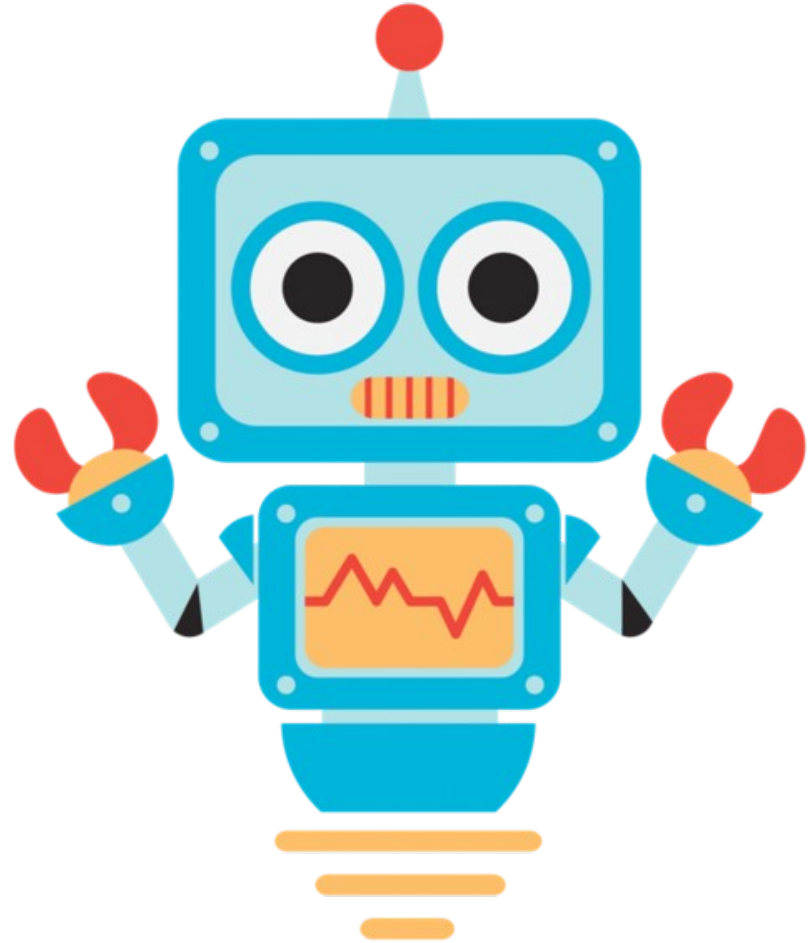


MediaWiki, SQL/XML dumps and Docker: Making tests easier

ariel@wikimedia.org, SMWCon Fall 2021

I forced a bot to watch over 1000 hours of my Q&A's about SQL/XML dump testing and then asked it to write the audience questions on its own.

Who the heck
are you and
why are you
talking to us?



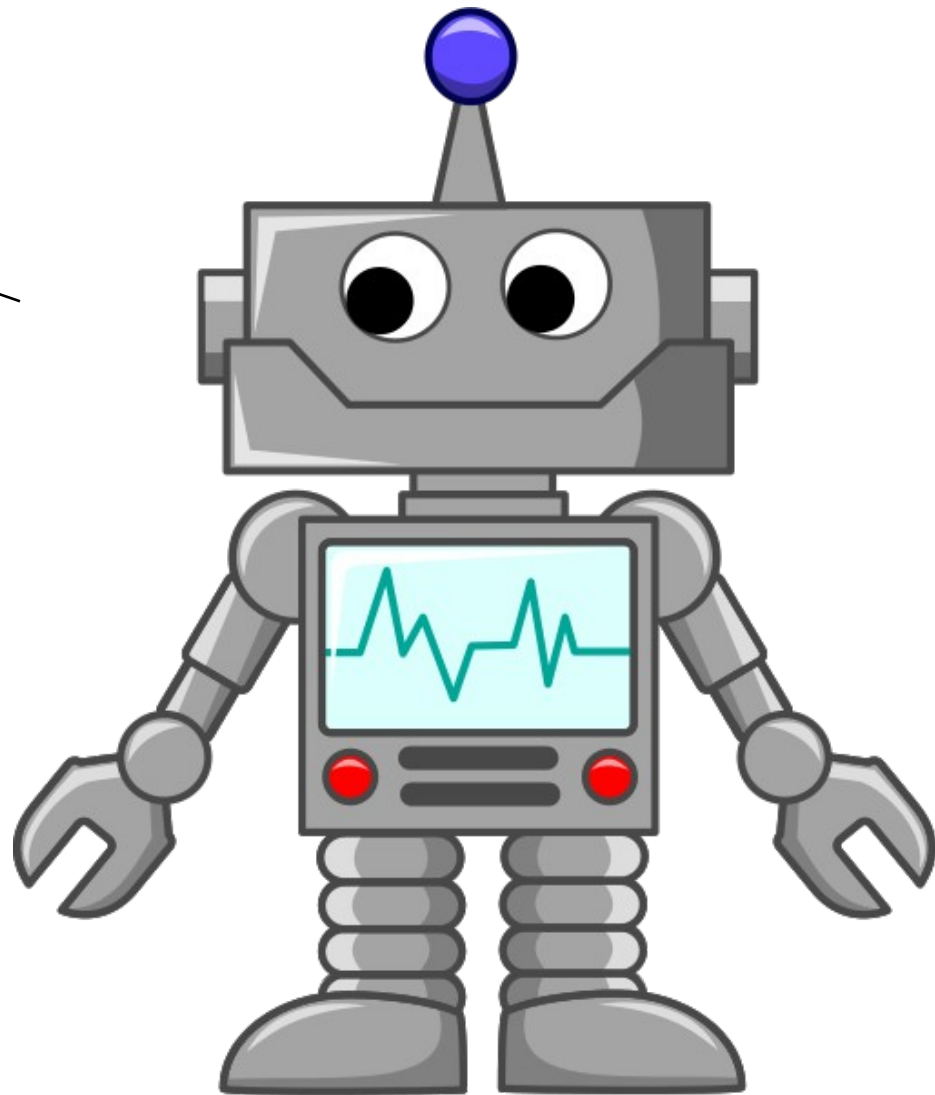
Name: Ariel Glenn

Title: Senior Software Engineer

Job: Manage production of dumps of Wiki content and other public datasets... for > 10 years



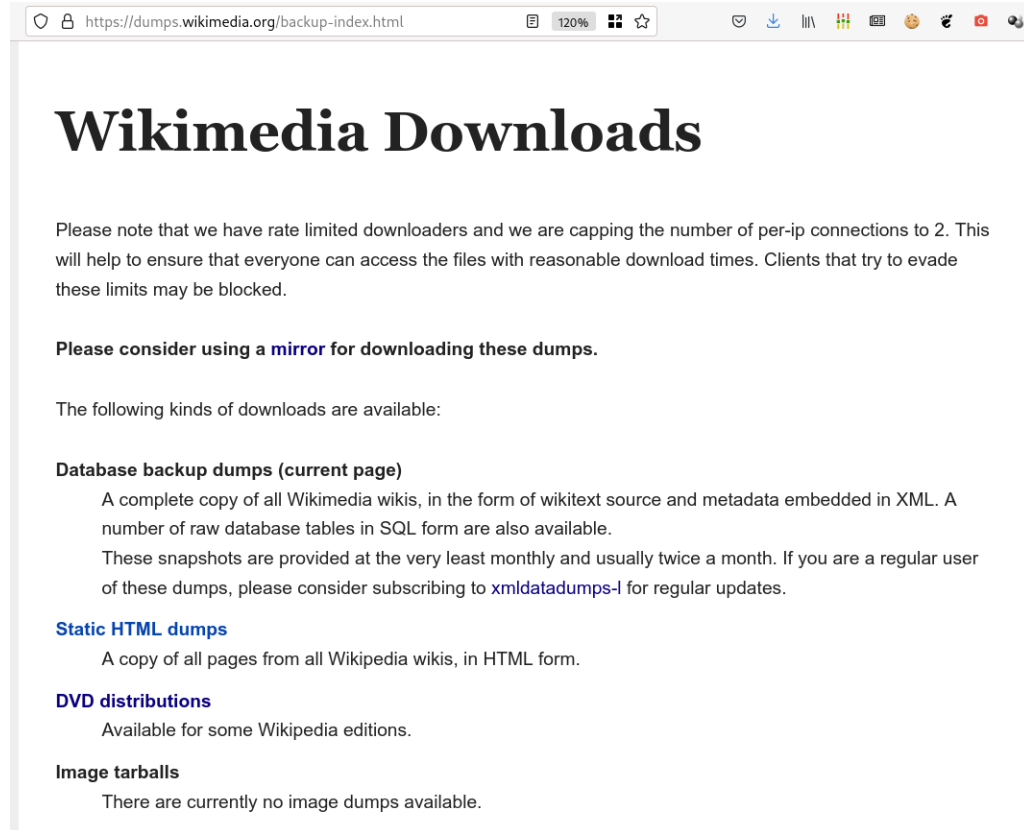
What are these dumps of which you speak, and why do we care?



Full content of all
public wiki projects in
all languages, twice a
month

Remix, Reuse,
Recycle!

Many other datasets
too

A screenshot of a web browser displaying the Wikimedia Downloads page. The browser's address bar shows the URL 'https://dumps.wikimedia.org/backup-index.html'. The page title is 'Wikimedia Downloads'. The main content includes a notice about rate-limited downloaders, a recommendation to use mirrors, and a list of available download types: Database backup dumps (current page), Static HTML dumps, DVD distributions, and Image tarballs. Each type has a brief description of what it contains and how often it is updated.

https://dumps.wikimedia.org/backup-index.html

Wikimedia Downloads

Please note that we have rate limited downloaders and we are capping the number of per-ip connections to 2. This will help to ensure that everyone can access the files with reasonable download times. Clients that try to evade these limits may be blocked.

Please consider using a [mirror](#) for downloading these dumps.

The following kinds of downloads are available:

- Database backup dumps (current page)**

A complete copy of all Wikimedia wikis, in the form of wikitext source and metadata embedded in XML. A number of raw database tables in SQL form are also available.

These snapshots are provided at the very least monthly and usually twice a month. If you are a regular user of these dumps, please consider subscribing to [xmldatadumps-l](#) for regular updates.
- Static HTML dumps**

A copy of all pages from all Wikipedia wikis, in HTML form.
- DVD distributions**

Available for some Wikipedia editions.
- Image tarballs**

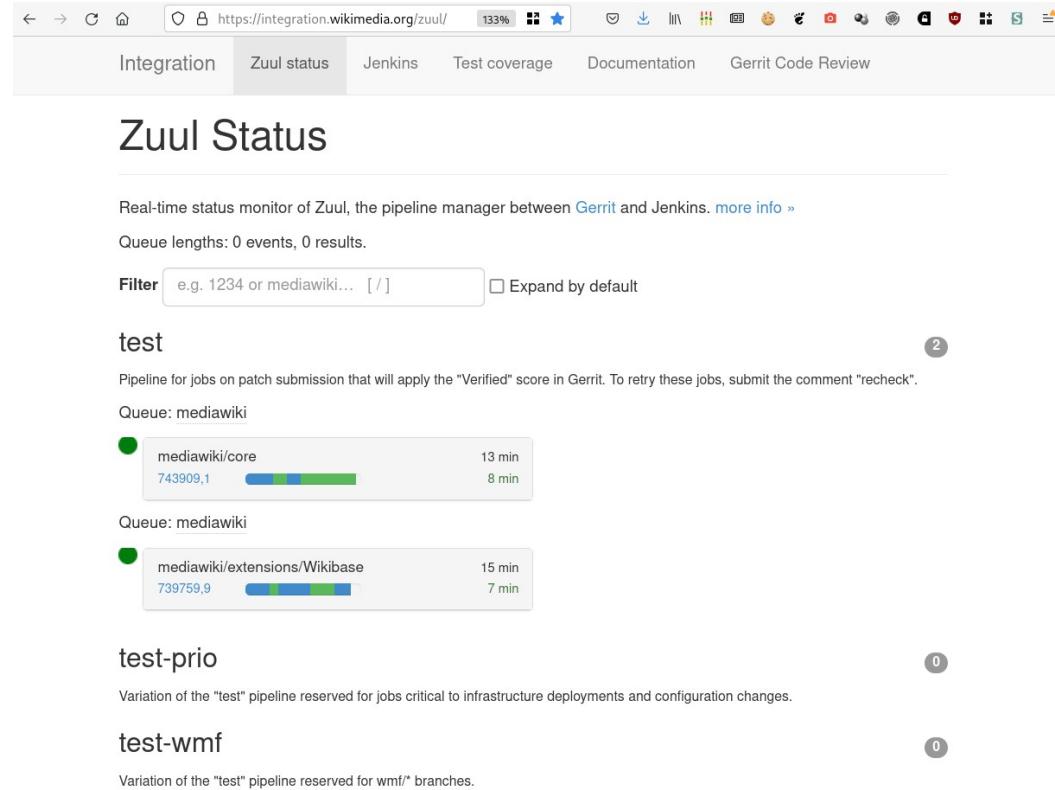
There are currently no image dumps available.

OK, I'm sold.
These
dumps are
important. So
you do CI
tests, right?

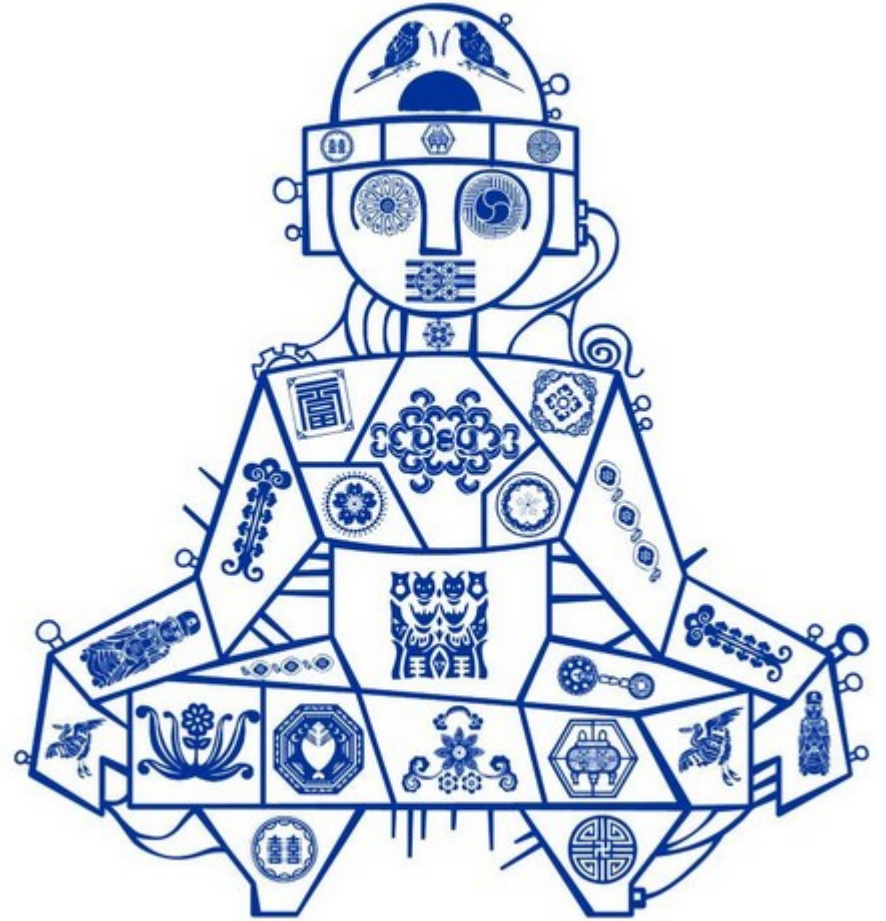


Proper tests would mean multiple batches across multiple servers

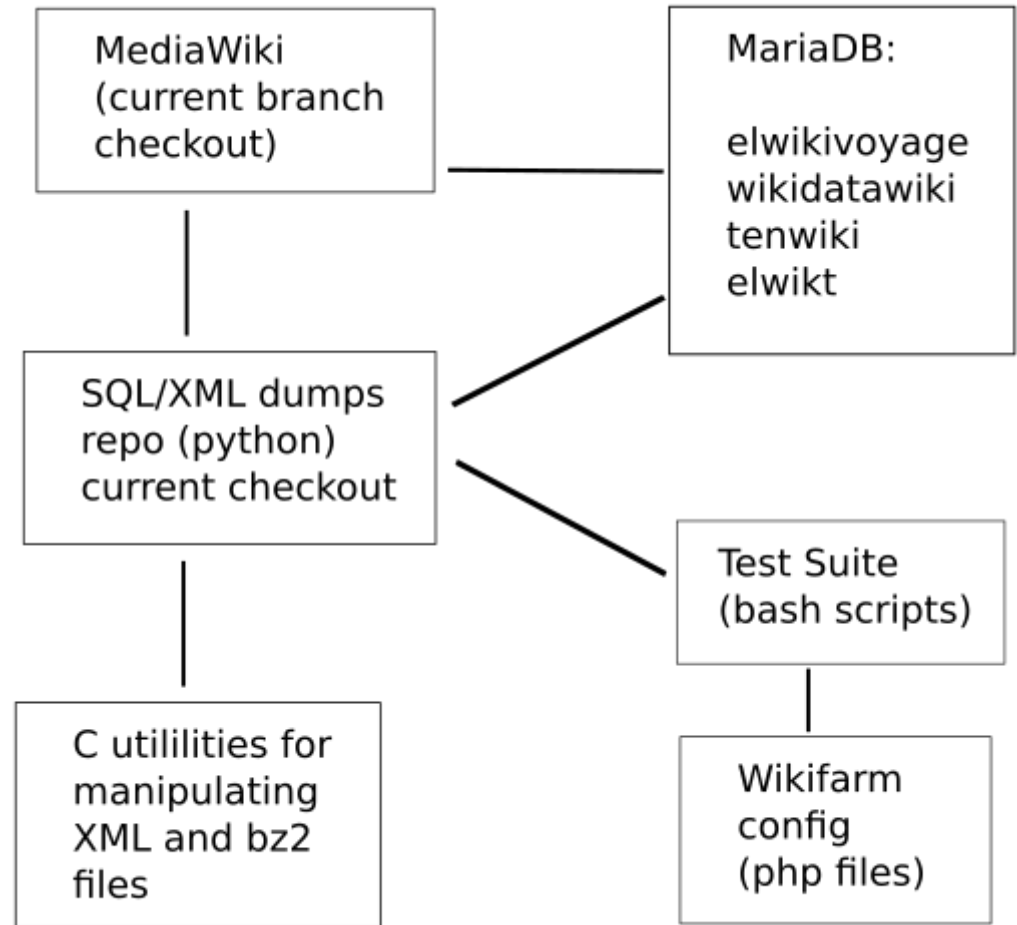
Changes to MediaWiki core currently take 30-40 minutes to pass CI already



So CI can't be used for in-depth testing. How about a local install?



I test with a local installation. But no one else has that same local install. A second person will begin dumps work soon. They are on a different OS... uh ohes!

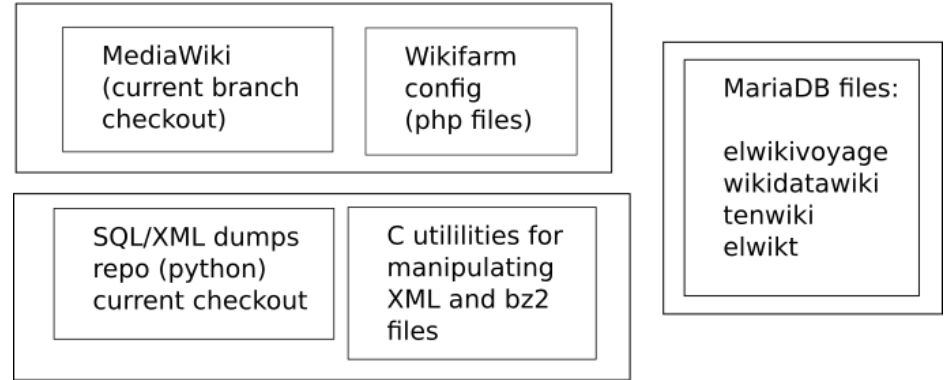


I see, no
local install.
But then
what?

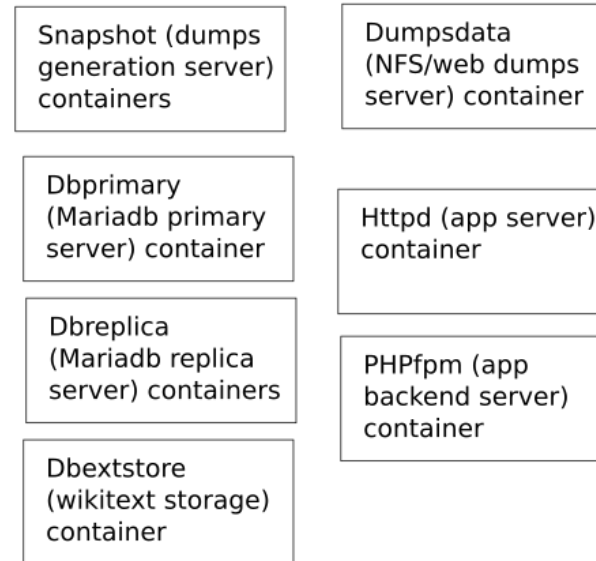


What can I share with other devs? What has all the binaries baked in on the right version of the right OS? What can have multiple db servers and multiple dumps generation servers with just a click? Hmm...

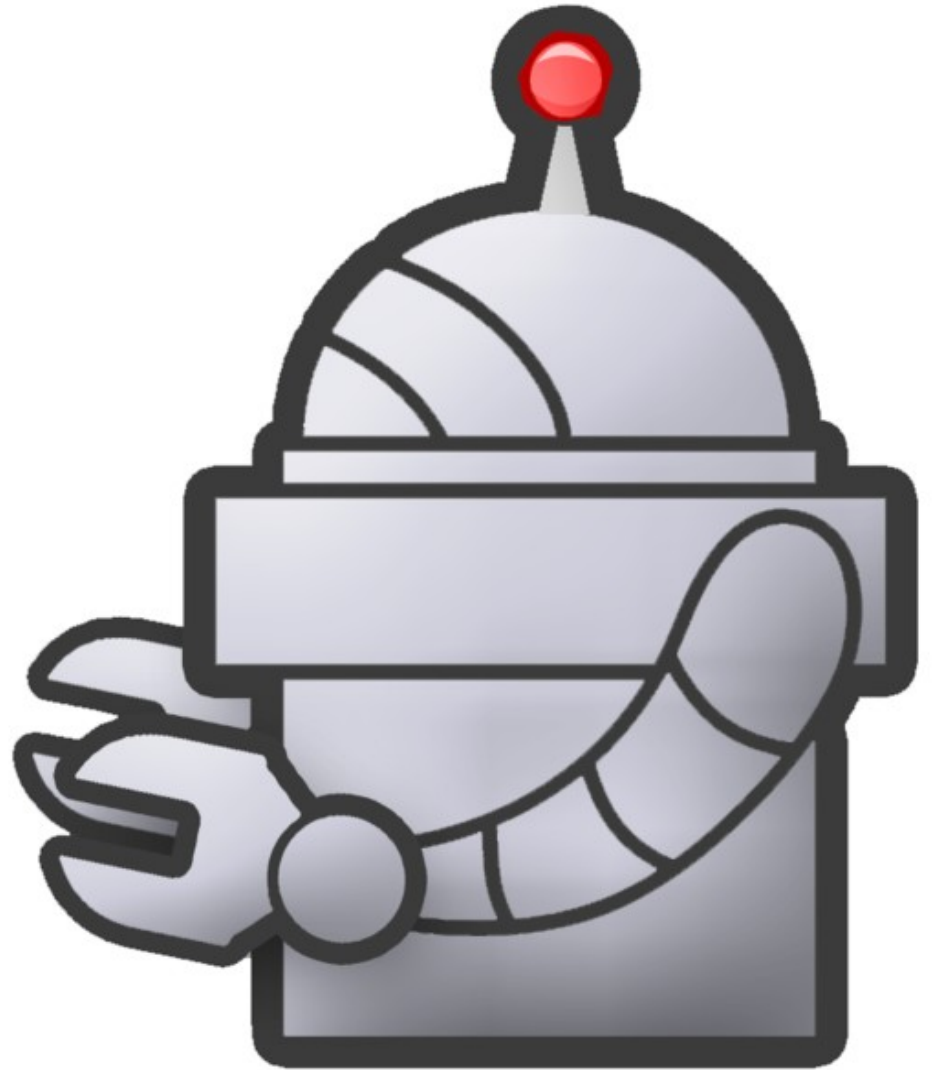
Mounted volumes



Containers



Docker, of course. But how hard will that be to install?



Easy networking →
dnsmasq, some config
file tweaks, a docker
network watcher...

Setup of db files on
mounted volume →
mariadb/mysql
executables on local
machine

Honestly, a bit gross

```
# Architecture, design, and configuration
```

```
## Networking
```

```
To make networking from the desktop/laptop suck less, we use a modified version of  
https://github.com/nicolai-budico/dockerhosts which is currently living at  
https://github.com/apergos/dockerhosts  
It relies on dnsmasq and a one-line modification to /etc/systemd/resolved.conf
```

```
This may not be the best choice, but the more popular solution, DPS, works by  
editing /etc/hosts periodically, and other solutions that I investigated had equally  
problematic approaches. This one seemed the least objectionable. Note however this  
method requires use of a Linux distro that uses systemd-resolved. Most modern distros do.
```

```
With that setup in place, we can get to any container in any container set by  
specifying the fqdn: container-name.set-name.lan  
All names end in ".lan" so that we are sorta-kinda in compliance with "don't  
use fake TLDs except these known ones that external resolvers know to ignore,  
in theory".
```

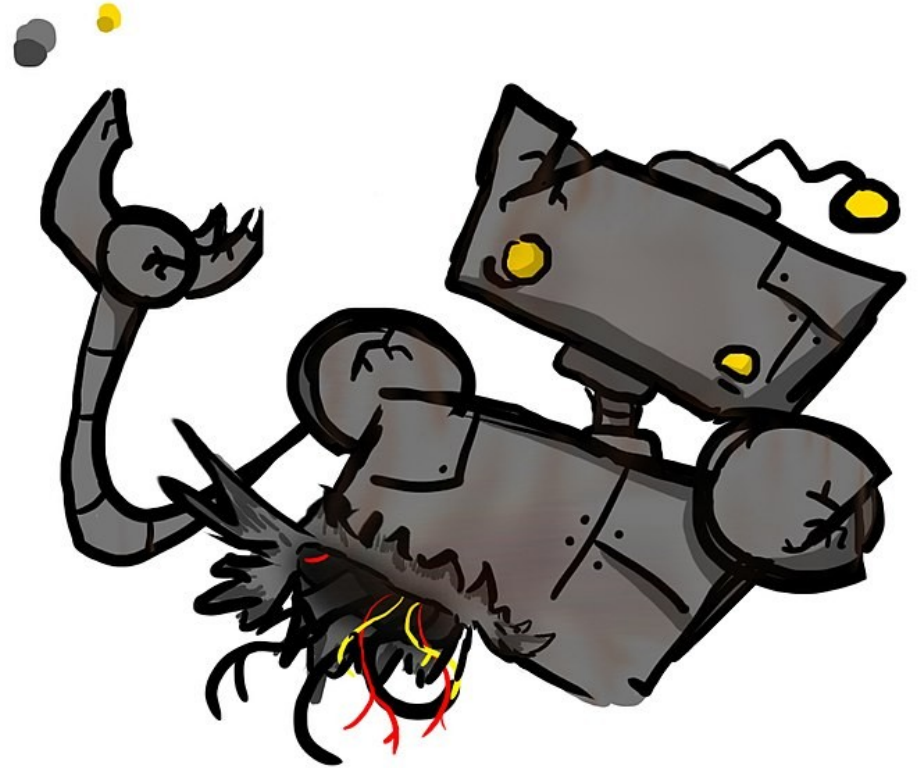
```
## Container customization
```

```
Some containers need to include the names of other containers in their  
configuration settings. We could pass this information on the fly  
to each container at container start, and have an ENTRYPOINT script  
make substitutions in a set of files varying on each container.
```

```
Or we can bake these changes into an image derived from the base  
image for each container type (httpd, dbprimary, and so on),  
so that on startup of the container, it just starts its specific  
services.
```

```
I still don't know which approach will be better in the long run.
```

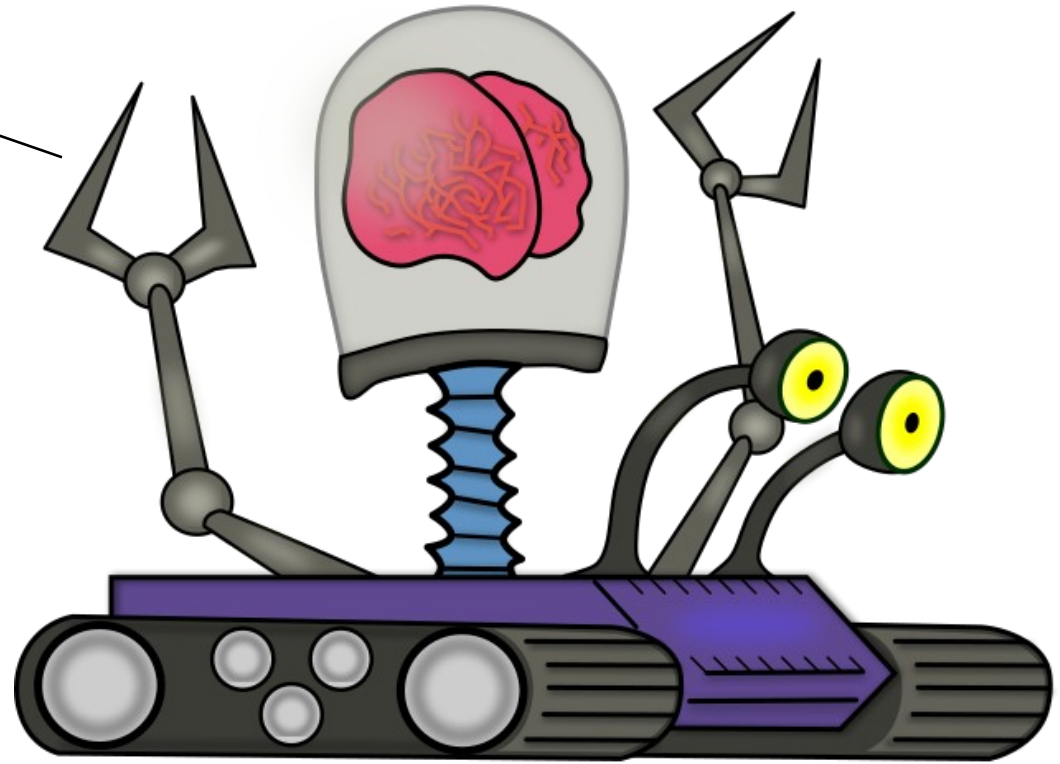
Complicated!
Why not
docker-
compose and
official Mariadb
and Apache
images?



We need everything to be reproducible and look as much like production as possible. We want to run on specific branches instead of HEAD, using committed vendor libraries instead of what composer finds.



That makes sense, but it's still complicated. Any plans to simplify?



Plan:

Move mariadb setup to run from a container

Reasonable default config

Automated checkouts of dumps, mw repos

Predefined sample test suites and sample content

```
# sample settings for setting up a wikifarm of containers for
# testing and a vanilla test

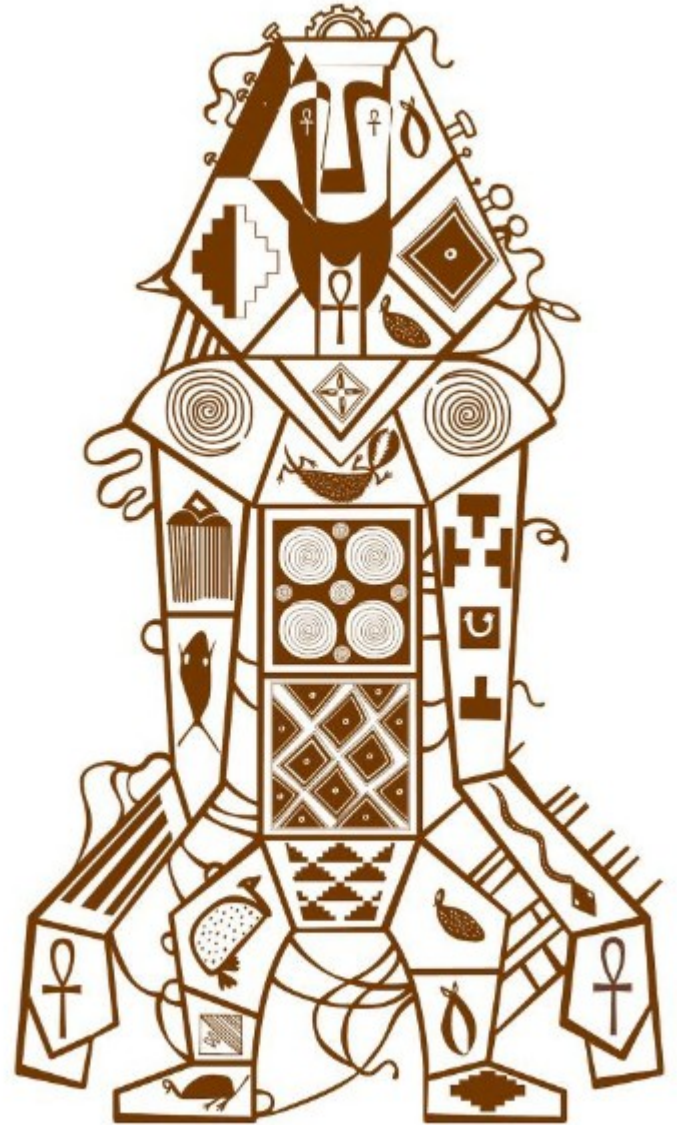
# these values are fallback for all container sets
global:
  passwords:
    dbs:
      wikidb_user: wikidbuser_defaultset
      wikidb_admin: wikidbadmin_defaultset
      root: notverysecure
  containers:
    root: testing

sets:
  defaultset:
    snapshots: 1
    dbdatadir: /secondary/wikifarms/defaultset/dbdata
    wikifarmdir: /secondary/wikifarms/defaultset/wikifarm
    dbprimary: true
    dbreplicas: 0
    dbextstore: false
    httpd: true
    phpfpd: true
    dumpsdata: false

    # these entries must correspond to <wikidb>.<something>.sql.gz files
    # that are in the docker_helpers/mariadb/imports/<setname> subdirectory.
    # dbs will be created and the data imported for each entry
    # in the list.
    # if you want the same data to be used in multiple wikifarms,
    # just symlink the file from one set subdir to another.
    wikidbs:
      - elwikivoyage

    # all wiki dbs have the same wikiuser and same wikiadmin
    # accounts, with a single shared password for each account
    # defined in your config. If you don't define one, the default
```

That sounds better. How about running it?



Once the containers are built,
one script with one argument
starts them all

Ssh in to any container is easy

Run a dumps test suite via bash
script (currently, maybe python in
future)

Examining the dbs can be done
from any container via mysql

Use your preferred editor to
make dumps repo or MediaWiki
code changes

```
# sample settings for setting up a wikifarm of containers for
# testing and a vanilla test

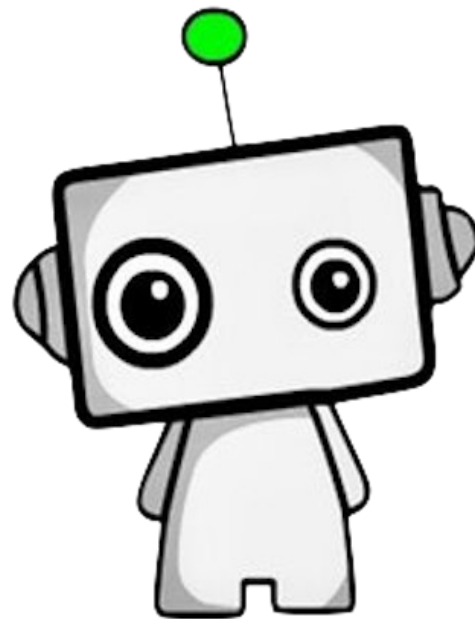
# these values are fallback for all container sets
global:
  passwords:
    dbs:
      wikidb_user: wikidbuser_defaultset
      wikidb_admin: wikidbadmin_defaultset
      root: notverysecure
  containers:
    root: testing

sets:
  defaultset:
    snapshots: 1
    dbdatadir: /secondary/wikifarms/defaultset/dbdata
    wikifarmdir: /secondary/wikifarms/defaultset/wikifarm
    dbprimary: true
    dbreplicas: 0
    dbextstore: false
    httpd: true
    phpfpd: true
    dumpsdata: false

    # these entries must correspond to <wikidb>.<something>.sql.gz files
    # that are in the docker_helpers/mariadb/imports/<setname> subdirectory.
    # dbs will be created and the data imported for each entry
    # in the list.
    # if you want the same data to be used in multiple wikifarms,
    # just symlink the file from one set subdir to another.
    wikidbs:
      - elwikivoyage

    # all wiki dbs have the same wikiuser and same wikiadmin
    # accounts, with a single shared password for each account
    # defined in your config. If you don't define one, the default
```

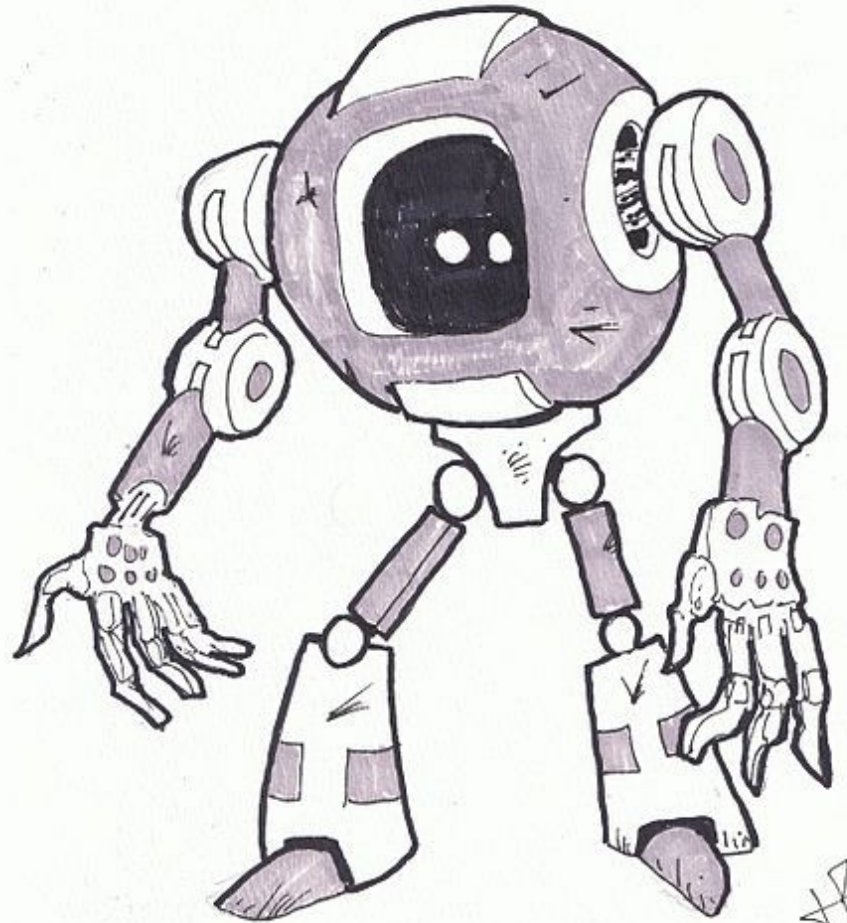
I'm going to
regret asking
this, but how
does it all
work?



All the bits and pieces

- Python Docker sdk
- One Docker network per test cluster, multiple clusters possible
- Base image from buster, sets up our apt repos, copies in setup scripts
- Base image per container flavour with all packages, using those setup scripts
- Final image per flavour with all configuration, container names substituted in where needed (human-readable)
- DB credentials defined in configuration, user names are fixed; number of snapshot instances, db replicas, external stores, and use of NFS dumps server set in configuration
- Imports of predefined wiki content (sql.gz files) to dbs

You mentioned
“external
store” and db
replicas. How
does that
work?



2016

“Soon”.

To be added:

db replicas

db external stores

multiple snapshot instances

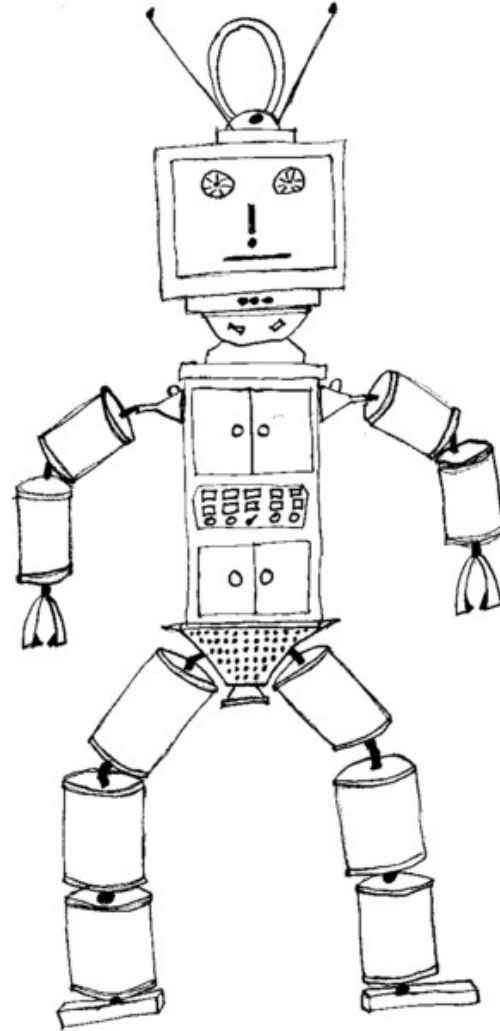
dumps NFS server

memcached, mcrouter,
https?

many bug fixes...



Ah, so this is
vaporware??



A real repo exists.

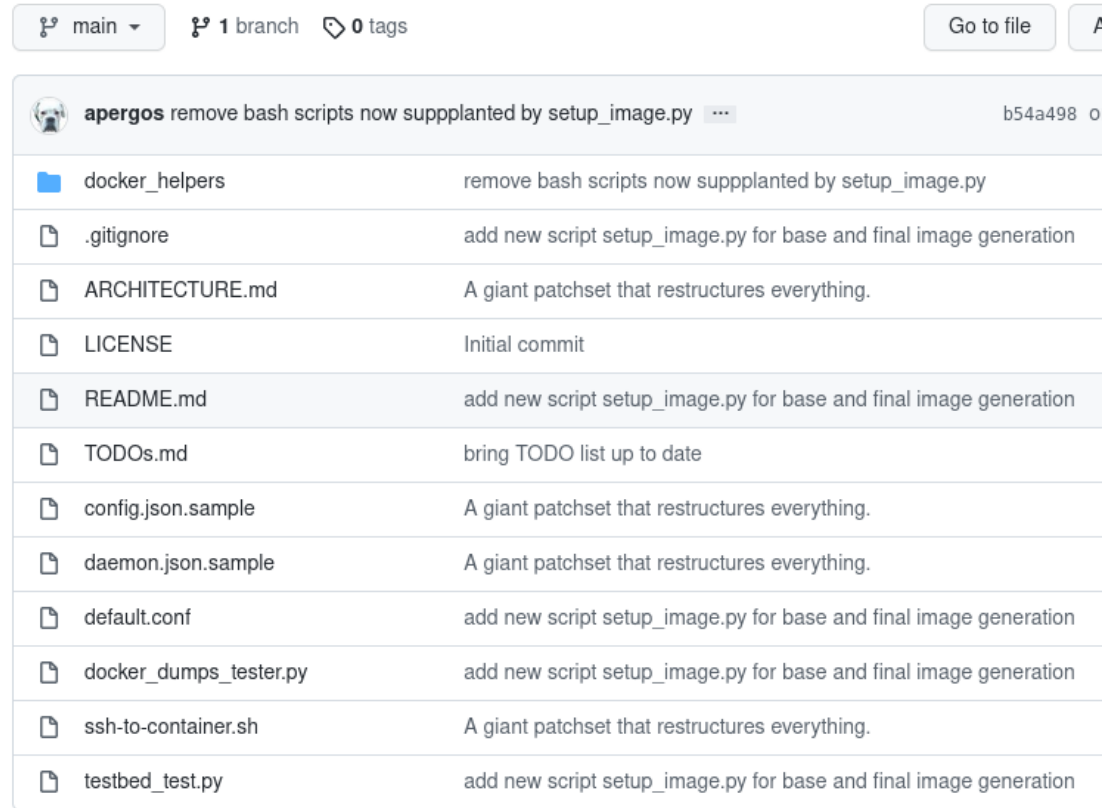
Code in that repo builds images and containers (or spins them down/destroyed) for dbprimary, httpd, php-fpm and snapshot instances.

These containers work!*

The TODO list is long :-)

* until I broke it 9-12-2021.

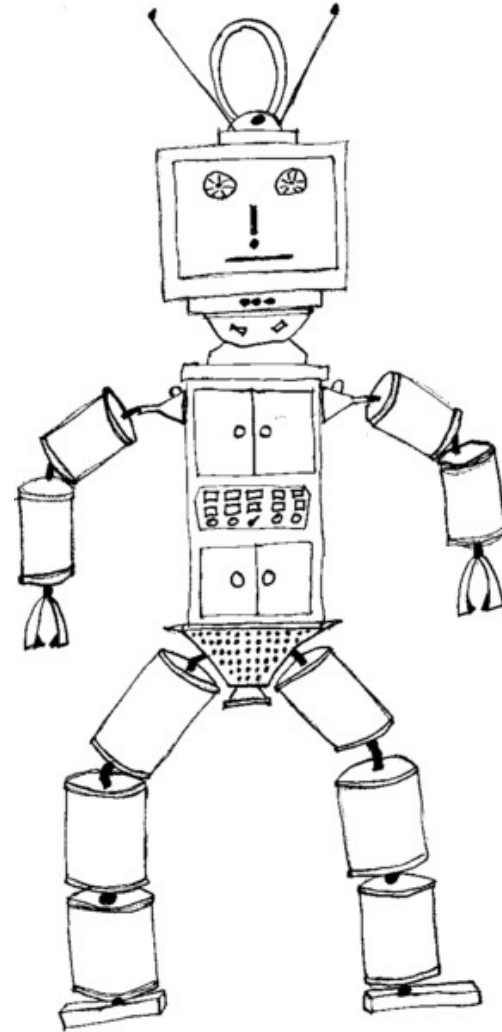
Fix coming soon!



The screenshot shows a GitHub repository interface. At the top, it displays 'main' as the selected branch, '1 branch', and '0 tags'. There are buttons for 'Go to file' and a search icon. Below this, the repository name 'apergos' is shown, followed by a commit message 'remove bash scripts now supplanted by setup_image.py' and a commit hash 'b54a498'. The main content is a list of files and folders with their corresponding commit messages:

File/Folder	Commit Message
docker_helpers	remove bash scripts now supplanted by setup_image.py
.gitignore	add new script setup_image.py for base and final image generation
ARCHITECTURE.md	A giant patchset that restructures everything.
LICENSE	Initial commit
README.md	add new script setup_image.py for base and final image generation
TODOs.md	bring TODO list up to date
config.json.sample	A giant patchset that restructures everything.
daemon.json.sample	A giant patchset that restructures everything.
default.conf	add new script setup_image.py for base and final image generation
docker_dumps_tester.py	add new script setup_image.py for base and final image generation
ssh-to-container.sh	A giant patchset that restructures everything.
testbed_test.py	add new script setup_image.py for base and final image generation

If I want to
keep up on
future
developments,
where do I go?



IRC: apergos (at libera.chat)

Email: ariel@wikimedia.org

Phabricator:

<https://phabricator.wikimedia.org/tag/dumps-generation/>

Mailing list: xmldatadumps-l

Docs:

https://meta.wikimedia.org/wiki/Data_dumps

Repo (temporary):

<https://github.com/apergos/docker-dumps-testbed>

Are we done
yet?



Credits:

https://commons.wikimedia.org/wiki/File:Cartoon_Robot.svg

https://commons.wikimedia.org/wiki/File:Brain_bot.svg

<https://commons.wikimedia.org/wiki/File:Robot-clip-art-book-covers-feJCV3-clipart.png>

https://commons.wikimedia.org/wiki/File:Wikibot_blue.jpg

[https://commons.wikimedia.org/wiki/File:MediaWiki_Bot_\(cropped\).png](https://commons.wikimedia.org/wiki/File:MediaWiki_Bot_(cropped).png)

<https://commons.wikimedia.org/wiki/File:Mini-Robot.png>

[https://commons.wikimedia.org/wiki/File:\(Ink_and_Marker_Robot_Illustration_29\).jpg](https://commons.wikimedia.org/wiki/File:(Ink_and_Marker_Robot_Illustration_29).jpg)

https://commons.wikimedia.org/wiki/File:Korean_Robot.png

https://commons.wikimedia.org/wiki/File:Robot_caillou_ccby_JNLafargue.png

<https://commons.wikimedia.org/wiki/File:Robot.png>

https://commons.wikimedia.org/wiki/File:Trashed_robot.jpg

https://commons.wikimedia.org/wiki/File:African_Robot.png

<https://commons.wikimedia.org/wiki/>

[File:Mark_Zuckerberg_F8_2019_Keynote_\(47721886632\)_\(cropped\).jpg](File:Mark_Zuckerberg_F8_2019_Keynote_(47721886632)_(cropped).jpg)

https://commons.wikimedia.org/wiki/File:Maker_faire_2009_palo_alto_Ariel_Glenn.jpg

<https://commons.wikimedia.org/wiki/File:Road-under-construction.png>

https://commons.wikimedia.org/wiki/File:A_close-up_view_of_a_patch_worn_by_personnel_in_the_B-1B_aircraft_production_acceptance_flight_test_program_-_DPLA_-_18b1fc0e42c0c995a031b703430ac28f.jpeg

Thank You! Keep reusing and sharing!