



Manual Xpress OnPrem installation (Debian 10 Buster)

Jabra Xpress

8th November 2020



Contents

- 1. **INSTALL Debian 10 Buster** **3**
- 2. **INSTALL Xpress On Prem on Debian 10** **3**
 - 2.1 Install Docker 3
 - 2.2 Install MSSQL 5
 - 2.3 Create docker-compose.yml file 6
- 3. **Start Docker** **8**
- 4. **INFO** **10**
- 5. **Docker-Compose Up Output** **11**
- Appendix **17**
- Revision History:..... **17**

1. INSTALL Debian 10 Buster

Estimated time: 30 -35 minutes

Tested on Lenovo X220, T420, T500, HP Stream, Intel NUC



DEBIAN_Installation
_ver_1_0.docx

***Please use this installation on a laptop/PC used in a test environment.**

****Please be aware about the MSSQL licensing in case of production usage:**

https://hub.docker.com/_/microsoft-mssql-server - LICENCE paragraph.

Install Wifi Driver:



Install_Wifi_Driver.d
ocx

SSH enabled on Debian machine

2. INSTALL Xpress On Prem on Debian 10

<https://hub.docker.com/r/gnaudio/jabra-xpress-frontend> - documentation.

2.1 Install Docker

Please go to <https://docs.docker.com/install/>

select Debian

Install using the repository

Supported platforms

Docker CE is available on multiple platforms. Use the following tables to choose the best installation path for you.

DESKTOP

Platform	x86_64
Docker Desktop for Mac (macOS)	✓
Docker Desktop for Windows (Microsoft Windows 10)	✓

SERVER

Platform	x86_64 / amd64	ARM	ARM64 / AARCH64	IBM Power (ppc64le)	IBM Z (s390x)
CentOS	✓		✓		
Debian	✓	✓	✓		
Fedora	✓		✓		
Ubuntu	✓	✓	✓	✓	✓

For Debian 10 Buster steps are below:

```
$ sudo apt-get update
```

Install packages to allow apt to use a repository over HTTPS:

```
$ sudo apt-get install -y apt-transport-https ca-certificates curl gnupg2 software-properties-common
```

Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

Set up the **stable** repository:

```
$ sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/debian \
$(lsb_release -cs) \
stable"
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io -> last docker is installed
```

```
$ apt-cache madison docker-ce - list the versions available in your repo
```

In case a specific one is needed use below command as example:

```
sudo apt-get install docker-ce=5:19.03.2~3-0~debian-buster docker-ce-cli=5:19.03.2~3-0~debian-buster containerd.io
```

```
$ sudo systemctl enable docker -enable Docker
```

```
$ systemctl status docker - check status
```

- docker.service - Docker Application Container Engine

```
Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor pres>
```

```
Active: active (running) since Wed 2019-07-03 09:58:20 CEST; 47s ago
```

```
Docs: http://docs.docker.com
```

```
Main PID: 9192 (dockerd)
```

```
Tasks: 27
```

```
Memory: 63.0M
```

```
CPU: 455ms
```

```
CGroup: /system.slice/docker.service
```

```
└─9192 /usr/bin/dockerd --add-runtime oci=/usr/sbin/docker-runc
```

```
└─9200 docker-containerd --config /var/run/docker/containerd/contai
```

2.2 Install MSSQL

Quickstart: Run SQL Server container images with Docker

Link: <https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-2017&pivots=cs1-bash>

Pull and run the container image

```
$ sudo docker pull mcr.microsoft.com/mssql/server:2017-latest
```

To run the container image with Docker:

```
$ sudo docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Jabra123' -p 1433:1433 --name sql1 -d mcr.microsoft.com/mssql/server:2017-latest
```

result similar to -> 5768c9cdca1502c366d6149ff30280a35a3a63d89587fc0a7d3d91da82828aba

```
$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			

```
5768c9cdca15 mcr.microsoft.com/mssql/server:2017-latest "/opt/mssql/bin/sqls..." 44 seconds ago
Up 40 seconds 0.0.0.0:1433->1433/tcp sql1
In case not up use $ sudo docker start 5768c9cdca15
```

Optional check connection: **sudo docker exec -it sql1 "bash"** -> a # prompter available, ready for next command:

```
/opt/mssql-tools/bin/sqlcmd -S localhost -U SA -P 'Jabra123'
```

- a new prompt as '1>' displayed. exit with CTRL C
- no reason to create a testDB, so type **exit** twice.

IP 172.25.1.23 is the docker machine interface (**ip addr** command returns the PC IP). User should replace it with his machine IP.

```
$ sudo docker run -e MSSQLServerConnectionString="Server=172.25.1.23,1433;Initial Catalog=xpress;User ID=SA;Password=Jabra123;" -it gnaudio/jabra-xpress-databaseconsoletool:v4.0.15121
```

-> pulling from gnaudio/jabra-xpress

When the tool is up, select option **2** to create the database. When the operation is done and if no errors are printed, then you can continue.

```
>2
```

```
Working... Calling db.Database.Migrate();
```

```
Please wait...
```

```
Done
```

2.3 Create docker-compose.yml file

Install docker-compose by launching below command in a terminal:

```
$ sudo apt install docker-compose
```

Install docker-compose by launching below command in a terminal:

```
sudo apt install docker-compose
```

Create a **docker-compose.yml**

Note: Docker-compose.yml file should be aligned with the file posted it by Jabra at <https://hub.docker.com/r/gnaudio/jabra-xpress-frontend>

Create the file:

\$ **sudo vi /home/jabra/docker-compose.yml** and paste below content (without ---//---). Replace **172.25.1.23** with your IP (command **ip addr** provides the machine IP).

```
----- // -----  
version: '3.4'  
services:  
  mongodb:  
    image: mongo:latest  
  
  assets:  
    image: gnaudio/jabra-xpress-assets:v4.0.15632  
  
  frontend:  
    image: gnaudio/jabra-xpress-frontend:v4.0.15471  
    environment:  
    - "ENDPOINT=http://172.25.1.23:85"  
    ports:  
    - "80:80"  
  
  backend:  
    image: gnaudio/jabra-xpress-api:v4.0.15121  
    environment:  
    - "ConnectionStrings:XpressDatabaseSqlServer=Server=tcp:172.25.1.23,1433;Initial Catalog=xpress;User  
ID=SA;Password=Jabra123;"  
    - "ConnectionStrings:MongoDb=mongodb://mongodb/raw"  
    - "DefaultAnalyticsEndpoint=http://172.25.1.23:88/api/v1/Analytics"  
    - "JDODownload=http://assets/"  
    - "NetworkEndpoint=http://172.25.1.23:85/api/Xpress/Network/"  
    - "SDK:fwURL_base=http://172.25.1.23:82"  
    - "SDK:fsURL_base=http://172.25.1.23:82"  
    ports:  
    - "85:80"  
    depends_on:  
    - mongodb  
    - assets  
    - sdkbackend  
  
  sdkbackend:  
    image: gnaudio/jabra-xpress-sdkbackend:5.1.75  
    environment:
```

```
- "Storage:SDKCopyUrl=http://172.25.1.23:82"
```

```
ports:
```

```
- "82:80"
```

```
analytics:
```

```
image: gnaudio/jabra-xpress-analytics:v4.0.11419
```

```
environment:
```

```
- "ConnectionStrings:MongodbServer=mongodb://mongodb/raw"
```

```
ports:
```

```
- "88:80"
```

```
depends_on:
```

```
- mongodb
```

```
analytics-cron:
```

```
image: gnaudio/jabra-xpress-analytics-cron:v4.0.15121
```

```
environment:
```

```
- "MONGODBCONNECTIONSTRING=mongodb://mongodb/raw"
```

```
- "SQLCONNECTIONSTRING=Server=tcp:172.25.1.23,1433;Initial Catalog=xpress;User  
ID=SA;Password=Jabra123;"
```

```
- "SDKFW_BASE=http://sdkbackend"
```

```
- "SDKDC_BASE=http://sdkbackend"
```

```
depends_on:
```

```
- mongodb
```

```
- sdkbackend
```

```
-----//-----
```

```
exit vi with :wq!
```

3. Start Docker

Launch below command into a terminal

```
$ sudo docker-compose up - this command will start the processes.
```

```
-> Creating network "jabra_default" with the default driver
```

```
Pulling frontend (gnaudio/jabra-xpress-frontend:v4.0.4342)...
```

```
v4.0.4342: Pulling from gnaudio/jabra-xpress-frontend
```

```
In case CTRL+C is used, the docker is stopped. Could be started with sudo docker-compose start
```

in browser load <http://172.25.1.23>, register a new user and then login



Jabra GN JABRA XPRESS REMOTE MASS DEPLOYMENT AND HEADSET MANAGEMENT
Frontend: 4.0.4342 Backend: 4.0.4336

Register

Company name

User id

Password

Jabra GN JABRA XPRESS REMOTE MASS DEPLOYMENT AND HEADSET MANAGEMENT
Frontend: 4.0.4342 Backend: 4.0.4336

- Home
- Devices
- Inventory
- Manage
- Analytics
- Administration

Inventory

Jabra Devices	
Jabra Engage 75 Stereo	1
Jabra Evolve 65 Stereo	2
Jabra LINK 370	1
Jabra SPEAK 510	1

Select devices to add

Device filter

<input type="checkbox"/>		Jabra Engage 65 Convertible	2.1.1
<input type="checkbox"/>		Jabra Engage 65 Mono	2.1.1
<input type="checkbox"/>		Jabra Engage 65 Stereo	2.1.1
<input type="checkbox"/>		Jabra Engage 75 Convertible	2.1.1
<input type="checkbox"/>		Jabra Engage 75 Mono	2.1.1
<input type="checkbox"/>		Jabra Engage 75 Stereo	2.1.1
<input type="checkbox"/>		Device is already added	2.1.1

Check running processes:
Jabra@onprem:~\$ sudo docker ps -a

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
78f2faad351e	gnaudio/jabra-xpress-analytics:v4.0.11419	"dotnet JabraAnalyti..."	20 minutes ago	Exited	
(0) 2 minutes ago	jabra_analytics_1				
3b94e3312874	gnaudio/jabra-xpress-api:v4.0.15121	"dotnet Xpress.Api.d..."	20 minutes ago	Exited	
(0) 2 minutes ago	jabra_backend_1				
b391bfc169f6	gnaudio/jabra-xpress-analytics-cron:v4.0.15121	"bash /run.sh"	20 minutes ago	Exited	
(137) 2 minutes ago	jabra_analytics-cron_1				
f38f259d8b43	gnaudio/jabra-xpress-assets:v4.0.15632	"/docker-entrypoint...."	20 minutes ago	Exited	
(0) 2 minutes ago	jabra_assets_1				
6204cb31d64d	gnaudio/jabra-xpress-sdkbackend:5.1.1.75	"dotnet FDS.API.dll"	20 minutes ago	Exited	
(0) 2 minutes ago	jabra_sdkbackend_1				
32916ef1fad0	mongo:latest	"docker-entrypoint.s..."	20 minutes ago	Exited (0) 2	
minutes ago	jabra_mongodb_1				
4a8e5f49c3bb	gnaudio/jabra-xpress-frontend:v4.0.15471	"bash /run.sh"	20 minutes ago	Exited	
(137) 2 minutes ago	jabra_frontend_1				
0ed036cc7cc5	gnaudio/jabra-xpress-databaseconsoletool:v4.0.15121	"dotnet Xpress.Datab..."	50 minutes ago	Exited (0) 49	
minutes ago	serene_mccarthy				
cfa5d0a42e49	mcr.microsoft.com/mssql/server:2017-latest	"/opt/mssql/bin/nonr..."	53 minutes ago	Up	
53 minutes	0.0.0.0:1433->1433/tcp	sql17			

Start a docker process:

In case a docker is not up, below command will start it:

sudo docker start 5768c9cdca15 where 5768c9cdca15 is the docker ID we need to start.

Normally the MSSQL docker should be started manually.

4. INFO

```
linux-thfq:/home/onprem # docker system df
```

TYPE	TOTAL	ACTIVE	SIZE	RECLAIMABLE
Images	9	9	3.663GB	235.5MB (6%)
Containers	9	8	77.52MB	0B (0%)
Local Volumes	24	3	2.52GB	2.205GB (87%)
Build Cache	0	0	0B	0B

```
linux-thfq:/home/onprem # docker system info
```

Containers: 9

Running: 8

Paused: 0

Stopped: 1
Images: 9
Server Version: 18.09.6
Storage Driver: btrfs
Build Version: Btrfs v4.19.1
Library Version: 102
Kernel Version: 4.12.14-lp151.27-default
Operating System: openSUSE Leap 15.1
OSType: linux
Architecture: x86_64
CPUs: 4
Total Memory: 7.482GiB

5. Docker-Compose Up Output

```
jabra@whiteHP:~$ sudo docker-compose up
Creating network "jabra_default" with the default driver
Pulling mongodb (mongo:latest)...
latest: Pulling from library/mongo
f22ccc0b8772: Pull complete
3cf8fb62ba5f: Pull complete
e80c964ece6a: Pull complete
329e632c35b3: Pull complete
3e1bd1325a3d: Pull complete
4aa6e3d64a4a: Pull complete
035bca87b778: Pull complete
874e4e43cb00: Pull complete
08cb97662b8b: Pull complete
f623ce2ba1e1: Pull complete
f100ac278196: Pull complete
b16ea696739f: Pull complete
Digest: sha256:d54fb51a493131117a8b85a2d46ed7d64a2cb79607fb3ce350d1722c0cafc812
Status: Downloaded newer image for mongo:latest
Pulling assets (gnaudio/jabra-xpress-assets:v4.0.15632)...
v4.0.15632: Pulling from gnaudio/jabra-xpress-assets
852e50cd189d: Pull complete
a29b129f4109: Pull complete
b3ddf1fa5595: Pull complete
c5df295936d3: Pull complete
232bf38931fc: Pull complete
d5000eb579b9: Pull complete
Digest: sha256:22ec9ddadad48c948b1cb2fa44e691931b8afc7aba223f144e1406598aab87e2
Status: Downloaded newer image for gnaudio/jabra-xpress-assets:v4.0.15632
```

Pulling frontend (gnaudio/jabra-xpress-frontend:v4.0.15471)...
v4.0.15471: Pulling from gnaudio/jabra-xpress-frontend
f2aa67a397c4: Pull complete
3c091c23e29d: Pull complete
4a99993b8636: Pull complete
d922e80cccc8: Pull complete
7578a4bf5d9f: Pull complete
677c98b6c4e8: Pull complete
Digest: sha256:488701d38fad3fcb16e5670ea7aafbc87a8eae101c7d10f987ffa9ad07bc0938
Status: Downloaded newer image for gnaudio/jabra-xpress-frontend:v4.0.15471
Pulling sdkbackend (gnaudio/jabra-xpress-sdkbackend:5.1.75)...
5.1.75: Pulling from gnaudio/jabra-xpress-sdkbackend
bb79b6b2107f: Pull complete
97805e17b1a2: Pull complete
b76800c2f59e: Pull complete
70eaf40cbbac: Pull complete
7ed8558c7d5d: Pull complete
80c73fac07e8: Pull complete
03ae36c1395e: Pull complete
903d1a301187: Pull complete
Digest: sha256:e46ab5f1cf0d6b6d436c59076b3df06158eb64425a1aa118b0d455950af038e9
Status: Downloaded newer image for gnaudio/jabra-xpress-sdkbackend:5.1.75
Pulling backend (gnaudio/jabra-xpress-api:v4.0.15121)...
v4.0.15121: Pulling from gnaudio/jabra-xpress-api
171857c49d0f: Already exists
419640447d26: Already exists
61e52f862619: Already exists
46c0e7a9f394: Already exists
d29a4c703ace: Already exists
77fae72da4a6: Already exists
da1fcc2bd21a: Already exists
b500667c3c57: Already exists
0e70fe3b4227: Pull complete
Digest: sha256:0ac8e7f9e6680562610183ee1982fc72e3211a2a4390f21be577d5c4c666c4e5
Status: Downloaded newer image for gnaudio/jabra-xpress-api:v4.0.15121
Pulling analytics (gnaudio/jabra-xpress-analytics:v4.0.11419)...
v4.0.11419: Pulling from gnaudio/jabra-xpress-analytics
5d9821c94847: Pull complete
a610eae58dfc: Pull complete
a40e0eb9f140: Pull complete
6135796bbd7a: Pull complete
dfcda5543ce6: Pull complete
799d5545e923: Pull complete
1e60f72e5148: Pull complete

```
d540998822c6: Pull complete
9909a68b77b1: Pull complete
Digest: sha256:9ed62efd2d3ae718c1e1df6cd9190cc10c74ceb5599d5bed9386e865b99fec22
Status: Downloaded newer image for gnaudio/jabra-xpress-analytics:v4.0.11419
Pulling analytics-cron (gnaudio/jabra-xpress-analytics-cron:v4.0.15121)...
v4.0.15121: Pulling from gnaudio/jabra-xpress-analytics-cron
171857c49d0f: Already exists
419640447d26: Already exists
61e52f862619: Already exists
46c0e7a9f394: Already exists
d29a4c703ace: Already exists
77fae72da4a6: Already exists
da1fcc2bd21a: Already exists
b500667c3c57: Already exists
c9981f31a7e4: Pull complete
fa2692a78483: Pull complete
2ec149c07906: Pull complete
0dcf2c8dbbf7: Pull complete
3ffd36554c8f: Pull complete
1e6b5caff81f: Pull complete
0371a1cac6fd: Pull complete
5b3599d5d3bf: Pull complete
7410f44b8be1: Pull complete
Digest: sha256:f130559292a0ab52a0dd93e2b229b8fa30c1607edb5d6923dc1892135e11f2b1
Status: Downloaded newer image for gnaudio/jabra-xpress-analytics-cron:v4.0.15121
Creating jabra_frontend_1 ... done
Creating jabra_mongodb_1 ... done
Creating jabra_sdkbackend_1 ... done
Creating jabra_assets_1 ... done
Creating jabra_analytics-cron_1 ... done
Creating jabra_analytics_1 ... done
Creating jabra_backend_1 ... done
Attaching to jabra_assets_1, jabra_sdkbackend_1, jabra_frontend_1, jabra_mongodb_1, jabra_analytics-
cron_1, jabra_analytics_1, jabra_backend_1
analytics-cron_1 | Starting run.sh... Version 06-07-2018 12:42
analytics-cron_1 | /app/appsettings.json before:
analytics-cron_1 | /run.sh: line 1: #!/bin/sh: No such file or directory
analytics-cron_1 | {
analytics-cron_1 |   "MongoDbConnectionString": "mongodb://localhost",
analytics-cron_1 |   "MongoDbCatalog": "raw", // Initial DB catalog the cron should connect to
analytics-cron_1 |   "KeepDocumentsDays": 30, // Set TTL in days. Defaults to the minimum value of 30
analytics-cron_1 |   "KeepDocumentsMonth": 1 // Legacy variable, not used anymore. Kept for support of
onprem instances.
analytics-cron_1 | }/app/appsettings.json after:
```

```

analytics-cron_1 | {
analytics-cron_1 | "MongoDbConnectionString": "mongodb://mongodb/raw",
analytics-cron_1 | "MongoDbCatalog": "raw", // Initial DB catalog the cron should connect to
analytics-cron_1 | "KeepDocumentsDays": 30, // Set TTL in days. Defaults to the minimum value of 30
analytics-cron_1 | "KeepDocumentsMonth": 1 // Legacy variable, not used anymore. Kept for support of
onprem instances.
analytics-cron_1 | }Starting nginx
analytics-cron_1 | Starting cron
assets_1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform
configuration
assets_1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
assets_1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
assets_1 | 10-listen-on-ipv6-by-default.sh: Getting the checksum of /etc/nginx/conf.d/default.conf
assets_1 | 10-listen-on-ipv6-by-default.sh: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
assets_1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
assets_1 | /docker-entrypoint.sh: Configuration complete; ready for start up
frontend_1 | Running run.sh...
frontend_1 | Filename:
frontend_1 | /usr/share/nginx/html/main.1363e5d26bf683f3a3a5.js
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.690+00:00"},"s":"I", "c":"CONTROL", "id":23285,
"ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols
'none'"}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.731+00:00"},"s":"W", "c":"ASIO", "id":22601,
"ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.731+00:00"},"s":"I", "c":"NETWORK", "id":4648601,
"ctx":"main","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer,
tcpFastOpenClient, and tcpFastOpenQueueSize."}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.735+00:00"},"s":"I", "c":"STORAGE", "id":4615611,
"ctx":"initandlisten","msg":"MongoDB
starting","attr":{"pid":1,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"32916ef1fad0"}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.736+00:00"},"s":"I", "c":"CONTROL", "id":23403,
"ctx":"initandlisten","msg":"Build
Info","attr":{"buildInfo":{"version":"4.4.2","gitVersion":"15e73dc5738d2278b688f8929aee605fe4279b0e","
opensslVersion":"OpenSSL 1.1.1 11 Sep
2018","modules":[],"allocator":"tcmalloc","environment":{"distmod":"ubuntu1804","distarch":"x86_64","t
arget_arch":"x86_64"}}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.736+00:00"},"s":"I", "c":"CONTROL", "id":51765,
"ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Ubuntu","version":"18.04"}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.736+00:00"},"s":"I", "c":"CONTROL", "id":21951,
"ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{"net":{"bindIp":"*"}}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.754+00:00"},"s":"I", "c":"STORAGE", "id":22297,
"ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage
engine. See http://dochub.mongodb.org/core/prodnotes-filesystem","tags":["startupWarnings"]}

```

```
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:00.754+00:00"},"s":"I", "c":"STORAGE", "id":22315,
"ctx":"initandlisten","msg":"Opening
WiredTiger","attr":{"config":{"create,cache_size=1386M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress,compact_progress],}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.315+00:00"},"s":"I", "c":"STORAGE", "id":22430,
"ctx":"initandlisten","msg":"WiredTiger
message","attr":{"message":"[1606499463:315432][1:0x7fe22cc5bac0], txn-recover: [WT_VERB_RECOVERY
| WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.315+00:00"},"s":"I", "c":"STORAGE", "id":22430,
"ctx":"initandlisten","msg":"WiredTiger
message","attr":{"message":"[1606499463:315705][1:0x7fe22cc5bac0], txn-recover: [WT_VERB_RECOVERY
| WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.329+00:00"},"s":"I", "c":"STORAGE", "id":4795906,
"ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMillis":2575}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.329+00:00"},"s":"I", "c":"RECOVERY", "id":23987,
"ctx":"initandlisten","msg":"WiredTiger
recoveryTimestamp","attr":{"recoveryTimestamp":{"timestamp":{"t":0,"i":0}}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.362+00:00"},"s":"I", "c":"STORAGE", "id":4366408,
"ctx":"initandlisten","msg":"No table logging settings modifications are required for existing WiredTiger
tables","attr":{"loggingEnabled":true}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.370+00:00"},"s":"I", "c":"STORAGE", "id":22262,
"ctx":"initandlisten","msg":"Timestamp monitor starting"}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.377+00:00"},"s":"W", "c":"CONTROL", "id":22120,
"ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data
and configuration is unrestricted","tags":["startupWarnings"]}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.398+00:00"},"s":"I", "c":"STORAGE", "id":20320,
"ctx":"initandlisten","msg":"createCollection","attr":{"namespace":"admin.system.version","uuidDisposition":"provided","uuid":{"uuid":{"$uuid":"62b24dd3-ef46-4125-8045-6793e6603d5c"},"options":{"uuid":{"$uuid":"62b24dd3-ef46-4125-8045-6793e6603d5c"}}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.431+00:00"},"s":"I", "c":"INDEX", "id":20345,
"ctx":"initandlisten","msg":"Index build: done
building","attr":{"buildUUID":null,"namespace":"admin.system.version","index":"_id_","commitTimestamp":{"$timestamp":{"t":0,"i":0}}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.440+00:00"},"s":"I", "c":"COMMAND", "id":20459,
"ctx":"initandlisten","msg":"Setting featureCompatibilityVersion","attr":{"newVersion":"4.4"}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.448+00:00"},"s":"I", "c":"STORAGE", "id":20536,
"ctx":"initandlisten","msg":"Flow Control is enabled on this deployment"}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.458+00:00"},"s":"I", "c":"STORAGE", "id":20320,
"ctx":"initandlisten","msg":"createCollection","attr":{"namespace":"local.startup_log","uuidDisposition":"generated","uuid":{"uuid":{"$uuid":"88e86896-570f-4257-bc06-5e895f2e50d5"},"options":{"capped":true,"size":10485760}}}}
```

```
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.497+00:00"},"s":"I", "c":"INDEX", "id":20345,
"ctx":"initandlisten","msg":"Index build: done
building","attr":{"buildUUID":null,"namespace":"local.startup_log","index":"_id_","commitTimestamp":{"t":0,"i":0}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.498+00:00"},"s":"I", "c":"FTDC", "id":20625,
"ctx":"initandlisten","msg":"Initializing full-time diagnostic data
capture","attr":{"dataDirectory":"/data/db/diagnostic.data"}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.525+00:00"},"s":"I", "c":"NETWORK", "id":23015,
"ctx":"listener","msg":"Listening on","attr":{"address":"/tmp/mongodb-27017.sock"}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.525+00:00"},"s":"I", "c":"NETWORK", "id":23015,
"ctx":"listener","msg":"Listening on","attr":{"address":"0.0.0.0"}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.525+00:00"},"s":"I", "c":"NETWORK", "id":23016,
"ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.574+00:00"},"s":"I", "c":"STORAGE", "id":20320,
"ctx":"LogicalSessionCacheRefresh","msg":"createCollection","attr":{"namespace":"config.system.sessions",
"uuidDisposition":"generated","uuid":{"uuid":{"$uuid":"8e366da3-aad1-490d-b9ed-13081d886cca"}},
"options":{}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.575+00:00"},"s":"I", "c":"CONTROL", "id":20712,
"ctx":"LogicalSessionCacheReap","msg":"Sessions collection is not set up; waiting until next sessions reap
interval","attr":{"error":"NamespaceNotFound: config.system.sessions does not exist"}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.613+00:00"},"s":"I", "c":"INDEX", "id":20345,
"ctx":"LogicalSessionCacheRefresh","msg":"Index build: done
building","attr":{"buildUUID":null,"namespace":"config.system.sessions","index":"_id_","commitTimestamp":{"t":0,"i":0}}}
mongodb_1 | {"t":{"$date":"2020-11-27T17:51:03.614+00:00"},"s":"I", "c":"INDEX", "id":20345,
"ctx":"LogicalSessionCacheRefresh","msg":"Index build: done
building","attr":{"buildUUID":null,"namespace":"config.system.sessions","index":"IsidTTLIndex","commitTimestamp":{"t":0,"i":0}}}
sdkbackend_1 | warn: Microsoft.AspNetCore.DataProtection.Repositories.FileSystemXmlRepository[60]
sdkbackend_1 | Storing keys in a directory '/root/.aspnet/DataProtection-Keys' that may not be
persisted outside of the container. Protected data will be unavailable when container is destroyed.
sdkbackend_1 | warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
sdkbackend_1 | No XML encryptor configured. Key {aa27900c-252c-4248-b06f-abb963469c2d} may
be persisted to storage in unencrypted form.
sdkbackend_1 | Hosting environment: Production
sdkbackend_1 | Content root path: /app
sdkbackend_1 | Now listening on: http://[::]:80
sdkbackend_1 | Application started. Press Ctrl+C to shut down.
```




Appendix

Revision History:

Maintenance interval: _____ (Half-yearly / Yearly / Bi-yearly).

Rev	Date	Editor	Reviewer(s)	Description
A	YYYY.MM.DD	Name	Name	Initial version

This document is the property of GN Audio A/S and is to be treated as confidential by the party whom it has been submitted by GN Audio A/S and it is not to be disclosed to any third party without specific prior written permission of GN Audio A/S.
© 2018 GN Audio A/S. All rights reserved