

Drupal local apache setup

I want to install a drupal project and host it in my local apache server. I want to create another virtual host in my apache server rather than var/www/html, the virtual host's name should be drupalSite. I am using ubuntu as my os. I also have mysql, composer installed. My php version is 8.3.24. I have all the extensions installed in my php. I want to create a virtual host for my project. And then in the host I want to install the drupal project and run it locally. The server name for this should be www.drupalSite.com. Walk me through all the steps to do that. If you have any query before answering please do that first, after that please start the walkthrough.

Awesome — I can walk you through it end-to-end.

Quick checks before we start (answer if you want, otherwise I'll proceed with sensible defaults):

1. Are we creating a brand-new Drupal project, or do you already have a repo/codebase?
2. Preferred project path? (I'll use `/var/www/drupalSite`.)
3. Install via browser or via Drush CLI? (I'll default to browser.)

I'll go ahead with: **new project**, `/var/www/drupalSite`, **browser installer**.

Step-by-step walkthrough (Ubuntu + Apache + PHP 8.3 + MySQL)

1) Prep: Apache modules & base packages

```
bash
```

```
sudo apt update
```

```
sudo a2enmod rewrite
```

```
# (optional but nice to have) headers, expires
```

```
sudo a2enmod headers expires
sudo systemctl restart apache2
```

2) Create the project directory

We'll put the Composer project in `/var/www/drupalSite`; the Drupal docroot will be the `web/` subfolder created by the template.

```
bash

sudo mkdir -p /var/www/drupalSite
sudo chown -R $USER:www-data /var/www/drupalSite
sudo chmod -R 775 /var/www/drupalSite
```

3) Create a new Drupal project with Composer

```
bash

cd /var/www/drupalSite
composer create-project drupal/recommended-project .
```

This creates:

- `/var/www/drupalSite/web` ← this is the **DocumentRoot**
- Composer-managed core & vendor files outside the docroot (best practice)

4) Create the MySQL database & user

Open MySQL:

```
bash

sudo mysql -u root
```

Inside the MySQL prompt:

```
sql

CREATE DATABASE drupal_site CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE USER 'drupal'@'localhost' IDENTIFIED BY 'a-very-strong-password';
GRANT ALL PRIVILEGES ON drupal_site.* TO 'drupal'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

5) Apache virtual host for `www.drupalSite.com`

Create a new vhost file:

```
bash

sudo nano /etc/apache2/sites-available/drupalSite.conf
```

Paste this:

```
apache

<VirtualHost *:80>
    ServerName www.drupalSite.com
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/drupalSite/web

    <Directory /var/www/drupalSite/web>
        AllowOverride All
        Require all granted
        Options FollowSymLinks
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/drupalSite_error.log
    CustomLog ${APACHE_LOG_DIR}/drupalSite_access.log combined
</VirtualHost>
```

Enable the site (you can keep the default site enabled too; they won't conflict if ServerName differs):

```
bash

sudo a2ensite drupalSite.conf
sudo systemctl reload apache2
```

6) Map the hostname to localhost

Add to `/etc/hosts` :

```
bash

sudo nano /etc/hosts
```

Add this line (keep your existing ones):

```
127.0.0.1 www.drupalSite.com
```

Save.

7) File/dir permissions Drupal needs to write

Drupal needs to write to `sites/default/files` (created during install) and sometimes to `settings.php` during installation:

```
bash
```

```
# Make sure default directory exists & is writable
```

```
mkdir -p /var/www/drupalSite/web/sites/default/files
```

```
chmod 775 /var/www/drupalSite/web/sites/default/files
```

```
chown -R $USER:www-data /var/www/drupalSite/web/sites/default/files
```

(If the installer asks to temporarily write `settings.php`, we'll handle that on the fly. Typically the installer can create it if directory perms allow.)

8) Run the installer in your browser

Open:

```
arduino
```

```
http://www.drupalSite.com
```

Follow the wizard:

- Choose **Standard** installation profile.
- Database type: **MySQL**
 - Database name: `drupal_site`
 - Username: `drupal`
 - Password: `a-very-strong-password`
- Site name, admin user, etc.

If the installer complains about `settings.php`:

```
bash
```

```
cp /var/www/drupalSite/web/sites/default/default.settings.php
/var/www/drupalSite/web/sites/default/settings.php
chmod 664 /var/www/drupalSite/web/sites/default/settings.php
chown $USER:www-data /var/www/drupalSite/web/sites/default/settings.php
```

Reload the page and continue. After install finishes, you can tighten it:

```
bash

chmod 644 /var/www/drupalSite/web/sites/default/settings.php
```

9) (Optional) Install Drush for handy CLI tasks

```
bash

cd /var/www/drupalSite
composer require drush/drush
./vendor/bin/drush status
```

Examples:

```
bash

# Clear caches
./vendor/bin/drush cr

# Run site install via CLI (alternative to the browser):
./vendor/bin/drush site:install -y --db-url=mysql://drupal:a-very-strong-
password@127.0.0.1/drupal_site --site-name="DrupalSite"
```

10) Verify clean URLs & .htaccess

Because we enabled `mod_rewrite` and `AllowOverride All`, pretty URLs should work. If you see 404s on subpages, double-check:

- `a2enmod rewrite` (already done)
- `<Directory>` block has `AllowOverride All`
- `DocumentRoot` points to `/var/www/drupalSite/web`
- `sudo systemctl reload apache2`

Common gotchas & fixes

- **403 Forbidden**

Usually permissions/ownership on `/var/www/drupalSite/web` or `files` directory.

Fix:

```
bash
```

```
sudo chown -R $USER:www-data /var/www/drupalSite
sudo find /var/www/drupalSite -type d -exec chmod 775 {} \;
sudo find /var/www/drupalSite -type f -exec chmod 664 {} \;
```

- **500 Error during install**

Check `/var/log/apache2/drupalSite_error.log` for the exact PHP error. Make sure you have required PHP extensions (you said you do) and that `DocumentRoot` is `/web`.

- **“Database not found / Access denied”**

Recheck the DB name/user/pass and host `localhost` vs `127.0.0.1`. If using socket vs TCP can matter; try `127.0.0.1` in Drush or installer advanced options.

- **Composer can't write in** `/var/www`

Make sure the directory is owned by your user (we set that). Otherwise run Composer as your user (not root).

If you prefer a different project path (e.g., `~/Sites/drupalSite`) or want HTTPS locally (self-signed cert), say the word and I'll add those steps.

↳ `chown -R $USER:www-data /var/www/drupalSite/web/sites/default/files`

while executing this command i am getting this error:
chown: changing ownership of
'/var/www/drupalSite/web/sites/default/files':
Operation not permitted

That's just Linux saying “you're not root.” `chown` needs elevated perms.

Try this exact sequence:

```
bash
```

```
# create the directory (as root, just to be safe)
sudo mkdir -p /var/www/drupalSite/web/sites/default/files

# give the whole project to you + www-data group
sudo chown -R $USER:www-data /var/www/drupalSite

# set sane perms (dirs 775, files 664)
sudo find /var/www/drupalSite -type d -exec chmod 775 {} \;
sudo find /var/www/drupalSite -type f -exec chmod 664 {} \;

# ensure the files dir specifically exists & is writable
sudo chown -R $USER:www-data /var/www/drupalSite/web/sites/default/files
```

If you still see “Operation not permitted,” quick checks:

1. Are you on WSL and working under `/mnt/c/...` ? Windows mounts don’t support `chown` . Keep the project under Linux FS (e.g., `/var/www` or `~/Sites`) instead.
2. Is the path a symlink to somewhere you don’t own?
3. What does this say?

```
bash
```

```
ls -ld /var/www/drupalSite /var/www/drupalSite/web/sites/default/files
```

Paste that output if it keeps failing and I’ll zero in.

↳ <http://www.drupalSite.com>

after going to the link in my browser instead of the installer another site from the internet is trying to load.

Gotcha—that means your hostname isn’t resolving to your local Apache, so the browser is hitting the real internet.

Do these quick checks/fixes in order:

1) Make sure `/etc/hosts` points to localhost

```
bash
```

```
sudo nano /etc/hosts
```

Add (or fix) these lines, then save:

```
makefile
```

```
127.0.0.1 www.drupalSite.com drupalSite.com
```

```
::1 www.drupalSite.com drupalSite.com
```

Then flush resolver cache:

```
bash
```

```
sudo resolvectl flush-caches
```

(If you use a VPN/proxy, disable it for this test.)

Sanity check:

```
bash
```

```
getent hosts www.drupalSite.com
```

```
ping -c1 www.drupalSite.com
```

Both should show `127.0.0.1`.

Also make sure you're using **http**, not https:

- Open <http://www.drupalSite.com>
- If you accidentally visited https earlier, the browser may remember it (HSTS). Try a private window or clear the site's HSTS entry.

2) Confirm your Apache vhost is active and matched

Enable & reload:

```
bash
```

```
sudo a2ensite drupalSite.conf
```

```
sudo systemctl reload apache2
```

List vhosts and confirm Apache sees `www.drupalSite.com`:


```
bash
```

```
sudo apache2ctl -S
```

You should see a line mapping `*:80 www.drupalSite.com (/etc/apache2/sites-enabled/drupalSite.conf:...)` .

Quick direct test (bypasses DNS and uses Host header):

```
bash
```

```
curl -I http://127.0.0.1 -H 'Host: www.drupalSite.com'
```

This should return your site (200/302) and **not** some external server.

3) Double-check the vhost file itself

`/etc/apache2/sites-available/drupalSite.conf` should look like:

```
apache
```

```
<VirtualHost *:80>
  ServerName www.drupalSite.com
  ServerAlias drupalSite.com
  DocumentRoot /var/www/drupalSite/web

  <Directory /var/www/drupalSite/web>
    AllowOverride All
    Require all granted
    Options FollowSymLinks
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/drupalSite_error.log
  CustomLog ${APACHE_LOG_DIR}/drupalSite_access.log combined
</VirtualHost>
```

Then:

```
bash
```

```
sudo systemctl reload apache2
```

4) If it still tries to load another site

- You're probably hitting **https://** (remote site has TLS, your local doesn't). Use **http://** explicitly.
- Or your `/etc/hosts` change didn't take effect—re-check step 1 and the outputs of `getent hosts` and `curl -I` with the Host header.
- Or a proxy/VPN/DoH client is intercepting DNS. Temporarily disable and try again.

If it's still messy, paste the outputs of:

```
bash

getent hosts www.drupalSite.com
sudo apache2ctl -S
curl -I http://127.0.0.1 -H 'Host: www.drupalSite.com'
```

and we'll pinpoint the exact culprit.

```
the content in my /etc/hosts is :
127.0.0.1 localhost
127.0.1.1 itminan-HP-Laptop-15s-du3xxx
127.0.0.1 www.drupalSite.com

# The following lines are desirable for IPv6 capable
hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# Added by Docker Desktop
# To allow the same kube context to work on the host
and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
```

Thanks for sharing `/etc/hosts` . That's almost there. Let's lock this down so your browser **must** hit your local Apache.

Do these in order:

1. Fix `/etc/hosts` (add the bare domain + IPv6 line)

```
bash

sudo nano /etc/hosts
```

Make sure these lines exist (together) — keep the others as-is:

```
makefile

127.0.0.1 www.drupalSite.com drupalSite.com
::1      www.drupalSite.com drupalSite.com
```

Save.

2. Flush resolver + sanity-check resolution

```
bash

sudo resolvectl flush-caches || true
getent hosts www.drupalSite.com
getent hosts drupalSite.com
```

Both should return `127.0.0.1`. If not, there's a typo/extra whitespace.

3. Ensure the Apache vhost matches that hostname and is enabled

Open the vhost file:

```
bash

sudo nano /etc/apache2/sites-available/drupalSite.conf
```

It should look like:

```
apache

<VirtualHost *:80>
  ServerName www.drupalSite.com
  ServerAlias drupalSite.com
  DocumentRoot /var/www/drupalSite/web

  <Directory /var/www/drupalSite/web>
    AllowOverride All
    Require all granted
    Options FollowSymLinks
```

```
</Directory>
```

```
ErrorLog ${APACHE_LOG_DIR}/drupalSite_error.log
```

```
CustomLog ${APACHE_LOG_DIR}/drupalSite_access.log combined
```

```
</VirtualHost>
```

Enable + reload:

```
bash
```

```
sudo a2ensite drupalSite.conf
```

```
sudo systemctl reload apache2
```

```
sudo apache2ctl -S
```

`apache2ctl -S` should list `*:80 www.drupalSite.com` pointing to your conf file.

4. Bypass DNS completely with a Host header test

```
bash
```

```
curl -I http://127.0.0.1 -H 'Host: www.drupalSite.com'
```

You should see a 200/302 from your local Apache (and **not** an external server).

5. Open the installer using http

Browsers sometimes auto-upgrade to HTTPS or remember a previous https visit.

Use:

```
arduino
```

```
http://www.drupalSite.com
```

If it still jumps to another site:

- Try a private window.
- Type the full scheme (`http://`) explicitly.
- Temporarily disable Secure DNS/DoH in your browser settings (very rare, but can bypass `/etc/hosts`).
- Verify again with step 4 (the `curl` test).

If anything still misbehaves, paste the outputs of:

```
nginx
```

```
getent hosts www.drupalSite.com
sudo apache2ctl -S
curl -I http://127.0.0.1 -H 'Host: www.drupalSite.com'
```

and we'll pinpoint the exact issue.

I have applied until the step 2 of your solution, My installer has started, but I got some errors, I will give you the errors, then you will help me solve them.

Perfect—send me the exact error messages/screenshots you're seeing on the installer screen.

While you grab those, these help me debug fast—run and paste outputs (or the relevant lines):

```
bash

# Apache vhost mapping + syntax
sudo apache2ctl -S

# Recent Apache errors for this vhost
sudo tail -n 200 /var/log/apache2/drupalSite_error.log || sudo tail -n 200
/var/log/apache2/error.log

# Check the web dir perms
ls -ld /var/www/drupalSite/web /var/www/drupalSite/web/sites/default
/var/www/drupalSite/web/sites/default/files
ls -l /var/www/drupalSite/web/sites/default | sed -n '1,120p'
```

And quick context:

- Are you visiting <http://www.drupalSite.com> (not https)?
- Which installer step shows the error (requirements check, DB config, or after DB import)?
- If it mentions specific PHP extensions or permissions, copy those lines verbatim.

Drop those here and I'll pinpoint fixes step-by-step.

Errors found

Settings file

The Settings file does not exist.

The Drupal installer requires that you create a `./sites/default/settings.php` as part of the installation process. Copy the `./sites/default/default.settings.php` file to `./sites/default/settings.php`. More details about installing Drupal are available in `INSTALL.txt`.

Warnings found

Clean URLs

Disabled

Your server is capable of using clean URLs, but it is not enabled. Using clean URLs gives an improved user experience and is recommended. Enable clean URLs

Nice—those are both straightforward.

1) Create `settings.php` and make it writable for install

Run these from your project root (`/var/www/drupalSite`):

```
bash
```

```
# create settings.php from the template
```

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

```
# (optional but recommended on modern Drupal)
```

```
# sudo cp web/sites/default/default.services.yml web/sites/default/services.yml
```

```
# give web server group access while you own the file
```

```
sudo chown $USER:www-data web/sites/default/settings.php
```

```
sudo chmod 664 web/sites/default/settings.php
```

```
# ensure the files directory exists and is writable
```

```
sudo mkdir -p web/sites/default/files
```

```
sudo chown -R $USER:www-data web/sites/default/files
```

```
sudo chmod -R 775 web/sites/default/files
```

Now reload the installer page; the “Settings file does not exist” error should disappear.

After installation completes, **tighten** the file back to read-only:

```
bash

sudo chmod 644 /var/www/drupalSite/web/sites/default/settings.php
```

2) Fix “Clean URLs: Disabled”

This means Apache isn’t applying Drupal’s `.htaccess` rewrites.

Do these:

```
bash

# make sure mod_rewrite is enabled
sudo a2enmod rewrite

# confirm your vhost allows .htaccess overrides
sudo nano /etc/apache2/sites-available/drupalSite.conf
```

Ensure your `<VirtualHost>` has this **exact** block (adjust path if you changed it):

```
apache

DocumentRoot /var/www/drupalSite/web

<Directory /var/www/drupalSite/web>
    AllowOverride All
    Require all granted
    Options FollowSymLinks
</Directory>
```

Then reload Apache:

```
bash

sudo systemctl reload apache2
```

Quick sanity checks:

```
bash
```

```
# .htaccess should exist here
```

```
ls -l /var/www/drupalSite/web/.htaccess
```

```
# Does a rewritten path respond? (should be 200/302, not 404)
```

```
curl -I http://www.drupalSite.com/user
```

If it's still "Disabled"

- Open `/var/www/drupalSite/web/.htaccess` and **uncomment** the line `# RewriteBase /` by changing it to:

```
nginx
```

```
RewriteBase /
```

Then `sudo systemctl reload apache2` and retry the installer.

- Make sure no other vhost is catching the request:

```
bash
```

```
sudo apache2ctl -S
```

You should see `*:80 www.drupalSite.com` pointing to `drupalSite.conf`.

- If you edited the vhost but nothing changes, verify you enabled the site:

```
bash
```

```
sudo a2ensite drupalSite.conf && sudo systemctl reload apache2
```

Try the installer again now. If anything still complains, paste:

```
bash
```

```
sudo apache2ctl -S
```

```
ls -l /var/www/drupalSite/web/.htaccess
```

```
curl -I http://www.drupalSite.com/user
```

and I'll zero in on what's left.

Can I configure the php version for the virtual hosts?
so that I don't have to change the php versions in
apache every time I run a different project?

Yes—you can run different PHP versions per virtual host. The clean way is to use **PHP-FPM** (one service per PHP version) and point each vhost to the FPM socket you want.

Here's a tidy, repeatable setup.

1) Install the PHP versions (FPM) you need

(You already have 8.3.24. If you also want, say, 8.2:)

```
bash

# Optional: add Ondrej PPA if you need older/newer versions
sudo add-apt-repository ppa:ondrej/php -y
sudo apt update

# PHP-FPM daemons
sudo apt install -y php8.3-fpm php8.2-fpm

# Any extensions you need, per version (examples)
sudo apt install -y php8.3-mysql php8.3-xml php8.3-gd php8.3-curl php8.3-mbstring
sudo apt install -y php8.2-mysql php8.2-xml php8.2-gd php8.2-curl php8.2-mbstring
```

Make sure both services run:

```
bash

systemctl status php8.3-fpm
systemctl status php8.2-fpm
```

2) Use Apache with proxy to FPM (disable mod_php)

```
bash

sudo a2dismod php8.3 # disable libapache2-mod-php if it's enabled
sudo a2enmod proxy proxy_fcgi setenvif rewrite
sudo a2dismod mpm_prefork || true
```

```
sudo a2enmod mpm_event
sudo systemctl restart apache2
```

3) Point each vhost to a specific PHP-FPM socket

Your Drupal vhost → PHP 8.3

```
/etc/apache2/sites-available/drupalSite.conf
```

```
apache

<VirtualHost *:80>
    ServerName www.drupalSite.com
    ServerAlias drupalSite.com
    DocumentRoot /var/www/drupalSite/web

    <Directory /var/www/drupalSite/web>
        AllowOverride All
        Require all granted
        Options FollowSymLinks
    </Directory>

    # Use PHP 8.3 for this host
    <FilesMatch "\.php$">
        SetHandler "proxy:unix:/run/php/php8.3-fpm.sock|fcgi://localhost/"
    </FilesMatch>

    ErrorLog ${APACHE_LOG_DIR}/drupalSite_error.log
    CustomLog ${APACHE_LOG_DIR}/drupalSite_access.log combined
</VirtualHost>
```

Example: another project → PHP 8.2

```
/etc/apache2/sites-available/otherSite.conf
```

```
apache

<VirtualHost *:80>
    ServerName other.local
    DocumentRoot /var/www/otherSite/public

    <Directory /var/www/otherSite/public>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

```
Options FollowSymLinks
</Directory>

# Use PHP 8.2 here
<FilesMatch "\.php$">
    SetHandler "proxy:unix:/run/php/php8.2-fpm.sock|fcgi://localhost/"
</FilesMatch>
</VirtualHost>
```

Enable and reload:

```
bash

sudo a2ensite drupalSite.conf otherSite.conf
sudo systemctl reload apache2
```

4) (Optional) Per-site PHP settings via separate FPM pool

If you want per-site limits (e.g., `upload_max_filesize`, `memory_limit`) **without** affecting other sites using the same PHP version, create a dedicated pool (new socket) for that site:

```
/etc/php/8.3/fpm/pool.d/drupalSite.conf
```

```
ini

[drupalSite]
user = www-data
group = www-data
listen = /run/php/php8.3-drupalSite.sock
pm = dynamic
pm.max_children = 10

; Per-site PHP admin values
php_admin_value[upload_max_filesize] = 32M
php_admin_value[post_max_size] = 32M
php_admin_value[memory_limit] = 512M
php_admin_value[max_execution_time] = 120
```

Enable it:

```
bash
```

```
sudo systemctl reload php8.3-fpm
```

Then point your vhost to this **site-specific** socket:

```
apache
```

```
<FilesMatch "\.php$">  
    SetHandler "proxy:unix:/run/php/php8.3-drupalSite.sock|fcgi://localhost/"  
</FilesMatch>
```

5) Quick test

Create a `phpinfo()` file in each site and confirm the version:

```
bash
```

```
echo "<?php phpinfo();" | sudo tee /var/www/drupalSite/web/info.php
```

Visit:

- `http://www.drupalSite.com/info.php` → should show **PHP 8.3**
- Your other site's `/info.php` → should show its configured version

Troubleshooting

- **503 Service Unavailable:** socket path wrong or FPM service not running.
- **Permission denied to socket:** make sure FPM's socket is owned by `www-data` (default on Ubuntu) and Apache runs as `www-data`. Check `listen.owner / listen.group` in the pool conf.
- **Another PHP handler still active:** disable any `phpX.conf` under `/etc/apache2/conf-enabled/` that globally sets a handler (e.g., `a2disconf` `php8.3-fpm` if it exists), then rely solely on the per-vhost `<FilesMatch>` blocks.

This way, you never switch PHP globally—each vhost always uses the version you wired it to.

so I can't do this with mod php?

Short answer: not really.

With **mod_php** (libapache2-mod-php), PHP is compiled/loaded **into Apache itself**, so only **one** PHP version can be active per Apache instance. That means you can switch versions globally (e.g., 8.2 ↔ 8.3), but you can't run different versions **per virtual host** in the same Apache.

Typical global switch with mod_php looks like:

```
bash

sudo a2dismod php8.3
sudo a2enmod php8.2
sudo systemctl restart apache2
```

...but that flips it for **every** vhost.

If you truly need per-vhost versions, your clean options are:

1. **PHP-FPM per vhost (recommended)**: point each vhost to its own FPM socket/version. Fast, stable, works with `mpm_event`, and easy to maintain.
2. **Separate Apache instances** (advanced): run two Apache daemons on different ports, each with a different mod_php version, and reverse-proxy from a front Apache/Nginx based on vhost. Works, but it's clunky and heavier.
3. **Containers** (nice isolation): each project in its own container with whatever PHP you want; front proxy routes by hostname.

So: per-vhost PHP versions with mod_php alone → **no**. Use **PHP-FPM** (or one of the workarounds above) if you need different versions side-by-side.