

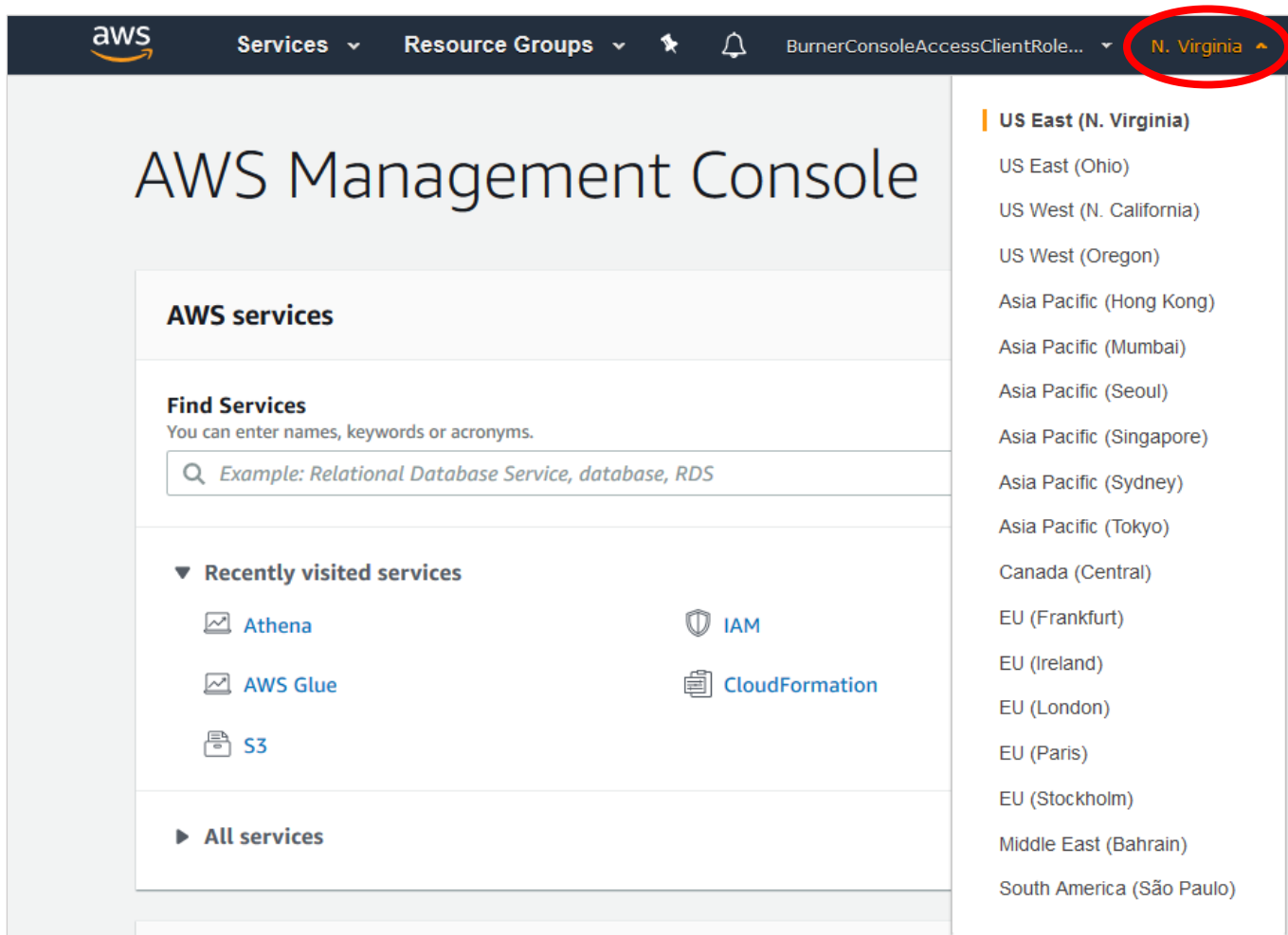


## Section 1

### Create an S3 bucket and subdirectories

In this section you will create a storage bucket and associated folders necessary to complete this lab using [Amazon S3](#), the AWS object storage service. These S3 folders will be used to hold the results returned from the query service [Amazon Athena](#), which will be generated by the queries you'll create in the next section. We'll delve more deeply into what Athena does when we get there. In this section you will also create an S3 folder to hold a CSV file that lists cities you could potentially use for the event. This CSV will be "joined," via Athena, to another CSV in an account that holds the temperature data.

- 1) Log on to the AWS console and change your region to N. Virginia.



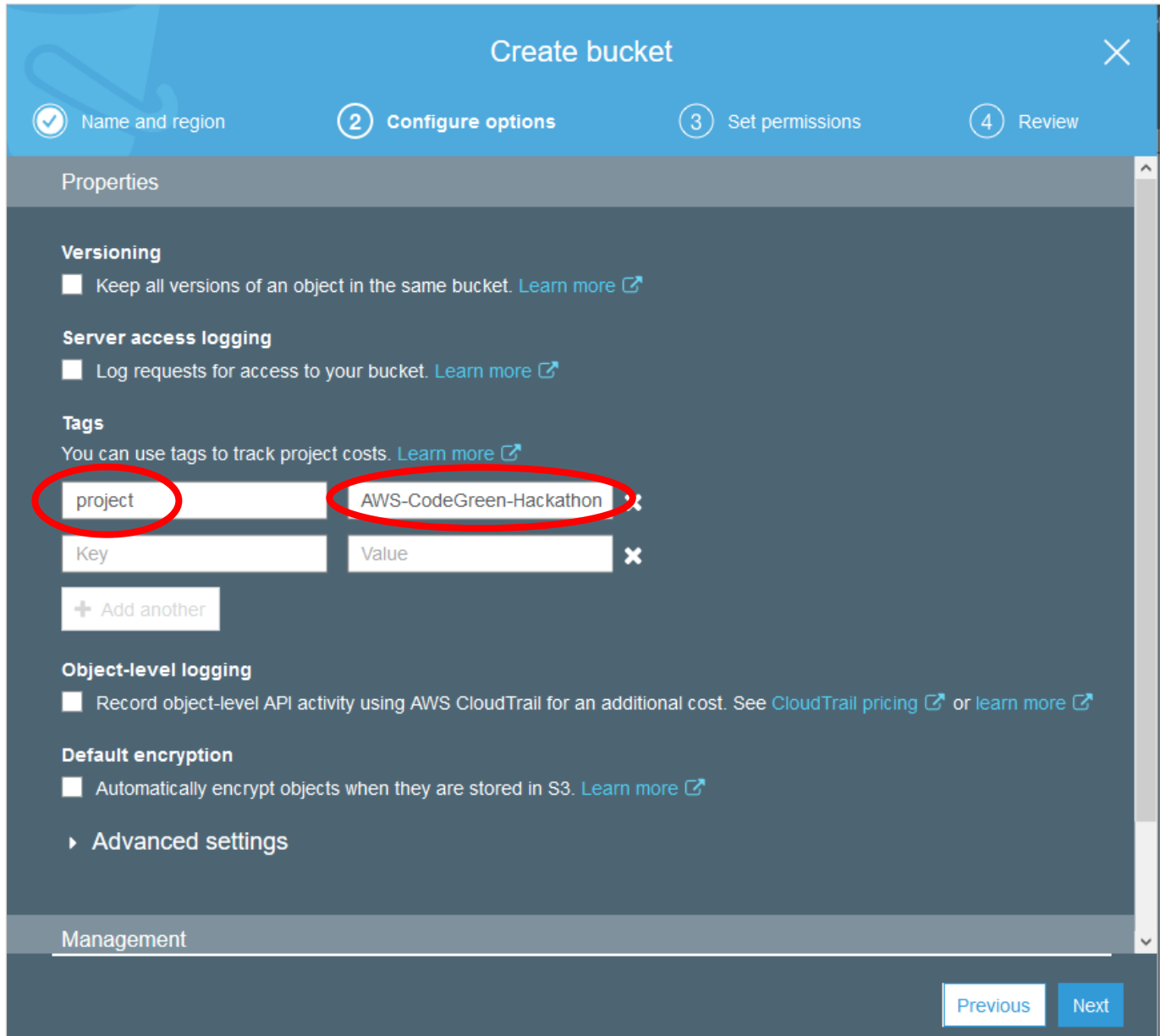
- 2) In the "Find Services" field search for S3 and navigate to the S3 dashboard, then click on "Create bucket". All S3 buckets must have a globally unique name and must comply with DNS



naming conventions—generally you should use lowercase letters and no underscores ([more information](#)). Select the region N. Virginia, enter a unique bucket name, and click “Next”.

The screenshot shows the 'Create bucket' wizard in the AWS Management Console. The title bar at the top says 'Create bucket' with a close button. Below the title bar are four steps: 1 Name and region, 2 Configure options, 3 Set permissions, and 4 Review. The first step, 'Name and region', is active. It contains a 'Bucket name' field with the text 'rcr-weatherbucket' and a 'Region' dropdown menu showing 'US East (N. Virginia)'. Both the text input and the dropdown are circled in red. Below these fields is a section titled 'Copy settings from an existing bucket' with a dropdown menu that says 'Select bucket (optional) 2 Buckets'. At the bottom of the form are three buttons: 'Create', 'Cancel', and 'Next'.

- 3) Add a tag with “project” as the key and “AWS-CodeGreen-Hackathon” as the value, and click “Next”.



**Create bucket**

1 Name and region    2 **Configure options**    3 Set permissions    4 Review

**Properties**

**Versioning**  
☐ Keep all versions of an object in the same bucket. [Learn more](#)

**Server access logging**  
☐ Log requests for access to your bucket. [Learn more](#)

**Tags**  
You can use tags to track project costs. [Learn more](#)

Key	Value
project	AWS-CodeGreen-Hackathon

[+ Add another](#)

**Object-level logging**  
☐ Record object-level API activity using AWS CloudTrail for an additional cost. See [CloudTrail pricing](#) or [learn more](#)

**Default encryption**  
☐ Automatically encrypt objects when they are stored in S3. [Learn more](#)

▸ **Advanced settings**

**Management**

[Previous](#) [Next](#)

- 4) You don't need this bucket to be publicly available, so accept the default ("Block *all* public access") and click "Next".

Create bucket

✓ Name and region

✓ Configure options

3 Set permissions

4 Review

Note: You can grant access to specific users after you create the bucket.

### Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block *all* public access. These settings apply only to this bucket. AWS recommends that you turn on Block *all* public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

The Block public access settings turned on at the account level affect public access to all buckets in the account. To determine which settings are on, check your Block public access (account settings).

☒ **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**

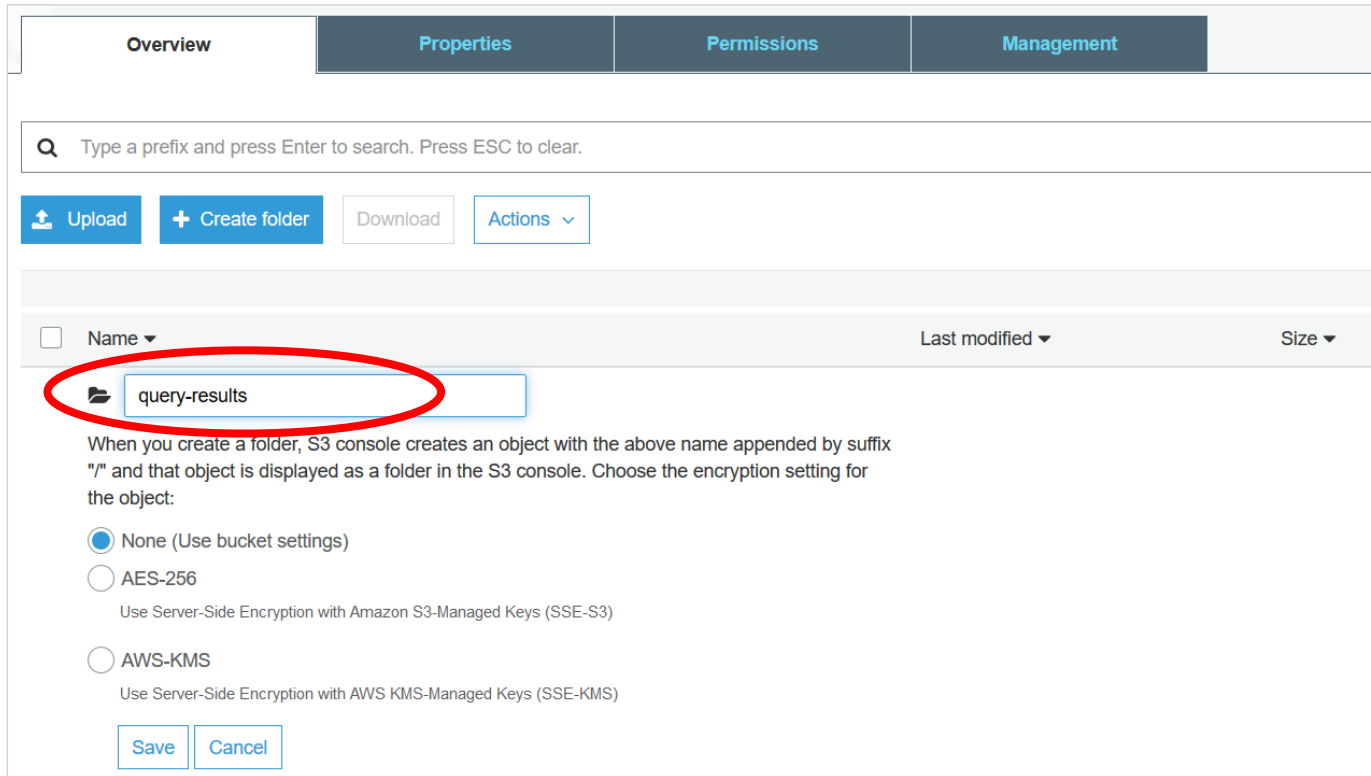
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.
- ☐ **Block public access to buckets and objects granted through *new* public bucket policies**

Previous

Next

- 5) Click "Create bucket".
- 6) Now you will see your bucket on the Amazon S3 dashboard, under Buckets.
- 7) Click on the bucket name you just created. It's time to create a few folders!
- 8) Click on "Create folder," then type "query-results" next to the folder icon. Accept the default encryption settings ("None") and click "Save".



Overview Properties Permissions Management

🔍 Type a prefix and press Enter to search. Press ESC to clear.

📁 Upload + Create folder Download Actions ▾

<input type="checkbox"/>	Name ▾	Last modified ▾	Size ▾
<input checked="" type="checkbox"/>	query-results		

When you create a folder, S3 console creates an object with the above name appended by suffix "/" and that object is displayed as a folder in the S3 console. Choose the encryption setting for the object:

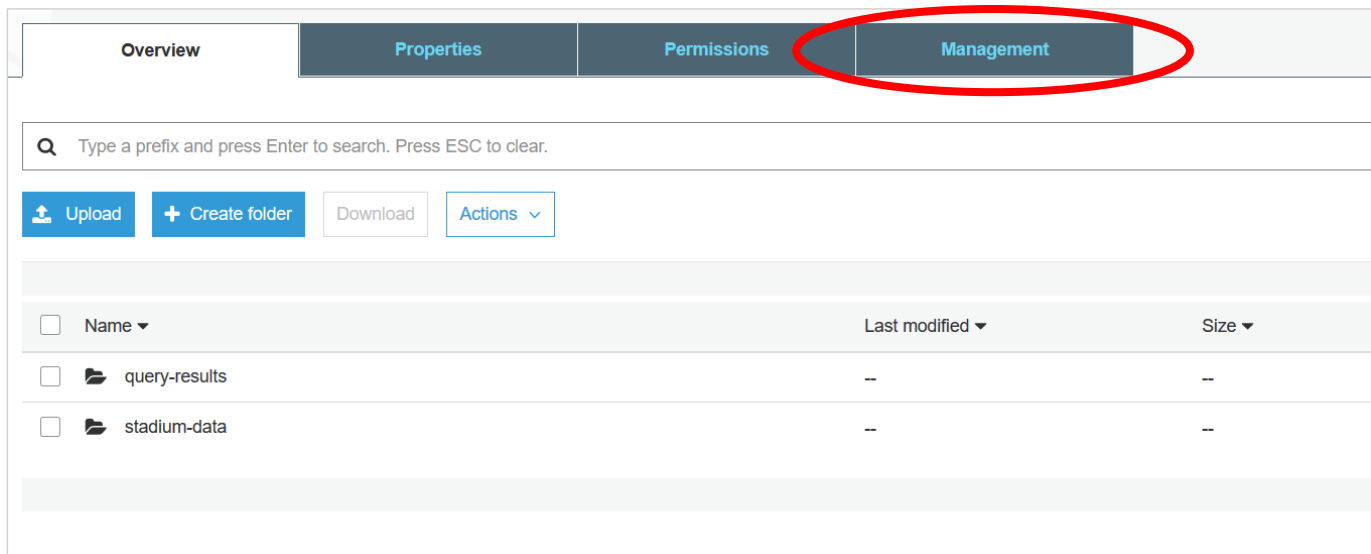
☒ None (Use bucket settings)

☐ AES-256  
Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

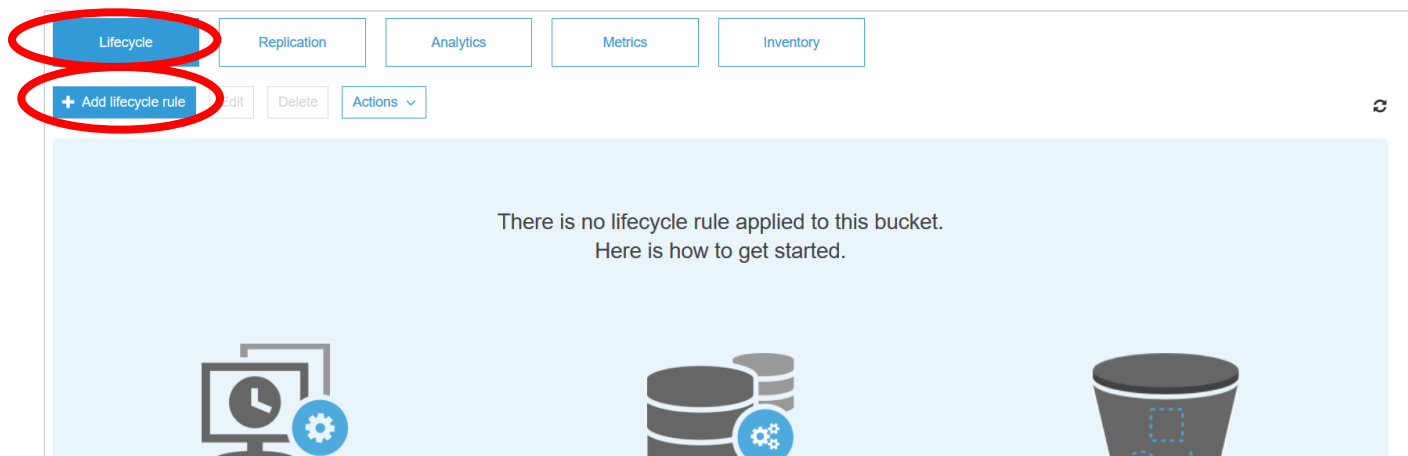
☐ AWS-KMS  
Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

Save Cancel

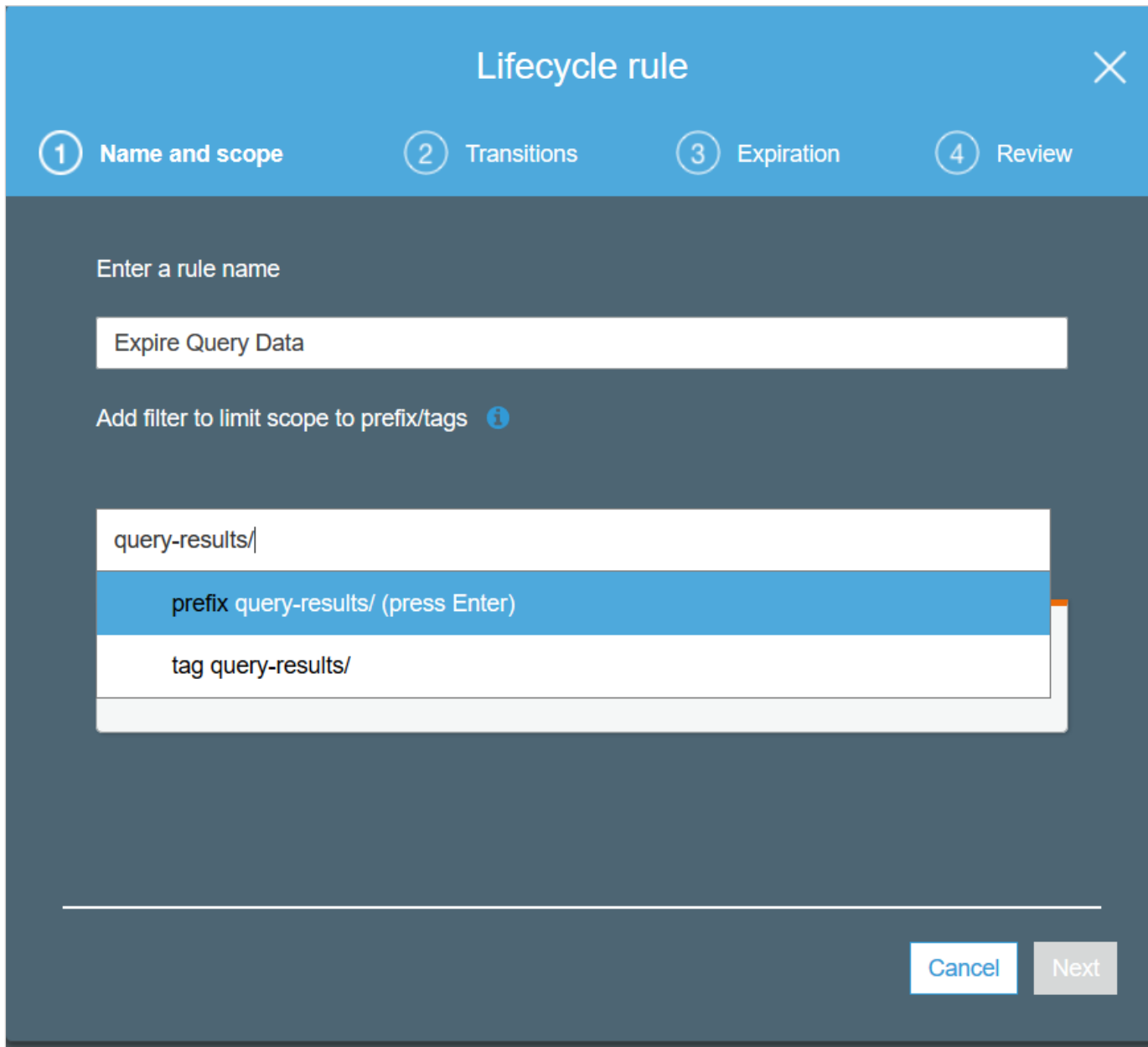
- 9) Repeat this process to create a folder named "stadium-data".
- 10) Using the AWS Console upload the [stadium\\_with\\_stations\\_global.csv](#) file to the stadium-data folder: click on the stadium-data folder, then click on the "Upload" button. Navigate to where you saved the file, select it to be uploaded, and click "Next". In the "Set permissions" section, accept the defaults and click "Next", and likewise in "Set properties" accept the default, "Standard", and click "Next". Finally, click "Upload". You should see the object you just uploaded in the folder when it is complete.
- 11) Click on the "Amazon S3" link above the Overview tab to go back to the S3 console. Then click on the bucket you created in this example.
- 12) Since the query-results folder will be used to temporarily store results from our Athena queries, you will want to create a [lifecycle policy](#) to delete these files when they are no longer necessary. Start by clicking on the "Management" tab.



13) With the Lifecycle tab highlighted, click the "Add lifecycle rule" button.



14) In the rule name field type "Expire Athena query results". In the prefix/tags field type the folder you created for the Athena query results followed by a slash ("query-results/"), select prefix, then click "Next".



**Lifecycle rule** ✕

① **Name and scope**    ② Transitions    ③ Expiration    ④ Review

Enter a rule name

Expire Query Data

Add filter to limit scope to prefix/tags ⓘ

query-results/

- prefix query-results/ (press Enter)
- tag query-results/

Cancel Next

- 15) On the “Transitions” pane leave the two boxes for “Current version” and “Previous version” unchecked (we are not moving files to other S3 storage classes), and click “Next”.
- 16) On the “Expiration” pane select the “Current version” checkbox, change the expiration value to 1 day, and click “Next”.
- 17) On the “Review” pane click “Save”.



---

In this session you created the S3 bucket that Athena will use to store query results, specified a folder to hold the mappings file of the cities to sensor locations, and created a lifecycle policy to delete items in your query-result/ folder.

You've now completed Section 1 of the workshop and can move on to the next section, "[Connecting Athena to the NOAA data repository](#)."