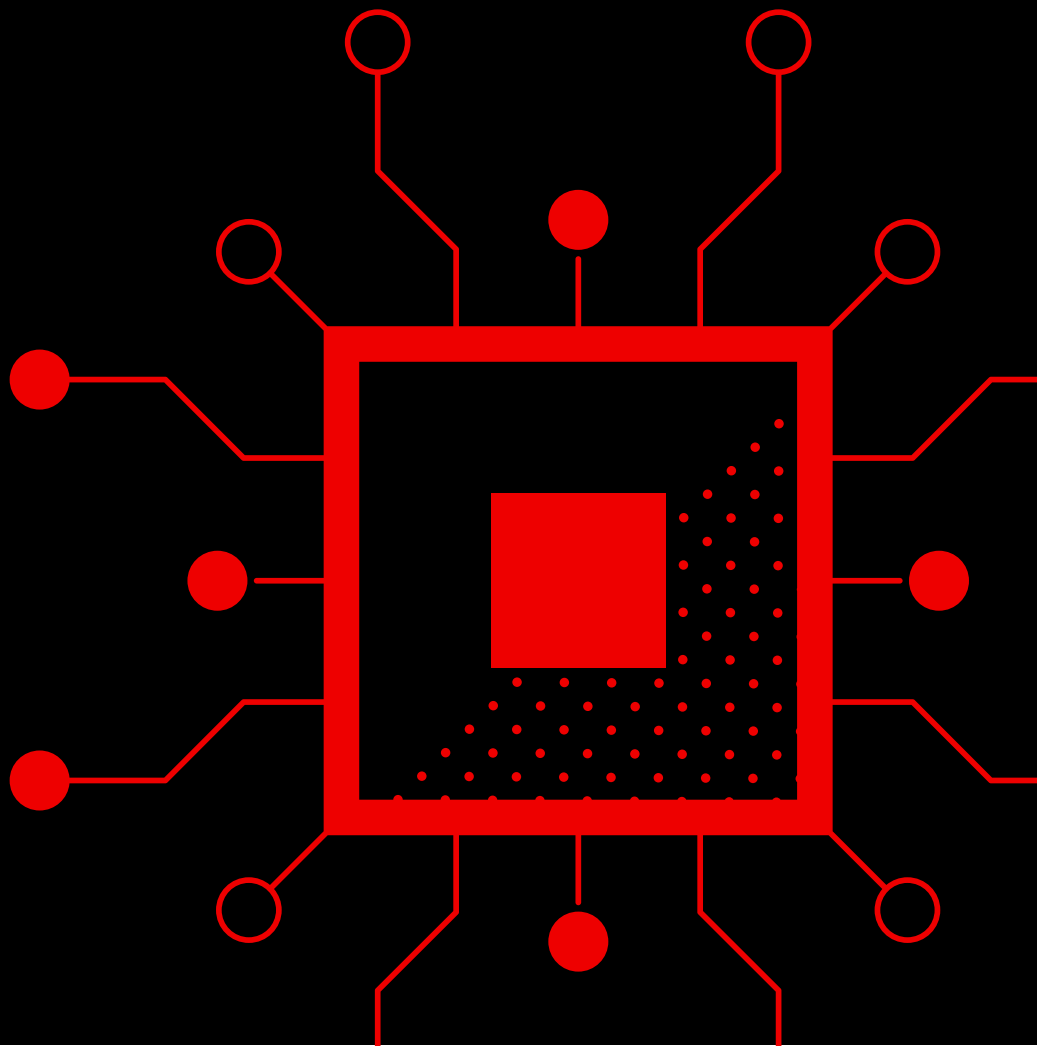
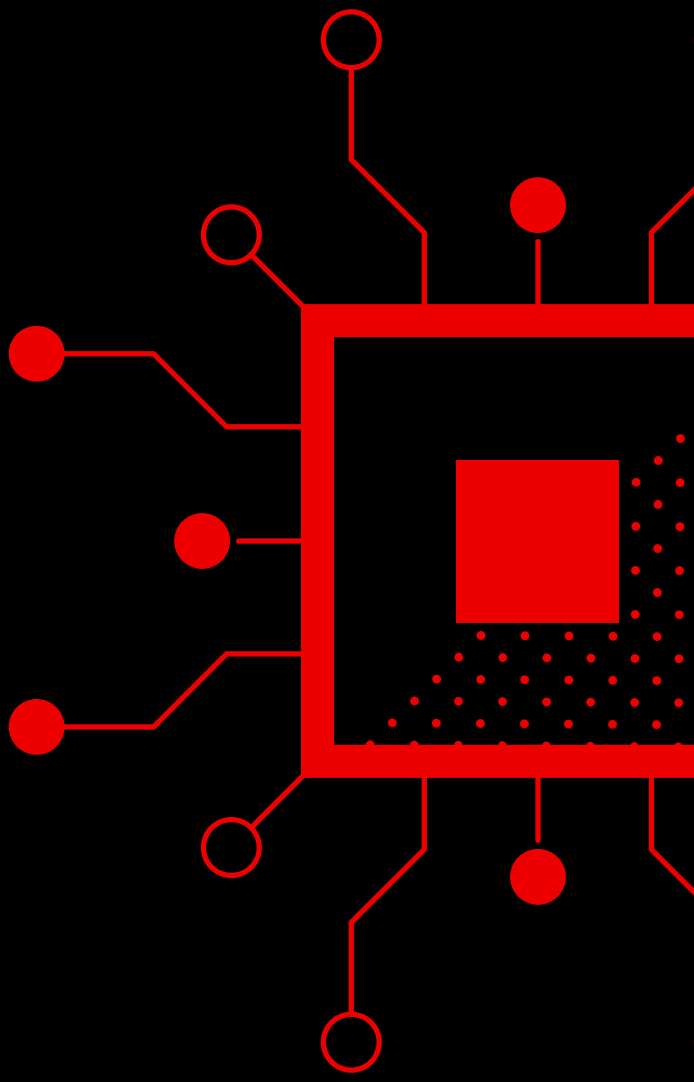
開発者向けのオープンソース AI

AI 対応アプリケーションの計画と構築に関するガイド



Red Hat

目次



オープンソースと AI：変革をもたらす組み合わせ

人工知能 (AI) と機械学習運用 (MLOps) は、アプリケーションと開発プロセスを変革しています。

AI モデルに基づく革新的なソリューションは、コンテンツの作成、意思決定プロセスの強化、ユーザーエクスペリエンスのパーソナライズに新たな可能性をもたらします。先進的な MLOps ワークフローは、本番環境への AI モデルの統合、デプロイ、管理を効率化し、信頼性とパフォーマンスを確保します。AI と MLOps を組み合わせると、動的なアプリケーション、効率的なワークフロー、開発サイクルの短縮によって開発者のアジリティが向上し、進化するビジネスニーズに迅速に対応できるようになります。

オープンソースの AI ツールは、柔軟性とカスタマイズの観点から開発チームに大きなメリットをもたらします。これらのツールを使用することで、開発者はインテリジェントなアプリケーションを修正したり適応させたりしてソリューションをカスタマイズし、ビジネスニーズを満たすことができます。オープンソース・プロジェクトは、さまざまなユーザーやコントリビューターから成る幅広いコミュニティ内のコラボレーションを促進し、主要な AI テクノロジーにおける継続的な改善と新機能の開発をサポートします。この適応性により、組織の要件に応じたカスタマイズが可能なことから、AI ツールは特殊な機能を必要とするプロジェクトに最適な選択肢となります。

この e ブックの対象者

この e ブックは、オープンソースの AI ツール、プラットフォーム、戦略を自身のプロジェクトに適用したいと考えている初心者から中級レベルの開発者やデータサイエンティストを対象としています。実践的な開発の側面に重点を置き、革新的な AI ベースのアプリケーションを作成するためのモデルの選択、統合、改良、デプロイについて説明します。

開発戦略の計画

有意義な成果を達成するには、AI ベースのアプリケーションを開発するための構造化されたアプローチが必要です。そのプロセスの各ステップでは、アプリケーションが意図した目標を達成するとともに確実かつ効率的に動作するよう、慎重な検討が求められます。明確な目標の設定からモデルの選択、倫理的実践の確保に至るまで、以下のステップを通じて、堅牢で信頼性の高い AI ベースのソリューションを作成することができます。

1 **AI の目標を特定する：**まず、AI ベースのアプリケーションによって対処しようとしている問題の概要を明確にします。アプリケーションが実行する必要がある具体的なタスクを特定します。これには、生成、分類、会話、要約、またはコード関連の機能などが該当します。

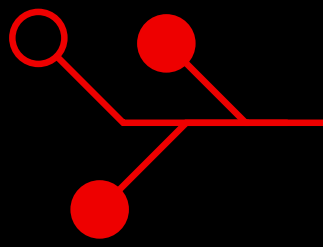
2 **利用可能なデータを評価する：**データの品質と範囲によってモデルの成功が左右されるため、データを徹底的に評価します。**生成 AI** モデルには通常、大規模なトレーニングデータセットが必要ですが、予測 AI モデルはラベル付けされた小規模なデータセットで効果を発揮できます。データに関連性があり、多様で、アプリケーションが遭遇するシナリオに対応するものであることを確認します。歪んだ結果や不正確な結果を避けるために、データ内のギャップやバイアスに対処します。

3 **AI タスクを分析する：**アプリケーションの要件に基づいて AI のアプローチを選択します。生成 AI モデルは、微妙な言語の理解、クリエイティブな出力、データの拡張、リアルタイムのインタラクションに優れており、コンテンツ作成、会話型 AI、テキストの要約に最適です。画像のセグメンテーションや不正検出などのように、タスクが構造化され、出力が明確に定義されている場合は、畳み込みニューラルネットワーク (CNN) やデシジョンツリーといった予測 AI モデルによって、リアルタイム・アプリケーションの推論時間を短縮できます。場合によっては、両方のアプローチを組み合わせると効果的です。たとえば、カスタマーサポートのチャットボットでは、意図分類に予測 AI モデルを使用し、自然言語処理に生成 AI モデルを使用できます。

4 **適切なモデルを選択する：**サードパーティのアプリケーション・プログラミング・インタフェース (API) やマネージドサービスを通じて提供される、すぐに使える AI モデルを利用すれば、高度な AI 機能をアプリケーションに効率的に追加できます。このようなモデルを使用することで、AI や機械学習に関する広範な専門知識がなくても、自然言語処理、画像認識、予測分析などの複雑な機能を統合することができます。サードパーティの API やマネージドサービスは、モデルのトレーニングとメンテナンスのプロセスを単純化するため、ユーザーエンゲージメントとアプリケーションのパフォーマンスを向上させる AI 機能を提供する上で役立ちます。

5 **倫理的コンプライアンスを確保する：**AI ベースのアプリケーションは機密データを扱うことが多く、意思決定に大きな影響を与える可能性があります。アプリケーションのアクションと予測が倫理的に与える影響を考慮してください。予期せぬ結果を回避し、ユーザーとの信頼関係を構築するために、透明性、公平性、説明責任を優先しましょう。

革新的な AI ベースのアプリケーションの構築



AI には、多様な目的に対応するさまざまなテクノロジーが含まれます。その中でも、予測 AI と生成 AI は、インテリジェントなアプリケーションを作成する上で重要なテクノロジーであり、それぞれ異なる機能を備えています。予測 AI は、既存のデータを分析して将来の結果や傾向を予測することに重点を置いています。履歴データを使用してイベント、行動、または状況を予測し、ユーザーが、起こりそうなシナリオに基づいた意思決定を行えるよう支援します。対照的に、生成 AI は既存の情報から学習したパターンに基づいて新しいコンテンツやデータを作成します。トレーニングデータに類似したテキスト、画像、その他のメディアを生成し、コンテンツの作成とパーソナライゼーションを実現する革新的なソリューションを提供します。予測 AI は将来の可能性に関する洞察を提供しますが、生成 AI は学習したパターンから新しい出力を生成します。

予測 AI

AI ベースのアプリケーションの構築は、多くの場合、タスクに適したモデルを選択することから始まります。予測 AI の場合、ニーズに合わせてカスタマイズされた事前トレーニング済みのモデルやアーキテクチャの選択が必要になります。一般的なモデルには、画像分類用の ResNet、オブジェクト検出用の YOLO (You Only Look Once)、異常検出用の Isolation Forest などがあります。モデルを選択する際は、推論速度 (トレーニングされたモデルが新しい入力データに基づいて予測を行う速度) との関連でモデルのサイズと複雑性を考慮することが重要です。AI の実践が確立されている組織の場合は、自社のデータを使用して独自の予測 AI モデルを構築することもできます。OpenCV、scikit-learn、TensorFlow、PyTorch などのオープンソースライブラリとディープラーニング・フレームワークは、内部モデルと外部モデルを AI ベースのアプリケーションに効率的に統合するのに役立ちます。

データの準備とモデルの評価は、次の重要なステップです。データを徹底的に分析することで、その特性をより深く理解し、潜在的な問題に対処できます。さまざまなモデル・アーキテクチャや事前トレーニング済みの重みを試してみると、パフォーマンスと推論速度の最適なバランスを見つけることができます。そのモデルが適切に一般化されていることを確認するには、別のテストデータセットで厳密な検証を行うことが有用です。

開発プロセスにおいて、モデルの選択と検証の後に続くのは改良とデプロイです。サイズ変更や正規化などの手法を使用してデータを前処理することが必要になる場合があります。事前トレーニング済みモデルを使用する場合、特定のデータセットでの**ファインチューニング**が重要です。また、後処理技術によってモデルの出力を改善できる場合もあります。推論応答時間やリソース使用量など、プロダクションでのモデルの

パフォーマンスを監視することで、モデルを効率的に再トレーニングして最適化し、有効性を維持できます。モデルのデータドリフトの監視は、推論中に表示されるデータが、モデルのトレーニングやチューニングに使用されたデータと大幅に異なることがないようにするために役立ちます。

予測 AI モデルは通常、生成 AI モデルよりも推論が速いため、多くの場合、リアルタイム・アプリケーションに最適です。予測 AI と生成 AI を組み合わせることで、より完全なソリューションを作成できます。しかし、そのようなソリューションではモデルのランタイムが長くなり、複雑性が増す可能性があります。

生成 AI

生成 AI に基づくアプリケーションの開発における最初のステップは、適切な**大規模言語モデル (LLM)** の選択です。Bidirectional Encoder Representations from Transformers (BERT)、Text-to-Text Transfer Transformer (T5)、**Granite モデル**など、オープンソースの選択肢は複数あり、それぞれが異なるタスクに対して独自の強みを発揮します。アプリケーションの目的に合った LLM を選択することが重要です。たとえば、Granite-7B-Starter はリスク要因、補償範囲、責任範囲に焦点を当てた保険関連のテキストを要約できるようにファインチューニングすることが可能で、BERT は感情分析に優れています。

アプリケーションに関連するタスクに対する LLM の正確性、流暢性、全体的な効果はさまざまであるため、モデルのパフォーマンスの評価は非常に重要です。さらに、GPT-3 や一部の Granite バリアントなどの高機能モデルでは、高価なグラフィックス・プロセッシング・ユニット (GPU) リソースをはじめ、大量の計算リソースが必要になる場合があるため、こうしたニーズと、利用可能なインフラストラクチャや予算とのバランスを取ることが不可欠です。また、ファインチューニングに十分な高品質のデータを利用できれば、アプリケーションの要件を満たす最適な LLM パフォーマンスを確保できます。

Langchain のようなフレームワークは、アプリケーションへの LLM の統合を単純化するため、コア・アプリケーション・ロジックに集中できるようになります。これらのフレームワークは、メモリーやコンテキストによって LLM ベースのコンポーネントを強化するとともに、プロンプトエンジニアリングとモデルチェーンのためのツールを提供します。

最適な LLM とフレームワークを選択したら、アプリケーションに生成機能を追加できるようになります。このプロセスには、AI が望ましい結果をもたらすように導くための、モデルのパフォーマンスの改良と正確かつ効果的なプロンプトの作成が含まれます。堅牢なフィードバックループの確立は継続的な改善に不可欠であり、これによってモデルが経時的に適応し、その出力が向上するようになります。

LLM の評価および比較方法

1. **プロンプトの生成：**
   さまざまなプロンプトを作成して、クリエイティブな能力と生成能力を評価します。プロンプトを改良して、さまざまなシナリオに対する最適な応答と出力を引き出します。
2. **データを使った実験：**
   アプリケーションに固有のプロプライエタリーデータを使用してモデルをテストします。プロンプトと設定を調整して、特定のタスクに対するモデルのパフォーマンスを最適化します。
3. **パフォーマンスのベンチマーク：**
   実験の結果を使用して、モデルのパフォーマンスを評価し、比較します。

プロンプトは、LLM に望ましい出力を生成するように指示するために有用です。明確で簡潔なプロンプトを作成し、構造化された指示のテンプレートを使用し、チェーニングなどの手法を採用して LLM が複雑なタスクを完了できるように導くことで、モデルの有効性を大幅に向上できます。これらの戦略により、AI モデルは複数ステップのインタラクションでも一貫性のある適切な応答を生成できるようになります。

人間のフィードバックによる強化学習 (RLHF) ループは、LLM のファインチューニングに不可欠です。モデルをデプロイした後、ユーザーとのインタラクションを収集し、そのフィードバックを使用して LLM のパフォーマンスを改良します。この反復的なプロセスにより、モデルは間違いから学習して継続的に改善され、実際のユースケースに適応しながら正確で関連性の高い出力を提供する能力を向上させます。

ファインチューニングは、事前トレーニング済みの LLM を特定のドメインやタスクに合わせてさらにカスタマイズします。タスクに特化した、より小規模なデータセットでモデルをトレーニングすることで、パフォーマンスを向上させ、アプリケーションの要件に合わせて出力をカスタマイズできます。**Hugging Face Transformers** などのツールを使用すると、事前トレーニング済みモデルの知識を活用しながら、目的に合わせて改良することができます。**InstructLab** のモデルアライメント手法は、モデルの出力を組織の価値観やユーザーのニーズに合わせて調整するのに役立ち、正確で状況に応じた適切な応答ができるようにします。

検索拡張生成 (RAG) は、LLM と情報検索システムを組み合わせて、モデルが生成中に外部ソースから関連データにアクセスし、そのデータを組み込むことができるようにします。このアプローチは、出力の事実上の正確性と一貫性を向上させます。LLM の結果を内部データや企業データで補強するときによく使用されます。Langchain に組み込まれた RAG 機能は、特に Granite モデルを使用して正確で状況に適した応答を生成する場合に、このプロセスを効率化します。

エージェントは、定義された環境内で動作して特定の目標を達成する自律システムです。インタラクティブで適応的な動作を組み込むことで、状況の変化に応じて動作コンテキストを動的に修正できます。これにより、複雑なタスクを処理し、リアルタイムの意思決定を行うことができます。このようなエージェントを開発するには、AI モデルの出力に基づいてアクションを計画、実行、評価するマルチコンポーネント・システムを構築する必要があります。リアルタイムの意思決定、外部 API とデータソースの統合など、複雑なタスクをオーケストレーションすることで、システムの運用能力を強化できます。

モデルチェーンは、複数の AI モデルまたはプロセスを 1 つのまとまったワークフローに接続します。各モデルは前のモデルの出力に基づいて構築されます。このアプローチにより、複数ステップのインタラクションを伴う複雑なタスクを処理できるアプリケーションの開発が可能になります。さまざまなモデルの機能を連携させて使用することで、要件に合わせた効率的なシステムを構築できます。

統合された AI によってアプリケーションのワークフローを徹底的に評価することで、ユーザーにとって使いやすく効率的なエクスペリエンスを実現できます。システム全体を厳密にテストすることで、問題や非効率性を特定してそれらに対処し、アプリケーションを改良して機能性と使いやすさを向上することができます。この反復的なプロセスによってパフォーマンスが向上するだけでなく、アプリケーションをユーザーのニーズや期待にさらに近づけることができます。

例 1：カスタマーサポートのチャットボット

- **問題：**サービス部門は、毎日 24 時間、顧客の質問や問題に迅速かつ効率的に対応する必要がある
- **データ：**チャットログ、製品ドキュメント、ナレッジベースの記事
- **モデル：**会話と意図分類に適応し、知識検索に対応する RAG で強化されている Granite-7B-Starter
- **倫理的考慮事項：**データプライバシーの確保、バイアスの軽減、ユーザーへの透明性の提供
- **デプロイと反復：**チャットボットのデプロイ、インタラクションの監視、フィードバックの収集、モデルの再トレーニング、ナレッジベースの定期的なアップデート

サードパーティ API を使用してチャットボットを構築

- **モデルサービス：Quarkus REST Client** 経由の OpenAI ChatGPT API
- **追加の考慮事項：**意図分類とナレッジベースの統合

例 2：保険引受リスク評価

- **問題：**複雑な保険書類を自動的に要約して引受プロセスを効率化する必要がある
- **データ：**保険証券、保険金請求書、医療報告書など、保険書類一式
- **モデル：**リスク要因、補償範囲、責任範囲に焦点を当てた保険関連のテキストを要約できるようにファインチューニングされている Granite-7B-Starter
- **倫理的考慮事項：**正確性の優先、法令遵守の確保、厳格なデータプライバシーの維持
- **デプロイと反復：**引受ワークフローへのモデルの統合、フィードバックの収集、モデルの改良によるリスク評価の向上

サードパーティ API を使用して保険リスクを評価

- **モデルサービス：**Quarkus REST Client 経由の Google Gemini API
- **追加の考慮事項：**ファインチューニングとデータの前処理

AI 向けの高度なツールとテクノロジーの導入

革新的な AI ベースのアプリケーションを開発するには、プロンプトエンジニアリング、ファインチューニング、RAG、エージェントを理解することが不可欠です。これらの手法はそれぞれ、複雑な課題に対処し、AI ベースのアプリケーションのインタラクティブ機能を改善するためのツールと戦略を提供します。このような技術をうまく適用することで、重要なビジネス目標を満たす、よりインテリジェントで応答性に優れた効果的な AI ベースのシステムを作成できます。

Red Hat® OpenShift® AI は **Red Hat OpenShift** をベースとしており、先進的な AI ソリューションのワークロードおよびパフォーマンスに関する要求を満たすと同時に、モデルとアプリケーションの構築、トレーニング、ファインチューニング、デプロイ、および監視のための包括的なプラットフォームを提供します。AI 開発の生産性を向上させるツールと環境が含まれています。

- **Jupyter Notebooks** と **PyTorch** は実験と共同開発を容易にし、プロトタイプからプロダクションへの移行を効率化します。
- **Red Hat OpenShift Pipelines** は**継続的インテグレーション/継続的デプロイメント (CI/CD)** ワークフローを自動化し、スムーズで効率的なモデル提供を実現します。
- **監視および可観測性強化ツール**は、モデルのパフォーマンスと健全性をリアルタイムで追跡し、データのドリフトとバイアスを監視して、プロアクティブな調整とメンテナンスをサポートします。

Red Hat OpenShift AI は、ハイブリッドクラウド環境全体での AI ベースのアプリケーションの開発とデプロイも効率化します。**KServe**、**vLLM**、**Text Generation Inference Server (TGIS)** などのモデルサーバーやランタイムのサポートをはじめとする、強化されたモデル提供機能により、AI モデルを簡単かつ柔軟にデプロイできます。Red Hat OpenShift AI は、モデル提供機能をエッジロケーションまで拡張するため、リソースに制約のある環境でも AI ベースのソリューションを提供できます。ハードウェア・アクセラレーターへのセルフサービスアクセスにより、アプリケーションを迅速に反復して最適化できます。

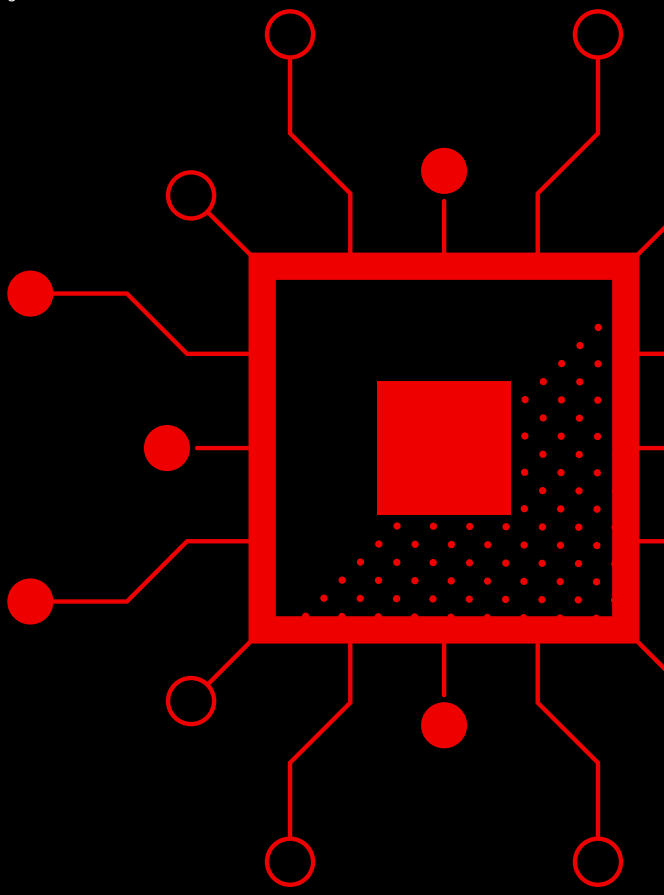
Red Hat Enterprise Linux® AI は、エンタープライズ・アプリケーションを強化する **Granite LLM ファミリー**をシームレスに開発し、テストし、実行するための基盤モデル・プラットフォームです。ハイブリッドクラウド環境間での可搬性が実現し、Red Hat OpenShift AI を使用して AI ワークフローを拡張できるようになります。

そして、**Podman Desktop AI Lab 拡張機能**は、AI ベースのアプリケーションのローカル開発およびテスト向けに最適化されたセットアップを提供するため、本番環境の正確かつ効率的なシミュレーションが可能になります。

開発をスタート

Red Hat OpenShift AI を使用して今すぐAI対応アプリケーションの構築を始めましょう。

Red Hat OpenShift AI は、複雑なワークフローを単純化し、開発サイクルを短縮し、デプロイメントオプションを拡張します。AI モデルの構築とデプロイに必要な時間と労力を削減することで、イノベーションに集中し、より短時間で魅力的なソリューションを作成できるようになります。



Red Hat OpenShift AI を無料で使い始める

無料の開発者サンドボックスで Red Hat OpenShift AI にアクセス

無料で試用する

Red Hat OpenShift AI の使用方法を学ぶ

さまざまな AI ユースケースのインタラクティブなラーニングパスにアクセス

詳細はこちら

1024_KVM

Red Hat