Red Hat Data Grid 8.5

Helm を使用した Data Grid クラスターのビルドおよびデプロイ

OpenShift で Data Grid クラスターを作成する

Last Updated: 2024-08-12

Red Hat Data Grid 8.5 Helm を使用した Data Grid クラスターのビルドおよびデプロイ

OpenShift で Data Grid クラスターを作成する

法律上の通知

概要

Helm を使用して Data Grid クラスターをビルドし、デプロイします。

目次

RED HAT DATA GRID

Data Grid は、高性能の分散型インメモリーデータストアです。

スキーマレスデータ構造

さまざまなオブジェクトをキーと値のペアとして格納する柔軟性があります。

グリッドベースのデータストレージ

クラスター間でデータを分散および複製するように設計されています。

エラスティックスケーリング

サービスを中断することなく、ノードの数を動的に調整して要件を満たします。

データの相互運用性

さまざまなエンドポイントからグリッド内のデータを保存、取得、およびクエリーします。

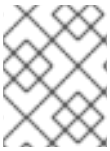
DATA GRID のドキュメント

Data Grid のドキュメントは、Red Hat カスタマーポータルで入手できます。

- Data Grid 8.5 ドキュメント
- Data Grid 8.5 コンポーネントの詳細
- Data Grid 8.5 でサポートされる構成
- Data Grid 8 機能のサポート
- Data Grid で非推奨の機能

DATA GRID のダウンロード

Red Hat カスタマーポータルで Data Grid Software Downloads にアクセスします。



注記

Data Grid ソフトウェアにアクセスしてダウンロードするには、Red Hat アカウントが必要です。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。詳細は、Red Hat CTO である Chris Wright のメッセージ を参照してください。

第1章 HELM チャートリリースとしての DATA GRID クラスターのデプロイメント

Helm を使用して Data Grid クラスターのビルド、設定、およびデプロイを行います。Data Grid は、OpenShift で Data Grid クラスターを実行するためのリソースをパッケージ化する Helm チャートを提供します。

Data Grid チャートをインストールして、Helm リリースを作成します。これにより、OpenShift プロジェクトで Data Grid クラスターがインスタンス化されます。

1.1. OPENSHIFT コンソールを使用した DATA GRID チャートのインストール

OpenShift Web コンソールを使用して、Red Hat 開発者カタログから Data Grid チャートをインストールします。チャートをインストールすると、Data Grid クラスターをデプロイする Helm リリースが作成されます。

前提条件

- OpenShift にアクセスできる。

手順

1. OpenShift Web コンソールにログインします。
2. **Developer** パースペクティブを選択します。
3. **Add** ビューを開き、**Helm Chart** を選択して Red Hat 開発者カタログを参照します。
4. Data Grid チャートを探し、選択します。
5. チャートの名前を指定し、バージョンを選択します。
6. Data Grid チャートの以下のセクションで値を定義します。
   - **Images** は、Data Grid クラスターの Pod を作成する際に使用するコンテナーイメージを設定します。
   - **Deploy** は Data Grid クラスターを設定します。

     ヒント

     各値の説明を見つけるには、**YAML ビュー** オプションを選択し、スキーマにアクセスします。yaml 設定を編集して、Data Grid チャートをカスタマイズします。
7. **Install** を選択します。

検証

1. **Developer** パースペクティブで **Helm** ビューを選択します。
2. 作成した Helm リリースを選択して、詳細、リソース、およびその他の情報を表示します。

1.2. コマンドラインでの DATA GRID チャートのインストール

コマンドラインを使用して OpenShift に Data Grid チャートをインストールし、Data Grid クラスターをインスタンス化します。チャートをインストールすると、Data Grid クラスターをデプロイする Helm リリースが作成されます。

前提条件

- **helm** クライアントをインストールしている。
- OpenShift Helm チャートリポジトリー を追加している。
- OpenShift クラスターにアクセスできる。
- **oc** クライアントがある。

手順

1. Data Grid クラスターを設定する values ファイルを作成します。
   たとえば、以下の values ファイルは、2 つのノードで設定されるクラスターを作成します。

```
$ cat > infinispan-values.yaml<<EOF
#Build configuration
images:
  server: registry.redhat.io/datagrid/datagrid-8-rhel8:latest
  initContainer: registry.access.redhat.com/ubi8-micro
#Deployment configuration
deploy:
  #Add a user with full security authorization.
  security:
    batch: "user create admin -p changeme"
  #Create a cluster with two pods.
  replicas: 2
  #Specify the internal Kubernetes cluster domain.
  clusterDomain: cluster.local
EOF
```

2. Data Grid チャートをインストールし、values ファイルを指定します。

```
$ helm install infinispan openshift-helm-charts/redhat-data-grid --values infinispan-values.yaml
```

ヒント

--set フラグを使用して、デプロイメントの設定値を上書きします。たとえば、3 つのノードで設定されるクラスターを作成するには、以下のように設定します。

```
--set deploy.replicas=3
```

検証

Pod を監視して、Data Grid クラスターのすべてのノードが正常に作成されていることを確認します。

```
$ oc get pods -w
```

1.3. DATA GRID HELM リリースのアップグレード

Helm リリースをアップグレードして、実行時に Data Grid クラスター設定を変更します。

前提条件

- Data Grid チャートをデプロイしている。
- **helm** クライアントがある。
- **oc** クライアントがある。

手順

1. 適宜、Data Grid デプロイメントの values ファイルを変更します。
2. **helm** クライアントを使用して変更を適用します。以下に例を示します。

```
$ helm upgrade infinispan openshift-helm-charts/redhat-data-grid --values infinispan-values.yaml
```

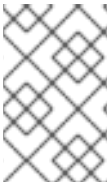
検証

Pod の再ビルドを監視して、すべての変更が Data Grid クラスターに正常に適用されているのを確認します。

```
$ oc get pods -w
```

1.4. DATA GRID HELM リリースのアンインストール

Data Grid チャートのリリースをアンインストールし、Pod およびその他のデプロイメントアーティファクトを削除します。



注記

この手順では、コマンドラインで Data Grid デプロイメントをアンインストールする方法を説明しますが、代わりに OpenShift Web コンソールを使用することもできます。特定の手順は、OpenShift のドキュメントを参照してください。

前提条件

- Data Grid チャートをデプロイしている。
- **helm** クライアントがある。
- **oc** クライアントがある。

手順

1. インストールされている Data Grid Helm リリースをリスト表示します。

```
$ helm list
```

2. **helm** クライアントを使用してリリースをアンインストールし、Data Grid クラスターを削除します。

```
$ helm uninstall <helm_release_name>
```

3. **oc** クライアントを使用して、生成されたシークレットを削除します。

```
$ oc delete secret <helm_release_name>-generated-secret
```

1.5. デプロイメント設定の値

デプロイメント設定の値により、Data Grid クラスターをカスタマイズできます。

ヒント

Data Grid チャートの README で、フィールドおよび値の説明を確認することもできます。

フィールド	説明	デフォルト値
deploy.clusterDomain	内部 Kubernetes クラスタードメインを指定します。	**cluster.local**
deploy.replicas	Pod を作成する Data Grid クラスター内のノードの数を指定します。	**1**
deploy.container.extraJvmOpts	JVM オプションを Data Grid Server に渡します。	デフォルト値はありません。
deploy.container.libraries	サーバー起動前にダウンロードするライブラリー。URL または Maven 座標として表される複数のアーティファクトをスペースで区切って指定します。.tar、.tar.gz、または .zip 形式のアーカイブアーティファクトが抽出されます。	デフォルト値はありません。
deploy.container.storage.ephemeral	ストレージが一時的または永続的であるかどうかを定義します。	デフォルト値は **false** で、データが永続的であることを意味します。一時ストレージを使用するには、値を **true** に設定します。これは、クラスターのシャットダウンまたは再起動時に、すべてのデータが削除されることを意味します。
deploy.container.storage.size	各 Data Grid Pod に割り当てられるストレージの量を定義します。	1Gi

フィールド	説明	デフォルト値
deploy.container.storage.storageClassName	永続ボリューム要求 (PVC) に使用する **StorageClass** オブジェクトの名前を指定します。	デフォルト値はありません。デフォルトでは、永続ボリューム要求は **storageclass.kubernetes.io/is-default-class** アノテーションが **true** に設定されたストレージクラスを使用します。このフィールドを含める場合は、既存のストレージクラスを値として指定する必要があります。
deploy.container.resources.limits.cpu	各 Data Grid Pod の CPU の制限を CPU 単位で定義します。	500 m
deploy.container.resources.limits.memory	各 Data Grid Pod のメモリーの最大量をバイト単位で定義します。	512 Mi
deploy.container.resources.requests.cpu	各 Data Grid Pod について、最大の CPU リクエストを CPU 単位で指定します。	500 m
deploy.container.resources.requests.memory	各 Data Grid Pod について、最大のメモリー要求をバイト単位で指定します。	512 Mi
deploy.security.secretName	認証情報を作成し、セキュリティー承認を設定するシークレットの名前を指定します。	デフォルト値はありません。カスタムのセキュリティーシークレットを作成する場合、**deploy.security.batch** は有効ではありません。
deploy.security.batch	起動時に認証情報を作成しセキュリティー承認を設定する、Data Grid コマンドラインインターフェイス (CLI) のバッチファイルを指定します。	デフォルト値はありません。
deploy.expose.type	ネットワークに Hot Rod および REST エンドポイントを公開し、Data Grid コンソールなどの Data Grid クラスターへのアクセスを提供するサービスを指定します。	**Route** Valid オプション:**""** (空の値)、**Route**、**LoadBalancer**、および **NodePort**ネットワークに Data Grid を公開したくない場合は、空の値 (**""**) を設定します。
deploy.expose.nodePort	30000 から 32767 のデフォルト範囲で、ノードポートサービスのネットワークポートを指定します。	0 ポートを指定しないと、プラットフォームは利用可能なポートを選択します。

フィールド	説明	デフォルト値
deploy.expose.host	オプションとして、Route が公開されるホスト名を指定します。	デフォルト値はありません。
deploy.expose.annotations	ネットワークに Data Grid を公開するサービスにアノテーションを追加します。	デフォルト値はありません。
deploy.logging.categories	Data Grid クラスターのログカテゴリーおよびレベルを設定します。	デフォルト値はありません。
deploy.podLabels	作成する各 Data Grid Pod にラベルを追加します。	デフォルト値はありません。
deploy.svcLabels	作成する各サービスにラベルを追加します。	デフォルト値はありません。
deploy.resourceLabels	Pod およびサービスを含むすべての Data Grid リソースにラベルを追加します。	デフォルト値はありません。
deploy.makeDataDirWritable	各 Data Grid Server ノードの **data** ディレクトリーへの書き込みアクセスを許可します。	**false** 値を **true** に設定すると、Data Grid は、パーミッションを変更するために **/opt/infinispan/server/data** ディレクトリーで **chmod -R** を実行する initContainer を作成します。
deploy.securityContext	StatefulSet Pod で使用される securityContext を設定します。	**{}** これを使用して、マウントされたファイルシステムのグループを変更できます。**/opt/infinispan/server/data** のグループ所有者を、デフォルトの Data Grid のグループと明示的に一致させる必要がある場合は、**securityContext.fsGroup** を **185** に設定します。
deploy.monitoring.enabled	**ServiceMonitor** を使用して監視を有効または無効にします。	**false ServiceMonitor** を有効にして Helm チャートをデプロイするには、管理者によって割り当てられた **monitoring-edit** ロールが必要です。
deploy.nameOverride	すべての Data Grid クラスターリソースの名前を指定します。	Helm チャートのリリース名。

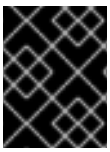
フィールド	説明	デフォルト値
deploy.infinispan	Data Grid Server の設定。	Data Grid は、デフォルトのサーバー設定を提供します。サーバーインスタンスの設定に関する詳細は Data Grid Server の設定値 を参照してください。

第2章 DATA GRID SERVER の設定

カスタム Data Grid Server 設定をデプロイメントに適用します。

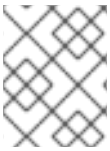
2.1. DATA GRID SERVER 設定のカスタマイズ

カスタムの **deploy.infinispan** 値を適用する Data Grid クラスターは、Cache Manager およびセキュリティーレルムまたは Hot Rod および REST エンドポイントなどの基礎となるサーバーメカニズムを設定します。



重要

deploy.infinispan 値を変更する場合は、常にすべての Data Grid Server 設定を指定する必要があります。



注記

Data Grid クラスターのモニタリング機能を使用する場合は、デフォルトの "metrics" 設定を変更または削除しないでください。

手順

必要に応じて Data Grid Server 設定を変更します。

- **deploy.infinispan.cacheContainer** フィールドで、Cache Manager の設定値を指定します。たとえば、任意の Data Grid 設定で起動時にキャッシュを作成したり、キャッシュテンプレートを追加し、それを使用してオンデマンドでキャッシュを作成したりできます。
- **deploy.infinispan.cacheContainer.security.authorization** フィールドで、ユーザーロールとパーミッションを制御するセキュリティー承認を設定します。
- デフォルトの JGroups スタックのいずれかを選択するか、**deploy.infinispan.cacheContainer.transport** フィールドでクラスタートランスポートを設定します。
- **deploy.infinispan.server.endpoints** フィールドで、Data Grid Server エンドポイントを設定します。
- **deploy.infinispan.server.interfaces** フィールドおよび **deploy.infinispan.server.socketBindings** フィールドで、Data Grid Server のネットワークインターフェイスおよびポートを設定します。
- **deploy.infinispan.server.security** フィールドで、Data Grid Server のセキュリティーメカニズムを設定します。

2.2. DATA GRID SERVER の設定値

Data Grid Server の設定値で、Cache Manager をカスタマイズし、OpenShift Pod で実行するサーバーインスタンスを変更できます。

Data Grid Server の設定

```
deploy:
  infinispan:
```

```
cacheContainer:
# [USER] Add cache, template, and counter configuration.
name: default
# [USER] Specify `security: null` to disable security authorization.
security:
  authorization: {}
transport:
  cluster: ${infinispan.cluster.name:cluster}
  node-name: ${infinispan.node.name:}
  stack: kubernetes
server:
  endpoints:
    # [USER] Hot Rod and REST endpoints.
    - securityRealm: default
      socketBinding: default
    # [METRICS] Metrics endpoint for cluster monitoring capabilities.
    - connectors:
        rest:
          restConnector:
            authentication:
              mechanisms: BASIC
      securityRealm: metrics
      socketBinding: metrics
  interfaces:
  - inetAddress:
      value: ${infinispan.bind.address:127.0.0.1}
    name: public
  security:
    credentialStores:
    - clearTextCredential:
        clearText: secret
      name: credentials
      path: credentials.pfx
    securityRealms:
    # [USER] Security realm for the Hot Rod and REST endpoints.
    - name: default
      # [USER] Comment or remove this properties realm to disable authentication.
      propertiesRealm:
        groupProperties:
          path: groups.properties
        groupsAttribute: Roles
        userProperties:
          path: users.properties
      # [METRICS] Security realm for the metrics endpoint.
    - name: metrics
      propertiesRealm:
        groupProperties:
          path: metrics-groups.properties
          relativeTo: infinispan.server.config.path
        groupsAttribute: Roles
        userProperties:
          path: metrics-users.properties
          plainText: true
          relativeTo: infinispan.server.config.path
    socketBindings:
      defaultInterface: public
```

```
      portOffset: ${infinispan.socket.binding.port-offset:0}
      socketBinding:
        # [USER] Socket binding for the Hot Rod and REST endpoints.
      - name: default
        port: 11222
        # [METRICS] Socket binding for the metrics endpoint.
      - name: metrics
        port: 11223
```

Data Grid キャッシュの設定

```
deploy:
  infinispan:
    cacheContainer:
      distributedCache:
        name: "mycache"
        mode: "SYNC"
        owners: "2"
        segments: "256"
        capacityFactor: "1.0"
        statistics: "true"
        encoding:
          mediaType: "application/x-protostream"
        expiration:
          lifespan: "5000"
          maxIdle: "1000"
        memory:
          maxCount: "1000000"
          whenFull: "REMOVE"
        partitionHandling:
          whenSplit: "ALLOW_READ_WRITES"
          mergePolicy: "PREFERRED_NON_NULL"
    #Provide additional Cache Manager configuration.
  server:
    #Provide configuration for server instances.
```

キャッシュテンプレート

```
deploy:
  infinispan:
    cacheContainer:
      distributedCacheConfiguration:
        name: "my-dist-template"
        mode: "SYNC"
        statistics: "true"
        encoding:
          mediaType: "application/x-protostream"
        expiration:
          lifespan: "5000"
          maxIdle: "1000"
        memory:
          maxCount: "1000000"
          whenFull: "REMOVE"
```

```
    #Provide additional Cache Manager configuration.
  server:
    #Provide configuration for server instances.
```

クラスタートランスポート

```
deploy:
  infinispan:
    cacheContainer:
      transport:
        #Specifies the name of a default JGroups stack.
        stack: kubernetes
    #Provide additional Cache Manager configuration.
  server:
    #Provide configuration for server instances.
```

関連情報

- Data Grid Server ガイド
- Data Grid の設定

第3章 認証および承認の設定

認証情報を追加し、異なるパーミッションを持つロールを割り当てて、Data Grid クラスターへのアクセスを制御します。

3.1. デフォルトの認証情報

Data Grid は、デフォルトの認証情報を **<helm_release_name>-generated-secret** シークレットに追加します。

ユーザー名	説明
developer	Data Grid リソースへのフルアクセスを持つ **admin** ロールを持つユーザー。
monitor	ポート **11223** を介した Data Grid メトリックにアクセスできる **monitor** ロールを持つ内部ユーザー。

関連情報

- Data Grid セキュリティーガイド

3.1.1. 認証情報の取得

認証シークレットから Data Grid の認証情報を取得します。

前提条件

- Data Grid Helm チャートをインストールしている。
- **oc** クライアントがある。

手順

- 以下のコマンドを使用して、**<helm_release_name>-generated-secret** からデフォルトの認証情報を取得するか、別のシークレットからカスタムの認証情報を取得します。

```
$ oc get secret <helm_release_name>-generated-secret \
-o jsonpath="{.data.identities-batch}" | base64 --decode
```

3.2. カスタムのユーザー認証情報またはクレデンシャルストアの追加

Data Grid ユーザー認証情報を作成し、クラスターアクセスのセキュリティー承認を付与するロールを割り当てます。

手順

- **deploy.security.batch** フィールドに **user create** コマンドを指定して、認証情報を作成します。

  暗黙的な承認を持つユーザー

```
deploy:
  security:
    batch: 'user create admin -p changeme'
```

特定のロールを持つユーザー

```
deploy:
  security:
    batch: 'user create personone -p changeme -g deployer'
```

3.2.1. ユーザーロールとパーミッション

Data Grid はロールベースのアクセス制御を使用して、ユーザーがクラスターリソースおよびデータにアクセスするのを承認します。セキュリティーを強化するには、認証情報を追加する際に Data Grid ユーザーに適切なロールを付与する必要があります。

Role	パーミッション	説明
admin	ALL	Cache Manager ライフサイクルの制御など、すべてのパーミッションを持つスーパーユーザー。
deployer	ALL_READ、ALL_WRITE、LISTEN、EXEC、MONITOR、CREATE	**application** パーミッションに加えて、Data Grid リソースを作成および削除できます。
application	ALL_READ、ALL_WRITE、LISTEN、EXEC、MONITOR	**observer** パーミッションに加え、Data Grid リソースへの読み取りおよび書き込みアクセスがあります。また、イベントをリッスンし、サーバータスクおよびスクリプトを実行することもできます。
observer	ALL_READ、MONITOR	**monitor** パーミッションに加え、Data Grid リソースへの読み取りアクセスがあります。
monitor	MONITOR	Data Grid クラスターの統計を表示できます。

関連情報

- Data Grid セキュリティーガイド

3.2.2. クレデンシャルストアの追加

サーバー設定の ConfigMap でパスワードがクリアテキストで公開されるのを防ぐために、Data Grid クレデンシャルストアを作成します。ユースケースについては、「TLS 暗号化の有効化」 を参照してください。

手順

1. **deploy.security.batch** フィールドに **credentials add** コマンドを指定して、クレデンシャルストアを作成します。

   ストアへのパスワードの追加

```
deploy:
  security:
    batch: 'credentials add keystore -c password -p secret --path="credentials.pfx"'
```

2. 次に、クレデンシャルストアをサーバー設定に追加する必要があります。

   クレデンシャルストアの設定

```
deploy:
  infinispan:
    server:
      security:
        credentialStores:
          - name: credentials
            path: credentials.pfx
            clearTextCredential:
              clearText: "secret"
```

3.2.3. 認証シークレットを使用した複数の認証情報の追加

認証シークレットを使用して、複数の認証情報を Data Grid クラスターに追加します。

前提条件

- **oc** クライアントがある。

手順

1. 認証情報を追加するコマンドが含まれる **identities-batch** ファイルを作成します。

```
apiVersion: v1
kind: Secret
metadata:
  name: connect-secret
type: Opaque
stringData:
  # The "monitor" user authenticates with the Prometheus ServiceMonitor.
  username: monitor
  # The password for the "monitor" user.
  password: password
  # The key must be 'identities-batch'.
  # The content is "user create" commands for the Data Grid CLI.
  identities-batch: |-
    user create user1 -p changeme -g admin
    user create user2 -p changeme -g deployer
```

```
    user create monitor -p password --users-file metrics-users.properties --groups-file metrics-
groups.properties
    credentials add keystore -c password -p secret --path="credentials.pfx"
```

2. **identities-batch** ファイルから認証シークレットを作成します。

```
$ oc apply -f identities-batch.yaml
```

3. **deploy.security.SecretName** フィールドに認証シークレットを指定します。

```
deploy:
  security:
    authentication: true
    secretName: 'connect-secret'
```

4. Data Grid Helm リリースをインストールまたはアップグレードします。

3.3. 認証の無効化

ユーザーが Data Grid クラスターにアクセスでき、認証情報を提供せずにデータを操作できるようにします。

重要

OpenShift クラスターの外部からエンドポイントにアクセスできる場合は、認証を無効にしないでください。開発環境の認証のみを無効にする必要があります。

手順

1. "default" セキュリティーレルムから **propertiesRealm** フィールドを削除します。
2. Data Grid Helm リリースをインストールまたはアップグレードします。

3.4. セキュリティー承認の無効化

Data Grid ユーザーがロールに関係なく任意の操作を実行できるようにします。

手順

1. **null** を **deploy.infinispan.cacheContainer.security** フィールドの値として設定します。

ヒント

helm クライアントで **--set deploy.infinispan.cacheContainer.security=null** 引数を使用します。

2. Data Grid Helm リリースをインストールまたはアップグレードします。

第4章 暗号化の設定

Data Grid の暗号化を設定します。

4.1. TLS 暗号化の有効化

暗号化は、エンドポイントとクラスタートランスポートに対して個別に有効にできます。

前提条件

- 証明書またはキーストアを含むシークレット。エンドポイントとクラスターで別々のシークレットを使用してください。
- キーストアにアクセスするために必要なパスワードを含むクレデンシャルキーストア。クレデンシャルキーストアの追加 を参照してください。

手順

1. デプロイ設定でシークレット名を設定します。
キーストアを含むシークレットの名前を指定します。

```
deploy:
  ssl:
    endpointSecretName: "tls-secret"
    transportSecretName: "tls-transport-secret"
```

2. クラスタートランスポートの TLS を有効にします。

```
deploy:
  infinispan:
    cacheContainer:
      transport:
        urn:infinispan:server:15.0:securityRealm: >
          "cluster-transport" 1
    server:
      security:
        securityRealms:
          - name: cluster-transport
            serverIdentities:
              ssl:
                keystore: 2
                  alias: "server"
                  path: "/etc/encrypt/transport/cert.p12"
                  credentialReference: 3
                    store: credentials
                    alias: keystore
                truststore: 4
                  path: "/etc/encrypt/transport/cert.p12"
                  credentialReference: 5
                    store: credentials
                    alias: truststore
```

1. 指定のセキュリティーレルムを使用してクラスターの暗号化を提供するようにトランスポートスタックを設定します。
2. トランスポートレルムでキーストアパスを設定します。シークレットは **/etc/encrypt/transport** にマウントされます。
3. 5. 同じキーストアを使用してトラストストアを設定し、ノードが相互に認証できるようにします。
4. シークレットにキーストアが含まれている場合、エイリアスとパスワードを指定する必要があります。

3. エンドポイントの TLS を有効にします。

```
deploy:
  infinispan:
    server:
      security:
        securityRealms:
          - name: default
            serverIdentities:
              ssl:
                keystore:
                  path: "/etc/encrypt/endpoint/keystore.p12" 1
                  alias: "server" 2
                  credentialReference:
                    store: credentials 3
                    alias: keystore 4
```

1. エンドポイントレルムでキーストアパスを設定します。シークレットは **/etc/encrypt/endpoint** にマウントされます。
2. シークレットにキーストアが含まれている場合、エイリアスを指定する必要があります。
3. 4. パスワードはクレデンシャルキーストア経由で提供する必要があります。

関連情報

- Data Grid セキュリティーガイド

第5章 ネットワークアクセスの設定

Data Grid デプロイメントのネットワークアクセスを設定し、内部ネットワークサービスを確認します。

5.1. ネットワークへの DATA GRID クラスターの公開

Data Grid コンソールならびに REST および Hot Rod エンドポイントにアクセスできるように、ネットワークで Data Grid クラスターが利用できるようにします。デフォルトでは、Data Grid チャートはルートを介してデプロイメントを公開しますが、ロードバランサーまたはノードポートを介してクラスターを公開するように設定できます。また、デプロイメントをネットワークに公開せず、内部的に OpenShift クラスターだけが利用できるように Data Grid チャートを設定することもできます。

手順

1. **deploy.expose.type** フィールドに、以下のいずれかを指定します。

オプション	説明
Route	ルートを使用して Data Grid を公開します。これはデフォルト値です。
LoadBalancer	ロードバランサーサービスを介して Data Grid を公開します。
NodePort	ノードポートサービスを介して Data Grid を公開します。
""(空の値)	ネットワークに Data Grid を公開するのを無効にします。

2. ルートを介して Data Grid を公開する場合は、オプションとして **deploy.expose.host** フィールドでホスト名を指定します。
3. ノードポートサービスを介して Data Grid を公開する場合は、オプションとして **deploy.expose.nodePort** フィールドでポートを指定します。
4. Data Grid Helm リリースをインストールまたはアップグレードします。

5.2. ネットワークサービスの詳細の取得

Data Grid クラスターに接続できるように、ネットワークサービスの詳細を取得します。

前提条件

- ネットワークに Data Grid クラスターを公開している。
- **oc** クライアントがある。

手順

以下のコマンドのいずれかを使用して、ネットワークサービスの詳細を取得します。

- ルートを使用して Data Grid を公開する場合:

```
$ oc get routes
```

- ロードバランサーまたはノードポートサービスを介して Data Grid を公開する場合:

```
$ oc get services
```

5.3. ネットワークサービス

Data Grid のチャートは、内部アクセス用のデフォルトのネットワークサービスを作成します。

サービス	ポート	プロトコル	説明
<helm_release_name>	**11222**	TCP	Data Grid Hot Rod および REST エンドポイントへのアクセスを提供します。
<helm_release_name>	**11223**	TCP	Data Grid メトリックへのアクセスを提供します。
<helm_release_name>-ping	**8888**	TCP	Data Grid Pod 同士が相互に検出し、クラスターを形成できるようにします。

以下のように、内部ネットワークサービスの詳細を取得できます。

```
$ oc get services

NAME              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
infinispan        ClusterIP   192.0.2.0        <none>        11222/TCP,11223/TCP
infinispan-ping   ClusterIP   None             <none>        8888/TCP
```

第6章 DATA GRID クラスターへの接続

Data Grid クラスターの設定およびデプロイ後に、Data Grid コンソール、コマンドラインインターフェイス (CLI)、Hot Rod クライアント、または REST API を使用してリモート接続を確立できます。

6.1. DATA GRID コンソールへのアクセス

コンソールにアクセスして、キャッシュの作成、管理操作の実行、および Data Grid クラスターの監視を行います。

前提条件

- ネットワークに Data Grid クラスターを公開している。
- ネットワークサービスの詳細を取得している。

手順

- **$SERVICE_HOSTNAME:$PORT** で任意のブラウザーから Data Grid コンソールにアクセスします。
  $SERVICE_HOSTNAME:$PORT を、ネットワーク上で Data Grid を使用できるホスト名とポートに置き換えます。

6.2. コマンドラインインターフェイス (CLI) を使用した接続

Data Grid CLI を使用してクラスターに接続し、キャッシュの作成、データの操作、管理操作を行います。

前提条件

- ネットワークに Data Grid クラスターを公開している。
- ネットワークサービスの詳細を取得している。
- Data Grid ソフトウェアダウンロード から、ネイティブ Data Grid CLI ディストリビューションをダウンロードします。
- ネイティブ Data Grid CLI ディストリビューションの **.zip** アーカイブをホストファイルシステムにデプロイメントします。

手順

1. **-c** 引数の値としてネットワークサービスを指定して、Data Grid CLI を起動します。以下に例を示します。

```
$ {native_cli} -c http://cluster-name-myroute.hostname.net/
```

2. プロンプトが表示されたら、Data Grid の認証情報を入力します。
3. 必要に応じて CLI 操作を実行します。

ヒント

Tab キーを押すか、**--help** 引数を使用して、利用可能なオプションとヘルプテキストを表示します。

4. **quit** コマンドを使用して CLI を終了します。

関連情報

- Data Grid コマンドラインインターフェイスの使用

6.3. OPENSHIFT で実行されている HOT ROD クライアントの接続

Data Grid クラスターと同じ OpenShift クラスターで実行している Hot Rod クライアントを使用して、リモートキャッシュにアクセスします。

前提条件

- ネットワークサービスの詳細を取得している。

手順

1. クライアント設定で、Data Grid クラスターの内部ネットワークサービスの詳細を指定します。以下の設定例では、**$SERVICE_HOSTNAME:$PORT** は、Data Grid クラスターへのアクセスが許可されるホスト名およびポートを示します。

2. クライアントが Data Grid に対して認証できるように、認証情報を指定します。

3. 必要に応じてクライアントのインテリジェンスを設定します。
OpenShift 上で実行される Hot Rod クライアントは、Data Grid Pod の内部 IP アドレスにアクセスできるため、任意のクライアントのインテリジェンスを使用できます。
デフォルトのインテリジェンスである **HASH_DISTRIBUTION_AWARE** が推奨されます。これにより、クライアントはリクエストをプライマリーオーナーにルーティングできるようになり、パフォーマンスが向上します。

プログラムによる設定

```
import org.infinispan.client.hotrod.configuration.ConfigurationBuilder;
import org.infinispan.client.hotrod.configuration.SaslQop;
import org.infinispan.client.hotrod.impl.ConfigurationProperties;
...

ConfigurationBuilder builder = new ConfigurationBuilder();
    builder.addServer()
            .host("$SERVICE_HOSTNAME")
            .port(ConfigurationProperties.DEFAULT_HOTROD_PORT)
           .security().authentication()
            .username("username")
            .password("changeme")
            .realm("default")
            .saslQop(SaslQop.AUTH)
            .saslMechanism("SCRAM-SHA-512");
```

Hot Rod クライアントプロパティー

```
# Connection
infinispan.client.hotrod.server_list=$SERVICE_HOSTNAME:$PORT

# Authentication
infinispan.client.hotrod.use_auth=true
infinispan.client.hotrod.auth_username=developer
infinispan.client.hotrod.auth_password=$PASSWORD
infinispan.client.hotrod.auth_server_name=$CLUSTER_NAME
infinispan.client.hotrod.sasl_properties.javax.security.sasl.qop=auth
infinispan.client.hotrod.sasl_mechanism=SCRAM-SHA-512
```

関連情報

- Hot Rod Java クライアントガイド

6.3.1. すべての Data Grid Pod の IP アドレスを取得する

実行中の Data Grid Pod の全 IP アドレスを含むリストを取得できます。

重要

OpenShift で実行されている Hot Rod クライアントの接続 が推奨される方法です。利用可能ないずれかの Pod への初期接続が確保されるためです。

手順

次の方法で、実行中の Data Grid Pod の全 IP アドレスを取得します。

- OpenShift API を使用する場合:
  - **${APISERVER}/api/v1/namespaces/<chart-namespace>/endpoints/<helm-release-name>** にアクセスして、**<helm-release-name>** サービスに関連付けられている **endpoints** OpenShift リソースを取得します。
- OpenShift DNS サービスを使用する場合:
  - DNS サービスに **<helm-release-name>-ping** という名前をクエリーして、クラスター内の全ノードの IP を取得します。

関連情報

- Accessing the Kubernetes API from a Pod
- DNS for Services and Pods

6.4. OPENSHIFT の外部で実行されている HOT ROD クライアントの接続

Data Grid クラスターをデプロイしている OpenShift クラスターの外部で実行されている Hot Rod クライアントを使用して、リモートキャッシュにアクセスします。

前提条件

- ネットワークに Data Grid クラスターを公開している。

- ネットワークサービスの詳細を取得している。

手順

1. クライアント設定で、Data Grid クラスターの内部ネットワークサービスの詳細を指定します。以下の設定例では、**$SERVICE_HOSTNAME:$PORT** は、Data Grid クラスターへのアクセスが許可されるホスト名およびポートを示します。
2. クライアントが Data Grid に対して認証できるように、認証情報を指定します。
3. **BASIC** インテリジェンスを使用するようにクライアントを設定します。

プログラムによる設定

```
import org.infinispan.client.hotrod.configuration.ClientIntelligence;
import org.infinispan.client.hotrod.configuration.ConfigurationBuilder;
import org.infinispan.client.hotrod.configuration.SaslQop;
...

ConfigurationBuilder builder = new ConfigurationBuilder();
    builder.addServer()
            .host("$SERVICE_HOSTNAME")
            .port("$PORT")
           .security().authentication()
            .username("username")
            .password("changeme")
            .realm("default")
            .saslQop(SaslQop.AUTH)
            .saslMechanism("SCRAM-SHA-512");
    builder.clientIntelligence(ClientIntelligence.BASIC);
```

Hot Rod クライアントプロパティー

```
# Connection
infinispan.client.hotrod.server_list=$SERVICE_HOSTNAME:$PORT

# Client intelligence
infinispan.client.hotrod.client_intelligence=BASIC

# Authentication
infinispan.client.hotrod.use_auth=true
infinispan.client.hotrod.auth_username=developer
infinispan.client.hotrod.auth_password=$PASSWORD
infinispan.client.hotrod.auth_server_name=$CLUSTER_NAME
infinispan.client.hotrod.sasl_properties.javax.security.sasl.qop=auth
infinispan.client.hotrod.sasl_mechanism=SCRAM-SHA-512
```

関連情報

- Hot Rod Java クライアントガイド

6.5. REST API へのアクセス

Data Grid は、HTTP クライアントを使用して対話できる RESTful インターフェイスを提供します。

前提条件

- ネットワークに Data Grid クラスターを公開している。
- ネットワークサービスの詳細を取得している。

手順

- **$SERVICE_HOSTNAME:$PORT/rest/v2** の任意の HTTP クライアントで REST API にアクセスします。
  $SERVICE_HOSTNAME:$PORT を、ネットワーク上で Data Grid を使用できるホスト名とポートに置き換えます。

関連情報

- Data Grid REST API