Instalace a nastavení webového serveru

Funkce, protokoly, porty, logy

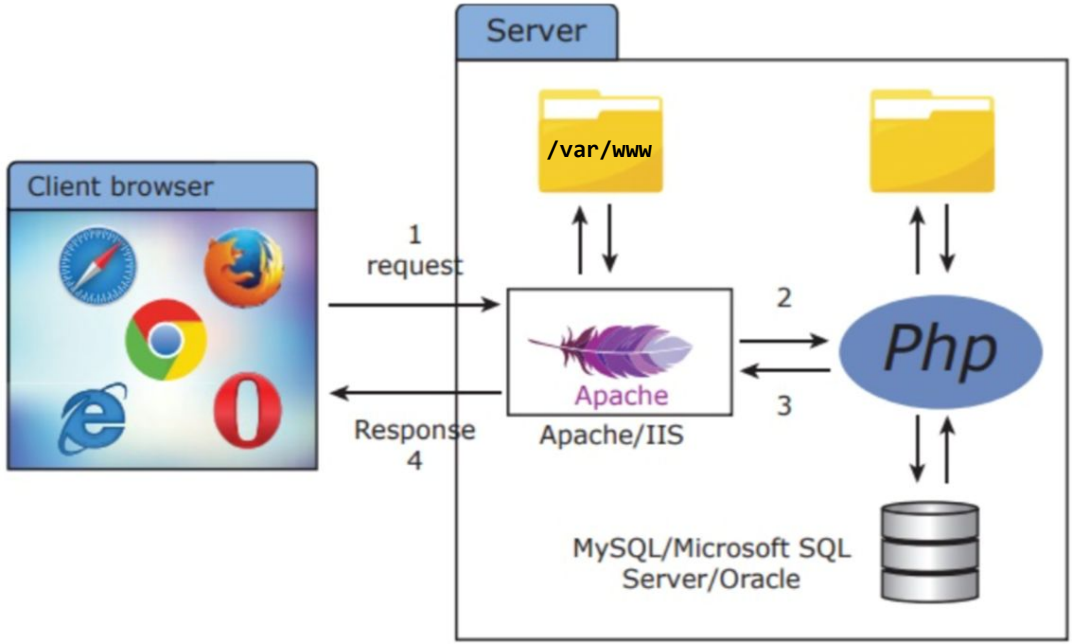
Webový server

Webový server je počítač připojený k počítačové síti.

Přijímá a vyřizuje požadavky na určených portech ve tvaru HTTP.

- Někdy pošle soubor uložený na disku, např. **index.html** nebo jiný soubor (dokument, obrázek…),
- jindy vytváří obsah až při zpracování požadavku (stránku **vygenerovanou skriptem PHP**).

Ať už ale odpověď pochází odkudkoli, **vždy se přenáší jako datový blok** - tedy něco, co se po síti chová **jako soubor** (sekvence bajtů).



🌐 Kdy to začalo

První webový server na světě byl CERN **httpd** a běžel na systému **NeXTSTEP** (předchůdce macOS).

Byl spuštěn v roce **1989** a hostoval tento obsah:

http://info.cern.ch/hypertext/WWW/TheProject.html

Za celým projektem stojí "otec" webu Sir Timothy "Tim" Bernes-Lee, emeritní ředitel konsorcia W3C, která dohlíží na rozvoj webu a tvorbu webových standardů.

První webový server světa

Současnost

Webových serverů je mnoho, např.:

- **Apache HTTP Server**
- **Nginx Web Server**
- **GWS** – Google Web Server
- **MS IIS** – Microsoft Internet Information Services
- **Lighttpd** – má velmi nízké nároky na HW, vhodný např. pro Raspberry Pi nebo Orange Pi
- **SJSAS** (Sun Java System Web Server)
- **Small HTTP server**

..a kolem patnácti dalších.

Apache HTTP Server

- Většina webhostingů nabízí právě tento.
- Domovská stránka projektu: httpd.apache.org
- Apache HTTP Server je softwarový webový server s otevřeným kódem pro Linux, BSD, UNIX, macOS i Windows.
- Apache je velmi konzervativní software, kde nové věci přibývají velmi málo.
- Ve verzi 2.4 je je už od roku 2012. Jeho funkcionalita se rozšiřuje zejména pomocí modulů (mods).

Versions of Apache HTTP Server

Version	Initial release	Latest release
1.3	1998-06-06[56]	2010-02-03 (1.3.42)[57]
2.0	2002-04-06[58]	2013-07-10 (2.0.65)[59]
2.2	2005-12-01[60]	2017-07-11 (2.2.34)[61]
2.4	2012-02-21[62]	2024-07-17 (2.4.62)[63]

Old version **Latest version**

Instalace

Instalaci webového serveru Apache provedeme příkazem

```
$ sudo apt install apache2
```

Jelikož jsou v provozu stále webové servery ve verzi 1.3, je "současný" Apache označen číslovkou 2 (aby se to nepletlo).

Není to nic exotického, jde o běžnou praxi. Např. Python také stále existuje ve dvou verzích: Původní Python 2 (python) a současný Python 3 (python3).

Server Apache se po instalaci **automaticky spustí jako služba** (démon).

Služba je jednoduše aplikace, která v počítači na pozadí trvale běží, a navíc **nemá přidělený terminál** → nemá přidělené proudy (stdin, stdout, stderr).

Služba – démon

V Linuxu (a obecně v unixových systémech) se službám říká démoni (anglicky daemons).

V 60. letech 20. století, když vědci z MIT pracovali na prvních operačních systémech, hledali vhodné označení pro procesy, které běží na pozadí a tiše vykonávají svou práci – například tisk, komunikaci po síti nebo správu úloh. Inspirovali se pojmem „démon“ z řecké mytologie (daimon), který **označuje neviditelnou bytost, jež pomáhá lidem vykonávat různé úkoly**.

Stejně tak v počítači démoni "neviditelně" pracují na pozadí (dělají svou práci), například:

- **sshd** zajišťuje přihlášení přes SSH,
- **cron** spouští úlohy podle času,
- **cupsd** obsluhuje tiskárny.

⚠️ Počítačový démon nemá nic společného s „ďáblem“.

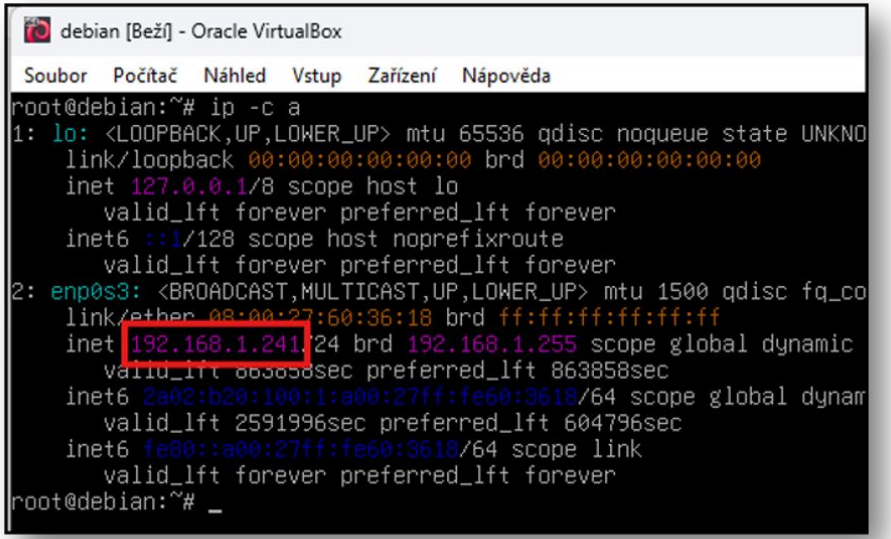
IP adresa serveru

Server **TuX** má **pevnou** a **veřejnou** IP adresu **217.195.162.12** a FQDN **tux.panska.cz**.

Server s webem **Panské** má **pevnou** a **veřejnou** IP adresu **217.195.162.4**, FQDN **www.panska.cz** a jeho název je **valbom.panska.cz**.

Náš server má **dynamickou** a **neveřejnou** IP adresu. Doménové jméno nemá, neboť není registrovaný v systému DNS. Každý má (musí mít) IP adresu jinou. Zjistíš ji příkazem:

```
$ ip a
```



To je to 4 bajtové fialové číslo za **inet** u rozhraní **enp0s3**.

Bude to něco jako **192.168.21.nnn**

Rychlá kontrola v prohlížeči

Pokud chceš obsah serveru zobrazit, spusť webový prohlížeč (Edge, Chrome, Safari...) a jako adresu zadej IP adresu serveru, na kterém běží Apache.

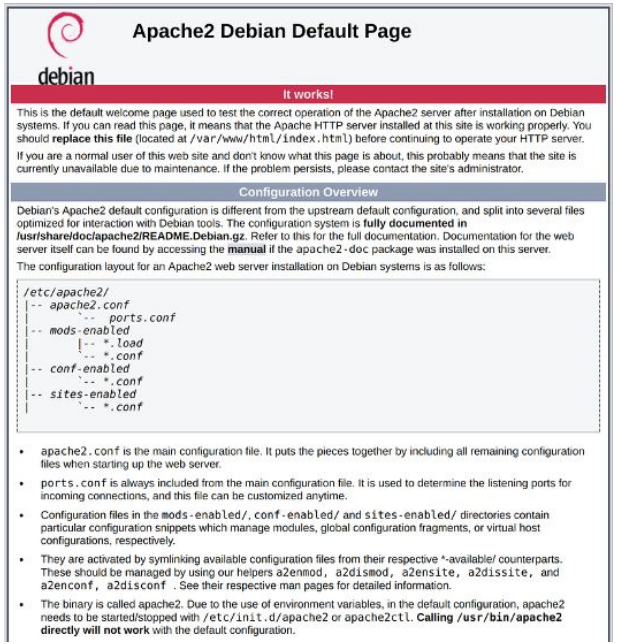
`http://192.168.21.nnn`

Apache v Debianu obsahuje testovací webovou stránku. Ve výchozím nastavení je webový obsah umístěn v adresáři

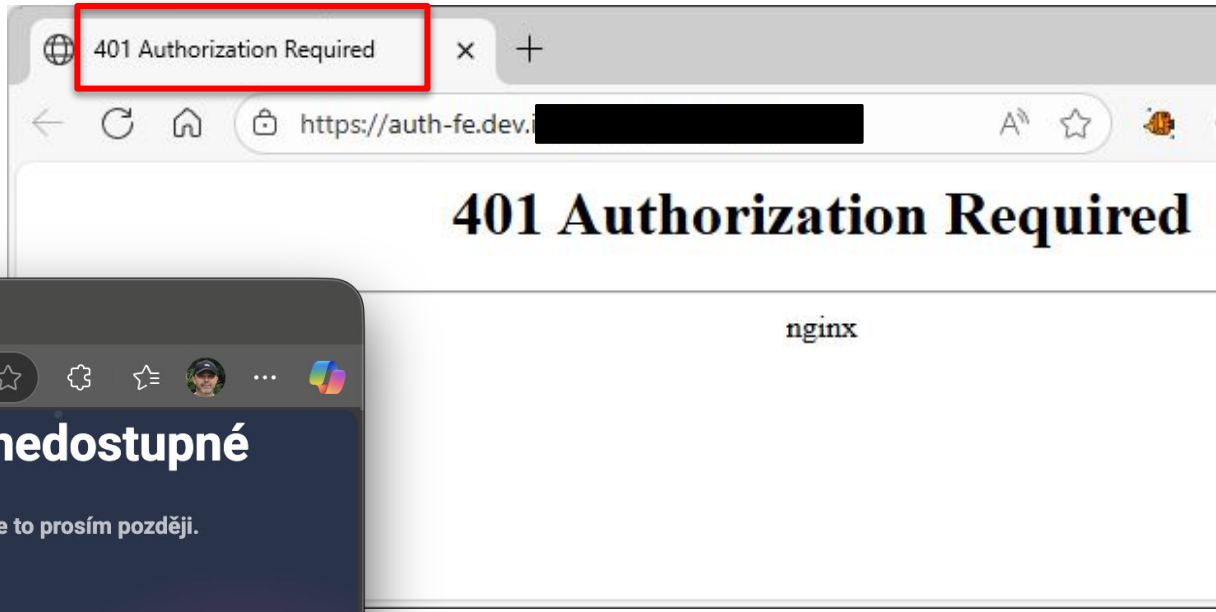
`/var/www/`**`html`**

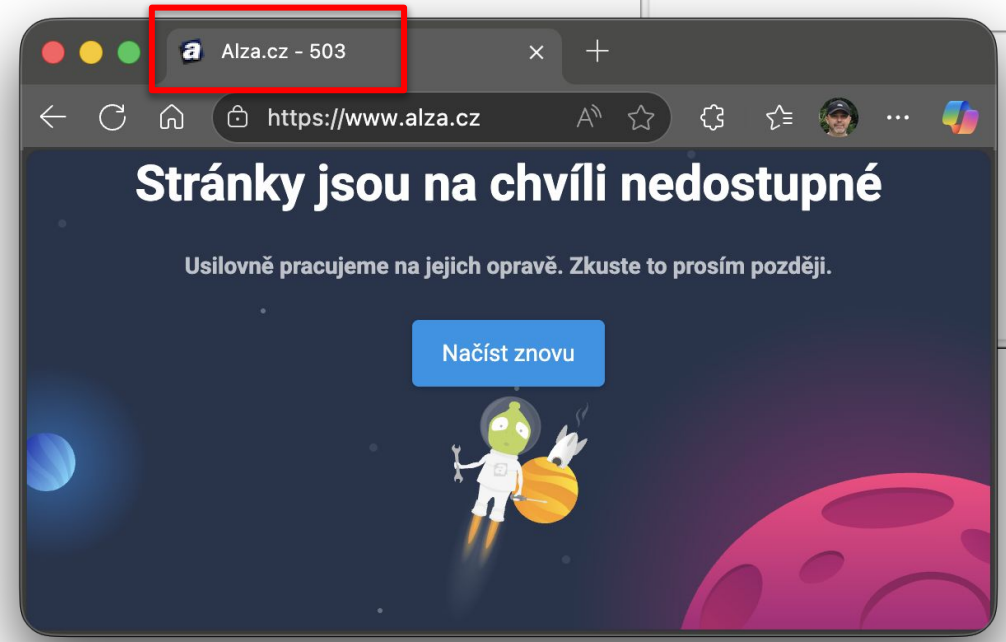
Všimni si, že nepožadujeme žádný konkrétní soubor. Apache v takovém případě hledá v kořenovém adresáři `/var/www/html` soubor s názvem "index" - `index.html` nebo `index.php`.

- Pokud ho najde, pošle ho,
- pokud ne, zobrazí seznam souborů kořenového adresáře.



Stavové kódy





Stavové kódy

Stavový kód odpovědi udává, zda byl požadavek vyřízen v pořádku, či zda došlo k nějakým problémům. Kódy jsou **trojciferná čísla** a dělí se do **pěti skupin**:

- **1xx Informační**
- **2xx Success**
  - 200 OK - Standardní odpověď pro úspěšný HTTP požadavek.
- **3xx Redirect**
- **4xx Client Error**
  - 401 Authorization Required – vyžadována autorizace, ale nebyla zatím provedena
  - 404 Not Found - Požadovaný zdroj nebyl nalezen.
  - 403 Forbidden - Uživatel nemá oprávnění k přístupu k požadovanému zdroji.
- **5xx Server Error**
  - 500 Internal Server Error – Obecná chybová zpráva.
  - 503 Service Unavailable – Služba je dočasně nedostupná.

Seznam všech kódů je tady.

Příkaz `curl`

```
$ curl -I <URL>
```

Příkaz `curl` s přepínačem `-I` slouží k odeslání **HTTP HEAD** požadavku na zadanou URL adresu. Získá pouze hlavičky odpovědi serveru (bez těla obsahu), což je užitečné například pro:

- Zjištění typu obsahu (např. Content-Type).
- Kontrolu stavu HTTP odpovědi
  - HTTP/1.1 200 OK, HTTP/2 200 OK
  - 404 Not Found apod.
- Ověření přesměrování (300 Redirect).
  - HTTP/2 302
- Získání dalších metadat, jako je velikost (Content-Length) nebo doba platnosti (Expires).

Tento příkaz je **vhodný pro rychlé ověření informací o serveru bez stahování celého obsahu**.

`curl`: instalace a zkouška

```
$ sudo apt install curl
```

```
$ curl -I http://192.168.21.nnn
$ curl -I http://192.168.21.nnn/index.html
```

```
$ curl -I https://linux.edumach.cz/lamp
```

```
$ curl -I http://www.panska.cz/index.php
$ curl -I https://www.panska.cz/index.php
$ curl -I https://www.panska.cz/kontakty
$ curl -I https://www.panska.cz/kontakty2
```

Co jsi zjistil(a)?

Vývoj protokolu HTTP

I protokol HTTP se průběžně vyvíjí - od jednoduchého přenosu textu k modernímu, rychlému a bezpečnému přenosu dat. Každá verze přináší zlepšení v rychlosti, efektivitě a spolehlivosti komunikace mezi klientem a serverem.

- ➔ **HTTP** (1990) Úplně první verze (CERN, Švýcarsko). Uměl přenášet jen čistý text HTML, bez hlaviček. Každý požadavek = jedna stránka, nic víc.
- ➔ **HTTP/1.0** (1996) Přidány hlavičky (např. typ obsahu, délka, stavový kód). Podpora různých typů dat (nejen HTML). Každý požadavek měl stále vlastní spojení → pomalejší.
- ➔ **HTTP/1.1** (1997) **Dnes nejrozšířenější verz**e. Zavádí trvalá spojení (keep-alive) – více požadavků přes jedno spojení. Přidána komprese, cache, lepší práce s proxy…
- ➔ **HTTP/2** (2015) Postaveno na binárním formátu (ne textovém). Umožňuje více požadavků současně přes jedno spojení (multiplexing). Lepší rychlost, efektivita a komprese hlaviček. Stav.
- ➔ **HTTP/3** (2022) Vyvíjí Google. Využívá protokol QUIC (TCP nahrazuje UDP). Rychlejší navazování spojení, lepší práce v mobilních sítích. Vhodné pro moderní webové aplikace a streamování. Využívá jej Google pro videa na YouTube.

Vysvětlení protokolů TCP a UDP

TCP – protokol, na kterém běží současné verze HTTP, vznikl už v 90. letech 20. století. Tehdy byly datové linky pomalé a často nestabilní. Část přenášených paketů se mohla cestou ztratit, proto byl TCP navržen tak, aby **každý přijatý paket potvrzoval. Teprve po potvrzení je odeslán další**. Tím je přenos spolehlivý, ale pomalejší a více zatěžuje síť.

UDP – tento protokol nevyžaduje potvrzování přijatých paketů. **Server odešle všechna data bez čekání na odpovědi klienta**. Přenos je rychlejší, ale méně spolehlivý. příklady služeb:

- **DNS** – dotazy na překlad doménových jmen (rychlá odezva, malý objem dat)
- **DHCP** – přidělování IP adres v síti
- **VoIP** (např. internetové hovory, Teams, Skype) – důležitější je plynulost než dokonalá spolehlivost
- **Online hry** – rychlá komunikace mezi hráči a serverem
- **Streamování videa a zvuku v reálném čase** (např. živé přenosy) – malé zpoždění má větší význam než úplná bezchybnost

Toto je pouze stručné porovnání. Podrobnosti se dozvíš v předmětu DS (Datové sítě).

Porty

Webový server naslouchá na portu **80** (protokol **HTTP**) nebo **443** (protokol **HTTP** s šifrovací podvrstvou **SSL** – čili **HTTPS**). Před odesláním se data zašifrují a teprve poté odešlou protokolem HTTP. Jediný rozdíl je v tom, že HTTPS cestují data šifrovaná, HTTP v otevřené (čitelné) podobě. Čísla portů byla stanovena dohodou

HTTP (80) – nešifrovaný přenos (čitelné):

```
GET /index.html HTTP/1.1
Host: www.example.com

<p>ahoj</p>
```

HTTPS (443)– šifrovaný přenos (nečitelné):

```
ê9¥£Zý#¬…øßÜœå§þ¬¶Ç{òØpð¨øÁß¤¾…¬
```

Port 443 si nevyzkoušíme - instalace certifikátu **Let's Encrypt SSL** je sice triviální (zabere 5 minut), ale vyžaduje funkční doménu (něco jako `example.com`) a tu my nemáme.

Příkaz `nmap`

Příkaz `nmap` umí otestovat otevřené porty. Nainstaluj balík balík `nmap` a otestuj nejprve na svém serveru a poté na serveru TuX nebo i jiném:

```
$ sudo apt install nmap
```

```
$ nmap 192.168.21.nnn
$ nmap tux.panska.cz
```

```
$ nmap tux.panska.cz
Starting Nmap 7.93 ( https://nmap.org ) at 2025-10-30 18:38 CET
Nmap scan report for tux.panska.cz (127.0.1.1)
Host is up (0.00029s latency).
rDNS record for 127.0.1.1: TuX.panska.cz
Not shown: 996 closed tcp ports (conn-refused)
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

Příklad výpisu serveru TuX

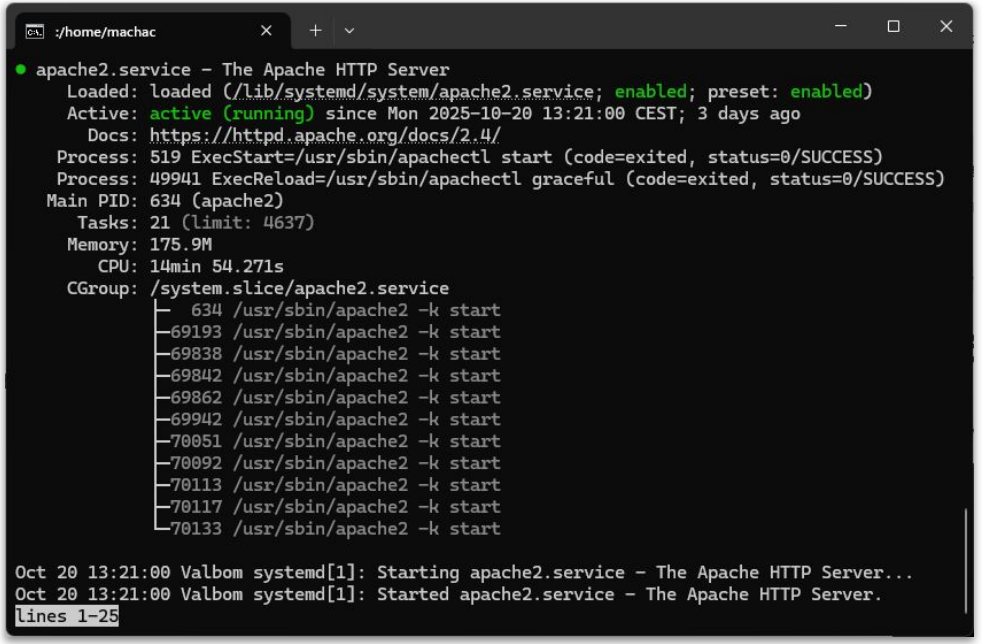
Řízení běhu Apache

Příkazem

```
$ systemctl status apache2
```

zobrazíš stav služby

Klávesa **Q** výpis ukončí.



Další příkazy pro řízení běhu

`$ sudo systemctl` **`stop`** `apache2` .. Zastaví webový server

`$ sudo systemctl` **`start`** `apache2` .. Spustí webový server

`$ sudo systemctl` **`restart`** `apache2` .. Restartuje webový server

`$ sudo systemctl` **`reload`** `apache2` .. Načte změněnou konfiguraci ("soft" restart bez výpadku služby)

Jen pro úplnost: Ve výchozím nastavení je Apache nakonfigurovaný na automatické spuštění po startu serveru. Toto chování jde změnit:

`$ sudo systemctl` **`disable`** `apache2` – Po restartu nespustí webový server.
`$ sudo systemctl` **`enable`** `apache2` – Zpět na automatické spouštění.

Běžně se to nepoužívá.

Protokolování požadavků

Apache protokoluje přijímané požadavky a taktéž zaznamenává případné chyby. To pomáhá správcům vytvářet statistiky a podle typu a množství požadavků optimalizovat obsah, způsob uložení i způsob prezentace požadovaných dat.

```
$ sudo tail /var/log/apache2/access.log
$ sudo tail /var/log/apache2/error.log
```

Instantní výpis posledních 10 řádků (tam jsou ty nejnovější záznamy) aktivuješ přepínačem `-f`. Zobraz si:

```
$ sudo tail -f /var/log/apache2/access.log
```

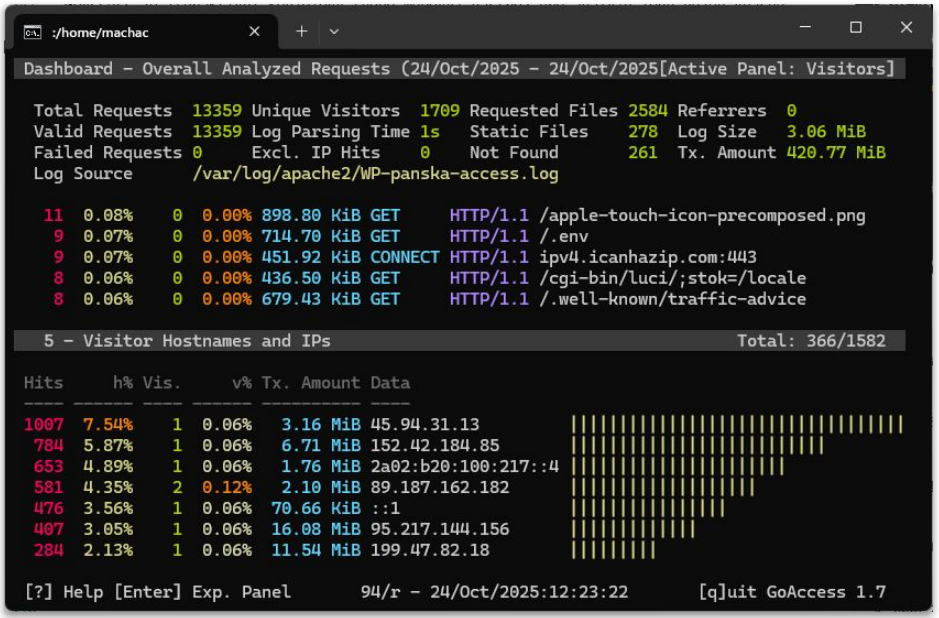
a několikrát aktualizuj webovou stránku. Sleduj výpis. Co vše se v něm zobrazuje? Výpis ukončíš **Ctrl+C**.

GoAccess

GoAccess je interaktivní konzolový (nebo webový) nástroj pro analýzu logů nejen Apache.

Instalace:

```
$ sudo apt install goaccess
```



GoAccess: Základní analýza

```
$ sudo goaccess /var/log/apache2/access.log --log-format=COMBINED
```

Pro pohyb v logu použij šipky ⬆️ a ⬇️. Pokud máš více souborů (rotace logů):

```
$ sudo zcat /var/log/apache2/access.log.*.gz | goaccess
--log-format=COMBINED
```

Generování HTML reportu:

```
$ sudo goaccess /var/log/apache2/access.log --log-format=COMBINED
-o /var/www/html/report.html
```

Report se uloží do `/var/www/html/report.html` – můžeš ho pak otevřít v prohlížeči `http://192.168.21.nnn/report.html`

Rotace logů

Během provozu serveru se soubory logů postupně zvětšují. Čistě technicky je tam posílá (přesměrovává pomocí >> Apache).

Rotace logů je proces, při kterém se staré soubory s logy automaticky uzavírají, přejmenovávají nebo komprimují a začne se zapisovat do nového logu.

Cílem je zabránit tomu, aby logovací soubory neustále rostly a nezabíraly příliš místa na disku.

```
root@Valbom:~# ls /var/log/apache2/
access.log       other_vhosts_access.log       WP-panska-access.log.8.gz
access.log.1     other_vhosts_access.log.1     WP-panska-access.log.9.gz
error.log        other_vhosts_access.log.2.gz  WP-panska-error.log
error.log.1      other_vhosts_access.log.3.gz  WP-panska-error.log.1
error.log.10.gz  WP-panska-access.log          WP-panska-error.log.10.gz
error.log.11.gz  WP-panska-access.log.1        WP-panska-error.log.11.gz
error.log.12.gz  WP-panska-access.log.10.gz    WP-panska-error.log.12.gz
error.log.13.gz  WP-panska-access.log.11.gz    WP-panska-error.log.13.gz
error.log.14.gz  WP-panska-access.log.12.gz    WP-panska-error.log.14.gz
error.log.2.gz   WP-panska-access.log.13.gz    WP-panska-error.log.2.gz
error.log.3.gz   WP-panska-access.log.14.gz    WP-panska-error.log.3.gz
error.log.4.gz   WP-panska-access.log.2.gz     WP-panska-error.log.4.gz
error.log.5.gz   WP-panska-access.log.3.gz     WP-panska-error.log.5.gz
error.log.6.gz   WP-panska-access.log.4.gz     WP-panska-error.log.6.gz
error.log.7.gz   WP-panska-access.log.5.gz     WP-panska-error.log.7.gz
error.log.8.gz   WP-panska-access.log.6.gz     WP-panska-error.log.8.gz
error.log.9.gz   WP-panska-access.log.7.gz     WP-panska-error.log.9.gz
```

Výpis souborů s logy webového serveru panska.cz

Jak rotace logů funguje

Po určité době (např. denně nebo týdně) nebo po dosažení určité velikosti se aktuální log uzavře. Dostane nové jméno, např.:

```
access.log.1
access.log.2.gz
```

Vytvoří se nový soubor `access.log`, do kterého se dál zapisuje. Staré logy se mohou automaticky mazat nebo komprimovat.

Rotaci logů provádí služba logrotate, která:

- se spouští automaticky (např. denně/týdně),
- má konfigurační soubory v adresáři `/etc/logrotate.d/`, (d = démon)

Jak rotace logů funguje

Umožňuje nastavit např.:

```
/var/log/apache2/*.log {
    weekly
    rotate 52
    compress
    missingok
    notifempty
}
```

Vysvětlení: jaké logy sledovat, rotovat každý týden, uchovávat 52 kopií (rok), komprimovat staré logy, neřešit chybějící soubory a přeskočit prázdné.

Vlastní "rotátor"

Pokud vytváříš vlastní logy (např. v nějakém skriptu nebo aplikaci), můžeš jim snadno přidat rotaci.

Stačí pro ně vytvořit nový soubor v adresáři

`/etc/logrotate.d/`

s podobnou strukturou.

```
machac@TuX  ~
$ ll /etc/logrotate.d/
total 40K
-rw-r--r-- 1 root root  120 Feb 14  2023 alternatives
-rw-r--r-- 1 root root  397 Apr 13  2023 apache2
-rw-r--r-- 1 root root  173 May 25  2023 apt
-rw-r--r-- 1 root root  130 Oct 14  2019 btmp
-rw-r--r-- 1 root root   82 May 26  2018 certbot
-rw-r--r-- 1 root root  112 Feb 14  2023 dpkg
-rw-r--r-- 1 root root  128 Sep 29  2023 exim4-base
-rw-r--r-- 1 root root  108 Sep 29  2023 exim4-paniclog
-rw-r--r-- 1 root root 1.9K Aug  3  2023 mariadb
-rw-r--r-- 1 root root  145 Oct 14  2019 wtmp
```

Ukázka serveru TuX