環境でStorageGRID を有効にする方法

How to enable StorageGRID in your environment

NetApp
April 26, 2024

目次

環境でStorageGRID を有効にする方法

検証済みのサードパーティソリューション

検証済みのサードパーティソリューション：概要

ネットアップはパートナー様と協力して、これらのソリューションをStorageGRID で使用できるように検証しました。このセクションの情報を参照して、検証済みのソリューションを確認し、必要に応じて追加の手順を入手してください。

ネットアップの業界最高水準のテスト済みソリューションを構築すると、力を合わせてネットアップのポートフォリオを強化し、市場認知度を高め、売上を拡大できます。 "今すぐアライアンスパートナーになりましょう"。

StorageGRID 11.8検証済みのサードパーティソリューション

以下のサードパーティソリューションは、StorageGRID 11.8での使用が検証済みです。[+]
お探しの解決策が表示されない場合は、NetAppのアカウント担当者にお問い合わせください。

StorageGRID で検証済みのサードパーティソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- Actifio
- アルクシオ
- Apache Kafka です
- AWSマウントポイント
- ブリッジストール
- カンテモ
- Citrixコンテンツコラボレーション
- Collibra（Collibra Data Qualityの最小バージョン2024.02）
- Commvault 11.
- CTERAポータル6.
- ダレト
- ダタドビ
- Data Dynamics StorageXのように入力します
- DefendX
- Diskoverデータ
- デレミオ
- eMAM

- FUJIFILMオブジェクトアーカイブ
- GitHubエンタープライズサーバ
- IBM FileNetの順にクリックします
- IBM Spectrum Protect Plusのサポート
- Interica
- Komprise
- Microsoft SQL Server Big Data Clustersの略
- モデル9.
- Modzy
- Moonwalk Universalの略
- 良いですね
- Nasuni
- OpenText Documentum 16.4
- OpenText Documentum 21.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloudを使用したOpenText Media Management 16.5
- Panzura
- PixitMedia ngenea.
- Point Archival Gateway 2.0の場合
- Point Storage Manager 6.4
- プリミティブストリーム
- Quantum StorNext 5.4.0.1
- Reveille v10 build 220706以上
- Rubrik CDMの略
- s3a
- シグニエント
- 雪の結晶
- Spectra Logic On-Prem Glacier
- Splunk Smartstore
- ストレージが簡単になりました
- トリノ
- ニスエンタープライズ6.0.4
- Veeam 12
- Veritas Enterprise Vault 14の略
- Veritas NetBackup 8.0

- Vertica 10.x
- ビディズパイン
- Virtalica StorageFabricの詳細を参照してください
- Weka v3.10以降

StorageGRID で検証済みの、オブジェクトロック機能を備えたサードパーティ製ソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- CommVault 11 Feature Release 26
- IBM FileNetの順にクリックします
- OpenText Documentum 21.4
- Veeam 12
- Veritas Enterprise Vault 14.2.2
- Veritas NetBackup 10.1.1以降

StorageGRIDでサポートされているサードパーティソリューション

これらのソリューションはテスト済みです。

- アーチウェア
- アクシスコミュニケーションズ
- コングルーシティ360
- DataFrameworksの略
- EcoDigital DIVAプラットフォーム
- Encoding.com
- FUJIFILMオブジェクトアーカイブ
- GE Centricity Enterprise Archiveの略
- Gitlab
- ハイランド・アクオ
- IBM Aspera
- マイルストーンシステム
- ONSSI
- REACHエンジン
- SilverTrak
- SoftNAS
- QSTAR
- ベラシア

StorageGRID 11.7で検証済みのサードパーティソリューション

以下のサードパーティソリューションは、StorageGRID 11.7での使用が検証済みです。+お探しの解決策 が表示されない場合は、ネットアップの担当者までお問い合わせください。

StorageGRID で検証済みのサードパーティソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- Actifio
- アルクシオ
- Apache Kafka です
- AWSマウントポイント
- ブリッジストール
- カンテモ
- Citrixコンテンツコラボレーション
- Collibra（Collibra Data Qualityの最小バージョン2024.02）
- Commvault 11.
- CTERAポータル6.
- ダレト
- ダタドビ
- Data Dynamics StorageXのように入力します
- DefendX
- Diskoverデータ
- デレミオ
- eMAM
- FUJIFILMオブジェクトアーカイブ
- GitHubエンタープライズサーバ
- IBM FileNetの順にクリックします
- IBM Spectrum Protect Plusのサポート
- Interica
- Komprise
- Microsoft SQL Server Big Data Clustersの略
- モデル9.
- Modzy
- Moonwalk Universalの略

- 良いですね
- Nasuni
- OpenText Documentum 16.4
- OpenText Documentum 21.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloudを使用したOpenText Media Management 16.5
- Panzura
- PixitMedia ngenea.
- Point Archival Gateway 2.0の場合
- Point Storage Manager 6.4
- プリミティブストリーム
- Quantum StorNext 5.4.0.1
- Reveille v10 build 220706以上
- Rubrik CDMの略
- s3a
- シグニエント
- 雪の結晶
- Spectra Logic On-Prem Glacier
- Splunk Smartstore
- ストレージが簡単になりました
- トリノ
- ニスエンタープライズ6.0.4
- Veeam 12
- Veritas Enterprise Vault 14の略
- Veritas NetBackup 8.0
- Vertica 10.x
- ビディズパイン
- Virtalica StorageFabricの詳細を参照してください
- Weka v3.10以降

StorageGRID で検証済みの、オブジェクトロック機能を備えたサードパーティ製ソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- CommVault 11 Feature Release 26
- IBM FileNetの順にクリックします
- OpenText Documentum 21.4

- Veeam 12
- Veritas Enterprise Vault 14.2.2
- Veritas NetBackup 10.1.1以降

StorageGRIDでサポートされているサードパーティソリューション

これらのソリューションはテスト済みです。

- アーチウェア
- アクシスコミュニケーションズ
- コングルーシティ360
- DataFrameworksの略
- EcoDigital DIVAプラットフォーム
- Encoding.com
- FUJIFILMオブジェクトアーカイブ
- GE Centricity Enterprise Archiveの略
- Gitlab
- ハイランド・アクオ
- IBM Aspera
- マイルストーンシステム
- ONSSI
- REACHエンジン
- SilverTrak
- SoftNAS
- QSTAR
- ベラシア

StorageGRID 11.6検証済みのサードパーティソリューション

StorageGRID 11.6では、以下のサードパーティソリューションの使用が検証されています。+お探しの解決策 が表示されない場合は、ネットアップの担当者までお問い合わせください。

StorageGRID で検証済みのサードパーティソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- Actifio
- アルクシオ
- Apache Kafka です

- ブリッジストール
- カンテモ
- Citrixコンテンツコラボレーション
- Commvault 11.
- CTERAポータル6.
- ダレト
- ダタドビ
- Data Dynamics StorageXのように入力します
- DefendX
- Diskoverデータ
- デレミオ
- eMAM
- FUJIFILMオブジェクトアーカイブ
- GitHubエンタープライズサーバ
- IBM FileNetの順にクリックします
- IBM Spectrum Protect Plusのサポート
- Interica
- Komprise
- Microsoft SQL Server Big Data Clustersの略
- モデル9.
- Modzy
- Moonwalk Universalの略
- 良いですね
- Nasuni
- OpenText Documentum 16.4
- OpenText Documentum 21.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloudを使用したOpenText Media Management 16.5
- Panzura
- PixitMedia ngenea.
- Point Archival Gateway 2.0の場合
- Point Storage Manager 6.4
- プリミティブストリーム
- Quantum StorNext 5.4.0.1
- Reveille v10 build 220706以上

- Rubrik CDMの略
- s3a
- シグニエント
- 雪の結晶
- Spectra Logic On-Prem Glacier
- Splunk Smartstore
- ストレージが簡単になりました
- トリノ
- ニスエンタープライズ6.0.4
- Veeam 12
- Veritas Enterprise Vault 14の略
- Veritas NetBackup 8.0
- Vertica 10.x
- ビディズパイン
- Virtalica StorageFabricの詳細を参照してください
- Weka v3.10以降

StorageGRID で検証済みの、オブジェクトロック機能を備えたサードパーティ製ソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- CommVault 11 Feature Release 26
- IBM FileNetの順にクリックします
- OpenText Documentum 21.4
- Veeam 12
- Veritas Enterprise Vault 14.2.2
- Veritas NetBackup 10.1.1以降

StorageGRIDでサポートされているサードパーティソリューション

これらのソリューションはテスト済みです。

- アーチウェア
- アクシスコミュニケーションズ
- コングルーシティ360
- DataFrameworksの略
- EcoDigital DIVAプラットフォーム
- Encoding.com

- FUJIFILMオブジェクトアーカイブ
- GE Centricity Enterprise Archiveの略
- Gitlab
- ハイランド・アクオ
- IBM Aspera
- マイルストーンシステム
- ONSSI
- REACHエンジン
- SilverTrak
- SoftNAS
- QSTAR
- ベラシア

StorageGRID 11.5で検証済みのサードパーティソリューション

次の他社製ソリューションは、StorageGRID 11.5で使用することが検証されています。+お探しの解決策 が表示されない場合は、ネットアップの担当者までお問い合わせください。

StorageGRID で検証済みのサードパーティソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- Actifio
- アルクシオ
- ブリッジストール
- カンテモ
- Citrixコンテンツコラボレーション
- Commvault 11.
- CTERAポータル6.
- ダレト
- ダタドビ
- Data Dynamics StorageXのように入力します
- DefendX
- Interica
- Komprise
- Moonwalk Universalの略
- 良いですね

- Nasuni
- OpenText Documentum 16.4
- OpenText Documentum 21.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloudを使用したOpenText Media Management 16.5
- Panzura
- Point Archival Gateway 2.0の場合
- Point Storage Manager 6.4
- プリミティブストリーム
- Quantum StorNext 5.4.0.1
- Rubrik CDMの略
- s3a
- シグニエント
- Splunk Smartstore
- トリノ
- ニスエンタープライズ6.0.4
- Veeam 11の統合によって
- Veritas Enterprise Vault 11の略
- Veritas Enterprise Vault 12.
- Veritas NetBackup 8.0
- Vertica 10.x
- ビディズパイン
- Virtalica StorageFabricの詳細を参照してください

StorageGRID で検証済みの、オブジェクトロック機能を備えたサードパーティ製ソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- OpenText Documentum 21.4
- Veeam 11の統合によって

StorageGRIDでサポートされているサードパーティソリューション

これらのソリューションはテスト済みです。

- アーチウェア
- アクシスコミュニケーションズ
- コングルーシティ360

- DataFrameworksの略
- EcoDigital DIVAプラットフォーム
- Encoding.com
- FUJIFILMオブジェクトアーカイブ
- GE Centricity Enterprise Archiveの略
- Gitlab
- ハイランド・アクオ
- IBM Aspera
- マイルストーンシステム
- ONSSI
- REACHエンジン
- SilverTrak
- SoftNAS
- QSTAR
- ベラシア

StorageGRID 11.4検証済みのサードパーティソリューション

次のサードパーティソリューションは、StorageGRID 11.4で使用することが検証されています。+お探しの解決策 が表示されない場合は、ネットアップの担当者までお問い合わせください。

StorageGRID で検証済みのサードパーティソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- Actifio
- ブリッジストール
- カンテモ
- Citrixコンテンツコラボレーション
- Commvault 11.
- CTERAポータル6.
- ダレト
- ダタドビ
- Data Dynamics StorageXのように入力します
- DefendX
- Interica
- Komprise

- 良いですね
- Nasuni
- OpenText Documentum 16.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloudを使用したOpenText Media Management 16.5
- Panzura
- Point Archival Gateway 2.0の場合
- Point Storage Manager 6.4
- プリミティブストリーム
- Quantum StorNext 5.4.0.1
- Rubrik CDMの略
- シグニエント
- Splunk Smartstore
- ニスエンタープライズ6.0.4
- Veeam 9.5.4
- Veritas Enterprise Vault 11の略
- Veritas Enterprise Vault 12.
- Veritas NetBackup 8.0
- Vertica 10.x
- ビディズパイン

StorageGRIDでサポートされているサードパーティソリューション

これらのソリューションはテスト済みです。

- アーチウェア
- アクシスコミュニケーションズ
- コングルーシティ360
- DataFrameworksの略
- EcoDigital DIVAプラットフォーム
- Encoding.com
- FUJIFILMオブジェクトアーカイブ
- GE Centricity Enterprise Archiveの略
- ハイランド・アクオ
- IBM Aspera
- マイルストーンシステム
- ONSSI

- REACHエンジン
- SilverTrak
- SoftNAS
- QSTAR
- ベラシア

StorageGRID 11.3は、検証済みのサードパーティソリューションです

StorageGRID 11.3では、次のサードパーティソリューションが検証されています。+お探しの解決策 が表示されない場合は、ネットアップの担当者までお問い合わせください。

StorageGRID で検証済みのサードパーティソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- Actifio
- ブリッジストール
- カンテモ
- Citrixコンテンツコラボレーション
- Commvault 11.
- CTERAポータル6.
- ダレト
- ダタドビ
- Data Dynamics StorageXのように入力します
- DefendX
- Interica
- Komprise
- 良いですね
- Nasuni
- OpenText Documentum 16.4
- CyanGate Cloudを使用したOpenText Media Management 16.5
- Panzura
- Point Archival Gateway 2.0の場合
- Point Storage Manager 6.4
- プリミティブストリーム
- Quantum StorNext 5.4.0.1

- RUBRIK CDM 5.0.1 p1-1342
- シグニエント
- Splunk Smartstore
- ニスエンタープライズ6.0.4
- Veeam 9.5.4
- Veritas Enterprise Vault 11の略
- Veritas Enterprise Vault 12.
- Veritas NetBackup 8.0
- ビディズパイン

StorageGRIDでサポートされているサードパーティソリューション

これらのソリューションはテスト済みです。

- アーチウェア
- アクシスコミュニケーションズ
- コングルーシティ360
- DataFrameworksの略
- EcoDigital DIVAプラットフォーム
- Encoding.com
- FUJIFILMオブジェクトアーカイブ
- GE Centricity Enterprise Archiveの略
- ハイランド・アクオ
- IBM Aspera
- マイルストーンシステム
- ONSSI
- REACHエンジン
- SilverTrak
- SoftNAS
- QSTAR
- ベラシア

StorageGRID 11.2で検証済みのサードパーティ製ソリューション

以下の他社製ソリューションは、StorageGRID 11.2で検証済みです。+お探しの解決策が表示されない場合は、ネットアップの担当者までお問い合わせください。

StorageGRID で検証済みのサードパーティソリューション

これらのソリューションは、それぞれのパートナーと協力してテストされています。

- Actifio
- ブリッジストール
- カンテモ
- Citrixコンテンツコラボレーション
- Commvault 11.
- CTERAポータル6.
- ダレト
- ダタドビ
- Data Dynamics StorageXのように入力します
- DefendX
- Interica
- Komprise
- 良いですね
- Nasuni
- OpenText Documentum 16.4
- CyanGate Cloudを使用したOpenText Media Management 16.5
- Panzura
- Point Archival Gateway 2.0の場合
- Point Storage Manager 6.4
- プリミティブストリーム
- Quantum StorNext 5.4.0.1
- RUBRIK CDM 5.0.1 p1-1342
- シグニエント
- Splunk Smartstore
- ニスエンタープライズ6.0.4
- Veeam 9.5.4
- Veritas Enterprise Vault 11の略
- Veritas Enterprise Vault 12.
- Veritas NetBackup 8.0
- ビディズパイン

StorageGRIDでサポートされているサードパーティソリューション

これらのソリューションはテスト済みです。

- アーチウェア
- アクシスコミュニケーションズ
- コングルーシティ360
- DataFrameworksの略
- EcoDigital DIVAプラットフォーム
- Encoding.com
- FUJIFILMオブジェクトアーカイブ
- GE Centricity Enterprise Archiveの略
- ハイランド・アクオ
- IBM Aspera
- マイルストーンシステム
- ONSSI
- REACHエンジン
- SilverTrak
- SoftNAS
- QSTAR
- ベラシア

製品機能ガイド

AWSまたはGoogle Cloud用のクラウドストレージプールを作成します

StorageGRID オブジェクトを外部のS3バケットに移動する場合は、クラウドストレージプールを使用できます。外部バケットはAmazon S3（AWS）またはGoogle Cloudに属することができます。

必要なもの

- StorageGRID 11.6が設定されました。
- AWSまたはGoogle Cloudで外部のS3バケットをすでにセットアップしておきます。

手順

1. Grid Managerで、* ILM *>*ストレージプール*に移動します。
2. ページのクラウドストレージプールセクションで、 * 作成 * を選択します。

   クラウドストレージプールの作成ポップアップが表示されます。

3. 表示名を入力します。
4. [Provider Type]ドロップダウンリストから[**Amazon S3**]を選択します。

   このプロバイダタイプはAWS S3またはGoogle Cloudに対応しています。

5. クラウドストレージプールに使用するS3バケットのURIを入力します。

   次の2つの形式を使用できます。

   https://host:port`

   http://host:port`

6. S3バケット名を入力します。

   指定する名前はS3バケットの名前と完全に一致する必要があります。一致していないと、クラウドストレージプールの作成が失敗します。クラウドストレージプールの保存後にこの値を変更することはできません。

7. 必要に応じて、アクセスキーIDとシークレットアクセスキーを入力します。
8. ドロップダウンから[* Do not verify Certificate*（証明書を検証しない*）]を選択します。
9. [保存（ Save ）] をクリックします。

想定される結果です

Amazon S3またはGoogle Cloud用のクラウドストレージプールが作成されていることを確認します。

ジョナサン・ウォン著

Azure Blob Storage用のクラウドストレージプールを作成します

StorageGRID オブジェクトを外部のAzureコンテナに移動する場合は、クラウドストレージプールを使用できます。

必要なもの

- StorageGRID 11.6が設定されました。
- 外部のAzureコンテナはすでにセットアップされています。

手順

1. Grid Managerで、* ILM *>*ストレージプール*に移動します。
2. ページのクラウドストレージプールセクションで、 * 作成 * を選択します。

   クラウドストレージプールの作成ポップアップが表示されます。

3. 表示名を入力します。
4. プロバイダタイプドロップダウンリストから「* Azure Blob Storage *」を選択します。
5. クラウドストレージプールに使用するS3バケットのURIを入力します。

   次の2つの形式を使用できます。

   https://host:port`

   http://host:port`

6. Azureコンテナ名を入力します。

   指定する名前はAzureコンテナ名と完全に一致する必要があります。一致していないと、クラウドストレージプールの作成は失敗します。クラウドストレージプールの保存後にこの値を変更することはできません。

7. 必要に応じて、Azureコンテナに関連付けられたアカウント名と認証用のアカウントキーを入力します。
8. ドロップダウンから[* Do not verify Certificate*（証明書を検証しない*）]を選択します。
9. [保存（ Save ）] をクリックします。

想定される結果です

Azure Blob Storage用のクラウドストレージプールが作成されていることを確認します。

ジョナサン・ウォン著

クラウドストレージプールをバックアップに使用する

バックアップ用にクラウドストレージプールにオブジェクトを移動するILMルールを作成できます。

必要なもの

- StorageGRID 11.6が設定されました。
- 外部のAzureコンテナはすでにセットアップされています。

手順

1. Grid Managerで、* ILM > Rules > Create *の順に移動します。
2. 概要 を入力します。
3. ルールをトリガーする基準を入力します。
4. 「 * 次へ * 」をクリックします。
5. オブジェクトをストレージノードにレプリケートします。
6. 配置ルールを追加します。
7. オブジェクトをクラウドストレージプールにレプリケートします
8. 「 * 次へ * 」をクリックします。
9. [保存（ Save ）] をクリックします。

想定される結果です

保持図に、バックアップ用にStorageGRID とクラウドストレージプールにローカルに格納されているオブジェクトが示されていることを確認します。

ILMルールがトリガーされたときにクラウドストレージプールにコピーが存在し、オブジェクトのリストアを実行せずにローカルでオブジェクトを読み出すことができることを確認します。

ジョナサン・ウォン著

StorageGRID 検索統合サービスを設定する

このガイドでは、Amazon StorageGRID 11.6検索統合サービスとオンプレミスのElasticsearchを使用するようにNetAppを設定する手順について詳しく説明します。

はじめに

StorageGRID は、3種類のプラットフォームサービスをサポートしています。

- * StorageGRID CloudMirrorレプリケーション*。StorageGRID バケットから指定された外部のデスティネーションに特定のオブジェクトをミラーリングします。
- 通知。バケット単位のイベント通知：オブジェクトに対して実行された特定の処理に関する通知を、指定された外部のAmazon Simple Notification Service（Amazon SNS）に送信します。
- 検索統合サービス。外部サービスを使用してメタデータを検索または分析できるように、指定されたElasticsearchインデックスにSimple Storage Service（S3）オブジェクトメタデータを送信します。

プラットフォームサービスは、テナントマネージャのUIを使用してS3テナントによって設定されます。詳細については、を参照してください "プラットフォームサービスの使用に関する考慮事項"。

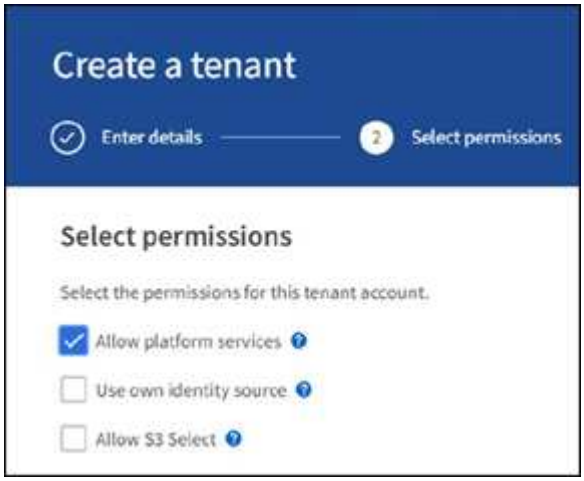
このドキュメントは、の補足資料として機能します "StorageGRID 11.6テナントガイド" およびに、検索統合サービス用のエンドポイントとバケットの設定手順と例を示します。ここで紹介するAmazon Web Services（AWS）またはオンプレミスのElasticsearchセットアップの手順は、基本的なテストやデモ目的にのみ使用

します。

対象読者は、Grid Manager、テナントマネージャに精通している必要があり、S3ブラウザにアクセスして、StorageGRID 検索統合テストの基本的なアップロード（PUT）処理とダウンロード（GET）処理を実行できます。

テナントを作成し、プラットフォームサービスを有効にします

1. Grid Managerを使用してS3テナントを作成し、表示名を入力してS3プロトコルを選択する。
2. [アクセス許可]ページで、[プラットフォームサービスを許可する]オプションを選択します。必要に応じて、他の権限を選択します。



3. テナントのrootユーザの初期パスワードを設定するか、グリッドでフェデレーションが有効になっている場合は、テナントアカウントを設定するためのrootアクセス権限を持つフェデレーテッドグループを選択します。
4. [ルートとしてサインイン]をクリックし、[バケット：バケットの作成と管理]を選択します。

   Tenant Managerのページが表示されます。

5. Tenant Managerで、My Access Keysを選択してS3アクセスキーを作成およびダウンロードし、あとでテストを実施します。

Amazon OpenSearchとの検索統合サービス

Amazon OpenSearch（旧Elasticsearch）サービスのセットアップ

この手順 は、テスト/デモ目的でのみOpenSearchサービスをすばやく簡単にセットアップするために使用します。検索統合サービスにオンプレミスのElasticsearchを使用している場合は、を参照してください 検索統合サービスをオンプレミスのElasticsearchと利用できます。

> OpenSearchサービスに登録するには、有効なAWSコンソールログイン、アクセスキー、シークレットアクセスキー、および権限が必要です。

1. の手順に従って、新しいドメインを作成します "AWS OpenSearchサービス開始前の準備"次の場合を除きます。
   - 手順 4ドメイン名：sgdemo

- 手順10：きめ細かなアクセスコントロール：「きめ細かなアクセスコントロールを有効にする」オプションの選択を解除します。
- 手順12.アクセスポリシー：Configure Level Access Policyを選択し、JSONタブを選択して次の例を使用してアクセスポリシーを変更します。
  - 強調表示されたテキストを、AWS Identity and Access Management（IAM）IDとユーザ名に置き換えます。
  - 強調表示されているテキスト（IPアドレス）を、AWSコンソールへのアクセスに使用したローカルコンピュータのパブリックIPアドレスに置き換えます。
  - ブラウザタブを開き、に移動します "https://checkip.amazonaws.com" をクリックして、パブリックIPを検索してください。

```
{

    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal":
        {"AWS": "arn:aws:iam:: nnnnnn:user/xyzabc"},
        "Action": "es:*",
        "Resource": "arn:aws:es:us-east-1:nnnnnn:domain/sgdemo/*"
        },
        {
        "Effect": "Allow",
        "Principal": {"AWS": "*"},
        "Action": [
        "es:ESHttp*"
                ],
        "Condition": {
            "IpAddress": {
                "aws:SourceIp": [ "nnn.nnn.nn.n/nn"
                    ]
                }
        },
        "Resource": "arn:aws:es:us-east-1:nnnnnn:domain/sgdemo/*"
        }
    ]
}
```

Fine-grained access control

Fine-grained access control provides numerous features to help you keep your data secure. Features include document-level security, field-level security, read-only users, and OpenSearch Dashboards/Kibana tenants. Fine-grained access control requires a master user. Learn more

☐ Enable fine-grained access control

SAML authentication for OpenSearch Dashboards/Kibana

SAML authentication lets you use your existing identity provider for single sign-on for OpenSearch Dashboards/Kibana. Learn more

Prepare SAML authentication

ⓘ To use SAML authentication, you must first enable fine-grained access control.

Amazon Cognito authentication

Enable to use Amazon Cognito authentication for OpenSearch Dashboards/Kibana. Amazon Cognito supports a variety of identity providers for username-password authentication. Learn more

☐ Enable Amazon Cognito authentication

Access policy

Access policies control whether a request is accepted or rejected when it reaches the Amazon OpenSearch Service domain. If you specify an account, user, or role in this policy, you must sign your requests. Learn more

Domain access policy

○ Only use fine-grained access control
Allow open access to the domain.

○ Do not set domain level access policy
All requests to the domain will be denied.

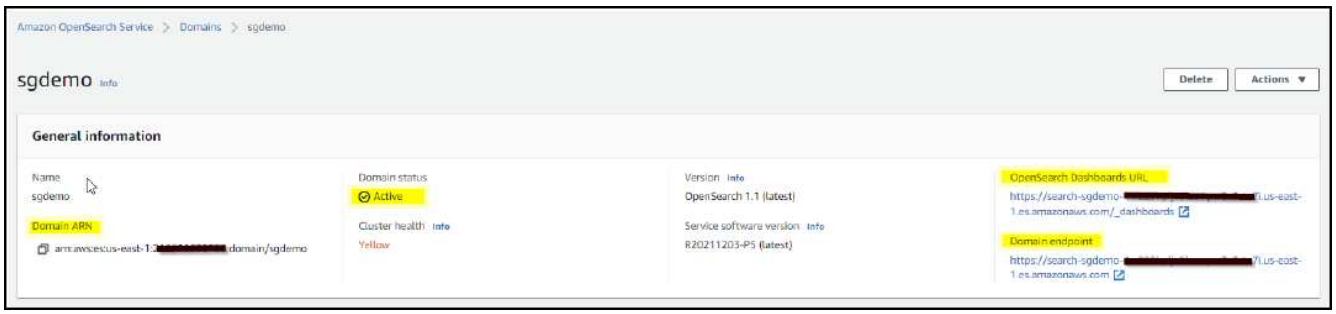
◉ Configure domain level access policy

Visual editor | JSON | Import policy

Access policy

```
3  "Statement": [
4    {
5      "Effect": "Allow",
6      "Principal": {
7        "AWS": "arn:aws:iam::2[redacted]:user/[redacted]"
8      },
9      "Action": "es:*",
10     "Resource": "arn:aws:es:us-east-1:2[redacted]:domain/sgdemo/*"
11   },
12   {
13     "Effect": "Allow",
14     "Principal": {
15       "AWS": "*"
16     },
17     "Action": [
18       "es:ESHttp*"
19     ],
20     "Condition": {
21       "IpAddress": {
22         "aws:SourceIp": [
23           "216.[redacted]/24"
24         ]
25       }
26     },
27     "Resource": "arn:aws:es:us-east-1:2[redacted]:domain/sgdemo/*"
28   }
```

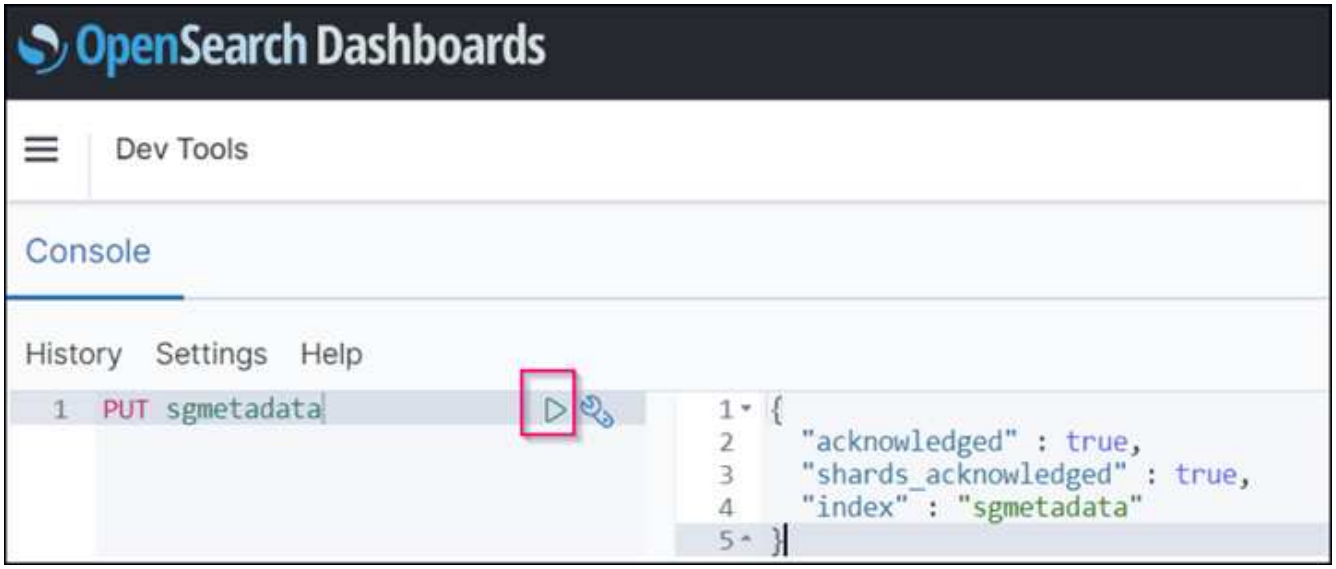
2. ドメインがアクティブになるまで15〜20分待ちます。



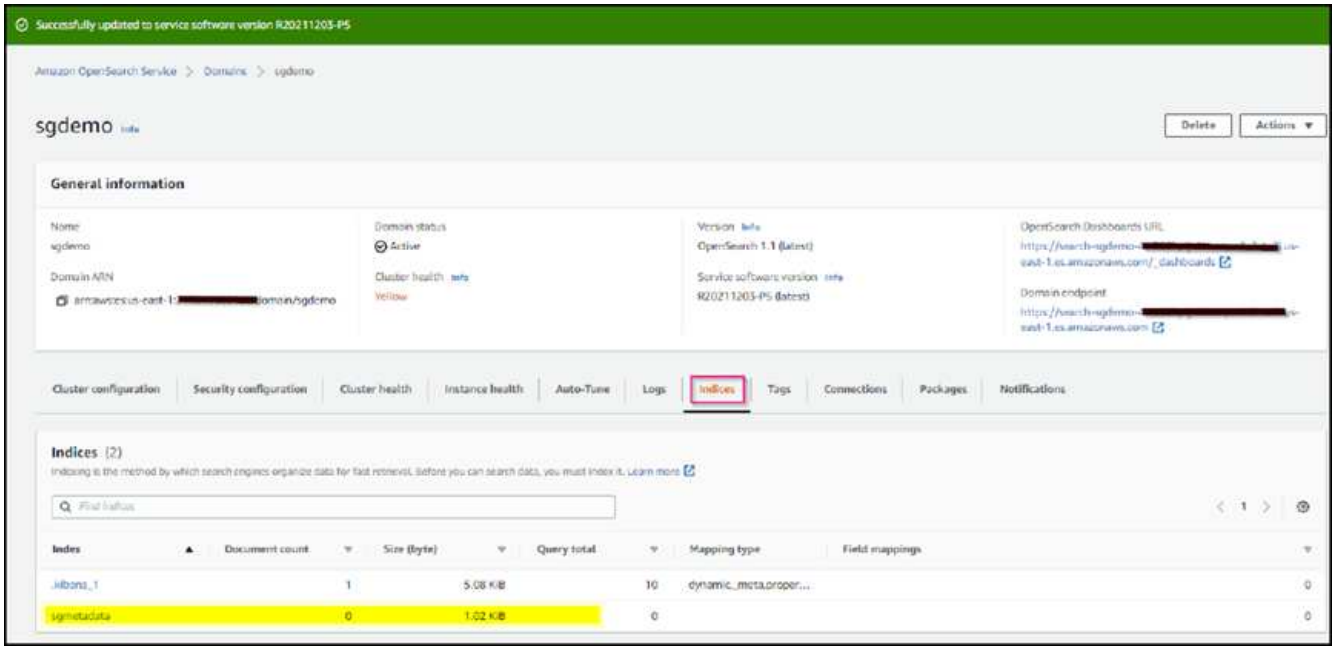
3. OpenSearch Dashboards URLをクリックして、新しいタブでドメインを開き、ダッシュボードにアクセスします。access deniedエラーが表示された場合は、アクセスポリシーのソースIPアドレスがコンピュータのパブリックIPに正しく設定されていて、ドメインダッシュボードへのアクセスが許可されていることを確認します。

4. ダッシュボードの開始ページで、自分で探索（Explore on your own）を選択します。メニューから、[管理]→[開発ツール]を選択します

5. Dev Tools → Consoleで、StorageGRID オブジェクトメタデータの保存にインデックスを使用する「Put <index>」と入力します。次の例では、インデックス名「メタデータ」を使用します。小さい三角形の記号をクリックして、PUTコマンドを実行します。次のスクリーンショットの例に示すように、正しい結果が右側のパネルに表示されます。



6. インデックスがAmazon OpenSearch UIのsgdomain > Indicesの下に表示されていることを確認します。

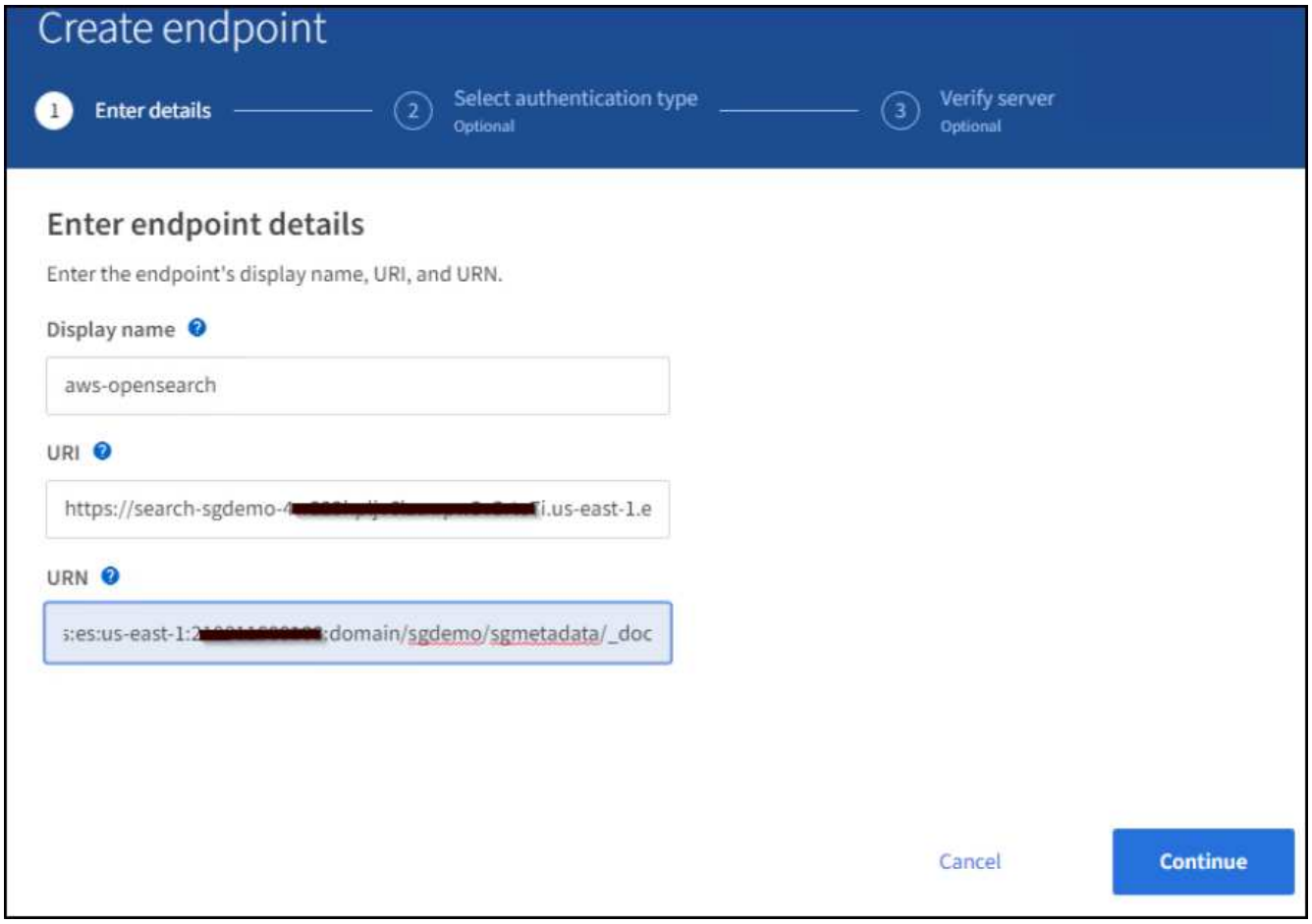


プラットフォームサービスエンドポイントの設定

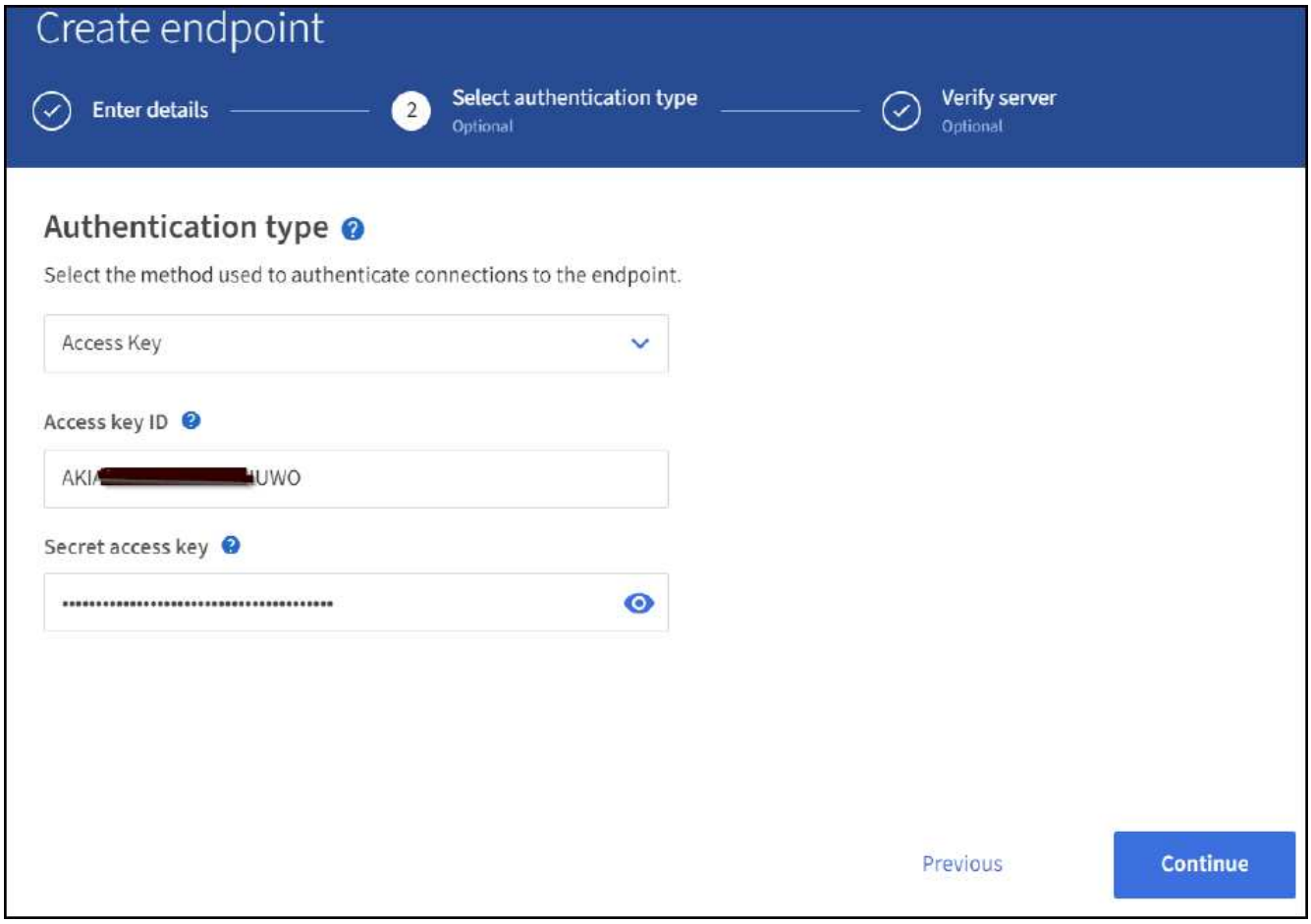
プラットフォームサービスエンドポイントを設定するには、次の手順を実行します。

1. Tenant Managerで、ストレージ（S3）>プラットフォームサービスのエンドポイントに移動します。
2. [エンドポイントの作成]をクリックし、次のように入力して、[続行]をクリックします。
    - 表示名の例は「AWS- OpenSearch」です
    - 手順 フィールドの前の「URI」の手順2の下のスクリーンショットのドメインエンドポイント。
    - URNフィールドで前の手順 の手順2で使用したドメインARNの末尾に'/<index>/_docを追加します

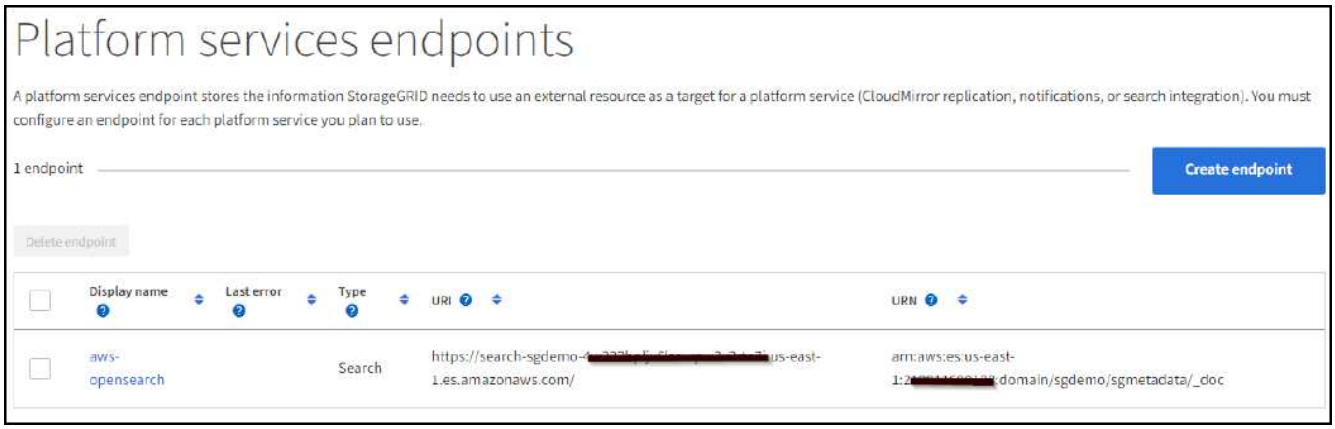
      この例では、URNはarn：aws：es：us-east -1：211234567890：domain/sgdemo/sgmedata/_docになります。



3. Amazon OpenSearchのsgdomainにアクセスするには、認証タイプとしてAccess Keyを選択し、Amazon S3のアクセスキーとシークレットキーを入力します。次のページに移動するには、[続行]をクリックします。



4. エンドポイントを確認するには、Use Operating System CA Certificate and Test and Create Endpointを選択します。検証に成功すると、次の図のようなエンドポイント画面が表示されます。検証に失敗した場合は、URNのパスの末尾に「/<index>/_doc」が含まれていて、AWSアクセスキーとシークレットキーが正しいことを確認してください。



検索統合サービスをオンプレミスの**Elasticsearch**と利用できます

オンプレミスの**Elasticsearch**セットアップ

この手順 は、テスト目的でのみDockerを使用するElasticsearchとKibanaオンプレミスを迅速にセットアップするためのものです。ElasticsearchサーバとKibanaサーバがすでに存在する場合は、ステップ5に進みます。

1. これを実行します "Dockerインストール手順 の略" Dockerをインストールするため。を使用します "CentOS Dockerは手順 をインストールする" このセットアップでは、

```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
sudo yum install docker-ce docker-ce-cli containerd.io
sudo systemctl start docker
```

   - リブート後にDockerを起動するには、次のように入力します。

```
sudo systemctl enable docker
```

   - 「vm.max_map_count」値を262144に設定します。

```
sysctl -w vm.max_map_count=262144
```

   - リブート後も設定を維持するには、次のように入力します。

```
echo 'vm.max_map_count=262144' >> /etc/sysctl.conf
```

2. に従ってください "Elasticsearchクイックスタートガイド" ElasticsearchとKibana Dockerを自己管理のためのセクションでインストールして実行できます。この例では、バージョン8.1をインストールしました。

> Elasticsearchが作成したユーザ名/パスワードとトークンをメモしておきます。これらのトークンは、Kibana UIおよびStorageGRID プラットフォームエンドポイント認証を開始するために必要です。

Elasticsearch Service | Self-managed

Install and run Elasticsearch

1. Install and start Docker Desktop.
2. Run:

```
docker network create elastic
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.1.0
docker run --name es-node01 --net elastic -p 9200:9200 -p 9300:9300 -it
```

When you start Elasticsearch for the first time, the following security configuration occurs automatically:

- Certificates and keys are generated for the transport and HTTP layers.
- The Transport Layer Security (TLS) configuration settings are written to `elasticsearch.yml`.
- A password is generated for the `elastic` user.
- An enrollment token is generated for Kibana.

NOTE: You might need to scroll back a bit in the terminal to view the password and enrollment token.

3. Copy the generated password and enrollment token and save them in a secure location. These values are shown only when you start Elasticsearch for the first time. You'll use these to enroll Kibana with your Elasticsearch cluster and log in.

NOTE: If you need to reset the password for the `elastic` user or other built-in users, run the `elasticsearch-reset-password` tool. To generate new enrollment tokens for Kibana or Elasticsearch nodes, run the `elasticsearch-create-enrollment-token` tool. These tools are available in the Elasticsearch `bin` directory.

Install and run Kibana

To analyze, visualize, and manage Elasticsearch data using an intuitive UI, install Kibana.

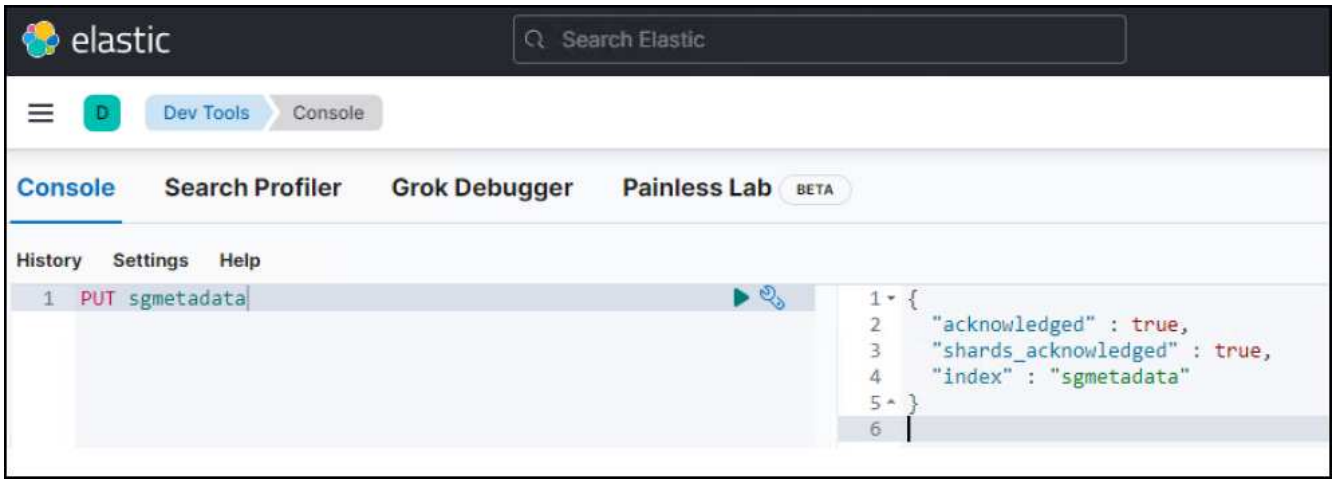
1. In a new terminal session, run:

```
docker pull docker.elastic.co/kibana/kibana:8.1.0
docker run --name kib-01 --net elastic -p 5601:5601 docker.elastic.co/ki
```

When you start Kibana, a unique link is output to your terminal.

2. To access Kibana, click the generated link in your terminal.
   a. In your browser, paste the enrollment token that you copied and click the button to connect your Kibana instance with Elasticsearch.
   b. Log in to Kibana as the `elastic` user with the password that was generated when you started Elasticsearch.

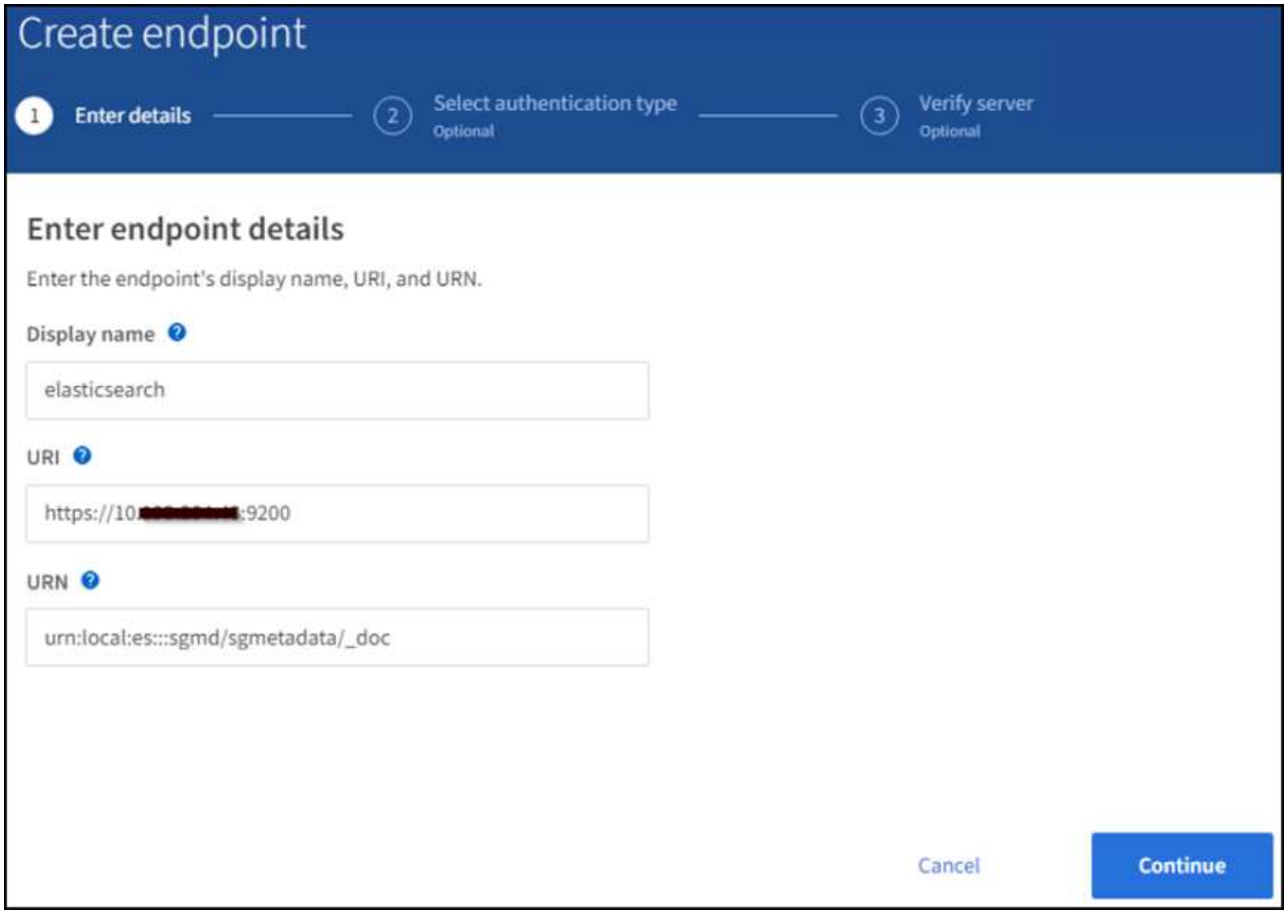
3. Kibana Dockerコンテナが起動すると、コンソールにURLリンク「\ https://0.0.0.0:5601`」が表示されます。0.0.0.0を、URL内のサーバIPアドレスと置き換えます。
4. ユーザ名「elastic」と、前述の手順でElasticによって生成されたパスワードを使用して、Kibana UIにログインします。
5. 初めてログインする場合は、ダッシュボードのようこそページで、自分でエクスプローラ（Explore on your own）を選択します。メニューから、Management > Dev Toolsを選択します。
6. Dev Tools Console画面で、StorageGRID オブジェクトメタデータの保存にこのインデックスを使用する「Put <index>」と入力します。この例では'インデックス名sgmetadataを使用します小さい三角形の記号をクリックして、PUTコマンドを実行します。次のスクリーンショットの例に示すように、正しい結果が右側のパネルに表示されます。



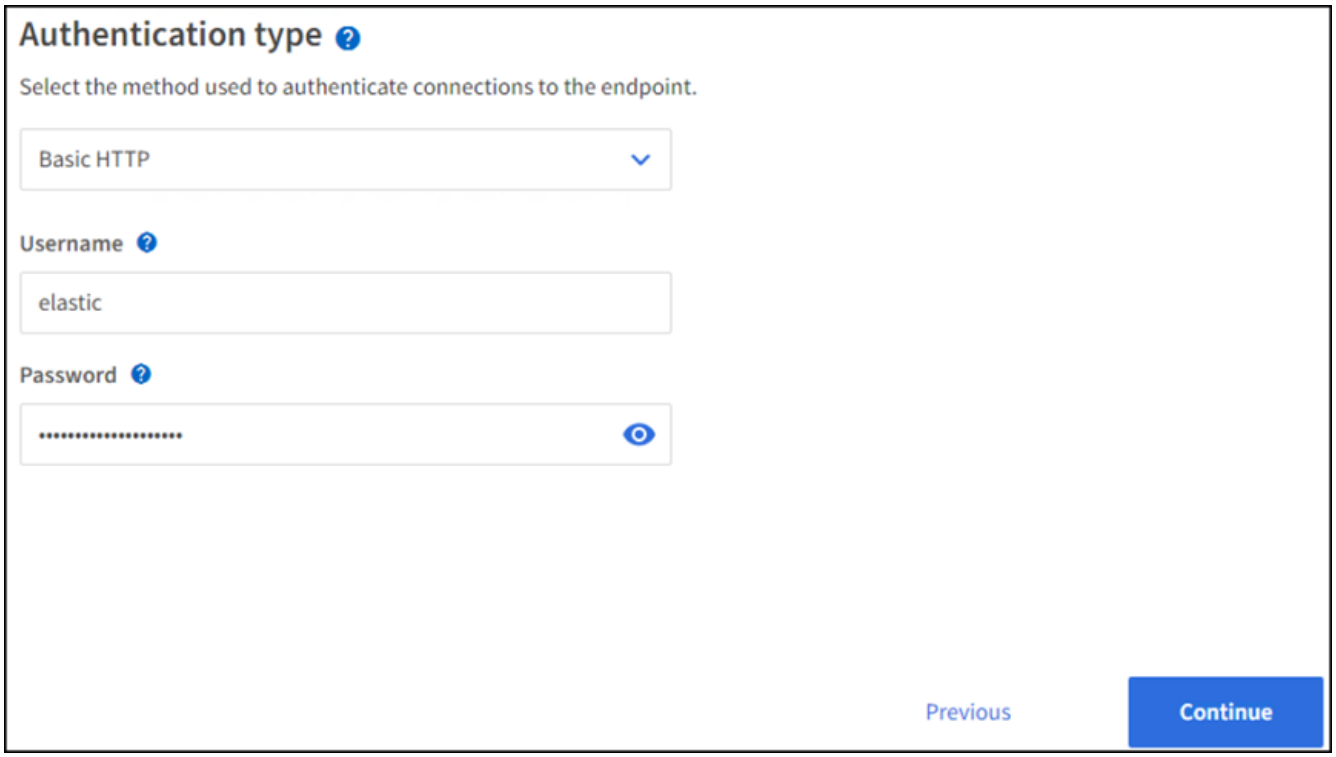
プラットフォームサービスエンドポイントの設定

プラットフォームサービスのエンドポイントを設定するには、次の手順を実行します。

1. Tenant Managerで、ストレージ（S3）>プラットフォームサービスのエンドポイントに移動します
2. [エンドポイントの作成]をクリックし、次のように入力して、[続行]をクリックします。
   - 表示名の例: elastic`
   - URI:`\https://<elasticsearch-server-ipまたはhostname>:9200`
   - urn:`urn:<何か>:es:::<se-unique text>/<index-name>/_doc`ここで、index-nameはKibanaコンソールで使用した名前です。例:`urn:local:es::sgmd/sgmetadata/_doc`

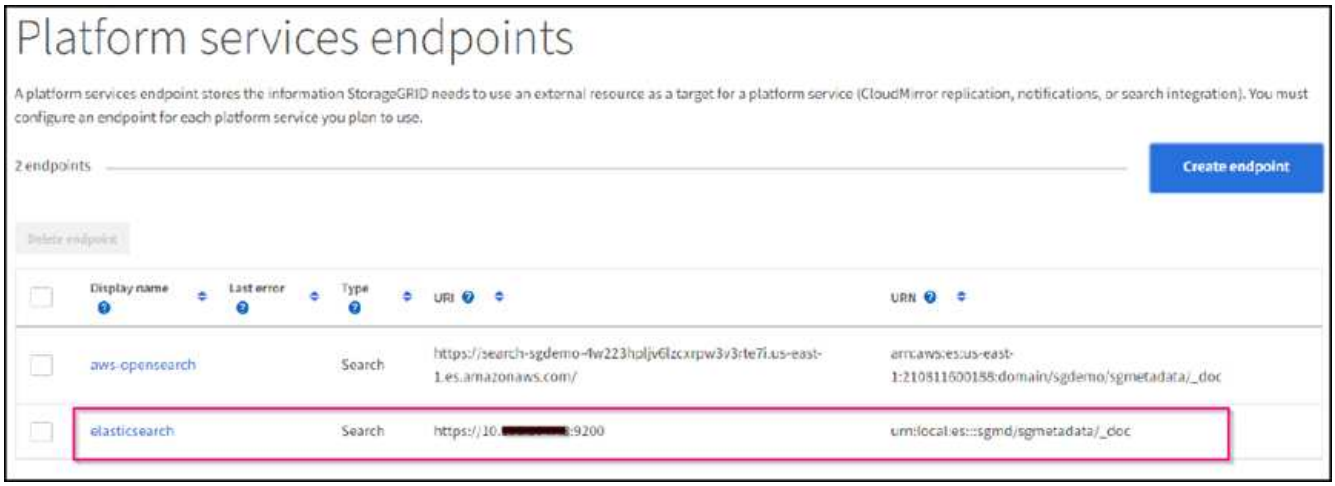


3. 認証タイプとしてBasic HTTPを選択し、Elasticsearchのインストールプロセスによって生成されたユーザ名「elastic」とパスワードを入力します。次のページに移動するには、[続行]をクリックします。



4. エンドポイントを確認するには、Do not verify Certificate and Test and Create Endpointを選択します。検

証に成功すると、次のスクリーンショットと同様のエンドポイント画面が表示されます。検証が失敗した場合は、URN、URI、およびユーザ名とパスワードのエントリが正しいことを確認してください。



バケット検索統合サービスの設定

プラットフォームサービスエンドポイントの作成後、次の手順では、オブジェクトの作成、削除、またはそのメタデータ/タグの更新が行われるたびに定義済みのエンドポイントにオブジェクトメタデータを送信するように、このサービスをバケットレベルで設定します。

Tenant Managerを使用して検索統合を設定し、カスタムのStorageGRID 設定XMLをバケットに次のように適用できます。

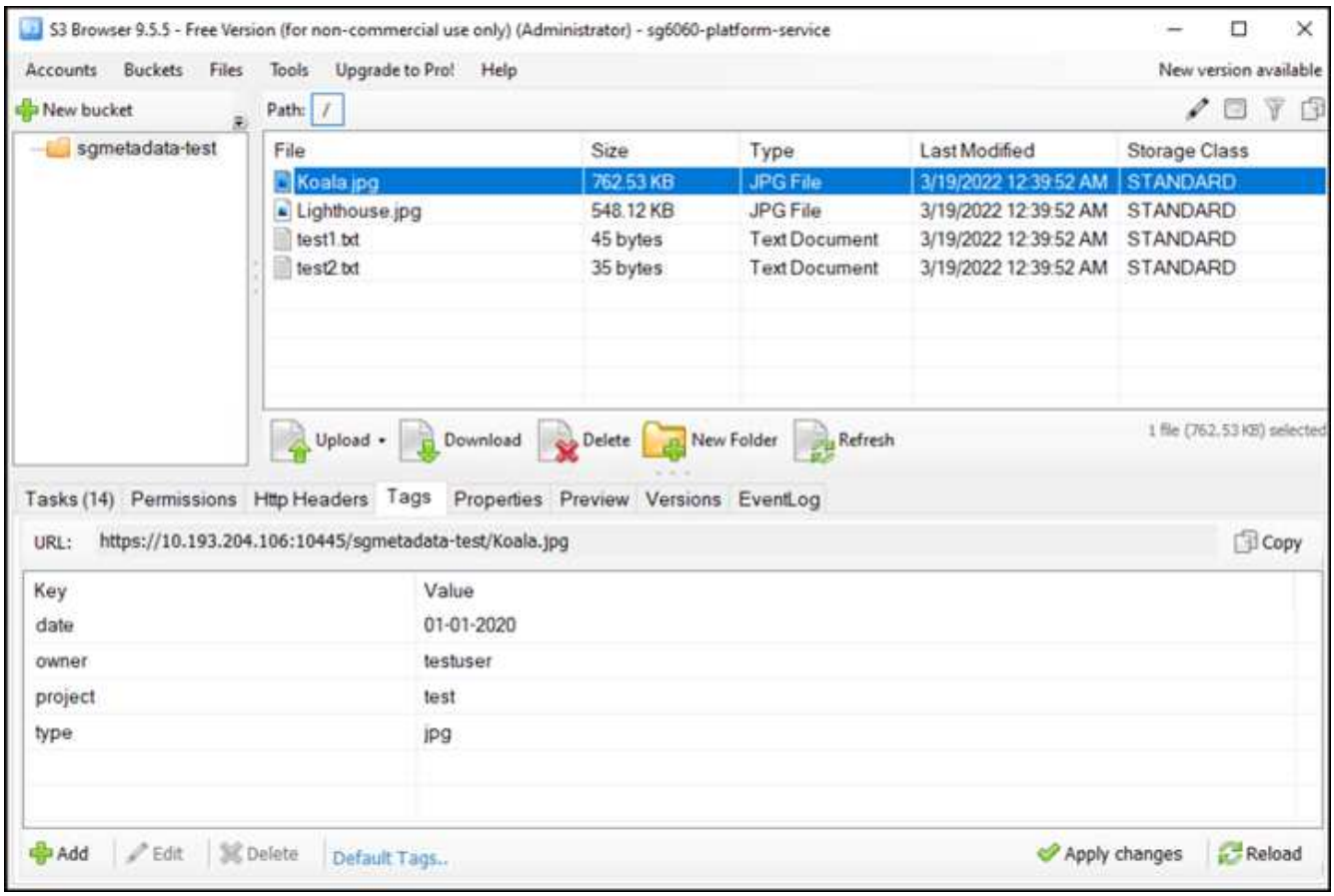
1. Tenant Managerで、Storage（S3）> Bucketsに移動します
2. Create Bucket（バケットの作成）をクリックし、バケット名（例：sgmetadatatest'）を入力して、デフォルトのus-east-1リージョンを受け入れます。
3. [Continue]>[Create Bucket]をクリックします。
4. バケットの概要ページを表示するには、バケット名をクリックし、プラットフォームサービスを選択します。
5. [検索統合を有効にする]ダイアログボックスを選択します。表示されたXMLボックスに、この構文を使用して設定XMLを入力します。

   強調表示されたURNは、定義したプラットフォームサービスエンドポイントと一致する必要があります。別のブラウザタブを開いてTenant Managerにアクセスし、定義済みのプラットフォームサービスエンドポイントからURNをコピーできます。

   この例ではプレフィックスを使用していません。つまり、このバケット内のすべてのオブジェクトのメタデータが、前に定義したElasticsearchエンドポイントに送信されます。

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled</Status>
        <Prefix></Prefix>
        <Destination>
            <Urn> urn:local:es:::sgmd/sgmetadata/_doc</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

6. S3 Browserを使用して、テナントアクセス/シークレットキーを使用してStorageGRID に接続し、テストオブジェクトを「sgmetadata-test」バケットにアップロードし、タグまたはカスタムメタデータをオブジェクトに追加します。



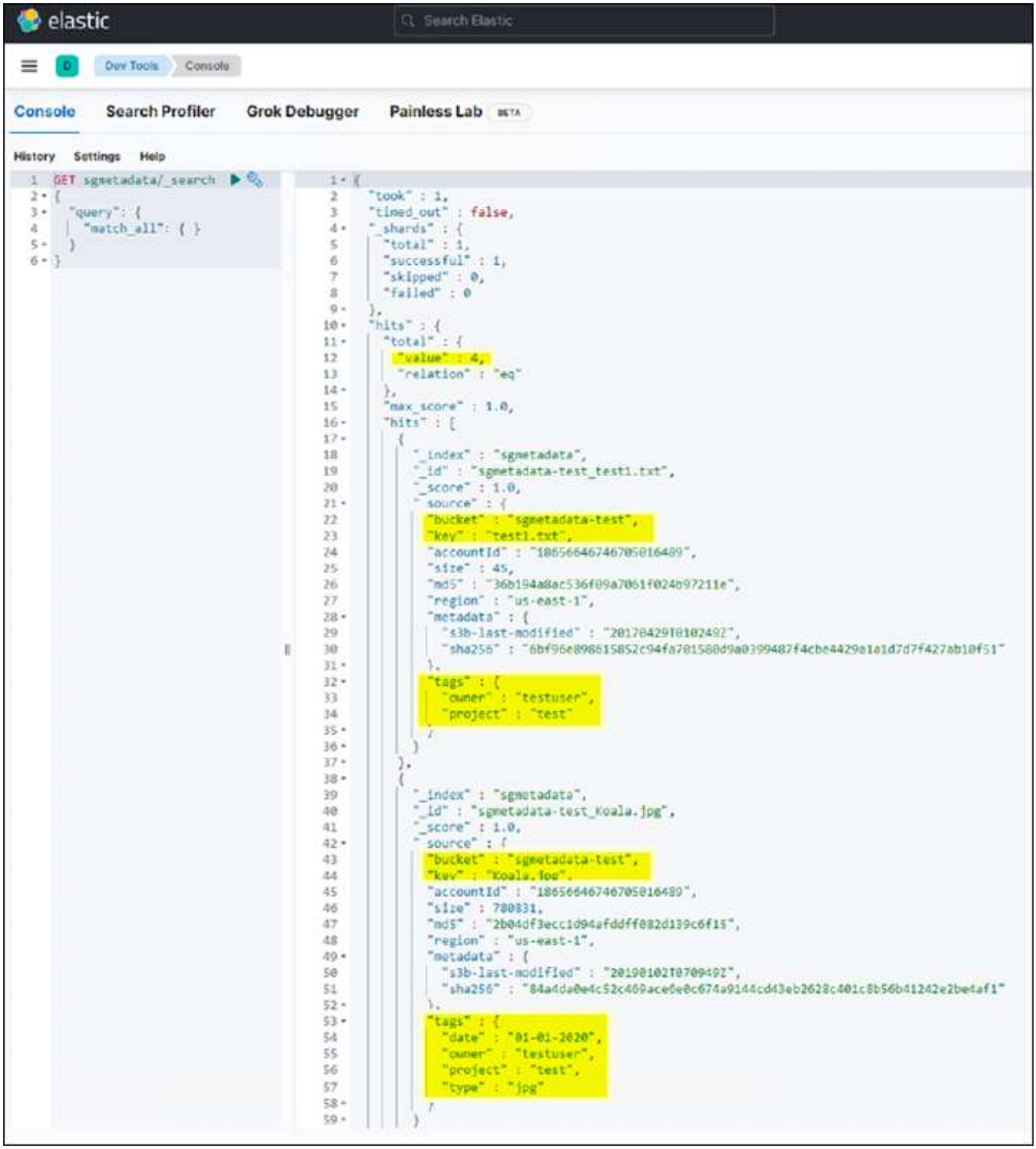
7. Kibana UIを使用して、オブジェクトメタデータがsgmetadataのインデックスにロードされたことを確認します。

   a. メニューから、Management > Dev Toolsを選択します。

   b. 左側のコンソールパネルにサンプルクエリを貼り付け、三角形の記号をクリックして実行します。

      次の例のスクリーンショットでは、クエリ1のサンプル結果に4つのレコードが表示されています。これはバケット内のオブジェクトの数に一致します。

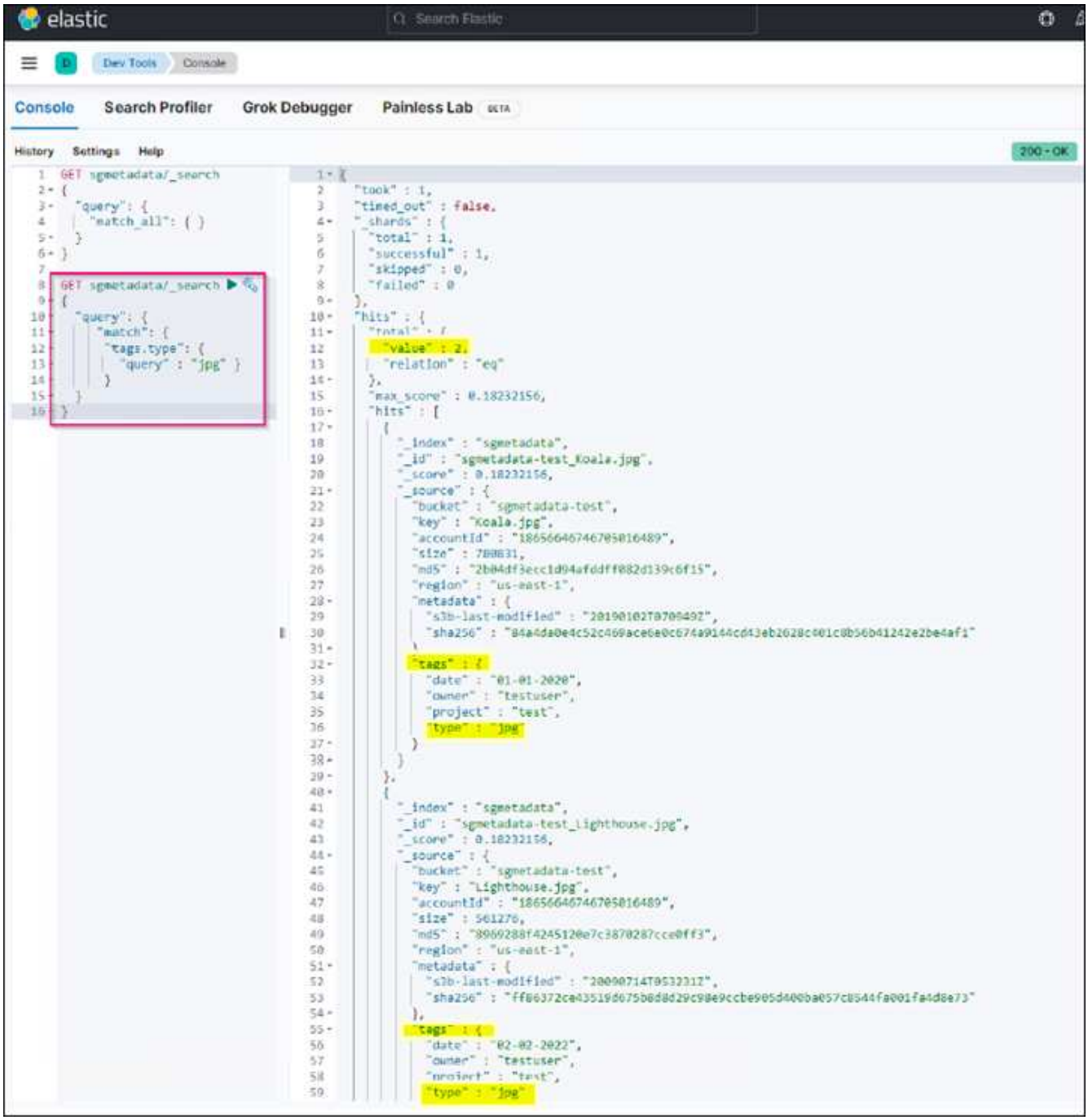
```
GET sgmetadata/_search
{
    "query": {
        "match_all": { }
}
}
```



次のスクリーンショットのクエリ2のサンプル結果は、タグタイプがjpgの2つのレコードを示しています。

```
GET sgmetadata/_search
{
    "query": {
        "match": {
            "tags.type": {
                "query" : "jpg" }
                }
            }
}
```

+



追加情報の参照先

このドキュメントに記載されている情報の詳細については、以下のドキュメントや Web サイトを参照してください。

- "プラットフォームサービスとは"
- "StorageGRID 11.6 ドキュメント"

Angela Cheng著_

ノードクローン

ノードクローンに関する考慮事項とパフォーマンス

ノードクローンに関する考慮事項

ノードクローンを使用すると、機器更改（Tech Refresh）の際に既存のアプライアンスノードをすばやく交換したり、容量を増やしたり、StorageGRID システムのパフォーマンスを向上させたりできます。ノードクローンは、KMSを使用したノード暗号化への変換や、ストレージノードをDDP8からDDP16に変更する場合にも役立ちます。

- ソースノードの使用済み容量は、クローンプロセスの完了に必要な時間とは関係ありません。ノードクローンは、ノードの空きスペースを含むノードのフルコピーです。
- ソースアプライアンスとデスティネーションアプライアンスのPGEバージョンが同じである必要があります
- デスティネーションノードの容量は常にソースノードよりも大きくする必要があります
  - 新しいデスティネーションアプライアンスのドライブサイズがソースよりも大きいことを確認します
  - デスティネーションアプライアンスのドライブサイズが同じで、DDP8用に設定されている場合は、DDP16用にデスティネーションを設定できます。ソースがすでにDDP16用に設定されている場合、ノードのクローニングは実行できません。
  - SG5660またはSG5760アプライアンスからSG6060アプライアンスに移行する場合、SG5x60には容量ドライブが60本搭載されていますが、SG6060には58本しか搭載されていません。
- ノードのクローニングプロセスでは、クローニングプロセスの実行中はソースノードがグリッドに対してオフラインになっている必要があります。この間に追加のノードがオフラインになると、クライアントサービスに影響する可能性があります。
- ストレージノードをオフラインにできるのは15日間だけです。クローニングプロセスの推定日数が15日に近い場合、または15日を超える場合は、拡張と運用停止の手順を使用します。
- 拡張シェルフを搭載したSG6060では、正しいシェルフドライブサイズの時間をベースアプライアンスの時間に追加して、フルクローン期間を取得する必要があります。
- ターゲットストレージアプライアンスのボリューム数は、ソースノードのボリューム数以上である必要があります。16個のオブジェクトストアボリューム（rangedb）を含むソースノードを、12個のオブジェクトストアボリュームを含むターゲットストレージアプライアンスにクローニングすることはできません。これは、ターゲットアプライアンスの容量がソースノードよりも大きい場合でも同様です。ほとんどのストレージアプライアンスにはオブジェクトストアボリュームが16個ありますが、オブジェクトストアボリュームが12個しかないSGF6112ストレージアプライアンスは除きます。たとえば、SG5760からSGF6112にクローニングすることはできません。

ノードクローンのパフォーマンスを見積もります

次の表に、ノードクローンの所要時間の推定値を示します。条件は状況によって異なるため、*太字*で示されたエントリは、ノードが停止した場合に15日を超えるリスクがあります。

DDP8

SG5612 →任意

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	1 日	2日	2.5日	3日	4日	4.5日
25GB	1 日	2日	2.5日	3日	4日	4.5日

SG5712 →任意

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	1 日	2日	2.5日	3日	4日	4.5日
25GB	1 日	2日	2.5日	3日	4日	4.5日

SG5660 → **SG5760**を選択してください

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	3日間	6日	7 日	8.5日	11.5日	• 13日*
25GB	3日間	6日	7 日	8.5日	11.5日	• 13日*

SG5660 → **SG6060**

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	2.5日	4.5日	5.5日	6.5日	9日	10日間
25GB	2日間	4日	5日	6日	8日間	9日

SG5760 → **SG5760**

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	3日間	6日	7 日	8.5日	11.5日	• 13日*
25GB	3日間	6日	7 日	8.5日	11.5日	• 13日*

SG5760 → SG6060

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	2.5日	4.5日	5.5日	6.5日	9日	10日間
25GB	1.5日	3日	3.5日	4.5日	6日	6.5日

SG6060 → SG6060

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	2.5日	4.5日	5.5日	6.5日	8.5日	9.5日
25GB	1.5日	3日	3.5日	4日	5.5日	6日

DDP16

SG5760 → SG5760

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	3.5日	6.5日	8日間	9.5日	12.5日	• 14日*
25GB	3.5日	6.5日	8日間	9.5日	12.5日	• 14日*

SG5760 → SG6060

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	2.5日	5日	6日	7.5日	10日間	11日だ
25GB	2日間	3.5日	4日	5日	6.5日	7 日

SG6060 → SG6060

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	3.5日	5日	6日	7 日	9.5日	10.5日
25GB	2日間	3日	4日	4.5日	6日	7 日

拡張シェルフ（ソースアプライアンスの各シェルフについて、上記の**SG6060**に追加）

ネットワークインターフェイスの速度	**4TB**ドライブサイズ	**8TB**ドライブサイズ	**10TB**ドライブサイズ	**12TB**ドライブサイズ	**16TB**ドライブサイズ	**18TB**のドライブサイズ
10Gb	3.5日	5日	6日	7 日	9.5日	10.5日
25GB	2日間	3日	4日	4.5日	6日	7 日

アロンクライン著

ポート再マッピングの使用方法

さまざまな理由で、着信ポートまたは発信ポートの再マッピングが必要になることがあります。従来のCLBロードバランササービスから現在のnginxサービスロードバランサエンドポイントに移行し、同じポートを維持してクライアントへの影響を軽減する場合、管理ノードクライアントネットワークのクライアントS3にポート443を使用する場合、またはファイアウォールの制限に使用する場合があります。

ポートの再マッピングを使用して、**S3**クライアントを**CLB**から**NGINX**に移行します

StorageGRID 11.3より前のリリースでは、ゲートウェイノードに含まれているロードバランササービスはConnection Load Balancer（CLB）です。StorageGRID 11.3では、HTTPトラフィックのロードバランシングを実現する機能豊富な統合解決策 として、NGINXサービスが導入されました。CLBサービスは現在のリリースのStorageGRID でも引き続き使用できるため、新しいロードバランサエンドポイントの設定でポート8082を再利用することはできません。この問題を回避するために、8082インバウンドポートが10443に再マッピングされます。これにより、ゲートウェイのポート8082に着信するすべてのHTTPS要求は、CLBサービスをバイパスしてNGINXサービスに接続し、ポート10443にリダイレクトされます。以下の手順はVMwareを対象としていますが、PORT_REMAP機能はすべてのインストール方法に適用され、ベアメタル環境とアプライアンスでも同様のプロセスを使用できます。

VMware仮想マシンゲートウェイノードの導入

次の手順は、StorageGRID Open Virtualization Format（OVF）を使用してゲートウェイノードをVMとしてVMware vSphere 7に導入するStorageGRID 環境を対象としています。このプロセスでは、VMを破壊的に削除し、同じ名前と構成でVMを再導入します。VMの電源をオンにする前に、vAppプロパティを変更してポートを再マッピングし、VMの電源をオンにしてノードのリカバリプロセスに従います。

前提条件

- StorageGRID 11.3以降を実行している
- インストールされているStorageGRID バージョンのVMwareインストールファイルをダウンロードし、アクセスできるようにしておきます。
- VMの電源オン/オフ、VMおよびvAppの設定の変更、vCenterからのVMの削除、OVFによるVMの導入を行う権限を持つvCenterアカウントが必要です。
- ロードバランサエンドポイントを作成しておきます
  - ポートが目的のリダイレクトポートに設定されている
  - エンドポイントのSSL証明書がCLBサービス用の[Configuration]/[Server Certificates]/[Object Storage API Service Endpoints Server Certificate]にインストールされているものと同じであるか、クライアントが証明書の変更を承認できる。

```
If your existing certificate is self-signed, you cannot reuse it
in the new endpoint. You must generate a new self-signed
certificate when creating the endpoint and configure the clients
to accept the new certificate.
```

最初のゲートウェイノードを削除します

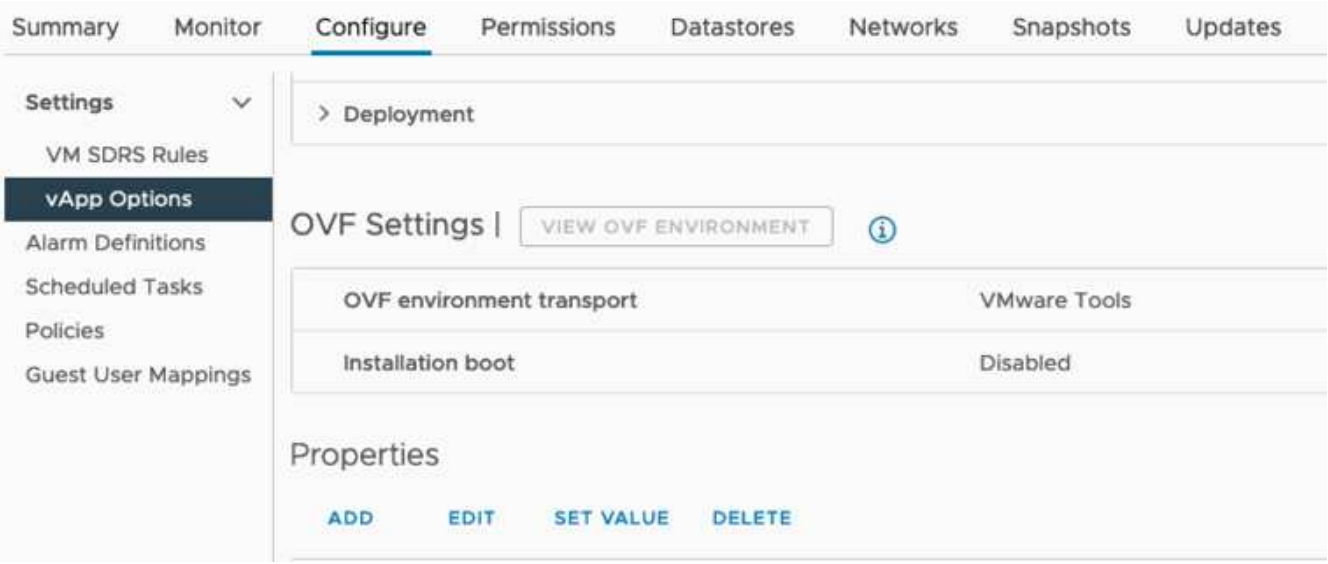
最初のゲートウェイノードを削除するには、次の手順を実行します。

1. グリッドに複数のノードがある場合は、開始するゲートウェイノードを選択します。
2. 必要に応じて、すべてのDNSラウンドロビンエンティティまたはロードバランサプールからノードIPを削除します。
3. Time-To-Live（TTL）と開いているセッションが期限切れになるまで待ちます。
4. VMノードの電源をオフにします。
5. ディスクからVMノードを削除します。

交換用ゲートウェイノードを導入します

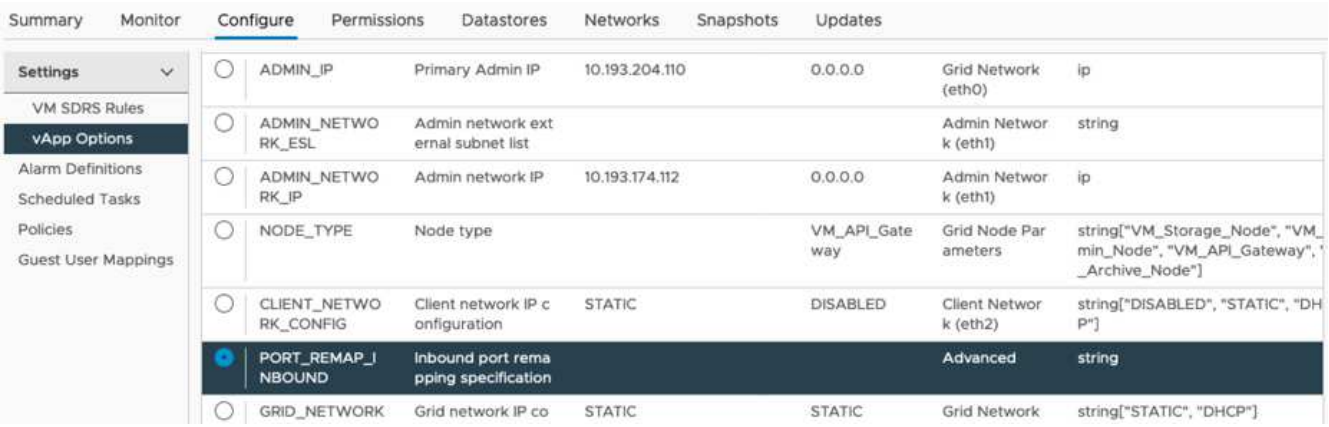
交換用ゲートウェイノードを導入するには、次の手順を実行します。

1. サポートサイトからダウンロードしたインストールパッケージから.ovf、.mf、.vmdkファイルを選択して、OVFから新しいVMを導入します。
   - vsphere-gateway.mf
   - vSphere-gateway.ovf
   - NetApp-sg-11.4.0-20200721.1338.d3969b3.vmdk
2. 導入が完了したら、VMのリストからVMを選択し、[Configure]タブ[vApp Options]を選択します。



タブ"]

3. [Properties]セクションまで下にスクロールし、PORT_REMAP_INBOUNDプロパティを選択します



4. [プロパティ]リストの一番上までスクロールし、[編集]をクリックします

ボタン"]

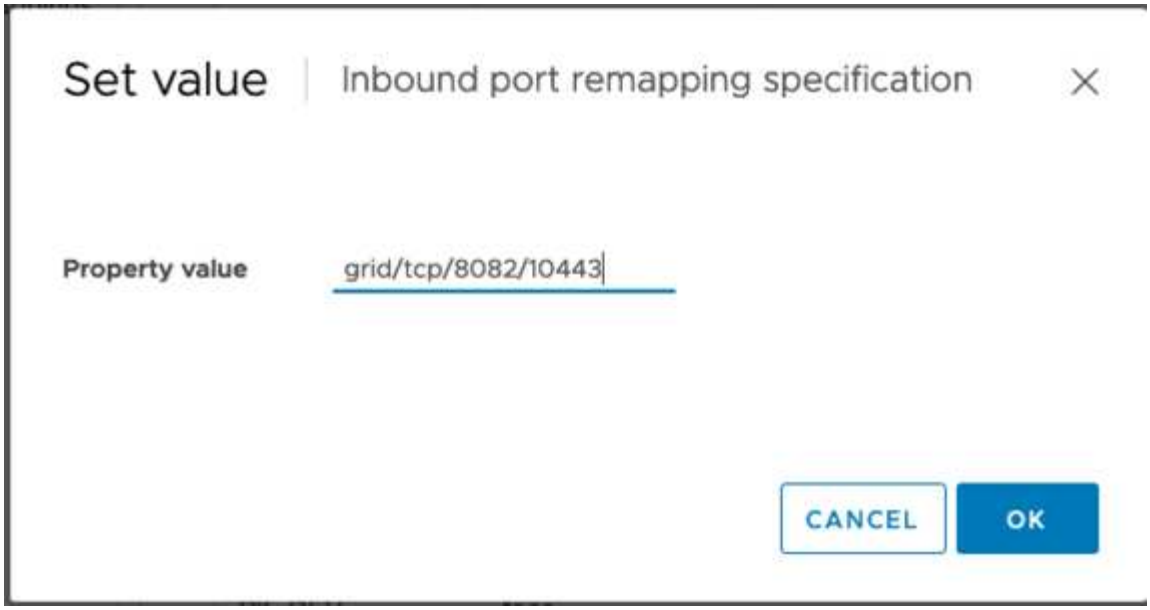
5. [タイプ]タブを選択し、[ユーザー設定可能]チェックボックスがオンになっていることを確認して、[保存]をクリックします。

タブ"]

6. 「PORT_REMAP_INBOUND」プロパティが選択された状態で、[Properties]リストの上部にある[Set Value]をクリックします。

ボタン"]

7. [Property Value]フィールドに、ネットワーク（グリッド、管理者、またはクライアント）、TCP、元のポート（8082）、および新しいポート（10443）をそれぞれの値の間にを含めて入力します（次の図を参照）。

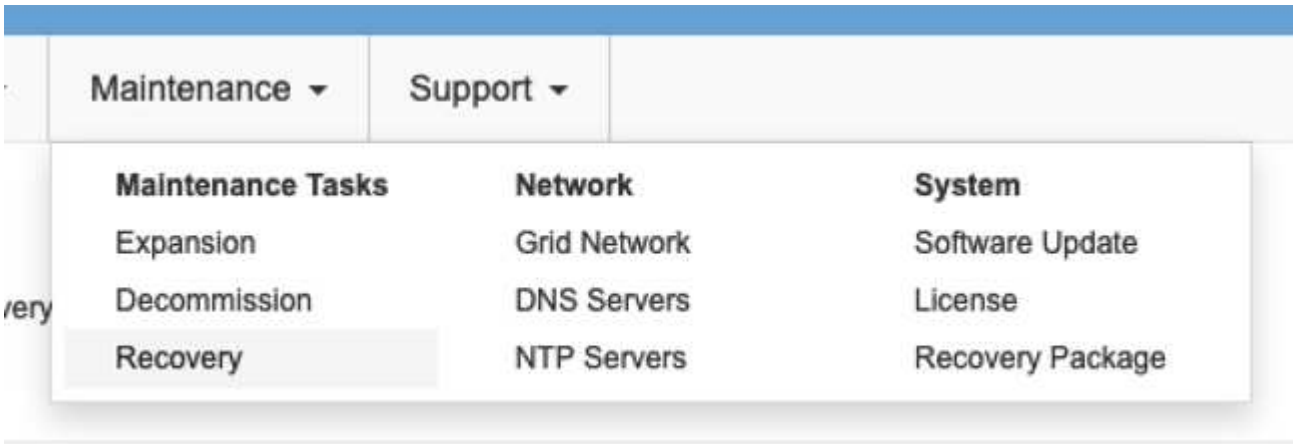


8. 複数のネットワークを使用している場合は、カンマ（,）を使用してネットワークの文字列を区切ります（例：grid/tcp/8082/10443、admin/tcp/8082/10443、client/tcp/8082/10443）

ゲートウェイノードをリカバリ

ゲートウェイノードをリカバリするには、次の手順を実行します。

1. グリッド管理UIの[Maintenance/Recovery]セクションに移動します。



メニュー"]

2. VMノードの電源をオンにし、ノードがグリッド管理UIの[Maintenance/Recovery Pending Nodes]セクションに表示されるまで待ちます。


```
For information and directions for node recovery, see the
https://docs.netapp.com/sgws-114/topic/com.netapp.doc.sg-
maint/GUID-7E22B1B9-4169-4800-8727-75F25FC0FFB1.html[Recovery
and Maintenance guide]
```

3. ノードのリカバリが完了したら、すべてのDNSラウンドロビンエンティティまたはロードバランサプール（該当する場合）にIPを含めることができます。

   これで、ポート8082のHTTPSセッションはポート10443に接続されます

管理ノードでクライアント**S3**アクセス用のポート**443**を再マッピングします

StorageGRID システムでの管理ノード、または管理ノードを含むHAグループのデフォルトの設定は、ポート443と80が管理およびテナントマネージャUI用に予約されており、ロードバランサエンドポイントには使用できません。これへの解決策 では、ポートの再マッピング機能を使用して、インバウンドポート443をロードバランサエンドポイントとして設定される新しいポートにリダイレクトします。完了したクライアントS3トラフィックでポート443を使用できるようになると、グリッド管理UIにはポート8443経由でのみアクセスでき、テナント管理UIにはポート9443経由でのみアクセスできるようになります。ポートの再マッピング機能は、ノードのインストール時にのみ設定できます。グリッド内のアクティブノードのポートの再マッピングを実装するには、そのノードをインストール前の状態にリセットする必要があります。これは破壊的な手順 であり、設定の変更後にノードをリカバリすることも含まれます。

ログとデータベースをバックアップします

管理ノードには、監査ログ、Prometheus指標のほか、属性、アラーム、アラートに関する履歴情報が格納されます。管理ノードが複数あるということは、このデータのコピーが複数あることを意味します。グリッドに管理ノードが複数ない場合は、このプロセスの最後でノードがリカバリされたあとにリストアできるように、このデータを保持しておく必要があります。グリッドに別の管理ノードがある場合は、リカバリプロセス中にそのノードからデータをコピーできます。グリッド内に別の管理ノードがない場合は、ノードを破棄する前に、次の手順に従ってデータをコピーできます。

監査ログをコピーする

1. 管理ノードにログインします。
   a. 次のコマンドを入力します。 `ssh admin@grid_node_IP`
   b. に記載されているパスワードを入力します `Passwords.txt` ファイル。
   c. 次のコマンドを入力してrootに切り替えます。 `su -`

d. に記載されているパスワードを入力します `Passwords.txt` ファイル。

e. SSH エージェントに SSH 秘密鍵を追加します。入力するコマンド `ssh-add`

f. に記載されているSSHアクセスパスワードを入力します `Passwords.txt` ファイル。

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. すべての監査ログファイルを別のグリッドノードの一時的な場所にコピーするディレクトリを作成します。lets use_storage_node_01_：

   a. `ssh admin@storage_node_01_IP`

   b. `mkdir -p /var/local/tmp/saved-audit-logs`

3. 管理ノードに戻り、AMSサービスを停止して新しいログファイルが作成されないようにします。 `service ams stop`

4. audit.log ファイルの名前を変更して、リカバリした管理ノードへのコピー時に既存のファイルが上書きされないようにします。

   a. audit.log の名前を、 yyyy-mm-dd.txt.1 などの一意の番号の付いたファイル名に変更します。たとえば、監査ログファイルの名前を2015-10-25.txt.1に変更できます

```
cd /var/local/audit/export
ls -l
mv audit.log 2015-10-25.txt.1
```

5. AMSサービスを再起動します。 `service ams start`

6. すべての監査ログファイルをコピーします。 `scp * admin@storage_node_01_IP:/var/local/tmp/saved-audit-logs`

Prometheusデータをコピー

> Prometheus データベースのコピーには 1 時間以上かかる場合があります。管理ノードでサービスが停止している間は、Grid Managerの一部の機能が使用できなくなります。

1. Prometheusデータを別のグリッドノードの一時的な場所にコピーするディレクトリを作成します。この場合もuser_storage_node_01_：

   a. ストレージノードにログインします。

      i. 次のコマンドを入力します。 `ssh admin@storage_node_01_IP`

      ii. に記載されているパスワードを入力します `Passwords.txt` ファイル。

      iii. mkdir -p /var/local/tmp/prometheus

2. 管理ノードにログインします。

   a. 次のコマンドを入力します。 `ssh admin@admin_node_IP`

   b. に記載されているパスワードを入力します `Passwords.txt` ファイル。

c. 次のコマンドを入力してrootに切り替えます。 `su -`

d. に記載されているパスワードを入力します `Passwords.txt` ファイル。

e. SSH エージェントに SSH 秘密鍵を追加します。入力するコマンド `ssh-add`

f. に記載されているSSHアクセスパスワードを入力します `Passwords.txt` ファイル。

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

3. 管理ノードから、Prometheusサービスを停止します。 `service prometheus stop`

   a. ソース管理ノードのPrometheusデータベースをストレージノードのバックアップ先ノードにコピーします。 `/rsync -azh --stats "/var/local/mysql_ibdata/prometheus/data" "storage_node_01_IP:/var/local/tmp/prometheus/"`

4. ソース管理ノードでPrometheusサービスを再起動します.`service prometheus start`

履歴情報をバックアップします

履歴情報はMySQLデータベースに保存されます。データベースのコピーをダンプするには、ネットアップのユーザとパスワードが必要です。グリッド内に別の管理ノードがある場合は、この手順は必要なく、リカバリプロセス中に残りの管理ノードからデータベースをクローニングできます。

1. 管理ノードにログインします。

   a. 次のコマンドを入力します。 `ssh admin@admin_node_IP`

   b. に記載されているパスワードを入力します `Passwords.txt` ファイル。

   c. 次のコマンドを入力してrootに切り替えます。 `su -`

   d. に記載されているパスワードを入力します `Passwords.txt` ファイル。

   e. SSH エージェントに SSH 秘密鍵を追加します。入力するコマンド `ssh-add`

   f. に記載されているSSHアクセスパスワードを入力します `Passwords.txt` ファイル。

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 管理ノードでStorageGRID サービスを停止し、NTPとMySQLを起動します

   a. すべてのサービスを停止します。 `service servermanager stop`

   b. NTPサービスを再開します。 `service ntp start`.. MySQLサービスを再起動します。 `service mysql start`

3. miデータベースを/var/local/tmpにダンプします

   a. 次のコマンドを入力します。 `mysqldump -u username -p password mi > /var/local/tmp/mysql-mi.sql`

4. MySQLダンプファイルを別のノードにコピーします。ここでは_storage_node_01を使用します。 `scp /var/local/tmp/mysql-mi.sql storage_node_01_IP:/var/local/tmp/mysql-mi.sql`

   a. 他のサーバにパスワードなしでアクセスする必要がなくなった場合は、 SSH エージェントから秘密鍵

を削除します。入力するコマンド `ssh-add -D`

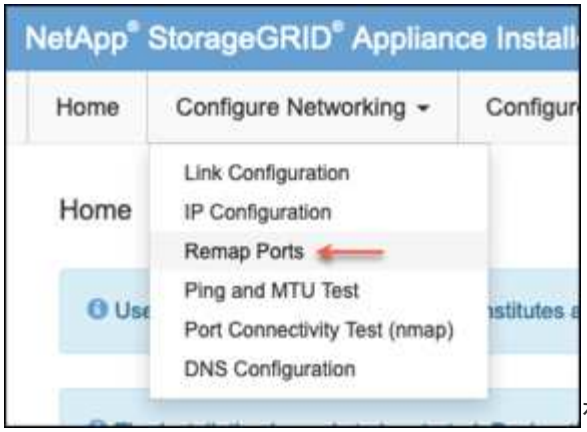
管理ノードをリビルドします

グリッド内の別の管理ノードに必要なすべてのデータとログのバックアップコピーが作成されたか、一時的な場所に格納されたので、次にアプライアンスをリセットしてポートの再マッピングを設定します。

1. アプライアンスをリセットすると、アプライアンスは事前にインストールされた状態に戻り、ホスト名、IP、およびネットワーク設定のみが保持されます。すべてのデータが失われるため、重要な情報のバックアップが確実に作成されます。
   a. 次のコマンドを入力します。 `sgareinstall`

```
root@sg100-01:~ # sgareinstall
WARNING: All StorageGRID Webscale services on this node will be shut
down.
WARNING: Data stored on this node may be lost.
WARNING: You will have to reinstall StorageGRID Webscale to this
node.

After running this command and waiting a few minutes for the node to
reboot,
browse to one of the following URLs to reinstall StorageGRID Webscale
on
this node:

    https://10.193.174.192:8443
    https://10.193.204.192:8443
    https://169.254.0.1:8443

Are you sure you want to continue (y/n)? y
Renaming SG installation flag file.
Initiating a reboot to trigger the StorageGRID Webscale appliance
installation wizard.
```

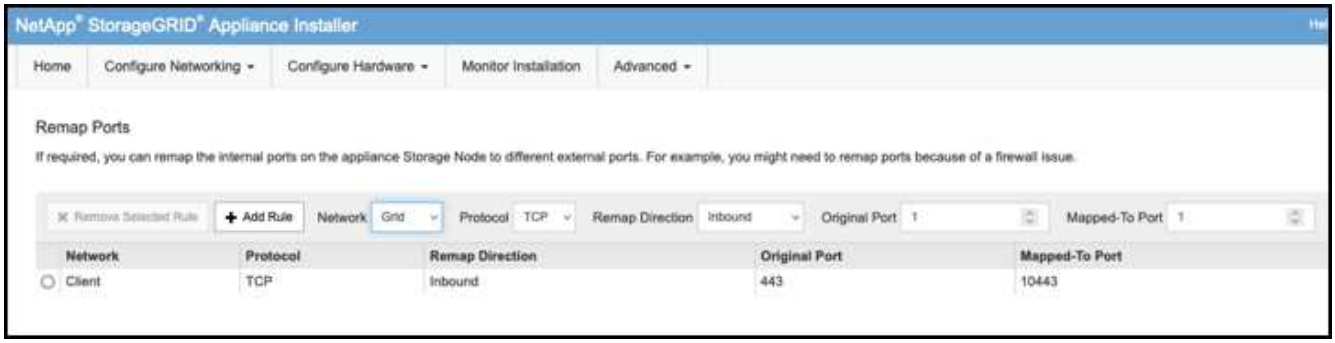
2. しばらくするとアプライアンスがリブートし、ノードのPGE UIにアクセスできるようになります。
3. [Configure Networking]にアクセスします



を選択します"]

4. 目的のネットワーク、プロトコル、方向、およびポートを選択し、[Add Rule]ボタンをクリックします。

> グリッドネットワーク上のインバウンドポート443を再マッピングすると、インストールおよび拡張手順が中断されます。グリッドネットワークのポート443を再マッピングすることは推奨されません。



5. 必要なポートの再マッピングが追加されている場合は、ホームタブに戻り、[Start Installation]ボタンをクリックします。

で管理ノードのリカバリ手順を実行できるようになりました "製品ドキュメント"

データベースとログをリストアします

管理ノードのリカバリが完了したら、指標、ログ、履歴情報をリストアできます。グリッドに別の管理ノードがある場合は、に従ってください "製品ドキュメント" _prometheus-clone-db.sh_and_mi-clone-db.sh_scripts を使用する。これが唯一の管理ノードであり、このデータをバックアップすることを選択した場合は、次の手順に従って情報を復元できます。

監査ログをコピーして元に戻します

1. 管理ノードにログインします。
   a. 次のコマンドを入力します。 `ssh admin@grid_node_IP`
   b. に記載されているパスワードを入力します `Passwords.txt` ファイル。
   c. 次のコマンドを入力してrootに切り替えます。 `su -`
   d. に記載されているパスワードを入力します `Passwords.txt` ファイル。

e. SSH エージェントに SSH 秘密鍵を追加します。入力するコマンド `ssh-add`

f. に記載されているSSHアクセスパスワードを入力します `Passwords.txt` ファイル。

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 保持されている監査ログファイルをリカバリされた管理ノードにコピーします。 `scp admin@grid_node_IP:/var/local/tmp/saved-audit-logs/YYYY* .`

3. セキュリティ上の理由により、監査ログがリカバリされた管理ノードにコピーされたことを確認したら、監査ログを障害グリッドノードから削除します。

4. リカバリされた管理ノードで、監査ログファイルのユーザとグループの設定を更新します。 `chown ams-user:bycast *`

監査共有への既存のクライアントアクセスもリストアする必要があります。詳細については、 StorageGRID の管理手順を参照してください。

Prometheus指標をリストア

> Prometheus データベースのコピーには 1 時間以上かかる場合があります。管理ノードでサービスが停止している間は、Grid Managerの一部の機能が使用できなくなります。

1. 管理ノードにログインします。

   a. 次のコマンドを入力します。 `ssh admin@grid_node_IP`

   b. に記載されているパスワードを入力します `Passwords.txt` ファイル。

   c. 次のコマンドを入力してrootに切り替えます。 `su -`

   d. に記載されているパスワードを入力します `Passwords.txt` ファイル。

   e. SSH エージェントに SSH 秘密鍵を追加します。入力するコマンド `ssh-add`

   f. に記載されているSSHアクセスパスワードを入力します `Passwords.txt` ファイル。

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 管理ノードから、Prometheusサービスを停止します。 `service prometheus stop`

   a. 一時的なバックアップ場所から管理ノードにPrometheusデータベースをコピーします。 `/rsync -azh --stats "backup_node:/var/local/tmp/prometheus/" "/var/local/mysql_ibdata/prometheus/"`

   b. データが正しいパスにあり、完全であることを確認します `ls /var/local/mysql_ibdata/prometheus/data/`

3. ソース管理ノードでPrometheusサービスを再起動します.`service prometheus start`

履歴情報をリストアします

1. 管理ノードにログインします。

a. 次のコマンドを入力します。 `ssh admin@grid_node_IP`

b. に記載されているパスワードを入力します `Passwords.txt` ファイル。

c. 次のコマンドを入力してrootに切り替えます。 `su -`

d. に記載されているパスワードを入力します `Passwords.txt` ファイル。

e. SSH エージェントに SSH 秘密鍵を追加します。入力するコマンド `ssh-add`

f. に記載されているSSHアクセスパスワードを入力します `Passwords.txt` ファイル。

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 代替ノードからMySQLダンプファイルをコピーします。 `scp grid_node_IP_:/var/local/tmp/mysql-mi.sql /var/local/tmp/mysql-mi.sql`
3. 管理ノードでStorageGRID サービスを停止し、NTPとMySQLを起動します
   a. すべてのサービスを停止します。 `service servermanager stop`
   b. NTPサービスを再開します。 `service ntp start`.. MySQLサービスを再起動します。 `service mysql start`
4. miデータベースを削除し、新しい空のデータベースを作成します。 `mysql -u username -p password -A mi -e "drop database mi; create database mi;"`
5. データベースダンプからMySQLデータベースをリストアします。 `mysql -u username -p password -A mi < /var/local/tmp/mysql-mi.sql`
6. 他のすべてのサービスを再起動します `service servermanager start`

アロンクライン著

グリッドサイトの再配置とサイト全体のネットワーク変更手順

このガイドでは、マルチサイトグリッドでのStorageGRIDサイトの再配置の準備と手順について説明します。この手順を完全に理解し、スムーズなプロセスを実現し、クライアントの中断を最小限に抑えるために事前に計画しておく必要があります。

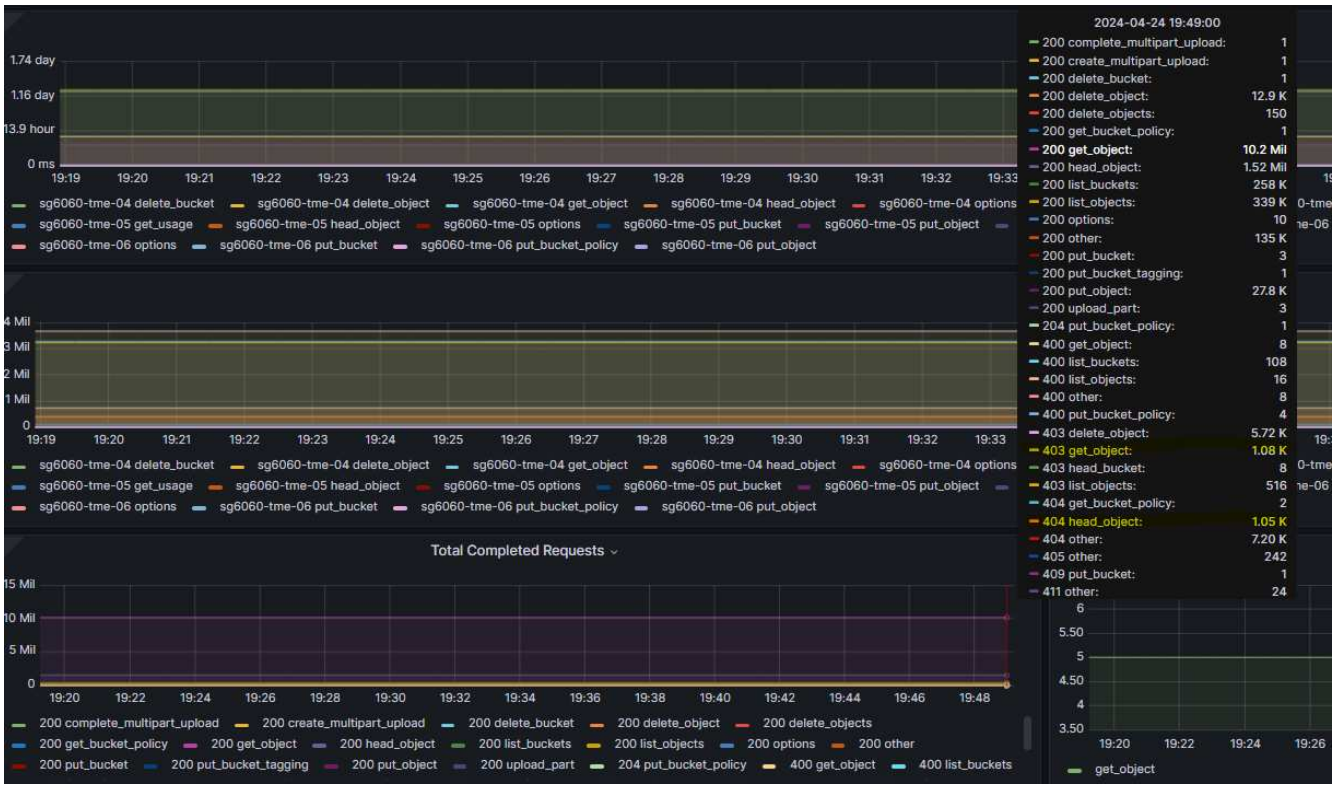
グリッド全体のグリッドネットワークを変更する必要がある場合は、を参照してください。
"グリッド内のすべてのノードの IP アドレスを変更します"。

サイトの再配置前の考慮事項

- Cassandraデータベースの再構築を回避するには、サイトの移動を完了し、すべてのノードを15日以内にオンラインにします。
  "ストレージノードを 15 日以上停止した状態にリカバリします"
- アクティブポリシー内のいずれかのILMルールで厳密な取り込み動作が使用されている場合は、サイトの再配置中にオブジェクトを引き続きグリッドに配置する必要がある場合は、負荷分散またはデュアルコミットに変更することを検討してください。
- ストレージアプライアンスに60本以上のドライブが搭載されている場合は、ディスクドライブが取り付け

られているシェルフを移動しないでください。 パック/移動の前に、各ディスクドライブにラベルを付け、ストレージエンクロージャから取り外します。

- StorageGRIDアプライアンスの変更グリッドネットワークVLANは、管理ネットワークまたはクライアントネットワーク経由でリモートで実行できます。 または、勤務地変更の前後にオンサイトで変更を実施する予定です。
- PUTの前に、お客様のアプリケーションがHEADを使用しているか、存在しないオブジェクトを取得しているかを確認「はい」の場合は、HTTP 500エラーを回避するためにバケットの整合性をstrong-siteに変更します。 不明な場合は、S3の概要Grafanaグラフ*[Grid manager]>[Support]>[Metrics]*を確認し、[Total Completed Request]グラフにカーソルを合わせます。 404 GET Objectまたは404 HEADオブジェクトの数が非常に多い場合は、1つ以上のアプリケーションがHEADまたはGET Non-existenceオブジェクトを使用している可能性があります。カウントは累積値です。異なるタイムライン上にマウスを移動すると、その差が表示されます。



サイトの再配置前に手順で**Grid IP**アドレスを変更

手順

1. 新しいグリッドネットワークサブネットが新しい場所で使用される場合は、"グリッドネットワークサブネットリストにサブネットを追加します。"
2. プライマリ管理ノードにログインし、change-ipを使用してグリッドIPを変更します。再配置用にノードをシャットダウンする前に、変更をステージングする必要があります*。
   a. [Grid IP]で[2]、[1]を選択します。

```
Editing: Node IP/subnet and gateway

Use up arrow to recall a previously typed value, which you can then edit
Use d or 0.0.0.0/0 as the IP/mask to delete the network from the node
Use q to complete the editing session early and return to the previous menu
Press <enter> to use the value shown in square brackets

================================================================================
Site: LONDON
================================================================================
LONDON-ADM1 Grid   IP/mask [     10.45.74.14/26 ]: 10.45.74.24/26
LONDON-S1   Grid   IP/mask [     10.45.74.16/26 ]: 10.45.74.26/26
LONDON-S2   Grid   IP/mask [     10.45.74.17/26 ]: 10.45.74.27/26
LONDON-S3   Grid   IP/mask [     10.45.74.18/26 ]: 10.45.74.28/26
================================================================================
LONDON-ADM1 Grid   Gateway [         10.45.74.1 ]:
LONDON-S1   Grid   Gateway [         10.45.74.1 ]:
LONDON-S2   Grid   Gateway [         10.45.74.1 ]:
LONDON-S3   Grid   Gateway [         10.45.74.1 ]:
================================================================================
================================================================================
Site: OXFORD
================================================================================
OXFORD-ADM1 Grid   IP/mask [     10.45.75.14/26 ]:
OXFORD-S1   Grid   IP/mask [     10.45.75.16/26 ]:
OXFORD-S2   Grid   IP/mask [     10.45.75.17/26 ]:
OXFORD-S3   Grid   IP/mask [     10.45.75.18/26 ]:
================================================================================
OXFORD-ADM1 Grid   Gateway [         10.45.75.1 ]:
OXFORD-S1   Grid   Gateway [         10.45.75.1 ]:
OXFORD-S2   Grid   Gateway [         10.45.75.1 ]:
OXFORD-S3   Grid   Gateway [         10.45.75.1 ]:
================================================================================
Finished editing. Press Enter to return to menu.
```

b. 5を選択して変更を表示

```
================================================================================
Site: LONDON
================================================================================
LONDON-ADM1 Grid   IP      [     10.45.74.14/26 ]: 10.45.74.24/26
LONDON-S1   Grid   IP      [     10.45.74.16/26 ]: 10.45.74.26/26
LONDON-S2   Grid   IP      [     10.45.74.17/26 ]: 10.45.74.27/26
LONDON-S3   Grid   IP      [     10.45.74.18/26 ]: 10.45.74.28/26
Press Enter to continue
```

c. [10]を選択して確定し、変更を適用します。

```
Welcome to the StorageGRID IP Change Tool.

Selected nodes: all

1:   SELECT NODES to edit
2:   EDIT IP/mask and gateway
3:   EDIT admin network subnet lists
4:   EDIT grid network subnet list
5:   SHOW changes
6:   SHOW full configuration, with changes highlighted
7:   VALIDATE changes
8:   SAVE changes, so you can resume later
9:   CLEAR all changes, to start fresh
10:  APPLY changes to the grid
0:   Exit

Selection: 10
```

d. このステップで* stage *を選択する必要があります。

```
Validating new networking configuration... PASSED.
Checking for Grid Network IP address swaps... PASSED.

Applying these changes will update the following nodes:

  LONDON-ADM1
  LONDON-S1
  LONDON-S2
  LONDON-S3

The following nodes will also require restarting:

  LONDON-ADM1
  LONDON-S1
  LONDON-S2
  LONDON-S3

Select one of the following options:

  apply:  apply all changes and automatically restart nodes (if necessary)
  stage:  stage the changes; no changes will take effect until the nodes are restarted
  cancel: do not make any network changes at this time

[apply/stage/cancel]> stage
```

e. 上記の変更にプライマリ管理ノードが含まれている場合は、「**a**」と入力して手動でプライマリ管理ノードを再起動します

```
10.45.74.14 - PuTTY
Validating new networking configuration... PASSED.
Checking for Grid Network IP address swaps... PASSED.

Applying these changes will update the following nodes:

  LONDON-ADM1
  LONDON-S1
  LONDON-S2
  LONDON-S3

The following nodes will also require restarting:

  LONDON-ADM1
  LONDON-S1
  LONDON-S2
  LONDON-S3

Select one of the following options:

  apply:  apply all changes and automatically restart nodes (if necessary)
  stage:  stage the changes; no changes will take effect until the nodes are restarted
  cancel: do not make any network changes at this time

[apply/stage/cancel]> stage

Generating new grid networking description file... PASSED.
Running provisioning... PASSED.
Updating network configuration on LONDON-S1... PASSED.
Updating network configuration on LONDON-S2... PASSED.
Updating network configuration on LONDON-S3... PASSED.
Updating network configuration on LONDON-ADM1... PASSED.
Finished staging network changes. You must manually restart these nodes for the changes to take effect:

  LONDON-ADM1 (has IP 10.45.74.14 until restart)
  LONDON-S1 (has IP 10.45.74.16 until restart)
  LONDON-S2 (has IP 10.45.74.17 until restart)
  LONDON-S3 (has IP 10.45.74.18 until restart)

Importing bundles... PASSED.
*****************************************************************
*                          IMPORTANT                            *
*                                                               *
*  A new recovery package has been generated as a result of the *
*  configuration change. Select Maintenance > Recovery Package  *
*  in the Grid Manager to download it.                          *
*****************************************************************

Network Update Complete.  Primary admin restart required. Select 'continue' to restart this node immediately, 'abort' to restart manually.
Enter a to abort, c to continue [a/c]>
```

f. Enterキーを押して前のメニューに戻り、IPインターフェイスの変更を終了します。

```
Network Update Complete.  Primary admin restart required. Select 'continue' to restart this node immediately, 'abort' to restart manually.
Enter a to abort, c to continue [a/c]> a
Restart aborted. You must manually restart this node as soon as possible
Press Enter to return to the previous menu.
```

3. Grid Managerから、新しいリカバリパッケージをダウンロードします。* Grid Manager **>***メンテナンス>* リカバリパッケージ*
4. StorageGRIDアプライアンスでVLANの変更が必要な場合は、を参照してください。 アプライアンスVLANの変更。
5. サイトのすべてのノードおよびアプライアンスをシャットダウンし、必要に応じてディスクドライブにラベルを付けて取り外し、ラックを開梱して梱包して移動します。
6. 管理ネットワークのIP、クライアントのVLAN、IPアドレスを変更する場合は、再配置後に変更を実行できます。

アプライアンス**VLAN**の変更

以下の手順は、リモートから変更を実行するために、StorageGRIDアプライアンスの管理ネットワークまたはクライアントネットワークにリモートアクセスできることを前提としています。

手順

1. アプライアンスをシャットダウンする前に、 "アプライアンスをメンテナンスモードにします"。

2. ブラウザを使用したStorageGRIDアプライアンスインストーラGUIへのアクセス https://<admin-or-client-network-ip>:8443。 アプライアンスをメンテナンスモードでブートすると、すでに使用されている新しいグリッドIPとしてグリッドIPを使用することはできません。
3. グリッドネットワークのVLANを変更します。 クライアント・ネットワーク経由でアプライアンスにアクセスする場合、現時点ではクライアントVLANは変更できません。移動後に変更できます。
4. アプライアンスにSSH接続し、「shutdown -h now」を使用してノードをシャットダウン
5. 新しいサイトでアプライアンスの準備が完了したら、を使用してStorageGRIDアプライアンスインストーラのGUIにアクセスします。 https://<grid-network-ip>:8443。 GUIでping / nmapツールを使用して、ストレージが最適な状態であり、他のグリッドノードへのネットワーク接続が確立されていることを確認します。
6. クライアントネットワークIPの変更を計画している場合は、この段階でクライアントVLANを変更できます。 クライアントネットワークは、このあとの手順でIP変更ツールを使用してクライアントネットワークIPを更新するまで準備ができていません。
7. メンテナンスモードを終了します。StorageGRID アプライアンス・インストーラから、 **Advanced**>* Reboot Controller* を選択し、 * Reboot into StorageGRID * を選択します。
8. すべてのノードが稼働し、[Grid]に接続問題が表示されなくなったら、必要に応じてchange-IPを使用してアプライアンスの管理ネットワークとクライアントネットワークを更新します。

ツールおよびアプリケーションガイド

StorageGRID でCloudera Hadoop S3Aコネクタを使用します

Hadoopは、しばらくの間データサイエンティストのお気に入りでした。Hadoopでは、シンプルなプログラミングフレームワークを使用して、複数のコンピュータクラスタにまたがる大規模なデータセットを分散処理できます。Hadoopは、ローカルのコンピューティングとストレージを所有するマシンごとに、単一のサーバから数千のマシンにスケールアップするように設計されています。

S3AをHadoopワークフローに使用する理由

データ量の増加に伴い、新しいマシンにコンピューティングとストレージを個別に追加するアプローチは非効率的になっています。リニアに拡張すると、リソースの効率的な使用やインフラの管理が難しくなります。

このような課題に対処するために、Hadoop S3AクライアントはS3オブジェクトストレージに対する高性能なI/Oを提供します。S3Aを使用してHadoopワークフローを実装することで、オブジェクトストレージをデータリポジトリとして活用でき、コンピューティングとストレージを分離することができます。これにより、コンピューティングとストレージを別々に拡張できます。コンピューティングリソースとストレージを分離することで、コンピューティングジョブに適切な量のリソースを割り当て、データセットのサイズに基づいて容量を提供することもできます。そのため、Hadoopワークフローの総所有コストを削減することができます。

StorageGRID を使用するようにS3Aコネクタを構成します

前提条件

- StorageGRID S3エンドポイントのURL、テナントS3アクセスキー、およびHadoop S3A接続テスト用のシークレットキー。
- クラスタ内の各ホストに対するCloudera クラスタとrootまたはsudo権限を付与して、Javaパッケージをインストールします。

2022年4月時点で、StorageGRID 11.0.14とCloudera 7.1.7のJava 11.0.14が、11.5および11.6に対してテストされました。ただし、Javaのバージョン番号は新規インストール時と異なる場合があります。

Javaパッケージをインストールします

1. を確認します "Clouderaサポートマトリックス" を参照してください。
2. をダウンロードします "Java 11.xパッケージ" Clouderaクラスタオペレーティングシステムと同じです。このパッケージをクラスタ内の各ホストにコピーします。この例では、CentOSにrpmパッケージを使用しています。
3. 各ホストにrootとしてログインするか、sudo権限を持つアカウントを使ってログインします。各ホストで次の手順を実行します。
   a. パッケージをインストールします。

```
$ sudo rpm -Uvh jdk-11.0.14_linux-x64_bin.rpm
```

b. Javaがインストールされている場所を確認します。複数のバージョンがインストールされている場合は、新しくインストールしたバージョンをデフォルトに設定します。

```
alternatives --config java

There are 2 programs which provide 'java'.

  Selection    Command
-----------------------------------------------
 +1           /usr/java/jre1.8.0_291-amd64/bin/java
  2           /usr/java/jdk-11.0.14/bin/java

Enter to keep the current selection[+], or type selection number: 2
```

c. この行を/etc/profile’の末尾に追加しますパスは、上記の選択のパスと一致する必要があります。

```
export JAVA_HOME=/usr/java/jdk-11.0.14
```

d. 次のコマンドを実行して、プロファイルを有効にします。

```
source /etc/profile
```

Cloudera HDFS S3A構成

- 手順 *

1. Cloudera Manager GUIで、クラスタ（Clusters）> HDFSを選択し、構成（Configuration）を選択します。
2. カテゴリでAdvancedを選択し、下にスクロールして「core-site .xml」用のクラスタ全体のAdvanced Configuration Snippet（Safety Valve）を探します。
3. （+）記号をクリックし、次の値ペアを追加します。

名前	価値
fs.s3a.access.key	*<tenant StorageGRID のs3アクセスキー_*
fs.s3a.secret.key	*<tenant s3 secret key from StorageGRID >*
FS.s3a.connection.ssl.enabled	[true or false]（このエントリがない場合のデフォルトはhttps）
FS.s3a.endpoint	_ StorageGRID S3エンドポイント：port>_

名前	価値
FS.s3a.impl	org.apache.hadoop .fs.s3a.S3AFileSystem
FS.s3a.path.style.access	[true or false]（このエントリがない場合のデフォルトの仮想ホスト形式）

サンプルスクリーンショット

Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml
core_site_safety_valve

HDFS (Service-Wide) Undo

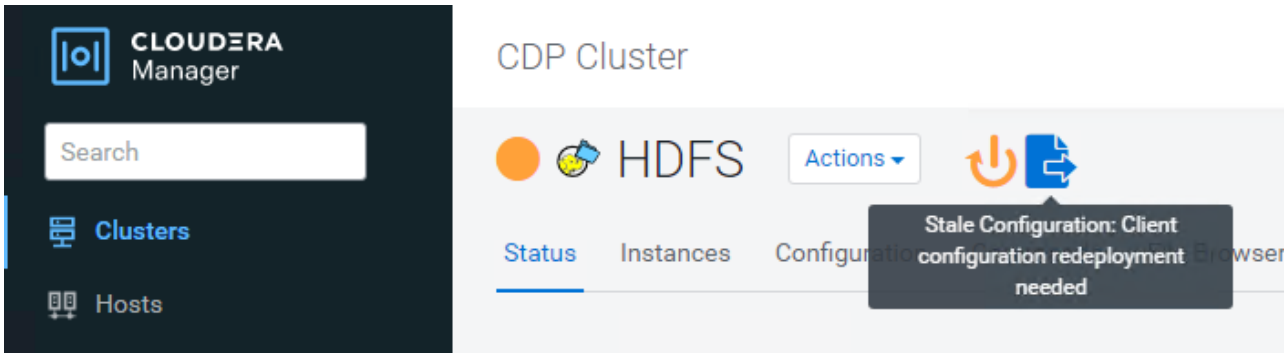
View as XML

Name	fs.s3a.endpoint
Value	sgdemo.netapp.com:10443
Description	StorageGRID s3 load balancer endpoint
	☑ Final
Name	fs.s3a.access.key
Value	OM[redacted]BAN
Description	SG CDP S3 access key
	☑ Final
Name	fs.s3a.secret.key
Value	mapz[redacted]Qfc
Description	SG CDP S3 secret key
	☑ Final
Name	fs.s3a.impl
Value	org.apache.hadoop.fs.s3a.S3AFileSystem
Description	
	☑ Final
Name	fs.s3a.path.style.access
Value	true
Description	
	☑ Final

Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml

Save Changes(CTRL+S)

1. [Save Changes]ボタンをクリックします。HDFSメニューバーからStale Configurationアイコンを選択し、次のページでRestart Stale Servicesを選択して、Restart Nowを選択します。



StorageGRID へのS3A接続をテストします

基本的な接続テストを実行します

Clouderaクラスタのいずれかのホストにログインし、「hadoop fs s-ls s3a：//<*bucket-name*>/`」と入力します。

次の例では、パスsyleと既存のHDFSテストバケットおよびテストオブジェクトを使用します。

```
[root@ce-n1 ~]# hadoop fs -ls s3a://hdfs-test/
22/02/15 18:24:37 WARN impl.MetricsConfig: Cannot locate configuration:
tried hadoop-metrics2-s3a-file-system.properties,hadoop-
metrics2.properties
22/02/15 18:24:37 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot
period at 10 second(s).
22/02/15 18:24:37 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system started
22/02/15 18:24:37 INFO Configuration.deprecation: No unit for
fs.s3a.connection.request.timeout(0) assuming SECONDS
Found 1 items
-rw-rw-rw-   1 root root       1679 2022-02-14 16:03 s3a://hdfs-test/test
22/02/15 18:24:38 INFO impl.MetricsSystemImpl: Stopping s3a-file-system
metrics system...
22/02/15 18:24:38 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system stopped.
22/02/15 18:24:38 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system shutdown complete.
```

トラブルシューティング

シナリオ **1**

StorageGRID へのHTTPS接続を使用し、15分後に「handshake_failure」エラーを取得します。

*理由：StorageGRID への接続に古いTLS暗号スイートまたはサポートされていないTLS暗号スイートを使用しているJRE／JDKの旧バージョン。

エラーメッセージの例

```
[root@ce-n1 ~]# hadoop fs -ls s3a://hdfs-test/
22/02/15 18:52:34 WARN impl.MetricsConfig: Cannot locate configuration:
tried hadoop-metrics2-s3a-file-system.properties,hadoop-
metrics2.properties
22/02/15 18:52:34 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot
period at 10 second(s).
22/02/15 18:52:34 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system started
22/02/15 18:52:35 INFO Configuration.deprecation: No unit for
fs.s3a.connection.request.timeout(0) assuming SECONDS
22/02/15 19:04:51 INFO impl.MetricsSystemImpl: Stopping s3a-file-system
metrics system...
22/02/15 19:04:51 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system stopped.
22/02/15 19:04:51 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system shutdown complete.
22/02/15 19:04:51 WARN fs.FileSystem: Failed to initialize fileystem
s3a://hdfs-test/: org.apache.hadoop.fs.s3a.AWSClientIOException:
doesBucketExistV2 on hdfs: com.amazonaws.SdkClientException: Unable to
execute HTTP request: Received fatal alert: handshake_failure: Unable to
execute HTTP request: Received fatal alert: handshake_failure
ls: doesBucketExistV2 on hdfs: com.amazonaws.SdkClientException: Unable to
execute HTTP request: Received fatal alert: handshake_failure: Unable to
execute HTTP request: Received fatal alert: handshake_failure
```

*解決策: JDK 11.x以降がインストールされていることを確認し'デフォルトのJavaライブラリに設定しますを参照してください Javaパッケージをインストールします 詳細については、を参照してください。

シナリオ**2**：

StorageGRID に接続できませんでした。エラーメッセージ「要求されたターゲットへの有効な証明書パスが見つかりませんでした」が表示されます。

理由： StorageGRID S3エンドポイントサーバ証明書がJavaプログラムで信頼されていません。

エラーメッセージの例：

```
[root@hdp6 ~]# hadoop fs -ls s3a://hdfs-test/
22/03/11 20:58:12 WARN impl.MetricsConfig: Cannot locate configuration:
tried hadoop-metrics2-s3a-file-system.properties,hadoop-
metrics2.properties
22/03/11 20:58:13 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot
period at 10 second(s).
22/03/11 20:58:13 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system started
22/03/11 20:58:13 INFO Configuration.deprecation: No unit for
fs.s3a.connection.request.timeout(0) assuming SECONDS
22/03/11 21:12:25 INFO impl.MetricsSystemImpl: Stopping s3a-file-system
metrics system...
22/03/11 21:12:25 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system stopped.
22/03/11 21:12:25 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system shutdown complete.
22/03/11 21:12:25 WARN fs.FileSystem: Failed to initialize fileystem
s3a://hdfs-test/: org.apache.hadoop.fs.s3a.AWSClientIOException:
doesBucketExistV2 on hdfs: com.amazonaws.SdkClientException: Unable to
execute HTTP request: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target: Unable to execute HTTP
request: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target
```

*解決策：ネットアップは、既知のパブリック証明書署名機関が発行するサーバ証明書を使用して、認証がセキュアであることを確認することを推奨しています。または、Javaの信頼ストアにカスタムのCA証明書またはサーバ証明書を追加します。

StorageGRID カスタムCA証明書またはサーバ証明書をJava信頼ストアに追加するには、次の手順を実行します。

1. 既存のデフォルトのJava cacertsファイルをバックアップします。

```
cp -ap $JAVA_HOME/lib/security/cacerts
$JAVA_HOME/lib/security/cacerts.orig
```

2. StorageGRID S3エンドポイント証明書をJava信頼ストアにインポートします。

```
keytool -import -trustcacerts -keystore $JAVA_HOME/lib/security/cacerts
-storepass changeit -noprompt -alias sg-lb -file <StorageGRID CA or
server cert in pem format>
```

トラブルシューティングのヒント

1. Hadoopログレベルを引き上げてデバッグします。

   'export hadoop root_logger = hadoop .root.logger = debug、console'

2. コマンドを実行し、ログメッセージをerror.logに送信します。

   「hadoop fs s-ls s3a：//<*bucket-name*>_ error.log

Angela Cheng著_

S3cmdを使用して、StorageGRID でS3アクセスをテストおよび実証します

S3cmdは、S3処理用の無償のコマンドラインツールおよびクライアントです。s3cmdを使用して、StorageGRID でのS3アクセスをテストして実証できます。

S3cmdをインストールして構成します

ワークステーションまたはサーバにS3cmdをインストールするには、からダウンロードします "コマンドラインS3クライアント"。s3cmdは、トラブルシューティング用のツールとして、各StorageGRID ノードにあらかじめインストールされています。

初期設定手順

1. s3cmd --設定
2. 残りのキーには、access-keyとsecret_keyだけを指定してデフォルトのままにします。
3. 指定したクレデンシャルでアクセスをテストします[Y/n]：n（失敗するため、テストをバイパスする）
4. 設定を保存しますか？[y/N] y
   a. 設定を「/root/.s3cfg」に保存しました。
5. s3cfgで、「=」記号のあとにhost_baseフィールドとhost_bucketフィールドを空にします。
   a. host_base=
   b. host_bucket=

手順4でhost_baseとhost_bucketを指定した場合は、CLIで—hostのエンドポイントを指定する必要はありません。例

```
host_base = 192.168.1.91:8082
host_bucket = bucketX.192.168.1.91:8082
s3cmd ls s3://bucketX --no-check-certificate
```

基本的なコマンドの例

- バケットを作成：

  s3cmd mb s3：//s3cmdbucket --host=<endpoint>：<port>--no-check-certificate-`

- すべてのバケットを表示：

  s3cmd ls --host=<endpoint>：<port>--no-check-certificate'

- すべてのバケットとその内容を表示：

  s3cmd la --host=<endpoint>：<port>-- no-check-certificate'

- 特定のバケット内のオブジェクトをリストします。

  s3cmd ls s3：//<bucket>--host=<endpoint>：<port>--no-check-certificate-`

- バケットを削除：

  s3cmd rb s3：//s3cmdbucket --host=<endpoint>：<port>--no-check-certificate'

- オブジェクトを置きなさい:

  s3cmd put <file>s3://<bucket>--host=<endpoint>:<port>--no-check-certificate`

- オブジェクトを取得：

  s3cmd get s3：//<バケット>/<オブジェクト><ファイル>--host=<endpoint>：<port>--no-check-certificate'

- オブジェクトを削除：

  s3cmd del s3：//<bucket>/<object>--host=<endpoint>：<port>：-no-check-certificate-`

アロンクライン著

NetApp StorageGRID を共有ストレージとして使用したVertica Eonモードのデータベース

このガイドでは、NetApp StorageGRID のパブリックストレージを使用してVertica Eon Modeデータベースを作成する手順 について説明します。

はじめに

Verticaは分析データベース管理ソフトウェアです。大量のデータを処理するように設計されたカラム型ストレージ・プラットフォームであり'従来の負荷の高いシナリオでは非常に高速なクエリー・パフォーマンスを実現しますVerticaデータベースは、EonまたはEnterpriseのいずれかのモードで動作します。両方のモードをオンプレミスまたはクラウドに導入できます。

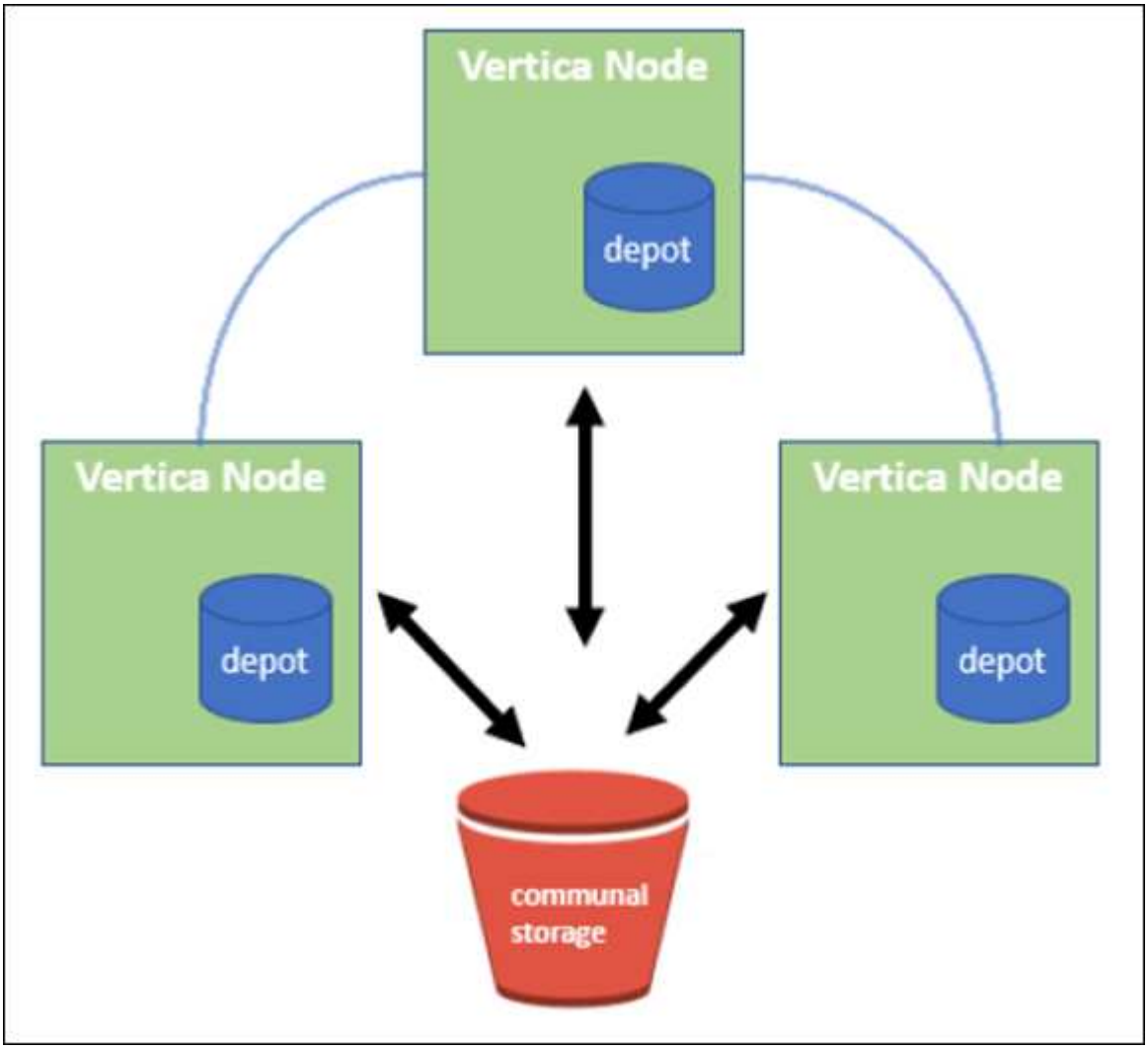
EonモードとEnterpriseモードは、主にデータの保存場所によって異なります。

- Eonモードのデータベースは、データ用に共有ストレージを使用します。これはVerticaがお勧めします。
- Enterprise Modeデータベースでは、データベースを構成するノードのファイルシステムにデータがローカルに格納されます。

Eon Modeアーキテクチャ

Eonモードでは、計算リソースがデータベースの共有ストレージレイヤから分離され、コンピューティングとストレージを別々に拡張できます。EonモードのVerticaは、さまざまなワークロードに対応し、コンピューティングリソースとストレージリソースを別々に使用してワークロードを分離するように最適化されています。

Eon Modeは、パブリックストレージと呼ばれる共有オブジェクトストアにデータを格納します。パブリックストレージとは、オンプレミスまたはAmazon S3上にホストされるS3バケットです。



共有ストレージ

Eonモードでは、データをローカルに格納する代わりに、すべてのデータとカタログ（メタデータ）に単一の共有ストレージロケーションを使用します。共有ストレージとは、データベースの一元管理されたストレージの場所で、データベースノード間で共有されるものです。

共有ストレージには次のプロパティがあります。

- クラウドまたはオンプレミスのオブジェクトストレージ内の共有ストレージは、個々のマシンのディスク上のストレージよりも耐障害性が高く、ストレージ障害によるデータ損失の影響を受けにくくなっています。
- すべてのデータは、同じパスを使用して任意のノードで読み取ることができます。
- ノードのディスクスペースによる容量制限はありません。
- データは通信環境に保管されるため、変化するニーズに合わせてクラスタを柔軟に拡張できます。データがノードにローカルに格納されていた場合は、ノードを追加または削除するときに、ノード間で移動するデータが大量に必要になります。これを行うには、削除対象のノードから移動するか、新しく作成したノードに移動する必要があります。

デポ

共有ストレージの欠点の1つは速度です。共有クラウド上の場所からデータにアクセスする場合、ローカルディスクからデータを読み取る場合よりも時間がかかります。また、多数のノードが一度にデータを読み取っている場合、共有ストレージへの接続がボトルネックになる可能性があります。データアクセス速度を向上させるために、Eon Modeデータベース内のノードは、デポと呼ばれるデータのローカルディスクキャッシュを保持します。クエリを実行するとき、ノードはまず、必要なデータがデポにあるかどうかをチェックします。存在する場合は、データのローカルコピーを使用してクエリが完了します。データがデポにない場合、ノードは共有ストレージからデータを取得し、デポにコピーを保存します。

NetApp StorageGRID の推奨事項

Verticaは、データベースのデータをオブジェクトストレージに何千（数百万）もの圧縮オブジェクトとして格納します（1オブジェクトあたり200～500MB）。ユーザーがデータベースクエリを実行すると、Verticaはバイト範囲GET呼び出しを使用して、圧縮されたオブジェクトから選択したデータ範囲を並列に取得します。バイト範囲GETはそれぞれ約8KBです。

10TBのデータベースデポのユーザクエリテストでは、1秒あたり4、000~ 10、000個のGET（バイト範囲GET）要求がグリッドに送信されました。SG6060アプライアンスを使用してこのテストを実行した場合、アプライアンスノードあたりのCPU利用率は（2030%程度）が低いため、CPU時間の2/3でI/Oを待機しているSGF6024では、I/O待機時間のごく一部（0% 0.5%）が確認されます。

IOPSは小さいが高いことから、レイテンシの要件は非常に低い（平均値は0.01秒未満）ため、オブジェクトストレージサービスにはSFG6024を使用することを推奨します。非常に大きなデータベースサイズにSG6060が必要な場合は、お客様はデポサイジングのVerticaアカウントチームと協力して、照会中のデータセットをサポートする必要があります。

管理ノードとAPIゲートウェイノードの場合は、お客様がSG100またはSG1000を使用できます。選択する内容は、ユーザのクエリ要求の並列サイズとデータベースサイズによって異なります。他社製ロードバランサを使用する場合は、ハイパフォーマンスが要求されるワークロードに専用のロードバランサを使用することを推奨します。StorageGRID のサイジングについては、ネットアップアカウントチームにお問い合わせください。

StorageGRID 構成に関するその他の推奨事項は次のとおりです。

- グリッドトポロジ。同じグリッドサイトにある他のストレージアプライアンスモデルとSGF6024を混在させないでください。長期アーカイブ保護にSG6060を使用する場合は、アクティブデータベース用に専用のグリッドロードバランサを使用してSGF6024の負荷を専用のグリッドサイト（物理サイトまたは論理サイト）に配置し、パフォーマンスを向上させます。同じサイトに異なるモデルのアプライアンスを混在させると、サイト全体のパフォーマンスが低下します。
- データ保護。レプリケートコピーを使用して保護します。アクティブデータベース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。イレイジャーコーディングを使用することで、アクセス頻度の低いデータベースを長期にわたって保護できます。

- グリッド圧縮を有効にしないでください。Verticaは、オブジェクトを圧縮してからオブジェクトストレージに格納します。グリッド圧縮を有効にしてもストレージ使用量はこれ以上削減されず、バイト範囲のGETパフォーマンスが大幅に低下します。
- * HTTPとHTTPS S3エンドポイント接続*。ベンチマークテストでは、VerticaクラスタからStorageGRIDロードバランサエンドポイントへのHTTP S3接続を使用した場合、パフォーマンスが約5%向上しました。この選択は、顧客のセキュリティ要件に基づいて行う必要があります。

Vertica構成に関する推奨事項は次のとおりです。

- * Verticaデータベースのデフォルトデポ設定は、読み取りおよび書き込み操作で有効(値=1)になっています。*パフォーマンスを向上させるために、これらのデポ設定を有効にしておくことを強く推奨します。
- *ストリーミング制限を無効にします。*設定の詳細については、を参照してください ストリーミング制限を無効にしています。

StorageGRID 上の共有ストレージを使用してオンプレミスモードをインストールする

以下のセクションでは、StorageGRID 上に共同ストレージを使用してオンプレミスにEonモードをインストールするための手順 について説明します。オンプレミスのSimple Storage Service（S3）互換オブジェクトストレージを設定する手順 は、Vertica guideの手順 に似ています。 "オンプレミスにEonモードデータベースをインストールします"。

機能テストには次のセットアップを使用しました。

- StorageGRID 11.4.0.4
- Vertica 10.1.0
- Verticaノードをクラスタに構成するために、CentOS 7.x OSを搭載した3台の仮想マシン（VM）。このセットアップは、Verticaプロダクションデータベースクラスタではなく、機能テストのみを対象としています。

これらの3つのノードにはSecure Shell（SSH）キーが設定されており、クラスタ内のノード間でパスワードを設定することなくSSHを使用できます。

NetApp StorageGRID で必要な情報

StorageGRID 上で共有ストレージを使用してオンプレミスにEonモードをインストールするには、次の前提条件情報が必要です。

- StorageGRID S3エンドポイントのIPアドレスまたは完全修飾ドメイン名（FQDN）とポート番号。HTTPSを使用する場合は、StorageGRID S3エンドポイントに実装されているカスタムの認証局（CA）または自己署名SSL証明書を使用します。
- バケット名。このパラメータは、あらかじめ存在し、空である必要があります。
- バケットへの読み取り/書き込みアクセスが可能なアクセスキーIDとシークレットアクセスキー。

S3エンドポイントにアクセスするための認証ファイルを作成します

S3エンドポイントにアクセスする許可ファイルを作成する際には、次の前提条件が適用されます。

- Verticaがインストールされている。
- クラスタをセットアップして設定し、データベースを作成できる状態にします。

S3エンドポイントにアクセスするための認証ファイルを作成するには、次の手順を実行します。

1. 「admintools」を実行してEon Modeデータベースを作成するVerticaノードにログインします。

   デフォルトのユーザーは'dbadmin'で'Verticaクラスタのインストール時に作成されます

2. テキスト・エディタを使用して'/HOME/dbadminディレクトリの下にファイルを作成しますファイル名には'たとえばsg_auth.confなど'任意の名前を指定できます
3. S3エンドポイントが標準のHTTPポート80またはHTTPSポート443を使用している場合は、ポート番号を省略します。HTTPSを使用するには、次の値を設定します。

   - `awsenablehttps=1'それ以外の場合は'0`に値を設定します
   - `awsauth=<s3 access key ID>：<secret access key>`
   - `awsendpoint=< StorageGRID s3 endpoint>:<port>`

     StorageGRID S3エンドポイントのHTTPS接続にカスタムCA証明書または自己署名SSL証明書を使用するには、証明書の完全なファイルパスとファイル名を指定します。このファイルは、各Verticaノード上の同じ場所にあり、すべてのユーザーに読み取り権限が与えられている必要があります。StorageGRID S3エンドポイントのSSL証明書が一般に知られているCAによって署名されている場合は、この手順を省略します。

     -awscafile=<filepath/filename>`

     たとえば、次のサンプルファイルを参照してください。

```
awsauth = MNVU4OYFAY2xyz123:03vuO4M4KmdfwffT8nqnBmnMVTr78Gu9wANabcxyz
awsendpoint = s3.england.connectlab.io:10443
awsenablehttps = 1
awscafile = /etc/custom-cert/grid.pem
```

+

> 本番環境では、一般に知られているCAによって署名されたサーバ証明書をStorageGRID S3ロードバランサエンドポイントに実装する必要があります。

すべての**Vertica**ノードのデポパスを選択します

デポストレージパスの各ノードにディレクトリを選択または作成します。デポストレージパスパラメータに指定するディレクトリには、次のものが必要です。

- クラスタ内のすべてのノードで同じパス（例：/home/dbadmin/depot）
- dbadminユーザによる読み書きが可能になります
- 十分なストレージ

デフォルトでは、Verticaはデポ保存用のディレクトリを含むファイルシステム領域の60%を使用します。'create-db'コマンドの—depot-size'引数を使用すると、デポのサイズを制限できます。を参照してください "EonモードデータベースのVertica Clusterのサイジング" Verticaの一般的なサイジングガイドラインについては、こちらをご覧ください。Vertica Account Managerにお問い合わせください。

'admintools create-db'ツールは'存在しない場合に備えて'デポパスを作成しようとします

オンプレミスデータベースの作成

オンプレミスデータベースを作成するには、次の手順を実行します。

1. データベースを作成するには'admintools create-db'ツールを使用します

   この例で使用されている引数の簡単な説明を次に示します。すべての必須引数とオプション引数の詳細については、Verticaのドキュメントを参照してください。

   - -x <で作成された認証ファイルのパス/ファイル名 「S3エンドポイントにアクセスするための認証ファイルの作成」 >。

     認証の詳細は、正常に作成された後、データベース内に保存されます。S3シークレットキーの公開を回避するために、このファイルを削除できます。

   - --son/storagegrid-sstorage -location <s3：//storagegrid bucketname>
   - -s <このデータベースに使用するVerticaノードのカンマ区切りリスト>
   - -d <作成するデータベースの名前>
   - -p <この新しいデータベースに設定するパスワード>。たとえば、次のコマンド例を参照してください。

     ```
     admintools -t create_db -x sg_auth.conf --communal-storage
     -location=s3://vertica --depot-path=/home/dbadmin/depot --shard
     -count=6 -s vertica-vm1,vertica-vm2,vertica-vm3 -d vmart -p
     '<password>'
     ```

     データベースのノード数によっては、新しいデータベースの作成に数分かかることがあります。データベースを初めて作成するときに、ライセンス契約に同意するように求められます。

たとえば'次のサンプル認証ファイルと'create db'コマンドを参照してください

```
[dbadmin@vertica-vm1 ~]$ cat sg_auth.conf
awsauth = MNVU4OYFAY2CPKVXVxxxx:03vuO4M4KmdfwffT8nqnBmnMVTr78Gu9wAN+xxxx
awsendpoint = s3.england.connectlab.io:10445
awsenablehttps = 1

[dbadmin@vertica-vm1 ~]$ admintools -t create_db -x sg_auth.conf
--communal-storage-location=s3://vertica --depot-path=/home/dbadmin/depot
--shard-count=6 -s vertica-vm1,vertica-vm2,vertica-vm3 -d vmart -p
'xxxxxxxx'
```

```
Default depot size in use
Distributing changes to cluster.
    Creating database vmart
    Starting bootstrap node v_vmart_node0007 (10.45.74.19)
    Starting nodes:
        v_vmart_node0007 (10.45.74.19)
    Starting Vertica on all nodes. Please wait, databases with a large
catalog may take a while to initialize.
    Node Status: v_vmart_node0007: (DOWN)
    Node Status: v_vmart_node0007: (DOWN)
    Node Status: v_vmart_node0007: (DOWN)
    Node Status: v_vmart_node0007: (UP)
    Creating database nodes
    Creating node v_vmart_node0008 (host 10.45.74.29)
    Creating node v_vmart_node0009 (host 10.45.74.39)
    Generating new configuration information
    Stopping single node db before adding additional nodes.
    Database shutdown complete
    Starting all nodes
Start hosts = ['10.45.74.19', '10.45.74.29', '10.45.74.39']
    Starting nodes:
        v_vmart_node0007 (10.45.74.19)
        v_vmart_node0008 (10.45.74.29)
        v_vmart_node0009 (10.45.74.39)
    Starting Vertica on all nodes. Please wait, databases with a large
catalog may take a while to initialize.
    Node Status: v_vmart_node0007: (DOWN) v_vmart_node0008: (DOWN)
v_vmart_node0009: (DOWN)
    Node Status: v_vmart_node0007: (DOWN) v_vmart_node0008: (DOWN)
v_vmart_node0009: (DOWN)
    Node Status: v_vmart_node0007: (DOWN) v_vmart_node0008: (DOWN)
v_vmart_node0009: (DOWN)
    Node Status: v_vmart_node0007: (DOWN) v_vmart_node0008: (DOWN)
v_vmart_node0009: (DOWN)
    Node Status: v_vmart_node0007: (UP) v_vmart_node0008: (UP)
v_vmart_node0009: (UP)
Creating depot locations for 3 nodes
Communal storage detected: rebalancing shards

Waiting for rebalance shards. We will wait for at most 36000 seconds.
Installing AWS package
    Success: package AWS installed
Installing ComplexTypes package
    Success: package ComplexTypes installed
Installing MachineLearning package
    Success: package MachineLearning installed
```

```
Installing ParquetExport package
    Success: package ParquetExport installed
Installing VFunctions package
    Success: package VFunctions installed
Installing approximate package
    Success: package approximate installed
Installing flextable package
    Success: package flextable installed
Installing kafka package
    Success: package kafka installed
Installing logsearch package
    Success: package logsearch installed
Installing place package
    Success: package place installed
Installing txtindex package
    Success: package txtindex installed
Installing voltagesecure package
    Success: package voltagesecure installed
Syncing catalog on vmart with 2000 attempts.
Database creation SQL tasks completed successfully. Database vmart created
successfully.
```

オブジェクトのサイズ（バイト）	バケット/オブジェクトキーの完全パス
61`	s3://Vertica/051/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a07/026d63ae9d4a33237bf0e2cf2a794a00a0000000000000021a07_00.dfd
145`	s3://Vertica/2c4/026d63ae9d4a33237bf0e2cf2a794a794a794a000000000000000021a3d/026d63ae9d4a33237bf0e2cf2a794a794a00a000000000000021a3_0.dfd
146 `	s3://Vertica/33C/026d63ae9d4a33237bf0e2cf2a794a0000000021a1d/026d63ae9d4a33237bf0e2cf2a794a00000000000021a1d_0.dfd
「40」	s3://Vertica/382/026d63ae9d4a33237bf0e2cf2a794a794a0000000021a31/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a31_0.dfs

オブジェクトのサイズ（バイト）	バケット/オブジェクトキーの完全パス
145`	s3://Vertica/42f/026d63ae9d4a33237bf0e2cf2a794a794a00000000211/026d63ae9d4a33237bf0e2cf2a794a0000000000000021a_0.dfd
34`	s3://Vertica/472/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a25/026d63ae9d4a33237bf0e2cf2a794a000000000000000021a25_0.df
41.	s3://Vertica/476/026d63ae9d4a33237bf0e2cf2a794a794a00000000000021a2d/026d63ae9d4a33237bf0e2c2cf2a794a00a00000000000021a2_0.dfd
61`	s3://Vertica/52A/026d63ae9d4a33237bf0e2cf2a794a794a0000000021a5d/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a5d_0.df
「131」	s3://Vertica/5d2/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a19/026d63ae9d4a33237bf0e2cf2a794a00a0000000000000021a19_0.df
「91」	s3://Vertica/5f7/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a11/026d63ae9d4a33237bf0e2cf2a794a00a00000000000021a11_0.df
「118」	s3://Vertica/82D/026d63ae9d4a33237bf0e2cf2a794a794a0000000021a15/026d63ae9d4a33237bf0e2cf2a794a000000000000000021a15_0.df
「115」	s3://Vertica/922/026d63ae9d4a33237bf0e2cf2a794a794a0000000021a61/026d63ae9d4a33237bf0e2cf2a794a000000000000000021a61_0.df
「33」	s3://Vertica/ACD/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a29/026d63ae9d4a33237bf0e2cf2a794a794a00000000000021a29_0.dfs

オブジェクトのサイズ（バイト）	バケット/オブジェクトキーの完全パス
「133」	s3://Vertica/b98/026d63ae9d4a33237bf0e2cf2a794a794a00a00000000000021a4d/026d63ae9d4a33237bf0e2cf2a794a794a00a000000000021a4d_0.df
「38」	s3://Vertica/db3/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a49/026d63ae9d4a33237bf0e2cf2a794a00a0000000000000021a49_0.df
「38」	`s3://Vertica/eba/026d63ae9d4a33237bf0e2cf2a794a794a000000000021a59/026d63ae9d4a33237bf0e2cf2a794a794a0000000021a59_0.dfdfs.dfs
21521920`	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a000000000000215e2/026d63ae9d4a33237bf0e2c2cf2a794a000000215e2.tar
6865408`	s3://vertica/metodat/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a0000000021602/026d63ae9d4a33237bf0e2c2cf2a794a0000000000000021602.tar
「204217344」	s3://vertica/metodata/mart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a0000000021610/026d63ae9d4a33237bf0e2c2cf2a794a00a00000000000000021610.tar
16109056`	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a00000000217e0/026d63ae9d4a33237bf0e2c2cf2a794a0000000000217e0.tar
12853248`	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a0000000000000021800/026d63ae9d4a33237bf0e2c2cf2a794a00a000000000000000000000000218.tar
「8937984」と入力します	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a00000000002187a/026d63ae9d4a33237bf0e2c2cf2a794a00a00002187a.tar

オブジェクトのサイズ（バイト）	バケット/オブジェクトキーの完全パス
「56260`606060860」	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a00000000000000218b2/026d63ae9d4a33237bf0e2c2cf2a794a0000000000218b2.tar
「53947904」	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a00000000000000219ba/026d63ae9d4a33237bf0e2c2cf2a794a0000000000219ba.tar
44932`608	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a000000000000219de/026d63ae9d4a33237bf0e2c2cf2a794a00000000000000219de.tar
「256306688」	s3://vertica/metadata/VMart/Librarys/026d63ae9d4a33237bf0e2c2cf2a794a00a00000000000021a6e/026d63ae9d4a33237bf0e2c2cf2a794a000000000000000021a6e.tar
「8062464`」	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a0000000021e34/026d63ae9d4a33237bf0e2c2cf2a794a0000000000000000021e34.tar
「20024832」	s3://vertica/metadata/VMart/Library/Libraryd63ae9d4a33237bf0e2c2cf2a794a00a0000000021e70/026d63ae9d4a33237bf0e2c2cf2a794a0000000000000000021e70.tar
「10444」	`s3://vertica/metadata/VMart/cluster _config.json
「823266」	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checks/C13_13/chkpt_1.cat.gz
「254」	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checks/C13_13/Completed
「2958」	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C2_2/chkpt_1.cat.gz

オブジェクトのサイズ（バイト）	バケット/オブジェクトキーの完全パス
231`	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checks/C2_2/Completed
「822521」	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checks/C4_4/chkpt_1.cat.gz
231`	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checks/C4_4/Completed
746513`	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_14_g14.cat
「2596」	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_3_g3.cat.gz
821065`	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_4_g4.cat.gz
6440`	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_5_g5.cat
「8518」	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_8_g8.cat
「 0 」	s3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/tiered_catalog.cat
822922`	s3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/Checkpoints /Checkpoints /C14-7/chkpt_1.cat.gz

オブジェクトのサイズ（バイト）	バケット/オブジェクトキーの完全パス
「232」	`s3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/Checkpoints /C14-7/Completed`
822930`	`s3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_14_g7.cat.gz`
755033`	`s3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_15_g8.cat`
「 0 」	`s3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/tiered_catalog.cat`
822922`	`s3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/Checkpoints /C14-7/chkpt_1.cat.gz`
「232」	`s3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/Checkpoints /C14-7/Completed`
822930`	`s3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_14_g7.cat.gz`
755033`	`s3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_15_g8.cat`
「 0 」	`s3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/tiered_catalog.cat`

ストリーミング制限を無効にしています

この手順 は、他のオンプレミスオブジェクトストレージのVertica guideに基づいており、StorageGRID に適用する必要があります。

1. データベースを作成したら'AWSStreamingConnectionPercentage'設定パラメータを0に設定して無効にしますこの設定は、共同ストレージを使用したオンプレミス環境でのEonモードのインストールには不要です。この設定パラメータは、Verticaがストリーミング読み取りに使用するオブジェクトストアへの接続数を制御します。クラウド環境では、この設定が有効な場合、オブジェクトストアからのストリーミングデータが使用可能なすべてのファイルハンドルを使い使わないようにすることができます。他のオブジェクトストア処理に使用できるファイルハンドルが残っています。オンプレミスのオブジェクトストアのレイテンシが低いため、このオプションは不要です。
2. パラメータ値を更新するには'vsql'文を使用しますパスワードは、「オンプレミスデータベースの作成」で設定したデータベースパスワードです。たとえば、次の出力例を参照してください。

```
[dbadmin@vertica-vm1 ~]$ vsql
Password:
Welcome to vsql, the Vertica Analytic Database interactive terminal.
Type:   \h or \? for help with vsql commands
        \g or terminate with semicolon to execute query
        \q to quit
dbadmin=> ALTER DATABASE DEFAULT SET PARAMETER
AWSStreamingConnectionPercentage = 0; ALTER DATABASE
dbadmin=> \q
```

デポの設定を確認してい

Verticaデータベースのデフォルトデポ設定は、読み取りおよび書き込み操作に対して有効(値=1)です。パフォーマンスを向上させるために、これらのデポ設定を有効にしておくことを強く推奨します。

```
vsql -c 'show current all;' | grep -i UseDepot
DATABASE | UseDepotForReads | 1
DATABASE | UseDepotForWrites | 1
```

サンプルデータのロード（オプション）

このデータベースをテスト用に使用し、削除する場合は、サンプルデータをテスト用にこのデータベースにロードできます。Verticaには、各Verticaノードの「/opt/vertica/examples/VMart _Schema/」にあるサンプルデータセットVMartが付属しています。このサンプルデータセットの詳細については、を参照してください "こちらをご覧ください"。

サンプルデータをロードするには、次の手順を実行します。

1. いずれかのVerticaノードにdbadminとしてログインします。cd /opt/vertica/examples/VMart _Schema/
2. サンプルデータをデータベースにロードし、手順cとdでプロンプトが表示されたらデータベースのパスワードを入力します。
   a. 「cd /opt/vertica/examples/VMart _Schema'」と入力します
   b. 「./vmart_gen」
   c. vsql <vmart_define_schema.sql
   d. 「vsql <vmart_load_data.sql」

3. 事前定義された複数のSQLクエリがあります。そのうちの一部を実行して、テストデータがデータベースに正常にロードされたことを確認できます。たとえば、「vsql <vmart_queries1.sql」のようになります

追加情報の参照先

このドキュメントに記載されている情報の詳細については、以下のドキュメントや Web サイトを参照してください。

- "NetApp StorageGRID 11.7製品ドキュメント"
- "StorageGRID データシート"
- "Vertica 10.1製品マニュアル"

バージョン履歴

バージョン	日付	ドキュメントのバージョン履歴
バージョン 1.0 以降	2021年9月	初版リリース

Angela Cheng著_

エルクスタックを使用した**StorageGRID** ログ分析

StorageGRID 11.6 syslog転送機能を使用すると、StorageGRID ログメッセージを収集および分析するように外部syslogサーバを設定できます。エルク（Elasticsearch、Logstash、Kibana）は、最も人気のあるログ分析ソリューションの1つになっています。をご覧ください "エルク・ビデオを使用したStorageGRID ログ解析" サンプルのエルク設定を表示し、失敗したS3要求を特定してトラブルシューティングするためにどのように使用できるかを確認する。この記事では、StorageGRID ログの管理と分析をすばやく開始できるように、Logstashの設定、Kibanaのクエリ、グラフ、およびダッシュボードのサンプルファイルを紹介します。

要件

- StorageGRID 11.6.0.2以降
- Elk（Elasticsearch、Logstash、Kibana）7.1x以降がインストールされており、動作中です

サンプルファイル

- "Logstash 7.xサンプルファイルパッケージをダウンロードします" +*MD5チェックサム*148c23d0021d9a4bb4a6c0287464deab +*SHA256チェックサム*f51ec9e2e3f842d5a7861566b167a561beb4373038b4e7bb3c8be3d522adf2d6
- "Logstash 8.xサンプルファイルパッケージをダウンロードします" *MD5チェックサム*e11bae3a662f87c310ef363d0fe06835* SHA256チェックサム*5c670755742cfdfd5aa723a596ba087e0153a65bcaef3934afdb682f61cd278d

前提条件

読者はStorageGRID およびElkの用語および操作に精通しています。

指示

grokパターンで定義される名前の違いにより、2つのサンプルバージョンが提供されます。+たとえば、Logstash設定ファイルのSYSLOGBASE grokパターンでは、インストールされているLogstashのバージョンによってフィールド名が異なります。

```
match => {"message" => '<%{POSINT:syslog_pri}>%{SYSLOGBASE}
%{GREEDYDATA:msg-details}'}
```

- Logstash 7.17サンプル*

Field	Value
t _id	7C1MaYEBRH8UbfKnIls8
t _index	sgrid2-2022.06.15
# _score	-
t _type	_doc
@timestamp	Jun 15, 2022 @ 17:36:46.038
t host	grid2-site2-s1
t logsource	SITE2-S1
t msg-details	Reloading syslog service
t pid	628
t program	update-sysl
t syslog_pri	37
t timestamp	Jun 15 21:36:46

ログスタシュ**8.23**サンプル

Table JSON

Search field names

Actions	Field	Value
	_id	yuhOiIEBVP6KX4EwqcyU
	_index	sglog-2022.06.21
	_score	-
	@timestamp	Jun 21, 2022 @ 18:07:45.444
	event.original	<28>Jun 21 22:07:45 SITE2-S3 ADE: syslog messages being dropped
	host.hostname	SITE2-S3
	msg-details	syslog messages being dropped
	process.name	ADE
	syslog_pri	28
	timestamp	Jun 21 22:07:45

- 手順 *

1. インストールされているエルクバージョンに基づいて、提供されたサンプルを解凍します。サンプル・フォルダには'Logstash configサンプルが2つ含まれています**sglog-2-file.conf :**この構成ファイルは’データ変換を行わずに**Logstash**上のファイルに**StorageGRID** ログ・メッセージを出力しますこの機能を使用すると、**Logstash**が**StorageGRID** メッセージを受信していることを確認したり、**StorageGRID** ログパターンを理解したりできます。**+** sglog-2-es.conf：*この構成ファイルは、さまざまなパターンやフィルタを使用してStorageGRID ログメッセージを変換します。この例には、パターンまたはフィルタに基づいてメッセージをドロップするDROPステートメントが含まれています。インデックスを作成するために出力がElasticsearchに送信されます。+ファイル内の指示に従って、選択した構成ファイルをカスタマイズします。
2. カスタマイズした構成ファイルをテストします。

```
/usr/share/logstash/bin/logstash --config.test_and_exit -f <config-
file-path/file>
```

返される最後の行が次の行に似ている場合、構成ファイルに構文エラーはありません。

```
[LogStash::Runner] runner - Using config.test_and_exit mode. Config
Validation Result: OK. Exiting Logstash
```

3. カスタマイズされたconfファイルをLogstashサーバのconfig:/etc/logstash/conf.d+にコピーします/etc/logstash/logstash.ymlでconfig.reload.automaticを有効にしていない場合は’Logstashサービスを再起動しますそれ以外の場合は、設定のリロード間隔が経過するのを待ちます。

```
 grep reload /etc/logstash/logstash.yml
# Periodically check if the configuration has changed and reload the
pipeline
config.reload.automatic: true
config.reload.interval: 5s
```

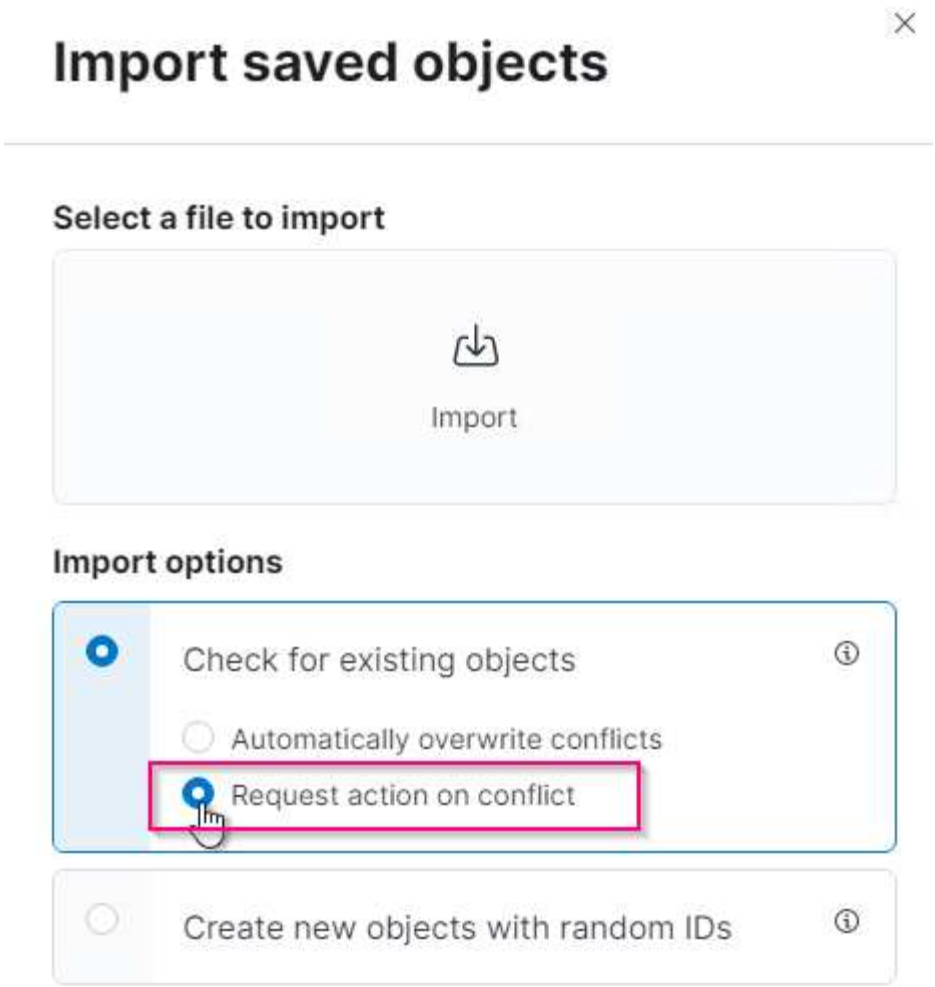
4. /var/log/logstash/logstash-plain.logを確認し、Logstashを新しい設定ファイルで起動する際にエラーがないことを確認します。
5. TCPポートが開始され、リスンしていることを確認する。+この例では、TCPポート5000が使用されています。

```
netstat -ntpa | grep 5000
tcp6       0      0 :::5000                 :::*
LISTEN      25744/java
```

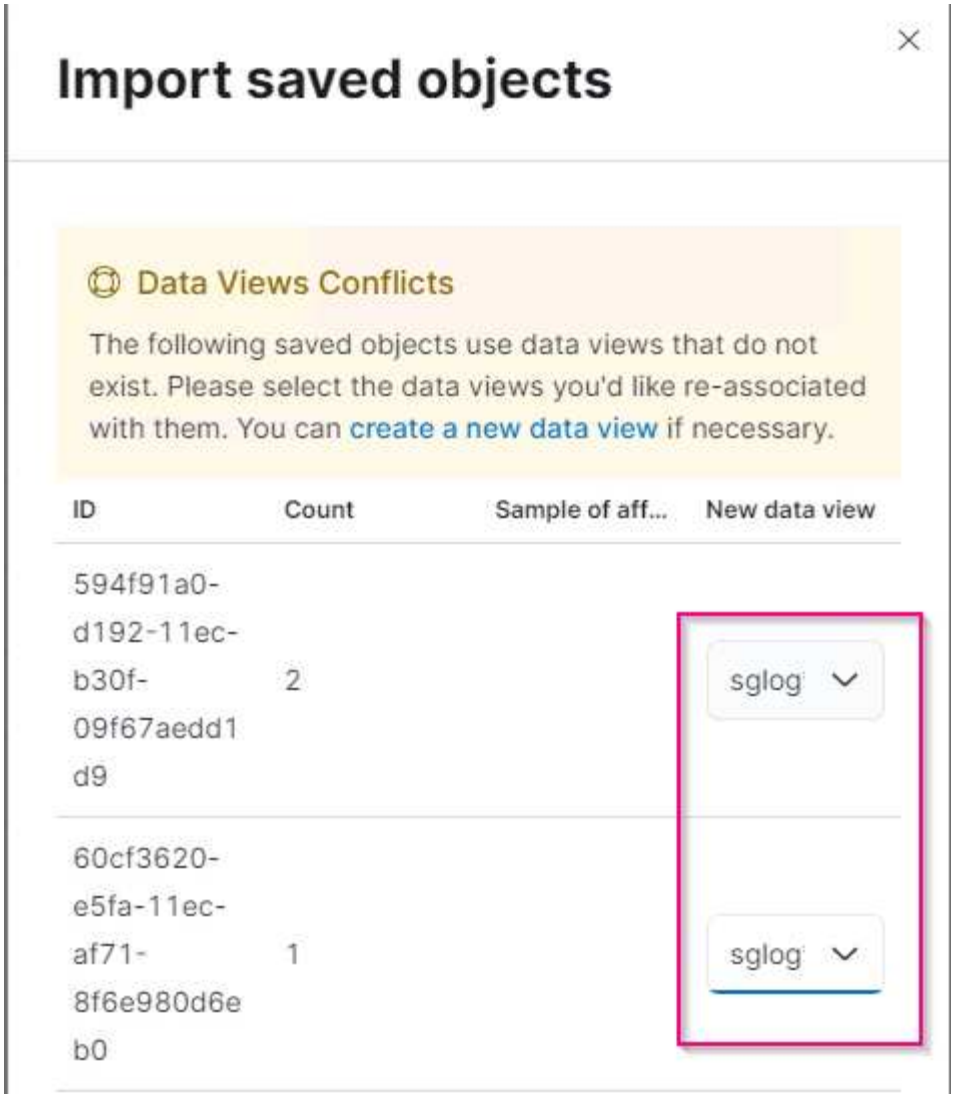
6. StorageGRID マネージャGUIから、ログメッセージをLogstashに送信するように外部syslogサーバを設定します。を参照してください "デモビデオ" を参照してください。
7. 定義されたTCPポートへのStorageGRID ノード接続を許可するには、Logstashサーバ上でファイアウォールを設定または無効にする必要があります。
8. Kibana GUIから、[Management]→[Dev Tools]を選択します。Consoleページで、次のgetコマンドを実行して、Elasticsearch上に新しいインデックスが作成されていることを確認します。

```
GET /_cat/indices/*?v=true&s=index
```

9. Kibana GUIから、索引パターン（Elk 7.x）またはデータビュー（Elk 8.x）を作成します。
10. Kibana GUIから、上部中央にある検索ボックスに「saved objects」と入力します。+[保存済みオブジェクト]ページで、[インポート]を選択します。[インポートオプション]で、[競合時にアクションを要求]を選択します。



elk <version>-query-chart-sample.ndjsonをインポートします。+競合を解決するよう求められたら、手順8で作成したインデックスパターンまたはデータビューを選択します。



次のKibanaオブジェクトがインポートされます。 *クエリ** audit-msg-s3rq -lm+* bycast log s3関連メッセージ+* loglevel warningまたはabove * failed security event +* Chart ** s3要求数bycast.log * HTTP status code +* audit breakdown by type +* average s3応答 上記のグラフを使用した、時間ダッシュボード+* S3要求ダッシュボード。

これで、Kibanaを使用してStorageGRID ログ分析を実行する準備ができました。

その他のリソース

- "syslog101"
- "エルクスタックとは何ですか"
- "grokパターンリスト"
- "初心者向けのLogstashガイド: Grok"
- "ログスタシュの実践的なガイド：syslogの詳細"
- "Kibanaガイド–ドキュメントを参照してください"
- "StorageGRID 監査ログメッセージリファレンスです"

Angela Cheng著_

PrometheusとGrafanaを使用して指標の保持を拡張します

このテクニカルレポートでは、外部のPrometheusサービスおよびGrafanaサービスでNetApp StorageGRID 11.6を設定する詳しい手順を説明します。

はじめに

StorageGRID は、Prometheusを使用して指標を保存し、組み込みのGrafanaダッシュボードでこれらの指標を視覚化します。Prometheus指標には、クライアントアクセス証明書を設定し、指定されたクライアントのPrometheusアクセスを有効にすることで、StorageGRID から安全にアクセスできます。現在、この指標データの保持期間は管理ノードのストレージ容量によって制限されています。これらの指標のカスタマイズされた可視化を実現するために、新しいPrometheusサーバとGrafanaサーバを導入し、新しいサーバでStorageGRIDWebscaleインスタンスから指標をスクラビングするように設定し、重要な指標を使用したダッシュボードを構築します。で収集されたPrometheus指標の詳細を確認できます "StorageGRID のドキュメント"。

Prometheusをフェデレーションする

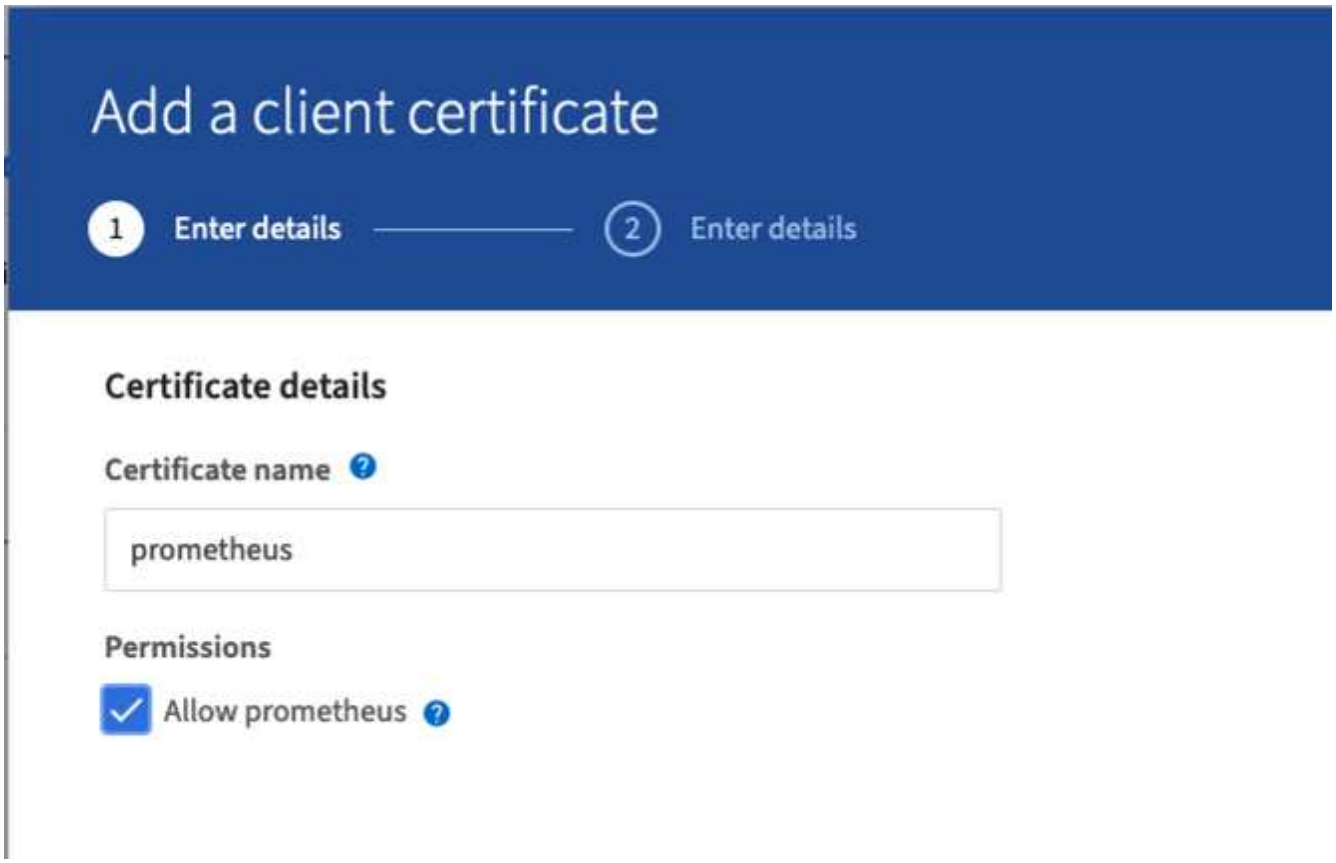
ラボの詳細

この例では、StorageGRID 11.6ノードとDebian 11サーバのすべての仮想マシンを使用します。StorageGRID管理インターフェイスには、公開されている信頼されたCA証明書が設定されています。この例では、StorageGRID システムやDebian Linuxのインストールと設定は行われません。PrometheusとGrafanaでサポートされている、任意のLinuxフレーバーを使用できます。PrometheusとGrafanaはどちらも、Dockerコンテナ、ソースからビルド、またはコンパイル済みのバイナリとしてインストールできます。この例では、PrometheusバイナリとGrafanaバイナリの両方を同じDebianサーバに直接インストールします。から基本的なインストール手順をダウンロードして実行します https://prometheus.io および https://grafana.com/grafana/ それぞれ。

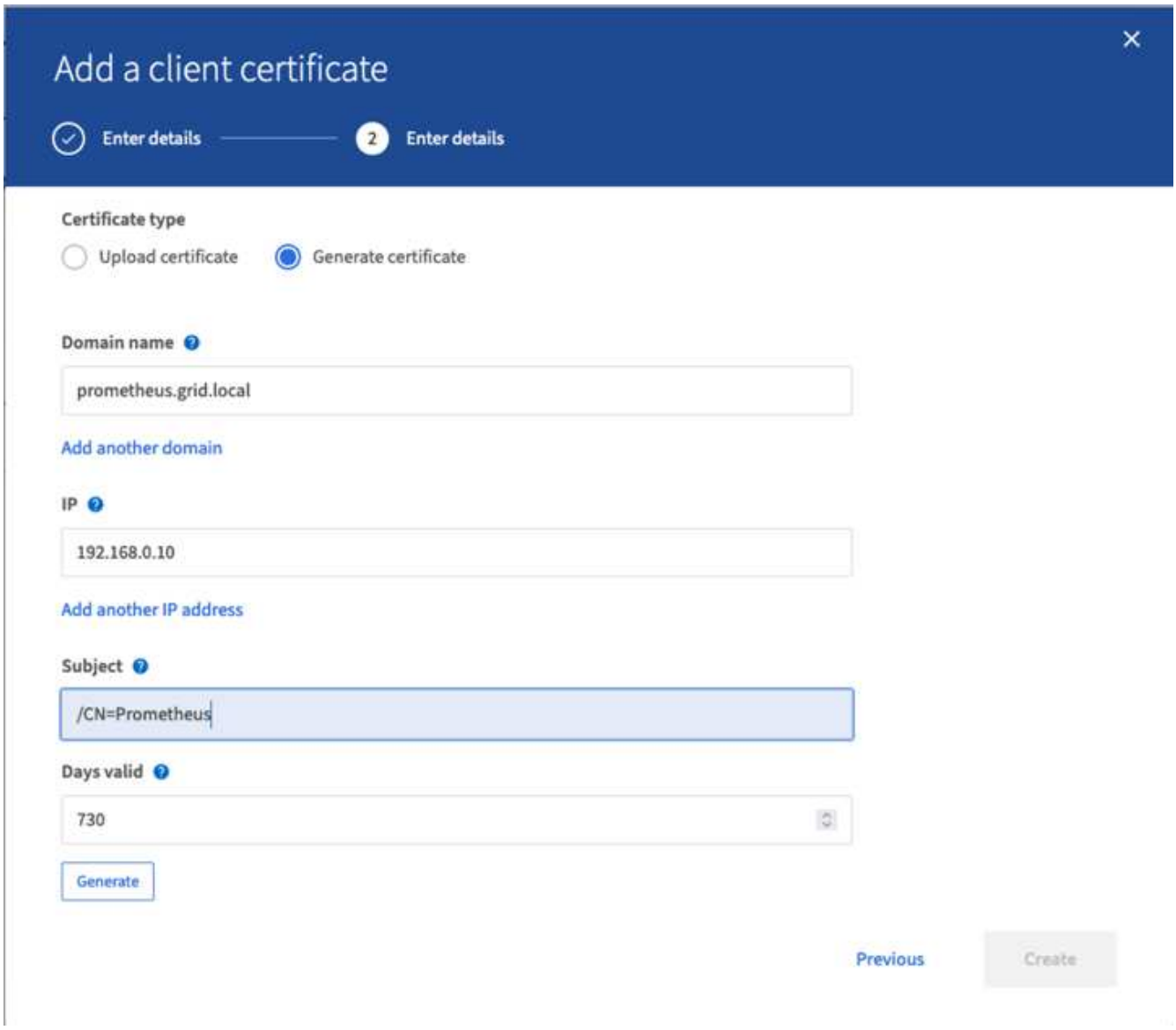
Prometheusクライアントアクセス用にStorageGRID を設定する

StorageGRID IDに格納されているPrometheus指標にアクセスするには、秘密鍵を使用してクライアント証明書を生成またはアップロードし、クライアントの権限を有効にする必要があります。StorageGRID 管理インターフェイスにはSSL証明書が必要です。この証明書は、信頼されたCAによってPrometheusサーバによって信頼されているか、自己署名されている場合は手動で信頼されている必要があります。詳細については、を参照してください "StorageGRID のドキュメント"。

1. StorageGRID 管理インターフェイスの左下にある「configuration」を選択し、2番目の列にある「Security」で「Certificates」をクリックします。
2. [証明書]ページで[クライアント]タブを選択し、[追加]ボタンをクリックします。
3. アクセスを許可するクライアントの名前を指定し、この証明書を使用します。「Allow Prometheus」の前の「Permissions」のボックスをクリックし、「Continue」ボタンをクリックします。

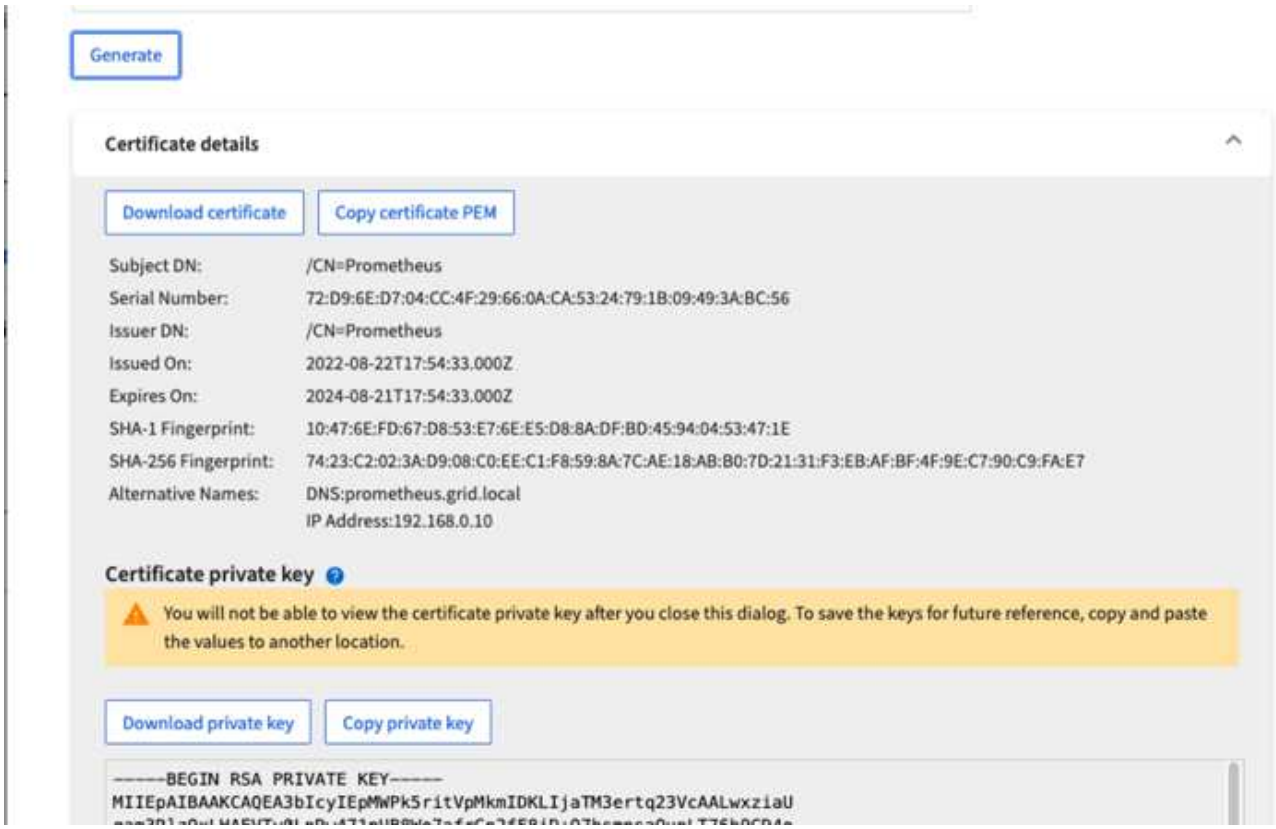


4. CA署名証明書がある場合は、[証明書のアップロード]のラジオボタンを選択できますが、この場合は、[証明書の生成]のラジオボタンを選択して、StorageGRID がクライアント証明書を生成できるようにします。入力する必須フィールドが表示されます。クライアントサーバのFQDN、サーバのIP、件名、有効日数を入力します。「生成」ボタンをクリックします。




```
Be mindful of the certificate days valid entry as you will need
to renew this certificate in both StorageGRID and the Prometheus
server before it expires to maintain uninterrupted collection.
```

1. 証明書のPEMファイルと秘密鍵のPEMファイルをダウンロードします。



> This is the only time you can download the private key, so make sure you do not skip this step.

Linuxサーバで**Prometheus**インストールを準備

Prometheusをインストールする前に、Prometheusユーザとディレクトリ構造を使用して環境を準備し、指標の格納場所の容量を設定します。

1. Prometheusユーザを作成します。

```
sudo useradd -M -r -s /bin/false Prometheus
```

2. Prometheus、クライアント証明書、指標データのディレクトリを作成します。

```
sudo mkdir /etc/Prometheus /etc/Prometheus/cert /var/lib/Prometheus
```

3. 私はext4ファイルシステムでのメトリック保持のために使用するディスクをフォーマットしました。

```
mkfs -t ext4 /dev/sdb
```

4. そのあと、Prometheusのmetricsディレクトリにファイルシステムをマウントしました。

```
sudo mount -t auto /dev/sdb /var/lib/prometheus/
```

5. 指標データに使用するディスクのUUIDを取得します。

```
sudo ls -al /dev/disk/by-uuid/
   lrwxrwxrwx 1 root root   9 Aug 18 17:02 9af2c5a3-bfc2-4ec1-85d9-
ebab850bb4a1 -> ../../sdb
```

6. /etc/fstabにエントリを追加してマウントをリブート後も/dev/sdbのUUIDを使用して維持するようにします。

```
/etc/fstab
UUID=9af2c5a3-bfc2-4ec1-85d9-ebab850bb4a1 /var/lib/prometheus   ext4
defaults    0   0
```

Prometheusをインストールして設定する

これでサーバの準備ができました。Prometheusのインストールを開始して、サービスを設定できます。

1. Prometheusインストールパッケージを展開します

```
tar xzf prometheus-2.38.0.linux-amd64.tar.gz
```

2. バイナリを/usr/local/binにコピーし、前の手順で作成したPrometheusユーザに所有権を変更します

```
sudo cp prometheus-2.38.0.linux-amd64/{prometheus,promtool}
/usr/local/bin
sudo chown prometheus:prometheus /usr/local/bin/{prometheus,promtool}
```

3. コンソールとライブラリを/etc/Prometheusにコピーします

```
sudo cp -r prometheus-2.38.0.linux-amd64/{consoles,console_libraries}
/etc/prometheus/
```

4. 以前にStorageGRID からダウンロードしたクライアント証明書と秘密鍵のPEMファイルを/etc/prometheus/certsにコピーします
5. Prometheus設定YAMLファイルを作成します

```
sudo nano /etc/prometheus/prometheus.yml
```

6. 次の構成を挿入します。ジョブ名には、任意の名前を指定できます。「-targets：[']]」を管理ノードのFQDNに変更し、証明書と秘密鍵のファイル名を変更した場合は、tls_configセクションを更新して一致させてください。次に、ファイルを保存します。グリッド管理インターフェイスで自己署名証明書を使用している場合は、証明書をダウンロードして一意の名前のクライアント証明書に格納し、tls_configセクションadd ca_file：/etc/prometheus/cert/UIcert.pemに格納します

   a. この例では、alertmanager、cassandra、node、およびStorageGRID で始まるすべての指標を収集しています。Prometheus指標の詳細については、を参照してください "StorageGRID のドキュメント"。

```
# my global config
global:
  scrape_interval: 60s # Set the scrape interval to every 15 seconds.
Default is every 1 minute.

scrape_configs:
  - job_name: 'StorageGRID'
    honor_labels: true
    scheme: https
    metrics_path: /federate
    scrape_interval: 60s
    scrape_timeout: 30s
    tls_config:
      cert_file: /etc/prometheus/cert/certificate.pem
      key_file: /etc/prometheus/cert/private_key.pem
    params:
      match[]:
        -
'{__name__=~"alertmanager_.*|cassandra_.*|node_.*|storagegrid_.*"}'
    static_configs:
    - targets: ['sgdemo-rtp.netapp.com:9091']
```

> グリッド管理インターフェイスで自己署名証明書が使用されている場合は、証明書をダウンロードして一意の名前でクライアント証明書に格納します。tls_configセクションで、クライアント証明書と秘密鍵の行の上に証明書を追加します
>
> ```
> ca_file: /etc/prometheus/cert/UIcert.pem
> ```

1. Prometheus内のすべてのファイルとディレクトリの所有権と、/var/lib/prometPrometheusユーザへの所有権を変更する

```
sudo chown -R prometheus:prometheus /etc/prometheus/
sudo chown -R prometheus:prometheus /var/lib/prometheus/
```

2. /etc/systemd/systemにPrometheusサービスファイルを作成します

```
sudo nano /etc/systemd/system/prometheus.service
```

3. 次の行を挿入します。--storage.tsd.dretention.time=1y#というメトリックデータの保持期間を1年にします。また、ストレージの制限に基づいて保持期間を設定する場合も、--storage.tsdb.retentionsize=300GiB#を使用することもできます。指標の保持を設定できるのは、この場所だけです。

```
[Unit]
Description=Prometheus Time Series Collection and Processing Server
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
        --config.file /etc/prometheus/prometheus.yml \
        --storage.tsdb.path /var/lib/prometheus/ \
        --storage.tsdb.retention.time=1y \
        --web.console.templates=/etc/prometheus/consoles \
        --web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

4. システムdサービスをリロードして新しいPrometheusサービスを登録します。その後、Prometheusサービスを開始して有効にします。

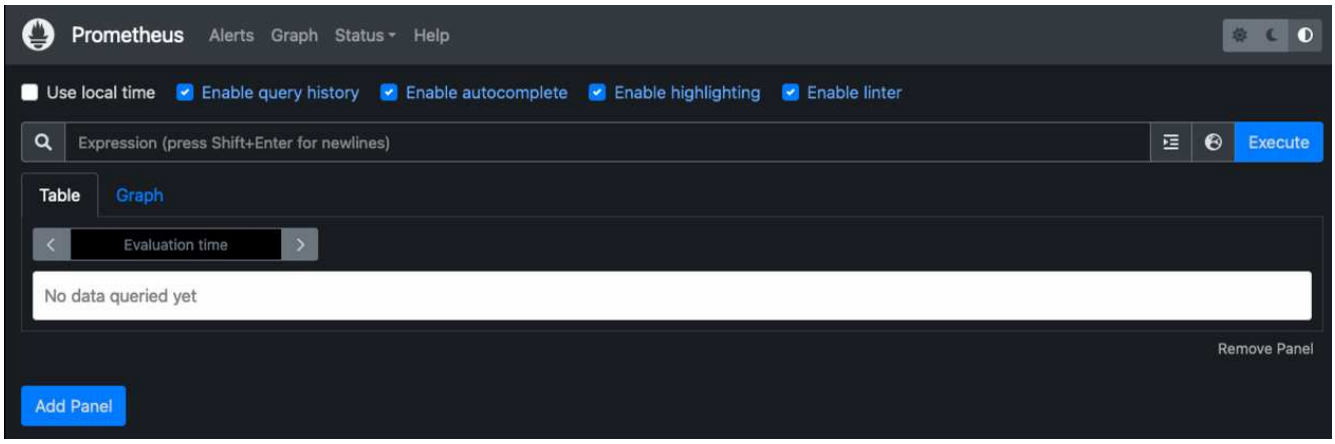
```
sudo systemctl daemon-reload
sudo systemctl start prometheus
sudo systemctl enable prometheus
```

5. サービスが正常に実行されていることを確認します

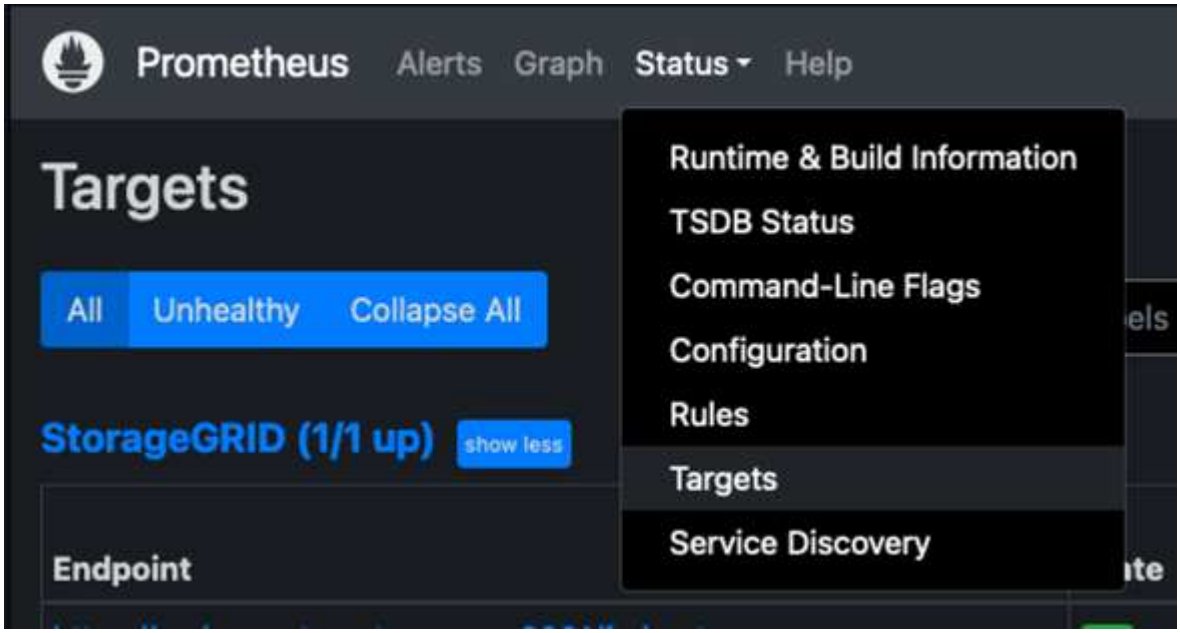
```
sudo systemctl status prometheus
```

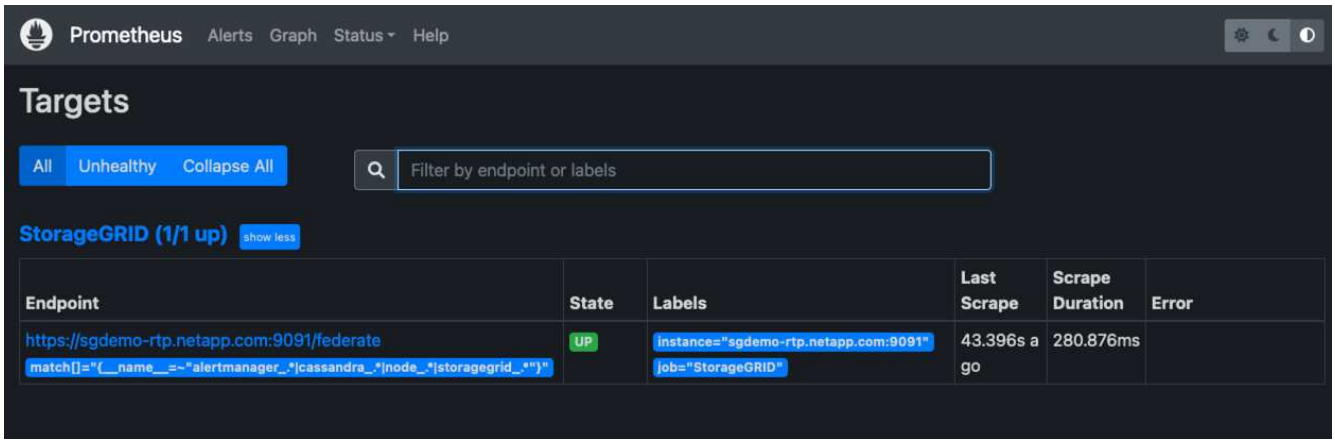
```
● prometheus.service - Prometheus Time Series Collection and Processing
Server
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled;
vendor preset: enabled)
     Active: active (running) since Mon 2022-08-22 15:14:24 EDT; 2s ago
   Main PID: 6498 (prometheus)
      Tasks: 13 (limit: 28818)
     Memory: 107.7M
        CPU: 1.143s
     CGroup: /system.slice/prometheus.service
             └─6498 /usr/local/bin/prometheus --config.file
/etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/
--web.console.templates=/etc/prometheus/consoles --web.con>

Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.510Z caller=head.go:544 level=info component=tsdb
msg="Replaying WAL, this may take a while"
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.816Z caller=head.go:615 level=info component=tsdb msg="WAL
segment loaded" segment=0 maxSegment=1
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.816Z caller=head.go:615 level=info component=tsdb msg="WAL
segment loaded" segment=1 maxSegment=1
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.816Z caller=head.go:621 level=info component=tsdb msg="WAL
replay completed" checkpoint_replay_duration=55.57µs wal_rep>
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.831Z caller=main.go:997 level=info fs_type=EXT4_SUPER_MAGIC
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.831Z caller=main.go:1000 level=info msg="TSDB started"
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.831Z caller=main.go:1181 level=info msg="Loading
configuration file" filename=/etc/prometheus/prometheus.yml
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.832Z caller=main.go:1218 level=info msg="Completed loading
of configuration file" filename=/etc/prometheus/prometheus.y>
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.832Z caller=main.go:961 level=info msg="Server is ready to
receive web requests."
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.832Z caller=manager.go:941 level=info component="rule
manager" msg="Starting rule manager..."
```

6. PrometheusサーバのUIにアクセスできるようになります http://Prometheus-server:9090 およびUIを参照してください

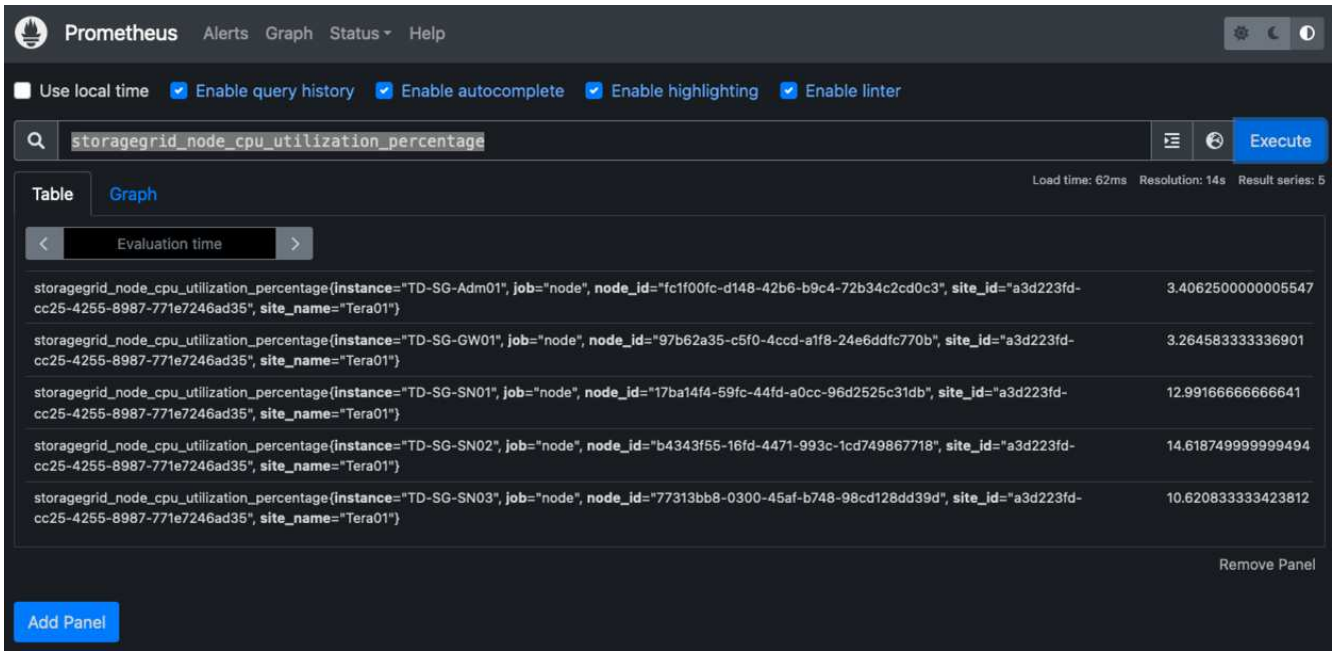


7. 「Status」ターゲットのPrometheusで設定したStorageGRID エンドポイントのステータスを確認できます





8. [グラフ]ページで、テストクエリを実行し、データが正常にスクレイピングされていることを確認できます。たとえば、クエリバーに「storagegrid_node_name utilization _percentage」と入力し、実行ボタンをクリックします。



Grafanaをインストールして設定します

Prometheusがインストールされて機能したので、Grafanaのインストールとダッシュボードの設定に進みます

Grafanaの分析

1. Grafanaの最新のエンタープライズエディションをインストールします

```
sudo apt-get install -y apt-transport-https
sudo apt-get install -y software-properties-common wget
sudo wget -q -O /usr/share/keyrings/grafana.key
https://packages.grafana.com/gpg.key
```

2. 安定版リリース用に次のリポジトリを追加します。

```
echo "deb [signed-by=/usr/share/keyrings/grafana.key]
https://packages.grafana.com/enterprise/deb stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list
```

3. リポジトリを追加した後。

```
sudo apt-get update
sudo apt-get install grafana-enterprise
```

4. systemdサービスをリロードして新しいgrafanaサービスを登録します。次に、Grafanaサービスを開始して有効にします。

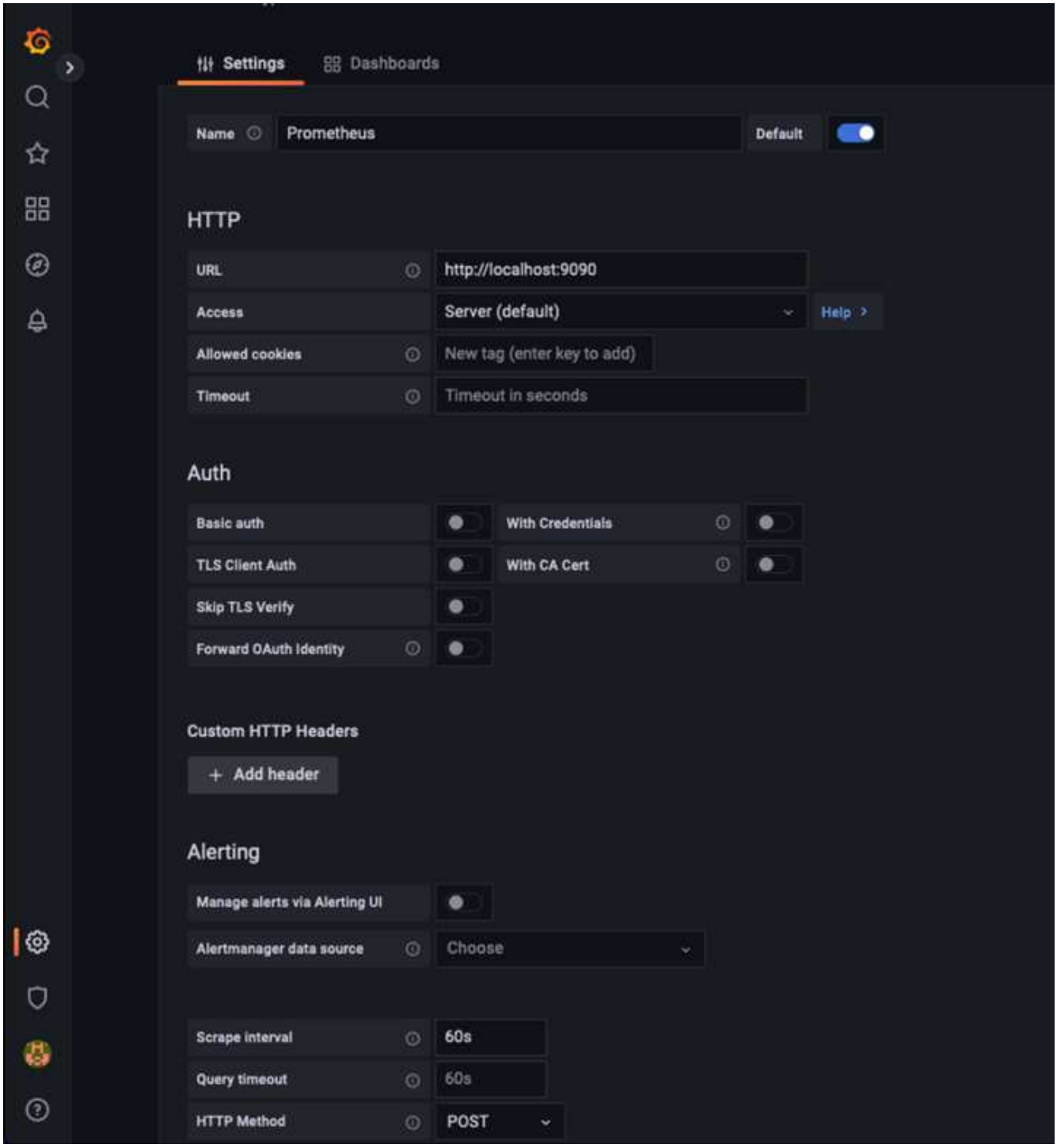
```
sudo systemctl daemon-reload
sudo systemctl start grafana-server
sudo systemctl enable grafana-server.service
```

5. Grafanaがインストールされて実行されるようになりました。ブラウザでHTTP：//prometheus-server：3000にアクセスすると、Grafanaのログインページが表示されます。
6. デフォルトのログインクレデンシャルはadmin / adminであり、新しいパスワードを要求されたときに設定する必要があります。

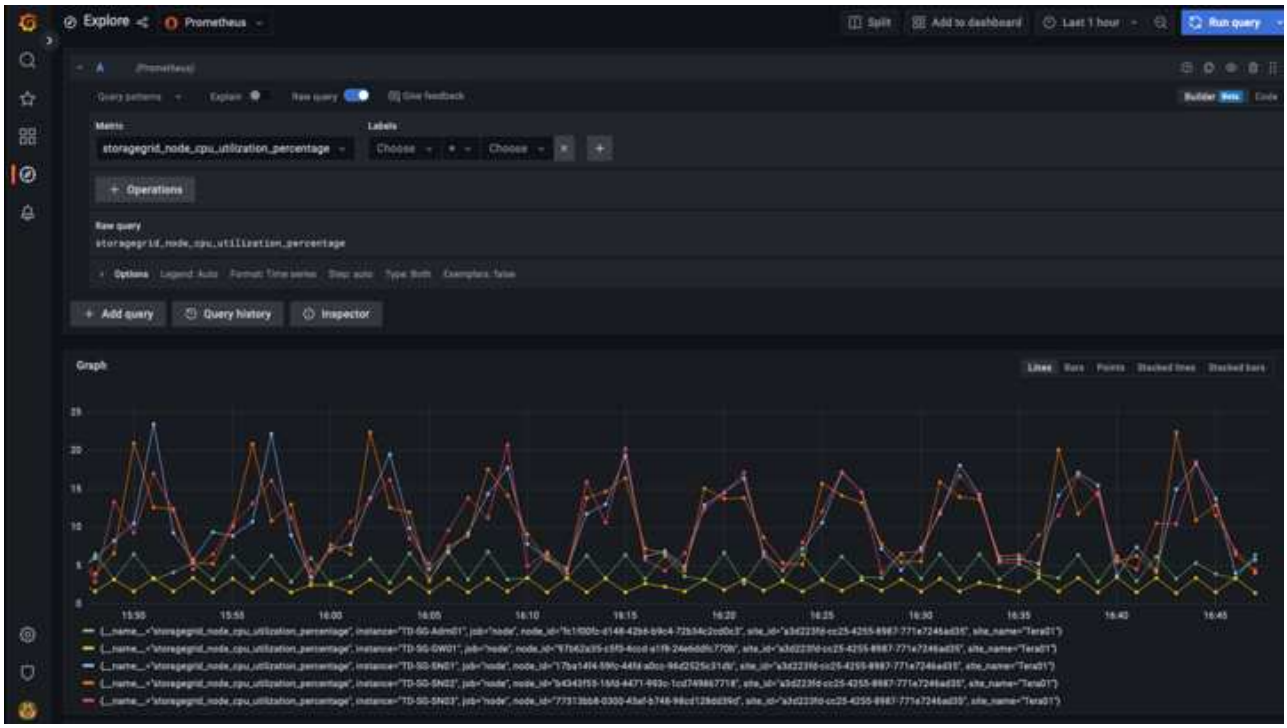
StorageGRID に対応した**Grafana**ダッシュボードを作成します

GrafanaとPrometheusがインストールされて実行されている状態で、データソースを作成してダッシュボードを構築することで、この2つを接続する時間が発生します

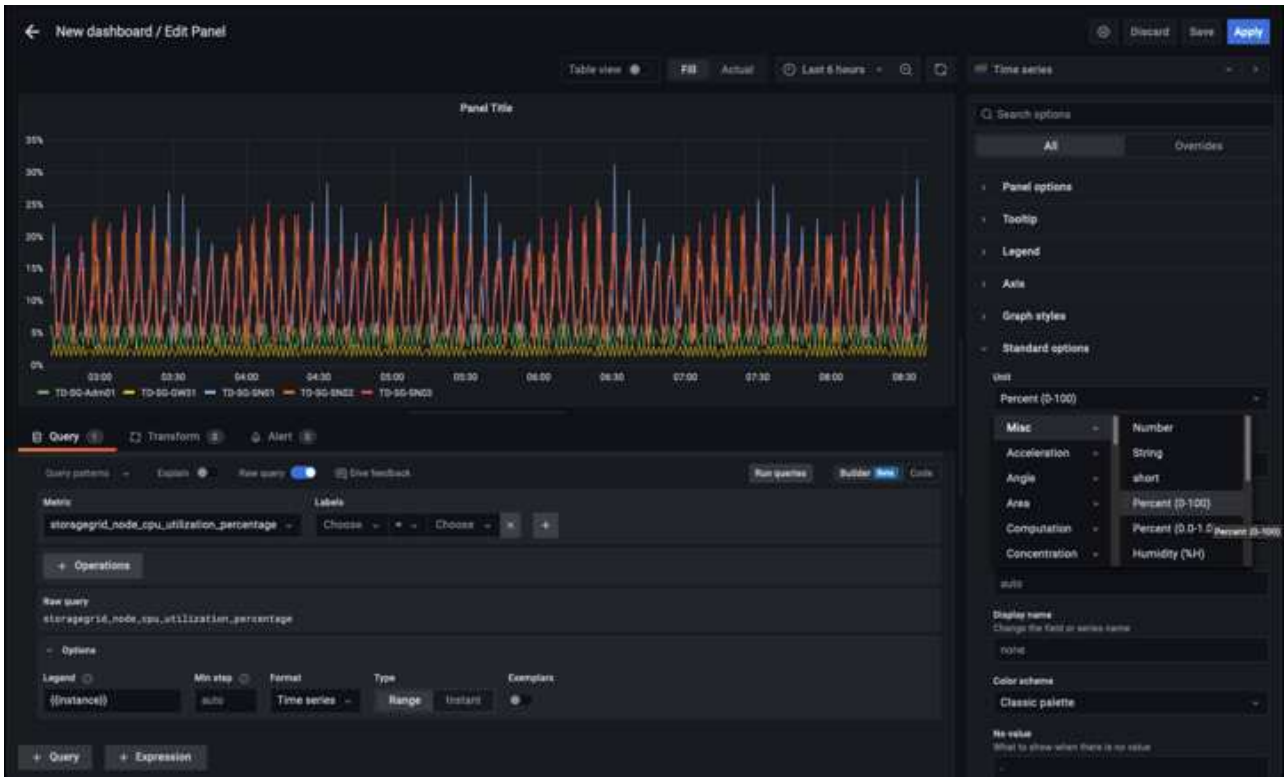
1. 左側のペインで[構成]を展開し、[データソース]を選択して、[データソースの追加]ボタンをクリックします
2. Prometheusは、最も人気のあるデータソースの1つです。検出されていない場合は、検索バーで「Prometheus」を特定します。
3. PrometheusインスタンスのURLとスクラビング間隔をPrometheusの間隔と一致するように入力して、Prometheusソースを設定します。Prometheusでアラートマネージャを設定しなかったため、アラートセクションも無効にしました。



4. 目的の設定を入力したら、下にスクロールして[保存してテスト]をクリックします。
5. 設定テストが正常に完了したら、[EXPLOR]ボタンをクリックします。
   a. 「調査」ウィンドウで、Prometheusで「storagegrid_node_name」に対してテストしたものと同じ指標を使用し、「Run query」ボタンをクリックします



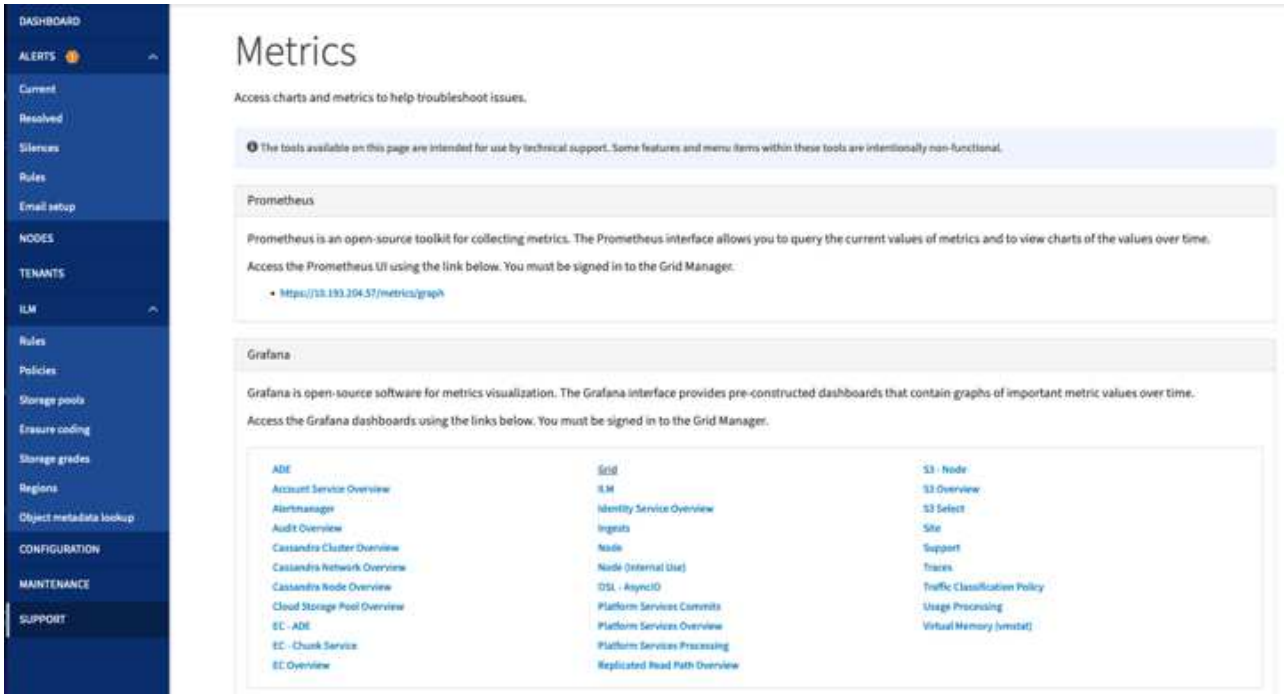
6. データソースを設定したら、ダッシュボードを作成します。
   a. 左側のペインで[ダッシュボード]を展開し、[+新しいダッシュボード]を選択します。
   b. 「新規パネルを追加」を選択します。
   c. メトリックを選択して新しいパネルを設定します。もう一度「storagegrid_node_name」を使用し、パネルのタイトルを入力し、下部に「Options」を展開して凡例をカスタムに変更し、「｛｛instance｝｝」と入力してノード名を定義します。右側のペインの「Standard options」set "Unit」を「Misc-100%」に設定します。[適用]をクリックして、パネルをダッシュボードに保存します。



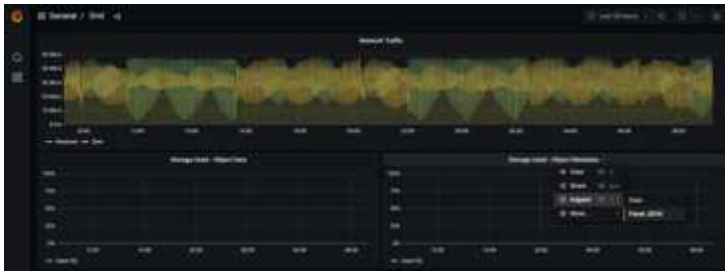
7. 必要な指標ごとにこのようなダッシュボードを構築し続けることもできますが、幸運にも、StorageGRID にはダッシュボードがすでに用意されており、カスタムダッシュボードにコピーすることができます。

   a. StorageGRID 管理インターフェイスの左側のペインで、[サポート]を選択し、[ツール]列の下部にある[指標]をクリックします。

   b. 指標内で、中央の列の上部にある「グリッド」リンクを選択します。

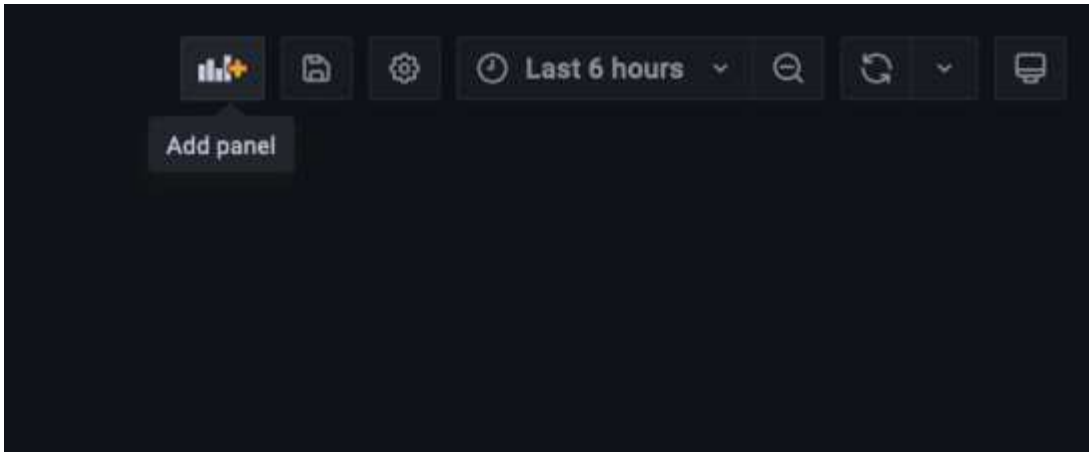


   c. グリッドダッシュボードで、「Storage Used - Object Metadata」パネルを選択します。メニューをドロップダウンするには、パネルタイトルの小さな下向き矢印と末尾をクリックします。このメニューから「Inspect」と「Panel JSON」を選択します。



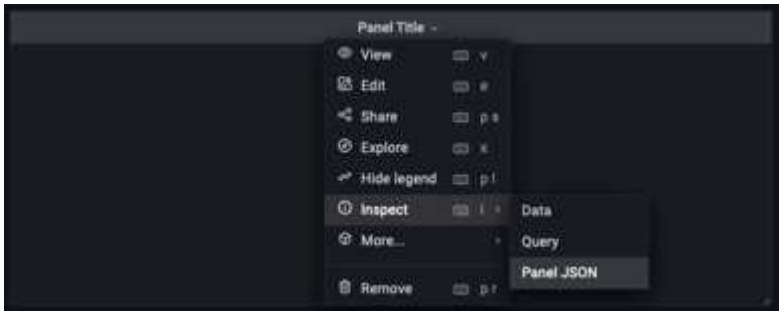
   d. JSONコードをコピーしてウィンドウを閉じます。

e. 新しいダッシュボードで、アイコンをクリックして新しいパネルを追加します。

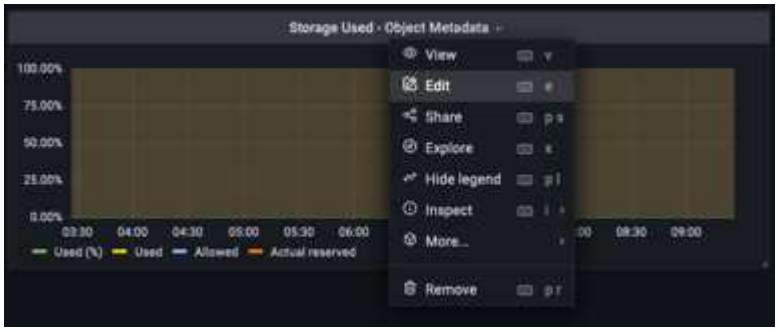


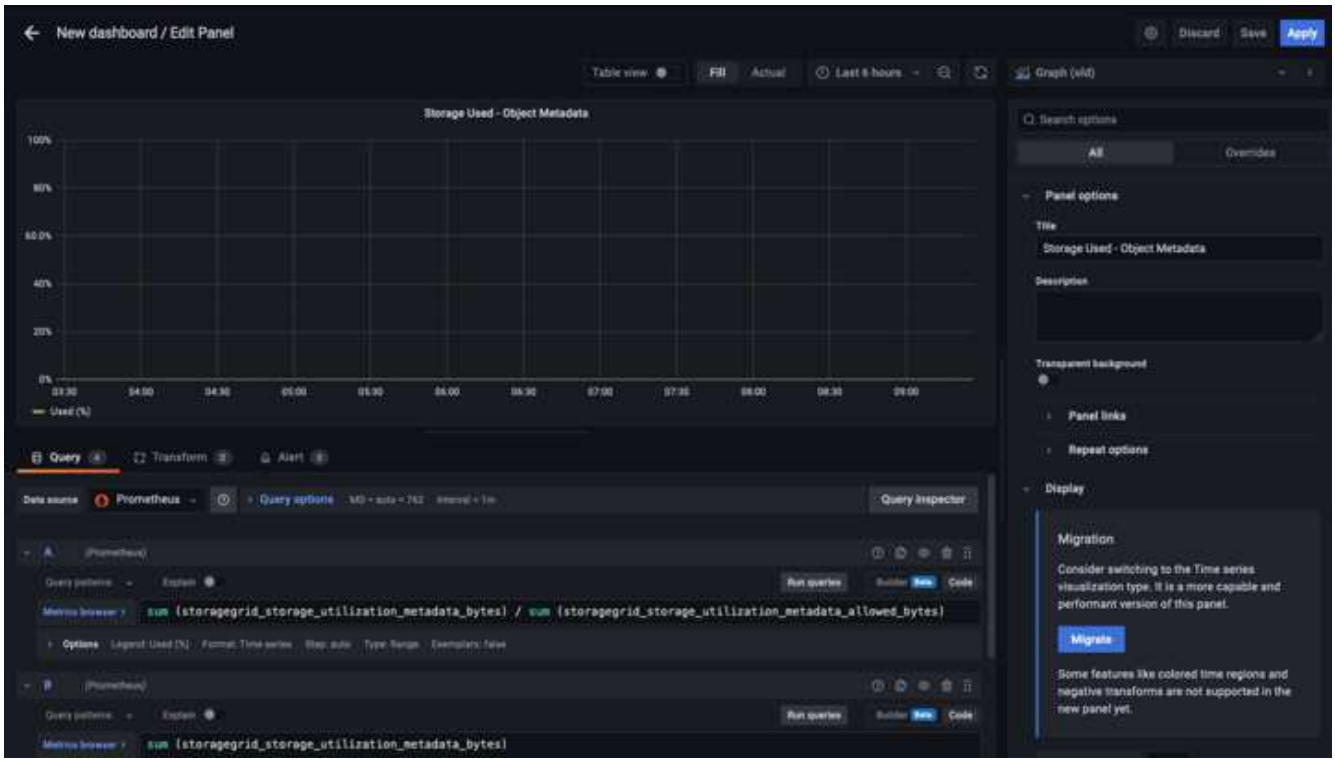
f. 変更を加えずに新しいパネルを適用します

g. StorageGRID パネルと同様に、JSONを確認します。JSONコードをすべて削除し、StorageGRID パネルからコピーしたコードに置き換えます。



h. 新しいパネルを編集すると、右側に「移行」ボタンを含む移行メッセージが表示されます。ボタンをクリックして、[適用]ボタンをクリックします。





8. すべてのパネルを所定の位置に配置し、必要に応じて構成したら、右上のディスクアイコンをクリックしてダッシュボードを保存し、名前を付けます。

まとめ

カスタマイズ可能なデータ保持機能とストレージ容量を備えたPrometheusサーバを導入しました。そのため、運用に最も関連性の高い指標を使用して独自のダッシュボードを構築し続けることができます。で収集されたPrometheus指標の詳細を確認できます "StorageGRID のドキュメント"。

アロンクライン著

Datadog SNMP構成

StorageGRID SNMPメトリクスとトラップを収集するようにDatadogを構成します。

Datadogを構成します

Datadogは、メトリクス、ビジュアライゼーション、アラートを提供する監視解決策 です。次の構成は、StorageGRID システムのローカルに配置されたUbuntu 22.04.1ホスト上のLinuxエージェントバージョン7.43.1で実装されました。

StorageGRID MIBファイルから生成されたDatadogプロファイルおよびトラップファイル

Datadogは、製品MIBファイルをSNMPメッセージのマッピングに必要なdatadog参照ファイルに変換する方法を提供します。

見つかった命令に従って生成されたDatadogトラップ解決マッピング用のこのStorageGRID YAMLファイル "こちらをご覧ください"。+このファイルを/etc/datadog-agent/conf.d/snmp.d/traps_db/+に配置します

- "トラップYAMLファイルをダウンロードします" [+]
  - * MD5チェックサム* 42e27e4210719945a46172b98c379517以降
  - * SHA256チェックサム* d0fe5c8e6ca3c902d054f854b70a85f928cba8b7c76391d356f05d2cf73b6887以降

見つかった命令に従って生成されたDatadogメトリクスマッピング用のこのStorageGRID プロファイルYAMLファイル "こちらをご覧ください"。+このファイルを/etc/datadog-agent/conf.d/snmp.d/profiles/+に配置します

- "プロファイルYAMLファイルをダウンロードします" [+]
  - * MD5チェックサム* 72bb7784f4801adda4e0c3ea77df19aa+
  - * SHA256チェックサム* b6b7fadd33063422a8bb8e39b3ead8ab349ee0229926eadc8585f0087b8cee+

メトリクスの**SNMP Datadog**構成

メトリックのSNMPの設定は、2つの方法で管理できます。自動検出を設定するには、StorageGRID システムを含むネットワークアドレス範囲を指定するか、個 々 のデバイスのIPを定義します。設定の場所は、決定内容によって異なります。自動検出は、datadogエージェントのYAMLファイルで定義されます。明示的なデバイス定義は、SNMP設定YAMLファイルで設定されます。以下に、同じStorageGRID システムのそれぞれの例を示します。

自動検出

設定は/etc/datadog-agent/datadog.yamlにあります

```
listeners:
  - name: snmp
snmp_listener:
  workers: 100  # number of workers used to discover devices concurrently
  discovery_interval: 3600  # interval between each autodiscovery in
seconds
  loader: core  # use core check implementation of SNMP integration.
recommended
  use_device_id_as_hostname: true  # recommended
  configs:
    - network_address: 10.0.0.0/24  # CIDR subnet
      snmp_version: 2
      port: 161
      community_string: 'st0r@gegrid'  # enclose with single quote
      profile: netapp-storagegrid
```

個 々 のデバイス

/etc/datadog-agent/conf.d/snmp.d/conf.yaml

```
init_config:
  loader: core  # use core check implementation of SNMP integration.
recommended
  use_device_id_as_hostname: true  # recommended
instances:
- ip_address: '10.0.0.1'
  profile: netapp-storagegrid
  community_string: 'st0r@gegrid'  # enclose with single quote
- ip_address: '10.0.0.2'
  profile: netapp-storagegrid
  community_string: 'st0r@gegrid'
- ip_address: '10.0.0.3'
  profile: netapp-storagegrid
 community_string: 'st0r@gegrid'
- ip_address: '10.0.0.4'
  profile: netapp-storagegrid
  community_string: 'st0r@gegrid'
```

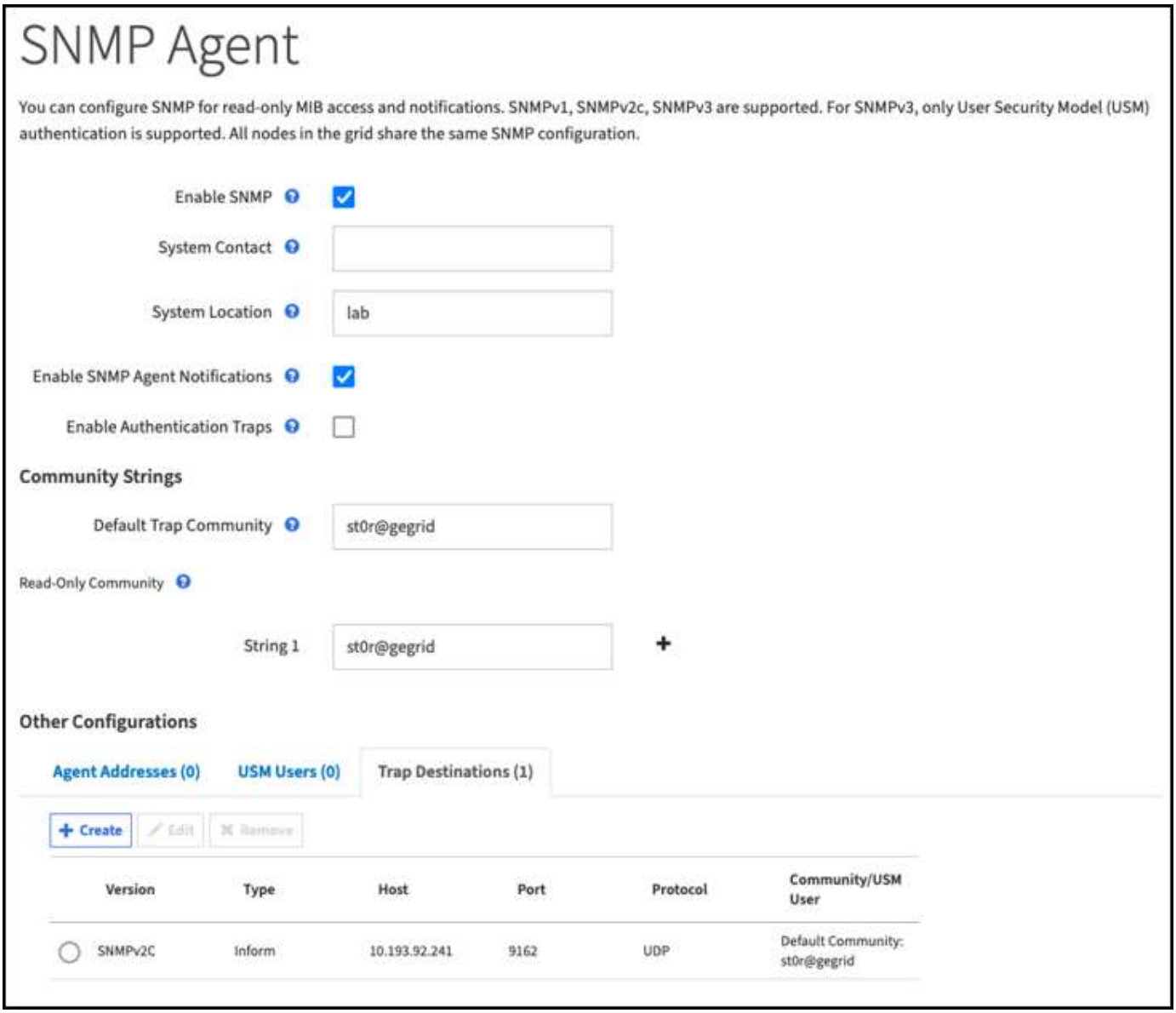
トラップの**SNMP**設定

SNMPトラップの構成は、datadog構成YAMLファイル/etc/datadog-agent/datadog.yamlで定義されています

```
network_devices:
  namespace:  # optional, defaults to “default”.
  snmp_traps:
    enabled: true
    port: 9162 # on which ports to listen for traps
    community_strings: # which community strings to allow for v2 traps
      - st0r@gegrid
```

StorageGRID の**SNMP**設定例

StorageGRID システムのSNMPエージェントは、[Configuration]タブの[Monitoring]列にあります。SNMPを有効にし、必要な情報を入力します。トラップを構成する場合は、[Traps Destinations]を選択し、トラップ構成を含むDatadogエージェントホストの宛先を作成します。



アロンクライン著

rcloneを使用して、StorageGRID 上のオブジェクトを移行、PUT、および削除します

rcloneは、S3処理用の無料のコマンドラインツールでクライアントです。rcloneを使用して、StorageGRID 上のオブジェクトデータを移行、コピー、および削除できます。rcloneには、次の例に示すように、「purge」機能を使用して空でなくてもバケットを削除する機能が含まれています。

rcloneをインストールして設定します

rcloneをワークステーションまたはサーバにインストールするには、からダウンロードします "rclone.org"。

初期設定手順

1. 設定スクリプトを実行するか、ファイルを手動で作成して、rclone構成ファイルを作成します。
2. この例では、rclone構成のリモートStorageGRID S3エンドポイントの名前にsgdemoを使用します。
   a. 設定ファイル~/.config/rclone/rclone.confを作成します

```
[sgdemo]
type = s3
provider = Other
access_key_id = ABCDEFGH123456789JKL
secret_access_key = 123456789ABCDEFGHIJKLMN0123456789PQRST+V
endpoint = sgdemo.netapp.com
```

   b. rclone configを実行します

#rclone設定

```
2023/04/13 14:22:45 NOTICE: Config file
"/root/.config/rclone/rclone.conf" not found - using defaults
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n
name> sgdemo
```

```
Option Storage.
Type of storage to configure.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value.
 1 / 1Fichier
   \ "fichier"
 2 / Alias for an existing remote
   \ "alias"
 3 / Amazon Drive
   \ "amazon cloud drive"
 4 / Amazon S3 Compliant Storage Providers including AWS,
Alibaba, Ceph, Digital Ocean, Dreamhost, IBM COS, Minio,
SeaweedFS, and Tencent COS
   \ "s3"
 5 / Backblaze B2
   \ "b2"
 6 / Better checksums for other remotes
   \ "hasher"
 7 / Box
   \ "box"
 8 / Cache a remote
   \ "cache"
 9 / Citrix Sharefile
   \ "sharefile"
10 / Compress a remote
   \ "compress"
11 / Dropbox
   \ "dropbox"
12 / Encrypt/Decrypt a remote
   \ "crypt"
13 / Enterprise File Fabric
   \ "filefabric"
14 / FTP Connection
```

```
   \ "ftp"
15 / Google Cloud Storage (this is not Google Drive)
   \ "google cloud storage"
16 / Google Drive
   \ "drive"
17 / Google Photos
   \ "google photos"
18 / Hadoop distributed file system
   \ "hdfs"
19 / Hubic
   \ "hubic"
20 / In memory object storage system.
   \ "memory"
21 / Jottacloud
   \ "jottacloud"
22 / Koofr
   \ "koofr"
23 / Local Disk
   \ "local"
24 / Mail.ru Cloud
   \ "mailru"
25 / Mega
   \ "mega"
26 / Microsoft Azure Blob Storage
   \ "azureblob"
27 / Microsoft OneDrive
   \ "onedrive"
28 / OpenDrive
   \ "opendrive"
29 / OpenStack Swift (Rackspace Cloud Files, Memset Memstore,
OVH)
   \ "swift"
30 / Pcloud
   \ "pcloud"
31 / Put.io
   \ "putio"
32 / QingCloud Object Storage
   \ "qingstor"
33 / SSH/SFTP Connection
   \ "sftp"
34 / Sia Decentralized Cloud
   \ "sia"
35 / Sugarsync
   \ "sugarsync"
36 / Tardigrade Decentralized Cloud Storage
   \ "tardigrade"
```

```
37 / Transparently chunk/split large files
   \ "chunker"
38 / Union merges the contents of several upstream fs
   \ "union"
39 / Uptobox
   \ "uptobox"
40 / Webdav
   \ "webdav"
41 / Yandex Disk
   \ "yandex"
42 / Zoho
   \ "zoho"
43 / http Connection
   \ "http"
44 / premiumize.me
   \ "premiumizeme"
45 / seafile
   \ "seafile"
```

```
Storage> 4
```

```
Option provider.
Choose your S3 provider.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value.
 1 / Amazon Web Services (AWS) S3
   \ "AWS"
 2 / Alibaba Cloud Object Storage System (OSS) formerly Aliyun
   \ "Alibaba"
 3 / Ceph Object Storage
   \ "Ceph"
 4 / Digital Ocean Spaces
   \ "DigitalOcean"
 5 / Dreamhost DreamObjects
   \ "Dreamhost"
 6 / IBM COS S3
   \ "IBMCOS"
 7 / Minio Object Storage
   \ "Minio"
 8 / Netease Object Storage (NOS)
   \ "Netease"
 9 / Scaleway Object Storage
   \ "Scaleway"
10 / SeaweedFS S3
   \ "SeaweedFS"
11 / StackPath Object Storage
   \ "StackPath"
12 / Tencent Cloud Object Storage (COS)
   \ "TencentCOS"
13 / Wasabi Object Storage
   \ "Wasabi"
14 / Any other S3 compatible provider
   \ "Other"
provider> 14
```

```
Option env_auth.
Get AWS credentials from runtime (environment variables or
EC2/ECS meta data if no env vars).
Only applies if access_key_id and secret_access_key is blank.
Enter a boolean value (true or false). Press Enter for the
default ("false").
Choose a number from below, or type in your own value.
 1 / Enter AWS credentials in the next step.
   \ "false"
 2 / Get AWS credentials from the environment (env vars or IAM).
   \ "true"
env_auth> 1
```

```
Option access_key_id.
AWS Access Key ID.
Leave blank for anonymous access or runtime credentials.
Enter a string value. Press Enter for the default ("").
access_key_id> ABCDEFGH123456789JKL
```

```
Option secret_access_key.
AWS Secret Access Key (password).
Leave blank for anonymous access or runtime credentials.
Enter a string value. Press Enter for the default ("").
secret_access_key> 123456789ABCDEFGHIJKLMN0123456789PQRST+V
```

```
Option region.
Region to connect to.
Leave blank if you are using an S3 clone and you don't have a
region.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value.
   / Use this if unsure.
 1 | Will use v4 signatures and an empty region.
   \ ""
   / Use this only if v4 signatures don't work.
 2 | E.g. pre Jewel/v10 CEPH.
   \ "other-v2-signature"
region> 1
```

```
Option endpoint.
Endpoint for S3 API.
Required when using an S3 clone.
Enter a string value. Press Enter for the default ("").
endpoint> sgdemo.netapp.com
```

```
Option location_constraint.
Location constraint - must be set to match the Region.
Leave blank if not sure. Used when creating buckets only.
Enter a string value. Press Enter for the default ("").
location_constraint>
```

```
Option acl.
Canned ACL used when creating buckets and storing or copying
objects.
This ACL is used for creating objects and if bucket_acl isn't
set, for creating buckets too.
For more info visit
https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-
overview.html#canned-acl
Note that this ACL is applied when server-side copying objects as
S3
doesn't copy the ACL from the source but rather writes a fresh
one.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value.
   / Owner gets FULL_CONTROL.
 1 | No one else has access rights (default).
   \ "private"
   / Owner gets FULL_CONTROL.
 2 | The AllUsers group gets READ access.
   \ "public-read"
   / Owner gets FULL_CONTROL.
 3 | The AllUsers group gets READ and WRITE access.
   | Granting this on a bucket is generally not recommended.
   \ "public-read-write"
   / Owner gets FULL_CONTROL.
 4 | The AuthenticatedUsers group gets READ access.
   \ "authenticated-read"
   / Object owner gets FULL_CONTROL.
 5 | Bucket owner gets READ access.
   | If you specify this canned ACL when creating a bucket,
Amazon S3 ignores it.
   \ "bucket-owner-read"
   / Both the object owner and the bucket owner get FULL_CONTROL
over the object.
 6 | If you specify this canned ACL when creating a bucket,
Amazon S3 ignores it.
   \ "bucket-owner-full-control"
acl>
```

```
Edit advanced config?
y) Yes
n) No (default)
y/n> n
```

```
--------------------
[sgdemo]
type = s3
provider = Other
access_key_id = ABCDEFGH123456789JKL
secret_access_key = 123456789ABCDEFGHIJKLMN0123456789PQRST+V
endpoint = sgdemo.netapp.com:443
--------------------
y) Yes this is OK (default)
e) Edit this remote
d) Delete this remote
y/e/d>
```

```
Current remotes:
```

```
Name                 Type
====                 ====
sgdemo               s3
```

```
e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> q
```

基本的なコマンドの例

- バケットを作成：

  `rclone mkdir remote:bucket`

  #rclone mkdir sgdemo：test01

  SSL証明書を無視する必要がある場合は、--no-check-certificateを使用します。

- すべてのバケットを表示：

```
rclone lsd remote:
```

#rclone lsd sgdemo：

- 特定のバケット内のオブジェクトをリストします。

```
rclone ls remote:bucket
```

rclone ls sgdemo：test01

```
     65536 TestObject.0
     65536 TestObject.1
     65536 TestObject.10
     65536 TestObject.12
     65536 TestObject.13
     65536 TestObject.14
     65536 TestObject.15
     65536 TestObject.16
     65536 TestObject.17
     65536 TestObject.18
     65536 TestObject.2
     65536 TestObject.3
     65536 TestObject.5
     65536 TestObject.6
     65536 TestObject.7
     65536 TestObject.8
     65536 TestObject.9
   33554432 bigobj
       102 key.json
        47 locked01.txt
 4294967296 sequential-read.0.0
        15 test.txt
       116 version.txt
```

- バケットを削除：

```
rclone rmdir remote:bucket
```

#rclone rmdir sgdemo：test02

- オブジェクトを置きなさい:

```
rclone copy filename remote:bucket
```

#rclone copy ~/test/ testfile.txt sgdemo：test01

- オブジェクトを取得：

```
rclone copy remote:bucket/objectname filename
```

#rclone copy sgdemo：test01 / testfile.txt ~/test/ testfileS3.txt

- オブジェクトを削除：

```
rclone delete remote:bucket/objectname
```

#rclone delete sgdemo：test01 / testfile.txt

- バケット内のオブジェクトの移行

```
rclone sync source:bucket destination:bucket --progress

rclone sync source_directory destination:bucket --progress
```

#rclone sync sgdemo：test01 sgdemo：clone01 — progress

```
Transferred:        4.032 GiB / 4.032 GiB, 100%, 95.484 KiB/s, ETA
0s
Transferred:           22 / 22, 100%
Elapsed time:       1m4.2s
```

— progressまたは- Pを使用して、タスクの進行状況を表示します。それ以外の場合、出力はありません。

- バケットとすべてのオブジェクトコンテンツを削除する

```
rclone purge remote:bucket --progress
```

#rclone purge sgdemo：test01 — progress

```
Transferred:              0 B / 0 B, -, 0 B/s, ETA -
Checks:                46 / 46, 100%
Deleted:               23 (files), 1 (dirs)
Elapsed time:        10.2s
```

rclone ls sgdemo：test01

```
2023/04/14 09:40:51 Failed to ls: directory not found
```

ジークフリート・ヘップとアロン・クライン著_

Veeam Backup & Replicationを使用した導入に関するStorageGRIDのベストプラクティス

このガイドでは、NetApp StorageGRIDの構成と、Veeam Backup & Replicationの一部を中心に説明します。本ドキュメントは、Linuxシステムに精通し、Veeam Backup & Replicationと組み合わせてNetApp StorageGRIDシステムの保守または実装を担当するストレージ管理者およびネットワーク管理者を対象としています。

概要

ストレージ管理者は、可用性、迅速なリカバリの目標を達成し、ニーズに合わせて拡張し、データの長期保存に関するポリシーを自動化するソリューションを使用して、データの増加を管理したいと考えています。これらのソリューションは、損失や悪意のある攻撃からも保護する必要があります。VeeamとNetAppは提携して、オンプレミスのオブジェクトストレージ向けのVeeam Backup & RecoveryとNetApp StorageGRIDを組み合わせたデータ保護解決策を作成しました。

VeeamとNetApp StorageGRIDが連携して動作する使いやすい解決策を提供することで、急速なデータ量の増大や世界的な規制強化のニーズに対応できます。クラウドベースのオブジェクトストレージは、耐障害性、拡張性、運用効率、コスト効率に優れていることで知られており、バックアップのターゲットとして最適です。本ドキュメントでは、Veeam Backup解決策およびStorageGRIDシステムの構成に関するガイダンスと推奨事項を提供します。

Veeamのオブジェクトワークロードによって、小規模オブジェクトのPUT、DELETE、LIST処理が同時に多数作成されます。書き換えや削除の防止を有効にすると、保持期間の設定やバージョンの表示に関する要求がオブジェクトストアに追加されます。バックアップジョブのプロセスでは、日次変更のためにオブジェクトが書き込まれます。その後、新しい書き込みが完了すると、バックアップの保持ポリシーに基づいてオブジェクトが削除されます。バックアップジョブのスケジュールは、ほとんどの場合重複します。その結果、バックアップウィンドウの大部分がオブジェクトストアに50分の50のPUT / DELETEワークロードで構成されます。タスクスロットの設定を使用して同時処理数をVeeamで調整し、バックアップジョブのブロックサイズを増やしてオブジェクトサイズを増やし、複数オブジェクトの削除要求に含まれるオブジェクト数を減らします。また、ジョブを完了する最大期間を選択することで、解決策のパフォーマンスとコストが最適化されます。

次の製品ドキュメントを参照してください： "Veeam Backup Replication" および "StorageGRID" 始める前に。Veeamには、StorageGRID 解決策 をサイジングする前に使用する必要があるVeeamインフラのサイジングと容量の要件を把握するための計算ツールが用意されています。Veeam Ready ProgramのWebサイトで、Veeamとネットアップによる検証済みの構成について "Veeam Readyのオブジェクト、オブジェクトの変更不可、リポジトリ"。

Veeam構成

推奨バージョン

常に最新の状態に保ち、Veeam Backup & Replication 12システムの最新のホットフィックスを適用することをお勧めします。現在、少なくともVeeamパッチP20230718のインストールを推奨しています。

S3リポジトリ設定

スケールアウトバックアップリポジトリ（SOBR）は、S3オブジェクトストレージの大容量階層です。大容量階層はプライマリリポジトリを拡張したもので、データ保持期間が長くなり、ストレージ解決策が低コストになります。Veeamには、S3 Object Lock APIを通じて不変性を提供する機能があります。Veeam 12では、スケールアウトリポジトリで複数のバケットを使用できます。StorageGRIDでは、1つのバケット内のオブジェクト数や容量に制限はありません。複数のバケットを使用すると、オブジェクトのバックアップデータがペタバイト規模になる可能性がある非常に大規模なデータセットをバックアップする際のパフォーマンスが向上する可能性があります。

特定の解決策のサイジングと要件によっては、同時に実行できるタスクを制限する必要があります。デフォルト設定では、CPUコアごとに1つのリポジトリタスクスロットを指定し、タスクスロットごとに最大64の同時タスクスロットを指定します。たとえば、サーバに2つのCPUコアがある場合、オブジェクトストアには合計128個の同時スレッドが使用されます。これには、PUT、GET、およびBATCH Deleteが含まれます。タスクスロットに控えめな制限を選択して開始し、Veeamバックアップが新しいバックアップの安定した状態と期限切れになるバックアップ・データに達したら、この値を調整することをお勧めします。NetAppアカウントチームと協力して、希望する時間枠とパフォーマンスに合わせてStorageGRIDシステムを適切にサイジングしてください。最適な解決策を提供するには、タスクスロットの数とスロットあたりのタスクの制限を調整する必要がある場合があります。

バックアップジョブの設定

Veeamバックアップジョブでは、さまざまなブロックサイズオプションを設定できますが、これらは慎重に検討する必要があります。デフォルトのブロックサイズは1MBで、Veeamの圧縮機能と重複排除機能を使用すると、最初のフルバックアップでは約500KB、増分ジョブでは100200KBのオブジェクトが作成されます。バックアップブロックサイズを大きくすることで、パフォーマンスを大幅に向上し、オブジェクトストレージの要件を縮小できます。ブロックサイズが大きいほどオブジェクトストアのパフォーマンスは大幅に向上しますが、ストレージ効率のパフォーマンスが低下するため、プライマリストレージの容量要件が増大する可能性があります。バックアップジョブのブロックサイズを4MBに設定することを推奨します。この場合、フルバックアップ用に約2MBのオブジェクトが作成され、増分バックアップ用に700KB1MBのオブジェクトサイズが作成されます。お客様は、8 MBのブロックサイズを使用してバックアップジョブを構成することも検討できます。これは、Veeamサポートの支援を受けて有効にすることができます。

変更不可のバックアップの実装では、オブジェクトストアのS3オブジェクトロックが使用されます。immutabilityオプションを指定すると、オブジェクトに対するリストおよび保持の更新要求がオブジェクトストアに対して生成される回数が増加します。

バックアップの保持期間が終了すると、バックアップジョブによってオブジェクトの削除が処理されます。Veeamは、1回の要求につき1、000個のオブジェクトを含む複数のオブジェクトの削除要求で、オブジェクトストアに削除要求を送信します。小規模なソリューションの場合は、リクエストあたりのオブジェクト数を減らすために調整が必要になることがあります。この値を小さくすると、削除要求がStorageGRIDシステム内のノードに均等に分散されるというメリットもあります。複数オブジェクトの削除制限を設定する場合は、次の表の値を開始点として使用することをお勧めします。表の値に選択したアプライアンスタイプのノード数

を掛けて、Veeamの設定値を取得します。この値が1000以上の場合、デフォルト値を調整する必要はありません。この値を調整する必要がある場合は、Veeamサポートに連絡して変更を行ってください。

アプライアンスモデル	ノードあたりの**S3MultiObjectDeleteLimit**
SG5712	34
SG5760	七五
SG6060 の設計	200です

お客様固有のニーズに基づいた推奨構成については、NetAppアカウントチームにお問い合わせください。Veeamの設定に関する推奨事項は次のとおりです。

- バックアップジョブのブロックサイズ= 4MB
- SOBRタスクスロット制限=2-16
- 複数オブジェクトの削除制限= 34-1000

StorageGRID構成

推奨バージョン

Veeam環境に推奨されるバージョンは、最新のホットフィックスが適用されたNetApp StorageGRID 11.6または11.7です。StorageGRID 11.6.0.11および11.7.0.4では、Veeamのワークロードに役立つ最適化機能が多数導入されました。常に最新の状態に保ち、StorageGRIDシステムに最新のホットフィックスを適用することを推奨します。

ロードバランサと**S3**エンドポイントの設定

Veeamでは、エンドポイントの接続にHTTPSのみを使用する必要があります。暗号化されていない接続はVeeamではサポートされていません。SSL証明書には、自己署名証明書、信頼されたプライベート認証局、または信頼されたパブリック認証局を使用できます。S3リポジトリへの継続的なアクセスを確保するために、HA構成で少なくとも2つのロードバランサを使用することを推奨します。ロードバランサには、すべての管理ノードとゲートウェイノードに配置されるStorageGRID提供の統合ロードバランササービス、またはF5、Kemp、HAProxy、Loadbalanacer.orgなどのサードパーティの解決策を使用できます。 StorageGRIDロードバランサを使用すると、Veeamのワークロードに優先順位を付けたり、StorageGRIDシステムの優先順位の高いワークロードに影響しないようにVeeamを制限したりできるトラフィック分類機能（QoSルール）を設定できます。

S3 バケット

StorageGRIDは、セキュアなマルチテナントストレージシステムです。Veeamワークロード専用のテナントを作成することを推奨します。ストレージクォータはオプションで割り当てることができます。ベストプラクティスとして、「独自のアイデンティティソースを使用する」を有効にします。テナントのroot管理ユーザを適切なパスワードで保護します。Veeam Backup 12では、S3バケットに対して強い整合性が必要です。StorageGRIDには、バケットレベルで設定できる複数の整合性オプションが用意されています。Veeamが複数の場所のデータにアクセスするマルチサイト環境の場合は、[strong-global]を選択します。Veeamのバックアップとリストアを単一サイトでのみ実行する場合は、整合性レベルを「strong-site」に設定する必要があります。バケットの整合性レベルの詳細については、 "ドキュメント"。Veeamの書き換え不可のバックアップにStorageGRIDを使用するには、S3オブジェクトロックをグローバルに有効にし、バケットの作成時にバケットで設定する必要があります。

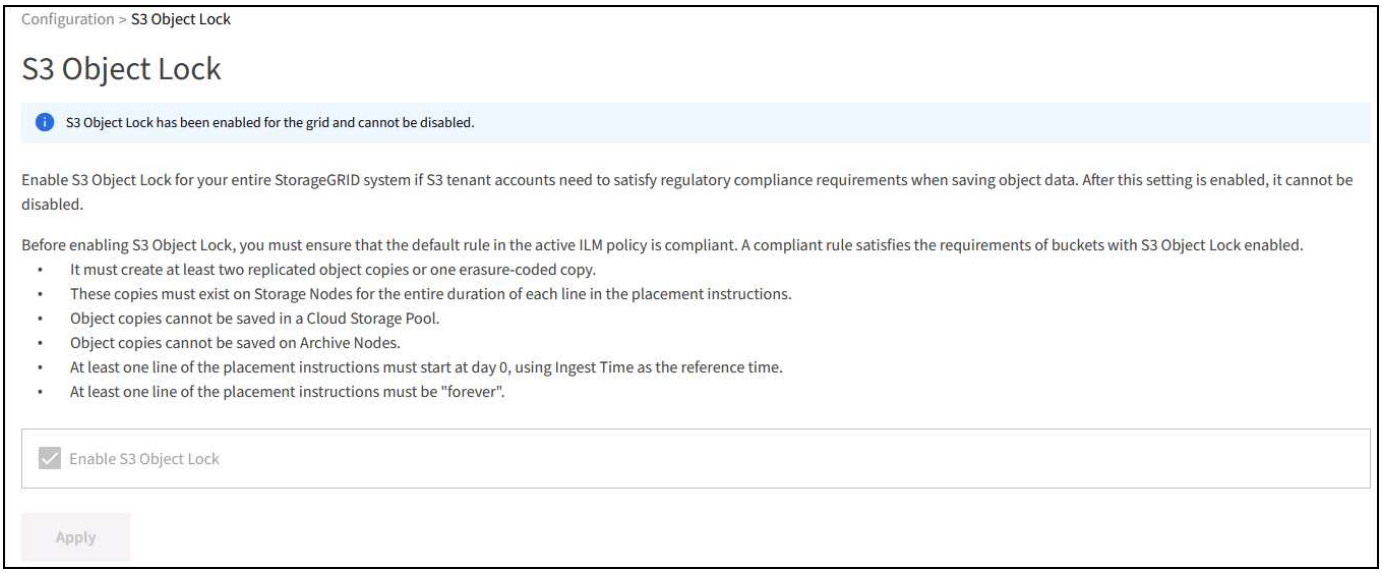
ライフサイクル管理

StorageGRIDは、レプリケーションとイレイジャーコーディングをサポートして、StorageGRIDのノードとサイト全体でオブジェクトレベルの保護を実現します。イレイジャーコーディングには、オブジェクトサイズが200KB以上必要です。Veeamのデフォルトのブロックサイズである1MBで作成されるオブジェクトサイズは、VeeamのStorage Efficiency機能と比較して、この200KBの推奨最小サイズよりも小さくなることがあります。解決策のパフォーマンスを高めるために、サイト間の接続が十分でない場合やStorageGRIDシステムの帯域幅が制限されない場合を除き、複数のサイトにまたがるイレイジャーコーディングプロファイルを使用することは推奨されません。マルチサイトStorageGRIDシステムでは、各サイトにコピーを1つ格納するようにILMルールを設定できます。データの保持性を最大限に高めるために、各サイトにイレイジャーコーディングコピーを格納するルールを設定できます。このワークロードには、Veeam Backupサーバのローカルコピーを2つ使用することを推奨します。

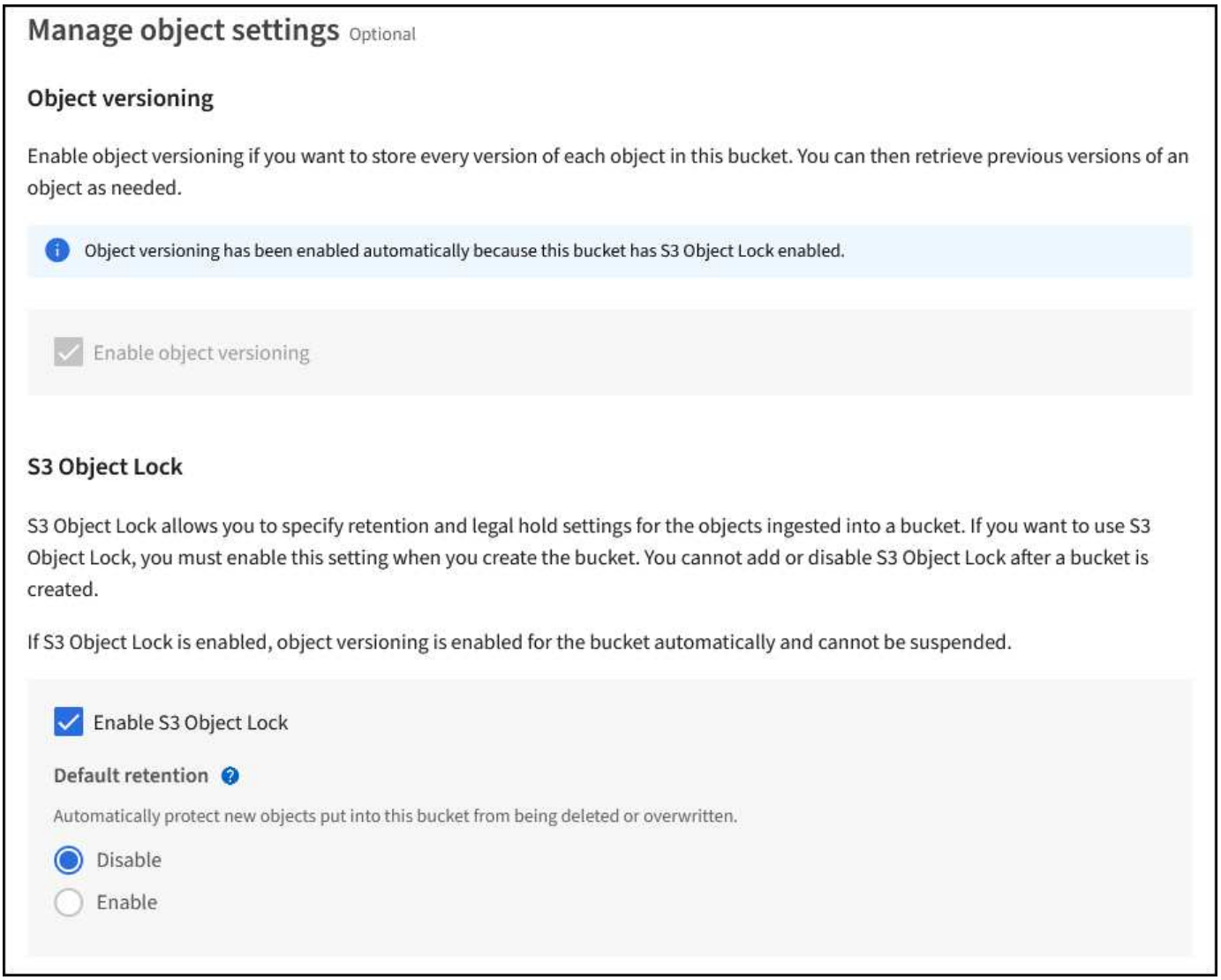
導入のキーポイント

StorageGRID

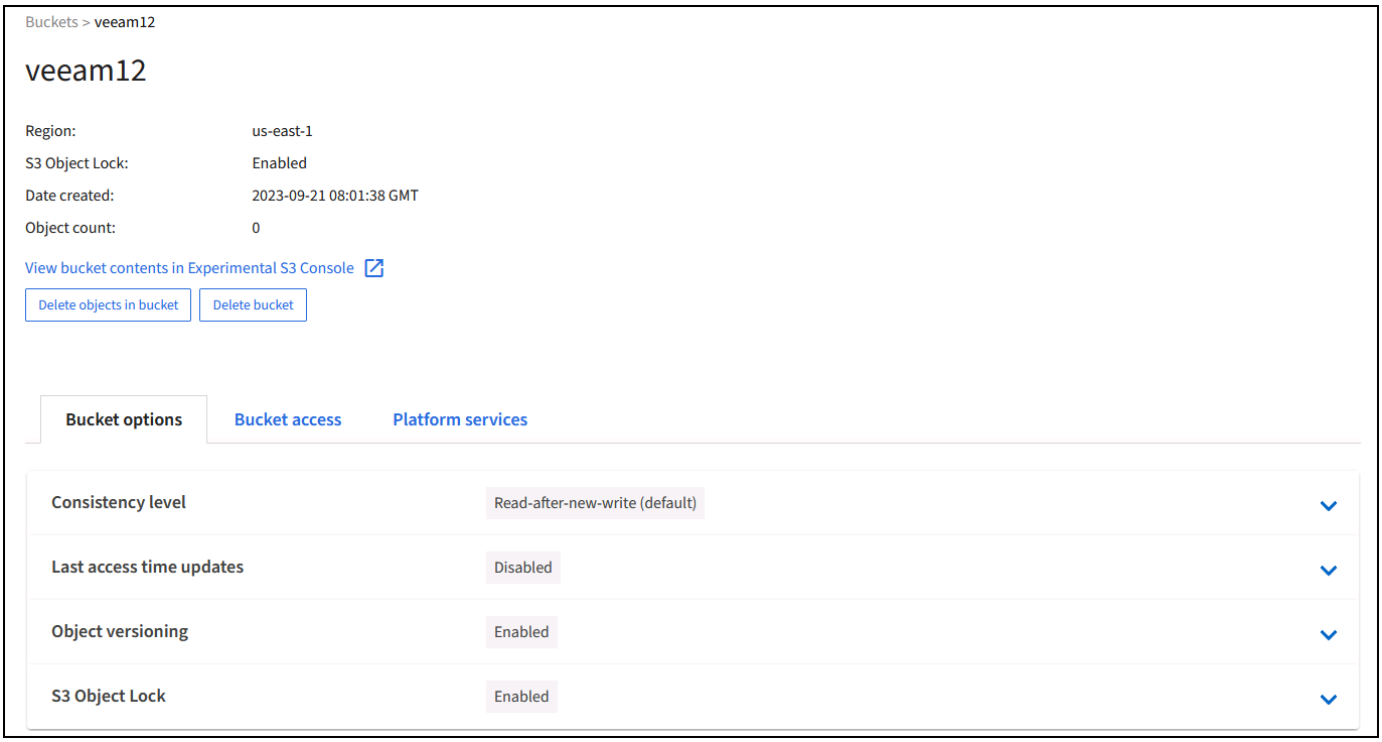
不変性が必要な場合は、StorageGRIDシステムでオブジェクトロックが有効になっていることを確認します。管理UIの[Configuration]/[S3][Object Lock]にあるオプションを選択します。



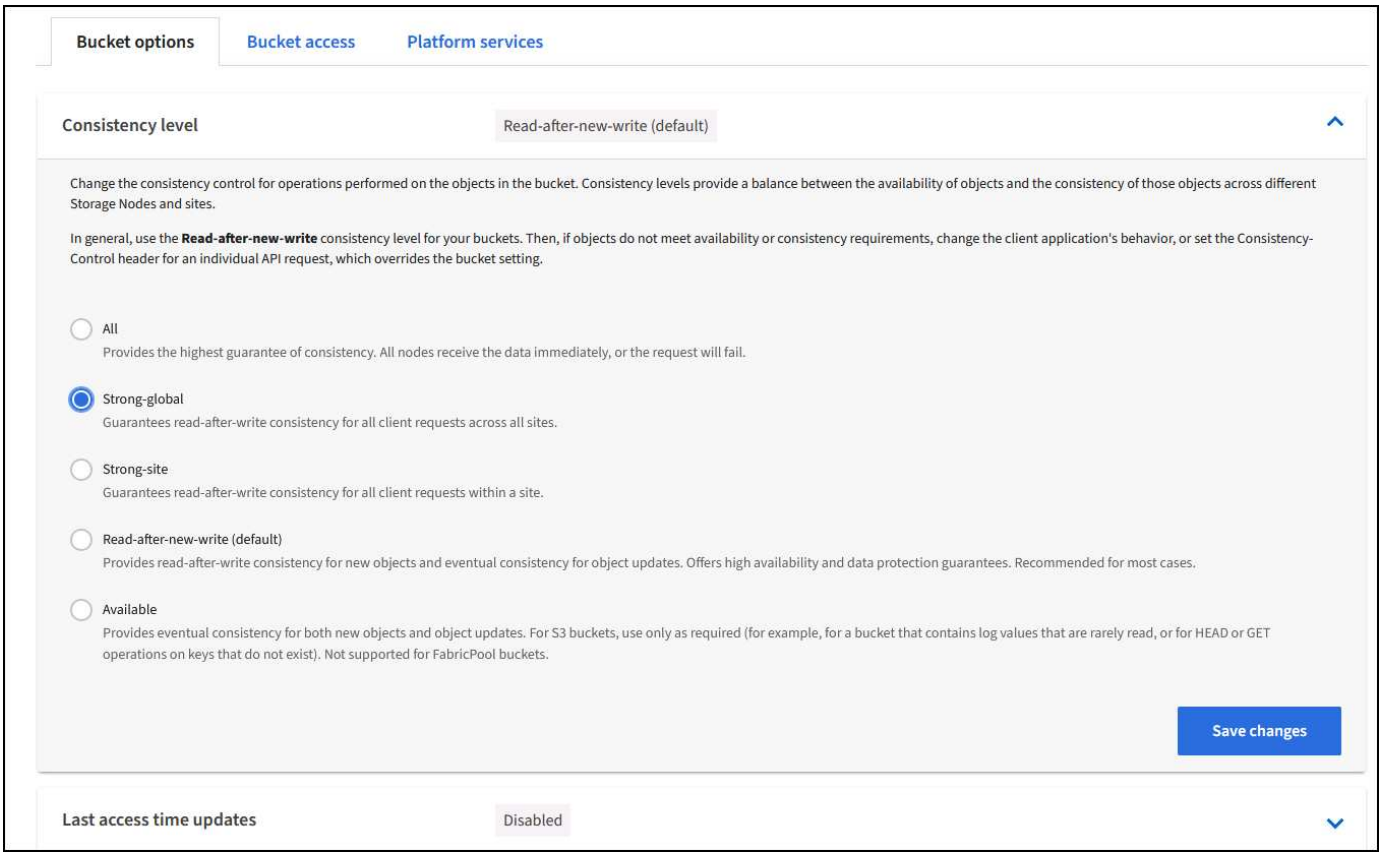
バケットを変更不可のバックアップに使用する場合は、バケットの作成時に[Enable S3 Object Lock]を選択します。これにより、バケットのバージョン管理が自動的に有効になります。オブジェクト保持期間はVeeamによって明示的に設定されるため、デフォルトの保持期間は無効のままにします。Veeamで変更不可のバックアップが作成されていない場合は、[Versioning]と[S3 Object Lock]を選択しないでください。



バケットが作成されたら、作成したバケットの詳細ページに移動します。整合性レベルを選択します。



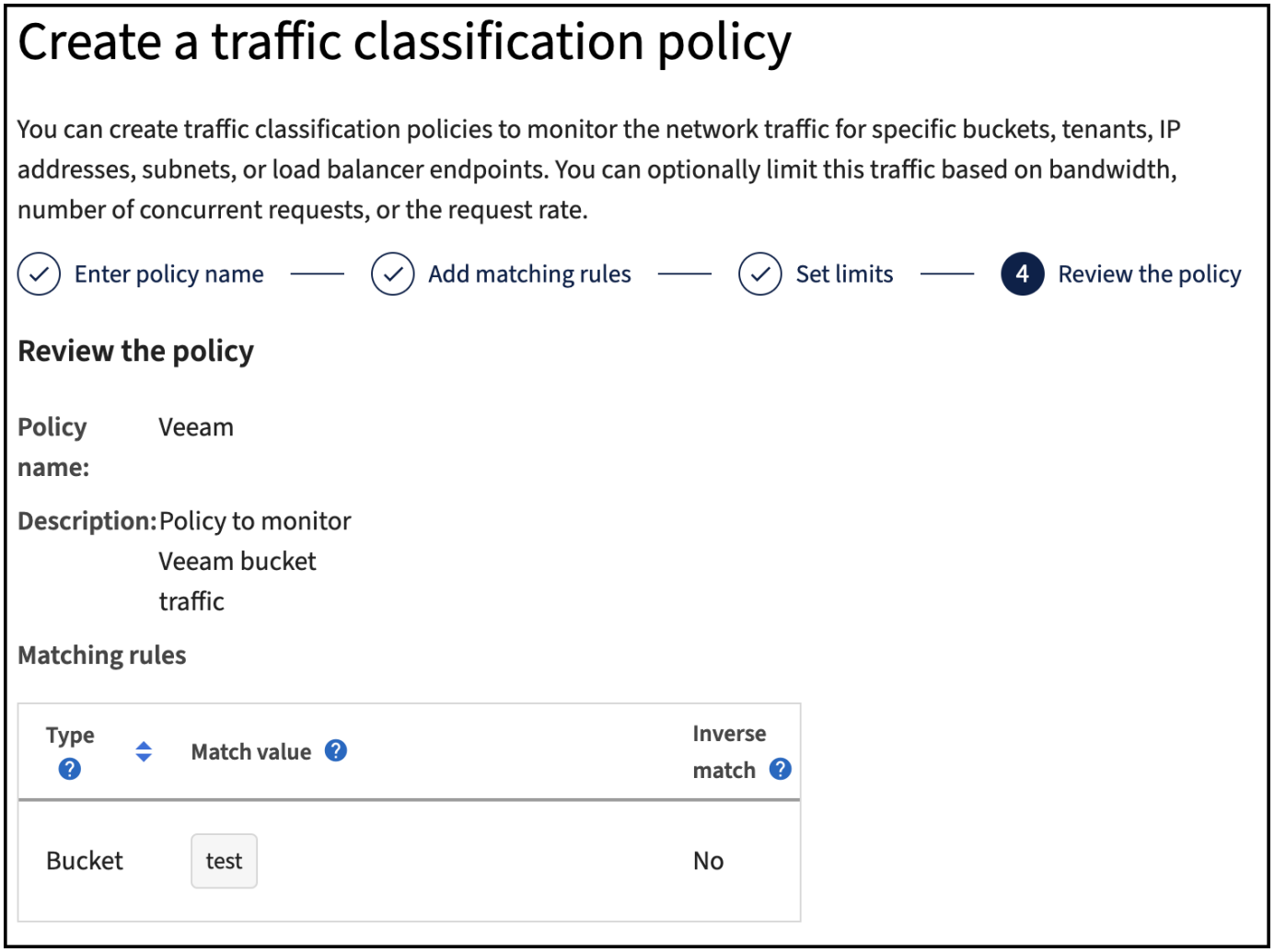
Veeamでは、S3バケットに対して強力な整合性が必要です。そのため、Veeamが複数の場所からデータにアクセスするマルチサイト環境の場合は、「strong-global」を選択します。Veeamのバックアップとリストアを単一サイトでのみ実行する場合は、整合性レベルを「strong-site」に設定する必要があります。変更を保存します。



StorageGRIDは、すべての管理ノードおよび専用のゲートウェイノードで統合されたロードバランササービス

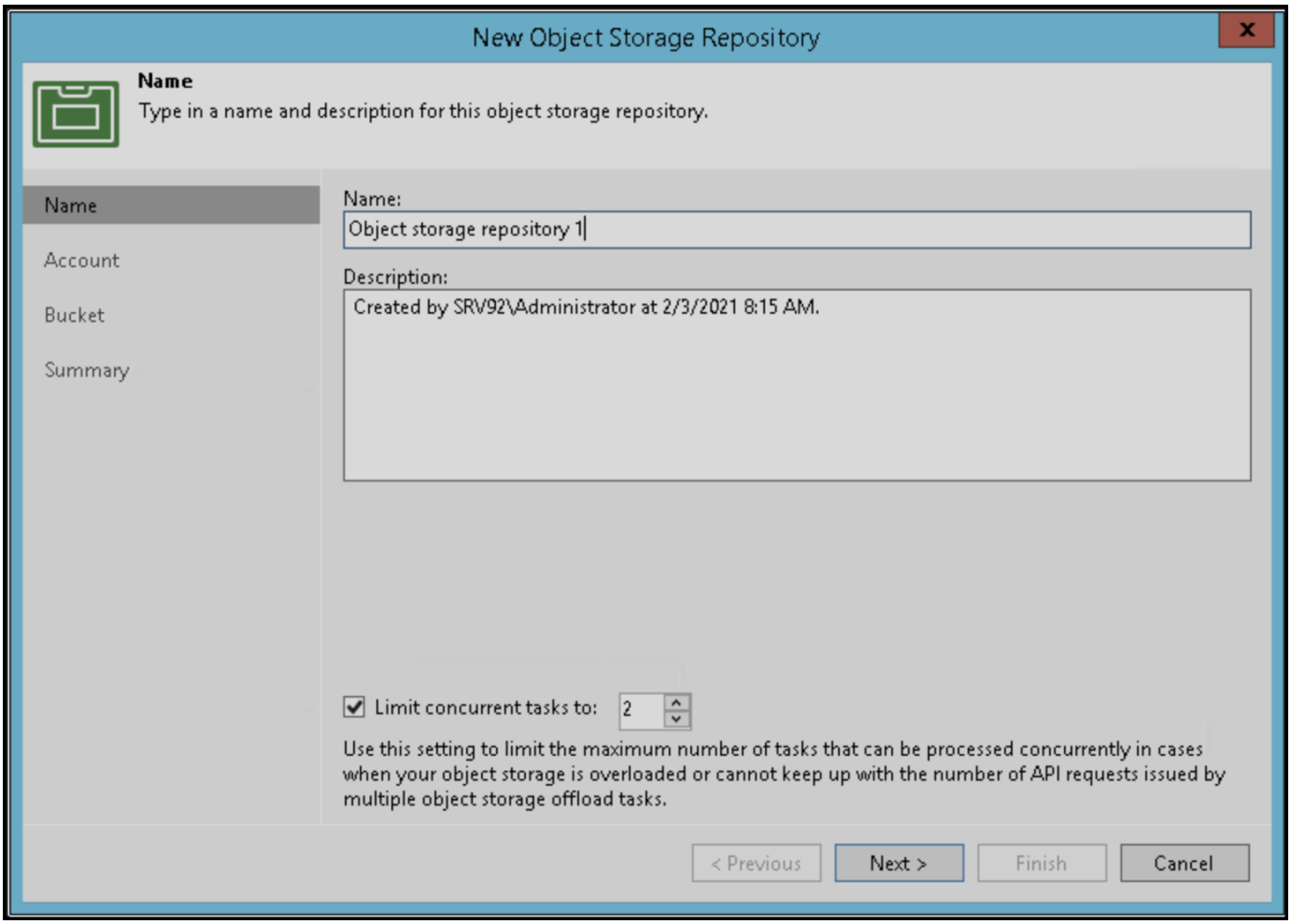
を提供します。このロードバランサを使用する多くの利点の1つは、トラフィック分類ポリシー（QoS）を設定できることです。主に、他のクライアントワークロードへのアプリケーションの影響を制限したり、他のクライアントワークロードよりもワークロードを優先したりするために使用されますが、監視に役立つ追加の指標収集のボーナスも提供します。

[Configuration]タブで、[Traffic Classification]を選択し、新しいポリシーを作成します。ルールに名前を付け、タイプとしてバケットまたはテナントを選択します。バケットまたはテナントの名前を入力します。QoSが必要な場合は制限を設定しますが、ほとんどの実装では、制限を設定しないでください。

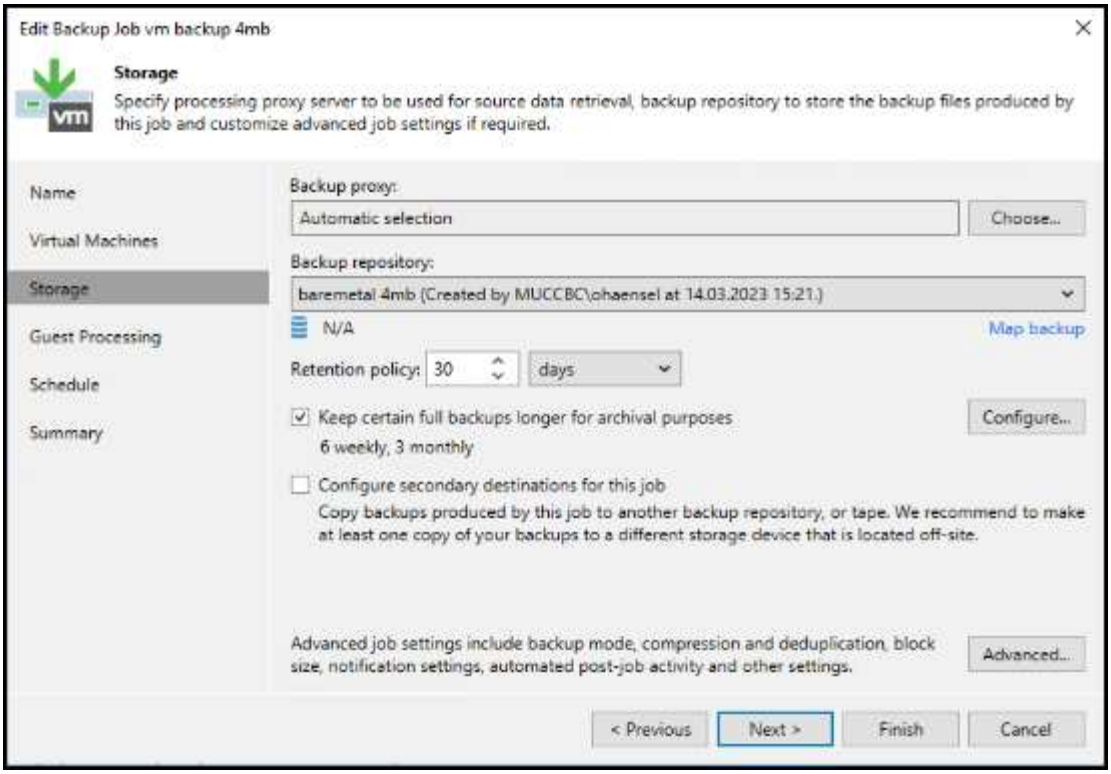


Veeamの統合によって

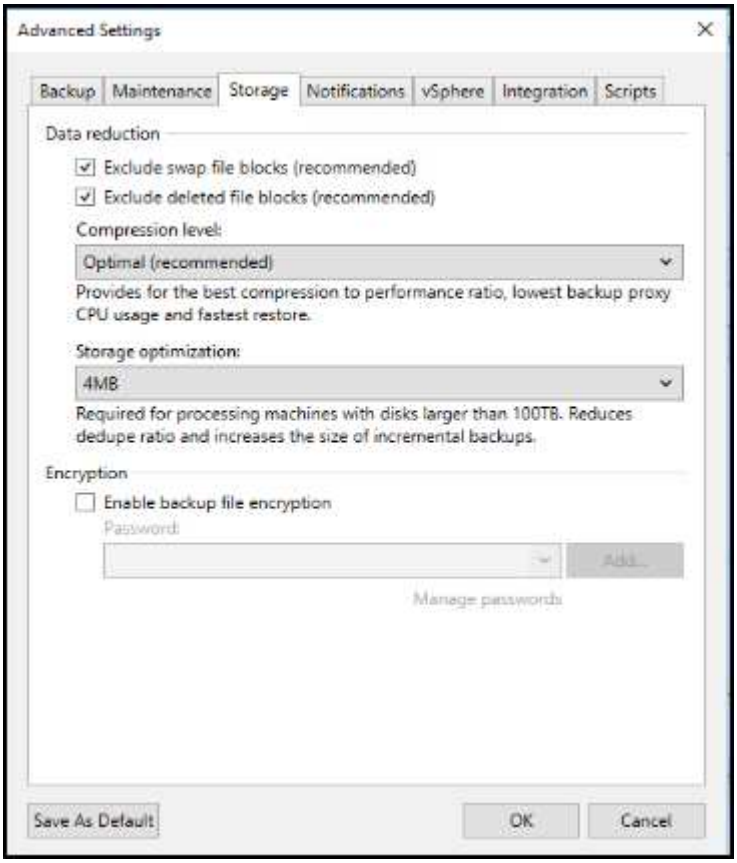
StorageGRIDアプライアンスのモデルと数によっては、バケットで同時に実行できる処理数の制限を選択して設定する必要があります。



Veeamコンソールのバックアップジョブ設定に関するVeeamのドキュメントに従って、ウィザードを開始します。VMを追加したら、SOBRリポジトリを選択します。



[詳細設定]をクリックし、ストレージ最適化設定を4 MB以上に変更します。圧縮機能と重複排除機能を有効にします。要件に応じてゲスト設定を変更し、バックアップジョブのスケジュールを設定します。



StorageGRID の監視

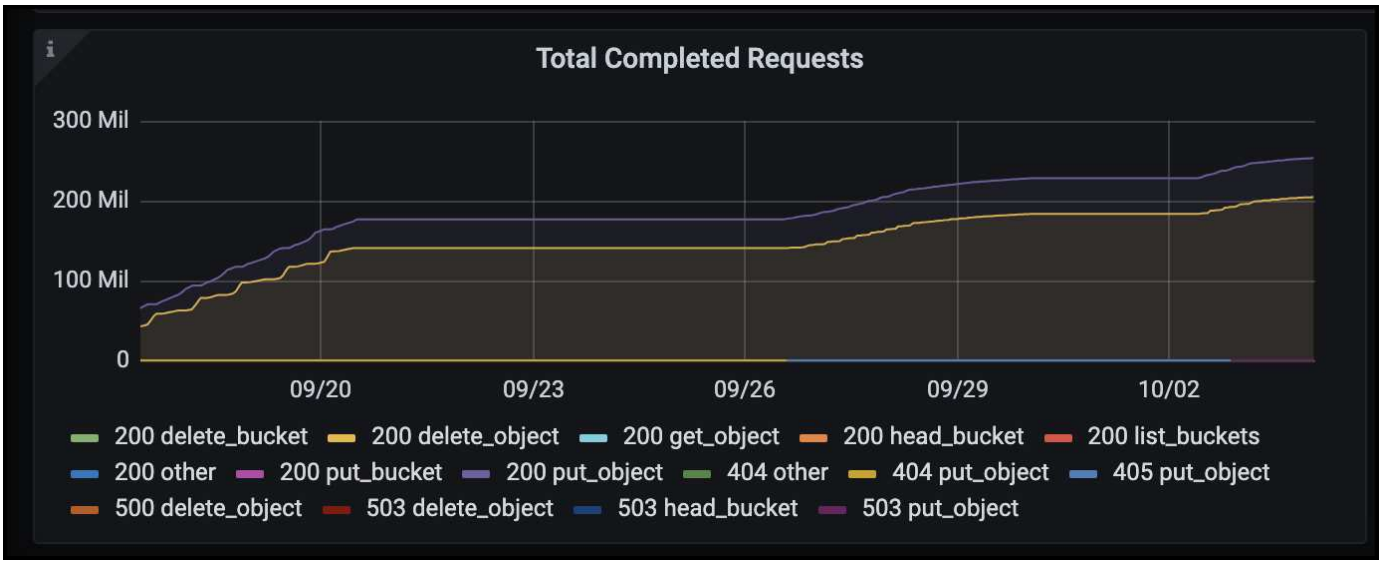
VeeamとStorageGRIDの連携によるパフォーマンスの全体像を把握するには、最初のバックアップの保持期限が切れるまで待つ必要があります。これまで、Veeamのワークロードは主にPUT処理で構成され、削除は行われていませんでした。バックアップデータの有効期限が近づいてクリーンアップを実行すると、オブジェクトストアに一貫した使用状況が表示され、必要に応じてVeeamで設定を調整できます。

StorageGRIDには、[Support]タブの[Metrics]ページにあるシステムの動作を監視するための便利なチャートが用意されています。主にS3の[Overview]、[ILM]、[Traffic Classification Policy]（ポリシーが作成されている場合）の各ダッシュボードを確認します。S3の[Overview]ダッシュボードには、S3の処理率、レイテンシ、要求応答に関する情報が表示されます。

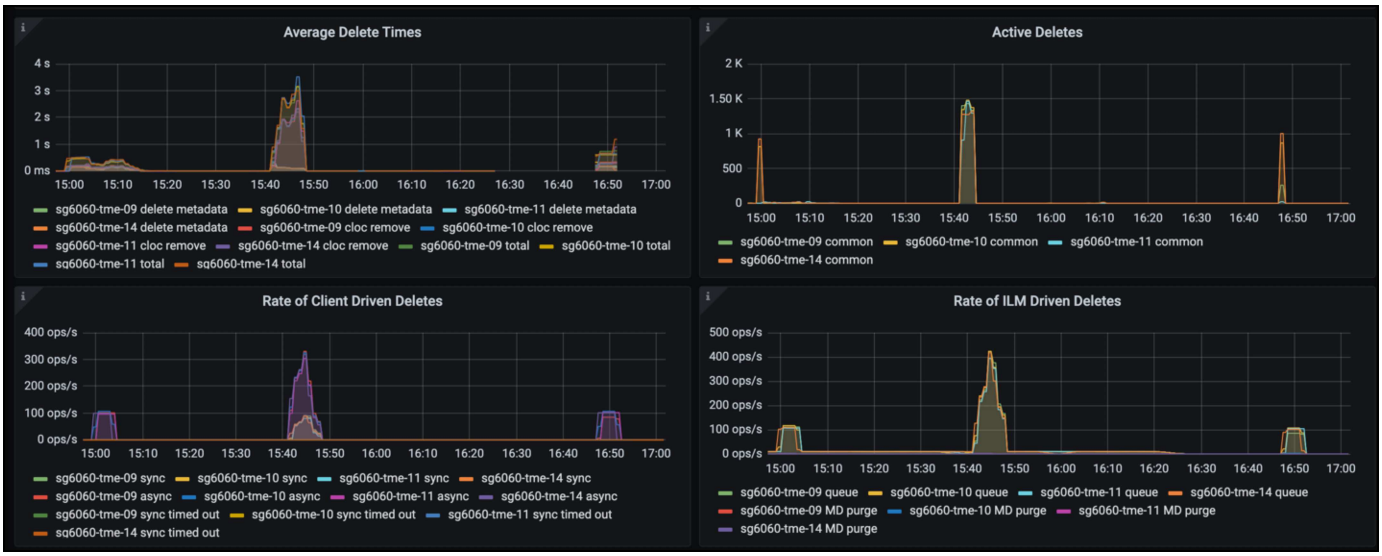
S3の速度とアクティブな要求を確認すると、各ノードで処理されている負荷の量と、タイプ別の要求の総数を確認できます。

[Average Duration]チャートには、各ノードの要求タイプごとの平均所要時間が表示されます。これはリクエストの平均遅延で、追加の調整が必要か、StorageGRIDシステムがより多くの負荷を引き受ける余地があることを示しているかもしれません。

[Total Completed Requests]チャートでは、リクエストをタイプコードと応答コード別に表示できます。応答に200（OK）以外の応答が表示された場合、これは、StorageGRIDシステムのような問題が503（スローダウン）応答を送信しており、追加の調整が必要になるか、負荷が増加するためにシステムを拡張する時間が来たことを示している可能性があります。



[ILM]ダッシュボードでは、StorageGRIDシステムの削除のパフォーマンスを監視できます。StorageGRIDでは、各ノードで同期削除と非同期削除を組み合わせて使用し、すべての要求の全体的なパフォーマンスを最適化しようとします。



トラフィック分類ポリシーを使用すると、ロードバランサ要求のスループット、レート、期間、およびVeeamが送受信するオブジェクトサイズに関するメトリックを表示できます。

追加情報の参照先

このドキュメントに記載されている情報の詳細については、以下のドキュメントや Web サイトを参照してください。

- "NetApp StorageGRID 11.7製品ドキュメント"
- "Veeam Backup Replication"

_ Oliver HaenselとAron Klein著_

StorageGRIDを使用したDremioデータソースの設定

Dremioは、クラウドベースやオンプレミスのオブジェクトストレージなど、多様なデータソースをサポートしています。 StorageGRIDをオブジェクトストレージデータソースとして使用するようにDremioを設定できます。

Dremioデータソースの設定

前提条件

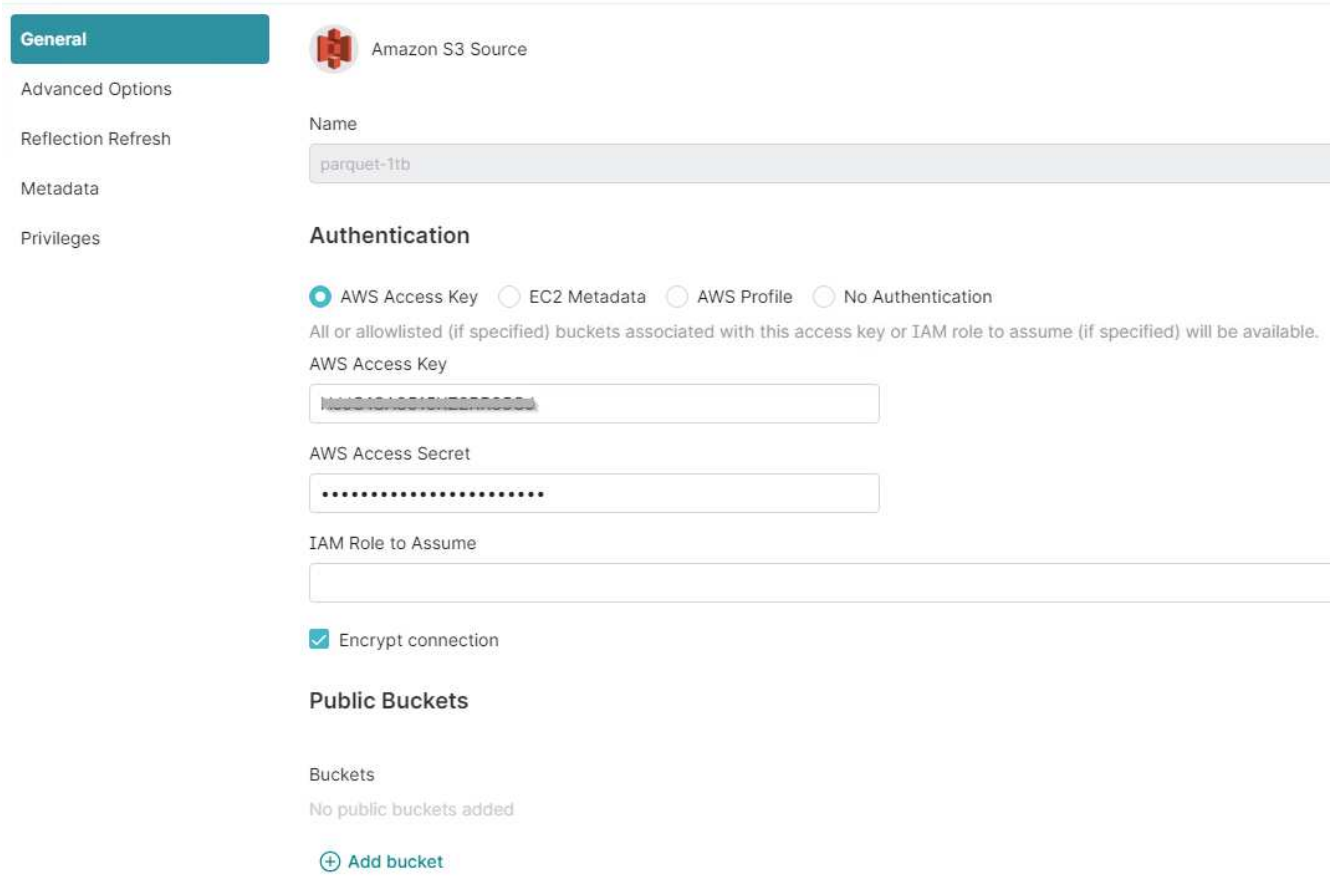
- StorageGRID S3エンドポイントURL、テナントs3アクセスキーID、シークレットアクセスキー。
- StorageGRID構成の推奨事項：圧縮を無効にします（デフォルトでは無効）。 [+]
  Dremioは、Byte range GETを使用して、クエリ中に同じオブジェクト内から異なるバイト範囲を同時に取得します。 バイト範囲要求の一般的なサイズは1MBです。圧縮オブジェクトを使用すると、バイト範囲GETのパフォーマンスが低下します。

Dremioガイド

"Amazon S3への接続- S3互換ストレージの設定"。

指示

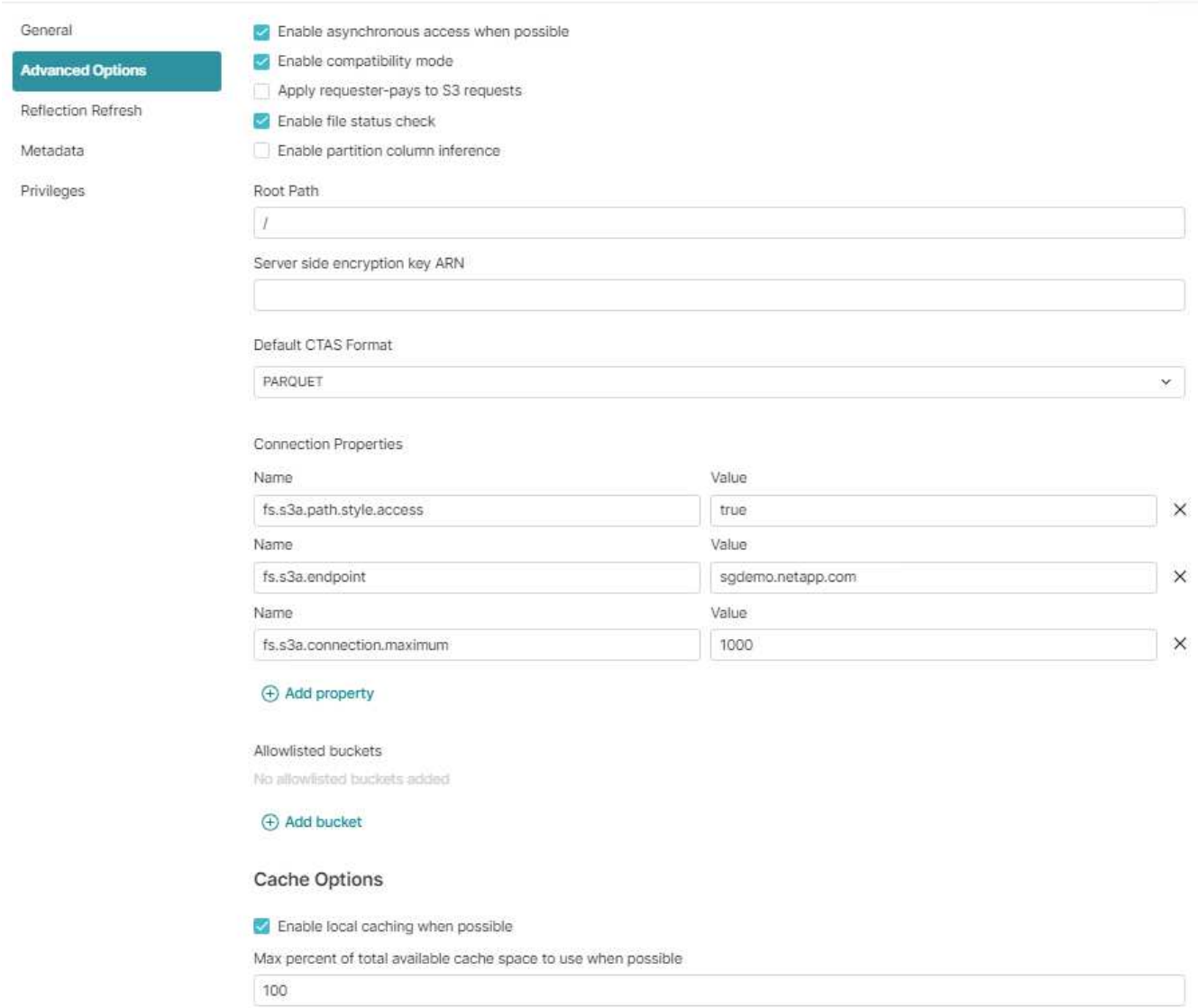
1. [Dremio Datasets]ページで、[+]をクリックしてソースを追加し、[Amazon S3]を選択します。
2. この新しいデータソースの名前（StorageGRID S3のテナントアクセスキーIDとシークレットアクセスキー）を入力します。
3. StorageGRID S3エンドポイントへの接続にhttpsを使用する場合は、[Encrypt connection]チェックボックスをオンにします。[+]
   このs3エンドポイントで自己署名CA証明書を使用する場合は、Dremioのガイド手順に従って、このCA証明書をDremioサーバの<JAVA_HOME>/jre/lib/security+に追加します。
   サンプルスクリーンショット



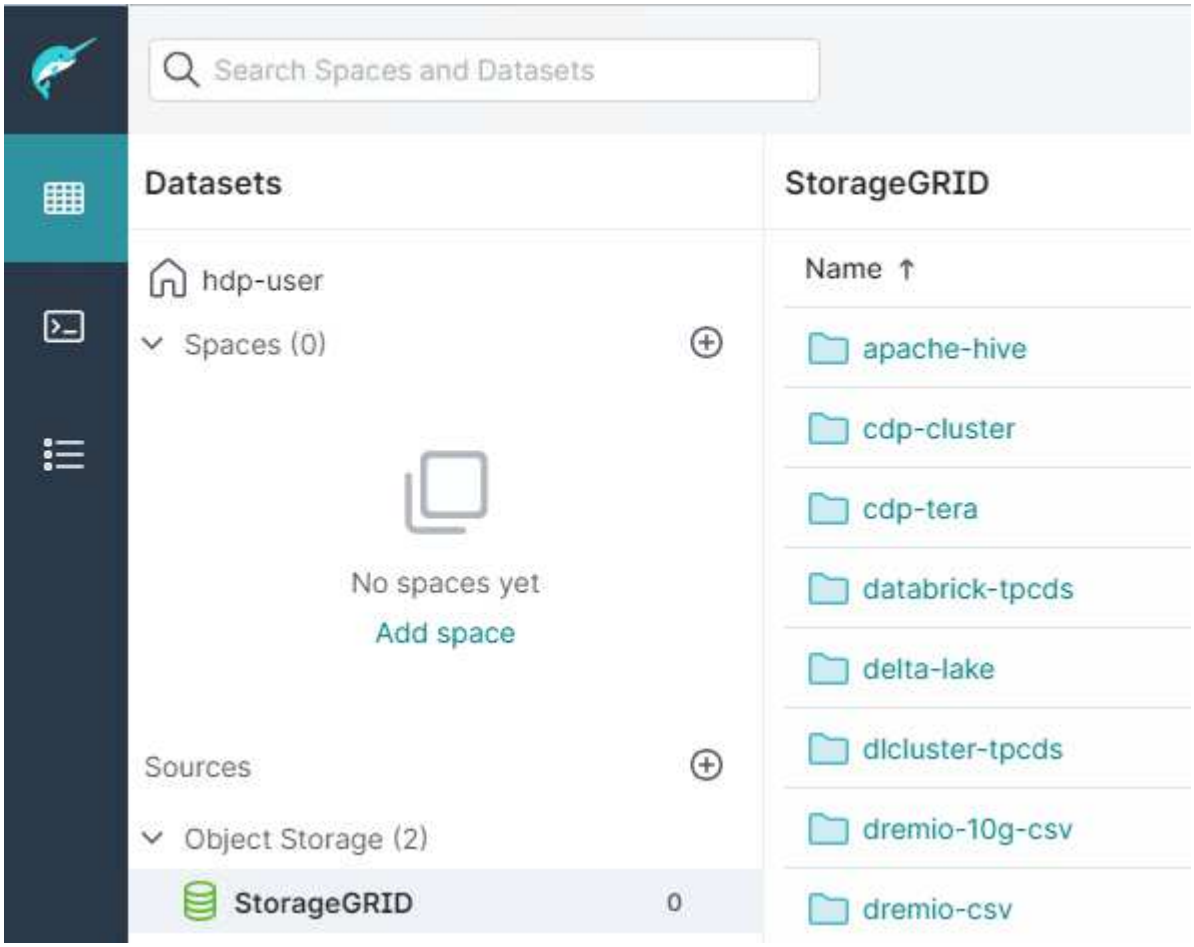
4. [詳細オプション]をクリックし、[互換モードを有効にする]をオンにします。
5. [Connection properties]で、[+ Add Properties]をクリックして、これらのs3aプロパティを追加します。
6. fs.s3a.connection.maximumデフォルトは100です。 s3データセットに100列以上の大きな寄木細工ファイルが含まれている場合は、100より大きい値を入力する必要があります。 この設定については、Dremioのガイドを参照してください。

名前	価値
FS.s3a.endpoint	_ StorageGRID S3エンドポイント：port>_
FS.s3a.path.style.access	正しいです
fs.s3a.connection.maximum	< *100*より大きい値>

サンプルスクリーンショット



7. 組織またはアプリケーションの要件に応じて、その他のDremioオプションを設定します。
8. [Save]ボタンをクリックして新しいデータソースを作成します。
9. StorageGRIDデータソースが正常に追加されると、バケットのリストが左側のパネルに表示されます。[+]サンプルスクリーンショット



Angela Cheng著_

NetApp StorageGRIDとGitLab

NetAppはStorageGRIDをGitLabでテストしました。以下のGitLabの設定例を参照してください。 を参照してください "GitLabオブジェクトストレージ構成ガイド" を参照してください。

オブジェクトストレージの接続例

Linuxパッケージのインストールの場合は、次の例を参照してください。 `connection` 統合フォームでの設定。編集 `/etc/gitlab/gitlab.rb` 次の行を追加し、必要な値を置き換えます。

```
# Consolidated object storage configuration
gitlab_rails['object_store']['enabled'] = true
gitlab_rails['object_store']['proxy_download'] = true
gitlab_rails['object_store']['connection'] = {
  'provider' => 'AWS',
  'region' => 'us-east-1',
  'endpoint' => 'https://<storagegrid-s3-endpoint:port>',
  'path_stype' => 'true',
  'aws_access_key_id' => '<AWS_ACCESS_KEY_ID>',
  'aws_secret_access_key' => '<AWS_SECRET_ACCESS_KEY>'
}
# OPTIONAL: The following lines are only needed if server side encryption
is required
gitlab_rails['object_store']['storage_options'] = {
  'server_side_encryption' => 'AES256'
}
gitlab_rails['object_store']['objects']['artifacts']['bucket'] = 'gitlab-
artifacts'
gitlab_rails['object_store']['objects']['external_diffs']['bucket'] =
'gitlab-mr-diffs'
gitlab_rails['object_store']['objects']['lfs']['bucket'] = 'gitlab-lfs'
gitlab_rails['object_store']['objects']['uploads']['bucket'] = 'gitlab-
uploads'
gitlab_rails['object_store']['objects']['packages']['bucket'] = 'gitlab-
packages'
gitlab_rails['object_store']['objects']['dependency_proxy']['bucket'] =
'gitlab-dependency-proxy'
gitlab_rails['object_store']['objects']['terraform_state']['bucket'] =
'gitlab-terraform-state'
gitlab_rails['object_store']['objects']['pages']['bucket'] = 'gitlab-
pages'
```

手順とAPIの例

StorageGRID でS3暗号化オプションをテストして実証

StorageGRID とS3 APIには、保存データを暗号化するためのさまざまな方法が用意されています。詳細については、を参照してください "StorageGRID の暗号化方式を確認します"。

このガイドでは、S3 APIの暗号化メソッドについて説明します。

サーバー側の暗号化（SSE）

SSEを使用すると、クライアントがオブジェクトを格納し、StorageGRID で管理される一意のキーで暗号化できます。オブジェクトが要求されると、StorageGRID に格納されたキーによってオブジェクトが復号化されます。

SSEの例

- SSEを持つオブジェクトを配置します

```
aws s3api put-object --bucket <bucket>  --key <file> --body "<file>"
--server-side-encryption AES256 --endpoint-url https://s3.example.com
```

- オブジェクトのヘッダーで暗号化を確認します

```
aws s3api head-object --bucket <bucket>  --key <file>   --endpoint-url
https://s3.example.com
```

```
{
    "AcceptRanges": "bytes",
    "LastModified": "2022-05-02T19:03:03+00:00",
    "ContentLength": 47,
    "ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
    "ContentType": "text/plain",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
```

- オブジェクトを取得します

```
aws s3api get-object --bucket <bucket>  --key <file> <file> --endpoint
-url https://s3.example.com
```

ユーザ指定のキーによるサーバ側の暗号化（SSE-C）

SSEを使用すると、クライアントがオブジェクトを格納し、クライアントがオブジェクトで提供する一意のキーでオブジェクトを暗号化できます。オブジェクトが要求されたときに、オブジェクトを復号化して返すために同じキーを指定する必要があります。

SSE-Cの例

- テストまたはデモ目的で暗号化キーを作成できます
  - 暗号化キーを作成します

```
openssl enc -aes-128-cbc -pass pass:secret -P`
```

```
salt=E9DBB6603C7B3D2A
key=23832BAC16516152E560F933F261BF03
iv =71E87C0F6EC3C45921C2754BA131A315
```

- 生成されたキーを持つオブジェクトを配置します

```
aws s3api put-object --bucket <bucket> --key <file> --body "file" --sse
-customer-algorithm AES256 --sse-customer-key
23832BAC16516152E560F933F261BF03 --endpoint-url https://s3.example.com
```

- オブジェクトの先頭に追加します

```
aws s3api head-object --bucket <bucket> --key <file> --sse-customer
-algorithm AES256 --sse-customer-key 23832BAC16516152E560F933F261BF03
--endpoint-url https://s3.example.com
```

```
{
    "AcceptRanges": "bytes",
    "LastModified": "2022-05-02T19:20:02+00:00",
    "ContentLength": 47,
    "ETag": "\"f92ef20ab87e0e13951d9bee862e9f9a\"",
    "ContentType": "binary/octet-stream",
    "Metadata": {},
    "SSECustomerAlgorithm": "AES256",
    "SSECustomerKeyMD5": "rjGuMdjLpPV1eRuotNaPMQ=="
}
```

暗号化キーを指定しないと、「The error occurred (404) when calling the HeadObject operation: not found"」(ヘッダオブジェクト操作:見つかりません)というエラーが表示されます。

- オブジェクトを取得します

```
aws s3api get-object --bucket <bucket> --key <file> <file> --sse
-customer-algorithm AES256 --sse-customer-key
23832BAC16516152E560F933F261BF03 --endpoint-url https://s3.example.com
```

暗号化キーを指定しないと、「An error occurred（InvalidRequest）when calling the GetObject operation：the object was stored using a form of Server Side Encryption」というエラーが表示されます。オブジェクトを読み出すには、正しいパラメータを指定する必要があります。"

バケットサーバ側の暗号化（SSE-C）

SSE-Cを使用すると、バケットに格納されているすべてのオブジェクトのデフォルトの暗号化動作をクライアントで定義できます。オブジェクトはStorageGRID で管理される一意のキーで暗号化されます。オブジェクトが要求されると、StorageGRID に格納されているキーによってオブジェクトが復号化されます。

バケット**SSE-C**の例

- 新しいバケットを作成し、デフォルトの暗号化ポリシーを設定
  - 新しいバケットを作成する

```
aws s3api create-bucket --bucket <bucket> --region us-east-1
--endpoint-url https://s3.example.com
```

  - PUT Bucket encryptionの設定

```
aws s3api put-bucket-encryption --bucket <bucket> --server-side
-encryption-configuration '{"Rules":
[{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm":
"AES256"}}]}' --endpoint-url https://s3.example.com
```

- オブジェクトをバケットに配置します

```
aws s3api put-object --bucket <bucket> --key <file> --body "file"
--endpoint-url https://s3.example.com
```

- オブジェクトの先頭に追加します

```
aws s3api head-object --bucket <bucket> --key <file> --endpoint-url
https://s3.example.com
```

```
{
"AcceptRanges": "bytes",
"LastModified": "2022-05-02T20:16:23+00:00",
"ContentLength": 47,
"ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
"ContentType": "binary/octet-stream",
"ServerSideEncryption": "AES256",
"Metadata": {}
}
```

- オブジェクトを取得します

```
aws s3api get-object --bucket <bucket>  --key <file> <file> --endpoint
-url https://s3.example.com
```

アロンクライン著

StorageGRID でS3オブジェクトロックをテストして実証

Object Lockは、オブジェクトが削除または上書きされないようにWORMモデルを提供します。StorageGRID によるオブジェクトロックの実装では、規制要件を満たし、オブジェクト保持のリーガルホールドとコンプライアンスモードをサポートし、バケットのデフォルト保持ポリシーをサポートするように、Cohassetが評価されます。

このガイドでは、S3オブジェクトロックAPIについて説明します。

リーガルホールド

- オブジェクトロックリーガルホールドは、オブジェクトに適用される単純なオン/オフステータスです。

```
aws s3api put-object-legal-hold --bucket <bucket> --key <file> --legal
-hold Status=ON --endpoint-url https://s3.company.com
```

- GET処理で検証します。

```
aws s3api get-object-legal-hold --bucket <bucket> --key <file>
--endpoint-url https://s3.company.com
```

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- リーガルホールドをオフにします

```
aws s3api put-object-legal-hold --bucket <bucket> --key <file> --legal
-hold Status=OFF --endpoint-url https://s3.company.com
```

- GET処理で検証します。

```
aws s3api get-object-legal-hold --bucket <bucket> --key <file>
--endpoint-url https://s3.company.com
```

```
{
    "LegalHold": {
        "Status": "OFF"
    }
}
```

Complianceモード

- オブジェクトの保持には、タイムスタンプがretain untilを使用します。

```
aws s3api put-object-retention --bucket <bucket> --key <file>
--retention '{"Mode":"COMPLIANCE", "RetainUntilDate": "2025-06-
10T16:00:00"}' --endpoint-url https://s3.company.com
```

- 保持ステータスを確認

```
aws s3api get-object-retention --bucket <bucket> --key <file> --endpoint
-url https://s3.company.com
+
```

```
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2025-06-10T16:00:00+00:00"
    }
}
```

デフォルトの保持

- オブジェクト単位のAPIで定義された保持期限を日数と年数で設定します。

```
aws s3api put-object-lock-configuration --bucket <bucket> --object-lock
-configuration '{"ObjectLockEnabled": "Enabled", "Rule": {
"DefaultRetention": { "Mode": "COMPLIANCE", "Days": 10 }}}' --endpoint
-url https://s3.company.com
```

- 保持ステータスを確認

```
aws s3api get-object-lock-configuration --bucket <bucket> --endpoint-url
https://s3.company.com
```

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 10
            }
        }
    }
}
```

- オブジェクトをバケットに配置します

```
aws s3api put-object --bucket <bucket> --key <file> --body "file"
--endpoint-url https://s3.example.com
```

- バケットで設定された保持期間がオブジェクトの保持タイムスタンプに変換されます。

```
aws s3api get-object-retention --bucket <bucket> --key <file> --endpoint
-url https://s3.company.com
```

```
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2022-03-02T15:22:47.202000+00:00"
    }
}
```

保持期間が定義されているオブジェクトの削除をテストします

オブジェクトロックは、バージョン管理の上に構築されます。保持期間はオブジェクトのバージョンで定義されます。保持が定義されているオブジェクトを削除しようとしたときに、バージョンが指定されていない場合は、削除マーカーがオブジェクトの現在のバージョンとして作成されます。

- 保持期間が定義されたオブジェクトを削除します

```
aws s3api delete-object --bucket <bucket> --key <file> --endpoint-url
https://s3.example.com
```

- バケット内のオブジェクトをリストします

```
aws s3api list-objects --bucket <bucket> --endpoint-url
https://s3.example.com
```

  - オブジェクトがリストされていないことに注意してください。

- 削除マーカーとロックされた元のバージョンを表示するには、バージョンをリストします

```
aws s3api list-object-versions --bucket <bucket> --prefix <file>
--endpoint-url https://s3.example.com
```

```
{
    "Versions": [
        {
            "ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
            "Size": 47,
            "StorageClass": "STANDARD",
            "Key": "file.txt",
            "VersionId":
"RDVDMjYwMTQtQkNDQS0xMUVDLThGOEUtNjQ3NTAwQzAxQTk1",
            "IsLatest": false,
            "LastModified": "2022-04-15T14:46:29.734000+00:00",
            "Owner": {
                "DisplayName": "Tenant01",
                "ID": "56622399308951294926"
            }
        }
    ],
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "Tenant01",
                "ID": "56622399308951294926"
            },
            "Key": "file01.txt",
            "VersionId":
"QjVDQzgzOTAtQ0FGNi0xMUVDLThFMzgtQ0RGMjAwQjk0MjM1",
            "IsLatest": true,
            "LastModified": "2022-05-03T15:35:50.248000+00:00"
        }
    ]
}
```

- ロックされているオブジェクトのバージョンを削除します

```
aws s3api delete-object  --bucket <bucket> --key <file> --version-id
"<VersionId>" --endpoint-url https://s3.example.com
```

```
An error occurred (AccessDenied) when calling the DeleteObject
operation: Access Denied
```

アロンクライン著

バケットポリシーとグループポリシー（IAM）の例

バケットポリシーとグループポリシー（IAMポリシー）の例を次に示します。

グループポリシー（IAM）

ホームディレクトリ形式のバケットアクセス

このグループポリシーでは、users usernameという名前のバケット内のオブジェクトへのアクセスのみがユーザに許可されます。

```
"Statement": [
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::home",
      "Condition": {
        "StringLike": {
          "s3:prefix": "${aws:username}/*"
        }
      }
    },
    {
      "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "arn:aws:s3:::home/?/?/${aws:username}/*"
    }

  ]
}
```

オブジェクトロックバケットの作成を拒否します

このグループポリシーでは、ユーザがバケットを作成してそのバケットでオブジェクトロックを有効にすることはできません。

このポリシーはStorageGRID UIでは適用されず、S3 APIでのみ適用されます。

```
{
    "Statement": [
        {
            "Action": "s3:*",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Action": [
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

オブジェクトロックの保持制限

このバケットポリシーでは、Object-Lockの保持期間が10日以下に制限されます

```
{
 "Version":"2012-10-17",
 "Id":"CustSetRetentionLimits",
 "Statement": [
   {
    "Sid":"CustSetRetentionPeriod",
    "Effect":"Deny",
    "Principal":"*",
    "Action": [
      "s3:PutObjectRetention"
    ],
    "Resource":"arn:aws:s3:::testlock-01/*",
    "Condition": {
      "NumericGreaterThan": {
        "s3:object-lock-remaining-retention-days":"10"
      }
    }
   }
  ]
}
```

ユーザーによるオブジェクトの削除を**versionId**で制限します

このグループポリシーは、versionIdでバージョン管理オブジェクトを削除することをユーザに制限します

```
{
    "Statement": [
        {
            "Action": [
                "s3:DeleteObjectVersion"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Action": "s3:*",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

このバケットポリシーは、ユーザ（ユーザID「56622399308951294926」で識別）がversionIdでバージョン管理オブジェクトを削除することを制限します

```
{
  "Statement": [
    {
      "Action": [
        "s3:DeleteObjectVersion"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::verdeny/*",
      "Principal": {
        "AWS": [
          "56622399308951294926"
        ]
      }
    },
    {
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::verdeny/*",
      "Principal": {
        "AWS": [
          "56622399308951294926"
        ]
      }
    }
  ]
}
```

バケットを読み取り専用アクセス権を持つ単一ユーザに制限します

このポリシーでは、1人のユーザにバケットへの読み取り専用アクセスを許可し、他のすべてのユーザへのアクセスを明示的に拒否します。評価を迅速に行うには、ポリシーの先頭にDenyステートメントをグループ化することを推奨します。

```
{
    "Statement": [
        {
            "Sid": "Deny non user1",
            "Effect": "Deny",
            "NotPrincipal": {
                "AWS":
"urn:sgws:identity::34921514133002833665:user/user1"
            },
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "urn:sgws:s3:::bucket1",
                "urn:sgws:s3:::bucket1/*"
            ]
        },
        {
            "Sid": "Allow user1 read access to bucket bucket1",
            "Effect": "Allow",
            "Principal": {
                "AWS":
"urn:sgws:identity::34921514133002833665:user/user1"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "urn:sgws:s3:::bucket1",
                "urn:sgws:s3:::bucket1/*"
            ]
        }
    ]
}
```

グループを読み取り専用アクセスで単一のサブディレクトリ（プレフィックス）に制限する

このポリシーでは、グループのメンバーにバケット内のサブディレクトリ（プレフィックス）への読み取り専用アクセスを許可します。バケット名は「study」、サブディレクトリは「study01」です。

```
{
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
```

```
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "AllowRootAndstudyListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3::: study"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "",
                        "study01/"
                    ],
                    "s3:delimiter": [
                        "/"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListingOfstudy01",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::study"
            ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "study01/*"
                    ]
                }
            }
        },
```

```
        {
            "Sid": "AllowAllS3ActionsInstudy01Folder",
            "Effect": "Allow",
            "Action": [
                "s3:Getobject"
            ],
            "Resource": [
                "arn:aws:s3:::study/study01/*"
            ]
        }
    ]
}
```

テクニカルレポート

NetApp StorageGRIDとビッグデータ分析

NetApp StorageGRIDのユースケース

NetApp StorageGRIDオブジェクトストレージ解決策は、拡張性、データ可用性、セキュリティ、ハイパフォーマンスを提供します。StorageGRID S3は、あらゆる規模のさまざまな業界の組織で幅広いユースケースに使用されています。典型的なシナリオをいくつか見てみましょう。

ビッグデータ分析： StorageGRID S3はデータレイクとしてよく使用されています。企業は、Apache Spark、Splunk Smartstore、Dremioなどのツールを使用して、分析用に大量の構造化データと非構造化データを保存します。

データ階層化： NetAppのお客様は、ONTAPのFabricPool機能を使用して、ハイパフォーマンスなローカル階層間でStorageGRIDにデータを自動的に移動します。階層化することで、高価なフラッシュストレージをホットデータ用に解放し、コールドデータを低コストのオブジェクトストレージでいつでも利用できる状態に維持できます。これにより、パフォーマンスとコスト削減が最大化されます。

*データのバックアップとディザスタリカバリ：*企業は、StorageGRID S3を信頼性とコスト効率に優れた解決策として使用して、重要なデータのバックアップと災害時のリカバリを実行できます。

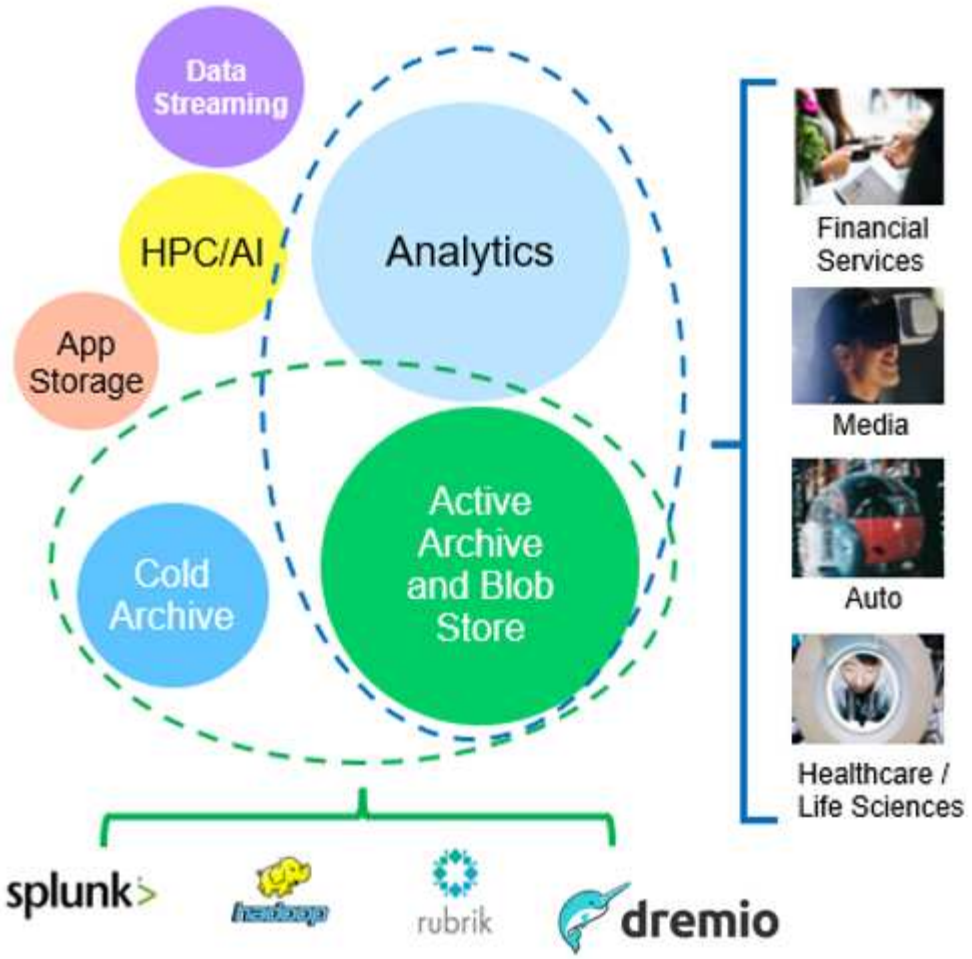
アプリケーション用のデータストレージ： StorageGRID S3はアプリケーションのストレージバックエンドとして使用できるため、開発者はファイル、画像、ビデオ、その他の種類のデータを簡単に保存および取得できます。

コンテンツ配信： StorageGRID S3を使用すると、静的なWebサイトコンテンツ、メディアファイル、ソフトウェアダウンロードを世界中のユーザに保存して配信できます。StorageGRIDの地理的な配信とグローバルネームスペースを活用して、高速で信頼性の高いコンテンツ配信を実現できます。

データ階層化： NetAppのお客様は、ONTAP FabricPool機能を使用して、ハイパフォーマンスなローカル階層間でStorageGRIDにデータを自動的に移動します。階層化することで、高価なフラッシュストレージをホットデータ用に解放し、コールドデータを低コストのオブジェクトストレージからいつでも利用できる状態に保ちます。これにより、パフォーマンスとコスト削減が最大化されます。

データアーカイブ： StorageGRIDは、さまざまな種類のストレージを提供し、パブリックな長期低コストストレージオプションへの階層化をサポートします。コンプライアンスや履歴目的で保持する必要があるデータのアーカイブや長期保存に最適な解決策です。

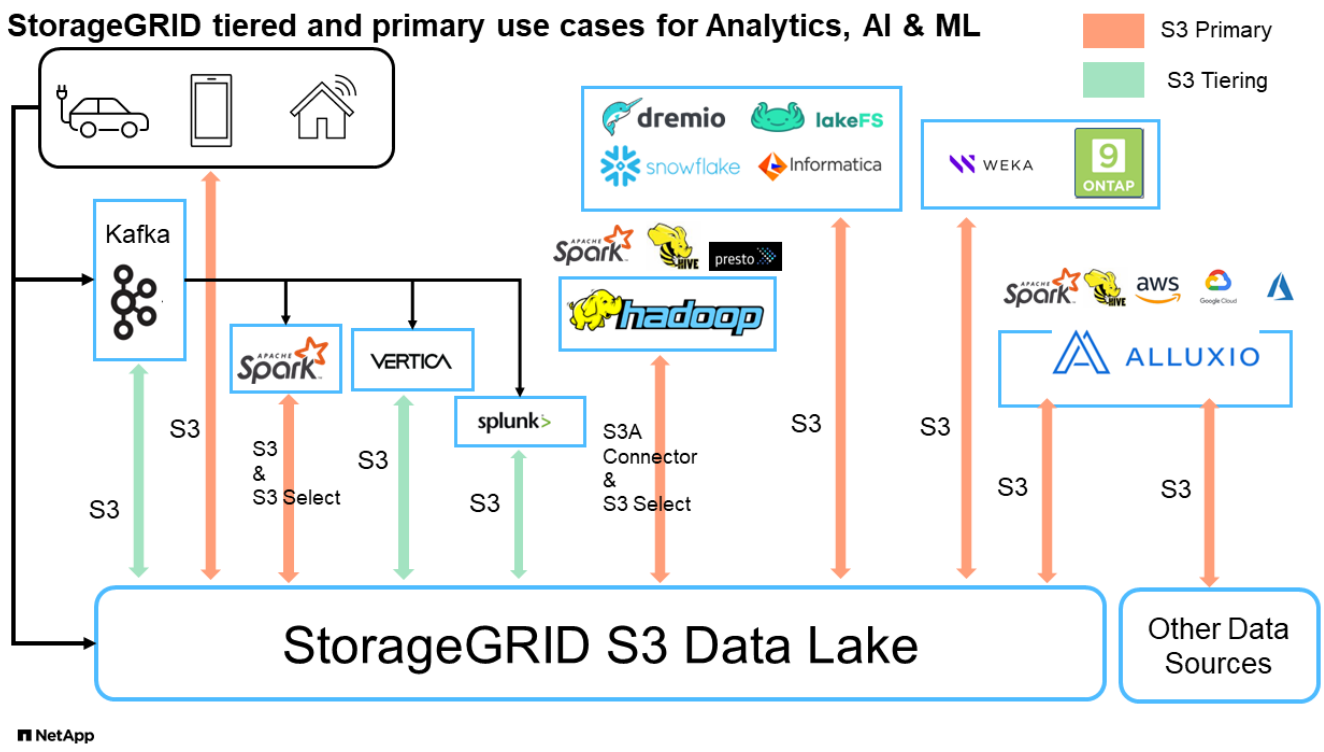
オブジェクトストレージのユースケース



上記の中で、ビッグデータ分析は最も多くのユースケースの1つであり、その使用量は増加傾向にあります。

データレイクに**StorageGRID**を選ぶ理由

- コラボレーションの強化-業界標準のAPIアクセスによる大規模な共有マルチサイト、マルチテナンシー
- 運用コストの削減-単一の自己回復型自動スケールアウトアーキテクチャによる運用の簡易化
- 拡張性-従来のHadoopやデータウェアハウスソリューションとは異なり、StorageGRID S3オブジェクトストレージはコンピューティングやデータからストレージを切り離し、ビジネスの成長に合わせてストレージニーズを拡張できます。
- 耐久性と信頼性- StorageGRIDは99.999999999%の耐久性を提供し、保存されたデータはデータ損失に対して非常に耐性があります。また、高可用性を提供し、データへの常時アクセスを保証します。
- セキュリティ- StorageGRIDは、暗号化、アクセス制御ポリシー、データライフサイクル管理、オブジェクトロック、S3バケットに格納されたデータを保護するバージョン管理など、さまざまなセキュリティ機能を提供します。
- StorageGRID S3データレイク*



S3オブジェクトストレージに最も適したデータウェアハウスまたはデータレイク

NetAppは、Hive、Delta Lake、Dremioの3つのデータウェアハウス/レイクハウスエコシステムでStorageGRIDをベンチマークしました。 "『Apache Iceberg: The Definitive Guide』" データウェアハウスとデータレイクハウスの簡単な紹介と、これら2つのアーキテクチャの長所と短所が含まれています。

- ベンチマークツール- TPC-DS- https://www.tpc.org/tpcds/
- ビッグデータエコシステム
  - 5台のVMで構成されるクラスタ。各VMに128G RAM、24個のvCPU、システムディスク用SSDストレージが搭載されています。
  - Hadoop 3.3.5とHive 3.1.3（1つのネームノード+ 4つのデータノード）
  - Delta LakeとSpark 3.2.0（1マスター+ 4ワーカー）およびHadoop 3.3.5
  - Dremio v23（マスター1名+エグゼキューター4名）
- オブジェクトストレージ
  - SG6060を3台+ SG1000ロードバランサを1台搭載した場合、NetApp ^ StorageGRID®SG®^11.6
  - オブジェクトの保護-コピー×2
- データベースサイズ1000GB
- クエリテストごとに一貫した結果を得るために、3つのエコシステムすべてでキャッシュが無効になりました。

TPC-DSには、クエリベンチマーク用に99の複雑なSQLクエリが付属しています。99個のクエリをすべて完了するまでの合計時間を分単位で測定し、結果を分析するためにS3要求のタイプと数を細かく分析しました。次の表は、全99件のクエリの合計期間を示しています。2番目の表は、各エコシステムがStorageGRIDに送信するS3要求の数とタイプを示しています。

- TPC-DSクエリ結果*

エコシステム	ハイブ	デルタレイク"	デレミオ
ストレージレイヤ	NetApp ® StorageGRID ®	NetApp ® StorageGRID ®	NetApp ® StorageGRID ®
ドライブタイプ	HDD	HDD	HDD
表形式	寄木細工	寄木細工	寄木細工[1]
データベースサイズ	1000 g	1000 g	1000 g
TPCDS 99クエリ+ 合計分数	1084[2]	55	47です

[1]寄木細工と氷山の両方のテーブル形式をテストしましたが、結果は似ています。

[2]Hiveクエリー番号72を完了できません。

- TPC-DSクエリ- S3要求の内訳*

S3要求	ハイブ	デルタレイク"	デレミオ
取得	一、一一七、一八四	2、074、610	四、四一四、二二二七
観察：+ すべての範囲GET	80%のGET：32MBオブジェクトから2KB～2MB、50~100要求/秒	73%の範囲は、32MBオブジェクトから100KB未満、1、000~1400要求/秒	90% 1Mバイト範囲は256MBのオブジェクトから取得、2000~2300の要求/秒
オブジェクトをリスト表示	三一二、〇 五三	二四、一五八	240
頭部+ （存在しないオブジェクト）	156、027	一二、一 〇 三	192年
頭部+ （既存のオブジェクト）	982、126	922、732	一、八四五
リクエスト総数	二、五六七、三九 〇	3、033、603	4、416、504

最初のテーブルから、デルタ湖とDremioがHiveよりもはるかに速いことがわかります。2つ目の表から、Hiveが大量のS3リストオブジェクト要求を送信していることがわかります。この要求は、すべてのオブジェクトストレージプラットフォーム（特に多数のオブジェクトを含むバケットを扱う場合）では通常低速です。これにより、全体的なクエリ時間が大幅に長くなります。もう1つの観測点は、Dremioが大量のGET要求を並行して送信することができたことで、Hiveでは毎秒50~100件の要求に対して、毎秒2,000～2300件の要求が送信されたことです。HiveとHadoop S3AのMimic standard filesystemは、S3オブジェクトストレージのHiveの低速化に貢献しています。

Hadoop（HDFSまたはS3オブジェクトストレージ上）をHiveまたはSparkで使用するには、HadoopとHive/Sparkの広範な知識と、各サービスの設定の相互作用に関する知識が必要です。これらの設定の合計数は1000以上です。多くの場合、設定は相互に関連しており、単独で変更することはできません。使用する設定と値の最適な組み合わせを見つけるには、膨大な時間と労力がかかります。

Dremioは、エンドツーエンドのApache Arrowを使用してクエリのパフォーマンスを劇的に向上させるデータレイクエンジンです。Apache Arrowは、効率的なデータ共有と高速分析のために標準化されたカラムナメモ

リフォーマットを提供します。Arrowは、言語に依存しないアプローチを採用しており、データのシリアライゼーションとデシリアライゼーションの必要性を排除し、複雑なデータプロセスとシステム間のパフォーマンスと相互運用性を向上させるように設計されています。

Dremioの性能は主にDremioクラスター上の計算能力によって駆動される。DremioはS3オブジェクトストレージ接続にHadoopのS3Aコネクタを使用しますが、Hadoopは必須ではなく、Hadoopのfs.s3a設定のほとんどはDremioでは使用されません。これにより、さまざまなHadoop s3a設定の学習とテストに時間を費やすことなく、Dremioのパフォーマンスを簡単に調整できます。

このベンチマーク結果から、S3ベースのワークロード向けに最適化されたビッグデータ分析システムがパフォーマンスの大きな要因であることがわかります。Dremioはクエリの実行を最適化し、メタデータを効率的に利用し、S3データへのシームレスなアクセスを提供するため、S3ストレージを使用する場合にHiveと比較してパフォーマンスが向上します。これを参照してください "ページ" StorageGRIDでDremio S3データソースを設定するには、次の手順を実行します。

StorageGRIDとDremioが連携して最新の効率的なデータレイクインフラを提供する方法や、NetAppがHive + HDFSからDremio + StorageGRIDに移行してビッグデータ分析の効率を劇的に向上させる方法については、以下のリンクをご覧ください。

- "NetApp StorageGRIDでビッグデータのパフォーマンスを向上"
- "StorageGRIDとDremioによる、パワフルで効率性に優れた最新のデータレイクインフラ"
- "NetAppが製品分析でカスタマーエクスペリエンスを再定義する方法"

Hadoop S3Aの調整

Hadoop S3Aコネクタは、HadoopベースのアプリケーションとS3オブジェクトストレージ間のシームレスなやり取りを容易にします。S3オブジェクトストレージを使用する際のパフォーマンスを最適化するには、Hadoop S3Aコネクタの調整が不可欠です。調整の詳細に進む前に、Hadoopとそのコンポーネントの基本を理解しておきましょう。

Hadoopとは

- Hadoop * は、大規模なデータ処理とストレージを処理するために設計された強力なオープンソース・フレームワークです。これにより、コンピュータのクラスタ間で分散ストレージと並列処理が可能になります。

Hadoopの3つのコアコンポーネントは次のとおりです。

- * Hadoop HDFS（Hadoop分散ファイルシステム）*：ストレージを処理し、データをブロックに分割してノード間で分散します。
- * Hadoop MapReduce *：タスクを小さなチャンクに分割し、並行して実行することでデータを処理します。
- * Hadoop YARN（Yet Another Resource Negotiator）：* "リソースの管理とタスクのスケジュール設定を効率的に行う"

Hadoop HDFSおよびS3Aコネクタ

HDFSはHadoopエコシステムの重要なコンポーネントであり、効率的なビッグデータ処理において重要な役割を果たします。HDFSは信頼性の高いストレージと管理を実現します。並列処理と最適化されたデータストレージを実現し、データアクセスと分析を高速化します。

ビッグデータ処理では、HDFSは大規模データセットにフォールトトレラントなストレージを提供することに優れています。これは、データレプリケーションによって実現されます。IT部門は、データウェアハウス環境に大量の構造化データと非構造化データを格納して管理できます。さらに、Apache Spark、Hive、Pig、Flinkなどの主要なビッグデータ処理フレームワークとシームレスに統合し、スケーラブルで効率的なデータ処理を可能にします。UNIXベース(Linux)オペレーティングシステムと互換性があり、ビッグデータ処理にLinuxベースの環境を使用することを好む組織にとって理想的な選択肢です。

時間の経過とともにデータ量が増大するにつれて、独自のコンピューティングとストレージを使用してHadoopクラスタに新しいマシンを追加するアプローチは非効率的になります。リニアに拡張すると、リソースの効率的な使用やインフラの管理が難しくなります。

これらの課題に対処するために、Hadoop S3AコネクタはS3オブジェクトストレージに対するハイパフォーマンスI/Oを提供します。S3Aを使用してHadoopワークフローを実装することで、オブジェクトストレージをデータリポジトリとして活用でき、コンピューティングとストレージを分離することができます。これにより、コンピューティングとストレージを別々に拡張できます。コンピューティングとストレージを分離することで、コンピューティングジョブ専用のリソースを確保し、データセットのサイズに基づいて容量を提供することもできます。そのため、Hadoopワークフローの総所有コストを削減することができます。

Hadoop S3Aコネクタの調整

S3の動作はHDFSとは異なり、ファイルシステムの外観を維持しようとすると積極的に最適化されません。S3リソースを最も効率的に使用するには、慎重な調整、テスト、実験が必要です。

本ドキュメントのHadoopオプションはHadoop 3.3.5に基づいています。を参照してください。 "Hadoop 3.3.5 core-site.xml" 使用可能なすべてのオプションについて。

注–一部のHadoop fs.s3a設定のデフォルト値は、Hadoopのバージョンによって異なります。現在のHadoopバージョンに固有のデフォルト値を確認してください。これらの設定がHadoop core-site.xmlに指定されていない場合は、デフォルト値が使用されます。SparkまたはHive構成オプションを使用して、実行時に値を上書きできます。

これに行く必要があります。 "Apache Hadoopページ" 各fs.s3aオプションを理解するため。可能であれば、非本番環境のHadoopクラスタでテストして最適な値を特定します。

お読みください "S3Aコネクタでの作業時のパフォーマンスの最大化" その他のチューニングの推奨事項については、

主な考慮事項をいくつか見ていきましょう。

- 1 。データ圧縮*

StorageGRID圧縮を有効にしないでください。ほとんどのビッグデータシステムでは、オブジェクト全体を読み出す代わりにバイト範囲GETを使用します。圧縮オブジェクトにbyte range getを使用すると、GETのパフォーマンスが大幅に低下します。

※ 2S3Aコミッタ*

一般的には、マジックs3aコミッターをお勧めします。これを参照してください "共通のS3Aコミッタオプションページ" マジックコミッタとそれに関連するs3a設定をよりよく理解するため。

マジックコミッター：

Magic Committerは、特にS3Guardを使用して、S3オブジェクトストアで一貫したディレクトリリストを提供

します。

整合性のあるS3（現在はそうなっています）を使用すると、Magic Committerは任意のS3バケットで安全に使用できます。

選択と実験：

ユースケースに応じて、Staging Committer（クラスタHDFSファイルシステムに依存）とMagic Committerのどちらかを選択できます。

両方を試して、ワークロードと要件に最適なものを判断してください。

要約すると、S3Aコミッタは、S3への一貫した、高性能で信頼性の高い出力コミットメントという基本的な課題に対する解決策を提供します。内部設計により、データの整合性を維持しながら効率的なデータ転送を実現します。

Option	Meaning	Default (Hadoop 3.3.5)
fs.s3a.committer.name	Committer to create for output to S3A, one of: "file", "directory", "partitioned", "magic".	file
fs.s3a.buffer.dir	Local filesystem directory for data being written and/or staged.	${env.LOCAL_DIRS:-${hadoop.tmp.dir}}/s3a
fs.s3a.committer.magic.enabled	Enable “magic committer” support in the filesystem.	true
fs.s3a.committer.abort.pending.uploads	list and abort all pending uploads under the destination path when the job is committed or aborted.	true
fs.s3a.committer.threads	Number of threads in committers for parallel operations on files.	8
fs.s3a.committer.generate.uuid	Generate a Job UUID if none is passed down from Spark	false
fs.s3a.committer.require.uuid	Require the Job UUID to be passed down from Spark	false
mapreduce.fileoutputcommitter.marksuccessfuljobs	Write a _SUCCESS file on the successful completion of the job.	true
mapreduce.outputcommitter.factory.scheme.s3a	The committer factory to use when writing data to S3A filesystems. If mapreduce.outputcommitter.factory.class is set, it will override this property. (This property is set in mapred-default.xml)	org.apache.hadoop.fs.s3a.commit.S3ACommitterFactory

3.スレッド、接続プールサイズ、ブロックサイズ

- 1つのバケットとやり取りする各* S3A *クライアントには、アップロードおよびコピー処理用のオープンHTTP 1.1接続とスレッドの専用プールがあります。
- "これらのプールサイズを調整して、パフォーマンスとメモリ/スレッド使用量のバランスをとることができます。"。
- S3にデータをアップロードする場合、データはブロックに分割されます。デフォルトのブロックサイズは32MBです。この値をカスタマイズするには、fs.s3a.block.sizeプロパティを設定します。
- ブロックサイズを大きくすると、アップロード中にマルチパートパートパートを管理するオーバーヘッドが軽減されるため、大規模なデータアップロードのパフォーマンスが向上します。大規模なデータセットの場合、推奨値は256 MB以上です。

Option	Meaning	Default (Hadoop 3.3.5)
fs.s3a.threads.max	The total number of threads available in the filesystem for data uploads *or any other queued filesystem operation*.	64
fs.s3a.connection.maximum	Controls the maximum number of simultaneous connections to S3. This must be bigger than the value of fs.s3a.threads.max so as to stop threads being blocked waiting for new HTTPS connections. Why not equal? The AWS SDK transfer manager also uses these connections.	96
fs.s3a.max.total.tasks	The number of operations which can be queued for execution. This is in addition to the number of active threads in fs.s3a.threads.max.	32
fs.s3a.committer.threads	Number of threads in committers for parallel operations on files (upload, commit, abort, delete...)	8
fs.s3a.executor.capacity	The maximum number of submitted tasks which is a single operation (e.g. rename(), delete()) may submit simultaneously for execution -excluding the IO-heavy block uploads, whose capacity is set in "fs.s3a.fast.upload.active.blocks" All tasks are submitted to the shared thread pool whose size is set in "fs.s3a.threads.max"; the value of capacity should be less than that of the thread pool itself, as the goal is to stop a single operation from overloading that thread pool.	16
fs.s3a.fast.upload.active.blocks (see also related fs.s3a.fast.upload.buffer option)	Maximum Number of blocks a single output stream can have active (uploading, or queued to the central FileSystem instance's pool of queued operations. This stops a single stream overloading the shared thread pool.	4
fs.s3a.block.size	Block size to use when reading files using s3a: file system. A suffix from the set {K,M,G,T,P} may be used to scale the numeric value.	32MB (tested 1TB data set with 256MB and 512MB block size shows significant improvement in both read and write)

4.マルチパートアップロード

s3aコミッタ*常に* MPU（マルチパートアップロード）を使用してデータをs3バケットにアップロードします。これは、タスクの失敗、タスクの投機的な実行、およびコミット前のジョブの中止を可能にするために必要です。マルチパートアップロードに関連する主な仕様を次に示します。

- 最大オブジェクトサイズ：5TiB（テラバイト）。
- アップロードあたりの最大パーツ数：10、000
- パーツ番号：1～10,000（含む）。
- パーツサイズ：5MiB~5GiB。特に、マルチパートアップロードの最後のパートには最小サイズの制限はありません。

S3マルチパートアップロードに小さいパートサイズを使用すると、メリットとデメリットの両方があります。

利点：

- ネットワークの問題からのクイックリカバリ:小さなパーツをアップロードすると、ネットワークエラーによるアップロードの再開による影響が最小限に抑えられます。パーツに障害が発生した場合は、オブジェ

クト全体ではなく、その特定のパーツのみを再アップロードする必要があります。

- 並列化の向上:マルチスレッディングまたは同時接続を利用して、より多くのパーツを並行してアップロードできます。この並列化により、特に大きなファイルを処理する場合のパフォーマンスが向上します。

欠点：

- ネットワークオーバーヘッド:部品サイズが小さいほど、アップロードする部品が増えます。各部品には独自のHTTPリクエストが必要です。HTTP要求が増えると、個 々 の要求の開始と完了のオーバーヘッドが増加します。多数の小さなパーツを管理すると、パフォーマンスに影響を与える可能性があります。
- 複雑さ：注文の管理、パーツの追跡、アップロードの成功の確認は面倒です。アップロードを中止する必要がある場合は、すでにアップロードされているすべてのパーツを追跡してパージする必要があります。

Hadoopの場合、fs.s3a.multipart.sizeには256MB以上のパーツサイズを推奨します。fs.s3a.mutlipart.threshold値は常に2 x fs.s3a.multipart.size値に設定します。たとえば、fs.s3a.multipart.size=256Mの場合、fs.s3a.mutlipart.thresholdは512Mにする必要があります。

大きなデータセットには大きなパーツサイズを使用してください。特定のユースケースとネットワーク条件に基づいて、これらの要因のバランスを取る部品サイズを選択することが重要です。

マルチパートアップロードは "3段階のプロセス"：

1. アップロードが開始され、StorageGRIDはupload-idを返します。
2. オブジェクトパーツはupload-idを使用してアップロードされます。
3. すべてのオブジェクトパートがアップロードされると、は、upload-idを指定して完全なマルチパートアップロード要求を送信します。StorageGRIDは、アップロードされたパーツからオブジェクトを構築し、クライアントがオブジェクトにアクセスできるようにします。

Complete multipart upload要求が正常に送信されなかった場合、パーツはStorageGRIDに残り、オブジェクトは作成されません。これは、ジョブが中断、失敗、または中止された場合に発生します。マルチパートアップロードが完了するか中止されるか、アップロードが開始されてから15日が経過するとStorageGRIDがそれらのパートをパージするまで、パートはグリッドに残ります。バケット内で実行中のマルチパートアップロードが多数（数十万から数百万）ある場合、Hadoopが「list-multipart-uploads」を送信すると（この要求はアップロードIDでフィルタリングされません）、要求の完了に時間がかかるか、最終的にタイムアウトになることがあります。fs.s3a.mutlipart.purgeをtrueに設定し、適切なfs.s3a.multipart.purge.ageの値を設定することを検討してください（例：5〜7日、デフォルト値の86400、つまり1日は使用しないでください）。または、NetAppサポートに状況を調査してください。

Option	Meaning	Default (Hadoop 3.3.5)
fs.s3a.multipart.size	How big (in bytes) to split upload or copy operations up into. A suffix from the set {K,M,G,T,P} may be used to scale the numeric value.	64M
fs.s3a.multipart.threshold	How big (in bytes) to split upload or copy operations up into. This also controls the partition size in renamed files, as rename() involves copying the source file(s). A suffix from the set {K,M,G,T,P} may be used to scale the numeric value.	128M
fs.s3a.multipart.purge	True if you want to purge existing multipart uploads that may not have been completed/aborted correctly. The corresponding purge age is defined in fs.s3a.multipart.purge.age. If set, when the filesystem is instantiated then all outstanding uploads older than the purge age will be terminated -across the entire bucket. This will impact multipart uploads by other applications and users. so should be used sparingly, with an age value chosen to stop failed uploads, without breaking ongoing operations.	false
fs.s3a.multipart.purge.age	Minimum age in seconds of multipart uploads to purge on startup if "fs.s3a.multipart.purge" is true	86400

5.メモリ内のバッファ書き込みデータ

パフォーマンスを向上させるには、書き込みデータをS3にアップロードする前にメモリにバッファします。これにより、少量の書き込み数が削減され、効率が向上します。

Option	Meaning	Default (Hadoop 3.3.5)
fs.s3a.fast.upload.buffer	The buffering mechanism to for data being written. Values: disk, array, bytebuffer. "disk" will use the directories listed in fs.s3a.buffer.dir as the location(s) to save data prior to being uploaded. "array" uses arrays in the JVM heap "bytebuffer" uses off-heap memory within the JVM. Both "array" and "bytebuffer" will consume memory in a single stream up to the number of blocks set by: fs.s3a.multipart.size * fs.s3a.fast.upload.active.blocks. If using either of these mechanisms, keep this value low The total number of threads performing work across all threads is set by fs.s3a.threads.max, with fs.s3a.max.total.tasks values setting the number of queued work items.	disk

S3とHDFSは別 々 の方法で機能することに注意してください。S3リソースを最も効率的に使用するには、慎

重な調整/テスト/実験が必要です。

ネットアップのStorageGRID ブログ

ネットアップのStorageGRID に関する優れたブログをいくつかご紹介します。

- 5月10日： "ラボオンデマンドはStorageGRIDに最適な営業ツール"
- 5月24日： "ネットアップとAlluxioによる分析ワークロードの刷新"
- 5月26日： "StorageGRID：オンプレミスのバックアップデータとレプリケーションデータの格納と管理"
- 6月9日 "StorageGRID でCloudera Hadoop S3Aコネクタを使用します"
- 7月26日 "StorageGRID向けの検証済みパートナーソリューションのリストが増え続けていますので、ぜひチェックしてください。"
- 8月5日： "NetApp StorageGRID は、Common Criteriaのセキュリティ認定を取得しています"
- 8月16日： "StorageGRID とオープンソースのELKスタックを統合して、カスタマーエクスペリエンスを強化します"
- 8月17日： "すべてはオブジェクトのロックから始まります。 重要なバックアップアプリケーション向けのS3ストレージエコシステムを構築"
- 8月23日： "StorageGRID 上にデータレイクを構築"
- 9月1日： "これらの指標を使用してグラフ化します"
- 9月19日： "StorageGRID 向けのDataLockおよびランサムウェア対策サポート"
- 9月26日： "NetApp StorageGRID （サービスプロバイダ向け"
- 10月5日： "StorageGRID for Snowflakeでデータを解凍します"
- 10月5日： "NetApp Cloud Insights に、StorageGRID のギャラリーダッシュボードが追加されました"
- 11月7日： "StorageGRID とONTAP S3のサポート：相違点、類似点、統合"
- 11月23日： "ネットアップとModzyを基盤とするMLOpsによる説明可能なAI"
- 12月6日： "StorageGRID はKPMGコンプライアンス認証を取得しています"
- 1月16日： "StorageGRID はNF203およびISO/IEC 25051準拠認定を更新します"
- 1月18日： "Veritas NetBackupでStorageGRID S3オブジェクトロックが検証されました"
- 2月14日： "チョコレート、スキー、時計、メインフレームにはどのような共通点がありますか？"
- 3月14日： "3：2：1準拠のアーキテクチャで1つのコマンドでEpic SystemsのEHRデータベースをバックアップする方法"
- 3月30日： "BlueXPを使用して、3：2：1に準拠したバックアップポリシーでEpic EHRを保護"
- 3月30日： "StorageGRID を使用したAmazon S3 alphaリリースのマウントポイント"
- 5月16日： "StorageGRIDオブジェクトストレージファミリーの新機能"
- 5月16日： "StorageGRID 11.7と新しいオールフラッシュオブジェクトストレージアプライアンスSGF6112の概要"
- 8月30日： "Amazon S3ファイルシステムのマウントポイントの一般提供を開始"
- 9月1日： "Fluent Bitを使用したCloud Insightsによるログの監視と収集"
- 10月17日： "HadoopからのS3移行：DremioとStorageGRIDによるデータ分析の刷新"

- 11月7日： "Spectra Logic On-Prem GlacierとStorageGRID"
- 12月12日： "StorageGRIDでのビッグデータ分析：DremioのパフォーマンスはApache Hiveの23倍"
- 2月2日： "StorageGRID + lakeFS解決策概要の発表"
- 2月16日： "StorageGRID 11.8の紹介：セキュリティ、簡易性、ユーザエクスペリエンスの強化"
- 2月16日： "StorageGRID 11.8の概要"

NetApp StorageGRID のドキュメント

NetApp StorageGRID の各リリースの完全なドキュメントは、次の場所にあります。

- "StorageGRIDアプライアンス"
- "StorageGRID 11.8"
- "StorageGRID 11.7."
- "StorageGRID 11.6"
- "StorageGRID 11.5"
- "StorageGRID 11.4"
- "StorageGRID 11.3"
- "StorageGRID 11.2"

法的通知

著作権に関する声明、商標、特許などにアクセスできます。

著作権

"https://www.netapp.com/company/legal/copyright/"

商標

NetApp 、 NetApp のロゴ、および NetApp の商標ページに記載されているマークは、 NetApp, Inc. の商標です。その他の会社名および製品名は、それぞれの所有者の商標である場合があります。

"https://www.netapp.com/company/legal/trademarks/"

特許

ネットアップが所有する特許の最新リストは、次のサイトで入手できます。

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

プライバシーポリシー

"https://www.netapp.com/company/legal/privacy-policy/"

オープンソース

通知ファイルには、ネットアップソフトウェアで使用されるサードパーティの著作権およびライセンスに関する情報が記載されています。

https://library.netapp.com/ecm/ecm_download_file/2879263

https://library.netapp.com/ecm/ecm_download_file/2881511