

# changelog.md for @push.rocks/smartstorage

## 2026-03-23 - 6.3.2 - fix(docs)

update license ownership and correct README license file reference

- Adjusts the copyright holder name in the license file
- Fixes the README link to match the lowercase license filename

## 2026-03-21 - 6.3.1 - fix(cluster)

improve shard reconstruction validation and start background healing service

- use the erasure read quorum when reconstructing chunks instead of assuming data shard count
- verify reconstructed shards before writing healed data back to disk
- start the healing service during server initialization with shared local shard stores
- simplify QUIC request handling by decoding the full request buffer including trailing shard data
- clean up unused variables and imports across cluster modules

## 2026-03-21 - 6.3.0 - feat(readme)

document distributed cluster mode, erasure coding, and QUIC-based architecture

- Expand README overview and feature matrix to highlight clustering, multi-drive awareness, and distributed storage capabilities
- Add standalone and cluster mode usage examples plus cluster configuration options
- Document clustering internals including erasure coding, quorum behavior, QUIC transport, self-healing, and on-disk layout

# 2026-03-21 - 6.2.0 - feat(cluster)

add shard healing, drive health heartbeats, and clustered policy directory support

- implements manifest-based healing that scans affected shards on offline nodes, reconstructs data with erasure coding, and rewrites recovered shards to local storage
- includes drive status reporting in membership heartbeats by wiring DriveManager health checks into cluster heartbeat messages
- adds clustered policies directory initialization and exposes policy storage paths from the distributed coordinator
- extends distributed coordinator support for remote shard read and delete operations plus multipart upload session metadata

# 2026-03-21 - 6.1.0 - feat(cluster)

add clustered storage backend with QUIC transport, erasure coding, and shard management

- introduces cluster configuration in Rust and TypeScript, including seed nodes, drive paths, heartbeat settings, and erasure coding options
- adds core cluster modules for membership, topology state, object manifests, placement, shard storage, drive management, healing scaffolding, and inter-node protocol handling
- adds QUIC-based transport for cluster communication and integrates a distributed storage backend alongside the existing standalone FileStore
- updates the server startup path to initialize standalone or clustered storage based on configuration and exposes a basic clusterStatus management endpoint
- refreshes build and dependency versions to support the new clustered storage implementation

# 2026-03-14 - 6.0.1 - fix(rust-bridge)

update smartrust and limit RustBridge binary lookup to dist\_rust

- Bumps @push.rocks/smartrust from ^1.0.0 to ^1.3.2.
- Removes rust target debug and release fallback paths from RustBridge local binary resolution, relying on dist\_rust/ruststorage.

# 2026-03-14 - 6.0.0 - BREAKING CHANGE(core)

Rebrand from smart3 to smartstorage

- Package renamed from @push.rocks/smarts3 to @push.rocks/smartstorage
- Class renamed from Smarts3 to SmartStorage (no backward-compatible re-export)
- Interface renamed from ISmarts3Config to ISmartStorageConfig
- Method renamed from getS3Descriptor to getStorageDescriptor
- Rust binary renamed from rusts3 to ruststorage
- Rust types renamed: S3Error→StorageError, S3Action→StorageAction, S3Config→SmartStorageConfig, S3Server→StorageServer
- On-disk file extension changed from `._S3_object` to `._storage_object` (BREAKING for existing stored data)
- Default credentials changed from S3RVER to STORAGE
- All internal S3 branding removed; AWS S3 protocol compatibility (IAM actions, ARNs, SigV4) fully maintained

# 2026-02-17 - 5.3.0 - feat(auth)

add AWS SigV4 authentication and bucket policy support

- Implement AWS SigV4 full verification (constant-time comparison, 15-minute clock skew enforcement) and expose default signing region (`server.region = 'us-east-1'`).
- Add IAM-style bucket policy engine with Put/Get/Delete policy APIs (`GetBucketPolicy/PutBucketPolicy/DeleteBucketPolicy`), wildcard action/resource matching, Allow/Deny evaluation, and on-disk persistence under `.policies/{bucket}.policy.json`.
- Documentation and README expanded with policy usage, examples, API table entries, and notes about policy CRUD and behavior for anonymous/authenticated requests.
- Rust code refactors: simplify storage/server result structs and multipart handling (removed several unused size/key/bucket fields), remove `S3Error::to_response` and `error_xml` helpers, and other internal cleanup to support new auth/policy features.

# 2026-02-17 - 5.2.0 - feat(auth,policy)

add AWS SigV4 authentication and S3 bucket policy support

- Implemented real AWS SigV4 verification (HMAC-SHA256), including x-amz-date handling, clock skew enforcement and constant-time signature comparison
- Added bucket policy model, validator and evaluation engine (Deny > Allow > NoOpinion) with a PolicyStore (RwLock cache + disk-backed .policies/\*.policy.json)
- Integrated action resolution and auth+policy pipeline into the HTTP server: authorization checks run per-request, anonymous requests are denied by default, ListAllMyBuckets requires authentication
- Added bucket policy CRUD handlers via ?policy query parameter (GET/PUT/DELETE) and cleanup of policies on bucket deletion
- Storage and config updates: created .policies dir and policy path helpers; default region added to server config (TS + Rust)
- Added comprehensive tests for auth and policy behavior (policy CRUD, evaluation, per-action enforcement, auth integration)
- Updated Rust dependencies and Cargo.toml/Cargo.lock to include hmac, sha2, hex, subtle, cpufeatures

## 2026-02-13 - 5.1.1 - fix(smarts3)

replace TypeScript server with Rust-powered core and IPC bridge

- Major refactor: Node.js/TypeScript in-process server replaced by a Rust crate ('rusts3') with a TypeScript IPC wrapper (RustBridge).
- Removed many TypeScript server modules (smarts3-server, filesystem-store, multipart-manager, controllers, router, context, logger, xml utils, etc.); Smarts3Server export removed — public API now proxies to the Rust binary.
- Smarts3 now spawns and communicates with the rusts3 binary via RustBridge IPC (commands include start, stop, createBucket).
- Build & packaging changes: build script now runs `tsrust` before `tsbuild`; added `@git.zone/tsrust` devDependency; added `dist_rust` artifacts and new cross-compile targets in npmextra.json; .gitignore updated for rust/target.
- Dependency changes: added `@push.rocks/smartrust` (RustBridge) and simplified plugins surface; previous smartfs/smartxml usage removed from TS code and replaced by the Rust implementation + IPC.
- Added Rust project files (rust/Cargo.toml, rust/src/\*) implementing server, IPC management loop, storage, XML responses, errors, and config.
- Documentation updated (README and hints) to describe the Rust core, supported prebuilt targets (linux\_amd64, linux\_arm64), IPC commands, and developer build notes.

## 2025-11-23 - 5.1.0 - feat(multipart)

Implement full multipart upload support with persistent manager, periodic cleanup, and API integration

- Add IMultipartConfig to server config with defaults (expirationDays: 7, cleanupIntervalMinutes: 60) and merge into existing config flow
- Introduce MultipartUploadManager: persistent upload metadata on disk, part upload/assembly, restore uploads on startup, listParts/listUploads, abort/cleanup functionality
- Start and stop multipart cleanup task from Smarts3Server lifecycle (startCleanupTask on start, stopCleanupTask on stop) with configurable interval and expiration
- ObjectController: support multipart endpoints (initiate, upload part, complete, abort) and move assembled final object into the object store on completion; set ETag headers and return proper XML responses
- BucketController: support listing in-progress multipart uploads via ?uploads query parameter and return S3-compatible XML
- Persist multipart state to disk and restore on initialization to survive restarts; perform automatic cleanup of expired uploads

## 2025-11-23 - 5.0.2 - fix(readme)

Clarify contribution agreement requirement in README

- Updated the Issue Reporting and Security section in readme.md to make it explicit that developers must sign and comply with the contribution agreement (and complete identification) before obtaining a code.foss.global account to submit pull requests.

## 2025-11-23 - 5.0.1 - fix(docs)

Clarify README wording about S3 compatibility and AWS SDK usage

- Update README wording to "Full S3 API compatibility" and clarify it works seamlessly with AWS SDK v3 and other S3 clients

## 2025-11-23 - 5.0.0 - BREAKING CHANGE(core)

Production-ready S3-compatible server: nested config, multipart uploads, CORS, structured logging, SmartFS migration and improved error handling

- Breaking change: configuration format migrated from flat to nested structure (server, storage, auth, cors, logging, limits). Update existing configs accordingly.
- Implemented full multipart upload support (initiate, upload part, complete, abort) with on-disk part management and final assembly.
- Added CORS middleware with configurable origins, methods, headers, exposed headers, maxAge and credentials support.
- Structured, configurable logging (levels: error|warn|info|debug; formats: text|json) and request/response logging middleware.
- Simple static credential authentication middleware (configurable list of credentials).
- Migrated filesystem operations to @push.rocks/smartfs (Web Streams interoperability) and removed smartbucket from production dependencies.
- Improved S3-compatible error handling and XML responses (S3Error class and XML utilities).
- Exposed Smarts3Server and made store/multipart managers accessible for tests and advanced usage; added helper methods like getS3Descriptor and createBucket.

## 2025-11-23 - 4.0.0 - BREAKING CHANGE(Smarts3)

Migrate Smarts3 configuration to nested server/storage objects and remove legacy flat config support

- Smarts3.createAndStart() and Smarts3 constructor now accept ISmarts3Config with nested `server` and `storage` objects.
- Removed support for the legacy flat config shape (top-level `port` and `cleanSlate`) / `ILegacySmarts3Config`.
- Updated tests to use new config shape (server:{ port, silent } and storage:{ cleanSlate }).
- mergeConfig and Smarts3Server now rely on the nested config shape; consumers must update their initialization code.

## 2025-11-23 - 3.2.0 - feat(multipart)

Add multipart upload support with MultipartUploadManager and controller integration

- Introduce MultipartUploadManager (ts/classes/multipart-manager.ts) to manage multipart upload lifecycle and store parts on disk
- Wire multipart manager into server and request context (S3Context, Smarts3Server) and initialize multipart storage on server start

- Add multipart-related routes and handlers in ObjectController: initiate (POST ?uploads), upload part (PUT ?partNumber&uploadId), complete (POST ?uploadId), and abort (DELETE ?uploadId)
- On complete, combine parts into final object and store via existing FilesystemStore workflow
- Expose multipart manager on Smarts3Server for controller access

## 2025-11-23 - 3.1.0 - feat(logging)

Add structured Logger and integrate into Smarts3Server; pass full config to server

- Introduce a new Logger class (ts/classes/logger.ts) providing leveled logging (error, warn, info, debug), text/json formats and an enable flag.
- Integrate Logger into Smarts3Server: use structured logging for server lifecycle events, HTTP request/response logging and S3 errors instead of direct console usage.
- Smarts3 now passes the full merged configuration into Smarts3Server (config.logging can control logging behavior).
- Server start/stop messages and internal request/error logs are emitted via the Logger and respect the configured logging level/format and silent option.

## 2025-11-23 - 3.0.4 - fix(smarts3)

Use filesystem store for bucket creation and remove smartbucket runtime dependency

- Switched createBucket to call the internal FilesystemStore.createBucket instead of using @push.rocks/smartbucket
- Made Smarts3Server.store public so Smarts3 can access the filesystem store directly
- Removed runtime import/export of @push.rocks/smartbucket from plugins and moved @push.rocks/smartbucket to devDependencies in package.json
- Updated createBucket to return a simple { name } object after creating the bucket via the filesystem store

## 2025-11-23 - 3.0.3 - fix(filesystem)

Migrate filesystem implementation to @push.rocks/smartfs and add Web Streams handling

- Replace dependency @push.rocks/smartfile with @push.rocks/smartfs and update README references

- plugins: instantiate SmartFs with SmartFsProviderNode and export smartfs (remove direct fs export)
- Refactor FilesystemStore to use smartfs directory/file APIs for initialize, reset, list, read, write, copy and delete
- Implement Web Stream ↔ Node.js stream conversion for uploads/downloads (Readable.fromWeb and writer.write with Uint8Array)
- Persist and read metadata (.metadata.json) and cached MD5 (.md5) via smartfs APIs
- Update readme.hints and documentation to note successful migration and next steps

## 2025-11-21 - 3.0.2 - fix(smarts3)

Prepare patch release 3.0.2 — no code changes detected

- No source changes in the diff
- Bump patch version from 3.0.1 to 3.0.2 for maintenance/release bookkeeping

## 2025-11-21 - 3.0.1 - fix(readme)

Add Issue Reporting and Security section to README

- Add guidance to report bugs, issues, and security vulnerabilities via [community.foss.global](https://community.foss.global)
- Inform developers how to sign a contribution agreement and get a [code.foss.global](https://code.foss.global) account to submit pull requests

## 2025-11-21 - 3.0.0 - BREAKING CHANGE(Smarts3)

Remove legacy s3rver backend, simplify Smarts3 server API, and bump dependencies

- Remove legacy s3rver backend: s3rver and its types were removed from dependencies and are no longer exported from plugins.
- Simplify Smarts3 API: removed useCustomServer option; Smarts3 now always uses the built-in Smarts3Server (s3Instance is Smarts3Server) and stop() always calls Smarts3Server.stop().
- Update README to remove legacy s3rver compatibility mention.

- Dependency updates: bumped @push.rocks/smartbucket to ^4.3.0 and @push.rocks/smartxml to ^2.0.0 (major upgrades), removed s3rver/@types/s3rver, bumped @aws-sdk/client-s3 to ^3.937.0 and @git.zone/tstest to ^3.1.0.

## 2025-11-21 - 2.3.0 - feat(smarts3-server)

Introduce native custom S3 server implementation (Smarts3Server) with routing, middleware, context, filesystem store, controllers and XML utilities; add SmartXml and AWS SDK test; keep optional legacy s3rver backend.

- Add Smarts3Server: native, Node.js http-based S3-compatible server (ts/classes/smarts3-server.ts)
- New routing and middleware system: S3Router and MiddlewareStack for pattern matching and middleware composition (ts/classes/router.ts, ts/classes/middleware-stack.ts)
- Introduce request context and helpers: S3Context for parsing requests, sending responses and XML (ts/classes/context.ts)
- Filesystem-backed storage: FilesystemStore with bucket/object operations, streaming uploads, MD5 handling and Windows-safe key encoding (ts/classes/filesystem-store.ts)
- S3 error handling: S3Error class that maps S3 error codes and produces XML error responses (ts/classes/s3-error.ts)
- Controllers for service, bucket and object operations with S3-compatible XML responses and copy/range support (ts/controllers/\*.ts)
- XML utilities and SmartXml integration for consistent XML generation/parsing (ts/utills/xml.utills.ts, ts/plugins.ts)
- Expose native plugins (http, crypto, url, fs) and SmartXml via plugins.ts
- ts/index.ts: add useCustomServer option, default to custom server, export Smarts3Server and handle start/stop for both custom and legacy backends
- Add AWS SDK v3 integration test (test/test.aws-sdk.node.ts) to validate compatibility
- package.json: add @aws-sdk/client-s3 devDependency and @push.rocks/smartxml dependency
- Documentation: readme.md updated to describe native custom server and legacy s3rver compatibility

## 2025-11-20 - 2.2.7 - fix(core)

Update dependencies, code style and project config; add pnpm overrides and ignore AI folders

- Bump devDependencies and runtime dependencies (@git.zone/, @push.rocks/, @tsclass/tsclass, s3rver) to newer compatible versions
- Add pnpm.overrides entry to package.json and normalize repository URL format
- Code style and formatting fixes in TypeScript sources (ts/index.ts, ts/00\_commitinfo\_data.ts): whitespace, trailing commas, parameter formatting and minor API-return typing preserved
- tsconfig.json: simplify compiler options and compact exclude list
- Update .gitignore to add AI-related folders (.claude/, .serena/) to avoid accidental commits
- Documentation and changelog formatting tweaks (readme.md, changelog.md, npmextra.json) — whitespace/newline cleanups and expanded changelog entries

## 2025-08-16 - 2.2.6 - fix(Smarts3)

Allow overriding S3 descriptor; update dependencies, test config and documentation

- ts/index.ts: getS3Descriptor now accepts an optional Partial<IS3Descriptor> to override defaults (backwards compatible)
- package.json: updated devDependencies and runtime dependency versions (tstest, smartpath, tsclass, s3rver, etc.) and added packageManager field
- package.json: expanded test script to run tstest with --web --verbose --logfile --timeout 60
- test/test.ts: test instance port changed to 3333
- readme.md: major rewrite and expansion of usage examples, API reference and guides
- added project config files: .claude/settings.local.json and .serena/project.yml

## 2024-11-06 - 2.2.5 - fix(ci)

Corrected docker image URLs in Gitea workflows to match the correct domain format.

- Updated IMAGE environment variable in .gitea/workflows/default\_nottags.yaml
- Updated IMAGE environment variable in .gitea/workflows/default\_tags.yaml

## 2024-11-06 - 2.2.4 - fix(core)

Improve code style and update dependencies

- Updated @push.rocks/tapbundle to version ^5.4.3 in package.json.
- Fixed markdown formatting in readme.md.
- Improved code consistency in ts/00\_commitinfo\_data.ts, ts/plugins.ts, and test/test.ts.

# 2024-11-06 - 2.2.3 - fix(core)

Fix endpoint address from 'localhost' to '127.0.0.1' for better compatibility in Smarts3.getS3Descriptor

- Corrected the endpoint address in Smarts3.getS3Descriptor to ensure proper functioning across different environments.

# 2024-11-06 - 2.2.2 - fix(core)

Fixed function call for fastPut in the test suite to ensure proper file upload handling.

- Updated dependencies in package.json to newer versions.
- Corrected the function call in test suite for file upload.

# 2024-10-26 - 2.2.1 - fix(core)

Fix import and typings for improved compatibility

- Corrected the type signature for `getS3Descriptor` to return `IS3Descriptor`.
- Fixed import structure and updated dependencies for consistent namespace usage across plugins.

# 2024-10-26 - 2.2.0 - feat(ci)

Migrate CI/CD workflow from GitLab CI to Gitea CI

- Added new Gitea CI workflows for both non-tag and tag-based pushes
- Removed existing GitLab CI configuration

# 2024-05-29 - 2.1.1 - Updates and minor changes

Updates and changes based on minor configuration improvements and organizational shifts.

- Updated description file.
- Updated tsconfig settings.
- Updated npmextra.json with new githost configuration.
- Shifted to new organizational scheme.

## 2022-07-30 - 2.1.0 - Core improvements and fixes

Minor improvements and important core changes.

- Removed tslint from the core setup.

## 2022-07-30 - 2.0.2 - Bucket creation improvement

Enhanced file structure management.

- Improved bucket creation to store locally within the .nokit directory.

## 2022-04-14 - 2.0.0 to 2.0.1 - Structural updates and fixes

This release focused on core updates and structural changes.

- Reformatted the project structure.
- Core updates with minor fixes.

## 2021-12-20 - 1.0.10 - ESM Transition

Breaking changes and minor fixes, transitioning to ES Modules.

- **BREAKING CHANGE:** Transitioned core setup to ESM.
- 

Revision #2

Created 2026-03-28 13:09:56 UTC by foss.global Team

Updated 2026-03-29 16:53:02 UTC by foss.global Team