

第5章: ICP Kubernetes環境での アプリケーション運用

章内目次

第5章: ICP Kubernetes環境でのアプリケーション運用

第1節: アプリケーション運用の概略

第2節: アプリケーションのリポジトリ登録

第3節: アプリケーションのリリース

第4節: アプリケーションの構成変更

第5節: アプリケーションの廃棄

第5章 第1節: アプリケーション運用の概略

本章（ICP Kubernetes環境でのアプリケーション運用）の解説範囲

- 本章（ICP Kubernetes環境でのアプリケーション運用）では、ICP Kubernetes環境でのアプリケーション運用を題材として、以下の趣旨での解説を行う
 - 「ICP製品」や「Kubernetes製品」の初心者を読者として想定した上で、ICP Kubernetes環境上にアプリケーションをリリース・運用するための最低限の知識を、整理された形でまとめる
- アプリケーション運用に関連する以下の話題は、本章での解説の範囲外とする
 - アプリケーションの開発・ビルド・テスト、および、アプリケーション設計の最適化
 - アプリケーションに対するロギング・モニタリング
 - CI/CD関連ソリューション（例: [Microclimate](#)）と連携する形でのアプリケーション運用
 - アプリケーション運用のための各種サポートツール（例: [Kustomize](#)）の紹介、解説

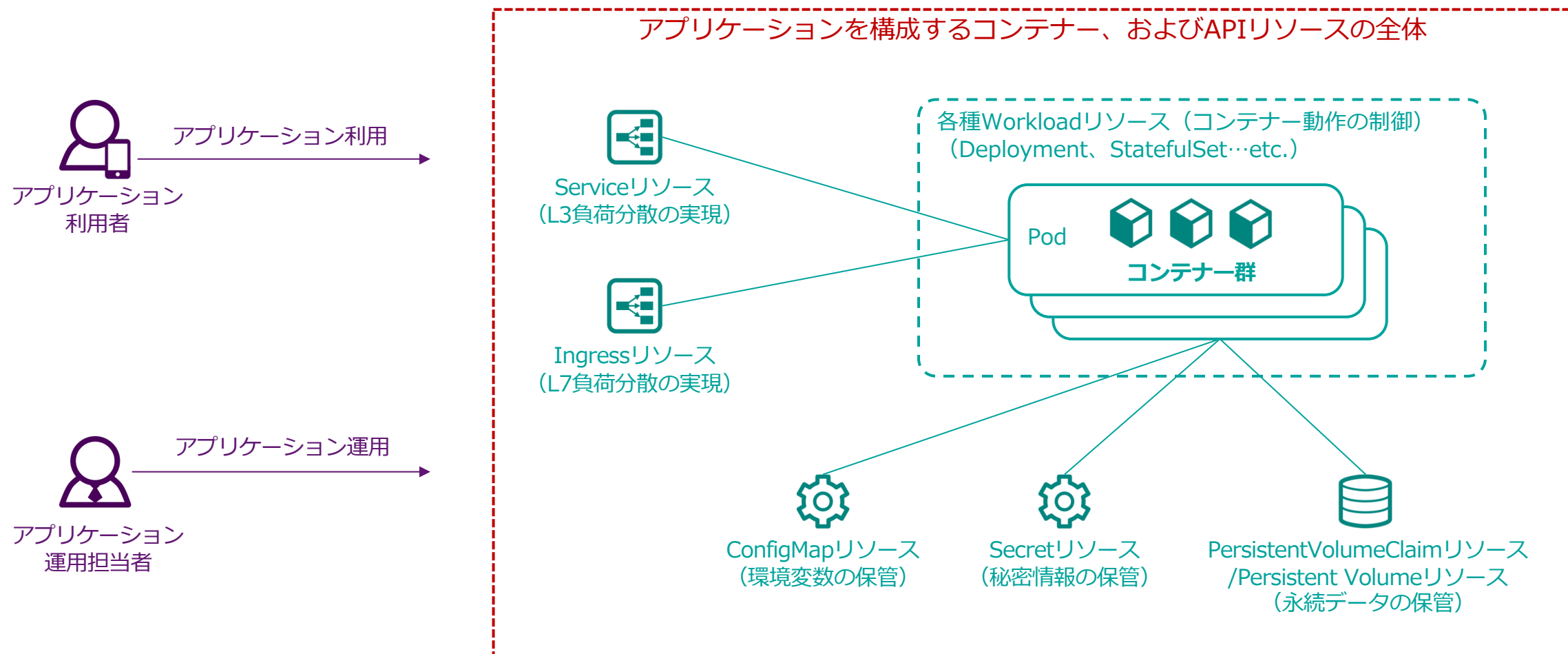
アプリケーションに関わる運用作業項目

- アプリケーションを稼働させるにあたって必要とされる主な運用作業項目、および各運用項目に対する本章での解説方針を下表にまとめる

運用項目	運用の概略	本章での解説方針
開発・ビルド・テスト	- アプリケーション・ソフトウェアを開発・ビルド・テストする - ソフトウェアをコンテナ・イメージにパッケージする - コンテナ・イメージとKubernetes APIリソースを組み合わせる形で、アプリケーションをパッケージ化する - アプリケーション・パッケージをテストする	- 解説の対象外とする - ただし、例外としてHelm Chartを用いたアプリケーションのパッケージ化の概略を本節で解説する
リポジトリ登録	コンテナ・イメージ、およびアプリケーション・パッケージ（Helm Chart）をリポジトリに登録する	解説の対象とする（5.2節）
リリース	アプリケーションをICP Kubernetes環境にリリースする	解説の対象とする（5.3節）
構成変更	アプリケーションの構成変更を行う	解説の対象とする（5.4節）
廃棄	- リリースしたアプリケーションを廃棄する - リポジトリに登録したコンテナ・イメージ、およびアプリケーション・パッケージ（Helm Chart）を削除する	解説の対象とする（5.5節）
モニタリング・レポーティング	リリースしたアプリケーションに対するモニタリング、およびレポーティングを行う	解説の対象外とする
停止・起動	アプリケーションを起動、もしくは停止する	- 環境やノードの保守に伴う停止・起動運用については本ガイドの別章（4.1節）で解説を行う - 構成変更に伴うローリング・アップデート運用については、本章での解説の対象とする（5.4節）
バックアップ・リストア	リリースしたアプリケーション、および、アプリケーションが取り扱う業務データに対するバックアップ・リストアを行う	本ガイドの別章（4.2節）で解説を行う

ICP Kubernetes環境での「アプリケーション」の実体

- Kubernetes環境での「アプリケーション」の実体は、コンテナ、およびコンテナの稼働を支援するための、多種多様なAPIリソースの組み合わせからなる



Helm Chartを用いたアプリケーションのパッケージ化

- 「Helm」とはKubernetes環境向けのパッケージ・マネージャー機能である
- 「Helm Chart」とは「Helm」機能を用いてパッケージ化されたアプリケーション・パッケージを指す
 - ICP製品では、運用するアプリケーションをパッケージとして扱うための手段として、「Helm Chart」の利用をサポートしている（ただし、Helm Chartの利用は任意である）
 - 参考情報: [IBM Knowledge Center - チャートおよびアプリケーションの管理](#)



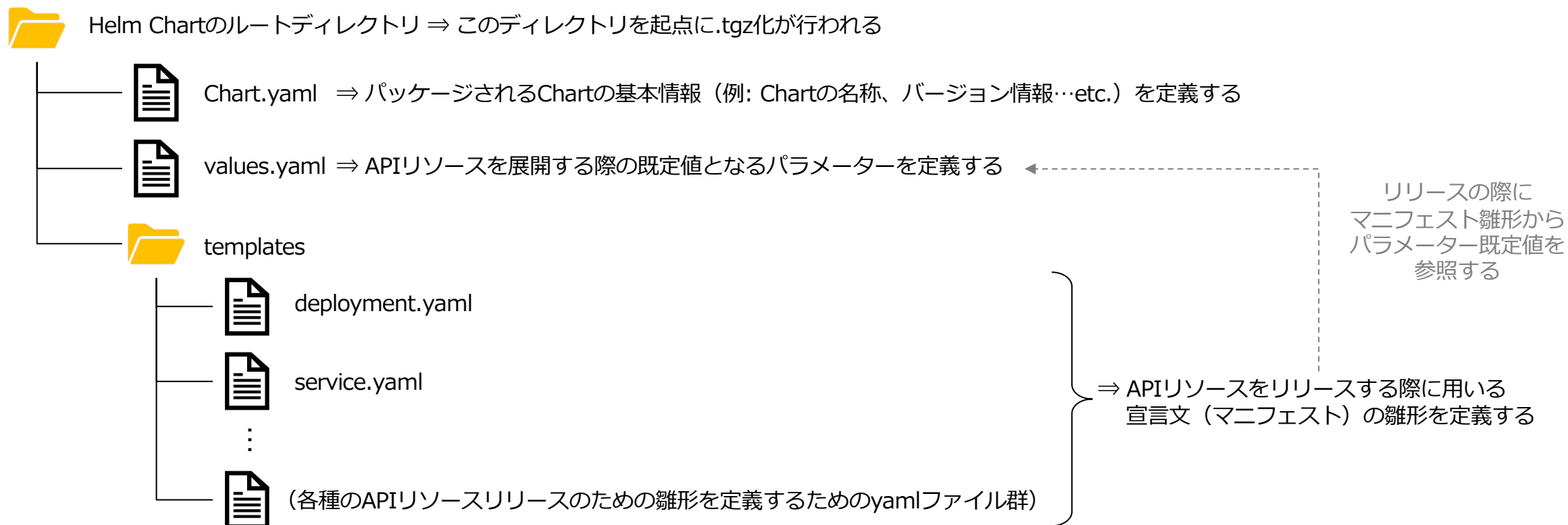
Helm Chartのパッケージ構造

- Helm Chartの実体は、Kubernetes環境でのAPIリソース定義のために用いる宣言文（「マニフェスト」と通称される）の集合を.tgz化したファイルである

– **Helm Chartのパッケージ・ファイル自体にはコンテナ・イメージが含まれていない**

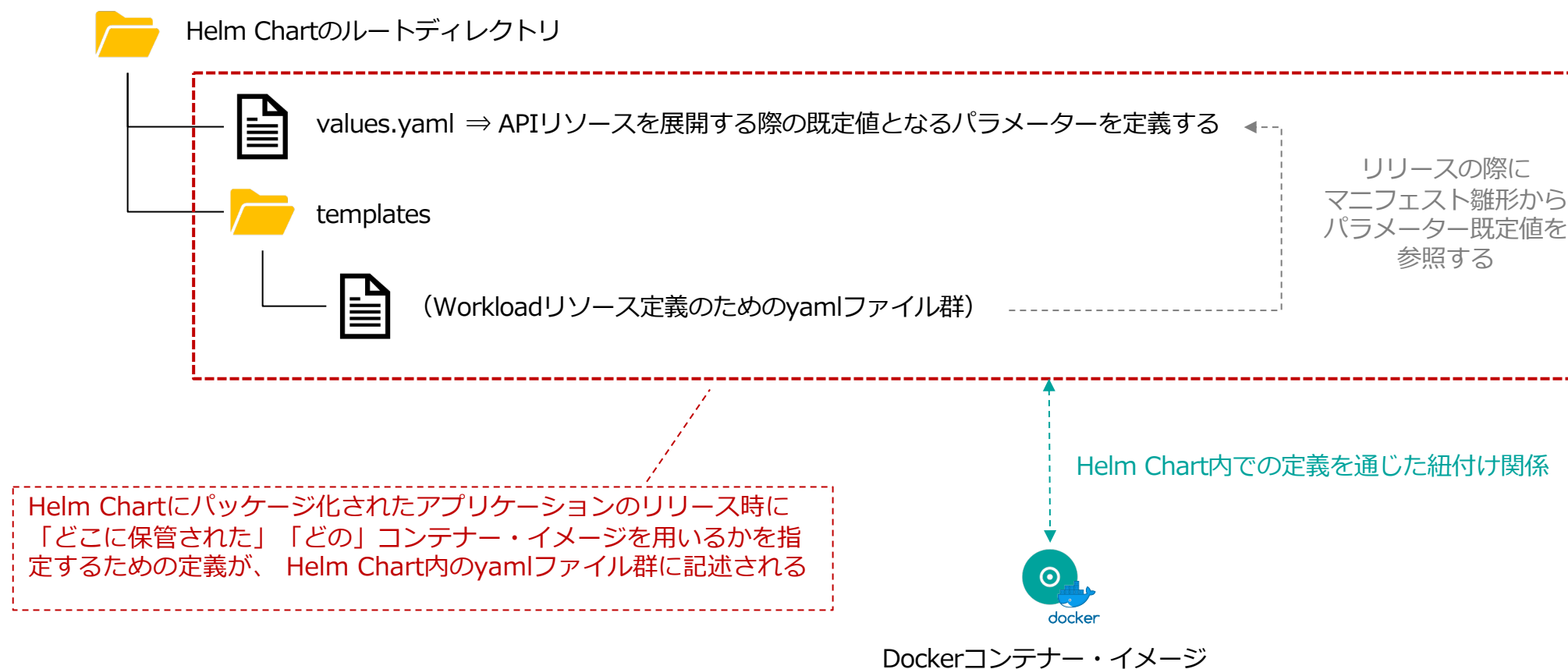
– Helm Chartパッケージ内での主要なディレクトリ、およびファイルの構成を下図に示す

• 参考情報: [Helm Documentation – Charts](#)



Helm ChartとDockerコンテナ・イメージとの紐付け

- Helm Chartとコンテナ・イメージとの紐付けは、Helm Chart内のyamlファイルで、パッケージされたアプリケーションが「どの」コンテナ・イメージを用いるかを定義することにより実現する



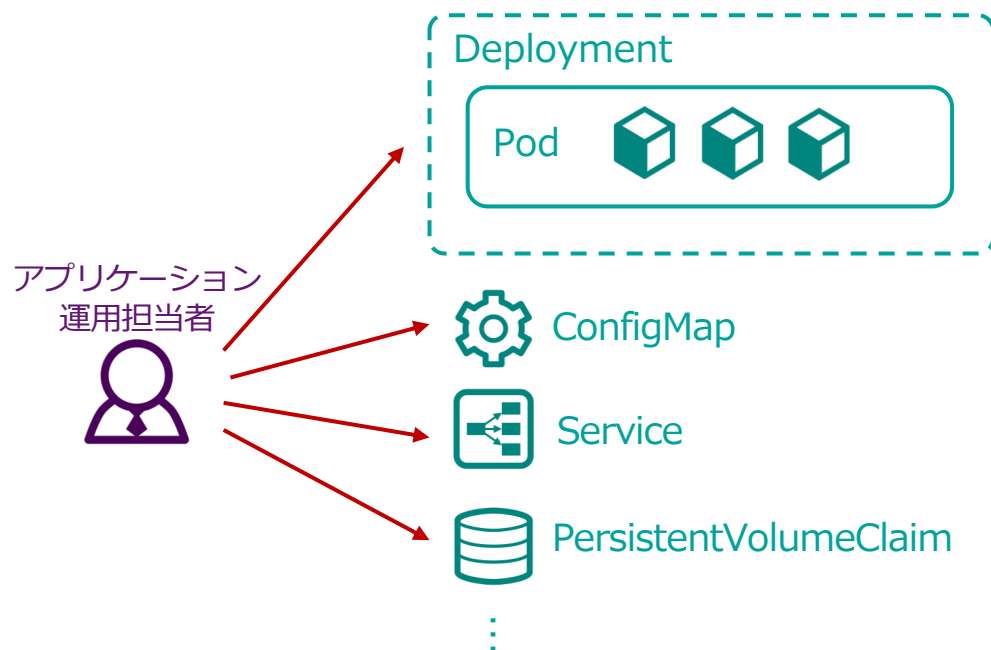
Helm Chartを用いたアプリケーション運用の利点

■ Helm Chartを用いることに伴う主な利点を以下に示す

- アプリケーション運用時における管理対象がHelm Chartにより整理/集約されることに伴い、運用作業の煩雑さを軽減する余地が生まれる
- Kubernetes APIリソースの構成がHelm Chartのパッケージ内でコード化（マニフェスト化）されることに伴い、アプリケーションの構成管理に際しての統制を確保することがより容易になる

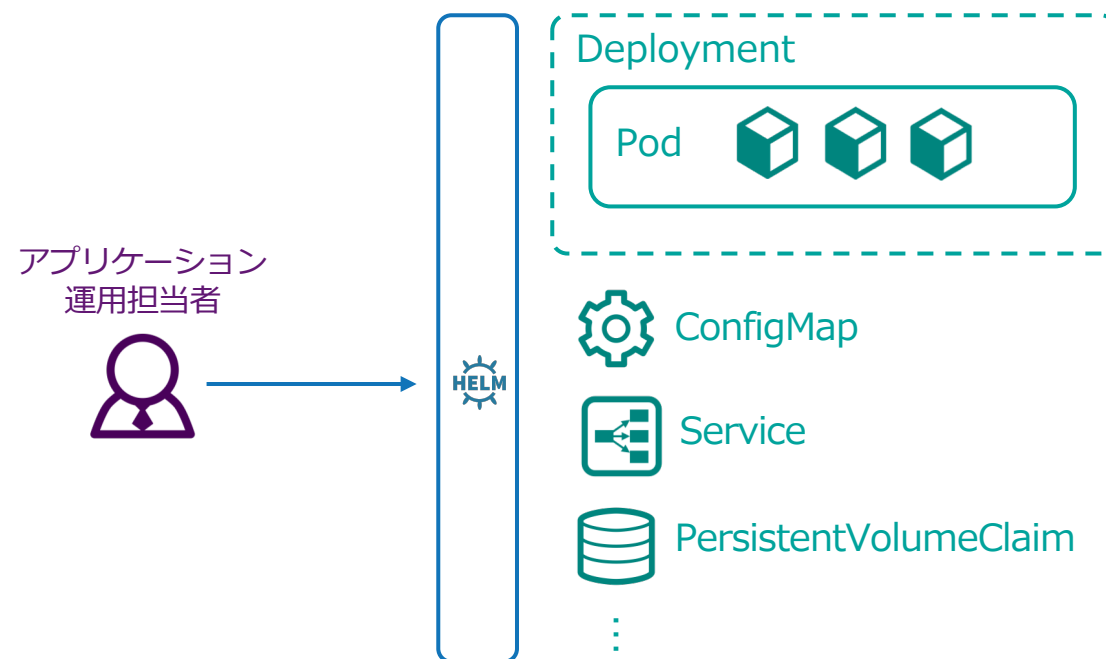
Helm Chartを用いない場合の運用イメージ

(運用自体は可能であるものの、個別のAPIリソースの管理が煩雑)



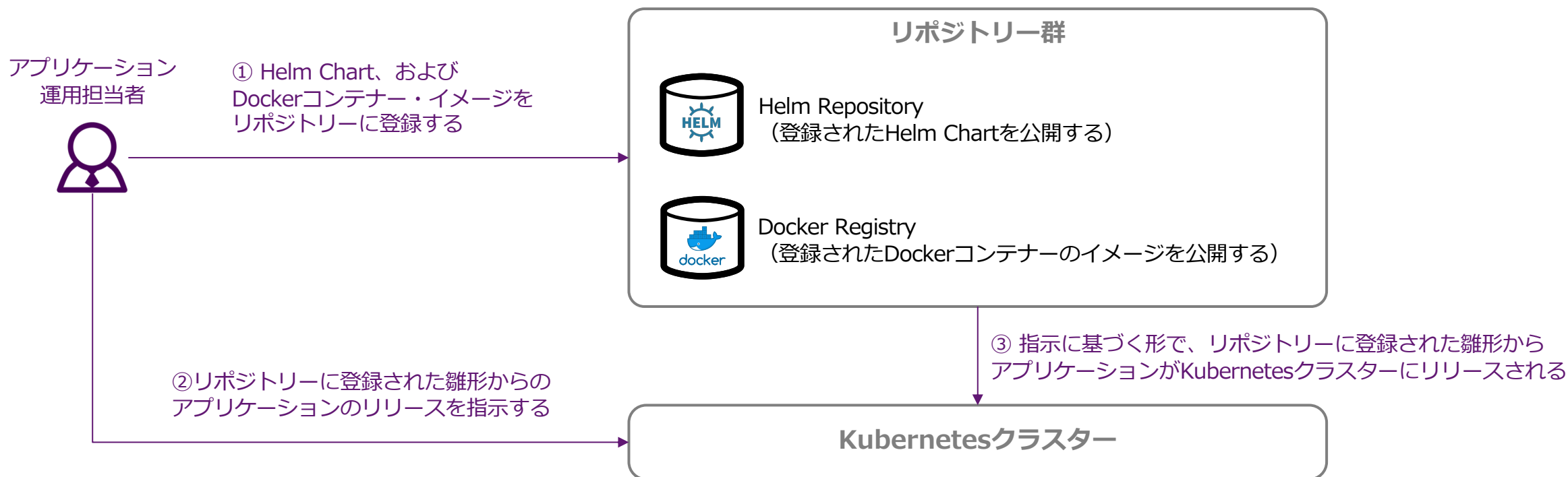
Helm Chartを用いた場合の運用イメージ

(Helm Chartにより管理対象が整理/集約される)



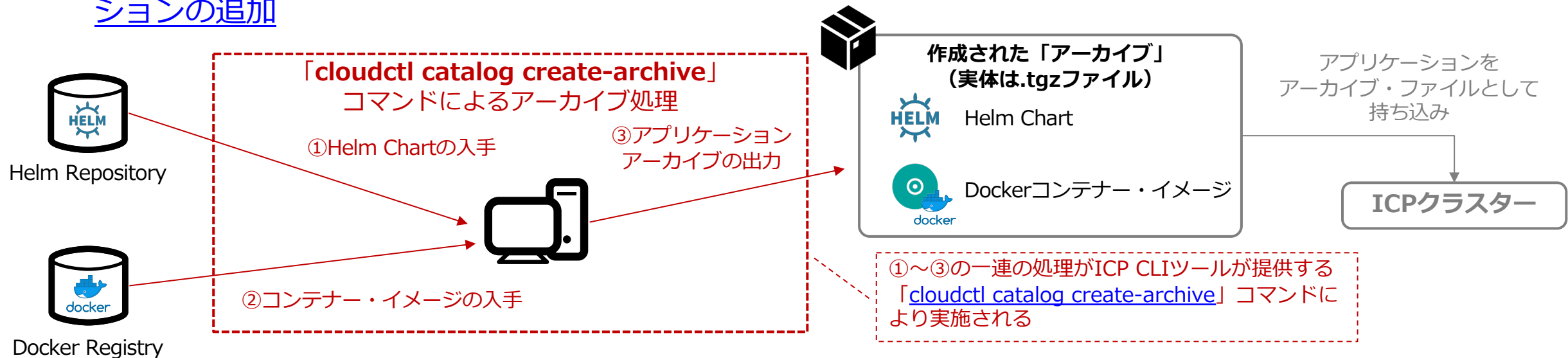
リポジトリを中心としたアプリケーション運用

- Kubernetes環境でのアプリケーション運用はリポジトリ（アプリケーション雛形を登録・公開するためのWebサービス）を中心とする形で行うのが基本となる
 - Kubernetes環境へのアプリケーションのリリースを、リポジトリに登録された雛形（Dockerコンテナのイメージ、およびHelm Chart）を用いて行う（下図）
 - アプリケーションの構成変更の際しても、構成変更済みのアプリケーション雛形をリポジトリに新たに登録した上で、登録した雛形からの再リリースを行う運用が基本となる



アプリケーションの「アーカイブ」化

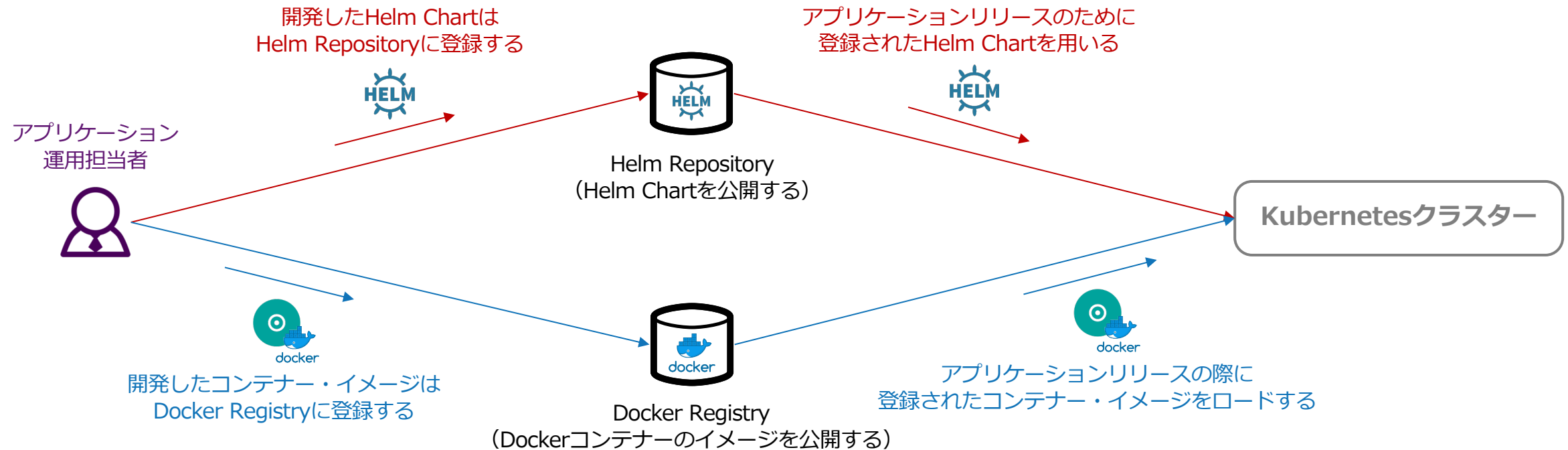
- Helm ChartとDockerコンテナ・イメージを同梱することにより、アプリケーションを「アーカイブ」化するための機能をICP製品が提供する
- ICP製品による「アーカイブ」化は、主として以下の目的のために用いられる
 - IBMが提供する有償アプリケーション（例: パスポート・アドバンテージ契約に基づき提供されるアプリケーション）の配布を目的として
 - ・参考情報: [IBM Knowledge Center - バンドルされた製品のインストール](#)
 - インターネット非接続のICPクラスタへのアプリケーションの持ち込みを目的として
 - ・参考情報: [IBM Knowledge Center - インターネット接続がないクラスタへのフィーチャー・アプリケーションの追加](#)



第5章 第2節: アプリケーションのリポジトリ登録

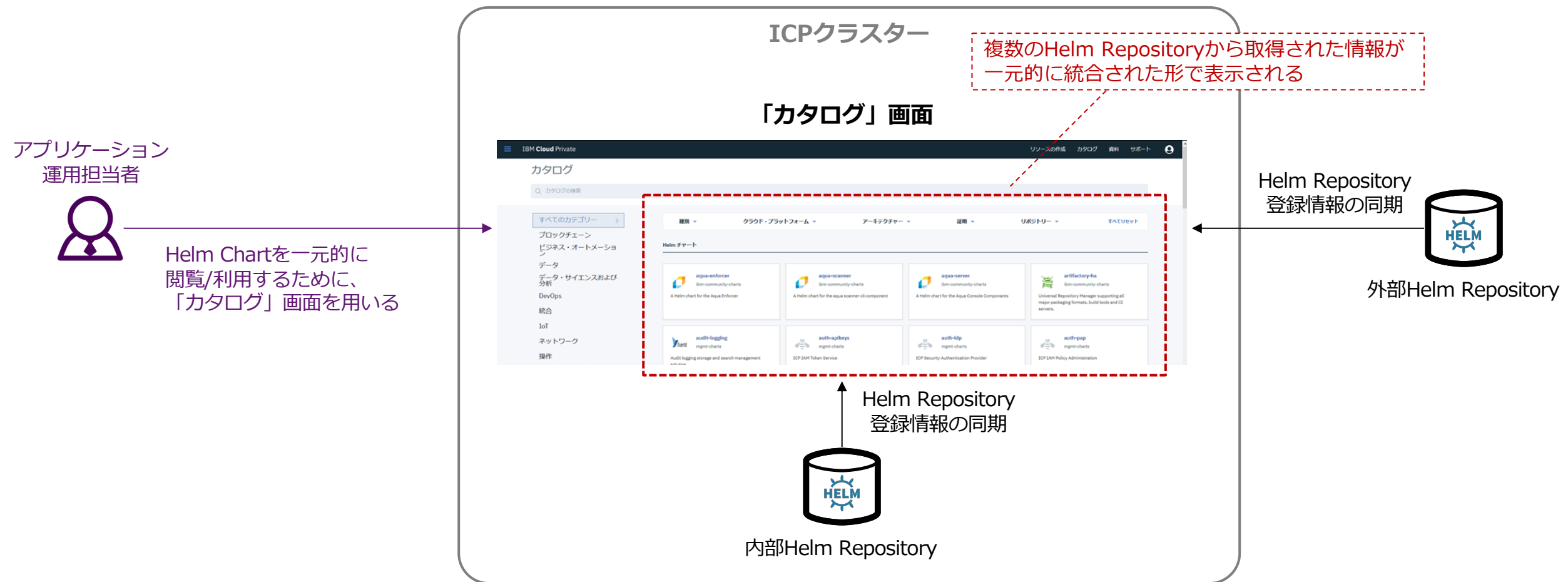
ICP Kubernetes環境における2種類のリポジトリの使い分け

- アプリケーション雛形の保管、公開のために用いるリポジトリ（アプリケーション雛形を保管・公開するためのWebサービス）として、互いに用途が異なる、2種類のリポジトリ（Helm Repository、およびDocker Registry）を使い分けること（場合によっては併用すること）が必要とされる



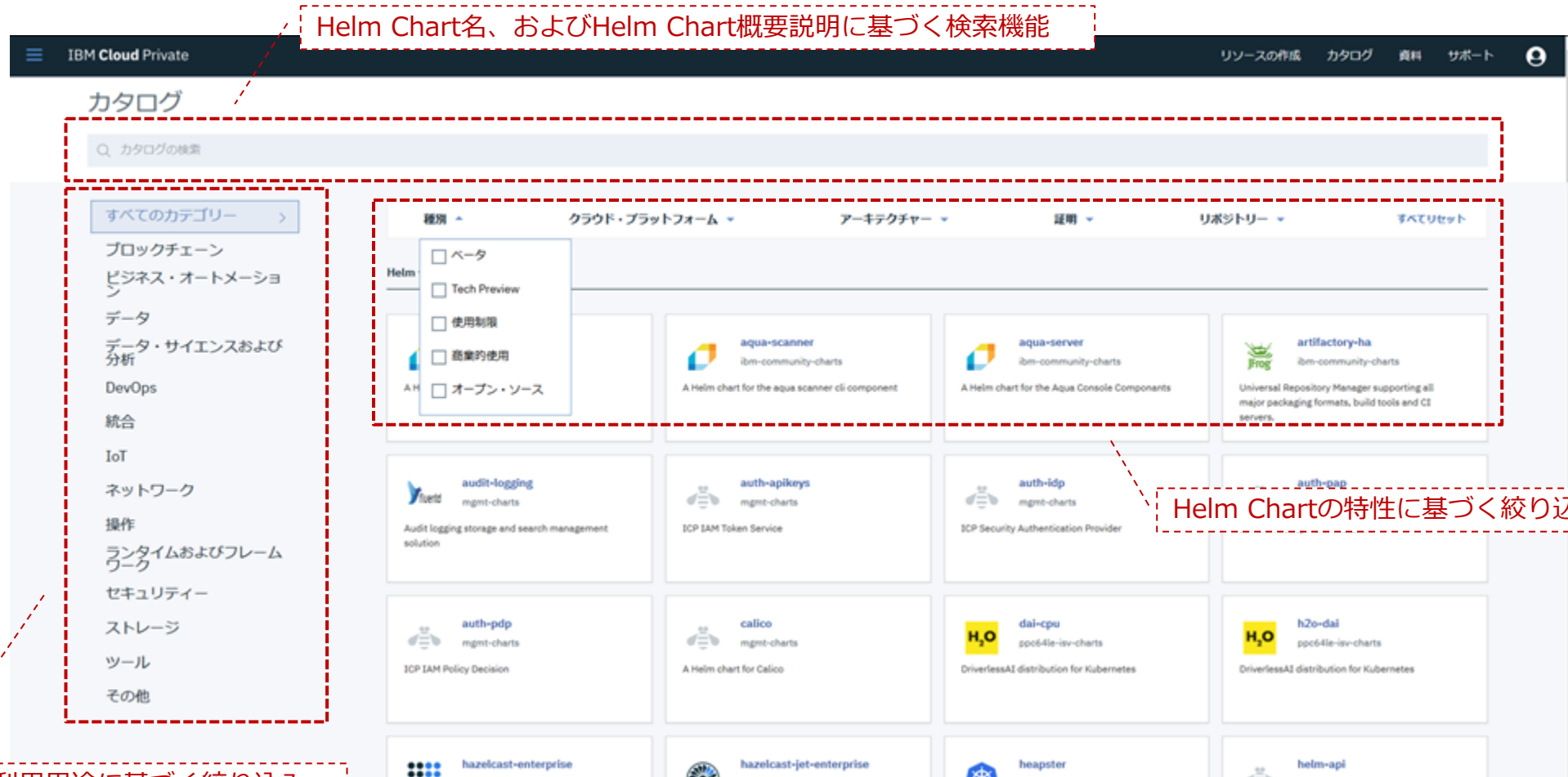
「カタログ」画面によるHelm Repository利用の一元化（概要）

- ICP製品の「カタログ」画面により、複数のHelm Repositoryにそれぞれ登録されたHelm Chartを、一元的に表示、利用することが可能となる



「カタログ」画面によるHelm Repository利用の一元化（画面構成）

- ICP製品の「カタログ」画面により、複数のHelm Repositoryにそれぞれ登録されたHelm Chartを、一元的に表示、利用することが可能となる



「カタログ」画面によるHelm Repository利用の一元化（機能設定）

- ICP製品の「カタログ」画面とHelm Repositoryとを連携させるためには、ICPクラスターへのHelm Repositoryの登録、および登録したHelm RepositoryからのHelm Chart情報の同期が必要である

– なお、いくつかのHelm RepositoryはICP製品の導入に伴い既定で登録される

- 参考情報: [IBM Knowledge Center - Helm リポジトリの管理](#)

リポジトリ

登録済みのHelm Repositoryの一覧表示

名前	URL
ibm-charts	https://raw.githubusercontent.com/IBM/charts/master/repo/stable/
ibm-community-charts	https://raw.githubusercontent.com/IBM/charts/master/repo/community/
local-charts	https://mycluster.icp:8443/helm-repo/charts
mgmt-charts	https://mycluster.icp:8443/mgmt-repo/charts
ppc64le-ibm-charts	https://raw.githubusercontent.com/ppc64le/charts/master/repo/stable/
ibm-charts-public	https://registry.bluemix.net/helm/ibm/

ページあたりの項目数 20 | 1-6 項目

リポジトリの同期 リポジトリの追加

登録処理、同期処理のためのインターフェース

リポジトリへのHelm Chart / コンテナ・イメージの登録（概要）

■ リポジトリへのHelm Chart / コンテナ・イメージの登録方法を下表に示す

- 内部リポジトリはICP製品が提供する認証・認可機能に基づくアクセス権管理を提供する
- 外部リポジトリはICP Kubernetes環境内外でリポジトリを共用する必要が生じた際に用いる

登録先リポジトリ	登録方法	参考情報
外部Helm Repository 	Helm Repositoryの管理者が示す登録手順に従う（多くの場合では、Helm Chartファイルのアップロード、およびHelm Repository Indexの再構築が求められることが想定される）	- 参考情報: The Chart Repository Guide - 参考情報: IBM Knowledge Center - 外部リポジトリへのチャートの追加
外部Docker Registry 	Docker Registryの管理者が示す登録手順に従う（多くの場合ではdocker pushコマンドによるコンテナ・イメージのアップロードが求められることが想定される）	参考情報: docker push Docker Documentation
内部Helm Repository 	- 以下に示す2通りの方法のいずれかに従う（2つ目の方法では、内部リポジトリに対する、アーカイブからのHelm Chart、およびDockerコンテナ・イメージの登録が一括で行われる） 「cloudctl catalog load-chart」コマンドを用いる 「cloudctl catalog load-archive」コマンドを用いる	参考情報: IBM Knowledge Center - 内部リポジトリへのチャートの追加
内部Docker Registry（「プライベート・イメージ・レジストリー」） 	- 以下に示す3通りの方法のいずれかに従う（3つ目の方法では、内部リポジトリに対する、アーカイブからのHelm Chart、およびDockerコンテナ・イメージの登録が一括で行われる） 「docker push」コマンドを用いる 「cloudctl catalog load-images」コマンドを用いる 「cloudctl catalog load-archive」コマンドを用いる	- 参考情報: IBM Knowledge Center - Docker CLI の認証の構成 - 参考情報: IBM Knowledge Center - イメージのプッシュおよびプル - 参考情報: docker push Docker Documentation

内部Helm RepositoryへのHelm Chartの登録手順/登録結果

- 参考情報: [IBM Knowledge Center - 内部リポジトリへのチャートの追加](#)

```
# (ICPクラスターにログイン済みの状態で) 内部Helm RepositoryへのHelm Chartの登録
> cloudctl catalog load-chart --archive testchart-0.1.0.tgz
Loading helm chart
Loaded helm chart

Synch charts
Synch started
OK
```

Helm Chartの登録手順



登録したHelm Chartが「カタログ」画面に反映される

Helm Chartの登録結果
（「カタログ」画面）

プライベート・イメージ・レジストリーへのイメージ登録手順

- 参考情報: [IBM Knowledge Center - イメージのプッシュおよびプル](#)

#プライベート・イメージ・レジストリーにログイン

```
> docker login mycluster.icp:8500
```

Username: admin

Password:

WARNING! Your password will be stored unencrypted in /root/.docker/config.json.

Configure a credential helper to remove this warning. See

<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

#端末上のコンテナ・イメージへのタグ付け

```
> docker tag nginx:latest mycluster.icp:8500/isolatednamespace/nginx:1.17.0
```

ICPクラスタのCAドメイン名

プライベート・イメージ・レジストリーのネットワークポート番号

コンテナ・イメージ名、およびタグ名

コンテナ・イメージを所属させるICP上のNamespace

#コンテナ・イメージのプライベート・イメージ・レジストリーへのプッシュ

```
> docker push mycluster.icp:8500/isolatednamespace/nginx:1.17.0
```

The push refers to repository [mycluster.icp:8500/isolatednamespace/nginx]

d7acf794921f: Pushed

d9569ca04881: Pushed

cf5b3c6798f7: Pushed

1.17.0: digest: sha256:079aa93463d2566b7a81cbdf856afc6d4d2a6f9100ca3bcbecf24ade92c9a7fe size: 948

プライベート・イメージ・レジストリーへのイメージ登録結果

- 参考情報: [IBM Knowledge Center - イメージのプッシュおよびプル](#)
- 参考情報: [IBM Knowledge Center - イメージ・マネージャー](#)

IBM Cloud Private

リソースの作成 カタログ 資料 サポート

コンテナ・イメージ

nginx

名前	セキュリティ・スキャン	所有者	有効範囲
ibmcom/nginx-ingress-controller-amd64	失敗しました	ibmcom	namespace
isolatednamespace/nginx	-	isolatednamespace	namespace

ページあたりの項目数: 20 | 1 から 2/2 項目

1/1 ページ

登録されたコンテナ・イメージ

コンテナ・イメージが属するNamespaceが「所有者」として管理画面上に表示される

第5章 第3節: アプリケーションのリリース

ICP Kubernetes環境でのアプリケーションのリリース手順

■ ICP Kubernetes環境でのアプリケーションのリリース手順はHelm Chartの利用有無、および、リリース操作のために用いるUIにより分類される

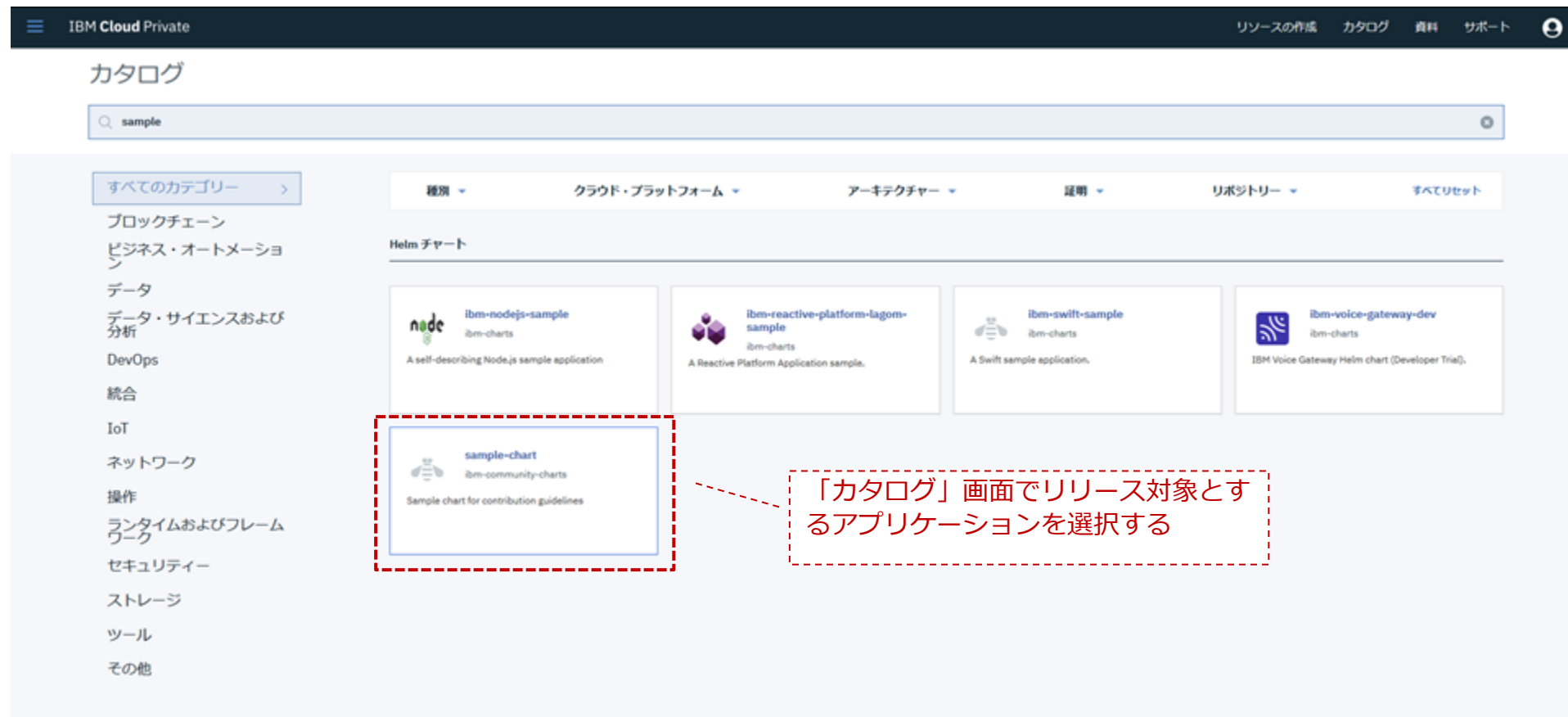
– 運用するアプリケーションの規模や運用者のスキルレベル、もしくは、運用アセットの横展開やCI/CDツール連携の要否を踏まえた上で、運用方針に即した手順を使い分けることが望ましい

Helm Chart利用有無	リリース操作向けUI	リリース手順の概要 / 参考情報
Helm Chart使用 (大規模アプリケーションの本番運用に適する)	ICP GUI (ICP独自のUI) (「カタログ」画面)	- ICP製品のWeb GUIが提供する「Catalog」画面を用いることにより、GUIのウィザード形式でHelm Chartからアプリケーションをリリースする - 参考情報: IBM Knowledge Center - Catalog 内の Helm チャートのデプロイ
	Helm CLI (helm install)	- Helm CLIのコマンドを用いてHelm Chartからアプリケーションをリリースする - CI/CDソリューション (例: Microclimate) との連携はCLIベースで行うことが一般的 - 参考情報: IBM Knowledge Center - Helm CLI (helm) のインストール - 参考情報: Install a chart archive
Helm Chart不使用 (小規模アプリケーションの運用や検証に適する)	ICP GUI (ICP独自のUI) (「Workload」メニュー、他)	- ICP GUIを用いることにより、アプリケーションを構成するKubernetes APIリソース (例: Deploymentリソース) を個別に作成する - 参考情報: IBM Knowledge Center - デプロイメントの作成
	Kubernetes CLI (kubectl)	- Kubernetes CLI (kubectlコマンド) を用いることにより、アプリケーションを構成するKubernetes APIリソース (例: Deploymentリソース) を個別に作成する - CI/CDソリューション (例: Microclimate) との連携はCLIベースで行うことが一般的 - 参考情報: IBM Knowledge Center - Kubernetes CLI (kubectl) のインストール - 参考情報: Overview of kubectl - Kubernetes

GUIからのリリース手順の画面遷移例（Helm Chart使用）（1/3）

■ ICP GUIを用いたアプリケーション・リリース（Helm Chart使用）の際における画面遷移例を以下に示す

- 参考情報: [IBM Knowledge Center - Catalog 内の Helm チャートのデプロイ](#)



GUIからのリリース手順の画面遷移例（Helm Chart使用）（2/3）

■ ICP GUIを用いたアプリケーション・リリース（Helm Chart使用）の際における画面遷移例を以下に示す

- ・参考情報: [IBM Knowledge Center - Catalog 内の Helm チャートのデプロイ](#)

IBM Cloud Private

← すべて表示

sample-chart V 1.1.2

概要 構成

構成

Sample chart for contribution guidelines. 構成のこれらのパラメーターを編集します。

Helm リリース名 *

sample-chart-release

ターゲット名前空間 *

default

使用許諾条件 *

☒ 使用条件を読んで、ご使用条件

ポッド・セキュリティ

このチャートではポッド・セキュリティ・ポリシーが指定されていません。 `ibm-anyuid-hostpath-psp` を持つ名前空間を選択するか、[チャート・セキュリティ参照テーブル](#) を参照して既知のポッド・セキュリティ・ポリシーが定義されているチャートのリストを確認してください。

ターゲット名前空間ポリシー

ibm-privileged-psp, db2oltp-dev-psp, ibm-anyuid-psp, ibm-restricted-psp, ibm-anyuid-hostaccess-psp, ibm-anyuid-hostpath-psp

リリースするアプリケーションのHelm Chartが求める構成値を入力した後に、画面最下部の「インストール」ボタンをクリックする

GUIからのリリース手順の画面遷移例（Helm Chart使用）（3/3）

■ ICP GUIを用いたアプリケーション・リリース（Helm Chart使用）の際における画面遷移例を以下に示す

- ・参考情報: [IBM Knowledge Center - Catalog 内の Helm チャートのデプロイ](#)



Helm Chartを用いたアプリケーションのリリース結果の確認

- Helm Chartを用いてリリースされたアプリケーションの構成情報、および稼働状況を、ICP製品の「Helmリリース」画面を経由する形で確認出来る

・参考情報: [IBM Knowledge Center - Helm リリースの管理](#)

sample-chart-release ● デプロイ済み
更新: 2019年6月23日 午後 5:55

← すべて表示

起動

詳細およびアップグレード

チャート名	実行バージョン	使用可能バージョン	
sample-chart 名前空間 default	1.1.2 インストール済み: 2019年6月23日 → リリース・ノート	1.1.2 リリース済み: 2018年12月6日 → リリース・ノート	アップグレード ロールバック

ConfigMap

名前	データ	経過時間
sample-chart-release-sample-chart-config	1	1m

デプロイメント

名前	必要数	実行	最新	使用可能	経過時間
sample-chart-release-sample-chart	1	1	1	1	1m

Helm Chartよりリリースされたアプリケーションの構成情報、および稼働状況を、ICP製品の「Helmリリース」画面配下の各アプリケーション向けの管理画面より確認出来る

GUIからのリリース手順（Helm Chart不使用）（1/2）

- ICP GUIを用いたアプリケーション・リリース時での画面遷移例を以下に示す
 - Helm Chartを用いない場合においては、アプリケーションを構成する様々なAPIリソース（例えばDeploymentリソース）を個別に作成しなければならない
 - 参考情報: [IBM Knowledge Center - デプロイメントの作成](#)

「デプロイメント」画面で「デプロイメントの作成」ボタンをクリックする

デプロイメントの作成

名前	名前空間	必要数	現行	準備完了	使用可能	作成日	
sample-chart-release-sample-chart	default	1	1	1	1	24 分前	
logging-elk-client	kube-system	1	1	1	1	5 時間前	起動
logging-elk-kibana	kube-system	1	1	1	1	5 時間前	起動
logging-elk-logstash	kube-system	1	1	1	1	5 時間前	起動
logging-elk-master	kube-system	1	1	1	1	5 時間前	起動
default-backend-isolatedproxy	kube-system	1	1	1	1	2 日前	起動
vulnerability-advisor-compliance-annotator	kube-system	1	1	1	1	3 日前	
vulnerability-advisor-compliance-indexer	kube-system	1	1	1	1	3 日前	
vulnerability-advisor-config-parser	kube-system	1	1	1	1	3 日前	
vulnerability-advisor-scanner	kube-system	1	1	1	1	3 日前	

GUIからのリリース手順（Helm Chart不使用）（2/2）

■ ICP GUIを用いたアプリケーション・リリース時での画面遷移例を以下に示す

–Helm Chartを用いないならば、アプリケーションを構成する様々なAPIリソース（例：Deploymentリソース）を個別に作成しなければならない

- 参考情報: [IBM Knowledge Center - デプロイメントの作成](#)

「デプロイメントの作成」画面で、Deploymentリソースを作成するために必要な情報を入力し、「作成」ボタンをクリックする

IBM Cloud Private

リソースの作成 カタログ 資料 サポート

すべての名前空間

デプロイメントの作成

作成日

24 分前	
5 時間前	起動
5 時間前	起動
5 時間前	起動
5 時間前	起動
2 日前	起動
3 日前	
3 日前	
3 日前	
3 日前	
3 日前	
3 日前	
3 日前	

キャンセル 作成

考慮事項: 管理者によるアプリケーション・リリースに関わる各種制限

- ICP Kubernetes環境の管理者は、環境の管理・統制を目的として、アプリケーション・リリース運用に関わる各種の制限を設定することがある
 - アプリケーション管理者は、それらの制限を意識した運用を行わなければならない

クラスター管理者



関心事: ICPクラスター全体でのリソース消費水準の管理・統制

⇒ 実施する制限: 「[Resource Quota](#)」機能を用いて利用者によるリソース消費を制限する ⇒

関心事: アプリケーションの展開元となるイメージの統制確保

⇒ 実施する制限: ICPクラスターから利用可能なリポジトリを制限する（後述） ⇒

関心事: コンテナが起点となる形でのセキュリティ侵害の防止

⇒ 実施する制限: コンテナに認める特権を制限する（後述） ⇒

アプリケーション管理者



制限を踏まえたあるべき対応:

自らに割り当てられたリソース消費制限の範囲内で、リリースするアプリケーションへのリソース配分を行う

制限を踏まえたあるべき対応:

利用が認められたリポジトリのみをアプリケーション・リリース運用のために用いる

制限を踏まえたあるべき対応:

制限された特権の枠内で稼働するよう、リリースするアプリケーションを設計・実装する

考慮事項: リポジトリ利用を制限するための各種機能 (1/2)

- アプリケーションの展開元となるリポジトリ、もしくは、リポジトリ上に保管されたアプリケーション雛形へのアクセスを制限するための「**複数の**」機能がICP製品により実装されているため、リリース運用時にはそれらの機能に起因する制限に抵触しないよう留意しなければならない（下表）

リポジトリ種別	アクセス制限の内容	参考情報
Helm Repository 	「カタログ」画面の利用者が所属する「Team」に応じて、「カタログ」画面上でアクセス可能なHelm Repositoryが制限される（同制限の具体的な内容が、本ガイドの7.2節で解説されている）	IBM Knowledge Center - チームへの Helm リポジトリと名前空間の追加
Docker Registry 	- あるNamespaceにおいて、アプリケーション・リリースのために用いることが出来るDocker Registryが制限される（「Image Policy」による制限）（次ページで具体的な制限内容を補足する） - Docker Registry上でコンテナ・イメージが所属するNamespaceに応じて、同イメージの展開先とすることが出来るICPクラスター上のNamespaceが制限される（同制限の具体的な内容が、本ガイドの7.1節で解説されている）	IBM Knowledge Center - コンテナ・イメージ・セキュリティの適用 IBM Knowledge Center - イメージ・スコープの変更 IBM Knowledge Center - 特定の名前空間の imagePullSecrets の作成

考慮事項: リポジトリ利用を制限するための各種機能 (2/2)

- (前ページの補足) 「Image Policy」により、アプリケーション・リリースのために利用可能なDocker Registryが制限される (制限に抵触するDocker Registryから取得したコンテナ・イメージを含むPodは**展開されない**)

・参考情報: [IBM Knowledge Center - コンテナ・イメージ・セキュリティの適用](#)

Image Policy 「ibmcloud-default-cluster-image-policy」が「既定で」ICPクラスター全体を対象として適用されている

The screenshot displays the IBM Cloud Private console interface. On the left, a sidebar menu shows 'リソース・セキュリティ' (Resources & Security) with sub-items 'イメージ・ポリシー' (Image Policy), 'ポッド・セキュリティ' (Pod Security), and '割り当て量' (Quota). The main content area is titled 'イメージ・ポリシー / ibmcloud-default-cluster-image-policy'. Below this, there's a section 'イメージ・ポリシーの詳細' (Image Policy Details) with a table showing the policy's name and scope. To the right, there's a section 'イメージ・ポリシーのレジストリー' (Image Policy Registries) with a table listing various Docker registries and their scan status.

タイプ	詳細
名前	ibmcloud-default-cluster-image-policy
有効範囲	クラスター

レジストリー URL	VA スキャン
mycluster.icp:8500/*	施行せず
registry.bluemix.net/ibm/*	施行せず
cp.icr.io/cp/*	施行せず
docker.io/apache/couchdb*	施行せず
docker.io/ppc64le/*	施行せず

アプリケーション・リリースのために利用可能なDocker Registry、およびVA (Vulnerability Advisor) による脆弱性スキャンの要否がポリシーとして定義される

Image Policy定義の際には、ポリシーの適用範囲 (Scope) を以下の2通りから選択する

- ・ ICPクラスター全体を対象に適用する (「クラスター」)
- ・ 特定のNamespaceを対象に適用する

※: 特定のNamespaceを対象に適用されたポリシーは、ICPクラスター全体を対象に適用されたポリシーより優先する

考慮事項: 「Pod Security Policy」によるコンテナ特権の制限

- アプリケーションが利用するコンテナが「Pod security Policy」によるコンテナ特権の制限に抵触する場合、そのコンテナを含むPodの起動が禁止される
- アプリケーションの実行に際して何らかの特権が必須であるならば、より穏やかなポリシーを適用することを通じて、特権を要するコンテナ（Pod）の起動を明示的に認めなければならない
- 参考情報: [IBM Knowledge Center - ポッドの分離](#)
- 参考情報: [IBM Knowledge Center - 非デフォルト名前空間での高い特権を必要とする Helm チャートのデプロイ](#)

ICP製品のv3.1.1以降のバージョンを用いてICPクラスターを新規構築した場合には、「**厳しい**」Pod Security Policy (ibm-restricted-ppsp) が「**既定で**」適用される（例外として、ICP製品の導入時に既定で作成される一部のNamespaceにはより穏やかなポリシーが適用される）。

ICPクラスターで事前定義された Pod Security Policy	Pod Security Policyの概要
ibm-restricted-ppsp (ICP v3.1.1以降での既定)	- Podの実行の際での非rootユーザーの利用が必須 - PodからICPノードへのアクセスが防止される
ibm-anyuid-ppsp	- Podの実行の際での任意の実行ユーザーの選択が可能 - PodからICPノードへのアクセスが防止される
ibm-anyuid-hostpath-ppsp	- Podの実行の際での任意の実行ユーザーの選択が可能 - PodからのhostPathボリュームの利用が可能
ibm-anyuid-hostaccess-ppsp	- Podの実行の際での任意の実行ユーザーの選択が可能 - PodからのICPノードへのフルアクセスが許可される
ibm-privileged-ppsp	- Podの実行の際での任意の実行ユーザーの選択が可能 - PodからのICPノード特権の利用が可能

より厳しい

 ポリシーにより実現する
セキュリティの水準

より緩い

第5章 第4節: アプリケーションの構成変更

Kubernetes APIリソースを通じたアプリケーション構成変更

- Kubernetes環境でのアプリケーション構成変更は、アプリケーションを構成するKubernetes APIリソースに対する構成変更を通じて実現される



「Helmリリース」画面を用いたアプリケーションの構成管理（1/3）

- アプリケーションのリリースをHelm Chartから行った場合には、Helm Chartのリリース履歴を基準とした構成管理/構成変更の操作が可能となる

・参考情報: [IBM Knowledge Center - Helm リリースの管理](#)

IBM Cloud Private

リソースの作成 カタログ 資料 サポート

Helm リリース

Helm Chartからリリースしたアプリケーションの一覧が表示される

名前 -	名前空間	状況	チャート名	実行バージョン	使用可能バージョン	更新	アクション
audit-logging	kube-system	● デプロイ済み	audit-logging	3.1.2	最新	2019年6月20日 午後 02:27	...
auth-apikeys	kube-system	● デプロイ済み	auth-apikeys	3.1.2	最新	2019年6月20日 午後 02:21	起動 ...
auth-idp	kube-system	● デプロイ済み	auth-idp	3.1.2	最新	2019年6月20日 午後 02:20	起動 ...
auth-pap	kube-system	● デプロイ済み	auth-pap	3.1.2	最新	2019年6月20日 午後 02:21	起動 ...
auth-pdp	kube-system	● デプロイ済み	auth-pdp	3.1.2	最新	2019年6月20日 午後 02:21	起動 ...
calico	kube-system	● デプロイ済み	calico	3.1.2	最新	2019年6月20日 午後 02:18	...
catalog-ui	kube-system	● デプロイ済み	icp-catalog-chart	3.1.2	最新	2019年6月20日 午後 02:27	...
cert-manager	cert-manager	● デプロイ済み	ibm-cert-manager	3.1.2	最新	2019年6月20日 午後 02:19	...
custom-metrics-adapter	kube-system	● デプロイ済み	ibm-custom-metrics-adapter	3.1.2	最新	2019年6月20日 午後 02:27	起動 ...
heapster	kube-system	● デプロイ済み	heapster	3.1.2	最新	2019年6月20日 午後 02:27	...
helm-api	kube-system	● デプロイ済み	helm-api	3.1.2	最新	2019年6月20日 午後 02:27	...
helm-repo	kube-system	● デプロイ済み	helm-repo	3.1.2	最新	2019年6月20日 午後 02:27	...
icp-management-ingress	kube-system	● デプロイ済み	icp-management-ingress	3.1.2	最新	2019年6月20日 午後 02:21	起動 ...

「Helmリリース」画面を用いたアプリケーションの構成管理（2/3）

- アプリケーションのリリースをHelm Chartから行った場合には、Helm Chartのリリース履歴を基準とした構成管理/構成変更の操作が可能となる

• 参考情報: [IBM Knowledge Center - Helm リリースの管理](#)

The screenshot shows the IBM Cloud Private console interface for managing Helm releases. The top navigation bar includes 'リソースの作成', 'カタログ', '資料', and 'サポート'. The main content area is titled 'mariadb' and shows it is 'デプロイ済み' (deployed) with an update timestamp of '2019年6月20日 午後 2:19'. A red dashed box highlights the 'Helm Chartのリリース履歴を基準として、リリース済みアプリケーションのアップグレード（もしくはロールバック）を行うことが可能' (Using the Helm Chart release history as a baseline, it is possible to upgrade (or rollback) the released application). Below this, a table shows the current version (3.1.2) and the available version (5.11.2), with buttons for 'アップグレード' (Upgrade) and 'ロールバック' (Rollback). The 'アップグレード' button is highlighted. Below the version information, there are sections for 'ConfigMap' and 'ポッド' (Pods). The 'ConfigMap' section shows a table with one entry: 'mariadb-start' with a value of '1' and a duration of '13d'. The 'ポッド' section shows a table with one entry: 'mariadb-0' with a status of 'Running' and a duration of '13d'. A 'ログの表示' (Show logs) link is present next to the pod entry.

IBM Cloud Private

リソースの作成 カタログ 資料 サポート

← すべて表示

mariadb ● デプロイ済み
更新: 2019年6月20日 午後 2:19

起動

Helm Chartのリリース履歴を基準として、リリース済みアプリケーションのアップグレード（もしくはロールバック）を行うことが可能

詳細およびアップグレード

チャート名	実行バージョン	使用可能バージョン	
mariadb	3.1.2	5.11.2 ▲	アップグレード ロールバック
名前空間 kube-system	インストール済み: 2019年6月20日 → ReadMe	リリース済み: 2019年6月18日 → ReadMe	

ConfigMap

名前	データ	経過時間
mariadb-start	1	13d

ポッド

名前	準備完了	状況	再始動	経過時間	
mariadb-0	1/1	Running	0	13d	ログの表示

秘密

「Helmリリース」画面を用いたアプリケーションの構成管理（3/3）

■ アプリケーションのリリースをHelm Chartから行った場合には、Helm Chartのリリース履歴を基準とした構成管理/構成変更の操作が可能となる

- 参考情報: [IBM Knowledge Center - Helm リリースの管理](#)

The screenshot shows the 'アップグレード' (Upgrade) screen for a Helm chart named 'mariadb' in the 'kube-system' namespace. The current version is 3.1.2. The interface includes sections for 'チャートのリポジトリを確認' (Check chart repository), 'パラメーター' (Parameters), and 'ログの表示' (Show logs). Red dashed boxes highlight specific features:

- アップグレード先とするHelm Chartのバージョン、および、Helm Chartの入手元とするHelm Repositoryを選択することが可能** (It is possible to select the Helm Chart version to upgrade to and the Helm Repository to obtain the Helm Chart from).
- リリース済みアプリケーションからのパラメーター引継ぎ有無の指定が可能（ただし、Helm リリースのアップグレード時にはパラメーターが検証されないため、必ず構成パラメーターを確認しなければならない）** (It is possible to specify whether to inherit parameters from the released application (however, parameters are not verified during Helm release upgrade, so you must always confirm the configuration parameters)).
- リリース済みのアプリケーションに対する、アップグレードを機会としたパラメーターの再指定が可能** (It is possible to re-specify parameters for the released application using the upgrade opportunity).

アプリケーション・コンテナの構成変更に伴うアップデート戦略

- Workloadリソースの設定としてアップデート戦略（Strategy）を定義することにより、構成変更時での新旧構成の並行稼働の有無を選択することが出来る

- ・参考情報: [Deployments - Kubernetes \(Strategy\)](#)
- ・参考情報: [Perform a Rolling Update on a DaemonSet - Kubernetes \(DaemonSet Update Strategy\)](#)
- ・参考情報: [StatefulSets - Kubernetes \(Update Strategies\)](#)

アップデート戦略	Rolling Update	Recreate	OnDelete
アップデート戦略の概要	構成変更の最中に 新旧構成のコンテナを並行稼働させる ことを通じて、コンテナを用いて 稼働するアプリケーションの可用性を確保する	旧構成のコンテナを廃棄した後に新構成のコンテナを再展開 する。そのため、コンテナを用いて 稼働するアプリケーションの中断が生じる	メンテナンスなどにより、 旧構成のコンテナが停止されたタイミングで旧構成のコンテナの廃棄、および新構成のコンテナの展開を行う
アップデート戦略に対応したWorkloadリソースの種別	• Deployment • DaemonSet • StatefulSet	• Deployment	• DaemonSet • StatefulSet
アップデート戦略の利点	• 構成変更時においてアプリケーションの可用性が確保される	• 新旧構成のコンテナを並行稼働させるためのリソースが不要	• 新旧構成のコンテナを並行稼働させるためのリソースが不要 • アプリケーション構成変更に伴うアプリケーション停止のタイミングを、他の運用イベントと統合することが出来る

結果として、アプリケーション構成変更時における可用性確保の有無に対応する設定がHelm Chartに含まれることに留意のこと
 （つまり、Helmの「Upgrade」を行う際でのアプリケーション停止の有無は、Helm Chartの内容次第となる）

Deploymentリソースでのコンテナ再生成のための判定基準

- Deploymentリソースの構成変更を行った際でのコンテナ（正確には ReplicaSet）の再生成の有無は、Deployment定義上の「template」セクションに変更が生じたかを基準に判断される（変更が生じたならば再生成）

・参考情報: [Deployments - Kubernetes \(Updating a Deployment\)](#)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

「spec.template」セクション以外の変更はコンテナの再生成（rollout）を引き起こさない。

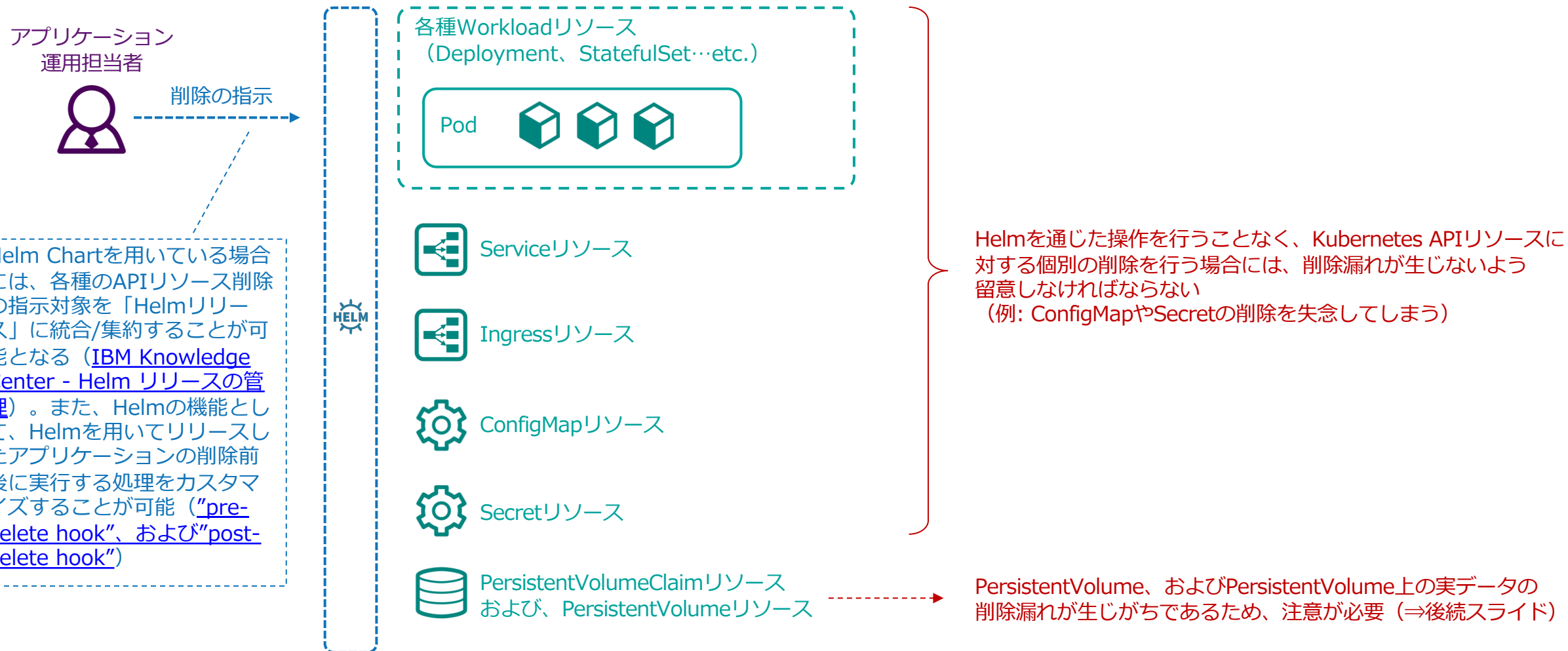
コンテナが再生成（rollout）されるためには、「spec.template」セクションへの変更が必要（例えば、「template.metadata.labels」の変更がそれに当たる）

ここで、コンテナ内のソフトウェア（例:ミドルウェア）の実装によっては、アプリケーションに対する構成変更（例:ConfigMap内の外部DBネットワークアドレスの変更）を反映するために、コンテナの再起動が必要となり得る点に留意が必要（コンテナが再生成されない限り、変更がいつまでも反映されない）

第5章 第5節: アプリケーションの廃棄

リリースされたアプリケーションの削除

- リリースされたアプリケーションの削除は、そのアプリケーションを構成する Kubernetes APIリソースへの削除指示を通じて実現される



PersistentVolume上の永続データの廃棄

- アプリケーションの廃棄に応じてPersistentVolumeClaimを削除した際での永続データ実体の削除有無は、PersistentVolumeに設定する「Reclaim Policy」により制御される（下表）

・ 参考情報: [Persistent Volumes - Kubernetes \(Reclaiming\)](#)

Reclaim Policy種別	Retain	Delete	Recycle
PersistentVolumeClaimを削除した際における保管データの削除有無	保管データは 削除されない	保管データが 削除される (PersistentVolumeの実体にあたるストレージ・ボリュームごと削除される)	保管データが 削除される (「rm」コマンドによる削除)
PersistentVolumeClaimを削除後におけるPersistentVolume状態	削除されずに保持される (PersistentVolumeの 再利用不可)	削除される (※: volume pluginが「Delete」ポリシーをサポートする前提において)	削除されずに保持される (PersistentVolumeの再利用可能)
アプリケーション廃棄運用の観点での考慮点	アプリケーションが利用したストレージ関連リソースを完全に削除するためには、PersistentVolume上の保管データの削除、および、PersistentVolume自体の削除を明示的に行わなければならない	アプリケーションが利用したストレージ関連リソースが完全に削除されることに伴い、削除漏れを避けることが出来る	将来的に廃止される予定のポリシー種別であるため、使用は非推奨

結果として、PersistentVolumeClaim削除時における永続データの削除有無に対応する設定がHelm Chart内に含まれてしまう事態が生じ得る
(その場合、**PersistentVolume**、および**PersistentVolume**に保管された永続データに対する**明示的な削除の要否は、Helm Chartの内容次第**となる)

リポジトリからのHelm Chart / コンテナ・イメージの削除

■ リポジトリからのHelm Chart / コンテナ・イメージの削除方法を下表に示す

削除元リポジトリ	削除方法	備考
外部Helm Repository 	Helm Repositoryの管理者が示す削除手順に従う	
外部Docker Registry 	Docker Registryの管理者が示す削除手順に従う	
内部Helm Repository 	「cloudctl catalog delete-chart」コマンドを用いる	
内部Docker Registry (「プライベート・イメージ・レジストリー」) 	ICP製品のGUIコンソールからの削除を行う	- コンテナ・イメージのタグ単位での削除（例:特定のイメージ・バージョンのみを選択しての削除）ではなく、あるコンテナ・イメージに属する全てのタグがまとめて削除される。 「IBM Knowledge Center - consoleからのイメージの削除」記事に記載の通り、GUIコンソールからの削除を行った際には、画面表示上の削除のみが行われる。そのため、プライベート・イメージ・レジストリーに保管されるコンテナ・イメージの実体を削除するためには、明示的なガーベッジ・コレクションを行わなければならない。