RBD INCREMENTAL BACKUP

This is a simple streaming file format for representing a diff between two snapshots (or a snapshot and the head) of an RBD image.

HEADER

“rbd diff v1\n”

METADATA RECORDS

Every record has a one byte “tag” that identifies the record type, followed by some other data.

Metadata records come in the first part of the image. Order is not important, as long as all the metadata records come before the data records.

FROM SNAP

- u8: ‘f’
- le32: snap name length
- snap name

TO SNAP

- u8: ‘t’
- le32: snap name length
- snap name

SIZE

- u8: ‘s’
- le64: (ending) image size

DATA RECORDS

These records come in the second part of the sequence.

UPDATED DATA

- u8: ‘w’
- le64: offset
- le64: length
- length bytes of actual data

ZERO DATA

- u8: ‘z’
- le64: offset
- le64: length

FINAL RECORD

END

- u8: ‘e’

HEADER

“rbd diff v2\n”

METADATA RECORDS

Every record has a one byte “tag” that identifies the record type, followed by length of data, and then some other data.

Metadata records come in the first part of the image. Order is not important, as long as all the metadata records come before the data records.

In v2, we have the following metadata in each section: (1 Bytes) tag. (8 Bytes) length. (n Bytes) data.

In this way, we can skip the unrecognized tag.

FROM SNAP

- u8: ‘f’
- le64: length of appending data (4 + length)
- le32: snap name length
- snap name

TO SNAP

- u8: ‘t’
- le64: length of appending data (4 + length)
- le32: snap name length
- snap name

SIZE

- u8: ‘s’
- le64: length of appending data (8)
- le64: (ending) image size

DATA RECORDS

These records come in the second part of the sequence.

UPDATED DATA

- u8: ‘w’
- le64: length of appending data (8 + 8 + length)
- le64: offset
- le64: length
- length bytes of actual data

ZERO DATA

- u8: ‘z’
- le64: length of appending data (8 + 8)
- le64: offset
- le64: length

FINAL RECORD

END

- u8: ‘e’