A Campus Petabyte: Building CephRDS for Researchers and the NRP

How an NSF-funded 3.2 PB object store was built,

split between campus research and the National Research Platform, and what is still unproven

Charles E. Forsyth III
Research Computing
University of California, Riverside
Riverside, California, USA
charles.forsyth@ucr.edu

Abstract

CephRDS is the research object store UC Riverside (UCR) built with National Science Foundation (NSF) Campus Cyberinfrastructure award OAC-2346636. The award PI (principal investigator) is Michael Kennedy; I am a co-PI, and the build was team work. Its 3.2 PB of raw disk yields 2,090.88 TB usable under 6+3 erasure coding. It is split two ways: 422.40 TB pledged to the National Research Platform (NRP), a national research computing federation, and 1,668.48 TB kept as a UCR pool. Researchers reach it over the Amazon Simple Storage Service (S3) protocol from the campus network or the campus virtual private network (VPN). Pilot labs hold about 560 TB of allocations.

The hardware was delivered as planned; the operational questions took longer: who manages which part of the capacity, how researchers get in, and how the campus pays for it after the grant. The team dropped a proposed self-service portal for split management with a fixed boundary around the NRP pledge. The recharge rates are still being finalized, and I have no on-campus performance baseline yet.

CCS Concepts

• **Information systems** → **Distributed storage**; • **Computer systems organization** → *Cloud computing*.

Keywords

Ceph, object storage, erasure coding, research data storage, campus cyberinfrastructure, National Research Platform, S3, cost recovery

Author note

The work was directed by me and carried out with AI agents under written briefs, independent checks and explicit approval steps. Figures come from the NSF award record and its project outcomes report, the `rds-cli` repository and its test run, the CephRDS technical report, the Research Computing knowledge base, and my capacity and allocation records. This note was drafted with AI assistance from those records and reviewed by me. No representative values are used: system figures are real, and lab names and lab-level figures are omitted. The web edition is listed in the Field Notes index at https://charles-forsyth.github.io/papers/.

Cite this note. C. E. Forsyth III. 2026. A Campus Petabyte: Building CephRDS for Researchers and the NRP. Field Notes.
doi:10.5281/zenodo.23752269

1 Context

Before CephRDS, much UCR research data lived wherever researchers could find room: aging network drives, USB disks and personal cloud accounts. The project outcomes report in the NSF award record describes the same picture [1]. None of those places is easy to share across a lab, and none is fast from a cluster. Labs producing imaging, sequencing or sensor data hit the problem first, because their datasets outgrow a laptop in weeks.

The Campus Cyberinfrastructure program buys equipment [2]. It does not pay to run it afterwards, and it asks campuses to connect to national resources. The team wrote both into the design from the start. The storage had to be cheap enough to run on recharge. A postdoc had to be able to use it from a script. And part of it had to be shared with researchers outside UCR through NRP. It serves UCR faculty, postdocs and students with active project data, and, through the NRP slice, researchers elsewhere.

It follows two earlier Field Notes about NRP: *A Road to Nautilus in One Day* [3], on the `nrp-mcp` tool, and *One Day on the Research Stack* [4].

2 Related Work

Ceph began as a research file system at UC Santa Cruz [5]. Two ideas from that work matter here. CRUSH computes where data lives from a cluster map instead of looking it up in a central table [6], and RADOS lets the storage daemons replicate, detect failures and recover among themselves [7]. Together they are why a small team can run 20 commodity servers as one store. A decade of production experience later moved Ceph off general-purpose local file systems onto its own BlueStore backend [8], the default storage backend in current Ceph releases.

Erasure coding rests on Reed–Solomon codes [9]. Weatherspoon and Kubiatowicz showed early that, for the same durability, erasure codes need far less storage and bandwidth than replication [10]. Large operators then made the trade-off concrete. Windows Azure Storage adopted erasure coding and designed new codes to cut the reads needed for reconstruction [11], and work at Facebook on Hadoop clusters quantified how repair traffic grows under Reed–Solomon codes and proposed locally repairable codes to reduce it [12]. The Hadoop Distributed File System itself was designed around three-way replication [13], the baseline CephRDS chose against. The repair cost those papers measure is the cost I describe in Section 3.

On the network side, the Science DMZ pattern separates research data transfer from the general-purpose campus network so large flows are not throttled by enterprise security appliances [14]. The Pacific Research Platform built on that pattern to link campus Science DMZs into a regional high-speed fabric [15], and it grew into the National Research Platform that receives the pledged slice [16]. For moving data in and out of such stores, Globus is the usual research tool: a hosted service for transfer, synchronization and sharing between endpoints [17]. CephRDS does not yet offer a Globus endpoint; access is S3 only (Section 5). Federation of campus resources has a longer history in the Open Science Grid [18], whose StashCache work built a caching federation for data across sites [19]. That line continues today as the Open Science Data Federation (OSDF) [20]. The Open Storage Network takes a different approach, with NSF-funded storage pods at sites across the country [21].

The governance questions are older than any of this hardware. The NIST definition of cloud computing names on-demand self-service, resource pooling and measured service as essential characteristics [22], and Armbrust and colleagues argued that elasticity and pay-as-you-go pricing are what make cloud economics work [23]. A campus store with fixed capacity and a fixed budget gets some of those benefits and not others, which is part of why the self-service portal was dropped. Finally, S3 is defined by Amazon's API reference [24]; Ceph's object gateway implements a RESTful API compatible with the basic data access model of that API [25], and that compatibility is what lets existing S3 tools work against CephRDS.

I found no peer-reviewed account of a CC* storage award run past its grant period on recharge, though I did not search exhaustively. This note is an operator's report, not a controlled study.

3 The Design

The award abstract fixes the hardware [1]: 20 nodes, each with 160 TB of physical storage on 8 hard disk drives (HDDs), plus 3 Non-Volatile Memory Express (NVMe) solid-state drives (SSDs) giving 23 TB of fast storage per node, or 460 TB of flash across the cluster. The 3.2 PB raw figure is 20 times 160 TB. The outcomes report places the servers in the university's core data centers, connected to UCR's 100 Gbps Science DMZ (demilitarized zone), the campus network segment built for large research transfers.

The storage software is Ceph, the open-source distributed storage system. It runs on commodity servers, so the campus is not tied to one vendor's price for the next expansion. It can serve object, block and file interfaces from one cluster, which let us wait and see what researchers used. And NRP's own Nautilus cluster runs Ceph, so the people we would share capacity with already knew it.

The biggest design choice was erasure coding instead of three-way replication. A 6+3 code splits each object into 6 data chunks and 3 parity chunks on different servers. The overhead is 1.5 times instead of 3, so the same disks hold about twice as much. For a campus that has to recover cost per terabyte, that ratio decides whether the service is affordable at all.

Any 6 of the 9 chunks rebuild the object, so data survives the loss of 3. Availability is different. Ceph's documentation recommends a `min_size` of $k + 1$ for an erasure-coded pool, which is 7 here [26].

Table 1: CephRDS capacity by pool, from my capacity and allocation records. Percentages are of usable capacity, rounded to the nearest whole percent. The pilot figure is the sum of allocated quotas, not bytes stored.

Item	Capacity (TB)	Share of usable
Raw (20 servers × 160 TB)	3,200	n/a
Usable after 6+3 erasure coding[a]	2,090.88	100%
NRP slice (pledged)	422.40	20%
UCR pool	1,668.48	80%
Pilot-lab quotas, approx.[b]	560	27%

[a]Reported publicly as 2.09 PB. [b]Approximate; about a third of the UCR pool.

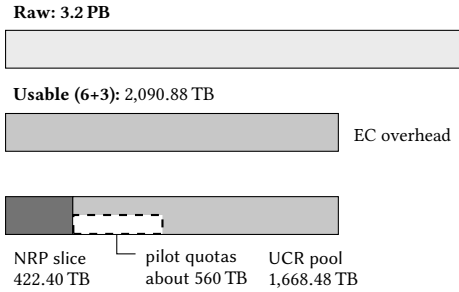


Figure 1: CephRDS capacity drawn to scale. Erasure coding turns 3.2 PB of raw disk into 2,090.88 TB usable; a fixed 422.40 TB slice is pledged to the NRP and the rest is the UCR pool, of which pilot quotas (dashed) claim about a third.

Lose more than 2 chunks of a placement group and that group stops serving reads and writes until recovery restores a seventh chunk, even though no data is lost. I describe the recommended setting here. Each object also needs 9 failure domains to place its chunks. With 20 servers there is room, but not a lot of it. Rebuilds cost more than copying a replica: restoring a lost chunk means reading 6 surviving chunks from other servers.

The flash and the disks became two classes of bucket. Standard buckets sit on hard disks and are the default. Fast buckets sit on the SSDs and are meant for small-file and metadata-heavy work. A lab can keep a sequencing archive and an active analysis directory side by side without paying flash prices for both.

4 Splitting the Capacity

About 20 percent of usable space is pledged to the NRP. Table 1 gives the figures from my capacity records, and Figure 1 draws them to scale. Raw capacity is what the vendor quote says; usable capacity is what a researcher can fill, and it is the number I plan against.

The two pools add up exactly: 422.40 TB plus 1,668.48 TB is 2,090.88 TB. The NRP slice matches the 20 percent stated in the NSF outcomes report [1].

A sharp reader will divide 3,200 TB by 1.5 and get about 2,133 TB, not 2,090.88 TB. The gap is about 42 TB, or 2 percent. Likely causes include decimal versus binary units, disk formatting and Ceph's

own overhead, but the records I drew on do not say which. I leave it unexplained rather than guess.

Table 1 also allocates every usable terabyte on paper, and Ceph does not let you fill every terabyte. By default a cluster warns at 85 percent full (`nearfull`) and stops accepting writes at 95 percent (`full`) [27]. It also needs free space to re-protect data after a server fails. One server is 5 percent of raw capacity, so at least that much has to stay empty. The space that can actually be sold is well below 2,090.88 TB. My records do not yet set a sellable ceiling, and at 560 TB allocated the question is not urgent, but the plan has to answer it before the UCR pool fills.

The original proposal imagined a single web portal where every user, campus or national, would sign in and claim storage. The team dropped it, and the outcomes report records why. High-volume research work runs on scripts, pipelines and cluster jobs that need keys and an endpoint, not a browser login. A fully self-service portal would also let storage run out without oversight. What replaced it is what the report calls split management. UCR manages the campus pool directly, sets a quota on each bucket and reviews each request. The NRP slice is a fixed pledge, so national use cannot spill into the campus pool, and campus growth cannot quietly eat the national commitment.

A fixed boundary is less flexible than a shared pool, and it wastes space when one side is full and the other is not. I accepted that cost to make the promises to both sides checkable. When a dean asks how much UCR storage is in use, the answer does not depend on what a researcher in another state did last week.

This note does not describe how NRP users reach the pledged slice. The outcomes report states the pledge and its size. The records I used for this note do not document the data path, credentials or operations split for that slice, so I do not describe them. A later note will, once I can write it from records.

The two sides carry different data rules. The campus pool accepts data at UCR protection levels P1 and P2. NRP states that its systems are not suitable for protected data, and the Research Computing guide to NRP storage applies that as a rule: only P1 (public) data may go on NRP resources. Lab requests are classified before storage is created for that reason.

5 How Researchers Get In

Access is S3 only. A lab gets a bucket, a quota and an access key pair, and connects to https://rds.ucr.edu with path-style addressing, where the bucket name goes in the path instead of the hostname. The service offers no Network File System (NFS) or Server Message Block (SMB) mount and no lifecycle (automatic deletion) rules. The endpoint is reachable only from the campus network or the campus VPN.

These limits were chosen deliberately. S3 alone means one protocol to secure, monitor and support, and it is the protocol analysis languages, cloud tools and pipeline frameworks already speak. Keeping the endpoint off the public internet removes a large class of attacks on a new service run by a small team, at the cost of making off-campus collaborators use the VPN. Leaving out lifecycle rules means no lab loses data to a rule someone set and forgot.

For researchers who do not want to learn S3, I directed the build of `rds-cli`, a small open-source command-line tool that already knows the endpoint and the path-style setting. It started on 13 March 2026, reached v0.1.13 on 28 May 2026, and moved to the UCR Research Computing GitHub organization on 28 October 2026. It uploads 10 files at a time, sends large files in 5 MB multipart chunks, logs a failed file and carries on with the rest, and stores keys with owner-only permissions. Its 20 unit tests all passed when I ran them on 28 October 2026. Those tests mock the storage and were not run against the cluster.

The Research Computing knowledge base (KB) carries the client guides: onboarding and keys (KB013), Python, desktop clients, mounting buckets with rclone, `rds-cli` (KB032) and other languages and tools (KB033). KB032 and KB033 went live on 28 October 2026, the day the tool moved, without being run against `rds.ucr.edu`; I had no VPN access that day. Both pages say so in their review notes, with a list of checks to run.

6 What Has Been Measured

The cluster is in pilot production. Pilot labs hold about 560 TB of allocations between them, about a third of the UCR pool. I give the total only.

The public outcomes report, written by the award PI, describes what pilot labs saw after moving onto CephRDS [1]. A neuroscience lab nearly doubled its transfer speed for large magnetic resonance imaging (MRI) datasets. A computer vision lab cut the time to load one large video dataset from 8–12 h to about 20 min. A geospatial lab moved 4 TB of satellite imagery without a failure after previously being unable to transfer it reliably. These are before-and-after observations by the labs, not controlled benchmarks, and the "before" in each case was a different and often much slower storage system.

My own measurements are not a cluster benchmark. On 13 March 2026 I ran the benchmark suite in the `cephrds-connect` repository from one remote client over a satellite internet link and the campus VPN, with single runs and a 20 MB test file. Table 2 lists the results. Uploads ran at about 0.6 MB/s and downloads at 1.4–2.5 MB/s, figures that mostly reflect the satellite link. The one operation that never left the data center, a server-side copy from the fast tier to the standard tier, ran at 36.73 MB/s. Multipart upload was slower than a single stream on the fast tier (0.32 against 0.58 MB/s). With 5 MB parts, a 20 MB file is only 4 parts, so per-request latency over a satellite link likely dominates; it tells me little about multipart on campus. The technical report's own summary says multipart overcomes satellite latency, which its data do not show.

I do not have a benchmark from a machine on the Science DMZ in my records. Until there is one, all I can say is that the cluster sits on a 100 Gbps network and that I have not measured what a campus client gets from it.

7 Paying for It After the Grant

An equipment award buys disks once. It does not buy replacement drives, support contracts, power, staff time or the next expansion. A campus storage service that cannot cover those costs ends up as an unsupported cluster with a lot of data on it. I designed a recharge model for CephRDS while the award was running so that the pilot would test the model and not just the hardware.

Table 2: Remote-client measurements of 13 March 2026: one client over a satellite link and the campus VPN, single runs, 20 MB file. Not a cluster benchmark.

Operation	Tier	MB/s
Upload, single stream	SSD	0.58
Upload, single stream	HDD	0.60
Upload, multipart	SSD	0.32
Upload, multipart	HDD	0.66
Download	SSD	2.48
Download	HDD	1.44
Server-side copy	SSD to HDD	36.73

Its structure follows the design. Rates are per usable terabyte, because that is what a lab fills, and erasure coding is what makes usable capacity cheap enough to sell. Standard and fast buckets carry different rates because flash costs more per terabyte than disk. The NRP slice sits outside the recharge model: it was committed to NRP under the award and is not sold to campus users.

The rates are being finalized through campus review. The service page says the same: under review, not yet approved. Pilot labs were told that terms would be set when their storage was allocated, and that joining the pilot does not lock in a price. That costs us some users. A lab that cannot put a storage price in a grant budget will not move irreplaceable data onto it, and I would not either.

8 Other Ways to Do This

Two national efforts solve parts of the same problem, and a peer will ask why UCR did not simply use them.

The Open Storage Network (OSN) is an NSF-funded network of petabyte storage pods at sites across the country, with allocations open to researchers [21]. OSN is built for sharing data between institutions. CephRDS is built first for active campus data under campus data rules, with a slice set aside for national use.

The Open Science Data Federation (OSDF), run by the OSG Consortium (formerly Open Science Grid) with software from the Pelican Platform, is the standard way to put campus storage into a national data network [20]. An "origin" service connects an object store, including an S3 endpoint, to the federation, and OSG offers a hosted origin for sites that want one. The OSDF page notes that Campus Cyberinfrastructure storage projects have committed to having their storage managed by OSG. Whether the NRP slice should be exposed this way is part of the federation write-up I have not done yet.

9 What Went Wrong

Failures, gaps and open risks as of 28 October 2026, most serious first.

- **There is no on-campus performance baseline.** The only throughput figures I have were taken over a satellite link. The lab outcomes are encouraging, but they are not measurements I can reproduce.
- **The NRP slice is pledged but not documented here.** I cannot yet describe from records how researchers outside UCR reach it, so a fifth of the system is outside this note.
- **The rates are not final.** The grant closed on 28 February 2026, and the recharge rates are still in campus review. Pilot labs carry uncertainty I wanted settled before they arrived.
- **Usable is not sellable.** Table 1 allocates all 2,090.88 TB, while Ceph's full ratios and failure headroom mean much less can be sold. The ceiling is not yet set.
- **The client documentation is untested.** KB032 and KB033 went live without a command run against the real endpoint. They say so, but they may still be wrong, and the checksum advice in KB033 is not yet verified for our gateway.
- **The VPN is a single dependency.** The restriction that protects the endpoint kept me from testing my own service for a day. Off-campus collaborators carry that dependency every time.
- **The bucket quota display is unconfirmed.** `rds-cli bucket-info` reads the quota from a response header Ceph's gateway can send. Whether ours sends it is not confirmed, and the tool silently skips the quota line when it is missing. A researcher could read that as "no quota".
- **The portal failed as a design.** The replacement works, but every bucket and key now goes through Research Computing by hand. That does not scale past a modest number of labs without better tooling.
- **Backup is not a feature of the service.** Erasure coding protects against hardware failure, not against a lab deleting its own data. Whether an allocation is backed up is part of its terms, and the service page tells researchers to ask before treating CephRDS as the only copy, and a lab that skips that step may hold its only copy here.

10 Lessons

(1) **Plan in sellable terabytes, not raw or usable.** Every promise is made in space that can be filled while the cluster can still heal itself.
(2) **Draw a hard line around anything you share with a federation.** A fixed pledge keeps both the national commitment and the campus pool checkable.
(3) **Give researchers keys and an endpoint before a portal.** Most research data moves through scripts and pipelines.
(4) **Design cost recovery while the grant is still paying.** The pilot is the cheapest chance to test whether labs will pay.
(5) **Benchmark from where the data is before you publish numbers from anywhere else.** My only numbers are from a satellite link, and that is not where most users are.

11 What's Next

- Take a throughput and small-file baseline from a client on the Science DMZ, on both tiers.
- Run every command in KB032 and KB033 against a real bucket, then clear the review notes or fix the pages.
- Set a sellable ceiling for the UCR pool that leaves room for Ceph's full ratios and a server failure.
- Confirm whether the gateway returns the bucket quota header, and make `rds-cli` say "quota unknown" instead of nothing.

- Publish the recharge rates on the service page once campus review approves them.
- Write up how the NRP slice is exposed and used, from records, and compare it with an OSDF origin.

Sources

Public: NSF award OAC-2346636 and its outcomes report; the CephRDS service page and knowledge base articles KB013, KB027, KB032 and KB033, at https://ucr-research-computing.github.io/; and the `rds-cli` repository, at https://github.com/UCR-Research-Computing/rds-cli. Private records used: my CephRDS capacity and allocation records (pool sizes, pilot total); the CephRDS technical report of 13 March 2026 in the `cephrds-connect` repository; the `rds-cli` git history and the test run of 28 October 2026; the review notes in the Research Computing knowledge base source for KB013, KB018 to KB020, KB027, KB032 and KB033; and the `nrp-nautilus` working specification.

References

[1] U.S. National Science Foundation. 2024. Award #2346636: Research Infrastructure: CC* Data Storage: Foundational Campus Research Storage for Digital Transformation. NSF Award Search, award abstract and Project Outcomes Report. https://www.nsf.gov/awardsearch/showAward?AWD_ID=2346636 Accessed 28 October 2026.

[2] U.S. National Science Foundation. 2024. Campus Cyberinfrastructure (CC*). Funding opportunity, program guidelines NSF 24-530. https://new.nsf.gov/funding/opportunities/cc-campus-cyberinfrastructure Accessed 28 October 2026.

[3] Charles E. Forsyth III. 2026. A Road to Nautilus in One Day. Field Notes. https://charles-forsyth.github.io/papers/2026-10-06-a-road-to-nautilus-in-one-day.html

[4] Charles E. Forsyth III. 2026. One Day on the Research Stack. Field Notes. https://charles-forsyth.github.io/papers/2026-10-03-one-day-on-the-research-stack.html

[5] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Carlos Maltzahn. 2006. Ceph: A Scalable, High-Performance Distributed File System. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06)*. USENIX Association, Berkeley, CA, USA. https://www.usenix.org/conference/osdi-06/ceph-scalable-high-performance-distributed-file-system

[6] Sage Weil, Scott Brandt, Ethan Miller, and Carlos Maltzahn. 2006. CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data. In *ACM/IEEE SC 2006 Conference (SC'06)*. IEEE, Piscataway, NJ, USA, 31. doi:10.1109/SC.2006.19

[7] Sage A. Weil, Andrew W. Leung, Scott A. Brandt, and Carlos Maltzahn. 2007. RADOS: A Scalable, Reliable Storage Service for Petabyte-Scale Storage Clusters. In *Proceedings of the 2nd International Workshop on Petascale Data Storage, held in conjunction with Supercomputing '07*. ACM, New York, NY, USA, 35–44. doi:10.1145/1374596.1374606

[8] Abutalib Aghayev, Sage Weil, Michael Kuchnik, Mark Nelson, Gregory R. Ganger, and George Amvrosiadis. 2019. File Systems Unfit as Distributed Storage Backends: Lessons from 10 Years of Ceph Evolution. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles (SOSP '19)*. ACM, New York, NY, USA, 353–369. doi:10.1145/3341301.3359656

[9] I. S. Reed and G. Solomon. 1960. Polynomial Codes Over Certain Finite Fields. *J. Soc. Indust. Appl. Math.* 8, 2 (1960), 300–304. doi:10.1137/0108018

[10] Hakim Weatherspoon and John D. Kubiatowicz. 2002. Erasure Coding Vs. Replication: A Quantitative Comparison. In *Peer-to-Peer Systems (IPTPS 2002) (Lecture Notes in Computer Science, Vol. 2429)*. Springer, Berlin, Heidelberg, 328–337. doi:10.1007/3-540-45748-8_31

[11] Cheng Huang, Huseyin Simitci, Yikang Xu, Aaron Ogus, Brad Calder, Parikshit Gopalan, Jin Li, and Sergey Yekhanin. 2012. Erasure Coding in Windows Azure Storage. In *Proceedings of the 2012 USENIX Annual Technical Conference (USENIX ATC '12)*. USENIX Association, Berkeley, CA, USA, 15–26. https://www.usenix.org/conference/atc12/technical-sessions/presentation/huang

[12] Maheswaran Sathiamoorthy, Megasthenis Asteris, Dimitris Papailiopoulos, Alexandros G. Dimakis, Ramkumar Vadali, Scott Chen, and Dhruba Borthakur. 2013. XORing Elephants: Novel Erasure Codes for Big Data. *Proceedings of the VLDB Endowment* 6, 5 (2013), 325–336. doi:10.14778/2535573.2488339

[13] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. The Hadoop Distributed File System. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, Piscataway, NJ, USA, 1–10. doi:10.1109/MSST.2010.5496972

[14] Eli Dart, Lauren Rotman, Brian Tierney, Mary Hester, and Jason Zurawski. 2013. The Science DMZ: A Network Design Pattern for Data-Intensive Science. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC '13)*. ACM, New York, NY, USA, 1–10. doi:10.1145/2503210.2503245

[15] Larry Smarr, Camille Crittenden, Thomas DeFanti, John Graham, Dmitry Mishin, Richard Moore, Philip Papadopoulos, and Frank Würthwein. 2018. The Pacific Research Platform: Making High-Speed Networking a Reality for the Scientist. In *Proceedings of the Practice and Experience on Advanced Research Computing (PEARC '18)*. ACM, New York, NY, USA, 1–8. doi:10.1145/3219104.3219108

[16] National Research Platform. 2026. National Research Platform. Project website. https://nationalresearchplatform.org/ Accessed 28 October 2026.

[17] Kyle Chard, Steven Tuecke, and Ian Foster. 2014. Efficient and Secure Transfer, Synchronization, and Sharing of Big Data. *IEEE Cloud Computing* 1, 3 (2014), 46–55. doi:10.1109/MCC.2014.52

[18] Ruth Pordes, Don Petravick, Bill Kramer, Doug Olson, Miron Livny, Alain Roy, Paul Avery, Kent Blackburn, Torre Wenaus, Frank Würthwein, Ian Foster, Rob Gardner, Mike Wilde, Alan Blatecky, John McGee, and Rob Quick. 2007. The Open Science Grid. *Journal of Physics: Conference Series* 78 (2007), 012057. doi:10.1088/1742-6596/78/1/012057

[19] Derek Weitzel, Marian Zvada, Ilija Vukotic, Rob Gardner, Brian Bockelman, Mats Rynge, Edgar Fajardo Hernandez, Brian Lin, and Mátyás Selmeci. 2019. StashCache: A Distributed Caching Federation for the Open Science Grid. In *Proceedings of the Practice and Experience in Advanced Research Computing (PEARC '19)*. ACM, New York, NY, USA, 1–7. doi:10.1145/3332186.3332212

[20] OSG Consortium. 2026. Open Science Data Federation (OSDF). OSG services documentation. https://osg-htc.org/services/osdf/ Accessed 28 October 2026.

[21] Open Storage Network. 2026. Open Storage Network. Project website. https://www.openstoragenetwork.org/ Accessed 28 October 2026.

[22] Peter Mell and Timothy Grance. 2011. *The NIST Definition of Cloud Computing*. Special Publication 800-145. National Institute of Standards and Technology. doi:10.6028/NIST.SP.800-145

[23] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A View of Cloud Computing. *Commun. ACM* 53, 4 (2010), 50–58. doi:10.1145/1721654.1721672

[24] Amazon Web Services. 2026. Amazon Simple Storage Service API Reference. AWS Documentation. https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html Accessed 28 October 2026.

[25] Ceph Project. 2026. Ceph Object Gateway S3 API. Ceph Documentation. https://docs.ceph.com/en/latest/radosgw/s3/ Accessed 28 October 2026.

[26] Ceph Project. 2026. Erasure Code. Ceph Documentation. https://docs.ceph.com/en/latest/rados/operations/erasure-code/ Accessed 28 October 2026.

[27] Ceph Project. 2026. Monitor Config Reference. Ceph Documentation. https://docs.ceph.com/en/latest/rados/configuration/mon-config-ref/ Accessed 28 October 2026.