Dependencies

Below are the basic dependencies needed to install and run Tiki Manager.

```
yum install app-php-engines git
```

- Package app-php-engines will provide PHP version 7.1 needed by Tiki Manager
- Package git will be used to checkout the Tiki Manager code

If you want to use a newer version of git, check Using Git 2.18 section.

If you want to use svn

If you have some legacy Tiki instances managed by SVN

```
yum install subversion
```

Additionally, you will need at least MariaDB and Apache if you want to run any Tiki in that ClearOS instance.

```
The MariaDB database (similar to MySQL)
yum install app-mariadb
```

Also, below we will use ClearOS functionalities to setup the Virtual Host for the Tiki Manager web interface, you will need to install/enable the "Web Server" app to easily create and manage websites.

```
The Apache Web Server
yum install app-web-server
```

Install Tiki Manager in ClearOS

The recommended layout, to keep things generic across Linux distributions, is to setup Tiki Manager under the folder `/opt/tiki-manager/app`, since ClearOS has its own system to manage virtual hosts, we will use a ClearOS managed virtualhost instead of setting up the web interface in `/opt/tiki-manager/webroot`. 
Install Tiki Manager Step By Step

```
mkdir -p /opt/tiki-manager/app
echo 71 > /opt/tiki-manager/.phpenv
git clone https://gitlab.com/tikiwiki/tiki-manager.git /opt/tiki-manager/app

cd /opt/tiki-manager/app
curl https://getcomposer.org/composer.phar > composer.phar
php composer.phar install --no-dev
```

Execute the first time to generate the keys:

```
php tiki-manager
```

Should generate an output similar to the one below, where the suggested approach is to leave the passphrase empty.

```
[root@server app]# php tiki-manager
If you enter a passphrase, you will need to enter it every time you run Tiki Manager, and thus, automatic, unattended operations (like backups, file integrity checks, etc.) will not be possible.
Enter passphrase (empty for no passphrase):

[...]
```

Update Tiki Manager

Tiki Manager is actively developed: https://gitlab.com/tikiwiki/tiki-manager/commits/master so you may want to do `git pull` every once in a while

```
An example of git pull to update Tiki Manager

[root@server tiki-manager]# git pull
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 30 (delta 7), reused 6 (delta 1)
Unpacking objects: 100% (30/30), done.
From https://gitlab.com/tikiwiki/tiki-manager
  6a77a47..f70ffa0  master     -> origin/master
Updating 6a77a47..f70ffa0
Fast-forward
  .env.dist                                        |  7 +++++++
  .gitlab-ci.yml                                   | 56
  +++++++++++++++++++++++++++++++++++++++++++++++----------
  README.md                                        |  8 ++++++++
  src/Application/Discovery.php                    | 17 +++++++++++---
  src/Libs/Helpers/Configuration.php               | 46 -------------------------------
  ----------
```
If the Composer dependencies have changed, you should do:

```bash
php composer.phar install --no-dev
```

## Tiki Manager Permissions

To make sure that you can use Tiki Manager

1. from the command line, as root,
2. from the web interface
3. and maybe have some backups copied offsite using Syncthing

you will need to adjust some of the permissions.

Enable Apache to always have write access to some of the folders.

```bash
cd /opt/tiki-manager/app
chown apache:apache -R backup cache data logs tmp
gtidac -R -m u:apache:rwX,d:u:apache:rwX backup cache data logs tmp
```

And to make sure the (default) temp directory can also be shared:

```bash
mkdir -p /tmp/trim_temp/
chown apache:apache /tmp/trim_temp/
gtidac -R -m u:apache:rwX,d:u:apache:rwX /tmp/trim_temp/
```

If you are using Syncthing, you will need also to enable the user running these processes to read from the backup folders, you can enable that by doing:

```bash
gtidac -R -m u:backupuser:rwX,d:u:backupuser:rwX /opt/tiki-manager/app/backup
```

Note: replace **backupuser** with the user running Syncthing
To easily configure Tiki-Manager application, copy .env.dist file to .env (inside /opt/tiki-manager/app/) and insert your configurations for the uncommented (#) entries.

Version Control System

Tiki Manager by default uses git and public repository. If you want to use SVN as your default vcs or another repository please add the following lines to your .env file.

```
DEFAULT_VCS=svn
GIT_TIKIWIKI_URI=<CUSTOM_GIT_REPOSITORY_URL>
SVN_TIKIWIKI_URI=<CUSTOM_SVN_REPOSITORY_URL>
```

Install Tiki Manager Web Interface Step By Step

Create a new virtual host using ClearOS admin interface, as an example we will call it tikimanager.example.org.

Use the folder layout "sandbox" (which is the default) and that you select "PHP 7.1" or above as the PHP Engine.

In that case, and for the remainder of this setup, we will consider:

Host: tikimanager.example.org
Path: /var/www/virtual/tikimanager.example.org/html

Execute:

```
php tiki-manager webmanager:enable
```

Follow the instructions on screen, when prompted enter the path above, and select the username and password to be used to authenticate you in the tiki manager web interface.

The interface / questions presented in the screen should look like the ones below:

```
[root@server app]# php tiki-manager webmanager:enable

Tiki Manager web administration files are located in the Tiki Manager directory. In order to make the interface available externally, the files will be copied to a web accessible location.

Permissions on the data folder will be changed to allow the web server to access the files.

For example, if your web root is /var/www/virtual/webtikimanager.example.org
* Files will be copied to /var/www/virtual/webtikimanager.example.org/html
* Tiki Manager web administration will be accessible from:
```
http://webtikimanager.example.org

You must have write access in /var/www/virtual

Simple authentication will be used. However, it is possible to restrict access to the administration panel to local users (safer).

This will enable the Tiki Manager administration web panel.
Continue with this action (y,n)? y

WWW Tiki Manager directory (ex: /var/www/virtual/webtikimanager.example.org/html): /var/www/virtual/tikimanager.example.org/html
Desired username: tiki
Desired password: password
Restrict use to localhost [no]: no
Do not run Composer as root/super user! See https://getcomposer.org/root for details
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Package operations: 3 installs, 0 updates, 0 removals
  - Installing components/jquery (3.3.1): Loading from cache
  - Installing fortawesome/font-awesome (v4.7.0): Loading from cache
  - Installing twitter/bootstrap (v4.1.3): Loading from cache
Generating autoload files
WWW Tiki Manager is now enabled.
Enjoy!

Once done, to launch the Tiki Manager for the first time you can use from your shell:

```bash
php tiki-manager
```

This will list the available command for Tiki Manager.

You can check your installation meet all the requirement by doing:

```bash
php tiki-manager manager:check
```

**bzip2 is missing**

If the manager:check result complains about bzip2 missing, just run:

```bash
yum install bzip2
```

**Timeouts during long running operations**

During long operations (like clone or clone and upgrade) you may receive an HTTP error code 503 with a message "Service Unavailable", you can increase Apache's proxy timeout to a more suitable value.
To do that at the Virtual Host level (so it's only enabled for tiki manager) you need to do the following:

- Copy the relevant sections from `/etc/httpd/conf.d/flex-80.conf` and `/etc/httpd/conf.d/flex-443.conf` (if using https) to a custom file:
Configure cron to run Tiki Manager automatically

There are some operations that you may want Tiki Manager to perform automatically for you in regular intervals like backups, updates, etc. for these you can setup cron to execute Tiki Manager, below are some examples.

For most of the examples you may need to check the instance id. You can list the instances and check the ids by executing:

```bash
php tiki-manager instance:list
```

Setup a nightly backup

In the example below, all instances will be backed up at midnight every day, except the instance with id 11.

```bash
# ensure /usr/clearos/bin/php is in the path (so multiple versions of PHP work as expected)
PATH=/usr/clearos/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

0 0 * * * cd /opt/tiki-manager/app && /usr/clearos/bin/php -d memory_limit=256M tiki-manager instance:backup --instances=all --exclude=11 --no-interaction
```

Setup a nightly update

In the example below, the instance with id 6 will be updated.

```bash
# ensure /usr/clearos/bin/php is in the path (so multiple versions of PHP work as expected)
PATH=/usr/clearos/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

0 1 * * * cd /opt/tiki-manager/app && /usr/clearos/bin/php -d memory_limit=256M tiki-manager instance:update --instances=6 --no-interaction
```

Setup a nightly clone and update

In the example below, instance 10 will be cloned to instance 11 and will be, after, upgraded to the branch
# ensure /usr/clearos/bin/php is in the path (so multiple versions of PHP work as expected)
PATH=/usr/clearos/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

0 2 * * * cd /opt/tiki-manager/app; /usr/clearos/bin/php tiki-manager instance:cloneandupgrade --source=10 --target=11 --branch=trunk --no-interaction

Using Git 2.18

By default ClearOS git package version is 1.8. If you want or need to use a newer version, you can use the following steps.

```bash
yum install rh-git218

# Make git available globally to all users
ln -s /opt/rh/rh-git218/enable /etc/profile.d/rh-git218.sh
ln -s /opt/rh/rh-git218/root/usr/bin/git /usr/clearos/bin/git

# Make git available in current session
source /opt/rh/rh-git218/enable
```

Enable Git 2.18 for Tiki Manager WebUI

```bash
echo 'LD_LIBRARY_PATH=/opt/rh/httpd24/root/usr/lib64/' >> /opt/tiki-manager/app/.env
```

Troubleshooting

Your Tiki may be locked with a never ending maintenance page displayed. You can manually turn off (or on) the maintenance status using CLI commands.

First check the ID of the instance you want to modify:

```bash
From within /opt/tiki-manager/app
```

<table>
<thead>
<tr>
<th>Check instances and turn maintenance status off</th>
</tr>
</thead>
<tbody>
<tr>
<td>php tiki-manager instance:detect</td>
</tr>
<tr>
<td>+---------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------</td>
</tr>
</tbody>
</table>

To turn maintenance if example2.com:
php tiki-manager instance:maintenance -i 3 -- off