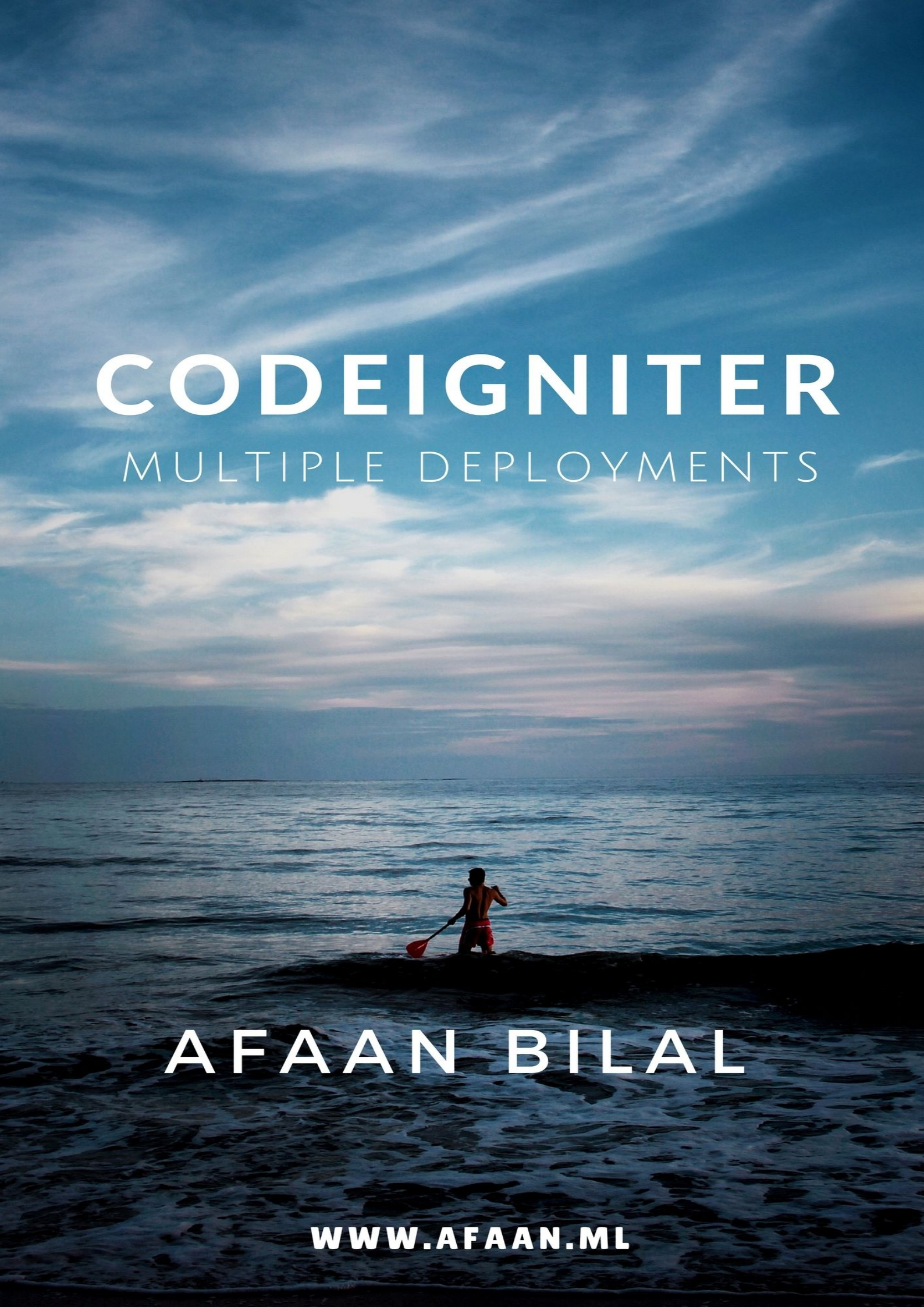
CODEIGNITER
MULTIPLE DEPLOYMENTS
AFAAN BILAL
WWW.AFAAN.ML

Table of Contents

CodeIgniter: Multiple Deployments

by Afaan Bilal

You've written a nice application using CodeIgniter and want to deploy for multiple clients. This guide will walk you through setting up your CodeIgniter application for multiple deployments. All deployments will be on the same server using the same application and system folders.

Let's start.

Formats available:

- Online
- PDF
- ePub
- Mobi

The Directory Structure

The directory structure of our application will be somewhat as follows:

```
application/
system/
public_html/
  assets/
  sub-domain-1.example.com/
    assets/
    .htaccess
    index.php
  sub-domain-2.example.com/
    assets/
    .htaccess
    index.php
  ...
  sub-domain-n.example.com/
    assets/
    .htaccess
    index.php
```

In the above structure:

- `application` is our application directory
- `system` contains the CodeIgniter system files, and
- `public_html` will be the *document root* of our domain `example.com`.
- `assets` directory inside our document root will hold the common assets for our application instances.
- `sub-domain-1.example.com` through `sub-domain-n.example.com` will hold our front controllers for each instance of our application. Set the document roots of the sub domains to these directories.
- `assets` directory inside each of the `sub-domain-n.example.com` will hold any instance specific assets that we may need (Optional).
- `.htaccess` inside our each of the `sub-domain-n.example` will help us remove the `index.php` from our URLs as well as implement a domain filter.

Now, we'll walk through the files that need to be modified one-by-one.

The .htaccess

Here's the code for our `.htaccess`:

```
# Disable directory listing
Options All -Indexes

# Domain filter
RewriteEngine on
RewriteCond %{HTTP_HOST} !^sub-domain-n\.example\.com$ [NC]
RewriteRule ^ - [F]

# Remove index.php from URLs
RewriteCond $1 !^(index\.php|assets|robots\.txt)
RewriteRule ^(.*)$ index.php/$1 [L]
```

The first rule - the domain filter - blocks all traffic except for the specified sub domain. The second rule removes the `index.php` from our URLs. It also allows direct requests to `robots.txt` (if we have one) and the `assets` directory. All other requests are routed through to `index.php`.

The index.php

The `index.php` is our front-controller for each application instance. Here are the things to change:

The system path

```
$system_path = '../../system';
```

The application path

```
$application_folder = '../../application';
```

Custom CONFIG

```
$assign_to_config['base_url'] = 'https://sub-domain-n.example.com';
// your database group (see next chapter)
$assign_to_config['db_group'] = 'subdomain-n';
$assign_to_config['encryption_key'] = 'my123secret456key';
$assign_to_config['cookie_domain'] = '.sub-domain-n.example.com';
```

The database.php

The database configuration of each instance is kept in `application/config/database.php` in a separate `db_group`. Since we have assigned the `db_group` in our front-controller, we just need to set the same in our `database.php`.

Here is our `database.php`:

```
$active_group = config_item('db_group');
$query_builder = TRUE;

$db['subdomain-1'] = array(
    'dsn'       => '',
    'hostname' => 'localhost',
    'username' => 'user_1',
    'password' => 'pass_1',
    'database' => 'db_1',
    'dbdriver' => 'mysqli',
    // ...
    'save_queries' => FALSE
);

$db['subdomain-2'] = array(
    'dsn'       => '',
    'hostname' => 'localhost',
    'username' => 'user_2',
    'password' => 'pass_2',
    'database' => 'db_2',
    'dbdriver' => 'mysqli',
    // ...
    'save_queries' => FALSE
);

// ... so on for every instance
```

The constants.php

We just need to add two custom constants in our `application/config/constants.php` for our common assets.

```
defined('ASSETS_BASE') OR define('ASSETS_BASE', 'https://www.example.com/assets/');
defined('ASSETS_PATH') OR define('ASSETS_PATH', '../assets/');
```

The `ASSETS_BASE` constant is for our `stylesheets`, `javascripts`, `images`, etc. You will need to prepend all of your asset links (css, js, img, etc) with this constant.

For example:

```
<link rel="stylesheet" type="text/css" href="<?=ASSETS_BASE;?>css/styles.css" />
<img src="<?=ASSETS_BASE;?>images/my-image.jpg" />
```

The `ASSETS_PATH` constant is for our `php` scripts to access any files within the `assets` folder. Use it for upload paths etc.

Conclusion

That's all there is to deploying multiple instances of your CodeIgniter application.

Now, you only need to update once and all your deployed instances will use that updated file. This reduces code redundancy and makes maintenance extremely easy.

In this guide, we have used sub-domains as our deployment points but there is no hard and fast rule that requires you to do this. You may deploy the individual instances in normal directories. However, you will want to remove the domain filter from your `.htaccess` and update your `base_url` accordingly.

Thank you for reading!