# Pipeline de AWS IAM Roles

## O que são **Roles**

De forma simplista, entenda ***Role*** (ou "Função", no *IAM* em pt-BR) como uma permissão de um objeto *AWS* chamar ou usar outro, exemplo: *Lambda* ler um registro no *DynamoDB* ou *Lambda* gravar um log no *CloudWatch*.
Em geral essas *roles* permitem a comunicação e interação entre os componentes *AWS*, principalmente aqueles que não possuem TCP/IP como porta de entrada (como um *Firewall,* onde se é configurado quais comunicações devem ser aceitas ou rejeitadas).

## E **Policies**?

***Policies*** na verdade são uma forma de organizar e aproveitar determinados grupos de permissões. Por exemplo, na *AWS* temos as ***AWS Managed Policies***, que são agrupamentos de permissões (*actions*) que podem ser anexas a uma *role*, a um usuário ou a um grupo de usuários.
Geralmente a *policy* já faz uma trava no recurso destino, por exemplo: o ARN da instância que quer enviar um objeto para o S3, ou seja se um *lambda* assumir uma *role* que contenha essa *policy* ele terá permissão de subir um objeto no respectivo S3.

```
{
  "Sid": "Stmt1453380948823",
  "Action": [
    "s3:DeleteObject",
    "s3:PutObject"
  ],
  "Effect": "Allow",
```

**Importante:** há várias formas de se escrever policy a depender da ferramenta de provisionamento: *JSON*, *YAML* ou *HCL* (*HashiCorp Language*).
Aqui na Rede você verá logo a seguir que usamos *YAML* via *CloudFormation*.

## Você disse **Action**?

A ***Action*** é a permissão propriamente dita, ou seja, se desejo enviar um objeto para o S3, no grupo de *actions* deve existir o valor [ s3:PutObject ] . Várias *actions* são agrupadas em uma *policy*.

## Vamos para o **SW7**

Para tudo isso, que trata-se de permissão, existe uma pipeline exclusiva e no SW7 foi criada na seguinte estrutura: http://gitlab.prd.useredecloud/Cloud-DevSecOps/roles/sw7

Essa estrutura encontra-se dentro do grupo Cloud-DevSecOps, portanto o acesso é restrito. Mas o subgrupo para a sigla SW7 já foi criado, sendo necessário apenas obedecer o GitFlow criado para esta estrutura.

Vamos entender a seguir a partir de um exemplo:

*Exemplo:*

Supomos que haja um novo projeto que necessitará de recursos AWS: Projeto Wolverine.
Este projeto conterá dois *Lambdas* e um *DynamoDB*. Neste caso apenas um somente postará no *DynamoDB* e outro fará somente a leitura.

Para este cenário claramente há roles envolvidas, pois sem as devidas permissões nenhum dos Lambdas conseguirá efeturar a respectiva operação no DynamoDB.
Assim sendo, será necessário projetar quais novas *policies* e quais permissões cada *policy* deverão conter para serem aplicadas em cada ambiente e os recursos fazerem uso.

Nesta situação você fará o seguinte:

1. Abri chamado para SCM DevOps via http://jirasd.prd.useredecloud/servicedesk/customer/portal/8/group/28
    a. Na opção 'Outros', abrir novo formulário e citar a estrutura final do projeto que subirá as IAM Roles*Exemplo:* Criar a estrutura gitlab.prd.useredecloud/Cloud-DevSecOps/roles/sw7/**projeto-wolverine** e atribuir as permissões de Maintener para <nome> e Developer para <nome>
2. Uma vez criado o repositório criar as branchs conforme mencionado na página Automação para criação de roles
    a. Importante já inserir os arquivos mencionados na página
3. Seguir estritamente o que descreve a página Como criar e estruturar um projeto para realizar o Setup do projeto
    a. Será necessário consultar alguém de Cloud para fornecer a API Key

4. Uma vez o projeto com o Setup realizado preencher os arquivos de template Cloudformation (YAML) conforme sugerido na página Estrutura dos templates
5. Realizar commit das alterações

Se você chegou até aqui observou a seguinte nota nas páginas mencionadas:

> ***Execução***
> *Todas alterações deverão acontecer em uma nova branch criada e então abrir um merge para a branch referente ao ambiente de destino,*
> *não deve ocorrer merge entre as branch de cada ambiente, ex: developmente -> homolog.*
> *A branch que será feita as mudanças pode ser criada a partir da branch de destino.*
>
> *A partir do momento que ocorre o merge, é feito o disparo de toda automação, que faz a leitura de todos os templates, e também a validação dos recursos.*
> *Durante a execução já é feita toda a parte dos GuardRails, no qual mantera todo o controle do que pode ser criado/adicionado, todos seguindo padrões e boas praticas recomendadas.*
>
> *Para a exceção em produção, será necessário a abertura com card. Que será feito através do próprio Gitlab, que vai executar a abertura do card a partir de uma tag gerada da branch que contem as mudanças.*

# Bora **validar!**

Acesse a console na opção IAM > Roles > *Filtre o nome das Roles criadas via YML CloudFormation* >  Confira se está tudo certo.

A seguir vídeos tutoriais de como aplicar essas roles:


https://web.microsoftstream.com/video/28eb0708-a8be-4de5-b5cf-0227bab9b35c