

# Rapport Projet Tuteuré Runderck

HEPPNER Tristan / ROYER Grégory  
TORRES Yanis / AISSI Ayoub

Janvier 2020

# Table des matières

<b>1</b>	<b>Remerciements</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
<b>3</b>	<b>Sujet de la soutenance</b>	<b>5</b>
<b>4</b>	<b>L'automatisation</b>	<b>6</b>
<b>5</b>	<b>Principes d'automatisations</b>	<b>6</b>
<b>6</b>	<b>Cron</b>	<b>7</b>
6.1	Présentation . . . . .	7
6.2	Historique et versions . . . . .	7
6.3	Contexte d'utilisation . . . . .	7
6.4	Fonctionnement . . . . .	8
6.5	Fonctionnalités . . . . .	8
6.6	Conclusions . . . . .	8
<b>7</b>	<b>Jenkins</b>	<b>8</b>
7.1	Présentation . . . . .	8
7.2	Historique et versions . . . . .	9
7.3	Contexte d'utilisation . . . . .	9
7.4	Fonctionnement . . . . .	9
7.5	Fonctionnalités . . . . .	10
7.6	Conclusions . . . . .	10
<b>8</b>	<b>Buildbot</b>	<b>10</b>
8.1	Présentation . . . . .	11
8.2	Historique et versions . . . . .	11
8.3	Contexte d'utilisation . . . . .	11
8.4	Fonctionnement . . . . .	11
8.5	Fonctionnalités . . . . .	11
8.6	Conclusions . . . . .	12
<b>9</b>	<b>Ansible Tower</b>	<b>12</b>
9.1	Présentation . . . . .	12
9.2	Historique et versions . . . . .	12
9.3	Contexte d'utilisation . . . . .	13
9.4	Fonctionnement . . . . .	13
9.5	Fonctionnalités . . . . .	13

9.6 Conclusions . . . . .	13
<b>10 JobScheduler</b>	<b>13</b>
10.1 Présentation . . . . .	14
10.2 Historique et versions . . . . .	14
10.3 Fonctionnement . . . . .	14
10.4 Fonctionnalités . . . . .	14
10.5 Conclusions . . . . .	14
<b>11 Rundeck</b>	<b>14</b>
11.1 Présentation . . . . .	14
11.2 Historique et versions . . . . .	15
11.3 Contexte d'utilisation . . . . .	15
11.4 Fonctionnement . . . . .	15
11.5 Fonctionnalités . . . . .	16
11.6 Conclusions . . . . .	16
<b>12 Mise en place de Rundeck</b>	<b>16</b>
12.1 Environnement de travail . . . . .	16
12.2 Installation . . . . .	17
12.2.1 Exigences système . . . . .	17
12.2.2 Windows . . . . .	18
12.2.3 Linux . . . . .	18
12.3 Configuration . . . . .	18
12.3.1 Arborescence . . . . .	18
12.3.2 Interface de connexion . . . . .	21
12.3.3 Création d'un projet . . . . .	22
12.3.4 Définition de la méthode de stockage . . . . .	25
12.3.5 Définition des nodes . . . . .	25
12.3.6 Clés SSH . . . . .	27
12.3.7 Définition des jobs . . . . .	27
<b>13 Conclusions</b>	<b>29</b>
<b>14 Bibliographie</b>	<b>30</b>
<b>15 Annexes</b>	<b>31</b>

# 1 Remerciements

Nous remercions notre tuteur de projet tutoré, Mr Fabien PASCALE, Expert en calcul scientifique au CNRS, 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 eu de cesse que de trouver divers moyens pour améliorer la vie de ses semblables. Nous sommes, par nature, des êtres sociables, sociaux et dotés d'intelligence. L'Homme, étant doté d'intelligence et de créativité

## 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 :

- Rundeck
- Jenkins
- Buildbot
- Ansible Tower
- JobScheduler
- Crontab

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.pacale@gmail.com

### **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 un bref historique, son contexte

d'utilisation, la présentation de la solution, son fonctionnement, ses fonctionnalités ainsi qu'un 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. Dans l'antiquité Romaine, le niveau de l'eau qui transitait sur les aqueducs était réguler par des valves.

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 fondamentale dans certains domaines notamment dans l'industrie ou encore les systèmes informatiques.

## 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 récurrents notamment pour les mises à jours de systèmes et/ou logiciels

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 Cron



FIGURE 1 – Logo de Crontab

### 6.1 Présentation

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

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

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

Cron et Crontab 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.

### 6.2 Historique et versions

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

### 6.3 Contexte d'utilisation

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

## 6.4 Fonctionnement

Cron permet la planification et l'exécution automatique de tâches programmées à l'avance. Ces tâches sont réglés dans un fichier contenant la/les commande(s) à exécuter ainsi que l'heure d'exécution. Cependant, cron ne permet pas la ré exécution d'une tâche si celle-ci échoue.

## 6.5 Fonctionnalités

Cron permet une planification de tâches pourrons s'exécuter quotidiennement, hebdomadairement, mensuellement ou annuellement. Cette fréquence se règle avec des paramètres

## 6.6 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.

# 7 Jenkins



# Jenkins

FIGURE 2 – Logo de Jenkins

## 7.1 Présentation

Jenkins est un logiciel Open-Source d'automatisation, tout comme Rundeck. Sorti en 2011, Jenkins est un dérivé de son prédécesseur 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 Rundeck de contribuer à son développement.

Jenkins 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 et requiert donc des dépendances JAVA.

## **7.2 Historique et versions**

Tout comme Rundeck, 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 automatique. Jenkins est publié sous la licence MIT

## **7.3 Contexte d'utilisation**

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

## **7.4 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,

Jenkins 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 d'erreur.

## 7.5 Fonctionnalités

Comme tous 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.

## 7.6 Conclusions

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

## 8 Buildbot



FIGURE 3 – Logo de Buildbot

## 8.1 Présentation

Tout comme Jenkins, Buildbot est un logiciel Open-Source d'automatisation. Buildbot est un logiciel cross-platform et est disponible sur les systèmes Windows et POSIX (Portable Operating System Interface).

Ce logiciel est présentée comme une alternative au projet de Mozilla : [Tindebox](#). Buildbot se démarque de ses concurrents notamment grâce à sa légèreté ainsi que la possibilité de l'installer sur des systèmes d'exploitations portables. En effet, en terme de légèreté, ce programme ne pèse que 4.6MB.

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.

## 8.2 Historique et versions

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

## 8.3 Contexte d'utilisation

Buildbot est utilisé dans le cadre de la compilation automatique de "builds". De grandes entreprises, telles que Mozilla, Chromium, WebKit et bien d'autres, utilisent Buildbot.

## 8.4 Fonctionnement

De la même manière que Jenkins, Buildbot propose une automatisation de la construction de paquets.

## 8.5 Fonctionnalités

Comme tout les logiciels d'automatisation, Buildbot permet une automatisation complète.



FIGURE 4 – Logo d'Ansible Tower

## 8.6 Conclusions

# 9 Ansible Tower

## 9.1 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, créer 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 a quelques mois entre elles.

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

## 9.2 Historique et versions

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

### 9.3 Contexte d'utilisation

Ansible Tower a été créée pour permettre à Ansible (la plateforme de configuration et de gestion d'ordinateurs) d'être utilisée 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 également une console et un environnement WEB

### 9.4 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 et d'autres sources d'authentifications. Les outils de lignes de commandes sont facilement utilisables avec Jenkins.

### 9.5 Fonctionnalités

### 9.6 Conclusions

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

## 10 JobScheduler



FIGURE 5 – Logo de JobScheduler

## **10.1 Présentation**

JobScheduler est un ordonnancier en Open-Source de travaux d'informations et d'automatisation de processus.

Il est distribué sous la licence GPLv2

## **10.2 Historique et versions**

## **10.3 Fonctionnement**

## **10.4 Fonctionnalités**

## **10.5 Conclusions**

# **11 Rundeck**



FIGURE 6 – Logo de Rundeck

## **11.1 Présentation**

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.

## **11.2 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

## **11.3 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.

## **11.4 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

## 11.5 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 pour l'enregistrement de données telles que les jobs, les clés SSH, les utilisateurs qui ont accès à Rundeck.

## 11.6 Conclusions

# 12 Mise en place de Rundeck

## 12.1 Environnement de travail

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

Notre infrastructure de travail et de démonstration se fait à l'aide 4 machines virtuelles, mais peut cependant être à un nombre plus grand de machines virtuelles. La seule 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 choisi 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 boîte 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

## **12.2 Installation**

### **12.2.1 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

### **12.2.2 Windows**

### **12.2.3 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

## **12.3 Configuration**

### **12.3.1 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 sur les fichiers utilisés :

*framework.properties* : Fichier de configuration utilisé par les services de base de Rundeck

*rundeck-conf.properties* : Fichier de configuration de l'application WEB

*realm.properties* : Fichier de configuration des propriétés des utilisateurs

### 12.3.2 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-conf.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 sont *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 ajoutés depuis ce même fichier

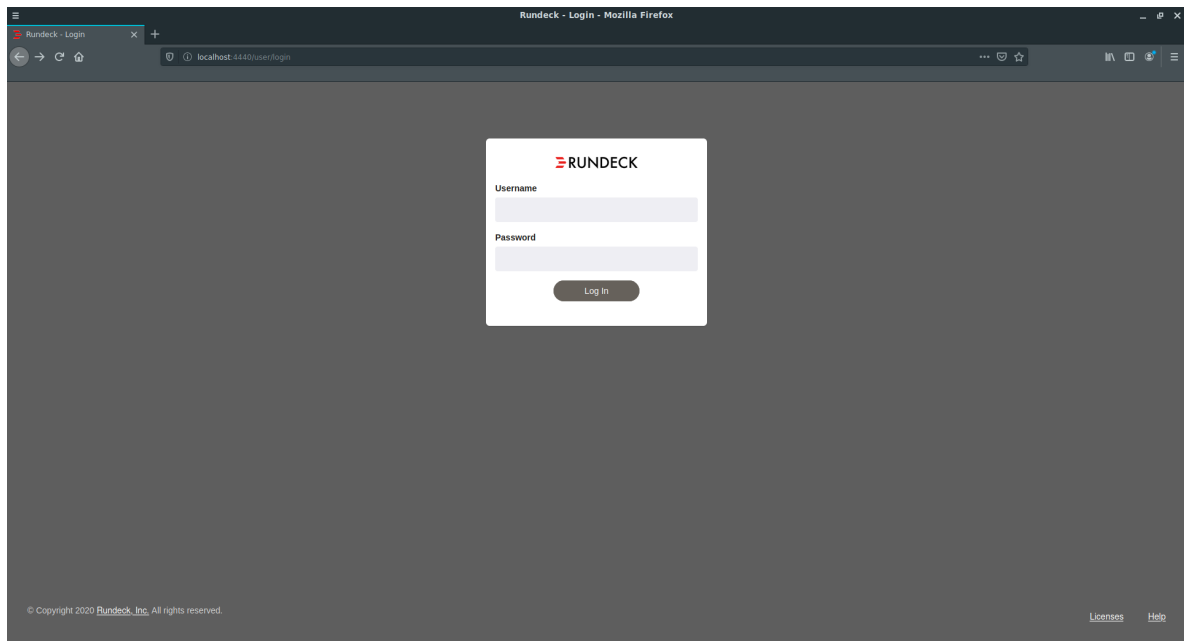


FIGURE 7 – Interface de connexion

*Username : admin*

*Password : admin*

### 12.3.3 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.

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**"

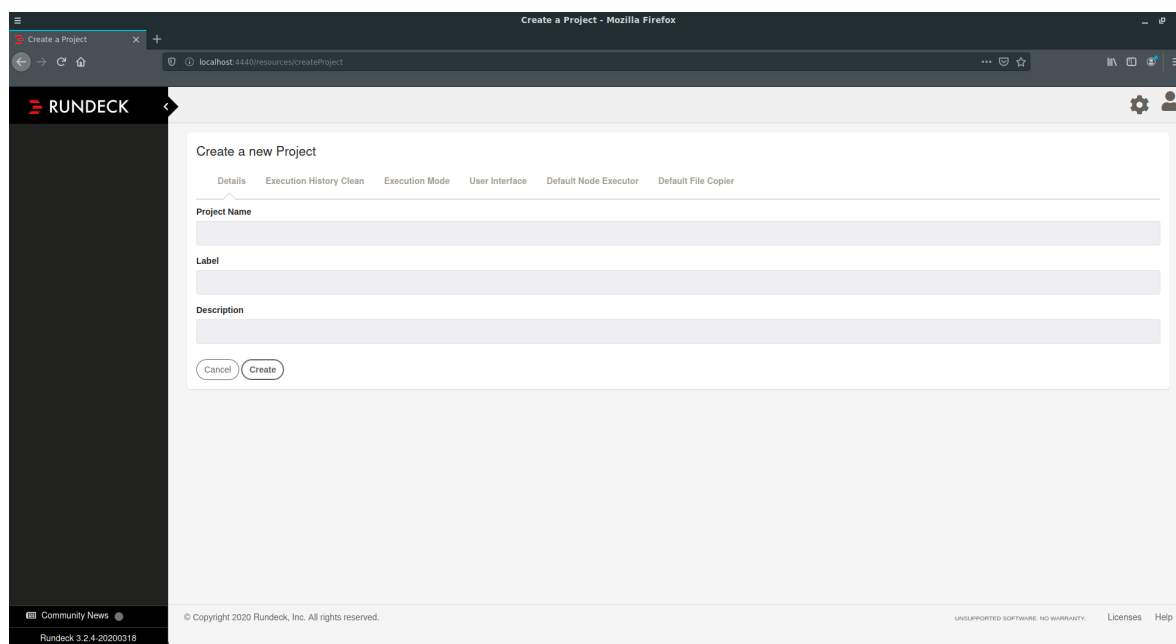


FIGURE 8 – Interface de connexion

— 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 affiché sur certaines page 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 télécharger 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 modifié l'utilitaire de transfert de fichiers d'une machine à une autre via Rundeck.

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

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



### 12.3.4 Définition de la méthode de stockage

Selon les choix de l'administrateur de Rundeck, plusieurs méthodes de stockages sont à disposition.

Méthode de stockage :

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

### 12.3.5 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.

### 12.3.6 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

### 12.3.7 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 défini 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 delais entre tentatives

## **13 Conclusions**

## 14 Bibliographie

- <https://docs.rundeck.com/docs/administration/install/>
- <https://docs.rundeck.com/docs/manual/>
- <http://docs.buildbot.net/current/index.html>
- <https://github.com/rundeck/rundeck>
- <https://docs.rundeck.com/docs/administration/configuration/config-file-reference.html>
- <https://tech.oyster.com/rundeck-vs-crontab-why-rundeck-won/>
- <https://www.edureka.co/blog/what-is-jenkins/>

## **15 Annexes**