

# Production Setup: Explained Simply

*n8n Performance Architecture — Burger Shop Analogy (200 VUs / 1 Hour Example)*

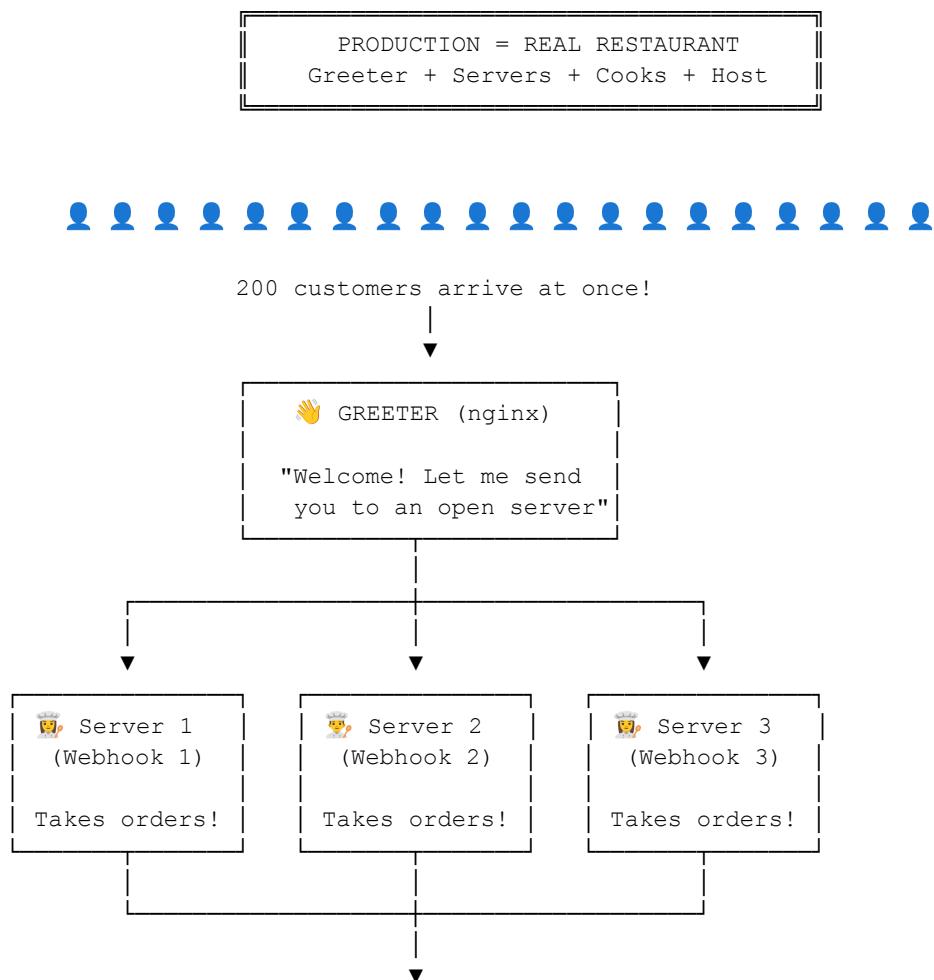
## 1) Big Idea (In One Sentence)

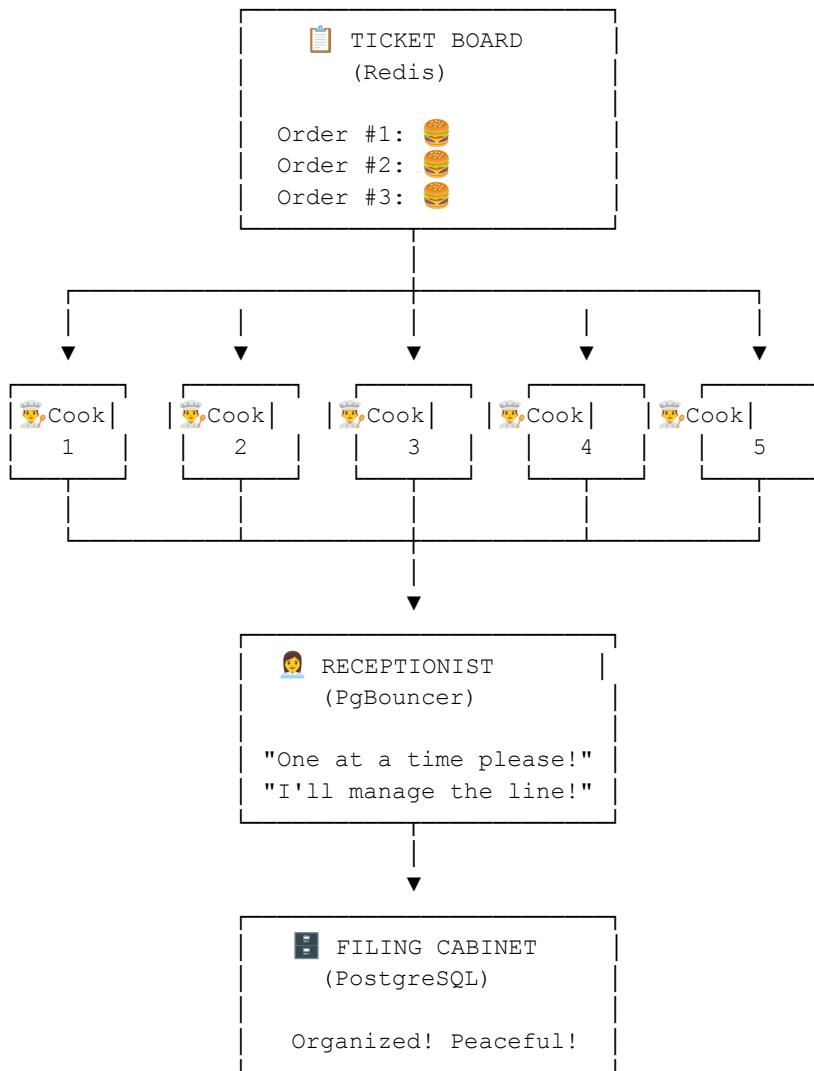
Production is like a real restaurant: a greeter (nginx) spreads customers across multiple servers, a ticket board (Redis) organizes work, cooks (workers) handle heavy execution, and a receptionist (PgBouncer) prevents database connection chaos—so everything stays calm under load.

## 2) Architecture Diagram (ASCII)

● Production Setup: Explained Simply

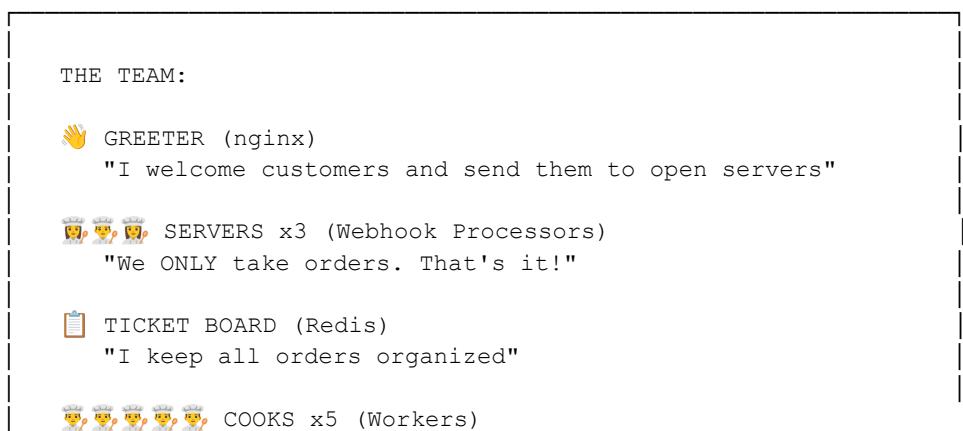
🍔 Imagine a REAL Professional Restaurant!





### 3) Everyone Has One Job (Why This Scales)

⌚ Everyone Has ONE Job!



"We ONLY cook burgers. That's it!"

👩 RECEPTIONIST (PgBouncer)

"I manage ALL traffic to the filing cabinet"

🗄 FILING CABINET (PostgreSQL)

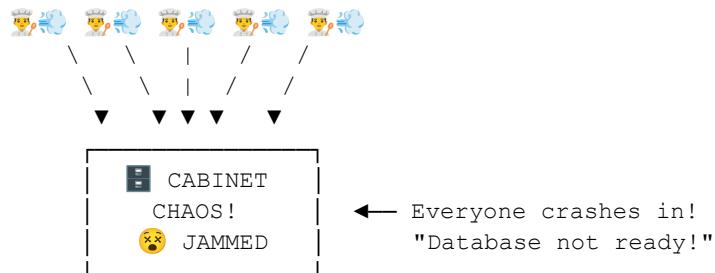
"I store everything safely"

EVERYONE DOES ONE THING REALLY WELL!

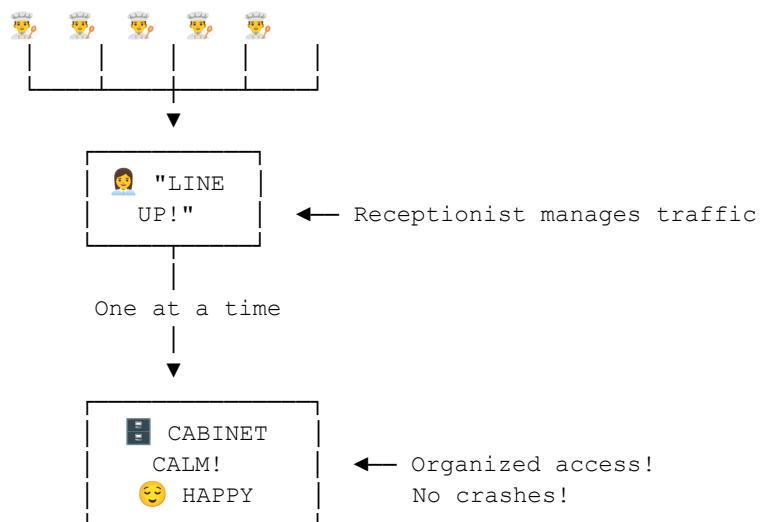
## 4) Why PgBouncer Matters (No More "Database not ready")

👩 The Magic of the Receptionist (PgBouncer)

WITHOUT RECEPTIONIST (Queue+PG) :

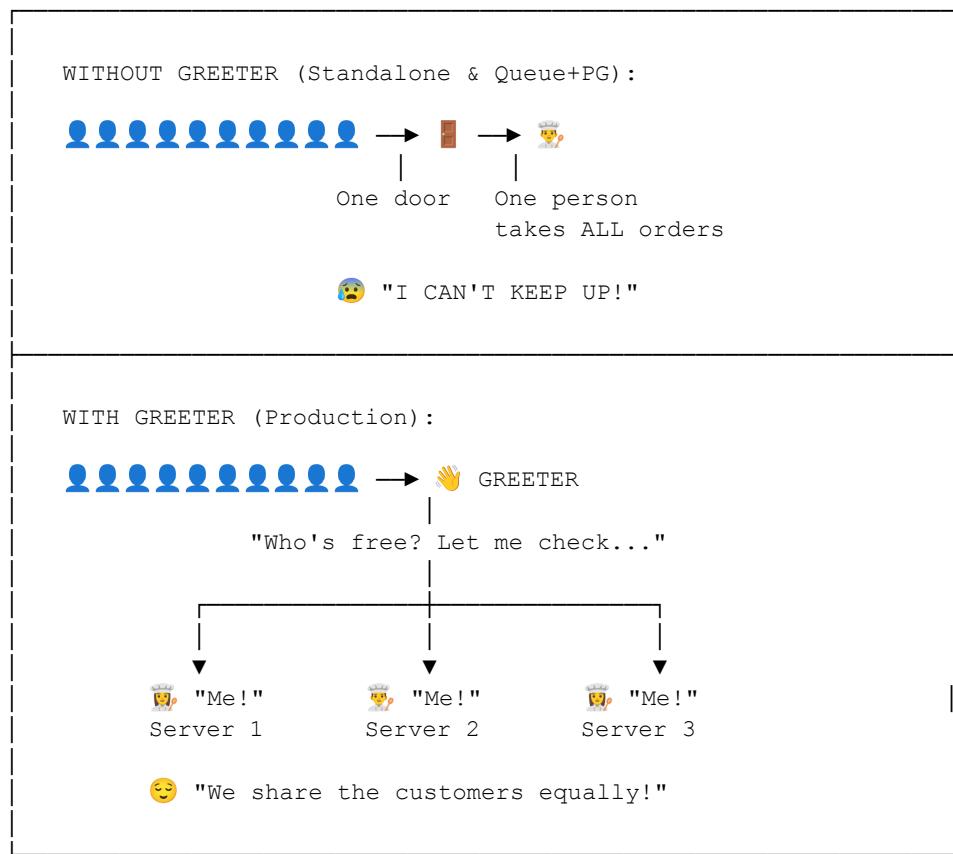


WITH RECEPTIONIST (Production) :



## 5) Why nginx Matters (No Single "One Door" Bottleneck)

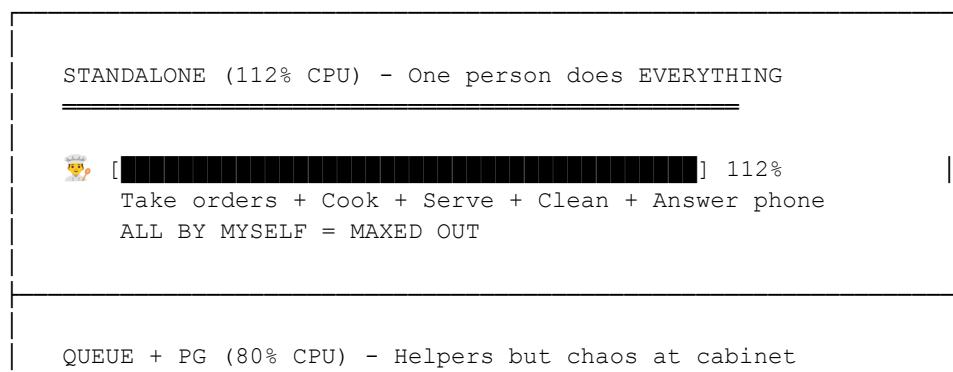
👋 The Magic of the Greeter (nginx)



## 6) Performance Insights (Why CPU Drops to ~6%)

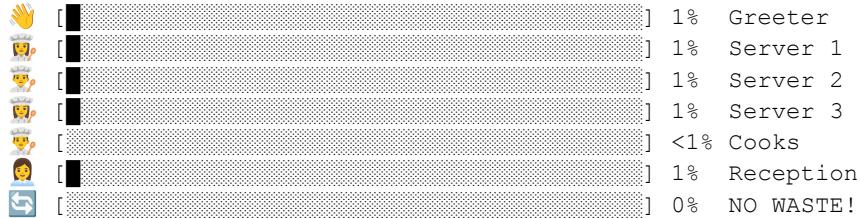
In Production, work stops piling up in one place. Requests are distributed across multiple webhook processors, heavy workflow execution runs in separate workers, and PgBouncer prevents connection storms to PostgreSQL. That combination cuts wasted CPU dramatically.

💻 Why CPU is Only 6%!





PRODUCTION (6% CPU) - Everyone has ONE job, no fighting



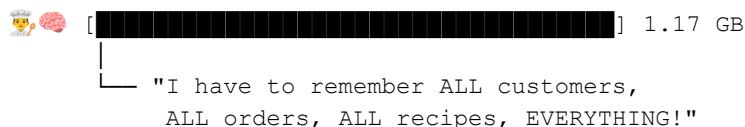
Total: ~6% CPU

## 7) Memory Insights (Why ~251 MB Stays Stable)

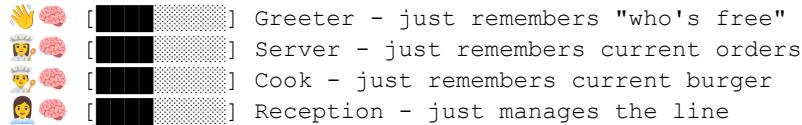
The same principle applies to memory: each component keeps only what it needs. Smaller responsibilities per container plus sensible limits/restarts help prevent slow growth and crashes under sustained load.

🧠 Memory: Why Only 251 MB?

STANDALONE (1.17 GB) - One brain remembering EVERYTHING



PRODUCTION (251 MB) - Shared memory + focused roles



## 8) What Happens at 200 Customers for 1 Hour (Result Summary)

💥 What Happens at 200 Customers for 1 HOUR

| STANDALONE (After 2 minutes)           |  |
|--|--|
| 🟡                                      | "I QUIT!" (crashed)                      |
| ✗                                      | 65% of customers left angry              |
| ✓                                      | 35% got served                           |
| QUEUE + PG (After 2 minutes)           |  |
| 🟡                                      | "Cabinet jammed!" (database overwhelmed) |
| ✗                                      | 33% of customers left angry              |
| ✓                                      | 67% got served                           |
| PRODUCTION (After 1 HOUR!)             |  |
| ✓                                      | 720,448 customers served                 |
| ✓                                      | 100% success rate                        |
| ✗                                      | 0 angry customers                        |
| Still running strong after 60 MINUTES! |  |

## 9) Final Comparison Table (All 3 Setups)

📊 The Final Comparison

| 🍔 BURGER SHOP COMPARISON 🍔 |            |          |            |
|----------------------------|------------|----------|------------|
|                            | STANDALONE | QUEUE+PG | PRODUCTION |
| CPU                        | 🔴 112%     | 🟡 80%    | 🟢 6%       |
| Memory                     | 🔴 1.17 GB  | 🟡 980 MB | 🟢 251 MB   |
| Success                    | 🔴 35%      | 🟡 67%    | 🟢 100%     |
| Duration                   | 🔴 2 min    | 🔴 2 min  | 🟢 1 HOUR+  |

---

## 10) Bottom Line (One-Liner)

**Production wins because it removes bottlenecks and removes waste:** nginx balances incoming load, workers execute separately, and PgBouncer prevents database connection storms—so performance stays stable even under sustained traffic.