

# Systemes distribués

TP2: Systeme distribué à base de courtier de messages

# Objectifs

---

- ▶ Développer un système distribué (en cluster) où les tâches sont réparties sur plusieurs nœuds qui communiquent de manière asynchrone en utilisant RabbitMQ
- ▶ Avoir un système modulaire pour le traitement de données générées par une application

# Énoncé

---

## ▶ Programme I: P I

### ▶ Mission:

- ▶ **Récupérer**, à l'intérieur d'une boucle, **des données** (texte en Anglais + images) d'une source distante (site web) ou locale (base de données ou fichiers json) et les soumettre à un cluster RabbitMQ pour qu'elles soient traitées par d'autres programme
- ▶ **PI soumis le texte** (un paragraphe à la fois) pour traduction à une file d'attente
- ▶ **PI soumis les images** (une à la fois) à une autre file d'attente pour que plusieurs versions de l'image soient produites

# Énoncé

---

## ▶ **Programme 2: P2**

### ▶ **Mission:**

- ▶ **Traduction** du texte de l'anglais au français
- ▶ P2 récupère les textes en anglais et les traduits en français
- ▶ P2 envoie le texte original et le texte traduit à rabbitMQ

# Énoncé

---

- ▶ **Programme 3: P3**

- ▶ **Mission:**

- ▶ **Traitement des images**

- ▶ P3 récupère une image à la fois et **génère deux images** de tailles différentes et les transmis à rabbitMQ

# Énoncé

---

## ▶ **Programme 4: P4**

### ▶ **Mission:**

#### ▶ **Persistance des résultats dans une BD relationnelle**

- ▶ P4 récupère de rabbitMQ les résultats des traitements effectués par P2 et P3 et les sauvegarde dans une base de données relationnelle (sqlite, mysql/mariadb, postgresql, etc.)
  - ▶ Texte: prévoir deux colonnes: une pour la version FR et l'autre pour la version EN (ou un champ langue)
  - ▶ Images: sous forme de blob

# Consignes

---

## ► Cluster:

- La base de données doit être mise en cluster multi-master en utilisant par exemple MariaDB Galera Cluster
- RabbitMQ doit être déployé en mode cluster avec des files d'attente en miroir.
- Des échangeurs de type topic sont à utiliser

# Consignes

---

- ▶ Environnement recommandé
  - ▶ Systèmes d'exploitation:
    - ▶ Linux (Centos 7.x)
  - ▶ Langages de programmation:
    - ▶ Utiliser au moins deux langages de programmation différents pour réaliser les quatre programmes (exemple: p1 en JS et p2, p3 et p4 en Java ou Ruby)
  - ▶ Courtier de messages:
    - ▶ RabbitMQ
  - ▶ Base de données:
    - ▶ MariaDB



# Consignes

---

- ▶ Travail à effectuer en équipes de 2 personnes
- ▶ À remettre sur Moodle par un seul membre de chaque équipe:
  - ▶ Fichier zip/7z/rar contenant le code de votre système et le rapport
- ▶ La version sur GitLab doit être aussi à jour
- ▶ Dernier délai pour la remise du travail:
  - ▶ Le 15 avril à 10h

# Évaluation

---

- ▶ Les éléments suivants seront pris en considération lors de l'évaluation de ce travail:
  - ▶ Fonctionnement:
    1. Sur une seule machine
    2. En mode distribué (idéalement un programme par VM, mais peut tolérer deux programmes par VM)
  - ▶ Un rapport expliquant:
    - ▶ La démarche
    - ▶ Le schéma de base de données relationnelle proposée
    - ▶ Les problèmes rencontrés et les solutions proposées
    - ▶ Le rôle de chaque classe/programme
    - ▶ Comment déployer et lancer les différents modules de votre système avec captures d'écrans
  - ▶ Une démonstration du système
    - ▶ Une séance sera réservée aux présentations
  - ▶ La qualité du code
  - ▶ La complétude du système
  - ▶ L'implication des coéquipiers (gestion de code source avec GIT est obligatoire et les commits doivent être réguliers et impliquant les différents coéquipiers)
  - ▶ La capacité à retrouver l'information pertinente et nécessaire à la réalisation du travail pratique

# Barème

---

- ▶ P1 : 1,5 points
- ▶ P2 : 1,5 points
- ▶ P3 : 1,5 points
- ▶ P4 : 1,5 points
- ▶ Mise en cluster: 1 point
- ▶ Rapport : 1 point
- ▶ Présentation et contributions de chaque coéquipier : 2 points