

# DOKUMENTACIJA PROJEKTNEGA DELA

Skupina "FERImdb"

Projekt semantični spletni portal



Vodja: Anja Hauptman

Člani:

Dominik Šbüll

Simona Siljanovska

Urška Nemet

## Kazalo

Kazalo .....	1
Uvod .....	2
Sklop 1: Namestitev in konfiguracija operacijskega sistema Linux .....	3
Sklop 2: Namestitev in konfiguracija strežnika Apache .....	8
Sklop 3: Namestitev in konfiguracija strežnikov MySQL + PHP/Python/Ruby .....	10
Sklop 4: Varnost in zaščita strežnika Linux .....	15
Sklop 5: Cron backup map, podatkovnih baz, in nastavitev sistema .....	21
Sklop 6: Samodejni ponovni zagon ob izpadu, izdelava prirojenega namestitvenega paketa .....	26
Sklop 9: Napredna sistemska administracija Linux .....	30
Sklop 10: DNS .....	43
Zaključek .....	49
Viri .....	50

## 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.

Za drugi sprint smo si izbrali naloge:

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

Sklop 5 je opravila Anja, sklop 6 Urška, sklop 9 Dominik in sklop 10 Simona.

## 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:

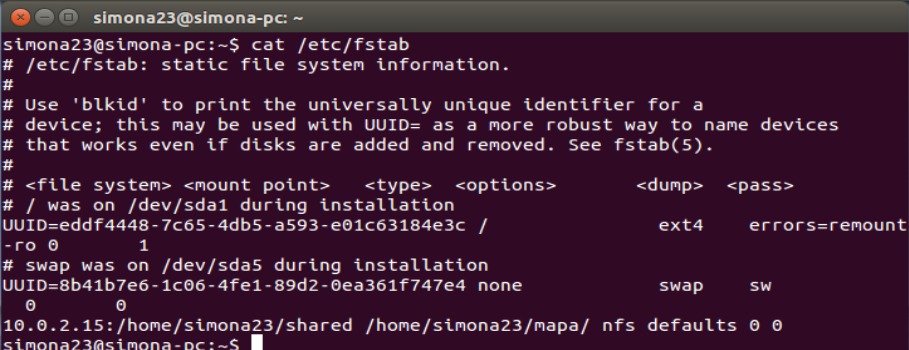
1. Kako konfiguriramo diskovni sistem s pomočjo datoteke `/etc/fstab`? Prikažite tudi, kako deluje diskovno polje RAID na operacijskem sistem Linux (namig: ukaz `mdadm`) in izpišite stanje sistema v `/proc/mdstat`.
2. Opišite delovanja programa `resize2fs`.
3. Kaj je to `ramdisk` in kako ga postavimo? 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 računalniku. 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 priklopljanje (mounting), to lahko odpravimo z urejevanjem datoteko `fstab`, kjer `fstab` je navadna tekstovna 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
```



```
simona23@simona-pc: ~  
simona23@simona-pc:~$ cat /etc/fstab  
# /etc/fstab: static file system information.  
#  
# Use 'blkid' to print the universally unique identifier for a  
# device; this may be used with UUID= as a more robust way to name devices  
# that works even if disks are added and removed. See fstab(5).  
#  
# <file system> <mount point> <type> <options> <dump> <pass>  
# / was on /dev/sda1 during installation  
UUID=eddf4448-7c65-4db5-a593-e01c63184e3c / ext4 errors=remount  
-ro 0 1  
# swap was on /dev/sda5 during installation  
UUID=8b41b7e6-1c06-4fe1-89d2-0ea361f747e4 none swap sw  
0 0  
10.0.2.15:/home/simona23/shared /home/simona23/mapa/ nfs defaults 0 0  
simona23@simona-pc:~$
```

- 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.

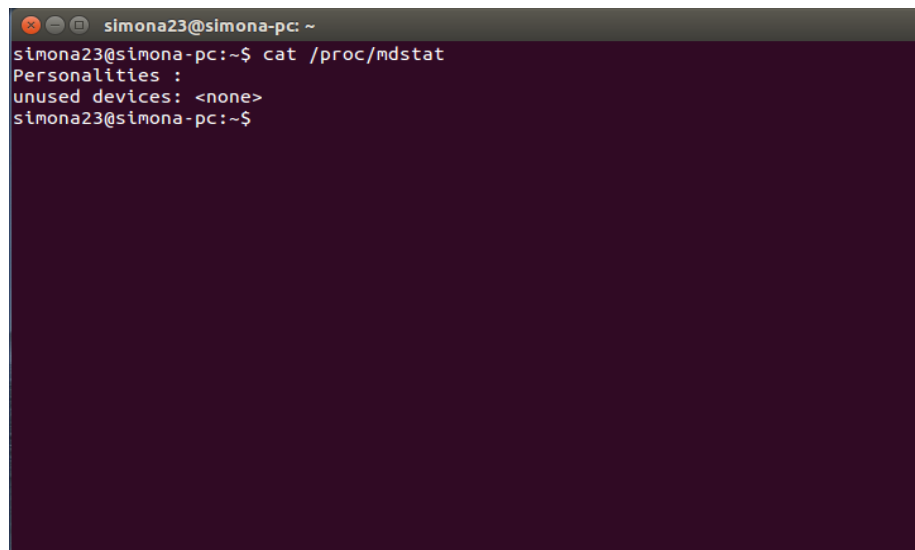
- RAID je standard povezovanja dveh ali več trdih diskov in upravljanja z njimi, ki je nastal z namenom, da bi lahko več manjših in počasnejših posameznih fizičnih diskov povezali v večjo in hitrejšo in/ali bolj zanesljivo logično enoto. Linux Software RAID naprave se izvajajo skozi gonilnik MD ( oz. 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 Linux z ukazom:

```
sudo apt-get install mdadm
```

- Stanje sistema izpisemo z ukazom:

```
cat /proc/mdstat
```



```
simona23@simona-pc: ~  
simona23@simona-pc:~$ cat /proc/mdstat  
Personalities :  
unused devices: <none>  
simona23@simona-pc:~$
```

**2.** Resize2fs program ga uporabljamo za spreminjanje velikosti ext2, ext3 in ext4 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 povečati velikost nekaterega datotečnega sistema, moremo prej vedeti, ali lahko prvič povečamo velikost osnovne particije . To lahko naredimo z ukazom fdisk, tako da zberemo 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 počistiti 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 natančno določi RAID korak, ki se lahko uporablja namesto resize2fs.

### 3. RAM disk ga vopostavimo na nasledni 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 uporabiti, kdaj delamo z nešifriranih podatkov iz šifriranih dokumentov.

- /etc/skel direktorij vsebuje datoteke in mape, ki se samodejno prepisujejo 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 poddirektorije 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 preskočiti 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 prvič dodamo v

GRUB\_BADRAM zacetni naslov uničenega pomnilnika, potem podamo vse naslovi za preskok v pomnilnika.

```
simona23@simona-pc: ~  
simona23@simona-pc:~$ cat /etc/default/grub  
# If you change this file, run 'update-grub' afterwards to update  
# /boot/grub/grub.cfg.  
# For full documentation of the options in this file, see:  
#   info -f grub -n 'Simple configuration'  
  
GRUB_DEFAULT=0  
GRUB_HIDDEN_TIMEOUT=0  
GRUB_HIDDEN_TIMEOUT_QUIET=true  
GRUB_TIMEOUT=10  
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`  
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"  
GRUB_CMDLINE_LINUX=""  
  
# Uncomment to enable BadRAM filtering, modify to suit your needs  
# This works with Linux (no patch required) and with any kernel that obtains  
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)  
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"  
  
# Uncomment to disable graphical terminal (grub-pc only)  
#GRUB_TERMINAL=console  
  
# The resolution used on graphical terminal
```

- Če želimo testirati 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 memoriji 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 prijavi 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 80` v 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@virtual-host.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`. 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 general 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
```

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 **Pythona**:

```
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.

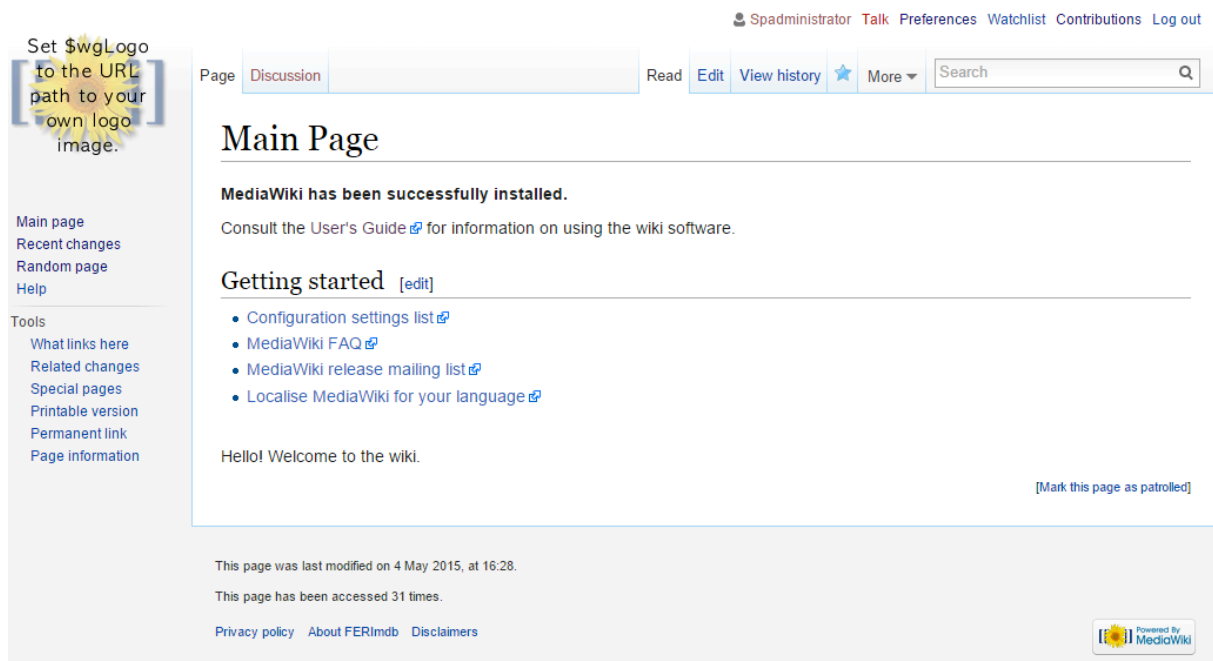
Za dodatno varnost sem vzpostavila tudi podatkovno bazo in jo povezala:

```
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](http://IP/mediawiki/index.php), kjer sem končala namestitvev z prenosom datoteke LocalSettings.php v mapo /etc/mediawiki.



## Wordpress:

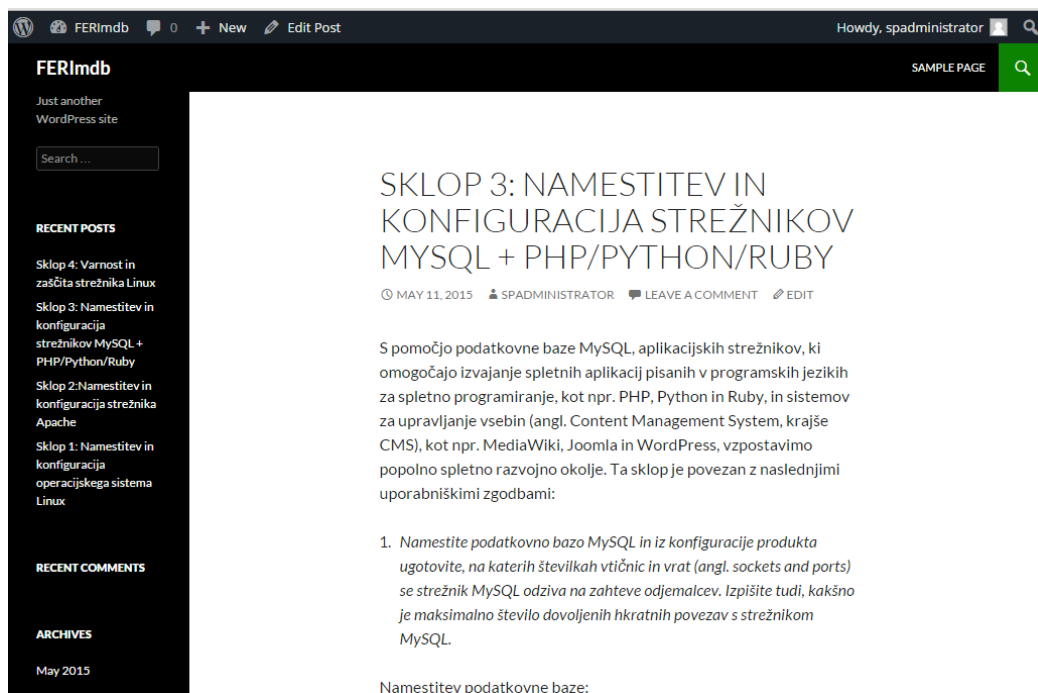
Postopek za namestitvev Wordpress-a je skoraj enak kot za Joomla. Po ustvarjeni bazi sem ustvarila konfiguracijsko datoteko wp-config.php, v katero sem shranila podatke o bazi:

```
/** 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');
```

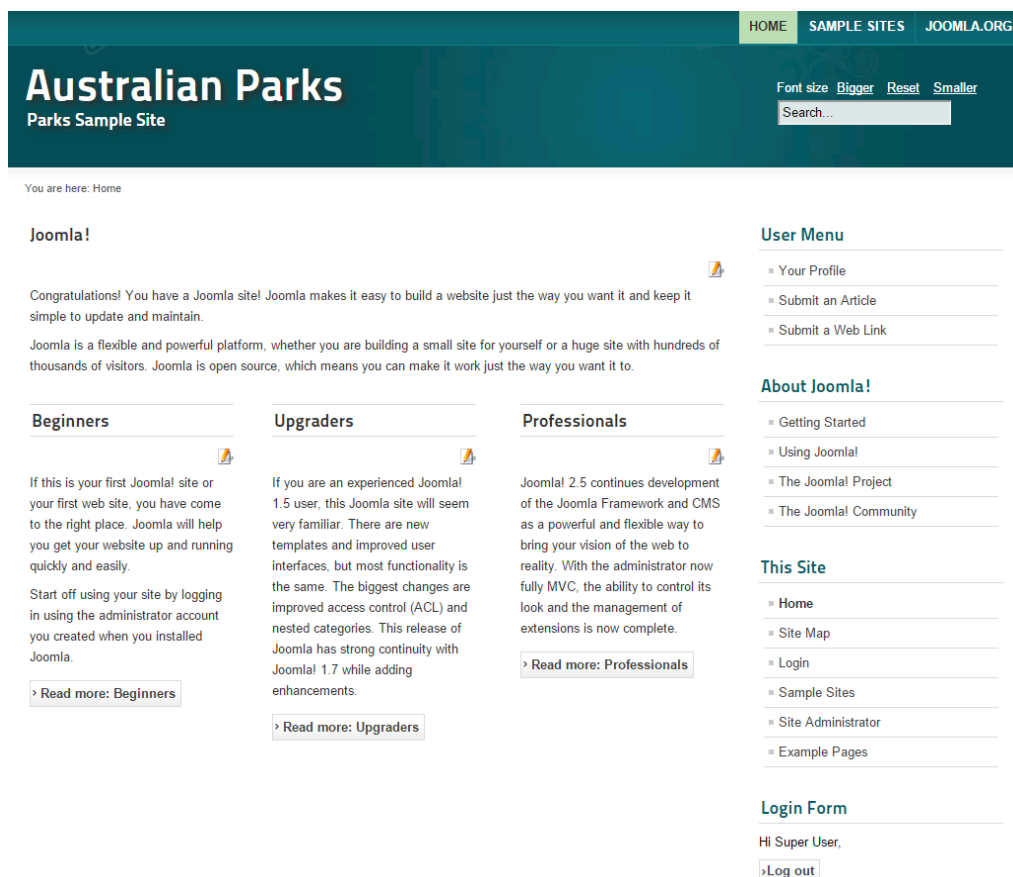
Zadnji korak je bil še obisk [IP/wordpress/wp-admin/install.php](http://IP/wordpress/wp-admin/install.php), ker sem sledila navodilom.



## 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](http://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.*

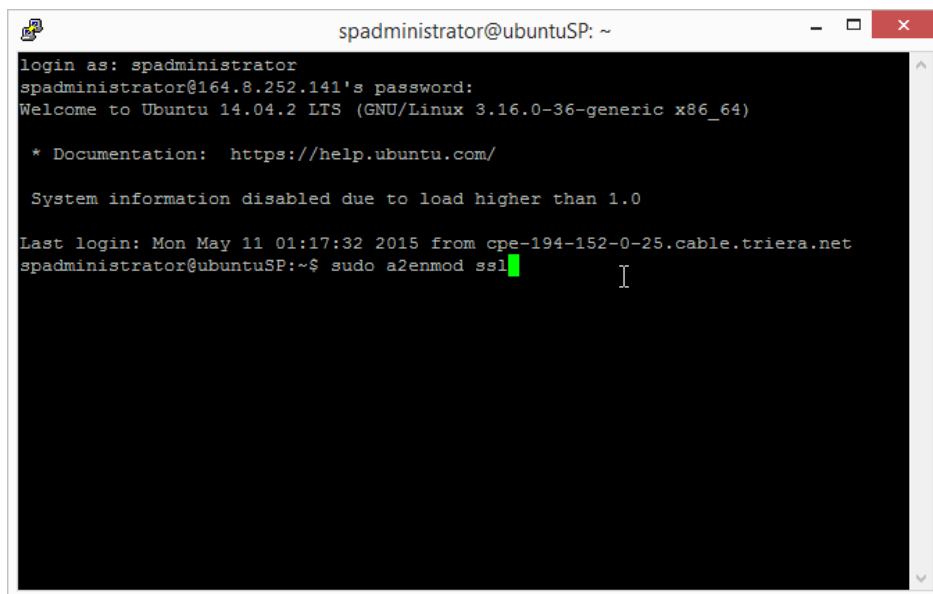
Izbrala sem Wordpress in vanj naložila projektno dokumentacijo.

## 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:

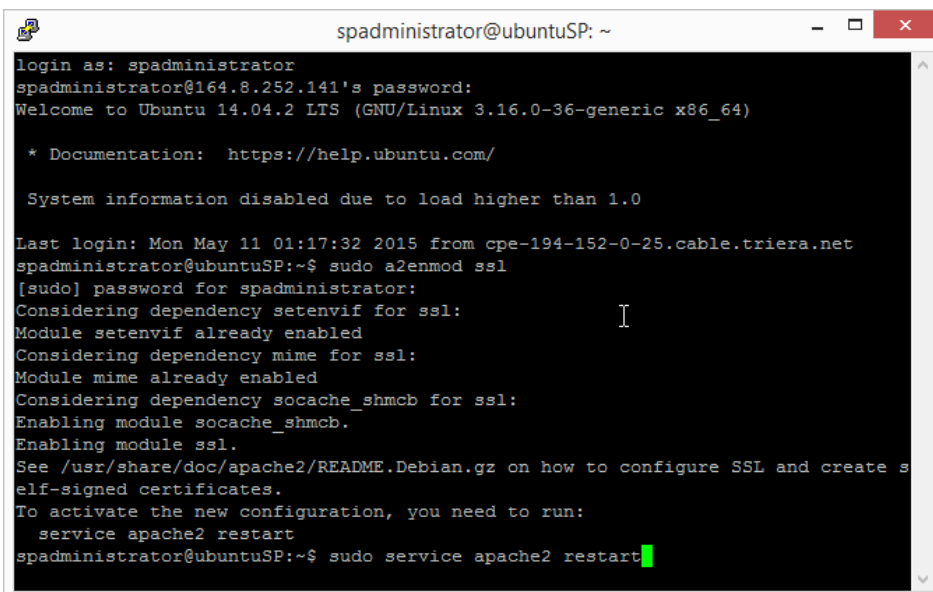
1. Vzpostavite varno povezavo SSL s strežnikom Apache. Opišite, kako generiramo strežniški certifikat in kako ga lahko uporabimo na strežniku Apache. (namig: **SSLCertificateFile**). V kateri mapi na sistemu Linux je shranjen certifikat?

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:



```
spadministrator@ubuntuSP: ~  
login as: spadministrator  
spadministrator@164.8.252.141's password:  
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-36-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
System information disabled due to load higher than 1.0  
  
Last login: Mon May 11 01:17:32 2015 from cpe-194-152-0-25.cable.triera.net  
spadministrator@ubuntuSP:~$ sudo a2enmod ssl
```

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 naslednjim ukazom.



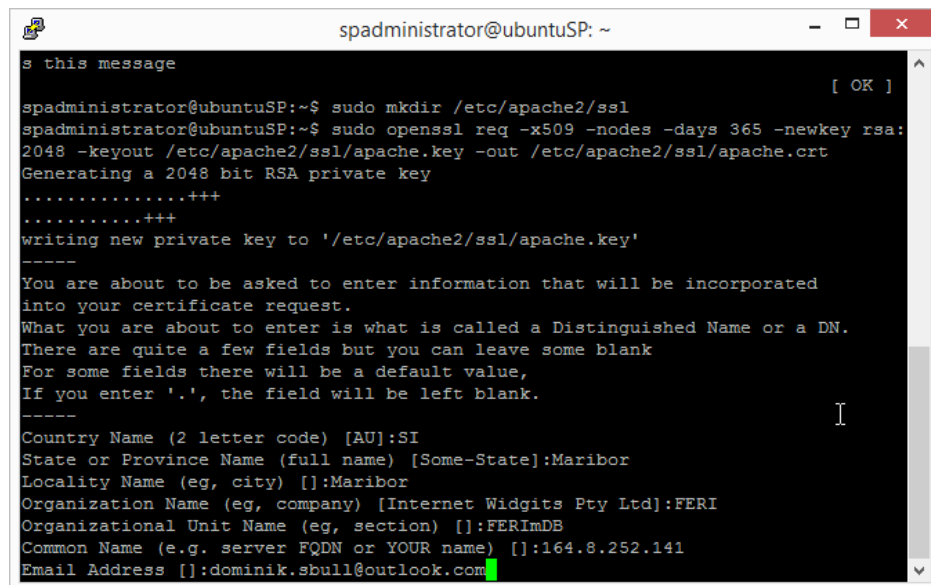
```
spadministrator@ubuntuSP: ~  
login as: spadministrator  
spadministrator@164.8.252.141's password:  
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-36-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
System information disabled due to load higher than 1.0  
  
Last login: Mon May 11 01:17:32 2015 from cpe-194-152-0-25.cable.triera.net  
spadministrator@ubuntuSP:~$ sudo a2enmod ssl  
[sudo] password for spadministrator:  
Considering dependency setenvif for ssl:  
Module setenvif already enabled  
Considering dependency mime for ssl:  
Module mime already enabled  
Considering dependency socache_shmcb for ssl:  
Enabling module socache_shmcb.  
Enabling module ssl.  
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.  
To activate the new configuration, you need to run:  
    service apache2 restart  
spadministrator@ubuntuSP:~$ sudo service apache2 restart
```



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«**



```
spadministrator@ubuntuSP: ~  
$ this message  
[ OK ]  
spadministrator@ubuntuSP:~$ sudo mkdir /etc/apache2/ssl  
spadministrator@ubuntuSP:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:  
2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt  
Generating a 2048 bit RSA private key  
.....++  
.....++  
writing new private key to '/etc/apache2/ssl/apache.key'  
-----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Country Name (2 letter code) [AU]:SI  
State or Province Name (full name) [Some-State]:Maribor  
Locality Name (eg, city) []:Maribor  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FERI  
Organizational Unit Name (eg, section) []:FERImDB  
Common Name (e.g. server FQDN or YOUR name) []:164.8.252.141  
Email Address []:dominik.sbull@outlook.com
```

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.

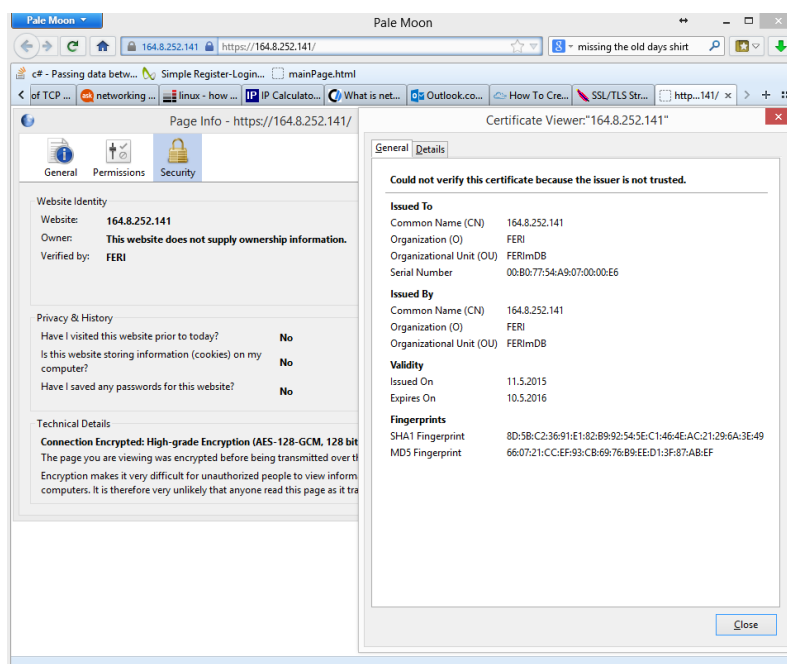
Kot prvo pa moremo datoteko odpreti z urejevalnikom kot administrator, saj imamo opravke z konfiguracijsko datoteko. To storimo z ukazom »sudo nano /etc/apache2/sites-available/default-ssl.conf«

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerAdmin admin@example.com
    ServerName FERImDB
    ServerAlias 164.8.252.141
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache.crt
    SSLCertificateKeyFile /etc/apache2/ssl/apache.key
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>
    BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
</VirtualHost>
</IfModule>
```

Zgoraj vidimo sliko konfiguracijske datoteke brez komentarjev, stvari z rdečo so bile prirejene najpomembnejše 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]**)

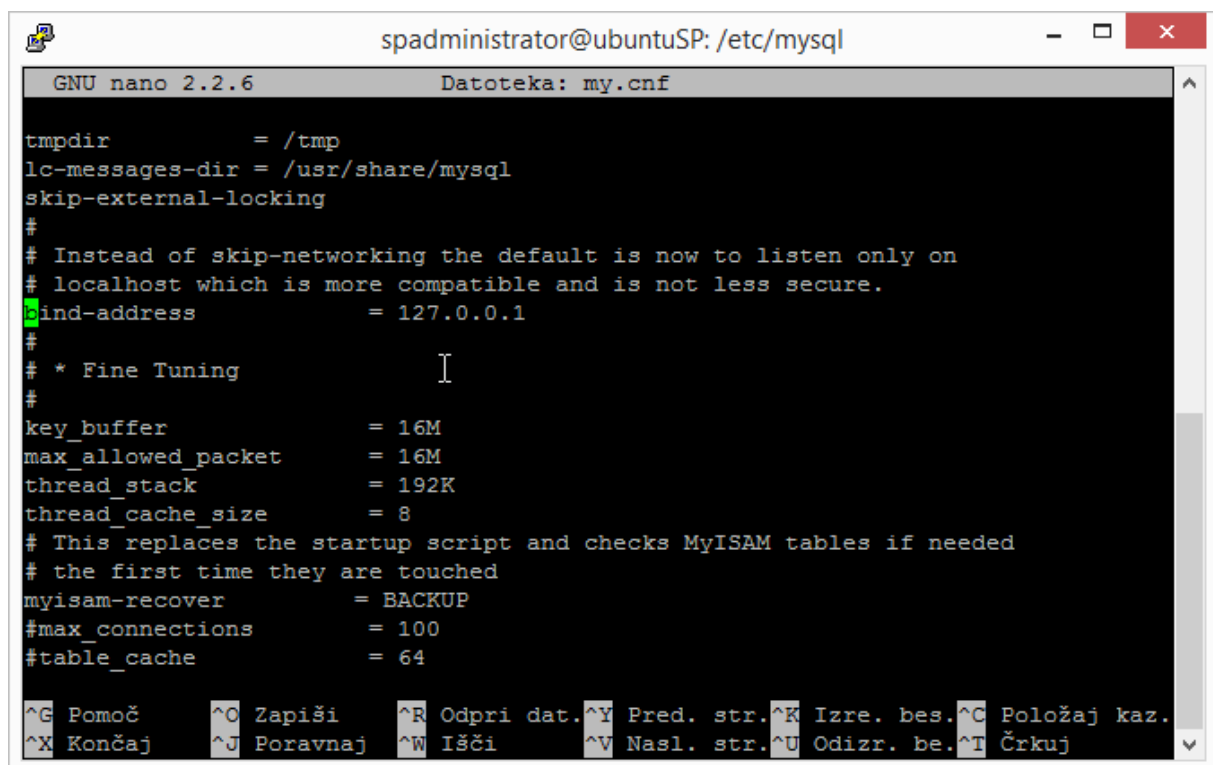
Da dosežemo cilj naloge je potrebno v datoteki, ki smo jo prej konfigurirali »/etc/apache2/sites-available/default-ssl.conf« pri ustvarjanju našega certifikata dodati še 3 vrstice:

```
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.



```
spadministrator@ubuntuSP: /etc/mysql  
GNU nano 2.2.6      Datoteka: my.cnf  
tmpdir              = /tmp  
lc-messages-dir     = /usr/share/mysql  
skip-external-locking  
#  
# Instead of skip-networking the default is now to listen only on  
# localhost which is more compatible and is not less secure.  
bind-address        = 127.0.0.1  
#  
# * Fine Tuning  
#  
key_buffer          = 16M  
max_allowed_packet  = 16M  
thread_stack        = 192K  
thread_cache_size   = 8  
# This replaces the startup script and checks MyISAM tables if needed  
# the first time they are touched  
myisam-recover       = BACKUP  
#max_connections    = 100  
#table_cache        = 64  
^G Pomoč      ^O Zapiši     ^R Odpri dat.^Y Pred. str.^K Izre. bes.^C Položaj kaz.  
^X Končaj     ^J Poravnaj  ^W Išči      ^V Nasl. str.^U Odizr. be.^T Črkuj
```

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.

```
#!/bin/sh
echo "Initializing Firewall..."

#Clear all other rules
iptables -F
iptables -X
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#Creation of my own chains
iptables -N MYDROP
iptables -N MYACCEPT

#Loopback communication
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -i lo -j ACCEPT

#Stateful Inspection
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state INVALID -j MYDROP
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

#Configuring our own chains
iptables -A MYDROP -j LOG --log-prefix "FW-DROP: "
iptables -A MYDROP -j DROP
iptables -A MYACCEPT -j LOG --log-prefix "FW-ACCEPT: "
iptables -A MYACCEPT -j ACCEPT

#SSH
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 "aces to port 22"

#ICMP Ping
iptables -A INPUT -p icmp -j MYACCEPT
iptables -A OUTPUT -p icmp -j MYACCEPT

#DNS
iptables -A INPUT -p udp --dport 53 -j MYACCEPT
iptables -A INPUT -p tcp --dport 53 -j MYACCEPT
iptables -A OUTPUT -p udp --dport 53 -j MYACCEPT
iptables -A OUTPUT -p tcp --dport 53 -j MYACCEPT

#WWW
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

#MAIL
iptables -A INPUT -p tcp --dport 25 -j MYACCEPT
iptables -A OUTPUT -p tcp --dport 25 -j MYACCEPT
iptables -A INPUT -p tcp --dport 110 -j MYACCEPT
iptables -A INPUT -p tcp --dport 143 -j MYACCEPT

#FTP
iptables -A INPUT -p tcp --dport 21 -j MYACCEPT
iptables -A OUTPUT -p tcp --dport 21 -j MYACCEPT

#DHCP
iptables -A INPUT -p udp --dport 67 -j MYACCEPT

echo "Firewall is configured and active!"
iptables -A INPUT -j LOG --log-prefix "FW-LAST-DROP: "
```

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'
```

## Sklop 5: Cron backup map, podatkovnih baz, in nastavitev sistema

Varnostno kopiranje (angl. backup) je osnova za brezprekinitveno delovanje produkcijskih strežnikov. Ta postopek lahko na operacijskem sistemu Linux avtomatiziramo s pomočjo strežnika cron, ki omogoča zagon varnostnega kopiranja ob predpisanem času. Običajno izberemo termine, ko je spletni strežnik najmanj obremenjen (npr. ponoči). Z varnostnim kopiranjem se ukvarjajo naslednje uporabniške zgodbe:

*1. Napišite skripto, ki z ukazom `rdiff-backup` kopira izbrane imenike iz domače mape uporabnika `~/mycodes` na oddaljeni strežnik prek varne povezave `sshfs`. Skripto dodajte v dnevni zagon uporabniškega strežnika cron. (namig: `crontab -e`)*

`Rdiff-backup` je orodje, s katerim lahko ustvarjamo varnostne kopije map in datotek na neki drugi lokaciji (lahko preko spleta). V našem primeru smo kopirali mapo uporabnika »mycodes« na oddaljen strežnik, kjer se shrani kopija tega direktorija in tudi nekaj dodatnih podatkov, ki omogočajo, da lahko po potrebi opravimo »restore« sistema za nekaj časa nazaj. Takšno kopiranje ohrani poddirektorije, povezave, dovoljenja in lastništvo nad datotekami.

Cron je programska oprema, s katero lahko na določene časovne intervale avtomatiziramo izvajanje določenih nalog na strežniku. Tem nalogam (skriptam, ukazom) lahko določimo čas izvanja in kako pogosto želimo da se izvedejo.

Najprej je bilo na strežniku treba namestiti `rdiff-backup` in `sshfs` z ukazom:

```
apt-get install rdiff-backup sshfs
```

Nato pa sem ustvarila tudi uporabnika `mycodes` in v njegov domači direktorij dodala mapo `.ssh` za njegov zaseben in javni `ssh` ključ.

```
sudo adduser mycodes
su - mycodes
mkdir /home/mycodes/.ssh
```

Nato sem ustvarila še prej omenjena ključa z ukazom:

```
ssh-keygen -t rsa
```

In sledila navodilom na zaslonu, da sem ustvarila ključa `id_rsa` in `id_rsa.pub`.

Za backup strežnik sem uporabila staro namestitev Linux Ubuntu na svojem računalniku v VirtualBox okolju in naslednje ukaze izvedla tam.

Tam je bilo treba namestiti `openssh-server` in ustvariti novega uporabnika, da lahko njegov domač direktorij uporabljamo kot backup mapo.

```
sudo apt-get install openssh-server
sudo adduser rdiffbackup
sudo usermod -a -G fuse rdiffbackup
```

Tudi na tem strežniku je bilo potrebno ustvariti ssh ključa in javnega dodati med `authorized_keys` oddaljenega strežnika, ki ga želimo varnostno kopirati. S tem se lahko povežemo na ta strežnik in dostopamo do njegovih datotek brez vnosa gesel.

```
su - rdiffbackup
ssh-keygen -t rsa
scp ~/.ssh/id_rsa.pub mycodes@164.8.252.141:/home/mycodes/.ssh/uploaded_key.pub
ssh mycodes@164.8.252.141 "echo `cat ~/.ssh/uploaded_key.pub` >>
~/.ssh/authorized_keys"
```

Nato sem ustvarila še mount-point za priklop oddaljene mape (to je domača mapa uporabnika `mycodes`) in shranjevanje backup datotek.

```
mkdir -p /home/rdiffbackup/mount
mkdir -p /home/rdiffbackup/back-up
chown -R rdiffbackup:rdiffbackup /home/rdiffbackup
```

Nato sem še omogočila mountanje oddaljene mape s preminjanjem datoteke `/etc/fstab`, v katero sem dodala sledeče:

```
sshfs#mycodes@164.8.252.141:/home/mycodes /home/rdiffbackup/mount fuse user,noauto,ro 0 0
```

Backup lahko testiramo z ukazom:

```
rdiff-backup -v5 /home/rdiffbackup/mount /home/rdiffbackup/back-up
```

Ker ne želimo ustvarjati varnostnih kopij manualno, jih lahko avtomatiziramo tako, da se same izvedejo dnevno ob določeni uri. Zato sem napisala skripto `backup.sh`:

```
#!/bin/sh
mount /home/rdiffbackup/mount
rdiff-backup /home/rdiffbackup/mount /home/rdiffbackup/back-up
umount /home/rdiffbackup/mount
```

In ji dodala pravice za izvajanje z ukazom:

```
chmod +x /home/rdiffbackup/backup.sh
```

Nato pa sem z ukazom:

```
crontab -e
```

V crontab dodala vrstico:

```
00 02 * * * /home/rdiffbackup/backup.sh
```

Ki omogoči skripti `backup.sh` izvajanje vsak dan ob 2. uri zjutraj.

Če želimo kopirati backup nazaj na oddaljen strežnik, uporabimo naslednji ukaz, pri katerem navedemo mape, ki jih želimo »restore«-ati (tukaj je ta mapa »zeljenaMapa«).

```
rdiff-backup -r now /home/rdiffbackup/back-up /zeljenaMapa
```

2. V ukazu *rdiff-backup* omogočite izključevanje določenih podmap in dodajte še tedensko varnostno kopiranje. Naredite ločene varnostne kopije še za podatkovno bazo MySQL.

SSHFS (SSH filesystem) je datotečni sistem, ki je ustvarjen na osnovi SSH protokola za prenos datotek. Uporablja se ga za priklop (mount) in interakcijo z datotekami na oddaljenem strežniku preko normalne SSH povezave.

Če želimo iz backup-a izključiti določene mape, lahko to storimo tako, da v vse take mape dodamo datoteko z enakim imenom (npr. »izkljuci-mapo«) in nato spremenimo vrstico prej napisane skripte *backup.sh*:

```
rdiff-backup -exclude-if-present izkljuci-mapo /home/rdiffbackup/mount  
/home/rdiffbackup/back-up
```

Za omogočenje tedenskega varnostnega kopiranja ponovno uporabimo ukaz:

```
crontab -e
```

in v datoteko dodamo:

```
00 02 * * 5 rdiff-backup -exclude-if-present izkljuci-mapo /home/rdiffbackup/mount  
/home/rdiffbackup/back-up
```

Za backup podatkovne baze sem namestila orodje automysqlbackup in mu v datoteki */etc/default/automysqlbackup* spremenila mapo, v katero se back-ups shranjujejo – zdaj je ta mapa *~/mycodes/automysqlbackup*. Tako se mapa backupa skupaj z ostalimi datotekami v *~/mycodes* direktoriju.

```
sudo apt-get install automysqlbackup  
sudo automysqlbackup  
sudo nano /etc/default/automysqlbackup
```

```
...  
# Backup directory location e.g /backups  
# Folders inside this one will be created (daily, weekly, etc.), and the  
# subfolders will be database names. Note that backups will be owned by  
# root, with Unix rights 0600.  
BACKUPDIR="/home/mycodes/automysqlbackup"  
...
```

3. Omogočite samodejni priklop in izklop oddaljenega podsistema *sshfs*. (namig: generirajte dodaten ključ SSH, za izklop pa uporabite ukaz *fusermount -u*)

Najprej sem zgenerirala drugi ssh ključ in ga dodala med *authorized\_keys* na strežniku (postopek enak kot prej).

Samodejni priklop sem uredila z spremembo */etc/fstab* datoteke:

```
sshfs#mycodes@164.8.252.141:/home/mycodes /home/rdiffbackup/mount fuse.sshfs  
delay_connect,_netdev,user,idmap=user,transform_symlinks,identityfile=/home/rdiffbackup/.ss  
h/second,allow_other,default_permissions,uid=1001,gid=1001 0 0
```



4. S pomočjo strežnika cron sprožite dnevno testiranje in prevajanje izvirne programske kode iz strežnika subversion. (namig: nightly-build)

Apache subversion (svn) je orodje, ki se ga uporablja za spremljanje verzij izvornih kod (trenutne in tistih, ki niso več aktualne).

Najprej sem na backup strežniku pognala naslednji ukaz, da sem naložila orodje subversion.

```
sudo apt-get install subversion subversion-tools
```

Delovanje ukaza svn lahko preverimo s:

```
svn co http://svn.apache.org/repos/asf/subversion/trunk subversion
```

Nato namestimo še ostalo potrebno programsko opremo:

```
sudo apt-get install autoconf libtool
```

```
sudo apt-get install libapr1-dev libaprutil1-dev
```

Za tem sem ustvarila skripto nightlybuild.sh:

```
#!/bin/sh

svn co http://svn.apache.org/repos/asf/subversion/trunk subversion
cd /home/rdiffbackup/subversion
sh ./autogen.sh
./configure
make
sudo make install
```

Ji dodala pravice za izvajanje in z naslednjim ukazom njeno izvajanje dodala v crontab datoteko.

```
chmod +x /home/rdiffbackup/nightlybuild.sh
```

```
crontab -e
```

```
0 2 * * * /home/rdiffbackup/nightlybuild.sh
```

5. Aktivirajte strežnik sendmail, ki ob napakah pri dnevnem prevajanju izvirne programske kode pošlje sporočilo obstoječemu uporabniku Linux z imenom "eagle".

Sendmail je starejši SMTP strežnik, ki omogoča pošiljanje e-poštne sporočil med uporabniki.

Za aktivacijo strežnika sendmail je bilo najprej potrebno inštalirati vse datoteke z ukazom:

```
sudo apt-get install sendmail sendmail-base sendmail-bin sendmail-cf mailutils
```

Nato sem ustvarila uporabnika »eagle« in ga v datoteki /etc/aliases dodala kot postmaster:

```
postmaster: root, eagle
eagle: eagle@localhost
```

Po končanem urejanju te datoteke, je potrebno zagnati ukaz, ki na novo generira datoteko, ki smo jo urejali.

```
sudo newaliases
```

Nato preverimo datoteko /etc/mail/sendmail.mc, da vidimo, če so DAEMON\_OPTIONS vrstice pravilno konfigurirane.

Nato spremenimo še prej ustvarjeno skripto in ji dodamo eno vrstico:

```
#!/bin/sh

svn co http://svn.apache.org/repos/asf/subversion/trunk subversion
cd /home/rdiffbackup/subversion
sh ./autogen.sh
./configure
make 2>errorr.log
sudo make install
```

Kot zadnje pa še v crontab dodamo eno vrstico, ki bo omogočila dnevno pošiljanje datoteke error.log:

```
00 02 * * * sendmail eagle<error.log
```

## Sklop 6: Samodejni ponovni zagon ob izpadu, izdelava prikrojenega namestitvenega paketa

Večkrat se lahko zgodi, da se ob primeru napake operacijski sistem Linux ne odziva več. Da se temu izognemo, lahko uporabimo strežnik heartbeat, ki testira odzivnost operacijskega sistema Linux avtomatsko in v primeru ne odzivanja, tega ponovno zažene. S tem sicer napake, ki je povzročila zastoj sistema, ne odpravimo, njegovo ne odzivnost pa zmanjšamo na čas ponovnega zagona. Pri novem nameščanju operacijskega sistema Linux dobimo privzeti sistem, ki običajno ni prikrojen zahtevam naših uporabnikov. Če želimo namestiti že prikrojen sistem, je potrebno izdelati t.i. namestitveni paket. Obe menjeni opciji sta zajeti v naslednjih uporabniških zgodbah:

1. *Namestite strežnik heartbeat, ki nadzoruje odzivnost operacijskega sistema Linux in v primeru izpada obvesti administratorja. Kaj je potrebno spremeniti na sistemu Linux, da postavimo delujoče okolje za strežnik heartbeat? (namig: ha.cf)*

Najprej namestimo heartbeat na oba računalnika, na node in node2. To storimo z ukazom `apt-get install heartbeat`. Potem moramo skonfigurirati heartbeat na obema računalnikoma. Za to moramo urediti 3 datoteke: *authkeys*, *ha.cf* in *haresources*. Pred konfiguracijo je potrebno še vse 3 datoteke kopirati v **/etc/ha.d/**. Prav tako je bilo potrebno odpakirati *haresources.gz* in *ha.cf.gz*. To sem storila za ukazi:

```
cp /usr/share/doc/heartbeat/authkeys /etc/ha.d/
gunzip haresources.gz
gunzip ha.cf.gz
odpakiranje v datoteki /usr/share/doc/heartbeat/
cp /usr/share/doc/heartbeat/ha.cf /etc/ha.d/
cp /usr/share/doc/heartbeat/haresources /etc/ha.d/
```

Po tem se začela s konfiguracijo. Najprej sem dodala v datoteko **/etc/ha.d/authkeys/** dodala `auth2`

```
2 sha1 test-ha
```

To pomeni, da sem za avtentikacijo izbrala metodo 2, ki je sha1. Prav tako sem spremenila pravice datoteki, da lahko lastnik to datoteko bere in v njo piše (`chmod 600 /etc/ha.d/authkeys`). Po tem sem v datoteko **/etc/ha.d/ha.cf** dodala:

```
logfile /var/log/ha-log
logfacility local0
keepalive 2
deadtime 30
initdead 120
bcast eth0
udpport 694
auto_failback on
node ubuntuSP
```

```
node ursy-VirtualBox
```

Ta korak je najpomembnejši, saj v to datoteko zapišemo celotno dogajanje strežnika heartbeat, kdo so naša vozlišča (node), kakšna so naša udp vrata (udpport). Nazadnje sem uredila še datoteko **/etc/ha.d/haresources/**, v katero sem dodala:

```
ubuntuSP 164.8.252.142 httpd
```

Ta datoteka vsebuje informacije o tem, kateri vir želimo omogočiti. Prikazan je primer za webserver (httpd). Tako so datoteke pripravljene za zagon strežnika heartbeat. Pred tem je bilo potrebno še:

- Skopirati **/etc/ha.d/** iz node1, v node2 (`scp -r /etc/ha.d/ root@ursy-VirtualBox:/etc/`)
- Ker smo želeli omogočiti httpd, sem morala spremeniti še datoteko **/etc/httpd/conf/httpd.conf**, in sicer tako, da sem dodala `Listen 164.8.252.142:80`.
- To spremenjeno datoteko sem kopirala še na node2 (`scp /etc/httpd/conf/httpd.conf root@ursy-VirtualBox:/etc/httpd/conf/`)
- Za demonstracijo sem ustvarila index2.html na obeh grozdih (ubuntuSP in ursy-VirtualBox):  

```
echo "ubuntuSP/node01 apache test server" > /var/www/html/index2.html
```

```
echo "ursy-VirtualBox/node02 apache test server" >
/var/www/html/index2.html
```

Zadnji korak, je da zaženemo heartbeat strežnik (**/etc/init.d/heartbeat start**), ter da potestiramo, če dela stran index2.html na obeh računalnikih oz. grozdij (ubuntuSP in ursy-VirtualBox).

## *2. Izdelajte namestitveni paket Debian, ki namesti skripte potrebne za namestitev in konfiguracijo vašega projekta*

V home direktoriju sem ustvarila datoteko FERImdb, ki je ime našega projekta in hkrati ime .deb datoteke. V FERImdb sem skopirala vse iz datoteke /var/www, ki vsebuje vse datoteke našega projekta. V FERImdb sem ustvarila DEBIAN poddirektorij in v njega dodala datoteko: control. V njo sem zapisala:

```
Package: FERImdb
Architecture: all
Maintainer: Urška Nemet, Anja Hauptman, Dominik Šbüll, Simona Siljanovska
Depends: php5, mysql-server, mysql-client, apache2, phpmyadmin, deb.conf
Priority: optional
Version: 1.0
Description: Projekt skupine FERImdb.
```

Vse datoteke so sedaj v direktoriju in lahko začnemo z kreiranjem .deb paketa. Najprej spremenimo lastnika na root vsem datotekam z ukazom:

```
chown -R root:root FERImdb/FERImdb_1.0/
```

Potem še ustvarimo paket: `dpkg-deb --build FERImdb/FERImdb_1.0`

3. V namestitvenem paketu Debian omogočite konfiguracijo paketa, ki uporabniku v primeru interaktivnega nameščanja omogoči vnos privzetih gesel.

V datoteko control dodamo `Type: [password]`.

4. Strežniku cron dodajte skripto za tedensko testiranje varnosti gesel. (namig: uporabite program john)

- najprej sem napisala skripto, ki preverja gesla (preveriGesla.sh), spodaj je vsebina skripte:

```
#naredimo kombinacijo uporabniškega imena in gesla
john unshadow /etc/passwd /etc/shadow > /home/ursy/kombinacijaGesla.txt

#preveri gesla s pomocjo liste najpogostejših gesel
john -wordlist:gesla.lst /home/ursy/kombinacijaGesla.txt > /home/ursy/statusPreverjanja.txt
```

- nato sem s pomočjo ukaza `crontab -e` dodala naslednjo vrstico v cron, ki preverja vsaki teden gesla (ob polnoči v soboto):

```
0 0 * * 6 /home/ursy/preveriGesla.sh
```

5. Namestite strežnik za zaznavo, poročanje in preprečevanje vdorov. (namig: denyhosts, rkhunter, snort)

- denyhosts sem namestila z naslednjim ukazom `sudo apt-get install denyhosts` nato sem v `/etc/hosts.allow` dodala svoj IP, ki mi omogoča `sshd: 1.1.1.1` dostop do strežnika ostale vse, sem pa blokirala v `/etc/denyhosts.conf`, dodala pa sem naslednjo vrstico `sshd: ALL`, na koncu sem restartala denyhosts, da sem ohranila nove nastavitve:

```
sudo /etc/init.d/denyhosts restart
```

- rkhunter sem namestila z naslednjim ukazom `sudo apt-get install rkhunter`
  - najprej sem ga updetala `sudo rkhunter -update`
  - nato pa sem pognala pregled za zlonamerne kode `sudo rkhunter -c --enable all --disable none`
  - rezultat analize pa sem prikazala z ukazom `cat /var/log/rkhunter.log`
- za zaščito ssh dostopa sem si še namestila fail2ban `sudo apt-get install fail2ban`
  - najprej sem skopirala `cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local`
  - nato sem uredila konfiguracijsko datoteko tako da te ob 3 napačnih poizkusih vnesenega gesla fail2 ban doda na črno listo v iptables in ne moreš več dostopat do strežnika preko ssh, ter mi pošlje mail na mail na server accountu ursy

**`/etc/fail2ban/jail.local`**

```
[ssh-iptables]

enabled = true

port = ssh

filter = sshd
```

```
action    = iptables[name=SSH, port=ssh, protocol=tcp]
            sendmail-whois[name=SSH, dest=ursy, sender=fail2ban@ursy-server.com]
logpath   = /var/log/auth.log
maxretry  = 3
```

## Sklop 9: Napredna sistemska administracija Linux


V tem sklopu se ukvarjamo z naprednimi opravili, s katerimi se soočajo sistemski administratorji na operacijskem sistemu Linux. V uporabniških zgodbah se dotaknemo postopka nadgradnje jedra operacijskega sistema Linux, namestitve in konfiguracije elektronske pošte, upravljanjem in spremljanjem sistemskih dnevnikov, delovanja tiskalnikov in izmenjalnega prostora. Študenti rešujejo naslednje uporabniške zgodbe:

*1. Nadgradite jedro sistema Linux. Postopek nadgradnje podrobneje opišite.*


Najlažja ter skoraj popolnoma avtomatska metoda nadgradnje je tista z ukazom »`sudo apt-get upgrade`«, tukaj se bom lotil čisto ročnega postopka nadgradnje mojega jedra z kernel-om najnovejše različice, ki je v trenutku pisanja tega 4.0.4 in jo lahko najdemo na spletnem naslovu »<https://www.kernel.org>«.

# The Linux Kernel Archives

[About](#) [Contact us](#) [FAQ](#) [Releases](#) [Signatures](#) [Site news](#)



Protocol	Location
<a href="#">HTTP</a>	<a href="https://www.kernel.org/pub/">https://www.kernel.org/pub/</a>
<a href="#">GIT</a>	<a href="https://git.kernel.org/">https://git.kernel.org/</a>
<a href="#">RSYNC</a>	<a href="rsync://rsync.kernel.org/pub/">rsync://rsync.kernel.org/pub/</a>

**Latest Stable Kernel:**  
 **4.0.4**

mainline:	<b>4.1-rc6</b>	2015-06-01	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a>
stable:	<b>4.0.4</b>	2015-05-17	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
stable:	<b>3.19.8 [EOL]</b>	2015-05-11	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>3.18.14</b>	2015-05-20	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>3.14.43</b>	2015-05-17	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>3.12.43</b>	2015-05-20	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>3.10.79</b>	2015-05-17	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>3.4.107</b>	2015-04-14	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>3.2.69</b>	2015-05-09	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>2.6.32.66</b>	2015-05-24	<a href="#">[tar.xz]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
linux-next:	<b>next-20150602</b>	2015-06-02		<a href="#">[browse]</a>

Po prenosu kernel-a bom ustvaril novo direktorij in vanj prenesel datoteko tar ki vsebuje kernel, to storim z naslednjim zaporedjem ukazov:

```
mkdir kernel
mv linux-4.0.4.tar.xz kernel
```

```
x kernel: ls
+ x kernel: ls
dominiksb@dominiksb-virtual-machine:~$ mkdir kernel
dominiksb@dominiksb-virtual-machine:~$ ls
4.0-sched-bfs-462.patch linux-4.0.4.tar.xz Templates
Documents Music Videos
Downloads Music Pictures
kernel Public
dominiksb@dominiksb-virtual-machine:~$ mv linux-4.0.4.tar.xz
kernel
dominiksb@dominiksb-virtual-machine:~$ mv 4.0-sched-bfs-462.p
atch kernel
dominiksb@dominiksb-virtual-machine:~$ ls
Documents kernel Pictures Templates
Downloads Music Public Videos
dominiksb@dominiksb-virtual-machine:~$ cd kernel
dominiksb@dominiksb-virtual-machine:~/kernel$ ls
4.0-sched-bfs-462.patch linux-4.0.4.tar.xz
dominiksb@dominiksb-virtual-machine:~/kernel$
```

Sedaj rabim še config datoteko ki sem jo dobil ob prvem kompiliranju oziroma namestitvi sistema in jo lahko najdem v direktoriju »/boot«. Ta config file sedaj premaknem oziroma kopiram v direktorij kernel ki sem ga prej ustvaril z ukazom `cp /boot/config-3.16.0-34-generic /home/dominiksb/kernel`, nato se premaknem v samo mapo kernel in z ukazom `ls` preverim ali je bila datoteka uspešno kopirana.

```
x kernel: ls
+ x kernel: ls
config-3.16.0-34-generic System.map-3.16.0-34-generic
grub vmlinuz-3.16.0-34-generic
dominiksb@dominiksb-virtual-machine:/boot$ cp /boot/config-3.
16.0-34-generic /home/dominiksb/kernel
dominiksb@dominiksb-virtual-machine:/boot$ cd..
cd..: command not found
dominiksb@dominiksb-virtual-machine:/boot$ cd ..
dominiksb@dominiksb-virtual-machine:/$ ls
bin dev initrd.img media proc sbin sys var
boot etc lib mnt root srv tmp vmlinuz
cdrom home lost+found opt run swapfile usr
dominiksb@dominiksb-virtual-machine:/$ cd home
dominiksb@dominiksb-virtual-machine:/home$ cd
dominiksb@dominiksb-virtual-machine:~$ ls
Documents kernel Pictures Templates
Downloads Music Public Videos
dominiksb@dominiksb-virtual-machine:~$ cd kernel
dominiksb@dominiksb-virtual-machine:~/kernel$ ls
4.0-sched-bfs-462.patch linux-4.0.4.tar.xz
config-3.16.0-34-generic
dominiksb@dominiksb-virtual-machine:~/kernel$
```

Sedaj je potrebno preneseni kernel file v mojem primeru »linux-4.0.4.tar.xz« razširiti oziroma razpakirati to storim z ukazom `tar -xvf linux-4.0.4.tar.xz` nato počakam da se proces konča.

Ko s tem končam se povežem »linux-4.0.4« z direktorijem »linux« in sicer z ukazom `ln -s linux-4.0.4 linux`. Nato se pomaknem v direktorij »linux« z »`cd linux`« ter zaženem makefile z ukazom »`make clean && make mrproper`« da pripravim datoteke za kompiliranje.

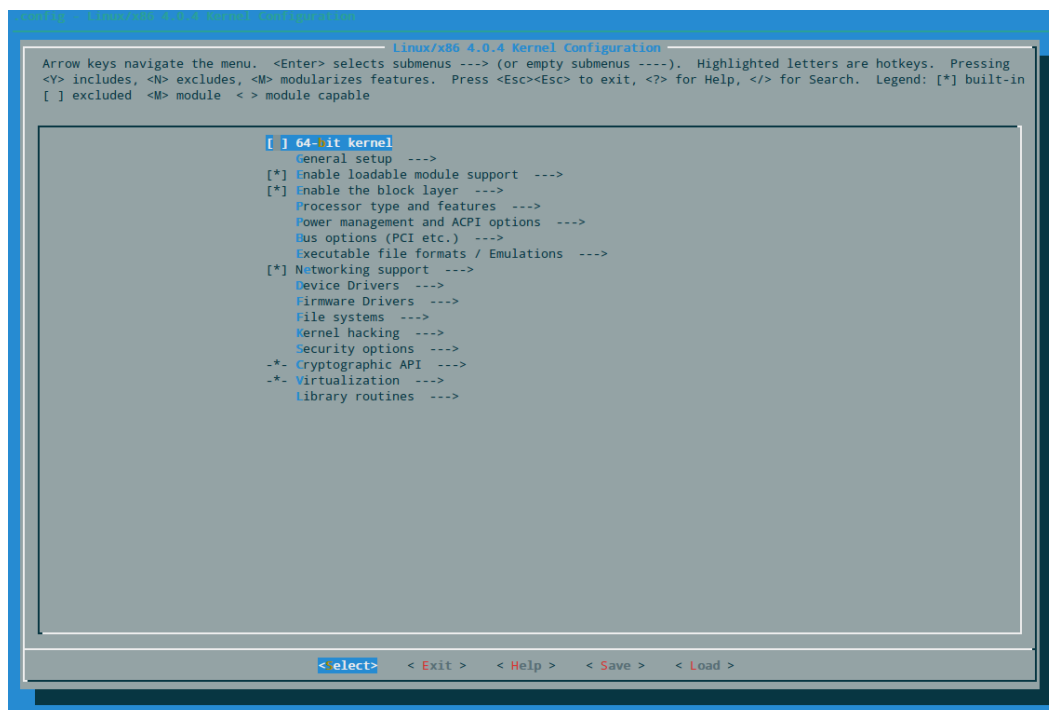


```

dominiksb@dominiksb-virtual-machine:~/kernel$ ln -s linux-4.0
.4 linux
dominiksb@dominiksb-virtual-machine:~/kernel$ ls
4.0-sched-bfs-462.patch  linux  linux-4.0.4.tar.xz
config-3.16.0-34-generic linux-4.0.4
dominiksb@dominiksb-virtual-machine:~/kernel$ cd linux
dominiksb@dominiksb-virtual-machine:~/kernel/linux$ ls
arch      fs      MAINTAINERS  security
block     include Makefile      sound
COPYING   init    mm            tools
CREDITS   ipc     net           usr
crypto    Kbuild  README       virt
Documentation Kconfig REPORTING-BUGS
drivers   kernel  samples
firmware  lib     scripts
dominiksb@dominiksb-virtual-machine:~/kernel/linux$ make clea
n && make mrproper
dominiksb@dominiksb-virtual-machine:~/kernel/linux$

```

V naslednjem koraku se premaknem en direktorij višje ter kopiram prej kopirano konfiguracijsko datoteko v direktorij ter jo preimenujem v config »linux« z ukazom »`cp config-3.16.0-34-generic linux/.config`«. Sedaj se premaknem nazaj v direktorij »linux« ter zaženemo ukaz »`make menuconfig`«. Prikaže se nekaj takega:

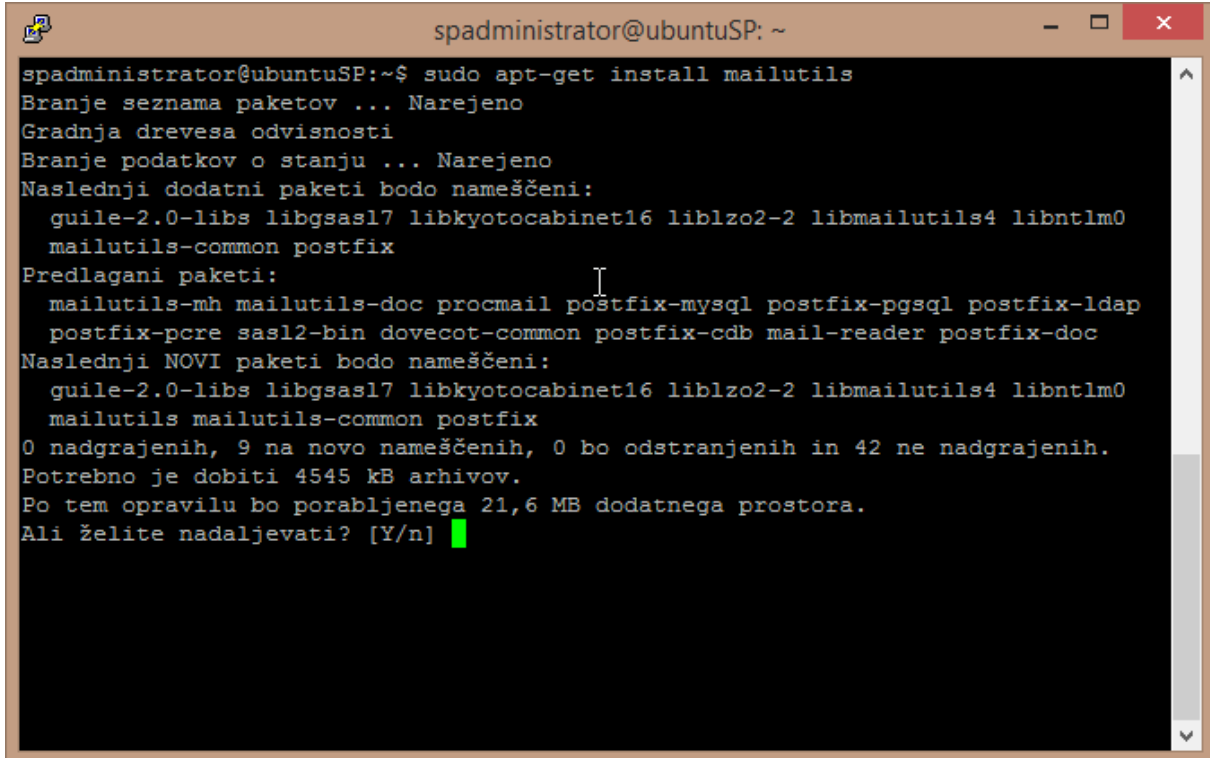


Tukaj imam zdaj možnost spreminjanja nastavitev mojega jedra, za demonstracijo bom izbral samo eno stvar in sicer za optimizacijo mojega sistema in sicer grem pod »Processor type and features« nato pa poiščem »Preemption Model« ter spremenim opcijo na »Preemptible Kernel (Low-Latency Desktop)«. Sedaj ko sem končal z modifikacijo mojega kernela se pomaknem na save in potrdim.

Naslednji korak je kompiliranje jedra to storimo z sledečimi ukazi »`fakerooot kpkg -j 3`«, »`--initrd`«, »`--append-to-version=-dominik-test-bfs kernel_image kernel_headers`«, sedaj bo moj sistem začel kompiliranje jedra. Počakam ter dobimo 2 .deb paketa v našem »kernel« direktoriju sedaj za namestitev le teh poženem ukaz »`sudo dpkg -i`

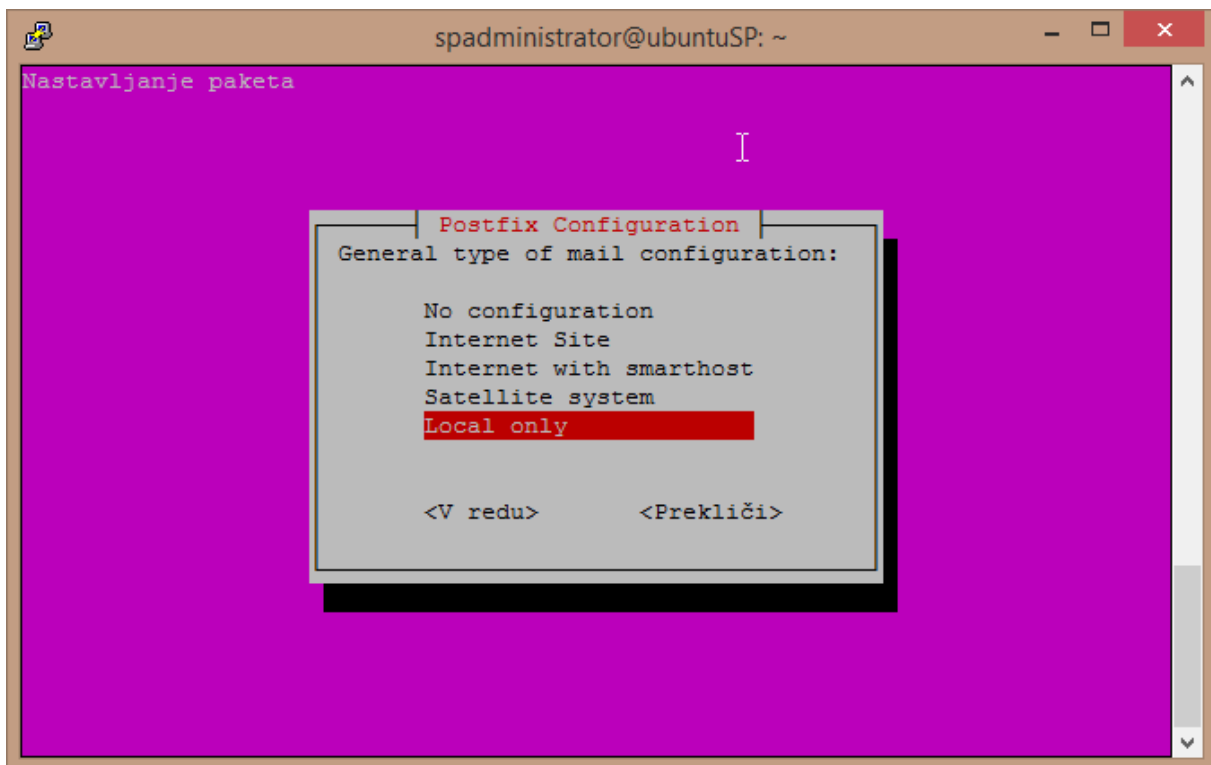
\*.deb«. Sedaj počakam da dokonča namestitev nadgrajenega jedra in GRUB-a ter ponovno zaženem računalnik in je proces končan.

2. Namestite in konfigurirajte strežnik elektronske pošte (npr. Postfix) na operacijskem sistemu Linux. Postopke podrobneje opišite.

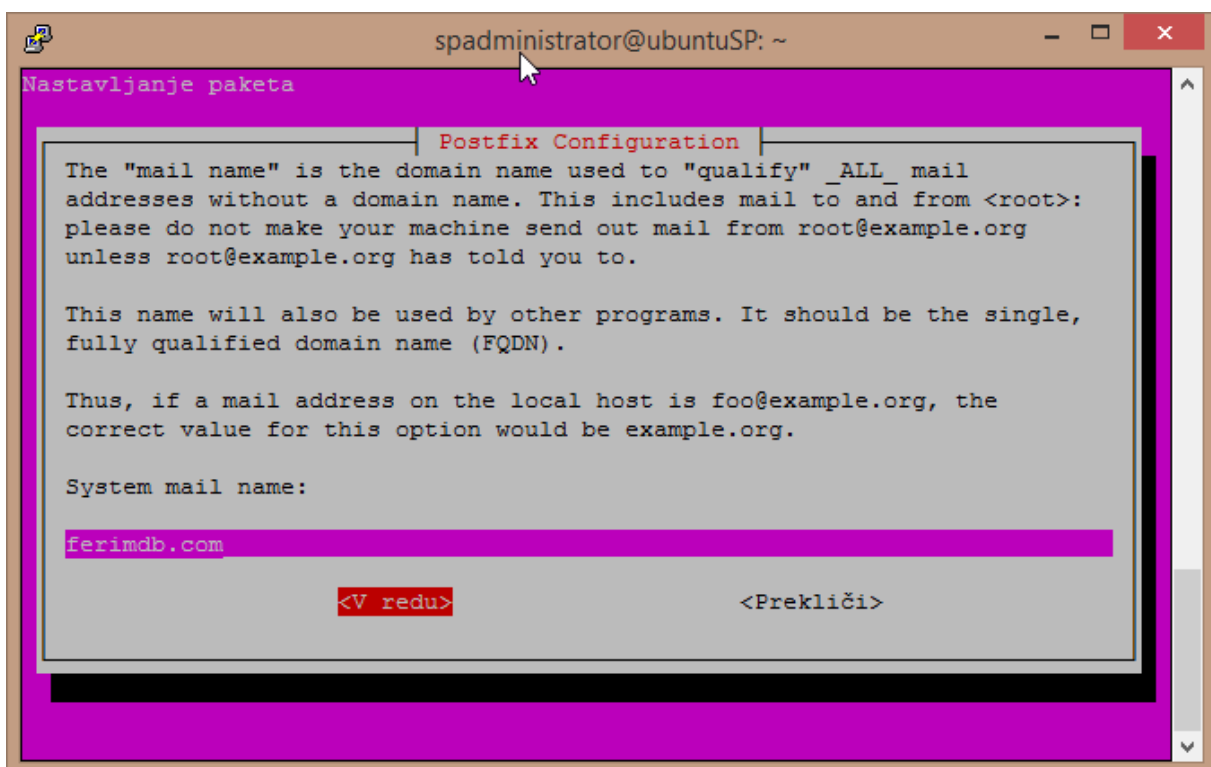


```
spadministrator@ubuntuSP: ~  
spadministrator@ubuntuSP:~$ sudo apt-get install mailutils  
Branje seznama paketov ... Narejeno  
Gradnja drevesa odvisnosti  
Branje podatkov o stanju ... Narejeno  
Naslednji dodatni paketi bodo nameščeni:  
  guile-2.0-libs libgsasl7 libkyotocabinet16 liblzo2-2 libmailutils4 libntlm0  
  mailutils-common postfix  
Predlagani paketi:  
  mailutils-mh mailutils-doc procmail postfix-mysql postfix-pgsql postfix-ldap  
  postfix-pcre sasl2-bin dovecot-common postfix-cdb mail-reader postfix-doc  
Naslednji NOVI paketi bodo nameščeni:  
  guile-2.0-libs libgsasl7 libkyotocabinet16 liblzo2-2 libmailutils4 libntlm0  
  mailutils mailutils-common postfix  
0 nadgrajenih, 9 na novo nameščenih, 0 bo odstranjenih in 42 ne nadgrajenih.  
Potrebno je dobiti 4545 kB arhivov.  
Po tem opravilu bo porabljenega 21,6 MB dodatnega prostora.  
Ali želite nadaljevati? [Y/n]
```

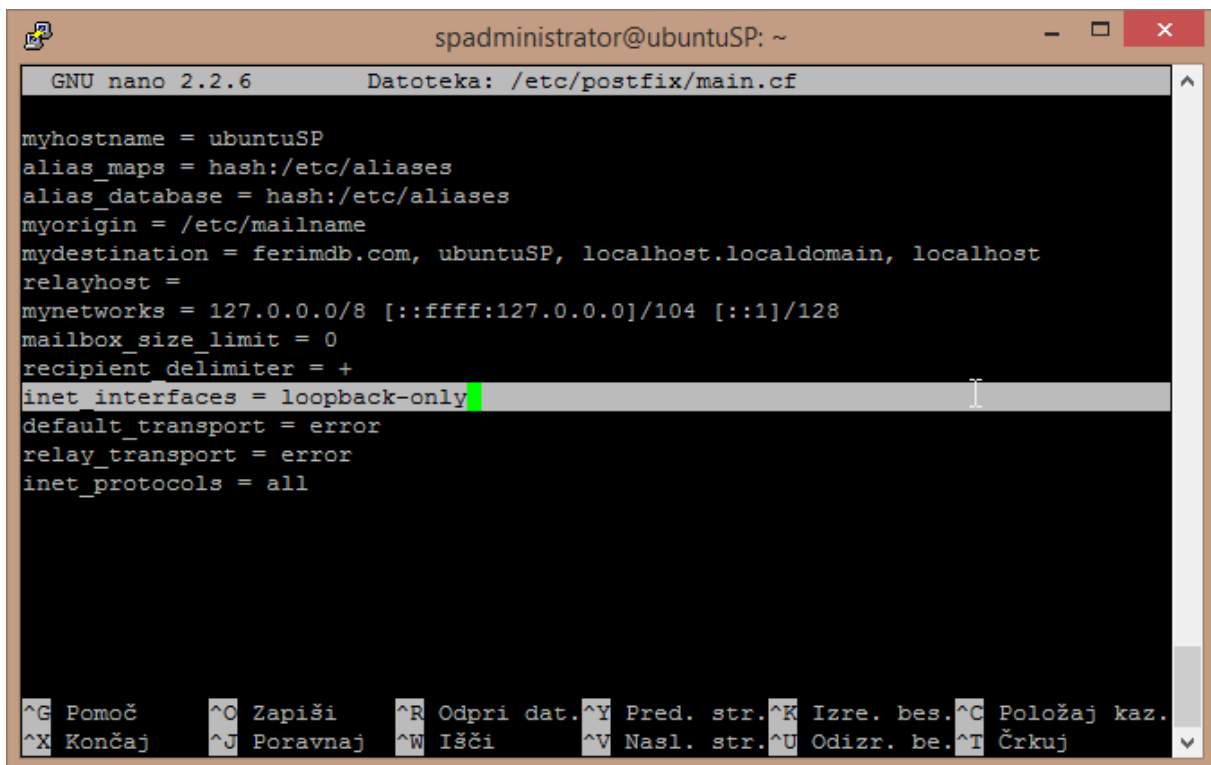
Prvi korak, ki ga je pri tem potrebno izvesti je vpis komande »`sudo apt-get install mailutils`«. Kot vidimo na spodnji sliki ni potrebno posebej nameščati »postfix« saj je paket in vse potrebno že vključeno v prvo omenjeno komando.



Ker na strežniku, ki je na razpolago nam nimamo možnosti upravljanja z DNS-i ter nimamo domene, se kar se tiče konfiguracije lahko osredotočimo le na lokalno.

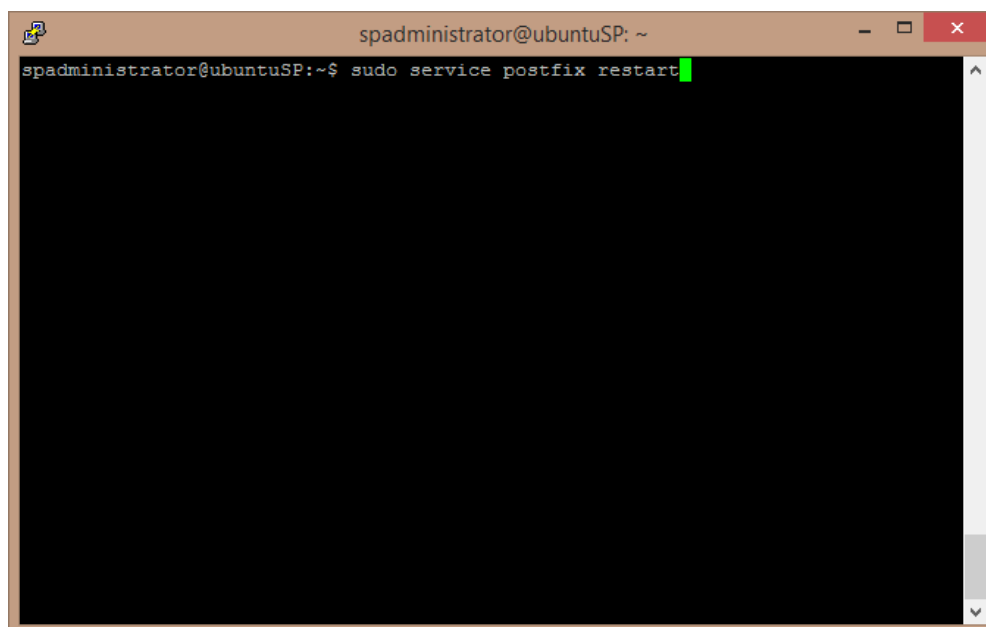


V naslednjem koraku si izberemo poljubno lokalno domeno katera bo pripeta vsakemu lokalnemu računu in jo lahko uporabljamo za pošiljanje lokalnih mail-ov npr. »XXXXXX@ferimdb.com«, kjer je XXXXX recimo ime uporabniškega računa.



```
spadministrator@ubuntuSP: ~  
GNU nano 2.2.6      Datoteka: /etc/postfix/main.cf  
  
myhostname = ubuntuSP  
alias_maps = hash:/etc/aliases  
alias_database = hash:/etc/aliases  
myorigin = /etc/mailname  
mydestination = ferimdb.com, ubuntuSP, localhost.localdomain, localhost  
relayhost =  
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128  
mailbox_size_limit = 0  
recipient_delimiter = +  
inet_interfaces = loopback-only  
default_transport = error  
relay_transport = error  
inet_protocols = all  
  
^G Pomoč    ^O Zapiši    ^R Odpri dat.^Y Pred. str.^K Izre. bes.^C Položaj kaz.  
^X Končaj   ^J Poravnaj ^W Išči     ^V Nasl. str.^U Odizr. be.^T Črkuj
```

V naslednjem koraku je potrebno preveriti konfiguracijsko datoteko z ukazom »`sudo nano /etc/postfix/main.cf`«, tukaj preverimo ali je kot na sliki označen del »`inet_interfaces`« res nastavljen na »`loopback-only`«.



```
spadministrator@ubuntuSP: ~  
spadministrator@ubuntuSP:~$ sudo service postfix restart
```

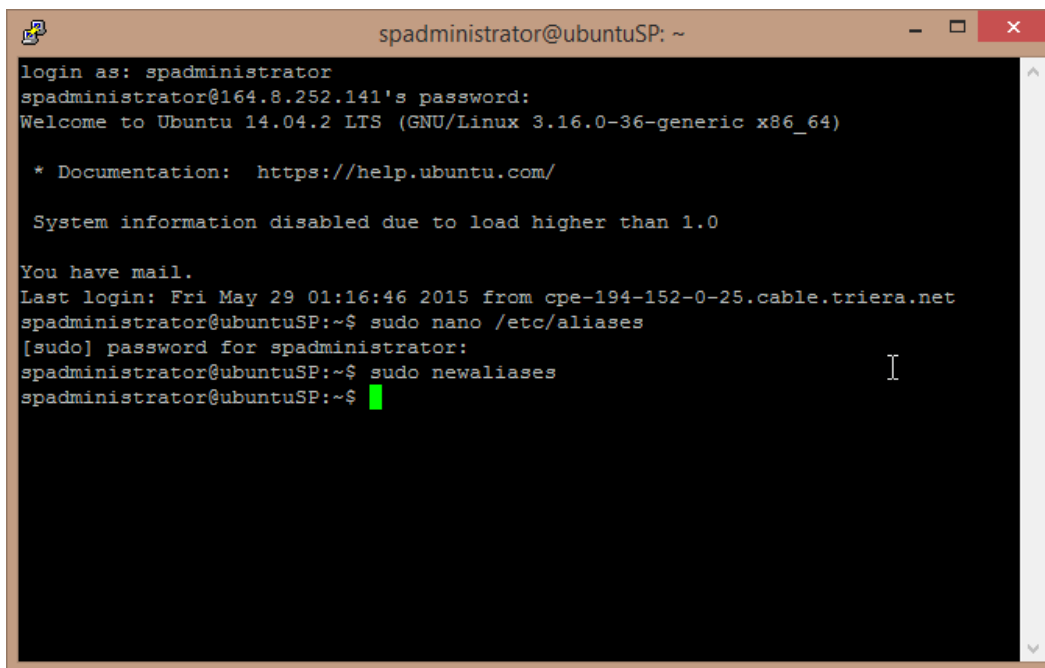
Nato ponovno zaženemo service postfix-a z ukazom »`sudo service postfix restart`«. Sedaj lahko pošljemo še testni mail našemu uporabniku kar storimo z »`echo 'Testno sporocilo' | mail -s 'To je zadeva sporocila' spadministrator@ferimdb.com`«.

```
spadministrator@ubuntuSP: ~  
login as: spadministrator  
spadministrator@164.8.252.141's password:  
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-36-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
System information disabled due to load higher than 1.0  
  
You have mail.  
Last login: Fri May 29 01:16:46 2015 from cpe-194-152-0-25.cable.triera.net  
spadministrator@ubuntuSP:~$ sudo nano /etc/aliases
```

Na zgornji sliki sedaj vidimo da nas čaka naš poslani mail. (»You have mail.«)  
Sedaj lahko še nastavimo »mail-forwarding« tako da dobimo mail tudi na naš ferimdb.com mail račun iz roota, v primeru da smo mi administrator. To storimo z ukazom »`sudo nano /etc/aliases`«.

```
GNU nano 2.2.6      Datoteka: /etc/aliases      Spremenjeno  
# See man 5 aliases for format  
postmaster:      root  
root:            admin@ferimdb.com
```

Sedaj lahko nastavimo recimo root računu privzeti ferimdb.com mail račun da npr. ob pisanju sporočila ne rabimo na koncu dodati kompletne domene npr. »admin@ferimdb.com« ampak lahko vnesemo namesto le tega samo uporabniško ime kot npr. v tem primeru »root«.

A terminal window titled 'spadministrator@ubuntuSP: ~' with standard window controls. The terminal shows a login sequence for 'spadministrator' on '164.8.252.141'. It displays the Ubuntu 14.04.2 LTS welcome message, documentation link, and system information. After a mail notification, the user runs 'sudo nano /etc/aliases', enters their password, and then runs 'sudo newaliases'. The prompt returns to '~\$' with a green cursor.

```
login as: spadministrator
spadministrator@164.8.252.141's password:
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information disabled due to load higher than 1.0

You have mail.
Last login: Fri May 29 01:16:46 2015 from cpe-194-152-0-25.cable.triara.net
spadministrator@ubuntuSP:~$ sudo nano /etc/aliases
[sudo] password for spadministrator:
spadministrator@ubuntuSP:~$ sudo newaliases
spadministrator@ubuntuSP:~$
```

Sedaj spremembe samo še uveljavimo in smo končali za namestitvijo ter konfiguracijo poštnega strežnika POSTFIX.

### 3. Kakšne načine upravljanja s sistemskimi dnevniki poznamo pod operacijskim sistemom Linux.

Poznamo dve možnosti upravljanja s sistemskimi dnevniki in sicer ročno ter programsko.

Ročno lahko beremo vnose na točno določenih lokacijah na našem sistemu kot npr. za vsako vpis(authentication) lahko najdemo datoteko na lokaciji »/var/log/auth.log«. Ponavadi nas zanimajo le zadnji vnosi, le te si lahko prikažemo z uporabo ukaza »last«, ta nam ponudi izpis zadnjih vnosov datoteke »/etc/log/wtmp«. Naslednji primer nam recimo lahko prikaže zadnje termine vpisov s sistem za izpis le teh uporabimo ukaz »lastlog«, ki nam vrne rezultate iz datoteke »/etc/log/lastlog«.

Za programsko upravljanje lahko uporabimo eno od obstoječih programskih rešitev kot npr. »Logrotate«, ki ga lahko namestimo z ukazom »**sudo apt-get install logrotate**« nato pa preverimo če deluje z ukazom »logrotate«. Nastavitve najdemo v konfiguracijski datoteki »/etc/logrotate.d/«. Podajmo primer za vodenje dnevnikov dpkg(Debian Package Management System).

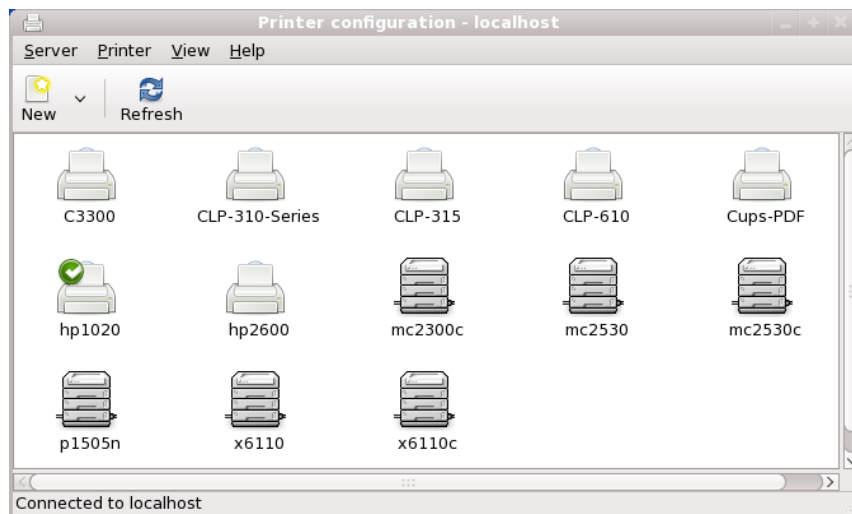
```
/var/log/dpkg.log {
    monthly
    rotate 12
    compress
    delaycompress
    missingok
    notifempty
    create 644 root root
}
```

- the logrotation for dpkg monitors the /var/log/dpkg.log file and does this on a monthly basis - this is the rotation interval.
- 'rotate 12' signifies that 12 days worth of logs would be kept.
- logfiles can be compressed using the gzip format by specifying 'compress' and 'delaycompress' delays the compression process till the next log rotation. 'delaycompress' will work only if 'compress' option is specified.
- 'missingok' avoids halting on any error and carries on with the next log file.
- 'notifempty' avoid log rotation if the logfile is empty.
- 'create <mode> <owner> <group>' creates a new empty file with the specified properties after log-rotation.

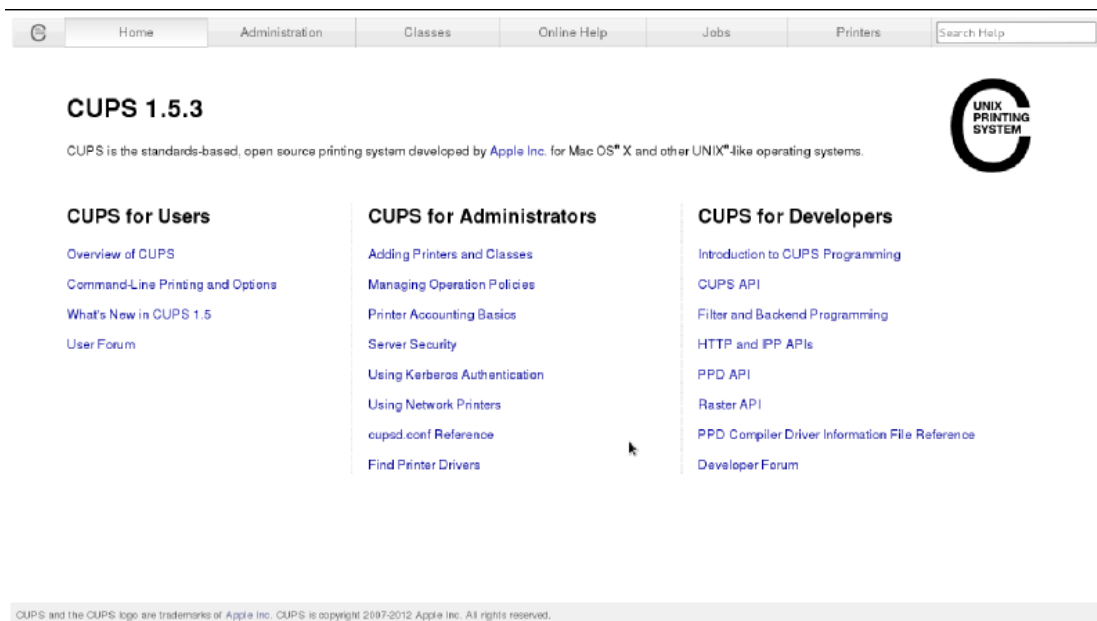
Zgoraj sedaj vidimo del kode v konfiguracijski datoteki, ter na desni opis vsake podane opcije(rezervirane besede). V primeru da bi želeli dnevnik posodablјati dnevno, lahko dodamo vnos v »/etc/cron.daily/logrotate«.

#### 4. Kako upravljamo tiskalnike pod operacijskim sistemom Linux.

Tiskalnike na večini distribucijskih sistemov lahko upravljamo na dva načina in sicer če uporabljamo GUI imamo na voljo orodje, ki se imenuje »system-config-printer« in ga lahko vidimo na spodnji sliki, razlikuje se lahko le malenkostno glede na oblikovanje od distribucije do distribucije. V tej aplikaciji nam je možno upravljati le te kot npr. dodajati lokalne priključene naprave, omrežne tiskalnike, skupno rabo tiskalnikov, itd.



Če imamo na voljo le ukazno vrstico lahko uporabimo za upravljanje le teh orodje, ki se imenuje »CUPS(Common UNIX Printing System)«. Ta varianta je zelo priljubljena predvsem tudi zaradi tega ker nam ob prenosu za samo konfiguracijo in administracijo ponuja možnost le tega na osnovni spletnega vmesnika ki se naloži ob inštalaciji le te. Prenos orodja je možen če vnesemo ukaz »**sudo apt-get install cups**«. Spletni vmesnik za »CUPS« je privzeto na naslovu »http://localhost:631/admin« primer izgleda je na spodnji sliki.



### 5. Kako upravljamo z izmenjalnim prostorom (angl. swap space) na operacijskem sistemu Linux?

S swapom lahko pod sistemom linux upravljamo prosto, ampak ponavadi ga le dodajmo, seveda ga lahko tudi izbrišemo, vklopimo ali izklopimo. Na primeru bom prikazal ustvaritev swap file-a, dodajanje le tega in avtomatski priklop s pomočjo fstab-a.

```
Home: swapon
dominiksb@dominiksb-virtual-machine:~$ sudo swapon -s
Filename                                Type    Size    Used    Priority
/dev/sda5                               partition 2094076 0       -1
dominiksb@dominiksb-virtual-machine:~$
```

Kot prvo si pogledjmo trenutno stanje SWAP-a, kot lahko razberemo z zgornje slike ki sem jo dobil ko sem zagnal ukaz »`sudo swapon -s`« imamo sedaj le eno swap particijo in sicer tisto katero smo ustvarili ob namestitvi našega linux operacijskega sistema. Vedeti moramo namreč da obstajajo tudi swap datoteke, ki pa niso bile uporabljene ker so bile počasnejše in so lahko bolj zavirale kot pomagale sistemu, odkar je izšla različica »kernel 2.6« so sedaj le te datoteke enakovredne tem particijam saj omogočajo enako hitrost in nam olajšajo upravljanje s swap-om v kar precejšnji meri saj ne potrebujemo uporabe disk managerjev s katerim bi novo ustvarili.



```
Home: ls
+ x Home: ls
dominiksb@dominiksb-virtual-machine:~$ sudo swapon -s
Filename                                Type              Size      Used      Priority
/dev/sda5                              partition         2094076 0         -1
dominiksb@dominiksb-virtual-machine:~$ sudo fallocate -l 4G /swapfile
[sudo] password for dominiksb:
dominiksb@dominiksb-virtual-machine:~$ ls -lh /swapfile
-rw-r--r-- 1 root root 4,0G jun  2 02:44 /swapfile
dominiksb@dominiksb-virtual-machine:~$
```

Sedaj ustvarimo 4GB veliki »swap-file« z ukazom »sudo fallocate -l 4G /swapfile«, nato še preverimo če je bil res ustvarjen z ukazom »ls -lh /swapfile«. Kot vidimo zgoraj je file bil uspešno ustvarjen in je velik 4GB.

```
Home: ls
+ x Home: ls
dominiksb@dominiksb-virtual-machine:~$ sudo chmod 600 /swapfile
dominiksb@dominiksb-virtual-machine:~$ ls -lh /swapfile
-rw----- 1 root root 4,0G jun  2 02:44 /swapfile
dominiksb@dominiksb-virtual-machine:~$
```

Naslednji korak je da sistemu povemo da se tukaj gre za datoteko, ki jo naj formatira in uporablja kot swap. Preden storimo le to zaradi varnosti priredimo pravice datoteke tako da bo imel pravice branja le administrator/root. To storimo z ukazom »sudo chmod 600 /swapfile«, nato še enkrat preverimo če so spremembe bile res uveljavljene z ukazom »ls -lh /swapfile«.

```
Home: swapon
dominiksb@dominiksb-virtual-machine:~$ sudo chmod 600 /swapfile
dominiksb@dominiksb-virtual-machine:~$ ls -lh /swapfile
-rw----- 1 root root 4,0G jun  2 02:44 /swapfile
dominiksb@dominiksb-virtual-machine:~$ sudo mkswap /swapfile
Setting up swspace version 1, size = 4194300 KiB
no label, UUID=07de181e-6c01-406d-887a-62a6e72e38be
dominiksb@dominiksb-virtual-machine:~$ sudo swapon /swapfile
dominiksb@dominiksb-virtual-machine:~$ sudo swapon -s
Filename                                Type              Size              Used              Priority
/dev/sda5                               partition         2094076           0                 -1
/swapfile                               file              4194300           0                 -2
dominiksb@dominiksb-virtual-machine:~$
```

Sedaj z ukazom »`sudo mkswap /swapfile`« sistemu povemo da naj nastavi našo swap datoteko kot swap. Sedaj je naša datoteka pripravljena za uporabo, uporabimo jo lahko z naslednjim »`sudo swapon /swapfile`«. Sedaj le še preverimo da so se spremembe uveljavile in da je sedaj naša datoteka zaznan kot swap prostor ponovno izvedemo ukaz »`sudo swapon -s`«.

```
sudo nano /etc/fstab
GNU nano 2.2.6      File: /etc/fstab      Modified

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>          <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=e8d2fd13-d33e-406e-8c3f-7da59f13a0e5 /          ext4      errors=remount-ro 0      $
# swap was on /dev/sda5 during installation
UUID=a73b173e-e2da-43d6-a85b-6bf78c3bff0f none      swap      sw              0      0
/dev/fd0      /media/floppy0  auto    rw,user,noauto,exec,utf8 0      0
/swapfile     none      swap    sw              0      0

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text   ^T To Spell
```

Po navadi bi želeli da postane ta swap datoteka sedaj permanento avtomatsko uporabljena oziroma priklopljena ob zagonu sistema, a le to dosežemo z prireditvijo fstab konfiguracijske datoteke z ukazom »`sudo nano /etc/fstab`« in na koncu le te dodamo:

```
/swapfile    none    swap    sw    0    0
```

nato shranimo ter zapremo datoteko in smo končali.

## Sklop 10: DNS

Osnovna naloga strežnika DNS je preslikava med imenom gostitelja in njegovim naslovom IP. Sistem DNS je porazdeljena podatkovna baza, kjer lokalni strežniki DNS upravljajo z lastno domeno lokalnih imen gostiteljev in izmenjujejo podatke o teh med seboj. Vsak lokalni strežnik DNS, ki nastopa v vlogi sekundarnega strežnika DNS, na vprašanje o imenu gostitelja izven svoje domene posreduje vprašanje primarnemu domenskemu strežniku v spletno omrežje. Uporabniške zgodbe o DNS pa so naslednje:

### Vprasanja:

- 1. Za vašo lokalno domeno namestite strežnik DNS na operacijskem sistemu Linux. Katere konfiguracijske datoteke potrebujemo pri postavljanju strežnika DNS?*
- 2. Razložite funkcije naslednjih zapisov v konfiguraciji DNS: SOA, PTR, A, MX in CNAME.*
- 3. Kaj je razlika med avtoriziranimi in neavtoriziranimi odgovori strežnika DNS na vprašanja odjemalcev? Kako lahko zagotovimo, da postanejo odgovori avtorizirani?*
- 4. Kakšno je ime vašega lokalnega strežnika DNS? Kako poteka razreševanje (angl. resolving) imena `www.google.com`, za katerega predpostavljamo, da ni v lokalni domeni?*
- 5. Ustvarite zapis SPF za nadzor spam-a v vaši lokalni domeni.*

### Odgovori:

1. Za našo lokalno domeno na operacijskem sistemu Linux, namestimo DNS strežnik tako da najprej namestimo `bind9` paketa, ki naredimo z naslednjega ukaza:

```
sudo apt-get install bind9
```

Obstaja veliko načinov da konfiguriramo `bind9`. Najpogostejše konfiguracije so `caching nameserver`, `primary master` in `secondary master`. Vse DNS konfiguracijske datoteke so shranjene v **/etc/bind** direktorij. Primarna konfiguracijska datoteka je:

```
/etc/bin/named.conf
```

Privzeta konfiguracija dela kot `caching nameserver`. `Caching nameserver` dela tako da si zapomni vse DNS poizvedbe in potem se uporablja lokalno, kdaj se domeno izvede drugač. Če gremo ga konfigurirat kot `caching nameserver`, še kaj potrebujemo naredit, je da dodamo javnih IP naslovov DNS strežnikov (kot so primer, javnih IP naslovov DNS strežnikov `8.8.8.8` in `8.8.4.4` ki jih Google uporablja) v konfiguracijski datoteki **named.conf.options**. To dodamo tako da odpremo datoteko **named.conf.options** z ukazom:

```
sudo nano /etc/bind/named.conf.options
```

```
GNU nano 2.2.6      File: /etc/bind/named.conf.options

options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
```

Kdaj končamo konfiguracijo, restartiramo DNS strežnika da se vse spremembe shranijo, z ukazom:

```
sudo service bind9 restart
```

Kdaj to končamo uporabimo ukaz **dig**, da lahko preverimo če naš DNS strežnik dela kot caching nameserver. Preverimo tudi za zunanje domene, da vidimo če si shrani DNS poizvedbe in potem dela lokalno in hitreje kot prvič.

Primary master konfiguracija je prav tako kot upravljanje DNS zapise za določeno domeno lokalno. Če gremo konfigurirat našega DNS strežnika kot primary master, najprej moremo narediti dve cone: forward in reverse cona. Forward cona omogoča da DNS razreši ime domena v IP naslova. Reverse cona pa dela obratno.

Najprej naredimo forward zone datoteka z ukazom:

```
sudo cp /etc/bind/db.local /etc/bind/db.simona23.com
```

Potem odpremo datoteko **db.simona23.com** z ukazom in jo uredimo:

```
sudo nano /etc/bind/db.simona23.com
```

```

GNU nano 2.2.6          File: /etc/bind/db.simona23.com
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      simona.simona23.com. root.simona.simona23.com. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS       simona.simona23.com.
@         IN      A        192.168.1.107
@         IN      AAAA     ::1
simona    IN      A        192.168.1.107
www       IN      A        192.168.1.1

```

Potem naredimo reverse zone datoteka z ukazom:

```
sudo cp /etc/bind/db.127 /etc/bind/db.192
```

Kdaj to naredimo, odpremo datoteko **db.192** z ukazom in jo uredimo:

```
sudo nano /etc/bind/db.192
```

```

GNU nano 2.2.6          File: /etc/bind/db.192
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS       simona.
107       IN      PTR      simona.simona23.com.
1         IN      PTR      www.simona23.com.

```

Na konec gremo še odpret **named.conf.local** konfiguracijsko datoteko, da notri dodamo forward in reverse zone datoteke in to naredimo z ukazom:

```
sudo nano /etc/bind/named.conf.local
```

```
GNU nano 2.2.6      File: /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "simona23.com" {
    type master;
    file "/etc/bind/db.simona23.com";
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
};
```

Kdaj končamo konfiguracijo, restartiramo DNS strežnika da se vse spremembe shranijo, z ukazom:

```
sudo service bind9 restart
```

Kdaj to končamo lahko uporabimo ukaz **dig**, **nslookup** ali **ping**, da lahko preverimo če naš DNS strežnik dela kot primary master. To preverimo tako da poskušamo razrešit nekatero domeno, da dobimo kot rezultat IP naslova.

Pa še moremo odpret konfiguracyjsko datoteko **resolv.conf** in spremenimo IP naslov nameserver-ja v 127.0.0.1 ali v IP naslova našega strežnika. To datoteko odpremo z ukazom:

```
sudo nano /etc/resolv.conf
```

```
GNU nano 2.2.6      File: /etc/resolv.conf

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
#nameserver 127.0.1.1
search simona23.com
nameserver 192.168.1.107
```

**2. SOA** - zapis ali **start of authority record** določi strežnik DNS, ki zagotavlja določene informacije o internetni domeni, elektronski pošti domenskega administratorja, domensko serijsko številko, in različne številce, ki osvežujejo področje.

**PTR** - zapis ali **pointer record** preslika naslov IPv4 v kanonično ime (vzdevek) (canonical name) za tega gostitelja.

**A** - zapis ali **address record** preslika ime gostitelja v 32-bitni naslov IPv4.

**MX** - zapis ali **mail exchange record** preslika domensko ime v spisek **mail exchange** strežnikov za to domeno.

**CNAME** - zapis ali **canonical name record** naredi alias domene. Domena s privzetim imenom (aliased domain) dobi vse poddomene in zapise DNS originalne domene.

**3.** Vsak od odziv oz. odgovor na DNS poizvedbo (query), ki izvira iz DNS strežnika in ima popolno kopijo datoteke iz cone, pravimo da je to "avtorizirani odgovor". Edini problem je da DNS strežniki predpomnijo (cache) odgovore, ki jih prejemajo. Če DNS strežnik ima SOA zapis, pri odgovoru izpolni en del, tako da tisti del signalizira, da je poizvedovan strežnik avtoriziran za tisto domeno in da je tudi odgovor avtoriziran. Vsak DNS strežnik zunaj te domene si bo shranil (cache) tisti avtorizirani odgovor ki ga je uspel pridobiti. Naslednjič, ko bo strežnik spet poizvedovan, nam pove da je odgovor avtoriziran, čeprav ni veljaven za to domeno. Razlika med avtorizirani in neavtorizirani odgovori ki prihajajo iz DNS strežnikov, je da neavtorizirani odgovori nimajo celotno kopijo datoteke iz cone. Tisti odgovori si shranijo odgovor za določen gostitelj, ampak podatke dobijo od strežnika ki ni avtoriziran za domeno.

**4.** Ime mojega lokalnega strežnika DNS je simona.simona23.com, pa ime moja domeno je simona23.com, katero dobimo če uporabimo ukaz:

```
hostname -f
```

Razreševanje (resolving) je pretvorba imena domen v IP naslove. Razreševanje imena `www.google.com`, za katerega predpostavimo da ni v lokalni domeni, poteka tako da prvič odpremo datoteko **resolv.conf**, ki se nahaja v direktorij **/etc** in v katero so zapisani IP naslove imenskih strežnikov oz. DNS name resolvers. To naredimo z ukazom:

```
sudo nano /etc/resolv.conf
```

in potem notri dodamo javni DNS strežniki oz. DNS name resolvers, ki jih Google uporablja. To so 8.8.8.8 in 8.8.4.4. To naredimo, tako da dodamo še dve vrstici v **resolv.conf** datoteko:

```
nameserver 8.8.8.8
```

```
nameserver 8.8.4.4
```

**5.** SPF zapisi so en preprost seznam potrjenih virov, strežnikov, ki so dovoljeni za pošiljanje e-pošte iz naše domene. Zapis SPF za nadzor spam-a v naši lokalni domeni naredimo z naslednjim ukazom:

```
ime_domena(domain.com.) IN TXT "v=spf1 a mx ~all"
```

`ime_domena(domain.com.)` - domeno, za katero velja SPF zapisa.

`IN TXT` - tip zapisa DNS con. SPF zapisi so zapisani kot TXT zapisi.

`v=spf1` - prikaže TXT zapis kot SPF.



a - prikaže primarni A zapis domeni kot odobritev za pošiljanje e-pošto.

mx - prikaže MX zapis(i) domeni kot odobritev za pošiljanje e-pošto.

~all - pomeni, da je ta seznam all inclusive in da ni nobenih drugih strežnikov je dovoljeno da pošiljajo e-pošte po SPF.

## **Zaključek**

Z opravljenim delom v obeh sprintih smo zadovoljni, saj smo si z opravljanjem nalog zagotovili delujoče okolje za delo v projektu semantični spletni portal.

Delujoč LAMP strežnik je osnova za delo na projektu, saj brez le-tega ne bi mogli med seboj povezati izdelkov pri drugih predmetih v funkcionalen spletni portal.

## 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](https://www.debian-administration.org/article/412/Hosting_multiple_websites_with_Apache2)

[http://www.linfo.org/etc\\_skel.html](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>

<https://www.linode.com/docs/security/backups/using-rdiff-backup-with-sshfs/>

<https://www.digitalocean.com/community/tutorials/how-to-backup-mysql-databases-on-an-ubuntu-vps>

<https://wiki.archlinux.org/index.php/Sshfs>

[http://www.tutorialspoint.com/svn/svn\\_checkout\\_process.htm](http://www.tutorialspoint.com/svn/svn_checkout_process.htm)

<http://www.yolinux.com/TUTORIALS/Sendmail.html>

<http://www.inetdaemon.com/tutorials/internet/dns/servers/authoritative.shtml>

<http://www.cyberciti.biz/faq/linux-unix-find-out-dns-server-ip-address-names/>

<http://www.thegeekstuff.com/2014/01/install-dns-server/>

<http://www.cyberciti.biz/tips/linux-how-to-setup-as-dns-client.html>

<http://ubuntuforums.org/showthread.php?t=236093>

<https://help.ubuntu.com/lts/serverguide/dns-configuration.html>

<http://www.liquidweb.com/kb/what-is-an-spf-record/>

<http://sharadchhetri.com/2013/07/23/set-hostname-and-fqdn-in-ubuntu-without-reboot/>

<http://www.krizna.com/ubuntu/configure-dns-server-ubuntu-14-04/>

<http://sl.wikipedia.org/wiki/DNS>

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-postfix-as-a-send-only-smtp-server-on-ubuntu-14-04>

<https://rudd-o.com/linux-and-free-software/setting-up-a-mail-server-using-postfix-in-5-minutes>

<https://www.digitalocean.com/community/tutorials/how-to-add-swap-on-ubuntu-14-04>

<https://help.ubuntu.com/community/NetworkPrintingWithUbuntu>

<https://www.linux.com/learn/tutorials/774476-how-to-manage-printers-in-linux>

<http://www.thegeekstuff.com/2010/07/logrotate-examples/>

<https://help.ubuntu.com/lts/serverguide/cups.html>

<http://searchenterpriselinux.techtarget.com/definition/Heartbeat>

[https://www.howtoforge.com/high\\_availability\\_heartbeat\\_centos](https://www.howtoforge.com/high_availability_heartbeat_centos)

<http://www.linuxjournal.com/article/9838>

<http://www.sj-vs.net/creating-a-simple-debian-deb-package-based-on-a-directory-structure>

<http://www.cioby.ro/linux/scheduling-tasks-using-crontab-in-linux.html>

<http://linuxconfig.org/password-cracking-with-john-the-ripper-on-linux>

<http://www.leaseweblabs.com/2013/06/creating-custom-debian-packages/>