Jellyfin Media Server on AWS

Project Overview

This project creates a Jellyfin media server with scalable storage hosted on AWS. It combines multiple AWS services to reliably stream via web browser access and flexible media management.

What You'll Build

- **Jellyfin Server** - A media server running on AWS EC2
- **Secure Access** - HTTPS-enabled access via Application Load Balancer
- **Scalable Storage** - EBS volumes for high-throughput media storage
- **Content Management** - S3 bucket integration for easy media uploads

Architecture Overview

This solution uses these AWS microservices:

- **VPC** provides secure networking with public and private subnets
- **EC2** hosts the Jellyfin media server application
- **EBS** provides high-performance storage for media content
- **S3** stores your media files for easy uploading and management
- **ALB** provides HTTPS access and could enable future scaling
- **IAM Roles** allow secure connections between services

Prerequisites

Software Requirements

- Web browser to access AWS Console and Jellyfin interface
- SSH client (built into browser for EC2)
- Media files to stream (videos, music, etc.)

Cost

- EC2 t2.micro: ~$8.50/month (free tier eligible for 12 months)
- EBS SC1 storage: ~$0.025/GB-month (SC1 because we don’t need as much throughput for small amount of people)
- S3 storage: ~$0.023/GB-month
- Application Load Balancer: ~$16.20/month

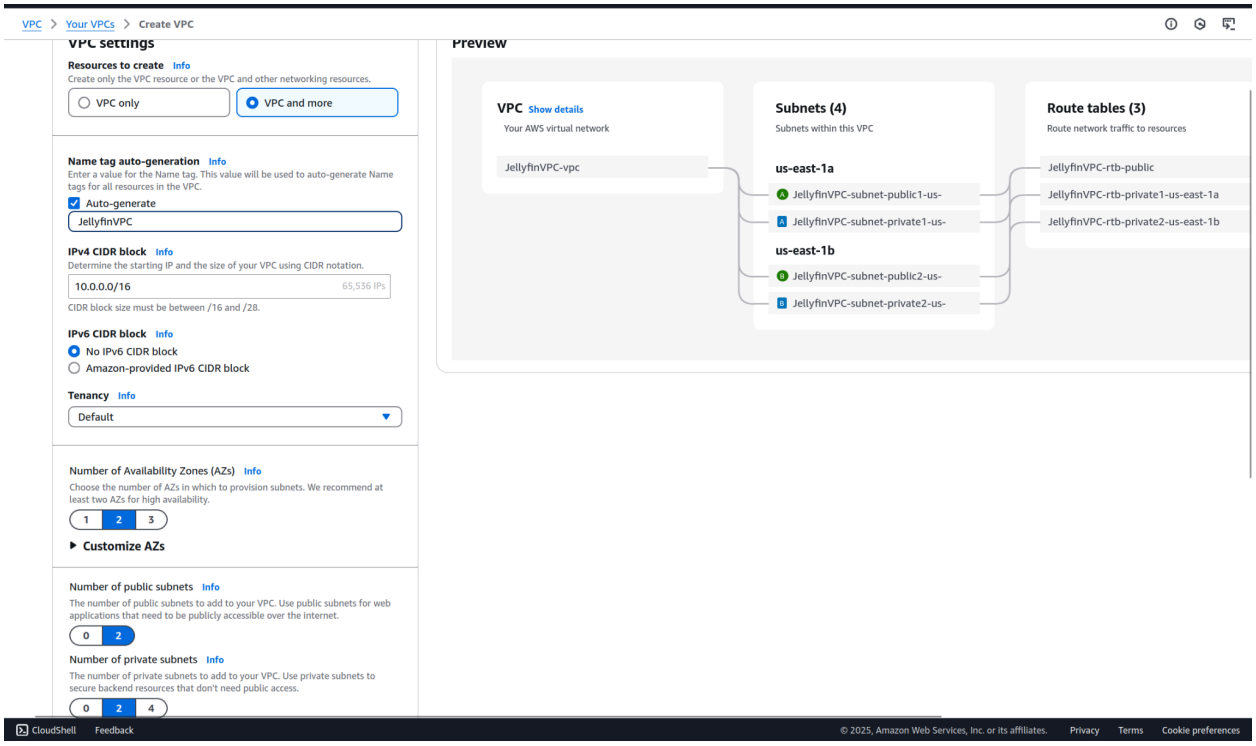
Knowledge

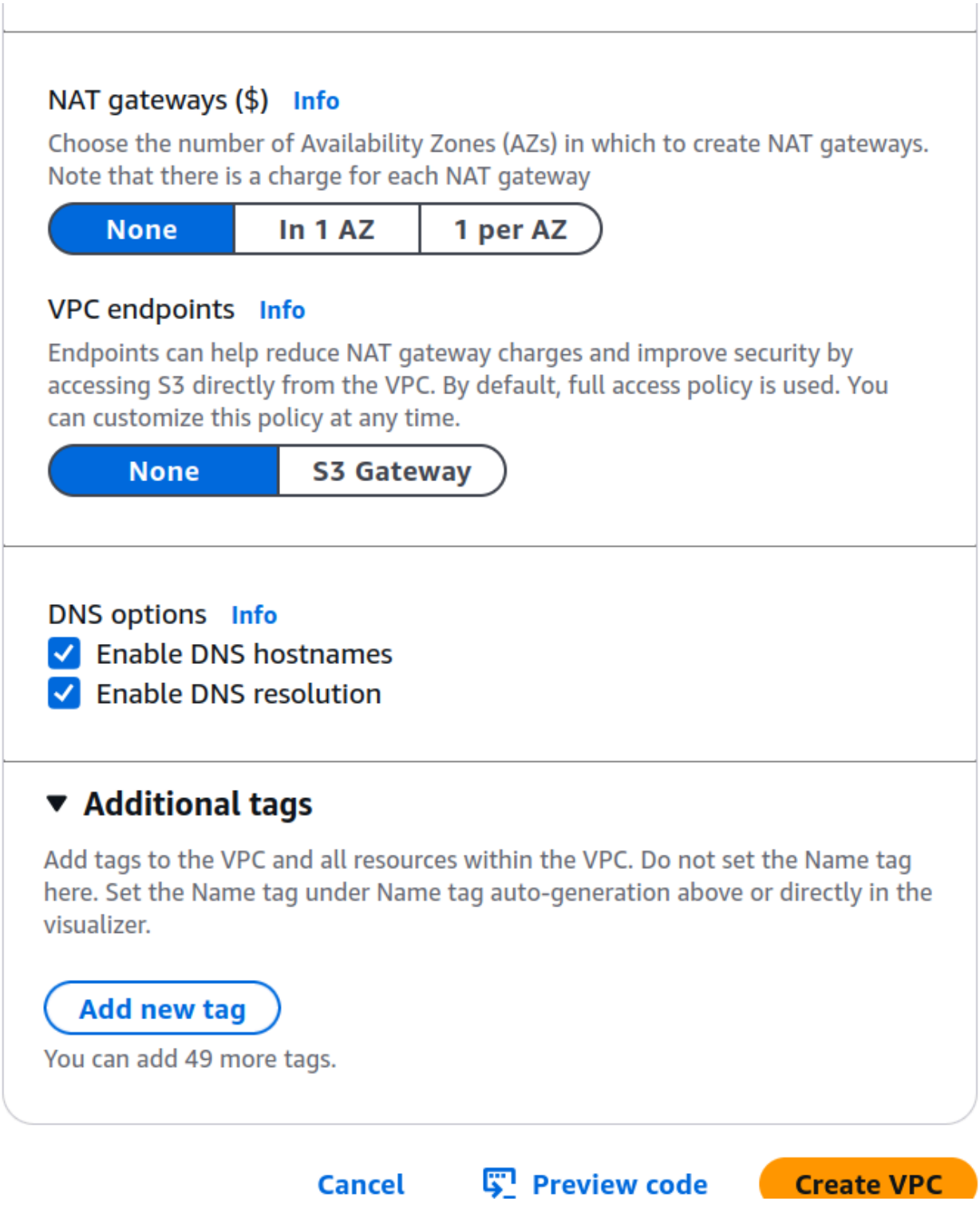
- Basic understanding of AWS services
- Basic Linux command line knowledge
- Familiarity with editing text files in terminal

Project Start

Create VPC

Create a VPC with 2 public subnets and a private subnet. The reason we are creating two public subnets is for the ALB for later that requires two availability zones (AZ).





Create EC2 Instance

We are naming it EC2Image because we will use this to make images (copies) later.

Name and tags Info

Name

JellyfinEC2Image

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | My AMIs | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible

ami-084568db4383264d4 (64-bit (x86)) / ami-0c4e709339fa8521a (64-bit (Arm))

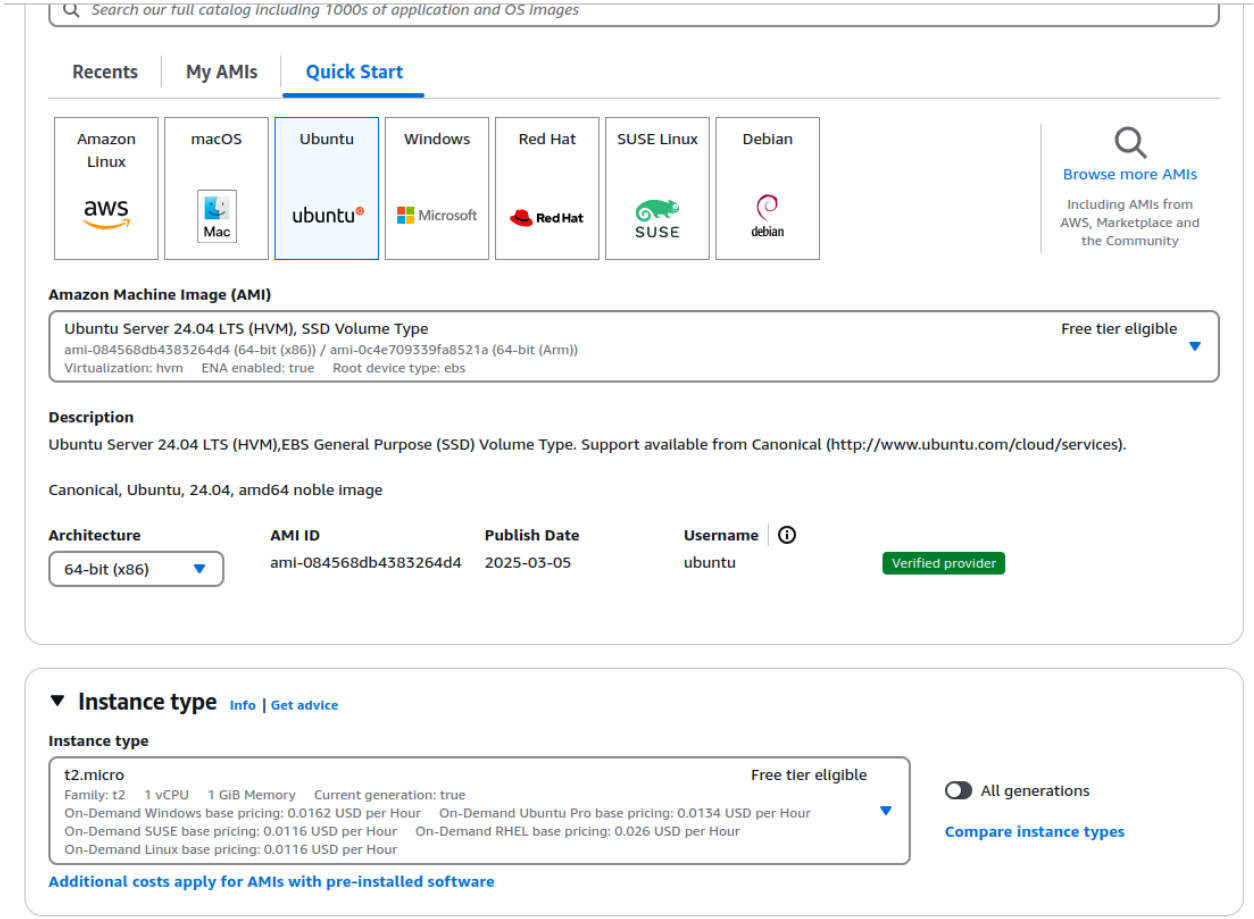
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

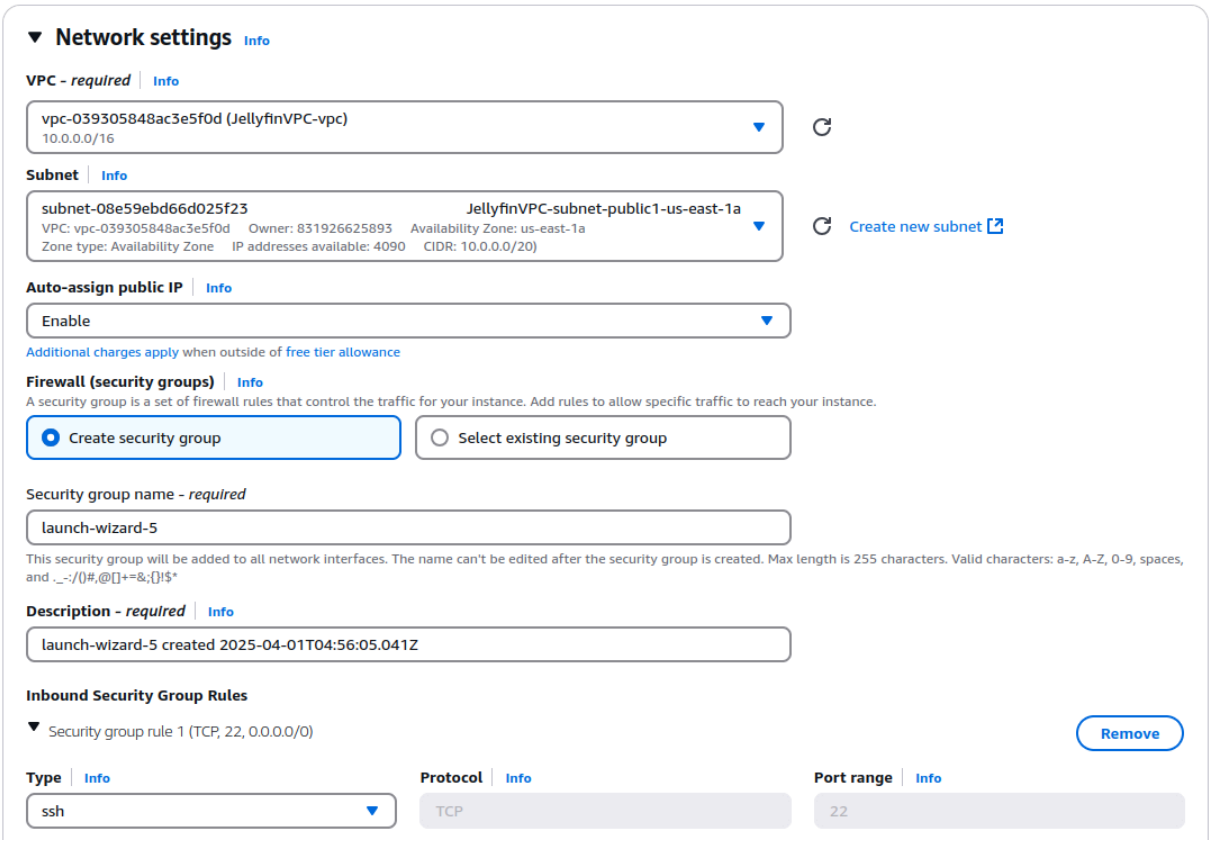
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

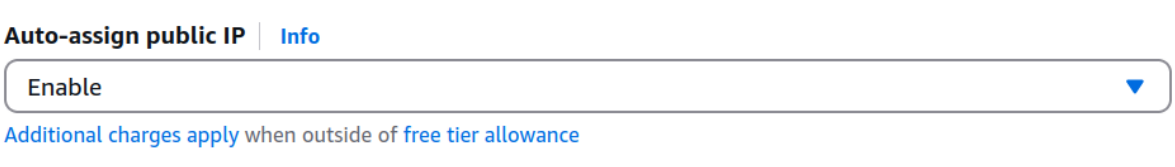
Architecture	AMI ID	Publish Date	Username	
64-bit (x86)	ami-084568db4383264d4	2025-03-05	ubuntu	Verified provider



Configure the network to use the VPC we just made. Do not enable http access yet.

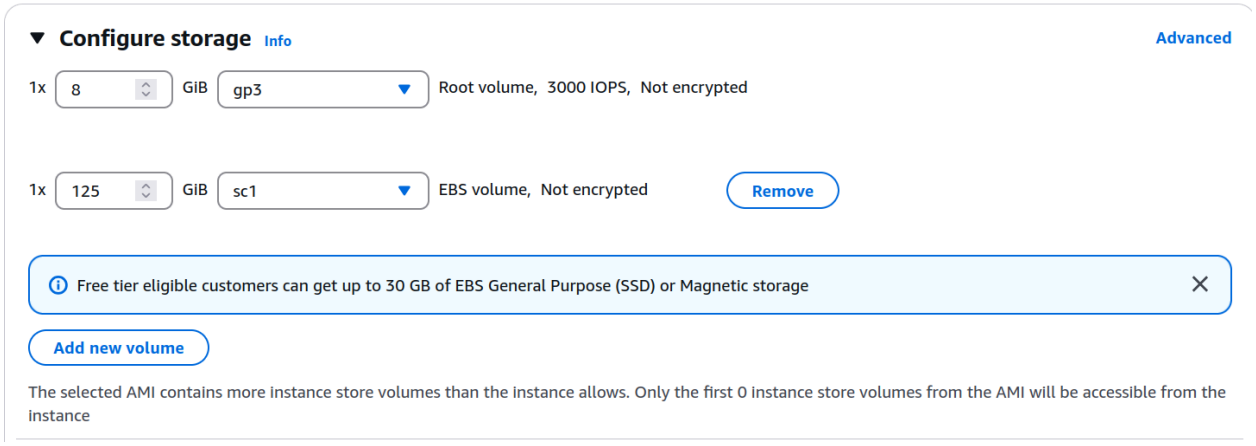


We’re going to need a public ip to connect to the EC2 for now. So enable this.



Configuring EBS Storage

EBS storage is a storage device that can mount to EC2 for high throughput operations, such as video streaming. We’re using SC1 for cheaper(but still fast enough) usage.



Setting up the EC2 Server

Connect to the EC2 instance using the browser connect.

Now let's download jellyfin and install it.

```
curl https://repo.jellyfin.org/install-debuntu.sh | sudo bash
```

Let it run through the process

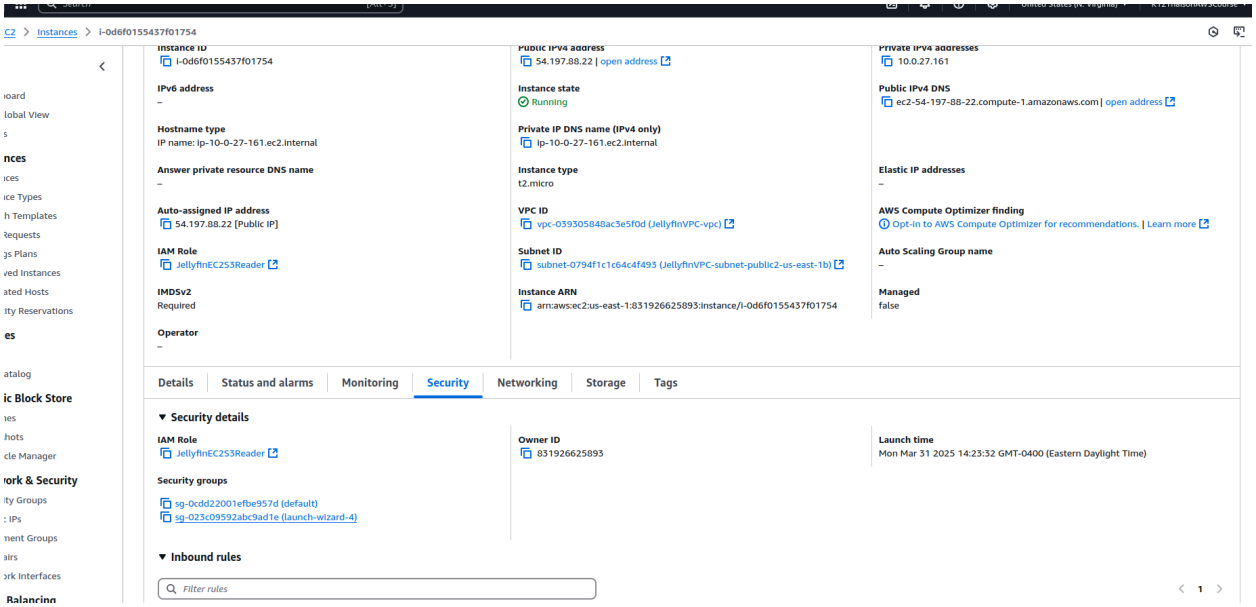
Attach the EBS

```
sudo mkfs -t ext4 /dev/xvdb
sudo mkdir /content
sudo mount /dev/xvdb /content
sudo mkdir /content/Movies
sudo chown -R jellyfin:jellyfin /content
sudo chmod 755 /content
echo "/dev/xvdb /content//Movies ext4 defaults,nofail 0 2" | sudo tee -a
/etc/fstab
```

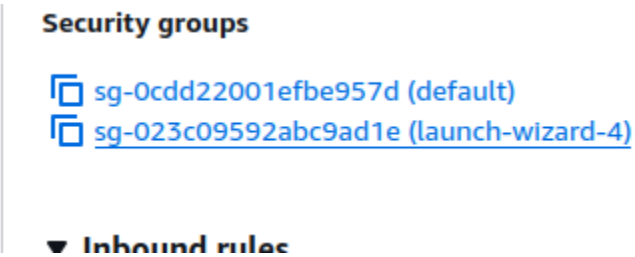
```
sudo systemctl restart jellyfin
```

Testing the EC2 server

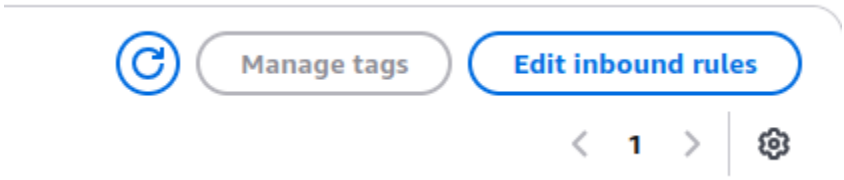
Let’s quickly test our server. Even though its public, we need to enable access to it in the security group



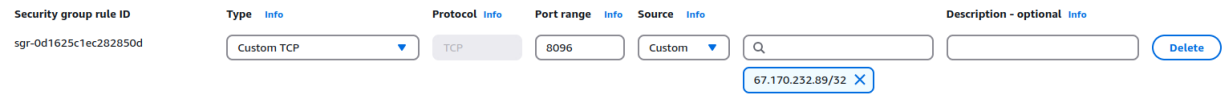
Click on launch wizard security group



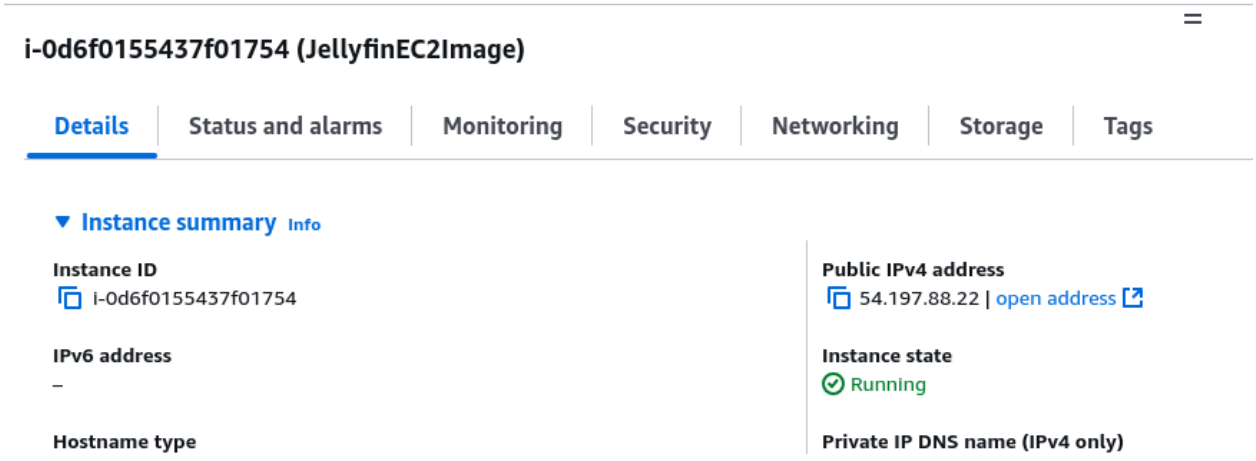
Edit the inbound rules



Add a TCP rule for 8096 for YOUR IP specifically.



Now connect to it with its public ip



Will have to use http, so replace the https with http

Do **NOT** go through the set up wizard yet. You are on http! It's not safe.

Setting up HTTPS -> Create a target group

On the EC2 Homepage, go to target groups and create one for instances. Name it something fitting.

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

- (•) Instances
  - Supports load balancing to instances within a specific VPC.
  - Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.
- ( ) IP addresses
  - Supports load balancing to VPC and on-premises resources.
  - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
  - Offers flexibility with microservice based architectures, simplifying inter-application communication.
  - Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- ( ) Lambda function
  - Facilitates routing to a single Lambda function.
  - Accessible to Application Load Balancers only.
- ( ) Application Load Balancer
  - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
  - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

JellyFinTargets

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

We're going to be using http(not https) protocol to communicate within the VPC.

( ) Application Load Balancer
  - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
  - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

JellyFinTargets

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation is created. This choice cannot be changed after creation

HTTP ▼ | 80 | 1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

- (•) IPv4
  Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.
- ( ) IPv6
  Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

VPC

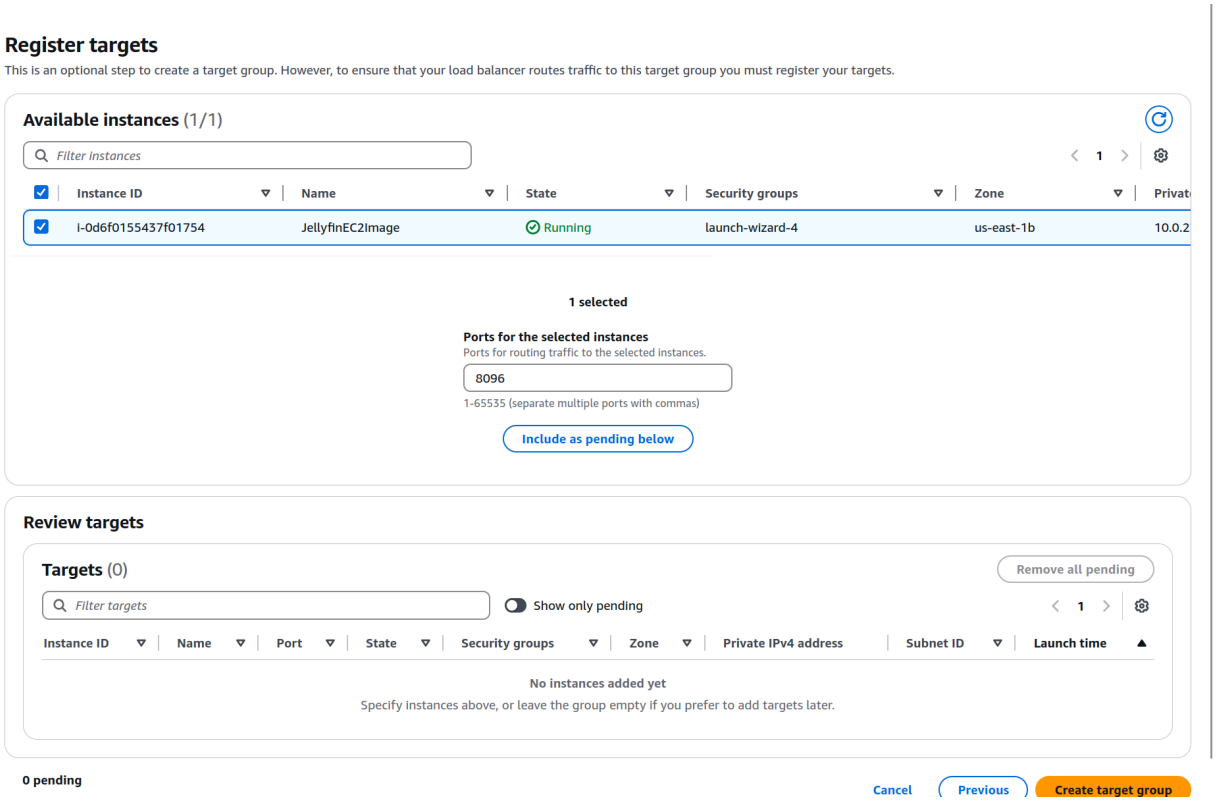
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

JellyfinVPC-vpc
vpc-039305848ac3e5f0d
IPv4 VPC CIDR: 10.0.0.0/16 ▼

Protocol version

- (•) HTTP1
  Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
- ( ) HTTP2
  Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
- ( ) gRPC
  Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Now target the 8096 port on the Jellyfin EC2. This is the port jellyfin will start on. Press include as a pending below, then create the target group.

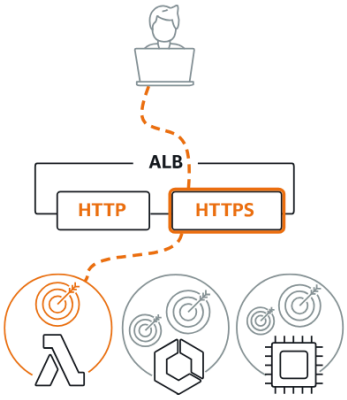


Create Application Load Balancer

We'll be using a public application load balancer has the public facing access for our EC2s

Load balancer types

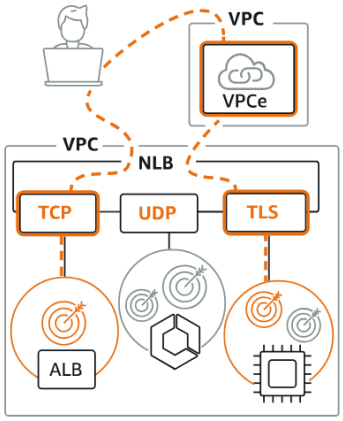
Application Load Balancer Info



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Network Load Balancer Info



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer Info

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

▶ **Classic Load Balancer** - *previous generation*

Create the ALB with these settings

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

JellyfinALB

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme | Info

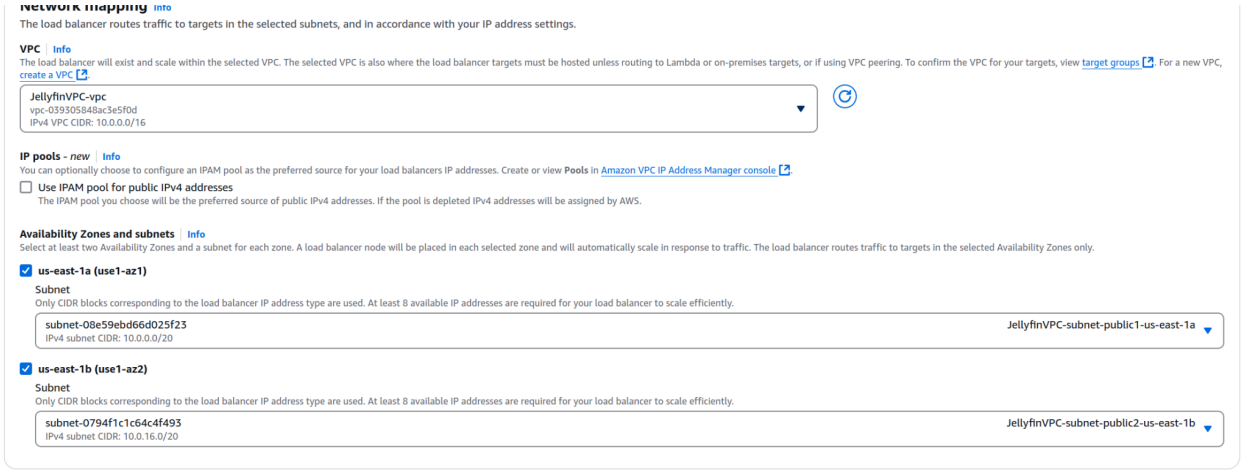
Scheme can't be changed after the load balancer is created.

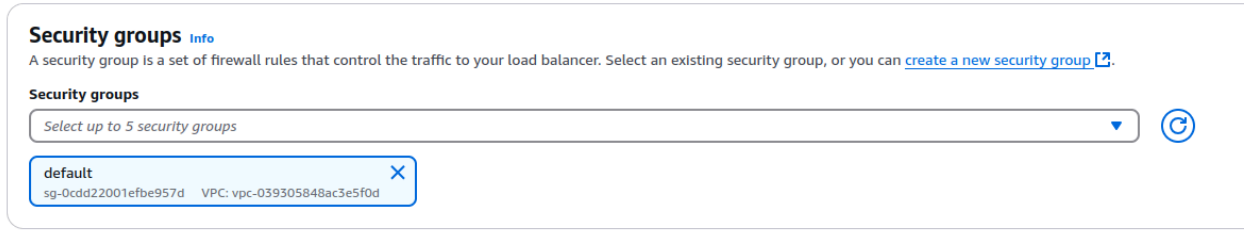
- (●) Internet-facing
  - Serves internet-facing traffic.
  - Has public IP addresses.
  - DNS name is publicly resolvable.
  - Requires a public subnet.
- ( ) Internal
  - Serves internal traffic.
  - Has private IP addresses.
  - DNS name is publicly resolvable.
  - Compatible with the **IPv4** and **Dualstack** IP address types.

Load balancer IP address type | Info

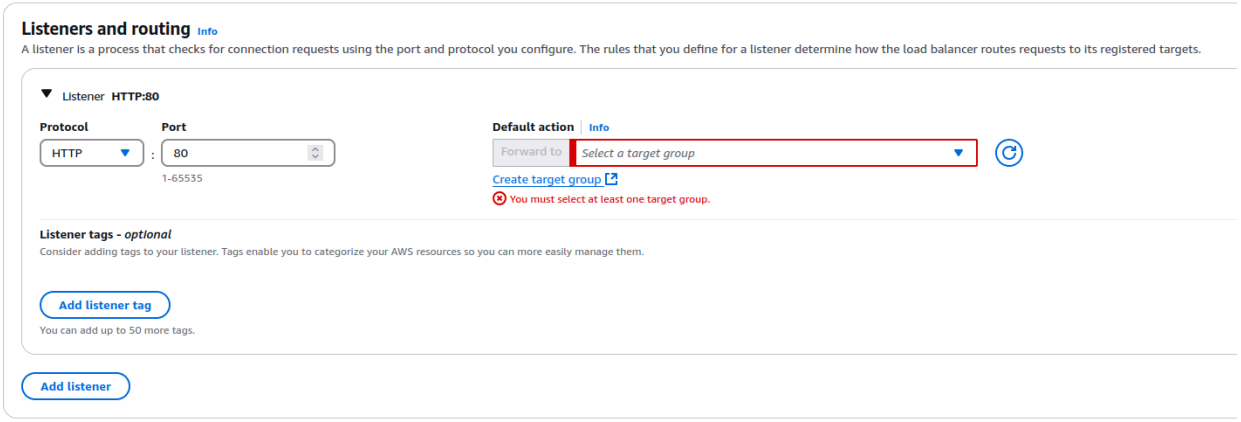
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

- (●) IPv4
  Includes only IPv4 addresses.
- ( ) Dualstack
  Includes IPv4 and IPv6 addresses.
- ( ) Dualstack without public IPv4
  Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.





Set the target as the previous target group we had earlier.



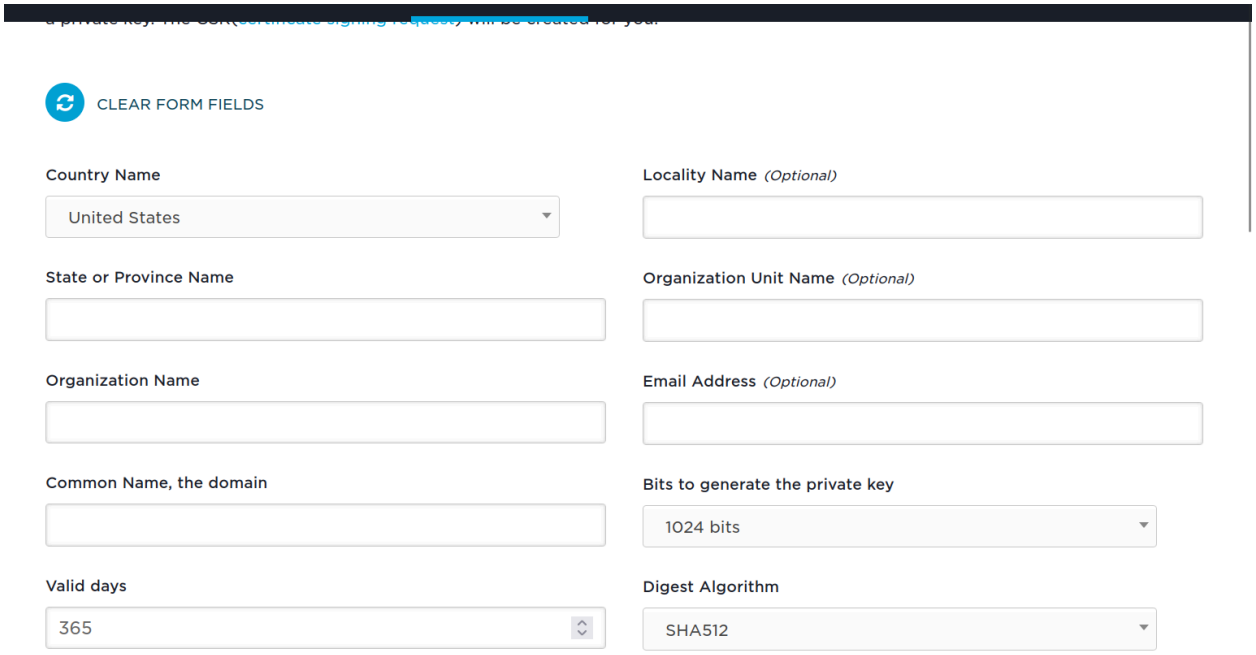
Create the ALB

Creating a self signed certificate

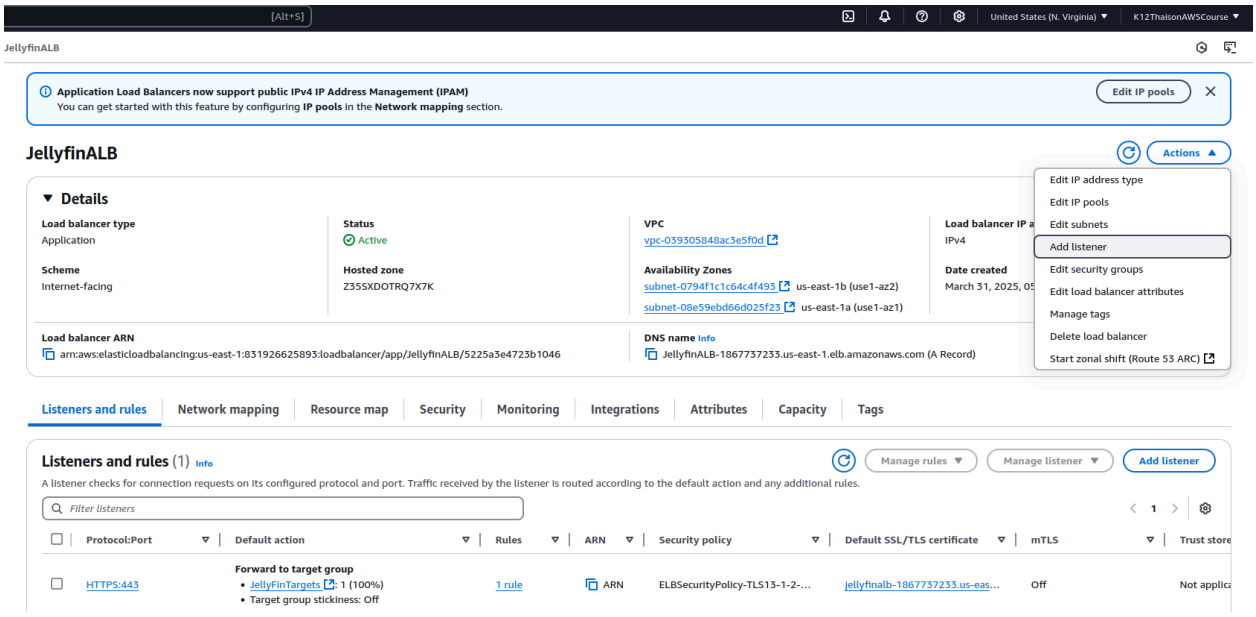
Create a self signed certificate here
https://www.samltool.com/self_signed_certs.php

This will allow us to use https on our server.

Use the ALB DNS for the Common Name.



Add an https listener to our ALB.



Listener details: HTTPS:443

A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

Listener configuration

The listener will be identified by the protocol and port.

Protocol
Used for connections from clients to the load balancer.
HTTPS

Port
The port on which the load balancer is listening for connections.
443
1-65535

Default actions | Info

The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Authentication | Info
Authentication requires IPv4 connectivity to authentication endpoints. Learn more

☐ Use OpenID or Amazon Cognito
Include authentication using either OpenID Connect (OIDC) or Amazon Cognito.

Routing actions

◉ Forward to target groups ○ Redirect to URL ○ Return fixed response

Forward to target group | Info
Choose a target group and specify routing weight or Create target group.

Target group		Weight	Percent
JellyFinTargets Target type: Instance, IPv4	HTTP	1 0-999	100%

Add target group
You can add up to 4 more target groups.

Target group stickiness | Info
Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to bind a user's session to a specific target, turn on the Target Group attribute Stickiness.

☐ Turn on target group stickiness

Secure listener settings | Info

These settings will apply to all of your secure listeners. Once created, you can manage these settings per listener.

Security policy | Info

Your load balancer uses a Secure Socket Layer (SSL) negotiation configuration called a security policy to manage SSL connections with clients. Compare security policies

Security category
All security policies

Policy name
ELBSecurityPolicy-TLS13-1-2-2021-06 (recommended)

Default SSL/TLS server certificate

The certificate used if a client connects without SNI protocol, or if there are no matching certificates. You can source this certificate from AWS Certificate Manager (ACM), Amazon Identity and Access Management (IAM), or import a cert to your listener certificate list.

Certificate source

○ From ACM ○ From IAM ◉ Import certificate

Certificate import destination
Import your certificate to either AWS Certificate Manager (ACM) or Amazon Identity and Access Management (IAM). It will be imported and made the default SSL/TLS server certificate for this load balancer's secure listeners.

◉ Import to ACM - *recommended*
Prerequisites apply. Learn more

○ Import to IAM
Use IAM as a certificate manager only if you must support HTTPS connections in a Region not supported by ACM. Learn more

Certificate private key

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDXRyXP+TvD4Ub2
ZUvV1IfJmisR7dQCQrRUKVULogFCC1vCOS1ih3Yx2A08JEFvyodbOxC/Dv3jDJCr
```

Certificate body

```
-----BEGIN CERTIFICATE-----
MIIDKzCCAhMCFHykVKdOkB1dvZXKtqnmGmyapKaxMA0GCSqGSIb3DQEBCwUAMFIx
CzAJBgNVBAYTAlVTMREwDwYDVQQIDAhOZXcgWW9yazERMA8GA1UEBwwIQnJvb2ts
```

Certificate chain - *optional*

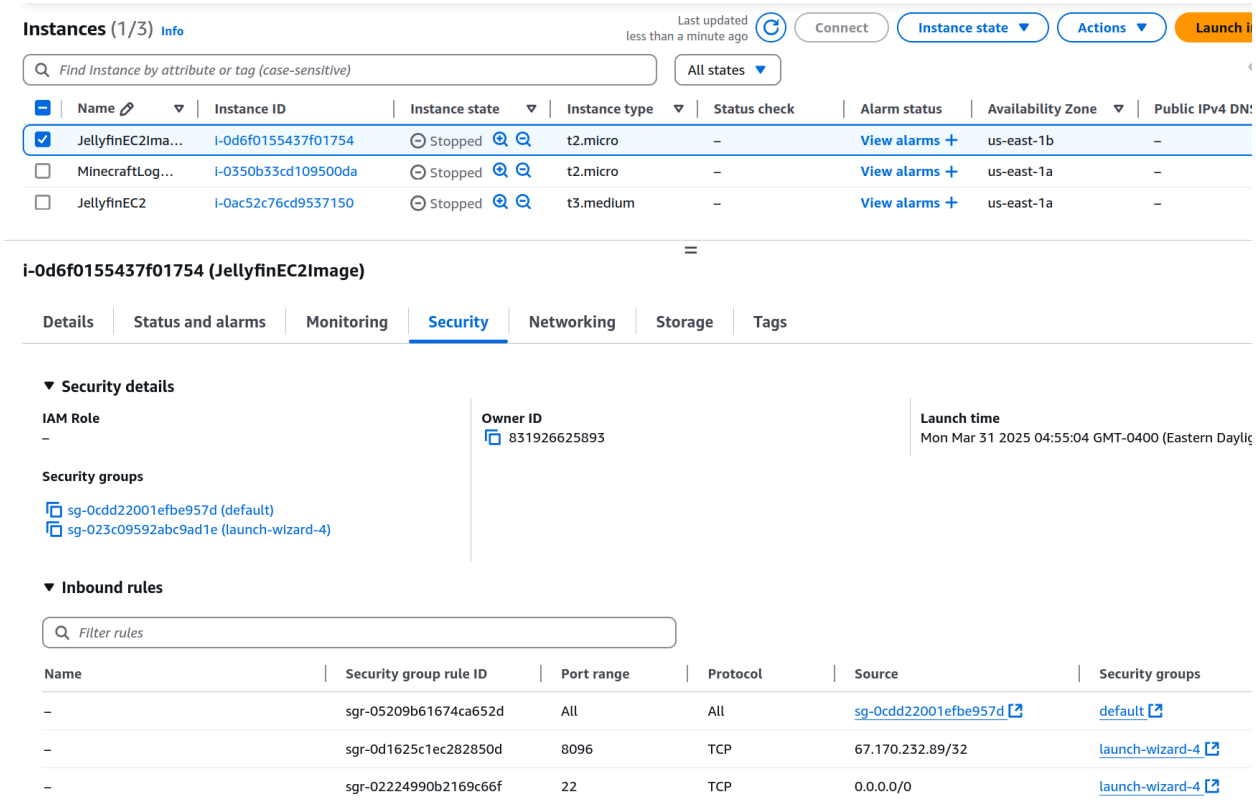
```
Paste the PEM-encoded certificate chain here
```

Client certificate handling | Info

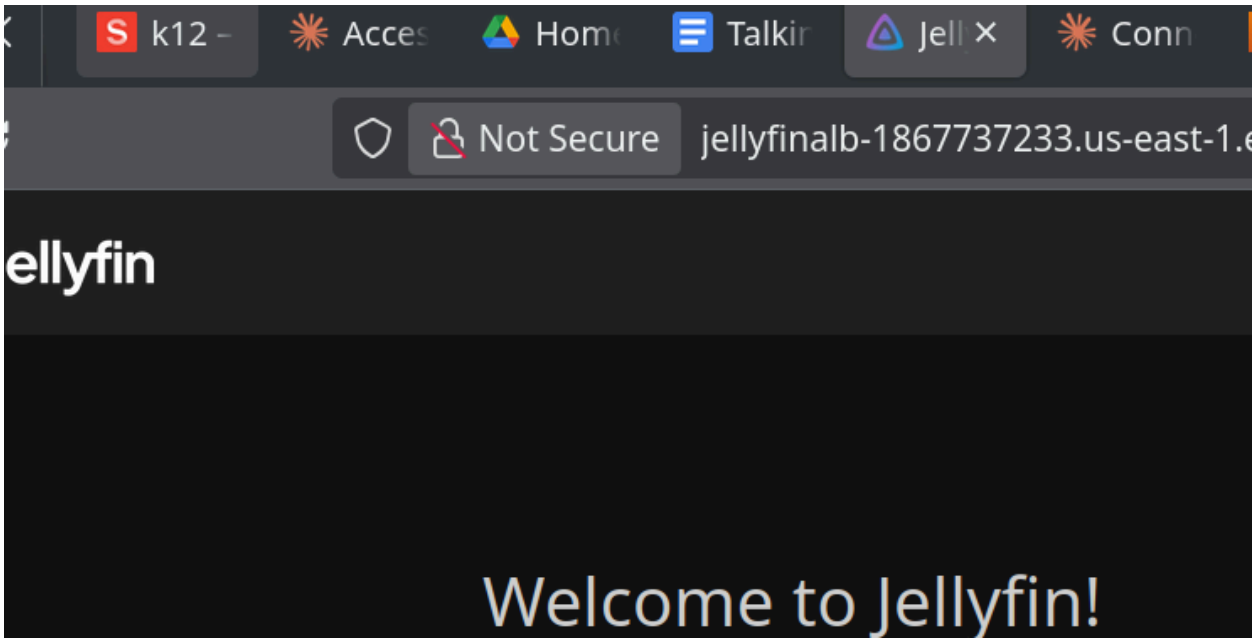
Client certificates are used to make authenticated requests to remote servers. Learn more

☐ **Mutual authentication (mTLS)**
Mutual TLS (Transport Layer Security) authentication offers two-way peer authentication. It adds a layer of security over TLS and allows your services to verify the client that's making the connection.

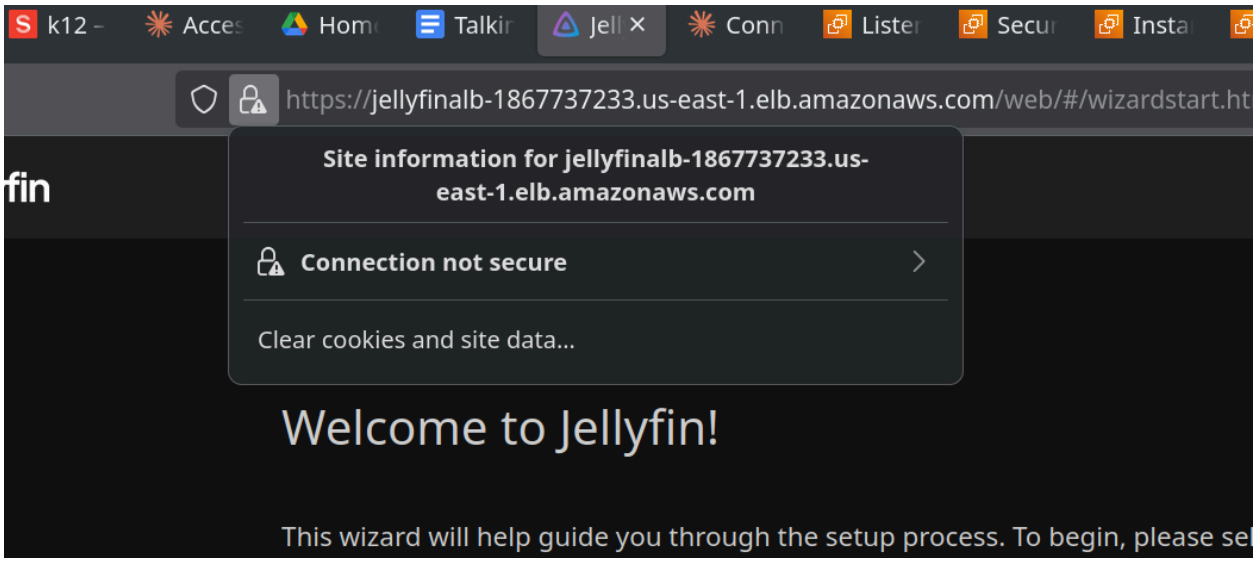
Add the default security group to the EC2, so that the ALB can send data to the EC2. This is because you can send data to other services in the same security group.



Now try connecting to the ALB DNS. Make sure we use https:// by putting that in front of the dns. If it shows not secure, it's http.

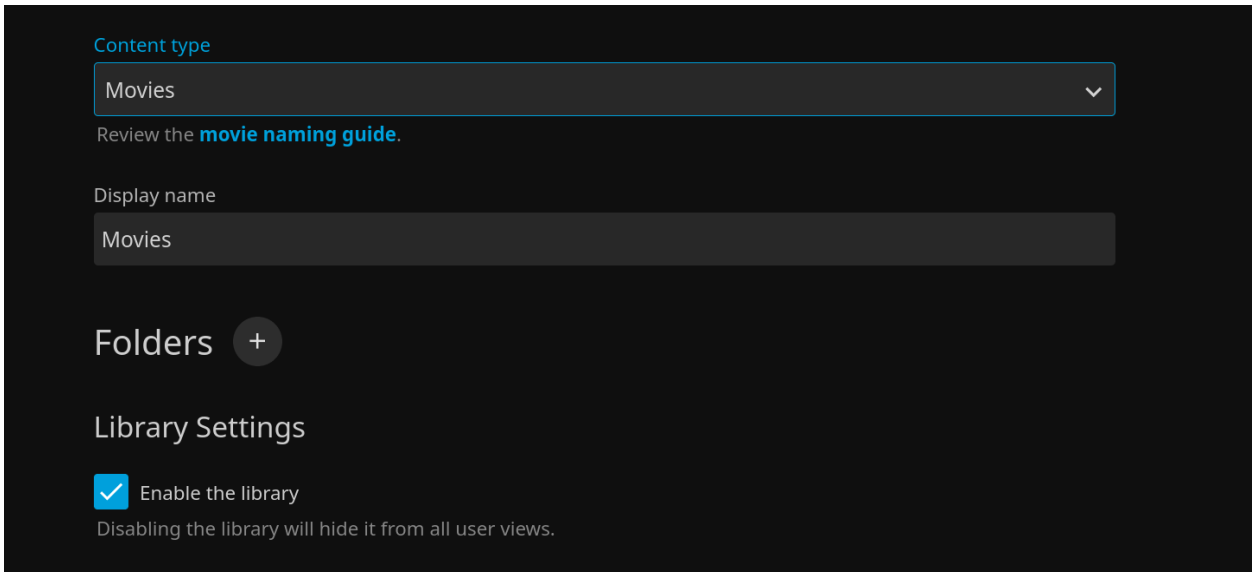


Example of https. Still says connection not secure though (because self signed cert)

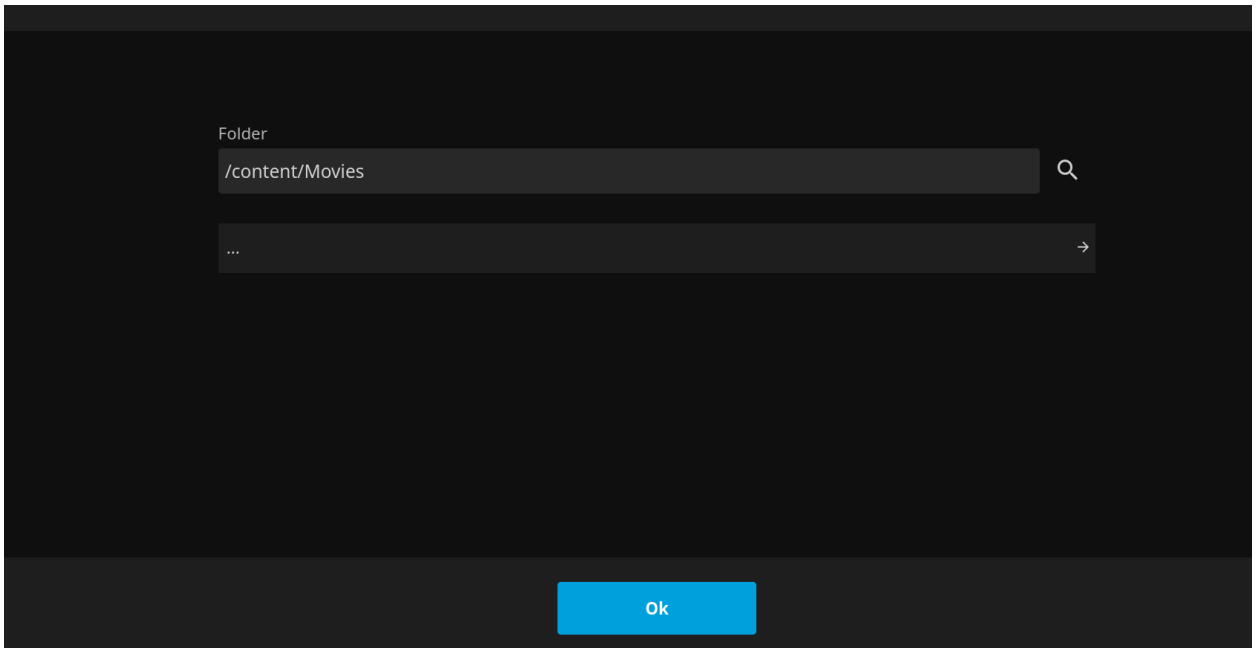


Now go through the process with https setup. Set up the admin credentials (Remember this!)
Set up a movie folder when prompted to create a library.

Change content type to movies. Press plus button on Folders



Put in this path

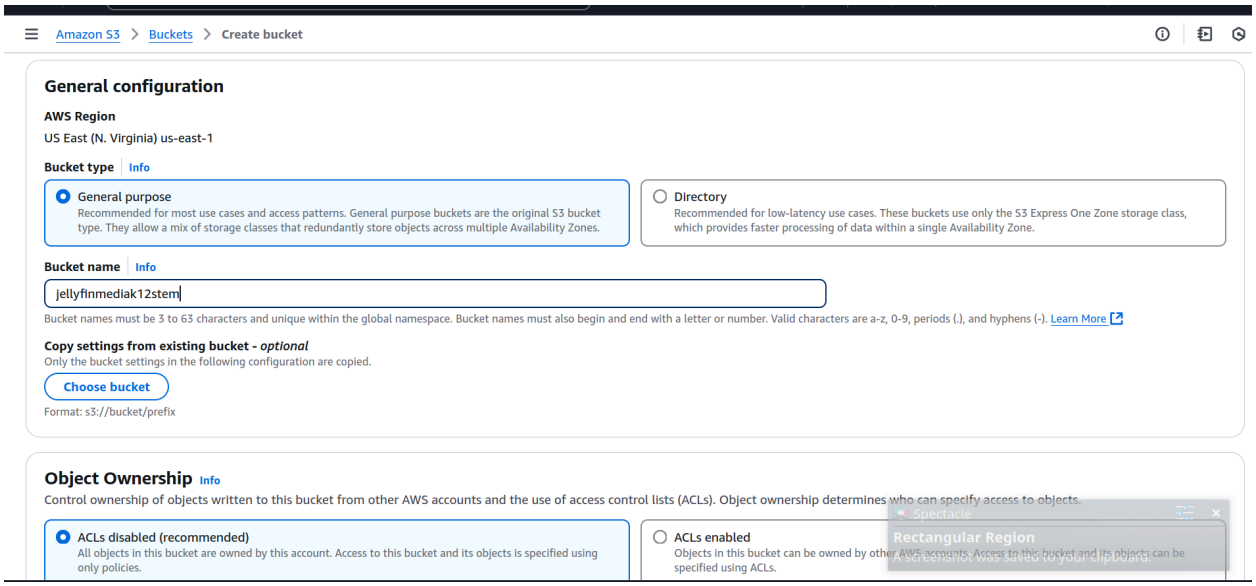


Finish installation and login with admin and password credentials created earlier

Creating S3 Bucket

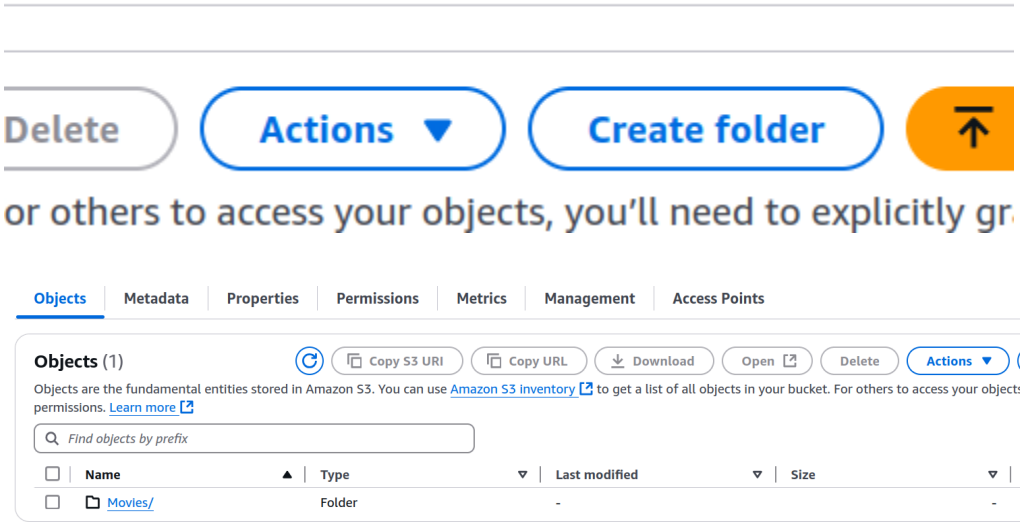
We will use an S3 bucket to easily upload files and store the videos.

Create the S3 bucket. Remember, bucket must be unique (globally too, across all users)

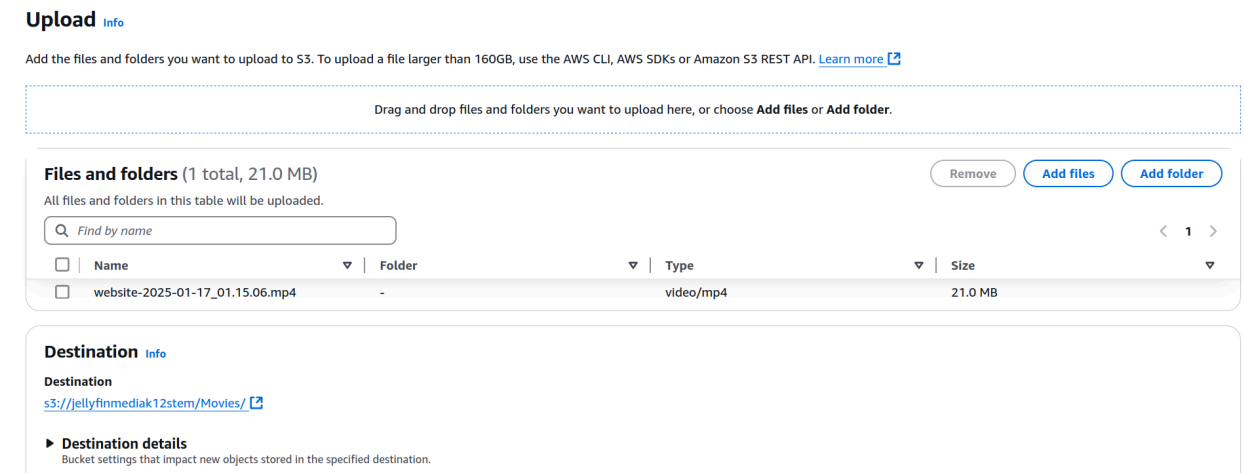


keep all default settings

Create a folder called Movies/



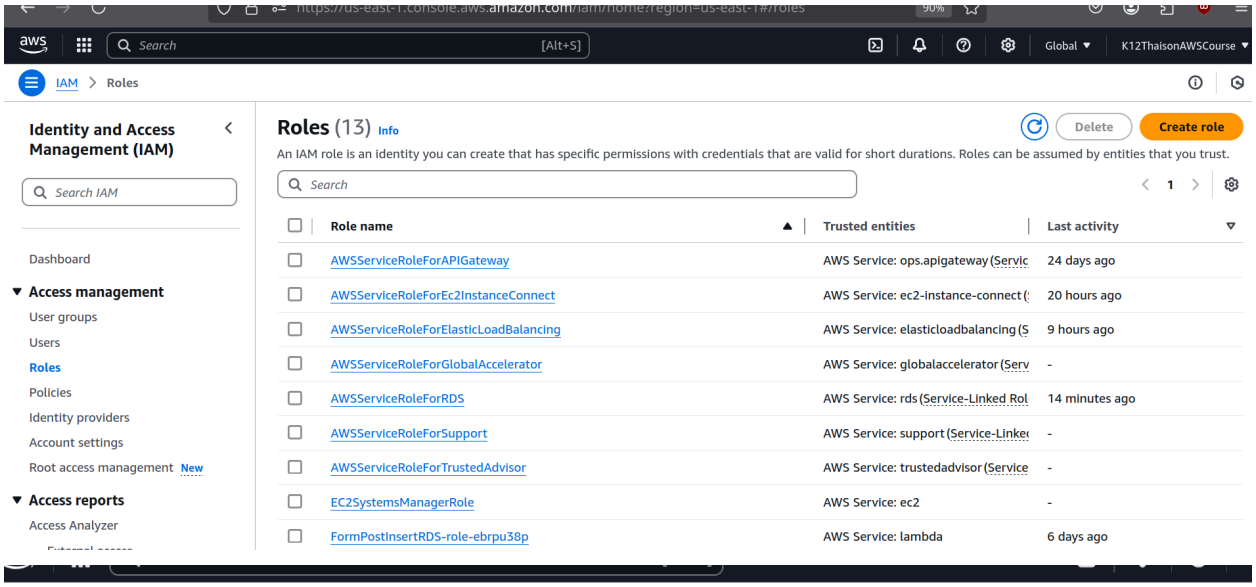
Upload your video of choice in the movies folder

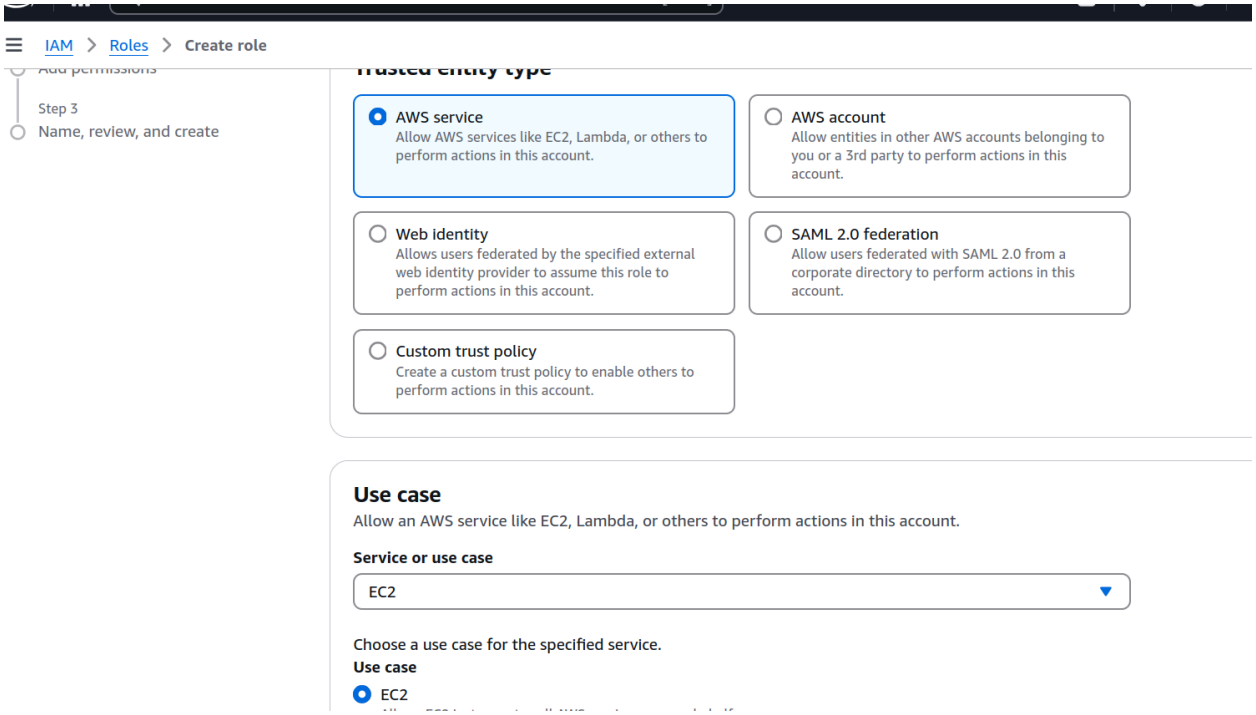


Syncing -> Creating an IAM Role

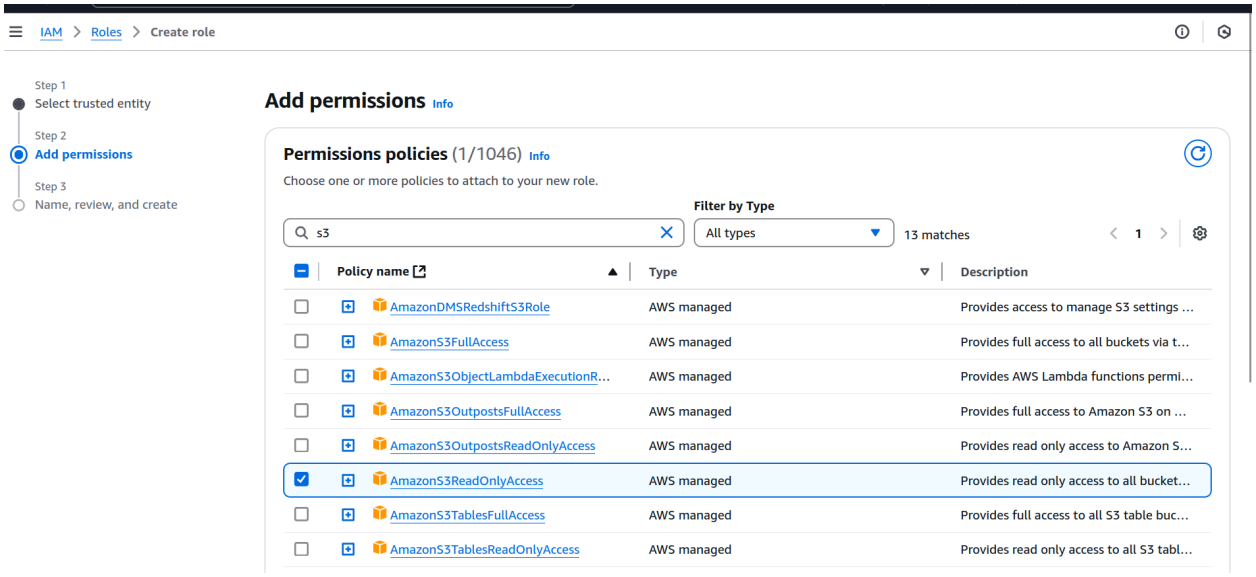
We need to create an IAM role to allow the EC2 to sync with the S3 bucket.

Go to IAM dashboard and create a new role.

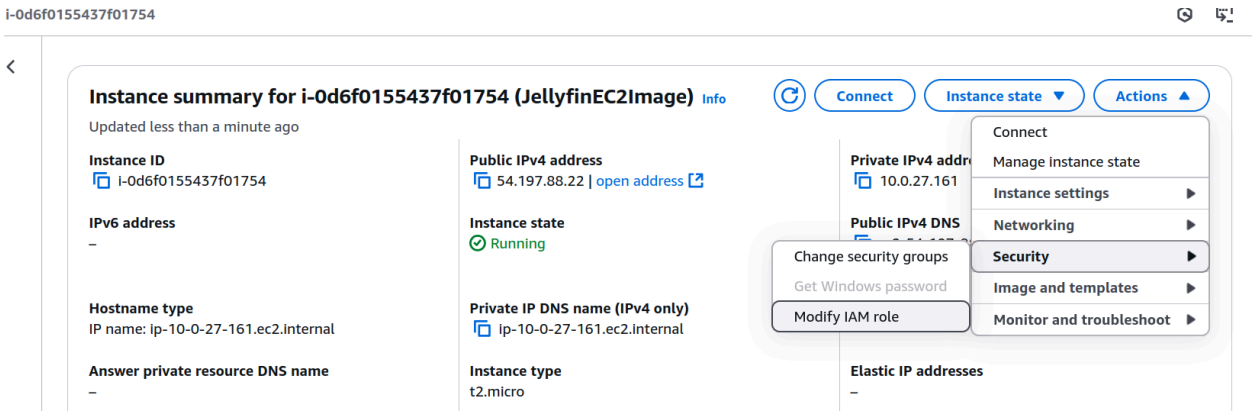

Search [Alt+S]
Global
K12ThaisonAWSCourse
IAM > Roles
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
Roles (13) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Delete
Create role
Search
1
Role name
Trusted entities
Last activity
AWSServiceRoleForAPIGateway
AWS Service: ops.apigateway (Servic
24 days ago
AWSServiceRoleForEc2InstanceConnect
AWS Service: ec2-instance-connect (
20 hours ago
AWSServiceRoleForElasticLoadBalancing
AWS Service: elasticloadbalancing (S
9 hours ago
AWSServiceRoleForGlobalAccelerator
AWS Service: globalaccelerator (Serv
-
AWSServiceRoleForRDS
AWS Service: rds (Service-Linked Rol
14 minutes ago
AWSServiceRoleForSupport
AWS Service: support (Service-Linke
-
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service
-
EC2SystemsManagerRole
AWS Service: ec2
-
FormPostInsertRDS-role-ebrpu38p
AWS Service: lambda
6 days ago

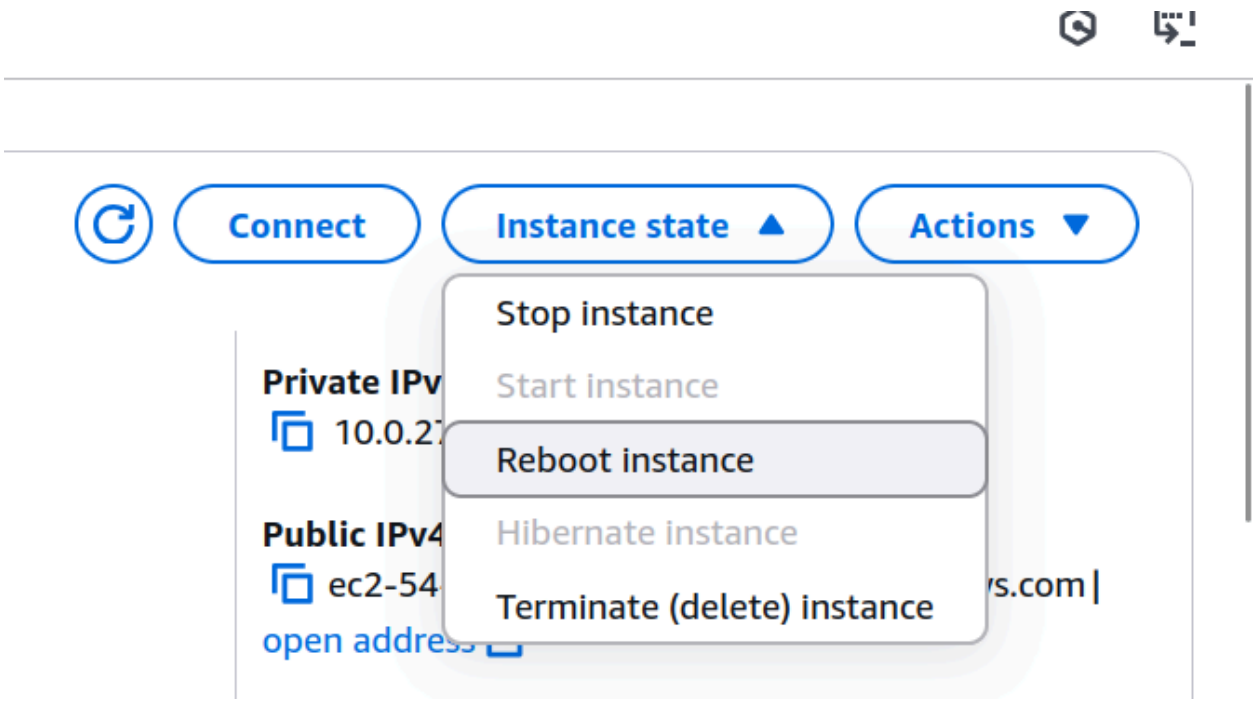

IAM > Roles > Create role
Step 3
Name, review, and create
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Service or use case
EC2
Choose a use case for the specified service.
Use case
EC2

Use the s3readonly policy. Name the role something reasonable.



Add the role to the EC2 instance





Sync bucket

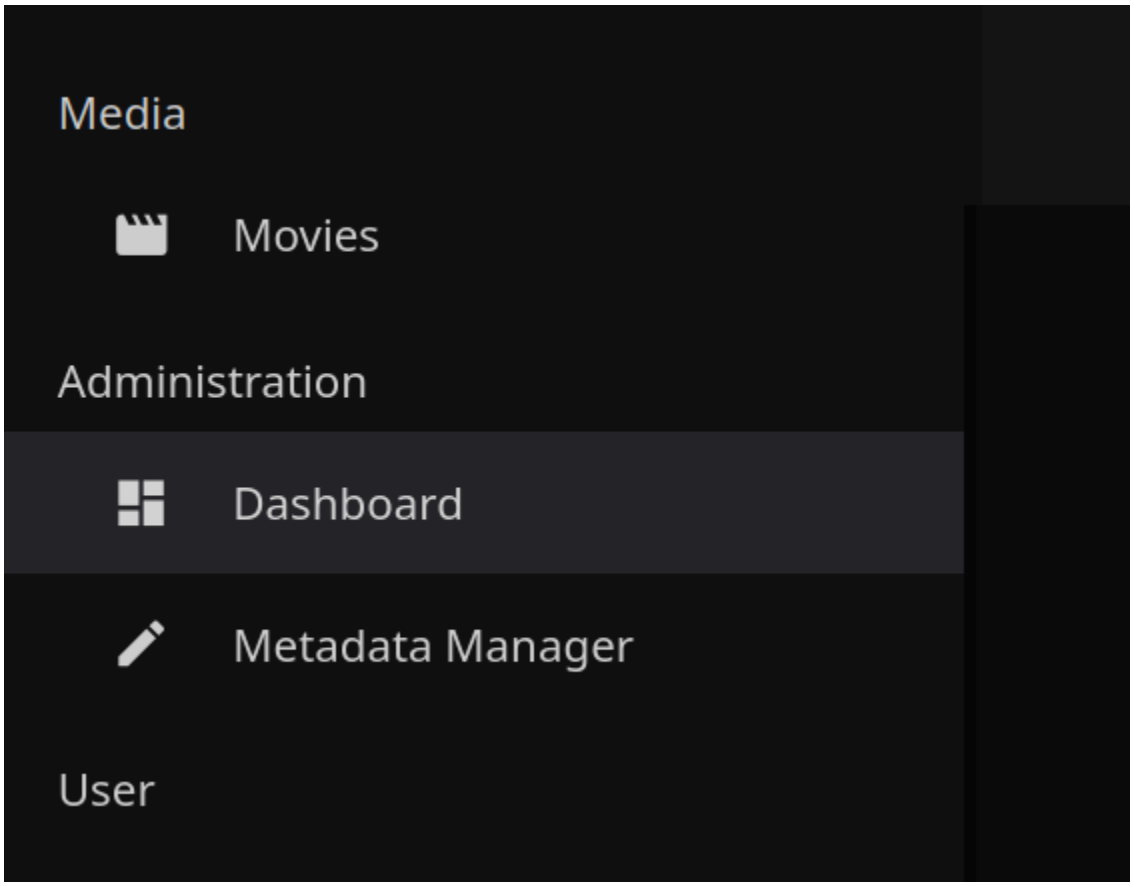
Sync your bucket by going through these commands

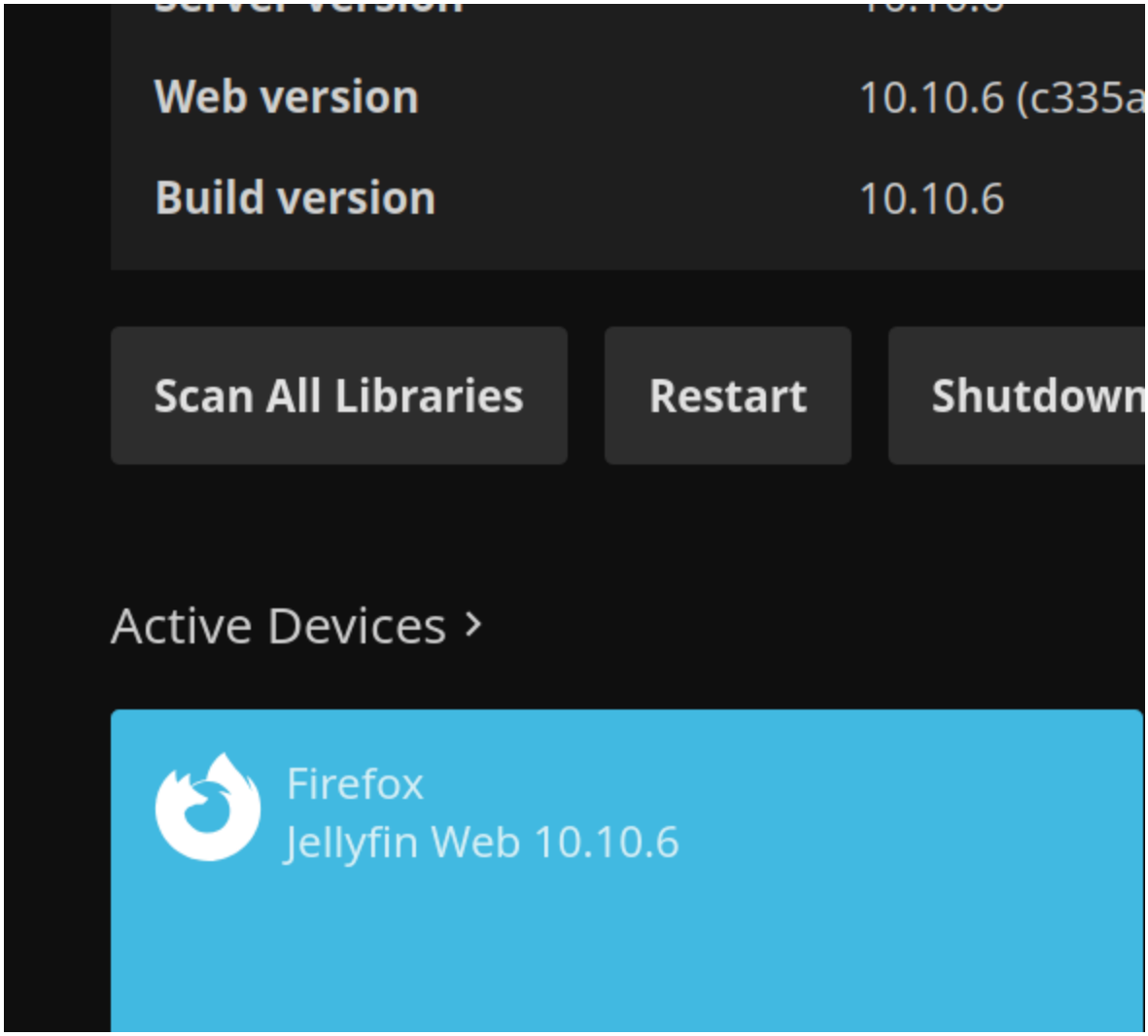
```
sudo apt update
sudo snap install aws-cli --classic

sudo -u jellyfin aws s3 sync s3://your-bucket-name/ /content/
example: sudo -u jellyfin aws s3 sync s3://jellyfinmediak12stem/ /content/
```

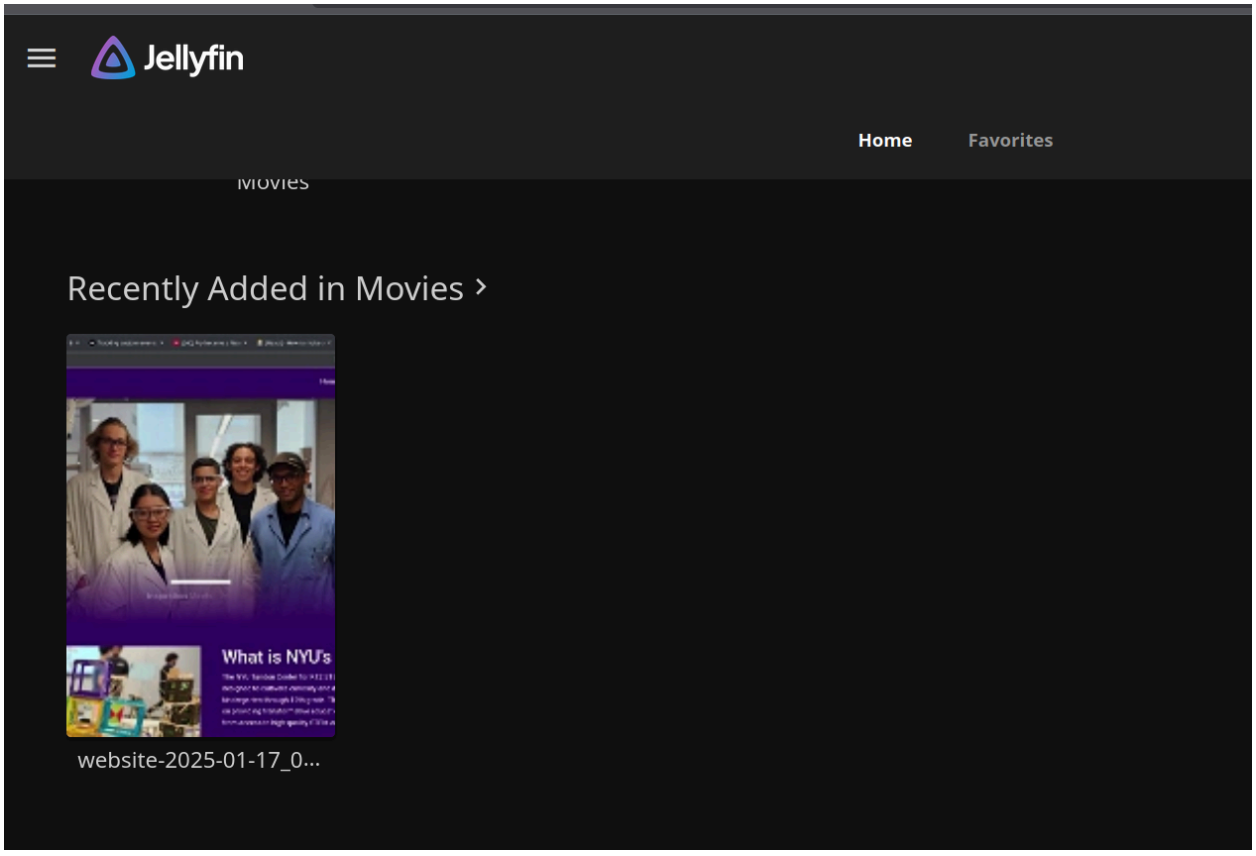
The video should now show up in your folders.

Refresh Jellyfin

Go to the administrator panel and refresh the library for jellyfin to find the files

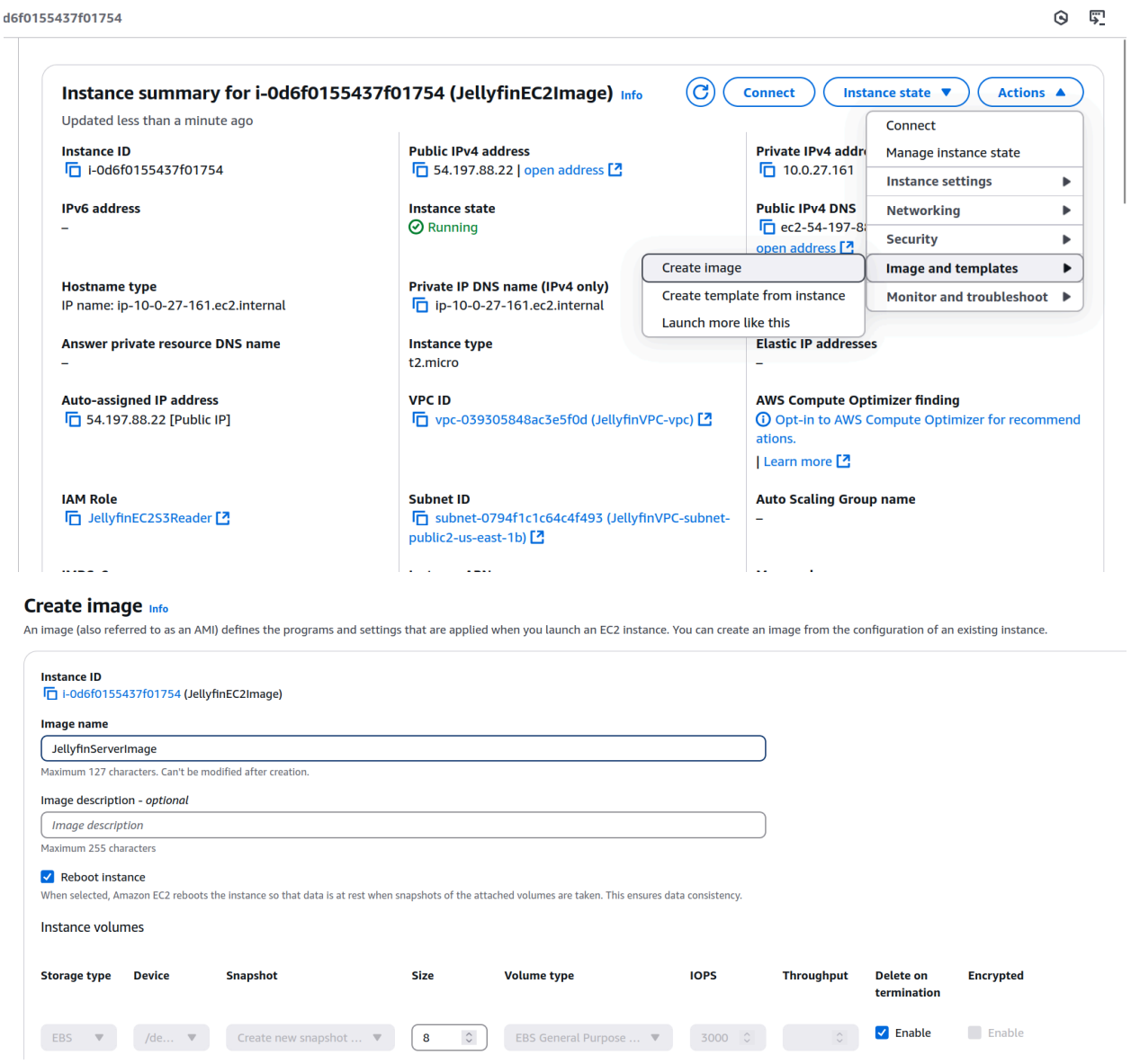



Web version
10.10.6 (c335a
Build version
10.10.6
Scan All Libraries
Restart
Shutdown
Active Devices ›
Firefox
Jellyfin Web 10.10.6



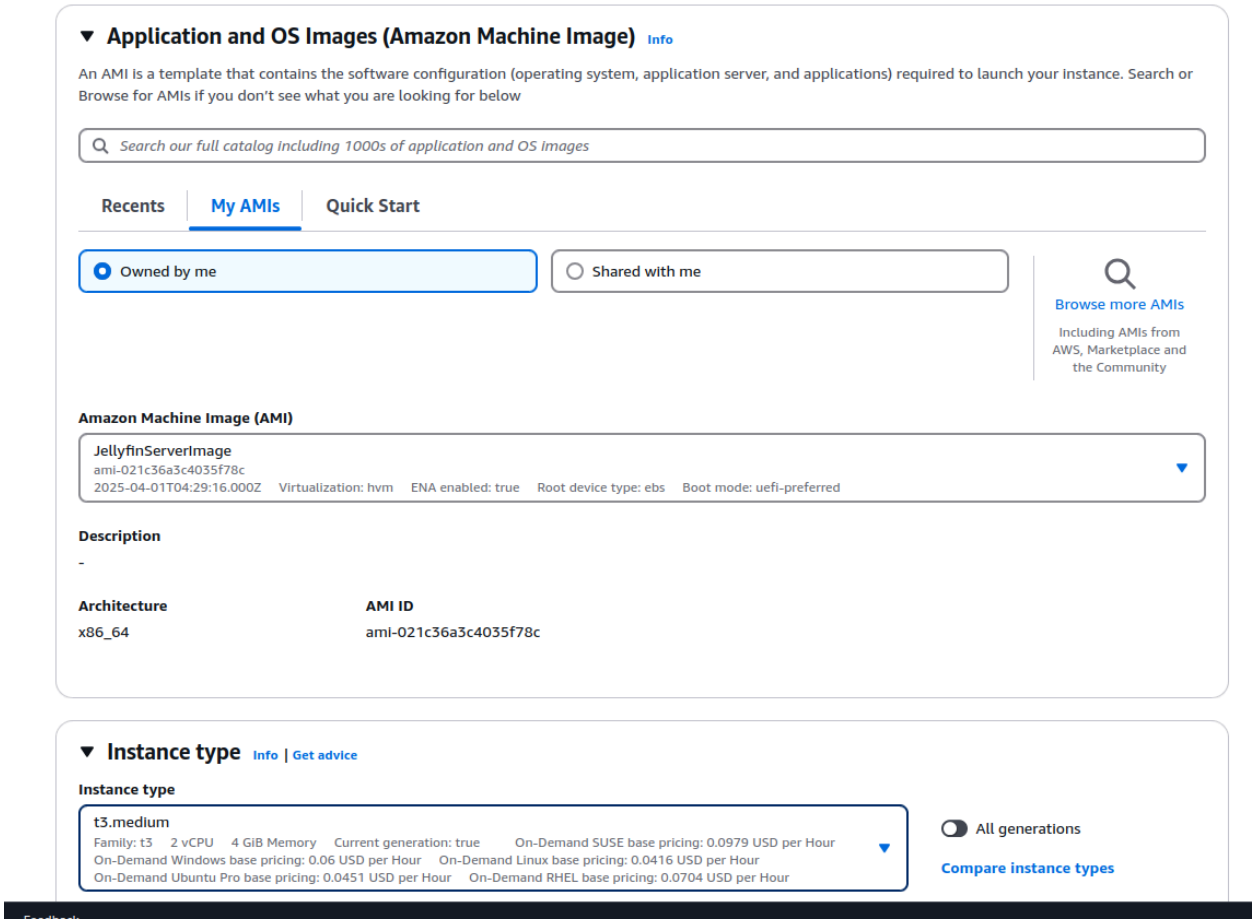
Creating an Image

Let’s create an Image of the server we just made (to save the work and also to create copies later)



Launching an EC2 instance with more Oomph (Do it yourself)

You can now use this image to launch instances that are stronger. The current t2.micro can only support direct play on very small videos. It will LAG on bigger videos like 1080p and 4k videos with multiple people watching at once. That's why we can use the image to create an instance that is exactly the same, but running with better hardware. We can also put it behind a private subnet now, if you automate the syncing, so that you don't have to interact with the server anymore.



End

Make sure to delete/stop and resources you are not using. Things to note:

- Your EBS storage volume is not automatically deleted when terminating the EC2. You have to manually delete it
- Your ALB must be deleted, it cannot be just turned off
-