



OpenShift Container Platform 4.14

API の概要

OpenShift Container Platform API の概要コンテンツ

OpenShift Container Platform 4.14 API の概要

OpenShift Container Platform API の概要コンテンツ

Legal Notice

Copyright © 2024 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.

Abstract

このドキュメントでは、OpenShift Container Platform API の概要情報を提供します。

Table of Contents

第1章 API 階層について	3
1.1. API 階層	3
1.2. API 階層の API グループへのマッピング	4
1.3. API の非推奨ポリシー	6
第2章 API 互換性ガイドラインについて	9
2.1. API 互換性ガイドライン	9
2.2. API の互換性例外	10
2.3. API の互換性の一般的な用語	10
第3章 KUBELET ログレベルの詳細設定およびログの収集	13
3.1.1 回限りのシナリオとしての KUBELET の変更	13
3.2. 永続的な KUBELET ログレベルの設定	14
3.3. ログの詳細レベルの説明	14
3.4. KUBELET ログの収集	15
第4章 API インデックス	16

第1章 API 階層について



重要

このガイダンスでは、レイヤー化された OpenShift Container Platform オファーリングを説明しません。

ベアメタル設定の API 層は、ハードウェアと直接対話する機能を除いて、仮想化設定にも適用されます。ハードウェアに直接関連するこれらの機能には、ハードウェアベンダーが提供するレベルを超えるアプリケーションオペレーティング環境 (AOE) の互換性レベルはありません。たとえば、グラフィック・プロセッシング・ユニット (GPU) 機能に依存するアプリケーションは、GPU ベンダードライバーによって提供される AOE 互換性の対象となります。

クラウド固有の統合ポイントのクラウド環境の API 層には、ホスティングクラウドベンダーによって提供されるレベルを超える API または AOE の互換性レベルはありません。たとえば、コンピューティング、入力、またはストレージの動的管理を実行する API は、クラウドプラットフォームによって公開されている基盤となる API 機能に依存しています。クラウドベンダーが前提条件の API を変更する場合、Red Hat は、クラウドインフラストラクチャーベンダーが現在提供している機能を使用して API のサポートを維持するための商業的に合理的な取り組みを提供します。

Red Hat をお使いのアプリケーション開発者は、未特定の実装に特有の動作への依存や、特定の API 実装におけるバグへの依存を防ぐために、アプリケーション開発者が依存する動作が、API の公式ドキュメントで明示的に定義されていることを確認することが求められます。たとえば、Ingress ルーターの新しいリリースは、アプリケーションが文書化されていない API を使用するか、未定義の動作に依存している場合に、過去のリリースとは互換性がない場合があります。

1.1. API 階層

商用サポートされているすべての API、コンポーネント、および機能は、次のいずれかのサポートレベルに関連付けられています。

API 階層 1

API とアプリケーション動作環境 (AOE) は、メジャーリリースの間は安定性が確保されます。これらはメジャーリリース内で非推奨になる可能性があります。ただし、次のメジャーリリースまで削除されません。

API 階層 2

API と AOE は、メジャーリリース内で最低 9 か月間、または非推奨の発表から 3 つのマイナーリリースのいずれか長い方で安定しています。

API ティア 3

このレベルは、Operator Hub を介して OpenShift Container Platform に含まれる言語、ツール、アプリケーション、およびオプションのオペレーターに適用されます。各コンポーネントは、API および AOE がサポートされるライフサイクルを指定します。言語ランタイム固有のコンポーネントの新しいバージョンは、マイナーバージョンからマイナーバージョンまで、API および AOE と可能な限り互換性があるように試みます。ただし、マイナーバージョン間の互換性は保証されません。

Operator Hub (Operator およびオペランドと呼ばれる) を介して継続的に更新を受け取るコンポーネントおよび開発ツールは、API 階層 3 とみなされます。開発者は注意を払い、これらのコンポーネントがマイナーリリースごとにどのように変化するかを理解する必要があります。コンポーネント別に文書化されている互換性ガイドラインを参照することが推奨されます。

API 階層 4

互換性は提供されません。API および AOE はいつでも変更できます。これらの機能は、長期サポートを必要とするアプリケーションで使用できません。

Operator がタスクを実行するためにカスタムリソース定義 (CRD) を内部で使用する方法は一般的な方法です。これらのオブジェクトは、Operator に含まれない外部アクターが使用するためのものではなく、非表示にすることを意図しています。CRD が Operator の外部のアクターが使用することを意図していない場合は、Operator **ClusterServiceVersion** (CSV) の

operators.operatorframework.io/internal-objects アノテーションを指定して、対応するリソースが内部使用のみであることを通知する必要があります。また、CRD は、明示的に階層 4 としてラベル付けできます。

1.2. API 階層の API グループへのマッピング

Red Hat が定義した API 階層ごとに、アップストリームコミュニティが上位互換性を維持することを約束している特定の API グループのマッピングテーブルを提供します。互換性レベルが明示的に指定されておらず、以下で具体的に説明されていない API グループには、デフォルトで階層 4 が割り当てられている **v1alpha1** API を除き、デフォルトで API 階層 3 が割り当てられます。

1.2.1. Kubernetes API グループのサポート

接尾辞 ***.k8s.io** で終わる、または接尾辞のないフォーム **version.<name>** を持つ API グループは、Kubernetes の非推奨ポリシーによって管理され、特に指定がない限り、公開されている API バージョンと対応するサポート階層の間の一般的なマッピングに従います。

API バージョンの例	API 階層
v1	階層 1
v1beta1	階層 2
v1alpha1	階層 4

1.2.2. OpenShift API グループのサポート

接尾辞 ***.openshift.io** で終わる API グループは、OpenShift Container Platform の非推奨ポリシーによって管理され、特に指定がない限り、公開されている API バージョンと対応する互換性レベルの間の一般的なマッピングに従います。

API バージョンの例	API 階層
apps.openshift.io/v1	階層 1
authorization.openshift.io/v1	階層 1 の一部で非推奨となった階層 1
build.openshift.io/v1	階層 1 の一部で非推奨となった階層 1
config.openshift.io/v1	階層 1
image.openshift.io/v1	階層 1

API バージョンの例	API 階層
<code>network.openshift.io/v1</code>	階層 1
<code>network.operator.openshift.io/v1</code>	階層 1
<code>oauth.openshift.io/v1</code>	階層 1
<code>imagecontentsourcepolicy.operator.openshift.io/v1alpha1</code>	階層 1
<code>project.openshift.io/v1</code>	階層 1
<code>quota.openshift.io/v1</code>	階層 1
<code>route.openshift.io/v1</code>	階層 1
<code>quota.openshift.io/v1</code>	階層 1
<code>security.openshift.io/v1</code>	RangeAllocation (ティア 4) と *Reviews (ティア 2) を除くティア 1
<code>template.openshift.io/v1</code>	階層 1
<code>console.openshift.io/v1</code>	階層 2

1.2.3. API グループのモニタリングサポート

接尾辞 `monitoring.coreos.com` で終わる API グループには、以下のマッピングがあります。

API バージョンの例	API 階層
<code>v1</code>	階層 1
<code>v1alpha1</code>	階層 1
<code>v1beta1</code>	階層 1

1.2.4. Operator Lifecycle Manager API グループのサポート

Operator Lifecycle Manager (OLM) は、接尾辞が `operators.coreos.com` の API グループを含む API を提供します。これらの API には次のマッピングがあります。

API バージョンの例	API 階層
v2	階層 1
v1	階層 1
v1alpha1	階層 1

1.3. API の非推奨ポリシー

OpenShift Container Platform は、多くのアップストリームコミュニティから供給された多くのコンポーネントで構成されています。コンポーネントのセット、関連する API インターフェイス、および相關する機能は時間の経過とともに進化し、機能を削除するために正式な非推奨が必要になる可能性があると予想されます。

1.3.1. API の一部の使用の終了

OpenShift Container Platform は、複数のコンポーネントが一連の構造化 API を介してクラスターコントロールプレーンによって管理される共有状態と相互作用する分散システムです。Kubernetes の規則に従い、OpenShift Container Platform によって提供される各 API はグループ識別子に関連付けられ、各 API グループは個別にバージョン管理されます。各 API グループは、Kubernetes、Metal3、Multus、Operator Framework、Open Cluster Management、OpenShift 自体などを含む個別のアップストリームコミュニティで管理されます。

各アップストリームコミュニティは、特定の API グループおよびバージョンに対して固有の非推奨ポリシーを定義する場合がありますが、Red Hat はコミュニティ固有のポリシーを各アップストリームコミュニティの統合と認識に基づいて定義された互換性レベルのいずれかに正規化し、エンドユーザーの消費やサポートを簡素化します。

API の非推奨ポリシーとスケジュールは、互換性レベルによって異なります。

非推奨ポリシーは、API のすべての要素を対象とします。以下に例を示します。

- API オブジェクトとしても知られる REST リソース
- REST リソースのフィールド
- バージョン固有の修飾子を除く、REST リソースのアノテーション
- 列挙値または定数値

各グループの最新の API バージョンを除き、古い API バージョンは、次の期間以上、非推奨が発表された後にサポートされる必要があります。

API 階層	期間
階層 1	メジャーリリース内で安定しています。これらはメジャーリリース内で非推奨になる可能性があります。ただし、次のメジャーリリースまで削除されません。

API 階層	期間
階層 2	非推奨の発表から 9か月または 3回のリリースのいずれか長い方。
階層 3	コンポーネント固有のスケジュールを参照してください。
階層 4	なし。互換性は保証されません。

次のルールは、すべての階層 1 API に適用されます。

- API 要素は、グループのバージョンをインクリメントすることによってのみ削除できます。
- API オブジェクトは、一部のバージョンに存在しない REST リソース全体を除いて、情報を失うことなく API バージョン間をラウンドトリップできるようにする必要があります。バージョン間に同等のフィールドが存在しない場合に、データは変換中にアノテーションの形式で保存されます。
- 特定のグループの API バージョンは、API オブジェクト全体が削除される場合を除いて、少なくとも安定した新しい API バージョンがリリースされるまで非推奨にすることはできません。

1.3.2. CLI 要素の廃止

クライアント向けの CLI コマンドは、API と同じ方法でバージョン管理されていませんが、ユーザー向けのコンポーネントシステムです。ユーザーが CLI を操作する 2 つの主な方法は、コマンドまたはフラグを使用することです。これは、このコンテキストでは CLI 要素と呼ばれます。

特に明記されていない限り、または CLI が下位層の API に依存している場合を除き、すべての CLI 要素はデフォルトで API 階層 1 になります。

要素	API 階層
一般提供 (GA)	フラグおよびコマンド
テクノロジープレビュー	階層 1
開発者プレビュー	階層 3
	階層 4

1.3.3. コンポーネント全体の非推奨化

コンポーネント全体を非推奨にする期間とスケジュールは、そのコンポーネントが公開している API の最上位 API 層に関連付けられている期間に直接マップされます。たとえば、ティア 1 とティア 2 の API を表面化したコンポーネントは、ティア 1 の非推奨スケジュールが満たされるまで削除できませんでした。

API 階層	期間

API 階層	期間
階層 1	メジャーリリース内で安定しています。これらはメジャーリリース内で非推奨になる可能性があります。ただし、次のメジャーリリースまで削除されません。
階層 2	非推奨の発表から 9か月または 3回のリリースのいずれか長い方。
階層 3	コンポーネント固有のスケジュールを参照してください。
階層 4	なし。互換性は保証されません。

第2章 API 互換性ガイドラインについて



重要

このガイダンスでは、レイヤー化された OpenShift Container Platform オファリングを説明しません。

2.1. API 互換性ガイドライン

Red Hat は、OpenShift Container Platform との互換性を向上させるために、アプリケーション開発者が以下の原則を採用することを推奨しています。

- アプリケーションのニーズに一致するサポート層を備えた API とコンポーネントを使用します。
- 可能な場合は、公開されたクライアントライブラリーを使用してアプリケーションを構築します。
- アプリケーションは、実行対象として構築された環境と同じくらい新しい環境で実行される場合にのみ、正しく実行されることが保証されます。OpenShift Container Platform 4.14 用にビルドされたアプリケーションは OpenShift Container Platform 4.13 で適切に機能することが保証されていません。
- システムパッケージまたは他のコンポーネントにより提供される設定ファイルに依存するアプリケーションを設計しないでください。アップストリームコミュニティーが、それを保存するように明示的にコミットしない限り、このファイルはバージョン間で変更する場合があります。必要に応じて、前方互換性を維持するために、これらの設定ファイルを通じて Red Hat が提供するインターフェイスの抽象化により異なります。設定ファイルを直接ファイルシステムで変更することは推奨しません。デュアルライターの競合を回避するために、利用可能な場合は、Operator が提供する API と統合することが強く推奨されます。
- **unsupported<FieldName>** の接頭辞が付いた API フィールドや、製品ドキュメントに明示的に記載されていないアノテーションに依存しないでください。
- お使いのアプリケーションより互換性の保証が短いコンポーネントには依存しないでください。
- etcd サーバーで直接ストレージ操作を実行しないでください。すべての etcd アクセスは、api-server を介して、または文書化されたバックアップと復元の手順を介して実行する必要があります。

Red Hat は、アプリケーション開発者が Red Hat Enterprise Linux (RHEL) によって定義された [互換性ガイドライン](#) に従うことを推奨しています。OpenShift Container Platform は、プラットフォーム上でアプリケーションを構築またはホストする際に、以下のガイドラインを強く推奨します。

- 特定の Linux カーネルまたは OpenShift Container Platform のバージョンに依存しないでください。
- **proc**、**sys**、および **debug** ファイルシステム、または他の擬似ファイルシステムから読み込みは避けてください。
- **ioctl**s は使用せず、ハードウェアと直接対話してください。
- コンテナー実行環境を提供する OpenShift Container Platform ホストエージェントと競合しないように、**cgroup** との直接の対話を避けてください。



注記

リリースのライフサイクルの間、Red Hat は、すべてのマイナーリリースと z-stream リリース間で API とアプリケーションオペレーティング環境 (AOE) の互換性を維持するために商業的に合理的な努力をします。ただし、必要に応じて、重大影響を及ぼすセキュリティー問題や、その他の重要な問題に対処するために、この互換性維持の目標に Red Hat が例外を設けることがあります。

2.2. API の互換性例外

以下は、OpenShift Container Platform での互換性の例外です。

サポートされている Operator で行われていない RHEL CoreOS ファイルシステムの変更
現時点では、ホストオペレーティングファイルシステムに加えられた変更がマイナーリリース間で保持されるという保証はありません。ただし、その変更が、Machine Config Operator や Node Tuning Operator などのサポートされている Operator を介して公開されたパブリックインターフェイスを介して行われる場合を除きます。

クラウドまたは仮想化環境でのクラスターインフラストラクチャーの変更

現時点では、クラスターをサポートするクラウドホスティング環境への変更が保持されるという保証はありません。ただし、その変更が製品で公開されているパブリックインターフェイスを介して行われる場合、またはサポートされる設定として文書化されている場合を除きます。クラスターインフラストラクチャープロバイダーは、API を介して製品にその権限を委任する場合を除いて、クラウドまたは仮想化インフラストラクチャーを維持する責任があります。

アップグレードされたクラスターと新規インストールの間の機能のデフォルト

現時点では、製品のマイナーリリースの新規インストールが、以前のマイナーリリースでインストールされ、同等のバージョンにアップグレードされた製品のバージョンと同じ機能デフォルトになるという保証はありません。たとえば、製品の今後のバージョンでは、以前のマイナーバージョンとは異なるデフォルト設定で、クラウドインフラストラクチャーがプロビジョニングされる可能性があります。さらに、選択されていたセキュリティに関するデフォルトの内容は、今後の製品バージョンと、過去の製品バージョンで異なる場合があります。製品の過去のバージョンはアップグレードされますが、特に下位互換性を維持するために、必要に応じて従来の選択肢が保持されます。

"unsupported" または undocumented annotations の接頭辞を持つ API フィールドの使用
製品で API を選択すると、**unsupported<FieldName>** の接頭辞が付いたフィールドが公開されます。

現時点では、このフィールドの使用がリリース間または同一リリース内でサポートされるという保証はありません。製品サポートは、特定の問題をデバッグする場合に、このフィールドで値を指定するようにお客様に依頼することができますが、それ以外での使用はサポートされていません。明示的に文書化されていないオブジェクトでのアノテーションの使用に関しては、マイナーリリース間でのサポートは保証されていません。

製品インストールトポロジーごとの API の可用性

OpenShift ディストリビューションは、サポートされているインストールトポロジーを進化させ続け、あるインストールトポロジーのすべての API が必ずしも別のトポロジーに含まれるとは限りません。たとえば、特定のトポロジーでは、製品のインストールトポロジーと競合している場合、またはそのトポロジーに関連しておらず、特定の API がまったく含まれていない場合に、特定の API への読み取り/書き込みアクセスを制限できます。特定のトポロジーに存在する API は、上記で定義した互換性の階層に従ってサポートされます。

2.3. API の互換性の一般的な用語

2.3.1. アプリケーションプログラミングインターフェイス (API)

API は、他のソフトウェアとの対話を可能にするソフトウェアプログラムによって実装されるパブリックインターフェイスです。OpenShift Container Platform では、API は中央の API サーバーから提供され、すべてのシステム対話のハブとして使用されます。

2.3.2. アプリケーションオペレーティング環境 (AOE)

AOE は、エンドユーザー・アプリケーションプログラムを実行する統合環境です。AOE は、ホストオペレーティングシステム (OS) からの分離を提供するコンテナー化された環境です。AOE を使用すると、少なくともアプリケーションをホスト OS ライブラリーおよびバイナリーから分離して実行できますが、ホスト上の他のすべてのコンテナーと同じ OS カーネルを共有することはできます。AOE はランタイム時に適用され、アプリケーションとそのオペレーティングシステム間のインターフェイスを記述します。これには、プラットフォーム、オペレーティングシステム、および環境間の交差点が含まれ、ユーザー・アプリケーションには、Downward API、DNS、リソースアカウンティング、デバイスアクセス、プラットフォームワークロード ID、コンテナー間の分離、コンテナーとホスト OS 間の分離が含まれます。

AOE には、Container Network Interface (CNI) プラグインの選択や、アドミッションフックをはじめとした製品の拡張機能など、インストールによって異なる可能性のあるコンポーネントは含まれていません。コンテナー環境より下層のクラスターと統合するコンポーネントは、バージョン間で追加のバリエーションが適用される場合があります。

2.3.3. 仮想化環境における互換性

仮想環境は、ベアメタル環境で動作する非特権アプリケーションが、対応する仮想環境で変更されずに実行されるベアメタル環境をエミュレートします。仮想環境は、物理リソースが抽象化された簡易ビューを表示するため、相違点がいくつか存在する場合があります。

2.3.4. クラウド環境での互換性

OpenShift Container Platform は、クラウドプロバイダー固有の統合を介してホスティングクラウド環境との統合ポイントを提供することを選択する場合があります。これらの統合ポイントの互換性は、ネイティブ・クラウド・ベンダーが提供する保証や、OpenShift Container Platform の互換性・ウインドウとの交差部分に固有のものです。OpenShift Container Platform がデフォルトのインストールの一部としてクラウド環境との統合をネイティブに提供する場合、Red Hat は安定したクラウド API エンド・ポイントに対して開発し、安定した非推奨ポリシーを含む将来を見据えた互換性を備えた商業的に合理的なサポートを提供します。クラウドプロバイダーと OpenShift Container Platform の統合の例には、動的ボリューム・プロビジョニング、サービス出力・ド・バランサー統合、Pod ワークロード ID、コンピューティングの動的管理、初期インストールの一部としてプロビジョニングされるインフラストラクチャーが含まれますが、これらに限定されません。

2.3.5. major、minor、および z-stream リリース

Red Hat メジャーリリースは、製品開発における重要なステップを表します。マイナーリリースは、メジャーリリースの範囲内でより頻繁に表示され、今後のアプリケーションの互換性に影響を与える可能性のある非推奨の境界を表します。z-stream リリースは、マイナーリリースへの更新で、関連するマイナーリリースに継続的な修正のストリームを提供します。予期しないセキュリティーの影響に対応するためにこのポリシーが明示的にオーバーライドされる場合を除いて、z-stream リリースで API と AOE の互換性が損なわれることはありません。

たとえば、リリース 4.13.2 では、以下のようになります。

- 4 は、メジャーリリースのバージョンです。
- 13 は、マイナーリリースのバージョンです。

- 2 は、z-stream リリースバージョンです。

2.3.6. 拡張ユーザーサポート (EUS)

重大なバグ修正のための拡張サポートウィンドウを備えた OpenShift Container Platform メジャーリリースのマイナーリリース。ユーザーは、EUS リリース間でマイナーバージョンを段階的に採用することにより、EUS リリース間で移行できます。非推奨のポリシーはマイナーリリース全体で定義され、EUS リリースではなく、留意することが重要です。その結果、EUS ユーザーは、マイナーリリースごとに順次アップグレードしながら、将来の EUS に移行するときに非推奨に対応する必要がある場合があります。

2.3.7. 開発者レビュー

Red Hat によって公式にサポートされておらず、初期段階のテクノロジーを探索するメカニズムを提供することを目的としたオプションの製品機能。デフォルトでは、Developer Preview 機能はオプトインであり、いつでも削除される可能性があります。開発者レビュー機能を有効にすると、機能のスコープによってクラスターがサポートされない可能性があります。

2.3.8. テクノロジープレビュー

機能をテストし、開発プロセス中にフィードバックを提供するために、今後の製品イノベーションへの早期アクセスを提供するオプションの製品機能。この機能は完全にはサポートされておらず、機能的に完全ではない可能性があり、実稼働での使用を目的としていません。テクノロジープレビューの機能を使用するには、明示的なオプトインが必要です。[テクノロジープレビュー機能のサポートスコープ](#) の詳細をご覧ください。

第3章 KUBELET ログレベルの詳細設定およびログの収集

ノードに関するいくつかの問題のトラブルシューティングを行うには、追跡する問題に応じて、kubelet のログレベルの詳細度を確立します。

3.1.1 回限りのシナリオとしての KUBELET の変更

`machine-config(spec":{"paused":false})` の変更によりノードを再起動せずに、1回限りのシナリオで kubelet を変更し、サービスに影響を与えずに kubelet を変更できるようにするには、次の手順に従います。

手順

1. デバッグモードでノードに接続します。

```
$ oc debug node/<node>
```

```
$ chroot /host
```

または、ノードに SSH で接続して root になることもできます。

2. アクセスが確立されたら、デフォルトのログレベルを確認します。

```
$ systemctl cat kubelet
```

出力例

```
# /etc/systemd/system/kubelet.service.d/20-logging.conf
[Service]
Environment="KUBELET_LOG_LEVEL=2"
```

3. 新しい `/etc/systemd/system/kubelet.service.d/30-logging.conf` ファイルで必要な新しい詳細度を定義します。このファイルは、`/etc/systemd/system/kubelet.service.d/20-logging.conf` をオーバーライドします。この例では、詳細度が 2 から 8 に変更になります。

```
$ echo -e "[Service]\nEnvironment=\"KUBELET_LOG_LEVEL=8\"" >
/etc/systemd/system/kubelet.service.d/30-logging.conf
```

4. systemd をリロードし、サービスを再起動します。

```
$ systemctl daemon-reload
```

```
$ systemctl restart kubelet
```

5. ログを収集し、ログレベルの増加を元に戻します。

```
$ rm -f /etc/systemd/system/kubelet.service.d/30-logging.conf
```

```
$ systemctl daemon-reload
```

```
$ systemctl restart kubelet
```

3.2. 永続的な KUBELET ログレベルの設定

手順

- 以下の **MachineConfig** オブジェクトを永続的な kubelet ログレベル設定に使用します。

```

apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: master
  name: 99-master-kubelet-loglevel
spec:
  config:
    ignition:
      version: 3.2.0
    systemd:
      units:
        - name: kubelet.service
          enabled: true
      dropins:
        - name: 30-logging.conf
          contents: |
            [Service]
            Environment="KUBELET_LOG_LEVEL=2"

```

一般に、デバッグレベルのログとして **0~4** を適用し、トレースレベルのログとして **5~8** を適用することを推奨します。

3.3. ログの詳細レベルの説明

ログの詳細レベル	Description
--v=0	常に Operator に表示されます。
--v=1	詳細度が必要ない場合は、妥当なデフォルトログレベル。
--v=2	システムの重要な変更に関連する可能性のある、サービスおよび重要なログメッセージに関する有用な定常状態情報。これは、推奨されるデフォルトのログレベルです。
--v=3	変更についての拡張情報。
--v=4	デバッグレベルの冗長性。
--v=6	要求されたリソースを表示します。
--v=7	HTTP リクエストヘッダーを表示します。

ログの詳細レベル	Description
--v=8	HTTP 要求の内容を表示します。

3.4. KUBELET ログの収集

手順

- kubelet のログレベルの詳細度が適切に設定されたら、次のコマンドを実行してログを収集できます。

```
$ oc adm node-logs --role master -u kubelet
```

```
$ oc adm node-logs --role worker -u kubelet
```

または、ノード内で以下のコマンドを実行します。

```
$ journalctl -b -f -u kubelet.service
```

- マスター コンテナ ログを収集するには、以下のコマンドを実行します。

```
$ sudo tail -f /var/log/containers/*
```

- 全ノードのログを直接収集するには、以下のコマンドを実行します。

```
- for n in $(oc get node --no-headers | awk '{print $1}'); do oc adm node-logs $n | gzip > $n.log.gz; done
```

第4章 API インデックス

API	API グループ
AdminPolicyBasedExternalRoute	k8s.ovn.org/v1
monitoring.openshift.io/v1	Alertmanager
monitoring.coreos.com/v1	AlertmanagerConfig
monitoring.coreos.com/v1beta1	monitoring.openshift.io/v1
APIRequestCount	apiserver.openshift.io/v1
APIServer	config.openshift.io/v1
APIService	apiregistration.k8s.io/v1
AppliedClusterResourceQuota	quota.openshift.io/v1
認証	config.openshift.io/v1
認証	operator.openshift.io/v1
BareMetalHost	metal3.io/v1alpha1
バインディング	v1
BMCEventSubscription	metal3.io/v1alpha1
BrokerTemplateInstance	template.openshift.io/v1
ビルド	build.openshift.io/v1
ビルド	config.openshift.io/v1
BuildConfig	build.openshift.io/v1
BuildLog	build.openshift.io/v1
BuildRequest	build.openshift.io/v1
CatalogSource	operators.coreos.com/v1alpha1
CertificateSigningRequest	certificates.k8s.io/v1
CloudCredential	operator.openshift.io/v1

API	API グループ
CloudPrivateIPConfig	cloud.network.openshift.io/v1
ClusterAutoscaler	autoscaling.openshift.io/v1
ClusterCSIDriver	operator.openshift.io/v1
ClusterOperator	config.openshift.io/v1
ClusterResourceQuota	quota.openshift.io/v1
ClusterRole	authorization.openshift.io/v1
ClusterRole	rbac.authorization.k8s.io/v1
ClusterRoleBinding	authorization.openshift.io/v1
ClusterRoleBinding	rbac.authorization.k8s.io/v1
ClusterServiceVersion	operators.coreos.com/v1alpha1
ClusterVersion	config.openshift.io/v1
ComponentStatus	v1
Config	imageregistry.operator.openshift.io/v1
Config	operator.openshift.io/v1
Config	samples.operator.openshift.io/v1
ConfigMap	v1
コンソール	config.openshift.io/v1
コンソール	operator.openshift.io/v1
ConsoleCLIDownload	console.openshift.io/v1
ConsoleExternalLogLink	console.openshift.io/v1
ConsoleLink	console.openshift.io/v1
ConsoleNotification	console.openshift.io/v1
ConsolePlugin	console.openshift.io/v1

API	API グループ
ConsoleQuickStart	console.openshift.io/v1
ConsoleYAMLSample	console.openshift.io/v1
ContainerRuntimeConfig	machineconfiguration.openshift.io/v1
ControllerConfig	machineconfiguration.openshift.io/v1
ControllerRevision	apps/v1
ControlPlaneMachineSet	machine.openshift.io/v1
CredentialsRequest	cloudcredential.openshift.io/v1
CronJob	batch/v1
CSIDriver	storage.k8s.io/v1
CSINode	storage.k8s.io/v1
CSISnapshotController	operator.openshift.io/v1
CSIStorageCapacity	storage.k8s.io/v1
CustomResourceDefinition	apiextensions.k8s.io/v1
DaemonSet	apps/v1
Deployment	apps/v1
DeploymentConfig	apps.openshift.io/v1
DeploymentConfigRollback	apps.openshift.io/v1
DeploymentLog	apps.openshift.io/v1
DeploymentRequest	apps.openshift.io/v1
DNS	config.openshift.io/v1
DNS	operator.openshift.io/v1
DNSRecord	ingress.operator.openshift.io/v1
EgressFirewall	k8s.ovn.org/v1

API	API グループ
EgressIP	k8s.ovn.org/v1
EgressQoS	k8s.ovn.org/v1
EgressRouter	network.operator.openshift.io/v1
エンドポイント	v1
EndpointSlice	discovery.k8s.io/v1
etcd	operator.openshift.io/v1
イベント	v1
イベント	events.k8s.io/v1
エビクション	policy/v1
FeatureGate	config.openshift.io/v1
FirmwareSchema	metal3.io/v1alpha1
FlowSchema	flowcontrol.apiserver.k8s.io/v1beta3
Group	user.openshift.io/v1
HardwareData	metal3.io/v1alpha1
HelmChartRepository	helm.openshift.io/v1beta1
HorizontalPodAutoscaler	autoscaling/v2
HostFirmwareSettings	metal3.io/v1alpha1
アイデンティティー	user.openshift.io/v1
Image	config.openshift.io/v1
Image	image.openshift.io/v1
ImageContentPolicy	config.openshift.io/v1
ImageContentSourcePolicy	operator.openshift.io/v1alpha1
ImageDigestMirrorSet	config.openshift.io/v1

API	API グループ
ImagePruner	imageregistry.operator.openshift.io/v1
ImageSignature	image.openshift.io/v1
ImageStream	image.openshift.io/v1
ImageStreamImage	image.openshift.io/v1
ImageStreamImport	image.openshift.io/v1
ImageStreamLayers	image.openshift.io/v1
ImageStreamMapping	image.openshift.io/v1
ImageStreamTag	image.openshift.io/v1
ImageTag	image.openshift.io/v1
ImageTagMirrorSet	config.openshift.io/v1
インフラストラクチャー	config.openshift.io/v1
Ingress	config.openshift.io/v1
Ingress	networking.k8s.io/v1
IngressClass	networking.k8s.io/v1
IngressController	operator.openshift.io/v1
InsightsOperator	operator.openshift.io/v1
InstallPlan	operators.coreos.com/v1alpha1
IPPool	whereabouts.cni.cncf.io/v1alpha1
ジョブ	batch/v1
KubeAPIServer	operator.openshift.io/v1
KubeControllerManager	operator.openshift.io/v1
KubeletConfig	machineconfiguration.openshift.io/v1
KubeScheduler	operator.openshift.io/v1

API	API グループ
KubeStorageVersionMigrator	operator.openshift.io/v1
Lease	coordination.k8s.io/v1
LimitRange	v1
LocalResourceAccessReview	authorization.openshift.io/v1
LocalSubjectAccessReview	authorization.k8s.io/v1
LocalSubjectAccessReview	authorization.openshift.io/v1
マシン	machine.openshift.io/v1beta1
MachineAutoscaler	autoscaling.openshift.io/v1beta1
MachineConfig	machineconfiguration.openshift.io/v1
MachineConfigPool	machineconfiguration.openshift.io/v1
MachineHealthCheck	machine.openshift.io/v1beta1
MachineSet	machine.openshift.io/v1beta1
Metal3Remediation	infrastructure.cluster.x-k8s.io/v1beta1
Metal3RemediationTemplate	infrastructure.cluster.x-k8s.io/v1beta1
MutatingWebhookConfiguration	admissionregistration.k8s.io/v1
Namespace	v1
ネットワーク	config.openshift.io/v1
ネットワーク	operator.openshift.io/v1
NetworkAttachmentDefinition	k8s.cni.cncf.io/v1
NetworkPolicy	networking.k8s.io/v1
Node	v1
Node	config.openshift.io/v1
OAuth	config.openshift.io/v1

API	API グループ
OAuthAccessToken	oauth.openshift.io/v1
OAuthAuthorizeToken	oauth.openshift.io/v1
OAuthClient	oauth.openshift.io/v1
OAuthClientAuthorization	oauth.openshift.io/v1
OLMConfig	operators.coreos.com/v1
OpenShiftAPIServer	operator.openshift.io/v1
OpenShiftControllerManager	operator.openshift.io/v1
Operator	operators.coreos.com/v1
OperatorCondition	operators.coreos.com/v2
OperatorGroup	operators.coreos.com/v1
OperatorHub	config.openshift.io/v1
OperatorPKI	network.operator.openshift.io/v1
OverlappingRangeIPReservation	whereabouts.cni.cncf.io/v1alpha1
PackageManifest	packages.operators.coreos.com/v1
PerformanceProfile	performance.openshift.io/v2
PersistentVolume	v1
PersistentVolumeClaim	v1
Pod	v1
PodDisruptionBudget	policy/v1
PodMonitor	monitoring.coreos.com/v1
PodNetworkConnectivityCheck	controlplane.operator.openshift.io/v1alpha1
PodSecurityPolicyReview	security.openshift.io/v1
PodSecurityPolicySelfSubjectReview	security.openshift.io/v1

API	API グループ
PodSecurityPolicySubjectReview	security.openshift.io/v1
PodTemplate	v1
PreprovisioningImage	metal3.io/v1alpha1
PriorityClass	scheduling.k8s.io/v1
PriorityLevelConfiguration	flowcontrol.apiserver.k8s.io/v1beta3
プローブ	monitoring.coreos.com/v1
Profile	tuned.openshift.io/v1
プロジェクト	config.openshift.io/v1
プロジェクト	project.openshift.io/v1
ProjectHelmChartRepository	helm.openshift.io/v1beta1
ProjectRequest	project.openshift.io/v1
Prometheus	monitoring.coreos.com/v1
PrometheusRule	monitoring.coreos.com/v1
プロジェクト	metal3.io/v1alpha1
Proxy	config.openshift.io/v1
RangeAllocation	security.openshift.io/v1
ReplicaSet	apps/v1
ReplicationController	v1
ResourceAccessReview	authorization.openshift.io/v1
ResourceQuota	v1
Role	authorization.openshift.io/v1
Role	rbac.authorization.k8s.io/v1

API	API グループ
RoleBinding	authorization.openshift.io/v1
RoleBinding	rbac.authorization.k8s.io/v1
RoleBindingRestriction	authorization.openshift.io/v1
ルート	route.openshift.io/v1
RuntimeClass	node.k8s.io/v1
スケーリング	autoscaling/v1
スケジューラー	config.openshift.io/v1
Secret	v1
SecretList	image.openshift.io/v1
SecurityContextConstraints	security.openshift.io/v1
SelfSubjectAccessReview	authorization.k8s.io/v1
SelfSubjectRulesReview	authorization.k8s.io/v1
SelfSubjectRulesReview	authorization.openshift.io/v1
サービス	v1
ServiceAccount	v1
ServiceCA	operator.openshift.io/v1
ServiceMonitor	monitoring.coreos.com/v1
StatefulSet	apps/v1
ストレージ	operator.openshift.io/v1
StorageClass	storage.k8s.io/v1
StorageState	migration.k8s.io/v1alpha1
StorageVersionMigration	migration.k8s.io/v1alpha1

API	API グループ
SubjectAccessReview	authorization.k8s.io/v1
SubjectAccessReview	authorization.openshift.io/v1
SubjectRulesReview	authorization.openshift.io/v1
Subscription	operators.coreos.com/v1alpha1
Template	template.openshift.io/v1
TemplateInstance	template.openshift.io/v1
ThanosRuler	monitoring.coreos.com/v1
TokenRequest	authentication.k8s.io/v1
TokenReview	authentication.k8s.io/v1
Tuned	tuned.openshift.io/v1
User	user.openshift.io/v1
UserIdentityMapping	user.openshift.io/v1
UserOAuthAccessToken	oauth.openshift.io/v1
ValidatingWebhookConfiguration	admissionregistration.k8s.io/v1
VolumeAttachment	storage.k8s.io/v1
VolumeSnapshot	snapshot.storage.k8s.io/v1
VolumeSnapshotClass	snapshot.storage.k8s.io/v1
VolumeSnapshotContent	snapshot.storage.k8s.io/v1