Drupal™

DevOps 環境 DDEV の基礎を学ぶハンズオン

Drupal をやさしく学ぶ勉強会 2019 年 10 月

改訂版

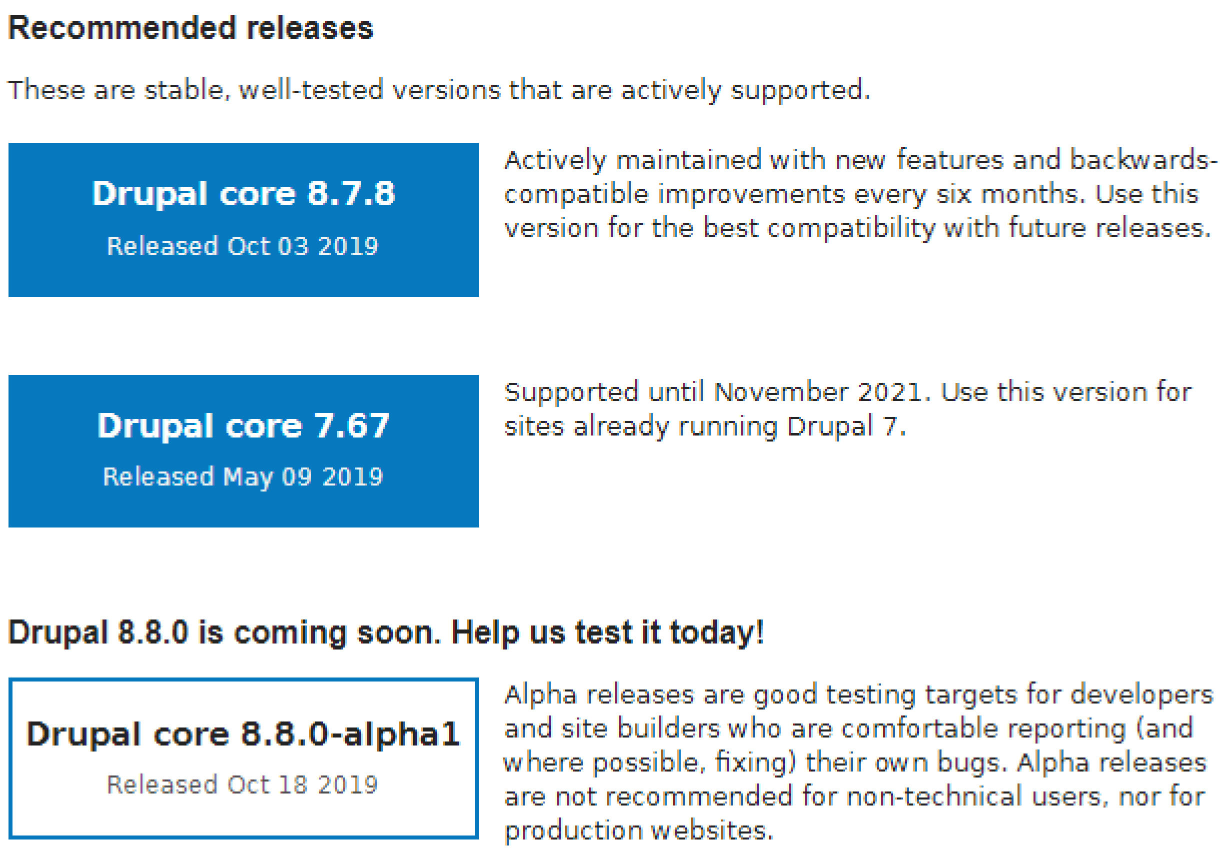
開催日　2019 年 10 月 23 日(水)　19:00-21:00

目 次

はじめに

Drupal 7＆8のコア

10 月 23 日(水) https://www.drupal.org/project/drupal



Recommended releases

These are stable, well-tested versions that are actively supported.

Drupal core 8.7.8
Released Oct 03 2019

Actively maintained with new features and backwards-compatible improvements every six months. Use this version for the best compatibility with future releases.

Drupal core 7.67
Released May 09 2019

Supported until November 2021. Use this version for sites already running Drupal 7.

Drupal 8.8.0 is coming soon. Help us test it today!

Drupal core 8.8.0-alpha1
Released Oct 18 2019

Alpha releases are good testing targets for developers and site builders who are comfortable reporting (and where possible, fixing) their own bugs. Alpha releases are not recommended for non-technical users, nor for production websites.

トピックス

Drupal Global Training Day & First time Sprinters 2019 年 12 月 7 日（土）

初めての方向けに Drupal をハンズオン体験する 1 日コースです。無料です。既に Drupal を使っている方も参加できるもくもく会まコースもあります。また、Global Sprint と Training を一緒に開催することになりました。自分で作成したモジュールやデザインテンプレートなどを Drupal.org サイトにアップロード登録する手順なども学びます。

申し込みサイトは　近日中にオープンします。
ぜひ。ご参加ください。

API-First Decoupled Drupal Camp Tokyo 2019 開催！

12 月 13 日（金）と 14 日（土）に JR 品川駅高輪口下車 3－4 分の会議室で Decoupled 関連のハンズオンワークショップとセッションを開催します。ハンズオン 5000 円前後、セッション　5000 円前後、どちらの日も、お弁当、懇親会の費用が含まれています。

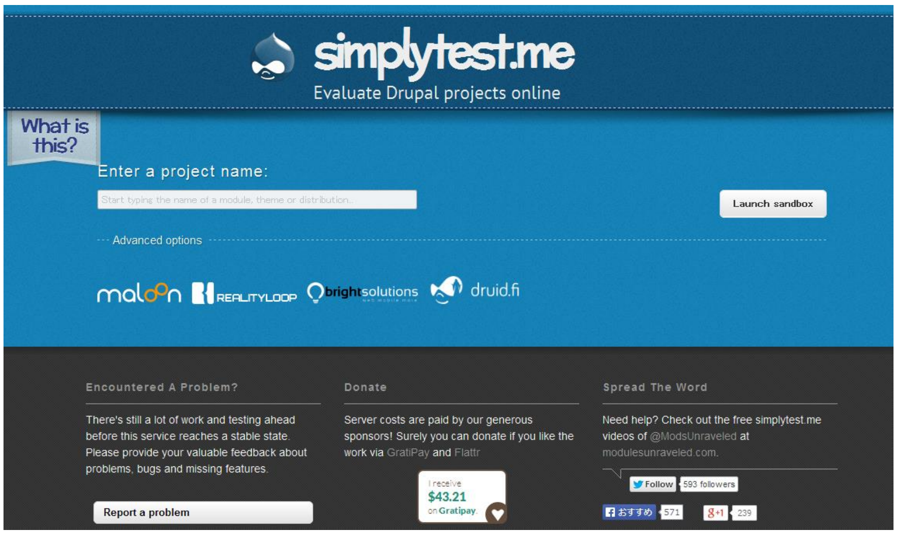
https://events.apifirstcms.org/

金曜日は DevOps/DDEV, Gatsby(React)+Drupal, Angular +Drupal のハンズオンです。

ぜひ、ご参加ください。

便利なツールやシステム

さまざま Drupal のコアや拡張モジュール、テーマ、ディストリビューションなどを無料で 24 時間、クラウドで体験できます。ただし、現在は新システム環境で稼働中　http://simplytest.me/



Encountered A Problem?

There's still a lot of work and testing ahead before this service reaches a stable state. Please provide your valuable feedback about problems, bugs and missing features.

Report a problem

Donate

Server costs are paid by our generous sponsors! Surely you can donate if you like the work via GratiPay and Flattr

I receive $43.21 on Gratipay.

Spread The Word

Need help? Check out the free simplytest.me videos of @ModsUnraveled at modulesunraveled.com.

Follow 593 followers

おすすめ 571　+1 239

デジタルマーケティング資料

- イベント特化 SNS(EventHub)
- 米アマゾン、服はプロのお勧め

次回の勉強会

11 月 26 日（水）午後 7 時から　中央区堀留町区民館 3 号室「Webform の基礎」

なんでも質問コーナー

Drupal や CMS、クラウドなど、ご質問をお受けいたします。

Dries さんのブログより

8 月はお休みです。

Dries さんのブログページ　https://dri.es/

今月のモジュール

8 月はお休みです。

コンテンツタイプとは

サイトで取り扱うさまざまコンテンツのデータをデータベースに登録する手順を学び、データ構造の考え方などを体験します。

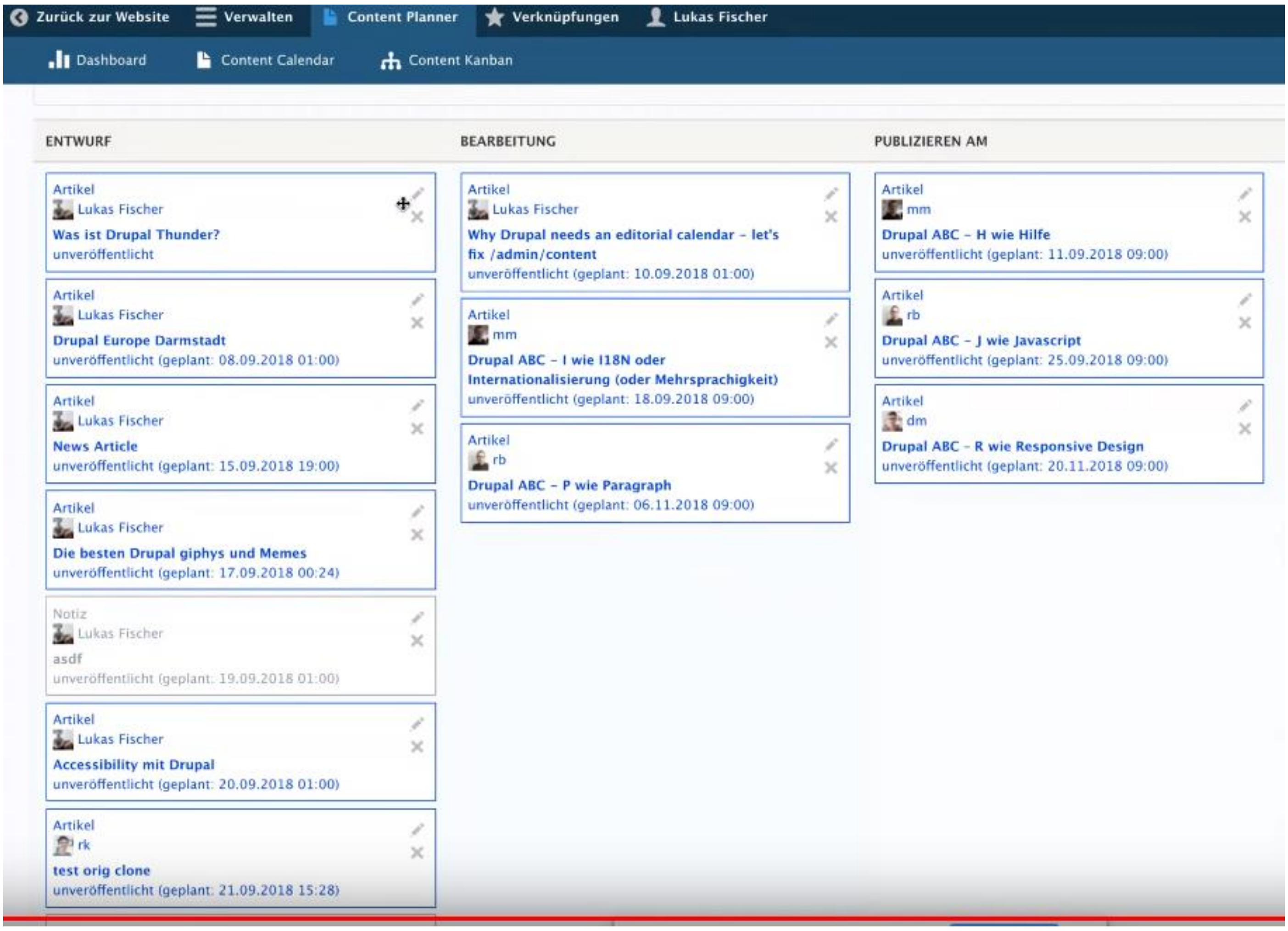
今年の 6 月の勉強会にてご紹介した、モジュールですが、最近、パッチ版がリリースされ、最新の Drupal 8 で稼働しますので、再度、紹介します。

Content Planner

Drupal のコンテンツ作成ワークフローをかんばんスタイルで使うことができるモジュールです。

モジュールのダウンロード

https://www.drupal.org/project/content_planner



ENTWURF

- Artikel / Lukas Fischer / **Was ist Drupal Thunder?** / unveröffentlicht
- Artikel / Lukas Fischer / **Drupal Europe Darmstadt** / unveröffentlicht (geplant: 08.09.2018 01:00)
- Artikel / Lukas Fischer / **News Article** / unveröffentlicht (geplant: 15.09.2018 19:00)
- Artikel / Lukas Fischer / **Die besten Drupal giphys und Memes** / unveröffentlicht (geplant: 17.09.2018 00:24)
- Notiz / Lukas Fischer / asdf / unveröffentlicht (geplant: 19.09.2018 01:00)
- Artikel / Lukas Fischer / **Accessibility mit Drupal** / unveröffentlicht (geplant: 20.09.2018 01:00)
- Artikel / rk / **test orig clone** / unveröffentlicht (geplant: 21.09.2018 15:28)

BEARBEITUNG

- Artikel / Lukas Fischer / **Why Drupal needs an editorial calendar – let's fix /admin/content** / unveröffentlicht (geplant: 10.09.2018 01:00)
- Artikel / mm / **Drupal ABC – I wie I18N oder Internationalisierung (oder Mehrsprachigkeit)** / unveröffentlicht (geplant: 18.09.2018 09:00)
- Artikel / rb / **Drupal ABC – P wie Paragraph** / unveröffentlicht (geplant: 06.11.2018 09:00)

PUBLIZIEREN AM

- Artikel / mm / **Drupal ABC – H wie Hilfe** / unveröffentlicht (geplant: 11.09.2018 09:00)
- Artikel / rb / **Drupal ABC – J wie Javascript** / unveröffentlicht (geplant: 25.09.2018 09:00)
- Artikel / dm / **Drupal ABC – R wie Responsive Design** / unveröffentlicht (geplant: 20.11.2018 09:00)

デモ動画　https://youtu.be/8TzJZR2j_34

7 月の勉強会でハンズオンしました。概要は以下のサイトにて

https://www.drupal-blog.ch/drupal-module/how-install-drupal-content-planner

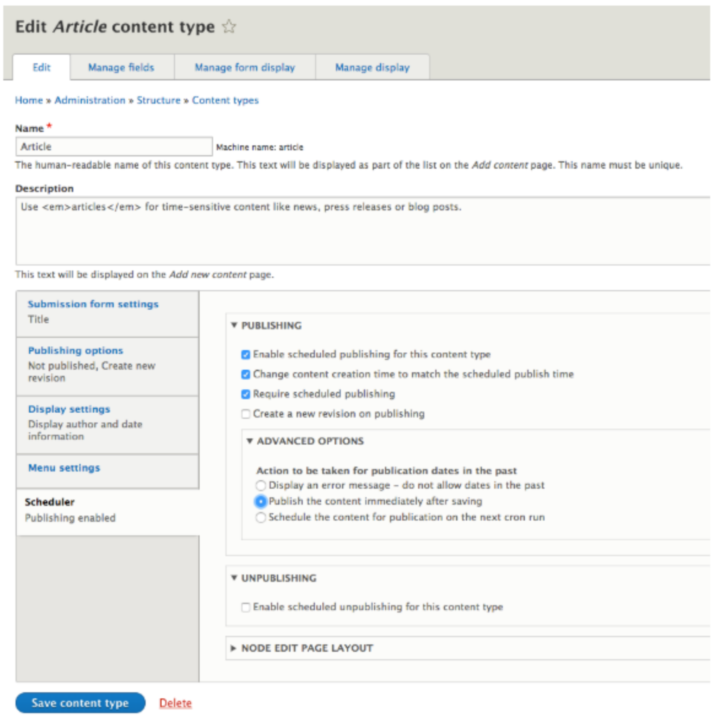
CONFIGURE YOUR CONTENT TYPES FOR CONTENT CALENDAR

Every Content Type you wish to be displayed on the Content Calendar has to be configured in conjunction with the Scheduler module.

For this go into the desired Content Type and set the following options:

- Option "Enable scheduler publishing for this content type"
- Option "Change content creation time to match the scheduled publish time"
- Option "Require scheduled publishing"

It is recommended to check the option "Publish the content immediately after saving" for dates in the past, but that is up to you.



Edit *Article* content type ☆

Edit | Manage fields | Manage form display | Manage display

Home » Administration » Structure » Content types

Name *
Article   Machine name: article
The human-readable name of this content type. This text will be displayed as part of the list on the *Add content* page. This name must be unique.

Description
Use <em>articles</em> for time-sensitive content like news, press releases or blog posts.
This text will be displayed on the *Add new content* page.

Submission form settings
Title

Publishing options
Not published, Create new revision

Display settings
Display author and date information

Menu settings

Scheduler
Publishing enabled

▼ PUBLISHING
- [✓] Enable scheduled publishing for this content type
- [✓] Change content creation time to match the scheduled publish time
- [✓] Require scheduled publishing
- [ ] Create a new revision on publishing

▼ ADVANCED OPTIONS
Action to be taken for publication dates in the past
- ○ Display an error message - do not allow dates in the past
- ● Publish the content immediately after saving
- ○ Schedule the content for publication on the next cron run

▼ UNPUBLISHING
- [ ] Enable scheduled unpublishing for this content type

▶ NODE EDIT PAGE LAYOUT

Save content type   Delete

Content Moderation Notifications

ワークフローのステータス変更通知メール送信

https://www.drupal.org/project/content_moderation_notifications

Token 対応になりました。

Nodeaccess

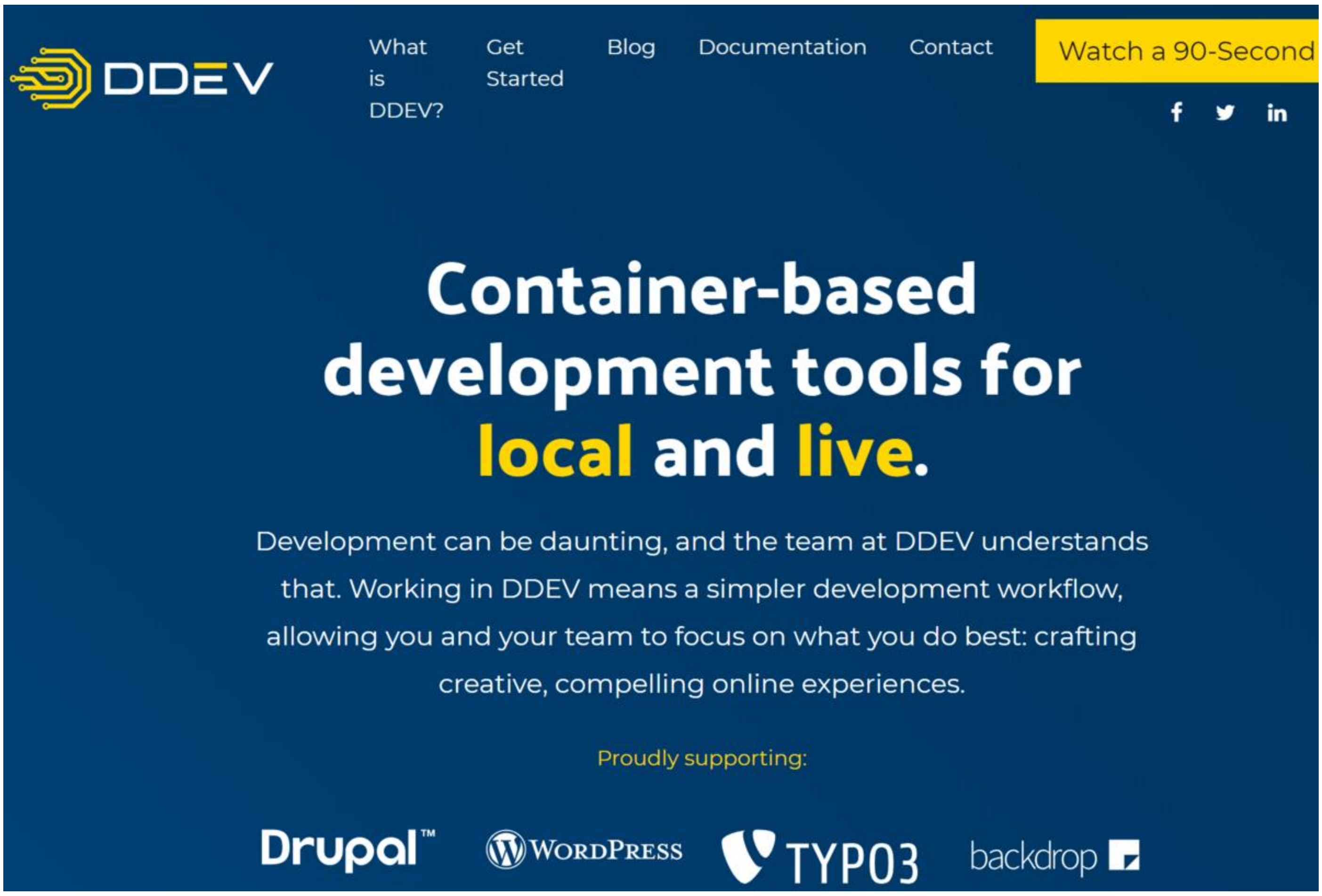
ノート（ページ）のアクセス制御
https://www.drupal.org/project/nodeaccess

Nodeaccess モジュールは、content accss モジュールよりも、現在は安定しているようです。
https://www.drupal.org/project/content_access

DevOps 環境のハンズオン

DDEV はオープンソースソフトウェアの DevOps 環境として、PHP 言語によるオープンソース CMS の WordPress、Drupal、TYPO3 などをターゲットにした、ローカルでコンテナベースのすぐ使える環境をサポートしています。XAMP を使っている方は、参考になると思います。その使い方を、Drupal 初心者向けに、「Local Web Development With DDEV Explained: Your Step-by-Step Guide to Local Web Development With DDEV」という英文教材を参考にしながらハンズオンを行います。

DDEV 開発元のサイト：https://www.drud.com/



Container-based development tools for local and live.

Development can be daunting, and the team at DDEV understands that. Working in DDEV means a simpler development workflow, allowing you and your team to focus on what you do best: crafting creative, compelling online experiences.

■ ハンズオンの概要

- DDEV の概要

マニュアル　https://ddev.readthedocs.io/en/stable/

- インストール

https://ddev.readthedocs.io/en/stable/#installation

以下の資料は、「Local Web Development with DDEV Explained」(OSTraining) から抜粋しました。この本はアマゾンにて購入可能です。　ここから

Linux のローカル（*PC* へ）にインストールする場合

This section will summarize the installation of DDEV-Local on Linux. The instructions may vary depending on the exact flavor and version of Linux that you are using.

In order to install DDEV-Local on Linux, there are several prerequisites:

First, you will need **Docker**. I recommend installing Docker using the Install using the repository method for your particular flavor of Linux. These links have installation instructions for three popular flavors of Linux:

- Debian: https://docs.docker.com/install/linux/docker-ce/debian/
- Fedora: https://docs.docker.com/install/linux/docker-ce/fedora/
- Ubuntu: https://docs.docker.com/install/linux/docker-ce/ubuntu/

Pay close attention to the instructions for your version of Linux, as the instructions may have subtle but important differences. For example, at the time of this writing, to install Docker on Ubuntu 18.04, you need to add the edge repository in addition to the stable repository in the following command.

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ ubuntu $(lsb_release -cs) stable edge"

Second, you need **Docker Compose,** which is a tool for managing sets of containers. The official documentation has instructions for installing Docker Compose on Linux at https://docs.docker.com/compose/install/.

Third, I recommend you complete these post-installation steps for Docker and Docker

Compose: https://docs.docker.com/install/linux/linux-postinstall/.

Finally, you should start running Docker. To run Docker from the command-line, use the following command:

```
sudo service docker start
```

The recommended way to install DDEV-Local on Linux is via Linuxbrew (http://linuxbrew.sh/), the Homebrew package manager for Linux.With Linuxbrew installed, DDEV-Local can be installed with

```
brew tap drud/ddev && brew install ddev
```

インストールした *DDEV* 環境の確認

Whether you’re on Mac, Windows or Linux, once installation is complete, it’s time to confirm that DDEV-Local is ready to use.

First, let’s verify that DDEV-Local is accessible using this command:

```
which ddev
```

This command should return the path to the ddev executable. On Windows, depending on your console emulator, you may have to use this command:

```
where ddev
```

Next, let’s confirm the version of DDEV-Local that was just installed, using this command:

ddev version

Finally, we can review a list of the available DDEV-Local commands with this command:

ddev help

If these three commands each return reasonable results, you have successfully installed DDEV-Local.

DDEV ローカルコマンド

Here’s that list of DDEV-Local commands again. Commands in bold were covered in this chapter:

1. auth
2. auth-pantheon
3. composer
4. config
5. describe
6. exec
7. export-db
8. **help**
9. hostname
10. import-db
11. import-files
12. list

13. logs
14. pause
15. pull
16. remove
17. restart
18. restore-snapshot
19. sequelpro
20. share
21. snapshot
22. ssh
23. start
24. stop
25. **version**

新規に *Drupal 8* をインストールする場合

In the previous chapter, I showed you how to install DDEV-Local.

In this chapter, I'll show you how to install a blank, new Drupal 8 site. You can use this to start a fresh project, or you can simply use it as a development playground.

The current best practice in the Drupal community for creating a new Drupal 8 site is to use the Drupal Composer/Drupal Project template: https://github.com/drupal-composer/drupal-project/.

I'm going to show you how to create a new Drupal 8 project with the following assumptions:

- Project dependencies will not be committed to the local Git repository.
- Composer is **not** installed on the host operating system. We will see how to utilize the version of Composer that is installed in the DDEV-Local web container. This will require us to create the DDEV-Local project containers prior to creating the codebase, as we will need Composer to create the codebase.

ステップ１　コードベースの作成

In this first step, we will create the DDEV-Local containers for this project and use Composer to create the Drupal 8 codebase.

From the command line, (create if necessary and then) navigate to your local Sites directory and create a new, empty, project directory using the command below. In this command, myproject is a short, machine-name type name for your project. This is often all lower-case and should not include any non-alphanumeric characters.

mkdir myproject

From the command line, navigate into the new project directory using this command.

cd myproject

From the command line, run the command below. This is an alternate way of running the "ddev config" command that passes in the information necessary to configure the default DDEV-Local containers. This will result in a ".ddev" directory in your project ("myproject") directory. This directory will contain the DDEV-Local configuration information for this project.

ddev config --project-name myproject --docroot . --project-type php

From the command line, run the following command to create the containers for this project:

ddev start

The next step is to use the “ddev composer create” command to install the Drupal Composer/Drupal Project template (https://github.com/drupal-composer/drupal-project/) to create the Drupal 8 codebase. This command uses most of the default parameters listed in the Drupal Composer/Drupal Project template’s installation instructions, but with a few minor changes.

ddev composer create drupal-composer/drupal-project:8.x-dev --stability dev

--no-interaction

Note that instead of using “composer create-project”, we’re using “ddev composer create”. Also note that we don’t need to pass in the name of the directory to install the project in.

The result of this command is the Drupal 8 codebase installed in your project directory. By using the “ddev composer” command, you are using the version of Composer that comes with the default DDEV-Local web container.

We are unable to use the “composer create-project” command to create the codebase directly in the myproject directory because Composer requires the directory to be empty, and the .ddev directory is already present. The “ddev composer create” command sidesteps this issue.

The “ddev composer create” command does several things:

- It downloads the project from the source repository (GitHub, in this case).
- It automatically runs “composer install” to download all dependencies and run any pre- and post-install Composer scripts, as specified in the composer.json file provided by the template.

The “ddev composer” command can be used to run other Composer commands during the lifecycle of a project, including update, require, and remove.

At this point, the codebase is just about ready for use.

ステップ２　コンテナと *Drupal* 設定ファイルの構成

With a Drupal 8 codebase now present, we can rerun the “ddev config” command as follows:

ddev config --project-name myproject --docroot web --project-type drupal8

This will accomplish several things:

1. An updated .ddev/config.yaml file with “docroot: web” and “name: myproject”.
2. The creation of a web/sites/default/settings.ddev.php file with the proper database connection information.
3. The modification of web/sites/default/settings.php with a conditional include of the settings.ddev.php file.

4. The creation of a web/sites/default/.gitignore file that ignores the settings.ddev.php file.

It is a good practice to utilize the core Drupal 8 web/sites/example.settings.local.php file. To do so, first run the following command to copy and rename the file for use:

cp web/sites/example.settings.local.php web/sites/default/   settings.local.php

Next, edit the settings.php file and uncomment the following lines near the bottom of the file:

if (file_exists($app_root . '/' . $site_path . '/settings.local.php')) {

include $app_root . '/' . $site_path . '/settings.local.php';

}

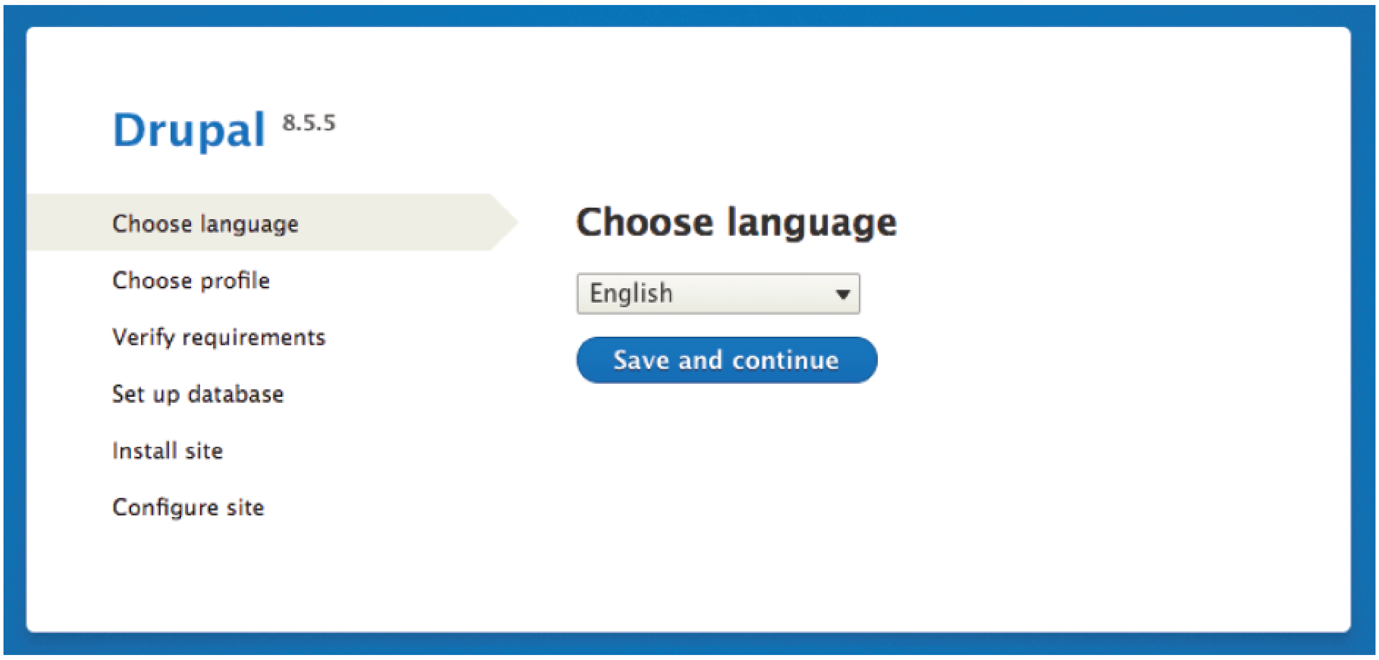
Because we made changes to the DDEV-Local project configuration above, we can now run the following command to restart the project’s containers:

ddev restart

Once restarted, use the following command to be reminded of the project’s local URLs:

ddev describe

Then, navigate to one of the project’s URLs where you should be greeted with the Drupal 8 installer page! As you go through the installer, you’ll notice that you will skip the “Set up database” step, as DDEV-Local’s creation of the settings.ddev.php file already took care of this for you!



他の方法で新しいサイトをインストールする場合：

This chapter assumes Composer isn't installed on the host operating system. This requires us to use the version of Composer that is installed in the default DDEV-Local web container. However, if Composer is installed on your operating system, then Step #1. Creating the Codebase section becomes much simpler.

Navigate into your Sites directory and run the following command to create the codebase:

composer create-project drupal-composer/drupal-project:8.x-dev myproject --stability dev --no-interaction

In this case, replace myproject with a short, machine-name for your project. This will be similar to naming the project directory earlier in this chapter.

That's it! At this point, your codebase is ready for Step #2. Configure the Containers and the Drupal Settings Files.

既存 Drupal サイトを DDEV 環境へ移行させる

In a previous chapter, I showed you how to get a new Drupal 8 site up-and-running in DDEV-Local.

If the project already exists, then you'll want to clone the site from your project's remote repository. This allows you to get an up-to-date copy of the codebase, plus a complete history of the code changes.

You will also need to get the site's database and content files in order to get the site up-and-running.

While DDEV-Local works with Drupal , TYPO3, WordPress, and other applications, in this chapter, we're going to clone a Drupal 8 site.

I've made a codebase, database, and content files directory available for you to practice

with: https://github.com/ultimike/ddevdemo. This codebase is configured so the Composer dependencies are committed to the repository. This means you won't need to run any Composer commands.

The database export is located in the /db_export/ directory of the repository. The files/ directory export is located in the /files_export/ directory of the repository.

ステップ 1　プロジェクトを複製

From your local command line, navigate to the directory where you want to clone the project to. Typically, this is your local sites/ directory. Now enter this command:

```
git clone https://github.com/ultimike/ddevdemo.git
```

This will clone the entire project’s history from GitHub to your computer, in a new directory named ***ddevdemo***. Once cloned, Git will create a working directory of the branch named master. The general structure of the git clone command is:

git clone protocol://url_of_repository name_of_local_directory_to_create

Note that when you omit the name_of_local_directory_to_create parameter, the name of the directory is set to the name of the repository, in this case ***ddevdemo***.

After the git clone command is done, your local directory structure should look like this (assuming you cloned into your sites/ directory):

- sites
- ddevdemo
- web/
- (several other files and directories)

The ***ddevdemo*** site utilizes a ***nested docroot***, meaning that Drupal is installed in the web/ directory (the ***docroot***) while the project root is in the ddevdemo/ directory. The main advantage of utilizing a nested docroot is that anything not in the web/ directory is normally not accessible via a web browser, as it is one-level “up” from the docroot. This is assuming a properly configured web server and that the site’s domain name is pointing to the docroot.

This setup provides developers with a location to place files that are related to the project that should not be accessible via a URL. Some examples are configuration files, local development configuration, and automated tests.

The ***ddevdemo*** codebase is a Drupal 8 project that utilizes the ***Composer template for Drupal projects***: https://github.com/drupal-composer/drupal-project. This is a relatively standard way of organizing a Drupal 8, Composer-managed site. This Composer template provides the following features:

• Nested docroot for the Drupal site (/web).

• “Above the docroot” directories for site configuration, drush, and other scripts.

• A fully-Composer managed codebase (including Drupal core being a dependency of the project).

• Dependencies are not normally committed to the project repository. In this example, a modification was made to the project’s .gitignore file so that dependencies are committed to the repository.

Once the site is successfully cloned, you can navigate into the project directory using this command:

```
cd ddevdemo
```

ステップ２　*DDEV* ローカルコンテナ

The second step is to use DDEV-Local to get the site up-and-running on your local machine. Keep in mind that DDEV-Local provides a set of Docker containers per project, so the containers will be configured specifically for this project.

From the command line, run the following command:

ddev config

When you run this command without passing any parameters or flags, it will ask several questions so that it can provide a set of default Docker containers for the project. DDEV-Local will guess the correct answers based on the contents of the project directory; default values will be displayed in parentheses at the end of each question.

You can respond to DDEV-Local's questions with the following replies:

• **Project name: ddevdemo**. Typically, the project name is set as the same as the project directory name.

• **Docroot location: web**. As we're using the Drupal Composer/Drupal Project template, this is the default location for the docroot. DDEV-Local will guess the proper value and provide it as an option in the prompt.

• **Project type: drupal8**. DDEV-Local will do its best to figure this out based on the codebase.

The entire output of the "ddev config" command should look similar to the following:

~/Sites/ddevdemo [master]$ ddev config

This will create a new ddev project config in the current directory (/Users/

michael/Sites/ddevdemo). Once completed, your configuration will be written to:

/Users/michael/Sites/ddevdemo/.ddev/config.yaml

```
Project name (ddevdemo):

The docroot is the directory from which your site is served. This is a

relative path from your project root (/Users/michael/Sites/ddevdemo)

You may leave this value blank if your site files are in the project root

Docroot Location (web):

Found a drupal8 codebase at /Users/michael/Sites/ddevdemo/web.

Project Type [wordpress, typo3, backdrop, php, drupal6, drupal7, drupal8]

(drupal8):

Ensuring write permissions for ddevdemo

Existing settings.php file does not include settings.ddev.php, modifying to

include

ddev settings

Configuration complete. You may now run 'ddev start'.
```

From the command line, run the following command to download, create and run the containers for this project:

```
ddev start
```

You may have to enter your computer’s password in order for DDEV-Local to add an entry to your hosts file for this project. The entire output of this command should look similar to the following:

```
 ~/Sites/ddevdemo [master*]$ ddev start

Starting ddevdemo...

ddev needs to add an entry to your hostfile.

It will require administrative privileges via the sudo command, so you may be required

to enter your password for sudo. ddev is about to issue the command:


    sudo /usr/local/bin/ddev hostname ddevdemo.ddev.site 127.0.0.1


Please enter your password if prompted.



Running Command Command=sudo /usr/local/bin/ddev hostname ddevdemo.ddev.site

127.0.0.1

Creating ddev-ddevdemo-db ... done

Creating ddev-ddevdemo-dba ... done

Creating ddev-ddevdemo-web ... done

Network ddev_default is external, skipping

Creating ddev-router ... done

Ensuring write permissions for ddevdemo

Successfully started ddevdemo
```

Project can be reached at http://ddevdemo.ddev.site, https://ddevdemo.ddev.site, http://127.0.0.1:32779

At this point, only the project's codebase is ready for you to use. Neither the database nor the files/ directory is ready yet. Visiting one of the URLs provided by DDEV-Local will redirect you to Drupal 8's installation page. But, since this project includes a database, the next step is not to install, but rather to import and connect to the site's database.

ステップ３　データベースのインポートと接続

Now that we have the complete codebase on our local environment with DDEV-Local up-andrunning, the next step is to set up our local settings files and import the database.

I recommend you set up three settings files to manage general site settings, local settings, and DDEVLocal- specific settings:

1. **ddevdemo/web/sites/default/settings.php**. This is the standard Drupal settings file and is typically part of the repository.
2. **ddevdemo/web/sites/default/settings.ddev.php**. This contains the settings generated by the ddev config command and is not typically part of the repository. This file includes the local database connection information. Git ignores this file via the /ddevdemo/web/sites/

   default/.gitignore file generated during the "ddev config" command.
3. **ddevdemo/web/sites/default/settings.local.php**. This is a copy of the ddevdemo/web/sites/ example.settings.local.php file, together with any additional

local settings provided by the developer. This file is typically not part of the repository. Git ignores this file via this project’s .gitignore file.

With a Drupal 8 site, the settings.ddev.php file is automatically created during the ddev config process. You can modify the settings.php file with the addition of a conditional include of the settings.ddev.php file.

To set up the settings.local.php file, run this command from the project root:

```
cp web/sites/example.settings.local.php web/sites/default/ settings.local.php
```

You can also use your operating system’s GUI to copy and rename web/sites/

example.settings.local.php to web/sites/default/settings.local.php. The settings.php file included with the codebase is already configured to conditionally include the

settings.local.php file with the following code:

```
if (file_exists($app_root . '/' . $site_path . '/settings.local.php')) {

include $app_root . '/' . $site_path . '/settings.local.php'; }
```

Now import the database provided with this project (in the ddevdemo/db_backup/ directory). You can accomplish this with the following command:

```
ddev import-db --src=db_backup/ddevdemo_db.sql
```

You will likely see a message confirming that the settings.ddev.php file is present and is being conditionally included in the settings.php file – this is normal.

A common pitfall with this command occurs when a database dump file has an incorrect extension. For example, if a database dump file is provided that is compressed, but has the .sql extension, this command will fail. The Linux “file” command is helpful in determining the actual file type, regardless of the extension. For example:

```
$ file test.sql.gz
```

```
test.sql.gz: gzip compressed data, max compression, from Unix
```

At this point, if you visit one of the local site’s URLs, your local copy of the DDEV Demo site should now load. However, you won’t see any images yet.

ステップ４　コンテンツファイルの取り込み

Now that we have the codebase and the database on our local environment with DDEV-Local up-and-running, it’s time to import our content files. The DDEV Demo repository includes an archive of the necessary content files, and you can easily import them into the site with the “ddev import-files”

command:

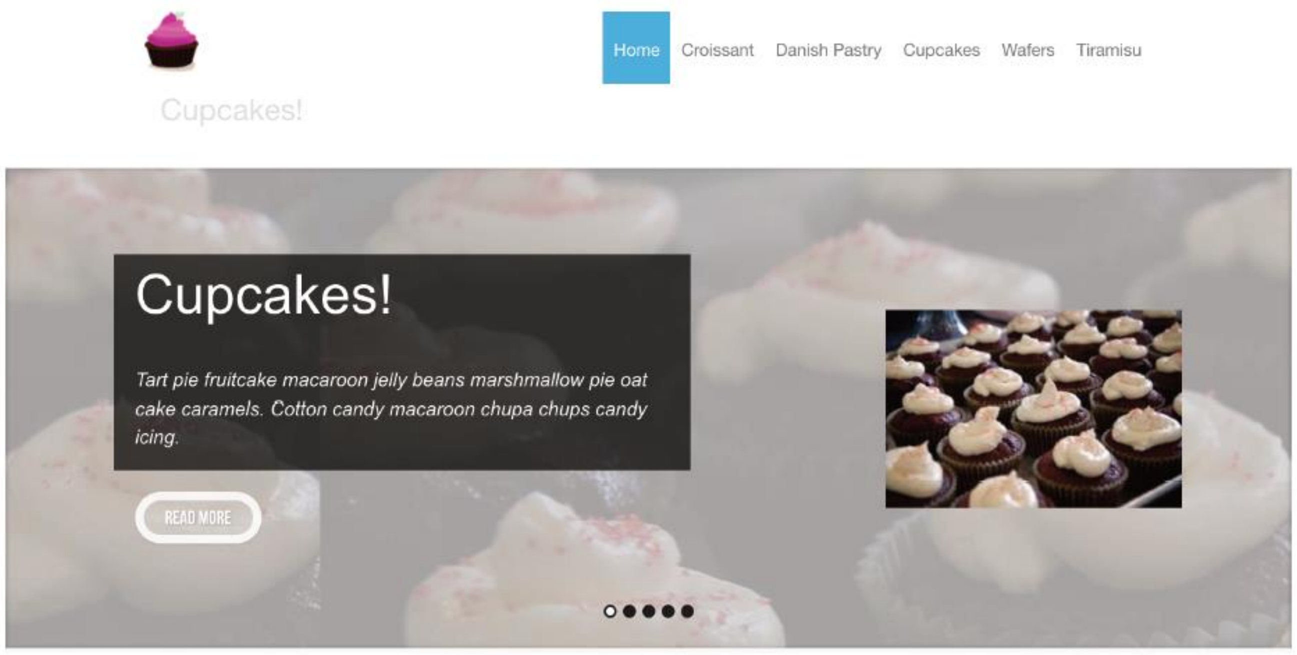
```
ddev import-files --src=files_backup/ddevdemo_files.zip
```

This command will simply extract the archive and move it to the proper location in the codebase,

which is /ddevdemo/web/sites/default/files/.

ステップ５　結果を確認

Navigate to the site in your web browser at http://ddevdemo.ddev.site and confirm that your site is up-and-running with images!


Cupcakes!
Home
Croissant
Danish Pastry
Cupcakes
Wafers
Tiramisu
Cupcakes!
Tart pie fruitcake macaroon jelly beans marshmallow pie oat cake caramels. Cotton candy macaroon chupa chups candy icing.
READ MORE

You may need to rebuild caches for the site to properly display. You can do this by logging into the site and then using the “Flush all caches” command from the admin menu.

Here are the admin credentials for the site:

- Username: admin
- Password: admin

ディスカッション

Drupal、WordPress、オープンソースコミュニティ、CMS などの質疑応答

クレジット、謝辞、ライセンス

クレジット

このマニュアル作者は、CMSLABO 有限責任事業組合の程田和義です。

お問合せ　電子メール　hodota@cmslabo.co.jp　電話 044-220-1588

謝辞

本マニュアル作成は、主に以下のサイトを参考にしました。心より感謝いたします。

出典：　　Drupal.org　simplytest.me

ライセンス