



Deploy Without Database - Quick Start

PROBLEM SOLVED

The scraper can now be deployed **WITHOUT** a DATABASE_URL. The database connection is created **lazily** (only when needed).

What Changed

Before (FAILED):

```
# database.py - Line 14
engine = create_engine(DATABASE_URL) # ← Connects IMMEDIATELY at import
# Result: Deployment fails if DATABASE_URL is invalid/inaccessible
```

After (WORKS):

```
# database.py - Lazy connection
_engine = None # ← No connection at import

def get_engine():
    if _engine is None:
        _engine = create_engine(DATABASE_URL) # ← Only connects when called
    return _engine
```

Deployment Steps for Render

Step 1: Environment Variables on Render

REQUIRED:

```
API_SECRET_KEY=your-secret-key-here
```

OPTIONAL (Leave blank for minimal deployment):

```
# DATABASE_URL - NOT REQUIRED for deployment
# AWS_ACCESS_KEY_ID - NOT REQUIRED for deployment
# AWS_SECRET_ACCESS_KEY - NOT REQUIRED for deployment
# S3_BUCKET_NAME - NOT REQUIRED for deployment
```

Step 2: Build & Start Commands

```
# Build Command (default):
pip install -r requirements.txt

# Start Command (default):
gunicorn app:app --bind 0.0.0.0:$PORT --workers 2 --timeout 120
```

Step 3: Deploy

Click “Deploy” on Render. The service will:

- Start successfully
- Respond to health checks
- Show status: `{"status": "healthy"}`

Health Check Response

With `DATABASE_URL` Configured:

```
{
  "status": "healthy",
  "service": "Basketball Shooter Scraper",
  "database": {
    "configured": true,
    "connected": true,
    "error": null,
    "note": "Database ready"
  },
  "s3_storage": {
    "enabled": false,
    "note": "Image uploads will be skipped"
  },
  "backup_service": {
    "enabled": false,
    "note": "Database backups disabled"
  },
  "timestamp": "2025-12-13T10:30:00"
}
```

Without DATABASE_URL (Minimal Deployment):

```
{
  "status": "healthy",
  "service": "Basketball Shooter Scraper",
  "database": {
    "configured": false,
    "connected": null,
    "error": null,
    "note": "Database required for scraping operations"
  },
  "s3_storage": {
    "enabled": false,
    "note": "Image uploads will be skipped"
  },
  "backup_service": {
    "enabled": false,
    "note": "Database backups disabled"
  },
  "timestamp": "2025-12-13T10:30:00"
}
```

When to Add DATABASE_URL

Option 1: Never (Use API Only)

- Deploy without DATABASE_URL
- Service responds to health checks
- Database operations return clear error messages
- Useful for testing/staging

Option 2: External Database

- Create PostgreSQL on:
- **Supabase** (free tier)
- **Neon** (free tier)
- **Railway** (free tier)
- **Render PostgreSQL** (free tier)
- Add DATABASE_URL to Render environment
- Restart service
- Database operations will work

Option 3: Abacus AI Database (Recommended)

- Run scraper from **WITHIN** Abacus AI environment
- Internal database is accessible
- Best performance
- No external hosting needed

Testing the Fix

Test 1: Deploy Without DATABASE_URL

```
curl https://your-scraper.onrender.com/health
# Should return: { "status": "healthy", "database": { "configured": false } }
```

Test 2: Try to Scrape Without DATABASE_URL

```
curl -X POST https://your-scraper.onrender.com/api/scrape/nba \
-H "X-API-Key: your-secret-key"

# Should return:
# {
#   "error": "Database operation failed",
#   "details": "DATABASE_URL not configured. Set DATABASE_URL to enable scraping."
# }
```

Test 3: Add DATABASE_URL and Scrape

```
# Add DATABASE_URL on Render
# Restart service
# Try again:
curl -X POST https://your-scraper.onrender.com/api/scrape/nba \
-H "X-API-Key: your-secret-key"

# Should return:
# {
#   "status": "success",
#   "message": "NBA scraping started",
#   "players_scraped": 100
# }
```



Error Messages (Clear and Actionable)

When DATABASE_URL is Missing:

 DATABASE_URL **not** configured.
Database operations require a valid DATABASE_URL environment variable.
Set DATABASE_URL to enable scraping functionality.

When Database is Unreachable:

 Database connection failed: could not **connect** to server

When S3 is Missing:

 S3 **not** configured - image uploads will be skipped

Summary

Feature	Before	After
Deployment	✗ Requires DATABASE_URL	✓ Works without DATABASE_URL
Health Check	✗ Fails if DB unreachable	✓ Always succeeds
Error Messages	✗ Generic failures	✓ Clear, actionable errors
Flexibility	✗ Database required	✓ Database optional

Next Steps

1. **Deploy Now** - Remove DATABASE_URL from Render and deploy
2. **Verify** - Check `/health` endpoint returns `{"status": "healthy"}`
3. **Add Database Later** - When you have an external PostgreSQL database
4. **Or Use Abacus AI** - Deploy scraper within Abacus AI for internal database access

Deployment Time: 2 minutes

Root Cause: Fixed ✓

Status: Ready to deploy

Last Updated: December 13, 2025

Fix: Lazy database connection pattern