

**Collaboard Reliability**

RELIABILITY recommendations

|  |  |
| --- | --- |
| Title: | Sizing Recommendations |
| Abstract: | This document details the steps suggested for having a reliable installation |
| Status\* | Completed |
| Document Owner: | IBV |
| Document Author: | Gian Paolo Santopaolo |
| \*Draft, In progress, Reviewed, Authorized, Approved, Completed | |

|  |  |  |
| --- | --- | --- |
| **Document Control** | | |
| Document Distribution: | Internal: | All dev Teams |
|  | External: | Whom of interest |
| Document Authorization: | Authorizer(s): | Gian Paolo Santopaolo |

|  |  |  |  |
| --- | --- | --- | --- |
| **Change History** | | | |
| Date | Ver | Changed By | Details |
| 14/02/2022 | 1.0 | G.P Santopaolo | Document release |

Index

[Reliability 3](#_Toc127285110)

[SQL Server 3](#_Toc127285111)

[Running database on a container 3](#_Toc127285112)

[SQL Server running in a separate environment 4](#_Toc127285113)

[Choosing between the solutions 4](#_Toc127285114)

[Recommendation 4](#_Toc127285115)

[SQL Server Licensing 5](#_Toc127285116)

[Storage 5](#_Toc127285117)

[Storage options 5](#_Toc127285118)

[Local storage 5](#_Toc127285119)

[Network Attached Storage (NAS) 6](#_Toc127285120)

[Storage Area Network (SAN) 6](#_Toc127285121)

[Back-ups 6](#_Toc127285122)

[Recommendation 6](#_Toc127285123)

[Redis Cache 7](#_Toc127285124)

[Running Redis on a container cluster 7](#_Toc127285125)

[Recommendation 7](#_Toc127285126)

[Network 8](#_Toc127285127)

[Networking - Secure protocols and bandwidth 8](#_Toc127285128)

[Logs 9](#_Toc127285129)

[EFK with APM 9](#_Toc127285130)

[Grafana Loki 9](#_Toc127285131)

[Recommendations 9](#_Toc127285132)

[Security 10](#_Toc127285133)

[Firewall 10](#_Toc127285134)

[Antivirus 10](#_Toc127285135)

# Reliability

Collaboard is based on the following technologies

* SQL Server for storing the data
* File storage to store all the files
* Redis cache used as the real-time backbone

All the technologies mentioned above are critical for the application to work. If they are not fault-tolerant, the application will stop working in the case of downtime (even if only one component is down).

Disaster recovery and a backup solution are also mandatory for the database and file storage to prevent data loss.

When the application runs on the Cloud for each technology mentioned above, we strongly suggest using the native cloud provider service offering. That way, we can have fault tolerance, disaster recovery, and backup provided by the service.

In the case of an on-premises installation, we can run all (or part) the technologies mentioned above in our cluster. Still, the customer is responsible for all that matters high availability, disaster recovery, and backup.

Most importantly, the customer has to determine how important the data stored in Collaboard are and how vital the application is to be highly available.

Once the customer has those answers read, it is essential to share with Collaboard's enginers so properly plan the infrastructure setup

## SQL Server

### Running database on a container

The heart of our system is the database. This is where we store all user-generated data; we use it to track items moving through the boards, which controls our back-end services. The database is one of the most vital parts of the system; thus, it is worth paying attention to its deployment.

We use SQL Server, and we have various options for storing the data and the location of the database engine. Therefore, to make an informed decision on how to deploy SQL Server for Collaboard we need to look at the different options and the choices, we must make.

By default, when we deploy Collaboard on a containerized platform, we also deliver SQL Server. This instance runs in a container. The actual datafiles generated and used by the server are on a shared server. This is done to make sure the database files survive a container restart in case of an outage.

Installing SQL Server on a container allows us to choose the edition we run. Each edition of SQL server has other options that can be used. For a minimal installation, the Express edition might suffice but we recommend strongly against running that for the production system. In that case we can choose from one of the following options:

* Standard
* Enterprise
* EnterpriseCore

Although we supply the container with the database engine in the requested edition, it is the customer's responsibility to ensure the proper license is available and used for this. Standard, Enterprise and Enterprise all require their license from Microsoft. We do not supply this as part of our offering.

**We strongly discourage running SQL Server in the container for any production environment. We suggest using this approach only for a POC**

### SQL Server running in a separate environment

The containers need to have a connection string to connect to the database. This means we do not really care where the actual engine is. We can use it if it is reachable from the docker or the orchestrated network.

Thus, if there already is SQL Server running or if there is room on the available hardware or cloud space to have SQL Server, we can just change the connection string and have the system use that. In this case the shared folder for the database files is not needed: we leave that to the database administrator to take care of.

### Choosing between the solutions

Which solution is chosen depends on the needs and requirements of the user of Collaboard.

We have the factors that influence this decision captured in the table below.

|  |  |  |
| --- | --- | --- |
|  | **Container** | **Outside container** |
| Installation type | Trying out, prototyping, small user base | Larger userbase (> 25 users), production environments |
| Need for backup | Backup means shutting down SQL Server, thus shutting down Collaboard | Regular backups of SQL Server can be used, minimizing downtime |
| Scalability | Small installation (<25 users with little activities). Not scalable besides increasing the hardware the container runs on. | Larger installation (>25 users). SQL Server can scale both on premises and in the Cloud, including clustering servers. Scaling is virtually limitless and not bound to region |
| Stability | Local database running on one process, thus prone to downtime.  Recovery after calamity is hard to do | With failover options, stability can be in the 99.999% area. Recovery can be done if backups are taken care of and can be done without causing downtime |

### **Recommendation**

For larger production usage, we recommend using a separate instance of SQL Server. This way the backup, scaling and disaster recovery can be handled the same way all other databases are done.

The containerized solution will be adequate for test installations if the database files are backed-up regularly.

### SQL Server Licensing

When planning a Collaboard installation, it is crucial to keep in consideration that Collaboard needs to store part of its data in a SQL Server 2019 database/

There is a description on this page https://www.microsoft.com/en-us/sql-server/sql-server-2019-pricing

(make sure you download the two pdf and scroll down to pricing).

For Collaboard to be fully functional, for us is enough:

* One SQL Standard – server license ($899,00)
* One Standard – CAL ($209,00) per each server connecting to the SQL plus two more that shall be used for maintenance / administrative purpose

Of course, all the reliability/disaster recovery/fault tolerance (for both database and files) is at the customer's discretion. Depending on how valuable the information stored in Collaborad, they may opt for a higher licensing level which may dramatically affect the cost.

## Storage

Collaboard enables users to share notes, ink, and shapes. But it also allows for sharing images, videos, and documents. That last category of shareable items needs to be stored somewhere so we can retrieve it again later.

Since containers lose their contents every time they get recycled, which could happen due to a system reset or an unexpected termination of the service, we need to store the files outside of the containers.

Having the files outside of the containers also enables us to create and restore backups of these files.

Since the application works in real time it needs to be fast in storing and retrieving data.

The minimum requirement for the storage (whatever storage option will be choosen) is a storage with a comparable IOPS of a local SSD with I/O latency not higher than 1-2 ms writes, 1-2 ms read.

### Storage options

Collaboard needs fast read and writes access to the files. Multiple services and users will need to access those files, so a fast storage medium is required. Next to that, when the usage of Collaboard grows, more storage space is needed as well.

Unfortunately, it is hard to predict the amount of storage needed. Some users generate little content in the form of videos, images, or documents and use sticky notes and ink a lot. Other users do the opposite.

But by monitoring the growth of the shared storage, the total future needs can be extrapolated from the current size of the current storage size.

### Local storage

When it comes to speed, faster is always better. But as a minimum, we recommend having a drive that can read/write at around 500 MB/s. But this is just the minimum and only applies when the shared drives are on the same physical device as the container runs, or at least is near it to reduce network latency.

A good and modern SSD can deliver these speeds easily and are very reliable. Still, the user is responsible for making sure the files in the shared folders are regularly backed up and that the backup is tested. Backing up can be done without shutting down Collaboard. Restoring lost files might require a Collaboard shut down, but that depends on the situation.

### Network Attached Storage (NAS)

NAS is a way of sharing storage space that is well suited for smaller installations. It is not intended to be used for large amounts of data and large amounts of users at the same time.

It is a better solution than having a local drive for storing Collaboard files: it is more scalable and easier to use in a fault-tolerant RAID environment. The speed of the storage depends on the speed of the network, so a stable and fast network connection is recommended.

### Storage Area Network (SAN)

A SAN is a more enterprise-oriented storage solution. Although the price is generally higher, stability, scalability and fault tolerance are also much higher. If Collaboard is serving a large group of users, or if the users have lots of images, videos, or documents on their boards (or both!) SAN is the preferred way to go. This will ensure Collaboard runs at its fastest without being blocked by file and network latency.

### Back-ups

Backing up, restoring and disaster recovery are always the responsibility of the users. This needs to be considered when determining the size and sort of storage.

### **Recommendation**

For a prodctuiion environment we always recommend the storage to have the performance, in terms of IOPS, around the same numbers of a local SSD/ NVMe.

High availability, backup and disaster recovery is something the customer shall consider depending on how data stored in Collaboard and how important is the application to be highly available.

## Redis Cache

The Redis backplane is the hearth of Collaboard's real-time technology.

When running on the Cloud, we always suggest using the Cloud's provider offering for the matter.

This way, we don't have to deal with high availability and maintenance for the Redis Cluster

### Running Redis on a container cluster

The backbone of our real-time technology.

When we run in our cluster, there are at least three containers instances

### Recommendation

If the customer decides to go for a highly available service, we need to discuss the deployment of the Redis cluster

We suggest using Redis inside the Collaboard cluster only for POC or test environments.

## Network

The network infrastructure of the data center must be able to handle the proper number of parallel connections and provide the adequate bandwidth

### Networking - Secure protocols and bandwidth

The formula below is usually used to measure performance on a microservices architecture to measure the response time of remote distributed services.

|  |  |
| --- | --- |
| Variable | Definition |
| RT | Response time. The total time from the user requesting a REST API (by clicking a button, and so on) to when the message come back and the UI shows desidered content with information coming from REST response. Typically measured in seconds. |
| Payload | Total bytes sent to the SOA service, RESTfull object + http(s) header and so on |
| Ecr | Encryption for the request, if absent 1 |
| Ect | Encryption for the transport channel eg. https, if absent 1 |
| Bandwidth | Rate of transfer to and from the server. This may be asymmetrical and might represent multiple speeds if a given page is generated from multiple sources. Usually, it is averaged together to create a single bandwidth expressed in bytes per second. |
| RTT | The time it takes to round-trip, regardless of bytes transferred. Every request pays a minimum of one RTT for the page itself. Typically measured in milliseconds |
| Concurrent Requests | Number of simultaneous requests a browser will make for resource files. By default, Internet Explorer performs two concurrent requests. This setting can be adjusted but rarely is |
| Cs | Compute time on the server. This is the time it takes for code to run, retrieve data from the database, and compose the response to be sent to the browser. Measured in milliseconds |
| Cc | Compute time on the client. This is the time it takes for a client for code to run, analyze the response and render the UI |

The main challenge in calculating this formula is by getting an exact measurement of each element.

This is challenging but not impossible, there are tools, and maybe some performance counters would help measure compute times.

Ecr and Ect are important when it come to security.

The higher these values will be, the more secure your service will be, but these two variables will exponentially impact the response time, as you can see from the formula.

## Logs

We have two options for logs. A distributed transactional one and a simple one

### EFK with APM

Such power comes with a price though, the cost to pay is the minimum hardware needed to run in a production environment which is 16 vCPU, 32Gb of RAM, and 1TB 3K dedicated IOPS.

### Grafana Loki

1vCPU and 255MB of ram, we can have Grafana Loki collection plain logs (not distributed)

### Recommendations

The resources needed by the logging system is to be considered outside the sizing recommendation we are suggesting in the following chapters

# Security

If you plan to expose Collaboard to the public Internet, we strongly recommend implementing suitable security systems to detect and prevent threats

It is the customer's responsibility to make sure Collaboard will be protected against security threats

Some tooling might be useful to mention

## Firewall

IPS intrusion prevention system

IDS intrusion detection system

Anti DDoS protection

Firewall packet inspection/filtering

Any other security system that fits your requirements

## Antivirus

Antivirus scan to check files uploaded via Collaboard app.