# Self Hosted Live Streaming and Media Server by Optick

## Customer Instructions for AWS Marketplace Users

*on Ubuntu 26.04 LTS*

**Important access note**

Connect to your instance using your Amazon private key and the **ubuntu** user. Do not try to SSH as root.

### 1. Product overview

This AMI provides a private, self hosted streaming and media stack on AWS. It includes **Ant Media Server Community** for live streaming and RTMP/WebRTC workflows, **Jellyfin** for private media library playback, **Cockpit** for browser based server administration, and an Optick landing page that automatically regenerates with the instance public IP on first launch.

| Component | Purpose |
|---|---|
| Ant Media Server Community | Live streaming, RTMP ingest, WebRTC workflow testing, and streaming application experiments |
| Jellyfin | Private media library server for movies, shows, music, and other media |
| Cockpit | Browser based server administration for system status, storage, networking, services, and logs |
| Nginx landing page | Simple public IP based entry page with links to the included applications |
| First boot automation | Regenerates URLs and first login instructions when a customer launches a new instance |

### 2. First launch checklist

After launching the AMI, wait about one minute for first boot automation to complete. Then connect by SSH:

```
ssh -i your-key.pem ubuntu@NEW_PUBLIC_IP
```

View the generated first login instructions:

```
cat /home/ubuntu/FIRST_LOGIN.txt
```

Show the application URLs at any time:

```
optick-streaming-url
```

Check application health:

```
optick-streaming-status
```

**If a page does not appear immediately**

Wait one minute and refresh the browser. Then run optick-streaming-status from SSH to confirm Docker, Nginx, Cockpit, Ant Media, and Jellyfin are running.

## 3. Application URLs

| Application | URL |
|---|---|
| Optick landing page | http://NEW_PUBLIC_IP/ |
| Ant Media Server | http://NEW_PUBLIC_IP:5080/ |
| Jellyfin | http://NEW_PUBLIC_IP:8096/ |
| Cockpit | https://NEW_PUBLIC_IP:9090/ |

Replace NEW_PUBLIC_IP with the public IPv4 address assigned to your EC2 instance.

## 4. Recommended AWS security group rules

A security group controls inbound and outbound traffic for the EC2 instance. For production use, restrict administrative ports to trusted IP addresses whenever possible.

| Protocol | Port | Source | Purpose |
|---|---|---|---|
| TCP | 22 | Your trusted IP | SSH access with Amazon private key and ubuntu user |
| TCP | 80 | 0.0.0.0/0 or restricted source | Optick landing page and optional HTTP validation |
| TCP | 443 | 0.0.0.0/0 or restricted source | Optional HTTPS when a domain and certificate are configured |
| TCP | 5080 | 0.0.0.0/0 or restricted source | Ant Media web interface over HTTP |
| TCP | 5443 | 0.0.0.0/0 or restricted source | Ant Media HTTPS after SSL is configured |
| TCP | 1935 | Streaming sources only if possible | RTMP ingest |
| UDP | 50000-60000 | Streaming clients or required source range | WebRTC media traffic for Ant Media |
| TCP | 8096 | 0.0.0.0/0 or restricted source | Jellyfin web interface |
| TCP | 9090 | Your trusted IP only | Cockpit server administration |

**Note:** Jellyfin also exposes container port 8920, but it is not required for the standard first launch path unless you separately configure Jellyfin HTTPS.

## 5. Ant Media Server first setup

Open Ant Media Server:

**http://NEW_PUBLIC_IP:5080/**

On first use, create the Ant Media administrator account when prompted. Save the username and password carefully. For RTMP ingest, use TCP port 1935. For browser based WebRTC publishing, configure HTTPS with a domain and make sure UDP 50000 to 60000 is open in the security group.

**WebRTC and HTTPS note**

Public IP access is suitable for setup and initial checks. Browser camera and microphone publishing generally requires a secure HTTPS context, so use a domain and SSL before relying on browser based WebRTC publishing.

## 6. Jellyfin first setup

Open Jellyfin:

**http://NEW_PUBLIC_IP:8096/**

The Jellyfin setup wizard should appear on first use. Create the administrator account and save the password carefully. When adding libraries inside Jellyfin, use these container paths:

| Media type | Path inside Jellyfin | Host path for uploads over SSH |
|---|---|---|
| Movies | /media/movies | /opt/optick-streaming/media/movies |
| TV shows | /media/shows | /opt/optick-streaming/media/shows |
| Music | /media/music | /opt/optick-streaming/media/music |
| Uploads or general media | /media/uploads | /opt/optick-streaming/media/uploads |

To upload media over SSH, copy files into the host paths above as the ubuntu user. Then scan or refresh the matching library in Jellyfin.

## 7. Cockpit server administration

Open Cockpit:

**https://NEW_PUBLIC_IP:9090/**

Cockpit uses the Linux server account. For this AMI, the Cockpit username is ubuntu. Before logging in to Cockpit, connect by SSH and set a local password for the ubuntu user:

**sudo optick-streaming-set-cockpit-password**

Use the password you just created to log in to Cockpit. SSH access should still use your Amazon private key. For security, restrict TCP port 9090 to your trusted IP address.

## 8. Helpful commands

| Command | Purpose |
|---|---|
| optick-streaming-url | Show landing page, Ant Media, Jellyfin, and Cockpit URLs |
| optick-streaming-status | Check Docker, Nginx, Cockpit, containers, local HTTP checks, and listening ports |
| optick-streaming-logs | Show recent Ant Media, Jellyfin, and Nginx logs |
| sudo optick-streaming-set-cockpit-password | Set or reset the ubuntu password used by Cockpit browser login |
| optick-streaming-ssl | Show domain, SSL, and WebRTC setup guidance |
| optick-streaming-ssl yourdomain.com | Check whether a domain points to the current public IP before configuring SSL |

**optick-streaming-url**

**optick-streaming-status**

**optick-streaming-logs**

**sudo optick-streaming-set-cockpit-password**

**optick-streaming-ssl**

**optick-streaming-ssl yourdomain.com**

## 9. Optional domain, SSL, and WebRTC setup

Public IP access is the default path for first launch. If you want browser camera and microphone publishing with WebRTC, configure a domain and SSL.

Recommended sequence:

1. Create a DNS A record pointing your domain to the instance public IPv4 address.
2. Allow TCP 80, TCP 443, TCP 5080, TCP 5443, TCP 1935, and UDP 50000 to 60000 in the AWS security group as appropriate.
3. Run the Optick SSL helper with your domain.
4. Log in to Ant Media and follow its SSL settings workflow.
5. Test the secure Ant Media URL after SSL is enabled.

```
optick-streaming-ssl yourdomain.com
```

## 10. Troubleshooting

| Issue | Recommended action |
|---|---|
| Landing page does not load | Confirm TCP 80 is open in the security group, wait one minute, refresh the browser, and run optick-streaming-status. |
| Ant Media does not load | Confirm TCP 5080 is open. Run optick-streaming-status and optick-streaming-logs. |
| Jellyfin does not load | Confirm TCP 8096 is open. Wait for the container health check to become healthy and run optick-streaming-status. |
| Cockpit does not log in | Run sudo optick-streaming-set-cockpit-password, then log in with username ubuntu and the password you just created. |
| Browser warns about Cockpit certificate | This is expected when opening Cockpit by public IP with a self signed certificate. Proceed only if you trust your own instance. |
| WebRTC publish has camera or microphone issues | Use a domain with HTTPS and confirm UDP 50000 to 60000 is open for the required source range. |

## 11. Important paths and files

| Path | Purpose |
|---|---|
| /opt/optick-streaming | Main application directory |
| /opt/optick-streaming/antmedia/docker-compose.yml | Ant Media Compose file |
| /opt/optick-streaming/jellyfin/docker-compose.yml | Jellyfin Compose file |
| /opt/optick-streaming/media | Host media storage path used by Jellyfin |
| /home/ubuntu/FIRST_LOGIN.txt | Generated first login instructions |
| /opt/optick-streaming/VERSIONS.txt | Installed version inventory |

## 12. Official learning resources

**Ant Media Documentation:** https://docs.antmedia.io/

**Ant Media WebRTC Playback Documentation:** https://docs.antmedia.io/guides/playing-live-stream/webrtc-playback/

**Jellyfin Quick Start:** https://jellyfin.org/docs/general/quick-start/

**Jellyfin Setup Wizard Walkthrough:** https://jellyfin.org/docs/general/post-install/setup-wizard/

**Cockpit Documentation:** https://cockpit-project.org/guide/latest/

**AWS EC2 Security Groups:** https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html