



Programme de formation

Architecture basée sur Docker

SYS-DOCK-001
V 1.0 – 07/01/2025



Objectifs opérationnels

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

- Comprendre et utiliser Docker pour conteneuriser une application
- Gérer images, containers et volumes
- Orchestrer plusieurs services avec Docker Compose
- Mettre en place une architecture de tâches asynchrones avec RabbitMQ et Celery
- Déployer et monitorer une stack complète conteneurisée

Public visé

- Développeurs backend
- DevOps
- Architectes techniques
- Ingénieurs logiciels

Prérequis

- Bases Linux (terminal, processus)
- Connaissances Python
- Notions d'API et d'architecture web

Durée

14 heures

Programme détaillé

Introduction à Docker et ses concepts clés (1h)

- Brève description de Docker (plateforme de conteneurisation).
- Pourquoi utiliser Docker ? (isolation, portabilité, légèreté)
- Container, Image
- Docker Engine
- Docker Hub
- Différences entre VM et conteneurs

Utilisation des containers, des images et des volumes (2h)

Images

- Utilisation de Docker Hub pour récupérer une image.
- Commandes : docker pull, docker build, docker tag, docker push
- Mise en pratique

Containers

- Créer un container à partir d'une image
- Commandes de base : docker run, docker ps, docker stop, docker rm



- Mise en pratique

Volumes

- Importance des volumes pour la persistance des données.
- Commandes liées aux volumes : docker volume create, docker volume ls, docker volume rm
- Mise en pratique

Docker Compose (1h)

- Qu'est-ce que Docker Compose ?
- Structure du fichier docker-compose.yml.
- Exemple d'un fichier docker-compose.yml simple.
- Commandes de base : docker-compose up, docker-compose down, docker-compose logs.
- Mise en pratique
- Introduction à Swarm

RabbitMQ et Celery (3h)

RabbitMQ

- Cas d'utilisation typiques : gestion des tâches asynchrones et traitement par lots.
- Notions clés : Exchange, Queue, Binding.
- Introduction à RabbitMQ Manager
- Mise en pratique

Celery

- Utiliser Celery avec RabbitMQ comme broker.
- Création du worker, paramétrage
- Exposer une tâche
- Démarrer et contrôler le worker
- Mise en pratique

Celery Client

- Création du client, paramétrage
- Exposer le prototype d'une tâche
- Démarrer une tâche, obtenir son id
- Récupérer une tâche à partir de son id, connaître son status, résultat ...
- Mise en pratique

Flower

- Vue d'ensemble des fonctionnalités (monitoring des tâches, workers, files d'attente).
- Installation de Flower via Docker ou pip.
- Mise en pratique

Customisation et documentation



- Introduction aux possibilités offertes par Celery
- Présentation de la doc
- Exemple des events

Mise en pratique : Déploiement d'une stack complète (4h)

- Présentation de la stack à déployer : worker Python, RabbitMQ, Celery, PostgreSQL
- Créer le fichier docker-compose.yml
- Lancer la stack
- Tester une tâche asynchrone avec le worker Python.
- Monitoring et gestion des files d'attente.

Conclusion et points à retenir (0,5h)

- Résumé des concepts et outils vus.
- Importance de l'orchestration avec Docker Compose.
- Comment Celery, RabbitMQ et Docker peuvent être utilisés pour des architectures distribuées.

Évaluation (1h30)

Modalités pédagogiques

- Apports théoriques structurés
- Démonstrations techniques
- Exercices guidés
- Mise en situation réelle
- 65 % de pratique

Modalités d'évaluation

- Quizz final
- Évaluation pratique finale : déploiement complet de la stack
- Validation des acquis via exercice supervisé

Moyens techniques

- Ordinateur personnel
- Docker Desktop ou Docker Engine installé
- Accès Internet
- IDE recommandé (VS Code ou équivalent)

Modalités d'accès

- Huit personnes maximums
- La formation est effectuée chez le client
- Le client doit disposer d'une salle capable d'accueillir l'ensemble des formés et deux formateurs



Accessibilité

Adaptations possibles sur demande (rythme, supports, modalités).

Prix

4000 € HT

Contact

John WALTER
John.walter@wasssa.io