INTERNAL DEVELOPER DOCUMENTATION

Note: If you’re looking for how to use Ceph as a library from your own software, please see API Documentation.

You can start a development mode Ceph cluster, after compiling the source, with:

```
cd src
install -d -m0755 out dev/osd0
./vstart.sh -n -x -l
# check that it's there
./ceph health
```

Todo: vstart is woefully undocumented and full of sharp sticks to poke yourself with.

Mailing list

The official development email list is `ceph-devel@vger.kernel.org`. Subscribe by sending a message to `majordomo@vger.kernel.org` with the line:

```
subscribe ceph-devel
```

in the body of the message.

Contents

- Tracing Ceph With BlkKin
  - Installing Blkin
  - Configuring Ceph with Blkin
  - Testing Blkin
  - Install Zipkin
  - Show Ceph’s Blkin Traces in Zipkin-web
- BlueStore Internals
  - Small write strategies
  - Possible future modes
  - Mapping
- Cache pool
  - Purpose
  - Use cases
    - Read-write pool, writeback
    - Read-only pool, weak consistency
  - Interface
  - Tiering agent
    - HitSet metadata
    - Cache mode
    - Cache sizing
    - Other tunables
- ceph-disk
  - device-mapper crypt
    - Settings
    - lockbox
  - partitions
  - prepare class hierarchy
  - state transition of partitions
- CephFS Snapshots
  - Overview
  - Important Data Structures
  - Creating a snapshot
  - Updating a snapshot
  - Generating a SnapContext
  - Storing snapshot data

    - Storing snapshot metadata
    - Snapshot writeback
    - Deleting snapshots
    - Hard links
    - Multi-FS
- A Detailed Description of the Cephx Authentication Protocol
    - Introduction
    - Getting Started With Authorization
    - Phase I:
    - Phase II
- Configuration Management System
    - The Configuration File
    - Metavariables
    - Reading configuration values
    - Changing configuration values
    - Defining config options
        - Levels
        - Description and long description
        - Default values
        - Safety
        - Service
        - Tags
        - Enums
- config-key layout
    - OSD dm-crypt keys
    - ceph-mgr modules
    - Configuration
- CephContext
- Corpus structure
    - How to generate an object corpus
- Installing Oprofile
    - Installation
    - Compiling Ceph for Profiling
    - Ceph Configuration
- C++17 and libstdc++ ABI
- CephFS delayed deletion
- Deploying a development cluster
    - Usage
    - Options
    - Environment variables
- Deploying multiple development clusters on the same machine
    - Usage
- Development workflows
    - Release Cycle
    - Merging bug fixes or features
    - Resolving bug reports and implementing features
    - Running and interpreting teuthology integration tests
    - Preparing a new release
    - Cutting a new stable release
    - Cutting a new development release
    - Publishing point releases and backporting
- Documenting Ceph
    - User documentation
    - Code Documentation
    - Drawing diagrams
        - Graphviz
        - Ditaa
        - Blockdiag
        - Inkscape
- Serialization (encode/decode)
    - Adding a field to a structure
- Erasure Coded pool
    - Purpose
    - Use cases
        - Cold storage

        - Cheap multidatacenter storage
    - Interface
- File striping
    - ceph_file_layout
- FreeBSD Implementation details
    - Disk layout
    - Configuration
    - MON creation
    - OSD creation
- Building Ceph Documentation
    - Clone the Ceph Repository
    - Install the Required Tools
    - Build the Documents
- IANA Numbers
    - Private Enterprise Number (PEN) Assignment
    - Port number (monitor)
- Contributing to Ceph: A Guide for Developers
    - Introduction
    - Essentials (tl;dr)
        - Leads
        - History
        - Licensing
        - Source code repositories
        - Redmine issue tracker
        - Mailing list
        - IRC
        - Submitting patches
        - Building from source
        - Using ccache to speed up local builds
        - Development-mode cluster
        - Backporting
        - Guidance for use of cluster log
    - What is merged where and when ?
        - Development releases (i.e. x.0.z)
            - What ?
            - Where ?
            - When ?
            - Branch merges
        - Stable release candidates (i.e. x.1.z) phase 1
            - What ?
            - Where ?
            - When ?
            - Branch merges
        - Stable release candidates (i.e. x.1.z) phase 2
            - What ?
            - Where ?
            - When ?
            - Branch merges
        - Stable releases (i.e. x.2.z)
            - What ?
            - Where ?
            - When ?
            - Branch merges
    - Issue tracker
        - Issue tracker conventions
    - Basic workflow
        - Update the tracker
        - Upstream code
        - Local environment
        - Bugfix branch
        - Fix bug locally
        - GitHub pull request
        - Automated PR validation
            - Notes on PR make check test
        - Integration tests AKA ceph-qa-suite

        - Code review
        - Amending your PR
        - Merge
    - Testing
    - Testing - make check
        - Caveats
    - Testing - integration tests
        - Teuthology consumes packages
        - The nightlies
        - Suites inventory
        - teuthology-describe-tests
        - How integration tests are run
        - How integration tests are defined
        - Reading a standalone test
        - Test descriptions
        - How are tests built from directories?
        - Convolution operator
        - Concatenation operator
        - Filtering tests by their description
        - Reducing the number of tests
    - Testing in the cloud
        - Assumptions and caveat
        - Prepare tenant
        - Getting ceph-workbench
        - Linking ceph-workbench with your OpenStack tenant
        - Run the dummy suite
        - Run a standalone test
        - Interrupt a running suite
        - Upload logs to archive server
        - Provision VMs ad hoc
        - Deploy a cluster for manual testing
    - Testing - how to run s3-tests locally
        - Step 1 - build Ceph
        - Step 2 - vstart
        - Step 3 - run s3-tests
- Kernel client troubleshooting (FS)
- Library architecture
- Use of the cluster log
    - Severity
    - Frequency
    - Language and formatting
- Debug logs
    - Performance counters
- build on MacOS
- Messenger notes
    - ceph_perf_msgr
- Monitor bootstrap
    - Logical id
    - Secret keys
    - Cluster fsid
    - Monitor address
    - Peers
    - Cluster creation
        - Names and addresses
        - Addresses only
        - Names only
    - Cluster expansion
        - Initially peerless expansion
        - Expanding with initial members
- msgr2 protocol
    - Goals
    - Definitions
    - Phases
    - Banner
    - Frame format

    - Authentication
    - Message frame format
    - Message flow handshake
    - Message exchange
- Network Encoding
    - Conventions
        - Integers
        - Complex Types
        - Variable Arrays
        - Primitive Aliases
    - Structures
        - Optional
        - Pair
        - Triple
        - List
        - Blob
        - Map
    - Complex Types
        - utime_t
        - ceph_entity_name
- Network Protocol
    - Hello
        - Banner
        - Connect
        - Connect Reply
    - MSGR Protocol
        - CEPH_MSGR_TAG_CLOSE (0x06)
        - CEPH_MSGR_TAG_MSG (0x07)
        - CEPH_MSGR_TAG_ACK (0x08)
        - CEPH_MSGR_TAG_KEEPALIVE (0x09)
        - CEPH_MSGR_TAG_KEEPALIVE2 (0x0E)
        - CEPH_MSGR_TAG_KEEPALIVE2_ACK (0x0F)
- Object Store Architecture Overview
- OSD class path issues
- Peering
    - Concepts
    - Description of the Peering Process
    - State Model
- Using perf
    - Common Issues
    - Flamegraphs
- Perf counters
    - Access
    - Collections
    - Schema
    - Dump
- Perf histograms
    - Access
    - Collections
    - Schema
    - Dump
- PG (Placement Group) notes
    - Overview
    - Mapping algorithm (simplified)
    - User-visible PG States
- Developer Guide (Quick)
    - Development
    - Running a development deployment
    - Resetting your vstart environment
    - Running a RadosGW development environment
    - Run unit tests
- RADOS client protocol
    - Basics
    - Resends
    - Backoff

- RBD Incremental Backup
  - Header
  - Metadata records
    - From snap
    - To snap
    - Size
  - Data Records
    - Updated data
    - Zero data
  - Final Record
    - End
  - Header
  - Metadata records
    - From snap
    - To snap
    - Size
  - Data Records
    - Updated data
    - Zero data
  - Final Record
    - End
- RBD Export & Import
  - Header
  - Metadata records
    - Image order
    - Image format
    - Image Features
    - Image Stripe unit
    - Image Stripe count
    - ImageMeta Key and Value
  - Final Record
    - End
  - Diffs records
    - Snap Protection status
    - Others
- RBD Layering
  - Command line interface
  - Implementation
    - Data Flow
    - Parent/Child relationships
    - Protection
    - Resizing
    - Renaming
  - Header changes
    - cls_rbd
    - librbd
- Ceph Release Process
  - 1. Build environment
  - 2. Setup keyring for signing packages
  - 3. Set up build area
  - 4. Update Build version numbers
  - 5. Create Makefiles
  - 6. Run the release scripts
  - 7. Create RPM Repo
  - 8. Create Debian repo
  - 9. Push repos to ceph.org
  - 10. Update Git
    - Point release
    - Development and Stable releases
- Notes on Ceph repositories
  - Special branches
  - Rules
- Sepia community test lab
- Session Authentication for the Cephx Protocol
  - Introduction

    - Storing the Key
    - Signing Messages
    - Checking Signatures
    - Adding New Session Authentication Methods
    - Adding Encryption to Sessions
    - Session Security Statistics
- Testing notes
    - build-integration-branch
        - Setup
        - Using
- Public OSD Version
- Wireshark Dissector
    - Using
    - Developing
- OSD developer documentation
    - Backfill Reservation
    - Erasure Coded Placement Groups
        - Glossary
        - Table of content
            - Developer notes
                - Introduction
                - Reading and writing encoded chunks from and to OSDs
                - Erasure code library
                - Notes
            - Jerasure plugin
                - Introduction
            - ECBackend Implementation Strategy
                - Misc initial design notes
                    - PGTemp and Crush
                    - Distinguished acting set positions
                    - Object Classes
                    - Scrub
                    - Crush
            - ECBackend
                - MAIN OPERATION OVERVIEW
                    - WHOLE STRIPE WRITE
                    - READ-MODIFY-WRITE
                    - OSD Object Write and Consistency
                    - ExtentCache
                    - Pipeline
    - last_epoch_started
    - Log Based PG
        - Background
            - Why PrimaryLogPG?
            - Primary log-based replication
        - ReplicatedBackend/ECBackend unification strategy
            - PGBackend
        - PGBackend Interface Explanation
            - Readable vs Degraded
            - Client Reads
            - Scrub
            - Recovery
    - Map and PG Message handling
        - Overview
        - MOSDMap
        - MOSDPGOp/MOSDPGSubOp
        - CEPH_MSG_OSD_OP processing
        - Peering Messages
    - OSD
        - Concepts
        - Overview
    - OSD Throttles
        - WBThrottle
        - op_queue_throttle
        - journal usage throttle

    - PG
        - Concepts
        - Peering Details and Gotchas
    - PG Removal
    - PGPool
    - Recovery Reservation
        - Things to Note
        - See Also
    - Scrubbing Behavior Table
        - State variables
    - Snaps
        - Overview
        - Ondisk Structures
        - Snap Removal
        - Recovery
        - SnapMapper
        - Split
    - Watch Notify
        - Overview
        - Watch Lifecycle
        - Notify Lifecycle
    - Writeback Throttle
- MDS developer documentation
    - MDS internal data structures
    - Subtree exports
        - Normal Migration
- RADOS Gateway developer documentation
    - Usage Design Overview
        - Testing
    - Admin Ops Nonimplemented
        - Get Object
            - Syntax
            - Request Parameters
            - Response Entities
            - Special Error Responses
        - Head Object
            - Syntax
            - Request Parameters
            - Response Entities
            - Special Error Responses
        - Get Zone Info
            - Syntax
            - Response Entities
            - Special Error Responses
            - Example Response
        - Add Placement Pool
            - Syntax
            - Request Parameters
            - Response Entities
            - Special Error Responses
        - Remove Placement Pool
            - Syntax
            - Request Parameters
            - Response Entities
            - Special Error Responses
        - List Available Data Placement Pools
            - Syntax
            - Response Entities
        - List Expired Garbage Collection Items
            - Syntax
            - Request Parameters
            - Response Entities
            - Special Error Responses
        - Manually Processes Garbage Collection Items
            - Syntax

            - Request Parameters
            - Response Entities
            - Special Error Responses
        - Show Log Objects
            - Syntax
            - Request Parameters
            - Response Entities
            - Special Error Responses
        - Standard Error Responses
    - Rados Gateway S3 API Compliance
        - Naming code reference
        - Common Request Headers
        - Common Response Headers
        - Operations on the Service
        - Operations on Buckets
        - Operations on Objects
- ceph-volume developer documentation
    - Plugins
        - Adding Plugins
    - LVM
        - Tag API
        - Metadata
        - **`type`**
        - **`cluster_fsid`**
        - **`data_device`**
        - **`data_uuid`**
        - **`journal_device`**
        - **`journal_uuid`**
        - **`encrypted`**
        - **`osd_fsid`**
        - **`osd_id`**
        - **`block_device`**
        - **`block_uuid`**
        - **`db_device`**
        - **`db_uuid`**
        - **`wal_device`**
        - **`wal_uuid`**
    - systemd