Red Hat Enterprise Linux 9

RHEL 8 から RHEL 9 へのアップグレード

Red Hat Enterprise Linux 8 から Red Hat Enterprise Linux 9 へのインプレースアップグレードの手順

Last Updated: 2022-07-28

Red Hat Enterprise Linux 9 RHEL 8 から RHEL 9 へのアップグレード

Red Hat Enterprise Linux 8 から Red Hat Enterprise Linux 9 へのインプレースアップグレードの手順

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本書は、Leapp ユーティリティーを使用して、Red Hat Enterprise Linux 8 から Red Hat Enterprise Linux 9 へのインプレースアップグレードを実行する方法を説明します。既存の RHEL 8 オペレーティングシステムは、インプレースアップグレード時に RHEL 9 バージョンに置き換えられます。

目次

多様性を受け入れるオープンソースの強化 3

RED HAT ドキュメントへのフィードバックの提供 4

主な移行の用語 5

第1章 サポート対象のアップグレードパス 6

第2章 アップグレードの計画 7

第3章 アップグレードの準備 9
3.1. アップグレードに向けた RHEL 8 システムの準備 9
3.2. アップグレードに向けた SATELLITE システムの準備 12

第4章 アップグレード前のレポートの確認 14
4.1. コマンドラインからのアップグレード可能性の評価 14
4.2. WEB コンソールを介したアップグレードの可能性の評価および自動修復の適用 15

第5章 RHEL 8 から RHEL 9 へのアップグレードの実行 21

第6章 RHEL 9 システムのアップグレード後の状態の確認 23

第7章 アップグレード後のタスクの実行 24

第8章 セキュリティーポリシーの適用 25
8.1. SELINUX モードの ENFORCING への変更 25
8.2. システム全体の暗号化ポリシー 26
8.3. セキュリティーベースラインが強化されたシステムのアップグレード 27
8.4. USBGUARD ポリシーの確認 28
8.5. FAPOLICYD データベースの更新 29
8.6. DBM から SQLITE への NSS データベースの更新 30
8.7. BERKELEY DB 形式から GDBM への CYRUS SASL データベースの移行 30

第9章 トラブルシューティング 32
9.1. トラブルシューティングのリソース 32
9.2. トラブルシューティングのヒント 32
9.3. 既知の問題 34
9.4. サポートの利用 37

第10章 関連情報 38

付録A RHEL 8 リポジトリー 39

付録B RHEL 9 のリポジトリー 41

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバックの提供

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
  1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
  2. マウスカーソルで、コメントを追加する部分を強調表示します。
  3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
  4. 表示される手順に従ってください。
- Bugzilla を介してフィードバックを送信するには、新しいチケットを作成します。
  1. Bugzilla の Web サイトに移動します。
  2. Component で **Documentation** を選択します。
  3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
  4. **Submit Bug** をクリックします。

主な移行の用語

以下の移行用語はソフトウェア業界で一般的に使用されますが、これらの定義は Red Hat Enterprise Linux (RHEL) に固有のものです。

Update

ソフトウェアパッチと呼ばれることもあります。更新は現行バージョン、オペレーティングシステム、または実行中のソフトウェアに追加されます。ソフトウェア更新は、問題またはバグに対応し、テクノロジーの操作が改善されます。RHEL では、更新は、RHEL 8.1 から 8.2 への更新といったマイナーリリースに関連します。

アップグレード

アップグレードは、現在実行しているアプリケーション、オペレーティングシステム、またはソフトウェアを置き換える場合です。通常、まず Red Hat の指示に従い、データをバックアップします。RHEL をアップグレードすると、以下の 2 つのオプションがあります。

- **In-place upgrade:** インプレースアップグレードの場合は、以前のバージョンを削除せずに、以前のバージョンを新しいバージョンに置き換えます。設定や設定と共にインストールされたアプリケーションとユーティリティーは、新規バージョンに組み込まれています。

- **clean install:** clean install は、以前にインストールされたオペレーティングシステム、システムデータ、設定、およびアプリケーションのすべてのトレースを削除し、最新バージョンのオペレーティングシステムをインストールします。システムに以前のデータまたはアプリケーションが必要ない場合や、以前のビルドに依存しない新規プロジェクトを開発する場合は、クリーンインストールに適しています。

オペレーティングシステムへの変換

変換は、オペレーティングシステムを別の Linux ディストリビューションから Red Hat Enterprise Linux に変換する際に使用されます。通常、まず Red Hat の指示に従い、データをバックアップします。

移行

通常、移行はプラットフォーム (ソフトウェアまたはハードウェア) の変更を示しています。Windows から Linux への移行は移行です。ユーザーをラップトップから別のサーバーに移動するか、あるサーバーから別のサーバーに会社を移行することは移行です。ただし、ほとんどの移行ではアップグレードが関係し、相互に意味のある用語が使用されることがあります。

- **RHEL への移行:** 既存のオペレーティングシステムの RHEL への移行

- **RHEL 間での移行: RHEL** のあるバージョンから別のバージョンへのアップグレード

第1章 サポート対象のアップグレードパス

インプレースアップグレードは、システムの RHEL 8 オペレーティングシステムを RHEL 9 バージョンに置き換えます。

重要

RHEL 7 から RHEL 9 へのインプレースアップグレードを直接実行することはできません。ただし、RHEL 7 から RHEL 8 へのインプレースアップグレードを実行してから、RHEL 9 への 2 回目のインプレースアップグレードを実行することはできます。詳細は、RHEL7 から RHEL8 へのアップグレードを 参照してください。

現在、以下のソースの RHEL 8 マイナーバージョンから、以下のターゲットの RHEL 9 マイナーバージョンへインプレースアップグレードを実行できます。

表1.1 サポート対象のアップグレードパス

製品バリアント	ソース OS バージョン	ターゲット OS バージョン
RHEL	RHEL 8.6	RHEL 9.0

サポート対象のアップグレードパスの詳細は、Supported in-place upgrade paths for Red Hat Enterprise Linux を参照してください。

第2章 アップグレードの計画

インプレースアップグレードは、システムを RHEL の次のメジャーバージョンにアップグレードする方法です。この方法は、推奨され、サポートされています。

RHEL 9 にアップグレードする前に、以下を考慮する必要があります。

- **オペレーティングシステム** - オペレーティングシステムは、以下の条件下で **Leapp** ユーティリティーでアップグレードが可能です。
  - ソース OS のバージョンは、以下のサポートされるアーキテクチャーのいずれかを持つシステムにインストールされています。
    - 64 ビット Intel、AMD、および ARM
    - IBM POWER (リトルエンディアン)
    - 64 ビット IBM Z
      詳細は、Red Hat certified hardware を参照してください。
  - RHEL 9 の最小 ハードウェア要件 を満たしている。
  - 提供されている最新の RHEL 8.6 および RHEL 9.0 コンテンツへのアクセス。詳細は、Preparing a RHEL 8 system for the upgrade の手順 1 を参照してください。
- **アプリケーション** - **Leapp** を使用して、システムにインストールされているアプリケーションを移行できます。ただし、特定のケースでは、アップグレード時に **Leapp** が実行するアクションを指定するカスタムアクターを作成する必要があります。たとえば、アプリケーションの再設定や特定のハードウェアドライバーのインストールなどです。詳細は、「Handling the migration of your custom and third-party applications」を参照してください。Red Hat は、カスタムアクターに対応していません。

  重要

  RHEL 9 では、**SHA1** は非推奨になりました。システムに **RSA/SHA1** 署名のあるパッケージが含まれる場合は、アップグレードは行われません。アップグレードの前に、これらのパッケージを削除するか、**RSA/SHA256** 署名のあるパッケージのベンダーにお問い合わせください。詳細は、SHA-1 deprecation in Red Hat Enterprise Linux 9 を参照してください。
- **セキュリティー** - アップグレード前にこの要素を評価し、アップグレードプロセスの完了時に追加の手順を実行する必要があります。特に以下の点を考慮してください。
  - アップグレードの前に、システムが準拠しなければならないセキュリティー標準を定義し、RHEL 9 におけるセキュリティーの変更 について理解してください。
  - **Leapp** ユーティリティーは、アップグレードプロセス時に SELinux モードを Permissive に設定します。
  - FIPS モードでのシステムのインプレースアップグレードはサポートされていません。
  - アップグレードが完了したら、セキュリティーポリシーを再評価し、再適用します。セキュリティーポリシーの適用および更新の詳細は、セキュリティーポリシーの適用 を参照してください。

- **ストレージおよびファイルシステム** - アップグレードする前に、必ずシステムのバックアップを作成してください。たとえば、ReaR(Relax-and-Recover)ユーティリティー、LVM スナップショット、RAID 分割、または仮想マシンのスナップショットを使用できます。

- **高可用性**: High Availability アドオンを使用したシステムのアップグレードはサポートされていません。

- **ダウンタイム** - アップグレードプロセスには数分から数時間かかる場合があります。

- **Satellite** - Satellite を介してホストを管理する場合は、Satellite Web UI を使用して、RHEL 8 から RHEL 9 に複数のホストを同時にアップグレードできます。詳細は、Upgrading Hosts from RHEL 7 to RHEL 8を参照してください。

- **パブリッククラウド** - インプレースアップグレードは、Red Hat Update Infrastructure (RHUI) を使用する Amazon Web Services (AWS) のオンデマンド Pay-As-You-Go (PAYG) インスタンスでサポートされます。RHUI を AWS インスタンスで使用するには、Red Hat が認定する公式の RHEL High Availability Amazon Machine Image(AMI)が必要です。インプレースアップグレードは、RHEL サブスクリプションに RHSM を使用するすべてのパブリッククラウドの Bring Your Own Subscription インスタンスでもサポートされます。

- **言語**: すべての **Leapp** のレポート、ログ、その他の生成されたドキュメントは、言語設定に関わらず、英語で表示されます。

- **ブートローダー** - RHEL 8 または RHEL 9 のブートローダーを BIOS から UEFI に切り替えることはできません。RHEL 8 システムで BIOS を使用し、RHEL 9 システムでは UEFI を使用する必要がある場合は、インプレースアップグレードの代わりに RHEL 8 の新規インストールを実行します。詳細は、Is it possible to switch the BIOS boot to UEFI boot on preinstalled Red Hat Enterprise Linux machine? を参照してください。

- **既知の制限** - 現在、 **Leapp** の注目すべき既知の制限には以下が含まれます。

  - 現在、ディスク全体またはパーティションの暗号化、またはファイルシステムの暗号化は、インプレースアップグレードの対象となるシステムでは使用できません。

  - ネットワークベースのマルチパスやネットワークストレージマウントは、システムパーティション (iSCSI、NFS など) として使用できません。

  - 現在、インプレースアップグレードは、RHEL サブスクリプションに Red Hat Update Infrastructure を使用して Red Hat Subscription Manager (RHSM) を使用しない、残りのパブリッククラウド (Microsoft Azure、Huawei Cloud、Alibaba Cloud、Google Cloud) のオンデマンド PAYG インスタンスではサポートされません。

既知の問題 も参照してください。

Red Hat Insights を使用して、Insights に登録したどのシステムが RHEL 9 へのサポート対象のアップグレードパスであるかを確認できます。これを行うには、Insights の該当する Advisor 推奨事項 に移動し、**Actions** ドロップダウンメニューで推奨事項を有効にして、**Affected systems** の見出しにある一覧を確認します。Advisor 推奨は RHEL 8 マイナーバージョンのみを考慮し、システムのアップグレード前の評価は行わないことに注意してください。「 Advisor Service Recommendations 」も参照してください。

第3章 アップグレードの準備

アップグレード後に問題を回避し、システムを RHEL の次のメジャーバージョンにアップグレードできることを確認するには、アップグレード前に必要なすべての準備手順を完了してください。

すべてのシステムで、Preparing a RHEL 8 system for the upgrade で説明されている準備手順を実施する必要があります。さらに、Satellite Server に登録されたシステムでは、Preparing a Satellite system for the upgrade で説明されている準備手順も実行する必要があります。

3.1. アップグレードに向けた RHEL 8 システムの準備

この手順では、**Leapp** ユーティリティーを使用して、RHEL 9 へのインプレースアップグレードを実行する前に必要な手順を説明します。

アップグレードプロセス中に Red Hat Subscription Manager (RHSM) を使用する予定がない場合は、Upgrading to RHEL 9 without Red Hat Subscription Manager の手順に従ってください。

前提条件

- システムが、アップグレードの計画 に記載されている条件を満たしている。

手順

1. 以前に RHEL 7 から RHEL 8 へのインプレースアップグレードを実行した場合、システムに **/root/tmp_leapp_py3** ディレクトリーが存在する場合はこれを削除します。

```
# rm -rf /root/tmp_leapp_py3
```

重要

アップグレードの実行時に **/root/tmp_leapp_py3** ディレクトリーがシステムに存在する場合は、アップグレード後にシステムが破損する可能性があります。

2. Red Hat Subscription Manager を使用して、システムが Red Hat コンテンツ配信ネットワーク (CDN) または Red Hat Satellite に正常に登録されていることを確認します。

3. システムが Satellite Server に登録されている場合は、アップグレードに向けた Satellite システムの準備 の手順を実行して、システムがアップグレードの要件を満たしていることを確認します。

4. Red Hat Enterprise Linux Server サブスクリプション が割り当てられていることを確認します。以下に例を示します。

```
# subscription-manager list --installed
+-------------------------------------------+
    Installed Product Status
+-------------------------------------------+
Product Name:   Red Hat Enterprise Linux x86_64
Product ID:     479
Version:        8.6
Arch:           x86_64
Status:         Subscribed
```

5. 適切なリポジトリーが有効になっていることを確認します。以下のコマンドは、64 ビット Intel アーキテクチャーの Base リポジトリーおよび Appstream リポジトリーを有効にします。その他のアーキテクチャーについては、RHEL 8 リポジトリー を参照してください。

```
# subscription-manager repos --enable rhel-8-for-x86_64-baseos-rpms --enable rhel-8-for-x86_64-appstream-rpms
```

注記

CodeReady Linux Builder リポジトリーまたは Supplementary リポジトリーも有効にすることができます。RHEL 8 リポジトリー の一覧を参照してください。この場合、**Leapp** は、RHEL 8 CodeReady Linux Builder リポジトリーまたは RHEL 8 Supplementary リポジトリーをそれぞれ有効にします。詳細は、Package manifest を参照してください。

6. RHSM を使用してサブスクライブしたシステムの場合は、システムを RHEL 8.6 にロックします。

```
# subscription-manager release --set 8.6
```

7. 必要に応じて、カスタムリポジトリーを使用する場合は、「Customizing your Red Hat Enterprise Linux in-place upgrade」の指示に従って設定します。

8. **dnf versionlock** プラグインを使用して、特定のバージョンにパッケージをロックする場合は、次のコマンドを実行してロックを解除します。

```
# dnf versionlock clear
```

詳細は、How to restrict dnf to install or upgrade a package to a fixed specific package version? を参照してください。

9. AWS で Red Hat Update Infrastructure (RHUI) を使用してアップグレードする場合は、必要な RHUI リポジトリーを有効にして、必要な RHUI パッケージをインストールし、システムがアップグレードの準備ができていることを確認します。

```
# dnf config-manager –set-enabled rhui-client-config-server-8
# dnf -y install rh-amazon-rhui-client-ha leapp-rhui-aws
```

10. すべてのパッケージを最新の RHEL 8 バージョンに更新します。

```
# dnf update
```

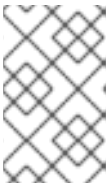
11. システムを再起動します。

```
# reboot
```

12. **Leapp** ユーティリティーをインストールします。

```
# dnf install leapp-upgrade
```

現在、**leapp-upgrade-el8toel9** RPM パッケージを含む **leapp** パッケージおよびバージョン 0.16.0 以降の **leapp-repository** パッケージが必要なことに注意してください。



注記

システムにインターネットアクセスがない場合は、Red Hat カスタマーポータルから Preupgrade Assistant および Red Hat Upgrade Tool をダウンロードします。

13. RPM パッケージの変更、RPM リポジトリーマッピング、サポート対象外のドライバーやデバイスなど、必要な追加データファイルの最新バージョンにアクセスできることを確認してください。

    a. アップグレードに RHSM を使用する場合で、システムが cloud.redhat.com にアクセスでき、必要なデータファイルの以前のバージョンをダウンロードしていない場合は、それ以上のアクションは必要ありません。データファイルは cloud.redhat.com から自動的にダウンロードされます。これは、開発者サブスクリプションにも適用されます。

    b. ナレッジベースアーティクル Leapp utility metadata in-place upgrades of RHEL for disconnected upgrades に添付されているデータファイルをダウンロードし、これを **/etc/leapp/files/** ディレクトリーに置きます。現在、**leapp-data17.tar.gz** アーカイブまたはそれ以降のデータファイルが必要になることに注意してください。これは、以下のシナリオでアップグレードを成功させるために必要になります。

        i. RHUI を使用してパブリッククラウドでアップグレードします。Red Hat サブスクリプションまたは Red Hat カスタマーポータルアカウントをお持ちでない場合は、ナレッジベースの記事にアクセスし、必要なデータパッケージをダウンロードできるように、無料の RHEL 開発者サブスクリプションを作成してください。詳細は「How do I get a no-cost Red Hat Enterprise Linux Developer Subscription or renew it?」を参照してください。

        ii. お使いのシステムにインターネットアクセスがありません。

        iii. アップグレードに RHSM を使用し、必要なデータファイルの古いバージョンを以前ダウンロードしたが、アップグレードを実行しませんでした (例: 自動化スクリプトの作成など)。古いバージョンのデータファイルを削除して、最新のファイルバージョンを自動的にダウンロードすることもできます。

14. アップグレードの失敗を防ぐために一時的にウイルス対策ソフトウェアを無効にします。

15. 設定管理システムがインプレースアップグレードプロセスに干渉しないことを確認します。

    - **Puppet**、**Salt**、**Chef** などのクライアントサーバーアーキテクチャーで設定管理システムを使用する場合は、**leapp preupgrade** コマンドを実行する前にシステムを無効にします。アップグレード時に問題が発生するのを防ぐために、アップグレードが完了するまで設定管理システムを有効にしないでください。

    - **Ansible** などのエージェントレスアーキテクチャーで設定管理システムを使用する場合は、RHEL 8 から RHEL 9 へのアップグレードの実行 で説明されているように、インプレースアップグレード中に、Ansible Playbook などの設定およびデプロイメントファイルを実行しないでください。
      設定管理システムを使用したアップグレード前およびアップグレードプロセスの自動化は、Red Hat ではサポートされていません。詳細は、Using configuration management systems to automate parts of the Leapp pre-upgrade and upgrade process on Red Hat Enterprise Linux を参照してください。

16. システムで、カーネル (**eth**) が使用する接頭辞に基づいた名前で、複数の Network Interface Card (NIC) が使用されていないことを確認します。RHEL 9 へのインプレースアップグレードの前に、別の命名スキームに移行する方法の手順については、How to perform an in-place

upgrade to RHEL 8 when using kernel NIC names on RHEL 7 を参照してください。命名スキームを移行するプロセスは、RHEL 7 から RHEL 8 へのアップグレードと、RHEL 8 から RHEL 9 へのアップグレードの両方で同じになります。

17. NSS データベースが RHEL 7 以前で作成された場合は、データベースが DBM データベース形式から SQLite に変換されていることを確認します。詳細は、Updating NSS databases from DBM to SQLite を参照してください。

18. RHEL 9 は、RHEL 8 で非推奨となった従来の **network-scripts** パッケージをサポートしていません。アップグレードする前に、カスタムネットワークスクリプトを移動し、既存のカスタムスクリプトを実行する NetworkManager の dispatcher スクリプトを作成します。詳細は、Migrating custom network scripts to NetworkManager dispatcher scripts を参照してください。

19. システム全体のバックアップまたは仮想マシンのスナップショットが存在することを確認してください。これにより、ご利用の環境で、以下の標準の災害復旧手順に従って、システムをアップグレード前と同じ状態に戻せるようになります。たとえば、ReaR (Relax-and-Recover) ユーティリティーを使用できます。詳細は、ReaR documentation および What is Relax and Recover (ReaR) and how can I use it for disaster recovery? を参照してください。または、LVM スナップショット または RAID 分割 を使用することもできます。仮想マシンをアップグレードする場合は、仮想マシン全体のスナップショットを作成できます。

3.2. アップグレードに向けた SATELLITE システムの準備

この手順では、RHEL 9 へのアップグレードに向け、Satellite に登録されているシステムを準備するために必要な手順について説明します。Satellite Server では以下の手順を実行します。

重要

Satellite システムのユーザーは、この手順と アップグレードに向けた RHEL 8 システムの準備 の両方で説明されている準備手順を完了する必要があります。

前提条件

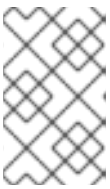
- Satellite Server の管理者権限がある。

手順

1. Satellite は、フルサポートまたはメンテナンスサポートがあるバージョンです。詳細は、「Red Hat Satellite の製品ライフサイクル」を参照してください。

2. RHEL 9 リポジトリーが含まれるサブスクリプションマニフェストを Satellite Server にインポートします。詳細は、Red Hat Satellite の特定のバージョン (version 6.10 など) の『Content Management Guide』の「Managing Subscriptions」を参照してください。

3. RHEL 8.6 および RHEL 9.0 の最新更新と、必要な RHEL 8 リポジトリーおよび RHEL 9 リポジトリーをすべて有効にして同期します。



注記

RHEL 9 リポジトリーの場合は、各リポジトリーのバージョン 9.0 を必ず有効にしてください。RHEL 9 バージョンのリポジトリーのみを有効にした場合は、インプレースアップグレードは行われません。

たとえば、延長アップデートサポート (EUS) サブスクリプションがない Intel アーキテクチャーの場合は、少なくとも以下のリポジトリーを有効にします。

- Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
  rhel-8-for-x86_64-appstream-rpms

  x86_64 8 または 8.6

- Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)
  rhel-8-for-x86_64-baseos-rpms

  x86_64 8 または 8.6

- Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
  rhel-9-for-x86_64-appstream-rpms

  x86_64 9.0

- Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
  rhel-9-for-x86_64-baseos-rpms

  x86_64 9.0

  その他のアーキテクチャーについては、RHEL 8 リポジトリー および RHEL 9 リポジトリー を参照してください。

  詳細は、Red Hat Satellite の特定バージョン (version 6.10 など) の **Content Management Guide** の **Importing Content** の章を参照してください。

4. 必要な RHEL 8 リポジトリーおよび RHEL 9 リポジトリーを含むコンテンツビューに、コンテンツホストをアタッチします。
   詳細は、Red Hat Satellite の特定のバージョン (version 6.10 など) の **Content Management Guide** の **Managing Content Views** を参照してください。

検証

- 最新の RHEL 8 リポジトリーが Satellite Server で有効になっていることを確認します。たとえば、ライブラリーライフサイクル環境のリポジトリーを確認するには、以下を実行します。

```
# hammer repository list --search 'content_label ~ rhel-9' --content-view
<content_view_name> --organization <organization> --lifecycle-environment Library
```

content_view_name をコンテンツビュー名に置き換え、**organization** を組織に置き換えます。

第4章 アップグレード前のレポートの確認

システムのアップグレード可能性を評価するには、**leapp preupgrade** コマンドでアップグレード前のプロセスを開始します。このフェーズでは、**Leapp** ユーティリティーがシステムに関するデータを収集し、アップグレードの可能性を評価し、アップグレード前のレポートを生成します。

アップグレード前のレポートは、**/var/log/leapp/leapp-report.txt** ファイルと Web コンソールの両方で利用できます。レポートは潜在的な問題を要約し、推奨される解決策を提案します。このレポートは、アップグレードを進めることが可能かどうかの判断にも役立ちます。

特定の設定では、**Leapp** により true/false 質問が生成され、続行方法が決まります。質問はすべて、**/var/log/leapp/answerfile** と、**Missing required answers in the answer file** メッセージのプレアップグレードレポートに保存されます。すべての質問に回答しなかった場合には、**Leapp** によりアップグレードが行われません。

アップグレード前のフェーズでアップグレード可能性を評価するには、以下のオプションがあります。

a. 生成された **leapp-report.txt** ファイルのアップグレード前レポートを確認し、コマンドラインインターフェースを使用して、報告された問題を手動で解決します。

b. Web コンソールを使用してレポートを確認し、利用可能な場合は自動修復を適用し、推奨される修復ヒントを使用して残りの問題を修正します。

重要

アップグレード前のフェーズでは、**Leapp** がインプレースアップグレードプロセス全体をシミュレートしたり、すべての RPM パッケージをダウンロードしません。

アップグレード前のレポートを確認すると、インプレースアップグレードプロセスなしで RHEL 8 システムを再デプロイする必要がある場合にも役立ちます。

注記

たとえば、独自のカスタムスクリプトを使用してアップグレード前のレポートを処理し、異なる環境間にある複数のレポートの結果を比較できます。詳細は「Red Hat Enterprise Linux のアップグレード前のレポートワークフローの自動化」を参照してください。

4.1. コマンドラインからのアップグレード可能性の評価

コマンドラインインターフェースを使用して、アップグレード前のフェーズで潜在的なアップグレードの問題を特定します。

前提条件

- アップグレードの準備 に記載されている手順を完了している。

手順

1. RHEL 8 システムで、アップグレード前のフェーズを実行します。

```
# leapp preupgrade --target 9.0
```

- アップグレードに **/etc/yum.repos.d/** ディレクトリーの カスタムリポジトリー を使用する場合は、以下のように選択したリポジトリーを有効にします。

```
# leapp preupgrade --enablerepo <repository_id1> --enablerepo <repository_id2> ...
```

- RHSM を使用しないアップグレード を行う場合や RHUI を使用する場合は、 **--no-rhsm** オプションを追加します。

- Extended Upgrade Support(EUS)、Advanced Update Support(AUS)、または Update Services for SAP Solutions(E4S) のサブスクリプションがある場合は、**--channel channel** オプションを追加します。**channel** を、**eus**、**aus**、または **e4s** などのチャネルに置き換えます。

2. 以下の方法のいずれかを使用して、**Leapp** が必要とする各質問に回答を提供します。

- **leapp answer** コマンドを実行して、応答している質問と、確認した質問を指定します。

```
# leapp answer --section <question_section>.<field_name>=<answer>
```

たとえば、**Are there no VDO devices on the system?**という質問に対して **True** の回答を確定するには、以下のコマンドを実行します。

```
# leapp answer --section check_vdo.no_vdo_devices=True
```

- **/var/log/leapp/answerfile** ファイルを手動で編集し、**#** 記号を削除してファイルの最後の行をコメント解除し、**True** または **False** で回答を確定します。Leapp answerfile を参照してください。

3. **/var/log/leapp/leapp-report.txt** ファイルのレポートを調べて、インプレースアップグレードに進む前に、報告されたすべての問題を手動で解決します。

4.2. WEB コンソールを介したアップグレードの可能性の評価および自動修復の適用

アップグレード前のフェーズで潜在的な問題と、Web コンソールを使用して自動修復を適用する方法を特定します。

前提条件

- アップグレードの準備 に記載されている手順を完了している。

手順

1. **cockpit-leapp** プラグインをインストールします。

```
# dnf install cockpit-leapp
```

2. ブラウザーで Web コンソールに移動し、**root** または **/etc/sudoers** ファイルで設定したユーザーとしてログインします。Web コンソールの詳細は、Managing systems using the RHEL 8 web console を参照してください。

3. RHEL 8 システムで、コマンドラインインターフェイスまたは Web コンソールの端末から、アップグレード前のフェーズを実行します。

```
# leapp preupgrade --target 9.0
```

- アップグレードに **/etc/yum.repos.d/** ディレクトリーの カスタムリポジトリー を使用する場合は、以下のように選択したリポジトリーを有効にします。

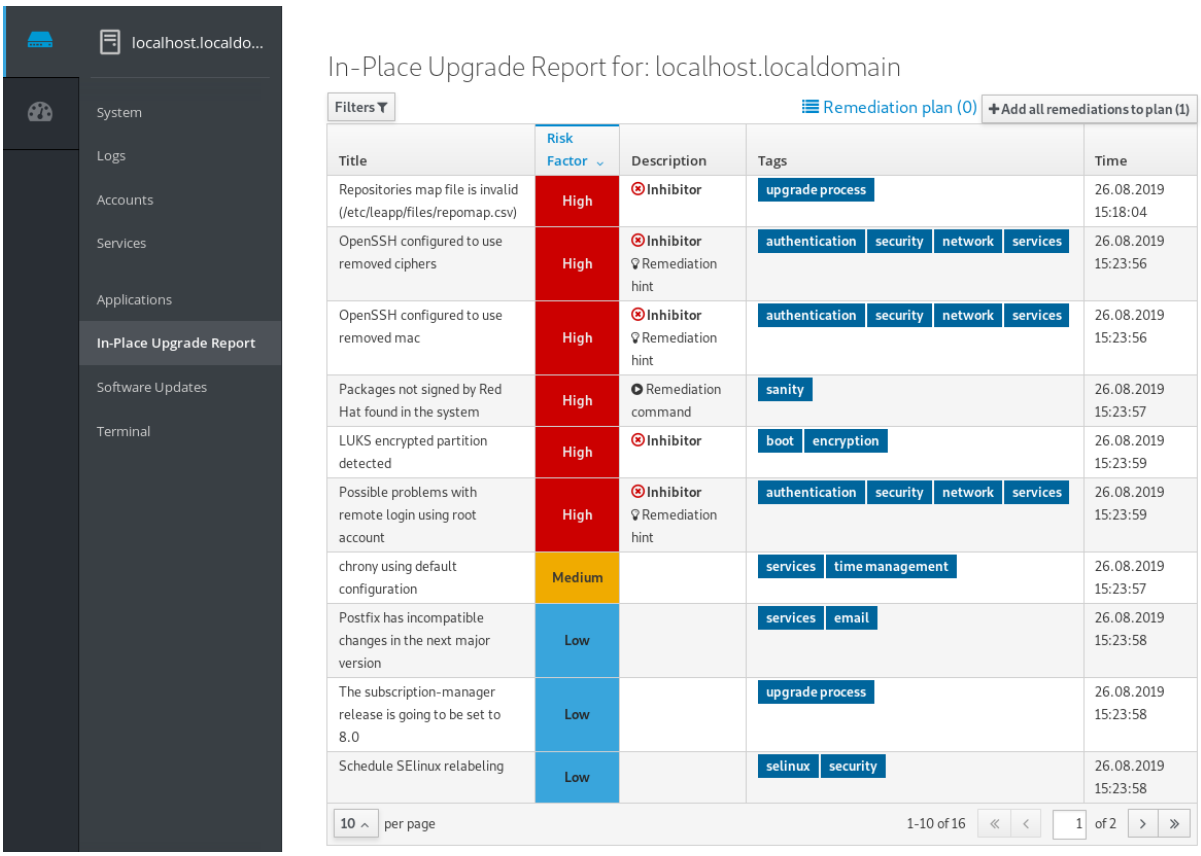
```
# leapp preupgrade --target 9.0 --enablerepo <repository_id1> --enablerepo
<repository_id2> ...
```

- RHSM を使用しないアップグレード を行う場合や RHUI を使用する場合は、 **--no-rhsm** オプションを追加します。

- Extended Upgrade Support(EUS)、Advanced Update Support(AUS)、または Update Services for SAP Solutions(E4S) のサブスクリプションがある場合は、**--channel channel** オプションを追加します。**channel** を、**eus**、**aus**、または **e4s** などのチャネルに置き換えます。

4. Web コンソールで、左側のメニューから **インプレースアップグレードレポート** を選択します。

図4.1 Web コンソールのインプレースアップグレードレポート



レポートの表には、見つかった問題の概要、リスク評価、および修復 (利用可能な場合) が記載されています。

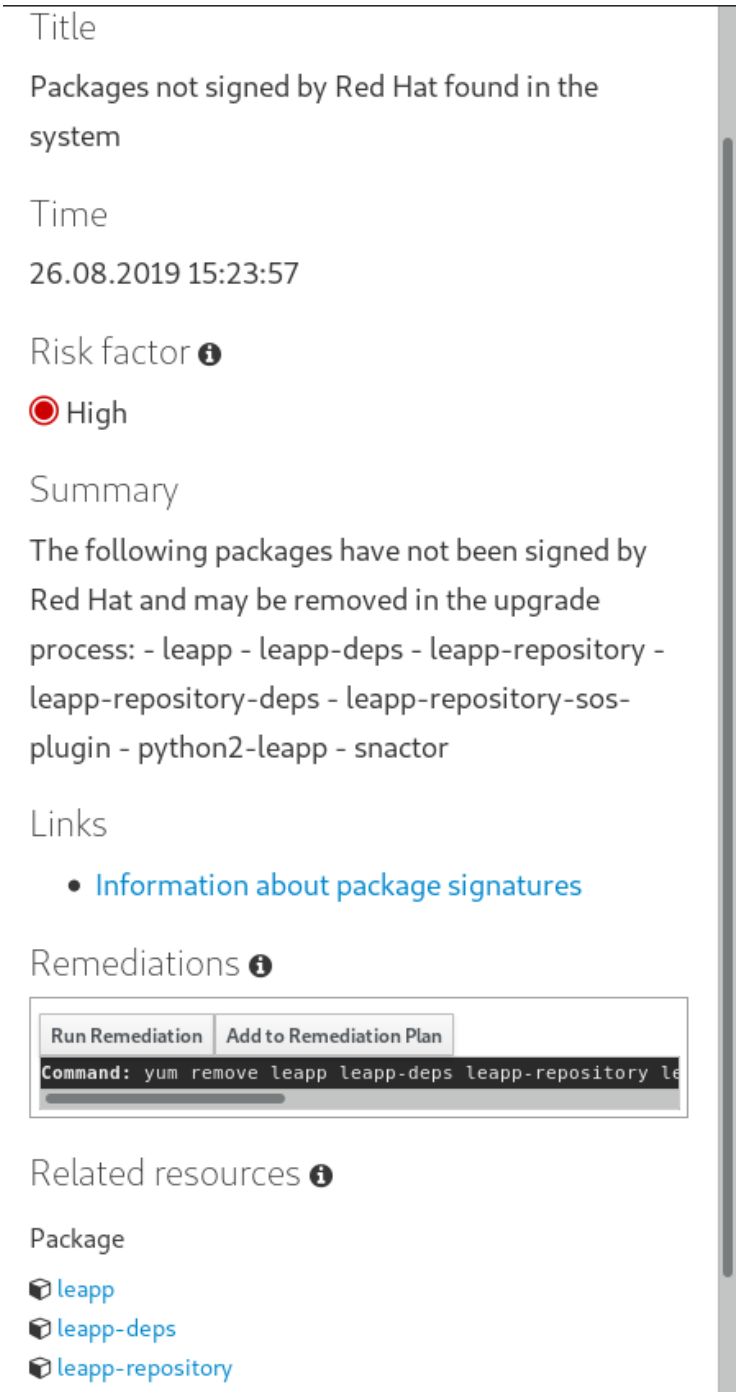
- リスク要因:
  - 高 - システム状態が悪化する可能性が非常に高い
  - 中 - システムとアプリケーションの両方に影響を与える可能性がある
  - 低 - システムに影響はないが、アプリケーションに影響を与える可能性がある

    - 情報 - システムまたはアプリケーションへの影響がないと考えられる情報

- インヒビター - アップグレードプロセスを抑制 (ハードストップ) する。抑制しないと、システムが起動できず、アクセスできず、または機能しなくなる可能性があります。
- 修復 - 報告された問題に対する実行可能な解決策
    - 修復コマンド - Web コンソールから直接実行可能
    - 修復のヒント - 問題を手動で解決する方法の手順

5. レポートの内容を調べます。ヘッダーをクリックして、テーブルを並べ替えることができます。詳細ペインを開くには、選択した行をクリックします。

図4.2 詳細ペイン

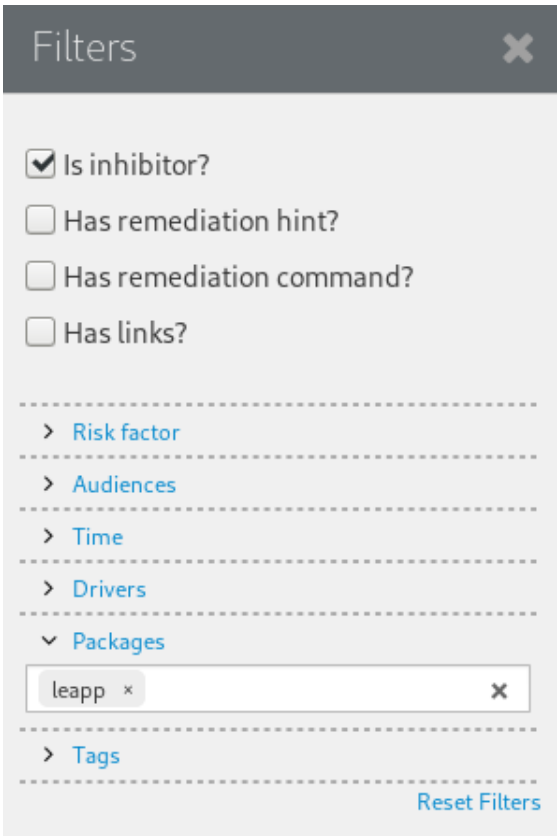


詳細ペインには、次の追加情報が表示されます。

- 問題の概要と、問題を詳細に説明するナレッジベース記事へのリンク
- 修復 - 自動修復 (利用可能な場合) を実行またはスケジュールし、適用時にその結果を確認できます。
- 影響を受けるシステムリソース: パッケージ、リポジトリー、ファイル (構成、データ)、ディスク、ボリューム

6. 必要に応じて、結果をフィルタリングします。レポートの左上隅にある **フィルター** ボタンをクリックし、設定に基づいてフィルターを適用します。フィルターカテゴリーは、相互に関連して適用されます。

図4.3 フィルター



7. 自動修復を適用する問題を選択します。2 つのオプションがあります。

   a. 詳細ペインの **Add to Remediation Plan** ボタンをクリックして、個々の項目を選択します。また、詳細ペインで **Run Remediation** をクリックして、個々の修復を直接実行できます。

   b. レポートの右上隅にある **Add all remediations to plan** ボタンをクリックして、修復が利用可能なすべての項目を選択します。

8. Web コンソールで **Leapp** で必要な質問を確認し、回答します。未回答の質問はそれぞれ、Upgrade Report で **Missing required answers in the answer file** として表示されます。質問に答えるタイトルを選択します。

   a. デフォルトの **True** の応答を確認するには、**Add to Remediation Plan** を選択して修復を後で実行するか、または **Run Remediation** を選択して修復を即座に実行します。

   b. デフォルト以外の回答を選択するには、以下のいずれかを実行します。

      i. **leapp answer** コマンドを実行して、応答している質問と、確認した質問を指定します。

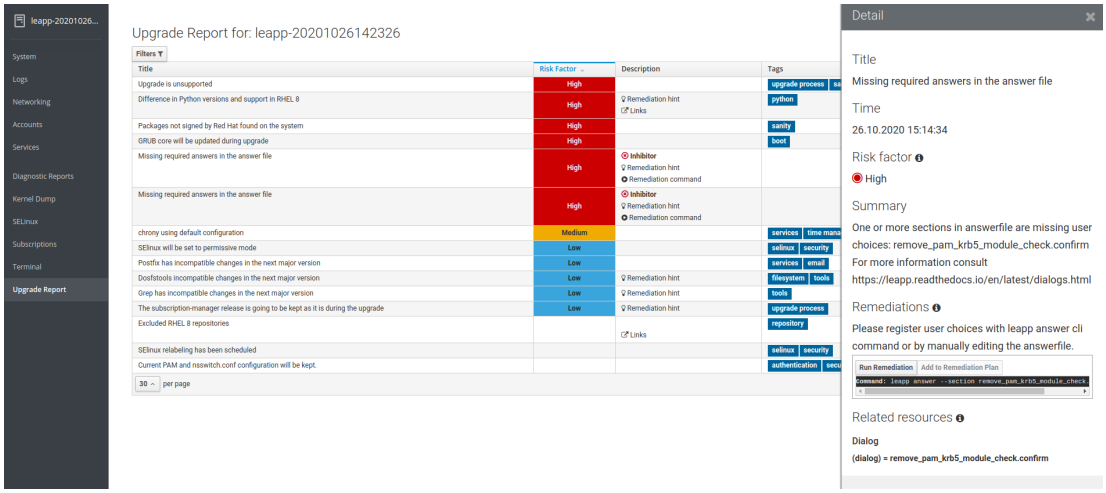
```
# leapp answer --section <question_section>.<field_name>=<answer>
```

たとえば、**Are there no VDO devices on the system?**という質問に対して **True** の回答を確定するには、以下のコマンドを実行します。

```
# leapp answer --section check_vdo.no_vdo_devices=True
```

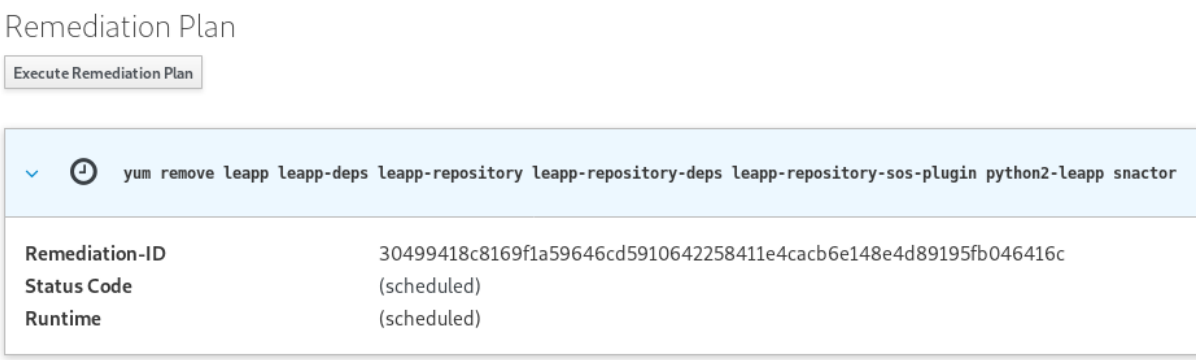
ii. **/var/log/leapp/answerfile** ファイルを手動で編集し、**#** 記号を削除してファイルの最後の行のコメントを解除し、**True** または **False** で回答を確定します。Leapp answerfile の例 を参照してください。

図4.4 未回答の Leapp 質問がない



9. レポートの右上隅にある **Remediation plan** リンクをクリックして、修復計画を開きます。修復計画には、実行した修復、または予定されている修復の一覧が示されます。

図4.5 修復計画



10. **Execute Remediation Plan** をクリックして、予定されている修復をすべて処理します。修復エントリーごとに次の情報が表示されます。

- 修復の一意の ID
- コマンドの終了ステータス
- 実行された修復の経過時間
- 標準出力
- 標準エラー

11. 選択した修復を実行した後に、**leapp preupgrade** コマンドを使用してアップグレード前のレポートを再生成し、新しいレポートを調べてから、必要に応じて追加の修復手順を実行します。

第5章 RHEL 8 から RHEL 9 へのアップグレードの実行

この手順では、**Leapp** ユーティリティーを使用して、RHEL 8 から RHEL 9 へのアップグレードを実行するために必要な手順を説明します。

前提条件

- フルシステムバックアップを含め、アップグレードの準備 に記載されている手順を完了している。
- アップグレード前のレポートの確認 に記載されている手順を完了し、報告されたすべての問題が解決されている。

手順

1. RHEL 8 システムで、アップグレードプロセスを開始します。

```
# leapp upgrade --target 9.0
```

   - アップグレードに **/etc/yum.repos.d/** ディレクトリーの カスタムリポジトリー を使用する場合は、以下のように選択したリポジトリーを有効にします。

```
# leapp upgrade --target 9.0 --enablerepo <repository_id1> --enablerepo
<repository_id2> ...
```

   - RHSM を使用しないアップグレード を行う場合や RHUI を使用する場合は、**--no-rhsm** オプションを追加します。
   - Extended Upgrade Support(EUS)、Advanced Update Support(AUS)、または Update Services for SAP Solutions(E4S) のサブスクリプションがある場合は、**--channel channel** オプションを追加します。**channel** を、**eus**、**aus**、または **e4s** などの **leapp preupgrade** コマンドで使用した値に置き換えます。**leapp preupgrade** および **leapp upgrade** コマンドの両方で、**--channel** オプションで同じ値を使用する必要があります。

2. アップグレードプロセスの開始時に、**Leapp** は、アップグレード前のレポートの確認 で説明されているアップグレード前のフェーズを実行します。
   - システムをアップグレードできる場合は、**Leapp** が必要なデータをダウンロードし、アップグレード用の RPM トランザクションを作成します。
   - システムで、信頼できるアップグレードの設定要因が満たされていない場合は、**Leapp** がアップグレードプロセスを中止し、問題を説明する記録と、推奨される解決策を **/var/log/leapp/leapp-report.txt** ファイルに出力します。詳細は、Troubleshooting を参照してください。

3. システムを手動で再起動します。

```
# reboot
```

   このフェーズでは、システムは RHEL 9 ベースの初期 RAM ディスクイメージ initramfs で起動します。**Leapp** は、すべてのパッケージをアップグレードして、自動的に RHEL 9 システムを再起動します。

   または、**--reboot** オプションを指定して **leapp upgrade** コマンドを実行し、この手動の手順を省略することもできます。

失敗した場合は、「トラブルシューティング」で説明されているようにログおよび既知の問題を調べます。

4. RHEL 9 システムにログインし、RHEL 9 システムのアップグレード後の状態の確認 で説明されているように状態を確認します。

5. アップグレード後のタスクの実行 で説明されているように、アップグレード後のタスクを完了します。特に、セキュリティポリシーを再評価して再適用します。

第6章 RHEL 9 システムのアップグレード後の状態の確認

この手順は、RHEL 9 へのインプレースアップグレード後に実行することが推奨される検証手順を紹介します。

前提条件

- RHEL 8 から RHEL 9 へのアップグレードの実行 に記載の手順に従ってシステムをアップグレードし、RHEL 9 にログインできる。

手順

アップグレードが完了したら、システムが必要な状態になっていることを確認します。少なくとも以下の確認を行います。

- 現在の OS バージョンが RHEL 9 であることを確認します。

```
# cat /etc/redhat-release
Red Hat Enterprise Linux release 9.0 (Plow)
```

- オペレーティングシステムのカーネルバージョンを確認します。

```
# uname -r
5.14.0-70.10.1.el9_0.x86_64
```

.el9 は重要であるため、このバージョンは 5.14.0 よりも前のバージョンにはならないことに注意してください。

- Red Hat Subscription Manager を使用している場合:
  - 正しい製品がインストールされていることを確認します。

```
# subscription-manager list --installed
+-------------------------------------------+
    Installed Product Status
+-------------------------------------------+
Product Name: Red Hat Enterprise Linux for x86_64
Product ID:   479
Version:      9.0
Arch:         x86_64
Status:       Subscribed
```

  - アップグレード直後にリリースバージョンが 9.0 に設定されていることを確認します。

```
# subscription-manager release
Release: 9.0
```

- ネットワークサービスが機能していることを確認します。たとえば、SSH を使用してサーバーに接続します。
- アプリケーションのアップグレード後のステータスを確認します。場合によっては、移行や設定を手動で変更しないといけない場合があります。たとえば、データベースを移行するには、Configuring and using database servers の手順に従ってください。

第7章 アップグレード後のタスクの実行

この手順では、RHEL 9 へのインプレースアップグレード後に実行が推奨される主要タスクを紹介します。

前提条件

- RHEL 8 から RHEL 9 へのアップグレードの実行 に記載の手順に従ってシステムをアップグレードし、RHEL 9 にログインできる。
- RHEL 9 システムのアップグレード後の状態の確認 で説明されている手順に従って、インプレースアップグレードのステータスを確認している。

手順

アップグレードが完了したら、以下のタスクを実行します。

1. **/etc/dnf/dnf.conf** 設定ファイルの除外リストから残りの **Leapp** パッケージを削除します。これには、アップグレードエクステンション開発用のツールである **snactor** パッケージが含まれます。インプレースアップグレード中に、**Leapp** ユーティリティーでインストールされた **Leapp** パッケージが exclude リストに自動的に追加され、重要なファイルが削除または更新されないようにします。インプレースアップグレード後、システムから削除する前に、これらの **Leapp** パッケージを exclude 一覧から削除する必要があります。
   - exclude 一覧からパッケージを手動で削除するには、**/etc/dnf/dnf.conf** 設定ファイルを編集し、除外一覧から必要な **Leapp** パッケージを削除します。
   - exclude 一覧からすべてのパッケージを削除するには、次のコマンドを実行します。

     ```
     # dnf config-manager --save --setopt exclude=''
     ```

2. 残りの **Leapp** パッケージを含め、残りの RHEL 8 パッケージを削除します。
   a. 残りの RHEL 8 パッケージを見つけます。

      ```
      # rpm -qa | grep -e '\.el[78]' | grep -vE '^(gpg-pubkey|libmodulemd|katello-ca-consumer)' | sort
      ```

   b. 古いカーネルパッケージを含む残りの RHEL 8 パッケージを RHEL 9 システムから削除します。
   c. 残りの **Leapp** 依存関係パッケージを削除します。

      ```
      # dnf remove leapp-deps-el9 leapp-repository-deps-el9
      ```

3. セキュリティポリシーを再評価して再適用します。具体的には、SELinux モードを Enforcing に変更します。詳細は、セキュリティーポリシーの適用 を参照してください。

第8章 セキュリティーポリシーの適用

インプレースアップグレードプロセス中に、SELinux ポリシーを Permissive モードに切り替える必要があります。さらに、セキュリティープロファイルには、メジャーリリース間の変更が含まれる可能性があります。このセクションでは、アップグレードされた RHEL システムを保護する際の説明と、セキュリティー関連コンポーネントのアップグレード前の手順の詳細について説明します。

8.1. SELINUX モードの ENFORCING への変更

Leapp ユーティリティーは、インプレースアップグレードプロセス時に SELinux モードを Permissive に設定します。システムが正常にアップグレードされたら、手動で SELinux モードを Enforcing に変更する必要があります。

前提条件

- システムがアップグレードされ、RHEL 9 システムのアップグレード後の状態の確認 で説明されている検証手順を実行している。

手順

1. **ausearch** ユーティリティーなどを使用して、SELinux 拒否がないことを確認します。

```
# ausearch -m AVC,USER_AVC -ts boot
```

前述の手順では、最も一般的なシナリオのみが扱われることに注意してください。考えられる SELinux 拒否をすべて確認するには、完全な手順を説明する Using SELinux の Identifying SELinux denials セクションを参照してください。

2. 任意のテキストエディターで **/etc/selinux/config** ファイルを開きます。以下に例を示します。

```
# vi /etc/selinux/config
```

3. **SELINUX=enforcing** オプションを設定します。

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

4. 変更を保存して、システムを再起動します。

```
# reboot
```

検証

1. システムの再起動後に、**getenforce** コマンドが **Enforcing** を返すことを確認します。

```
$ getenforce
Enforcing
```

関連情報

- SELinux 関連の問題のトラブルシューティング
- SELinux のステータスおよびモードの変更

8.2. システム全体の暗号化ポリシー

システム全体の暗号化ポリシーは、コア暗号化サブシステムを構成するシステムコンポーネントで、TLS、IPSec、SSH、DNSSec、および Kerberos の各プロトコルに対応します。

インプレースアップグレードプロセスでは、RHEL 8 で使用した暗号化ポリシーが保持されます。たとえば、RHEL 8 で **DEFAULT** 暗号化ポリシーを使用した場合、RHEL 9 にアップグレードしたシステムでは **DEFAULT** も使用します。事前定義されたポリシーの特定の設定は異なり、RHEL 9 暗号化ポリシーには、より厳密でより安全なデフォルト値が含まれていることに注意してください。たとえば、RHEL 9 **DEFAULT** 暗号化ポリシーは署名の SHA-1 の使用を制限し、**LEGACY** ポリシーは 2048 ビット未満の DH および RSA 暗号を許可しなくなりました。詳細は、Security hardening の Strong crypto defaults セクションを参照してください。カスタム暗号化ポリシーは、インプレースアップグレード全体で保持されます。

現在のシステム全体の暗号化ポリシーを表示または変更するには、update-crypto-policies tool ツールを使用します。

```
$ update-crypto-policies --show
DEFAULT
```

たとえば、以下のコマンドは、システム全体の暗号化ポリシーレベルを **FUTURE** に切り替えます。これで、近い将来の攻撃に耐えられるはずです。

```
# update-crypto-policies --set FUTURE
Setting system policy to FUTURE
```

シナリオで既存またはサードパーティーの暗号署名を検証するために SHA-1 を使用する必要がある場合は、次のコマンドを入力して有効にできます。

```
# update-crypto-policies --set DEFAULT:SHA1
```

または、システム全体の暗号化ポリシーを **LEGACY** ポリシーに切り替えることもできます。ただし、**LEGACY** は、安全ではない他の多くのアルゴリズムも有効にします。

警告

SHA サブポリシーを有効にすると、システムがデフォルトの RHEL 9 設定よりも脆弱になります。**LEGACY** ポリシーへの切り替えはセキュリティーレベルがさらに低くなるため、使用に際して注意が必要です。

システム全体の暗号化ポリシーをカスタマイズすることもできます。詳細は、Customizing system-wide cryptographic policies with policy modifiers および Creating and setting a custom system-wide cryptographic policy を参照してください。カスタム暗号化ポリシーを使用する場合は、暗号化とコンピューターハードウェアの進歩によってもたらされる脅威を軽減するために、ポリシーを確認および更新することを検討してください。

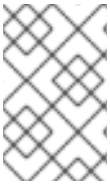
関連情報

- システム全体の暗号化ポリシーの使用
- **update-crypto-policies(8)** の man ページ。

8.3. セキュリティーベースラインが強化されたシステムのアップグレード

正常に RHEL 9 へアップグレードした後に、システムを完全に強化するには、OpenSCAP スイートが提供する自動修復を使用できます。OpenSCAP 修復は、PCI-DSS、OSPP、または ACSC Essential Eight などのセキュリティーベースラインに、お使いのシステムを合わせます。設定コンプライアンスに関する推奨事項は、セキュリティーオファリングが進化したため、RHEL のメジャーバージョン間で異なります。

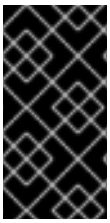
強化された RHEL 8 システムをアップグレードする場合、**Leapp** ツールは完全な強化を保持する直接的な手段を **提供しません**。コンポーネント設定の変更によっては、アップグレード中に RHEL 9 の推奨環境とは異なる場合があります。



注記

RHEL 8 および RHEL 9 のスキャンに同じ SCAP コンテンツを使用することはできません。システムのコンプライアンスが Red Hat Satellite や Red Hat Insights などのツールで管理されている場合は、管理プラットフォームを更新します。

自動修復の代わりに、OpenSCAP で生成されたレポートに従って、手動で変更を行うことができます。コンプライアンスレポートの生成に関する情報は、Scanning the system for security compliance and vulnerabilities を参照してください。



重要

自動修復は、デフォルト設定の RHEL システムで対応しています。アップグレード後にシステム設定が変更されたため、自動修復を実行しても、システムが必要なセキュリティープロファイルに完全に準拠しない場合があります。一部の要件を手動で修正する必要がある場合があります。

以下の手順の例では、PCI-DSS プロファイルに従ってシステム設定を強化します。

前提条件

- RHEL 9 システムに、**scap-security-guide** パッケージがインストールされている。

手順

1. 適切なセキュリティーコンプライアンスデータストリームの **.xml** ファイルを見つけます。

```
$ ls /usr/share/xml/scap/ssg/content/
...
```

```
ssg-rhel9-ds.xml
...
```

詳細は、Viewing compliance profiles のセクションを参照してください。

2. 適切なデータストリームから選択したプロファイルに従って、システムを修正します。

```
# oscap xccdf eval --profile pci-dss --remediate /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
```

--profile 引数の **pci-dss** 値は、システムを強化するプロファイルの ID に置き換えることができます。RHEL 9 でサポートされるプロファイルの完全なリストについては、SCAP security profiles supported in RHEL を参照してください。

警告

--remediate オプションを有効にしてシステム評価を実行した場合、慎重に行わないと、システムが機能不全に陥る場合があります。Red Hat は、セキュリティーを強化した修復で加えられた変更を元に戻す自動手段は提供していません。修復は、デフォルト設定の RHEL システムで対応しています。インストール後にシステムが変更した場合は、修復を実行しても、必要なセキュリティープロファイルに準拠しない場合があります。

3. システムを再起動します。

```
# reboot
```

検証

1. システムがプロファイルに準拠していることを確認し、結果を HTML ファイルに保存します。

```
$ oscap xccdf eval --report pcidss_report.html --profile pci-dss
/usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
```

関連情報

- **scap-security-guide(8)** および **oscap(8)** の man ページ
- セキュリティーコンプライアンスおよび脆弱性スキャンの開始
- Red Hat Insights Security Policy
- Red Hat Satellite Security Policy

8.4. USBGUARD ポリシーの確認

USBGuard ソフトウェアフレームワークを使用すると、カーネルの USB デバイス認証機能に基づいて、許可されているデバイスおよび禁止されているデバイスのリストを使用して、侵入型 USB デバイスからシステムを保護できます。

前提条件

- アップグレードの前に、シナリオの要件を反映した USB デバイス用のルールセットを作成している。
- **usbguard** サービスが RHEL 9 システムにインストールされ、実行されている。

手順

1. **/etc/usbguard/** ディレクトリーに保存されている *.conf ファイルをバックアップします。
2. **usbguard generate-policy** を使用して、新しいポリシーファイルを生成します。このコマンドは、現在存在するUSBデバイスのルールのみを生成することに注意してください。
3. 新たに生成されたルールを、以前のポリシーのルールと比較します。
   a. 新しいポリシーを生成したときに存在したデバイスのルールと、同じデバイスのアップグレード前のルールに違いがあることが確認された場合、後で挿入される可能性のあるデバイスについても、元のルールを相応に修正する必要があります。
   b. 新規生成されたルールとアップグレード前のルールに違いがない場合は、RHEL8 で作成されたポリシーファイルを変更せずに使用できます。

関連情報

- Protecting systems against intrusive USB devices .

8.5. FAPOLICYD データベースの更新

fapolicyd ソフトウェアフレームワークは、ユーザー定義のポリシーに基づいてアプリケーションの実行を制御します。

まれに、**fapolicyd** 信頼データベース形式で問題が発生する場合があります。データベースを再構築するには、以下を実行します。

1. サービスを停止します。

```
# systemctl stop fapolicyd
```

2. データベースを削除します。

```
# fapolicyd-cli –delete-db
```

3. サービスを起動します。

```
# systemctl start fapolicyd
```

カスタム信頼ファイルを信頼データベースに追加した場合は、**fapolicyd-cli -f update <FILE>** コマンドを使用して個別に更新するか、**fapolicyd-cli -f update** を使用してまとめて更新します。変更を適用するには、**fapolicyd-cli –update** コマンドを使用するか、**fapolicyd** サービスを再起動します。

また、カスタムバイナリーには、新しい RHEL バージョン用の再構築が必要になる場合があります。このような更新は、fapolicyd データベースを更新する前に行ってください。

関連情報

- fapolicyd を使用したアプリケーションの拒否および許可

8.6. DBM から SQLITE への NSS データベースの更新

多くのアプリケーションでは、**NSS_DEFAULT_DB_TYPE** 環境変数をシステムの **sql** に設定すると、NSS データベース形式が DBM から SQLite に自動的に変換されます。**certutil** ツールを使用すると、すべてのデータベースが変換されていることを確認できます。

注記

RHEL 9 にアップグレードする前に、DBM 形式に保存されている NSS データベースを変換します。つまり、RHEL 9 にアップグレードする RHEL システム (6、7、および 8) で以下の手順を実行します。

前提条件

- **nss-utils** パッケージがインストールされている。

手順

1. **NSS_DEFAULT_DB_TYPE** を、システム上で **sql** に設定します。

```
# export NSS_DEFAULT_DB_TYPE=sql
```

2. すべてのディレクトリーで変換コマンドを使用する[1] これには、以下のように DBM 形式の NSS データベースファイルが含まれます。

```
# certutil -K -X -d /etc/ipsec.d/
```

データベースファイルがパスワードで保護されている場合は、**-f** オプションの値としてパスワードまたはパスワードファイルへのパスを指定する必要があります。以下に例を示します。

```
# certutil -K -X -f /etc/ipsec.d/nsspassword -d /etc/ipsec.d/
```

関連情報

- **certutil(1)** man ページ。

8.7. BERKELEY DB 形式から GDBM への CYRUS SASL データベースの移行

RHEL 9 の **cyrus-sasl** パッケージは **libdb** 依存関係なしでビルドされ、**sasldb** プラグインは Berkeley DB の代わりに GDBM データベース形式を使用します。

前提条件

- **cyrus-sasl-lib** パッケージがシステムにインストールされている。

手順

- 古い Berkeley DB 形式で保存されている既存のSimple Authentication and Security Layer (SASL) データベースを移行するには、**cyrusbdb2current** を使用します。以下の構文を使用します。

```
# cyrusbdb2current <sasldb_path> <new_path>
```

関連情報

- **cyrusbdb2current(1)** man page

[1] RHEL には、**/etc/pki/nssdb** ディレクトリーにシステム全体の NSS データベースが含まれています。その他の場所は、使用するアプリケーションによって異なります。たとえば、Libreswan はデータベースを **/etc/ipsec.d/** ディレクトリーに保存し、Firefox は / **home/<username>/.mozilla/firefox/** ディレクトリーを使用します。

第9章 トラブルシューティング

RHEL 8 から RHEL 9 へのアップグレードのトラブルシューティングには、以下のヒントを参照してください。

9.1. トラブルシューティングのリソース

以下のトラブルシューティングリソースを参照してください。

コンソールの出力

デフォルトでは、**Leapp** ユーティリティーにより、エラーおよび重要なログレベルメッセージのみがコンソールに出力されます。ログレベルを変更するには、**leapp upgrade** コマンドで **--verbose** オプションまたは **--debug** オプションを使用します。

- verbose モードでは、**Leapp** により情報、警告、エラー、および重要なメッセージが出力されます。
- debug モードでは、**Leapp** によりデバッグ、情報、警告、エラー、および重要なメッセージを出力します。

ログ

- **/var/log/leapp/leapp-upgrade.log** ファイルには、initramfs フェーズで見つかった問題が記載されます。
- **/var/log/leapp/dnf-debugdata/** ディレクトリーには、トランザクションのデバッグデータが含まれます。このディレクトリーは、**leapp upgrade** コマンドに **--debug** オプションを使用して実行した場合に限り表示されます。
- **/var/log/leapp/answerfile** には、**Leapp** による回答が必要な質問が含まれています。
- **journalctl** ユーティリティーでは、すべてのログが出力されます。

レポート

- **/var/log/leapp/leapp-report.txt** ファイルには、アップグレード前のフェーズで見つかった問題が記載されます。レポートは、Web コンソールでも利用できます。Web コンソールを介したアップグレードの可能性の評価および自動修復の適用 を参照してください。
- **/var/log/leapp/leapp-report.json** ファイルには、マシンが判読可能な形式でアップグレード前のフェーズで見つかった問題が記載され、カスタムスクリプトを使用してレポートを処理することができます。詳細は「Red Hat Enterprise Linux のアップグレード前のレポートワークフローの自動化」を参照してください。

9.2. トラブルシューティングのヒント

以下のトラブルシューティングのヒントを参照してください。

アップグレード前のフェーズ

- アップグレードの計画 に記載されている条件をすべて満たしていることを確認します。
- アップグレードの準備 に記載されているすべての手順を実行してください。たとえば、システムで、カーネル (**eth**) が使用する接頭辞に基づいた名前を持つ NIC (Network Interface Card) を複数使用しないようにします。

- **/var/log/leapp/answerfile** ファイルで、**Leapp** に必要な質問をすべて回答している。回答が見つからない場合は、**Leapp** によりアップグレードが行われません。以下に例を示します。
  - Are there no VDO devices on the system?
- アップグレード前のレポートで特定されたすべての問題は、**/var/log/leapp/leapp-report.txt** にあることを確認してください。これを行うには、Web コンソールを介したアップグレードの可能性の評価および自動修復の適用 で説明されているように、Web コンソールを使用することもできます。

例9.1 Leapp answerfile

以下は、編集されていない **/var/log/leapp/answerfile** ファイルの例です。

```
[check_vdo]
# Title:              None
# Reason:             Confirmation
# ========================== check_vdo.no_vdo_devices
==========================
# Label:              Are there no VDO devices on the system?
# Description:        Enter True if there are no VDO devices on the system and False continue the
upgrade. If the system has no VDO devices, then it is safe to continue the upgrade. If there are
VDO devices they must all be converted to LVM management before the upgrade can proceed.
# Reason:             Based on installed packages it is possible that VDO devices exist on the
system.  All VDO devices must be converted to being managed by LVM before the upgrade
occurs. Because the 'vdo' package is not installed, Leapp cannot determine whether any VDO
devices exist that have not yet been converted.  If the devices are not converted and the upgrade
proceeds the data on unconverted VDO devices will be inaccessible. If you have any doubts you
should choose to install the 'vdo' package and re-run the upgrade process to check for
unconverted VDO devices. If you are certain that the system has no VDO devices or that all VDO
devices have been converted to LVM management you may opt to allow the upgrade to proceed.
# Type:               bool
# Default:            None
# Available choices: True/False
# Unanswered question. Uncomment the following line with your answer
# no_vdo_devices =
```

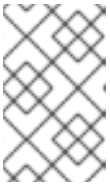
Label フィールドは、回答が必要な質問を指定します。この例では、質問は **Are there no VDO devices on the system?** になります。

この質問に回答するには、最後の行をコメント解除し、回答として **True** または **False** を入力します。この例では、選択した回答は **True** です。

```
[check_vdo]
...
# Available choices: True/False
# Unanswered question. Uncomment the following line with your answer
no_vdo_devices = True
```

ダウンロードフェーズ

- RPM パッケージのダウンロード中に問題が発生した場合は、**/var/log/leapp/dnf-debugdata/** ディレクトリーにあるトランザクションデバッグデータを調べてください。



注記

/var/log/leapp/dnf-debugdata/ ディレクトリーが空であるか、トランザクションのデバッグデータが生成されていない場合は存在しません。これは、必要なリポジトリーが利用できない場合に発生する可能性があります。

Initramfs フェーズ

- このフェーズでは、潜在的な失敗により Dracut シェルにリダイレクトされます。ジャーナルを確認してください。

```
# journalctl
```

あるいは、**reboot** コマンドを実行して、Dracut シェルからシステムを再起動し、**/var/log/leapp/leapp-upgrade.log** ファイルを確認します。

アップグレード後のフェーズ

- システムが正常にアップグレードされたように見えても、古い RHEL 8 カーネルで起動した場合は、システムを再起動して、GRUB でデフォルトエントリーのカーネルバージョンを確認してください。
- RHEL 9 システムのアップグレード後の状態の確認 で推奨される手順を必ず行ってください。
- SELinux を Enforcing モードに切り替えてから、アプリケーションやサービスが停止したり、適切に動作しなかったりした場合は、**ausearch**、**journalctl**、**dmesg** のいずれかのユーティリティーで、サービスの拒否を検索します。

```
# ausearch -m AVC,USER_AVC -ts boot
# journalctl -t setroubleshoot
# dmesg | grep -i -e selinux -e type=1400
```

最も一般的な問題は、ラベルが間違っていることにより発生します。詳細は、Troubleshooting problems related to SELinux を参照してください。

9.3. 既知の問題

以下は、RHEL 8 から RHEL 9 にアップグレードする際に発生する可能性のある既知の問題です。

- 現在、ネットワークチーミングは、Network Manager を無効にするかインストールしていない場合にインプレースアップグレードを実行すると動作しません。
- HTTP プロキシーを使用する場合は、Red Hat Subscription Manager がこのようなプロキシーを使用するように設定するか、**--proxy <hostname>** オプションで **subscription-manager** コマンドを実行する必要があります。そうでない場合は、**subscription-manager** コマンドの実行に失敗します。設定変更の代わりに **--proxy** オプションを使用する場合は、**Leapp** がプロキシーを検出できないため、アップグレードプロセスが失敗します。この問題が発生しないようにするには、「Red Hat Subscription Management に HTTP プロキシを設定する」 の説明に従って **rhsm.conf** ファイルを手動で編集します。(BZ#1689294)
- RHEL 8 システムが、Red Hat が提供しているにもかかわらず RHEL 9 で利用できないデバイスドライバーを使用している場合、**Leapp** はアップグレードを行いません。ただし、RHEL 8 システムが、**Leapp** が **/etc/leapp/files/device_driver_deprecation_data.json** ファイルにデータ

を持たないサードパーティーのデバイスドライバーを使用している場合、**Leapp** はそのようなドライバーを検出せず、アップグレードを続行します。したがって、アップグレード後にシステムが起動しない場合があります。

- お使いのシステムに (Red Hat が署名していない) サードパーティーパッケージの名前が、Red Hat が提供するパッケージの名前と同じ場合は、インプレースアップグレードに失敗します。この問題を回避するには、アップグレードする前に、以下のいずれかのオプションを選択します。

  a. サードパーティーパッケージの削除

  b. サードパーティーパッケージを、Red Hat が提供するパッケージに置き換えます。

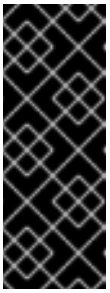
- RHEL 8 では、VDO マネージャーまたは論理ボリュームマネージャー (LVM) を使用して、Virtual Data Optimizer (VDO) ボリュームを管理できます。RHEL 9 では、LVM を使用して VDO ボリュームのみを管理できます。アップグレード後も VDO が管理するボリュームを引き続き使用するには、このようなボリュームを LVM が管理する VDO ボリュームにインポートします。

  1. 最新バージョンの VDO および LVM が、RHEL 8 システムにインストールされていることを確認します。

  2. インプレースアップグレードを開始する前に、VDO が管理する VDO ボリュームを LVM が管理する VDO ボリュームにインポートします。

     ```
     # lvm_import_vdo --name <volume_group_name>/<lvm_name>
     /dev/mapper/<vdo_name>
     ```

     volume_group_name を新しいボリュームグループ名に、**lvm_name** を LVM が管理する VDO ボリュームの新しい名前に、**vdo_name** をインポートする VDO が管理するボリュームの名前に置き換えます。



重要

LVM が管理する VDO ボリュームを、VDO 管理の VDO ボリュームにインポートすることはできません。したがって、今後 VDO マネージャーを使用して、これらの VDO ボリュームにアクセスしようとする場合は、インポートの手順を元に戻すことはできません。LVM が管理する VDO ボリュームの詳細は、Deduplicating and compressing logical volumes on RHEL を参照してください。

- RAID (Redundant array of independent disks) を備えたシステムでは、インプレースアップグレードに失敗します。(BZ#1957192)

- **Leapp** ユーティリティーは通常、インプレースアップグレード時に、RHEL 8 と RHEL 9 の間のネットワークインターフェイスコントローラー (NIC) 名を保持します。ただし、ネットワークボンディングを持つシステムなど、一部のシステムでは、RHEL 8 と RHEL 9 の間で NIC 名を更新する必要がある場合があります。これらのシステムで、**LEAPP_NO_NETWORK_RENAMING=1** 環境変数を設定してインプレースアップグレードを実行し、ネットワークが想定どおりに機能していることを確認します。必要に応じて、ネットワーク設定を手動で更新します。(BZ#1919382)

- NFS サーバーで実行している NSFD サービスがあるシステムでは、インプレースアップグレード中に存在しない NFS パーティションが誤って検出され、アップグレードが妨げられる可能性があります。この問題を防ぐには、インプレースアップグレードを実行する前に NFSD サービスを停止します。

```
# systemctl stop /proc/fs/nfsd
```

(BZ#2036069)

- **Leapp** ユーティリティーは、空きディスク領域が十分にないことを誤って検出するため、アップグレードを実行する前にインプレースアップグレードが停止する可能性があります。ftype 属性のない XFS ファイルシステムでフォーマットされたパーティションがシステムに含まれている場合は、**LEAPP_OVL_SIZE** 環境変数のデフォルトサイズを変更して、コンテナー内の指定された不足ディスク容量を最低限考慮することで、この問題を回避することが可能です。エラーメッセージが繰り返されないように、指定された不足ディスク容量よりもデフォルトのサイズを大きくすることをお勧めします。たとえば、**Leapp** ユーティリティーがさらに 400 MB が必要であることを検知すると、デフォルトのサイズを 2048 MB から少なくとも 2500 MB に増やします。

注記

この回避策では、**/var** パーティションに多くの空き領域が必要になる場合があります。

この回避策が問題を解決しない場合、またはシステムに ftype 属性のないこれらのパーティションが含まれていない場合は、Red Hat サポートにお問い合わせください。(BZ#1832730)

- 64 ビット ARM アーキテクチャーのシステムでは、カーネルページサイズが、RHEL8 の 64k から RHEL 9 の 4k に変更されました。その結果、インプレースアップグレード後に、スワップパーティションは自動的にマウントされません。この問題を回避するには、インプレースアップグレードの実行後に、スワップパーティションを手動でマウントし、スワップを再初期化します。

```
# swapon -a --fixpgsz
```

(BZ#2040470)

- インプレースアップグレードでは、RoCE Express アダプターを使用する IBM Z のネットワークが破損します。RHEL 9.0 は、RoCE Express アダプターに予測可能なインターフェース名を使用します。これは、RHEL 8.6 ディストリビューションで利用可能な名前とは異なります。したがって、RHEL 8 から RHEL 9 へのインプレースアップグレードを実行すると、RoCE Express アダプターの既存のネットワーク設定は、現在の RHEL 9 マイナーリリースで破損します。

- インプレースアップグレード時に、「 RHEL 8 から RHEL 9 へのアップグレードの実行 」の手順 3 で手動で再起動した後に問題が発生した場合は、システムが緊急モードになります。-debug オプションを指定せずに leapp upgrade コマンドを実行する場合や、手動再起動後にコマンドラインインターフェースでキーボードで入力すると、システムが緊急モードに誤って入ります。これが発生すると、緊急モードを入力しても、アップグレードが期待どおりに続行されます。システムが緊急モードに入る場合は、解決する問題がなく、アップグレードが続行されていることを確認してください。

```
# ps -e | grep leapp && echo "The upgrade is still running; do nothing and wait"
```

このコマンドは、必要に応じて複数回実行できます。アップグレードが続行されている場合は、追加のアクションを実行する必要はなく、インプレースアップグレードの完了時にアップグレードされたシステムを起動できます。

(BZ#2092005)

- Red Hat Update Infrastructure(RHUI) を使用する Amazon Web Services(AWS)上のオンデマン

ド Pay-As-You-Go(PAYG)インスタンスのインプレースアップグレードは、RHEL High Availability Amazon Machine Image(AMI)でのみ可能です。現在、他の RHEL AMI を使用してアップグレードを実行することはできません。(BZ#2106904)

9.4. サポートの利用

サポートケースを作成するには、製品で **RHEL 7** を選択し、システムの **sosreport** を添付します。

- システムで **sosreport** を生成するには、次のコマンドを実行します。

```
# sosreport
```

ケース ID は空のままにできます。

sosreport を生成する方法は、ナレッジベースのソリューション「Red Hat Enterprise Linux 上での sosreport の役割と取得方法」を参照してください。

カスタマーポータルでサポートケースを作成し、管理する方法は、ナレッジベースのアーティクル「カスタマーポータルでサポートケースを作成および管理する」を参照してください。

第10章 関連情報

以下の説明情報を参照できます。

- Red Hat Enterprise Linux テクノロジーの機能と制限
- Supported in-place upgrade paths for Red Hat Enterprise Linux
- RHEL 9 の採用における考慮事項
- Customizing your Red Hat Enterprise Linux in-place upgrade
- Red Hat Enterprise Linux のアップグレード前のレポートワークフローの自動化
- Using configuration management systems to automate parts of the Leapp pre-upgrade and upgrade process on Red Hat Enterprise Linux
- 非接続アップグレードのための RHEL の Leapp ユーティリティーメタデータのインプレースアップグレード
- RHEL 7 から RHEL 8 へのアップグレード
- How to convert from CentOS or Oracle Linux to RHEL
- Red Hat Insights ドキュメント

付録A RHEL 8 リポジトリー

アップグレードの前に、Preparing a RHEL 8 system for the upgrade の手順 4 で説明されているように、適切なリポジトリーが有効になっていることを確認します。

アップグレード時に Red Hat Subscription Manager を使用する予定がある場合には、**subscription-manager repos --enable repository_id** コマンドを使用して、アップグレードの前に以下のリポジトリーを **有効にする必要があります**。

表A.1 RHEL 8 リポジトリー

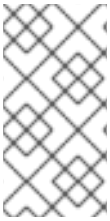
アーキテクチャー	リポジトリー	リポジトリー ID
64 ビット Intel および AMD	Base	**rhel-8-for-x86_64-baseos-rpms**
	Appstream	**rhel-8-for-x86_64-appstream-rpms**
64 ビット ARM	Base	**rhel-8-for-aarch64-baseos-rpms**
	Extras	**rhel-8-for-aarch64-appstream-rpms**
IBM POWER (リトルエンディアン)	Base	**rhel-8-for-ppc64le-baseos-rpms**
	Appstream	**rhel-8-for-ppc64le-appstream-rpmss**
IBM Z	Base	**rhel-8-for-s390x-baseos-rpms**
	Appstream	**rhel-8-for-s390x-appstream-rpms**

次のリポジトリーは、アップグレード前に **subscription-manager repos --enable repository_id** コマンドを使用して **有効にできます**。

表A.2 自発的な RHEL 8 リポジトリー

アーキテクチャー	リポジトリー	リポジトリー ID
64 ビット Intel および AMD	CodeReady Linux Builder	**codeready-builder-for-rhel-8-x86_64-rpms**
	Supplementary	**rhel-8-for-x86_64-supplementary-rpms**
64 ビット ARM	CodeReady Linux Builder	**codeready-builder-for-rhel-8-aarch64-rpms**
	Supplementary	**rhel-8-for-aarch64-supplementary-rpms**
IBM POWER (リトルエンディアン)	CodeReady Linux Builder	**codeready-builder-for-rhel-8-ppc64le-rpms**

アーキテクチャー	リポジトリー	リポジトリー ID
	Supplementary	**rhel-8-for-ppc64le-supplementary-rpms**
IBM Z	CodeReady Linux Builder	**codeready-builder-for-rhel-8-s390x-rpms**
	Supplementary	**rhel-8-for-s390x-supplementary-rpms**



注記

インプレースアップグレードの前に、RHEL 8 CodeReady Linux Builder または RHEL 8 Supplementary リポジトリーを有効にすると、**Leapp** は RHEL 8 CodeReady Linux Builder または RHEL 8 Supplementary リポジトリーをそれぞれ有効にします。詳細は、Package manifest を参照してください。

カスタムリポジトリーを使用する場合は、「Configuring custom repositories」の指示に従って、カスタムリポジトリーを有効にします。

付録B RHEL 9 のリポジトリー

システムが、Red Hat Subscription Manager (RHSM) を使用して Red Hat コンテンツ配信ネットワーク (CDN) に登録されている場合は、インプレースアップグレード時に RHEL 9 リポジトリーが自動的に有効になります。ただし、RHSM を使用して Red Hat Satellite に登録したシステムでは、アップグレード前のレポートを実行する前に、RHEL 8 と RHEL 9 の両方のリポジトリーを手動で有効化して同期する必要があります。

注記

各リポジトリーのバージョン 9.0 を必ず有効にしてください。RHEL 9 バージョンのリポジトリーのみを有効にした場合は、インプレースアップグレードは行われません。

アップグレード時に Red Hat Satellite を使用する予定の場合には、Satellite Web UI または **hammer repository-set enable** コマンドおよび **hammer product synchronize** コマンドを使用して、アップグレード前に少なくとも以下の RHEL 9 リポジトリーを **有効にして同期する必要** があります。

表B.1 RHEL 9 のリポジトリー

アーキテクチャー	リポジトリー	リポジトリー ID	リポジトリー名	Release version
64 ビット Intel および AMD	BaseOS	**rhel-9-for-x86_64-baseos-rpms**	Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)	x86_64 9.0
	Appstream	**rhel-9-for-x86_64-appstream-rpms**	Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)	x86_64 9.0
64 ビット ARM	BaseOS	**rhel-9-for-aarch64-baseos-rpms**	Red Hat Enterprise Linux 9 for ARM 64 - BaseOS (RPMs)	aarch64 9.0
	Appstream	**rhel-9-for-aarch64-appstream-rpms**	Red Hat Enterprise Linux 9 for ARM 64 - AppStream (RPMs)	aarch64 9.0
IBM Power (リトルエンディアン)	BaseOS	**rhel-9-for-ppc64le-baseos-rpms**	Red Hat Enterprise Linux 9 for Power, little endian - BaseOS (RPMs)	ppc64le 9.0

アーキテクチャー	リポジトリー	リポジトリー ID	リポジトリー名	Release version
	Appstream	**rhel-9-for-ppc64le-appstream-rpms**	Red Hat Enterprise Linux 9 for Power, little endian - AppStream (RPMs)	ppc64le 9.0
IBM Z	BaseOS	**rhel-9-for-s390x-baseos-rpms**	Red Hat Enterprise Linux 9 for IBM z Systems - BaseOS (RPMs)	s390x 9.0
	Appstream	**rhel-9-for-s390x-appstream-rpms**	Red Hat Enterprise Linux 9 for IBM z Systems - AppStream (RPMs)	s390x 9.0