Red Hat と AWS のソリューションを使用したコンテナの開発とデプロイ

Red Hat と AWS でコンテナの実践を加速する

クラウド・テクノロジーの進歩、マイクロサービス、および市場の破壊的革新の収束により、デジタル・トランスフォーメーションに対する切迫感が生まれています。企業はより迅速に行動し、よりアジャイルで効率的でありたいと考えています。アプリケーションはこれらのデジタルイニシアチブの成功の鍵を握っていますが、時間が非常に重要です。アプリケーションをわずか数週間でデプロイすることが求められるのは、今や一般的です。

アプリケーションコンテナは、アプリケーション提供の高速化に役立ちます。重要なのがコンテナイメージです。これは、実行可能コードを含む変更不可能な静的ファイルであり、IT インフラストラクチャ上で分離されたプロセスとして実行できます。イメージは、コンテナ化されたプラットフォームでソフトウェアプログラムを実行するために必要なシステムライブラリ、システムツール、およびその他のプラットフォーム設定で構成されます。イメージは、そのホストマシンのオペレーティングカーネルを共有します。

開発環境とデプロイ環境が同一であるため、コンテナはプロセスに一貫性をもたらします。その結果、各段階でアプリケーションの標準的な不変の表現に対する検証が行われるため、リリースサイクルが大幅に短縮されます。

Linux® コンテナは、分離したプロセスを作成する名前空間と共に Linux のプロセス分離を利用するため、ほとんどのデジタル・トランスフォーメーションのニーズに最適です。Red Hat® Enterprise Linux は、エンタープライズ環境での Linux コンテナの実行における業界標準であり、コンテナの構築方法に関する多数の選択肢を備えています。開発者は、これらの選択肢を使用してコンテナのスピンアップを簡単に実行できますが、デプロイとオーケストレーションも管理する必要があります。

この資料では、開発者がアプリケーションをコンテナ化して提供する上で、Red Hat のコンテナツールと Amazon Web Services (AWS) の開発者用ツールがどのように役立つかを説明します。

Linux アプリケーションコンテナのメリット

Linux アプリケーションコンテナは、IT がビジネス要求に迅速に対応するために役立ちます。開発者は、アプリケーションをモジュール式のマイクロサービスに分割し、Linux コンテナとしてデプロイします。コンテナ化することで、これらのアプリケーション・コンポーネントはフットプリントが小さくなり、デプロイが迅速化されます。

Linux コンテナアプリケーションは、ハードウェアとオペレーティングシステムから抽象化されているため、アーキテクチャ間での可搬性が高くなります。コンテナを使用すると、開発者のラップトップからプロダクションのクラスタに至るまで、さまざまな環境でのアプリケーションのデプロイと実行が容易になります。また、アプリケーションのアップグレードとロールバックを、アプリケーションが実行されているサーバーから分離します。コンテナは、運用の単純化、モジュール性、柔軟性を提供し、IT 管理者の負担を軽減します。コンテナの管理に Kubernetes や Red Hat OpenShift などのオーケストレーション・システムを使用すると、モジュール性と柔軟性のメリットが大きくなります。

Linux コンテナは、インフラストラクチャのコストを削減できます。コンテナは小さく、軽量なので、ベアメタルシステムだけでなく、同じオペレーティングシステムを実行するパブリッククラウド、プライベートクラウド、ハイブリッドクラウド、およびマルチクラウド環境間でも、簡単に移行できます。コンテナが基盤となるオペレーティングシステム (OS) と互換性がある場合、コンテナによって新しいアプリケーションの開発を加速し、既存のアプリケーションを最適化し、すべてのアプリケーションを接続することができます。

仮想マシン (VM) と比較すると、コンテナは次の用途に最適です。

- クラウドネイティブ・アプリケーションを構築する
- マイクロサービスをパッケージ化する
- DevOps または継続的インテグレーション/継続的デリバリー (CI/CD) プラクティスを導入する
- 同じオペレーティングシステムを共有する多様な IT フットプリント全体でスケーラブルな IT プロジェクトを移行する

対照的に、VM は単一のコンテナよりも多くの操作を実行できるため、従来はモノリシックなワークロードに使用されていました。しかし、その豊かな機能は、OS、アプリケーション、ライブラリに依存するため、VM の可搬性は低下します。

コンテナと比較すると、VM は次の用途に最適です。

- 従来型、レガシー、モノリシックのワークロードを収容する
- リスクの高い開発サイクルを分離する
- インフラストラクチャのリソース (ネットワーク、サーバー、データなど) をプロビジョニングする
- 別の OS 内で異なる OS を実行する (Linux で Unix を実行するなど)

1 つの環境で実行される VM とオペレーティングシステムの数を減らすと、ライセンス要件が大幅に軽減されます。Linux コンテナは、これらのコストをさらに削減できます。Red Hat と AWS のツールとプラットフォームによって、クラウドネイティブのコンテナ化アプリケーションを構築し、DevOps および CI/CD プラクティスを導入してアジリティを高め、市場投入時間を短縮することができます。

Red Hat Enterprise Linux でのアプリケーションコンテナ開発

Red Hat ソリューションを使用すると、Red Hat Enterprise Linux 上にアプリケーションコンテナを構築できます。Red Hat Enterprise Linux 8 には、フルサポート付きの軽量でオープンな標準ベースのコンテナツールキットが含まれています。これらのツールは AWS ソフトウェア開発キット (SDK) およびクラウド開発キット (CDK) ともリンクしているため、エンタープライズ・アプリケーションの構築とテストを行うことができます。

コンテナを開発したら、それをデプロイする必要があります。Kubernetes でオーケストレーションをしている場合、複雑性、スケーラビリティ、サービス検出などに関連する問題が発生し、進行が妨げられることがあります。Red Hat OpenShift は、Kubernetes およびコンテナのデプロイと管理に関連する潜在的な課題を抽象化する設計であるため、ユーザーはコードの作成に集中できます。Red Hat Enterprise Linux の場合、小規模で機敏なツールセットを使用してコンテナを容易に開発できます。

Red Hat Enterprise Linux 8 とコンテナ開発

Red Hat Enterprise Linux は少ないリポジトリと、多くの開発者用ツールでコンテナ開発を単純化します。これらのツールに加えて、Red Hat は、組織独自のイメージの基盤として機能するベースイメージを提供します。これらのベースイメージの一部は、Node.js、PHP、Java™、Python を使用して構築されるビジネスアプリケーションから、ロギング、データ収集、認証などのインフラストラクチャ機能まで、さまざまなユースケースを対象としています。

Red Hat Enterprise Linux 8 では、Buildah、Podman、runC、Skopeo などのコマンドラインツール、ユニバーサル・ベース・イメージ (UBI)、Red Hat Quay のリポジトリ、および Supplemental リポジトリの新機能が導入されています。これらはすべて、コンテナ開発の複雑性を軽減します。

以下の図は、Red Hat Enterprise Linux 8 コンテナのアーキテクチャを示しています。

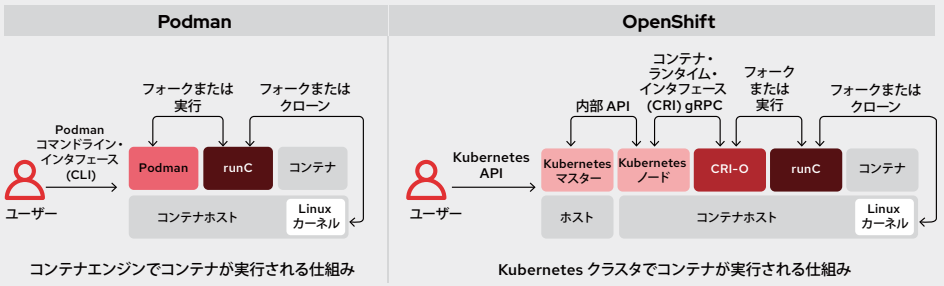

図 1. Red Hat Enterprise Linux 8 コンテナがコンテナエンジンまたは Kubernetes クラスタで実行される仕組み

Buildah

Buildah を使用すると、デーモンや Docker をインストールせずにコンテナをスクラッチから構築できます。スクラッチビルドに適したユースケースは、Java アプリケーションのステージングイメージまたはプロダクションイメージに対する開発イメージの検討です。開発中は、Java アプリケーションのコンテナイメージには、Java コンパイラーと Maven の他にもツールが必要になることがあります。しかし、プロダクションで必要になるのは、Java ランタイムとパッケージのみである場合があります。

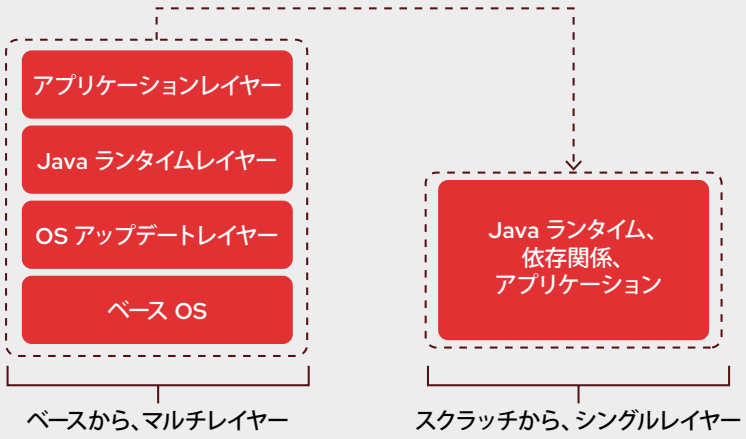

図 2. ベースイメージから、またはスクラッチから構築される Java アプリケーションレイヤー

Buildah はデーモンレスであるため、ホストに特別なインフラストラクチャを設定することなく、コンテナ内で簡単に実行できます。

Buildah は、Open Container Initiative (OCI) 準拠の Linux コンテナイメージを作成または変更するために必要な基本要件のみを提供するため、既存のビルドパイプラインへの統合が容易になります。たとえば、Buildah は、パッケージマネージャー (DNF/YUM) を含まないコンテナが最終イメージで必要とされない場合、それらをアセンブルすることができます。Buildah を使用すると、これらのコンテナを簡単に構築でき、一貫したセキュリティを確保できます。しかし、オーバーヘッド (したがってイメージサイズ) を削減し、クラウドネイティブ・アプリケーションに必要なものに合わせてカスタマイズを拡張することもできます。

Buildah の用途として最も強力なのは、イメージ作成のための Bash スクリプトの記述であり、Dockerfile の記述も同様です。

Podman

Podman を使用すると、個別のデーモンなしでコンテナを管理できます。また、Podman は Docker コマンドライン・インタフェース (CLI) と互換性があります。Podman により、イメージレジストリ、コンテナとイメージストレージ、および runC コンテナ・ランタイム・プロセスを介した Linux カーネルとの直接のインタラクションが可能になります。

Podman は root 権限がなくても実行でき、また、root 機能を提供するデーモンがなくても実行できるため、Podman がイメージを記述できる別の場所が必要です。Podman は、/var/lib/containers を誰でも修正できるようにしたり、セキュリティ上の問題を引き起こす可能性のあるプラクティスが適用されたりしないように、ユーザーのホームディレクトリにあるリポジトリを使用します。これにより、各ユーザーが別々のコンテナとイメージのセットを持ち、互いに干渉することなく同じホストで Podman を同時に使用できるようになります。ユーザーが作業を終了したら、共通のレジストリに変更をプッシュして、他のユーザーとイメージを共有できます。

runC

Buildah と Podman はコンテナを起動する際、OCI ランタイムの runC を使用します。ユーザーは、イメージをビルドして実行するか、runC を使用して Docker 形式のイメージを実行できます。この Go 言語ベースのツールは、ランタイム仕様を読み取り、Linux カーネルを構成し、最終的にコンテナプロセスを作成して開始します。runC の低レベルコードは Docker と共有できますが、runC は Docker プラットフォームのどのコンポーネントにも依存しません。Linux 名前空間、ライブマイグレーションをサポートし、ポータブルなパフォーマンス・プロファイルを備えています。また、SELinux、コントロールグループ (cgroups)、セキュア・コンピューティング・モード (seccomp) などの Linux のセキュリティ機能も完全にサポートします。

Skopeo

Skopeo はイメージを検査し、OCI イメージを保存できる任意の場所に転送します。Skopeo がリリースされる前は、一部のメタデータのみを検査したい場合でも、イメージを検査するにはイメージ全体を取得する必要がありました。Skopeo はすべてのイメージのプロパティを表示するため (レイヤーなど)、ホストにイメージをプルする必要はありません。検査の結果、ある場所から別の場所にコンテナイメージをコピーする必要があることが示された場合は、Skopeo を使用して実行できます。

また、Skopeo を使用すると、リポジトリからイメージを削除し、外部のイメージリポジトリを内部レジストリに同期して、エアギャップされた、または接続されていないネットワークをデプロイできます。リポジトリによって要求された場合、Skopeo は認証用に適切な資格情報と証明書を渡すことができます。

Red Hat Universal Base Image

従来、Linux コンテナは、宛先プラットフォームごとに構築する必要がありました。Red Hat Universal Base Image (UBI) はこの問題に対処しています。UBI は、エンタープライズグレードのベースコンテナイメージで、開発者はその上でアプリケーションを構築し提供することができます。Red Hat Enterprise Linux 8 では、すべての Red Hat ベースイメージが UBI として利用できます。UBI を使用すると、Red Hat サブスクリプションがなくても Red Hat Enterprise Linux に基づいてコンテナイメージをビルドして再配布することができ、ユーザーもサブスクリプションを必要としません。

実証済みの Red Hat Enterprise Linux ベースイメージに基づく UBI は、コンテナで開発されるクラウドネイティブのユースケースや Web アプリケーションのユースケースの基盤となるように設計されており、コミュニティプロジェクトやセルフサポートを希望する人にとっては、CentOS ベースのコンテナイメージの作成という余分な作業が不要になります。

UBI を使用してコンテナ化されたアプリケーションを構築し、それを任意のレジストリサーバーにプッシュし、プロダクションに移行し、他の人と簡単に共有できます。自由に再配布できるため、他のベンダーのプラットフォームにもデプロイできます。この機能は Ubuntu または別の Linux プラットフォームを使用している場合に便利であり、プラットフォームが異なる場合でも Red Hat Enterprise Linux の使用に慣れることができます。

Red Hat Quay のリポジトリ

Red Hat Quay にイメージリポジトリを作成することもできます。これは、コンテナイメージの保存、ビルド、デプロイのためのプライベート・コンテナ・レジストリです。イメージのセキュリティの脆弱性を分析し、セキュリティ上のリスクの緩和に役立つ潜在的な問題を特定します。Red Hat Quay でリポジトリを作成するには、プッシュ (Docker または Podman から) を使用する方法と、Red Hat Quay ユーザー・インタフェース (UI) を使用する方法の 2 つがあります。Red Hat Quay の公開リポジトリにもアクセスできます。

Supplementary リポジトリ

Supplementary リポジトリには、オープンソースの Red Hat Enterprise Linux リポジトリには含まれていないプロプライエタリー・ライセンスのパッケージが含まれています。これらのパッケージは Red Hat Enterprise Linux 8 ではサポートされていませんが、アプリケーションのコンテナ化の過程で他のソフトウェアにアクセスできるという柔軟性を提供します。

Red Hat Enterprise Linux on Amazon Elastic Compute Cloud (EC2)

Red Hat Enterprise Linux on Amazon Elastic Compute Cloud (Amazon EC2) は、Amazon EC2 のコンピューティングと Red Hat Enterprise Linux を組み合わせたものです。Red Hat が Amazon EC2 用の Red Hat Enterprise Linux ベースイメージを維持します。更新は、利用可能になり次第 Red Hat から送信されます。コンピューティング環境とセキュリティの信頼性はそのままで、Red Hat Enterprise Linux 認定アプリケーションのサポート性が維持されます。Red Hat Enterprise Linux を実行する Amazon EC2 は、コンテナ化アプリケーションを含む幅広いアプリケーションをもたらす仮想開発環境と信頼できるプラットフォームを提供します。その分散アーキテクチャを使用して、より柔軟かつアジャイルにアプリケーションをデプロイできます。

Red Hat Enterprise Linux on EC2 では、アプリケーションのコンテナ化に使用できる C++、Java、Rust とともに、中核的なコンテナ言語と見なされる Go などの一般的な言語の AWS SDK にアクセスできます。また、AWS SDK を Red Hat Enterprise Linux に含まれるコンテナの構成要素と組み合わせることもできます。さらに、Red Hat Universal Base Image (UBI) とアプリケーション・ストリームを利用して、必要な場所でコンテナ化アプリケーションを構築、共有し、協業することができます。

お分かりいただけたと思いますが、Red Hat Enterprise Linux 8 でのコンテナ開発は、個々のコンテナとコンテナイメージを開発、構築、および管理するために実行可能な選択肢です。しかし、Red Hat OpenShift® を使用すると、コンテナの開発とデプロイがさらにスケーラブルで効率的になります。

Red Hat OpenShift on AWS：コンテナのデプロイと開発を組み合わせる

クラウドネイティブ・アプリケーションの構築、マイクロサービスのパッケージ化、DevOps や CI/CD プラクティスの使用、または同じオペレーティングシステムを共有する多様な IT フットプリント全体でのスケーラブルな IT プロジェクトの移行において、コンテナは理想的です。オペレーティングシステム、アプリケーション、ライブラリに関連付けられていないため、ポータブルで軽量です。しかし、課題もあります。コンテナを利用する場合は、コンテナのスケーリング、デプロイ、および置き換えの方法、永続ストレージの管理方法、サービス検出の処理方法に対処する必要があります。

また、コンテナランタイムのみを使用している場合は、複雑性が障害になることがあります。運用を拡大するにはコンテナを追加する必要があり、すぐに一連の別のプロセスが発生します。これにより、サーバー上に非常に多くのコンテナが作成されるため、パフォーマンスが低下することがあります。また、コンテナ (ボリューム) から切り離されたストレージを簡単に見失うこともあります。したがって、オーケストレーション・レイヤーがコンテナのデプロイに不可欠です。Kubernetes は最適なオーケストレーターですが、独自のネイティブ Kubernetes をデプロイしようとすると、複雑で問題が生じる可能性があります。Red Hat Quay と Kubernetes を併せて使用すると、完全なアプリケーションを稼働させるために必要なイメージと構成の詳細を管理することができます。しかし、Red Hat OpenShift は Kubernetes の複雑さをほぼすべて解消するため、Kubernetes を可能な限り迅速に起動して実行することができます。

Red Hat OpenShift で利用可能な開発ツール：

- CodeReady Workspaces を使用すると、リモートの開発チームがボタンをクリックするだけで環境をプロビジョニングして共有できるため、起動が速くなり、インタラクションのレイテンシーが低くなります。
- Quarkus フレームワークにより、Kubernetes ネイティブの Java アプリケーションを構築できます。
- Buildpacks と Kaniko のプレビューサポートは、Buildah を介して S2I および Dockerfile ビルドと共に利用できます。
- Helm は、チャートとリリースでの作業を単純化します。

Red Hat OpenShift でコンテナをデプロイする

Red Hat OpenShift は、コンテナ・プラットフォームのデプロイと管理を単純化することを目的として設計されています。統合されたプラットフォーム監視、自動化されたメンテナンス作業とアップグレードが組み込まれた、管理者にとって使いやすいエンタープライズ向け Kubernetes コンテナ・プラットフォームです。

Red Hat OpenShift は「write once, run anywhere (一度書けば、どこでも実行できる)」戦略に対応しており、オンプレミス、パブリッククラウド環境、プライベートクラウド環境、ハイブリッドクラウド環境など、最も意義のある場所でワークロードを実行できます。AWS では、Red Hat OpenShift が別のサービスとして統合およびパッケージ化され、AWS コンソールのリストに表示されるため、OpenShift クラスタをより少ない時間と労力で作成し、一貫して管理することができます。また、オンデマンドの請求、単一の請求書、AWS にサポートを依頼するオプションも利用できます。デフォルト設定またはカスタム AWS 設定で、AWS に Red Hat OpenShift クラスタをデプロイできます。自分でプロビジョニングした AWS インフラストラクチャにクラスタをデプロイすることもできます。付属の AWS CloudFormation テンプレートを必要に応じて変更できます。

Red Hat OpenShift でコンテナを開発する

Red Hat OpenShift には、Source-to-Image フレームワーク (S2I) が含まれています。これにより、アプリケーションコードからコンテナに直接移行できるため、コンテナ開発の複雑性が大幅に軽減します。ユーザーは、アプリケーションのソースコードを入力として受け取るイメージを記述し、アセンブルされたアプリケーションを出力として実行する新しいイメージを生成するだけです。OpenShift で構築する方法は 2 つあり、Dockerfile と S2I を使用します。

ユーザー提供の Dockerfile は、OpenShift ビルダーに含まれる Buildah を呼び出してイメージを作成します。Buildah は、生成されたイメージにタグを付けてプッシュします。あるいは、S2I を使用して、Buildah ベースの Dockerfile ビルドと同じフローに従って Dockerfile を生成することもできます。結果として得られるビルダーイメージには、その実行可能イメージ (ビルドアーティファクトとも呼ばれる) を生成するために必要な特定のインテリジェンスが含まれています。

S2I が示すように、Red Hat OpenShift は、デプロイ用であると同時にコンテナ開発用のプラットフォームでもあります。開発者は、既存の AWS インフラストラクチャの仕様に合わせてコーディングするのではなく、Kubernetes プラットフォーム上で構築できます。さらに、Red Hat OpenShift では、さまざまな方法でコンテナを使用してアプリケーション開発に取り組むことができるため、さまざまな状況に適した方法を使用できます。これは、Kubernetes に不慣れでコーディングのみを行う開発者のニーズに対応するだけでなく、最大限の柔軟性を求める熟練した Kubernetes 開発者のニーズにも対応します。

Red Hat OpenShift は、開発者がすでに使用している言語、データベース、ツールをサポートします。また、AWS が提供するようなアプリケーション開発サービスにも簡単にアクセスできます。

Red Hat OpenShift 上のコンテナ向けの AWS 開発者用ツール

AWS ツールを使用すると、Red Hat OpenShift でのコンテナ開発がさらに容易になります。これらのツールは、Red Hat Enterprise Linux など、Red Hat OpenShift 以外のプラットフォームでも使用できます。

AWS CodeArtifact は、フルマネージドのアーティファクト・リポジトリ・サービスで、ソフトウェア開発プロセスで使用されるソフトウェアパッケージの保存、公開、共有の安全性を、組織の規模を問わず容易に向上できるようにします。

AWS Cloud9 は、ブラウザーベースのシェルでコードを記述、実行、およびデバッグするための統合開発環境 (IDE) で、追加のソフトウェアのインストール、git push の実行、コマンド入力を行うことができます。

AWS Cloud Development Kit は、プログラミング言語の親しみやすさと表現力をアプリケーションのモデリングに使用しており、新たに習得すべきことがほとんどないため、AWS へのオンボーディングを加速します。

Amazon CodeGuru は、パフォーマンスの問題を修正する方法と非効率的なコードの実行にかかる推定コストを視覚化して、それらに関する推奨事項を提供し、開発者が修復の優先順位を付けるために役立ちます。Red Hat OpenShift の既存のソフトウェア開発ワークフローと統合しており、コードの品質を向上させ、アプリケーションの最もコストのかかるコード行を特定するためのインテリジェントな提案を行います。

コンテナの構築、デプロイ、管理を今すぐ始める

アプリケーションコンテナの構築は難しいものではありません。Red Hat のコンテナツールと AWS の開発者用ツールが役立ちます。コンテナのデプロイとオーケストレーション (コンテナ実装の最後のステップ) は、プロセスの中で最も難しい部分になる可能性があります。Red Hat OpenShift は Kubernetes の複雑さをほぼ解消し、コンテナの構築方法に関係なく、大規模にデプロイおよび管理できるようにします。

Red Hat OpenShift の価値は、単なるデプロイメント・ソリューションではないことです。開発者には、このプラットフォームでコンテナを開発し、デプロイするためのさまざまな選択肢があります。AWS では、Red Hat OpenShift のセルフマネージドおよびフルマネージドの両方のオプションを利用できます。Red Hat OpenShift を使用すると、AWS で独自の OpenShift 実装をデプロイして管理できます。Red Hat OpenShift Service on AWS と Red Hat OpenShift Dedicated は、Red Hat Cloud Services に完全に対応しているため、アプリケーションの構築と市場への投入に集中できます。

Red Hat OpenShift のその他の開発ツール：

- Odo は、開発者がその CLI を使用してコードを反復する手段を提供します。Kubernetes と Red Hat OpenShift、標準定義によるツールのオープンモデル、Quarkus を使用した迅速な反復型 Java 開発をサポートします。
- Knative サービングとイベンティングのサーバーレスのサポートにより、開発者は、Strimzi (OpenShift で Apache Kafka を実行するため) と Service Mesh を含むサーバーレスでイベント駆動型のアプリケーションを構築できます。
- Red Hat OpenShift Pipelines の Tekton、GitHub アクション用の OpenShift プラグイン、Jenkins、GitLab ランナーのサポートにより、継続的な統合機能が提供されます。

Red Hat と AWS は、コンテナの開発方法やデプロイ方法に関係なく、そのプロセスを最適化し加速するソリューションを提供し、お客様がビジネスに必要なアプリケーションをより迅速に作成してデプロイできるようにします。

Red Hat と AWS のパートナーシップが、Red Hat Enterprise Linux による先進的なインフラストラクチャの構築をどのように支援するのかについてご覧ください。または、以下の詳細をご覧ください。

- Red Hat Enterprise Linux のドキュメント
- Podman
- AWS の開発者用ツール

Red Hat について

エンタープライズ・オープンソース・ソフトウェア・ソリューションのプロバイダーとして世界をリードする Red Hat は、コミュニティとの協業により高い信頼性と性能を備える Linux、ハイブリッドクラウド、コンテナ、および Kubernetes テクノロジーを提供しています。Red Hat は、クラウドネイティブ・アプリケーションの開発、既存および新規 IT アプリケーションの統合、複雑な環境の自動化および運用管理を支援します。受賞歴のあるサポート、トレーニング、コンサルティングサービスを提供する Red Hat は、フォーチュン 500 企業に信頼されるアドバイザーであり、オープンな技術革新によるメリットをあらゆる業界に提供します。Red Hat は企業、パートナー、およびコミュニティのグローバルネットワークの中核として、企業の成長と変革を支え、デジタル化が進む将来に備える支援を提供しています。

アジア太平洋
+65 6490 4200
apac@redhat.com

オーストラリア
1 800 733 428

インド
+91 22 3987 8888

インドネシア
001 803 440 224

日本
03 4590 7472

韓国
080 708 0880

マレーシア
1 800 812 678

ニュージーランド
0800 450 503

シンガポール
800 448 1430

中国
800 810 2100

香港
800 901 222

台湾
0800 666 052

fb.com/RedHatJapan
twitter.com/RedHatJapan
linkedin.com/company/red-hat

jp.redhat.com
#F30136_1021