

Quick Deploy Guide - Python Backend to Railway

Estimated Time: 10-15 minutes

Prerequisites


Before starting:

- [x] Code is in GitHub: `https://github.com/baller70/BasketballAnalysisAssessmentApp.git`
 - [x] Backend is in `python-backend/` directory
 - [x] You have a GitHub account
 - [x] You know your Abacus AI frontend URL
-

Step-by-Step Deployment (Railway)


Step 1: Create Railway Account

1. Go to railway.app (`https://railway.app`)
2. Click **“Start a New Project”**
3. Sign in with **GitHub**

 Time: 2 minutes


Step 2: Deploy from GitHub


1. Click **“Deploy from GitHub repo”**
2. Select: `BasketballAnalysisAssessmentApp`
3. Railway will detect your Dockerfile automatically

 Time: 1 minute

Step 3: Configure Root Directory

1. Click on the deployed service
2. Go to **Settings** tab
3. Find **“Root Directory”**
4. Set to: `python-backend`
5. Click **Save**

 **Important:** This step is crucial! Your backend is in a subdirectory.

 Time: 1 minute

Step 4: Add Environment Variables

1. Click **“Variables”** tab
2. Click **“+ New Variable”**
3. Add these **5 variables**:

```
# 1. Replicate API Token
REPLICATE_API_TOKEN=r8_XVbSqNpDmahHdfRWDjmivN2ZNPk3MUH2w1N4x

# 2. CORS Origins (⚠️ REPLACE WITH YOUR FRONTEND URL!)
ALLOWED_ORIGINS=https://your-abacus-frontend-url.com,http://localhost:3000

# 3. Host
HOST=0.0.0.0

# 4. Port
PORT=$PORT

# 5. MediaPipe Complexity
MEDIAPIPE_MODEL_COMPLEXITY=2
```

⚠️ **Critical:** Replace `your-abacus-frontend-url.com` with your actual Abacus AI frontend URL!

🕒 Time: 3 minutes

Step 5: Generate Public URL

1. Go to **Settings > Networking**
2. Click **“Generate Domain”**
3. Copy the URL (looks like: `https://basketball-backend-production-xxxx.up.railway.app`)

📄 **Save this URL** - you'll need it for the frontend!

🕒 Time: 1 minute

Step 6: Wait for Deployment

1. Go to **“Deployments”** tab
2. Watch the build logs
3. Wait for **“Deployment live”** (green status)

🕒 Time: 3-5 minutes

Step 7: Test Your Backend

Open in browser or run in terminal:

```
curl https://your-railway-url.up.railway.app/health
```

Expected Response:

```
{
  "status": "healthy",
  "version": "1.0.0",
  "mediapipe_available": true
}
```

✔ If you see this, your backend is live!

🕒 Time: 1 minute

Step 8: Update Frontend

1. Go to your **Abacus AI dashboard**
2. Find your frontend deployment settings
3. Update environment variable:

```
bash
```

```
NEXT_PUBLIC_PYTHON_API_URL=https://your-railway-url.up.railway.app
```

4. **Redeploy** the frontend

⚠ **Important:** No trailing slash!

🕒 Time: 2 minutes

Step 9: Update CORS (If Needed)

If you forgot to add your frontend URL in Step 4:

1. Go back to Railway **Variables** tab
2. Edit `ALLOWED_ORIGINS`
3. Add your Abacus AI frontend URL
4. Example:

```
bash
```

```
ALLOWED_ORIGINS=https://basketball-app.abacus.ai,http://localhost:3000
```

5. Save (Railway will automatically redeploy)

🕒 Time: 1 minute

Step 10: Test Integration

1. Open your Abacus AI frontend
2. Navigate to analysis page
3. Upload a basketball image
4. Verify:
 - ✔ Image uploads
 - ✔ Pose detection works
 - ✔ Skeleton overlay displays
 - ✔ No CORS errors in console (F12)

🕒 Time: 2 minutes

You're Done!

Your backend is deployed and connected to your frontend.

URLs to Save:

- Backend: `https://your-railway-url.up.railway.app`
- Frontend: `https://your-abacus-frontend-url.com`
- API Docs: `https://your-railway-url.up.railway.app/docs`

Quick Troubleshooting

Problem: CORS Error

Symptom: Console shows “blocked by CORS policy”

Solution:

1. Go to Railway → Variables
2. Edit `ALLOWED_ORIGINS`
3. Add your frontend URL
4. Save and wait for redeployment

Problem: 404 Not Found

Symptom: `/health` returns 404

Solution:

1. Check **Root Directory** is set to `python-backend`
2. Redeploy if needed

Problem: Build Failed

Symptom: Deployment shows “Build failed”

Solution:

1. Check deployment logs for errors
2. Verify all files are in GitHub
3. Ensure `Dockerfile` exists in `python-backend/`

Problem: MediaPipe Not Available

Symptom: `mediapipe_available: false` in health check

Solution:

1. Your Dockerfile already includes the fix
2. Try redeploying from scratch
3. Check build logs for “Successfully installed mediapipe”

Problem: Frontend Can't Connect

Symptom: Network errors in browser

Solution:

1. Verify `NEXT_PUBLIC_PYTHON_API_URL` is set correctly
 2. Check no trailing slash in URL
 3. Ensure backend is running (test `/health`)
 4. Check CORS settings
-



Next Steps

- **Working?** Great! Monitor performance and costs
 - **Issues?** See full troubleshooting in `PYTHON_BACKEND_DEPLOYMENT_GUIDE.md`
 - **Learn more:** Railway docs at docs.railway.app (<https://docs.railway.app>)
-



Cost Breakdown (Railway)

Free Tier:

- \$5 credit per month
- Approximately 500-1000 API requests
- Good for testing and small projects

Hobby Plan (\$5/month):

- Unlimited usage (fair use)
- Always-on service
- Better performance
- Recommended for production

Estimated Costs:

- Development: Free tier is sufficient
 - Production (100 users/day): ~\$5-10/month
 - High traffic (1000+ users/day): ~\$20-50/month
-



Important Links

- **Full Deployment Guide:** `/home/ubuntu/basketball_app/PYTHON_BACKEND_DEPLOYMENT_GUIDE.md`
 - **Railway Dashboard:** railway.app/dashboard (<https://railway.app/dashboard>)
 - **API Documentation:** <https://your-railway-url/docs> (auto-generated by FastAPI)
 - **Backend README:** `/home/ubuntu/basketball_app/python-backend/README.md`
-

Need Help?

1. **Check logs:** Railway → Deployments → View Logs
 2. **Read full guide:** `PYTHON_BACKEND_DEPLOYMENT_GUIDE.md`
 3. **Railway support:** railway.app/help (<https://railway.app/help>)
 4. **GitHub issues:** Report issues in your repository
-

Quick Deploy Guide Version: 1.0.0

Last Updated: December 4, 2025

Platform: Railway (Recommended)

Deployment Checklist

Use this to track your progress:

- ☐ Railway account created
- ☐ Repository connected
- ☐ Root directory set to `python-backend`
- ☐ 5 environment variables added
- ☐ Public domain generated
- ☐ Deployment succeeded (green status)
- ☐ `/health` endpoint tested
- ☐ Frontend environment variable updated
- ☐ Frontend redeployed
- ☐ CORS configured correctly
- ☐ Integration test passed

All checked? 🎉 You're ready to go!