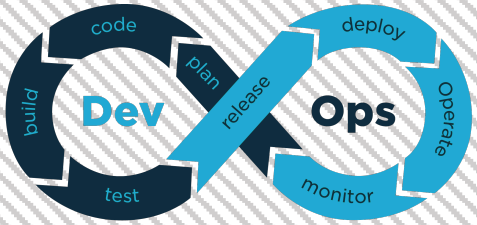


Mise en place d’une chaîne d’intégration continue DevOps

Sami AMOURA

Sommaire

- Présentation du projet
- Les outils utilisés
- Schéma de l’infrastructure
- Description du pipeline
- Questions

Présentation du projet

- Objectifs :

→ Mettre en place une chaine d'intégration continue complète

→ Déployer une application web python

Les outils

- EC2 : Elastic Compute Cloud
- CloudFormation: IaC
- IAM : Identity and Access Managment
- AWS CLI

Les outils

- Micro service et agilité
- Dockerfile
- GitLab et Jmeter

Les outils

- Orchestrateur
- Configuration Managment
- IaC

Les outils

GitLab

- Source Code Managment
- Container Registry Docker
- GitFlow Mutli-branches
- Conteneurisé

Les outils

- Outil de tests
- Tests fonctionnels et de charge
- Conteneurisé

Les outils

- Scanner d’image

GAUNTLT

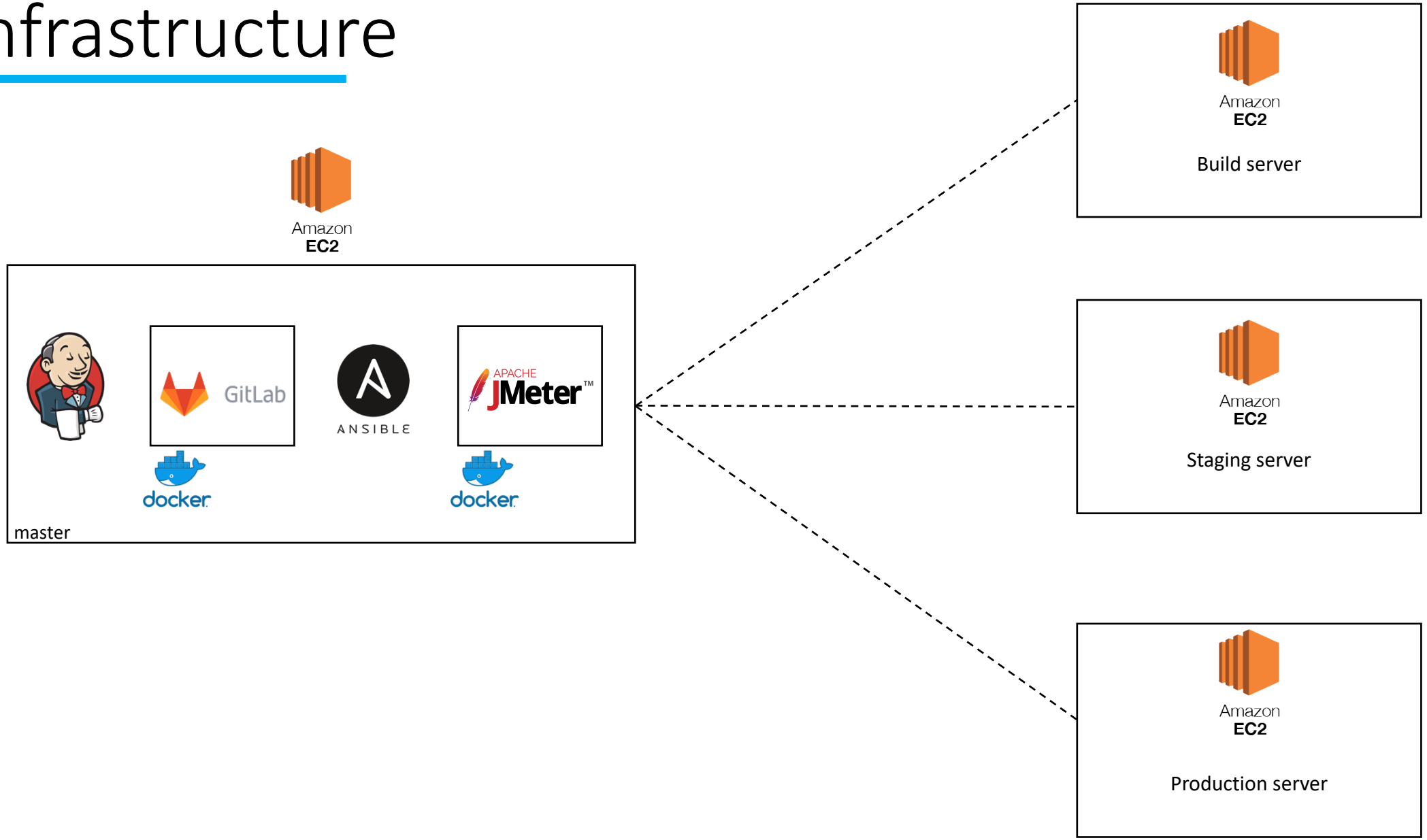
BE MEAN TO YOUR CODE AND LIKE IT

- Simulateur d’attaques (xss, curl…)
- Déceler les vulnérabilités à l’aide de scénarios

Les outils

- Ordonnanceur
- Nombreux plug-in
- Mise en place de pipeline : Jenkinsfile → script
- Shared-library

Infrastructure



Pipeline

- Continuous Integration (CI)
- Continuous Deployment (CD)

➢ Continuous Delivery (CD) : CI + CD

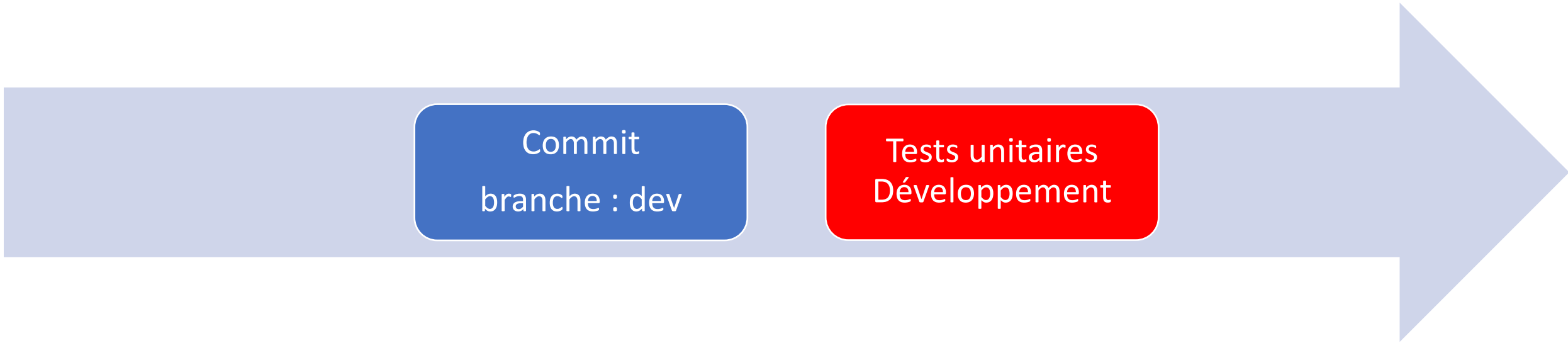
Pipeline : Continuous Integration (CI)

Les stages

L’utilisateur met à jour le code source de l’application ou celui de l’infrastructure et il commit les changements sur la branche de développement des features appelée « dev ». Les modifications sont alors envoyées sur le SCM GitLab.

Pipeline : Continuous Integration (CI)

Les stages



Le code source de l'application est testé (Dockerfile et Python) afin de s'assurer qu'il n'y a pas d'erreur

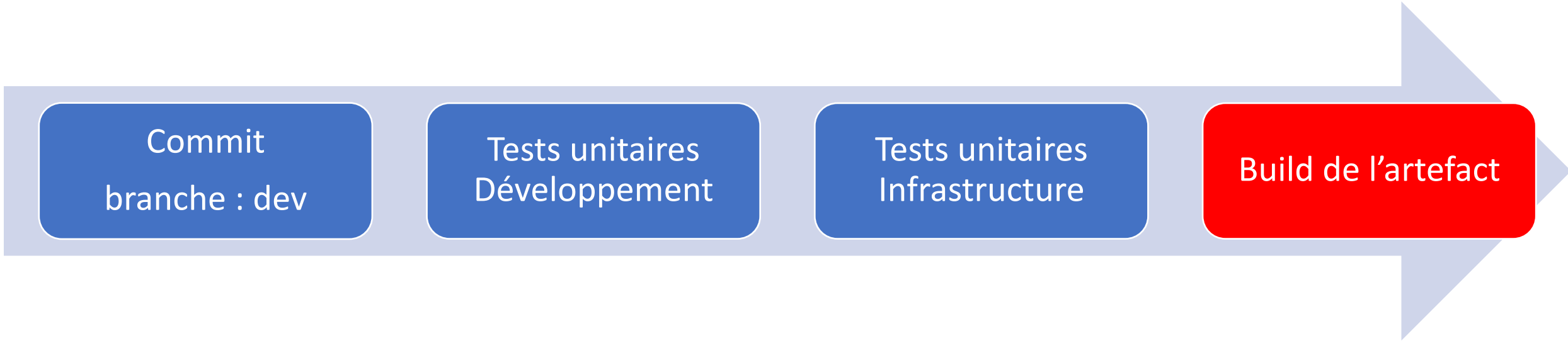
Pipeline : Continuous Integration (CI)

Les stages

Le code permettant de déployer l'infrastructure est testé (Bash, Yaml, Ansible) afin de s'assurer qu'il n'y a pas d'erreur

Pipeline : Continuous Integration (CI)

Les stages



Plusieurs artefacts (images Docker) sont buildés sur le serveur de build

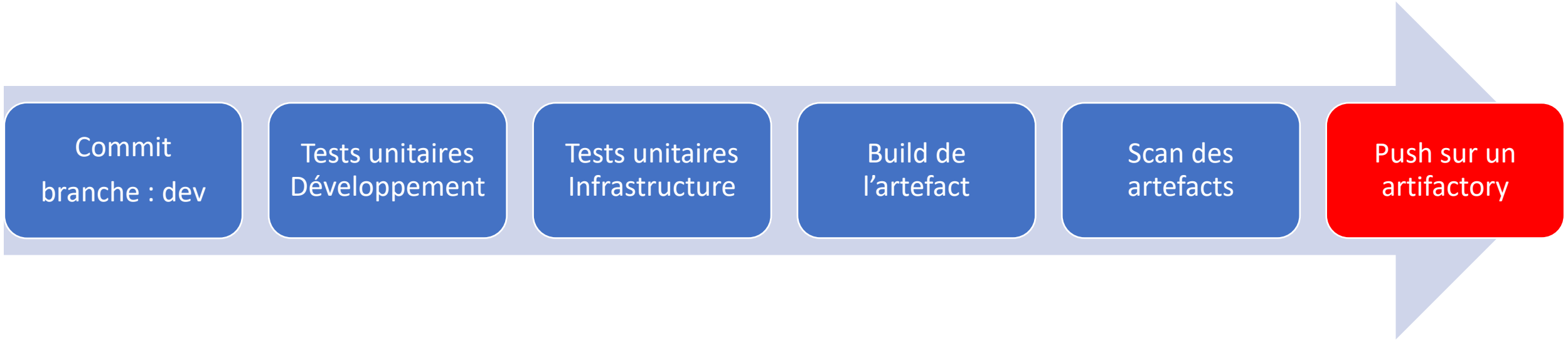
Pipeline : Continuous Integration (CI)

Les stages

Les deux images sont scannées afin de détecter les éventuelles failles de sécurité

Pipeline : Continuous Integration (CI)

Les stages



Les images docker sont poussées sur le container registry de GitLab

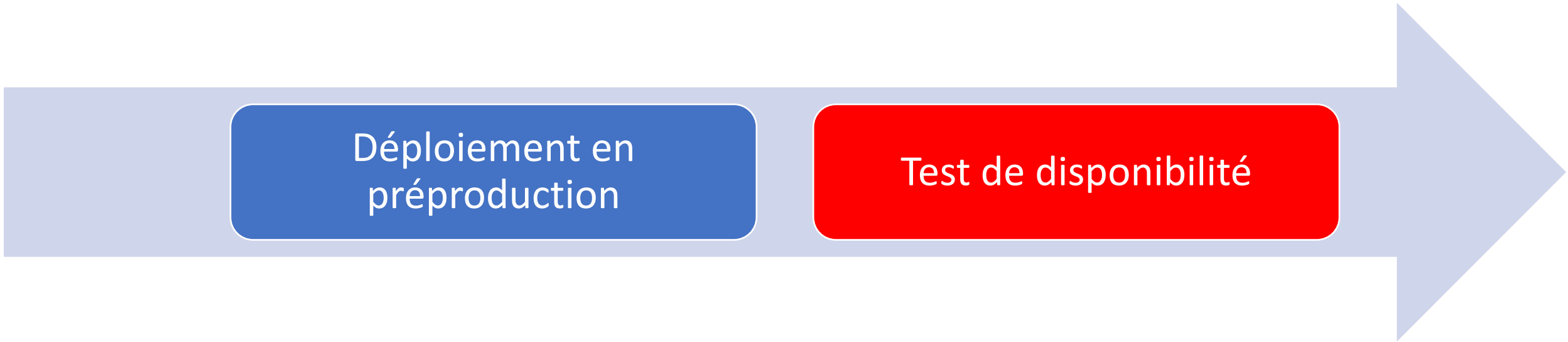
Pipeline : Continuous Deployment (CD)

Les stages

L’application est déployée dans des conteneurs docker sur le serveur de staging grâce aux deux images récupérées sur le container registry GitLab

Pipeline : Continuous Deployment (CD)

Les stages



L’application est testée afin de vérifier que celle-ci est bien disponible

Pipeline : Continuous Deployment (CD)

Les stages

Différents scenarios d’attaques sont exécutés sur notre applications afin de s’assurer que celle-ci est robuste

Pipeline : Continuous Deployment (CD)

Les stages

Des tests de fonctionnalités et de charges importantes sont effectués sur l'application afin d'analyser son comportement dans des cas extrêmes

Pipeline : Pull Ruquest

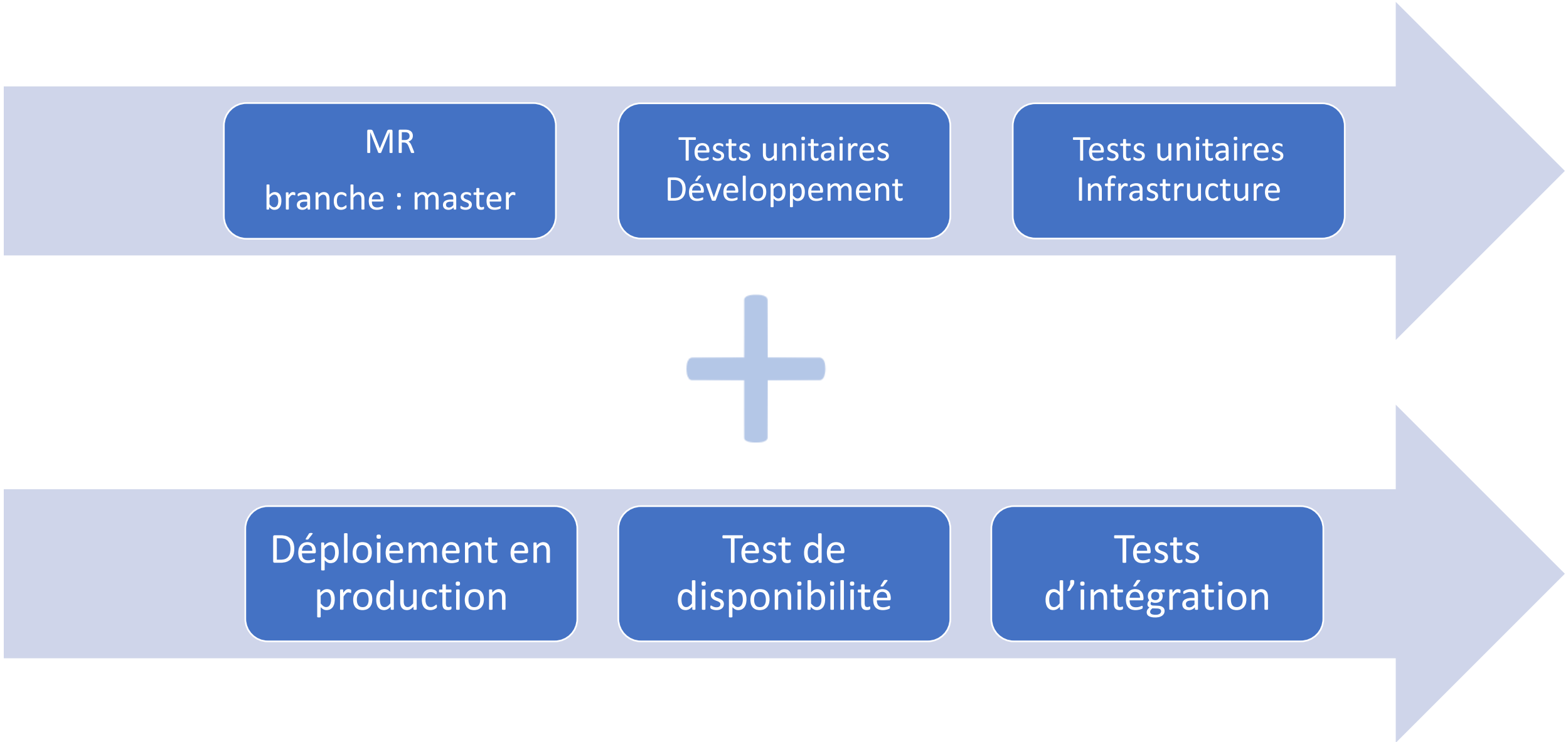
L’application est déployée en préprodcution et les tests ont été concluants pour le développeur ou le sysadmin

Il décide alors de soumettre les résultats à son techlead en effectuant une Pull Request (PR) afin de déployer l’application en production

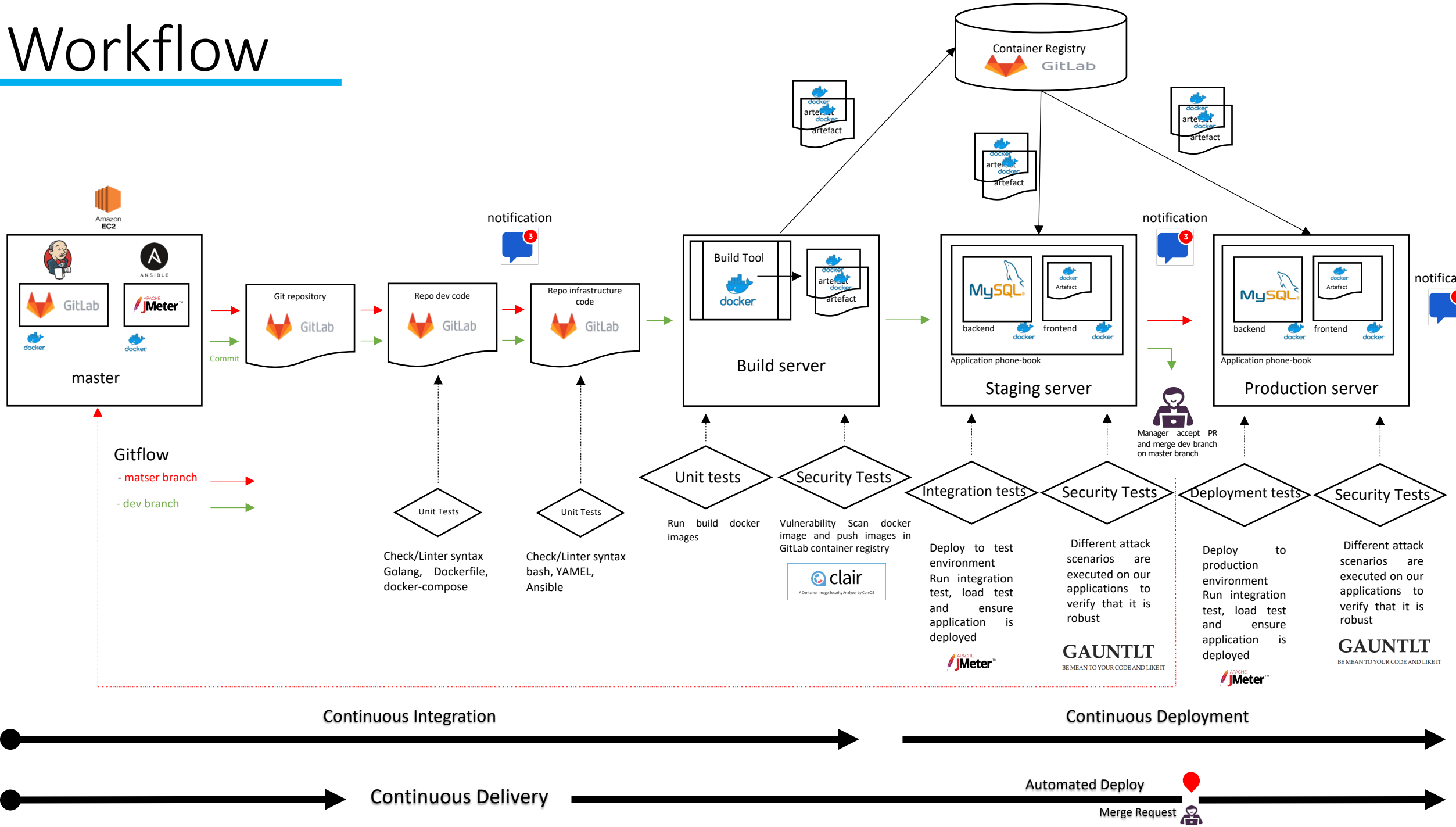
La Techlead décide alors de merger la branche dev sur la branche de production master

→ Le pipeline est une nouvelle fois exécuté

Pipeline : Continuous Deployment : CI + CD



Workflow


Container Registry
GitLab
artefact
Amazon EC2
GitLab
JMeter
docker
master
Commit
Git repository
Repo dev code
Repo infrastructure code
notification
Build Tool
Build server
MySQL
backend
frontend
Application phone-book
Staging server
Production server
Manager accept PR and merge dev branch on master branch
Gitflow
- matser branch
- dev branch
Unit Tests
Check/Linter syntax Golang, Dockerfile, docker-compose
Check/Linter syntax bash, YAMEL, Ansible
Unit tests
Run build docker images
Security Tests
Vulnerability Scan docker image and push images in GitLab container registry
clair
Integration tests
Deploy to test environment
Run integration test, load test and ensure application is deployed
Different attack scenarios are executed on our applications to verify that it is robust
GAUNTLT
BE MEAN TO YOUR CODE AND LIKE IT
Deployment tests
Deploy to production environment
Run integration test, load test and ensure application is deployed
Continuous Integration
Continuous Deployment
Continuous Delivery
Automated Deploy
Merge Request

Questions

Merci pour votre attention