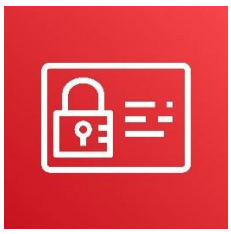
Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)
Department of Computer Science and Engineering (Data Science)

# Cloud Computing Lab

# Experiment No.: 2

## Implement user level authentication on your cloud applications.:

## AWS Identity and Access Management (IAM)



Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Department of Computer Science and Engineering (Data Science)**

**Subject: Cloud Computing Laboratory (DJ19DSL6011)**

**AY: 2022-23**

**Experiment 2**

**Name : Sarvagya Singh**

**SAPID : 60009200030** **BATCH : K1**

## Aim: Implement user level authentication on your cloud applications.

- Objectives: In this lab, you will explore users, groups, and policies in the AWS Identity and Access Management (IAM) service.
    - Exploring pre-created IAM Users and Groups
    - Inspecting IAM policies as applied to the pre-created groups
    - Following a real-world scenario, adding users to groups with specific capabilities enabled.
    - Locating and using the IAM sign-in URL.
    - Experimenting with the effects of policies on service access.
- Outcomes: The learner will be able to create and manage Users and Groups as well as understand the effect of policies on various service access.
- Hardware / Software Required: Internet, AWS console

## Theory:

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform.

IAM provides the infrastructure necessary to control authentication and authorization for your account. The IAM infrastructure includes the following elements:

- **Terms :** Learn more about IAM terms.
    - **IAM Resources :** The user, group, role, policy, and identity provider objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.
    - **IAM Identities:** The IAM resource objects that are used to identify and group. You can attach a

policy to an IAM identity. These include users, groups, and roles.

- **IAM Entities :** The IAM resource objects that AWS uses for authentication. These include IAM users and roles.

➢ **Principals :** A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS. The principal is authenticated as the AWS account root user or an IAM entity to make requests to AWS. As a best practice, do not use your root user credentials for your daily work. Instead, create IAM entities (users and roles). You can also support federated users or programmatic access to allow an application to access your AWS account.

➢ **Request** : When a principal tries to use the AWS Management Console, the AWS API, or the AWS CLI, that principal sends a *request* to AWS. The request includes the following information:

- **Actions or operations** – The actions or operations that the principal wants to perform. This can be an action in the AWS Management Console, or an operation in the AWS CLI or AWS API.
- **Resources** – The AWS resource object upon which the actions or operations are performed.
- **Principal** – The person or application that used an entity (user or role) to send the request. Information about the principal includes the policies that are associated with the entity that the principal used to sign in.
- **Environment data** – Information about the IP address, user agent, SSL enabled status, or the time of day.
- **Resource data** – Data related to the resource that is being requested. This can include information such as a DynamoDB table name or a tag on an Amazon EC2 instance.

➢ **Authentication :** A principal must be authenticated (signed in to AWS) using their credentials to send a request to AWS. Some services, such as Amazon S3 and AWS STS, allow a few requests from anonymous users. However, they are the exception to the rule. To authenticate from the console as a root user, you must sign in with your email address and password. As an IAM user, provide your account ID or alias, and then your user name and password. To authenticate from the API or AWS CLI, you must provide your access key and secret key. You might also be required to provide additional security information. For example, AWS recommends that you use multi- factor authentication (MFA) to increase the security of your account. To learn more about the IAM entities that AWS can authenticate, see IAM users and IAM roles.

➢ **Authorization :** You must also be authorized (allowed) to complete your request. During authorization, AWS uses values from the request context to check for policies that apply to the request. It then uses the policies to determine whether to allow or deny the request. Most policies are stored in AWS as JSON documents and specify the permissions for principal entities. There are several types of policies that can affect whether a request is authorized. To provide your users with permissions to access the AWS resources in their own account, you need only identity-based policies. Resource-based policies are popular for granting cross- account access. The other policy types are advanced features and should be used carefully.

AWS checks each policy that applies to the context of your request. If a single permissions policy includes a denied action, AWS denies the entire request and stops evaluating. This is called an explicit deny. Because requests are denied by default, AWS authorizes your request only if every part of your request is allowed by the applicable permissions policies. The evaluation logic for a request within a single account follows these general rules:

- By default, all requests are denied. (In general, requests made using the AWS account root user credentials for resources in the account are always allowed.)
- An explicit allow in any permissions policy (identity-based or resource-based) overrides this default.
- The existence of an Organizations SCP, IAM permissions boundary, or a session policy overrides the allow. If one or more of these policy types exists, they must all allow the request. Otherwise, it is implicitly denied.
- An explicit deny in any policy overrides any allows.

➢ **Actions or operations :** After your request has been authenticated and authorized, AWS approves the actions or operations in your request. Operations are defined by a service, and include things that you can do to a resource, such as viewing, creating, editing, and deleting that resource. For example, IAM supports approximately 40 actions for a user resource, including the following actions:

  - CreateUser
  - DeleteUser
  - GetUser
  - UpdateUser

- To allow a principal to perform an operation, you must include the necessary actions in a policy that applies to the principal or the affected resource.

➢ **Recourses :** After AWS approves the operations in your request, they can be performed on the related resources within your account. A resource is an object that exists within a service. Examples include an Amazon EC2 instance, an IAM user, and an Amazon S3 bucket. The service defines a set of actions that can be performed on each resource. If you create a request to perform an unrelated action on a resource, that request is denied. For example, if you request to delete an IAM role but provide an IAM group resource, the request fails.

✓ **Task1 : Explore the users and groups**

In this task, you will explore the users and groups that have already been created for you in IAM. First, note the Region that you are in; for example, N. Virginia. The Region is displayed in the upper-right corner of the console page. You might need this information later in the lab.

1. Choose the Services menu, locate the Security, Identity, & Compliance services, and choose IAM.
2. In the navigation pane on the left, choose Users.
3. The following IAM users have been created:
   - user-1
   - user-2

- user-3

4. Choose the name of user-1.
   - This brings you to a summary page for user-1. The Permissions tab will be displayed.
   - Notice that user-1 does not have any permissions.
5. Choose the Groups tab. Notice that user- 1 also is not a member of any groups.
6. Choose the Security credentials tab. Notice that user-1 is assigned a Console password. This allows the user to access the AWS Management Console.
7. In the navigation pane on the left, choose User groups. The following groups have already been created for you:
   - EC2-Admin
   - EC2-Support
   - S3-Support
8. Choose the name of the EC2-Support group. This brings you to the summary page for the EC2-Support group.
9. Choose the Permissions tab. This group has a managed policy called AmazonEC2ReadOnlyAccess associated with it. Managed policies are prebuilt policies (built either by AWS or by your administrators) that can be attached to IAM users and groups. When the policy is updated, the changes to the policy are immediately applied against all users and groups that are attached to the policy.
10. Under Policy Name, choose the link for the **AmazonEC2ReadOnlyAccess** policy.

    **Choose the {} JSON tab**.
    - A policy defines what actions are allowed or denied for specific AWS resources. This policy is granting permission to List and Describe (view) information about Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing, Amazon CloudWatch, and Amazon EC2 Auto Scaling. This ability to view resources, but not modify them, is ideal for assigning to a support role.
    - Statements in an IAM policy have the following basic structure:
      - Effect says whether to Allow or Deny the permissions.
      - Action specifies the API calls that can be made against an AWS service (for example, cloudwatch:ListMetrics).
      - Resource defines the scope of entities covered by the policy rule (for example, a specific Amazon Simple Storage Service [Amazon S3] bucket or Amazon EC2 instance; an asterisk [ * means any resource).
11. In the navigation pane on the left, choose

**User groups**.

✓ **Task 2: Add users to groups**

12. Choose the name of the **S3-Support group**.
13. Choose the **Permissions** tab. The S3- Support group has the AmazonS3ReadOnlyAccess policy attached.
14. Under Policy Name, choose the link for the **AmazonS3ReadOnlyAccess** policy.
15. Choose the **{} JSON** tab. This policy has permissions to Get and List for all resources in Amazon S3.
16. In the navigation pane on the left, choose

**User groups**.

17. Choose the name of the **EC2-Admin** group.
18. Choose the **Permissions** tab. This group is different from the other two. Instead of a managed policy, the group has an inline policy, which is a policy assigned to just one user or group. Inline policies are typically used to apply permissions for specific situations.
19. Under **Policy Name**, choose the name of the **EC2-Admin-Policy** policy.
20. Choose the **JSON** tab. This policy grants permission to Describe information about Amazon EC2 instances, and also the ability to Start and Stop instances.
21. At the bottom of the screen, choose Cancel to close the policy.

You have recently hired *user-1* into a role where they will provide support for Amazon S3. You will add them to the *S3-Support* group so that they inherit the necessary permissions via the attached *AmazonS3ReadOnlyAccess* policy. Ignore any "not authorized" errors that appear during this task. They are caused by your lab account having limited permissions and will not impact your ability to complete the lab.

| User | In Group | Permissions |
|---|---|---|
| user-1 | S3-Support | Read-Only access to Amazon S3 |
| user-2 | EC2-Support | Read-Only access to Amazon EC2 |
| user-3 | EC2-Admin | View, Start and Stop Amazon EC2 instances |

- **Add user-1 to the S3-Support group:**

22. In the left navigation pane, choose User groups.
23. Choose the name of the S3-Support group.
24. On the Users tab, choose Add users.
25. Select user-1, and choose Add users. On the Users tab, notice that *user-1* has been added to the group.

- **Add user-2 to the EC2-Support group :** You have hired ***user-2*** into a role where they will provide support for Amazon EC2. You will add them to the *EC2-Support* group so that they inherit the necessary permissions via the attached *AmazonEC2ReadOnlyAccess* policy.

26. Use what you learned from the previous steps to add *user-2* to the *EC2-Support* group. *user-2* should now be part of the *EC2-Support* group.

- **Add user-3 to the EC2-Admin group :** You have hired *user-3* as your Amazon EC2 administrator to manage your EC2 instances. You will add them to the *EC2-Admin* group so that they inherit the necessary permissions via the attached *EC2-Admin-Policy*.

27. Use what you learned from the previous steps to add *user-3* to the *EC2-Admin* group. *user-3* should now be part of the *EC2-Admin* group.
28. In the navigation pane on the left, choose **User groups**. Each group should have a 1 in the Users column. This indicates the number of users in each group.

## ✓ Task 3: Sign in and test users :

In this task, you will test the permissions of each IAM user in the console.

### Get the console sign-in URL

29. In the navigation pane on the left, choose **Dashboard**.

Notice the **Sign-in URL for IAM users in this account** section at the top of the page. The sign-in URL looks similar to the following: https://123456789012.signin.aws.amazon.com/console This link can be used to sign in to the AWS account that you are currently using.

30. Copy the sign-in link to a text editor.

### Test user-1 permissions

31. Open a private or incognito window in your browser.
32. Paste the sign-in link into the private browser, and press **ENTER.** You will now sign-in as *user-1*, who has been
    - **IAM user name:** user-1
    - **Password:** Lab-Password1

    hired as your Amazon S3 storage support staff.
33. Sign in with the following credentials:
34. Choose the **Services** menu, and choose **S3**.
35. Choose the name of one of your buckets, and browse the contents.
36. Because this user is part of the *S3-Support* group in IAM, they have permissions to view a list of Amazon S3 buckets and their contents. Now, test whether the user has access to Amazon EC2.
37. Choose the **Services** menu, and choose **EC2**.
38. In the left navigation pane, choose **Instances**.
39. You cannot see any instances. Instead, an error message says *you are not authorized to perform this operation*. This user has not been assigned any permissions to use Amazon EC2. You will now sign in as *user-2*, who has been hired as your Amazon EC2 support person.
40. First, sign out *user-1* from the console:
    - In the upper-right corner of the page, choose **user-1**.
    - Choose **Sign Out**.

### Test user-2 permissions

- **IAM user name:** user-2
- **Password:** Lab-Password2

41. Paste the sign-in link into the private browser again, and press ENTER.
42. Sign in with the following credentials:
43. Choose the **Services** menu, and choose **EC2**.
44. In the navigation pane on the left, choose **Instances**.
45. You are now able to see an EC2 instance. However, you cannot make any changes to Amazon

EC2 resources because you have read-only permissions. If you cannot see an EC2 instance, then your Region might be incorrect. In the upper-right corner of the page, choose the Region name, and then choose the Region that you were in at the beginning of the lab (for example, **N. Virginia**).

46. Select the **EC2 instance**.
47. Choose the **Instance state** menu, and then choose **Stop instance**.
48. To confirm that you want to stop the instance, choose **Stop**.

An error message appears and says that *You are not authorized to perform this operation*. This demonstrates that the policy only allows you to view information without making changes.

49. Next, check if *user-2* can access Amazon S3.
50. Choose the **Services** menu, and choose **S3**. An error message says *You don't have permissions to list buckets* because *user-2* does not have permissions to use Amazon S3. You will now sign-in as *user-3*, who has been hired as your Amazon EC2 administrator.
51. First, sign out *user-2* from the console:
    - **In the upper-right corner of the page, choose user-2.**
    - **Choose Sign Out.**

**Test user-3 permissions**

52. Paste the sign-in link into the private browser again, and press ENTER. Sign in with the following credentials:
    - **IAM user name: user-3**
    - **Password: Lab-Password3**
53. Choose the **Services** menu, and choose **EC2**.
54. In the navigation pane on the left, choose **Instances**.
55. An EC2 instance is listed. As an Amazon EC2 Administrator, this user should have permissions to *Stop* the EC2 instance.
56. If you cannot see an EC2 instance, then your Region might be incorrect. In the upper-right corner of the page, choose the Region name, and then choose the Region that you were in at the beginning of the lab (for example, **N. Virginia**).
57. Select the EC2 instance.
58. Choose the **Instance state** menu, and then choose **Stop instance**.
59. To confirm that you want to stop the instance, choose **Stop**.
60. This time, the action is successful because *user-3* has permissions to stop EC2 instances. The **Instance state** changes to *Stopping* and starts to shut down.
61. Close your private browser window.

**Result : Paste your screen shots for every task.**

**Conclusions :**

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access. You use IAM to control who is authenticated (signed in) and authorized

(has permissions) to use resources.

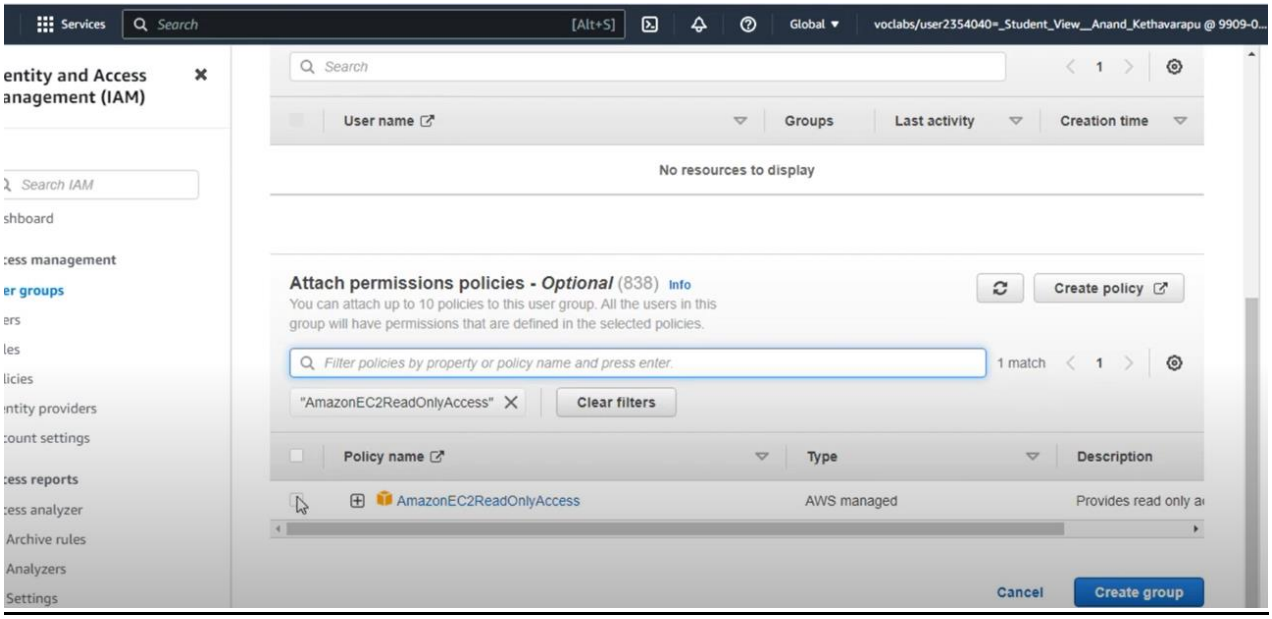
It is necessary to use IAM in every account inorder to assign the role and give the policy needed to someone. IAM is used to assign the role and attach the policy needed to that role.

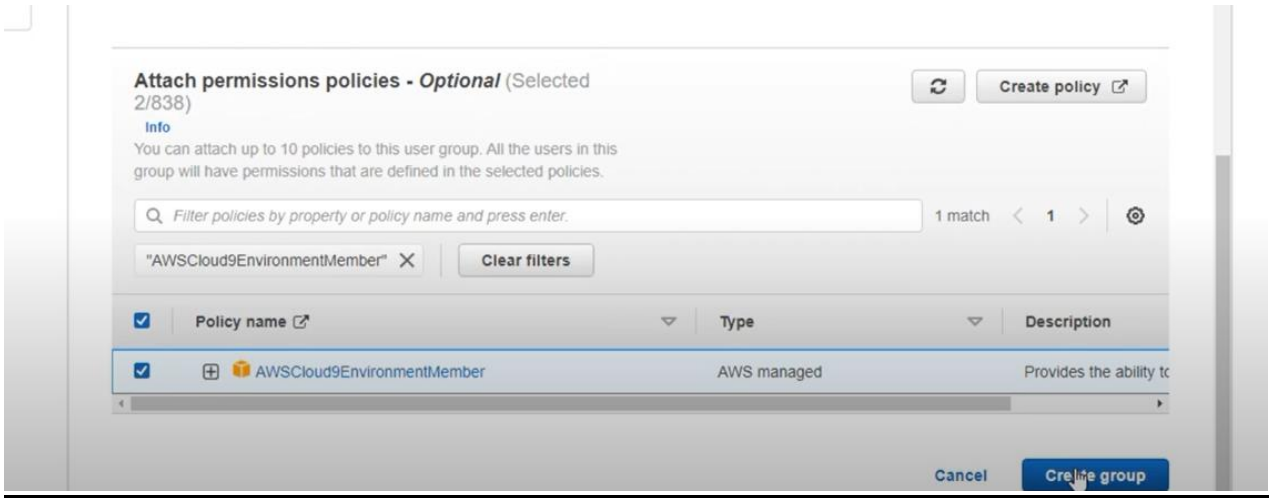
**Viva Questions:**

- What is AWS IAM ?
- Why IAM Needed?
- What is the difference between Role and Policy?

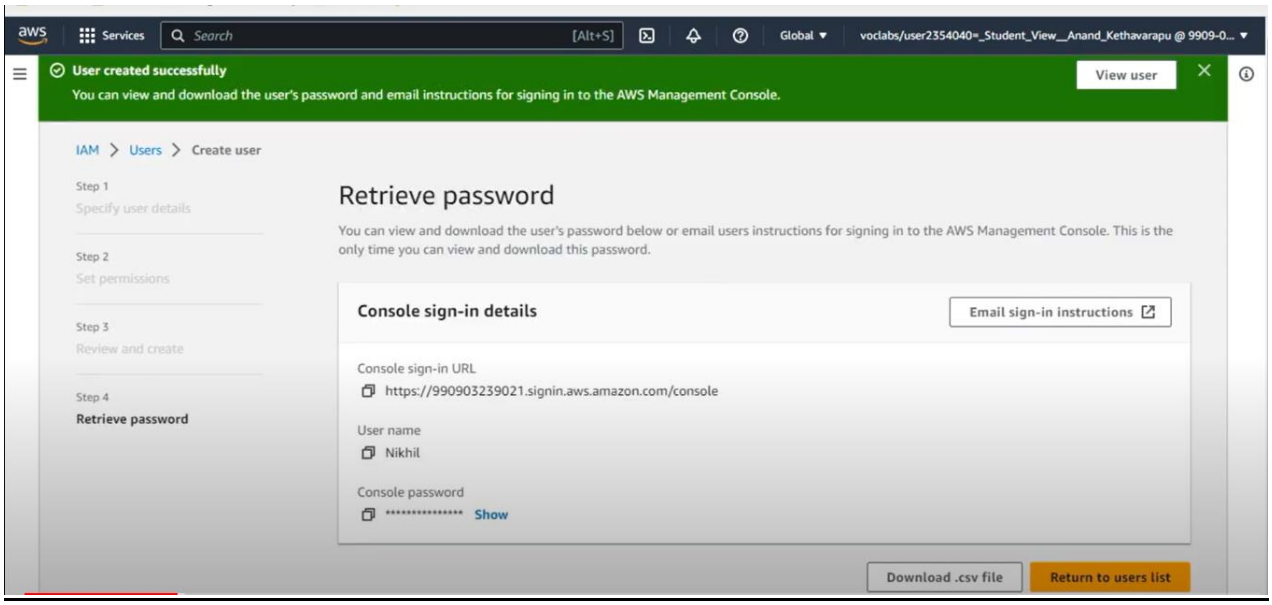
**References:** https://docs.aws.amazon.com/ec2/index.html

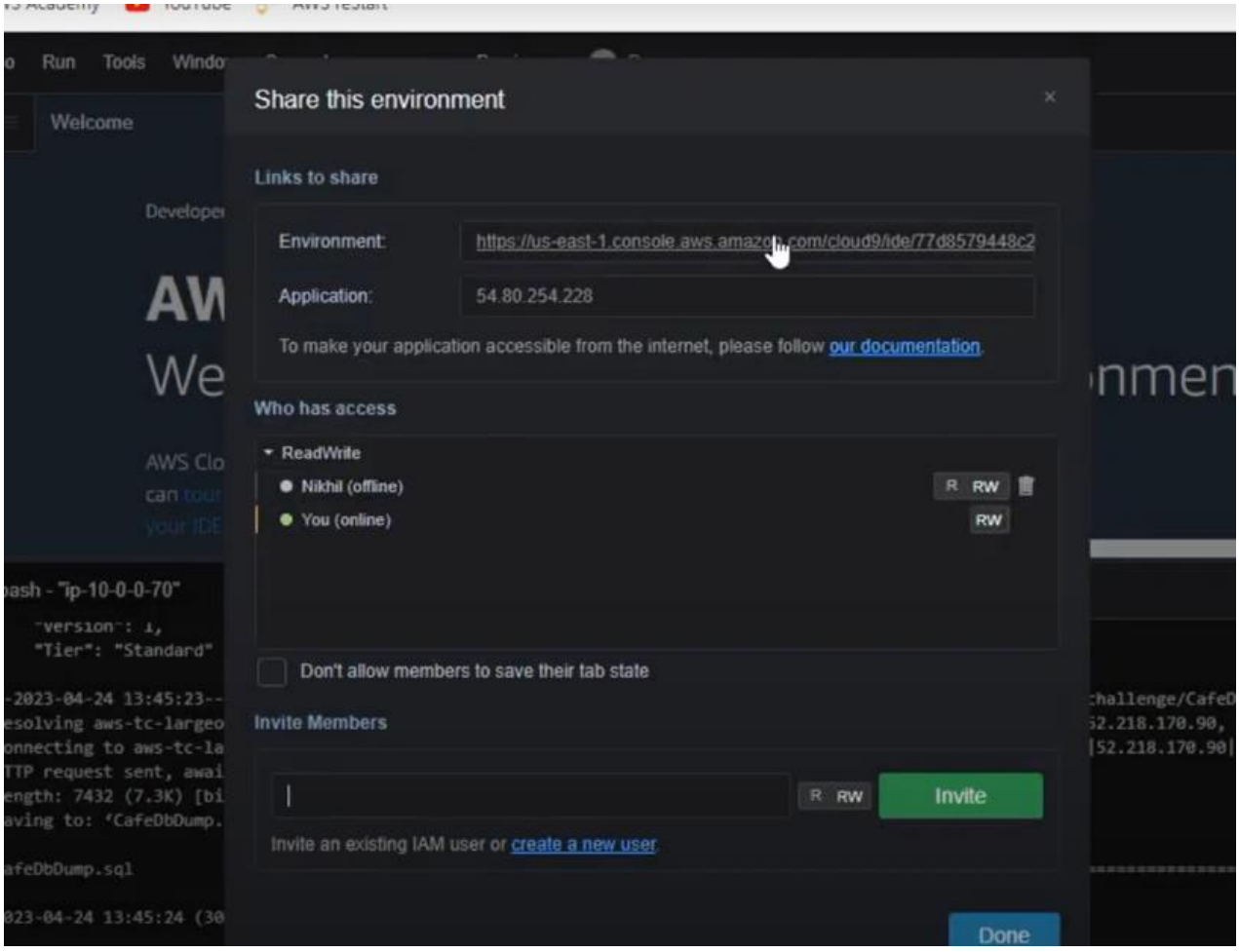
## Screenshots :

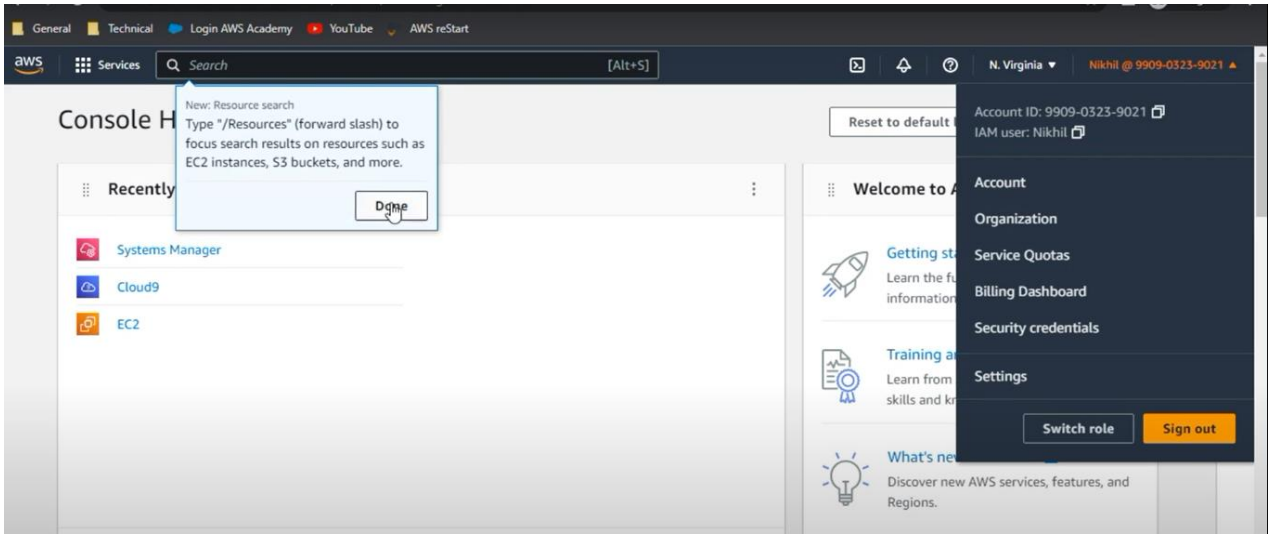


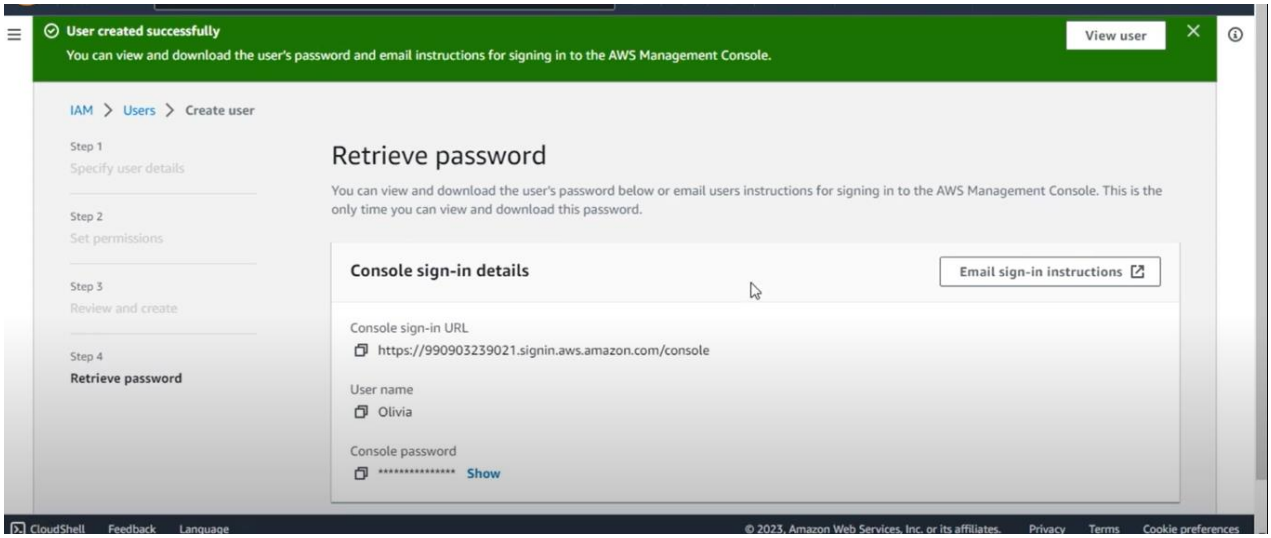



- Creating the user nikhil

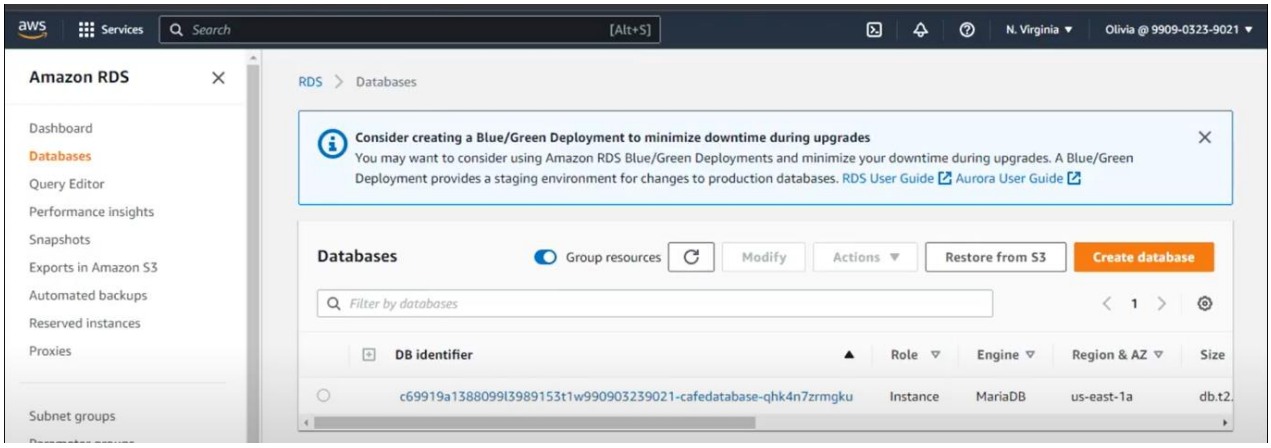


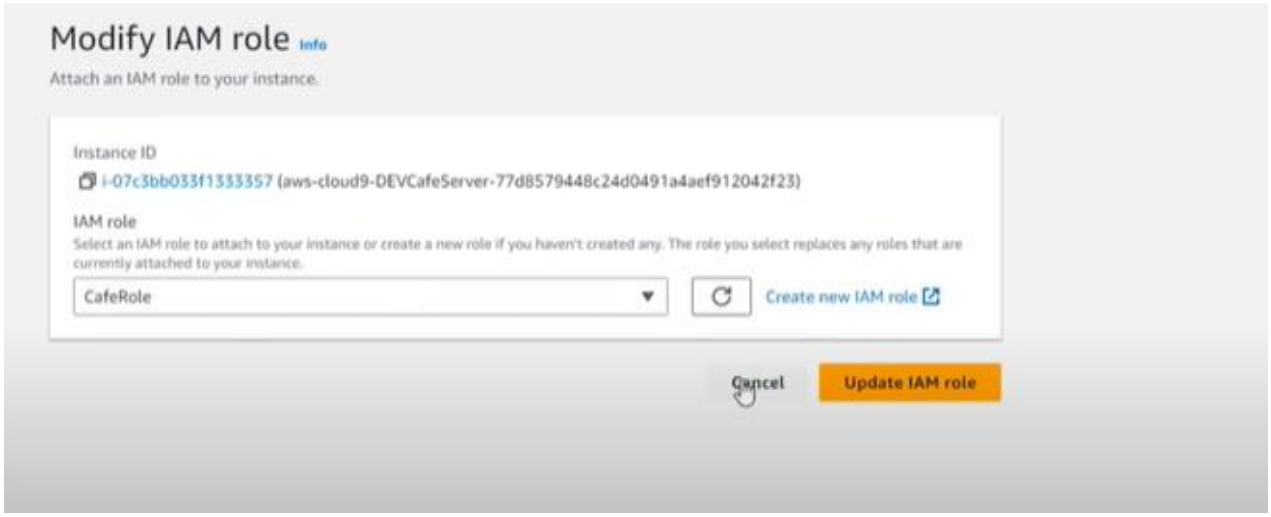




General
Technical
Login AWS Academy
YouTube
AWS reStart
aws
Services
Search
[Alt+S]
N. Virginia
Nikhil @ 9909-0323-9021
New: Resource search
Type "/Resources" (forward slash) to focus search results on resources such as EC2 instances, S3 buckets, and more.
Done
Console H
Reset to default
Recently
Welcome to A
Systems Manager
Cloud9
EC2
Account ID: 9909-0323-9021
IAM user: Nikhil
Account
Organization
Service Quotas
Billing Dashboard
Security credentials
Settings
Switch role
Sign out
Getting st
Learn the fu
Training a
Learn from
What's ne
Discover new AWS services, features, and Regions.


User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.
View user
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.
Console sign-in details
Email sign-in instructions
Console sign-in URL
https://990903239021.signin.aws.amazon.com/console
User name
Olivia
Console password
Show
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

- **Amazon RDS**


aws
Services
Search
[Alt+S]
N. Virginia
Olivia @ 9909-0323-9021
Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
RDS > Databases
Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide
Databases
Group resources
Modify
Actions
Restore from S3
Create database
Filter by databases
1
DB identifier
Role
Engine
Region & AZ
Size
c69919a1388099l3989153t1w990903239021-cafedatabase-qhk4n7zrmgku
Instance
MariaDB
us-east-1a
db.t2.




- Created the policy