

# DEVOPS CODE2UTF('8211',0) LA DÉMARCHE POUR DÉLIVRER EN CONTINU

**CODE STAGE : D0001**

## OBJECTIFS

Pouvoir identifier les problèmes liés à la communication inter-équipes  
Savoir mettre en place des objectifs dcode2utf('8217',0)équipe communs  
Être en mesure de former les développeurs aux opérations de production  
Comprendre comment standardiser les livrables en accord avec les équipes  
Savoir automatiser le provisioning des environnements  
Maîtriser lcode2utf('8217',0)automatisation des déploiements applicatifs  
Disposer des compétences nécessaires à la mise en oeuvre dcode2utf('8217',0)un monitoring applicatif

## DURÉE

2 jours

## PUBLIC

Développeurs, architectes, administrateurs systèmes

## PRÉ-REQUIS

Connaissance des services IT

## PROGRAMME

### LA PROBLÉMATIQUE DEVOPS

code2utf('8211',0) Les différents métiers, du dev à la prod  
code2utf('8211',0) Les divergences entre les équipes de développement et les équipes opérationnelles  
code2utf('8211',0) Les sources de conflits entre équipes (déploiement, technologies mises en oeuvrecode2utf('8230',0))

### INDUSTRIALISATION DES DÉPLOIEMENTS

code2utf('8211',0) Standardisation des livrables  
code2utf('8211',0) Mise en place dcode2utf('8217',0)une usine logicielle et de lcode2utf('8217',0)intégration continue

code2utf('8211',0) Standardisation du provisioning

## LE CLOUD ET LA VIRTUALISATION

code2utf('8211',0) Les apports des infrastructures Cloud, PaaS et IaaS

code2utf('8211',0) Les apports de la virtualisation hardware

code2utf('8211',0) Les nouveaux outils de virtualisation

## LA VIRTUALISATION DES ENVIRONNEMENTS

code2utf('8211',0) Présentation des différents types de virtualisation

code2utf('8211',0) Mise en oeuvre de Vagrant pour faciliter la gestion des environnements de développement

code2utf('8211',0) Mise en oeuvre de Docker pour faciliter la gestion des socles

## AUTOMATISATION DU PROVISIONING DES ENVIRONNEMENTS

code2utf('8211',0) Définir un processus commun et répétable

code2utf('8211',0) Présentation des alternatives (Dockerfile, Puppet, Chef, Ansible, Salt)

code2utf('8211',0) Mise en oeuvre via Docker

code2utf('8211',0) Mise en oeuvre via Ansible

## AUTOMATISATION DES DÉPLOIEMENTS APPLICATIFS

code2utf('8211',0) Présentation des options (Capistrano, Fabric, Ansible, Salt)

code2utf('8211',0) Mise en oeuvre via Ansible

code2utf('8211',0) Gérer les évolutions de base de données avec Liquibase

## MONITORING APPLICATIF

code2utf('8211',0) Centralisation des logs applicatifs (ElasticSearch code2utf('8211',0) Logstash

code2utf('8211',0) Kibana)

code2utf('8211',0) Définition du plan de monitoring

## COLLABORATION ENTRE LES ÉQUIPES

code2utf('8211',0) Les types d'organisations possibles

code2utf('8211',0) Prise en compte des user stories de production

code2utf('8211',0) Organisation de cérémonies communes

code2utf('8211',0) Coopération sur les choix techniques

code2utf('8211',0) Outils de communication issus de la démarche ChatOps (Hubot, Lita code2utf('8230',0))

