Deployer gebruiken in plaats van .gitlab ci.yml

Info over hoe je Deployer kan gebruiken om Drupal websites te deployen (van lokaal naar productie op Combell)

- Instructies

Instructies

Eerst deployer toevoegen aan je project:

```
ddev composer require deployer/deployer --dev
```

Dan `deploy.php` kopieren vanuit starterkit en de nodige aanpassingen doen. Voor subsites moet je na succesvolle deployment in my combell zelf de subsite bewerken en de servermap `current/web` selecteren ipv `/web`.

Vergeet niet om `.gitlab-ci.yml` te hernoemen naar bv `.gitlab-ci.yml.old` ofzo zodat er geen gitlab deployments meer zijn.

Om een volledige deploy te doen, kies een van:

```
ddev exec dep deploy
ddev exec dep deploy -v
ddev exec dep deploy -vvv
```

Om te deployen naar development (als die host is gedefiniëert):

```
ddev exec dep deploy development
```

Om een snelle deploy te doen die enkel gewijzigde files synchroniseert (NIET contrib)

```
ddev exec dep fast-deploy
```

Om een deploy te doen die geen nieuwe release maakt enz maar wel files synct en dan drush deploy doet:

```
ddev exec dep medium-deploy
```

Als deployment succesvol is en alles werkt, dan mag in principe de hele /production folder verwijderd worden.

```
chmod -R 777 production
rm -rf production
```

Allereerste keer moet data uiteraard gekopieerd worden van huidige /production folder:

```
ssh nukamelcom@ssh063.webhosting.be
mkdir shared
mkdir shared/web
mkdir shared/web/sites
mkdir shared/web/sites/default
cp production/web/sites/default/settings.db.php shared/web/sites/default
mkdir shared/web/sites/default/files
cp -r production/web/sites/default/files/* shared/web/sites/default/files
mkdir shared/tmp
cp -r production/tmp/* shared/tmp
mkdir shared/private
cp -r production/private/* shared/private
```

Vergeet niet om ook in crontab eventuele paden aan te passen van `/production` naar `/current`. Ook mag je `--uri=...` weglaten: dat is in orde dankzij drush.yml file die wordt aangemaakt via deployments.

```
nano /etc/crontab
```

Voeg volgende toe in .gitignore, dat gaan we custom genereren per hosting tijdens deployments:

```
/drush/drush.yml
```

Ook mag de vendor folder uit git verwijderd worden. Eerst toevoegen aan `.gitignore`:

```
/vendor
```

Daarna:

```
git add . && git commit -am "added /vendor into .gitignore" && git push
git rm -rf vendor && git add . && git commit -am "removed vendor folder from git" && git push
ddev composer install
```

Via SSH een subsite opkuisen die omgezet is naar deployments via deployer

```
cd /data/sites/web/henradeu/subsites/dev.henrad.eu
rm -rf .ddev .vscode assets config drush img private recipes scripts vendor web
rm -f .deploy.gitignore .editorconfig .gitattributes .gitlab-ci.back.yml auth.json
composer.json composer.lock composer.lock.old gulpfile.js.old package.json.old
package.json.old
```