🎙️ VibeMaster Voice Server - Complete Setup Reference

Date: November 1, 2025

Status: ✅ Production Ready

Server: DigitalOcean Droplet (Ubuntu 24)

Domain: audio.netrunners.xyz

🎯 System Architecture

```
audio.netrunners.xyz (Nginx HTTPS on ports 80/443)
    │
    ├──> /vibemaster/*  → VibeMaster Server (Node.js on port 3001)
    │     ├──> /vibemaster/health  (Health check)
    │     ├──> /vibemaster/tts     (Text-to-Speech via ElevenLabs)
    │     ├──> /vibemaster/stt     (Speech-to-Text via ElevenLabs)
    │     └──> /vibemaster/claude  (AI Chat via Claude API)
    │
    └──> /*  → MidniteMansion Server (Node.js on port 3000)
          ├──> /health
          ├──> /tts
          ├──> /stt
          └──> /claude
```

📡 API Endpoints

VibeMaster Endpoints

Base URL: `https://audio.netrunners.xyz/vibemaster`

Health Check

```bash
```

```
GET https://audio.netrunners.xyz/vibemaster/health

Response:
{
  "status": "ok",
  "service": "VibeMaster Voice AI Proxy",
  "timestamp": "2025-11-01T05:06:16.696Z",
  "endpoints": {
    "tts": "/vibemaster/tts",
    "stt": "/vibemaster/stt",
    "claude": "/vibemaster/claude"
  }
}
```

Text-to-Speech

bash

```
POST https://audio.netrunners.xyz/vibemaster/tts
Content-Type: application/json

{
  "text": "Your text here",
  "apiKey": "YOUR_ELEVENLABS_KEY",
  "voiceId": "21m00Tcm4TlvDq8ikWAM" // Optional
}

Response:
{
  "success": true,
  "audioUrl": "https://audio.netrunners.xyz/vibemaster/audio/audio_123.mp3",
  "duration": 2.5,
  "fileSize": 45678,
  "filename": "audio_123.mp3"
}
```

Speech-to-Text

bash

```
POST https://audio.netrunners.xyz/vibemaster/stt
Content-Type: application/json

{
  "audioData": "BASE64_ENCODED_AUDIO",
  "apiKey": "YOUR_ELEVENLABS_KEY",
  "language": "en" // Optional
}

Response:
{
  "success": true,
  "text": "Transcribed text here",
  "language": "en"
}
```

Claude AI Chat

bash

```
POST https://audio.netrunners.xyz/vibemaster/claude
Content-Type: application/json

{
  "message": "Your message here",
  "apiKey": "YOUR_CLAUDE_KEY",
  "conversationHistory": [], // Optional
  "systemPrompt": "Your system prompt" // Optional
}

Response:
{
  "success": true,
  "response": "Claude's response here",
  "conversationHistory": [...]
}
```

💻 Server Management

PM2 Commands

bash

```
# View status
pm2 status

# View logs
pm2 logs vibemaster-proxy
pm2 logs elevenlabs-proxy  # MidniteMansion

# Restart servers
pm2 restart vibemaster-proxy
pm2 restart elevenlabs-proxy

# Stop servers
pm2 stop vibemaster-proxy
pm2 stop elevenlabs-proxy

# Start servers
pm2 start vibemaster-proxy
pm2 start elevenlabs-proxy

# Monitor
pm2 monit
```

Server Locations

VibeMaster:

- Directory: `/root/vibemaster-proxy/`
- Config: `/root/vibemaster-proxy/server.js`
- Port: 3001
- PM2 Name: `vibemaster-proxy`

MidniteMansion:

- Directory: `/root/elevenlabs-proxy/`
- Config: `/root/elevenlabs-proxy/server.js`
- Port: 3000
- PM2 Name: `elevenlabs-proxy`

Nginx Configuration

Config File: `/etc/nginx/sites-available/elevenlabs-proxy`

```
nginx
```

```
server {
    server_name audio.netrunners.xyz;

    # VibeMaster routes
    location /vibemaster/ {
        proxy_pass http://localhost:3001/vibemaster/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # Default route (MidniteMansion)
    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/audio.netrunners.xyz/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/audio.netrunners.xyz/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
}

server {
    if ($host = audio.netrunners.xyz) {
        return 301 https://$host$request_uri;
    }

    listen 80;
    server_name audio.netrunners.xyz;
    return 404;
}
```

Nginx Commands:

```bash
# Test config
sudo nginx -t

# Reload
sudo systemctl reload nginx

# Restart
sudo systemctl restart nginx

# Status
sudo systemctl status nginx
```

🔧 Troubleshooting

Check Server Status

```bash
# PM2 status
pm2 status

# Direct test (bypass Nginx)
curl http://localhost:3001/vibemaster/health
curl http://localhost:3000/health

# HTTPS test (through Nginx)
curl https://audio.netrunners.xyz/vibemaster/health
curl https://audio.netrunners.xyz/health
```

Check Logs

```bash
# Server logs
pm2 logs vibemaster-proxy --lines 50
pm2 logs elevenlabs-proxy --lines 50

# Nginx logs
sudo tail -50 /var/log/nginx/error.log
sudo tail -50 /var/log/nginx/access.log
```

Check Ports

```
bash

# What's listening
sudo netstat -tlnp | grep -E ':(3000|3001|80|443)'
```

Restart Everything

```
bash

# Restart both servers
pm2 restart all

# Restart Nginx
sudo systemctl restart nginx

# Check status
pm2 status
sudo systemctl status nginx
```

💾 Backup & Recovery

Backup Server Code

```
bash

# VibeMaster
cd ~
tar -czf vibemaster-backup-$(date +%Y%m%d).tar.gz vibemaster-proxy/

# MidniteMansion
tar -czf elevenlabs-backup-$(date +%Y%m%d).tar.gz elevenlabs-proxy/
```

Backup Nginx Config

```
bash

sudo cp /etc/nginx/sites-available/elevenlabs-proxy ~/nginx-backup-$(date +%Y%m%d).conf
```

🔑 API Keys Required

ElevenLabs API Key:

- Used for: TTS and STT
- Get from: https://elevenlabs.io

- Format: `sk_...`

Claude API Key:

- Used for: AI chat responses
- Get from: https://console.anthropic.com
- Format: `sk-ant-...`

⚠️ **Security:** Never commit API keys to Git. Use environment variables or config files in `.gitignore`.

🚀 Decentraland Integration

Configuration in VibeMaster Project

typescript

```typescript
// src/config.ts
export const VIBEMASTER_CONFIG = {
  proxyUrl: 'https://audio.netrunners.xyz/vibemaster',

  endpoints: {
    tts: 'https://audio.netrunners.xyz/vibemaster/tts',
    stt: 'https://audio.netrunners.xyz/vibemaster/stt',
    claude: 'https://audio.netrunners.xyz/vibemaster/claude',
    health: 'https://audio.netrunners.xyz/vibemaster/health'
  },

  // API keys - load from environment or UI
  elevenLabsApiKey: '', // Set at runtime
  claudeApiKey: '',     // Set at runtime
}
```

Example Usage in Decentraland

typescript

```
// audioManager.ts
import { VIBEMASTER_CONFIG } from './config'

export class AudioManager {
  async playTextAsAudio(text: string) {
    const response = await fetch(`${VIBEMASTER_CONFIG.proxyUrl}/tts`, {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({
        text: text,
        apiKey: VIBEMASTER_CONFIG.elevenLabsApiKey
      })
    })

    const data = await response.json()
    // Play audio from data.audioUrl
  }
}
```

📊 Monitoring

Health Checks

```bash
# Quick health check script
#!/bin/bash
echo "VibeMaster: $(curl -s https://audio.netrunners.xyz/vibemaster/health | jq -r .status)"
echo "MidniteMansion: $(curl -s https://audio.netrunners.xyz/health | jq -r .status)"
```

Set Up Monitoring (Optional)

- **UptimeRobot**: Free monitoring service
- **PM2 Plus**: Advanced PM2 monitoring
- **Custom webhook**: Call health endpoints periodically

🎯 Success Criteria

✅ Both services respond to health checks
✅ VibeMaster TTS generates audio files
✅ VibeMaster STT transcribes voice
✅ Claude AI responds to messages
✅ MidniteMansion production unchanged

✅ HTTPS working for all endpoints
✅ PM2 keeps services running
✅ Nginx routes requests correctly

📞 Emergency Contacts

If Something Breaks:

1. Check PM2: `pm2 status`
2. Check logs: `pm2 logs`
3. Check Nginx: `sudo systemctl status nginx`
4. Restart everything: `pm2 restart all && sudo systemctl restart nginx`

Rollback Plan:

- VibeMaster is isolated - stopping it won't affect MidniteMansion
- `pm2 stop vibemaster-proxy` to disable VibeMaster
- MidniteMansion continues running normally

🎉 You're All Set!

Your VibeMaster voice server is now running alongside your production MidniteMansion game without any interference. Both services are isolated, secure, and ready for development.

Next Steps:

1. Test TTS endpoint with actual voice generation
2. Test STT endpoint with audio recording
3. Test Claude endpoint with conversation
4. Integrate into VibeMaster Decentraland project
5. Build amazing voice-driven games!

Document Version: 1.0
Last Updated: November 1, 2025
Status: Production Ready ✅