Python Scraper - Quick Deployment Checklist

📋 Pre-Deployment

AWS Setup (REQUIRED)

- [ ] Create AWS account
- [ ] Create IAM user: `basketball-scraper`
- [ ] Get access keys (save securely):
- `AWS_ACCESS_KEY_ID`
- `AWS_SECRET_ACCESS_KEY`
- [ ] Create S3 bucket: `basketball-shooters-media`
- [ ] Configure bucket for public read access
- [ ] Test bucket access

Environment Configuration

- [ ] Update `.env` with AWS credentials
- [ ] Update `NEXTJS_API_URL` with frontend URL
- [ ] Verify `DATABASE_URL` is correct
- [ ] Save `API_SECRET_KEY` for frontend: `MSR9VABa1ETXlBkLoXzGpLfnGXsNDQm5C7VcrM-GnjI`

🚀 Render Deployment

Option A: Update Existing Service

- [ ] Go to `basketball-analysis-backend` on Render
- [ ] Update Root Directory: `python-scraper`
- [ ] Update Build Command: `pip install -r requirements.txt`
- [ ] Update Start Command: `gunicorn app:app --bind 0.0.0.0:$PORT --workers 2 --timeout 120`
- [ ] Add all environment variables (see full list below)
- [ ] Deploy

Option B: Create New Service

- [ ] Create new Web Service on Render
- [ ] Name: `basketball-scraper-service`
- [ ] Connect GitHub repo
- [ ] Root Directory: `python-scraper`
- [ ] Runtime: Python 3
- [ ] Add environment variables
- [ ] Create service
- [ ] Delete old service after testing

🌱 Seed Database

After deployment, run these commands:

```
SCRAPER_URL="https://your-scraper.onrender.com"
API_KEY="MSR9VABa1ETXlBkLoXzGpLfnGXsNDQm5C7VcrM-GnjI"

# 1. Scrape NBA players
curl -X POST "$SCRAPER_URL/api/scrape/nba" \
  -H "X-API-Key: $API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"limit": 100}'

# 2. Scrape historical players
curl -X POST "$SCRAPER_URL/api/scrape/historical" \
  -H "X-API-Key: $API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"limit": 50}'

# 3. Download images
curl -X POST "$SCRAPER_URL/api/scrape/images" \
  -H "X-API-Key: $API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"limit": 50}'

# 4. Verify
curl "$SCRAPER_URL/api/shooters?limit=10"
```

✅ Environment Variables (Render)

Required variables to set in Render dashboard:

```
DATABASE_URL=postgresql://
role_98aaf8ef8:A0YpOM7klQCJsj6RHvTtg2wgXkzmmGoT@db-98aaf8ef8.db003.hosteddb.reai.io:
5432/98aaf8ef8?connect_timeout=15

API_SECRET_KEY=MSR9VABa1ETXlBkLoXzGpLfnGXsNDQm5C7VcrM-GnjI

AWS_ACCESS_KEY_ID=[Your AWS Key]
AWS_SECRET_ACCESS_KEY=[Your AWS Secret]
AWS_REGION=us-west-2
S3_BUCKET_NAME=basketball-shooters-media

NEXTJS_API_URL=[Your Frontend URL]

DEBUG=false
FLASK_ENV=production

BACKUP_BUCKET=basketball-shooters-media
LOCAL_BACKUP_DIR=/tmp/db_backups
BACKUP_ENCRYPTION_ENABLED=false
```

🔍 Testing

```
# Health check
curl https://your-scraper.onrender.com/health

# Expected: {"status":"healthy","database":"connected"}
```

📚 Full Documentation

See `/home/ubuntu/basketball_app/PYTHON_SCRAPER_DEPLOYMENT.md` for:
- Detailed AWS setup instructions
- Complete API reference
- Troubleshooting guide
- Security best practices
- Monitoring and logging
- Scheduled scraping setup

🆘 Quick Troubleshooting

Database connection failed?
- Check DATABASE_URL in Render environment

S3 upload failed?
- Verify AWS credentials
- Check bucket permissions

Module not found?
- Clear build cache and redeploy

Service timeout?
- Increase timeout in Procfile to 180
- Reduce scrape limits

API Key: `MSR9VABa1ETXlBkLoXzGpLfnGXsNDQm5C7VcrM-GnjI`
Save this key - you’ll need it for frontend integration!