TR-4921: Defensa contra ransomware

How to enable StorageGRID in your environment

NetApp
July 05, 2024

Tabla de contenidos

TR-4921: Defensa contra ransomware

Protege objetos de StorageGRID S3 contra el ransomware

Obtén más información sobre los ataques de ransomware y cómo proteger los datos con las prácticas recomendadas de seguridad de StorageGRID.

Los ataques de ransomware están en auge. Este documento proporciona algunas recomendaciones sobre cómo proteger los datos de objetos en StorageGRID.

Hoy en día, el ransomware es el peligro siempre presente en los centros de datos. El ransomware está diseñado para cifrar datos y dejarlos inutilizables por los usuarios y las aplicaciones que dependen de ellos. La protección comienza con las defensas habituales de las redes reforzadas y las prácticas de seguridad de usuario sólidas, y tenemos que seguir con las prácticas de seguridad de acceso a los datos.

El ransomware es una de las mayores amenazas de seguridad de hoy en día. El equipo de NetApp StorageGRID está trabajando con nuestros clientes para mantenerse a la cabeza de estas amenazas. Con el uso de bloqueo de objetos y control de versiones, puede protegerse frente a alteraciones no deseadas y recuperarse de ataques maliciosos. La seguridad de datos es una aventura de múltiples capas, donde tu almacenamiento de objetos es solo una parte del centro de datos.

Mejores prácticas de StorageGRID

Para StorageGRID, las prácticas recomendadas de seguridad deben incluir el uso de HTTPS con certificados firmados tanto para la gestión como para el acceso a los objetos. Cree cuentas de usuario dedicadas para aplicaciones e individuos, y no utilice las cuentas raíz de inquilino para el acceso a aplicaciones o el acceso a los datos de usuario. En otras palabras, siga el principio de privilegio mínimo. Utilice grupos de seguridad con directivas de gestión de acceso e identidad definidas (IAM) para controlar los derechos de usuario y acceder a cuentas específicas de las aplicaciones y los usuarios. Con estas medidas, aún debe asegurarse de que los datos estén protegidos. En el caso de Simple Storage Service (S3), cuando se modifican objetos para cifrarlos, se realiza mediante una sobrescritura del objeto original.

Métodos de defensa

El principal mecanismo de protección contra ransomware en la API S3 es implementar el bloqueo de objetos. No todas las aplicaciones son compatibles con el bloqueo de objetos, por lo que hay otras dos opciones para proteger los objetos que se describen en este informe: Replicación a otro depósito con control de versiones activado y control de versiones con políticas de IAM.

Dónde encontrar información adicional

Si quiere más información sobre el contenido de este documento, consulte los siguientes documentos o sitios web:

- Centro de documentación de NetApp StorageGRID https://docs.netapp.com/us-en/storagegrid-118/
- Habilitación para NetApp StorageGRID https://docs.netapp.com/us-en/storagegrid-enable/
- Página de recursos de documentación de StorageGRID https://www.netapp.com/data-storage/storagegrid/documentation/
- Documentación de producto de NetApp https://www.netapp.com/support-and-training/documentation/

Defensa contra ransomware mediante bloqueo de objetos

Explore cómo el bloqueo de objetos en StorageGRID ofrece un MODELO WORM para evitar la eliminación o sobrescritura de datos y cómo cumple con los requisitos normativos.

El bloqueo de objetos proporciona un MODELO WORM para evitar que se eliminen o se sobrescriban objetos. La implementación de StorageGRID del bloqueo de objetos es "Cohasset evaluado" para ayudar a cumplir los requisitos normativos, dar soporte a la conservación legal, el modo de cumplimiento de normativas y el modo de gobierno para la retención de objetos y las políticas de retención de bloques predeterminadas. Debe habilitar el bloqueo del objeto como parte de la creación y el control de versiones del bloque. Se bloquea una versión específica de un objeto y, si no se define ningún ID de versión, la retención se coloca en la versión actual del objeto. Si la versión actual tiene la retención configurada y se intenta suprimir, modificar o sobrescribir el objeto, se crea una nueva versión con un marcador de supresión o la nueva revisión del objeto como la versión actual, y la versión bloqueada se conserva como una versión no actual. Para las aplicaciones que aún no son compatibles, es posible que pueda seguir usando el bloqueo de objetos y una configuración de retención predeterminada ubicada en el bloque. Una vez definida la configuración, esto aplica una retención de objetos a cada nuevo objeto puesto en el bloque. Esto funciona siempre que la aplicación esté configurada para no eliminar ni sobrescribir los objetos antes de que haya pasado el tiempo de retención.

A continuación, se muestran algunos ejemplos que utilizan la API de bloqueo de objetos:

La retención legal de bloqueo de objeto es un estado simple de activación/desactivación aplicado a un objeto.

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

Si se establece el estado de retención legal, no se devuelve ningún valor si se realiza correctamente, por lo que se puede verificar con una OPERACIÓN GET.

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

Para desactivar la retención legal, aplique el estado OFF.

```
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"
    }
}
```

La configuración de la retención de objetos se realiza con una marca de tiempo Retain until.

```
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
```

Una vez más, no hay valor devuelto en el éxito, por lo que puede verificar el estado de retención de manera similar con una llamada GET.

```
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"
    }
```

Al colocar una retención predeterminada en un bloque habilitado para el bloqueo de objetos, se utiliza un período de retención en días y años.

```
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
```

Al igual que con la mayoría de estas operaciones, no se devuelve ninguna respuesta al éxito, por lo que podemos realizar un GET para que la configuración se verifique.

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

A continuación, puede colocar un objeto en el depósito con la configuración de retención aplicada.

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

La operación PUT devuelve una respuesta.

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

En el objeto de retención, la duración de retención definida en el bloque en el ejemplo anterior se convierte en una marca de tiempo de retención en el objeto.

```
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"
    }
}
```

Protección contra ransomware mediante bloque replicado con control de versiones

Descubre cómo replicar objetos en un bloque secundario mediante CloudMirror de StorageGRID.

No todas las aplicaciones y cargas de trabajo serán compatibles con el bloqueo de objetos. Otra opción es replicar los objetos en un depósito secundario, ya sea en la misma cuadrícula (preferiblemente un inquilino

diferente con acceso restringido), o cualquier otro extremo S3 con el servicio de plataforma StorageGRID, CloudMirror.

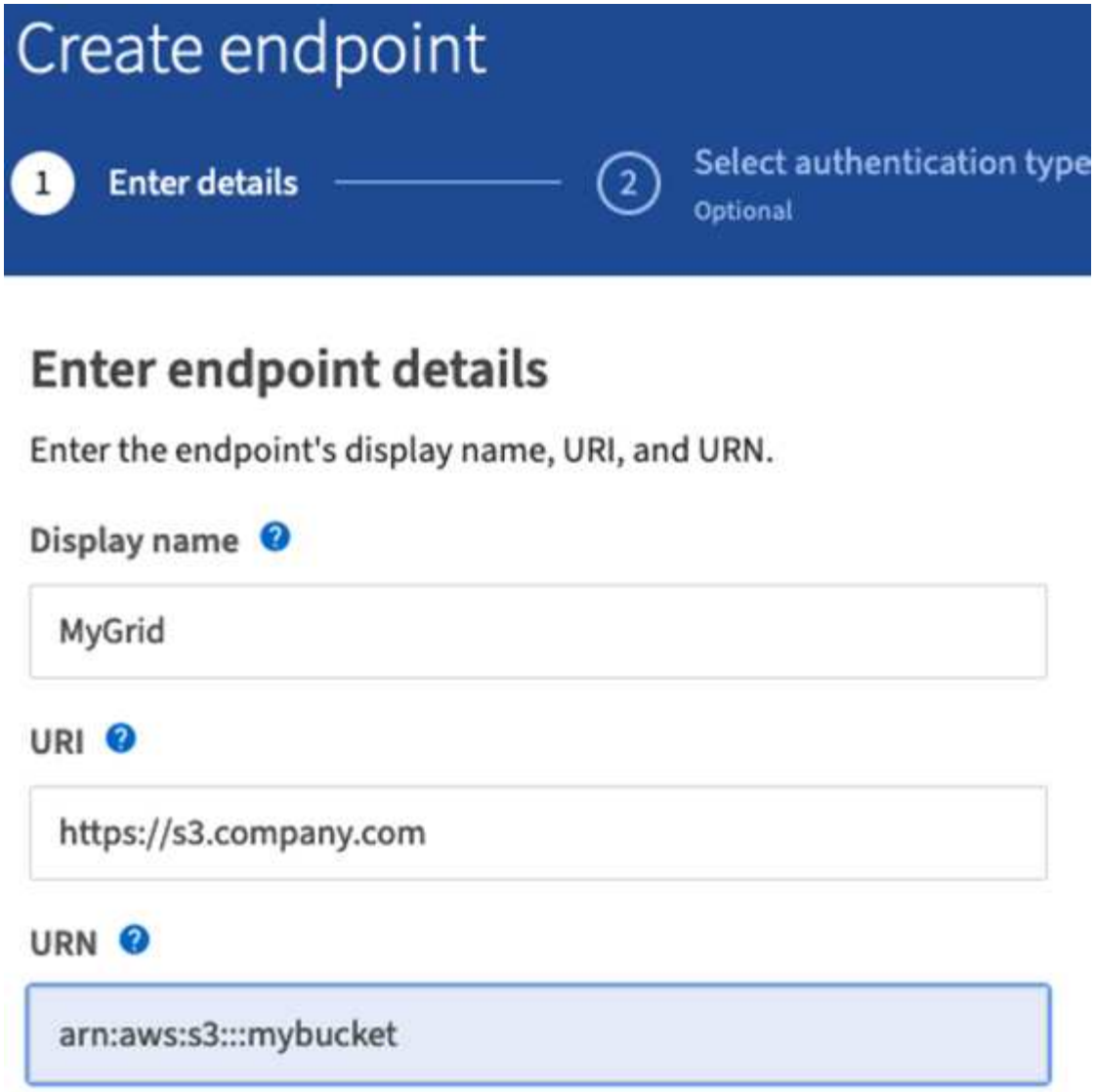
StorageGRID CloudMirror es un componente de StorageGRID que se puede configurar para replicar los objetos de un bucket en un destino definido a medida que se ingieren en el bloque de origen y no replica los eliminaciones. Puesto que CloudMirror es un componente integrado de StorageGRID, no se puede desactivar ni manipular mediante un ataque basado en API S3. Puede configurar este bucket replicado con el control de versiones activado. En este escenario, necesita una limpieza automática de las versiones antiguas del depósito replicado que son seguras de descartar. Para ello, puede utilizar el motor de políticas de gestión de la vida útil de la información de StorageGRID. Cree reglas para administrar la ubicación del objeto en función del tiempo no corriente durante varios días suficientes para identificarse y recuperarse de un ataque.

Un inconveniente de este enfoque es que consume más almacenamiento al disponer de una segunda copia completa del bloque y varias versiones de los objetos que se conservan durante cierto tiempo. Además, los objetos que se eliminaron intencionalmente del bloque primario deben eliminarse manualmente del bloque replicado. Hay otras opciones de replicación fuera del producto, como NetApp CloudSync, que pueden replicar eliminaciones para una solución similar. Otra desventaja para el bucket secundario que está activado el control de versiones y no el bloqueo de objetos activado es que existe una serie de cuentas con privilegios que se pueden utilizar para causar daños en la ubicación secundaria. La ventaja es que debe ser una cuenta única para ese extremo o bloque de inquilinos y es probable que el compromiso no incluya el acceso a las cuentas en la ubicación principal o viceversa.

Después de crear los buckets de origen y de destino y de configurar el destino con el control de versiones, puede configurar y habilitar la replicación del siguiente modo:

Pasos

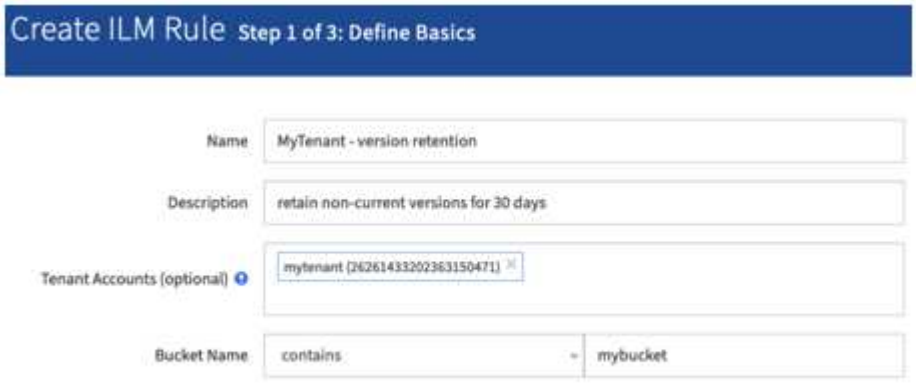
1. Para configurar CloudMirror, cree un punto final de servicios de plataforma para el destino S3.

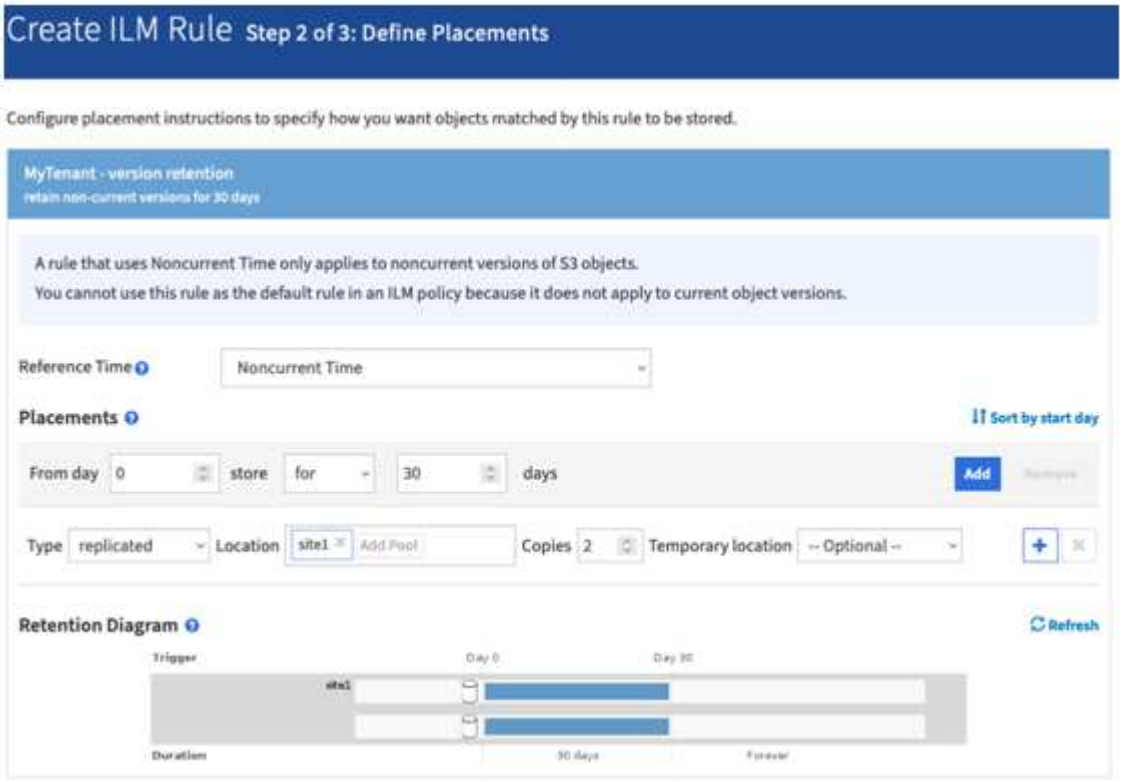


2. En el bloque de origen, configure la replicación para usar el punto final configurado.

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

3. Cree reglas de ILM para gestionar la ubicación del almacenamiento y la gestión de la duración del almacenamiento de versiones. En este ejemplo, se configuran las versiones no actuales de los objetos que se van a almacenar.





Hay dos copias en el sitio 1 durante 30 días. También configura las reglas para la versión actual de los objetos en función de usar el tiempo de ingesta como tiempo de referencia en la regla de ILM para que coincida con la duración del almacenamiento del bloque de origen. La ubicación del almacenamiento para las versiones de objetos puede codificarse para el borrado o replicarse.

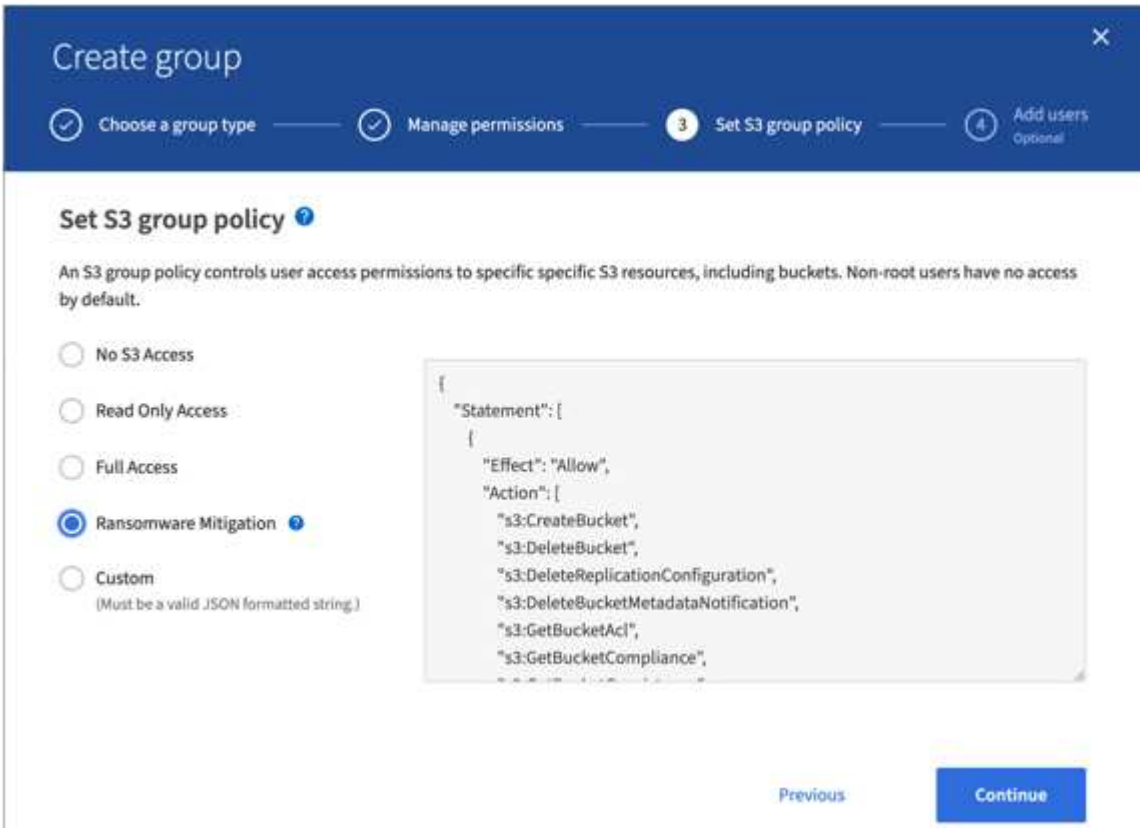
Protección frente al ransomware mediante el control de versiones con la política de protección IAM

Aprenda a proteger sus datos habilitando el control de versiones en el bloque e implementando políticas de IAM en grupos de seguridad de usuarios en StorageGRID.

Un método para proteger los datos sin utilizar el bloqueo de objetos o la replicación consiste en habilitar el control de versiones en el bloque e implementar políticas de IAM en los grupos de seguridad de usuarios para limitar la capacidad de los usuarios de administrar versiones de los objetos. En caso de un ataque, se crean

nuevas versiones erróneas de los datos como la versión actual, y la versión no actual más reciente son los datos limpios seguros. Las cuentas comprometidas para obtener acceso a los datos no tienen acceso para eliminar o alterar de otro modo la versión no actual que los protege para operaciones de restauración posteriores. Al igual que en la situación anterior, las reglas de ILM gestionan la retención de las versiones no actualizadas con el tiempo que elija. El inconveniente es que todavía existe la posibilidad de que existan cuentas privilegiadas para un ataque de agente malicioso, pero todas las cuentas de servicio de aplicaciones y los usuarios deben configurarse con un acceso más restrictivo. La política de grupo restrictivo debe permitir explícitamente que cada acción que desee que los usuarios o la aplicación sean capaces de rechazar y de forma explícita cualquier acción que no desee que sean capaces de realizar. NetApp no recomienda utilizar un comodín Permitir porque se puede introducir una nueva acción en el futuro y se quiere controlar si se permite o se rechaza. Para esta solución, la lista de denegación debe incluir DeleteObjectVersion, PutBucketPolicy, DeleteBucketPolicy, PutLifecycleConfiguration y PutBucketVersioning para proteger la configuración de versiones del bucket y las versiones del objeto de los cambios programáticos o del usuario.

En StorageGRID 11,7 se ha introducido una nueva opción de política de grupo de S3 «Mitigación de ransomware» para facilitar la implementación de esta solución. Al crear un grupo de usuarios en el inquilino, después de seleccionar los permisos del grupo, puede ver esta nueva política opcional.



A continuación se muestra el contenido de la política de grupo que incluye la mayoría de las operaciones disponibles explícitamente permitidas y el mínimo requerido denegado.

```
{
    "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:::*"
        }
    ]
}
```