



# Services d'automatisation & construction d'architecture découplée

Par Lahda Biassou Alphonsine



# Lahda Biassou Alphonsine

## Ingénieure cloud et formatrice





# Plan

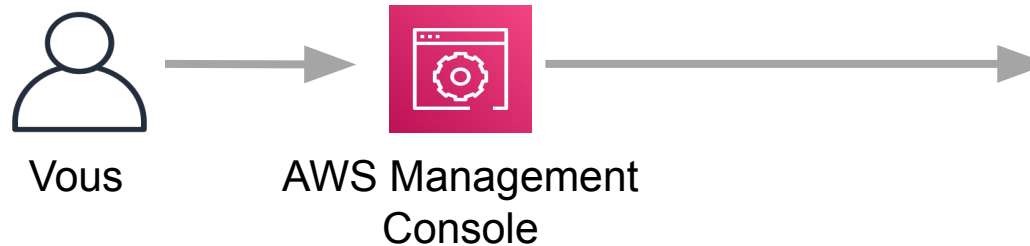
- **Raison d'automatiser**
- **Automatisation de vos infrastructures**
- **Automatisation de vos déploiement**
- **AWS Elastic Beanstalk**
- **Amazon SQS**
- **Amazon MQ**
- **AWS Step Function**





# Raison d'automatiser - sans automatisation

Long processus **manuel** d'élaboration d'une architecture



AWS Cloud

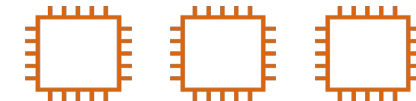
Configure data storage



Route your network



Create your instances

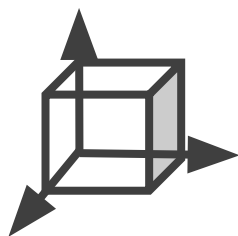


Build your databases





## Les risques d'un processus manuel



### Ne prend pas en charge la répétabilité à l'échelle

- Comment répliquer les déploiements dans plusieurs régions ?



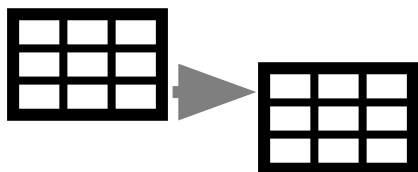
### Pas de contrôle de version

- Comment revenir à une version antérieure de l'environnement de production ?



### Absence de pistes d'audit

- Comment allez-vous garantir la conformité ? Comment allez-vous suivre les modifications apportées aux détails de la configuration au niveau des ressources ?



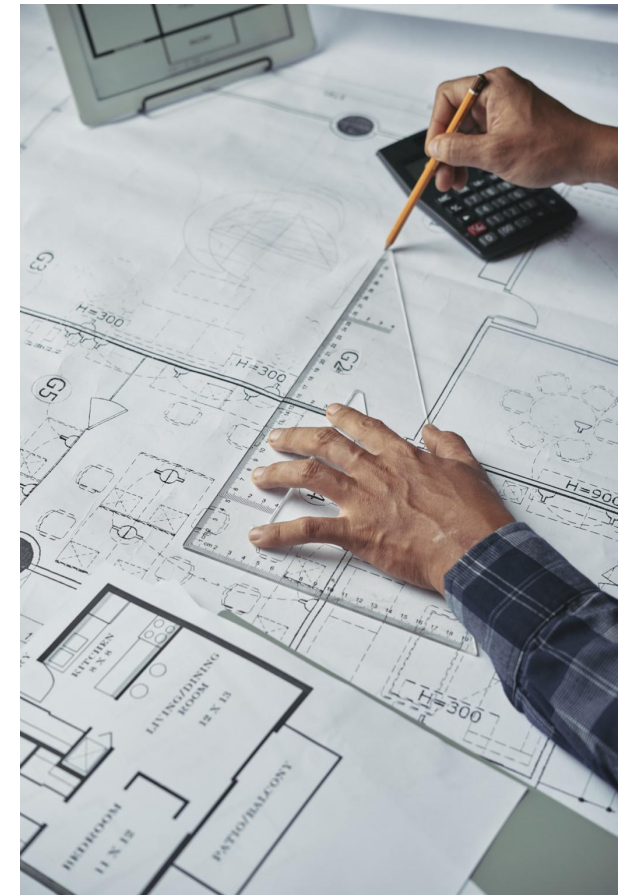
### Gestion incohérente des données

- Par exemple, comment garantissez-vous la concordance des configurations entre plusieurs instances Amazon Elastic Compute Cloud (Amazon EC2) ?



## Respecter les principes du cadre bien architecturé d'AWS

- Principes de conception de l'excellence opérationnelle
  - Effectuer des opérations en tant que code
  - Effectuer des changements fréquents, modestes et réversibles
- Principes de conception du pilier fiabilité
  - Gérer le changement dans l'automatisation





# Plan

- Raison d'automatiser
- **Automatisation de vos infrastructures**
- Automatisation de vos déploiement
- **AWS Elastic Beanstalk**
- **Amazon SQS**
- **Amazon MQ**
- **AWS Step Function**





# Automatisation de votre infrastructure



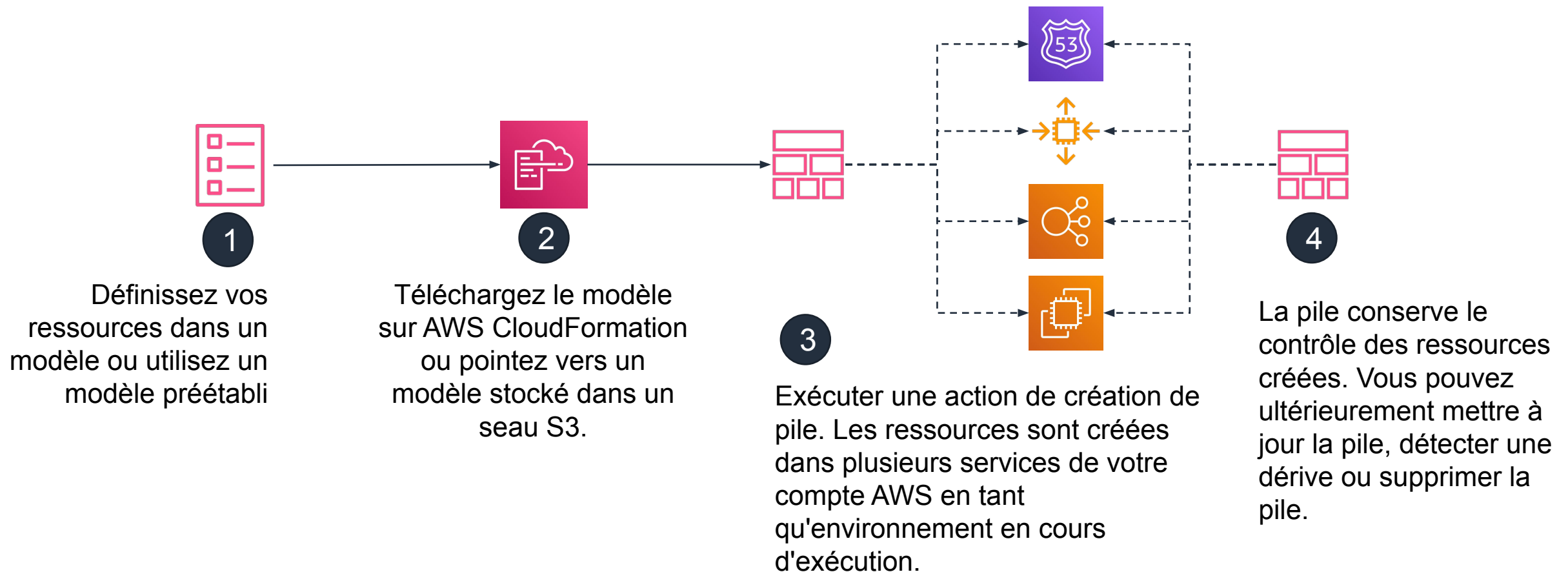
AWS  
CloudFormation

- AWS CloudFormation offre un moyen simplifié de modéliser, de créer et de gérer une collection de ressources AWS
- La collection de ressources est appelée pile AWS CloudFormation.
- Pas de frais supplémentaires (vous ne payez que pour les ressources que vous créez)
- Possibilité de créer, mettre à jour et supprimer des piles
- Permet un provisionnement et une mise à jour ordonnés et prévisibles des ressources
- Permet le contrôle des versions des déploiements de ressources AWS





# AWS CloudFormation -vue globale

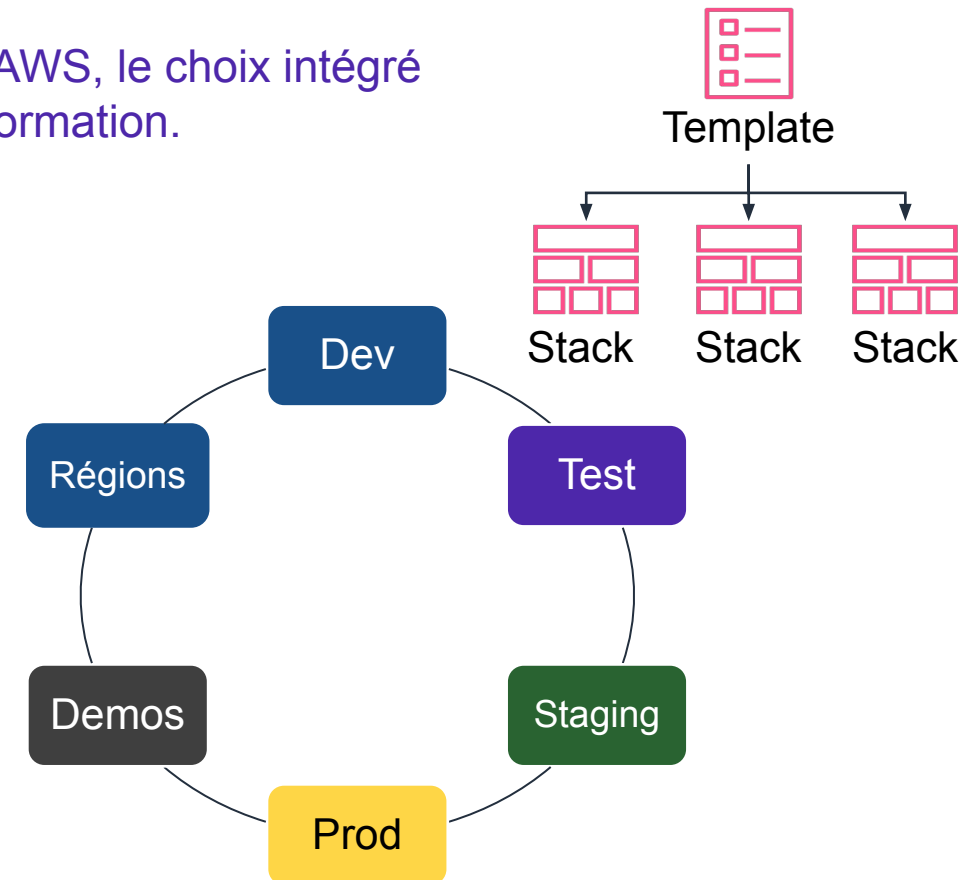




# Infrastructure as a code (IaC)

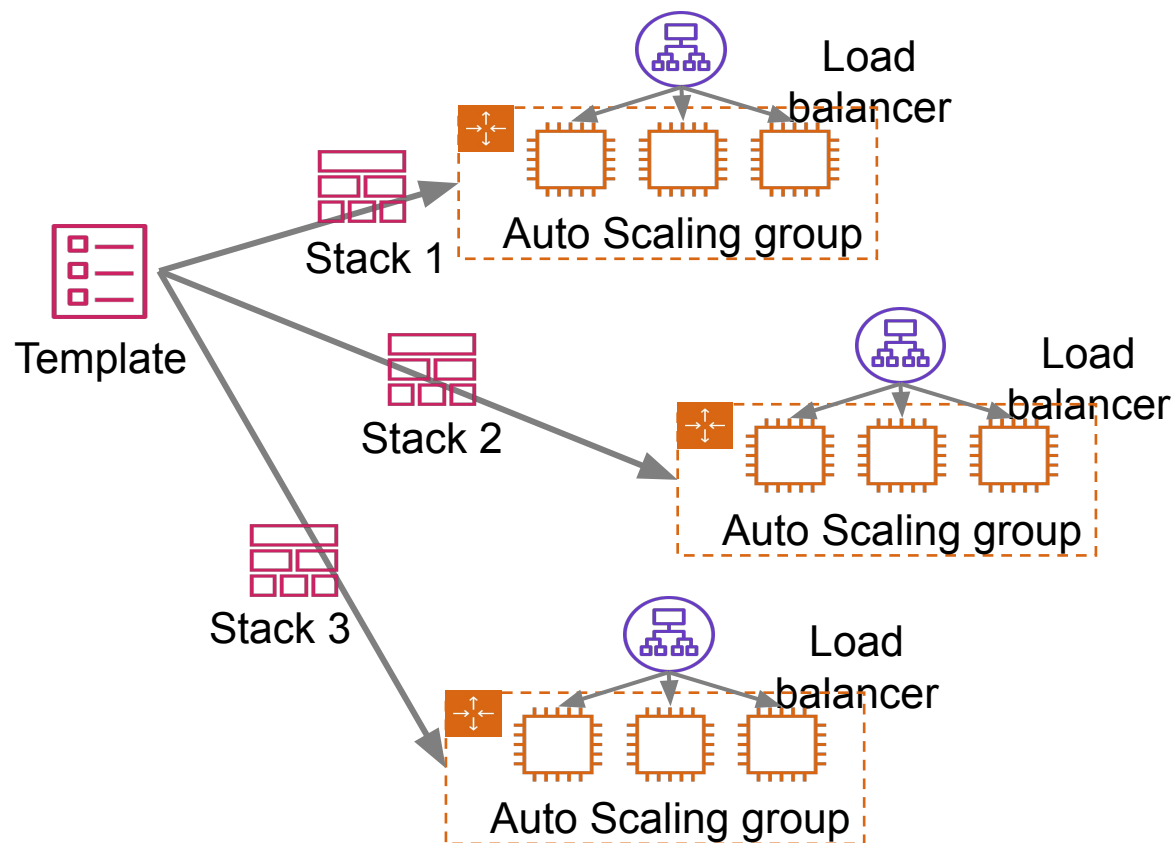
Pour le développement dans le nuage AWS, le choix intégré pour l'IaC est AWS CloudFormation.

- L'IaC est le processus de provisionnement et de gestion de vos ressources en nuage par l'écriture d'un fichier modèle -
  - Lisible par l'homme
  - Consommable par la machine
- Il s'agit d'une infrastructure que vous pouvez répliquer, redéployer et réorienter.
- Vous pouvez revenir au dernier état correct en cas de défaillance.





# Infrastructure as a code (IaC) -avantages



## Réduction du nombre d'environnements correspondants

- Déploiement rapide d'environnements complexes
- Cohérence de la configuration
- Nettoyage simple en cas de besoin (la suppression de la pile efface les ressources créées)
- Facilité de propagation d'un changement à toutes les piles
- Modifier le modèle, exécuter la mise à jour de la pile sur toutes les piles.

## Avantages

- Réutilisation
- Répétabilité
- Maintenabilité



# Syntaxe du modèle AWS CloudFormation

- **Modèles AWS CloudFormation**
  - Auteur en JavaScript Object Notation (JSON) ou YAML Ain't Markup Language (YAML)
- **Avantages YAML –**
  - Moins verbeux (pas de caractères {}, "", ,)
  - Prend en charge les commentaires intégrés
- **Avantages JSON –**
  - Plus largement utilisé par d'autres systèmes informatiques (par exemple, les API)
- **Recommandation – Traiter les modèles comme du code source**
  - Les stocker dans un référentiel de code



```
{
  "AWSTemplateFormatVersion":
    "2010-09-09",
  "Resources" : {
    "awsexamplebucket1" : {
      "Type" : "AWS::S3::Bucket"
    }
  }
}
```

JSON  
example

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  awsexamplebucket1:
    Type: AWS::S3::Bucket
```

YAML  
example

Les modèles peuvent également être créés dans AWS **CloudFormation Designer**, une interface de conception graphique dans la console de gestion AWS



## Template simple: création d'une instance Ec2

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create EC2 instance",
  "Parameters": {
    "KeyPair": {
      "Description": "SSH Key Pair",
      "Type": "String"
    }
  },
  "Resources": {
    "Ec2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "ami-9d23aeea",
        "InstanceType": "m3.medium",
        "KeyName": {"Ref": "KeyPair"}
      }
    }
  },
  "Outputs": {
    "InstanceId": {
      "Description": "InstanceId",
      "Value": {"Ref": "Ec2Instance"}
    }
  }
}
```

← **Paramètres** – Spécifiez les valeurs qui peuvent être définies au moment de l'exécution lorsque vous créez la pile

- Exemples d'utilisation : paramètres spécifiques à la région ou paramètres d'environnement de production ou de test

← **Ressources** – Définir ce qui doit être créé dans le compte AWS

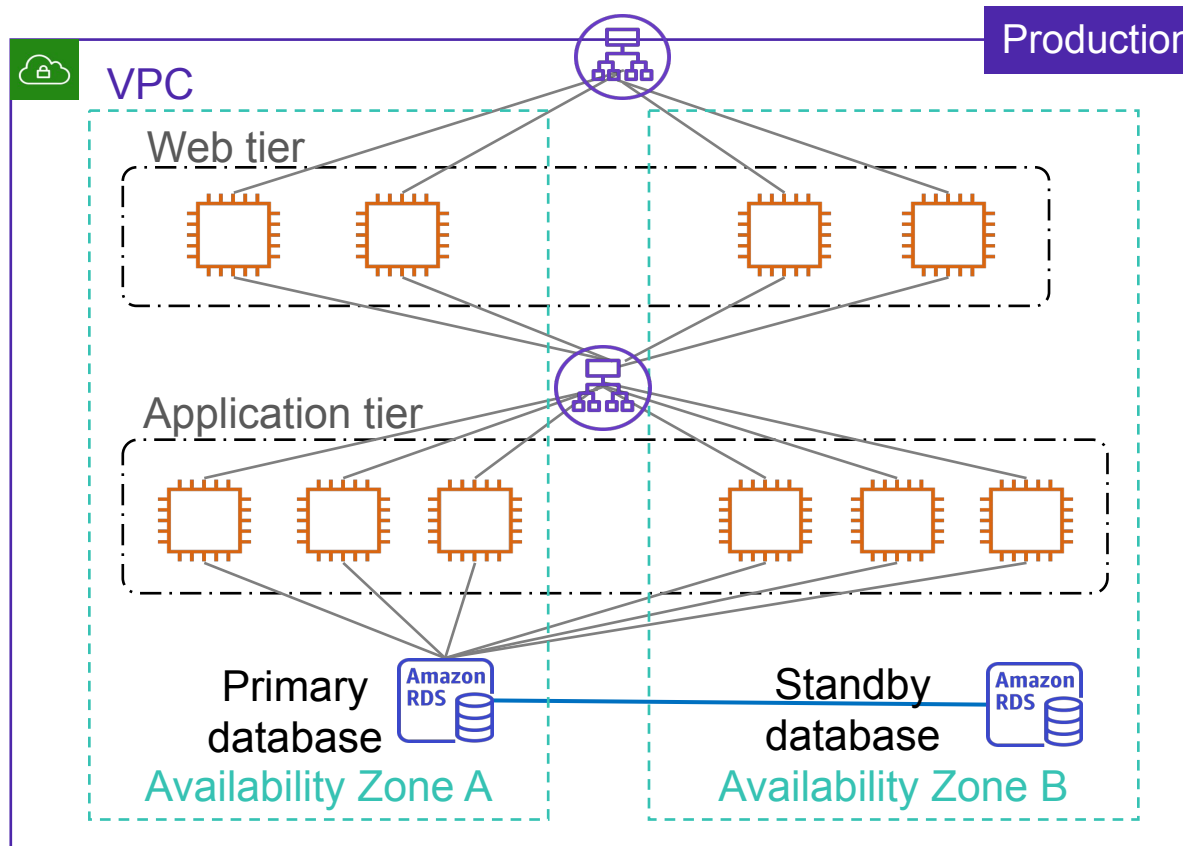
- Exemple : créer tous les composants d'un cloud privé virtuel (VPC) dans une région, puis créer des instances EC2 dans le VPC
- Peut référencer des paramètres

← **Sorties** – Spécifiez les valeurs renvoyées après la création de la pile

- Exemple d'utilisation : Renvoyer l'instanceId ou l'adresse IP publique d'une instance EC2.



# AWS CloudFormation

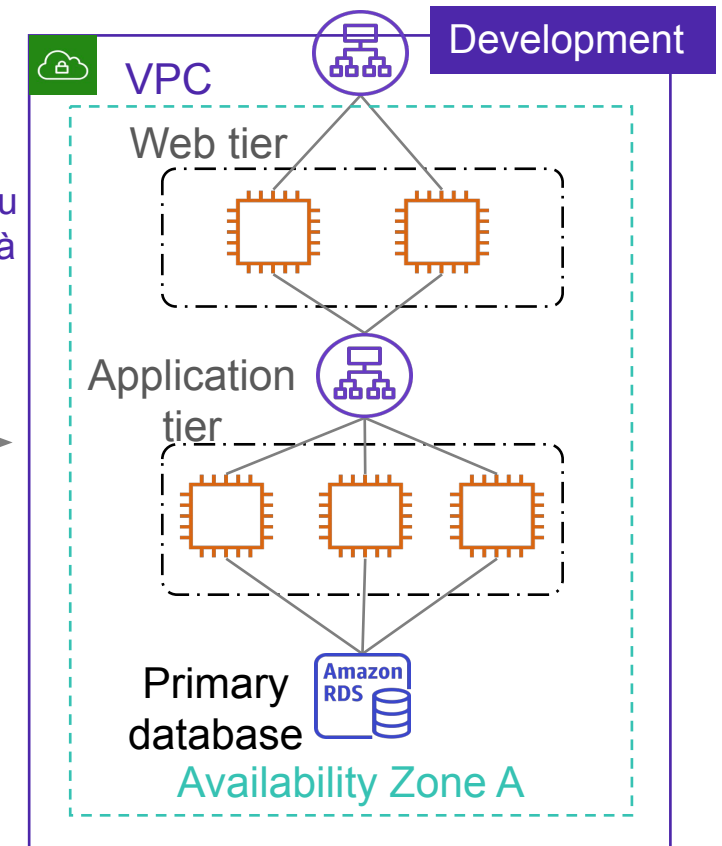


Besoin de mettre à jour une configuration ?

Modifiez les détails du modèle, puis mettez à jour les piles.



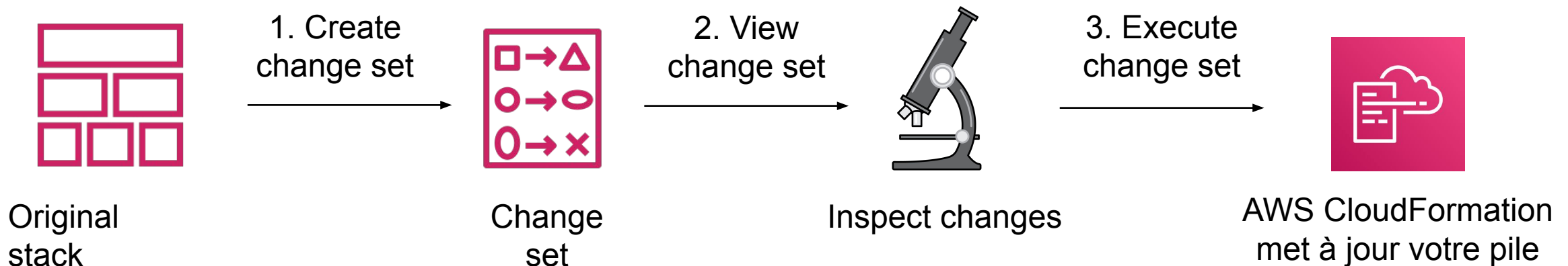
Un seul modèle AWS CloudFormation avec des conditions définies peut déployer une infrastructure d'application différente dans différents environnements.





# AWS CloudFormation - change sets

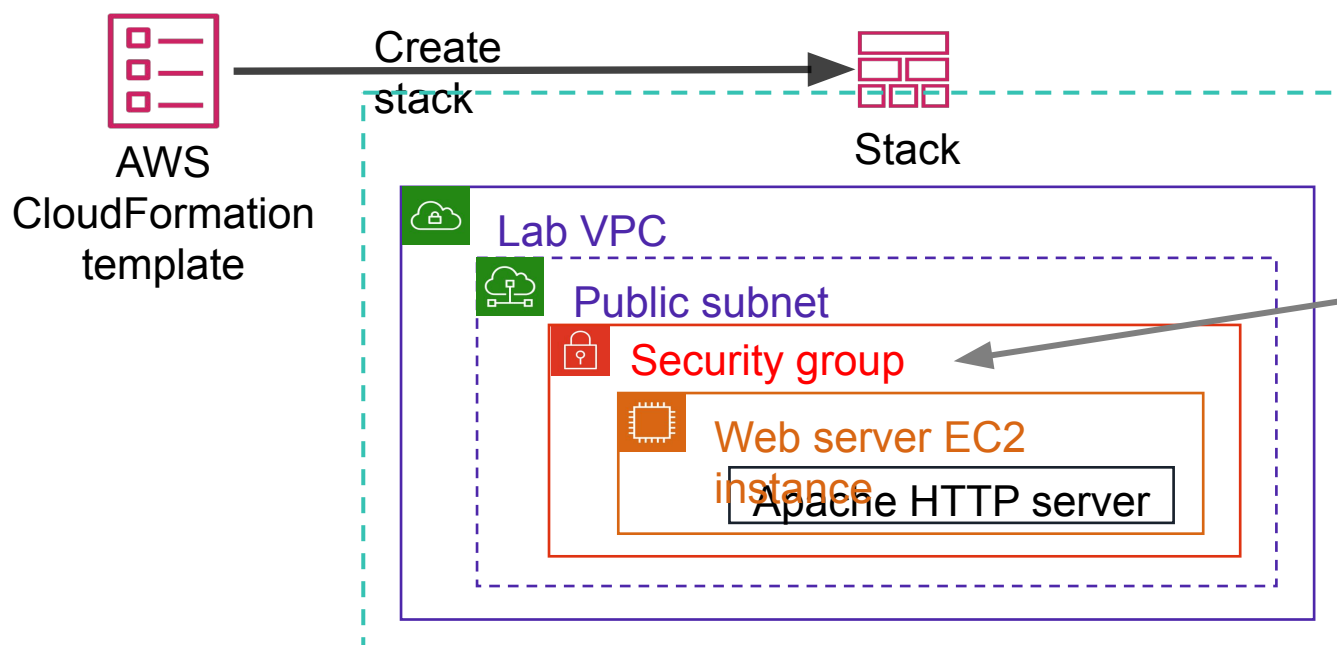
Les ensembles de modifications vous permettent de prévisualiser les modifications avant de les implémenter.



Utilisez l'attribut DeletionPolicy pour conserver ou sauvegarder une ressource lorsque sa pile est supprimée ou mise à jour.



# Drift detection



Scénario:

1. Un environnement d'application est créé par une pile AWS CloudFormation.
2. Plus tard, quelqu'un modifie manuellement le groupe de sécurité et ouvre un nouveau port TCP entrant.
3. La détection de dérive est exécutée sur la pile.
4. Toutes les ressources, à l'exception du groupe de sécurité, affichent le résultat IN\_SYNC, mais le groupe de sécurité affiche un statut MODIFIED, avec des détails.

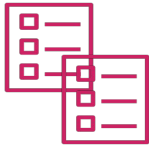
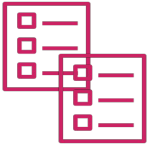
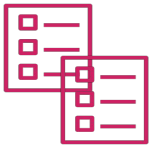
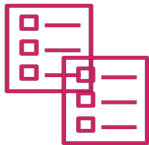
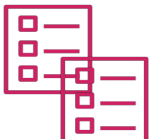
**Question :** Dans ce scénario, quelle serait la meilleure approche si l'équipe souhaite modifier le paramètre du groupe de sécurité ?

**Réponse :** Modifiez les paramètres du groupe de sécurité du modèle AWS CloudFormation. Ensuite, exécutez Update Stack. AWS CloudFormation mettra à jour le groupe de sécurité. Le déploiement du modèle reste synchronisé avec le déploiement réel.



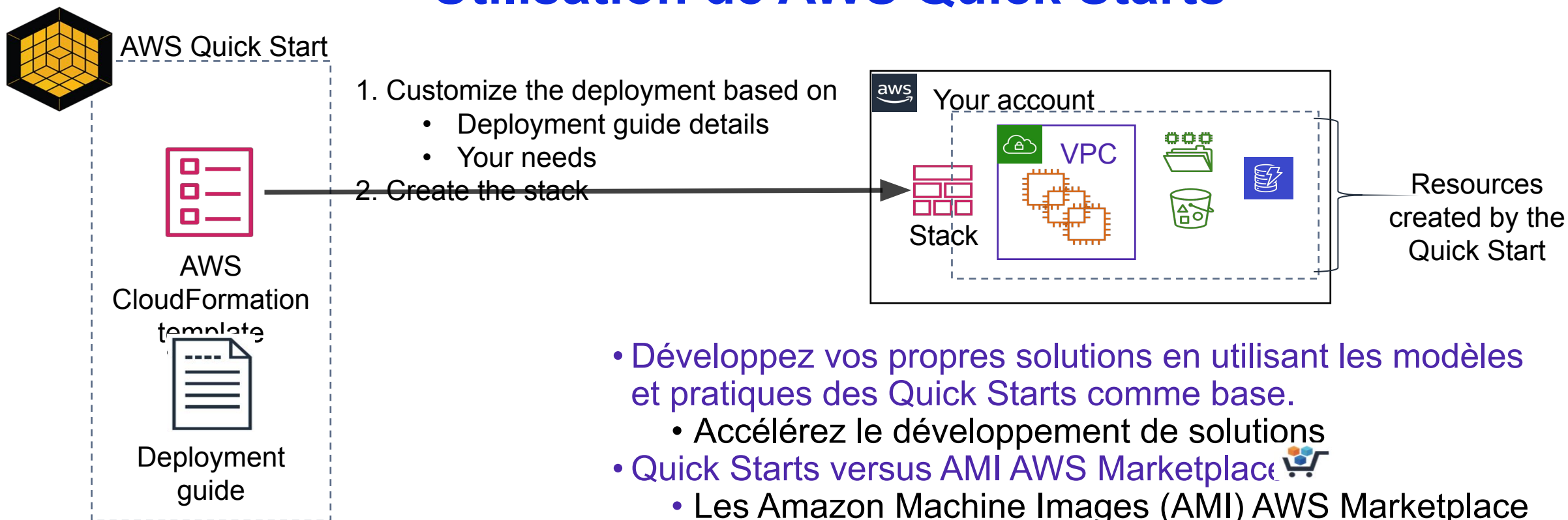


# Scoping et Modèles d'organisation

Services Frontend		Web interfaces, mobile access, analytics dashboard
Services Backend		Search, payments, reviews, recommendations
Services partagés		Customer relationship management (CRM) databases, common monitoring, alarms, subnets, security groups
Réseau		VPCs, internet gateways, virtual private networks (VPNs), Network Address Translation (NAT) devices
Sécurité		AWS Identity and Access Management (IAM) policies, users, groups, and roles



# Utilisation de AWS Quick Starts



- Développez vos propres solutions en utilisant les modèles et pratiques des Quick Starts comme base.
  - Accélérez le développement de solutions
- Quick Starts versus AMI AWS Marketplace 
  - Les Amazon Machine Images (AMI) AWS Marketplace sont des solutions à fournisseur unique basées sur Amazon EC2, sans personnalisation
  - Les Quick Starts sont modulaires et personnalisables



# Plan

- Raison d'automatiser
- Automatisation de vos infrastructures
- **Automatisation de vos déploiement**
- **AWS Elastic Beanstalk**
- **Amazon SQS**
- **Amazon MQ**
- **AWS Step Function**

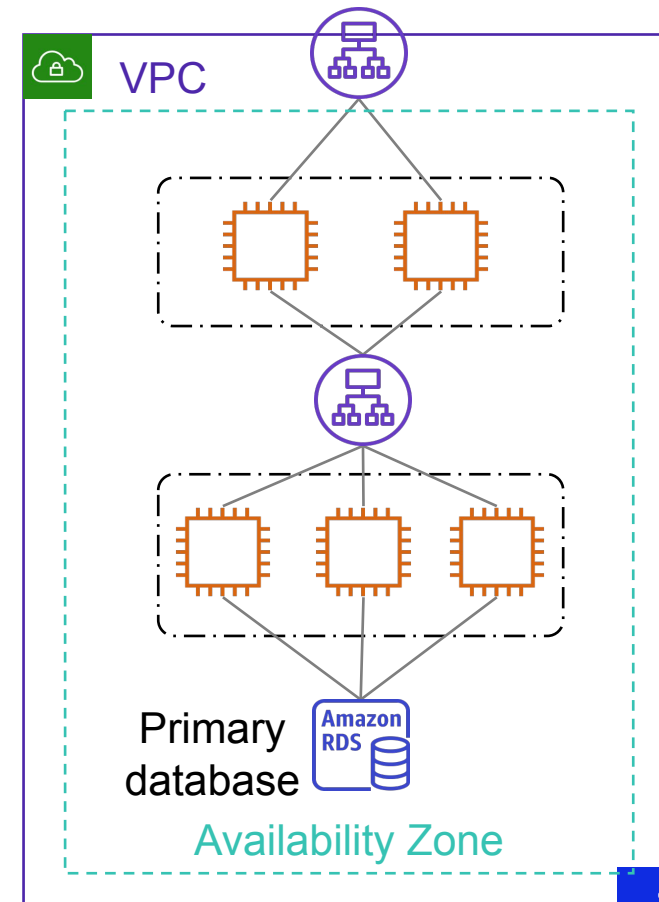




## Automatisation de vos déploiement - Comment allez-vous maintenir votre flotte à jour ?



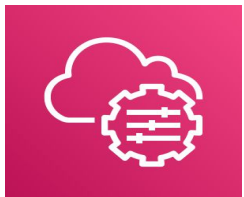
Vous pouvez avoir des centaines d'instances à gérer.  
Comment appliquer les correctifs du système d'exploitation invité et mettre à jour les logiciels qui y sont installés ?





# AWS System Manager

*Gain operational insights and  
take action on AWS resources.*

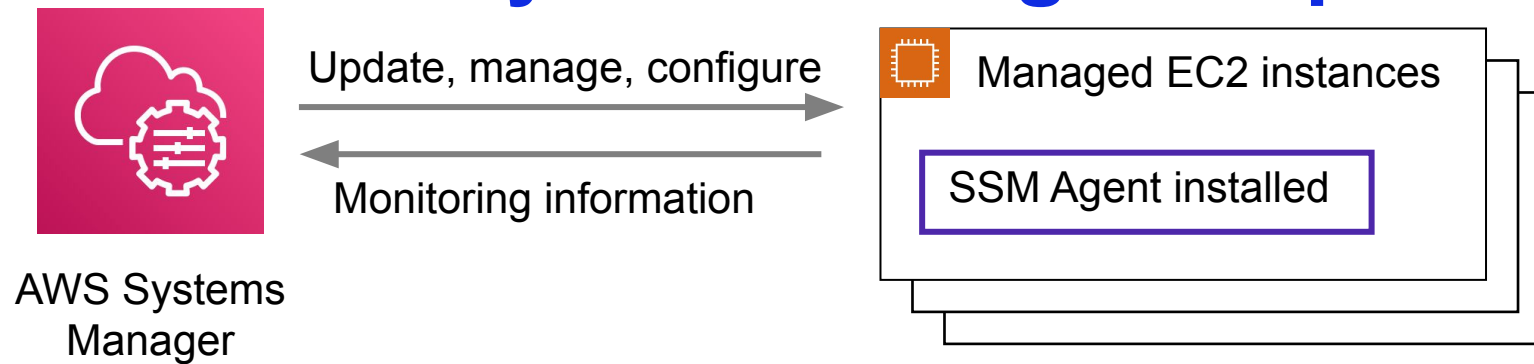


AWS Systems  
Manager

- Automatise les tâches opérationnelles
  - Exemple : appliquer des correctifs de système d'exploitation et des mises à niveau logicielles sur un parc d'instances EC2
- Simplifie la gestion des ressources et des applications
  - Gère l'inventaire des logiciels
  - Affiche les configurations système détaillées sur l'ensemble du parc
- Gère les serveurs sur site et dans le cloud



# AWS System Manager -capacités



Run Command



Maintenance Windows



Parameter Store



Patch Manager



State Manager



Automation



Session Manager



Inventory



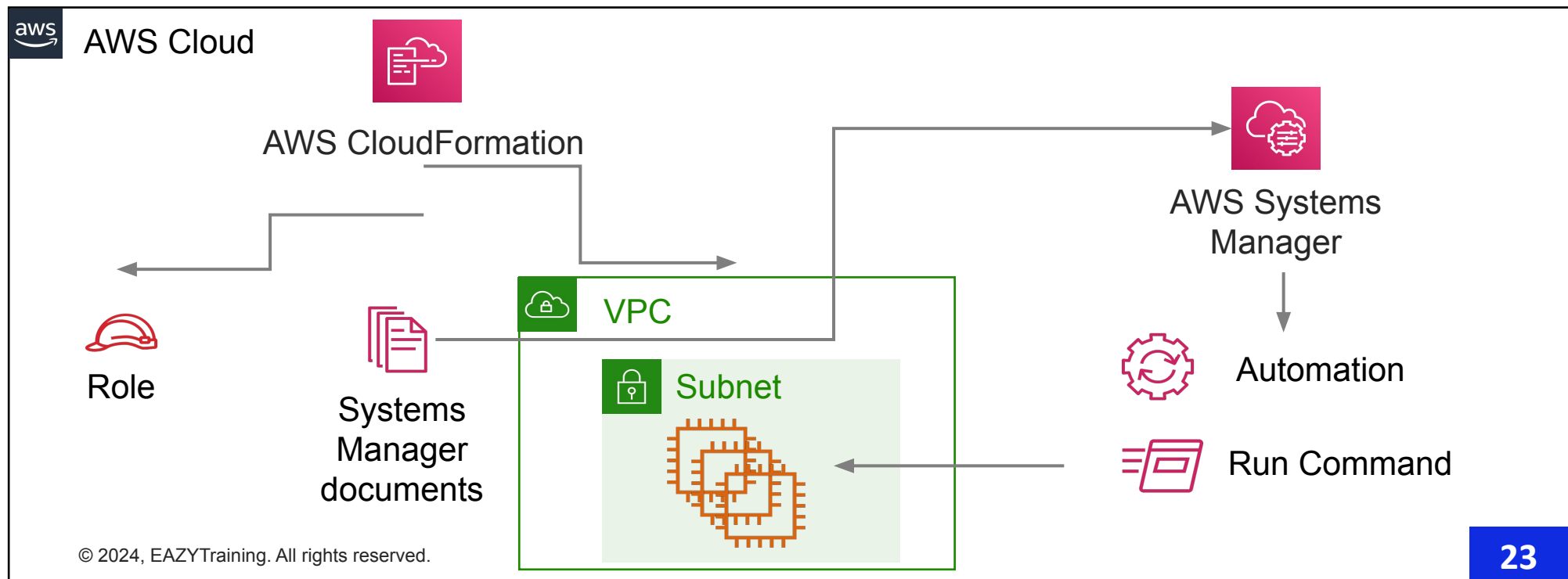
Documents



# AWS CloudFormation et Systems Manager se complètent

AWS CloudFormation fonctionne bien pour définir les ressources AWS Cloud.

Systems Manager fonctionne bien pour l'automatisation au sein des systèmes d'exploitation invités.





# Plan

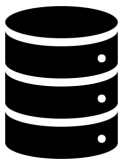
- Raison d'automatiser
- Automatisation de vos infrastructures
- Automatisation de vos déploiement
- **AWS Elastic Beanstalk**
- Amazon SQS
- Amazon MQ
- AWS Step Function







# AWS Elastic Beanstalk -challenges generals



La gestion de l'infrastructure autour du déploiement d'applications peut être difficile



La gestion et la configuration des serveurs peuvent prendre du temps.



Vous pourriez avoir un manque de cohérence entre plusieurs projets ou applications



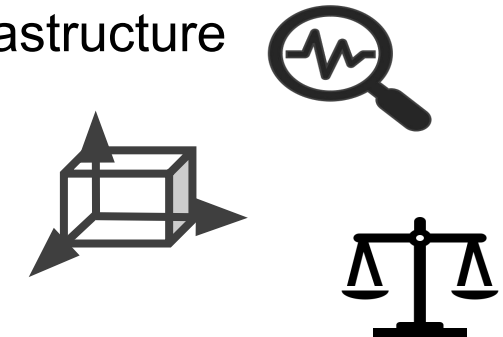


## AWS Elastic Beanstalk



AWS Elastic  
Beanstalk

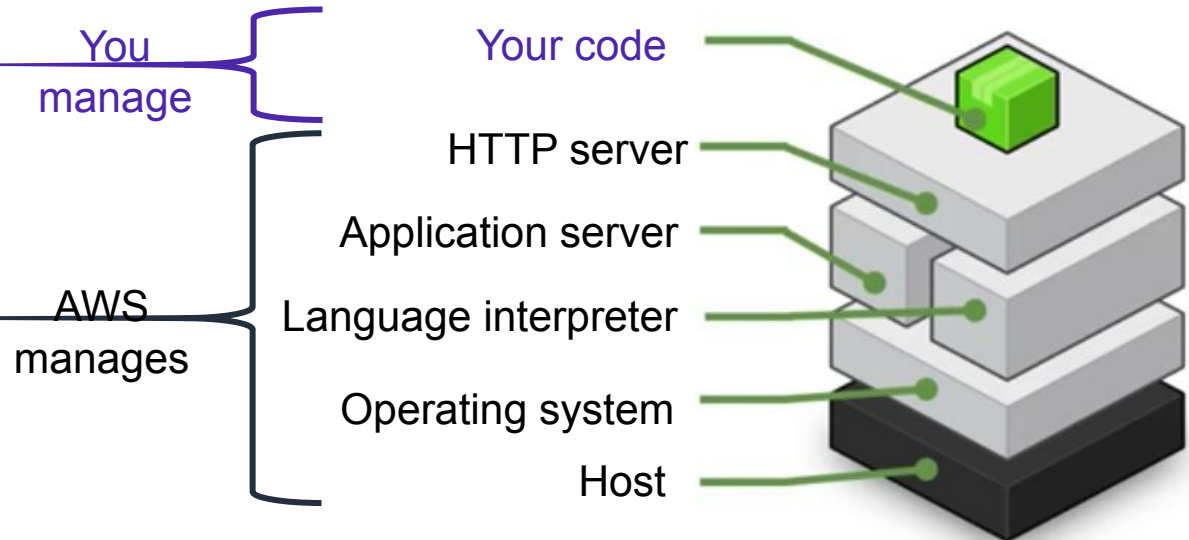
- Un moyen simple de mettre en place et d'exécuter des applications Web.
- Service géré qui gère automatiquement –
  - Provisionnement et configuration de l'infrastructure
  - Déploiement
  - Équilibrage de charge
  - Mise à l'échelle automatique
  - Surveillance de l'état de santé
  - Analyse et débogage
  - Journalisation
- Aucun frais supplémentaire pour son utilisation
  - Payez uniquement pour les ressources sous-jacentes utilisées.





# AWS Elastic Beanstalk deployment

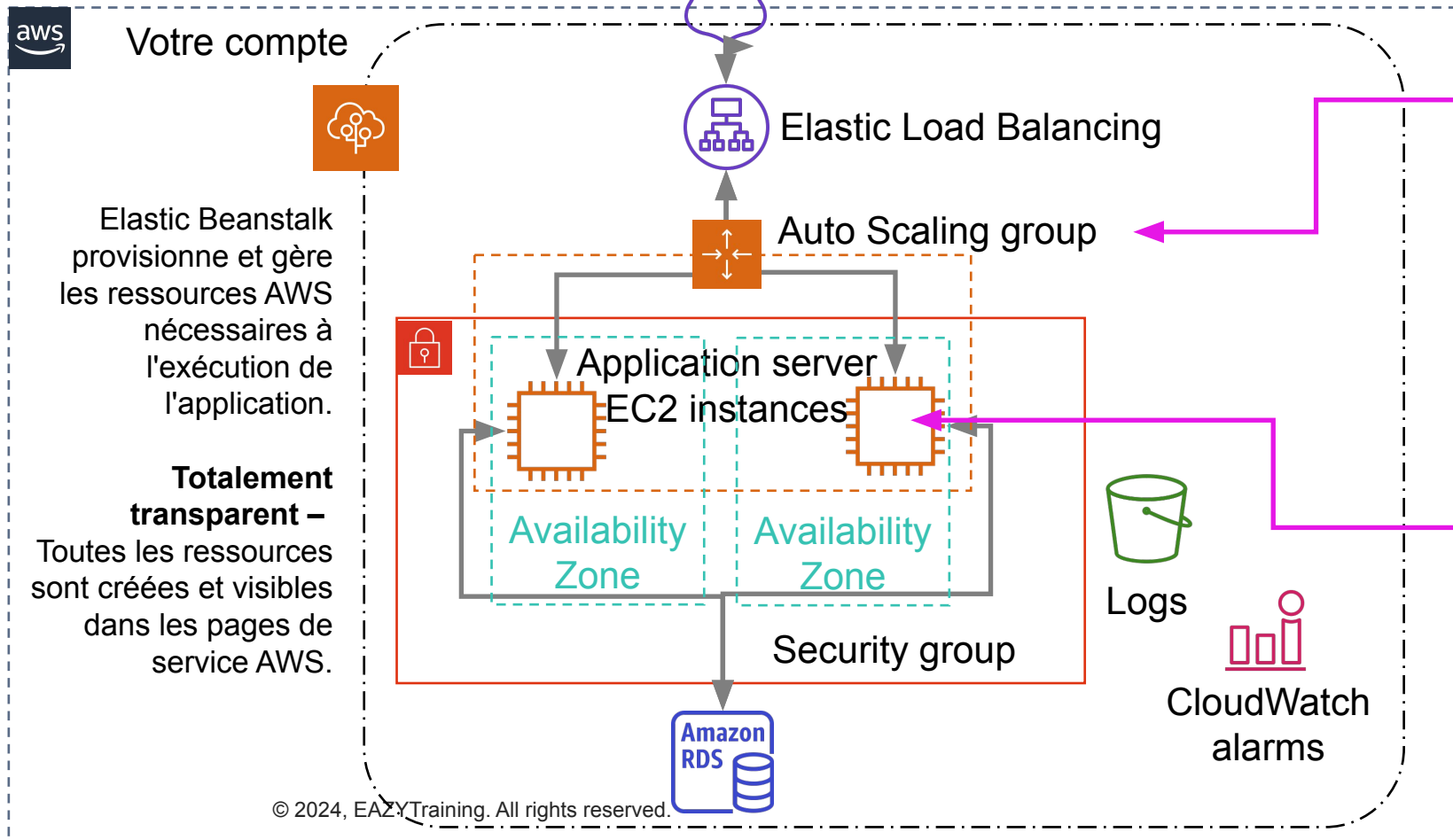
- prend en charge les applications Web écrites pour les plateformes courantes
  - Java, .NET, PHP, Node.js, Python, Ruby, Go et Docker
- Vous téléchargez votre code
  - Elastic Beanstalk gère automatiquement le déploiement
  - Déploiement sur des serveurs tels qu'Apache, NGINX, Passenger, Puma et Microsoft Internet Information Services (IIS).





# AWS Elastic Beanstalk application environment

<http://<your-application>.elasticbeanstalk.com>



Automatically scales to meet your application's needs.

Example Elastic Beanstalk environment where your web application code can run:

**c605-samp-1LC0VOOT45NYX**

---

**Environment tier:** Web Server

**Platform:** Tomcat 8.5 with Java 8 running on 64bit Amazon Linux/3.3.0

**Running versions:** c6053a99747u340397t1w450372118841-sampleapplicationversion-gpu2fua37td0

**Last modified:** 2019-10-24 12:21:08 UTC-0700

**URL:** c605-samp-1LC0VOOT45NYX.mqx9akx6sr.us-east-1.elasticb...

**Health status:** Green



# Choisir la solution adéquate d'automatisation

Higher-level services

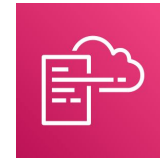
Faire par vous même



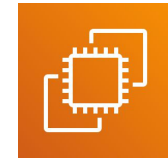
AWS Elastic  
Beanstalk



AWS  
OpsWorks



AWS  
CloudFormation



Amazon  
EC2

Commodité

Contrôle



# Plan

- Raison d'automatiser
- Automatisation de vos infrastructures
- Automatisation de vos déploiement
- AWS Elastic Beanstalk
- **Amazon SQS**
- Amazon MQ
- AWS Step Function

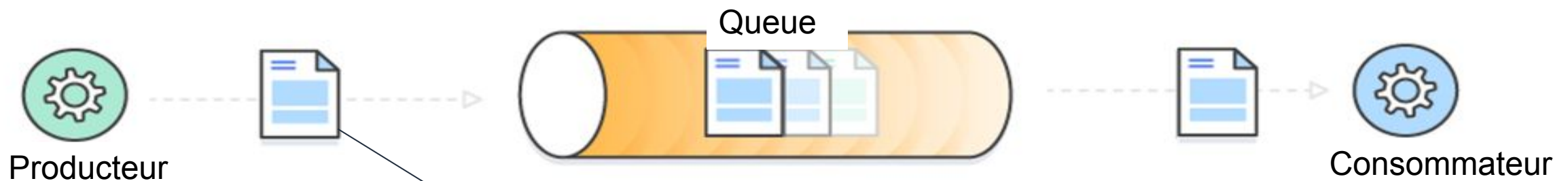




## Architecture découplée: Amazon SQS

Producteur – Composant d'application qui produit des messages et les ajoute à la file d'attente.

Consommateur – Composant d'application qui interroge la file d'attente pour les messages et les traite



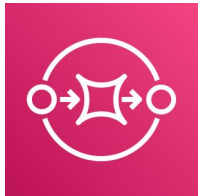
Un message est destiné à la communication entre des composants logiciels, et non entre des personnes (c'est-à-dire qu'il ne s'agit pas d'un e-mail ou d'un SMS).

*Pull mechanism*





## Amazon Simple Queue Service



Amazon Simple  
Queue Service  
(Amazon SQS)

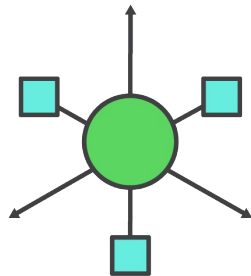
- Service de mise en file d'attente de messages entièrement géré
- Utilise un mécanisme d'extraction
- Les messages sont chiffrés et stockés jusqu'à ce qu'ils soient traités et supprimés
- Agit comme un tampon entre les producteurs et les consommateurs.





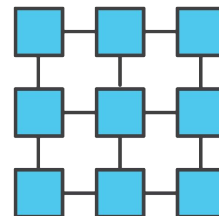
## Amazon Simple Queue Service - couplage faible

Avec Amazon SQS, vous pouvez:

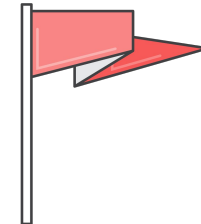


Utilisez le traitement asynchrone pour obtenir rapidement vos réponses à chaque étape

© 2024, EAZYTraining. All rights reserved.



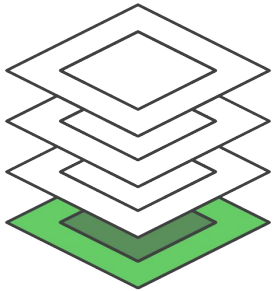
Gérez les exigences de performance et de service en augmentant le nombre d'instances de travail.



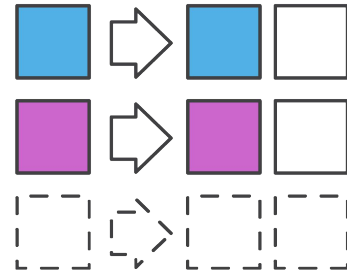
Récupérez facilement des étapes ayant échoué car les messages resteront dans la file d'attente.



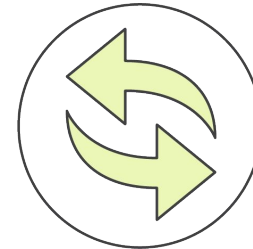
## Amazon Simple Queue Service - cas d'usage général



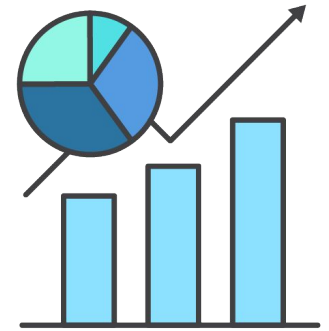
Work queues



Buffering batch operations



Request offloading

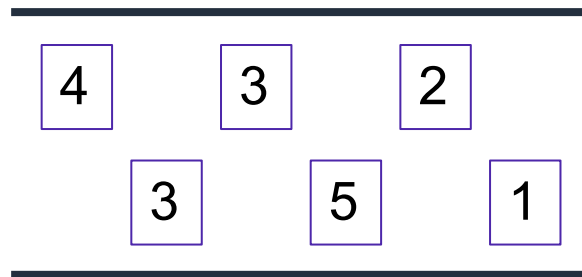


Trigger Amazon EC2 Auto Scaling



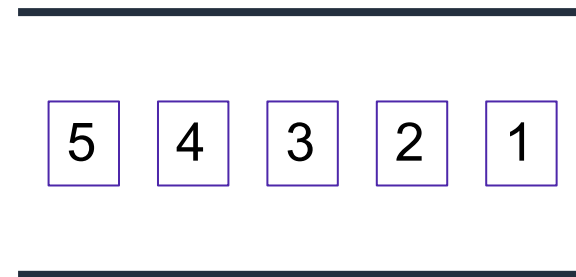
## Type de queue

### Standard queues



- Livraison au moins une fois
- Commande dans les meilleurs délais
- Débit quasi illimité

### First in, first out (FIFO) queues

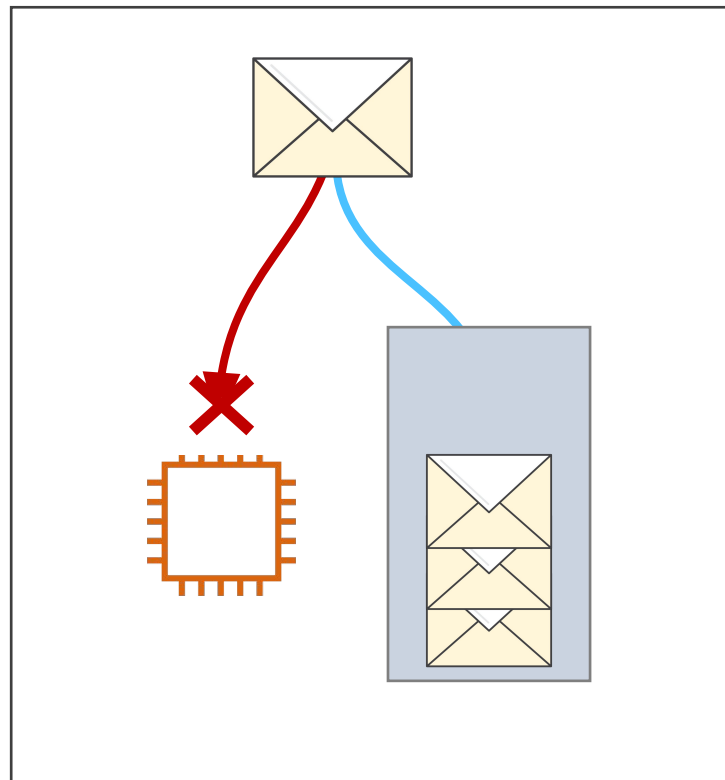


- Livraison premier entré, premier sorti
- Traitement en une seule fois
- Débit élevé



## Amazon SQS -caractéristiques (1/3)

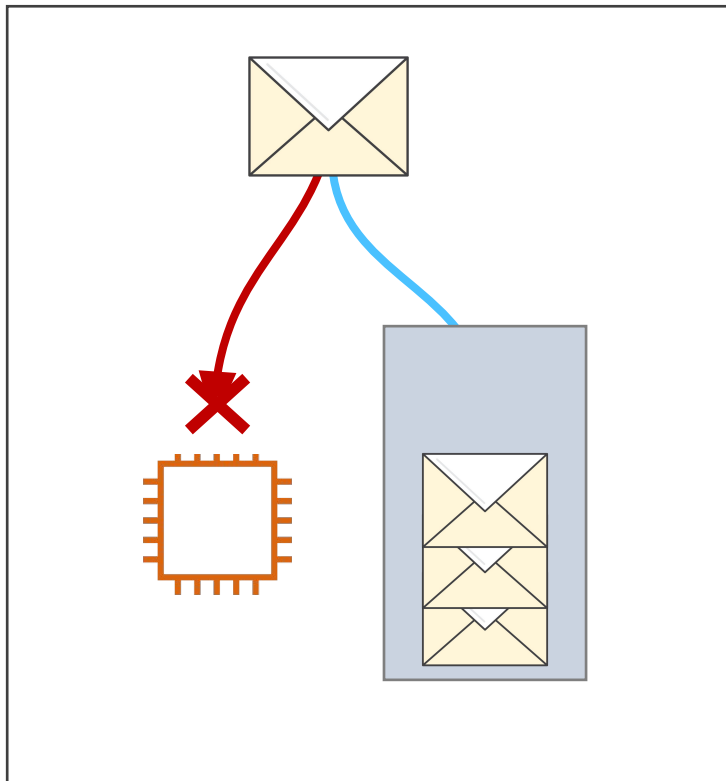
### Dead-letter queue support



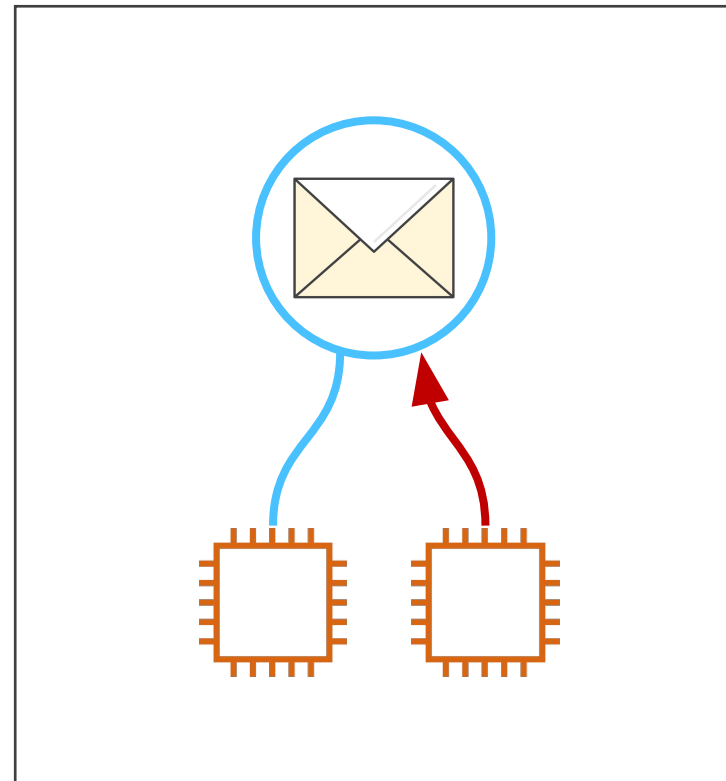


## Amazon SQS -caractéristiques (2/3)

Dead-letter queue support



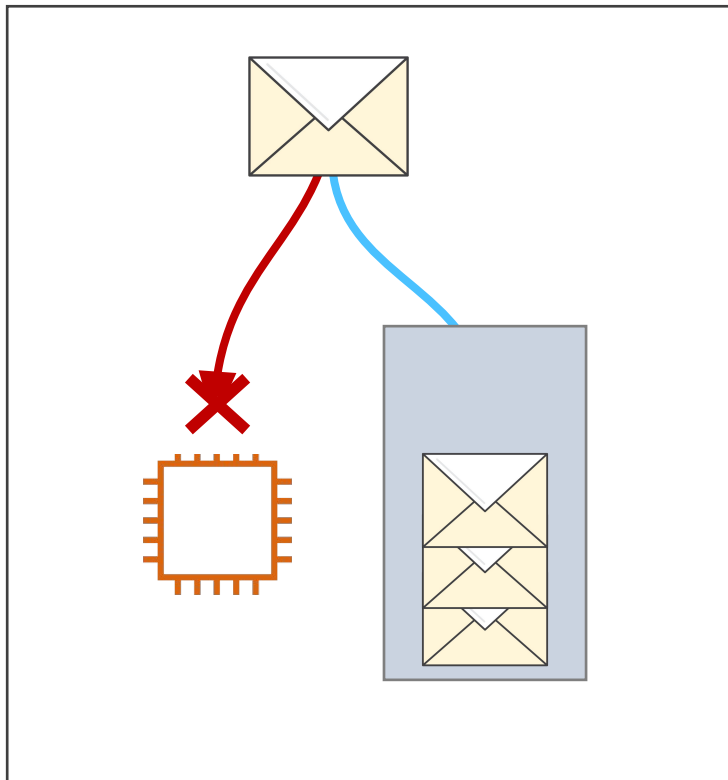
Visibility timeout



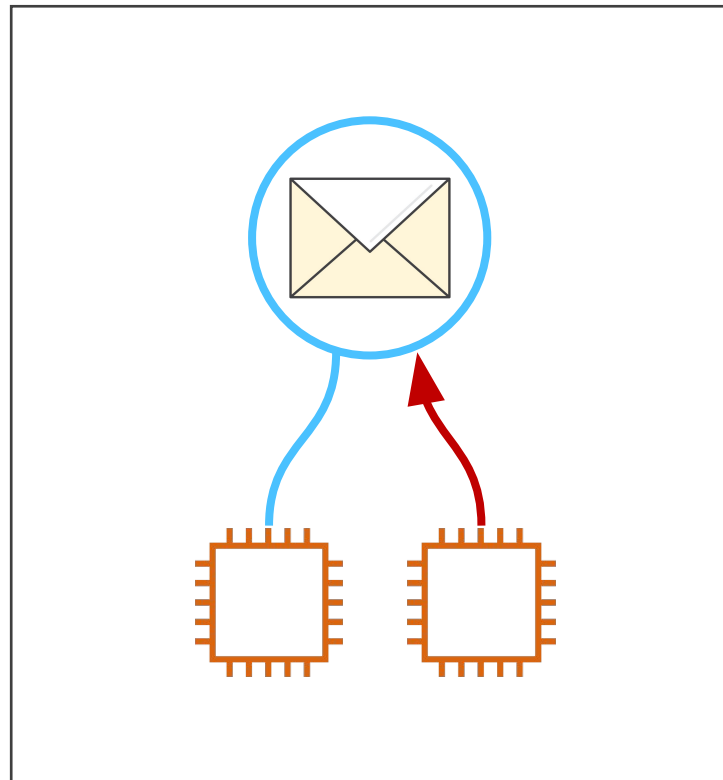


## Amazon SQS -caractéristiques (3/3)

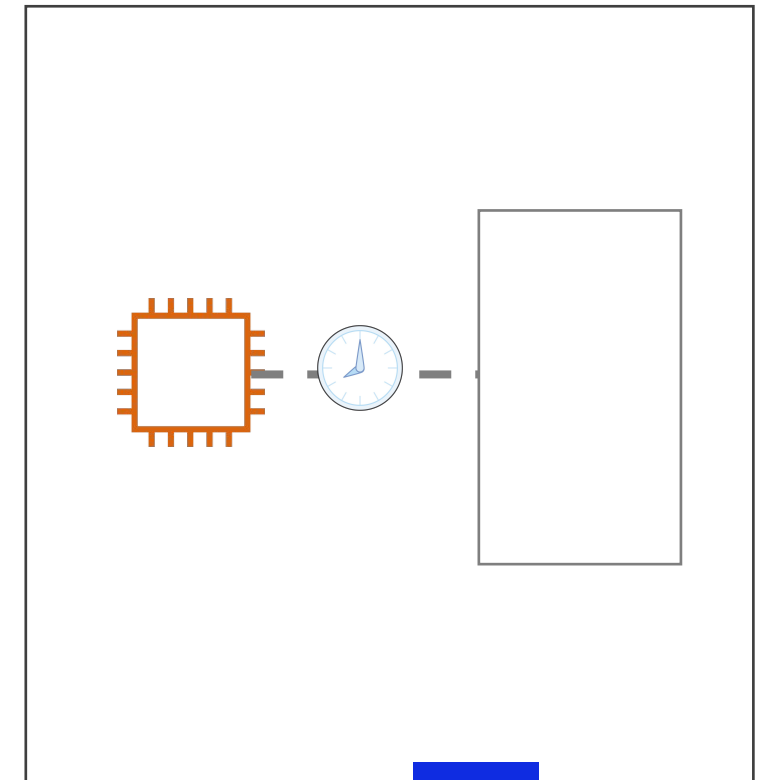
Dead-letter queue support



Visibility timeout

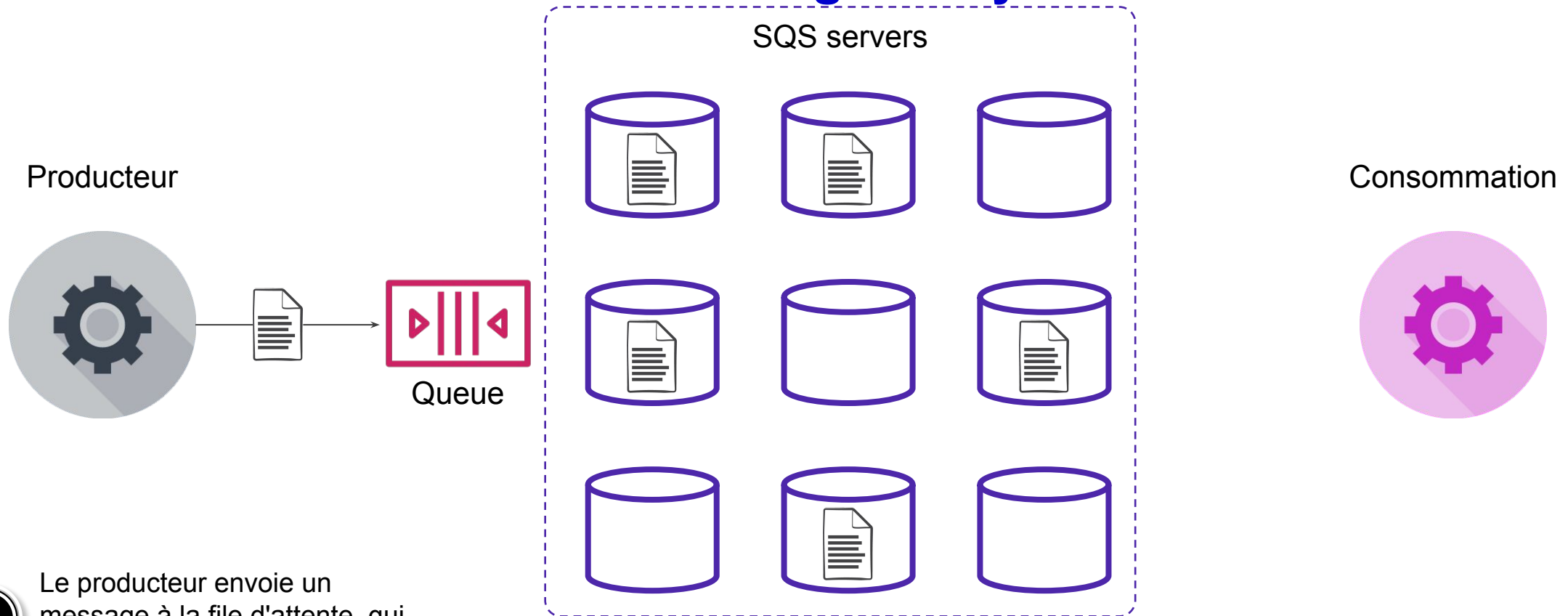


Long polling





## Amazon SQS message lifecycle: creer



1

Le producteur envoie un message à la file d'attente, qui est distribué de manière redondante.



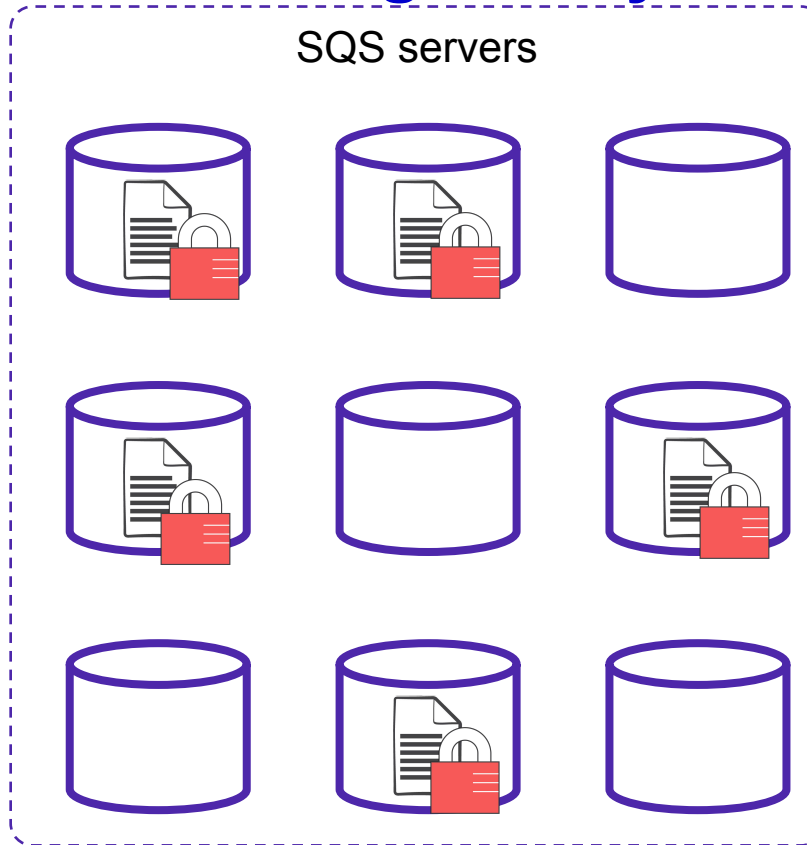
## Amazon SQS message lifecycle: processus

Producteur

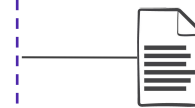
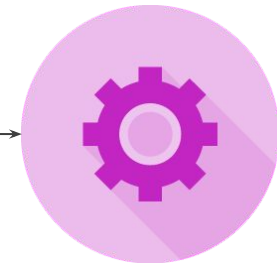


Queue

SQS servers

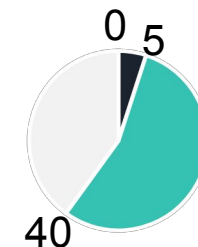


Consommateur



2

Le message est récupéré pour être traité par un consommateur et le délai de visibilité démarre.



Visibility timeout





## Amazon SQS message lifecycle: supprimer

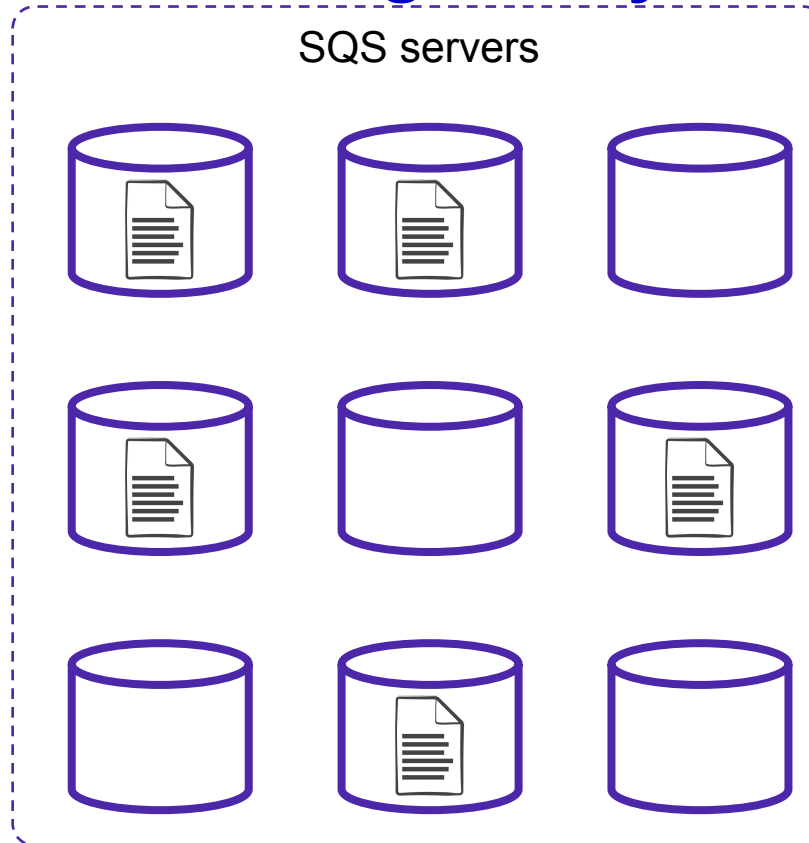
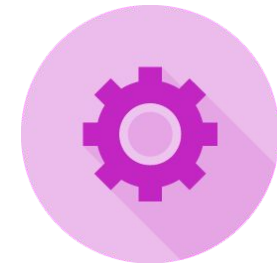
SQS servers

Producteur



Queue

Consommation

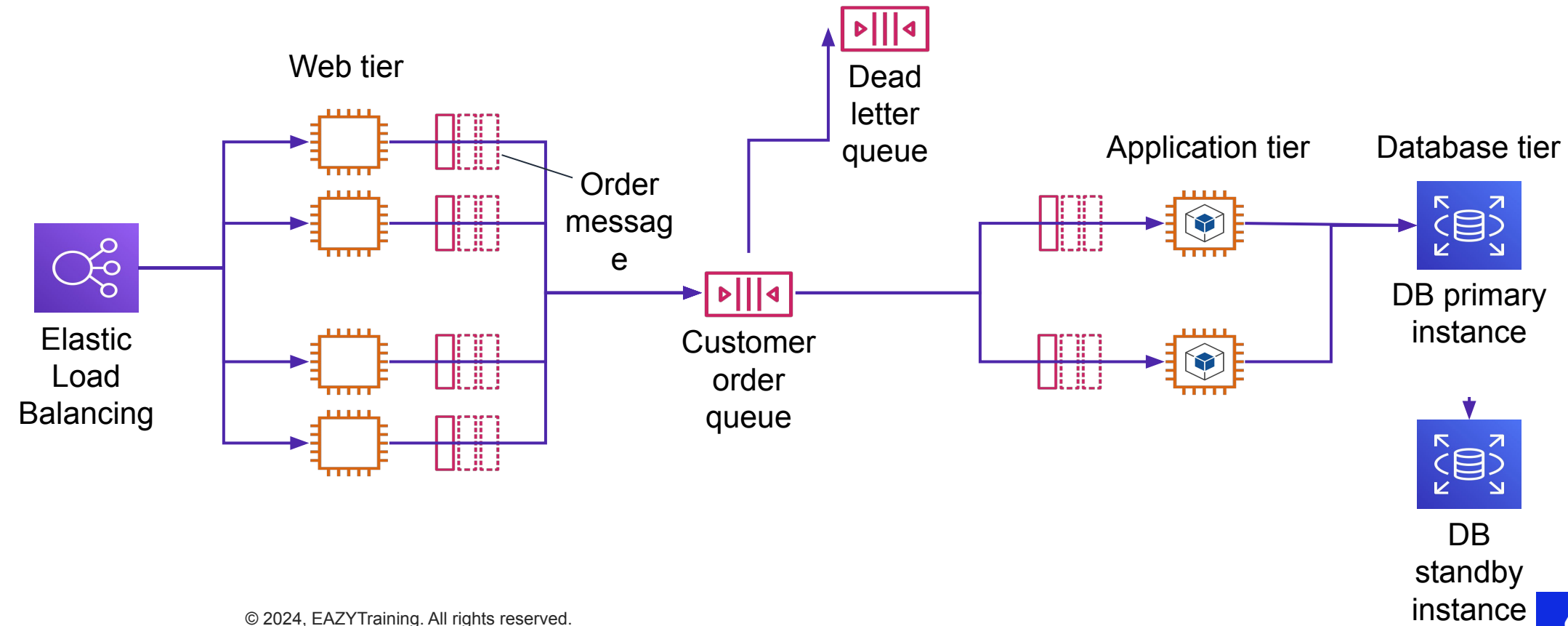


1

Le message est supprimé par le consommateur après avoir été traité.



## Amazon SQS: exemple





## Amazon SQS - cas d'usage



Communication de service à service



Éléments de travail asynchrones



Notifications de changement d'état



# Plan

- Raison d'automatiser
- Automatisation de vos infrastructures
- Automatisation de vos déploiement
- AWS Elastic Beanstalk
- Amazon SQS
- **Amazon MQ**
- AWS Step Function





## Envoi de messages entre applications cloud et sur site avec Amazon MQ



Amazon  
MQ

- Il s'agit d'un service de courtage de messages géré pour Apache ActiveMQ
- Gère l'approvisionnement, la configuration et la maintenance d'ActiveMQ
- Simplifie la migration des messages vers le cloud
- Est compatible avec les API et protocoles standard ouverts
- JMS, NMS, AMQP, STOMP, MQTT et WebSockets



## Amazon MQ - cas d'usage environnement hybride



Le producteur du message est sur site

Le consommateur de messages est basé sur le cloud



## Amazon MQ versus Amazon SQS et Amazon SNS

Amazon MQ	Amazon SQS and SNS
For <b>application migration</b>	For <b>born-in-the-cloud</b> applications
Protocols: JMS, NMS, AMQP, STOMP, MQTT, and WebSockets	Protocol: HTTPS
Feature-rich	Unlimited throughput
Pay per hour and pay per GB	Pay per request
Can do pub/sub	Cannot do pub/sub in Amazon SQS, but you can do pub/sub in Amazon SNS



# Plan

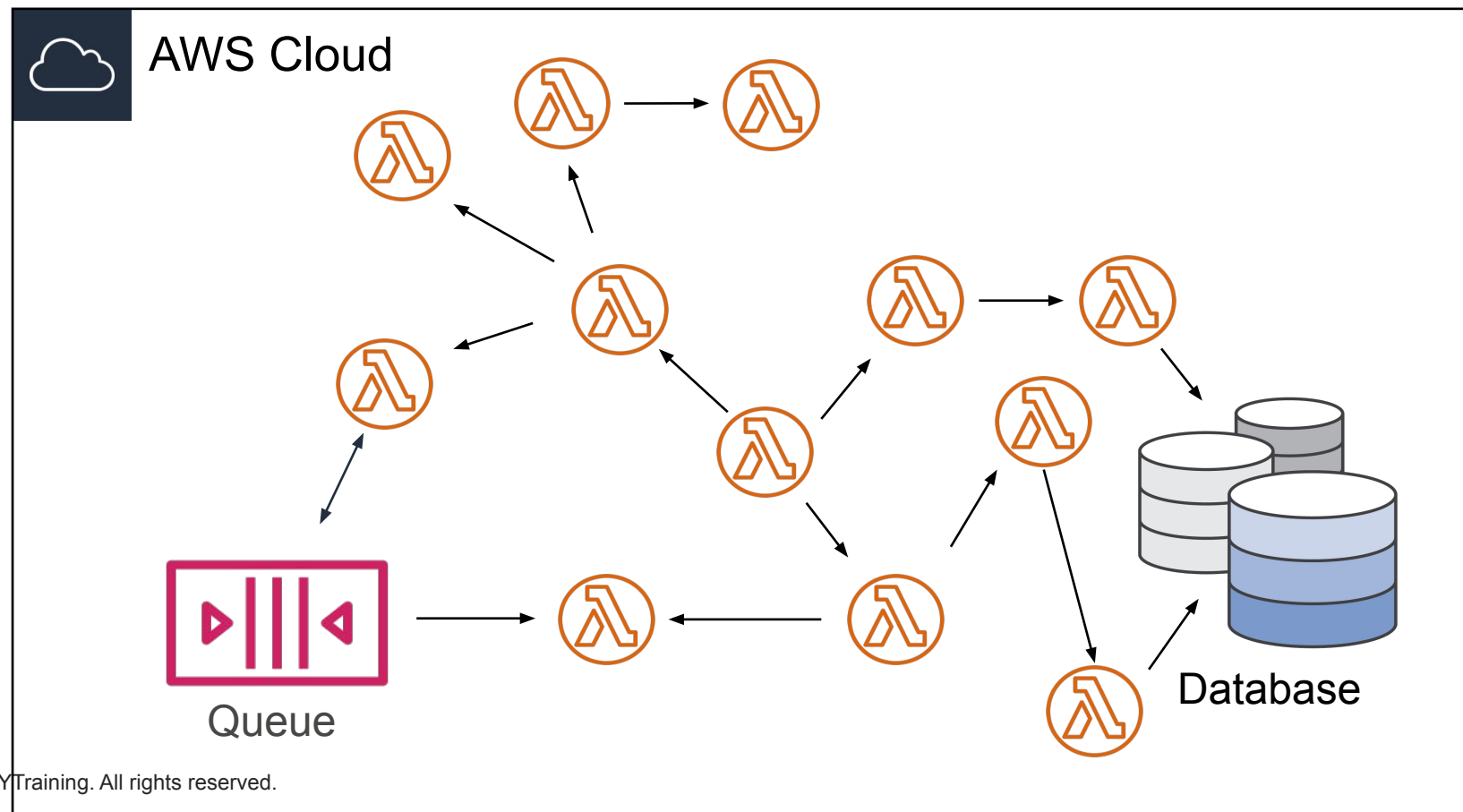
- Raison d'automatiser
- Automatisation de vos infrastructures
- Automatisation de vos déploiement
- AWS Elastic Beanstalk
- Amazon SQS
- Amazon MQ
- **AWS Step Function**





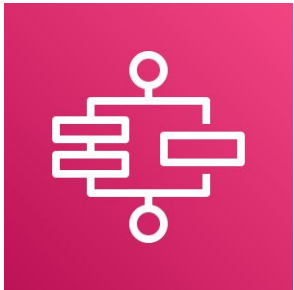


# AWS Step Function - solution pour l'orchestration des microservices





## AWS Step Function

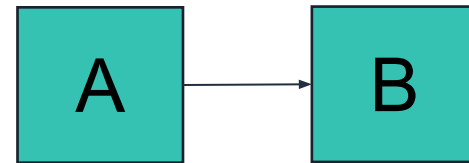


AWS Step  
Functions

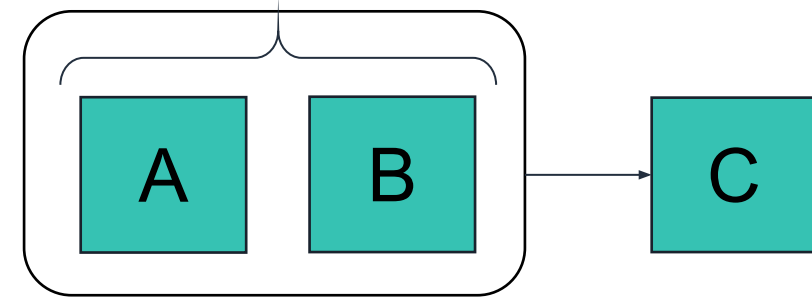
- Coordonne les microservices à l'aide de flux de travail visuels
- Vous permet de parcourir les fonctions de votre application
- Déclenche et suit automatiquement chaque étape
- Fournit une détection et une journalisation simples des erreurs en cas d'échec d'une étape



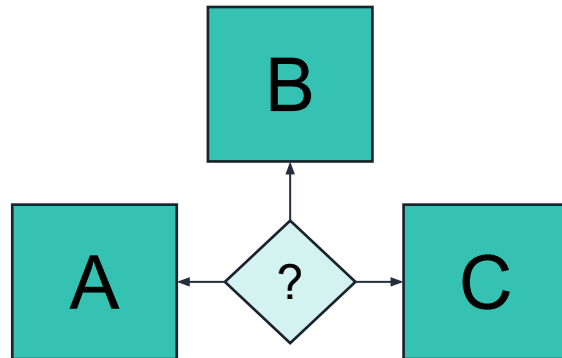
## AWS Step -coordination de workflow



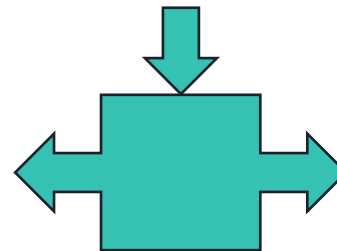
Exécuter les tâches dans l'ordre



Exécuter des tâches en parallèle



Sélectionnez une tâche en fonction des données



Gérer try-catch-finally comportement



Réessayer les tâches ayant échoué



# AWS Step Function -state machine ou état machine



Une machine à états est une collection d'États capables de faire du travail.

Distributeur automatique

attente de transaction



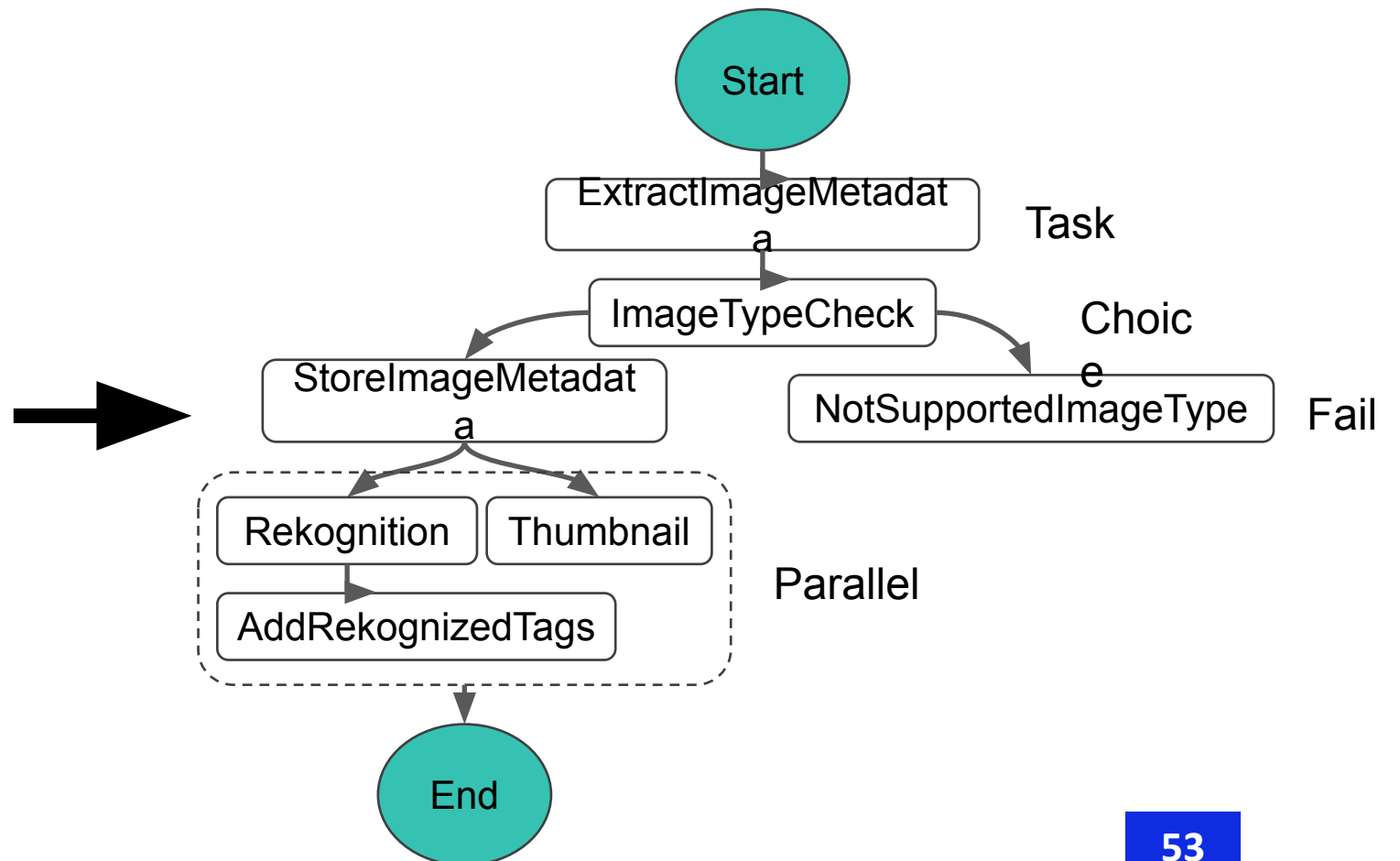
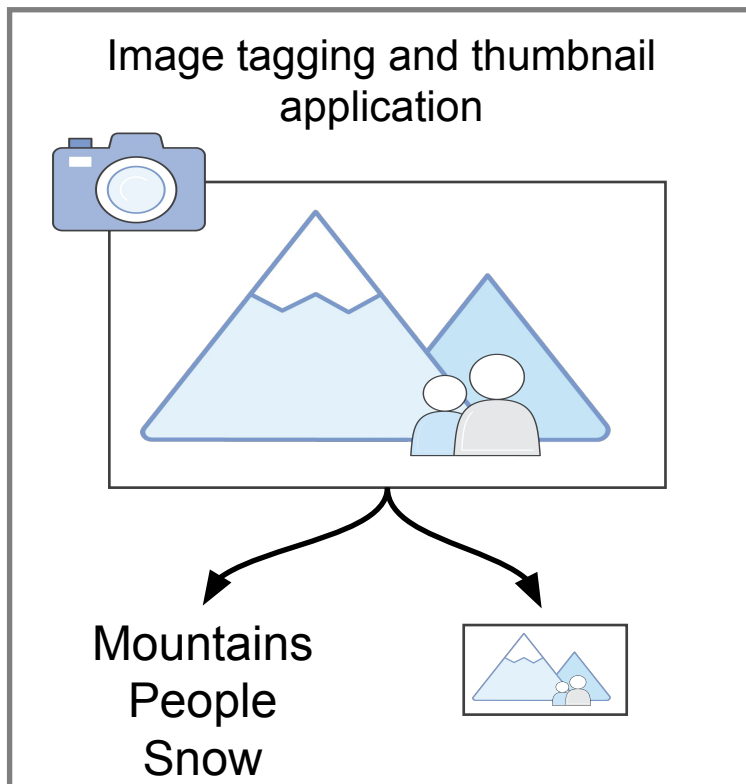
Sélection du soda



Vendre du soda



## AWS Step Function -states ou etats





## AWS Step Function -types d'état

Task	Une seule unité de travail
Choice	Ajoute une logique de branchement à une machine à états
Fail	Arrête une exécution et la marque comme un échec
Succeed	Arrête une exécution avec succès
Pass	Passe son entrée à sa sortie, sans effectuer de travail
Wait	Retards avant de continuer pendant une durée spécifiée
Parallel	Crée des branches d'exécution parallèles
Map	Carte Itère dynamiquement les étapes

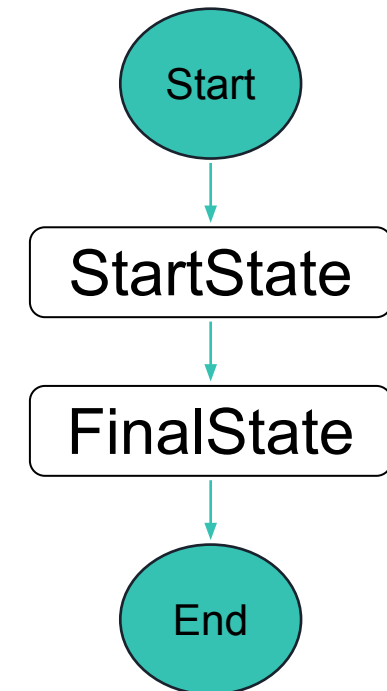


# AWS Step Function -States Language

Définir le workflow en JSON  
en utilisant la langue des États d'Amazonie

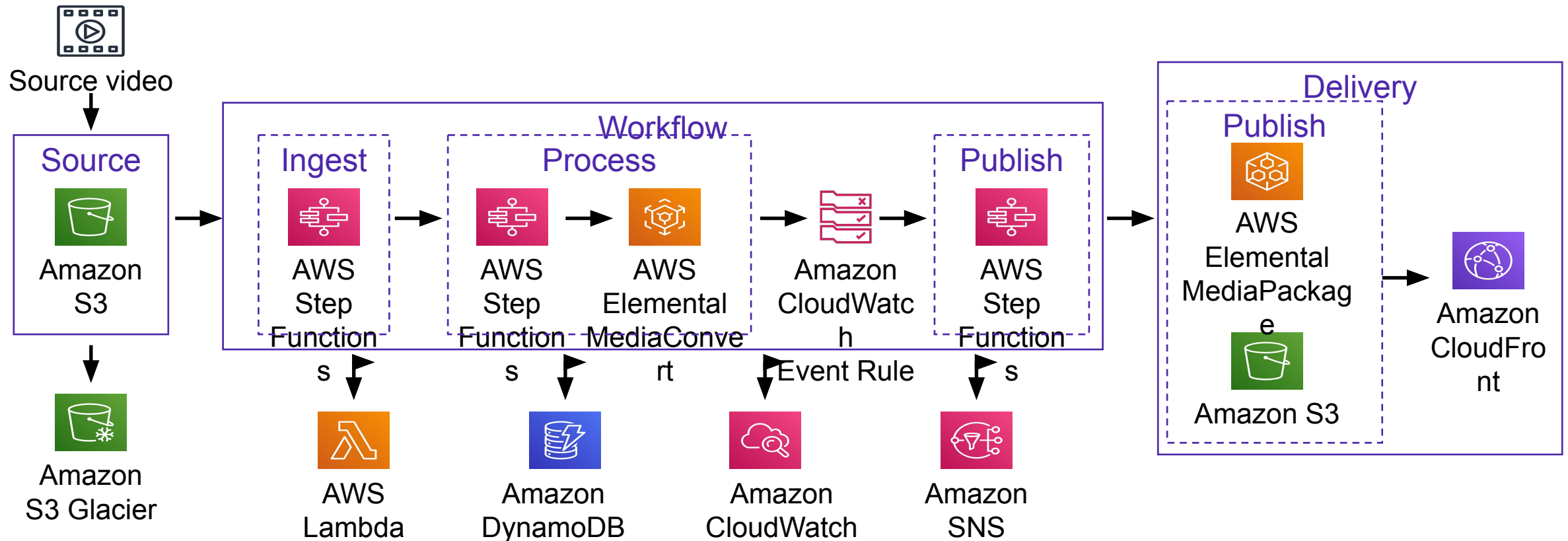
```
{  
  "Comment": "An example of the ASL.",  
  "StartAt": "StartState",  
  "States": {  
    "StartState": {  
      "Type": "Task",  
      "Resource":  
        "arn:aws:lambda:us-east-1:123456789012:function:my-lambda-function",  
      "Next": "FinalState"  
    },  
    "FinalState": {  
      "Type": "Task",  
      "Resource":  
        "arn:aws:lambda:us-east-1:123456789012:function:my-lambda-function",  
      "End": true  
    }  
  }  
}
```

Visualisez le flux de travail dans le  
Console de fonctions étape





## AWS Step Function -exemple





MERCI POUR VOTRE AIMABLE  
ATTENTION!



**Lahda Biassou Alphonsine**

Ingénieure cloud et Formatrice