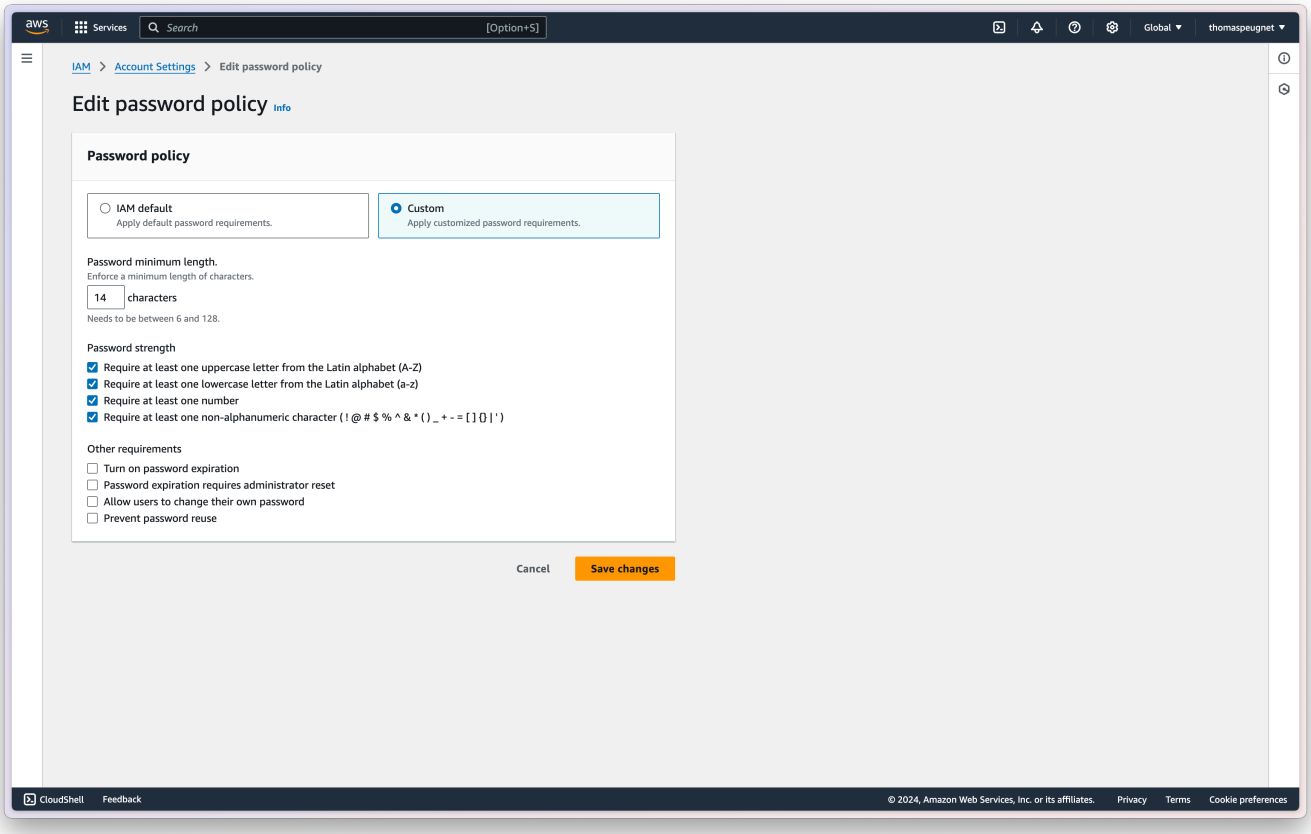
Rendu TP02

Ce TP a été réalisé par `Thomas PEUGNET`.

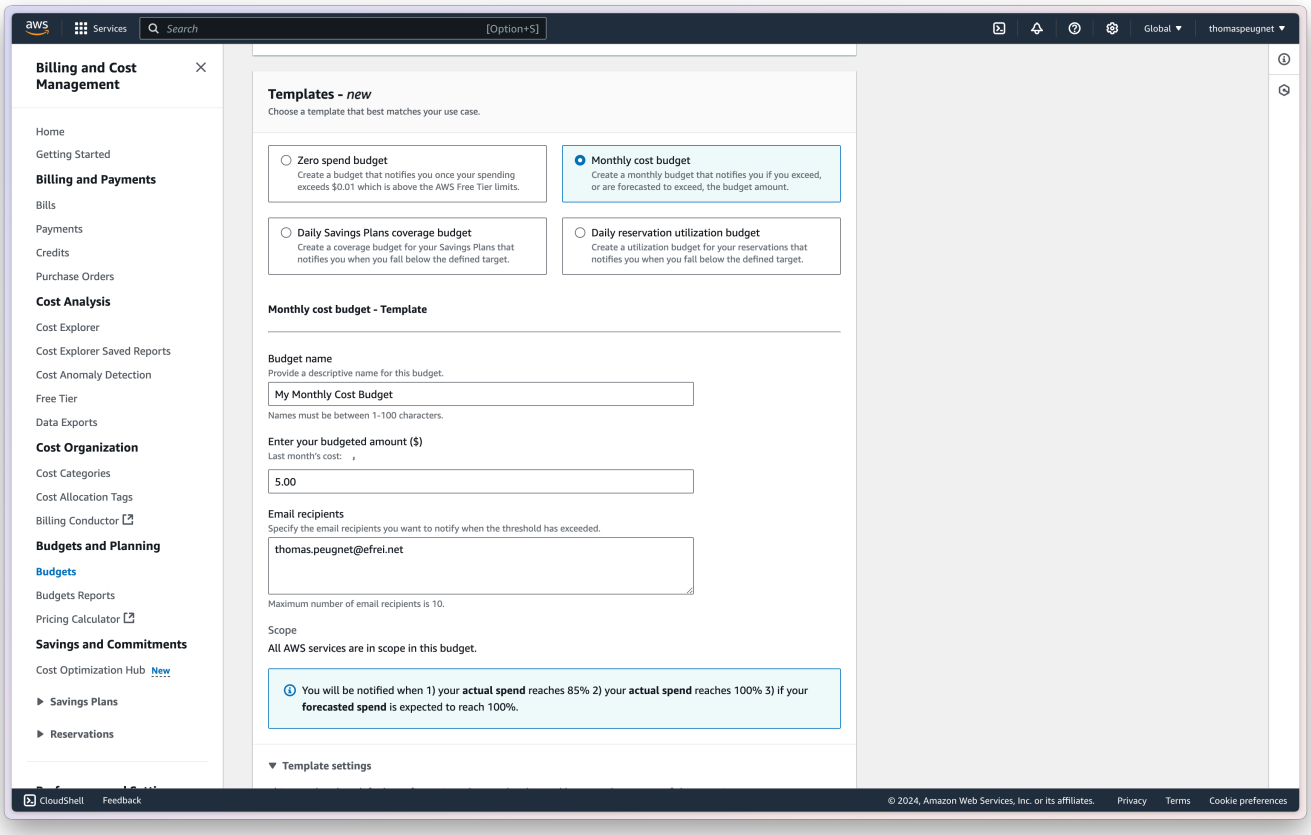
Sécurisation du compte AWS

Nous configurons une Alert Preference.

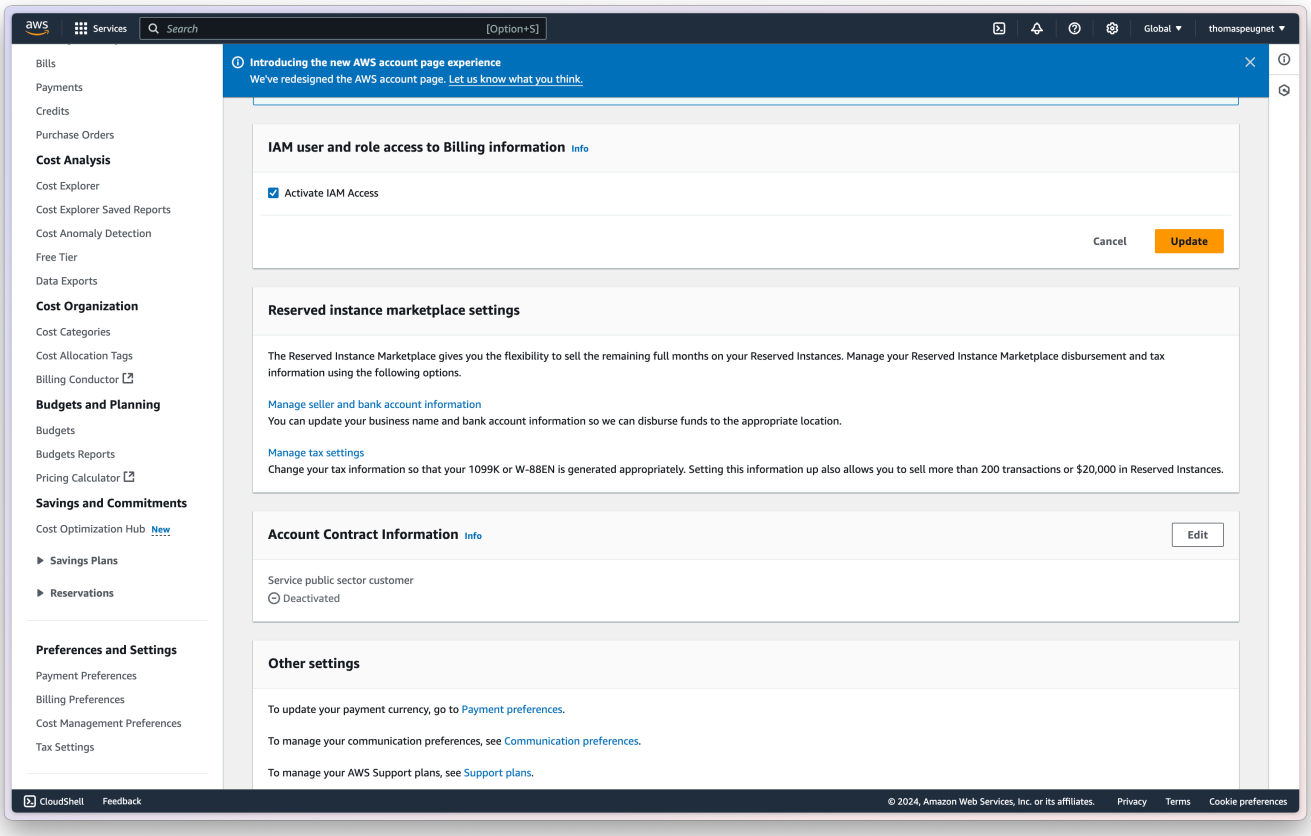
Nous mettons à jour la politique par défaut.



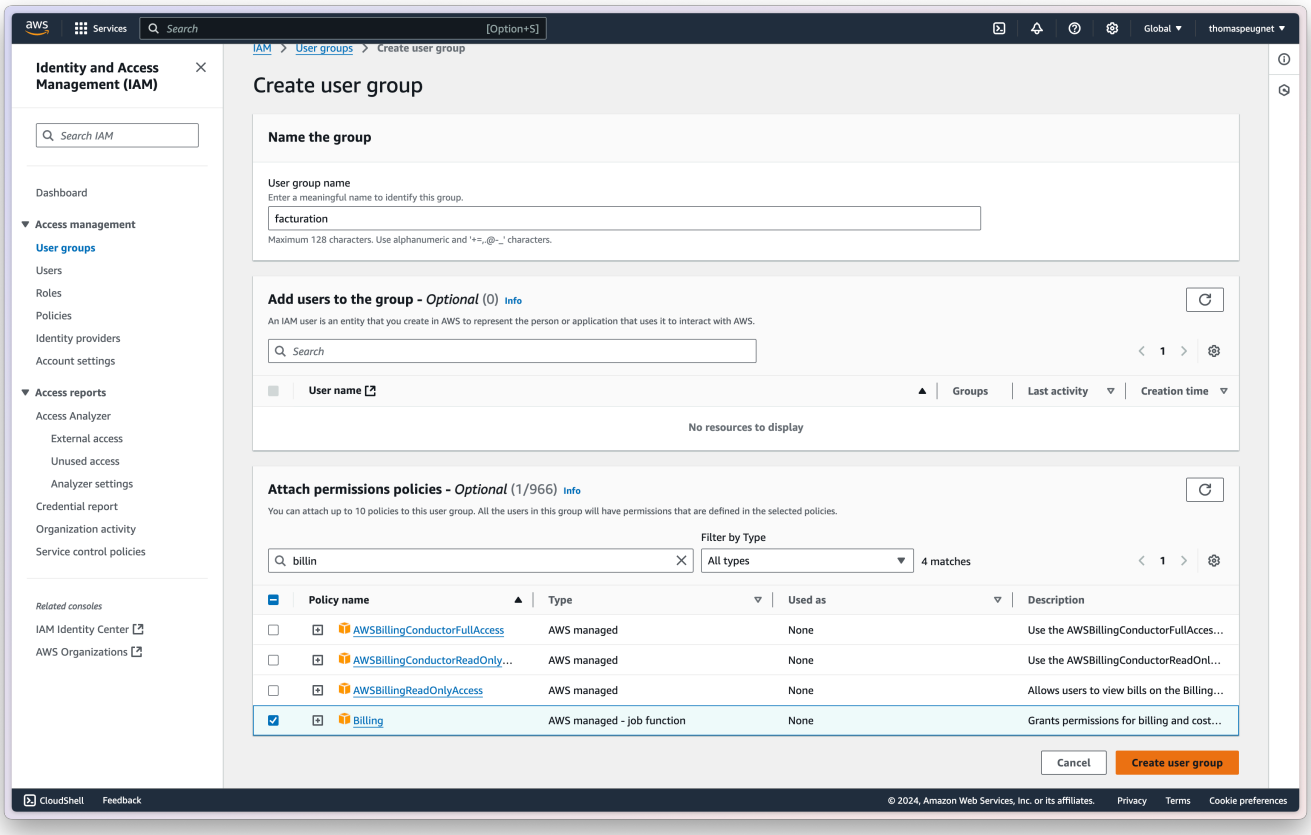
Nous créons un budget mensuel de 5$.



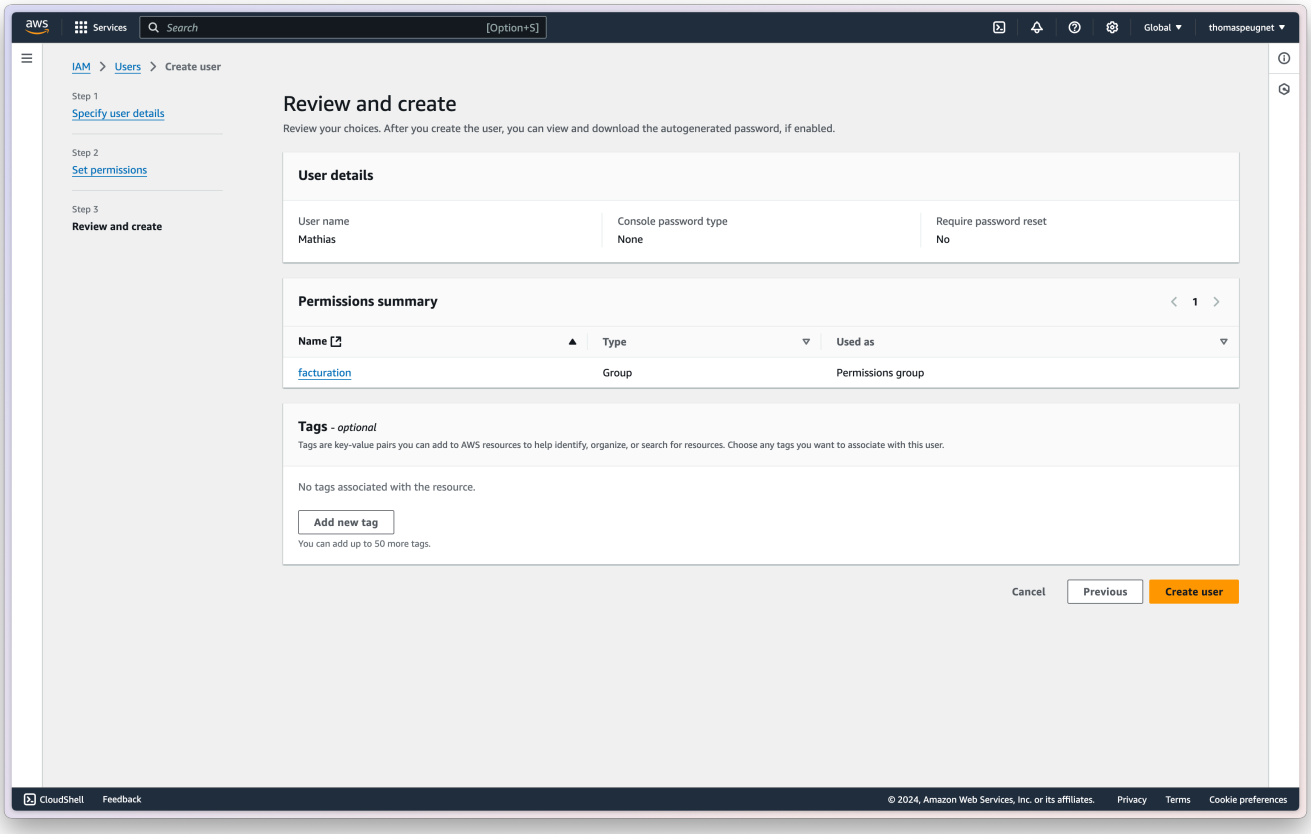
Nous activons l'accès des utilisateurs IAM aux informations de facturation.



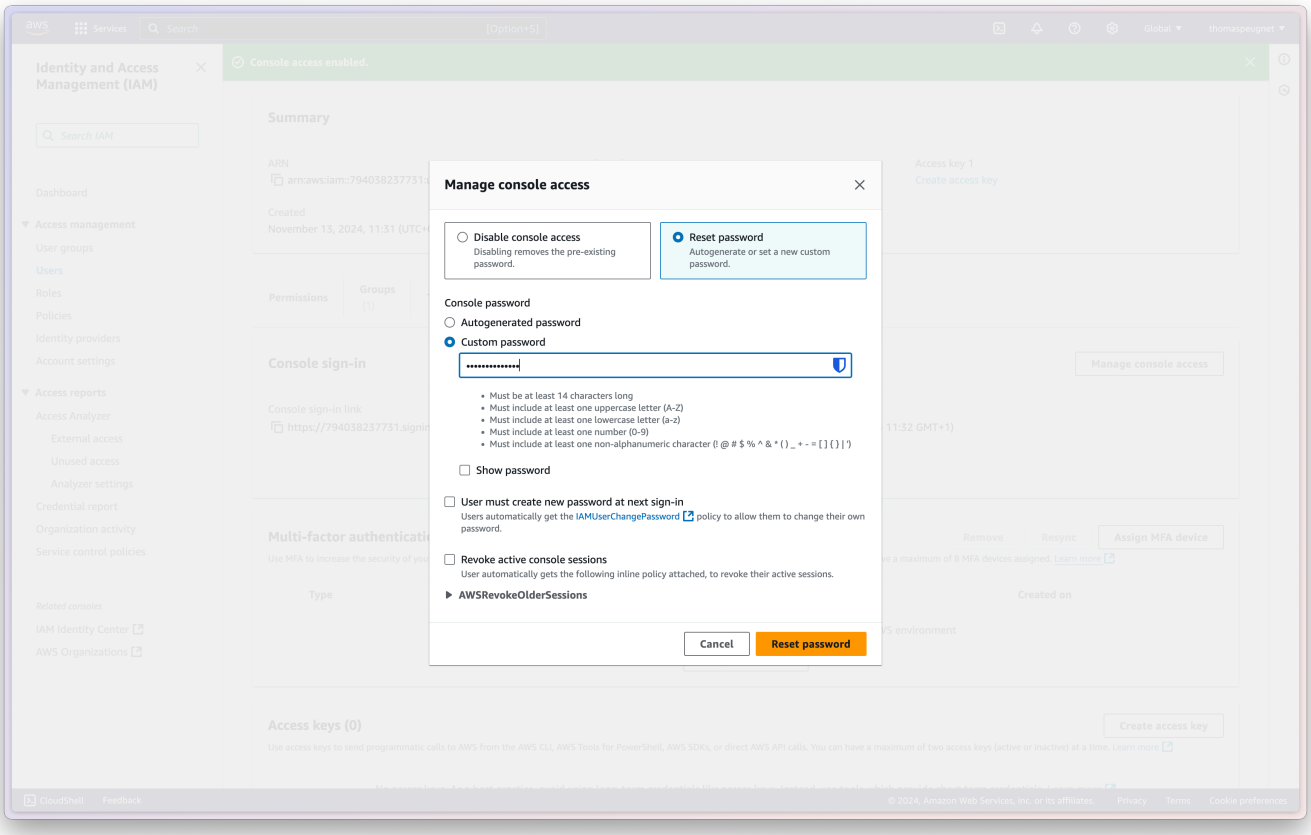
Nous ajoutons un groupe **facturation** et nous le rattachons aux permissions `Billing`.



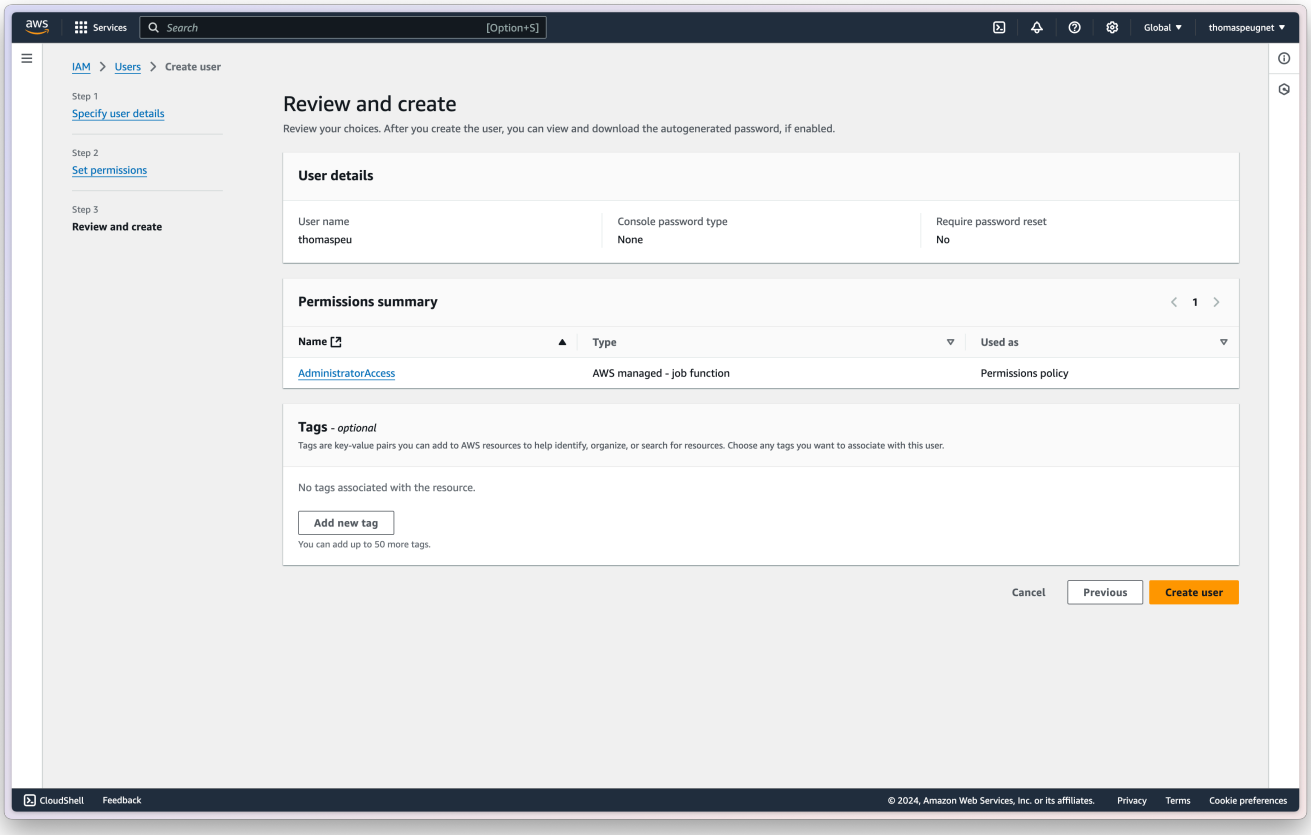
Nous créons un utilisateur `Mathias` et le mettons dans le groupe **facturation**.

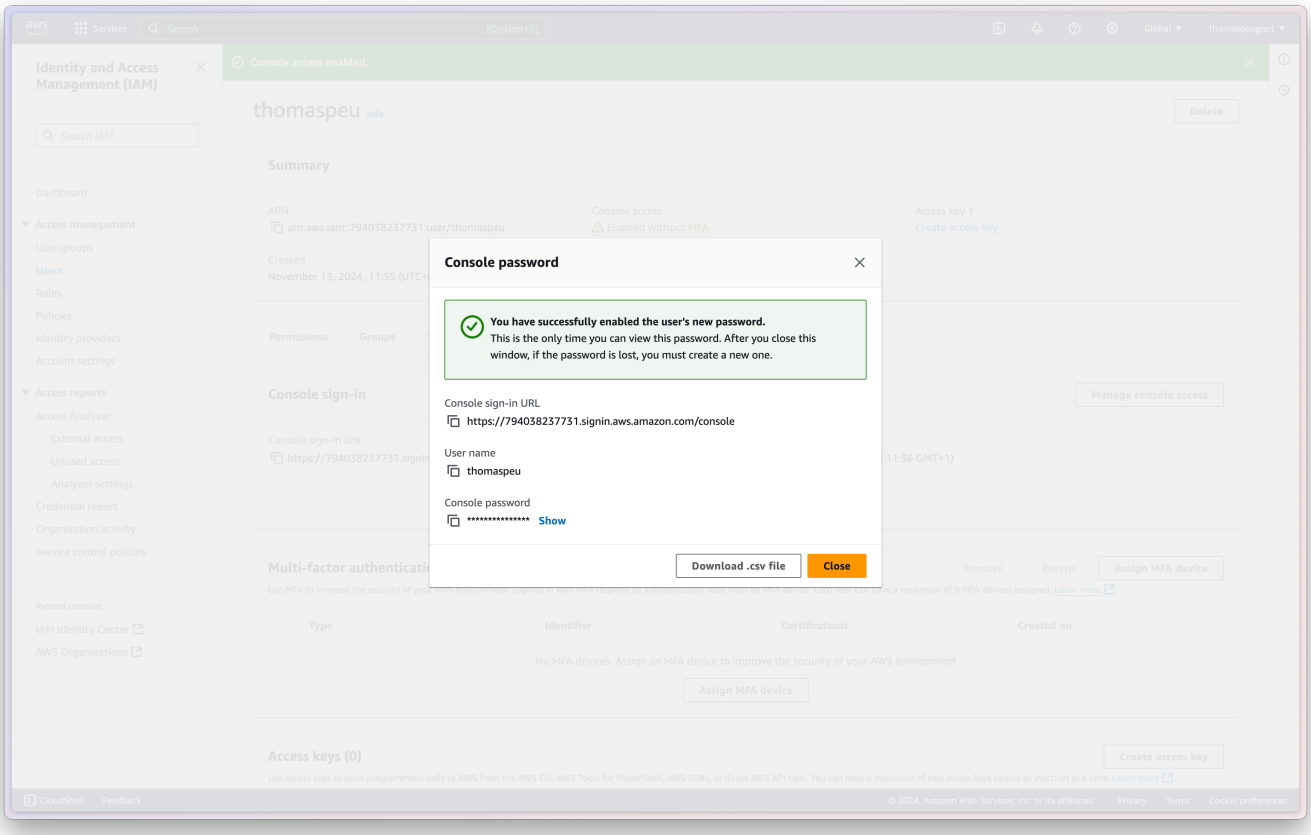


Nous activons l'accès console pour notre utilisateur.



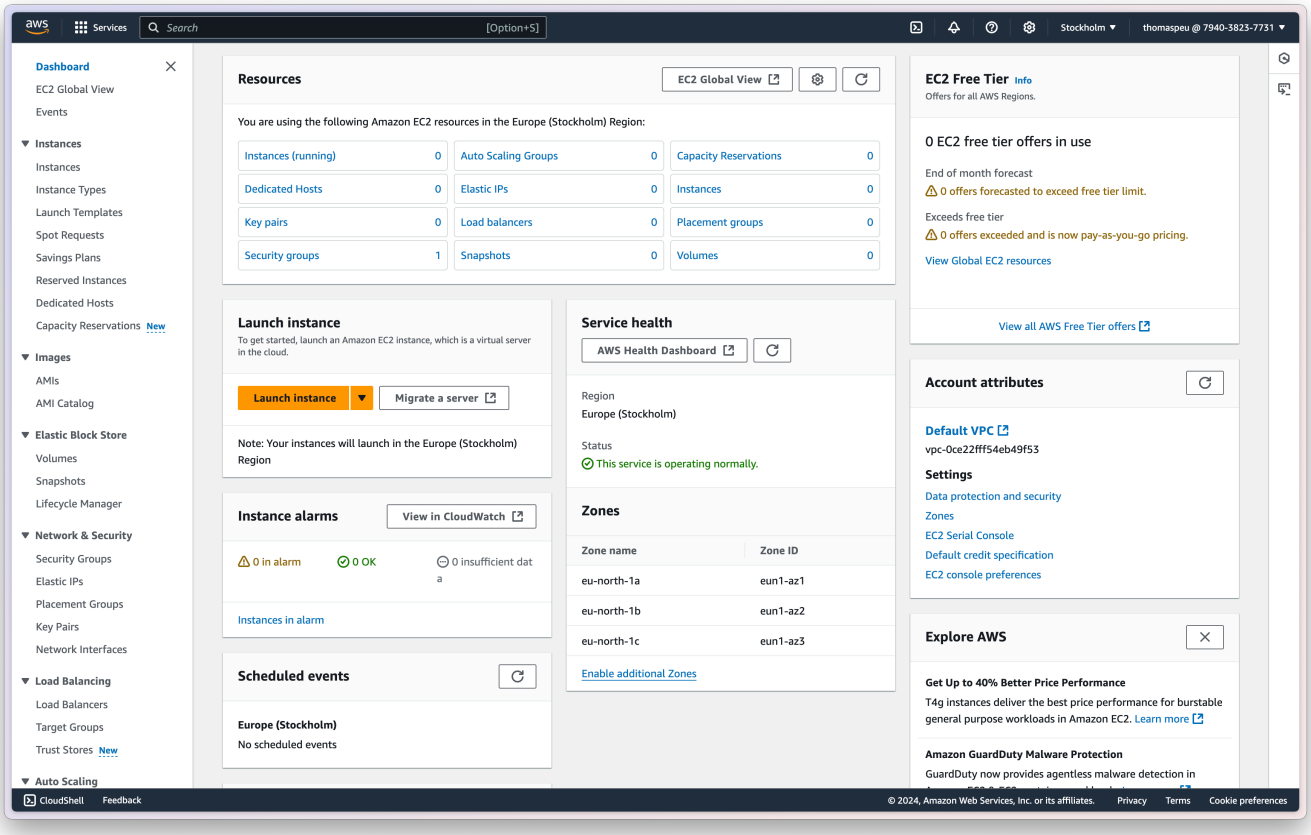
Nous créons un utilisatateur `thomaspeu`, l'attachons à la politique `AdminAccess` et lui activons son accès console.



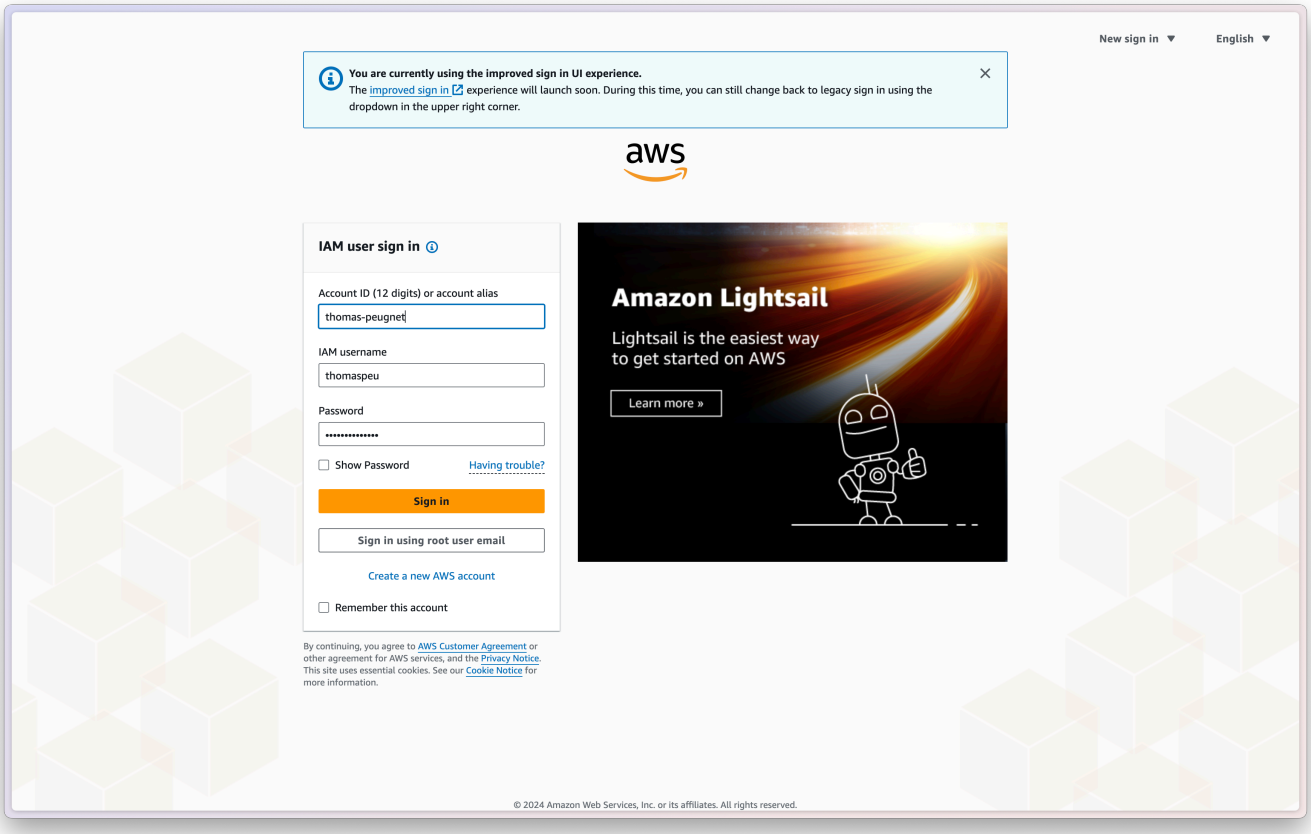


Nous activons l'utilisation de la MFA pour cet utilisateur et nous reconnectons avec ce dernier.

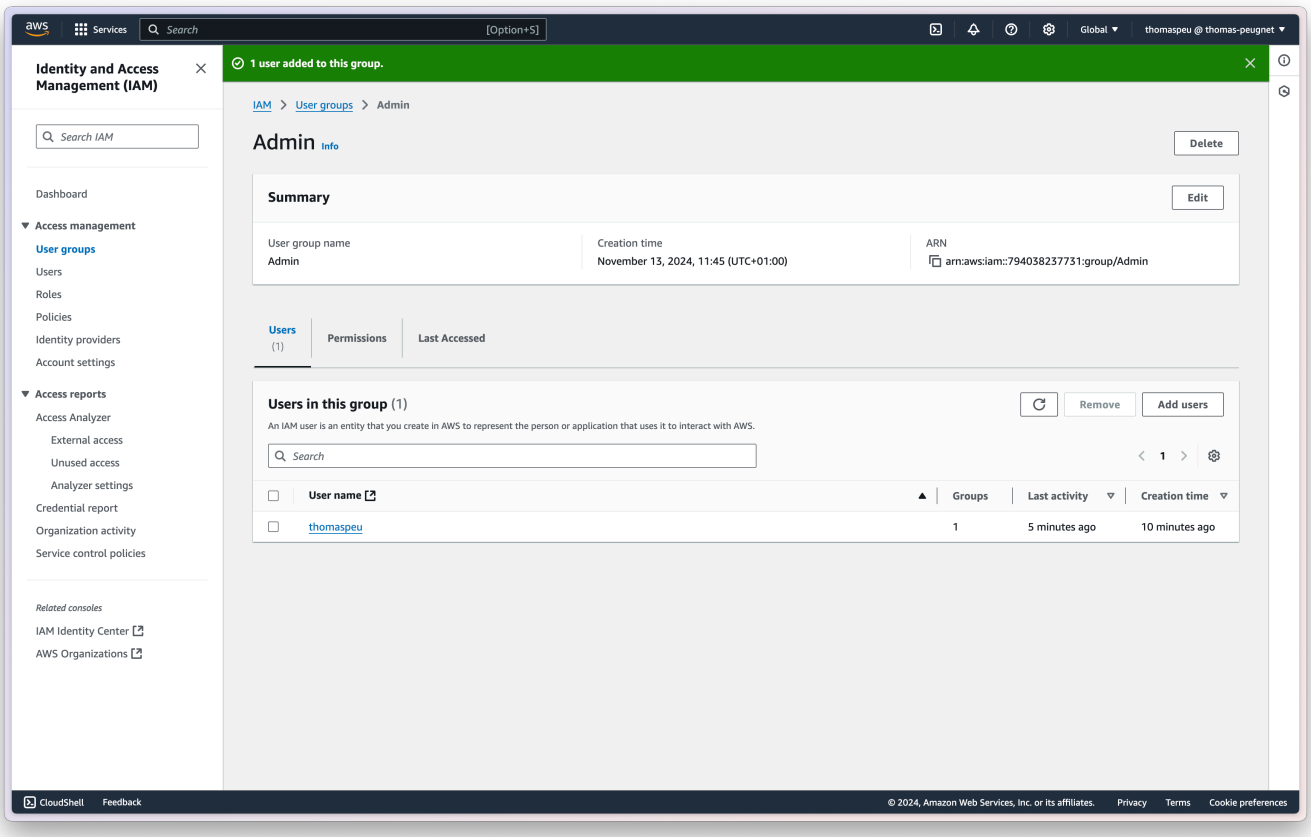
Nous pouvons constater que nous avons bien toutes les permissions nécessaires.



Nous créons un alias `thomas-peugnet` pour notre compte AWS et nous reconnectons.

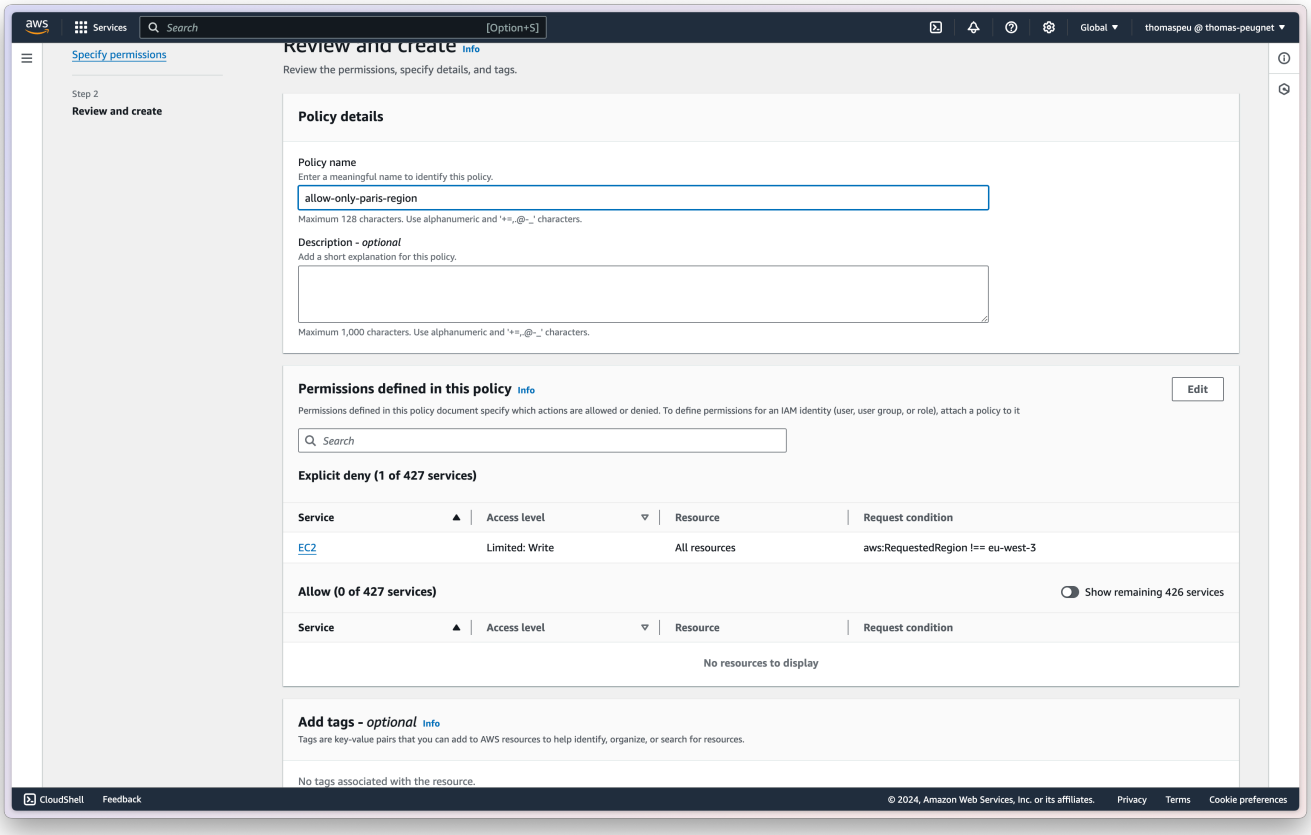


Nous créons le groupe `Admin` et y ajoutons notre utilisateur.

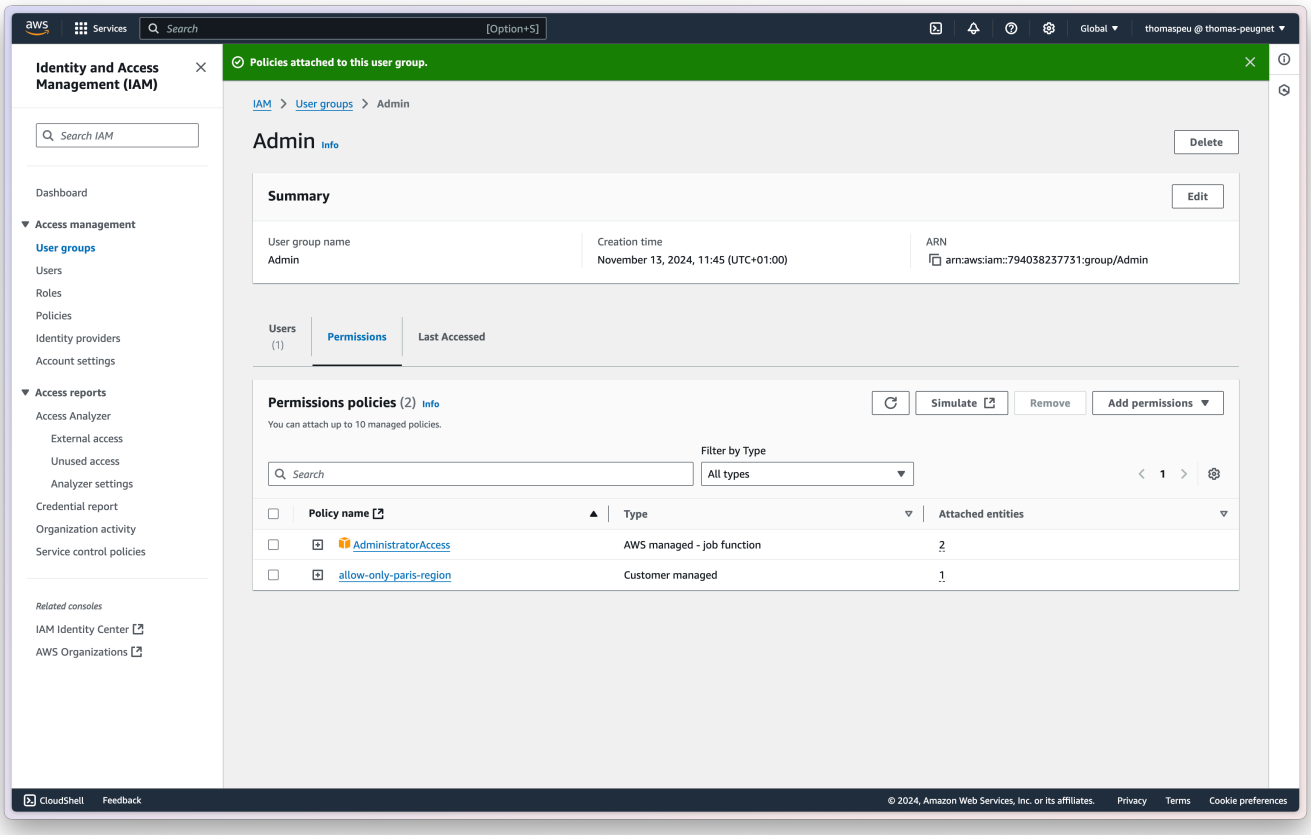


Nous créons maintenant une politique pour rester dans le Free Tier qui aura le JSON suivant. Son objectif est d'interdire le lancement des VM si elles ne sont pas dans la région de Paris.

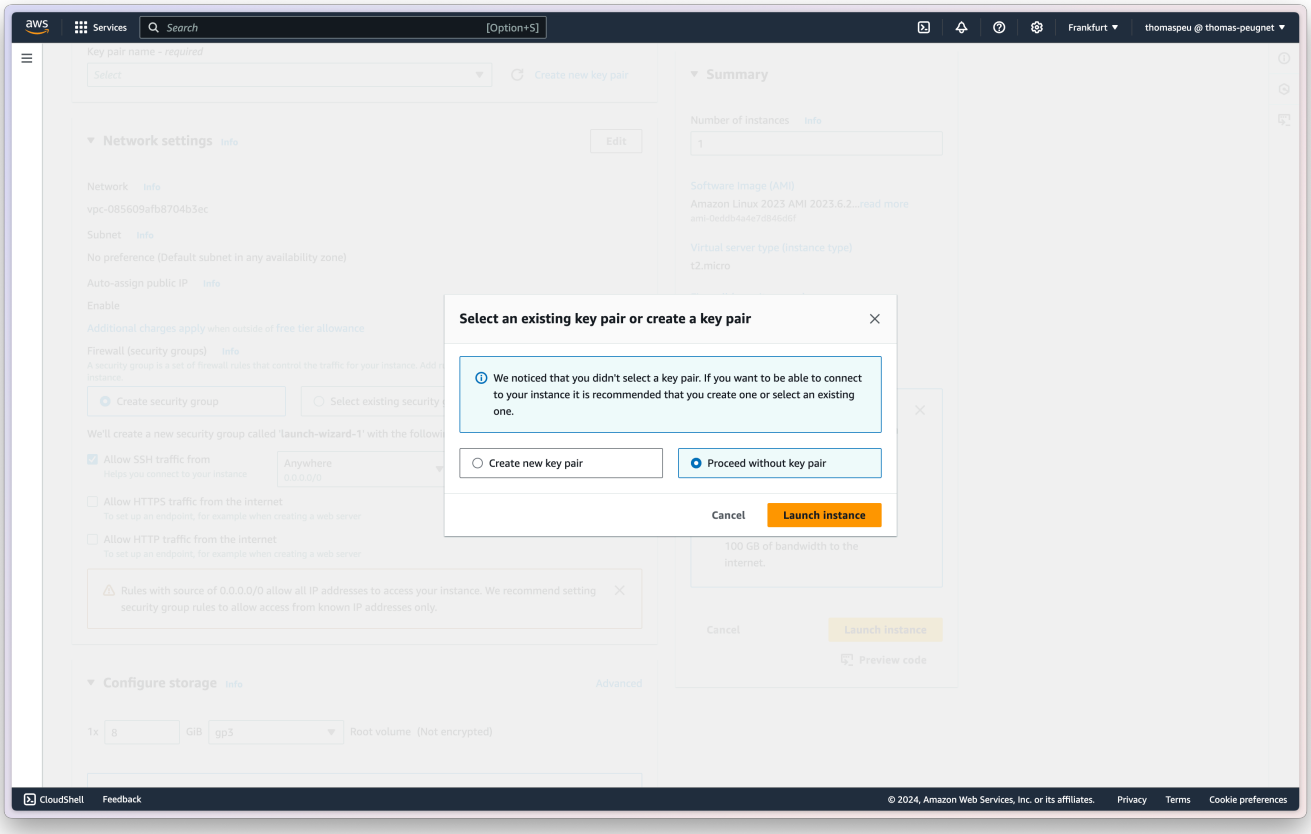
```
{
  "Version": "2012-10-17",
  "Statement": [
                    {
                        "Effect": "Deny",
                        "Action": "ec2:RunInstances",
                        "Resource": "*",
                        "Condition": {
                            "StringNotEquals": {
                                "aws:RequestedRegion": "eu-west-3"
                            }
                        }
                    }
  ]
}
```



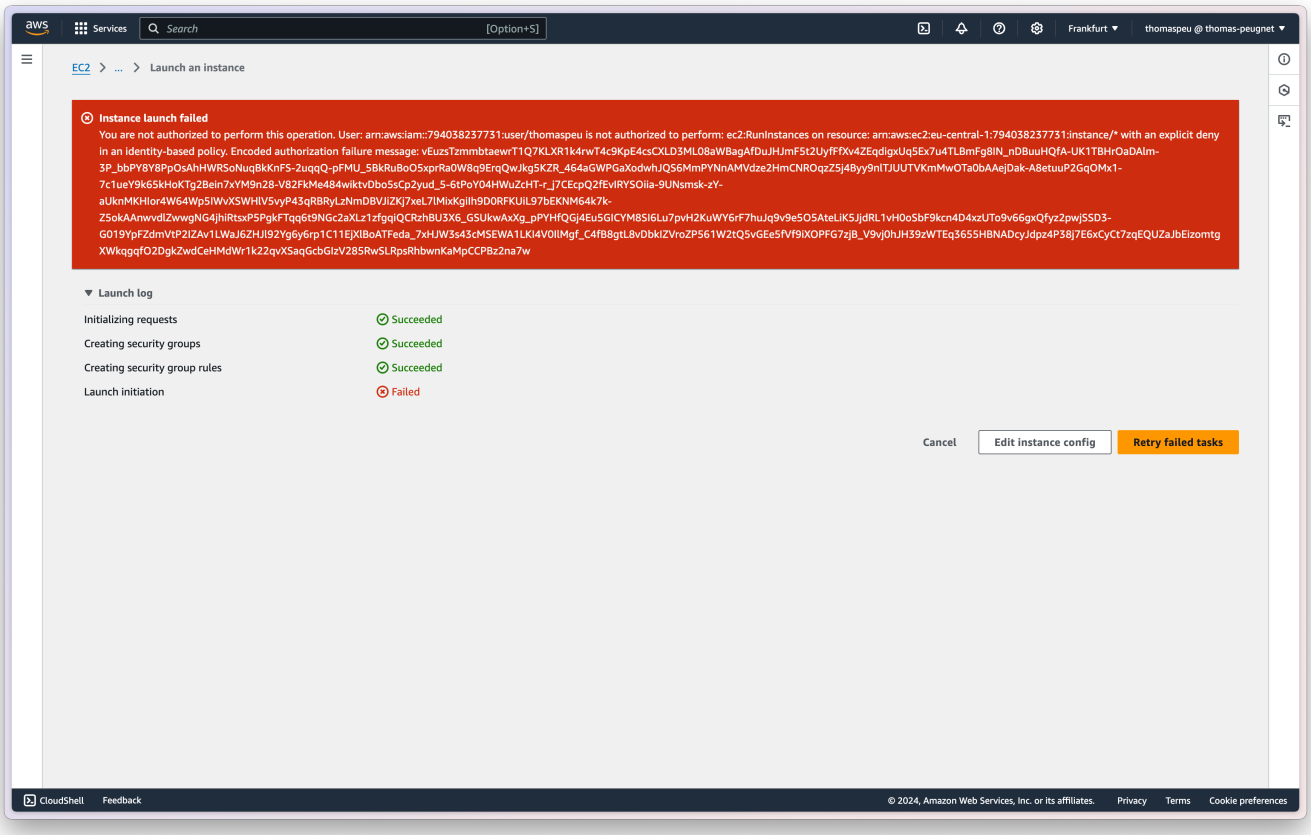
Nous attachons cette nouvelle politique `allow-only-paris-region` à notre groupe `Admin`.



Nous testons de lancer une instance `t2.micro` hors de France (Frankfurt).



Comme prévu, nous avons un message d'erreur.

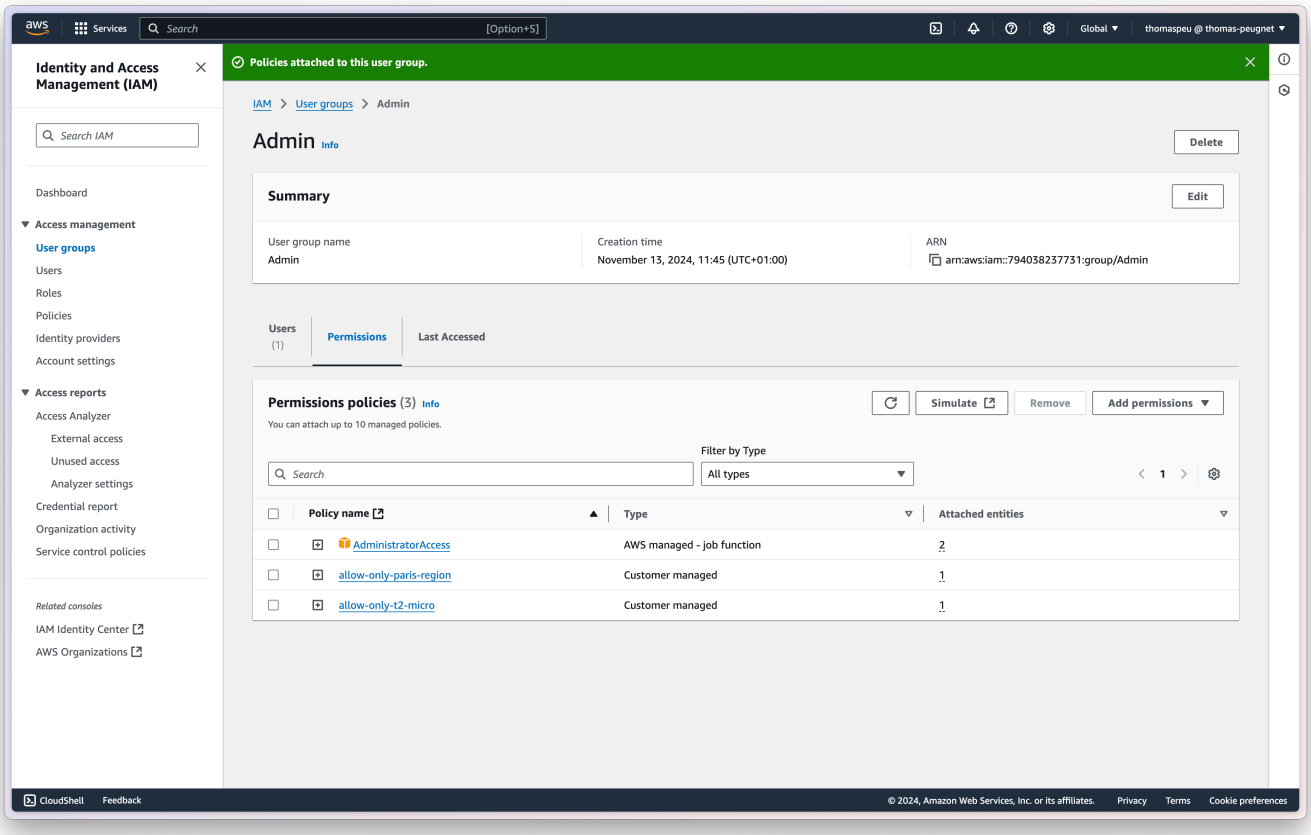


Nous créons ensuite la politique `allow-only-t2-micro` ayant pour objectif d'interdire le lancement de VM différentes du type `t2.micro`. Cette politique aura le JSON suivant:

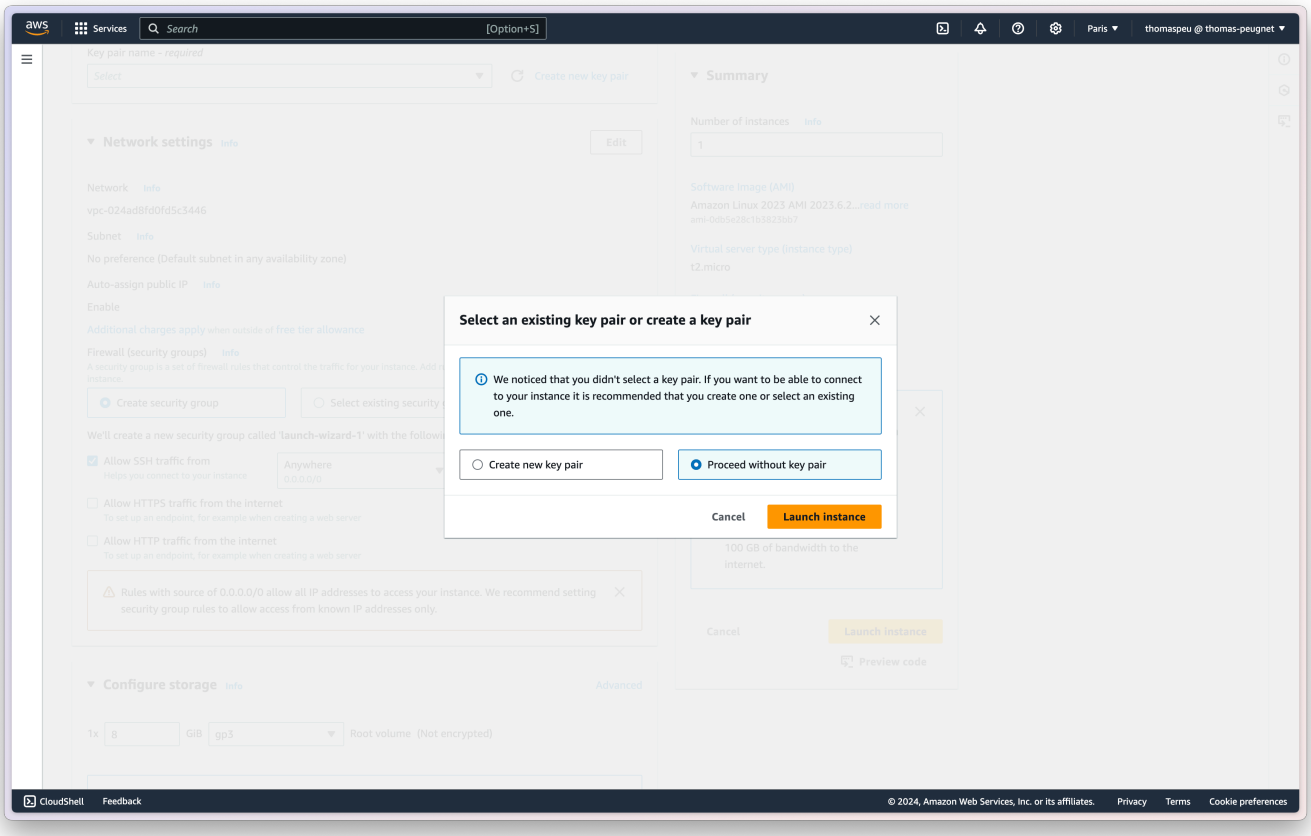
```
{
  "Version": "2012-10-17",
  "Statement": [
                    {
                        "Effect": "Deny",
                        "Action": "ec2:RunInstances",
                        "Resource": "*",
                        "Condition": {
                            "StringNotEquals": {
                                "ec2:InstanceType": "t2.micro"

                            }
                        }
                    }
  ]
}
```

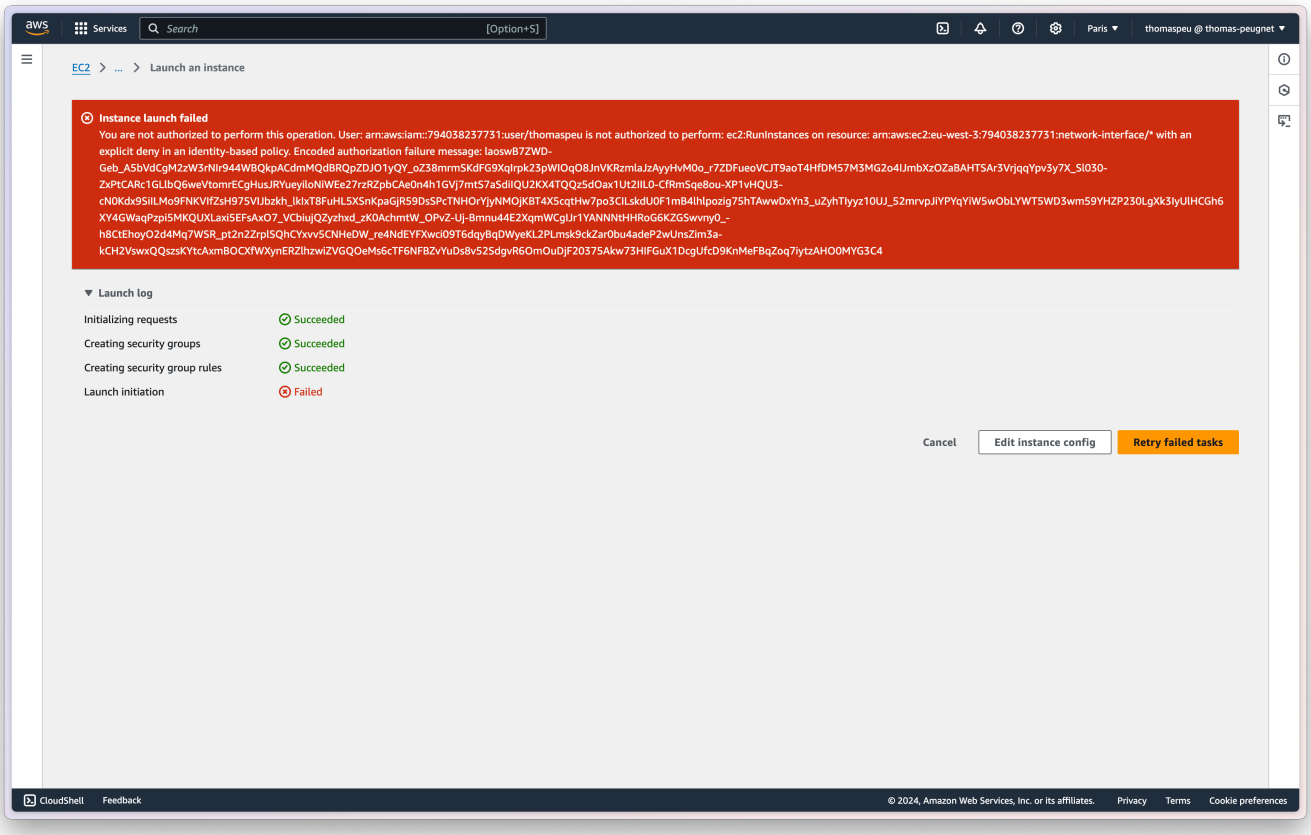
Nous associons cette nouvelle politique à notre groupe `Admin`.



Nous testons notre politique avec le lancement d'une instance t2.micro en se situant dans la région Paris.



Nous obtenons, comme prévu, un message d'erreur.

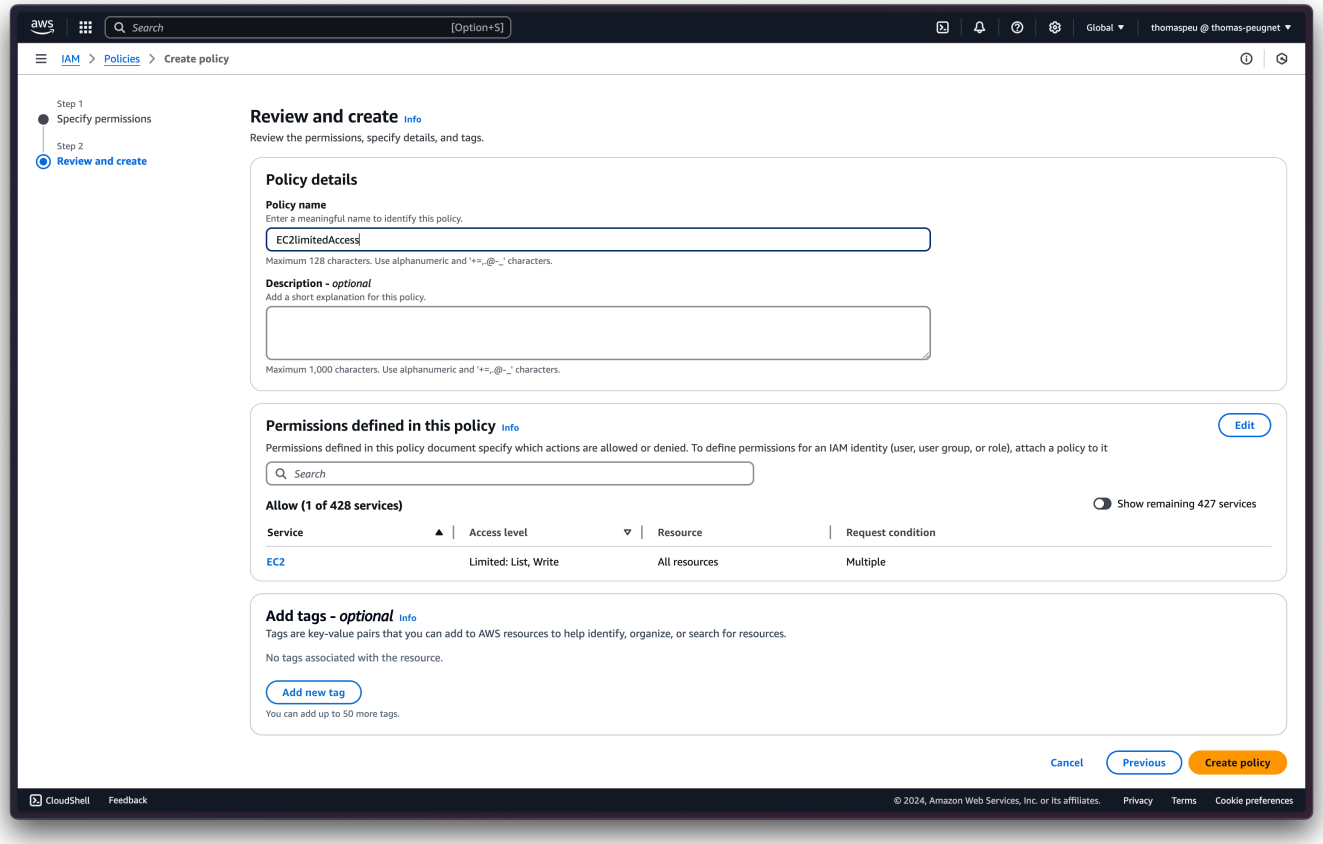


Nous allons maintenant gérer les autorisations AWS en nous basans sur les attributs.

Nous créons une Politique `EC2limitedAccess` avec le JSON suivant:

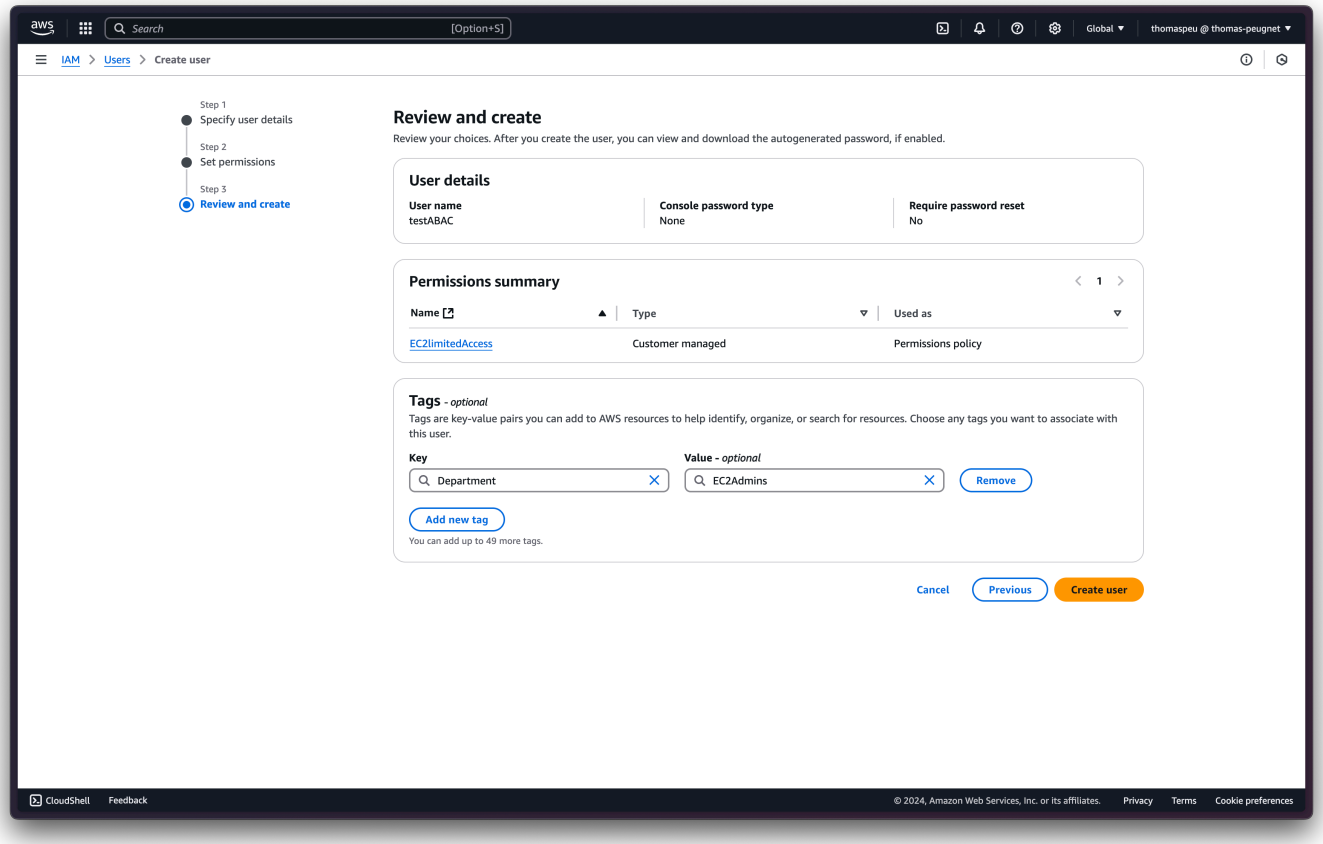
```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeInstances",
            "ec2:DescribeImages",
            "ec2:DescribeTags"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:RebootInstances",
            "ec2:StartInstances",
            "ec2:StopInstances"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
```

```
                "aws:PrincipalTag/Department":"EC2Admins",
                "ec2:ResourceTag/Environment":"Production"
            }
        }
    }
  ]
}
```

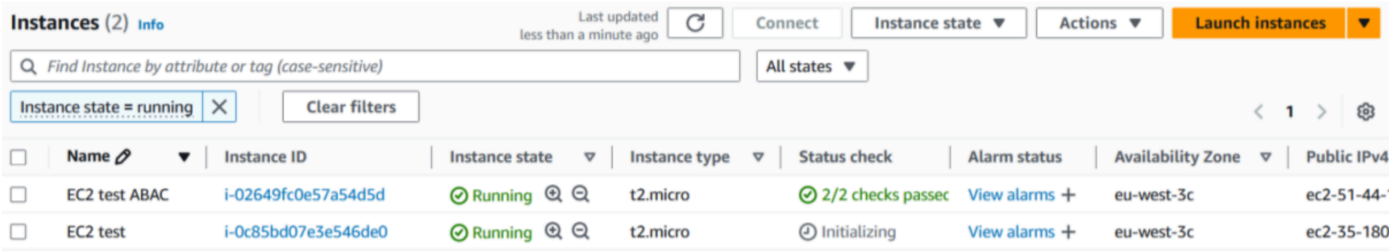


Nous créons ensuite un utilisateur nommé `testABAC` auquel nous assignons la politique `EC2limitedAccess`. Nous assignons également les tags suivant:

- `key` = Department
- `value` = EC2Admins



Nous lançons ensuite nos 2 instances EC2 avec les tags `Environment:Production` et `Environment:Development`.



Nous nous reconnectons avec l'utilisateur `testABAC` et constatons que nous pouvons bien redémarrer l'instance de Production mais pas celle de Developpement.