Instant hands-on su AWS policy

Le policy in aws vengono scritte in formato JSON pertanto è opportuno saperlo come premessa.

Come possiamo vedere le policy sono caratterizzate da varie chiavi che insieme formano la policy. le prime 2 chiavi che saltano all'occhio sono Version che è la versione della policy e Statement. dove ogni blocco di statement inizia con le parentesi graffe { }

Statment è particolarmente importante perché divide all'interno di un file le diverse azioni di allow e denied di diverse azioni così da avere un foglio piu ordinato e non è possibile avere 1 azione di deny e 1 di allow nello stesso statement.

Da quest'altro esempio possiamo notare i 2 blocchi di statement distinti basta notare le parentesi graffe { } presenti nella stessa colonna della parola statement.
Un altra cosa che si nota sono **id** e **Sid** possiamo dire che queste 2 nuove chiavi possiamo usare o come identificatori oppure come descrittori cosi da inserire dei commenti su cosa fa quello statement o cosa fa quella policy.

il simbolo “*” significa qualunque opzione.

Effect: Può essere "Allow" o "Deny" per consentire o negare l'accesso.
Action: Le operazioni che possono essere eseguite su risorse AWS (es. s3:GetObject).
Resource: Le risorse AWS specifiche a cui l'azione si applica (es. arn:aws:s3:::nome-del-tuo-bucket/*).
Condition: (Opzionale) Specifica le condizioni per quando la policy è in effetto.
Delle chiavi che abbiamo citato prima abbiamo anche le versioni NOT[chiave]:

Action vs NotAction:

Action specifica le operazioni consentite. Esempio: consentire solo l'azione s3:GetObject.

NotAction specifica tutte le operazioni eccetto quelle elencate. Esempio: consentire tutte le azioni tranne s3:DeleteBucket.

Resource vs NotResource:

Resource definisce su quali risorse si applicano le azioni. Esempio: applicare regole su un bucket S3 specifico.

NotResource specifica tutte le risorse eccetto quelle elencate. Esempio: applicare regole a tutte le risorse tranne un bucket S3 specifico.

Principal vs NotPrincipal:

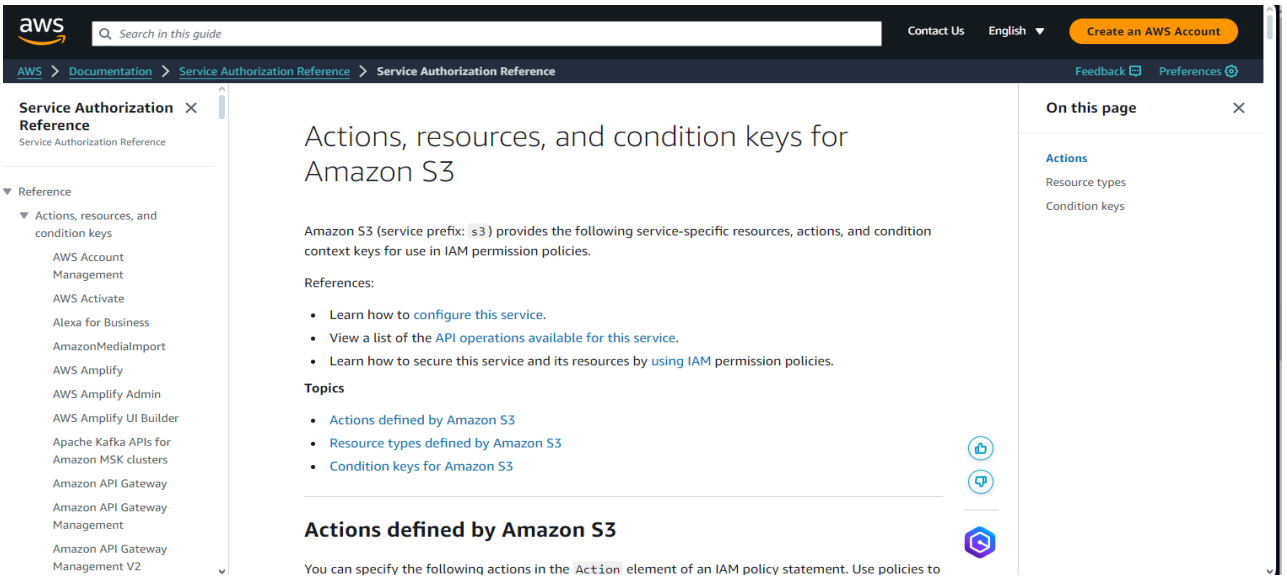
Principal indica chi è autorizzato a eseguire le azioni. Esempio: consentire accesso solo ad un utente IAM specifico.

NotPrincipal specifica chi è escluso dalle azioni concesse. Esempio: consentire accesso a tutti tranne un utente IAM specifico

Ora dopo aver descritto brevemente e grossolanamente le varie chiavi andiamo a cercare gli elementi che ci servono per la costruzione della nostra policy

Come usare la guida per Action,Resource e Condition:

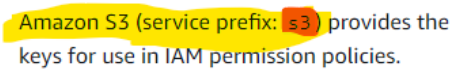
Come abbiamo detto precedentemente in questa sezione inseriamo tutte le azioni che vorremmo che faccia la nostra policy e il miglior modo per trovarle è su: https://docs.aws.amazon.com/service-authorization/latest/reference/reference_policies_actions-resources-contextkeys.html



Da come possiamo notare sulla sinistra abbiamo tutti i servizi da cui estrapolare le azioni, una volta trovato ed aperto il servizio aws che desideriamo utilizzare per la nostra policy. La pagina sarà composta da 3 sezioni. **Actions, Resource types** e **condition keys**

Actions, Resource types, Condition and variables:

Qui ci troviamo all'inizio della pagina dove verrà indicata la prefix:



Questa prefix dovremmo usarla all'inizio di ogni comando action che troveremo nella tabella sottostante quindi la formula è: `"Action": "{prefix}:{ActionComand}"`

example: `"Action": "s3:ListAllMyBuckets"`

Invece la tabella è composta da:

Actions	Description	Access level	Resource types (*required)	Condition keys	Dependent actions

1. Actions

Le "Actions" specificano le operazioni che possono essere eseguite su una risorsa AWS. Ad esempio, s3:PutObject è un'azione che permette di caricare un file in un bucket S3. Queste azioni sono ciò che effettivamente si concede o si nega in una policy IAM.

2. Description

La "Description" fornisce una spiegazione dell'azione

3. Access Level

L'"Access Level" categorizza le azioni in base al tipo di accesso che concedono:

- List: Permette di elencare risorse o informazioni su di esse.
- Read: Permette di leggere i contenuti e gli attributi delle risorse.
- Write: Permette di creare, modificare o cancellare risorse.
- Permissions management: Permette di gestire i permessi di accesso alle risorse.
- Tagging: Permette di assegnare o rimuovere tag dalle risorse.

4. *Resource Types (required)

I "Resource Types" indicano su quali tipi di risorse può essere eseguita l'azione. Alcune azioni possono essere eseguite su più tipi di risorse, mentre altre sono specifiche per un singolo tipo. Il simbolo "*" indica che l'azione richiede una risorsa specifica su cui operare.

```
{
    "Sid": "AllowManageOwnMFADevices",
    "Effect": "Allow",
    "Action": [
        "iam:CreateVirtualMFADevice",
        "iam:DeleteVirtualMFADevice",
        "iam:EnableMFADevice",
        "iam:ResyncMFADevice"
    ],
    "Resource": [
        "arn:aws:iam::123456789012:mfa/${aws:username}",
        "arn:aws:iam::123456789012:user/${aws:username}"
    ],
```

Da come possiamo vedere dall'esempio a sinistra con azioni come CreateVirtualMFADevice è richiesto come risorsa mfa* , mentre per azioni come EnableMFADevice è richiesta come risorsa user* anche per le risorse è presente una tabella nella quale sono presenti elementi come: Condition type, ARN e Condition key.

Resource types	ARN	Condition keys
mfa	arn:${Partition}:iam::${Account}:mfa/${MfaTokenIdWithPath}	aws:ResourceTag/${TagKey}
user	arn:${Partition}:iam::${Account}:user/${UserNameWithPath}	aws:ResourceTag/${TagKey} iam:ResourceTag/${TagKey}

4.5 Variabili:

Le variabili sono un concetto che si mettono in mezzo tra le condition key e le resource. Guardando la tabella delle risorse indicata sopra l' ARN che identifica la risorsa specifica è composto da variabili che dobbiamo inserire noi come: **${Partition}, ${Account}, ${UsernameWithPath}** e cosi a continuare per tutta la tabella…
${Partition}: tipicamente si sostituisce con AWS a meno che non siamo in cina o un ente gov.
Le altre variabili possiamo indicarle noi o farle assumere da altre variabili(condition key)

```
"Resource": [
    "arn:aws:iam::123456789012:user/${aws:username}"
    "arn:aws:iam::${aws:PrincipalAccount}:user/${aws:username}"
```

Queste 2 stringhe di codice possono indicare la stessa cosa se eseguita dall'account 123456789012. qui siamo andati a lavorare su ${Account} come possiamo notare. e per trovare la "tabella" in cui sono presenti tutte le variabili dobbiamo andare a cercare le GLOBAL CONDITION KEY:
https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_condition-keys.html#condition-keys-principalaccount

5. Condition e Condition Keys

Le condition è lo strumento che ci permette di imporre delle condizioni prima di far eseguire determinate "Action" la formula è:

```
"Condition" : { "{condition-operator}" : { "{condition-key}" : "{condition-value}" }}
```

Condition Operators:

Li troviamo in:
https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_condition_operators.html
Appunto descrivono la logica dell operatore come deve leggere e trattare il valore di tale condizione, tra i piu comuni abbiamo:
`StringEquals,StringLike,Bool,IpAddress,NotIpAddress ecc…`

Condition keys:

I "Condition Keys" permettono di definire le condizioni sotto le quali l'azione è permessa o negata e da come indicato precedentemente possiamo dire di averne di 2 tipi: **GLOBAL CONDITION KEY** e **Resource Condition Key.**
Le **GLOBAL CONDITION KEY:** sono condizioni che possiamo usare su qualsiasi risorsa e vengono utilizzate per controlli di sicurezza su: utente/account,ruolo,network,tipologia di richiesta, attributi della risorsa e cosi via…

Resource Condition Key: sono condizioni utilizzabili specificatamente per quella risorsa le quali ci permettono di fare dei controlli specifici su quella risorsa come nell esempio →
dove usiamo s3:prefix che possiamo usare in questa specifica action che ha quella specifica risorsa con la propria Resource condition key. Quindi gli utenti possono visualizzare solo la propria prefix nel bucket d esempio

6. Dependent Actions

Le "Dependent Actions" sono azioni che devono essere permesse affinché l'azione principale sia eseguibile. Ad esempio, per eseguire un'azione che legge dati da un bucket S3, potrebbe essere necessario anche avere il permesso di eseguire un'azione che elenca i bucket (come s3:ListBucket), a seconda della configurazione specifica della policy.