

Délai maximum : 2 mois.

Parcours concourant au développement des compétences. Action de formation réalisée en application des articles L 6313-1 et L 6313-2 du Code du travail.

Si vous êtes en situation de handicap, contactez-nous avant le début de votre formation pour que nous puissions vous orienter efficacement et vous accueillir dans les meilleures conditions.

- ### Objectifs pédagogiques
- Comprendre le principe de Docker Savoir mettre en œuvre la solution ainsi que les produits de l'écosystème
 - Être en mesure de déployer des images tout en intégrant les contraintes de production

- ### Pré-requis
- Connaissances de base d'un système Linux/Unix et de la programmation
 - Expérience souhaitable de l'utilisation d'un de ces systèmes

- ### Modalités pédagogiques
- Modalités de formation:
- Formation réalisée en présentiel, à distance ou mixte,
 - Toutes nos formations peuvent être organisées dans nos locaux ou sur site
 - Feuille de présence signée en demi-journée, questionnaires d'évaluation de la satisfaction en fin de stage et 60 jours après, attestation de stage et certificat de réalisation.
 - Horaires de la formation: 9h - 12h30 et 13h30 - 17h.
 - Les horaires de la formation sont adaptables sur demande.

- ### Moyens pédagogiques
- Formateur expert dans le domaine,
 - Mise à disposition d'un ordinateur, d'un support de cours remis à chaque participant,
 - Vidéo projecteur, tableau blanc et paperboard,
 - Formation basée sur une alternance d'apports théoriques et de mises en pratique
 - Formation à distance à l'aide du logiciel Teams pour assurer les interactions avec le formateur et les autres stagiaires, accès aux supports et aux évaluations. Assistance pédagogique afin de permettre à l'apprenant de s'approprier son parcours. Assistance technique pour la prise en main des équipements et la résolution des problèmes de connexion ou d'accès. Méthodes pédagogiques : méthode expositive 50%, méthode active 50%.

Public visé

- Administrateurs, chefs de projet et toute personne souhaitant mettre en œuvre Docker pour déployer ses applications

Modalités d'évaluation et de suivi

- Evaluation des acquis tout au long de la formation : QCM, mises en situation, TP, évaluations orales...

Programme de formation

1. Introduction

- Présentation docker, principe, fonctionnalités
- Besoins : packaging d'applications, déploiement rapides, coexistence de plusieurs versions d'une application sur un même serveur

2. Principe et architecture

- Les conteneurs lxc.
- Présentation de lxc : Linux containers, historique, principe de fonctionnement
- Les Cgroups

Contacts

 Notre centre à **Mérignac**
14 rue Euler
33700 MERIGNAC
 05 57 92 22 00
 contact@afib.fr

 Notre centre à **Périgueux**
371 Boulevard des Saveurs,
24660 COULOUNIEUX CHAMIERES
 05 64 31 02 15
 contact@afib.fr

- 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

3. Gestion des images et des conteneurs

- Utilisation de DockerFile pour créer de images personnalisées : principales instructions (RUN, FROM, ENV, EXPOSE, etc ...)
- Recommandations et bonnes pratiques d'écriture de DockerFile
- Gestion des conteneurs : création, affichage, sauvegarde de l'état
- Présentation du Docker Hub
- Publication d'images vers un registry

4. Volumes de données

- Initialisation des volumes de données lors de la création d'un container
- Ajout de volumes de données, contrôle avec la commande Docker inspect
- Sauvegarde, migration, restauration de volumes
- Création de conteneurs de volumes de données

5. 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