Installing Cassandra on Windows

Important! Cassandra is not supported on Windows!

We do not provide support for Cassandra installations on Windows as those installations are not natively supported by Apache. **We strongly recommend using Linux deployments.** The information below is provided as a courtesy only.

One option for running Cassandra 4.0.3 on Windows is Docker. See Install Docker Desktop on Windows for official installation instructions.

Additional resources:

- https://hub.docker.com/_/cassandra/ – official Cassandra Docker image
- https://docs.docker.com/ – Docker documentation
- https://github.com/apache/cassandra/blob/cassandra-4.0/NEWS.txt – Cassandra's migration notes

Docker Cassandra basics

This section provides the basics for starting a Docker Cassandra 4.0.3 container, for testing purposes only. For production deployment, we recommend generating a custom Docker image as described in the section Custom Docker Cassandra image for Teamwork Cloud.

Start an instance of the official Cassandra 4.0.3 Docker image by executing this command:

```
docker run --name cassandra4_test –p 9042:9042 –v e:\cassandra\data:/var/lib/cassandra -d cassandra:4.0.3
```

Parameter	Value	Description
--name	cassandra4_test	User-defined container instance name. This is an optional parameter. The name must be unique and cannot be the same as the official Docker image name.
-v	e:/cassandra/data/: /var/lib/cassandra	Mount an external drive to a path in the Docker container instance. */var/lib/cassandra* is the default Cassandra path for data and commit log storage. An external drive must be mounted for permanent data storage.
-d		Detach and run in the background.
-p	9042:9042	Publish port 9042 to Windows host. 9042 is the native client port.
[image name]: version	cassandra:4.0.3	Pull the official Apache Cassandra 4.0.3 Docker image for this running instance.

If this is a new Docker installation, the first **docker run** execution downloads the Cassandra image to the local system.

To access logs:

```
docker logs <container name or ID>
```

To access nodetool (or other tools):

```
docker exec -it <container name or ID> nodetool status
```

Newer Java versions require the **-Dcom.sun.jndi.rmiURLParsing=legacy** option. If the "Malformed IPv6 address" error occurs, use:

```
docker exec -it <container name or ID> nodetool -Dcom.sun.jndi.rmiURLParsing=legacy status
```

Docker instances are ephemeral by design. If an instance terminates or is deleted, all data, logs, and configuration files modified inside the instance are lost.

Custom Docker Cassandra image for Teamwork Cloud

Production deployment of Cassandra 4.0.3 on Windows requires modifying system configurations. Therefore, a custom Docker image must be created with the configuration changes. This section covers setting up and running a custom Docker image. Details of the configuration changes are discussed in section Cassandra configuration.

1. Download and extract the ZIP file package into an empty working directory.
2. Execute the following command to build the custom Docker image named *cassandra4_twc*:

```
docker build –t cassandra4_twc:latest .
```

3. Start an instance of the custom Docker Cassandra 4.0.3 image you just built:

```
docker run –p 9042:9042 –v d:\data:/data –v e:\log\commitlog:/logs/commitlog -d cassandra4_twc:latest
```

> The data and commit log directories are mounted separately, to different paths.

With the custom Docker Cassandra instance running and mounted to permanent storage, the database is ready to connect to your Teamwork Cloud installation.

If more configuration changes are needed, the running instance has to be terminated and a new Docker image needs to be built. Edit the configuration file from the working directory in Step 1. Proceed with steps 2-3 to redeploy the Cassandra node on Windows.

Cassandra configuration

This section is a summary of the configuration changes to optimize Docker Cassandra deployment for Teamwork Cloud. If additional configuration changes are needed, the Docker Cassandra image has to be rebuilt. All configuration files are located in */etc/cassandra*.

> Synchronize the CPU clocks on all Cassandra cluster nodes. Otherwise, you may encounter issues when creating an empty Cassandra cluster.

Modification of the *cassandra.yaml* file

If the new Cassandra 4.x node is going to access existing data from a Cassandra 3.x database, please make a note of the previous **cluster_name** and **num_tokens** settings. These parameters must match in the new Cassandra 4.0.3 *cassandra.yaml* configuration.

The following parameters are for controlling thresholds to ensure that the data being sent is processed properly.

```
commitlog_segment_size_in_mb: 192
read_request_timeout_in_ms: 1800000
range_request_timeout_in_ms: 1800000
write_request_timeout_in_ms: 1800000
cas_contention_timeout_in_ms: 1000
truncate_request_timeout_in_ms: 1800000
request_timeout_in_ms: 1800000
batch_size_warn_threshold_in_kb: 3000
batch_size_fail_threshold_in_kb: 5000
```

Ensure that the default commit log size is 8GB (recommended).

```
commitlog_total_space_in_mb: 8192
```

Point the data to the appropriate locations.

```
data_file_directories:
- /data/data
commitlog_directory: /logs/commitlog
hints_directory: /data/hints
saved_caches_directory: /data/saved_caches
```

JVM configuration for Cassandra Node

To change the amount of RAM used by Cassandra, activate the -Xms4G and -Xmx4g properties in *jvm-server.options* (remove leading '#') and specify their values. By default, Cassandra uses no more than 8 GB of RAM.

- In the *jvm11-server.options* and *jvm8-server.options* files, CMS Settings are disabled.
- In the *jvm11-server.options* and *jvm8-server.options* files, the parameters below are activated

```
-XX:+UseG1GC
-XX:MaxGCPauseMillis=500
```

- In the *jvm11-server.options* and *jvm8-server.options* files, the two parameters below are set to the physical CPU core count (the values of both parameters should be the same):

```
-XX:ParallelGCThreads=16
-XX:ConcGCThreads=16
```

- In the *jvm8-server.options* file, disable all GC logging options.
- In the logback.xml file, disable debug.log by commenting out "<appender-ref ref="ASYNCDEBUGLOG" />"