Kubernetesを使用したBlack Duckのインストール

Black Duck 2023.1.0

目次

まえがき

Black Duck ドキュメント

Black Duckのドキュメントは、オンラインヘルプと次のドキュメントで構成されています。

タイトル	ファイル	説明
リリースノート	release_notes.pdf	新機能と改善された機能、解決された問題、現在のリリースおよび以前のリリースの既知の問題に関する情報が記載されています。
Docker Swarmを使用したBlack Duckのインストール	install_swarm.pdf	Docker Swarmを使用したBlack Duckのインストールとアップグレードに関する情報が記載されています。
使用する前に	getting_started.pdf	初めて使用するユーザーにBlack Duckの使用法に関する情報を提供します。
スキャンベストプラクティス	scanning_best_practices.pdf	スキャンのベストプラクティスについて説明します。
SDKを使用する前に	getting_started_sdk.pdf	概要およびサンプルのユースケースが記載されています。
レポートデータベース	report_db.pdf	レポートデータベースの使用に関する情報が含まれています。
ユーザーガイド	user_guide.pdf	Black DuckのUI使用に関する情報が含まれています。

KubernetesまたはOpenShift環境にBlack Duckソフトウェアをインストールするためのインストール方法は、SynopsysctlとHelmです。次のリンクをクリックすると、マニュアルが表示されます。

- Helmは、Black Duckのインストールに使用できるKubernetesのパッケージマネージャです。Black DuckはHelm3をサポートしており、Kubernetesの最小バージョンは1.13です。
- Synopsysctlは、KubernetesおよびRed Hat OpenShiftにBlack Duckソフトウェアを展開するためのクラウドネイティブの管理コマンドラインツールです。

Black Duck 統合に関するドキュメントはConfluenceで入手できます。

カスタマサポート

ソフトウェアまたはドキュメントについて問題がある場合は、Synopsysカスタマサポートに問い合わせてください。

Synopsysサポートには、複数の方法で問い合わせできます。

- オンライン：https://www.synopsys.com/software-integrity/support.html
- 電話：お住まいの地域の電話番号については、サポートページの下段にあるお問い合わせのセクションを参照してください。

サポートケースを開くには、Synopsys Software Integrityコミュニティサイト（https://community.synopsys.com/s/contactsupport）にログインしてください。

常時対応している便利なリソースとして、オンラインカスタマポータルを利用できます。

Synopsys Software Integrityコミュニティ

Synopsys Software Integrityコミュニティは、カスタマサポート、ソリューション、および情報を提供する主要なオンラインリソースです。コミュニティでは、サポートケースをすばやく簡単に開いて進捗状況を監視したり、重要な製品情報を確認したり、ナレッジベースを検索したり、他のSoftware Integrityグループ(SIG)のお客様から情報を得ることができます。コミュニティセンターには、共同作業に関する次の機能があります。

- つながる – サポートケースを開いて進行状況を監視するとともに、エンジニアリング担当や製品管理担当の支援が必要になる問題を監視します。
- 学ぶ – 他のSIG製品ユーザーの知見とベストプラクティスを通じて、業界をリードするさまざまな企業から貴重な教訓を学ぶことができます。さらにCustomer Hubでは、最新の製品ニュースやSynopsysの最新情報をすべて指先の操作で確認できます。これは、オープンソースの価値を組織内で最大限に高めるように当社の製品やサービスをより上手に活用するのに役立ちます。
- 解決する – SIGの専門家やナレッジベースが提供する豊富なコンテンツや製品知識にアクセスして、探している回答をすばやく簡単に得ることができます。
- 共有する – Software Integrityグループのスタッフや他のお客様とのコラボレーションを通じて、クラウドソースソリューションに接続し、製品の方向性について考えを共有できます。

Customer Successコミュニティにアクセスしましょう。アカウントをお持ちでない場合や、システムへのアクセスに問題がある場合は、こちらをクリックして開始するか、community.manager@synopsys.comにメールを送信してください。

トレーニング

Synopsys Software Integrity, Customer Education(SIG Edu)は、すべてのBlack Duck教育ニーズに対応するワンストップリソースです。ここでは、オンライントレーニングコースやハウツービデオへの24時間365日のアクセスを利用できます。

新しいビデオやコースが毎月追加されます。

Synopsys Software Integrity, Customer Education(SIG Edu)では、次のことができます。

- 自分のペースで学習する。
- 希望する頻度でコースを復習する。
- 試験を受けて自分のスキルをテストする。
- 終了証明書を印刷して、成績を示す。

詳細については、https://community.synopsys.com/s/educationを参照してください。また、Black Duckのヘルプについては、Black Duck UIの[ヘルプ]メニュー()から、[Black Duckチュートリアル]を選択します。

包括性と多様性に関するSynopsysの声明

Synopsysは、すべての従業員、お客様、パートナーが歓迎されていると感じられる包括的な環境の構築に取り組んでいます。当社では、製品およびお客様向けのサポート資料から排他的な言葉を確認して削除しています。また、当社の取り組みには、設計および作業環境から偏見のある言葉を取り除く社内イニシアチブも含まれ、これはソフトウェアやIPに組み込まれている言葉も対象になっています。同時に、当社は、能力の異なるさまざまな人々が当社のWebコンテンツおよびソフトウェアアプリケーションを利用できるように取り組んでいます。なお、当社のIPは、排他

的な言葉を削除するための現在検討中である業界標準仕様を実装しているため、当社のソフトウェアまたはドキュメントには、非包括的な言葉の例がまだ見つかる場合があります。

1. Synopsysctlを使用したインストール

Kubernetesは、コンテナを介してクラウドワークロードを管理するためのオーケストレーションツールです。

synopsysctlを使用したインストール

synopsysctlを使用して、Black DuckをKubernetesまたはは OpenShiftにインストールします。

synopsysctlは、KubernetesおよびOpenShiftクラスタでのSynopsysソフトウェアの導入と管理を支援するコマンドラインツールです。synopsysctlをインストールした後には、それを利用してSynopsysソフトウェアを簡単に導入および管理することができます。

- synopsysctlの概要についてはここをクリックしてください。
- synossysctlのインストールと使用方法に関するドキュメントについては、ここをクリックしてください。

注： スケーラビリティのサイジングのガイドラインについては、『Black Duckリリースノート』の「コンテナのスケーラビリティ」セクションを参照してください。Synopsysのテクニカルサポート担当者から特に要請がない限り、Black Duckデータベース（bds_hub）からデータを削除しないでください。必ず適切なバックアップ手順に従ってください。データを削除すると、UIの問題からBlack Duckが完全に起動しなくなるという障害に至る、いくつかのエラーが発生する可能性があります。Synopsysのテクニカルサポートは、削除されたデータを再作成することはできません。利用可能なバックアップがない場合、Synopsysは可能な範囲で最善のサポートを提供します。

2. ハードウェア要件

以下のパフォーマンスデータは、署名スキャンの永続性が減少したBlack Duck 2022.10.0（デフォルト）とSynopsys Detect 8.0.0を使用して収集されました。SPH値は、署名スキャン、パッケージマネージャdetectorスキャンおよび高速スキャンの組み合わせを使用して計算されています。平均スキャンサイズはお客様によって異なるため、正確なSPHスループットはお客様固有のものです。これらのメトリックは、Google Cloud Platformから収集されました。Google Cloud Platformは、構成ごとに異なるデータベース読み取り/書き込みIOPSを提供します。

10sph	1時間あたりのスキャン回数：50 SPH増加率：400% 1時間あたりのAPI数：2500 プロジェクトバージョン：10000	IOPS： ・ 読み取り：15000 ・ 書き込み：9000 Black Duckサービス： ・ CPU：12コア ・ メモリ：30 GB PostgreSQL： ・ CPU：2コア ・ メモリ：8 GB	合計： ・ CPU：14コア ・ メモリ：38 GB
120sph	1時間あたりのスキャン回数：120 SPH増加率：0% 1時間あたりのAPI数：3000 プロジェクトバージョン：13000	IOPS： ・ 読み取り：15000 ・ 書き込み：15000 Black Duckサービス： ・ CPU：13コア ・ メモリ：46 GB PostgreSQL： ・ CPU：4コア ・ メモリ：16 GB	合計： ・ CPU：17コア ・ メモリ：62 GB
250sph	1時間あたりのスキャン回数：300 SPH増加率：20% 1時間あたりのAPI数：7500 プロジェクトバージョン：15000	IOPS： ・ 読み取り：15000 ・ 書き込み：15000 Black Duckサービス： ・ CPU：17コア ・ メモリ：118 GB PostgreSQL： ・ CPU：6コア ・ メモリ：24 GB	合計： ・ CPU：23コア ・ メモリ：142 GB
500sph	1時間あたりのスキャン回数：650 SPH増加率：30% 1時間あたりのAPI数：18000 プロジェクトバージョン：18000	IOPS： ・ 読み取り：15000 ・ 書き込み：15000 Black Duckサービス：	合計： ・ CPU：38コア ・ メモリ：250 GB

		・ CPU:28コア ・ メモリ:210 GB PostgreSQL: ・ CPU:10コア ・ メモリ:40 GB	
1000sph	1時間あたりのスキャン回数:1400 SPH増加率:40% 1時間あたりのAPI数:26000 プロジェクトバージョン:25000	IOPS: ・ 読み取り:25000 ・ 書き込み:25000 Black Duckサービス: ・ CPU:47コア ・ メモリ:411 GB PostgreSQL: ・ CPU:18コア ・ メモリ:72 GB	合計: ・ CPU:65コア ・ メモリ:483 GB
1500sph	1時間あたりのスキャン回数:1600 SPH増加率:6% 1時間あたりのAPI数:41000 プロジェクトバージョン:28000	IOPS: ・ 読み取り:25000 ・ 書き込み:25000 Black Duckサービス: ・ CPU:60コア ・ メモリ:597 GB PostgreSQL: ・ CPU:26コア ・ メモリ:104 GB	合計: ・ CPU:92コア ・ メモリ:701 GB
2000sph	1時間あたりのスキャン回数:2300 SPH増加率:15% 1時間あたりのAPI数:50000 プロジェクトバージョン:35000	IOPS: ・ 読み取り:60000 ・ 書き込み:25000 Black Duckサービス: ・ CPU:66コア ・ メモリ:597 GB PostgreSQL: ・ CPU:34コア ・ メモリ:136 GB	合計: ・ CPU:100コア ・ メモリ:733 GB

この新しいガイダンスは、現在のBlack Duck 2022.10.0アーキテクチャに基づいています。このガイダンスは、以降のリリースでさらに改良される可能性があります。ご質問やご不明な点がある場合は、製品管理までお問い合わせください。

注： 必要なディスク容量は、管理するプロジェクトの数によって異なります。したがって、個々の要件が異なる場合があります。各プロジェクトには約200 MBが必要であることを考慮してください。

Black Duck Softwareでは、Black Duckサーバーのディスク使用率を監視して、ディスクが最大容量に達しないようにすることを推奨しています。最大容量に達すると、Black Duckで問題が発生する可能性があります。

BDBAのスケーリングは、1時間あたりに実行される予想バイナリスキャン数に基づいて、binaryscannerレプリカの数を調整し、PostgreSQLリソースを追加することによって行われます。1時間あたり15回のバイナリスキャンごとに、次を追加します。

- 1つのbinaryscannerレプリカ
- PostgreSQL用の1つのCPU
- PostgreSQL用の4 GBのメモリ

予想されるスキャンレートが15の倍数でない場合は、切り上げます。たとえば、1時間あたり24回のバイナリスキャンでは、次のものが必要です。

- 2つのbinaryscannerレプリカ
- PostgreSQL用の2つの追加CPU、および
- PostgreSQL用の8 GBの追加メモリ。

このガイダンスは、バイナリスキャンが合計スキャンボリューム（スキャン数）の20%以下である場合に有効です。

バイナリスキャン

バイナリスキャンのライセンスがある場合、uploadcacheコンテナ/ポッドのメモリを増やす必要がある場合があります。これは、バイナリスキャナがバイナリを抽出して処理する場所であるためです。デフォルトでは、メモリは512MBに設定されていますが、これは大規模なスキャンには適切ではありません。大規模なバイナリをスキャンする場合は、uploadcacheコンテナ/ポッドのメモリを4 GB以上に増やすことをお勧めします。これを実行するには、YAMLを上書きして、メモリ制限を4096MBに更新します。

Swarmインストールの場合：

```
uploadcache:
  deploy:
    resources:
      limits:
        cpus: ".200"
        memory: "4096M"
      reservations:
        cpus: ".100"
        memory: "4096M"
      replicas: 1
```

Kubernetesインストールの場合：

```
uploadcache:
  replicas: 1
  resources:
    limits:
      cpu: "200m"
      memory: "4096Mi"
    requests:
      cpu: "100m"
      memory: "4096Mi"
```

注： Black Duck Alertをインストールするには、1 GBの追加メモリが必要です。

3. PostgreSQLのバージョン

Black Duck 2022.10.0は、新しいPostgreSQLの機能をサポートし、Black Duckサービスのパフォーマンスと信頼性を向上させます。Black Duck 2022.10.0の時点で、内部PostgreSQLコンテナの現在サポートされているPostgreSQLのバージョンは、PostgreSQLコンテナ13です。

以前のバージョンのBlack Duck（2022.10.0より前）からアップグレードする場合は、PostgreSQL 13に移行する必要があります。Black Duck 2022.10.0アップデートは、内部Black Duck PostgreSQLデータベースコンテナをPostgreSQLのバージョン13に移行します。データベースコンテナを使用してOpenShiftに導入する場合は、Black Duckのリリースノートとインストールガイドに記載されているように、1回限りの移行ジョブを実行する必要があります。

注： PostgreSQLのサイジングのガイドラインについては、『Black Duck Hardware Scaling Guidelines』を参照してください。

独自の外部PostgreSQLインスタンスを実行する場合は、新規インストールにPostgreSQL 14を使用することをお勧めします。PostgreSQL 14.0から14.3には、インデックスが破損するというバグがあるため、サポートされているPostgreSQL 14の最小バージョンは14.4です。

注意： PostgreSQLデータディレクトリでウイルス対策スキャンを実行しないでください。ウイルス対策ソフトウェアは、大量のファイルを開いたり、ファイルをロックしたりします。これらはPostgreSQLの操作を妨げます。特定のエラーは製品によって異なりますが、通常、PostgreSQLがデータファイルにアクセスできなくなります。たとえば、PostgreSQLが「システムで開かれているファイルが多すぎます」というエラーを伴って失敗することがあります。

一般的な移行プロセス

このガイダンスは、任意のPG 9.6ベースのHub（2022.2.0より前のリリース）から2022.10.0以降にアップグレードする場合に該当します。

1. 移行は、blackadue-postgres-upgraderコンテナによって実行されます。
2. PostgreSQL 9.6ベースのBlack Duckバージョンからアップグレードする場合：
   - 将来のPostgreSQLバージョンのアップグレードがより簡単になるように、PostgreSQLデータボリュームのフォルダレイアウトが再構成されます。
   - データボリュームの所有者のUIDが変更されます。新しいデフォルトUIDは1001です。ただし、導入固有の説明を参照してください。
3. pg_upgradeスクリプトを実行して、データベースをPostgreSQL 13に移行します。
4. クエリプランナ統計情報を初期化するために、PostgreSQL 13データベース上でプレーンなANALYZEが実行されます。
5. blackduck-postgres-upgraderが終了します。

4. Helmを使用したBlack Duckのインストール

Helmチャートは、HelmがBlack Duckを導入するのに必要なKubernetesのリソースセットを示しています。Black DuckはHelm3をサポートしており、Kubernetesの最小バージョンは1.13です。

Helmチャートは、https://sig-repo.synopsys.com/artifactory/sig-cloudnativeから入手できます

Helmを使用してBlack Duckをインストールする手順については、ここをクリックしてください。Helmチャートは、Helmパッケージマネージャを使用して、Kubernetesクラスタ上でBlack Duckの導入をbootstrapします。

Helmを使用したKubernetes上での移行

PostgreSQL 9.6ベースのBlack Duckバージョンからアップグレードする場合、この移行ではCentOS PostgreSQLコンテナの使用がSynopsys提供のコンテナに置き換えられます。また、synopsys-initコンテナは、blackduck-postgres-waiterコンテナに置き換えられます。

プレーンなKubernetesでは、上書きされない限り、アップグレードジョブのコンテナはルートとして実行されます。ただし、唯一の要件は、ジョブがPostgreSQLデータボリュームの所有者と同じUID（デフォルトではUID=26）で実行されることです。

OpenShiftでは、アップグレードジョブは、PostgreSQLデータボリュームの所有者と同じUIDで実行されることを前提としています。

5. 管理タスク

Kubernetesでのシークレットの暗号化の構成

Black Duckは、システム内で重要なデータの、保存データの暗号化をサポートします。この暗号化は、オーケストレーション環境（Docker SwarmまたはKubernetes）によってBlack Duckインストールにプロビジョニングされたシークレットに基づいています。次に、組織のセキュリティポリシーに基づいてこのシークレットを作成および管理し、バックアップシークレットを作成し、シークレットをローテーションするプロセスの概要を示します。

暗号化される重要なデータは、次のとおりです。

- SCM統合OAuthトークン
- SCM統合プロバイダOAuthアプリケーションクライアントシークレット
- LDAP認証情報
- SAMLプライベート署名証明書

注： シークレットの暗号化は、いったん有効にすると、無効にすることはできません。

暗号化シークレットの概要

暗号化シークレットは、システム内のリソースをアンロックする目的で内部暗号化キーを生成するために使用されるランダムなシーケンスです。Black Duckのシークレットの暗号化は、3つの対称キー、つまりルートキー、バックアップキーおよび以前のキーによって制御されます。これらの3つのキーは、KubernetesおよびDocker SwarmシークレットとしてBlack Duckに渡されたシードによって生成されます。3つのシークレットは、次のように名前が付けられます。

- crypto-root-seed
- crypto-backup-seed
- crypto-prev-seed

通常の状態では、3つのシードはすべて、アクティブな使用中にはなりません。ローテーションアクションが進行中でない限り、アクティブな唯一のシードはルートシードになります。

ルートシードのセキュリティ保護

ルートシードを保護することは重要です。システムデータのコピーとともにルートシードを所有するユーザーは、システムの保護されたコンテンツをアンロックし、読み取る可能性があります。一部のDocker Swarmシステムまたは Kubernetesシステムは、デフォルトでは、保存時のシークレットを暗号化しません。これらのオーケストレーションシステムを内部で暗号化するように構成して、後でシステムに作成されるシークレットが安全に保たれるようにすることを強くお勧めします。

ルートシードは、災害復旧計画の一部としてバックアップからシステム状態を再作成するのに必要です。ルートシードファイルのコピーは、オーケストレーションシステムとは別の秘密の場所に保存して、シードとバックアップの組み合わせでシステムを再作成できるようにする必要があります。ルートシードをバックアップファイルと同じ場所に保存することはお勧めしません。一方のファイルセットが漏洩したり盗まれたりした場合、両方が漏洩したり盗まれたりしたことになります。したがって、バックアップデータ用とシードバックアップ用で別々の場所を用意することをお勧めします。

Kubernetesでのシークレットの暗号化の有効化

Kubernetesでシークレットの暗号化を有効にするには、values.yamlオーケストレーションファイルのenableApplicationLevelEncryptionの値をtrueに変更する必要があります。

```
# if true, enables application level encryption
enableApplicationLevelEncryption: true
```

キーシード管理スクリプト

サンプル管理スクリプトは、Black Duck GitHubパブリックリポジトリで確認できます。

https://github.com/blackducksoftware/secrets-encryption-scripts

このスクリプトは、Black Duckシークレットの暗号化を管理するためではなく、ここに文書化されている、低レベルのDockerおよびKubernetesコマンドの使用を示すためのものです。2つのスクリプトセットがあり、それぞれが専用のサブディレクトリにあります（Kubernetesプラットフォームでの使用、Docker Swarmプラットフォームでの使用に対応しています）。KubernetesおよびDocker Swarm用の個々のスクリプト間に1対1の対応があります（該当する場合）。たとえば、両方のスクリプトセットに次のスクリプトが含まれています。

createInitialSeeds.sh

Kubernetesでのシードの生成

OpenSSLでのシードの生成

シードの内容は、少なくとも1024バイト長の、セキュリティで保護されたランダムな内容を生成する任意のメカニズムを使用して生成できます。シードは、作成され、シークレットに保存されたら、すぐにファイルシステムから削除し、プライベートな、セキュリティで保護された場所に保存する必要があります。

OpenSSLコマンドは、次のとおりです。

```
openssl rand -hex 1024 > root_seed
```

Kubernetesでのシードの生成

シークレットを作成するKubernetesコマンドラインは多数あります。以下にリストされているコマンドにより、シークレットとその変更の有無をより適切に追跡でき、オンラインシステムでシークレットを操作できることとの互換性が保証されます。シークレットは、Black Duckがアクティブに実行されているKubernetesで作成および削除できます。

```
kubectl create secret generic crypto-root-seed -n $NAMESPACE --save-config --dry-run=client --
from-file=crypto-root-seed=./root_seed -o yaml | kubectl apply -f -
```

Kubernetesで前のキーシークレットを削除するには

```
kubectl delete secret crypto-prev-seed -n $NAMESPACE
```

バックアップシードの構成

災害復旧シナリオでシステムを確実に復元できるように、バックアップルートシードを用意することをお勧めします。バックアップルートシードは、システムを復元するために配置できる代替ルートシードです。したがって、ルートシードと同じ方法で安全に保管する必要があります。

バックアップルートシードは、いったんシステムに関連付けられると、ルートシードのローテーションにわたって利用できるという特別な機能がいくつかあります。バックアップシードがシステムによって処理されたら、攻撃および漏洩へ

の暴露を限定するために、シークレットから削除する必要があります。バックアップルートシードは、シークレットがシステム内でどの時点でも「アクティブ」にならないようにする必要があるため、異なる（あまり頻繁ではない）ローテーションスケジュールを持つことができます。

ルートシードをローテーションする必要があるかローテーションしたい場合は、まず、現在のルートシードを前のルートシードとして定義する必要があります。その後、新しいルートシードを生成し、所定の場所に配置できます。

システムがこれらのシードを処理するとき、リソースをローテーションして新しいルートシードを使用するために、前のルートキーが使用されます。この処理の後、前のルートシードをシークレットから削除して、ローテーションを完了し、古いリソースをクリーンアップする必要があります。

バックアップルートシードの作成

バックアップシード/キーは、最初に作成されると、TDEK（テナントの復号化、暗号化キー）低レベルキーをラップします。サンプルスクリプトcreateInitialSeeds.shは、ルートシードとバックアップシードの両方を作成します。Black Duckは、実行されると、両方のキーを使用してTDEKをラップします。

この操作が完了し、ルートシードとバックアップシードの両方が別の場所に安全に保存されたら、バックアップシードシークレットを削除する必要があります。サンプルスクリプトcleanupBackupSeed.shを参照してください。

ルートキーが紛失または漏洩した場合、バックアップキーを使用してルートキーを置き換えることができます。サンプルスクリプトuseRootSeed.shを参照してください。

バックアップシードのローテーション

ルートキーと同様に、バックアップシードは定期的にローテーションする必要があります。ルートシード（古いルートシードが以前のシードシークレットとして保存され、新しいルートシードシークレットがシステムに提示されます）とは異なり、バックアップシードは新しいバックアップシードを作成するだけでローテーションされます。サンプルスクリプトrotateBackupSeed.shを参照してください。

ローテーションが完了したら、新しいバックアップシードを安全に保存し、Black Duckホストファイルシステムから削除する必要があります。

Kubernetesでのシークレットローテーションの管理

組織のセキュリティポリシーに従って、使用中のルートシードを定期的にローテーションすることをお勧めします。これを行うには、ローテーションを実行するために追加のシークレットが必要です。ルートシードをローテーションするために、現在のルートシードが「前のルートシード」として構成され、新しく生成されるルートシードがルートシードとして生成および構成されます。システムがこの構成を処理すると（詳細は以下）、シークレットがローテーションされます。

その時点では、古いシードと新しいシードの両方が、システムの内容をアンロックできます。デフォルトでは、新しいルートシードが使用され、システムが意図したとおりに動作していることをテストおよび確認できます。すべてが確認されたら、「前のルートシード」を削除することで、ローテーションを完了します。

前のルートシードは、システムから削除されると、システムの内容のアンロックに使用できなくなるため、破棄してかまいません。これで、新しいルートシードが適切なルートシードになりました。このルートシードは、適切にバックアップおよびセキュリティ保護する必要があります。

ルートキーは、Black Duckのシークレットを実際に暗号化および復号化する、低レベルのTDEK（テナントの復号化、暗号化キー）をラップするために使用されます。定期的に、Black Duck管理者にとって都合が良く、ユーザー組織のルールに準拠しているタイミングで、ルートキーをローテーションする必要があります。

ルートキーをローテーションする手順としては、現在のルートシードの内容で以前のシードシークレットを作成します。その後、新しいルートシードが作成され、ルートシードシークレットに保存される必要があります。

Kubernetesでのシークレットローテーション

Kubernetesでは、Black Duckを実行しながら、3つの操作を行うことができます。Kubernetesサンプルスクリプトは、ルートシードをprev_rootに抽出し、新しいルートシードを作成してから、前のシードとルートシードを再作成します。

ローテーションが完了したら、前のシードシークレットを削除する必要があります。サンプルスクリプトを参照してください。繰り返しになりますが、このクリーンアップは、実行中のKubernetes Black Duckインスタンスに対して実行できます。

ローテーションの状態は、ユーザーインターフェイスで、[管理者] > [システム情報] > [暗号]の順に移動し、暗号診断タブを表示することで追跡できます。