DEVOPS-020 — OpenShift : développer et déployer des applications conteneurisées — Programme de formation

Informations générales

Code	DEVOPS-020
Titre	OpenShift : développer et déployer des applications conteneurisées
Durée	21 heures — 3 jours
Format	Intra-entreprise — Présentiel (distanciel possible sur demande)
Public	Développeurs, architectes logiciels, ingénieurs DevOps, intégrateurs d'applications
Prérequis	Bases des conteneurs (image, Dockerfile, registre) et de Kubernetes (pod, Deployment, Service), ligne de commande et Git (DEVOPS-001 et DEVOPS-012 recommandés)
Niveau	Intermédiaire
Financement	OPCO — Plan de Développement des Compétences (PDC)
Organisme	Telemach Learning — NDA 11 92 27843 92
Formateur	Alexandre BOIGUES

Objectifs pédagogiques

À l'issue de cette formation, le participant est capable de :

- Déployer une application sur un cluster OpenShift, l'exposer, la configurer et l'administrer au quotidien
- Concevoir des images de conteneurs qui s'exécutent sans root, les publier dans un registre et en régler les droits d'accès
- Faire construire ses applications par OpenShift : BuildConfig, déclencheurs, contrôles avant publication, Source-to-Image personnalisé
- Décrire une application à plusieurs conteneurs dans un modèle paramétrable et la réinstaller en une commande
- Surveiller la santé d'une application et choisir une stratégie de déploiement sans interruption de service
- Migrer vers OpenShift une application existante et la relier à des systèmes restés hors du cluster

© ALEXANDRE BOIGUES 2026 TOUS DROITS RÉSERVÉS — NDA 11 92 27843 92 — v1.0

Programme détaillé

Les ateliers s'appuient sur une même application, Télémarché (interface web, API et base de données PostgreSQL), déployée dès le premier module puis construite, publiée et mise à jour par la plateforme, module après module. Le dernier module migre une seconde application, écrite en Java et déployée sur un serveur WildFly.

Jour 1 — Déployer sur OpenShift et concevoir des images qui s'y exécutent

Module 1 — Déployer et exploiter une application sur OpenShift (3h30)

- Ce qu'OpenShift ajoute à Kubernetes : projets, Routes, ImageStreams, builds, console web
- Le client `oc` et la console : se connecter, créer un projet, lire l'état d'une application
- Déployer une image existante avec `oc new-app` ; ce que la commande crée et ce qu'elle laisse à faire
- Exposer en HTTPS : Service, Route, terminaisons TLS
- Configurer et mettre à l'échelle : variables, ConfigMaps, Secrets, volumes, ressources
- Deployment et DeploymentConfig : reconnaître l'objet historique et le convertir
- Diagnostiquer : journaux, événements, `oc debug`, pannes courantes
- Atelier Télémarché : déployer la vitrine et l'API, les exposer, les configurer, mettre à jour puis revenir en arrière

Module 2 — Concevoir et publier des images prêtes pour OpenShift (3h30)

- Principes de conception appliqués aux conteneurs : simplicité, absence de doublon, rien de superflu, séparation des responsabilités
- Du monolithe aux microservices : ce que l'on gagne, ce que l'on paie, comment découper progressivement
- La contrainte de sécurité d'OpenShift : utilisateur imposé, groupe 0, aucune élévation de privilèges
- Écrire un Dockerfile avancé : images de base UBI, construction en plusieurs étapes, cache, arguments, labels, arrêt propre
- Tester sur son poste une image comme elle sera exécutée sur OpenShift
- Registre d'entreprise et registre intégré ; ImageStreams, étiquettes et promotion d'une image
- Droits d'accès aux images : rôles, partage entre projets, identifiants d'un registre externe
- Atelier Télémarché : corriger une image qui échoue sur OpenShift, la publier, la promouvoir et en ouvrir l'accès à un second projet

© ALEXANDRE BOIGUES 2026 TOUS DROITS RÉSERVÉS — NDA 11 92 27843 92 — v1.0

Programme détaillé (suite)

Jour 2 — Faire construire et décrire ses applications par la plateforme

Module 3 — Construire dans le cluster : builds, déclencheurs et contrôles (3h30)

- Le processus de build d'OpenShift : BuildConfig, Build, pod de build, publication
- Sources (Git, binaire, Dockerfile) et stratégies (Docker, Source, Custom)
- Lancer, suivre et diagnostiquer un build ; build depuis le poste de travail
- Déclencheurs : webhook de la forge, changement de l'image de base, lien avec le déploiement
- Variables, arguments et secrets de build ; ressources et historique
- Hooks post-commit : contrôler l'image avant qu'elle n'atteigne le registre
- Situer le BuildConfig face aux pipelines (Tekton, chaînes d'intégration continue existantes)
- Atelier Télémarché : un push sur le dépôt reconstruit et redéploie l'API ; un hook bloque une version défectueuse

Module 4 — Source-to-Image et modèles d'application (3h30)

- Source-to-Image : du code source à l'image sans Dockerfile ; scripts `assemble` et `run`
- Personnaliser à trois niveaux : variables, scripts dans le dépôt, image de construction sur mesure
- Builds incrémentaux ; séparer l'image de construction de l'image d'exécution
- Les modèles d'OpenShift : objets, paramètres, valeurs générées
- Analyser un modèle fourni, puis en créer un à partir d'une application qui fonctionne
- Modèle à plusieurs conteneurs : secret partagé, liens entre composants, déclencheurs
- Limites des modèles et relais par Helm
- Atelier Télémarché : image de construction personnalisée, puis installation complète (web, API, base) à partir d'un modèle

© ALEXANDRE BOIGUES 2026 TOUS DROITS RÉSERVÉS — NDA 11 92 27843 92 — v1.0

Programme détaillé (fin)

Jour 3 — Livrer sans interruption et migrer l'existant

Module 5 — Santé et stratégies de déploiement (3h30)

- Sondes de démarrage, de disponibilité et de vie : rôle de chacune, réglages, pièges
- Concevoir les points de contrôle côté application ; arrêt propre
- Superviser : journaux, événements, métriques et alertes du projet
- Déploiement progressif et par recréation : conditions de réussite, pilotage, retour arrière
- Bleu-vert et canari avec les poids d'une Route
- Faire évoluer le schéma d'une base sans coupure
- Choisir la stratégie que sa supervision permet de piloter
- Atelier Télémarché : mise à jour sous trafic avec mesure des erreurs, sondes, bascule bleu-vert puis canari

Module 6 — Migrer une application existante vers OpenShift (3h30)

- Démarche de migration : évaluer, conteneuriser, externaliser, déployer, basculer
- Intégrer des services externes : Service sans sélecteur, ExternalName, identifiants et certificats
- Migrer une application Java déployée sur JBoss ou WildFly : ce qui change dans un conteneur
- Construire l'image : Source-to-Image, Dockerfile ou graphique Helm
- Externaliser la configuration ; santé, journaux et mémoire de la JVM
- Sessions et fonctionnement à plusieurs réplicas ; pièges classiques d'une migration
- Atelier final : relier l'application à une base située hors du projet, puis migrer une application Java de WildFly vers OpenShift

© ALEXANDRE BOIGUES 2026 TOUS DROITS RÉSERVÉS — NDA 11 92 27843 92 — v1.0

Méthodes pédagogiques

- Alternance d'apports courts et d'ateliers pratiques (environ 60 % du temps en pratique)
- Un projet OpenShift par participant ; schémas et animations pas à pas pour chaque mécanisme
- Travaux pratiques progressifs : fichiers de départ à compléter, solution de référence, script de vérification du résultat attendu
- Cas réels tirés de la pratique du formateur : images refusées par la plateforme, déploiements interrompus par des sondes mal réglées
- Dépôt de travaux pratiques remis aux participants pour continuer à pratiquer après la formation

Modalités d'évaluation

- Questionnaire de positionnement avant la formation
- Script de vérification à l'issue de chaque atelier
- Quiz de fin de formation
- Mise en situation finale : migrer une application existante et la déployer sans interruption
- Évaluation des acquis par le formateur en continu

© ALEXANDRE BOIGUES 2026 TOUS DROITS RÉSERVÉS — NDA 11 92 27843 92 — v1.0

Parcours

Cette formation fait suite à DEVOPS-001 — Conteneurisation avec Docker et à DEVOPS-012 — Kubernetes : déployer et exploiter un cluster, qui en couvrent les prérequis. Elle se prolonge avec DEVOPS-040 — GitLab CI/CD : industrialiser le cycle de vie logiciel, pour les équipes qui pilotent leurs déploiements depuis une chaîne d'intégration continue.

OpenShift est une marque de Red Hat. Cette formation est indépendante de l'éditeur et ne prépare pas à une certification.

Accessibilité

Pour toute situation de handicap, contacter Telemach Learning avant l'inscription afin d'étudier les adaptations possibles.

Financement

Formation éligible au Plan de Développement des Compétences (PDC).
OPCOs ciblés : OPCO 2i, Atlas, Constructys, Afdas, AKTO.

Telemach Learning — 24 Rue Chanzy, 92600 Asnières-sur-Seine
SIRET 522 540 061 00010 — NDA 11 92 27843 92
contact@telemach-learning.fr — www.telemach-learning.fr

© ALEXANDRE BOIGUES 2026 TOUS DROITS RÉSERVÉS — NDA 11 92 27843 92 — v1.0