Working with Remote PHP Interpreters in PhpStorm

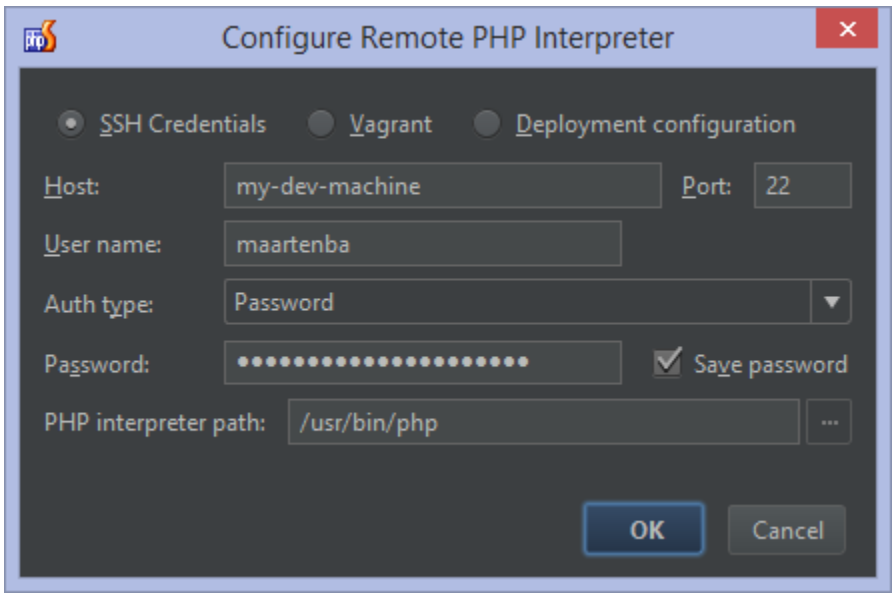
Using a remote PHP interpreter instead of a local one lets us run our application and PHP-based tools on a production-like environment, be it the real production server or a virtualized one that uses a tool like Vagrant. Speaking of Vagrant, with remote PHP interpreters we can install only PhpStorm on our development machine, and run, debug and unit test our application on the Vagrant machine. In this tutorial, we'll see how we can work with remote PHP interpreters in PhpStorm.

1. Registering a new PHP interpreter

A remote interpreter can be configured in a similar way as a local one. We can specify the PHP interpreter we want to use in File | Settings (or PhpStorm | Preferences for Mac) | Languages & Frameworks | PHP. This can be a local PHP installation or a remote PHP executable. Clicking ... next to the selected interpreter will open a list of registered PHP interpreters. Click the green + to add a new Remote....

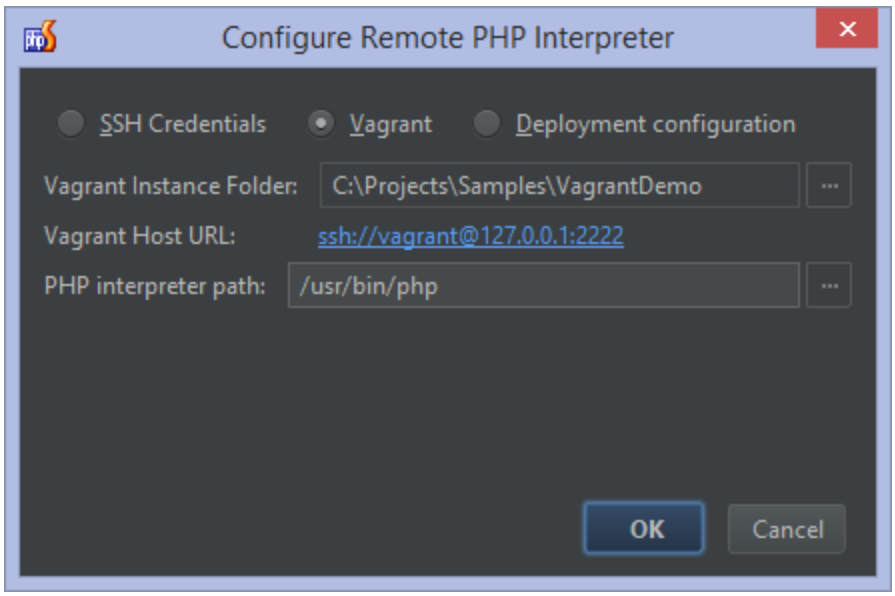
Adding a remote PHP interpreter over SSH...

When using a remote server, enter the SSH connection details to the host. Note that password authentication is supported, as is an OpenSSH or Putty key pair.



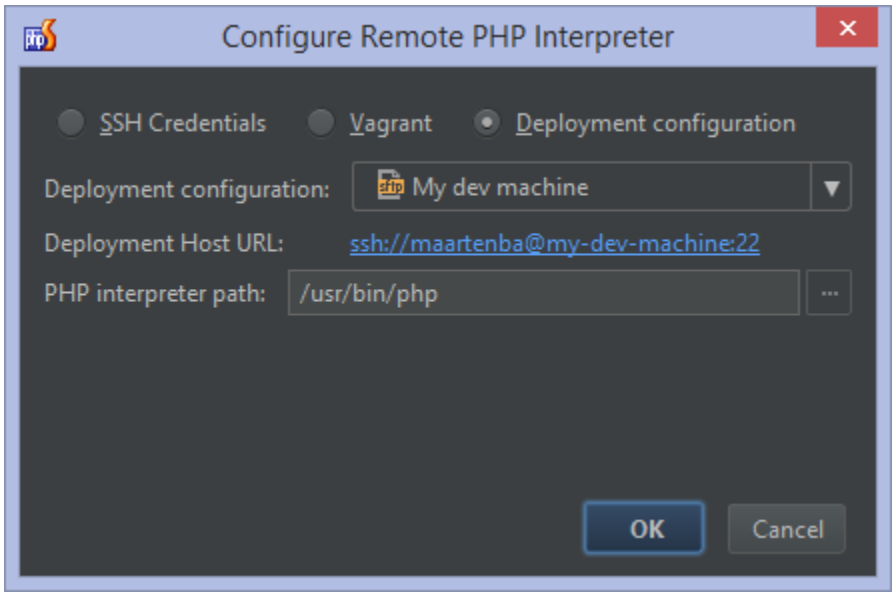
Adding a remote PHP interpreter from Vagrant...

When using Vagrant support in PhpStorm, we can select the Vagrant option to automatically populate connection details. In most cases, PhpStorm can also figure out the path to the PHP interpreter on the Vagrant machine. If not, we can adjust it manually.



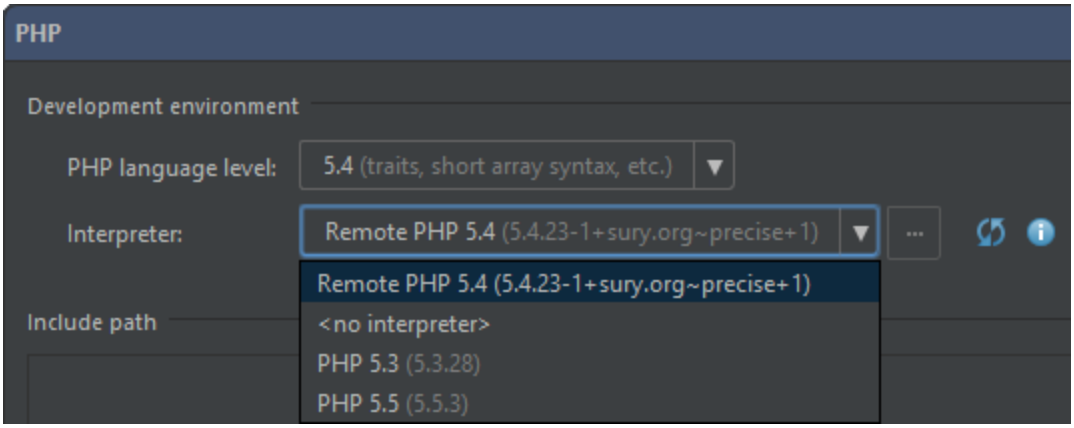
Adding a remote PHP interpreter from deployment configuration...

When one or more SFTP deployment configurations are present, for example when using Deployments in PhpStorm, the IDE will offer a third option: we can select the Deployment configuration option to use an existing configuration to connect to a remote PHP interpreter.

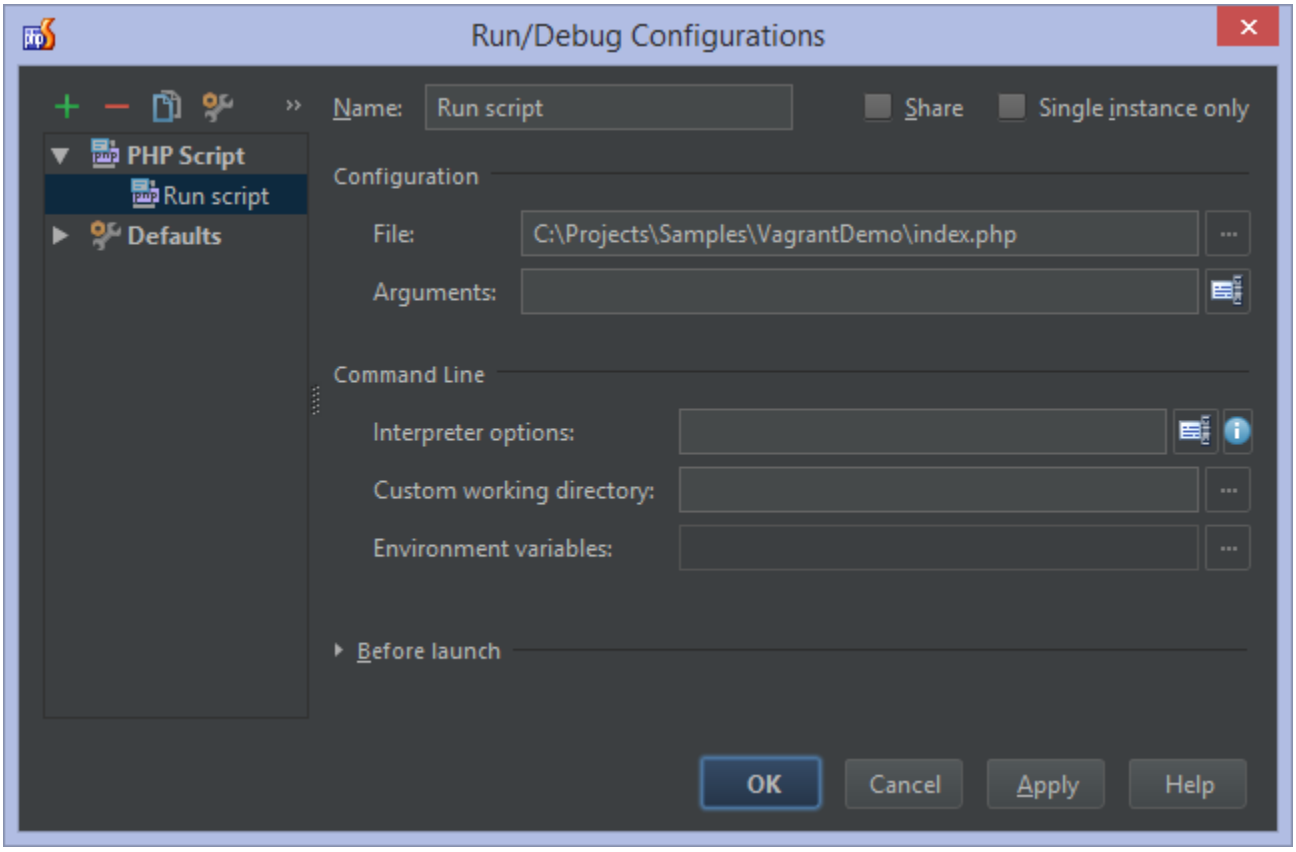


2. Selecting the remote PHP interpreter for the current PhpStorm project

Once a remote PHP interpreter is registered and configured, we can set up our project to use it during development. Simply select it in the dropdown list in File | Settings (or PhpStorm | Preferences for Mac) | Languages & Frameworks | PHP.



Once configured, all our Run/Debug configurations will make use of the configured PHP interpreter. Since we just configured a remote PHP interpreter as the one that should be used for our project, the following run configuration will always run on the remote server or Vagrant machine. All paths in Run/Debug Configuration should be local, they'll be automatically mapped to remote server local paths by the IDE.



3. Using the remote PHP interpreter

Once configured, we can start using our remote PHP interpreter, for example for unit testing. Check the tutorial Running PHPUnit tests over SSH on a remote server with PhpStorm for more information.

Debugging and Remote PHP Interpreters

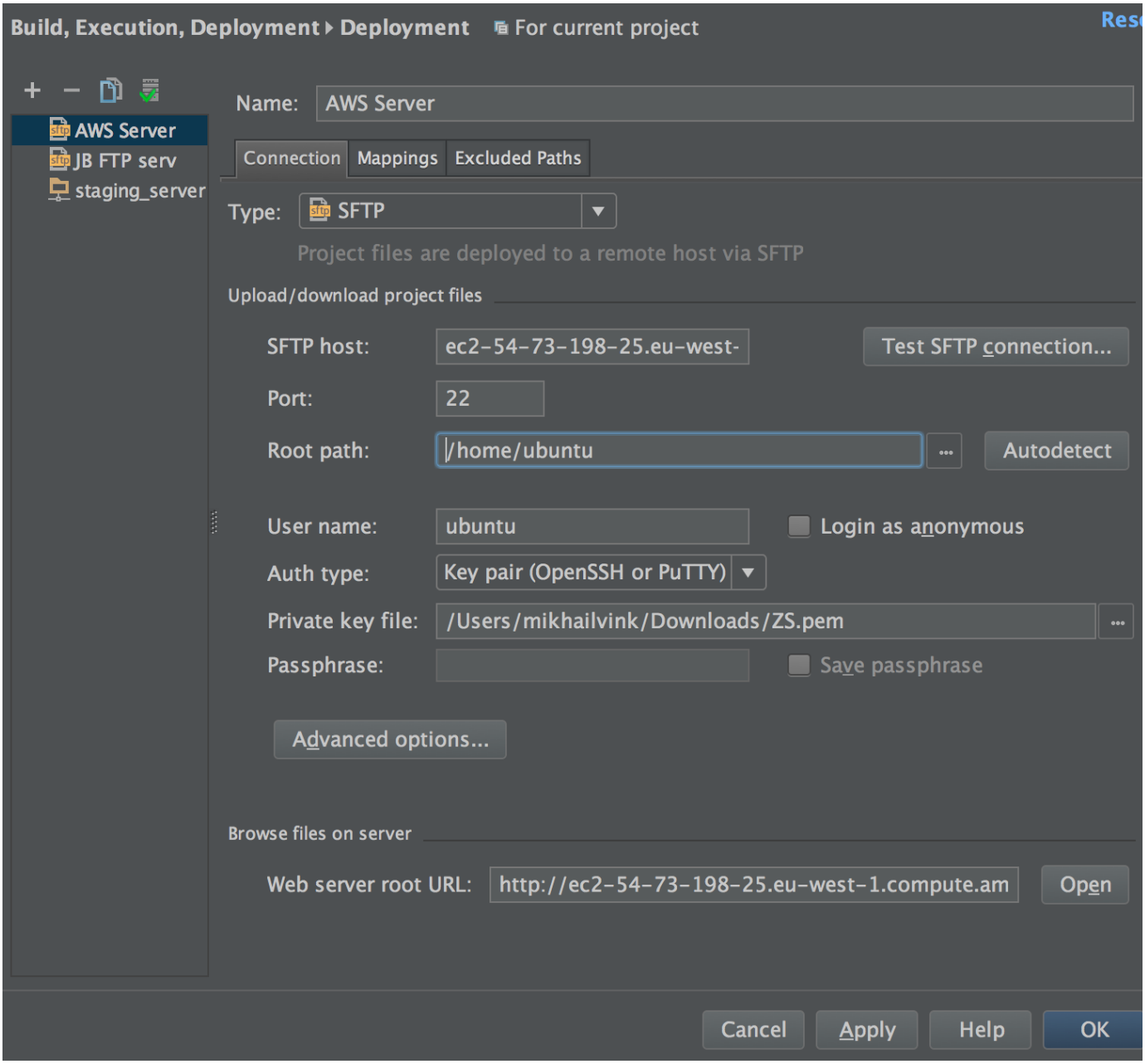
Remote debugging of PHP CLI scripts is possible when the remote PHP interpreter and SFTP deployment server with proper mappings are configured. Mappings can be also fetched from valid .Vagrantfile.

In order to take advantage of debugging with remote PHP interpreters, follow this workflow:

1. Make sure that remote PHP interpreter is configured and selected for current project in File | Settings (or PhpStorm |

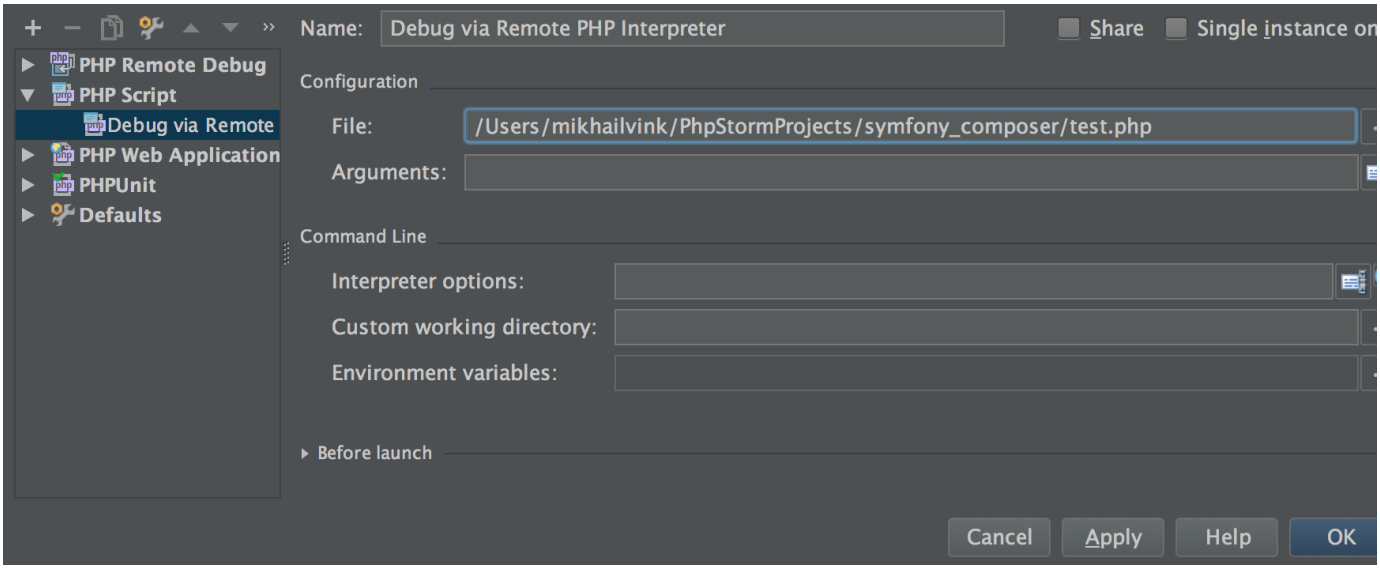
Preferences for Mac) | Languages & Frameworks | PHP, and Xdebug extension is installed (read more on Xdebug installation).

2. Configure SFTP deployment server that matches Remote Interpreter settings in File | Settings (or PhpStorm | Preferences for Mac) | Build, Execution, Deployment | Deployment with proper mappings (if you don't have it configured yet).



3. Put the breakpoint and start the debugging session.

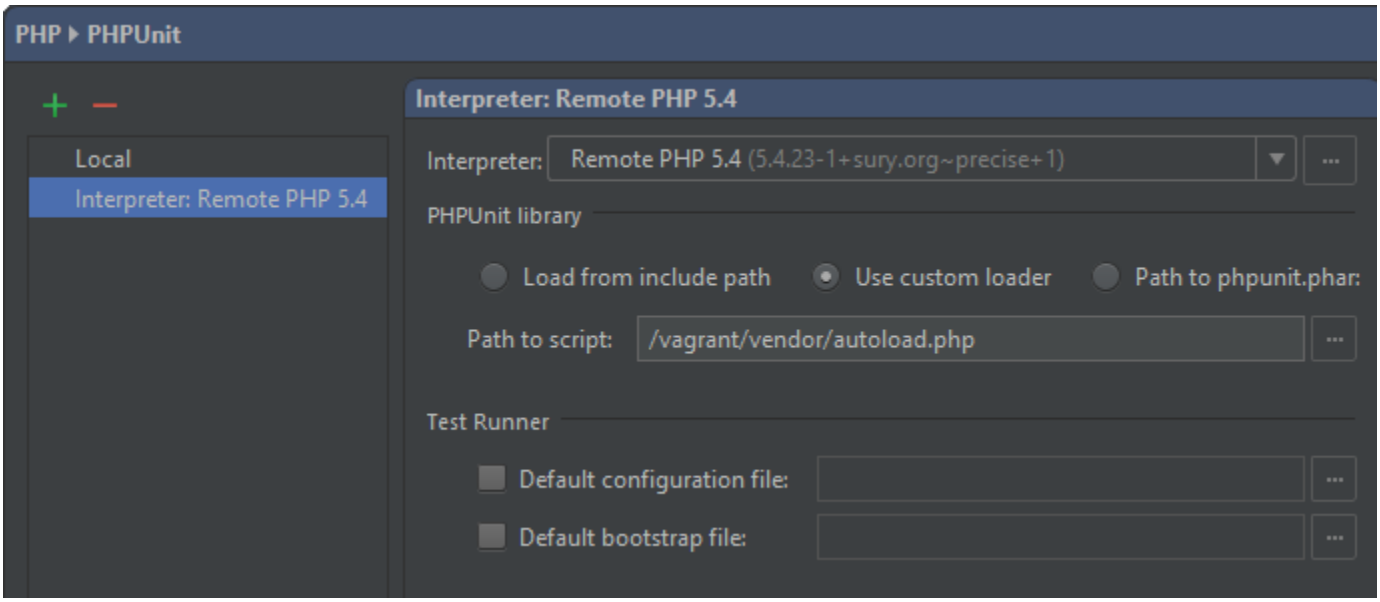
4. Create a PHP Script run/debug configuration or auto-generate run/debug configuration from the context menu of the file in P roject View tool window.



Please note that remote debugging with remote PHP interpreters is currently available for Xdebug only (not for Zend Debugger).

PHPUnit and Remote PHP Interpreters

We can also run PHPUnit tests using remote PHP interpreters. To do so, configure a remote PHP interpreter first. Next, click the green + button in File | Settings (or PhpStorm | Preferences for Mac) | Languages & Frameworks | PHP | Test Frameworks and add PHPUnit settings By Remote Interpreter.



PhpStorm will need the remote path to the PHPUnit configuration file and/or autoloader.

Once done, create a new PHPUnit Run/Debug configuration as we would do normally. We can then run, debug and profile PHPUnit using the remote PHP interpreter.

See Running PHPUnit tests over SSH on a remote server with PhpStorm for a complete tutorial.

Troubleshooting Remote PHP Interpreters

PhpStorm will take all necessary efforts in mapping the local project path in PhpStorm to the remote path. For example for a

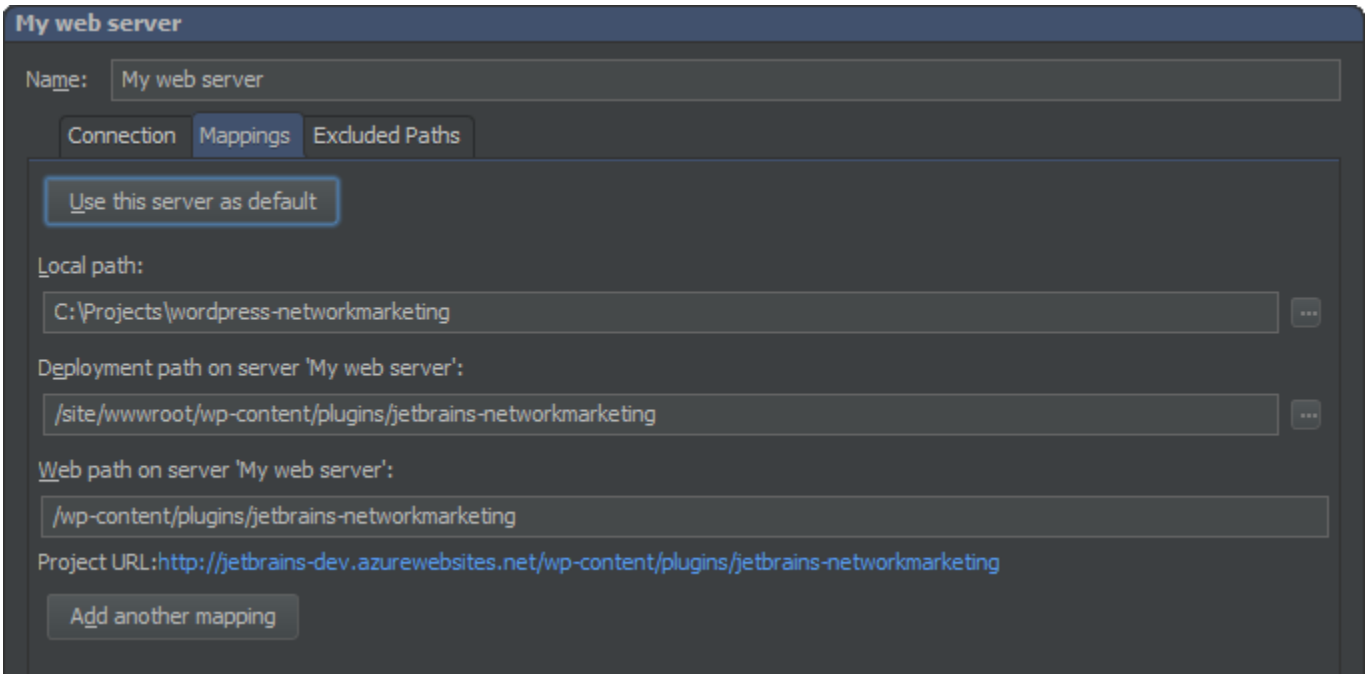
simple Vagrant configuration (e.g. `config.vm.synced_folder "src/", "/srv/website"`), the IDE will know how to map all folders under the project's `src/` folder to the remote `/srv/website` path.

In some cases, for example when additional shared folders have been configured, it may be required to add the remote PHP interpreter from SFTP deployment configuration and configure mappings manually. A mapping is the relation between the project folders, the folders on the remote server, and the URLs to access the data on the server over HTTP. These mappings are used to help PhpStorm find a file or folder in our project or on the remote server.

The easiest way is to map the entire project root folder to a folder on the server, where the project folder structure will be identical on the server. Depending on the server configuration and project layout, it may be useful to specify one or more mappings through the Mappings tab when configuring a deployment server.

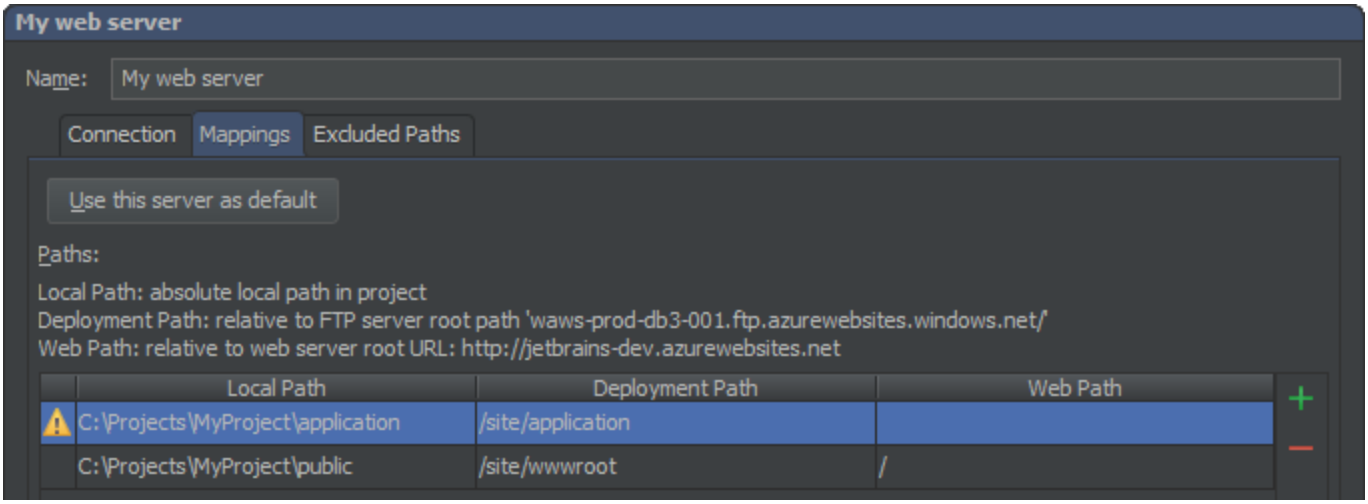
In the screenshot below, we're specifying the following relations:

Local path	Path on the remote server	Web pa
`C:\Projects\wordpress-networkmarketing`	`/site/wwwroot/wp-content/plugins/jetbrains-networkmarketing`	`/wp-co`



In the case where multiple folders should be mapped, which is common for frameworks like Symfony2, Zend Framework, Laravel or CakePHP that typically make use of two main folders, we can configure additional mappings:

Local path	Path on the remote server	Web path (URL)
`C:\Projects\MyProject\application`	`/site/application`	
`C:\Projects\MyProject\public`	`/site/wwwroot`	`/`



Note the warning shown is because the /application path has no web path specified, which in this setup is perfectly fine.

> ⚠ Note that when folders match multiple relevant mappings, the nearest one is applied.

Tweet