



Canasta

Welcome to Canasta

Jeffrey Wang

EMWCon Spring 2022 – April 7, 2022 – Houston, TX, USA

What's Canasta?



Canasta

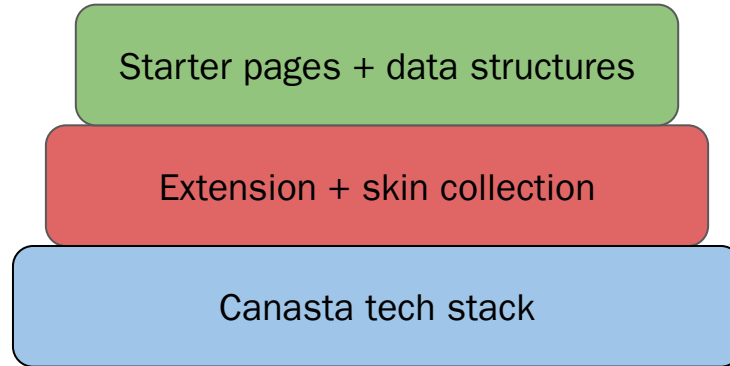
- “A comfy home for your data”
- All-in-one MediaWiki package
 - Docker-based stack powering MediaWiki
 - 100+ extensions already included
 - Data structures and packages ready for install
- Reducing technical and administrative complexity
 - Spend less time setting up, gain back more time for using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



Three components of Canasta



Canasta



Canasta stack's values



Canasta

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

Canasta tech stack

Extension + skin collection

Starter pages + data structures

Advantages of the Canasta stack



Canasta

- 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

Canasta tech stack

Extension + skin collection

Starter pages + data structures

Who's using the Canasta stack?



Canasta

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

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



Canasta tech stack

Extension + skin collection

Starter pages + data structures

Switching to Canasta



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

Canasta tech stack

Extension + skin collection

Starter pages + data structures

Loading built-in extensions and skins



Canasta

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!

Canasta tech stack

Extension + skin collection

Starter pages + data structures

Under the hood



Canasta

- 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)

Canasta tech stack

Extension + skin collection

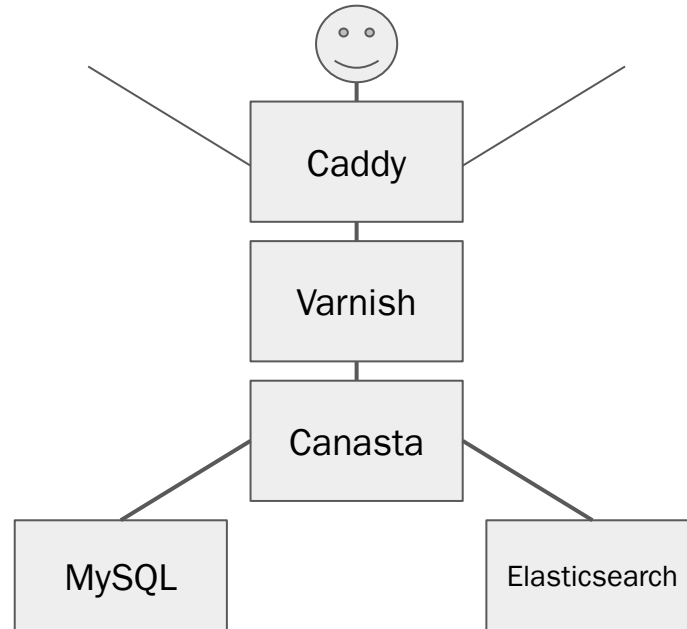
Starter pages + data structures

The Canasta container stack



Canasta

(As of v1.0.0)



Canasta tech stack

Extension + skin collection

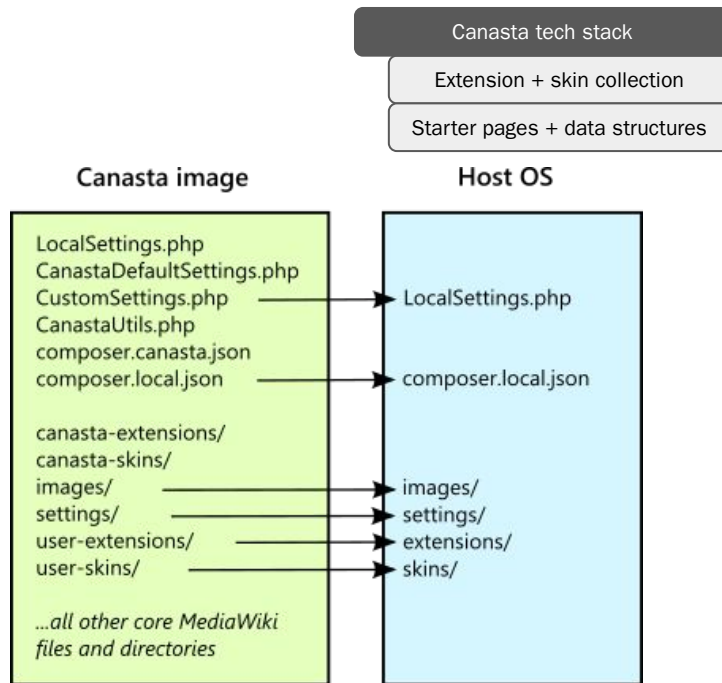
Starter pages + data structures

More than meets the eye



Canasta

- 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

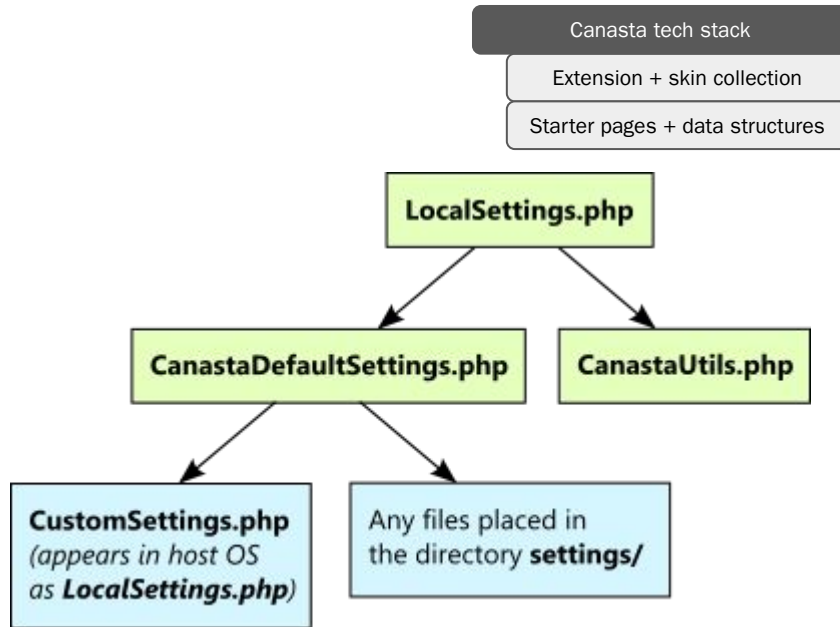


LocalSettings “Inception”



Canasta

- 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



Canasta

- 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.

Canasta tech stack

Extension + skin collection

Starter pages + data structures

100+ extensions and skins



Canasta

- 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

Skins included in Canasta

- Chameleon
- Cologne Blue
- Mirrored News
- Modern
- MonoBook
- Pivot
- Refreshed
- Timeless
- Vector

Extensions included in Canasta

Administrative tools

[Admin Links](#) · [Interwiki](#)

Analytics

[ContributionScores](#) · [Editcount](#) · [GoogleAnalyticsMetrics](#) · [GTag](#) · [HotnetLinks](#) · [WhoisWatching](#)

Automated page creation & modification

[Data Transfer](#) · [Page Exchange](#) · [Replace Text](#)

Communication

[CommentStreams](#) · [ContactPage](#) · [DiscussionTools](#) · [DismissableSiteNotice](#) · [Echo](#) · [GlobalNotice](#) · [MassMessage](#) · [MassMessageEmail](#) · [NewUserMessage](#) · [Thanks](#) · [WikiForum](#)

Data storage and querying

[Cargo](#) · [Disambiguator](#) · [PageImages](#) · [Semantic Brackets](#) · [Links](#) · [Semantic Compound Queries](#) · [Semantic Drilldown](#) · [Semantic Extra Special Properties](#) · [Semantic MediaWiki](#) · [Semantic Result Formats](#) · [Semantic Scribunto](#) · [SimpleChanges](#) · [SubPageList](#)

Editing

[CharInsert](#) · [CodeEditor](#) · [CodeMirror](#) · [InputBox](#) · [LinkSuggest](#) · [Linter](#) · [Page Forms](#) · [SaveSpinner](#) · [TemplateData](#) · [TinyMCE](#) · [VEForm](#) · [VisualEditor](#) · [WikiEditor](#)

File upload & handling

[CommonsMetadata](#) · [GoogleDocCreator](#) · [ImageMap](#) · [MultiUpload](#) · [MultimediaViewer](#) · [PdfHandler](#) · [SimilesBatchUpload](#) · [TimedMediaHandler](#) · [UploadWizard](#)

HTML & multimedia support

[EmbedVideo](#) · [Flex Diagrams](#) · [Graph](#) · [HTML Tags](#) · [Image](#) · [Math](#) · [SimpleMathJax](#) · [Wikipedia](#)

Interface & display customization

[BetaFeatures](#) · [Gadgets](#) · [Header Footer](#) · [HeadScript](#) · [SkinPerPage](#)

Meta tags

[Description2](#) · [OpenGraphMeta](#) · [WikiSEO](#)

PDF export

[Collection](#)

Read & write access

[Approved Revs](#) · [LockAuthor](#) · [Lockdown](#)

Search

[AdvancedSearch](#) · [CirrusSearch](#) · [Elastic](#)

Social tools

[AJAXPoll](#) · [SocialProfile](#) · [VoteNY](#)

Spam control

[AbuseFilter](#) · [ConfirmEdit](#) · [Nuke](#) · [SmiteSeam](#) · [SpamBlacklist](#) · [TitleBlacklist](#)

Canasta tech stack

Extension + skin collection

Starter pages + data structures

Predefined data structures



Canasta

- 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

```
cfLoadExtension( 'PageExchange' );
$wgPageExchangePackageFiles[] =
'https://raw.githubusercontent.com/CanastaWiki/CanastaPXConsultingCRM/main/page-exchange.json';
$wgPageExchangePackageFiles[] = 'https://raw.githubusercontent.com/CanastaWiki/CanastaPXWidgets/main/page-exchange.json';
```

- Import, update, delete
 - No risk in trying a new package

Canasta tech stack

Extension + skin collection

Starter pages + data structures

Updates since December 2021



Canasta

- 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!



Canasta

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

Version 1.0 can be installed now.

Details at www.canasta.wiki

Demo!



Canasta

Installing Canasta is so fast, **we'll do it right now** from a brand new EC2 instance.

What's next? (Canasta tech stack)



Canasta

- 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

Canasta tech stack

Extension + skin collection

Starter pages + data structures

What's next? (Extensions + skins collection)



Canasta

- 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

Canasta tech stack

Extension + skin collection

Starter pages + data structures

What's next? (Starter pages + data structures)



Canasta

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

Canasta tech stack

Extension + skin collection

Starter pages + data structures

More info



Canasta

- GitHub: github.com/CanastaWiki/Canasta
 - File bug reports and feature requests on our GitHub Issues page
- Website: canasta.wiki
 - Official resource for learning about Canasta and how to use it

Appendix: .env file example



Canasta

```
[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



Canasta

```
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$
```


Appendix: Configure DB



Canasta

MediaWiki 1.35.5 installation



[MediaWiki home](#)
[User's Guide](#)
[Administrator's Guide](#)
[FAQ](#)
[Seek help](#)
[Bug tracker](#)
[Contribute](#)
[Release notes](#)
[Copying](#)
[Upgrading](#)

Connect to database



MediaWiki supports the following database systems:

- MariaDB is the primary target for MediaWiki and is best supported. MediaWiki also works with MySQL and Percona Server, which are MariaDB compatible. ([How to compile PHP with MySQL support](#))
- PostgreSQL is a popular open source database system as an alternative to MySQL. ([How to compile PHP with PostgreSQL support](#))
- SQLite is a lightweight database system that is very well supported. ([How to compile PHP with SQLite support, uses PDO](#))

If you do not see the database system you are trying to use listed below, then follow the instructions linked above to enable support.

Database type:

MariaDB, MySQL, or compatible

MariaDB/MySQL settings

Database host:

[help](#)

db

Identify this wiki

Database name (no hyphens):

[help](#)

my_wiki

Database table prefix (no hyphens):

[help](#)

User account for installation

Database username:

[help](#)

root

Database password:

[help](#)

mediawiki

- [Language](#)
- [Existing wiki](#)
- [Welcome to MediaWiki!](#)
- **Connect to database**
- [Upgrade existing installation](#)
- [Database settings](#)
- [Name](#)
- [Options](#)
- [Install](#)
- [Completer](#)
- [Restart installation](#)

[← Back](#) [Continue →](#)

Appendix: Installation finished



Canasta



MediaWiki home
User's Guide
Administrator's Guide
FAQ
Seek help
Bug tracker
Contribute
Release notes
Copying
Upgrading

MediaWiki 1.35.5 installation

Complete!



Congratulations! You have installed MediaWiki.

The installer has generated a `LocalSettings.php` file. [Main Page](#) It 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



Canasta

```
# 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



Canasta

```
ubuntu@ip-172-31-25-239:~/Canasta-DockerCompose-2$ sudo docker-compose exec web php /var/www/mediawiki/w/canasta-extensions/CirrusSearch/maintenance/UpdateSearchIndexConfig.php
indexing namespaces...
mw_cirrus_metastore missing, creating new metastore index.
Creating metastore index... mw_cirrus_metastore_first Scanning available plugins...none
ok
    Green!Creating mw_cirrus_metastore alias to mw_cirrus_metastore_first.
    Indexing namespaces...done
content index...
    Fetching Elasticsearch version...6.8.20...ok
    Scanning available plugins...none
    Picking analyzer...english
    Inferring index identifier...my_wiki_content_first
    Creating index...ok
        Validating number of shards...ok
        Validating replica range...ok
        Validating shard allocation settings...done
        Validating max shards per node...ok
    Validating analyzers...ok
    Validating mappings...
        Validating mapping...different...corrected
    Validating aliases...
        Validating my_wiki_content alias...alias is free...corrected
        Validating my_wiki alias...alias not already assigned to this index...corrected
    Updating tracking indexes...done
general index...
    Fetching Elasticsearch version...6.8.20...ok
    Scanning available plugins...none
    Picking analyzer...english
    Inferring index identifier...my_wiki_general_first
    Creating index...ok
        Validating number of shards...ok
        Validating replica range...ok
        Validating shard allocation settings...done
        Validating max shards per node...ok
    Validating analyzers...ok
    Validating mappings...
        Validating mapping...different...corrected
    Validating aliases...
        Validating my_wiki_general alias...alias is free...corrected
        Validating my_wiki alias...alias not already assigned to this index...corrected
    Updating tracking indexes...done
archive index...
    Fetching Elasticsearch version...6.8.20...ok
    Scanning available plugins...none
    Picking analyzer...english
    Inferring index identifier...my_wiki_archive_first
    Creating index...ok
        Validating number of shards...ok
        Validating replica range...ok
        Validating shard allocation settings...done
        Validating max shards per node...ok
    Validating analyzers...ok
    Validating mappings...
        Validating mapping...different...corrected
    Validating aliases...
        Validating my_wiki_archive alias...alias is free...corrected
    Updating tracking indexes...done
ubuntu@ip-172-31-25-239:~/Canasta-DockerCompose-2$
```

Appendix: Initialization commands



Canasta

```
$ 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
```