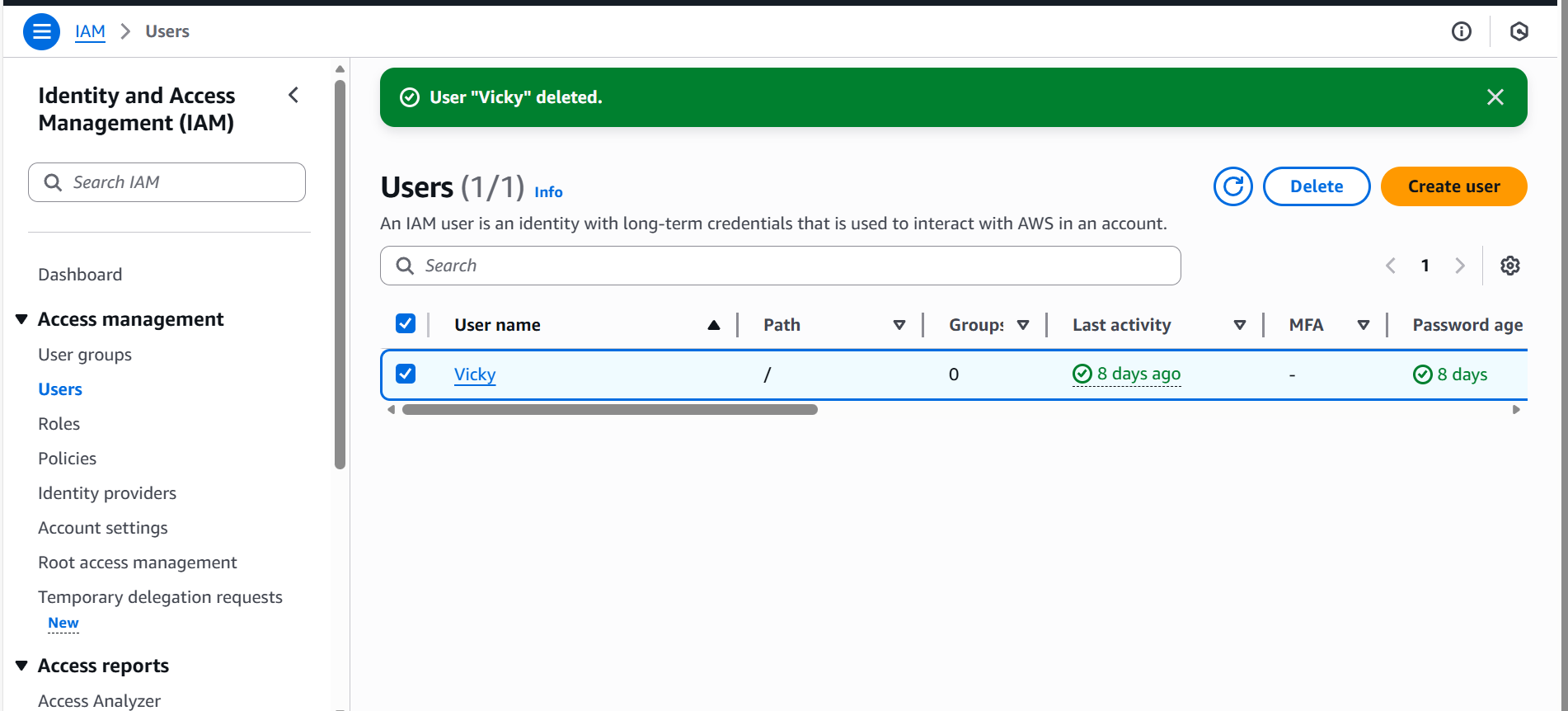
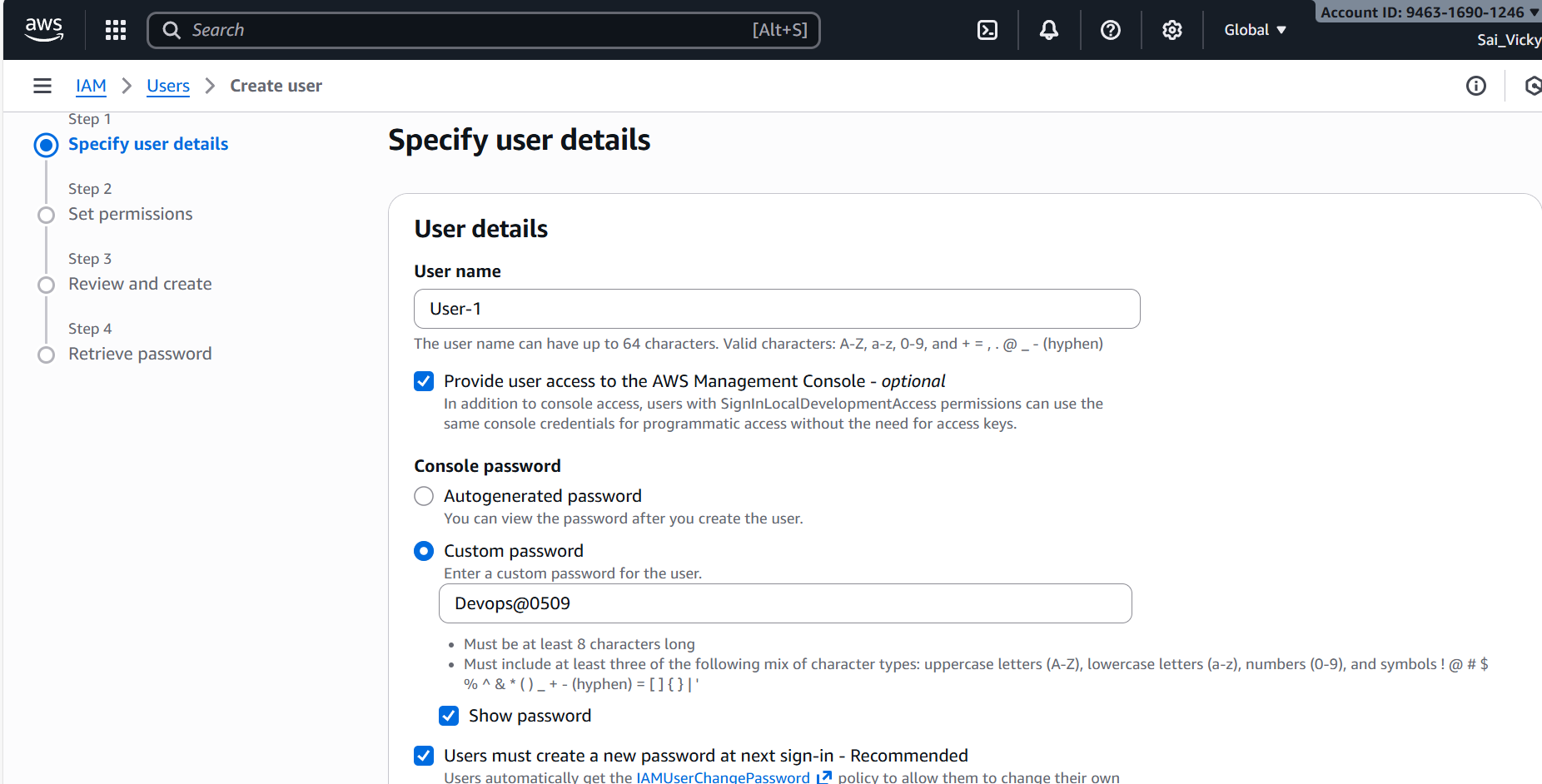
**AWS- IAM**

1. **Create one IAM user and assign EC2 and S3 full access roles.**

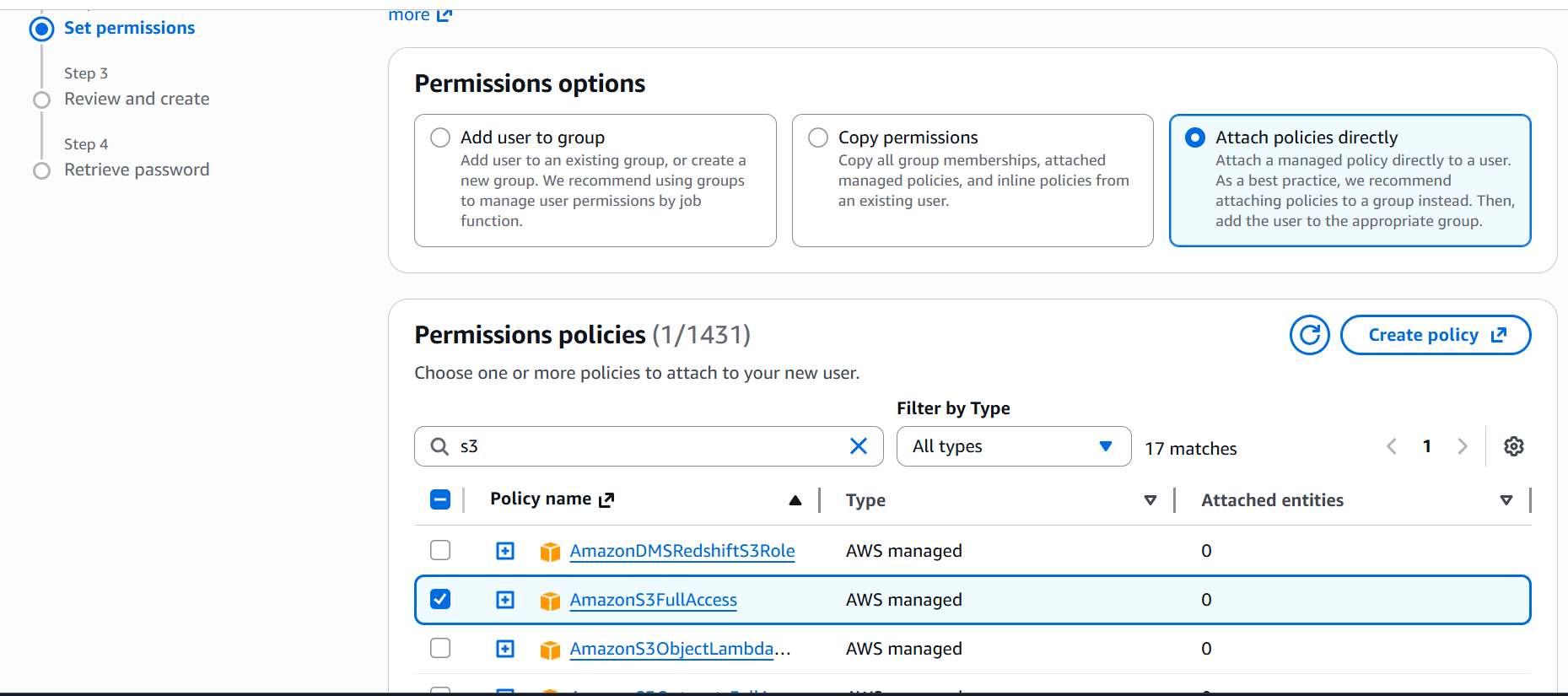
* **Navigate to IAM** in AWS.
* I had a previous IAM user created, so before starting this task, **deleted** as not required.
* Scroll down at the left and **select user**s to **create a user**.



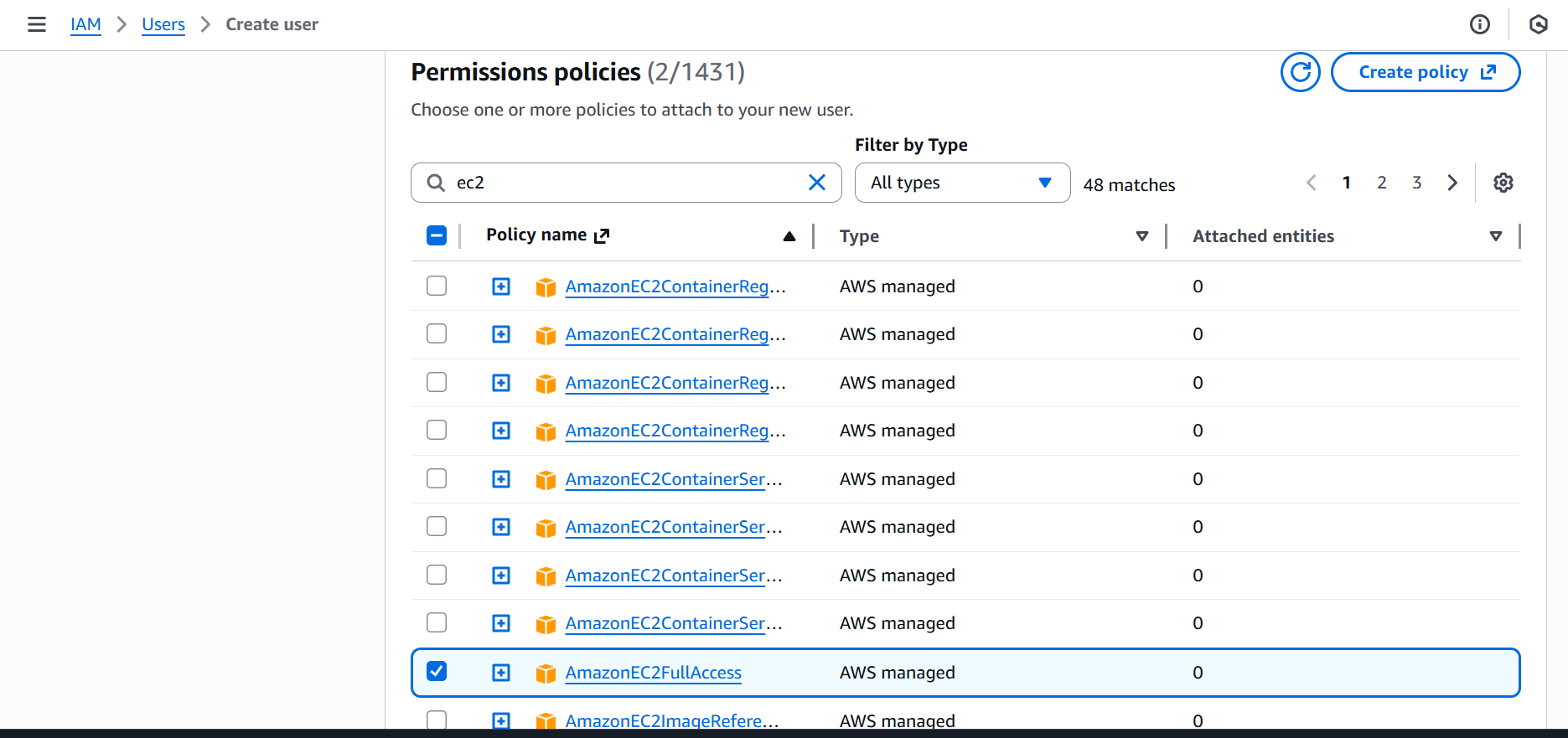
* Creating a user, **specify user details**.
* Username entered is **user-1**
* **Custom password selected** and proceeded to next step.



* **Setting up policies**, where we are attaching policies directly.
* In the permission policies, we are selecting 2 policies here, 1 is **S3 and 2nd is ec2.**
* Below we are providing full **S3 access**.



* Below we are providing **full ec2 access**. Both policies are done in 1 step.



* Now, we can see **both policies are ready for the creation.**

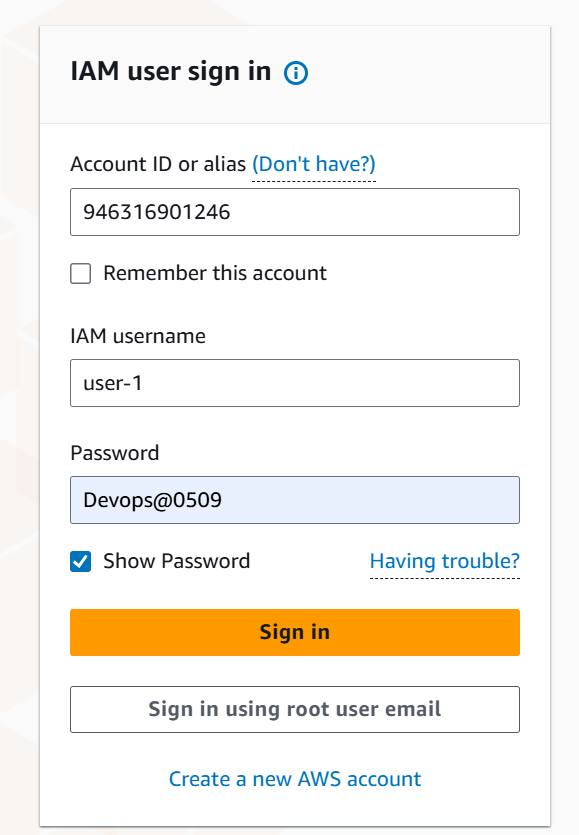


* User is created.

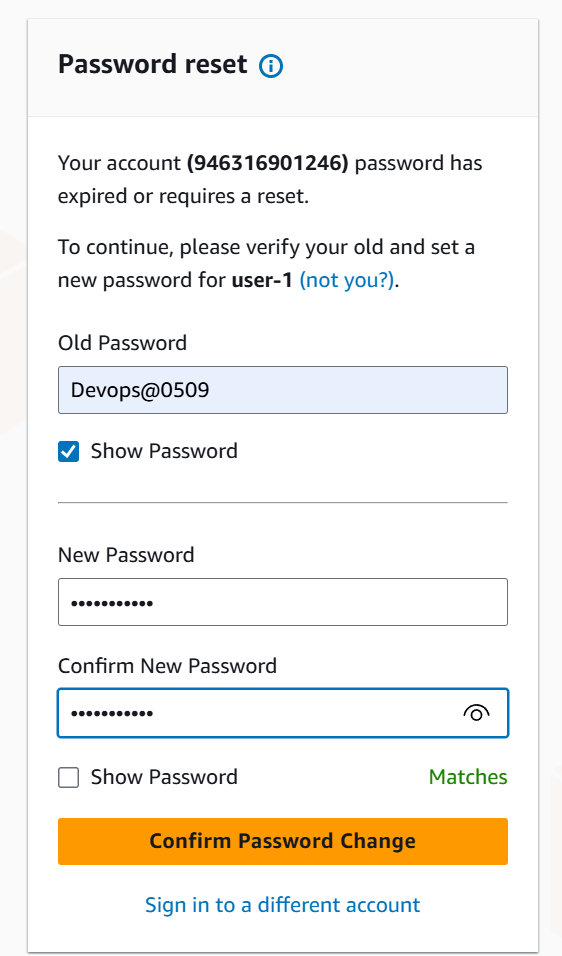
A screenshot of a login page

AI-generated content may be incorrect.

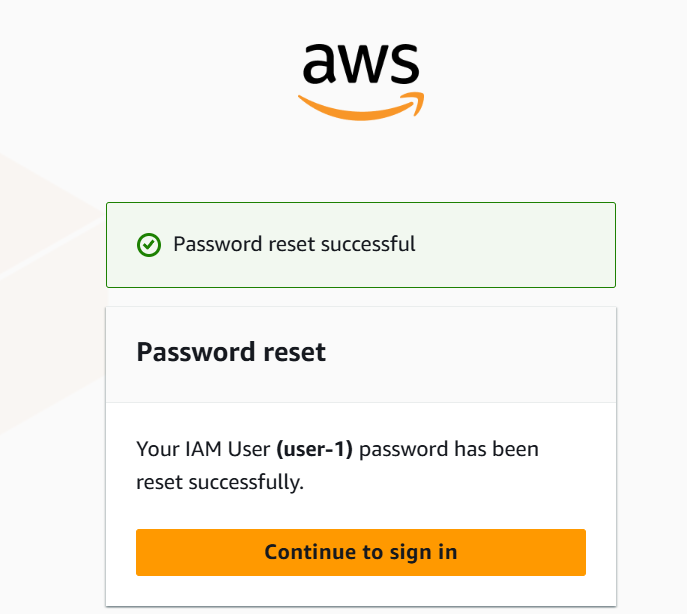
* Now, its time to paste the above URL in browser and shall proceed for next steps.
* Signing in with **IAM user sign in to verify** if the login is working.



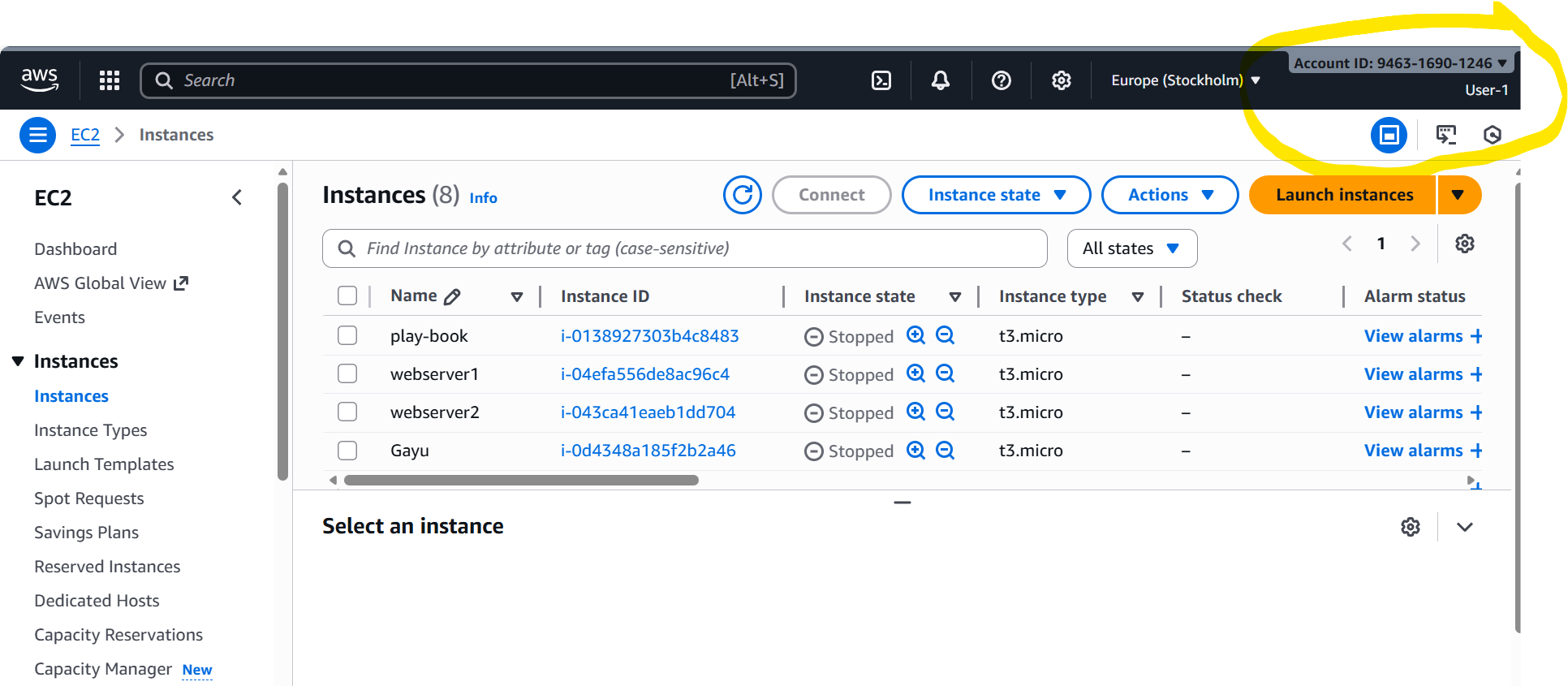
* While user logs in to their account for the first time, **its mandatory to reset the password.**



* Password **reset** successful.



* As we have provided **full access for ec2 and S3** to the **user-1**, we shall verify now.
* Lets verify ec2 first as below.
* Success- we can see in the below screenshot, **user-1 has full access my ec2 instances**.



* Now its time to **verify aws s3**.
* Success, **user-1 has full access to s3 buckets** as well. Task completed.

A screenshot of a computer

AI-generated content may be incorrect.

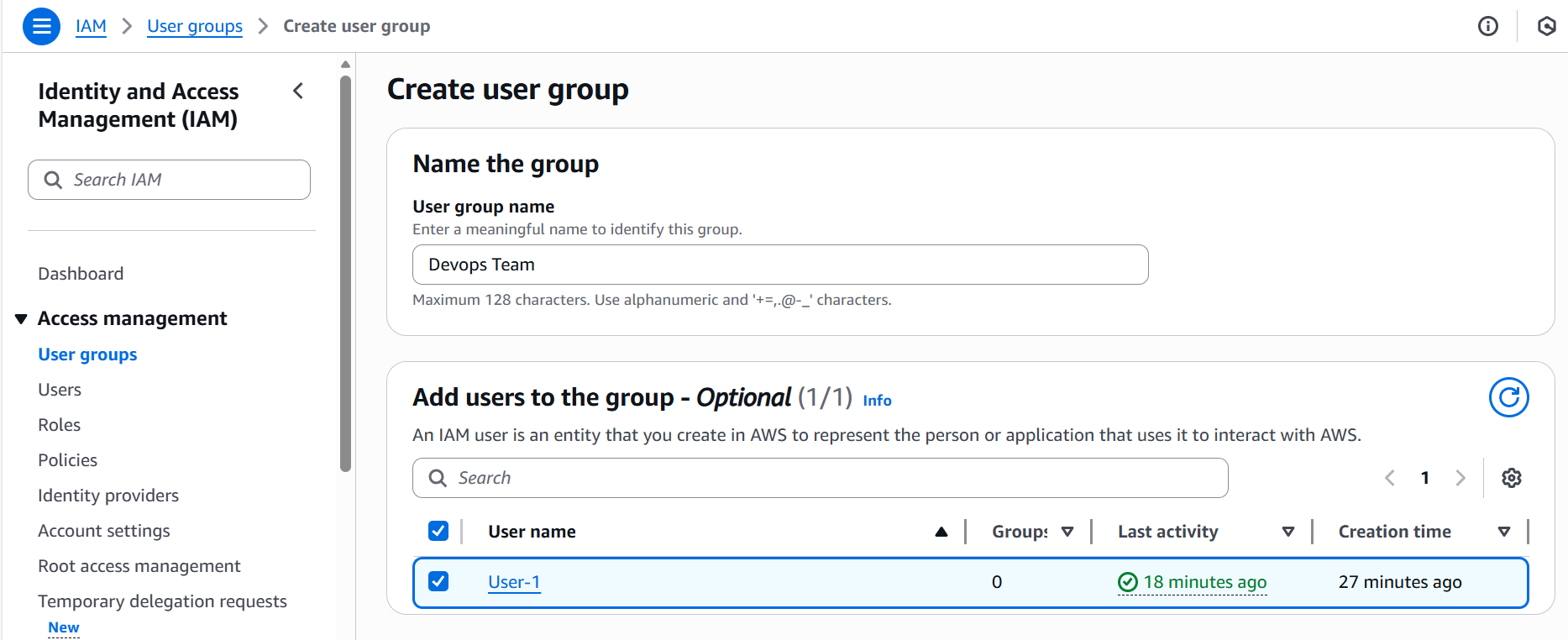
1. **Create one group in IAM and assign read access for EC2.**

* **Navigate to IAM** and scroll to the left and search for **User groups.**
* Create a group

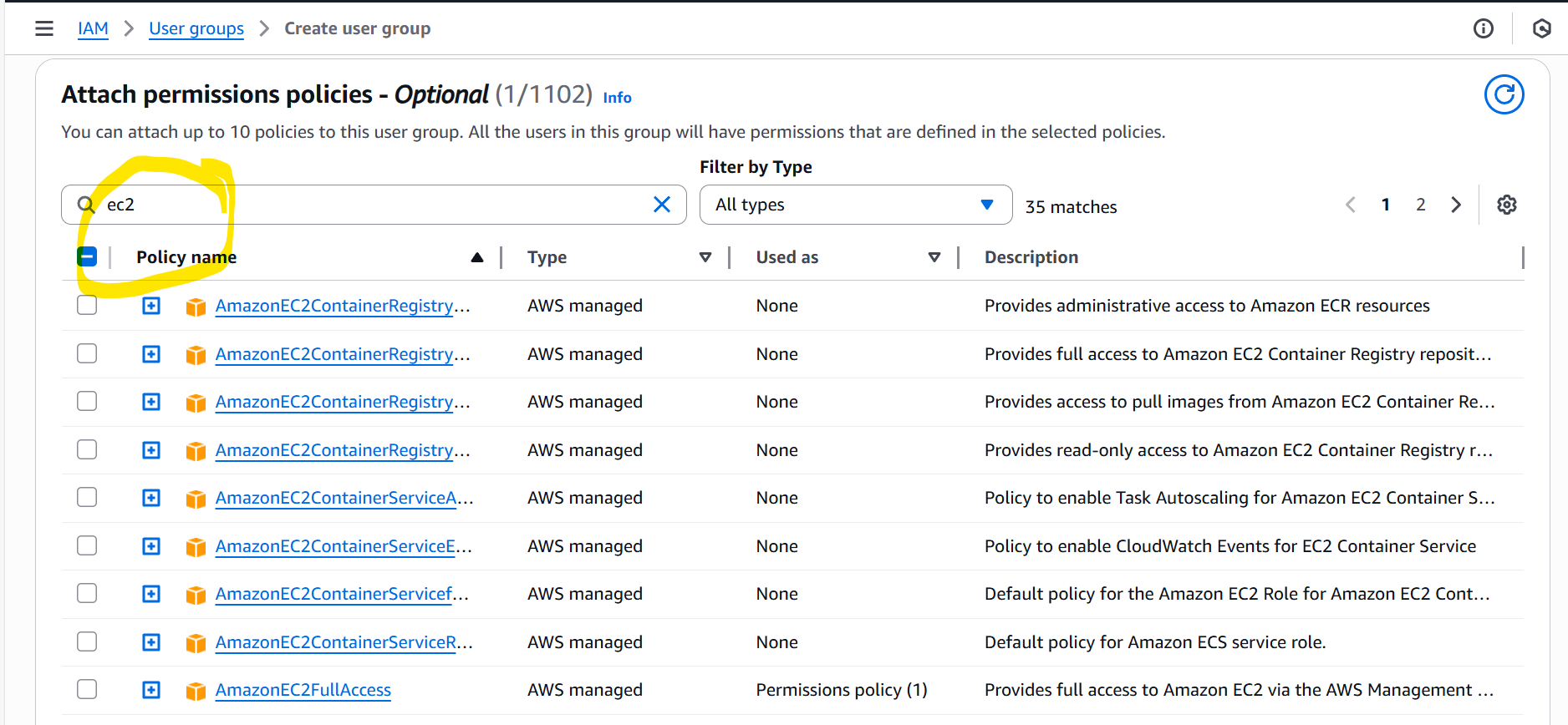
A screenshot of a computer

AI-generated content may be incorrect.

* Created user group named as- **Devops Team.**



* Now, we need to provide permission for the user in the group- **ec2 read only access.**
* **Ec2 read only** access selected.

**A screenshot of a computer

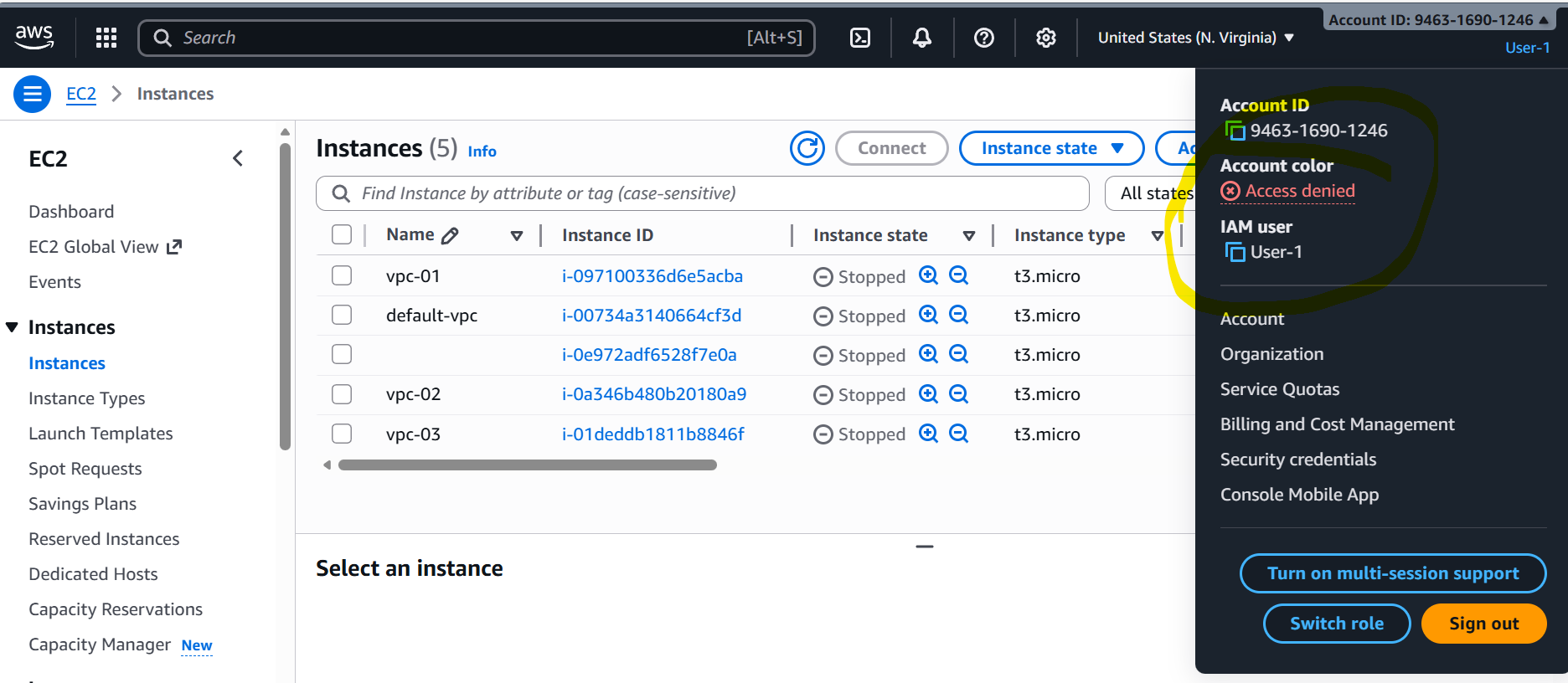
AI-generated content may be incorrect.**

* **User group created successfully.**

**A screenshot of a computer

AI-generated content may be incorrect.**

* As we can see in the below screenshot**, ec2 read only** access is providedas access is denied to update anything in the instance level. **Task completed.**

****

1. **Create a new user named "Devops" and add to the group created in task 2.**

* Creating a user to our team**, named- Devops** and **password** created as well.

**A screenshot of a computer

AI-generated content may be incorrect.**

* **For this task,** we are not **attaching any policies,** so we are selecting**- Add user to group.**

**A screenshot of a computer

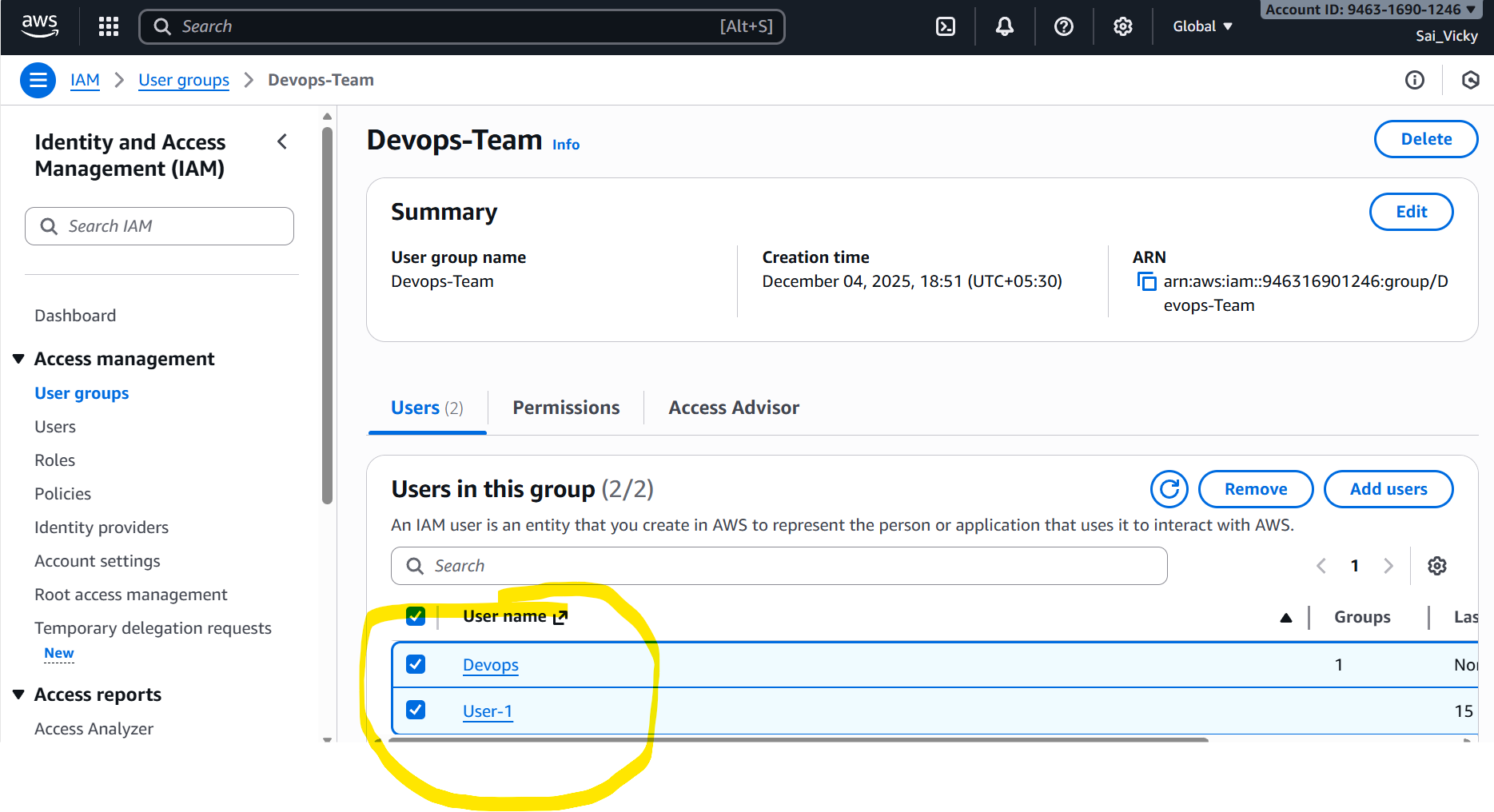
AI-generated content may be incorrect.**

* **User created successfully.**
* **Now its time to add the user to our group.**

**A screenshot of a login screen

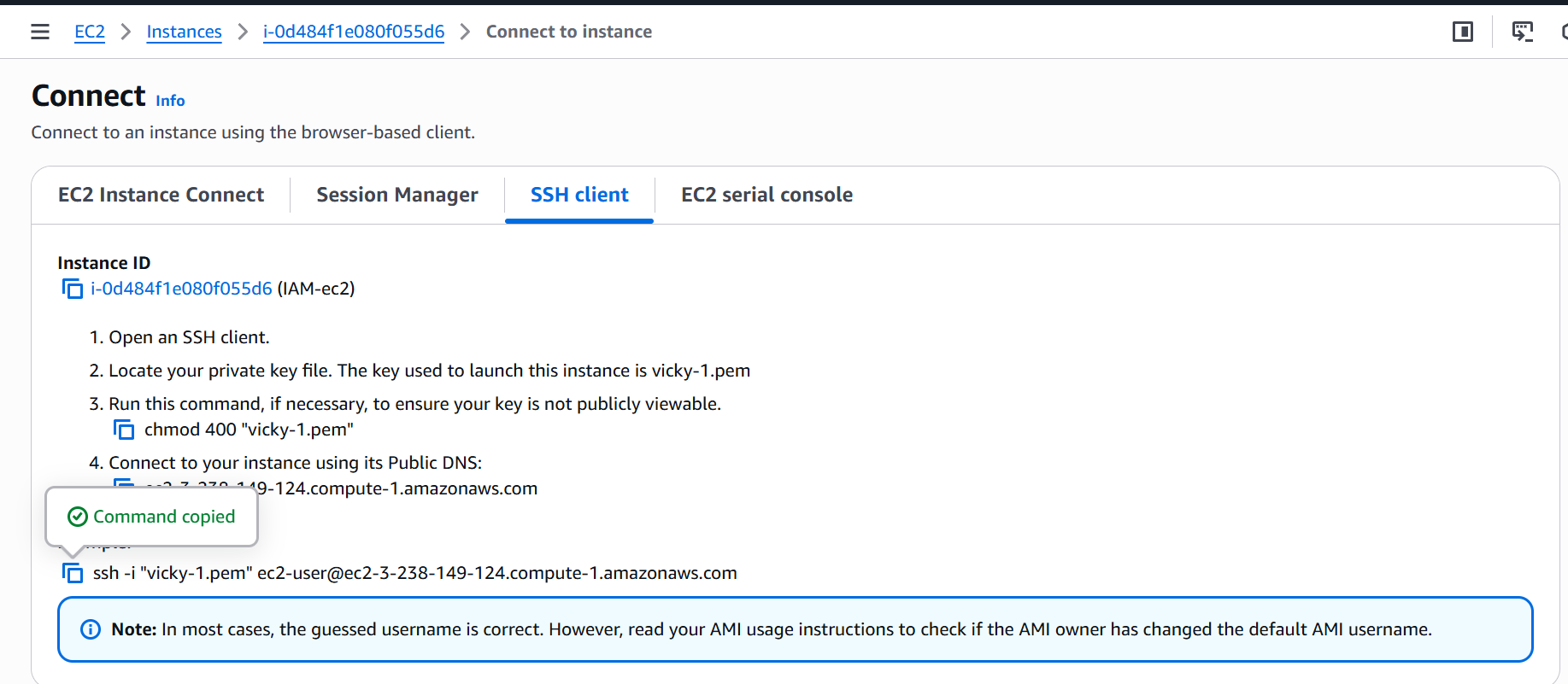
AI-generated content may be incorrect.**

* **User-Devops** has been successfully added to the users group- **Devops-Team. Verified below.**

****

1. **Write a bash script to create an IAM user with VPC full access.**

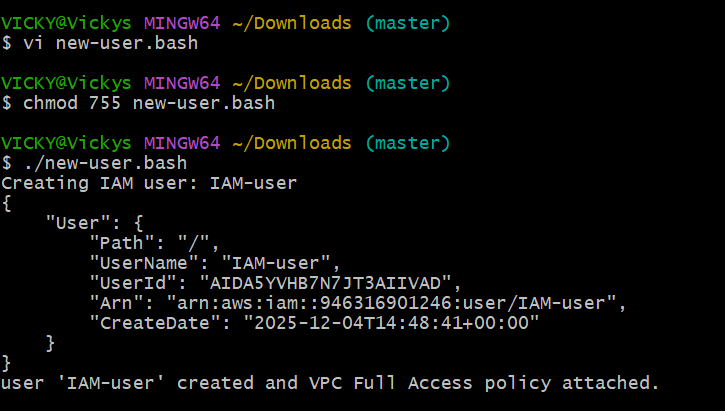
* Launching a **instance** from aws to perform this task.



* Once connected to instance in the terminal.
* We need to **get the configuration details** updated in the terminal from our aws account. We can get the config details from our **profile- security credentials🡪Access keys**- create one and update in local terminal.
* To configure- **aws configure**
* **Access ID** and **Access key** updated.
* My default region is**- us-east-1**
* Output format is- **json**
* To get the complete list of users added- **aws iam list-users**
* In the below screenshot, we can see the **users** available.



* Now, its time to create the iam user by **writing a script** in our local terminal.
* Script added using **vi new-user.bash**
* Provided permissions using command- **chmod 755 new-user.bash**
* Executed the script- **./new-user.bash**



* Below is the script added to get the **iam user creation in aws.**
* Now lets verify if the **iam user** is added in **aws.**

A computer screen with text on it

AI-generated content may be incorrect.

* **Iam user is added successfully.**

A screenshot of a computer

AI-generated content may be incorrect.

* Final step is to verify in terminal as well using command- **aws iam list-users**
* **User added successfully.**



1. **Create an IAM policy to allow EC2 access for a specific user in specific regions only.**

* Navigate to **Iam 🡪** scroll down to left and select **policies.**
* Under the policies **🡪 create a policy.**
* Update the script in **json format.**
* Copied the script fromgoogle and posted in the **json format.**
* Click **Next**

**A screenshot of a computer

AI-generated content may be incorrect.A computer screen shot of a computer code

AI-generated content may be incorrect.**

* **Create a policy name** which is easy to identify.
* Add a **description** as well to retrieve **the Policy** in future.

**A screenshot of a computer

AI-generated content may be incorrect.**

* Finally, we created a policy and proceeding to next step.

**A screenshot of a computer

AI-generated content may be incorrect.**

* Now we need to **attach this policy to a specific user and a specific region.**
* Navigate to **user** and attach the **policy.**
* Now, we selected **“Iam-user”** and attaching the policy as per below screenshot.

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

* Permissions provided
* The policy is attached to the **user now** and it will be for a **specific region** as per the script **updated in json format.**

1. **We have two accounts: Account A and Account B. Account A user should access an S3 bucket in Account B.**

* Used the below **script** to add **Account B** details.

{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowAccountAUserAccess",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::ACCOUNT\_A\_ID:user/SomeUser"  
      },  
      "Action": [  
        "s3:GetObject",  
        "s3:PutObject",  
        "s3:ListBucket"  
      ],  
      "Resource": [  
        "arn:aws:s3:::bucket-in-account-b",  
        "arn:aws:s3:::bucket-in-account-b/\*"  
      ]  
    }  
  ]  
}

* We created a bucket name- **vicky-1995 and Edit bucket policy** to update the script using Account B ID and User name.

A screenshot of a bucket policy

AI-generated content may be incorrect.A computer screen shot of a computer screen

AI-generated content may be incorrect.

* Updated the script in my bucket using json format.
* In the ARN- I used **Wasim Akram user id and user name.**
* In the resource, I updated my **bucket name- vicky-1995.**

A screenshot of a computer

AI-generated content may be incorrect.

* **Added a file ec2-task** since this bucket was empty.

A screenshot of a computer

AI-generated content may be incorrect.

* Account B- Wasim used his credentials and used command- aws s3 ls **s3://vicky-1995 (my bucket-name).**
* **Success**- Account B can view my buckets.

A screenshot of a computer

AI-generated content may be incorrect.