Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Tuomas Korva
Aleksanteri Nieminen

Personoitujen GNU/Linux-jakeluiden automaattinen asennus ja keskitetty käyttäjähallinta

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinöörityö

15.10.2019

Tekijä Otsikko Sivumäärä Aika	Korva Tuomas Nieminen Aleksanteri Personoitujen GNU/Linux-jakeluiden automaattinen asennus ja keskitetty käyttäjähallinta. 34 sivua + 1 liitettä 15.10.2019
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikan tutkinto-ohjelma
Ammatillinen pääaine	IoT and Cloud Computing
Ohjaajat	Lehtori Jukka Louhelainen

Insinöörityössä luotiin järjestelmä automaattiselle ja keskitetylle verkon yli tapahtuvalle GNU/Linux-käyttöjärjestelmän asennukselle sekä konfiguraatiolle. Tarkoituksena oli käyttää omaa ratkaisua, joka soveltaa useampia pieniä ratkaisuja. Päätavoitteena oli tehdä kaikki loppukäyttäjälle mahdollisimman automaattiseksi ja helpoksi, jolloin käyttöönotto ei olisi kiinni loppukäyttäjän GNU/Linux-taidoista.

Insinöörityössä pyrittiin käyttämään mahdollisimman paljon vapaan lähdekoodin ohjelmistoja. Nykyisessä tilassaan insinöörityö on käytettävissä ja helposti laajennettavissa ohjelmistojen ja verkkoresurssien osalta.

Kehitystyössämme käytimme omaa kokemustamme sekä kykyämme hakea ja soveltaa tietoa. Eteneminen työssä tapahtui vaiheittain levykuvan luomisesta jakeluun ja verkkoympäristön sekä teknologioiden käyttöönottoon.

Kehitysmenetelmänämme hyödynsimme ketteriä menetelmiä, jossa jokaisella osa-alueella oli oma prioriteettinsa. Pyrimme etenemään näiden prioriteettien mukaan loogisesti kohti nykyistä ratkaisua. Suunnittelulistallemme jäi vielä lukuisia osa-alueita mahdollista jatkokehitystä varten.

Insinöörityön onnistuminen ylitti omat odotuksemme, ja työn lopputuote on välittömästi käytettävissä maksimissaan 190 asiakaskoneen verkossa. Työ on suunniteltu jatkokehitystä ajatellen, ja se onkin helposti laajennettavissa kattavammaksi.

Avainsanat	Linux, PXE, GNU, Ubuntu, Kickstart

Author Title Number of Pages Date	Aleksanteri Nieminen Tuomas Korva Automatic installation of personalized GNU/Linux distributions and centralized user management. 34 pages + 1 appendices 15 October 2019
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	IoT and Cloud Computing
Instructors	Jukka Louhelainen, Lecturer

In this thesis a system was created for automatic and centralized network installation and configuration of GNU/Linux operating system. The goal was to create a custom solution that consists of several smaller solutions. The product was designed so that everything would be as automated and easy to use for a user without extensive knowledge of GNU/Linux.

The bachelor's thesis used as much free software as possible. The product of the thesis is usable in its current form and very scalable in terms of software and network resources.

We used our own experience as well as our ability to retrieve and apply knowledge in the creation process while also keeping in mind the desired outcome. Work progressed in stages, starting with the creation of the disk image, distribution and implementation of the network environment and technologies.

In development and implementation, agile development methods were emphasized, with each component having its own priority, and according to these priorities, we logically proceeded towards the current service. There are still many areas on our list for further development.

The thesis was a success and is immediately available for up to 190 client computers and is easily scalable. The system works very well as a basis for further development.

Keywords	Linux, PXE, GNU, Ubuntu, Kickstart

Sisällys

Lyhenteet

1 Johdanto 1

2 Vaatimusmäärittely 2

3 Kehitysmenetelmä 3

4 Suunnittelu 4

4.1 Käyttöjärjestelmävalinta 5
4.2 Käyttäjien hallinta. 6

5 Toteutus 7

5.1 Verkko 7
5.2 Palvelin 8
5.3 Levykuva 21
5.4 Loppukäyttäjä 31

6 Testaus 32

6.1 Käytettävyystestaus 32
6.2 Itsearviointi 33

Lähteet 34

Liitteet

Liite 1. Ks.cfg

Lyhenteet

DHCP	Dynamic Host Configuration Protocol eli protokolla, jota käytetään verkon IP-osoitteiden ja muiden verkkoasetusten dynaamiseen jakeluun.
LTS	Long-Term Support eli tuote, jota ylläpidetään normaalia julkaisua pidempään.
Non-Free	Ohjelmisto, joka ei ole vapaa eli ohjelmisto, jonka jakelua ja muokkaamista on rajoitettu.
GNU	Gnu's Not Unix eli Richard Stallmanin aloittama vapaan käyttöjärjestelmän kehitysprojekti.
PXE	*Pre-Boot Execution Environment* eli standardoitu ja yleisimmin käytetty tietokoneen verkkokäynnistystapa. Käynnistyessään asiakaskone yhdistää palvelimelle, josta se noutaa käynnistykseen tarvittavan ohjelmistokokoelman.
IP	Internet Protocol eli pakettikytkentäisessä verkossa käytettävä protokolla, jonka tehtävä on kuljettaa tietoliikennepaketit perille. Internetin toiminnan ydin.
TFTP	Trivial File Transfer Protocol eli tiedostojen siirtoon tarkoitetun verkkoprotokollan yksinkertaistettu versio.
UDP	User Datagram Protocol eli yhteydetön protokolla, joka mahdollistaa pakettien siirtämisen IP-verkossa.
LDAP	Lightweight Directory Access Protocol eli verkkoprotokolla käyttäjäntunnistukseen ja käyttöoikeuksien tarkistamiseen.
RHEL	Red Hat Enterprise Linux eli Red Hat-yrityksen ylläpitämä kaupallinen GNU/Linux-jakelu.

VirtualBox	Käyttöjärjestelmien virtualisointiin tarkoitettu ohjelmisto.
Kickstart	Red Hat-yrityksen kehittämä automaattisten GNU/Linux-käyttöjärjestelmien asennuksen mahdollistava ohjelma.
Sudo	Ohjelmisto, joka mahdollistaa komentojen suorittamisen pääkäyttäjänä.
Grub	Ohjelmisto, joka aloittaa käyttöjärjestelmän lataamisen tietokoneen käynnistyessä.
MBR	Master Boot Record eli muistilaitteella oleva alue, josta BIOS etsii käyttöjärjestelmän latauskoodin.
BIOS	Basic Input-Output System eli ohjelmisto, joka lataa käyttöjärjestelmän keskusmuistiin sekä käynnistää sen tietokoneen käynnistyessä.
Gateway	Yhdyskäytävä, joka yhdistää kaksi verkkoa toisiinsa.
Kerberos	Massachusettsin teknillisen korkeakoulun kehittämä tietoverkko todennusprotokolla.
AAA	Ryhmä protokollia, joita käytetään käyttäjien tunnistamiseen verkoissa. Muodostuu englanninkielen sanoista Authentication, Authorization ja Accounting.
FQDN	Fully qualified domain name, on niin sanottu absoluuttinen verkkotunnus, joka sisältää jokaisen verkkotunnuksen tason.
SSH	Secure Shell eli protokolla, joka on tarkoitettu suojattuun tietoliikenteeseen.
Classful network	Eli ABCDE luokiteltu verkko, jossa verkon osat on jaoteltu niiden koon ja käytön mukaan.
Apache 2	Apache Software Foundationin kehittämä http-palvelinohjelmisto.

1 Johdanto

Elämä ilman vapautta on sortoa, ja se pätee myös nykyisessä IT-alan ympäristössä, jossa yritykset käyttävät paljon suljetun lähdekoodin ‘non-free’-ohjelmistoja. Nämä ohjelmistot eivät anna käyttäjilleen mahdollisuutta käyttää niitä haluamillaan tavoilla, ja niiden todellisesta toiminnasta ei käyttäjä myöskään voi olla koskaan varma. Suljetun lähdekoodin ohjelmistot voivat tehdä epätoivottuja asioita kuten ohjelmiston käyttäjän vakoilua, manipulointia tai sensurointia [1]. Siksi pyrimme työssä käyttämään mahdollisimman paljon vapaita ja avoimen lähdekoodin ohjelmistoja.

Toinen syy käyttää työssä avoimia ohjelmistoja ovat halvat kustannukset verrattuna suljetun lähdekoodin ohjelmistoihin. Monissa suljetun lähdekoodin ohjelmistoissa on hyvin tarkasti määritelty, voiko tiettyä versiota käyttää esimerkiksi kaupallisiin tarkoituksiin. Vapaan lähdekoodin ohjelmistoissa tätä ongelmaa ei ole, vaan loppukäyttäjä saa käyttää, muokata ja jakaa ohjelmistoja melko vapaasti. Kaikissa lisenssiehdoissa määritellään, miten ohjelmistoa saa muokata ja jakaa, mutta vapaissa ohjelmistoissa nämä määritelmät eivät päde. Avointen ohjelmistojen huonona puolena yleensä on niiden vaatima käytön osaaminen sekä tuen saamisen hankaluus. Suljettuihin ja kaupallisiin tuotteisiin sisältyy yleensä käytön tuki, kun taas avoimissa ohjelmistoissa käyttäjä joutuu turvautumaan yhteisön tukeen.

Insinöörityön tavoitteena on suunnitella ja luoda järjestelmä, joka mahdollistaa GNU/Linux-jakeluiden mahdollisimman helpon käyttöönoton ja hallinnan. Toteutamme järjestelmän testiympäristössä, joka koostuu palvelimen, reitittimen sekä asiakaskoneiden muodostamasta lähiverkosta. Hyväkseen asennusten jakelupalvelin käyttää Kickstart-, DHCP- sekä PXE Server -ohjelmistoja, joita tarpeen mukaan räätälöimme omaan ratkaisumme sopiviksi. Lisäksi järjestelmään suunnitellaan keskitetty käyttäjien- ja resurssienhallinta.

Pienille yrityksille on mielestämme liian vähän avoimen lähdekoodin ratkaisuja saatavilla siinä muodossa, että ne olisivat helposti käyttöönotettavissa. Nykyisissä avoimen lähdekoodin ratkaisuissa vaaditaan syvällistä tietoa ja osaamista teknologiasta, vaikka suureen osaan työtehtävistä riittäisivätkin pinnalliset taidot. Yksi motivaation lähteistämme

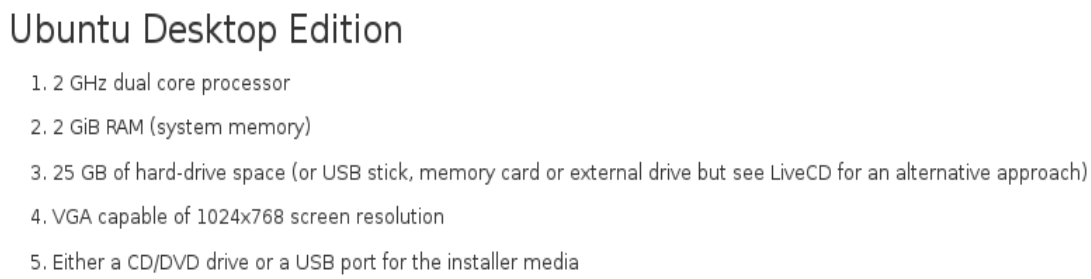
olikin mahdollistaa kustannustehokkaan ja tietoturvallisen järjestelmän helppo käyttöönotto ja ylläpito, jolloin loppukäyttäjä voi keskittyä työtehtäviinsä eikä järjestelmien käyttöönoton kanssa kamppailuun.

2 Vaatimusmäärittely

Tässä luvussa luomme vaatimukset avoimen lähdekoodin työympäristön toteuttamiselle. Otamme huomioon ylläpitäjän sekä loppukäyttäjän näkökulmat. Näiden vaatimusmäärittelyjen avulla mahdollistamme selkeät vaatimukset työympäristön toteuttamiselle kolmesta eri perspektiivistä. Laitteiston kannalta keskitymme työympäristön kunnolliseen toimintaan, ylläpitäjän näkökulmasta työympäristön helppoon konfigurointiin ja loppukäyttäjän perspektiivissä keskitymme enimmäkseen käyttöönoton helppouteen.

Ylläpitäjän näkökulmasta järjestelmän täytyy olla helppo ylläpitää. Ylläpidon vaikeuksia tulemme helpottamaan keskittämällä käyttäjien hallinnan. Järjestelmä toteutetaan myös yksinkertaisia työkaluja käyttäen. Yksinkertaisten työkalujen ansiosta onkin helppo oppia käyttämään järjestelmää tehokkaimmillaan. Pyrimme myös dokumentoimaan mahdollisimman hyvin työkalujen ominaisuuksia ja käyttöä, jotta mahdollisissa ongelmatilanteissa olisi vianetsintä ja korjaaminen helppoa.

Loppukäyttäjän näkökulmasta tärkeintä on käyttöönoton sujuvuus ja helppous. Tarkoituksena on, että loppukäyttäjän työasema ja resurssit ovat heti käytettävissä, eikä käyttäjä tarvitse aikaisempaa syvällistä tietoa Ubuntu-käyttöjärjestelmän toiminnallisuuksista. Tärkeää on, että loppukäyttäjä voi aloittaa työskentelyn mahdollisimman nopeasti ja myöhemmin tutustua tarkemmin käytössä olevien ohjelmistojen sielunelämään. Ensimmäinen askel, joka täyttää tavoitteemme on automaattinen asennus, jolloin käyttäjän ei tarvitse tietää, mitä asetuksia hänen tulee valita asennuksen yhteydessä. Toinen askel on tarvittavien ohjelmien ja asetuksien automaattinen asennus ja määrittely. Näiden osakokonaisuuksien summana loppukäyttäjällä tulisi olla täysin toimintavalmis kone yhteyksineen ja työkaluineen heti käytössä.

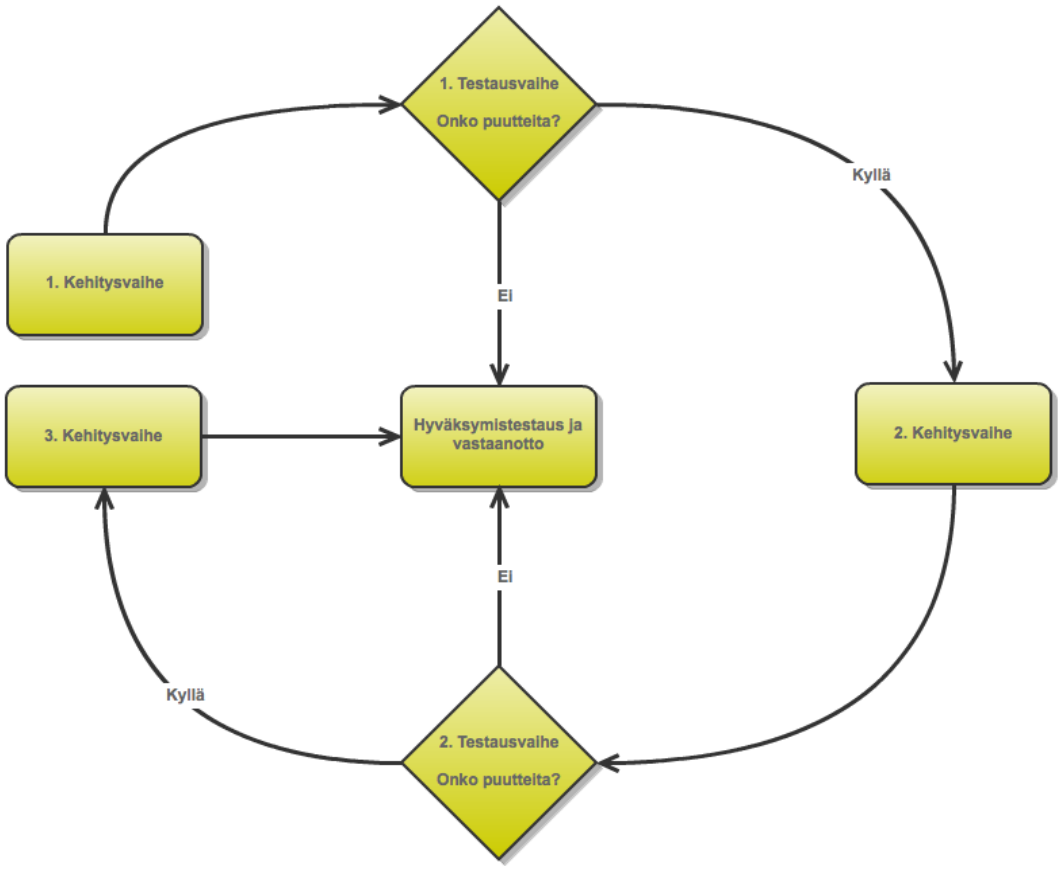


Kuva 1. Ubuntu Desktopin minimivaatimukset [5]

Laitteiston vaatimusmäärittelyssä täytyy ottaa huomioon sekä järjestelmän perustoiminnallisuus että käyttäjäkohtaiset vaatimukset esimerkiksi ohjelmistojen muodossa. Perusperiaatteena on, että laitteiston tulee toimia sujuvasti Ubuntu-käyttöjärjestelmän kanssa C-luokan verkossa [6], mutta laitteiston tulisi pystyä vastaamaan myös erilaisten ohjelmistojen vaatimuksiin. Palvelinkoneella ja loppukäyttäjän koneella vaatimukset ovat hieman erilaiset, mutta teknisesti laitteiden väliset erot ovat pieniä. Tärkeimpänä osa-alueena, joka tulee ottaa huomioon laitteistoa määritellessä, on graafisen suorituskyvyn tarve, joka on loppukäyttäjällä suurempi kuin palvelimessa. Apuna laitteiston valinnassa kannattaakin käyttää käyttöjärjestelmän kehittäjän kotisivuja, joissa on selkeästi tuotu esille paljonko resursseja tulee varata käyttöjärjestelmän perustoiminallisuuksien sujuvaan toimintaan. Esimerkiksi Ubuntun sivuilla onkin nähtävillä Ubuntu-käyttöjärjestelmän minimivaatimukset.

3 Kehitysmenetelmä

Insinöörityössä käytössämme oli iteratiivinen kehitysmalli [3]. Iteratiivinen kehitysmalli, tunnettu myös ketteränä kehitysmenetelmänä, on malli, jossa projekti jaetaan lyhyiksi työjaksoiksi, jotka rytmittävät projektia. Jokaisella iteraatiolla on välitavoitteensa, joka auttaa jakamaan projektia pienemmiksi osiksi. Jokaisen iteraation tarkoituksena on tuottaa hieman edellistä valmiimpi lopputulos. Välitavoitteista ja pienistä kehityskohteista muodostuu kehitysjono.



Kuva 2. Iteratiivinen kehitysmenetelmä [4].

Kehitysjonoa käyttämällä organisoimme työtämme jakaen kehitysaskeleet pieniksi osakokonaisuuksiksi, jotka arvioimme mielenkiinnon ja tärkeyden mukaan. Näitä kehitysjonojen osia yhdistäen saimme nykyisen kokonaisuuden aikaiseksi. Työjaksojen eri vaiheissa kirjasimme ylös myös dokumentaatiota. Versioiden välillä suunnittelimme uuteen versioon uusia toiminnallisuuksia. Kehitysjonoomme jäi vielä lukuisia jatkokehitysideoita myöhempää kehittämistä varten.

4 Suunnittelu

Idean insinöörityöhön saimme Innovaatioprojekti-kurssista, jossa projektinamme oli Ubuntu-käyttöjärjestelmän automaattisen asennuksen suorittaminen Kickstart-ohjelman

ja USB-massamuistilaitteen avulla. Päädyimme tähän ideaan, sillä halusimme edistää GNU/Linux-käyttöjärjestelmien käyttöönottoa sekä yritys- että kotikäytössä. Alun perin levykuvan automaattinen asennus oli tarkoitettu puhtaasti kotikäyttöön, mutta tutkiessamme asiaa tarkemmin huomasimme, että nykyisten teknologioiden avulla työmme on laajennettavissa myös yritysten käyttöön. Yhtenä lähtökohtana pidimmekin, että yrityksen tietotekninen ylläpitäjä voi ylläpitää levykuvat ja asennustiedostot, eikä loppukäyttäjän tarvitse kuin ottaa kone käyttöönsä.

Suunnitteluvaiheessa tutkimme myös erilaisia valmiita ratkaisuja käyttöjärjestelmien etäasennukseen loppukäyttäjälle, kuten FAI-projektia [7] ja CloneDeploy-ohjelmaa. CloneDeploy-projektin kehitys lopetettiin insinöörityömme aikana, joten sen käyttö oli lähtökohtaisesti turhaa. Työssä päädyimmekin mieluummin käyttämään olemassa olevilla työkaluilla itse kehitettyä etäasennusratkaisua. Keksimmekin nopeasti käyttää erillistä palvelinta ja PXE:tä levykuvien jakeluun, jolloin työmme on toteutettavissa ja että sen käyttäminen olisi ylläpitäjille mahdollisimman helppoa.

Päätimme suunnittelun jälkeen, että käytämme työssämme erillistä palvelinta, joka jakelee levykuvat verkon uusille koneille. Meidän täytyi myös miettiä, millainen verkko tukisi parhaiten suunnitelmaamme. Päätimme suunnitteluvaiheessa, että suuren verkon luominen ei ole tarkoituksenmukaista, joten päätimme keskittyä pienen mutta helposti laajennettavan verkon suunnitteluun.

4.1 Käyttöjärjestelmävalinta

Ideoidessamme tulevaa työtä meidän täytyi valita käyttöjärjestelmä. Otimme vertailuun jakelut, jotka olivat Kickstart-yhteensopivia kuten Red Hat Enterprise Linux, CentOS, Ubuntu, Fedora ja Debian.

Taulukko 1. Käyttöjärjestelmien vertailu.

Distro	**Pakettimanageri**	**Vakiotyöpöytä**	**Pohjautuu**	**Ilmainen**
Debian	apt	Gnome	-	Kyllä

Ubuntu 18.04	apt	Gnome	Debian	Kyllä
Ubuntu 16.04	apt	Unity	Debian	Kyllä
RHEL	rpm	Gnome	Fedora	Ei
CentOS	yum	Gnome	RHEL	Kyllä
Fedora	rpm	Gnome	-	Kyllä

Vertailun jälkeen päädyimme valitsemaan GNU/Linux Ubuntu -käyttöjärjestelmäjakelun. Ubuntu on yksi suosituimmista GNU/Linux-jakeluista [8]. Laaja pakettien saatavuus, ohjelmistotuki ja helppokäyttöisyys on iso osa Ubuntun suosiota. Myös aikaisempi tuntemus Debian-pohjaisiin jakeluihin otettiin huomioon valinnassa.

Versioksi valitsimme 18.04 LTS -version. Meille tuli kuitenkin melko nopeasti selville, että käyttämämme Kickstart-ohjelmisto ei tue kyseistä Ubuntu-jakelua. Päädyimme kokeilemaan Ubuntun aiempaa versiota 16.04 LTS, mutta emme saaneet aluksi sitäkään toimimaan. Lopulta saimme automaattisen asennuksen toimimaan vaihdettuamme käyttöjärjestelmän palvelinversioon sekä asentamalla graafisen käyttöliittymän manuaalisesti asennuksen yhteydessä.

4.2 Käyttäjien hallinta.

Keskitettyä käyttäjienhallintaa suunnitellessa kantavina ajatuksinamme olivat helppokäyttöisyys ja automaattisuuden helppo toteuttaminen. Hallintaa varten vertailimme Canonicalin Landscape-, OpenLDAP- ja FreeIPA-ohjelmistoja. OpenLDAP on avoimen lähdekoodin toteutus LDAP-protokollalle, joka on riisuttu ratkaisu muihin vaihtoehtoihin verrattuna. Myös OpenLDAP:iin on saatavilla verkkokäyttöliittymä phpLDAPadmin-sovelluksella [9]. FreeIPA ja Landscape ovat kattavampia ratkaisuja, joihin kuuluu verkkokäyttöliittymä.

Landscape on palvelimien, pilvien ja isojen verkkojen ylläpitoon ja käyttöönottoon tarkoitettu ohjelma, jossa on osana myös käyttäjänhallinta [10]. Landscapella olisimme voineet toteuttaa myös keskitetyn pakettien jakelun, laitteiden ryhmittämisen, turvallisuusauditoinnin ja paikallisia pakettisäiliöitä. Päätimme kuitenkin KISS-periaatetta käyttäen olla

ottamatta sitä käyttöön, sillä se olisi ollut laajuuden takia epäkäytännöllinen tässä työssä [11].

OpenLDAP vaikutti aluksi hyvältä ratkaisulta, mutta suoritettuamme käytännön kokeilua huomasimme, ettei se olekaan toimiva vaihtoehto meidän ratkaisullemme. Kun kokeilimme OpenLDAP:ia järjestelmässämme törmäsimme jatkuviin ongelmiin loppukäyttäjän koneen graafisen käyttöliittymän kanssa. Koska työmme kantavana ajatuksena olivat helppokäyttöisyys, automaattisesti toimivat asennukset ja asetuksien muokkaaminen, päätimme, ettei OpenLDAP ole sopiva järjestelmäämme sen aiheuttamien yhteensopivuusongelmien vuoksi.

Kun huomasimme, etteivät ensimmäiset kaksi vaihtoehtoamme olleet käytännöllisiä työssämme, päätimme jatkaa testauksia FreeIPA-ohjelmiston kanssa. FreeIPA-ohjelmiston käyttöönottoa testatessamme huomasimme, että se on huomattavasti helpompi ottaa käyttöön automaattisesti ja ulkonäöltään laadukkaampi kuin kaksi muuta vaihtoehtoa. Graafinen verkkokäyttöliittymä oli selkeä ja hyvin samanlainen kuin Microsoft Active Directory, joka on tutumpi käyttäjienhallintaohjelmisto ylläpitäjille.

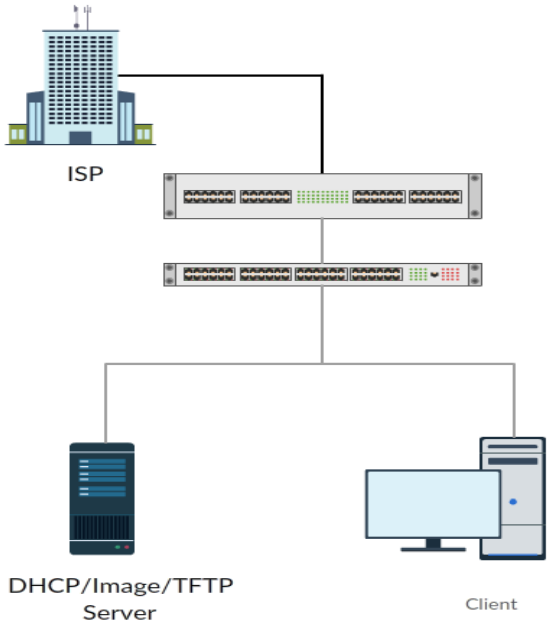
FreeIPA on integroitu identiteetti- ja todennusratkaisu GNU/Linux-verkkoympäristöön. Ratkaisu mahdollistaa keskitetyn todentamisen, valtuutuksen ja tilastoinnin AAA-protokollan mukaisesti. FreeIPA käyttää LDAP-version 3 hakemistoarkkitehtuuria ja mahdollistaa kertakirjautumistodentamisen käyttäen Kerberos-todennusprotokollaa. Kokeiltuamme FreeIPA:a käytännössä päätimme käyttää sitä myös jatkokehityksessä.

5 Toteutus

5.1 Verkko

Testiympäristössämme toimii verkkona yksinkertainen C-luokan 192.168.0.0/24-verkko, jossa palvelimen kautta jaamme DHCP-protokollan välityksellä osoitteita asiakaskoneille. DHCP jakaa kaikki muut osoitteet laitteille niitä lukuun ottamatta, jotka olemme

manuaalisesti määritelleet. Kiinteän IP-osoitteen määritämme pelkästään DHCP-palvelimelle. Pyrimme pitämään testiympäristön verkon yksinkertaisena, mutta riittävänä omiin testaustarpeisiimme. Käytimme testiympäristön kehittämiseen useampaa reititintä.



Kuva 3. Esimerkki verkkomme topologiasta

5.2 Palvelin

Testiympäristössämme palvelimenamme toimii HP Z210 workstation-pöytäkone, joka suorituskykynsä puolesta hoitaa tehtäväänsä mallikkaasti.

Taulukko 2. Palvelimen komponentit.

Prosessori	3.4GHz quad core - 8 thread processor Intel i7-2600
Keskusmuisti	12 GB RAM

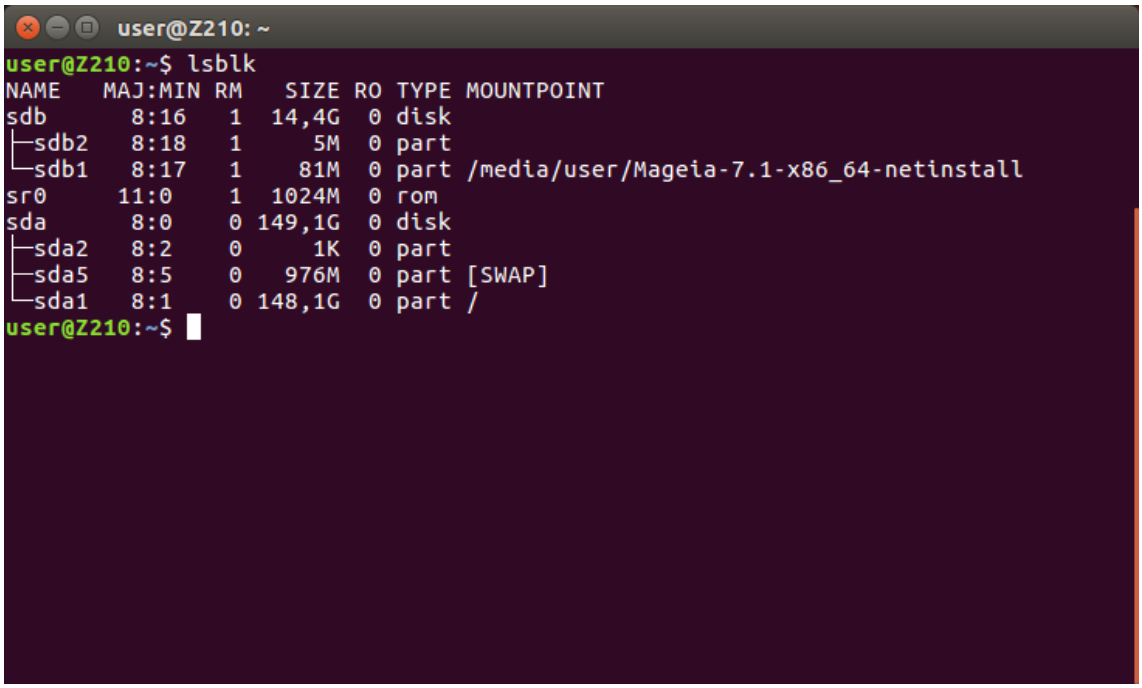
Kovalevy	120 GB SSD
Näytönohjain	GPU GTX 670

Palvelimen käyttöönotto aloitetaan asentamalla tulevaan palvelimeen Ubuntu Desktop -käyttöjärjestelmä. Asennusmedian valmistelu tapahtuu lataamalla Ubuntun 16.04 LTS -version tuorein tarjolla oleva levykuva Canonical Ubuntun kotisivuilta. Jotta Ubuntu voidaan asentaa palvelinkoneelle, täytyy levykuva kirjoittaa ensin USB-muistitikulle. Levykuvan kirjoitus USB-muistitikulle tapahtuu käyttämällä terminaalissa dd-ohjelmaa käyttävän komennon [11].

```
sudo dd if=/home/user/Downloads/ubuntu-16.04.6-desktop-amd64.iso of=/dev/sdx
status=progress
```

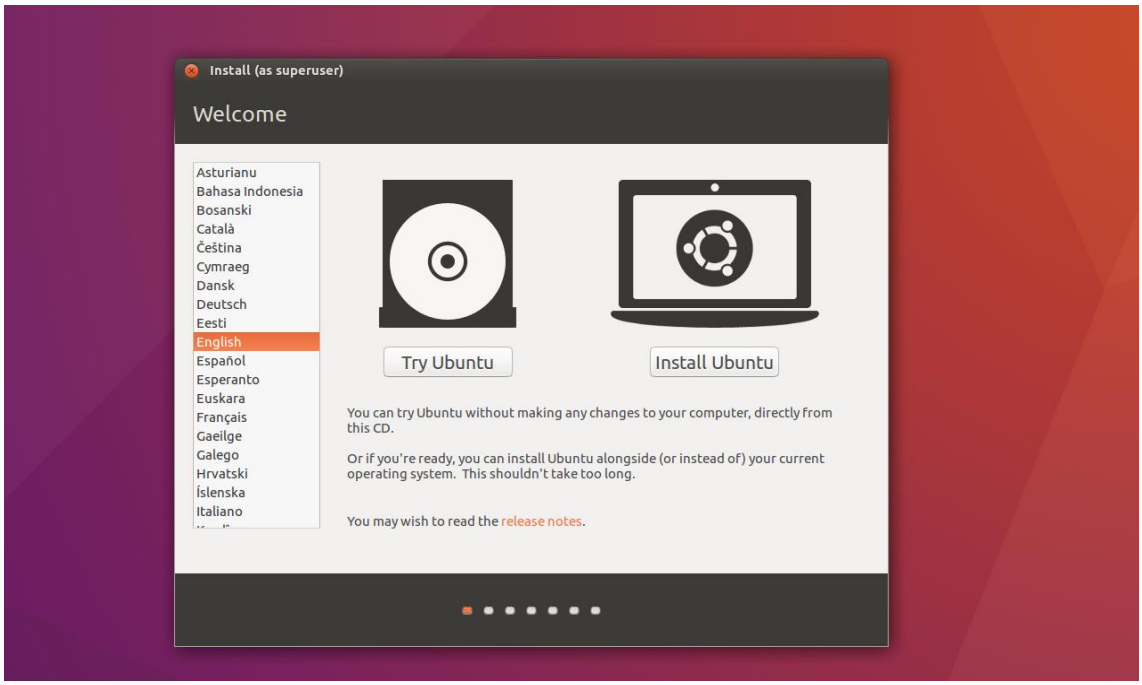
Esimerkkikoodi 1. dd-komennon käyttö.

Komennon dd-parametriin ‘if=’ määritellään sisälle tulevan tiedoston polku, joka tässä tapauksessa on Ubuntun levykuva. Parametriin ‘of=’ määritellään laitteen osoite, johon tiedosto kirjoitetaan. Kirjoituskohteen osoite /dev/sdx selvitetään lsblk-komennolla, joka listaa tietokoneessa kiinni olevat lohkolaitteet.



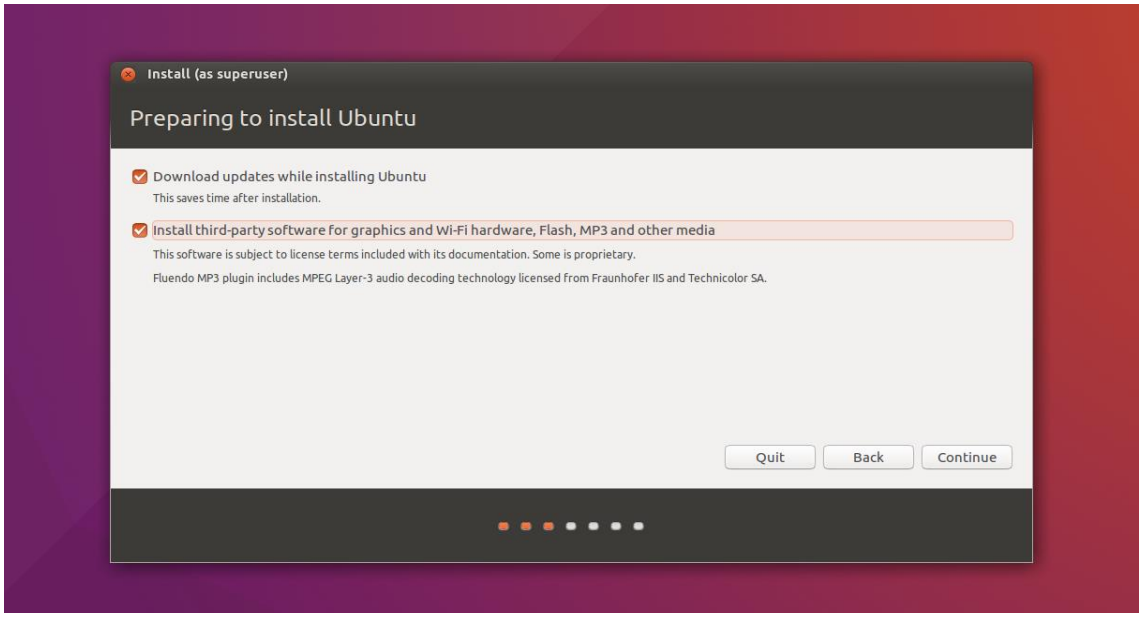
Kuva 4. Lista lohkolaitteista

Tässä tapauksessa kirjoituskohteen osoite on /dev/sdb. Tunnistimme oikean lohkolaitteen, sillä tiesimme, että liittämämme USB-muisti on 16 Gt:n kokoinen. Parametri ‘status=progress’ näyttää kirjoituksen etenemisen. Ilman ‘status=progress’-valintaa komento toimii myös, mutta se ei näytä kirjoituksen etenemistä eikä valmistumista.



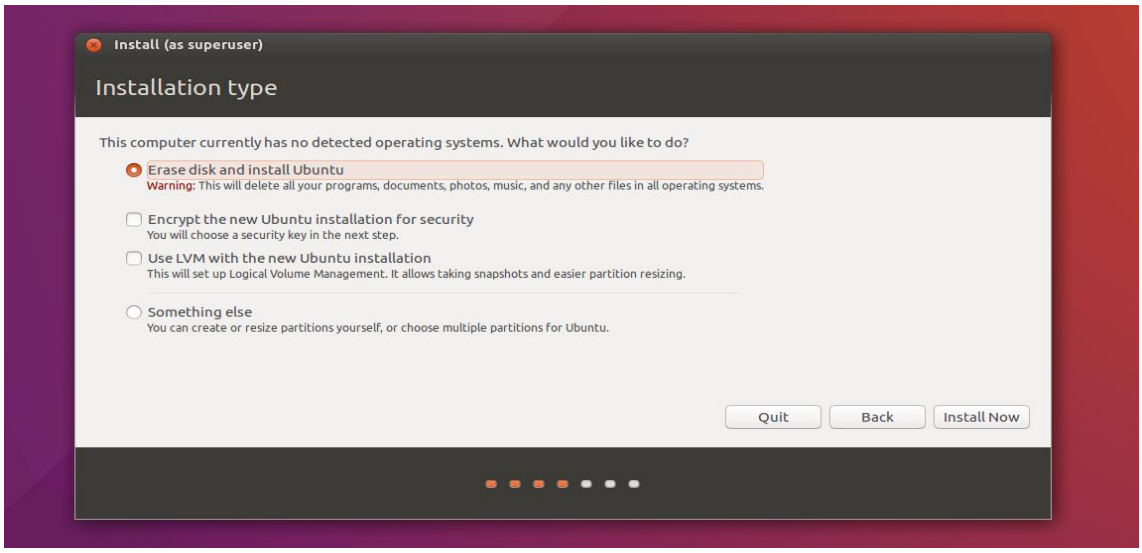
Kuva 5. Ubuntu asennus - kielivalinta.

Palvelimeen asennetaan Ubuntun työpöytäversio. Asennuksen alussa valitaan kieli, jolla Ubuntu halutaan asentaa. Olemme valinneet kieleksi englannin, jonka jälkeen valitaan ‘Install Ubuntu’.



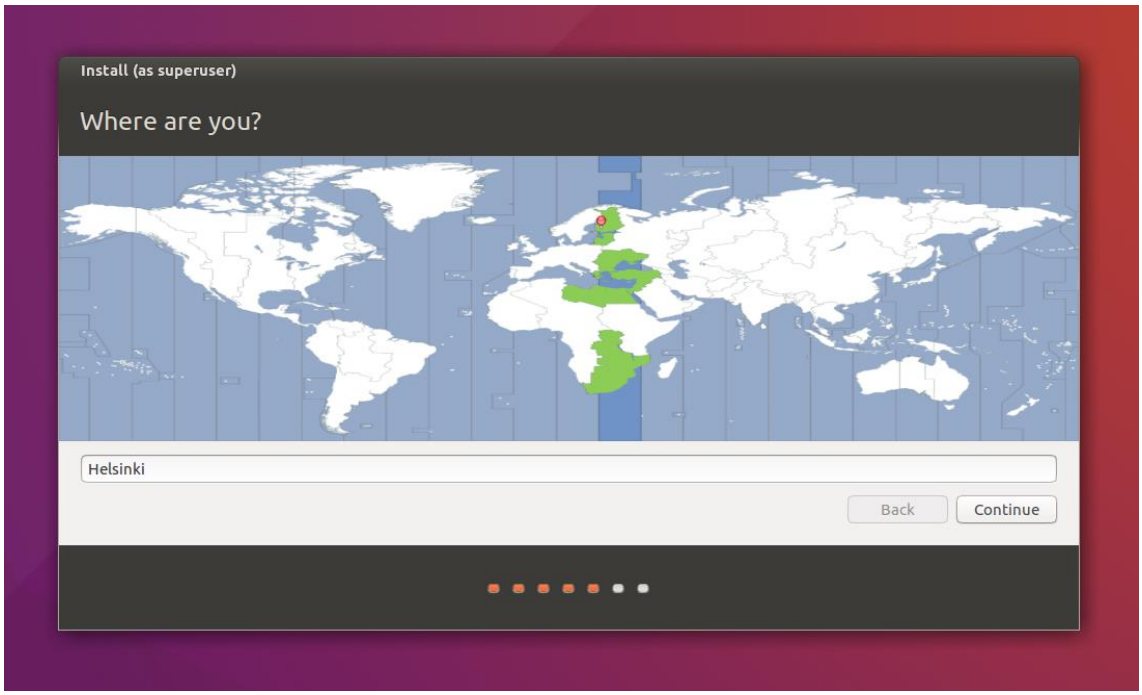
Kuva 6. Ubuntu asennus - Päivitysten määrittely

Valitsemme ‘Download updates while installing Ubuntu’-vaihtoehdon asennusvaiheessa, jotta päivitystä ei tarvitse suorittaa heti asennuksen jälkeen. ‘Install third-party software for graphics and Wi-Fi hardware, Flash, MP3 and other media’-valinnalla asentuvat mahdolliset tarvittavat non-free-ajurit.



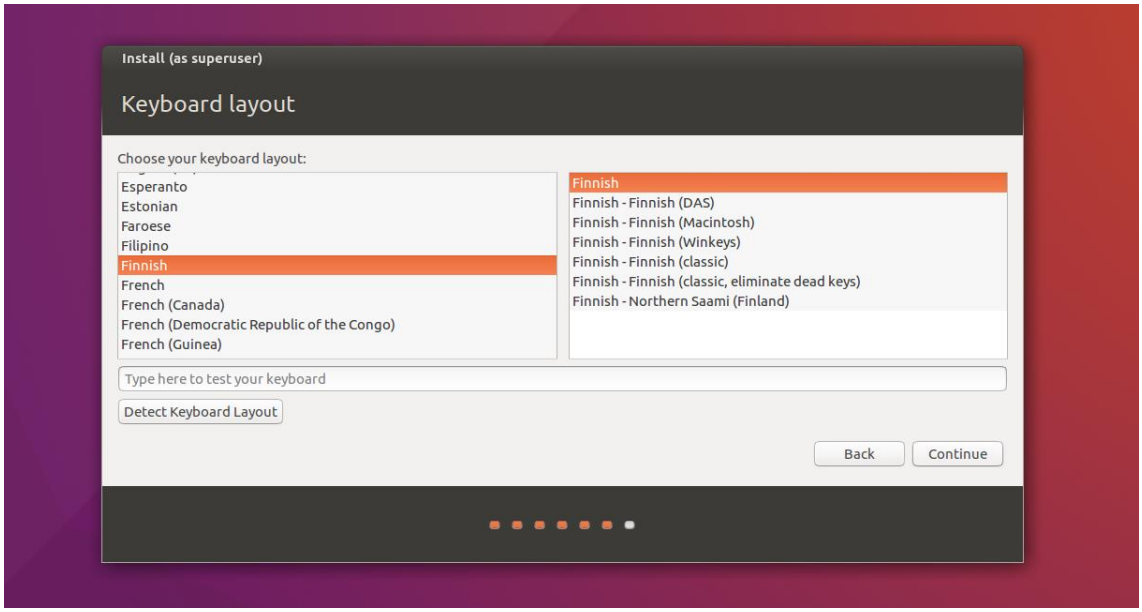
Kuva 7. Ubuntu asennus - Alustus.

Kun laitteessa ei ole asennettua käyttöjärjestelmää, asennusohjelmassa täytyy valita toiminnot käytettävissä oleville kovalevyille. Valittuna on vaihtoehto, jossa asennusohjelma poistaa kaiken datan kovalevyltä ja asentaa Ubuntu-käyttöjärjestelmän. Muut vaihtoehdot, joita tässä emme käytä, ovat ‘Encrypt the new Ubuntu installation for security’, joka tarkoittaa kovalevyn salausta tietoturvan parantamiseksi. ‘Use LVM with the new Ubuntu installation’ tarkoittaa Logical Volume Management -järjestelmää, joka mahdollistaa useiden kovalevyjen joustavan osioinnin. ‘Something else’ -valinnasta pääsee perinteiseen kovalevyjen osiointiin.



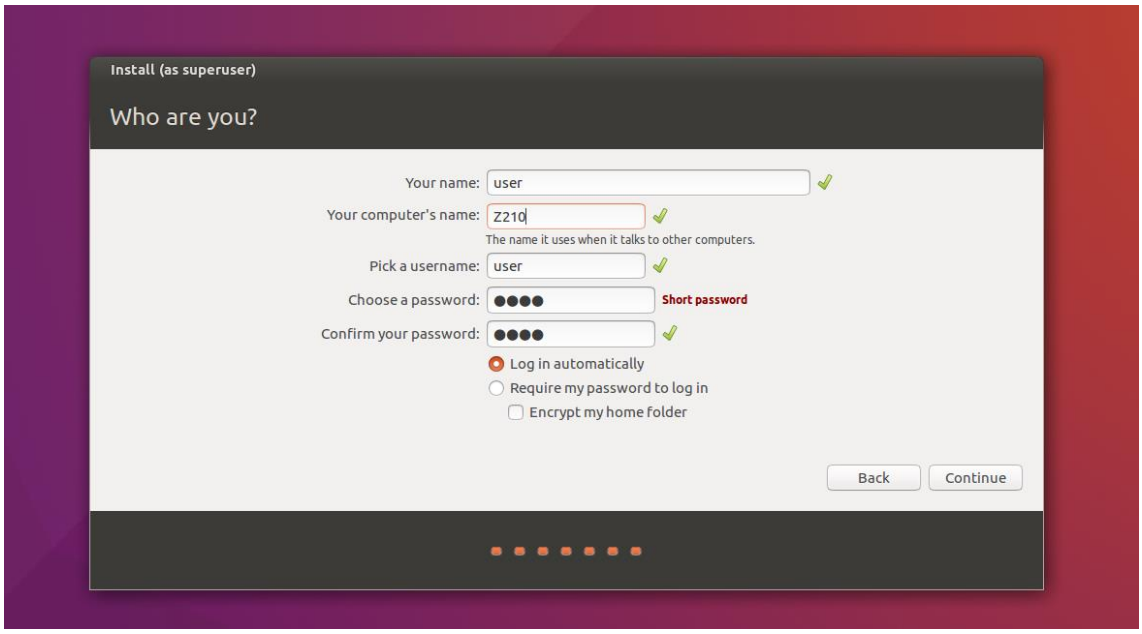
Kuva 8. Ubuntu asennus - aikavyöhyke

Seuraavassa vaiheessa määritetään aikavyöhyke. Käytämme aikavyöhykettä Helsinki.



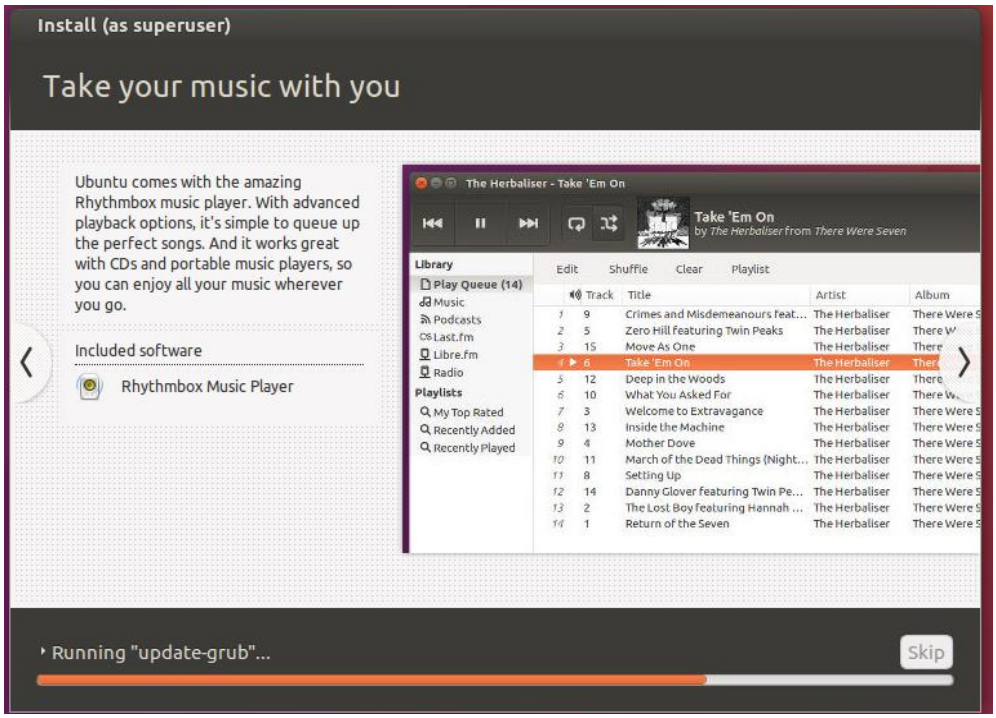
Kuva 9. Ubuntu asennus - näppäimistön kieli

Kuvassa määrittelemme näppäimistön käyttämään suomenkielistä pohjaa.



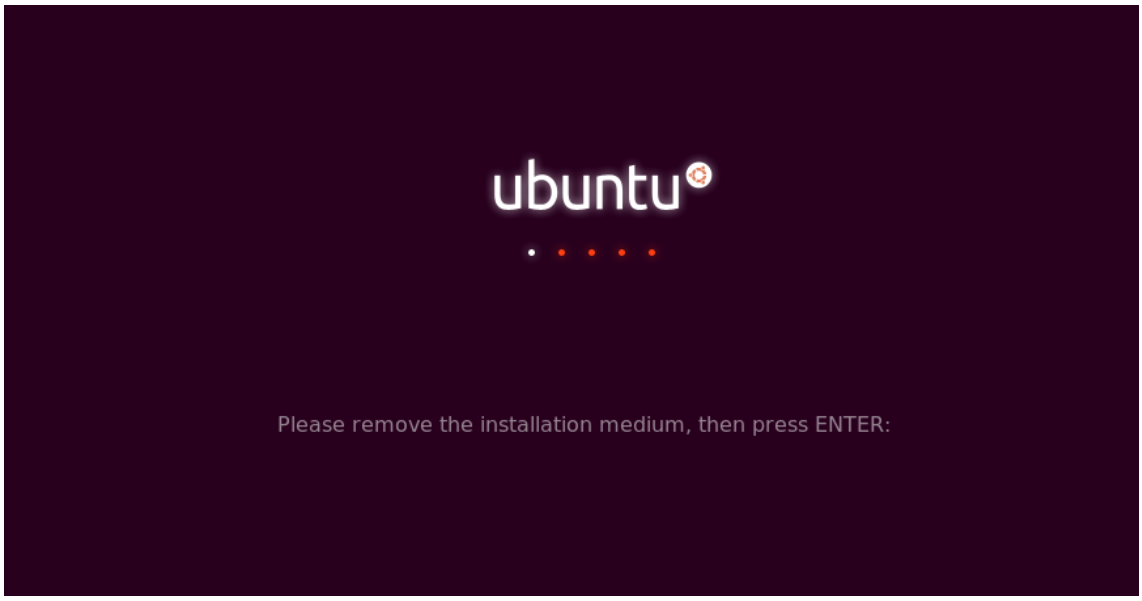
Kuva 10. Ubuntu asennus - Käyttäjän määrittely.

Määritellään user-testikäyttäjä ja asetetaan palvelimen isäntänimi, joka on palvelimena toimivan tietokoneen malli. Valitaan automaattinen kirjautuminen tälle tunnukselle.



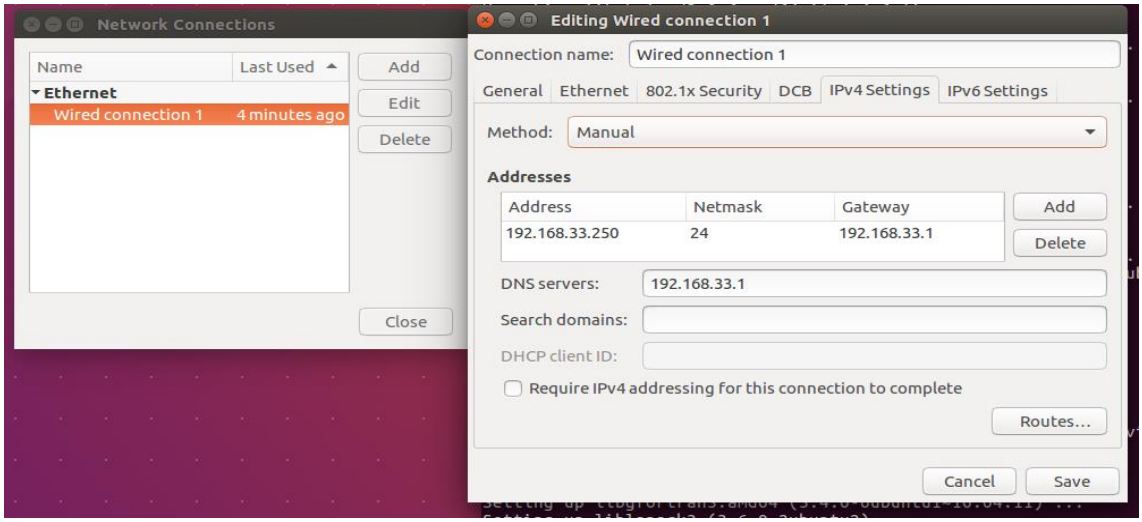
Kuva 11. Ubuntu asennus - grub.

Tässä vaiheessa käyttöjärjestelmä asentuu. Edellä näkyy ehdotuksia ohjelmistoista, joita suositellaan käytettäväksi. Kuvassa asentuu ‘grub’, eli käyttöjärjestelmän lataaja.



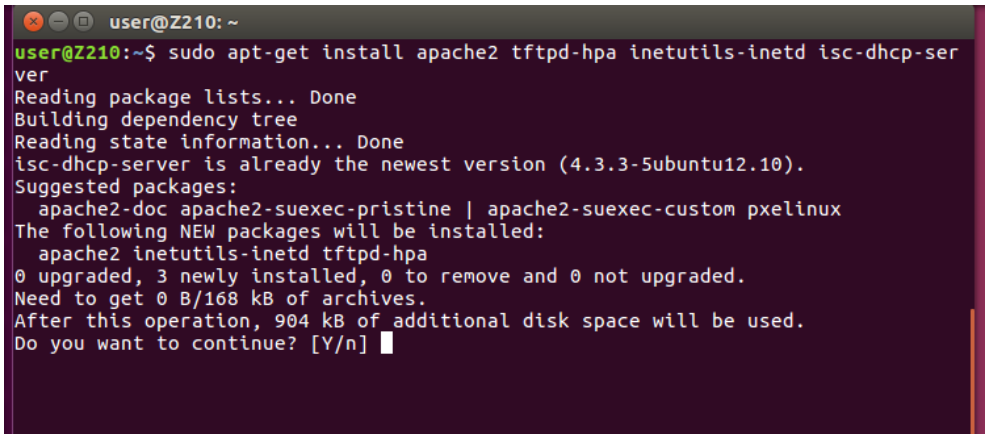
Kuva 12. Ubuntu asennus - valmis

Asennuksen valmistuttua poistetaan asennusmedia laitteesta ja käynnistetään uudelleen. Uudelleenkäynnistyksen jälkeen Ubuntu on asennettu ja valmiina personoitavaksi palvelinkäyttöön.



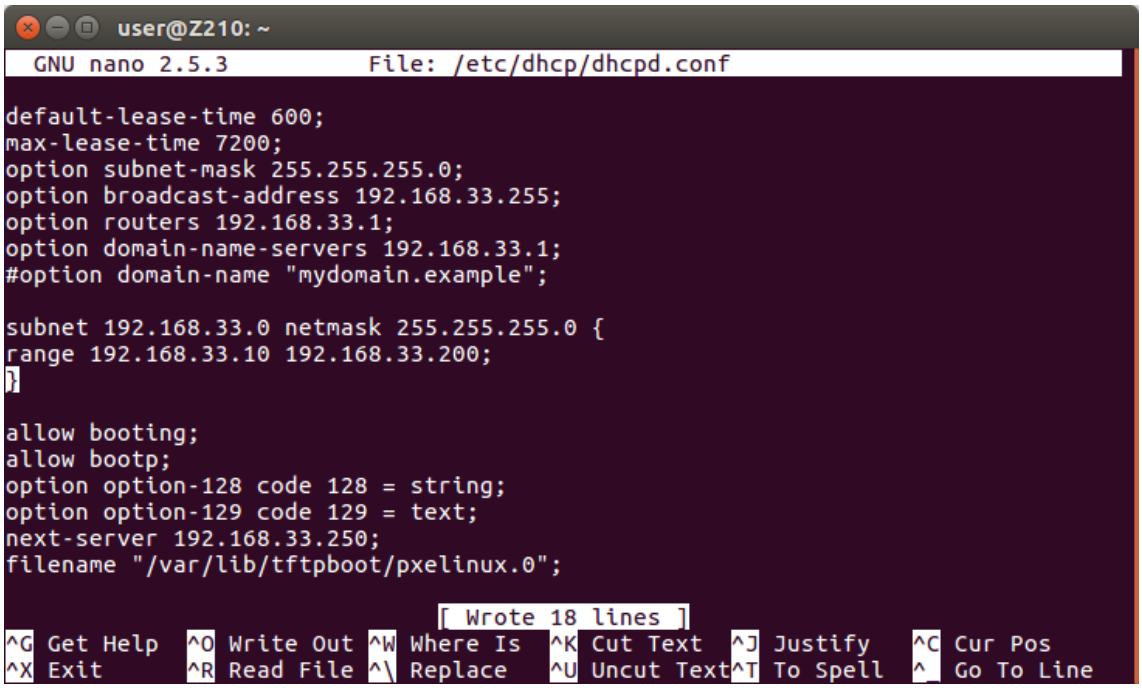
Kuva 13. Palvelin - verkkoasetukset.

Kun käyttöjärjestelemä on asennettu palvelimeen, aloitamme palvelimen konfiguroinnin. Ensimmäisenä syötämme manuaalisesti verkkoon tarvittavat osoitteet. Palvelimeemme asetetaan osoitteeksi 192.168.33.250, 24-bittinen maski ja gatewaynä toimii verkkomme reititin. DNS-kyselyt ohjataan verkon reitittimelle ja sieltä eteenpäin.



Kuva 14. Pakettien asennus.

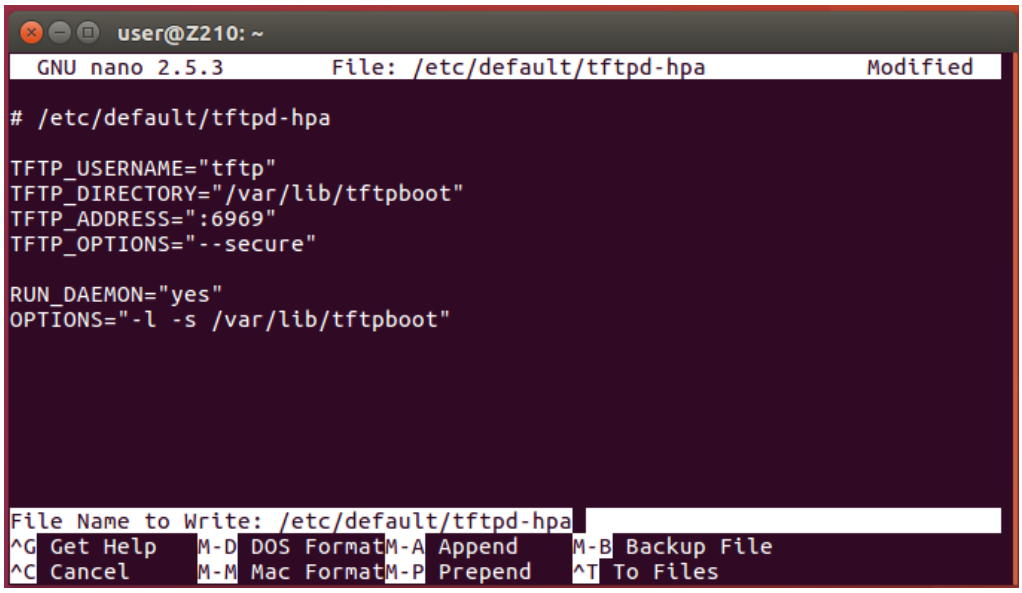
Kun käyttöjärjestelmä on asennettu ja verkkoyhteys määritetty, voimme aloittaa pakettien asentamisen. Seuraavaksi asennamme palvelimelle seuraavat paketit: Apache2:n, Tftpd-hpa:n, inetutils-inetd:n ja Isc-dhcp-serverin. Tftpd-hpa-paketti on palvelinohjelma TFTP -verkkoprotokollalle. TFTP-protokollaa käyttämällä ladataan palvelimelta PXE Serverin käyttämät tiedostot. Isc-dhcp-server-paketti asentaa koneelle DHCP-palvelimen. Koska olemme asentamassa systeemin pääpalvelinta, on DHCP tärkeä, koska se hoitaa IP-osoitteiden jakelun verkkoon liittyville laitteille.



Kuva 15. DHCP-konfiguraatio

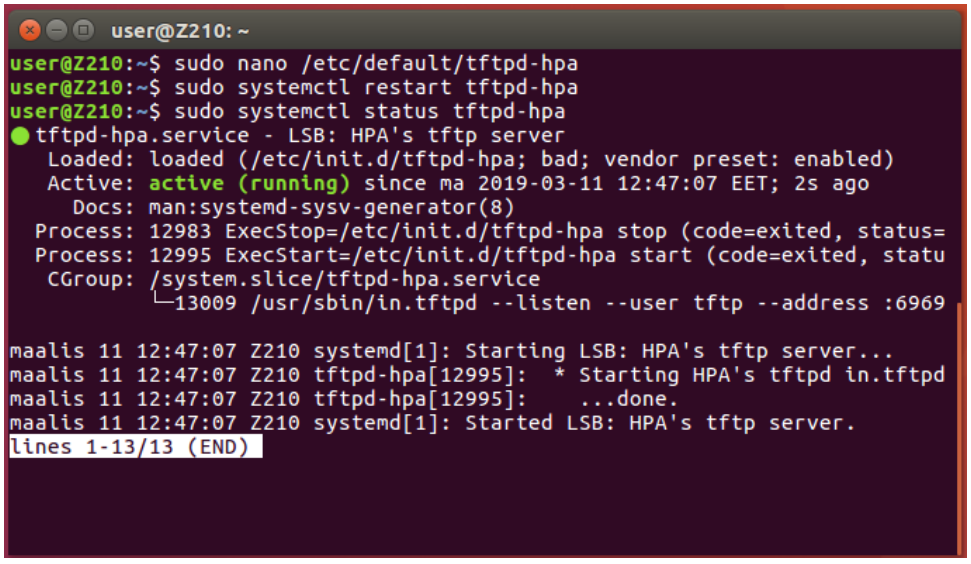
Tiedostoon dhcpd.conf konfiguroidaan suunnittelemamme verkon asetukset, jotta asiakaskoneet saavat ne haettua. Dhcpd.conf:iin määrittelemme myös, että verkkokäynnistys on sallittua ja mistä osoitteesta verkkokäynnistys on suoritettava. Dhcp-alueeksi määritellään .10-200, joka käyttää lähes kaikki verkossa vapaana olevat osoitteet ja mahdollistaa yhteensä 190 loppukäyttäjän laitetta. Määrittelemme myös reitittimemme toimimaan oletusyhdyskäytävänä ja nimipalvelimena loppukäyttäjien laitteille.

Allow booting; ja allow bootp; ovat myös tärkeät, sillä ne sallivat PXE:n käytön verkossa. Next-server määrittelee verkkokäynnistyksessä käytettävän palvelimen osoitteen. Filename “/var/lib/tftpboot/pxelinux.0” määrittelee tiedoston, josta verkkokäynnistys suoritetaan.



Kuva 16. Tftpd-hpa-konfiguraatio.

Tftpd-hpa konfiguroidaan käyttämään porttia 6969. Asetuksella -l asetamme tftpd-hpa:n kuuntelemaan sille tulevia kutsuja UDP-portista 6969, kun taas -s tarkoittaa –secure, jolla määritellään tftpd:n käyttämä kansio.



Kuva 17. Tftpd-hpa:n käynnistys.

Kun tftpd-hpa on konfiguroitu, tulee se käynnistää uudelleen komennolla ’systemctl restart tftpd-hpa’. Status-komennolla varmistetaan, että prosessi käynnistyi.

FreeIPA vaatii toimiakseen FQDN-formaatissa olevan ns. absoluuttisen isäntänimen. Se asetetaan järjestelmään käyttämällä hosnamectl-komentoa ja sen parametreinä ‘set-hostname’ sekä haluttu isäntänimi. Isäntänimi myös kirjoitetaan järjestelmän hosts-tiedostoon, josta järjestelmä hakee ensimmäisenä nimikyselyitä.

```
sudo hostnamectl set-hostname z210.testi.net
echo "192.168.33.250 z210.testi.net" >> /etc/hosts
```

Esimerkkikoodi 2. Isäntänimen asettaminen.

Päivitetään pakettisäiliöt admin-oikeuksilla käyttäen komentoa: ’sudo apt update -y’.

Asennetaan rng-tools-paketti pakettisäiliöstä komennolla ’sudo apt install rng-tools’.

Kun asennus on suoritettu, muokataan tiedostoa /etc/default/rng-tools ja asetetaan lähteeksi satunnaisdatalle /dev/urandom lisäämällä rivi HRNGDEVICE=/dev/urandom tiedoston loppuun. Tiedoston muokkaus onnistuu komennolla:

```
sudo nano /etc/default/rng-tools
```

Kun muokkaukset ovat valmiita, aktivoidaan rng-tools-prosessi komennolla:

```
sudo systemctl enable rng-tools
```

Aktivointi täytyy tehdä, jotta rng-tools-prosessi käynnistyy automaattisesti koneen käynnistyessä.

Kun rng-tools on aktivoitu, voidaan se käynnistää joko kone uudelleenkäynnistämällä tai komennolla:

```
sudo systemctl start rng-tools
```

Seuraavaksi asennamme freeipa-server-paketin komennolla:

```
sudo apt -y install freeipa-server
```

Paketin asennuksen yhteydessä syötetään komentokehotteisiin Kerberos-alue “realm” TESTI.NET, Kerberos palvelin z210.testi.net ja isäntänimi hallinnointipalvelimelle z210.testi.net.

Seuraavaksi määrittelemme FreeIPA-palvelimen asetukset ajamalla määrittelytyökalun:

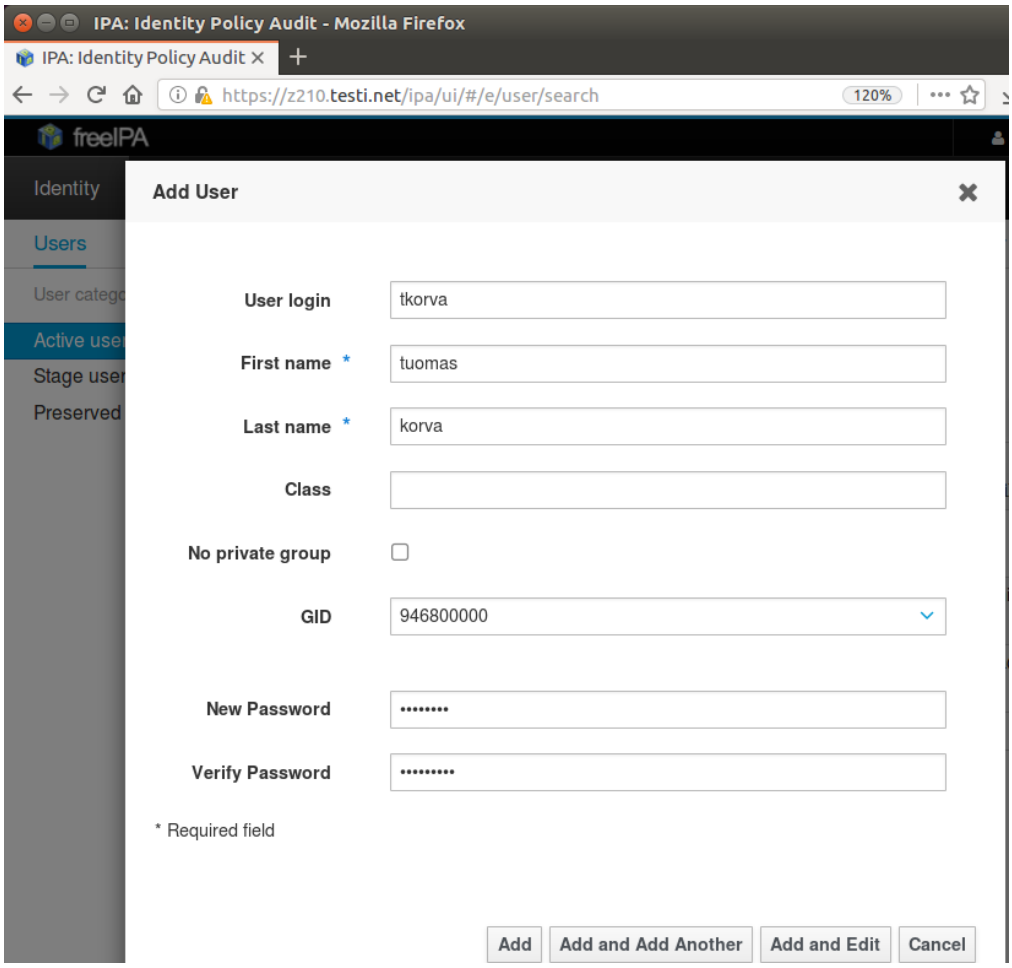
```
sudo ipa-server-install
```

Ensimmäisessä komentokehotteessa kysytään, käyttääkö FreeIPA:n integroitua DNS-palvelua, valitsemme “No”, sillä emme käytä palvelintamme nimipalvelimena.

Toisessa komentokehotteessa kysytään FreeIPA:n admin-käyttäjän salasanaa. Asetamme käyttäjälle salasanaksi: testi123. Tämän jälkeen asennukseen ei tarvitse syöttää enempää tietoja.

Asennuksen jälkeen siirrymme käyttämään verkkokäyttöliittymää. Verkkokäyttöliittymästä lisäämme testiksi käyttäjiä. Testikäyttäjä lisätään myös admins-ryhmään.

FreeIPA:n verkkokäyttöliittymään päästään syöttämällä selaimeen osoite https://z210.testi.net ja käyttämällä tunnuksia admin, testi123.



Kuva 18. FreeIPA - Käyttäjän lisäys.

Add user -valikosta lisäämme käyttäjiä. FreeIPA vaatii etunimen ja sukunimen, jonka perusteella ohjelmisto generoi käyttäjätunnuksen. GID-numero määrittää käyttäjän ryhmän numeerisen arvon. Numero 946800000 viittaa admins-ryhmään. Tässä näkymässä ylläpitäjä voi määritellä käyttäjän salasanan. Salasana tulee vaihtaa ensikirjautumisen jälkeen. Käyttäjän voi liittää tarvittavaan ryhmään joko tässä näkymässä tai valitsemalla päänäkymästä valitsemalla Identity - User Groups, jossa voi tehdä myös lisää ryhmiä.

5.3 Levykuva

Tässä luvussa käymme läpi, miten konfiguroimme levykuvan järjestelmäämme. Aluksi täytyy meidän ladata palvelimelle Ubuntu 16.04 -levykuvan uusin versio osoitteesta

http://releases.ubuntu.com/16.04/. Kun levykuva on ladattu, sen sisältö otetaan käyttöjärjestelmän käyttöön komennolla:

```
mount -o loop ubuntu-16.04* /mnt/iso
```

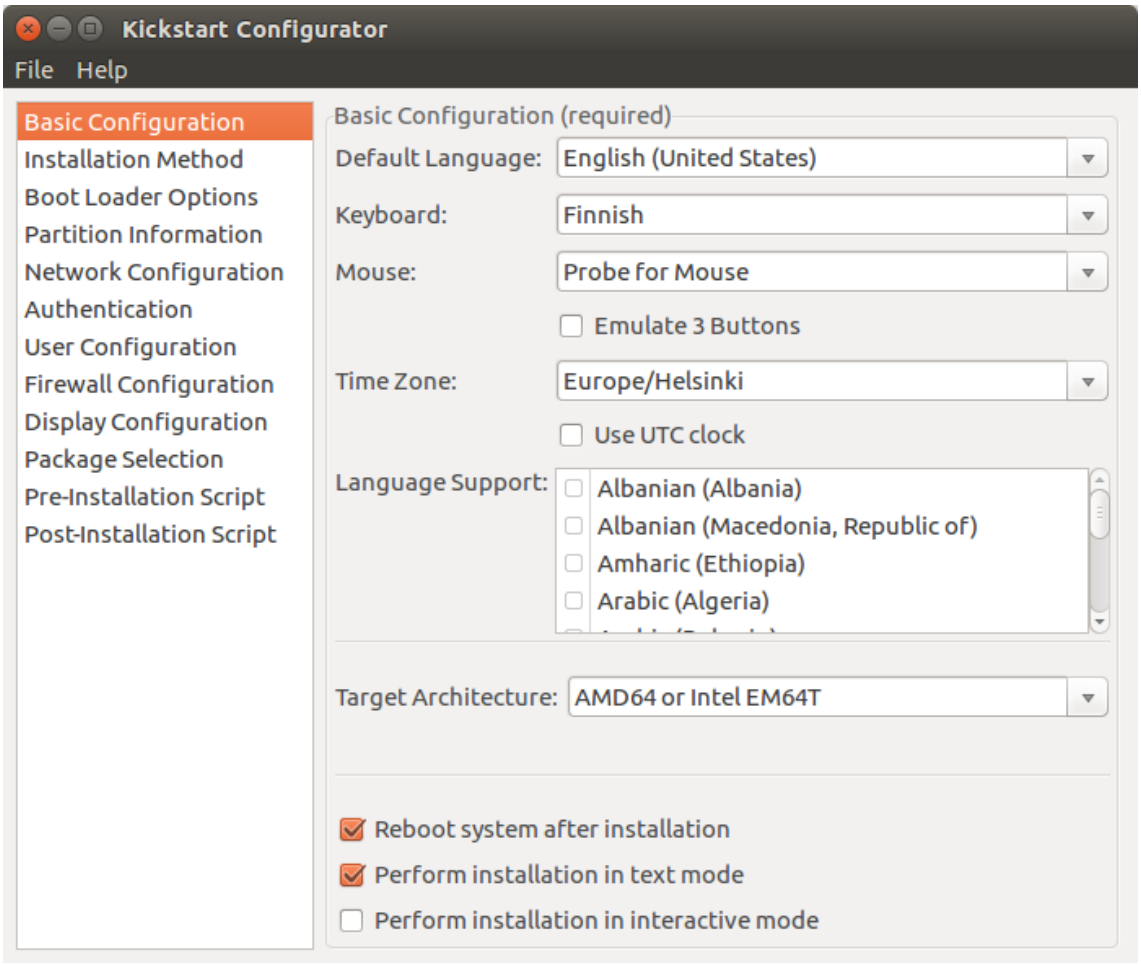
Tämän jälkeen levykuvan tiedostot kopioidaan palvelimemme /var/www/html/ubuntu-hakemistoon komennolla:

```
cp -rT /mnt/iso /var/www/ubuntu/
```

Koska meidän täytyy ajaa asennus automaattisesti ja suorittaa samalla täydentäviä terminaalikomentoja, tulee meidän asentaa Kickstart-ohjelma. Kickstart-ohjelma asennetaan terminaalin kautta komennolla:

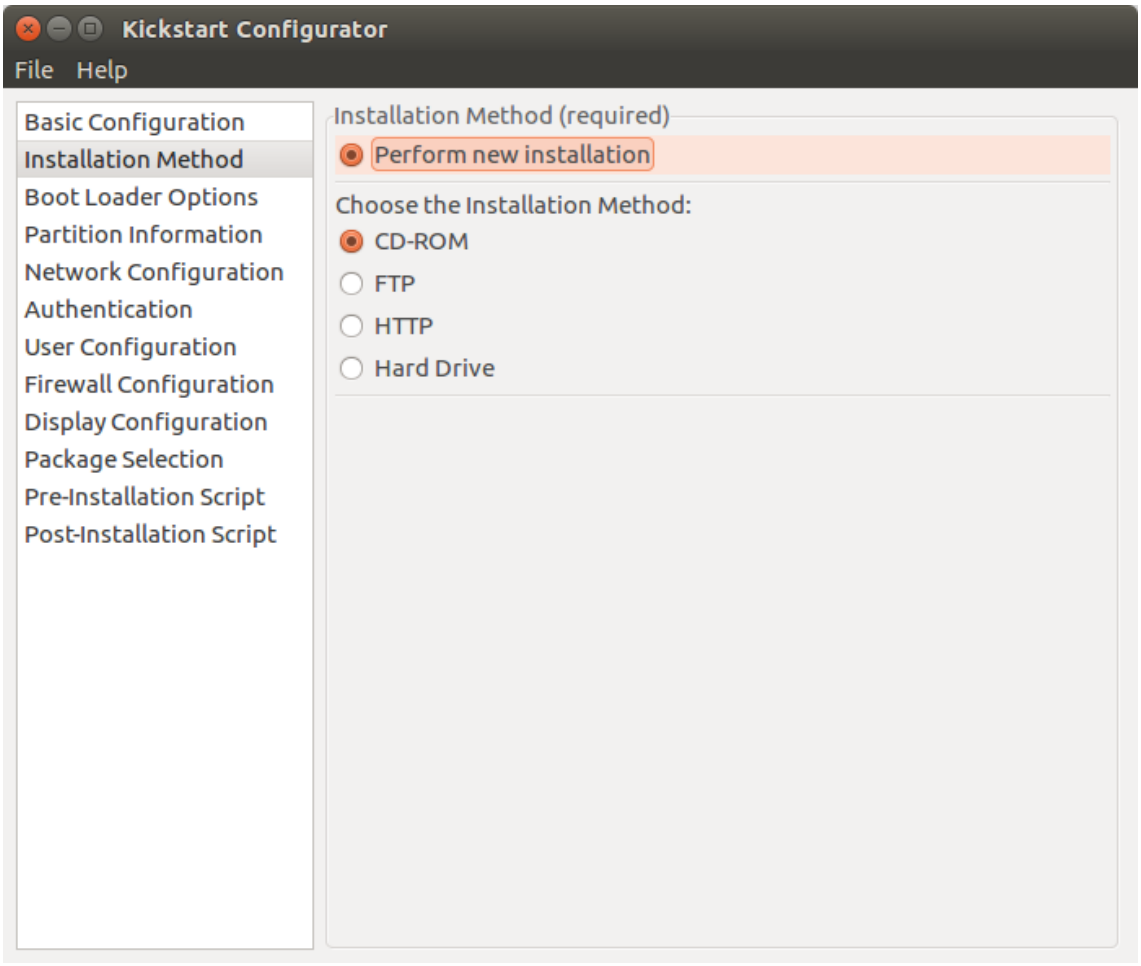
```
sudo apt install system-config-kickstart
```

Kickstart-konfiguroinnissa käytämme kickstartin omaa graafista konfigurointityökalua, joka asentuu paketissa system-config-kickstart. Kickstartissa on kaksitoista valikkoa, joista osaan täytyy tehdä muutoksia, jotta haluamamme lopputulos olisi mahdollista. Seuraavaksi käymme läpi askel askeleelta, mihin valikoihin tulee tehdä muutoksia, jotta automaattinen asennus toimisi loistavasti.



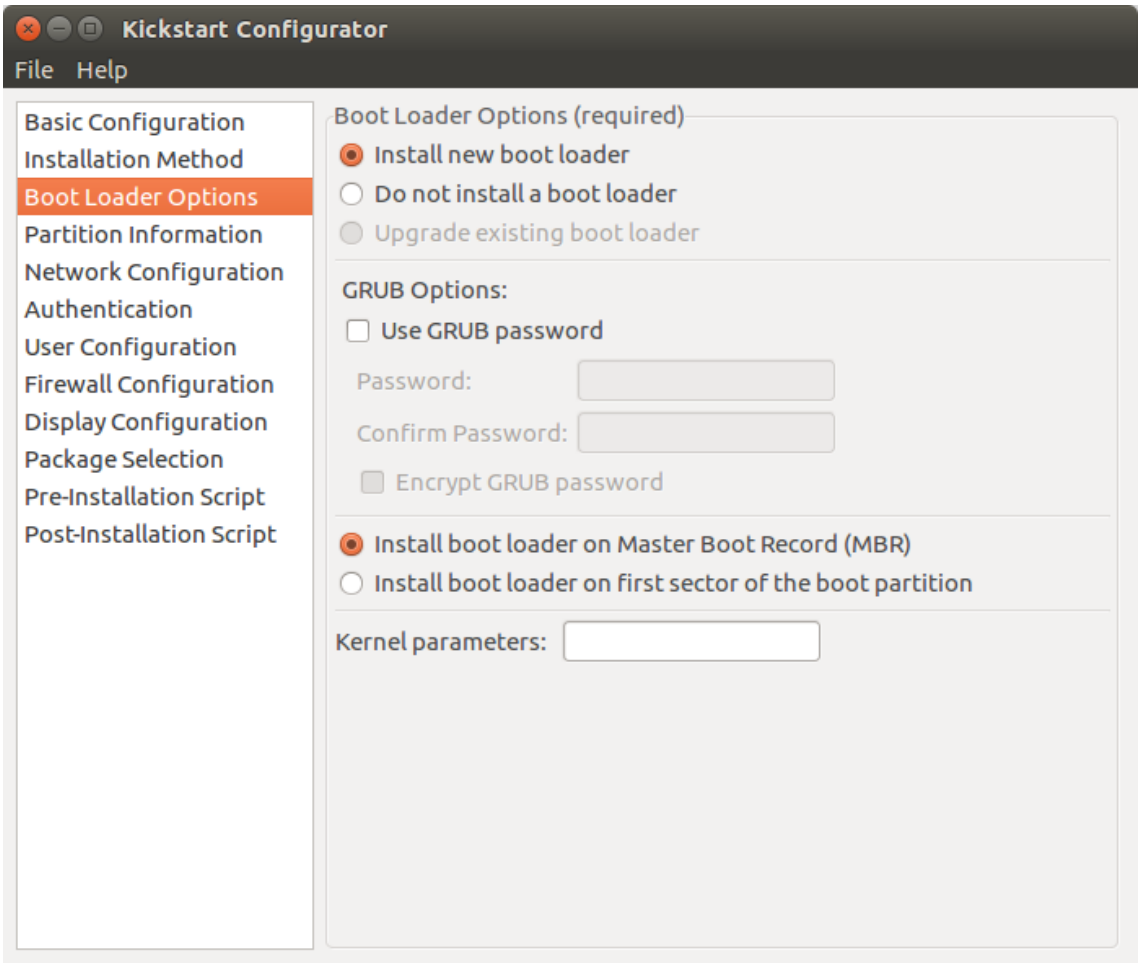
Kuva 19. Kickstart - Perusasetukset.

Vaiheessa Basic Configuration määrittelemme käyttöjärjestelmän kielen, näppäimistön kielen, aikavyöhykkeen, kohdearkkitehtuurin, uudelleenkäynnistyksen asennuksen jälkeen ja sen, että asennus suoritetaan tekstitilassa. Kickstartin ja Ubuntun yhteensopivuusongelmien takia tässä määrittelemämme näppäimistön kielen, ja käyttöjärjestelmätuetut kielet eivät toimi, vaan määrittelemme ne myöhemmin uudestaan asennuksen jälkeisessä komentosarjassa.



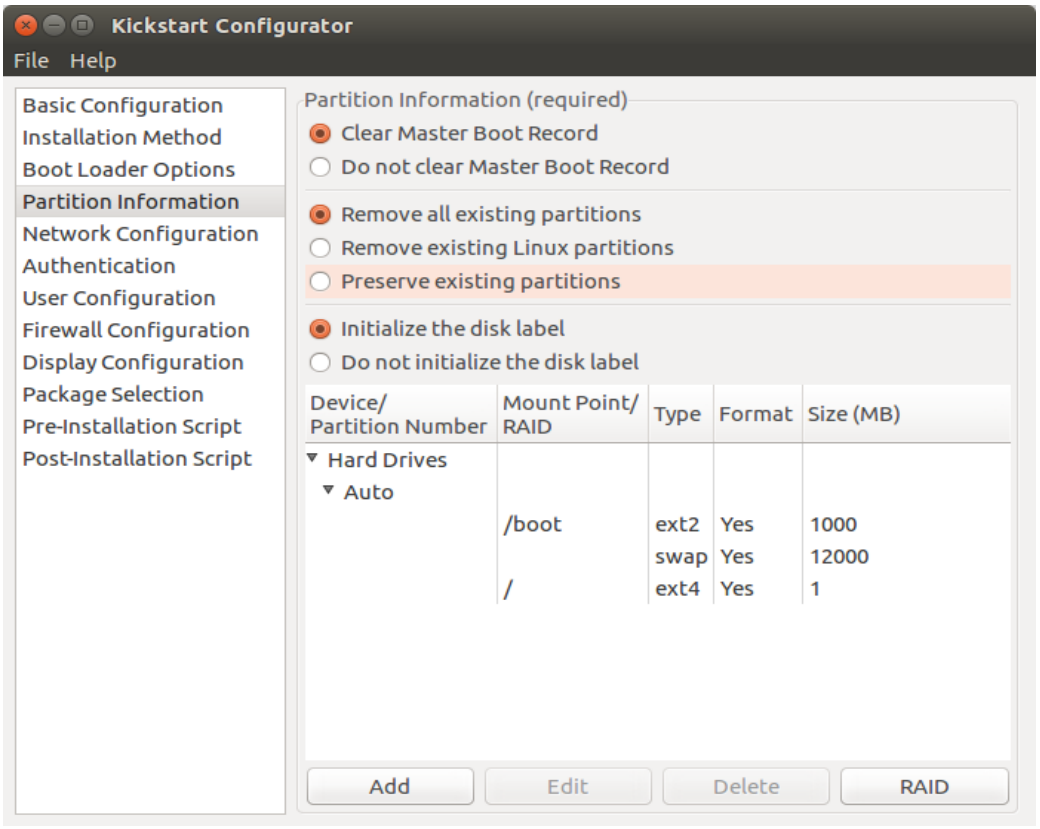
Kuva 20. Kickstart-asennusmetodi.

Installation Method -kohdassa valitsemme, että suoritamme uuden asennuksen ja että asennusmetodina toimii CD-ROM.



Kuva 21. Kickstart - Käynnistyslataaja.

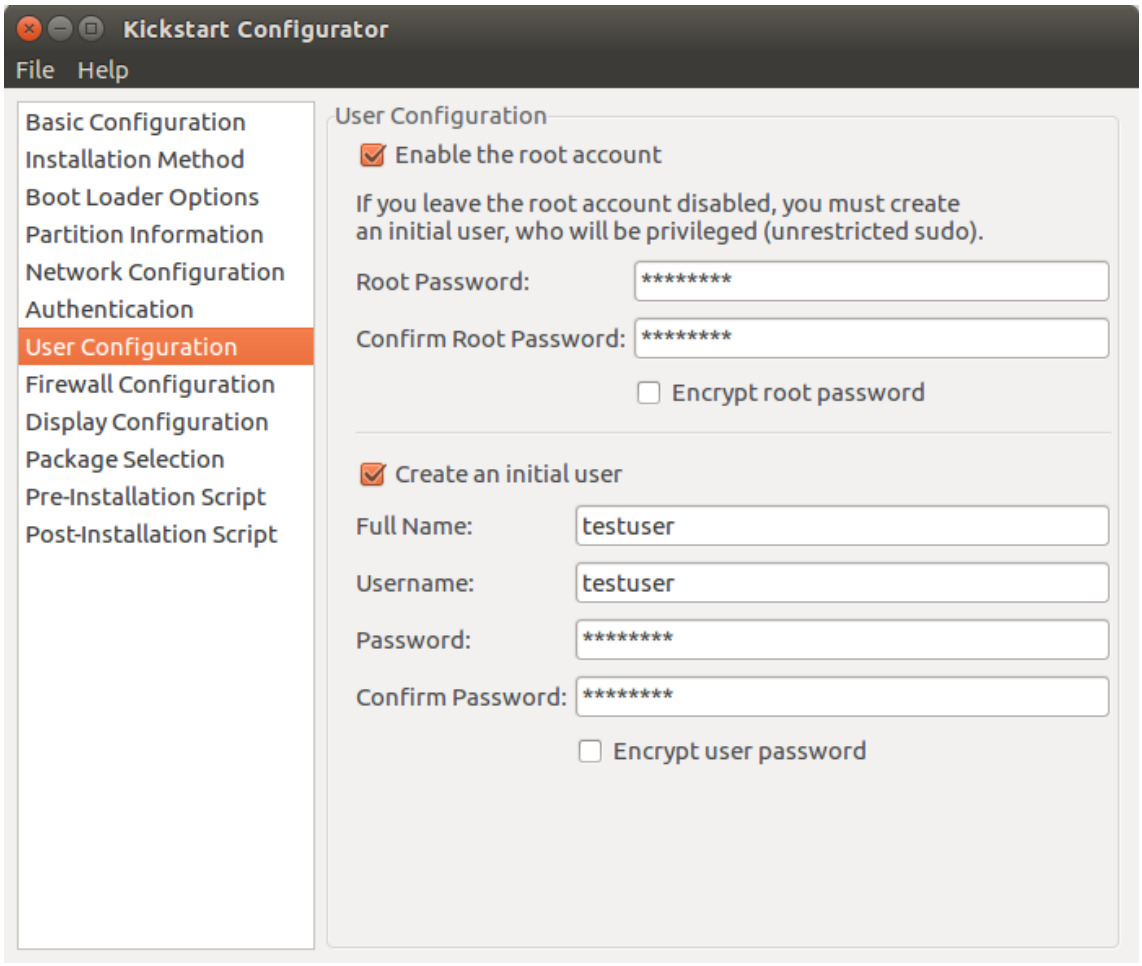
Boot Loader Options -valikosta valitsemme uuden käyttöjärjestelmän käynnistyslataajan asennettavaksi, ja se asentuu Master Boot Record -osioon.



Kuva 22. Kickstart - Osiointi.

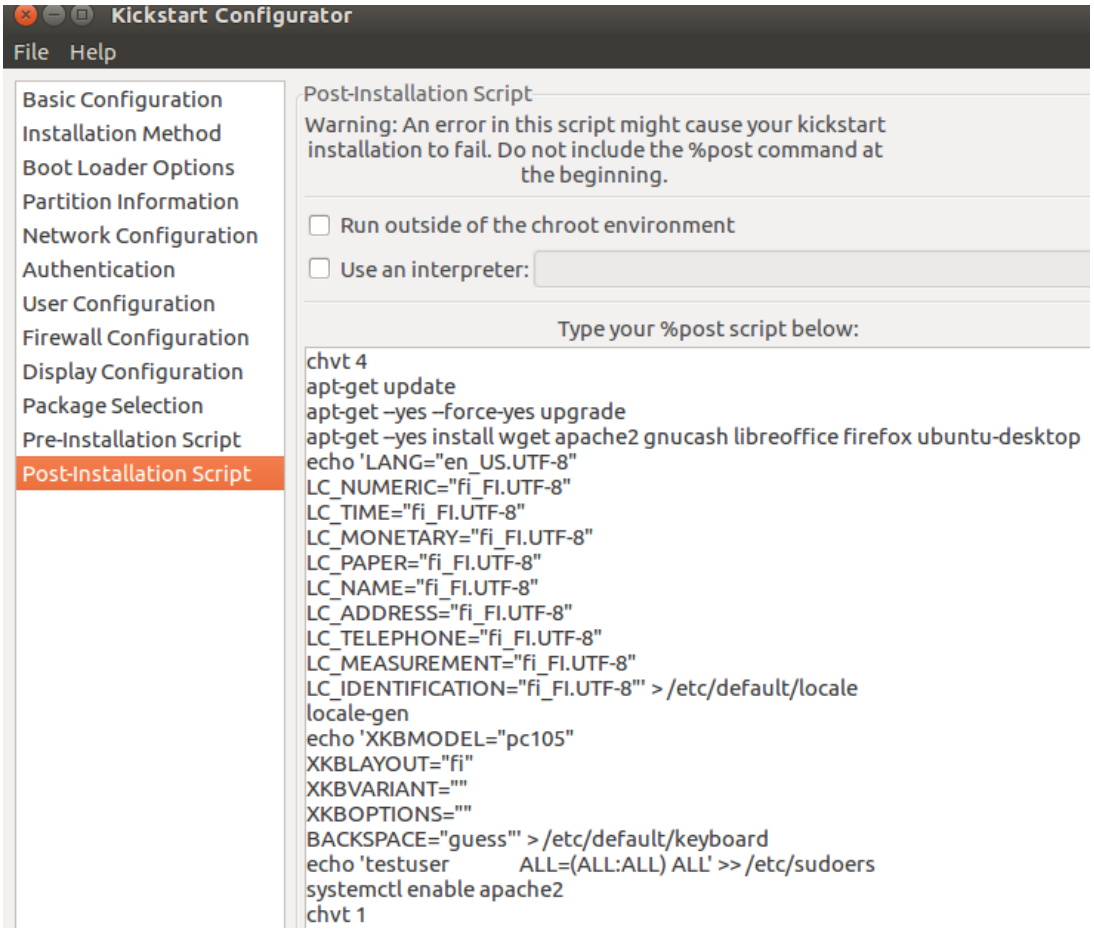
Partition Information -kohdassa valitsemme, että Master Boot Record -osio puhdistetaan, poistetaan olemassa olevat osiot kohdelevystä ja alustetaan se uudelleen. Levylle määritellään 1000MB /boot -osio ext2 -tiedostojärjestelmätyyppiä käyttäen 12GB swap -osiota, joka toimii keskusmuistin jatkeena. Loput levylle jääneestä tilasta käytetään asennettavan käyttöjärjestelmän tiedosto juurena ext4-tiedostojärjestelmässä. Boot-osioon tulee ‘grub’-tiedosto, joka on käyttöjärjestelmän käynnistysohjelma.

Emme tee muutoksia seuraaviin kahteen valikkoon, koska nämä tiedot määritellään jälkiasennuskomentosarjassa.



Kuva 23. Kickstart-käyttäjän määrittely.

Käyttäjien konfiguraatiokohdassa määrittelemme root-käyttäjän aktivoiduksi ja määrittelemme salasanan sille. Luomme myös tässä vaiheessa testikäyttäjän ja testikäyttäjän salasanan. Koska käytämme jälkiasennuskomentoja kattavasti ja muokkaamme ne suoraan Kickstartin luomaan ks.cfg-tiedostoon, ei seuraaviin valikoihin tarvitse tehdä mitään muutoksia, vaan loput komennot voi tehdä joko viimeisessä valikossa ‘post-installation script’ tai suoraan ks.cfg-tiedostossa.



Kuva 24. Kickstart-jälkiasennuskomennot.

Koska Ubuntu ei tue Kickstart-ohjelmistoa kokonaisuudessaan, asennamme osan käyttöön tulevista paketeista käyttäen asennuksen jälkeistä komentosarjaa. Komentosarjassa päivitämme ensin pakettisäiliön pakettilistat uusimpiin versioihin ja asennamme mahdolliset päivitykset järjestelmään. Päivityksen jälkeen asennetaan wget-työkalu, apache2 http-palvelin, gnucash-taloudenhallintaohjelma, Libreoffice-toimistoohjelmisto, Firefox-selain ja Ubuntu-desktop -ohjelmistopaketti, jossa asentuu 16.04 Ubuntun vakio työpöytäympäristö Unity.

Komentosarjassa määrittelemme myös käytettäväksi suomenkielisen näppäimistön ja ympäristömuuttujat käyttäen echo-komentoa konfiguraatiotiedostojen kirjoittamisessa ja locale-gen-komentoa uusien ympäristömuuttujien luomisessa. Määritellään myös testuser-käyttäjän sudo-oikeudet echo-komentoa käyttäen. systemctl enable apache2 taas aktivoi apache2 http -palvelimen.

Valmiiksi konfiguroitu ks.cfg-tiedosto näyttää miten Kickstart Configurator:ssa valitut valmiit asetukset sekä lisätyt komentosarjat asettuvat .cfg-tiedostoon. Kaikki asetukset ovat myös tehtävissä ilman konfiguraattorin apua kopioimalla liitteen 1 ks.cfg-tiedoston sisältö.

Toimiakseen FreeIPA tarvitsee myös loppukäyttäjän päähän ns. absoluuttisen FQDN-isäntänimen. Isäntänimien tulisi olla uniikkeja, jotta FreeIPA pystyy erottelemaan laitteiden välillä. Testiympäristössämme generoimme satunnaisen isäntänimen käyttämällä apuna linuxin /dev/urandom pseudosatunnaisnumerogeneraattoria, josta poimimme kymmenen merkkiä alueilta a-z ja 0-9. Tämä merkkijono asetetaan Kickstartin käyttämän ks.cfg-tiedoston muuttujaan HOSTNAME, ja tätä muuttujaa käytämme, kun syötämme komentoja isäntänimen asettamiseksi:

```
HOSTNAME=$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 10 | head -n 1)
hostnamectl set-hostname "${HOSTNAME}.testi.net"
echo "${HOSTNAME}.testi.net" > /etc/hostname
```

Esimerkkikoodi 3. Loppukäyttäjälaitteen isäntänimi.

FQDN-isäntänimeksi siis tulee generoimamme merkkijono+.testi.net.

Kickstart-ohjelmiston käyttämässä KS.cfg-tiedostossa asennetaan myös tarvittava freeipa-client-paketti.

```
apt-get --yes install htop ssh freeipa-client wget apache2 gnucash libreoffice
firefox ubuntu-desktop
```

Esimerkkikoodi 4. loppukäyttäjälaitteen ohjelmistojen asennus.

Wget-ohjelmaa käyttämällä ladataan myös palvelimelta komentosarja, jossa toteutetaan itse FreeIPA-käyttäjäksi liittyminen. Komentosarja ladataan KS.cfg:eessä luodun testikäyttäjän kotihakemistoon ja sille asetetaan suoritusoikeudet.

```
wget -P /home/testuser/ 192.168.33.250/freeipa.sh
chmod +x /home/testuser/freeipa.sh
```

Esimerkkikoodi 5. FreeIPA-asennuskomentosarjan hakeminen.

Komentosarja asetetaan suoritettavaksi järjestelmän käynnistyksen yhteydessä käyttämällä cron-ohjelmaa. Ensin kirjoitetaan /tmp/-hakemistoon väliaikainen crontab.root-tiedosto, missä on haluttava cron-konfiguraatio. Tämän jälkeen asetetaan se käyttöön käyttämällä crontab-komentoa, joka osoitetaan luotuun väliaikaistiedostoon. Tämän jälkeen väliaikaistiedosto poistetaan.

```
cat <<EOF >/tmp/crontab.root
@reboot sudo /home/testuser/freeipa.sh
EOF
crontab /tmp/crontab.root
rm /tmp/crontab.root
```

Esimerkkikoodi 6. FreeIPA-asennuskomentosarjan automaattinen suoritus.

Komentosarja itsessään sisältää tarkistuksen, että kyseessä on ensimmäinen järjestelmän käynnistys. Suoritettuaan tarkistuksen komentosarja joko synkronoi järjestelmän kellot tai tekee edellä mainitun ja aloittaa free-ipa-client-ohjelmiston asennuksen. Ensimmäinen järjestelmän käynnistys tunnistetaan, kun kokeillaan, onko testuser-käyttäjän kotihakemistossa tiedosto test.

```
#!/bin/bash
if test -f "/home/testuser/test"; then
      sudo service ntp stop
      sudo ntpd -gq
      sudo service ntp start
      touch /home/testuser/test
else
      sudo chown testuser:testuser /home/testuser/.config
      sudo chown testuser:testuser /home/testuser/.config/autostart/
      sudo service ntp stop
      sudo ntpd -gq
      sudo service ntp start
      sudo ipa-client-install --hostname=`hostname -f` --mkhomedir --
server=z210.testi.net --domain testi.net --realm TESTI.NET --principal admin -
-password=testi123 -U --force-join
      touch /home/testuser/test
      sleep 15
      sudo reboot now
```

Esimerkkikoodi 7. FreeIPA.sh-tiedoston sisältö.

Freeipa-client-install-ohjelmaan syötetään parametreinä tarvittavat tiedot. --hostname=`hostname -f` hakee järjestelmän käytössä olevan isäntänimen. --mkhomedir määrittää, että uuden käyttäjän kirjautumisen yhteydessä luodaan myös kotihakemisto. --server=z210.testi.net asettaa käytettäväksi palvelimeksi palvelimemme. --domain

testi.net asettaa käytettäväksi toimialueeksi testi.net:in ja --realm TESTI.NET-järjestelmän alueeksi TESTI.NET:in. --principal admin on hallintakäyttäjän nimi ja --password=testi123 on hallintakäyttäjän salasana. -U määrittää asennuksen sellaiseksi, että se ei pyydä käyttäjältä toimintoja tai syötteitä. --force-join pakottaa liittymään FreeIPA -asiakkaaksi mahdollisten virheiden sattuessa.

Vielä komentosarjan lopussa luodaan tiedosto test testuser -käyttäjän kotihakemistoon ja käynnistetään järjestelmä uudelleen 15 sekunnin odotuksen jälkeen komennoilla ‘sleep 15 ja sudo reboot now’.

Päädyimme liitteen 1 muotoon ks.cfg asetuksissa yrityksen ja erheen kautta. Ongelmia tuottivat vakiona näkyvät valikot, joiden poistaminen oli työlästä ja jokainen muutos vaati koko käyttöjärjestelmän uudelleenasennuksen.

```
  GNU nano 2.5.3                        File: pxelinux.cfg/default

path ubuntu-installer/amd64/boot-screens/
include ubuntu-installer/amd64/boot-screens/menu.cfg
#default ubuntu-installer/amd64/boot-screens/vesamenu.c32
prompt 0
timeout 0
label auto
        kernel ubuntu-installer/amd64/linux
        append ks=http://192.168.33.250/ubuntu/ks.cfg vga=normal initrd=ubuntu-installer/amd64/initrd.gz auto=true priority=critical

ramdisk_size=16432 root=/dev/rd/0 rw  --
```

Kuva 25. Pxelinux.cfg.

PXElinux.cfg/default on se tiedosto, jonka loppukäyttäjän laite lataa ensimmäisenä palvelimeltamme verkkokäynnistyksessä.

pxelinux.cfg/default-tiedosto kopioidaan Ubuntun asennuslevykuvasta palvelimen /var/lib/tftboot/ hakemistoon ja se muokataan ohittamaan normaalissa käynnistyksessä ilmestyvät valikot. Valikoiden ohituksen lisäksi asetamme käyttöön luomamme kickstart -tiedoston.

5.4 Loppukäyttäjä

Jotta loppukäyttäjä voi asentaa Ubuntu-käyttöjärjestelmän täytyy hänellä olla riittävän tehokas kone Ubuntulle. Laitteen kovalevy täytyy olla tyhjä, jotta loppukäyttäjän ei tarvitse vaihtaa BIOS-asetuksista käynnistysasetuksia, vaan laite yrittää ensin käynnistyä

kovalevyltä, mutta käyttöjärjestelmän puuttuessa käyttää laite PXE Server -ohjelmaa verkkokäynnistykseen, joka käynnistää Ubuntu-asennuksen.

Asennuksen käynnistyessä käyttäjän ei tarvitse tehdä mitään. Ks.cfg-tiedostossa on määritetty jo kaikki toiminnot, jotka pitää tehdä asennuksen aikana, joten ensimmäinen loppukäyttäjän suorittama toiminto uudelleen käynnistymisen jälkeen on kirjautuminen uuteen järjestelmään väliaikaistunnuksilla, jotka olemme määrittäneet Kickstart-ohjelmiston ‘user configuration’ -asetuksissa. Tässä tapauksessa käyttäjätunnus on testuser ja salasana testuser. Sisäänkirjautumisen jälkeen tulee loppukäyttäjän vielä aktivoida FreeIPA-ohjelmistoon määritellyt tunnukset. Käyttöjärjestelmä, käynnistyessään testuser-tunnuksilla, käynnistää automaattisesti komentosarjan SSH-kirjautumiselle loppukäyttäjän laitteeseen, joka taas aktivoi FreeIPA-asiakasohjelman hakemaan käyttäjän tiedot FreeIPA-palvelimelta.

```
#!/bin/bash
echo 'Welcome to ipa account activation'
read -p 'Insert your login name:' varname
ssh -oStrictHostKeyChecking=no $varname@localhost
```

Esimerkkikoodi 8. FreeIPA-käyttäjän aktivointi.

Kun loppukäyttäjä on kirjautunut tunnuksillaan, on hänen tunnuksensa aktivoitu ja hänen tulee kirjautua testuser-käyttäjältä ulos, jolloin kirjautumisruutu näyttää henkilön nimen ja mahdollistaa kirjautumisen omilla henkilökohtaisilla tunnuksilla.

6 Testaus

6.1 Käytettävyystestaus

Järjestelmän testausta on suoritettu VirtualBox-virtuaaliympäristössä tekemällä testiasennuksia. Työssämme automaattisesti asentuvaa levykuvaa testattiin virtuaaliympäristössä ja fyysisillä laitteilla. Virtuaaliympäristö oli käytössämme, sillä se nopeutti testaamista.

Virtuaaliympäristössä varmistettiin, että tekemämme muutokset levykuvaan vievät asennusprosessia eteenpäin askel askeleelta kohti valmista tuotetta, ja fyysisillä laitteilla testasimme lopullisen toimivuuden oikeassa työympäristössä. Jotta käyttöjärjestelmä saatiin täysin automaattisesti asentumaan, täytyi asennusprosessi käydä läpi yli sata kertaa.

Näihin testauskertoihin mahtui lukuisia ongelmia niin graafisen käyttöliittymän käynnistymisongelmista terminaalin käynnistymättömyyteen. Jokaiseen ongelmaan löysimme kuitenkin ratkaisun ja jokaista ratkaisua tuli testata asentamalla käyttöjärjestelmä uudelleen.

Jatkoimme tällä tavalla, kunnes lopputulos vastasi haluamaamme.

6.2 Itsearviointi

Mielestämme onnistuimme tehtävässä. Asetimme itsellemme selvät ja haasteelliset tavoitteet, jotka olemme saavuttaneet. Aluksi työmme piti koskea vain Kickstart-ohjelmistolla tehtyjä Ubuntu-levykuvia, jotka asentuvat automaattisesti. Aiheemme alkoi kuitenkin nopeasti laajentua verkon yli tapahtuvasta asennuksesta keskitettyyn käyttäjähallintaan.

Massa-asennuksissa järjestelmä ei ole kovinkaan käytännöllinen nykyisessä tilassaan, koska loppukäyttäjän laite lataa paketit Ubuntun verkossa olevasta pakettivarastosta, mikä aiheuttaa verkon tukkeutumista. Ongelman voi kuitenkin kiertää esimerkiksi ylläpitämällä paikallisia pakettivarastoja. Kun loppukäyttäjän laite lataa paketit paikallisesta pakettivarastosta, vältetään verkon tukkeutuminen.

Tätä järjestelmää ja työtä tehdessämme meidät yllätti eniten työmäärä. Siihen nähden mitä arvioimme tämän työn kestävän, teimme moninkertaisen määrän töitä. Pääsimme yhden askeleen työssä eteenpäin ja kohtasimme jälleen ongelman, joka ei ollutkaan aivan yksinkertainen korjata. Tätä työskentelytapaa käytimme satoja tunteja ennen kuin voimme todeta, että järjestelmämme toimii. Nyt järjestelmämme on laajennettavissa moneen eri suuntaan. Tämän työn pohjalle on helppo lisätä lukuisia eri käyttöjärjestelmiä, eri ohjelmistokonfiguraatioita, satoja koneita lisää, satoja käyttäjiä lisää. Vain mielikuvitus on rajana, mihin kaikkeen tämä työ voi toimia pohjana.

Lähteet

1 Proprietary Software Is Often Malware. Verkkoaineisto. <https://www.gnu.org/proprietary/proprietary.html>. Luettu 01.10.2019.

2 Richard Stallman. Verkkoaineisto. <http://www.gnu.org/gnu/gnu-linux-faq#why>. Luettu 01.10.2019.

3 Iterative Development. Verkkoaineisto. <http://wiki.c2.com/?IterativeDevelopment>. Luettu 05.09.2019.

4 Iteratiivinen kehitysmenetelmä. Kuva. <https://wiki.metropolia.fi/pages/viewpage.action?pageId=30257236>.Luettu 05.09.2019

5 Installation/SystemRequirements. Verkkoaineisto. <https://help.ubuntu.com/community/Installation/SystemRequirements#Ubuntu_Desktop_Edition>. Luettu 05.09.2019

6 Eric Hall. Verkkoaineisto. <https://web.archive.org/web/20110401192204/http://oreilly.com/catalog/coreprot/chapter/appb.html>. Luettu 01.10.2019.

7 FAI - Fully Automatic Installation. Verkkoaineisto. <http://fai-project.org/>. Luettu 01.10.2019.

8 Martins D. Okoi. Verkkoaineisto. <https://www.fossmint.com/best-linux-desktop-distros-of-2018/>. Luettu 15.08.2019. Päivitetty 05.03.2019.

9 OpenLDAP. Verkkoaineisto. <https://www.openldap.org/>. Luettu 05.09.2019.

10 Landscape-features. Verkkoaineisto. <https://landscape.canonical.com/landscape-features>. Luettu 05.09.2019.

11 What does KISS stand for. Verkkoaineisto. <https://people.apache.org/~fhanik/kiss.html> Luettu 26.09.2019.

12 Paul Rubin, David MacKenzie, Stuart Kemp. Verkkoaineisto. <http://man7.org/linux/man-pages/man1/dd.1.html> Luettu 01.10.2019.

Ks.cfg -tiedoston lopullinen sisältö

```
#Generated by Kickstart Configurator
#platform=AMD64 or Intel EM64T

#System language
lang en_US
#Language modules to install
langsupport fi_FI --default=fi_FI
#System keyboard
keyboard fi
#System mouse
mouse
#System timezone
timezone Europe/Helsinki
#Root password
rootpw testuser
#Initial user
user testuser --fullname "testuser" --password testuser
#Reboot after installation
reboot
#Use text mode install
text
#Install OS instead of upgrade
install
#Use CDROM installation media
cdrom
#System bootloader configuration
bootloader --location=mbr
#Clear the Master Boot Record
zerombr yes
#Partition clearing information
clearpart --all --initlabel
#Disk partitioning information
part /boot --fstype ext2 --size 1000
part swap --size 12000
part / --fstype ext4 --size 1 --grow
#System authorization infomation
auth  --useshadow  --enablemd5
#Firewall configuration
firewall --disabled
#Do not configure the X Window System
skipx
%post
chvt 4
HOSTNAME=$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 10 | head -n
1)
service ntp stop
ntpd -gq
service ntp start


apt-get update
```

```
apt-get --yes --force-yes upgrade
apt-get --yes install htop ssh freeipa-client wget apache2 gnucash li-
breoffice firefox ubuntu-desktop

wget -P /home/testuser/ 192.168.33.250/freeipa.sh
chmod +x /home/testuser/freeipa.sh

echo 'LANG="en_US.UTF-8"
LC_NUMERIC="fi_FI.UTF-8"
LC_TIME="fi_FI.UTF-8"
LC_MONETARY="fi_FI.UTF-8"
LC_PAPER="fi_FI.UTF-8"
LC_NAME="fi_FI.UTF-8"
LC_ADDRESS="fi_FI.UTF-8"
LC_TELEPHONE="fi_FI.UTF-8"
LC_MEASUREMENT="fi_FI.UTF-8"
LC_IDENTIFICATION="fi_FI.UTF-8"' > /etc/default/locale

locale-gen

echo 'XKBMODEL="pc105"
XKBLAYOUT="fi"
XKBVARIANT=""
XKBOPTIONS=""

BACKSPACE="guess"' > /etc/default/keyboard

systemctl enable apache2
echo 'testuser          ALL=(ALL:ALL) ALL' >> /etc/sudoers
echo 'admins          ALL=(ALL:ALL) ALL' >> /etc/sudoers


systemctl enable apache2

### LISÄTÄÄN FQDN ###
echo '192.168.33.250 z210.testi.net z210' >> /etc/hosts

### hostname  ###
HOSTNAME=$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 10 | head -n
1)
hostnamectl set-hostname "${HOSTNAME}.testi.net"
echo "${HOSTNAME}.testi.net" > /etc/hostname

### ipa ###
#ipa-client-install --hostname=`hostname -f` --mkhomedir --
server=z210.testi.net --domain testi.net --realm TESTI.NET --principal
admin --password=testi123 -U

cat <<EOF >/tmp/crontab.root
@reboot sudo /home/testuser/freeipa.sh
EOF

crontab /tmp/crontab.root
rm /tmp/crontab.root
### homedirectory create on login ###
echo 'Name: activate mkhomedir
```

```
Default: yes
Priority: 900
Session-Type: Additional
Session:
required pam_mkhomedir.so umask=0022 skel=/etc/skel' > /usr/share/pam-
configs/mkhomedir

echo 'session required pam_mkhomedir.so umask=0022 skel=/etc/skel' >>
/etc/pam.d/common-session

chvt 1
```