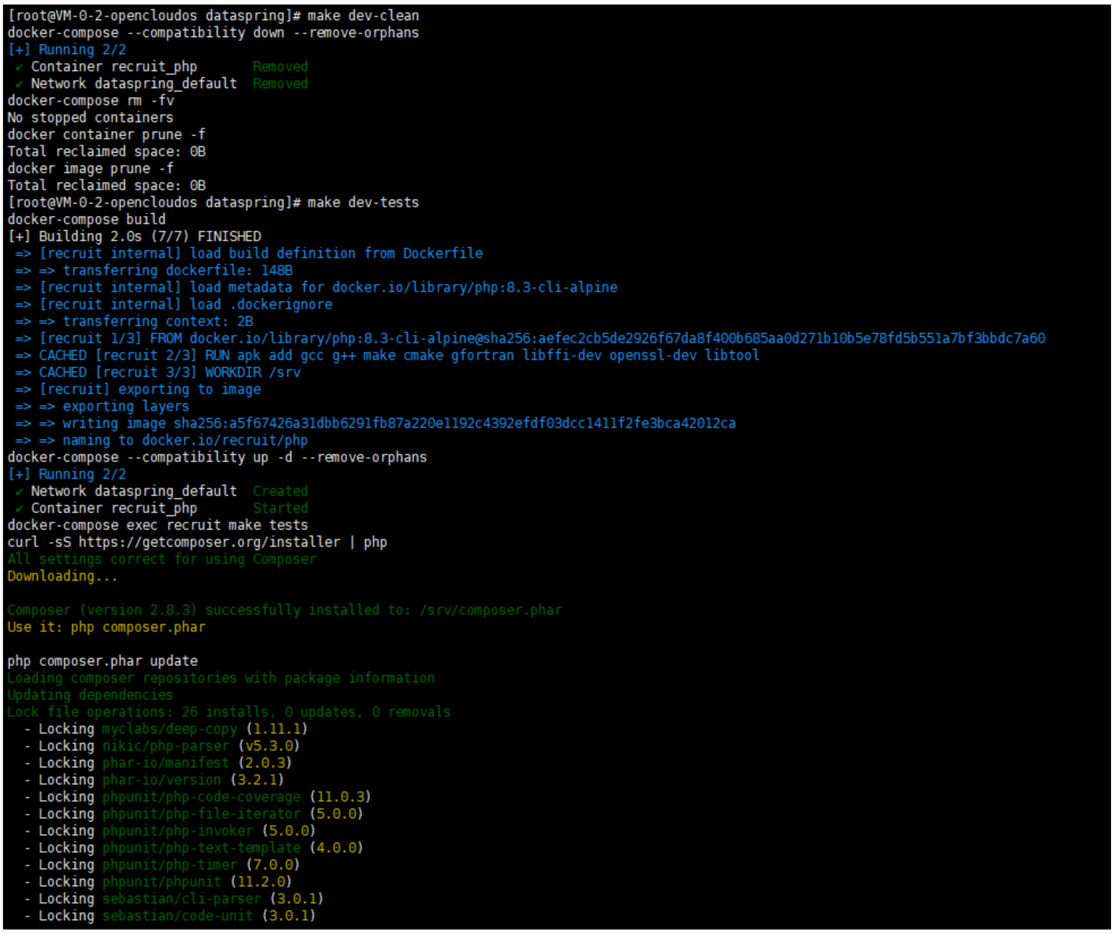
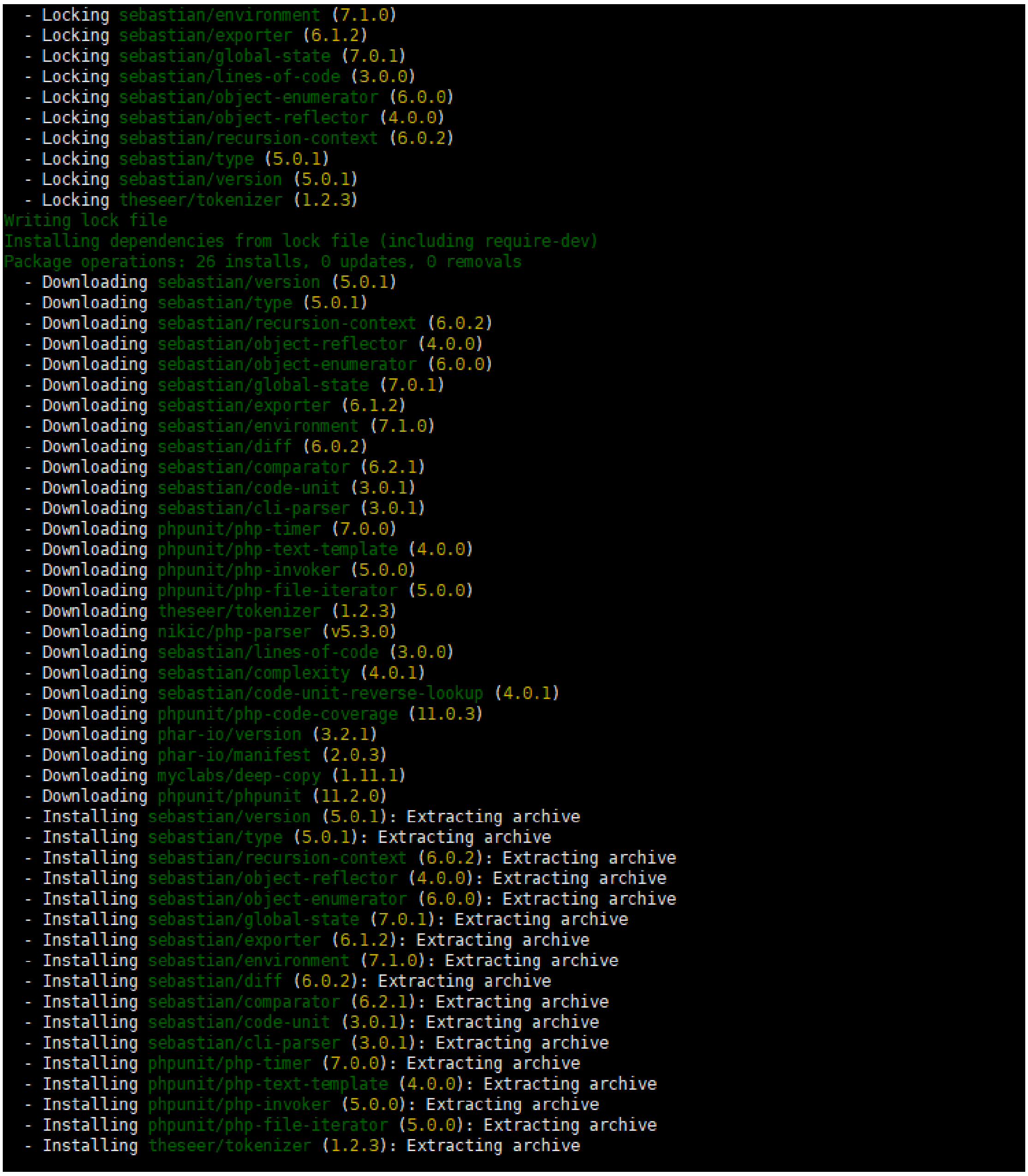
dataSpring 面试题目补充
题目来源：
https://github.com/researchpanelasia/ds-recruit/

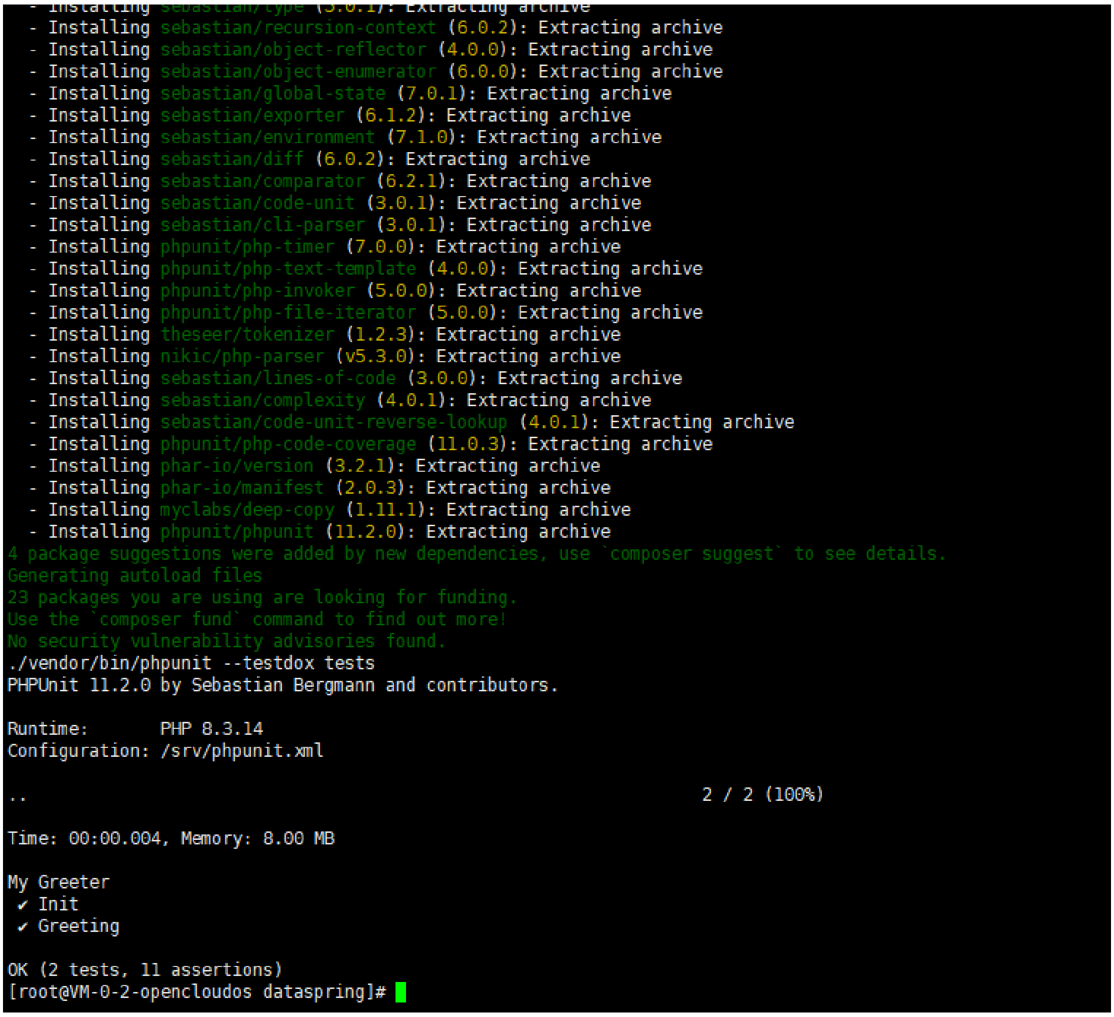
1. **代码实例**：https://github.com/richard-wzr/for-questionnaire-php
2. **Docker 环境进行单元测试的过程记录**
    a) 使用一台海外 Linux 服务器（为了 docker 网络使用正常），我使用了 centos stream9
    b) 在服务器上安装 docker 和 docker-compose
    c) 将项目文件通过 ftp（我用的是 sftp）上传到服务器
    d) 执行 make dev-tests 命令



```
[root@VM-0-2-opencloudos dataspring]# make dev-clean
docker-compose --compatibility down --remove-orphans
[+] Running 2/2
 ✓ Container recruit_php       Removed
 ✓ Network dataspring_default  Removed
docker-compose rm -fv
No stopped containers
docker container prune -f
Total reclaimed space: 0B
docker image prune -f
Total reclaimed space: 0B
[root@VM-0-2-opencloudos dataspring]# make dev-tests
docker-compose build
[+] Building 2.0s (7/7) FINISHED
 => [recruit internal] load build definition from Dockerfile
 => => transferring dockerfile: 148B
 => [recruit internal] load metadata for docker.io/library/php:8.3-cli-alpine
 => [recruit internal] load .dockerignore
 => => transferring context: 2B
 => [recruit 1/3] FROM docker.io/library/php:8.3-cli-alpine@sha256:aefec2cb5de2926f67da8f400b685aa0d271b10b5e78fd5b551a7bf3bbdc7a60
 => CACHED [recruit 2/3] RUN apk add gcc g++ make cmake gfortran libffi-dev openssl-dev libtool
 => CACHED [recruit 3/3] WORKDIR /srv
 => [recruit] exporting to image
 => => exporting layers
 => => writing image sha256:a5f67426a31dbb6291fb87a220e1192c4392efdf03dcc1411f2fe3bca42012ca
 => => naming to docker.io/recruit/php
docker-compose --compatibility up -d --remove-orphans
[+] Running 2/2
 ✓ Network dataspring_default  Created
 ✓ Container recruit_php       Started
docker-compose exec recruit make tests
curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.8.3) successfully installed to: /srv/composer.phar
Use it: php composer.phar

php composer.phar update
Loading composer repositories with package information
Updating dependencies
Lock file operations: 26 installs, 0 updates, 0 removals
  - Locking myclabs/deep-copy (1.11.1)
  - Locking nikic/php-parser (v5.3.0)
  - Locking phar-io/manifest (2.0.3)
  - Locking phar-io/version (3.2.1)
  - Locking phpunit/php-code-coverage (11.0.3)
  - Locking phpunit/php-file-iterator (5.0.0)
  - Locking phpunit/php-invoker (5.0.0)
  - Locking phpunit/php-text-template (4.0.0)
  - Locking phpunit/php-timer (7.0.0)
  - Locking phpunit/phpunit (11.2.0)
  - Locking sebastian/cli-parser (3.0.1)
  - Locking sebastian/code-unit (3.0.1)
```



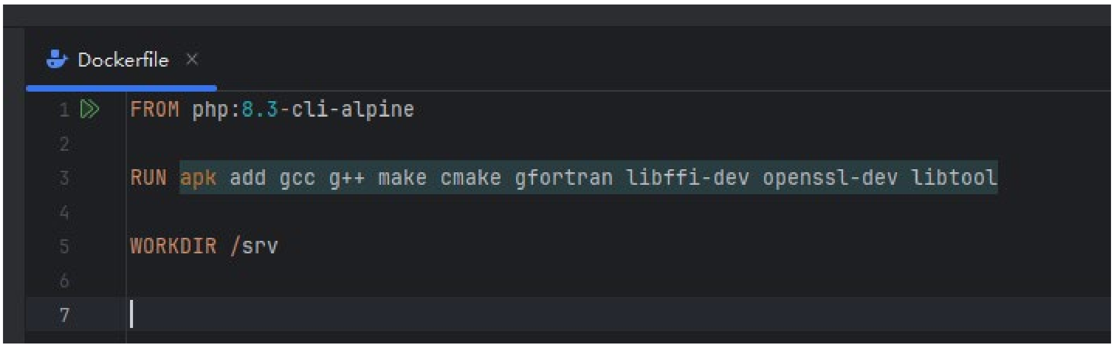
```
  - Locking sebastian/environment (7.1.0)
  - Locking sebastian/exporter (6.1.2)
  - Locking sebastian/global-state (7.0.1)
  - Locking sebastian/lines-of-code (3.0.0)
  - Locking sebastian/object-enumerator (6.0.0)
  - Locking sebastian/object-reflector (4.0.0)
  - Locking sebastian/recursion-context (6.0.2)
  - Locking sebastian/type (5.0.1)
  - Locking sebastian/version (5.0.1)
  - Locking theseer/tokenizer (1.2.3)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 26 installs, 0 updates, 0 removals
  - Downloading sebastian/version (5.0.1)
  - Downloading sebastian/type (5.0.1)
  - Downloading sebastian/recursion-context (6.0.2)
  - Downloading sebastian/object-reflector (4.0.0)
  - Downloading sebastian/object-enumerator (6.0.0)
  - Downloading sebastian/global-state (7.0.1)
  - Downloading sebastian/exporter (6.1.2)
  - Downloading sebastian/environment (7.1.0)
  - Downloading sebastian/diff (6.0.2)
  - Downloading sebastian/comparator (6.2.1)
  - Downloading sebastian/code-unit (3.0.1)
  - Downloading sebastian/cli-parser (3.0.1)
  - Downloading phpunit/php-timer (7.0.0)
  - Downloading phpunit/php-text-template (4.0.0)
  - Downloading phpunit/php-invoker (5.0.0)
  - Downloading phpunit/php-file-iterator (5.0.0)
  - Downloading theseer/tokenizer (1.2.3)
  - Downloading nikic/php-parser (v5.3.0)
  - Downloading sebastian/lines-of-code (3.0.0)
  - Downloading sebastian/complexity (4.0.1)
  - Downloading sebastian/code-unit-reverse-lookup (4.0.1)
  - Downloading phpunit/php-code-coverage (11.0.3)
  - Downloading phar-io/version (3.2.1)
  - Downloading phar-io/manifest (2.0.3)
  - Downloading myclabs/deep-copy (1.11.1)
  - Downloading phpunit/phpunit (11.2.0)
  - Installing sebastian/version (5.0.1): Extracting archive
  - Installing sebastian/type (5.0.1): Extracting archive
  - Installing sebastian/recursion-context (6.0.2): Extracting archive
  - Installing sebastian/object-reflector (4.0.0): Extracting archive
  - Installing sebastian/object-enumerator (6.0.0): Extracting archive
  - Installing sebastian/global-state (7.0.1): Extracting archive
  - Installing sebastian/exporter (6.1.2): Extracting archive
  - Installing sebastian/environment (7.1.0): Extracting archive
  - Installing sebastian/diff (6.0.2): Extracting archive
  - Installing sebastian/comparator (6.2.1): Extracting archive
  - Installing sebastian/code-unit (3.0.1): Extracting archive
  - Installing sebastian/cli-parser (3.0.1): Extracting archive
  - Installing phpunit/php-timer (7.0.0): Extracting archive
  - Installing phpunit/php-text-template (4.0.0): Extracting archive
  - Installing phpunit/php-invoker (5.0.0): Extracting archive
  - Installing phpunit/php-file-iterator (5.0.0): Extracting archive
  - Installing theseer/tokenizer (1.2.3): Extracting archive
```



```
  - Installing sebastian/type (5.0.1): Extracting archive
  - Installing sebastian/recursion-context (6.0.2): Extracting archive
  - Installing sebastian/object-reflector (4.0.0): Extracting archive
  - Installing sebastian/object-enumerator (6.0.0): Extracting archive
  - Installing sebastian/global-state (7.0.1): Extracting archive
  - Installing sebastian/exporter (6.1.2): Extracting archive
  - Installing sebastian/environment (7.1.0): Extracting archive
  - Installing sebastian/diff (6.0.2): Extracting archive
  - Installing sebastian/comparator (6.2.1): Extracting archive
  - Installing sebastian/code-unit (3.0.1): Extracting archive
  - Installing sebastian/cli-parser (3.0.1): Extracting archive
  - Installing phpunit/php-timer (7.0.0): Extracting archive
  - Installing phpunit/php-text-template (4.0.0): Extracting archive
  - Installing phpunit/php-invoker (5.0.0): Extracting archive
  - Installing phpunit/php-file-iterator (5.0.0): Extracting archive
  - Installing theseer/tokenizer (1.2.3): Extracting archive
  - Installing nikic/php-parser (v5.3.0): Extracting archive
  - Installing sebastian/lines-of-code (3.0.0): Extracting archive
  - Installing sebastian/complexity (4.0.1): Extracting archive
  - Installing sebastian/code-unit-reverse-lookup (4.0.1): Extracting archive
  - Installing phpunit/php-code-coverage (11.0.3): Extracting archive
  - Installing phar-io/version (3.2.1): Extracting archive
  - Installing phar-io/manifest (2.0.3): Extracting archive
  - Installing myclabs/deep-copy (1.11.1): Extracting archive
  - Installing phpunit/phpunit (11.2.0): Extracting archive
4 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files
23 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.
./vendor/bin/phpunit --testdox tests
PHPUnit 11.2.0 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.3.14
Configuration: /srv/phpunit.xml

..                                                                  2 / 2 (100%)

Time: 00:00.004, Memory: 8.00 MB

My Greeter
 ✔ Init
 ✔ Greeting

OK (2 tests, 11 assertions)
[root@VM-0-2-opencloudos dataspring]# 
```

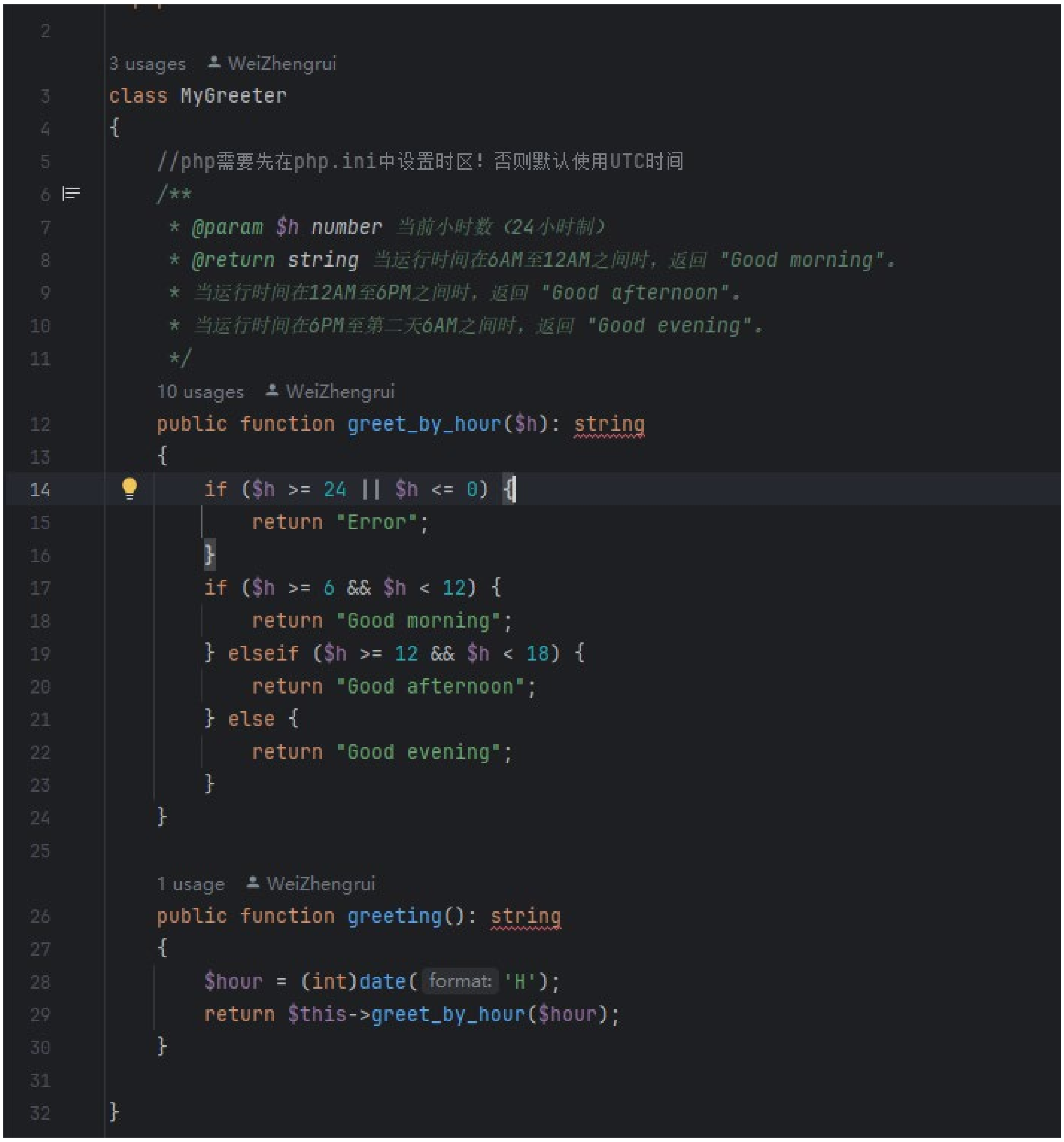
3. 改进点/修改点说明

a) Dockerfile 需要改进，原始容器中不存在 make 命令，需要在容器创建的过程中安装 make 相关工具

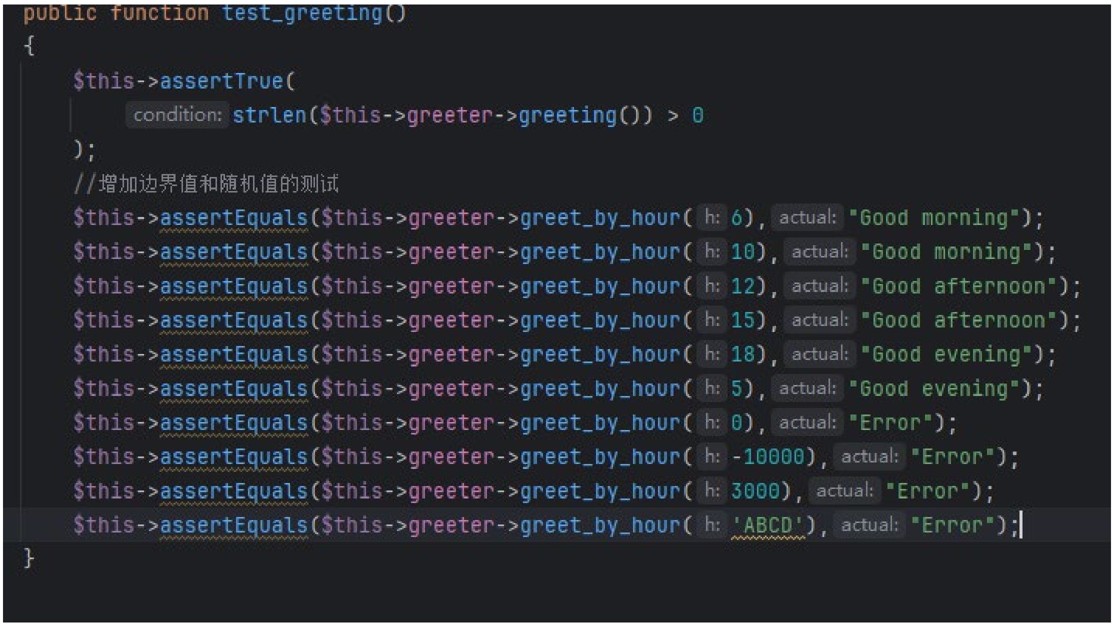


```
FROM php:8.3-cli-alpine

RUN apk add gcc g++ make cmake gfortran libffi-dev openssl-dev libtool

WORKDIR /srv

```

b) 单元测试需要进行改进，当前单元测试仅仅对结果的长度进行了校验，覆盖不完全，greeting 方法是根据不同系统时间返回不同数值的，将 greeting 方法的“时间”单独抽出来作为参数。即将 greeting 方法拆分成 greeting()和 greet_by_hour($h)，对 greet_by_hour($h)撰写单元测试。单元测试内容包括：**边界值测试、非法值测试、随机值测试**等



```
3 usages  WeiZhengrui
class MyGreeter
{
    //php需要先在php.ini中设置时区！否则默认使用UTC时间
    /**
     * @param $h number 当前小时数（24小时制）
     * @return string 当运行时间在6AM至12AM之间时，返回 "Good morning".
     * 当运行时间在12AM至6PM之间时，返回 "Good afternoon".
     * 当运行时间在6PM至第二天6AM之间时，返回 "Good evening".
     */
    10 usages  WeiZhengrui
    public function greet_by_hour($h): string
    {
        if ($h >= 24 || $h <= 0) {
            return "Error";
        }
        if ($h >= 6 && $h < 12) {
            return "Good morning";
        } elseif ($h >= 12 && $h < 18) {
            return "Good afternoon";
        } else {
            return "Good evening";
        }
    }

    1 usage  WeiZhengrui
    public function greeting(): string
    {
        $hour = (int)date( format: 'H');
        return $this->greet_by_hour($hour);
    }

}
```



```
public function test_greeting()
{
    $this->assertTrue(
        condition: strlen($this->greeter->greeting()) > 0
    );
    //增加边界值和随机值的测试
    $this->assertEquals($this->greeter->greet_by_hour( h: 6), actual: "Good morning");
    $this->assertEquals($this->greeter->greet_by_hour( h: 10), actual: "Good morning");
    $this->assertEquals($this->greeter->greet_by_hour( h: 12), actual: "Good afternoon");
    $this->assertEquals($this->greeter->greet_by_hour( h: 15), actual: "Good afternoon");
    $this->assertEquals($this->greeter->greet_by_hour( h: 18), actual: "Good evening");
    $this->assertEquals($this->greeter->greet_by_hour( h: 5), actual: "Good evening");
    $this->assertEquals($this->greeter->greet_by_hour( h: 0), actual: "Error");
    $this->assertEquals($this->greeter->greet_by_hour( h: -10000), actual: "Error");
    $this->assertEquals($this->greeter->greet_by_hour( h: 3000), actual: "Error");
    $this->assertEquals($this->greeter->greet_by_hour( h: 'ABCD'), actual: "Error");
}
```