

Gradio Deployment Guide - Quick Reference

TL;DR

Local: `cd myapp && python demo.py` → `http://localhost:7860`

Docker: `docker build -t myapp:latest . && docker run -p 7860:7860 myapp:latest` → `http://localhost:7860`

⚠ **Critical:** Use `localhost:7860` in browser, not `0.0.0.0:7860` (Windows limitation)

Setup (One-time)

`setup_gradio_project_FINAL.bat`

Creates:

- `myapp/demo.py` - Gradio app
 - `myapp/requirements.txt` - Dependencies
 - `myapp/Dockerfile` - Container config
-

Method 1: Local Development

```
cd C:\Users\User\Documents\curso_python\2_deploy_app_with_code_engine
.venv\Scripts\activate
cd myapp
python demo.py
```

Access: **`http://localhost:7860`**

Stop: CTRL+C

Method 2: Docker

Prerequisites

- Docker Desktop running
- Local app stopped (port conflict)

Build & Run

```
cd
C:\Users\User\Documents\curso_python\2_deploy_app_with_code_engine\mya
pp
docker build -t myapp:latest .
docker run -p 7860:7860 myapp:latest
```

Access: **http://localhost:7860** (or click port in Docker Desktop)

Stop: CTRL+C

Background Mode

```
docker run -d -p 7860:7860 --name gradio-app myapp:latest
docker logs -f gradio-app          # View logs
docker stop gradio-app             # Stop
docker start gradio-app             # Restart
docker rm gradio-app               # Remove
```

Key Files

demo.py

```
import gradio as gr

def greet(name, intensity):
    return "Hello, " + name + "!" * int(intensity)

demo = gr.Interface(
    fn=greet,
    inputs=["text", "slider"],
    outputs=["text"],
)

# 0.0.0.0 required for Docker, but use localhost:7860 in browser
demo.launch(server_name="0.0.0.0", server_port=7860)
```

Dockerfile

```
FROM python:3.12-slim
WORKDIR /app
COPY requirements.txt requirements.txt
RUN pip3 install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 7860
CMD ["python", "demo.py"]
```

requirements.txt

```
gradio>=5.11.0
```

Common Issues

Issue	Fix
Can't access 0.0.0.0:7860	Use <code>localhost:7860</code>
Port already allocated	Stop other instance first
<code>ModuleNotFoundError</code>	Activate <code>venv</code>
Docker build requires 1 argument	Add <code>.</code> at end
Docker connection error	Start Docker Desktop
Code changes not in container	Rebuild image

Quick Commands

```
# Local
.venv\Scripts\activate && cd myapp && python demo.py

# Docker - build & run
cd myapp && docker build -t myapp:latest . && docker run -p 7860:7860 myapp:latest

# Docker - background
docker run -d -p 7860:7860 --name gradio-app myapp:latest

# Docker - manage
docker ps                # List running
docker logs gradio-app   # View logs
docker stop gradio-app   # Stop
docker rm gradio-app     # Remove container
docker rmi myapp:latest  # Remove image

# Docker - alternative port if 7860 busy
docker run -p 8080:7860 myapp:latest # Access at localhost:8080
```

Project Structure

```
2_deploy_app_with_code_engine/
├── .venv/                # Virtual environment
├── myapp/
│   ├── demo.py           # App code
│   ├── requirements.txt  # Dependencies
│   └── Dockerfile        # Container config
```

Why 0.0.0.0 vs localhost?

- **Server binding (0.0.0.0):** Listen on all interfaces (required for Docker)
- **Browser access (localhost):** Connect to local machine (Windows requirement)
- **Don't change demo.py** - it's correct. Just use `localhost` in browser.

Workflow

Development: Edit code → python demo.py → test → CTRL+C → repeat

Containerization: Edit code → docker build → docker run → test → CTRL+C

⚠ **Stop one before starting the other** (both use port 7860)

Success Indicators

Local:

* Running on local URL: http://0.0.0.0:7860

☒ Open localhost:7860

Docker:

[+] Building 45.2s (10/10) FINISHED
=> => naming to docker.io/library/myapp:latest

☒ Image built

* Running on local URL: http://0.0.0.0:7860

☒ Container running, open localhost:7860