



devoteam

AI-driven tech consulting



Tribe Azure - Knowledge Sharing - Décembre 2025

# Du code à la prod en 5 minutes



Thibault Gibard  
[thibault.gibard@devoteam.com](mailto:thibault.gibard@devoteam.com)

AI-driven tech consulting



# Le problème ?

- Déploiement manuel = lent et risqué
- Configuration différente entre environnements
- Pas de traçabilité des changements
- Processus non reproductible

**"Ça marche sur ma machine"**

...mais pas en production



## Objectif

Push du code,

c'est en ligne automatiquement

# La solution : CI/CD



## CONTINUOUS INTEGRATION

### CI

Build et test automatiques à chaque commit. Détecte les erreurs immédiatement.

## CONTINUOUS DEPLOYMENT

### CD

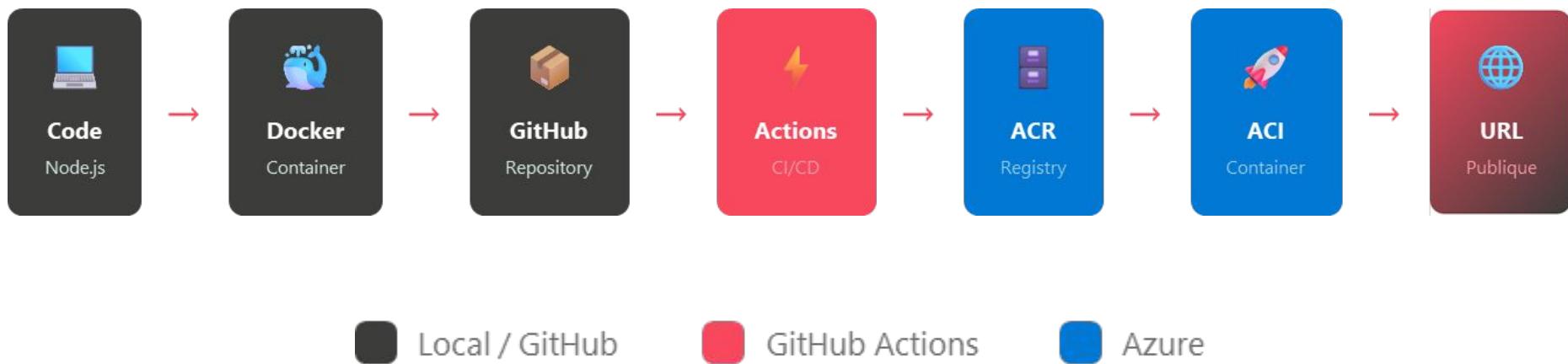
Déploie automatiquement en production. Pas d'intervention manuelle.

## L'ORCHESTRATEUR

### GitHub Actions

Workflows YAML déclenchés sur événements Git. Intégré à GitHub.

# L'architecture qu'on va construire



# Les briques Azure



## ACR

Container Registry

Stocke tes images Docker dans Azure. C'est notre "Docker Hub privé".

 acrdevoteamdemo.azurecr.io



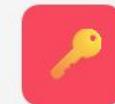
## ACI

Container Instances

Exécute notre container. Serverless, pas d'infra à gérer.



Paiement à l'usage (CPU + RAM)



## Service Principal

Authentification

Permet à GitHub de s'authentifier à Azure.



4 secrets dans GitHub

# GitHub Actions



Workflow YAML déclenché automatiquement



## 1 Trigger

Push sur main ou workflow\_dispatch

## 2 Build

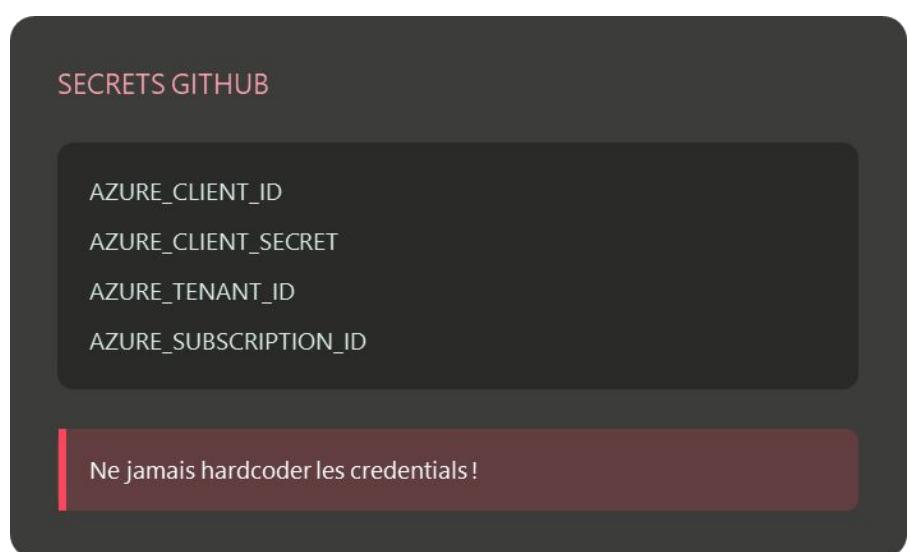
az acr build crée l'image Docker

## 3 Push

Image vers ACR

## 4 Deploy

az container create vers ACI



# GitHub Copilot (pour le fun)



## Un bon prompt = workflow généré en 10 secondes

L'IA comme accélérateur, pas comme remplacement

Pour que notre prompt soit parfait :

- Être spécifique sur les outils (azure/login@v2)
- Préciser le type d'auth (Service Principal, pas OIDC)
- Donner les noms exacts des ressources
- Spécifier les paramètres requis (--os-type, --cpu)

// Prompt optimisé

Crée un workflow GitHub Actions :

Trigger : push sur main + workflow\_dispatch

Steps :

1. Checkout code
2. Azure login avec creds JSON
3. az acr login
4. az acr build
5. az container create avec  
--os-type Linux  
--cpu 1 --memory 1

Variables : ACR, Image, RG, ACI, DNS, Port

# Démonstration



- Démo live
- Je modifie une ligne, je push, on regarde

# Conclusion

Ce que nous avons mis en place 

-  Site Node.js containerisé
-  Workflow CI/CD complet
-  Déploiement Azure automatisé
-  Copilot pour générer le workflow

## À retenir

### Reproductible

Même process à chaque déploiement

### Versionné

Workflow dans le repo Git

### Auditable

Historique complet dans Actions



Thibault Gibard  
[thibault.gibard@devoteam.com](mailto:thibault.gibard@devoteam.com)



devoteam

# Thank you!