Admin Tools

On this page

The Admin Tools package is a set of Linux utility scripts that facilitate server-side tasks. You can use these scripts to manage services (start, stop, check status), archive log and configuration files, and check the installation environment. Furthermore, you can generate shortcuts to log and configuration files in a central location for troubleshooting and monitoring configuration changes.

> **Where can I find the Admin Tools package?**
>
> The Admin Tools package is included with 2024x and newer releases.
>
> You can also download *utilities_2024x.zip* and use it on older releases (19.0 to 2022x).

Creating shortcuts

The **add-twc-shortcuts.sh** script creates symbolic links to key log and configuration files so everything can be accessed in one locaton. For older versions of Teamwork Cloud, the *Utilities/AdminTools* path is created in the installation path and all utilities are copied over as well. A system path entry is set optionally in /etc/profile.d (applied at the next login session). Elevated permission is required to execute this script.

A custom install path can be specified by providing the path as a commandline argument:

```
./add-twc-shortcuts /path/to/install
```

Shortcut files mapping for add-twc-shortcuts.sh

Shortcut name	Original filename
conf-auth	authserver.properties
conf-twc	application.conf
conf-webapp	webappplatform.properties
log-auth	authentication.log
log-cassandra	system.log
log-twc	server.log
log-webapp	webapp.log or web-app.log
log-webapp-admin	admin.log

Managing services

You can use this set of utility scripts to manage all Teamwork Cloud services. Elevated permission is required to execute these scripts.

Note that if add-twc-shortcuts.sh was executed and system path is set, these scripts can be executed anywhere.

Script name	Details
twcstats.sh [network]	Check the status of all services. If service is running, output will display: ACTIVE. When run as root/sudo, service log will be generated for each failed or stopped service. If Cassandra is installed and running on the same server, nodetool status will be used to report on Cassandra mode. Add "network" option to check port bindings of each service.

twc-start.sh [db]	Start all Teamwork Cloud services that are installed on the system. Services will only be started if they are installed and not active. Add “db” option to start Cassandra service (only performed if Cassandra is installed on same server).
twc-stop.sh [db]	Stop all Teamwork Cloud services that are installed on the system. Only services that are currently running will be stopped. Add “db” option to stop Cassandra service as well. A nodetool drain will be performed first before service is stopped (only performed if Cassandra is installed on same server).
cassandra-stop.sh	Perform proper nodetool drain before stopping the Cassandra service. This script is called by twc-stop.sh.

Archiving configuration files

Use the **backup-twc-configs.sh** script to create a backup package of the current *certificate/keystore* and all Teamwork Cloud configuration files. The default backup path is */opt/local/TWC_Backup*. Elevated permission may be required to access files and archive path.

Comparing configuration files

Use **compare-twc-config.sh** to compare the current configuration against a reference set of configurations. Reference set can be either be from a backup or the no-install package.

Checking installation environment

The **check-env.sh** script is a tool for checking the installation environment on a server. This script does not require elevated permission to execute. See the table below for a list of checks performed.

check-env.sh
TWC version installed
Java version(s) installed
OS and version
Execution permissions on /tmp & /home
SELinux setting
fapolicyd service status
FIPS mode setting
Cassandra version installed
Cassandra /data & /logs mount types

Trimming webapps

Use the **webapp-trim.sh** script to control the number of webapps to run on the server. Run with the minimum set of webapps for troubleshooting. Optimize system resources by running only the webapps that users need.

Command	Option	Details
webapp-trim.sh		Only run webapps in the $RUN_LIST specified inside the script file.
	min	For troubleshooting and fast startup, set to minimum webapps needed.
	restore	Restore server to run all the webapps.
	help	Display list of options for script.