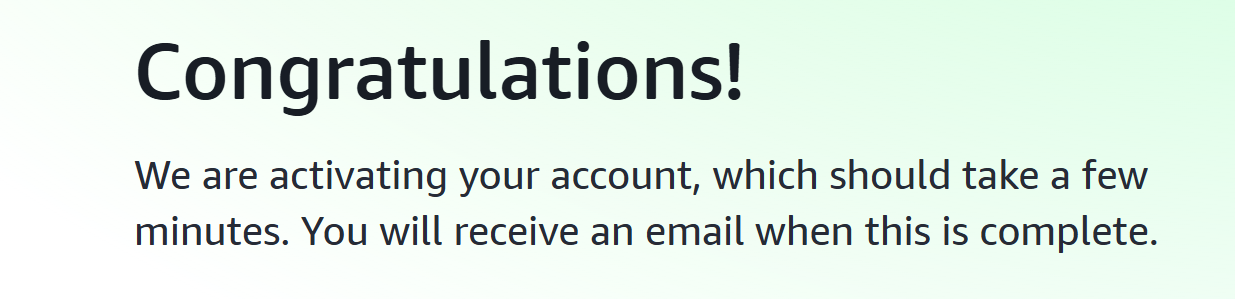
**Лабораторна робота № 1**

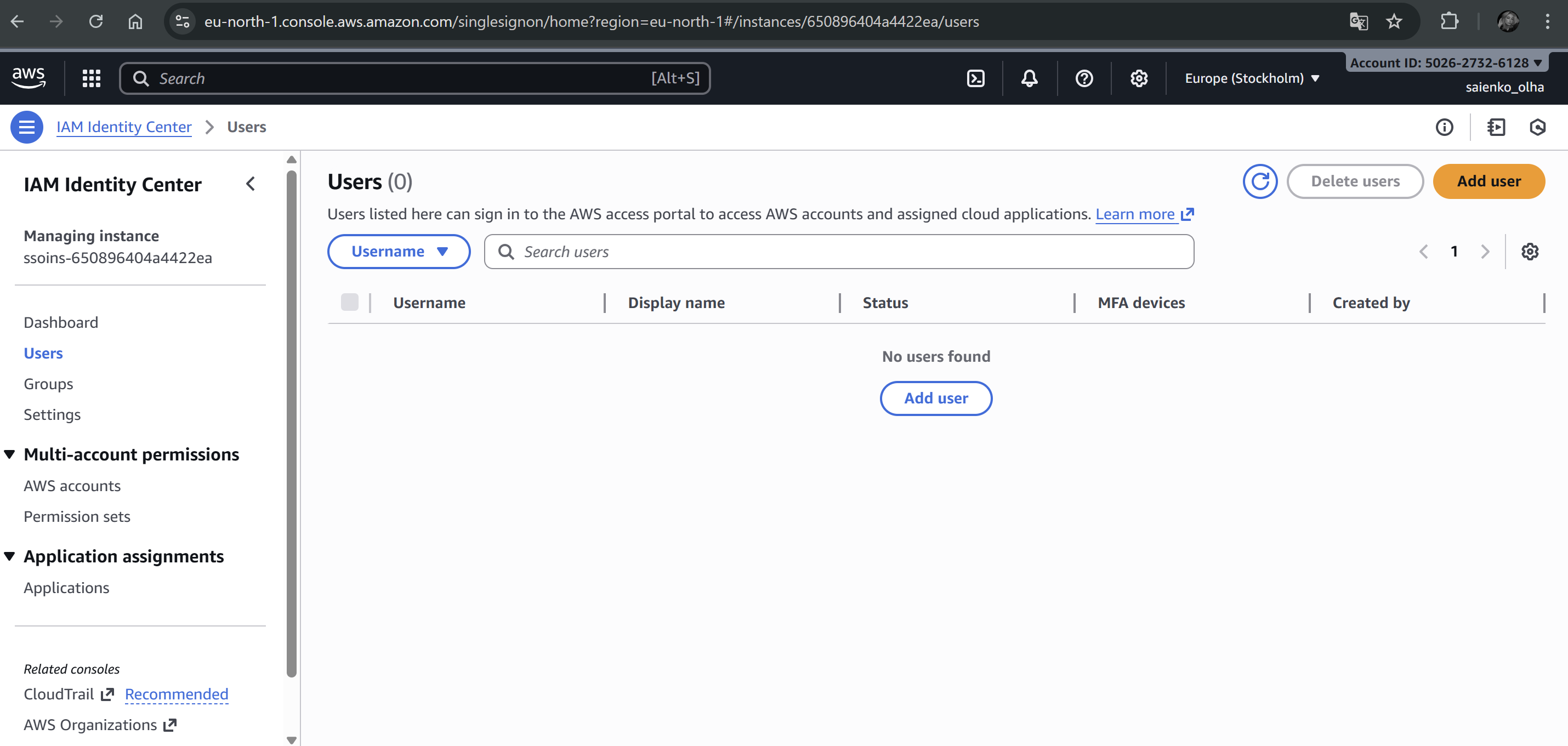
Infrastructure setup

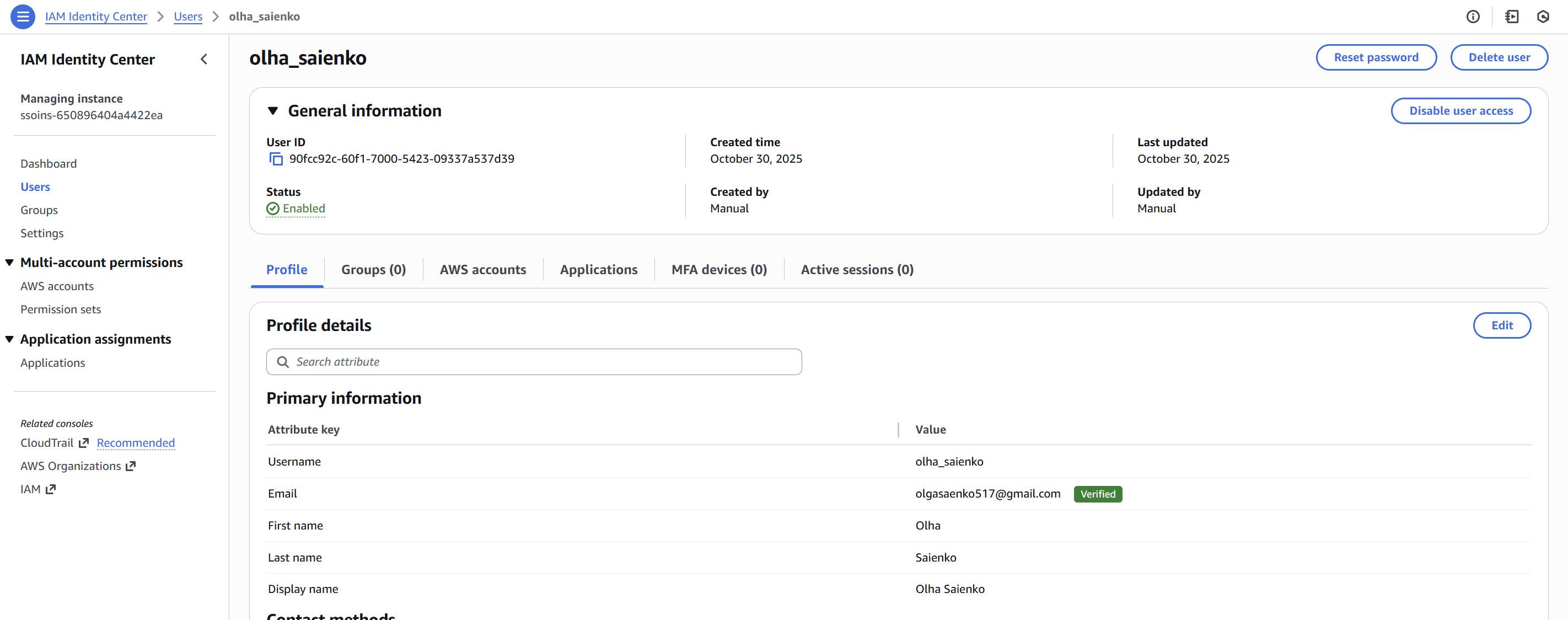
1. Реєстрація



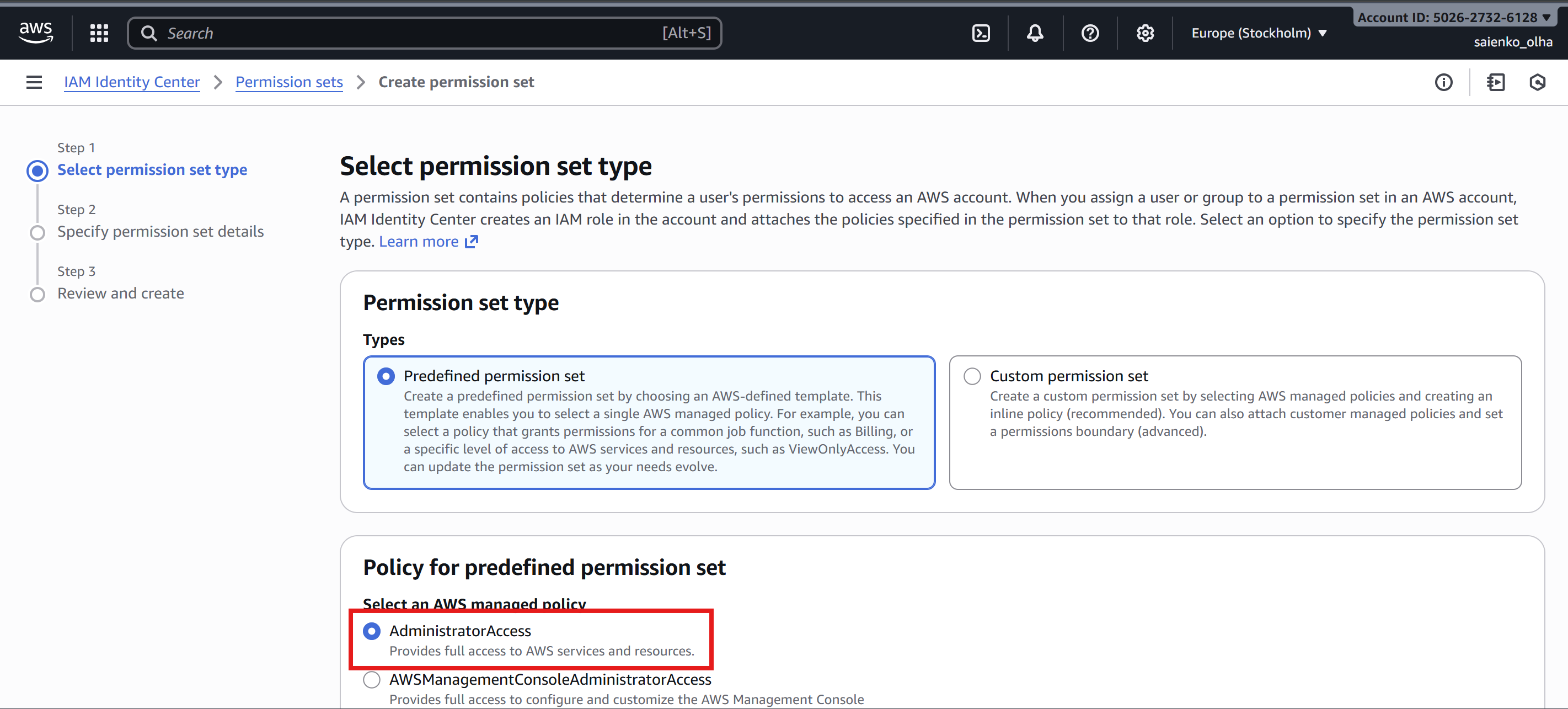
1. Identity Center (SSO) з Full Admin

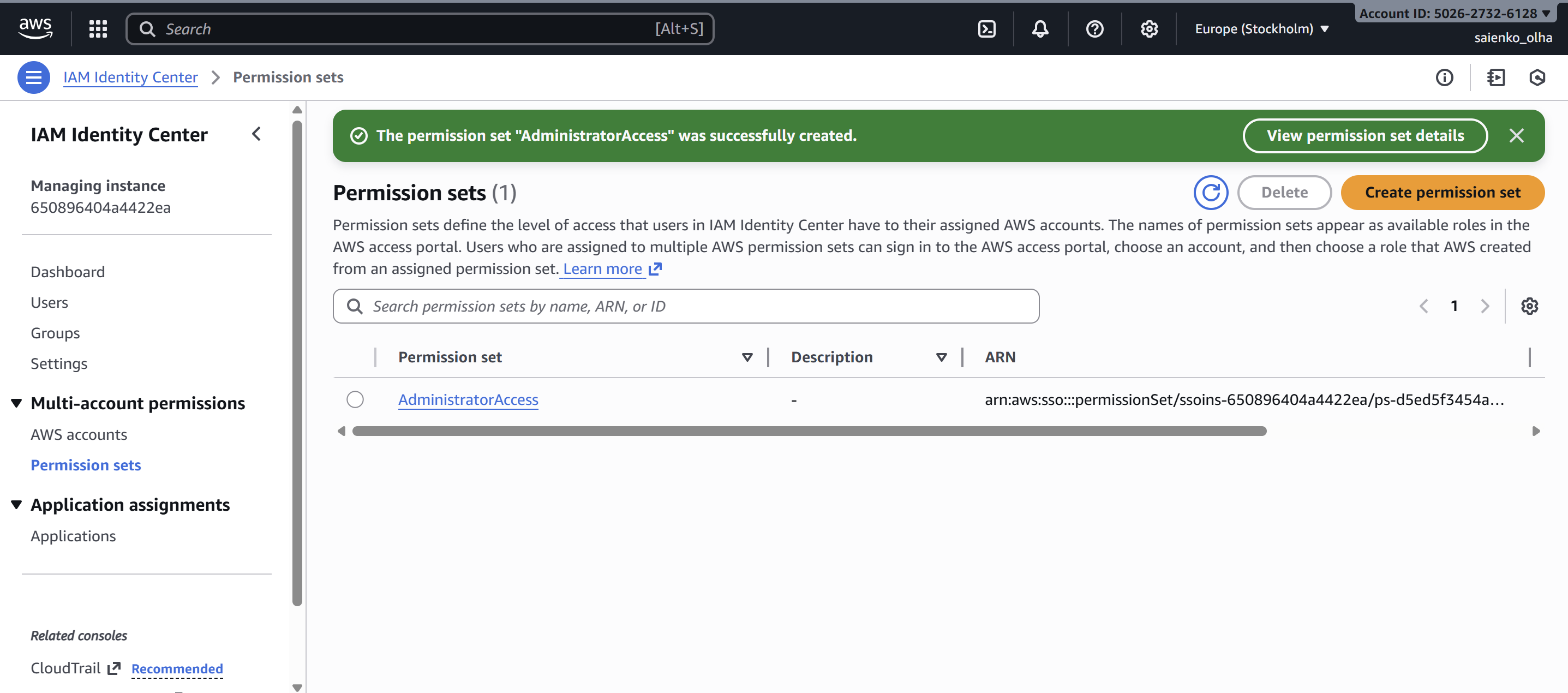
Переходимо до Identity Center і створюємо користувача



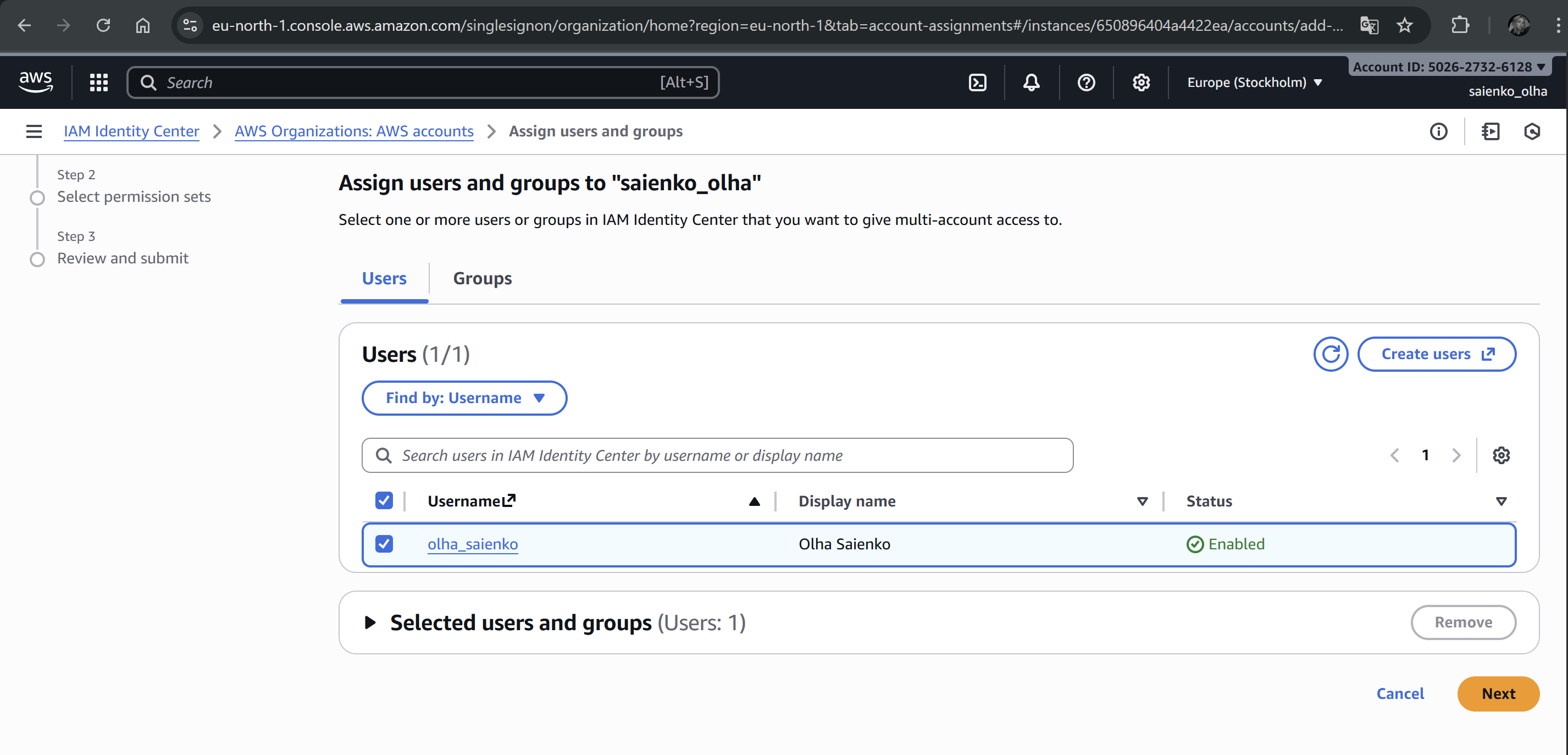


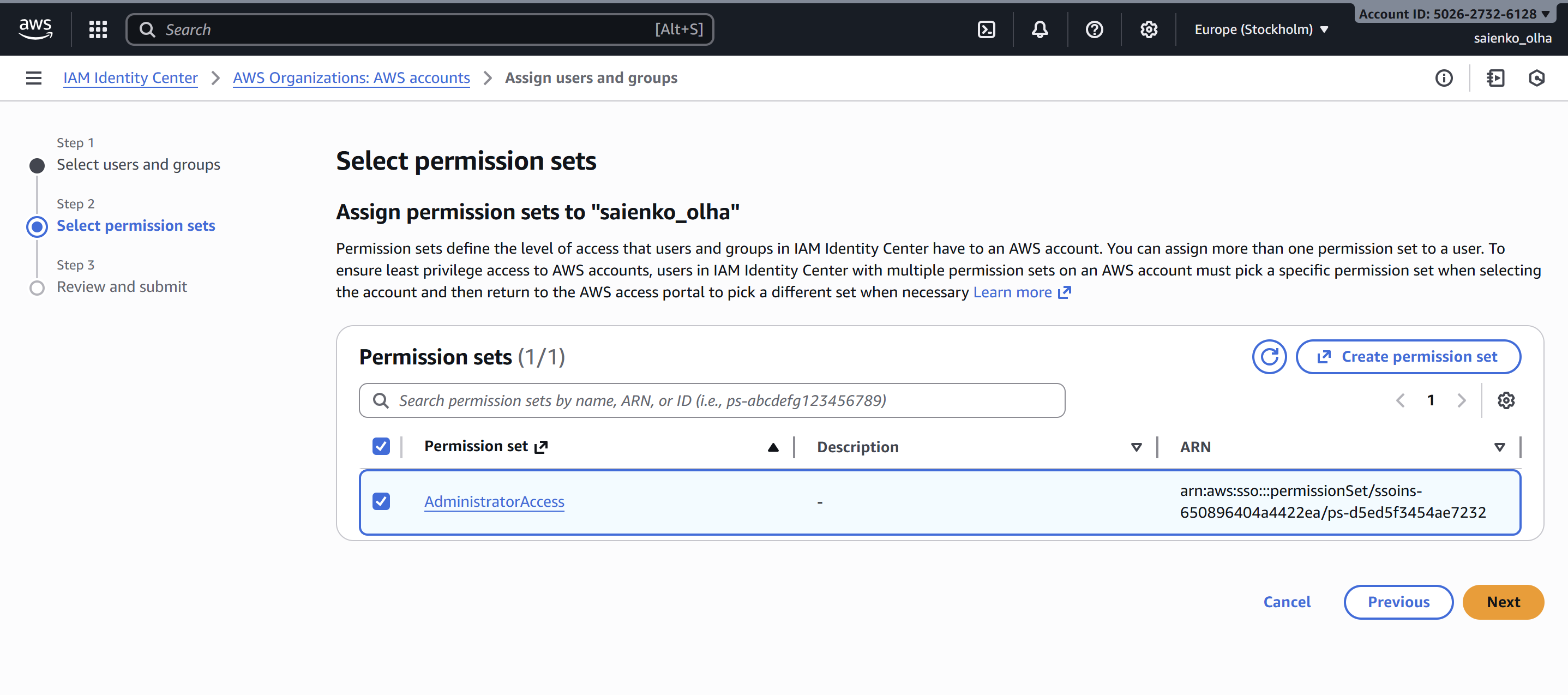
Створюємо Permission set:

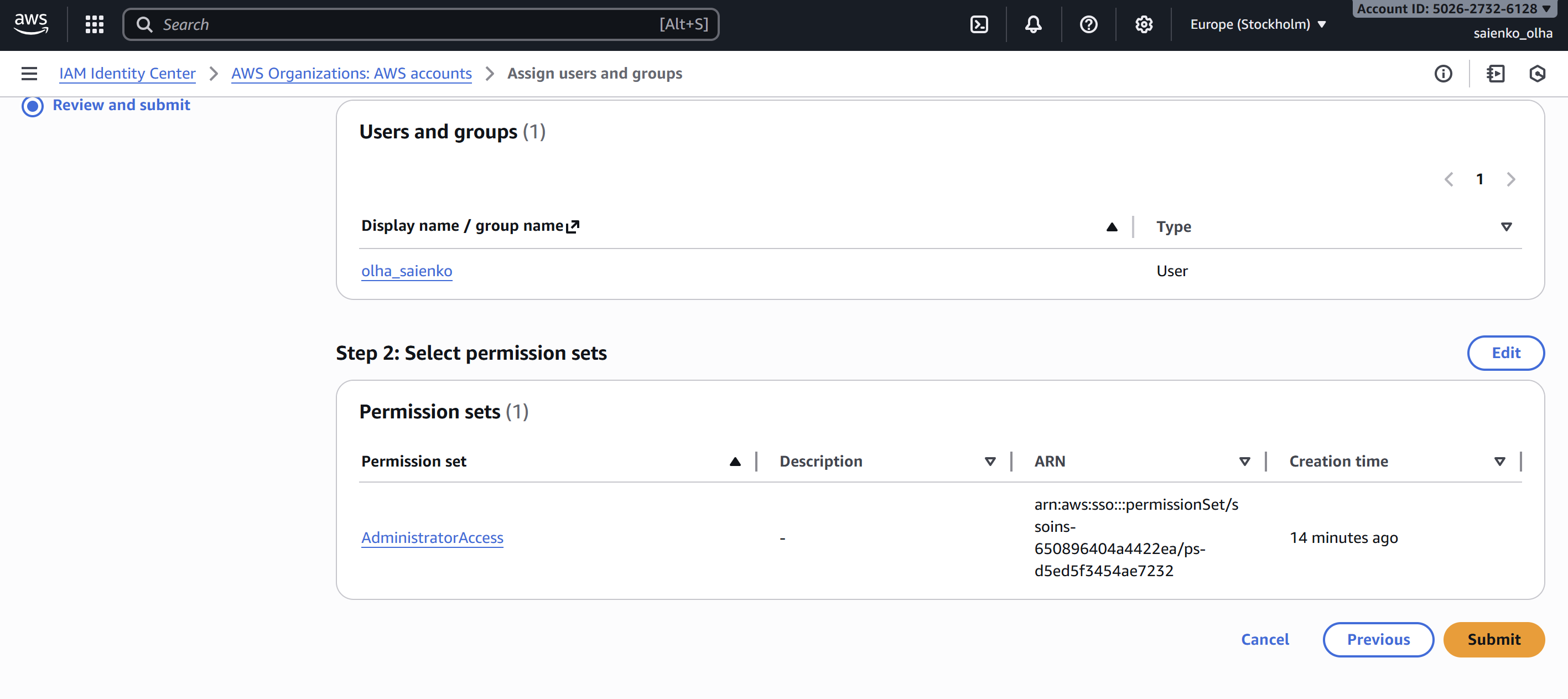


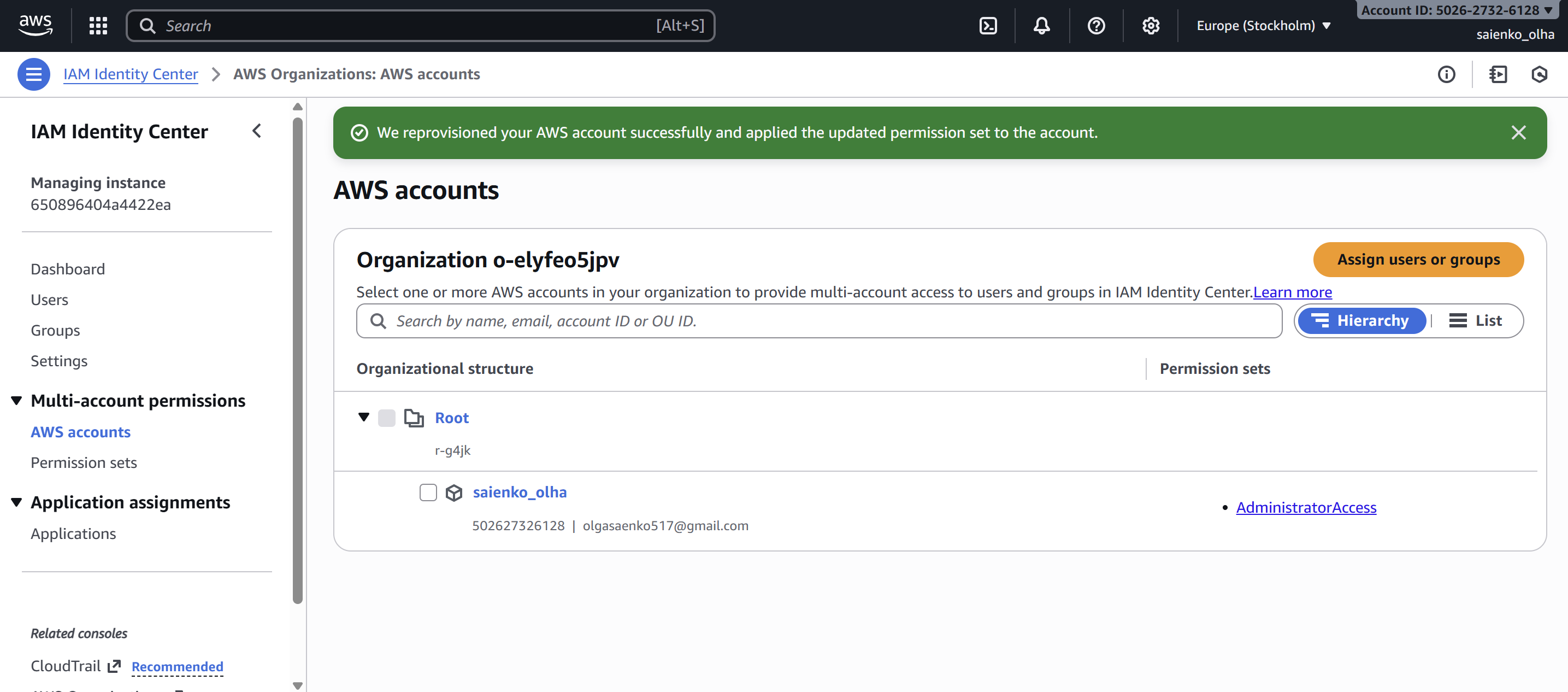


Призначаємо Permission Set користувачу:







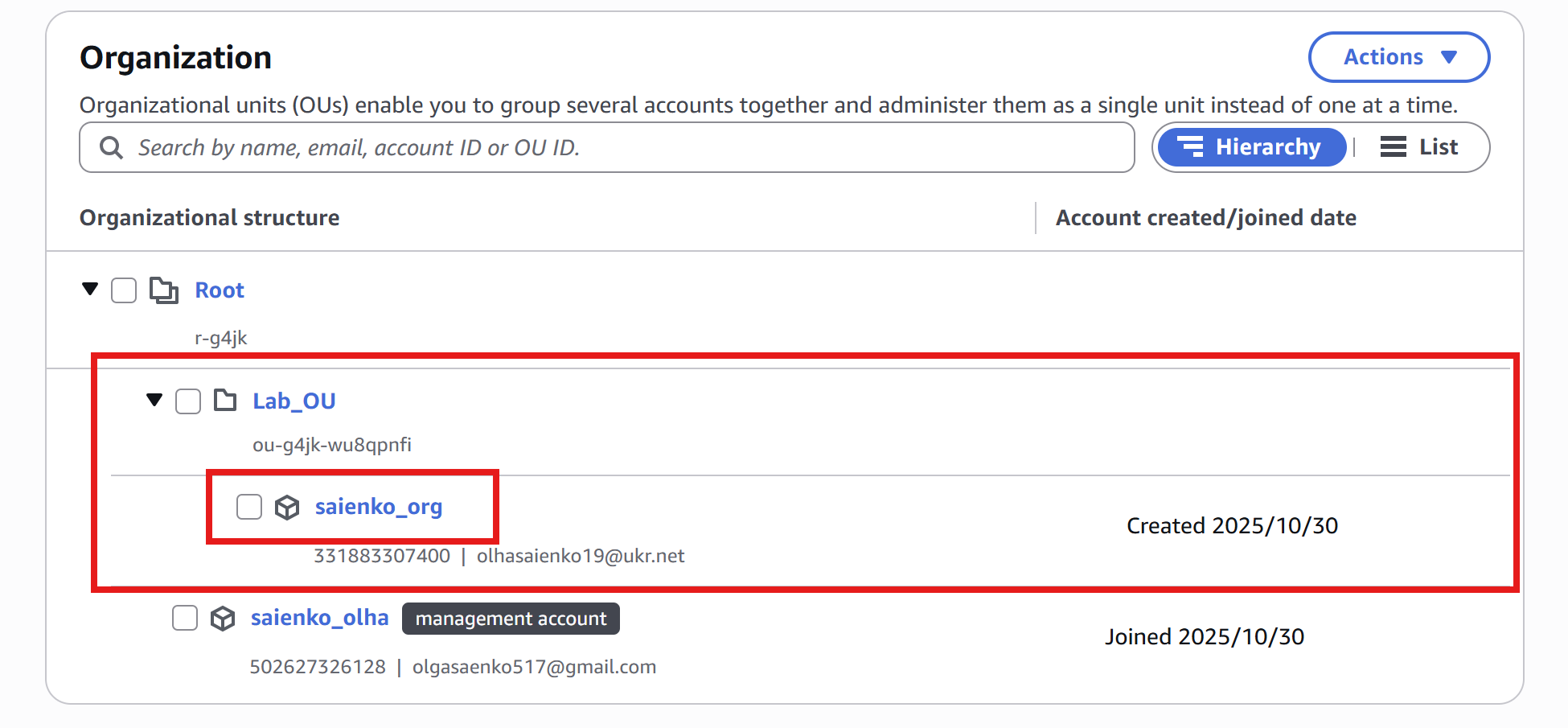


1. AWS Organization

Створимо новий акаунт:

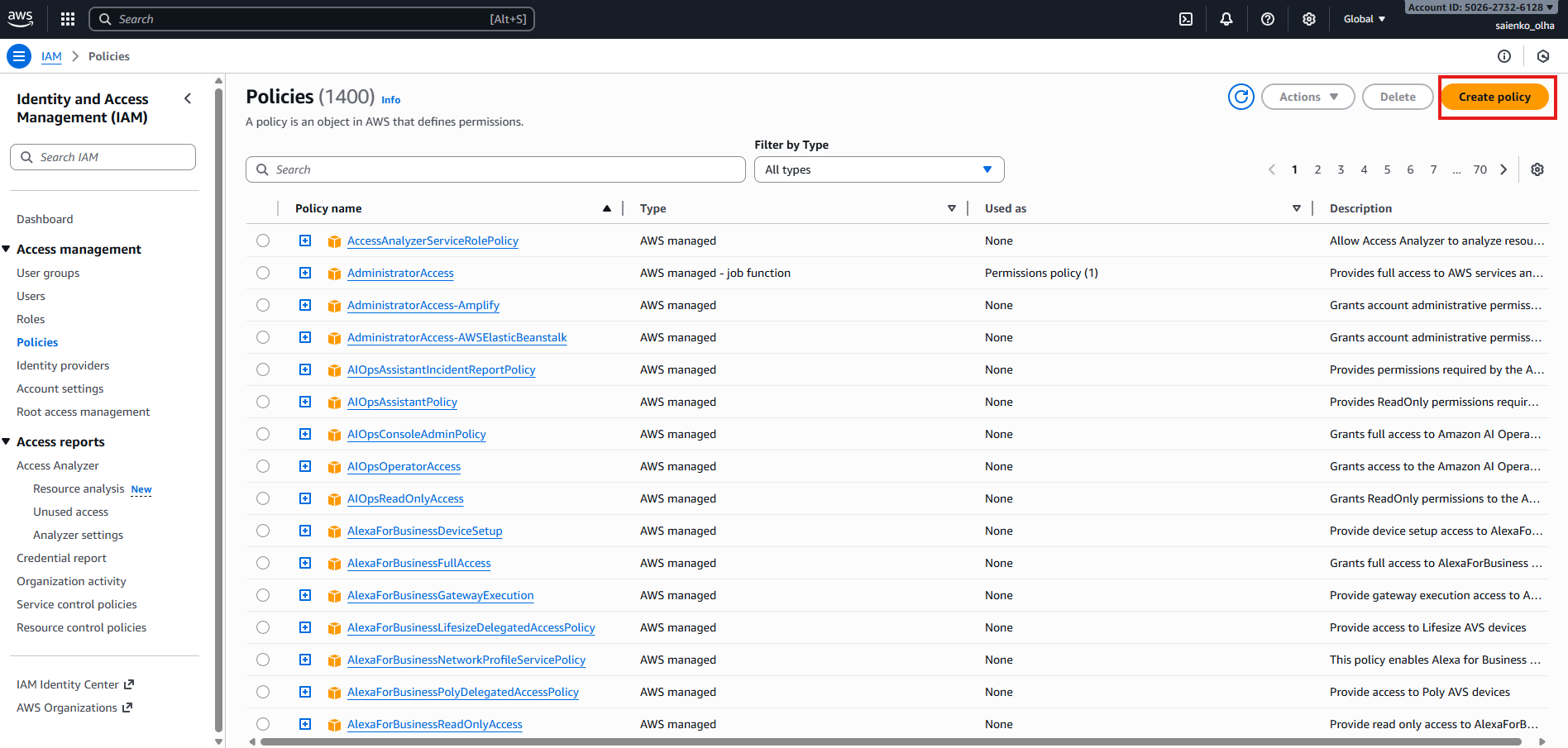


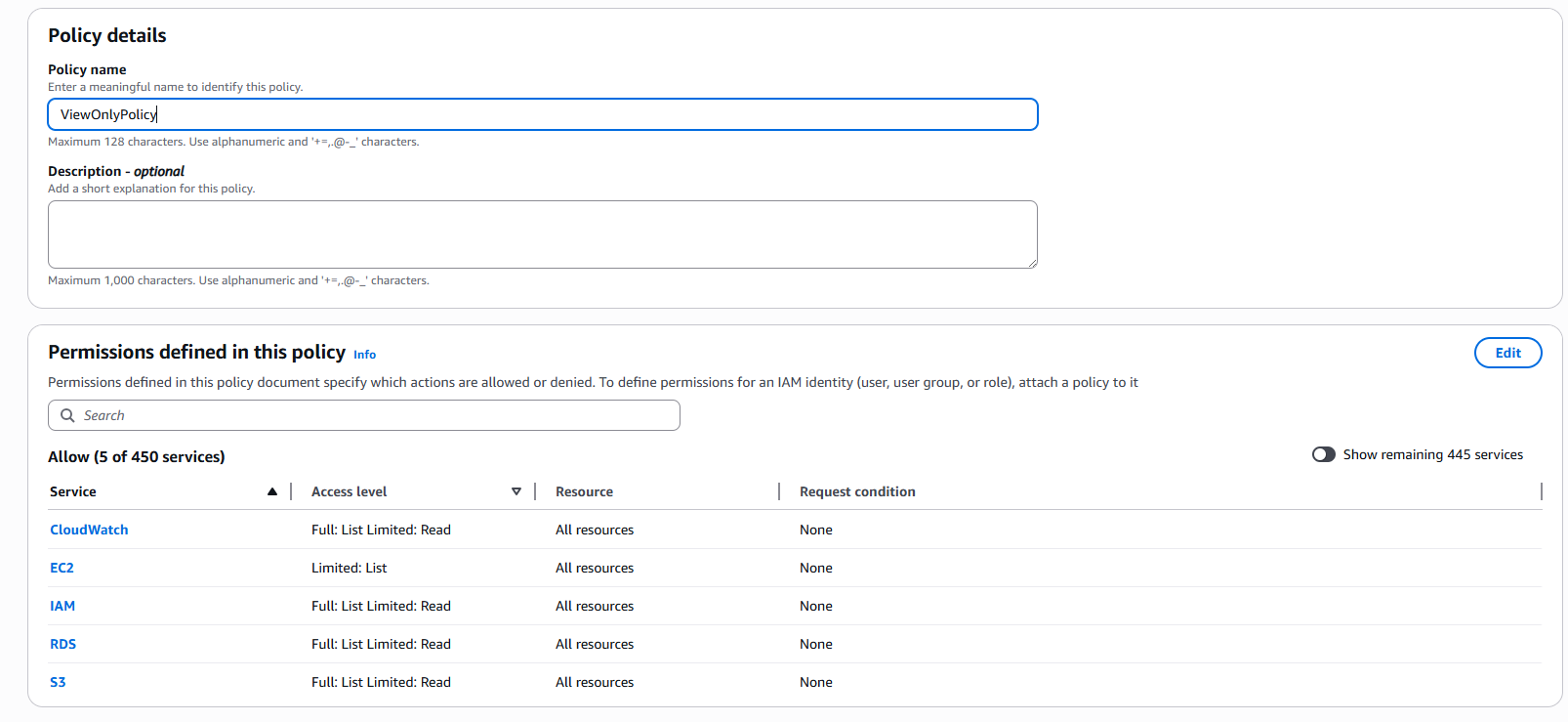
Створимо Organizational Unit (OU) в Root і перемістимо туди новий акаунт:

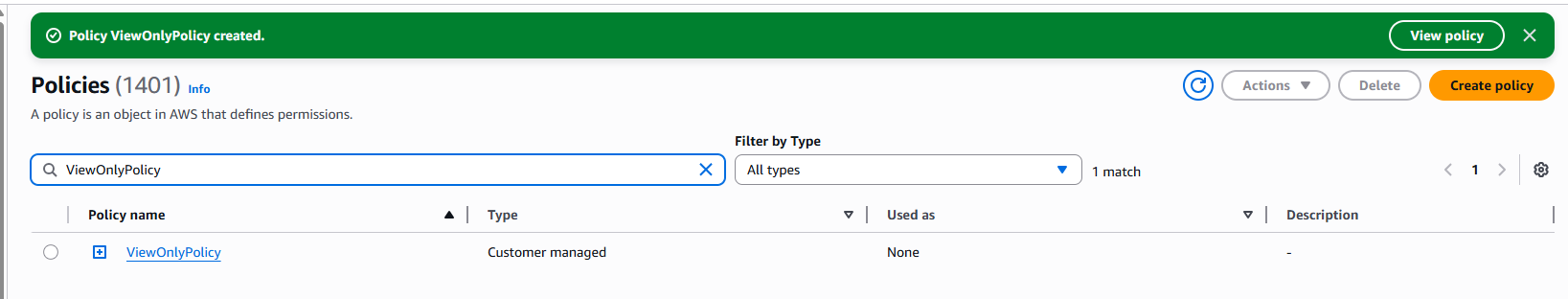


1. Створення користувача та політики лише для перегляду

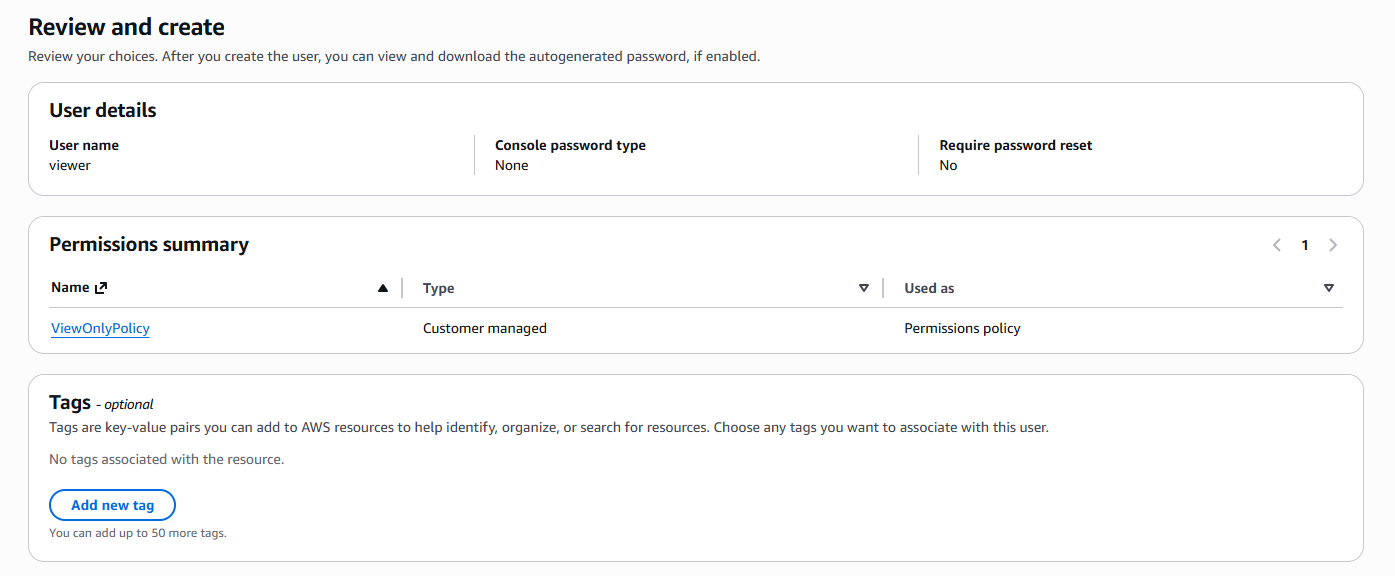
Створюємо політику:



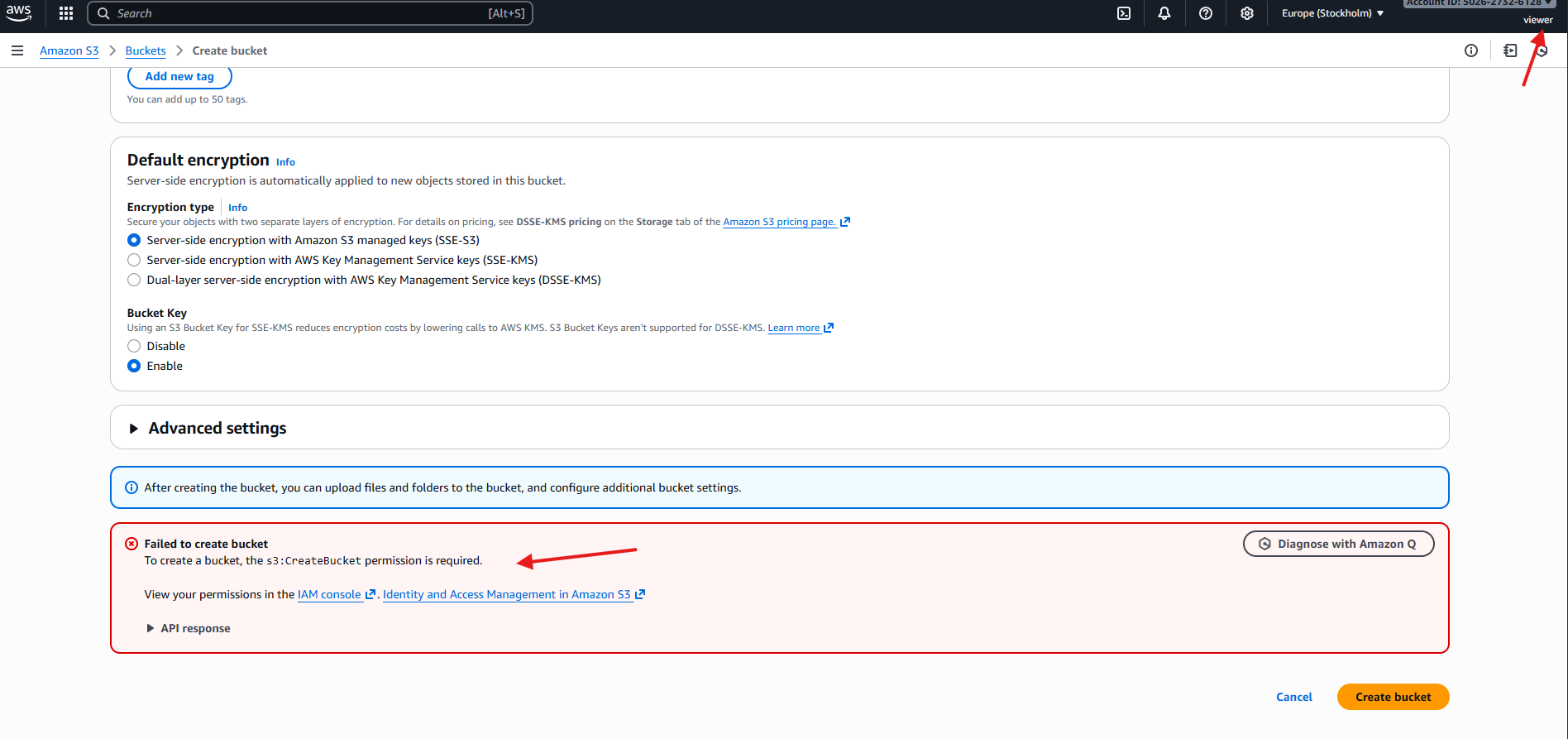


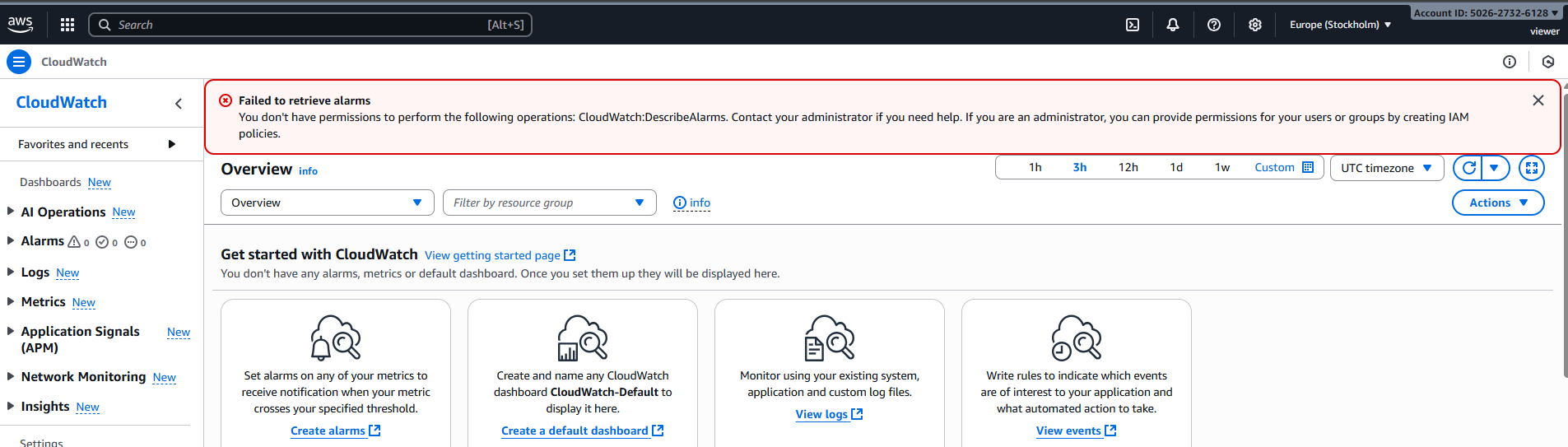


Створюємо користувача та призначаємо створену політику:



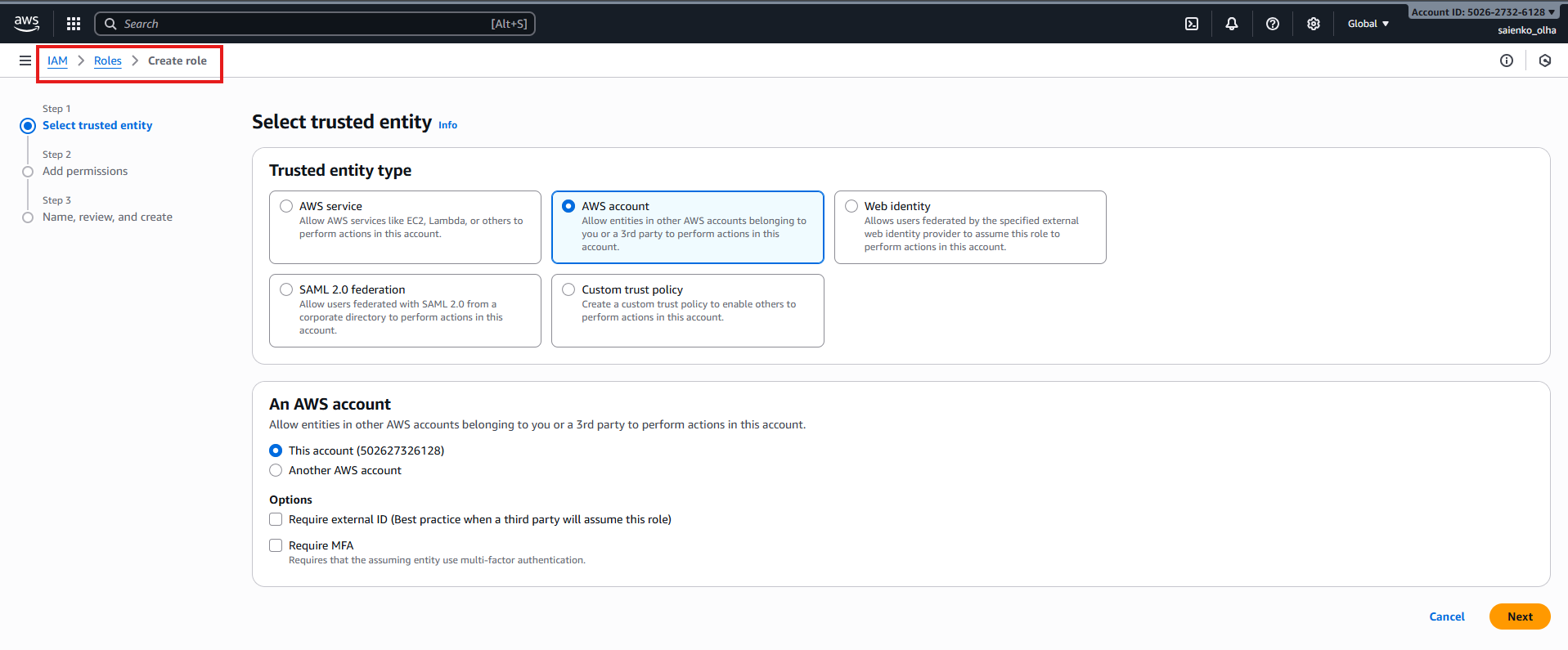
Перевіримо. Зайдемо під створеним користувачем:

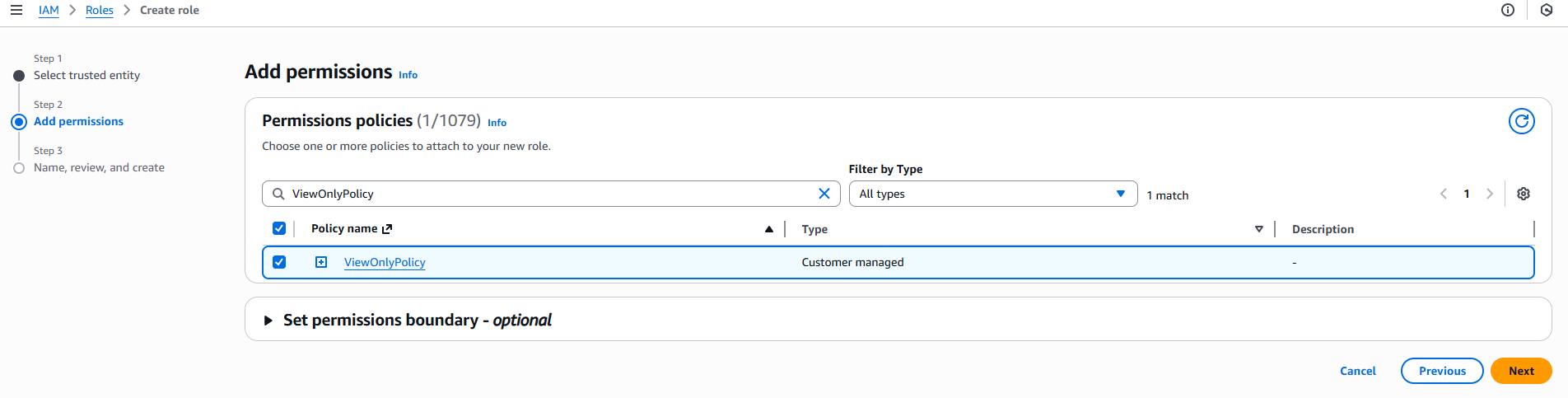


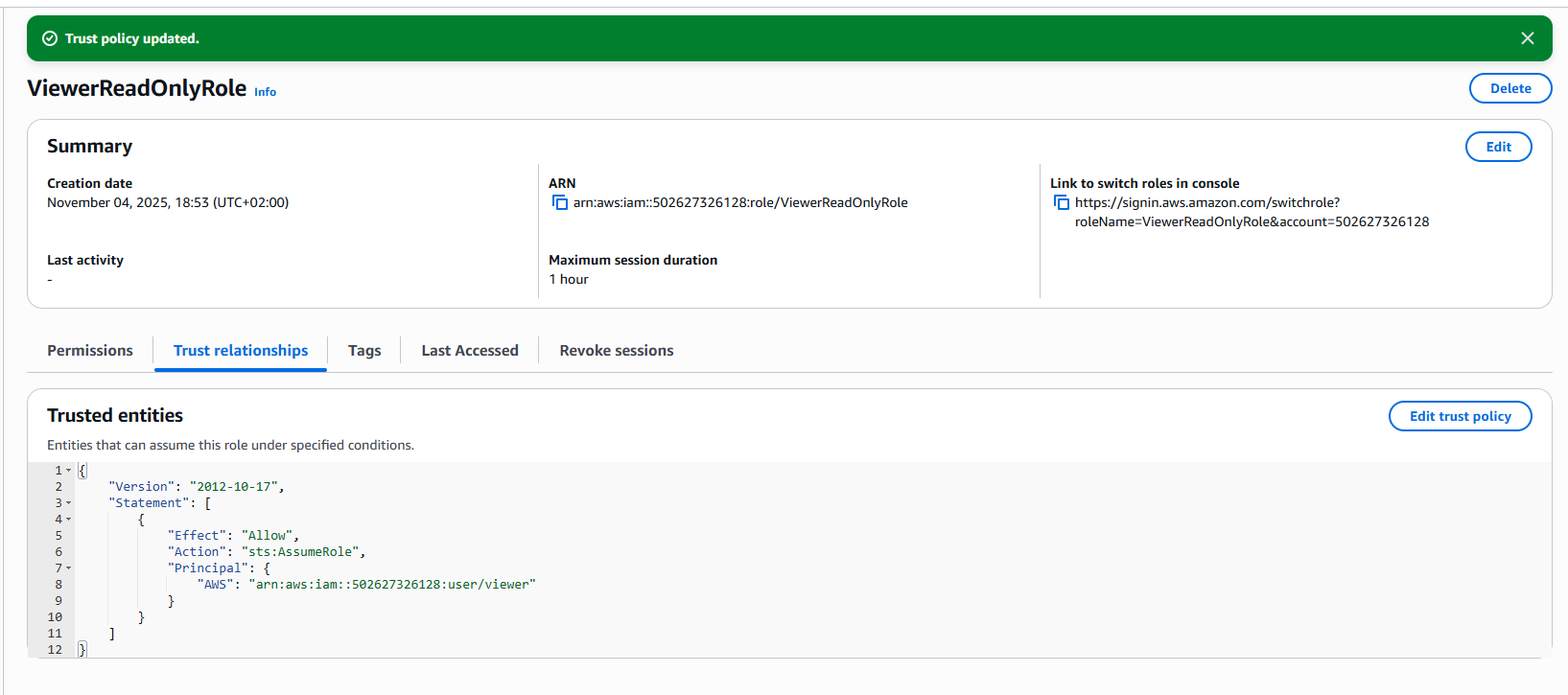


1. Cтворити IAM Role, до якої прикріплена політика лише перегляду і дозволити користувачу viewer “брати” цю роль.

Створюємо роль:





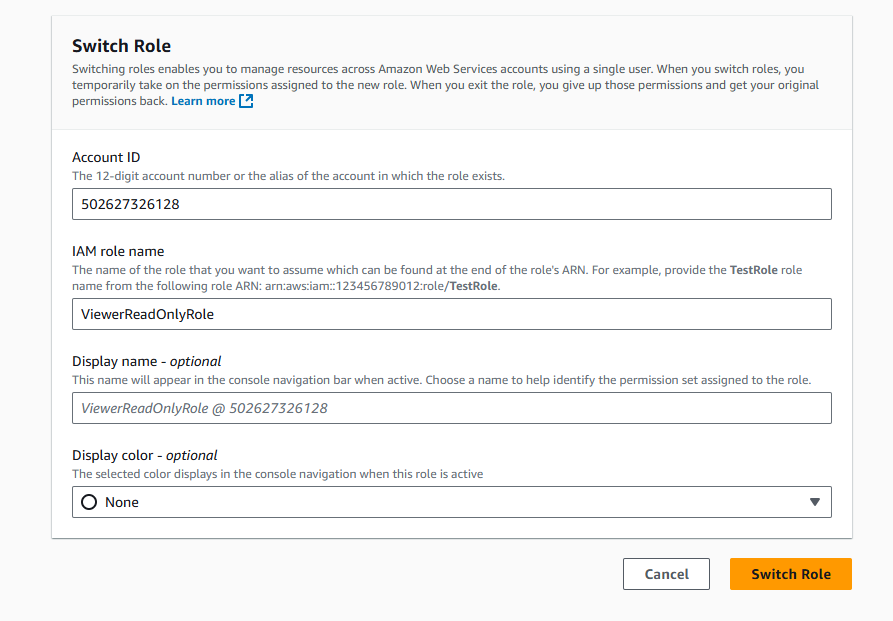


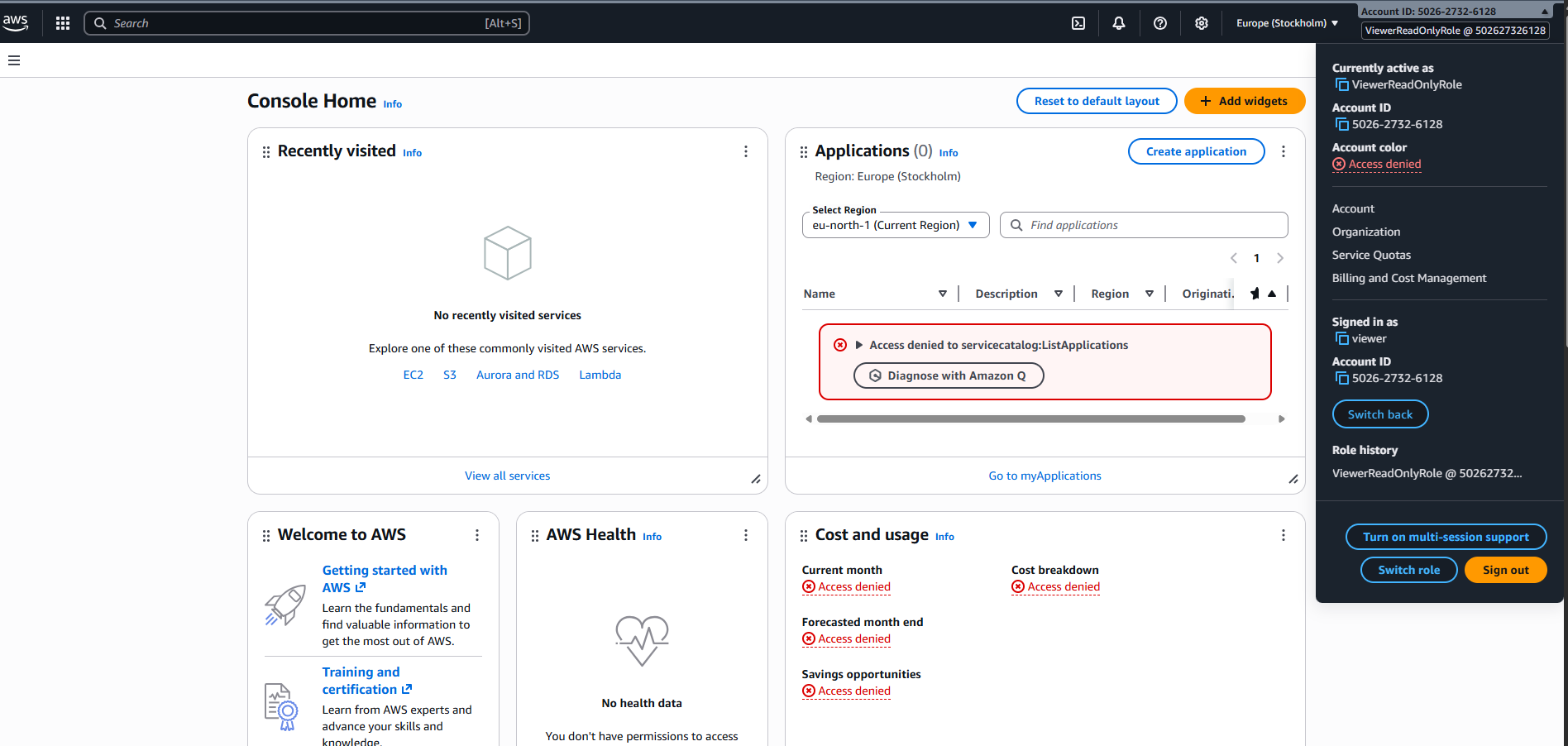
Заходимо в акаунт, де потрібно замінити Роль.



Було виконано перехід на роль ViewerReadOnlyRole через меню Switch Role.

У формі введено Account ID 502627326128 та Role name ViewerReadOnlyRole.





1. Створення CloudFormation-шаблону для опису політики та ролі (ViewOnlyPolicy і ViewerReadOnlyRole)

Створюємо файл iam-viewonly-role.yml з таким вмістом:

AWSTemplateFormatVersion: '2010-09-09'

Parameters:

AccountId:

Type: String

Description: AWS Account ID where the user exists

Default: "502627326128"

UserName:

Type: String

Description: IAM username that will assume the role

Default: "viewer"

Resources:

ViewOnlyPolicy:

Type: AWS::IAM::ManagedPolicy

Properties:

ManagedPolicyName: !Sub "${AWS::StackName}-ViewOnlyPolicy"

PolicyDocument:

Version: "2012-10-17"

Statement:

- Sid: ViewOnlyStatement

Effect: Allow

Action:

- ec2:Describe\*

- s3:Get\*

- s3:List\*

- rds:Describe\*

- cloudwatch:Get\*

- cloudwatch:List\*

- iam:Get\*

- iam:List\*

Resource: "\*"

ViewerReadOnlyRole:

Type: AWS::IAM::Role

Properties:

RoleName: ViewerReadOnlyRole

AssumeRolePolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: Allow

Action: sts:AssumeRole

Principal:

AWS: !Sub "arn:aws:iam::${AccountId}:user/${UserName}"

ManagedPolicyArns:

- !Ref ViewOnlyPolicy

Outputs:

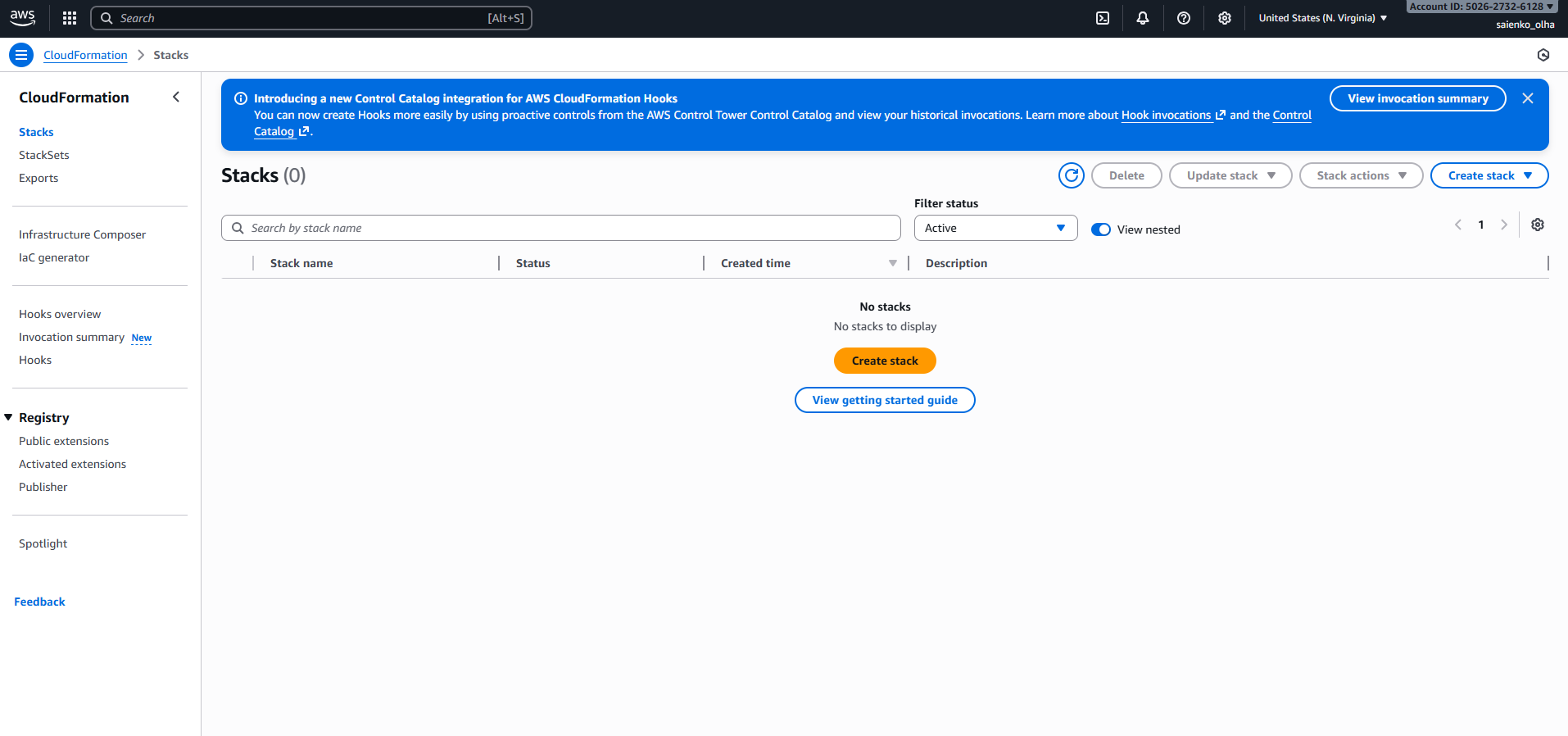
PolicyArn:

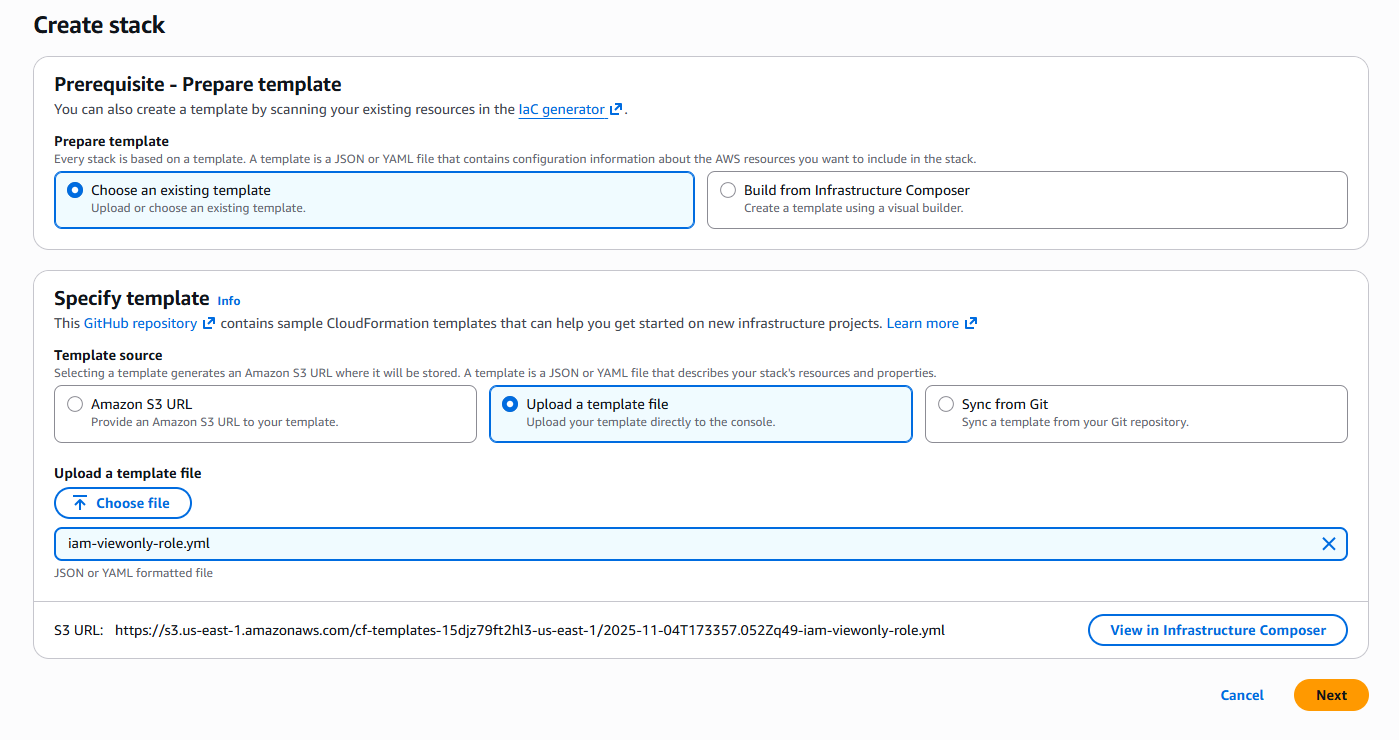
Description: ARN of the created managed policy

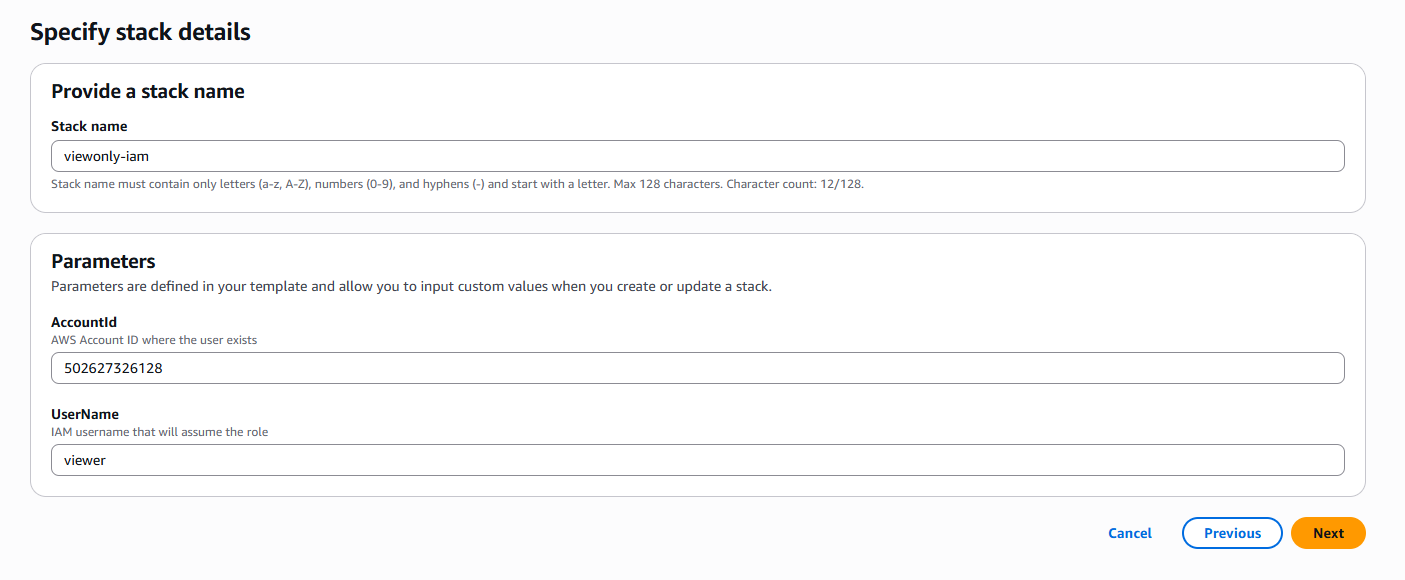
Value: !Ref ViewOnlyPolicy

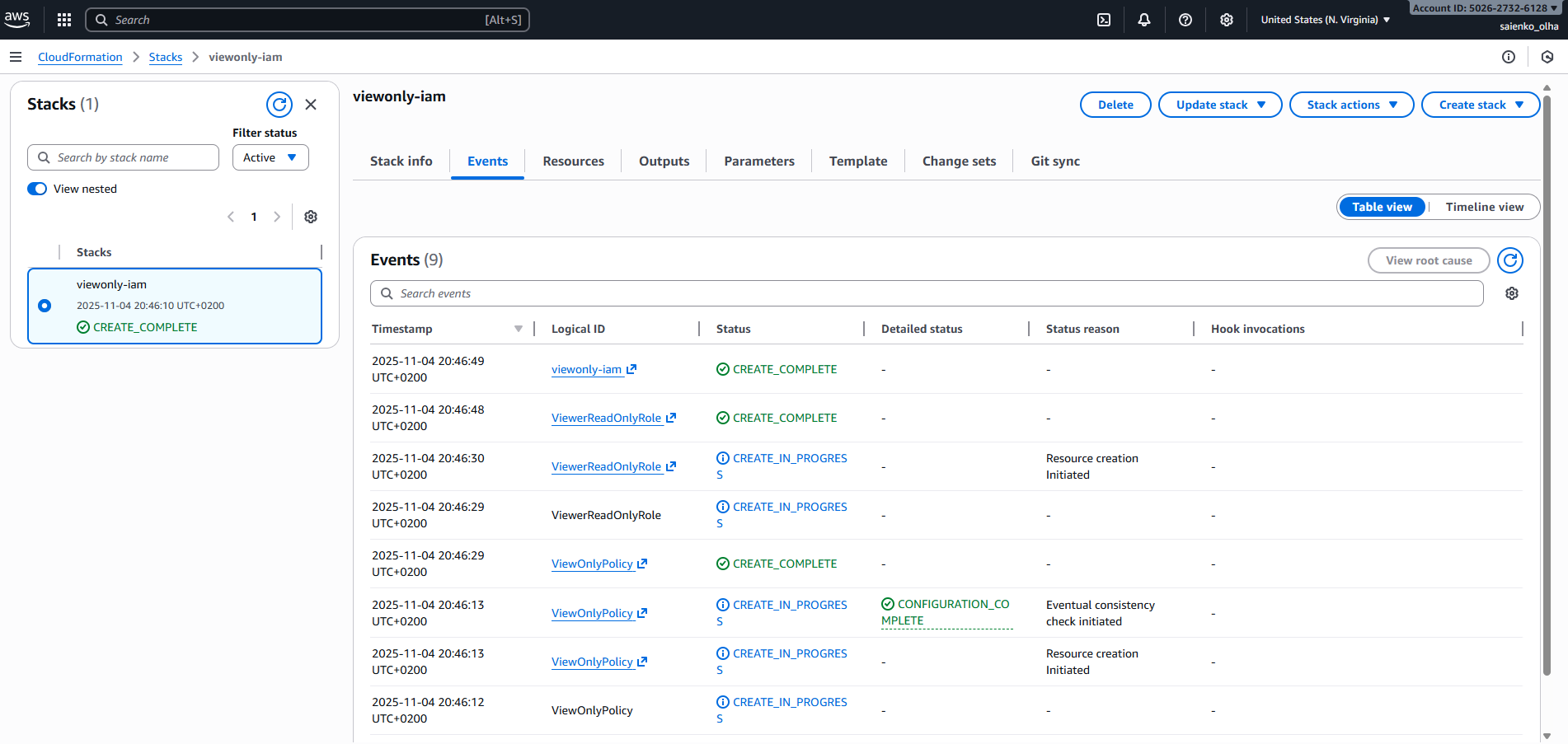
RoleArn:

Description: ARN of the assumable role

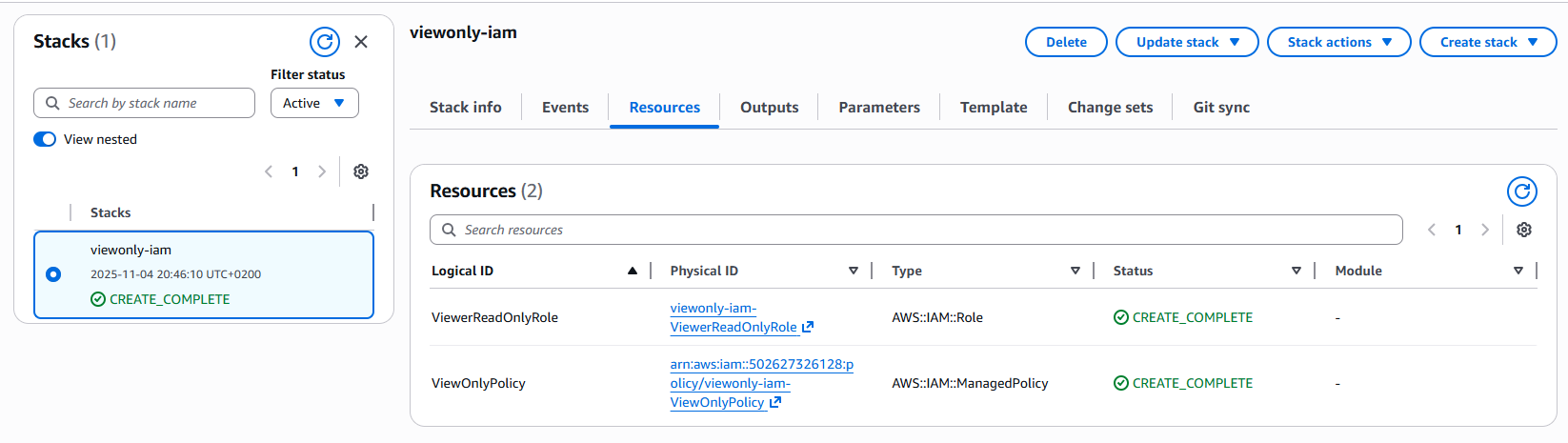
Value: !GetAtt ViewerReadOnlyRole.Arn

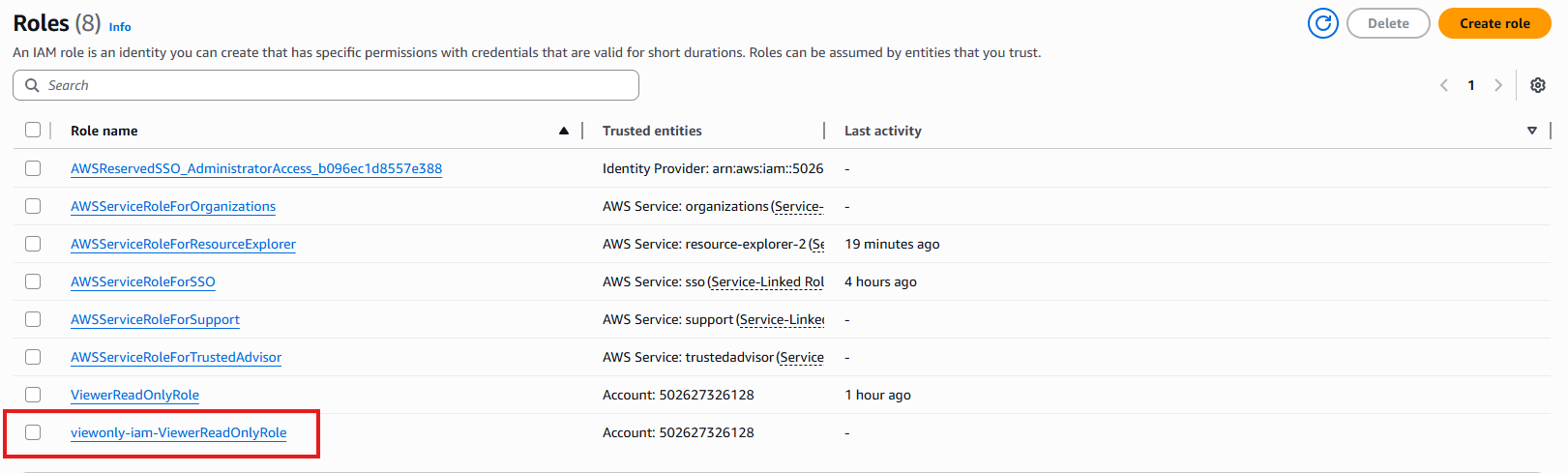






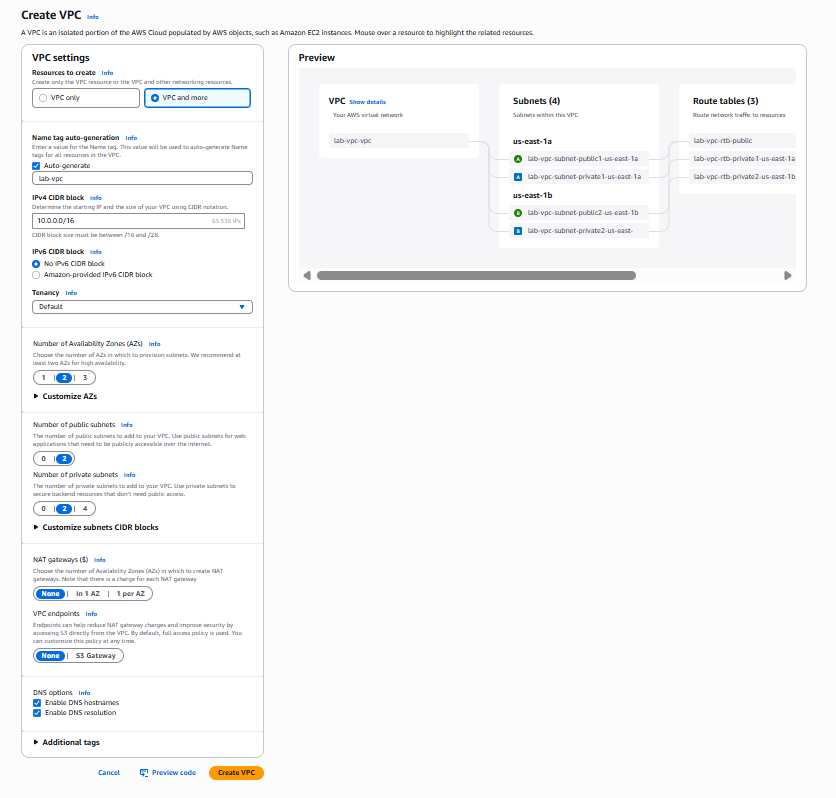
Перевіримо створені ресурси:

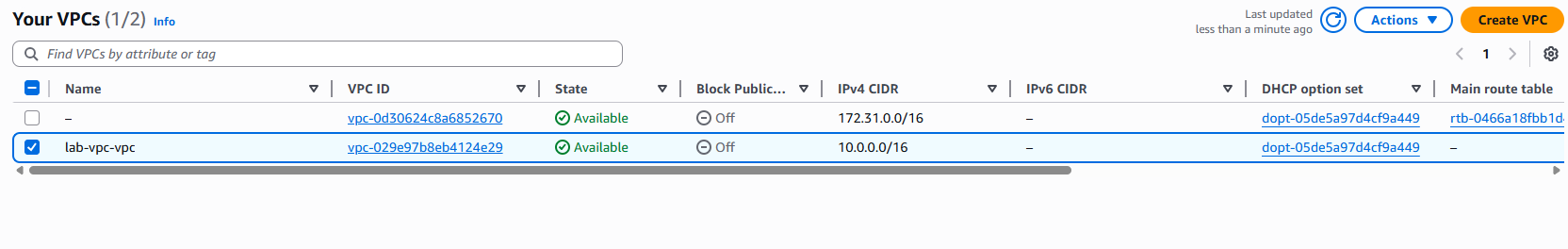


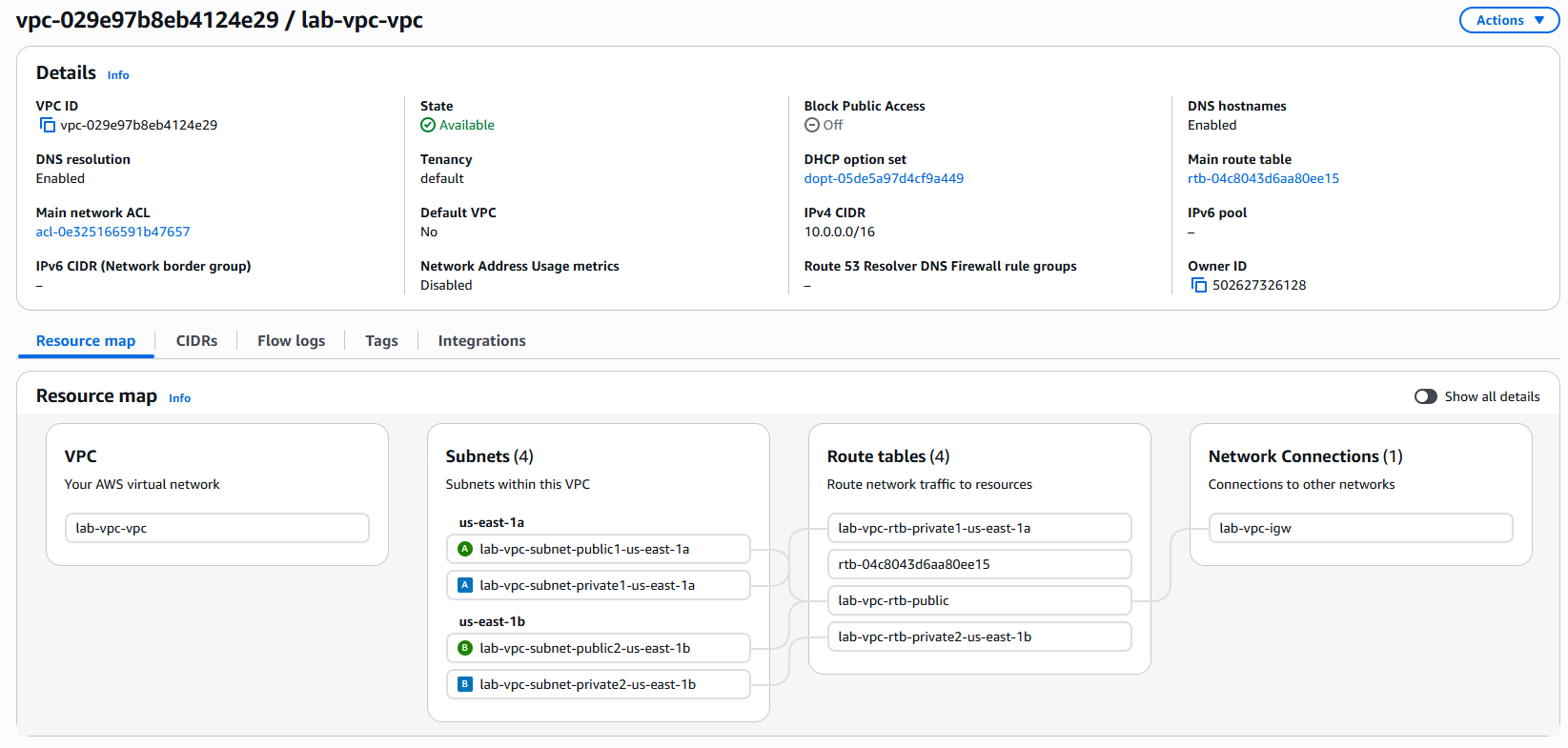


1. Створення мережі VPC з публічною та приватною підмережами

Створюємо нову VPC:







Було створено власну віртуальну мережу lab-vpc (10.0.0.0/16) з двома підмережами:

- Public Subnet (10.0.1.0/24) – має маршрут через Internet Gateway (IGW) для доступу до Інтернету.

- Private Subnet (10.0.2.0/24) – використовується для внутрішніх ресурсів, без зовнішнього доступу.

1. Створення CloudFormation-шаблону для мережі (VPC, public/private subnets, IGW, Route Tables)

Створюємо новий файл vpc-lab.yml з таким вмістом:

AWSTemplateFormatVersion: "2010-09-09"

Description: >

Creates a simple VPC with one public and one private subnet.

Public subnet has an Internet Gateway and route table.

Private subnet has internal-only access.

Parameters:

VpcCidr:

Type: String

Default: 10.0.0.0/16

PublicSubnetCidr:

Type: String

Default: 10.0.1.0/24

PrivateSubnetCidr:

Type: String

Default: 10.0.2.0/24

Resources:

# VPC

LabVPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VpcCidr

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: lab-vpc

# Internet Gateway

InternetGateway:

Type: AWS::EC2::InternetGateway

Properties:

Tags:

- Key: Name

Value: lab-igw

AttachGateway:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

VpcId: !Ref LabVPC

InternetGatewayId: !Ref InternetGateway

# Public Subnet

PublicSubnet:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref LabVPC

CidrBlock: !Ref PublicSubnetCidr

MapPublicIpOnLaunch: true

AvailabilityZone: !Select [0, !GetAZs ""]

Tags:

- Key: Name

Value: public-subnet

# Private Subnet

PrivateSubnet:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref LabVPC

CidrBlock: !Ref PrivateSubnetCidr

MapPublicIpOnLaunch: false

AvailabilityZone: !Select [0, !GetAZs ""]

Tags:

- Key: Name

Value: private-subnet

# Route Table for Public Subnet

PublicRouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref LabVPC

Tags:

- Key: Name

Value: public-rt

PublicRoute:

Type: AWS::EC2::Route

DependsOn: AttachGateway

Properties:

RouteTableId: !Ref PublicRouteTable

DestinationCidrBlock: 0.0.0.0/0

GatewayId: !Ref InternetGateway

PublicSubnetRouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PublicSubnet

RouteTableId: !Ref PublicRouteTable

# Private Route Table

PrivateRouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref LabVPC

Tags:

- Key: Name

Value: private-rt

PrivateSubnetRouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PrivateSubnet

RouteTableId: !Ref PrivateRouteTable

Outputs:

VpcId:

Description: VPC ID

Value: !Ref LabVPC

PublicSubnetId:

Description: Public subnet ID

Value: !Ref PublicSubnet

PrivateSubnetId:

Description: Private subnet ID

Value: !Ref PrivateSubnet

InternetGatewayId:

Description: IGW ID

Value: !Ref InternetGateway

Створюємо новий stack:

