

WHITE PAPER | MARCH 2015

Integrating Configuration Management Into Your Release Automation Strategy

Tim Mueting / Paul Peterson
Application Delivery
CA Technologies



Table of Contents

Executive Summary	3
Section 1: Application Release Automation for the Application Economy	4
Section 2: Specifics of Configuration Management	7
Section 3: Conclusions	11
Section 4: About the Authors	12

Executive Summary

Challenge

The introduction of configuration management solutions and the concept of Infrastructure as Code have provided many benefits to organizations struggling to configure and maintain the exploding number of virtual machines, operating systems and middleware within our datacenters. These solutions have provided tremendous value in reducing failures due to configuration errors. So much so, that many organizations have attempted to apply these same tools to automate application deployments.

Opportunity

We believe there is a more efficient and productive methodology that leverages the strengths of both configuration management and application release automation solutions. In this paper, we'll describe our approach and the key requirements we believe an effective, enterprise-class solution that uses both release automation and configuration management, should provide. We'll describe the unique value you can get from a release automation solution that incorporates the power of a full configuration management tool directly within the deployment workflow.

Benefits

Enterprise Application Release Automation solutions provide the capabilities needed to rapidly deploy today's complex applications with higher quality and fewer resources than ever before. Tightly integrating a configuration management solution into your release automation processes further ensures that applications are deployed to a consistent, known environment. An integrated solution will also give you a clear picture of the entire release process while highlighting any configuration changes made prior to or during the release deployment, which is a valuable feature when performing any root-cause analysis needed in the event of a failure.

Section 1:

Application Release Automation for the Application Economy

It has been proven time and again that enterprise application release automation platforms help improve visibility and governance, reduce errors, mitigate business risks, accelerate deployment times and decrease costs.

In the world of rapid, agile development and complex hybrid infrastructures, enterprise IT organizations must incorporate a more systematic approach to the deployment of applications. In order to maintain a competitive advantage, today's most successful companies are finding that they must deliver new and innovative capabilities to their customers faster and with higher quality.

Increasing application complexity, dynamic IT environments and antiquated manual processes have led to longer release cycles, increased errors, higher costs as well as dissatisfied customers and a tarnished brand. Manual, resource-laden processes simply cannot meet the challenge while maintaining quality, reliability and efficiency. An application release automation platform should empower organizations to meet these challenges head-on by automating today's complex release processes while orchestrating the integration of tools such as continuous integration, source code management, artifact repositories, infrastructure provisioning, configuration management, automated testing and issue tracking into a continuous delivery tool chain.

Requirements for an Application Release Automation Platform

The benefits of an application release automation platform can be immense. The challenge for any organization is to ensure that they select the right solution for the job. Fundamentally, an effective platform should be able to:

- Provide centralized control and automatic execution of application release deployment tasks including full-scale deployment, patches, emergency hot fixes and complete rollbacks.
- Streamline, automate and coordinate processes across users, applications and environments (Dev, QA, Ops).
- Support consistent deployments across heterogeneous infrastructures, including physical, virtual and cloud environments, without process proliferation or rework.
- Speed application deployments for both routine and complex multi-tier applications.
- Integrate into change management solutions in support of automated change management activity.
- Scale application service workload capacities.
- Provide granular audits and application service reports.
- Provide a sophisticated and comprehensive dashboard of release trends, enabling high-level IT managers to monitor and audit the deployment process.
- Enable seamless integration with supported Software Delivery Lifecycle (SDLC) tools from continuous integration and infrastructure and configuration management solutions through to change management and issue tracking.

An Application Release Automation (ARA) platform should also provide:

- A visual design environment for creating and defining the application model, release flows and processes based on that model.
- An application model that includes constructs for everything that comprises the application and its supporting infrastructure: technology components, deployment process workflows, varying architectures and environments.
- Methods to create standard reusable deployment models across applications and environments.
- A data-driven release execution model that builds, configures and executes release processes.
- Customizable user interaction for approval gating, automated release rollback in case of failure and execution of post-deployment operations.
- Built-in application actions to replace all existing manual operations, without the need for scripts and command line APIs.
- An extensive library of integrations and plug-ins, which can be deployed and updated independently of the solution.
- Automatic failover of release automation environments and agents to ensure continuity of all release execution processes.
- A single view clearly displaying the workflows across application tiers including dependencies.
- A single view of the release pipeline across application environments (Dev, QA, Ops, etc.).
- The capacity to store and share parameters per application, application components, environments and releases.
- The capacity to control and select parallel or serial executions between multiple servers.
- The ability to perform “one-click” automation processes on multi-tier and multi-server workflows with trigger-dependent actions between the application tiers. For example: a workflow is started on the database servers, after a certain step the workflow will start on the application servers and, when completed, another workflow will continue on the database servers.

Configuration Management and Application Release Automation

As mentioned, many organizations have adopted configuration management and infrastructure provisioning tools such as Puppet or Chef to ensure that provisioned servers adhere to corporate policies and guidelines.

These tools use predefined policies to deploy and configure infrastructure software, both physical and virtual. This enables system administrators to define and enforce specific policies for infrastructure configuration across environments. For example, policies could be enforced to ensure servers have appropriate logging and security settings. In addition, updates to operating systems, middleware and application components could be applied to base instances eliminating the need to build new machine images.

A release automation platform takes care of automating all the steps involved in the subsequent promotion of an application from one environment to the next, ultimately to production. Deployment plans are defined with shared, reusable components, which combine the build artifacts, application content, the

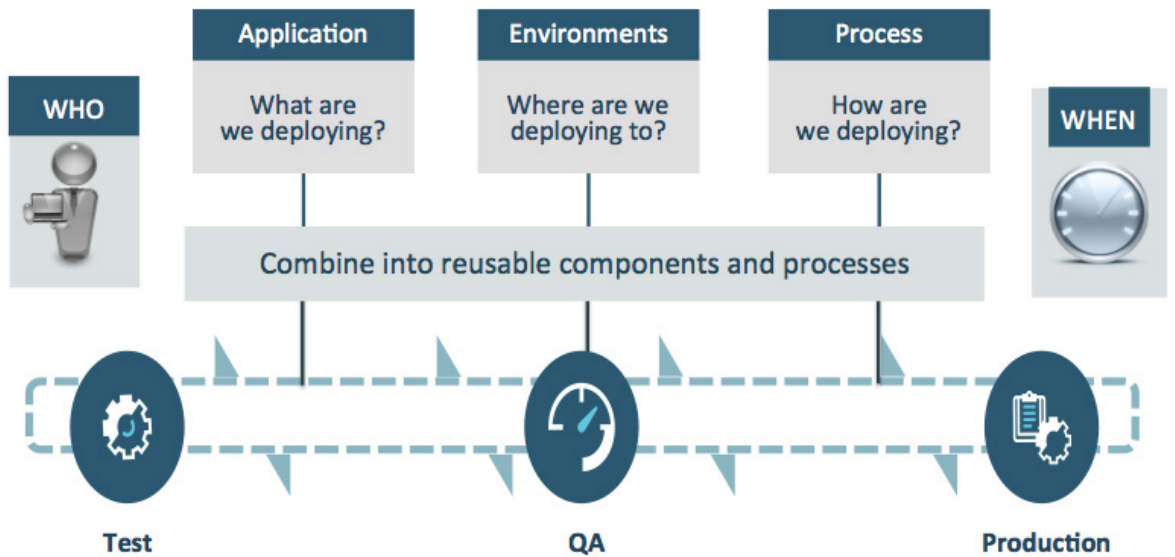
specific order of deployment with dependencies and the configurations required to release an application to a specific environment.

Configuration management solutions can play a crucial role during this stage by guaranteeing the configurations of each target environment haven't been modified outside the normal update process. This is referred to as *configuration drift*. If discrepancies are found, these solutions can be defined to automatically (or manually) take action to correct the configuration and then hand back control to the ARA solution to continue on with the deployments. In some situations it may be determined that the discrepancy is such that the entire deployment should be stopped for further analysis. In others situations, it may be determined that it isn't necessary to apply the corrective action and the deployment can continue on without interruption.

Five Advantages of a Purpose-Build Application Release Automation Solution

1. Graphical workflow designer and release modeling capabilities
 - ARA solutions focus on the application layer while configuration management tools focus is at the node layer. A key capability of any ARA solution should include the ability to graphically design and model all application components, workflows and interdependencies required for deployment for multi-tier applications across large heterogeneous environments and multiple data centers.
2. Automated, full and incremental deployments with intelligent rollback capability
 - Any enterprise ARA solution should provide the ability to support both full and incremental application deployments with automated rollback processes that return the application to a previous state in case of failure. This capability should also have the provision for accepting manual input and approval.
3. Automated promotion and environment independence
 - Today's enterprises can execute thousands of application deployments a week, each with interlocking dependencies. As noted above, leading ARA solutions model all the components of an application, including deployment workflows and the interdependencies among application tiers. This provides the abstraction needed to standardize deployments with common reusable components and workflows that can be used for multiple application deployments and the promotion across any number of environments.
4. Ability to orchestrate the integration of a continuous delivery tool chain
 - Leading ARA solutions provide a library of integrations to an abundance of SDLC tools such as continuous integration, build, artifact repositories, infrastructure provisioning, configuration management, automated testing and service desk solutions for seamless and rapid application deployment
5. Scalability for large, enterprise environments
 - Configuration management solutions are typically node and environment oriented and are highly scalable from that perspective. ARA solutions model the entire application and provide the scalability needed to orchestrate the deployment and subsequent promotion of applications across large, heterogeneous environments and data center locations. In addition, enterprise ARA solutions provide role based security, approval gating and detailed reporting and auditing at both the application and release level for improved visibility and governance.

Figure 1.
Application Release Model



Section 2:

Specifics of Configuration Management

To reiterate, the goal of application release automation and configuration management integration is to provide a seamless exercise by which application deployments can be deployed quickly and with high quality.

Achieving that goal involves working effectively in three areas: process, organizational and technical.

- Process
 - As organizations move to rapidly deliver new capabilities to the market, they must adopt new processes. Common, repeatable and efficient processes must be put in place to support both configuration management and application release teams who, in many cases are responsible for hundreds, if not thousands of deployments a week.
- Organizational
 - In a DevOps world, the configuration management team is typically part of the operations organization and the release team is often part of the application development organization. More and more we see the release management team as part of a centralized organization responsible for coordinating and executing all application deployments. In either case, it is imperative that all teams work together in a collaborative and coordinated way. Application release teams rely on the configuration management team to ensure the underlying environments are created and configured correctly so that application deployments do not fail when deployed on top of the infrastructure.

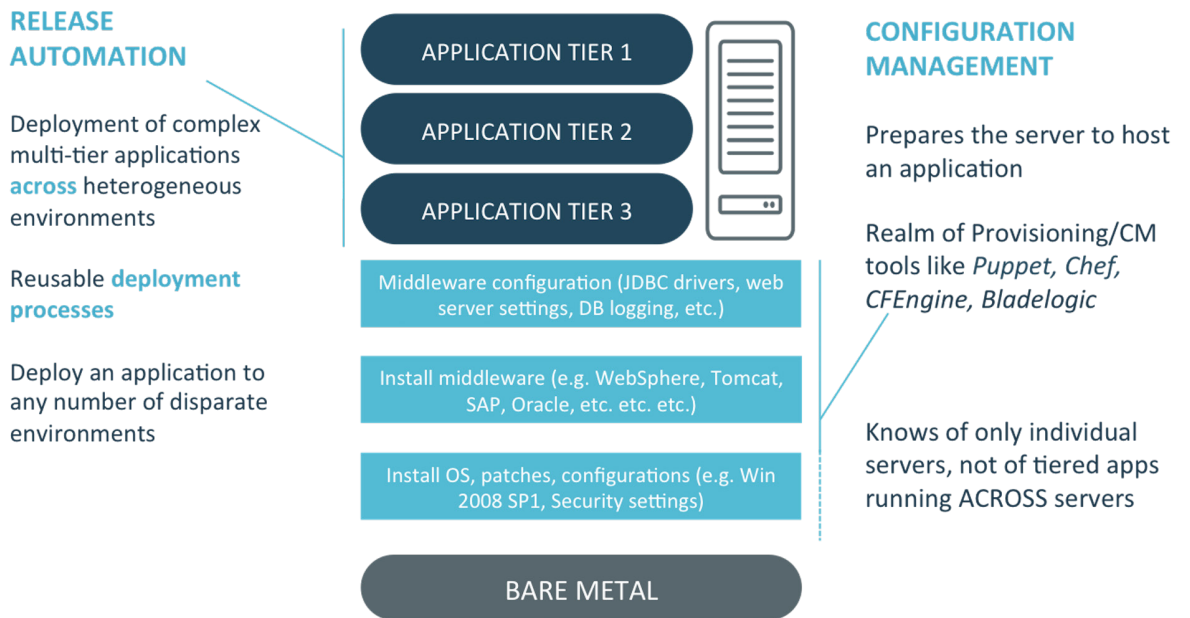
- Technical
 - As mentioned, configuration management and release automation solutions must seamlessly work together to support hundreds or thousands of deployments a week. There is an intricate dance of changes that need to be coordinated at both the infrastructure and application levels in order to achieve a successful deployment. As such there must be a tight integration between the solutions to ensure alignment so that deployments are not delayed or missed entirely.

Leveraging the Strengths of Each Solution

Knowing that these three areas need to be addressed, what might a solution look like that helps achieve these lofty goals? Both teams are very busy and each has different skill sets. An effective solution must leverage the strengths of each and provide more value as the sum of the parts.

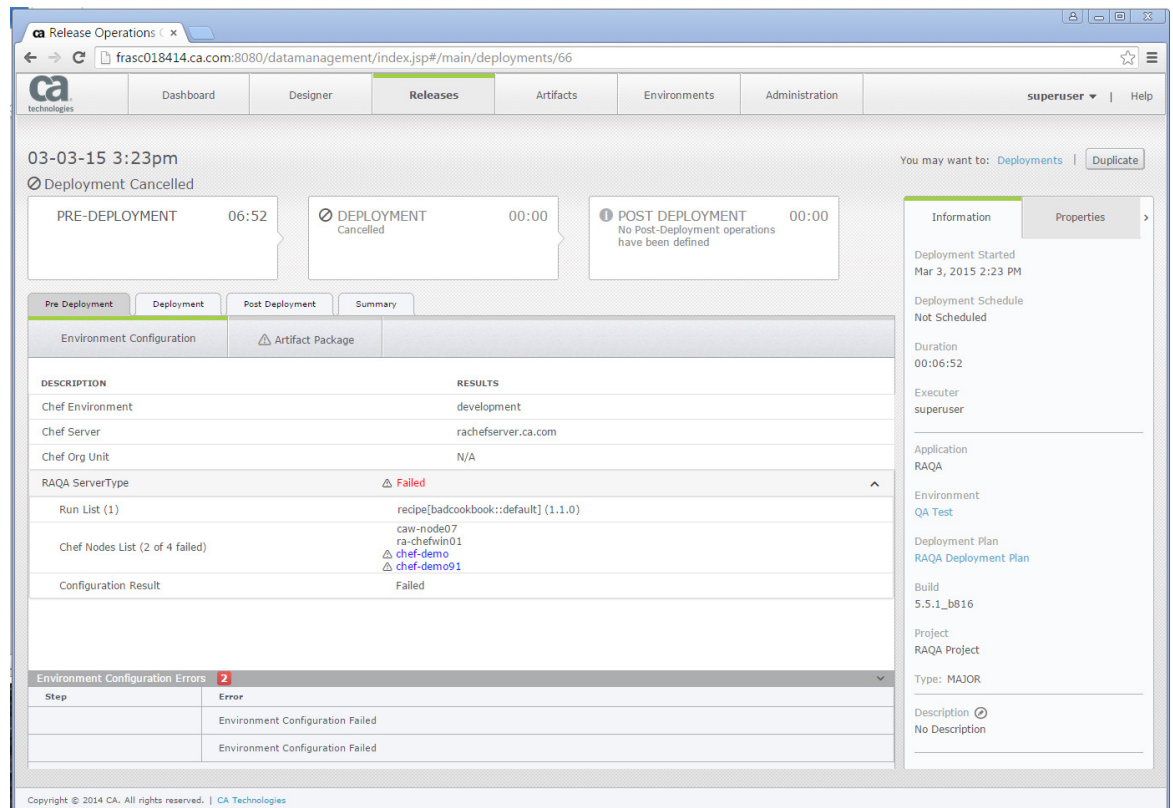
Release managers and release administrators are adept at orchestrating the deployment of an application across each stage of the release pipeline—development through to production—but most are not adept in designing and writing the types of scripts required by the typical configuration management solution. Conversely, those responsible for configuration management are typically very good at understanding and implementing scripts but often lack the skills and knowledge needed to shepherd an application across the stages of the SDLC.

Figure 2.



So, one thing that must be done is to give them a way to work together. Most ARA solutions typically include an Author/User model; one role for designing the release processes and workflows, the Author, and another role for managing the release process, the User. This same model can be adopted for configurations as well. The configuration management team authors the configuration policies and definitions as agreed to by the teams, and in turn the release management team incorporates them into the release process seamlessly. Thereby achieving both teams' goals and providing a seamless, automated, repeatable, auditable and scalable solution.

Figure 3.
Pre-Deployment
Configuration

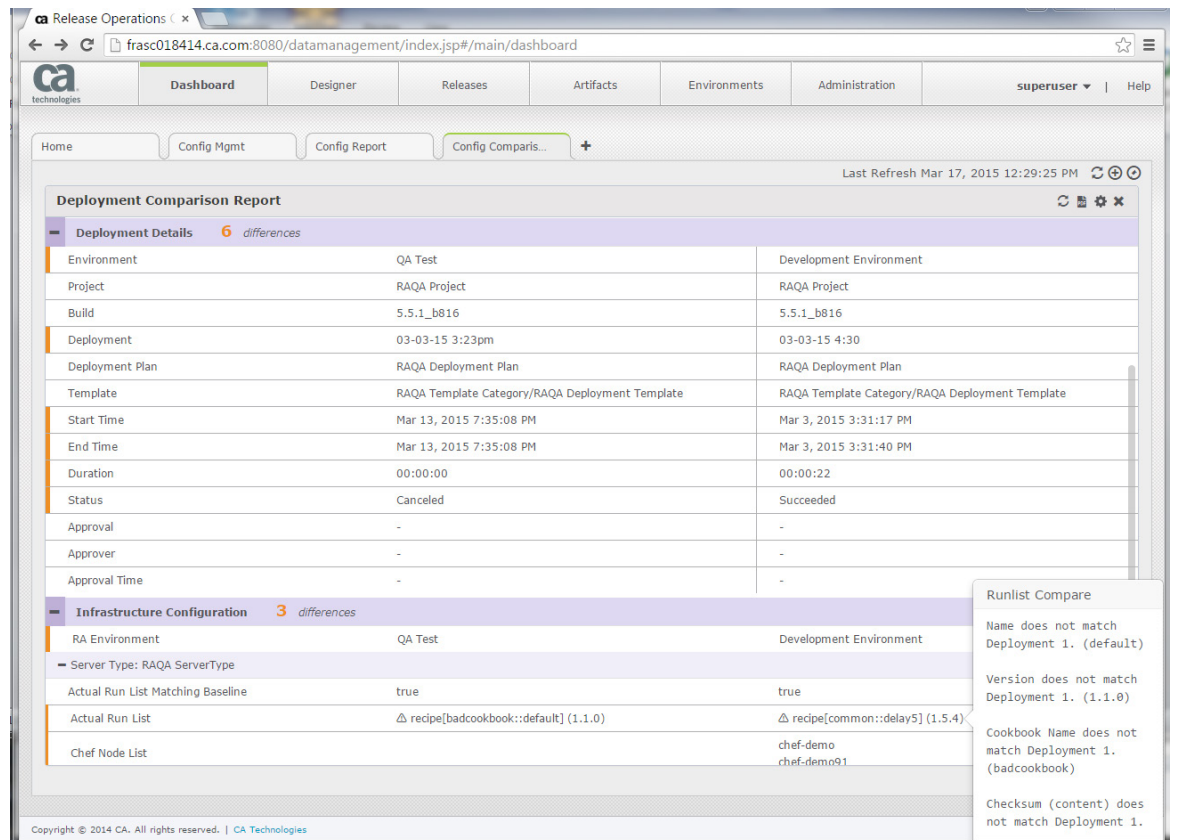


As such, an effective configuration management and release automation integration should include:

- Seamless/OOB integration to configuration management solutions that enables access to multiple instances of a configuration management product(s).
 - For example, Define, Login and use multiple Chef instances and security organizations from within the UI of release automation.
- Simple set-up and configuration
 - Provide Server Type and Environment mapping to the configuration management solution's role/type and environment mapping.
 - Provide a policy mapping capability for the configuration management vendor's policy definitions into the release automation solution.
 - E.g Chef Recipe and Cookbook version selection by environment for deployments as part of the SDLC
 - Ability to bootstrap/install a Chef client on a node within the release automation solutions UI.
- Application/deployment baselines
 - Ability to associate the configuration management policy definitions by version and environment to an application/deployment baseline within the release automation solution.
 - Ability to understand the changes to those policies that might break the expected application deployment.
 - Ability to establish a baseline in order to track configuration drift and provide reporting across applications, environments and deployments.

- Environment configuration stage as part of the release process
 - Natively incorporate calls to the configuration management solution to verify and or update the underlying nodes baselines as part of the release deployment process.
 - Report the status of the configuration management solution’s activity as part of the release automation deployment status, including success or failure.
 - Have the ability to quickly execute the configuration management stage. The solution should support multi-threaded calls to the underlying environment in order to execute the configuration process for each environment in parallel. A large volume of nodes can take a long time to process within a deployment window.
- Reporting and analysis
 - Deployment comparison reports
 - Highlight differences in policies, including version changes as part of application and deployment drift reporting.
 - Configuration drift — key performance indicators
 - Highlight the number and type of configuration changes over time spanning individual or composite views of applications, releases and environments.
 - Detailed views of infrastructure policy changes that have occurred

Figure 4.
Deployment
Comparison Report



Section 3:

Conclusions

Configuration management tools provide a critical component of an overall enterprise application release automation strategy but, as we've seen, do not replace the need for an Application Release Automation solution itself. The use of these tools will help reduce the risk of failure of application deployments due to configuration errors but they do have their limits in terms of scale and functionality when it comes to application release automation for enterprise environments.

The integration of configuration management and application release automation help make infrastructure configuration management an easy and seamless exercise for DevOps and application release teams. Maintaining consistent environments is critical to ensuring that what you develop and test is the same as what you deliver to customers. Application delivery teams need validation that systems are in a known state and are aligned to their application deployment baselines. Configuration management solutions help to define and maintain that known state.

CA Release Automation enables organizations to incorporate configuration management seamlessly into the deployment workflow where configurations can be crosschecked against the application and deployment baselines. Any issues discovered concerning configuration drift across middleware, applications, and environments can be corrected immediately, tracked and reported on within a single user interface.

Section 4:

About the Authors

Tim Mueting, Director, Product Marketing

Tim Mueting is responsible for product marketing for CA Technologies Application Delivery solutions focusing on Continuous Delivery and Release Automation. Tim has more than 20 years experience in delivering and marketing large-scale enterprise IT solutions as a pre-sales consultant, product manager and product marketing manager and has been a regular speaker at industry events on a variety of topics including Enterprise Management, Virtualization and Cloud Computing.

Paul Peterson, Senior Director, Product Management

Paul Peterson is CA Technologies Senior Director of Product Management with responsibilities in the CA DevOps product line. Paul is currently responsible for driving high-value use cases in the Environment on Demand space with CA Release Automation via tight integrations between CA Release Automation, Infrastructure Configuration Management solutions like Chef and Puppet and Cloud Management Platforms. He has been in the IT industry for over 20 years in both public and private sectors; managing performance, capacity, configuration and availability in operations teams as well as delivering services to customers as a managed service provider and as software product manager.



Connect with CA Technologies at ca.com



CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate – across mobile, private and public cloud, distributed and mainframe environments. Learn more at ca.com.