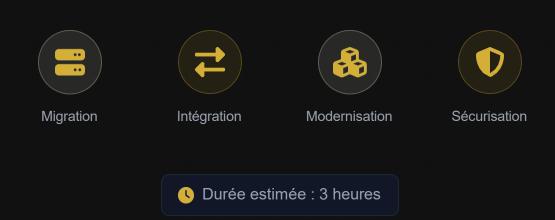
MODULE 8

Migration d'applications vers OpenShift

Stratégies et approches d'entreprise



Introduction & Enjeux de la migration

Pourquoi migrer vers OpenShift?

La migration vers OpenShift représente une transition stratégique pour les entreprises souhaitant moderniser leur infrastructure IT. Cette démarche s'inscrit dans une vision à long terme d'agilité, de sécurité et d'optimisation des ressources.

Bénéfices d'affaires

- Accélération du time-to-market et de l'innovation
- Conformité renforcée et sécurité améliorée
- Réduction des coûts opérationnels à long terme
- Flexibilité et mise à l'échelle simplifiées

Vision stratégique



La migration vers OpenShift n'est pas simplement un changement technique, mais une transformation de la culture organisationnelle, permettant d'adopter pleinement les pratiques DevOps et de favoriser l'innovation continue.

Enjeux et considérations clés



Évaluation et planification

Une analyse approfondie de l'existant est indispensable pour identifier les applications candidates à la migration, leurs dépendances et les défis potentiels.



Stratégies de migration

Différentes approches existent : Lift & Shift, Re-platforming ou Refactoring complet. Le choix dépend des objectifs business, des contraintes techniques et du ROI attendu.



Compétences et culture

La transition vers OpenShift nécessite une montée en compétence des équipes et souvent un changement culturel vers le DevOps et les pratiques cloud-native.



Persistance et données

La gestion des données persistantes, l'intégration avec les services externes et la configuration des secrets constituent des défis techniques majeurs.

Facteurs de risque à anticiper

- Dépendances legacy complexes
- Courbe d'apprentissage technique
- Résistance au changement
- Coûts initiaux de transition



Stratégies de migration : panorama







Analyse comparative

Critère	Lift & Shift	Re-platforming	Modernisation
Délai	Court	Moyen	× Long
Coût initial	✓ Faible	Modéré	× Élevé
ROI long terme	× Limité	• Bon	Excellent
Compétences	De base	Intermédiaires	× Avancées
▼ Critères de choix professionnels Valeur stratégique App. critiques : modernisation Besoins d'évolutivité Forte croissance : modernisation		Horizon de vie Courte durée : Lift & Shift Dette technique Dette élevée : modernisation	

Évaluation des applications existantes

Méthodologie d'audit

Une analyse méthodique des applications existantes constitue la fondation d'une migration réussie vers OpenShift, permettant d'identifier opportunités et risques.

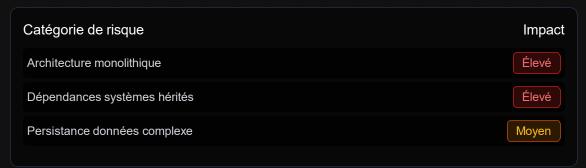
Check-list essentielle

- P Architecture applicative
 Composants, services et interdépendances.
- Cartographie des dépendances Librairies, frameworks, services externes.
- Persistance et stockage Besoins en données persistantes.

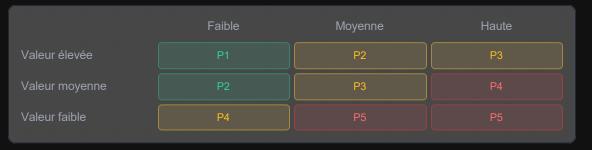
Priorisation stratégique

Établissez une matrice de priorisation basée sur la valeur métier et la complexité technique pour optimiser l'ordre de migration.

Grille d'analyse des risques



Matrice d'évaluation simplifiée



Une analyse des risques bien documentée permet de mettre en place des stratégies d'atténuation adaptées et d'optimiser l'allocation des ressources.



Intégration de services externes

Sécurisation des connections externes

L'intégration sécurisée de services externes constitue un enjeu critique lors de la migration vers OpenShift, particulièrement pour les applications d'entreprise.

Types de services à intégrer



Systèmes de fichiers
Accès aux stockages réseau via PersistentVolumes.
Criticité:

Authentification SSO

Fédération d'identité avec systèmes existants (LDAP, AD).

Criticité:

Meilleures pratiques d'intégration

- Sécurisation des accès externes
- Utiliser Secrets pour les informations d'identification
- Configurer le TLS mutuel pour l'authentification
- Utiliser Service Mesh pour sécuriser le trafic
- 몲 Connectivité fiable
- Mécanismes de retry et circuit breaker
- Health checks adaptés

- Performance
- Connection pooling
- Caching approprié

Configuration BDD: Exemple

apiVersion: v1 kind: Secret metadata:

name: external-db-credentials type:

Opaque

data:

username: dXNlcm5hbWU= # Base64 encoded password: cGFzc3dvcmQ= # Base64 encoded

Migration d'applications middleware (JBoss/Wildfly, etc.)

Spécificités de la migration middleware

La migration JBoss/Wildfly vers OpenShift nécessite une approche adaptée aux spécificités du middleware, ses dépendances et sa configuration.

Méthodologie spécialisée

Analyse du déploiement existant

Cartographie de l'environnement JBoss, identification des configurations, datasources et modules.

Adaptation de la configuration

Transformation des fichiers XML en ConfigMaps ou propriétés d'environnement.

3 Conteneurisation de l'application

Images JBoss optimisées pour OpenShift et adaptation des déploiements.

(4) Test et validation incrémentale

Vérification progressive des fonctionnalités, transactions et sécurité.

Outils et points de vigilance

Migration Toolkit for Applications

Analyse du code pour identifier les problèmes de compatibilité spécifiques à JBoss.

JBoss EAP Operator

Déploiement simplifié d'applications JBoss sur OpenShift avec configurations adaptées.

A Points de vigilance spécifiques

Datasources & JNDI
 Adaptation des configurations pour conteneurs

Transactions XA Intégrité dans environnement dynamique Clustering JGroupsConfiguration pour Kubernetes

Sécurité
 Migration vers mécanismes OpenShift

Conseil d'expert

•

Privilégiez une migration progressive, en commençant par les applications les moins critiques et en utilisant les images officielles Red Hat pour JBoss EAP sur OpenShift.



Modernisation applicative dans OpenShift

Pourquoi moderniser les applications?

La modernisation transforme l'architecture pour exploiter pleinement les capacités cloud-native d'OpenShift, maximisant le ROI et facilitant l'évolution future.

Décomposition en microservices

- 1 Identification des domaines métier

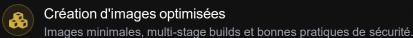
 Analyser pour identifier les limites contextuelles et responsabilités distinctes.
- Extraction progressive
 Extraire d'abord les services périphériques puis progresser vers le cœur.
- Refonte des interfaces

 Redéfinir les API pour une communication efficace entre microservices.

Approche pragmatique

Adoptez une modernisation incrémentale en ciblant d'abord les composants à forte valeur ajoutée ou techniquement obsolètes.

Stratégies de conteneurisation



Configuration externalisée
Séparer code et configuration via ConfigMaps et Secrets d'OpenShift.

Intégration des health checks
Readiness et liveness probes pour une gestion optimale du cycle de vie.

Refactoring cloud-native

Principe			Bénéfice				
Stateless			Scalabilité horizontale				
Résilience			Tolérance aux pannes				
Observabilité			Monitoring simplifié				
Automatisation			CI/CD et déploiement				

★ Apports pour la maintenance et l'évolutivité
 ✓ Mises à jour indépendantes
 ✓ Déploiements fréquents
 ✓ Isolation des défaillances
 ✓ Scaling adapté à l'usage

Best practices et pièges à éviter

Facteurs clés de succès

La réussite d'une migration vers OpenShift repose sur une méthodologie éprouvée et une anticipation des défis techniques et organisationnels.

Approche méthodique

- Établir une feuille de route progressive et réaliste
- Commencer par les applications les moins critiques
- ✓ Mettre en place des KPIs clairs et mesurables
- Prévoir des périodes de cohabitation entre anciens et nouveaux systèmes

"La migration la plus réussie est celle qui a été minutieusement préparée. Documentez vos architectures actuelles, cartographiez vos dépendances et utilisez une approche incrémentale."

- Retour d'expérience d'un Architecte Cloud

Pièges à éviter



Sous-estimer la complexité

Ne pas anticiper les spécificités d'OpenShift (sécurité, réseau, stockage) peut considérablement ralentir le projet.



Négliger la formation des équipes

Un manque de compétences internes sur Kubernetes et OpenShift génère une dépendance excessive aux consultants externes.



Migrer sans adapter l'application

Simple "lift & shift" sans optimisation pour l'environnement conteneurisé : performances dégradées et coûts plus élevés.



Ignorer les questions de persistance

La gestion incorrecte des volumes persistants et des données d'état peut conduire à des pertes de données.

Checklist de migration

- Cartographie des applications et dépendances
- ✓ Plan de formation des équipes DevOps
- Stratégie de gestion des secrets et configurations
- Plan de tests et validation pré-production
- Stratégie de rollback et plan de contingence



Cas d'étude : migration d'une application réelle

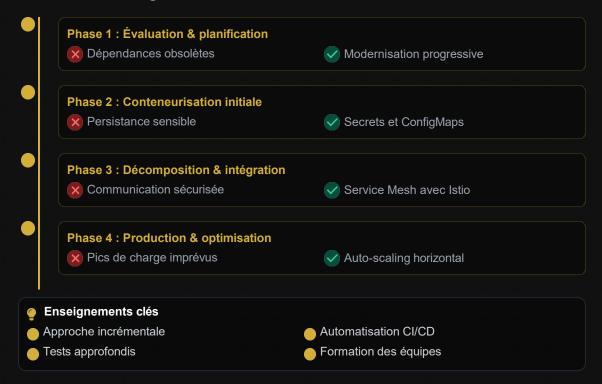
FinTech Solutions — Application métier critique

Migration d'un système de gestion de transactions financières initialement déployé sur un environnement legacy vers OpenShift. Application Java/Spring traitant 10 000+ transactions quotidiennes.

Situation initiale & objectifs



Parcours de migration





Travaux pratiques : atelier migration

Objectifs de l'atelier

Réaliser une migration complète d'application métier vers OpenShift. Développer une méthodologie structurée et identifier les défis courants de migration en environnement d'entreprise.

Cas d'étude : Application ERP légère

Application JBoss/Wildfly

PostgreSQL externalisée

Services externes (SSO, API)

Tâches cron périodiques

Outils fournis



X Application source

Documentation technique

Cluster OpenShift

✓ Templates de base

Processus de migration guidé

Étape 1 : Analyse & planification

Évaluation des composants et création d'un plan de migration.

Livrable: Document d'analyse d'architecture avec dépendances et plan d'action.

Étape 2 : Préparation de l'environnement

Configuration du projet, secrets et connexions externes.

Exercice: Création des ressources nécessaires et validation des accès.

Étape 3 : Migration

Conteneurisation et configuration des déploiements OpenShift.

Focus: Adaptation JBoss/Wildfly et gestion des volumes persistants.

Étape 4 : Test et validation

Vérification complète des fonctionnalités post-migration.

Critères: Fonctionnalités identiques, intégrations opérationnelles, performances.

- Points d'apprentissage clés
- Diagnostic d'erreurs
- Techniques de rollback

- Connexions externes
- Documentation migration

Synthèse professionnelle & perspectives

Points clés à retenir

La migration vers OpenShift requiert planification, évaluation approfondie et approche méthodique pour garantir le succès de votre transformation.



Analysez vos applications pour identifier dépendances, configurations et middlewares avant migration.



Sécurisez les connexions aux services externes avec les mécanismes OpenShift (Secrets, ConfigMaps)



Stratégies adaptées

Choisissez entre Lift & Shift, Re-platforming ou Refactoring selon vos objectifs stratégiques.

Modernisation progressive

Décomposez progressivement les applications monolithiques en microservices.

Recommandation finale

Adoptez une approche incrémentale et priorisez les applications selon leur valeur business et complexité technique. Un plan de migration progressif augmente vos chances de succès.

Questions fréquentes

Quel est le temps moyen d'une migration?

2 à 6 semaines par application complexe (analyse, migration, tests).

Comment gérer les états persistants ?

PersistentVolumes et opérateurs DB pour garantir la persistance.

Ressources pour aller plus loin



Documentation officielle

Guide de migration - Red Hat Developer Portal



Outils d'analyse

Migration Toolkit for Applications (MTA)



Communauté et support

OpenShift Commons - Groupes d'intérêt sur la migration