atomic upgrade

By atomic upgrade, we mean:

- Put Tiki in read-only
- Make an exact copy
- Upgrade the copy
- Swap the two (rename or symlink directories)
  - No downtime
  - Easily rollback

Symlink approach

This might be an option Tiki-Manager can automate.

Supposing all sites are stored on `/data/www`, a new site having the domain `tiki.vbox` will be stored at `/data/www/tiki.vbox`. The folder structure for this site will be

```
/data/www/tiki.vbox
├── conf
│   ├── local.ini
│   └── local.php
├── html -> releases/20200227
├── logs
│   ├── httpd-access.log
│   └── httpd-error.log
├── releases
│   └── 20200227
│       └── db
│           └── local.php -> ../../../conf/local.php
└── data
    ├── img
    ├── modules
    └── temp
```

The `conf` folder

In this folder should exists at least the `local.php` file, that contains credentials for database access. Optionally, it might contains `local.ini` file, that contains overrides for tiki preferences.

The `logs` folder

Contains any log files. Usually, it holds webserver logs in it, but it might be a good idea put Tiki logs there.

The `releases` folder

Contains one or more Tiki installation. Each subfolder is a Tiki installation and the subfolder name is related
to Tiki version or to the installation date.

The **html** link

The html is the document root for the webserver. It should point to the latest Tiki installation on releases folder.

The **data** folder

Should have in folders Tiki needs to work properly.

- vendor
- temp
- img/wiki
- img/wiki_up

Secrets to make it works

1) You need to symlink local.php from `db/local.php` to the one in the conf folder

   ```bash
cd /data/www/tiki.vbox
ln -rs conf/local.php html/db/local.php
```

2) Tiki installations should have composer dependencies installed

3) Tiki installations should have permissions correctly set.

4) The data folder should be linked into Tiki structure

   ```bash
cd /data/www/tiki.vbox/

# data files
mkdir -p data/{img,modules}
mv html/img/trackers data/img
mv html/img/wiki data/img
mv html/img/wiki_up data/img
mv html/storage data/
ln -rTsf data/img/trackers html/img/trackers
ln -rTsf data/img/wiki html/img/wiki
ln -rTsf data/img/wiki_up html/img/wiki_up
ln -rTsf data/storage html/storage

# custom packages
mv html/vendor data/vendor
[ -f "html/composer.json" ] && mv html/composer.json data/composer.json
ln -rTsf data/vendor html/vendor
ln -rsf data/composer.json html/composer.json

# temp files
```
mv html/modules/cache data/modules
mv html/temp data/
ln -rTsf data/modules/cache html/modules/cache
ln -rTsf data/temp html/temp

Steps to update tiki

1) Copy the new tiki installation to releases folder
2) Ensure composer deps are installed
3) Ensure permissions are correctly set
5) Make html symlink point to the new tiki installation

ln -Tsf releases/new html

Steps to rollback

Point html symlink to previous selected Tiki installation.

ln -Tsf releases/old html

Concernings

- Tiki writes a lot of files in a lot of different folders. It is possible there is more folders to symlink
- The amount of necessary links make it complex to automate
- Keeping many installation on releases folder will take a lot of disk space
- On Windows might not be possible this approach
- Custom translations is not covered and will be lost in a upgrade

References

- https://roots.io/trellis/ playbooks

We want this for Tiki Manager and for Docker