Linux videot: <https://www.youtube.com/playlist?list=PLZw_fXxI6L50ovlxunMvWw-bEDc3TiBQp>

Contents

[TLDR 1](#_Toc180563972)

[Alkuun 1](#_Toc180563973)

[Asenna bind9 2](#_Toc180563974)

[Luodaan primary zone 2](#_Toc180563975)

[Luodaan tietokanta josta zone hakee tiedot 2](#_Toc180563976)

[Tietokannan muokkaaminen 2](#_Toc180563977)

[Konfiguraation testaaminen 4](#_Toc180563978)

[DNS ja nettisivuston testaaminen 4](#_Toc180563979)

[Käänteisten pyyntöjen konfigurointi (Reverse Lookup zone) 5](#_Toc180563980)

[Reverse Lookup Zonen testaaminen 7](#_Toc180563981)

[Forwarders 8](#_Toc180563982)

[Forwarders ongelma (harvoin) 8](#_Toc180563983)

[Master ja Slave (bind9) 9](#_Toc180563984)

[Haproxy (bind9 + apache2 + haproxy) 11](#_Toc180563985)

[Haproxy HTTPS 14](#_Toc180563986)

[Haproxy ongelmia 15](#_Toc180563987)

# TLDR

1. Asenna paketti bind9
2. Selvitä mikä on haluamasi domain nimi
3. Luo domainille zone named.conf.local tiedostoon
4. Luo tietokanta kyseiselle zonelle esim. db.taitaja2017.fi
5. Tee tarvittavat asetukset tietokantaan kuten määritä domain nimi vastaamaan haluamaasi IP-osoitteeseen
6. Muista vaihtaa Serial kohtaa aina kun teet muutoksia!
7. Käynnistä palvelu uudestaan
8. Testaa toiminta esim. avaamalla web-sivu nimellä selaimessa
9. Tarvittaessa luo reverse zone (käänteinen)

DNS on palvelu, jolla saadaan ohjattua nimet, kuten www.yle.fi johonkin IP-osoitteeseen kuten 52.85.47.74. DNS avulla Internetin käyttö on helpompaa. Ilman sitä käyttäjät joutuisivat kirjoittamaan selaimiin IP-osoitteita nimien sijaan.

# Alkuun

* Tehdään DNS palvelin Debian palvelimeen, jonka IP-osoite on 172.22.1.69
* Nimeksi halutaan taitaja2017.fi
* Eli kun käyttäjä ottaa selaimella yhteyden osoitteeseen taitaja2017.fi se on sama kuin jos ottaisi yhteyden osoitteeseen 172.22.1.69.
* Ennen kuin menet eteenpäin varmista, että tietokoneesi IP-osoite on 172.22.1.69.

# Asenna bind9

**apt install bind9**

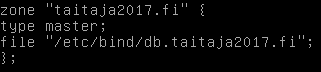
Asenna myös apache2, tämä helpottaa DNS testaamista.

**apt install apache2**

# Luodaan primary zone

Zone luodaan muokkaamalla /etc/bind/named.conf.local tiedostoa

**nano /etc/bind/named.conf.local**



Lisää sinne kuvan mukainen teksti.

Siinä määritellään että zonen nimi on taitaja2017.fi.

Se on tyypiltään Primary Master

Ja sen tiedot löytyvät polusta /etc/bind/db.taitaja2017.fi Tämä polku luodaan seuraavassa kohdassa.

# Luodaan tietokanta josta zone hakee tiedot

/etc/bind/ kansiosta löytyy valmiiksi tietokantoja joten voimme käyttää niitä pohjana meidän db.taitaja2017.fi tietokannalle.

Kansiosta löytyy db.local niminen tietokanta ja sitä käytämme pohjana. Joten otetaan siitä kopio.

**cp /etc/bind/db.local /etc/bind/db.taitaja2017.fi**

Eli kopioidaan db.local tietokanta ja annetaan kopion nimeksi db.taitaja2017.fi.

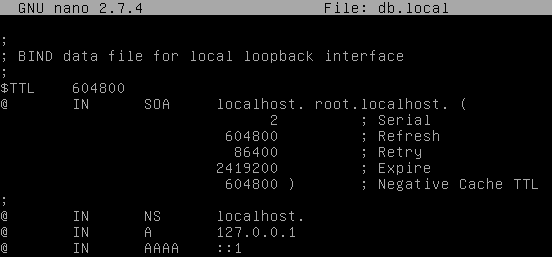
Yllä olevassa kuvassa, jossa loimme primary zonen viitattiin nimenomaan tähän tietokantaan eli db.taitaja2017.fi. **Näiden kahden tiedon tulee vastata toisiaan!!!**



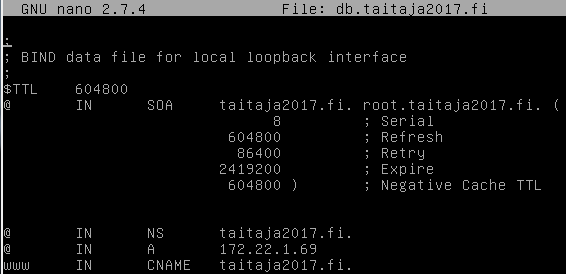
/etc/bind/ kansiosta tulee siis löytyä kuvan mukainen db.taitaja2017.fi tietokanta.

# Tietokannan muokkaaminen

db.local tietokanta näyttää tältä ja se toimii meidän pohjanamme.



Jotta osoite taitaja2017.fi toimisi tulee muokata db.taitaja2017.fi tietokantaa ja tehdä siitä tällainen:



Kaikki kohdat joissa luki localhost, korvataan haluamallamme domain nimellä, tässä tapauksessa taitaja2017.fi.

**@ IN NS taitaja2017.fi.**

Tällä rivillä määritetään että taitaja2017.fi on NS, NameServer.

**@ IN A 172.22.1.69**

Tällä rivillä määritetään että 172.22.1.69 on A tyypin recordi.

Näiden kahden yhdistelmällä määritetään että osoite 172.22.1.69 = taitaja2017.fi

**www IN CNAME taitaja2017.fi.**

Tällä rivillä määritetään www CNAME tyypin recordi (alias) eli jos käyttäjä kirjoittaa osoiteriville www.taitaja2017.fi se vie samaan paikkaan kuin taitaja2017.fi. Tämä ei ole välttämätön toisin kuin A tyypin recordi. Lue tehtävä aina huolella mitä halutaan.

**HUOM! AINA, aina kun teet muutoksia näihin asetuksiin joudut vaihtamaan Serial kohdan numeroa.**

# Konfiguraation testaaminen

Voit testata konfiguraation named-checkconf ja named-checkzone komennoilla. Bind9 zone konfiguraation testaat näin:

**named-checkconf /etc/bind/named.conf.local**

Zonelle luodut tietueet testaat näin:

**named-checkzone taitaja2017.fi /etc/bind/db.taitaja2017.fi**

Komento siis testaa että taitaja2017.fi domainille löytyy tarvittavat tiedot /etc/bind/db.taitaja2017.fi nimisestä tietokannasta.

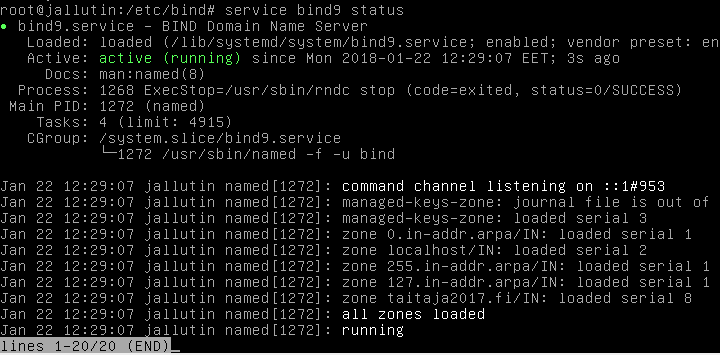
# DNS ja nettisivuston testaaminen

Kun olet tehnyt yllä olevat konfiguroinnit on aika käynnistä bind9 uudestaan:

**systemctl restart bind9**

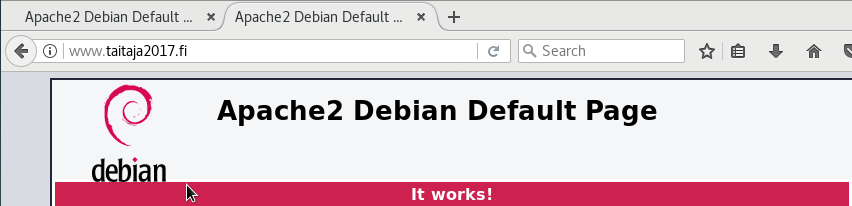
Katso että kaikki on kunnossa:

**systemctl status bind9**



Testaa että taitaja2017.fi ja www.taitaja2017.fi osoitteet toimivat jollakin toisella koneella, esim linux työasemalla.

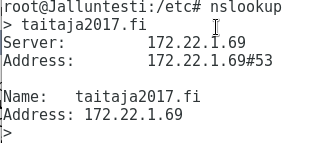




nslookup ei ole Debianissa oletuksena. Saat sen asennettua seuraavalla komennolla:

**apt install dnsutils**

Myös nslookup komennolla voit testata, että DNS toimii.



Testaaminen onnistuu myös Linuxin dig työkalulla.

# Käänteisten pyyntöjen konfigurointi (Reverse Lookup zone)

Tehnyt: Aleksi Sievälä

**Reverse Lookup Zonea ei välttämättä aina tarvitse konfiguroida. Nimenselvennys esim. nettisivua varten toimii ilman sitäkin. Lue tehtävänanto aina huolella, että vaaditaanko tätä ominaisuutta.**

**HUOM! Tämä alla oleva esimerkki ei ole kytkettynä yllä tehtyyn Forward Lookup zonen luontiin (172.22.1.69). Perusidea on kuitenkin sama ja joudut lähinnä muokkaamaan IP-osoitteita ja nimiä, jotta saisit sen toimimaan.**

Seuraavaksi konfiguroidaan DNS vastaamaan myös käänteisiin pyyntöihin. Eli jos selain kysyy osoitetta 10.3.57.103 se vastaa osoitteella taitaja2016.local.

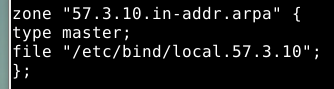
Lisätään **/etc/bind/named.conf.local** tiedostoon rivit:

**zone “57.3.10.in-addr.arpa” {**

**type master;**

**file “/etc/bind/local.57.3.10”;**

**};**

****

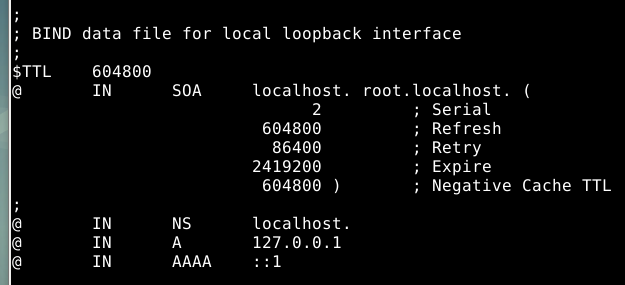
Käänteisen zonen osoite tulee netin sisäisestä osoitteesta, esim. tässä käytetty **10.3.57.103**. Kolme ensimmäistä lukua (10.3.57) käännetään ja host numero (103) jätetään pois myöhempään käyttöön (alla olevan kohdan ptr tietue).

Luodaan tekstitiedosto **/etc/bind/local.57.3.10**,pohjana käytetään **/etc/bind/db.local** tiedostoa:

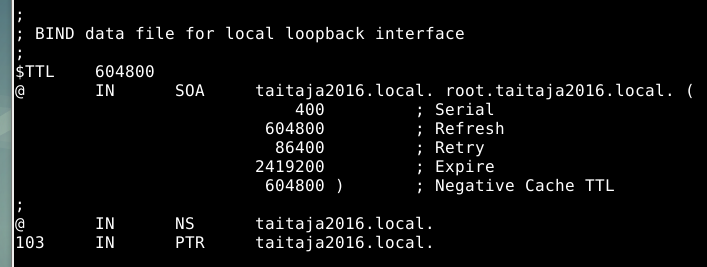
**sudo cp /etc/bind/db.local /etc/bind/local.57.3.10**

Muokataan **/etc/bind/local.57.3.10** tiedosto oman tarpeen mukaisesti:

**Alkuperäinen tiedosto:**

****

**Muokattu tiedosto:**

****

Sama kuin Forward Lookup Zonessa. Kaikkiin localhosti kohtiin laitettiin haluttu domain nimi, tässä tapauksessa taitaja2016.local.

**@ IN NS taitaja2016.local.**

Tällä rivillä määritetään että taitaja2016.local on name server eli DNS.

**103 IN PTR taitaja2016.local.**

Tällä rivillä määritetään, että käänteinen selvitys tehdään IP-osoitteelle joka loppuu numeroon 103. (Osoitealue tässä esimerkissä on 10.3.57.0/24, ja tästä verkosta laite jonka IP-osoite on 10.3.57.103)

**Muista päivittää Serial numero aina tiedoston muokkauksen jälkeen!**

Viimeiseksi vielä käynnistetään bind uudelleen:

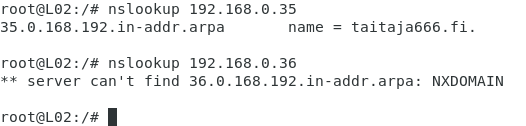
**sudo service bind9 restart**

**Jos apachen kanssa käytetään virtualhostia, sama reverse zone käy kaikille nettisivuille johtuen samasta ip-osoitteesta.**

# Reverse Lookup Zonen testaaminen

Reverse Lookup Zonen toiminta ei näy normaalissa käytössä, joten sen testaaminenkin on erilaista. Reverse Lookup Zone on tärkeässä roolissa mm. sähköpostin toiminnassa.

Reverse Lookup Zonen saa testattua nslookup komennolla. nslookup ”testattava IP”, kuten alla on testattu 192.168.0.35 palvelimella sijaitseva Reverse Lookup Zone. Osoitteessa 192.168.0.36 ei ole Reverse Lookup Zonea, joten tulee virheilmoitus että sitä ei voi löytää.



# Forwarders

Forwarders on eri asia kuin Forward Lookup Zone. Forwarders avulla oma DNS-palvelin voi ohjata DNS-kyselyt, joihin se ei osaa itse vastata jollekin toiselle DNS-palvelimelle. Usein kyselyt ohjataan julkisille DNS-palvelimille, kuten Googlen tai internet palveluntarjoajan DNS-palvelimille.

Forwarders konfigurointi bind9 on helppoa.

Avaa tiedosto **/etc/bind9/named.conf.options**

Lisää sinne kohta:

**forwarders {**

**8.8.8.8;**

**};**

Kuten kuvassa on tehty. Ota huomioon, että options { kohta tulee myös olla suljettu }; merkeillä (kuvan viimeinen rivi)

A computer screen with white text

Description automatically generated

## Forwarders ongelma (harvoin)

Jos tulee ongelmia Forwarders kanssa niin tämmöiset säädöt saavat sen toimimaan:

A computer screen with white text

Description automatically generated

Ongelmaa voi tulla vastaan, jos työasemat ovat eri aliverkossa, kuin bind9 palvelin.

# Master ja Slave (bind9)

DNS Master ja Slave konfiguraatiossa on ideana se, että DNS tiedot ovat useammilla palvelimilla. Jos toinen palvelimista kaatuu niin palvelut toimivat edelleen. Master palvelimelle luodaan zonet ja tietueet ja nämä ”kopioidaan” Slave palvelimelle bind9 toimesta.

Tässä Master palvelimena toimii IP-osoite 10.134.24.10 ja Slave palvelimena toimii IP-osoite 10.134.24.11

Konfiguroidaan ensin Master palvelin. Tässä esimerkissä käytössä on IP-osoite alue 10.134.24.0 ja domain nimi on taitaja9hyria.fi

Luo named.conf.local tiedostoon alla olevat zonet. Poikkeuksena aikaisempiin on allow-transfer kohta, joka mahdollistaa sen, että Master palvelimelta voi kopioida tietoja Slavelle. Tässä on konfiguroitu molemmat forward ja reverse lookup zonet.

A screenshot of a computer program

Description automatically generated

Seuraavaksi luo db. tiedostot /etc/bind kansion alle. Käytä kopiota db.local tiedostosta, kuten ennenkin.

db.taitaja9azure.fi tietueet alla. Huomio palvelinten nimet ja IP-osoitteet.

A screenshot of a computer screen

Description automatically generated

Ja db.local.24.134.10 tietueet:

A screenshot of a computer

Description automatically generated

Muutosten jälkeen käynnistä bind9 uudestaan.

Seuraavaksi konfiguroidaan Slave palvelin. Slave palvelimelle ei luoda lainkaan db tiedostoja vaan pelkästään konfguroidaan named.conf.local tiedosto seuraavasti.

Eli erona Master palvelimeen on se, että tyyppi on slave, file kohtaan täytyy määrittää polku /var/cache/bind hakemiston alta ja masters kohtaan määritetään Master-palvelimen IP-osoite. Huomioi, että Slaven file polku tulee olla /var/cache/bind, muuten zonen tiedot eivät siirry Masterilta Slavelle.

A screenshot of a computer program

Description automatically generated

Muutosten jälkeen käynnistä bind9 uudestaan.

Voit tarkistaa onnistuiko zone transfer (eli Master ja Slave välinen toimivuus) kun katsot Slave-palvelimelle bind9 statuksen. Tiedoista näkee kuinka Slave haki zone tiedot Masterilta (Transfer status: success). Lisäksi status kertoo millä serial numerolla muutokset haettiin.

A screen shot of a computer

Description automatically generated

**Master ja Slave mallissa on erityisen tärkeää, että serial kohtaa muutetaan aina, kun tehdään muutoksia,** kuten aikaisemmin tässä ohjeessa on mainittukin. Se kuitenkin korostuu tässä tapauksessa.

Master ja Slave toimintaa voit myös hiukan tarkastella katsomalla tulevatko db. tiedostot Masterilta Slavelle määritettyyn hakemistoon, joka on tässä esimerkissä /var/cache/bind. Sen alta löytyy kuvassa samat db tiedostot kuin Master-palvelimen /etc/bind kansiosta. Voit mm. seurata ls komennolla tiedostojen muokkaus/luonti aikoja ja varmistaa, että ne ovat päivittyneet.

A screenshot of a computer

Description automatically generated

# Haproxy (bind9 + apache2 + haproxy)

Haproxy on paketti, jonka avulla voi tasapainottaa kuormaa (high availability = ha) erilaisille verkon palveluille, kuten WWW-palveluille eli apachelle. Sellainen tehdään tässä. Frontend palvelimelle tulee liikennettä ja se ohjataan jommallekummalle backend palvelimista. Lisätietoja haproxysta: <https://upcloud.com/resources/tutorials/haproxy-load-balancer-debian>

Tässä esimerkissä on käytössä 3 palvelinta:

bck01 = 10.134.24.10, tällä palvelimella on bind9 ja apache2

bck02 = 10.134.24.11, tällä palvelimella on bind9 ja apache2

frnt = 10.134.24.15, tällä palvelimella on haproxy

Ensimmäiseksi luo Master-palvelimelle uusi A tietue, joka osoittaa frnt palvelimen IP-osoitteeseen.

A screenshot of a computer

Description automatically generated

**Muuta serial numeroa** ja käynnistä bind9 uudestaan.

Sen jälkeen lataa molemmille backend palvelimille apache2 paketti. Muuta sen oletussivua palvelimilla niin, että molempien sisältö on **ERI**, jotta testaaminen on helpompaa.

Muokkaa molemmilla backend palvelimilla /etc/apache2/sites-available/000-default.conf tiedostoa ja määritä ServerName kohtaan sama domain nimi, jonka määritettiin bind9 puolelle (tässä tapauksessa intra.taitaja9azure.fi.

A screenshot of a computer screen

Description automatically generated

Sitten asenna haproxy Frontend palvelimelle

**apt install haproxy**

Muokkaa /etc/haproxy/haproxy.cfg tiedostoa ja lisää sinne kuvan mukaiset frontend ja backend kohdat (älä koske aikaisempiin konfiguraatioihin, joita on esimerkiksi defaults kohdan alla).

Konfiguraatioon lisätään frontend kohta, joka määrittää frontend palvelimen eli sen IP-osoitteen, portin ja tilan (mode). default\_backend kohdassa annetaan ikään kuin nimi sille backendille, jota frontend käyttää. Näet kuvasta, että frontend alle on määritetty sen käyttävän backendiä nimeltään apache\_servers, joka määritetään seuraavaksi.

backend kohdassa määritään taas tila (mode), balance eli miten kuormantasausta tehdään (tässä helposti testattava random) ja määritetään kaikki ne backend serverit, joita käytetään (tässä tapauksessa bck01 ja bck02 palvelimet ja niiden IP-osoitteet sekä portit.

A screenshot of a computer

Description automatically generated

Käynnistä haproxy uudestaan

**systemctl restart haproxy**

Siirry jollekin koneelle, jolla on selain ja testaa toiminta päivittämällä sivua intra.taitaja9azure.fi. Sen sisällön tulisi muuttua välillä, sillä haproxy randomisti valitsee kumpaan backend palvelimeen frontend palvelin välittää liikenteen.

Tässä esimerkkejä:

bck01:

A screenshot of a computer

Description automatically generated

bck02:

A screenshot of a computer

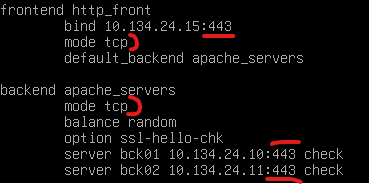
Description automatically generated

## Haproxy HTTPS

Jos haluat haproxy toimimaan HTTPS apache2 sivuilla ota HTTPS käyttöön normaalisti apache2 puolelle (katso apache2 ohjeista kohta **SSL Sertifikaatin luominen**).

Tässä esimerkissä tehdään ns. haproxy with SSL pass-through, eli backend palvelimet huolehtivat HTTPS osuudesta eikä haproxy itse. Lisätietoja täältä: <https://serversforhackers.com/c/using-ssl-certificates-with-haproxy>

Haproxyyn tee seuraavat muutokset. Eli kuunneltavat portit vaihdetaan 443 porttiin. Mode muutetaan tcp.



Käynnistä haproxy uudestaan.

Tämän jälkeen testaa toimintaa työaseman selaimella. Toiminta kannattaa testata yksityisessä ikkunassa/välilehdessä. Kun sivun avaa tulee tällainen näkymä, etkä vielä tiedä kummalle backend palvelimelle olet ohjautumassa. Hyväksy riskit, koska käytät itseluotua SSL-sertifikaattia.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

HTTPS testatessa et voi enää päivittää sivua, ja sisältö muuttuisi vaan joudut jokaisella testillä sulkemaan yksityisen ikkunan ja avaamaan uuden sekä hyväksymään riskit. Sitten vasta näet pääsetkö sillä kerralla toiselle backend palvelimelle.

A screenshot of a computer

Description automatically generated

## Haproxy ongelmia

Jos haproxy ei tunnu toimivan domain nimillä voit testata sitä myös IP-osoitteilla. Eli jos työaseman selaimella kirjoitat frontend palvelimen IP-osoitteen niin sen tulisi ohjata liikenne jommankumman backend palvelimen apache2 sivuille.