

# Running PHPUnit tests over SSH on a remote server with PhpStorm



## Redirection Notice

This page will redirect to <https://www.jetbrains.com/help/phpstorm/enabling-php-unit-support.html> in about 2 seconds.

[Tweet](#)

In this tutorial, we'll see how we can run PHPUnit on a remote server, Vagrant or Docker machine using PhpStorm.

By using remote PHP interpreter we can get a lot of flexibility in how we develop our PHP applications. For instance, we can have our complete development environment on a Docker machine while only having the PhpStorm IDE on our local machine. We can run and debug our application on the Vagrant or Docker machine, and even run PHPUnit there. Or on a remote server, that's possible too. Let's see how we can get this all set up.

- Prerequisites
  - PHPUnit
  - Configure a Remote PHP Interpreter
- Setting up a PHPUnit Run/Debug configuration
  - 1. Configuring PHPUnit By Remote Interpreter
  - 2. Creating a PHPUnit Run/Debug configuration
  - 3. Running PHPUnit tests on a remote machine
- Troubleshooting Remote PHP Interpreters

## Prerequisites

To make use of PHPUnit on a remote server, there are some prerequisites that should be in place.

## PHPUnit

Before going through this tutorial, make yourself familiar with [Testing PHP Applications](#). Running PHPUnit tests on a remote machine with PhpStorm does not require a special setup of PHPUnit, but it is important to get acquainted with how PHPUnit works in PhpStorm.



If you wish to run PHPUnit on a remote machine with PhpStorm and will not be using [Composer to load PHPUnit](#), make sure that PHPUnit is available on the remote machine by [downloading the latest phpunit.phar](#) onto the machine.

PHPUnit in Docker can be used by simply choosing the right docker image: `phpunit/phpunit`.

## Configure a Remote PHP Interpreter

To run PHPUnit tests on a remote server, we need to setup a remote PHP interpreter as described in the [Working with Remote PHP Interpreters in PhpStorm](#) tutorial.

Optionally, we can configure the remote PHP interpreter to make use of Vagrant or Docker. In order for this to work, refer to [Vagrant Support in PhpStorm](#) or [Docker Remote Interpreters](#) to get the IDE ready.

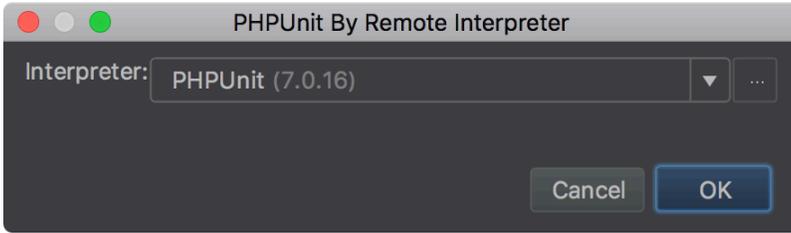
## Setting up a PHPUnit Run/Debug configuration

Since PhpStorm treats remote PHP interpreters in the same way it treats local PHP interpreters, setting up a PHPUnit Run/Debug configuration is similar to how it's done when testing PHP applications on a local machine. There is just one additional step required: configuring the way PHPUnit is loaded.

### 1. Configuring PHPUnit By Remote Interpreter

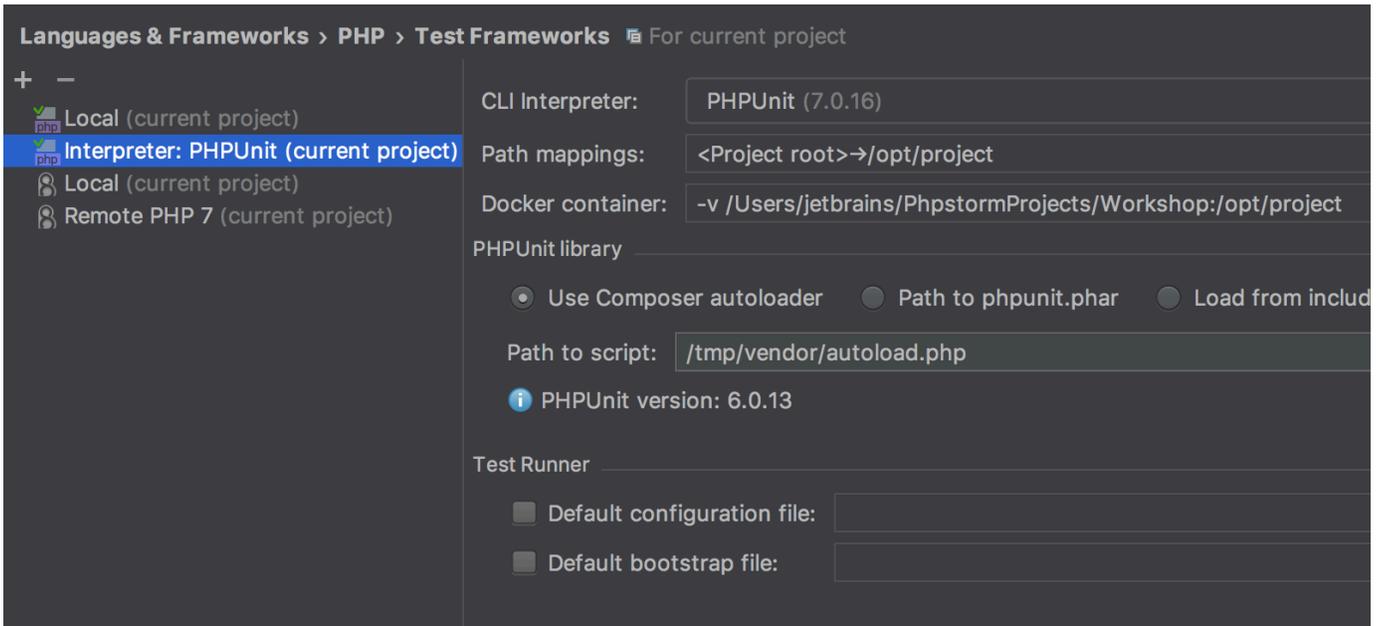
In File | Settings (PhpStorm | Preferences on Mac) | Languages & Frameworks | PHP | Test Frameworks, click the + button and add a new PHPUnit By Remote Interpreter configuration type. This allows us to configure PHPUnit for our remote PHP

interpreter.



Once added, we can specify how PHPUnit will be loaded:

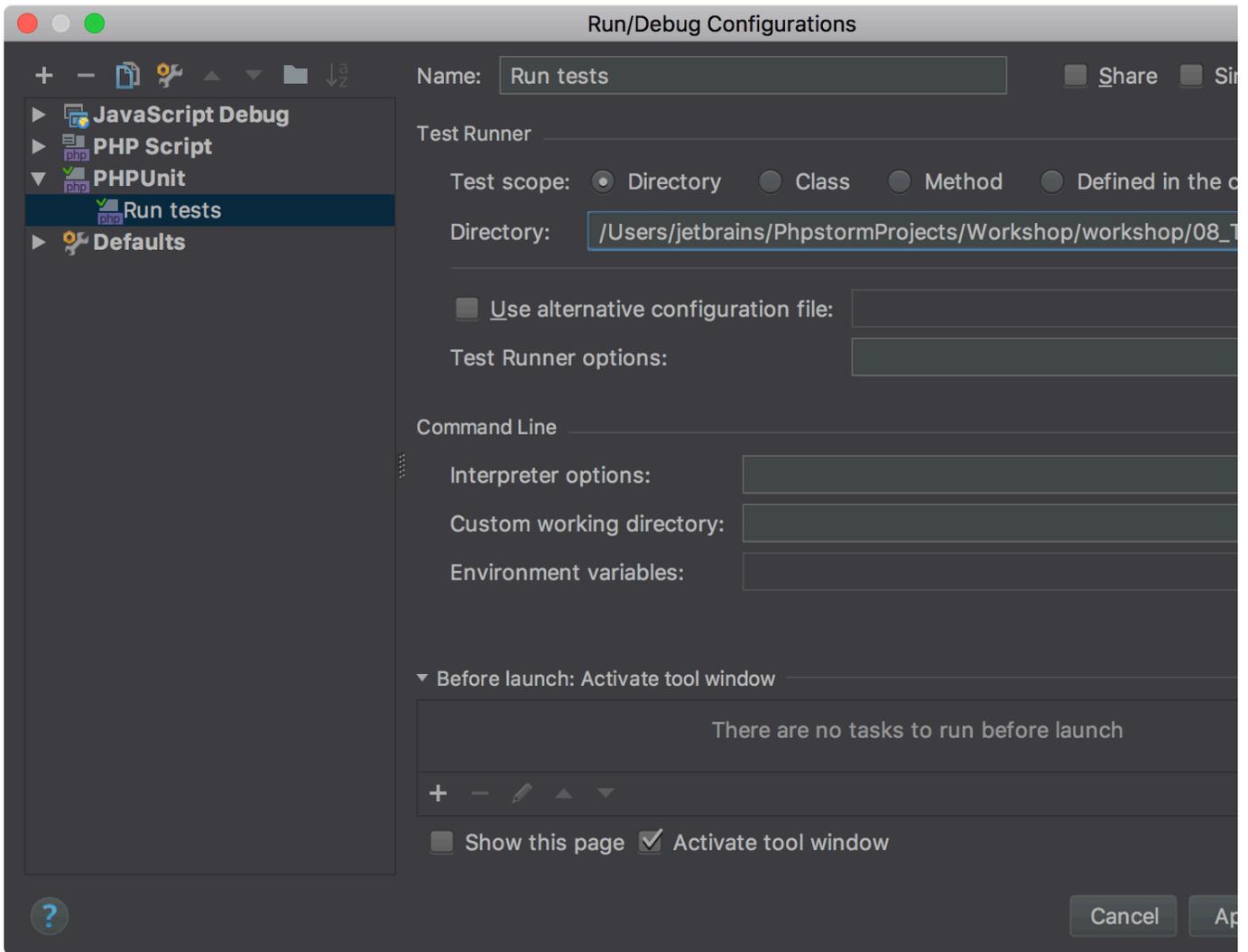
- Use Composer autoloader and specify the path to autoload.php in a vendor directory, e.g. set it to vagrant/vendor/autoload.php referring to the mount point of the shared project folder.
- Provide the path to phpunit.phar which should be present on the remote machine
- When installed through PEAR (obsolete for PHPUnit < 5), we can load it from the include path. Note that PHPUnit should be installed using PEAR on the remote server (or Vagrant/Docker machine).



Note that we can also specify the path to a PHPUnit configuration file and/or bootstrap file.

## 2. Creating a PHPUnit Run/Debug configuration

We can create a Run/Debug configuration for PHPUnit. Note that this should not be the PHPUnit by HTTP configuration (deprecated in PhpStorm 2017.1): a regular PHPUnit configuration is what we will need. As with local PHPUnit, we have to select which tests we want to run (by directory, class, method, or a configuration file) and can provide additional PHPUnit and PHP interpreter settings.



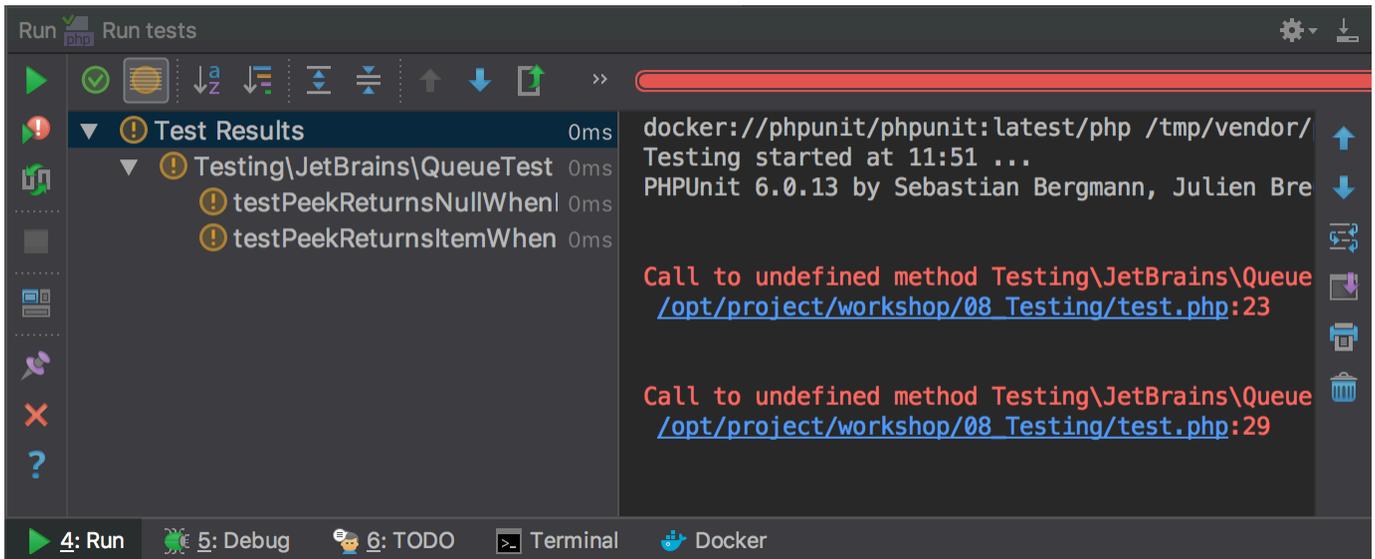
✓ The fact that running PHPUnit tests on a remote machine with PhpStorm does not require a special Run/Debug configuration means that we can easily switch where unit tests are run. By switching the project's PHP interpreter through File | Settings (PhpStorm > Preferences on Mac) | Languages & Frameworks | PHP , we can run PHPUnit tests either local or remote without having to reconfigure anything else.

### 3. Running PHPUnit tests on a remote machine

Once everything is configured, this is the only step we want to repeat during development: running PHPUnit tests. Start the Run configuration we have just created and see that PhpStorm connects to the remote server, Vagrant or Docker machine.

⚠ The first time a connection is made to the remote server, Vagrant or Docker machine, PhpStorm may ask to confirm the SSH thumbprint.

Test results will be displayed in the tests tool window, just as if we were running them locally. When the remote PHP interpreter is [configured for debugging using XDebug](#) or [Zend Debugger](#), our tests can be debugged and/or profiled as well.



## Troubleshooting Remote PHP Interpreters

Here are some tips for making sure the configuration is correct:

- Under the File | Settings (PhpStorm > Preferences on Mac) | Languages & Frameworks | PHP , make sure that the PHP language level is set and the remote PHP interpreter is selected.
- Make sure to [configure PHPUnit "by remote interpreter"](#). PhpStorm will need the remote path to the PHPUnit configuration file and/or autoloader.
- Verify the PHPUnit Run/Debug Configuration. Make sure it is a normal PHPUnit configuration (not PHPUnit by HTTP).

See [Working with Remote PHP Interpreters in PhpStorm](#) for additional troubleshooting tips when working with remote PHP interpreters.

[Tweet](#)