

Rapport Projet Tuteuré Runderck

HEPPNER Tristan / ROYER Grégory
TORRES Yanis / AISSI Ayoub

Janvier 2020

Table des matières

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

9.6 Conclusions	13
10 JobScheduler	13
10.1 Présentation	13
10.2 Historique et versions	14
10.3 Contexte d'utilisation	14
10.4 Fonctionnement	14
10.5 Fonctionnalités	14
10.6 Conclusions	14
11 Cron	14
11.1 Présentation	14
11.2 Historique et versions	15
11.3 Contexte d'utilisation	15
11.4 Fonctionnement	15
11.5 Fonctionnalités	15
11.6 Conclusions	15
12 Mise en place de Rundeck	15
12.1 Environnement de travail	15
12.2 Installation	15
12.2.1 Exigences système	15
12.2.2 Windows	16
12.2.3 Linux	16
12.3 Configuration	16
12.3.1 Arborescence	16
12.3.2 URL d'accès	19
12.3.3 Définition de la méthode de stockage	19
12.3.4 Définition des nodes	20
12.3.5 Clés SSH	22
12.3.6 Définition des jobs	22
12.4 Utilisation	23
12.4.1 Pilotage de service	23
12.4.2 Création de jobs	23
13 Bibliographie	24
14 Annexes	25

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 Rundeck



FIGURE 1 – Logo de Rundeck

6.1 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.

6.2 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.

6.3 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.

6.4 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

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

6.6 Conclusions

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.

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 erreur.

7.5 Fonctionnalités

7.6 Conclusions

8 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 un alternative au projet de Mozilla : [Tindebox](#) Buildbot se démarque de ses concurrents notamment grâce à sa légèreté ainsi



FIGURE 3 – Logo de Buildbot

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.

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

8.5 Fonctionnalités

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.



FIGURE 4 – Logo d'Ansible Tower

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.

9.2 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.

9.3 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 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 ordonnanceur en Open-Source .

10.2 Historique et versions

10.3 Contexte d'utilisation

10.4 Fonctionnement

10.5 Fonctionnalités

10.6 Conclusions

11 Cron



FIGURE 6 – Logo de Crontab

11.1 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 un planificateur de tâches

11.2 Historique et versions

11.3 Contexte d'utilisation

11.4 Fonctionnement

11.5 Fonctionnalités

11.6 Conclusions

12 Mise en place de Rundeck

12.1 Environnement de travail

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 composée 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 URL d'accès

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

Cette URL peut, toutefois être modifiée, 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

12.3.3 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

12.3.4 Définition des nodes

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

Les nodes sont définis, communément, 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 imposée par Rundeck. On peut définir autant de nodes que l'on souhaite dans ce fichier.

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

Ci-dessous, la déclaration d'un node dans Rundeck, créée par l'utilisateur au format **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>
```

Ci-dessous, la déclaration d'un node dans Rundeck, créée par l'utilisateur au format **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 }

```

Ci-dessous, la déclaration d'un node dans Rundeck, créée par l'utilisateur au format **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.

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.

12.3.5 Clés SSH

Rundeck requiert les clés SSH de chaque machine qu'il a sous son contrôle

12.3.6 Définition des jobs

Un job est décomposé en plusieurs étapes (*step en anglais*).

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**"

Onglet "Details" :

Onglet "Workflow" :

Onglet "Nodes" :

Onglet "Schedule" :

Onglet "Notifications" :

Onglet "Others" :

12.4 Utilisation

12.4.1 Pilotage de service

12.4.2 Création de jobs

La création d'un job peut être fait directement sur l'interface Rundeck. Rundeck propose également à l'utilisateur de pouvoir exécuter un script provenant d'un site web externe, ou même de pouvoir exécuter un script déjà écrit à l'avance.

13 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/>

14 Annexes