TR-4921 ：勒索軟體防禦

StorageGRID solutions and resources

NetApp
November 21, 2025

目錄

TR-4921 ：勒索軟體防禦

保護 StorageGRID S3 物件免受勒索軟體的侵害

瞭解勒索軟體攻擊、以及如何使用 StorageGRID 安全最佳實務來保護資料。

勒索軟體攻擊不斷增加。本文件針對如何保護 StorageGRID 上的物件資料提供一些建議。

現今的勒索軟體是資料中心永遠存在的危險。勒索軟體旨在加密資料、使依賴資料的使用者和應用程式無法使用。保護從強化網路和可靠使用者安全實務的一般防禦措施開始、我們必須遵循資料存取安全實務。

勒索軟體是當今最大的安全威脅之一。NetApp StorageGRID 團隊正與我們的客戶合作、以對抗這些威脅。使用物件鎖定和版本設定功能、您可以防範不必要的變更、並從惡意攻擊中恢復。資料安全性是一項多層風險、您的物件儲存設備只是資料中心的一部分。

StorageGRID 最佳實務做法

對於 StorageGRID 、安全性最佳實務做法應包括使用 HTTPS 搭配簽署的憑證、以進行管理和物件存取。為應用程式和個人建立專屬的使用者帳戶、請勿使用租戶根帳戶來存取應用程式或存取使用者資料。換句話說、請遵循最低權限原則。使用具有定義身分識別與存取管理（ IAM ）原則的安全性群組來管理使用者權限、以及存取應用程式和使用者專屬的帳戶。有了這些措施、您仍必須確保資料受到保護。在 Simple Storage Service （ S3 ）的情況下、當物件經過修改以加密時、則是透過覆寫原始物件來完成。

防禦方法

S3 API 的主要勒索軟體保護機制是實作物件鎖定。並非所有應用程式都與物件鎖定相容、因此本報告中說明的還有兩個其他選項可保護您的物件：複寫到另一個已啟用版本設定的儲存庫、以及使用 IAM 原則進行版本設定。

何處可找到其他資訊

若要深入瞭解本文所述資訊、請檢閱下列文件和 / 或網站：

- NetApp StorageGRID文檔中心https://docs.netapp.com/us-en/storagegrid/[]
- NetApp StorageGRID 啟用 https://docs.netapp.com/us-en/storagegrid-enable/
- NetApp 產品文件 https://www.netapp.com/support-and-training/documentation/

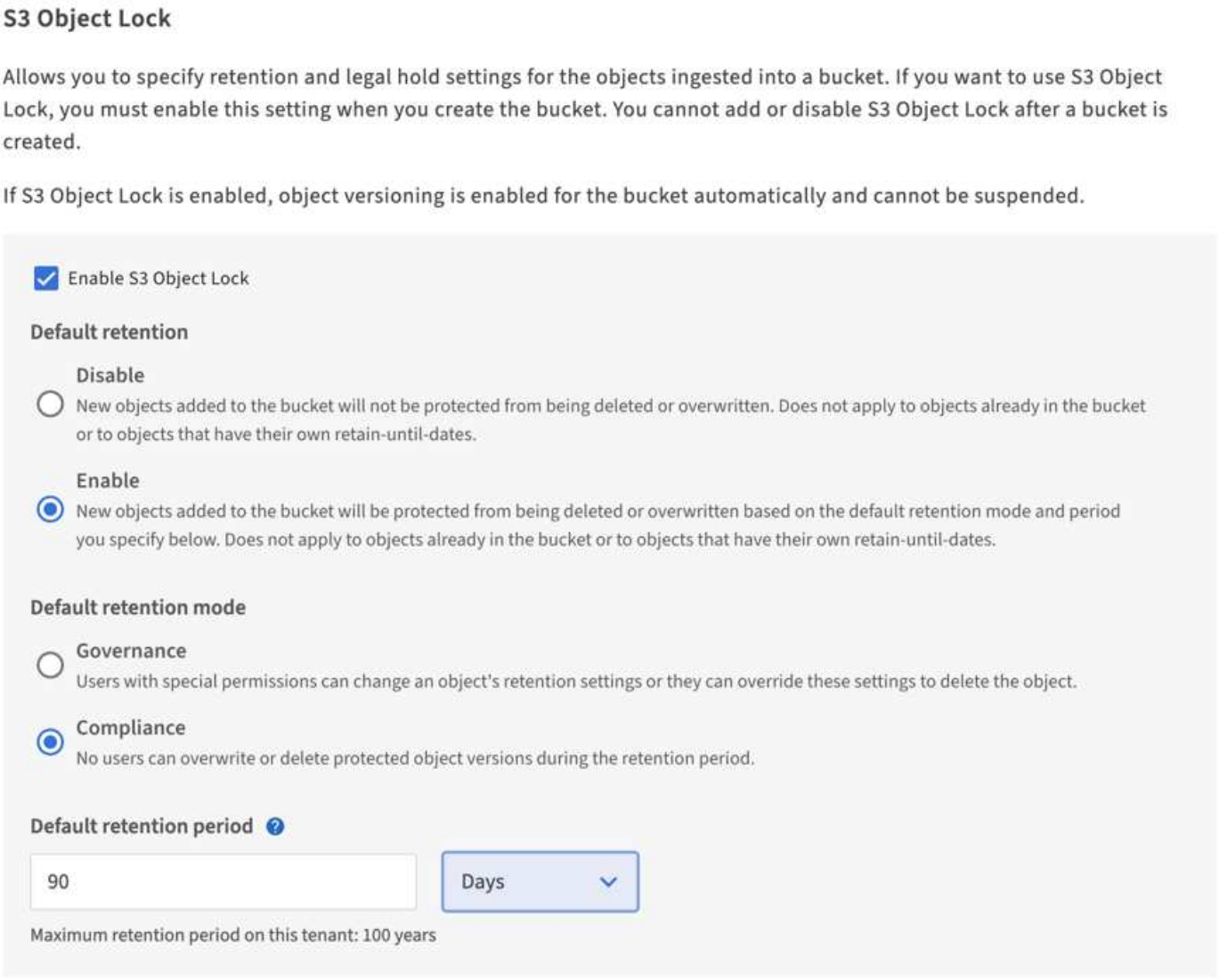
使用物件鎖定的勒索軟體防禦

探索 StorageGRID 中的物件鎖定如何提供 WORM 模式來防止資料刪除或覆寫、以及它如何符合法規要求。

物件鎖定提供 WORM 模式、可防止物件遭到刪除或覆寫。StorageGRID 實作物件鎖定 "Cohasset 已評估" 有助於符合法規要求、支援合法持有、符合性模式、以及物件保留的治理模式、以及預設貯體保留原則。您必須啟用物件鎖定、以做為貯體建立和版本設定的一部分。物件的特定版本已鎖定、如果未定義版本 ID 、則保留會置於物件的目前版本上。如果目前版本已設定保留、並嘗試刪除、修改或覆寫物件、則會以刪除標記或物件的新修訂版做為目前版本來建立新版本、 鎖定的版本會保留為非目前版本。對於尚未相容的應用程式、您仍可以使用物件鎖定和儲存在貯體上的預設保留組態。定義組態之後、這會將物件保留套用至置入貯體的每個新物件。只要將

應用程式設定為在保留時間過去之前不刪除或覆寫物件、就會生效。

在租用戶管理 UI 中建立儲存桶時，您可以啟用物件鎖定並配置預設保留模式和保留期限。配置後，這將為提取到該儲存桶的每個物件設定最小物件鎖定保留。



以下是使用物件鎖定 API 的幾個範例：

物件鎖定合法保留是套用至物件的簡單開 / 關狀態。

```
aws s3api put-object-legal-hold --bucket mybucket --key myfile.txt --legal
-hold Status=ON --endpoint-url https://s3.company.com
```

如果成功、設定合法保留狀態不會傳回任何值、因此可以使用「取得」操作來驗證。

```
aws s3api get-object-legal-hold --bucket mybucket --key myfile.txt
--endpoint-url https://s3.company.com
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

若要關閉合法保留、請套用關閉狀態。

```
aws s3api put-object-legal-hold --bucket mybucket --key myfile.txt --legal
-hold Status=OFF --endpoint-url https://s3.company.com
aws s3api get-object-legal-hold --bucket mybucket --key myfile.txt
--endpoint-url https://s3.company.com
{
    "LegalHold": {
        "Status": "OFF"
    }
}
```

設定物件保留會以保留到時間戳記完成。

```
aws s3api put-object-retention --bucket mybucket --key myfile.txt
--retention '{"Mode":"COMPLIANCE", "RetainUntilDate": "2022-06-
10T16:00:00"}'  --endpoint-url https://s3.company.com
```

同樣地、成功後沒有傳回的值、因此您可以使用 GET 通話來驗證保留狀態。

```
aws s3api get-object-retention --bucket mybucket --key myfile.txt
--endpoint-url https://s3.company.com
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2022-06-10T16:00:00+00:00"
    }
```

在啟用物件鎖定的貯體上放置預設保留、會使用以天和年為單位的保留期間。

```
aws s3api put-object-lock-configuration --bucket mybucket --object-lock
-configuration '{ "ObjectLockEnabled": "Enabled", "Rule": {
"DefaultRetention": { "Mode": "COMPLIANCE", "Days": 1 }}}' --endpoint-url
https://s3.company.com
```

如同大多數的作業一樣、成功時不會傳回任何回應、因此我們可以執行 Get 來驗證組態。

```
aws s3api get-object-lock-configuration --bucket mybucket --endpoint-url
https://s3.company.com
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 1
            }
        }
    }
}
```

接下來、您可以在套用保留組態的情況下、將物件放入貯體中。

```
aws s3 cp myfile.txt s3://mybucket --endpoint-url https://s3.company.com
```

Put 作業會傳回回應。

```
upload: ./myfile.txt to s3://mybucket/myfile.txt
```

在保留物件上、上一個範例中、貯體上設定的保留期間會轉換成物件上的保留時間戳記。

```
aws s3api get-object-retention --bucket mybucket --key myfile.txt
--endpoint-url https://s3.company.com
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2022-03-02T15:22:47.202000+00:00"
    }
}
```

使用複製的貯體搭配版本管理功能來防範勒索軟體

瞭解如何使用 StorageGRID CloudMirror 將物件複寫到次要儲存庫。

並非所有應用程式和工作負載都能與物件鎖相容。另一個選項是將物件複寫到同一個網格中的次要儲存格（最好是存取受限的不同租戶）、或是任何其他具有 StorageGRID 平台服務 CloudMirror 的 S3 端點。

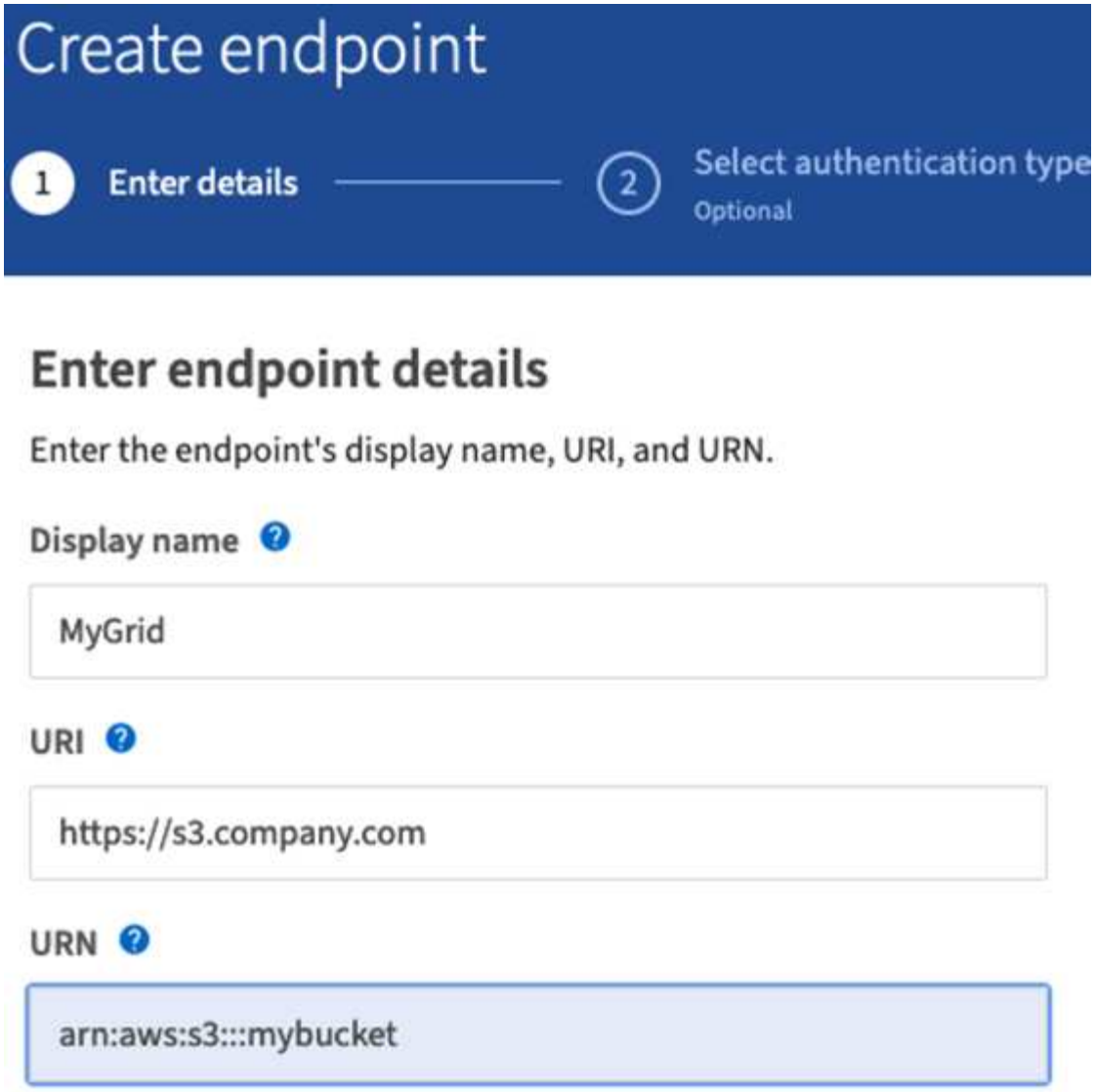
StorageGRID CloudMirror 是 StorageGRID 的一個元件、可設定為在物件擷取到來源儲存區時、將儲存區的物件複寫到定義的目的地、而不會複寫刪除內容。由於 CloudMirror 是 StorageGRID 的整合式元件、因此它無法被 S3 API 型攻擊關閉或操控。您可以在啟用版本設定的情況下、設定此複寫儲存區。在這種情況下、您需要自動清理複寫貯體的舊版本、以安全丟棄。因此、您可以使用 StorageGRID ILM 原則引擎。建立規則、根據非目前時間管理物件放置、時間足以識別攻擊並從攻擊中恢復。

這種方法的一個缺點是、它會使用完整的第二個儲存區複本、以及保留一段時間的多個物件版本、來消耗更多儲存空間。此外、有意從主要貯體中刪除的物件必須從複寫的貯體中手動移除。產品外部還有其他複寫選項、例如 NetApp CloudSync 、可針對類似解決方案複寫刪除內容。啟用版本設定功能而未啟用物件鎖定功能的次要貯體另一個缺點是存在許多可用來在次要位置造成損害的特殊權限帳戶。優點是、它應該是該端點或租戶貯體的唯一帳戶、而且可能會造成的影響不包括主要位置上的帳戶存取權、反之亦然。

建立來源和目的地儲存區、並使用版本設定目的地之後、您可以設定及啟用複寫、如下所示：

步驟

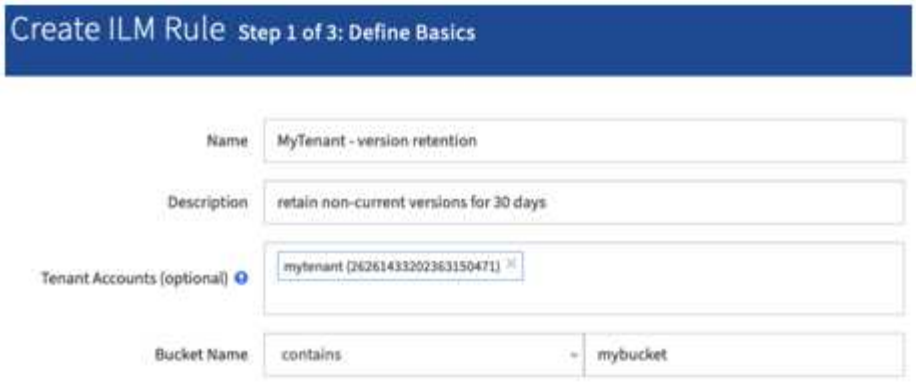
1. 若要設定 CloudMirror 、請為 S3 目的地建立平台服務端點。

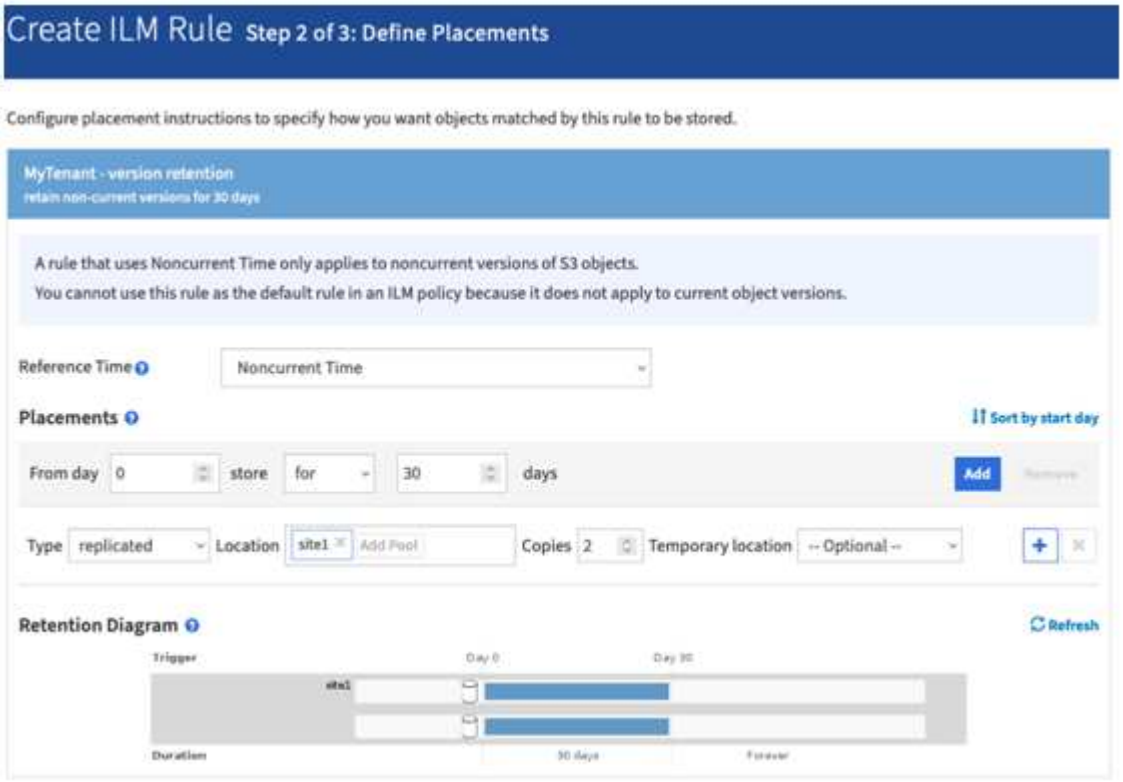


2. 在來源貯體上、將複寫設定為使用已設定的端點。

```
<ReplicationConfiguration>
    <Role></Role>
    <Rule>
        <Status>Enabled</Status>
        <Prefix></Prefix>
        <Destination>
            <Bucket>arn:aws:s3:::mybucket</Bucket>
            <StorageClass>STANDARD</StorageClass>
        </Destination>
    </Rule>
</ReplicationConfiguration>
```

3. 建立 ILM 規則以管理儲存配置和版本儲存持續時間管理。在此範例中、會設定要儲存之物件的非最新版本。





站台 1 有兩份 30 天的複本。您也可以根據在 ILM 規則中使用擷取時間做為參考時間、來設定物件目前版本的規則、以符合來源貯體儲存持續時間。物件版本的儲存位置可以進行銷毀編碼或複寫。

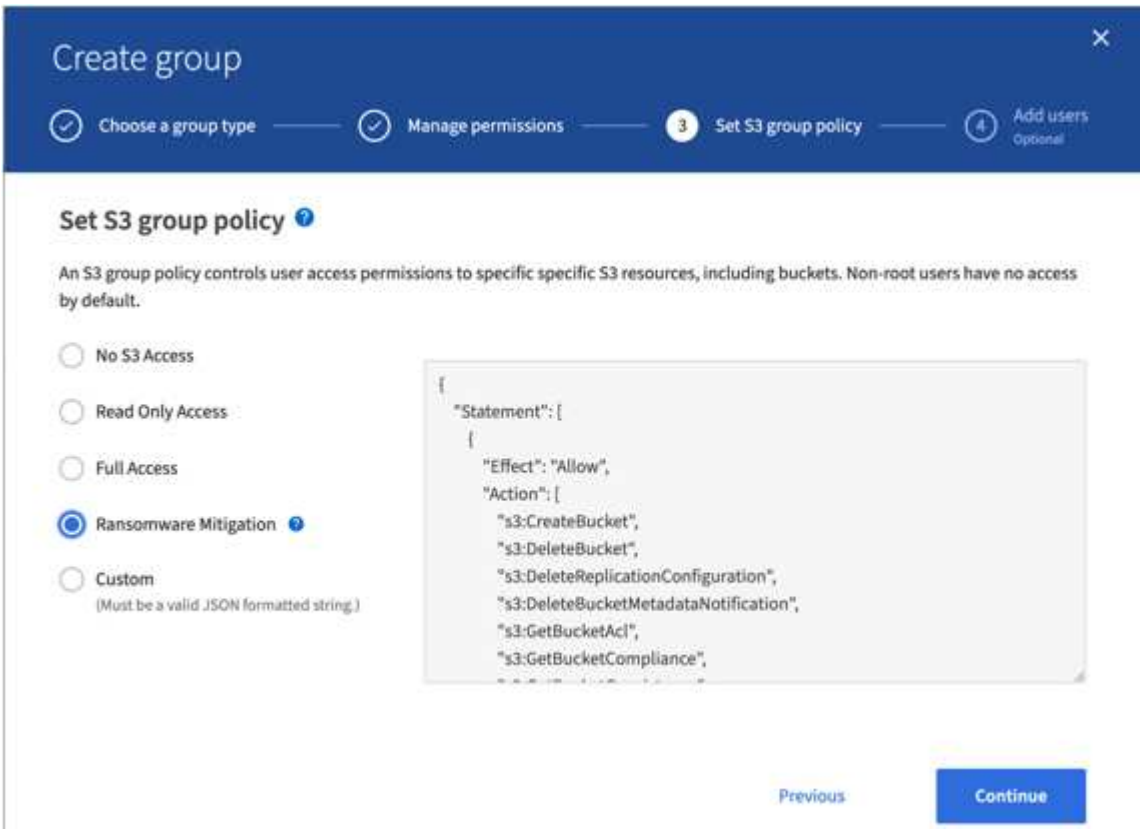
使用具有保護性 IAM 原則的版本管理功能來防範勒索軟體

瞭解如何啟用貯體的版本設定功能、並在 StorageGRID 中的使用者安全性群組上實作 IAM 原則、以保護您的資料。

在不使用物件鎖定或複寫的情況下保護資料的方法、是在貯體上啟用版本設定、並在使用者安全性群組上實作 IAM 原則、以限制使用者管理物件版本的能力。發生攻擊時、會建立新的不良資料版本作為目前版本、而最新的非最新版本則是安全無虞的資料。為了取得資料存取權而遭入侵的帳戶無法刪除或以其他方式變更保護資料的非目前版本、以供日後還原作業使用。如同先前的案例、 ILM 規則會在您選擇的期間內、管理非目前版本的保留。缺點是、仍然可能存在授權帳戶、導致不良攻擊者遭受攻擊、但所有應用程式服務帳戶和使用者都必須設定更具限制性的存取。限制性群組原則必須明確允許您希望使用者或應用程式能夠執行的每個動作、並明確拒絕您

不希望使用者或應用程式能夠執行的任何動作。NetApp 不建議使用萬用字元允許、因為未來可能會引進新的動作、而且您會想要控制是否允許或拒絕。對於此解決方案、拒絕清單必須包含 DeleteObjectVersion 、 PuttBucketPolicy 、 DeleteBucketPolicy 、 PuttLifecycleConfiguration 和 PuttBucketVersioning 、以保護貯體和物件版本的版本設定、避免使用者或程式設計上的變更。

在StorageGRID中，S3 群組原則選項「勒索軟體緩解」使得實施此解決方案變得更加容易。在租用戶中建立使用者群組時，選擇群組權限後，可以看到這個可選策略。



以下是群組原則的內容、其中包含明確允許的大部分可用作業、以及最低要求的拒絕。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "s3:DeleteReplicationConfiguration",
"s3:DeleteBucketMetadataNotification",
                "s3:GetBucketAcl",
                "s3:GetBucketCompliance",
                "s3:GetBucketConsistency",
                "s3:GetBucketLastAccessTime",
                "s3:GetBucketLocation",
                "s3:GetBucketNotification"
"s3:GetBucketObjectLockConfiguration",
```

```
                "s3:GetBucketPolicy",
                "s3:GetBucketMetadataNotification",
                "s3:GetReplicationConfiguration",
                "s3:GetBucketCORS",
                "s3:GetBucketVersioning",
                "s3:GetBucketTagging",
                "s3:GetEncryptionConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "s3:ListAllMyBuckets",
                "s3:ListBucketMultipartUploads",
                "s3:PutBucketConsistency",
                "s3:PutBucketLastAccessTime",
                "s3:PutBucketNotification",
"s3:PutBucketObjectLockConfiguration",
                "s3:PutReplicationConfiguration",
                "s3:PutBucketCORS",
                "s3:PutBucketMetadataNotification",
                "s3:PutBucketTagging",
                "s3:PutEncryptionConfiguration",
                "s3:AbortMultipartUpload",
                "s3:DeleteObject",
                "s3:DeleteObjectTagging",
                "s3:DeleteObjectVersionTagging",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectLegalHold",
                "s3:GetObjectRetention",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectLegalHold",
                "s3:PutObjectRetention",
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging",
                "s3:RestoreObject",
                "s3:ValidateObject",
                "s3:PutBucketCompliance",
                "s3:PutObjectVersionAcl"
            ],
            "Resource": "arn:aws:s3:::*"
```

```
        },
        {
            "Effect": "Deny",
            "Action": [
                "s3:DeleteObjectVersion",
                "s3:DeleteBucketPolicy",
                "s3:PutBucketPolicy",
                "s3:PutLifecycleConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

勒索軟體調查和補救

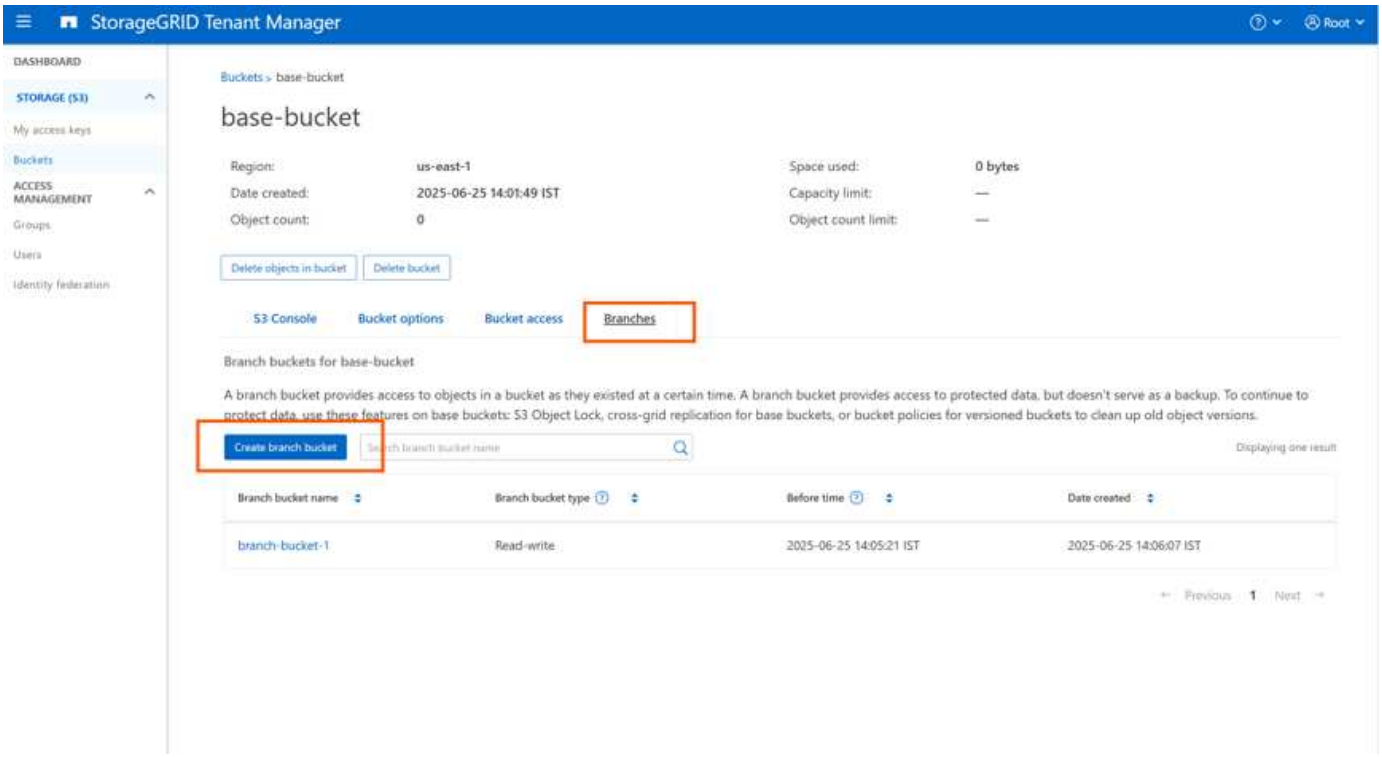
了解如何使用StorageGRID在可能發生勒索軟體攻擊後調查和修復儲存桶。

在StorageGRID 12.0 中，新增了新的分支儲存桶功能，以擴展版本控制對於勒索軟體防禦的實用性。分支儲存桶提供對儲存桶中物件的訪問，因為這些物件在某個時間存在，但前提是它們仍然存在於儲存桶中。只能為啟用版本控制的基本儲存桶建立分支儲存桶。

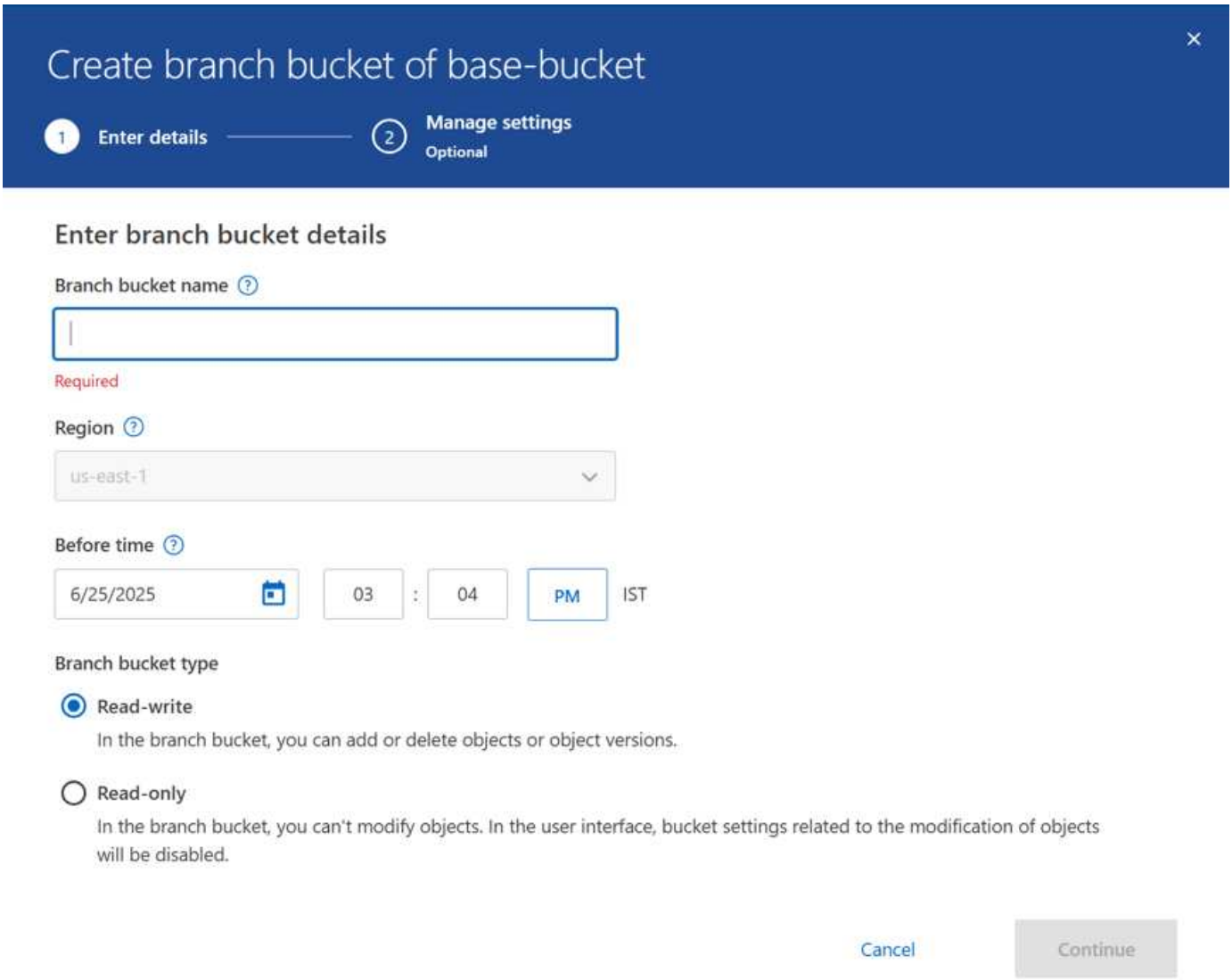
這意味著如果您懷疑發生了勒索軟體攻擊，您可以建立一個讀/寫或只讀分支儲存桶，其中包含初始攻擊之前存在的所有物件和版本。您可以使用此分支儲存桶與基本儲存桶內容進行比較，以確定哪些物件發生了變更以及該變更是否為攻擊的一部分。您也可以使用分支儲存桶，在調查攻擊的同時使用乾淨的分支繼續客戶端操作。

建立分支儲存桶

- 導航至基本儲存桶詳細資料頁面和分支標籤以建立分支儲存桶。



- 點擊「建立分支儲存桶」按鈕後，將開啟一個彈出窗口，其中預先填寫了與基本儲存桶關聯的區域的詳情。
- 在時間之前提供分支儲存桶名稱，並選擇要建立的分支儲存桶類型。


Create branch bucket of base-bucket
1 Enter details
2 Manage settings
Optional
Enter branch bucket details
Branch bucket name
Required
Region
us-east-1
Before time
6/25/2025
03
04
PM
IST
Branch bucket type
Read-write
In the branch bucket, you can add or delete objects or object versions.
Read-only
In the branch bucket, you can't modify objects. In the user interface, bucket settings related to the modification of objects will be disabled.
Cancel
Continue