Index

Experiment 1:

Create storage S3 bucket and upload files into bucket.

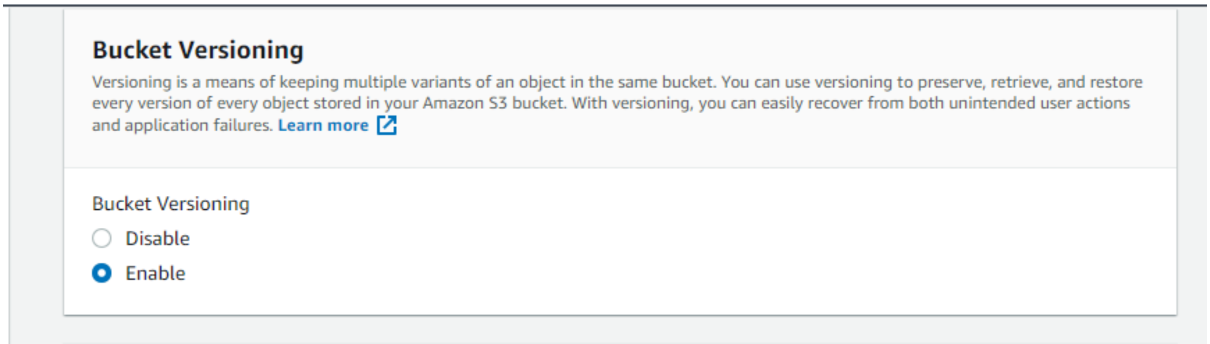
- Login to AWS Educate account.
- Select AWS Storage lab.
- To launch the lab, at the top of the page, choose Start lab.
- You must wait for the provisioned AWS services to be ready before you can continue.
- To open the lab, choose Open Console.
- You are automatically signed in to the AWS Management Console in a new web browser tab.

1. In Amazon management Console in services under the storage click s3.
2. Click create bucket.
3. Enter bucket name, select region.
3. For Object Ownership, choose ACLs enabled.
4. For Block Public Access settings for this bucket, clear the check box for Block *all* public access, and then select the box that states I acknowledge that the current settings might result in this bucket and the objects within becoming public.
5. Click create bucket.
6. Click on bucket name in bucket console click overview.
7. In overview click on upload files in that click add files. Select the files and add click next.
8. We can choose the storage class for this project we will keep it as standard.
9. Click upload.
10. If we click on the file we can see the information of the file on right hand side.
11. In link u can see the URL of file. Copy and open it will show access denied.
12. To access the files go to bucket click on permissions.
13. Click on manage public permissions setting and enable all options to give access to everyone keep storage class as standard.
14. Click on save.
15. Right click on uploaded file and click make public.
16. Copy and open link. You can see uploaded file content.

Experiment 2:

Demonstrate Creating bucket uploading files and creating Versioning on S3 bucket.

1. Create S3 bucket and upload files into the bucket. (Refer experiment 1)
2. Bucket versioning will disable by default. Enable versioning.



Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning

○ Disable

● Enable

3. Upload same files again.
4. We can see the file and latest version ID.
5. Try and upload same file again.
6. One more version of file will be created.
7. You can also delete the latest version and also disable the versioning.

Experiment 3:

Demonstrate using bucket policy to secure bucket.

You want to protect your website files and make sure that no one can delete them. To do this, you apply a bucket policy that denies delete privileges on your website files.

1. Create S3 bucket and upload files into the bucket. (Refer experiment 1)
2. Return to the Amazon S3 console, and choose the Permissions tab.
3. Under Bucket policy, choose Edit
4. Click on Generate bucket policy and generate policy to deny delete bucket.
5. Copy the following policy text. In the Policy text editor, replace the existing policy text with this text:

```
{
    "Version": "2012-10-17",
    "Id": "MyBucketPolicy",
    "Statement": [
        {
            "Sid": "BucketPutDelete",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:DeleteObject",
            "Resource": [
                "arn:aws:s3:::<bucket-name>/index.html",
                "arn:aws:s3:::<bucket-name>/script.js",
                "arn:aws:s3:::<bucket-name>/style.css"
   ]
        }
    ]
}
```

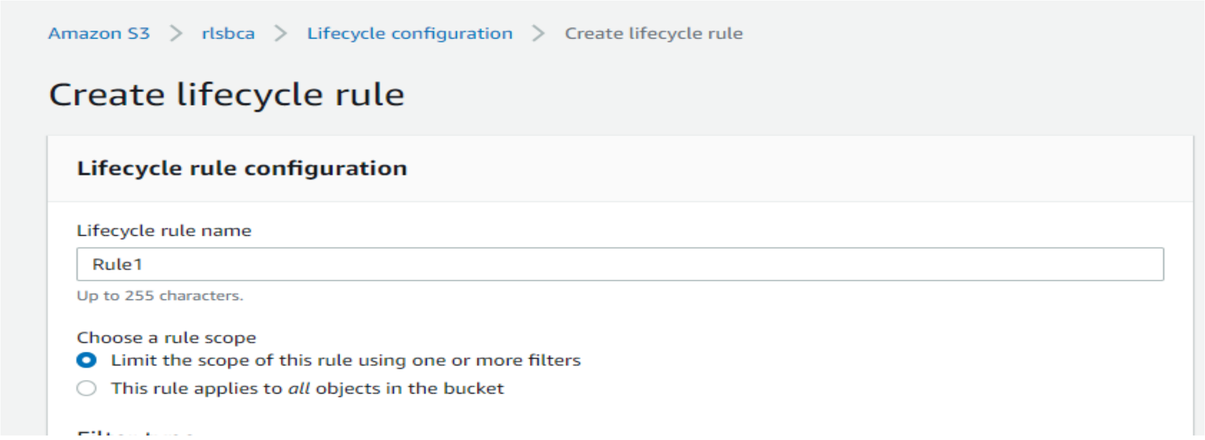
6. Choose Save changes.

7. Return to the Object tab
8. Select file(object)
9. Choose Delete.
10. In the Delete objects panel, enter delete to confirm that you want to remove this file.
11. Choose Delete objects
12. Notice that the index.html file is listed in the Failed to delete pane.
13. This confirms that your policy is working and preventing the website's files from being deleted.
14. Choose Close to return to the Objects tab.

Experiment 4:

Demonstration to create lifecycle bucket policy for S3.

1. In Amazon management Console in services under the storage click s3.
2. Create a bucket and upload files into bucket.
3. Click on bucket name.
4. Click on Management tab.
5. Click create lifecycle rule.
6. Enter lifecycle rule name. Add prefix any tag.
7. In lifecycle rule action select Transition *current* versions of objects between storage classes
8. In Transition current versions of objects between storage classes
   Select storage class transitions, Enter 30 days
9. Click on create


Amazon S3 > rlsbca > Lifecycle configuration > Create lifecycle rule

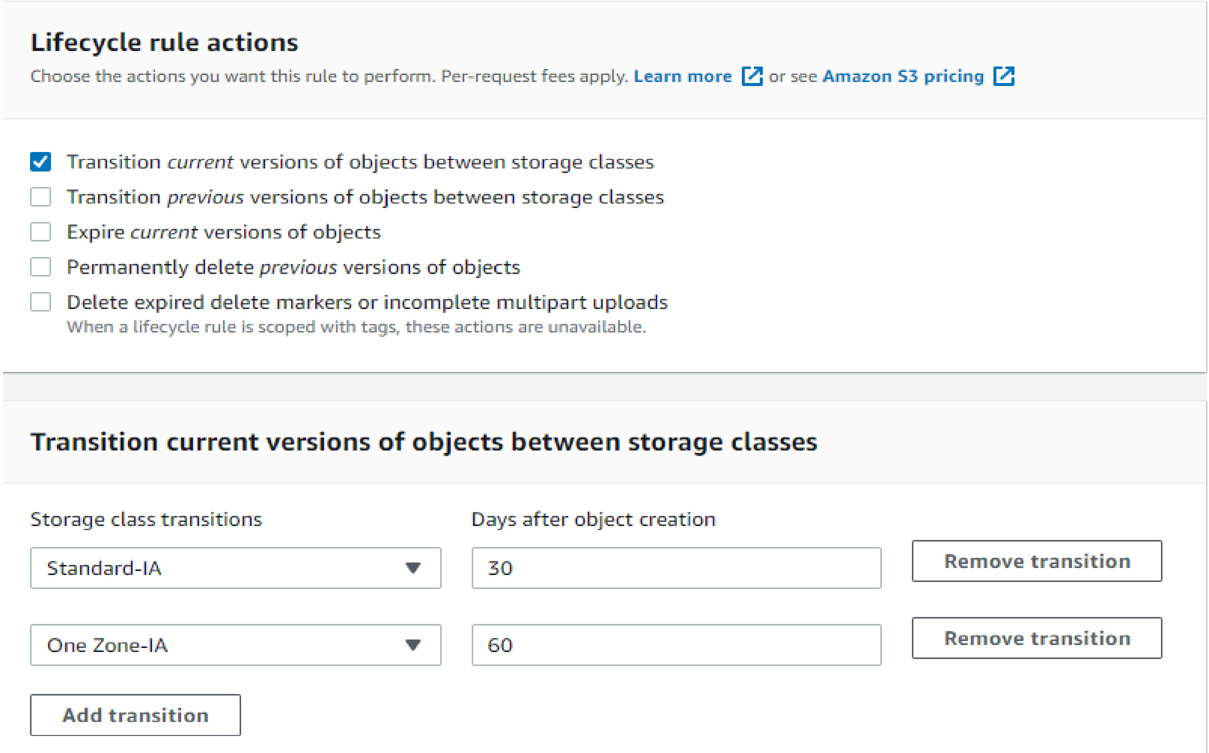
Create lifecycle rule

Lifecycle rule configuration

Lifecycle rule name: Rule1

Up to 255 characters.

Choose a rule scope
● Limit the scope of this rule using one or more filters
○ This rule applies to *all* objects in the bucket



Lifecycle rule actions

Choose the actions you want this rule to perform. Per-request fees apply. Learn more or see Amazon S3 pricing

☑ Transition *current* versions of objects between storage classes

□ Transition *previous* versions of objects between storage classes

□ Expire *current* versions of objects

□ Permanently delete *previous* versions of objects

□ Delete expired delete markers or incomplete multipart uploads
When a lifecycle rule is scoped with tags, these actions are unavailable.

Transition current versions of objects between storage classes

Storage class transitions	Days after object creation	
Standard-IA	30	Remove transition
One Zone-IA	60	Remove transition

Add transition

10. Create 5 different rules for transition current objects, expire current versions of object, permanently delete objects, delete expired objects.
11. You can edit, delete the rule.

Experiment 5:

Set up a static website using Amazon S3.

Create a bucket and enable static website hosting.

Hosting static website:

1. Create S3 bucket (follow steps of experiment no. 1) Click on bucket upload following files and add file into the bucket.
   a. index.html
   b. script.js
   c. style.css

2. In manage public permissions click on the grant public read access, also change file permissions.
3. click upload.
4. Go to bucket properties in static website hosting click on use this bucket to host a website.
5. Choose **Edit**
6. Configure the following settings:

   Static web hosting: Choose **Enable**.

   Hosting type: Choose **Host a static website**.

   Index document: Enter index.html

   Error document: Enter error.html
7. **Note**: You must enter index.html and error.html even though they are already displayed.
8. Choose **Save changes**
9. In the **Static website hosting** panel under **Bucket website endpoint**, choose the link.
10. Copy URL from static website hosting and open in browser.

Updating the website

- Although you have configured a policy to prevent deletion of website files, you can still update the website by editing the HTML file and uploading it to the S3 bucket again.
- Amazon S3 is an object storage service, so you must upload the whole file. This action replaces the existing object in your bucket. You cannot edit the contents of an object; instead, you must replace the whole object.

1. On your computer, load the **index.html** file into a text editor (for example, Notepad or TextEdit).
2. Find the text **Served from Amazon S3**, and replace it with
3. Created by <YOUR-NAME>
4. and substitute your name for *<YOUR-NAME>* (for example, **Created by Jane**).
5. Save the file.
6. Return to the Amazon S3 console, and upload the **index.html** file that you just edited.
7. Choose **index.html**, and in the **Actions** menu, choose the **Make public using ACL** option again.
8. Choose **Make public**.
9. Return to the web browser tab with the static website, and refresh the page.
10. Your name should now be on the page.
11. Your static website is now accessible on the internet. Because it is hosted on Amazon S3, the website has high availability and can serve high volumes of traffic without using any servers.

Experiment 6:

Create EC2 server to host website.

Launching your EC2 instance

In this task, you launch an EC2 instance with termination protection. Termination protection prevents you from accidentally terminating an EC2 instance.

In the AWS Management Console on the **Services** menu, enter **EC2**. From the search results, choose **EC2**.

In the left navigation pane, choose **EC2 Dashboard** to ensure that you are on the dashboard page.

In the **Launch instance** section, choose the **Launch instance** dropdown list, and then choose **Launch instance**.

Step 1: Name your EC2 instance

In the **Name and tags** pane, in the **Name** text box, enter `Web-Server`

Step 2: Choose an AMI

1. An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud.
2. The **Quick Start** list contains the most commonly used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.
3. Locate the **Application and OS Images (Amazon Machine Image)** section. It is just below the **Name and tags** section.
4. In the search box, enter `Windows Server 2019 Base` and press Enter.
5. Next to **Microsoft Windows Server 2019 Base**, choose **Select**.
6. Choose **Confirm Changes**.

Step 3: Choose an instance type

1. In this step, you choose a **t2.micro** instance. This instance type has 1 virtual CPU and 1 GiB of memory.
2. In the **Instance type** section, keep the default instance type, **t2.micro**.

Step 4: Configure a key pair

In the **Key pair (login)** section, from the **Key pair name - *required*** dropdown list, choose **Proceed without a key pair (not recommended)**.

Step 5: Configure the network settings

1. In the **Network settings** section, choose **Edit**.
2. From the **VPC - *required*** dropdown list, choose **Lab VPC**.
3. The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.
4. For **Security group name - *required***, enter `Web Server security group`

Step 6: Launch an EC2 instance

1. Now that you have configured your EC2 instance settings, it is time to launch your instance.
2. In the **Summary** section, choose **Launch instance**.
3. A message indicates that you have successfully initiated the launch of your instance.
4. Choose **View all instances**

Experiment no 7:

Demonstrate creating database and configuring web application communication with database instance.

Creating an Amazon RDS database

1. On the **Services** menu, choose **RDS**.
2. Choose **Create database**
3. For this lab, you will keep the **Choose a database creation method** as **Standard create** to understand the full set of features available.
4. Under **Engine options**, select **MySQL**.
5. For Version, keep MySQL 8.0.28.
6. In the **Templates** section, select **Dev/Test**.
7. In the **Availability and durability** section, choose **Single DB instance**
8. In the **Settings** section next, configure the following options:
9. **DB instance identifier:** inventory-db
10. **Master username:** admin
11. **Master password:** lab-password
12. **Confirm password:** lab-password
13. In the **Instance configuration** section, configure the following options for DB instance class:
14. Choose **Burstable classes (includes t classes)**.
15. Choose **db.t3.micro**.
16. In the **Storage** section next,
17. For **Storage type** choose **General Purpose SSD (gp2)** from the Dropdown menu.
18. For **Allocated storage** keep 20
19. Clear or Deselect **Enable storage autoscaling**.
20. In the **Connectivity** section, configure the following options:
21. For **Compute resource**: keep default **Don't connect to an EC2 compute resource**, as you will establish this manually at a later stage.
22. For **Virtual private cloud (VPC)** Choose **Lab VPC** from the Dropdown menu.
23. For **DB Subnet group**, keep default value **rds-lab-db-subnet-group**
24. For **Public access** Keep default value (**No**)

25. For **VPC security group (firewall)**
    a. Choose the **X** on **default** to remove this security group.
    b. Choose **DB-SG** from the dropdown list to add it.
    c. For **Availability Zone**, Keep default **No preference**
26. For **Database authentication** keep default value **Password authentication**
27. In the **Monitoring** section
28. Clear/DeSelect the **Enable Enhanced monitoring** option.
29. Expand the following **Additional configuration** section by choosing
30. Under **Database options**, for **Initial database name**, enter inventory
31. At the bottom of the page, choose **Create database**
32. You should receive this message: **Creating database inventory-db**.

Configuring web application communication with the database instance

1. On the **Services** menu, choose **EC2**.
2. In the left navigation pane, choose **Instances**.
3. In the center pane, there should be a running instance that is named **App Server**.
4. Select the check box for the **App Server** instance.
5. In the **Details** tab, copy the **Public IPv4 address** to your clipboard.
6. **Tip:** You can choose copy to copy the displayed value the displayed value.
7. Open a new web browser tab, paste the IP address into the address bar, and then press Enter.
8. The web application should appear. It does not display much information because the application is not yet connected to the database.
9. Choose **Settings**.
10. You can now configure the application to use the Amazon RDS database instance that you created earlier. You first retrieve the database endpoint so that the application knows how to connect to a database.
11. Return to the AWS Management Console, but do not close the application tab. (You will return to it soon).
12. On the **Services** menu, choose **RDS**.
13. In the left navigation pane, choose **Databases**.
14. Under **DB identifier**, Choose ‘inventory-db’.

15. From the **Connectivity & security** section, copy the **Endpoint** to your clipboard.
16. It should look similar to this example: **inventorydb.crwxbgqad61a.rds.amazonaws.com**
17. Return to the browser tab with the inventory application, and enter the following values:
18. For **Endpoint**, paste the endpoint you copied earlier.
19. For **Database**, enter inventory
20. For **Username**, enter admin
21. For **Password**, enter lab-password
22. Choose **Save**.
23. The application will now Save this information into **AWS Secrets Manager ** and connect to the database, load some initial data, and display information.
24. You can use the web application to Add inventory, edit, and delete inventory information.
25. The inventory information is stored in the Amazon RDS MySQL database that you created earlier in the lab. This means that any failure in the application server will not lead to loss of any data. It also means that multiple application servers can access the same data.
26. Insert new records into the table. Ensure that the table has **5** or more inventory records.

Experiment 8:

Create IAM users, group and roles

To create one or more IAM users (console)

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users** and then choose **Add user**.
3. Type the user name for the new user. This is the sign-in name for AWS. If you want to add more than one user at the same time, choose **Add another user** for each additional user and type their user names. You can add up to 10 users at one time.

4. Select the type of access this set of users will have. You can select programmatic access, access to the AWS Management Console, or both.

- Select **AWS Management Console access** if the users require access to the AWS Management Console. This creates a password for each new user.

5. For **Console password**, choose one of the following:

- **Autogenerated password**. Each user gets a randomly generated password that meets the account password policy in effect (if any). You can view or download the passwords when you get to the **Final** page.
- **Custom password**. Each user is assigned the password that you type in the box.
- (Optional) We recommend that you select **Require password reset** to ensure that users are forced to change their password the first time they sign in.

6. Choose **Next: Permissions**.
7. On the **Set permissions** page, specify how you want to assign permissions to this set of new users. Choose one of the following three options:
    1. **Add user to group**. Choose this option if you want to assign the users to one or more groups that already have permissions policies. IAM displays a list of the groups in your account, along with their attached policies.
    2. **Copy permissions from existing user**. Choose this option to copy all of the group memberships, attached managed policies, embedded inline policies, and any existing permissions boundaries from an existing user to the new users. IAM displays a list of the users in your account. Select the one whose permissions most closely match the needs of your new users.
    3. **Attach existing policies to user directly**. Choose this option to see a list of the AWS managed and customer managed policies in your account. Select the policies that you want to attach to the new users or choose **Create policy** to open a new browser tab and create a new policy from scratch. After you create the policy, close that tab and return to your original tab to add the policy to the new user. As a best practice, we recommend that you instead attach your policies to a group and then make users members of the appropriate groups.

8. Choose **Next: Tags**.
9. (Optional) Add metadata to the user by attaching tags as key-value pairs. Choose **Next: Review** to see all of the choices you made up to this point. When you are ready to proceed, choose **Create user**.

10. To view the users' access keys (access key IDs and secret access keys), choose **Show** next to each password and access key that you want to see. To save the access keys, choose **Download .csv** and then save the file to a safe location.

To create an IAM group and attach policies (console)

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Groups** and then click **Create New Group**.
3. In the **Group Name** box, type the name of the group and then click **Next Step**.
4. In the list of policies, select the check box for each policy that you want to apply to all members of the group. Then click **Next Step**.
5. Click **Create Group**.

To create a role (console)

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the console, choose **Roles** and then choose **Create role**.
3. Choose the **Another AWS account** role type.
4. For **Account ID**, type the AWS account ID to which you want to grant access to your resources.
5. Choose **Next: Permissions**.
6. IAM includes a list of the AWS managed and customer managed policies in your account. Select the policy to use for the permissions policy or choose **Create policy** to open a new browser tab and create a new policy from scratch. Select the check box next to the permissions policies that you want anyone who assumes the role to have. If you prefer, you can select no policies at this time, and then attach policies to the role later. By default, a role has no permissions.
7. Choose **Next: Tags**.
8. (Optional) Add metadata to the role by attaching tags as key–value pairs. For more information about using tags in IAM, see Tagging IAM Users and Roles.
9. Choose **Next: Review**.
10. For **Role name**, type a name for your role. Role names must be unique within your AWS account. (Optional) For **Role description**, type a description for the new role.
11. Review the role and then choose **Create role**.

Experiment 9

Enabling a Virtual Multi-Factor Authentication (MFA) Device

To enable a virtual MFA device for an IAM user (console)

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. In the **User Name** list, choose the name of the intended MFA user.
4. Choose the **Security credentials** tab. Next to **Assigned MFA device**, choose **Manage**.
5. In the **Manage MFA Device** wizard, choose **Virtual MFA device**, and then choose **Continue**.

   IAM generates and displays configuration information for the virtual MFA device, including a QR code graphic.

   Open your virtual MFA app. For a list of apps that you can use for hosting virtual MFA devices

   Virtual MFA Applications

   Applications for your smartphone can be installed from the application store that is specific to your phone type. The following table lists some applications for different smartphone types.

Android/iPhone	Authy, Duo Mobile, LastPass Authenticator, Microsoft Authenticator, Google Authenticator

   If the virtual MFA app supports multiple virtual MFA devices or accounts, choose the option to create a new virtual MFA device or account.

6. Determine whether the MFA app supports QR codes, and then do one of the following:

- From the wizard, choose **Show QR code**, and then use the app to scan the QR code. For example, you might choose the camera icon or choose an option similar to **Scan code**, and then use the device's camera to scan the code.
- In the **Manage MFA Device** wizard, choose **Show secret key**, and then type the secret key into your MFA app.

  When you are finished, the virtual MFA device starts generating one-time passwords.

7. In the **Manage MFA Device** wizard, in the **MFA code 1** box, type the one-time password that currently appears in the virtual MFA device. Wait up to 30 seconds for the device to generate a new one-time password. Then type the second one-time password into the **MFA code 2** box. Choose **Assign MFA**.

   The virtual MFA device is now ready for use with AWS.

Experiment no. 10

Create & Delete a topic and subscriber using Simple Notification Service

Step 1: Create a topic

1. Sign in to the Amazon SNS console.
2. In the left navigation pane, choose **Topics**.
3. On the **Topics** page, choose **Create topic**.
4. By default, the console creates a FIFO topic. Choose **Standard**.
5. In the **Details** section, enter a **Name** for the topic, such as ***MyTopic*.**
6. Scroll to the end of the form and choose **Create topic**.

   The console opens the new topic's **Details** page.

Step 2: Create a subscription to the topic

1. In the left navigation pane, choose **Subscriptions**.
2. On the **Subscriptions** page, choose **Create subscription**.
3. On the **Create subscription** page, choose the **Topic ARN** field to see a list of the topics in your AWS account.
4. Choose the topic that you created in the previous step.
5. For **Protocol**, choose **Email**.
6. For **Endpoint**, enter an email address that can receive notifications.
7. Choose **Create subscription**.

   The console opens the new subscription's **Details** page.
8. Check your email inbox and choose **Confirm subscription** in the email from AWS Notifications. The sender ID is usually "no-reply@sns.amazonaws.com".
9. Amazon SNS opens your web browser and displays a subscription confirmation with your subscription ID.

Step 3: Publish a message to the topic

1. In the left navigation pane, choose **Topics**.
2. On the **Topics** page, choose the topic that you created earlier, and then choose **Publish message**.

   The console opens the **Publish message to topic** page.
3. (Optional) In the **Message details** section, enter a **Subject**, such as:

```
Hello from Amazon SNS!
```

4. In the **Message body** section, choose **Identical payload for all delivery protocols**, and then enter a message body, such as:

   ```
   Publishing a message to an SNS topic.
   ```

5. Choose **Publish message**.

   The message is published to the topic, and the console opens the topic's **Details** page.

6. Check your email inbox and verify that you received an email from Amazon SNS with the published message.

Step 4: Delete the subscription and topic

1. On the navigation panel, choose **Subscriptions**.
2. On the **Subscriptions** page, choose a *confirmed* subscription and then choose **Delete**.

   Note

   You can't delete a pending confirmation. After 3 days, Amazon SNS deletes it automatically.

3. In the **Delete subscription** dialog box, choose **Delete**.

   The subscription is deleted.

4. On the navigation panel, choose **Topics**.
5. On the **Topics** page, choose a topic and then choose **Delete**.

   Important

   When you delete a topic, you also delete all subscriptions to the topic.

6. On the **Delete topic** *MyTopic* dialog box, enter `delete me` and then choose **Delete**.

   The topic is deleted.

Experiment no 11:

Your company wants to be notified when users delete files from their Amazon S3 bucket.

You will need to do the following:

1. Configure an SNS topic

2. Add your email address as a subscriber to the topic

3. Configure the topic policy to allow permissions from other resources

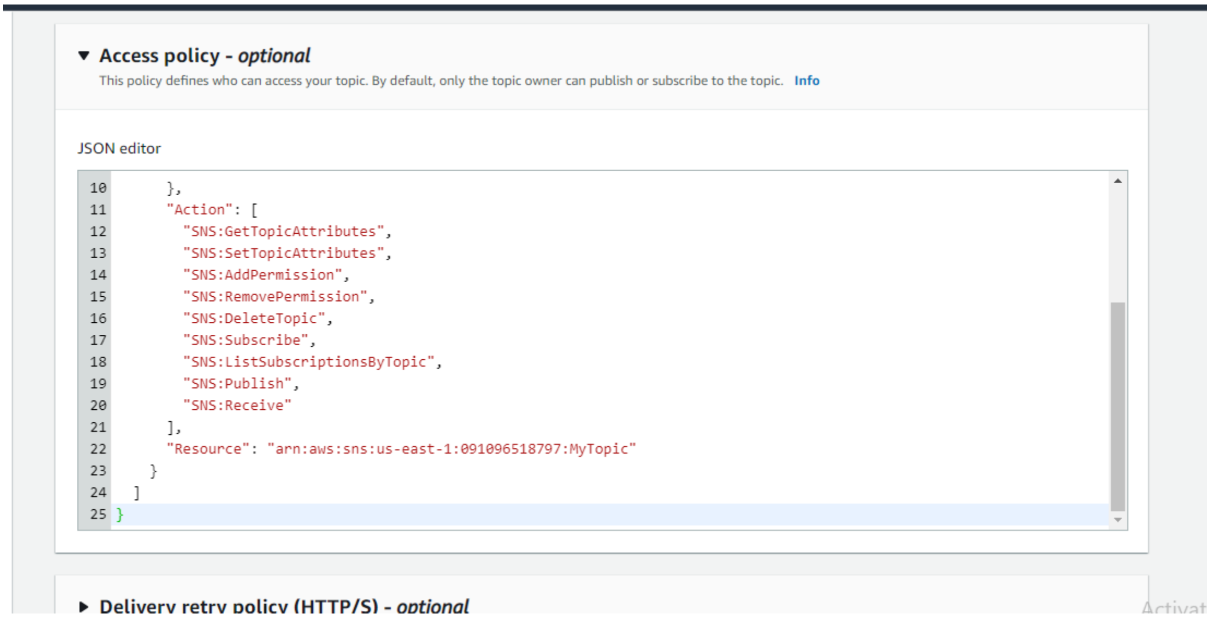
4. Configure an S3 event for the delete

5. Verify that your event notification worked.

Steps from 1 to 8 follow experiment no. 8

1. Login to the SNS dashboard.
2. Click on “Create Topic”.
3. Enter a topic name and click on “Create Topic”.
4. Click on the new topic and select “Create Subscription”.
5. Select the “Protocol” as “Email”.
6. Select the “Endpoint” as your email address.
7. Click “Create Subscription”.
8. Check your email and verify the subscription on the email from AWS.
9. Go to topic, Edit Topic and go to Access policy
10. Click on access policy and delete the following code.

```
"Condition": {
      "StringEquals": {
       "AWS:SourceOwner": "091096518797"
      }
}
```



▼ **Access policy - *optional***

This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. Info

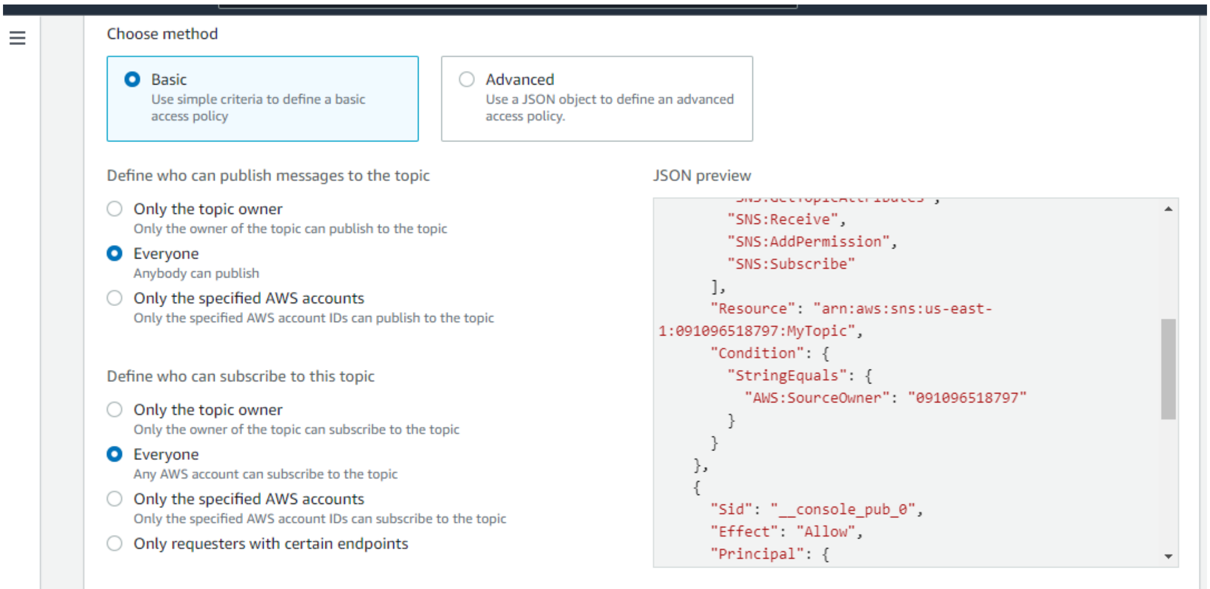
JSON editor

```
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
        "SNS:RemovePermission",
        "SNS:DeleteTopic",
        "SNS:Subscribe",
        "SNS:ListSubscriptionsByTopic",
        "SNS:Publish",
        "SNS:Receive"
      ],
      "Resource": "arn:aws:sns:us-east-1:091096518797:MyTopic"
    }
  ]
}
```

▶ **Delivery retry policy (HTTP/S) - *optional***

Or

You will get below window



Choose method

● Basic
Use simple criteria to define a basic access policy

○ Advanced
Use a JSON object to define an advanced access policy.

Define who can publish messages to the topic

○ Only the topic owner
Only the owner of the topic can publish to the topic

● Everyone
Anybody can publish

○ Only the specified AWS accounts
Only the specified AWS account IDs can publish to the topic

Define who can subscribe to this topic

○ Only the topic owner
Only the owner of the topic can subscribe to the topic

● Everyone
Any AWS account can subscribe to the topic

○ Only the specified AWS accounts
Only the specified AWS account IDs can subscribe to the topic

○ Only requesters with certain endpoints

JSON preview

```
        "SNS:GetTopicAttributes",
        "SNS:Receive",
        "SNS:AddPermission",
        "SNS:Subscribe"
      ],
      "Resource": "arn:aws:sns:us-east-
1:091096518797:MyTopic",
      "Condition": {
        "StringEquals": {
          "AWS:SourceOwner": "091096518797"
        }
      }
    },
    {
      "Sid": "__console_pub_0",
      "Effect": "Allow",
      "Principal": {
```

11. Click on “Save Policy”.

12. Switch to the S3 dashboard.

13. Create bucket and upload files. (Refer Experiment no. 1)

14. Click on a bucket and select “Events Notifications” from the “Properties” section.

15. In Event Notifications click on create event notifications.

16. Enter the following details:

    Event Name: S3DELETE

    Event Types: Select “All object delete event”

    Select Destination: SNS Topic

    SNS Topic: Enter the name of the topic you created in the SNS dashboard

    Click on “Save”.

17. Delete an object from your bucket and wait for an email to arrive that informs you of the file deletion.

Experiment No. 12

Your company wants to send a notification to the SNS topic when the EC2 instance Average CPU utilization exceeds its threshold value. Perform the demonstration.

1. Create a topic and subscription.
2. Launch an Ec2 Instance
3. Configure CloudWatch Alarm Metrics
4. Send a Message to SNS topic using CloudWatch Alarm
5. Create Topic and subscription (Refer Experiment no. 8)
6. Launch an EC2 Instance

For creating and launching instances refer experiment no. 6

7. Go to instances and click on add + alarm or you can create alarms from cloudwatch.
8. Create alarm.
9. Setup threshold for CPU utilization
10. Once the CPU Utilization exceeds 85% threshold value, the alarm status would change from **OK** to **ALARM**.
11. You will receive an Email. Verify the **SNS Notification.**