

# Authentification et autorisation d'accès avec AWS IAM



Julien Simon

Principal Technical Evangelist, AWS

[julsimon@amazon.fr](mailto:julsimon@amazon.fr)

@julsimon

# Agenda

- Modèle de sécurité AWS
- Bases d'IAM
- Stratégies IAM
- Stratégies gérées IAM
- Rôles IAM
- Fédération d'identité avec IAM
- Questions et réponses



# Modèle de sécurité AWS



# Résumé des épisodes précédents ;)

Webinaire “Modèle de sécurité AWS” : <https://www.youtube.com/watch?v=1QeKH-5nTlc>

Vous

Applications et contenu client

Gestion de plateforme, d'applications, d'identité et d'accès

Configuration du système d'exploitation, du réseau et du pare-feu

Chiffrement des données  
côté client

Chiffrement des données  
côté serveur

Protection du  
trafic réseau

Vous devez définir  
vos contrôles  
**DANS** le cloud

Services de base AWS

Calcul

Stockage

Base de données

Mise en réseau

AWS se charge  
de la sécurité  
**DU** cloud



Infrastructure globale AWS

Zones de disponibilité

Régions

Emplacements  
périphériques

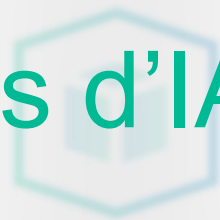
# Ce que cela signifie

- Vous conservez la **propriété** et le **contrôle** total de vos données
- Vous profitez d'un environnement créé pour les organisations **les plus exigeantes** en termes de sécurité
- AWS gère plus de **1 800** contrôles de sécurité et vous décharge d'une grande partie du travail de sécurisation
- **Vous** devez définir les contrôles de sécurité appropriés à votre charge de travail

# AWS Identity and Access Management

- IAM vous permet de contrôler **qui** a le droit de faire **quoi** dans votre compte et sur vos ressources AWS
- Pour chaque utilisateur, vous pouvez **autoriser** ou **interdire** individuellement **chaque opération** sur toute **ressource** AWS
- Vous pouvez également **enregistrer** tous les appels à l'API AWS dans AWS CloudTrail : nous en parlerons en détail dans un autre webinaire

# Bases d'IAM

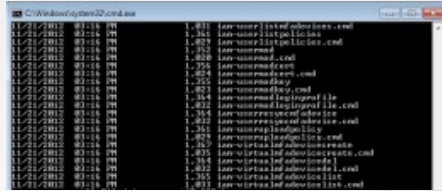


# Configurer et utiliser AWS IAM

# Interfaces Opérations



## Console AWS



## Ligne de commande AWS

# Interfaces Développement



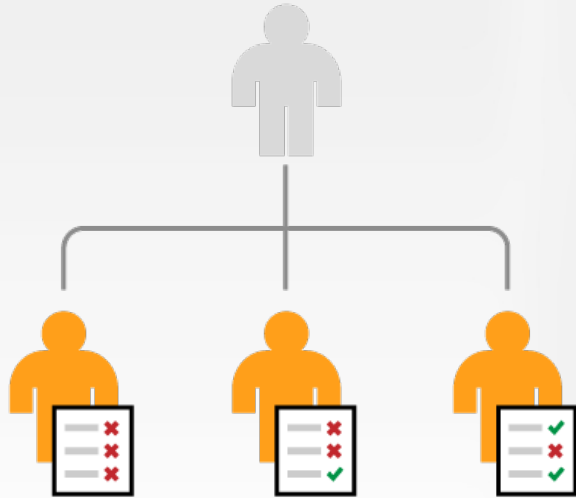
# Kits de développement logiciel AWS



## Autres services AWS



# Utilisateurs et Groupes IAM



# Authentification d'un utilisateur

Les utilisateurs IAM peuvent être authentifiés à l'aide des éléments suivants :

- Nom utilisateur + mot de passe
- Clés d'accès
  - *Access Key Id*
  - *Secret Access Key*
- Authentification multi-facteurs
  - *Token* matériel
  - Google Authenticator



Ne partagez jamais vos éléments d'authentification. J-A-M-A-I-S !

# Informations sur un utilisateur IAM

## Utilisateurs : webinarUser

ARN d'utilisateur arn:aws:iam:[REDACTED]:user/webinarUser

Chemin /

Heure de création 2016-12-12 10:20 UTC+0100

Autorisations

Groupes (0)

Informations d'identification de sécurité

Access Advisor

### Informations d'identification de connexion

Mot de passe de la console Activé  [Gérer le mot de passe](#)

lien de connexion de la console https://[REDACTED].signin.aws.amazon.com/console

Dernière connexion 2016-12-12 10:21 UTC+0100

Appareil MFA attribué Non 

Certificats de signature Aucun 

### Clés d'accès

Utilisez les clés d'accès pour effectuer des demandes de protocole REST ou HTTP Query sécurisées vers les API de service AWS. Pour votre protection, ne communiquez jamais vos clés secrètes à quiconque. Nous vous recommandons de procéder à une rotation fréquente des clés. [En savoir plus](#)

Créez une clé d'accès

ID de clé d'accès	Créé	Dernière utilisation	Statut	
[REDACTED]	2016-12-12 10:20 UTC+0100	N/A	Actif	<a href="#">Rendre inactif</a> 

# Stratégies IAM



# Stratégies IAM

Les stratégies IAM permettent d'associer des **permissions** aux utilisateurs et aux groupes IAM.

Chaque stratégie contient des **instructions** IAM qui définissent les **droits** de l'utilisateur.

Elles sont exprimées en **JSON**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:Describe*",
      "Resource": "*"
    }
  ]
}
```

AmazonEC2ReadOnlyAccess

[http://docs.aws.amazon.com/fr\\_fr/IAM/latest/UserGuide/reference\\_policies\\_elements.html](http://docs.aws.amazon.com/fr_fr/IAM/latest/UserGuide/reference_policies_elements.html)

# Instruction IAM

Principal

Action

Resource

Condition



```
{
  "Statement": [{
    "Effect": "effect",
    "Principal": "principal",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  }]
}
```

# Principal – Exemples

Une **entité** dont l'accès à une ressource est **autorisé** ou **refusé**  
Désignée par un **ARN** (Amazon Resource Name)

```
<!-- Tout le monde (utilisateurs anonymes) -->
"Principal": "AWS": "*"

<!-- Compte ou comptes spécifiques -->
"Principal": { "AWS": "arn:aws:iam::123456789012:root" }
"Principal": { "AWS": "123456789012" }

<!-- Utilisateur IAM individuel -->
"Principal": "AWS": "arn:aws:iam::123456789012:user/username"

<!-- Utilisateur fédéré (avec la fédération d'identité web) -->
"Principal": { "Federated": "www.amazon.com" }
"Principal": { "Federated": "graph.facebook.com" }
"Principal": { "Federated": "accounts.google.com" }

<!-- Rôle spécifique -->
"Principal": { "AWS": "arn:aws:iam::123456789012:role/rolename" }

<!-- Service spécifique -->
"Principal": { "Service": "ec2.amazonaws.com" }
```

Remplacez  
par votre  
numéro de  
compte

# Action – Exemples

- Décrit le **type d'accès** qui doit être autorisé ou refusé (i.e. l'API AWS)
- Vous trouverez la description complète dans la **documentation des services AWS**
- Les instructions doivent inclure un élément **Action** ou **NotAction**

```
<!-- Action EC2 -->  
"Action": "ec2:StartInstances"
```

```
<!-- Action IAM -->  
"Action": "iam:ChangePassword"
```

```
<!-- Action S3 -->  
"Action": "s3:GetObject"
```

```
<!-- Spécifiez plusieurs valeurs pour l'élément Action-->  
"Action": ["sqs:SendMessage", "sqs:ReceiveMessage"]
```

```
<--Utilisez des caractères génériques (* ou ?) dans le nom de l'action.  
Cela couvre Créer/Supprimer/Répertorier/Mettre à jour-->  
"Action": "iam:*AccessKey*"
```



# Comprendre NotAction

- Vous permet de spécifier une **liste d'actions**
- Peut permettre d'obtenir des **stratégies plus courtes** qu'avec l'utilisation de l'élément **Action** et le refus de nombreuses actions
- Exemple : supposons que vous souhaitiez tout autoriser à l'exception des API IAM

```
{  
  "Version": "2012-10-17",  
  "Statement": [ {  
    "Effect": "Allow",  
    "NotAction": "iam:*",  
    "Resource": "*"   
  }  
]
```



# Comprendre NotAction

- Vous permet de spécifier une **liste d'actions**
- Peut permettre d'obtenir des **stratégies plus courtes** qu'avec l'utilisation de l'élément **Action** et le refus de nombreuses actions
- Exemple : supposons que vous souhaitiez tout autoriser à l'exception des API IAM

```
{  
  "Version": "2012-10-17",  
  "Statement": [ {  
    "Effect": "Allow",  
    "NotAction": "iam:*",  
    "Resource": "*" }  
  ]  
}
```



ou

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Action": "*",  
    "Resource": "*" }  
  ],  
  {  
    "Effect": "Deny",  
    "Action": "iam:*",  
    "Resource": "*" }  
  ]  
}
```

# Comprendre NotAction

- Vous permet de spécifier une **liste d'actions**
- Peut permettre d'obtenir des **stratégies plus courtes** qu'avec l'utilisation de l'élément **Action** et le refus de nombreuses actions
- Exemple : supposons que vous souhaitiez tout autoriser à l'exception des API IAM

```
{  
  "Version": "2012-10-17",  
  "Statement": [ {  
    "Effect": "Allow",  
    "NotAction": "iam:*",  
    "Resource": "*"   
  }  
]
```

Il ne s'agit pas d'un **Deny**. Un utilisateur peut toujours avoir une stratégie distincte qui octroie **IAM: \***



ou

```
{  
  "Version": "2012-10-17",  
  "Statement": [ {  
    "Effect": "Allow",  
    "Action": "iam:*",  
    "Resource": "*"   
  },  
  {  
    "Effect": "Deny",  
    "Action": "iam:*",  
    "Resource": "*"   
  }  
]
```

Si vous souhaitez empêcher l'utilisateur de pouvoir appeler les API IAM pour toujours, utilisez un refus explicite.

# Resource – Exemples

- L'objet ou les objets demandés
- Les instructions doivent inclure un élément **Resource** ou **NotResource**

```
<-- Compartiment S3 -->
"Resource": "arn:aws:s3:::my_corporate_bucket/*"

<-- File d'attente SQS-->
"Resource": "arn:aws:sqs:us-west-2:123456789012:queue1"

<-- Plusieurs tables DynamoDB -->
"Resource": ["arn:aws:dynamodb:us-west-2:123456789012:table/
books_table",

"arn:aws:dynamodb:us-west-2:123456789012:table/magazines_table"]

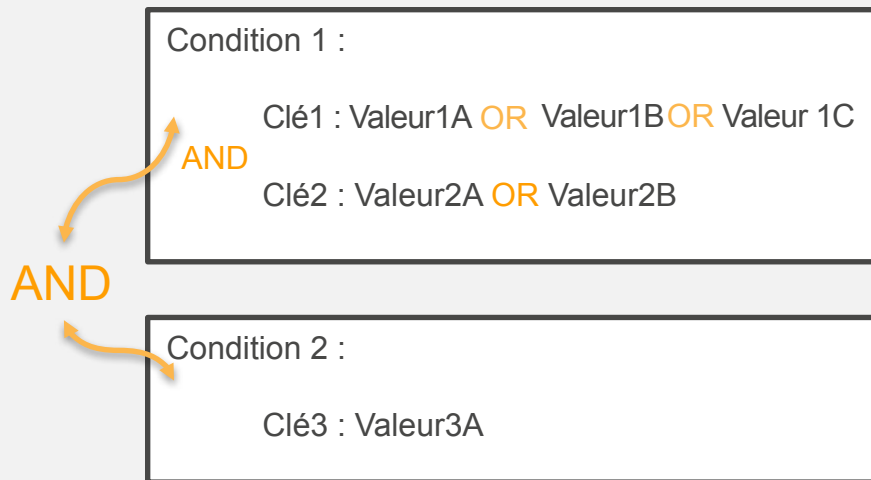
<-- Toutes les instances EC2 d'un compte dans une région -->
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

# Conditions

**Critères** qui doivent être **valides** pour que la stratégie s'applique

- Peut contenir **plusieurs conditions**
- Les clés de condition peuvent contenir **plusieurs valeurs**
- Si une seule condition comprend plusieurs valeurs pour une clé, la condition est évaluée à l'aide de l'opérateur logique **OR**
- Plusieurs conditions (ou plusieurs clés dans une seule condition) : les conditions sont évaluées à l'aide de l'opérateur logique **AND**

## Élément Condition



# Conditions – Exemple

Comment procéder si vous souhaitez limiter l'accès à une période et une plage d'adresses IP spécifiques ?

```
AND {  
  "Condition" : {  
    "DateGreaterThan" : {"aws:CurrentTime" : "2015-10-08T12:00:00Z"},  
    "DateLessThan": {"aws:CurrentTime" : "2015-10-08T15:00:00Z"},  
    "IpAddress" : {"aws:SourceIp" : ["192.0.2.0/24", "203.0.113.0/24"]}  
  }  
}
```

OR

Permet à un utilisateur d'accéder à une ressource dans les conditions suivantes :

- L'heure se situe après 12h le 08/10/2015 ET
- L'heure se situe avant 15h le 08/10/2015 ET
- La demande est issue d'une adresse IP située dans la plage 192.0.2.0 /24 OU 203.0.113.0 /24

Toutes ces conditions doivent être respectées pour que l'instruction soit appliquée.

# Variables de stratégie

- Variables **prédéfinies** basées sur le contexte de la demande de service
  - Clés **existantes** (aws:SourceIP, aws:CurrentTime, etc.)
  - Clés **propres à l'élément Principal** (aws:username, aws:userid, aws:principaltype)
  - Clés **propres au prestataire** (graph.facebook.com:id, www.amazon.com:user\_id)
  - Clés **SAML** (saml:aud, saml:iss)
- Avantages
  - Simplifie la gestion des stratégies
  - Réduit la nécessité d'avoir des stratégies codées en dur

# Stratégie avec variables

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["s3:ListBucket"],
    "Resource": ["arn:aws:s3:::myBucket"],
    "Condition": {
      "StringLike": {
        "s3:prefix": ["home/${aws:username}/*"]
      }
    },
  },
  {
    "Effect": "Allow",
    "Action": ["s3:*"],
    "Resource": [
      "arn:aws:s3:::myBucket/home/${aws:username}",
      "arn:aws:s3:::myBucket/home/${aws:username}/*"
    ]
  }
]
```

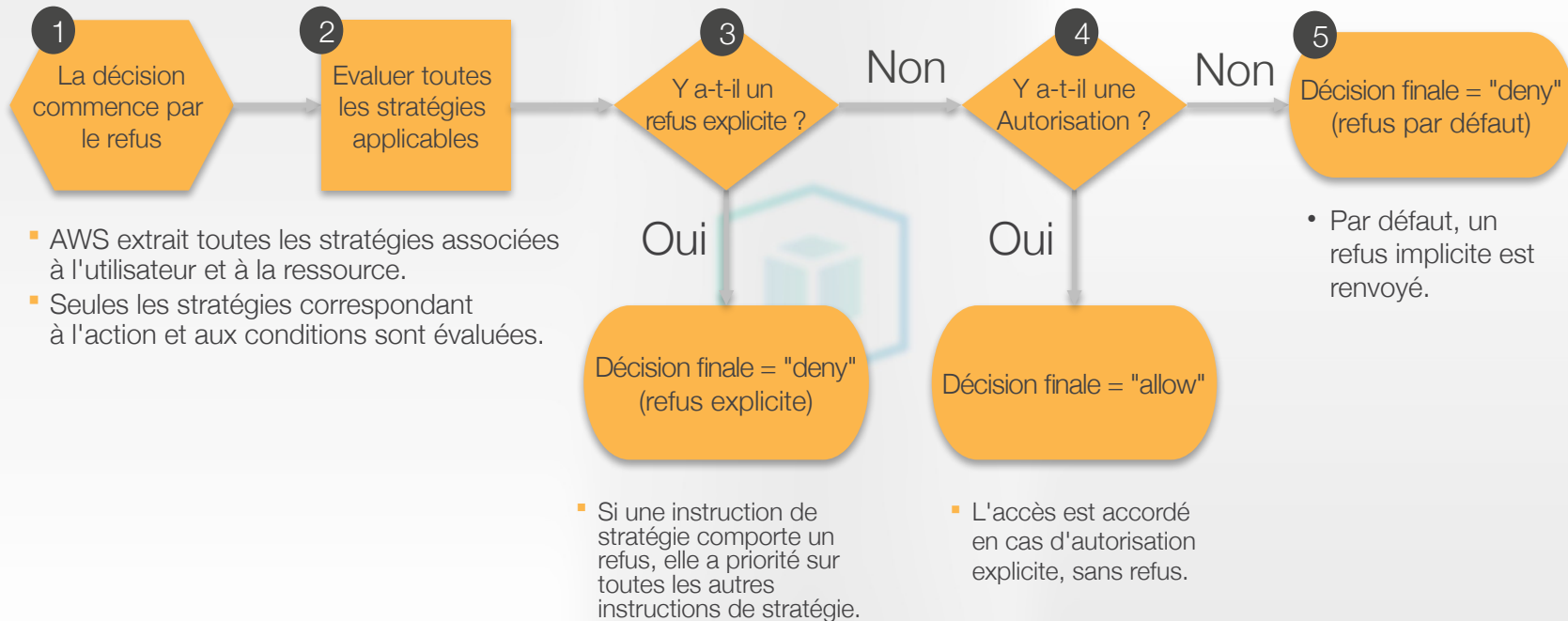
Variable dans les conditions

Variable dans les ARN  
de ressource

Un utilisateur a accès à son répertoire principal, contenu dans le bucket S3 'myBucket'



# Application des stratégies



Un refus est toujours prioritaire sur une autorisation.








# Rédiger une stratégie IAM






- Vous pouvez utiliser une **stratégie gérée**, liée au **poste** de l'utilisateur (Sys Admin, DBA, etc.) ou au **service AWS** (*AmazonS3ReadOnlyAccess*)
- Vous pouvez également vous **inspirer** d'une stratégie existante et l'adapter, par exemple en précisant les ressources concernées.
- Vous pouvez enfin partir d'une **feuille blanche**, en utilisant par exemple le **générateur de stratégies**.

Stratégies gérées



# Stratégies IAM gérées par AWS

Filtre : Stratégies gérées par AWS <input type="text" value="Q-Filtre"/>					239 résultats affichés
		Nom de la stratégie ↕	Entités attachées ▼	Heure de création ↕	Heure de modification ↕
<input type="checkbox"/>		AmazonS3FullAccess	3	2015-02-06 19:40 UTC+0100	2015-02-06 19:40 UTC+0100
<input type="checkbox"/>		AdministratorAccess	2	2015-02-06 19:39 UTC+0100	2015-02-06 19:39 UTC+0100
<input type="checkbox"/>		AmazonEC2ContainerServiceforEC2Role	2	2015-03-19 19:45 UTC+0100	2016-05-04 20:56 UTC+0100
<input type="checkbox"/>		AmazonEC2ContainerServiceRole	1	2015-04-09 18:14 UTC+0100	2016-08-11 15:08 UTC+0100
<input type="checkbox"/>		AmazonEC2ReadOnlyAccess	1	2015-02-06 19:40 UTC+0100	2015-02-06 19:40 UTC+0100
<input type="checkbox"/>		AmazonElasticMapReduceforEC2Role	1	2015-02-06 19:41 UTC+0100	2015-05-13 23:27 UTC+0100
<input type="checkbox"/>		AmazonElasticMapReduceFullAccess	1	2015-02-06 19:40 UTC+0100	2015-12-22 00:20 UTC+0100

Filtre : Job function <input type="text" value="Q-Filtre"/>					10 résultats affichés
		Nom de la stratégie ↕	Entités attachées ▼	Heure de création ↕	Heure de modification ↕
<input type="checkbox"/>		AdministratorAccess	2	2015-02-06 19:39 UTC+0100	2015-02-06 19:39 UTC+0100
<input type="checkbox"/>		PowerUserAccess	1	2015-02-06 19:39 UTC+0100	2016-12-06 19:11 UTC+0100
<input type="checkbox"/>		SecurityAudit	1	2015-02-06 19:41 UTC+0100	2016-12-09 19:51 UTC+0100
<input type="checkbox"/>		Billing	0	2016-11-10 18:33 UTC+0100	2016-11-10 18:33 UTC+0100
<input type="checkbox"/>		DatabaseAdministrator	0	2016-11-10 18:25 UTC+0100	2016-11-10 18:25 UTC+0100

# Générateur de stratégies IAM

## Modifier les autorisations

Le générateur de stratégies vous permet de créer des stratégies qui contrôlent l'accès aux produits et aux ressources AWS (Amazon Web Services). Pour en savoir plus sur la création de stratégies, consultez [Présentation des stratégies](#) dans Utilisation d'AWS Identity and Access Management.

Effet Autoriser ☒ Refuser ☐

Service AWS

AWS Application Discovery Se

Actions

-- Sélectionner des actions --

Amazon Resource Name  
(ARN)

\*

[Ajouter des conditions \(facultatif\)](#)

Ajouter une instruction

Effet	Action	Ressource	
Allow	ec2:*	*	<a href="#">Supprimer</a>
Allow	s3:CreateBucket s3>DeleteBucket	*	<a href="#">Supprimer</a>

# Générateur de stratégies IAM (suite)

## Examiner une stratégie

Personnalisez les autorisations en modifiant le document de stratégie suivant. Pour en savoir plus sur le langage de la stratégie d'accès, consultez [Présentation des stratégies](#) dans le guide *Utilisation d'IAM*. Pour tester les effets de cette stratégie avant d'appliquer vos modifications, utilisez le [simulateur de stratégies IAM](#).

### Nom de la stratégie

policygen-201612121043

### Description

### Document de stratégie

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "Stmt1481535725000",  
6       "Effect": "Allow",  
7       "Action": [  
8         "ec2:*"  
9       ],  
10      "Resource": [  
11        "*"   
12      ]  
13    },  
14    {  
15      "Sid": "Stmt1481535747000",  
16      "Effect": "Allow",  
17      "Action": [  
18        "s3:CreateBucket",  
19        "s3:DeleteBucket"
```

☒ Utiliser la mise en forme automatique pour la modification de stratégie

Annuler

Valider la stratégie

Précédent

Créer une stratégie

# Création d'une stratégie IAM à partir du code

```
public Policy GeneratePolicy(string bucket, string username, string ipAddress)
{
    var statement = new Statement(Statement.StatementEffect.Allow);

    // Autoriser l'accès au sous-dossier représenté par le nom d'utilisateur dans le
    // compartiment
    statement.Resources.Add(ResourceFactory.NewS3ObjectResource(bucket, username + "/*"));

    // Autoriser les demandes d'objets Get et Put
    statement.Actions = new List()
    { S3ActionIdentifiers.GetObject, S3ActionIdentifiers.PutObject };

    // Verrouiller les demandes issues de la machine cliente.
    statement.Conditions.Add(ConditionFactory.NewIpAddressCondition(ipAddress));

    var policy = new Policy();
    policy.Statements.Add(statement);

    return policy;
}
```

# Simuler une stratégie IAM

## Policies

Back

Editing policy: **PowerUserAccess**

AWS Managed Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "NotAction": [
        "iam:*",
        "organizations:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "organizations:DescribeOrganization",
      "Resource": "*"
    }
  ]
}
```

## Policy Simulator

Amazon S3 2 Action(s) sele... 

Select All Deselect All

Reset Contexts Clear Results Run Simulation

Global Settings ⓘ

Action Settings and Results [2 actions selected. 0 actions not simulated. 2 actions allowed. 0 actions denied.]

Service	Action	Resource Type	Simulation Resource	Permission
▶ Amazon S3	CreateBucket	not required	*	ⓘ allowed 1 matching statements.
▶ Amazon S3	DeleteBucket	not required	*	ⓘ allowed 1 matching statements.

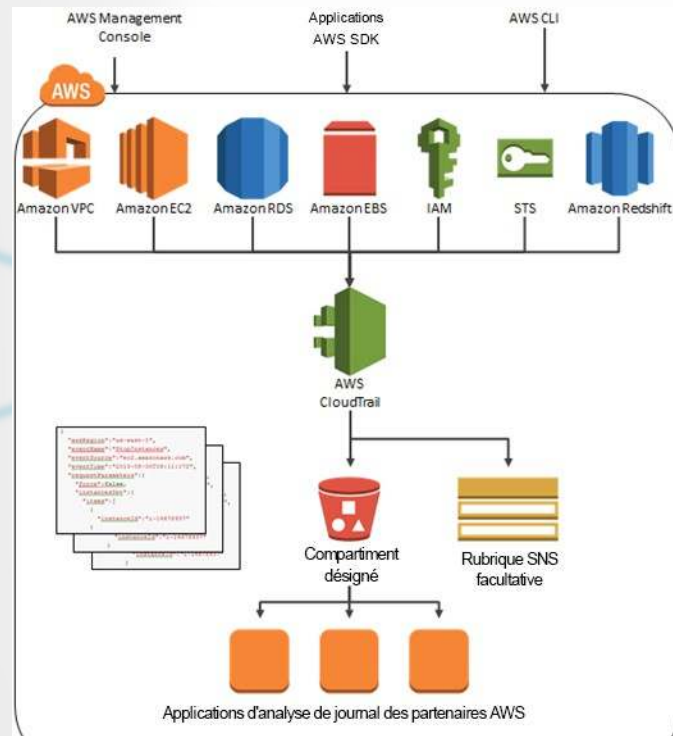
Le **simulateur IAM** vous permet de vérifier l'**efficacité** de vos stratégies... et bien sûr de les déboguer ;)

<https://policysim.aws.amazon.com/>



# Visibilité de l'accès aux API avec AWS CloudTrail

- AWS CloudTrail **capture** et **centralise** l'accès aux API :
  - Analyse de sécurité
  - Conformité
  - Dépannage
- Chaque entrée du journal AWS CloudTrail contient **l'identité IAM** de l'appelant



# Rôles IAM



# Rôles IAM

- Un **rôle** est composé d'une ou plusieurs **stratégies IAM**
- Un rôle peut être associé à :
  - Un **utilisateur IAM**
  - Un **groupe d'utilisateurs IAM**
  - Un **service AWS** (par ex., une instance EC2 ou une fonction Lambda)
  - Un **compte AWS**
- Pour chaque rôle, il faut définir :
  - Une **stratégie d'approbation** (*Trust Policy*) :  
**qui** est autorisé à assumer ce rôle (authentification MFA, etc.) ?
  - Une **stratégie d'autorisation** (*Access Policy*) :  
quels sont les **droits** associés au rôle ?

# Exemple – associer un rôle à une instance EC2

- Par défaut, une instance EC2 n'a le droit d'accéder à aucun service AWS.
- Au démarrage, il faut donc lui **associer un rôle** l'autorisant à utiliser les services dont elle a besoin, par exemple S3 ou RDS.

**Étape 3 : Configurer les détails de l'instance**  
Configurez l'instance en fonction de vos besoins. Vous pouvez lancer plusieurs instances à partir de la même AMI, demander des instances ponctuelles

Nombre d'instances  [Lancer dans un groupe Auto Scaling](#)

Option d'achat ☐ Demander des instances ponctuelles

Réseau  [Créer un nouveau VPC](#)

Sous-réseau  [Créer un nouveau sous-réseau \(subnet\)](#)

Attribuer automatiquement l'adresse IP publique

**Rôle IAM**  [Créer un nouveau rôle IAM](#)

Comportement d'arrêt

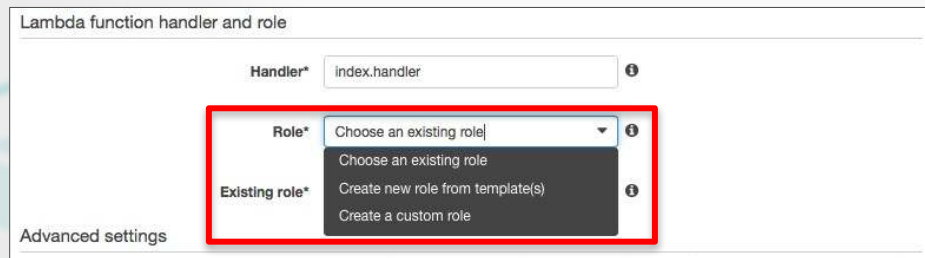
Activer la protection de la résiliation ☐ Protéger contre la résiliation accidentelle

Supervision ☐ Activer la surveillance détaillée de Cloudwatch  
Des frais supplémentaires seront facturés.

Location   
Des frais supplémentaires seront facturés pour la location dédiée.

# Exemple – associer un rôle à une fonction Lambda

- De même, une fonction Lambda n'a le droit d'accéder à aucun service AWS.
- Au démarrage, il faut donc lui associer un rôle l'autorisant à utiliser les services dont elle a besoin.



Lambda function handler and role

Handler\*  ⓘ

Role\*  ⓘ

Existing role\*  ⓘ

Advanced settings

# AWS Security Token Service

- STS permet à un **utilisateur IAM** de créer des permissions **temporaires** pour le compte d'utilisateurs **externes**.
- Ces permissions ne peuvent pas excéder ceux de l'utilisateur !
- **SDK** ou **API**
  - Création d'un jeton (*GetFederationToken*)
  - Valable de 15 minutes à 36 heures
  - Droits définis par la stratégie IAM passée en paramètre
  - MFA pas supporté
- **Console**
  - Création d'une session (*AssumeRole*)
  - Valable de 15 à 60 minutes
  - Droits définis par l'intersection du rôle et de la stratégie IAM passée en paramètre
  - MFA supporté



# Fédération d'identités

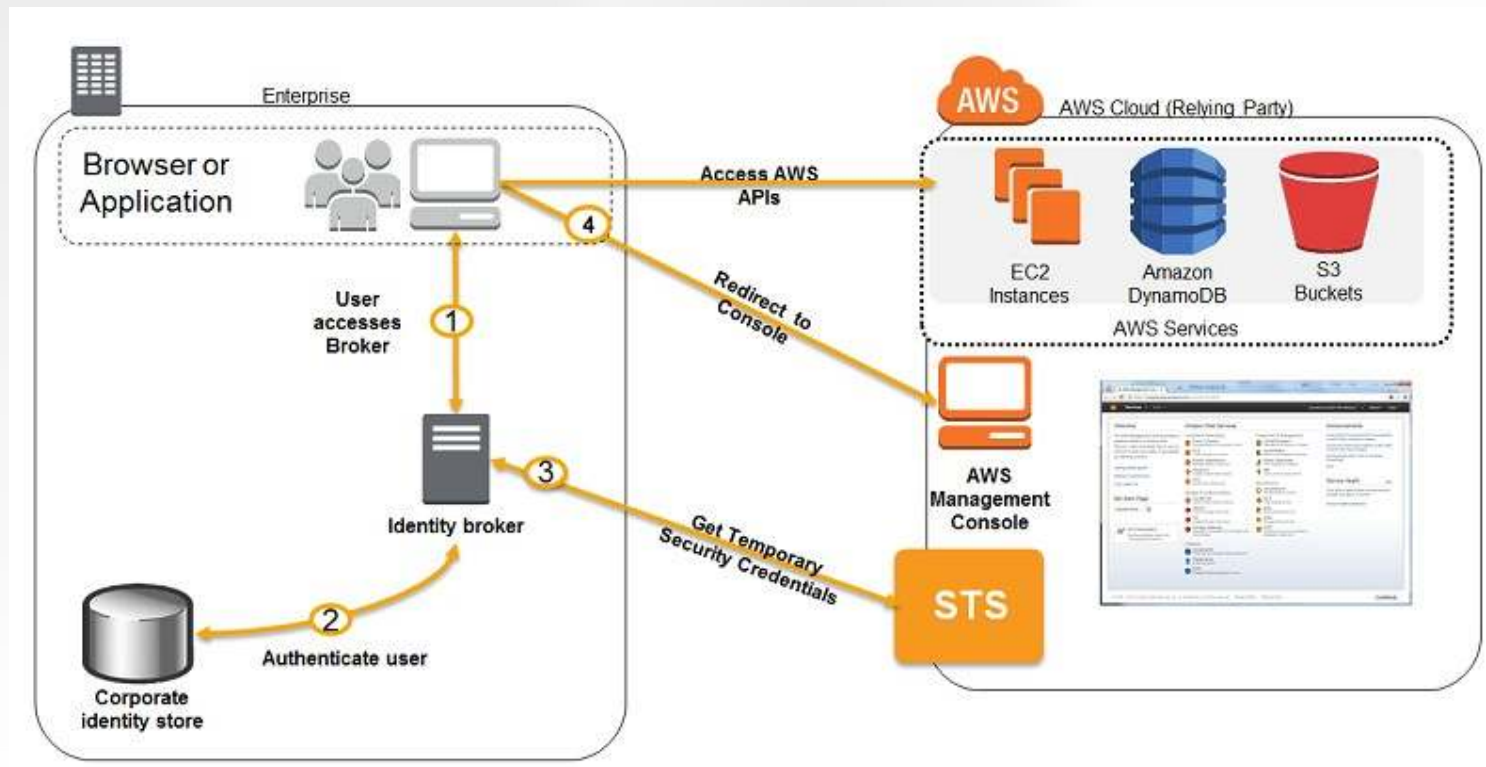


# Fédération d'identité

- Comment autoriser des **utilisateurs externes** à AWS, dotés de leur **propre identité**, à accéder à vos ressources AWS ?
- Utilisateurs avec une **identité “entreprise”**
  - Autoriser **l'utilisateur** à accéder au compte en *Single Sign-On*
    - Fédération avec un *broker* propriétaire
    - Fédération avec SAML 2.0
    - AWS Directory Service
- Utilisateurs avec une **identité “sociale”**
  - Autoriser **l'application** à accéder au compte
    - Fédération d'identité web
    - Fédération d'identité web avec AWS Cognito

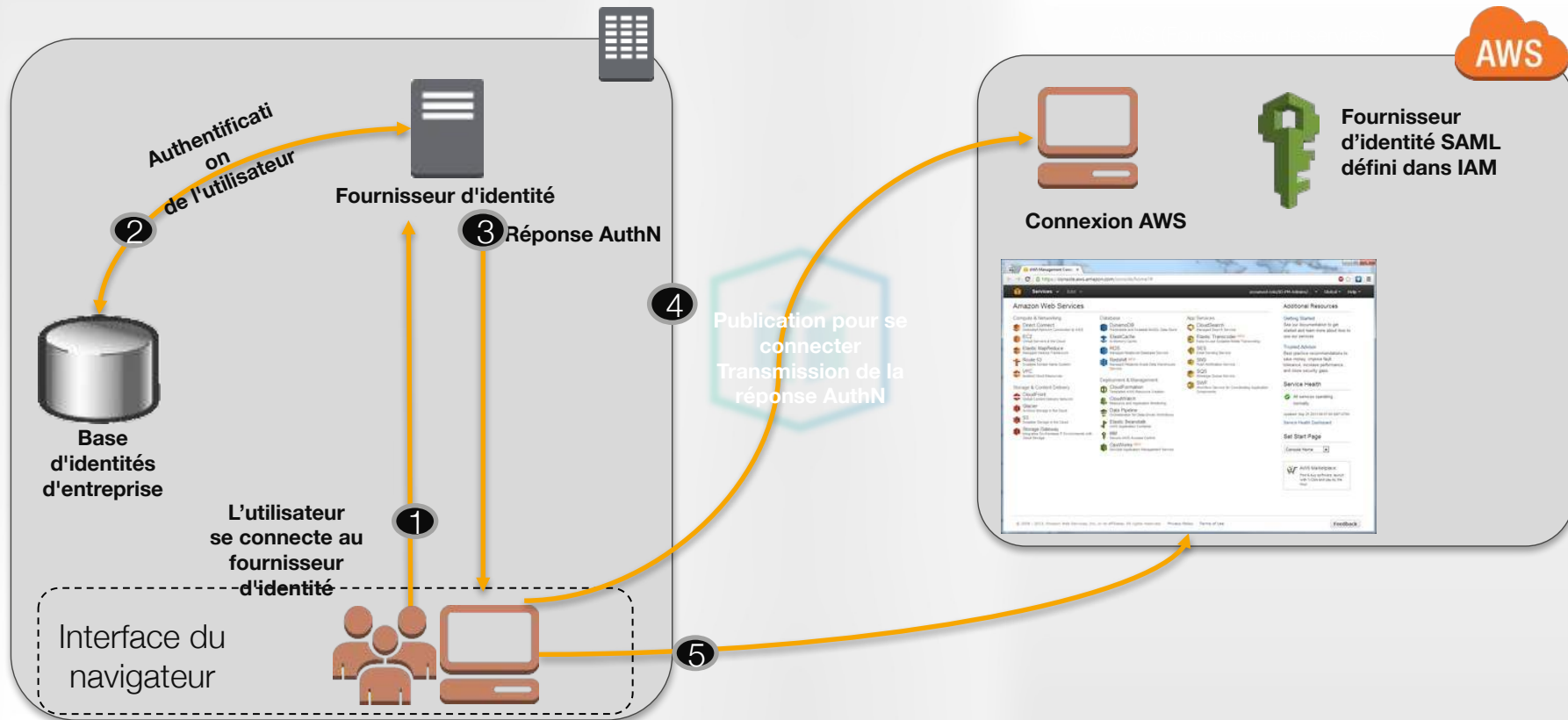


# Fédération d'identité avec *broker* propriétaire



# Fédération d'identité avec SAML 2.0

## AssumeRoleWithSAML



<https://aws.amazon.com/fr/blogs/aws/aws-identity-and-access-management-using-saml/>

<https://aws.amazon.com/fr/blogs/security/enabling-federation-to-aws-using-windows-active-directory-adfs-and-saml-2-0/>

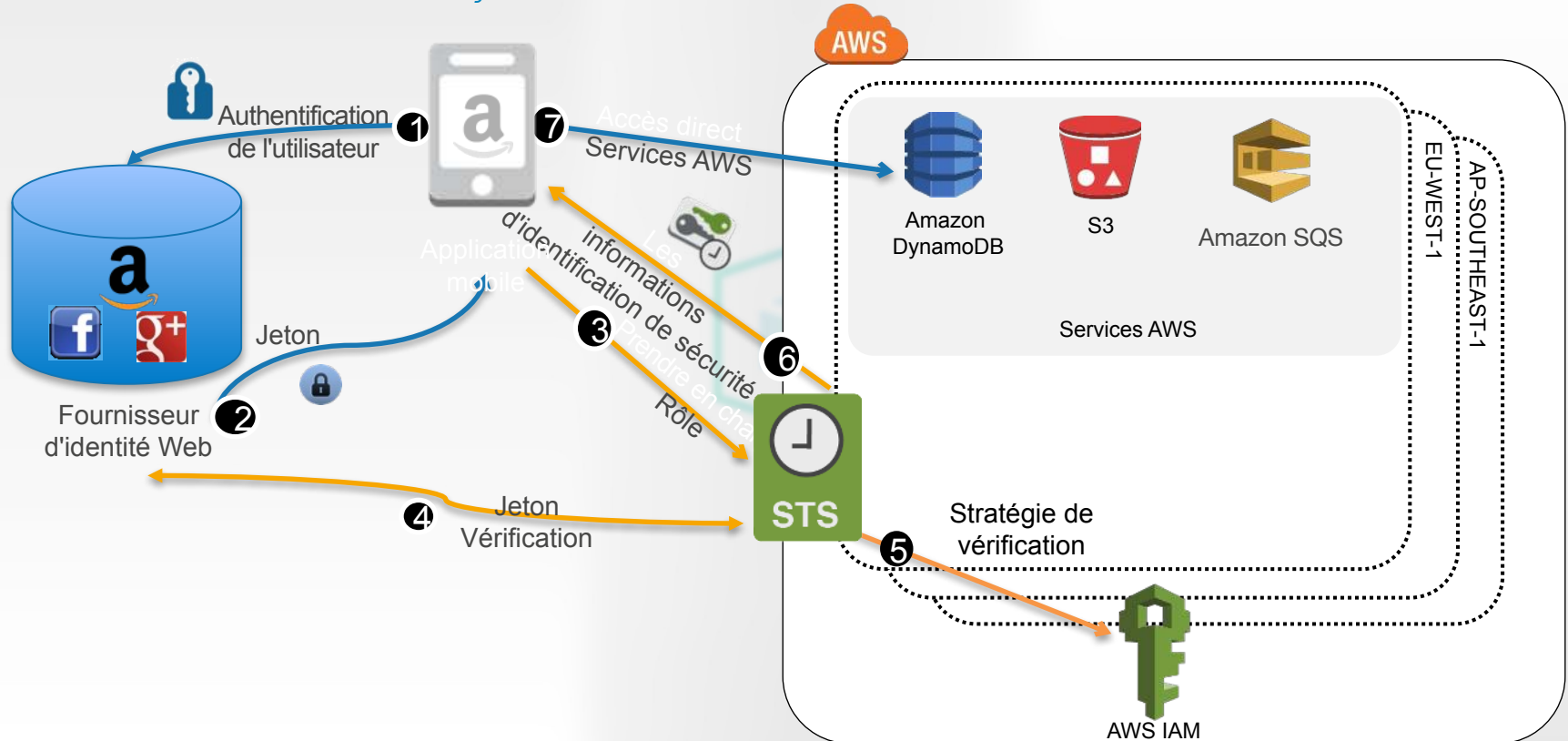
# AWS Directory Service



- Ce **service géré** vous permet de :
  - Connecter facilement un serveur Microsoft Active Directory **sur site** à vos ressources AWS **site** (*AD Connector*)
  - Configurer **un nouveau serveur AD** autonome dans AWS (*Simple AD*)
- DS peut utiliser l'annuaire pour autoriser la **fédération** des utilisateurs et des groupes de l'annuaire sur le Cloud AWS
- Pas besoin de développer un fournisseur d'identité externe

# Fédération d'identité Web

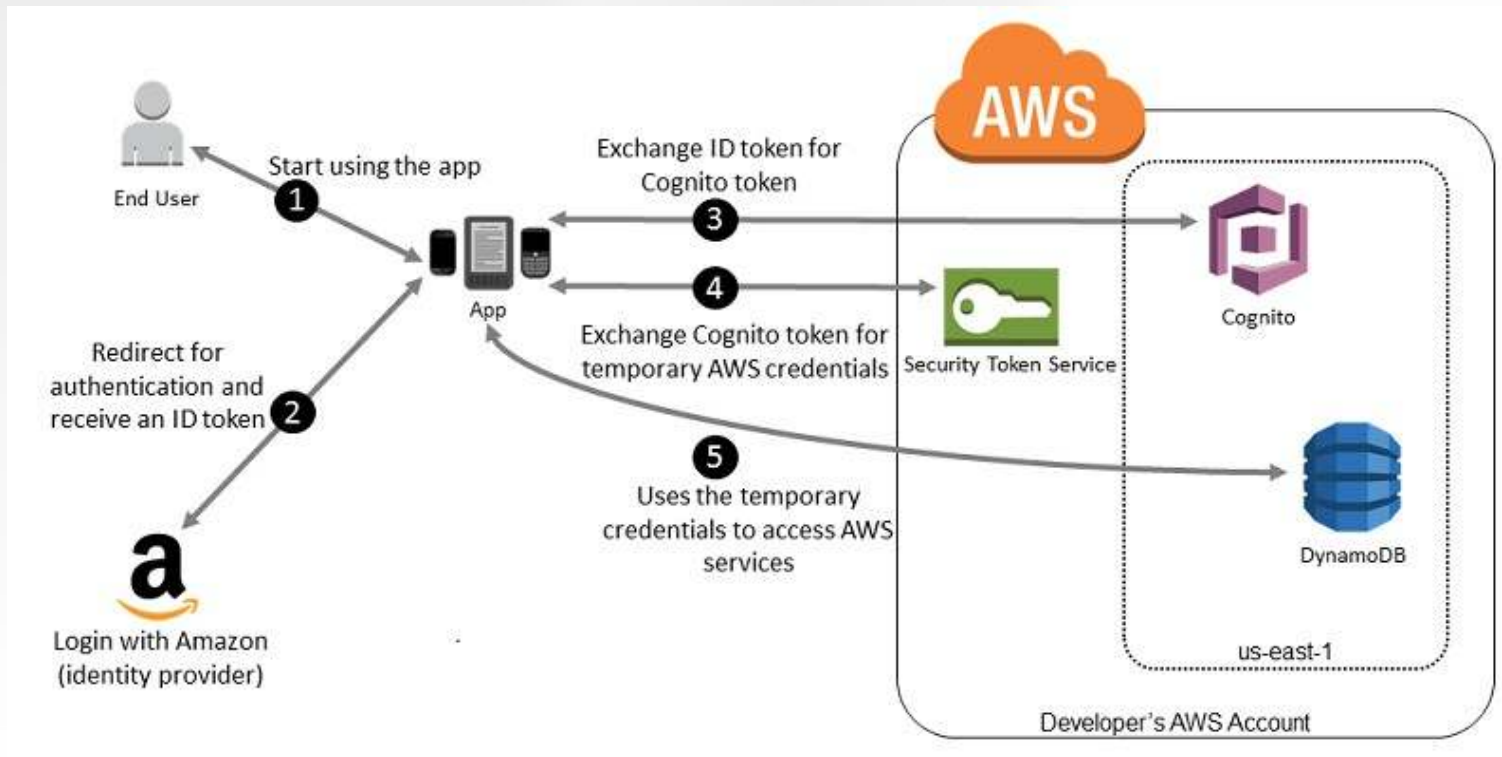
*AssumeRoleWithWebIdentity*



<https://aws.amazon.com/fr/blogs/aws/aws-iam-now-supports-amazon-facebook-and-google-identity-federation/>

<https://aws.amazon.com/fr/blogs/security/new-playground-app-to-explore-web-identity-federation-with-amazon-facebook-and-google/>

# Fédération d'identité Web avec Cognito



<https://aws.amazon.com/cognito/>

<https://aws.amazon.com/fr/blogs/aws/openid-connect-support/>

<https://aws.amazon.com/fr/blogs/security/building-an-app-using-amazon-cognito-and-an-openid-connect-identity-provider/>

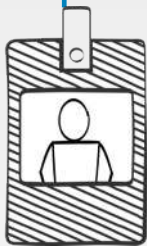
# Partenaires de fédération d'identité



# Conclusion



# Comment utiliser les fonctionnalités IAM de façon optimale ?



## Evitez le codage en dur

Inutile d'intégrer des informations d'identification aux applications – accédez aux ressources AWS à l'aide des rôles IAM pour EC2

- Pas de clés d'accès codées en dur dans le code source
- Pas de clés d'accès copiées sur les instances EC2
- Créez des rôles IAM avec des autorisations ayant le privilège le plus faible
- Utilisez les rôles IAM dans l'application
- Lancez vos instances EC2 avec un rôle minimal

## Changez les clés d'accès IAM AWS régulièrement

Une durée d'utilisation limitée des clés d'accès réduit les problèmes potentiels

- Créez une clé d'accès supplémentaire
- Mettez à jour toutes les applications afin d'utiliser la nouvelle clé
- Vérifiez que les applications fonctionnent
- Marquez la clé d'accès précédente comme inactive
- Vérifiez que les applications fonctionnent toujours
- Supprimez la clé d'accès inactive



## Pour résumer

- IAM a pour objectif de vous aider à **contrôler qui a le droit de faire quoi** à votre compte et à vos ressources
- IAM peut être **fédéré** avec des systèmes externes
- D'autres services AWS peuvent être utilisés pour configurer et enrichir IAM, par exemple :
  - AWS Directory Service
  - AWS Cognito
  - AWS CloudTrail

# Ressources supplémentaires

AWS IAM <http://aws.amazon.com/iam>

Blog sécurité AWS <https://blogs.aws.amazon.com/security/>

AWS re:Invent 2016: Become an AWS IAM Policy Ninja in 60mn (SAC303)

<https://www.youtube.com/watch?v=y7-fAT3z8Lo>

AWS re:Invent 2016: IAM Best Practices to Live By (SAC317) <https://www.youtube.com/watch?v=SGntDzEn30s>

AWS re:Invent 2014: Bring Your Own Identities – Federating Access to Your AWS Environment (SEC304)

<https://www.youtube.com/watch?v=debJ3o5w0MA>

AWS re:Invent 2015: A Progressive Journey Through AWS IAM Federation Options (SEC307)

<https://www.youtube.com/watch?v=-XARG9W2bGc>

# Merci !

Julien Simon

Principal Technical Evangelist, AWS

[julsimon@amazon.fr](mailto:julsimon@amazon.fr)

@julsimon

## Lundi

- Bonnes pratiques d'authentification avec AWS IAM
- Chiffrez vos données avec AWS

## Mardi

- Fireside chat avec Matthieu Bouthors et Julien Simon
- Re:Invent update 1

## Mercredi

- Deep dive : Amazon Virtual Private Cloud
- Bonnes pratiques anti-DDoS

## Jeudi

- Re:Invent update 2
- Gérez les incidents de sécurité avec AWS CloudTrail

## Vendredi

- Automatisez vos audits de sécurité avec Amazon Inspector
- Bonnes pratiques de sécurité sur AWS