**CS486 User Manual**

Revision 1.0

**April 23, 2025**

**StratoSplit**

**Client:**

General Dynamics Mission Systems

**Mentor:**

Brian Donnelly, Savannah Chappus

**Team Members:**

Sam Cain

Nolan Newman

Dallon Jarman

Elliot Hull

# Introduction

We are pleased you have chosen StratoSplit for your business needs. StratoSplit is a powerful system designed to integrate multicast audio streaming within an operator web console for real time 3d audio management and transmission that has been custom-designed to meet your needs. Some of the key features include: user role management, passwordless login, bidirectional audio panning, configuration management, and team management. The purpose of this user manual is to help you, the client, successfully install, administer, and maintain the StratoSplit product in your actual business context going forward. We aim to make sure that you can integrate this system into existing systems, build upon it, and maintain it.

# Installation

As part of final delivery, the StratoSplit system should have been installed on a platform of your choice. Over time, however, you may want to move to a new platform or re-install the product. Below are the necessary hardware, toolchain, and steps to recreate our deployment environment.

Hardware: Two t2 micro EC2 Ubuntu 24.0.4 instances

Toolchain: EC2, Transit Gateway, VPC, Cloudflare, Hanko, MongoDB, Node JS, Python

## AWS Setup:

1. Create VPC:
   Navigate to ‘VPC > Your VPCs > Create VPC’ and create a VPC providing a name and IPv4 CIDR ‘10.99.0.0/16’. Leave everything else default.

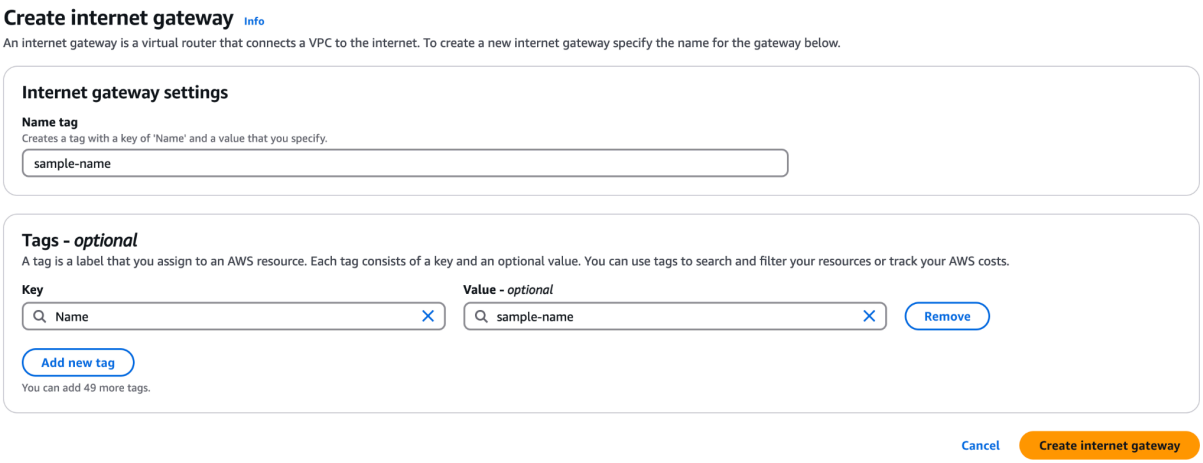

2. Create Subnets:
   Navigate to ‘VPC > Subnets > Create Subnets’ and create subnets as follows. Select newly created VPC and provide names, subnet zones, and CIDRs as follows:
   Public 1: east 1a - 10.99.0.0/18
   Public 2: east 1b - 10.99.64.0/18
   Private 1: east 1a - 10.99.128.0/18

Private 2: east 1b - 10.99.192.0/18

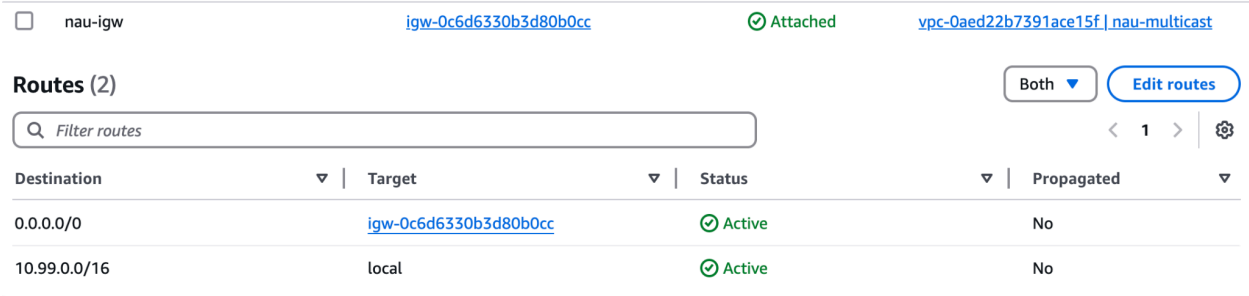
| | | | | | | |
|---|---|---|---|---|---|---|
| ☐ | nau-public1 | subnet-... | Available | vpc-0ae... | Off | 10.99.0.0/18 |
| ☐ | nau-private2 | subnet-... | Available | vpc-0ae... | Off | 10.99.192.0/18 |
| ☐ | nau-public2 | subnet-... | Available | vpc-0ae... | Off | 10.99.64.0/18 |
| ☐ | nau-private1 | subnet-... | Available | vpc-0ae... | Off | 10.99.128.0/18 |

3. Create Internet Gateway:

   Navigate to 'VPC > Internet Gateways > Create Internet Gateway' within AWS and create a new internet gateway.

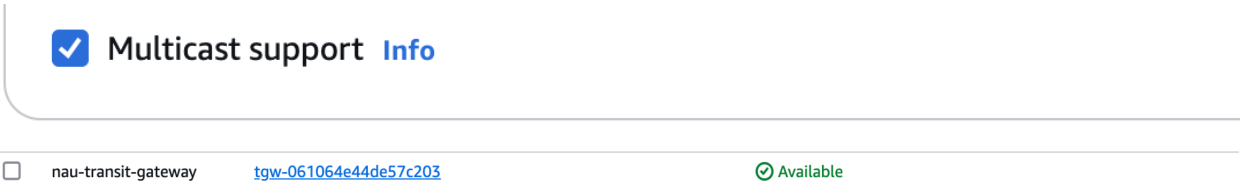



   Add new IGW to 'Public 1' routing table and 'Public 2' routing table.




4. Create Transit Gateway:

   Navigate to 'VPC > Transit Gateways > Create Transit Gateway' within AWS. Check enable multicast support, leave all else default, and confirm.



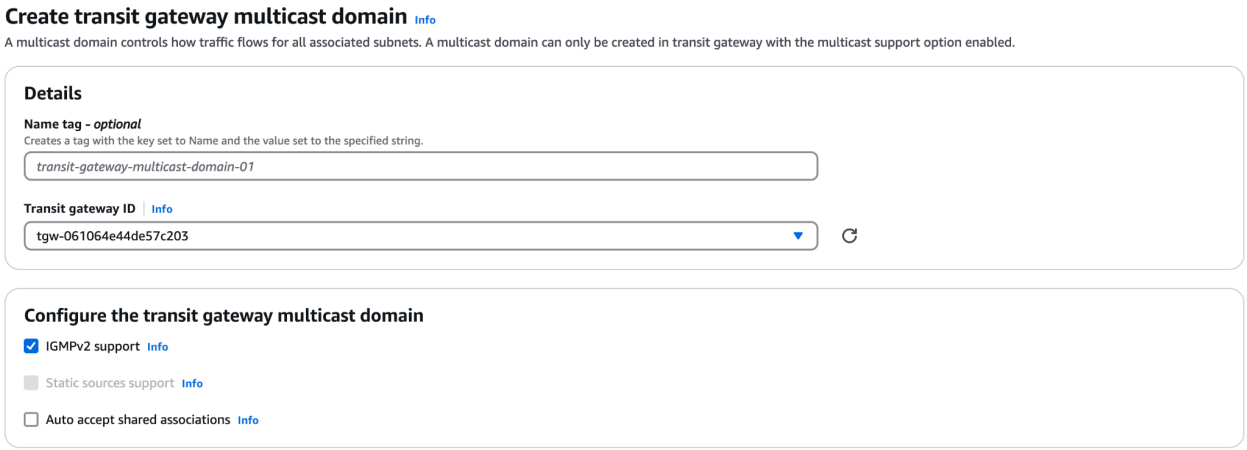

| | | | |
|---|---|---|---|
| ☐ | nau-transit-gateway | tgw-061064e44de57c203 | Available |

5. Create Transit Gateway Attachment:

   Navigate to ‘VPC > Transit gateway attachments > Create transit gateway attachment‘. Select the new Transit Gateway and the VPC. The subnets should auto populate with the public subnets. Create the transit gateway attachment.

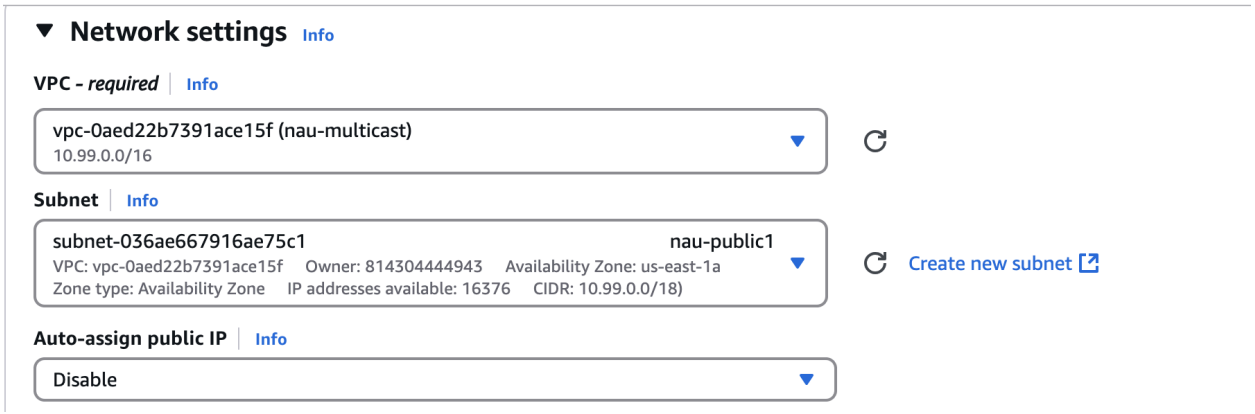

6. Create Transit Gateway Multicast Domain:

   Navigate to ‘VPC > Transit gateway multicast domains > Create transit gateway multicast domain’ check enable IGMP2 support and attach transit gateway. Create the multicast domain.

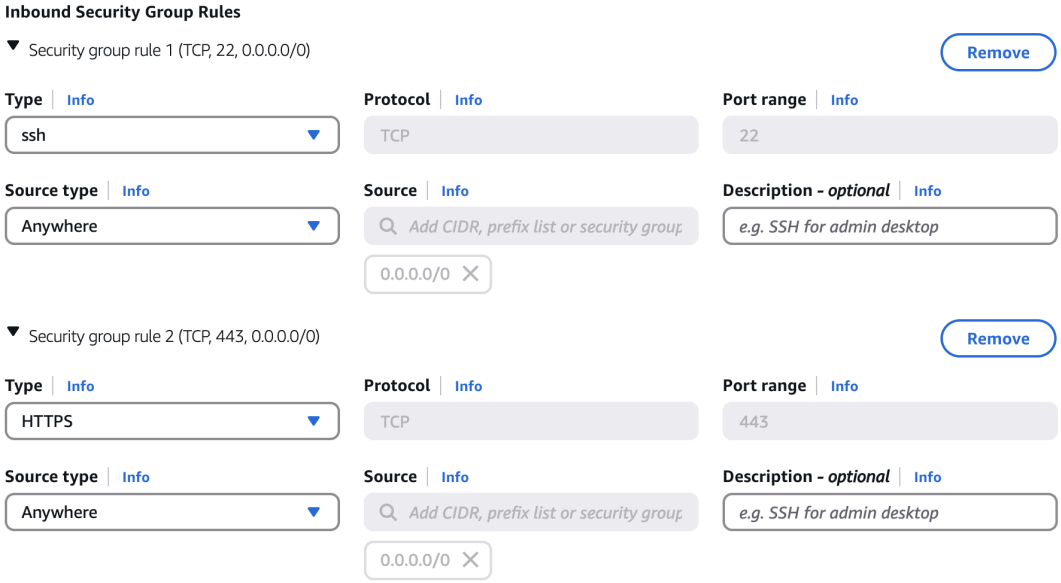



7. Set up EC2 Host:

   Navigate to ‘EC2 > Instances > Launch an instance’ and select your VPC.




Make sure inbound SSH and HTTPS traffic is enabled.



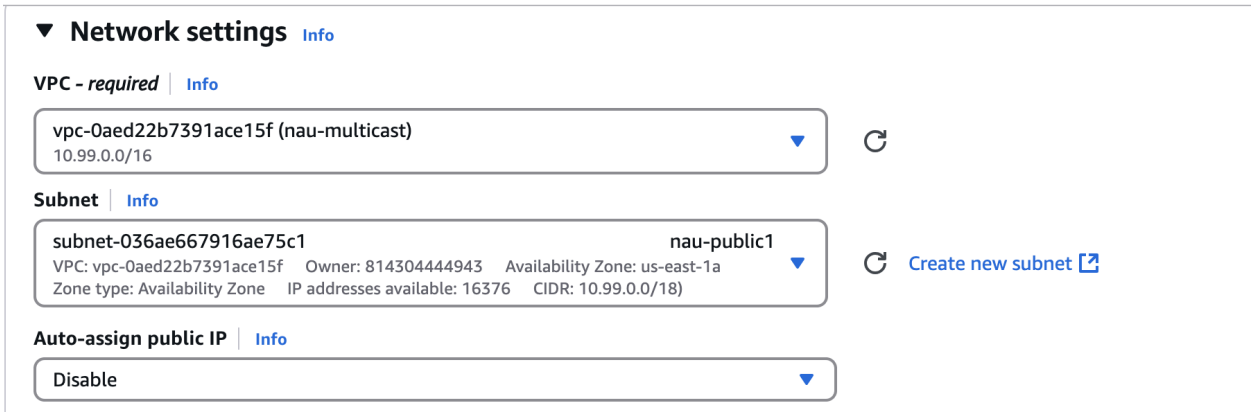

Add an additional inbound rule for ‘Custom UDP’ over port range ‘5001 - 5019’ from the source ‘ 10.99.0.0/16’


Navigate to ‘VPC > Elastic IP Addresses > Allocate Elastic IP Address’ and allocate and associate Elastic IP to the new EC2 Instance.

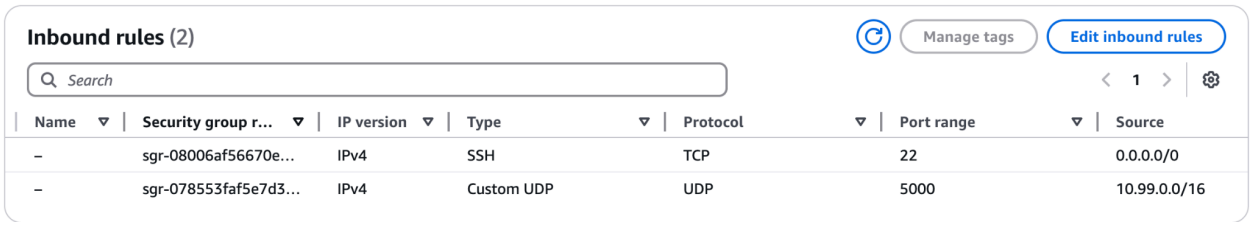
Configure DNS using the provider of choice with this new Elastic IP Address.

8. <u>Set up EC2 Audio Generator:</u>

   Navigate to ‘EC2 > Instances > Launch an instance’ and select your VPC.



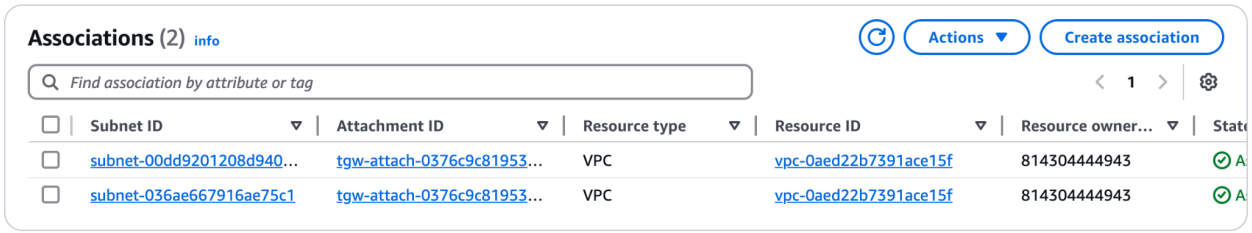

Configure inbound security as per the following rules:




Navigate to ‘VPC > Elastic IP Addresses > Allocate Elastic IP Address’ and allocate and associate Elastic IP to the new EC2 Instance.

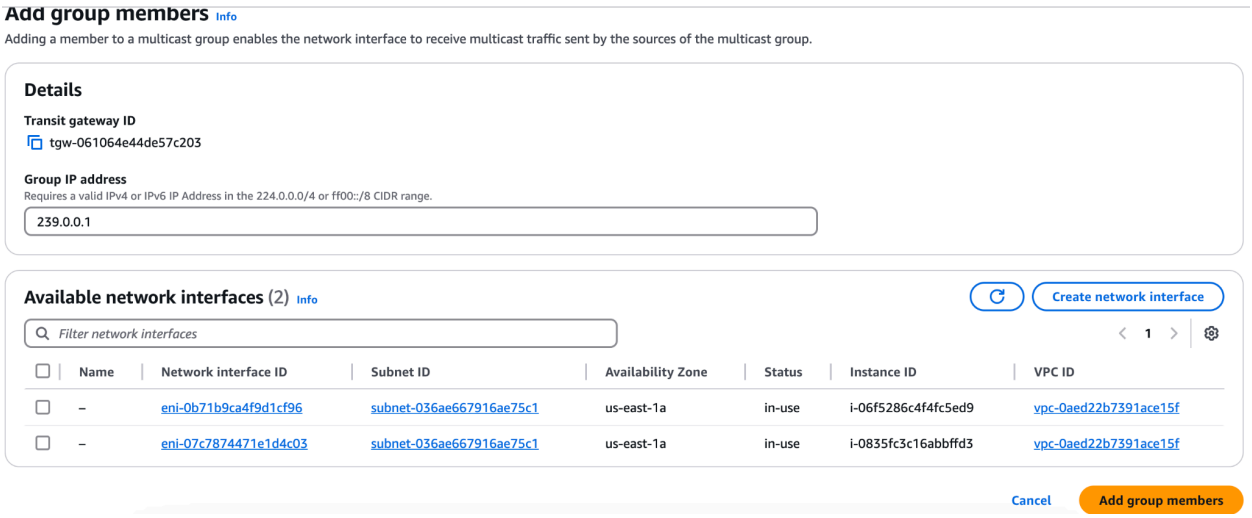
9. Create Transit Gateway Multicast Domain Associations:

   Navigate to your Transit Gateway Multicast Domain and create 1 association per public subnet within your VPC.



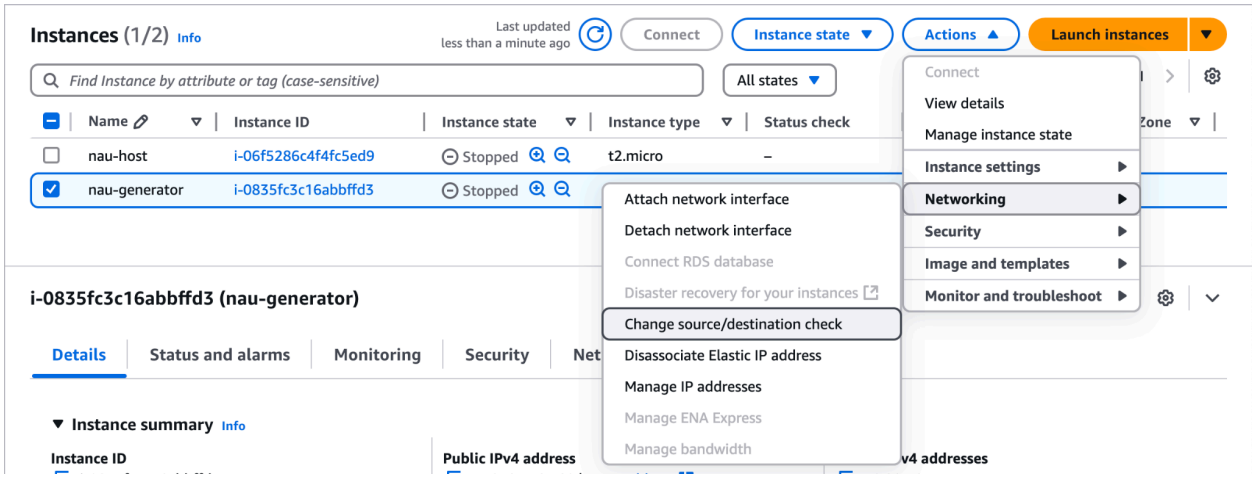

10. Create Multicast Groups:

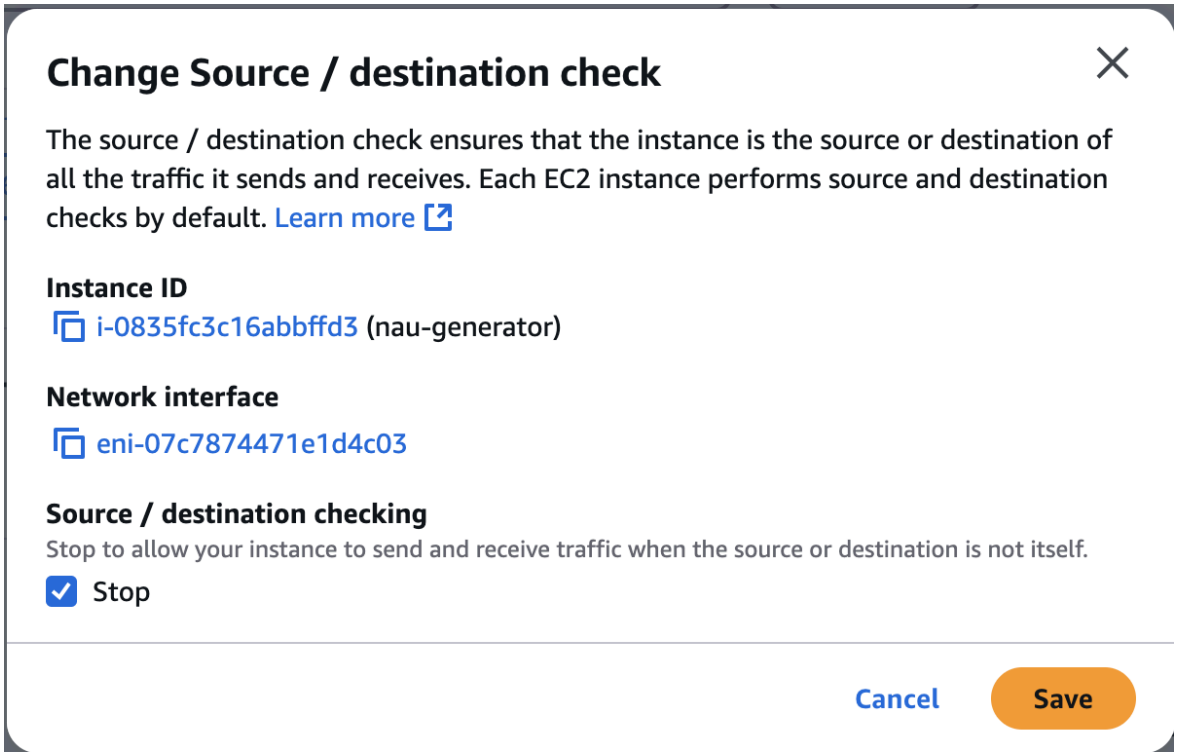
    Navigate to your Transit Gateway Multicast Domain. Select groups and add members based on your specific needs. Select both network interfaces and provide a multicast IP address to enable.




11. <u>Disable Source Destination Check:</u>

    Navigate to ‘EC2 > Instances’ select the ‘Actions’ dropdown. In the networking tab click ‘Change source/destination check’, check ‘Stop’, and save the changes.

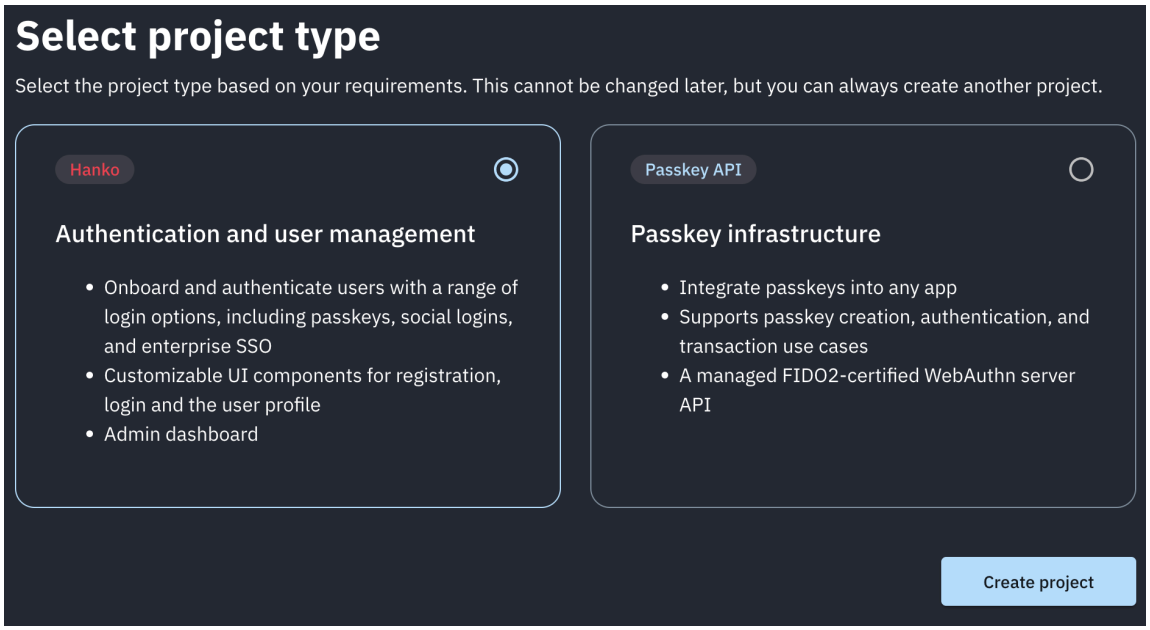




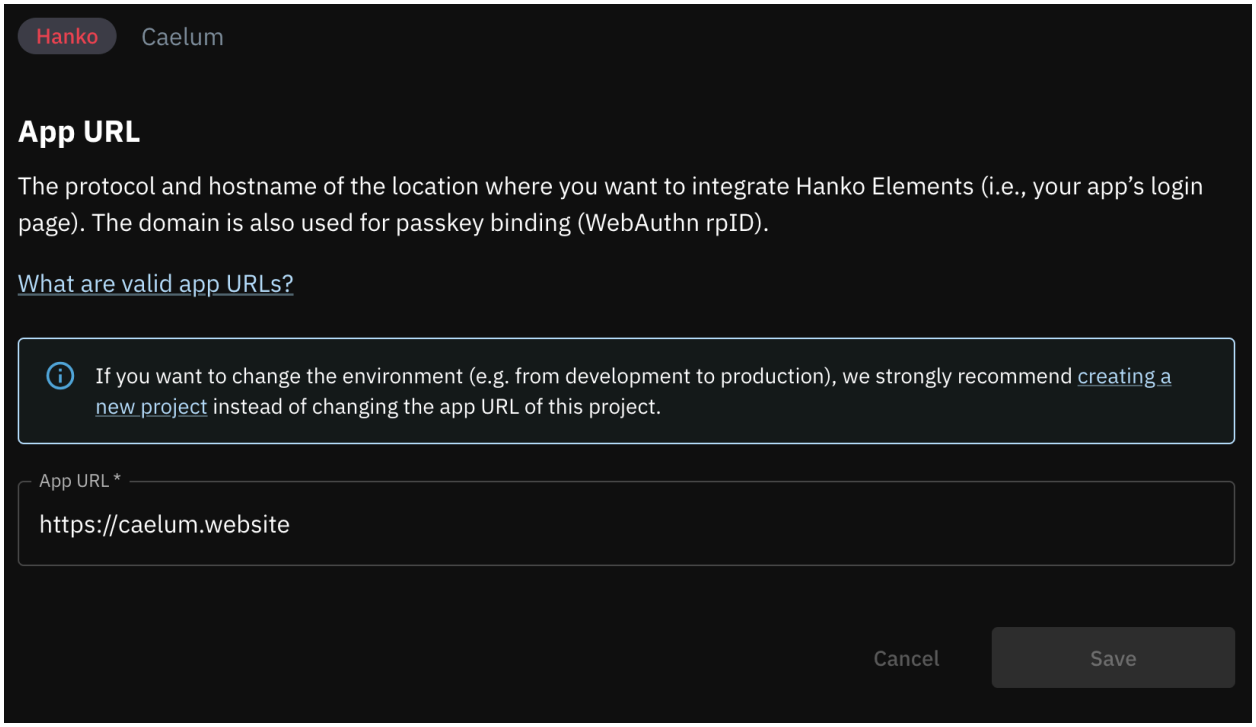

Multicast Traffic should now be successfully enabled across your EC2 instances. Note traffic will only come through on ports with inbound traffic enabled via the security wizard. The ports and IPs used in this guide are relevant to the application we created.

## Hanko Cloud:

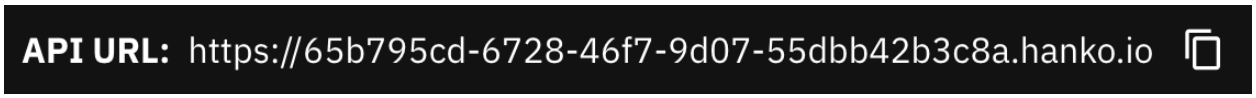
Navigate to cloud.hanko.io and create a new Hanko project.



Set the app URL to the domain name or ‘localhost:443’.

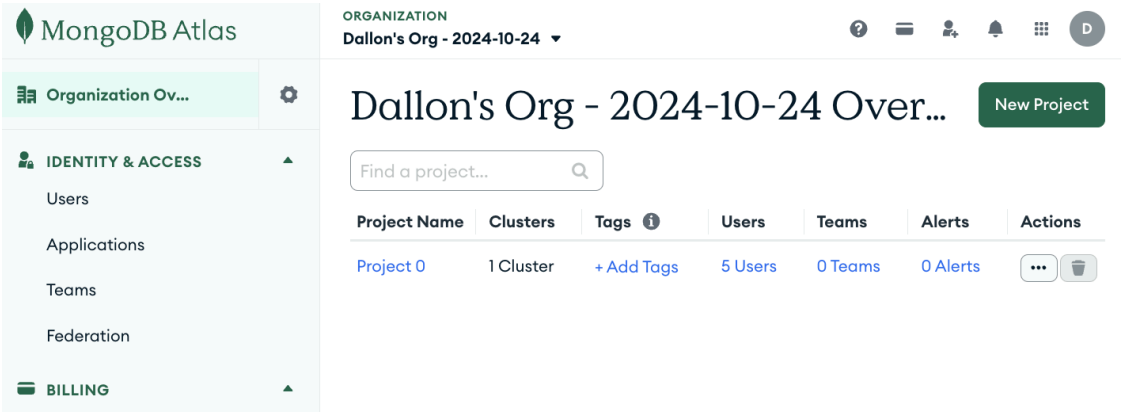



Store the API url in the Dashboard page for the ‘.env’ file on the host machine.

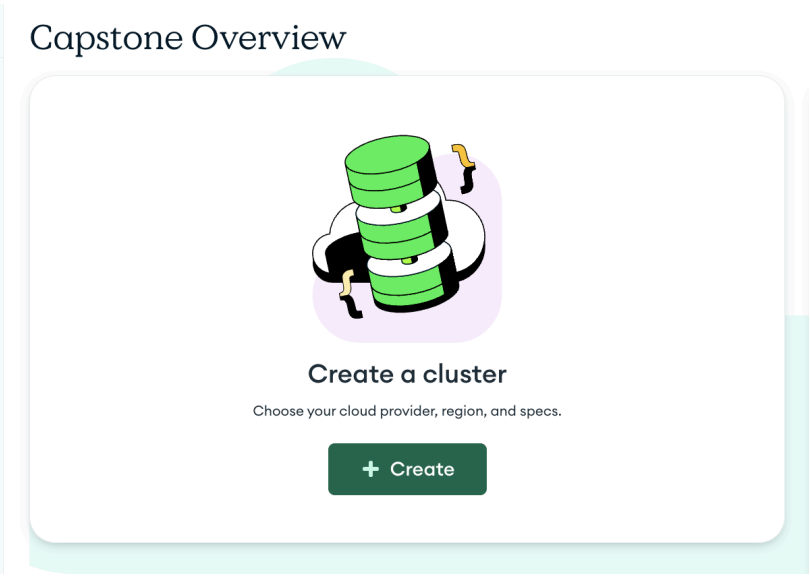



## MongoDB:

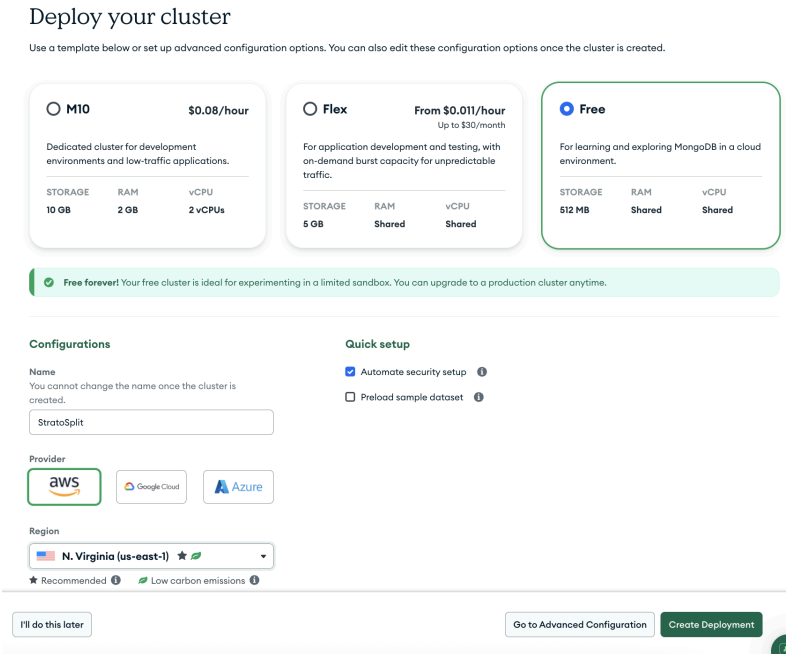
1. Navigate to mongodb.com and log into the cloud console.
2. Once logged in, navigate to the top right of the page and click on New Project



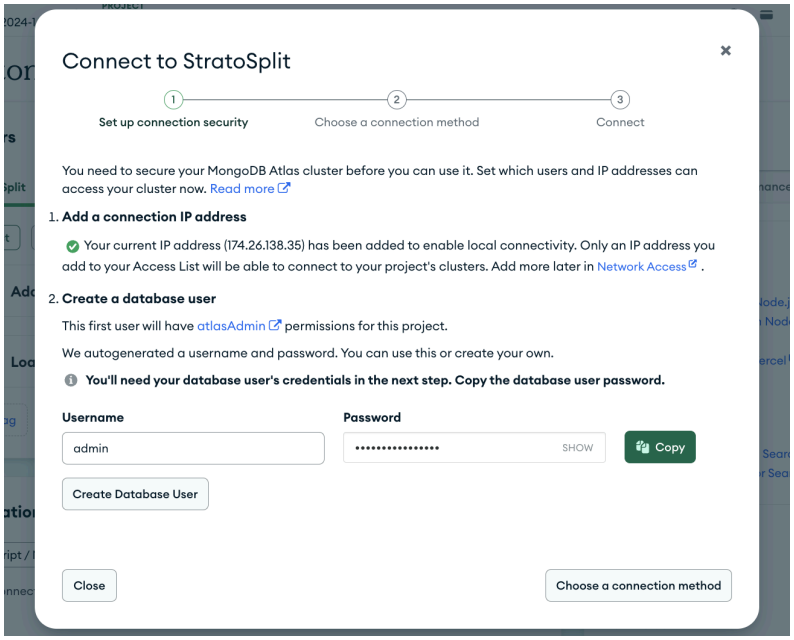

3. Give the project a useful name such as Capstone and hit next then Create Project.
4. Once the project is created, you will be able to create your database. Click on Create



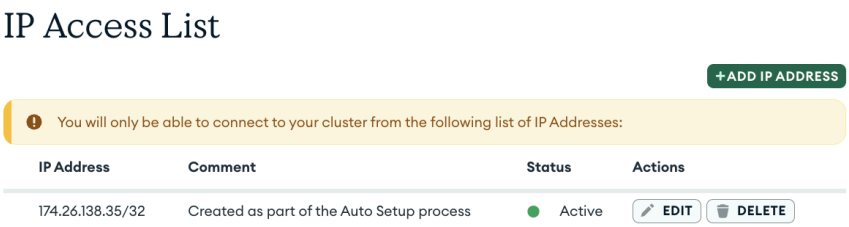

5. Choose the Free Tier Cluster, name your cluster the name of the application, your preferred provider, and the Region that is closest to the location of server hosting your project. Then hit Create Deployment




6. Create your user account that will run as the administrator. Make sure to remember this password, it can be reset later if forgotten. Then hit **Create Database User**.

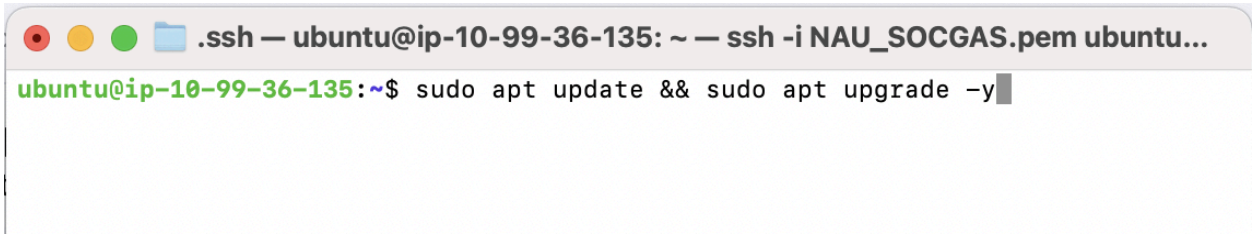



7. For the connection method, choose Drivers and copy step 3 and paste that into the .env file
8. For the last step, on the left hand side, click on **Network Access** and edit the IP address in the options and change it to either 0.0.0.0/0 or to your server's IP address.




## Machine 1 Host:

1. Update System Packages:




2. Pull Github Repository:

```
ubuntu@ip-10-99-63-138:~$ git clone https://github.com/StratoSplit/Caelum.git
Cloning into 'Caelum'...
remote: Enumerating objects: 323, done.
remote: Counting objects: 100% (323/323), done.
remote: Compressing objects: 100% (199/199), done.
remote: Total 323 (delta 163), reused 258 (delta 111), pack-reused 0 (from 0)
Receiving objects: 100% (323/323), 18.15 MiB | 37.54 MiB/s, done.
Resolving deltas: 100% (163/163), done.
ubuntu@ip-10-99-63-138:~$
```

3. Install Node js and Dependencies:

   Install Node

```
ubuntu@ip-10-99-63-138:~$ sudo apt install -y nodejs
```

   Install base dependencies and run fix

```
ubuntu@ip-10-99-63-138:~$ cd Caelum/app
ubuntu@ip-10-99-63-138:~/Caelum/app$ npm i
```

```
ubuntu@ip-10-99-63-138:~/Caelum/app$ npm audit fix
```

   Install OS specific dependencies

```
.ssh — ubuntu@ip-10-99-63-138: ~/Caelum/app — ssh -i NAU_SOCGAS....
ubuntu@ip-10-99-63-138:~/Caelum/app$ npm install dotenv bcrypt
```

4. Set up .env:

   Use your personal API URLs to set up the .env file.

```
 GNU nano 7.2                                                    .env
MONGO_URI=mongodb+srv://root:NokkikBSFJJp1WvA@capstone.zgone.mongodb.net/?retryWrites=true&w=majority
MONGO_DB_NAME=Caelum-Dallon
HANKO_API_URL=https://65b795cd-6728-46f7-9d07-55dbb42b3c8a.hanko.io
SSL_KEY_PATH=./key.pem
SSL_CERT_PATH=./cert.pem
```

5. Manually configure first admin account:

   *User data was configured using MongoDB Compass connected to the database.

```
_id: ObjectId('67e5acd9a7dea2310b86695c')
userId : "ea2eb43f-a7e1-4723-a8d7-f546bafc5861"
username : "nolan"
email : "nolannew259@gmail.com"
lastLogin : 2025-03-27T21:55:26.052+00:00
createdAt : 2025-03-27T19:54:01.820+00:00
role : "admin"
team : "67da11b0dc8cdfec242b7dff"
```

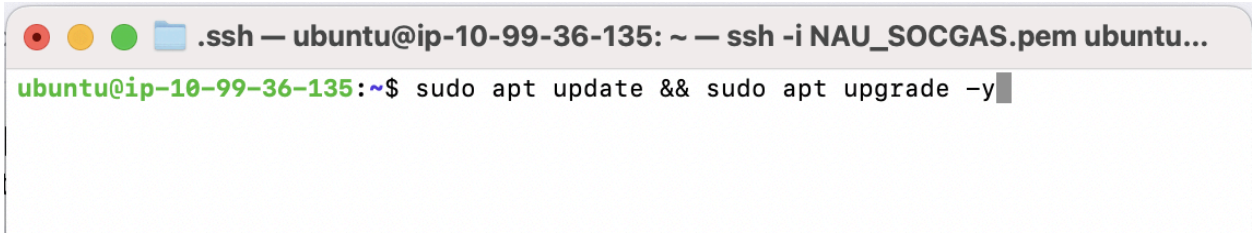
Role was manually set to “admin”.

6. Run node server:

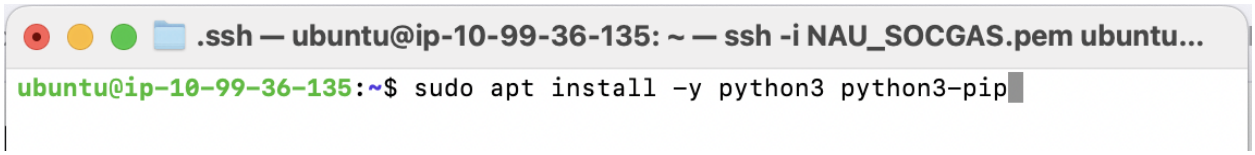
```
ubuntu@ip-10-99-63-138:~/Caelum/app$ sudo node server.js
```

## Machine 2 Audio Stream Generator:

1. Update System Packages:

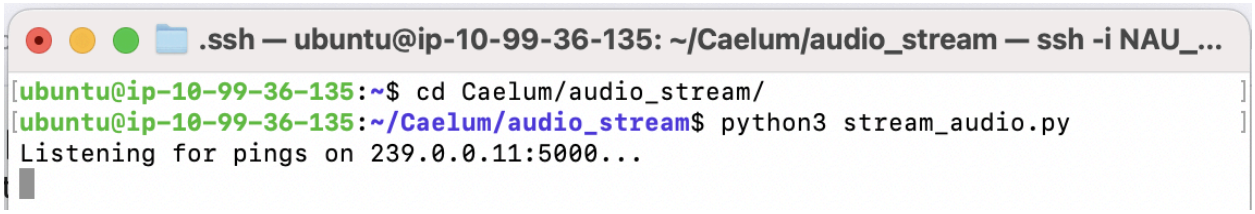



2. Install Python3 and pip:




3. Pull Github Repository:


4. Run stream_audio.py:




## Configuration and Daily Operation

In the last section, you hopefully ended with the client being able to log into (or connect to, or whatever) the installed product. In this section, you will detail whatever tasks need to be done to get the product deployed and operational. Details depend on the individual product but might include steps like “Configure admin user profile and password”, “Create user accounts”, etc. This is the bulk of your user manual, and should simply cover any tasks the client may need to do (while consulting this manual) on a regular basis to operate the product.

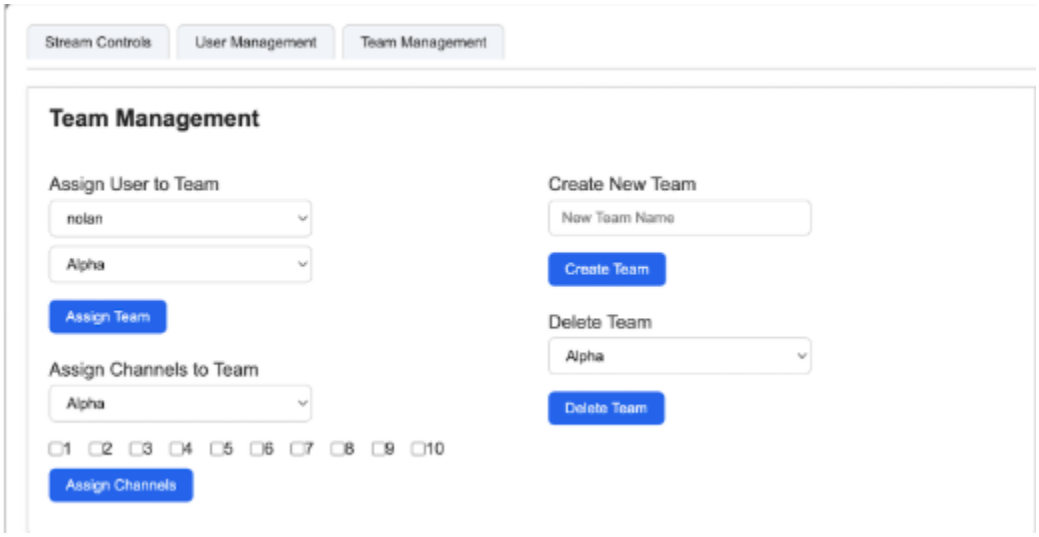



Image 1

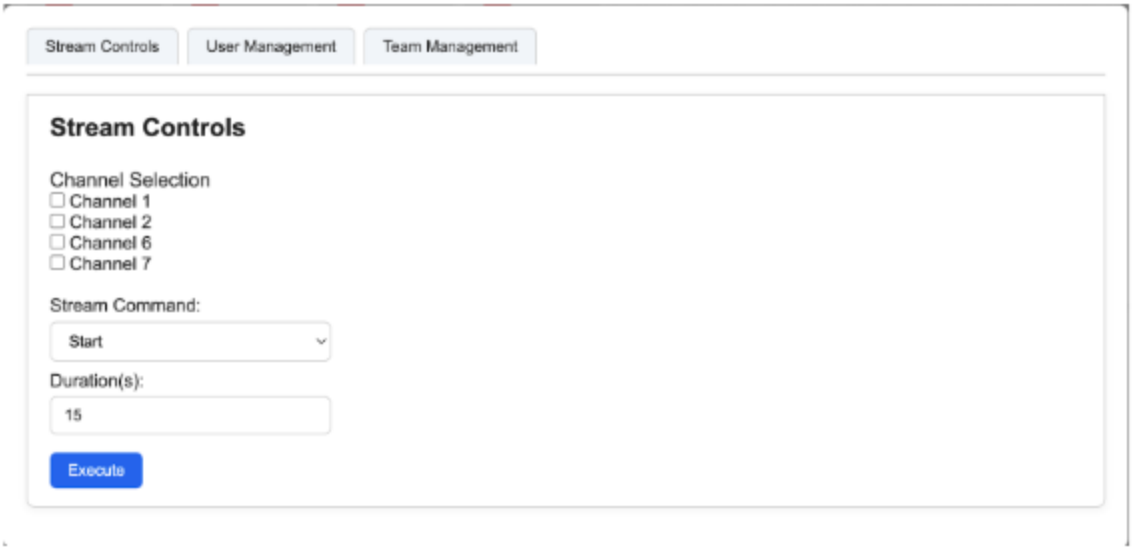



Image 2

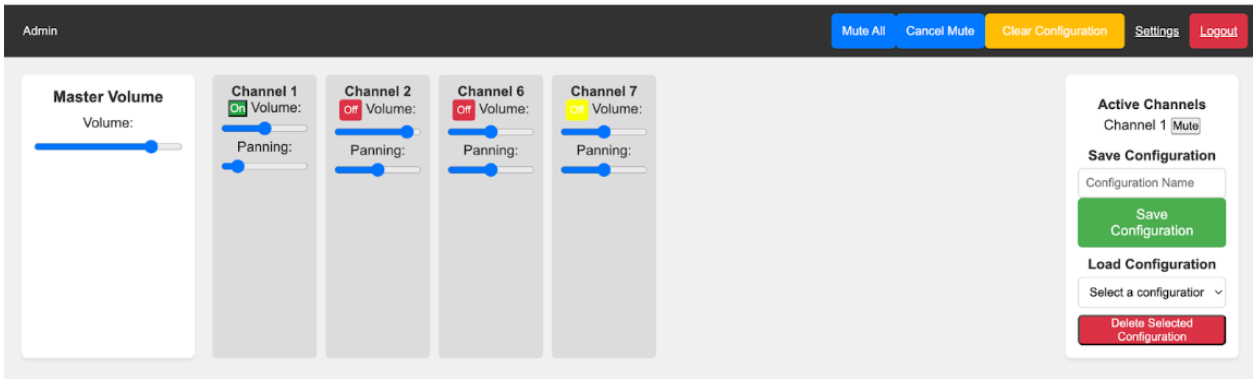



Image 3

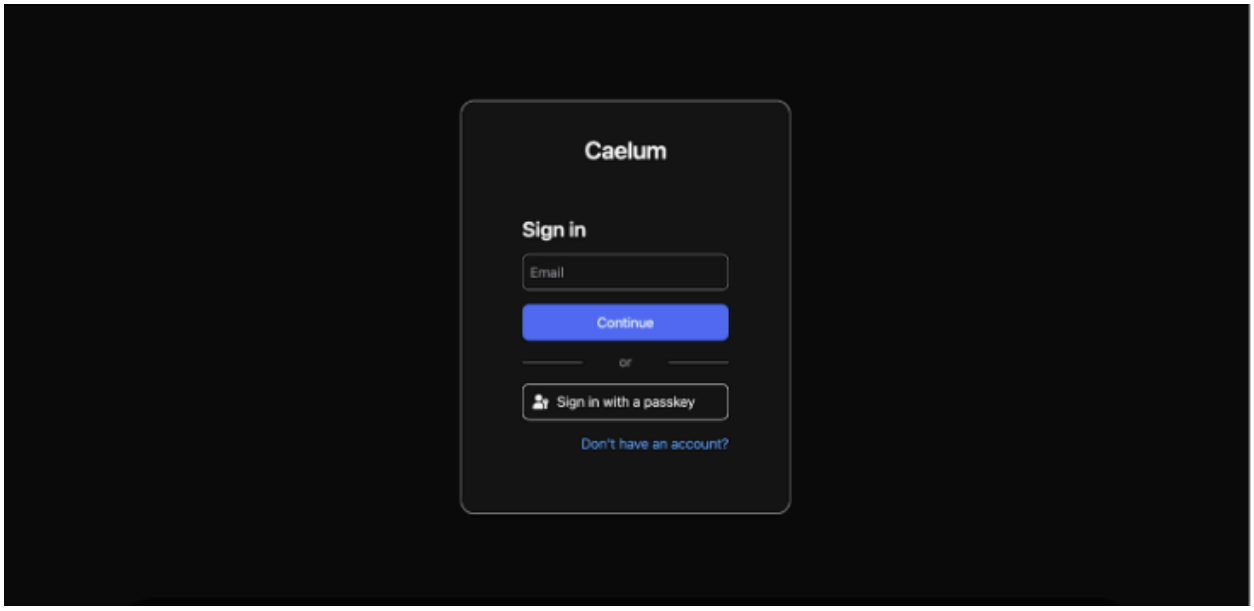



Image 4

**Assigning user to a team**

- Enter the admin panel in team management use the drop-down menu to assign a specific user to a specific team(image 1)

**Creating a team**

- Enter the admin panel under team management and enter team name into new team name box and press Create Team button(image 1)

**Deleting Teams**

- Enter the Admin Panel under Team Management and select team name in dropdown then press Delete Team button(image 1)

**Assigning Channels to a team**

- Enter the Admin Panel under Team Management and select the team from the drop down. Press channels required then, Assign Channels button(image 1)

**Start Streams**

- Enter admin panel under Stream Controls select streams out of currently available then enter duration and press Execute button.(Image 2)

**Create Configuration**

- After configuring dashboard as preferred enter configuration name and press save configuration(image 3)

**Load Configuration**

- Select saved configuration under drop down menu(image 3)

**Deleting configuration**

- Load configuration and press Delete Configuration button below it(image 3)

**Controlling volume**

- Volume can be controlled in a multitude of ways including; changing the master volume of the website, controlling individual channel audio using volume bar, mute all channels using mute all button. Can be saved using configurations(image 3)

**Using Panning audio**

- Panning can be controlled using a panning bar below the volume bar. Can be saved using configurations(image 3)

**Login to website**

- Enter website url and sign in using passkey, face id, or touch id(image 4)

# Maintenance

1. **System Updates (EC2 Instances)**

Both EC2 instances—host console and audio generator—run Ubuntu and depend on up-to-date system packages for stability and security.

SSH into each instance and run:

sudo apt update && sudo apt upgrade -y

2. **Dependency Maintenance (Node.js and Python)**

StratoSplit uses Node.js and Python for its backend and audio streaming services. Keeping dependencies current ensures security patches and compatibility with evolving system libraries.

npm outdated

npm audit fix

pip list --outdated

3. **Authentication Token Policy (Hanko)**

Passwordless authentication is handled through Hanko, which issues session tokens to users. These tokens may expire or require policy updates.

Log in to the Hanko Cloud Console, navigate to your project settings, and review the current token expiration policies. If necessary, update token lifespans, revoke inactive users, or reset credentials through the Hanko dashboard or the MongoDB users collection.

# Troubleshooting

**AWS:**
In the case that you have set up the AWS infrastructure as specific yet multicast traffic is still not enabled ensure that you disable source/destination checking and your inbound traffic rules are properly configured.

**Node js:**
Node may be prone to various issues during set up. For example, ‘Bcrypt’, a package we are using, has different versions depending on the host operating system. These must be manually installed outside of running the normal “npm install” command. Refer to the ‘Machine 1 Host’ step 3 to see these packages. Further, packages may be outdated depending on the version of Node and npm you are running. To fix these refer to the “Dependency Maintenance” section. Lastly, if you are still having issues setting up the web application, check that your ‘.env’ file is configured properly.

**Hanko API:**
When interfacing with Hanko Cloud strict CORS policies on certain browsers such as Chrome may cause problems. In order to use this system without overriding browser security policies, the console web app must be associated with a domain name. Within Hanko Cloud it is important to make sure your app url is simply the domain name without any prefixes or suffixes as seen in the screenshot below.


Beyond just this you must also add the domain with the prefix ‘www.’ to the allowed origins in order to ensure compatibility across all modern browsers as seen in the screenshot below.


**Testing:**
The Jest testing suite must be run in the “/app” subfolder of the application in order to run correctly. The web ui, to help visualize the successful tests is not functional, so the command line test coverage is the most effective way to visualize results. This command line visualization is enabled by default. Test results can also be viewed through the github web interface.

# Conclusion

Thank you for choosing StratoSplit as your secure audio simulation and management platform. This user manual was developed to empower your organization with the tools and knowledge needed to confidently deploy, operate, and maintain the system in real-world contexts. StratoSplit was engineered with mission-critical environments in mind, combining modern web technologies with secure authentication, multicast audio streaming, and responsive dashboard controls.We hope this product provides you with good performance, ease of use, and flexibility to adapt to your specific needs.

With best wishes from your StratoSplit development team:

Sam Cain

Nolan Newman

Dallon Jarman

Elliot Hull

While we are all moving on to professional careers, we'd be happy to assist with brief questions in the coming months to help ensure that StratoSplit is successfully integrated and optimized for your operations. We wish you continued success in your mission and are proud to have contributed to your technological capabilities