Akhilesh Mishra • 2nd
Senior DevOps Engineer | Write to create value | 10.5K on Medium | 25k on Linkedin | Mentor |...
Book an appointment
12h • Edited • 🌐

Follow

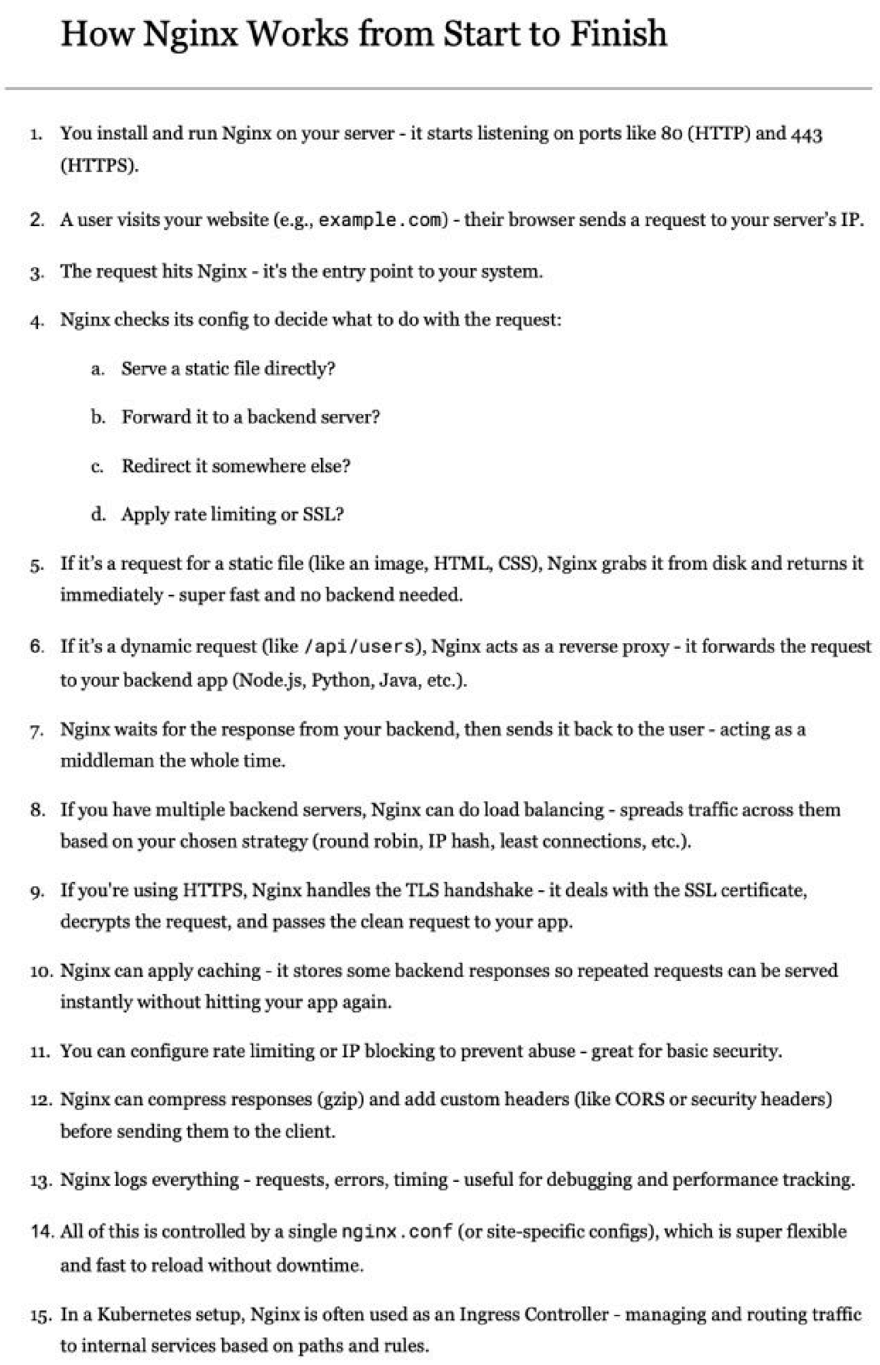
As a devops engineers, we all use nginx for one or another purpose.

But most don’t understand how powerful Nginx really is.

This cheat sheet(From X) will help you understand Nginx.

👉 If you want to learn DevOps with advanced, real-world projects, then join my 16-week bootcamp starting on May 10th.

Link: https://lnkd.in/gG_3REeB



How Nginx Works from Start to Finish

1. You install and run Nginx on your server - it starts listening on ports like 80 (HTTP) and 443 (HTTPS).
2. A user visits your website (e.g., `example.com`) - their browser sends a request to your server's IP.
3. The request hits Nginx - it's the entry point to your system.
4. Nginx checks its config to decide what to do with the request:
   a. Serve a static file directly?
   b. Forward it to a backend server?
   c. Redirect it somewhere else?
   d. Apply rate limiting or SSL?
5. If it's a request for a static file (like an image, HTML, CSS), Nginx grabs it from disk and returns it immediately - super fast and no backend needed.
6. If it's a dynamic request (like `/api/users`), Nginx acts as a reverse proxy - it forwards the request to your backend app (Node.js, Python, Java, etc.).
7. Nginx waits for the response from your backend, then sends it back to the user - acting as a middleman the whole time.
8. If you have multiple backend servers, Nginx can do load balancing - spreads traffic across them based on your chosen strategy (round robin, IP hash, least connections, etc.).
9. If you're using HTTPS, Nginx handles the TLS handshake - it deals with the SSL certificate, decrypts the request, and passes the clean request to your app.
10. Nginx can apply caching - it stores some backend responses so repeated requests can be served instantly without hitting your app again.
11. You can configure rate limiting or IP blocking to prevent abuse - great for basic security.
12. Nginx can compress responses (gzip) and add custom headers (like CORS or security headers) before sending them to the client.
13. Nginx logs everything - requests, errors, timing - useful for debugging and performance tracking.
14. All of this is controlled by a single `nginx.conf` (or site-specific configs), which is super flexible and fast to reload without downtime.
15. In a Kubernetes setup, Nginx is often used as an Ingress Controller - managing and routing traffic to internal services based on paths and rules.

524 9 comments 35 reposts