ClearVocals Production Setup Guide

RunPod Backend + Cloudflare Configuration

Complete Deployment Instructions

October 11, 2025

Contents

1 Overview

This guide provides step-by-step instructions to deploy your ClearVocals backend on RunPod with Cloudflare DNS and R2 storage integration.

1.1 Architecture

- **Frontend:** `https://clearvocals.ai`
- **Backend API:** `https://api.clearvocals.ai`
- **R2 Storage:** `https://files.clearvocals.ai`

1.2 Prerequisites

1. Active RunPod account with a GPU pod
2. Cloudflare account (free plan sufficient)
3. Domain added to Cloudflare DNS
4. SSH access to RunPod pod

1.3 Estimated Time

Total setup time: **10-15 minutes**

2 Part 1: RunPod Configuration

2.1 Step 1: SSH into RunPod

Access your RunPod pod via SSH:

```
ssh root@<pod-id>.ssh.runpod.io -p <port>
```

Navigate to your project directory:

```
cd /workspace/runpod-backend
```

2.2 Step 2: Configure Domain

Run the Cloudflare setup script:

```
./setup_cloudflare.sh clearvocals.ai
```

Expected Output:

```
 Configuration saved
Frontend:  https://clearvocals.ai
Backend:  https://api.clearvocals.ai
Files R2:  https://files.clearvocals.ai
```

This creates `.env.proxy` with:

- `FRONTEND_URL=https://clearvocals.ai`
- `BACKEND_URL=https://api.clearvocals.ai`
- `R2_PUBLIC_URL=https://files.clearvocals.ai`
- `NODE_ENV=production`

2.3 Step 3: Load Configuration

Load the proxy configuration:

```
set -a
source .env.proxy
set +a
```

2.4 Step 4: Start Backend Server

Start all services:

```
./runpod_setup.sh
```

Verify services are running:

```
ps aux | grep uvicorn
curl http://localhost:8000/health/ready
```

2.5 Step 5: Get RunPod Proxy URL

Method 1 - From Dashboard:

1. Go to RunPod Dashboard
2. Click on your pod
3. Click `Connect`
4. Find `HTTP Service [8000]`
5. Copy the URL (format: `<pod-id>-8000.proxy.runpod.net`)

Method 2 - From Terminal: The URL is displayed when you deploy your pod. Example:

```
abc123def456-8000.proxy.runpod.net
```

Important: Save this URL for Cloudflare DNS configuration.

3 Part 2: Cloudflare Configuration

3.1 DNS Record Setup

3.1.1 Step 1: Access DNS Dashboard

1. Login to Cloudflare Dashboard
2. Select domain: `clearvocals.ai`
3. Navigate to: `DNS → Records`

3.1.2 Step 2: Add CNAME Record

Click `Add Record` and configure:

Field	**Value**
Type	CNAME
Name	api
Target	`<pod-id>-8000.proxy.runpod.net`
Proxy status	Proxied (Orange cloud)
TTL	Auto

Critical:

- Do NOT include `https://` in Target
- Proxy status MUST be **Proxied** (orange cloud icon)
- Use your actual RunPod URL from Step 5

Click `Save`.

3.1.3 Verification

DNS propagation: 1-5 minutes

Test DNS resolution:

```
nslookup api.clearvocals.ai
```

3.2 R2 Bucket Custom Domain

3.2.1 Step 1: Navigate to R2

1. Cloudflare Dashboard → `R2 Object Storage`
2. Select your bucket
3. Click `Settings` tab

3.2.2 Step 2: Configure Public Access

Scroll to `Public Access` section

3.2.3 Step 3: Add Custom Domain

1. Click `Custom Domains`
2. Click `Connect Domain`
3. Enter: `files.clearvocals.ai`
4. Click `Continue`

Cloudflare automatically:

- Creates DNS record
- Provisions SSL certificate
- Enables CDN caching

Activation time: 2-5 minutes

3.2.4 Multiple Buckets (Optional)

If you have a second bucket, repeat for:

- Bucket 2 domain: `media.clearvocals.ai`

3.3 SSL/TLS Configuration

3.3.1 Step 1: Access SSL Settings

Navigate to: `SSL/TLS → Overview`

3.3.2 Step 2: Set Encryption Mode

Select: `Full (strict)` **(Recommended)**

Mode	**Description**
Full (strict)	End-to-end encryption with validated certificate
Full	End-to-end encryption (accepts self-signed)

Click `Save`.

3.4 Additional Security (Optional)

3.4.1 CORS Headers

Navigate to: `Rules → Transform Rules → Modify Response Header`

Create rule:

- **Name:** Add CORS Headers
- **Match:** Hostname equals `api.clearvocals.ai`
- **Action:** Set response headers
  - `Access-Control-Allow-Origin: https://clearvocals.ai`
  - `Access-Control-Allow-Credentials: true`

4 Part 3: Testing & Verification

4.1 Backend Health Check

```
curl https://api.clearvocals.ai/health/ready
```

Expected Response:

```
{
  "status": "ok",
  "mongodb": "ok",
  "redis": "ok",
  "service": "ClearVocals API"
}
```

4.2 R2 Storage Access

Wait 5 minutes after R2 domain setup, then:

```
curl -I https://files.clearvocals.ai
```

Expected: `200 OK` or `404 Not Found` (both indicate DNS works)

4.3 CORS Test

From browser console on `https://clearvocals.ai`:

```
fetch('https://api.clearvocals.ai/health/ready')
  .then(r => r.json())
  .then(console.log)
```

Expected: No CORS errors

4.4 SSL Certificate

Open in browser:

- `https://api.clearvocals.ai`
- Check for valid SSL certificate (green padlock)

5 Part 4: Frontend Configuration

5.1 Environment Variables

Update your frontend `.env` file:

5.1.1 Vite (Vue/React)

```
VITE_API_URL=https://api.clearvocals.ai
VITE_FILES_URL=https://files.clearvocals.ai
```

5.1.2 Next.js

```
NEXT_PUBLIC_API_URL=https://api.clearvocals.ai
NEXT_PUBLIC_FILES_URL=https://files.clearvocals.ai
```

5.1.3 Create React App

```
REACT_APP_API_URL=https://api.clearvocals.ai
REACT_APP_FILES_URL=https://files.clearvocals.ai
```

5.2 API Integration

Example API call:

```
const API_URL = import.meta.env.VITE_API_URL;

const response = await fetch(`${API_URL}/api/auth/login`, {
  method: 'POST',
  credentials: 'include',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({ email, password })
});
```

6 Troubleshooting

6.1 502 Bad Gateway

Causes:

- Backend not running
- Incorrect RunPod URL in DNS
- Port mismatch

Solutions:

```
# Check backend status
ps aux | grep uvicorn

# Restart backend
./runpod_setup.sh

# Verify port 8000 is open
netstat -tulpn | grep 8000
```

6.2 CORS Errors

Causes:

- Proxy config not loaded
- Wrong domain in CORS settings

Solutions:

```
# Verify config
cat .env.proxy

# Reload config
set -a; source .env.proxy; set +a

# Restart backend
./runpod_setup.sh
```

6.3 R2 Domain Not Working

Causes:

- DNS not propagated (wait 5-10 min)
- Public access not enabled
- Domain typo

Solutions:

```
# Check DNS
nslookup files.clearvocals.ai

# Wait and retry
sleep 300
curl -I https://files.clearvocals.ai
```

6.4 SSL Certificate Errors

Solution: Ensure SSL mode is set to `Full (strict)` in Cloudflare.

7 Maintenance

7.1 Restarting Backend

```
cd /workspace/runpod-backend
set -a; source .env.proxy; set +a
./runpod_setup.sh
```

7.2 Viewing Logs

```
# API logs
tail -f logs/info.log

# Worker logs
tail -f logs/workers.log

# All logs
tail -f logs/*.log
```

7.3 Monitoring

Check health endpoint regularly:

```
watch -n 30 'curl -s https://api.clearvocals.ai/health/ready'
```

8 Deployment Checklist

8.1 RunPod

SSH access verified

`./setup_cloudflare.sh clearvocals.ai` executed

`.env.proxy` file created

Environment loaded: `source .env.proxy`

`./runpod_setup.sh` running successfully

Backend responding on port 8000

RunPod proxy URL copied

8.2 Cloudflare DNS

CNAME record `api` created

Target set to RunPod URL

Proxy status: Proxied (orange cloud)

DNS resolves correctly

8.3 Cloudflare R2

Custom domain `files.clearvocals.ai` added

Public access enabled

DNS record auto-created

Domain accessible (wait 5 min)

8.4 SSL/TLS

Mode set to `Full (strict)`

Certificates valid

8.5 Testing

Backend health check: `https://api.clearvocals.ai/health/ready`

R2 access: `https://files.clearvocals.ai`

CORS working from frontend

SSL certificates valid

8.6 Frontend

Environment variables updated

API calls using correct URL

Credentials: include for CORS

9 Summary

9.1 Final Configuration

Service	**URL**
Frontend	`https://clearvocals.ai`
Backend API	`https://api.clearvocals.ai`
R2 Storage	`https://files.clearvocals.ai`

9.2 Production Ready

Your deployment is now:

- Secured with SSL/TLS encryption
- Protected by Cloudflare CDN & DDoS
- Custom domains configured
- CORS properly configured
- Scalable and production-ready

9.3 Support

For issues, check:

- Backend logs: `logs/info.log`
- Worker logs: `logs/workers.log`
- Health endpoint: `/health/ready`