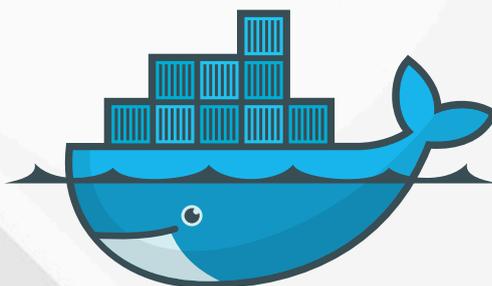




Plan de Formation

Docker



Description de la formation



En très peu de temps, Docker est devenu une plateforme incontournable que les développeurs et les administrateurs système utilisent pour le packaging, le déploiement et l'exécution d'applications.

Docker permet entre autres aux développeurs de créer des images, artefacts immutables contenant le code d'une application ainsi que l'ensemble des dépendances nécessaires pour la lancer. Une image est instanciée en un container, processus ayant sa propre vision du système et ses propres contraintes d'utilisation des ressources. Ces images sont portables et peuvent être déployées et instanciées sur tous les environnements (intégration, QA, production).

Docker facilite notamment l'automatisation des opérations de déploiement, permet d'isoler les applications et d'optimiser l'utilisation de ressources.

Docker



Objectifs

Opérationnel

Créer et administrer des conteneurs d'applications avec Docker, et déployer, gérer et sécuriser les hôtes Docker.

Pédagogiques

Concrètement, à l'issue de ce cours Docker, les participants disposeront des connaissances et des compétences nécessaires pour :

- Comprendre la place de Docker dans le nouveau paradigme du mouvement DevOps
- Créer et administrer des conteneurs Docker (image, registre, conteneur...)
- Construire ses propres images Docker et les déployer
- Comprendre comment créer et déployer des applications multi-conteneurs
- Comprendre le nouveau paradigme des applications Micro-services
- Créer et déployer des applications Micro-services avec Docker-compose
- Gérer la journalisation des conteneurs avec la suite ELK

Docker

- Comprendre ce qu'est un pipeline de développement et d'intégration continue (CI, CD)
- Comprendre l'apport d'un orchestrateur comme Kubernetes
- Identifier les risques et challenges inhérents à Docker afin d'anticiper les bonnes solutions
- Savoir gérer les problématiques de sécurité des conteneurs, connaître les bonnes pratiques à adopter
- Les fonctions avancées de la construction d'image avec Dockerfile, Arguments, variables, sondes...
- Développer et déployer des applications avancées multi-containers avec Docker Compose
- Concepts d'orchestration de conteneurs
- Comprendre le concept d'applications « Orchestrator-ready »
- Comprendre l'architecture de Kubernetes : Les différents types et rôles des nœuds
- Notion de pods, service, stockage et déploiements
- Gérer le cycle de ses déploiements

Docker

- Gérer les mises à jour des applications
- Déploiement et partage des éléments de configuration
- Comprendre les applications Stateless et Stateful
- Comprendre les outils de l'écosystème Kubernetes
- Déployer des applications complexes avec le manager Helm
- Gérer le réseau avec le load balancer et les ingress traefik, nginx...
- L'importance des services mesh avec Istio
- L'importance des services des solutions de stockage
- L'importance du monitoring de l'infrastructurelle et des applications

A qui s'adresse cette formation ?

Public

Le cours s'adresse aux administrateurs, architectes ou développeurs d'applications voulant apprendre à gérer les problématiques d'exploitation et optimiser leurs bonnes pratiques.



Prérequis

Des connaissances de base de l'administration Linux et du langage Shell sont nécessaires pour suivre cette formation dans de bonnes conditions.

Docker

Contenu du cours

Présentation de Docker

- Les différents niveaux de virtualisation (système, applicatif)
- Avantages de la virtualisation applicative
- DevOps, des nouveaux outils et de nouvelles habitudes de travail
- Comprendre l'Infrastructure As Code
- Vocabulaire autour du paysage informatique (Web Services, Micro-services, SOA, CI/CD, etc.)
- La place de Docker et des conteneurs dans ce paysage

Travaux pratiques : Séquence de travail type d'un utilisateur de Docker pour bien comprendre les fonctionnalités apportées par Docker et en quoi il change les habitudes techniques et organisationnelles des équipes

Docker

Contenu du cours

Architecture de Docker

- Client et moteur Docker
- Les composants de base : images, registry, container
- Comprendre le daemon Docker et l'architecture client/serveur, l'API Docker
- Les éditions de Docker
- Implémentation sous Linux et Windows
- La gestion des images Docker
- Kernel, images et conteneurs
- Ecosystème Docker (Docker Hub, Docker registry, Dockerfile, Docker Compose)
- Docker Desktop
- Docker et le Cloud

Travaux pratiques :

- Installation de Docker Community Edition
- Initiation aux conteneurs
- Présentation d'un conteneur
- Création d'une image
- Exploitation applicative standard
- xploitation applicative du stockage

Contenu du cours

Images et conteneurs Docker

- Liens entre image et conteneur
- Qu'est-ce qu'un registry de conteneurs ?
- L'exemple de Docker Hub
- La registry privée
- Nomenclature d'appels de conteneurs
- Savoir télécharger (pull) des images, comprendre le nommage des images Docker
- Créer un compte sur le Hub Docker et pousser (push) des images via la CLI Docker
- Comprendre l'historique des images
- Administrer les images en local, nommage, suppression, le cache
- Principes d'instanciation d'un conteneur
- Créer une image à partir d'un conteneur
- Gestion de la console au quotidien

Docker

Contenu du cours

Travaux pratiques :

- Navigation dans le Docker Hub
- Administrer ses images, pull, tag, push, history
- Créer un compte sur le registre
- Envoyer des images dans le registre Docker (Hub Docker)
- Créer un registre privé
- Administrer ses conteneurs avec les options du run
- Se connecter sur les conteneurs
- Installation d'applications Web dans un conteneur et création -images à partir de ce dernier

Contenu du cours

Docker et le réseau

- Comment exposer ses images vers l'extérieur ?
- Comprendre le réseau Docker
- La notion de publication de port
- Publier des ports avec publish
- Créer des réseaux Docker, dhcp, dns
- Communications inter conteneurs
-

Travaux pratiques :

Tests de l'exposition de nos images selon différents modes de configuration du réseau (bridge, hôte, join, link)

Docker

Contenu du cours

Les volumes de données avec Docker

- Le problème de la persistance dans les conteneurs
- Gestion du stockage
- Bien maîtriser les volumes Docker, bind, files, volumes distants
- Identification des volumes orphelins

Travaux pratiques :

- Créer des volumes Docker
- Associer des systèmes de fichiers avec vos conteneurs (local et distant)

Contenu du cours

Dockerfile

- Comprendre la construction d'image avec le Dockerfile
- Les instructions du Dockerfile : FROM, RUN CMD, ENTRYPOINT
- Les bonnes pratiques (nettoyage, lancement des services, sécurité...)
- Déployer son propre registre Docker
- Présentation d'Harbor

Travaux pratiques :

- Création d'une image à l'aide de Dockerfile
- Optimisation de la création des images
- Créer une image MongoDB à l'aide de la description GitHub
- Création d'une image LAMP
- Construction, déploiement d'un registre Docker via Dockerfile
- Identification et push d'images personnalisés sur le registre

Contenu du cours

Application multi-conteneurs (docker-compose)

- Comprendre l'orchestration des Micro services avec docker-compose
- Le format yaml de fichier défini par docker-compose
- La commande docker-compose
- Mon premier service
- Le réseau avec docker-compose
- Les volumes de données avec docker-compose
- Les dépendances entre services
- Mise à jour des services

Travaux pratiques :

- Déploiement d'une application Web, PHP, SQL, hautement disponible avec HAProxy, Apache...
- Déploiement d'un stack de journalisation avec la suite ELK (Elasticsearch, Logstash, Kibana)
- Configurer la journalisation avec Docker-compose

Contenu du cours

Fonctions avancées de docker-compose

- Construire des images avec docker-compose, les contextes
- Rendre dynamiques ses fichiers compose avec les variables et le fichier. env
- Les sondes de vitalité et de disponibilité
- Comprendre les labels
- Configuration avancée des volumes
- Le Rollback et update des services

Travaux pratiques :

- Créer un environnement complexe avec le fichier. env et les variables
- Créer des sondes vitalité health check
- Créer des labels pour échanger avec le démon Docker et d'autres applications Docker
- Gérer ses mises à jour

Docker

Contenu du cours

Sécurité

- Éléments à risque : noyau, conteneurs, Docker daemon...
- Quelques bonnes pratiques d'administration
- Scan de vulnérabilité des images
- Les bonnes pratiques dans le Dockerfile
- La capabilities Linux/Syscall

Travaux pratiques :

Sécurisation des images et tests d'accès

Contenu du cours

Exploitation de docker au quotidien

- Les plugins Docker
- Administration des plugins
- Principe de la copie sur écriture (CoW)
- Intégration à Docker de la copie sur écriture
- Les pilotes de stockage Docker (AUFS, OverlayFS, DeviceMapper...)
- Configuration du daemon Docker
- Démarrage automatique des services
- Gestion des logs
- Restriction sur les ressources
- Protection du contenu d'un conteneur
- Bonnes pratiques de sécurité

Travaux Pratiques :

- Implémentation d'un plugin de stockage
- Redémarrage automatique
- Déploiement d'un frontal web d'administration

Contenu du cours

Introduction à l'intégration continue avec GitLab/Jenkins

- Retour sur l'automatisation de sa chaîne de production
- Construire et pousser des images Docker avec Jenkins et GitLab
- Le Dockerfile et Jenkinsfile

Travaux pratiques :

Mise en place d'un pipeline d'intégration continue avec GitLab et Jenkins

Introduction à l'orchestration de conteneurs

- Limites de Docker seul et apport d'un orchestrateur (haute disponibilité, scalabilité...)
- Comprendre l'architecture d'un orchestrateur de conteneurs
- Dialogue entre les nœuds d'un cluster
- L'exemple de Kubernetes
- Le vocabulaire de Kubernetes (pods, service...)
- Comprendre les applications stateless et statefull

Contenu du cours

Travaux Pratiques :

Démonstration du formateur pour comprendre comment s'exécutent les images Docker dans un contexte d'orchestration et la plus-value concrète d'un outil comme Kubernetes pour les développeurs (déploiement d'une application LAMP à haute disponibilité par exemple)

La sécurité avec Docker

- Éléments à risque : noyau, conteneurs, Docker daemon...
- Quelques bonnes pratiques d'administration
- Scan de vulnérabilité des images
- Utilisation d'Harbor pour sécuriser ses images
- Les bonnes pratiques dans le Dockerfile
- La capability Linux/SysCall

Contenu du cours

Dockerfile : Build d'images avancées

- Présentation d'un environnement de build
- Les variables du fichier .env
- Les instructions : HEATCHECK...
- La construction multi-stage
- Rendre dynamiques ses builds avec les arguments et variables
- Le DockerIgnore

Travaux pratiques :

Déploiement d'un environnement de construction d'image destiné à la production

Contenu du cours

Fonctions avancées de Docker Compose

- Retour sur Docker Compose
- Construire des images avec Docker Compose, les contextes
- Rendre dynamiques ses fichiers Compose avec les variables et le fichier. env
- Configuration avancée du réseau
- Configuration avancée des volumes
- Gérer les logs Docker avec ELK

Travaux pratiques :

Créer un environnement complexe et dynamique avec le fichier. env et les variables

Configurer un réseau

Configurer des volumes distants NFS...

Mise en œuvre d'une stack de journalisation avec la suite Elastic, Kibana

Contenu du cours

Orchestration des conteneurs

- Concepts d'orchestration des conteneurs
- Comprendre l'orchestration, la haute disponibilité, la mise à l'échelle des applications
- Comprendre l'architecture d'un orchestrateur de conteneurs
- Le manifest YAML
- La notion de pods, de service, de stockage et de déploiements
- Gérer les sondes, les ressources dans les pods
- Comprendre les applications Stateless et Stateful

Travaux pratiques :

Créer nos premiers pods

Gérer les ressources dans les pods

Gérer des sondes d'activité des pods

Contenu du cours

Le déploiement d'applications

- Comprendre la ressource Kubernetes « deployment »
- Comprendre la réplication des pods et leur cycle de vie
- Comprendre et savoir gérer les mises à jour des deployments, rollouts

Travaux pratiques :

- Créer et déployer notre déploiement d'applications
- Administrer les ressources dans les pods
- Créer des sondes d'activité dans les pods

Docker

Contenu du cours

La communication réseaux

- Comprendre la ressource Service
- Comprendre et savoir exposer nos applications aux utilisateurs
- Comprendre et savoir-faire communiquer nos applications entre elles

Travaux pratiques :

- Exposer des applications par ports - NodePort
- Exposer des applications par IP – LoadBalancer
- Créer un réseau interne pour nos applications – Cluster IP
- Exposer ses applications via des URLs avec les Ingress

Contenu du cours

Le stockage pour nos applications

- Comprendre la gestion des Volumes avec Kubernetes, PV, PVC...
- Savoir fournir du stockage dynamique à nos déploiements
- Connecter le stockage avec les pods

Travaux pratiques :

- Créer une demande de stockage dynamique
- Associer le stockage avec ses applications
- Le déploiement d'applications avec états
- Comprendre la ressource Kubernetes « StatefulSets»
- Comprendre le déploiement d'applications avec états
- Exemple d'une base de données Postgre clusterisée

Contenu du cours

Les projets de l'écosystème Kubernetes

- Revue de différents projets développés autour de Kubernetes via la Cloud Native Computing Foundation
- (Containerd, Istio, Rook, Argo CD, Harbor, Fluentd, Prometheus...)
- Packaging et déploiements d'application complexes avec Helm
- Comprendre l'implication des services Mesh, les réseaux maillés avec Istio
- Études des solutions de stockage pour Kubernetes, cephfs, Rook
- Étude de cas du monitoring avec Prometheus et Grafana

Travaux pratiques :

- Déployer des application complexe avec le gestionnaire Helm
- Mettre en œuvre le réseau maillé avec Istio



Knowledge forges Empires

Nos Contacts



empire-training.tn



+216 55 826 628
+216 74 201 616



16 Rue D'Athènes, Sfax



Contact@empire-training.tn