

AWS VPC Networking (Public & Private Subnets, NAT Gateway) + S3 Lifecycle Rules

Step 1 : Created VPC

The screenshot shows the AWS VPC console interface. A green success message at the top says "You successfully created vpc-0ad5785162d2d314f / MyVPC". The main card displays VPC details: VPC ID (vpc-0ad5785162d2d314f), State (Available), Block Public Access (Off), DNS hostnames (Disabled), and other configuration like Main network ACL, Default VPC, IPv4 CIDR (10.0.0.0/16), and Route 53 Resolver DNS Firewall rule groups. Below the main card, there are tabs for Resource map, CIDs, Flow logs, Tags, and Integrations. The Resource map section shows three cards: VPC (Your AWS virtual network), Subnets (0) (Subnets within this VPC), and Route tables (1) (Route network traffic). The left sidebar lists various VPC management options like Subnets, Route tables, Internet gateways, and Carrier gateways.

Step 2: Created subnets

The screenshot shows the AWS VPC console interface, specifically the Subnets section. A green success message at the top says "You have successfully created 1 subnet: subnet-00ec5bc3f1c59542c". The main card displays subnet details: Subnet ID (subnet-00ec5bc3f1c59542c), Name (Public-Subnet), State (Available), and VPC (vpc-0ad5785162d2d314f | MyVPC). Below the main card, there is a "Select a subnet" dropdown menu. The left sidebar lists various VPC management options like Subnets, Route tables, Internet gateways, and Carrier gateways.

Step 3: Turn ON auto-assign public IP

The screenshot shows the 'Edit subnet settings' page for a subnet named 'Public-Subnet'. In the 'Auto-assign IP settings' section, the 'Enable auto-assign public IPv4 address' checkbox is checked. Below it, a note states 'Option disabled because no customer owned pools found.' Other sections like 'Resource-based name (RBN) settings' and 'Hostname type' are also visible.

The screenshot shows the 'Subnets' page with a green success message: 'You have successfully changed subnet settings: Enable auto-assign public IPv4 address'. The subnet table lists one subnet: 'Public-Subnet' (Subnet ID: subnet-00ec5bc3f1c59542c, State: Available, VPC: vpc-0ad5785162d2d314f). The 'Details' tab is selected, showing details like Subnet ID, Subnet ARN, State, and VPC.

Step 4: Creating private subnets for the same vpc

The screenshot shows the 'Create subnet' wizard. In the 'Subnet settings' step, a single subnet named 'Private-Subnet' is being created. It is assigned to the 'United States (N. Virginia) / us-east-1a (us-east-1b)' availability zone. The IPv4 CIDR block is set to '10.0.0.0/16'. The IPv4 subnet CIDR block is set to '10.0.2.0/24', which provides 256 IP addresses. There are no tags added to this subnet.

Step 5: Final Created

The screenshot shows the 'Subnets' page with one subnet listed: 'Private-Subnet' with Subnet ID 'subnet-0528256b9a6192925'. The subnet is in the 'Available' state and is associated with the VPC 'vpc-0ad5785162d2d314f | MyV...'. The subnet has a CIDR block of '10.0.2.0/24' and 256 IP addresses.

Step 6: Create Internet Gateway (for Public Subnet)

The screenshot shows the 'Create internet gateway' wizard. A new internet gateway is being created with the name 'MyIGW'. A single tag 'MyIGW' is attached to the gateway. The 'Create internet gateway' button is visible at the bottom right.

Step 7: Created

The screenshot shows the AWS VPC Internet Gateways page. A green success message at the top states: "The following internet gateway was created: igw-03feddf0a06f1cb4a - MyIGW. You can now attach to a VPC to enable the VPC to communicate with the internet." Below this, the internet gateway details are shown: Internet gateway ID is igw-03feddf0a06f1cb4a, State is Detached, VPC ID is -, and Owner is 164242284028. There is one tag named "Name" with the value "MyIGW".

Step 8 : Now attaching Internet Gateway to VPC

The screenshot shows the "Attach to VPC" dialog box. It asks to attach an internet gateway to a VPC to enable communication with the internet. A search bar shows "vpc-0ad5785162d2d314f". At the bottom are "Cancel" and "Attach internet gateway" buttons.

Step 9: Successfully attached

The screenshot shows the AWS VPC Internet Gateways page again. A green success message at the top states: "Internet gateway igw-03feddf0a06f1cb4a successfully attached to vpc-0ad5785162d2d314f". Below this, the internet gateway details are shown: Internet gateway ID is igw-03feddf0a06f1cb4a, State is Attached, VPC ID is vpc-0ad5785162d2d314f (labeled "MyVPC"), and Owner is 164242284028. There is one tag named "Name" with the value "MyIGW".

Step 10: Creating Route Table for Public Subnet

The screenshot shows the 'Create route table' page in the AWS VPC console. In the 'Route table settings' section, a 'Name - optional' field contains 'Public-RT'. Under 'VPC', the dropdown shows 'vpc-0ad5785162d2d314f (MyVPC)'. In the 'Tags' section, there is one tag named 'Name' with value 'Public-RT'. At the bottom right are 'Cancel' and 'Create route table' buttons.

Step 11: Created Route table

The screenshot shows the 'Route tables' page in the AWS VPC console. A success message at the top states 'rtb-08e8853c525f0160d | Public-RT was created successfully.' The main area displays the details for 'rtb-08e8853c525f0160d / Public-RT'. The 'Details' section includes fields for Route table ID (rtb-08e8853c525f0160d), Main (No), VPC (vpc-0ad5785162d2d314f | MyVPC), and Owner ID (164242284028). Below this are tabs for 'Routes', 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'. The 'Routes' tab shows one route entry: Destination 10.0.0.0/16, Target local, Status Active, Propagated No, and Route Origin Create Route Table. There is also an 'Edit routes' button.

Step 12: Adding Internet Route:

The screenshot shows the AWS VPC Route Tables interface. In the 'Edit routes' section, a new route is being added for destination 10.0.0.0/16. The target is set to 'local' (Active status) and 'Internet Gateway' (igw-03feddf0a06f1cb4a). The 'Add route' button is visible at the bottom left, and 'Save changes' is highlighted at the bottom right.

- Added Successfully:

The screenshot shows the AWS VPC Route Tables interface with a success message: 'Updated routes for rtb-08e8853c525f0160d / Public-RT successfully'. The 'Details' tab is selected, showing the route table ID (rtb-08e8853c525f0160d), owner (vpc-0ad5785162d2d314f | MyVPC), and other details. The 'Routes' tab is selected, displaying two routes: one to 0.0.0.0/0 via igw-03feddf0a06f1cb4a (Active, Create Route) and another to 10.0.0.0/16 via local (Active, Create Route Table).

Step 13: Associating with Public Subnet:

The screenshot shows the 'Edit subnet associations' page in the AWS VPC console. At the top, the navigation path is VPC > Route tables > rtb-08e8853c525f0160d > Edit subnet associations. The main section is titled 'Available subnets (1/2)' and lists two subnets: 'Public-Subnet' and 'Private-Subnet'. The 'Public-Subnet' row is selected, indicated by a blue border. The 'Selected subnets' section contains the entry 'subnet-00ec5bc3f1c59542c / Public-Subnet'. At the bottom right are 'Cancel' and 'Save associations' buttons.

• Finally Associated

The screenshot shows the 'Route tables' page in the AWS VPC console. The route table 'rtb-08e8853c525f0160d / Public-RT' is selected. A green success message at the top states 'You have successfully updated subnet associations for rtb-08e8853c525f0160d / Public-RT.' The 'Details' section shows the route table ID, owner ID, and explicit subnet associations. The 'Routes' section displays two routes: one to 'igw-03feddf0a06f1cb4a' and another to 'local'. The left sidebar shows navigation links for Virtual private cloud, Route tables, Security, and PrivateLink and Lattice.

Step 14: Creating Route Table for Private Subnet

The screenshot shows the 'Create route table' page in the AWS VPC console. In the 'Route table settings' section, a name 'Private-RT' is entered. Under 'VPC', the 'MyVPC' VPC is selected. In the 'Tags' section, a single tag 'Name: Private-RT' is added. At the bottom right, the 'Create route table' button is highlighted.

Created

The screenshot shows the details of the newly created route table 'rtb-05afa769f08cc545e / Private-RT'. The 'Details' section shows the route table ID, VPC ID, and owner ID. The 'Routes' section displays one route entry: Destination 10.0.0.0/16, Target local, Status Active, Propagated No, and a 'Create Route Table' link. The left sidebar shows navigation links for Virtual private cloud, Security, and PrivateLink and Lattice.

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	Create Route Table

Step 15: Editing subnet Association

Route tables

- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections
- Route servers

▼ Security

- Network ACLs
- Security groups

▼ PrivateLink and Lattice

- Getting started
- Endpoints

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS Route Table Details page. The route table ID is rtb-05afa769f08cc545e. It is set as the Main route table (No) and owned by the user with ID 164242284028, associated with the VPC vpc-0ad5785162d2d314f named MyVPC. Under the 'Routes' tab, there is one route entry: Destination 10.0.0.0/16, Target local, Status Active, Propag... No, and Route Origin Create Route Table. There are also tabs for Subnet associations, Edge associations, Route propagation, and Tags.

The screenshot shows the 'Edit subnet associations' dialog box. At the top, it displays the account ID 1642-4228-4028 and the route table ID rtb-05afa769f08cc545e. The title is 'Edit subnet associations'. A sub-header says 'Change which subnets are associated with this route table.' Below this, the 'Available subnets (1/2)' section lists two subnets: 'Public-Subnet' and 'Private-Subnet'. The 'Private-Subnet' checkbox is selected. The 'Selected subnets' section contains the entry 'subnet-0528256b9a6192925 / Private-Subnet' with a close button (X). At the bottom are 'Cancel' and 'Save associations' buttons.

Name	Subnet ID	IPv4 CIDR	IPv6 C
Public-Subnet	subnet-00ec5bc3f1c59...	10.0.1.0/24	-
Private-Subnet	subnet-0528256b9a61...	10.0.2.0/24	-

Step 16: Launch EC2 Public and Private Subnets

The screenshot shows the AWS EC2 'Launch Instance' success page. At the top, a green banner displays the message 'Successfully initiated launch of instance (i-0cd25b0cac952fa0)'. Below the banner, there's a 'Next Steps' section with six items:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Buttons: 'Create billing alerts' and 'Learn more'.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Buttons: 'Connect to instance' and 'Learn more'.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Buttons: 'Connect an RDS database' and 'Learn more'.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Buttons: 'Create EBS snapshot policy' and 'Learn more'.

At the bottom of the page, there are links for CloudShell and Feedback, and a footer with copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

Showing both subnets

The screenshot shows the AWS EC2 Instances page. On the left, a navigation sidebar lists categories like EC2, Instances, Images, and Elastic Block Store. The Instances category is expanded, showing sub-options such as Instances, Instance Types, Launch Templates, and Capacity Reservations. The main content area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Private-EC2	i-0bc5058842edf9e57	Running	t3.micro	Initializing	View alarms +
Public-EC2	i-0cd25b0cac952fa0	Running	t2.micro	Initializing	View alarms +

Below the table, a 'Select an instance' dropdown menu is open, listing the two instances: 'Private-EC2' and 'Public-EC2'. The page includes standard AWS navigation elements like CloudShell and Feedback at the bottom, and a footer with copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

Step 18: Creating NAT Gateway (for private EC2 internet)

The screenshot shows the 'Create NAT gateway' settings page. It includes fields for 'Name' (MyNAT), 'Subnet' (selected as 'subnet-01808763f41eb3dca (Public-Subnet)'), 'Connectivity type' (set to 'Public'), and 'Elastic IP allocation ID' (eipalloc-018b9ee8b622e43a1). A green banner at the top indicates an elastic IP has been allocated. The 'Additional settings' section is collapsed.

- Created:

The screenshot shows the 'nat-0f211d692e9e8bf30 / MyNAT' details page. It displays the NAT gateway ID (nat-0f211d692e9e8bf30), ARN (arn:aws:ec2:us-east-1:68064372990:natgateway/nat-0f211d692e9e8bf30), and VPC (vpc-07faeeb3b33281fa6 / MyVPC). The connectivity type is set to 'Public'. The state is 'Pending'. The primary public IPv4 address is listed as '10.0.1.134'. The primary private IPv4 address is listed as '10.0.1.134'. The created date is 'Tuesday, November 18, 2025 at 14:31:23 GMT+5:30'. The secondary IPv4 addresses section shows no available addresses.

Step 19 :Adding Route in Private Route Table

The screenshot shows the AWS VPC Route Tables page. A new route is being added to a private route table. The destination is 0.0.0.0/0, and the target is set to 'local'. The status is 'Active'. The propagation setting is 'No', and the route origin is 'CreateRouteTable'. There is also a second row for a route to 10.0.0.16 with target 'local'.

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	local	Active	No	CreateRouteTable
10.0.0.16	local	Active	No	CreateRouteTable

Add route | **Cancel** | **Preview** | **Save changes**

- Finally Added:

The screenshot shows the AWS VPC Route Tables page after the routes have been successfully added. A green notification bar at the top indicates 'Updated routes for rtb-05388494bf042f5e2 / Private-RT successfully'. The main view shows the details of the route table, including its ID, VPC, and associations. The 'Routes' tab is selected, displaying two routes: one to 0.0.0.0/0 via a NAT gateway and one to 10.0.0.16 via 'local'.

rtb-05388494bf042f5e2 / Private-RT

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	nat-0f211d692e9e8bf...	Active	No	Create Route
10.0.0.16	local	Active	No	Create Route Table

Step 20: Testing INTERNET From PRIVATE EC2

The screenshot shows the AWS EC2 Instance Connect interface. At the top, it displays the instance ID: i-0dcdb25b0cac952fa0 (Public-EC2). Below this, under 'Connection type', the 'Public IPv4 address' option is selected, showing the IP 54.147.166.31. The 'Username' field contains 'ec2-user'. A note at the bottom states: 'Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom right are 'Cancel' and 'Connect' buttons.

Step 21: After typing the command

Sudo dnf update -y

Output is :

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-1-171 ~]$ sudo dnf update -y
Amazon Linux 2023 Kernel Livepatch repository      [===
264 kB/s | 29 kB    00:00      Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Tue Nov 18 09:05:18 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-171 ~]$ ssh ec2-user@10.0.1.171
The authenticity of host '10.0.1.171 (10.0.1.171)' can't be established.
ED25519 key fingerprint is SHA256:UWunL/19FREDXbinE9PR8medywTbitfGUL20liYWKAE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.171' (ED25519) to the list of known hosts.
[ec2-user@ip-10-0-1-171: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-1-171 ~]$ sudo dnf update -y
Last metadata expiration check: 0:04:08 ago on Tue Nov 18 09:05:18 2025.
Dependencies resolved.
Nothing to do.
Complete!
```

i-0dcdb25b0cac952fa0 (Public-EC2)

PublicIPs: 54.147.166.31 PrivateIPs: 10.0.1.171

After checking with the another command ping google.com

```
aws | Search [Alt+S] Account ID: 6806-4372-7990
United States (N. Virginia) v
voclabs/user3790785=anishakumari23@lpu.in

64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=57 ttl=106 time=2.35 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=58 ttl=106 time=2.47 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=59 ttl=106 time=2.26 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=60 ttl=106 time=2.25 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=61 ttl=106 time=2.23 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=62 ttl=106 time=2.23 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=63 ttl=106 time=2.24 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=64 ttl=106 time=2.26 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=65 ttl=106 time=2.48 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=66 ttl=106 time=2.25 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=67 ttl=106 time=2.55 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=68 ttl=106 time=2.24 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=69 ttl=106 time=2.55 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=70 ttl=106 time=2.23 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=71 ttl=106 time=2.63 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=72 ttl=106 time=2.28 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=73 ttl=106 time=2.22 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=74 ttl=106 time=2.23 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=75 ttl=106 time=2.23 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=76 ttl=106 time=2.25 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=77 ttl=106 time=2.24 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=78 ttl=106 time=2.61 ms
64 bytes from yuiadtq-in-f101.le100.net (192.178.218.101): icmp_seq=79 ttl=106 time=2.24 ms
^C
--- google.com ping statistics ---
79 packets transmitted, 79 received, 0% packet loss, time 78137ms
rtt min/avg/max/mdev = 2.188/2.368/3.336/0.202 ms
[ec2-user@ip-10-0-1-171 ~]$
```

i-0dcdb5b0cac952fa0 (Public-EC2)

PublicIPs: 54.147.166.31 PrivateIPs: 10.0.1.171

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

S3 Lifecycle Rules

Step 1: Creating S3 Bucket

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region: US East (N. Virginia) us-east-1

Bucket type: Info

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name: Info
anisha-logs-bucket

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn more](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

Choose bucket
Format: s3://bucket/prefix

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS S3 buckets page. At the top, there is a green banner with the message "Successfully created bucket 'anisha-logs-bucket'". Below the banner, there are two tabs: "General purpose buckets" (selected) and "All AWS Regions". On the left, under "General purpose buckets (1)", there is a table with one row for "anisha-logs-bucket". The table columns are "Name", "AWS Region", and "Creation date". The "Name" column shows "anisha-logs-bucket", the "AWS Region" column shows "US East (N. Virginia) us-east-1", and the "Creation date" column shows "November 18, 2025, 14:45:15 (UTC+05:30)". There are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". To the right of the table, there are two boxes: "Account snapshot" and "External access summary - new". Both boxes have "Updated daily" status indicators and "View dashboard" buttons. The "Account snapshot" box also includes a note about Storage Lens providing visibility into storage usage and activity trends.

Step 2: Creating rule

The screenshot shows the "Create lifecycle rule" page. The URL is "Amazon S3 > Buckets > anisha-logs-bucket > Lifecycle configuration > Create lifecycle rule". The page has a section titled "Lifecycle rule configuration" with a "Lifecycle rule name" input field containing "move-and-delete-logs". Below it is a note: "Up to 255 characters". A "Choose a rule scope" section contains two options: "Limit the scope of this rule using one or more filters" (unchecked) and "Apply to all objects in the bucket" (checked). A warning message in a box says: "If you want the rule to apply to specific objects, you must use a filter to identify those objects. Choose "Limit the scope of this rule using one or more filters". [Learn more](#)". A checkbox below the message is checked, stating: "I acknowledge that this rule will apply to all objects in the bucket". The "Lifecycle rule actions" section allows selecting actions: "Transition current versions of objects between storage classes" (checked) and "Transition noncurrent versions of objects between storage classes" (unchecked). The checked action has a note: "This action will move current versions." The bottom of the page includes standard AWS navigation links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Creation of Transition

- o Transition 1 — Move to S3 Standard-IA (after 30 days)
- o Transition 2 — Move to Glacier (after 90 days)
- o Add Expiration Rule (Delete after 180 days)

Screenshot of the AWS S3 Lifecycle Configuration page for the 'anisha-logs-bucket'. The 'Create lifecycle rule' section is active.

Transition current versions of objects between storage classes

Choose transitions to move current versions of objects between storage classes based on your use case scenario and performance access requirements. These transitions start from when the objects are created and are consecutively applied. [Learn more](#)

Choose storage class transitions

Storage Class	Days after object creation	Action
Standard-IA	30	Remove
Glacier Flexible Retrieval (formerly Glacier)	90	Remove

[Add transition](#)

Expire current versions of objects

For version-enabled buckets, Amazon S3 adds a delete marker and the current version of an object is retained as a noncurrent version. For non-versioned buckets, Amazon S3 permanently removes the object. [Learn more](#)

Days after object creation

180

Delete expired object delete markers or incomplete multipart uploads

Expired object delete markers

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step : Final Output

Screenshot of the AWS S3 Lifecycle Configuration review page for the 'anisha-logs-bucket'.

Review transition and expiration actions

Current version actions

Day	Actions
0	Objects uploaded
30	Objects move to Standard-IA
90	Objects move to Glacier Flexible Retrieval (formerly Glacier)
180	Objects expire

Noncurrent versions actions

Day	Actions
0	No actions defined.

[Cancel](#) [Create rule](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Your Lifecycle Rule Is Now Configured!

After few steps rule is created :

The screenshot shows the AWS S3 Lifecycle configuration page for the 'anisha-logs-bucket'. A green success message at the top states: 'The rule "move-and-delete-logs" has been successfully added and the lifecycle configuration has been updated. It may take some time for the configuration to be updated. Refresh the lifecycle rules list if changes to the configuration aren't displayed.' Below this, the 'Lifecycle configuration' section is visible, with a note about managing object storage cost effectively. Under 'Default minimum object size for transitions', it says 'All storage classes 128K'. The 'Lifecycle rules (1)' section shows a single rule named 'move-and-delete-logs' which is enabled and applies to the 'Entire bucket'. The rule action is 'Transition to Standard-'.

Lifecycle configuration

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.

Default minimum object size for transitions
All storage classes 128K

Lifecycle rules (1)

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. [Learn more](#)

Lifecycle rule ...	Status	Scope	Current versio...	Noncurrent ve...	Expired object...	Incomplete m...
move-and-delete-logs	Enabled	Entire bucket	Transition to Standard-	-	-	-

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences