Web Meetings Manual

Release 1.0.0

Kopano BV

Aug 13, 2018

This document, the Web Meetings Manual, describes how to install, upgrade, configure and maintain Web Meetings with Kopano on your system. In addition various advanced configurations and integration options are covered.

CHAPTER 1

Introduction

Kopano Web Meetings is an extension package for Kopano, it enables Real-Time-Communication for all users on the Kopano platform. The additional communication features are video, voice, chat, screen-sharing, content-sharing and filetransfer. All functionality is integrated into the WebApp. A presence-service indicates if a user is logged in and ready to receive a call or a chat-message.

This document explains how a user can work with these new communication features.

Important: Although we, Kopano, try our best to keep the information in this manual as accurate as possible, we withold the right to modify this information at any time, without prior notice.

1.1 Intended Audience

This manual is intended for system administrators responsible for installing, maintaining, and supporting the Kopano deployment. We assume readers of this manual will have a thorough understanding of:

- Linux system administration concepts and tasks
- Assigning ports up HTTP servers and proxy concepts

1.2 Architecture

The Kopano Web Meetings setup combines multiple server technologies running behind a common proxy server. The frontend proxy server is an NGINX server, which passes requests and connections on to an Apache server running Webapp and the Web Meetings server.

1.3 Components

The installations of Kopano Web Meetings will require modification of the following components:

- **Kopano WebApp** (`kopano-webapp`) - The next generation collaboration web client, which offers integration with chat, presence and video conferencing.
- **Kopano Web Meetings service** (`kopano-webmeetings`) - A full featured WebRTC video conferencing interface that enables users to collaborate from any computer with an internet connection, a modern webbrowser (Iridium, Google Chrome, Firefox or Opera browser), a camera and a microphone.
- **Apache** - Serves web pages of the WebApp to the users browser, this server will need to be reconfigured to use a different port.
- **NGINX** - Serves as the frontend proxy server for both WebApp and Web Meetings and delivers tgese to the users browser from a common domain.

1.4 Protocols and Connections

All WebApp applications connect to the Kopano Server using HTTPS. The Web Meetings server maintains persisitent Web Socket connections which are upgraded from HTTPS connections. These are proxied through the frontend NGINX server.

1.4.1 Secure HTTP (HTTPS)

The Kopano Web Meetings browser Client needs to connect to the server over HTTP secured with SSL (HTTPS). All connections over the network will then be encrypted, making eavesdropping virtually impossible.

The Kopano Web Meetings server must be configured to also accept SSL connections (NGINX configuration). This requires the creation of SSL certificates. When the server certificate is created, SSL connections can be directly accepted from a client.

CHAPTER 2

Installation

2.1 System Requirements

2.1.1 Hardware Recommendations

To give an estimate on the resource use of KC we have created the table below. These are merely guidelines, giving a rough estimation on what hardware is required. In this table we assume the CPU is under low load from other applications.

Table 2.1. Minimal Hardware Recommendations

Database Size / Users	CPU (Cores)*	Memory	Harddisk	Raid level
< 5 GB / 1-25 users	2	2 GB	SATA, SAS, 7.2k	RAID 1

Important: Tuning of the server configuration and the individual software components for the specific onsite usage can drastically improve performance of your Kopano Web Meetings instance. For more than 100 users, as well as any high availability structures it is advised to seek professional engineering support.

2.1.2 Connection/bandwidth Recommendation

In order to seamlessly connect Web Meetings clients to Kopano the network latency should not be higher than 20ms. Network latencies of 200ms (500ms under exceptional circumstances) should not be exceeded in order to aid the user acceptance.

The needed bandwidth is very much depended on the individual user behaviour. Based on large scale projects we use the following key figures to calculate the minimal needed bandwidth:

For implementations with more than 100 users (with external access) we calculate with an average bandwidth utilization of “x (actual amount of users) * 8kbit/s (ISDN speed)”. In real world scenarios not all users will require exactly the same amount of bandwidth at the exact same time, which still leaves room to serve short term higher demands of single users (like requesting an attachment from the server).

Given these key figures (with +20% TCP protocol overhead) the following minimum bandwidth for Web Meetings users can be calculated: .Minimum Bandwidth Recommendations

Amount of concurrent users in meetings	Connection speed	Connection speed incl. TCP overhead
25	20000 kbit/s	24000 kbit/s

Of course these are only bare minimums and providing a higher bandwidth will increase download speeds.

2.1.3 Supported Platforms

KC and Kopano Web Meetinsg consists of a large variety of components: some back-end components that are run on Linux platforms, and components that can be installed on the computers of end-users in the form of a web browser. In this section we list the different platforms that we support.

At the start of each general release cycle (like 7.x.x) we decide which plaforms are supported. Usually that means the current release of that platform and the most recent previous release. During the major release cycle supported platforms can be added but not removed.

Please use the x86_64 or 64bit packages if 64bit hardware and OS are available. It is recommended to run on 64bit whenever possible.

Table 2.2. Supported platforms for Kopano Web Meetings back-end components

OS Release	Supported CPU Architectures
Debian 7.0 (Wheezy)	i386, x86_64
Debian 8.0 (Jessie)	x86_64
RHEL 6	i686, x86_64
RHEL 7	x86_64
SLES 12	x86_64
Ubuntu 14.04 LTS	i386, x86_64
Ubuntu 16.04 LTS	i386, x86_64
Univention 4.x	i386, x86_64

2.1.4 Dependencies

In order to build or install Kopano Web Meetings back-end components a bunch of requirements have to be met. These are the main dependencies of Kopano Web Meetings:

- **Apache** or any other webserver that supports PHP. KC is tested with Apache 2.0 and 2.2.
- **NGINX** to provide SSL and proxying HTTPS and Web Socket connections to the backend servers.

Most of these dependencies are resolved automatically by the package manager of the Linux distribution that Kopano Web Meetings is being installed on. This allows the 3rd party components used by KC to be installed and upgraded automatically through the package manager of the distribution.

The default method of deploying KC is installing the packages on one of the Linux distributions we support, allowing the 3rd party components used by KC to be installed automatically through the package manager of the distribution. In this case the 3rd party components are upgraded in a standard way according to that distribution.

2.2 Installing with the distribution’s package manager

The recommended way to install the Kopano Web Meeting is by installing Kopano prebuilt packages using the distributions package manager.

2.2.1 Adding the NGINX package repositories

You will need to add the official NGINX package repositories to your systems’s configuration. This can be done in the following way:

Important: An NGINX version >= 1.4 is required for websocket proxying!

Adding the NGINX package repositories on Debian wheezy (7.0)

Add the NGINX repository to your `/etc/apt/sources.list`

```
deb http://nginx.org/packages/debian/ wheezy nginx
```

And add the release keys

```
curl http://nginx.org/packages/keys/nginx_signing.key | sudo apt-key add -
```

Adding the NGINX package repositories on Ubuntu precise (12.04)

Add the NGINX repository to your `/etc/apt/sources.list`

```
deb http://nginx.org/packages/ubuntu/ precise nginx
```

And add the release keys

```
curl http://nginx.org/packages/keys/nginx_signing.key | sudo apt-key add -
```

Adding the NGINX package repositories on Ubuntu trusty (14.04)

Add the NGINX repository to your `/etc/apt/sources.list`

```
deb http://nginx.org/packages/ubuntu/ trusty nginx
```

And add the release keys

```
curl http://nginx.org/packages/keys/nginx_signing.key | sudo apt-key add -
```

Adding the NGINX package repositories on RHEL 7.0

Add the NGINX repository to your yum repo list

```
rpm -ivh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.
↪el7.ngx.noarch.rpm
```

2.2.2 Installing the NGINX package on Debian wheezy (7.0)

Install NGINX on Debian platforms with `apt-get`

```
apt-get install nginx
```

2.2.3 Installing the NGINX package on RHEL 7.0

Install NGINX on RHEL platforms with `yum`

```
yum -y install nginx
```

2.2.4 Installing Kopano basic server and the Web Meetings packages

The suggested Setup combines WebApp together with the Web Meetings packages on one host. For this documentation we will assume a basic KC server is already installed. Please refer to the KC Administrator Manual for setup details.

In the Web Meetings manual we are only installing the Web Meetings packages and its dependencies from the additional Kopano Extras Repository on top of your KC server. Kopano Web Meetings requires you to install the kopano-presence and kopano-webapp package from the KC distribution. Furtherore the php-curl package is required. Please ensure that these packages are installed. In case they are not yet installed please perform the following steps.

Install Web Meetings on Debian wheezy (7.0)

Install your Kopano server according to the KC Administrator Manual. To install the kopano-presence package from the KC distribution please enter the following command in the directory for your platform where you unpacked the packages included in the KC tarball.

```
dpkg -i kopano-presence_*.deb
```

To install the kopano-webapp package from the KC distribution please enter the following command in the directory for your platform where you unpacked the packages included in the KC tarball.

```
dpkg -i kopano-webapp_*.deb
```

Note: Add any additional packages as described in the KC Administrator Manual.

Install `php-curl` on Debian platforms with `apt-get`

```
apt-get install php5-curl
```

Note: Please ensure that the curl.so file gets loaded. This is normally loaded by /etc/php5/conf.d/20-curl.ini

Please unpack the kopano-extras distribution tarball for your platform and enter the unpacked directory and into the binaries subdirectory.

Install Kopano WebApp Web Meetings Plugin and Web Meetings

```
dpkg -i kopano-webapp-plugins-meetings_*.deb kopano-webmeetings_*.deb
```

Install Web Meetings on RHEL 7.0

Install your Kopano server according to the KC Administrator Manual. To install the kopano-presence package from the KC distribution please enter the following command in the directory for your platform where you unpacked the packages included in the KC tarball.

```
rpm -Uvh kopano-presence_*.rpm
```

To install the kopano-webapp package from the KC distribution please enter the following command in the directory for your platform where you unpacked the packages included in the KC tarball.

```
rpm -Uvh kopano-webapp-*.rpm
```

Note: Add any additional packages as described in the KC Administrator Manual.

On RHEL PHP curl support is built-in. On other RPM based platforms you will need to check if an additional package must be installed.

Please unpack the kopano-extras distribution tarball for your platform and enter the unpacked directory and into the binaries subdirectory.

Install Kopano WebApp Web Meetings Plugin and Web Meetings

```
rpm -Uvh kopano-webapp-plugins-meetings-*.rpm kopano-webmeetings-*.rpm
```

CHAPTER 3

Configuration

The major difference between the traditional setup of WebApp and the setup of Web Meetings is that the front end server, which provides SSL secured HTTPS connectivity, is the NGINX web server. The Apache server that hosts WebApp will need to be configured to listen to different port than 443. As an example we are choosing port 8000. This is due to the necessity to support secure web sockets for the web meetings service.

Important: On Web Meetings servers NGINX provides the frontend HTTPS connection.

3.1 Combined Setup of Web Meetings Components and WebApp on the same host

The suggested setup for single host KC systems is to combine WebApp and the Web Meeting components on the same server.

Note: Other setup variations are possible when multiple hosts are involed. Please contact Kopano Professional Services for more info.

3.1.1 Configuring Apache

Please configure Apache and WebApp as described in the KC Administrators Manual. In the following steps we will configure Apache to use a different port. Furthermore, if you are modifying an existing system, please remove the SSL configuration from Apache.

Configuring apache on Debian Wheezy (7.0) and Ubuntu

To configure Apache2 to listen on port `8000`, edit `/etc/apache2/ports.conf`

```
NameVirtualHost *:8000
Listen 127.0.0.1:8000
```

(Remove /etc/apache2/mods-enabled/ssl.conf and ssl.load if SSL was configured in Apache)

Restart apache2:

```
service apache2 restart
```

Configuring Apache on CentOS 7.0

To configure Apache2 to listen on port `8000`, edit `/etc/httpd/conf/httpd.conf` and locate the Listen setting and replace with the following:

```
NameVirtualHost *:8000
Listen 127.0.0.1:8000
```

(Please deactivate SSL if SSL was priorly configured in apache)

Restart apache2

```
service httpd restart
```

3.1.2 Configuring NGINX

Please configure NGINX by adding a `conf.d` file for your server that passes `/webapp` requests to the Apache2 server on port 8000 and everything else to the spreed.webrtc server. Below is an example for the test server spreed.example.com. Please note that Chrome, Firefox or Iridium will only remember your camera/mic access grant if the connection is a HTTPS connection. A simple HTTP setup will require granting camera/mic access every time. Here is the example `/etc/nginx/conf.d/webapp_webmeetings_ssl.conf` file, the bottom part includes the support for upgrading HTTP(S) connections to websocket. We assume for this setup that WebApp will be reached under the /webapp/ URL path and Web Meetings under the /webmeetings/ URL path.

Please enter this configuration into the existing default.conf file or remove the default.conf file. For further info please refer to http://nginx.org. Please note that the setting `ssl_ciphers` is one line and must end with a semicolon:

```
server {
        listen 212.234.234.234:443 ssl;
        server_name webapp.example.com;
        ssl on;
        ssl_certificate /etc/nginx/webapp.example.com.chained.crt;
        ssl_certificate_key /etc/nginx/webapp.example.com.key;
        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout 5m;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-CHACHA20-POLY1305:
↪ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
↪SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-
↪GCM-SHA256:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES256-
↪SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES128-SHA:ECDHE-
↪ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA:
↪DHE-RSA-AES128-SHA:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA256:DHE-DSS-AES128-
↪SHA256:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256:AES128:AES:DES-CBC3-SHA:HIGH:!
↪aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK ;
        ssl_prefer_server_ciphers on;
        #
        # ssl_dhparam require you to create a dhparam.pem, this takes a long time
        #ssl_dhparam /etc/ssl/certs/dhparam.pem;
        #
        location /webmeetings {
                proxy_pass http://localhost:8090;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection $connection_upgrade;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header Host $http_host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }
        location /webapp {
                proxy_pass http://localhost:8000;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection $connection_upgrade;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header Host $http_host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }
}
map $http_upgrade $connection_upgrade {
        default upgrade;
        '' close;
}
proxy_buffering on;
proxy_ignore_client_abort off;
proxy_redirect off;
proxy_connect_timeout 90;
proxy_send_timeout 90;
proxy_read_timeout 90;
proxy_buffer_size 4k;
proxy_buffers 4 32k;
proxy_busy_buffers_size 64k;
proxy_temp_file_write_size 64k;
proxy_next_upstream error timeout invalid_header http_502 http_503 http_504;
```

Note: Please consider, that when also proxying Z-Push connections you should raise your proxy timeouts accordingly. e.g. proxy_connect_timeout 3600; proxy_send_timeout 3600; proxy_read_timeout 3600;

Then optionally generate a `dhparam.pem` file (this is going to take a long time):

```
cd /etc/ssl/certs && openssl dhparam -out dhparam.pem 4096
```

And then uncomment the following line in your configuration file:

```
ssl_dhparam /etc/ssl/certs/dhparam.pem;
```

Important: Do not forget to create the ssl-certificate and -key as configured above.

Important: Make sure you have installed a version from the nginx repository with support for the websockets (>=1.4).

Important: Make sure that port in the `proxy_pass` setting for the `location /` matches the port that kopano-webmeetings is running on. See the following section about Configuring kopano-webmeetings.

Restart NGINX

```
/etc/init.d/nginx restart
```

3.1.3 Configuring kopano-webmeetings

Please configure kopano-webmeetings powered by Spreed.WebRTC by editing its config file at `/etc/kopano/webmeetings.cfg`. If you set up your installation according to our standard recommendation, then the only changes you need to make are to assign your own secrets to `sessionSecret`, `encryptionSecret` and `sharedsecret_secret`. Please note that the length of `sessionSecret` and `encryptionSecret` must be exactly 16, 24 or 32 bytes to select AES-128, AES-192 or AES-256.

You need to choose a free port in case the default port defined in `listen` 8090 is not free.

The relevant settings are listed below, please search for these config options in your config file and adapt appropriately.

The `turnURIs` and `turnSecret` options require a TURN server to be setup or to be present on a different machine. For a local LAN setup you should be able to test without, so the `turnURIs` parameter is optional. If you use the Kopano TURN service then you don't need to set `turnURIs` and `turnSecret` (Kopano TURN service is configured in the Kopano webmeetings WebApp plugin).

Please edit the following items in the webmeetings.cfg in addtion to what is already in there:

```
[http]
; HTTP listener in format ip:port.
listen = 127.0.0.1:8090
; Full path to directory where to find the server web assets.
root = /usr/share/kopano-webmeetings/www
...
; Use basePath if the server does not run on the root path (/) of your server.
basePath = /webmeetings
```

```
[app]
;turnURIs = turn:turnserver:port?transport=udp
;turnSecret = the-default-turn-shared-secret-do-not-keep
sessionSecret = the-default-secret-do-not-keep-me
plugin = /webapp/plugins/spreedwebrtc/php/AngularPluginWrapper.php
encryptionSecret = the-default-encryption-block-key
...
authorizeRoomJoin = true
authorizeRoomCreation = true
```

```
[users]
enabled = true
mode = sharedsecret
sharedsecret_secret = TopSecretReplaceAndEnterTheSameInTheWebAppPluginConfigPHP
```

Restart the kopano-webmeetings server:

```
service kopano-webmeetings restart
```

Important: Make sure the port configured in the `listen` property is available and not blocked by a different service. If you change the port in the `listen` property, then you will need to adapt it in the `proxy_pass` setting for `/` in the NGINX configuration.

Important: Make sure you use the same shared secret you configured in `sharedsecret_secret` in kopano-presence and in the WebApp plugin. See the respective following sections.

Important: The value of `sessionSecret` and `encryptionSecret` should each have their own individual hex encoded string value. Their length must be exactly 16, 24 or 32 bytes to select AES-128, AES-192 or AES-

256. You can generate a secret easily with `xxd -ps -l 32 -c 32 /dev/random`

3.1.4 Configuring kopano-presence

Please configure the kopano-presence service settings in `/etc/kopano/presence.cfg`.

```
plugins = spreed
```

Please configure the shared secret in `/etc/kopano/presence.cfg` in the `server_secret_key` property. This should be the same as the one configured in kopano-webmeetings.

```
server_secret_key = TopSecretReplaceAndEnterTheSameInTheWebAppPluginConfigPHP
```

You will need to add and activate the apache modules `proxy` and `proxy_http`:

```
a2enmod proxy
a2enmod proxy_http
```

Please configure a proxy pass entry in the kopano-webapp apache2 sites-available file for the kopano-presence service. Edit the file `/etc/apache2/sites-available/kopano-webapp.conf` and add the `ProxyPass` and `ProxyPassReverse` config lines below and ensure that the port number matches the port configured in `/etc/kopano/presence.cfg`.

```
Alias /webapp /usr/share/kopano-webapp
ProxyPass /webapp/presence http://localhost:1234/
ProxyPassReverse /webapp/presence http://localhost:1234/
```

Restart apache2

```
service apache2 restart
```

3.1.5 Configuring kopano-webmeetings WebApp plugin

Please configure the kopano-webmeetings powered by Spreed.WebRTC WebApp plugin. The recommended setup is that webapp and webmeetings are running behind the same NGINX host and webmeetings below the `/webmeetings/` URL path. In this case no changes are required to `PLUGIN_SPREEDWEBRTC_SPREED_URL`. In this case `PLUGIN_SPREEDWEBRTC_SPREED_DOMAIN` and `PLUGIN_SPREEDWEBRTC_WEBAPP_DOMAIN` remain empty.

Edit `/etc/kopano/webapp/config-meetings.php` and make the following changes (set `PLUGIN_SPREEDWEBRTC_USER_DEFAULT_ENABLE` to `true` and set `PLUGIN_SPREEDWEBRTC_WEBMEETINGS_SHARED_SECRET` to your secret from `webmeetings.cfg` and assign your own secrets to `PLUGIN_SPREEDWEBRTC_WEBAUTH_CODE_TOKEN_SECRET` and `PLUGIN_SPREEDWEBRTC_WEBAUTH_ACCESS_TOKEN_SECRET`):

```
DEFINE('PLUGIN_SPREEDWEBRTC_USER_DEFAULT_ENABLE', true);
```

```
DEFINE('PLUGIN_SPREEDWEBRTC_SPREED_DOMAIN', '');
```

```
DEFINE('PLUGIN_SPREEDWEBRTC_SPREED_URL', '/webmeetings/');
```

```
DEFINE('PLUGIN_SPREEDWEBRTC_WEBAPP_DOMAIN', '');
```

```
DEFINE('PLUGIN_SPREEDWEBRTC_WEBMEETINGS_SHARED_SECRET',
↪'TopSecretReplaceAndEnterTheSameInTheWebAppPluginConfigPHP');
```

```
DEFINE('PLUGIN_SPREEDWEBRTC_WEBAUTH_CODE_TOKEN_SECRET', 'the-default-secret-do-not-
↪keep-me');
```

```
DEFINE('PLUGIN_SPREEDWEBRTC_WEBAUTH_ACCESS_TOKEN_SECRET', 'the-default-secret-do-
↪not-keep-me');
```

Important: Upgrading from kopano-webmeetings WebApp plugin version 1.0 to 1.1: please take note, that the `PLUGIN_SPREEDWEBRTC_WEBAPP_URL`, `PLUGIN_SPREEDWEBRTC_WEBAUTH_CODE_TOKEN_SECRET` and `PLUGIN_SPREEDWEBRTC_WEBAUTH_ACCESS_TOKEN` settings are new and require configuration. Please backup your configuration files, install the latest configuration files from the package and merge your settings.

If WebApp and Web Meetings are not running behind a common NGINX host, you will need to make further settings to `PLUGIN_SPREEDWEBRTC_SPREED_DOMAIN` and `PLUGIN_SPREEDWEBRTC_WEBAPP_DOMAIN`.

Important: Please note that this kind of distributed setup limits the available features, such as screen sharing.

Set the Spreed Base URL, which will be your server URL. If you server doesn't yet have a DNS entry you can use an IP in the URL like https://212.234.234.234. In general these are the URLs to call from the client browser in front of NGINX.

```
DEFINE('PLUGIN_SPREEDWEBRTC_SPREED_DOMAIN', 'https://webmeetings.example.com');
```

```
DEFINE('PLUGIN_SPREEDWEBRTC_SPREED_URL', '/webmeetings/');
```

```
DEFINE('PLUGIN_SPREEDWEBRTC_WEBAPP_DOMAIN', 'https://webapp.example.com');
```

You need to configure the shared secret for the kopano-presence service in `PLUGIN_SPREEDWEBRTC_PRESENCE_SHARED_SECRET`:

```
DEFINE('PLUGIN_SPREEDWEBRTC_PRESENCE_SHARED_SECRET',
↪'TopSecretReplaceAndEnterTheSameInTheWebAppPluginConfigPHP');
```

To simplify your setup Kopano provides a TURN service. Please fill out the form at https://portal.kopano.com/content/turn-server-access-request to obtain your personal credentials. To enable the TURN service please set `PLUGIN_SPREEDWEBRTC_TURN_USE_KOPANO_SERVICE` to `true`. You enter the credentials supplied by Kopano in `PLUGIN_SPREEDWEBRTC_TURN_AUTHENTICATION_USER` and `PLUGIN_SPREEDWEBRTC_TURN_AUTHENTICATION_PASSWORD`:

```
DEFINE('PLUGIN_SPREEDWEBRTC_TURN_USE_KOPANO_SERVICE', true);
DEFINE('PLUGIN_SPREEDWEBRTC_TURN_AUTHENTICATION_URL', 'https://turnauth.kopano.com/
↪turnserverauth/');
DEFINE('PLUGIN_SPREEDWEBRTC_TURN_AUTHENTICATION_USER', 'turn-server-account-
↪provided-by-kopano');
DEFINE('PLUGIN_SPREEDWEBRTC_TURN_AUTHENTICATION_PASSWORD', 'turn-server-password-
↪provided-by-kopano');
```

Restart Apache2

```
service apache2 restart
```

Important: Make sure you have the same shared secret in PLUGIN_SPREEDWEBRTC_WEBMEETINGS_SHARED_SECRET as in `webmeetings.cfg`.

Important: Make sure you have the same shared secret in PLUGIN_SPREEDWEBRTC_PRESENCE_SHARED_SECRET as in `presence.cfg`.

In the following table we summarize the shared secrets that must be defined or generated and in which configuration files and parameters they are located.

Table 3.1. Shared secrets used between the Kopano Web Meetings components

Component	config-meetings.php	webmeetings.cfg	presence.cfg
webmeetings session secret		sessionSecret	
webmeetings encryption secret		encryption-Secret	
webmeetings shared secret	PLUGIN_SPREEDWEBRTC_ WEBMEETINGS_SHARED_SECRET	sharedsecret_secret	
presence shared secret	PLUGIN_SPREEDWEBRTC_ PRESENCE_SHARED_SECRET		server_secret_key
webauth code token shared secret	PLUGIN_SPREEDWEBRTC_ WEBAUTH_CODE_TOKEN_SECRET		
webauth access tok en shared secret	PLUGIN_SPREEDWEBRTC_ WEBAUTH_ACCESS_TOKEN_SECRET		

CHAPTER 4

Special Environments

Important: Tuning of the server configuration and the individual software components for the specific onsite usage can drastically improve performance of your Kopano Web Meetings instance. For more than 100 users, as well as any high availability structures it is advised to seek professional engineering support.

4.1 Using only Apache to host Kopano Web Meetings

Important: Especially when it comes to a lot of connections Nginx is known to scale better. If you have the chance to use Nginx for the WebSocket part, please do so.

With Apache 2.4.5 the support for proxying WebSockets was introduced. For the following configuration to work the following modules need to be enabled: proxy, proxy_http, proxy_wstunnel, headers. On a Debian/Ubuntu system this can be achieved with the following command:

```
a2enmod proxy proxy_http proxy_wstunnel headers
```

After this is done a vhost with the following configuration can be created:

```
# reverse proxy configuration for kopano-webmeetings
<Location /webmeetings>
  ProxyPass http://127.0.0.1:8090/webmeetings
  ProxyPassReverse /webmeetings
</Location>
<Location /webmeetings/ws>
  ProxyPass ws://127.0.0.1:8090/webmeetings/ws
</Location>
ProxyVia On
ProxyPreserveHost On
RequestHeader set X-Forwarded-Proto 'https' env=HTTPS
```

For kopano-presence the following vhost can be used:

```
# reverse proxy configuration for kopano-presence
<Location /webapp/presence>
  ProxyPass http://localhost:1234/
  ProxyPassReverse http://localhost:1234/
</Location>
```

CHAPTER 5

Legal Notice

Adobe, Acrobat, Acrobat Reader and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apache is a trademark of The Apache Software Foundation.

Apple, Mac, Macintosh, Mac OS, iOS, Safari and TrueType are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Blackberry is the trademark or registered trademark of BlackBerry Limited, the exclusive rights to which are expressly reserved. Kopano is not affiliated with, endorsed, sponsored, or otherwise authorized by BlackBerry Limited.

Collax is a trademark of Collax GmbH.

Debian is a registered trademark of Software in the Public Interest, Inc.

ECMAScript is the registered trademark of Ecma International.

Gentoo is a trademark of Gentoo Foundation, Inc.

Google, Android and Google Chrome are trademarks or registered trademarks of Google Inc.

IBM and PowerPC are trademarks of International Business Machines Corporation in the United States, other countries, or both.

MariaDB is a registered trademark of MariaDB Corporation AB.

Microsoft, Microsoft Internet Explorer, the Microsoft logo, the Microsoft Internet Explorer logo, Windows, Windows Phone, Office Outlook, Office 365, Exchange, Active Directory and the Microsoft Internet Explorer interfaces are trademarks or registered trademarks of Microsoft, Inc.

Mozilla, Firefox, Mozilla Firefox, the Mozilla logo, the Mozilla Firefox logo, and the Mozilla Firefox interfaces are trademarks or registered trademarks of Mozilla Corporation.

MySQL, InnoDB, JavaScript and Oracle are registered trademarks of Oracle Corporation Inc.

NDS and eDirectory are registered trademarks of Novell, Inc.

NGINX is a registered trademark of Nginx Inc. NGINX Plus is a registered trademark of Nginx Inc.

Opera and the Opera "O" are registered trademarks or trademarks of Opera Software AS in Norway, the European Union and other countries.

Postfix is a registered trademark of Wietse Zweitze Venema.

QMAIL is a trademark of Tencent Holdings Limited.

Red Hat, Red Hat Enterprise Linux, Fedora, RHCE and the Fedora Infinity Design logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

SUSE, SLES, SUSE Linux Enterprise Server, openSUSE, YaST and AppArmor are registered trademarks of SUSE LLC.

Disclaimer: Although all documentation is written and compiled with care, Kopano is not responsible for direct actions or consequences derived from using this documentation, including unclear instructions or missing information not contained in these documents.