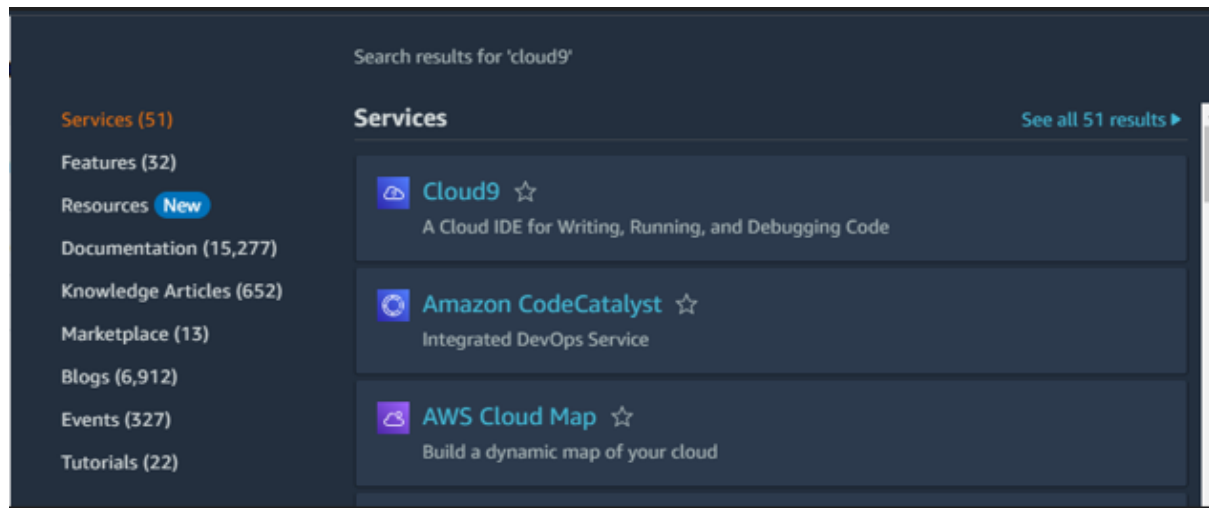
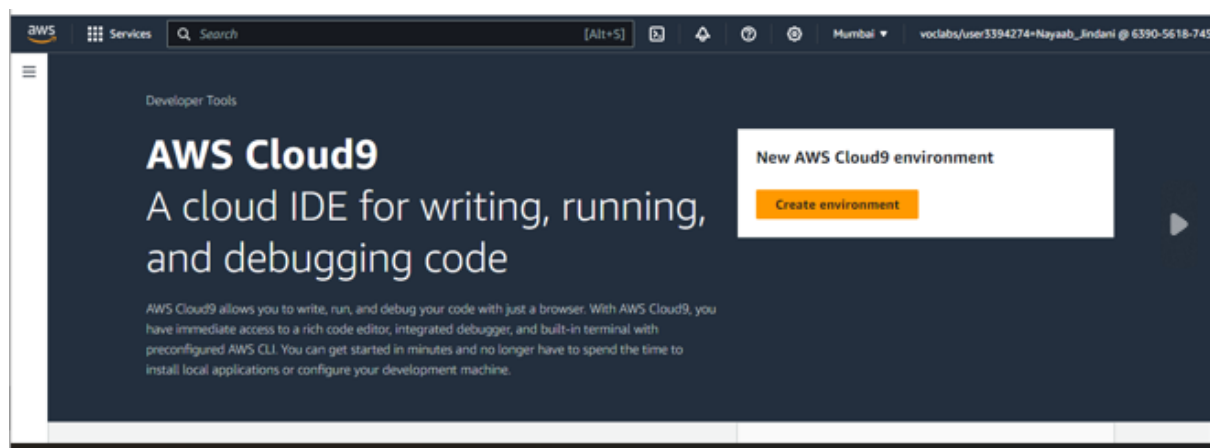


EXPERIMENT 1(B):To understand the benefits of the cloud infrastructure and setup AWS Cloud9 IDE ,Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

STEP 1:Go on AWS academy account and search Cloud9.



STEP 2:By clicking on the Cloud 9 an interface will be seen and on that click on create environment.



STEP 3: Enter the name of the environment and select the environmental type as New EC2 instance .

[AWS Cloud9](#) > [Environments](#) > Create environment

Create environment [Info](#)

Details

Name

WebAppIDE

Limit of 60 characters, alphanumeric, and unique per user.

Description - optional

Limit 200 characters.

Environment type [Info](#)

Determines what the Cloud9 IDE will run on.

☒ New EC2 instance
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

☐ Existing compute
You have an existing instance or server that you'd like to use.

STEP 4: Go on Connection and click on Secure Shell(SSH).

Connection

How your environment is accessed.

☐ AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).

☒ Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.

[VPC settings](#) [Info](#)

STEP 5: Go on Environments and click on create environments and the environment will successfully get created.

AWS Cloud9

My environments

Shared with me

All account environments

Documentation [Info](#)

[AWS Cloud9](#) > [Environments](#)

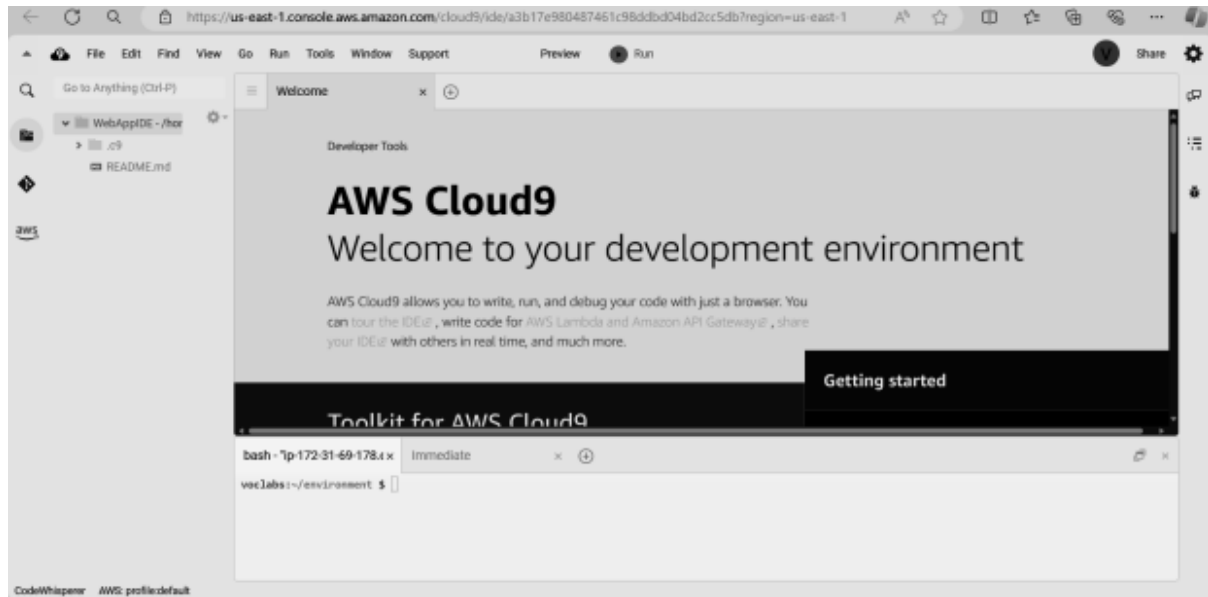
Environments (1) [Delete](#) [View details](#) [Open in Cloud9](#) [Create environment](#)

My environments

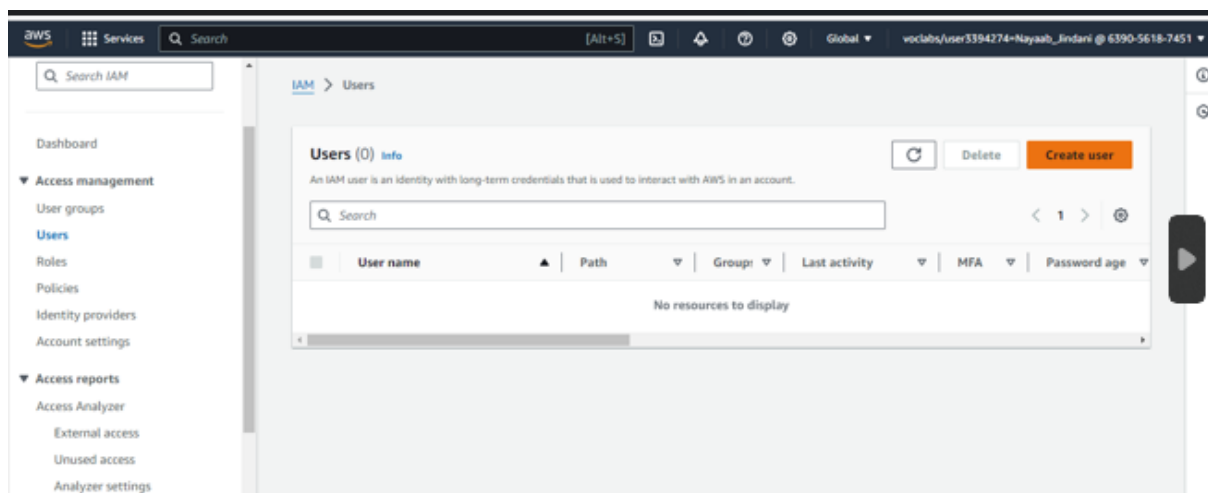
Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
WebAppIDE	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::663539227562:assumed-role/voclabs/user3385469-KULKARNI

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

STEP 6: After the environment gets created, an interface will be displayed.



STEP 7: Till that time, open IAM Identity and Access Management in order to add a user in another tab.



STEP 8: Add username and click on next button.

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name

user1

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

i If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

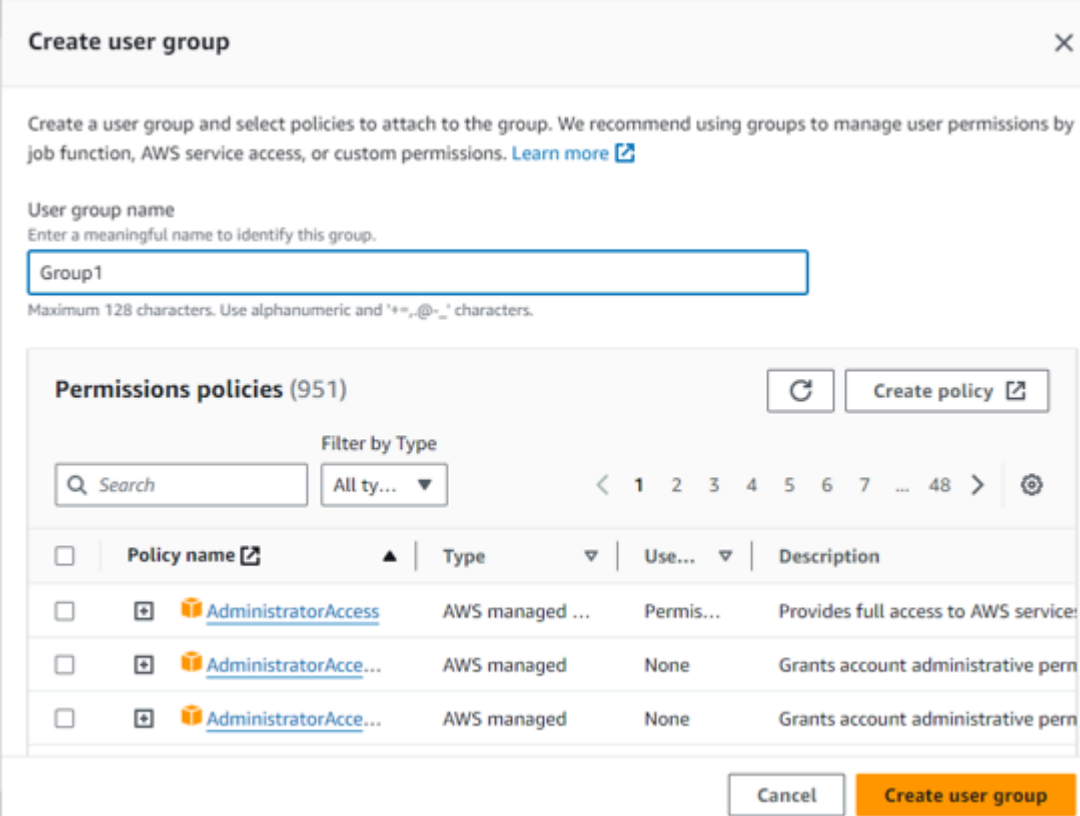
STEP 9: After creating user an interface will display and on it click on create group.

i Get started with groups

Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

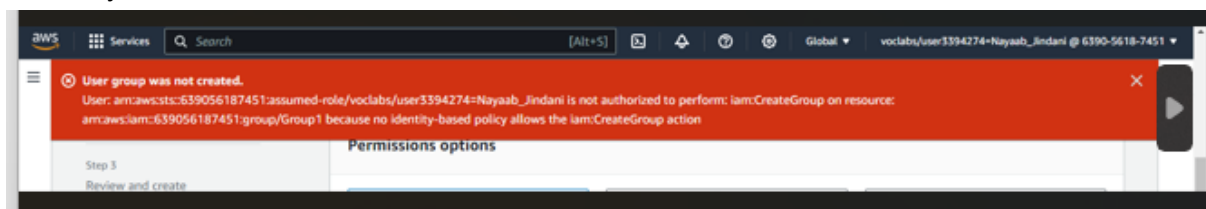
Create group

STEP 10: Add group name and click on create user group.

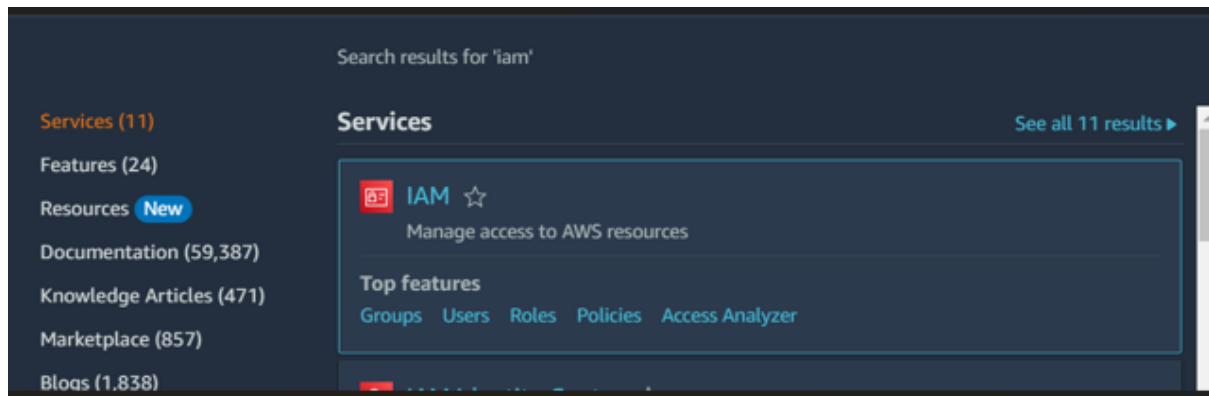


The screenshot shows the 'Create user group' console page. At the top, there's a title bar with 'Create user group' and a close button. Below it, a message says: 'Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)'. The 'User group name' section has a text input field containing 'Group1' and a note: 'Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.' Below this is a 'Permissions policies (951)' section with a search bar, a 'Filter by Type' dropdown set to 'All ty...', and a pagination control showing '1 2 3 4 5 6 7 ... 48'. A table lists policies with columns for checkboxes, policy names (e.g., 'AdministratorAccess'), types ('AWS managed'), use cases ('Permis...'), and descriptions ('Provides full access to AWS service...'). At the bottom right are 'Cancel' and 'Create user group' buttons.

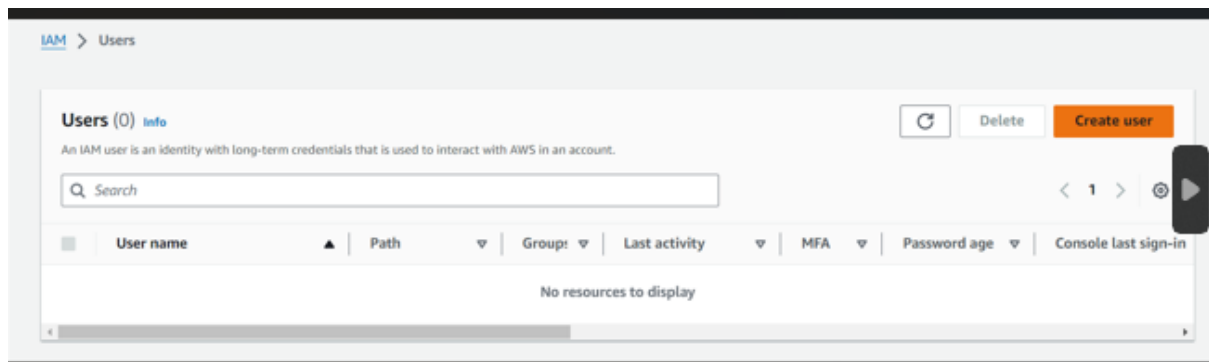
STEP 11: The user group will get created because it is not authorised to perform in the AWS academy account.



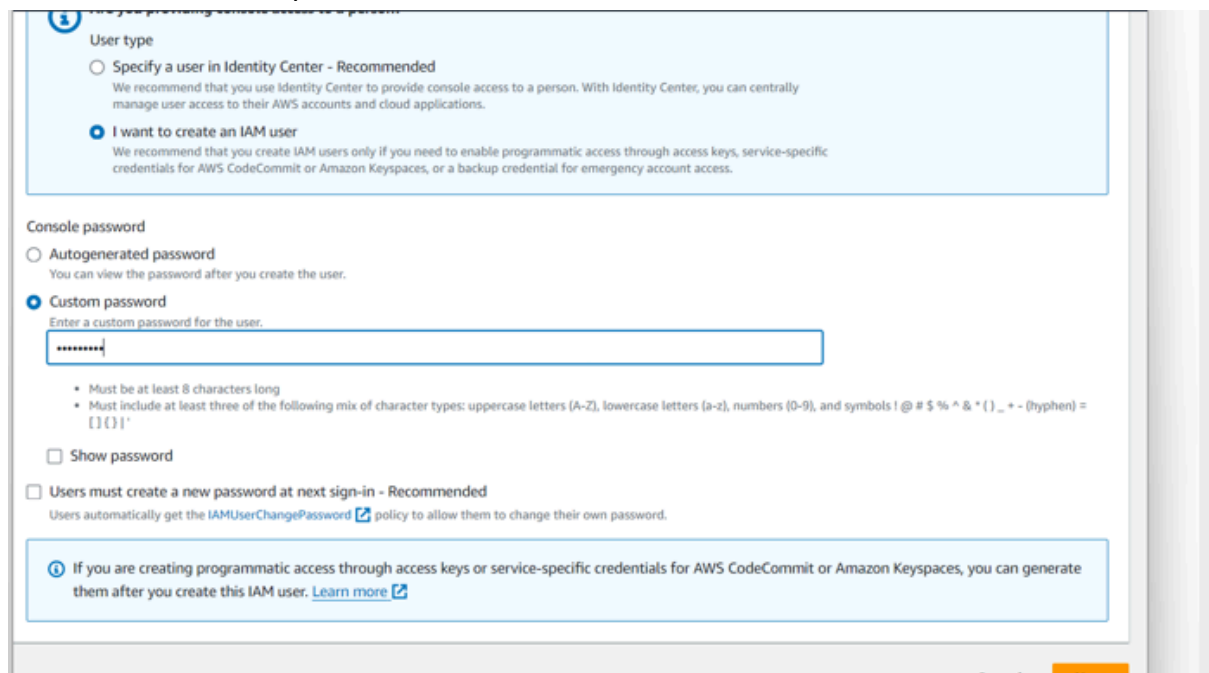
STEP 12: Since the user does not get created in AWS academy account we will try to create it on AWS account so search IAM and click on it .



STEP 13: Go on users and click on create user.



STEP 14: Add custom password and click on next.



STEP 15: Go on create user group add group name and click on create user group..

Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

User group name
Enter a meaningful name to identify this group.

Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.

Permissions policies (947)

Filter by Type: All ty... < 1 2 3 4 5 6 7 ... 48 >

<input type="checkbox"/>	Policy name	Type	Used as	Description
<input type="checkbox"/>	AdministratorAccess	AWS managed ...	None	Provides full access to AWS s
<input type="checkbox"/>	AdministratorAcce...	AWS managed	None	Grants account administrativ
<input type="checkbox"/>	AdministratorAcce...	AWS managed	None	Grants account administrativ
<input type="checkbox"/>	AlexaForBusinessD...	AWS managed	None	Provide device setup access
<input type="checkbox"/>	AlexaForBusinessF...	AWS managed	None	Grants full access to AlexaFo
<input type="checkbox"/>	AlexaForBusinessG...	AWS managed	None	Provide gateway execution a

STEP 16: Group gets created successfully.

Group1 user group created.

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Step 2: Set permissions

Permissions options

- ☒ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ☐ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- ☐ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1)

<input type="checkbox"/>	Group name	Users	Attached policies	Created
<input type="checkbox"/>	Group1	0	-	2024-08-08 (Now)

Set permissions boundary - optional

STEP 17: Then click on create user for creating the user.

Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Create user](#)

STEP 18: Click on permission and click on the add permissions a dropdown list will get displayed click on attach policies.

Permissions policies (0) [Info](#)

You can attach up to 10 managed policies.

[Add permissions](#) [Attach policies](#) [Create inline policy](#)

[Filter by Type](#) [All types](#)

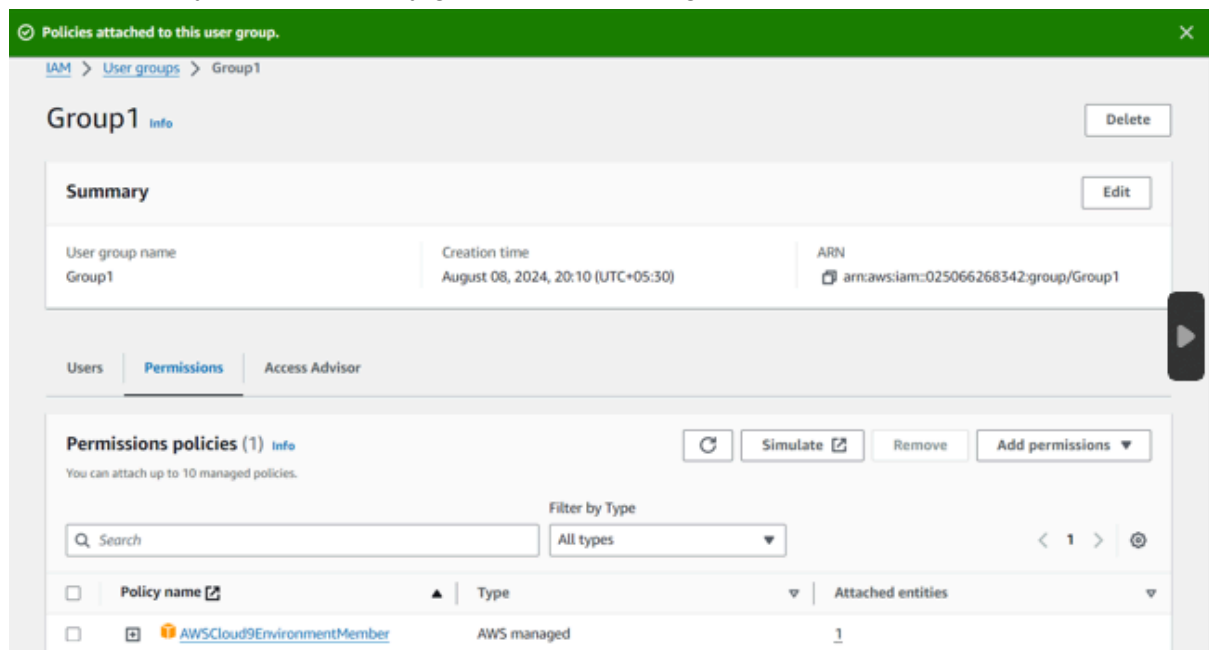
Policy name	Type	Attached entities
No resources to display		

STEP 19: Select the policy and click on attach policy.

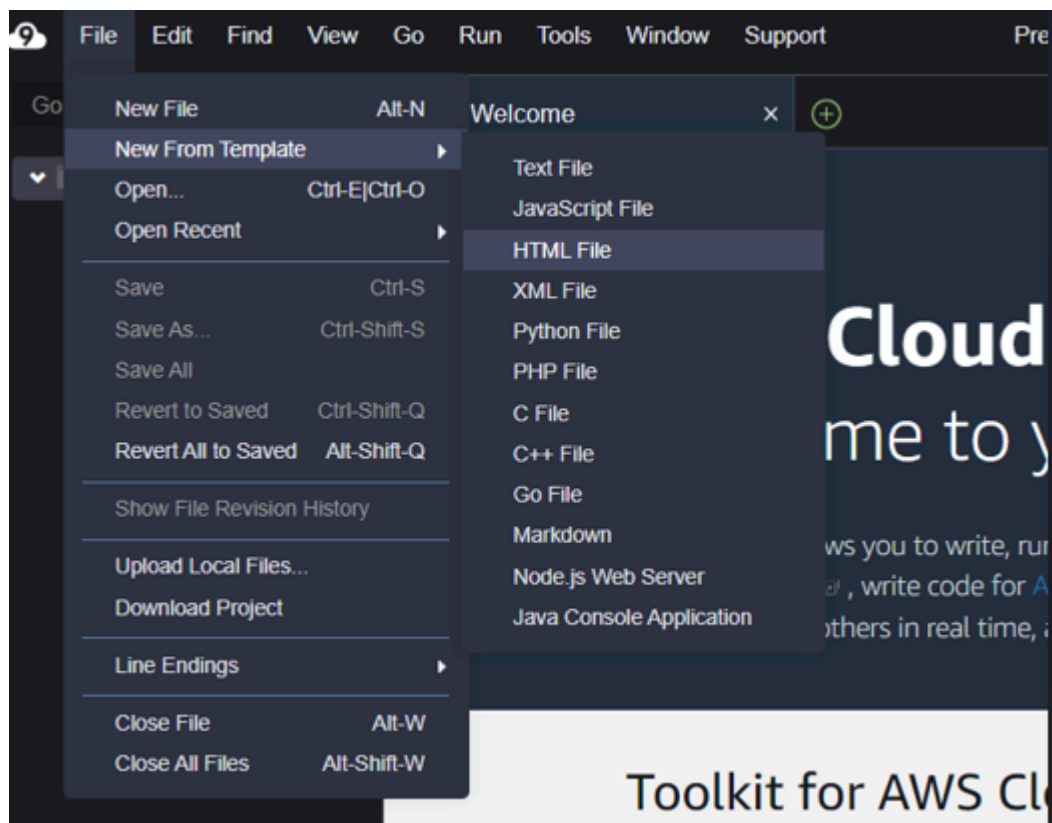
<input type="checkbox"/>	AmazonCloudWatchRUMFullAccess	AWS managed	None	Grants full access permissions for the ...
<input type="checkbox"/>	AmazonCloudWatchRUMReadOnly...	AWS managed	None	Grants read only permissions for the A...
<input type="checkbox"/>	AmazonDMSCloudWatchLogsRole	AWS managed	None	Provides access to upload DMS replicat...
<input type="checkbox"/>	AmazonGrafanaCloudWatchAccess	AWS managed	None	This policy grants access to Amazon CI...
<input type="checkbox"/>	AmazonSageMakerPartnerService...	AWS managed	None	Service role policy used by the AWS CI...
<input type="checkbox"/>	AmazonSageMakerServiceCatalog...	AWS managed	None	Service role policy used by the AWS CI...
<input type="checkbox"/>	AWSAppSyncPushToCloudWatch...	AWS managed	None	Allows AppSync to push logs to user's ...
<input type="checkbox"/>	AWSCloud9Administrator	AWS managed	None	Provides administrator access to AWS ...
<input checked="" type="checkbox"/>	AWSCloud9EnvironmentMember	AWS managed	None	Provides the ability to be invited into ...
<input type="checkbox"/>	AWSCloud9SSMInstanceProfile	AWS managed	None	This policy will be used to attach a rel...
<input type="checkbox"/>	AWSCloud9User	AWS managed	None	Provides permission to create AWS Clo...
<input type="checkbox"/>	AWSCloudFormationFullAccess	AWS managed	None	Provides full access to AWS CloudForm...
<input type="checkbox"/>	AWSCloudFormationReadOnlyAccess	AWS managed	None	Provides access to AWS CloudFormatio...
<input type="checkbox"/>	AWSCloudHSMFullAccess	AWS managed	None	Provides full access to all CloudHSM re...
<input type="checkbox"/>	AWSCloudHSMReadOnlyAccess	AWS managed	None	Provides read only access to all Cloud...

[Cancel](#) [Attach policies](#)

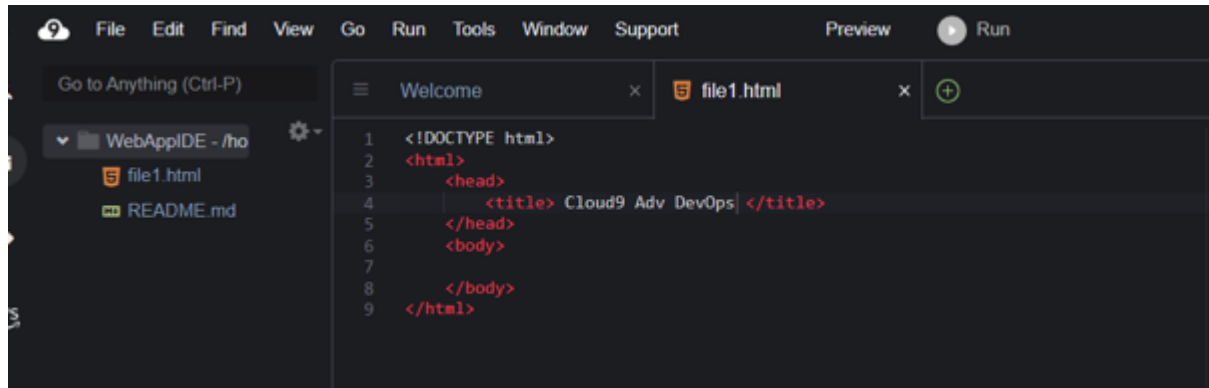
STEP 20: Policy will successfully get attached to the group.



STEP 21: Go on Cloud9 and click on File in that select New From Template and select HTML File.

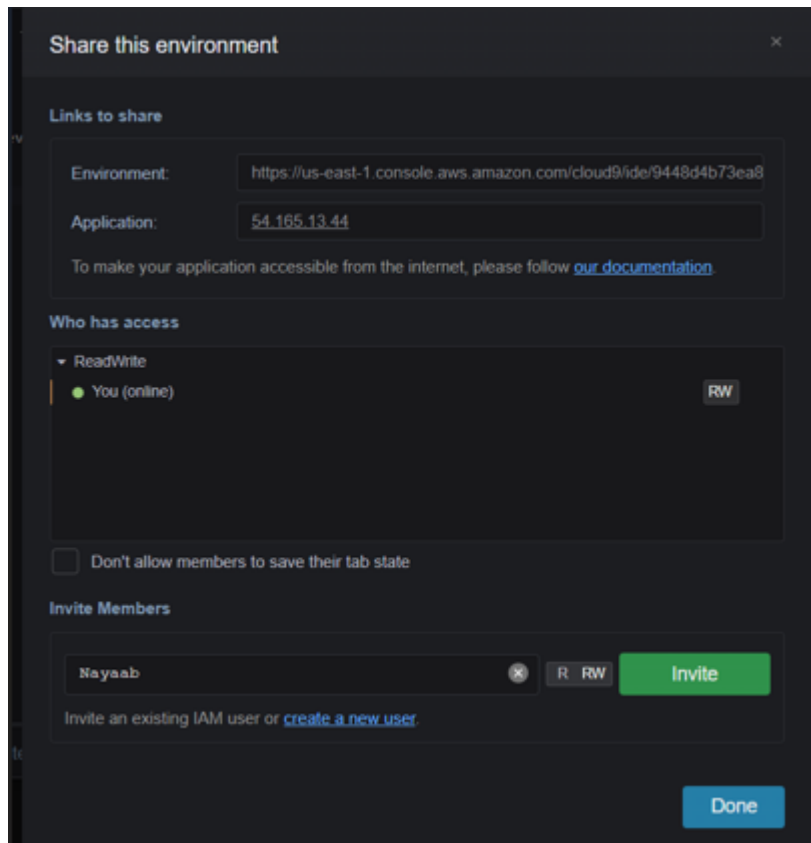


STEP 22: Write a html code .



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Cloud9 Adv DevOps </title>
5   </head>
6   <body>
7
8   </body>
9 </html>
```

STEP 23: Go on Share this environment and edit who you want to give access to and then click on done.



Share this environment

Links to share

Environment:

Application:

To make your application accessible from the internet, please follow [our documentation](#)

Who has access

ReadWrite

- You (online) RW

☐ Don't allow members to save their tab state

Invite Members

Invite an existing IAM user or [create a new user](#)