

---

AI translation · View original & related papers at  
[chinaxiv.org/items/chinaxiv-202310.03219](https://chinaxiv.org/items/chinaxiv-202310.03219)

---

## Nginx Server “Cheat Sheet” Post-Print

**Authors:** Liu Minhao

**Date:** 2023-10-08T00:00:00+00:00

### Abstract

The new editorial and distribution system, developed by the Technical Bureau of Xinhua News Agency, serves the daily news gathering, editing, and reporting operations across the entire agency. As a key component of the Xinhua All-Media Project implemented in 2016, the system provides four major functional categories: information aggregation, reporting planning, multimedia production, and product distribution. Faced with rapidly changing business requirements, increasingly complex operational and...

### Full Text

#### Preamble

The new editorial and distribution system, developed by the Technical Bureau of Xinhua News Agency, serves the daily news gathering, editing, and reporting operations across the entire agency. As a key component of the Xinhua All-Media Project implemented in 2016, the system provides four major functional categories: information aggregation, reporting planning, multimedia production, and product distribution. Faced with rapidly changing business requirements, increasingly complex operational and maintenance challenges, and fast-paced IT technology evolution, the project team recognized the need to “follow the laws of news dissemination and emerging media development, utilize internet technology systems, and enhance technical personnel’s autonomous mastery of advanced technologies.” This philosophy guided the selection of Nginx as the web server solution for the Xinhua All-Media Project.

#### Nginx: A High-Performance Solution

Nginx, a grassroots-born software, has long been renowned in the industry for elegantly solving the C10K problem (where traditional Apache servers would buckle under 10,000 concurrent connections on a single physical server). Today, Nginx plays a critical role as a key component of the access layer in Xinhua’s

next-generation technical architecture. By achieving autonomous mastery of Nginx installation and configuration, and flexibly leveraging built-in modules, third-party extensions, and custom Lua scripts, the agency can accomplish not only basic proxy forwarding and software load balancing, but also implement authentication and authorization, SSL offloading, rate limiting and traffic shaping, and access auditing through sophisticated traffic control mechanisms.

Nginx is an open-source HTTP and reverse proxy server originally written by Russian developer Igor Sysoev and first released in 2004. Over the past decade, its exceptional performance has fueled rapid growth during the internet boom. According to W3Techs, nearly 60% of the world's top 1,000 websites now use Nginx, including Netflix, Dropbox, and Wikipedia, while Apache ranks second at only 16.5%. Nginx's standout characteristics—high concurrency, lightweight footprint, and stability—stem from its asynchronous, non-blocking, event-driven request processing architecture. Tests demonstrate that Nginx can handle up to four times the concurrent request volume of Apache while consuming significantly less memory.

## Installation and Configuration

Mastering Nginx installation begins with three prerequisite packages: the PCRE regular expression library, zlib compression library, and OpenSSL cryptographic toolkit, which are required for configuration, compression, and SSL functionality respectively. The installation process comprises three steps: configuration, compilation, and installation. First, download the source package from the official Nginx website (<http://nginx.org/>), with the stable version generally recommended [Figure 1: see original paper].

During the configuration phase, Nginx's highly modular architecture requires manual specification of module options. Built-in modules that are disabled by default can be activated using `--with-module_{name}`, while third-party modules are integrated via `--add-module=/path/to/module`. Common built-in modules include:

- `--with-stream` for Layer 4 protocol forwarding
- `--with-http_{ss1}_{module}` for HTTPS support
- `--with-http_{gzip}_{static}_{module}` for static Gzip compression
- `--with-http_{geoip}_{module}` for GeoIP functionality
- `--with-http_{flv}_{module}` for server-side pseudo-streaming of FLV files
- `--with-http_{stub}_{status}_{module}` for status information pages

Nginx offers extensive third-party module support, with a complete listing available at <https://www.nginx.com/resources/wiki/modules/index.html>. Popular third-party modules include:

- `nginx_{{upstream}}_{{check}}_{{module}}` for active health checks
- `nginx-sticky-module` for session persistence
- `ngx_{{log}}_{{if}}` for conditional logging

Compilation and installation are completed using `make` and `make install` commands. Following installation, administrators should master basic Nginx control commands:

- `nginx` to start the server
- `nginx -s stop` for immediate shutdown
- `nginx -s quit` for graceful shutdown
- `nginx -s reload` to reload configuration files

## 1. Reverse Proxy

Once applications are deployed, Nginx's reverse proxy capabilities enable user access. Requests arriving at the Nginx server via domain names can be distinguished by domain and path, then forwarded to different backend server clusters—even when users cannot directly reach these backend servers over the network. Both Layer 4 and Layer 7 proxying utilize the `proxy_{{pass}}` URL directive, implemented through the `ngx_{{stream}}_{{proxy}}_{{module}}` and `ngx_{{http}}_{{proxy}}_{{module}}` modules respectively.

## 2. Load Balancing

When application servers encounter issues, using Nginx as a software load balancer minimizes impact. Nginx provides flexible load balancing configurations including round-robin, weighted round-robin, IP hash, and primary-backup modes to meet diverse requirements. Furthermore, integrating the `nginx_{{upstream}}_{{check}}_{{module}}` third-party module enables active health checks based on TCP and HTTP, significantly enhancing Nginx's load balancing capabilities.

## 3. Access Control

Not all systems are fully exposed to the internet; some permit only office network access, while others may expose certain pages to the internet while restricting others to specific internal IPs. For these granular access control requirements, the `ngx_{{http}}_{{access}}_{{module}}` provides flexible Layer 7 whitelist and blacklist capabilities through straightforward configuration directives.

## 4. Authentication and Authorization

Some applications require an additional security layer, mandating correct username and password credentials for access. Nginx's built-in authentication module, `ngx_{{http}}_{{auth}}_{{basic}}_{{module}}`, easily implements this functionality when combined with the `htpasswd` command for generating

username-password database files. Another popular approach uses Nginx with Lua (similar to OpenResty) to enforce unified user authentication system login, filtering, and authorization before accessing business systems.

## 5. SSL Offloading

In recent years, HTTPS has gradually replaced HTTP as the preferred transmission protocol, with numerous websites migrating to HTTPS for encrypted, trusted access. Nginx's `ngx_{{http}}_{{ssl}}_{{module}}` provides essential HTTPS support, enabling websites to install SSL certificates on the Nginx server for cryptographic data stream offloading.

## 6. Rate Limiting and Traffic Shaping

When actual traffic far exceeds system-tested capacity, systems risk collapse and outbound bandwidth congestion, necessitating comprehensive rate limiting. As the outermost barrier, the access layer must protect the entire system. Nginx offers multiple rate limiting methods: the `ngx_{{http}}_{{core}}_{{module}}` can limit connection counts, request frequencies can be throttled, and the `sys-guard` module enables overload protection.

## 7. Access Auditing

By deploying an ELK (Elasticsearch, Logstash, Kibana) platform to collect and analyze Nginx access layer logs, organizations can implement a visualized access auditing and analysis platform. Message queues such as Redis or Kafka can be employed for log streaming.

## Smooth Upgrades

Beyond these application scenarios, Nginx supports a crucial operation for administrators: smooth upgrades. As internet software with frequent version iterations, Nginx requires regular upgrades to leverage new features. The smooth upgrade process completes without service disruption—administrators simply back up and replace the executable file in the `sbin` directory, then execute `kill -s USR2 \cat /path/to/nginx.pid` to run old and new versions in parallel before gracefully terminating the old process.

Ready to experience this powerful web server? Explore Nginx's capabilities today.

*(Author: Xinhua News Agency Communication Technology Bureau)*

*Note: Figure translations are in progress. See original paper for figures.*

*Source: ChinaXiv — Machine translation. Verify with original.*