OpenVidu Setup

Complete Installation and Configuration Guide

ChourangiHealth Video Consultation Platform

Version 1.0 | February 2026

1. What is OpenVidu

OpenVidu is an open-source platform that makes it easy to add video call capabilities into web and mobile applications. It is built on top of Kurento Media Server (KMS) which handles all WebRTC media processing. OpenVidu v2.32 was used in this project.

1.1 OpenVidu Architecture

- OpenVidu Server: Java Spring Boot application that manages sessions, connections, and recording via REST API
- Kurento Media Server (KMS): C++ media server that handles WebRTC streams, ICE negotiation, and recording
- COTURN: TURN/STUN server that relays media when direct peer-to-peer connection is not possible
- Nginx: Reverse proxy that handles SSL termination and routes traffic to correct services
- OpenVidu Call: Default web application UI (replaced in this project with custom React frontend)

1.2 Port Usage

- 443 HTTPS: Main entry point via Nginx
- 5443 HTTP: Internal OpenVidu server port
- 3478 UDP/TCP: COTURN TURN/STUN server
- 40000-57000 UDP/TCP: KMS RTP media ports
- 57001-65535 UDP: COTURN relay media ports

2. AWS EC2 Instance Setup

2.1 Instance Configuration

- Instance type: c6a.xlarge (4 vCPUs, 8 GB RAM, AMD EPYC processor)
- Operating system: Ubuntu 24.04 LTS
- Region: eu-west-2 (London)
- Public IP: 18.130.149.212
- Private IP: 172.31.33.5

2.2 AWS Security Group Rules

These inbound rules must be configured before installing OpenVidu. Missing any of these will cause video or TURN to fail.

- 22 TCP: SSH access
- 80 TCP: HTTP (required for Let's Encrypt certificate validation)
- 443 TCP: HTTPS main application traffic
- 3478 TCP and UDP: COTURN TURN and STUN signaling
- 40000-65535 TCP: KMS RTP media ports
- 40000-65535 UDP: KMS RTP and COTURN relay media ports

NOTE: All rules must allow both IPv4 (0.0.0.0/0) and IPv6 (::/0)

3. Domain and SSL

3.1 DuckDNS Domain Setup

A free dynamic DNS domain was registered at duckdns.org. DuckDNS provides free subdomains under duckdns.org that can be pointed to any IP address.

- Domain registered: chourangi.duckdns.org
- Pointed to EC2 public IP: 18.130.149.212
- DuckDNS token saved for automatic updates

3.2 SSL Certificate

OpenVidu automatically provisions a Let's Encrypt SSL certificate during installation when CERTIFICATE_TYPE=letsencrypt is set. Let's Encrypt is a free Certificate Authority. The certificate is auto-renewed by OpenVidu.

4. OpenVidu Installation

4.1 Installation Command

Run the following commands on the EC2 instance as root or with sudo:

```
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh | bash
```

This installs OpenVidu in /opt/openvidu/ and creates all required directories and docker-compose files.

4.2 Main Configuration File

After installation, edit /opt/openvidu/.env with the following values:

```
DOMAIN_OR_PUBLIC_IP=chourangi.duckdns.org
CERTIFICATE_TYPE=letsencrypt
LETSENCRYPT_EMAIL=ashutosh@xtensible.in
OPENVIDU_SECRET=openvidu-v2-32
COTURN_IP=18.130.149.212
COTURN_EXTERNAL_IP=18.130.149.212/172.18.0.2
KMS_PUBLIC_IP=18.130.149.212
OPENVIDU_RECORDING=true
OPENVIDU_RECORDING_PATH=/opt/openvidu/recordings
```

4.3 Start OpenVidu

```
cd /opt/openvidu
./openvidu start
```

First start may take 2-3 minutes as Docker images are pulled. After start, OpenVidu is accessible at https://chourangi.duckdns.org

4.4 OpenVidu Management Commands

```
./openvidu start    # Start all containers
./openvidu stop     # Stop all containers
./openvidu restart  # Restart all containers
./openvidu logs     # Stream logs
./openvidu version  # Show installed version
```

5. Docker Compose Configuration

5.1 Container List

OpenVidu runs as multiple Docker containers managed by docker-compose:

- openvidu-openvidu-server-1: Main OpenVidu Java server
- openvidu-kms-1: Kurento Media Server
- openvidu-coturn-1: COTURN TURN/STUN server
- openvidu-nginx-1: Nginx reverse proxy with SSL
- openvidu-app-1: Default OpenVidu Call application (replaced by custom frontend)

5.2 KMS Container Environment Variables (Critical Fix)

These environment variables must be added to the kms service in /opt/openvidu/docker-compose.yml. Without these, KMS advertises wrong private IP to clients and video fails.

```
environment:
  - KMS_EXTERNAL_IPV4=18.130.149.212
  - KMS_NETWORK_INTERFACES=ens5
  - KMS_ICE_TCP=0
```

NOTE: On AWS EC2, the network interface is ens5, NOT eth0. Using eth0 causes ICE_GATHER_CANDIDATES_ERROR and video completely fails to start.

5.3 COTURN Container Configuration

The COTURN container must use host network mode on AWS EC2. Add these to the coturn service in docker-compose.yml:

```
network_mode: host
command:
  - --listening-ip=0.0.0.0
  - --external-ip=18.130.149.212/172.31.33.5
```

NOTE: Without network_mode: host, COTURN cannot see real client IPs. All connections appear to come from the EC2 own public IP causing TURN relay to fail.

6. Custom Frontend Integration

6.1 How Custom Frontend Works

OpenVidu Nginx serves files from /opt/openvidu/custom-layout/ directory when accessed at the root URL. The React app is built and copied here to replace the default OpenVidu Call application.

6.2 Custom Nginx Location

Create /opt/openvidu/custom-nginx-locations/frontend.conf with:

```
location / {
  root /opt/openvidu/custom-layout;
  try_files $uri $uri/ /index.html;
}
location /api/ {
  proxy_pass http://172.17.0.1:8000;
  proxy_set_header Host $host;
  proxy_set_header X-Real-IP $remote_addr;
}
```

NOTE: 172.17.0.1 is the Docker bridge gateway IP. It routes from inside the Nginx container to the host machine where FastAPI uvicorn is running on port 8000.

7. Recording Configuration

7.1 Enable Recording

Recording requires these settings in /opt/openvidu/.env:

```
OPENVIDU_RECORDING=true
OPENVIDU_RECORDING_PATH=/opt/openvidu/recordings
```

Restart OpenVidu after adding these settings.

7.2 Recording Modes

- COMPOSED: All publisher streams recorded into one single MP4 file in a grid layout. This is like Zoom recording. Used in this project.
- INDIVIDUAL: Each publisher stream recorded into its own separate file in a ZIP archive. More efficient but produces multiple files.

7.3 Recording Storage

- Recordings stored at: /opt/openvidu/recordings/ on EC2
- File naming: {sessionId}/{sessionId}.mp4
- Accessible via: https://chourangi.duckdns.org/openvidu/recordings/{id}/{id}.mp4
- Requires OpenVidu admin credentials (OPENVIDUAPP / openvidu-v2-32) to access directly
- Download via backend proxy endpoint to avoid exposing credentials to browser

8. WebRTC ICE Troubleshooting

8.1 Common ICE Errors

- ICE_CONNECTION_DISCONNECTED: Client browser cannot reach KMS. Usually means KMS is advertising wrong IP.
- ICE_GATHER_CANDIDATES_ERROR: KMS itself cannot gather ICE candidates. Usually means wrong network interface configured.
- TURN relay failing: COTURN not in host network mode or wrong external IP configured.

8.2 Diagnostic Commands

```
# Check COTURN is listening
ss -ulnp | grep 3478

# Check KMS config file inside container
docker exec openvidu-kms-1 cat /etc/kurento/modules/kurento/WebRtcEndpoint.conf.ini

# Check KMS environment variables
docker exec openvidu-kms-1 env | grep -i kms

# View KMS logs
docker logs openvidu-kms-1 --tail 50

# View COTURN logs
docker logs openvidu-coturn-1 --tail 30
```

8.3 Root Cause on AWS EC2

AWS EC2 uses NAT. The public IP 18.130.149.212 is not directly assigned to any network interface. The instance only has the private IP 172.31.33.5 on interface ens5. KMS must be explicitly told its public IP via KMS_EXTERNAL_IPV4 and the correct interface via KMS_NETWORK_INTERFACES=ens5 so it can advertise the right IP to WebRTC clients.