



Red Hat JBoss Enterprise Application Platform 7.1

Introduction à JBoss EAP

À utiliser avec Red Hat JBoss Enterprise Application Platform 7.1

Red Hat JBoss Enterprise Application Platform 7.1 Introduction à JBoss EAP

À utiliser avec Red Hat JBoss Enterprise Application Platform 7.1

Notice légale

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Résumé

Ce document vous propose un aperçu global et conceptuel de Red Hat JBoss Enterprise Application Platform, ainsi que de divers thèmes sur lesquels repose JBoss EAP.

Table des matières

CHAPITRE 1. APERÇU DES CONCEPTS GÉNÉRAUX	3
1.1. JAVA	3
1.2. SERVEURS D'APPLICATIONS	3
1.3. JAVA EE 7	3
CHAPITRE 2. PRÉSENTATION DE JBOSS EAP	4
2.1. À PROPOS DE JBOSS EAP 7	4
2.2. SOUS-SYSTÈMES	5
2.3. HAUTE DISPONIBILITÉ	5
2.4. MODES D'EXPLOITATION	5
CHAPITRE 3. EXEMPLES	6
3.1. EXEMPLE SIMPLE	6
3.2. EXEMPLE ÉTENDU	6

CHAPITRE 1. APERÇU DES CONCEPTS GÉNÉRAUX

Avant de comprendre la manière dont Red Hat JBoss Enterprise Application Platform peut être configuré et déployé, vous devez vous familiariser avec quelques concepts importants.

1.1. JAVA

Java est à la fois un langage de programmation et une plateforme informatique qui intègre des concepts tels que l'orientation objets, les classes et la simultanéité. Les applications Java sont compilées au niveau du code à octets et exécutées dans une machine virtuelle Java (JVM).

1.2. SERVEURS D'APPLICATIONS

Un serveur d'applications est un logiciel qui fournit un environnement destiné à l'exécution d'applications web. La plupart des serveurs d'applications fournissent également des fonctionnalités aux applications web qui s'exécutent dans leur environnement par le biais d'API. Un serveur d'applications peut, par exemple, fournir une API en vue d'une connexion à une base de données.

1.3. JAVA EE 7

Java EE (Java Platform, Enterprise Edition) est une plateforme d'entreprise standard qui fournit à la fois une API et un environnement pour l'exécution et le développement d'applications Java. L'objectif est ici d'améliorer la productivité des développeurs en proposant de puissantes fonctionnalités d'entreprise dans des structures faciles à exploiter, qui éliminent le prototypage et réduisent la charge technique. Les structures qui composent Java EE font l'objet de tests intensifs conjoints. Java EE 7, basé sur [JSR 342](#), fait suite à Java EE 6. Le principal objectif de cette version est de simplifier davantage encore les API afin d'accéder à des services de conteneur, tout en améliorant la prise en charge des nouvelles technologies web, telles que HTML5.

Java EE 7 assure la prise en charge de plusieurs profils, ou sous-ensembles, d'API. La spécification Java EE 7 définit le profil *full* et le profil *web*.

Profil web Java EE 7

Le profil web Java EE 7 est conçu pour le développement d'applications web. Il prend en charge un sous-ensemble des API définies par les technologies web associées à Java EE 7.

Profil full Java EE 7

Le profil full Java EE 7 contient toutes les API définies par Java EE 7 (y compris l'ensemble des éléments du profil web). Lors du développement d'EJB, d'applications de messagerie et de services web (contrairement aux applications web), utilisez le profil full.

CHAPITRE 2. PRÉSENTATION DE JBOSS EAP

2.1. À PROPOS DE JBOSS EAP 7

Red Hat JBoss Enterprise Application Platform 7.1 (JBoss EAP 7) est une implémentation certifiée des spécifications des profils full et web de Java Enterprise Edition 7 (Java EE 7).

Les versions majeures de JBoss EAP sont issues du projet communautaire [WildFly](#) en certains points, lorsque ce projet a atteint le niveau d'achèvement souhaité d'une fonctionnalité. S'ouvre alors une période prolongée de tests et de productisation, au cours de laquelle JBoss EAP est stabilisé, certifié et amélioré en vue de la production. Pendant la durée de vie d'une version majeure de JBoss EAP, certaines fonctionnalités peuvent faire l'objet d'une sélection minutieuses et être rétroportées, depuis le projet communautaire, dans une série de versions mineures au sein de la même famille de versions majeures. JBoss EAP 7.1, par exemple, est issu d'une branche opérationnelle de WildFly 11.

JBoss EAP fournit des options préconfigurées pour des fonctions telles que le clustering à haute disponibilité, la messagerie et la mise en cache distribuée. Il permet également aux utilisateurs d'écrire, de déployer et d'exécuter des applications à l'aide des différents services et API fournis.

JBoss EAP comprend une structure modulaire qui permet l'activation des services uniquement lorsque cela s'avère nécessaire, d'où un démarrage plus rapide. La console et l'interface de ligne de commande (CLI) web rendent inutile la modification des fichiers de configuration XML. Elles offrent, en outre, la possibilité d'écrire le script des tâches et d'automatiser ces dernières. JBoss EAP comprend également des API et des infrastructures permettant un développement rapide d'applications Java EE sécurisées et évolutives.

Tableau 2.1. Fonctions de JBoss EAP

Fonction	Description
Conformité avec Java EE 7	Profils full et web Java Enterprise Edition 7 certifiés.
Domaine géré	Gestion centralisée de plusieurs instances de serveur et hôtes physiques, alors qu'un serveur autonome n'autorise qu'une seule instance de serveur. Gestion de la configuration, du déploiement, des liaisons de sockets, des modules, des extensions et des propriétés système par groupe de serveurs. Gestion centralisée et simplifiée de la sécurité de l'application (y compris les domaines de sécurité).
Interface de ligne de commande et console de gestion	Nouvelles interfaces de gestion de serveur autonome ou de domaine. L'interface de ligne de commande de gestion comprend également un mode d'exécution par lots permettant d'écrire le script des tâches de gestion et d'automatiser ces dernières. Il est déconseillé de modifier directement les fichiers de configuration XML de JBoss EAP.

Fonction	Description
Disposition de répertoire simplifiée	Le répertoire modules contient tous les modules du serveur d'applications. Les répertoires domain et standalone contiennent, respectivement, les artefacts et les fichiers de configuration relatifs aux déploiements de domaine et autonomes.
Mécanisme de chargement de classes modulaire	Les modules sont chargés et déchargés à la demande. Outre une augmentation des performances, ce mécanisme présente des avantages sur le plan de la sécurité, et réduit les temps de démarrage et de redémarrage.
Gestion simplifiée des sources de données	Les pilotes de base de données sont déployés comme d'autres services. En outre, les sources de données sont créées et gérées à l'aide de l'interface de ligne de commande et de la console de gestion.

2.2. SOUS-SYSTÈMES

Bon nombre des API et des fonctionnalités exposées aux applications déployées sur JBoss EAP sont organisées dans des sous-systèmes. Ceux-ci peuvent être configurés par les administrateurs de manière à présenter un comportement différent suivant l'objectif de l'application. Par exemple, si une application nécessite une base de données, une source de données peut-être configurée dans le sous-système **datasources**. L'application pourra alors y accéder après avoir été déployée sur ce domaine ou serveur JBoss EAP.

2.3. HAUTE DISPONIBILITÉ

Dans JBoss EAP, la haute disponibilité (HA) désigne le fait que plusieurs instances JBoss EAP collaborent pour fournir des applications plus résistantes aux variations de trafic, à la charge du serveur et aux pannes de serveur. HA intègre des concepts tels que l'évolutivité, l'équilibrage de charge et la tolérance aux pannes.

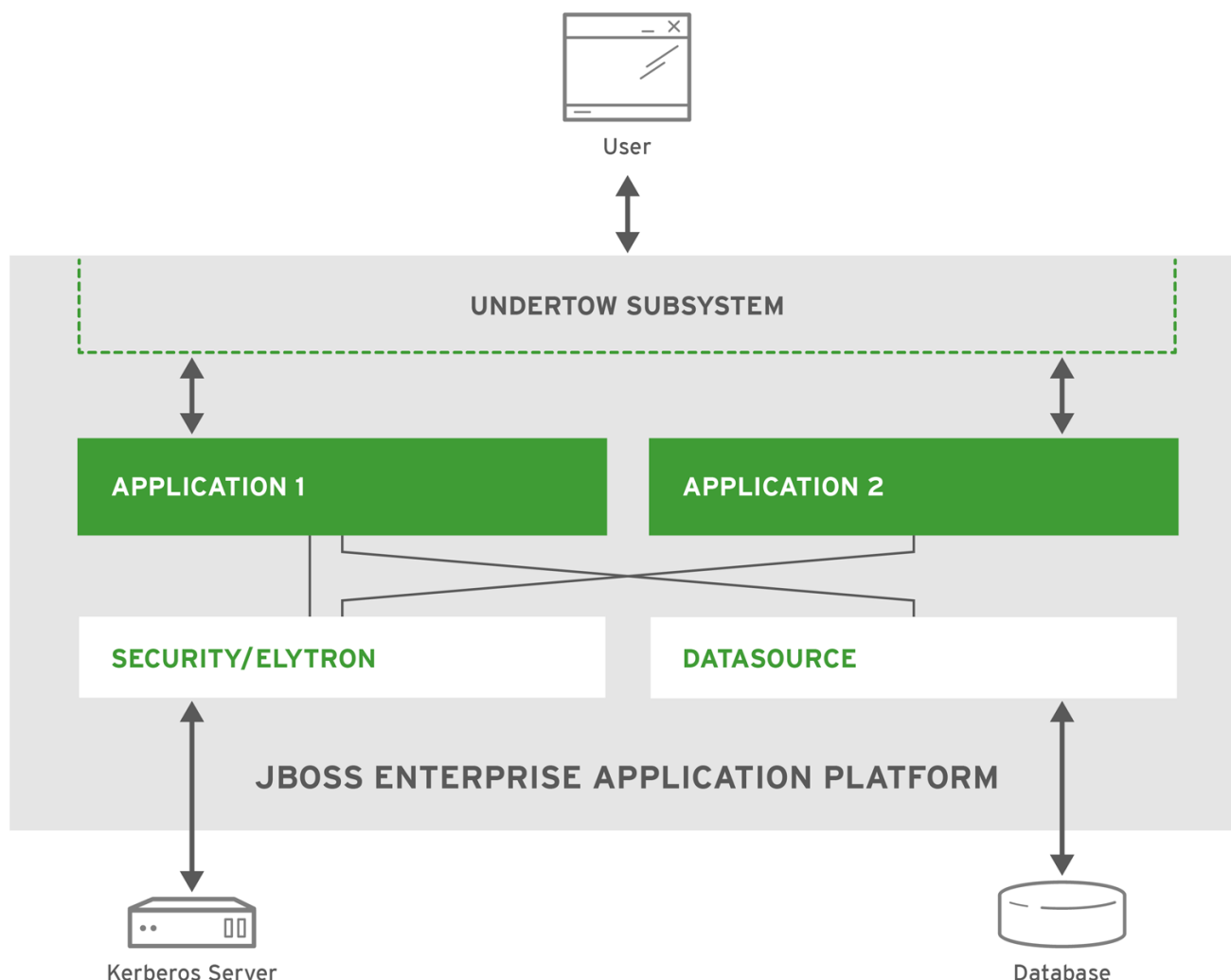
2.4. MODES D'EXPLOITATION

Outre la fourniture de fonctionnalités et d'API à ses applications, JBoss EAP offre de puissantes fonctions de gestion. Ces dernières varient selon le mode d'exploitation utilisé pour démarrer JBoss EAP. JBoss EAP propose un mode d'exploitation *serveur autonome* pour gérer des instances discrètes et un mode *domaine géré* pour gérer des groupes d'instances depuis un seul point de contrôle.

CHAPITRE 3. EXEMPLES

Les exemples ci-dessous illustrent le fonctionnement de JBoss EAP, ainsi que son intégration dans différents environnements.

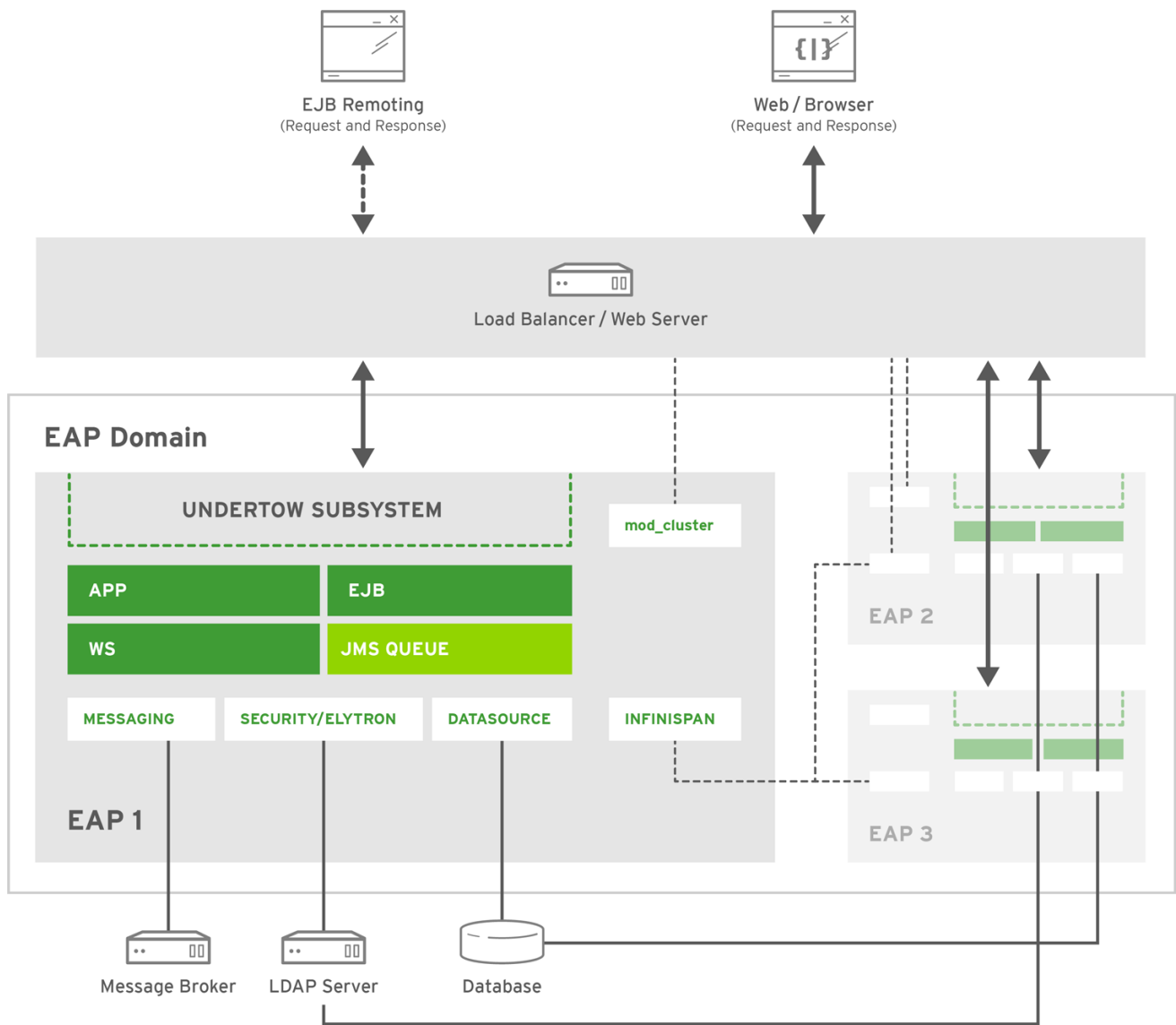
3.1. EXEMPLE SIMPLE



JBoss_430110_1216

Cet exemple illustre une configuration simple de JBoss EAP. Deux applications sont déployées sur cette instance de JBoss EAP. Cette dernière est également configurée de manière à se connecter à une base de données à l'aide du sous-système **datasources** et à un serveur Kerberos pouvant utiliser le sous-système **security** hérité ou le sous-système **elytron**. Ces connexions sont présentées aux applications déployées. L'instance de JBoss EAP traite les requêtes par le biais du sous-système **undertow** et les redirige vers l'application appropriée. L'application utilise les API exposées par JBoss EAP pour se connecter à la base de données et au serveur Kerberos, et exécuter la logique métier mise en œuvre. Une fois l'opération effectuée, les applications renvoient une réponse au demandeur par l'intermédiaire du sous-système **undertow**.

3.2. EXEMPLE ÉTENDU



JBoss_430110_1216

Cet exemple illustre une configuration plus complexe dans laquelle trois instances JBoss EAP sont organisées dans un domaine géré avec un équilibreur de charge ou un serveur web. Ces trois instances sont également configurées pour prendre en charge la haute disponibilité via l'équilibrage de charge à l'aide de `mod_cluster` et la réplication de session à l'aide d'`Infinispan`. Une application web, un service web et EJB sont, en outre, déployés sur ces instances. Sur une instance JBoss EAP, une file d'attente JMS est configurée via le sous-système **messaging-activemq**. Les trois instances JBoss EAP disposent de connexions à une base de données via la source de données. Elles sont également connectées au serveur LDAP à l'aide du sous-système **security** hérité ou du sous-système **elytron**. Une seule instance JBoss EAP est configurée pour se connecter à un broker de messages externe via le sous-système **messaging-activemq**. Ces connexions configurées sont exposées aux applications, services web, EJB et files d'attente JMS déployés sur cette instance.

Toutes les requêtes entrantes destinées à l'application, au service web ou à EJB sont d'abord reçues par l'équilibreur de charge ou le serveur web. En fonction de l'algorithme d'équilibrage de charge configuré et des informations fournies par chaque instance de JBoss EAP, le serveur web ou l'équilibreur de charge redirige la requête vers l'instance appropriée de JBoss EAP. Cette dernière traite les requêtes via le sous-système **undertow** et les redirige vers l'application appropriée. Les applications utilisent les API exposées par JBoss EAP pour se connecter à la base de données et au serveur Kerberos, et exécutent la logique métier mise en œuvre. Une fois l'opération effectuée, les applications renvoient une réponse au demandeur par l'intermédiaire du sous-système **undertow**. Toutes les informations non persistantes (des informations de session, par exemple) sont propagées parmi les instances de JBoss EAP par le biais du sous-système **infinispan**.

Revised on 2018-01-11 05:29:28 EST