



# ILM とオブジェクトライフサイクル StorageGRID

NetApp  
November 04, 2025

# 目次

ILM とオブジェクトライフサイクル	1
オブジェクトのライフサイクル全体にわたる ILM の動作	1
オブジェクトの取り込み方法	2
取り込みオプション	2
取り込みオプションのメリット、デメリット、および制限事項	4
オブジェクトの格納方法（レプリケーションまたはイレイジャーコーディング）	7
レプリケーションとは	7
シングルコピーレプリケーションを使用しない理由	8
イレイジャーコーディングとは	11
イレイジャーコーディングスキームとは	13
イレイジャーコーディングのメリット、デメリット、および要件	16
オブジェクト保持期間の決定方法	18
テナントユーザによるオブジェクト保持期間の制御方法	18
グリッド管理者によるオブジェクト保持期間の制御方法	18
S3 バケットライフサイクルと ILM の相互作用	18
オブジェクト保持の例	19
オブジェクトの削除方法	20
オブジェクトの削除に必要な時間	21
S3 バージョン管理オブジェクトの削除方法	22

# ILM とオブジェクトライフサイクル

## オブジェクトのライフサイクル全体にわたる ILM の動作

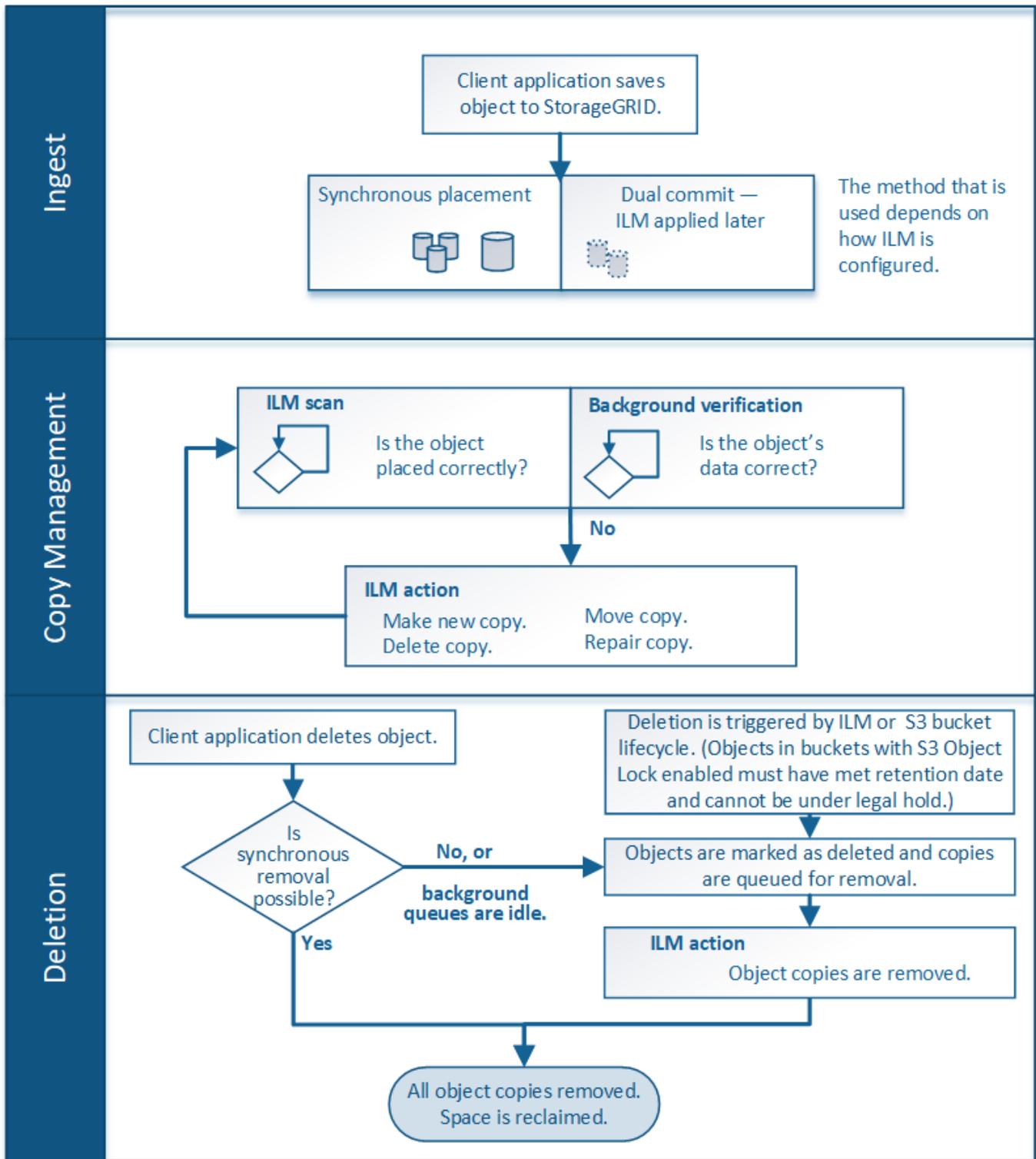
StorageGRID での ILM を使用したオブジェクト管理方法を理解することは、ポリシーをより効果的に設計するうえで役立ちます。

- \* 取り込み : S3 または Swift クライアントアプリケーションが StorageGRID システムへの接続を確立してオブジェクトを保存すると取り込みが開始され、StorageGRID がクライアントに「ingest successful」メッセージを返すと取り込みが完了します。ILM 要件の指定方法に応じて、ILM の手順を即座に適用（同期配置）するか、中間コピーを作成して ILM をあとから適用（デュアルコミット）することで、オブジェクトデータは取り込み時に保護されます。
- \* コピー管理 \* : ILM の配置手順に指定された数とタイプのオブジェクトコピーを作成すると、StorageGRID はオブジェクトの場所を管理し、オブジェクトを損失から保護します。
  - ILM のスキャンと評価 : StorageGRID は、グリッドに格納されているオブジェクトのリストを継続的にスキャンし、現在のコピーが ILM 要件を満たしているかどうかを確認します。タイプ、数、または場所が異なるオブジェクトコピーが必要となった場合、StorageGRID は必要に応じてコピーを作成、削除、または移動します。
  - バックグラウンド検証 : StorageGRID は、バックグラウンド検証を継続的に実行して、オブジェクトデータの整合性をチェックします。問題が検出されると、StorageGRID は、現在の ILM 要件を満たす場所に、新しいオブジェクトコピーまたは置き換え用のイレイジャーコーディングオブジェクトフラグメントを自動的に作成します。を参照してください ["オブジェクトの整合性を検証"](#)。
- \* オブジェクトの削除 \* : StorageGRID システムからすべてのコピーが削除されると、オブジェクトの管理は終了します。オブジェクトは、クライアントによる削除要求、または S3 バケットライフサイクルの終了が原因の ILM による削除または削除が原因で削除されます。



S3オブジェクトロックが有効になっているバケット内のオブジェクトは、リーガルホールドの対象になっている場合やretain-until-dateが指定されていてもまだ満たされていない場合は削除できません。

次の図は、オブジェクトのライフサイクル全体にわたる ILM の動作をまとめたものです。



## オブジェクトの取り込み方法

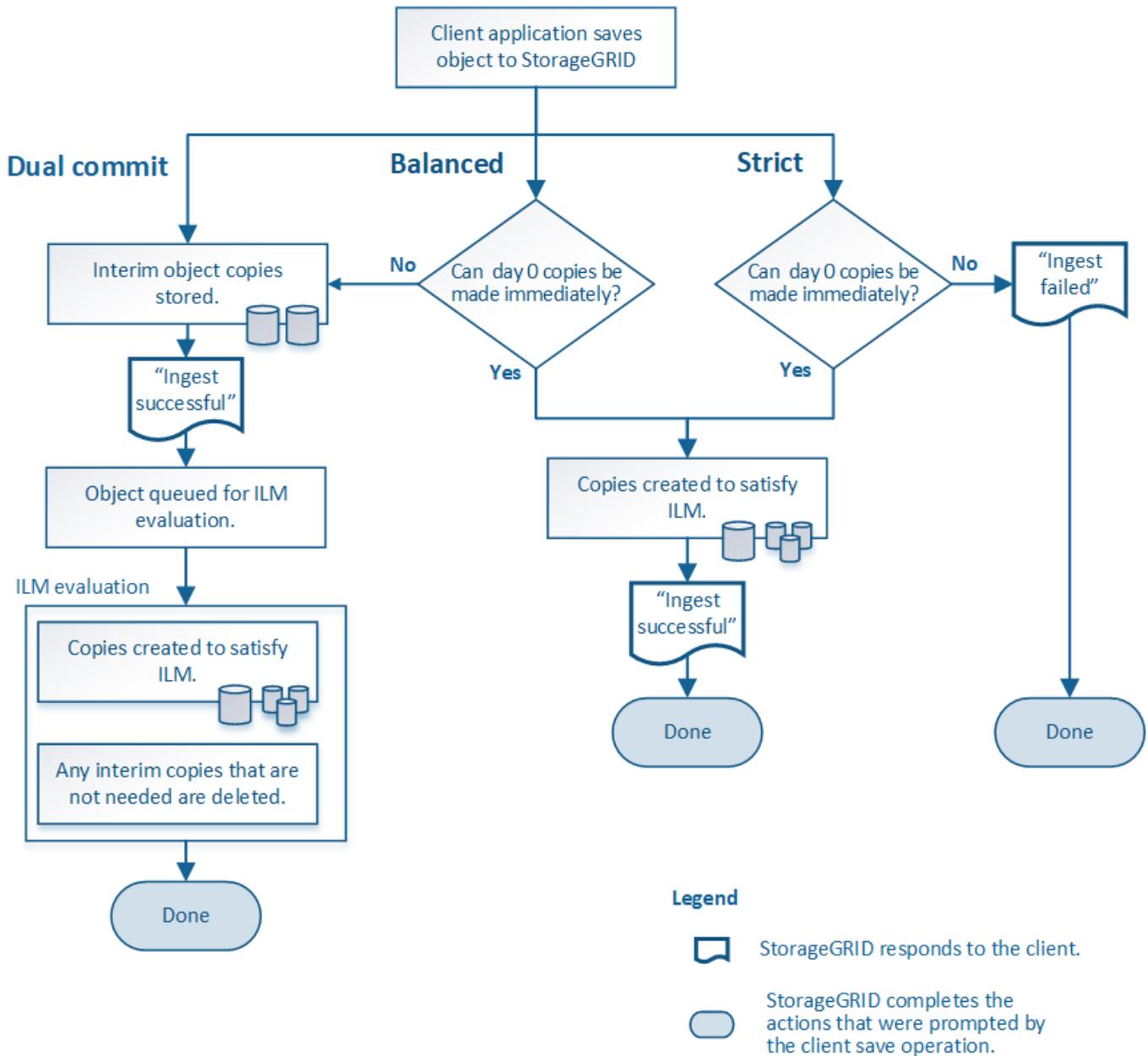
### 取り込みオプション

ILMルールを作成するときは、取り込み時にオブジェクトを保護するための3つのオプション（Dual commit、Strict、またはBalanced）のいずれかを指定します。

選択したオプションに応じて、StorageGRID は、中間コピーを作成してオブジェクトをキューに登録し、あとで ILM 評価を実行するか、または同期配置を使用してコピーをただちに作成して ILM 要件を満たします。

### 取り込みオプションのフローチャート

次のフローチャートは、3つの取り込みオプションのそれぞれを使用する ILM ルールにオブジェクトが一致した場合の動作を示しています。



### デュアルコミット

Dual commit オプションを選択すると、StorageGRID は2つの異なるストレージノード上に中間オブジェクトコピーをただちに作成し、「ingest successful」メッセージをクライアントに返します。オブジェクトは ILM 評価のキューに登録され、ルール of 配置手順を満たすコピーはあとで作成されます。

## Dual commit オプションを使用する状況

次のいずれかの場合に Dual commit オプションを使用します。

- マルチサイトの ILM ルールを使用しており、クライアントの取り込みレイテンシを考慮する必要があります。Dual commitを使用する場合は、デュアルコミットコピーがILMを満たしていない場合にデュアルコミットコピーを作成および削除する追加作業をグリッドで実行できるようにする必要があります。具体的には、
  - ILM のバックログが発生しないように、グリッドの負荷が十分に低い必要があります。
  - グリッドにハードウェアリソース（IOPS、CPU、メモリ、ネットワーク帯域幅など）が余剰である。
- マルチサイトの ILM ルールを使用していて、通常はサイト間の WAN 接続のレイテンシが高くなっているか、帯域幅が制限されている。このシナリオでは、Dual commit オプションを使用するとクライアントのタイムアウトを回避できます。Dual commit オプションを選択する前に、現実的なワークロードでクライアントアプリケーションをテストする必要があります。

## strict

Strict オプションを選択すると、StorageGRID は取り込み時に同期配置を使用してルールの配置手順で指定されたすべてのオブジェクトコピーをただちに作成します。必要なストレージの場所が一時的に使用できないなどの理由で、StorageGRID がすべてのコピーを作成できない場合、取り込みは失敗します。クライアントは処理を再試行する必要があります。

### Strict オプションを使用する場合

Strict オプションは、ILM ルールに指定された場所にもみオブジェクトをただちに格納するための運用または規制上の要件がある場合に使用してください。たとえば、規制要件を満たすために、Strictオプションと高度なフィルタ「Location Constraint」を使用して、特定のデータセンターにオブジェクトが格納されないようにする必要があります。

を参照してください "[例 5：取り込み動作が Strict の場合の ILM ルールとポリシー](#)"。

## Balanced（デフォルト）

Balanced オプションを選択した場合も、StorageGRID は、取り込み時に同期配置を使用してルールの配置手順で指定されたすべてのコピーをただちに作成します。Strictオプションとは対照的に、すべてのコピーをただちに作成できない場合、StorageGRID は代わりにDual commitを使用します。

### Balanced オプションを使用する状況

Balanced オプションは、データ保護、グリッドパフォーマンス、および取り込みの成功の最適な組み合わせを実現するために使用します。Balancedは、Create ILM Ruleウィザードのデフォルトのオプションです。

## 取り込みオプションのメリット、デメリット、および制限事項

取り込み時にデータを保護するための3つのオプション（Balanced、Strict、Dual commit）のそれぞれのメリットとデメリットを理解することは、ILM ルールに選択するオプションを決定する際に役立ちます。

取り込みオプションの概要については、を参照してください "[取り込みオプション](#)"。

## Balanced オプションと Strict オプションのメリット

取り込み時に中間コピーを作成する Dual commit と比較すると、2つの同期配置オプションには次のメリットがあります。

- **\* Better データ セキュリティ \*** : オブジェクトデータは、ILM ルールの配置手順に従ってただちに保護されます。配置手順は、複数の格納場所の障害など、さまざまな障害状況からオブジェクトを保護するように設定できます。Dual commit で保護できるのは、単一のローカルコピーの損失のみです。
- **\* グリッド処理の効率化 \*** : 各オブジェクトは、取り込み時に 1 回だけ処理されます。StorageGRID システムで中間コピーを追跡または削除する必要がないため、処理の負荷が軽減され、消費されるデータベーススペースも少なくて済みます。
- **\* ( Balanced ) Recommended \*** : Balanced オプションは、最適な ILM 効率を実現します。Strict 取り込み動作が必要な場合、またはグリッドが Dual commit の使用条件をすべて満たしている場合を除き、Balanced オプションを使用することを推奨します。
- **\* ( Strict ) オブジェクトの場所が明らか \*** : Strict オプションは、ILM ルールの配置手順に従ってオブジェクトがただちに格納されることを保証します。

## Balanced オプションと Strict オプションのデメリット

Dual commit と比較すると、Balanced オプションと Strict オプションにはいくつかのデメリットがあります。

- **\* クライアントの取り込み時間が長くなる \*** : クライアントの取り込みレイテンシが長くなる可能性があります。Balanced オプションまたは Strict オプションを使用した場合、すべてのイレイジャーコーディングフラグメントまたはレプリケートコピーが作成されて格納されるまで、「ingest successful」メッセージはクライアントに返されません。しかし、ほとんどの場合、オブジェクトデータは最終的な配置までの時間をはるかに短縮できます。
- **( Strict ) 取り込みエラーの発生率が高い** : Strict オプションを使用すると、StorageGRID が ILM ルールで指定されたすべてのコピーをすぐに作成できない場合に取り込みが失敗します。必要なストレージの場所が一時的にオフラインになっている場合や、ネットワークでサイト間のオブジェクトコピーが原因で遅延している場合には、取り込みに失敗する可能性が高くなります。
- **( Strict ) S3 マルチパートアップロードでは、状況によっては想定どおりに配置されない可能性がある \*** : Strict では、オブジェクトが ILM ルールの指定どおりに配置されるか、あるいは取り込みが失敗するかのどちらかの結果が想定されます。ただし、S3 マルチパートアップロードの場合は、オブジェクトの各パートの取り込み時に ILM が評価され、マルチパートアップロードの完了時にオブジェクト全体に対して ILM が評価されます。そのため、次の状況では想定どおりに配置されないことがあります。
  - **\* S3 マルチパートアップロードの実行中に ILM が変更された場合 \*** : 各パートはその取り込み時にアクティブなルールに従って配置されるため、マルチパートアップロードが完了した時点でオブジェクトの一部のパートが現在の ILM 要件を満たしていない可能性があります。この場合、オブジェクトの取り込みは失敗しません。代わりに、正しく配置されていないパートは ILM ルールによる再評価のためにキューに登録され、あとで正しい場所に移動されます。
  - **\* ILM ルールがサイズでフィルタリングする場合 \*** : パーツに対して ILM を評価する際、StorageGRID はオブジェクトのサイズではなくパーツのサイズでフィルタリングします。つまり、オブジェクト全体の ILM 要件を満たしていない場所にオブジェクトの一部を格納できます。たとえば、10GB 以上のオブジェクトをすべて DC1 に格納し、それより小さいオブジェクトをすべて DC2 に格納するルールの場合、10 パートからなるマルチパートアップロードの 1GB の各パートは取り込み時に DC2 に格納されます。オブジェクトに対して ILM が評価されると、オブジェクトのすべてのパートが DC1 に移動されます。
- **( Strict ) オブジェクトタグまたはメタデータが更新され、新たに必要となった配置を実行できなくて**

も取り込みが失敗しない \* : Strict では、オブジェクトが ILM ルールの指定どおりに配置されるか、あるいは取り込みが失敗するかのどちらかの結果が想定されます。ただし、グリッドにすでに格納されているオブジェクトのメタデータまたはタグを更新しても、オブジェクトは再取り込みされません。つまり、更新によってトリガーされたオブジェクト配置の変更はすぐには行われません。通常のバックグラウンド ILM プロセスで ILM が再評価されると、配置変更が行われます。必要な配置変更ができない場合（新たに必要な場所が使用できない場合など）、更新されたオブジェクトは配置変更が可能になるまで現在の配置を保持します。

## Balanced オプションと Strict オプションを使用したオブジェクトの配置に関する制限事項

Balanced オプションまたは Strict オプションは、次のいずれかの配置手順を含む ILM ルールには使用できません。

- クラウドストレージプールへの配置：0 日目
- アーカイブノードへの配置：0 日目
- クラウドストレージプールまたはアーカイブノードへの配置（ルールの作成時間が [Reference Time] に設定されている場合）。

これらの制限事項は、StorageGRID がクラウドストレージプールまたはアーカイブノードに同期的にコピーを作成できず、ユーザが定義した作成時間が現在の状態になる可能性があるためです。

## ILM ルールと整合性制御がデータ保護に与える影響

ILM ルールと選択した整合性制御は、どちらもオブジェクトの保護方法に影響します。これらの設定は対話的に操作できます。

たとえば、ILM ルールに対して選択した取り込み動作はオブジェクトコピーの初期配置に影響し、オブジェクトの格納時に使用される整合性制御はオブジェクトメタデータの初期配置に影響します。StorageGRID では、クライアント要求に対応するためにオブジェクトのデータとメタデータの両方にアクセスするため、整合性レベルと取り込み動作で同じ保護レベルを選択すると、初期データ保護が向上し、システム応答の予測性が向上します。

StorageGRID で使用できる整合性制御の概要を以下に示します。

- \* all \* : すべてのノードが即座にオブジェクトメタデータを受け取り、受け取れない場合は要求が失敗します。
- \* strong-global \* : オブジェクトのメタデータがすべてのサイトにただちに分散されます。すべてのサイトのすべてのクライアント要求について、リードアフターライト整合性が保証されます。
- \* strong-site \* : オブジェクトのメタデータがただちにサイトの他のノードに分散されます。1 つのサイト内のすべてのクライアント要求について、リードアフターライト整合性が保証されます。
- \* read-after-new-write \* : 新規オブジェクトについてはリードアフターライト整合性が提供され、オブジェクトの更新については結果整合性が提供されます。高可用性が確保され、データ保護が保証されます。ほとんどの場合に推奨されます。
- \* available \* : 新しいオブジェクトとオブジェクトの更新の両方について、結果整合性を提供します。S3 バケットの場合は、必要な場合にのみ使用します（読み取り頻度の低いログ値を含むバケットや、存在しないキーに対する HEAD 処理や GET 処理など）。S3 FabricPool バケットではサポートされません。



整合性レベルを選択する前に、の手順に記載されている整合性制御の完全な概要をお読みください ["S3 REST API を使用する"](#)。デフォルト値を変更する前に、利点と制限事項を理解しておく必要があります。

## 整合性制御と ILM ルールの連動の例

次の ILM ルールと次の整合性レベル設定の 2 サイトグリッドがあるとします。

- \* ILM ルール \* : ローカルサイトとリモートサイトに 1 つずつ、2 つのオブジェクトコピーを作成します。Strict 取り込み動作が選択されています。
- \* 整合性レベル \*: "Strong-GLOBAL" ( オブジェクトメタデータはすべてのサイトにただちに分散されます )

クライアントがオブジェクトをグリッドに格納すると、StorageGRID は両方のオブジェクトをコピーし、両方のサイトにメタデータを分散してからクライアントに成功を返します。

オブジェクトは、取り込みが成功したことを示すメッセージが表示された時点で損失から完全に保護されません。たとえば、取り込み直後にローカルサイトが失われた場合、オブジェクトデータとオブジェクトメタデータの両方のコピーがリモートサイトに残っています。オブジェクトを完全に読み出し可能にしている。

代わりに同じ ILM ルールと「strong-site」整合性レベルを使用する場合は、オブジェクトデータがリモートサイトにレプリケートされたあとで、オブジェクトメタデータがそこに分散される前に、クライアントに成功メッセージが送信される可能性があります。この場合、オブジェクトメタデータの保護レベルがオブジェクトデータの保護レベルと一致しません。取り込み直後にローカルサイトが失われると、オブジェクトメタデータが失われます。オブジェクトを取得できません。

整合性レベルと ILM ルール間の関係は複雑になる可能性があります。サポートが必要な場合は、ネットアップにお問い合わせください。

### 関連情報

- ["例 5 : 取り込み動作が Strict の場合の ILM ルールとポリシー"](#)

## オブジェクトの格納方法（レプリケーションまたはイレイジャーコーディング）

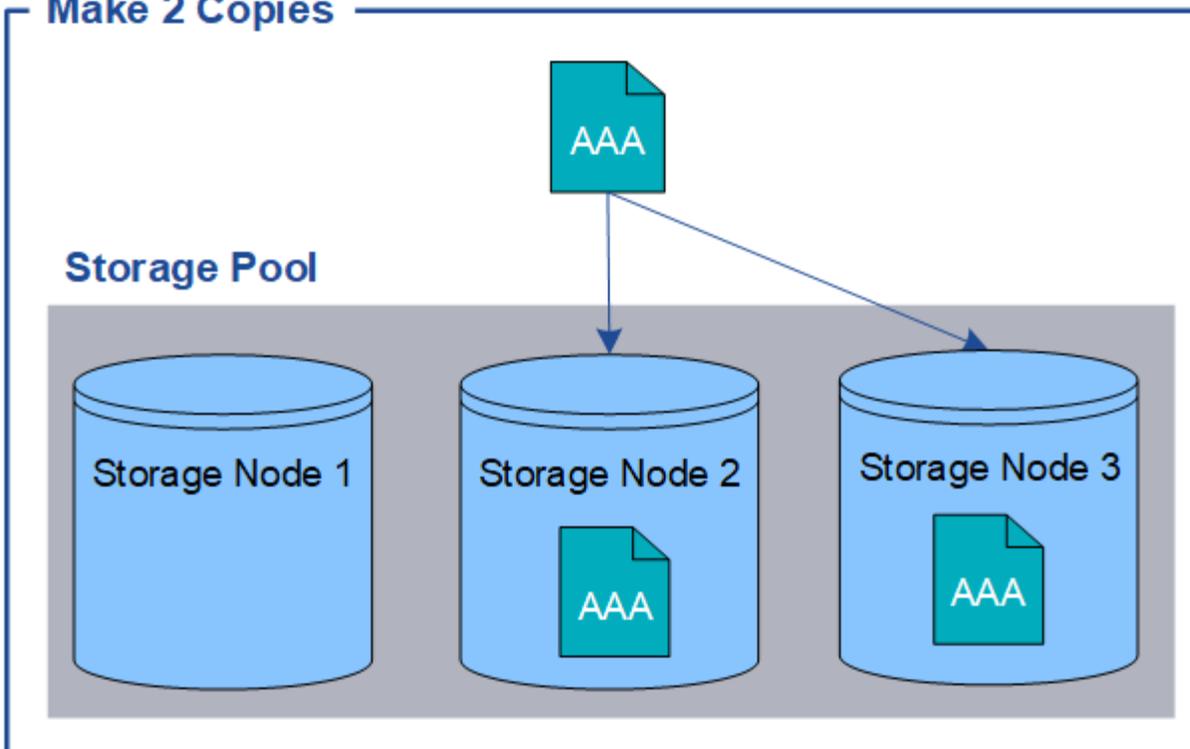
### レプリケーションとは

レプリケーションは、StorageGRID がオブジェクトデータを格納するために使用する 2 つの方法のうちの一つです。レプリケーションを使用する ILM ルールにオブジェクトが一致すると、オブジェクトデータの完全なコピーが作成され、ストレージノードまたはアーカイブノードに格納されます。

レプリケートコピーを作成するように ILM ルールを設定する場合は、作成するコピーの数、コピーを配置する場所、およびそれぞれの場所にコピーを格納する期間を指定します。

次の例の ILM ルールは、各オブジェクトのレプリケートコピーを 2 つずつ、3 つのストレージノードからなるストレージプールに配置するように指定されています。

## Make 2 Copies



このルールにオブジェクトが一致した場合、StorageGRID はオブジェクトのコピーを 2 つ作成して、ストレージプール内の別々のストレージノードにそれぞれのコピーを配置します。この 2 つのコピーは、使用可能な 3 つのストレージノードのうちいずれか 2 つに配置されます。この場合、ストレージノード 2 と 3 に配置されています。コピーは 2 つあるため、ストレージプール内のいずれかのノードで障害が発生した場合でもオブジェクトを読み出すことができます。



StorageGRID が任意のストレージノードに格納できるレプリケートコピーは 1 つのオブジェクトにつき 1 つだけです。グリッドにストレージノードが 3 つあり、4 コピーの ILM ルールを作成した場合、作成されるコピーはストレージノードごとに 1 つだけになります。ILM placement unAchievable \* アラートがトリガーされ、ILM ルールを完全に適用できなかったことを示します。

### 関連情報

- ["イレイジャーコーディングとは"](#)
- ["ストレージプールとは"](#)
- ["レプリケーションとイレイジャーコーディングを使用してサイト障害から保護"](#)

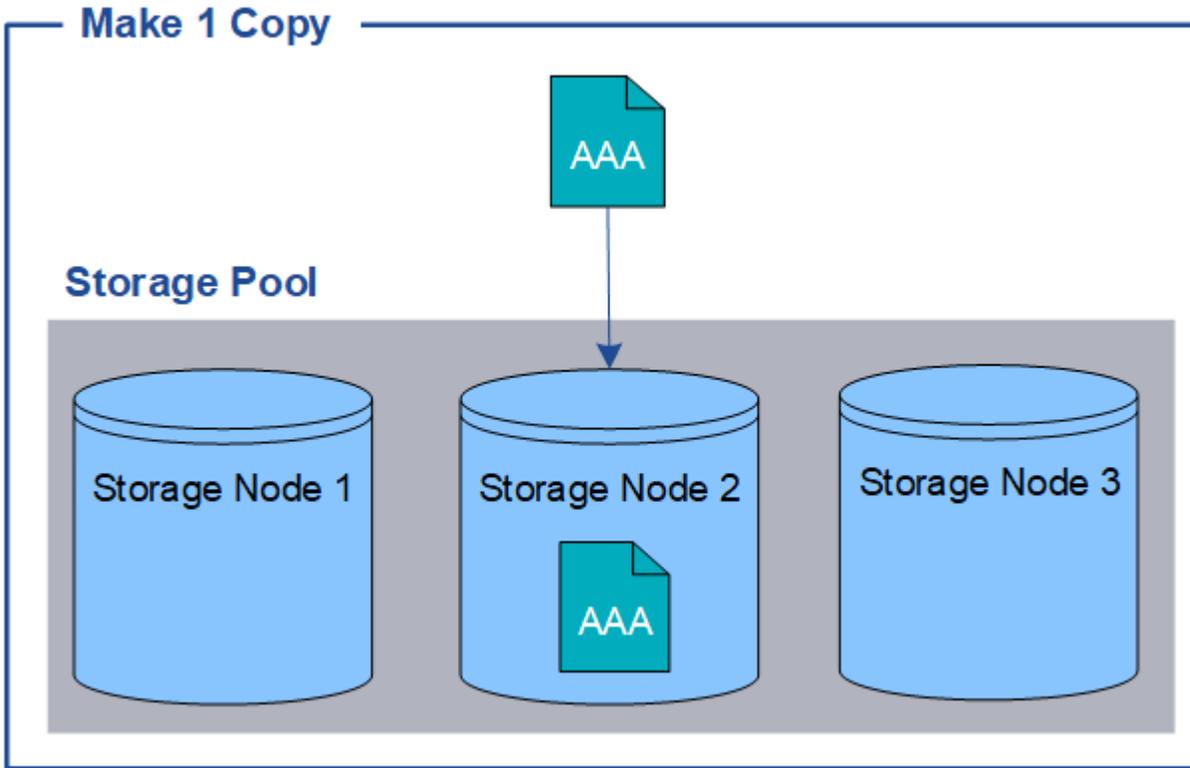
### シングルコピーレプリケーションを使用しない理由

レプリケートコピーを作成する ILM ルールを作成するときは、配置手順の任意の期間に少なくとも 2 つのコピーを指定する必要があります。

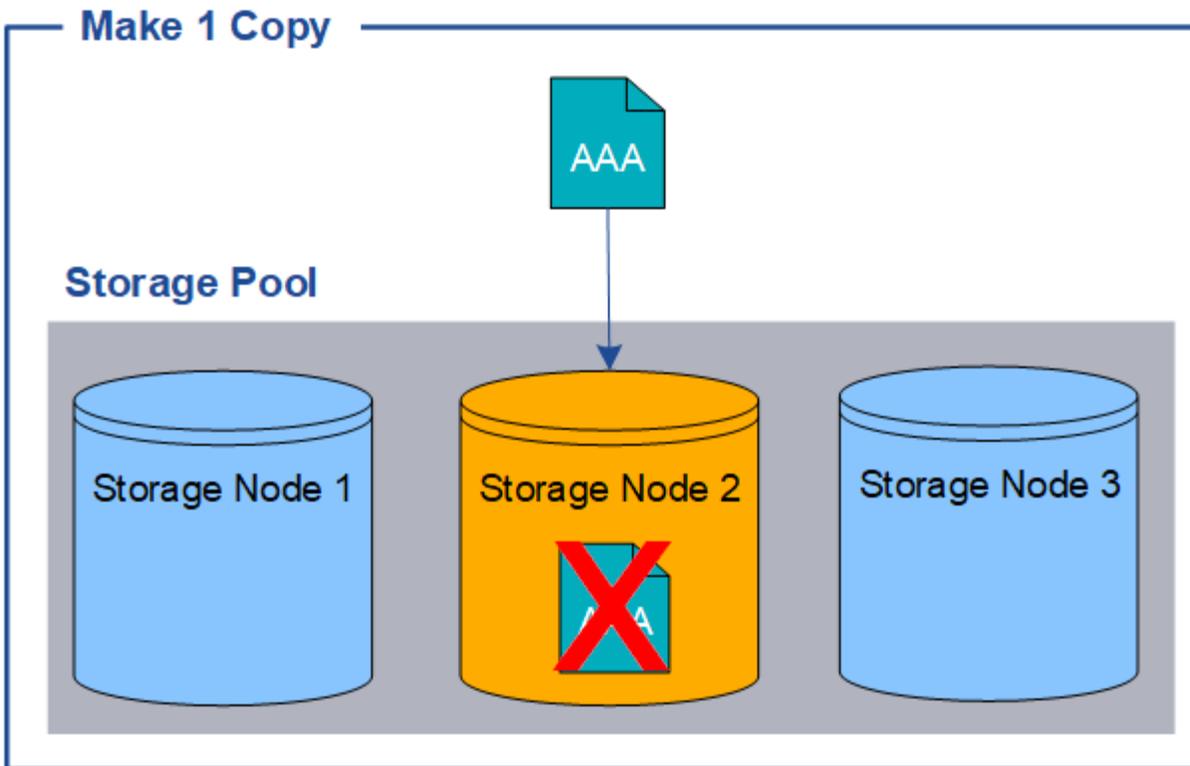


任意の期間にレプリケートコピーを 1 つだけ作成する ILM ルールは使用しないでください。オブジェクトのレプリケートコピーが 1 つしかない場合、ストレージノードに障害が発生したり、重大なエラーが発生すると、そのオブジェクトは失われます。また、アップグレードなどのメンテナンス作業中は、オブジェクトへのアクセスが一時的に失われます。

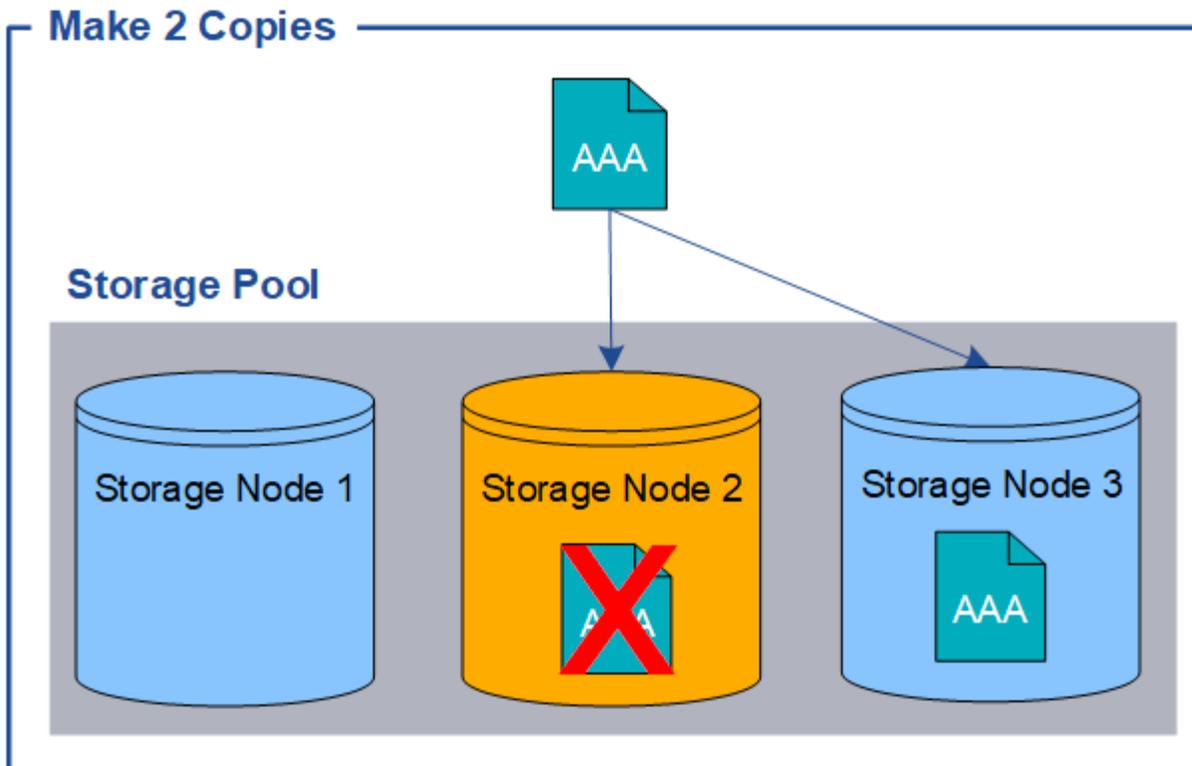
次の例では、Make 1 Copy ILM ルールによって、1つのオブジェクトのレプリケートコピーを3つのストレージノードからなるストレージプールに配置するように指定しています。このルールに一致するオブジェクトが取り込まれると、StorageGRID は1つのストレージノードにのみコピーを配置します。



ILM ルールにオブジェクトのレプリケートコピーが1つしか作成されていない場合、ストレージノードが使用できなくなるとオブジェクトにアクセスできなくなります。この例では、アップグレードやその他のメンテナンス手順の実行中など、ストレージノード2がオフラインになるとオブジェクトAAAへのアクセスが一時的に失われます。ストレージノード2で障害が発生すると、オブジェクトAAAが完全に失われます。



オブジェクトデータの損失を防ぐには、レプリケーションで保護するすべてのオブジェクトのコピーを常に2つ以上作成する必要があります。コピーが複数ある場合も、1つのストレージノードに障害が発生した場合やオフラインになった場合でもオブジェクトにアクセスできます。



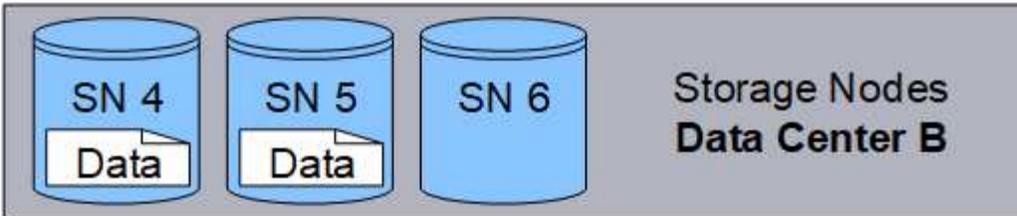
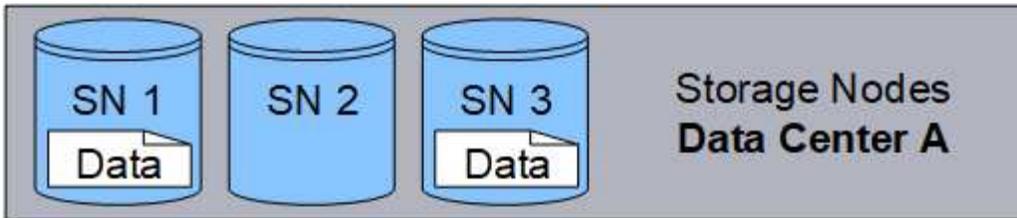
## イレイジャーコーディングとは

イレイジャーコーディングは、StorageGRID がオブジェクトデータを格納するために使用する2つの方法のうちの一つです。イレイジャーコーディングを使用するILMルールにオブジェクトが一致した場合、それらのオブジェクトはデータフラグメントにスライスされ、追加のパリティフラグメントが計算されて、各フラグメントが別々のストレージノードに格納されます。

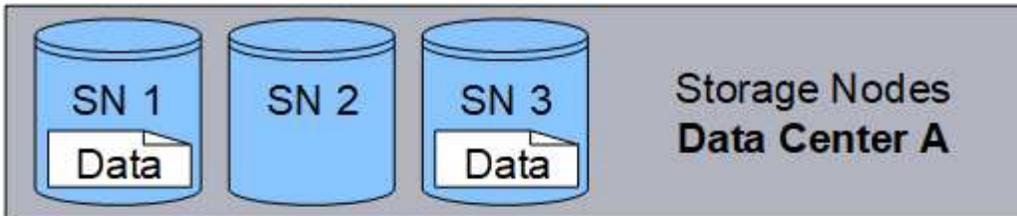
アクセスされたオブジェクトは、格納されたフラグメントを使用して再アSEMBルされます。データフラグメントまたはパリティフラグメントが破損したり失われたりしても、イレイジャーコーディングアルゴリズムが残りのデータフラグメントとパリティフラグメントを使用してそのフラグメントを再作成します。

ILMルールを作成すると、それらのルールをサポートするイレイジャーコーディングプロファイルがStorageGRID によって作成されます。イレイジャーコーディングプロファイルのリストを表示できます。"[イレイジャーコーディングプロファイルの名前を変更します](#)"または"[イレイジャーコーディングプロファイルがどのILMルールでも使用されていない場合は、非アクティブ化します](#)"。

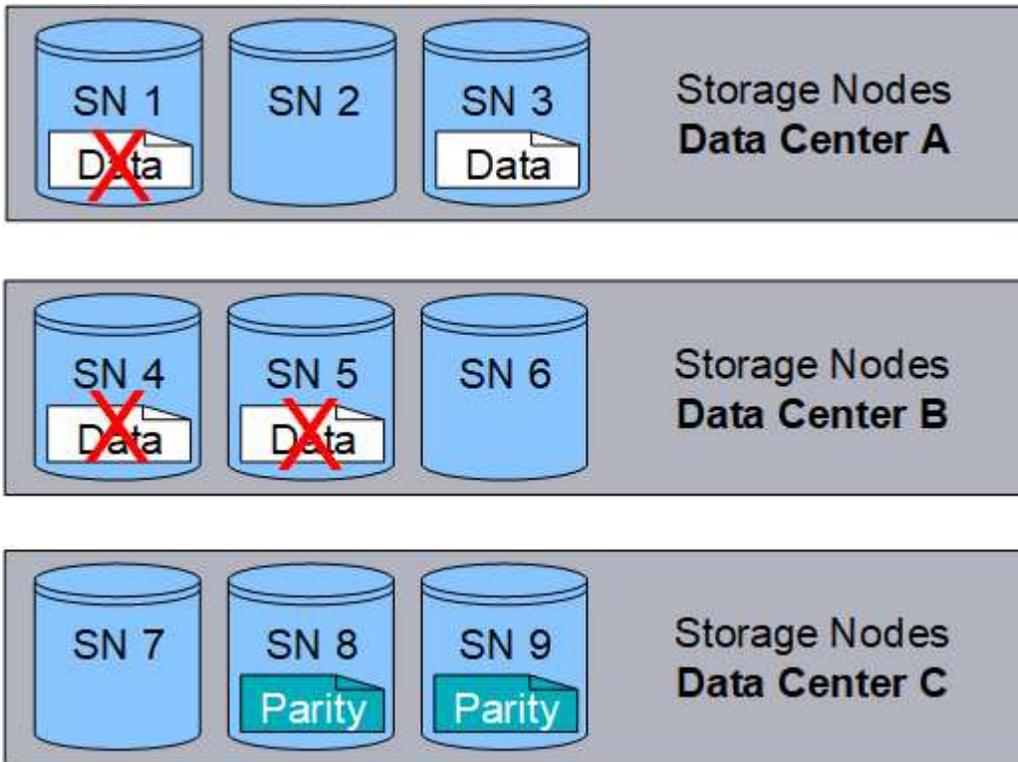
次の例は、オブジェクトのデータに対するイレイジャーコーディングアルゴリズムの使用法を示しています。この例の ILM ルールでは4+2のイレイジャーコーディングスキームを使用します。各オブジェクトは4つのデータフラグメントに等分され、オブジェクトデータから2つのパリティフラグメントが計算されます。ノードやサイトの障害時にもデータが保護されるよう、6つの各フラグメントは3つのデータセンターサイトの別々のノードに格納されます。



4+2レイジャーコーディングスキームはさまざまな方法で設定できます。たとえば、6つのストレージノードで構成される単一サイトのストレージプールを設定できます。の場合 "[サイト障害からの保護](#)"では、3つのサイトで構成されるストレージプールを使用し、各サイトに3つのストレージノードを配置できます。6つのうちのいずれか4つのフラグメント（データまたはパリティ）が使用可能であれば、オブジェクトを読み出すことができます。最大2つのフラグメントが失われても、オブジェクトデータが失われることはありません。サイト全体が失われても、他のすべてのフラグメントに引き続きアクセスできるかぎり、オブジェクトの読み出しまたは修復が可能です。



3つ以上のストレージノードが失われると、オブジェクトを読み出せなくなります。



#### 関連情報

- ["レプリケーションとは"](#)
- ["ストレージプールとは"](#)
- ["イレイジャーコーディングスキームとは"](#)
- ["イレイジャーコーディングプロファイルの名前を変更します"](#)
- ["イレイジャーコーディングプロファイルを非アクティブ化します"](#)

#### イレイジャーコーディングスキームとは

イレイジャーコーディングスキームは、各オブジェクト用に作成されるデータフラグメントとパリティフラグメントの数を制御します。

ILMルールにイレイジャーコーディングプロファイルを設定する場合は、使用するストレージプールを構成するストレージノードとサイトの数に基づいて、使用可能なイレイジャーコーディングスキームを選択します。

StorageGRID システムは、Reed-Solomon イレイジャーコーディングアルゴリズムを使用します。アルゴリズムはオブジェクトを  $k$  データのフラグメント化と  $m$  パリティフラグメントにスライスします。  $k + m = n$  フラグメントは全体に分散されます。  $n$  データ保護を提供するストレージノード。オブジェクトは最大で維持できます  $m$  フラグメントの損失または破損オブジェクトを読み出したり修復するには、次の手順に従います。  $k$  フラグメントが必要です。

イレイジャーコーディングコピーを作成するルールに使用するストレージプールを選択する場合は、ストレージプールについて次のガイドラインに従ってください。

- ストレージプールには3つ以上のサイト、または1つのサイトだけが含まれている必要があります。



ストレージプールにサイトが2つ含まれている場合はイレイジャーコーディングを使用できません。

- 3つ以上のサイトを含むストレージプールのイレイジャーコーディングスキーム
- 1サイトのストレージプールのイレイジャーコーディングスキーム

- デフォルトのサイトである[All Sites]を含むストレージプールは使用しないでください。
- ストレージプールには少なくともを含める必要があります  $k+m + 1$  ストレージノード：

必要なストレージノードの最小数は  $k+m$ 。ただし、必要なストレージノードが一時的に使用できない場合に、少なくとも1つのストレージノードを追加することで、取り込みエラーやILMバックログが発生するのを防ぐことができます。

イレイジャーコーディングスキームのストレージオーバーヘッドは、パリティフラグメントの数を除算して計算されます ( $m$ ) をデータフラグメント数で計算します ( $k$ )。ストレージオーバーヘッドを使用して、各イレイジャーコーディングオブジェクトに必要なディスクスペースを計算できます。

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

たとえば、4+2スキームを使用して10MBのオブジェクト（ストレージオーバーヘッドが50%）を格納すると、そのオブジェクトが消費するグリッドストレージは15MBです。6+3のストレージオーバーヘッドを含む6+2スキームを使用して同じ10MBのオブジェクトを格納すると、オブジェクトが消費するサイズは約13.3MBになります。

合計値が最も小さいイレイジャーコーディングスキームを選択します  $k+m$  それはあなたのニーズに合っています。フラグメント数が少ないイレイジャーコーディングスキームは、オブジェクトごとに作成されて分散（または読み出される）フラグメント数が少ないため、全体的に計算効率が高く、フラグメントサイズが大きい場合パフォーマンスも向上します。また、ストレージの追加が必要になった場合に拡張で追加するノード数も少なく済みます。（ストレージの拡張計画の詳細については、["StorageGRIDの拡張手順"](#)。）

### 3つ以上のサイトを含むストレージプールのイレイジャーコーディングスキーム

次の表に、3つ以上のサイトを含むストレージプールについて、StorageGRIDで現在サポートされているイレイジャーコーディングスキームを示します。これらのスキームはすべて、サイト障害からの保護を提供します。1つのサイトが失われてもオブジェクトには引き続きアクセスできます。

サイト障害からの保護を提供するイレイジャーコーディングスキームの場合、ストレージプール内の推奨されるストレージノード数がを超えています  $k+m + 1$  各サイトに少なくとも3つのストレージノードが必要であるためです。

イレイジャーコーディングスキーム ( $k+m$ )	サイトの最小数	各サイトで推奨されるストレージノードの数	推奨されるストレージノードの総数	サイト障害からの保護	ストレージオーバーヘッド
4+2	3.	3.	9.	はい。	50%
6+2	4.	3.	12.	はい。	33%
8+2	5.	3.	15	はい。	25%

イレイジャーコーディングスキーム ( $k+m$ )	サイトの最小数	各サイトで推奨されるストレージノードの数	推奨されるストレージノードの総数	サイト障害からの保護	ストレージオーバーヘッド
6+3	3.	4.	12.	はい。	50%
9+3	4.	4.	16	はい。	33%
2+1	3.	3.	9.	はい。	50%
4+1	5.	3.	15	はい。	25%
6+1	7.	3.	21	はい。	17%
7+5	3.	5.	15	はい。	71%



StorageGRID では、サイトごとに少なくとも 3 つのストレージノードが必要です。7+5 スキームを使用するには、各サイトに少なくとも 4 つのストレージノードが必要。サイトごとに 5 つのストレージノードを使用することを推奨します。

サイト保護を提供するイレイジャーコーディングスキームを選択する場合は、次の要素の相対的な重要性を調整します。

- \* フラグメント数 \* : フラグメントの総数が少ないほど、一般にパフォーマンスと拡張の柔軟性が向上します。
- フォールトトレランス : パリティセグメントの数が増える (つまり、フォールトトレランスが向上することで、フォールトトレランスが向上します  $m$  の値が大きくなります) 。
- ネットワークトラフィック : フラグメント数が多い (の合計数が多い) スキームを使用して、障害からリカバリする場合  $k+m$ ) より多くのネットワークトラフィックを作成します。
- \* ストレージ・オーバーヘッド \* : オーバーヘッドの大きいスキームでは、オブジェクトごとにより多くのストレージ・スペースが必要です。

たとえば、4+2 と 6+3 のどちらかのスキーム (どちらも 50% のストレージオーバーヘッドがある) を選ぶ場合、フォールトトレランスをさらに高める必要がある場合は 6+3 のスキームを選択します。ネットワークリソースが制限されている場合は、4+2 のスキームを選択します。他のすべての要素が等しい場合は、フラグメントの合計数が少ないため、4+2 を選択します。



使用するスキームが不明な場合は、4+2 または 6+3 を選択するか、テクニカルサポートにお問い合わせください。

## 1 サイトのストレージプールのイレイジャーコーディングスキーム

1 サイトのストレージプールでは、サイトに十分な数のストレージノードがある場合、3 つ以上のサイト用に定義されたすべてのイレイジャーコーディングスキームがサポートされます。

必要なストレージノードの最小数は  $k+m$  がありますが、ストレージプールには  $k+m+1$  ストレージノードを推奨します。たとえば、2+1 イレイジャーコーディングスキームには少なくとも 3 つの

ストレージノードからなるストレージプールが必要ですが、推奨されるストレージノード数は4つです。

イレイジャーコーディングスキーム ( $k+m$ )	ストレージノードの最小数	推奨されるストレージノードの数	ストレージオーバーヘッド
4+2	6.	7.	50%
6+2	8.	9.	33%
8+2	10.	11.	25%
6+3	9.	10.	50%
9+3	12.	13.	33%
2+1	3.	4.	50%
4+1	5.	6.	25%
6+1	7.	8.	17%
7+5	12.	13.	71%

## イレイジャーコーディングのメリット、デメリット、および要件

レプリケーションとイレイジャーコーディングのどちらを使用してオブジェクトデータを損失から保護するかを決定する前に、イレイジャーコーディングのメリット、デメリット、および要件を理解しておく必要があります。

### イレイジャーコーディングのメリット

イレイジャーコーディングは、レプリケーションに比べて信頼性、可用性、ストレージ効率に優れています。

- \* 信頼性 \***：信頼性はフォールトトレランス、つまり同時にデータを失うことなく維持できる障害の数によって判断されます。レプリケーションでは、複数の同一コピーが異なるノード上およびサイト間に格納されます。イレイジャーコーディングの場合、オブジェクトはデータフラグメントとパリティフラグメントにエンコードされ、多数のノードとサイトに分散されます。この分散によってサイトとノード両方の障害からの保護を提供します。イレイジャーコーディングは、同等のストレージコストでレプリケーションよりも優れた信頼性を提供します。
- \* 可用性 \***：可用性は、ストレージノードに障害が発生した場合や、ノードにアクセスできなくなった場合にオブジェクトを読み出すことができるかどうかによって定義されます。イレイジャーコーディングは、同等のストレージコストでレプリケーションよりも優れた可用性を提供します。
- \* Storage Efficiency \***：可用性と信頼性が同等レベルの場合、イレイジャーコーディングで保護されたオブジェクトが消費するディスクスペースは、同じオブジェクトをレプリケーションで保護する場合よりも少なくなります。たとえば、10MBのオブジェクトを2つのサイトにレプリケートするとディスクスペースを20MB（2つのコピー）消費しますが、6+3のイレイジャーコーディングスキームを使用して3つのサイトにイレイジャーコーディングされたオブジェクトが消費するディスクスペースは15MBのみです。



イレイジャーコーディングオブジェクトのディスクスペースは、オブジェクトサイズにストレージオーバーヘッドを加えたものです。ストレージオーバーヘッドの割合は、パーティフラグメント数をデータフラグメント数で割って算出します。

## イレイジャーコーディングのデメリット

レプリケーションと比較した場合のイレイジャーコーディングのデメリットは次のとおりです。

- イレイジャーコーディングスキームに応じて、ストレージノードとサイトの数を増やすことを推奨します。一方、オブジェクトデータをレプリケートする場合、コピーごとに必要なストレージノードは1つだけです。を参照してください ["3つ以上のサイトを含むストレージプールのイレイジャーコーディングスキーム"](#) および ["1サイトのストレージプール用のイレイジャーコーディングスキーム"](#)。
- ストレージの拡張にかかるコストと複雑さが増大します。レプリケーションを使用する環境を拡張するには、オブジェクトコピーを作成するすべての場所にストレージ容量を追加します。イレイジャーコーディングを使用する環境を拡張する場合は、使用中のイレイジャーコーディングスキームと、既存のストレージノードの使用率の両方を考慮する必要があります。たとえば、既存のノードが100%フルになるまで待つ場合は、少なくともを追加する必要があります  $k+m$  ストレージノード。ただし、既存のノードの使用率が70%に達した時点で拡張する場合は、サイトごとにノードを2つ追加しても、使用可能なストレージ容量を最大化できます。詳細については、を参照してください ["イレイジャーコーディングオブジェクトのストレージ容量を追加します"](#)。
- 地理的に分散したサイトでイレイジャーコーディングを使用する場合は、読み出しのレイテンシが上昇します。イレイジャーコーディングされてリモートサイトに分散されたオブジェクトのフラグメントを WAN 接続経由で読み出す場合、レプリケートされてローカル（クライアントの接続先と同じサイト）で利用可能なオブジェクトよりも時間がかかります。
- 地理的に分散したサイトでイレイジャーコーディングを使用する場合は、特に WAN ネットワーク接続経由でオブジェクトを頻繁に読み出ししたり修復したりするケースでは読み出しと修復の WAN ネットワークトラフィックが増大します。
- サイト間でイレイジャーコーディングを使用する場合は、サイト間のネットワークレイテンシの上昇に伴ってオブジェクトの最大スループットが大幅に低下します。この最大スループットの低下は TCP ネットワークのスループットが低下したことによるもので、StorageGRID システムによるオブジェクトフラグメントの格納 / 読み出し速度に影響します。
- コンピューティングリソースの利用率が向上します。

## イレイジャーコーディングを使用する状況

イレイジャーコーディングは次の要件に最適です。

- 1MB 超のオブジェクト



イレイジャーコーディングは 1MB を超えるオブジェクトに適しています。非常に小さいイレイジャーコーディングフラグメントを管理するオーバーヘッドを回避するため、200KB未満のオブジェクトにはイレイジャーコーディングを使用しないでください。

- 頻繁に読み出されないコンテンツの長期保存またはコールドストレージ。
- 高いデータ可用性と信頼性。
- サイトやノードの障害に対する保護
- ストレージ効率

- 複数のレプリケートコピーではなく 1 つのイレイジャーコーディングコピーのみを使用して効率的にデータを保護する必要のある単一サイト環境
- サイト間レイテンシが 100 ミリ秒未満の複数サイト環境

## オブジェクト保持期間の決定方法

StorageGRID には、グリッド管理者と個々のテナントユーザが、オブジェクトを格納する期間を指定するためのオプションがあります。通常、テナントユーザが指定した保持手順は、グリッド管理者が指定した保持手順よりも優先されます。

### テナントユーザによるオブジェクト保持期間の制御方法

テナントユーザは、主に次の 3 つの方法でオブジェクトを StorageGRID に格納する期間を制御できます。

- グリッドでグローバルな S3 オブジェクトのロック設定が有効になっている場合、S3 テナントユーザは S3 オブジェクトのロックを有効にしたバケットを作成し、S3 REST API を使用して、そのバケットに追加された各オブジェクトバージョンの最新の保持設定とリーガルホールド設定を指定できます。
  - リーガルホールドの対象となっているオブジェクトバージョンは、どの方法でも削除できません。
  - オブジェクトバージョンのretain-until-dateに達する前は、どの方法でもそのバージョンを削除できません。
  - S3 オブジェクトロックが有効なバケット内のオブジェクトは ILM によって「無期限」に保持されます。ただし、それまでの保持期間が終了したあとは、クライアント要求やバケットライフサイクルの終了によってオブジェクトバージョンを削除できます。を参照してください ["S3 オブジェクトロックでオブジェクトを管理します"](#)。
- S3 テナントユーザは、Expiration アクションを指定するライフサイクル設定をバケットに追加できます。バケットライフサイクルが存在する場合、クライアントがオブジェクトを削除しないかぎり、StorageGRID は Expiration アクションで指定された日付または日数が経過するまでオブジェクトを格納します。を参照してください ["S3 ライフサイクル設定を作成する"](#)。
- S3 / Swift クライアントは、オブジェクトの削除要求を問題 に送信できます。StorageGRID は、オブジェクトを削除するか保持するかを決定する際に、常に S3 バケットライフサイクルまたは ILM よりもクライアントの削除要求を優先します。

### グリッド管理者によるオブジェクト保持期間の制御方法

グリッド管理者は、ILM の配置手順を使用してオブジェクトの格納期間を制御します。オブジェクトが ILM ルールに一致した場合、StorageGRID は ILM ルールの最後の期間が経過するまでそのオブジェクトを格納します。配置手順に「forever」が指定されている場合、オブジェクトは無期限に保持されます。

オブジェクトの保持期間を誰が制御するかに関係なく、格納するオブジェクトコピーのタイプ（レプリケートまたはイレイジャーコーディング）とコピーの場所（ストレージノード、クラウドストレージプール、またはアーカイブノード）は ILM 設定によって制御されます。

### S3 バケットライフサイクルと ILM の相互作用

S3 バケットライフサイクルの Expiration アクションは、常に ILM 設定よりも優先されます。その結果、ILM のオブジェクト配置手順がすべて終了したあとも、オブジェクトがグリッドに保持されることがあります。

## オブジェクト保持の例

S3 オブジェクトロック、バケットライフサイクル設定、クライアントの削除要求、ILM の相互作用について、より深く理解するために次の例を検討してください。

**例 1 : S3 バケットライフサイクルのオブジェクト保持期間が ILM よりも長い**

### ILM

2 つのコピーを 1 年間保存 ( 365 日 )

### バケットライフサイクル

2 年 ( 730 日 ) でオブジェクトが期限切れになる

### 結果

StorageGRID はオブジェクトを 730 日間格納します。StorageGRID は、バケットライフサイクル設定を使用して、オブジェクトを削除するか保持するかを決定します。



ILM よりもバケットライフサイクルのオブジェクト保持期間の方が長い場合でも、格納するコピーの数とタイプを決定する際には引き続き StorageGRID の配置手順が使用されます。この例では、366 日目から 730 日目までの間、オブジェクトの 2 つのコピーが StorageGRID に引き続き格納されます。

**例 2 : S3 バケットライフサイクルのオブジェクト保持期間よりも短い**

### ILM

2 つのコピーを 2 年間 ( 730 日 ) 格納する

### バケットライフサイクル

1 年 ( 365 日 ) でオブジェクトを期限切れにする

### 結果

StorageGRID は 365 日目にオブジェクトのコピーを両方削除します。

**例 3 : クライアントによる削除は、バケットライフサイクルと ILM よりも優先されます**

### ILM

2 つのコピーをストレージ・ノードに無期限に保存

### バケットライフサイクル

2 年 ( 730 日 ) でオブジェクトが期限切れになる

### クライアントの削除要求

発行日 : 400 日目

### 結果

StorageGRID は、クライアントの削除要求に応じて 400 日目にオブジェクトのコピーを両方削除します。

例 4 : S3 オブジェクトロックはクライアントの削除要求を上書きします

### S3 オブジェクトのロック

オブジェクトバージョンの retain-until は、2026-03-31 です。リーガルホールドは有効ではありません。

### 準拠 ILM ルール

2 つのコピーをストレージ・ノードに無期限に保存します

### クライアントの削除要求

2024-03-31 発行。

### 結果

retain-until はまだ 2 年前の時点であるため、StorageGRID はオブジェクトバージョンを削除しません。

## オブジェクトの削除方法

StorageGRID は、クライアント要求に直接応答してオブジェクトを削除するか、S3 バケットライフサイクルの終了または ILM ポリシーの要件に応じて自動的にオブジェクトを削除します。オブジェクトのさまざまな削除方法と StorageGRID による削除要求の処理方法を理解しておく、オブジェクトをより効率的に管理できるようになります。

StorageGRID では、次のいずれかの方法でオブジェクトを削除できます。

- 同期削除：StorageGRID がクライアントの削除要求を受け取ると、すべてのオブジェクトコピーがただちに削除されます。コピーが削除されると、削除が成功したことがクライアントに通知されます。
- オブジェクトは削除キューに登録されます。StorageGRID が削除要求を受け取ると、オブジェクトは削除キューに登録され、削除が成功したことがクライアントにすぐに通知されます。オブジェクトコピーは、あとでバックグラウンド ILM 処理によって削除されます。

StorageGRID では、オブジェクトを削除する際に、削除のパフォーマンスを最適化し、削除のバックログを最小限に抑え、スペースを最も早く解放する方法を使用します。

次の表は、StorageGRID がどのような場合に各メソッドを使用するかを

削除方法	使用時
<p>オブジェクトは削除キューに登録されます</p>	<p>次の条件のいずれか * が当てはまる場合：</p> <ul style="list-style-type: none"> <li>• 次のいずれかのイベントによってオブジェクトの自動削除がトリガーされた： <ul style="list-style-type: none"> <li>◦ S3 バケットのライフサイクル設定の有効期限または日数に達した。</li> <li>◦ ILM ルールに指定された最後の期間が経過した。</li> </ul> </li> </ul> <p>注： S3オブジェクトロックが有効になっているバケット内のオブジェクトは、リーガルホールドの対象である場合、またはretain-until-dateが指定されていてもまだ満たされていない場合は削除できません。</p> <ul style="list-style-type: none"> <li>• S3 / Swift クライアントが削除を要求し、次の条件を 1 つ以上満たしている： <ul style="list-style-type: none"> <li>◦ オブジェクトの場所が一時的に使用できない場合など、30秒以内にコピーを削除することはできません。</li> <li>◦ バックグラウンド削除キューがアイドル状態である。</li> </ul> </li> </ul>
<p>オブジェクトをただちに削除（同期削除）</p>	<p>S3 / Swift クライアントが削除要求を行い、次の * すべての条件が満たされている場合：</p> <ul style="list-style-type: none"> <li>• すべてのコピーを 30 秒以内に削除できる。</li> <li>• バックグラウンド削除キューには処理するオブジェクトが含まれていません。</li> </ul>

S3またはSwiftクライアントが削除要求を行うと、StorageGRID はまずオブジェクトを削除キューに追加します。その後、同期削除の実行に切り替えます。処理対象となるオブジェクトがバックグラウンド削除キューに含まれていることを確認することで、StorageGRID は、クライアントによる削除のバックログが発生しないようにしつつ、特に同時実行性の低いクライアントに対してより効率的に削除を処理できます。

## オブジェクトの削除に必要な時間

StorageGRID によるオブジェクトの削除方法は、システムの動作に影響を及ぼす可能性があります。

- StorageGRID StorageGRID で同期削除が実行されると、結果がクライアントに返されるまでに最大 30 秒かかることがあります。つまり、実際には StorageGRID がオブジェクトを削除キューに登録する場合よりも短時間でコピーが削除されるにもかかわらず、より長くかかっているという印象をクライアントに与える可能性があります。
- 一括削除の実行中にそのパフォーマンスを注意深く監視していると、一定数のオブジェクトが削除されたあとに削除の速度が遅くなったように見ることがあります。この変更は、StorageGRID がオブジェクトを削除キューへ登録する方法から同期削除に切り替えたときに発生します。削除速度が低下したように見えても、オブジェクトコピーの削除速度が遅くなったわけではありません。一方で、スペースの開放にかかる時間は、平均すると短くなっています。

大量のオブジェクトを削除する場合に、スペースを短時間で解放することが優先されるのであれば、ILM などの方法を使用してオブジェクトを削除するのではなく、クライアント要求を使用することを検討してください

い。一般に、クライアントによって削除が実行された場合、StorageGRID は同期削除を使用できるため、スペースはより短時間で解放されます。

オブジェクトの削除後にスペースを解放するために必要な時間は、いくつかの要因によって異なります。

- オブジェクトコピーが同期的に削除されるか、またはキューに登録されたあとで削除されるか（クライアントの削除要求の場合）。
- グリッド内のオブジェクトの数や、オブジェクトコピーが削除対象キューに登録される場合のグリッドリソースの可用性などのその他の要因（クライアントによる削除およびその他の方法の場合）。

### S3 バージョン管理オブジェクトの削除方法

S3 バケットでバージョン管理が有効になっている場合、StorageGRID は、削除要求に応答する際、要求が S3 クライアント、S3 バケットライフサイクルの終了、ILM ポリシーの要件のいずれによるものであるかにかかわらず、Amazon S3 の動作に従います。

オブジェクトがバージョン管理されている場合、オブジェクトの削除要求ではオブジェクトの現在のバージョンは削除されず、スペースも解放されません。代わりに、オブジェクトの削除要求では、オブジェクトの現在のバージョンとして削除マーカーが作成されます。これにより、オブジェクトの以前のバージョンが「noncurrent」になります。

オブジェクトが削除されていない場合でも、StorageGRID は現在のバージョンのオブジェクトが使用できなくなったかのように動作します。そのオブジェクトに対する要求は 404 NotFound を返します。ただし、最新でないオブジェクトデータは削除されていないため、最新でないバージョンのオブジェクトを指定する要求は成功します。

バージョン管理オブジェクトを削除するときにスペースを解放するには、次のいずれかを使用します。

- **\* S3クライアント要求\***：S3 DELETE Object要求にオブジェクトのバージョンIDを指定します (DELETE /object?versionId=ID)。この要求は、指定したバージョンのオブジェクトコピーだけを削除します (他のバージョンは引き続きスペースを消費します)。
- **バケットライフサイクル**：を使用します NoncurrentVersionExpiration をクリックします。NoncurrentDays で指定した日数に達すると、StorageGRID は最新でないオブジェクトバージョンのコピーをすべて完全に削除します。これらのオブジェクトバージョンはリカバリできません。
  - NewerNoncurrentVersions バケットライフサイクル設定の処理は、バージョン管理されたS3バケットで保持する最新でないバージョンの数を指定します。最新でないバージョンの数がより多い場合 NewerNoncurrentVersions NoncurrentDaysの値が経過すると、StorageGRID は古いバージョンを削除します。◦ NewerNoncurrentVersions しきい値は、ILMが提供するライフサイクルルール（内のバージョンが最新でないオブジェクト）よりも優先されます NewerNoncurrentVersions しきい値は、ILMが削除を要求した場合に保持されます。
- **\* ILM \***：**"アクティブポリシーのクローンを作成します"** 2つのILMルールを新しいドラフトポリシーに追加します。
  - 最初のルール：オブジェクトの最新でないバージョンを照合するには、参照時間として「noncurrent time」を使用します。インチ ["ILMルールの作成ウィザードの手順1 \(詳細を入力\)"](#)で、「Apply this rule to old object versions only (S3バケットでバージョン管理が有効になっている場合)？」という質問に対して**\* Yes \***を選択します。
  - 2つ目のルール：**\*取り込み時間\***を使用して現在のバージョンと一致させます。「noncurrent time」ルールは、ポリシーの**\*取り込み時間\***ルールの上に配置する必要があります。

### S3削除マーカの削除方法

バージョン管理オブジェクトが削除されると、StorageGRID はオブジェクトの現在のバージョンとして削除マーカを作成します。バケットからゼロバイトの削除マーカを削除するには、S3クライアントがオブジェクトのバージョンを明示的に削除する必要があります。ILM、バケットライフサイクルルール、バケット処理のDeleteオブジェクトで削除マーカが削除されることはありません。

#### 関連情報

- ["S3 REST APIを使用する"](#)
- ["例 4 : S3 バージョン管理オブジェクトの ILM ルールとポリシー"](#)

## 著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。