



OpenShift Container Platform 4.2

リリースノート

新機能のハイライトおよび OpenShift Container Platform 4.2 リリースの変更内容

OpenShift Container Platform 4.2 リリースノート

新機能のハイライトおよび OpenShift Container Platform 4.2 リリースの変更内容

法律上の通知

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

以下の OpenShift Container Platform 4.2 リリースノートでは、新機能および拡張機能のすべて、以前のバージョンからの主な技術上の変更点、主な修正、および一般公開バージョンの既知の問題についてまとめています。

目次

第1章 OPENSIFT CONTAINER PLATFORM 4.2 リリースノート	3
1.1. 本リリースについて	3
1.2. 新機能および改良された機能	3
1.3. 主な技術上の変更点	12
1.4. バグ修正	14
1.5. テクノロジープレビューの機能	21
1.6. 既知の問題	25
1.7. エラータの非同期更新	27
第2章 OPENSIFT CONTAINER PLATFORM のバージョン管理ポリシー	39

第1章 OPENSIFT CONTAINER PLATFORM 4.2 リリースノート

Red Hat OpenShift Container Platform では、設定や管理のオーバーヘッドを最小限に抑えながら、セキュアでスケーラブルなリソースに新規および既存のアプリケーションをデプロイするハイブリッドクラウドアプリケーションプラットフォームを開発者や IT 組織に提供します。OpenShift Container Platform は、Java、Javascript、Python、Ruby および PHP など、幅広いプログラミング言語およびフレームワークをサポートしています。

Red Hat Enterprise Linux および Kubernetes にビルドされる OpenShift Container Platform は、エンタープライズレベルの最新アプリケーションに対してよりセキュアでスケーラブルなマルチテナント対応のオペレーティングシステムを提供するだけでなく、統合アプリケーションランタイムやライブラリーを提供します。OpenShift Container Platform を使用することで、組織はセキュリティー、プライバシー、コンプライアンス、ガバナンスの各種の要件を満たすことができます。

1.1. 本リリースについて

Red Hat OpenShift Container Platform ([RHBA-2019:2921](#)) をご利用いただけるようになりました。本リリースでは、CRI-O ランタイムで Kubernetes 1.14 を使用します。以下では、OpenShift Container Platform 4.2 に関連する新機能、変更点および既知の問題について説明します。

OpenShift Container Platform 4.2 クラスターは <https://cloud.redhat.com/openshift> でご利用いただけます。OpenShift Container Platform 向けの Red Hat OpenShift Cluster Manager アプリケーションを使って、OpenShift クラスターをオンプレミスまたはクラウド環境のいずれかにデプロイすることができます。

OpenShift Container Platform 4.2 は、Red Hat Enterprise Linux 7.6 以降、および Red Hat Enterprise Linux CoreOS 4.2 でサポートされます。

コントロールプレーンまたはマスターマシンには Red Hat Enterprise Linux CoreOS (RHCOS) を使用する必要があります。コンピュートまたはワーカーマシンには RHCOS または Red Hat Enterprise Linux 7.6 以降のいずれかを使用できます。



重要

コンピュートマシン用にサポートされているのは Red Hat Enterprise Linux バージョン 7.6 以降であるため、Red Hat Enterprise Linux コンピュートマシンをバージョン 8 にアップグレードすることはできません。

1.2. 新機能および改良された機能

今回のリリースでは、以下のコンポーネントおよび概念に関連する拡張機能が追加されました。

1.2.1. インストールおよびアップグレード

1.2.1.1. OpenShift Container Platform アップグレードの段階的ロールアウト

OpenShift Container Platform 4.1 で、Red Hat はクラスターに対する適切なアップグレードバージョンを推奨するアップグレードチャンネルの概念を導入しました。アップグレードチャンネルは、アップグレードストラテジーを切り分け、更新の頻度を制御するためにも使用されます。チャンネルは OpenShift Container Platform のマイナーバージョンに関連付けられます。たとえば、OpenShift Container Platform 4.2 チャンネルには 4.3 リリースへのアップグレードが含まれることはありません。これにより、管理者は OpenShift Container Platform の次のマイナーバージョンへのアップグレードに関して明

確な決定を行うことができます。チャンネルは更新のみを制御し、インストールするクラスタのバージョンには影響を与えません。OpenShift Container Platform の特定のパッチレベルの **openshift-install** バイナリーは、そのパッチレベルのインストールを常に実行します。

更新のタイプおよびアップグレードチャンネルについての詳細は、「[OpenShift 4.2 Upgrades phased roll out](#)」を参照してください。

アップグレードは Red Hat Service Reliability Engineering (SRE) チームからのデータに基づいて段階的に展開される際にチャンネルに公開されるため、初期リリースでバージョン 4.1.z から 4.2 への更新が利用できるという通知が Web コンソールですぐに表示されない可能性があります。

1.2.1.2. CLI ベースのインストール

OpenShift Container Platform 4.2 では、**openshift-install** という CLI ベースのインストーラーが新たに導入されました。これは、対話型のガイド付きワークフローを使用し、30 分程度でイミュータブルなインストーラーでプロビジョニングされたインフラストラクチャーに OpenShift をプロビジョニングできるように単純化を目的として設計されています。この方法は、OpenShift デプロイメントにおいてホスト OS およびプラットフォームの更新、およびインフラストラクチャー管理のフルスタック自動化を実現し、独自のインフラストラクチャーをプロビジョニングする際の複雑な作業を排除します。すべての必須ではないインストール設定オプションについてのみ最小限のユーザー入力が必要となり、これらのオプションはコンポーネント Operator のカスタムリソース定義 (CRD、Customer Resource Definition) を使用してインストール後に設定されます。

更新はほぼ同じ方法で実行されます。OpenShift の更新コンテンツはまずローカルコンテナレジストリーにミラーリングする必要があり、その後管理者は **oc adm upgrade** に対して更新コンテンツをどこからプルするかを指定します。

1.2.1.3. ネットワークが制限されたインストール

OpenShift Container Platform 4.2 では、接続するネットワークが制限された環境での OpenShift Container Platform クラスタのインストールおよび更新のサポートが導入されました。これは、OpenShift Container Platform コンテンツをホストするための **docker** 2.2 仕様に準拠するコンテナレジストリーを使用するように設計されています。

まず、管理者は Quay.io のコンテンツをそれぞれのローカルのコンテナレジストリーに複製する必要があります。次に、コンテンツのプルを Quay.io からではなくローカルに実行する Ignition 設定を生成するように **openshift-install** を設定できます。これは、ユーザーによってプロビジョニングされるインフラストラクチャー (UPI、User-Provisioned Infrastructure) のデプロイメントでのみ機能するように設計されています。

ネットワークが制限された環境では、OLM カタログの Operator を引き続き使用できます。ただし、オフラインカタログを設定するために Operator ソースを手動でプルする必要があります。この手動プロセスは、OpenShift Container Platform 4.2 での一時的な回避策としてのみ使用されます。今後のリリースでは、自動化ソリューションが提供されます。

詳細は、「[Installing in restricted networks](#)」および「[Using Operator Lifecycle Manager on restricted networks](#)」を参照してください。

1.2.1.4. 3 ノードのベアメタルデプロイメント (テクノロジープレビュー)

OpenShift Container Platform 4.2 では、異なるコンピューターノードまたはワーカーノードのないクラスタをベアメタルクラスタにデプロイする機能をテクノロジープレビュー機能として利用できます。プロビジョニングするベアメタルインフラストラクチャーへのクラスタのインストール時にワーカーマシンを作成しない場合、ルーター Pod を含むすべての Pod が代わりにコントロールプレーンま

たはマスターマシンにデプロイされます。このデプロイメント方法は、OpenShift Container Platform 4.2 のクラウドベースのクラスターでは利用できないことに注意してください。実験用クラスターでテクノロジープレビュー機能を使用することは推奨していません。

1.2.1.5. クラスター全体の egress プロキシ

OpenShift Container Platform 4.2 では、ユーザーによってプロビジョニングされるインフラストラクチャーでの企業プロキシサーバーを使用した OpenShift Container Platform クラスターのインストールおよび更新のサポートが導入されました。プロキシ情報 (httpProxy、httpsProxy、および noProxy) は、インストールプロセスで使用される **install-config.yaml** ファイルに定義でき、**cluster** プロキシオブジェクトを使用してインストール後に管理することもできます。



重要

クラスター全体のプロキシは、サポートされるプロバイダーについてユーザーによってプロビジョニングされるインフラストラクチャーのインストールを使用している場合にのみサポートされます。

また、MITM HTTPS の企業プロキシを許可する独自の CA バンドルを提供するためのサポートも提供されています。

1.2.1.6. 新規のプラットフォーム境界

OpenShift Container Platform はインフラストラクチャー全体を認識し、オペレーティングシステムは管理下に置かれるようになります。これにより、インストールおよび更新がプラットフォームや基礎となるオペレーティングシステム全体でシームレスに行われるようになります。すべてのものが単一エンティティとして管理されます。

1.2.1.7. IPI および UPI

OpenShift Container Platform 4.2 では、フルスタック自動化 (IPI) と既存のインフラストラクチャー (UPI) の主に 2 つのインストールタイプに基づいてインストールが行われます。

フルスタック自動化の場合、インストーラーが OpenShift Container Platform の使用しやすい事前に設定されたデプロイにより、インフラストラクチャーのプロビジョニングを含むインストールのすべての領域を制御します。既存インフラストラクチャーのデプロイメントでは、管理者が独自のインフラストラクチャーを作成し、管理します。この場合、より多くのカスタマイズやより柔軟な運用が可能になります。

1.2.1.8. フルスタック自動化によるデプロイ

OpenShift Container Platform 4.2 では、フルスタック自動化デプロイメントのサポートが拡張し、AWS、Microsoft Azure、Google Cloud Platform (GCP)、および Red Hat OpenStack Platform が対象に含まれています。さらに、(AWS、ベアメタルおよび VMware vSphere がすでに含まれる) ユーザーによってプロビジョニングされるインフラストラクチャーでサポートされるプロバイダーの一覧に GCP が追加されました。

1.2.1.9. Red Hat Cluster Application Migration Tool および Red Hat Control Plane Migration Assistant

CAM および CPMA についての詳細は、「[Where can I find Red Hat Cluster Application Migration Tool \(CAM\) and Red Hat Control Plane Migration Assistant \(CPMA\) now that OpenShift 4.2 has GA'ed](#)」を参照してください。

1.2.2. Operator

1.2.2.1. Operator 関連の新規製品ドキュメント

Operator についてはこれまで OpenShift Container Platform 製品ドキュメントの『**Applications**』ガイドに記載されていましたが、今回より『**Operator**』ガイドが新たにご利用いただけるようになりました。このガイドには、Operator、Operator Lifecycle Manager、および Operator SDK についての既存情報および更新された情報と、OpenShift Container Platform 4.2 に固有の新規の内容が盛り込まれています。

1.2.2.2. スコープ付き Operator のインストール

以前のバージョンでは、**cluster-admin** ロールを持つユーザーのみが Operator をインストールすることができました。OpenShift Container Platform 4.2 では、**cluster-admin** で namespace を選択でき、その namespace では管理者は Operator を各自でインストールできます。**cluster-admin** はこの namespace で ServiceAccount を定義します。この namespace にあるすべてのインストールされた Operator は、この ServiceAccount と同レベルまたはそれよりも低いレベルのパーミッションを取得します。

詳細は、「[Creating policy for Operator installations and upgrades](#)」を参照してください。

1.2.2.3. Ingress Operator

Ingress Operator は、Azure および GCP のインストーラーでプロビジョニングされるインフラストラクチャーに関連した 4.2 のすべての Ingress 機能をサポートします。

1.2.2.4. Machine Config Operator

Machine Config Operator (MCO) はクラスターレベルの設定を提供し、ローリングアップグレードを有効にし、新規ノードと既存ノード間に差異が生じることを防ぎます。

1.2.2.5. Node Feature Discovery Operator

Node Feature Discovery (NFD) Operator は各ノードで利用可能なハードウェア機能を検出し、ノードラベルを使用してそれらの機能を公開します。

NFD によって管理される CPU 機能には、以下が含まれます。

- cpuid
- hardware_multithreading
- power
- pstate

NFD によって管理されるカーネル機能には、以下が含まれます。

- config
- selinux_enabled
- version
- os_version

NFD によって管理されるその他の機能には、以下が含まれます。

- NVMe
- NUMA
- SR-IOV
- GPU

NFD Operator は NFD DaemonSet のインストールおよびライフサイクルを管理します。OperatorHub で NFD Operator にアクセスします。

1.2.2.6. Node Tuning Operator の拡張機能

Node Tuning Operator は最初に OpenShift Container Platform 4.1 で導入されました。これはクラスターノードレベルのチューニングを管理します。デフォルトの CR は標準的なノードレベルのチューニングに使用されることが意図されています。今回の OpenShift Container Platform 4.2 の拡張機能により、高パフォーマンス化などのチューニングのカスタマイズが可能になりました。

カスタムチューニングの場合、独自のチューニング済みのカスタムリソース (CR) を作成します。新規に作成された CR は、デフォルトの CR、およびノードまたは Pod のラベルおよびプロファイルの優先順位に基づいてノードに適用されるカスタムチューニングと組み合わせられます。

1.2.3. Storage

1.2.3.1. ローカルストレージ Operator を使用した永続ボリューム

ローカルストレージ Operator を使用する永続ボリュームが OpenShift Container Platform 4.2 で利用可能になりました。

1.2.3.2. OpenShift Container Storage Interface (CSI)

Container Storage Interface (CSI) が OpenShift Container Platform 4.2 で利用可能になりました。CSI は、Red Hat OpenShift Container Storage (OCS) を有効にし、それらの CSI プラグインと連動させるために Kubernetes で導入されています。CSI の導入により、Kubernetes ボリュームレイヤーの [拡張性](#) は強化されます。

現時点では、API のみご利用いただけます。Operator に含まれる CSI ドライバーは、今後のリリースでご利用いただけます。

1.2.3.3. raw ブロックボリュームのサポート

以下の raw ブロックボリュームは OpenShift Container Platform 4.2 で完全にサポートされるようになりました。

- ローカルボリューム
- クラウドプロバイダー (AWS、GCP、Azure、および vSphere)

iSCSI を使用した raw ブロックボリュームは、[テクノロジープレビュー](#)としてご利用いただけます。

1.2.4. スケーリング

1.2.4.1. クラスターの制限

OpenShift Container Platform 4.2 の [クラスター制限](#) に関する指示が更新されました。

ご使用の環境のクラスター制限を見積もるには、[OpenShift Container Platform Limit Calculator](#) を使用します。

1.2.5. 開発者のエクスペリエンス

1.2.5.1. OpenShift Do

OpenShift Do (odo) は、OpenShift でアプリケーションを作成、ビルド、およびデプロイするための開発者向けの CLI ツールです。odo ツールは完全にクライアントベースで、デプロイに OpenShift Container Platform クラスター内のサーバーを必要としません。これは、ローカルコードの変更を検知し、これをクラスターに自動的にデプロイすると共に、変更を検証するためのインスタントフィードバックをリアルタイムで提供します。複数のプログラミング言語やフレームワークをサポートします。

1.2.5.2. CodeReady コンテナ

CodeReady コンテナは OpenShift Container Platform 4 以降の最小クラスターのローカルデスクトップインスタンスを提供します。このクラスターは、開発およびテスト目的で使用される最小限の環境を開発者に提供します。これには、OpenShift クラスターを実行する CodeReady コンテナの仮想マシンと対話するための **crc** CLI が含まれます。

1.2.6. ノード

1.2.6.1. CRI-O サポート

CRI-O は、Kubernetes のバージョンに一致する kubernetes 固有のコンテナエンジンであり、バージョンの不一致を生じさせないためにサポートを単純化します。既存の Docker および OCI コンテナすべてがサポートされ、それらは CRI-O で適切に実行されるため、導入が簡単です。CRI-O は軽量の kubernetes ネイティブの、OCI と互換性のあるコンテナランタイムであり、OpenShift Container Platform によってライフサイクルが設定され、管理されます。どのコンテナランタイムが使用されているかについて把握する必要はありません。OpenShift Container Platform では、常に適切なコンテナランタイムが使用され、Kubernetes Container Runtime Interface (CRI) の完全な実装が提供されます。さらに、コンテナエンジンを個別に管理する必要もありません。CRI-O には CRI-O の制御機能およびセキュリティーを提供するチューニング可能なパラメーターが含まれます。それらは CRD で簡単に設定でき、設定はクラスター全体に伝播されます。

1.2.6.2. sysctl 設定のホワイトリスト化

システム管理者は、ノードごとに sysctl をホワイトリストに設定できます。すべての安全な sysctl はデフォルトで有効にされ、すべての安全でない sysctl はデフォルトで無効にされます。詳細は、「[Using sysctls in containers](#)」を参照してください。

1.2.6.3. マスターノードがスケジュール対象になる

OpenShift Container Platform 4.2 では、マスターノードがスケジュール対象になりました。詳細は、「[Working with nodes](#)」を参照してください。



重要

Pod をマスターノードにスケジュールすることも可能ですが、クラスターからすべてのワーカーノードを削除する機能はベアメタルに制限されています。「[3 ノードのベアメタルデプロイメント \(テクノロジープレビュー\)](#)」を参照してください。

1.2.7. ネットワーク

1.2.7.1. OpenStack 上にインストーラーでプロビジョニングされる OpenShift

OpenShift Container Platform 4.2 では、Red Hat OpenStack に OpenShift Container Platform クラスターをデプロイするためのフルスタック自動化サポートが導入されました。

インストーラーは、Kubernetes OpenStack Cloud Provider と共に OpenStack API を使用して、仮想マシン (VM)、ネットワーク、セキュリティグループ、オブジェクトストレージなど、OpenShift Container Platform をデプロイし、クラスターを Red Hat OpenStack Platform で実行できるよう適切に設定するために必要なすべての OpenStack リソースを作成します。

OpenShift Container Platform 4.2 は Red Hat OpenStack バージョン 13 でデプロイできます。

1.2.7.2. OVN (テクノロジープレビュー)

現時点で[テクノロジープレビュー](#)である Open vSwitch の Open Virtual Networking (OVN) には、お客様主導の機能要件を加速するなど、数多くの利点があります。これらの要件には事前に有効にされるものがあり、以下のような各種の利点が含まれます。

- OVS による仮想ネットワークの実装により統合がより容易になる。
- SDN ポートフォリオの統合。
- iptables のスケールアップ関連の問題がほぼ存在しなくなる。
- Windows ノードを持つ異種のクラスター。
- オンプレミスおよびクラウドノードにまたがる機能
- フルネットワークポリシーサポート
- Pod ごとの Egress IP。
- 分散 L4 Ingress/Egress ファイアウォール。
- 分散サービスロードバランサー。
- トラフィックの分離とマルチテナンシー。
- Data Plane Development Kit (DPDK) サポート。
- 暗号化されたトンネル。
- IPv6 および DHCPv6。
- QoS、コントロールおよびデータプレーンの分離。

1.2.7.3. プライベートクラスターの内部 Ingress コントローラーの有効化

クラウドプラットフォームで Ingress コントローラーを作成する場合、Ingress コントローラーはデフォルトでパブリッククラウドロードバランサーによって公開されます。

ユーザーは、Ingress コントローラーを内部クラウドロードバランサーで公開できるようになりました。以下は例になります。

```
apiVersion: operator.openshift.io/v1
kind: IngressController
metadata:
  namespace: openshift-ingress-operator
  name: internal
spec:
  endpointPublishingStrategy:
    type: LoadBalancerService
  loadBalancer:
    scope: Internal
```

実装の詳細については、[Kubernetes サービスドキュメント](#)を参照してください。

.spec.endpointPublishingStrategy.loadBalancer.scope は、いったん設定されると変更することができません。スコープを変更するには、Ingress コントローラーを削除し、再作成します。

デフォルト Ingress コントローラーは、削除および再作成を実行して内部で利用可能にすることができます。

詳細は、「[Ingress Operator in OpenShift Container Platform](#)」を参照してください。

1.2.7.4. Kubernetes CNI プラグインの追加および機能強化

機能の拡大に向けて、いくつかの Kubernetes CNI プラグインが OpenShift Container Platform 4.2 で追加され、拡張されています。

これらの SR-IOV ソリューションは OpenShift Container Platform 4.2 でも引き続きテクノロジープレビューとしてご利用いただけます。

- RDMA および RoCE サポート
- SR-IOV VF の DPDK モード
- 受付コントローラー
- Operator

新規 CNI プラグイン:

- IPVLAN
- VLAN を使用したブリッジ
- 静的 IPAM

1.2.7.5. OpenShift クラスターでの GPU の有効化

GPU が Red Hat Enterprise Linux (RHEL) 7 ノードでサポートされるようになりました。これらは CUDA ドライバーおよびドライバーコンテナのサポートがないため、RHEL CoreOS ノードではサポートされません。

1.2.8. Web コンソール

1.2.8.1. コンソールのカスタマイズオプション

OpenShift Container Platform Web コンソールをカスタマイズして、カスタムロゴ、リンク、通知バー、およびコマンドラインのダウンロードを設定できます。これは、Web コンソールを企業や政府の特定要件を満たすように調整する必要がある場合にとくに役立ちます。

詳細は、「[Customizing the web console](#)」を参照してください。

1.2.8.2. 新規 API Explorer

API リソースは、Home → Explore にある **Explore API Resources** ダッシュボードで簡単に検索し、管理できます。

各 API のスキーマとサポートされるパラメーターを確認し、API のインスタンスを管理し、各 API のアクセスを確認できます。

1.2.8.3. Machine Autoscaler

Machine Autoscaler を使用してクラスターをスケールリングできるようになりました。Machine Autoscaler は、クラスターにデプロイされる MachineSet でマシン数を調整します。追加のデプロイメントをサポートするための十分なリソースがクラスターにない場合にマシンを追加します。インスタンスの最小数または最大数の変更などが生じると、それらの変更は MachineAutoscaler がターゲットに設定する MachineSet に即時に適用されます。

1.2.8.4. Developer パースペクティブ

Developer パースペクティブは、開発者中心の視点を Web コンソールに追加します。これは、複数のオプションを使用したアプリケーションの作成や OpenShift Container Platform へのデプロイメントなど、開発者のユースケースに固有のワークフローを提供します。これは、プロジェクト内のアプリケーションを視覚的に表示し、それらのビルドステータスおよびアプリケーションに関連するコンポーネントやサービスを表示し、対話およびモニタリングを容易にします。また、Serverless 機能 (テクノロジープレビュー)、および Eclipse Che を使用してアプリケーションコードを編集するためのワークスペースを作成する機能を組み込みます。

1.2.8.5. Prometheus クエリー

Prometheus クエリーを Web コンソールで直接実行できるようになりました。Monitoring → Metrics に移動します。

1.2.8.6. アイデンティティプロバイダー

クラスターの OAuth 設定ページでは、ユーザーがクラスターにログインするために使用するアイデンティティプロバイダー (IDP) が追加されています。IDP には GitHub、GitLab、Google、LDAP、Keystone などが含まれます。

1.2.8.7. Web コンソールの一般的な更新情報

- ダッシュボードのデザインが新たになり、メトリクスが新たに追加されました。
- **Catalog** は **Developer** パースペクティブに移動しました: **Developer** → **Add+** → **From Catalog** に移動します。

- プロジェクトのステータスは、プロジェクトの詳細ページの **Workloads** タブに移動しました。
- OperatorHub は **Operators** メニューの下に置かれています。
- チャージバックのサポートが利用可能になりました。アプリケーションごとに要求される予約済みおよび使用済みのリソースを表示できます。
- 現時点で非推奨となっているサービスカタログを有効にしなくても、ネイティブテンプレートのサポートが利用可能になりました。

1.3. 主な技術上の変更点

OpenShift Container Platform 4.2 では、主に以下のような技術的な変更点を加えられています。

corsAllowedOrigins

corsAllowedOrigins を設定できるようになりました。詳細は、「[Allowing JavaScript-based access to the API server from additional hosts](#)」を参照してください。

新規 CNI プラグイン

Multus の新規 CNI プラグインとして、bridge および ipvlan の 2 つのプラグインを新たに使用できません。

Cluster Network Operator による SimpleMacvlan のサポート

Cluster Network Operator (CNO) が SimpleMacvlan の設定に対応するようになりました。

ビルドによるレイヤーの維持

OpenShift Container Platform 4.2 では、ビルドがデフォルトでそれぞれのレイヤーを保持します。

Windows でのビルド

ビルドは Windows ノードでスケジュールされません。

Ingress コントローラーのサポートが無効になる

Ingress コントローラー TLS 1.0 および 1.1 サポートは、Mozilla Intermediate のセキュリティープロファイルに一致するように無効にされています。

新規およびアップグレードされた Ingress コントローラーはこれらの TLS バージョンをサポートしなくなりました。

CatalogSourceConfig の不使用による OperatorHub の複雑性の軽減

OperatorHub は、クラスター管理者が対話する API リソースの数を減らし、OpenShift Container Platform 4.2 での新規 Operator のインストールを単純化するために更新されました。

OpenShift Container Platform 4.1 で OperatorHub を使用するために、クラスター管理者は主に OperatorSource および CatalogSourceConfig API リソースと対話する必要がありました。OperatorSource は、Operator バンドルが保管される外部データストアを追加するために使用されません。

CatalogSourceConfig は、クラスターの OperatorSource にある Operator を有効にするために使用されます。この背後では、Operator が OLM で管理されるように Operator Lifecycle Manager (OLM) CatalogSource が設定されていました。

複雑さの軽減を図るため、OpenShift Container Platform 4.2 では OperatorHub は Operator をインストールするワークフローで CatalogSourceConfig を使用しなくなりました。その代わりに、CatalogSource が OperatorSource をクラスターに追加する結果として引き続き作成されます。ただし、サブスクリプションリソースが CatalogSource を使用して直接作成できるようになりました。



注記

OperatorHub は CatalogSourceConfig リソースを使用しなくなりましたが、それらは OpenShift Container Platform で引き続きサポートされます。

グローバルカタログ namespace の変更

OpenShift Container Platform 4.1 では、CatalogSource がデフォルトでインストールされるデフォルトのグローバルカタログ namespace は **openshift-operator-lifecycle-manager** でした。OpenShift Container Platform 4.2 より、これは **openshift-marketplace** namespace に変更されました。

OpenShift Container Platform 4.1 クラスターで OperatorHub から Operator をインストールしている場合、CatalogSource はサブスクリプションと同じ namespace に置かれます。これらのサブスクリプションはこの変更による影響を受けず、クラスターのアップグレード後も引き続き通常通りに動作することが予想されます。

OpenShift Container Platform 4.2 では、Operator を OperatorHub からインストールする場合、作成されるサブスクリプションは新規グローバルカタログ namespace の **openshift-marketplace** にある CatalogSource を参照します。

以前のグローバルカタログ namespace にある既存のサブスクリプションについての回避策

古い **openshift-operator-lifecycle-manager** namespace に既存の CatalogSource がある場合、その CatalogSource を参照する既存のサブスクリプションオブジェクトはアップグレードに失敗し、その CatalogSource を参照する新規サブスクリプションオブジェクトはインストールに失敗します。

このようなアップグレードの失敗を回避するために、以下を実行します。

手順

1. CatalogSource オブジェクトを以前のグローバルカタログ namespace から **openshift-marketplace** namespace に移動します。

1.3.1. 非推奨の機能

サービスカタログ、テンプレートサービスブローカー、**Ansible Service Broker** およびそれらの **Operator** が非推奨になる

OpenShift Container Platform 4.2 では、サービスカタログ、テンプレートサービスブローカー、Ansible Service Broker およびそれらの Operator が非推奨になりました。これらは今後の OpenShift Container Platform リリースで削除されます。

以下の関連する API は今後のリリースで削除されます。

- ***.servicecatalog.k8s.io/v1beta1**
- ***.automationbroker.io/v1alpha1**
- ***.osb.openshift.io/v1**

クラスターロール API が非推奨になる

以下の API は非推奨となり、今後のリリースで削除されます。

- **ClusterRole.authorization.openshift.io: ClusterRole.rbac.authorization.k8s.io** を代わりに使用します。
- **ClusterRoleBinding.authorization.openshift.io: ClusterRoleBinding.rbac.authorization.k8s.io** を代わりに使用します。

- **Role.authorization.openshift.io: Role.rbac.authorization.k8s.io** を代わりに使用します。
- **RoleBinding.authorization.openshift.io: RoleBinding.rbac.authorization.k8s.io** を代わりに使用します。

OperatorSource および **CatalogSourceConfig** が非推奨になる

OperatorSources および **CatalogSourceConfig** は **OperatorHub** から非推奨になりました。以下の関連する API は今後のリリースで削除されます。

- **operatorsources.operators.coreos.com/v1**
- **catalogsourceconfigs.operators.coreos.com/v2**
- **catalogsourceconfigs.operators.coreos.com/v1**

oc の **/oapi** エンドポイントが非推奨になる

oc の **/oapi** エンドポイントの使用は非推奨となり、今後のリリースで削除されます。**/oapi** エンドポイントはグループのパーミッションに依存しない OpenShift Container Platform API を提供しますが、4.1 で削除されました。

oc version の **-short** フラグが非推奨になる

oc version --short フラグが非推奨になりました。**--short** フラグはデフォルトの出力を印刷するために使用されました。

oc adm migrate コマンド

oc adm migrate コマンド、および **oc adm migrate template-instances** を除くそのすべてのサブコマンドは非推奨になりました。

永続ボリュームスナップショット

永続ボリュームスナップショットは OpenShift Container Platform 4.2 で非推奨になっています。

EFS

『OpenShift Container Platform 4.1 リリースノート』では、EFS が誤って一般に利用可能であると示されていました。これは OpenShift Container Platform 4.2 でテクノロジープレビュー機能として組み込まれています。

Recycle 回収ポリシー

Recycle 回収ポリシーが非推奨になりました。動的プロビジョニングが推奨されています。

1.4. バグ修正

ビルド

- ブロックされたレジストリーは、Buildah で使用される **registries.conf** に設定されませんでした。そのため、Buildah はクラスターイメージポリシーによってブロックされたレジストリーにイメージをプッシュすることができました。今回のバグ修正により、ビルド用に生成される **registries.conf** ファイルにブロックされたレジストリーが含まれるようになりました。ビルドでは、イメージのプルおよびプッシュについてブロックされたレジストリーの設定が考慮に入れられるようになりました。(BZ#1730722)
- シェル変数が Source-to-Image (S2B) を使用したビルド設定で参照される際に、Source-to-Image (S2B) を実行するために使用される可能性のある Dockerfile の生成を試行するロジックがそれらの変数の評価を誤って試行しました。その結果、一部のシェル変数は空の値として誤って評価され、これによりビルドエラーが発生し、その他の変数は、それらの誤って試行さ

れた評価に関連したエラーメッセージをトリガーしました。ビルド設定で参照されるシェル変数は適切にエスケープされるようになり、それらは予想通りのタイミングで評価されるようになりました。これらのエラーは以後確認されなくなりました。(BZ#1712245)

- ロジックのバグにより、ビルド後のイメージのレジストリーへのプッシュの試行は、ビルドの BuildConfig が DockerImage タイプの出力を指定しているものの、その出力イメージに指定されている名前にタグコンポーネントが含まれていない場合に失敗しました。その結果、ビルドされたイメージのプッシュの試行は失敗しました。ビルダーは、名前が指定されていない場合に「latest」タグを名前に追加するようになりました。タグコンポーネントを含まない名前の、DockerImage タイプの出力を指定する BuildConfig を使用してビルドされたイメージは、「latest」タグを使用してプッシュできるようになりました。(BZ#1746499)

クラウド認証情報 Operator

- 以前のバージョンでは、Pod にメモリー制限がありました。そのため、認証情報 Operator は多数のプロジェクト/namespace を持つクラスターでクラッシュする可能性がありました。今回のバグ修正により、メモリー制限が取り除かれ、Operator がクラッシュしなくなり、メモリーはクラスター自体によって処理されるようになりました。(BZ#1711402)
- **cloud-credential** ClusterOperator は関連するリソースを定義しないため、Operator ログは **oc adm must-gather** コマンドで生成した tarball に存在しませんでした。今回のバグ修正により、Operator が関連するリソースを追加するように更新され、その結果としてログが含まれるようになりました。(BZ#1717631)

コンテナ

- **rshared** の伝播により、**/sys** ファイルシステムがその上部に再帰的にマウントし、コンテナが「no space left on device (デバイス上に領域がない)」というエラーを出して起動に失敗する可能性がありました。今回のバグ修正により、それぞれの上部に **/sys** が再帰的にマウントされなくなり、その結果としてコンテナが **rshared: true** オプションが設定された状態で正常に実行されるようになりました。(BZ#1711200)
- Dockerfile ビルダーがコンテンツをビルダーコンテキストや直前のステージからではなく、イメージからコピーするように指定するために **--from** フラグを使用した **COPY** 命令を処理した場合、イメージの名前は **FROM** 命令で指定されているかのようにログに記録されることがありました。この名前は、複数の **COPY** 命令がこれを **--from** フラグの引数として指定している場合に複数回一覧表示されました。今回のバグ修正により、ビルダーは、ビルドプロセスの開始時にこのように参照されるイメージのプルについてはトリガーを試行しなくなりました。その結果、**--from** フラグを使用して **COPY** 命令で参照されるイメージは、それらのコンテンツが必要になるまでプルされなくなり、ビルドログは、このようなイメージの名前を指定する **FROM** 命令をログに記録しなくなりました。(BZ#1684427)
- ビルドコンテキストディレクトリーに **.dockerignore** ファイルが含まれる場合に **COPY** および **ADD** 命令を処理するログは、一部のシンボリックリンクおよびサブディレクトリーを適切に処理しませんでした。バグをトリガーする **COPY** または **ADD** 命令の処理を試みる際に、この影響を受けるビルドは失敗しました。今回のバグ修正によりこのケースを処理するロジックが拡張されたため、同様のエラーは発生しなくなりました。(BZ#1707941)

イメージ

- 長期間実行される jenkins エージェントまたはスレーブ Pod に、以前に jenkins マスターで確認された不具合プロセスの現象が生じる可能性があります。Pod が終了するまでいくつかの不具合プロセスがプロセスの一覧に表示されます。今回のバグ修正により、OpenShift Jenkins マスターイメージで **dumb-init** を使用して、jenkins ジョブの処理中に発生するこれらの不具合プロセスをクリーンアップできるようになりました。その結果、エージェントまたはスレーブ Pod 内や、それらの Pod が置かれているホストで表示されるプロセス一覧に不具合プロセスが含まれなくなりました。(BZ#1705123)

- 4.2 での OAuth サポートの変更により、 Jenkins サービスアカウント証明書と OAuth サーバーのルーターで使用される証明書間での異なる証明書の設定の使用が許可されるようになりました。そのため、 Jenkins コンソールにログインできなくなる可能性があります。今回のバグ修正により、 OpenShift Container Platform Jenkins ログインプラグインは、 Pod にマウントされた証明書のほか、 JVM で利用可能なデフォルト証明書を使って TLS 接続を試行するように更新されました。現時点では、 jenkins コンソールにログインできるようになりました。(BZ#1709575)
- OpenShift Container Platform Jenkins 同期プラグインでは、 Jenkins Kubernetes プラグイン PodTemplate について処理する際に同じ名前を持つ ImageStream と ConfigMap による混乱が生じ、一方のタイプのイベントで他方のタイプ用に作成された Pod テンプレートが削除される状態が見られました。今回のバグ修正により、 OpenShift Container Platform Jenkins Sync プラグインが変更され、特定の名称の Pod テンプレートを作成した API オブジェクトタイプを追跡できるようになりました。今回のリリースより、 OpenShift Container Platform Sync プラグインの ConfigMap および ImageStream からのマッピングによって作成された Jenkins Kubernetes プラグイン PodTemplate が、同じ名前を持つ 2 つのタイプがクラスターに存在する場合でも誤って削除されなくなりました。(BZ#1711340)
- Samples Operator 設定を迅速に、かつ連続して削除すると、最後に実行する削除がハングし、 Operator の **ImageChangesInProgress** 状態が **True** のままになる可能性があります。これにより、 Samples Operator の **clusteroperator** オブジェクトが **Progressing==True** のままとなり、クラスターサンプルが不定の状態になりました。今回のバグ修正により、削除ファイナライザーと Samples upsert 間の調整に関連する修正が導入されました。 Samples Operator 設定オブジェクトを迅速に、かつ連続して削除しても、予想通りに正常に機能するようになりました。(BZ#1735711)
- 以前のバージョンでは、プルーナーが単一の要求ですべてのイメージを取得していたため、要求に時間がかかりすぎていました。今回のバグ修正により、ページャーを使用してすべてのイメージを取得できるようになりました。プルーナーはタイムアウトなしですべてのイメージを取得できるようになりました。(BZ#1702757)
- 以前のバージョンでは、インポーターは最大で 3 つの署名のみをインポートできましたが、 registry.redhat.io には通常 4 つ以上の署名が含まれます。そのため、署名がインポートされないことがありました。今回のバグ修正により、インポーターの上限が引き上げられ、署名をインポートできるようになりました。(BZ#1722568)

イメージレジストリー

- 以前のバージョンでは、 CRD に OpenAPI スキーマが含まれませんでした。つまり、 **oc explain** は **config.imageregistry** リソースについて機能しませんでした。今回のバグ修正により OpenAPI スキーマの生成が有効になり、 **oc explain** が **config.imageregistry.operator.openshift.io** リソースについての情報を提供できるようになりました。(BZ#1705752)
- イメージレジストリー Operator は **openshift-image-registry** namespace を関連オブジェクトとして登録しませんでした。そのため、イメージレジストリーまたはイメージレジストリー Operator のデータが **must-gather** から収集されないことがありました。今回のバグ修正により、 **openshift-image-registry** namespace は常にイメージレジストリー Operator の関連オブジェクトに組み込まれるようになりました。その結果、 Pod、デプロイメント、およびサービスを含む **openshift-image-registry** namespace からの基本的な情報が常に **must-gather** によって収集されるようになりました。(BZ#1748436)
- 以前のバージョンでは、 CVO、イメージレジストリー Operator、および **service-ca** インジェクションコントローラーは、イメージレジストリーによって使用される CA バンドルの監視および更新を同時に実行しました。そのため、内部レジストリーと OpenShift の残りの部分との間の信頼を確立するために使用される CA バンドルの削除および再作成が絶えず実行されました。今回のバグ修正により、 CVO は CA バンドルを管理しなくなりました。イメージレジスト

リー Operator は CA バンドルを保持する ConfigMap が作成されることを確認しますが、その内容は管理しません。内部レジストリーの CA バンドルを保持する ConfigMap は、**service-ca** インジェクションコントローラーによって必要な場合にのみ更新されるようになりました。(BZ#1734564)

- TLS キーはレジストリールートに追加されていませんでした。これは TLS キーが **Secret.StringData** に保管され、Operator でシークレットの実際のデータを確認できないためでした。現時点では、Secret.Data が代わりに使用され、Operator は値を認識できるようになりました。(BZ#1719965)
- ドレイン (解放) プロセスでは、image-registry Pod をエビクトするのに最大 600 秒の時間がかかりました。これは、イメージレジストリーが sh から実行され、シグナルがイメージレジストリーに伝播されず、SIGTERM を受信できないためです。今回のリリースより、レジストリープロセスは exec を使用し、レジストリーは pid1 プロセスで、SIGTERM を受信できるようになりました。ドレイン (解放) によりエビクトが正常に実行されるようになりました。(BZ#1729979)
- **must-gather** は、**openshift-image-registry** namespace で PVC およびイベントを収集しませんでした。今回のリリースより、PVC は **must-gather** プロセスの一部として収集されるようになりました。(BZ#1737558)

インストーラー

- Red Hat Enterprise Linux のデフォルトの最小インストールでは、firewalld.service をインストールし、これを有効にしました。ファイアウォールサービスは、**oc rsh/exec/logs** の予想通りの機能を妨げるポートをブロックしていました。今回のリリースより、**firewalld.service** は、テスト標準に準拠するようにインストールされている場合は無効にされるようになりました。(BZ#1740439)
- イメージのデフォルトパスがデフォルト設定から変更される場合、ワーカーは作成されませんでした。今回のリリースより、インストーラーは独自のストレージプールを作成し、使用するようになりました。(BZ#1707479)

kube-apiserver

- ConfigMap を作成して、証明書のローテーションを短縮することが可能でした。この動作はサポートされなくなりました。今回のリリースより、この ConfigMap を作成する場合は、**Upgradable** は kubeapiserver-operator で **False** に設定されるようになりました。つまり、クラスターを更新できなくなりました。(BZ#1722550)

ロギング

- Elasticsearch の動的シード機能は効率的に実行されず、多数のプロジェクトを含むクラスターでは、呼び出しが過剰に実行されました。この問題はキャッシュ機能がないためにさらに複雑になりました。シード機能の効率が悪く、キャッシュ機能がないことから、Elasticsearch の呼び出しは応答が返される前にタイムアウトになりました。今回のバグ修正により、API 呼び出しのキャッシュおよび ACL シード機能が追加されました。この機能強化により、ページのタイムアウトが発生する可能性が低くなりました。(BZ#1705589)
- Logging Operator は、ログタイプを設定する際に情報ストリームのタイプを使用していました。そのため、**stdout** に送信されるログには **INFO** のタグが付けられ、**stderr** に送信されるログには **ERROR** のタグが付けられました。この方法は、情報タイプを常に正しく伝えるものではありませんでした。今回のリリースにより、ログレベルは情報ストリームに基づいて設定されなくなりました。その代わりに、正確なログレベルを判断できない場合、**unknown** に設定されるようになりました。(BZ#1726639)
- クラスターロギングインスタンスの削除時に、クラスターロギング以下のリソースは別個に削

除されました。そのため、Fluentd または Rsyslog DaemonSet の実行中にログコレクターのサービスアカウントが削除された場合、短い空白期間が生じる可能性があります。これにより、この期間に処理されるログに namespace 名などの Kubernetes 情報が含まれないことがありました。今回のバグ修正により、サービスアカウントは、すべての子リソースが削除されるまで待機してから削除されるようになりました。ログコレクターのサービスアカウントなしでコレクターの DaemonSet が実行される可能性はなくなりました。(BZ#1715370)

Web コンソール

- 以前のリリースでは、イベントについてのコンソール Operator のログにより、重複するメッセージが出力されました。依存関係リポジトリのバージョン更新によりこの問題は修正され、コンソール Operator ログのメッセージの重複はなくなりました。(BZ#1687666)
- シークレットの値が難読化されたため、ユーザーは Webhook URL 全体をコピーできませんでした。リンクが追加され、シークレットの値が含まれる Webhook URL 全体をコピーできるようになりました。(BZ#1665010)
- Web コンソールの「Machine」および「Machine Set details」ページには **Events** タブが含まれていませんでした。**Events** タブが追加され、**Machine** および **Machine Set details** ページから選択できるようになりました。(BZ#1693180)
- 以前のバージョンでは、Web コンソールの詳細ページからノードのステータスを表示することはできませんでした。ステータスフィールドが追加され、ユーザーは詳細ページからノードのステータスを表示できるようになりました。(BZ#1706868)
- 以前のバージョンでは、Operator を OperatorHub でインストールした直後に Operator リソースを作成しようとすると、Web コンソールに空のポップアップが表示されることがありました。今回のバグ修正により、Operator が利用可能になる前にリソースの作成を試行する場合に明確メッセージが表示されるようになりました。(BZ#1710079)
- 以前のバージョンでは、Web コンソールの「Deployment Config Details」ページでは、最初のリビジョンがロールアウトされる前にステータスが **Active** と表示されました。今回のバグ修正により、ステータスはロールアウト前は **Updating** と表示され、ロールアウトの完了後は **Up to date** と表示されるようになりました。(BZ#1714897)
- 以前のバージョンでは、Web コンソールのノードのメトリクスチャートが複数のノードの使用率の合計を間違っ表示する場合があります。今回のバグ修正により、ノードページのチャートに該当ノードの使用率のみが表示されるようになりました。(BZ#1720119)
- 以前のリリースでは、OpenID アイデンティティプロバイダーの ca.crt 値は、Web コンソールで作成された場合に適切に設定されませんでした。この問題は修正され、ca.crt が正しく設定されるようになりました。(BZ#1727282)
- 以前のバージョンでは、ユーザーが CRD 一覧から ClusterResourceQuota インスタンスに移動する際にエラーが Web コンソールに表示されました。この問題は修正され、CRD ページから ClusterResourceQuota インスタンスを正常に一覧表示できるようになりました。(BZ#1742952)
- 以前のバージョンでは、ノードのスケジュールできない状態が Web コンソールのノード一覧に表示されませんでした。この情報には、CLI との整合性がありませんでした。コンソールでは、ノードのスケジュールできない状態がノード一覧およびノード詳細ページに表示されるようになりました。(BZ#1748405)
- 以前のバージョンでは、Web コンソールのリソース詳細ページに、設定マップおよびシークレットキーがすべて大文字で表示されました。キー名はファイル名であることが多く、大文字/小文字が区別されることが多いため、これによって問題が生まれました。OpenShift Container

Platform 4.2 の Web コンソールでは、設定マップおよびシークレットキーが適切に表示されるようになりました。(BZ#1752572)

ネットワーク

- Egress IP アドレスは、制限的な NetworkPolicy のある namespace で適切に機能しませんでした。そのため、特定のソースからのトラフィックを受け入れた Pod は、外部サーバーの応答が NetworkPolicy によって誤って拒否されるため、Egress IP 経由で Egress トラフィックを送信できませんでした。今回のバグ修正により、Egress トラフィックからの応答は新規接続としてではなく、応答として適切に認識されるようになりました。Egress IP および NetworkPolicy は連携します。(BZ#1700431)
- 外部ホストと通信するために外部 IP アドレスを使用する Pod が応答しない外部 IP アドレスを受信する場合、Egress IP モニターコードではホストが応答していないと誤って判別していました。その結果、可用性の高い Egress IP は別のノードに切り替わる可能性があります。モニターコードは、Egress ノードが応答しない場合と最終宛先が反応しない場合との区別ができるように修正されました。可用性の高い Egress IP のノード間の切り替えが不要に行われなくなりました。(BZ#1717639)
- デフォルトで、etcd namespace は net id を指定せずに作成されました。そのため、API サーバーコンポーネントは etcd に接続できませんでした。コードは、etcd namespace に **netid=1** を指定するように修正されました。(BZ#1719653)

ノード

- ノードに追加された追加の IP アドレスをマージする際に使用されるアルゴリズムに誤りがありました。そのため、IP アドレスをノードに追加する際に、アドレスの一覧の順序が入れ替えられ、ノードが API サーバーと通信できなくなりました。アドレスのマージアルゴリズムは、アドレスを並び替えないように変更されました。セカンダリー IP アドレスをノードに追加しても、順序が変更されなくなり、ノードは API サーバーと引き続き通信できるようになりました。(BZ#1696628)
- kubeconfig コントローラーに関連する問題により、クラスターを別の OS リリースを使用するバージョンにアップグレードする場合に kubelet Config への変更は元に戻されます。ソースで正しいコントローラーを指定できるようコードが修正されました。kubeletConfig のカスタマイズは保持されます。(BZ#1718726)

oc

- ノードセクターのラベルの誤った検証により、ラベル上のキーの空の値が許可されないことがありました。今回の更新により、ノードセクターのラベル検証メカニズムが修正され、ラベル上のキーの空の値が有効なノードセクターになりました。(BZ#*1683819)
- **oc get** コマンドは、空の結果一覧を受信すると適切な情報を返しませんでした。今回の更新により、**oc get** で空の一覧を受信する際に返される情報が改善されました。(BZ#1708280)

OLM

- Marketplace Cluster Operator は、Marketplace Operator の再起動またはアップグレード時に **degraded** を報告していました。そのため、OpenShift Container Platform のアップグレードテストは失敗しました。今回のバグ修正により、OpenShift Container Platform では Operator が停止している場合に **degraded** を報告しなくなったため、アップグレードテストにパスするようになりました。(BZ#1706867)
- アップグレード時のバグにより、Marketplace Operator のパフォーマンスの低下および終了が生じていました。ClusterOperator ステータスは、Marketplace Operator の正常性について正確な説明を示しませんでした。

以下の要素がこの問題に関連しています。

- 複数の Marketplace Operator の実行、および ClusterOperatorStatus の更新が同時に行われました。
- オペランド (OperatorSource または CatalogSourceConfig) の調整中にエラーが発生すると同期が失敗しました。これにより、ネットワークの問題や無効なオペランドが生じ、パフォーマンスが低下しました。同期の失敗が同期の合計数のしきい値を超える場合にも Marketplace Operator の低下が生じました。
- Telemetry 経由で ClusterOperatorStatus が特定の状態にある理由を特定することは容易ではありません。Telemetry には状態および理由が含まれていますが、Marketplace には理由の設定がありませんでした。
今回のバグ修正により、以下が可能になります。
- Operator SDK によって提供されるリーダー選択により、複数の Marketplace Operator が ClusterOperatorStatus を同時に更新できなくなります。
- Marketplace は、オペランドの調整中にエラーが発生する場合ではなく、オペランドを取得または更新できない場合にのみ degrade 状態を報告します。
- Marketplace には状態の設定時に理由が含まれるようになり、Telemetry で Marketplace Operator の状態についてのより多くの洞察が得られるようになりました。(BZ#1721537)
- Marketplace が所有する CatalogSourceConfig (**csc**) および OperatorSource (**opsrc**) カスタムリソース定義 (CRD) には説明が含まれていませんでした。そのため、**oc explain csc** と **oc explain opsrc** は空の説明を返しました。今回のバグ修正により、OpenAPI CRD 定義が追加され、**oc explain csc** および **oc explain opsrc** が機能するようになりました。(BZ#1723835)
- Marketplace Operator は、OperatorSource に関連するレジストリーデプロイメントの Pod デプロイメント仕様を上書きしました。そのため、ユーザーはノードセクターを追加できませんでした。今回のバグ修正により、OperatorSource 更新に関連してデプロイメント仕様の必須フィールドのみが置き換えられ、ユーザーは OperatorSource に関連付けられた Operator レジストリー Pod にノードセクターを追加できるようになりました。デフォルトで、NodeSelector は存在しません。ユーザーは、レジストリー Pod デプロイメントの Pod 仕様に NodeSelector を追加できるようになりました。たとえば、**community-operators** OperatorSource の場合、**openshift-marketplace** namespace で **community-operators** デプロイメントを編集します。(BZ#1696726)
- Marketplace Operator はレジストリーのデプロイメントをスケールダウンしてから、更新を強制的に実行するためにこれを再び元に戻していました。これにより、レジストリー Pod に問題がある場合には Pod のクラッシュループが生じる可能性があります。今回のバグ修正により、デプロイメントが利用不可能にならないようにレジストリーのデプロイメントのスケールアップ/ダウンを実行するのではなく、アノテーションを使用して更新を強制的に実行できるようになりました。ただし、これだけではこのバグの修正にはならないことに注意してください。**openshift-marketplace** namespace のクラッシュループ状態にある Pod で失敗が生じないようにエンドツーエンドテストに対する修正が必要です。(BZ#1700100)
- **must-gather** ツールでは、ClusterOperator カスタムリソースに Operator に関連付けられたリソースの ObjectReference が設定されるよう **RelatedObjects** というフィールドが必要です。Marketplace にはこのフィールドがないため、**must-gather** ツールは Marketplace Operator についての十分な情報を収集できませんでした。今回の更新により、**RelatedObjects** フィールドに Operator の namespace および OperatorSource、CatalogSourceConfig、および CatalogSource リリースが設定されるようになりました。これにより、**must-gather** ツールで Marketplace Operator についての十分な情報を収集できるようになりました。(BZ#1717439)

- OpenShift Controller Manager Operator を **Unmanaged** または **Removed** に設定することはサポートされないため、対応する ClusterOperator オブジェクトの状態が **Unknown** ステータスになる可能性があります。今回のバグ修正により、OpenShift Controller Manager Operator は、管理状態についてのサポートされない **Unmanaged** および **Removed** 設定を無視するようになりました。この点は、ClusterOperator ステータスの状態についてのメッセージで確認できます。(BZ#1719188)

Red Hat Enterprise Linux CoreOS (RHCOS)

- 以前のバージョンでは、sshd 設定内で ClientAliveInterval が Microsoft Azure で必要とされる 180 に設定されていない場合に SSH 接続がハングしていました。今回のバグ修正により、sshd の設定がデフォルトで 180 に設定され、SSH が Azure 内でハングしなくなりました。(BZ#1701050)

サービスブローカー

- 以前のバージョンでは、Automation Broker はターゲット namespace への一時的な namespace アクセスを付与するためのネットワークポリシーを常に作成していました。そのため、新たに作成されたポリシーにロックされたターゲット namespace と他の namespace が相互に通信できました。今回の修正により、Automation Broker はターゲット namespace についてネットワークポリシーが実施されているかどうかを確認し、ネットワークポリシーがない場合に新規ネットワークポリシーが作成されなくなりました。今回の修正により、Automation Broker は、ターゲット namespace で実行される既存のサービスに影響を与えずに Ansible Playbook Bundle アクションを実行できるようになりました。(BZ#1643303)
- 以前のリリースでは、適切なパーミッションが手動で適用されない限り、OpenShift Ansible Service Broker Operator はメトリクスを Prometheus に渡しませんでした。今回の更新により、Operator のインストールが必要なパーミッションで自動的に行われるようになりました。(BZ#1692281)

テンプレート

- 以前のリリースでは、Samples Operator 設定オブジェクト (**configs.samples.operator.openshift.io**) のカスタムリソース定義には openAPIV3Schema 検証が定義されていませんでした。そのため、**oc explain** はオブジェクトについての役立つ情報を提供できませんでした。今回の修正により、openAPIV3Schema 検証が追加され、**oc explain** がオブジェクトで機能するようになりました。(BZ#1705753)
- 以前のバージョンでは、Samples Operator はシークレット、イメージストリームおよびテンプレートについてのコントローラー/インフォーマーベースの監視機能を維持すべく、直接の OpenShift Container Platform go クライアントを使用して GET 呼び出しを実行しました。このため、OpenShift Container Platform API サーバーに対して不必要な API 呼び出しが行われました。今回の修正により、インフォーマー/リスナー API を活用し、OpenShift Container Platform API サーバーに対するアクティビティーを軽減できるようになりました。(BZ#1707834)
- 以前のバージョンでは、Samples Operator は cluster-reader ロールに集約されるクラスターロールを作成していませんでした。そのため、cluster-reader ロールのあるユーザーは、Samples Operator の設定オブジェクトを読み取ることができませんでした。今回の更新により、Sample Operator のマニフェストが、設定オブジェクトへの読み取り専用アクセスのためのクラスターロールを組み込むように更新され、このロールは cluster-reader ロールに集約されるようになりました。cluster-reader ロールを持つユーザーは、Samples Operator の設定オブジェクトの読み取り、一覧表示、および監視を実行できるようになりました。(BZ#1717124)

1.5. テクノロジープレビューの機能

現在、今回のリリースに含まれる機能にはテクノロジープレビューのものがあります。これらの実験的機能は、実稼働環境での使用を目的としていません。これらの機能に関しては、Red Hat カスタマーポータル以下のサポート範囲を参照してください。

テクノロジープレビュー機能のサポート範囲

以下の表では、**TP** というマークの付いた機能は テクノロジープレビュー を指し、**GA** というマークの付いた機能は 一般公開機能 を指します。- としてマークされている機能は、機能がリリースから削除されているか、非推奨にされていることを示しています。

表1.1 テクノロジープレビュートラッカー

機能	OCP 3.11	OCP 4.1	OCP 4.2
Prometheus クラスター モニタリング	GA	GA	GA
ローカルストレージ永続 ボリューム	TP	TP	GA
ランタイム Pod の CRI- O	GA*[a]	GA	GA
oc CLI プラグイン	TP	TP	TP
サービスカタログ	GA	GA	-
テンプレートサービスブ ローカー	GA	GA	-
OpenShift Ansible Service Broker	GA	GA	-
ネットワークポリシー	GA	GA	GA
Multus	-	GA	GA
プロジェクト追加に関す る新たなフロー	GA	GA	GA
検索カタログ	GA	GA	GA
Cron ジョブ	GA	GA	GA
Kubernetes デプロイメ ント	GA	GA	GA
StatefulSets	GA	GA	GA
明示的なクォータ	GA	GA	GA

機能	OCP 3.11	OCP 4.1	OCP 4.2
マウントオプション	GA	GA	GA
Docker のシステムコンテナ、CRI-O	-	-	-
Hawkular エージェント	-	-	-
Pod の PreSet	-	-	-
experimental-qos-reserved	TP	TP	TP
Pod sysctl	GA	GA	GA。現在の制限については、「 既知の問題 」を参照してください。
中央監査	GA	-	-
外部プロジェクトラフィックの静的 IP	GA	GA	GA
テンプレート完了の検出	GA	GA	GA
replicaSet	GA	GA	GA
Fluentd Mux	TP	TP	TP
クラスター化された MongoDB テンプレート	-	-	-
クラスター化された MySQL テンプレート	-	-	-
Kubernetes リソースのあるイメージストリーム	GA	GA	GA
デバイスマネージャー	GA	GA	GA
永続ボリュームのサイズ変更	GA	GA	GA
Huge Page	GA	GA	GA
CPU ピニング	GA	GA	GA
受付 Webhook	TP	GA	GA

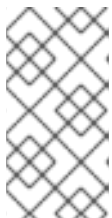
機能	OCP 3.11	OCP 4.1	OCP 4.2
AWS EFS の外部プロビジョナー	TP	TP	TP
Pod Unidler	TP	TP	TP
Node Problem Detector	TP	TP	TP
一時ストレージの制限/要求	TP	TP	TP
CephFS Provisioner	TP	-	-
Podman	TP	TP	TP
Kuryr CNI プラグイン	GA		TP
PID Namespace のコントロール共有	TP	TP	TP
Manila Provisioner	TP	-	-
クラスター管理者コンソール	GA	GA	GA
クラスターの自動スケールリング	GA	GA	GA
Container Storage Interface (CSI)	TP	TP	GA
Operator Lifecycle Manager	TP	GA	GA
Red Hat OpenShift Service Mesh	TP	GA	GA
「完全に自動化された」Egress IP	GA	GA	GA
Pod の優先順位とプリエンプション	GA	GA	GA
Dockerfile のマルチステージビルド	TP	GA	GA
OVN SDN		TP	TP

機能	OCP 3.11	OCP 4.1	OCP 4.2
Prometheus に基づく HPA カスタムメトリク スアダプター		TP	TP
マシンのヘルスチェック		TP	TP
iSCSI での raw ブロック	-	-	TP
OperatorHub			GA
3 ノードのベアメタルデ プロイメント			TP
SR-IOV ネットワーク Operator			TP
[a]* のマークが付いた機能は z-stream パッチでの提供を示しています。			

1.6. 既知の問題

- Service Mesh をインストールしている場合、OpenShift Container Platform をアップグレードする前に Service Mesh をアップグレードします。回避策については、「[Updating OpenShift Service Mesh from version 1.0.1 to 1.0.2](#)」を参照してください。
- クラスターのロールバインディングや SCC (Security Context Constraints) などのリソースを含む、クラスタースコープのリソースが依然としてアプリケーション移行ツールによって処理されません。移行するアプリケーションがソースクラスターのこれらのクラスタースコープのリソースに依存する場合、それらが宛先クラスターで再作成されていることを手動で確認します。これらのリソースを処理できるように、今後のリソースで対象範囲が拡張されます。
- 4.2.0 Dynamic Host Configuration Protocol (DHCP) は現時点でいずれの Multus CNI プラグインとも連動しません。(BZ#1754686)
- クラスターローダーは設定なしで呼び出される場合に失敗します。(BZ#1761925)
- Cluster Network Operator は、追加のネットワークが **additionalNetworks** コレクションから削除される際に、Operator が以前に作成した **NetworkAttachmentDefinition** を削除しません。(BZ#1755586)
- Prometheus Operator は **StatefulSet** をデプロイし、**rules-configmap-reloader** コンテナ上に値が小さすぎるメモリー制限を作成します。(BZ#1735691)
- Multus CNI IPAM で dhcp モードを設定すると失敗します。(BZ#1754682)
- **ClusterResourceQuota** のバージョン 4.1 から 4.2 でのスキーマの変更により、破損が生じます。(BZ#1755125)
- ベアメタルや vSphere を含む各種のデプロイメントについての障害復旧が機能しません。(BZ#1718436)

- Cluster Network Operator から **simpleMacvlanConfig** を削除しても、古い **network Cluster Network Operator-attachment-definition** は削除されません。これらのリソースは手動で削除する必要があります。(BZ#1755586)
- NAD が手動で作成されると、SRIVO Operator Pod がクラッシュします。(BZ#1755188)
- **kube-controller-manager**にプロキシが設定されません。(BZ#1753467)
- HTTPS プロキシを通過する **git clone** 操作は失敗します。非 TLS (HTTP) プロキシは問題なく使用できます。(BZ#1750650)
- 非接続環境のイメージミラーに関連するイメージ参照を使用するビルドは、ミラーで認証が必要な場合にこれらのイメージ参照のプルまたはプッシュに失敗します。(BZ#1745192)
- イメージストリームのインポートはミラーを使用しません。多くの場合、これは非接続環境で使用されます。(BZ#1741391)
- OpenShift Container Platform 4.2 は、この Red Hat OpenStack Platform 15 バグが修正されるまで Red Hat OpenStack Platform 15 では機能しません。(BZ#1751942)
- ビルドがビルドシークレットを使用する場合、**imageOptimizationPolicy: SkipLayers** を使用してレイヤーを非表示の状態にすることを強くお勧めします。そうしない場合、シークレットは **source** および **docker** ストラテジービルドでリークされる可能性があります。
- AllowVolumeExpansion および他の StorageClass 属性は、OpenShift Container Platform のアップグレード時に更新されません。デフォルトの StorageClass を削除し、アップグレードの完了後に ClusterStorageOperator が適切な属性でこれを再作成できるようにすることが推奨されます。(BZ#1751641)
- **Topology** リソースパネルのサーバーレス以外のワークロードにサーバーレスワークロードのリソースが表示されます。(BZ#1760810)
- **Topology** ビュー、**Resources** サイドパネル、および **Deployment Config Details** ページの Pod ステータスの記述に一貫性がありません。(BZ#1760827)
- **Add** ページオプションを使用してアプリケーションが作成される場合、デプロイされたイメージは選択されたターゲットポートを無視し、常に最初のエントリを使用します。(BZ#1760836)
- アプリケーション名やビルドステータスなどの一部の特長について、Edge ブラウザーの **Topology** ビューでレンダリングされません。(BZ#1760858)
- ロールアウト失敗時のアクティブな Pod の判別が **Topology** ビューで正しく行われない可能性があります。(BZ#1760828)
- クラスター全体の egress プロキシが設定され、後に設定が解除される場合、OLM 管理の Operator によって以前にデプロイされたアプリケーションの Pod は **CrashLoopBackOff** 状態になります。これは、デプロイされた Operator がプロキシに依存するように設定されているために生じます。



注記

この問題は、クラスター全体の egress プロキシで作成される環境変数、Volume、および VolumeMount に適用されます。これは、SubscriptionsConfig オブジェクトを使用して環境変数、Volume、および VolumeMount を設定する際にも同じ問題が発生します。

OpenShift Container Platform の今後のリリースで修正が予定されていますが、CLI または Web コンソールを使用してデプロイメントを削除することで問題を回避することができます。これにより、OLM が Deployment を再生成し、正しいネットワーク設定で Pod を開始します。

クラスター管理者は、以下のコマンドを実行して、影響を受ける OLM が管理する全 Deployment の一覧を取得できます。

```
$ oc get deployments --all-namespaces \
  -l olm.owner,olm.owner!=packageserver 1
```

1 影響を受けない **packageserver** を除きます。

([BZ#1751903](#))

- Day 2 のプロキシサポート対応に関して Machine Config Operator (MCO) で問題が生じます。既存のプロキシされていないクラスターがプロキシを使用するように再設定されるタイミングが記述されます。MCO は ConfigMap の新たに設定されたプロキシ CA 証明書を RHCOS 信頼バンドルに適用する必要がありますが、これが正常に機能しません。回避策として、プロキシ CA 証明書を信頼バンドルに手動で追加してから、信頼バンドルを更新する必要があります。

```
$ cp /opt/registry/certs/<my_root_ca>.crt /etc/pki/ca-trust/source/anchors/
$ update-ca-trust extract
$ oc adm drain <node>
$ systemctl reboot
```

([BZ#1784201](#))

1.7. エラータの非同期更新

OpenShift Container Platform 4.2 のセキュリティー、バグ修正、拡張機能の更新は、Red Hat Network 経由で非同期エラータとして発表されます。OpenShift Container Platform 4.2 の全エラータは [Red Hat カスタマーポータル](#) から入手できます。非同期エラータについては、[OpenShift Container Platform ライフサイクル](#) を参照してください。

Red Hat カスタマーポータルのユーザーは、Red Hat Subscription Management (RHSM) のアカウント設定でエラータの通知を有効にすることができます。エラータの通知を有効にすると、登録しているシステムに関連するエラータが新たに発表されるたびに、メールで通知が送信されます。



注記

OpenShift Container Platform のエラータ通知メールを生成させるには、Red Hat カスタマーポータルのユーザーアカウントでシステムが登録されており、OpenShift Container Platform エンタイトルメントを使用している必要があります。

以下のセクションは、これからも継続して更新され、今後の OpenShift Container Platform 4.2 バージョンの非同期リリースで発表されるエラータの拡張機能およびバグ修正に関する情報を追加していきます。たとえば、OpenShift Container Platform 4.2.z などのバージョン付けされた非同期リリースについてはサブセクションで説明します。さらに、エラータの本文がアドバイザーで指定されたスペースに収まらないリリースについては、詳細についてその後のサブセクションで説明します。



重要

OpenShift Container Platform のいずれのバージョンについても、[クラスターの更新](#)に関する指示には必ず目を通してください。

1.7.1. RHBA-2019:2921 - OpenShift Container Platform 4.2 RPM リリースアドバイザー

発行日: 2019-10-16

OpenShift Container Platform リリース 4.2 が公開されました。この更新に含まれるパッケージおよびバグ修正は、[RHBA-2019:2921](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージは、[RHBA-2019:2922](#) アドバイザリーで提供されています。

このアドバイザーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.0 container image list](#)

1.7.2. RHBA-2019:3150 - OpenShift Container Platform 4.2.1 バグ修正の更新

発行日: 2019-10-29

OpenShift Container Platform リリース 4.2 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2019:3151](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2019:3150](#) アドバイザリーで提供されています。

このアドバイザーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.1 container image list](#)

1.7.2.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.3. RHBA-2019:3304 - OpenShift Container Platform 4.2.4 バグ修正の更新

発行日: 2019-11-13

OpenShift Container Platform リリース 4.2.4 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2019:3304](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2019:3303](#) アドバイザリーで提供されています。

このアドバイザーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.4 container image list](#)

1.7.3.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.4. RHBA-2019:3868 - OpenShift Container Platform 4.2.5 バグ修正の更新

発行日: 2019-11-19

OpenShift Container Platform リリース 4.2.5 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2019:3868](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2019:3869](#) アドバイザリーで提供されています。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.5 container image list](#)

1.7.4.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.5. RHBA-2019:3918 - OpenShift Container Platform 4.2.8 バグ修正の更新

発行日: 2019-11-26

OpenShift Container Platform リリース 4.2.8 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2019:3918](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2019:3919](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.8 コンテナイメージの一覧](#)

1.7.5.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.6. RHBA-2019:3952 - OpenShift Container Platform 4.2.9 バグ修正の更新

発行日: 2019-12-03

OpenShift Container Platform リリース 4.2.9 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2019:3952](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2019:3953](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.9 コンテナイメージの一覧](#)

1.7.6.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.7. RHBA-2019:4092 - OpenShift Container Platform 4.2.10 バグ修正の更新

発行日: 2019-12-10

OpenShift Container Platform リリース 4.2.10 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2019:4092](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2019:4093](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.10 コンテナイメージの一覧](#)

1.7.7.1. 機能

本リリースでは、IBM System Z は OpenShift Container Platform 4.2 と互換性があります。インストール手順については、[IBM Z へのクラスタのインストール](#) について参照してください。

制限

IBM System Z の OpenShift Container Platform については、以下の制限に注意してください。

- IBM System Z 向けの OpenShift Container Platform には、以下のテクノロジープレビューが含まれていません。
 - Container-native virtualization (CNV)
 - Knative - Serverless
- 以下のサポートはありません。
 - Service Mesh (Istio、Kiali、および Jaeger を含む)
 - odo
 - CodeReady Workspace
 - Tekton - Pipeline
 - OpenShift Metering (Presto および Hive を含む)
 - Multus プラグイン
- ワーカーノードは Red Hat Enterprise Linux CoreOS を実行する必要があります。
- 永続ストレージのタイプは Filesystem: NFS である必要があります。
- これらの機能は 4.2 の場合に IBM System Z での OpenShift Container Platform に利用できますが、x86 での OpenShift Container Platform 4.2 には利用できません。
 - IBM System Z で有効にされている HyperPAV または FICON 接続の ECKD ストレージ用の仮想マシン。

1.7.7.2. アップグレード

既存の OpenShift Container Platform 4.2 クラスタをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスタの更新](#) について参照してください。

1.7.8. RHBA-2019:4182 - OpenShift Container Platform 4.2.11 バグ修正の更新

発行日: 2019-12-17

OpenShift Container Platform リリース 4.2.11 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2019:4182](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2019:4181](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.11 コンテナイメージの一覧](#)

1.7.8.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.9. RHBA-2020:0013 - OpenShift Container Platform 4.2.13 バグ修正の更新

発行日: 2020-01-06

OpenShift Container Platform リリース 4.2.13 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0013](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0014](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.13 コンテナイメージの一覧](#)

1.7.9.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.10. RHBA-2020:0067 - OpenShift Container Platform 4.2.14 バグ修正の更新

発行日: 2020-01-14

OpenShift Container Platform リリース 4.2.14 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0067](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0066](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.14 コンテナイメージの一覧](#)

1.7.10.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.11. RHBA-2020:0106 - OpenShift Container Platform 4.2.16 バグ修正の更新

発行日: 2020-01-21

OpenShift Container Platform リリース 4.2.16 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0106](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0107](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.16 コンテナイメージの一覧](#)

1.7.11.1. バグ修正

- Kube Controller Manager (KCM) にはそのリリース ConfigMap を再作成するパーミッションがありません。そのため、リリース ConfigMap の削除時に、KCM はリリース ConfigMap のセキュリティを適切に保護できませんでした。今回のバグ修正により、KCM Operator によって作成される KCM パーミッションが更新され、KCM のリリース ConfigMap の削除からの回復が可能になりました。(BZ#1718061)
- AWS Billing 統合には 4.2 リリースサイクル時に障害が発生し、ユーザーがこれを使用することができませんでした。これは以下の問題によって生じていました。
 - パーティションの値/場所が誤って引用されている。
 - AWS Billing データソースのデータベース名の正しくないチェックによりパニックが発生する可能性がある。
 - AWS Billing データソースは **aws-ec2-billing-raw** クエリーの入力ではない。入力の依存関係がないためにデータソースの初期化が依然として実行されていると **table not found** エラーが生じる。
 - Hive テーブルのパーティションの管理ではパーティションの変更時に Hive データベース/スキーマ名を使用しない。これにより、パーティションが管理できない状態になる。
 - Hive テーブル **spec.fileFormat** は無視され、テーブル形式を指定できない状態になる。AWS Billing 統合を修正できるように前述の構造および **values.yaml** フィールドに適切な調整が加えられました。(BZ#1763305)

1.7.11.2. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.12. RHBA-2020:0394 - OpenShift Container Platform 4.2.18 バグ修正の更新

発行日: 2020-02-12

OpenShift Container Platform リリース 4.2.18 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0394](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0395](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.18 コンテナイメージの一覧](#)

1.7.12.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.12.2. 既知の問題

NFD Operator のデフォルトの更新チャンネルは **4.3** です。デフォルトの更新チャンネル値 **4.3** を使用すると、NFD Operator はデプロイに失敗します。デプロイメントを正常に実行するには、Update Channel を **4.2** に設定する必要があります。

1.7.13. RHSA-2020:0463 - 低: OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-02-12

ose-installer-container の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:0463](#) アドバイザリーに記載されています。

1.7.14. RHSA-2020:0476 - 低: OpenShift Container Platform 4.2 イメージセキュリティー更新

発行日: 2020-02-12

ose-baremetal-installer-container および ose-cli-artifacts-container の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:0476](#) アドバイザリーに記載されています。

1.7.15. RHBA-2020:0460 - OpenShift Container Platform 4.2.19 バグ修正の更新

発行日: 2020-02-18

OpenShift Container Platform リリース 4.2.19 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0459](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0460](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.19 コンテナイメージの一覧](#)

1.7.15.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.16. RHBA-2020:0523 - OpenShift Container Platform 4.2.20 バグ修正の更新

発行日: 2020-02-25

OpenShift Container Platform リリース 4.2.20 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0522](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0523](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.20 コンテナイメージの一覧](#)

1.7.16.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.17. RHSA-2020:0526 - Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-02-25

jenkins-slave-base-rhel7-container の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:0526](#) アドバイザリーに記載されています。

1.7.18. RHBA-2020:0613 - OpenShift Container Platform 4.2.21 バグ修正の更新

発行日: 2020-03-03

OpenShift Container Platform リリース 4.2.21 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0613](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0614](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.21 コンテナイメージの一覧](#)

1.7.18.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.18.2. 既知の問題

NFD Operator デプロイメントを正常に実行するには、デフォルトの更新チャンネルである **4.3** を使用し、カスタム namespace ではなくデフォルトの namespace を選択する必要があります。
([BZ#1808503](#))

1.7.19. RHSA-2020:0617 - Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-03-03

更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:0617](#) アドバイザリーに記載されています。

1.7.20. RHSA-2020:0652:1633 - Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-03-03

ose-installer-artifacts-container および ose-installer-container の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:0652](#) アドバイザリーに記載されています。

1.7.21. RHBA-2020:0684 - OpenShift Container Platform 4.2.22 バグ修正の更新

発行日: 2020-03-10

OpenShift Container Platform リリース 4.2.22 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0684](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0685](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.22 コンテナイメージの一覧](#)

1.7.21.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.22. RHSA-2020:0688: Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-03-10

runc の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:0688](#) アドバイザリーに記載されています。

1.7.23. RHSA-2020:0689: Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-03-10

skopeo の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:0689](#) アドバイザリーに記載されています。

1.7.24. RHBA-2020:0786 - OpenShift Container Platform 4.2.23 バグ修正の更新

発行日: 2020-03-18

OpenShift Container Platform リリース 4.2.23 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0786](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0787](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.23 コンテナイメージの一覧](#)

1.7.24.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.25. RHBA-2020:0826 - OpenShift Container Platform 4.2.25 バグ修正の更新

発行日: 2020-03-25

OpenShift Container Platform リリース 4.2.25 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0825](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0826](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.25 コンテナイメージの一覧](#)

1.7.25.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.26. RHSA-2020:0830 - Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-03-25

openshift-enterprise-mediawiki-container の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:0830](#) アドバイザリーに記載されています。

1.7.27. RHBA-2020:0935 - OpenShift Container Platform 4.2.26 バグ修正の更新

発行日: 2020-04-02

OpenShift Container Platform リリース 4.2.26 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:0935](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:0936](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.26 コンテナイメージの一覧](#)

1.7.27.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.28. RHBA-2020:1263 - OpenShift Container Platform 4.2.27 バグ修正の更新

発行日: 2020-04-07

OpenShift Container Platform リリース 4.2.27 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:12560](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:1263](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

OpenShift Container Platform 4.2.27 コンテナイメージの一覧

1.7.28.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.29. RHBA-2020:1398 - OpenShift Container Platform 4.2.28 バグ修正の更新

発行日: 2020-04-14

OpenShift Container Platform リリース 4.2.28 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:1397](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2019:1398](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

OpenShift Container Platform 4.2.28 コンテナイメージの一覧

1.7.29.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.30. RHSA-2020:1401 - Important (重要): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-04-14

buildah の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:1401](#) アドバイザリーに記載されています。

1.7.31. RHSA-2020:1402 - Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-04-14

openshift-enterprise-builder-container の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:1402](#) アドバイザリーに記載されています。

1.7.32. RHBA-2020:1450 - OpenShift Container Platform 4.2.29 バグ修正の更新

発行日: 2020-04-21

OpenShift Container Platform リリース 4.2.29 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:1451](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:1450](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

OpenShift Container Platform 4.2.29 コンテナイメージの一覧

1.7.32.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.33. RHSA-2020:1526 - Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-04-21

openshift-enterprise-hyperkube-container の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:1526](#) アドバイザリーに記載されています。

1.7.34. RHSA-2020:1527 - Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-04-21

openshift の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:1527](#) アドバイザリーに記載されています。

1.7.35. RHBA-2020:2023 - OpenShift Container Platform 4.2.33 バグ修正の更新

発行日: 2020-05-13

OpenShift Container Platform リリース 4.2.33 が公開されました。この更新に含まれるパッケージの一覧は、[RHBA-2020:2022](#) アドバイザリーにまとめられています。この更新に含まれるコンテナイメージおよびバグ修正は、[RHBA-2020:2023](#) アドバイザリーで提供されます。

このアドバイザリーでは、このリリースのすべてのコンテナイメージに関する説明は除外されています。このリリースのコンテナイメージに関する情報については、以下の記事を参照してください。

[OpenShift Container Platform 4.2.33 コンテナイメージの一覧](#)

1.7.35.1. アップグレード

既存の OpenShift Container Platform 4.2 クラスターをこの最新リリースにアップグレードする方法については、[CLI の使用によるクラスターの更新](#)について参照してください。

1.7.36. RHSA-2020:2026 - Important (重要): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-05-13

openshift-cluster-image-registry-operator の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:2026](#) アドバイザリーに記載されています。

1.7.37. RHSA-2020:2027 - Moderate (中程度): OpenShift Container Platform 4.2 セキュリティー更新

発行日: 2020-05-13

oproglottis/gpgme の更新が OpenShift Container Platform 4.2 で利用可能になりました。更新の詳細については、[RHSA-2020:2027](#) アドバイザリーに記載されています。

第2章 OPENSIFT CONTAINER PLATFORM のバージョン管理ポリシー

OpenShift Container Platform では、サポートされているすべての API の厳密な後方互換対応を保証しています。ただし、アルファ API (通知なしに変更される可能性がある) およびベータ API (後方互換性の対応なしに変更されることがある) は例外となります。

Red Hat では OpenShift Container Platform 4.0 を公的にリリースせず、バージョン 3.11 の後に OpenShift Container Platform 4.1 を直接リリースしました。

OpenShift Container Platform のバージョンは、マスターとノードホストの間で一致している必要があります。ただし、クラスターのアップグレード時にバージョンが一時的に一致しなくなる場合を除きます。たとえば、4.2 クラスターではすべてのマスターは 4.2 で、すべてのノードが 4.2 である必要があります。以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.2 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールする必要があります。

セキュリティとは関連性のない理由で API が変更された場合には、古いバージョンの **oc** が更新されるように 2 つ以上のマイナーリリース (例: 3.11、4.1、4.2) 間での更新が行われます。新機能を使用するには新規バージョンの **oc** が必要になる可能性があります。4.2 サーバーにはバージョン 4.1 の **oc** で使用できない機能が追加されている場合や、バージョン 4.2 の **oc** には 4.1 サーバーでサポートされていない追加機能が含まれる場合があります。

表2.1 互換性に関する表

	X.Y (oc クライアント)	X.Y+N ^[a] (oc Client)
X.Y (サーバー)	①	③
X.Y+N ^[a] (サーバー)	②	①

[a] ここで、N は 1 よりも大きい数値です。

- ① 完全に互換性がある。
- ② **oc** クライアントはサーバー機能にアクセスできない場合があります。
- ③ **oc** クライアントでは、アクセスされるサーバーと互換性のないオプションや機能を提供する可能性があります。