Welcome to Canasta

Jeffrey Wang

EMWCon Spring 2022 – April 7, 2022 – Houston, TX, USA
What’s Canasta?

- “A comfy home for your data”
- All-in-one MediaWiki package
  - Docker-based stack powering MediaWiki
  - 100+ extensions already included
  - Data structures and packages ready for install
- Reducing technical and administrative complexity
  - Spend less time setting up, gain back more time for using your wiki
- Initial development by WikiWorks, WikiTeq, MyWikis

© 2022 The Canasta Project  canasta.wiki
Three components of Canasta

- Extension + skin collection
- Starter pages + data structures
- Canasta tech stack
Canasta stack’s values

- Fully open-source
- Convenient
- Same core MediaWiki
- Stable
- Customizable
- Extensible
- Source control compatible

© 2022 The Canasta Project  canasta.wiki
Advantages of the Canasta stack

- Easier, faster to set up
- Low maintenance requirements
  - Easy to upgrade MediaWiki
- Eliminate configuration drift
- Scalable and performant
- Written as a compatibility layer over pure MediaWiki
  - Future versions of MediaWiki should be easy for Canasta to support
- Easy to migrate back and forth between Canasta and non-Canasta
- Reduced development time and overhead
  - Canasta stakeholders all can contribute to each others’ success
Who’s using the Canasta stack?

Several companies in the MediaWiki industry have adopted the Canasta stack:

- WikiWorks
- WikiTeq
- MyWikis
- ...and more? :)

© 2022 The Canasta Project  canasta.wiki
Switching to Canasta

● Super easy!
● Drag and drop your existing LocalSettings.php
  ○ Except changing your database if you want to use the built-in Canasta DB
● Ditch the complicated setup of CirrusSearch, VisualEditor, Varnish, and HTTPS
Loading built-in extensions and skins

It’s as easy as 1-2-3:

1. Pick the extension you want to load from the list at canasta.wiki.
2. In LocalSettings.php, add the following (with config settings too):
   a. For extensions: cfLoadExtension( 'PageForms' );
   b. For skins: cfLoadSkin( 'Timeless' );
3. Restart the Canasta container.

That’s it!
Under the hood

- Runs on Docker, orchestrated by either Docker Compose or Kubernetes
- Easy setup instructions, set everything up in under 15 minutes
- Stack comprised of:
  - Canasta application container
    - MediaWiki, Apache, PHP, job queue runner, maintenance scripts runner, media transcoder, automated DB backups, 100+ bundled extensions
  - MySQL container
  - Elasticsearch container (for CirrusSearch)
  - Caddy container (for HTTPS/TLS termination)
  - Varnish (for frontend caching)
The Canasta container stack

(As of v1.0.0)

- Caddy
- Varnish
- Canasta
- MySQL
- Elasticsearch

Canasta tech stack:
- Extension + skin collection
- Starter pages + data structures
More than meets the eye

- Host OS shows all user-customizable MediaWiki files and directories
  - Same nomenclature as vanilla MediaWiki
- Canasta-specific code and shims + bundled extensions/skins located in Canasta image/container only
  - Not accessible to sysadmin—this prevents accidental changes contrary to how Canasta is meant to be used
  - Clear divide between persistent data and ephemeral code
LocalSettings “Inception”

- What the sysadmin sees:
  - LocalSettings.php
  - Settings snippets under settings directory

- What Canasta loads to LocalSettings.php
  - CanastaUtils.php (shims)
  - CanastaDefaultSettings.php (the absolute necessities for Canasta to work; the behind-the-scenes “plumbing”)
Derivative images

- Adapt Canasta to your needs
  - Start from the base Canasta image and make changes using a Dockerfile
- Making modifications is officially supported
- Things you could change
  - Bundled extensions/skins
  - Supporting services
  - CanastaDefaultSettings.php – change how the Canasta image does things
- Why make derivative images?
  - Maybe you manage many wikis but add your own special way of doing things.
  - Maybe you want to only use part of Canasta.
100+ extensions and skins

- Curated and vetted
- Criteria for adding extensions and skins
  - Works with the version of MediaWiki used by Canasta (currently 1.35)
  - Stable
  - Actively maintained and low risk of abandonment
  - Provide unique value that isn’t provided by any other bundled extension
Predefined data structures

- Everything is made for you
- Page Exchange + Canasta = a match made in heaven
  - Included with Canasta
  - One line to enable, one more line for each package

```php
cfLoadExtension('PageExchange');
```

- Import, update, delete
  - No risk in trying a new package
Updates since December 2021

- Finalized LocalSettings.php format
  - Added `settings/` directory for loading settings by snippets rather than by a monolithic file
    - You can think of it as “LocalSettings.php.d/”
    - Will help with CLI-based installation of extensions, skins
  - LocalSettings.php no longer needs to include CanastaUtils.php
  - Several settings are automatically set behind the scenes
    - Automatic Varnish configuration
- Added built-in support for TLS termination (Caddy) and frontend caching (Varnish)
- Refined our bundled extensions + skins
Welcome to Canasta 1.0!

As of today, Canasta is officially released to the general public!

Version 1.0 can be installed now.

Details at www.canasta.wiki
Demo!

Installing Canasta is so fast, we’ll do it right now from a brand new EC2 instance.
What’s next? (Canasta tech stack)

- memcached built-in
- Command line interface/CLI (expected summer 2022)
  - Manage Docker stuff directly from the command line
- ARM (AArch64) support
- Wiki farm support
- Improved support for Kubernetes
- Improved support for cloud container services
  - AWS ECS
  - Azure Container Instances
What’s next? (Extensions + skins collection)

- Add more extensions and skins
  - As long as they fit the criteria
- Criteria for adding extensions and skins
  - Works with the version of MediaWiki used by Canasta (currently 1.35)
  - Stable
  - Actively maintained and low risk of abandonment
  - Provide unique value that isn’t provided by any other bundled extension
What’s next? (Starter pages + data structures)

- Add more starter pages and data structures via Page Exchange
- Support for adding starter pages and data structures from the future Canasta CLI
More info

- GitHub: [github.com/CanastaWiki/Canasta](https://github.com/CanastaWiki/Canasta)
  - File bug reports and feature requests on our GitHub Issues page
- Website: [canasta.wiki](https://canasta.wiki)
  - Official resource for learning about Canasta and how to use it
Appendix: .env file example

```
[ubuntu@ip-172-31-25-239:~/Canasta-DockerCompose]$ cat .env
PORT=80
HTTPS_PORT=443
MYSQL_PASSWORD=mediawiki
MW_SITE_SERVER=https://canastatest1.mywikis.net
MW_SITE_FQDN=canastatest1.mywikis.net
PHP_UPLOAD_MAX_FILESIZE=10M
PHP_POST_MAX_SIZE=10M
PHP_MAX_INPUT_VARS=1000
MW_SITEMAP_SUBDIR=
MW_SITEMAP_IDENTIFIER=mediawiki
```
Appendix: Starting Canasta for the first time

```bash
[ubuntu@ip-172-31-25-239:~/canasta-dockercompose-2$ vim .env
[ubuntu@ip-172-31-25-239:~/Canasta-DockerCompose-2$ sudo docker-compose up -d
Creating network "canasta-dockercompose-2_default" with the default driver
Creating volume "canasta-dockercompose-2_mysql-data-volume" with default driver
Creating volume "canasta-dockercompose-2_elasticsearch" with default driver
Creating volume "canasta-dockercompose-2_caddy-data" with default driver
Creating volume "canasta-dockercompose-2_sitemap" with default driver
Creating canasta-dockercompose-2_varnish_1 ... done
Creating canasta-dockercompose-2_db_1 ... done
Creating canasta-dockercompose-2_caddy_1 ... done
Creating canasta-dockercompose-2_elasticsearch_1 ... done
Creating canasta-dockercompose-2_web_1 ... done
[ubuntu@ip-172-31-25-239:~/Canasta-DockerCompose-2$ ]
```

© 2022 The Canasta Project  canasta.wiki
Appendix: Configure DB
Appendix: Installation finished

MediaWiki 1.35.5 installation

Complete!

Congratulations! You have installed MediaWiki.

The installer has generated a `LocalSettings.php` file that contains all your configuration.

You will need to download it and put it in the base of your wiki installation (the same directory as index.php). The download should have started automatically.

If the download was not offered, or if you cancelled it, you can restart the download by clicking the link below:

Download LocalSettings.php

Note: If you do not do this now, this generated configuration file will not be available to you later if you exit the installation without downloading it.

When that has been done, you can enter your wiki.

Did you know that your wiki supports extensions?

You can browse extensions by category.

• Language
• Existing wiki
• Welcome to MediaWiki
• Connect to database
• Upgrade existing installation
• Database settings
• Name
• Options
• Install
• Complete!
• Restart installation
Appendix: Add most important exts + skins

```c
# End of automatically generated settings.
# Add more configuration options below.

cfLoadSkin( 'Vector' );

cfLoadExtension( 'VisualEditor' );

cfLoadExtension( 'Elastica' );
cfLoadExtension( 'CirrusSearch' );
cfLoadExtension( 'AdvancedSearch' );

$wgSearchType = 'CirrusSearch';
"config/LocalSettings.php" 145L, 4997C written
```
Appendix: Initialize CirrusSearch
Appendix: Initialization commands

$ sudo docker-compose exec web php 
/var/www/mediawiki/w/canasta-extensions/CirrusSearch/maintenance/UpdateSearchIndexConfig.php && sudo 
docker-compose exec web php 
/var/www/mediawiki/w/canasta-extensions/CirrusSearch/maintenance/ForceSearchIndex.php --skipLinks 
--indexOnSkip && sudo docker-compose exec web php 
/var/www/mediawiki/w/canasta-extensions/CirrusSearch/maintenance/ForceSearchIndex.php --skipParse