



INFRASTRUCTURE AS A SERVICE

GOOGLE CLOUD PLATFORM

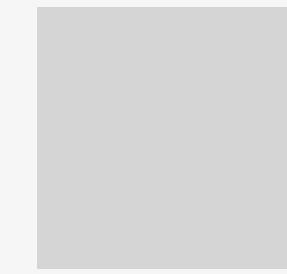
Brad Niepceron - niepceron.brad@gmail.com

**TEMPS**

- 30 heures.
Pause pendant TP.

**OBJECTIFS**

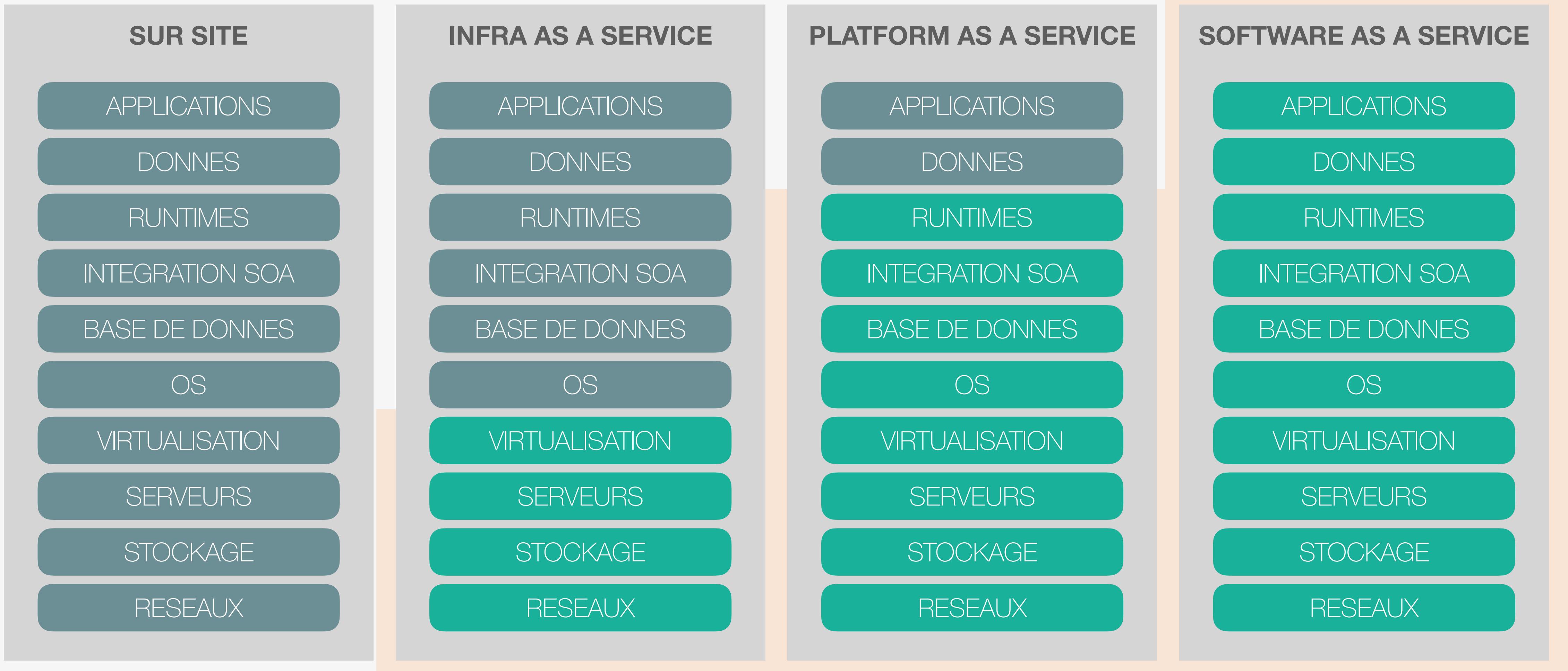
- Automatisation de déploiement d'applications
- Commencer le projet de fin d'année.

**NOTATION**

- 4 TPs + 1 Projet
- 12 points pour les TPs
- 8 points pour le projet

RAPPEL CLOUD COMPUTING

Dans le cloud



INTRODUCTION À GOOGLE CLOUD PLATFORM

03

INTRODUCTION À GCP

Services

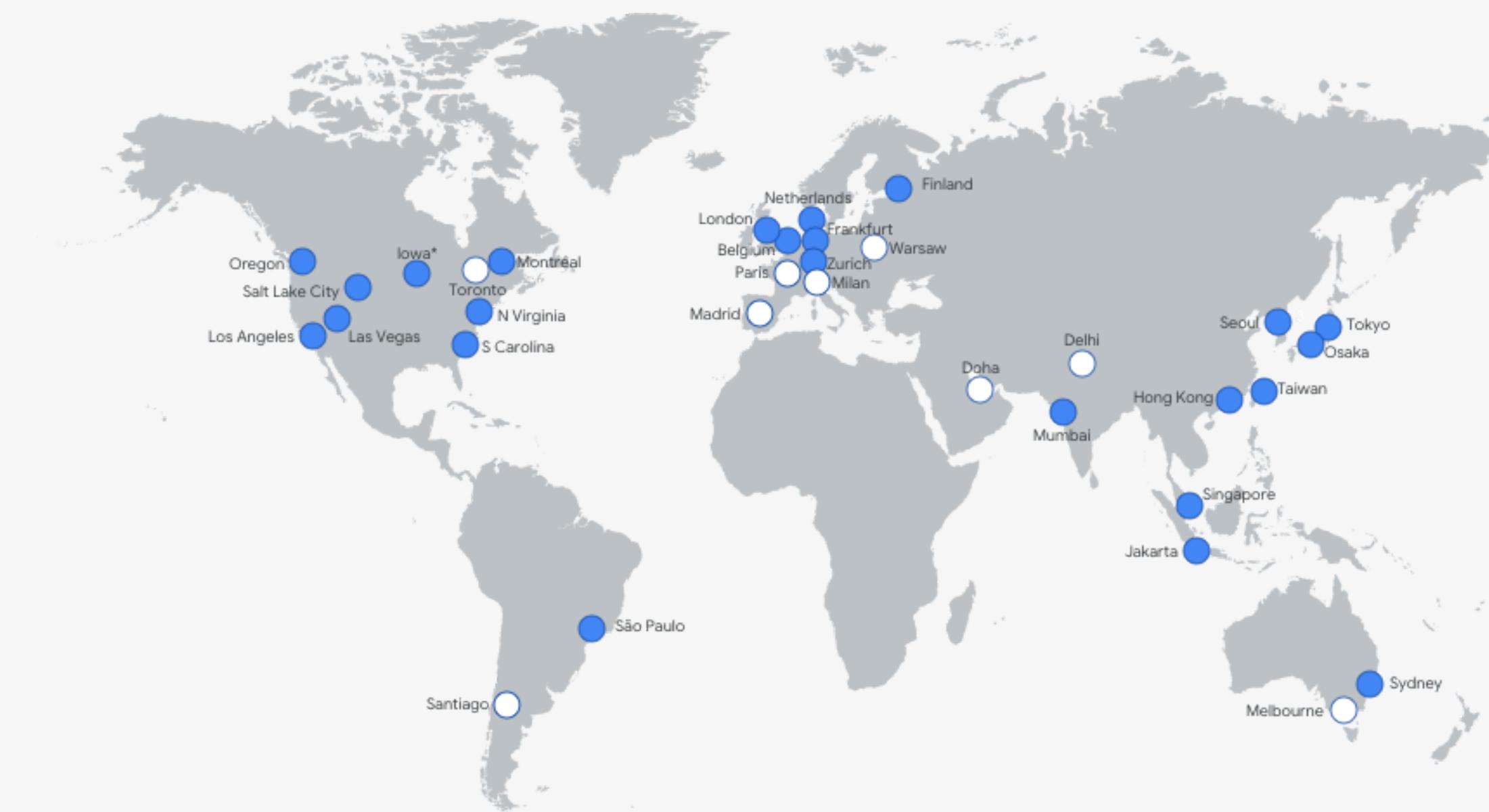


CLIENTS : Twitter, Paypal, HSBC, Ebay, Spotify ...

03

INTRODUCTION À GCP

Zones Cloud



DISPONIBLES DANS

24



RÉGIONS

73



ZONES

144



EMPLACEMENTS DU RÉSEAU PÉRIPHÉRIQUE

PLUS DE 200

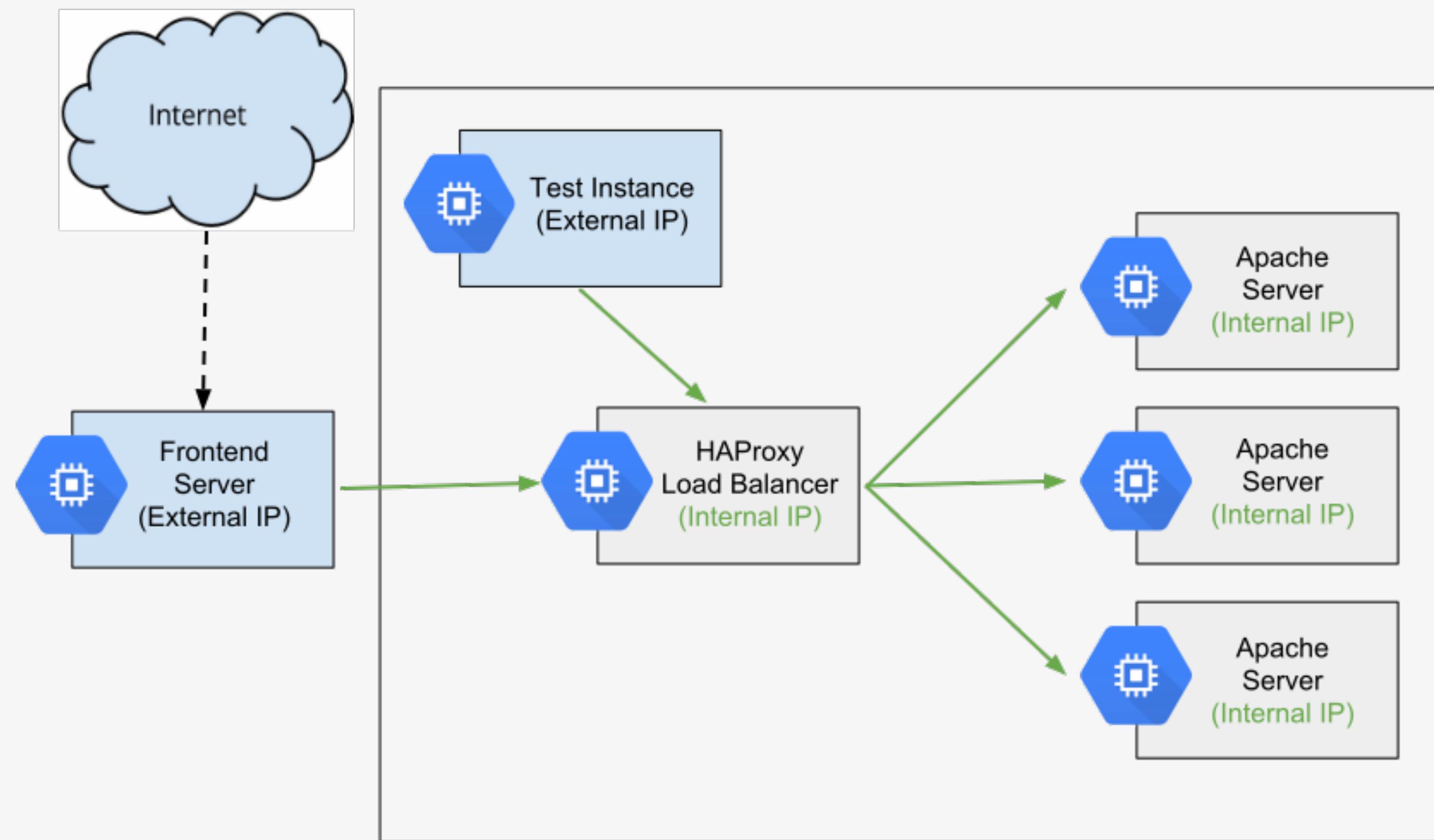


PAYS ET TERRITOIRES

BIENTÔT DISPONIBLE ! Google Cloud va continuer à se développer dans les régions suivantes : Delhi (Inde), Doha (Qatar), Madrid (Espagne), Melbourne (Australie), Milan (Italie), Paris (France), Santiago (Chili), Toronto (Canada) et Varsovie (Pologne).

INTRODUCTION À GCP

Google Compute Engine



INTRODUCTION À GCP

Google Compute Engine - Prix

Type de machine	Processseurs virtuels	Mémoire	Prix (USD)	Prix des VM préemptives (USD)
n1-standard-1	1	3.75GB	\$0.0523	\$0.0110
n1-standard-2	2	7.5GB	\$0.1046	\$0.0220
n1-standard-4	4	15GB	\$0.2092	\$0.0440
n1-standard-8	8	30GB	\$0.4184	\$0.0880
n1-standard-16	16	60GB	\$0.8368	\$0.1760

INTRODUCTION À GCP

Google Compute Engine - Prix

Modèle	États-Unis	Europe/Asie	États-Unis	Europe/Asie	États-Unis	Europe/Asie
	Prix à la demande	Prix à la demande	Remise automatique proportionnelle à une utilisation soutenue	Remise automatique proportionnelle à une utilisation soutenue	Prix des GPU préemptifs	Prix des GPU préemptifs
NVIDIA K80	0,45 \$	0,49 \$	À partir de 0,315 \$	(jusqu'à - 30 %)	À partir de 0,346 \$	(jusqu'à - 30 %)
NVIDIA P4	À partir de 0,60 \$	À partir de 0,65 \$	À partir de 0,42 \$	À partir de 0,455 \$	0,216 \$	0,216 \$
NVIDIA P100	1,46 \$	1,60 \$	À partir de 1,022 \$	À partir de 1,242 \$	0,43 \$	0,43 \$
NVIDIA V100	2,48 \$	2,55 \$	À partir de 1,736 \$	À partir de 1,785 \$	0,74 \$	0,74 \$

03

INTRODUCTION À GCP

Google Compute Engine - Images

Système d'exploitation	Canal d'assistance	Famille de l'image	Projet de l'image	Commentaires
CentOS	Compute Engine	centos-7	gce-uefi-images	
Container-Optimized OS de Google	Compute Engine	cos-69-lts cos-stable cos-beta cos-dev	gce-uefi-images	
Red Hat Enterprise Linux (RHEL)	Compute Engine	rhel-7	gce-uefi-images	Image payante
Ubuntu	Compute Engine	ubuntu-1804-lts	gce-uefi-images	
Windows Server	Compute Engine	windows-2019 windows-2019-core windows-1809-core windows-1803-core windows-2016 windows-2016-core windows-2012-r2 windows-2012-r2-core	gce-uefi-images	Image payante

01

Images d'OS pré-configurées

02

Équivalent des AMI chez AWS

03

Peuvent servir de base aux déploiements de services.

INTRODUCTION À GCP

Google Compute Engine - TPU



Machines très couteuses dédiées à l'apprentissage profond.

03

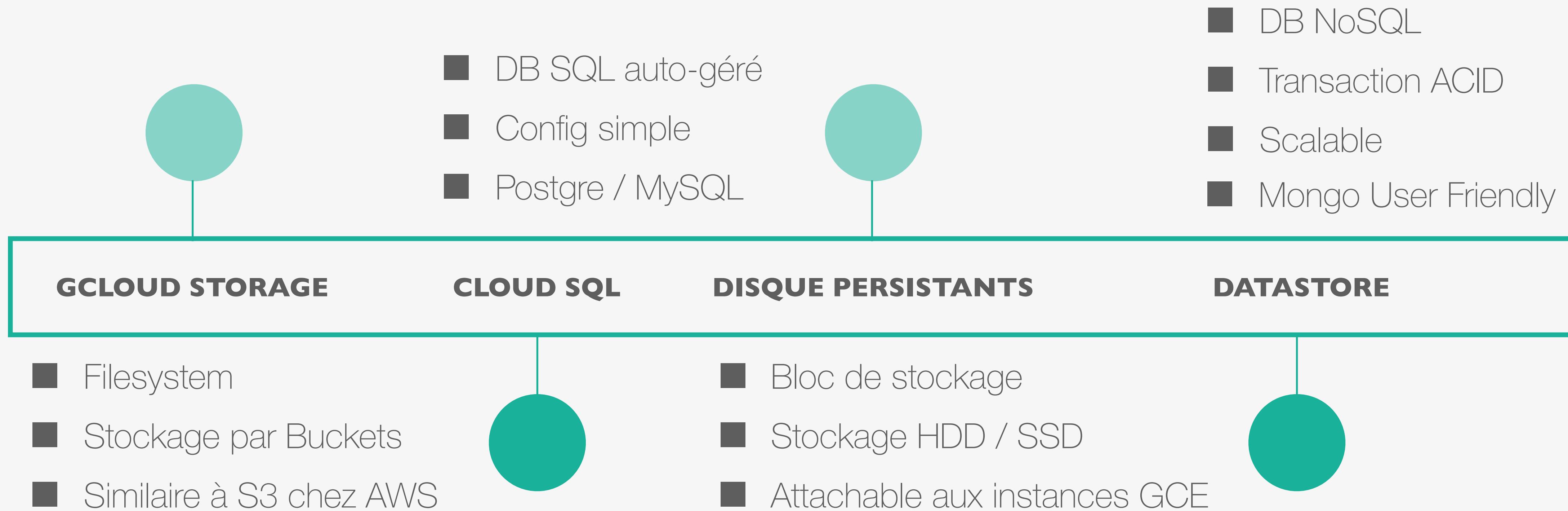
INTRODUCTION À GCP

Google Compute Engine - TPU

US	EUROPE	ASIE-PACIFIQUE
Cloud TPU v2		4,95 USD par TPU et par heure
TPU v2 préemptif		1,485 USD par TPU et par heure

INTRODUCTION À GCP

Google Compute Engine - Storage



03

INTRODUCTION À GCP

Google Compute Engine - SDK

PETITE DÉMO

INTRODUCTION À GCP

Google Compute Engine - TP

- 01** Créer un projet GCP
- 02** Créer une instance à l'aide d'un template (f1 -micro)
- 03** Créer une image de l'instance
- 04** Créer un template de cette image
- 05** Créer un groupe d'instance
- 06** Créer une instance utilisant le template
- 07** Lancer un service simple (nginx, nodejs ...)
- 08** Le rendre disponible (IP/Pare-feu/Port?)
- 09** Terminer et supprimer

BONUS - Utiliser le SDK seulement ... ;-)

INTRODUCTION À GCP

Google App Engine



- Développement et déploiement d'app.
- Payant au delà d'un nombre de ressources.
- Similaire à Amazon Elastic Compute Cloud.
- Scaling automatique.

INTRODUCTION À GCP

Google App Engine

```
runtime: nodejs
env: flex

manual_scaling:
  instances: 1
resources:
  cpu: 1
  memory_gb: 0.5
  disk_size_gb: 10
```

Fichier **app.yaml**

Commande de déploiement : **gcloud app deploy**

INTRODUCTION À GCP

Google App Engine - TP

- 01** Déployez une app sur AppEngine
- 02** Développer un CRUD (framework et DB au choix)

ARCHITECTURE SCALABLE

04

ARCHITECTURE SCALABLE

Montée en charge



RAISONS ?

- Requête DB lourde
- Plusieurs connexions en simultané
- API tier qui lag

ARCHITECTURE SCALABLE

Montée en charge



SOLUTIONS ?

- Augmenter la puissance du serveur
- Changer certaines briques
- NoSQL over SQL

04

ARCHITECTURE SCALABLE

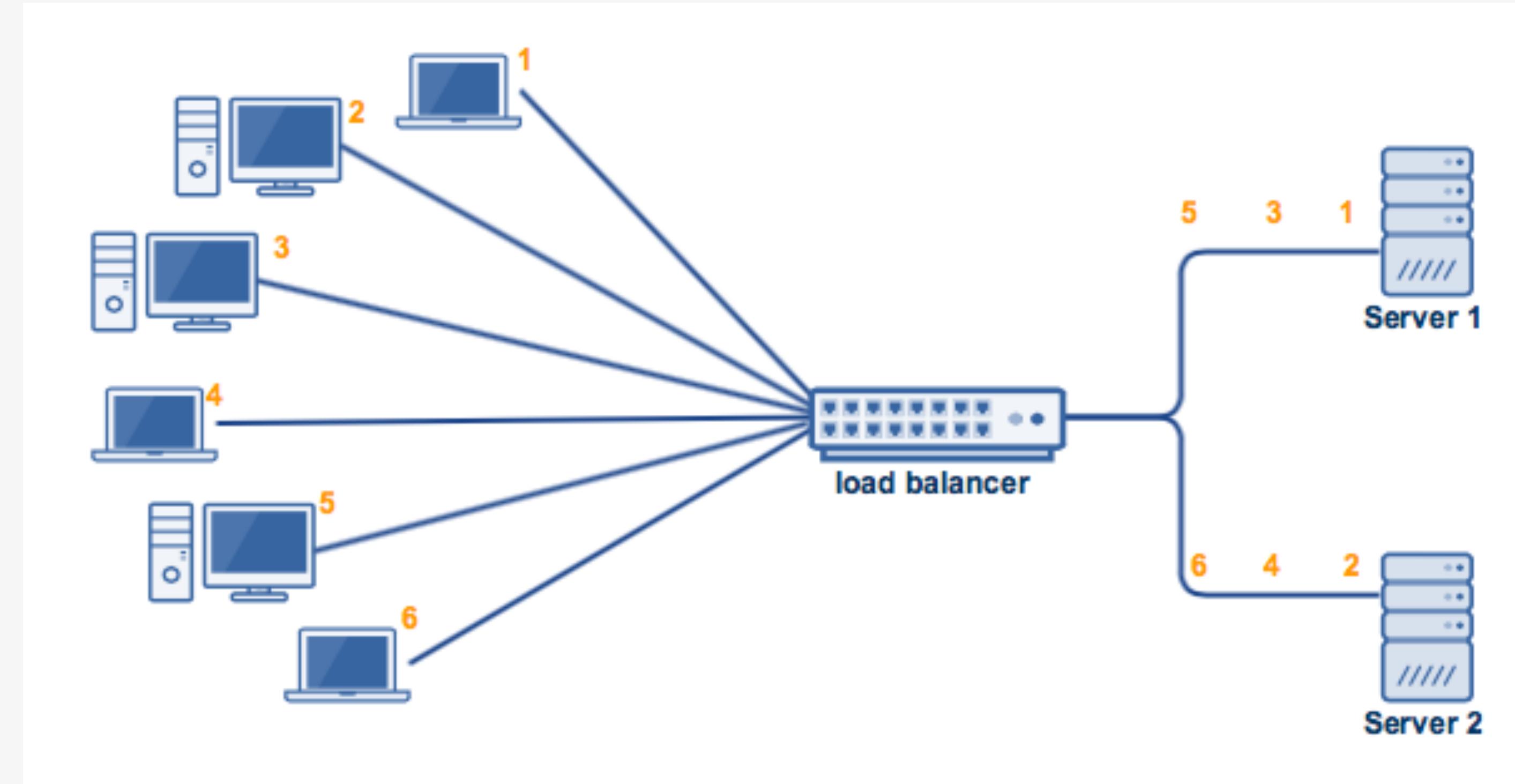
Montée en charge



Lancer de nouvelles instances sur le Cloud.
Répartir leur charge.

ARCHITECTURE SCALABLE

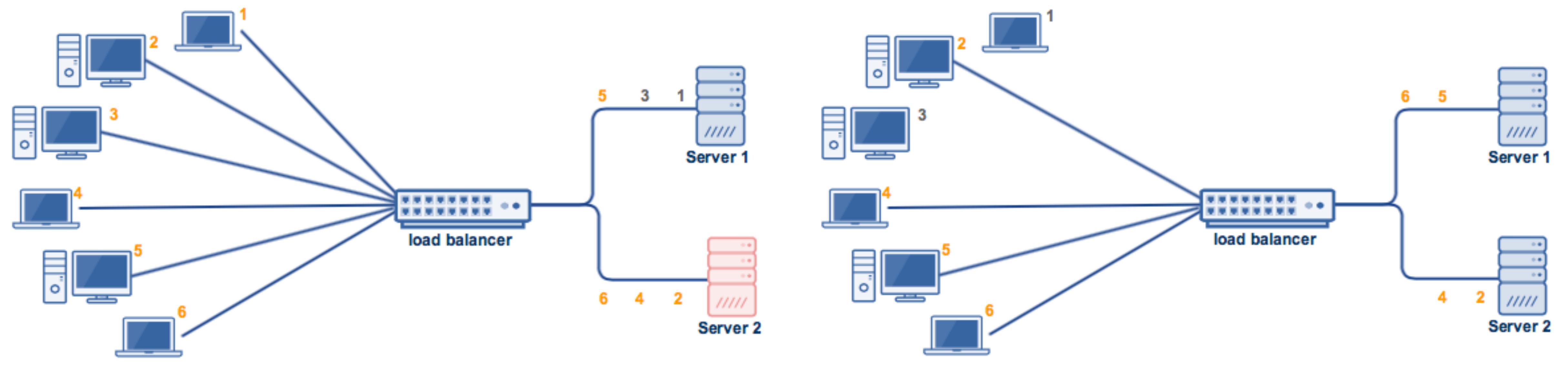
Load Balancing



ROUND ROBIN

ARCHITECTURE SCALABLE

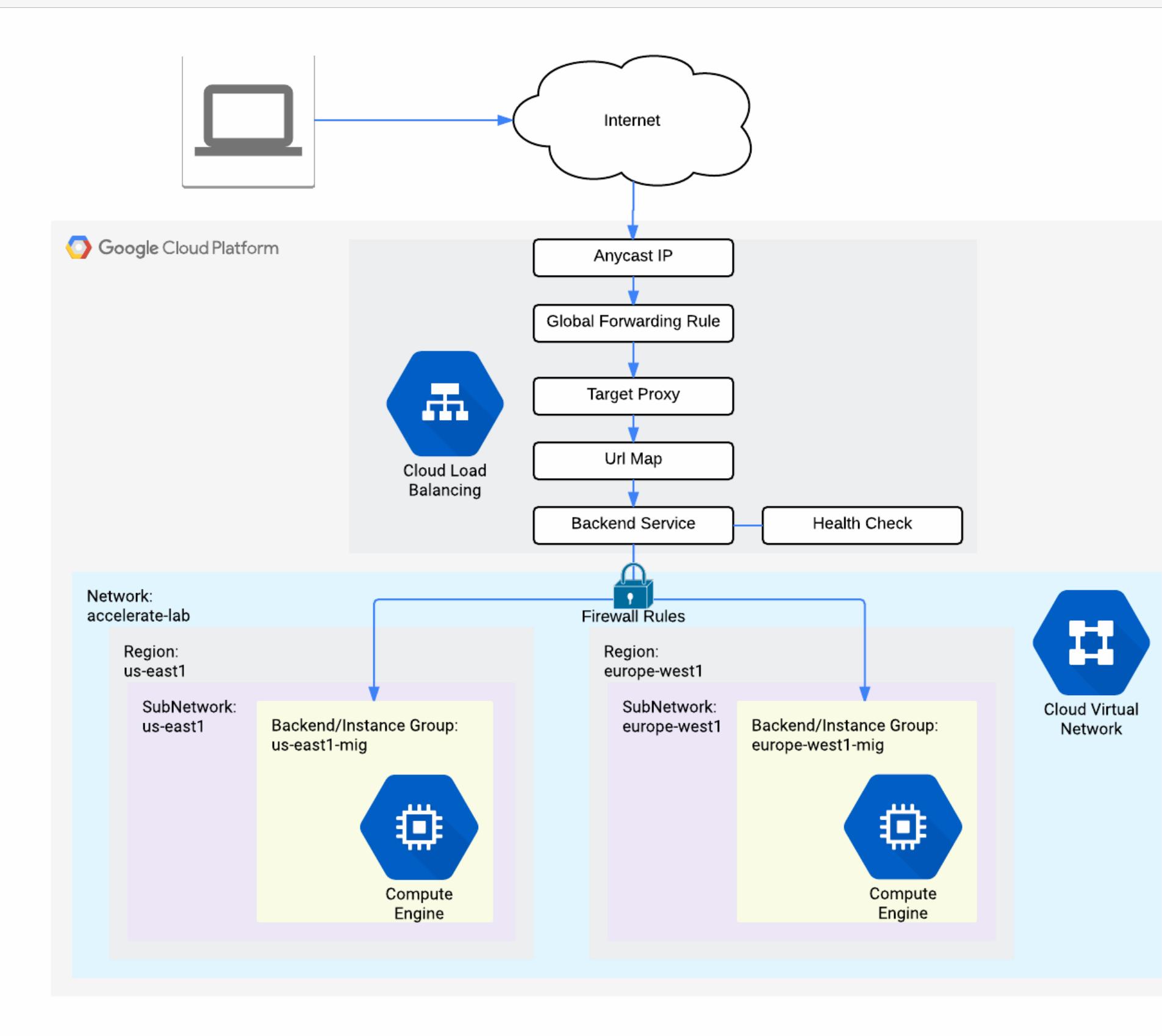
Load Balancing



LEAST CONNECTED

ARCHITECTURE SCALABLE

Solution GCloud



Health Check

Permet de vérifier l'état de la VM.

Backend Service

Ressource contenant la config des services de load balancing. Redirige les requêtes vers un groupe d'instance ou des network endpoints groups

Url Map

Passe les requêtes entrantes à un Backend service.

Target Proxy

Redirige les requêtes vers l'Url Map.

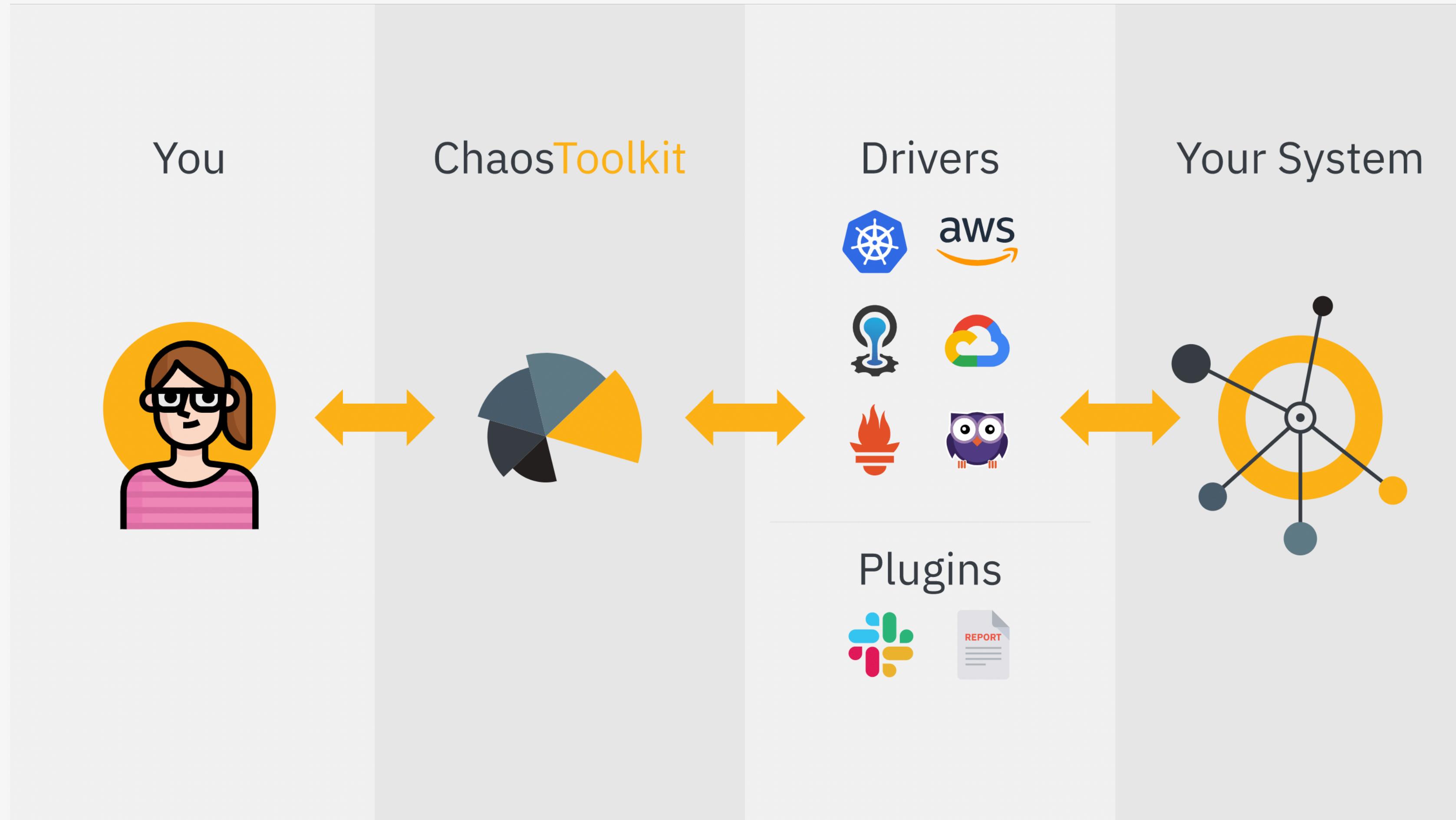
Forwarding Rules

Redirige le traffic vers un proxy. Expose les IPs sur lesquelles le load balancer accepte le traffic.

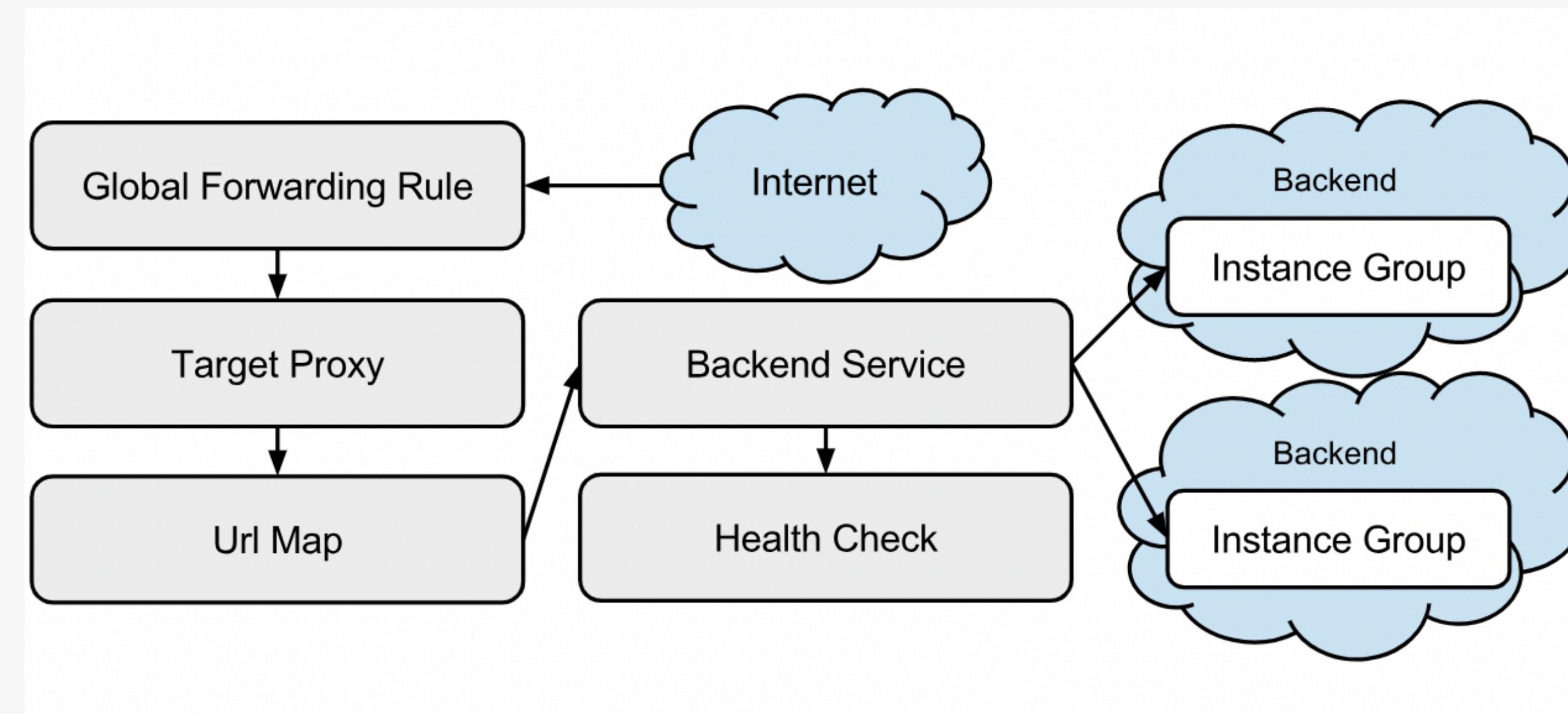
04

ARCHITECTURE SCALABLE

Testeur de charge



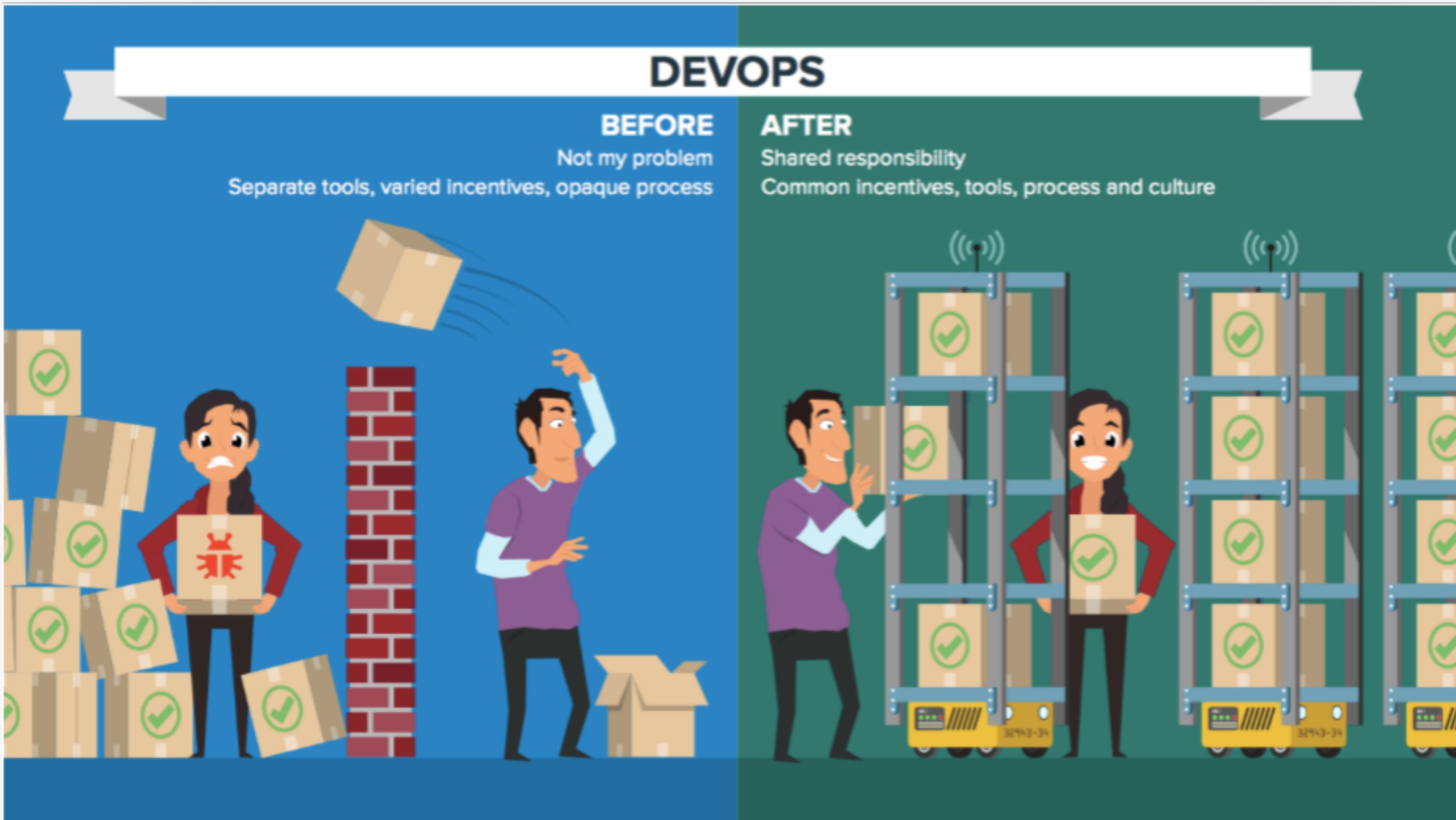
- 01 Déployer une application avec GCE
- 02 Créer un groupe de VMs
- 03 Réaliser l'architecture suivante ...



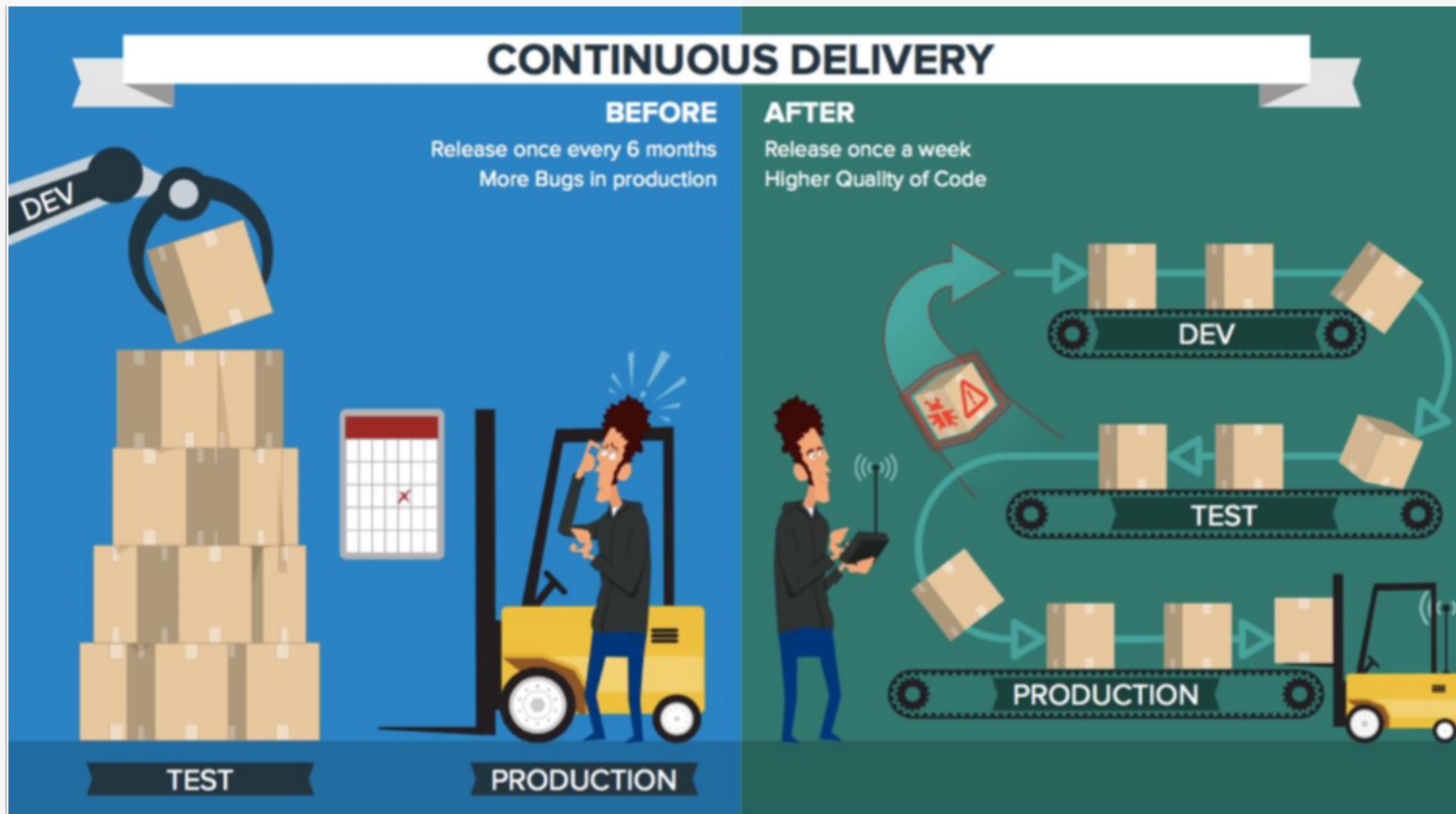
INTRODUCTION AU DEVOPS AVEC GKE

05

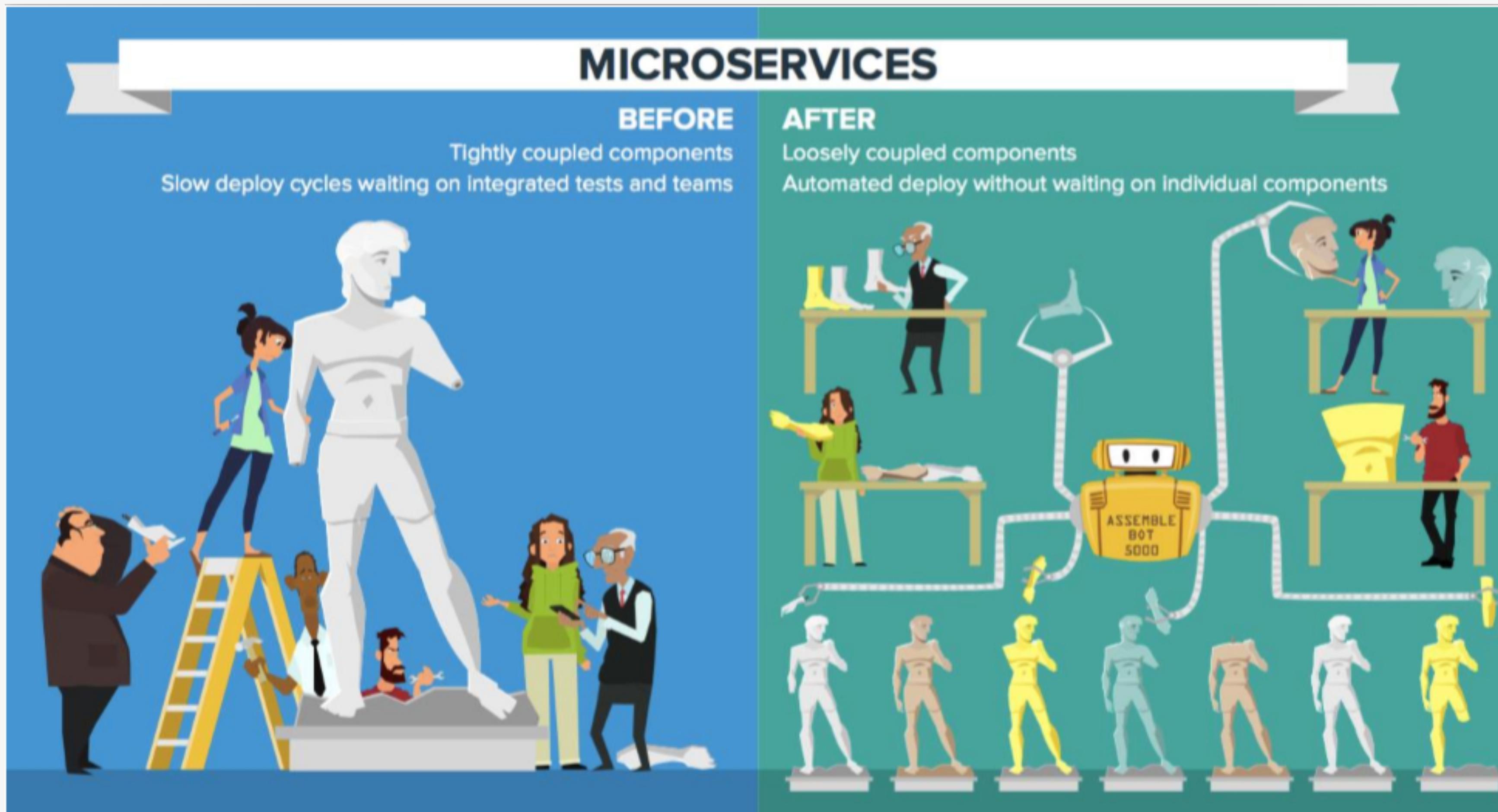
INTRODUCTION AU DEVOPS AVEC GKE



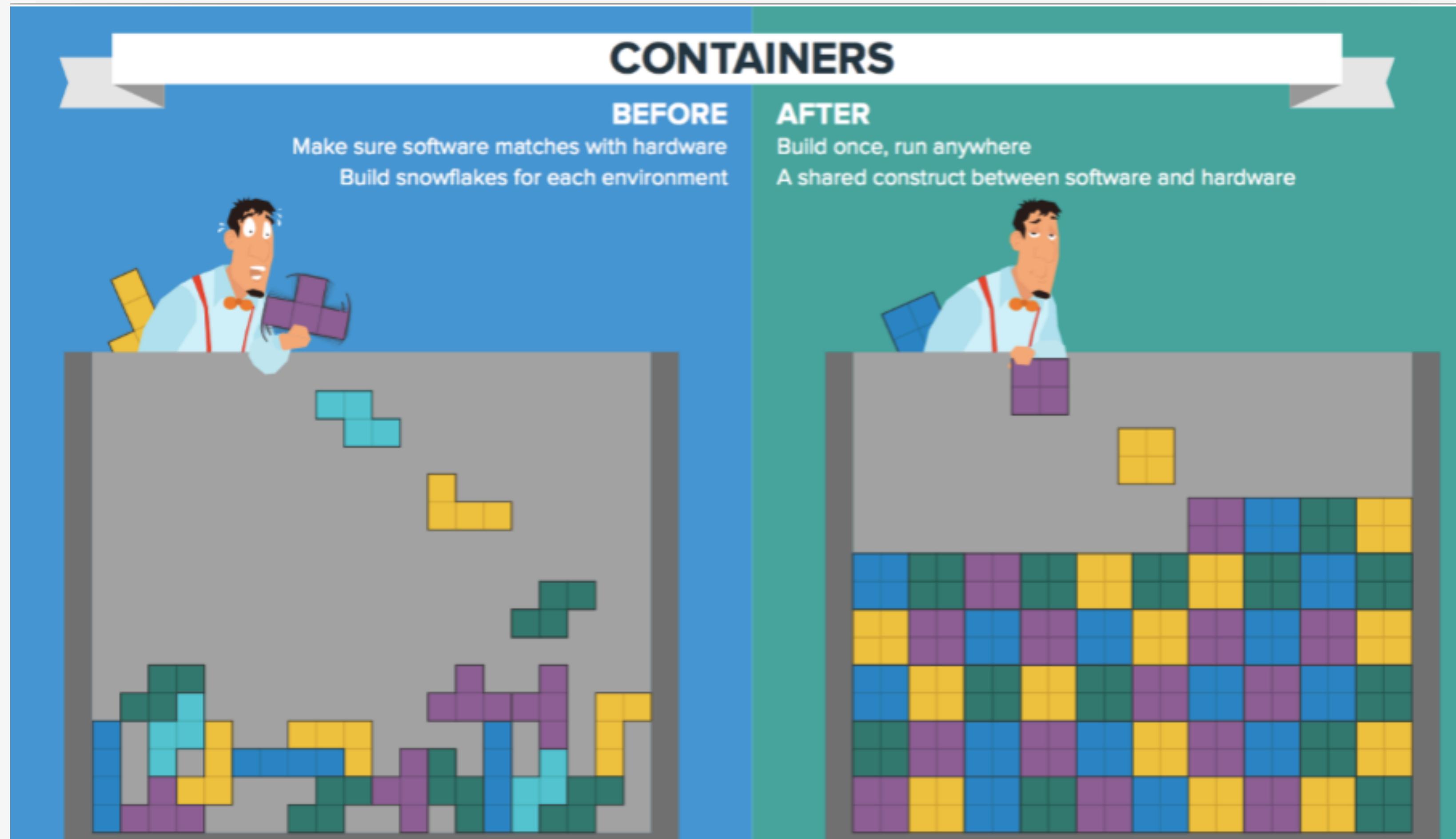
INTRODUCTION AU DEVOPS AVEC GKE



INTRODUCTION AU DEVOPS AVEC GKE

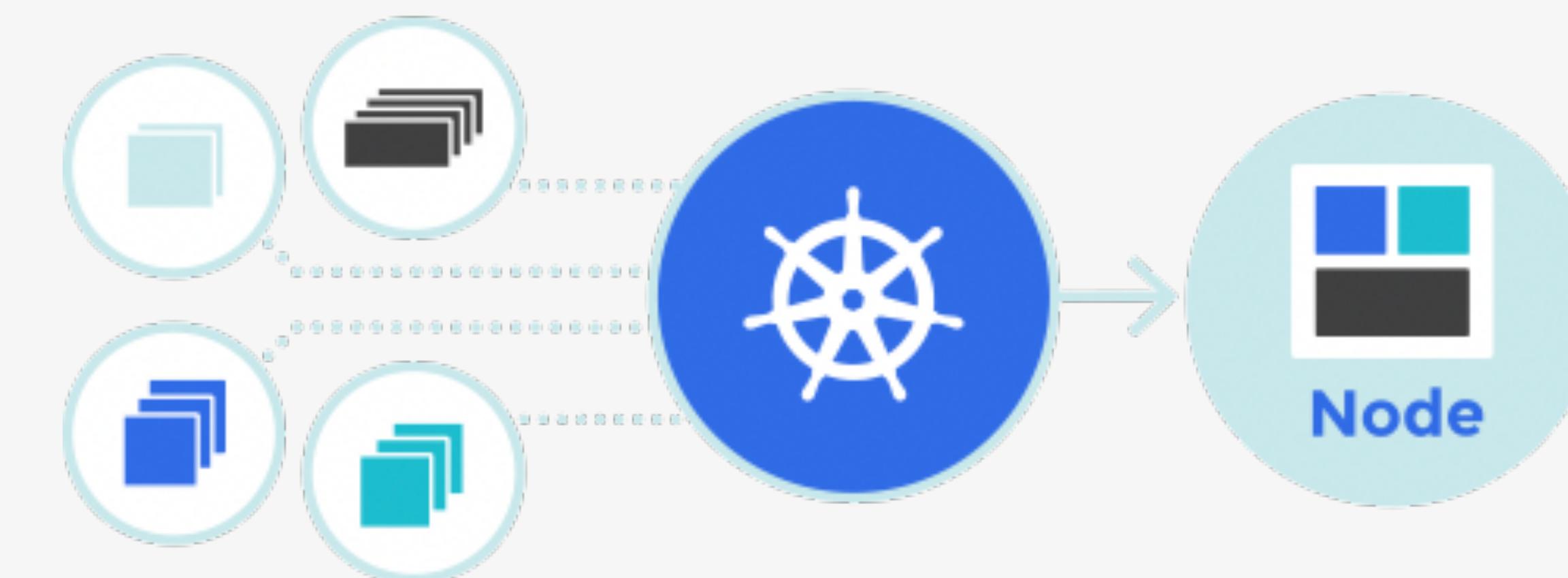
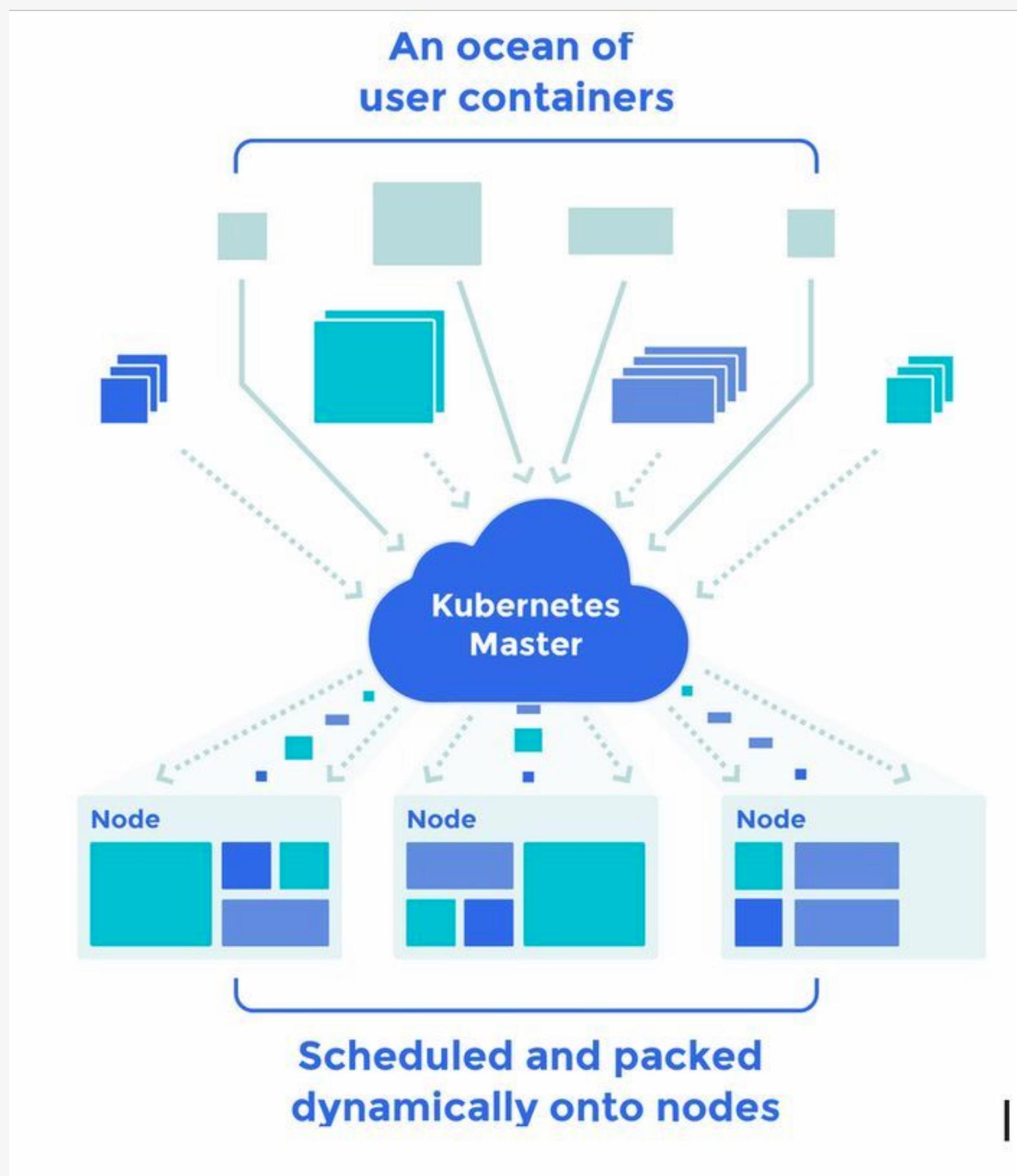


INTRODUCTION AU DEVOPS AVEC GKE



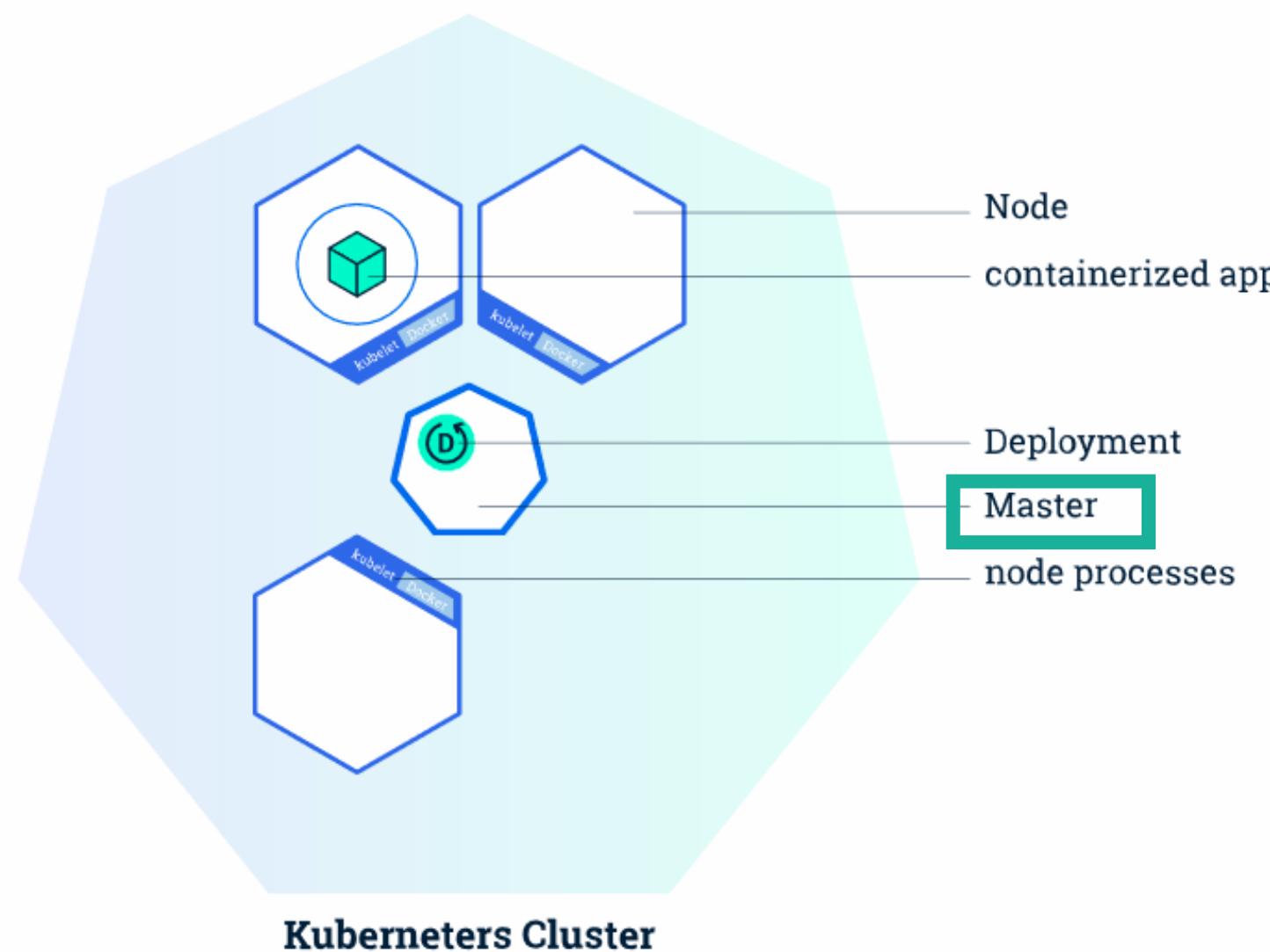
INTRODUCTION AU DEVOPS AVEC GKE

Orchestration



INTRODUCTION AU DEVOPS AVEC GKE

Terminologie Kubernetes



MASTER

Noeud responsable du maintien de l'état souhaité du cluster. (grâce à 3 process)

Kube-apiserver

Expose l'API Kubernetes.

Kube-scheduler

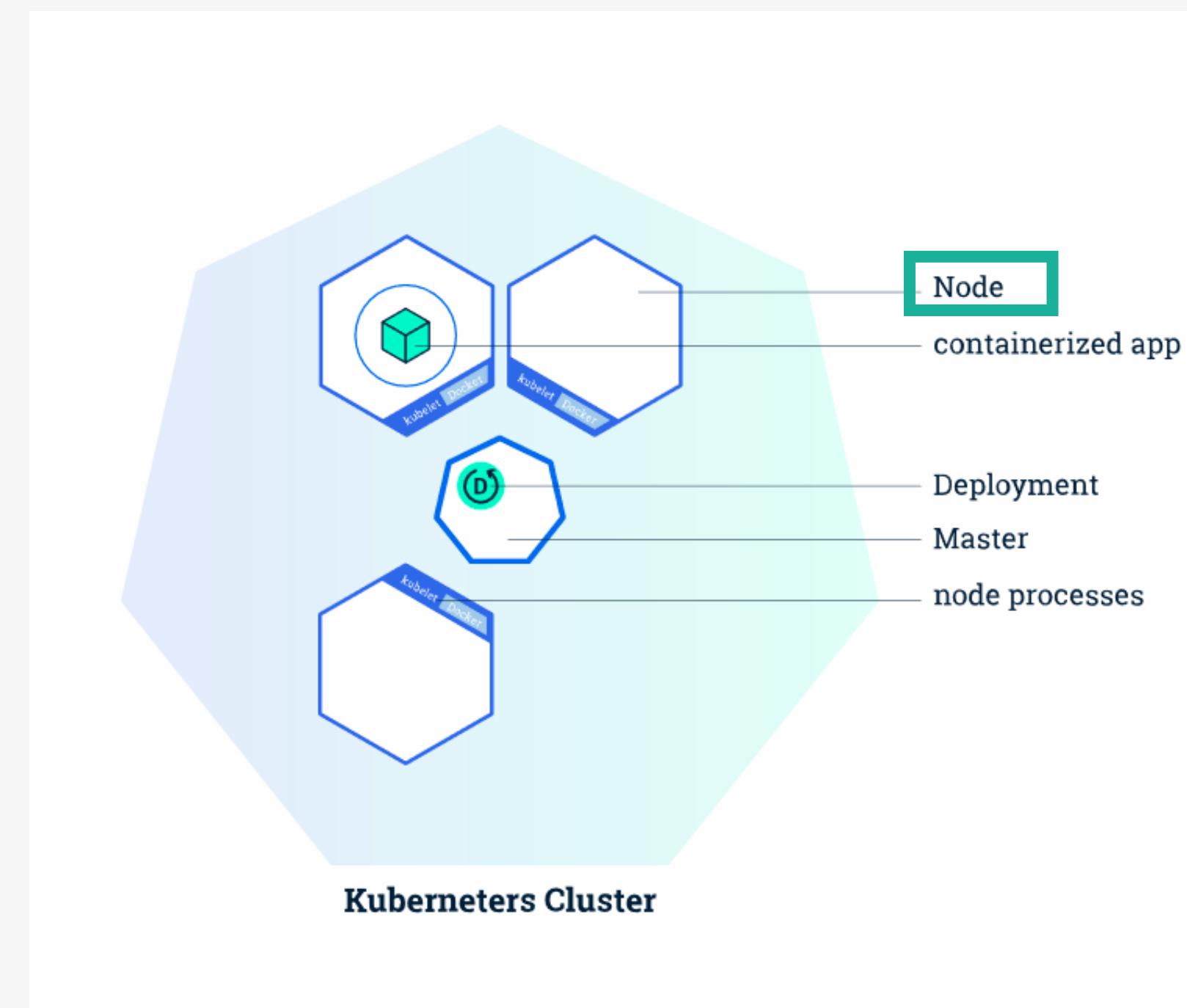
Prend connaissances des Pods non-assignés et sélectionne un noeud pour les exécuter.

Kube-controller-manager

Vérifie l'état du cluster via l'apiserver et applique les changements nécessaires pour le changer.

INTRODUCTION AU DEVOPS AVEC GKE

Terminologie Kubernetes



NODE

Machine virtuelle ou physique selon le cluster. Les noeuds contiennent les services nécessaires à l'exécution des pods et sont gérés par le master.

Container runtime

Gère le téléchargement d'images et l'exécution de containers (Docker, Linux VServer, Java containers ...)

Kubelet

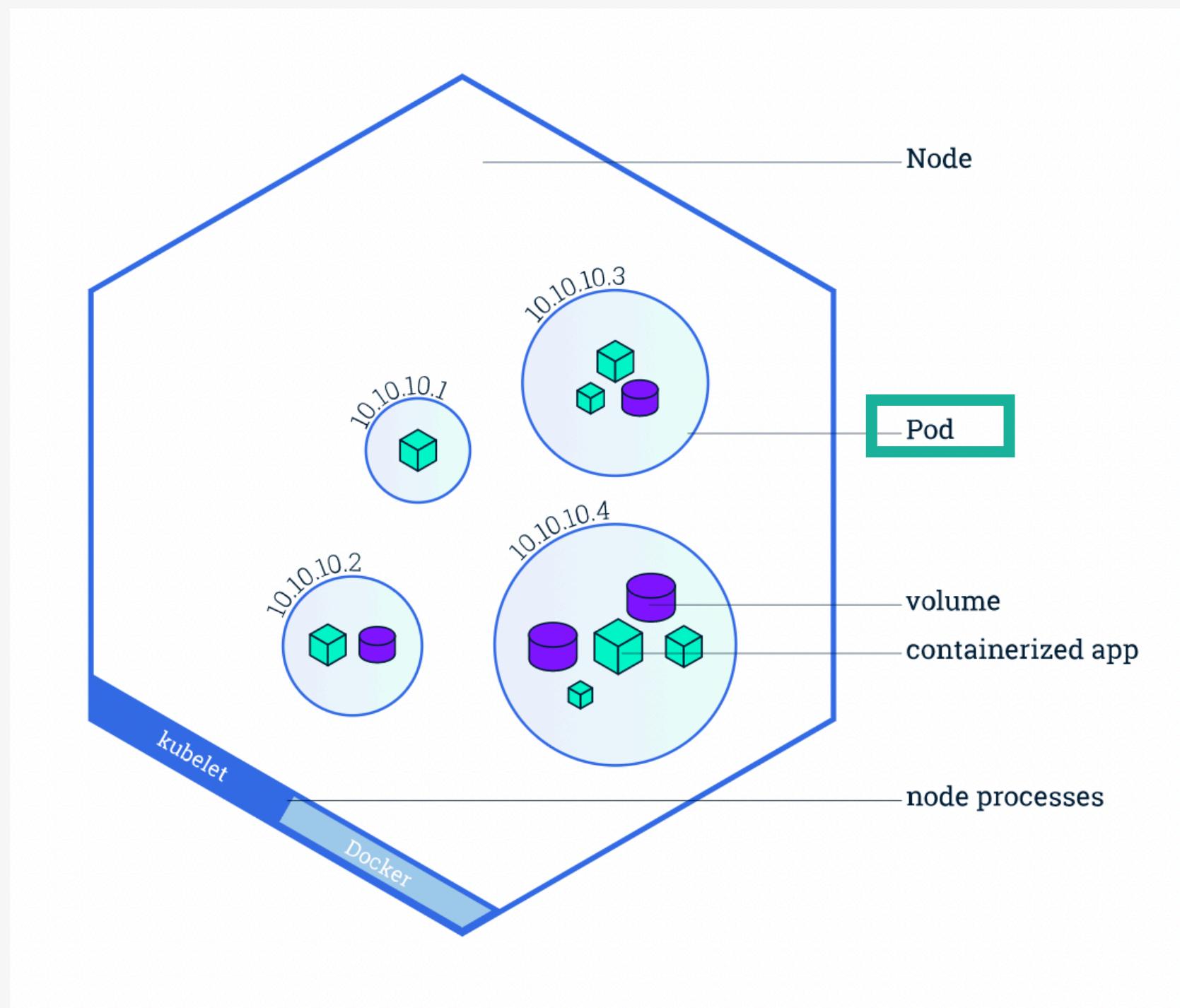
Un agent exécuté sur chaque noeud du cluster. Agit comme le cerveau d'un pod, il surveille l'app et s'assure que les spécifications sont respectées.

Kube-proxy

Proxy qui maintient les règles réseau de chaque noeud. Autorise la communication vers les Pods.

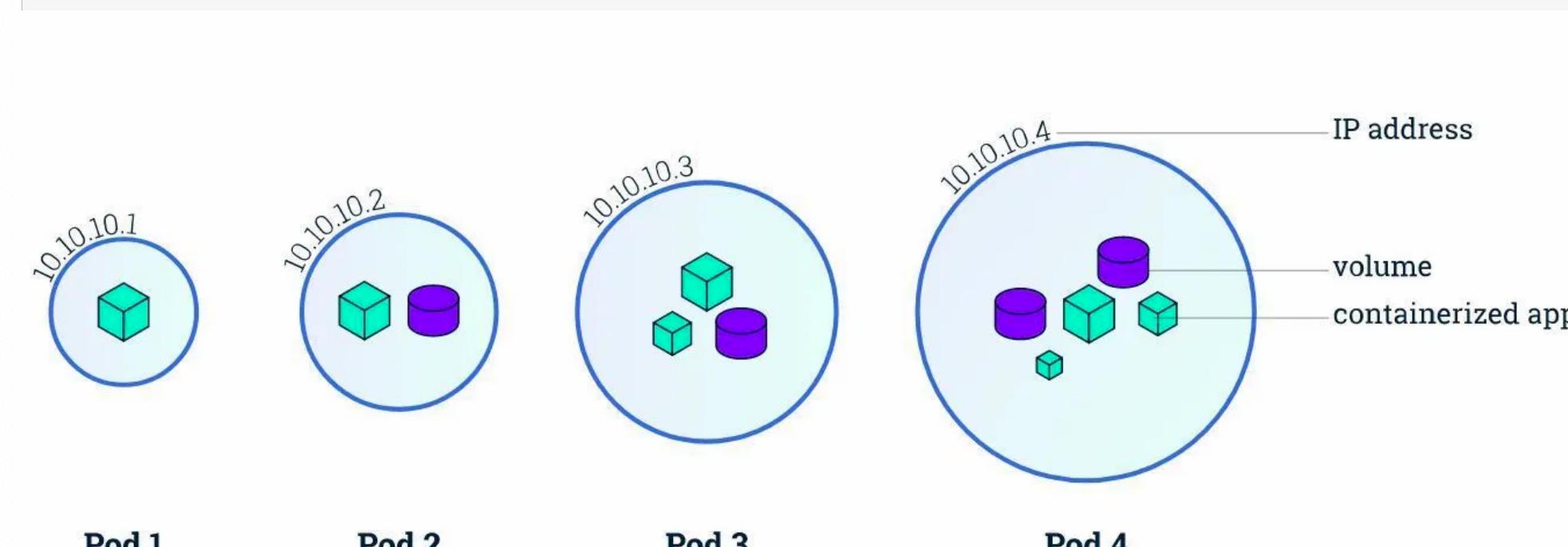
INTRODUCTION AU DEVOPS AVEC GKE

Terminologie Kubernetes



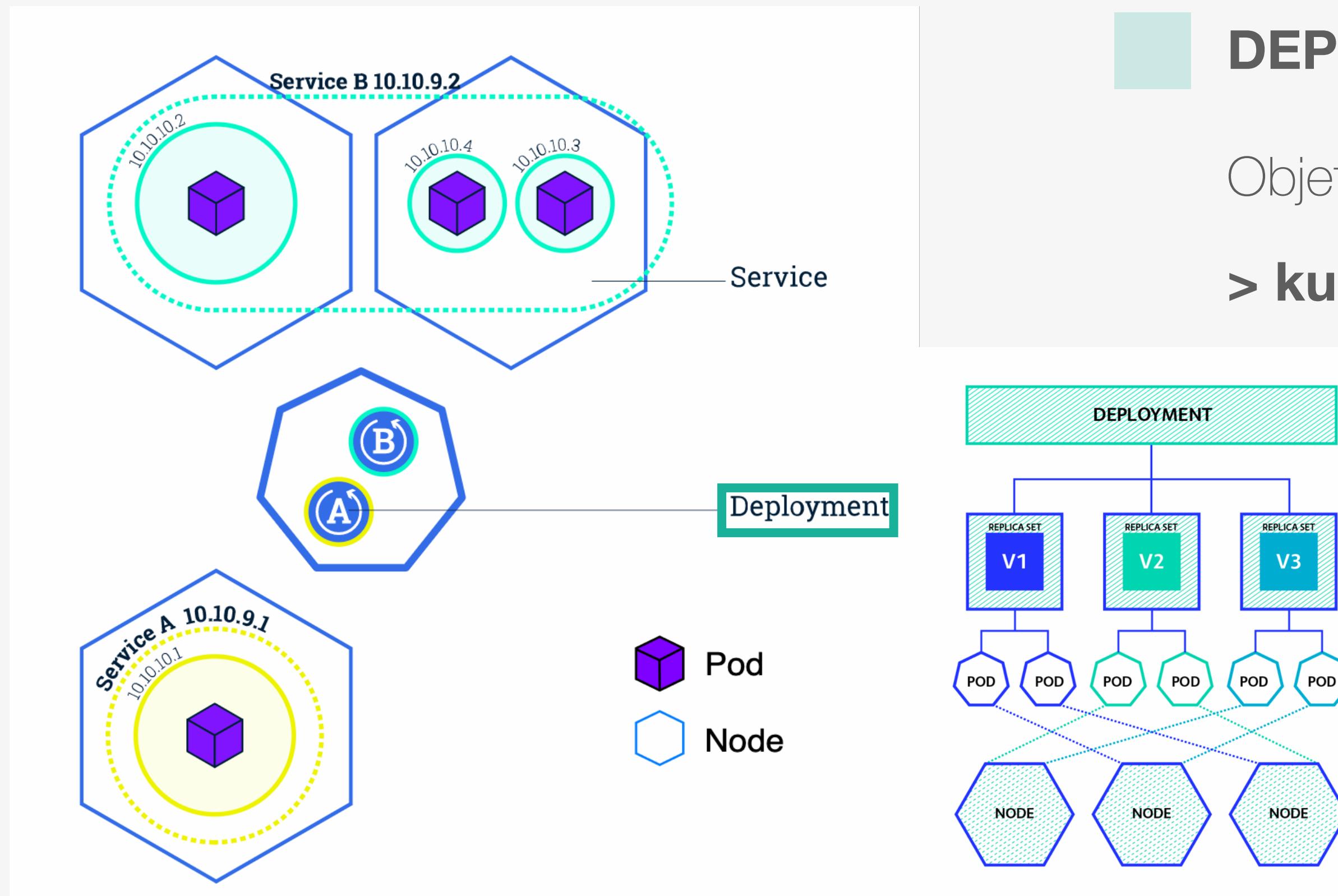
PODS

Groupe d'un ou plusieurs conteneurs possédant les specs sur la manière de les exécuter. Ils sont créés lors d'un **Deployment**.



INTRODUCTION AU DEVOPS AVEC GKE

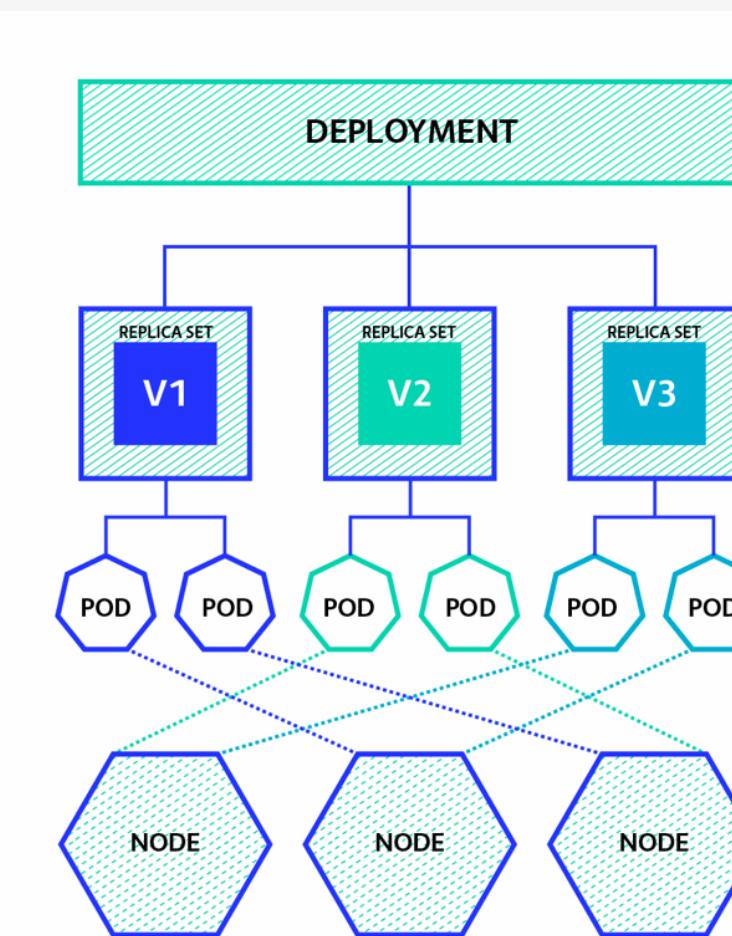
Terminologie Kubernetes



DEPLOYMENT

Objet qui permet le lancement des pods.

> `kubectl create deployment`



INTRODUCTION AU DEVOPS AVEC GKE

Terminologie Kubernetes

```
controllers/nginx-deployment.yaml □

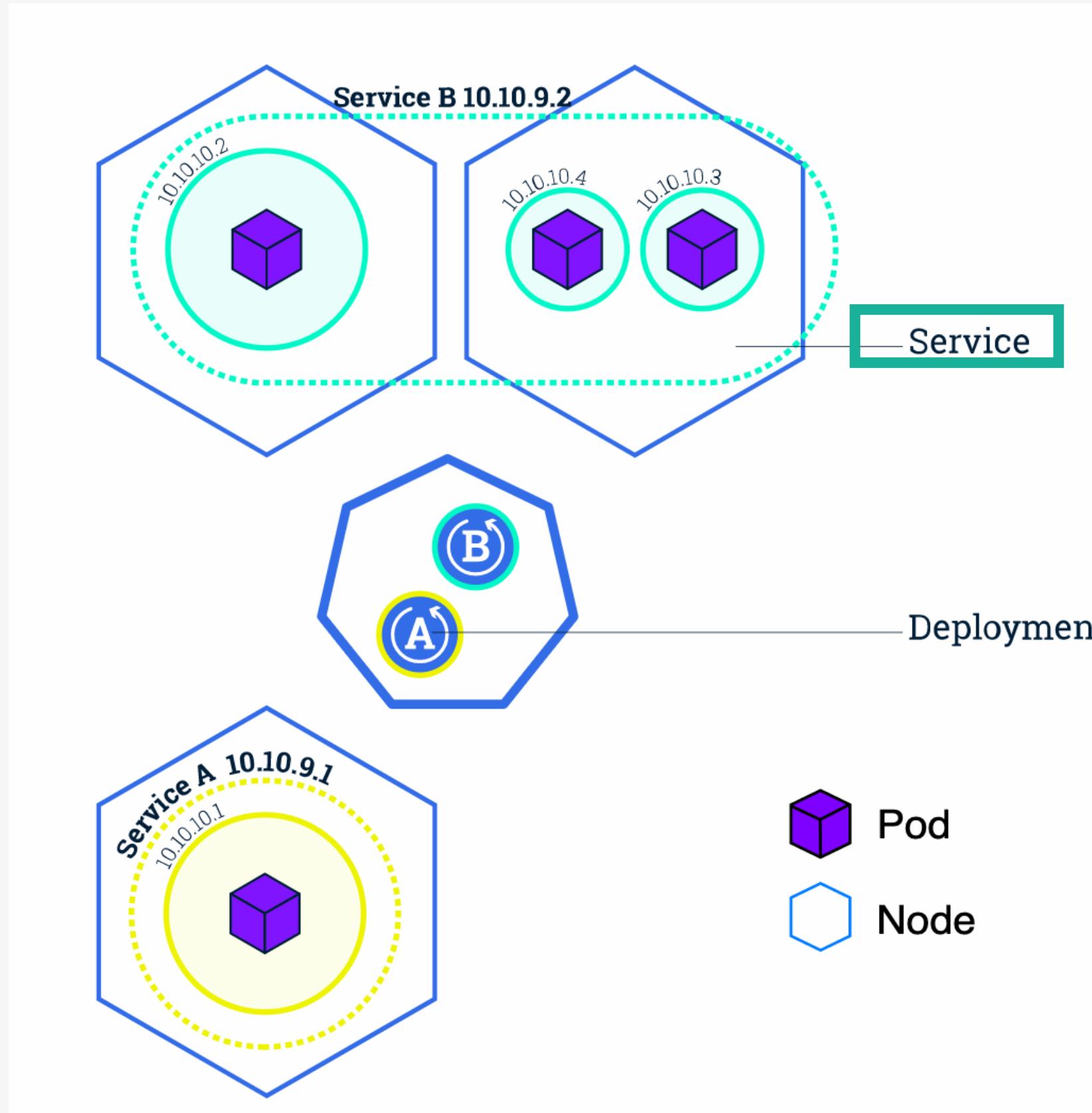
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

Exemple de déploiement.

> kubectl apply -f nginx-deployment.yaml

INTRODUCTION AU DEVOPS AVEC GKE

Terminologie Kubernetes



■ SERVICE

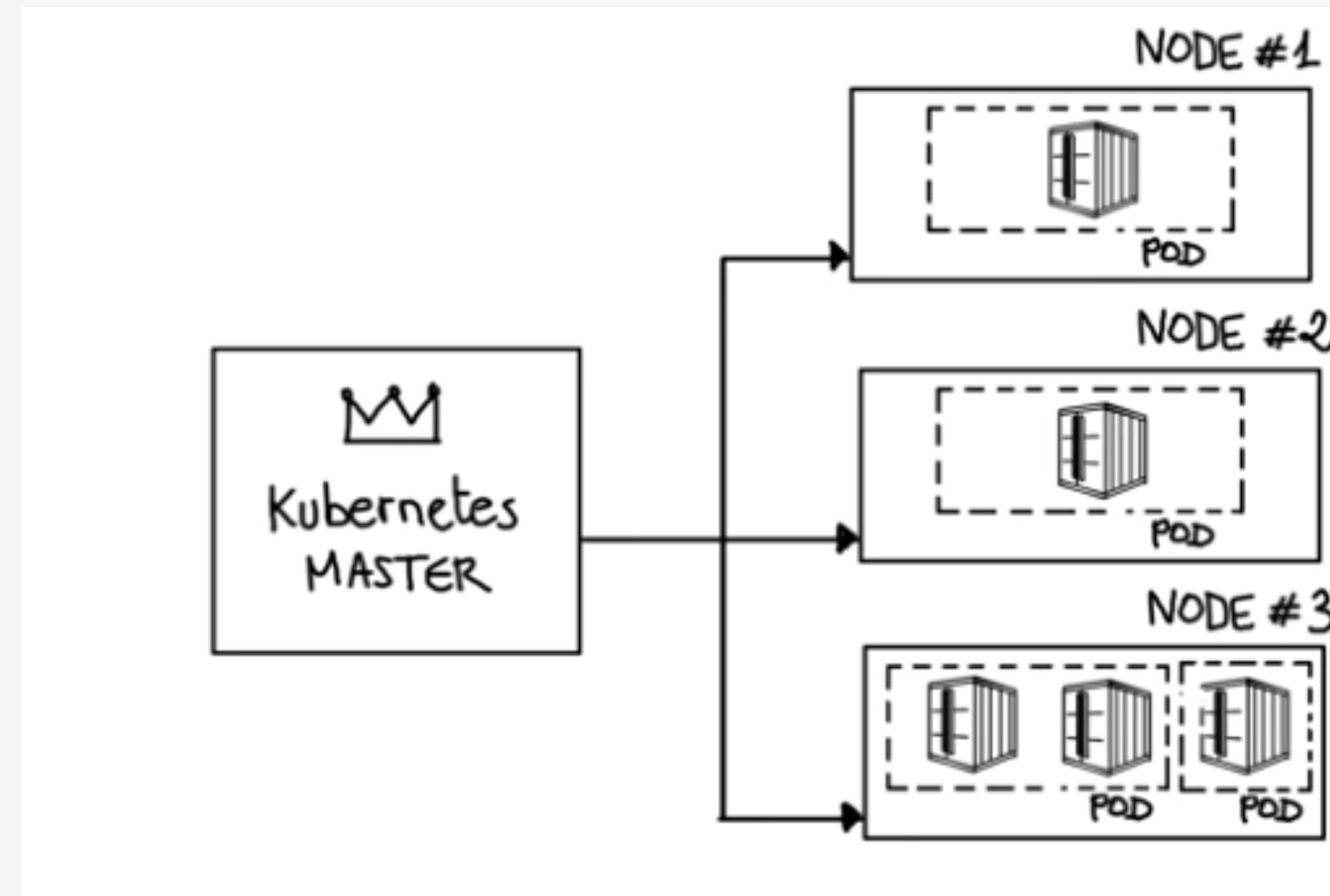
Expose l'application exécutée par un ensemble de Pods. Il possède une IP statique utilisée comme référence pour l'accès aux pods.

> kubectl expose deployment

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

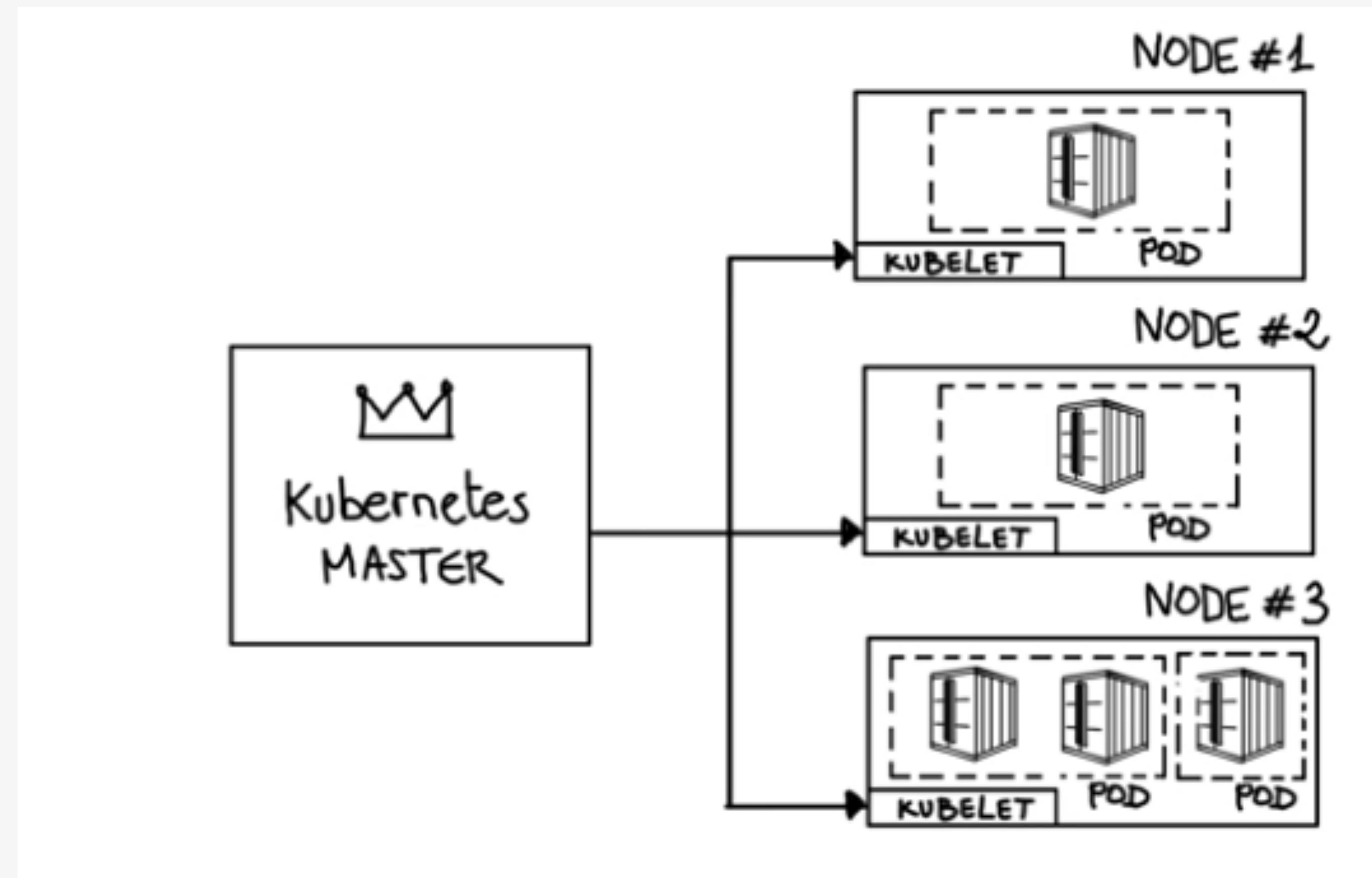
INTRODUCTION AU DEVOPS AVEC GKE

Concept de Kubernetes



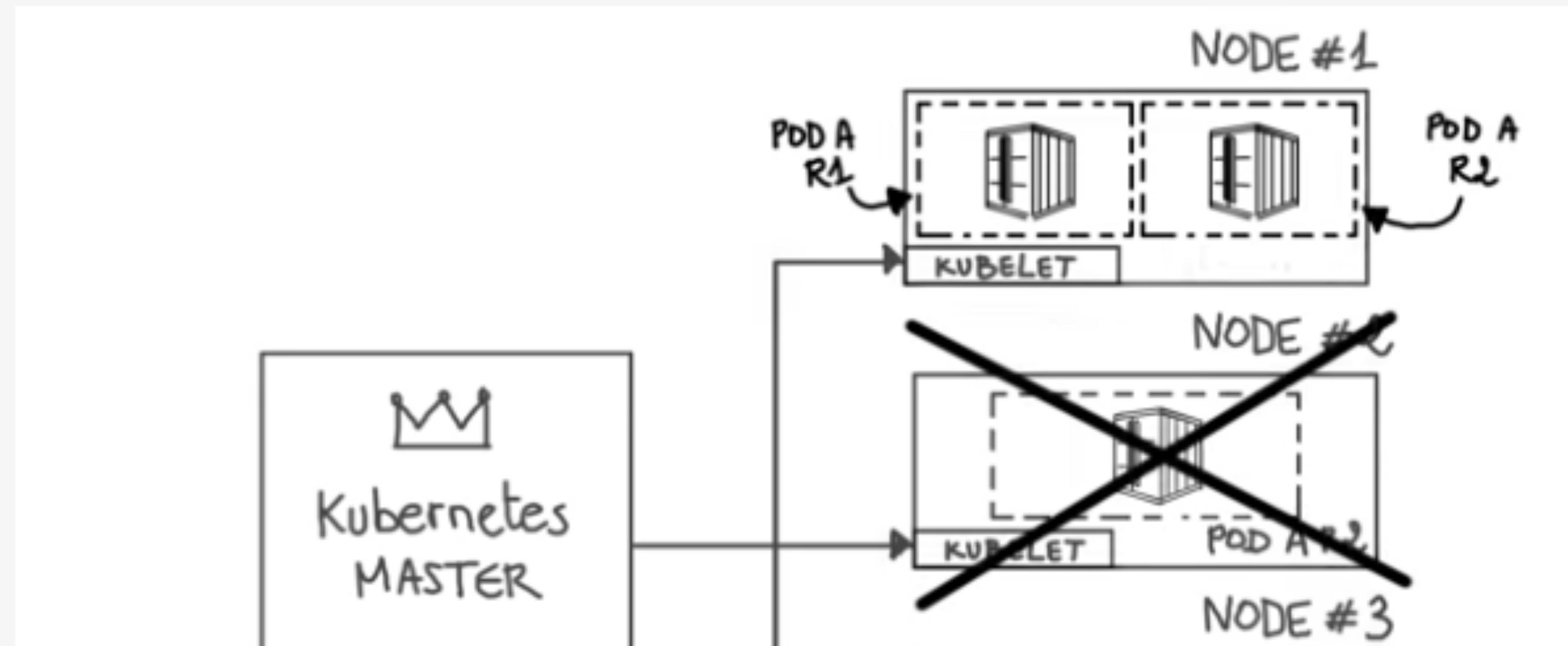
INTRODUCTION AU DEVOPS AVEC GKE

Concept de Kubernetes



INTRODUCTION AU DEVOPS AVEC GKE

Concept de Kubernetes



INTRODUCTION AU DEVOPS AVEC GKE

Introduction Docker

```
FROM node:12

WORKDIR /usr/src/app
COPY package*.json ./

RUN npm install
COPY . .

EXPOSE 8080
CMD [ "node", "server.js" ]

> docker build -t brad/node-app .
> docker images
> docker run -p 8080:8080 -d brad/node-app
> docker ps
> docker attach
> docker exec -it container_id bash
```

INTRODUCTION AU DEVOPS AVEC GKE

Introduction Docker

```
version: "2"

services:

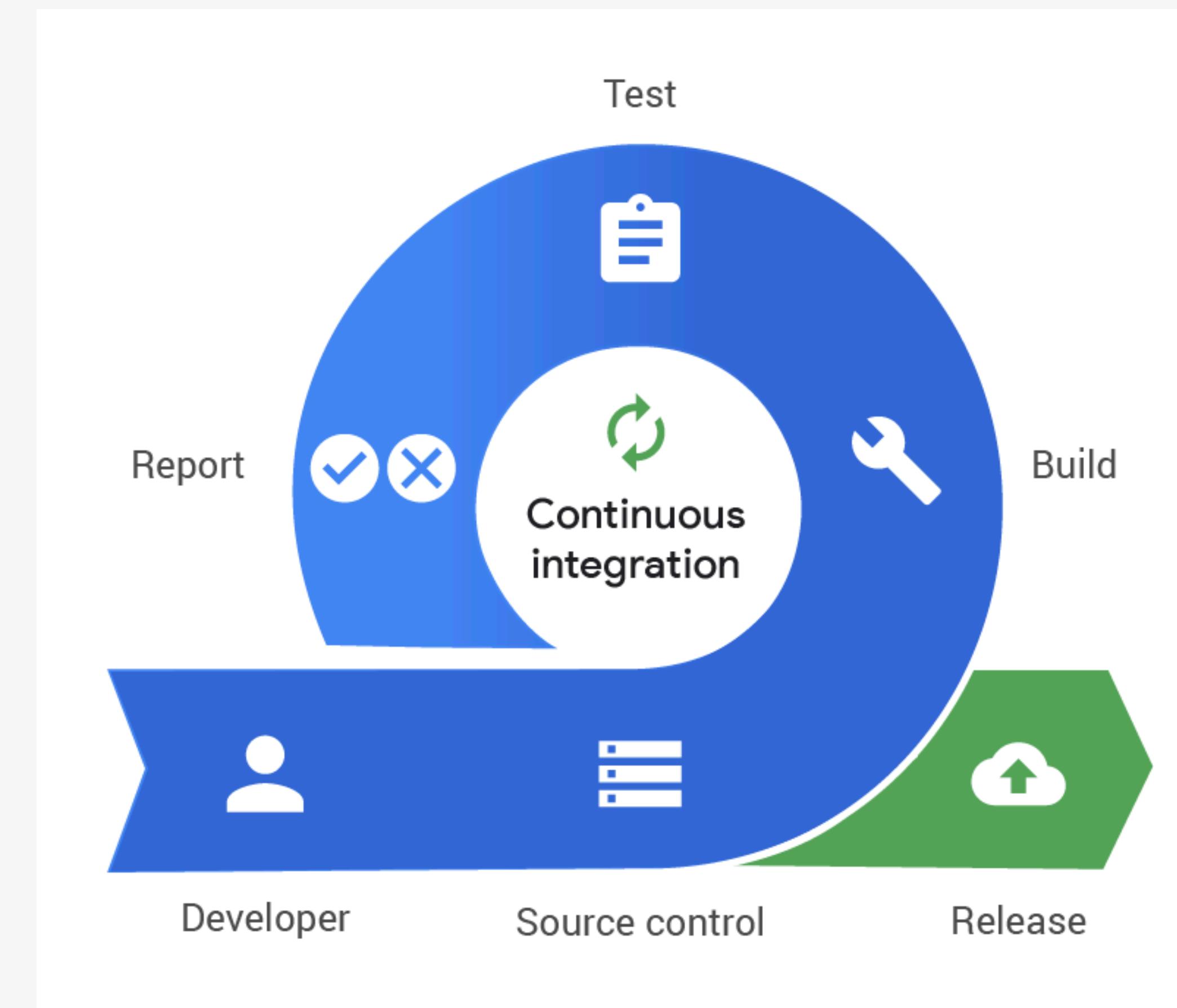
  redis-master:
    image: k8s.gcr.io/redis:e2e
    ports:
      - "6379"

  redis-slave:
    image: gcr.io/google_samples/gb-redisslave:v3
    ports:
      - "6379"
    environment:
      - GET_HOSTS_FROM=dns

  frontend:
    image: gcr.io/google-samples/gb-frontend:v4
    ports:
      - "80:80"
    environment:
      - GET_HOSTS_FROM=dns
    labels:
      kompose.service.type: LoadBalancer
```

INTRODUCTION AU DEVOPS AVEC GKE

Intégration / Déploiement continu



INTRODUCTION AU DEVOPS AVEC GKE

Outil Kubernetes



■ MiniKube

- En gros : on exécute K8s en local
- Ce sert d'un cluster à noeud unique dans une VM
- Pratique pour la pédagogie ou le test
- Utilise kube-ctl pour l'interaction avec le cluster

INTRODUCTION AU DEVOPS AVEC GKE

Outil Kubernetes



Helm

- Outil d'empaquetage
- Permet la gestion du cycle de vie de vos applis k8s
- Les paquets sont appelés « **charts** »
- Permet donc la réutilisation d'outils développés par la communauté.
- Permet de définir des templates de déploiement

INTRODUCTION AU DEVOPS AVEC GKE

Outil Kubernetes



PGA Stack

- Collecte des métriques = **Prometheus**
- Visualization = **Grafana**
- Alertes = **AlertManager**

Démonstration MiniKube, Helm, PGA

INTRODUCTION AU DEVOPS AVEC GKE

TP - Déploiement K8s

- 01** Déployer un service sur un cluster Minikube
- 02** Déployer plusieurs services sur un cluster Minikube (en même temps)
- 03** Faire en sorte que le Load balancing soit en place
- 04** Recommencer sur GKE

INTRODUCTION AU DEVOPS AVEC GKE

TP - Projet autonomie

- 01** Utiliser la stack PGA pour proposer une stratégie de monitoring de micro-services. Tester la charge de votre application.

- 02** Développer une appli basée sur des micro-services et la déployer sur un cluster Kubernetes. Mettre en place le déploiement continue. L'appli doit contenir un appel à une API distante avec authentification (Ex : BattleNet)

INTRODUCTION AU DEVOPS AVEC GKE

Instructions - TRAVIS

- 01** Activer votre repo sur Travis-ci
- 02** Ajouter un fichier .travis.yml au repo
- 03** Créer un compte de service gcloud pour travis
- 04** Sécuriser le fichier json du compte de service
- 05** Mettre en place deux déploiements Dev / Prod (différent cluster)

```
travis encrypt-file client-secret.json
```