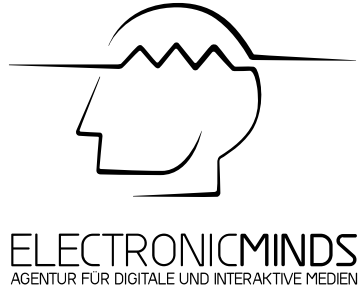


Tame the deployment fear

Deployments mit Deployer

Über uns

- Working @



- PHP Entwickler seit ~10 Jahren
- Triathlet
IRONMAN Hawaii 2019

Über uns

- Working @



- PHP Entwickler seit ~10 Jahren
- Organisator PHP Usergroup Darmstadt

 @WaltherIngo

The fear



<https://imgur.com/gallery/n2XOV8e>

Warum Angst?

- Manuelle Deployments
 - Files werden einzeln deployt
 - Stand auf Produktion ist undefiniert
- Keine (vernünftige) Rollback-Strategie

Our Tamer



Deployer

Was ist Deployer?

“A deployment tool written in PHP with support for popular frameworks out of the box.”

deployer.org

Installation

- Composer Package „deployer/deployer“
- PHAR (<https://deployer.org/download/>)

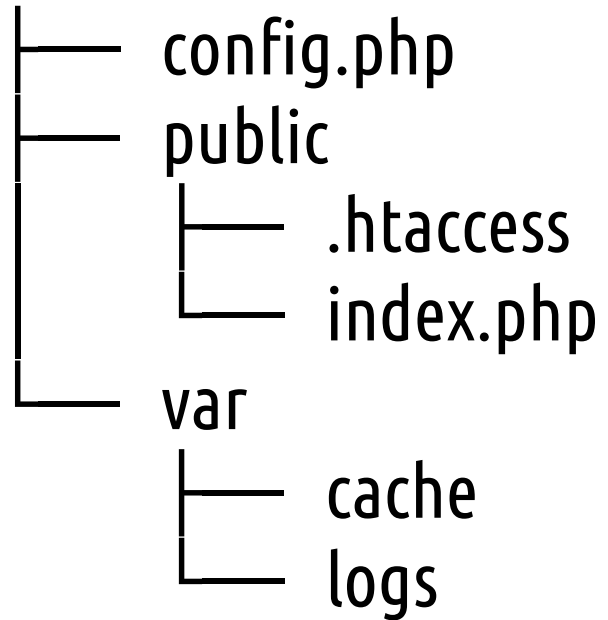
Wie definiere ich mein Deployment?



Beispiel

Minimal-Setup mit einem Beispielprojekt

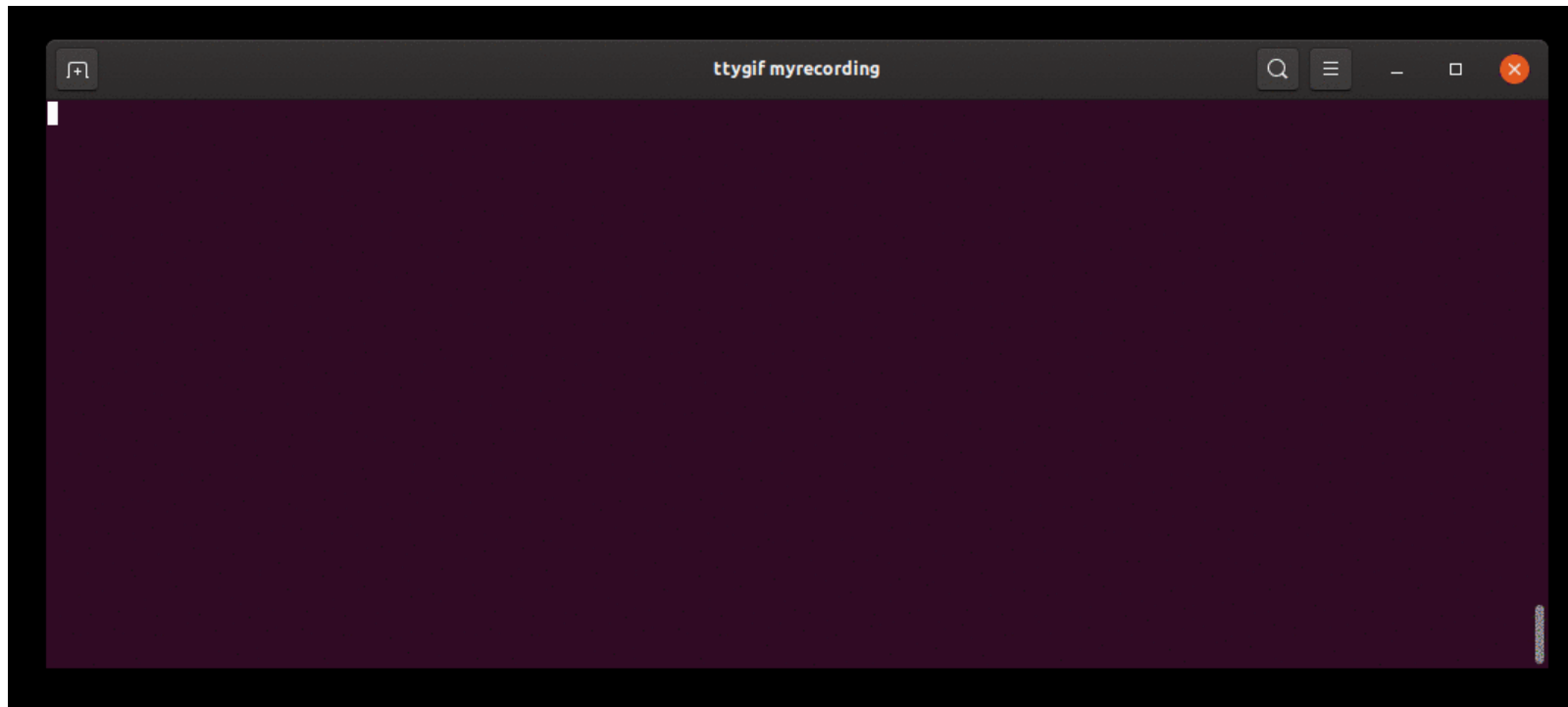
Beispielprojekt



Ziel

- Projekt läuft auf Webserver
- Änderungen können einfach deployt werden
- Deployment erfolgt automatisiert
- Deployment erfolgt standartisiert

Deployer Recipes



Deployer Recipes



```
dep init

Welcome to the Deployer config generator

This utility will walk you through creating a deploy.php file.
It only covers the most common items, and tries to guess sensible defaults.

Press ^C at any time to quit.

Please select your project type [Common]:
[0 ] Common
[1 ] Laravel
[2 ] Symfony
[3 ] Yii
[4 ] Yii2 Basic App
[5 ] Yii2 Advanced App
[6 ] Zend Framework
[7 ] CakePHP
[8 ] CodeIgniter
[9 ] Drupal
[10] TYPO3
> 0
```

Deployer Recipes



```
dep init

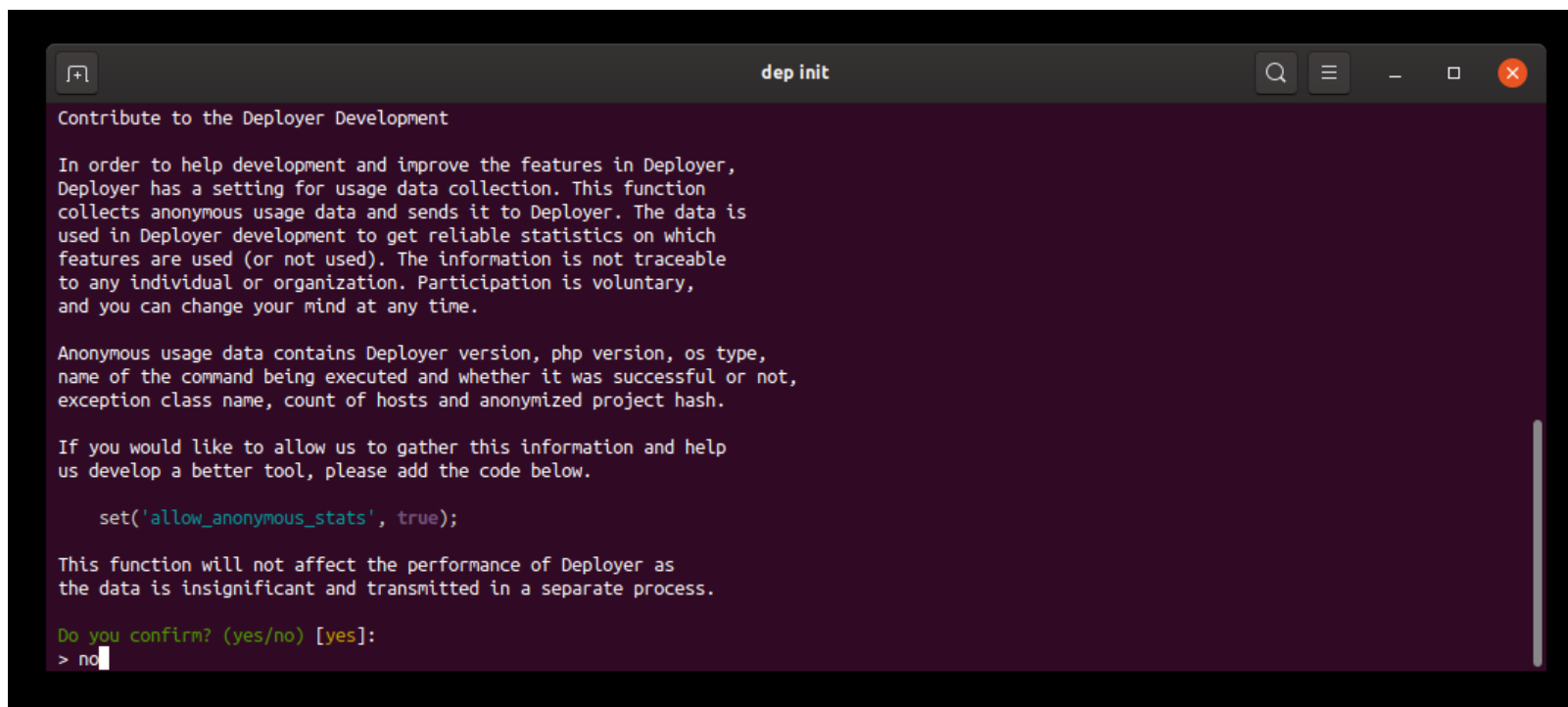
This utility will walk you through creating a deploy.php file.
It only covers the most common items, and tries to guess sensible defaults.

Press ^C at any time to quit.

Please select your project type [Common]:
[0 ] Common
[1 ] Laravel
[2 ] Symfony
[3 ] Yii
[4 ] Yii2 Basic App
[5 ] Yii2 Advanced App
[6 ] Zend Framework
[7 ] CakePHP
[8 ] CodeIgniter
[9 ] Drupal
[10] TYPO3
> 0

Repository [git@git.em.lan:lw/deployer-example.git]:
> 
```

Deployer Recipes



```
dep init

Contribute to the Deployer Development

In order to help development and improve the features in Deployer,
Deployer has a setting for usage data collection. This function
collects anonymous usage data and sends it to Deployer. The data is
used in Deployer development to get reliable statistics on which
features are used (or not used). The information is not traceable
to any individual or organization. Participation is voluntary,
and you can change your mind at any time.

Anonymous usage data contains Deployer version, php version, os type,
name of the command being executed and whether it was successful or not,
exception class name, count of hosts and anonymized project hash.

If you would like to allow us to gather this information and help
us develop a better tool, please add the code below.

    set('allow_anonymous_stats', true);

This function will not affect the performance of Deployer as
the data is insignificant and transmitted in a separate process.

Do you confirm? (yes/no) [yes]:
> no
```


Deployer Recipes



```
iw@em-ub-iw: ~/PhpStormProjects/deployer-example
Deployer has a setting for usage data collection. This function
collects anonymous usage data and sends it to Deployer. The data is
used in Deployer development to get reliable statistics on which
features are used (or not used). The information is not traceable
to any individual or organization. Participation is voluntary,
and you can change your mind at any time.

Anonymous usage data contains Deployer version, php version, os type,
name of the command being executed and whether it was successful or not,
exception class name, count of hosts and anonymized project hash.

If you would like to allow us to gather this information and help
us develop a better tool, please add the code below.

    set('allow_anonymous_stats', true);

This function will not affect the performance of Deployer as
the data is insignificant and transmitted in a separate process.

Do you confirm? (yes/no) [yes]:
> no

Successfully created: /home/iw/PhpStormProjects/deployer-example/deploy.php
iw@em-ub-iw ~/PhpStormProjects/deployer-example master
```

Generiertes Recipe

```

<?php
namespace Deployer;

require 'recipe/common.php';

// Project name
set('application', 'my_project');

// Project repository
set('repository', 'git@git.em.lan:iw/deployer-example.git');

// [Optional] Allocate tty for git clone. Default value is false.
set('git_tty', true);

// Shared files/dirs between deploys
set('shared_files', []);
set('shared_dirs', []);

// Writable dirs by web server
set('writable_dirs', []);
set('allow_anonymous_stats', false);

```

Generiertes Recipe

```
// Hosts
host('project.com')
  ->set('deploy_path', '~/{{application}}');

// Tasks
desc('Deploy your project');
task('deploy', [
  'deploy:info',
  'deploy:prepare',
  'deploy:lock',
  'deploy:release',
  'deploy:update_code',
  'deploy:shared',
  'deploy:writable',
  'deploy:vendors',
  'deploy:clear_paths',
  'deploy:symlink',
  'deploy:unlock',
  'cleanup',
  'success'
]);

// [Optional] If deploy fails automatically unlock.
after('deploy:failed', 'deploy:unlock');
```

Recipe für unser Projekt

```

<?php
namespace Deployer;

require 'recipe/common.php';

// Project name
set('application', 'my_project');

// Project repository
set('repository', 'git@git.em.lan:iw/deployer-example.git');

// [Optional] Allocate tty for git clone. Default value is false.
set('git_tty', true);

// Shared files/dirs between deploys
set('shared_files', []);
set('shared_dirs', []);

// Writable dirs by web server
set('writable_dirs', []);
set('allow_anonymous_stats', false);

```

Recipe für unser Projekt

```

<?php
namespace Deployer;

require 'recipe/common.php';

// Project name
set('application', 'example_project');

// Project repository
set('repository', 'git@git.em.lan:iw/deployer-example.git');

// [Optional] Allocate tty for git clone. Default value is false.
set('git_tty', true);

// Shared files/dirs between deploys
set('shared_files', []);
set('shared_dirs', []);

// Writable dirs by web server
set('writable_dirs', []);
set('allow_anonymous_stats', false);

```

Recipe für unser Projekt

```
<?php
namespace Deployer;

require 'recipe/common.php';

// Project name
set('application', 'example_project');

// Project repository
set('repository', 'git@git.em.lan:iw/deployer-example.git');

// [Optional] Allocate tty for git clone. Default value is false.
set('git_tty', true);

// Shared files/dirs between deploys
set('shared_files', ['config.php', 'public/.htaccess']);
set('shared_dirs', ['var/logs']);

// Writable dirs by web server
set('writable_dirs', []);
set('allow_anonymous_stats', false);
```

Recipe für unser Projekt

```
<?php
namespace Deployer;

require 'recipe/common.php';

// Project name
set('application', 'example_project');

// Project repository
set('repository', 'git@git.em.lan:iw/deployer-example.git');

// [Optional] Allocate tty for git clone. Default value is false.
set('git_tty', true);

// Shared files/dirs between deploys
set('shared_files', ['config.php', 'public/.htaccess']);
set('shared_dirs', ['var/logs']);

// Writable dirs by web server
set('writable_dirs', ['var/cache', 'var/logs']);
set('allow_anonymous_stats', false);
```

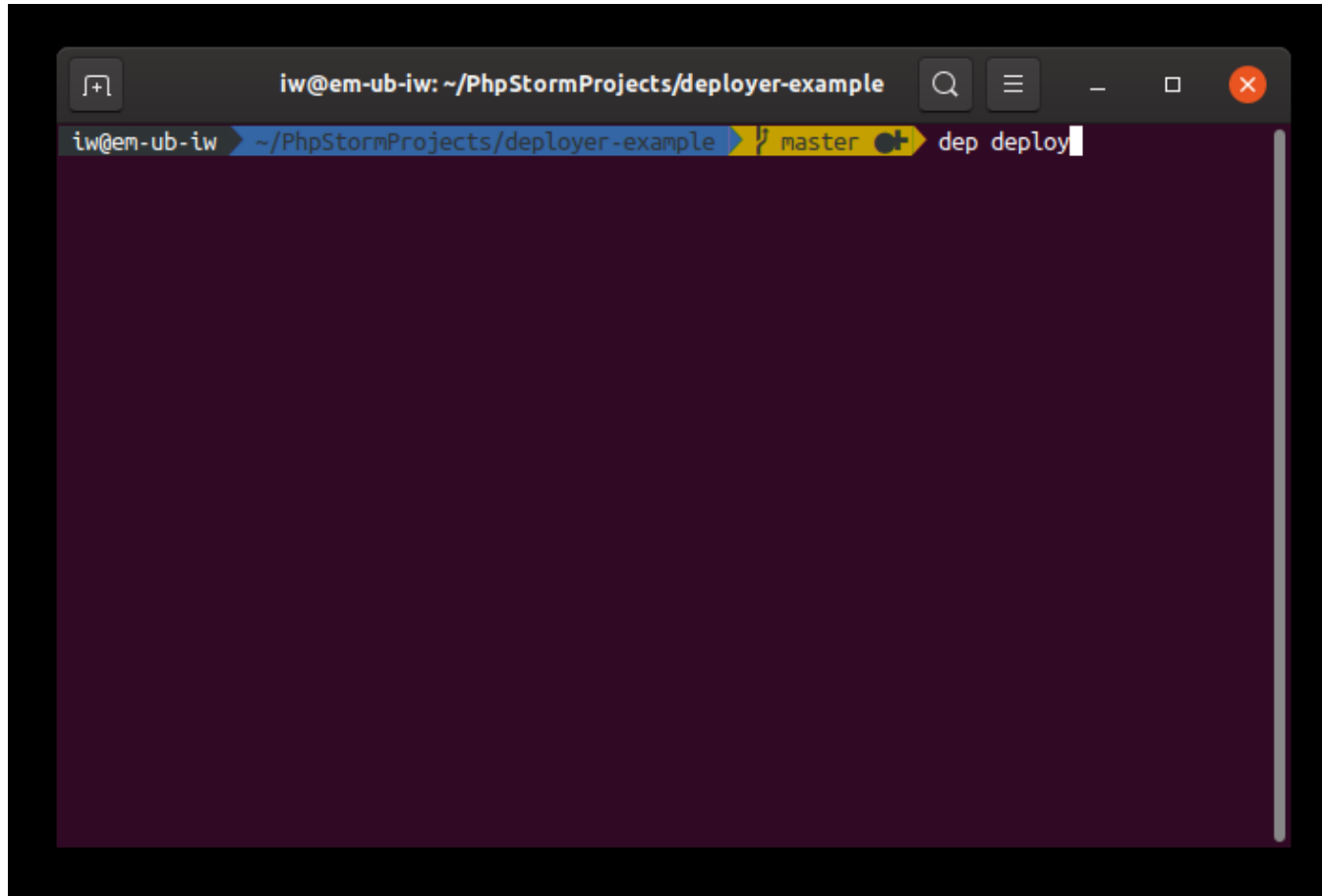
Recipe für unser Projekt

```
// Hosts
host('iw.user.lan')
  ->set('deploy_path', '~/{{application}}');

// Tasks
desc('Deploy your project');
task('deploy', [
  'deploy:info',
  'deploy:prepare',
  'deploy:lock',
  'deploy:release',
  'deploy:update_code',
  'deploy:shared',
  'deploy:writable',
  'deploy:vendors',
  'deploy:clear_paths',
  'deploy:symlink',
  'deploy:unlock',
  'cleanup',
  'success'
]);

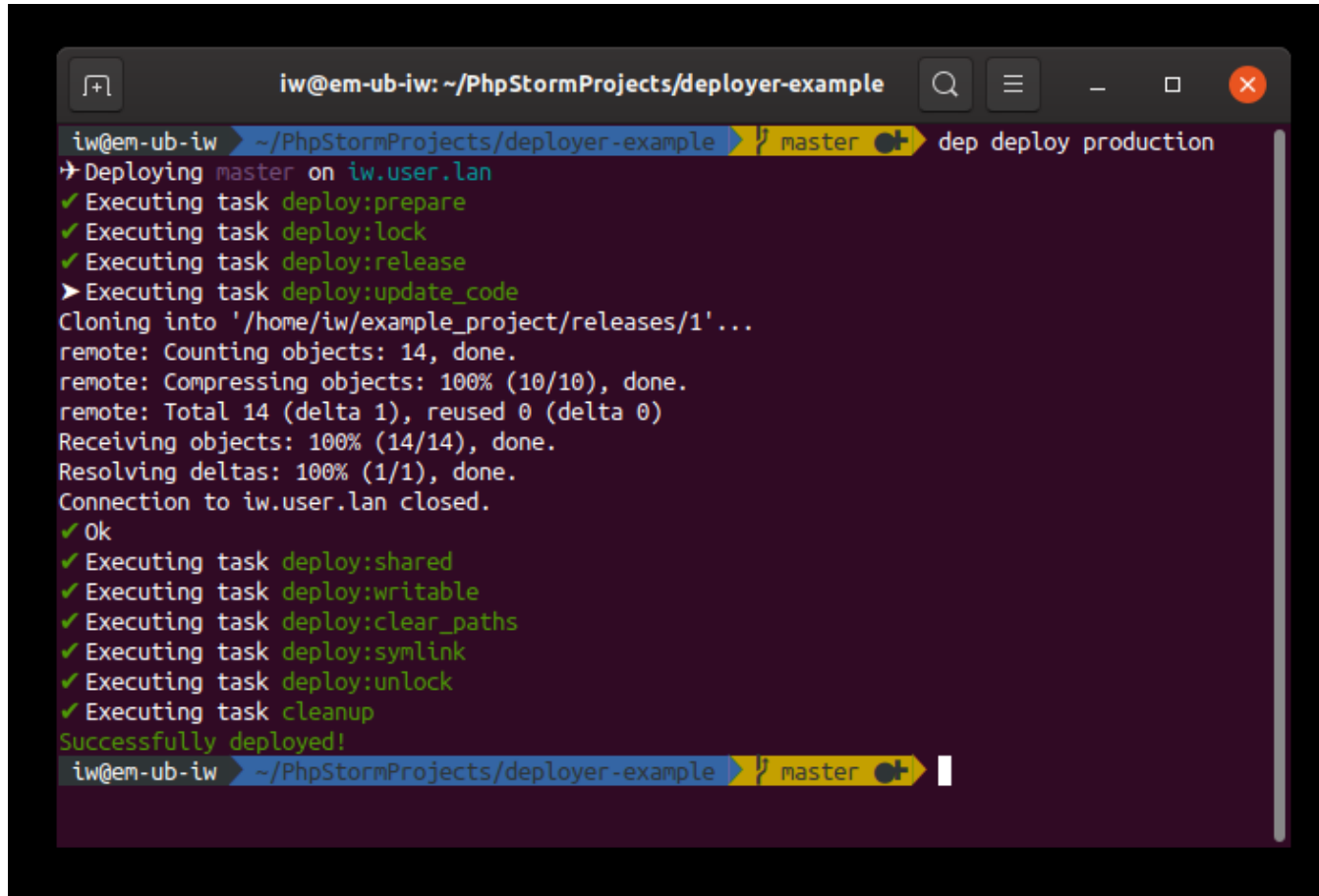
// [Optional] If deploy fails automatically unlock.
after('deploy:failed', 'deploy:unlock');
```


Let's deploy!

A terminal window with a dark background. The title bar shows the user 'iw@em-ub-iw' and the directory '~/PhpStormProjects/deployer-example'. The prompt 'iw@em-ub-iw' is followed by the directory path and a branch indicator 'master' with a plus icon. The command 'dep deploy' is entered at the end of the line.

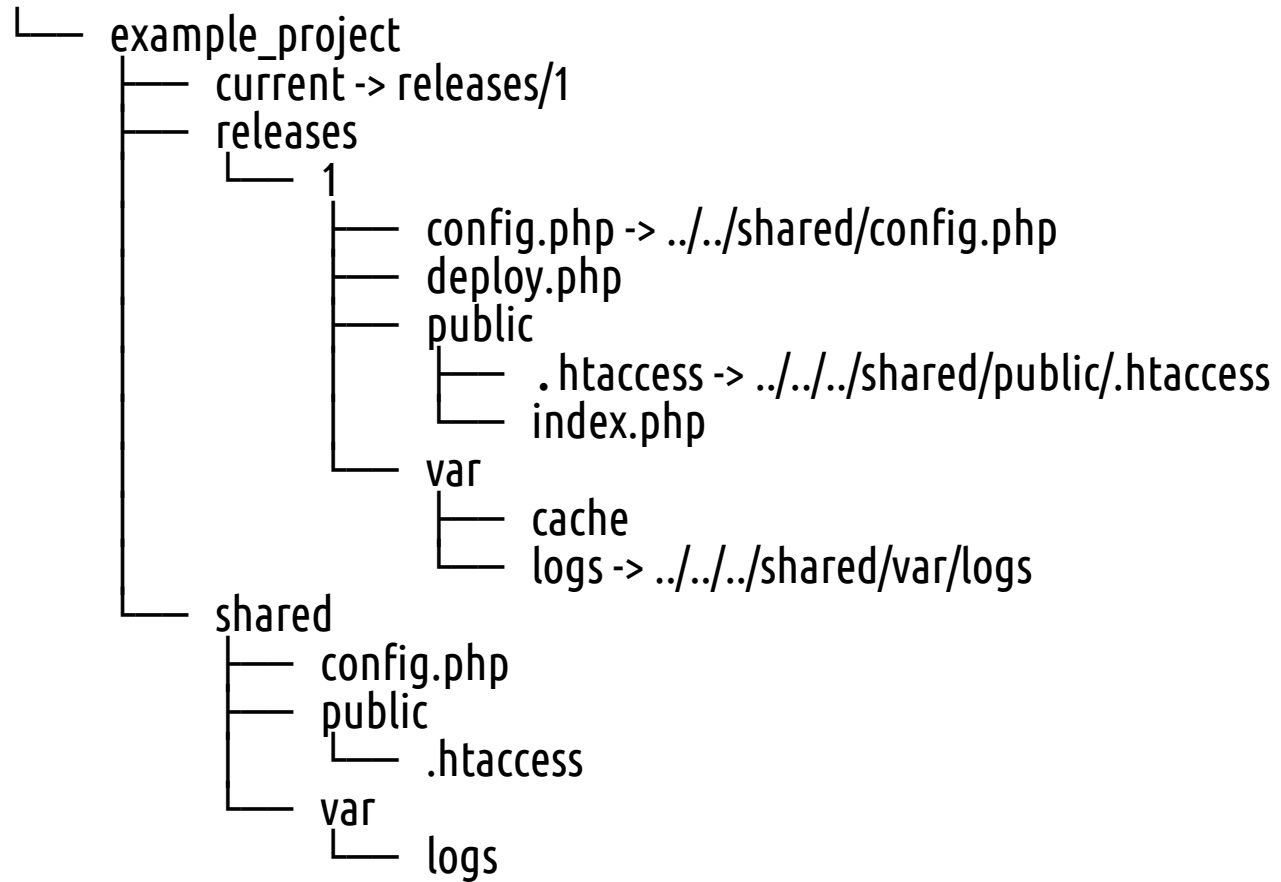
```
iw@em-ub-iw: ~/PhpStormProjects/deployer-example  
iw@em-ub-iw ~/PhpStormProjects/deployer-example master ➤ dep deploy
```

Let's deploy!

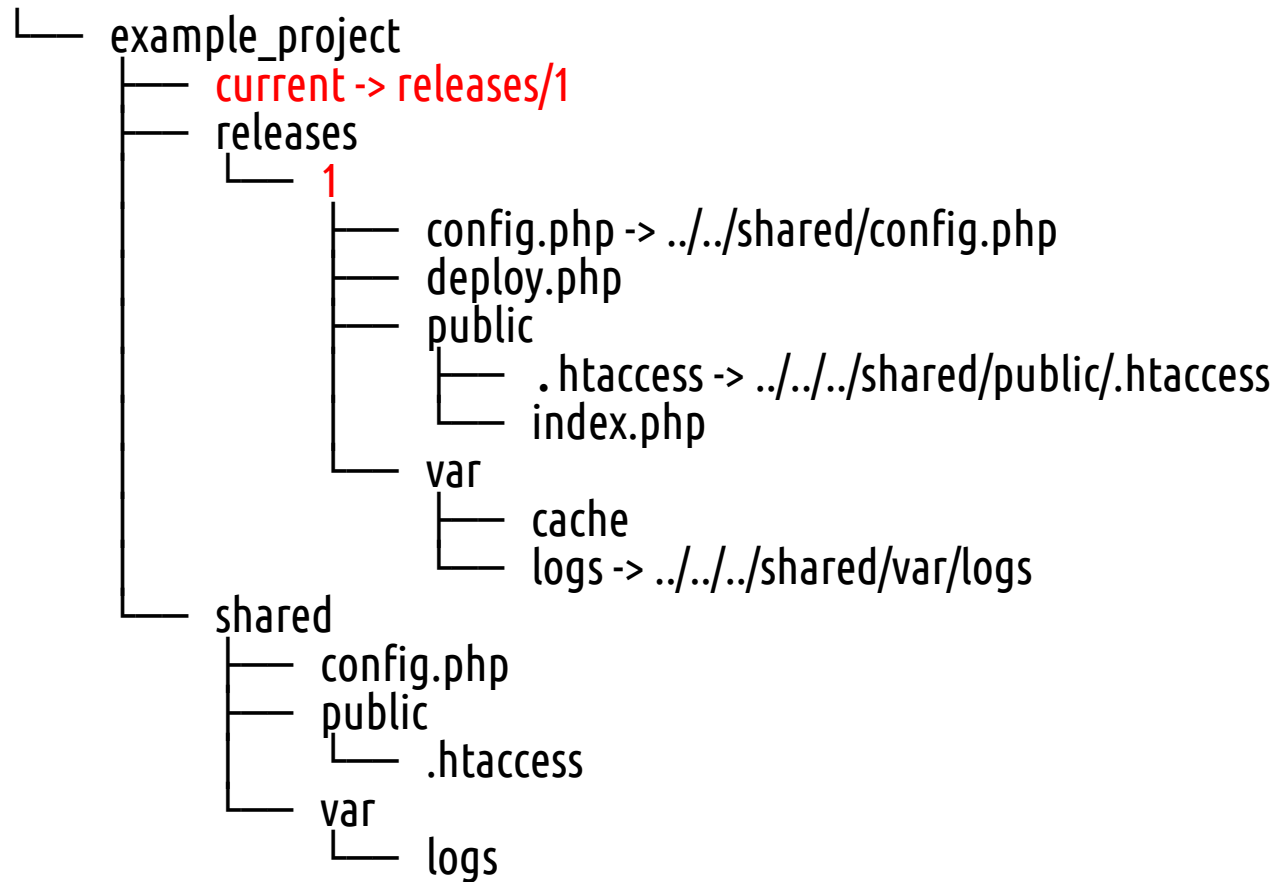
A terminal window titled 'iw@em-ub-iw: ~/PhpStormProjects/deployer-example' with standard window controls. The prompt is 'iw@em-ub-iw' and the command entered is 'dep deploy production'. The output shows a series of tasks being executed: 'Deploying master on iw.user.lan', 'Executing task deploy:prepare', 'Executing task deploy:lock', 'Executing task deploy:release', and 'Executing task deploy:update_code'. The 'update_code' task involves cloning into '/home/iw/example_project/releases/1...', with progress for counting, compressing, and receiving objects. After the connection closes, several more tasks are executed: 'Ok', 'deploy:shared', 'deploy:writable', 'deploy:clear_paths', 'deploy:symlink', 'deploy:unlock', and 'cleanup'. The process ends with 'Successfully deployed!'. The prompt returns to 'iw@em-ub-iw' and the command prompt 'dep' is visible.

```
iw@em-ub-iw: ~/PhpStormProjects/deployer-example
iw@em-ub-iw ➤ ~/PhpStormProjects/deployer-example ➤ master ➤ dep deploy production
➔ Deploying master on iw.user.lan
✓ Executing task deploy:prepare
✓ Executing task deploy:lock
✓ Executing task deploy:release
➤ Executing task deploy:update_code
Cloning into '/home/iw/example_project/releases/1'...
remote: Counting objects: 14, done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 14 (delta 1), reused 0 (delta 0)
Receiving objects: 100% (14/14), done.
Resolving deltas: 100% (1/1), done.
Connection to iw.user.lan closed.
✓ Ok
✓ Executing task deploy:shared
✓ Executing task deploy:writable
✓ Executing task deploy:clear_paths
✓ Executing task deploy:symlink
✓ Executing task deploy:unlock
✓ Executing task cleanup
Successfully deployed!
iw@em-ub-iw ➤ ~/PhpStormProjects/deployer-example ➤ master ➤
```

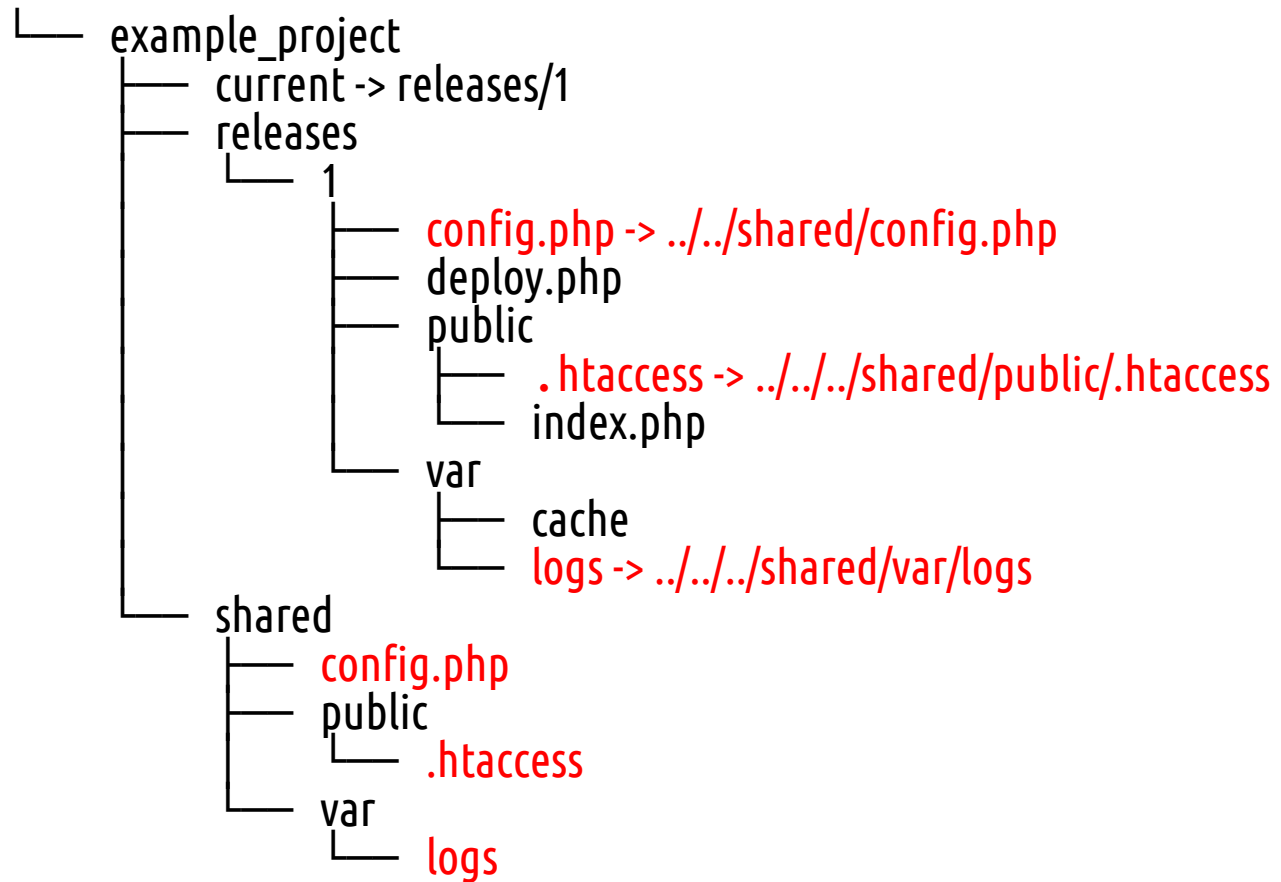
Tadaaaaaa



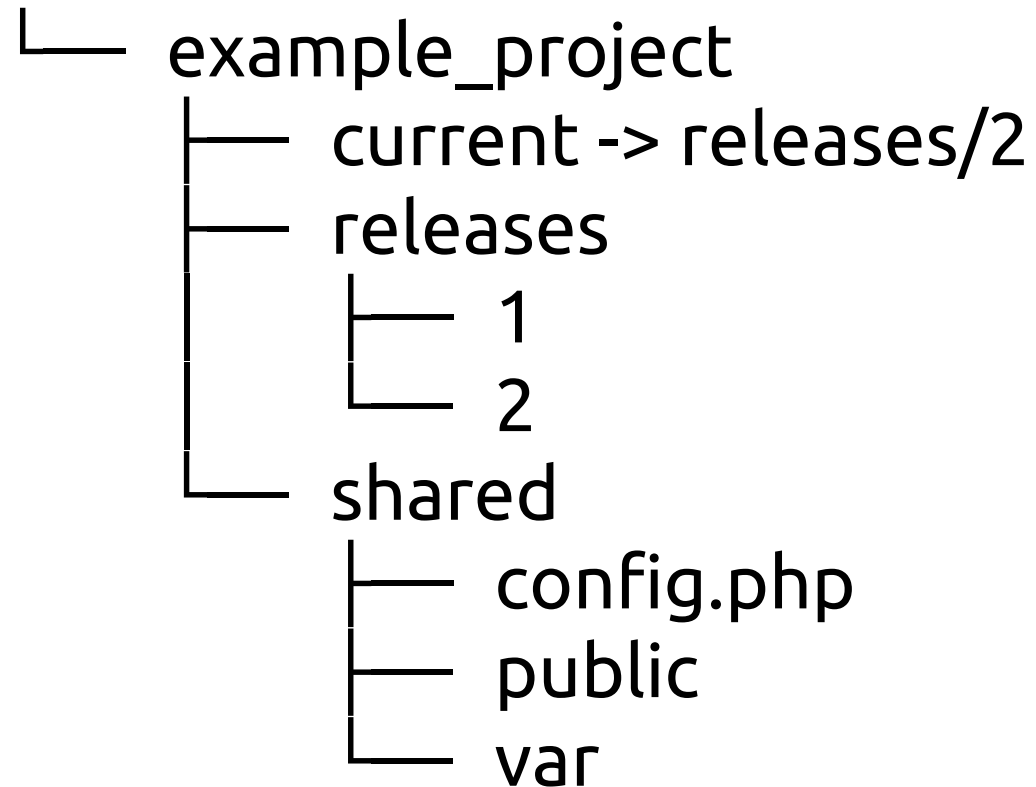
Tadaaaaaa



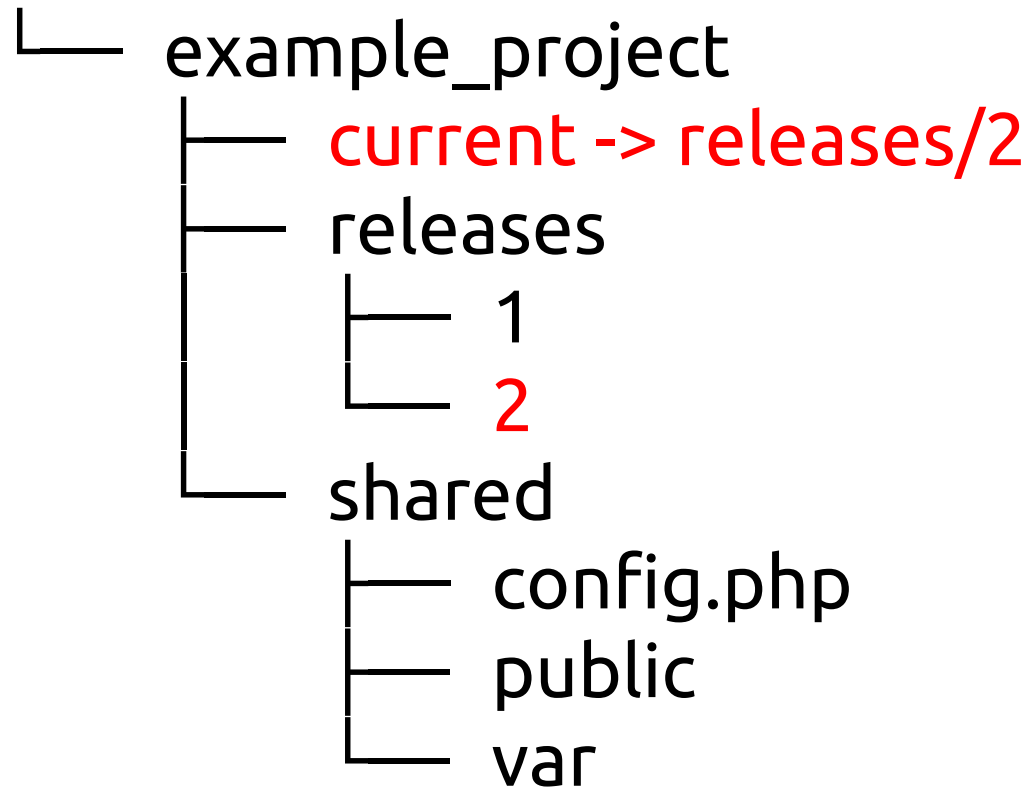
Tadaaaaaa



Let's deploy! Again!



Let's deploy! Again!



Fuck! I broke production!



Keep calm & Rollback!

```
iw@em-ub-iw ➤ ~/PhpStormProjects/deployer-example ➤ master ● ➤ dep rollback  
✓ Executing task rollback
```

Tadaaaaaa (again)

```
└─ example_project
    ├── current -> releases/1
    ├── releases
    │   └─ 1
    └── shared
        ├── config.php
        ├── public
        └── var
```

Any Questions?

Mehrere Zielsysteme

Mehrere Zielsysteme

```
// Hosts
host('iw.user.lan')
  ->stage('production')
  ->set('deploy_path', '~/{{application}}');

host('staging.iw.user.lan')
  ->stage('staging')
  ->set('deploy_path', '~/{{application}}');

// Tasks
desc('Deploy your project');
task('deploy', [
  'deploy:info',
  'deploy:prepare',
  'deploy:lock',
  'deploy:release',
  'deploy:update_code',
  'deploy:shared',
  'deploy:writable',
  'deploy:vendors',
  'deploy:clear_paths',
  'deploy:symlink',
  'deploy:unlock',
  'cleanup',
  'success'
]);

// [Optional] If deploy fails automatically unlock.
after('deploy:failed', 'deploy:unlock');
```

Let's deploy!

```
iw@em-ub-iw > ~/PhpStormProjects/deployer-example > master ● dep deploy
```

In HostSelector.php line 75:

You need to specify at least one host or stage.

Let's deploy!

```
iw@em-ub-iw ~/PhpStormProjects/deployer-example master ● dep deploy production
➔ Deploying master on iw.user.lan
✓ Executing task deploy:prepare
✓ Executing task deploy:lock
✓ Executing task deploy:release
➤ Executing task deploy:update_code
Cloning into '/home/iw/example_project/releases/3'...
Counting objects: 14, done.
Delta compression using up to 6 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (14/14), done.
Total 14 (delta 1), reused 14 (delta 1)
Connection to iw.user.lan closed.
✓ Ok
✓ Executing task deploy:shared
✓ Executing task deploy:writable
✓ Executing task deploy:clear_paths
✓ Executing task deploy:symlink
✓ Executing task deploy:unlock
✓ Executing task cleanup
Successfully deployed!
```

Mehrere Zielsysteme

```
// Hosts
host('iw.user.lan')
  ->stage('production')
  ->set('deploy_path', '~/{{application}}');

host('staging.iw.user.lan')
  ->stage('staging')
  ->set('deploy_path', '~/{{application}}');

// Tasks
desc('Deploy your project');
task('deploy', [
  'deploy:info',
  'deploy:prepare',
  'deploy:lock',
  'deploy:release',
  'deploy:update_code',
  'deploy:shared',
  'deploy:writable',
  'deploy:vendors',
  'deploy:clear_paths',
  'deploy:symlink',
  'deploy:unlock',
  'cleanup',
  'success'
]);

// [Optional] If deploy fails automatically unlock.
after('deploy:failed', 'deploy:unlock');
```


Mehrere Zielsysteme

- Konfiguration der Zielsysteme über yml

```
production:
  hostname: iw.user.lan
  deploy_path: '~/{{application}}'
staging:
  hostname: staging.iw.user.lan
  deploy_path: '~/{{application}}'
```

Mehrere Zielsysteme

```
// Hosts
inventory('hosts.yml');

// Tasks
desc('Deploy your project');
task('deploy', [
    'deploy:info',
    'deploy:prepare',
    'deploy:lock',
    'deploy:release',
    'deploy:update_code',
    'deploy:shared',
    'deploy:writable',
    'deploy:vendors',
    'deploy:clear_paths',
    'deploy:symlink',
    'deploy:unlock',
    'cleanup',
    'success'
]);


// [Optional] If deploy fails automatically unlock.
after('deploy:failed', 'deploy:unlock');
```

Mehrere Zielsysteme

- Konfiguration der Zielsysteme über yml

```
production:
  hostname: iw.user.lan
  deploy_path: '~/{{application}}'
staging:
  hostname: staging.iw.user.lan
  deploy_path: '~/{{application}}'
```

Mehrere Zielsysteme



```
.default: &default
  deploy_path: '~/{{application}}'
production:
  <<: *default
  hostname: iw.user.lan
staging:
  <<: *default
  hostname: staging.iw.user.lan
```

Eigene Tasks

Eigene Tasks

- Tasks werden in PHP definiert

```
// Custom Task
task('deploy:echo_path', function () {
    writeln('Deploy to {{deploy_path}}');
});
```

Eigene Tasks

```
// Tasks
desc('Deploy your project');
task('deploy', [
  'deploy:info',
  'deploy:echo_path',
  'deploy:prepare',
  'deploy:lock',
  'deploy:release',
  'deploy:update_code',
  'deploy:shared',
  'deploy:writable',
  'deploy:vendors',
  'deploy:clear_paths',
  'deploy:symlink',
  'deploy:unlock',
  'cleanup',
  'success'
]);
```

Eigene Tasks

```
iw@em-ub-iw > ~/PhpStormProjects/deployer-example > master dep deploy production
→ Deploying master on iw.user.lan
▶ Executing task deploy:echo_path
Deploy to ~/example_project
✓ Ok
✓ Executing task deploy:prepare
✓ Executing task deploy:lock
✓ Executing task deploy:release
▶ Executing task deploy:update_code
Cloning into '/home/iw/example_project/releases/4'...
Counting objects: 14, done.
Delta compression using up to 6 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (14/14), done.
Total 14 (delta 1), reused 14 (delta 1)
Connection to iw.user.lan closed.
✓ Ok
✓ Executing task deploy:shared
✓ Executing task deploy:writable
✓ Executing task deploy:clear_paths
✓ Executing task deploy:symlink
✓ Executing task deploy:unlock
✓ Executing task cleanup
Successfully deployed!
```


Let's celebrate!



Summary of benefits

- Standardisierte Struktur
- Wiederverwendbarkeit
- „One-Click“ Deployment
- Einfache Rollbacks
- Definition in PHP
- Grundrezepte für Frameworks

Disclaimer

- Was nimmt Deployer uns nicht komplett ab?
 - Datenbankmigrationen
- Grundvoraussetzung für angstfreies Deployment?
 - Logging
 - Monitoring
 - Automatisierte Tests

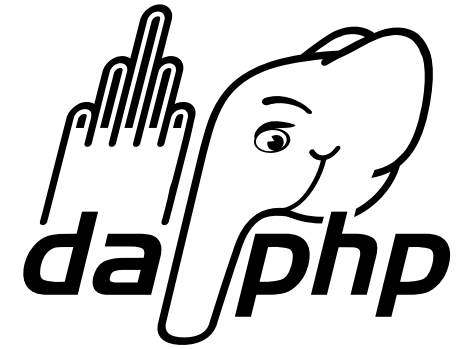
“If you still don’t have tests, this is unprofessional. This is engineering, not banging rocks together.”

Marco Pivetta (26.09.2019)

Danke für eure Aufmerksamkeit!



PHP Usergroup Darmstadt



Termine

- Morgen auf dem Weihnachtsmarkt
- 12.02.20 erstes Treffen bei Electronic Minds

Meetup

