

---

LARAVEL CHEAT SHEET

# Reverb in Docker

Public WSS vs internal process, proxy, scaling, observability

## TL;DR

---

- Reverb is a **long-running WebSocket server**, not a dev command promoted to production.
- Publish the config, set the *public* and *internal* addresses explicitly.
- Supervise the process with a container restart policy / process manager.
- Horizontal scaling is a shared-state decision, not the load balancer alone.

## Two addresses

---

```
# .env – SERVER_* is the process binding inside the container
REVERB_SERVER_HOST=0.0.0.0
REVERB_SERVER_PORT=8080

# public address – behind TLS, this is how Echo connects
REVERB_HOST=ws.example.com
REVERB_PORT=443
REVERB_SCHEME=https

# secrets stay server-side; VITE_ is connection metadata only
VITE_REVERB_APP_KEY=${REVERB_APP_KEY}
VITE_REVERB_HOST=ws.example.com
```

Expose **WSS only**. Allowed origins = real application origins. Never expose port 8080 to the internet.

## The proxy forwards the upgrade

---

```
location /app/ {
    proxy_pass http://reverb:8080;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_read_timeout 60s;
}
```

# Scaling = a shared-state decision

---

- **One instance** = a valid first deployment.
- **Multiple instances** need supported horizontal scaling + a shared backend (Redis) so every node sees broadcasts and channel state.
- A load balancer alone does **not** scale isolated processes.
- Test **two connections landing on different instances**.

# Supervision and observability

---

```
$ php artisan reverb:start
```

Supervise with a container restart policy or a process manager; a health signal; structured logs; a deliberate restart on deploy.

Monitor: connections, reconnects, event latency, CPU, file descriptors, proxy responses / upgrade errors.

# Test the whole chain

---

- a `ShouldBroadcast` event → queue → Echo listener
- private channel authorization
- reconnect after a restart
- before replicas: a documented shared-Redis path across 2 instances



## RECAP

# The sheet in short

- `SERVER_*` = process in the container; `REVERB_HOST` = Echo address
- WSS only, port 8080 never to the internet
- Proxy: `proxy_http_version 1.1` + upgrade headers
- Multiple instances → shared Redis, not the LB alone
- Process supervision + monitor connections/FD/latency
- Not WebSockets when polling is enough



# Found this useful?

Save this cheat sheet for later.

Full article: [dominik-dev.pl](https://dominik-dev.pl)

Share • Comment