

## Déroulé de l'action

### Modalités

Session Inter-ou Intra  
En présentiel/Classe Virtuelle

### Horaires

9H00-12H00 /13H30-17H00

### Méthode pédagogique

Alternance exposés  
théoriques  
et exercices pratiques (80%  
de pratique)

### Suivi et assistance

Support de cours adapté  
Assistance téléphonique  
gratuite et illimitée

### Validation des acquis

Evaluation formative au  
travers de nombreux  
exercices tout au long de la  
formation permettant de  
valider les compétences  
acquises  
Attestation de stage  
Emergement quotidien d'une  
feuille de présence

### Accessibilité aux personnes handicapées

Pour tout besoin  
d'accessibilité, contactez  
Véronique Constantin,  
référente handicap :  
v.constantin@cit.fr

# DOCKER - créer et administrer des conteneurs virtuels d'applications

**Objectif :** Acquérir les compétences nécessaires pour construire et gérer des conteneurs d'applications avec la solution Docker et les produits de son écosystème.

À l'issue de la formation, le stagiaire sera capable de :

- Comprendre les principes et le positionnement de Docker et des conteneurs
- Savoir installer Docker dans plusieurs contextes systèmes
- Savoir mettre en oeuvre la solution ainsi que les produits de l'écosystème pour déployer rapidement des applications à l'aide de conteneurs
- Être capable de manipuler l'interface en ligne de commande de Docker pour créer des conteneurs

**Prérequis :** Avoir les connaissances de base de l'environnement Unix/Linux

- 3 jours -

## Introduction

- Présentation Docker, principe, fonctionnalités
- Différences machines virtuelles et conteneurs
- Cas d'utilisations : packaging d'applications, déploiement rapides, coexistence de plusieurs versions d'une application sur un même serveur

## Architecture

- Les conteneurs lxc.
- Présentation de lxc : Linux containers, historique, principe de fonctionnement
- Les Cgroups
- L'isolation de ressources, création d'un environnement utilisateur
- Positionnement par rapport aux autres solutions de virtualisation
- Les systèmes de fichiers superposés
- Présentation de aufs
- Apports de Docker : Docker Engine pour créer et gérer des conteneurs Dockers
- Plates-formes supportées
- L'écosystème Docker : Docker Machine, Docker Compose, Kitematic, Docker Swarm, Docker Registry

## Gestion des conteneurs et des images

- Analyse plus approfondie
- Gestion des conteneurs : création, modification et affichage, sauvegarde de l'état
- Utilisation d'un conteneur comme modèle
- Création d'images personnalisées : principes et instructions
- Recommandations et bonnes pratiques
- Publication d'images

## Volumes de données

- Étude des volumes de données lors de la création d'un container
- Ajout de volumes de données, audit des volumes
- Sauvegarde, migration, restauration de volumes
- Création de conteneurs liés aux volumes de données

## Déroulé de l'action

### Modalités

Session Inter-ou Intra  
En présentiel/Classe Virtuelle

### Horaires

9H00-12H00 /13H30-17H00

### Méthode pédagogique

Alternance exposés  
théoriques  
et exercices pratiques (80%  
de pratique)

### Suivi et assistance

Support de cours adapté  
Assistance téléphonique  
gratuite et illimitée

### Validation des acquis

Evaluation formative au  
travers de nombreux  
exercices tout au long de la  
formation permettant de  
valider les compétences  
acquises  
Attestation de stage  
Emargement quotidien d'une  
feuille de présence

### Accessibilité aux personnes handicapées

Pour tout besoin  
d'accessibilité, contactez  
Véronique Constantin,  
référente handicap :  
v.constantin@cit.fr

# DOCKER - créer et administrer des conteneurs virtuels d'applications

## Réseaux

- Comprendre la pile réseau de Docker
- Réseau virtuel et lien entre conteneurs, inspection des réseaux
- Étude des drivers réseaux standards
- Vie des réseaux personnalisés

## Administration

- Applications multi-conteneurs avec Compose
- Méthodes d'administration de conteneurs en production
- Orchestration avec Docker Machine
- Présentation de Swarm pour le clustering
- Configuration réseau et sécurité dans Docker