MAGICAL WORDPRESS MANAGEMENT USING WP-CLI

WordCamp San Francisco 2013

Mike Schroder | @GetSource
Developer, WordPress Specialist at DreamHost
Mike Schroder, a.k.a. DH-Shredder, a.k.a @GetSource

CREDENTIALS
Happy DreamHost Employee
WordPress Core Contributor
WordPress 3.5 Recent Rockstar
Trusted WP-CLI Feature Contributor

RANDOM FACTS
Third Culture Kid
Enjoys Coffee & Sailing
Blogs at getsource.net
THERE ARE TWO GROUPS OF PEOPLE.
THOSE WHO USE THE COMMAND LINE
THOSE WHO ARE GOING TO USE THE COMMAND LINE
DON’T BE AFRAID OF THE CLI
IT’S YOUR FRIEND.
OH, YOU LIKE THE CLI?

WP-CLI WILL MAKE YOUR LIFE BETTER
SUPER COOL OPEN SOURCE TOOL
TO MANAGE WORDPRESS
HEADED UP BY CRISTI BURCĂ (SCRIBU) AND ANDREAS CRETEN
USES WORDPRESS ITSELF TO PERFORM OPERATIONS
AUTOMATION!
WHAT CAN I **DO** WITH IT?
MANAGE

ALL THE THINGS
NO, REALLY.
UPDATE WORDPRESS

$ wp core update
INSTALL A PLUGIN

$ wp plugin install <slug or .zip>
$ wp theme activate <slug>
BACKUP YOUR DATABASE

$ wp db export [filename]
UPDATE PLUGINS

$ wp plugin update-all
UPDATE THEMES

$ wp theme update-all
REPLACE TEXT IN DB

$ wp search-replace <old> <new>
WHAT IS THIS SORCERY?
I DON'T ALWAYS TEST MY CODE

BUT WHEN I DO, I DO IT IN PRODUCTION.
SOLD. HOW DO I GET THIS AWESOMENESS?
WHAT DO I NEED TO RUN IT?

- SSH access to your WordPress install's directory
- PHP 5.3.2+
- WordPress 3.4+
- Enough RAM for shell processes to run WordPress
- Linux or MacOS (for now)
INSTALL IN LOCAL USER

Full Instructions at:
wp-cli.org

BUT WAIT, THERE’S MORE!
WP-CLI IS EXTENSIBLE!
ADD YOUR OWN **LOCALLY**

Run/Alias:

```
$ wp --require='cmd_name.php'
```

wp-cli.yml

```
require: /path/to/cmd_name.php
```
OR IN YOUR PLUGINS

```php
if ( defined('WP_CLI') && WP_CLI ) {
    include( PLUGIN_DIR . '/lib/wp-cli.php' );
}
```
CUSTOM COMMAND: BACKUP!

```
$ wp migrate backup [file.tar.gz]
[--no-db] [--db-name=<file.sql>]
```
CUSTOM COMMAND: BACKUP!

Strategy

- Use Built-in SQL Backup Command
- Create .tar.gz of install’s files and database.
<?php

WP_CLI::add_command( 'migrate', 'DH_Migrate_Command' );

/**
 * DreamHost Migrate Plugin
 * @package DH_Migrate_Command
 * @subpackage commands/community
 * @maintainer Mike Schroder
 */
class DH_Migrate_Command extends WP_CLI_Command {

// ...
DEFINE YOUR SUBCOMMAND

$\text{args}$
Stand-alone arguments

$\text{assoc\_args}$

\texttt{--arg=value} style in associative array

```php
/**
 * Backup entire WordPress install, including core, plugins and database.
 *
 * @param array $\text{args}$
 * @param array $\text{assoc\_args}$
 * @synopsis [backup_filename] [--no-db] [--db-name=<filename>]
 */

function backup( $\text{args}$, $\text{assoc\_args}$ ) { 
  // ...
```

Mike Schroder | @GetSource | #wcsf 2013
DEFINE YOUR SUBCOMMAND

@synopsis
WP-CLI specific comment to automatically force proper CLI syntax

/**
 * Backup entire WordPress install, including core, plugins and database.
 *
 * @param array $args
 * @param array $assoc_args
 * @synopsis [backup_filename] [--no-db] [--db-name=<filename>]
 */

function backup( $args, $assoc_args ) {
    // ...

function backup( $args, $assoc_args ) {
    $filename = $dbname = null;
    $backup_directory = '../';

    // If a filename isn't specified, default to "Site's Title.tar.gz".
    if ( empty( $args ) )
        $filename = $backup_directory . get_bloginfo() . '.tar.gz';
    else
        $filename = $args[0];

    // ...

// If --no-db is specified, don't include the database in backup
if ( ! isset( $assoc_args['no-db'] ) ) {

    $dbname = isset( $assoc_args['db-name'] ) ?
        $assoc_args['db-name'] : 'database_backup.sql';

    WP_CLI::run_command(
        array( 'db', 'export', $backup_directory . $dbname), array() );
}

// ...
// Using esc_cmd to automatically escape parameters.
// We can't use --exclude-vcs, because it's not available on OSX.
WP_CLI::line( "Backing up to $filename ..." );
$result = WP_CLI::launch( \\WP_CLI\Utils\esc_cmd( "
  tar
  --exclude '.git' \
  --exclude '.svn' \
  --exclude '.hg' \
  --exclude '.bzr' \
  -czf %s . -C %s %s \
  ", $filename, $backup_directory, $dbname ), false );

// ...
// If we created a database backup, remove the temp file.
if ( $dbname && !unlink( $backup_directory . $dbname ) )
    WP_CLI::warning(
        "Couldn't remove temporary database backup, '$dbname'." );

if ( 0 == $result ) {
    WP_CLI::success( "Backup Complete." );
} else {
    WP_CLI::error( "Backup Failed." );
}

} // End backup

} // End DH_Migrate_Command
FURTHER LEARNIFICATION:

WP-CLI:
- wp-cli.org
- getsource.net/tag/wp-cli/
- halfelf.org/2012/command-line-wp

CLI Cheat Sheet: