

Résumé :

Jenkins/Hudson est déjà fortement présent dans le domaine du développement et de l'intégration continue (CI). Le présent document répond aux deux questions les plus fréquemment posées par les clients habitués à travailler avec Jenkins : « Pourquoi ne pas utiliser Jenkins dans le cadre d'un scénario DevOps », et « En quoi Atomic ARA est-il différent ? ». L'utilisation d'Atomic ARA en complément des installations Jenkins existantes et la création d'un processus de déploiement de bout en bout seront également abordées.

Quels sont les points essentiels à prendre en compte lors de la recherche d'une solution ARA / DevOps

DevOps n'est ni un marché ni une méthodologie, c'est une philosophie. DevOps épouse l'idée selon laquelle les entreprises du secteur informatique doivent faire de la flexibilité une priorité fondamentale. Les équipes de développement et d'exploitation collaborent avec comme seul objectif une livraison flexible pour la production. Le DevOps est né de cet objectif d'intégration et de livraison continue. Son succès dépend de l'utilisation d'outils permettant d'automatiser et de normaliser la configuration, la libération et le contrôle des processus liés au déploiement. Les solutions ARA visent à répondre aux besoins des entreprises qui adoptent cette philosophie DevOps.

Qu'est-ce qu'une solution ARA ou non-ARA ? Le rapport Gartner de Mai 2013 « Know the Application Release Automation Vendor Landscape to Shortlist the Best Vendors for your Organization » (connaître le marché des solutions ARA afin de sélectionner les profils les plus adaptés à votre entreprise), identifie les quatre capacités fondamentales devant être présentes dans une solution ARA :

1. Automatisation
2. Modélisation d'environnement
3. Gestion de workflow
4. Adaptateurs d'intégration

Dans ce même rapport, Gartner définit également les quatre fonctionnalités fondamentales devant être incluses dans une solution ARA :

- Le déploiement de données réelles, de codes d'application ou d'artéfacts
- Le déploiement de paramètres de configuration spécifiques pour chaque environnement (développement, test, assurance de qualité, mise en place et production)
- La conception de workflow de processus pour les tâches automatisées, les étapes attribuées au personnel ou les deux
- La modélisation d'environnement et/ou de binaires (intergiciels, base de données et serveurs d'application)

Automic définit cinq compétences essentielles devant être traitées dans tout scénario ARA :

1. **Packaging** – création d'un ensemble d'applications et de configurations devant être déployées simultanément
2. **Analyse de dépendance** – modélisation intégrale des dépendances entre applications et composants
3. **Modèle de promotion** – définition d'un chemin pour la livraison des packages testés vers les environnements plus critiques
4. **Modèle de déploiement** – utilisation d'étapes pour l'installation du contenu et des configurations du package sur les environnements d'exploitation
5. **Conformité et visibilité** – respect des processus, validation des applications et des configurations déployées

Ces cinq compétences ARA sont implémentées sous la forme de trois capacités : les Quoi, Comment et Où d'un déploiement. Les trois capacités d'Automic ARA correspondent parfaitement aux fonctionnalités essentielles d'une solution ARA définies par Gartner.


- **Packages** – Quelle version de l'application et des configurations liées va-t-on déployer ?
Déploiement de données réelles, de codes d'application ou d'artéfacts

- **Workflows** – Comment les étapes d'approbation, de déploiement et de validation seront-elles définies pour cette application ?
Conception de workflow de processus pour les tâches automatisées, les étapes attribuées au personnel ou les deux

- **Modèles** – Où l'application sera-t-elle déployée et quelles configurations spécifiques à l'environnement devront être solutionnées ?
Modélisation d'environnement et/ou de binaires
Déploiement de paramètres de configuration spécifiques pour chaque environnement individuel

Pourquoi ne pas utiliser Jenkins dans le cadre d'un scénario DevOps ?

Jenkins est un outil d'intégration continue (CI) comportant des capacités d'automatisation servant de base aux différentes versions de logiciels, aux développements basiques et aux sandboxes de développement. Jenkins permet de rassembler tous les types de scripts et de sélectionner un environnement cible unique. Certains administrateurs ont donc essayé d'étendre son utilisation au-delà de l'équipe de développement afin de répondre aux besoins ARA grandissants et de plus en plus complexes à l'échelle de l'entreprise. La plupart du temps, scripter des extensions au sein de Jenkins peut s'avérer être une étape initiale dans le processus complet ARA (peut-être déjà disponible en interne). Malheureusement, Jenkins n'est pas conçu pour être une solution ARA et ne bénéficie donc pas de la plupart des compétences et des capacités ARA fondamentales dont les équipes DevOps



ont besoin. Lorsqu'on le compare aux fonctions définies par Gartner pour ARA, Jenkins présente des lacunes dans les domaines de gestion du workflow, de modélisation d'environnement, de conception de workflow de processus et de déploiement de paramètres de configuration spécifiques à chaque environnement individuel.

Jenkins est capable d'exécuter des scripts, ce qui lui permet d'exceller dans le codage et le transfert automatique de logiciels vers un sandbox de développement. ARA offre bien plus que le transfert automatique d'un logiciel vers un sandbox de développement. Les solutions ARA utilisent des structures pour gérer et s'adapter aux déploiements d'applications complexes sur de multiples environnements, environnements comportant souvent différentes configurations et propriétés à gérer. Afin que Jenkins se rapproche de ce que la solution ARA a à offrir, de nombreux scripts personnalisés et de nombreuses maintenances desdits scripts sont requis. Il n'est pas rare de compter, au sein d'une équipe, un ou deux « super-héros de l'automatisation » ne pouvant rien faire d'autre que de s'occuper du système de déploiement basé sur le script Jenkins. Le problème rencontré par le système de déploiement personnalisé comporte deux volets :

1. Les scripteurs Jenkins sont généralement des développeurs ne travaillant pas en collaboration dans le but de créer une solution globale.
2. Les régulations portant sur les problèmes de ségrégation des tâches sont transgressées. Les systèmes fortement modifiés par des scripts ont également tendance à présenter des faiblesses en matière d'implémentation de normalisations permettant de gagner en vérifiabilité et visibilité sur les activités survenant pendant les déploiements.

Le fait d'offrir des mises à jour d'applications propres et régulières devient de plus en plus crucial pour les entreprises. Un déploiement malencontreux représente non seulement un risque pour les entreprises, mais également un possible embarras public entraînant une perte de confiance de la clientèle et, bien sûr, des pertes financières.

La comparaison des capacités Jenkins avec la norme Gartner ARA a permis de démontrer que Jenkins présente des lacunes dans les domaines de gestion du workflow, de modélisation d'environnement, de conception de workflow de processus et de déploiement de paramètres de configuration spécifiques à chaque environnement individuel.

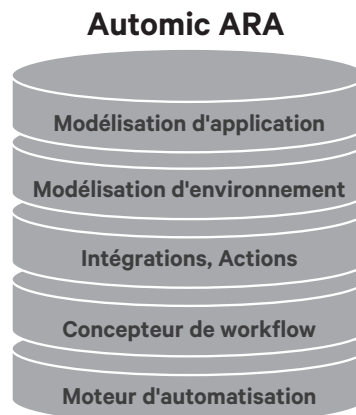
En quoi Automic ARA répond-il aux quatre capacités principales Gartner ?

Automatisation

Le premier élément à prendre en compte dans une solution ARA est un moteur de gestion. L'élément fondamental d'Automic ARA est la plateforme d'automatisation Automic ONE. Automic, anciennement UC4, a plus de 20 ans d'expérience dans le domaine de l'automatisation. Il est doté d'un moteur d'automatisation (Automation Engine) capable de traiter quotidiennement des millions de tâches, d'une logique conditionnelle et de capacités de planification, de files d'attente et de réservations d'environnement qui s'étendent dans le domaine ARA. La plateforme d'automatisation ONE est un outil extrêmement paramétrable pouvant s'adapter à tout scénario d'automatisation et rapide à déployer avec l'aide d'un agent par serveur permettant d'activer toutes ses capacités d'automatisation.

Gestion du workflow et adaptateurs d'intégration

La plateforme d'automatisation Automic comporte deux autres capacités essentielles d'une solution ARA : Intégration et gestion du workflow



La conception d'Automic ARA intègre des capacités d'intégration et de définition du workflow à tous les systèmes d'une entreprise à partir d'un seul et même lieu. Un AE commande tout, mais il est multi-tenant et autorise jusqu'à 9 999 clients différents au sein de l'entreprise.

Tous les processus d'automatisation Automic sont implémentés en tant que workflows. Les workflows Automic ARA sont composés à l'aide d'une interface glisser-déposer simple. Chaque workflow comprend une série d'Actions (pouvant également être appelées étapes d'automatisation ou adaptateurs d'intégration). Chaque action au sein du workflow Automic est conçue pour exécuter une fonction discrète s'interfaçant avec tout système d'exploitation ou toute application. La figure 1 est un exemple montrant un workflow simple composé de quatre actions.

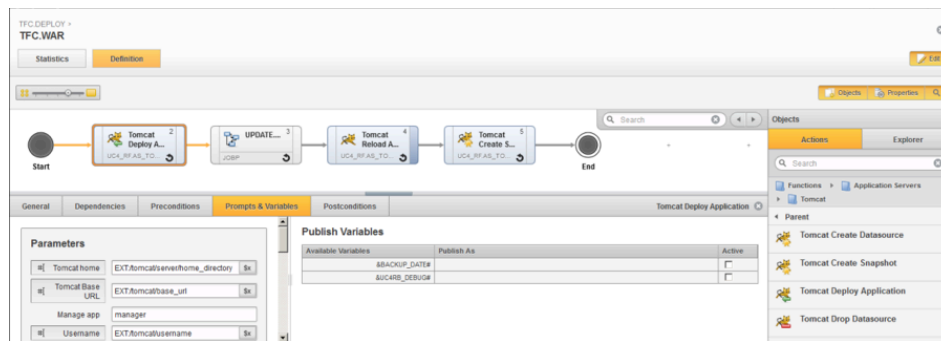


Figure 1 : L'éditeur de workflow Automic ARA

Automic ARA comprend plus de 500 actions clés en main, chacune comprenant un point d'intégration défini permettant de tout faire, du déplacement d'un fichier au redémarrage d'un serveur, en passant par la mise à jour d'un incident dans le système d'assistance et par le déploiement d'une mise à jour du schéma. En plus de ces actions prédéfinies, des actions personnalisées peuvent également être intégrées à n'importe quel système utilisant un Service Web, un API ou une interface de ligne de commande. Les actions doivent seulement être créées une fois. Elles sont ensuite disponibles pour être réutilisées dans n'importe quel workflow et pour n'importe quel client.

Automic ARA est un maître de l'orchestration – une implémentation Automic ARA n'est pas destinée à supprimer et à remplacer. Elle est destinée à organiser les processus entre différents systèmes et différentes équipes au sein d'un hub d'automatisation complet impactant n'importe quel nombre d'applications et de systèmes opérationnels.


Modélisation d'environnement

L'une des plus grandes préoccupations de l'ARA est la Modélisation. L'implémentation de la modélisation dans l'Automic ARA contrôle le plus grand défi de chaque implémentation : les configurations.

Certaines configurations ou des configurations incorrectes peuvent rendre les déploiements compliqués à gérer et difficiles à déboguer. La complexité des applications est souvent reflétée par toutes les configurations devant être gérées. Les serveurs Web comportent souvent des dizaines et même des centaines de configurations spécifiques devant être gérées à chaque fois qu'une application est déployée. Une solution ARA devrait atténuer les préoccupations concernant la gestion des configurations et faire gagner de nombreuses heures de travail sur console et/ou de mise à jour et de jonglage entre les différents fichiers de propriétés.

La modélisation Automic est divisée en deux zones logiques : la modélisation d'application et la modélisation d'environnement.

Les modèles d'application définissent à quoi l'application ressemble, ceci incluant la pile technologique,



les différents composants et leur agencement, ou encore les configurations générales des applications correspondantes.

Les modèles d'environnement définissent les cibles de déploiement de la pile de serveur (physique, VM, Cloud) et les configurations spécifiques à l'environnement.

Diviser le modèle en deux aspects permet de contrôler la configuration granulaire. Personne ne veut appliquer la même configuration à 100 objets si toutes ces configurations sont exactement similaires. Alors, si une configuration est la même sur toutes les instances d'un déploiement dans Atomic ARA, il la définira comme une configuration de niveau d'application dans le modèle d'application. Environnement - les configurations sensibles, telles que les noms de bases de données, les ports et chaînes de définition du serveur Web, sont enregistrées séparément dans le modèle d'environnement Atomic ARA.

Lors de l'exécution, un workflow Atomic prendra les configurations du modèle d'application et les configurations du modèle d'environnement afin de déployer l'application de manière dynamique. Si une nouvelle configuration ou un nouveau serveur est ajouté à l'équation, seule une mise à jour du modèle est nécessaire. Le workflow et le package de déploiement restent les mêmes.

Quelles sont les fonctionnalités supplémentaires d'Atomic ARA permettant un déploiement fiable et répétable ?

Packaging et promotion

Atomic ARA ne déploie pas de nouvelles versions, mais plutôt des packages. Le package représente la charge devant être déployée, ce qui ne signifie pas nécessairement une nouvelle version. Il peut par exemple s'agir d'un patch autonome fourni par un vendeur.

Dans un scénario d'intégration CI, un package en Atomic ARA est souvent directement lié à une nouvelle version. On définit dans le modèle d'application le chemin de promotion de la nouvelle version/ du déploiement ainsi que les individus ou groupes ayant le droit de déclencher une promotion sur ledit chemin. Les chemins de promotion et les permissions utilisateur peuvent être définis et configurés individuellement pour chaque application.

La Figure 2 illustre comment Jenkins poursuit la gestion des nouvelles versions en CI. Lorsque Jenkins termine une nouvelle version, un Package de déploiement Atomic ARA est créé. Le package de déploiement est ensuite déplacé vers un chemin de promotion d'environnement cible créé dans le modèle d'application. Dans un exemple CI, la nouvelle version déclenchée par une vérification de fichier par le développeur serait automatiquement déployée dans un sandbox de développement. La promotion du package de déploiement sur l'ensemble des autres environnements serait déterminée selon un ensemble configurable de permissions de groupes et de rôles. Les approbations intégrées Atomic et les approbations provenant de systèmes de tickets externes peuvent également être évaluées avant autorisation du déploiement.

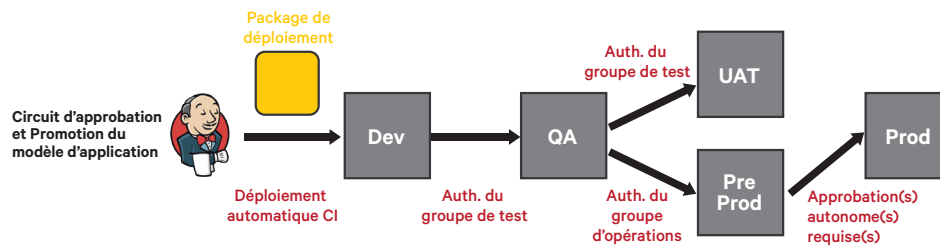


Figure 2 : Exemple d'intégration de Jenkins dans un modèle de promotion d'application

Visibilité et conformité

Automatic ARA permet d'expliquer facilement au public et aux équipes de gestion la manière dont les processus de déploiement d'application sont implémentés et comment la conformité est assurée pour les déploiements, avec un suivi graphique clair des exécutions. De plus, ces schémas doivent répondre aux attentes des administrateurs et des développeurs en offrant toutes les fonctionnalités profondes d'analyse et de recherche nécessaires à l'enregistrement de l'ensemble des exécutions ou comme assistance pour le débogage des erreurs.

La figure 3 montre un exemple de Gestion Automatic ARA en temps réel. Un suivi graphique de l'exécution du workflow incluant des rapports écrits est généré et enregistré pour chaque exécution du déploiement. Un historique de déploiement de package est également conservé pour le suivi des modifications.

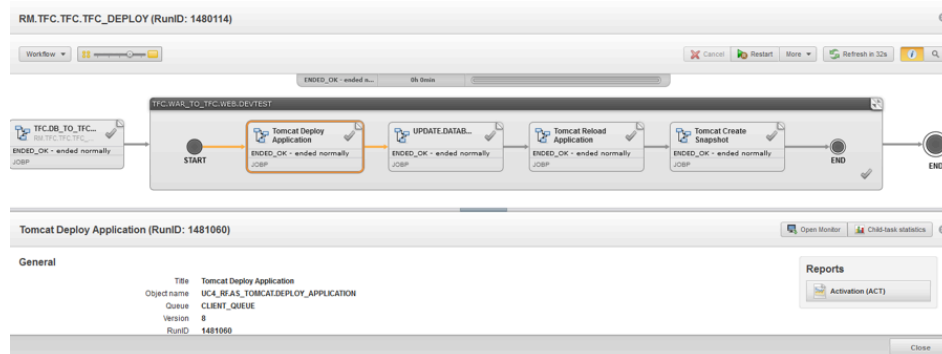


Figure 3 : Exemple de surveillance d'exécution du workflow Automatic ARA

Les **Aperçus d'environnement**, illustrés dans la Figure 4, permettent le suivi de chaque configuration et de chaque composant déployé. Comprendre rapidement et facilement l'historique et le contenu d'un package permet de gérer la **dérive de configuration** de manière beaucoup plus simple. Vous voulez savoir si un package a déjà été déployé dans un environnement spécifique ? Consultez son **historique de déploiement**. Vous voulez comprendre pourquoi un environnement qui fonctionnait normalement avant de déjeuner est en panne complète à votre retour à 14h00 ? Utilisez une **comparaison instantanée** pour voir si des mises à jour non-autorisées/manuelles ont été appliquées à des configurations ou fichiers.

Snapshot Content Report of TFC.WEB.DEVTEST

Environment List > TFC.DEVTEST > Snapshots > TFC (2014-01-30) > TFC.WAR > TFC.WEB.DEVTEST

General

Snapshot Content Report

Tomcat server snapshot

Configuration

- Server configuration (conf/server.xml)
 - Global JNDI Resources
 - UserDatabase (auth="Container"; description="User database that can be updated and saved"; factory="org.apache.catalina.users.MemoryUserDatabaseFactory"; id="UserDatabase"; pathname="conf/tomcat-users.xml"; type="org.apache.catalina.UserDatabase";)
 - Context parameters
- Global context (conf/context.xml)
 - Global JNDI Resources
 - Context parameters
- Global deployment descriptor (conf/web.xml)
 - Global servlet
 - jsp (fork="false", id="jsp", mapping=".jsp,*.jspx"; servlet-class="org.apache.jasper.servlet.JspServlet"; xpoweredBy="false";)
 - default (debug="0"; id="default"; listings="false"; mapping="/*"; servlet-class="org.apache.catalina.servlets.DefaultServlet";)
 - Global context
- Other configuration files
 - conf/catalina.policy (size= 12.4 kB, last modified = 12/4/12 3:54:00 AM EST)
 - conf/catalina.properties (size= 6.5 kB, last modified = 12/4/12 3:54:00 AM EST)
 - conf/logging.properties (size= 3.4 kB, last modified = 12/4/12 3:54:00 AM EST)
 - conf/tomcat-users.xml (size= 1.7 kB, last modified = 1/6/13 3:01:08 AM EST)

Figure 4 : Exemple d'une configuration instantanée détaillée

Aujourd'hui, de quelle façon puis-je utiliser Jenkins ainsi que les outils que je possède avec Automic ARA ?

Avec Automic ARA, il n'y a aucune raison de supprimer et de remplacer un moteur de construction existant. La figure 5 illustre qu'Automic peut parfaitement s'intégrer à Jenkins afin de gérer les étapes de post-production d'un déploiement intégré. De ce fait, tout logiciel actuellement utilisé en interne (des systèmes de tickets aux suites de test en passant par les configurations d'infrastructure) peut être intégré à un workflow Automic ARA, comme montre la figure 6.

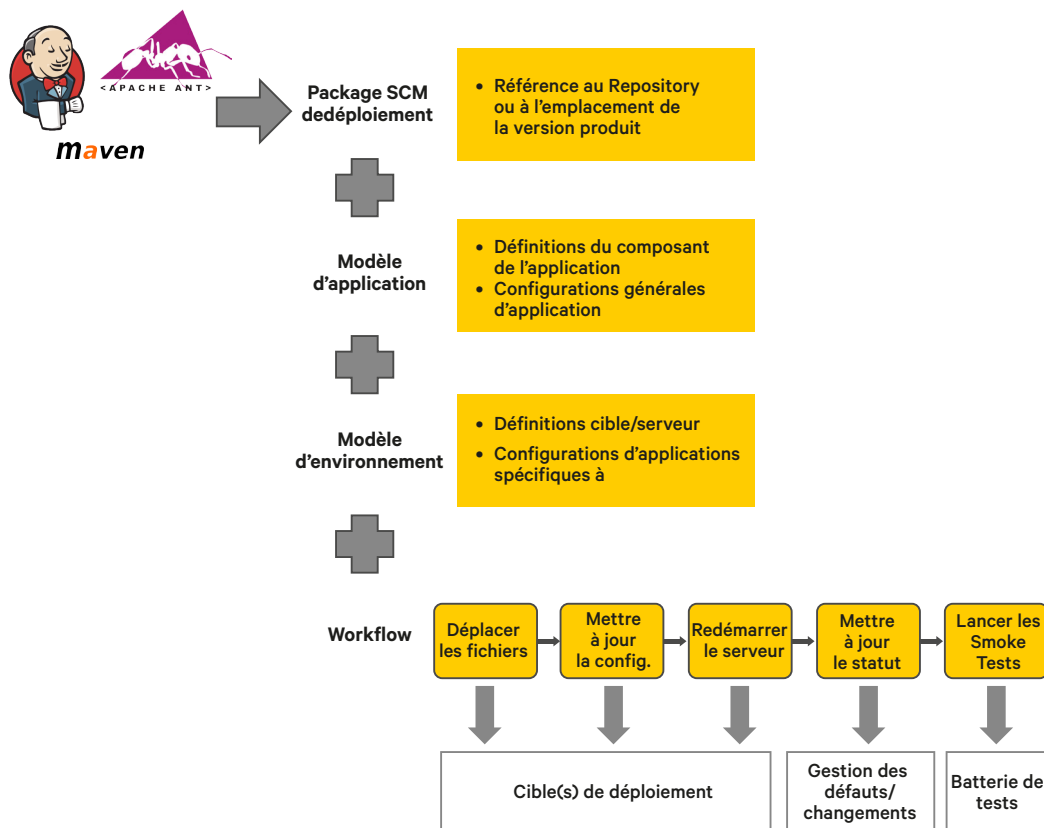


Figure 5 : Exemple de la façon dont Jenkins s'intègre à la structure Automic ARA

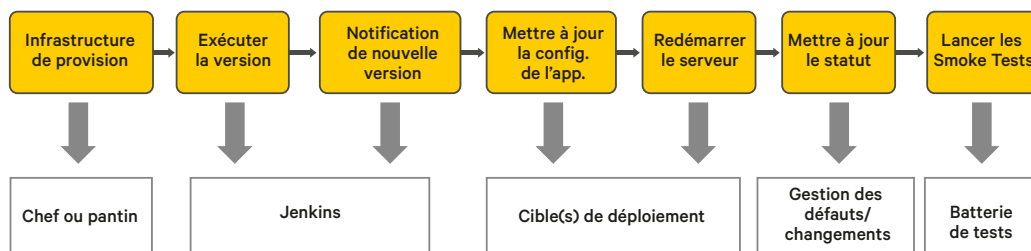


Figure 6 : Workflow ARA intégral conceptuel

Conclusion

Automatic fonctionne correctement avec les capacités d'intégration continue de Jenkins. Automatic étant une plateforme indépendante, elle fonctionne aussi bien avec TFS, Maven et Ant qu'avec tous les autres systèmes de mise à jour actuellement utilisés. Automatic ARA brille dans l'orchestration et l'implémentation de processus complexes actuellement traités par des systèmes de scripting improvisés ou des configurations manuelles. Les clients qui utilisent et apprécient déjà Jenkins peuvent bénéficier pleinement de ses capacités CI fondamentales grâce un processus Automatic ARA global intégré permettant de définir et de gérer de manière centralisée la modélisation, les intégrations et le workflow pour un résultat fiable, répétable et vérifiable.