



# AWS Lambda

Ping Sen & Tas Emine [5min]

ping.sen@hotmail.com & tas.emine@outlook.be

27/02/19

# Sommaire

- Introduction
- Développement
- Fonction Lambda
- Cas d'utilisation
- Démonstration
- Payement
- Conclusion

# Introduction (1)

Qu'est-ce **AWS**?

- **Amazon Web Service** est une plateforme

Pourquoi nommé « **Lambda** » ?

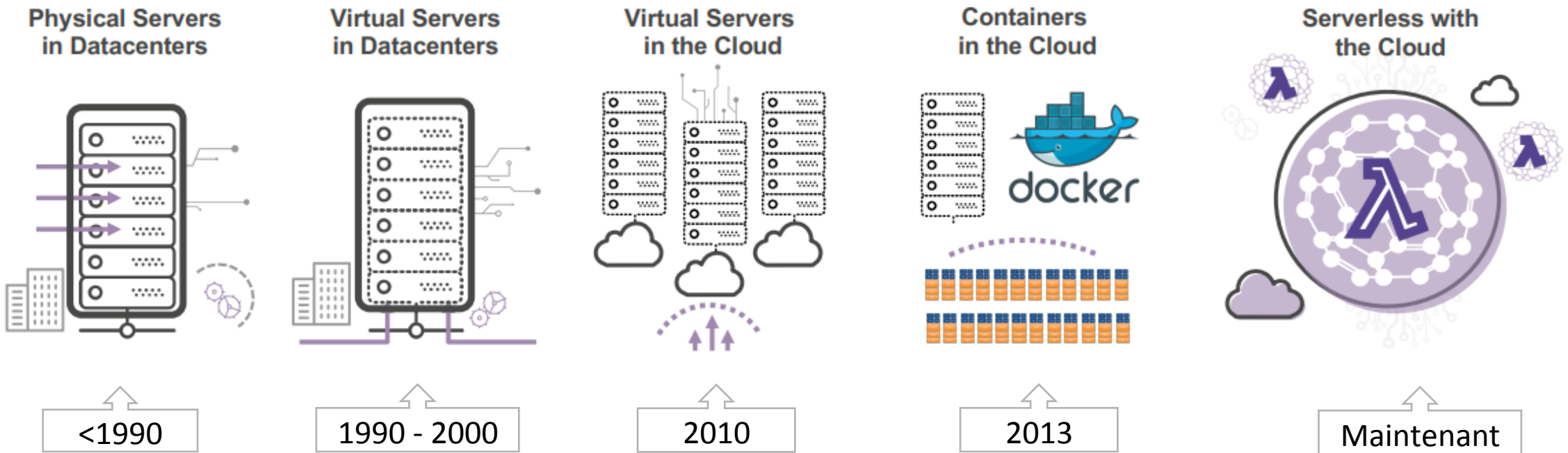
- À cause de la fonction lambda

Qu'offre-t-il?

- Service informatique
- Exécution de code lorsque nécessaire et adaptable
- Paiement uniquement du temps de calcul
- Pour n'importe quel type d'application ou service backend
- Infrastructure de calcul à haute disponibilité
- Langages compatibles : Node.js, Java, C#, Go & Python
- Responsabilité du code uniquement

# Introduction (2)

Comment ont évolué les serveurs?



# AWS Lambda

- **AWS Lambda** est un service informatique
- Function as a service (Faas)
- Paradigme de programmation événementielle



# Serverless

- Infrastructure
- Tolérance aux pannes
- Maintenance
- Dimensionnement
- Mise à l'échelle



# Développement d'une application basée sur AWS (1)

- Création de code pour la fonction **Lambda**
  - Dans les langages pris en charge
  - Avec des outils fournis
- Déploiement : Importation de code et création de fonction **Lambda**
  - Création d'un package de déploiement – Organisation du code et des dépendances
  - Chargement d'un package de déploiement – Création d'une fonction **Lambda**
  - Test d'une fonction **Lambda**
- Surveillance et dépannage
  - Surveillance automatique
  - Métrique via Amazon CloudWatch
  - Aide à la résolution de problèmes via log

# Développement d'une application basée sur **AWS** (2)





# Fonction Lambda

Qu'est-ce que c'est ?

- Code
- Dépendances (bibliothèques, modules, ...)
- Configuration

Comment est-il ?

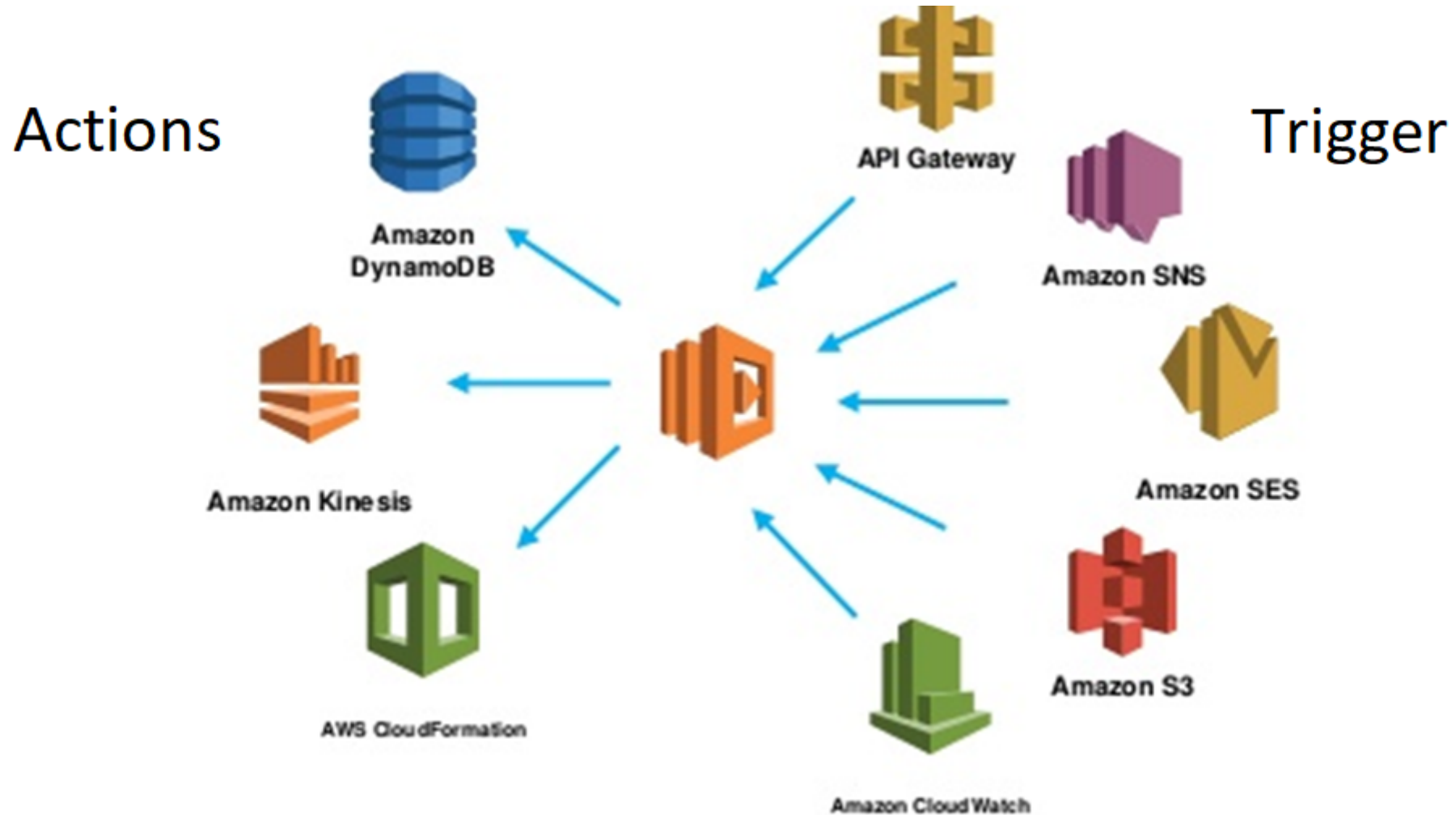
- Ready
- Stateless



# Comparaison

	Fonction Lambda	Fonction
Maintenance	Automatique	Manuelle
Scalabilité	Automatique	Manuelle
Opération	Automatique	Manuelle
Exécution	Déclenchement	En attente
Design & implémentation	Règle	Libre
Etats	Stateless	Stateless/Statefull
Prix	Variable	Fixe

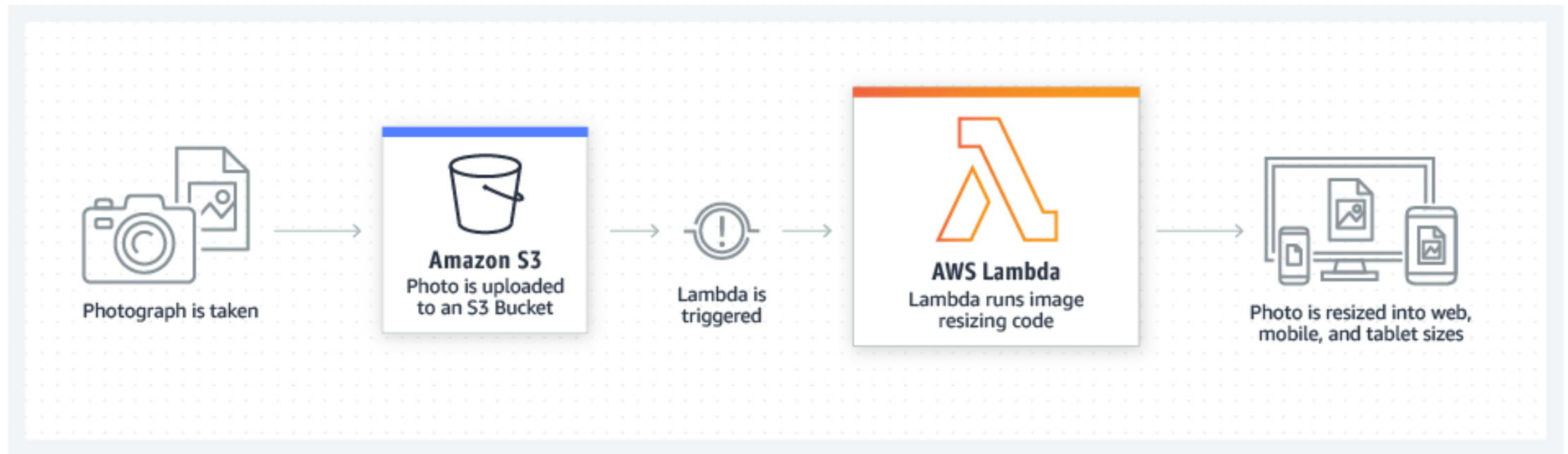
# Comment exécuter un Lambda ?



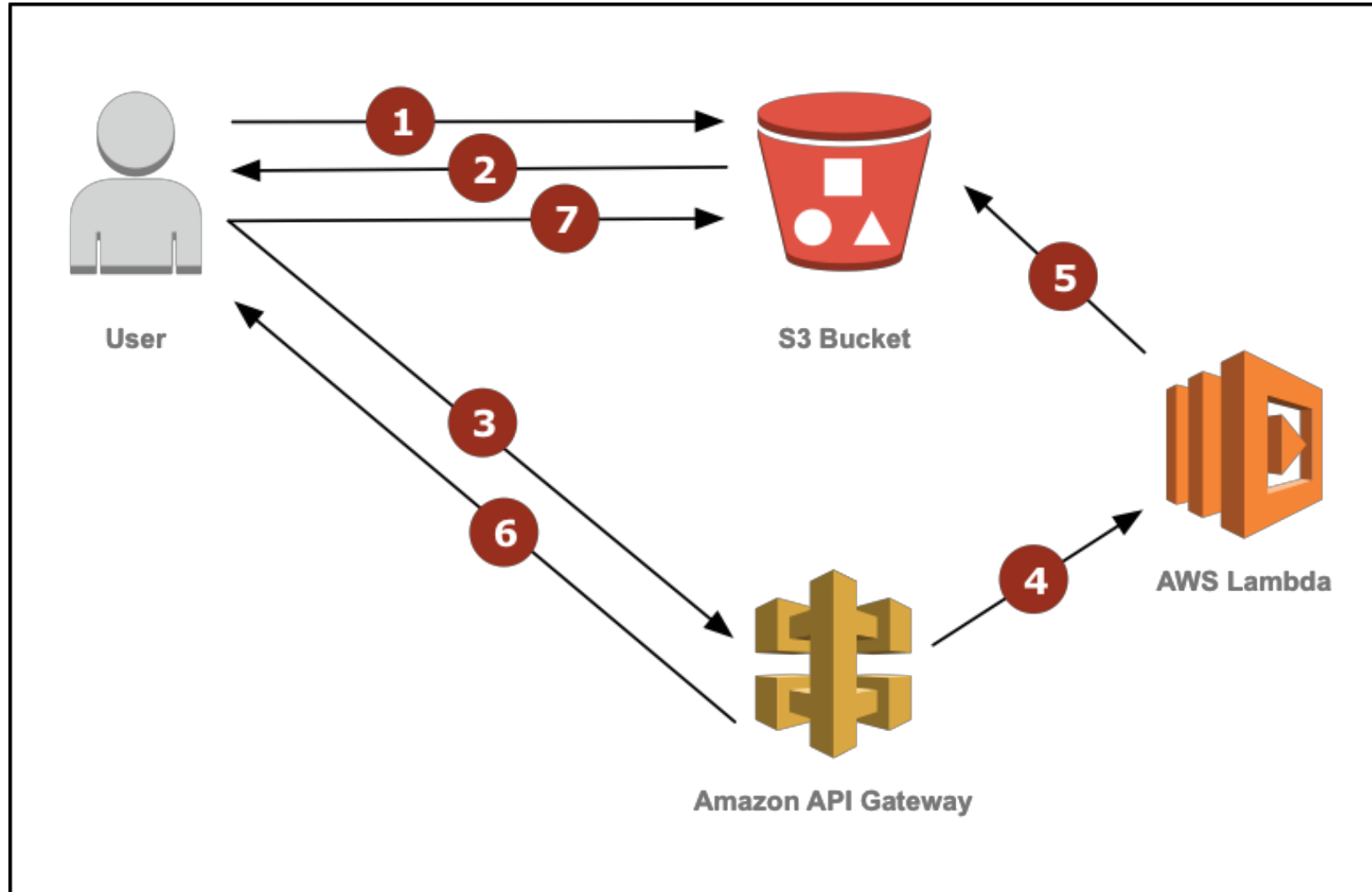
# Exemples d'utilisation d'AWS Lambda

- Utilisation d'AWS Lambda avec les services AWS en tant que sources d'événements
- Appel de la fonction Lambda à la demande via HTTPS (Amazon API Gateway)
- Appel de la fonction Lambda à la demande (création de vos propres sources d'événements à l'aide d'applications personnalisées)
- Événements planifiés

# Cas d'utilisation



# Démonstration



# Payements

Qu'en est-il des coûts ?

- Croiser le nombre de requêtes mensuelles et la durée d'exécution de celles-ci

## Exemple

- Une fonction, alloue 128Mb, exécuté 30 millions de fois, lancé pendant 200ms à chaque fois :
  - Charge de calcul : 5,83\$
  - Charge de requête : 5,80\$
  - Charge totale (mois) : 11,63\$



# Conclusion

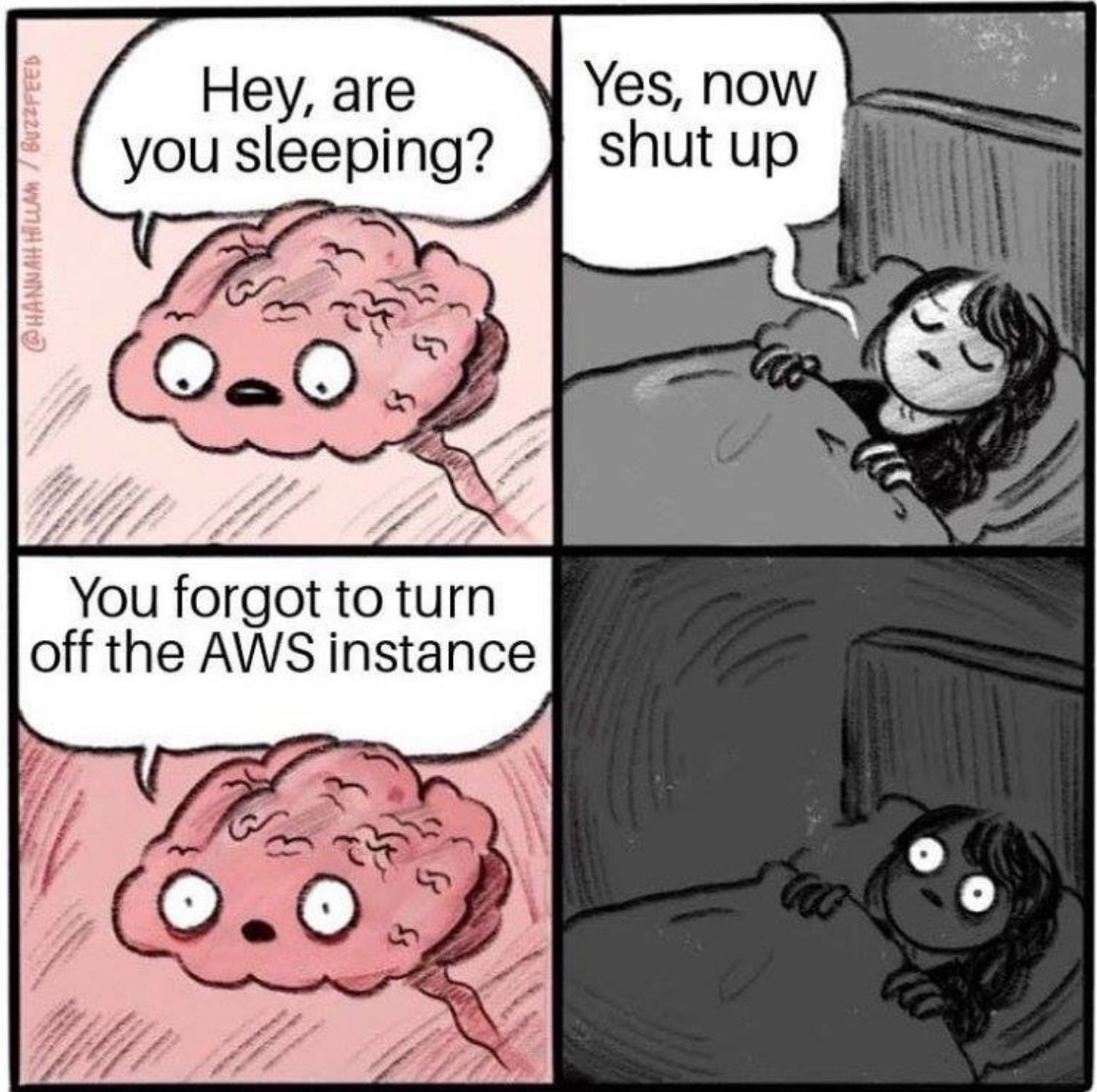
## Avantages :

- Plus de préoccupation pour l'infrastructure
- Scalabilité
- Serverless
- Architecture orientée événements
- Paiement
- Sécurité

## Désavantages :

- Technologie Serverless non standardisée
- Dépendance du système d'api tiers
- Intégration continue





@HANNAHILLMAN / BUZZFEED

Hey, are you sleeping?

Yes, now shut up

You forgot to turn off the AWS instance

# Crédits

- <https://aws.amazon.com/blogs/compute/resize-images-on-the-fly-with-amazon-s3-aws-lambda-and-amazon-api-gateway/>
- [https://docs.aws.amazon.com/fr\\_fr/lambda/latest/dg/welcome.html](https://docs.aws.amazon.com/fr_fr/lambda/latest/dg/welcome.html)
- <https://aws.amazon.com/fr/lambda/>
- <https://whatis.techtarget.com/definition/event-driven-computing>
- <https://www.slideshare.net/AmazonWebServices/deep-dive-on-aws-lambda>
- <https://www.slideshare.net/AmazonWebServices/intro-to-aws-lambda-london-loft>
- <https://aws.amazon.com/fr/lambda/features/>
- <https://www.contino.io/files/Contino-Introduction-to-Serverless-Computing-with-AWS-Lambda.pdf>