Dokumentacija projektnega dela

Skupina “FILMI”

Vodja: Anja Hauptman

Člani:

Dominik Šbüll

Simona Siljanovska

Urška Nemet

Kazalo

[Kazalo 1](#_Toc419199375)

[Uvod 2](#_Toc419199376)

[Sklop 1: Namestitev in konfiguracija operacijskega sistema Linux 3](#_Toc419199377)

[Sklop 2:Namestitev in konfiguracija strežnika Apache 8](#_Toc419199378)

[Sklop 3: Namestitev in konfiguracija strežnikov MySQL + PHP/Python/Ruby 10](#_Toc419199379)

[Sklop 4: Varnost in zaščita strežnika Linux 15](#_Toc419199380)

[Zaključek 21](#_Toc419199381)

[Viri 22](#_Toc419199382)

Uvod

V okviru projekta Semantični spletni portal smo za predmet Sistemska administracija dobili podano nalogo vzpostavitve LAMP (Linux Apache MySQL Php) strežnika in nekaterih dodatnih funkcionalnosti na le-tem. V prvem sprintu izvajanja projekta smo si zadali nalogo, da opravimo naloge iz prvih štirih uporabniških zgodb, ki se glasijo:

* Namestitev in konfiguracija operacijskega sistema Linux
* Namestitev in konfiguracija strežnika Apache
* Namestitev in konfiguracija strežnikov MySQL + PHP/Python/Ruby
* Varnost in zaščita strežnika Linux

Prvi sklop nalog je opravila Simona, drugi Urška, tretji Anja in četrti Dominik.

Sklop 1: Namestitev in konfiguracija operacijskega sistema Linux

Na svojega strežnika sem namestila operacijski sistem Linux, tako da sem uporabila navidezni strežnik (VirtualBox). Sem uporabila Ubuntu 14.04 iso slika, katero sem vzela iz Ubuntujevo spletno stran, zagnala v VirtualBox-a in namestila OS Linux.

**Vprasanja:**

*1. Kako konfiguriramo diskovni sistem s pomočjo datoteke /etc/fstab? Prikažite tudi, kako deluje diskovno polje RAID na operacijskem sistem Linux (namig: ukaz mdadm) in izpišite stanje sistema v /proc/mdstat.*

*2. Opišite delovanja programa resize2fs.*

*3. Kaj je to ramdisk in kako ga postavimo? Čemu služi imenik /etc/skel?*

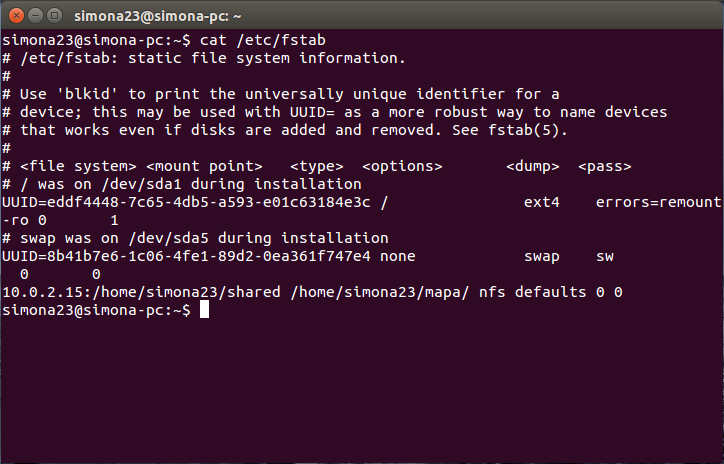
*4. Konfigurirajte zaganjalnik grub (angl. bootloader) tako, da jedro preskoči uničene bajte v pomnilniku (namig: ukaz badram=). Kaj je memtest86?*

*5. Kako vzpostavimo povezavo sshfs?*

**Odgovori:**

**1.** Fstab je konfiguracijska datoteka, ki se nahaja pod /etc direktorij in polna pot do te datoteke je /etc/fstab. Fstab je datoteka, ki vsebuje informacije o vseh diskovnih particij in diskov na našem računalku. V /etc/fstab so shranjene vse informacije o tem, kje naj bi naše diskovne particije in diski priklopljenje (mounted) in kako se to naredi. Če imamo težave z priklapljanje (mounting), to lahko odpravimo z urejevanjem datoteko fstab, kjer fstab je navadna teskstovna datoteka. Lahko to datoteko odpremo in urejamo z katerikoli urejevalnik besedil, ampak če zelimo urejevat to datoteko moremo imet root privilegije oz. uporabimo ukaz su da postanemo root in lahko spreminjamo.

- Če želimo videt kaj ima notri v fstab datoteko lahko izvedemo ukaz:

cat /etc/fstab

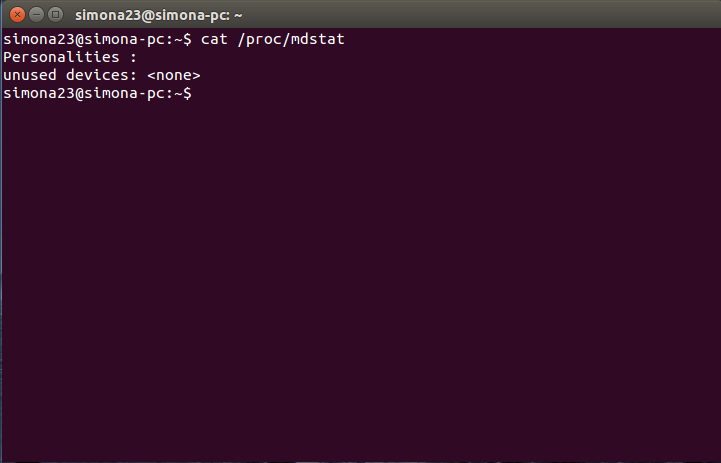
- Kdaj odpremo to datoteko vidimo da vsaka vrstica vsebuje informacije o eni napravi ali particijo. Prvi stolpec vsebuje ime naprave oz. particijo, drugi pa njegova priklopna točka, tretji njegov tip datotečnega sistema, četrti možnosti priklopa itd.

- RAID je [standard](http://sl.wikipedia.org/wiki/Standardizacija" \o "Standardizacija) povezovanja dveh ali več [trdih diskov](http://sl.wikipedia.org/wiki/Trdi_disk" \o "Trdi disk) in upravljanja z njimi, ki je nastal z namenom, da bi lahko več manjših in počasnejših posameznih fizičnih diskov povezali v večjo in hitrejšo in/ali bolj zanesljivo logično enoto. Linux Software RAID naprave se izvajajo skozi gonilnik MD ( oz. skozi več naprav - multiple devices). Trenutno Linux podpira več vrst RAID povezovanja, kot so: LINEAR md naprave, RAID0 (striping), RAID 1 (Mirroring), RAID4, RAID5, RAID6, RAID10, MULTIPATH, FAULTY, and CONTAINER.

- Za upravljanje z RAID napravami, uporabljamo orodje mdadm. Mdadm je orodje, ki ga Linux uporablja za upravljanje in spremljanje programske opreme naprave RAID. To orodje lahko namestimo na Linuxa z ukazom:

sudo apt-get install mdadm

- Stanje sistema izpisemo z ukazom:

cat /proc/mdstat

**2.** Resize2fs program ga uporabljamo za spreminjanje velikosti ext2, ext3 in etx4 datotečne sisteme. Velikost datotečnega sistema ne more biti večji kot velikost same particije. Če velikost ni določena, se vedno privzeto uporabi velikost same particije. Resize2fs program ga tudi uporabljamo za povečanje ali zmanjšanje velikost nekateri nepriklopljen datotečni sistem, ki se nahaja na napravi. Če je datotečni sistem priklopljen, lahko ta program uporabimo za povečanje velikost priklopljenega datotečnega sistema. Resize2fs program ne manipulira z velikosti same particije. Če želimo povečat velikost nekaterega datotečnega sistema, moremo prej vedet, ali lahko prvič povečamo velikost osnovne particije . To lahko naredimo z ukazom fdisk, tako da zbrišemo particijo in datotečnega sistema ponovno oblikujemo z večjo velikostjo.

- ukaz resize2fs:

resize2fs [ -fFpPM ] [ -d debug-flags ] [ -S RAID-stride ] device [ size ]

- f prisili program, da nadaljuje z spreminjanjem velikosti,

-F pomeni da gremo počistit predpomnilnik prej začnemo nasega programa,

-p prikaže odstotek vse opravljene rezise2fs operacije,

-P prikaže najmanjšo velikost nasega datotečnega sistema in pomeni da je konec samega programa,

-M pomeni zmanjšanje datotečnega sistema na minimalno velikostjo,

-d debug-flags so zastavice za razhroščevanje oz. debugging features,

-device pomeni napravo,

-size pomeni velikost naprave in

-S RAID-stride omogoča uporabnika da natačno določi RAID korak, ki se lahko uporablja namesto resize2fs.

**3.** RAM disk ga vospostavimo na nasljedni način:

sudo mkfs -q /dev/ram1 10024 (10024 = 10M)

sudo mkdir -p /ramcache

sudo mount /dev/ram1 /ramcache

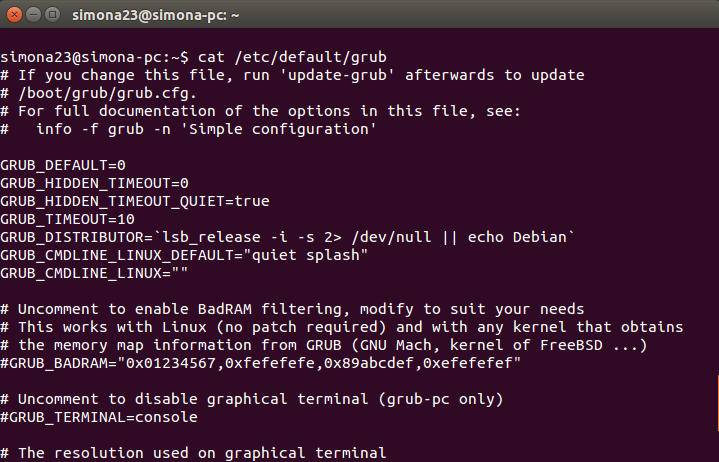
df -H | grep ramcache

- Prvič naredimo particijo velikosti 10M, potem naredimo direktorij /ramcache in na koncu ga tisti direktorij priklopimo na particijo /dev/ram1. Kdaj, končamo na koncu izpisemo podatke za ramdiska s df -H | grep ramcache.

- RAM disk je particija v RAM memoriji, ki se uporablja kot da je trdi disk. RAM diski imajo fiksno velikostjo in delujejo kot redovne oz. osnovne diskovne particije. Čas dostopa je hitrejši pri RAM diska, kot navadni oz. fizični disk. Ampak slabost uporabe RAM diska je da podatki ki so notri tega, izginejo ko se izklopi računalnik. RAM diska je odličen za shranjevanje začasnih podatkov. Tudi je dober za uporabit, kdaj delamo z nešifriranih podatkov iz šifriranih dokumentov.

- /etc/skel direktorij vsebuje datoteke in mape, ki se samodejno prepišejo oz. kopirajo v domači direktorij novega uporabnika, kateri je bil ustvarjen z useradd programa. /etc/skel je direktorij ki omogoča skrbnik sistema ustvariti privzeti domači direktorij za vse nove uporabnike na računalniku ali v omrežju, in s tem prepričati, da vsi uporabniki začnejo z enakimi nastavitvami ali okolje. /etc direktorij in njegove podirektorije vsebujejo več pomembne konfiguracijske datoteke za sistema. Več takih konfiguracijskih datotek so postavljeni v /etc/skel direktorij privzeto, ko je nameščen operacijski sistem. Takšne datoteke so: .bash\_profile, .bashsrc, .bash\_logout, .inputrc, .vimrc, privzetih vrednosti itd.

**4.** Kdaj želimo preskočit uničene bajte v pomnilniku, v grub zaganjalnika podamo direktiva GRUB\_BADRAM. To naredimo tak, da odpremo datoteko /etc/default/grub in dodamo to direktivo, kot je prikazano na sliki. Na sliki vidimo da prvic dodamo v GRUB\_BADRAM zacetni naslov uničenega pomnilnika, potem podamo vse naslovi za preskok v pomnilnika.



- Če želimo testirat kje imamo napake v našem pomnilniku, lahko uporabimo memtest86 program. Memtest86 je odprtokodna programska oprema, izdelana za testiranje RAM našega računalnika oz. naredi test in preveri če imamo napake v našega RAM-a. Memtest86 je program ki testira 32-bitno arhitekturo računalnika. Testiranje ki ga memtest86 naredi je zelo obsežno kjer, lahko najde tudi skrite probleme, ki se zdi da normalno delujejo. Pri testi vedno preveri ali je naš RAM sprejel in pravilno obdržal neke podatke, zapisani notri v RAM memorijo in da ni konflikte med pomnilniških naslovov.

**5.** SSH je varen protokol za komunikacijo med stroji. SSHFS je orodje, ki uporablja SSH in katero omogoča priklop oddaljenega datotečnega sistema na lokalnem stroju (omrežje je pregledno uporabnika). To omogoča da lahko mi upravljamo z datotekami, kot da bi bile v našem direktoriju. Ker SSH šifrira povezave, nihče ne more videti vaše datoteke, kdaj se prenašajo prek omrežja. SSHFS je zgrajen tak da uporablja zaščita oz. FUSE. To pomeni da tudi lastni root uporabnik, lahko vidi svoje datoteke, s prijavo v svojega računa, tak da pri prijavo uporablja ukaz su (potrebuje root privilegije).

- SSHFS povezavo postavimo tak da prvič namestimo programsko opremo, z ukazom:

sudo apt-get install sshfs

apt-get install fuse-sshfs

- Potem če želimo uporabiti normalni uporabniški profil, za priklop datotečnega sistema, z pomočjo SSHFS, moremo najprej dodat uporabnika v skupino FUSE, ki ni superuser in mu dovolimo da uporablja nameščeno programsko opremo. To naredimo na primer, za nekega uporabnika "someuser", z ukazom:

usermod -a -G fuse someuser

- Če želimo da priklopimo domači direktorij nekaterega uporabnika "someuser", na oddaljenem strežniku z imenom something.example.com, to naredimo z naslednjimi ukazami:

mkdir mapa\_someuser\_sshfs **(prvic naredimo mapo)**

sshfs someuser@something.example.com:/home/someuser mapa\_someuser\_sshfs **(potem povežemo ustvarjeno mapo z oddaljenim datotečnim sistemom)**

Sklop 2:Namestitev in konfiguracija strežnika Apache

Na operacijski sistem Linux namestite strežnik Apache. S konfiguracijo tega strežnika so povezane naslednje uporabniške zgodbe:

sudo apt-get install apache2

*1. Kako spremenimo privzeta vrata (angl. port) za dostop do strežnika Apache?*

Tako, da v datoteki */etc/apache2/ports.conf* spremenimo Listen 80v poljubna vrata recimo Listen 4400

*2. Kakšne spremembe je potrebno narediti v konfiguraciji* ***apache2.conf*** *pri ustvarjanju navideznega strežnika? (namig: ukaz* ***VirtualHost****)*

V datoteko sites-enabled zapišemo spletne strani, za katere želimo ustvariti navidezni strežnik. Najprej moramo spremeniti ServerAdmin direktivno na recimo [admin@virtual-host.com](mailto:admin@example.com), da lahko administrator strani prejema maile. Potem moramo dodati 2 direktivi ServerName in ServerAlias. Server name določa osnovno domeno, ki bo ustrezala navideznemu strežniku, primer: Server name: virtual-host.com. ServerAlias pa določa vse poddomene, primer: ServerAlias: [www.virtual-host.com](http://www.virtual-host.com/). Nazadnje spremenimo samo še DocumentRoot direktivo, ki določa lokacijo (recimo na disku) korena dokumentov za to domeno, primer: DocumentRoot: /var/www/virtual-host.com/public\_html.

*3. Kje je shranjeno ime strežnika Apache? (namig:* ***ServerName****) Kako spremenimo ime strežnika Apache na delujočem sistemu? (namig: ukaz* ***reload****)*

Ime strežnika Apache je shranjeno v datoteki: */etc/apache2/sites-available/000-default.conf*

Spremenimo ServerName v datoteki */etc/apache2/sites-available/000-default.conf* in z ukazom restart ponovno zaženemo server, da se uveljavi novi ServerName.

*4. Kako uporabnikom sistema omogočimo dostop do domače javne spletne mape* ***~/public\_html****?*

Omogočimo tako, da v */etc/apache2/sites-available/000-default.conf* k direktivi DocumentRoot lokacijo korena dokumentov za domeno in pripišemo public\_html (/var/www/virtual-host.com/public\_html).

5*. Izpišite konfiguracijo datotek za beleženje napak in dostopa ter opišite njihov format. Izpišite nabor odjemalcev, ki dostopa do strežnika. (namig: uporabite ukaz* ***awk -F\" '{print $(NF-1)}' | sort | uniq****).*

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

Format datotek:

*error.log*

[Sat May 02 17:07:01.032498 2015] [:error] [pid 25436] [client 86.58.94.172:64972] PHP Deprecated: preg\_replace(): The /e modifier is deprecated, use preg\_replace\_callback instead in /var/www/libraries/joomla/filter/input.php on line 652

Najprej je napisan datum in čas, potem je napisana vrsta in stopnja resnosti napake. Sledi IP naslov odjemalca, ki je geneiral oz. sprožil napako. Sledijo sporočila, ki opisujejo napako.

*access.log*

86.58.100.173 - - [02/May/2015:23:35:35 +0200] "GET /phpmyadmin HTTP/1.1" 401 728 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0"

Najprej je IP naslov odjemalca, potem userid od osebe, ki pošilja zahtevek, potem je zapisan datum in čas. Sledi zahtevek (najprej metoda, potem odjemalčev vir in potem odjemalčev protokol). Za zahtevkom je zapisana statusna koda, ki jo pošlje strežnik nazaj odjemalcu in na koncu še velikost objekta, ki je bil vrnjen odjemalcu.

Nabor odjemalcev:

Apache/2.4.7 (Ubuntu) PHP/5.5.9-1ubuntu4.9 (internal dummy connection)

Apache/2.4.7 (Ubuntu) PHP/5.5.9-1ubuntu4.9 mod\_wsgi/3.4 Python/2.7.6 (internal dummy connection)

masscan/1.0 (https://github.com/robertdavidgraham/masscan)

Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/31.0

Mozilla/5.0 (Windows NT 6.1; rv:6.0) Gecko/20110814 Firefox/6.0 Google favicon

Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0

Mozilla/5.0 (Windows NT 6.3; rv:36.0) Gecko/20100101 Firefox/36.0

Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36

Mozilla/5.0 (X11; Linux i686 (x86\_64)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36

Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:37.0) Gecko/20100101 Firefox/37.0

WordPress/4.2.1; [http://164.8.252.141](http://164.8.252.141/)

Torej odjemalci so: masscan,Mozilla in WordPress.

Sklop 3: Namestitev in konfiguracija strežnikov MySQL + PHP/Python/Ruby

S pomočjo podatkovne baze MySQL, aplikacijskih strežnikov, ki omogočajo izvajanje spletnih aplikacij pisanih v programskih jezikih za spletno programiranje, kot npr. PHP, Python in Ruby, in sistemov za upravljanje vsebin (angl. Content Management System, krajše CMS), kot npr. MediaWiki, Joomla in Wordpress, vzpostavimo popolno spletno razvojno okolje. Ta sklop je povezan z naslednjimi uporabniškimi zgodbami:

*1. Namestite podatkovno bazo MySQL in iz konfiguracije produkta ugotovite, na katerih številkah vtičnic in vrat (angl. sockets and ports) se strežnik MySQL odziva na zahteve odjemalcev. Izpišite tudi, kakšno je maksimalno število dovoljenih hkratnih povezav s strežnikom MySQL.*

Namestitev podatkovne baze:

sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql

(namestila sem tudi modul za apache strežnik in modul za php5)

Aktiviranje baze:

sudo mysql\_install\_db

sudo /usr/bin/mysql\_secure\_installation

Po izvedbi zadnjega ukaza sledimo navodilom na zaslonu.

Za prikaz številke vtičnice, vrat in maksimalnega števila dovoljenih hkratnih povezav moramo prikazati naslednjo datoteko:

cat /etc/mysql/my.cnf

V kateri je zapisano:

port = 3306

socket = /var/run/mysqld/mysqld.sock

#max\_connections = 100

*2. Namestite aplikacijske module, ki omogočajo programsko okolje za izvajanje aplikacij pisanih v programskih jezikih PHP, Python in Ruby, ter jih omogočite na strežniku Apache. (namig: a2enmod)*

Instalacija **PHP** (omogočen je privzeto):

sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt

Omogočenje **Python**a:

sudo a2enmod mpm\_prefork cgi

(tako damo strežniku Apache dovoljenje, da požene skripte)

Da bo strežnik razpoznal Pythonove datoteke kot izvedljive, je treba spremeniti datoteko:

sudo nano /etc/apache2/sites-enabled/000-default.conf

<VirtualHost \*:80>

<Directory /var/www/test>

Options +ExecCGI

DirectoryIndex index.py

</Directory>

AddHandler cgi-script .py

…

**Ruby**:

sudo apt-get install ruby-full build-essential

S skripto se požene instalacija (jaz sem izbrala instalacijo z Ruby Version Managerjem).

gem install rails

gem install passenger

sudo passenger-install-apache2-module

Sprememba konfiguracijskih datotek:

sudo nano /etc/apache2/apache2.conf

…

LoadModule passenger\_module /usr/local/lib/ruby/gems/2.2.0/gems/passenger-5.0.7/buildout/apache2/mod\_passenger.so

<IfModule mod\_passenger.c>

PassengerRoot /usr/local/lib/ruby/gems/2.2.0/gems/passenger-5.0.7

PassengerDefaultRuby /usr/local/bin/ruby

</IfModule>

sudo nano /etc/apache2/sites-enabled/000-default.conf

…

RailsEnv development

<Directory /var/www/html/helloapp/public>

# This relaxes Apache security settings.

AllowOverride all

Options FollowSymLinks

# Uncomment this if you're on Apache >= 2.4:

Require all granted

</Directory>

…

*3. Namestite naslednje sisteme CMS: MediaWiki, Joomla! in Wordpress.*

**MediaWiki:**

Inštalacijsko datoteko sem snela iz uradne spletne strani, jo razpakirala in naredila povezavo med Apache-jevo glavno mapo ter mapo v kateri je MediaWiki.

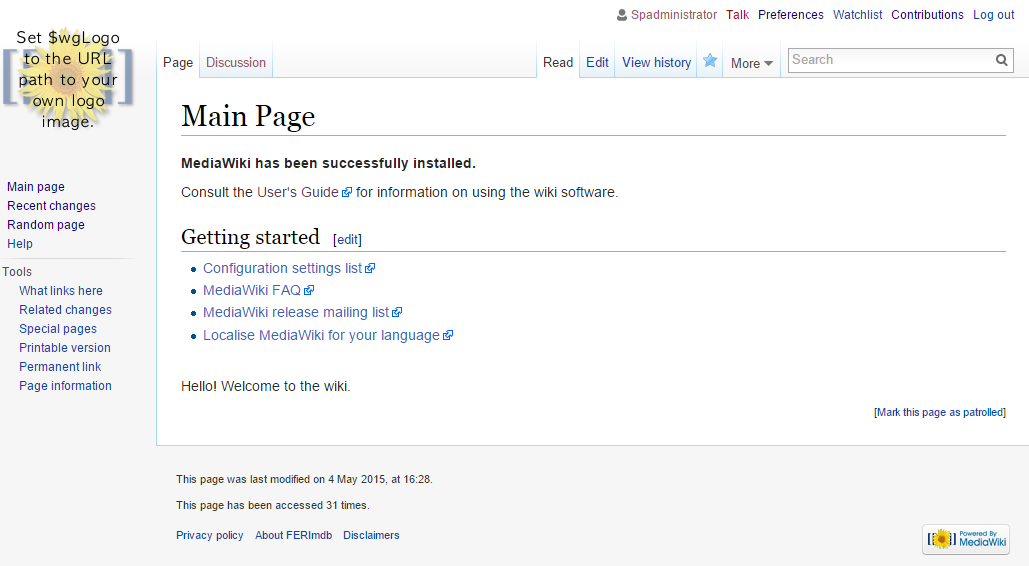
Za dodatno varnost sem vzpostavila tudi podatkovno bazo in jo povezala:

mysql -u root -p

create database my\_wiki;

grant index, create, select, insert, update, delete, alter, lock tables on my\_wiki.\* to 'wikiuser'@'localhost' identified by 'password';

Kot naslednji korak sem sledila navodilom po obisku IP/mediawiki/index.php, kjer sem končala namestitev z prenosom datoteke LocalSettings.php v mapo /etc/mediawiki.



**Wordpress:**

Postopek za namestitev Wordpress-a je skoraj enak kot za Joomla. Po ustvarjeni bazi sem ustvarila konfiguracijsko datoteko wp-config.php, v katero sem shranila podatke o bazi:

/\*\* The name of the database for WordPress \*/

define('DB\_NAME', 'wordpress');

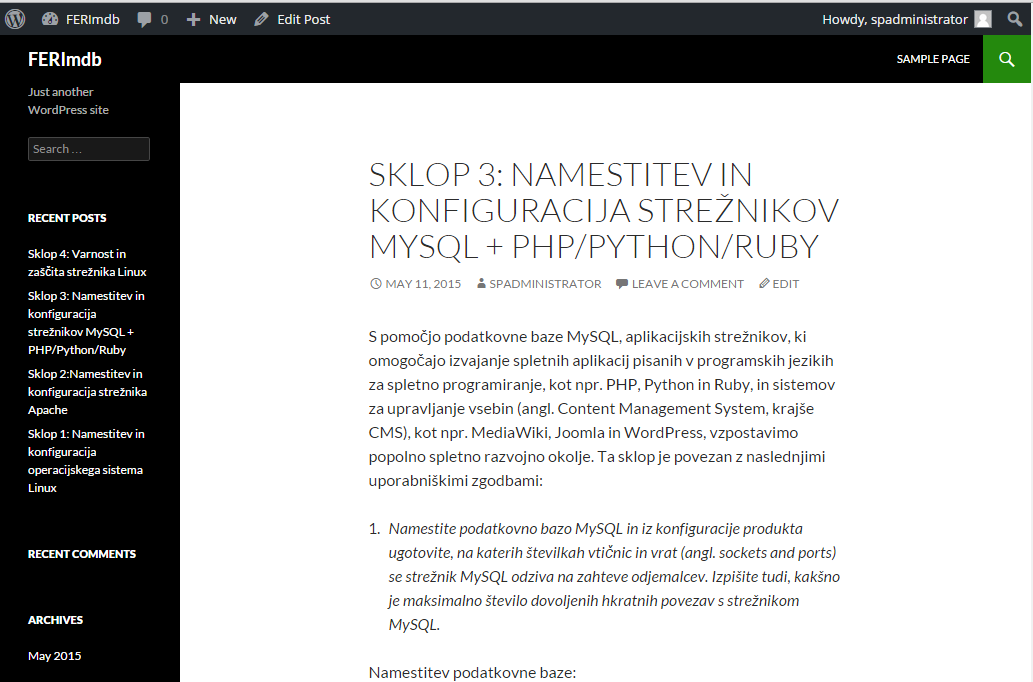
/\*\* MySQL database username \*/

define('DB\_USER', 'wordpressuser');

/\*\* MySQL database password \*/

define('DB\_PASSWORD', 'password');

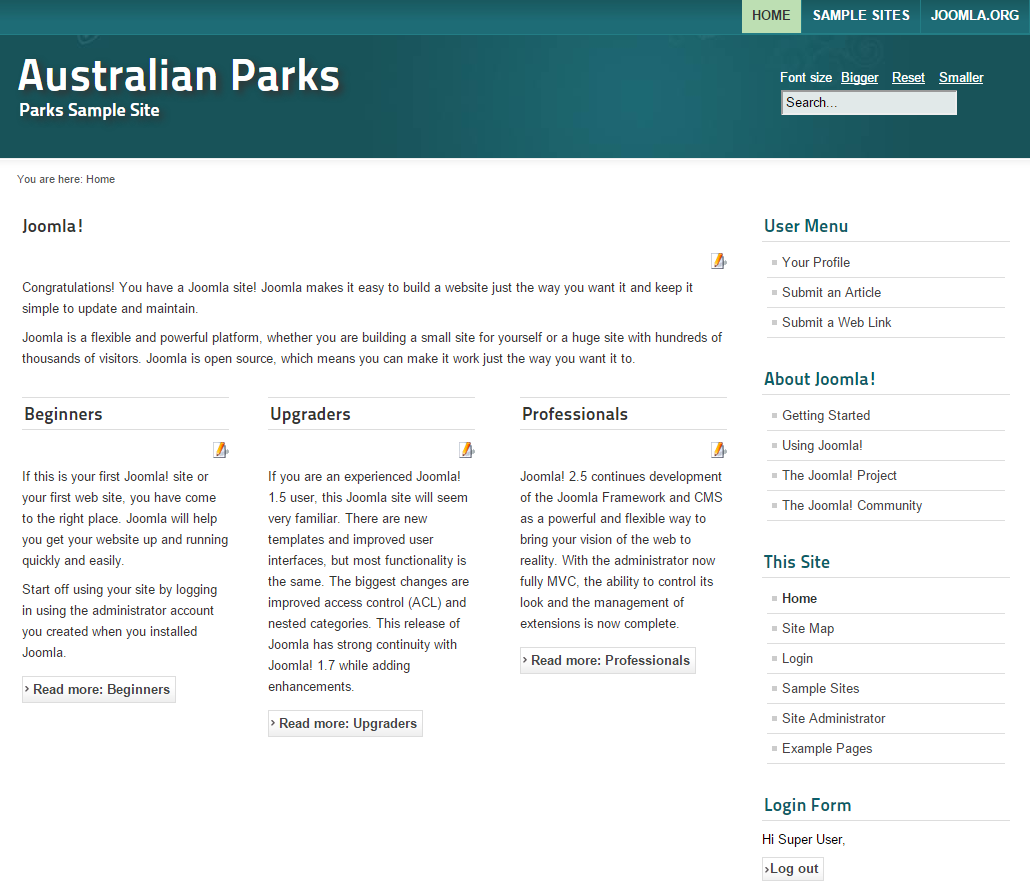
Zadnji korak je bil še obisk IP/wordpress/wp-admin/install.php, ker sem sledila navodilom.



**Joomla:**

Podobno kot pri MediaWiki, sem snela datoteke in jih razpakirala, a tokrat v privzeti apache direktorij (/var/www). Nato sem ustvarila datoteko configuration.php ter tudi novo tabelo v mysql bazi (podobno kot pri MediaWiki, le da sem tukaj ustvarila še uporabnika in mu dodelila geslo).

Z delom sem nadaljevala z obiskom IP/joomla, kjer sem sledila kratkim navodilom v brskalniku.



*4. V vsak nameščen sistem CMS vstavite svojo predlogo za prikaz (angl. render) ekrana in izpišite lokacijo teme na disku. (namig: theme)*

**Wordpress**:

Obiščemo IP/wordpress/wp-admin, kjer izberemo zavihek Appearance -> Themes in nato kliknemo še Add Theme, kjer izberemo temo.

Teme se namestijo v mapo /var/www/wordpress/wp-content/themes.

**Joomla:**

Obiščemo IP/joomla, kjer se prijavimo kot administrator. Izberemo zavihek Extensions -> Template manager, ter zavihek Templates.

Če izberemo zavihek Extensions -> Extension Manager, lahko namestimo svojo temo.

Nahajajo se v mapi /var/www/joomla/templates.

**Mediawiki:**

Obiščemo IP/mediawiki, kjer se ponovno prijavimo kot administrator. Izberemo zavihek Preferences, nato še Appearance, kjer izberemo Skin.

Nahajajo se v mapi /var/www/mediawiki/skins.

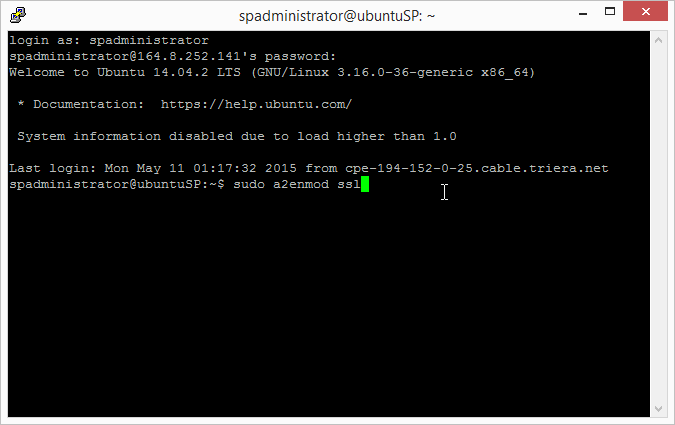
*5. izberite enega izmed nameščenih sistemov CMS in v njem prikažite vsebino z dokumentacijo vašega projekta.*

Sklop 4: Varnost in zaščita strežnika Linux

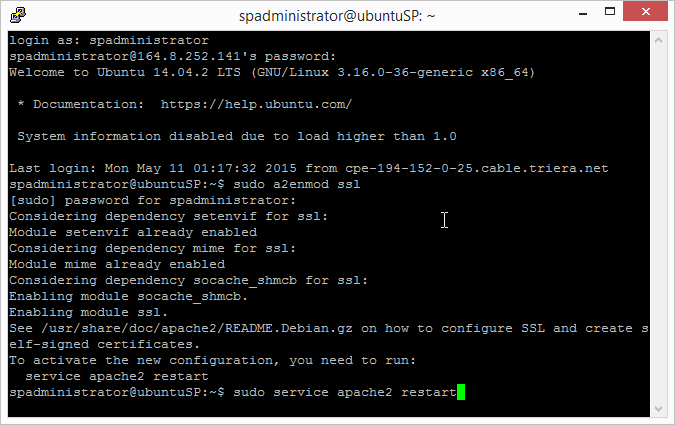
Operacijski sistem Linux skupaj z nameščenimi programskimi produkti omogoča več vrst zaščite pred vsiljivci s spleta. Nekaj načinov zaščite prinašajo naslednje uporabniške zgodbe:

1*. Vzpostavite varno povezavo SSL s strežnikom Apache. Opišite, kako generiramo strežniški certifikat in kako ga lahko uporabimo na strežniku Apache. (namig:* ***SSLCertificateFile****). V kateri mapi na sistemu Linux je shranjen certifikat?*

Pri vzpostavitvi varne povezave SSL s strežnikom Apache je kot prvo potreben certifikat ter z njim povezani ključ katerega je potrebno ustvariti, v našem primeru bo samopodpisani, za ustvaritev le tega bomo kot prvo vključili SSL modul, ki pride v paketu Apache strežnika z ukazom:



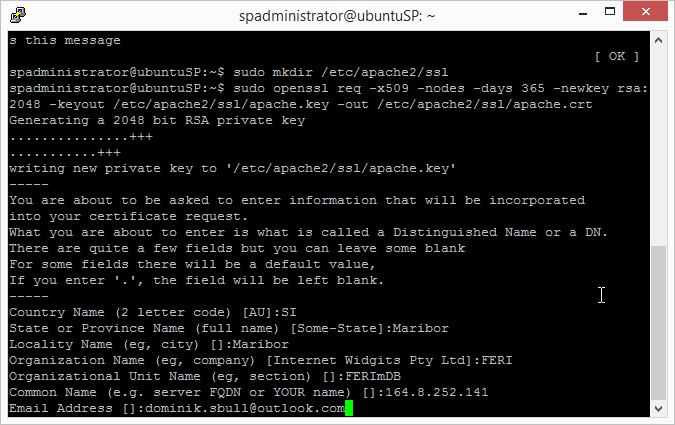
Po le tem, bi se mogel uspešno SSL modul vključiti. Sedaj je za uporabo le tega, potreben še ponovni zagon samega strežnika Apache, kar storimo z nasljednjim ukazom.



Sedaj, k samemu ustvarjanju samo-podpisanega certifikata, kot prvo si ustvarimo novi direktorij, v katerem bomo imeli shranjen ključ ter certifikat. To storimo z ukazom »sudo mkdir /etc/apache2/ssl«

Sedaj ko imamo direktorij kamor lahko shranimo naš ključ ter certifikat, ga ustvarimo in sicer z ukazom prikazanim v spodnjem »screenshot-u« ter vnesimo še nekaj podatkov, za naš certifikat.

Pozor: »TUKAJ JE IZJEMNO POMEMBNO DA VNESEMO POD COMMON NAME IP OZIROMA DOMENO NAŠEGA STREŽNIKA«



Definirajmo sedaj še kaj dejansko podani elementi ukaza pomenijo:

**openssl** – to je osnovni ukaz s katerim povemo da bomo delali z SSL-om, uporabljamo ga, za upravljanje s certifikati, podpisovanje certifikatov, itd.

**req** – s tem podukazom standarda X.509, povemo da želimo podpisati certifikat.

**-x509** – s tem ukazom povemo da želimo ustvariti samo-podpisani certifikat v tem primeru bomo uporabljali standard X.509 certifikata, ta standard uporablja javni ključ.

**-nodes** – ta ukaz pove OpenSSL-u da ključ ne želimo zaščititi z geslom v nasprotnem primeru, bi bilo potrebno po konfiguraciji, po vsakem zagon-u Apache strežnika vnesti izbrano geslo.

**-days 365** – tukaj definiramo da bo naš certifikat veljaven eno leto.

**-newkey rsa:2048** – s tem elementom ustvarimo novi zasebni ključ RSA ki bo dolg 2048 bitov.

**-keyout:** - s tem parametrom definiramo ime zasebnega ključa katerega bomo ustvarili.

**-out:** - s tem parametrom pa podamo ime certifikata pod katerim bo shranjen.

Sedaj ko smo ustvarili certifikat ter ključ, je potrebno konfigurirati Apache strežnik za uporabo SSL – a. Uporabili bomo konfiguracijsko datoteko z imenom »default-ssl.conf«, ki vsebuje že kot že samo ime pove privzete nastavitve za SSL konfiguracijo, mi jo bomo tukaj le nekoliko priredili, saj nam po večini nastavitve odgovarjajo.

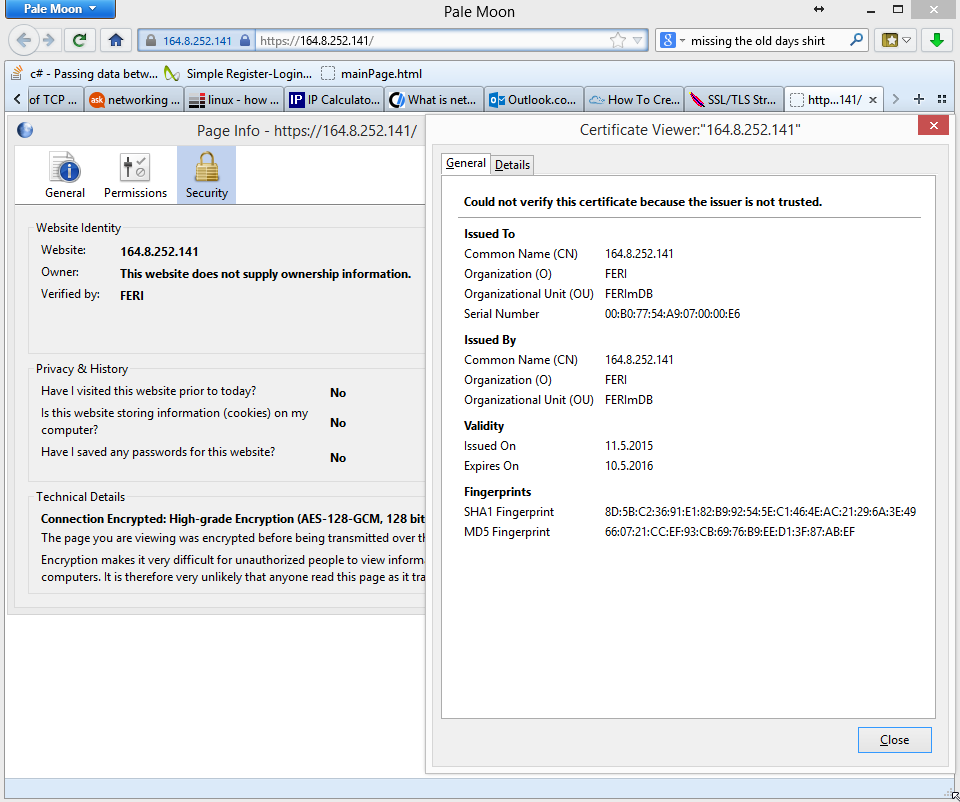
Kot prvo pa moremo datoteko odpreti z urejevalnikom kot administrator, saj imamo opravek z konfiguracijsko datoteko. To storimo z ukazom »sudo nano /etc/apache2/sites-available/default-ssl.conf«



Zgoraj vidimo sliko konfiguracijske datoteke brez komentarjev, stvari z rdečo so bile prirejene najpomembneje je seveda da ne pozabimo na »ServerAlias« ter na poti do certifikata ter ključa.

Sedaj je potrebna le še aktivacija naše konfiguracije to storimo z ukazoma »sudo a2ensite default-ssl.conf« nato spet ponovno zaženemo strežnik Apache z »sudo service apache2 restart«.

Tukaj pa je že končni rezultat, ko bomo sedaj odprli našo spletno stran, bi se nam moglo prikazati opozorilo, kjer sprejmemo oziroma prenesemo certifikat našega strežnika. Kot vidimo na sliki spodaj je sedaj povezava s strežnikom sedaj zaščitena z SSL.



2. *Ob kliku na prijavo preusmerite uporabnika na varno povezavo. (namig:* ***RewriteEngine On****;* ***RewriteCond %{QUERY\_STRING} ^title=Special:UserLogin; RewriteRule ^(.\*)$ https://%{SERVER\_NAME}/ [R]****)*

Da dosežemo cilj naloge je potrebno v datoteki, ki smo jo prej konfigurirali »/etc/apache2/sites-available/default-ssl.conf« pri ustvarjanju našega certifikata dodati še 3 vrstice:

RewriteEngine On

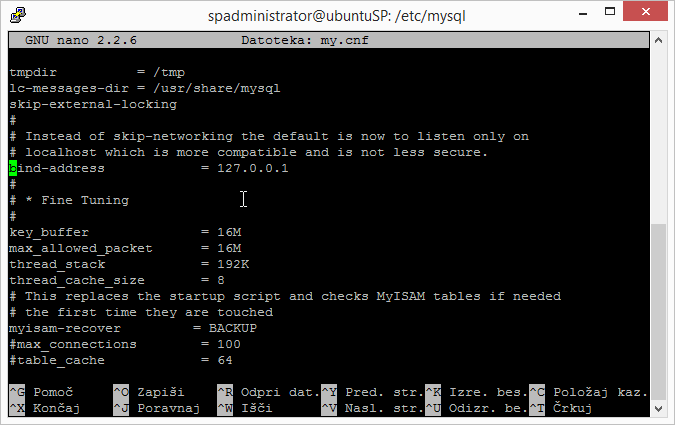
RewriteCond %{HTTPS} off

RewriteRule (.\*) https://%{HTTP\_HOST}%{REQUEST\_URI}

Ob potrebi tudi spremenimo parameter AllowOverride, če je le ta nastavljen na None na All.

3*. Do baze MySQL dovolite samo lokalni dostop. (namig:* ***localhost****) Kje se nahaja konfiguracija MySQL na sistemu Linux?*

MySQL je prej potreboval za limitiranje dostopa od zunaj spremembo v konfiguraciji dodati je bilo potrebno »skip-networking«, sedaj je MySQL strežnik privzeto nastavljen že tako da je dostop možen le lokalno, razen ob spremembi konfiguracije. To lahko vidimo na spodnji sliki, ker »bind-address« vsebuje vrednost 127.0.0.1 kar je localhost.

**

Konfiguracija za »MySQL« se na sistemu Linux privzeto nahaja na lokaciji »/etc/mysql/my.cnf«

4. *Na strežniku Apache ustvarite imenik* ***private/****, v katerem dovolite dostop le izbranim avtentificiranim uporabnikom. Kaj je pri tem potrebno spremeniti v sami konfiguraciji strežnika Apache?*

Najprej ustvarimo nov imenik »sudo mkdir /var/www/private«, potem uporabimo za kriptiranje gesla ukaz »htpasswd –c /var/www/private/.htpasswd up1«, nato ustvarimo datoteko .htaccess v katero zapišemo sledeče: »AuthUserFile /var/www/private/.htpasswd«, »AuthName 'Password required'«, »AuthType Basic«, »require user up1 up1«.

5. *S pomočjo* ***iptables*** *postavite požarni zid na sistemu Linux, ki vsem uporabnikom dovoljuje promet TCP na vratih 80 in 443, beleži vsak dostop do vrat 22 (namig:* ***LOG****), in dovoljuje dostop do vrat 22 iz podomrežja* ***164.8.0.0/16*** *ter treh izbranih računalnikov z 32-bitno masko podomrežja.*



Na zgornji sliki je prikazan bash script, ki ima nastavljena vsa pomembna osnovna pravila ter pravila iz naloge in sicer:

iptables -A INPUT -p tcp --dport 80 -j MYACCEPT

iptables -A OUTPUT -p tcp --dport 80 -j MYACCEPT

iptables -A INPUT -p tcp --dport 443 -j MYACCEPT

iptables -A OUTPUT -p tcp --dport 443 -j MYACCEPT

iptables -A INPUT -p tcp --dport 22 -s 164.8.0.0/16 -j MYACCEPT

iptables -A INPUT -p tcp --dport 22 -s 192.168.0.1/32 -j MYACCEPT

iptables -A INPUT -p tcp --dport 22 -s 192.168.0.2/32 -j MYACCEPT

iptables -A INPUT -p tcp --dport 22 -s 192.168.0.3/32 -j MYACCEPT

iptables -A INPUT -p tcp --dport 22 -j LOG –log-prefix 'acces to port 22'

Zaključek

Z opravljenim delom smo zadovoljni in bomo z njim nadaljevali v drugem sprintu, kjer se bomo lotili naslednjih sklopov nalog, ki se glasijo:

* Cron backup map, podatkovnih baz in nastavitev
* Samodejni ponovni zagon ob izpadu, izdelava prikrojenega namestitvenega paketa
* Napredna sistemska administracija Linux
* Domain Name System (DNS)

In jih bomo zajeli v drugem delu projektne dokumentacije.

Viri

<http://www.discretelogix.com/blog/ruby-rails/installing-apache-ruby-mysql-passenger-ubuntu-14-04>

<https://www.digitalocean.com/community/tutorials/how-to-install-mediawiki-on-ubuntu-12-04>

<https://www.digitalocean.com/community/tutorials/how-to-install-joomla-on-a-virtual-server-running-ubuntu-12-04>

<https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-on-ubuntu-12-04>

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu>

<https://www.digitalocean.com/community/tutorials/how-to-set-up-an-apache-mysql-and-python-lamp-server-without-frameworks-on-ubuntu-14-04>

<http://wiki.centos.org/HowTos/Network/IPTables>

<https://www.digicert.com/ssl-certificate-installation-ubuntu-server-with-apache2.htm>

<https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-14-04-lts>

<https://www.debian-administration.org/article/412/Hosting_multiple_websites_with_Apache2>

<http://www.linfo.org/etc_skel.html>

<http://www.vanemery.com/Linux/Ramdisk/ramdisk.html>

<http://www.cyberciti.biz/faq/howto-create-linux-ram-disk-filesystem/>

<http://linux.die.net/man/8/mdadm>

<http://en.wikipedia.org/wiki/RAID>

<http://linux.die.net/man/8/resize2fs>

<http://en.wikipedia.org/wiki/Memtest86>

<https://www.linode.com/docs/networking/ssh/using-sshfs-on-linux-and-macos-x>

<http://www.cyberciti.biz/faq/how-to-mount-remote-directory-filesystems-with-sshfs-on-linux/>

<http://pclosmag.com/html/issues/200709/page07.html>