



Grundlagen der Web-Entwicklung

INF3172

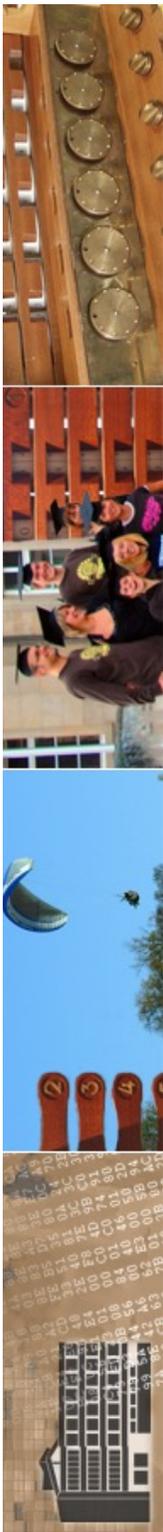
Performante Webserver:
NGINX und Caddy

Thomas Walter

14.11.2024

Version 1.0





Software-Anbieter aufgepasst: Sie haften jetzt für fehlerhafte Produkte

Mit der neuen Produkthaftungsrichtlinie kommen auf Software-Hersteller verschärfte Auflagen hinzu. Und nicht nur sie: Viele andere Akteure haften jetzt auch.

🇬🇧 🛡️ 🔊 🖨️ 💬 711



(Bild: iX)

12.11.2024, 14:43 Uhr | Lesezeit: 5 Min. | iX Magazin

Von Christina Kiefer

INHALTSVERZEICHNIS ▾

Die Digitalisierung von Produkten wirft neue Haftungsfragen auf. Die bisher geltende Produkthaftungsrichtlinie 85/374/EWG von 1985 trägt den spezifischen Risiken digitaler Produkte nicht vollständig Rechnung. So fehlten unter anderem hinreichend klare Regelungen zur Haftung für Software oder Produkte mit Softwarekomponenten. Die Europäische Kommission hatte daher am 28. September 2022 einen Entwurf für eine Neufassung der Produkthaftungsrichtlinie (ProdHaftRL) vorgelegt, der am 10. Oktober 2024 vom Rat angenommen wurde. Die ProdHaftRL wird die bisher geltende Richtlinie vollständig ersetzen.



das Weihnachtsrätsel

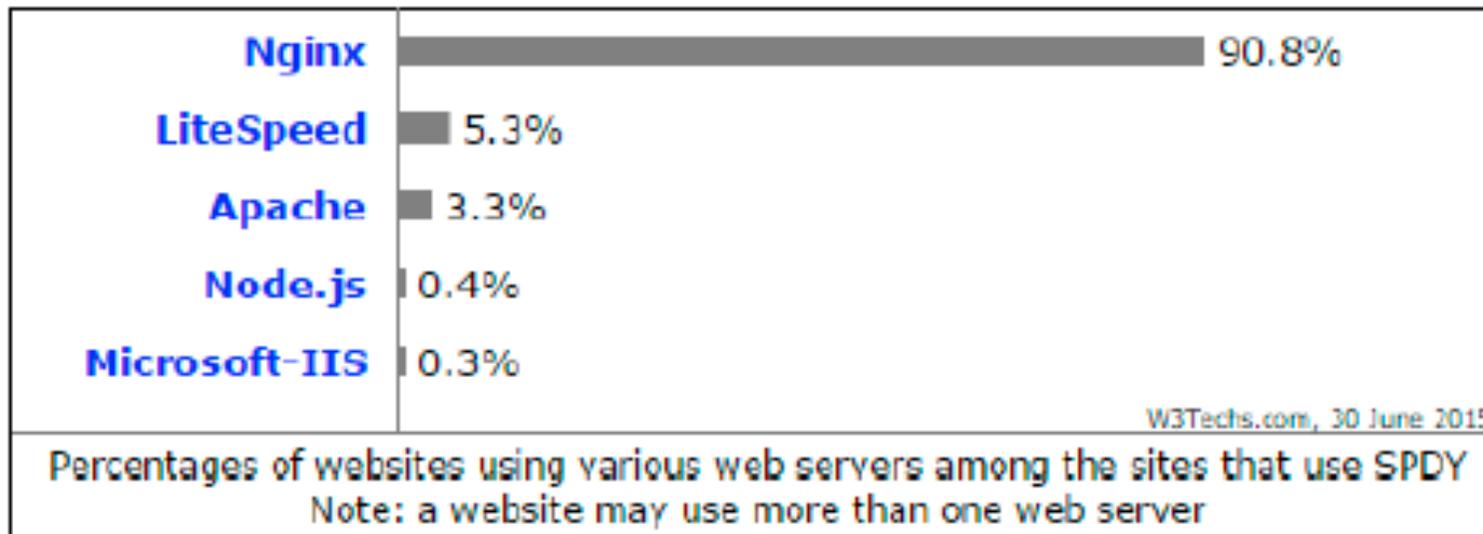
- am 1. Advent 01.12.2024
ab 11.00h:
das Weihnachtsrätsel!
- Preise für schnellste Lösung und Verlosung
weiterer
- Auflösung und Verlosung der Gewinne in der
Vorlesung vor Weihnachten

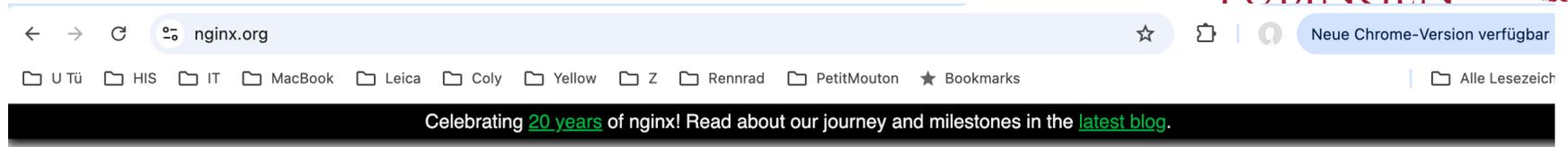




NGINX

- Web Server & Reverse/Mail Proxy
- nginx.org (nginx.com existiert auch)
- Optimiert auf Performance





nginx

nginx ("engine x") is an HTTP web server, reverse proxy, content cache, load balancer, TCP/UDP proxy server, and mail proxy server. Originally written by [Igor Sysoev](#) and distributed under the [2-clause BSD License](#). Enterprise distributions, commercial support and training are [available from F5, Inc.](#)

[Docs](#) • [Code](#) • [Install](#) • [Beginner's Guide](#)

Latest News

2024-10-22 [njs-0.8.7](#) bugfix version has been [released](#).

2024-10-02 [nginx-1.27.2](#) mainline version has been released.

2024-10-02 [njs-0.8.6](#) version has been [released](#), featuring the [QuickJS](#) engine support.

2024-09-06 The [nginx](#) project has officially [moved](#) to [GitHub](#). Use [GitHub issues](#) instead of [trac.nginx.org](#), [GitHub discussions](#) instead of nginx mailing list.

[Older news](#)

Other NGINX Projects



njs

NGINX JavaScript (njs) extends nginx functionality with an ECMAScript-compatible interpreter for HTTP and Stream modules.

[Docs](#) • [Code](#)



Unit

NGINX Unit is a lightweight application runtime that simplifies the application stack for running web apps and APIs.

[Docs](#) • [Code](#)



Ingress

NGINX Ingress Controller connects Kubernetes apps and services with rock solid request handling, auth, self-service CRDs, and easy debugging.

[Docs](#) • [Code](#)



english

[русский](#)

[news](#)

[about](#)

[download](#)

[security](#)

[documentation](#)

[faq](#)

[books](#)

[community](#)

[enterprise](#)

[x.com](#)

[blog](#)

[unit](#)

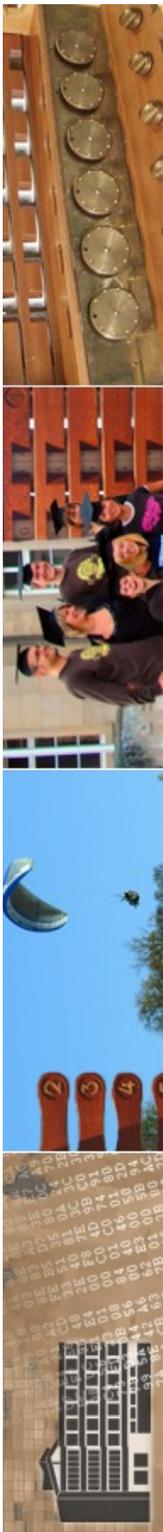
[njs](#)





Eigenschaften

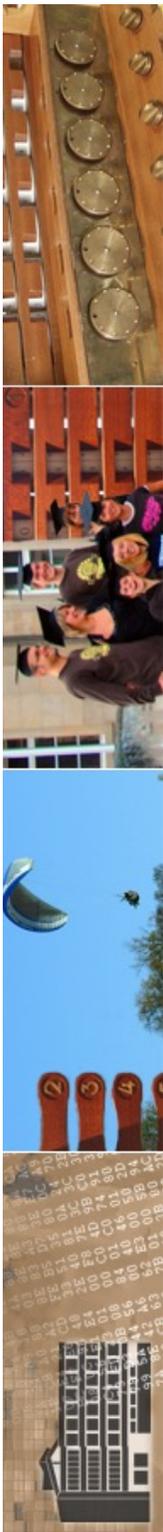
- Auslieferung von statischem Inhalt
- Reverse Proxy mit Caching
- Load Balancing
- TLS (SSL)
- FastCGI, CGI
- Streaming
- HTTP 1.1/2.0, SPDY
- Websockets
- Embedded Perl Scripting





Genese

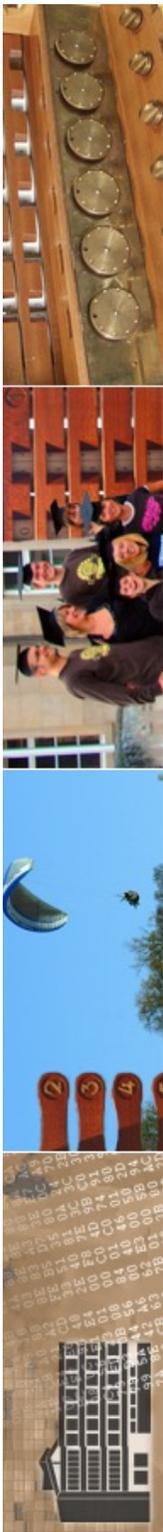
- 2002: Entwicklungsbeginn durch Igor Sysoev
- 2004: Veröffentlichung von Version 0.1.0
- 2011: Veröffentlichung von Version 1.0
- 2012 Gründung von NGINX Inc.
- 2013 NGINX plus (kommerzielle Version)
- Oktober 2024: NGINX 1.27.2





NGINX in Details

- Architektur
 - Modular
 - C
- Betriebssysteme
 - alle gängigen
 - Linux, OIS-X, FreeBSD
 - Windows nicht so stark performance-optimiert





Celebrating [20 years](#) of nginx! Read about our journey and milestones in the [latest blog](#).

nginx: Linux packages

[Supported distributions and versions](#)

[Installation instructions](#)

[RHEL and derivatives](#)

[Debian](#)

[Ubuntu](#)

[SLES](#)

[Alpine](#)

[Amazon Linux](#)

[Source Packages](#)

[Dynamic Modules](#)

[Signatures](#)

Supported distributions and versions

nginx packages are available for the following Linux distributions and versions:

[RHEL and derivatives](#)

Version	Supported Platforms
8.x	x86_64, aarch64/arm64
9.x	x86_64, aarch64/arm64

[Debian](#)

Version	Supported Platforms
11.x "bullseye"	x86_64, aarch64/arm64
12.x "bookworm"	x86_64, aarch64/arm64

[Ubuntu](#)

Version	Supported Platforms
20.04 "focal"	x86_64, aarch64/arm64
22.04 "jammy"	x86_64, aarch64/arm64
24.04 "noble"	x86_64, aarch64/arm64



english
[русский](#)

[news](#)
[about](#)
[download](#)
[security](#)
[documentation](#)
[faq](#)
[books](#)
[community](#)
[enterprise](#)

[x.com](#)
[blog](#)

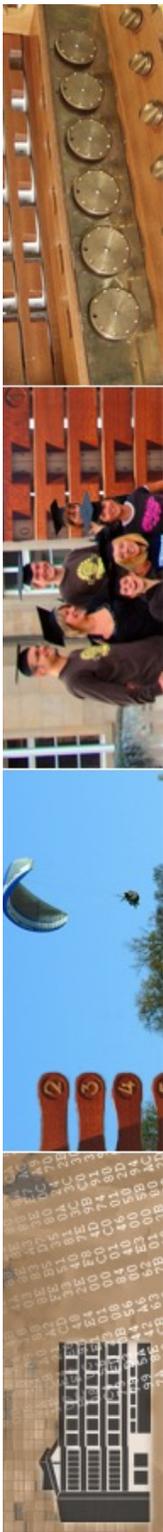
[unit](#)
[njs](#)





manuelle Installation

- ähnlich zu Apache
 - (als fertiges Linux-Paket)
 - aus den Source-Files mit
 - configure
 - make
 - make install



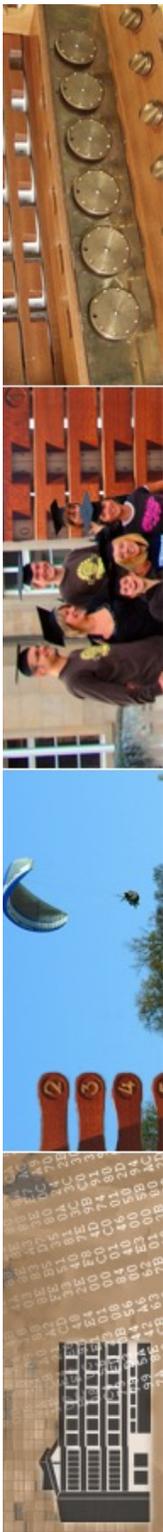


Building nginx from Sources

The build is configured using the `configure` command. It defines various aspects of the system, including the methods nginx is allowed to use for connection processing. At the end it creates a `Makefile`. The `configure` command supports the following parameters:

- `--prefix=path` — defines a directory that will keep server files. This same directory will also be used for all relative paths set by `configure` (except for paths to libraries sources) and in the `nginx.conf` configuration file. It is set to the `/usr/local/nginx` directory by default.
- `--sbin-path=path` — sets the name of an nginx executable file. This name is used only during installation. By default the file is named `prefix/sbin/nginx`.
- `--conf-path=path` — sets the name of an `nginx.conf` configuration file. If needs be, nginx can always be started with a different configuration file, by specifying it in the command-line parameter `-c file`. By default the file is named `prefix/conf/nginx.conf`.
- `--pid-path=path` — sets the name of an `nginx.pid` file that will store the process ID of the main process. After installation, the file name can always be changed in the `nginx.conf` configuration file using the [pid](#) directive. By default the file is named `prefix/logs/nginx.pid`.
- `--error-log-path=path` — sets the name of the primary error, warnings, and diagnostic file. After installation, the file name can always be changed in the `nginx.conf` configuration file using the [error_log](#) directive. By default the file is named `prefix/logs/error.log`.
- `--http-log-path=path` — sets the name of the primary request log file of the HTTP server. After





```
zrvwa01@infodienste =>  
zrvwa01@infodienste => ./configure --help
```

```
--help                print this message  
  
--prefix=PATH        set installation prefix  
--sbin-path=PATH     set nginx binary pathname  
--modules-path=PATH  set modules path  
--conf-path=PATH     set nginx.conf pathname  
--error-log-path=PATH set error log pathname  
--pid-path=PATH      set nginx.pid pathname  
--lock-path=PATH     set nginx.lock pathname  
  
--user=USER          set non-privileged user for  
worker processes  
--group=GROUP        set non-privileged group for  
worker processes  
  
--build=NAME         set build name  
--builddir=DIR       set build directory  
  
--with-select_module enable select module  
--without-select_module disable select module  
--with-poll_module   enable poll module  
--without-poll_module disable poll module  
  
--with-threads       enable thread pool support  
  
--with-file-aio       enable file AIO support  
  
--with-http_ssl_module enable ngx_http_ssl_module  
--with-http_v2_module enable ngx_http_v2_module  
--with-http_realip_module enable ngx_http_realip_module  
--with-http_addition_module enable ngx_http_addition_module  
--with-http_xslt_module enable ngx_http_xslt_module  
--with-http_xslt_module=dynamic enable dynamic ngx_http_xslt_module  
--with-http_image_filter_module enable ngx_http_image_filter_module  
--with-http_image_filter_module=dynamic enable dynamic ngx_http_image_filter_module  
  
--with-http_geoip_module enable ngx_http_geoip_module  
--with-http_geoip_module=dynamic enable dynamic ngx_http_geoip_module  
--with-http_sub_module enable ngx_http_sub_module  
--with-http_dav_module enable ngx_http_dav_module  
--with-http_flv_module enable ngx_http_flv_module  
--with-http_mp4_module enable ngx_http_mp4_module
```



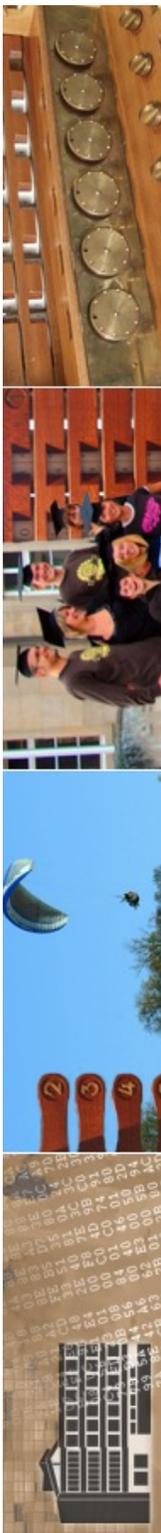


```
[thomas@Vaux =>
[thomas@Vaux => more myconfig
./configure
    --prefix=/Users/thomas/temp/nginx-1.9.6
    --user=thomas
    --group=staff
    --with-threads
    --with-http_perl_module
    --with-perl=/usr/bin/perl
    --without-http_rewrite_module
```

Configuration summary

- + using threads
- + PCRE library is not used
- + OpenSSL library is not used
- + using builtin md5 code
- + sha1 library is not found
- + using system zlib library

```
nginx path prefix: "/Users/thomas/temp/nginx-1.9.6"
nginx binary file: "/Users/thomas/temp/nginx-1.9.6/sbin/nginx"
nginx configuration prefix: "/Users/thomas/temp/nginx-1.9.6/conf"
nginx configuration file: "/Users/thomas/temp/nginx-1.9.6/conf/nginx.conf"
nginx pid file: "/Users/thomas/temp/nginx-1.9.6/logs/nginx.pid"
nginx error log file: "/Users/thomas/temp/nginx-1.9.6/logs/error.log"
nginx http access log file: "/Users/thomas/temp/nginx-1.9.6/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"
```



```

zrvwa01@infodienste =>
zrvwa01@infodienste => more myconfig_nginx
./configure \
--prefix=/home/zrvwa01/nginx_test \
--user=zrvwa01
zrvwa01@infodienste =>
zrvwa01@infodienste =>
checking for zlib library ... found
creating objs/Makefile

Configuration summary
+ using system PCRE library
+ OpenSSL library is not used
+ using system zlib library

nginx path prefix: "/home/zrvwa01/nginx_test"
nginx binary file: "/home/zrvwa01/nginx_test/sbin/nginx"
nginx modules path: "/home/zrvwa01/nginx_test/modules"
nginx configuration prefix: "/home/zrvwa01/nginx_test/conf"
nginx configuration file: "/home/zrvwa01/nginx_test/conf/nginx.conf"
nginx pid file: "/home/zrvwa01/nginx_test/logs/nginx.pid"
nginx error log file: "/home/zrvwa01/nginx_test/logs/error.log"
nginx http access log file: "/home/zrvwa01/nginx_test/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"

```

```

zrvwa01@infodienste =>
checking for poll() ... found
checking for /dev/poll ... not found

```





```
zrvwa01@infodienste =>
zrvwa01@infodienste => make
make -f objs/Makefile
make[1]: Verzeichnis „/home/zrvwa01/nginx_install/nginx-1.19.5“ wird betreten
gcc -c -pipe -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g -I src/core -I src/event -I src/event/modules -I src/os/unix -I objs \
    -o objs/src/core/nginx.o \
    src/core/nginx.c
gcc -c -pipe -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g -I src/core -I src/event -I src/event/modules -I src/os/unix -I objs \
    -o objs/src/core/nginx_log.o \
    src/core/nginx_log.c
gcc -c -pipe -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g -I src/core -I src/event -I src/event/modules -I src/os/unix -I objs \
    -o objs/src/core/nginx_palloc.o \
    src/core/nginx_palloc.c
gcc -c -pipe -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g -I src/core -I src/event -I src/event/modules -I src/os/unix -I objs \
    -o objs/src/core/nginx_array.o \
    src/core/nginx_array.c
gcc -c -pipe -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g -I src/core -I src/event -I src/event/modules -I src/os/unix -I objs \
    -o objs/src/core/nginx_list.o \
    src/core/nginx_list.c
```





```
zrvwa01@infodienste =>
zrvwa01@infodienste => make install
make -f objs/Makefile install
make[1]: Verzeichnis „/home/zrvwa01/nginx_install/nginx-1.19.5“ wird betreten
test -d '/home/zrvwa01/nginx_test' || mkdir -p '/home/zrvwa01/nginx_test'
test -d '/home/zrvwa01/nginx_test/sbin' \
    || mkdir -p '/home/zrvwa01/nginx_test/sbin'
test ! -f '/home/zrvwa01/nginx_test/sbin/nginx' \
    || mv '/home/zrvwa01/nginx_test/sbin/nginx' \
        '/home/zrvwa01/nginx_test/sbin/nginx.old'
cp objs/nginx '/home/zrvwa01/nginx_test/sbin/nginx'
test -d '/home/zrvwa01/nginx_test/conf' \
    || mkdir -p '/home/zrvwa01/nginx_test/conf'
cp conf/koi-win '/home/zrvwa01/nginx_test/conf'
cp conf/koi-utf '/home/zrvwa01/nginx_test/conf'
cp conf/win-utf '/home/zrvwa01/nginx_test/conf'
test -f '/home/zrvwa01/nginx_test/conf/mime.types' \
    || cp conf/mime.types '/home/zrvwa01/nginx_test/conf'
cp conf/mime.types '/home/zrvwa01/nginx_test/conf/mime.types.default'
test -f '/home/zrvwa01/nginx_test/conf/fastcgi_params' \
    || cp conf/fastcgi_params '/home/zrvwa01/nginx_test/conf'
cp conf/fastcgi_params \
    '/home/zrvwa01/nginx_test/conf/fastcgi_params.default'
test -f '/home/zrvwa01/nginx_test/conf/fastcgi.conf' \
    || cp conf/fastcgi.conf '/home/zrvwa01/nginx_test/conf'
```





```
zrvwa01@infodienste =>
zrvwa01@infodienste => ll
insgesamt 8
drwxr-xr-x 11 zrvwa01 142 Dez 1 21:22 ./
drwx---r-x 24 zrvwa01 4096 Dez 1 21:20 ../
drwx----- 2 zrvwa01 6 Dez 1 21:22 client_body_temp/
drwxr-xr-x 2 zrvwa01 4096 Dez 1 21:20 conf/
drwx----- 2 zrvwa01 6 Dez 1 21:22 fastcgi_temp/
drwxr-xr-x 2 zrvwa01 38 Dez 1 21:09 html/
drwxr-xr-x 2 zrvwa01 55 Dez 1 21:22 logs/
drwx----- 2 zrvwa01 6 Dez 1 21:22 proxy_temp/
drwxr-xr-x 2 zrvwa01 18 Dez 1 21:09 sbin/
drwx----- 2 zrvwa01 6 Dez 1 21:22 scgi_temp/
drwx----- 2 zrvwa01 6 Dez 1 21:22 uwsgi_temp/
zrvwa01@infodienste =>
```





Starting, Stopping, and Reloading Configuration

To start nginx, run the executable file. Once nginx is started, it can be controlled by invoking the executable with the `-s` parameter. Use the following syntax:

```
nginx -s signal
```

Where *signal* may be one of the following:

- `stop` — fast shutdown
- `quit` — graceful shutdown
- `reload` — reloading the configuration file
- `reopen` — reopening the log files

For example, to stop nginx processes with waiting for the worker processes to finish serving current requests, the following command can be executed:

```
nginx -s quit
```

This command should be executed under the same user that started nginx.

Changes made in the configuration file will not be applied until the command to reload configuration is sent to nginx or it is restarted. To reload configuration, execute:

```
nginx -s reload
```





It works!

```
zrvwa01@infodienste =>  
zrvwa01@infodienste =>  
zrvwa01@infodienste => ./nginx  
zrvwa01@infodienste =>
```





Konfiguration

- zentrale Konfigurationsdatei ist **`nginx.conf`**
im Unterordner `conf`
- hat nur **117 Zeilen...**





compilieren: Apache versus nginx

Apache 2.4.51

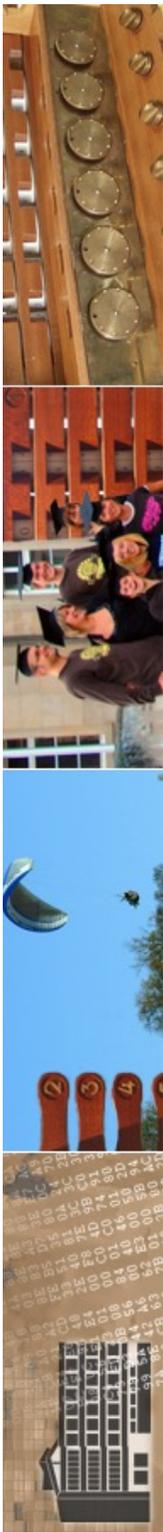
nginx 1.22.1

- `make | wc -l`

- `make | wc -l`

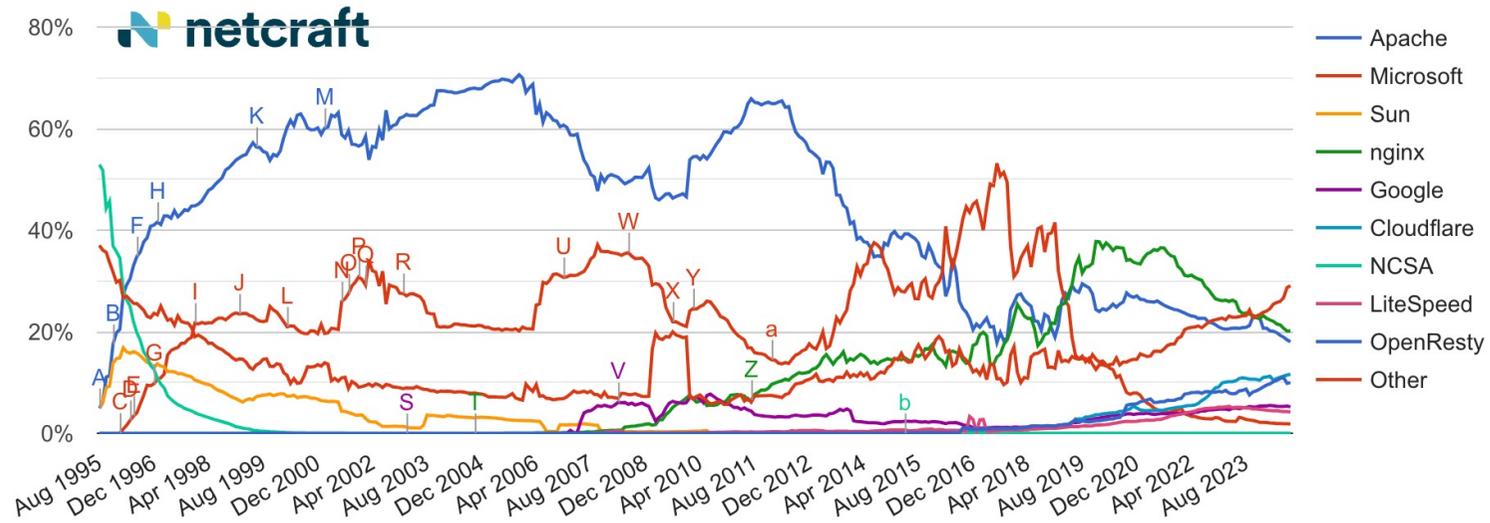
527

491



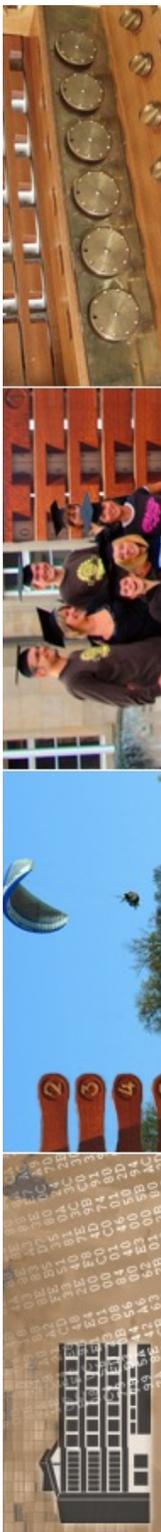


Web server developers: Market share of all sites



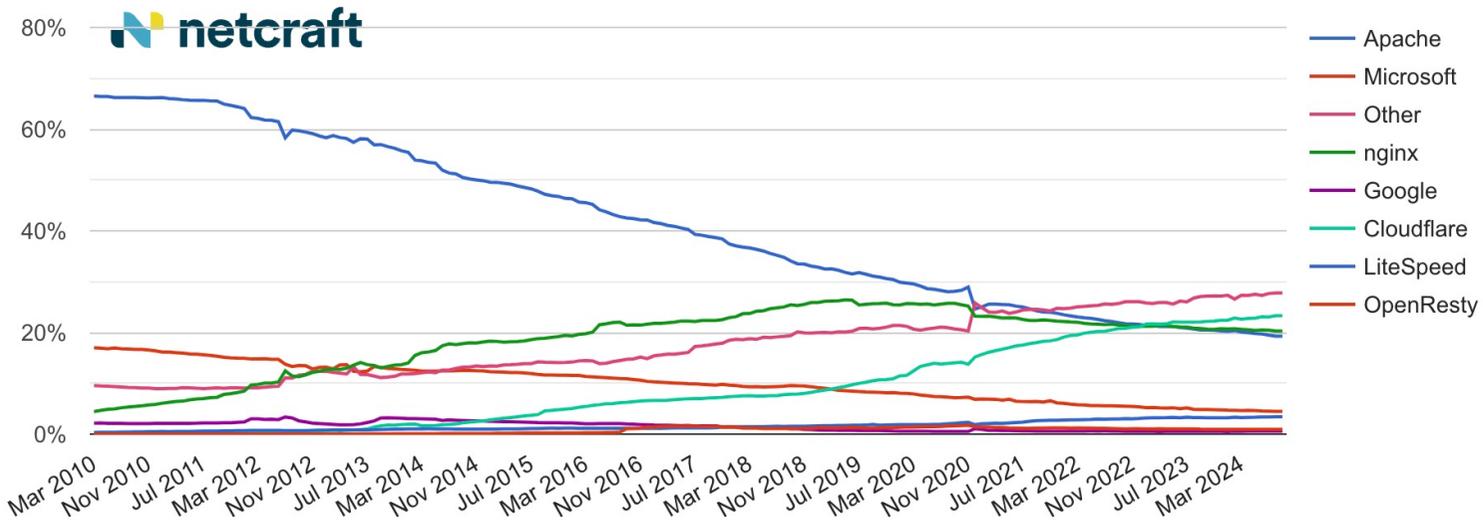
Developer	August 2024	Percent	September 2024	Percent	Change
nginx	223,025,645	20.13%	225,640,032	20.16%	0.03
Apache	203,825,341	18.40%	201,390,151	18.00%	-0.40
Cloudflare	127,028,522	11.47%	130,093,325	11.63%	0.16
OpenResty	108,954,196	9.84%	111,723,893	9.98%	0.15

<https://news.netcraft.com/archives/category/web-server-survey/>





Web server developers: Market share of the top million busiest sites

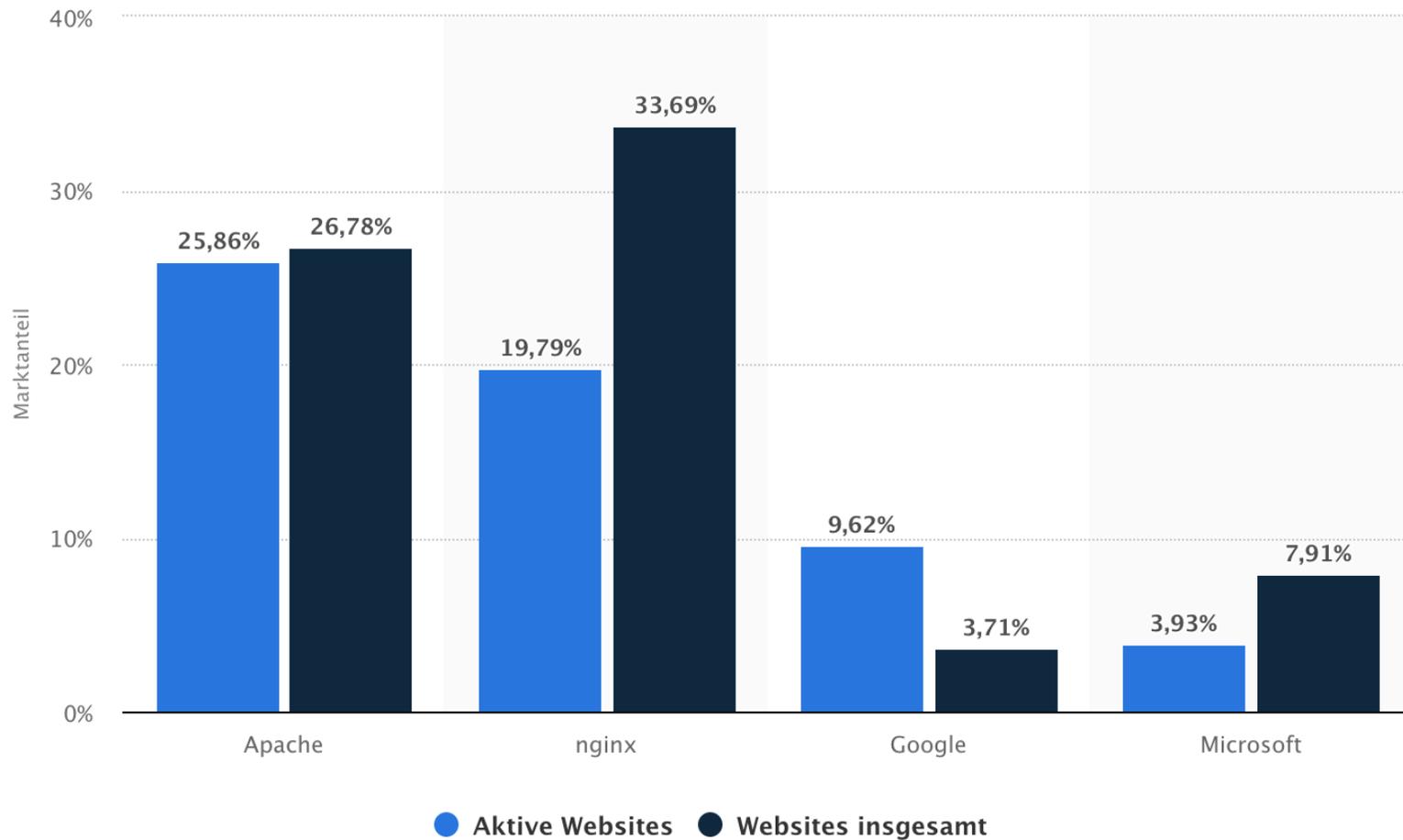


Developer	August 2024	Percent	September 2024	Percent	Change
Cloudflare	232,823	23.28%	232,767	23.28%	-0.01
nginx	202,769	20.28%	202,880	20.29%	0.01
Apache	192,880	19.29%	192,821	19.28%	-0.01
Microsoft	44,580	4.46%	44,538	4.45%	-0.00





Marktanteile der führenden Webserver weltweit im November

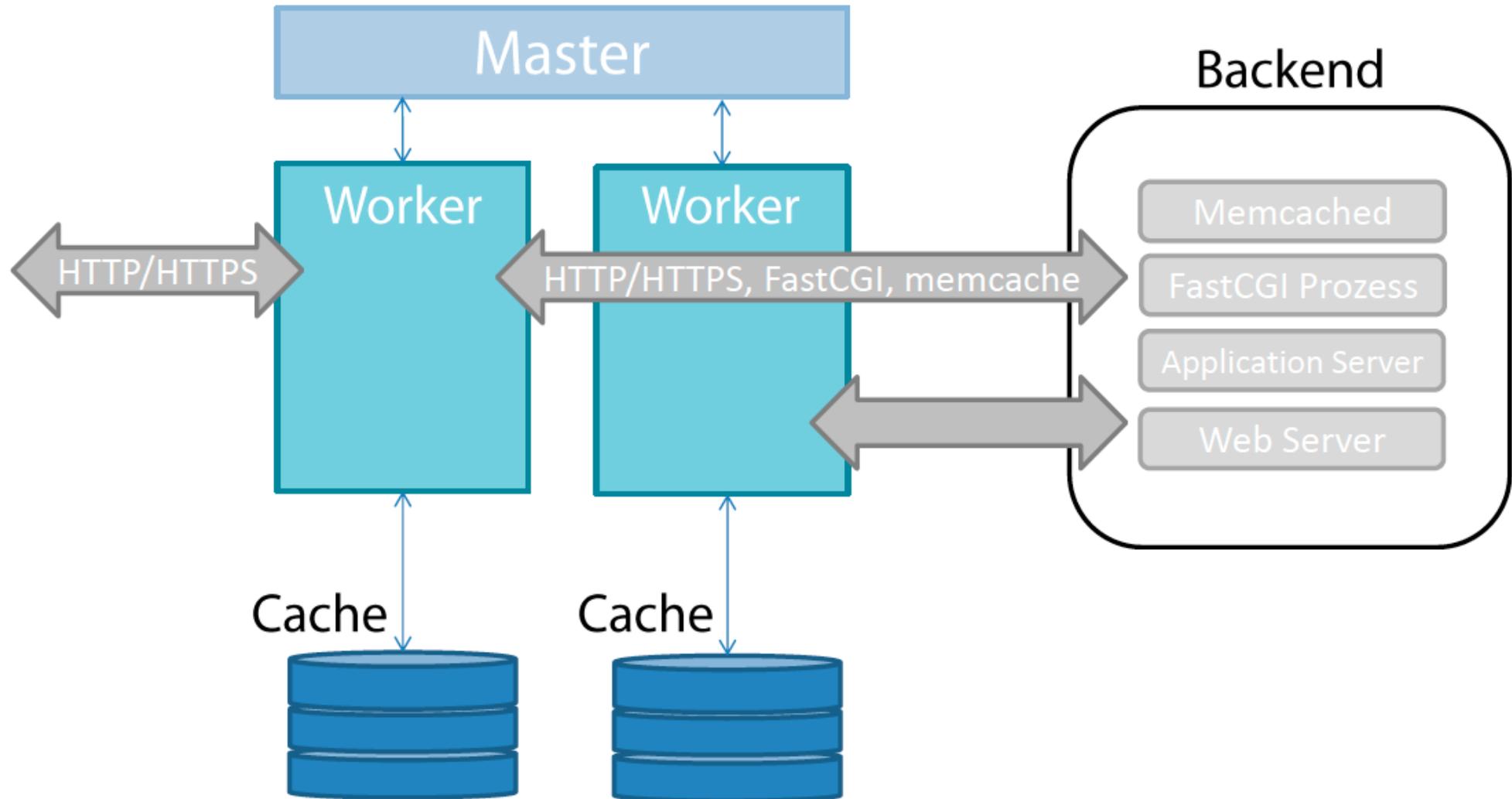


Details: Weltweit

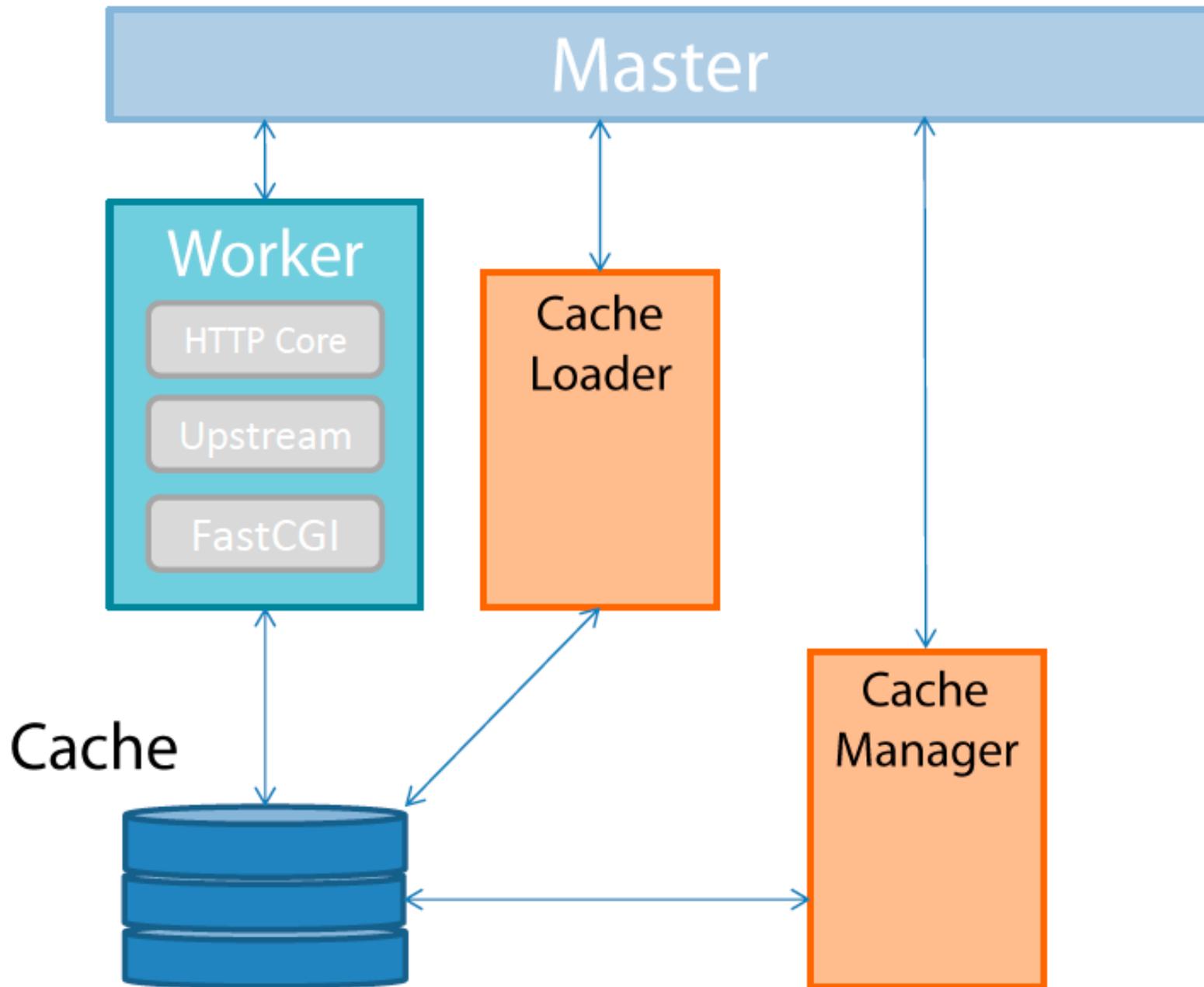
© Statista 2020



Nginx Architektur



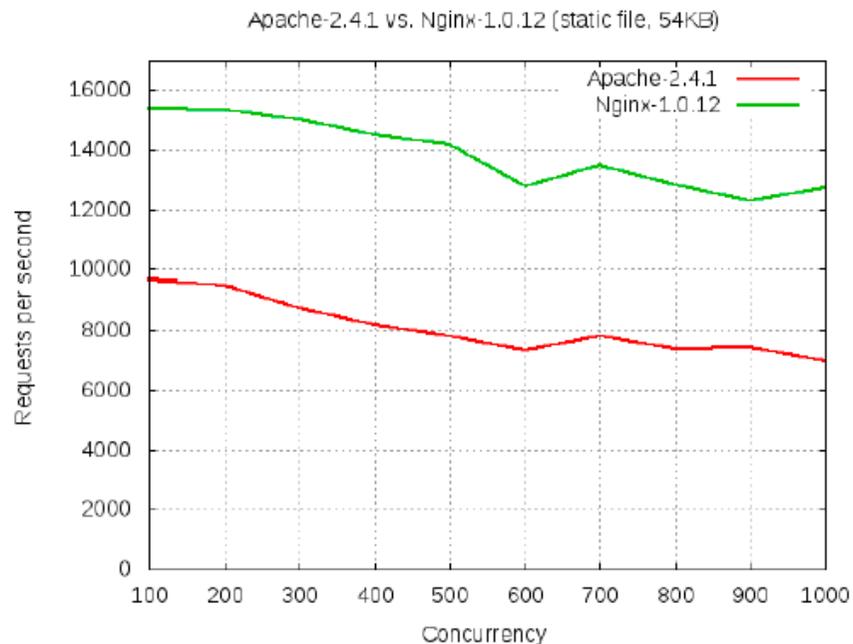
Nginx Architektur Cache



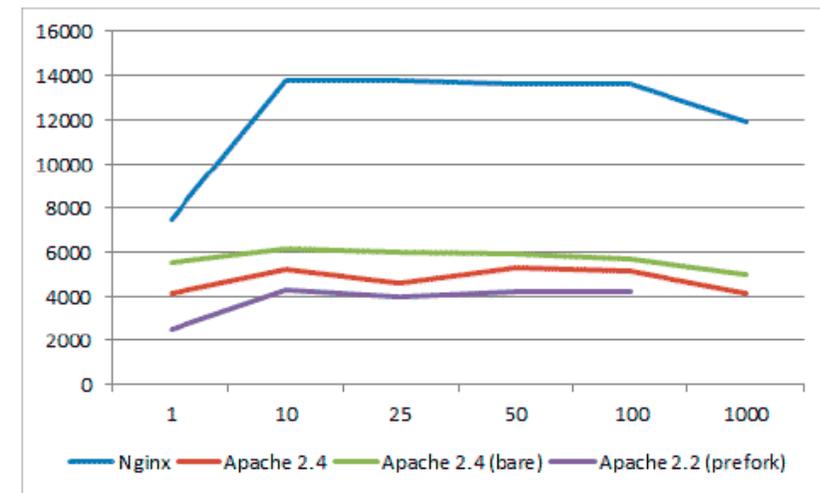


Benchmarks

- Webserver-Benchmarks: schwierig wegen zahlreicher Parameter
- NGINX vorteilhaft bei großer Nutzerzahl und *statischem Inhalt*



<http://tengine.taobao.org/images/benchmark2.png>

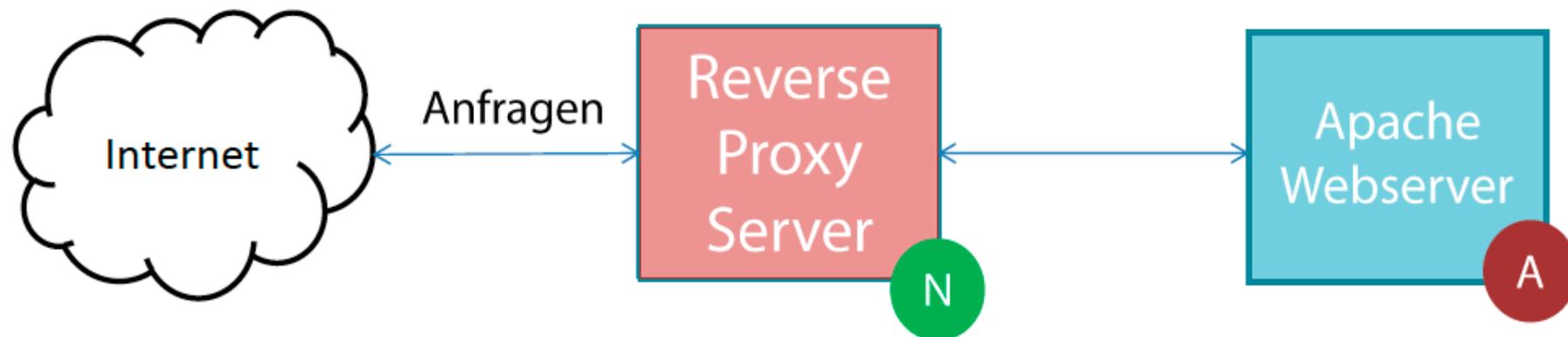


<http://www.eschrade.com/page/performance-of-apache-2-4-with-the-event-mpm-compared-to-nginx/>



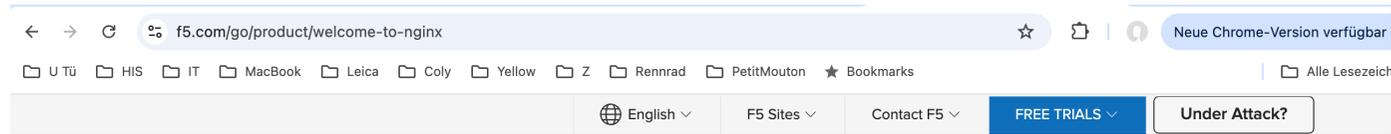
Hybrid-Model: Apache und NGINX

- NGINX: schnelle Auslieferung statischer Inhalte
- Apache: dynamische Inhalte, komplexere Anwendungen, Application Server





kommerzielle Version



SOLUTIONS ▾ PRODUCTS ▾ PARTNERS ▾ RESOURCES ▾ SUPPORT ▾ COMPANY ▾ GET F5 ▾

🔍 Search all F5 sites

F5 NGINX

**Optimize, deliver, and secure
apps across the entire
enterprise with NGINX**



What is NGINX?

NGINX has evolved from a web server to a comprehensive platform for app delivery, optimization, and security in Kubernetes environments. Now, with the SaaS-based web console NGINX One, enterprises can manage web traffic, load balancing, API gateway capabilities, and security in a single, easy-to-use package.

Products

Use Cases



F5 NGINX One

NGINX One takes the core NGINX



F5 NGINX App Protect

NGINX App Protect is a



F5 NGINXaaS for Azure

NGINXaaS for Azure is an



Enterprise

Free

Data Plane



NGINX Plus



Load Balancing



API Gateway



Content Cache



NGINX App Protect



Web Application Firewall



Denial of Service

Data Plane



NGINX Open Source



Web Server



Reverse Proxy



NGINX Unit



App Server



Reverse Proxy

Control Plane



NGINX Management Suite *Instance Manager*



Instance Discovery



Configuration Management



NGINX Ingress Controller



Kubernetes Ingress



Open Source Edition Available

Control Plane



NGINX Service Mesh



Lightweight Mesh



Flexible & Portable

Management Plane



NGINX Management Suite *API Connectivity Manager*



Developer Portal



API Gateway Management

Management Plane



NGINX Amplify



Monitoring



Observability



Analytics



njs scripting language

← → ↻ nginx.org/en/docs/njs/ ☆ | Neue Chrome-Version verfügbar

U Tü HIS IT MacBook Leica Coly Yellow Z Rennrad PetitMouton ★ Bookmarks | Alle Lesezeich

Celebrating [20 years](#) of nginx! Read about our journey and milestones in the [latest blog](#).

nginx JavaScript module

njs is an nginx module that extends the server's functionality through JavaScript scripting, enabling the creation of custom server-side logic and [more](#).

- [Download and install](#)
- [Changes](#)
- [Reference](#)
- [JavaScript Engine](#)
- [Examples](#)
- [Security](#)
- [Compatibility](#)
- [Command-line interface](#)
- [Understanding preloaded objects](#)
- [Tested OS and platforms](#)
- [ngx_http_js_module](#)
- [ngx_stream_js_module](#)
- [Writing njs code using TypeScript definition files](#)
- [Using node modules with njs](#)

Use cases

- Complex access control and security checks in njs before a request reaches an upstream server
- Manipulating response headers
- Writing flexible asynchronous content handlers and filters

See [examples](#) for more njs use cases.

nginx 20 Years

english
[русский](#)

[news](#)
[about](#)
[download](#)
[security](#)
[documentation](#)
[faq](#)
[books](#)
[community](#)
[enterprise](#)

[x.com](#)
[blog](#)

[unit](#)
[njs](#)



Caddy



← → ↻ 📄 caddyserver.com ☆ 🗄️ | 🗨️ Neue Chrome-Version verfügbar

📁 U Tü 📁 HIS 📁 IT 📁 MacBook 📁 Leica 📁 Coly 📁 Yellow 📁 Z 📁 Rennrad 📁 PetitMouton ★ Bookmarks | 📁 Alle Lesezeich

A ZeroSSL Project 🗄️ Store 🗨️ Forum 🐙 GitHub ⚙️ Theme: System

caddy[®] Documentation Features Account Support [Download](#) [Sponsor](#)

THE ULTIMATE SERVER

makes your sites more **secure**, more **reliable**, and more **scalable** than any other solution.

[Download](#) [Docs](#) [Star](#) 58,487

```
GNU nano 7.2 Caddyfile Modified
caddy.dyanim.com

file_server
log

# load balance API requests between two backends
```

Help **Write Out** **Where Is** **Cut**
Exit **Read File** **Replace** **Paste**

00:13

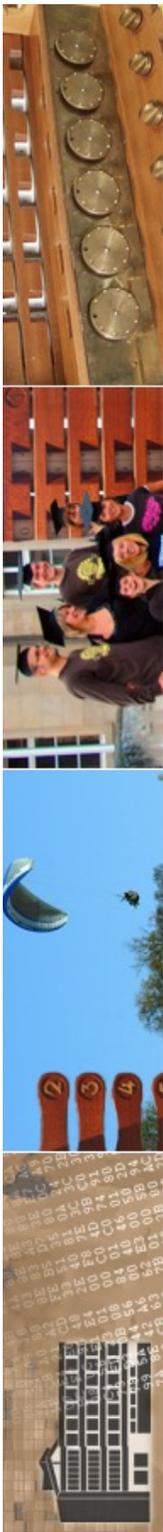
Watch in real-time as Caddy serves HTTPS in < 1 minute.





Caddy

- Hauptsächlich für statische Inhalt
 - Kann auch Dynamik mit fastCGI
 - Einfache Nutzbarkeit
 - Automatische SSL Zertifikate mit Let's Encrypt
 - Support von IPV6, HTTP/2
 - Performance für meiste Seiten ähnlich zu Nginx





Caddy

- Installation maximal einfach
 - Download
 - Entpacken
 - `./caddy` ausführen
 - Aufrufen der Seiten über `localhost:2015`

- Erweiterungen aktuell in Entwicklung





Caddy

```

134.2.2.38 - PuTTY
zrskk01@infodienste:~/caddy$ ll
insgesamt 29888
-rwxr-xr-x 1 zrskk01 benutzer 15241963 Okt 20 03:28 caddy
-rw-r--r-- 1 zrskk01 benutzer 15306752 Nov 16 22:59 caddy_linux_amd64_custom.tar
-rw-r--r-- 1 zrskk01 benutzer 13218 Sep 28 21:07 CHANGES.txt
-rw-r--r-- 1 zrskk01 benutzer 6 Nov 16 23:04 index.html
drwxr-xr-x 6 zrskk01 benutzer 97 Sep 28 21:07 init
-rw-r--r-- 1 zrskk01 benutzer 25261 Sep 28 21:07 LICENSES.txt
-rw-r--r-- 1 zrskk01 benutzer 994 Sep 28 21:07 README.txt
zrskk01@infodienste:~/caddy$ ./caddy
Activating privacy features... done.
http://:2015
  
```





The Caddyfile

This page describes how to configure Caddy using the Caddyfile.

Introduction

The term "Caddyfile" describes a text file that changes how Caddy works. It's similar in purpose to `httpd.conf` or `nginx.conf`. The Caddyfile file can be named anything, but by default, Caddy will look for a file called `Caddyfile` in the current directory. You can specify another location for the Caddyfile using the `-conf` [flag](#):

```
$ caddy -conf="/path/to/Caddyfile"
```

If your Caddyfile is within the root of your site, don't worry. Caddy will respond with "404 Not Found" to keep it hidden for you.

Syntax

The Caddyfile always starts with the address of the site to serve:

```
localhost:2020
```

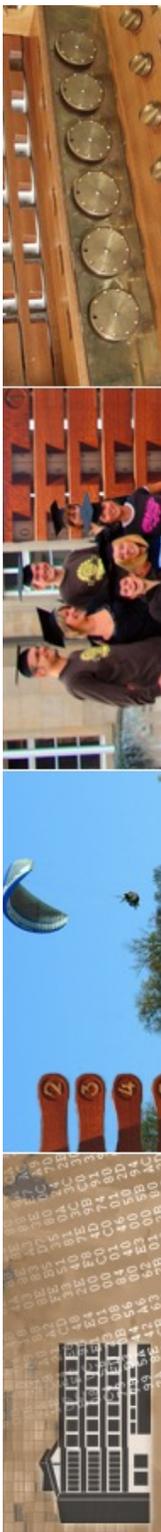




Addresses

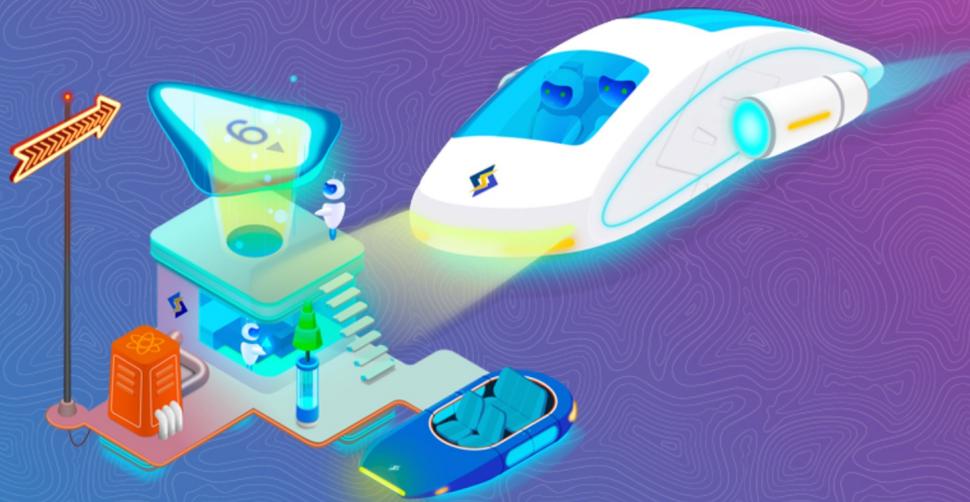
Addresses are specified in the form `scheme://host:port/path`, where all but one are optional. The host portion is usually localhost or the domain name. The default port is 2015 (unless the site qualifies for [automatic HTTPS](#), in which case it's 443). The scheme portion is another way to specify a port. Valid schemes are "http" or "https" which represent, respectively, ports 80 and 443. If both a scheme and port are specified, the port will override the scheme. For example:

```
:2015           # Host: (any), Port: 2015
localhost       # Host: localhost, Port: 2015
localhost:8080  # Host: localhost, Port: 8080
example.com     # Host: example.com, Port: 443
http://example.com # Host: example.com, Port: 80
https://example.com # Host: example.com, Port: 443
http://example.com:1234 # Host: example.com, Port: 1234
https://example.com:80 # Error! HTTPS on port 80
*.example.com   # Hosts: *.example.com, Port: 443
example.com/foo/ # Host: example.com, Port: 443, Path: /foo/
/foo/          # Host: (any), Port: 2015, Path: /foo/
```





LiteSpeed Web Server Enterprise v6.3



Major Features:

- Namespace containers
- Cgroup resource limits
- Advanced anti-DDoS
- Firewall controller
- Improved HTTP2/HTTP3 engine

[Learn More Now](#)

The FASTEST WordPress Stack



...und nun...

- haben wir zwei aktuelle, sehr performante Web-Server kennen gelernt:
nginx und Caddy
- als nächstes:
fastCGI und ServerSideIncludes

