

SHEC ERASURE CODE PLUGIN

The *shec* plugin encapsulates the **multiple SHEC** library. It allows ceph to recover data more efficiently than Reed Solomon codes.

CREATE AN SHEC PROFILE

To create a new *shec* erasure code profile:

```
ceph osd erasure-code-profile set {name} \  
  plugin=shec \  
  [k={data-chunks}] \  
  [m={coding-chunks}] \  
  [c={durability-estimator}] \  
  [crush-root={root}] \  
  [crush-failure-domain={bucket-type}] \  
  [crush-device-class={device-class}] \  
  [directory={directory}] \  
  [--force]
```

Where:

k={data-chunks}

Description: Each object is split in **data-chunks** parts, each stored on a different OSD.
Type: Integer
Required: No.
Default: 4

m={coding-chunks}

Description: Compute **coding-chunks** for each object and store them on different OSDs. The number of **coding-chunks** does not necessarily equal the number of OSDs that can be down without losing data.
Type: Integer
Required: No.
Default: 3

c={durability-estimator}

Description: The number of parity chunks each of which includes each data chunk in its calculation range. The number is used as a **durability estimator**. For instance, if c=2, 2 OSDs can be down without losing data.
Type: Integer
Required: No.
Default: 2

crush-root={root}

Description: The name of the crush bucket used for the first step of the CRUSH rule. For instance **step take default**.
Type: String
Required: No.
Default: default

crush-failure-domain={bucket-type}

Description: Ensure that no two chunks are in a bucket with the same failure domain. For instance, if the failure domain is **host** no two chunks will be stored on the same host. It is used to create a CRUSH rule step such as **step chooseleaf host**.
Type: String
Required: No.
Default: host

`crush-device-class={device-class}`

Description: Restrict placement to devices of a specific class (e.g., `ssd` or `hdd`), using the crush device class names in the CRUSH map.
Type: String
Required: No.
Default:

`directory={directory}`

Description: Set the **directory** name from which the erasure code plugin is loaded.
Type: String
Required: No.
Default: `/usr/lib/ceph/erasure-code`

`--force`

Description: Override an existing profile by the same name.
Type: String
Required: No.

BRIEF DESCRIPTION OF SHEC'S LAYOUTS

SPACE EFFICIENCY

Space efficiency is a ratio of data chunks to all ones in a object and represented as $k/(k+m)$. In order to improve space efficiency, you should increase k or decrease m .

space efficiency of $\text{SHEC}(4,3,2) = 4/(4+3) = 0.57$
 $\text{SHEC}(5,3,2)$ or $\text{SHEC}(4,2,2)$ improves $\text{SHEC}(4,3,2)$'s space efficiency

DURABILITY

The third parameter of SHEC ($=c$) is a durability estimator, which approximates the number of OSDs that can be down without losing data.

durability estimator of $\text{SHEC}(4,3,2) = 2$

RECOVERY EFFICIENCY

Describing calculation of recovery efficiency is beyond the scope of this document, but at least increasing m without increasing c achieves improvement of recovery efficiency. (However, we must pay attention to the sacrifice of space efficiency in this case.)

$\text{SHEC}(4,2,2) \rightarrow \text{SHEC}(4,3,2)$: achieves improvement of recovery efficiency

ERASURE CODE PROFILE EXAMPLES

```
$ ceph osd erasure-code-profile set SHECprofile \  
  plugin=shc \  
  k=8 m=4 c=3 \  
  crush-failure-domain=host  
$ ceph osd pool create shecpool 256 256 erasure SHECprofile
```