

Guide du développeur pour la migration « lift and shift » des applications vers le cloud

Table des matières

1 Introduction.....	2
2 Simplifier le processus de migration avec Red Hat Application Migration Toolkit	3
2.1 Analyser une application Java EE.....	3
2.2 Comprendre le rapport.....	6
2.3 Résoudre les problèmes.....	8
2.4 Tester à nouveau la version.....	8
3 Exécuter le projet dans Red Hat JBoss Enterprise Application Platform	10
3.1 Installer Red Hat JBoss Enterprise Application Platform.....	10
3.2 Le plug-in WildFly Maven	10
3.3 Configurer l'application.....	10
3.4 Déployer l'application.....	10
3.5 Tester l'application	10
3.6 Arrêter l'application.....	11
4 Déployer le monolithe sur Red Hat OpenShift Container Platform	11
4.1 Ajouter un profil Red Hat OpenShift	11
4.2 Créer un projet Red Hat OpenShift	12
4.3 Déployer l'application monolithique.....	14
4.4 Déployer l'application à l'aide de la version binaire.....	15
5 Synthèse.....	17



Introduction

Face à l'adoption croissante des modèles de cloud, de nombreuses entreprises cherchent à passer au développement cloud-native. Les services et les modèles de cloud permettent de développer entièrement de nouvelles applications grâce aux microservices, aux équipes de développement autonomes, au déploiement continu et agile ainsi qu'aux déploiements de clouds conteneurisés et orchestrés. Toutefois, la réécriture complète de toutes les applications existantes est rarement possible pour des questions de temps et d'argent.

Pour rester agiles et compétitives, les entreprises doivent migrer et moderniser leurs applications Java™ et adopter le développement d'applications cloud-native. Pour la plupart des entreprises, ce processus nécessite de réutiliser au maximum les fonctionnalités et données existantes, de déplacer des charges de travail vers une plateforme de déploiement moderne et, enfin, de mettre en œuvre de nouveaux processus, produits et technologies.

Le modèle de modernisation « lift-and-shift » représente un premier pas intéressant vers le développement d'applications cloud-native. Il implique les pratiques suivantes :

- La conteneurisation des charges de travail monolithiques existantes
- Le déploiement des charges de travail sur une solution PaaS (Platform-as-a-Service)
- La conservation des données et des intégrations externes sur la plateforme existante

Une fois ces opérations accomplies, les développeurs peuvent commencer le processus de « démantèlement du monolithe » afin de remplacer progressivement certaines fonctionnalités de l'application par des microservices.

Ce guide décrit le processus « lift-and-shift » à l'aide des logiciels Red Hat® détaillés dans la Figure 1 ci-dessous. Ce processus comprend les étapes suivantes :

1. Analyse d'une application monolithique à l'aide de [Red Hat Application Migration Toolkit](#) et migration vers des interfaces standard ouvertes
2. Modification du code et de la configuration en vue de leur exécution sur [Red Hat JBoss® Enterprise Application Platform](#).
3. Déploiement du monolithe en tant que conteneur sur [Red Hat OpenShift® Container Platform](#) où l'application peut bénéficier de fonctions telles que la mise à l'échelle automatique, la mise en cluster intégrée et le basculement.

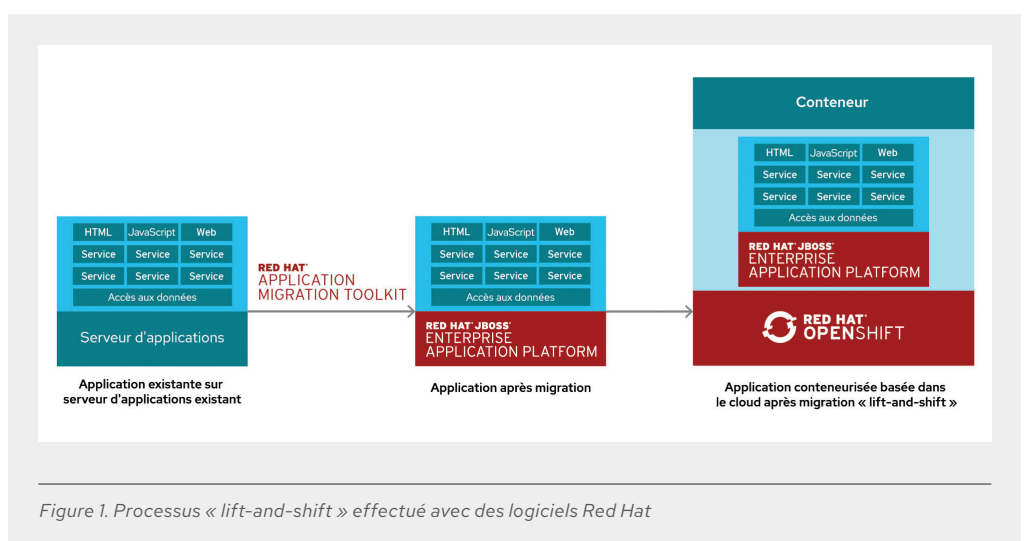


Figure 1. Processus « lift-and-shift » effectué avec des logiciels Red Hat

Une fois l'application déployée sur la solution Red Hat OpenShift Container Platform, vous pouvez utiliser d'autres outils pour convertir des fonctionnalités spécifiques en microservices. Ce processus peut servir à diviser les fonctionnalités afin de bénéficier de davantage de parallélisation et d'autonomie, avec une réduction supplémentaire des délais de rentabilisation grâce à des outils tels que :

- [Thorntail/MicroProfile](#)
- [Spring Boot](#)
- [Node.js](#)
- [Eclipse Vert.x](#)

Simplifier le processus de migration avec Red Hat Application Migration Toolkit

Red Hat Application Migration Toolkit est un outil extensible et personnalisable basé sur des règles qui permet de simplifier la migration des applications Java. Les entreprises l'utilisent pour les tâches suivantes :

- Planifier et évaluer des tâches
- Identifier les problèmes de migration et proposer des solutions
- Analyser des rapports détaillés
- Utiliser des règles et des parcours de migration intégrés
- Proposer des extensions et des personnalisations de règles
- Analyser du code source ou des archives d'applications

La solution Red Hat Application Migration Toolkit recherche les ressources communes et identifie les technologies et les points problématiques connus lors de la migration des applications. Elle offre une vue d'ensemble des technologies utilisées par chaque application et propose un rapport détaillé que les entreprises peuvent exploiter pour analyser, documenter et migrer des applications Java EE vers la plateforme Red Hat JBoss Enterprise Application Platform.

Remarque : Red Hat Application Migration Toolkit s'intègre généralement dans le cadre d'un programme plus vaste de migration et de modernisation des applications. De tels programmes sont souvent constitués de phases bien définies et reproductibles qui s'étalent sur plusieurs semaines ou mois, et peuvent demander l'intervention de nombreux salariés au sein de chaque entreprise.

Pour en savoir plus sur la philosophie de Red Hat et ses méthodes éprouvées, consultez la documentation relative à [Red Hat Application Migration Toolkit](#) et la [page d'accueil pour les développeurs](#).

Analyser une application Java EE

La solution Red Hat Application Migration Toolkit examine les artefacts d'application, notamment les répertoires sources des projets et les archives d'applications. Elle génère alors un rapport HTML indiquant les domaines où des changements sont nécessaires. La solution peut ensuite servir à migrer des applications Java depuis les versions précédentes de la plateforme Red Hat JBoss Enterprise Application Platform (EAP) ou depuis d'autres plateformes de serveurs d'applications, telles qu'Oracle WebLogic ou IBM® WebSphere.

Vous pouvez utiliser Red Hat Application Migration Toolkit de plusieurs façons différentes :

- **Console web.** La console web de Red Hat Application Migration Toolkit permet à une équipe de développeurs d'identifier les opérations de migration et de modernisation nécessaires et d'en établir l'ordre de priorité lorsqu'elles portent sur un grand nombre d'applications. Elle vous permet de regrouper les applications par projets pour en faciliter l'analyse, et génère différents rapports qui présentent les résultats.

- **Interface en ligne de commande.** L'interface en ligne de commande est un outil qui permet aux développeurs d'identifier les opérations de migration et de modernisation nécessaires et d'en établir l'ordre de priorité. Elle génère différents rapports qui présentent les résultats de l'analyse. La simplicité de l'interface en ligne de commande en fait un outil idéal pour l'analyse d'une seule application.
- **Plug-in Eclipse.** Le plug-in Eclipse pour Red Hat Application Migration Toolkit fournit de l'aide directement dans Eclipse et Red Hat Developer Studio aux développeurs qui apportent des modifications dans le cadre d'une opération de migration ou de modernisation. Il analyse vos projets à l'aide de l'outil Red Hat Application Migration Toolkit, signale les problèmes de migration dans le code source, donne des instructions pour les résoudre et fournit automatiquement du code de remplacement lorsque c'est possible.

Toutes ces interfaces sont disponibles en téléchargement sur la page de la solution [Red Hat Application Migration Toolkit](#). Pour obtenir des instructions d'installation, veuillez consulter la documentation correspondant à chacune de ces interfaces. Si vous prévoyez plusieurs utilisateurs, la console web est une bonne option. Pour concevoir ce document, nous avons utilisé l'interface en ligne de commande pour plus de simplicité. Les étapes qui suivent résumant l'utilisation de l'outil Red Hat Application Migration Toolkit.

1. Vérifiez l'installation de l'interface en ligne de commande de Red Hat Application Migration Toolkit

Une fois l'interface en ligne de commande de Red Hat Application Migration Toolkit installée, vérifiez qu'elle fonctionne correctement en exécutant la commande suivante :

```
$ ${HOME}/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli --version
```

Vous devriez obtenir un résultat similaire à ceci :

```
Using RHAMT at /root/rhamt-cli-4.0.0.Beta4
> Red Hat Application Migration Toolkit (RHAMT) CLI, version 4.0.0.Beta4.
```

2. Exécutez l'interface en ligne de commande de Red Hat Application Migration Toolkit dans votre projet

Cette interface dispose de plusieurs options pour contrôler son fonctionnement. Ici, l'interface en ligne de commande est utilisée pour un projet appelé « monolith » et génère un rapport du même nom. Utilisez le nom de votre projet à la place de « monolith » dans l'expression `~/projects/monolith` ci-dessous.

```
$ ~/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli \
  --sourcemode \
  --input ~/projects/monolith \
  --output ~/rhamt-reports/monolith \
  --overwrite \
  --source weblogic \
  --target eap:7 \
  --packages com.redhat weblogic
```

Notez que nous avons utilisé les options `--source` et `--target`. Ces options vous permettent de cibler des parcours de migration spécifiques pris en charge par Red Hat Application Migration Toolkit. Les parcours de migration comprennent les éléments suivants :

- Oracle WebLogic Server
- IBM WebSphere Application Server

- Red Hat JBoss EAP 5/6/7

Attendez que l'outil ait terminé ses opérations avant de poursuivre. Le résultat devrait se présenter sous la forme suivante :

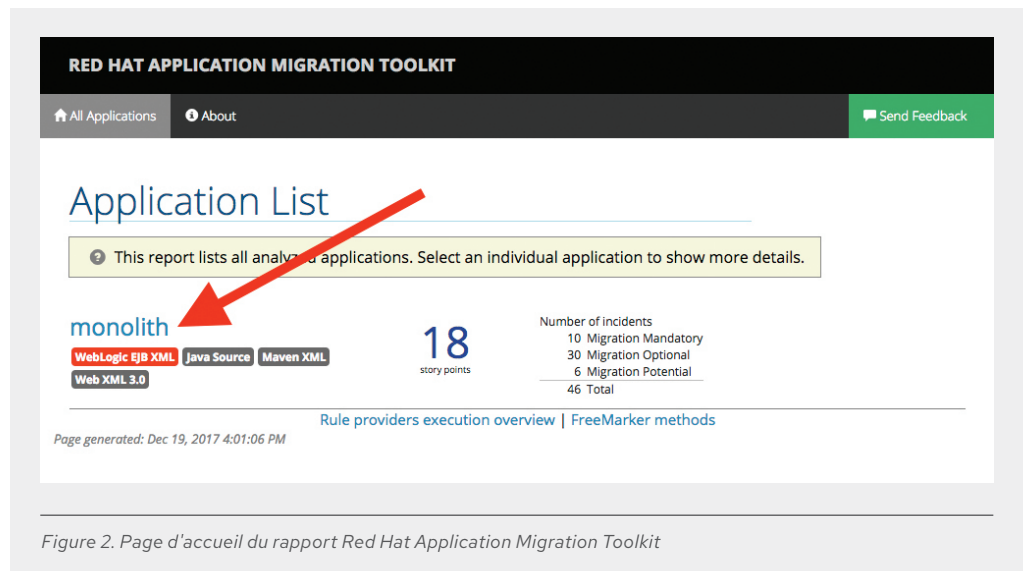
```
Report created: ~/rhamt-reports/monolith/index.html
```

3. Consultez les résultats

Pour afficher le rapport, accédez au fichier suivant à l'aide de votre navigateur local :

```
~/rhamt-reports/monolith/index.html
```

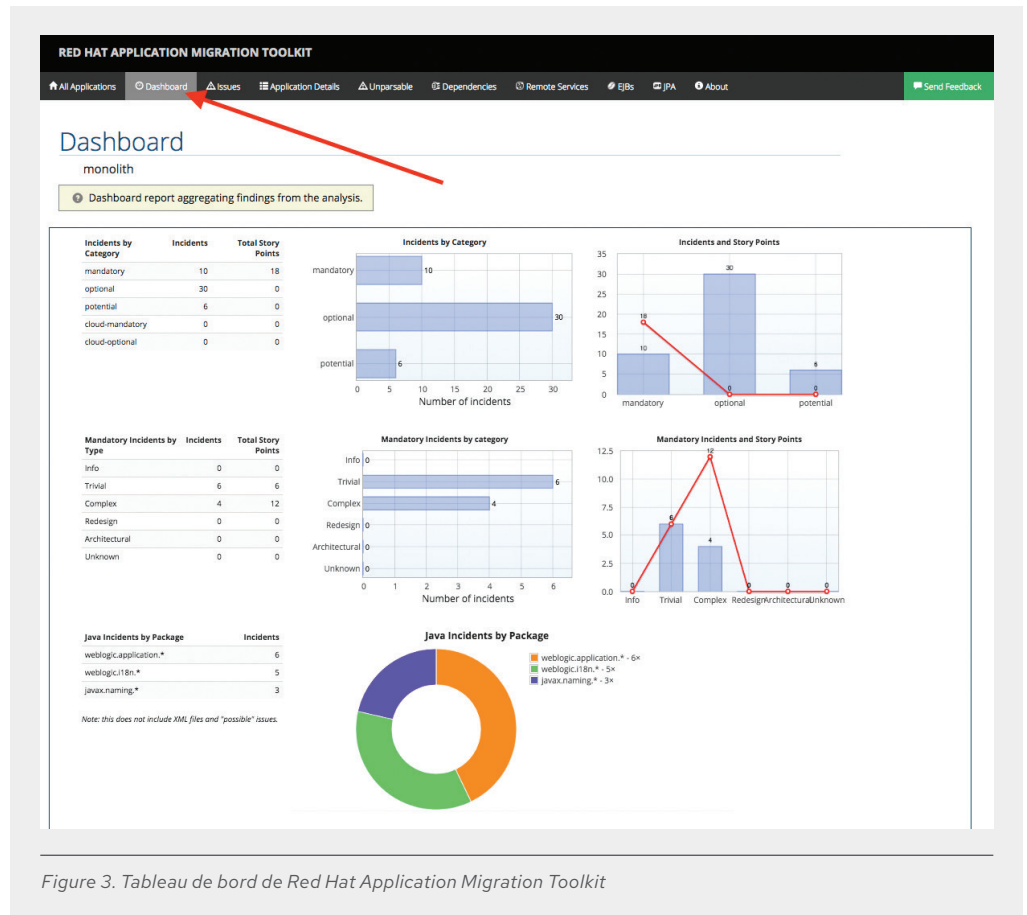
Vous devriez voir la page d'accueil du rapport, comme indiqué dans la Figure 2. Cette page présente la liste des applications traitées et des problèmes rencontrés. Chaque ligne de la page contient un aperçu synthétique des Story Points, du nombre d'incidents et des technologies découvertes dans l'application.



The screenshot shows the 'Application List' page of the Red Hat Application Migration Toolkit. The page title is 'Application List'. A navigation bar at the top includes 'All Applications' and 'About', with a 'Send Feedback' button on the right. A message box states: 'This report lists all analyzed applications. Select an individual application to show more details.' The main content area features a card for the application 'monolith'. This card includes a red arrow pointing to the application name, a list of technologies (WebLogic EJB XML, Java Source, Maven XML, Web XML 3.0), a large number '18' representing 'story points', and a summary of incidents: 10 Migration Mandatory, 30 Migration Optional, 6 Migration Potential, and 46 Total. At the bottom of the card, there are links for 'Rule providers execution overview' and 'FreeMarker methods'. The footer of the page indicates 'Page generated: Dec 19, 2017 4:01:06 PM'.

Figure 2. Page d'accueil du rapport Red Hat Application Migration Toolkit

Dans cet exemple, en cliquant sur le lien « monolith » vous accédez au tableau de bord du projet, comme illustré dans la Figure 3.



Comprendre le rapport

Le tableau de bord donne un aperçu de l'ensemble des opérations de migration des applications. Il résume :

- Les incidents et Story Points par catégorie
- Les incidents et les Story Points selon le niveau d'effort requis pour les modifications suggérées
- Les incidents associés par paquet

Remarque : les « Story Points » sont un outil de mesure abstrait utilisé dans le développement agile de logiciels pour estimer la quantité relative de travail requise pour ajouter une fonction ou apporter une modification. L'outil Red Hat Application Migration Toolkit utilise des Story Points pour indiquer la quantité relative de travail requise pour la migration de certaines composantes d'une application et de l'application dans son intégralité. La quantité de travail varie énormément en fonction de la taille et de la complexité des applications à migrer.

Le menu situé en haut de l'écran contient plusieurs autres onglets.

- **All Applications** fournit une liste de toutes les applications analysées.
- **Dashboard** fournit un aperçu d'une application en particulier.
- **Issues** rassemble tous les problèmes à examiner.
- **Application Details** fournit des informations détaillées sur toutes les ressources de l'application qu'il recommandé de surveiller pendant la migration.
- **Unparsable** affiche tous les fichiers que Red Hat Application Migration Toolkit n'a pas pu analyser dans le format attendu. Par exemple, un fichier portant le suffixe .xml ou .wsdl est censé être un fichier XML. Si l'analyseur XML échoue, le problème est signalé dans cette section et dans la liste des fichiers individuels.
- **Dependencies** affiche toutes les dépendances présentes sous forme de paquets Java identifiées dans l'application.
- **Remote Services** affiche toutes les références à des services distants trouvées dans l'application.
- **EJBs** contient une liste des composants EJB identifiés dans l'application.
- **JBPM** contient toutes les ressources JBPM découvertes lors de l'analyse.
- **JPA** présente toutes les ressources JPA trouvées dans l'application.
- **About** décrit la version actuelle de Red Hat Application Migration Toolkit et fournit des liens utiles pour obtenir de l'aide.

Remarque : certaines sections décrites ci-dessus peuvent ne pas apparaître en fonction des problèmes effectivement détectés dans le projet.

Maintenant que le rapport de Red Hat Application Migration Toolkit est disponible, vous pouvez lancer la migration de l'application

Résoudre les problèmes

L'onglet « Issues » de la Figure 4 fournit une liste plus détaillée de tous les problèmes identifiés dans l'application analysée. Il contient des liens utiles pour permettre aux utilisateurs de mieux comprendre le problème et donne des indications sur la migration et la modernisation.

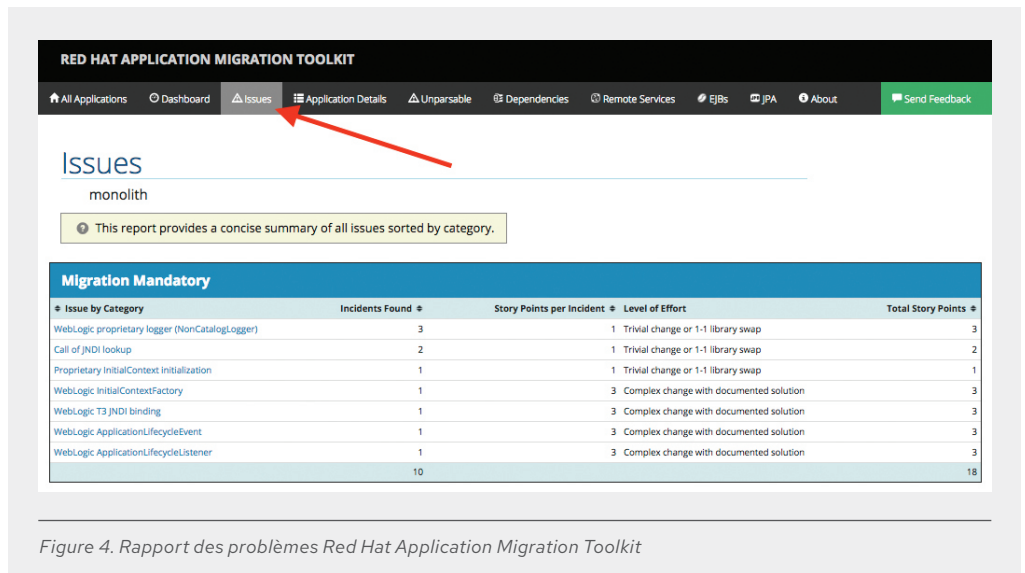


Figure 4. Rapport des problèmes Red Hat Application Migration Toolkit

De nombreuses applications contiennent du code propre à une plateforme donnée, que vous devrez mettre à jour pour utiliser les interfaces standard Java EE. Cette étape concerne :

- **Le code de démarrage** pour lancer des fonctions ou planifier des tâches au démarrage ou à l'arrêt de l'application
- **Le code de journalisation** qui utilise des fonctions de journalisations propres à une plateforme donnée
- **Le code de messagerie** qui utilise des mécanismes obsolètes ou propres à une plateforme donnée

Dans chaque cas, le rapport sur les problèmes proposera des solutions possibles. En pratique, la résolution de certains de ces problèmes peut nécessiter une réflexion supplémentaire dans le cadre de la migration. Toutefois, l'application de ces changements améliore significativement la portabilité du code.

Une fois les modifications nécessaires apportées, vous pouvez utiliser Maven pour créer l'application et la mettre en paquet afin de vous assurer que la compilation du code modifié s'effectue toujours correctement :

```
$ mvn clean package
```

Si la version exécutable fonctionne (le message BUILD SUCCESS s'affiche), vous pouvez passer au problème suivant dans le rapport.

Tester à nouveau la version

Une fois que tous les problèmes ont été résolus, vous devez exécuter Red Hat Application Migration Toolkit à nouveau pour vérifier que la migration s'est réalisée correctement.

1. Exécutez l'interface en ligne de commande Red Hat Application Migration Toolkit dans le projet

Exécutez la commande ci-dessous pour nettoyer les artefacts de la version précédente, lancer à nouveau Red Hat Application Migration Toolkit et analyser le nouveau projet. Comme précédemment, remplacez l'expression `~/projects/monolith` par l'emplacement et le nom de votre projet.

```
$ mvn clean && \  
~/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli \  
  --sourceMode \  
  --input ~/projects/monolith \  
  --output ~/rhamt-reports/monolith \  
  --overwrite \  
  --source weblogic \  
  --target eap:7 \  
  --packages com.redhat weblogic
```

Une fois le processus terminé, l'interface en ligne de commande devrait afficher :

```
Report created: /root/rhamt-reports/monolith/index.html
```

2. Consultez les résultats

Rechargez la page web locale du rapport à l'adresse :

```
~/rhamt-reports/monolith/index.html
```

Si la migration est réussie, le nombre de Story Points signalé pour l'application devrait être égal à zéro (0), ce qui signifie que sa modernisation s'est déroulée correctement et que vous pouvez la déplacer vers la plateforme Red Hat JBoss EAP.

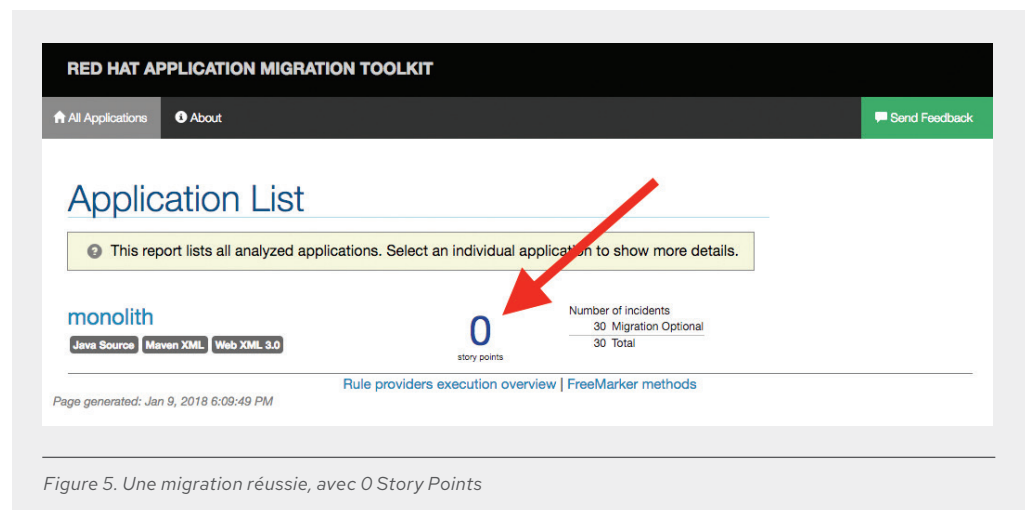


Figure 5. Une migration réussie, avec 0 Story Points

Exécuter le projet sur Red Hat JBoss Enterprise Application Platform

Maintenant que l'application est mise à jour et peut utiliser des interfaces Java EE standard, vous pouvez la déployer, la tester et découvrir certaines des fonctions proposées par la plateforme Red Hat JBoss EAP.

Installer Red Hat JBoss Enterprise Application Platform

Téléchargez la dernière version de [Red Hat JBoss EAP](#). Ensuite, installez Red Hat JBoss EAP localement en exécutant la commande suivante dans la fenêtre de terminal. Modifiez le nom du fichier pour qu'il corresponde à la version téléchargée.

```
$ unzip -d $HOME $HOME/jboss-eap-7.2.0.zip
```

Ensuite, définissez la variable d'environnement JBOSS_HOME :

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0
```

Voilà, vous avez installé Red Hat JBoss EAP.

Le plug-in WildFly Maven

La plateforme Red Hat JBoss EAP fournit un outil [wildfly-maven-plugin](#) qui permet de l'arrêter, de la démarrer, de la déployer et de la configurer directement depuis Apache Maven. Une fois l'outil maven-plugin configuré, il peut être utilisé pour :

- Choisir entre Java EE uniquement ou Java EE Web Profile
- Configurer les ressources des bases de données
- Configurer des services tels que Java Message Service Topic

Toutes ces fonctionnalités sont configurées dans le fichier `pom.xml`.

Configurer l'application

Une fois analysée par Red Hat Application Migration Toolkit, l'application est en grande partie basée sur des normes. Elle peut être configurée en lançant Red Hat JBoss EAP pour démarrer, ajouter des ressources et arrêter l'application, comme suit :

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0 ; \ mvn wildfly:start wildfly:add-resource wildfly:shutdown
```

Cette opération devrait se conclure par le message `BUILD SUCCESS`.

Remarque : nous devons utiliser `wildfly:start` et `wildfly:shutdown`, car la commande `add-resource` requiert un serveur d'application actif. Une fois cette ressource ajoutée, il ne sera plus nécessaire d'exécuter cette commande.

Déployer l'application

Vous êtes prêt à déployer l'application :

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0 ; mvn wildfly:run
```

Une fois que le serveur démarre, il doit afficher :

```
Deployed "ROOT.war" (runtime-name: "ROOT.war")
```

Tester l'application

Accédez à l'application en chargeant l'URL suivante dans votre navigateur :

```
http://localhost:8080
```

Le cas échéant, des avertissements s'afficheront dans la console.

Arrêter l'application

Avant de passer à la suite, arrêtez l'application en saisissant CTRL-C dans la fenêtre de terminal.

Déployer le monolithe sur la plateforme Red Hat OpenShift

Une fois que l'application fonctionne sur la plateforme Red Hat JBoss EAP, vous pouvez déplacer le monolithe que vous venez de mettre à jour vers la solution Red Hat OpenShift Container Platform. Pour les étapes suivantes, vous devez avoir accès à un cluster Red Hat OpenShift Container Platform. Si ce n'est pas le cas, vous pouvez [installer Red Hat OpenShift Container Platform](#) localement.

Ajouter un profil Red Hat OpenShift

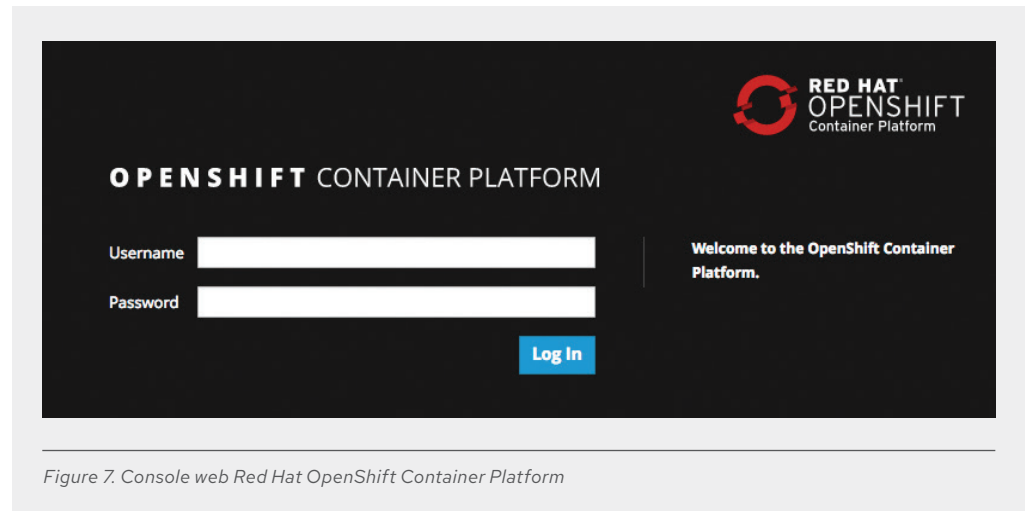
Ouvrez le fichier `pom.xml`, puis ajoutez un profil Red Hat OpenShift.

```
<profile>
  <id>openshift</id>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-war-plugin</artifactId>
        <version>2.6</version>
        <configuration>
          <webResources>
            <resource>
              <directory>${basedir}/src/main/webapp/WEB-INF</directory>
              <filtering>true</filtering>
              <targetPath>WEB-INF</targetPath>
            </resource>
          </webResources>
          <outputDirectory>deployments</outputDirectory>
          <warName>ROOT</warName>
        </configuration>
      </plugin>
    </plugins>
  </build>
</profile>
```

Figure 6. Profil Red Hat OpenShift (fichier `pom.xml`)

Créer un projet Red Hat OpenShift

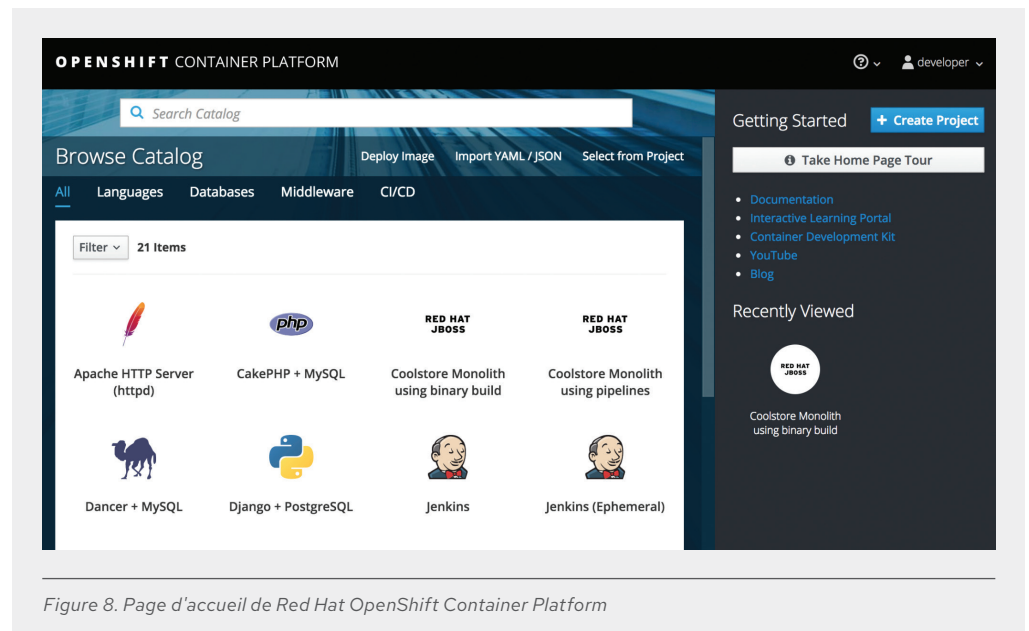
Pour commencer, ouvrez la console web Red Hat OpenShift Container Platform.



Connectez-vous à l'aide de vos identifiants Red Hat OpenShift :

- Nom d'utilisateur : XXXXXX
- Mot de passe : XXXXXX

Vous accédez alors à la page d'accueil de Red Hat OpenShift Container Platform :



Cliquez sur Create Project, renseignez les champs, puis cliquez sur Create. Dans cet exemple, l'application est appelée « coolstore ».

- Nom : coolstore-dev
- Nom à afficher : Coolstore Monolith - Dev
- Description : [laisser ce champ vide]

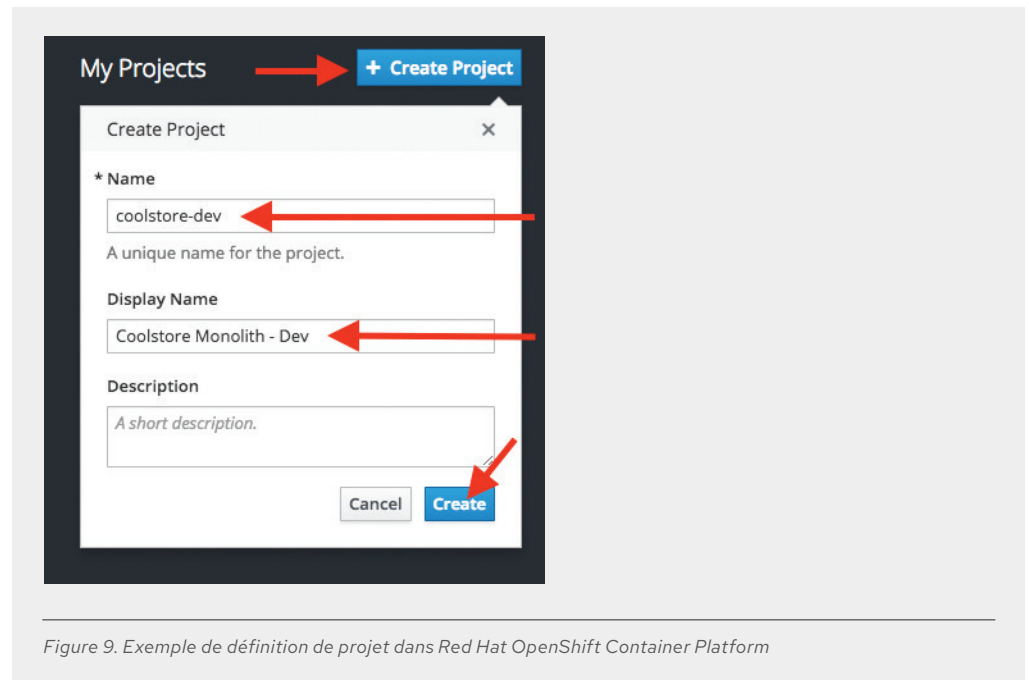


Figure 9. Exemple de définition de projet dans Red Hat OpenShift Container Platform

Cliquez sur le nom du projet que vous venez de créer :

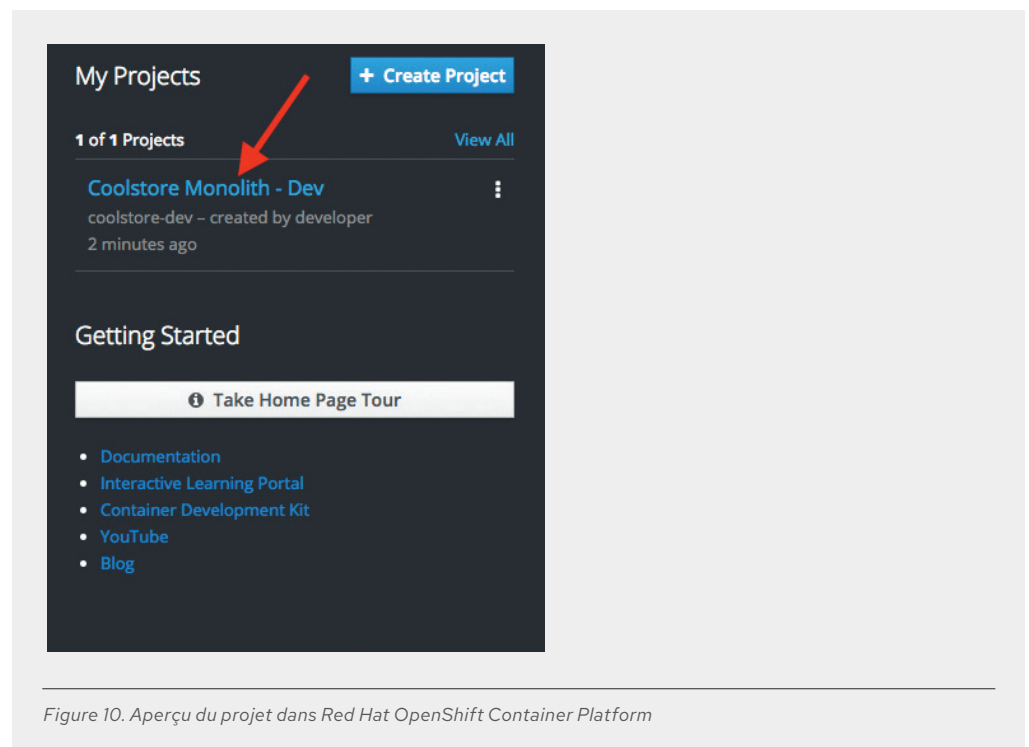


Figure 10. Aperçu du projet dans Red Hat OpenShift Container Platform

Vous accéderez ainsi à l'aperçu du projet. Cet emplacement est vide pour le moment, mais cela va bientôt changer.

Déployer l'application monolithique

Utilisez l'interface en ligne de commande pour déployer des composantes de l'application monolithique. Pour déployer le modèle monolithique à l'aide de l'interface en ligne de commande, exécutez les commandes ci-dessous.

Tout d'abord, basculez sur le projet de développeur que vous avez créé plus tôt :

```
$ oc project coolstore-dev
```

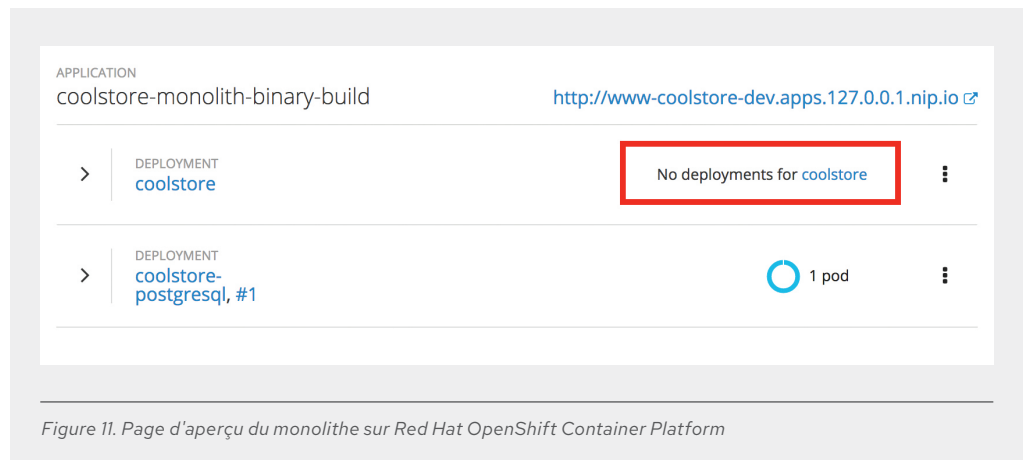
Ensuite, déployez le modèle :

```
$ oc new-app coolstore-monolith-binary-build
```

Cette opération permet de déployer toutes les bases de données requises par l'application ainsi que Red Hat JBoss EAP, mais elle ne démarre pas de version exécutable pour l'application.

Ensuite, ouvrez la page de l'aperçu du monolithe à l'adresse suivante :

`https://$OPENSIFT_MASTER/console/project/coolstore-dev/` puis vérifiez que les éléments du modèle du monolithe ont bien été créés.



Vous pouvez voir les composants en cours de déploiement dans l'aperçu du projet, mais notez qu'aucun déploiement n'existe pour « coolstore ». Vous n'avez pas encore déployé l'image du conteneur créée lors des étapes précédentes, mais vous le ferez à l'étape suivante.

Déployer l'application à l'aide de la version binaire

Dans ce projet de développement, nous avons choisi d'utiliser un processus appelé « versions binaires ». Ce processus crée une version en local et importe uniquement l'artefact (c'est-à-dire le fichier `.war`) plutôt que de renvoyer vers un référentiel Git et d'utiliser le processus de création de la version S2I (source-to-image) pour télécharger, créer, puis générer une image de conteneur. Le déploiement binaire accélère le processus de création de façon significative.

Pour commencer, créez le projet une nouvelle fois à l'aide du profil OpenShift Maven, ce qui génère une version binaire compatible avec Red Hat OpenShift Container Platform (il ne s'agit pas encore d'une image de conteneur, mais simplement du fichier `.war`). Nous allons créer le projet avec l'interface en ligne de commande.

Créez le projet :

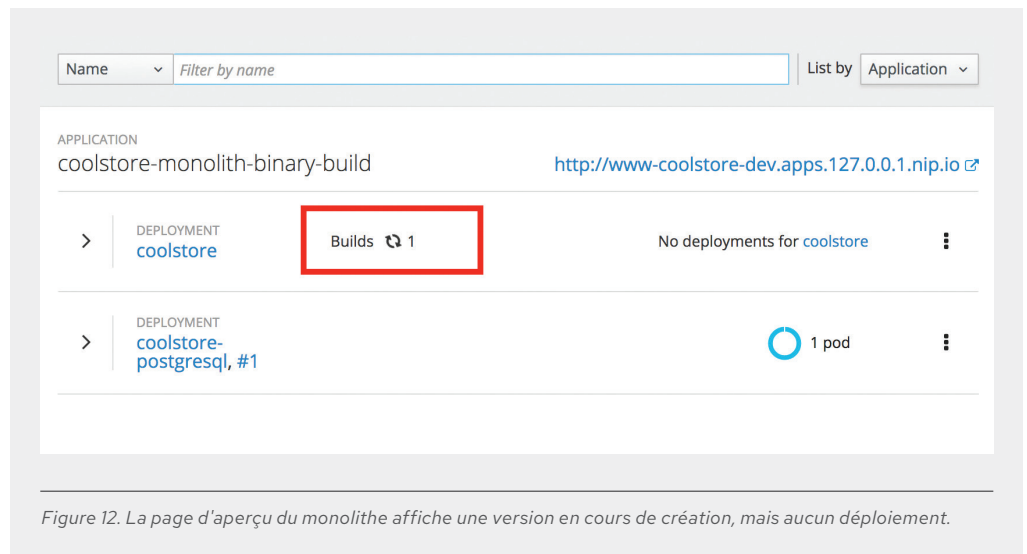
```
$ mvn clean package -Popenshift
```

Attendez que cette opération se termine et que le message `BUILD SUCCESS` s'affiche.

Enfin, lancez le processus de création qui combinera le fichier `.war` avec Red Hat JBoss EAP afin de produire une image de conteneur Linux®. Cette image sera automatiquement déployée dans le projet grâce à l'objet `DeploymentConfig` créé à partir du modèle :

```
$ oc start-build coolstore --from-file=deployments/ROOT.war
```

Consultez la console web de Red Hat OpenShift, où vous devriez retrouver l'application en cours de création. Dans ce cas, une base de données PostgreSQL a déjà été déployée.



Attendez que la création et le déploiement soient effectués :

```
$ oc rollout status -w dc/coolstore
```

Cette commande sera souvent utilisée pour attendre la fin des déploiements. Assurez-vous qu'elle confirme la réussite de l'opération lorsque vous l'utilisez. Voici le message qui devrait s'afficher en fin de processus :

```
replication controller "coolstore-1" successfully rolled out.
```

Remarque : si la commande ci-dessus génère un message « *Error from server (ServerTimeout)* », vous n'avez qu'à exécuter la commande de nouveau jusqu'à obtenir un message de confirmation.

Lorsque le processus est terminé, l'application devrait être déployée et des cercles bleus devraient s'afficher pour la base de données et le monolithe :



Testez l'application en cliquant sur le lien « Route » à l'adresse :

`http://www-coolstore-dev.$ROUTE_SUFFIX`. Le monolithe « coolstore » s'ouvre alors dans votre navigateur, mais en s'exécutant sur la plateforme Red Hat OpenShift Container Platform cette fois.

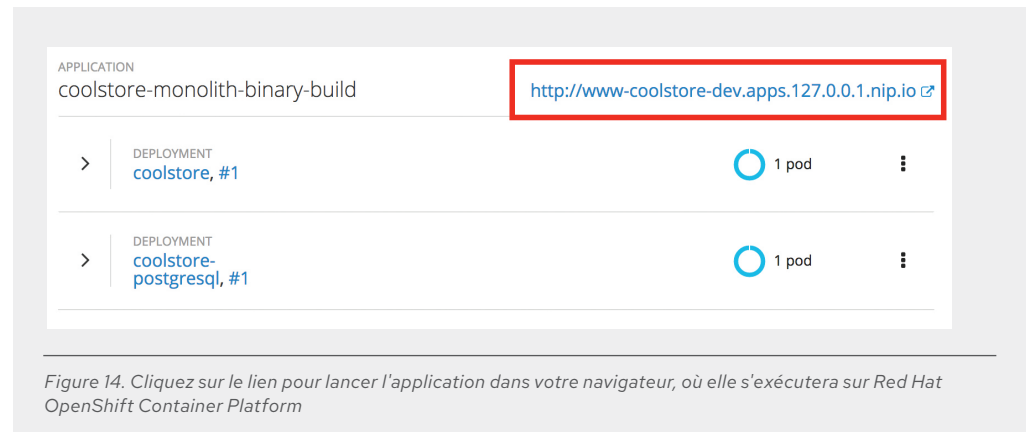
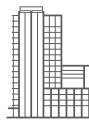


Figure 14. Cliquez sur le lien pour lancer l'application dans votre navigateur, où elle s'exécutera sur Red Hat OpenShift Container Platform

Synthèse

Ce processus vous a montré comment migrer une application Java EE existante vers le cloud avec les solutions Red Hat JBoss Enterprise Application Platform et Red Hat OpenShift Container Platform. Vous êtes maintenant prêt à moderniser l'application en décomposant progressivement le monolithe en microservices plus petits, à l'aide de techniques modernes pour garantir le bon fonctionnement de l'application dans un environnement distribué et conteneurisé.



À PROPOS DE RED HAT

Premier éditeur mondial de solutions logicielles Open Source pour les entreprises, Red Hat s'appuie sur une approche communautaire pour proposer des technologies Linux, de cloud hybride, de conteneur et Kubernetes fiables et performantes. Red Hat aide ses clients à intégrer des applications nouvelles et existantes, à développer des applications natives pour le cloud, à standardiser leur environnement sur son système d'exploitation leader sur le marché ainsi qu'à automatiser, sécuriser et gérer des environnements complexes. Red Hat propose également des services d'assistance, de formation et de certification primés qui lui ont valu le titre de conseiller de confiance auprès des entreprises du Fortune 500. Partenaire stratégique des prestataires de cloud, intégrateurs système, fournisseurs d'applications, clients et communautés Open Source, Red Hat aide les entreprises à se préparer à un avenir toujours plus numérique.



facebook.com/redhatinc
@RedHat_France
linkedin.com/company/red-hat

EUROPE, MOYEN-ORIENT
ET AFRIQUE (EMEA)
00800 7334 2835
europe@redhat.com

FRANCE
00 33 1 4191 2323
fr.redhat.com