

---

# AWS Java SDK

---

Prise en main

---

# AWS Java SDK Prise en main

---

Mathieu Bolla

Independent Father / GoAndCloud.it

@mathieubolla

<http://about.me/mathieubolla>

---

# AWS Java SDK Prise en main

---

Les services AWS

Introduction à S3

Introduction à EC2

Introduction à SQS

Conclusion

---

# AWS Java SDK Introduction

---

## **Calcul et réseau**

EC2 - AutoScaling - ELB - EMR - VPC - Route53 - DirectConnect - Market Place

## **Stockage et CDN**

S3 - Glacier - EBS - Import/Export - Gateway - CloudFront - Market Place

## **Base de données**

RDS - DynamoDb - ElastiCache - RedShift - Market Place

## **Applicatif**

CloudSearch - SNS - SQS - SWF - SES - FPS - Elastic Transcoder

## **Déploiement et gestion**

Management console - IAM - CloudWatch - Beanstalk - CloudFormation - DataPipeline - OpsWorks - HSM

---

# AWS Java SDK Introduction

---

## Calcul et réseau

**EC2** - AutoScaling - ELB - EMR - VPC - Route53 - DirectConnect - Market Place

## Stockage et CDN

**S3** - Glacier - EBS - Import/Export - Gateway - CloudFront - Market Place

## Base de données

RDS - DynamoDb - ElastiCache - RedShift - Market Place

## Applicatif

CloudSearch - SNS - **SQS** - SWF - SES - FPS - Elastic Transcoder

## Déploiement et gestion

**Management console** - **IAM** - CloudWatch - Beanstalk - CloudFormation - DataPipeline - OpsWorks - HSM

---

# AWS Java SDK Introduction

---

Exclusivement webservices

API SOAP ou REST

Amazon signature V2 ou V4

Amazon Management Console

Amazon AWS Java SDK

Autres langages supportés (Android, iOS, .Net, Node.js, Python, Ruby, PHP)

---

# AWS Java SDK Introduction: S3

---

Stockage binaire

Key Value store

Pas de notion de fichiers

Emulation de répertoires par clés “/”

Accès public HTTP possible

Gestion fine par métadonnées

---

# AWS Java SDK Introduction: EC2

---

Machines virtuelles “Virtual Private Server”

Amazon Machine Image par centaines

Multi data-center, multi régions

Réseau très haut débit

Tarifification à l’heure

Très nombreux services associés

---



# AWS Java SDK Introduction: SQS

---

Garantie de livraison

Transactions par reçus de livraison (garantie de processing)

Queue “probablement FIFO”

Pas de garantie d’unicité de livraison

Gestion fine des accès

---

# AWS Java SDK Introduction: IAM

---

Contrôle de sécurité de l'ensemble d'AWS

Utilisateurs, groupes, rôles

Amazon Resource Names

Policies

---

---

# AWS Java SDK

---

Mise en pratique

---

# AWS Java SDK: Cas concret

---

Application “Internet Scale”

Jouons à Maven: Téléchargeons Internet

Contrôleur dans nos locaux

Distribution de tâches par SQS

Sauvegarde des données sur S3

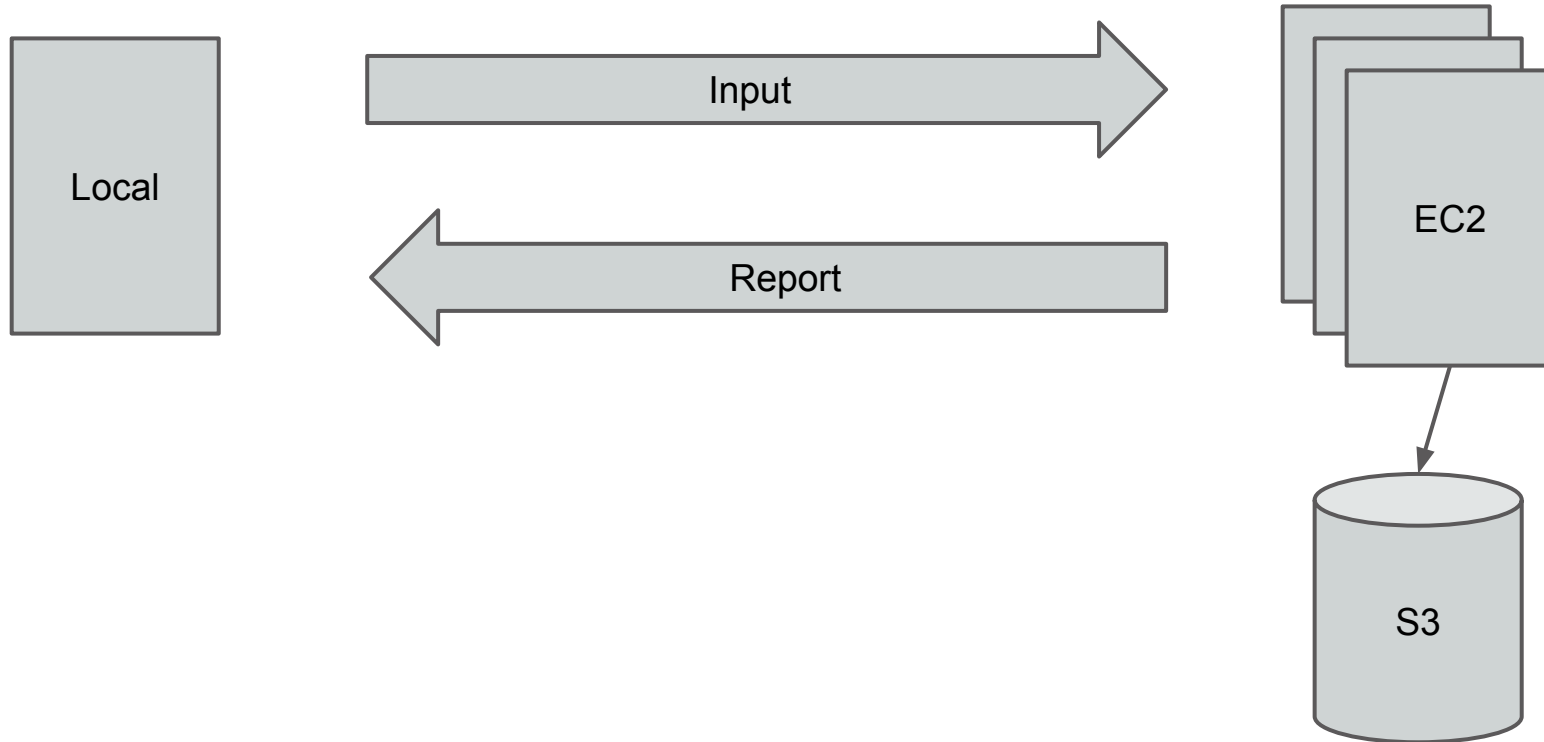
Collecte et extraction d'URL sur EC2

Séparation des droits par IAM

---

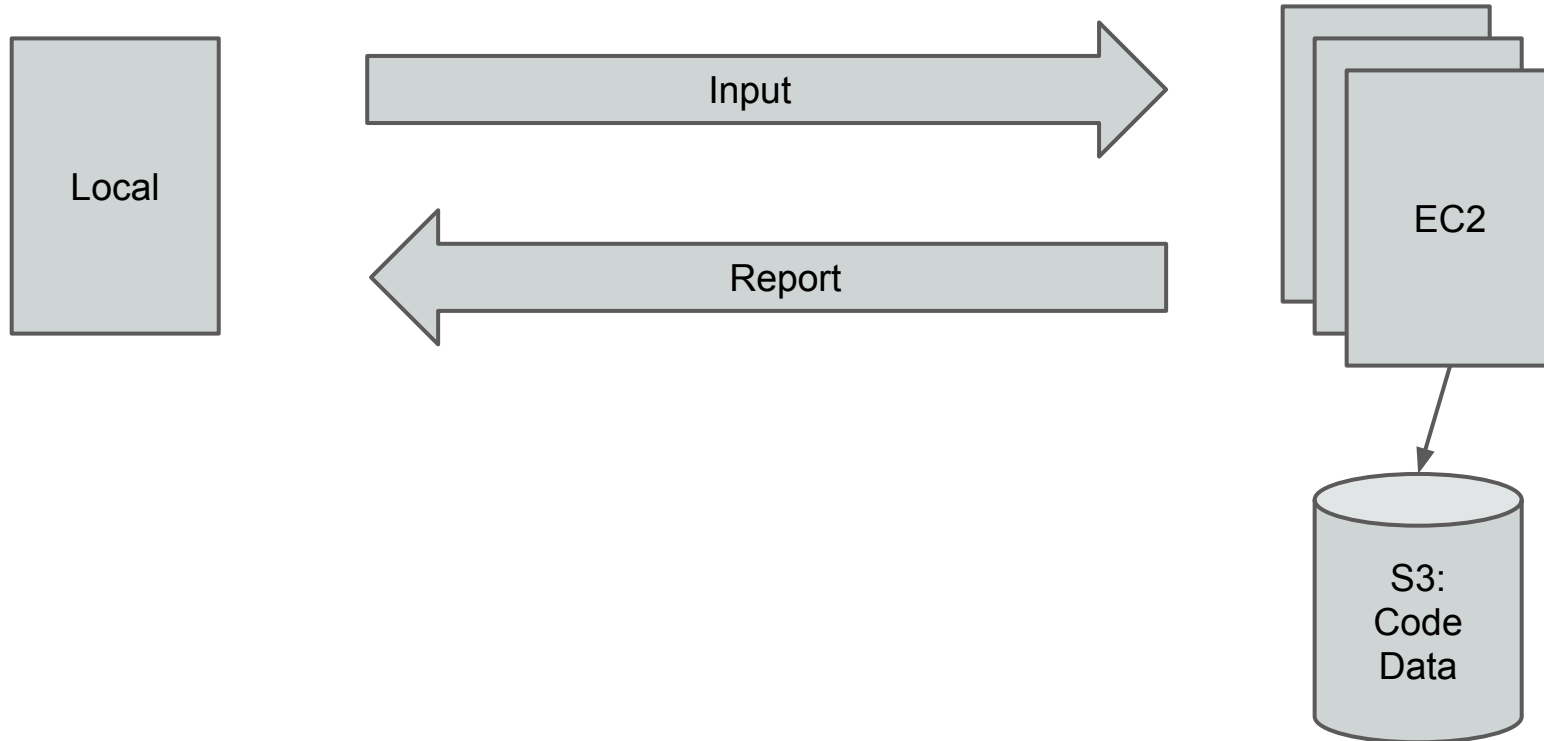
# AWS Java SDK: Cas concret

---



# AWS Java SDK: Cas concret

---



# AWS Java SDK: Cas concret

---

Fonctionnement tout automatique

Déploiement de code intégré

Démarrage et arrêt gérés

Pure Java / Maven / Bash

---

---

# AWS Java SDK

---

## Démonstration

---



---

# AWS Java SDK

---

## Conclusions

---

# AWS Java SDK: Généralités

---

## Sécurité par requêtes signés

Abstraction via le `AWSCredentials`

## Sécurité des credentials dans EC2

Mise à disposition des machines de “Role credentials”, abstraction via le `AWSCredentialsProvider` et `DefaultAWSCredentialsProviderChain`

## Limitation des roles

Un seul rôle par machine virtuelle

## Généralisation des ARN

Mis à disposition du code par chaque partie du framework, pour IAM

---

# AWS Java SDK: S3

---

## Sécurisation via IAM sur des wildcards ARN

"arn:aws:s3:::code.myproject.com/\*"

## Utilisation en parallèle à très haut débit

Plusieurs gigabits seconde, latence constante

## Fonctionnement transactionnel

Tout ou rien: Le fichier est complet ou absent

## Gestion complémentaire des droits

CannedAcl et accès HTTP public

---

# AWS Java SDK: EC2

---

## Choix d'un AMI

Très nombreux AMI spécifiques ou plus généralistes

## Choix d'une région

Répartition des risques, réduction de latence

## Choix d'un hardware

Très nombreux templates à disposition suivant l'usage et le coût

---

# AWS Java SDK: EC2

---

## Assignation d'un rôle

Gestion des droits d'accès, ségrégation d'environnement ou de responsabilités, sécurité des credentials

## Personnalisation

Utilisation des users data et notamment des scripts de démarrage de Cloud Init sur certaines AMI

## Gestion des disques

Utilisation d'AMI "Instance store" ou "EBS backed" et suppression des volumes à l'extinction

---

# AWS Java SDK: SQS

---

## Règles de nommage

Noms des queues, utilisation de queue URL au lieu des noms utilisateur, utilisation des ARN pour IAM

## Sécurité fine

Gestion par ARN de chaque verbe de l'interface REST (send, receive, delete, change, et d'autres pas vus en démo)

---

# AWS Java SDK: Questions

---

# AWS Java SDK: Resources

---

Cette présentation

<https://github.com/mathieubolla/aws-sdk-prise-en-main>

## Amazon Web Services

Console : <https://console.aws.amazon.com>

Website: <http://aws.amazon.com>

Java SDK: <http://aws.amazon.com/fr/documentation/sdkforjava/>

---