

# Zprovoznění vlastního webového serveru

Tuan Kiet Nguyen

27. dubna 2021

## Abstrakt

Tato práce zaměřuje především na obecný přehled struktury webového serveru a vysvětlení jeho součástí. Na úvod se věnuje na přesný příklad nezbytnosti webového serveru pro internet a principy, jak přijme a zpracuje data. Poté dojde ke charakteristice jednotlivých protokolů, částí a jejich funkce. V další části pozorování dochází ke konkrétní implementaci vlastního webového serveru, v nichž zahrnuje popis použitých hardwarů, konfigurace potřebných programů. Následně dochází ke sledování výsledku práce.

## 1 Úvod

V dnešní technologické době je určitě obrovským handicapem (nejen) pro podniky, konkrétně malé až středně velké, které nevyužívají sílu sítě. Viděl jsem ze vlastních zážitků podniky, kde zaměstnanec každodenně nese osobně dokumenty ke tisku svému kolegovi nebo v jednom asijském bistru křičí pokladník do kuchyně to, co zákazník objednal. Tyto přístupy přináší určitá rizika, v uvedených případech se jedná o časové ztráty. Někdy lze také vidět jejich neefektivnosti, například u kuchařů by došlo ke zmeškání zprávy. Jeden z řešení, jak tyto problémy odstranit, je budování vlastního síťového systému.

Jelikož se jedná o zapojení jen v rámci jedné lokality, zaměstnanci mezi oddíly mohou komunikovat a posílat důležitá data mnohem rychlejší a bezpečnější. Firmy si svá zařízení propojují přes Wifi či LAN a vyberou si aplikace v programovacím jazyce podle svého zájmu. Mezi nákladově nejlacinějšími a užitečnými způsoby patří zprovoznění vlastního webového serveru. Následně si společnost nechá naprogramovat webovou aplikaci v podporovaných jazycích např. Java, C#, PHP, React, Angular apod. Klíčovým důvodem tohoto řešení je to, že téměř každý přístroj, který má přístup k síti, má předem nainstalovaný webový prohlížeč. Díky tomu může společenství využít veškeré zařízení v jakékoli operačním systému, dokonce aplikace by mohla běžet i na starý notebook či tablet.

V sekci 2 se zaměřuje na podrobné charakteristice jednotlivých protokolů, jež se používá k provozu webového serveru.

V praktické části doloží konkrétní implementace v reálném životě. Sekce 3 se vrhne především na stručný postup, jak sestavit a konfigurovat webového serveru. Do podrobnosti se také věnuje popisy použitých nástrojů a balíčků.

Cílem této studie je představit způsob fungování webového serveru. Následně dochází k seznámení se základními principy, jak mezi počítačem serveru a uživatelem zpracuje a odesílá data.

## 2 Teoretická část

Pojem webový server v sobě skrývá 2 hlavní významy:

- je počítač nebo zařízení, které nabízí a spravuje své služby koncovým uživatelům. Na jeho pevném disku (HDD či SSD) jsou uloženy obsahy jednotlivé webové stránky ve formě souborů, zejména soubory HTML, CSS, Javascript, a jiné technologie určené k budování webu jako PHP, C# apod. Kromě toho slouží server klientům taktéž jako databázový systém. [viz. [1]]
- je program, který zpracuje požadavky od klientů. Mezi sebou komunikují přesně definovaným způsobem. V internetu se těmito pravidlům komunikace říká protokoly. Pro přenos webových stránek se používá protokol HTTP (HyperText Transfer Protocol) či HTTPS2.1 . [viz. [2]]

### 2.1 Princip fungování webového serveru 1

Webový server představuje jeden z nejdůležitějších součástí World Wide Web (www). Každý uživatel počítače přes něho jednoduše dostane k jakékoli dostupné webové stránce na internetu. Tento proces je možné díky tomu, že každý počítač / zařízení připojené k internetu je identifikován pomocí jedinečné *IP adresy* 2.2 (internetový protokol). Prostřednictvím této příležitosti se počítače navzájem komunikují. [viz. [3]]

Ke každé webové stránce se sděluje tzv. *domain name* 2.4 (doménové jméno). *DNS server* 2.4 k němu přiřazuje IP adresu webového serveru, ve kterém jsou uloženy soubory dané stránky.

Zadá-li klient do webového prohlížeče (ten zde vystupuje v roli klienta) odkaz URL (Uniform Resource Locator), například: „https://vse.cz/“, pošle dotaz na IP adresu webového serveru stránky „vse.cz“ na DNS server(y). Následně, kdyby ze DNS záznamu úspěšně našlo, obdrží webový server požadavek na zobrazení obsahu webové stránky „vse.cz“ prostřednictvím protokolu *HTTPS* (popř. *HTTP*)2.3 . Tento server klientovi zpětně odesílá obsah webu (články, obrázky, videa, ...) v HTML, Javascript a CSS a zobrazí ho v prohlížeči.

### 2.2 IP adresa

IP adresa představuje unikátní číslo, které jednoznačně identifikuje síťové rozhraní v počítačové síti, která používá IP protokol. Pojem síťového rozhraní se rozumí jako např.: Ethernet, Wi-Fi, IrDA port, virtuální zařízení. V současné době je nejrozšířenější IPv4, která používá 32bitové IP adresy, které jsou zapisovány dekadicky po jednotlivých oktetech (tj. po osmicích bitů), například 192.168.0.2. Z důvodu neočekávaného růstu technologie, byla v roce

2019 vyčerpána poslední adresa typu IPv4[10]. Postupně ji nahrazuje protokol IPv6, který používá 128bitové IP adresy zapsané hexadecimálně, například 2001:db8:0:1234:0:567:8:1. [viz [11] ]

## 2.3 HTTP, HTTPS

HTTP (Hypertext Transfer Protocol) je internetový protokol, jež představuje způsob přenosu hypertextových dokumentů (HTML, XML, ...) mezi webovým prohlížečem a vzdáleným WWW serverem. Posílá obvykle na portu 80. Společně s elektronickou poštou se stává HTTP nejpoužívanějším protokolem. Uživatel ho identifikuje přímo zadáním do URL. Komunikace přes HTTP je však otevřená a nezašifrovaná (text-plain), citlivé informace, jako heslo a bankovní transakce, může po cestě kdokoli vyloupit. K řešení tohoto problému bylo vyvinuto v květnu 2000 v rámci RFC 2818 [19] - HTTPS.[viz [6, 5] ]

K původnímu HTTP byly přidány kryptografické protokoly SSL/TLS k zašifrování přenosu dat. Přestože je jen jedna strana komunikace ověřená, typicky pouze server (např. uživatel potvrdí certifikát), HTTPS dokáže bez problému provádět. HTTPS protokoly u virtuálních serverů se rozlišují pomocí rozšíření *SNI*. Od května 2018 v rámci GDPR je HTTPS povinen pro všechny webové stránky, které zpracují osobní údaje. Striktně vyžaduje u všech eshopů, u důvěryhodných organizací (úřady, školy, banky,...), u stránek s přihlašovacími údaji, platebními metodami, apod. [viz [4, 7, 5] ]

## 2.4 Domain name a DNS

Výzkumem bylo dokázáno [12], že lidský mozek si pamatuje řetězec slov lépe než čtyř-oktetových čísel, příkladem je „Facebook.com“ s IP adresou 69.63.176.13. Z toho důvodu byly vyvíjeny tzv. *Internetové domény*. Domain name (internetová doména) je jedinečný identifikátor počítače nebo počítačové sítě, které jsou připojené k internetu. Skládá se z tzv. *domény nejvyšší úrovně* (Top-level Domain, TLD), jsou například .com, .cz, .org, atd., a z nižších úrovní tzv. *subdomény* (alza.cz, google.com, eshop.ruano.cz, atd.). Registraci a správu těchto domén má na starosti registrátoři domén. Mezinárodní domény nejvyššího řádu .com, .org, ... kontroluje společnost **IANA** (resp. **ICANN**), u nás v České republice doménu .cz zajišťuje **CZ.NIC z.s.p.o.** [viz [8]]

Pojem DNS (Domain Name System) představuje systém doménových jmen. Tento úkol řeší DNS servery. Překládá lidsky srozumitelná doménová jména na numerické síťové adresy. [9, 13] Funguje podobně jako telefonní seznam. To znamená, že místo toho, aby člověk si musel pamatovat desítky 9-ti místních telefonních čísel, stačí si zapamatovat jméno jeho vlastníka. V takovém případě bude telefonní číslo odpovídat IP adrese webové stránky, zatímco jméno vlastníka je název domény webové stránky. Když například do prohlížeče zadáme „www.google.com“, server DNS získá adresu serveru Google jako „74.125.236.37“. Kromě toho každý DNS server také uloží do své cache paměti názvy nalezených výsledků, který pomáhá uživatelům v příští vyhledávání získat odpověď mnohem snadnější a rychlejší.

DNS servery se dělí na:

- autoritativní - v něm se trvale uloží záznamy domén. Je jich obvykle více než dva a jsou provozovány registrátorem domény nebo poskytovatelem webhostingu. [viz [9, 14]]
- rekurzivní (caching only) - rekurzivně vyřeší DNS dotazy. Slouží hlavně pro rychlejší odpovědi a k ochraně přes zatížením autoritativního serveru. Většinou tyto servery provozuje ISP (poskytovatel připojení k internetu) [viz [9, 14]]

### 3 Praktická část

Tato sekce práce se zaměřuje na postup budování vlastního jednoduchého webového serveru. Následně dochází ke sledování jeho výsledku.

#### 3.1 Použité nástroje a hardware

- Raspberry Pi 4 Model B (4GB RAM) – Webový server podle potřeby jednotlivce lze nainstalovat na jakékoli počítači. Rozhodl jsem si zprovoznovat server na tento minipočítač hlavně proto, že podporuje operační systém Linux a běží na velmi nízký zdroj (5V=3A). Z toho důvodu bez prostředně ho nechám běžet 24 hodin denně. Kromě toho je velikostně velmi flexibilní, umístím ho pohodlně přímo za routerem. Jeho parametry jsou úplně dostačující k účelu provozu 4-5 webových stránek najednou.
- Samsung MicroSDXC 64 Gb SD karta Class 10 – Rychlost zápisu dat rozhoduje, jak rychle se načítá soubory. Tuto kartu jsem původně používal u bez-zrcadlovém fotoaparátu. Jeho zápis dosahuje okolo 90 MB za sekundu.
- Osobní notebook s Windows 10 pro vzdálené konfigurace.

#### 3.2 Postup a konfigurace

Na Pi 4 jsem nainstaloval Ubuntu Server 20.04.2 LTS, který neobsahuje žádné uživatelské rozhraní. Výhodou toho operačního systému je to, že patří mezi nejpopulárnějšími Linux distributorů, proto má obrovskou podporu z komunity a na vše, včetně programů, se neplatí. Pro stabilní a rychlejší připojení jsem ho připojil přímo do routeru síťovým kabelem CAT6. Hned po spuštění se musí zjistit jeho IP adresu. Získal jsem ji tedy přímo v konfiguraci routeru, protože je potřeba ihned pro dané zařízení nastavit statickou IP adresu, jestliže nechceme ji, například v situaci výpadku elektriny, ztratit. Následně v programu PuTTY v notebooku, po zadání IP adresy na portu 22, ovládám server přes SSH. Vše probíhá v správcovské rozhraní Terminal.

Jako hlavní program webového serveru jsem si vybral Apache, protože je jednak open-source a jednak patří mezi světově nejběžnějších s 50.9% rozsahem [viz [3]]. Samozřejmě lze vybrat jiného jako například Nginx.

Vzhledem k tomu, že jsem si naprogramoval aplikaci v PHP, potřeboval jsem tedy balíčky PHP. Nakonfiguroval jsem také svou databázi v programu MySQL. Přidal jsem na svůj serveru systém správy verzí Git, který mě hodně pomáhá při aktualizaci obsahu.

V tomto kroku si člověk rozhodne, zda bude chtít zveřejnit:

- 1 webovou stránku - Webové soubory se přenesou do adresáře `/var/www/html` (výchozí adresář Apache) buď pomocí FTP anebo přes Git. 3
- nebo více webových stránek - Tento výběr vyžaduje si vytvořit *virtuální hosty*. Nejprve jednotlivé stránky se uloží do podadresáře se jménem své domény. Vše se vloží do adresáře `/var/www/*`. Následně upravíme konfiguraci umístěné na `/etc/apache2/sites-available/000-název-domény.conf`. Podrobnosti lze dočíst na odkaz [17]. Našel jsem na Githubu velmi užitečný skript od uživatele RoverWire [15], který tyto předchozí instrukce sám provede a ušetří spoustu času, dokonce napsal i skript pro program Nginx. Jako příklad jsem založil doménu „semestralka.lan“. 4. Poté jsem vložil do adresáře `/var/www/semestralka.lan` soubor `phpinfo.php`. 5

Po instalaci nám zbývá upravit přístupová práva, chceme-li povolit přístup pro jiného uživatele.

V příloze doložím bash skripty, který jsem tvořil pro snadnější instalaci a správu webového serveru.

### 3.3 Výsledek práce

Momentálně na webové stránce vložené na server se dostane z jakékoli zařízení v rámci sítě. V routeru je předem nastaven automatické přesměrování, proto bez problému server vidí i z jiné podsítě. Stačí si do prohlížeče napsat IP adresu serveru.<sup>2</sup>

Jediný nepříjemný problém zůstává u vytvořených doménových jmen. Prohlížeč v notebooku automaticky začne doménu zeptat do veřejných rekurzivních DNS serverů, viz sekce 2.4 a rozhodně pro takový výsledek neexistuje žádný záznam. Z toho důsledku je potřeba upravit soubor *hosts* na zařízení, ve kterém chceme přesměrovat k našemu serveru<sup>6</sup>. Bohužel tento proces lze postupovat bez omezení jen na počítačových operačních systémech, u mobilních telefonů a tabletů s Androidem nebo iOS jen přes IP adresu serveru, jinak vyžaduje neoprávněný přístup do operačního systému jako rootování nebo jailbreak.

Testování stránky „semestralka.lan“ je proveden zadáním do prohlížeče 7 a nebo pomocí příkazu *ping*.<sup>8</sup>

Ačkoli tento server přenáší obsahy protokolem HTTP, pro soukromý účel není až tak velký problém. Samozřejmě kdybychom si zaplatili svým poskytovatelům internetu za veřejnou IP adresu a SSL certifikát, poté přesměrovali pomocí NAT, je potom náš server veřejně dostupný.

## 4 Závěr

Při provozu náročných webových aplikací doporučuji si vyřídit výkonnější počítač alespoň s 16 Gb RAM a SSD. Větší firmy, většinou u eshopů a bank, obvykle provozují svůj webový server, ve kterém drží pod svou kontrolou všechna data.

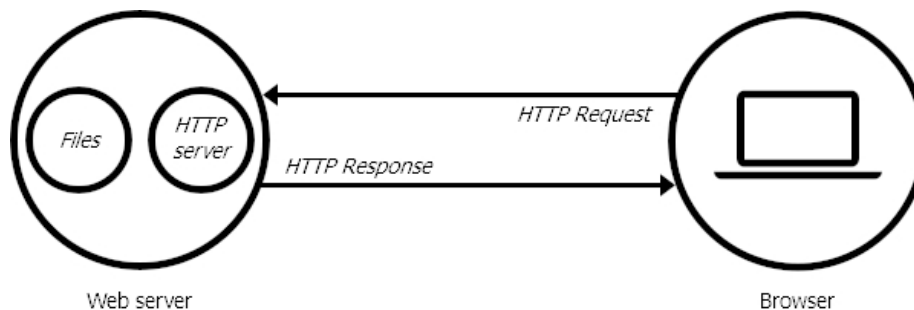
Tento server aplikuji přímo v mém vietnamském bistru. Naprogramoval jsem si v PHP program, který z pokladny posílá objednávku do tabletu v kuchyni. Současně na televizi u zákazníka zobrazí jeho pořadové číslo. Pomocí téhle příležitosti udržuji velmi komfortně a stabilně svůj podnik.

Odkaz na mého skriptu: <https://github.com/luckykiet/4iz110-Web-server.git>

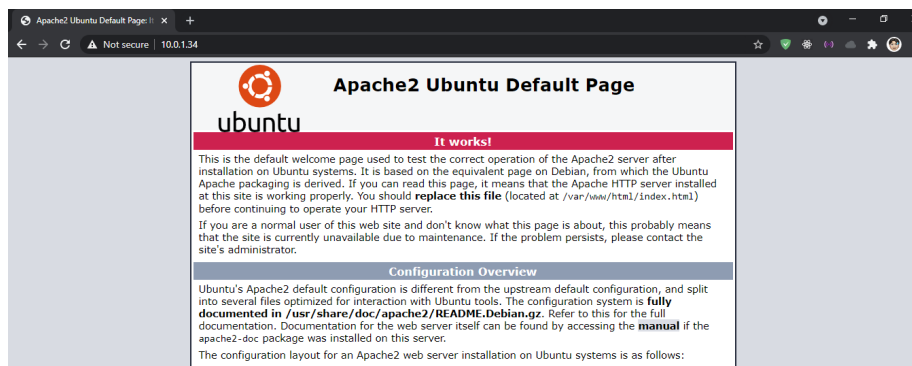
## Reference

- [1] . VASHIST, Shikha a Ayush GUPTA. Web Technology: Http, Web Server & Web Services. International Journal of Advanced Research in Computer Science [online]. 2014, vol. 5, no. 8.
- [2] *What is a web server.* Mozilla [online]. [cit. 2021-5-1]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server)
- [3] *Web server la gi? Hieu ro ve web server.* TopDev [online]. [cit. 2021-5-1]. Dostupné z: <https://topdev.vn/blog/web-server/>
- [4] *Protokol HTTPS.* TopDev [online]. [cit. 2021-5-1]. Dostupné z: <https://www.ssls.cz/https.html>
- [5] *HTTPS důležité nastavení - letos již nezbytné.* Absolutus [online]. [cit. 2021-5-1]. Dostupné z: <https://www.absolutus.cz/gdpr/https-dulezite-nastaveni-letos-jiz-nezbytnee>
- [6] *Hypertext Transfer Protocol.* Wikipedia [online]. [cit. 2021-5-1]. Dostupné z: [https://cs.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
- [7] *HTTPS.* Wikipedia [online]. [cit. 2021-5-1]. Dostupné z: <https://en.wikipedia.org/wiki/HTTPS>
- [8] *Internetová doména.* Wikipedia [online]. [cit. 2021-5-1]. Dostupné z: [https://cs.wikipedia.org/wiki/Internetov%C3%A1\\_dom%C3%A9na](https://cs.wikipedia.org/wiki/Internetov%C3%A1_dom%C3%A9na)
- [9] *DNS.* VADINSKÝ, Ondřej. [cit. 2021-5-1] 4IZ110-cv04-DHCP-TCP-UDP-DNS.pdf
- [10] *Very last IPv4 addresses to be assigned later this year.* SIDN [online]. [cit. 2021-5-2]. Dostupné z: <https://www.sidn.nl/en/news-and-blogs/very-las-ipv4-addresses-to-be-assigned-later-this-year>

- [11] *IP Adresa*. Wikipedia [online]. [cit. 2021-5-1]. Dostupné z: [https://cs.wikipedia.org/wiki/IP\\_adresa](https://cs.wikipedia.org/wiki/IP_adresa)
- [12] Remembering Numbers. The Memory Institute [online]. [cit. 2021-5-2]. Dostupné z: <http://www.thememoryinstitute.com/remembering-numbers.html>
- [13] LAPRADE, Craig. Domain Name Service Trust Delegation in Cloud Computing: Exploitation, Risks, and Defense. Ann Arbor: The George Washington University, 2021. Order No. 28156620. ISBN 9798691282607.
- [14] Domain Name System. Wikipedia [online]. [cit. 2021-5-2]. Dostupné z: [https://cs.wikipedia.org/wiki/Domain\\_Name\\_System](https://cs.wikipedia.org/wiki/Domain_Name_System)
- [15] <https://github.com/RoverWire/virtualhost>
- [16] <https://ubuntu.com/download/raspberry-pi>
- [17] <https://httpd.apache.org/docs/2.4/vhosts/examples.html>
- [18] <https://www.howtogeek.com/howto/27350/beginner-geek-how-to-edit-your-hosts-file/>
- [19] <https://whois.smartweb.cz/rfc/rfc2818/>



Obrázek 1: Komunikace mezi prohlížečem a serverem - zdroj: [2]



Obrázek 2: Ověření zobrazení webové stránky přes IP adresu na prohlížeči notebooku

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ ls /var/www/html  
index.html  
ubuntu@ubuntu:~$
```

Obrázek 3: Kontrola obsahu v adresáři /var/www/html

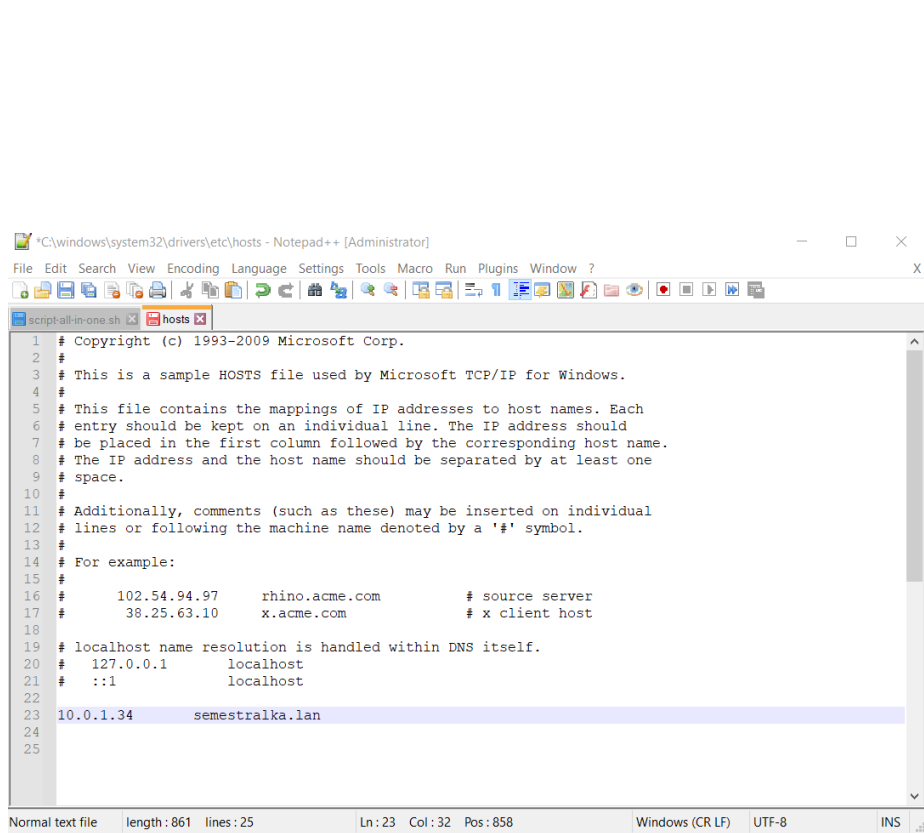


```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ sudo virtualhost create 'semestralka.lan' '/var/www/semestralka.lan'  
Added content to /var/www/semestralka.lan/phpinfo.php  
  
New Virtual Host Created  
  
Host added to /etc/hosts file  
  
Enabling site semestralka.lan.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
Reloading apache2 configuration (via systemctl): apache2.service.  
Complete!  
You now have a new Virtual Host  
Your new host is: http://semestralka.lan  
And its located at /var/www/semestralka.lan  
ubuntu@ubuntu:~$ ping -c4 semestralka.lan  
PING semestralka.lan (127.0.0.1) 56(84) bytes of data.  
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.080 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.101 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.113 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.100 ms  
  
--- semestralka.lan ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3060ms  
rtt min/avg/max/mdev = 0.080/0.098/0.113/0.011 ms  
ubuntu@ubuntu:~$
```

Obrázek 4: Založení nového virtuálního hostu s doménou „semestralka.lan“ pomocí skriptu [15]

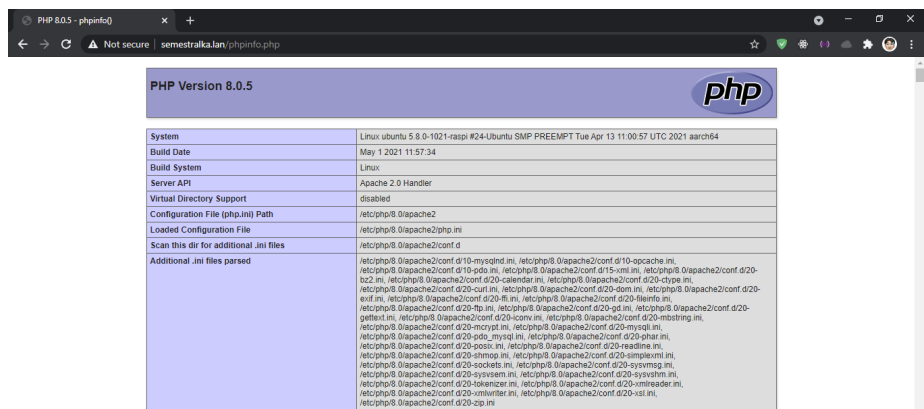
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ ls /var/www/semestralka.lan/  
phpinfo.php  
ubuntu@ubuntu:~$
```

Obrázek 5: Ověření obsahu nově vytvořené domény „semestralka.lan“



```
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97       rhino.acme.com   # source server
17 #       38.25.63.10       x.acme.com       # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1             localhost
22
23 10.0.1.34       semestralka.lan
24
25
```

Obrázek 6: Úprava souboru *hosts*<sup>[18]</sup> v notebooku



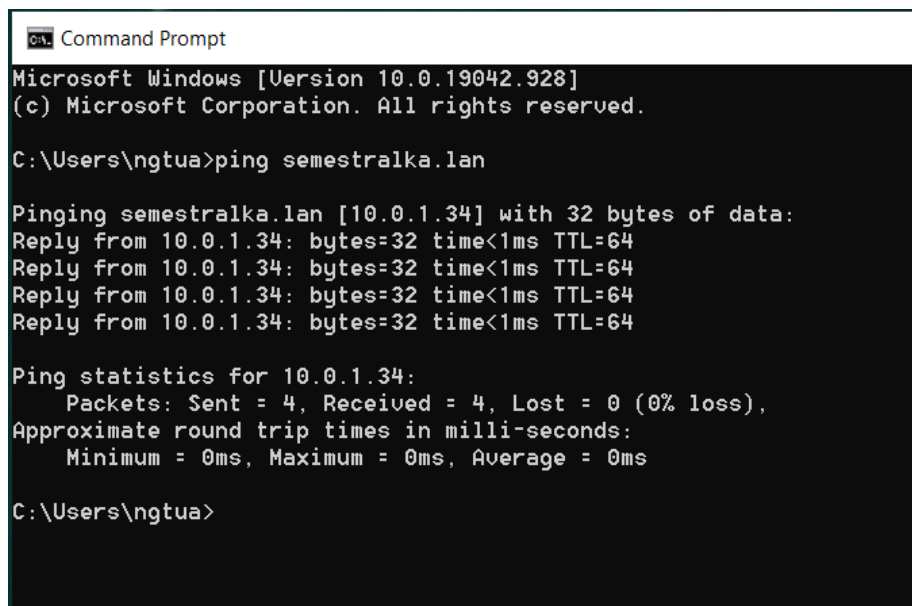
PHP 8.0.5 - phpinfo()

semestralka.lan/phpinfo.php

PHP Version 8.0.5

System	Linux ubuntu 5.8.0-1021-raspi #24-Ubuntu SMP PREEMPT Tue Apr 13 11:00:57 UTC 2021 aarch64
Build Date	May 1 2021 11:57:34
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.0/apache2
Loaded Configuration File	/etc/php/8.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.0/apache2/conf.d
Additional .ini files parsed	/etc/php/8.0/apache2/conf.d/10-mysqld.ini, /etc/php/8.0/apache2/conf.d/10-opcache.ini, /etc/php/8.0/apache2/conf.d/10-pdo.ini, /etc/php/8.0/apache2/conf.d/15-mb.ini, /etc/php/8.0/apache2/conf.d/20-bcmath.ini, /etc/php/8.0/apache2/conf.d/20-calendar.ini, /etc/php/8.0/apache2/conf.d/20-ctype.ini, /etc/php/8.0/apache2/conf.d/20-curl.ini, /etc/php/8.0/apache2/conf.d/20-dom.ini, /etc/php/8.0/apache2/conf.d/20-enchant.ini, /etc/php/8.0/apache2/conf.d/20-ffi.ini, /etc/php/8.0/apache2/conf.d/20-fileinfo.ini, /etc/php/8.0/apache2/conf.d/20-ftp.ini, /etc/php/8.0/apache2/conf.d/20-gd.ini, /etc/php/8.0/apache2/conf.d/20-gettext.ini, /etc/php/8.0/apache2/conf.d/20-iconv.ini, /etc/php/8.0/apache2/conf.d/20-imagick.ini, /etc/php/8.0/apache2/conf.d/20-ldap.ini, /etc/php/8.0/apache2/conf.d/20-mbstring.ini, /etc/php/8.0/apache2/conf.d/20-mcrypt.ini, /etc/php/8.0/apache2/conf.d/20-mysql.ini, /etc/php/8.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.0/apache2/conf.d/20-phar.ini, /etc/php/8.0/apache2/conf.d/20-posix.ini, /etc/php/8.0/apache2/conf.d/20-readline.ini, /etc/php/8.0/apache2/conf.d/20-shmop.ini, /etc/php/8.0/apache2/conf.d/20-simplexml.ini, /etc/php/8.0/apache2/conf.d/20-sockets.ini, /etc/php/8.0/apache2/conf.d/20-sysmsgm.ini, /etc/php/8.0/apache2/conf.d/20-sysvsem.ini, /etc/php/8.0/apache2/conf.d/20-sysvshm.ini, /etc/php/8.0/apache2/conf.d/20-tokenizer.ini, /etc/php/8.0/apache2/conf.d/20-xmlreader.ini, /etc/php/8.0/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.0/apache2/conf.d/20-xsl.ini, /etc/php/8.0/apache2/conf.d/20-zip.ini

Obrázek 7: Ověření zobrazení serveru přes „semestralka.lan/phpinfo.php“ na prohlížeči notebooku



```
Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ngtua>ping semestralka.lan

Pinging semestralka.lan [10.0.1.34] with 32 bytes of data:
Reply from 10.0.1.34: bytes=32 time<1ms TTL=64
Reply from 10.0.1.34: bytes=32 time<1ms TTL=64
Reply from 10.0.1.34: bytes=32 time<1ms TTL=64
Reply from 10.0.1.34: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.1.34:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\ngtua>
```

Obrázek 8: Ověření z notebooku pomocí příkazu „ping semestralka.lan“