Uvicorn ASGI web server

Client browser → HTTPS → **nginx** TLS, static → unix socket → **Uvicorn** HTTP and ASGI → ASGI → **Your app** FastAPI etc.

CONCEPTS

main:app means module:variable. Uvicorn serves ASGI apps (FastAPI, Starlette, Django ASGI) and listens on 127.0.0.1:8000 by default.

Defaults to 127.0.0.1, so a container needs --host 0.0.0.0 or it is unreachable from outside.

--reload and --workers cannot be combined. Use --reload for development only, workers or Gunicorn for production.

RUN

uvicorn *<module>*:*<app>*
Serve an ASGI app

uvicorn --factory *<module>*:*<func>*
Treat app as a factory

uvicorn *<module>*:*<app>* **--reload**
Auto-restart on file change

uvicorn *<module>*:*<app>* **--workers** *<n>*
Run n worker processes

uvicorn --app-dir *<dir>* *<module>*:*<app>*
Look for the app in a directory

RUN (cont.)

uvicorn --env-file *<file>* *<module>*:*<app>*
Load environment from a file

uvicorn --version
Print the version

uvicorn --help
List all options

python -m uvicorn *<module>*:*<app>*
Run via the interpreter

BINDING

--host *<addr>*
Bind address (default 127.0.0.1)

--port *<n>*
Bind port (default 8000)

--uds *<path>*
Bind a Unix socket

--root-path *<prefix>*
App mounted below a URL path

RELOAD

--reload
Watch files and restart

--reload-dir *<dir>*
Watch this directory only

--reload-include *<glob>*
Also watch matching files

RELOAD (cont.)

Install uvicorn[standard] to get watchfiles for faster, glob-aware reloading.

WORKERS AND LIMITS

--workers *<n>*
Number of worker processes

--timeout-keep-alive *<sec>*
Idle keep-alive timeout (default 5)

kill -HUP *<pid>*
Gracefully restart workers

PROTOCOLS

--http *<impl>*
h11 or httptools

LOGGING

--log-level *<level>*
critical, error, warning, info, debug, trace

--no-access-log
Silence request lines

--access-log
Enable request lines (default)

HTTPS AND PROXIES

--ssl-keyfile *<key>* **--ssl-certfile** *<crt>*
Serve HTTPS directly

--proxy-headers
Read X-Forwarded-For and -Proto

--forwarded-allow-ips *<ips>*
Trusted proxy IPs or sockets

Behind nginx over a Unix socket, set X-Forwarded-For and X-Forwarded-Proto in the proxy and list the proxy in --forwarded-allow-ips.

GUNICORN AND CODE

Gunicorn with Uvicorn workers

```
pip install uvicorn-worker
gunicorn main:app -w 4 -k
uvicorn_worker.UvicornWorker
```

uvicorn.workers.UvicornWorker still exists but is deprecated; the docs recommend the separate uvicorn-worker package. Use UvicornH11Worker on PyPy.

GUNICORN AND CODE (cont.)

Run from Python

```
import uvicorn
if __name__ == "__main__":
    uvicorn.run("main:app",
host="0.0.0.0", port=8000,
reload=True)
```

Pass an import string, not the app object, to uvicorn.run when using reload or workers.

RECIPES

Dev server

```
uvicorn main:app --reload --
reload-dir src \
  --log-level debug
```

Container entrypoint

```
uvicorn main:app --host
0.0.0.0 --port 8000 \
  --proxy-headers --
forwarded-allow-ips '*' \
  --no-access-log
```

Behind nginx on a socket

```
uvicorn main:app --uds /run/
app.sock --workers 4 \
  --proxy-headers
# nginx: proxy_pass http://
unix:/run/app.sock;
```

Local HTTPS

```
mkcert localhost
uvicorn main:app --ssl-
keyfile localhost-key.pem \
  --ssl-certfile
localhost.pem
```