

Pools nach Classen anlegen (SSD Pool / HDD Pool)

Beschreibung:

In Ceph können Sie verschiedene Pools anlegen, um unterschiedliche Typen von Speichergeräten zu nutzen. Zum Beispiel können Sie SSD-, NVME- und HDD-Pools erstellen, um eine optimale Nutzung der verfügbaren Speichergeräte zu erreichen. Jeder Pool kann unterschiedliche Konfigurationen für die Leistung und die Ausfallsicherheit aufweisen, um den Anforderungen der Anwendung gerecht zu werden. Durch die Verteilung von Daten auf verschiedene Speichergeräte können Sie auch sicherstellen, dass Daten bei Ausfällen von Speichergeräten zugänglich bleiben.

Festplatten Typen (Classes)

In Ceph können einer OSD verschiedene Klassen hinzugefügt werden.

Diese sind:

1. `ssd`: Eine OSD-Klasse für SSD-Speichergeräte. Diese OSDs werden normalerweise für Daten verwendet, die eine höhere Leistung erfordern.
2. `hdd`: Eine OSD-Klasse für HDD-Speichergeräte. Diese OSDs werden normalerweise für Daten verwendet, die keine hohe Leistung erfordern, sondern eher für Archiv- oder Backupzwecke geeignet sind.
3. `nvme`: Eine OSD-Klasse für NVMe-Speichergeräte. Diese OSDs werden normalerweise für Daten verwendet, die eine sehr hohe Leistung erfordern, z.B. für Anwendungen mit hohem I/O-Durchsatz.

Festplatten Klasse löschen/setzen

Ist schon eine Festplattenklasse gesetzt muss diese erst gelöscht werden, bevor eine neue vergeben werden kann. Im Terminal dienen folgende Befehle zum löschen/setzen. In unserem Fall die OSD.12 auf NVME setzen.

Löschen

```
ceph osd crush rm-device-class <osdnr>
```

```
ceph osd crush rm-device-class osd.12
```

Ausgabe:

done removing class of osd(s): 12

Setzen

Syntax

```
classtype = hdd,ssd,nvme
```

```
ceph osd crush set-device-class <classtype> <osdnr>
```

```
ceph osd crush set-device-class nvme osd.12
```

Ausgabe:

```
set osd(s) 12 to class 'nvme'
```

Crushmap um weitere replicated rule erweitern/ändern.

!Wichtige Hinweise!!!

!!! WICHTIG!!! Es muss schon eine OSD mit der Klasse geben, bevor eine Rule dafür gebaut werden kann !!!

Ist dies nicht der Fall müssen wir erst die erste Rule nur anpassen für die Klasse die schon da ist.

Danach legen wir die OSD mit der neuen Klasse an, aber vor allem alles auf Maintenance setzen!!!

!!! WICHTIG!!! Alles auf Maintenance setzen, wie no backfill, norebalance etc setzen.

Name	Class	OSD Type	Status	Version	weight	reweight	Used (%)
default							
vserv0001							
osd.4	nvme	bluestore	up / in	16.2.6	2.911	1.00	62.84
osd.1	nvme	bluestore	down / out	16.2.6	2.911	0.00	0.00
osd.0							
osd.5							
osd.3							
osd.2							

Enable	Name	Description
<input checked="" type="checkbox"/>	nobackfill	Backfilling of PGs is suspended.
<input checked="" type="checkbox"/>	nodeep-scrub	Deep Scrubbing is disabled.
<input type="checkbox"/>	nodown	OSD failure reports are being ignored, such that the monitors will not mark OSDs do...
<input type="checkbox"/>	noin	OSDs that were previously marked out will not be marked back in when they start.
<input type="checkbox"/>	noout	OSDs will not automatically be marked out after the configured interval.
<input checked="" type="checkbox"/>	norebalance	Rebalancing of PGs is suspended.
<input checked="" type="checkbox"/>	norecover	Recovery of PGs is suspended.
<input checked="" type="checkbox"/>	noscrub	Scrubbing is disabled.
<input type="checkbox"/>	notieragent	Cache tiering activity is suspended.
<input type="checkbox"/>	noup	OSDs are not allowed to start.
<input type="checkbox"/>	pause	Pauses read and writes.

Erst nach Erstellung der Rule die Haken wieder entfernen.

Über manuelles bearbeiten

Die Crushmap decompilieren so, dass man diese auch bearbeiten kann. Siehe Seite : [Crushmap decompilieren / kompilieren](#)

Ist die Map decompiliert. Der ersten Rule den Parameter für die Klassenzuweisung hinzufügen. Damit wäre dann die erste Regel definiert, dass diese für HDDs gilt. Sollten nur NVMEs verbaut sein. Dann natürlich die erste Rule mit NVME setzen und die zweite dann auf HDDs. In unserem Szenario sind aber HDDs vorhanden und wir wollen NVME Pool hinzufügen. Sind noch gar keine Pools definiert, so kann man auch die erste Rule noch umbenennen, z.B. noch `_hdd` hinzufügen. Ist die Rule allerdings schon einem Pool zugewiesen, muss dies aktualisiert werden. Was in einer Prod-Umgebung nicht wirklich sinnvoll ist, dann einfach als Gedanken lassen, die erste Rule ist die Rule von dem Klassentyp, die zu erst da war. Wie bei uns die HDDs.

Hier der Parameter der einer Rule hinzugefügt werden muss

```
step take default class <classtype>
step take default class hdd
```

In unsere Crushmap sehe das so aus.

Auszug.

Original

```
...  
# rules  
rule replicated_rule {  
    id 0  
    type replicated  
    min_size 1  
    max_size 10  
    step chooseleaf firstn 0 type host  
    step emit  
}  
...
```

Nun geändert mit HDD class

```
...  
# rules  
rule replicated_rule {  
    id 0  
    type replicated  
    min_size 1  
    max_size 10  
    step take default class hdd  
    step chooseleaf firstn 0 type host  
    step emit  
}
```

Diese speichern und wieder zurückzuspielen.

Siehe Seite : [Crushmap dekompileieren / kompilieren](#)

Nun eine zweite Rule anlegen, dazu die erste kopieren und die ID und Namen ändern.

z.b so

```
...  
# rules  
rule replicated_rule {  
    id 0  
    type replicated  
    min_size 1
```

```

    max_size 10
    step take default class hdd
    step chooseleaf firstn 0 type host
    step emit
}

# rules
rule replicated_rule_nvme {
    id 1
    type replicated
    min_size 1
    max_size 10
    step take default class nvme
    step chooseleaf firstn 0 type host
    step emit
}

```

Diese speichern und wieder zurückzuspielen.

Siehe Seite : [Crushmap dekompilieren / kompilieren](#)

Fertig, nun Pool anlegen.

Über Kommandozeilen Befehl

Die Erste Rule die Klasse hinzufügen, wenn die noch keine Klasse hat

```
ceph osd crush rule modify <rulename> set class <classtype>
```

Beispiel

```
ceph osd crush rule modify replicated_rule set class hdd
```

Nun wenn nicht schon vorhanden OSD mit neuer Klasse hinzufügen.

Dann die weitere Rule erstellen

Erklärung der Parameter

<rule-name>	Name der Regel, um sie mit einem Pool zu verbinden (angezeigt in der GUI und CLI)
<root>	Zu welchem Crush-Root sie gehören sollte (Standard-Ceph-Root "default")

<failure-domain>	An welchem Fehlerbereich die Objekte verteilt werden sollten (normalerweise Host) (Bei one Node wäre das hier osd)
<class>	Welcher Typ von OSD-Backing-Store verwendet werden soll (e.g., nvme, ssd, hdd)

Syntax

```
ceph osd crush rule create-replicated <rule-name> <root> <failure-domain> <class>
```

Beispiel

```
ceph osd crush rule create-replicated rule_nvme default host nvme
```

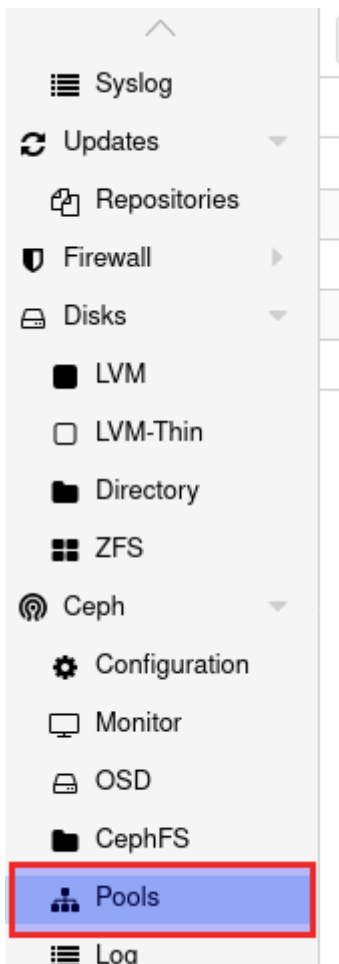
!!! WICHTIG!!! Maintenance mode wieder rausnehmen no backfill, norebalance etc aushaken.

Fertig nun Pool anlegen

Pool anlegen

Nun können wir einen weiteren Pool mit der neuen Rule anlegen.

Dazu auf der Weboberfläche proxmox anmelden, einen Host auswählen dann auf Ceph / Pools klicken



Dann auf Create klicken

Create	Edit	Destroy	
Name	Size/min	# of Placement Groups	Optimal # of PGs
device_health_metrics	3/2	1	1
rbd	3/2	32	128
rbd_hdd	3/2	32	32
cephfs_data	3/2	32	32
cephfs_metadata	3/2	32	16

nun einen Namen vergeben und unser Ruleset auswählen. Auto PG auf on und den Haken bei Add Storage rein.

Dann bekommen wir auch gleich den Speicher zum Datastore um darauf zuzugreifen.

Create: Ceph Pool ⊗

Name: PG Autoscale Mode:

Size: Add as Storage:

Min. Size: Target Ratio:

Crush Rule: Target Size: GiB

of PGs: **Target Ratio takes precedence.**

Min. # of PGs:

Advanced

Fertig.

So könnte man zum Beispiel ein RBD Pool für NVMEs bauen und einen CephFS Pool mit HDDs.
Als Beispiel

!!!ACHTUNG! NICHT ZU EMPFEHLEN!!!

Man kann natürlich auch einen Vorhandenen Pool ein Komplettes anderes Ruleset geben.
Aber Achtung die Daten ziehen dann nicht mit um. Die Änderungen gelten nur für neu geschriebene Daten.

Deshalb lieber einen neuen Pool Anlegen, Daten dort hin verschieben, alten Pool löschen.
Dann einen neuen Pool mit altem Namen und dann der richtigen Ruleset auswählen.
So bleiben die Daten konsistent

Version #9

Erstellt: 29 März 2023 05:00:04 von Admin

Zuletzt aktualisiert: 2 August 2023 10:03:17 von Admin