Gestion de tâches centralisées avec RunDeck

RunDeck est un outil centralisé de gestion de tâches (cron par exemple). Il est écrit en Java et gère la connexion vers ses «nodes» en SSH. C'est un outil très puissant avec divers plugins intéressants permettant entre autre l'interconnexion avec des outils de gestion de configurations (Salt, Puppet …).

Nous allons donc voir comment mettre en place un serveur RunDeck et le relier à différents nodes afin de programmer divers tâches. Dans la plupart des cas, il est utilisé pour automatiser des déploiements ou des tests unitaires cependant on peut également l'utiliser pour des tâches classiques utilisées dans les cron jobs.

On va utiliser le paquet *.deb fournit par le site officiel pour une installation sur un serveur Debian / Ubuntu :

```
wget dl.bintray.com/rundeck/rundeck-deb/rundeck-2.6.2-1-GA.deb

apt-get install openjdk-7-jdk

dpkg -i rundeck-2.6.2-1-GA.deb
```

L'installation est terminée !

RunDeck fournit une interface Web, une interface en ligne de commande et une API Web. Nous allons utiliser principalement l'interface web pour gérer nos tâches.

Le service web écoute par défaut sur **localhost**, on va donc éditer le fichier de configuration **/etc/rundeck/rundeck-config.properties** et modifier cette ligne :

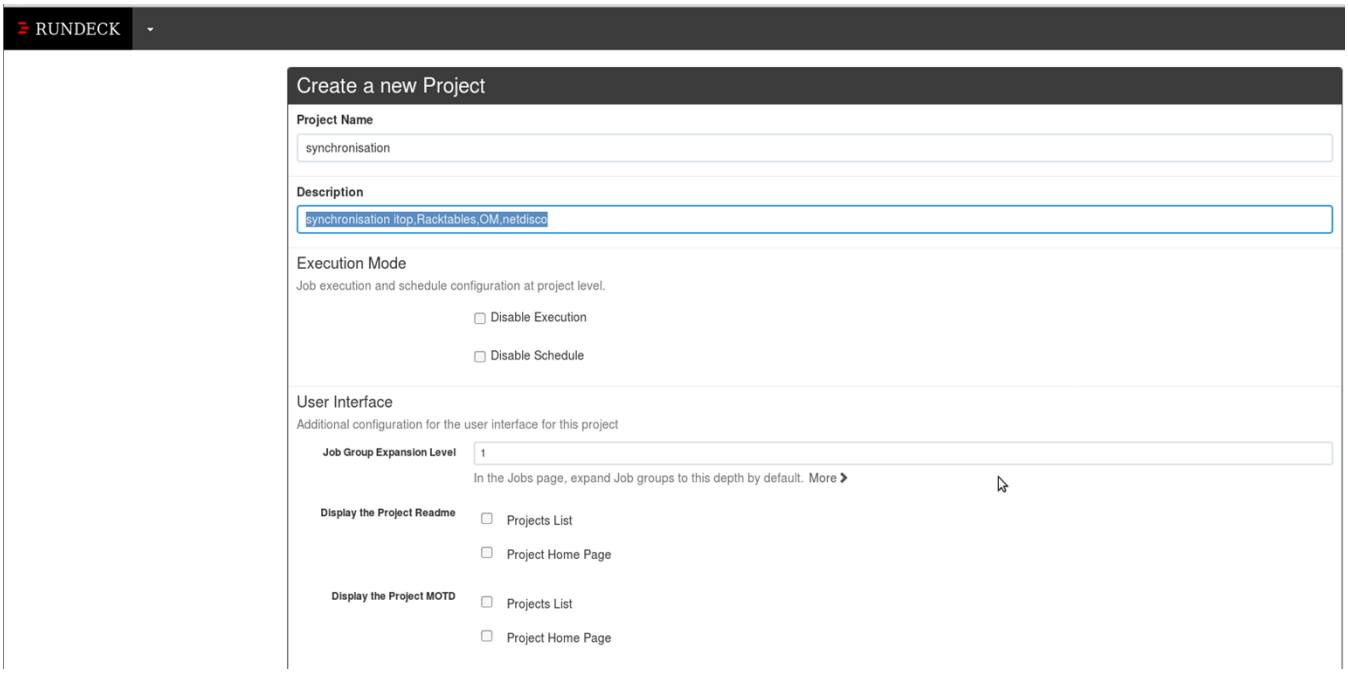
```
# change hostname here
 grails.serverURL=http://your-server:4440
```

On va maintenant redémarrer RunDeck pour qu'il prennent cela en compte :

```
service rundeckd restart
```

Il faut attendre un petit peu que le service soit démarré et vous devriez pouvoir joindre l'interface web sur le port 4440. Le login par défaut est **admin;admin**, n'oubliez pas de le changer.

On va donc commencer par créer un projet :



Laissez les paramètres par défaut si ils vous conviennent, ce sera très bien.
RunDeck utilise une clé privée SSH pour se connecter à ses nodes, elle est situé à **/var/lib/rundeck/.ssh/id_rsa**.

La liste des nodes se situe dans le fichier **/var/rundeck/projects/synchronisation/etc/resources.xml**, c'est ici qu'on ajoute les serveurs qu'on souhaitent administrer.

```
<?xml version="1.0" encoding="UTF-8"?>

<project>
  <node name="rundek-win-pp" description="Rundeck server node" tags="" hostname="rundek-win-pp.in.ac-versailles.fr" osArch="amd64" osFamily="unix" osName="L$


  <node name="racktables-server" description="Racktables node" tags="" hostname="racktables-server.in.ac-versailles.fr" osArch="amd64" osFamily="unix" osNam$


</project>
```

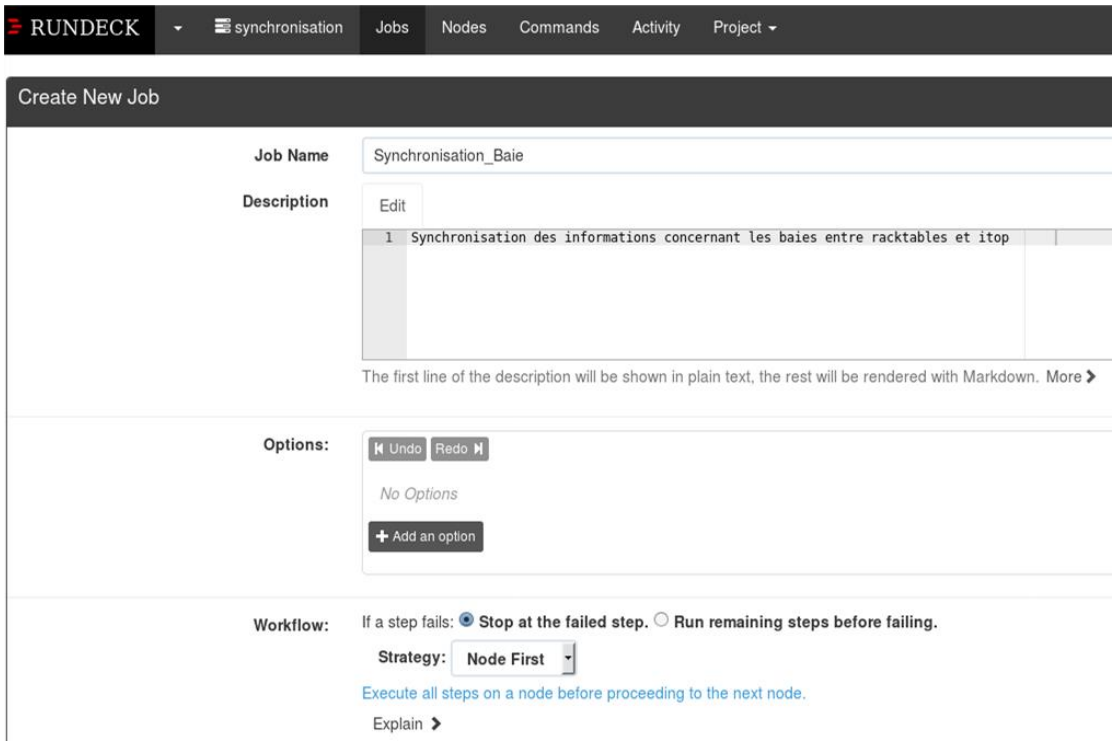
Dans cet exemple de projet, le serveur Rundeck est relié au serveur racktables et à lui même (pour gérer ses propres tâches). Il suffit maintenant d'ajouter la clé publique du serveur Rundeck sur le serveur racktables à manager:

```
ssh-copy-id -i /var/lib/rundeck/.ssh/id_rsa.pub root@racktables-server
```

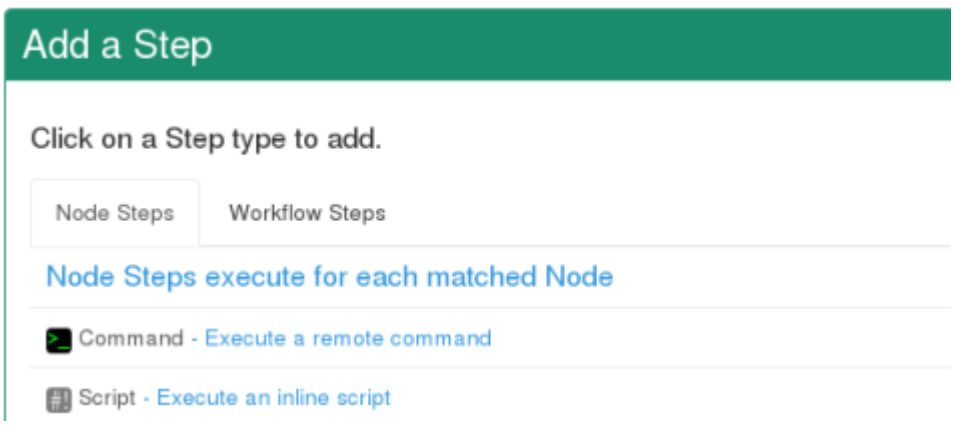
Le serveur est donc joignable par Rundeck sans saisir de mot de passe ! On peut vérifier avec cette commande qui devrait vous connecter directement :

ssh root@racktables-server -i /var/lib/rundeck/.ssh/id_rsa

Maintenant qu'ils sont reliés, on va pouvoir créer nos premières tâches à effectuer sur ces serveurs :



Le fonctionnement des Jobs sous RunDeck s'effectue en étapes (Steps), elles peuvent être l'exécution d'une commande, la lancement d'un script à distance ou même l'exécution d'un autre job. On ajoute donc une étape pour exécuter un script à distance :



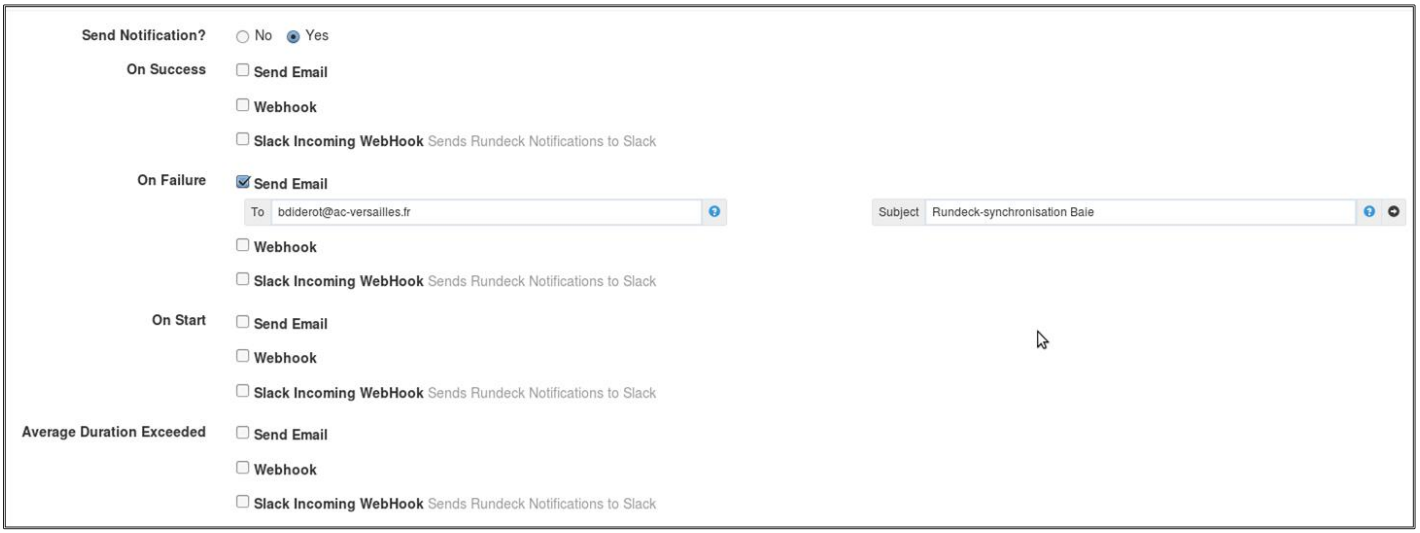
Nous allons prendre l'exemple que nous souhaitons lancer la commande ./baie.sh qui permet d' exécuter le script de synchronisation des données concernant les emplacements des serveurs dans le baie. Pour cela on ajoute donc une étape **«Command»**.

On a ensuite la possibilité:

-de choisir sur quel node l'exécuter (**locally** ou **dispatch to nodes**),on va choisir d'exécuter le script sur le nœud ***racktables-server***

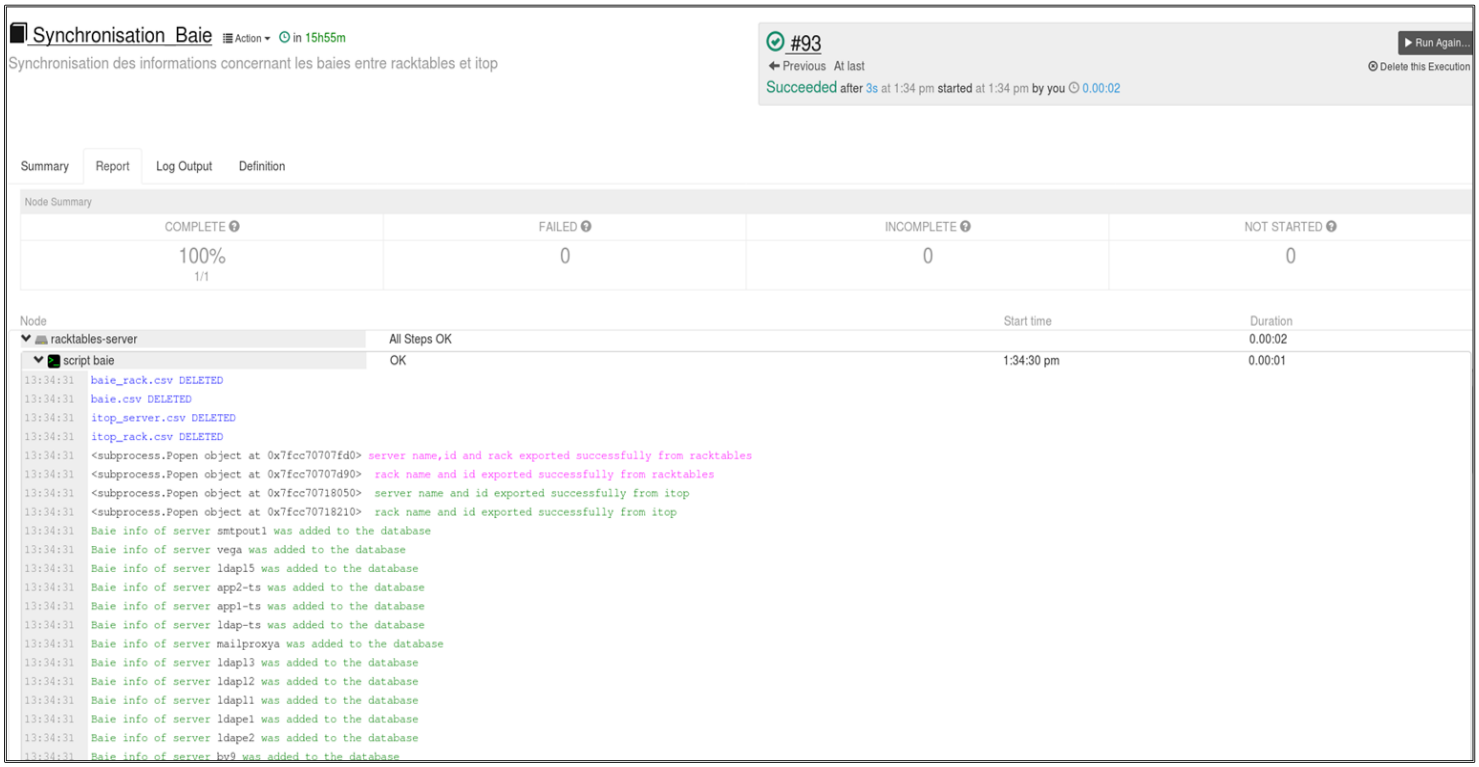
-de choisir une fréquence de répétition (**schedule to run**), on va programmer l'exécuter du script quotidiennement à 5h30. On peut bien sur utiliser une syntaxe Crontab à la place.

-de recevoir des notifications ou de relancer en cas d'échec, on va configurer le rundeck pour qu'on reçoit une notification lors d'un échec.



Maintenant que notre job est créé, on va pouvoir le trouver dans notre liste de jobs et l'exécuter :

Notre job semble s'être correctement exécuté,



Maintenant que vous savez installer Rundeck et créer un job, vous pourrez remplacer peu à peu les centaines de tâches cron qui moisissent sur vos serveurs. RunDeck est génial car il permet de créer plusieurs projets en fonction de ses besoins (Dev, Ops, Tests …) et de gérer plein de nodes ce qui ajoute une grosse couche d'automatisation aux tâches récurrentes. Il est également possible d'installer plein de plugins qui étendent les possibilités de RunDeck (Saltstack, Puppet, Hipchat, Chef, Fluentd …).

lien utile: https://blog.ouvrard.it/2016/02/25/gestion-de-taches-centralisees-avec-rundeck/