Performance Suggestions

As SupportPro is self-hosted software ran on-premise, your server can usually be optimised to make it run faster. The following is a list of performance related suggestions.

Hardware Suggestions
Hard Drive

As SupportPro uses a file-based cache by default, performance can be greatly be improved by using SSDs with high read and write speeds instead of mechanical HDDs.

Separate Database Server

Setting up a separate server specifically for running the database can perform better than having both on the same server.

Other Server Specifications

Though there are no specific hardware requirements, we recommend a minimum of 2 GB RAM and 2 vCPU. If you anticipate the server to be under high load then we would recommend more resources.

Software Suggestions
Web Server
Linux

SupportPro works out of the box with Apache web server, however there are other web servers that perform much better albeit they're more complex to configure. We would recommend either nginx or LiteSpeed.

Windows, IIS

Read over the recommended practices for PHP on IIS.

Gzip Compression

We highly recommend to enable Gzip compression of asset files such as CSS and JavaScript. Gzip compression is widely supported

amongst internet browsers and will dramatically reduce your bandwidth usage.

Gzip compression is setup within our default .htaccess file, however you may need to enable [redacted] and [redacted] modules.

Our [redacted] nginx configuration enables gzip by default. If you're using Docker, everything is already setup for you.

We always recommend to run the latest version of PHP that fits within our [redacted].

OPcache can improve performance by storing compiled scripts in memory to avoid them having to be loaded from the file system on each page load. The default OPcache configuration is not optimal for SupportPro, so we following configuration is recommended.

It assumes you have at least 1GB RAM, you may need to set a lower memory_consumption value otherwise.

```
; Determines if Zend OPCache is enabledopcache.enable=1; The OPcache shared memory storage size.opcache.memory_consumption=512; The amount of memory for interned strings in Mbytes.opcache.interned_strings_buffer=64; The maximum number of keys (scripts) in the OPcache hash table.; Only numbers between 200 and 100000 are allowed.opcache.max_accelerated_files=32531
```

Realpath Cache

When a relative path is transformed into its real and absolute path, PHP caches the result to improve performance. SupportPro opens many files so we recommend *at least* these values:

```
; maximum memory allocated to store the resultsrealpath_cache_size=4096K; save the results for 10 minutes (600 seconds)realpath_cache_ttl=600
```

Xdebug

The xdebug extension should be disabled.

Database Configuration

MySQL (or equivalent) query caching should be enabled. This is usually enabled by default, but you may wish to increase the cache size (total data stored) or the cache limit (single query limit).

SupportPro uses the InnoDB storage engine. Please read over [redacted].

SupportPro Configuration

Redis

Switch from the file-based storage to Redis. Redis is an in-memory caching system that works much faster than reading and writing to disk. Find out how to configure [redacted].

Search

Switch from the database search driver to Meilisearch or Algolia, both of which can be faster and produce better results. Find out how to change the [redacted].

Web Sockets

Switch from XHR short polling to web sockets. XHR polling is resource intensive and increases bandwidth usage - web sockets are the more modern alternative. Find out how to enable [redacted].

Troubleshooting
MySQL: Slow Queries Log

Slow queries can affect database performance and overall server performance. The slow query log feature in MySQL enables you to log queries that exceed a predefined time limit.

To enable the slow query log in MySQL, follow these steps:

1. Login to MySQL via command line:

```
mysql -u root -p
```

2. Enable the slow query log for the current session.

```
SET GLOBAL slow_query_log = 'ON';
```

3. Change the log file location using the [redacted] system variable:

```
SET GLOBAL slow_query_log_file = '/var/log/mysql/slow-query.log';
```

4. By default in MySQL 8.0 the database will log queries which take longer than 10 seconds to run. You can change this using the [redacted] system variable. For example, the below will log queries which take longer than 5 seconds:

```
SET GLOBAL long_query_time = 5;
```

5. To verify that the slow query log is enabled, log out of mysql and then log back in. Type the following command, replacing X with a value that is greater than the long_query_time setting:

```
SELECT SLEEP(X);
```

Note, if you restart the MySQL service these changes will not persist. To persist the changes please edit your my.cnf directly.

Online URL:
https://docs.supportpro.vn/article/performance-suggestions-297.html