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

- A Detailed Description of the Cephx Authentication Protocol
  - Introduction
  - Getting Started With Authorization
  - Phase I:
  - Phase II
- Configuration Management System
  - The Configuration File
  - Metavariables
  - Readin configuration values
  - Changing configuration values
- CephContext
- CephFS delayed deletion
- Differences from POSIX
- Documenting Ceph
  - Code Documentation
  - Drawing diagrams
    - Graphviz
    - Ditaa
    - Blockdiag
    - Inkscape
- File striping
  - ceph_file_layout
- Filestore filesystem compatilibity
  - ext4 limits total xattrs for 4KB
  - OSD journal replay of non-idempotent transactions
- Building Ceph Documentation
  - Clone the Ceph Repository
  - Install the Required Tools
  - Build the Documents
- Kernel client troubleshooting (FS)
- Library architecture
- Debug logs
  - Performance counters
- 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
- Object Store Architecture Overview
- OSD class path issues
- Peering
    - Concepts
    - Description of the Peering Process
    - State Model
- Perf counters
    - Access
    - Collections
    - Schema
    - Dump
- PG (Placement Group) notes
    - Overview
    - Mapping algorithm (simplified)
    - User-visible PG States
- RBD Incremental Backup
    - Header
    - Metadata records
        - From snap
        - To snap
        - Size
    - Data Records
        - Updated data
        - Zero data
    - Final Record
        - End
- RBD Layering
    - Command line interface
    - Implementation
        - Data Flow
        - Parent/Child relationships
        - Protection
        - Resizing
        - Renaming
    - Header changes
        - cls_rbd
        - librbd
- 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
        - Development release
        - Stable release
        - Point release
- 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
- OSD developer documentation
    - Backfill Reservation
    - Map and PG Message handling
        - Overview
        - MOSDMap
        - MOSDPGOp/MOSDPGSubOp
        - Peering Messages
    - OSD
        - Concepts
        - Overview
    - PG
        - Concepts
        - Peering Details and Gotchas
    - PG Removal
    - 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
- RADOS Gateway developer documentation
    - Usage Design Overview
        - Testing