

Cheat-Sheet — Serveurs TCP/HTTP & Pipeline DevOps

Pipeline d'exécution

1. Activer l'environnement Python : `source venv/bin/activate`
2. Générer les fichiers HTTP : `python3 create_http_files.py`
3. Compilation optimisée : `make clean` `make -j$(nproc)`
4. Démarrage de tous les serveurs : `./scripts/start_all.sh`

Commandes clés

<code>make clean</code>	— nettoyage complet	<code>make -j\$(nproc)</code>	— compilation rapide	<code>make</code>
<code>debug</code>	— compilation avec sanitizers	<code>make test</code>	— tests automatiques	<code>make</code>
<code>kill_servers</code>	— arrêt propre des serveurs			

Debug & Analyse

Valgrind (memory) : `valgrind --leak-check=full ./bin/serveur_multi` Valgrind (threads) : `valgrind --tool=helgrind ./bin/serveur_multi` Sanitizers GCC : `make debug`

Benchmarks & Stress Tests

Stress TCP mono-thread : `python3 python/client_stress_tcp.py --port 5050 --clients 200`
Stress TCP multi-thread : `python3 python/client_stress_tcp.py --port 5051 --clients 200`
Stress HTTP (mono/multi) : `python3 python/client_stress_http.py --port 8080 --clients 200`
`python3 python/client_stress_http.py --port 8081 --clients 200` Benchmarks extrêmes : `make benchmark_extreme`

CI/CD – GitHub Actions

• `build.yml` : build + tests + valgrind • `cppcheck.yml` : analyse statique • `codeql.yml` : sécurité
• `benchmarks.yml` : bench + badge throughput • `deploy_docs.yml` : GitHub Pages