Journal of Open Research Software* 5:1–7. https://doi.org/10.5334/jors.188.

Sommestad, Tomas, and Jonny Karlsson. 2014. “The Role of Open-Source Software in High Availability Systems”. *International Journal of Computer Science and Information Security* 12 (5): 18–25.

Weil, Sage A., Andrew W. Leung, Scott A. Brandt, and Ethan L. Miller. 2006. *Ceph: A Scalable, High-Performance Distributed File System.* 307–320. USENIX Association.

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersbergand, Gabrielle Appleton, Myles Axtonand, Arie Baakand, Niklas Blombergand, et al. 2016. “The FAIR Guiding Principles for scientific data management and stewardship”. *Scientific data* 3 (1): 1–9. https://doi.org/10.1038/sdata.2016.18.

Zhao, Wei, and Rajesh Kumar. 2017. “Fencing Mechanisms in Cloud-Based HA Clusters: STONITH and Beyond”. In *Proceedings of the IEEE International Conference on Cloud Computing Technology,* 150–158.