

Rapport Projet Tuteuré Runderck

HEPPNER Tristan / ROYER Grégory
TORRES Yanis / AISSI Ayoub

Avril 2020

Table des matières

1	Remerciements	5
2	Introduction	6
3	Sujet de la soutenance	6
4	L'automatisation	7
5	Principes d'automatisations	7
6	Notions Importantes	8
6.1	L'expression cron	8
6.2	L'intégration continue	9
7	Cron	11
7.1	Présentation	11
7.2	Historique et versions	12
7.3	Contexte d'utilisation	12
7.4	Fonctionnement	12
7.5	Fonctionnalités	12
7.6	Installation	13
7.7	Conclusions	13
8	Jenkins	13
8.1	Présentation	14
8.2	Historique et versions	14
8.3	Contexte d'utilisation	14
8.4	Fonctionnement	15
8.5	Fonctionnalités	15
8.6	Installation	15
8.7	Conclusions	17
9	Buildbot	17
9.1	Présentation	17
9.2	Historique et versions	18
9.3	Contexte d'utilisation	18
9.4	Fonctionnement	18
9.5	Fonctionnalités	18
9.6	Installation	19
9.7	Conclusions	19

10 Ansible Tower	19
10.1 Présentation	19
10.2 Historique et versions	20
10.3 Contexte d'utilisation	20
10.4 Fonctionnement	20
10.5 Fonctionnalités	21
10.6 Installation	21
10.7 Conclusions	22
11 JobScheduler	22
11.1 Présentation	22
11.2 Historique et versions	23
11.3 Contexte d'utilisation	23
11.4 Fonctionnement	23
11.5 Fonctionnalités	24
11.6 Installation	25
11.7 Conclusions	26
12 Rundeck	26
12.1 Présentation	26
12.2 Historique et versions	27
12.3 Contexte d'utilisation	27
12.4 Fonctionnement	27
12.5 Fonctionnalités	28
12.6 Installation	28
12.7 Conclusions	28
13 Tableaux synthétiques	29
14 Mise en place de Rundeck	31
14.1 Environnement de travail	31
14.2 Installation	32
14.2.1 Exigences système	32
14.2.2 Windows	33
14.2.3 Linux	33
14.3 Configuration	33
14.3.1 Arborescence	33
14.3.2 Interface de connexion	37
14.3.3 Création d'un projet	38
14.3.4 Configuration du stockage	40
14.3.5 Définition des nodes	41

14.3.6 Clés SSH	43
14.3.7 Définition des jobs	43
14.3.8 Gestions des utilisateurs	45
15 Rundeck et Ansible	45
16 Authentification via un annuaire LDAP	46
17 Conclusions Générales	47
18 Bibliographie	49
19 Annexes	50

1 Remerciements

Nous remercions notre tuteur de projet tutoré, Mr Fabien PASCALE, responsable du service informatique du laboratoire de physique et chimie théoriques (unité mixte de recherche 7019) de l'université de Lorraine, pour nous avoir fait découvrir Rundeck, ses enjeux ainsi que l'ampleur et l'importance de ce projet.

Merci à Monsieur Lucas Nussbaum, Debian Project Leader (DPL), maître de conférences à l'Université de Lorraine et chercheur auprès du laboratoire LORIA, pour nous avoir permis de nous enrichir intellectuellement.

Que cette licence ASRALL(Administration de Systèmes, Réseaux et Applications à base de Logiciels Libres) puisse perdurer dans le temps, et toujours apporter les connaissances indispensables dont nous, les élèves et futurs administrateurs, avons besoin. Un grand merci à Monsieur Philippe Dosch, enseignant et responsable de la licence, qui a su, quand il le fallait, nous écouter et nous donner les indications nous permettant de prendre la bonne direction.

2 Introduction

Depuis l'apparition de l'Homme sur la terre, ce dernier n'a cessé de trouver divers moyens pour améliorer la vie de ses semblables. Nous sommes, par nature, des êtres sociables et dotés d'intelligence.

L'homme, étant doté d'intelligence et de créativité, est capable de créer des systèmes permettant de simplifier sa propre vie mais aussi celles de ses pairs. Un type de système que l'Homme connaît depuis la nuit des temps est l'**automatisme**.

3 Sujet de la soutenance

La finalité de ce projet est de proposer une solution permettant d'automatiser la gestion d'un parc informatique grâce à une seule et unique machine.

Pour cela, nous allons analyser les solutions existantes :

- Cron
- Jenkins
- Buildbot
- Ansible Tower
- JobScheduler
- Rundeck

L'objectif de ce projet, est quant à lui de démontrer l'utilité de la solution Rundeck.

A la fin du projet, nous disposerons :

- Tutoriel de mise en place de Rundeck
- Comparaison des solutions existantes

Tuteur du projet :

PASCALE Fabien

fabien.pascale@univ-lorraine.fr ;

Étudiants :

AISSI Ayoub

aa.w-a@hotmail.fr

HEPPNER Tristan

tristan.heppner@outlook.com

ROYER Grégory

gregory.royer@hotmail.com

TORRES Yanis

yanis.torres@outlook.com

Avant de débiter l'analyse des solutions, une définition du mot **Automatisation** est de rigueur afin de pouvoir aborder, dans les meilleures conditions possibles, ce sujet.

Nous allons ensuite analyser chaque solution avec une présentation de la solution, un bref historique, son contexte d'utilisation, son fonctionnement, ses fonctionnalités ainsi qu'une bref conclusion sur cette solution.

Nous aborderons ensuite la solution choisit avec un tutoriel sur sa mise en place.

4 L'automatisation

Déf : L'automatique est une science qui traite de la modélisation, de l'analyse, de l'identification et de la commande des systèmes dynamiques.

Dans le monde que nous connaissons, on peut voir, sans s'en rendre compte, un quantité astronomique de systèmes automatiques. En effet, l'automatisation d'un système est une chose à laquelle l'homme ne cesse de penser.

L'automatisation est une science très ancienne datant de l'Antiquité. Un exemple bien connu datant de l'Antiquité Romaine est l'aqueduc, un système permettant de réguler automatiquement le niveau de l'eau.

Depuis, l'Homme n'a jamais cessé d'automatiser les éléments qui l'entoure. De célèbre invention ont vu le jour telles que le régulateur à boules de James Watt en 1769.

La science de l'automatique n'a renoncer d'évoluer et de gagner en popularité. L'automatisation, telle que nous l'a connaissons aujourd'hui est devenue pratiquement indispensable et ceux peu importe le domaine.

5 Principes d'automatisations

L'automatisation d'un parc informatique consiste à diminuer le nombre de tâches d'un administrateur en automatisant certains processus.

Un administrateur système est responsable de chaque machine présente dans un parc et par conséquent sur un réseau.

Le responsable du parc est régulièrement confronté à des tâches récurrentes notamment pour les mises à jours de systèmes et/ou logiciels, nettoyage ou encore sauvegarde des systèmes.

L'automatisation permet d'effectuer les tâches sans que l'administrateur ai besoin de faire ces tâches sur chaque machine, une à une.

En termes de tâches redondantes, on peut retrouver les sauvegardes du système, les arrêts automatique des machines sur une plage horaire définie, le

nettoyage des logs etc...

L'objectif de l'automatisation d'un ou plusieurs systèmes est de pouvoir faire gagner en productivité, en temps de travailler mais aussi de pouvoir réduire les coûts.

6 Notions Importantes

6.1 L'expression cron

Avant de passer à la suite, une explication plus complète sur l'expression cron est de mise.

L'expression cron est propre au programme de planification cron. Par ailleurs, cette expression se retrouve dans la majorité des logiciels de planification, ordonnanceurs ou encore orchestrateur.

Cette expression est la base de tous les programmes d'automatisation que nous connaissons.

Ci-dessous, un aperçu, de manière visuelle, de cette expression

```
# minute (0 - 59)
# hour (0 - 23)
# day of the month (1 - 31)
# month (1 - 12)
# day of the week (0 - 6) (Sunday to Saturday;
#                          7 is also Sunday on some systems)
# * * * * * command to execute
```

FIGURE 1 – Source : Wikipédia

Certaines versions de cron peuvent supporter certains standards qui ne sont, quant à eux, pas définis et supportés dans les versions de bases de cron. Ces standards sont utilisables dans les versions de cron sur FreeBSD.

Standards	Significations	Équivalent
@yearly	Une fois par an	0 0 1 1 *
@annually	Une fois par an (identique à @yearly)	0 0 1 1 *
@monthly	Une fois par mois	0 0 1 * *
@weekly	Une fois par semaine	0 0 * * 0
@daily	Une fois par jour	0 0 * * *
@midnight	Une fois par jour 'identique à @daily)	0 0 * * *
@hourly	Une fois par heure	0 * * * *
@every_minute	Une fois par minute	*/1 * * * *
@every_second	Une fois par seconde	N/A
@reboot	Une fois au démarrage de cron	N/A

6.2 L'intégration continue

Dans l'analyse des solutions, une notion importantes sera mentionnée : L'intégration continue.

L'intégration continue permet aux créateur de sites web, applications mobiles ou logiciels de réduire considérablement leur temps sur les compilations et tests de paquets.

Cela leur permet de consacrer plus temps au développement de leur application.

Ci-dessous, un représentation graphique de l'intégration continue

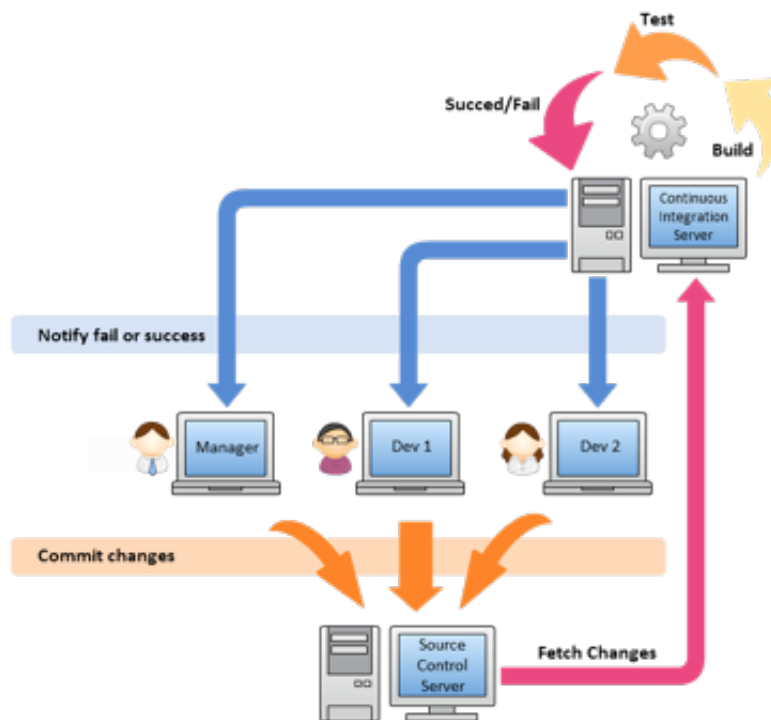


FIGURE 2 – Intégration continue

L'intégration s'accompagne de 2 autres notions : Le déploiement continu et l'exécution continue.

Ces 2 notions correspondent à un prolongement de l'intégration continue, c'est-à-dire :

L'exécution continue ou continuous integration (CI) en anglais comprends toutes les étapes de l'intégration mais en ajoutant les étapes de publication du paquet aux utilisateurs

Le déploiement continue ou continuous deployment (CD) en anglais contient les étapes des 2 notions précédente en ajoutant l'étape de l'installation automatique du paquets

Ci-dessous, un schéma détaillé de ces différentes notions.

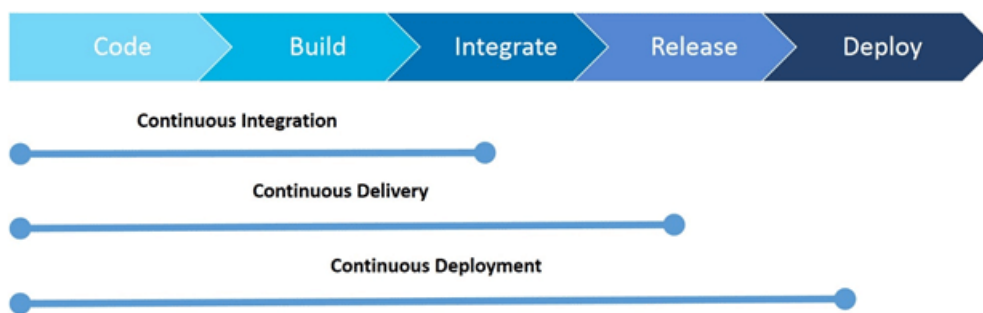


FIGURE 3 – Notions d'intégrations

7 Cron



FIGURE 4 – Logo de Crontab

7.1 Présentation

Cron est la troncation de crontab, lui-même la troncation de chrono table qui signifie « table de planification » source : Wikipédia

ATTENTION : Cron et Crontab sont 2 programmes différents mais ils ne vont pas l'un sans l'autre. En effet, cron est le planificateur de tâches tandis que Crontab est le programme qui permet d'éditer les fichiers de configurations de cron.

Les origines de cron se situent dans les systèmes Unix de Berkeley et AT&T et a été ensuite développé par Paul Vixie, ce qui fait de lui son auteur original. Le programme cron a été repris et développé par AT&T Bell Laboratories, aussi connu sous le nom de Nokia Bell Labs.

Cron est une fonctionnalité native des systèmes Unix. C'est le planificateur de tâches par défaut et intégré à chaque distribution Linux.

Cron est l'ancêtre et l'outil de planification le plus simple mais aussi le plus accessible pour tout le monde sur Linux

Cron est disponible sous les systèmes Linux, MacOS et FreeBSD sous une interface CLI.

Cron est un daemon (démon), c'est-à-dire que c'est un programme qui s'exécute en arrière plan

7.2 Historique et versions

Le programme cron apparaît pour la première en mai 1975, c'est-à-dire, bien avant les distribution Linux que l'on connaît actuellement.

Il est, à l'heure actuelle, maintenu par ces mêmes laboratoires, mais également par des mainteneurs indépendants.

7.3 Contexte d'utilisation

Étant un planificateur de tâches, cron est utilisé pour la planification quotidienne, hebdomadaire, mensuelle ou encore annuelle de tâches de maintenances.

7.4 Fonctionnement

Cron permet la planification et l'exécution automatique de tâches programmées à l'avance.

Ces tâches sont réglées dans un fichier contenant la commande à exécuter ainsi que l'heure d'exécution.

Le paramétrage d'une tâche est, quant à lui, très simple. Pour automatiser une tâche, il suffit d'écrire dans les fichiers de configuration de cron avec crontab, la commande à exécuter ainsi que l'heure et le jour à laquelle cron doit l'exécuter. Il est possible de recevoir une notification par mail sur le statut de la tâche, mais cela nécessite l'installation d'un client mail sur la machine.

Ces tâches sont exécutées par l'utilisateur root afin d'éviter les problèmes de droits

7.5 Fonctionnalités

Cron permet une planification de tâches pourrons s'exécuter quotidiennement, hebdomadairement, mensuellement ou annuellement.

Cette fréquence se règle avec des paramètres. Une notification par mail, facultative, peut être également paramétrée.

7.6 Installation

Cron ne requiert pas d'installation, il est, cependant, possible d'être installé si ce dernier n'est pas présent sur la machine.

Si l'installation de cron est nécessaire, il suffit de télécharger cron depuis les dépôts via la commande "sudo apt-get install cron".

Cette commande variera selon la distribution mais le paquet de cron porte, quant à lui, même nom.

7.7 Conclusions

Cron est un logiciel simple de planification et d'exécution automatique de tâches. Il est considéré comme le "père" des logiciels d'automatisations. Son utilisation simple séduit encore de grands nombres d'adeptes de l'automatisation.

Après 45 ans de service, Cron reste populaire de par sa simplicité et continue de séduire un grand nombre d'utilisateurs à travers le monde.

8 Jenkins



Jenkins

FIGURE 5 – Logo de Jenkins

8.1 Présentation

Jenkins est un logiciel Open-Source d'automatisation de processus. Sorti en 2011, Jenkins est un dérivé de son prédécesseur Hudson, il est présenté comme étant un fork¹ d'Hudson.

Jenkins est également un logiciel cross-platform disponible sur les systèmes UNIX, Windows et Mac OS.

Jenkins est hébergé sur la plate forme GitHub, ce qui permet à chaque utilisateur de Jenkins de contribuer à son développement.

Jenkins est en majeure partie, amélioré grâce à ses utilisateurs. Ces derniers peuvent participer au développement de Jenkins par la notification de bugs/-fonctionnalités manquantes ou en rejoignant une équipe de développement de Jenkins.

Rundeck est développé en JAVA et propose une interface WEB et requiert donc des dépendances JAVA.

La version "Entreprise" de Jenkins est CloudBees.

CloudBees fonctionne, quant à lui, en mode PaaS (Platform as a Service) et support le cycle de vie entier d'une application.

8.2 Historique et versions

Jenkins sort en 2011 et est également disponible sur la plate forme GitHub. Sa dernière version en date est la 2.204.1 datant du 18 décembre 2019.

Jenkins apparaît suite à la cessation d'Hudson par Oracle à la fondation Eclipse. Jenkins devint en 2011, le successeur d'Hudson.

Contrairement à Jenkins, Hudson est un framework d'intégration continue.

Jenkins est publié sous la licence MIT

Jenkins et Hudson ont été développés par le même développeur : Kohsuke Kawaguchi

8.3 Contexte d'utilisation

Jenkins est un logiciel d'intégration continue utilisé pour la compilation automatique de code source lors de la création de nouveaux programmes et/ou mises à jours .

1. fork : dérivé de ... (trad. litt. : fourchette)

8.4 Fonctionnement

Jenkins est une application WEB en JAVA et se déploie de manière autonome dans un conteneur WEB de type Tomcat.

Cependant, Jenkins peut fonctionner sur toutes les plates-formes pouvant exécuter une JVM (JAVA Virtual Machine) parmi lesquelles on retrouve Ubuntu/Debian, Windows ou même Mac OS à la seule conditions d'avoir les paquets Java correspondants sous peine de rencontrer des problèmes d'exécution ou d'affichage.

Jenkins exécute de manière continue la construction des projets ou "build".

Afin que Jenkins puisse permettre l'intégration continue, il exécute les directives de fabrication du logiciel (compilation, assemblage, tests, packaging, ...).

De la même manière que Rundeck, Jenkins peut exécuter des jobs, soit sur lui-même, soit sur des serveurs distants.

Les jobs constituant les builds peuvent être exécutés sur le serveur Jenkins lui-même ou répartis sur plusieurs machines distantes. Jenkins conserve l'historique de toutes les exécutions des jobs et peut notifier par mail ou par RSS en cas erreur.

8.5 Fonctionnalités

Comme tout les logiciels d'automatisations, Jenkins propose des fonctionnalités similaires à Rundeck, à savoir, exécuter des jobs à distance via SSH, le transfert de fichiers via SCP ou FTP.

A contrario de Rundeck, Jenkins peut très bien s'intégrer avec différents gestionnaires de versions, outils de fabrications ainsi que différents outils de suivi d'incidents/bogues.

Grâce à Jenkins des outils de contrôle de qualité de code mais aussi des tests d'intégrations, performance ou encore des tests de fonctionnement peuvent être effectués.

Jenkins permet également le transfert d'artefacts vers un répertoire.

8.6 Installation

Jenkins demande certaines exigences pour permettre son bon fonctionnement.

Minimum :

— 256 MB de mémoire RAM

- 1 GB d'espace disque (10 GB sont requis si Jenkins fonctionne dans un conteneur docker)

Recommandé :

- 1 GB ou plus de mémoire RAM
- 50 GB ou plus d'espace disque

Programmes :

- Java : OpenJDK JDK / JRE 8 - 64 bits ou OpenJDK JDK / JRE 11 - 64 bits
- Nivagatuer : Dernière version de Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, Microsoft Edge, Apple Safari

Un des avantages de Jenkins est qu'il peut être installé sur n'importe quel système d'exploitation. Il peut être installé, soit en programme seul, soit dans un conteneur Docker ou encore via les archives WAR.

Linux : Pour une installation de Jenkins hors Docker, il suffit de récupérer le répertoire de Jenkins via le terminal, qui contient la dernière version de celui-ci. Une fois ce répertoire récupéré, il suffit de l'installer via le terminal.

Les commandes varient selon la distribution utilisée.

Dés lors que l'installation est terminée, il est préférable de vérifier que Jenkins est bien démarré sur la machine, en utilisant `systemctl`.

Exemple d'installation sur Ubuntu/Debian :

- `wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -`
- `sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'`
- `sudo apt-get update`
- `sudo apt-get install jenkins`

Windows : L'installation de Jenkins sur Windows peut se faire de 2 manières :

- Sur un conteneur Docker (se référer à la page d'installation contenant les instructions)
- En téléchargeant l'installateur et en l'exécutant

macOS :

- `brew install jenkins-lts` : Installation de la dernière version
- `brew install jenkins-lts@VERSION` : Installation d'une version spécifique
- `brew services start jenkins-lts` : Démarrage des services de Jenkins
- `brew services restart jenkins-lts` : Redémarrage de Jenkins

— brew upgrade jenkins-lts : Mise à jour

Jenkins peut être installé de plusieurs manières pour n'importe quel système d'exploitation

8.7 Conclusions

Malgré d'énormes ressemblances avec Rundeck, Jenkins est un outil d'intégration continue pour la construction automatique de paquets.

9 Buildbot



FIGURE 6 – Logo de Buildbot

9.1 Présentation

Buildbot est un framework Open-Source d'intégration continue. Buildbot est un framework cross-platform et est disponible sur les systèmes Windows et POSIX (Portable Operating System Interface et X signifiant l'héritage UNIX).

Ce framework est présentée comme une alternative au projet de Mozilla : [Tindebox](#). Bien qu'il possède d'énormes similitude avec Jenkins, Buildbot est un framework d'intégration continue développé en Python.

Buildbot se démarque de ses concurrents de par son type ainsi qu'à sa légèreté mais par la possibilité de pouvoir l'installer sur des systèmes d'exploitations portables.

Buildbot est également disponible sur la plate forme GitHub et permet également aux utilisateurs de contribuer à son développement.

Buildbot est distribué sous la licence GPLv2.

Depuis Mars 2013, Buildbot support l'intégration SCM avec CVS, Bazaar, Darcs, Subversion, Perforce, Mercurial, Git, Monotone, Repo et Bitkeeper.

9.2 Historique et versions

La première version de Buildbot est sortie le 26 avril 2003. Sa dernière version est la 2.0.7 et est daté du 27 février 2020.

9.3 Contexte d'utilisation

Buildbot est utilisé dans le cadre de l'intégration continue, notamment dans le développement de site Web. De grandes entreprises et de nombreux logiciels utilisent Buildbot, parmi lesquels, on peut retrouver Mozilla, Chromium, WebKit et bien d'autres.

9.4 Fonctionnement

Le fonctionnement de Buildbot similaire à celui de Jenkins. Son fonctionnement diffère par rapport à Jenkins, de par le fait que Buildbot soit un framework. De la même manière que Jenkins, Buildbot propose une automatisation de la construction de paquets.

Buildbot exécute de manière continue la construction des projets ou "build".

Afin que Buildbot puisse permettre l'intégration continue, il exécute les directives de fabrication du logiciel (compilation, assemblage et tests).

De la même manière que Jenkins, Buildbot peut exécuter des jobs, soit sur lui-même, soit sur des serveurs distants.

Les jobs constituant les builds peuvent être exécutés sur le serveur Buildbot lui-même ou répartis sur plusieurs machines distantes. Buildbot conserve l'historique de toutes les exécutions des jobs et peut notifier par mail ou par RSS en cas d'erreur.

9.5 Fonctionnalités

Du point de vue de Buildbot, des applications comme CruiseControl ou Jenkins sont structurées de manière à être prêtes pour l'utilisation.

Les autres fonctionnalités sont, quant à elle, identiques à celles proposées par Jenkins.

Il permet aux développeurs d'envoyer leur code sur un serveur de contrôle de version sur lequel Buildbot ira rechercher les dernières versions du projet afin de le construire et effectuer des tests.

En cas d'erreur ou de réussite, Buildbot notifiera le/les développeur(s) en charge du projet.

9.6 Installation

Buildbot fonctionne sous le principe de master/worker. Il possède, cependant quelques exigences :

- Le master fonctionne avec la version 3.5 ou supérieur de Python
- Le worker fonctionne avec la version 2.7, 3.5 ou supérieur de Python

Tout comme Jenkins, Buildbot peut être installé de différentes manières :

- pip install buildbot et pip install buildbot-worker : Installation classique via la commande pip
- Après acquisition des Tarballs contenant Buildbot, il suffit de lancer les commandes `python setup.py build` et `python setup.py install`

9.7 Conclusions

Buildbot est un framework d'intégration continue présentant beaucoup de similitude avec Jenkins. Il reste néanmoins consacré à un certain domaine : le développement Web.

10 Ansible Tower



FIGURE 7 – Logo d'Ansible Tower

10.1 Présentation

Ansible Tower est un outil d'intégration et de déploiement d'application et/ou de composants divers (comme Nginx ou encore Icinga).

Il fournit une interface WEB dans laquelle, pourra être gérer Ansible, ainsi que de pouvoir, par la suite, créer, supprimer ou encore configurer de nouvelle entité

via cette interface.

Ansible Tower est un outil en Open Source, créé par Red Hat et Ansible, et est complètement compatible avec le cloud Amazon AWS (de par sa version Ansible), le déploiement de conteneur Docker qui permet un bon fonctionnement avec peu de ressources.

De nouvelles versions sont régulièrement disponibles allant de quelque semaine à quelques mois entre elles.

Il existe une autre version gratuite appelée AWX qui sert de version beta test pour la version originale TOWER. Cette version gratuite peut être utilisée en entreprise mais ne fournira pas autant de fonctionnalité que la version TOWER.

10.2 Historique et versions

Ansible Tower a été créé pour permettre à Ansible (la plateforme de configuration et de gestion d'ordinateurs) d'être utilisé plus facilement sur les équipements informatiques grâce aux API et console web.

10.3 Contexte d'utilisation

Ansible Tower a été créé pour permettre à Ansible (la plateforme de configuration et de gestion d'ordinateurs) d'être utilisé plus facilement sur les équipements informatiques grâce aux API et console web.

Ansible Tower est utilisé dans le même cadre d'utilisation que Rundeck. Ansible Tower est l'API de son logiciel principal : Ansible.

Ansible Tower est disposée également d'une console et d'un environnement WEB

10.4 Fonctionnement

Ansible Tower va permettre une meilleure sécurité, une adaptabilité et un champ d'application amélioré comparé à une utilisation de Ansible sans cet outil. Il permet un accès général au contrôle de la plateforme et gère les identifications SSH. Son inventaire peut être relié à une grande variété de ressources du cloud et de créer des processus complexes.

Il va créer des logs de suivi des jobs qu'il lance. Il peut être intégré avec un serveur LDAP ou SAML ainsi que d'autres sources d'authentifications. Les outils de lignes de commandes sont facilement utilisables avec Jenkins.

10.5 Fonctionnalités

Ansible Tower dispose d'un tableau de bord graphique pour l'utilisateur, la possibilité de planifier des jobs ainsi que de plusieurs API REST.

A.T¹ permet également une gestion de l'inventaire de manière graphique mais également une intégration d'un système de logging² externe.

Cette API dispose d'un système de contrôle d'accès basé sur les rôles.

L'équipe technique de Red-Hat propose un support au utilisateur ainsi qu'un portail d'accès pour les clients

Comme certain programme de gestion et exécution de jobs, Tower permet de suivre les mises à jour du statut des jobs en temps réel mais aussi Tower permet de régler l'ordre d'exécution des playbooks.

10.6 Installation

Contrairement aux 3 solutions présentées précédemment, Ansible Tower a, quant à lui, de plus grandes exigences au niveau du système :

Systèmes supportés :

- Red Hat Enterprise Linux 8.0 ou supérieur 64-bit (x86) (Ansible Tower 3.5 ou supérieur seulement peut être installé)
- Red Hat Enterprise Linux 7.4 ou supérieur 64-bit (x86)
- CentOS 7.4 ou supérieur 64-bit (x86)

Exigences systèmes :

- Dernière version de Google Chrome ou Mozilla Firefox
- 2 CPUs minimum
- 4 GB de RAM minimum pour l'installation de Tower (pour base de données externe ou une démo d'Ansible Tower via Vagrant)
- 20 GB d'espace disque dédié pour les services des noeuds de Tower
- 20 GB d'espace disque dédié pour les noeuds contenant une base de données (150 GB+ recommandé)
- 64-bit support requis (kernel et runtime)
- PostgreSQL version 10 requis pour le fonctionnement d'Ansible Tower 3.6 ou supérieur. Les sauvegarde est restauration fonctionnent uniquement sur les version de PostgreSQL supporter par la version installée d'Ansible Tower.
- Ansible version 2.2 (minimum) requis pour utiliser les versions Ansible Tower 3.2 ou supérieur

1. Ansible Tower

2. Système de gestion de logs externe

Une fois ces exigences remplies, il suffit de télécharger la dernière version d'Ansible Tower et de l'extraire.

Pour installer ou mettre à jour Tower, il suffit de modifier le fichier d'inventaire se situant dans le dossier `ansible-tower-setup-<tower_version>` en remplaçant le numéro de version de la ligne `<tower_version>` par le numéro de version tels que 2.4.5 ou 3.0.0

10.7 Conclusions

Ansible Tower va permettre une meilleure gestion de ce que propose Ansible ainsi qu'une prise en main plus simple et rapide.

11 JobScheduler



FIGURE 8 – Logo de JobScheduler

11.1 Présentation

Développé par "Software- und Organisations-Service GmbH", JobScheduler est un ordonnancier en Open-Source de travaux d'informations et d'automatisation de processus.

Il est distribué sous la licence GPLv2 mais est également disponible sous une licence commerciale pour un support et des services de niveau entreprise.

Ce logiciel est fourni avec plusieurs types d'interfaces : une interface Web, un client graphique ainsi qu'une interface CLI.

JobScheduler se décompose en 2 sous programmes : **JobScheduler Master and JOC¹ Cockpit** et **JobScheduler Universal Agent**.

1. <https://docs.ansible.com/ansible-tower/latest/html/quickinstall/index.html>

Ce programme est présent sur certaines plateformes telles que Windows Server 2008 R2, Windows Server 2012 R2, 2016, Windows 7 et 10, mais également sur Linux RHEL 6, 7 et 8 ainsi que CentOS 6, 7 et 8.

11.2 Historique et versions

JobScheduler sort pour la première fois en 2005, la dernière version stable est la 1.13.3 datée du 4 février 2020.

Plusieurs versions majeures ont vues le jour avant la version 1.13.3 : il s'agit des versions 1.9 le 30-04-2015, la 1.10 le 15-09-2015, 1.11 le 20-03-2017 ainsi que la version 1.12 le 21-12-2017.

La première version de la 1.13 est sortie le 29-08-2019

11.3 Contexte d'utilisation

Ce logiciel est utilisé pour lancer des fichiers exécutables ainsi que des script shells. Il permet également d'exécuter, de manière automatique, des procédures sur des bases de données.

11.4 Fonctionnement

Le type de fonctionne que propose JobScheduler est le mode SaaS, c'est-à-dire : Software as a Service (Logiciels comme service Le fonctionnement de JobScheduler par rapport à ses pairs est différents. En effet, Jobscheduler est un ensemble de programmes qui ont tous un rôles différents.

Le programme principal, qui dispose d'interface Web, un client graphique ainsi que L'interface CLI est **JOC Cockpit**.

Au travers de cette interface, l'utilisateur va envoyer des ordres d'exécutions de jobs aux "Masters", JobScheduler Master est par extension le 2ème sous-programme de JobSchduler.

Les Masters gèrent l'exécution des jobs et orchestrent également les Agents. JobScheduler Agent est le 3 ème sous-programme, les Agents sont déployés sur les serveurs sur lesquels les programmes et/ou scripts sont exécutés et qui doivent être ordonnancés.

Les informations obtenues par les Masters, à la suite de l'exécution des jobs, sont envoyés à la base de données de JobScheduler afin de pouvoir avoir un historique des jobs.

1. JOC : JobScheduler Opération Center

JOC Cockpit ainsi que JobScheduler sont disponibles sur seulement 2 plateformes : Windows et Linux

En revanche, ces masters, bien qu'ils soient disponibles sur 2 plateformes, peuvent commander les agents, qui eux sont disponibles pour un plus grand nombre de plateformes.

Les Agents fonctionnent sur les plateformes HP-UX, MacOS, Docker, Raspberry Pi, Solaris, AIX ainsi que toutes autres plateformes.

Les Agents sont disponibles pour n'importe quelle plateforme, du moment que cette dernière supporte une machine virtuelle JAVA.

11.5 Fonctionnalités

JobScheduler propose une interface graphique, un client graphique et une interface CLI.

Son mode de fonctionnement étant différents de celui de ses pairs, certaines fonctionnalités changent en fonction du service utilisé.

JobScheduler offre 2 services pour le cloud : Cloud Hosting Service et Cloud Managed Service.

Certaines fonctionnalités entre ces 2 services telles que la planification de jobs ou encore l'accès au JOC Cockpit pour le monitoring des jobs, restent les mêmes.

Cloud Hosting Service :

- Mise en place d'une infrastructure de planification
- Mise en place d'un système de management des mises à jours et patches
- Accès à des serveurs dédiés via SSH ou SFTP
- Accès à JOC Cockpit pour le monitoring et le contrôle des jobs
- Intégration du cloud au réseau de l'entreprise
- Déploiement des Agents de JobScheduler sur un environnement informatique
- Gestion des jobs
- Upload des jobs sur des serveurs dédiés via SFTP
- Monitoring de l'exécution des jobs

Cloud Managed Service :

- Mise en place d'une infrastructure de planification
- Mise en place d'un système de management des mises à jours et patches
- Accès à JOC Cockpit pour le monitoring et le contrôle des jobs
- Intégration du cloud au réseau de l'entreprise
- Management des jobs ainsi que des dépendances de jobs par l'équipe de JobScheduler
- Monitoring de l'exécution des jobs 24/7

- Déploiement des Agents de JobScheduler sur un environnement informatique
- Spécification des jobs ainsi que de leur dépendances

11.6 Installation

Contrairement aux autres logiciel d'automatisations de processus, JobScheduler n'impose pas une certaine configuration.

Les installateurs sont à télécharger depuis la page des téléchargement de JobScheduler, que ce soit pour Windows ou pour Linux

Il est possible d'installer JobScheduler de manière graphique par l'intermédiaire d'un installateur graphique, ou en ligne de commande.

L'installation graphique n'est valable que pour Windows. Cependant il est possible d'installer JobScheduler sur Windows en ligne de commande via l'invité de commande.

L'installation de JobScheduler sur Linux est exclusivement en ligne de commande et se fait au travers du terminal de commandes.

Quelque soit la plateforme sur laquelle JobScheduler doit être installé, il est demandé à l'utilisateur de configurer plusieurs éléments lors de la mise en place.

Éléments à configurer :

- Langues disponibles
- Documentation
- Licence
- Contrat
- Répertoire d'installation
- Répertoire d'exploitation
- Composant
- Caractéristiques
- Cluster
- Messagerie
- Envoi de mail
- Gestionnaire de bases de données
- Base de données
- Crontable (optionnel)
- Installation
- Journal
- Statut

L'installation silencieuse : L'installation silencieuse est présente sur Win-

dows, sous la forme de fichier ayant l'extension .msi, ainsi que sur Linux. Pour procéder L'installation silencieuse sur Linux, consiste à remplir un formulaire pour le programme en question. Ce formulaire, une fois rempli, permettra de générer un fichier d'installation silencieuse. Pour procéder à l'installation avec ce fichier, il suffit, simplement, de l'exécuter. L'installation silencieuse est une étape facultative mais qui présente de nombreux avantages. Pour JobScheduler, il est possible, sur Linux, de l'installer par l'intermédiaire de l'installation silencieuse. Si le fichier d'installation silencieuse n'est pas généré, il est possible de configurer JobScheduler directement dans son fichier d'installation. Le fichier d'installation silencieuse à le même rôle que l'installateur graphique, mais de manière plus compacte. L'installation via le cmd ou le terminal se fera de manière un peu plus condensé que la version graphique. Néanmoins les éléments à configurer restent les mêmes

11.7 Conclusions

12 Rundeck



FIGURE 9 – Logo de Rundeck

12.1 Présentation

Dans le cadre de notre projet, nous avons choisi la solution Rundeck. Rundeck est un logiciel Open-Source d'automatisation de gestion de parc informatique. Rundeck est défini comme un orchestrateur de tâches.

Sorti en 2011, il dispose d'une variante entreprise appelé "Rundeck Entreprise" permettant la gestion d'un parc informatique de plus grande envergure.

Rundeck est également un logiciel cross-platform disponible pour les systèmes UNIX, Windows et Mac OS.

Rundeck est hébergé sur la plate forme de développement GitHub, ce qui permet à chaque utilisateur de Rundeck de contribuer à son développement.

Rundeck est en majeure partie, amélioré grâce à ses utilisateurs. Ces derniers peuvent participer au développement de Rundeck par la notification de bugs/-fonctionnalités manquantes ou en rejoignant une équipe de développement de Rundeck.

Rundeck est développé en JAVA et propose une interface WEB (requiert les paquets JAVA), une interface CLI (Command-Line Interface) ainsi qu'une API REST.

12.2 Historique et versions

Rundeck est apparu en 2011 suite à la demande de plus en plus forte du besoin de pouvoir administrer tous les serveurs d'un parc informatique depuis un seul serveur d'administration central.

La dernière version en date est la version 3.2.0 datant du 10 février 2020.

Rundeck est diffusé sous la licence Apache Software 2.0

12.3 Contexte d'utilisation

Rundeck est, parmi un large panel de logiciel d'automatisation, un des plus utilisé lorsque l'utilisateur souhaite automatiser la gestion d'un parc informatique depuis une seule machine physique.

12.4 Fonctionnement

Rundeck permet l'exécution de tâches et/ou jobs sur des serveurs distant via une connexion SSH. Rundeck fonctionne sur des réseaux privées où les machines possèdent des adresses IP statiques.

Les adresses IP dynamique des serveurs distants peuvent causer des conflits d'IP et de clés SSH sur le serveur de Rundeck.

Rundeck également basé sur le principe maître/esclave (master/slave) : La machine où se trouve l'application Rundeck est le maître tandis que le serveur distant est l'esclave.

De plus, Rundeck montrera un meilleur fonctionnement si celui-ci est installé sur une distribution orientée serveur telles que CentOS.

Par défaut, Rundeck écoute sur le port 4440 (http) mais peut également écouté sur le port 4443 (https)

Cependant, Rundeck peut être installé sur d'autres distributions telles qu'Ubuntu, Debian ou même Windows à la seule conditions d'avoir les paquets Java correspondants sous peine de rencontrer des problèmes d'exécution ou d'affichage. Rundeck, par défaut, fonctionne avec sa propre base de données, un interface WEB ainsi qu'un compte administrateur dont les identifiants sont "admin" pour le nom d'utilisateur et "admin" pour le mot de passe

12.5 Fonctionnalités

Rundeck possède une grande gamme de fonctionnalités qui permettent à l'utilisateur de Rundeck, une gestion optimale d'un parc informatique. Rundeck mets à disposition un choix de plus de 1000 plugins de nature diverses et variées.

Rundeck propose également diverses fonctionnalités notamment l'exécution de commandes/tâches à distances

Rundeck également très modulable. En effet, Rundeck utilise une base de données embarquée pour l'enregistrement de données telles que les jobs, les clés SSH, les utilisateurs qui ont accès à Rundeck.

12.6 Installation

L'installation de Rundeck est très simple, il est recommandé de l'installer sur CentOS afin de pouvoir utiliser Rundeck de manière optimale. Il est également possible d'installer ce programme sur Windows.

Étant au centre de notre projet, la procédure d'installation de Rundeck se trouve plus bas dans le document, au point 15.2

12.7 Conclusions

13 Tableaux synthétiques

Tableaux comparatifs des solutions envisagées. Les critères sont organisés par catégories afin que cela soit plus lisible. Des informations complémentaires pourrait apparaître dans ces tableaux

Années de sortie :

Solutions	Années de sortie
Cron	1975
Jenkins	2011
Buildbot	2003
Ansible Tower	2015
JobScheduler	2005
Rundeck	2011

Installation : Les pré-requis caractérisent une configuration matérielle et logicielle minimum pour installer le programme

Solutions	Pré-requis	Plateforme
Cron	Non	Linux, MacOS
Jenkins	Oui	Tous OS
Buildbot	Oui	POSIX, Windows
Ansible Tower	Oui	Windows, Linux, Linux-like
JobScheduler	Oui	Windows, Linux, Linux-like
Rundeck	Oui	Windows, Linux, Linux-like

Cadre d'utilisation : Un résumé très bref sur la nature et le rôle de chaque solution

Solutions	Type	Rôle / Utilisation
Cron	Daemon	Planificateur de tâches
Jenkins	Programme	Intégration continue
Buildbot	Framework	Intégration continue (web)
Ansible Tower	API	Planificateur de tâches
JobScheduler	Programme	Planificateur de tâches
Rundeck	Programme	Planificateur de tâches

Administration : Interface : type d'interface

BDD : Base de données pré-intégré et/ou possibilité d'en intégrer

Extensible : Extension du programme grâce à des plugins

Solutions	Interface	BDD	Extensible
Cron	CLI	Non	Non
Jenkins	Graphique	Non	Oui
Buildbot	Graphique	Oui	Oui
Ansible Tower	Graphique	Oui	Oui
JobScheduler	Graphique	Oui	Oui
Rundeck	Graphique	Oui	Oui

Gestion :

Solutions	Suivi des jobs	Logs	Stockage des logs
Cron	Via logs	Oui	Machine locale
Jenkins	Après exécution	Oui	Machine Locale
Buildbot	Après exécution	Oui	BDD
Ansible Tower	En temps réel	Oui	Machine Locale
JobScheduler	En temps réel	Oui	Machine Locale
Rundeck	En temps réel	Oui	BDD, Machine Locale

LDAP : Possibilité de mettre en place un système d'authentification via un annuaire LDAP

Plugin : requiert l'installation d'un plugin

Pré-intégré : Pas de plugins, spécification des adresses des annuaires dans les fichiers de configuration et/ou depuis l'interface graphique

Solutions	Connecteur LDAP	Intégration
Cron	Non	Aucune
Jenkins	Oui	Plugin
Buildbot	Oui	Plugin
Ansible Tower	Oui	Pré-intégré
JobScheduler	Oui	Pré-intégré
Rundeck	Oui	Pré-intégré

Autres : Officiel : Support fourni par l'équipe créatrice du programme

Solutions	Support	Type
Cron	Oui	Communautaire
Jenkins	Oui	Officiel
Buildbot	Oui	Officiel
Ansible Tower	Oui	Officiel
JobScheduler	Oui	Officiel
Rundeck	Oui	Officiel

14 Mise en place de Rundeck

14.1 Environnement de travail

Dans le cadre de notre projet, nous avons choisit un environnement de travail sous Linux ainsi que l'appel de plusieurs outils de virtualisation tels que Vagrant, Oracle VirtualBox.

L'intérêt d'utiliser Oracle VirtualBox pour mettre en place cette infrastructure, est qu'il présentera plus de facilité à s'installer sur des machines hôtes sur Linux. Notre infrastructure de travail et de démonstration se fait à l'aide 4 machines virtuelles, mais peut cependant être étendu à un nombre plus grand de machines virtuelles. L'unique contrainte étant les performances des machines hôtes.

Infrastructure :

- Machine virtuelle **Rundeck** sous CentOS
- Machine virtuelle **web** sous Debian
- Machine virtuelle **mail** sous Debian
- Machine virtuelle **nfs** sous Debian

Informations relatives à chaque machine virtuelle :

Rundeck :

Le système d'exploitation choisit est CentOS de par son optimisation pour les serveurs. Cette VM, appelée "Rundeck", du même nom que l'outil d'orchestration installé sur cette dernière. Cette VM est la "VM Maître" étant donné qu'elle supervise et gère toutes les autres VMs qui se trouvent sous son autorité. Cette machine virtuelle est générée grâce à une box Vagrant

web : Cette machine virtuelle joue le rôle d'un serveur web. Cette machine est installée avec un système Debian GNU/Linux, ainsi que tous les composants et paquets requis pour assurer le bon fonctionnement d'un serveur web simple. Cette machine virtuelle est générée grâce à une boxe Vagrant

mail : Cette machine virtuelle joue le rôle d'un serveur mail. Cette machine est installée avec un système Debian GNU/Linux, ainsi que tous les composants et paquets requis pour assurer le bon fonctionnement d'un serveur mail. Cette machine virtuelle est générée grâce à une boxe Vagrant

nfs : Cette machine virtuelle constitue un point de montage afin de pouvoir partager des fichiers entre chaque machine. Cette machine est installée avec un système Debian GNU/Linux, ainsi que tous les composants et paquets requis pour assurer le bon fonctionnement d'un point de montage de partage de fichiers. Cette machine virtuelle est générée grâce à une boxe Vagrant

14.2 Installation

14.2.1 Exigences système

- Linux : Distributions récentes conseillées pour un fonctionnement optimal
- Windows : XP, Server et supérieures (Distributions récentes conseillées)
- Mac : OS X 10.4 ou supérieure

Accès root (ou administrateur) non requis ; création d'un compte utilisateur dédié créer conseillé par Rundeck

Informations complémentaires fournies par Rundeck

- OS supportés : Red Hat Enterprise Linux - CentOS - Ubuntu - Windows Server
- Dernière versions supportés de Mozilla Firefox ou Google Chrome (les autres navigateurs peuvent fonctionner mais présentant des problèmes d'affichage)
- 2 CPU
- 4 GB de mémoire RAM minimum
- 20 GB d'espace disques minimum
- Base de données supportées : MySQL - MariaDB - PostgreSQL - Oracle
- Logs : Système de fichiers

La version 1.8 de JAVA est également requise

14.2.2 Windows

14.2.3 Linux

L'installation de Rundeck sous Linux est très simpliste. Afin d'obtenir un fonctionnement optimal, Rundeck a été mis en place sous CentOS. Son installation s'est composé des étapes suivantes :

- `sudo yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel -y` # Installation de la version de Java requise
- `sudo rpm -Uvh http://repo.rundeck.org/latest.rpm` # Récupération de la dernière version de Rundeck
- `sudo yum -y install rundeck` # Installation de Rundeck

14.3 Configuration

14.3.1 Arborescence

Les fichiers de configurations de Rundeck sont répartis de manière à pouvoir se repérer dans sa configuration, c'est-à-dire, les fichiers de configuration de Rundeck sont stockés dans le répertoire `/etc/rundeck` tandis que les fichiers de configurations propres aux tâches/jobs, est stockés dans le répertoire `/var/lib/rundeck`.

Ci-dessous, la disposition DEB/RPM :

```
/etc/rundeck/
├─ admin.aclpolicy
├─ apitoken.aclpolicy
├─ artifact-repositories.yaml
├─ cli-log4j.properties
├─ framework.properties
├─ jaas-loginmodule.conf
├─ profile
├─ project.properties
├─ realm.properties
├─ rundeck-config.properties
├─ rundeckpro-licence.key
├─ ssl
│   └─ ssl.properties
│       ├── keystore
│       └─ truststore
├─ system-job_readeraclpolicy_template
├─ system-job_runneraclpolicy_template
├─ system-job_vieweraclpolicy_template
├─ system-job_writeraclpolicy_template
└─ system-project_adminaclpolicy_template
```

/var/lib/rundeck

- ├ bootstrap
- ├ cli
- ├ data
- ├ libext
- ├ logs
- ├ projects
- ├ repository
- ├ var
- └ work

Ci-dessous, la disposition de l'interface de Rundeck :

\$RDECK_BASE/etc/

- ├ admin.acmpolicy
- ├ apitoken.aclpolicy
- ├ cli-log4j.properties
- ├ framework.properties
- ├ preferences.properties
- ├ profile
- ├ profile.bat
- └ project.properties

```
$RDECK_BASE/server/config/
├─ artifact-repositories.yaml
├─ jaas-loginmodule.conf
├─ log4j.properties
├─ realm.properties
├─ rundeck-config.properties
└─ ssl.properties
```

Dans le cadre de notre projet, nous avons appliqué une configuration simple, impliquant les fichiers de configurations suivants :

- /etc/rundeck/framework.properties
- /etc/rundeck/realm.properties
- /etc/rundeck/rundeck-conf.properties

Informations relatives à chaque onglet :

framework.properties :

Ce fichiers contient diverses informations : Les informations de connexion au serveur Rundeck, l'emplacement des fichiers d'installations ainsi que l'exécuteur SSH et Copier de fichiers par défaut

Définition des propriétés de connexion à l'interface Web : Définition du nom du serveur, nom de l'hôte, le port ainsi que l'URL

Définition des emplacements des fichiers d'installations : Stockage des logs, projets, fichiers temporaires etc...

Définition des emplacements des clés SSH ainsi que du temps de connections avant le "timeout"

rundeck-conf.properties :

Ce fichier permet de définir les méthodes de stockages, chiffrement des fichiers ainsi que les niveaux de logs.

realm.properties :

Fichier contenant identifiants, mots de passe et droits de chaque utilisateurs.

Seulement 4 droits sont disponibles : job-runner (peut lancer des jobs), job-writer

(peut créer des jobs)), job-reader(peut lire les jobs), job-viewer(peut voir les jobs)

14.3.2 Interface de connexion

Rundeck est fourni, par défaut, avec une interface web dont l'URL est : <http://localhost:4440>

Cette URL peut, toutefois être modifié, en éditant les fichiers *framework.properties* et *rundeck-config.properties*.

Le paramètre *framework.rundeck.url* contient l'URL de base de Rundeck et peut être remplacé par une adresse au choix.

Lors de la première utilisation de Rundeck suivant son installation, les identifiants de connexion par défaut son *admin* pour l'"username" et *admin* pour le mot de passe.

Ces identifiants peuvent être changés en éditant le fichier *realm.properties*

Des comptes utilisateurs peuvent être également ajouté depuis ce même fichier

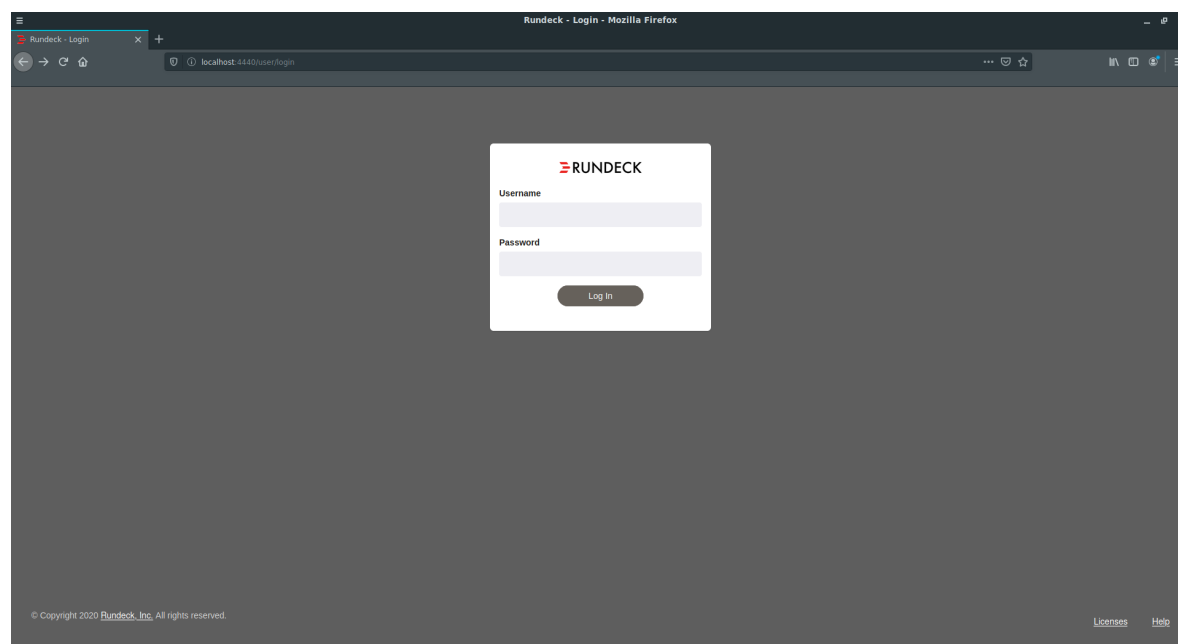


FIGURE 10 – Interface de connexion

Username : admin

Password : admin

14.3.3 Création d'un projet

La particularité de Rundeck est de pouvoir fonctionner sous forme de projet. Cette fonctionnalité est native à Rundeck et est aussi indispensable. En effet, les projets permettent de gérer plusieurs parcs informatiques d'une même entreprise depuis une seule et unique machine.

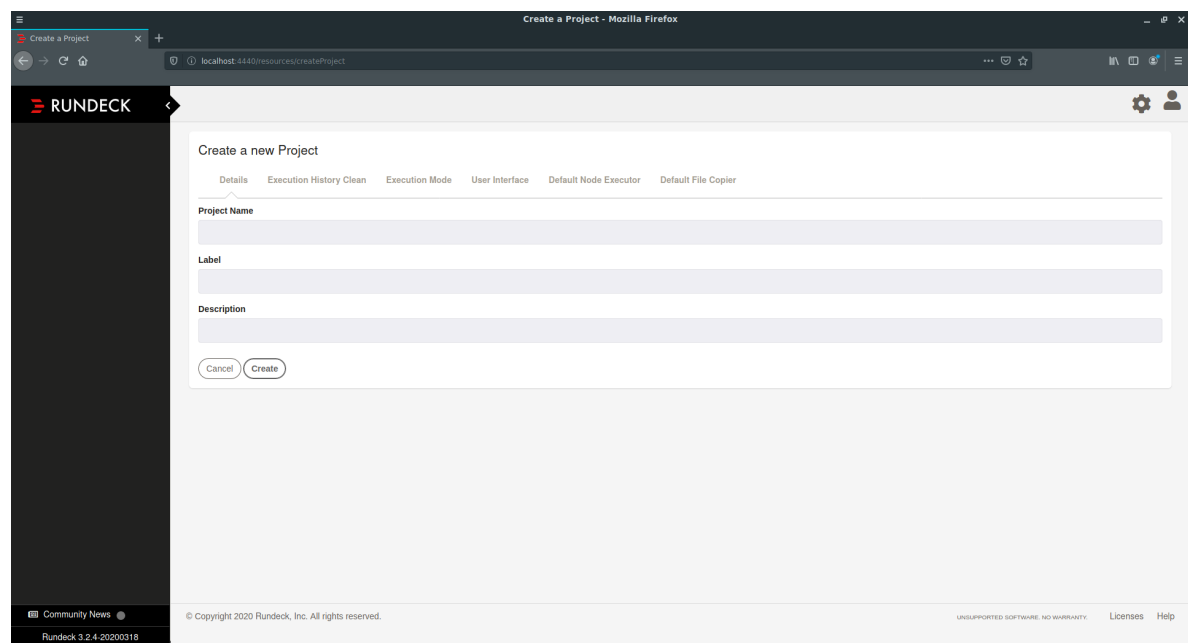


FIGURE 11 – Interface de connexion

Lors de la création du projet, plusieurs onglets sont mis à dispositions afin de permettre une configuration complète d'un projet

- Les détails du jobs "**Details**"
- Nettoyage automatique de l'historique d'exécution "**Execution History Clean**"
- Paramètres d'exécution et planification "**Execution Mode**"
- Personnalisation de l'interface utilisateur "**User Interface**"
- Méthode par défaut d'exécution sur un node "**Default Node Executor**"
- Méthode par défaut de copie de fichier sur un node "**Default File Copier**"

Informations relatives à chaque onglet :

Onglet "Details" :

L'utilisateur peut définir, depuis cet onglet, le nom ainsi qu'une courte description du projet

Onglet "Execution History Clean" :

trad : Nettoyage de l'historique d'exécution

Cet onglet permet de régler le/les fréquences de nettoyage automatiques des exécutions. Ce nettoyage peut être activé ou désactivé à la demande de l'utilisateur.

Des valeurs par défaut sont imposé par Rundeck telles que le nombre de jours que l'on souhaite garder l'historique (ex : Rundeck supprimera l'historique au bout de 60 jours).

Rundeck propose un nombre d'exécution minimum fixé à 50 exécution, une taille maximum de l'historique limité à 500Mo ainsi que la possibilité de paramétrer via **une expression de type cron**, le nettoyage automatique de l'historique.

Ces valeurs sont les valeurs par défaut imposé par Rundeck mais peuvent être modifié par l'utilisateur de Rundeck.

Onglet "Execution Mode" :

trad : Mode d'exécution

Cet onglet permet d'activer/désactiver l'exécution des jobs, des commandes ad-hoc ainsi que la planification des jobs et commandes ad-hoc.

Onglet "User Interface" :

trad : Interface utilisateur

Depuis cet onglet, un README ainsi qu'un MOTD ("Message Of The Day") peuvent être affichés sur certaines pages lorsque qu'un utilisateur de Rundeck se connecte à l'interface

Onglet "Default Node Executor" :

trad : Exécuteur de node par défaut

Par défaut, Rundeck règle l'exécuteur sur SSH, c'est-à-dire que les jobs/commandes seront exécutés via SSH.

Cette méthode peut toutefois être changée si Rundeck possède des machines d'une autre distributions que Linux sous son contrôle.

Rundeck propose plusieurs autres exécuteurs. Néanmoins, l'administrateur souhaite utiliser un autre exécuteur qui n'est pas présent dans la liste, ce dernier peut être téléchargé via la liste des plugins (si celui se trouve à l'intérieur)

Si des clés SSH et passphrases sont stockés directement sur Rundeck, il est

possible de spécifier les chemins de fichiers afin de permettre l'accès via ces protocoles

Onglet "Default File Copier" :

trad : Copier de fichier par défaut

Même principe que l'onglet précédent, celui-ci permet de modifier l'utilitaire de transfert de fichiers d'une machine à une autre via Rundeck.

Plusieurs méthodes sont mises à disposition par Rundeck et d'autres méthodes peuvent être téléchargées depuis la liste des plugins si la méthode demandée par l'administrateur ne se trouve pas dans la liste fournie par Rundeck.

Si des clés SSH et passphrases sont stockées directement sur Rundeck, il est possible de spécifier les chemins de fichiers afin de permettre l'accès via ces protocoles.

14.3.4 Configuration du stockage

Rundeck propose plusieurs méthodes de stockage :

Méthodes de stockage :

- Base de données embarquée
- Base de données "spécifique"
- Système de fichiers

L'emplacement des données peut être sur le système de fichiers de la machine, dans une base de données (spécifique ou celle fournie par Rundeck) ou encore sur un système de stockage externe. Un plugin pour le stockage externe sera requis.*

Rundeck fournit également 2 implémentations possibles :

- filesystem : stockage local des données
- db - stockage dans une base de données

Dans notre cas, nous avons fait appel à la base de données embarquée de Rundeck. L'utilisation d'une base de données est également fortement recommandée par Rundeck.

14.3.5 Définition des nodes

Un node permet d'ajouter un serveur, une machine que l'on souhaite automatiser avec Rundeck.

Les nodes sont définis dans un fichier au format XML mais peuvent également être définis dans un fichier au format YAML et JSON. Cette définition doit suivre une syntaxe imposé par Rundeck. On peut définir autant de nodes que l'on souhaite dans ce même fichier.

Ces exemples sont à titres indicatifs et sont fournis dans la documentation de Rundeck

Syntaxe XML :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project>
4   <node name="localhost"
5       description="Rundeck server node"
6       tags=""
7       hostname="localhost"
8       osArch="amd64"
9       osFamily="unix"
10      osName="Linux"
11      osVersion="2.6.21.7-2.fc8xen"
12      username="rundeck"/>
13 </project>
```

Syntaxe JSON :

```
1 {
2   "madmartigan.local": {
3     "tags": "local , server",
4     "osFamily": "unix",
5     "username": "greg",
6     "osVersion": "10.10.3",
7     "osArch": "x86_64",
8     "description": "Rundeck server node",
9     "hostname": "madmartigan.local",
10    "nodename": "madmartigan.local",
11    "osName": "Mac OS X"
12  },
13  "test": {
14    "tags": "alphabet , soup",
```

```

15     "osFamily": "unix",
16     "ssh-key-storage-path": "keys/testkey1.pem",
17     "username": "vagrant",
18     "osVersion": "10.10.3",
19     "osArch": "x86_64",
20     "description": "Rundeck server node",
21     "hostname": "192.168.33.12",
22     "nodename": "test",
23     "osName": "Mac OS X"
24   }
25 }

```

Syntaxe YAML :

```

1 Venkman.local:
2   description: Rundeck server node
3   hostname: Venkman.local
4   nodename: Venkman.local
5   osArch: x86_64
6   osFamily: unix
7   osName: Mac OS X
8   osVersion: 10.6.6
9   tags: ''
10  username: greg

```

Cette configuration est identique pour chaque ordinateur/serveur que l'on souhaite automatiser et peut également se faire dans plusieurs fichiers dès lors que l'on utilise une base de données. Dans notre cas, toutes les machines sont enregistrées dans un seul fichier.

Informations sur la syntaxe

- name : nom de l'ordinateur/serveur (obligatoire)
- description : brève description du node (facultatif)
- tags : nom permettant l'identification du node (facultatif)
- hostname : adresse IP de la machine (obligatoire)
- osArch : Architecture du système (facultatif)
- osFamily : Famille du système (facultatif)
- osName : Nom du système (facultatif)
- osVersion : Version de l'OS (facultatif)
- username : compte avec lequel rundeck se connecte à la machine distante (obligatoire)

La définition d'un node dans le format YAML ou JSON diffère légèrement du format XML.

14.3.6 Clés SSH

Rundeck requiert les clés SSH de chaque machine qu'il a sous son contrôle. Ces clés peuvent être créées et enregistrées sur les machines en prenant soins d'être connecté avec un utilisateur dédié. Ces clés peuvent également être enregistré sur la base de données de Rundeck

14.3.7 Définition des jobs

La définition d'un "job" est l'étape la plus importante. En effet, ce système est l'atout principal de Rundeck.

Un job est décomposé en plusieurs étapes (*step en anglais*). Une étape correspond à une commande ou un script

Les jobs se définissent de la même manière, et ceux peut importe la manière dont Rundeck est configuré.

La définition d'un job se fait sur l'interface WEB et propose plusieurs onglets de configurations :

- Les détails du jobs "**Details**"
- L'ordre d'exécution "**Workflow**"
- Les noeuds "**Nodes**"
- La planification de l'exécution "**Schedule**"
- Les notifications "**Notifications**"
- Les options diverses "**Others**"

Informations relatives à chaque onglet :

Onglet "**Details**" :

L'utilisateur peut définir, depuis cet onglet, le nom ainsi qu'une courte description du job

Onglet "**Workflow**" :

trad : flux de travail

Cet onglet permet la définition de la méthode d'exécution. Cet onglet permet également de définir le/les tâches à exécuter.

Rundeck propose lors du paramétrage, de continuer l'exécution du jobs même si une "step" échoue.

Rappel : une step n'est qu'une petite partie d'un job et non le job complet.

Méthodes d'exécution :

- Node First : Exécute toute les étapes d'un job sur le noeud 1, puis sur le noeud 2 etc...
- Parallel : Exécution de tous en parallèle et en même temps sur tous les noeuds
- Sequential : Le step 1 sera joué sur le noeud 1 puis noeuds 2 puis noeud X ensuite le step 2 ainsi de suite

Onglet "Nodes" :

trad : Noeuds

Cet onglet permet de définir les nodes sur lequel/lesquels le job va être exécuté. Rundeck propose à l'utilisateur si le job soit être soit lancé sur la machine qui contient Rundeck, soit sur les machines que Rundeck à sous son contrôle.

Beaucoup de paramètres sont disponibles tels que les filtres où l'administrateur peut décider sur quels noeuds il souhaite exécuter ce job. Cette notion implique de bien rédiger le fichier contenant les nodes.

Depuis cette onglet, l'administrateur peut également dire à Rundeck que faire pour une situation donnée (ex : Si le node échoue, on peut soit arrêter complètement le job, soit continuer sur les nodes restants.

Onglet "Schedule" :

trad : Planifier

Comme son nom l'indique, cet onglet permet de programmer l'heure d'exécution du job, mais il permet également de définir la fréquence d'exécution.

La possibilité de planification peut être désactivé depuis cet onglet, de même pour l'exécution.

Rundeck propose 2 manières de planifier un job : De manière simple en sélectionnant l'heure et la fréquence, ou alors via une expression cron

Onglet "Notifications" :

Depuis cet onglet, peut être définit la manière dont sera notifié l'administrateur de Rundeck.

Une notification peut être envoyée en cas de succès de job, d'échec, une notification au démarrage, sur des nouvelles tentatives d'exécution de jobs si celui-ci échoue une première fois.

Deux méthodes de notifications sont disponibles : Webhook ou Email

Email : Mail contenant diverses informations telles que le job exécuter, l'heure

d'exécution, son statut (réussite/échec/démarrage/nouvelle tentative), la/les machine(s) où le job à était exécuté. Toutes ces informations sont envoyés à l'administrateur de Rundeck

Webhook : Un webhook contiendra les mêmes informations qu'un mail et se feront au travers d'un rappel HTTP. Cela peut être traduis par une notification en temps réel

Onglet "Others" :

trad : Autres/Divers

Cet onglet est essentiellement consacrés au logs et aux tentatives sur les jobs, on peut définir la nature du fichier de logs (fichier classique ou fichier pour debug) ainsi que la taille d'un fichier de logs (ex : 100 lignes par fichiers), autoriser les multiples exécution du job, le nombre de tentatives, le temps de réponses d'un job, le délais entre tentatives.

14.3.8 Gestions des utilisateurs

La gestion des utilisateurs peut se faire de 2 manières : en ajoutant les utilisateurs dans le fichier `realm.properties` et dans l'interface de Rundeck

Realm.properties : L'utilisateur "admin" est l'utilisateur créer par défaut et par Rundeck. Il possède tous les droits et peut effectuer toutes les actions possibles et disponibles sur Rundeck.

La création de nouveaux utilisateurs permet à l'administrateur de Rundeck, par extension, de déléguer des jobs.

La syntaxe pour nouvel utilisateur ainsi que pour ses droits est spécifié dans ce même fichiers

Interface de Rundeck : Depuis l'interface de Rundeck, l'administrateur peut générer des "tokens" pour les utilisateurs. Ces "tokens" permettent de renforcer la sécurité.

Un utilisateur peut avoir plusieurs "tokens" mais un token ne peut pas être créer pour plusieurs utilisateurs

Ces tokens on une durée de vie définie par l'administrateur ou l'utilisateur lui même.

De manière imagée, un token correspond à un passeport.

15 Rundeck et Ansible

Rundeck à la particularité de pouvoir fonctionner avec Ansible. Par ailleurs, le couplage de ces 2 programmes est très populaire dans la communauté de

l'administration et gestion de système.

Rundeck peut être intégré à Ansible et inversement par l'intermédiaire d'un plugin.

Plusieurs versions de ce plugin sont développées par les communautés de Rundeck et d'Ansible.

Du point de vue d'Ansible, l'intégration de Rundeck à Ansible permet de le rendre meilleur, c'est-à-dire que grâce à ce couplage, l'utilisateur d'Ansible aura la possibilité d'exécuter toutes les commandes de Rundeck depuis Ansible.

A l'inverse, du point de vue de Rundeck, ce couplage permettra à l'utilisateur de Rundeck de pouvoir exécuter des playbooks, qui sont propres à Ansible.

Cet alliance est très populaire chez les administrateurs et présente de nombreux avantages.

16 Authentification via un annuaire LDAP

Il est possible, comme pour beaucoup d'autre logiciel d'automatisation de processus, de mettre en place une authentification via un annuaire LDAP¹.

Un annuaire LDAP contient les noms, prénoms, mots de passe ainsi que toute autre information complémentaire pour un utilisateur.

Cet annuaire est uniquement destiné à gérer les données sur les utilisateurs.

Le principe est de connecter un annuaire à un serveur de gestion, par exemple Rundeck, Ansible ou encore Jenkins.

Pour connecter un annuaire de ce type, il est nécessaire de se référer à la documentation du programme en question.

Dans le cas de Rundeck, cet ajout se fait dans les fichiers de configuration et ne requiert donc aucun plugin.

Pour certains logiciels tels que Jenkins ou Buildbot, un plugin sera nécessaire.

Configurer un annuaire LDAP consiste à définir un fichier de configuration JAAS² (e.g. "jaas-ldap.conf"), et changer le script de démarrage pour utiliser ce fichier et donc utiliser le module de Login se trouvant à l'intérieur - Documentation de Rundeck

1. LDAP : Lightweight Directory Access Protocol

2. JAAS : Java Authentication and Authorization Service

17 Conclusions Générales

L'objectif principal de ce projet est de mettre en place une infrastructure sur laquelle une machine "maître" pourrait exécuter des tâches de maintenances, à distance et de manière automatique.

Dans un premier temps, nous nous sommes orienté sur une analyse des systèmes automatiques déjà existant afin d'avoir une base pour notre projet.

Ensuite nous nous sommes concentrés sur l'analyse de diverses solutions envisageables pour ce projet parmi lesquelles se trouve, Cron, Jenkins, Buildbot, Ansible Tower, JobScheduler ainsi que Rundeck.

Cron : Cron est un planificateur de tâches basique et pré-intégré aux distributions Linux. Il est le plus vieux systèmes de planification et d'automatisation de tâches. Il sert notamment de bases pour les logiciels d'automatisations présents sur le marché.

Jenkins : Jenkins est un outil d'intégration continue, il est utilisé dans le cadre de automatisation de la compilation, construction et l'assemblage de paquets.

Buildbot : Buildbot est un framework d'intégration continue, il est utilisé dans le cadre de automatisation de la compilation, construction et l'assemblage de paquets. Il est notamment utilisé dans le développement Web

Ansible Tower : Ansible Tower est une API d'Ansible, qui est le programme principale, permettant à l'utilisateur de gérer de manière graphique, une infrastructure informatique.

JobScheduler : JobScheduler est un ordonnanceur de tâches spécialement utilisé pour la gestion de tâches sur un cloud. Il fonctionne en mode SaaS

Rundeck : Rundeck est un orchestrateur de tâches permettant à l'utilisateur de gérer de manière graphique mais aussi via une interface CLI, une infrastructure informatique

Dans le cadre de notre projet, nous avons choisit Rundeck pour sa facilité de prise en main, sa facilité à comprendre son interface mais aussi pour les nombreux avantages qu'il présente. Rundeck peut fonctionner de manière indépendante ou alors en étant couplé à des logiciels externes tels qu'Ansible.

Nous concernant, il faut dire que cela été un défi à relever mais qui s'est avéré

intéressant. Nous avons consacré beaucoup de temps pour ce projet afin d'en saisir toutes les subtilités, son importance ainsi que ses enjeux. Nous avons passé du temps pour rechercher les informations spécifiques aux divers programmes mais également pour les assimiler. Ce travail de conception et de rédaction a été très enrichissant.

18 Bibliographie

- <https://jenkins.io/doc/>
- <https://jenkins.io/doc/book/installing/>
- <http://docs.buildbot.net/current/index.html>
- <http://docs.buildbot.net/current/manual/installation/index.html>
- <https://docs.ansible.com/ansible-tower/>
- <https://docs.ansible.com/ansible-tower/latest/html/quickinstall/index.html>
- <https://www.sos-berlin.com/en/jobscheduler-documentation>
- <https://kb.sos-berlin.com/display/PKB/Knowledge+Guide+-+Getting+Started>

19 Annexes