

Production Environment

Attention! The Production Environment is not so easy to make than the Development Environment. E.g you need a lot of knowledge of using GIT and configuring your Apache HTTP Server right way. So read this documentation carefully first you're going to do anything.

First step is install Front-end

In Ubuntu based Servers you need to run **git clone** <https://bitbucket.i4ware.fi/scm/log/login-form.git> in folder `/var/www` and after this case run **npm run build** after run **npm install** in folder `/var/www/login-form/` then start from configuring the Apache HTTP Server's Virtual Host file like `saas.i4ware.conf` correctly with like this example below:

```
<VirtualHost *:80>
    ServerAdmin info@i4ware.fi
    ServerName saas.i4ware.fi

    DirectoryIndex index.php index.html
    DocumentRoot /var/www/login-form/build

    LogLevel warn
    ErrorLog /var/log/apache2/error.log

<IfModule mod_rewrite.c>
    RewriteEngine on

    RewriteRule (.*) https://%{HTTP_HOST} [NC,L]

</IfModule>
```

Then configure `saas.i4ware-ssl.conf` like below:

```
</VirtualHost>
<VirtualHost *:443>
    ServerAdmin info@i4ware.fi
    DocumentRoot /var/www/login-form/build
    ServerName saas.i4ware.fi
    SSLEngine On
    SSLProxyEngine On
    SSLCertificateKeyFile /etc/apache2/ssl.key/server.key
    SSLCertificateFile /etc/apache2/ssl.crt/server.crt
    SSLCACertificateFile /etc/apache2/ssl.crt/WILDCARD_i4ware_fi.ca-bundle
    SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
    SSLHonorCipherOrder on
    SSLCipherSuite "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384 EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256
EECDH+aRSA+RC4 EECDH EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS"
    SetEnvIf Authorization "(.*)" HTTP_AUTHORIZATION=$1

<IfModule mod_headers.c>
Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains; preload"
Header set Referrer-Policy "no-referrer-when-downgrade"
Header always set Public-Key-Pins "pin-sha256='your-sha256-pin-1-here'; pin-sha256='your-sha256-pin-2-here'; max-age=31536000"
Header always set X-Xss-Protection "1; mode=block"
Header always set X-Content-Type-Options "nosniff"
Header set Permissions-Policy: "accelerometer=(), camera=(), geolocation=(), gyroscope=(), magnetometer=(), microphone=(), payment=(), usb=()"
Header always set X-Frame-Options "SAMEORIGIN"
Header set Cache-Control: "no-cache, no-store"
</IfModule>

    LogLevel warn
    ErrorLog /var/log/apache2/error.log

    <Directory "/var/www/login-form/build">
        #
        # Possible values for the Options directive are "None", "All",
        # or any combination of:
        #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
        #
        # Note that "MultiViews" must be named *explicitly* --- "Options All"
        # doesn't give it to you.
        #
        # The Options directive is both complicated and important. Please see
        # http://httpd.apache.org/docs/2.2/mod/core.html#options
        # for more information.
        #
        Options +Indexes +MultiViews +FollowSymLinks
        #
        # AllowOverride controls what directives may be placed in .htaccess files.
        # It can be "All", "None", or any combination of the keywords:
        #   Options FileInfo AuthConfig Limit
        #
        AllowOverride All
```

```
#
# Controls who can get stuff from this server.
#
Order allow,deny
Allow from all
</Directory>
```

```
</VirtualHost>
```

Note! Change Header always set Public-Key-Pins with this documentation: https://developer.mozilla.org/en-US/docs/Web/HTTP/Public_Key_Pinning

Note! Use this manual to make CAA Record to your DNS: <https://www.thesslstore.com/blog/what-is-caa-record-certificate-authority-authorization/>

Now is the right time to run command `service apache2 reload` after you're ran command `a2ensite` for `saas.i4ware.conf` and `saas.i4ware-ssl.conf`.

Note! Do not forget to place .htaccess file to /var/login-form/build with configuration below:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} -s [OR]
RewriteCond %{REQUEST_FILENAME} -l [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^.*$ - [NC,L]
RewriteRule ^.*$ index.html [NC,L]
```

In last case point to URL <https://saas.i4ware.fi> in your local machine with your favorite web-browser.

Second step is install Back-end

Be noticed! SMTP on your Virtual Private Server does not work with Laravel 9 without official Wildcard SSL Certificate in Postfix and Laravel 9 needs PHP 8 or never to work.

Here is correct configuration in Postfix main.cf below:

```
# TLS parameters
smtpd_tls_cert_file = /etc/apache2/ssl.crt/server.crt
smtpd_tls_key_file = /etc/apache2/ssl.key/server.key
smtpd_tls_CAfile = /etc/apache2/ssl.crt/WILDCARD_i4ware.fi.ca-bundle
smtpd_tls_security_level=may

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = i4ware.fi
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = mail.i4ware.fi, localhost, localhost.localdomain
relayhost =
mynetworks = 127.0.0.0/8
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
html_directory = /usr/share/doc/postfix/html
virtual_alias_domains =
virtual_alias_maps = proxy:mysql:/etc/postfix/mysql-virtual_forwardings.cf, mysql:/etc/postfix/mysql-virtual_email2email.cf
virtual_mailbox_domains = proxy:mysql:/etc/postfix/mysql-virtual_domains.cf
virtual_mailbox_maps = proxy:mysql:/etc/postfix/mysql-virtual_mailboxes.cf
virtual_mailbox_base = /home/vmail
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_sasl_authenticated_header = yes
smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated, reject_unauth_destination
smtpd_use_tls = yes
transport_maps = proxy:mysql:/etc/postfix/mysql-virtual_transports.cf
virtual_maildir_extended = yes
virtual_mailbox_limit_maps = proxy:mysql:/etc/postfix/mysql-virtual_mailbox_limit_maps.cf
virtual_mailbox_limit_override = yes
virtual_maildir_limit_message = "The user you are trying to reach is over quota."
virtual_overquota_bounce = yes
proxy_read_maps = $local_recipient_maps $mydestination $virtual_alias_maps $virtual_alias_domains $virtual_mailbox_maps $virtual_mailbox_domains $relay_recipient_maps
$relay_domains $canonical_maps $sender_canonical_maps $recipient_canonical_maps $relocated_maps $transport_maps $mynetworks
$virtual_mailbox_limit_maps message_size_limit = 52428800
virtual_mailbox_limit = 52428800
```

Installation of Back-end via GIT

Installation into Production Environment goes mostly similar than in Development Environment Install Instructions but to folder `/var/www`.