WHITE PAPER | 2015 年 3 月

リリース自動化戦略への構成管理の統合

Tim Mueting / Paul Peterson
アプリケーション・デリバリ
CA Technologies

ca technologies

目次

エグゼクティブ・サマリー

課題

構成管理ソリューションと Infrastructure as Code（コードとしてのインフラ構成管理）の概念の導入は、データセンタ内で膨大な数の仮想マシン、オペレーティング・システム、ミドルウェアの構成と保守に取り組む企業に多くのメリットをもたらしてきました。これらのソリューションは、構成エラーが引き起こす障害を減らす上で大きな価値を提供しています。そのため、多くの企業がアプリケーションのデプロイを自動化するために、これらの同じツールを使用しようと試みてきました。

ビジネス・チャンス

構成管理とアプリケーション・リリース自動化ソリューションの両方の強みを活用するためには、より効率的で生産性の高い方法があると弊社では考えています。本書では CA のアプローチについて、そして、リリース自動化と構成管理の両方を使用する効果的なエンタープライズクラスのソリューションによって提供されるべきだと思われる重要な要件について説明します。また、構成管理ツールの機能のすべてをデプロイ・ワークフローに直接組み込むリリース自動化ソリューションから得られる、特有の価値について説明します。

メリット

エンタープライズ向けのアプリケーション・リリース自動化ソリューションは、現在の複雑なアプリケーションを、これまでになく高品質な状態でリソースの使用を抑えながら、迅速にデプロイするために必要な機能を提供します。構成管理ソリューションをリリース自動化プロセスに緊密に統合することで、アプリケーションを一貫した既知の環境にデプロイできます。また、統合されたソリューションはリリース・プロセス全体を明確に示し、リリースのデプロイ前とデプロイ中に行われた構成の変更を強調表示します。これは障害発生時に必要な根本原因分析を行う際に非常に有益な機能です。

セクション 1

アプリケーション・エコノミーに向けたアプリケーション・リリースの自動化

エンタープライズ・アプリケーション・リリース自動化プラットフォームが、可視性とガバナンスの改善やエラーの削減、ビジネス・リスクの緩和、デプロイにかかる時間の短縮、コスト削減に役に立つことは、長年の実績により証明されています。

アジャイル開発と複雑なハイブリッド・インフラストラクチャの環境では、企業の IT 組織はより組織的なアプローチをアプリケーションのデプロイに組み込む必要があります。現在最も成功を収めている企業は、競争優位性を維持するために、新しい革新的な機能をより迅速に高い品質で顧客に提供しなければならないことに気づいています。

増大するアプリケーションの複雑性、動的な IT 環境、そして旧式の手作業によるプロセスが原因で、リリース・サイクルの長期化、エラーやコストの増大、顧客の満足度とブランド・イメージの低下などにつながっています。多くのリソースを必要とする手作業のプロセスでは、品質や信頼性、効率性を維持しながら問題に対応することは不可能です。アプリケーション・リリース自動化プラットフォームを使用すれば、現在の複雑なリリース・プロセスを自動化することでこうした問題に正面から対応でき、同時に、継続的統合、ソース･コード管理、アーティファクト･リポジトリ、インフラストラクチャ･プロビジョニング、構成管理、テストの自動化、問題追跡などのツールの、継続的デリバリ・ツール・チェーンへの統合を調整できます。

アプリケーション・リリース自動化プラットフォームの要件

アプリケーション・リリース自動化プラットフォームには大きなメリットがあります。どの企業にとっても課題となるのは、業務に適したソリューションを選択することです。基本的に、効果的なプラットフォームは、以下が可能である必要があります。

- 一元的な制御機能を提供し、全面的なデプロイ、パッチ、緊急時ホット・フィックス、完全なロールバックなどアプリケーション・リリース・デプロイ作業を自動実行する
- ユーザ、アプリケーションおよび環境（開発、QA、運用）でプロセスを合理化し、自動化し、調整する。
- プロセスの増加や再作業なしに、物理、仮想、クラウド環境を含む異種混合のインフラストラクチャにわたり一貫したデプロイをサポートする
- 一般的なアプリケーションと複雑な多階層アプリケーションの両方のデプロイを加速する
- 変更管理ソリューションと統合し、自動変更管理アクティビティをサポートする
- アプリケーション・サービスのワークロード・キャパシティをスケーリングする
- 詳細な監査およびアプリケーション・サービス・レポートを出力する
- リリース・トレンドの詳細で包括的なダッシュボードを提供し、高レベルの IT マネージャがデプロイ・プロセスを監視し監査できるようにする
- 継続的インテグレーション、インフラストラクチャ、構成管理ソリューションから変更管理、問題追跡など、サポート対象のソフトウェア・デリバリ・ライフサイクル（SDLC）ツールとシームレスに統合する

アプリケーション・リリース自動化（ARA）プラットフォームはまた、以下を提供できることが必要です。

- アプリケーション・モデル、リリース・フロー、そのモデルに基づいたプロセスを構築し定義するためのビジュアルな設計環境
- テクノロジ・コンポーネント、デプロイ・プロセス・ワークフロー、多様なアーキテクチャおよび環境など、アプリケーションとそれをサポートするインフラストラクチャで構成されるすべてに関する構成要素を含むアプリケーション・モデル
- アプリケーションと環境にわたって標準の再利用可能なデプロイ・モデルを構築する方法
- リリース・プロセスを構築、構成、実行するデータ主導のリリース実行モデル
- 承認フローのためのカスタマイズ可能なユーザ・インタラクション、障害発生時の自動リリース・ロールバック、デプロイ後の操作の実行
- スクリプトやコマンド・ライン API を必要とせず、既存のすべての手作業による処理に変わる組み込み済みアプリケーション・アクション
- ソリューションから独立してデプロイし更新できる豊富な統合ライブラリとプラグイン
- すべてのリリース実行プロセスの継続性を確保する、リリース自動化環境とエージェントの自動フェイルオーバ
- 依存性を含むアプリケーション階層全体のワークフローを明確に表示する単一のビュー
- アプリケーション環境にまたがるリリース・パイプライン（開発、QA、運用など）を表示する単一のビュー
- アプリケーション、アプリケーション・コンポーネント、環境、リリース別にパラメータを格納および共有できるキャパシティ
- 複数サーバ間での並行または連続実行を管理および選択できるキャパシティ
- アプリケーション階層間で、トリガに基づいたアクションにより、多階層および複数サーバ・ワークフローに「ワンクリック」の自動化プロセスを実行できる機能。例：ワークフローがデータベース・サーバから開始され、所定のステップの後、このワークフローがアプリケーション・サーバで開始され、完了すると、別のワークフローがデータベース・サーバで継続される。

構成管理とアプリケーション・リリース自動化

前述の通り、多くの企業では Puppet や Chef などの構成管理およびインフラストラクチャ・プロビジョニングのツールを採用し、プロビジョニングされたサーバが企業ポリシーとガイドラインに沿うようにしています。

これらのツールは事前定義されたポリシーを使用して、物理および仮想のインフラストラクチャ・ソフトウェアをデプロイし構成します。これによってシステム管理者は、異なる環境間にまたがるインフラストラクチャの構成について特定のポリシーを定義して適用することができます。たとえば、サーバが適切なロギングおよびセキュリティ設定を備えるようにするポリシーを適用できます。また、オペレーティング・システムの更新、ミドルウェア、アプリケーション・コンポーネントを基本インスタンスに適用でき、新しいマシン・イメージを構築する必要がありません。

リリース自動化プラットフォームは、1 つの環境から次の環境へ、そして最終的に本番へと、次の段階へのアプリケーションのプロモーションに関わるあらゆる手順の自動化に対応します。デプロイ計画は共有の再利用可能なコンポーネントで定義され、これはビルドのアーティファクト、アプリケーションのコンテンツ、デプロイの特定順序を、アプリケーションを特定の環境にリリースするために必要な依存関係や構成と組み合わせます。

構成管理ソリューションは、各ターゲット環境の構成が通常の更新プロセス以外で変更されていないことを保証するために、この段階できわめて重要な役割を果たします。これは**構成ドリフト**と呼ばれるものです。不一致が見つかった場合は、自動的に（または手作業で）構成の修正作業を行い、その後 ARA ソリューションに制御を戻してデプロイが継続されるようにソリューションを定義することができます。状況によってはさらに詳細な分析を行うために、デプロイ全体を停止する必要があるような不一致があると判断されることがあります。あるいは修正作業を適用する必要がなく、 中断せずにデプロイを継続できると判断される場合もあります。

専用のアプリケーション・リリース自動化ソリューションの 5 つのメリット

1. グラフィカルなワークフロー・デザイナとリリース・モデル化機能

   - ARA ソリューションはアプリケーション層を中心にしており、構成管理ツールはノード層を中心にしています。ARA ソリューションの重要な機能には、大規模な異種混合環境と多数のデータセンタにまたがる多階層アプリケーションのデプロイに必要なグラフィカルな設計機能と、アプリケーション･コンポーネント、ワークフロー、相互依存関係をすべてモデル化する機能が含まれている必要があります。

2. インテリジェントなロールバック機能による自動化された完全デプロイおよび増分デプロイ

   - エンタープライズ ARA ソリューションは、障害発生時にアプリケーションを前の状態に戻す自動ロールバック・プロセスを備えた、アプリケーションの完全デプロイとインクリメンタルなデプロイをサポートする機能を提供する必要があります。また、この機能は手作業による入力と承認も受け入れられることが必要です。

3. 自動プロモーションと環境の自立性

   - 今日の企業は、相互に連動し依存関係にあるアプリケーションを週に数千もデプロイを実行することがあります。前述のように、主要な ARA ソリューションはデプロイ・ワークフローやアプリケーション層の相互依存関係など、アプリケーションのすべてのコンポーネントをモデル化します。これは、いくつもの環境にまたがる多数のアプリケーションのデプロイとプロモーションに使用できる共通の再利用可能なコンポーネントとワークフローによって、デプロイを標準化するために必要な抽象化が提供されます。

4. 継続的デリバリ・ツール・チェーンの統合を調整する機能

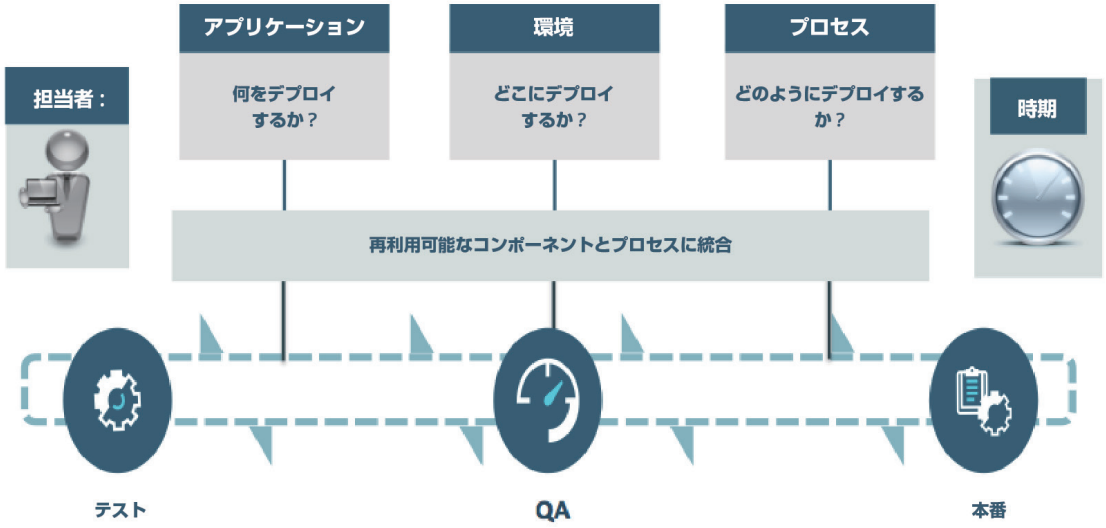
   - 主要な ARA ソリューションは、継続的インテグレーション、ビルド、成果物リポジトリ、インフラストラクチャ・プロビジョニング、構成管理、自動テスト、サービス・デスク・ソリューションなど、豊富な SDLC ツールへの統合ライブラリを提供し、シームレスで迅速なアプリケーション・デプロイを可能にします。

5. 大規模なエンタープライズ環境に対応するスケーラビリティ

   - 構成管理ソリューションは通常、ノードおよび環境指向で、その観点から高度なスケーラビリティを備えています。ARA ソリューションはアプリケーション全体をモデル化し、大規模な異種混合の環境とデータセンタ・ロケーションにわたるアプリケーションのデプロイと次のステージへのプロモーションの調整に必要なスケーラビリティを提供します。また ARA ソリューションは、アプリケーションとリリース・レベルの両方に役割ベースのセキュリティ、承認ワークフロー、詳細なレポート機能、監査機能を提供し、可視性とガバナンスを向上させます。

図 1

アプリケーション・リリース・モデル



セクション 2

構成管理の詳細

繰り返しますが、アプリケーション・リリースの自動化と構成管理の統合の目標は、アプリケーション・デプロイを迅速に高い品質でシームレスに行えるようにすることです。

この目標を達成するには、プロセス、組織、テクノロジーという 3 つの領域で効果的な取り組みを行うことが必要です。

- プロセス
  - 企業は新規機能を市場に迅速に提供するために、新しいプロセスを採用する必要があります。共通の再利用可能な効率性の高いプロセスを導入し、構成管理チームとアプリケーション・リリース・チームの両方をサポートすることが必要です。これらのチームは多くの場合、週に数千件とはいわないまでも、数百件ものデプロイを担っています。
- 組織
  - DevOps 環境では多くの場合、構成管理チームは運用チームの一部であり、リリース・チームはアプリケーション開発チームの一部です。最近ではリリース管理チームがアプリケーション・デプロイのすべての調整と実行を担う中央組織の一部になっている例が増えています。いずれの場合も、すべてのチームが連携し強調して作業することが必須です。アプリケーション・リリース・チームはインフラストラクチャ上にデプロイされたときにアプリケーション・デプロイに障害が起きないようにするために、基盤となる環境の適切な構築・構成を構成管理チームに委ねています。

▪ テクノロジー

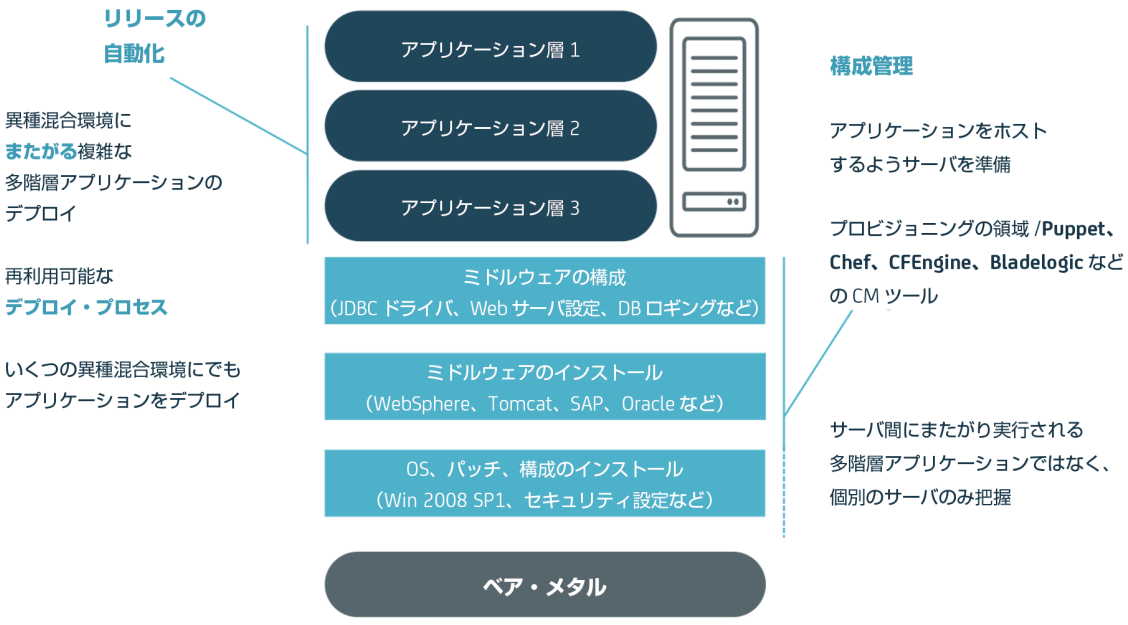
- 前述のとおり、週に数百件あるいは数千件のデプロイをサポートするために、構成管理とリリース自動化ソリューションはシームレスに連携する必要があります。デプロイを成功させるには、複雑な変更の動作をインフラストラクチャ・レベルとアプリケーション・レベルの両方で調整する必要があります。そのため、ソリューション間を緊密に統合させて整合性を確保し、デプロイの遅れや全体的な障害を回避することが重要です。

各ソリューションの能力の活用

これら 3 つの領域への対応の必要性を踏まえた上で、こうした高度な目標の達成に役立つソリューションとはどのようなものが考えられるでしょうか？どちらのチームも非常に多忙で、それぞれが異なるスキル・セットを備えています。効果的なソリューションとはそれぞれの能力を活用し、それらを合わせてよりいっそうの価値を引き出せるものであることが必要です。

リリース・マネージャとリリース管理者は、開発から本番までリリース・パイプラインの各ステージにわたるアプリケーションのデプロイの調整に熟達しています。しかし彼らの多くは通常の構成管理ソリューションで必要な種類のスクリプトの設計と構築にはそれほど詳しくありません。また逆に、構成管理チームのスタッフはスクリプトの理解と実装には非常に長けていますが、SDLC のステージを通じてアプリケーションを先に進めるためのスキルやナレッジは不足していることがあります。

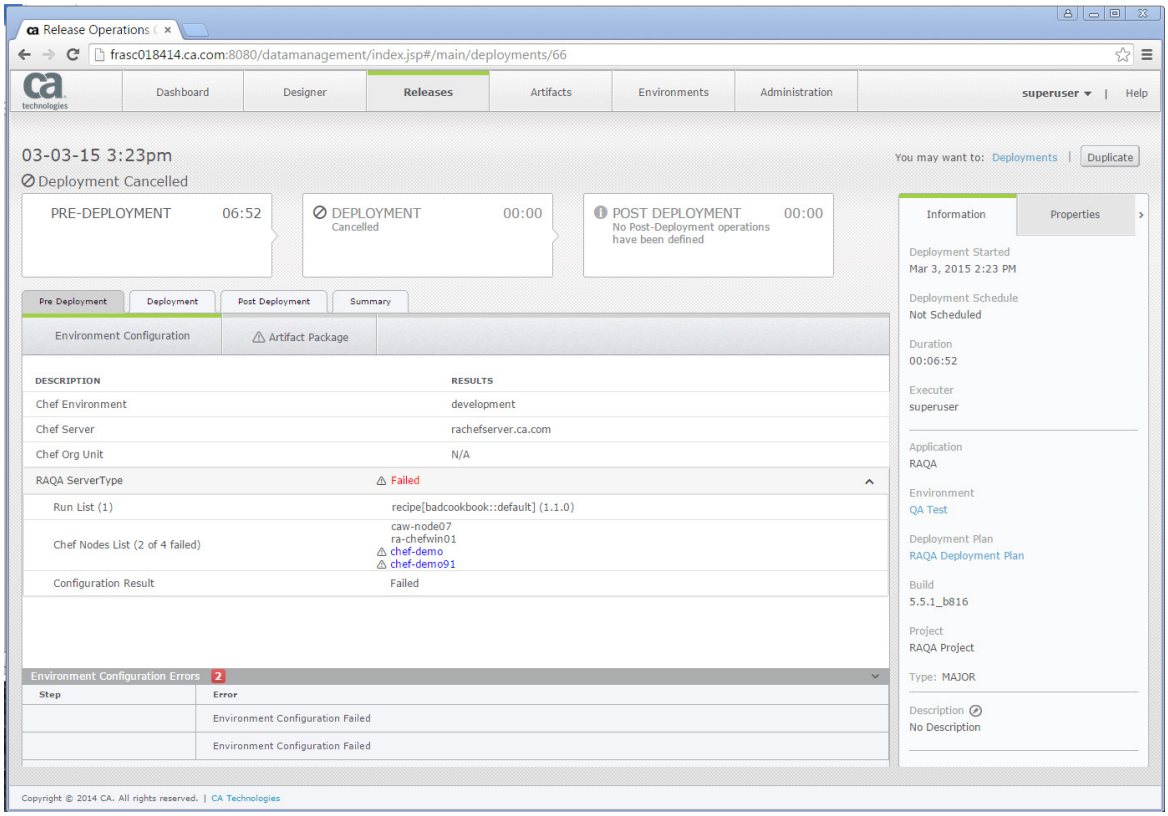
図 2



つまり、両方のチームが連携できる方法を用意することが重要です。ほとんどの ARA ソリューションでは通常、作者 / ユーザ・モデルが使用されており、「作者」の役割がリリース・プロセスおよびワークフローの設計を行い、「ユーザ」の役割がリリース・プロセスの管理を行います。同じモデルを構成にも適用できます。構成管理チームは各チームからの同意にしたがい構成ポリシーと定義を作成し、その後、リリ - ス管理チームがそれらをリリース・プロセスにシームレスに統合します。これによって両方のチームの目標を実現でき、シームレスな自動化された再利用可能で監査可能な、スケーラビリティに優れたソリューションを提供できます。

図 3

デプロイ前の構成



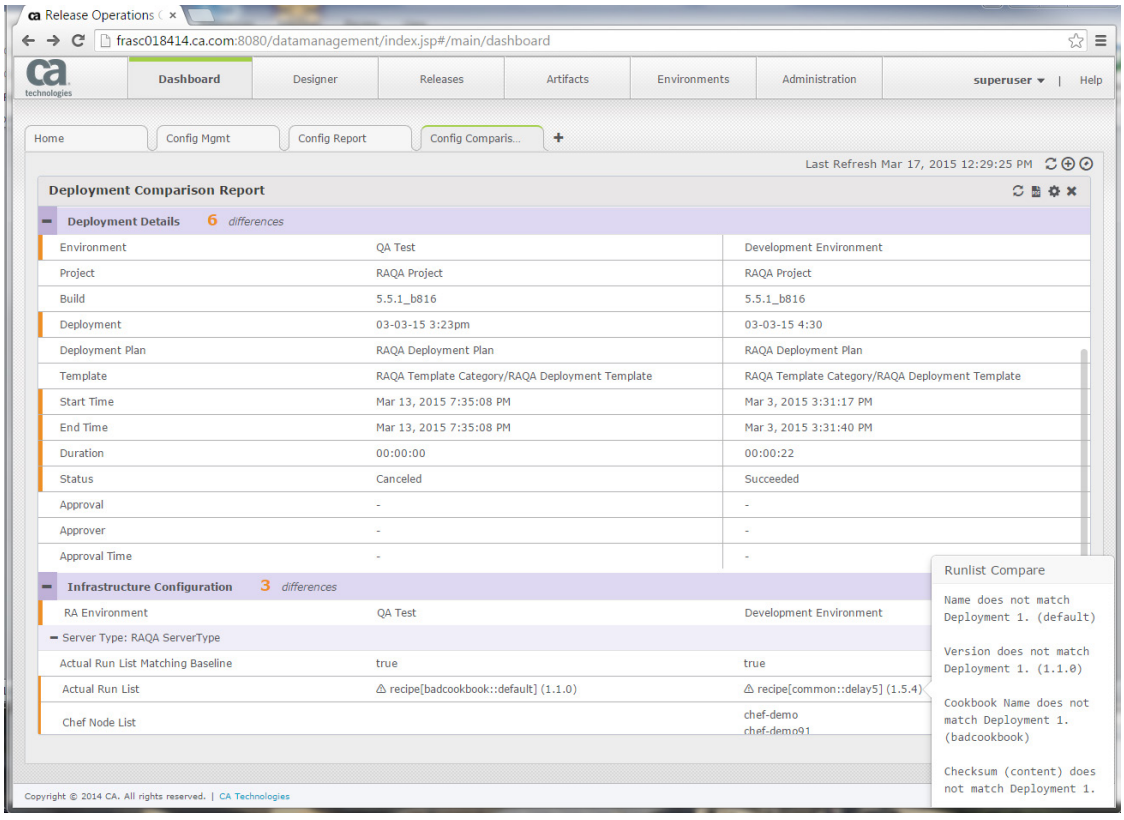
つまり、構成管理とリリース自動化の効果的な統合には以下を含める必要があります。

- 構成管理製品の複数のインスタンスへのアクセスを可能にする、構成管理ソリューションへのシームレスな統合が標準で用意されていること
  - たとえば、リリース自動化の UI 内から多数の Chef インスタンスやセキュリティ組織を定義、ログイン、使用
- 簡単なセットアップと構成
  - 構成管理ソリューションの役割 / タイプへのサーバ・タイプと環境のマッピングを提供する
  - 構成管理ベンダのポリシーの定義をリリース自動化ソリューションへ、ポリシー・マッピング機能を提供する
    - たとえば、SDLC の一部としてデプロイのために環境によって Chef のレシピやクックブックのバージョンを選択
  - リリース自動化ソリューションの UI 内からノードに Chef クライアントをブートストラップ / インストールする機能
- アプリケーション / デプロイの基準
  - バージョンおよび環境別の構成管理ポリシーの定義を、リリース自動化ソリューション内のアプリケーション / デプロイの基準に関連づける機能
  - 期待されるアプリケーション・デプロイを損ねる可能性がある、これらのポリシーへの変更を把握する機能
  - 構成ドリフトを追跡するための基準を確立し、アプリケーション、環境、デプロイにわたってレポートを作成する機能

- リリース・プロセスの一部としての環境構成ステージ
  - 要求を構成管理ソリューションにネイティブに組み込み、リリース・デプロイ・プロセスの一部として基礎となるノード基準を検証または更新する
  - リリース自動化デプロイ・ステータスの一部として、構成管理ソリューションのアクティビティのステータスを、成功か失敗かを含めレポートする
  - 構成管理ステージを迅速に実行する機能。ソリューションは、各環境で並行して構成プロセスを実行するために、基盤となる環境へのマルチスレッド呼び出しをサポートする必要があります。大量のノードがあると、デプロイ期間中に処理に時間がかかることがあります。
- レポートと分析
  - デプロイ比較レポート
    - アプリケーションおよびデプロイのドリフトに関するレポートの一部として、バージョンの変更など、ポリシーの違いを強調表示する
  - 構成ドリフト— 重要性能評価指数（KPI）
    - アプリケーション、リリース、環境について、個別ビューまたは複合ビューで長期的に構成の変更に関する数値とタイプを強調表示する
  - すでに行われたインフラストラクチャ・ポリシーの変更に関する詳細なビュー

図 4

デプロイ比較レポート



セクション 3

まとめ

構成管理ツールは全体的なエンタープライズ・アプリケーション・リリース自動化戦略の重要なコンポーネントを提供しますが、これまで見てきたとおり、アプリケーション・リリース自動化ソリューションそのものに置き換わるものではありません。構成管理ツールを使用すれば構成エラーが原因で起きるアプリケーション・デプロイの障害リスクは軽減されますが、エンタープライズ環境のアプリケーション･リリース自動化の場合は、スケーラビリティと機能の点で構成管理ツールには限界があります。

構成管理とアプリケーション・リリース自動化を統合させることで、インフラストラクチャ構成管理は DevOps およびアプリケーション・リリース・チームにとって容易でシームレスな作業になります。一貫した環境を維持することは、開発しテストしたアプリケーションが、顧客に提供するアプリケーションと同じになるようにするために非常に重要です。アプリケーション・デリバリ・チームは、システムが既知の状態にあり、アプリケーション・デプロイの基準に沿っていることを確認できることが必要です。構成管理ソリューションはこの既知の状態を定義し維持するために役立ちます。

CA Release Automation を使用することで、企業は構成管理をデプロイ･ワークフローにシームレスに統合でき、アプリケーションとデプロイの基準に対して構成を照合し確認できます。ミドルウェア、アプリケーション、環境における構成ドリフトに関して問題が検出されれば、単一のユーザ・インタフェース内で即座に修正し、追跡してレポートを作成することができます。

セクション 4

著者について

Tim Mueting、製品マーケティング担当ディレクター

Tim Mueting は CA Technologies のアプリケーション・デリバリ・ソリューションの製品マーケティング責任者で、継続的デリバリとリリース自動化を主に担当しています。プリセールス・コンサルタント、製品マネージャ、製品マーケティング・マネージャとして大規模エンタープライズ IT ソリューションのデリバリとマーケティングに 20 年以上携わった経験があり、エンタープライズ管理、仮想化、クラウド・コンピューティングなど、さまざまなトピックについて業界のイベントで頻繁に講演者を務めています。

Paul Peterson、製品管理担当シニア・ディレクター

Paul Peterson は CA Technologies の製品管理担当シニア・ディレクターで、CA DevOps 製品ラインについて責任を担っています。Paul Peterson は現在、CA Release Automation、Chef や Puppet などのインフラストラクチャ構成管理ソリューション、そしてクラウド管理プラットフォームの間の緊密な統合によって、CA Release Automation に関する需要領域の環境における高価値の使用事例を推進する役割を担っています。IT 業界では公共機関と民間企業の両方で 20 年を超えるキャリアを有し、運用チームにおける性能、キャパシティ、構成、可用性の管理のほか、マネージド・サービス・プロバイダやソフトウェア製品マネージャとして顧客へのサービス・デリバリを担当した経験があります。

ca.com/jp/でCA Technologiesにアクセスしてください

CA Technologies（NASDAQ:CA）は、企業の変革を推進するソフトウェアを作成し、アプリケーション・エコノミーにおいて企業がビジネス・チャンスを獲得できるよう支援します。ソフトウェアはあらゆる業界であらゆるビジネスの中核を担っています。プランニングから開発、管理、セキュリティまで、CA は世界中の企業と協力し、モバイル、プライベート・クラウドやパブリック・クラウド、分散環境、メインフレーム環境にわたって、人々の生活やビジネス、コミュニケーションの方法に変化をもたらしています。詳細については **ca.com/jp** をご覧ください。

CS200_123958_0315