

Retour d'expérience

# INTÉGRATION CONTINUE DANS UN PROJET MICROSERVICES

*JDEV2017 - T4 P01 - Frédéric Woelffel*

Marseille / 6 juillet 2017



INSTITUT DE CHIRURGIE  
GUIDÉE PAR L'IMAGE

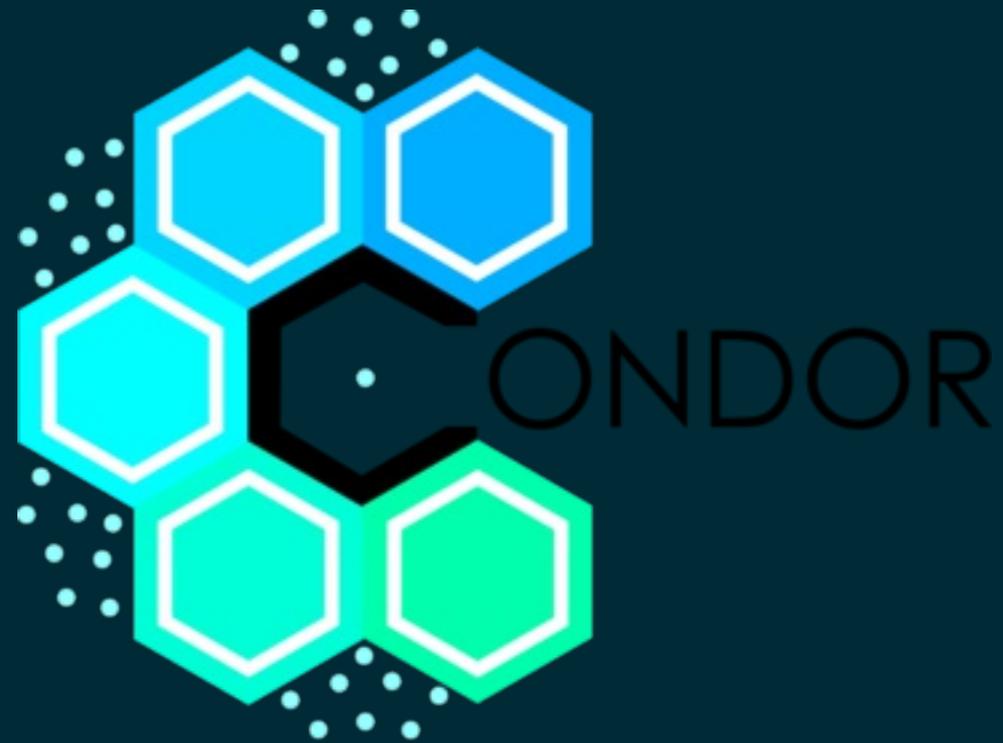


- Frédéric Woelffel
- Ingénieur R&D à l'IHU de Strasbourg depuis septembre 2016
- Développeur Node.js



INSTITUT DE CHIRURGIE  
GUIDÉE PAR L'IMAGE

# LE PROJET CONDOR



# LA PROBLÉMATIQUE

- 6,5 millions d'opérations
- 8000 blocs chirurgicaux
- 90000 problèmes

# LA RÉPONSE

- Captation des données du parcours du patient
- Visualisation des informations obtenues
- Protocoles de communication et données de natures différentes

*The right tool for the right job*

# UNE ARCHITECTURE MICROSERVICES

- Composants simples
- Déploiement facilité
- Adapté à un besoin

*Write programs that do one thing and do  
it well*

- Système distribué
- Incohérence éventuelle
- Complexité opérationnelle

# L'INTÉGRATION CONTINUE

- *Continuous Integration* : Assure l'exécution des suites de tests
- *Continuous Delivery* : Propose un livrable à tout moment
- *Continuous Deployment* : Automatise la mise à jour des services

- Test immédiat (ou presque) des nouvelles révisions du code
- Notification efficace des membres de l'équipe
- Mise à disposition d'un livrable fonctionnel à tout moment

# NOTRE CYCLE DE DÉVELOPPEMENT SANS INTÉGRATION CONTINUE

- Développer un service
- Compiler
- Tester
- Faire un commit
- Eventuellement releaser
- Eventuellement déployer
- Eventuellement vérifier que tout fonctionne
- Aller prendre un café bien mérité

*Si seulement...*

## Fonctionne bien pour

- Un nombre de services restreint
- Un effectif réduit

# LA (TRISTE) RÉALITÉ

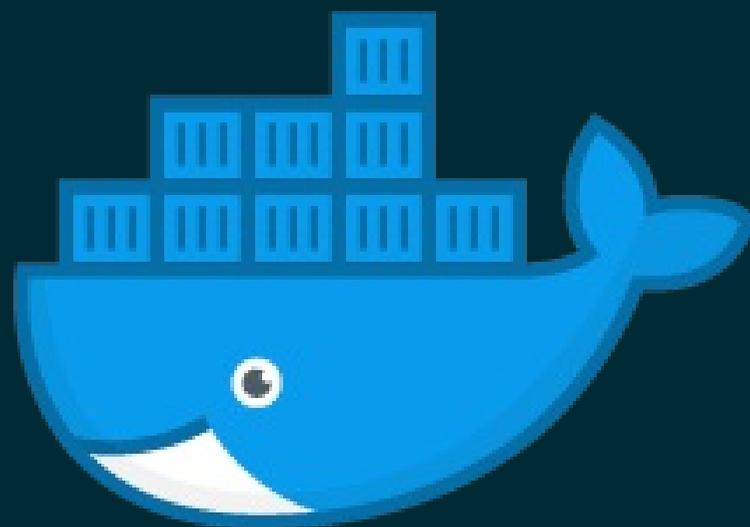
- Développer un service
- Compiler
- Se dépêcher d'aller prendre un café
- Faire un commit
- Releaser
- Déployer
- Démarrer la tâche suivante
- Avec de la chance, se rendre compte qu'on a introduit un bug
- Signaler le bug

Recommencer

- Nous sommes humains (maladresses, oublis)
- Beaucoup de tâches répétitives
- Cycle de développement lent : mauvaise qualité du code, détection des erreurs plus lente, communication compliquée avec le product owner, etc.

Pénible...

# NOTRE ENVIRONNEMENT

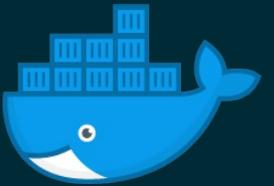


# GITLAB



- RMS git
- Intégration continue assurée par les *Gitlab runners*
- "Community Edition" opensource sous license MIT  
Expat
- Auto hébergeable

# DOCKER & REGISTRY



- Technologie de containerisation
- "Community Edition" opensource sous license Apache 2.0
- Fournit une image de Registry
  - Plateforme d'hébergement de nos images

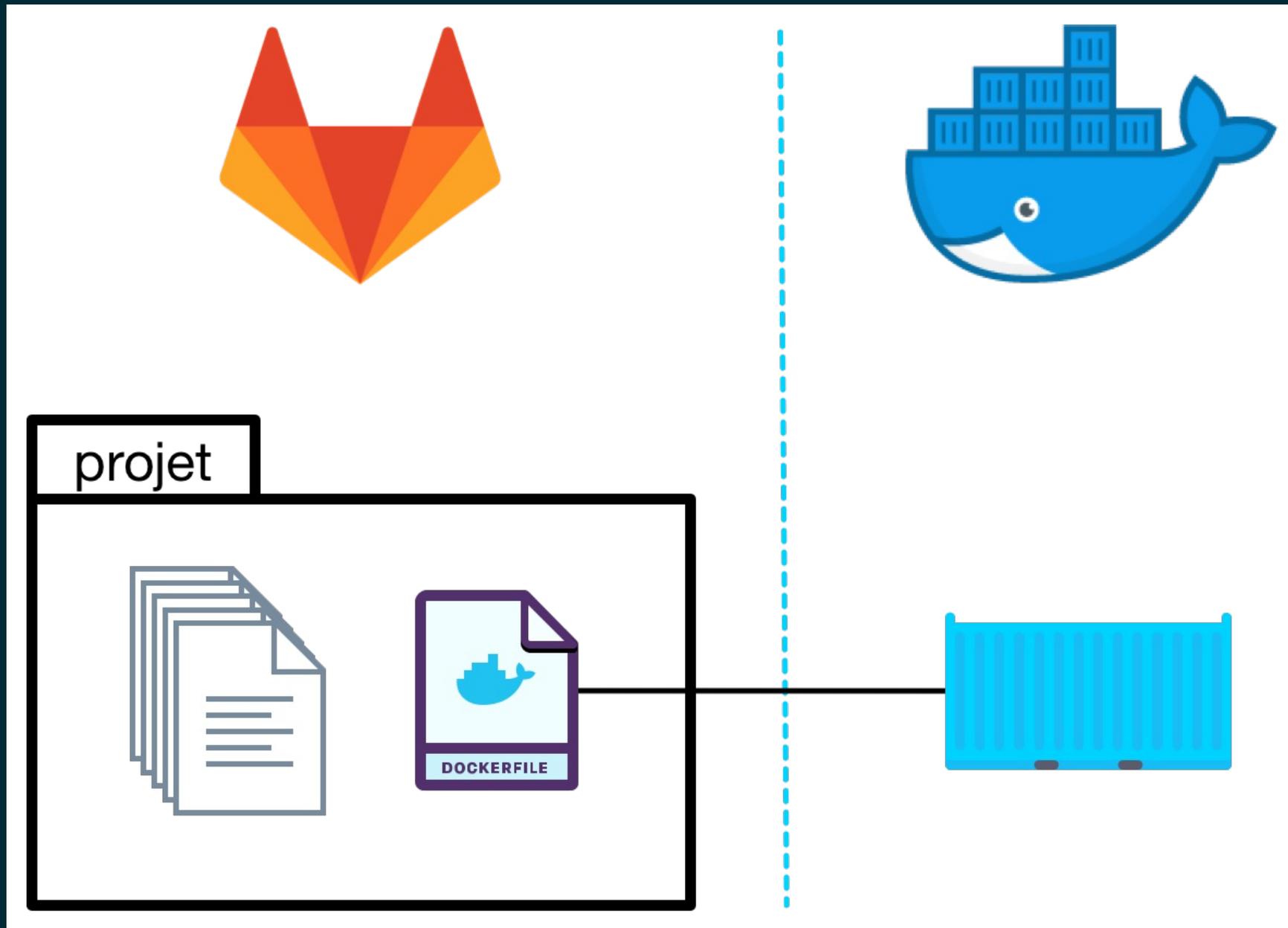
# RANCHER



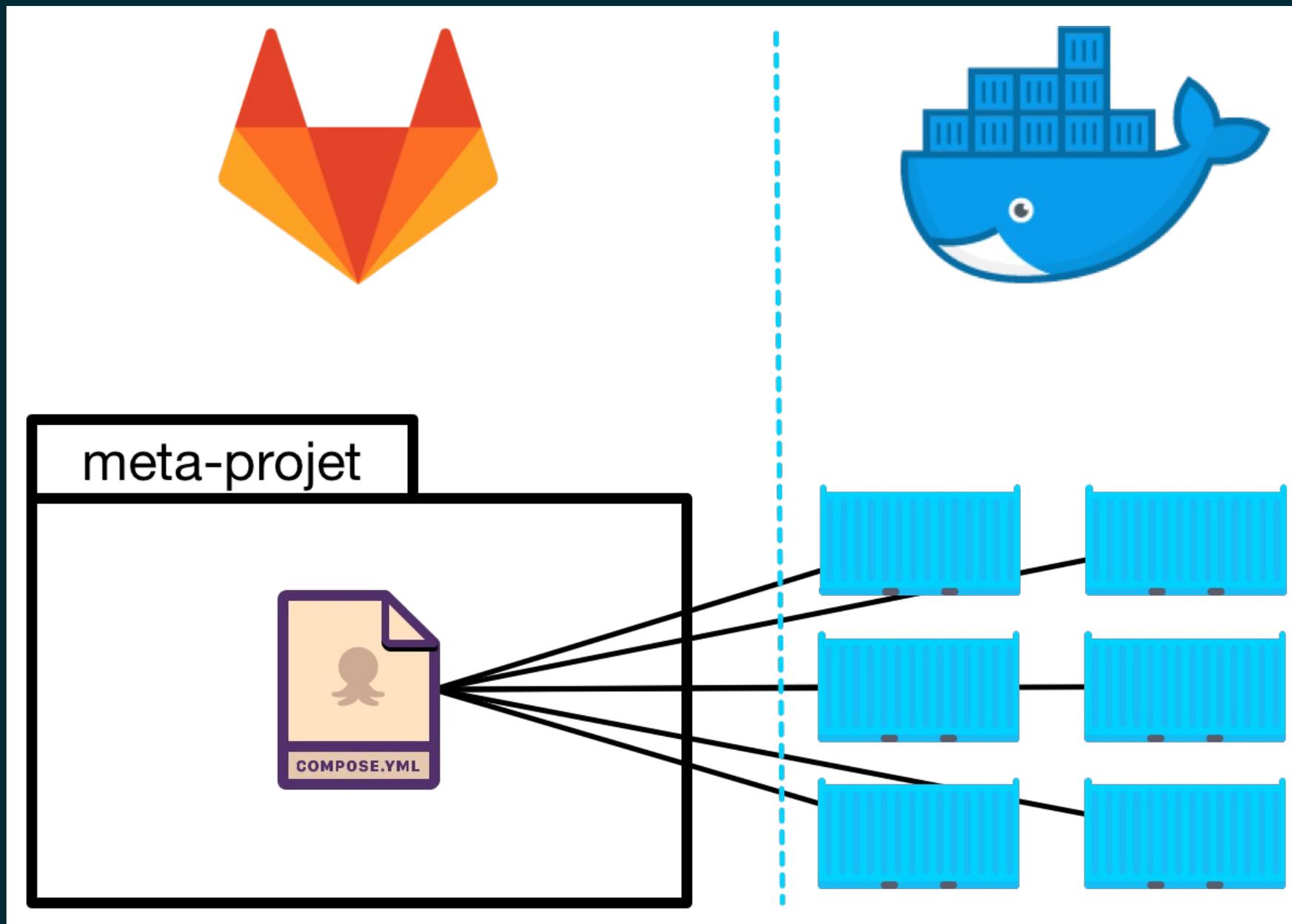
- Outil d'orchestration de containers
- Interface web intuitive
- API REST
- Destiné aux équipes de développement
- Opensource sous license Apache 2.0

# NOTRE SOLUTION

1 service = 1 container = 1 projet



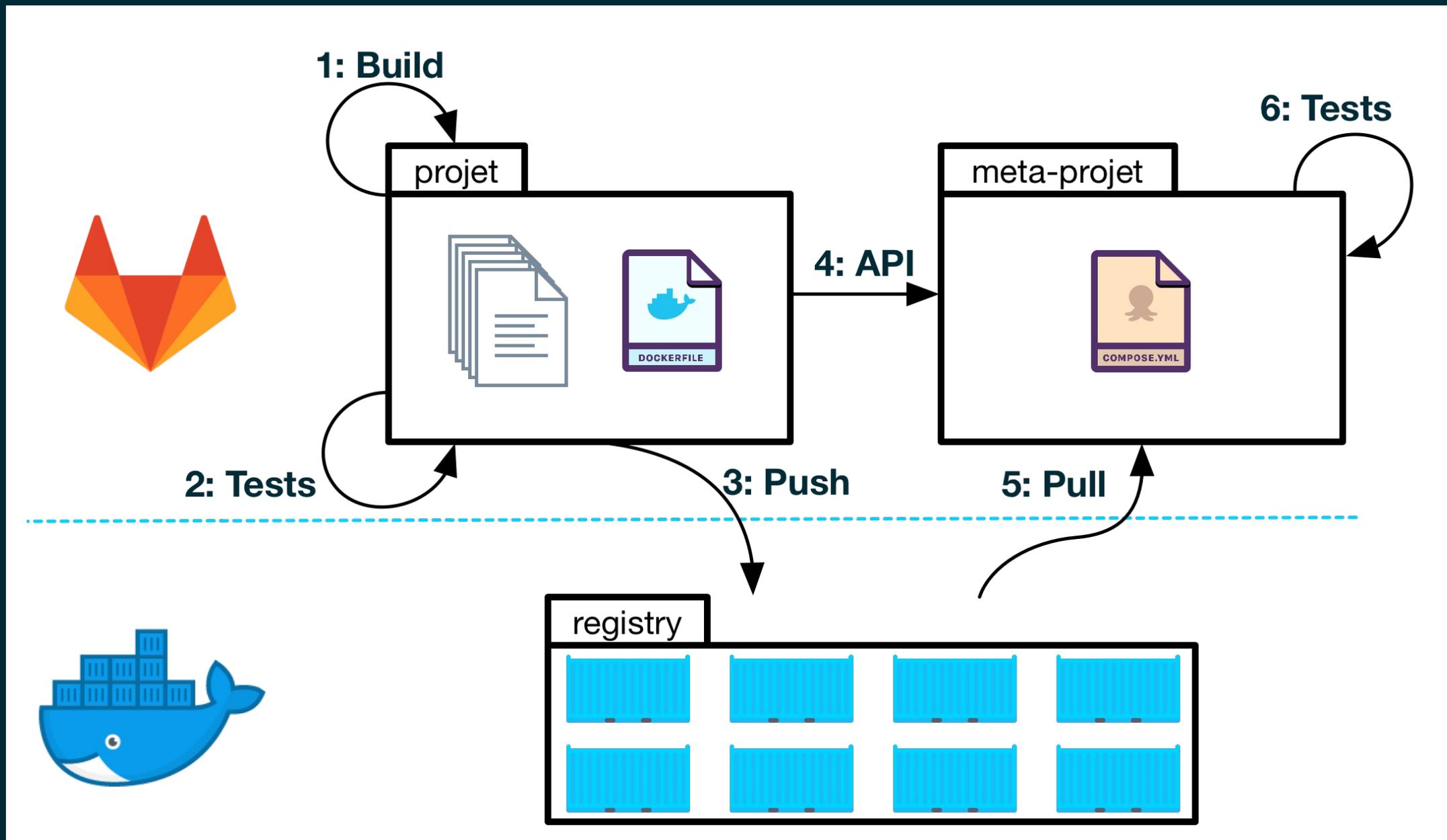
1 stack = 1 "meta"-projet pour les gouverner tous



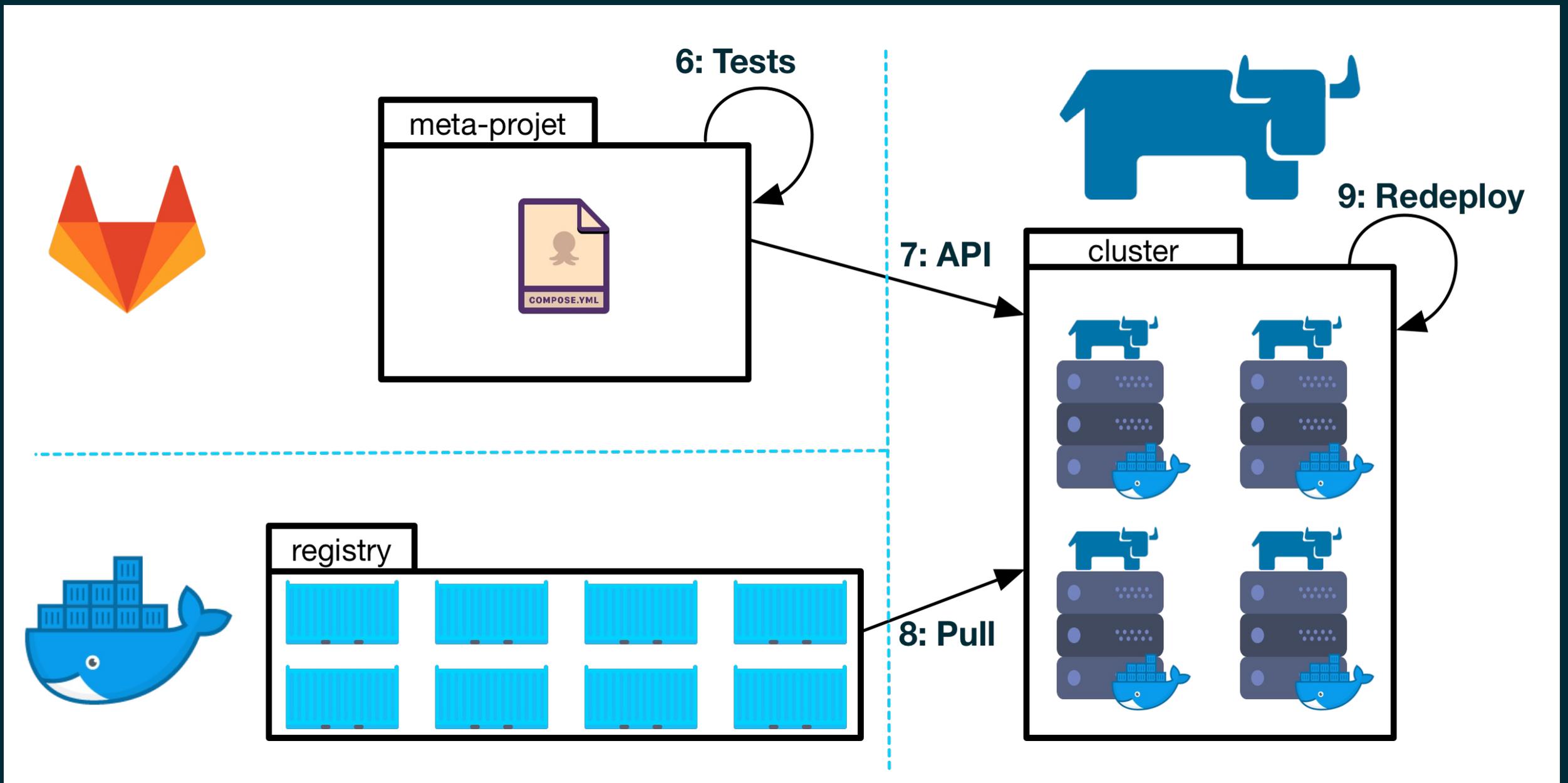
## 2 contextes d'intégration continue

- contexte projet
- contexte meta-projet

# Continuous Integration et Continuous Delivery



# Continuous Deployment



## AFFAIRE À SUIVRE

- Automatiser la création de "review stacks"
- Gérer les configurations dynamiquement
- Améliorer le feedback

## NOUVELLE RÉPARTITION DES TÂCHES

### Développeur

- Définition des tests
- Développement

### Outils d'intégration continue

- Compilation
- Tests
- Releases
- Déploiement
- Des tests d'intégration
- Du feedback

# NOTRE BILAN

- Automatisation des tâches répétitives
- Recentrage du développeur sur le développement
- Cycle de développement plus efficace
- Meilleure connaissance de l'état fonctionnel des services
- Qualité du code accrue
- Réduction du nombre d'erreurs humaines

- Courbe d'apprentissage
- Beaucoup (*trop ?*) de notifications
- Forme de dépendance à l'outil

# QUELQUES CONSEILS

- Donner de l'importance à la CI dès le début d'un projet
- Etre prêt à y consacrer les ressources nécessaires
- Savoir faire preuve de persévérance

Retour d'expérience

# INTÉGRATION CONTINUE DANS UN PROJET MICROSERVICES

*JDEV2017 - Frédéric Woelffel*

---

[frederic.woelffel@ihu-strasbourg.eu](mailto:frederic.woelffel@ihu-strasbourg.eu)



**JDEV2017**

