Cross-cluster resource synchronization

On this page

Teamwork Cloud offers a one-way resource synchronization functionality allowing you to move resources from one Teamwork Cloud server to another and periodically update them. This is a convenient way to collaborate with disconnected teams or departments and share not only resources but their entire history as well. Therefore you can use resource synchronization as an alternative to exchanging .mdzip files and then updating server projects with local changes. In addition, you can perform a one-time synchronization operation to offload resources to another server for archiving.

> ⚠
> - Cross-cluster resource synchronization works only in one direction - from the source to the target server. Synchronized resources on the target server are read-only and cannot be edited to ensure version consistency.
> - The synchronization feature checks project uniqueness by Project ID, Resource ID, and Project Name.

Using resource synchronization for collaborating

You can periodically synchronize Teamwork Cloud resources from the source server to the target server so that other users can use synchronized projects in other projects stored on the target server.
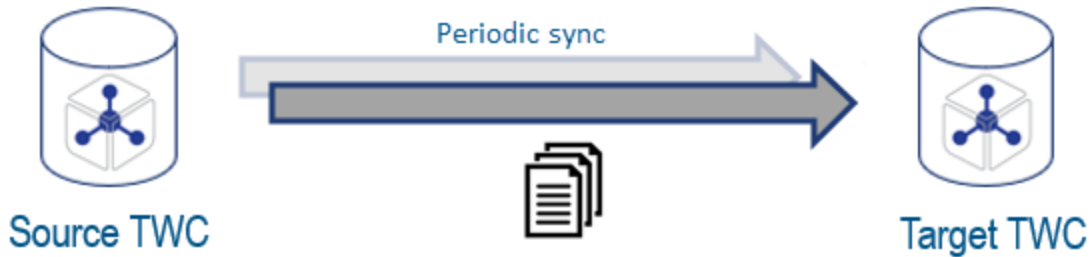
Setting up a periodic synchronization operation to share resources that can be reused on the target server.

To use resource sync for collaboration between disconnected Teamwork Cloud sites

1. On the target Teamwork Cloud server, create a remote resource configuration to synchronize selected resources from the source to the target server:
   a. Connect to the source Teamwork Cloud server and select the resources and/or categories you want to synchronize.
   b. Select the category on the target Teamwork Cloud server where you want the synced resources to be stored.
   c. Select one of the following synchronization triggers: on every commit, on commit with a specific tag, or manual synchronization
2. Use the synchronized resources in your projects stored on the target server.
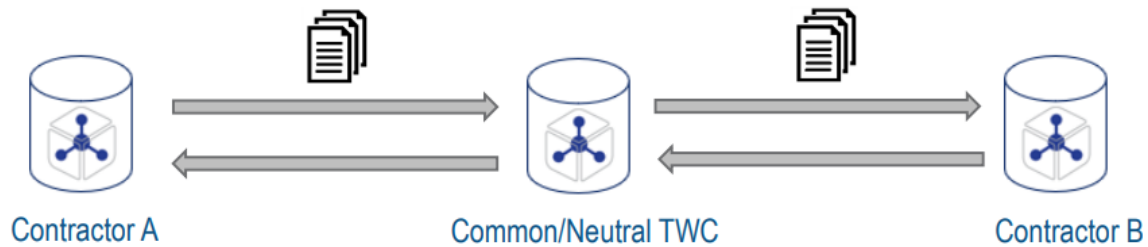3. Update synchronized resources when new project versions become available.

> ⚠
> - The source server determines all of the resource's metadata. Therefore, if you change the name of a synchronized project on the source server, the name also changes on the target server on subsequent synchronization iteration. If you change the name of a synchronized project on the target server, the name changes back to the one on the source server during synchronization.
> - If you move a synchronized project to a different category on the source server, the project continues synchronizing as before. If you do so on the target server, on subsequent synchronization iteration, the synchronized project returns to its original category specified in the synchronization configuration.
> - However, if a whole category is being synchronized and you remove a resource from that category, this resource is not synchronized anymore. Only the remaining resources in the category are synchronized further.
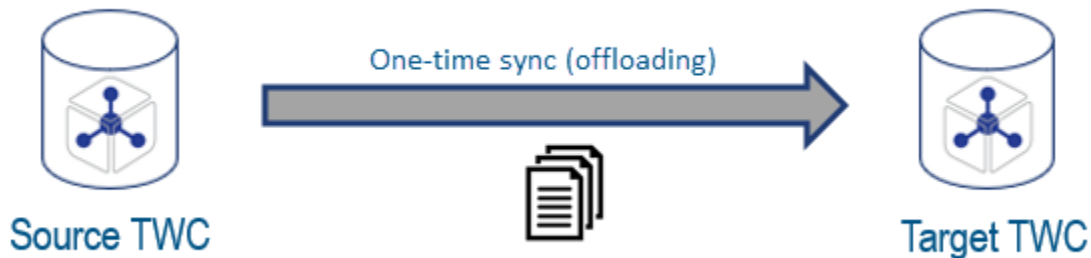
Multi-hop synchronization

In the case of external collaboration between separate entities, direct network connectivity is usually not permitted between the two parties due to IT-level security constraints. As a remedy to this issue, the cross-cluster sync feature can be set up to support the so-called multi-hop deployment topology. That is, a separate Teamwork Cloud instance can be spun up in a neutral environment outside of the involved organizations' network, so the need to connect the two together is eliminated. Instead, resources can be shared and synced between the source and target in multiple hops through the common/neutral Teamwork Cloud instance in the middle.

Synchronizing resources in multiple hops via a common or neutral environment.

Using resource synchronization for resource offloading

If there are dormant recourses in your Teamwork Cloud server, you can move them to another server for archiving to free up space on the main server (e.g., production server).

Setting up a one-time synchronization operation to move resources to another server.

To use resource sync for moving resources to another Teamwork Cloud server

---

1. On the target Teamwork Cloud server (e.g., archive server), create a remote resource configuration for a single time synchronization operation:
   a. Connect to the source Teamwork Cloud server (e.g., production server) and select the resources and/or categories you want to synchronize.
   b. Select the category on the target Teamwork Cloud server where you want the synced resources to be stored.
   c. Select one the **On every new commit** synchronization trigger.
2. Use Data Manager to remove the synced resources from the source server to free up server space.
3. Remove the remote resource configuration created in step 1.

Removing the read-only flag from offloaded resources

If you have used resource synchronization to offload resources to a remote server, these resources are read-only and you can no longer edit them. However, there is a way to remove the read-only flag from synchronized resources and make them active again.

> Do not remove the read-only flag from periodically synchronized resources or categories. This will cause version inconsistencies after subsequent synchronizations.

The read-only flag removal is recommended only if resources or categories were moved to another server as a result of a one-time synchronization

To remove the read-only flag from offloaded resources
operation and you want to make them active again.

- Run Data Manager and execute the following command:

```
-rrof, --remove-read-only-flag <-resourceID=<value1>,<value2> --categoryID=<value1>,<value2>>
```

> **Example**
> datamanager -rrof --resourceID=a4c498c9-f125-453c-b9cb-3d5a6e675714 --categoryID=a1d569ea-fe6a-402f-8a25-aef6b932de48

Configuring TLS encrypted connection for cross-cluster synchronization

Both source and target server must have the same values set in Webapp:

```
twc.client.keystore.type
twc.client.keystore.path
twc.client.keystore.password
twc.client.tls=true
```

It is not necessary to set the same port for TLS in Webapp. However, keep in mind that during login, you need to set a port when adding a resource. Otherwise, it will use the **target** server's TLS port to log in to the **source** server, where it has a different TLS port.

To enable encrypted synchronization in Teamwork Cloud, append the following parameter to the end of *application.conf*:

```
esi.server.sync.tls-enabled = true
```

Both Teamwork Cloud servers must have a certificate signed by a trusted Certificate Authority (CA). You can use self-signed certificates, but you may need to add the certificate to your JVM keystore and OS path as well.