To install the required dependencies for a Laravel project, such as the one in your `home\_care` repository, you'll follow these steps. I'll provide a detailed description and the necessary commands to ensure your Laravel application is properly set up and running.

### Step 1: Access Your EC2 Instance

First, log in to your EC2 instance where you have cloned the `home\_care` repository.

```bash

ssh -i /path/to/your-key.pem ubuntu@your-ec2-instance-ip

```

### Step 2: Navigate to Your Project Directory

Once you're logged in, navigate to the directory where you cloned the `home\_care` repository.

```bash

cd /var/www/home\_care

```

### Step 3: Install PHP and Necessary Extensions

Laravel requires PHP and some specific extensions. Ensure that PHP is installed along with the necessary extensions.

```bash

sudo apt update

sudo apt install php php-mbstring php-xml php-bcmath php-json php-zip php-mysql unzip -y

```

### Step 4: Install Composer

Composer is a dependency manager for PHP, which Laravel uses to manage its packages.

1. Download and install Composer:

```bash

curl -sS https://getcomposer.org/installer | php

```

2. Move the Composer binary to a global location:

```bash

sudo mv composer.phar /usr/local/bin/composer

```

### Step 5: Install Laravel Dependencies

With Composer installed, you can now install the necessary dependencies for your Laravel project.

1. Run the following command in the `home\_care` project directory:

```bash

composer install

```

This will download and install all dependencies specified in the `composer.json` file.

### Step 6: Set File Permissions

Laravel needs specific permissions for certain directories like `storage` and `bootstrap/cache`.

1. Set the correct permissions:

```bash

sudo chown -R www-data:www-data /var/www/home\_care

sudo chmod -R 775 /var/www/home\_care/storage /var/www/home\_care/bootstrap/cache

```

### Step 7: Set Up the Environment File

The `.env` file contains configuration settings for your application. Since you've already generated an `APP\_KEY` locally, ensure it is correctly set on your EC2 instance.

1. If necessary, upload the `.env` file from your local machine to the server, or copy the `APP\_KEY` and other settings to the existing `.env` file on the server.

Example:

```bash

nano /var/www/home\_care/.env

```

Paste your settings, particularly the `APP\_KEY`.

### Step 8: Generate Application Key (if not done locally)

If you haven't already generated the application key on your local machine, do it now:

```bash

php artisan key:generate

```

This command sets the `APP\_KEY` in your `.env` file.

### Step 9: Run Database Migrations

If your application requires a database, run the migrations to set up the database schema.

```bash

php artisan migrate

```

### Step 10: Set Up Your Web Server (Apache/Nginx)

Assuming you are using Apache or Nginx, you need to configure your web server to serve the Laravel application.

For Apache, ensure the following configuration:

```bash

sudo nano /etc/apache2/sites-available/000-default.conf

```

Update the `DocumentRoot` to point to the `public` directory of your Laravel project:

```apache

DocumentRoot /var/www/home\_care/public

```

Enable mod\_rewrite and restart Apache:

```bash

sudo a2enmod rewrite

sudo systemctl restart apache2

```

For Nginx, you would create a configuration file pointing to the `public` directory and restart Nginx.

### Step 11: Access Your Application

Finally, ensure your application is running by accessing it via the public IP address or domain name of your EC2 instance.

```bash

http://your-ec2-instance-ip

```

### Summary of Commands

Here’s a quick summary of the key commands:

1. \*\*Update system and install PHP:\*\*

```bash

sudo apt update

sudo apt install php php-mbstring php-xml php-bcmath php-json php-zip php-mysql unzip -y

```

2. \*\*Install Composer:\*\*

```bash

curl -sS https://getcomposer.org/installer | php

sudo mv composer.phar /usr/local/bin/composer

```

3. \*\*Install Laravel dependencies:\*\*

```bash

composer install

```

4. \*\*Set permissions:\*\*

```bash

sudo chown -R www-data:www-data /var/www/home\_care

sudo chmod -R 775 /var/www/home\_care/storage /var/www/home\_care/bootstrap/cache

```

5. \*\*Generate app key:\*\*

```bash

php artisan key:generate

```

6. \*\*Run migrations:\*\*

```bash

php artisan migrate

```

With these steps, your Laravel project should be up and running on your EC2 instance.