Migration strategies

On this page

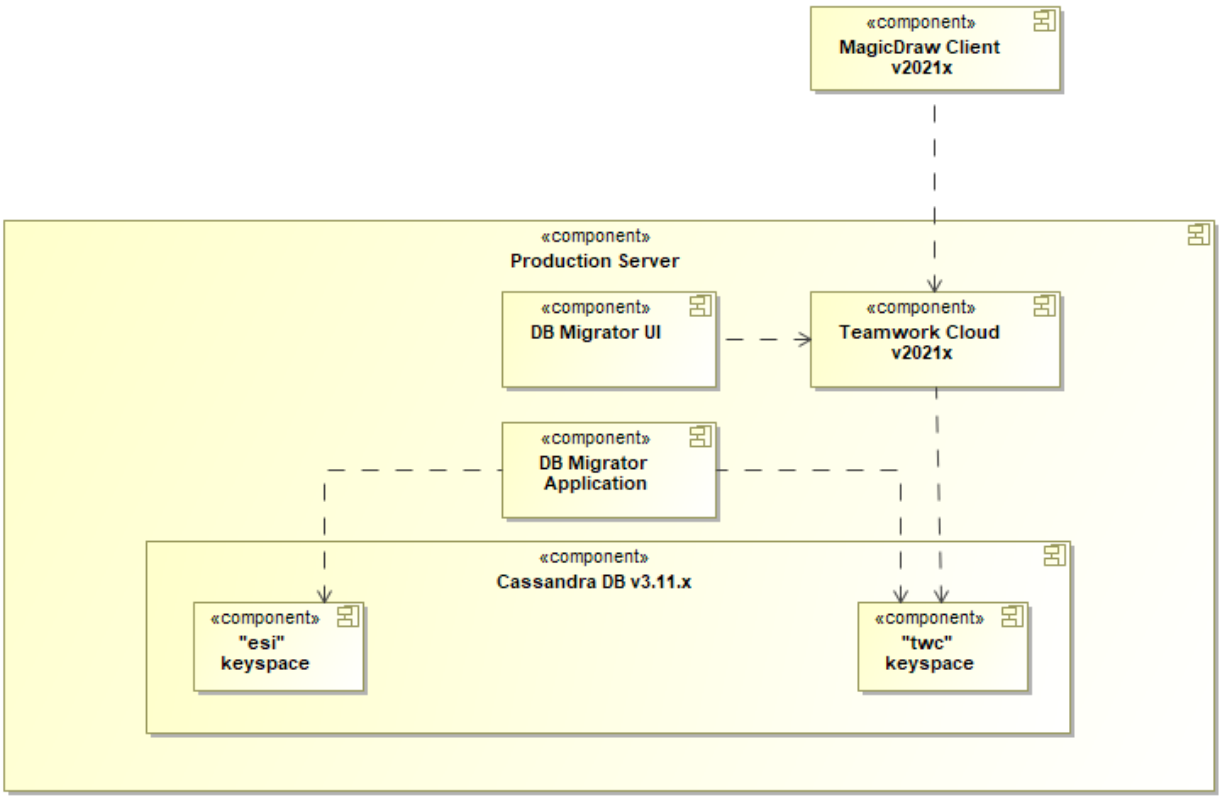
Incremental migration prerequisites

Incremental migration approaches require having both versions of Teamwork Cloud servers running - 19.0 (SP3 or SP4) and 2021x Refresh1.

Please contact your sales executive to get a temporal license key.

All in one go

You can migrate the entire Teamwork Cloud database in one go. However, there will be downtime in production while data is being migrated from the source database ("esi" keyspace) to the target database ("twc" keyspace). This strategy may not be applicable if the source database is of significant size. During migration, the source database size increases by ~30-40% (the amount used by the new "twc" keyspace). However, after full migration, the old "esi" keyspace in the source database can be effectively dropped using the data manager command which leaves the database only with the "twc" keyspace.



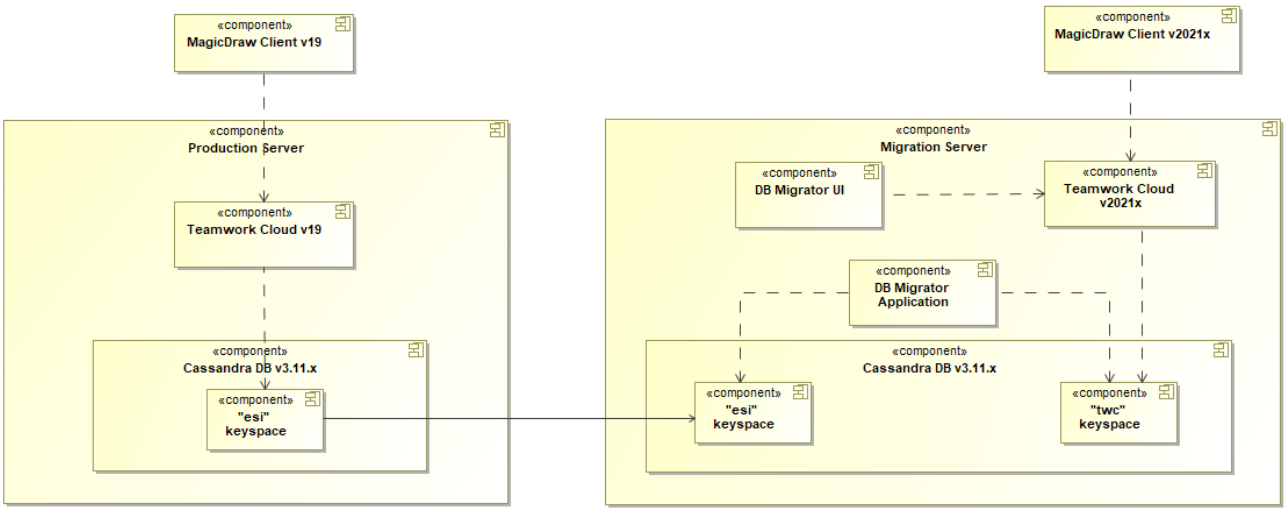
The strategy of migrating the Teamwork Cloud database all in one go.

To migrate the Teamwork Cloud database all in one go

1. Uninstall Teamwork Cloud 19.0 (SP3 or SP4).
2. Install Teamwork Cloud 2021x Refresh1.
3. Launch the migration tools and perform full database migration.
4. Drop the "esi" keyspace in the source database using the Data Manager.

Incremental migration using separate servers

You can migrate the Teamwork Cloud database incrementally by using separate database servers. This way users can continue working on the production server while the migration is in progress with a minor downtime for the final migration step.



The strategy of an incremental Teamwork Cloud database migration using separate database servers.

To migrate the Teamwork Cloud database when a single-node Cassandra cluster is used

---

1. Install Cassandra 3.11.x on an external (migration) server.
2. Remove data and commit logs on the migration server by executing the following commands:

```
nodetool drain
systemctl stop cassandra
rm –fr /data/data/*
rm -fr /logs/commitlog/*
```

3. Perform the initial move of data from the production (source) database to the migration server without stopping work on the production server. To do this, execute the following command on the production server:

```
rsync –av /data/data/ user@target:/data/data/
```

> In the command above, **user** is the OS user of the migration server and **target** is the migration server address.

4. Carry out the final data move from the production database to the migration server while temporarily stopping work on the production server. After you finish, the migration server will have the entire data of the production database (the "esi" keyspace). To do this, execute the following commands on the production server:

```
nodetool drain
systemctl stop cassandra
rsync –av /data/data/ user@target:/data/data/ --delete
```

> In the commands above, **user** is the OS user of the migration server and **target** is the migration server address.

5. Install Teamwork Cloud 2021x Refresh1 on the migration server. The "twc" keyspace will be created.
6. Launch the migration tools on the migration server and perform initial database migration. Users can continue working on the production server as usual.

When performing initial database migration, make sure that the **Remove resources after successful migration** check-box is cleared in the resource migration wizard. For more information, see step 6 in Migrating database.

7. Carry out the final production database migration while stopping work on the production server:
   a. Stop Teamwork Cloud services on both the production and migration servers.
   b. Stop Cassandra services on both the production and migration servers by executing the following commands:

```
nodetool drain
systemctl stop cassandra
```

   c. Move the newly created data from the production database to the migration server. To do this, execute the following command:

```
rsync –av /data/data/esi/ user@target:/data/data/esi/ --delete
```

In the command above, **user** is the OS user of the migration server and **target** is the migration server address.

   d. Restart Cassandra and Teamwork Cloud 2021x Refresh1 on the migration server.
   e. Perform the final migration of the newly created production data in the migration server.
   f. Make a backup of the "esi" keyspace on the production server.
   g. Drop the "esi" keyspace and remove the "esi" directory with commit logs by executing the following commands in the production server:

```
nodetool drain
sudo systemctl stop cassandra
rm –fr /data/data/esi/*
rm -fr /logs/commitlog/*
```

   h. When migration on the migration server is finished, drop the "esi" keyspace in the migration server database by executing the following c ommand:

```
nodetool flush
cqlsh
drop keyspace esi;
exit
nodetool clearsnapshot -- esi
```

   i. Move migrated data from the migration server to the production server by executing the following command on the migration server:

```
rsync –av /data/data/ user@source:/data/data/
```

In the command above, **user** is the OS user of the production server and **source** is the production server address.

8. Uninstall Teamwork Cloud 19.0 (SP3 or SP4) on the production server.
9. Install Teamwork Cloud 2021x Refresh1 on the production server.

Prerequisites

The following strategy is only applicable when the Cassandra node count in the cluster equals the Replication Factor (RF), for example, 3 Cassandra nodes with RF=3.

To migrate the Teamwork Cloud database when the production server is a multi-node Cassandra cluster and the migration server is a single-node Cassandra cluster

---

1. Install Cassandra 3.11.x and Teamwork Cloud 19.0 (SP4 or SP3) on an external (migration) server to create the v19 database schema.
2. Uninstall Teamwork Cloud 19.0 (SP4 or SP3) on the migration server.
3. Stop Cassandra service on the migration server.

4. Perform the initial data move from the production (source) database to the migration server without stopping work on the production server. To do this, execute the following command in any cluster node on the production server:

```
rsync –av /data/data/esi user@target:/data/data/esi-new
```

> In the command above, **user** is the OS user of the migration server, **target** is the migration server address, and **esi-new** is a temporary production data location on the migration server.

5. Carry out the final data move of the production database while stopping work on the production server:

   Ensure that the user has the write permission in the esi-new directory.

   a. Stop Teamwork Cloud services on both the production and migration servers.
   b. Execute the following commands on any of the production database Cassandra cluster nodes:

```
nodetool repair
nodetool flush
nodetool drain
```

   c. Stop Cassandra services on all cluster nodes of the production server.
   d. Copy the remaining newly created data from the production server to the migration server. To do this, execute the following command on any of the production cluster nodes:

```
rsync –av /data/data/esi/ user@target:/data/data/esi-new/ --delete
```

> In the commands above, **user** is the OS user of the migration server, **target** is the migration server address, and **esi-new** is a temporary production data location on the migration server.

6. Restart Teamwork Cloud and Cassandra on the production server so that users could continue their work.
7. Slip-stream the data from the "esi-new" keyspace to the "esi" keyspace on the migration server by running the convert_esi.sh script. Now the migration server has the current data of the production database (the "esi" keyspace).
8. Install Teamwork Cloud 2021x Refresh1 on the migration server. The "twc" keyspace will be created.
9. Launch the migration tools and perform initial database migration. Users can continue working on the production server as usual.

> When performing initial database migration, make sure that the **Remove resources after successful migration** check-box is cleared in the resource migration wizard. For more information, see step 6 in Migrating database.

10. Carry out the final move of the data that was newly created since the previous migration iteration from the production server:

    a. Stop Teamwork Cloud services on both the production and migration servers.
    b. Execute the following commands on any of the production database Cassandra cluster nodes:

```
nodetool repair
nodetool flush
nodetool drain
```

    c. Stop Cassandra services on all cluster nodes of the production and migration servers.
    d. Copy the newly created data from the production server to the migration server while stopping work on the production server. To do this, execute the following command on the production server:

```
rsync –av /data/data/esi/ user@target:/data/data/esi-new/ --delete
```

> In the commands above, **user** is the OS user of the migration server, **target** is the migration server address, and **esi-new** is a temporary production data location on the migration server.

    e. Re-run the convert_esi.sh script on the migration server to slip-stream data.
    f. Restart Cassandra and Teamwork Cloud 2021x Refresh1 on the migration server.
    g. Perform the final migration of the newly created data moved from the production server.

> When performing final migration, make sure that the **Remove resources after successful migration** check-box is selected in the resource migration wizard. For more information, see step 6 in Migrating database.

11. On the production server, start Cassandra nodes and make a backup of the "esi" keyspace.
12. Drop the "esi" keyspace and remove the "esi" directory with commit logs by executing the following commands:

```
nodetool flush
cqlsh
drop keyspace esi;
exit
nodetool clearsnapshot -- esi
```

With the Cassandra service stopped, the "esi" directory and commit logs need to be removed on each of the production nodes by executing the following commands:

```
rm -fr /data/data/esi/
rm –fr /logs/commitlog/*
```

13. Uninstall Teamwork Cloud 19.0 (SP3 or SP4) on all the nodes of the production server.
14. Install Teamwork Cloud 2021x Refresh1 on the production server to create the "twc" keyspace and initial database schema on all cluster nodes.
15. With Teamwork Cloud, Authentication server, and Web Application Platform being shut down, execute the commands below, then stop Cassandra.

```
nodetool flush
nodetool drain
```

16. Move the migrated data from the migration server "twc" keyspace to every node of the production cluster by executing the following command on the migration server:

```
rsync –av /data/data/twc/ user@target:/data/data/twc-new/
```

In the command above, **user** is the OS user of the production server, **target** is the production server address, and **twc-new** is a temporary production data location on the production server.

17. Start Cassandra and run the create_migration_table.cql script by executing the commands below. If needed, enter the required credentials to ensure a successful cqlsh connection. This will create an empty table on the production server.

Ensure that the user has the write permission in the twc-new directory.

```
cqlsh
SOURCE '/opt/create_migration_table.cql'
exit
```

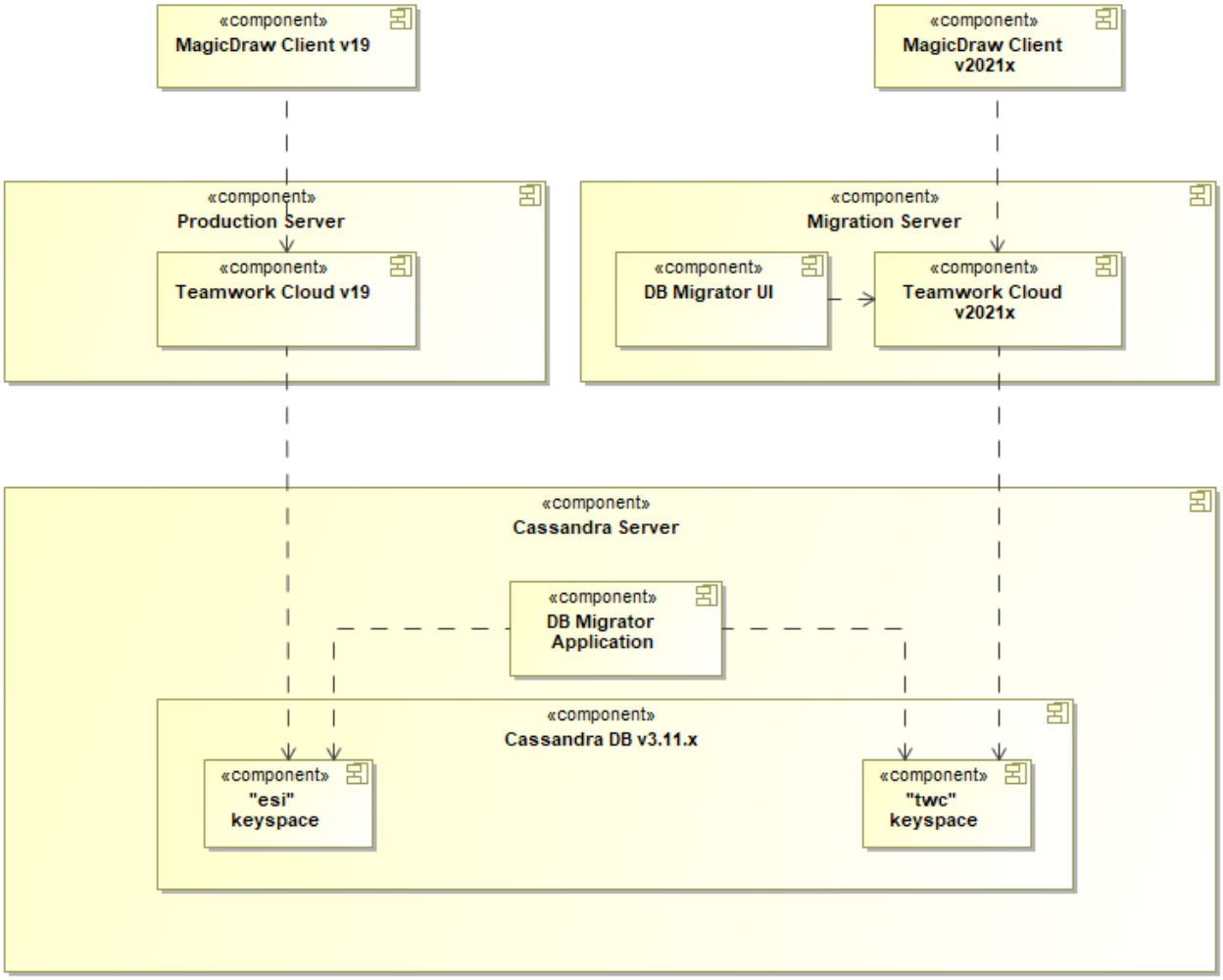
18. Stop Cassandra on all nodes of the production server including Teamwork Cloud by executing the following commands:

```
nodetool flush
nodetool drain
systemctl stop Cassandra
```

19. Slip-stream data from the "twc-new" keyspace to the "twc" keyspace in every node of the production cluster by running the convert_twc.sh script.
20. Power up Cassandra and Teamwork Cloud 2021x Refresh1 services in the production cluster.

Incremental migration using separate servers and one Cassandra instance

You can migrate the Teamwork Cloud database incrementally with Teamwork Cloud 19.0 (SP3 or SP4) and Teamwork Cloud 2021x Refresh1 running on different server machines, but pointing to the same Cassandra instance. This strategy requires no downtime, but you may experience reduced Teamwork Cloud 19.0 (SP3 or SP4) performance during migration due to additional Cassandra load.



The strategy of an incremental Teamwork Cloud database migration when Teamwork Cloud 19.0 series and Teamwork Cloud 2021x series are on separate servers but point to the same Cassandra instance.

To migrate the Teamwork Cloud database incrementally using separate servers and one Cassandra instance

---

1. Install Teamwork Cloud 2021x Refresh1 on a separate migration server.
2. Point Teamwork Cloud 2021x Refresh1 to the Cassandra database server holding Teamwork Cloud 19.0 (SP3 or SP4) data by specifying an appropriate Cassandra seed node address in the Teamwork Cloud **application.conf** file.
3. Deploy the Teamwork Cloud migrator application on the migration server and point it to the Cassandra database server holding Teamwork Cloud 19.0 (SP3 or SP4) data by specifying an appropriate Cassandra seed node address in the database migrator **application.conf** file.
4. Launch the migration tools and perform initial database migration.

   ⚠ When performing initial database migration, make sure that the **Remove resources after successful migration** check-box is cleared in the resource migration wizard. For more information, see step 6 in Migrating database.
5. At a chosen point in time, stop the Teamwork Cloud 19.0 (SP3 or SP4) service to prohibit the creation of new data and execute the final migration iteration.

   ⚠ When performing the final migration iteration, make sure that the **Remove resources after successful migration** check-box is selected in the resource migration wizard. For more information, see step 6 in Migrating database.
6. Once the migration is finished, drop the "esi" keyspace by using the Data Manager.
7. Uninstall Teamwork Cloud 19.0 (SP3 or SP4) on the production server and install Teamwork Cloud 2021x Refresh1. At this point, the migration server is no longer needed.