INF5190 - Déploiements

Jean-Philippe Caissy

6 novembre 2019

Déploiement

Le déploiement consiste à rendre une application accessible au public.

- Pour une application Web : il s'agit d'exposer l'application à travers une URL accessible à partir d'Internet.

Déploiement

```
$ FLASK_DEBUG=1 FLASK_APP=poll flask run
 * Serving Flask app "poll" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 137-450-813
```

L'application est accessible localement à partir de
`http://127.0.0.1:5000/`

Comment la rendre accessible sur une adresse publique ?

Déploiement

Modèle *4-tier*

La norme pour les application Web est d'utiliser un déploiement avec *4-tier*, ou 4 environnements :

- Développement
- Test
- Mise en scène (*staging*)
- Production

Déploiement

Modèle *4-tier*

Développement

- Environnement local, là où un développeur modifie le code (e.g.: votre laptop)
- Depuis le début du cours, vous travaillez dans un environnement de développement

Test

- Environnement d'intégration
- Un développeur *merge* son code dans cet environnement pour faire rouler une suite de test complète et s'assurer du fonctionnement des modifications (e.g.: intégration continue, QA,)

Déploiement

Modèle *4-tier*

Mise en scène (*staging*)

- Reproduction de l'infrastructure de production
- S'assure que les changements vont fonctionner avec des données et une infrastructure similaire à celle de production
- Peut souvent ne pas être utilisé au détriment d'un plus haut risque de défaillance

Production

- L'environnement accessible directement par le public

Déploiement

Un déploiement vers l'environnement de production peut se faire manuellement ou automatique

Manuel

- Envoie du code sur une machine distance (copier-coller, `git pull`, upload FTP
- Création d'une image Docker manuellement avec `docker build`, lancement de l'image sur le serveur distant
- Ouverture des connexions réseaux
- Migration de la base de donnée manuellement

Automatique

- Déploiement assisté
- Chaque *commit* génère une image Docker
- L'image Docker peut être déployé avec l'interaction d'un utilisateur, ou automatique
- **Automatisation**

Déploiement

Automatique

- Un déploiement automatique est aussi utile pour un petit service qu'un déploiement complexe sur des centaines de serveurs
  - Exemple :
    - une commande qui fait le déploiement
    - intégration continue qui déploie si les tests passent
    - déploiement à partir d'une interface graphique

Déploiement

Automatique

Un système de déploiement plus complexe permet d'automatiser et facilite les fonctionnalités suivantes :

- Gestion d'un groupe de conteneur Docker
- Mise à l'échelle (*scaling*)
- Balançage de charge (*load balancing*)
- Modèle déclaratif : définition des états voulus
- Surveillance et réconciliation des états
- Gestion du réseau
- Déploiement sans interruption

Déploiement

Automatique

Cas d'utilisation : Shopify et le déploiement automatisé
https://engineering.shopify.com/blogs/engineering/automatic-deployment-at-shopify

Déploiement

Hébergement d'une application Web

Il existe 4 options pour déployer et héberger une application Web :

1. Serveur physique (*bare metal*)
2. Serveur virtuel (*VPS*)
3. IaaS (*Infrastructure as a Service*)
4. PaaS (*Platform as a Service*)

Déploiement

Hébergement d'une application Web

Serveur Physique

- Peut consister à acheter le serveur physique
  - Quelques milliers de dollars, plus une location d'emplacement dans un centre de donnée
- La location d'un serveur consiste à payer un montant mensuel (ou annuel) pour avoir accès à un servuer complet
  - Dans les centaines de dollars par mois
  - Exemple : OVH permet de louer des serveurs dans la région de Montréal à partir de 80$/mois

Déploiement

Hébergement d'une application Web

Serveur Virtuel

- Tel que vu au dernier cours, il s'agit d'avoir un environnement entièrement virtualisé. Il s'agit de rouler un OS complet (Linux ou Windows) sous virtualisation
- Prix à partir de quelques dollars par mois
- Exemple : Digital Ocean, OVH, Linode, Lightsail (AWS), Google Compute Engine

Déploiement

Hébergement d'une application Web

Serveur Virtuel

Démo avec Digital Ocean

Déploiement

Hébergement d'une application Web

IaaS

- *Infrastructure as a Service* sont des solutions complète de gestion d'infrastructure sans avoir à manuellement gérer l'infrastructure physique.
- Souvent utilisé avec des serveurs virtuels
- Analogue à des blocs Lego : on assemble plusieurs blocs ensemble pour créer une infrastructure
  - Machines virtuels, volumes, réseau, système de logs, base de donnée, DNS, etc
- Coûts très flexible et granularité souvent à l'heure
- Exemple : Amazon Web Services (AWS), Google Cloud Patform (GCP)

Déploiement

Hébergement d'une application Web

PaaS

- Une solution de *Platform as a Service* abstrait l'infrastructure complète
  - Il suffit de respecter un standard, et le déploiement est fait automatiquement
- Le *PaaS* définit les mécanismes d'accès des ressources par l'application Web (temps de serveur, base de donnée, etc)
- Coûts à partir de quelques dizaines de dollars par mois
- Exemple : Heroku, Python Anywhere, Google App Engine

Déploiement

Hébergement d'une application Web

PaaS

Exemple avec Heroku!

Déploiement

Docker Compose

- Utilitaire qui permet de définir et rouler plusieurs conteneurs Docker
- Pratique lorsqu'une application a besoin de plusieurs services
- Les conteneurs sont définis par un fichier YAML nommé `docker-compose.yml`
- Efficace pour un environnement de développement ou de test

Déploiement

Docker Compose

Fichier `docker-compose.yml`

```yaml
version: '3'
services:
  web:
    build: .
    ports:
      - "5000:5000"
  mysql:
    image: "mysql:5.7"
    restart: always
    ports:
      - "3306:3306"
  volumes:
    - my-db:/var/lib/mysql
volumes:
  my-db:
```

Déploiement

Docker Compose

Demo !

Déploiement

Orchestration

La gestion de 10 conteneurs pour 4 applications web est très facile.
Quand est-il lorsqu'on a 1 000 conteneurs et 400 applications Web?

- Gestion du cycle de vie d'un conteneur
  - Construction et déploiement des conteneurs
  - Redondance et disponibilité des conteneurs
  - Mise à l'échelle (*scaling*)
  - Déplacement d'un conteneur vers un autre hôte
  - Exposition des services à Internet
  - Monitoring des hôtes et conteneurs (*healthcheck*)
  - Configuration d'une application

Déploiement

Orchestration

Illustration au tableau

Déploiement

Orchestration

Pour le moment, la solution standard d'orchestration est **Kubernetes**. Docker offre un système similaire avec **Docker Swarm**.

Déploiement

Orchestration

Docker Swarm

@ TODO

Kubernetes

@ TODO

Déploiement

Progressif

- Un déploiement progressif se fait sans interrompre une application Web.
- L'objectif est d'instancier des nouvelles images de l'application de manière progressif
- Exemple :
  - On arrête 25% des conteneurs de l'ancienne version
  - On les remplace par une nouvelle version de l'application
  - On continue pour un autre 25%, etc
- Illustration au tableau

Ce qui veut dire que deux versions de l'application peu rouler en même temps!

Déploiement

Progressif

- Puisque deux applications peuvent rouler à tout instant en même temps, il faut faire attention au code.
- Déploiement de fonctionnalités en deux phases lorsque nécessaire :
  - Le code qui introduit le changement est déployé, mais non utilisé
  - On redéploie une deuxième fois et activant les fonctionnalités

Exemple : démonstration de l'ajout d'une nouvelle route au tableau

Déploiement

Canarie

Figure 1: Canarie dans une mine

Déploiement

Canarie

Objectif : détecter les défaillances et problèmes le plus rapidement possible

1. Déploiement partiel d'une nouvelle version de l'application (ex: 1% des conteneurs)
2. Faire passer un pourcentage minimale (ex: 1%, 5%) du trafic sur cette nouvelle version
3. Observer les problèmes et exceptions
4. Si aucune exception, déployer sur un plus grand nombre de conteneurs (ou sur tous les conteneurs)

Liens

- The Four-Tier Deployment model
- Get started with Docker Compose
- How to Create a MySql Instance with DockerCompose
- Automatic Deployment at Shopify