

eine Vorstellung der
Container-Virtualisierung

DOCKER

und ihr Einsatz in der
Entwicklung von VuFind(2)

Ulf Seltmann <seltmann@ub.uni-leipzig.de>
Webmaster
Projekt finc
Universitätsbibliothek Leipzig



ÜBER MICH

- 2000 bis 2007 Administrator im Web-/Linux-Umfeld
- 2007 Programmierer für PHP im Bereich e-Commerce
- seit April 2013 Programmierer in der UB Leipzig,
erst im Projekt finc, jetzt als Webmaster

WARUM EINE EINHEITLICHE TESTUMGEBUNG

- einfache Installation
- Vermeidung von abweichenden Ergebnissen
- Entwicklungswerkzeuge out-of-the-Box nutzbar
- einfaches Umschalten der Testumgebung

WAS IST DOCKER

- Containervirtualisierung für linuxoide Betriebssysteme
- Images beinhalten die notwendigen Abhängigkeiten eines auszuführenden Programmes
- ein Container ist der isolierte Bereich, in dem das Programm ausgeführt wird
- Ressourcen werden per *kernel namespaces*, *cgroups*, *chroot* und *selinux* vom Host-System isoliert

DOCKER-IMAGES

- leitet ab von Basis-Image
- Dockerfile beschreibt die zusätzliche Installation/Konfiguration
- Image-Änderungen werden revisioniert
- Images werden ver-/ge-teilt über öffentliches oder privates Repository

DOCKER-CONTAINER

- basieren auf einem Image
- startet mit dem Start eines Programms und beendet mit dessen Beendigung
- können Dateien und Ordner des Hostsystems oder anderer Container einbinden
- können Ports an das Hostsystem oder andere Container forwarden

HELLO WORLD!

```
docker run --name "hello-world" ubuntu echo "Hello World!"
```

```
Unable to find image 'ubuntu:latest' locally
Pulling repository ubuntu
91e54dfb1179: Download complete
d3a1f33e8a5a: Download complete
c22013c84729: Download complete
d74508fb6632: Download complete
Status: Downloaded newer image for ubuntu:latest
Hello World!
#$
```

- *run*-Kommando legt die Container-Konfiguration fest

HELLO WORLD! (AGAIN)

```
docker start -i -a hello-world
```

```
Hello World!  
#$
```

- erneute Container-Starts führen den Container mit der Konfiguration aus, die mit dem *run*-Kommando festgelegt wurde

VUFIND2 MIT DOCKER

VuFind-Sourcen klonen

```
git clone https://github.com/vufind-org/vufind.git
```

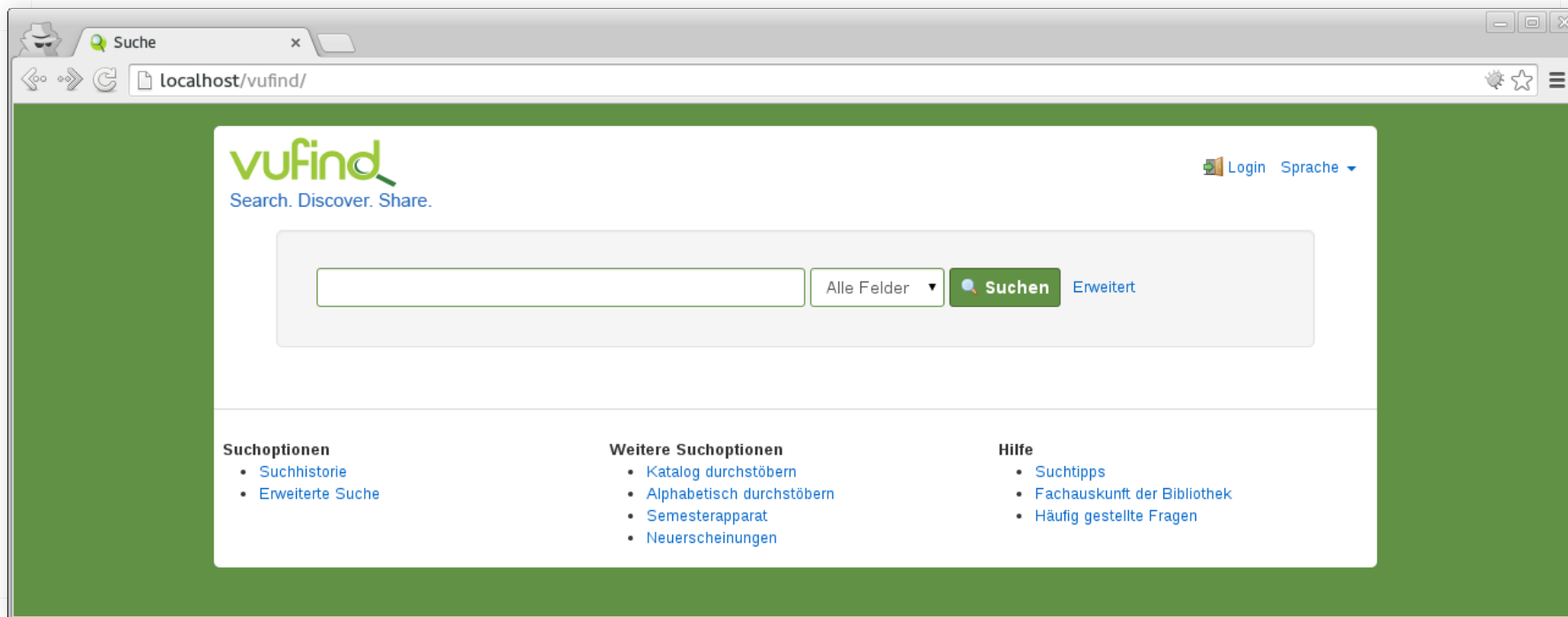
```
Klone nach 'vufind'...  
remote: Counting objects: 106741, done.  
remote: Compressing objects: 100% (20/20), done.  
remote: Total 106741 (delta 6), reused 0 (delta 0), pack-reused 106720  
Empfange Objekte: 100% (106741/106741), 388.15 MiB | 3.83 MiB/s, Fertig.  
Löse Unterschiede auf: 100% (69166/69166), Fertig.  
Prüfe Konnektivität... Fertig.
```

VuFind-Container starten

```
sudo docker run -t -i \  
  -v $(pwd)/vufind:/usr/local/vufind2 \  
  -p 80:80 -p 443:443 -p 8080:8080 -p 3306:3306 \  
  useltmann/vufind2
```

```
Unable to find image 'useltmann/vufind2:latest' locally  
Pulling repository useltmann/vufind2  
dbce69cc9264: Download complete  
4c8cbfd2973e: Download complete  
...  
138278fd23d3: Download complete  
Status: Downloaded newer image for useltmann/vufind2:latest  
Setting up owner/group of dev...  
Setting up runtime environment...  
Setting up 10-mysqld...  
...
```

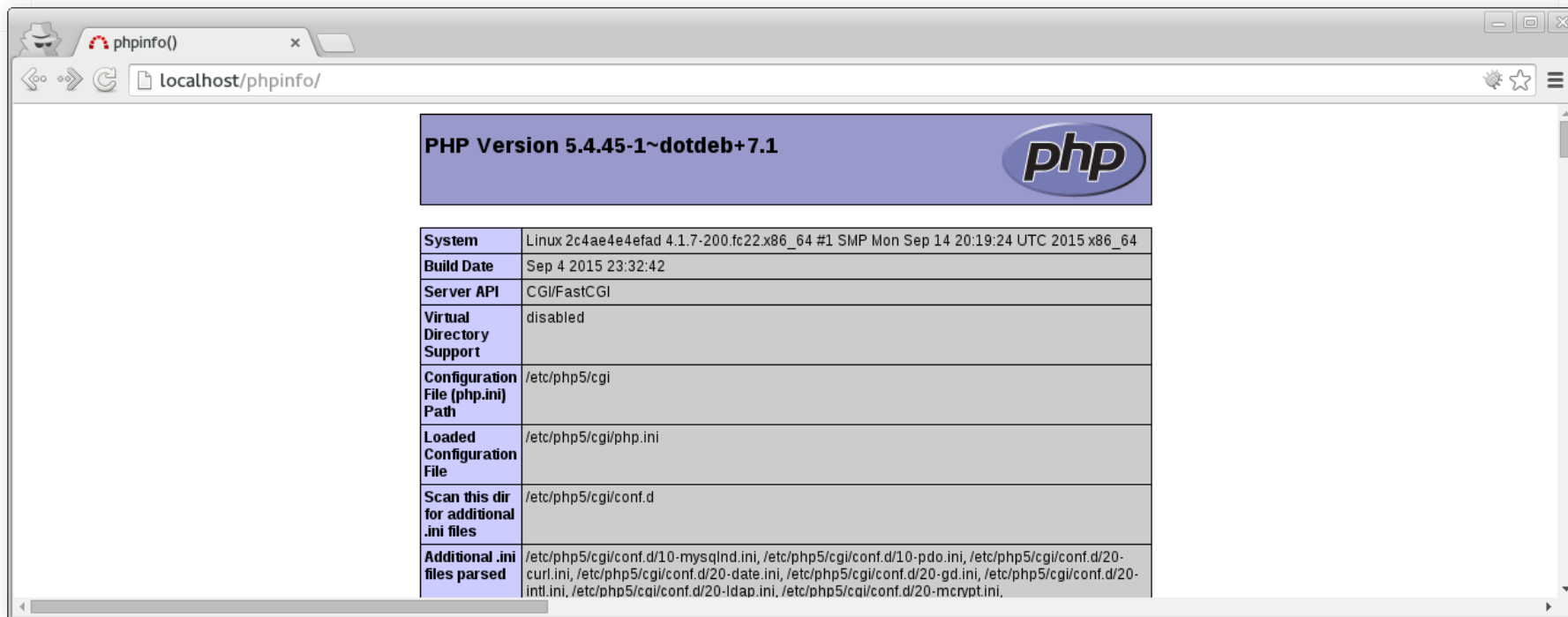
Container aufrufen



`http://localhost/vufind/`

DIE IMAGE-TOOLS

PHP-Einstellungen mit *PHPInfo*



The screenshot shows a web browser window with the address bar displaying `localhost/phpinfo/`. The page content includes a header with the PHP logo and version information, followed by a table of system and configuration details.

PHP Version 5.4.45-1~dotdeb+7.1	
System	Linux 2c4ae4e4efad 4.1.7-200.fc22.x86_64 #1 SMP Mon Sep 14 20:19:24 UTC 2015 x86_64
Build Date	Sep 4 2015 23:32:42
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
Additional .ini files parsed	/etc/php5/cgi/conf.d/10-mysqld.ini, /etc/php5/cgi/conf.d/10-pdo.ini, /etc/php5/cgi/conf.d/20-curl.ini, /etc/php5/cgi/conf.d/20-date.ini, /etc/php5/cgi/conf.d/20-gd.ini, /etc/php5/cgi/conf.d/20-intl.ini, /etc/php5/cgi/conf.d/20-ldap.ini, /etc/php5/cgi/conf.d/20-mcrypt.ini

<http://localhost/phpinfo/>

Bildquelle: <https://commons.wikimedia.org/wiki/User:Eurobas>

Debugging mit XDebug

The screenshot displays the PhpStorm 9.0.1 interface. The main editor shows the file `vufind2 - [~/projects/php/vufind2] - .../module/VuFind/src/VuFind/Bootstrapper.php`. The code is in the `initSession` method, with a breakpoint set at line 112: `if (Console::isConsole()) {`. The code includes comments like `/* Set up the session. This should be done early since other startup routines may rely on session access. */` and `/* @return void */`. The `protected function initSession()` block contains logic to skip session setup in CLI mode and to throw an exception if session configuration is missing.

The Debug console at the bottom shows the execution stack for `index.php`. The current frame is `Bootstrapper.php:112, VuFind\Bootstrapper->initSession()`. The `Variables` pane shows the following state:

- `$this = (VuFind\Bootstrapper) [3]`
- `$_COOKIE = (array) [3]`
- `$_SERVER = (array) [37]`
- `APPLICATION_PATH = "/usr/local/vufind2"`
- `APPLICATION_ENV = "production"`
- `LOCAL_OVERRIDE_DIR = "/usr/local/vufind2/local"`
- `ORIGINAL_WORKING_DIRECTORY = "/usr/local/vufind2/public"`

On the right side, a floating window shows a menu with options: `Debug` (selected), `Profile`, `Trace`, and `Disable`.

Profiling mit XDebug und webgrind

webgrind^{v1.1}
profiling in the browser

Show 90% of Auto (newest) in percent update
 Hide PHP functions

/usr/local/vufind2/public/index.php
cachegrind.out_usr_local_vufind2_public_index_php @ 2015-09-30 14:31:19

1974 different functions called in 446 milliseconds (1 runs, 185 shown)
Show Call Graph

Filter: (regex too)

Function	Invocation Count	Total Self Cost	Total Inclusive Cost
▶ Zend\ServiceManager\ServiceManager->get	861	4.39	108.93
▼ ComposerAutoloadClassLoader->findFileWithExtension	464	3.94	4.30
Calls			
▶ php::is_file @ 354	336	0.28	
▶ php::strpos @ 347	464	0.02	
▶ php::is_file @ 383	24	0.02	
php::strpos @ 352	357	0.01	
php::substr @ 354	336	0.01	
php::strpos @ 370	128	0.00	
php::substr @ 372	128	0.00	
php::substr @ 373	128	0.00	
▶ php::strpos @ 373	128	0.00	
php::strpos @ 381	232	0.00	
Called From			
▶ ComposerAutoloadClassLoader->findFile @ 329	464	4.47	
▶ Zend\MvcRouter\Http\Segment->parseRouteDefinition	184	3.87	4.32
▶ Zend\ServiceManager\ServiceManager->has	1119	3.59	4.36
▶ Zend\ServiceManager\ServiceManager->doCreate	422	3.58	98.58
▶ Zend\ServiceManager\AbstractPluginManager->get	673	3.35	42.70
▶ ComposerAutoload\includeFile	361	2.47	8.11
▶ php::uasort	1	2.43	2.55

Debug
Profile
Trace
Disable

http://localhost/webgrind/

OPCode- und Variable Cache mit XCache

The screenshot displays the XCache 3.2.0 - Cacher web interface. The browser address bar shows `localhost/xcache/cacher/`. The page title is "Cache 3.2.0 - Cacher". The navigation menu includes "Cacher", "Coverager", and "Diagnosis".

The "Summary" tab is active, showing a table of caches:

Caches																		
Cache	Slots	Size	Avail	Percent Graph	Operations	Status	Hits	Hits*24H	Hits/H	Hits/S	Updates	Skips	OOMs	Errs	Protected	Cached	Deleted	GC
php#0	8.00 K	32.00 M	18.56 M		Clear Disable	Normal	123		5.13	7.80	515	0	0	0	no	515	0	
var#0	8.00 K	32.00 M	31.94 M		Clear Disable	-	0		0.00	0.00	0	0	0	0	no	0	0	10

Legends: ■ % Used ■ Used Blocks ■ Hits

Module Info

XCache		
XCache Version	3.2.0	
Modules Built	catcher	
Directive	Local Value	Master Value
<code>xcache.coredump_directory</code>	<i>no value</i>	<i>no value</i>
<code>xcache.disable_on_crash</code>	Off	Off
<code>xcache.experimental</code>	Off	Off
<code>xcache.test</code>	Off	Off
XCache Cacher		
XCache Cacher Module	enabled	
Readonly Protection	disabled	
Page Request Time	2015-09-30 14:35:53	
Cache Init Time	2015-09-30 14:32:01	
Cache Instance Id	685	
Opcode Cache	enabled, 33,554,432 bytes, 1 split(s), with 8192 slots each	
Variable Cache	enabled, 33,554,432 bytes, 1 split(s), with 8192 slots each	
Shared Memory Schemes	mmap	

<http://localhost/xcache/>

PHPUnit

```
sudo docker run --rm -t -i -v $(pwd)/vufind:/usr/local/vufind2 \
    useltmann/vufind2 phpunit
```

```
PHPUnit 4.8.9 by Sebastian Bergmann and contributors.
```

```
....
```

```
Time: 9.07 seconds, Memory: 95.00Mb
```

```
OK, but incomplete, skipped, or risky tests!
```

```
Tests: 801, Assertions: 2137, Skipped: 36.
```


BUILD-SOURCEN AUF GITHUB

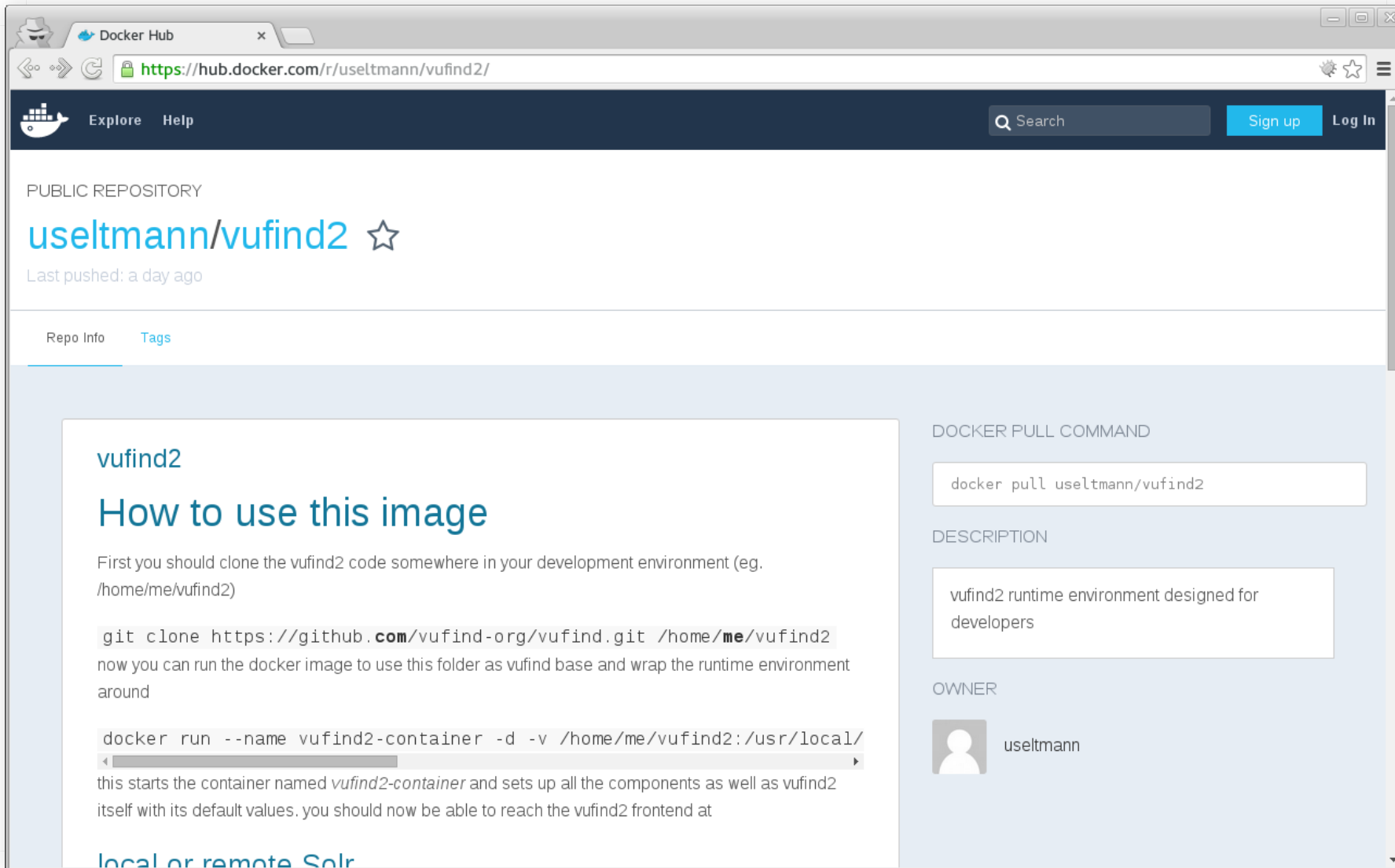
The screenshot shows the GitHub interface for the repository 'finc/docker-vufind2'. The page includes a navigation bar with the GitHub logo, a search bar, and links for 'Explore', 'Features', 'Enterprise', and 'Pricing'. There are 'Sign up' and 'Sign in' buttons. The repository name 'finc / docker-vufind2' is displayed, along with statistics: 8 watchers, 1 star, and 0 forks. A description reads: 'Docker files for creating a docker image that suits our developers needs to a vufind2 runtime environment'. Below this, there are statistics for 11 commits, 1 branch, 0 releases, and 1 contributor. A commit history table is shown, with the most recent commit by 'usetmann' 6 days ago. The commit message is 'fixed typo' and the latest commit hash is '99d504afbd'. The file list includes 'assets', '.dockerignore', '.gitignore', 'Docker.md', 'Dockerfile', 'LICENSE', and 'Readme.md'. On the right side, there are links for 'Code', 'Issues', 'Pull requests', 'Pulse', and 'Graphs'. The 'HTTPS clone URL' is 'https://github.com/finc/docker-vufind2/' and there is a 'Download ZIP' button.

Commit Message	Author	Time
fixed typo	usetmann	6 days ago
* vufind2 is now under /usr/local/vufind2		6 days ago
* solr user is now 'dev'		5 months ago
* solr user is now 'dev'		5 months ago
* vufind2 is now under /usr/local/vufind2		6 days ago
fixed typo		6 days ago
initial commit		8 months ago
* vufind2 is now under /usr/local/vufind2		6 days ago

<https://github.com/finc/docker-vufind2/>

Bildquelle: <http://openitagency.eu/open-source/>

IMAGE AUF DOCKER HUB



DOCKER HUB

Explore Help

Search Sign up Log In

PUBLIC REPOSITORY

useltmann/vufind2 ☆

Last pushed: a day ago

Repo Info Tags

vufind2

How to use this image

First you should clone the vufind2 code somewhere in your development environment (eg. /home/me/vufind2)

```
git clone https://github.com/vufind-org/vufind.git /home/me/vufind2
```

now you can run the docker image to use this folder as vufind base and wrap the runtime environment around

```
docker run --name vufind2-container -d -v /home/me/vufind2:/usr/local/
```

this starts the container named *vufind2-container* and sets up all the components as well as vufind2 itself with its default values. you should now be able to reach the vufind2 frontend at

[local or remote Solr](#)

DOCKER PULL COMMAND

```
docker pull useltmann/vufind2
```

DESCRIPTION

vufind2 runtime environment designed for developers

OWNER

useltmann

<https://hub.docker.com/r/useltmann/vufind2/>

DOCKER



ALL THE THINGS

VIELEN DANK

Ulf Seltmann <seltmann@ub.uni-leipzig.de>

Webmaster

Projekt finc

Universitätsbibliothek Leipzig

UNIVERSITÄTS
BIBLIOTHEK *ubl* LEIPZIG

finc *vufind*
Search. Discover. Share.

- <https://hub.docker.com/r/useltmann/vufind2/>
- <https://github.com/finc/docker-vufind2/>
- <https://finc.info/de/>
- <https://ub.uni-leipzig.de/>