Backup/Restore

This documentation is here to help you set up your backup. Be careful with this, it's just a guide and you will probably want to backup more things.

If you installed Tuleap on a virtual environment and you are able to use snapshots, the simplest backup solution is to suspend tuleap services and then make a snapshot.

Otherwise here are some tips to backup and restore your Tuleap infrastructure.

Backup for RPM Deployment on RHEL/Alma/Rocky

SUSPEND SERVICES

Depending on services you use, you will want to stop them before the backup (It should guarantee you a consistent backup):

```
$ systemctl stop nginx
$ systemctl stop httpd
$ systemctl stop tuleap
$ su - gitolite -c "gitolite writable @all off 'Backup in progress'"
```

Don't forget to restart services once the backup is done.

DATABASE BACKUP

Tuleap main database is `tuleap`, but additionnal databases can be used for plugins. To show them use:

```
$ mysql -u codendiadm -p -e "show databases;"
```

You can write a script to backup each database independently or use `mysqldump` to backup all databases:

```
$ mysqldump -u codendiadm -p --all-databases > mybackup.sql
```

TULEAP DATA PATHS BACKUP

You need to save the following directories:

- /etc/nginx
- /etc/redis
- /etc/tuleap
- /var/lib/gitolite
- /var/lib/tuleap

Some directories might not exist depending on your configuration (plugins installed or not).

Backup for Docker/Compose Deployment

SUSPEND SERVICES

Depending on the services you use, you will want to stop them before the process, it should guarantee you a consistent backup:

```
$ cd /path_to/tuleap_compose
$ docker compose stop tuleap
```

Don't forget to restart services once the backup is done.

DATABASE BACKUP

Tuleap main database is `tuleap`, but additionnal databases can be used for plugins. To show them, use:

```
$ docker exec SQL_CONTAINER /usr/bin/mysql -u codendiadm -p -e "show databases;"
```

You can write a script to backup each database independently or use `mysqldump` to backup all databases:

```
# Backup
docker exec SQL_CONTAINER /usr/bin/mysqldump -u codendiadm -p --all-databases > sqldump_all_tuleap.sql

# Restore (use root user for an empty server)
cat sqldump_all_tuleap.sql | docker exec -i SQL_CONTAINER /usr/bin/mysql -u root -p
```

TULEAP DOCKER DATA PATHS

For a docker deployment, all data is stored in `/data` directory mapped from the container: backup all its content from the docker host.

Restore Tuleap

As only data were backed up, you first need a Tuleap server to restore them. It can be your old server or a new server you have just installed following the installation guide. Then you will need to:

- suspend all services
- restore databases (from a sqldump is the safest method to ensure compatibility between instances)

Restoration from RPM Deployment on RHEL/Alma/Rocky

UIDS/GIDS

If you plan to **restore a backup from an RPM instance** to a new one (RPM or Docker), its likely that linux user and group **ids have changed** in between.

First you need to **identify the old ids** from your backup archive or your old instance, looking for each of those files (that should be owned by) :

- /etc/tuleap/conf/encryption_secret.key (codendiadm)
- /var/lib/gitolite/.gitolite/conf/gitolite.conf (gitolite)
- /var/lib/tuleap/ftp/pub (ftpadmin)

Example use of `ls -ldn` to display those ids

```
[root@tuleap ~]# ls -ldn /etc/tuleap/conf/encryption_secret.key /var/lib/gitolite/.gitolite/conf/gitolite.conf /var/lib/tuleap/ftp/pub
-r--------  1 980 980   64 Aug 16  2018 /etc/tuleap/conf/encryption_secret.key
-rw-rw----  1 976 976  867 Dec 21  2020 /var/lib/gitolite/.gitolite/conf/gitolite.conf
drwxr-xr-x 22 979 978 4096 Oct 22 08:59 /var/lib/tuleap/ftp/pub
```

Then **construct the maps** from the 3rd (uid) and 4th (gid) columns and the associated owner name

- usermap: `980:codendiadm,976:gitolite,979:ftpadmin`
- groupmap: `980:codendiadm,976:gitolite,978:ftpadmin`

As you can see, for ftpadmin uids and gids are not the same in this example.

Migrate/Restore to Docker

If you plan to **migrate RPM to docker**, you must use the following numerical ids as the users won't be created on the docker host:

- codendiadm: 900
- gitolite: 902
- ftpadmin: 904

So the maps will be `--usermap=980:900,976:902,979:904` and `--groupmap=980:900,976:902,978:904` for a docker target in this exemple.

If you are only **restoring a docker** instance **onto a docker** instance, those ids should already be set and a **remap is not necessary**.

Remap uids with rsync

Finally you can use a tool like `rsync` to help you remap your data while restoring/resyncing:

```
rsync [OPTION...] --usermap=980:codendiadm,976:gitolite,979:ftpadmin \
                  --groupmap=980:codendiadm,976:gitolite,978:ftpadmin \
                  SRC... [DEST]
```

Moving Tuleap folders to an external disk

For the mentioned tuleap data paths, you could move them on a **separate data disk** for easier backup.

Like for the "Restore Tuleap" process decribed before, you need a running tuleap instance for the users/uids/gids to be created.

Then, after suspending **all services including mysql** (if you want mysql to be on a data disk too) :

- move each directory to its new location, example: `mv /etc/nginx /data/etc_nginx`
- then at your convenience:
  - create symbolic links for each directory, example: `ln -s /data/etc_nginx /etc/nginx`
  - or use bind mounts through `/etc/fstab` like below (you need to `mount` them after updating the file)
- you can then restart services

Example of bind mounts via local fstab

```
/data/etc_nginx /etc/nginx none bind,nofail 0 0
/data/etc_redis /etc/redis none bind,nofail 0 0
/data/etc_tuleap /etc/tuleap none bind,nofail 0 0
/data/var_lib_gitolite /var/lib/gitolite none bind,nofail 0 0
/data/var_lib_mysql /var/lib/mysql none bind,nofail 0 0
/data/var_lib_tuleap /var/lib/tuleap none bind,nofail 0 0
```

Previous Next