



# **Speicherung von Objekten (Replizierung oder Erasure Coding)**

StorageGRID

NetApp  
March 12, 2025

# Inhalt

Speicherung von Objekten (Replizierung oder Erasure Coding) .....	1
Was ist Replikation? .....	1
Warum sollten Sie keine Replizierung mit nur einer Kopie verwenden .....	2
Was ist Erasure Coding? .....	4
Was sind Erasure Coding-Systeme? .....	6
Datensicherung .....	6
Storage Overhead .....	7
Richtlinien für Speicherpools .....	7
Verfahren zur Einhaltung von Datenkonsistenz für Storage-Pools mit drei oder mehr Standorten .....	8
Verfahren zur Einhaltung von Datenkonsistenz für Storage-Pools an einem Standort .....	9
Vor- und Nachteile sowie Anforderungen für Erasure Coding .....	9
Vorteile von Erasure Coding .....	10
Nachteile des Erasure Coding .....	10
Wann sollte das Erasure Coding verwendet werden .....	11

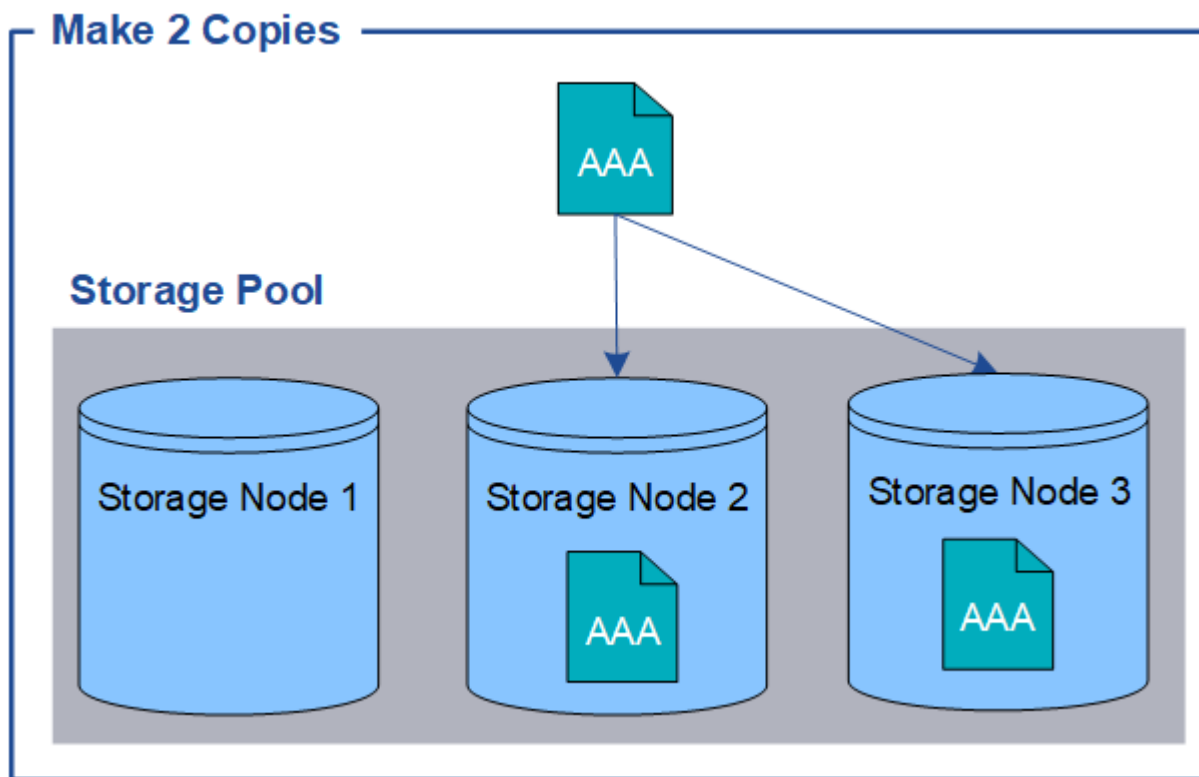
# Speicherung von Objekten (Replizierung oder Erasure Coding)

## Was ist Replikation?

Die Replikation ist eine von zwei Methoden, die von StorageGRID zur Speicherung von Objektdaten verwendet werden (bei Erasure Coding handelt es sich um die andere Methode). Wenn Objekte mit einer ILM-Regel übereinstimmen, die Replikation verwendet, erstellt das System exakte Kopien von Objektdaten und speichert die Kopien auf Storage Nodes.

Wenn Sie eine ILM-Regel zum Erstellen replizierter Kopien konfigurieren, geben Sie an, wie viele Kopien erstellt werden sollen, wo diese Kopien erstellt werden sollen und wie lange die Kopien an jedem Standort gespeichert werden sollen.

Im folgenden Beispiel gibt die ILM-Regel an, dass zwei replizierte Kopien jedes Objekts in einem Storage-Pool mit drei Storage-Nodes platziert werden.



Wenn StorageGRID Objekte mit dieser Regel übereinstimmt, werden zwei Kopien des Objekts erstellt, wobei jede Kopie auf einem anderen Storage-Node im Storage-Pool platziert wird. Die beiden Kopien können auf zwei der drei verfügbaren Storage-Nodes platziert werden. In diesem Fall wurden in der Regel Objektkopien auf Speicherknoten 2 und 3 platziert. Da es zwei Kopien gibt, kann das Objekt abgerufen werden, wenn einer der Nodes im Speicherpool ausfällt.



StorageGRID kann nur eine replizierte Kopie eines Objekts auf einem beliebigen Storage Node speichern. Wenn Ihr Grid drei Storage-Nodes enthält und Sie eine ILM-Regel mit 4 Kopien erstellen, werden nur drei Kopien erstellt: Eine Kopie für jeden Storage-Node. Die Warnung **ILM-Platzierung unerreichbar** wird ausgelöst, um anzuzeigen, dass die ILM-Regel nicht vollständig angewendet werden konnte.

#### Verwandte Informationen

- ["Was ist Erasure Coding"](#)
- ["Was ist ein Speicherpool"](#)
- ["Schutz vor Standortausfällen durch Replizierung und Erasure Coding"](#)

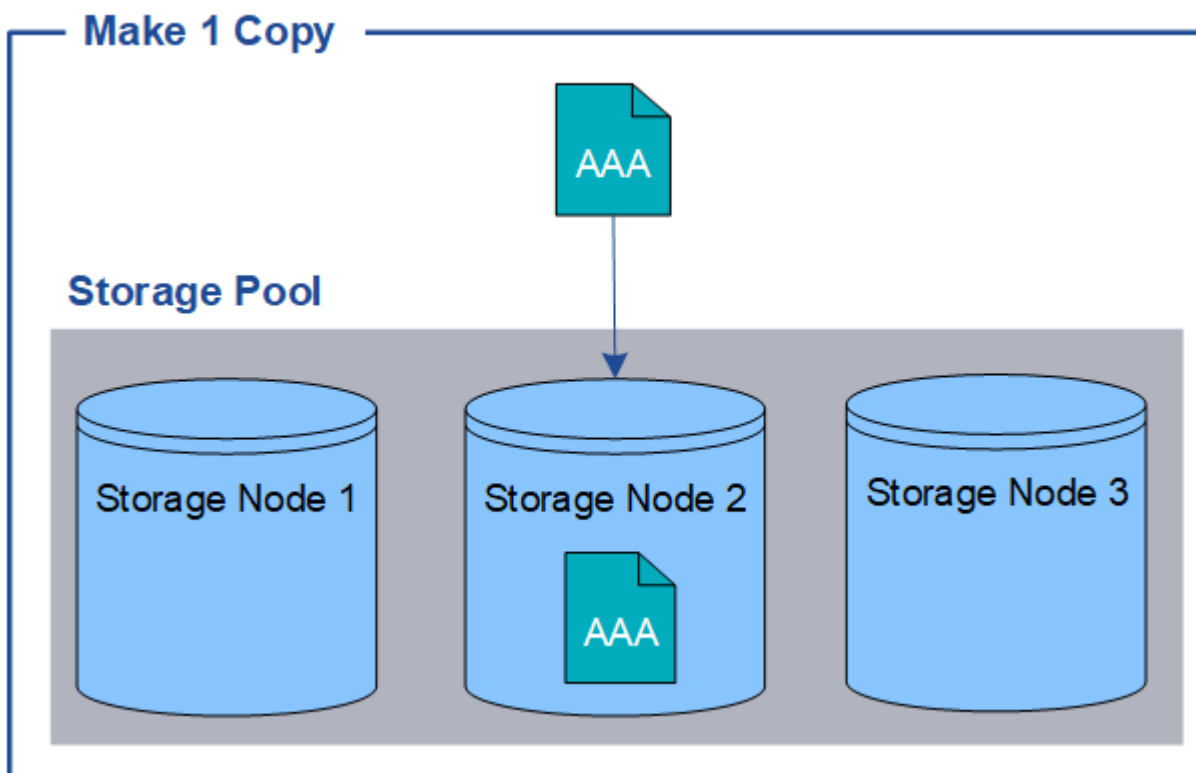
## Warum sollten Sie keine Replizierung mit nur einer Kopie verwenden

Beim Erstellen einer ILM-Regel zum Erstellen replizierter Kopien sollten Sie immer mindestens zwei Kopien für einen beliebigen Zeitraum in den Anweisungen zur Platzierung angeben.

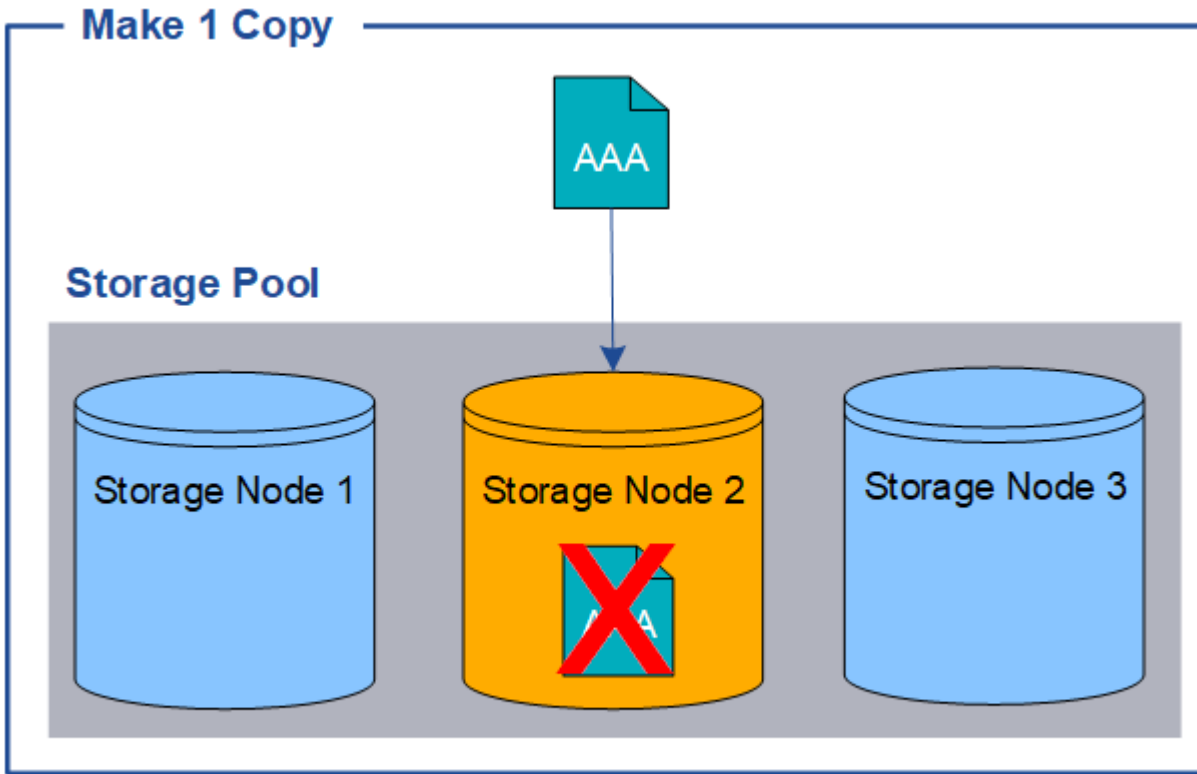


Verwenden Sie keine ILM-Regel, die nur eine replizierte Kopie für einen beliebigen Zeitraum erstellt. Wenn nur eine replizierte Kopie eines Objekts vorhanden ist, geht dieses Objekt verloren, wenn ein Speicherknoten ausfällt oder einen beträchtlichen Fehler hat. Während Wartungsarbeiten wie Upgrades verlieren Sie auch vorübergehend den Zugriff auf das Objekt.

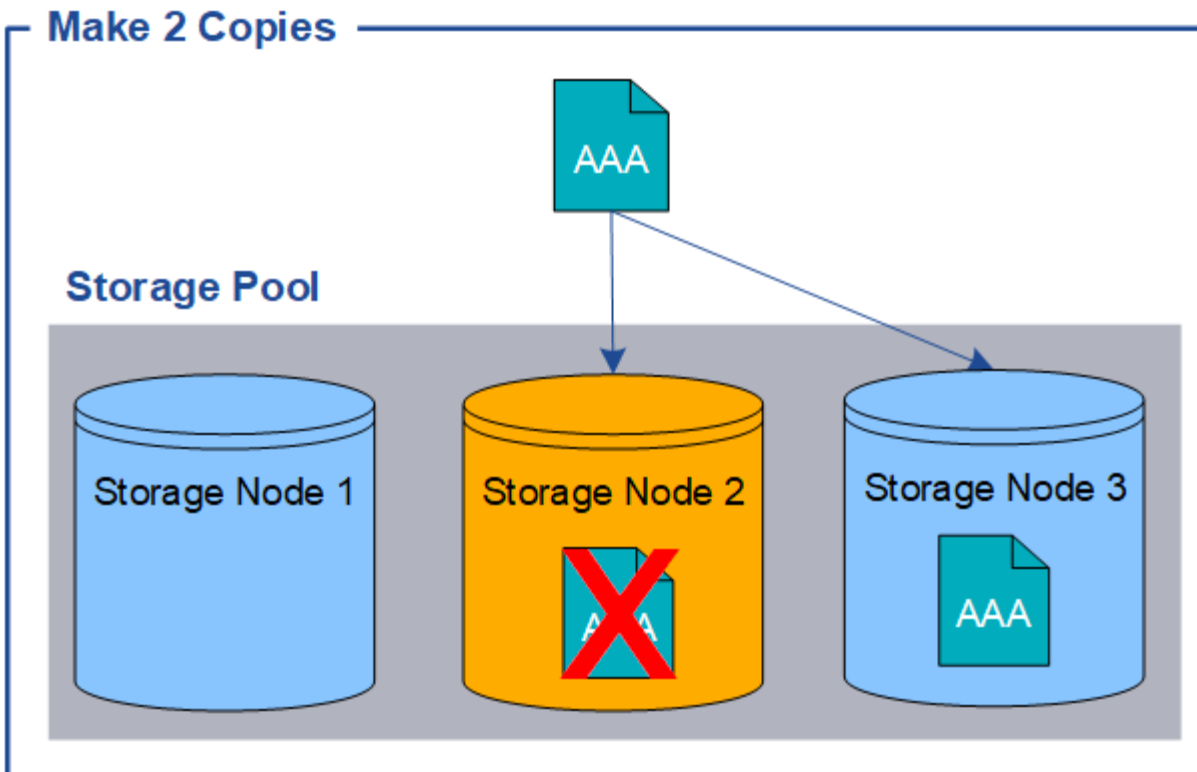
Im folgenden Beispiel gibt die ILM-Regel „1 Kopie erstellen“ an, dass eine replizierte Kopie eines Objekts in einem Speicherpool platziert wird, der drei Storage-Nodes enthält. Wenn ein Objekt aufgenommen wird, das dieser Regel entspricht, platziert StorageGRID eine einzelne Kopie auf nur einem Storage-Node.



Wenn eine ILM-Regel nur eine replizierte Kopie eines Objekts erstellt, ist der Zugriff auf das Objekt möglich, wenn der Storage-Node nicht verfügbar ist. In diesem Beispiel verlieren Sie vorübergehend den Zugriff auf das Objekt AAA, wenn Storage Node 2 offline ist, z. B. während eines Upgrades oder eines anderen Wartungsverfahrens. Sie verlieren das Objekt AAA vollständig, wenn Storage Node 2 ausfällt.



Um den Verlust von Objektdaten zu vermeiden, sollten immer mindestens zwei Kopien aller Objekte erstellt werden, die durch die Replizierung gesichert werden sollen. Wenn zwei oder mehr Kopien vorhanden sind, können Sie weiterhin auf das Objekt zugreifen, wenn ein Storage-Node ausfällt oder offline geht.



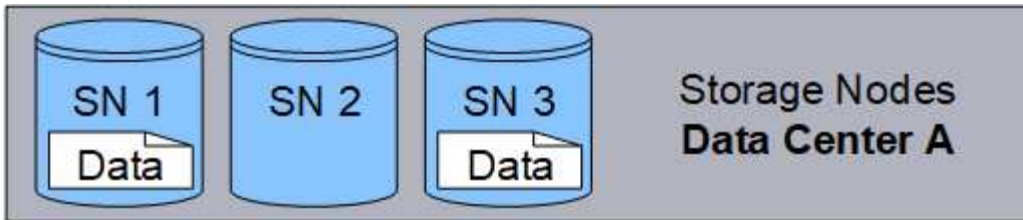
## Was ist Erasure Coding?

Erasure Coding ist eine von zwei Methoden, die StorageGRID zum Speichern von Objektdaten verwendet (bei der Replizierung handelt es sich um die andere Methode). Wenn Objekte mit einer ILM-Regel übereinstimmen, die Erasure Coding verwendet, werden diese Objekte in Datenfragmente geteilt, weitere Paritätsfragmente werden berechnet und jedes Fragment wird auf einem anderen Storage Node gespeichert.

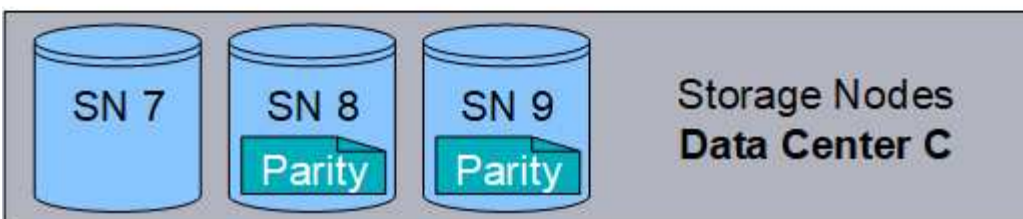
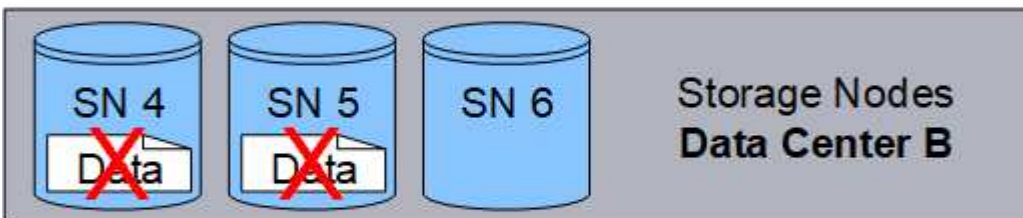
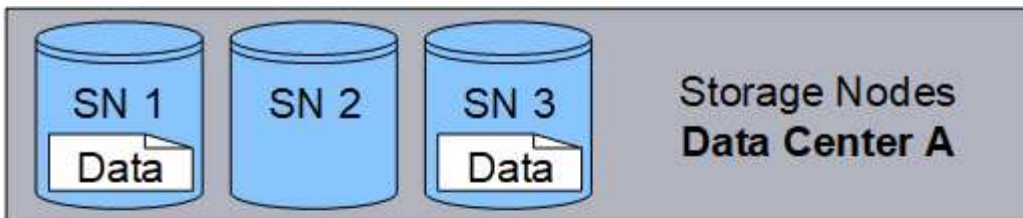
Wenn auf ein Objekt zugegriffen wird, wird es anhand der gespeicherten Fragmente neu zusammengesetzt. Wenn ein Daten- oder ein Paritätsfragment beschädigt wird oder verloren geht, kann der Algorithmus zur Fehlerkorrektur dieses Fragment mit einer Teilmenge der verbleibenden Daten und Paritätsfragmente neu erstellen.

Beim Erstellen von ILM-Regeln erstellt StorageGRID Profile zur Einhaltung von Datenkonsistenz, die diese Regeln unterstützen. Sie können eine Liste von Erasure-Coding-Profilen anzeigen, ["Umbenennen eines Profils für die Erasure Coding"](#), oder ["Deaktivieren Sie ein Erasure Coding-Profil, wenn es derzeit nicht in ILM-Regeln verwendet wird"](#).

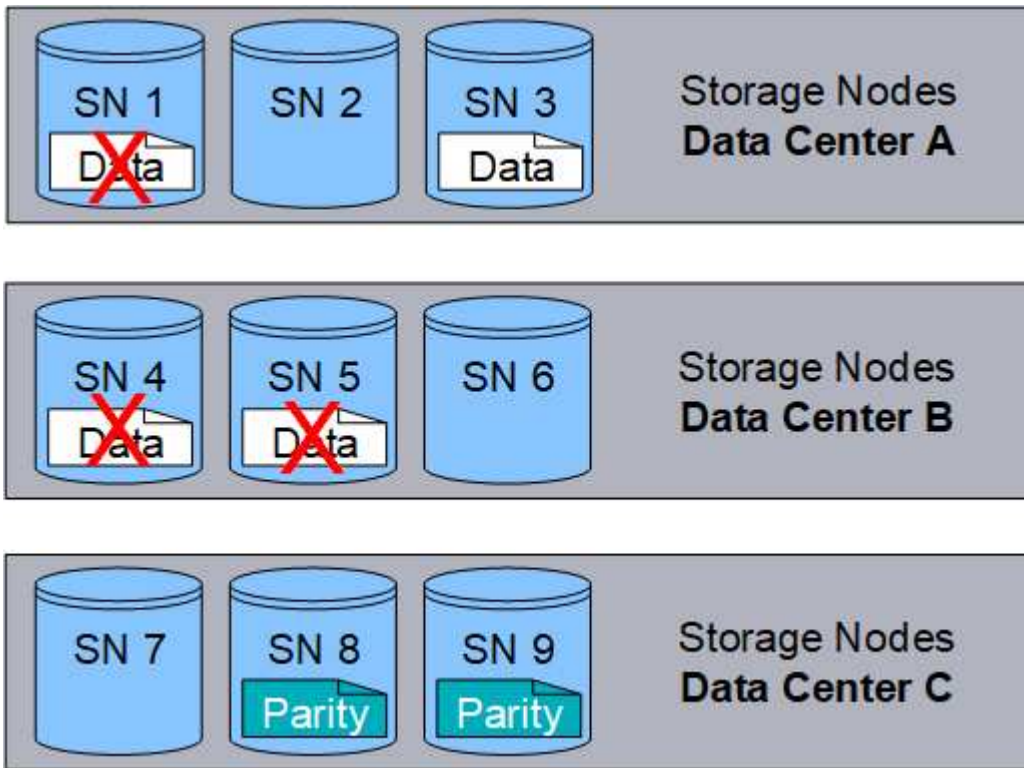
Im folgenden Beispiel wird der Algorithmus zur Einhaltung von Datenkonsistenz (Erasure Coding) für Objektdaten dargestellt. In diesem Beispiel verwendet die ILM-Regel ein 4+2-Schema zur Einhaltung von Datenkonsistenz. Jedes Objekt wird in vier gleiche Datenfragmente geteilt und aus den Objektdaten werden zwei Paritätsfragmente berechnet. Jedes der sechs Fragmente wird auf einem anderen Node über drei Datacenter-Standorte gespeichert, um Daten bei Node-Ausfällen oder Standortausfällen zu sichern.



Das 4+2 Erasure Coding-Schema kann auf verschiedene Weise konfiguriert werden. Sie können beispielsweise einen Speicherpool mit einem Standort konfigurieren, der sechs Storage-Nodes enthält. Für "Schutz vor Standortausfällen" können Sie einen Speicherpool verwenden, der drei Standorte mit drei Storage-Nodes an jedem Standort enthält. Ein Objekt kann abgerufen werden, solange vier der sechs Fragmente (Daten oder Parität) verfügbar sind. Bis zu zwei Fragmente können ohne Verlust der Objektdaten verloren gehen. Wenn ein ganzer Standort verloren geht, kann das Objekt dennoch abgerufen oder repariert werden, solange alle anderen Fragmente zugänglich bleiben.



Wenn mehr als zwei Speicherknoten verloren gehen, kann das Objekt nicht abgerufen werden.



#### Verwandte Informationen

- ["Was ist Replikation"](#)
- ["Was ist ein Speicherpool"](#)
- ["Was sind Erasure Coding-Systeme"](#)
- ["Umbenennen eines Profils für die Erasure Coding"](#)
- ["Deaktivieren Sie ein Erasure Coding-Profil"](#)

## Was sind Erasure Coding-Systeme?

Erasure Coding steuert die Anzahl von Datenfragmenten und die Anzahl der Parity-Fragmente für jedes Objekt.

Wenn Sie eine ILM-Regel erstellen oder bearbeiten, wählen Sie ein verfügbares Schema für die Einhaltung von Datenkonsistenz aus. StorageGRID erstellt automatisch Schemata für die Einhaltung von Datenkonsistenz anhand der Anzahl der Storage-Nodes und Standorte des Storage-Pools, die Sie verwenden möchten.

### Datensicherung

Das StorageGRID-System verwendet den Reed-Solomon-Erasure-Coding-Algorithmus. Der Algorithmus unterteilt ein Objekt in  $k$  Datenfragmente und berechnet  $m$  Paritätsfragmente.

Die  $k + m = n$  Fragmente werden über Storage-Nodes verteilt  $n$ , um folgende Datensicherungsmaßnahmen zu ermöglichen:

- Um ein Objekt abzurufen oder zu reparieren,  $k$  werden Fragmente benötigt.



- Ein Objekt kann verlorene oder beschädigte Fragmente erhalten  $m$ . Je höher der Wert von  $m$ , desto höher die Ausfalltoleranz.

Die beste Datensicherung wird durch das Erasure Coding-Schema mit der höchsten Ausfalltoleranz für Nodes oder Volumes innerhalb eines Storage-Pools erreicht.

## Storage Overhead

Der Speicher-Overhead eines Erasure-Coding-Schemas wird berechnet, indem die Anzahl der Paritäts-Fragmente ( $m$ ) durch die Anzahl der Datenfragmente geteilt wird ( $k$ ). Der Storage Overhead lässt sich ermitteln, wie viel Festplattenspeicher jedes mit Erasure-Coding-Objekt benötigt:

$$\text{disk space} = \text{object size} + (\text{object size} * \text{storage overhead})$$

Wenn Sie beispielsweise ein Objekt mit 10 MB unter Verwendung des Schemas von 4+2 speichern (mit einem Mehraufwand von 50 %), verbraucht das Objekt 15 MB Grid Storage. Wenn Sie dasselbe 10 MB große Objekt mit dem Schema 6+2 speichern (mit einem Mehraufwand von 33 %), verbraucht das Objekt etwa 13.3 MB.

Wählen Sie das Erasure Coding-Schema mit dem niedrigsten Gesamtwert  $k+m$ , der Ihren Anforderungen entspricht. Erasure Coding-Schemata mit einer geringeren Anzahl von Fragmenten sind aus folgenden Gründen recheneffizienter:

- Pro Objekt werden weniger Fragmente erstellt und verteilt (oder abgerufen)
- Sie zeigen eine bessere Performance, da die Fragmentgröße größer ist
- Das Hinzufügen von weniger Nodes in einem ist möglich **"Erweiterung, wenn mehr Storage benötigt wird"**

## Richtlinien für Speicherpools

Verwenden Sie bei der Auswahl des Speicherpools für eine Regel, die eine Kopie mit Verfahren zur Fehlerkorrektur erstellt, die folgenden Richtlinien für Speicherpools:

- Der Speicherpool muss drei oder mehr Standorte oder exakt einen Standort umfassen.



Sie können kein Erasure Coding verwenden, wenn der Storage-Pool zwei Standorte umfasst.

- [Verfahren zur Einhaltung von Datenkonsistenz für Storage-Pools mit drei oder mehr Standorten](#)
- [Verfahren zur Einhaltung von Datenkonsistenz für Storage-Pools an einem Standort](#)
- Verwenden Sie keinen Speicherpool, der den Standort „Alle Standorte“ umfasst.
- Der Speicherpool sollte mindestens Storage Nodes enthalten  $k+m + 1$ , die Objektdaten speichern können.



Storage-Nodes können während der Installation so konfiguriert werden, dass sie nur Objektmetadaten und keine Objektdaten enthalten. Weitere Informationen finden Sie unter ["Typen von Storage-Nodes"](#).

Die Mindestanzahl der erforderlichen Storage-Nodes ist  $k+m$ . Durch mindestens einen zusätzlichen Storage-Node können jedoch Ingest- oder ILM-Backlogs verhindert werden, wenn ein erforderlicher Storage-Node vorübergehend nicht verfügbar ist.

## Verfahren zur Einhaltung von Datenkonsistenz für Storage-Pools mit drei oder mehr Standorten

Die folgende Tabelle beschreibt die von StorageGRID derzeit unterstützten Erasure Coding-Schemata für Storage-Pools, die drei oder mehr Standorte umfassen. Alle diese Maßnahmen bieten einen Standortausfallschutz. Ein Standort kann verloren gehen, und das Objekt ist weiterhin verfügbar.

Für Erasure Coding-Schemata, die Schutz vor Standortausfällen bieten, übersteigt die empfohlene Anzahl von Storage-Nodes im Speicherpool  $k+m + 1$ , da für jeden Standort mindestens drei Storage-Nodes erforderlich sind.

Schema zur Einhaltung von Datenkonsistenz (Erasure Coding) ( $k+m$ )	Mindestanzahl der bereitgestellten Standorte	Empfohlene Anzahl von Storage-Nodes an jedem Standort	Insgesamt empfohlene Anzahl von Storage-Nodes	Schutz vor Standortausfällen?	Storage Overhead
4+2	3	3	9	Ja.	50 % erreicht
6+2	4	3	12	Ja.	33 % erreicht
8+2	5	3	15	Ja.	25 % erreicht
6+3	3	4	12	Ja.	50 % erreicht
9+3	4	4	16	Ja.	33 % erreicht
2+1	3	3	9	Ja.	50 % erreicht
4+1	5	3	15	Ja.	25 % erreicht
6+1	7	3	21	Ja.	17 % erreicht
7+5	3	5	15	Ja.	71 % erreicht



StorageGRID erfordert mindestens drei Storage-Nodes pro Standort. Für die Verwendung des Schemas 7+5 benötigt jeder Standort mindestens vier Speicherknoten. Es wird empfohlen, fünf Storage-Nodes pro Standort zu verwenden.

Bei der Auswahl eines Löschungsschemas, das Standortschutz bietet, sollte die relative Bedeutung der folgenden Faktoren in Einklang gestellt werden:

- **Anzahl der Fragmente:** Leistung und Expansionsflexibilität sind im Allgemeinen besser, wenn die Gesamtzahl der Fragmente geringer ist.
- **Fehlertoleranz:** Die Fehlertoleranz wird durch mehr Paritätssegmente erhöht (d.h. wenn  $m$  einen höheren Wert hat).
- **Netzwerkverkehr:** Bei der Wiederherstellung nach Ausfällen erzeugt die Verwendung eines Schemas mit mehr Fragmenten (also einer höheren Summe für  $k+m$ ) mehr Netzwerkverkehr.

- **Storage Overhead:** Bei Systemen mit höherem Overhead wird mehr Speicherplatz pro Objekt benötigt.

Wenn Sie beispielsweise zwischen einem Schema 4+2 und dem Schema 6+3 (mit jeweils 50 % Storage Overhead) entscheiden, wählen Sie das Schema 6+3 aus, wenn eine zusätzliche Fehlertoleranz erforderlich ist. Wählen Sie das Schema 4+2 aus, wenn die Netzwerkressourcen begrenzt sind. Wenn alle anderen Faktoren gleich sind, wählen Sie 4+2 aus, da die Gesamtzahl der Fragmente geringer ist.



Wenn Sie sich nicht sicher sind, welches Schema Sie verwenden möchten, wählen Sie 4+2 oder 6+3 aus, oder wenden Sie sich an den technischen Support.

## Verfahren zur Einhaltung von Datenkonsistenz für Storage-Pools an einem Standort

Ein Storage-Pool an einem Standort unterstützt alle Erasure Coding-Schemata, die für drei oder mehr Standorte definiert sind, sofern der Standort über ausreichend Storage-Nodes verfügt.

Die Mindestanzahl der erforderlichen Storage-Nodes ist  $k+m$ , jedoch wird ein Speicherpool mit  $k+m + 1$  Storage-Nodes empfohlen. Zum Beispiel erfordert das Verfahren zur Einhaltung von Datenkonsistenz (Erasure Coding) 2+1 einen Speicherpool mit mindestens drei Storage-Nodes, es werden jedoch vier Storage-Nodes empfohlen.

Schema zur Einhaltung von Datenkonsistenz (Erasure Coding) ( $k+m$ )	Mindestanzahl Storage-Nodes	Empfohlene Anzahl von Storage-Nodes	Storage Overhead
4+2	6	7	50 % erreicht
6+2	8	9	33 % erreicht
8+2	10	11	25 % erreicht
6+3	9	10	50 % erreicht
9+3	12	13	33 % erreicht
2+1	3	4	50 % erreicht
4+1	5	6	25 % erreicht
6+1	7	8	17 % erreicht
7+5	12	13	71 % erreicht

## Vor- und Nachteile sowie Anforderungen für Erasure Coding

Bevor Sie sich entscheiden, ob Sie zum Schutz von Objektdaten mithilfe von Replizierungs- oder Erasure Coding vor Verlust schützen möchten, sollten Sie die Vorteile und Nachteile sowie die Anforderungen für Verfahren zur Einhaltung von

Datenkonsistenz kennen.

## Vorteile von Erasure Coding

Im Vergleich zur Replizierung bietet das Verfahren zur Einhaltung von Datenkonsistenz (Erasure Coding) verbesserte Zuverlässigkeit, Verfügbarkeit und Storage-Effizienz.

- **Zuverlässigkeit:** Die Zuverlässigkeit wird in Bezug auf Fehlertoleranz gemessen - das ist die Anzahl der gleichzeitigen Ausfälle, die ohne Datenverlust aufrechterhalten werden können. Mithilfe der Replizierung werden mehrere identische Kopien auf unterschiedlichen Nodes und über mehrere Standorte hinweg gespeichert. Bei der Einhaltung von Datenkonsistenz wird ein Objekt in Daten- und Paritätsfragmente codiert und über viele Nodes und Standorte verteilt. Diese Verteilung bietet Schutz vor Standort- und Node-Ausfällen. Im Vergleich zur Replizierung bietet Erasure Coding eine höhere Zuverlässigkeit bei vergleichbaren Storage-Kosten.
- **Verfügbarkeit:** Verfügbarkeit kann definiert werden als die Möglichkeit, Objekte abzurufen, wenn Speicherknoten ausfallen oder unzugänglich werden. Im Vergleich zur Replizierung bietet Erasure Coding eine höhere Verfügbarkeit bei vergleichbaren Storage-Kosten.
- **Storage-Effizienz:** Für ein ähnliches Maß an Verfügbarkeit und Zuverlässigkeit benötigen die durch das Erasure Coding geschützten Objekte weniger Speicherplatz als die gleichen Objekte, wenn sie durch Replikation geschützt sind. Beispielsweise belegt ein 10-MB-Objekt, das an zwei Standorten repliziert wird, 20 MB Festplattenspeicher (zwei Kopien), während ein Objekt, das zur Fehlerkorrektur codiert wird, an drei Standorten mit einem 6+3-Erasure-Coding-Schema nur 15 MB Festplattenspeicher belegt.



Der Festplattenspeicher für Objekte, die mit Erasure-Coding-Verfahren codiert wurden, wird als Objektgröße und als Storage Overhead berechnet. Der prozentuale Storage Overhead entspricht der Anzahl der Paritätsfragmente, geteilt durch die Anzahl an Datenfragmenten.

## Nachteile des Erasure Coding

Im Vergleich zur Replizierung hat das Verfahren zur Einhaltung von Datenkonsistenz folgende Nachteile:

- Je nach Erasure Coding-Schema wird eine erhöhte Anzahl von Storage-Nodes und -Standorten empfohlen. Wenn Sie hingegen Objektdaten replizieren, benötigen Sie pro Kopie nur einen Storage Node. Siehe "[Verfahren zur Einhaltung von Datenkonsistenz für Storage-Pools mit drei oder mehr Standorten](#)" und "[Verfahren zur Einhaltung von Datenkonsistenz für Storage-Pools an einem Standort](#)".
- Höhere Kosten und Komplexität der Storage-Erweiterungen. Um eine Implementierung zu erweitern, bei der Replizierung verwendet wird, fügen Sie an jedem Ort, an dem Objektkopien erstellt werden, Storage-Kapazitäten hinzu. Um eine Implementierung zu erweitern, bei der Erasure Coding zum Einsatz kommt, müssen Sie sowohl das verwendete Verfahren zur Einhaltung von Datenkonsistenz als auch die Kapazität vorhandener Storage-Nodes in Betracht ziehen. Wenn Sie beispielsweise warten, bis die vorhandenen Nodes zu 100 % voll sind, müssen Sie mindestens Storage-Nodes hinzufügen  $k+m$ . Wenn Sie jedoch erweitern, wenn vorhandene Nodes zu 70 % voll sind, können Sie pro Standort zwei Nodes hinzufügen und gleichzeitig die nutzbare Storage-Kapazität maximieren. Weitere Informationen finden Sie unter "[Erweitern Sie Storage-Kapazität für Objekte, die nach dem Erasure-Coding-Verfahren codiert wurden](#)".
- Wenn Erasure Coding über geografisch verteilte Standorte hinweg verwendet wird, erhöht sich die Latenzzeiten beim Abruf. Die Objektfragmente für ein Objekt, das mit Erasure Coding versehen ist und über Remote-Standorte verteilt ist, benötigen über WAN-Verbindungen länger für den Abruf als ein Objekt, das repliziert und lokal verfügbar ist (der gleiche Standort, mit dem der Client eine Verbindung herstellt).
- Bei Verwendung von Erasure Coding für geografisch verteilte Standorte kommt ein höherer WAN-Netzwerkverkehr für Abrufvorgänge und Reparaturen zum Einsatz, insbesondere bei häufig abgerufenen Objekten oder bei Objektreparaturen über WAN-Netzwerkverbindungen.

- Wenn Sie standortübergreifend Erasure Coding verwenden, nimmt der maximale Objektdurchsatz ab, da die Netzwerklatenz zwischen Standorten zunimmt. Diese Abnahme ist auf die entsprechende Abnahme des TCP-Netzwerkdurchsatzes zurückzuführen, was sich darauf auswirkt, wie schnell das StorageGRID-System Objektfragmente speichern und abrufen kann.
- Höhere Auslastung von Computing-Ressourcen:

## Wann sollte das Erasure Coding verwendet werden

Das Verfahren zur Einhaltung von Datenkonsistenz eignet sich am besten für folgende Anforderungen:

- Objekte größer als 1 MB.



Das Verfahren zur Einhaltung von Datenkonsistenz eignet sich am besten für Objekte mit einer Größe von mehr als 1 MB. Verwenden Sie kein Erasure Coding für Objekte, die kleiner als 200 KB sind, um zu vermeiden, dass man sehr kleine Fragmente, die zur Fehlerkorrektur codiert wurden, managen muss.

- Langfristige oder kalte Storage-Lösung für selten abgerufene Inhalte
- Hohe Datenverfügbarkeit und -Zuverlässigkeit
- Schutz vor vollständigem Standort- und Node-Ausfall.
- Storage-Effizienz:
- Implementierungen an einem einzigen Standort, die eine effiziente Datensicherung benötigen und nur eine einzige Kopie mit Verfahren zur Einhaltung von Datenkonsistenz (Erasure Coding) als mehrere replizierte Kopien benötigen
- Implementierungen an mehreren Standorten, bei denen die Latenz zwischen den Standorten weniger als 100 ms beträgt

## Copyright-Informationen

Copyright © 2025 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

## Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.