

# Laboratorijska vježba 2

29.03.2017.

Zadatke je potrebno napraviti proizvoljno na vlastitom računalu, virtualnom stroju ili serveru. Preporučamo distribucije bazirane na Debianu koje koriste systemd.

## 1. Zadatak - mreža

Saznali ste da će zbog radova na žičnoj mreži uskoro prestati raditi DHCP server. Kako biste izbjegli potencijalni gubitak veze s internetom u to vrijeme, kolega s faksa vam je sugerirao da postavite **statičku IP adresu** svom mrežnom uređaju. Ne želite uzrokovati probleme drugim korisnicima mreže pa ste mu odlučili dodijeliti istu IP adresu koja vam je trenutno automatski dodijeljena od DHCP servera pri spajanju na mrežu. Također, za svaki slučaj odlučujete koristiti Google javni DNS server dostupan na adresi **8.8.8.8** i **8.8.4.4**

### Potrebno je:

- konfigurirati */etc/network/interfaces* tako da mrežni uređaj koristi statičku IP adresu
- eksplicitno osim IP adrese navesti: *subnet* masku, adresu *gateway* uređaja te adrese primarnog i sekundarnog DNS servera
- kopirati konfiguracijsku datoteku u svoj home direktorij za laganu demonstraciju

## 2. Zadatak - systemd

Vaš je kolega Deen razvio vrlo impresivnu web aplikaciju čiji je izvorni kod dostupan na: <https://pastebin.com/KwGiKByi>.

Ona sluša na portu 1337 i ovisno o tome što upišete u URL, aplikacija odgovara.

Na primjer, ako aplikaciji pristupite uz URL `http://localhost:1337/hello-world`, ona odgovori odgovarajućom porukom. Aplikacija također ispisuje odgovarajuću informaciju na *stdout*.

Iako vrstan developer, kolega je tek počeo proučavati Linux pa traži vašu pomoć. Do sada je aplikaciju pokretao iz naredbenog retka pomoću *python webserver.py*, ali ne zna kako da drži svoju aplikaciju uključenu izvan terminala, kako da se aplikacija pali pri pokretanju računala.

Sram ga je priznati ali ima i bug koji ne zna popraviti. Ako koji korisnik slučajno posjeti */crash*, aplikacija mu se sruši!

Htio bi da se privremeno, dok ne popravi bug, aplikacija automatski ponovno pokreće u slučaju *crasha*. Kolega je čuo da je **systemd** dobar alat za ovaj posao i traži vašu pomoć.

Razmišljajući veselo na putu do vas o svojoj aplikaciji, sjetio se da je prije mogao pratiti ponašanje korisnika u svojoj aplikaciji direktno iz terminala. Zabrinulo ga je to na trenutak, ali se sjetio da systemd ima ugrađen **logging sustav - journald**. Shodno tome, zamolio vas je da se pobrinite i za to da može kroz **journalctl** promatrati uporabu svoje aplikacije.

### Potrebno je:

- napraviti systemd unit (service) file koji pokreće danu python skriptu
- pobrinuti se da se aplikacija automatski restarta u slučaju *crasha*
- postaviti unit tako da se pali pri pokretanju računala
- osigurati da se servis diže, i to **nakon** što je dignuta mreža
- pratiti korištenje aplikacije (logove) kroz journalctl

### 3. Zadatak - nginx

Ambiciozni je kolega iz prošlog zadatka odlučio podići svoj sustav na novu razinu dodajući u njega **nginx**. Naime, neki su mu korisnici javili da im se ne sviđa to što moraju upisivati port 1337 na kraju adrese kako bi pristupili aplikaciji. Kolega zna da browseri implicitno koriste **port 80** i odlučio je prebaciti svoju aplikaciju baš na taj port! Ipak, nije sve tako lako. Uz spomenutu aplikaciju, kolega želi kroz browser posluživati i neke datoteke, također na portu 80.

Ne sviđa mu se ideja dodavanja logike za **posluživanje datoteka** u svoju aplikaciju jer to nije njena poanta. Dodatno, kolegica Joanna mu je na kavi rekla da python nije baš najbrži jezik i da bi takvo posluživanje datoteka moglo biti sporo.

Zato je kolega odlučio dodati fleksibilnost i brzinu u svoj sustav u obliku **nginx web servera**. I, naravno, za to je tražio vašu pomoć.

Zamislio je to ovako: na njegovoj će adresi na portu 80 slušati nginx, koji će se ovisno o putanji *http* zahtjeva ponašati različito.

- Ako putanja počinje s */app*, nginx će proslijediti zahtjev aplikaciji koju ste mu pomogli pretvoriti u servis u drugom zadatku na portu 1337.
- Ako putanja počinje s */public*, nginx će sâm poslužiti datoteke iz datotečnog sustava na lokaciji */home/USERNAME/public/*.  
Primjerice, ako je napravljen zahtjev *http://localhost/public/slika.jpg*, nginx treba (ako datoteka *slika.jpg* postoji) odgovoriti datotekom */home/USERNAME/public/slika.jpg*

Oduševljen zamišljenom fleksibilnošću, kolega se odlučuje malo zabaviti i podesiti nginx tako da na */kset* odgovara privremenim **http redirectom** na *https://www.kset.org*.

Kolega je pedantan i nije zaboravio na **važnost logova**, pa planira informacije o posjetima spremati na lokaciju */home/USERNAME/public/nginx-access.log* jer je ljubitelj transparentnog poslovanja, a informacije o greškama iz sigurnosnih razloga na */var/log/nginx/nginx-error.log*. Pomozite mu i steknite njegovu zahvalnost!

#### Napomene:

- treba vam samo jedan nginx config file, recimo: */etc/nginx/sites-available/localhost*
- */etc/nginx/sites-enabled/localhost* je obično symlink na */etc/nginx/sites-available/localhost*
- nakon promjene konfiguracijskih datoteka, potrebno ih je ponovno učitati pomoću *nginx -s reload*
- treba vam samo jedan nginx server block
- pripazite na ime servera u konfiguraciji - inače se koristi domena koja se poslužuje, vama sada to zamjenjuje "localhost"
- privremeni *http redirect* ima kôd 302
- u praksi je gotovo uvijek sigurnosni propust javno izlagati logove - obično se i access log i error log drže u */var/log*
- **BONUS ZADATAK:** konfigurirati nginx da sve poslužuje preko *https* protokola, uz **self-signed** certifikat i redirect sa *http* na *https*