Quick Deployment Guide - Sports Bar TV Controller

Intel NUC13ANHi5 Production Deployment

Target System: Intel NUC13ANHi5 (i5-1340P, 12 cores, 16GB RAM)
Estimated Time: 2-3 hours
Difficulty: Intermediate

Pre-Deployment Checklist

Before You Start

- [ ] New NUC13ANHi5 system is unboxed and connected
- [ ] Ubuntu Server 22.04 LTS is installed on new system
- [ ] You have SSH access to old system (135.131.39.26:223)
- [ ] You have the SSH password: `6809233DjD$$$`
- [ ] You have a backup of critical data (recommended)
- [ ] You have 2-3 hours of uninterrupted time

What You'll Need

- Ethernet cable (connected to new system)
- Monitor, keyboard, mouse (for initial setup)
- SSH client (Terminal on Mac/Linux, PuTTY on Windows)
- This guide

Step-by-Step Deployment

Phase 1: Initial System Setup (30 minutes)

Step 1: Install Ubuntu Server

1. Download Ubuntu Server 22.04 LTS from https://ubuntu.com/download/server
2. Create bootable USB drive
3. Boot NUC13 from USB
4. Follow installation wizard:
   - Hostname: `sports-bar-nuc13`
   - Username: `ubuntu`
   - Enable OpenSSH server
   - Install security updates

Step 2: First Boot Configuration

```
# SSH into new system
ssh ubuntu@<NEW_SYSTEM_IP>

# Update system
sudo apt update && sudo apt upgrade -y

# Set timezone (adjust to your location)
sudo timedatectl set-timezone America/New_York

# Reboot
sudo reboot
```

✓ **Checkpoint**: System is updated and rebooted

Phase 2: Clone Repository and Run Setup (45 minutes)

Step 3: Clone Repository

```
# SSH back into system after reboot
ssh ubuntu@<NEW_SYSTEM_IP>

# Clone repository
cd ~
git clone https://github.com/dfultonthebar/Sports-Bar-TV-Controller.git
cd Sports-Bar-TV-Controller

# Switch to deployment branch
git checkout production-deployment-nuc13

# Make scripts executable
chmod +x scripts/*.sh
```

✓ **Checkpoint**: Repository cloned and scripts are executable

Step 4: Run System Setup

```
# Run system setup script
./scripts/system-setup.sh
```

What happens:
- Installs Node.js, PM2, PostgreSQL, Ollama
- Configures Intel Iris Xe graphics
- Sets up monitoring tools
- Configures firewall

Expected output:

```
[✓] System setup completed successfully!
[!] IMPORTANT: Please reboot the system to apply kernel parameters for GPU optimiza-
tion.
```

⚠️ **IMPORTANT**: Reboot now!

```
sudo reboot
```

✓ **Checkpoint**: System setup complete, system rebooted

Phase 3: Configure Ollama (30 minutes)

Step 5: Setup Ollama

```
# SSH back in after reboot
ssh ubuntu@<NEW_SYSTEM_IP>
cd ~/Sports-Bar-TV-Controller

# Run Ollama setup
./scripts/ollama-setup.sh
```

What happens:
- Configures Ollama for Intel GPU
- Optimizes for 12-core CPU
- Downloads AI models (llama3.2:3b, qwen2.5:3b)
- Creates monitoring scripts

Expected output:

```
[✓] Ollama setup completed successfully!
Ollama Configuration Summary:
  - Service: Running on 0.0.0.0:11434
  - Models: llama3.2:3b, qwen2.5:3b
  - CPU Threads: 10 (optimized for 12-core CPU)
  - Intel GPU: Enabled (Iris Xe)
```

Step 6: Verify Ollama

```
# Check Ollama status
systemctl status ollama

# List models
ollama list

# Test model (should respond in a few seconds)
ollama run llama3.2:3b "Hello, respond with OK if working"
```

✓ **Checkpoint**: Ollama is running and models are loaded

Phase 4: Deploy Application (30 minutes)

Step 7: Deploy Application

```
cd ~/Sports-Bar-TV-Controller

# Run application deployment
./scripts/app-deploy.sh
```

What happens:
- Clones repository to /opt/sports-bar-tv
- Installs dependencies
- Creates PostgreSQL database
- Generates .env file
- Builds Next.js application
- Configures PM2 with 10 instances
- Starts application

Expected output:

```
[✓] Application deployed successfully!
Application URL: http://localhost:3000
```

Step 8: Verify Application

```
# Check PM2 status
pm2 status

# Should show:
# ┌─────┬──────────────────┬─────────┬─────┬────────┬──────────┐
# │ id  │ name             │ mode    │ ↺   │ status │ cpu      │
# ├─────┼──────────────────┼─────────┼─────┼────────┼──────────┤
# │ 0   │ sports-bar-tv    │ cluster │ 0   │ online │ 0%       │
# └─────┴──────────────────┴─────────┴─────┴────────┴──────────┘

# Test application
curl http://localhost:3000

# Should return HTML content
```

✓ **Checkpoint**: Application is running

Phase 5: Migrate Data (45 minutes)

Step 9: Run Data Migration

```
cd ~/Sports-Bar-TV-Controller

# Run migration script
./scripts/data-migration.sh
```

You will be prompted:

```
Do you want to continue? (yes/no):
```

Type `yes` and press Enter

You will be asked for SSH password:

```
ubuntu@135.131.39.26's password:
```

Enter: `6809233DjD$$$`

What happens:
- Connects to old system
- Backs up PostgreSQL database
- Backs up environment variables
- Backs up knowledge base
- Backs up PM2 configuration
- Restores everything to new system
- Restarts application

Expected output:

```
[✓] Migration completed successfully!
Backup location: ~/migration-backup-YYYYMMDD-HHMMSS
```

Step 10: Verify Migration

```
# Check database
sudo -u postgres psql -d sportsbar_tv -c "SELECT COUNT(*) FROM users;"

# Should show number of users from old system

# Check application logs
pm2 logs sports-bar-tv --lines 20

# Should show no errors
```

✓ **Checkpoint**: Data migration complete

Phase 6: Performance Setup (20 minutes)

Step 11: Configure Performance Monitoring

```
cd ~/Sports-Bar-TV-Controller

# Run performance setup
./scripts/performance-setup.sh
```

What happens:
- Optimizes PostgreSQL for 12-core CPU
- Creates monitoring scripts

- Sets up automated reports
- Configures weekly optimization

Expected output:

```
[✓] Performance monitoring setup completed!
```

Step 12: Run Initial Performance Check

```
# Run performance monitor
~/monitor-performance.sh
```

Review the output:
- CPU usage should be low (< 20%)
- Memory usage should be reasonable (< 8GB)
- All services should be running
- No critical errors

✓ **Checkpoint**: Performance monitoring configured

Phase 7: Final Configuration (15 minutes)

Step 13: Update Environment Variables

```
# Edit environment file
nano /opt/sports-bar-tv/.env
```

Update these values:

```
# Change database password (recommended)
DATABASE_URL="postgresql://sportsbar:NEW_SECURE_PASSWORD@localhost:5432/sportsbar_tv"

# Update API URL with your server IP
NEXT_PUBLIC_API_URL=http://YOUR_SERVER_IP:3000

# Add external API keys if you have them
YOUTUBE_API_KEY=your_key_here
TWITCH_CLIENT_ID=your_client_id_here
TWITCH_CLIENT_SECRET=your_client_secret_here
```

Save and exit (Ctrl+X, Y, Enter)

If you changed the database password:

```
# Update PostgreSQL password
sudo -u postgres psql
ALTER USER sportsbar WITH PASSWORD 'NEW_SECURE_PASSWORD';
\q
```

Restart application:

```
pm2 restart sports-bar-tv
```

Step 14: Test Everything

```
# Test homepage
curl http://localhost:3000

# Test AI chat
curl -X POST http://localhost:3000/api/chat \
  -H "Content-Type: application/json" \
  -d '{"message": "Hello, test message"}'

# Check PM2 status
pm2 status

# Check Ollama status
systemctl status ollama

# Check PostgreSQL status
sudo systemctl status postgresql
```

✓ **Checkpoint**: All services configured and tested

Post-Deployment Verification

Quick Health Check

```
# Run this command to verify everything
~/monitor-performance.sh
```

What to look for:
- ✅ CPU usage < 20%
- ✅ Memory usage < 8GB
- ✅ All PM2 instances online
- ✅ PostgreSQL running
- ✅ Ollama running
- ✅ No critical errors in logs

Browser Testing

1. Open browser to: `http://YOUR_SERVER_IP:3000`
2. Verify homepage loads
3. Test navigation
4. Test TV control interface
5. Test AI chat (should respond in 2-5 seconds)
6. Test streaming integrations

What to Do If Something Goes Wrong

Application Won’t Start

```
# Check logs
pm2 logs sports-bar-tv --err --lines 50

# Restart application
pm2 restart sports-bar-tv

# If still failing, check PostgreSQL
sudo systemctl status postgresql
sudo systemctl restart postgresql
```

Ollama Not Responding

```
# Check Ollama status
systemctl status ollama

# Restart Ollama
sudo systemctl restart ollama

# Check logs
sudo journalctl -u ollama -n 50

# Test Ollama
ollama list
ollama run llama3.2:3b "test"
```

Database Connection Errors

```
# Check PostgreSQL
sudo systemctl status postgresql

# Verify database exists
sudo -u postgres psql -l | grep sportsbar_tv

# Check connection
sudo -u postgres psql -d sportsbar_tv -c "SELECT 1;"
```

High CPU or Memory Usage

```
# Check what's using resources
htop

# Reduce PM2 instances if needed
pm2 scale sports-bar-tv 8

# Check for runaway processes
ps aux | sort -nrk 3,3 | head -n 10
```

Need to Rollback

```
# Stop new system
pm2 stop sports-bar-tv
sudo systemctl stop ollama
sudo systemctl stop postgresql

# On old system (135.131.39.26:223)
ssh -p 223 ubuntu@135.131.39.26
cd ~/Sports-Bar-TV-Controller
pm2 restart all
```

Useful Commands Reference

PM2 Commands

```
pm2 status                    # View status
pm2 logs sports-bar-tv        # View logs
pm2 monit                     # Monitor resources
pm2 restart sports-bar-tv     # Restart app
pm2 reload sports-bar-tv      # Zero-downtime reload
pm2 stop sports-bar-tv        # Stop app
pm2 start sports-bar-tv       # Start app
```

Ollama Commands

```
ollama list                   # List models
ollama run MODEL "prompt"     # Test model
systemctl status ollama       # Check status
sudo systemctl restart ollama # Restart service
```

PostgreSQL Commands

```
sudo systemctl status postgresql              # Check status
sudo -u postgres psql -d sportsbar_tv        # Connect to DB
sudo -u postgres psql -c "SELECT version();" # Check version
```

Monitoring Commands

```
~/monitor-performance.sh      # Full performance report
htop                          # Interactive process viewer
pm2 monit                     # PM2 resource monitor
df -h                         # Disk space
free -h                       # Memory usage
```

Next Steps After Deployment

Immediate (First 24 Hours)

1. **Monitor Performance**
   ```bash
   # Check every few hours
   ~/monitor-performance.sh
   ```

2. **Review Logs**
   ```bash
   pm2 logs sports-bar-tv
   ```

3. **Test All Features**
   - Homepage
   - TV controls
   - AI chat
   - Streaming integrations

First Week

1. **Fine-tune Performance**
   - Adjust PM2 instances if needed
   - Optimize database queries
   - Monitor resource usage

2. **Set Up Backups**
   - Configure automated backups
   - Test backup restoration

3. **Configure SSL** (if needed)
   ```bash
   sudo apt install certbot python3-certbot-nginx
   sudo certbot --nginx -d your-domain.com
   ```

Ongoing

1. **Weekly Maintenance**
   - Review performance reports
   - Check for updates
   - Clean old logs

2. **Monthly Maintenance**
   - Full system update
   - Database optimization
   - Review security

Performance Expectations

Expected Metrics on NUC13ANHi5

Metric	Target	Your System
Homepage Load	< 500ms	_____
API Response	< 200ms	_____
AI Chat Response	< 3s	_____
CPU Usage (idle)	< 10%	_____
Memory Usage	< 8GB	_____
Database Query	< 50ms	_____

Fill in “Your System” column after deployment to track performance

---

Support and Resources

Documentation

- Full Deployment Guide: `docs/PRODUCTION_DEPLOYMENT.md`
- GitHub Repository: https://github.com/dfultonthebar/Sports-Bar-TV-Controller
- Deployment Branch: `production-deployment-nuc13`

Backup Locations

- Migration Backup: `~/migration-backup-YYYYMMDD-HHMMSS/`
- Performance Reports: `~/performance-reports/`
- Application Logs: `~/logs/sports-bar-tv/`

Emergency Contacts

- Old System: 135.131.39.26:223 (keep online for 30 days)
- New System: [Your IP here]

Deployment Checklist

Use this checklist to track your progress:

- [ ] Phase 1: Initial system setup complete
- [ ] Phase 2: Repository cloned and system setup run
- [ ] Phase 3: Ollama configured and tested
- [ ] Phase 4: Application deployed and running
- [ ] Phase 5: Data migrated from old system
- [ ] Phase 6: Performance monitoring configured

- [ ] Phase 7: Environment variables updated
- [ ] Post-deployment verification complete
- [ ] Browser testing successful
- [ ] Performance metrics recorded
- [ ] Backup strategy configured
- [ ] Old system kept online as backup

Deployment Date: __
Deployed By: __
New System IP: __
Notes: __

Good luck with your deployment! 🚀

If you encounter any issues not covered in this guide, refer to the full deployment guide in `docs/PRODUCTION_DEPLOYMENT.md` or the troubleshooting section above.