# Asennusohjeet

Sivusto käyttää Laravel frameworkia (<https://laravel.com/>), versio 5.1.

Laravel vaatii seuraavaa palvelimelta toimiakseen:

* PHP >= 5.5.9
* OpenSSL PHP Extension
* PDO PHP Extension
* Mbstring PHP Extension
* Tokenizer PHP Extension

Sivusto ladataan versionhallinnasta ja Laravel hyödyntää composeria ladataakseen riippuvuuksia, joten tarvitset myös git-versionhallintaohjelman paketin.

Ubuntu 14.04 ne saa ladattua käyttämällä komentoa (olettaen että palvelin käyttää nginxiä):

sudo apt-get install php5-fpm php5-cli php5-mcrypt git

Mcrypt pitää asettaa käyttöön manuaalisesti

sudo php5enmod mcrypt

sudo service php5-fpm restart

# Nginx

Siistiä osoitepolkua varten nginx-sivustokonfiguraatiolla pitää olla seuraava pätkä server-blockissa:

location / {

try\_files $uri $uri/ /index.php?$query\_string;

}

Kokonaisen nginx sivustokonfiguraation suhteen kannattaa referoida alta löytyvän linkin ohjeisiin.

Tehtyjen muutosten jälkeen nginx pitää käynnistää uudestaan:

sudo service nginx restart

Koko ohje puhtaan Laravel asennuksen asentamiseen nginxin päälle:

<https://www.digitalocean.com/community/tutorials/how-to-install-laravel-with-an-nginx-web-server-on-ubuntu-14-04>

**HUOM!** Tarkoitus ei ole seurata linkin ohjeita Laravelin asennukseen, vaan selventämään prosessia kokonaisuudessaan. Olemassaoleva Laravel asennus kloonataan muualta.

# Composer

Laravel hyödyntää composeria riippuuvuspaketinhallintaan.

Voit asentaa sen seuraavasti:

cd ~

curl -sS https://getcomposer.org/installer | php

sudo mv composer.phar /usr/local/bin/composer

Tarkemmat (vaihtoehtoiset) asennusohjeet ja lisätietoa löytyvät osoitteesta <https://getcomposer.org/download/>

# Nodejs ja npm

Sivuston JavaScript ja CSS tiedostot siistitään käyttämällä Grunt-työkalua. Grunt on riippuvainen Nodejs:stä ja sen pakettimanagerista npm:stä.

Ubuntu 14.04 paketinhallinnan versio on todennäköisesti liian vanha, joten pitää käyttää ulkoista lähdettä.

curl -sL https://deb.nodesource.com/setup | sudo bash -

sudo apt-get install nodejs npm build-essential

Varmistaaksesi, että palvelimella on tarpeeksi uusi npm versio, aja vielä npm päivitys sen itsensä kautta.

sudo npm install -g npm

Onnistunut päivitys saattaa vaatia käskyn syöttämistä kahdesti: <http://askubuntu.com/a/562432>

# Grunt

Nyt voit asentaa Grunt-komentoriviohjelman.

sudo npm install -g grunt-cli

Tarkempia ohjeita Gruntin kanssa työskentelystä: <http://gruntjs.com/getting-started>

# Sivuston valmistelu käyttöön

Sivuston koodi pitää kloonata palvelimelle GitHubista SSH kautta. SSH autentikaatio vaatii lokaalin SSH-keyn joka voidaan generoida ssh-keygenillä jos palvelimella ei vielä ole sellaista.

ssh-keygen -t rsa -b 4096 -C "Deploy key"

Lue lisää: <https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/>

Generoinnin jälkeen SSH-avain pitää pitää kopioida ja lisätä GitHubissa projektin aktiivisiin ”deploy keyhin”. Yleensä haettu SSH avain löytyy ~/.ssh/id\_rsa.pub tiedostosta.

Deploy key listaus löytyy Projektin etusivu > Settings > Deploy keys.  
Lue: <https://developer.github.com/guides/managing-deploy-keys/>

Tämän jälkeen voidaan suorittaa repositorin kloonaus. Varmista ensin, että olet oikeassa hakemistossa.

git clone git@github.com:Sonaza/mlraamattu.git

Jos kloonaus onnistui voidaan siirtyä seuraavaan vaiheeseen.

Laravel vaatii riittäviä oikeuksia kahteen hakemistoon. Varmista, että hakemistot omistaa oikea käyttäjä ja että niiden oikeudet ovat riittävät kirjoittamiselle. Esim:

chmod 755 storage

chmod 755 bootstrap/cache

Kloonaus ei lataa Laravelin tai Gruntin kolmannen osapuolen riippuvuuksia vaan ne pitää asentaa erikseen composerilla ja npm:llä.

composer install

npm install

Varmuuden vuoksi voidaan vielä ajaa composerin autoload-generaatio jolla varmistetaan, että kaikki kirjastot ovat ladattu.

composer dump-autoload

Kun riippuvuuksien lataus on valmis, voit kopioida ympäristomuuttuja-konfiguraatiotiedoston. Tiedosto sisältää palvelimen salaisia tietoja, kuten tietokannan salasanan, joten **on tärkeää, ettei ko. tiedosto joudu ulkopuolisten käsiin**. Tiedosto onkin oletuksena .gitignore listalla.

Sitten tiedostoon voidaan generoida uusi Laravelin salausavain käyttämällä artisan-työkalua.

cp .env.example .env

php artisan key:generate

Käytä jotain tekstieditoria muokataksesi uutta .env tiedostoa. Esim:

nano .env

Jos tietokanta-asetukset ovat nyt oikein, voidaan suorittaa tietokannan migraatio ja alustus artisan-työkalulla.

php artisan migrate

php artisan db:seed

Tietokantaan luodaan samalla oletusarvoinen ylläpitäjän käyttäjätili, jonka tunnukset ovat seuraavat:

**Tunnus:** admin **Salasana:** password

**HUOM!** Tämän käyttäjätilin tiedot tulisi vaihtaa oikeisiin mahdollisimman pian!

Vielä lopuksi pitää ajaa Grunt-käännösprosessi.

grunt build

Käännösprosessi optimoi JavaScript ja CSS tiedostoja /resources/assets/ hakemistosta julkiseen hakemistoon /public/. Asetukset prosessille ovat Gruntfile.js tiedostossa, joten jos tarkoitus on lisätä uusia .js tai .css tiedostoja, ne pitää kirjata sinne.

Pikaopas Gruntin kanssa työskentelystä:  
<http://blog.elenakolevska.com/using-grunt-with-laravel-and-bootstrap/>

# Sivuston päivitys GitHubiin kehityskoneelta

Git-versiohallinnan käyttö on oma asiansa, mutta helpoin tapa hyödyntää sitä on GitHubin kautta heidän tarjoamallaan [GitHub Desktop](https://desktop.github.com/) sovelluksella.

Kun halutaan tallentaa tehdyt muutokset versionhallintaan sovelluksella, pitää ensin mennä ”Uncommitted Changes” tms. välilehdelle. Vasemmassa alareunassa voidaan kirjoittaa tehdyille muutoksille yleisotsikko (commit message/summary) ja sitten tallentaa tämä commitiksi painamalla **”Commit to Master”.**

Tallennetut commitit eivät automaattisesti lataudu GitHubin palvelimelle vaan ne pitää vielä työntää (push) sinne painamalla **Sync**-painiketta yläoikealla. Sama nappula lataa myös muutokset GitHubilta paikalliselle koneelle.

Gitiä voidaan käyttää myös komentoriviltä, mutta se on monimutkaisempaa.

# Sivuston päivitys GitHubista palvelimelle

Hakemiston juuressa on kätevä shellskripti joka ajaa automaattisesti GitHubista uuden datan lataamisen ja sitten ajaa Gruntin käännösprosessin.

./deploy.sh

Operaatio voi epäonnistua jos palvelimen paikallisia tiedostoja on muokattu ja niitä yritetään päivittää gitin kautta. **On siis suositeltavaa**, että kaikki muutokset tehdään versionhallintaan ja sitten päivitetään palvelimella skriptin kautta.

Konfliktin tapauksessa palvelimen paikallinen status voidaan palauttaa viimeisimpään git-versioon käyttämällä komentoa:

git checkout

Tämä komento **poistaa** kaikki paikalliset muutokset.

# **Tietokannan varmuuskopiointi**

Tietokannan varmuuskopiointia varten on asennettu simppeli paketti. Komento sen suorittamiseen oletusparametreilla löytyy shellskriptinä hakemiston juuresta:

./dobackup.sh

Komento suorittaa mysql-dumpin tiedostoon joka tallennetaan hakemistoon /storage/app/backups/.

# Tietokannan muokkaus

Laravel hyödyntää artisan-työkalua ja migraatioita tietokannan asentamiseen. Tästä syystä on tärkeää, ettei tietokannan rakennetta mennä muuttamaan käsin, esimerkiksi phpMyAdminin kautta, sillä se rikkoo työkalun toiminnallisuuden.

Ohjeistus migraatioiden käyttöön on melko monimutkainen, joten on parasta lukea Laravelin oma dokumentaatio ja ehkä katsoa Laracasts video aiheesta:

<https://laravel.com/docs/5.1/migrations>

<https://laracasts.com/series/laravel-5-from-scratch/episodes/7>

Yksinkertaisimmillaan, jos tietokantaan pitää lisätä uusi sarake voidaan suorittaa seuraavat komennot.

php artisan make:migration:schema add\_column\_to\_users\_table -s "foobar:integer"  
php artisan migrate

Komento luo uuden migraation hakemistoon /database/migrations/ joka lisää users-taulukkoon uuden sarakkeen nimeltään foobar.

Migraation peruuttaminen uusimmasta vanhimpaan onnistuu käyttämällä seuraavaa komentoa:

php artisan migrate:rollback

Peruuttaminen poistaa muutokset ns. erissä eli kaikki kerralla tehdyt muutokset poistetaan myös kerralla, eikä yksittäisen migraation peruuttaminen ei ole mahdollista.