

Le "Cloud-native" est-il le nouveau DevOps ?

Benjamin Brial & Chamseddine Saadoune

18 Nov 2021

Cloud-native, kezaiko ?



Cloud-native = ?

À votre avis, quel est le **mode de transport le plus optimisé** sur autoroute ?



Cloud-native = ?

Une charrette ? Plutôt optimisée pour fonctionner sur un chemin de terre ...



Cloud-native = ?

Une Ford Model T ? On ira plus vite (jusqu'à 70 km/h !), mais gérer la direction et ralentir sera plus compliqué ...



Cloud-native = ?

Sans forcément se payer une Tesla,
une voiture moderne est bien
plus adaptée !



Cloud-native = ?



Aujourd'hui, l'autoroute = le cloud

Les applications cloud-native sont conçues pour pleinement utiliser les capacités fournies par le cloud

← Seulement dans vos rêves ...

Cloud-native = ?



**CLOUD NATIVE
COMPUTING FOUNDATION**

Les technologies *Cloud Native* permettent aux entreprises de construire et d'exploiter des applications élastiques dans des environnements modernes et dynamiques comme des clouds publics, privés ou bien hybrides. Les conteneurs, le maillage de services, les microservices, les infrastructures immuables et les API déclaratives illustrent cette approche.

La *Cloud Native Computing Foundation* cherche à favoriser l'adoption de ce paradigme en encourageant et en soutenant un écosystème de projets *open source* et indépendants. Nous démocratisons l'état de l'art des bonnes pratiques afin de rendre l'innovation accessible à tous.

Cloud-native, kezeko ?

Cloud-native \neq Cloud

Cloud-native ≠ Cloud computing

Comment
s'exécutent les
apps'

Infrastructures et
services à la
demande

Automatisation,
Observabilité,
Agilité

Où s'exécutent les
apps'

Résilience

Elasticité,
mutualisation,
accessibilité

Microservices

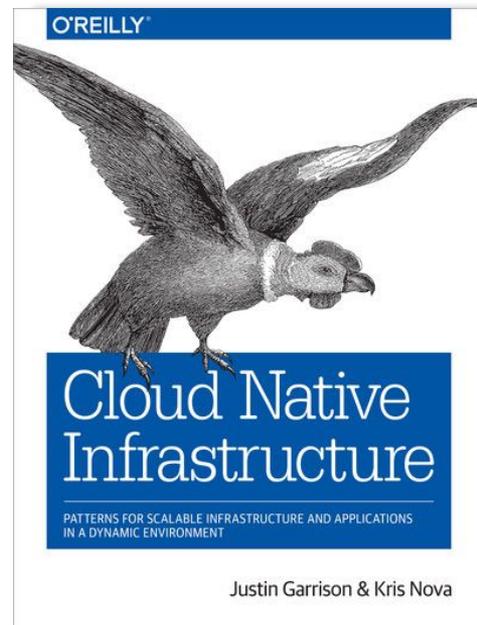
Public, Privé,
Hybride, Multi

Approche,
philosophie, culture

Cloud-native ≠ Cloud Public

Une infrastructure Cloud-native ?

“Une infrastructure Cloud-native va au-delà des serveurs, du réseau et du stockage dans le cloud. Il s’agit autant d’hygiène opérationnelle que d’élasticité et d’évolutivité.”



*Cloud Native Infrastructure
Justin Garrison & Kris Nova*

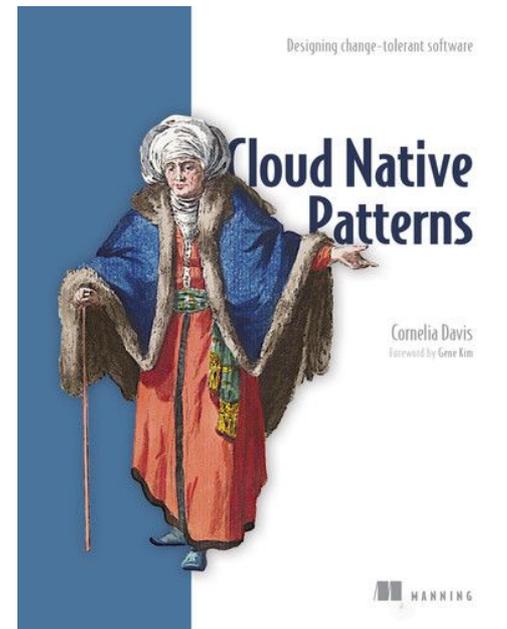
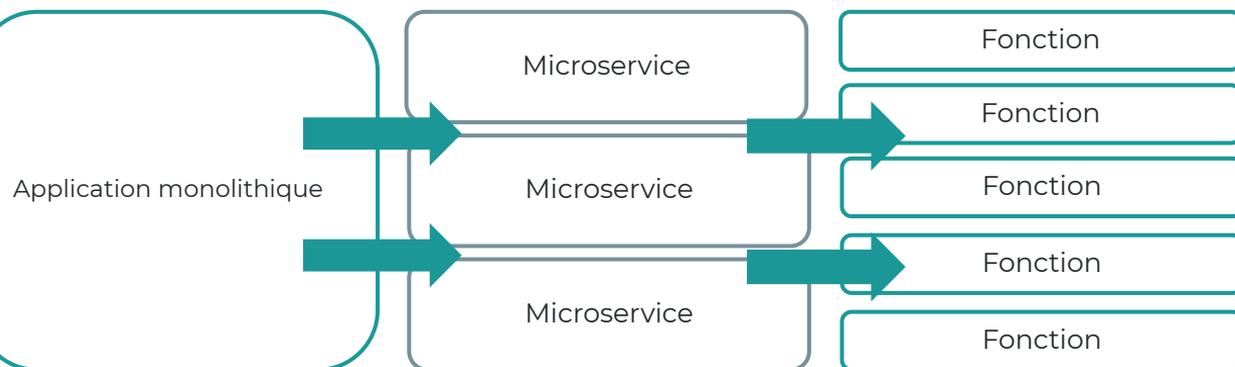
Cloud-native, ke zako ?

Cloud-native \neq microservices

Cloud-native ≠ microservices

Une application Cloud-native ?

“Les applications Cloud-native sont hautement distribués, doivent fonctionner dans un environnement en constante évolution et sont elles-mêmes en constante évolution.”



Cloud Native Patterns
Cornelia Davis

Cloud-native, kezako ?

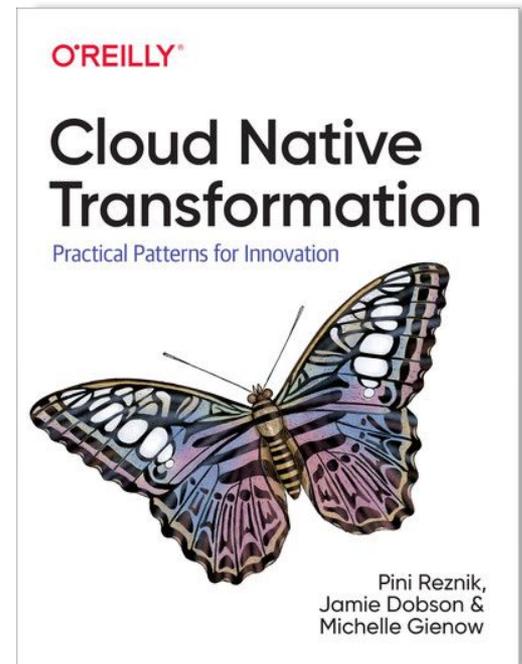
Cloud-native  ***patterns***

Cloud-native patterns

Une entreprise Cloud-native ?

“Les organisations Cloud-native sont conçues pour tirer le meilleur parti du fonctionnement des technologies cloud; le cloud va bien sûr continuer à évoluer et sera très différent à l’avenir, mais nous “buildons” également pour anticiper cela.”

<https://www.cnpatterns.org/>



*Cloud Native Transformation
Pini Reznik, Jamie Dobson,
Michelle Gienow*

Une nouvelle définition ?



**CLOUD NATIVE
COMPUTING FOUNDATION**

Les technologies *Cloud Native* permettent aux entreprises de construire et d'exploiter des applications élastiques dans des environnements modernes et dynamiques comme des clouds publics, privés ou bien hybrides. Les conteneurs, le maillage de services, les microservices, les infrastructures immuables et les API déclaratives illustrent cette approche.

cycloid

Le terme *Cloud-Native* désigne l'adoption à l'échelle d'une organisation, des meilleures pratiques, modèles et usages modernes du cloud. Autant d'un point de vue infrastructure que dans la conception, le déploiement et l'exploitation de ses applications.

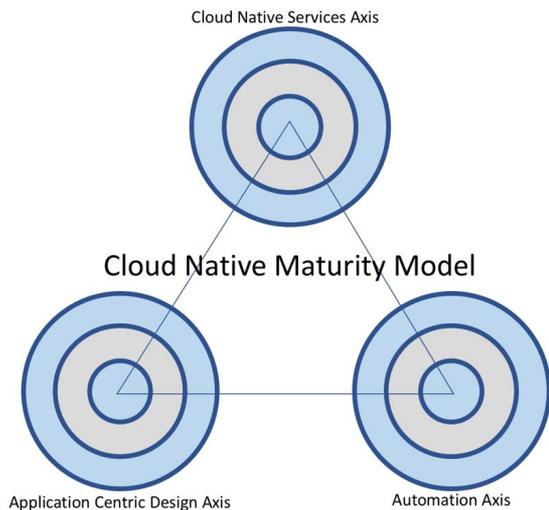


Adopter une approche Cloud-native, comment s'y prendre ?



Définir honnêtement sa maturité

Cloud Native Maturity Model



Cloud Native Maturity Matrix

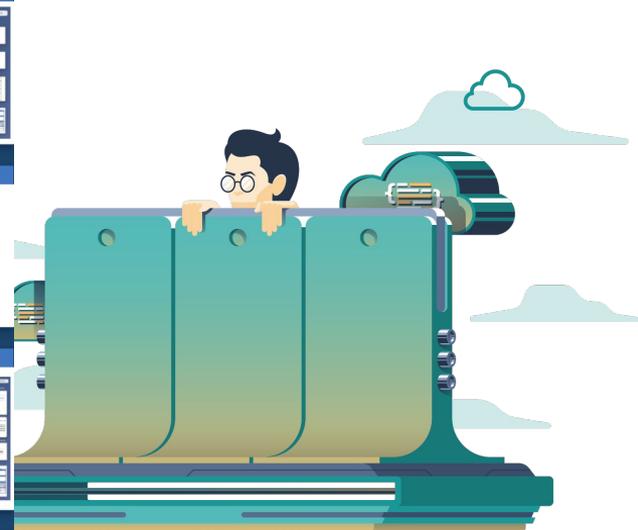
Stage	NO PROCESS	WATERFALL	AGILE	CLOUD NATIVE	NEXT
CULTURE	Individualist ○	Predictive ○	Iterative ○	Collaborative ○	Experimental ○
PROD/SERVICE DESIGN	Arbitrary ○	Long-term plan ○	Feature driven ○	Data driven ○	All driven ○
TEAM	No organization, single contributor ○	Hierarchy ○	Cross-functional teams ○	DevOps / SRE ○	Internal supply chains ○
PROCESS	Random ○	Waterfall ○	Agile (Scrum/Kanban) ○	Design Thinking + Agile + Lean ○	Distributed, self-organized ○
ARCHITECTURE	Emerging from trial and error ○	Tightly coupled monolith ○	Client server ○	Microservices ○	Functions ○
MAINTENANCE	Respond to users complaints ○	Ad-hoc monitoring ○	Alerting ○	Full observability & self-healing ○	Preventive ML, AI ○
DELIVERY	Irregular releases ○	Periodic releases ○	Continuous Integration ○	Continuous Delivery ○	Continuous Deployment ○
PROVISIONING	Manual ○	Scripted ○	Config. management (Puppet/Chef/Ansible) ○	Orchestration (Kubernetes) ○	Serverless ○
INFRASTRUCTURE	Single server ○	Multiple servers ○	VMs (pets) ○	Containers/ hybrid cloud (cattle) ○	Edge computing ○

CURRENT SITUATION
GOAL

Accepter les difficultés potentielles et y faire face

The image displays a comprehensive grid of logos for various cloud native technologies and services, organized into several categories:

- Database:** KV, V, etc.
- Streaming & Messaging:** cloudstream, etc.
- Application Definition & Image Build:** HELM, etc.
- Continuous Integration & Delivery:** flux, etc.
- Platform:** Certified Kubernetes - Distribution, Hosted, Installer, PaaS/Container Service.
- Serverless:** etc.
- Scheduling & Orchestration:** Kubernetes, etc.
- Coordination & Service Discovery:** etcd, etc.
- Remote Procedure Call:** gRPC, etc.
- Service Proxy:** envoy, etc.
- API Gateway:** etc.
- Service Mesh:** etc.
- Cloud Native Storage:** etc.
- Container Runtime:** cri-o, etc.
- Cloud Native Network:** cilium, etc.
- Automation & Configuration:** etc.
- Container Registry:** etc.
- Security & Compliance:** falco, etc.
- Key Management:** etc.
- Observability and Analysis:** Monitoring, Logging, Tracing, Chaos Engineering.
- CD Foundation Landscape:** etc.
- Members:** etc.
- Cloud Native Landscape:** etc.



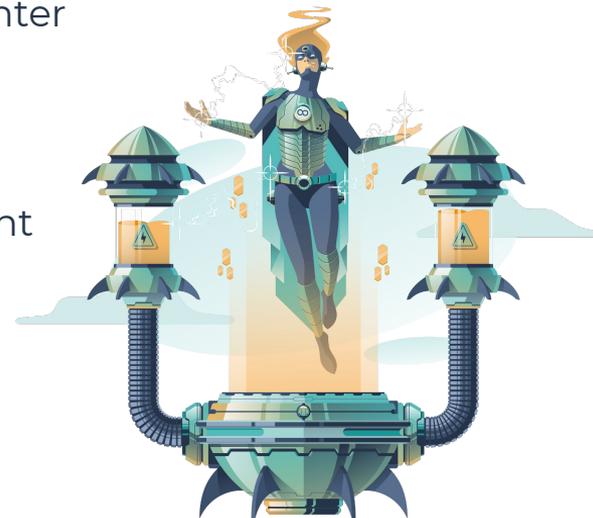
CLOUD NATIVE Landscape

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application with CNCF Projects representing a particular well-trodden path.

[cncf.io](https://www.cncf.io)

Engager les premières actions

- ❖ Choisissez un Cloud provider interne ou externe pour expérimenter
- ❖ Pensez au Cloud hybride/multi-cloud de facto
- ❖ Commencez et maîtrisez votre automatisation avec une approche GitOps étape par étape: Infra as code / CI/CD / Conf management
- ❖ Concevez et déployez vos nouvelles applications Cloud Native sans over engineer et ne complexez pas sur votre legacy
- ❖ Pensez au DevX, expérience développeur et sortez de la chimère NoOps, all DevOps
- ❖ Pensez à vos DevOps et à l'industrialisation y compris dans leur équipe: nécessité de leur permettre de se concentrer sur l'évangélisation et l'automatisation et non les 50% de tâches répétitives
- ❖ Soyez dans une volonté de responsabiliser les équipes et participer à la désilotisation des organisations au fur et à mesure: aucun silos est éternel.



Bénéficiaire du DevOps pour passer au Cloud-Native

Avant

Configuration et état d'esprit 100% "on-prem", pas de ressources cloud, pas de microservices ...

Déploiement manuels, sans tests ...

Très peu de visibilité et de contrôle sur l'infrastructure ...

Après

Application des bonnes pratiques IaC rendant les processus explicites et reproductibles, facilitant chaque déploiement

Passage d'une architecture monolithique (en "on-prem") à des microservices (dans le cloud)

Confiance partagée par les équipes afin de poursuivre leurs cheminements vers une transformation numérique dans le cloud



Tranquillité d'esprit



5 heures/semaine économisées sur les déploiements



Confiance pour mettre en oeuvre une migration et une utilisation plus large du cloud

Migration simultanée vers le DevOps, l'laC et le Cloud

Avant

Le déploiement manuel de nouveaux projets prends trop de temps

Aucune logique d'*infrastructure as code*, gestion de configuration

Un système complexe d'outils open-source et propriétaires

Aucune vision globale pour assurer une certaine efficacité et une gouvernance

Après

Utilisation d'une plateforme agnostique permettant une interopérabilité entre les outils et environnements

Les déploiements prennent quelques heures plutôt que quelques jours

Mise en place de bonnes pratiques IaC et des fondations pour une migration réussie vers AWS et Flexible Engine



Industrialisation, automatisation et bonnes pratiques



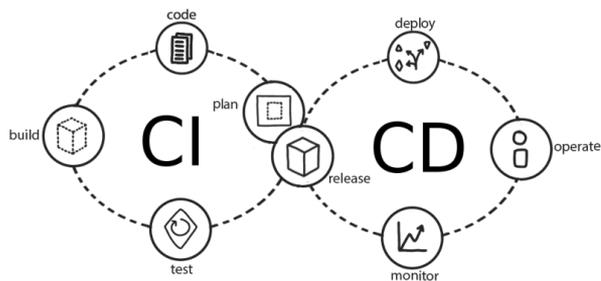
Clients plus efficace grâce au déploiement de stacks reproductible



Petites améliorations continues menant à une meilleure vitesse de déploiement

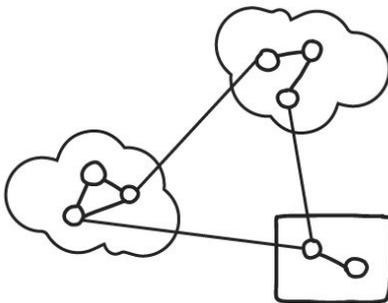
Une approche Cloud-native chez Cycloid ?

CI/CD en tant que plateforme



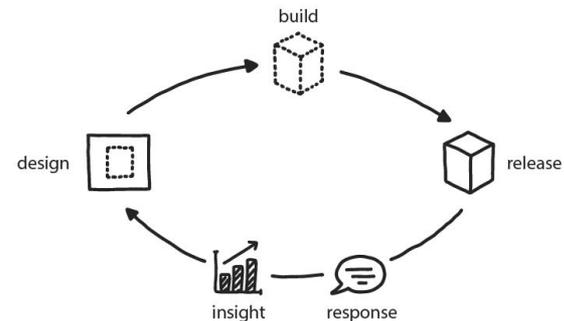
Source :
<https://www.cnpatterns.org/operations/cicd-as-platform>

Systemes distribués



Source :
<https://www.cnpatterns.org/development-design/distributed-systems>

Boucle d'apprentissage



Source :
<https://www.cnpatterns.org/strategy-risk-reduction/learning-loop>

Et le DevOps ?



Le DevOps est philosophie centrée sur l'adoption de cultures, pratiques et outils destinés à fournir continuellement de la valeur ajoutée au travers de l'union des personnes, processus et technologies liés aux mondes du développement (Dev) et des opérations (Ops).

