

Deployment Troubleshooting Guide

Problem Solved: Torch Dependency Error

The error “ERROR: No matching distribution found for torch==2.7.1+cpu” was caused by:

1. Global torch installation in the system Python environment
2. Potential pip cache containing torch references
3. Render’s dependency resolution picking up cached or global dependencies

Solution Applied

1. Clean Requirements File

Created minimal `requirements.txt` with only essential packages:

```
Flask==2.3.3
twilio==8.10.0
requests==2.31.0
python-dotenv==1.0.0
gunicorn==21.2.0
Werkzeug==2.3.7
```

2. Deployment Configuration Files

- `render.yaml` : Explicit build configuration with `--no-cache-dir`
- `.dockerignore` : Excludes Python cache and build files
- `Procfile` : Alternative deployment configuration
- `clean_deploy.sh` : Automated clean deployment script

3. Cache Clearing

- Cleared pip cache: `pip cache purge`
- Removed Python cache files
- Tested in isolated virtual environment

Render Deployment Steps

1. Commit Changes:

```
bash
git add .
git commit -m "Fix torch dependency error - clean deployment"
git push
```

2. Render Dashboard:

- Go to your Render service
- Click “Manual Deploy” → “Deploy latest commit”
- Monitor build logs for any torch references

3. **Build Command Override** (if needed):

In Render dashboard, set build command to:

```
pip install --no-cache-dir -r requirements.txt
```

Alternative Deployment Platforms

If Render continues to have issues, try these alternatives:

Option 1: Railway

1. Connect your GitHub repo to Railway
2. Railway will auto-detect Python and use requirements.txt
3. Set environment variables in Railway dashboard

Option 2: Heroku

1. Install Heroku CLI
2. Create new app: `heroku create your-app-name`
3. Set environment variables: `heroku config:set VAR_NAME=value`
4. Deploy: `git push heroku main`

Option 3: DigitalOcean App Platform

1. Connect GitHub repo
2. Configure as Python app
3. Set environment variables
4. Deploy

Option 4: Fly.io

1. Install flyctl
2. Run `fly launch` in project directory
3. Configure environment variables
4. Deploy with `fly deploy`

Verification Steps

After deployment, verify the fix:

1. **Check Build Logs:** No torch references should appear
2. **Test Health Endpoint:** `GET /health` should return 200
3. **Test TTS Endpoint:** `GET /test_tts` should return TwiML
4. **Monitor Memory Usage:** Should be much lower without ML libraries

Environment Variables Required

Ensure these are set in your deployment platform:

```
TWILIO_ACCOUNT_SID=your_account_sid  
TWILIO_AUTH_TOKEN=your_auth_token  
OWNER_PHONE=+1234567890  
WEBHOOK_BASE_URL=https://your-app.onrender.com  
CHATBOT_ID=3947607fe  
BUSINESS_NAME=Green Slice Lawn Care and Window Washing
```

Local Testing

Run the clean deployment script:

```
./clean_deploy.sh
```

This will:

- Clear all caches
- Create fresh virtual environment
- Install clean dependencies
- Verify no torch modules
- Test app import

Common Issues and Solutions

Issue: “torch still appears in build logs”

Solution:

- Clear Render’s build cache by changing Python version temporarily
- Use `--no-cache-dir` flag in build command
- Check for hidden dependency files

Issue: “Module not found errors”

Solution:

- Verify all required packages are in requirements.txt
- Check for typos in package names
- Ensure compatible versions

Issue: “Memory limit exceeded”

Solution:

- This should be resolved with torch removal
- Monitor memory usage after deployment
- Consider upgrading plan if needed

Success Indicators

- ✓ Build completes without torch references
- ✓ App starts successfully
- ✓ Health endpoint responds
- ✓ Memory usage under 100MB
- ✓ Phone integration works

Support

If issues persist:

1. Check build logs for specific error messages
2. Verify environment variables are set correctly
3. Test locally with the clean deployment script
4. Consider alternative deployment platforms listed above