

# Render.com Deployment Guide for vitalDSP Webapp

---

This guide explains how to deploy the vitalDSP webapp to Render.com.

## Prerequisites

1. A Render.com account
2. vitalDSP repository connected to Render.com
3. Python 3.9+ support

## Configuration Files

The following configuration files have been created/updated for Render.com deployment:

### 1. `render.yaml`

- Main Render.com configuration file
- Defines the web service with build and start commands
- Sets environment variables for production
- Configures health check endpoint

### 2. `Dockerfile`

- Updated for proper containerization
- Sets correct Python path and working directory
- Includes health check configuration
- Optimized for production deployment

### 3. `src/vitalDSP_webapp/requirements-prod.txt`

- Production dependencies with pinned versions
- Includes all necessary packages for the webapp
- Optimized for stability in production

### 4. `start.sh`

- Startup script for Render.com
- Handles environment setup
- Creates necessary directories

## Environment Variables

The following environment variables are configured:

- `PYTHON_VERSION`: 3.9.18
- `PORT`: 8000 (automatically set by Render.com)
- `HOST`: 0.0.0.0

- **DEBUG:** false
- **PYTHONPATH:** /opt/render/project/src

## Health Check

A health check endpoint is available at `/api/health` that returns:

- Service status
- Timestamp
- Memory and disk usage
- Service version

## Deployment Steps

1. **Connect Repository:** Link vitalDSP GitHub repository to Render.com
2. **Select Service Type:** Choose "Web Service"
3. **Configuration:** Render.com will automatically detect the `render.yaml` file
4. **Deploy:** Click "Deploy" to start the deployment process

## Manual Configuration (if needed)

If manual configuration:

1. **Build Command:**

```
pip install --upgrade pip
pip install -r src/vitalDSP_webapp/requirements-prod.txt
pip install -e .
```

2. **Start Command:**

```
python src/vitalDSP_webapp/run_webapp.py
```

3. **Environment Variables:**

- **PYTHON\_VERSION:** 3.9.18
- **PORT:** 8000
- **HOST:** 0.0.0.0
- **DEBUG:** false

## Monitoring

- Health checks are performed at `/api/health`
- Render.com will automatically restart the service if health checks fail
- Logs are available in the Render.com dashboard

# Troubleshooting

## Common Issues

1. **Import Errors:** Ensure `PYTHONPATH` is set correctly
2. **Port Issues:** Make sure the app binds to `0.0.0.0` and uses the `PORT` environment variable
3. **Dependencies:** Check that all dependencies are in `requirements-prod.txt`

## Logs

Check the Render.com service logs for detailed error information.

## Production Considerations

1. **File Uploads:** The uploads directory is created automatically
2. **Memory Usage:** Monitor memory usage through the health check endpoint
3. **Scaling:** Consider upgrading to a higher plan for production workloads
4. **Security:** Ensure proper security headers and HTTPS are configured

## Support

For issues specific to Render.com deployment, check:

- Render.com documentation
- Service logs in the Render.com dashboard
- Health check endpoint response