Red Hat OpenShiftで進化するアプリケーションデリバリー

Toshihiro Araki

Specialist Solution Architect

OpenShiftを選ぶべき理由

Trusted

Container engine

トラステッドな
コンテナエンジン

Reduce Risk

リスク = ↓↓

Comprehensive

Application platform

包括的なアプリケーション
プラットフォーム

Improve Productivity

生産性 = ↑↑

Consistent

Across hybrid cloud

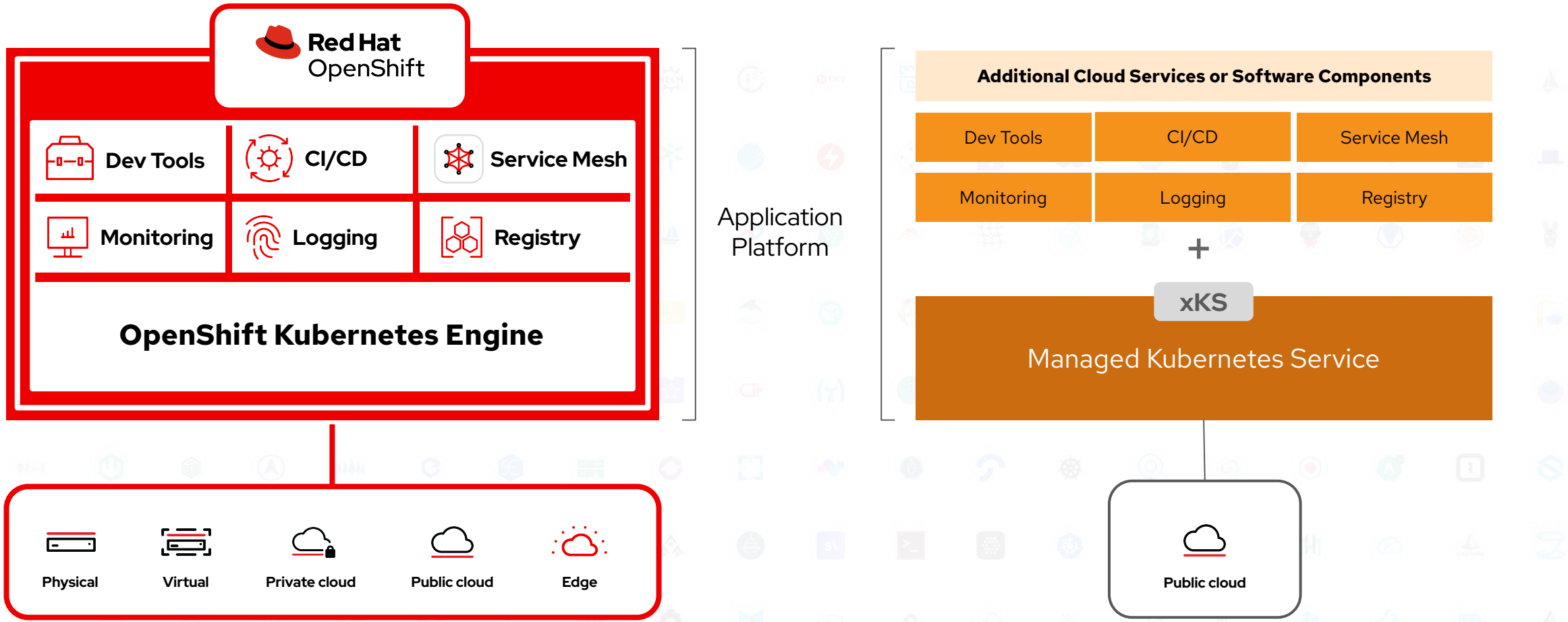
ハイブリッドクラウド全体
にわたる一貫性

Increase Flexibility

柔軟性 = ↑↑

OpenShift 包括的なアプリケーションプラットフォーム

ハイブリッド クラウド全体にわたって信頼性と一貫性を実現



Red Hat

Red Hat OpenShift

トラステッドで、包括的で一貫した
アプリケーション プラットフォーム

Developers	Web Console & CLI	Developer Tools	Developer Workspaces	
Build apps	 Runtimes	 Image build	 Serverless	 Cloud APIs
Deliver apps	 CI/CD	 GitOps	 Image registry	 Observability
Secure apps	 Auth & SSO	 Service mesh	 CVE scanning	 Enforce policy

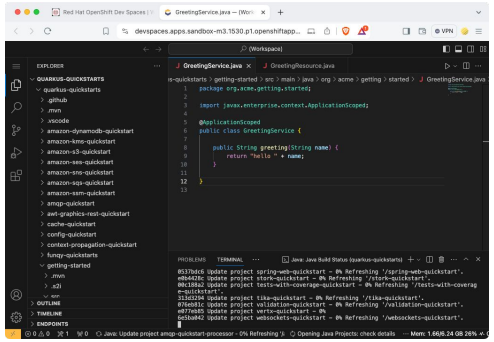
Build Applications

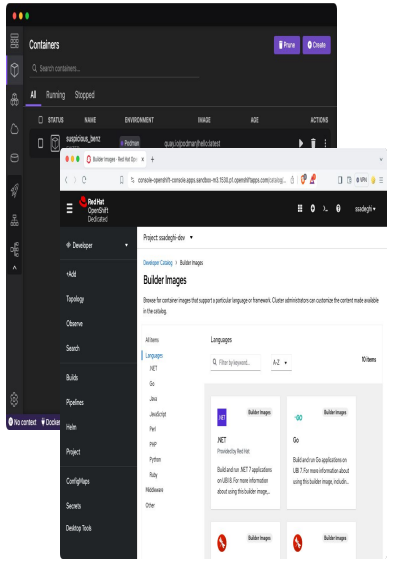
アプリケーションのビルド

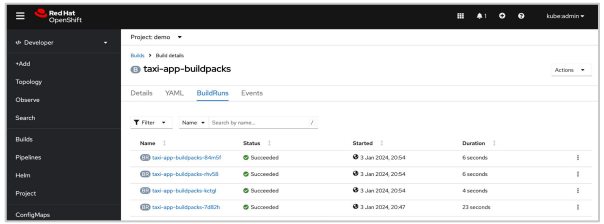
Red Hat

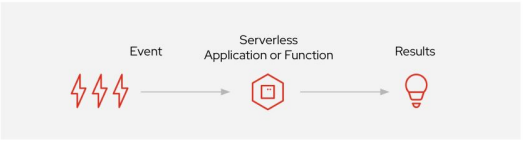
Build Applications
アプリケーションのビルド

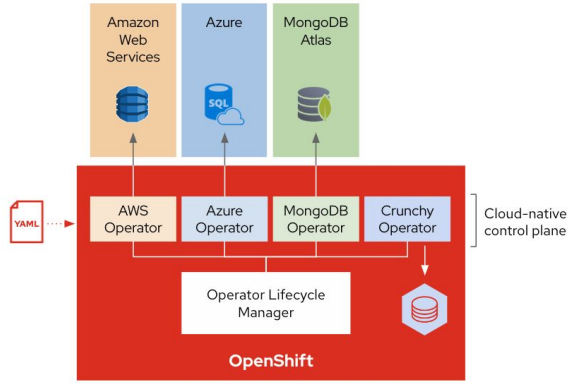
- 開発者の迅速なオンボーディングを可能にするWeb IDEを提供
- 既存アプリケーションにも対応した多様な開発言語やフレームワークをサポート
- ソースコードからコンテナイメージを自動生成するSource-to-Image（S2I）とBuildpacksを提供
- イベント駆動型のサーバーレスアプリや関数により、インフラ管理から解放されビジネスロジックに集中
- パブリッククラウドやコンテナサービスを統合的に管理するクラウドネイティブなコントロールプレーンを提供









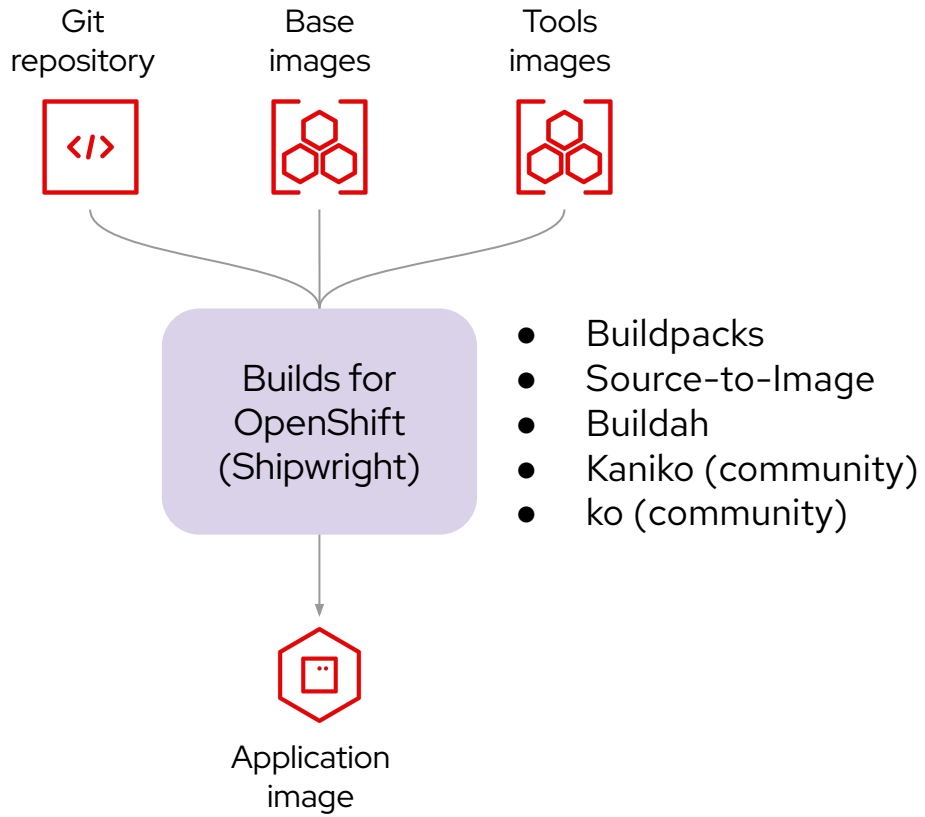


Red Hat

What's next:

Shipwright Buildsを用いたソースコードからのイメージ構築

- ソースコードからS2IやBuildpacksを使ってイメージをビルド
- PaketoコミュニティのBuildpacksとUBI Buildpacksのサポート
- 自分自身のBuildpacksを持ち込むことが可能
- Tekton、GitHub、GitLabのパイプラインでShipwrightビルドを実行
- マルチアーキテクチャビルド戦略（例：Armおよびx86）
- BuildConfigsからShipwrightビルドへの移行ガイド

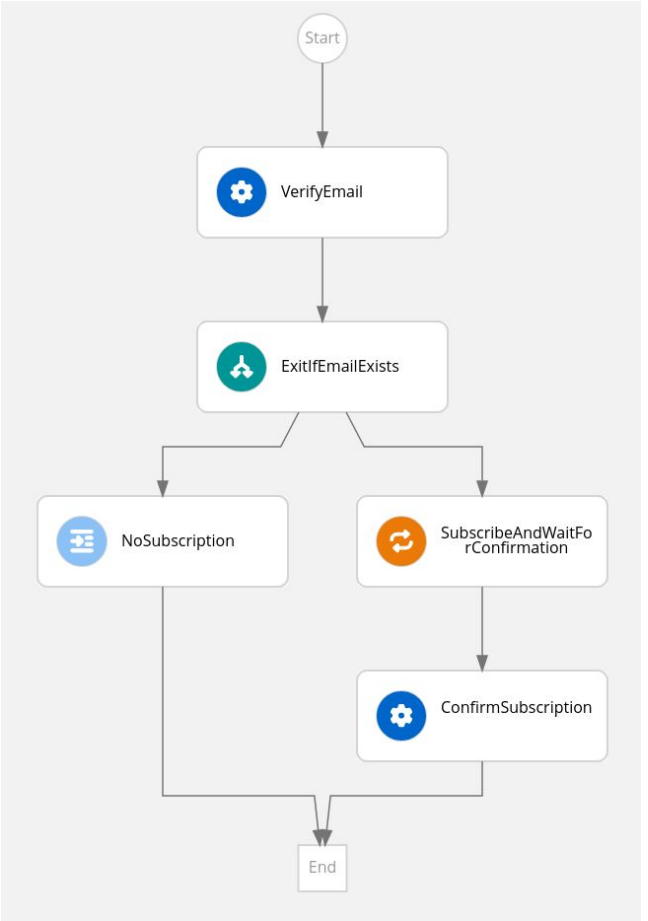


Red Hat

What's next:

Orchestrate serverless apps with serverless Logic

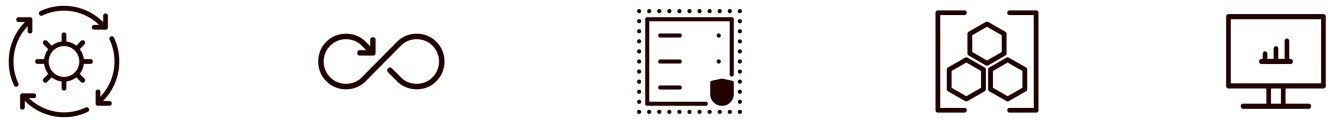
- サービスの呼び出しやイベントの統合した処理
- 状態を保持した長時間実行のワークフローを実現
- 長時間実行のワークフローでもリソースをゼロにスケールダウン
- データの整合性を保つためのエラーハンドリングと復旧策
- CNCFが定めたサーバーレスワークフローの仕様に準拠



Red Hat

Deliver Applications

アプリケーションの配信



Red Hat

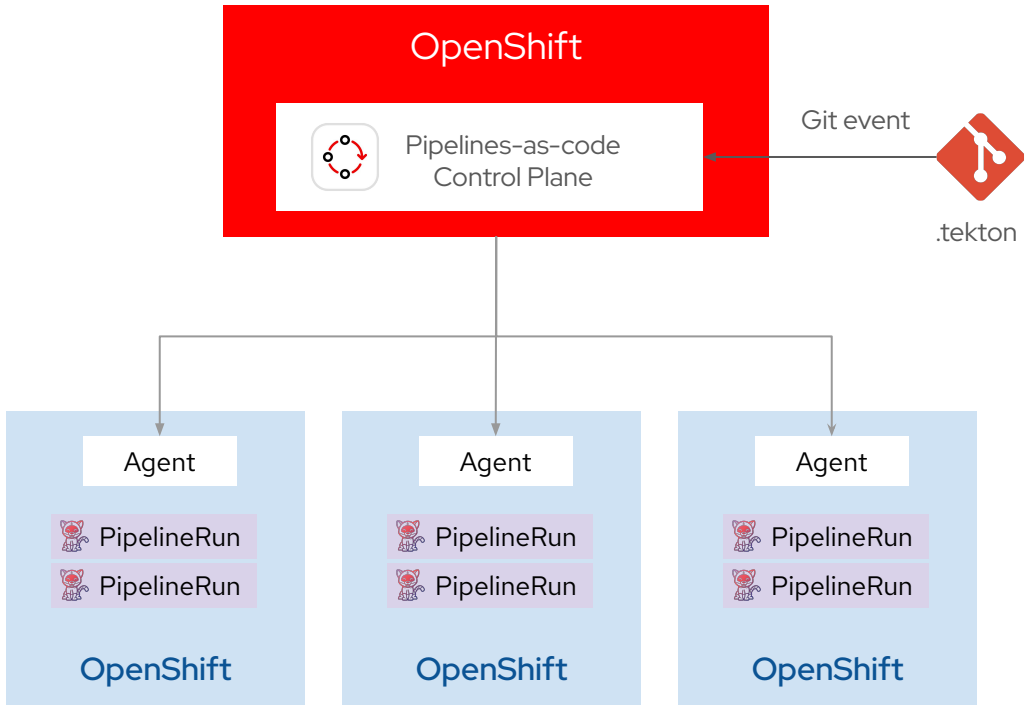
Deliver Applications
アプリケーションの配信

- ソフトウェアのビルド、テスト、セキュリティを自動化し、継続的インテグレーションを実現
- GitOps、カナリアリリース、デプロイ分析を用いた継続的デリバリー
- レガシーアプリケーションをクラウドネイティブな仮想マシンに移行
- 組織全体でコンテナイメージを安全に保存・配布
- メトリクス、アラート、トレース、ログ、電力消費監視を通じたフルスタックの可観測性

What's next:

マルチクラスタ環境でのPipelines-as-code

- ハブアンドスポークアーキテクチャ
- スポーククラスタ上のパイプラインエージェント
- ハブクラスタ上のリポジトリ、資格情報、設定
- コントロールプレーンがスポーククラスタにPipelineRunsをスケジュール：負荷分散、タグ戦略など
- メトリクス、ログ、アーティファクトに関する中央集約的なインサイト

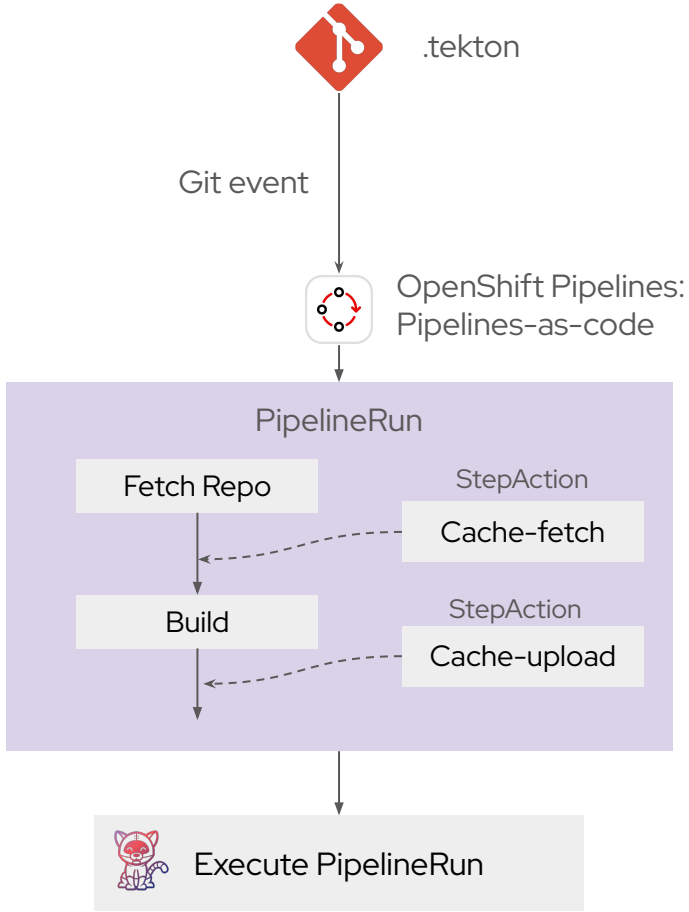


Red Hat

What's next:

Tekton Pipelinesでの依存関係のキャッシュ

- Tektonにおける依存関係キャッシングのステップアクション
- PipelineRun上のアノテーションを通じて設定可能
- Pipelines-as-Codeリゾルバーでの標準機能
- キャッシュストレージ:OCIアーティファクトおよびS3

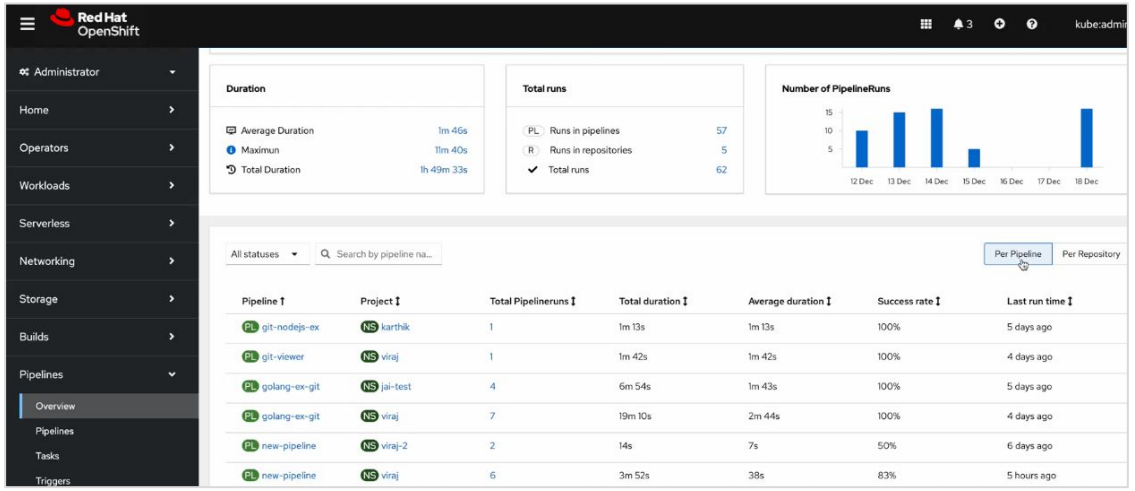


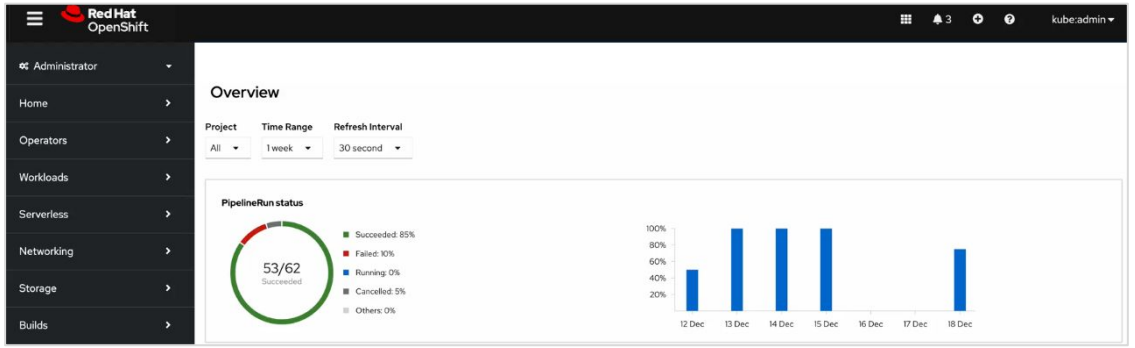
Red Hat

What's next:

長期的なパイプライン実行履歴

- パイプライン履歴のストレージがPipelineRunのカスタムリソースから切り離され、柔軟性を向上
- 実行メタデータがTekton Resultsを通じて保存され、監査に役立つ情報を提供
- ログ用のプラガブルストレージプロバイダーを利用可能で、保存の選択肢を拡大
- Tektonリソースの積極的な整理を実現し、効率的な管理をサポート
- クラスター内でのTektonパフォーマンスを向上させ、安定性とスピードを強化



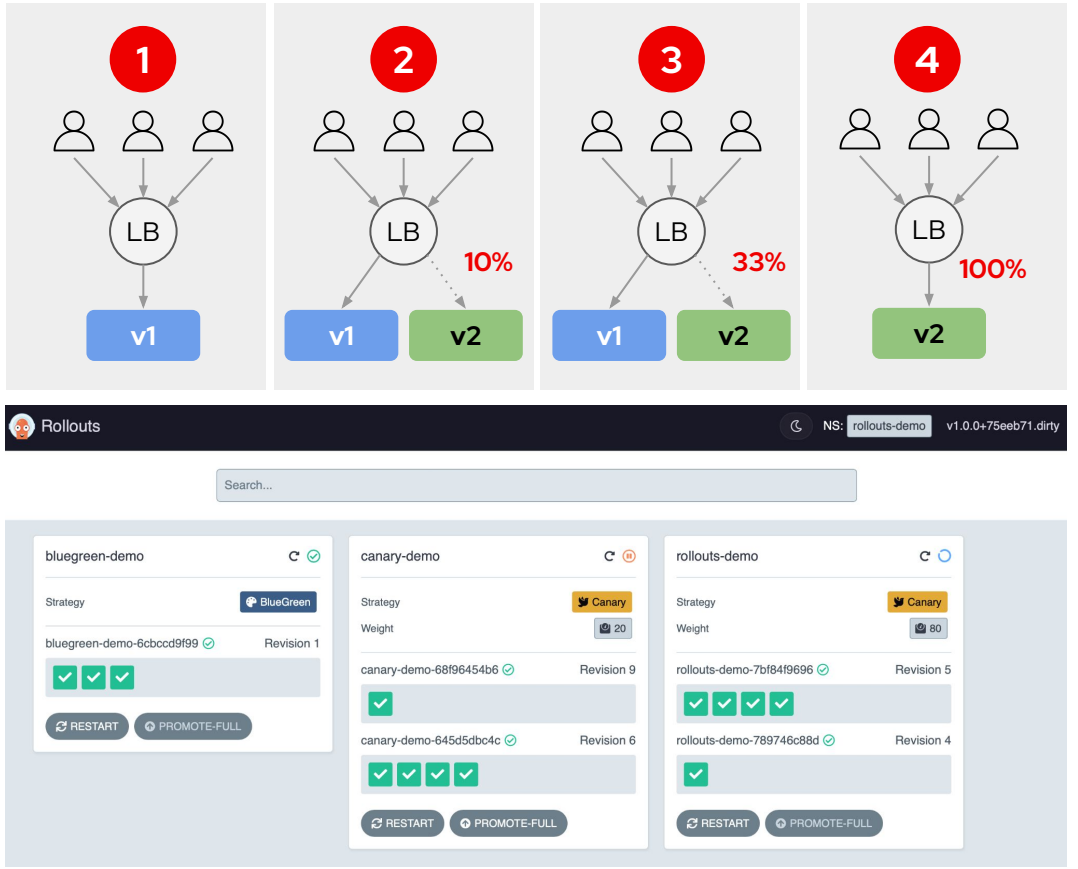


Red Hat

What's next:

"Argo Rollouts":デプロイ時のリスク低減

- ゼロダウンタイムのデプロイメントをGitOpsで実現:
  カナリアリリースやブルーグリーンデプロイメントの実施
- デプロイメント成功を評価するためのメトリクスベースの分析:
  Prometheus、DataDog、NewRelic、CloudWatch、HTTPなど
- 手動および自動によるロールバックとプロモーションの実行
- サービスメッシュによる細かいトラフィックシフトの実現



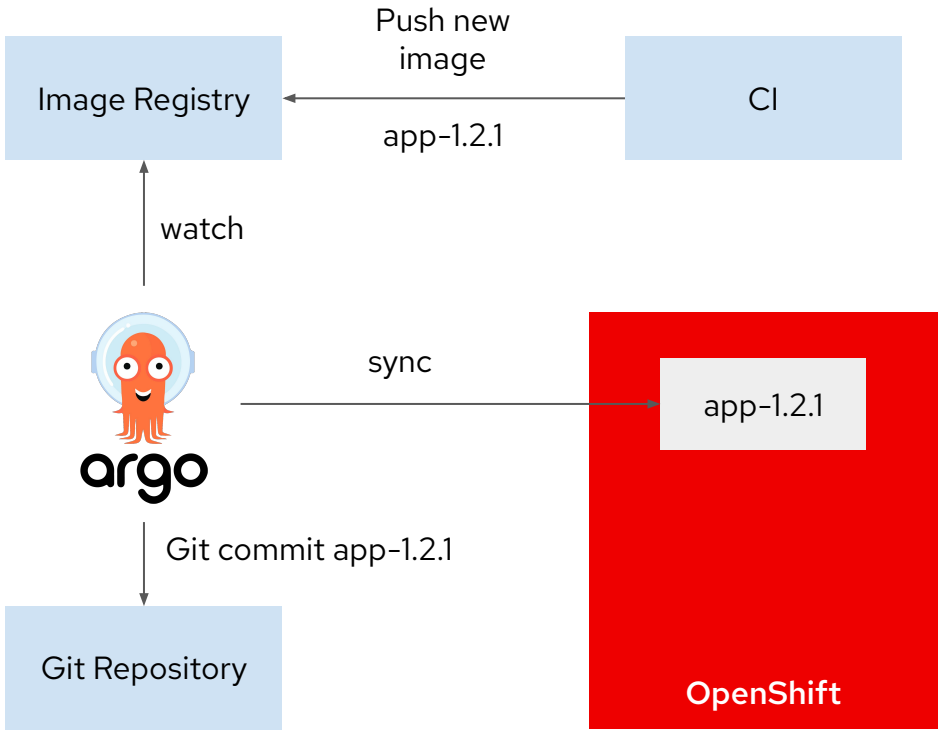
Red Hat

What's next:

"Argo CD Image Updater"新しいイメージの自動デプロイ

- イメージレジストリの監視 Quay、JFrog、GitHub、GitLab、GCPなどのイメージに対する更新を追跡
- Gitの書き戻しサポート: 永続的な更新を実現
- 複数の更新戦略: セマンティックバージョニング(semvar)、latest、digest

  Ex: `1.x`, `1.2.x`
- タグのフィルタリング: マッチャー関数を使用してタグを絞り込む

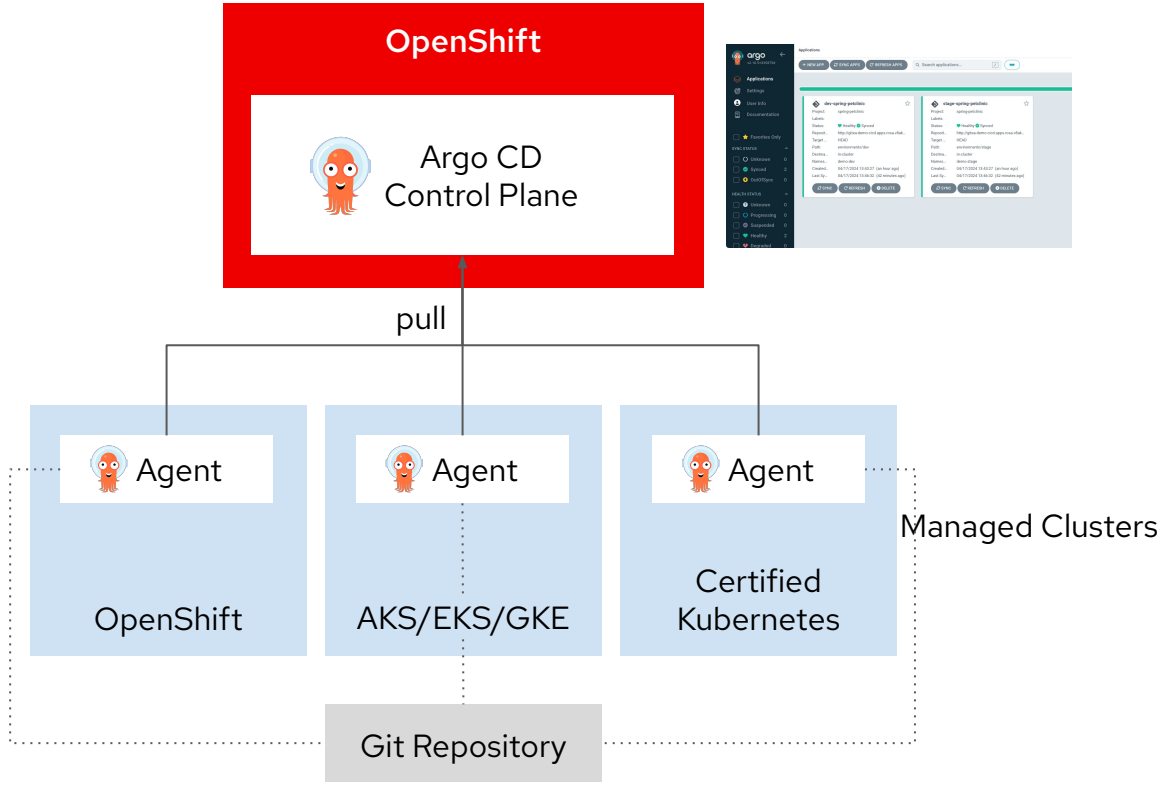


Red Hat

What's next:

Argo CD controlplane : マルチクラスターでGitOpsを拡張

- ハブ＆スポークアーキテクチャ: 中心にコントロールプレーンを持つ構成
- ハブクラスター上の単一コントロールプレーン: すべての管理を集中管理
- 管理クラスターに展開された小型エージェント: 各クラスターでの軽量な運用
- エージェントからコントロールプレーンへの通信: シームレスな接続
- 管理クラスターでのリコンシリエーション: 状態の一致を保持
- Red Hat Advanced Cluster Management (ACM)との統合: フリートとポリシー管理の効率化



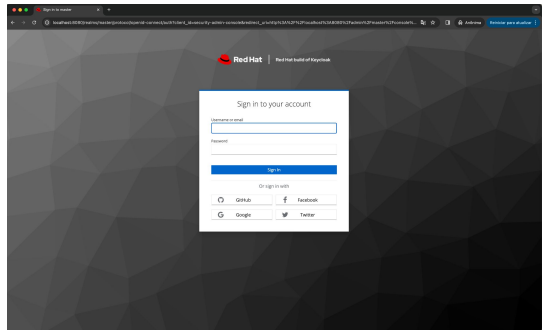
Red Hat

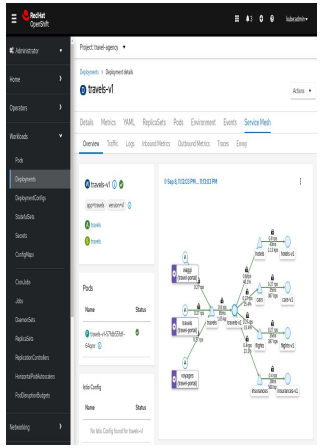
Secure Applications

安全なアプリケーション

Red Hat

Secure Applications





レガシーおよびクラウドネイティブアプリケーションの認証とシングルサインオン

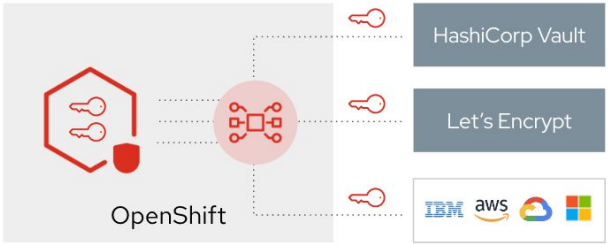
ゼロトラストセキュリティを実現するアプリケーションネットワーキングとサービスメッシュ

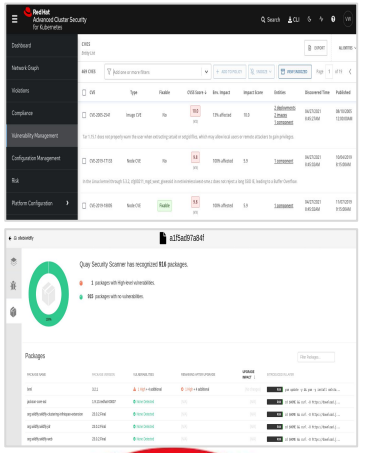
SSL/TLS証明書や機密データのオンデマンドライフサイクル管理

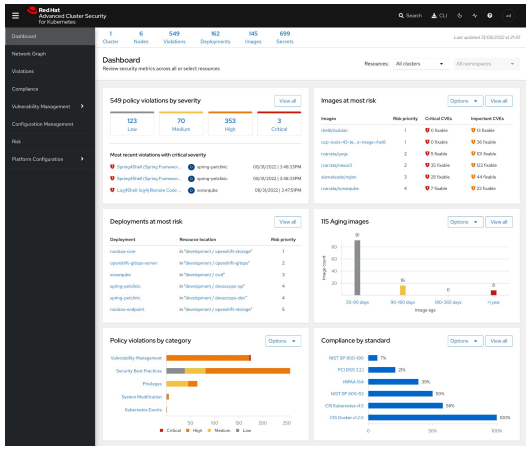
脆弱性管理とCVEスキャンによる早期修正

アプリケーションのビルド、デプロイ、実行時におけるセキュリティポリシーの適用

With Red Hat Advances Cluster Security





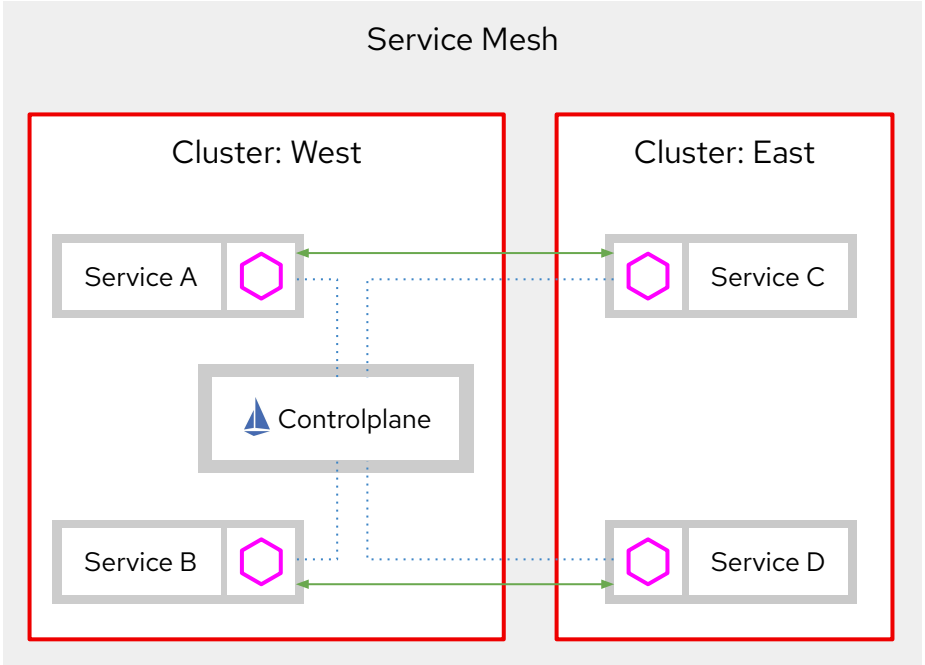


Red Hat

What's next:

OpenShift Service Mesh 3

- Istioとの統合: Maistraを使用せず、上流のIstioに統合
- Istioのコマンドラインユーティリティ: Istioctlが利用可能
- コントロールプレーンのリビジョンとカナリアアップグレード: バージョン管理とカナリアアップグレードの機能を提供
- マルチクラスターのトポロジー: マルチプライマリーや外部コントロールプレーンなどの構成をサポート
- Gateway API: サービスネットワーキングのための新しいKubernetes標準
- Ambientモードのデータプレーン: サイドカーなしでの設定が可能なデータプレーン
- Kubernetes外のサービスのメッシュへの統合: Kubernetes以外のサービスもメッシュに含めることができる



Red Hat

Red Hat Summit

Thank you

 linkedin.com/company/red-hat

 facebook.com/redhatinc

 youtube.com/user/RedHatVideos

 twitter.com/RedHat

Red Hat