

Level Up your DDEVry

A collection of time saving tips

Bernardo Martinez 13/7/2024

<https://www.drupal.org/u/bernardm28>

<https://www.linkedin.com/in/bernardm28>

Goals

What you might get from this talk

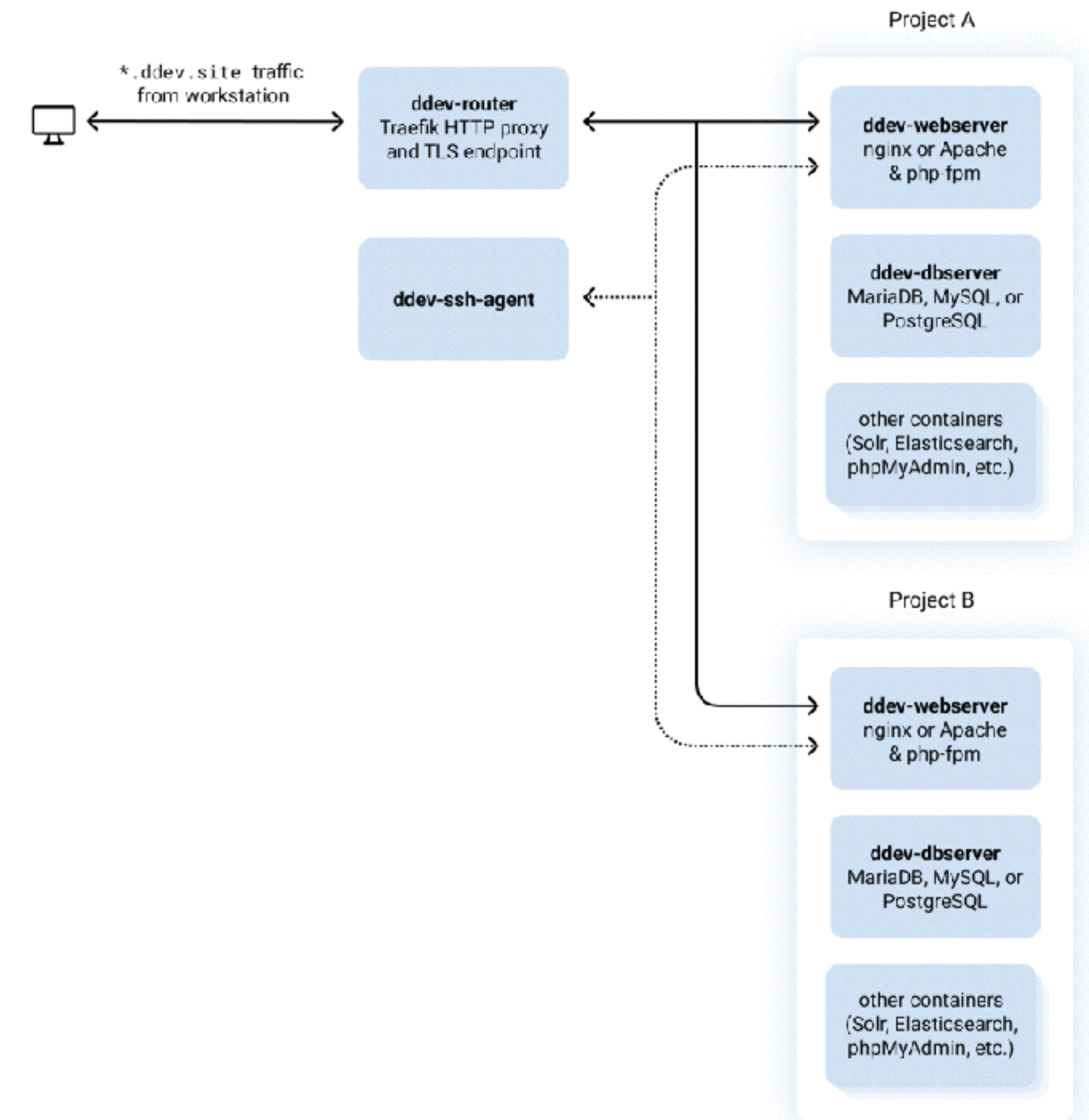
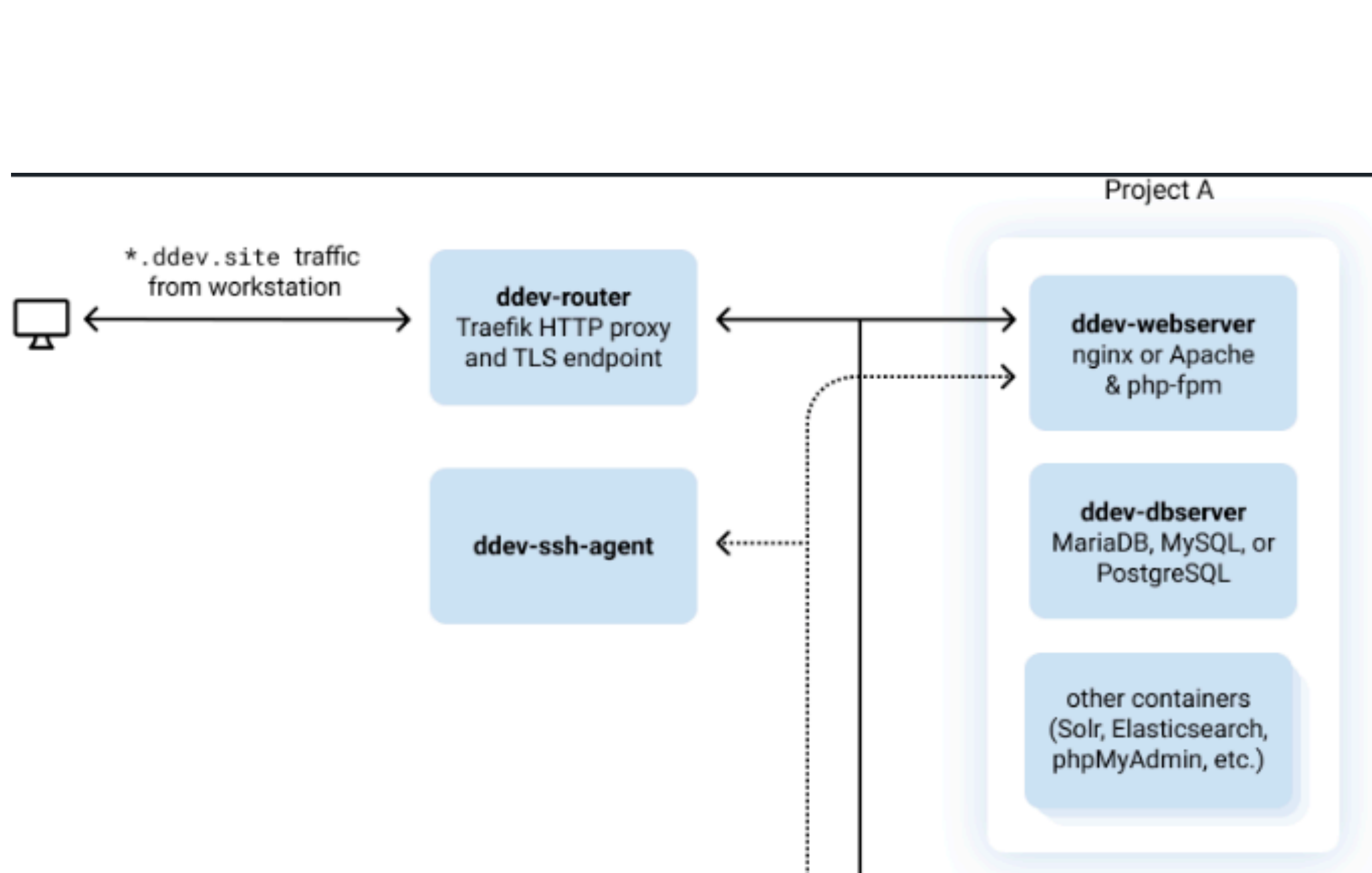
- A quick overview of DDEV commands and its flags.
- A overview of DDEV hosting providers.
- An introduction to DDEV community and where to ask for support.

Table of content

Topics covered

architecture plus project structure	Advanced DDEV custom commands
DDEV provided Commands	Environment variables (provided and user generated)
Database Management	DDEV router
DDEV hooks	DDEV hosting providers
DDEV Live hosting	Resources

DDEV architecture



<https://ddev.readthedocs.io/en/latest/users/usage/architecture/>

<https://www.dropbox.com/scl/fi/rh7o01yc4uriittvth9wp/>

2023-09-12_DDEV_architecture.mp4?rlkey=f44r4kyhq7flfwbcody3omcs0&e=1&dl=0

Project structure

ddev config

```
└─ .ddev
  ├── .global_commands
  ├── addon-metadata
  ├── commands
  ├── db-build
  ├── homeadditions
  ├── providers
  ├── web-build
  ├── web-entrypoint.d
  ├── xhprof
  ├── .gitignore
  └─ ! config.yaml
```

ddev config --help
ddev config --auto --update

```
.ddev > ! config.yaml
1  type: drupal10
2  docroot: web
3  php_version: "8.1"
4  webserver_type: nginx-fpm
5  xdebug_enabled: false
6  additional_hostnames: []
7  additional_fqdns: []
8  database:
9      type: mariadb
10     version: "10.4"
11     use_dns_when_possible: true
12     composer_version: "2"
13     web_environment: []
14
15 # Key features of DDEV's config.yaml:
```

DDEV includes 263 lines of comments explaining each inside config.yml. They can be deleted but they explain the boilerplate code.

.ddev/config.local.yaml
ddev mutagen reset

DDEV perks

Items you don't have to worry about.

- NPM built-in
- XDebug built-in
- Drush alias built-in
- DB Snapshots
- Bash conditional logic
- Offline support

Add Node Service

Assuming you're starting with a "Lando-ized" app, open the `.lando` directory. In this example we'll assume you're using a very basic **LAMP**

To install our frontend tooling we need to be able to run Node. Fortunately we can add a basic Node service to our app:

```
1 name: myapp
2 recipe: lamp
3
4 services:
5   node:
6     type: node
7     build:
8       - npm install
```

Note the `build` section. This bash command will automatically run when we install any Node packages specified in our package.json file.

```
services:
  appserver:
    environment:
      DRUSH_OPTIONS_URI: "https://mysite.lndo.site/"
```

Enable Xdebug by adding the `xdebug: true` line to your `.lando.yml`.

When using a recipe, add it under the `config` key:

```
1 name: mywebsite
2 recipe: drupal10
3 config:
4   xdebug: true
```

Otherwise, override your php service, usually named `appserver`:

```
1 name: mywebsite
2 services:
3   appserver:
4     webroot: web
5     xdebug: true
```

Rebuild your environment.

```
1 lando rebuild -y
```

Finally, create a custom `launch.json` file in your workspace in order to map paths so that XDebug

Making Tooling Available on the CLI

Almost there! All our services are installed, but how do we run a command on the fly, say starting a watch task or running `lando npm install hot-new-thing` to start experimenting with a new package? We could SSH into our node container, but that's SO 2016. Instead, we'll expose our new tooling via the CLI by adding this `tooling` section to our `.lando.yml` file:

```
1 tooling:
2   npr:
3     service: node
4   node:
5     service: node
6   gulp:
7     service: node
8   yarn:
9     service: node
```

After restarting your app, you should be able to run `lando node`, `lando gulp` or `lando npm` and have the corresponding commands run. This is particularly useful if you want to kickoff a watch task you might have configured, say `lando gulp watch`.

Highlighted features

2024

ddev start --skip-config-hooks ([v1.23.3](#))

ddev delete images / ddev delete images --all ([v1.23.3](#))

ddev launch <https://ddev.com> / ddev launch :3000 ([v1.23.0](#))

ddev_version_constraint: '>=v1.23.0' ([v1.23.0](#))

ddev xdebug toggle ([v1.23.1](#))

ddev xhprof toggle ([v1.22.6](#))

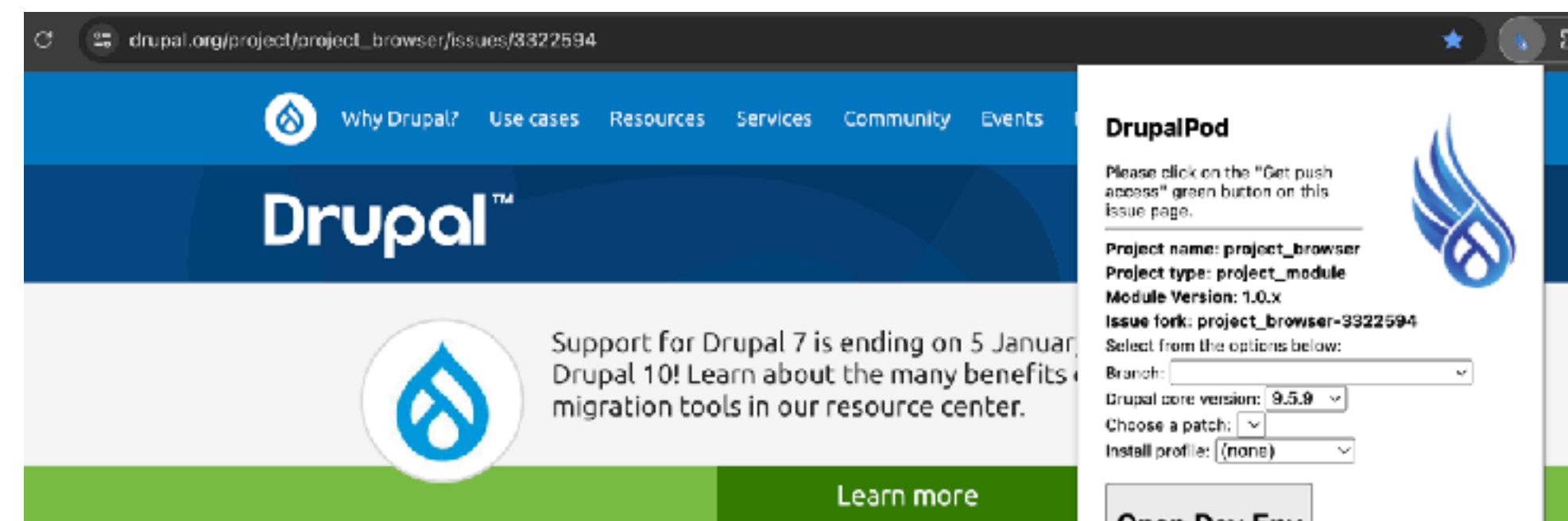
corepack_enable: true ([v1.23.0](#))

DrupalPod

Easier Drupal contribution setup

👉 Start here

1. Download the DrupalPod browser extension
[Chrome](#) or [Firefox](#).
2. Go to any issue page on Drupal.org
(core, module, or theme).
3. Click on the DrupalPod extension.
4. (Optional) Choose a patch / issue fork / branch.



<https://dgo.to/3322594>

<https://github.com/shaal/DrupalPod>

Let's see a demo

Q Search page

Authorize Gitpod to use your account?

An application called [Gitpod](#) is requesting access to your GitLab account. This application was created by [Gitpod](#). Please note that this application is not provided by GitLab and you should verify its authenticity before allowing access.

This application will be able to:

- **Access the authenticated user's API**
Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.
- **Read the authenticated user's personal information**
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.

New Workspace

Create a new workspace in the `bmartinez287` organization.

You are not authenticated.
[Authorize with git.drupalcode.org](#)

`<>` DP_PROJECT_NAME=project_browser,DP_ISSUE_FOR... ▾

`>_` VS Code · 1.90.2
Editor · Browser ▾

Standard
Class · Up to 4 cores, 8GB RAM, 30GB storage ▾

git.drupalcode.org/~user_settings/applications

User Settings / Applications

GitLab now enforces expiry dates on tokens that originally had no set expiration date. Those tokens were given an expiration date of one year later. Please review your personal access tokens, project access tokens and group access tokens to ensure you are aware of upcoming expirations. Administrators of GitLab can find more information on how to identify and mitigate interruption, please take a look at our documentation.

Q Search page

Applications

Manage applications that can use GitLab as an OAuth provider, and applications that you've authorized to use your account.

Your applications 0

You don't have any applications.

Authorized applications 1

Name	Authorized At	Scope	
Gitpod	2024-06-26 14:19:35 UTC	api read:user	<input type="button" value="Revoke"/>

Environment variables (provided)

Comparison of environment variables

```
## Concatenates path to theme directory for your project.
DDEV_THEMEROOT=/var/www/html/${DDEV_DOCROOT}/${THEME}

## Presupposes that gulp is installed in ${DDEV_THEMEROOT}
## You may need to "npm install" theme dependencies before
if [ -d ${DDEV_THEMEROOT} ]
then
  if [ -f ${DDEV_THEMEROOT}/node_modules/gulp/bin/gulp.js ]
  then
    pushd ${DDEV_THEMEROOT}
    nvm install 18.19.1
    nvm use 18.19.1
    nvm alias default 18.19.1
    ./node_modules/gulp/bin/gulp.js $@
    nvm install 14.19.1
    nvm use 14.19.1
    nvm alias default 14.19.1
  popd
```

Useful variables for container scripts are:

- `DDEV_DOCROOT` : Relative path from approot to docroot
- `DDEV_FILES_DIR` : *Deprecated*, first directory of user-uploaded files
- `DDEV_FILES_DIRS` : Comma-separated list of directories of user-uploaded files
- `DDEV_HOSTNAME` : Comma-separated list of FQDN hostnames
- `DDEV_MUTAGEN_ENABLED` : `true` if Mutagen is enabled
- `DDEV_PHP_VERSION` : Current PHP version
- `DDEV_PRIMARY_URL` : Primary URL for the project
- `DDEV_PROJECT` : Project name, like `d8composer`
- `DDEV_PROJECT_TYPE` : `drupal8`, `typo3`, `backdrop`, `wordpress`, etc.
- `DDEV_ROUTER_HTTP_PORT` : Router port for HTTP
- `DDEV_ROUTER_HTTPS_PORT` : Router port for HTTPS
- `DDEV_SITENAME` : Project name, like `d8composer`
- `DDEV_TLD` : Top-level project domain, like `ddev.site`
- `DDEV_WEBSERVER_TYPE` : `nginx-fpm`, `apache-fpm`, or `nginx-gunicorn`
- `IS_DDEV_PROJECT` : If `true`, PHP is running under DDEV

<https://docs.lando.dev/core/v3/env.html#default-environment-variables>

<https://ddev.readthedocs.io/en/latest/users/extend/custom-commands/#environment-variables-provided>


```

.ddev > commands > web > $ gulp
1  #!/bin/bash
2
3  ## Description: Run gulp in theme directory.
4  ## Usage: gulp build|watch
5  ## Example: "ddev gulp", "ddev gulp build", "ddev gulp watch"
6
7  ## Edit the following line to set the relative path from DDEV_DOCROOT
8  ## to your custom theme root (this example works for Drupal projects.)
9  THEME=themes/custom/customthemename
10
11 ## Concatenates path to theme directory for your project.
12 DDEV_THEMEROOT=/var/www/html/${DDEV_DOCROOT}/${THEME}
13
14 ## Presupposes that gulp is installed in ${DDEV_THEMEROOT}/node_modules
15 ## You may need to "npm install" theme dependencies before running this.
16 if [ -d ${DDEV_THEMEROOT} ]
17 then
18     if [ -f ${DDEV_THEMEROOT}/node_modules/gulp/bin/gulp.js ]
19     then
20         pushd ${DDEV_THEMEROOT}
21         nvm install 18.19.1
22         nvm use 18.19.1
23         nvm alias default 18.19.1
24         ./node_modules/gulp/bin/gulp.js $@
25         nvm install 14.19.1
26         nvm use 14.19.1
27         nvm alias default 14.19.1
28         popd
29     else
30         echo "Error: No gulp.js found at ./node_modules/gulp/bin/gulp.js."
31         echo '(You may need to "npm install" theme dependencies first.)'
32     fi
33 else
34     echo "Error: Directory ${DDEV_THEMEROOT} does not exist."
35     echo '(Edit THEME variable in ./ddev/commands/web/gulp to fix this.)'
36 fi
37 exit
38
39 ## The same approach should work for other front-end build tools
40 ## (e.g., grunt, brunch, less, sass, compass, etc.)
41 ## which expect to be run with the theme directory as working directory.
42 ## It should also work when building theme assets for WordPress, etc.,
43 ## just by changing the DDEV_THEMEROOT to an appropriate directory.

```

```

.ddev > commands > host > $ sync
You, 2 weeks ago | 1 author (You)
1  #!/bin/bash
2  ## Description: Run sync
3  ## Usage: sync [flags] [args]
4  ## Example: "ddev sync [environment] ddev sync -d-q(dev env, db only) or ddev sync -s-q(stage env, db only)"
5  ## ExecRaw: true
6  ## HostWorkingDir: true
7  ## Flags: [{"Name":"development","Shorthand":"d","Usage":"ddev sync -d syncs the development environment"}, {"Name":"stage","Shorthand":"s","Usage":"ddev sync -s syncs the stage environment"}]
8
9  if [ "${DDEV_PROJECT_STATUS}" != "running" ] && [ -z "$no_recursion" ]; then
10     echo "Project ${DDEV_PROJECT} is not running, starting it"
11     ddev start
12     start_exit_code=$?
13     if [ $start_exit_code -ne 0 ]; then
14         exit $start_exit_code
15     fi
16     # run this script again, as the environment is updated after "ddev start"
17     no_recursion=true ddev "$(basename "$0")" "$@"
18     exit $?
19 fi
20
21 while ;; do
22     case ${1:-} in
23         -d-q|--development--database|--dev--database)
24             ddev pull pantheon --environment=project=dccha24.dev --skip-files -y
25             ;;
26         -d-f|--development--files|--dev--files)
27             ddev pull pantheon --environment=project=dccha24.dev --skip-db -y
28             ;;
29         -d|--development|--dev)
30             ddev pull pantheon --environment=project=dccha24.dev
31             ;;
32         -s-q|--stage--database|--stg--database)
33             ddev pull pantheon --environment=project=dccha24.stage --skip-files -y
34             ;;
35         -s-f|--stage--files|--stg--files)
36             ddev pull pantheon --environment=project=dccha24.stage --skip-db -y
37             ;;

```

Updating an old DDEV file config file

```
.ddev > ! config.yaml
You, 12 months ago | 2 authors (shaal and one other)
1  name: fldc22
2  type: drupal9
3  docroot: "web"
4  php_version: "8.1"
5  webserver_type: nginx-fpm
6  router_http_port: "80"
7  router_https_port: "443"
8  xdebug_enabled: false
9  additional_hostnames: []
10 additional_fqdns: []
11 mariadb_version: "10.3"
12 mysql_version: ""
13 nfs_mount_enabled: false
14 mutagen_enabled: false
15 use_dns_when_possible: true
16 composer_version: "2"
17 web_environment: []
18
19 # Key features of ddev's config.yaml:
20
```

```
.ddev > ! config.yaml
1  name: provdemo
2  type: drupal
3  docroot: web
4  php_version: "8.3"
5  webserver_type: nginx-fpm
6  xdebug_enabled: false
7  additional_hostnames: []
8  additional_fqdns: []
9  database:
10 |   type: mariadb
11 |   version: "10.11"
12 use_dns_when_possible: true
13 composer_version: "2"
14 web_environment: []
15 corepack_enable: false
16
```

`ddev_version_constraint: v1.23.0-alpha1`

`ddev config --auto --update`

DDEV 2019

```
.ddev > ! config.yaml
You, 12 months ago | 2 authors (shaal and one other)
1 name: fldc22
2 type: drupal9
3 docroot: "web"
4 php_version: "8.1"
5 webserver_type: nginx-fpm
6 router_http_port: "80"
7 router_https_port: "443"
8 xdebug_enabled: false
9 additional_hostnames: []
10 additional_fqdns: []
11 mariadb_version: "10.3"
12 mysql_version: ""
13 nfs_mount_enabled: false
14 mutagen_enabled: false
15 use_dns_when_possible: true
16 composer_version: "2"
17 web_environment: []
18
19 # Key features of ddev's config.yaml:
20
```

DDEV 2024

```
You, 32 seconds ago | 2 authors (You and one other)
1 name: fldc22
2 type: drupal
3 docroot: web
4 php_version: "8.3"
5 webserver_type: nginx-fpm
6 router_http_port: "80"
7 router_https_port: "443"
8 xdebug_enabled: false
9 additional_hostnames: []
10 additional_fqdns: []
11 database:
12 | type: mariadb
13 | version: "10.3"
14 use_dns_when_possible: true
15 composer_version: "2"
16 web_environment: []
17 corepack_enable: false
18
19 # Key features of DDEV's config.yaml:
```

```
ddev_version_constraint: v1.23.0-alpha1
```

corepack_enable

What does it do?

- ddev config --corepack-enable
- DDEV uses yarn 1.22.22 unless corepack is enable and ddev restart has run.

Flags:

```
--additional-fqdns string      A comma-delimited list of F
--additional-hostnames string  A comma-delimited list of h
--auto                          Automatically run config wi
--bind-all-interfaces          Bind host ports on all inte
--composer-root string        Overrides the default Compo
--composer-root-default       Unsets a web service Compos
--composer-version string     Specify override for Compos
version.
--corepack-enable             Do 'corepack enable' to ena
--database string             Specify the database type:v
--db-image string             Sets the db container image
--db-image-default            Sets the default db contain
--db-working-dir string       Overrides the default worki
--db-working-dir-default     Unsets a db service working
--dbimage-extra-packages string A comma-delimited list of D
--ddev-version-constraint string Specify a ddev version cons
--default-container-timeout int default time in seconds tha
--disable-settings-management Prevent DDEV from creating
--disable-upload-dirs-warning Disable warnings about uplo
--docroot string             Provide the relative docroo
--fail-on-hook-fail          Decide whether 'ddev start'
-h, --help                   help for config
--host-db-port string        The db container's localhos
--host-https-port string     The web container's localho
--host-webserver-port string The web container's localho
--image-defaults             Sets the default web and db
--mailpit-http-port string   Router HTTP port to be used
--mailpit-https-port string Router port to be used for
--ngrok-args string         Provide extra args to ngrok
--no-project-mount          Whether or not to skip moun
--nodejs-version string     Specify the nodejs version
--omit-containers string     A comma-delimited list of c
--performance-mode string   Performance optimization mo
--performance-mode-reset    Reset performance mode to g
--php-version string        The version of PHP that wil
--project-name string       Provide the project name of
--project-tld string        set the top-level domain to
--project-type string       Provide the project type (o
magento, magento2, php, python, shopware6, silverstripe, typo3, word
--router-http-port string   The router HTTP port for th
```


Common Commands

Most used Commands

- ddev start
- ddev describe
- ddev list
- ddev restart
- ddev stop
- ddev delete

```
~ ddev list
```

NAME	STATUS	LOCATION	URL	TYPE
3365435-update-markup-classnames	stopped	~/Projects/contribution/drupal-contrib-addon/project_browser-3365435		drupal10
astro-paper	stopped	~/Projects/ddev/new-features/astro-paper		php
chadug24	stopped	~/Projects/drupal/ddev/Chadc23		drupal9
contribd10	stopped	~/Projects/drupal/ddev/contribd10		drupal10
dccha24	stopped	~/Projects/ddev/chadug/dccha24		drupal
ddev.com	stopped	~/Projects/ddev/ddev-website/ddev.com		php
drupal	stopped	~/Projects/drupal/ddev/test2olivero/drupal		drupal10
drupal10.2stable	stopped	~/Projects/contribution/drupal10.2stable		drupal
drupal10.3	stopped	~/Projects/contribution/drupal10.3		drupal10
drupalgovcon	stopped	~/Projects/ddev/govcon/Drupal-GovCon-2017		drupal10
emptyddev	stopped	~/Projects/ddev/emptyddev		drupal10
gratisphp	stopped	~/Projects/ddev/gratisphp		laravel
hostingintegrationddev	stopped	~/Projects/contribution/hostingintegrationddev		drupal
laravel-jetstream	stopped	~/Projects/drupal/ddev/laravel-jetstream		laravel
laravelmicro	stopped	~/Projects/ddev/laravelmicro		laravel
mautic	stopped	~/Projects/ddev/mautic		php

ddev start -- help

ddev start <project1> <project2>

ddev launch

ddev launch -- help

```
acchaz@git:(master) * ddev launch --help
Launch a browser with the current site (global shell host container command)

Usage:
  ddev launch [path] [-m|--mailpit] [flags]

Examples:
  "ddev launch" or "ddev launch /admin/reports/status/php" or "ddev launch phpinfo.php" or "ddev launch https://your.ddev.site" or "ddev launch :3000", for Mailpit "ddev launch -m"

Flags:
  -h, --help          help for launch
  -m, --mailpit       ddev launch -m launches the mailpit UI

Global Flags:
  -j, --json-output   If true, user-oriented output will be in JSON format.
```

ddev launch mailpit
ddev launch /admin
ddev launch :3000

Runs globally on any folder

DDEV	Type
start	
stop	
help	
list	
version	
restart	
delete	
debug	
config	
clean	
poweroff	

Runs inside a specific Drupal Project

DDEV	Type
nvm	Global shell web
mysql	Global shell db
npm	Global shell web
php	Global shell web
drush	Global shell web
xdebug	Global shell web
xhprof	Global shell web
composer	Global shell web
yarn	Global shell web
self-upgrade	Global shell host
launch	Global shell host
logs	

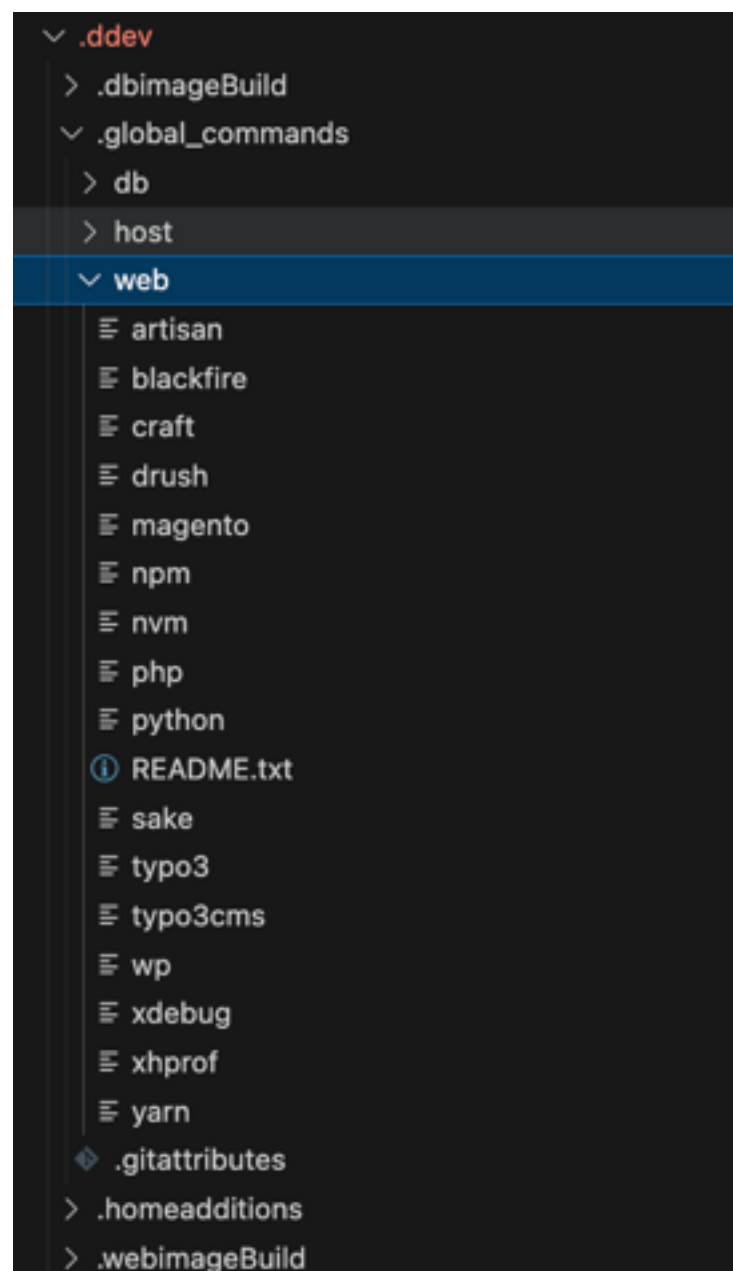
Specific to a project type or tool

DDEV	Type
artisan	Global shell web
tableplus	Global shell host
wp	Global shell web
sake	Global shell web
cake	Global shell web
magento	Global shell web
pint	Global shell web
python	Global shell web
blackfire	Global shell web
craft	Global shell web
Typo3	Global shell web
Beaver	Global shell host
heidisql	Global shell host
sequelace	Global shell host
sequelpro	Global shell host
mysqlworkbench	Global shell host

`$HOME/.ddev/global_config.yaml`

DDEV custom commands

Placement and steps required.



```
MONGO_INITDB_DATABASE: No results
1  #!/bin/bash
2
3  #ddev-generated
4  ## Description: Run drush CLI inside the web container
5  ## Usage: drush [flags] [args]
6  ## Example: "ddev drush uli" or "ddev drush sql-cli" or "ddev drush --version"
7  ## ProjectTypes: drupal7,drupal8,drupal9,drupal10,backdrop
8  ## ExecRaw: true
9
10 if ! command -v drush >/dev/null; then
11     echo "drush is not available. You may need to 'ddev composer require drush/drush'"
12     exit 1
13 fi
14 drush "$@"
15
```

```
.ddev > .global_commands > web > $ npm
1  #!/bin/bash
2  #ddev-generated
3  ## Description: Run npm inside the web container
4  ## Usage: npm [flags] [args]
5  ## Example: "ddev npm install" or "ddev npm update"
6  ## ExecRaw: true
7  ## HostWorkingDir: true
8
9  npm "$@"
10
```

DDEV host commands

What if you want to bundle commands?

ddev backup (Dump database volume and files and zip them).

ddev sync (dev, stage, prod)


ddev dbi (give it a tar.gz file and let it do the unzip and import)

You can make your own custom ddev commands that bundle existing ones as host commands.

DDEV bash conditional logic

- Flags
- Status codes
- User input
- DDEV variables available
- Many more.

```
cat << 'EOF' > while-loop-01.sh
#!/usr/bin/env bash
x=1;
while [ $x -le 5 ]; do
    echo "Hello World"
    ((x=x+1))
done
EOF
bash while-loop-01.sh
```

 Copy snippet

<https://developers.redhat.com/cheat-sheets/bash-shell-cheat-sheet>

<https://www.redhat.com/sysadmin/stupid-bash-tricks>

https://www.gnu.org/software/bash/manual/html_node/Bash-Conditional-Expressions.html

Bash conditional logic

```
web_extra_daemons:  
  - name: astro-dev-daemon  
    command: bash -c 'npm install && touch /var/tmp/npminstalldone && npm run dev -- --host'  
    directory: /var/www/html
```

```
hooks:  
  post-start:  
    - exec: bash -c 'while [ ! -f /var/tmp/npminstalldone ]; do sleep 1; done'  
    - exec: bash -c 'if [ ! -d /var/www/html/dist ]; then npm run build; fi'  
    - exec: echo -e "                                NOTICE\nTo troubleshoot any issues run \e[35mddev describe\e[0m or \e[35mddev logs --follow --time\e[0m \n"
```

What is `\e[35mddev`?

```
Starting ddev-router if necessary...  
Container ddev-router Running  
Waiting for additional project containers to become ready...  
All project containers are now ready.  
                                NOTICE  
=====
```

```
The Astro dev container is ready  
Hot Module Reloading (HMR) is available at https://ddev.com.ddev.site:4321  
To troubleshoot any issues run ddev describe or ddev logs --follow --time  
  
Successfully started ddev.com  
Project can be reached at https://ddev.com.ddev.site https://127.0.0.1:62053
```

```
.ddev > commands > web > $ gulp
1  #!/bin/bash
2
3  ## Description: Run gulp in theme directory.
4  ## Usage: gulp build|watch
5  ## Example: "ddev gulp", "ddev gulp build", "ddev gulp watch"
6
7  ## Edit the following line to set the relative path from DDEV_DOCROOT
8  ## to your custom theme root (this example works for Drupal projects.)
9  THEME=themes/custom/customthemename
10
11 ## Concatenates path to theme directory for your project.
12 DDEV_THEMEROOT=/var/www/html/${DDEV_DOCROOT}/${THEME}
13
14 ## Presupposes that gulp is installed in ${DDEV_THEMEROOT}/node_modules
15 ## You may need to "npm install" theme dependencies before running this.
16 if [ -d ${DDEV_THEMEROOT} ]
17 then
18     if [ -f ${DDEV_THEMEROOT}/node_modules/gulp/bin/gulp.js ]
19     then
20         pushd ${DDEV_THEMEROOT}
21         nvm install 18.19.1
22         nvm use 18.19.1
23         nvm alias default 18.19.1
24         ./node_modules/gulp/bin/gulp.js $@
25         nvm install 14.19.1
26         nvm use 14.19.1
27         nvm alias default 14.19.1
28         popd
29     else
30         echo "Error: No gulp.js found at ./node_modules/gulp/bin/gulp.js."
31         echo '(You may need to "npm install" theme dependencies first.)'
32     fi
33 else
34     echo "Error: Directory ${DDEV_THEMEROOT} does not exist."
35     echo '(Edit THEME variable in ./ddev/commands/web/gulp to fix this.)'
36 fi
37 exit
38
39 ## The same approach should work for other front-end build tools
40 ## (e.g., grunt, brunch, less, sass, compass, etc.)
41 ## which expect to be run with the theme directory as working directory.
42 ## It should also work when building theme assets for WordPress, etc.,
43 ## just by changing the DDEV_THEMEROOT to an appropriate directory.
```

```
.ddev > commands > web > $ gulp
You, 1 minute ago | 2 authors (Bernardo Martinez and one other)
1  #!/bin/bash
2  ## Description: Run gulp inside the web container
3  ## Usage: gulp [flags] [args]
4  ## Example: "ddev gulp"
5  ## ExecRaw: true
6
7  npm run --prefix /var/www/html/web/themes/custom/sba/ gulp "$@"
8
```

DDEV Exec

ddev exec or **ddev .**

ddev . -d /var/www/html/web/themes/custom/mytheme/ gulp

```
.ddev > commands > web > $ gulp
You, 1 minute ago | 2 authors (Bernardo Martínez and one other)
1  #!/bin/bash
2  ## Description: Run gulp inside the web container
3  ## Usage: gulp [flags] [args]
4  ## Example: "ddev gulp"
5  ## ExecRaw: true
6
7  npm run --prefix /var/www/html/web/themes/custom/sba/ gulp "$@"
8  |
```

exec

Alias: ..

Execute a shell command in the container for a service. Uses the web service by default.

To run your command in a different service container, run `ddev exec --service <service> <cmd>`. Use the `--raw` flag if you'd like to run a raw, uninterpreted command in a container.

Flags:

- `--dir, -d`: Defines the execution directory within the container.
- `--raw`: Use raw `exec` (do not interpret with Bash inside container). (default `true`)
- `--service, -s`: Defines the service to connect to. (e.g. `web, db`) (default `"web"`)

Example:

```
# List the web container's docroot contents
ddev exec ls /var/www/html

# List the web container's vendor directory contents
ddev exec --dir /var/www/html/vendor ls

# Output a long, recursive list of the files in the web container
ddev exec --raw -- ls -lR
```


DDEV hooks

Supported Command Hooks

- `pre-start` : Hooks into `ddev start` . Execute tasks before the project environment starts.



Tip

Only `exec-host` tasks can run during `pre-start` because the containers are not yet running. See [Supported Tasks](#) below.

- `post-start` : Execute tasks after the project environment has started.
- `pre-import-db` and `post-import-db` : Execute tasks before or after database import.
- `pre-import-files` and `post-import-files` : Execute tasks before or after files are imported.
- `pre-composer` and `post-composer` : Execute tasks before or after the `composer` command.
- `pre-stop`, `pre-config`, `post-config`, `pre-exec`, `post-exec`, `pre-pull`, `post-pull`, `pre-push`, `post-push`, `pre-snapshot`, `post-snapshot`, `pre-restore-snapshot`, `post-restore-snapshot` : Execute as the name suggests.
- `post-stop` : Hooks into `ddev stop` . Execute tasks after the project environment stopped.



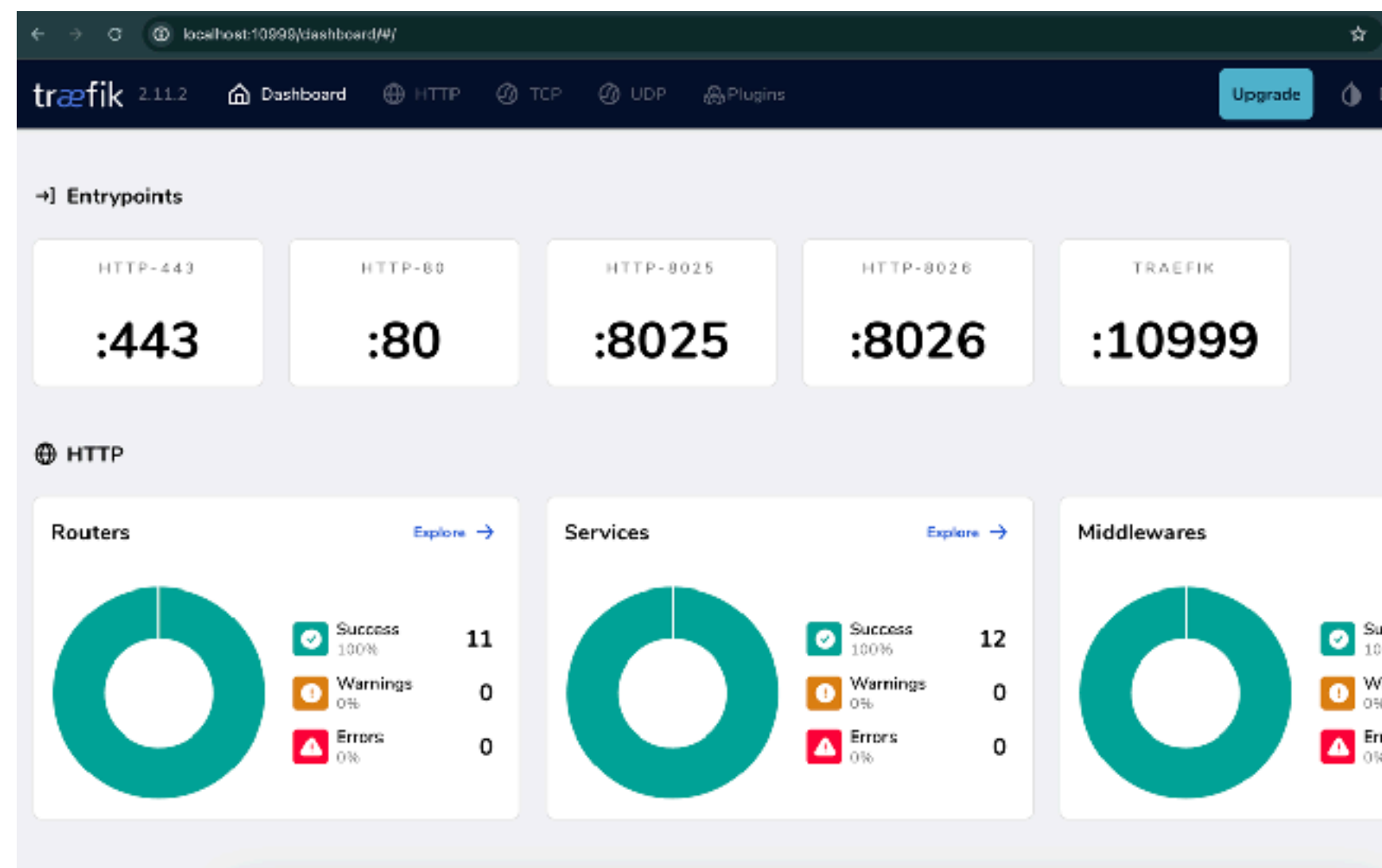
Tip

Only `exec-host` tasks can run during `post-stop` . See [Supported Tasks](#) below.

```
26 hooks:
27   post-start:
28     - exec: bash -c 'while [ ! -f /var/tmp/npminstalldone ]; do sleep 1; done'
29     - exec: bash -c 'if [ ! -d /var/www/html/dist ]; then npm run build; fi'
30     - exec: echo -e "
31                                     NOTICE
32                                     =====\n
33                                     =====\n
34                                     The Astro dev container is ready \n
35                                     Hot Module Reloadin (HMR) is available at \e[32m${DDEV_PRIMARY_URL}:4321\e[0m \n
36                                     To troubleshoot any issues run \e[35mddev describe\e[0m or \e[35mddev logs --follow --time\e[0m \n"
```

DDEV router

Traefik router



Exposing Extra Ports via `ddev-router`

If your `web` container has additional HTTP servers running inside it on different ports, those can be exposed using `web_extra_exposed_ports` in `.ddev/config.yaml`. For example, this configuration would expose a `node-vite` HTTP server running on port 3000 inside the `web` container, via `ddev-router`, to ports 9998 (HTTP) and 9999 (HTTPS), so it could be accessed via `https://<project>.ddev.site:9999`:

```
web_extra_exposed_ports:  
  - name: node-vite  
    container_port: 3000  
    http_port: 9998  
    https_port: 9999
```

<http://localhost:10999/>

<https://ddev.com/blog/working-with-vite-in-ddev/>

<https://ddev.readthedocs.io/en/stable/users/extend/traefik-router/>

DDEV custom images

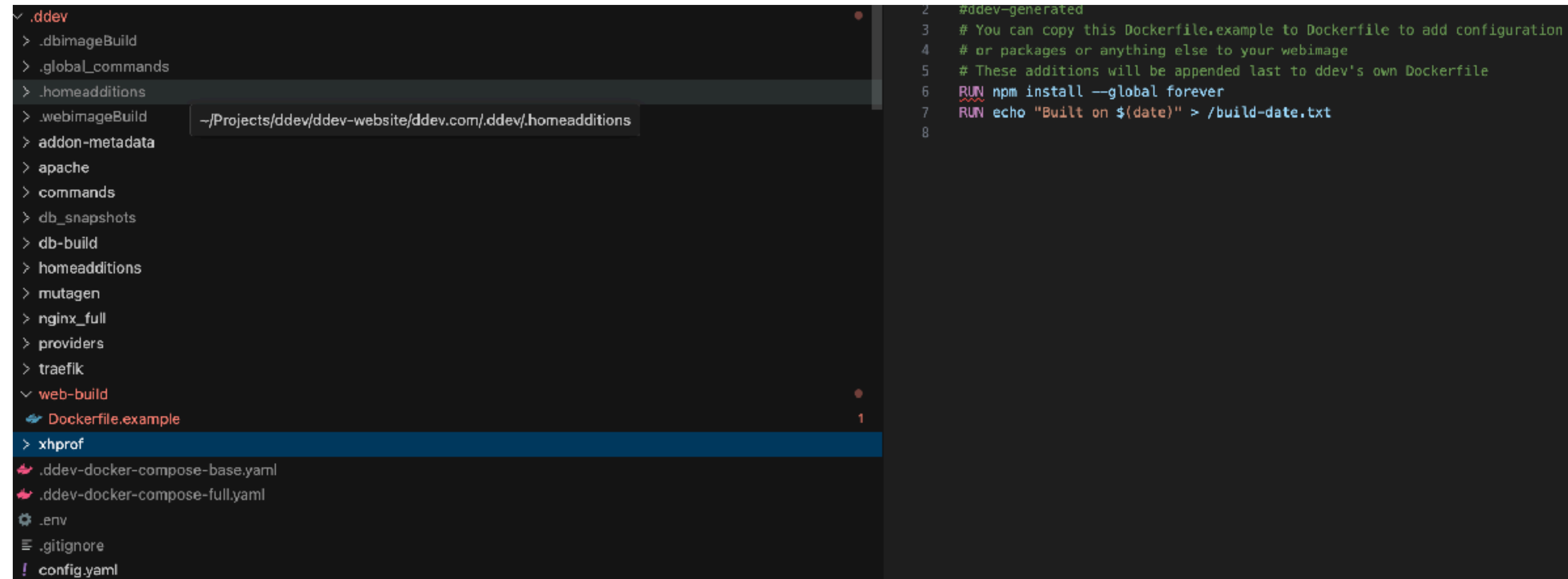
Adding extra settings to a given container

Vite HMR server

*[.ddev.site:4321](#) - works with a hook

Or

*[.ddev.site/hrm](#) - requires a custom rule



```
.ddev
├── .dbimageBuild
├── .global_commands
├── .homeadditions
├── .webimageBuild
├── addon-metadata
├── apache
├── commands
├── db_snapshots
├── db-build
├── homeadditions
├── mutagen
├── nginx_full
├── providers
├── traefik
├── web-build
├── Dockerfile.example
├── xhprof
├── .ddev-docker-compose-base.yaml
├── .ddev-docker-compose-full.yaml
├── .env
├── .gitignore
└── config.yaml
```

```
2 #ddev-generated
3 # You can copy this Dockerfile.example to Dockerfile to add configuration
4 # or packages or anything else to your webimage
5 # These additions will be appended last to ddev's own Dockerfile
6 RUN npm install --global forever
7 RUN echo "Built on $(date)" > /build-date.txt
8
```

Adding Extra Dockerfiles for `webimage` and `dbimage`

An example web image `.ddev/web-build/Dockerfile` might be:

For more complex requirements, you can add:

- `.ddev/web-build/Dockerfile`
- `.ddev/web-build/Dockerfile.*`
- `.ddev/db-build/Dockerfile`
- `.ddev/db-build/Dockerfile.*`

```
RUN npm install -g gatsby-cli
```

<https://www.lullabot.com/articles/nodejs-development-ddev>

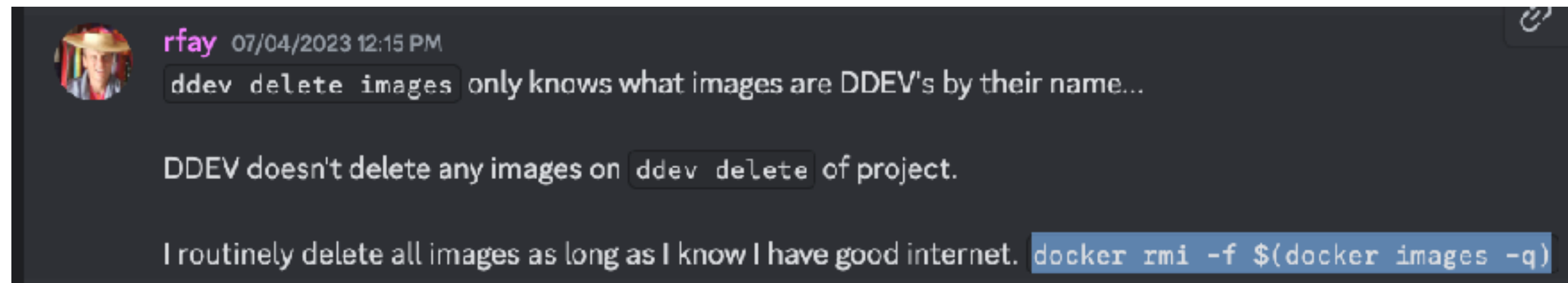
<https://ddev.readthedocs.io/en/latest/users/extend/customizing-images/#adding-extra-dockerfiles-for-webimage-and-dbimage>

<https://github.com/ddev/ddev/issues/4854#issuecomment-1876040237>

DDEV (custom images or hooks with bash conditional)

Adding extra settings to a given container

```
docker image ls
docker rmi -f $(docker images -q)
ddev delete images (v1.23.3)
```



Executing Commands in Containers

The `ddev exec` command allows you to run shell commands in the container for a DDEV service. By default, commands are executed on the web service container, in the `docroot` path of your project. This allows you to use [the developer tools included in the web container](#). For example, to run the `ls` command in the web container, you would run `ddev exec ls` or `ddev . ls`.

To run a shell command in the container for a different service, use the `--service` (or `-s`) flag at the beginning of your `exec` command to specify the service the command should be run against. For example, to run the MySQL client in the database container, you would run `ddev exec --service db mysql`. To specify the directory in which a shell command will be run, use the `--dir` flag. For example, to see the contents of the `/usr/bin` directory, you would run `ddev exec --dir /usr/bin ls`.

To run privileged commands, `sudo` can be passed into `ddev exec`. For example, to update the container's apt package lists, use `ddev exec sudo apt-get update`.

Commands can also be executed using the shorter `ddev . <cmd>` alias.

Normally, `ddev exec` commands are executed in the container using Bash, which means that environment variables and redirection and pipes can be used. For example, a complex command like `ddev exec 'ls -l ${DDEV_FILES_DIR} | grep x >/tmp/junk.out'` will be interpreted by Bash and will work. However, there are cases where Bash introduces too much

Examples

.ddev > ! config.yaml

> Find Aa ab .* No results

You, 37 seconds ago | 3 authors (Bernardo Martinez and others)

```
1 name: ddev.com
2 type: php
3 docroot: dist
4 php_version: "8.1"
5 webserver_type: nginx-fpm
6 xdebug_enabled: false
7 additional_hostnames: []
8 additional_fqdns: []
9 use_dns_when_possible: true
10 composer_version: "2"
11 web_environment: []
12 nodejs_version: "20"
13 omit_containers: ["db"]
14 disable_upload_dirs_warning: true
15 web_extra_exposed_ports:
16   - name: astro-dev
17     container_port: 4321
18     http_port: 4322
19     https_port: 4321
20 # The extra -- in `npm run dev -- --host` is a Vite requirement
21 # https://github.com/vitejs/vite/discussions/3396
22 web_extra_daemons:
23   - name: astro-dev-daemon
24     command: bash -c 'npm install && touch /var/tmp/npminstalldone && npm run dev -- --host'
25     directory: /var/www/html
26 hooks:
27   post-start:
28     - exec: bash -c 'while [ ! -f /var/tmp/npminstalldone ]; do sleep 1; done'
29     - exec: bash -c 'if [ ! -d /var/www/html/dist ]; then npm run build; fi'
30     - exec: echo -e "
31     NOTICE
32     =====\n
33     =====\n
34     The Astro dev container is ready \n
35     Hot Module Reloadin (HMR) is available at \e[32m\${DDEV_PRIMARY_URL}:4321\e[0m \n
36     To troubleshoot any issues run \e[35mddev describe\e[0m or \e[35mddev logs --follow --time\e[0m \n"
```

Maps the container port to the host

Installs the packages and it adds the background thread

Waits for npm install to be done and runs the build

ddev.com config

<https://github.com/ddev/ddev.com/blob/main/.ddev/config.yaml>

Drupalgovcon

Supports both DDEV and lando

Drupal4Gov / Drupal-GovCon-2017

Code Issues Pull requests 1 Actions Projects Wiki Security Insights

Drupal-GovCon-2017 Public Edit Pins Watch 18

develop 2 Branches 8 Tags Go to file Add file Code

mikemadison13 Merge pull request #1084 from rk12-max/DGC-8 7b453a1 · 19 hours ago 1,940 Commits

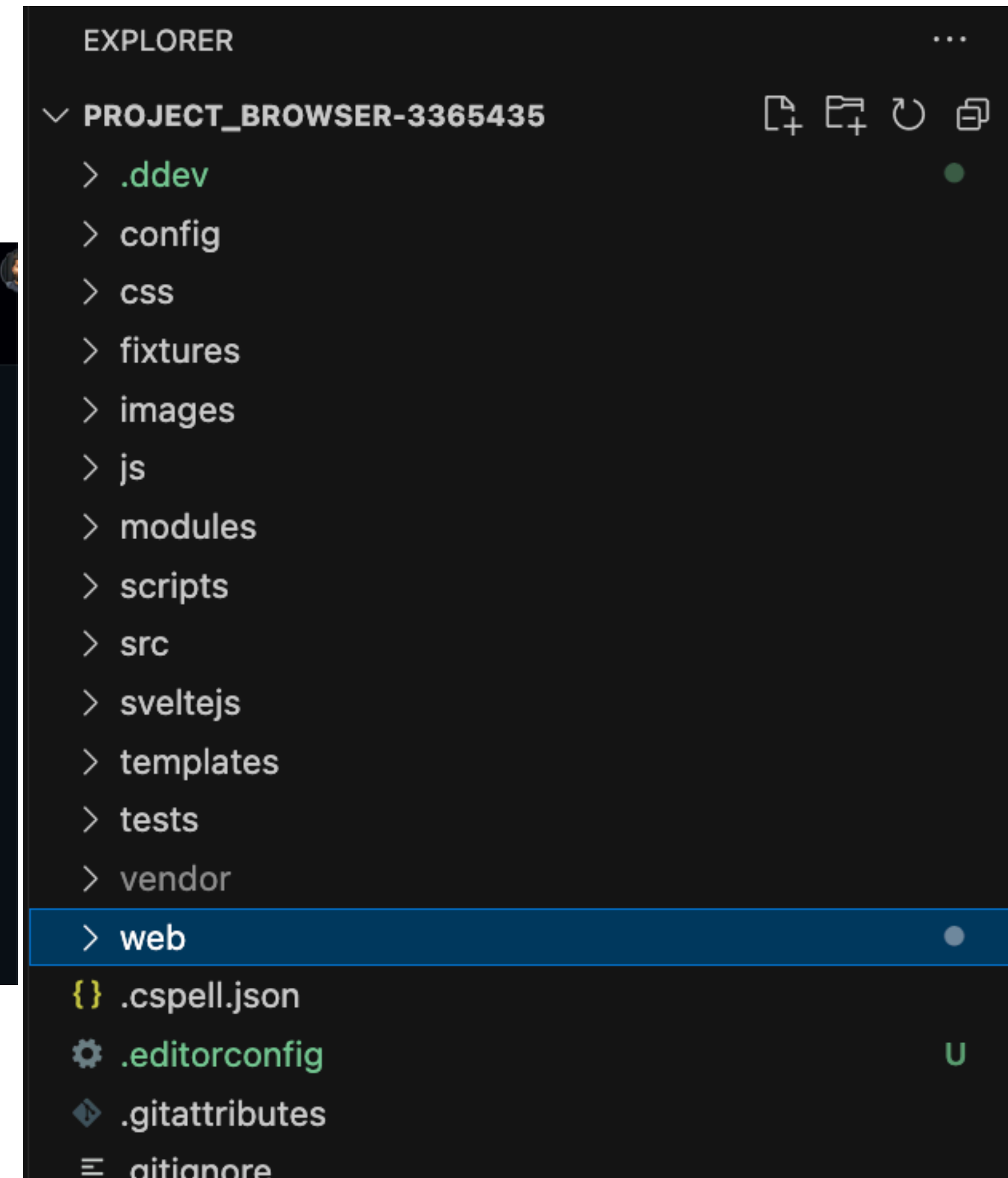
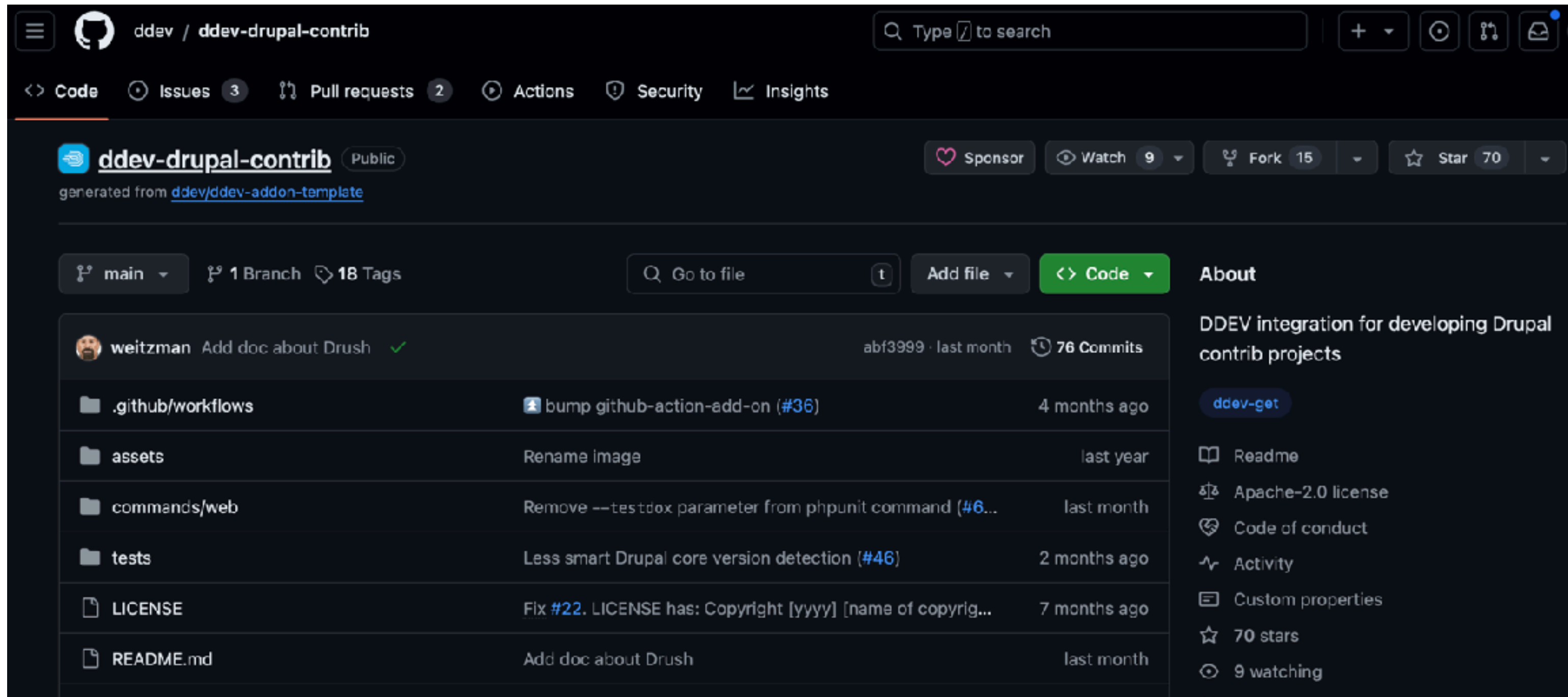
.ddev	Updated npm versions to closely match each other.	last month
.lando	Updating Lando setup to support M1 compatible Google ...	2 years ago
blt	disabling security tests for now.	3 months ago
config	additional webform config updates.	last week
docroot	DGC-8: Update the padding on image	19 hours ago
drush	acn drush alias update.	9 months ago
hooks	DGC-000: update to BLT 10.x.	5 years ago
patches	debugging behat failures.	2 years ago
tests	swapping antispam capabilities.	last year
.acquia-cli.yml	DGC-000: adding acli.	3 years ago

<https://github.com/Drupal4Gov/Drupal-GovCon-2017>

<https://www.drupalgovcon.org/>

DDEV drupal contrib

Great starting point for contrib locally.



DDEV .env files

- The global `web_environment` setting in `.ddev/global_config.yaml`.
- An optional, project-level `.ddev/.env` file, which could look something like this:

```
MY_ENV_VAR='someval'  
MY_OTHER_ENV_VAR='someotherval'
```

If you'd rather use the CLI to set the project or global `web_environment` value, you can use the `ddev config` command:

```
# Set MY_ENV_VAR for the project  
ddev config --web-environment-add="MY_ENV_VAR=someval"  
  
# Set MY_ENV_VAR globally  
ddev config global --web-environment-add="MY_ENV_VAR=someval"
```

You can use the `--web-environment` flag to overwrite existing values rather than adding them.

```
▼ .ddev  
  > .dbimageBuild  
  > .global_commands  
  > .homeadditions  
  > .webimageBuild  
  > addon-metadata  
  > apache  
  > commands  
  > db_snapshots  
  > db-build  
  > homeadditions  
  > mutagen  
  > nginx_full  
  > providers  
  > traefik  
  > web-build  
  > xhprof  
  🚢 .ddev-docker-compose-base.yaml  
  🚢 .ddev-docker-compose-full.yaml  
  ⚙️ .env  
  ☰ .gitignore  
  ! config.yaml
```

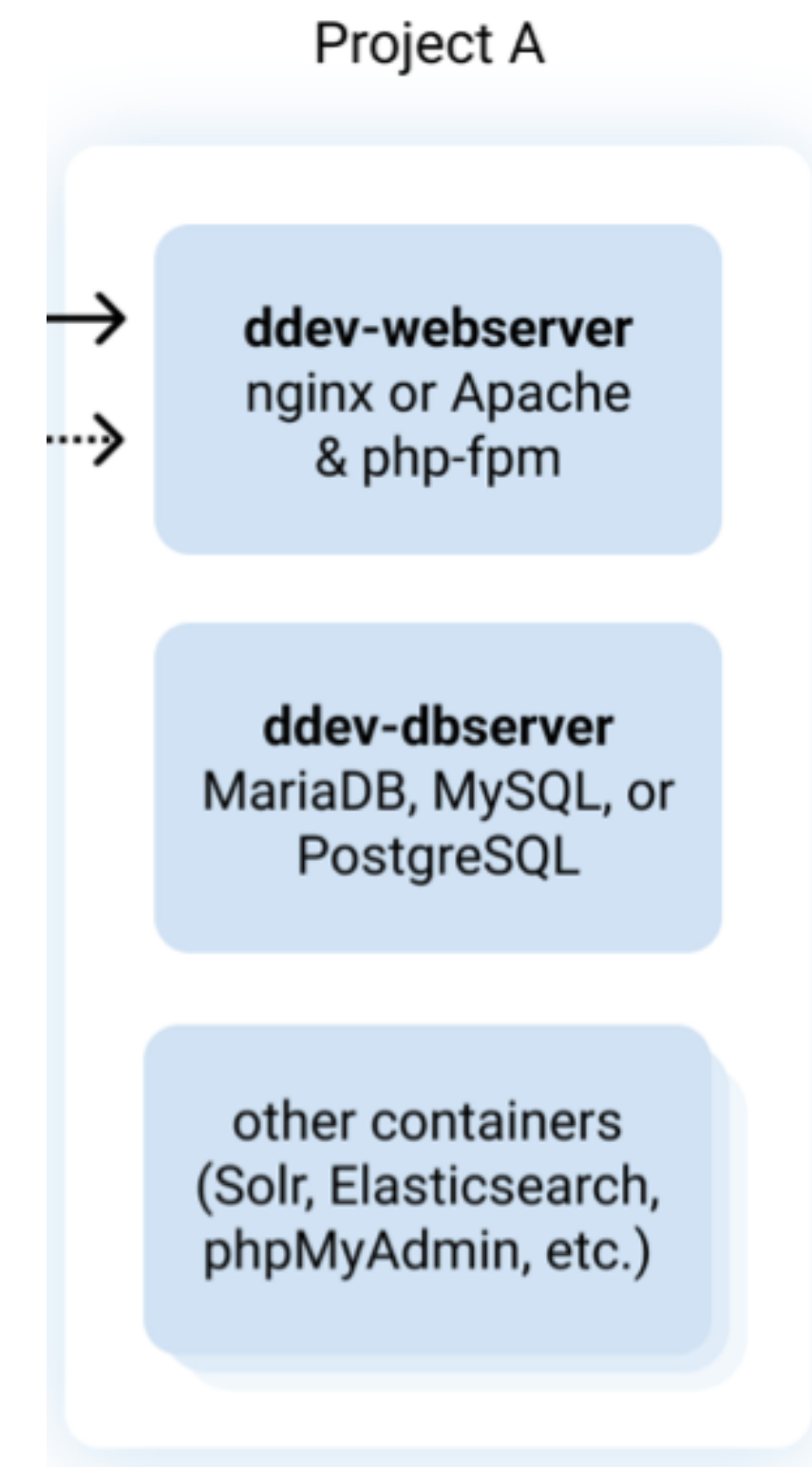
Database Management

DDEV

DDEV has one database image

One can add multiple databases of the same type on the one image.

Adding mysql and Postgres on one project requires a DDEV add-on or custom docker file.



What if you need a second database?

To either migrate or read data from

1. Add that to a separate DDEV repo and expose that project to this one.
2. Add a custom docker file with the other database. (Does not support snapshots or other custom db commands)

Example

Adding a separate project db

Let's say we have two projects, for example: project A, and project B.

To enable server-side HTTP/S communication (i.e. server-side API calls) between projects you can:

1. Either access the web container of project B directly with the hostname `ddev-<projectb>-web` and port 80 or 443 from project A:

```
# call from project A web container to project B's web container
curl https://ddev-projectb-web
```




2. Or add a `.ddev/docker-compose.communicate.yaml` to project A:






```
# add this to project A, allows connection to project B
services:
  web:
    external_links:
      - "ddev-router:projectb.ddev.site"
```

This lets the `ddev-router` know that project A can access the web container on project B's official FQDN.

Example

Adding a custom docker file

```
openmass / .ddev / docker-compose.dbmass.yaml 
weitzman and arthurbaghdas Pin portainer, fix DB persistence (#1665)  6ab4cfb · 2 years ago  History

Code Blame 29 lines (28 loc) · 1.2 KB     

1 # The following database service uses a `percona-server:5.6.40` image where mysql initialization done by
2 # a custom `docker-entrypoint.sh` script. You can find that script and the `Dockerfile` in `.circleci/mysql-rebuil
3 # See https://medium.com/massgovdigital/dev-env-5d35b97f3473#cba3 for more info.
4 # Further, the image is super sanitized (no unpublished content, old revisions, user data, etc.).
5 version: '3.6'
6
7 services:
8   dbmass:
9     container_name: ddev-${DDEV_SITENAME}-dbmass
10    image: ${MASS_DB_IMAGE-massgov/mysql-sanitized:super}
11    ports:
12      - 3206:3306
13    # These labels ensure this service is discoverable by ddev.
14    labels:
15      com.ddev.site-name: ${DDEV_SITENAME}
16      com.ddev.approot: $DDEV_APPROOT
17    environment:
18      MYSQL_USER: circle
19      MYSQL_PASSWORD: circle
20      MYSQL_DATABASE: circle
21      MYSQL_RANDOM_ROOT_PASSWORD: 1
22    command: --max_allowed_packet=32M --innodb_flush_method=0_DIRECT --tmp_table_size=16M --query_cache_size=16M
23    volumes:
24      - dbmass:/var/lib/mysql
25    web:
26      links:
27        - dbmass:dbmass
28    volumes:
29      dbmass:
```

Handy database commands

One less item to worry about

- ddev tableplus
- ddev querious
- ddev dbeaver
- ddev sequelpro
- ddev sequence

```
commands > host > $ tableplus
1  #!/bin/bash
2
3  #ddev-generated
4  # Support for TablePlus, https://tableplus.com/
5  # This command is available on macOS and WSL2 if TablePlus is installed in the default location.
6  ## Description: Run tableplus with current project database
7  ## Usage: tableplus
8  ## Example: "ddev tableplus"
9  ## OSTypes: darwin,wsl2
10 ## HostBinaryExists: /Applications/TablePlus.app,/Applications/Setapp/TablePlus.app,/mnt/c/Program Files/TablePlus/TablePlu
11
12 if [ "${DDEV_PROJECT_STATUS}" != "running" ] && [ -z "$no_recursion" ]; then
13     echo "Project ${DDEV_PROJECT} is not running, starting it"
14     ddev start
15     start_exit_code=$?
16     if [ $start_exit_code -ne 0 ]; then
17         exit $start_exit_code
18     fi
19     # run this script again, as the environment is updated after "ddev start"
20     no_recursion=true ddev "$(basename "$0")" "$@"
21     exit $?
22 fi
23
24 dbtype=${DDEV_DBIMAGE%:*}
25 driver=mysql
26 if [[ $dbtype == "postgres" ]]; then
27     driver=$dbtype
28 fi
29 query="${driver}://db:db@127.0.0.1:${DDEV_HOST_DB_PORT}/db?Environment=local&Name=ddev-${DDEV_SITENAME}"
30
31 case $OSTYPE in
32     "linux-gnu")
33         "/mnt/c/Program Files/TablePlus/TablePlus.exe" $query >/dev/null &
34         ;;
35     "darwin"*)
36         set -eu -o pipefail
37         if [ -d "/Applications/Setapp/TablePlus.app" ]; then
38             open "$query" -a "/Applications/Setapp/TablePlus.app/Contents/MacOS/TablePlus"
39         else
40             open "$query" -a "/Applications/TablePlus.app/Contents/MacOS/TablePlus"
41         fi
42         ;;
43 esac
44
```

Environment variables (user generated)

Comparison of environment variables

Providing Custom Environment Variables to a Container

You can set custom environment variables in several places:

- The project's `web_environment` setting in `.ddev/config.yaml` or `.ddev/config.*.yaml`:

```
web_environment:  
- MY_ENV_VAR=someval  
- MY_OTHER_ENV_VAR=someotherval
```

- The global `web_environment` setting in `.ddev/global_config.yaml`.
- An optional, project-level `.ddev/.env` file, which could look something like this:

```
MY_ENV_VAR='someval'  
MY_OTHER_ENV_VAR='someotherval'
```


Environment variables

```
name: slides
type: drupal10
docroot: web
php_version: "8.1"
webserver_type: nginx-fpm
xdebug_enabled: false
additional_hostnames: []
additional_fqdns: []
database:
  type: mariadb
  version: "10.4"
use_dns_when_possible: true
web_environment:
  - DRUPAL_PRIVATE=../super/secret/path
  - DRUPAL_TEMP=../secret/location
  - DRUPAL_CONFIG_SYNC=../secret/path/config/sync
  - DRUSH_OPTIONS_URI=$DDEV_PRIMARY_URL #only needed if one disables settings management.
composer_version: "2"
web_environment: []
```

- An optional, project-level `.ddev/.env` file, which could look something like this:

```
MY_ENV_VAR='someval'
MY_OTHER_ENV_VAR='someotherval'
```

disable_settings_management

What does it do?

Tells DDEV to use a specific project type without creating setting files and creates the .ddev/.gitignore

```
web/sites/default/settings.ddev.php
```

```
web/sites/default/settings.php
```

```
// Automatically generated include for settings managed by ddev.  
if (getenv('IS_DDEV_PROJECT') == 'true' && file_exists(__DIR__ . '/settings.ddev.php')) {  
    include __DIR__ . '/settings.ddev.php';  
}
```

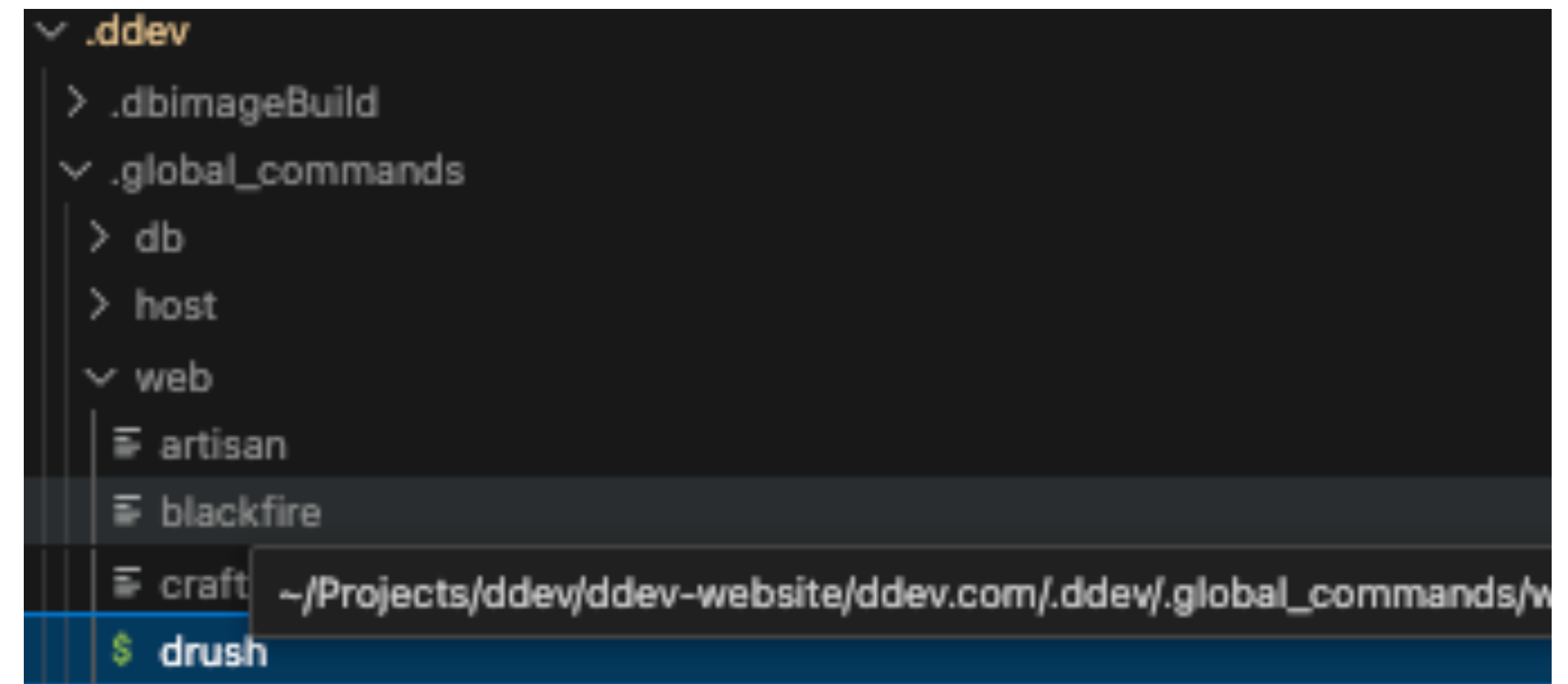
But what about the BLT/ACLI command

```
ddev exec acli
```

```
#!/bin/bash
```

```
#ddev-generated  
## Description: Run drush CLI inside the web container  
## Usage: drush [flags] [args]  
## Example: "ddev drush uli" or "ddev drush sql-cli" or "ddev drush --version"  
## ProjectTypes: drupal7,drupal8,drupal9,drupal10,backdrop  
## ExecRaw: true
```

```
if ! command -v drush >/dev/null; then  
  echo "drush is not available. You may need to 'ddev composer require drush/drush'"  
  exit 1  
fi  
drush "$@"
```



Hosting providers

Steps needed

- Platform.sh
- Acquia
- Pantheon
- Lagoon
- Rsync

```
.ddev > providers > ! platform.yaml
1 #ddev-generated
2 # Platform.sh provider configuration. This works out of the box, but can be edited
3 # your own preferences. If you edit it, remove the 'ddev-generated' line from the
4 # that it won't be overwritten.
5
6 # Consider using 'ddev get ddev/ddev-platformsh' (https://github.com/ddev/ddev-pla)
7 # complete platform integration.
8
9 # To use this configuration,
10
11 # 1. Check out the site from platform.sh and then configure it with 'ddev config'.
12 # 2. Obtain and configure an API token.
13 #   a. Login to the Platform.sh Dashboard and go to Account->API Tokens to create
14 #   b. Add the API token to the 'web_environment' section in your global ddev con
15 #   ```yaml
16 #   web_environment:
17 #     - PLATFORMSH_CLI_TOKEN=abcdeyourtoken
18 #   ```
19 # 3. Add PLATFORM_PROJECT and PLATFORM_ENVIRONMENT and optional PLATFORM_APP (only
20 #   ```yaml
21 #   web_environment:
22 #     - PLATFORM_PROJECT=nf4amudfn23biyourproject
23 #     - PLATFORM_ENVIRONMENT=main
24 #     - PLATFORM_APP=app
25 # 4. 'ddev restart'
26 # 5. Run 'ddev pull platform'. After you agree to the prompt, the current upstream
27 # 6. Optionally use 'ddev push platform' to push local files and database to platf
28
```

```
.ddev > providers > ! acquia.yaml
1 #ddev-generated
2 # Acquia provider configuration.
3
4 # To use this configuration,
5
6 # 1. Get your Acquia API token from your Account Settings->API Tokens.
7 # 2. Make sure your ssh key is authorized on your Acquia account at Account Set
8 # 3. 'ddev auth ssh' (this typically needs only be done once per ddev session,
9 # 4. Add / update the web_environment section in ~/.ddev/global_config.yaml
10 # or your project config.yaml with the API keys:
11 #   ```yaml
12 #   web_environment:
13 #     - ACQUIA_API_KEY=xxxxxxxxx
14 #     - ACQUIA_API_SECRET=xxxxxx
15 #   ```
16 # 5. Add the ACQUIA_ENVIRONMENT_ID environment variable to your project config.
17 #   ```yaml
18 #   web_environment:
19 #     - ACQUIA_ENVIRONMENT_ID=project1.dev
20 #   - On the Acquia Cloud Platform you can find this out by navigating to the e
21 #   clicking on the header and look for the "SSH URL" line.
22 #   Eg. 'project1.dev@cool-projects.acquia-sites.com' would have a project ID
```

```
.ddev > providers > ! lagoon.yaml
1 #ddev-generated
2 # Lagoon provider configuration.
3
4 # To use this configuration,
5
6 # 1. Check out the project and then configure it with 'ddev co
7 # 2. Configure an SSH key for your Lagoon user https://docs.la
8 # 3. 'ddev auth ssh'.
9 # 4. Add LAGOON_PROJECT and LAGOON_ENVIRONMENT variables to yo
10 # 5. 'ddev restart'
11 #
12 # 'ddev pull lagoon'
13
14 auth_command:
15   command: |
16     set -eu -o pipefail
17     ssh-add -l >/dev/null || ( echo "Please 'ddev auth ssh' be
18     if [ -z "${LAGOON_PROJECT:-}" ]; then echo "Please make su
19     if [ -z "${LAGOON_ENVIRONMENT:-}" ]; then echo "Please make
```

Pantheon

Needs some sponsorship.

```
! pantheon.yaml 27
☰ pantheon.yaml.example 28
! platform.yaml 29
```

```
# Debugging: Use `ddev exec terminus auth:whoami` to see what terminus knows about
# `ddev exec terminus site:list` will show available sites

environment_variables:
  project: dccha24.dev
```

```
web_environment:
  - TERMINUS_MACHINE_TOKEN=a2s1fUhirYjKA4UDi23R8QS87TSDi23-92KA4UDi23qWV
  corepack_enable: true
```

Fake look alike token

```
# Pull a backup from the configured Pantheon project to use locally
ddev pull pantheon

# Pull a backup from the configured Platform.sh project to use locally
ddev pull platform

# Pull a backup from the configured Pantheon project without confirming
ddev pull pantheon -y

# Pull the Platform.sh database archive *only* without confirming
ddev pull platform --skip-files -y

# Pull the localfile integration's files *only* without confirming
ddev pull localfile --skip-db -y

# Pull from Platform.sh specifying the environment variables PLATFORM_ENVIRONMENT
ddev pull platform --environment=PLATFORM_ENVIRONMENT=main,PLATFORMSH_CLI_TOKEN=at
```

ddev pull pantheon --environment=project=dccha24.prod
<https://ddev.readthedocs.io/en/stable/users/usage/commands/#pull>

ddev sync -d = dev env files and db
ddev sync -d-f = dev env files only
ddev sync -d-q = dev env database only

```
.ddev > commands > host > $ sync
You, 2 weeks ago | 1 author (You)
1  #!/bin/bash
2  ## Description: Run sync
3  ## Usage: sync [flags] [args]
4  ## Example: "ddev sync [environment] ddev sync -d-q(dev env, db only) or ddev sync -d-f(dev env, files only)"
5  ## ExecRaw: true
6  ## HostWorkingDir: true
7  ## Flags: [{"Name":"development","Shorthand":"d","Usage":"ddev sync -d syncs the dev environment"}, {"Name":"stage","Shorthand":"s","Usage":"ddev sync -s syncs the
8
9  if [ "${DDEV_PROJECT_STATUS}" != "running" ] && [ -z "$no_recursion" ]; then
10 echo "Project ${DDEV_PROJECT} is not running, starting it"
11 ddev start
12 start_exit_code=$?
13 if [ $start_exit_code -ne 0 ]; then
14   exit $start_exit_code
15 fi
16 # run this script again, as the environment is updated after "ddev start"
17 no_recursion=true ddev "s$(basename "$0")" "$@"
18 exit $?
19 fi
20
21 while ;; do
22   case ${1:-} in
23     -d-q|--development--database|--dev--database)
24       ddev pull pantheon --environment=project=dccha24.dev --skip-files -y
25       ;;
26     -d-f|--development--files|--dev--files)
27       ddev pull pantheon --environment=project=dccha24.dev --skip-db -y
28       ;;
29     -d|--development|--dev)
30       ddev pull pantheon --environment=project=dccha24.dev
31       ;;
32     -s-q|--stage--database|--stg--database)
33       ddev pull pantheon --environment=project=dccha24.stage --skip-files -y
34       ;;
35     -s-f|--stage--files|--stg--files)
36       ddev pull pantheon --environment=project=dccha24.stage --skip-db -y
37       ;;
38     -s|--stage|--stg)
39       ddev pull pantheon --environment=project=dccha24.stage
40       ;;
41     -p-q|--production--database|--prod--database)
42       ddev pull pantheon --environment=project=dccha24.prod --skip-files -y
43       ;;
44     -p-f|--production--files|--prod--files)
45       ddev pull pantheon --environment=project=dccha24.prod --skip-db -y
46       ;;
47     -p|--production|--prod)
48       ddev pull pantheon --environment=project=dccha24.prod
49       ;;
50     --)
51       # End of all options.
52       shift
53       break
54       ;;
55     -?*)
56       printf 'WARN: Unknown option (ignored): %s\n' "$1" >>2
57       ;;
58     *)
59       # Default case: No more options, so break out of the loop.
60       ddev sync --help
61       echo "Don't forget to use the flags, ddev sync -d-q for the development without the files."
62       break
63     esac
64   shift
65 done
```


Example

DDEVs Addons

```
ddev get --list --all
ddev get <filepath>
ddev get remove <addon> (CI)
```

```
no read instructions for this add-on at the source repo at
://github.com/ddev/ddev-phpmyadmin
ie file issues and create pull requests there to improve it.
illed phpmyadmin:v0.3.5 from ddev/ddev-phpmyadmin
rfay-tag1-m1:~/workspace/d10$ ddev get ~/workspace/ddev-phpmyadmin/
Container ddev-router Recreated
Container ddev-router Started
Waiting for additional project containers to become ready...
Project containers are now ready.
Updated d10
Project can be reached at https://d10.ddev.site https://127.0.0.1:33202
... completed in 1m43s.
```

<https://github.com/ddev/ddev-drupal-contrib>

<https://ddev.readthedocs.io/en/latest/users/extend/additional-services/>

```
ddev-contrib / docker-compose-services / mongodb / docker-compose.mongo.yaml
```

rfay Use expose: instead of ports: with mongodb recipe

Code Blame 48 lines (43 loc) · 1000 Bytes

```
1 version: '3.6'
2
3 services:
4   mongo:
5     container_name: ddev-$(DDEV_SITENAME)-mongo
6     image: mongo:4.0
7     volumes:
8     - type: "volume"
9       source: mongo
10      target: "/data/db"
11      volume:
12        nocopy: true
13    restart: "no"
14    expose:
15    - "27017"
16    labels:
17      com.ddev.site-name: ${DDEV_SITENAME}
18      com.ddev.approot: $DDEV_APPROOT
19    environment:
20    - MONGO_INITDB_ROOT_USERNAME=db
21    - MONGO_INITDB_ROOT_PASSWORD=db
22    - MONGO_INITDB_DATABASE=db
23
24   mongo-express:
25     container_name: ddev-$(DDEV_SITENAME)-mongo-express
26     image: mongo-express:0.40
27     restart: "no"
28     labels:
29       com.ddev.site-name: ${DDEV_SITENAME}
30       com.ddev.approot: ${DDEV_APPROOT}
31       com.ddev.platform: ddev
32
33     links:
34     - mongo:mongo
35     expose:
36     - "8081"
37     environment:
38       VIRTUAL_HOST: $DDEV_HOSTNAME
39       ME_CONFIG_MONGODB_ADMINUSERNAME: db
40       ME_CONFIG_MONGODB_ADMINPASSWORD: db
41       HTTP_EXPOSE: "8081:8081"
42
43     web:
44       links:
45       - mongo:mongo
46
47   volumes:
48     mongo:
```

Resources

DDEV code contributor workflow

Training

Date	Description and Recording
2023-07-11	Testing DDEV Features, testing DDEV v1.22.0 prereleases, how to help
2023-07-18	Maintaining and Improving the Docs
2023-07-25	Setting up a Go Development Environment
2023-08-08	Running Automated Tests Locally
2023-08-15	Maintain and improve ddev.com see outline and blog post
2023-08-22	Supporting others: How can I help? see blog
2023-08-29	Building and pushing an improved Docker image see outline
2023-09-05	Marketing: How can I help?
2023-09-12	DDEV Architecture: Go, Docker Compose, Containers , see outline
2023-09-19	DDEV Testing Infrastructure , see outline
2023-09-26	DDEV Automated Tests and How to Improve Them, Randy forgot to record! but see outline
2023-10-31	Measuring and Analyzing User Data with Amplitude see outline
2023-11-07	DDEV Add-ons: Creating, maintaining, testing see outline
2023-11-14	DDEV Hosting Providers: Creating, Maintaining, Using , see outline .

Upcoming Trainings:

- Advanced DDEV Add-On Techniques, including automated testing and debugging tests
- DDEV automated tests and how to improve them
- Hostname Resolution and Debugging in DDEV
- Using AI in DDEV Development (Continue.dev, etc.)
- Interpreting **ddev debug test** To Support Users
- Maintaining DDEV Docker Images
- Implementing a New DDEV Project Type
- Managing and Maintaining DDEV Docs (Readthedocs.io, etc.)
- Using Tmate to Debug GitHub Workflows
- Using WSL2 with DDEV
- Git for Fun and Profit
- Making Your Own **ddev share** Variant
- Go Development Environment with Golang or vscode

<https://ddev.com/blog/contributor-training/>

<https://ddev.readthedocs.io/en/latest/developers/building-contributing/>

DDEV quick starts

Drupal

Drupal 10

Drupal 9

Drupal 6/7

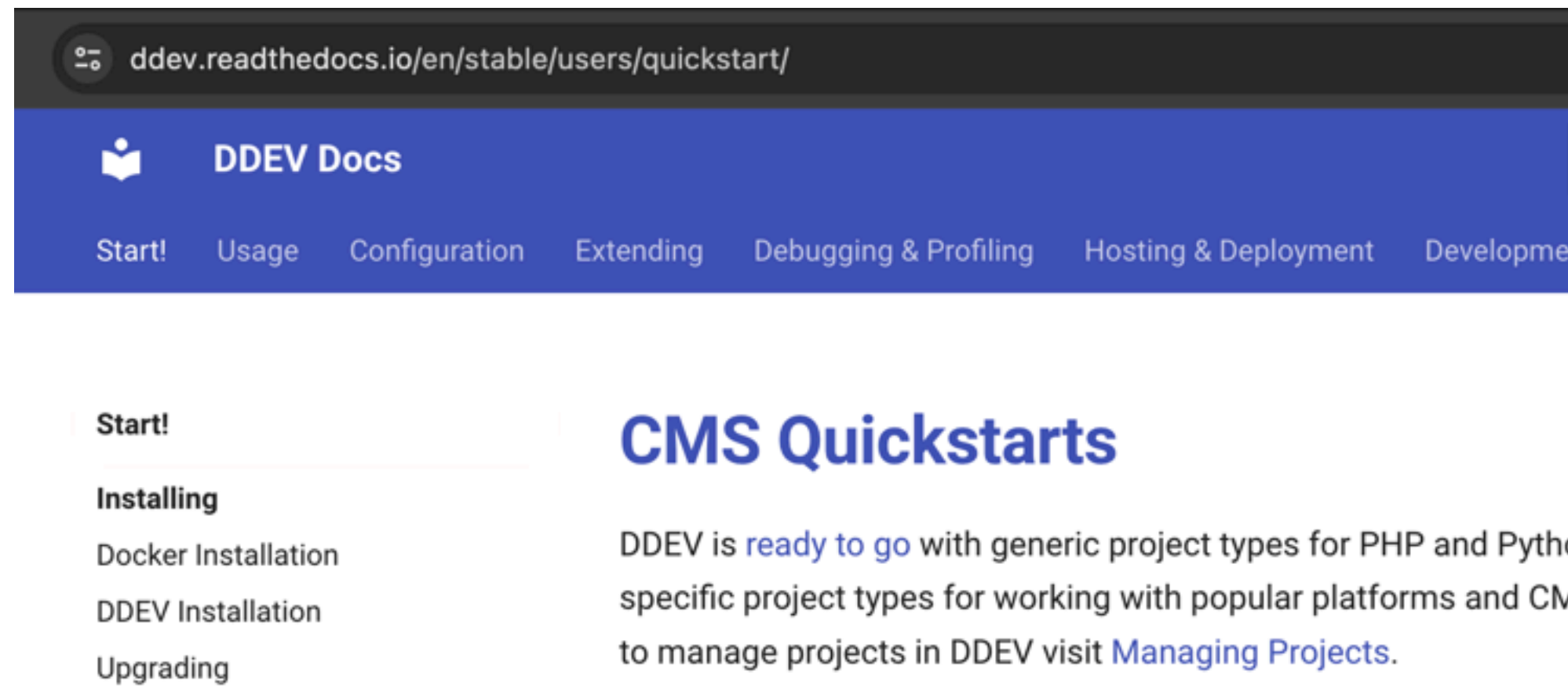
Git Clone

```
mkdir my-drupal10-site
cd my-drupal10-site
ddev config --project-type=drupal10 --docroot=web
ddev start
ddev composer create drupal/recommended-project
ddev composer require drush/drush
ddev drush site:install --account-name=admin --account-pass=admin -y
# use the one-time link (CTRL/CMD + Click) from the command below to edit your
ddev drush uli
ddev launch
```

<https://docs.lando.dev/getting-started/first-app.html>

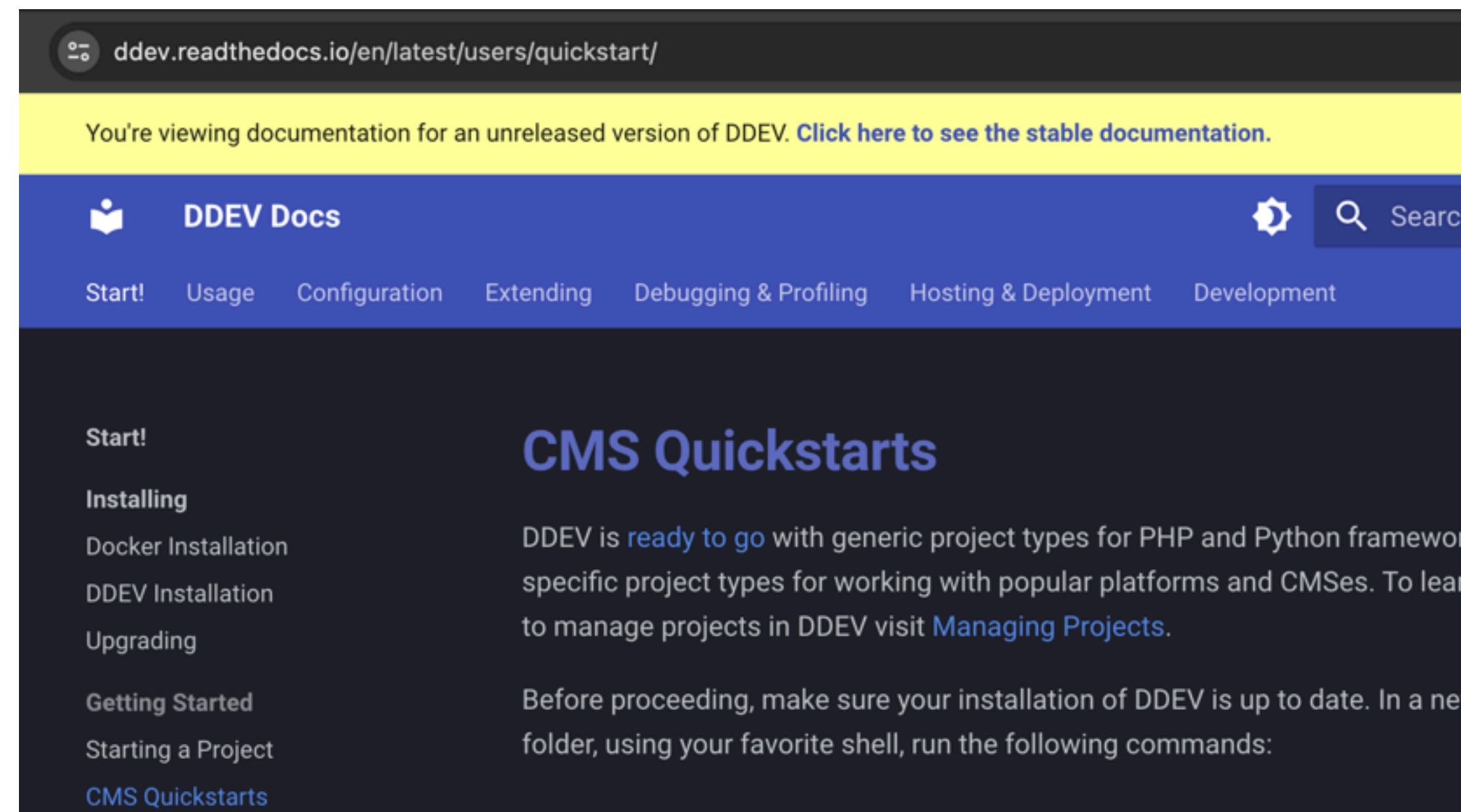
<https://ddev.readthedocs.io/en/latest/users/quickstart/#craft-cms>

DDEV latest vs DDEV stable docs



The screenshot shows the DDEV stable documentation page. The browser address bar displays `ddev.readthedocs.io/en/stable/users/quickstart/`. The page features a blue header with the 'DDEV Docs' logo and a navigation menu including 'Start!', 'Usage', 'Configuration', 'Extending', 'Debugging & Profiling', 'Hosting & Deployment', and 'Development'. On the left, a sidebar lists sections: 'Start!', 'Installing' (with sub-items: 'Docker Installation', 'DDEV Installation', 'Upgrading'), 'Getting Started', and 'Starting a Project'. The main content area is titled 'CMS Quickstarts' and contains the text: 'DDEV is ready to go with generic project types for PHP and Python specific project types for working with popular platforms and CMSes. To learn more about how to manage projects in DDEV visit [Managing Projects](#).'

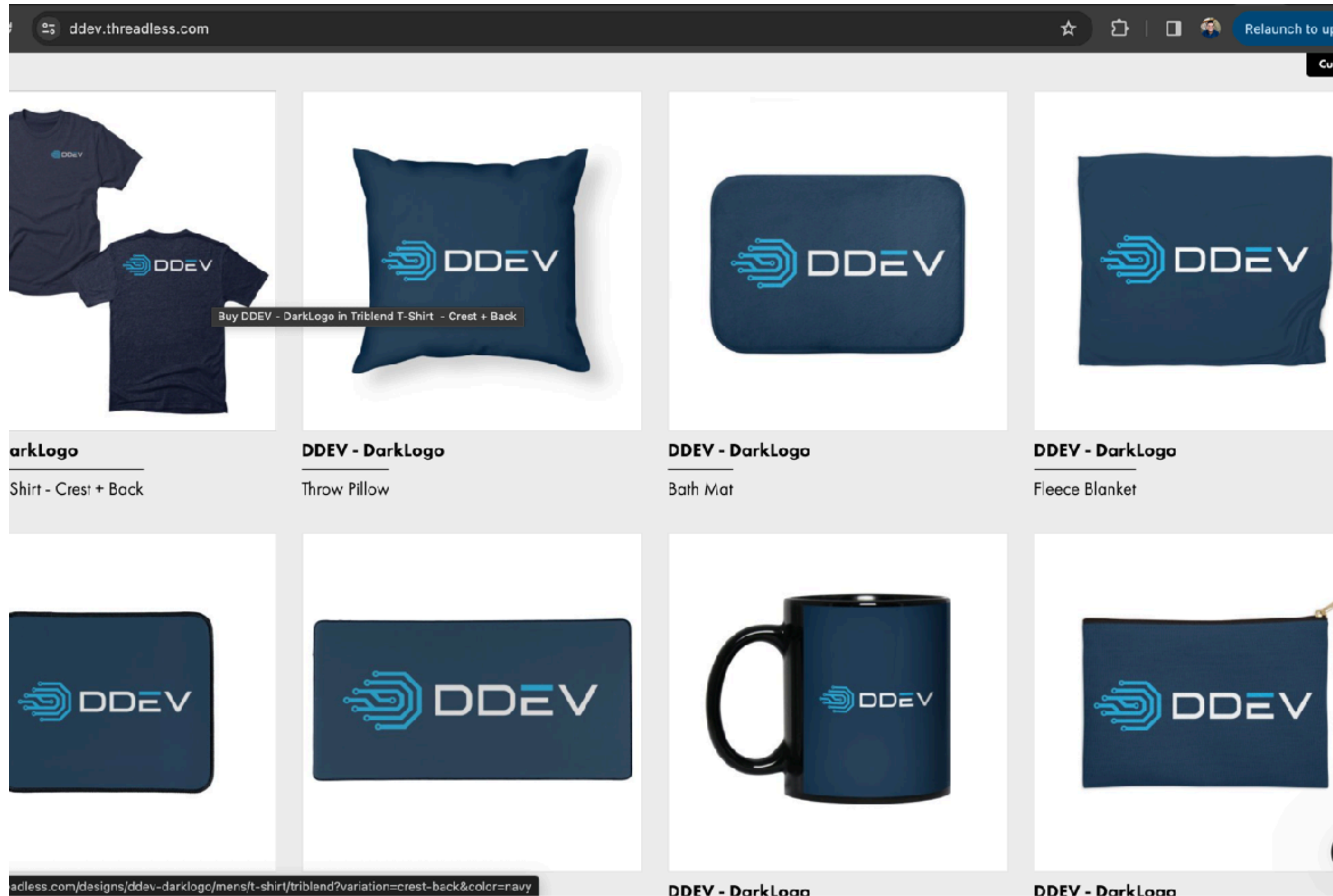
<https://ddev.readthedocs.io/en/stable/users/quickstart/>



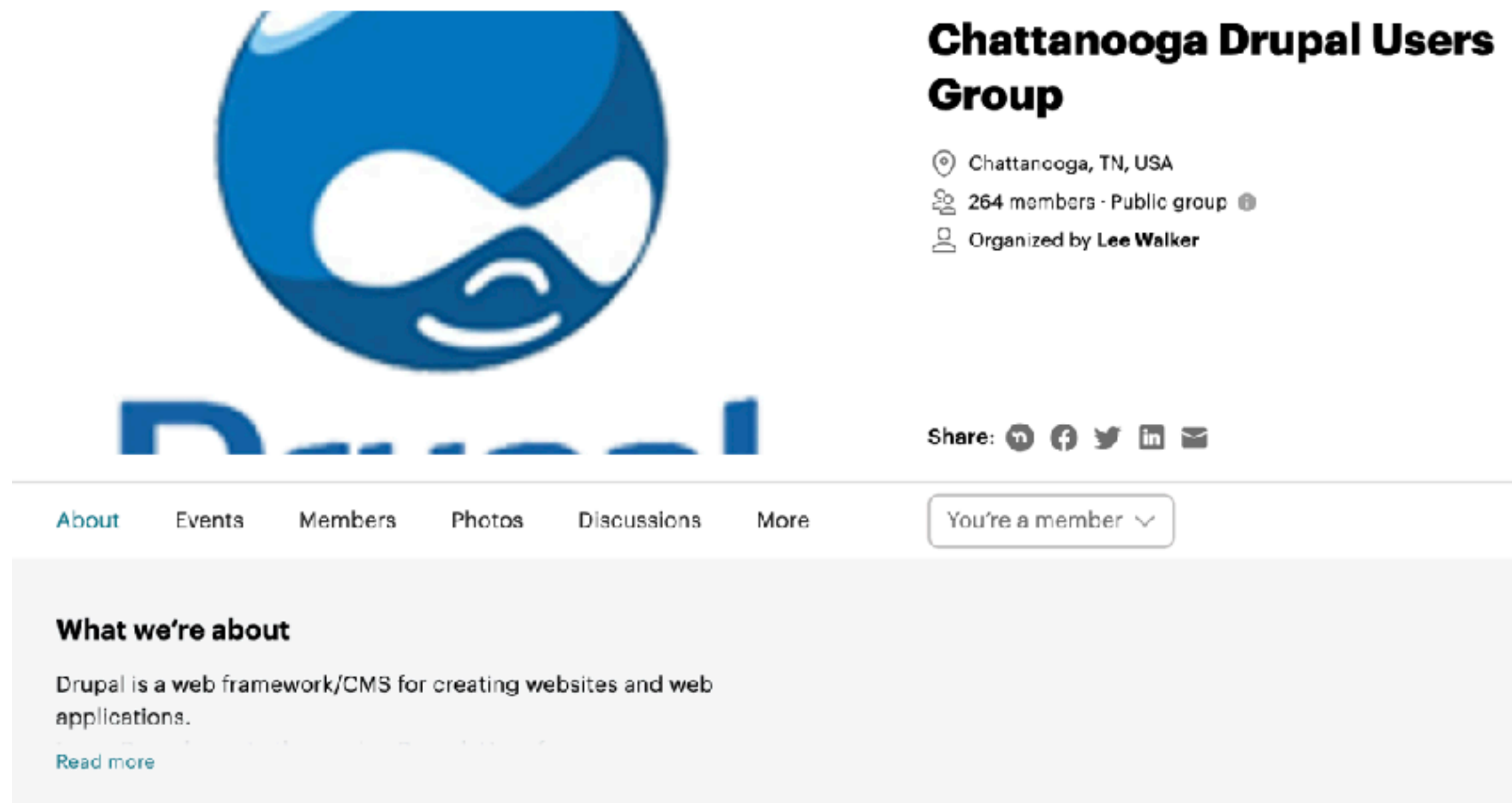
The screenshot shows the DDEV latest documentation page. The browser address bar displays `ddev.readthedocs.io/en/latest/users/quickstart/`. A yellow warning banner at the top states: 'You're viewing documentation for an unreleased version of DDEV. [Click here to see the stable documentation](#).' The page features a blue header with the 'DDEV Docs' logo, a search bar, and a navigation menu including 'Start!', 'Usage', 'Configuration', 'Extending', 'Debugging & Profiling', 'Hosting & Deployment', and 'Development'. On the left, a sidebar lists sections: 'Start!', 'Installing' (with sub-items: 'Docker Installation', 'DDEV Installation', 'Upgrading'), 'Getting Started' (with sub-items: 'Starting a Project', 'CMS Quickstarts'), and 'Starting a Project'. The main content area is titled 'CMS Quickstarts' and contains the text: 'DDEV is ready to go with generic project types for PHP and Python framework specific project types for working with popular platforms and CMSes. To learn more about how to manage projects in DDEV visit [Managing Projects](#).' Below this, it says: 'Before proceeding, make sure your installation of DDEV is up to date. In a new folder, using your favorite shell, run the following commands:'.

<https://ddev.readthedocs.io/en/latest/users/quickstart/>

Checkout the DDEV store



Connect with us



The screenshot shows the Meetup profile for the Chattanooga Drupal Users Group. On the left is a large blue and white Drupal logo. To the right, the group name "Chattanooga Drupal Users Group" is displayed in bold. Below the name, it lists the location as "Chattanooga, TN, USA", the number of members as "264 members · Public group", and the organizer as "Lee Walker". A "Share" button with social media icons is visible. At the bottom, there is a navigation menu with "About", "Events", "Members", "Photos", "Discussions", and "More", along with a "You're a member" button. A section titled "What we're about" contains the text: "Drupal is a web framework/CMS for creating websites and web applications." and a "Read more" link.

<https://www.meetup.com/Chattanooga-Drupal-Users-Group/>

<https://ddev.com/>



<https://discord.com/invite/hCZFfAMc5k>



<https://www.drupalcampchattanooga.com/>

Questions?

THANK YOU!