

# Develop and Test Your PHP Applications on the IBM cloud

Leverage the IBM Smart Business Development & Test  
Cloud to improve your PHP applications

Daniel Krook  
Software Engineer, IBM  
[krook@us.ibm.com](mailto:krook@us.ibm.com)



# Table of contents

1 Introduction	1
2 The Development & Test cloud	1
3 How this cloud benefits PHP developers	1
4 Create PHP development environment on the Cloud	4
5 Next steps	19
6 Conclusion	19
7 Resources	20

## **About the author**

Daniel Krook is an IBM/Open Group Master Certified IT Specialist based in the greater New York City area. He has over ten years experience in Web application development and currently builds a Cloud infrastructure for IBM using Java EE, DB2, REST, and mobile technologies. He holds certifications in PHP, Java EE, BlackBerry, DB2, and Solaris. He occasionally writes PHP-related articles for IBM developerWorks and co-authored the IBM Redbook “Developing PHP Applications for IBM Data Servers.”

**Note:**

If you are reading this document in printed form, please refer to the links in the “Resources” section on page 20.

**Signing up for the Development & Test Cloud**

If you are not already an academic or corporate user of the Development and Test Cloud, [sign up here](#).

## 1 Introduction

This article describes several ways that cloud computing can improve the quality of—and expand the market for—your PHP applications. In particular, it will explain how the [IBM Smart Business Development & Test Cloud](#) can be used to realize many of these benefits as well as provide step-by-step instructions for setting up a typical PHP test environment on the IBM Cloud.

## 2 The Development & Test Cloud

As described in a [previous article](#) on developerWorks, the IBM Development & Test Cloud is a dynamically provisioned environment intended for use in the construction and validation of applications.

The IBM Development & Test Cloud is an Infrastructure as a Service (IaaS) that provides access to a variety of Linux server and desktop images that are pre-configured with IBM software.

A Web, mobile, command-line or custom application can be used to provision, configure and manage these virtual resources. You can then access them over SSH, HTTP, and FreeNX to install, run and test your applications.

Among the greatest risks to the planned release date of a project and its overall quality at launch is the time and cost of setting up a traditional development and test infrastructure comprised of physical servers. The IBM Development & Test Cloud significantly reduces these two risk factors. Instead of purchasing computers, and installing operating systems and middleware (which can cost time and money—and be error prone), the IBM cloud reduces that risk and cost by enabling pre-configured virtual servers to be instantiated simply, quickly and repeatedly.

You can find out more about the [pricing](#) and [technical details](#) of the Development & Test Cloud on its [offering page](#).

## 3 How this Cloud benefits PHP developers

As an IaaS, IBM’s Development & Test Cloud can help you ensure that your applications are released faster, with fewer bugs, to a larger market than an alternative Cloud or physical development and test environment.

Central to this offering is a catalog of Linux images that are pre-configured with IBM software. An instance that is created from an image can mount a persistent disk, have a static IP address assigned to it, and it can be accessed using public key

### Why not use the IBM images on EC2?

There are several pre-configured [IBM middleware images on Amazon EC2](#). You could also develop your applications there, but the Development and Test Cloud offers the following advantages:

- Free access for academic and pilot users.
- Premium IBM and operating system vendor support.
- Detailed control over network and capacity usage.
- Simplified custom instance image creation.

authentication. All of these resources can be provisioned quickly and arranged into a topology that matches the needs of your PHP application.

In addition, images can be saved from your customized instances into a private image catalog, which makes your configurations reusable, retest-able and repeatable. All of this adds up to higher quality applications.

The IBM Development & Test Cloud can address many types of challenges that affect PHP developers of all stripes, including corporate and freelance programmers, students and writers, as well as system administrators and architects.

Here are a few specific scenarios where the Development & Test Cloud can help you achieve your quality goals and expand the market for your software.

#### Verify your deployment environment

Eventually you're going to need to host your project somewhere. The Development & Test Cloud can help you document your deployment steps earlier in the process. You can also use the cloud to set up a test configuration so that you can perform UAT earlier in your release cycle, before your ultimate production hosting environment is available. You'll know exactly what you need to order (or provision on a production cloud), reducing the risk of error and misconfiguration.

For example, you may need to evaluate whether your application will run better on one version or packaged distribution of PHP rather than another, or you may want to verify the publication process from staging to production or the replication configuration between two MySQL database servers.

#### Speed innovation and encourage experimentation

Whether you are starting a new project, or planning the next evolution of your current legacy project, the IBM Development & Test Cloud can help you by empowering your individual developers to innovate and compare platforms side-by-side. You can give them freedom, reduce risk, and flesh out concepts outside of the main branch without impacting stable code and configurations.

For example, if you are evaluating a move from one PHP framework to another, such as Lithium, creating a separate, virtual instance of PHP 5.3 with MongoDB on the Cloud will be quicker and less disruptive than setting up the same configuration on an existing development server.

#### Resolve performance problems and pre-empt issues

Perhaps you have a performance problem, and you want to tweak your platform and test out theories quickly—but you don't want to make major changes to your development or staging areas. Instead, you can now use the IBM Development & Test Cloud to run those tests. Installation steps can be recorded and back out plans can be validated.

For example, multiple possible configurations can be evaluated and tested using a cloud infrastructure rather than a physical system. In addition, by saving a catalog of private images, you can quickly retest or isolate issues to particular configurations.

### **Expand the market for your software**

If your current project only runs on a Linux, Apache, MySQL, PHP (LAMP) platform and you want to expand your market, you can try out DB2, Informix or WebSphere on the cloud, and quickly port and verify your apps on these platforms.

For example, if you are considering supporting DB2 in addition to MySQL, or porting your PHP application to Java, you can save time by using the IBM Development and Test Cloud, which offers servers preconfigured with DB2 and WebSphere Application Server.

### **Improve your open source project**

As a distributor of an open source framework, your goal is to deliver a stable product that works on as many platforms as possible. Perhaps you get a bug report from a user in a particular environment. With the IBM cloud, you can easily replicate the user's environment to identify the source of the bug.

For example, if a user reported an issue with a version of SUSE Enterprise Linux, but all you have is Ubuntu or Fedora, you could quickly provision an instance of SUSE Enterprise Linux to verify the bug report. In addition, you could expand the list of supported platforms after performing such a test by maintaining several enterprise Linux configurations.

### **Build your skills and broaden your knowledge of IBM software**

Beyond the perspective of an independent software vendor (ISV) or small business, many others can benefit as well from the IBM Development and Test Cloud. Technical documentation writers, students, instructors and authors, or industry analysts can also benefit by being able to document new software or experimenting with a piece of software they need for only a limited time. Perhaps they have a final project or tight deadline, but don't want to make a long-term commitment to the purchase or setup of complete systems from scratch. The IBM Development & Test Cloud could be the answer.

## 4 Create a PHP development environment on the Cloud

### The catalog of images

The catalog of available images is subject to change, and will include regular updates to IBM software, along with subsequently refreshed software images.

If you do not see the exact image names I've used in the examples below, a similar image will usually suffice.

If you prefer a certain configuration, you can also save a private image from an instance you provisioned from the public catalog and use that for future testing.

Now that you have a feel for the value the Development & Test Cloud can provide, I'll take you through a few steps to configure a PHP test environment.

We'll provision two instances; a DB2 data server running on a Linux operating system, and a Linux operating system with a DB2 client. On the DB2 client we will install a pre-packaged, GUI-driven PHP distribution with many included extensions.

This will let us simulate a typical two-tier test environment where the database is given a dedicated server, and the application is installed on its own server.

This example will also illustrate how much faster it is to set up a topology using the IBM cloud instead of purchasing, configuring, and installing two physical servers.

### Provision a Linux data server

For our data server we will select a Red Hat Enterprise Linux server from the catalog that already has DB2 installed. We will create a Web server instance later, where we will install PHP and connect to this server, so no special configuration is required once we provision this instance.

1. Log into the Development & Test application and select the Control Panel tab. Select "Add Instance".

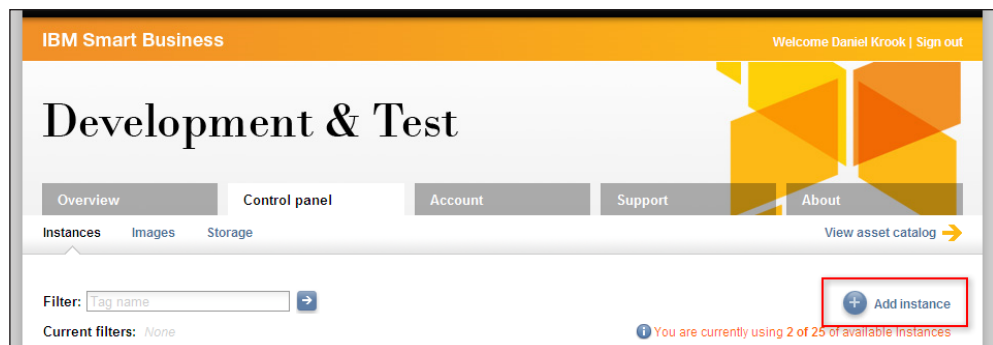


Figure 1: The Add Instance button on the Control Panel

2. Choose “DB2 Enterprise Developer Edition 9.7.1 64bit” from the catalog.

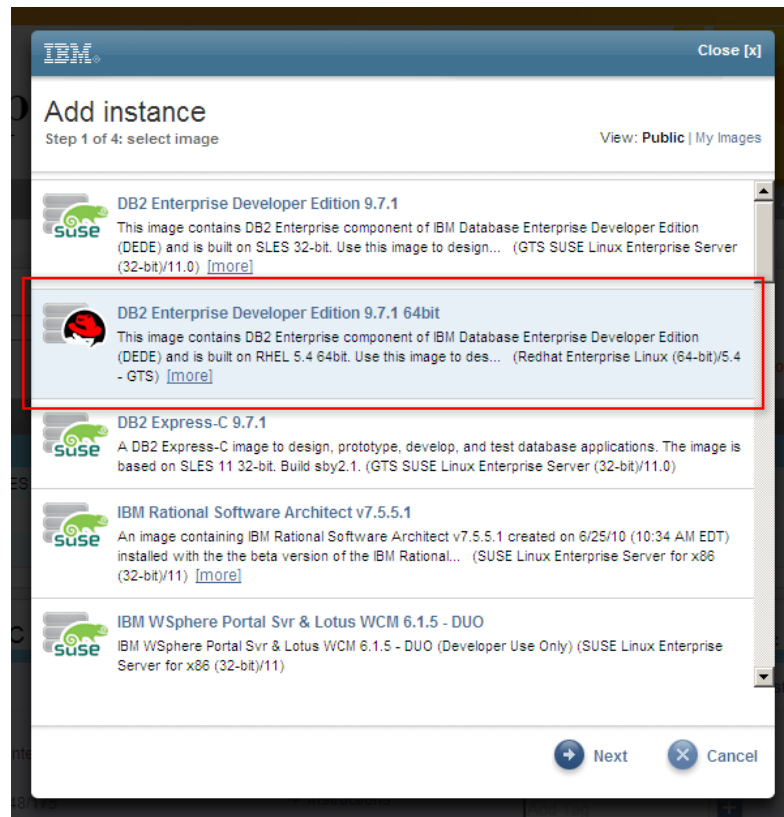


Figure 2: Step 1 of 4, select a DB2 image

3. Give it a name, and generate a public/private key if necessary. I've called my instance “dW DB2 data server” and chosen my “dW” public key. You use the corresponding private key on your workstation to access this instance, instead of a user name and password.

**IBM** Close [x]

### Add instance

Step 2 of 4: configure image

**You selected: DB2 Enterprise Developer Edition 9.7.1 64bit**  
 This image contains DB2 Enterprise component of IBM Database Enterprise Developer Edition (DEDE) and is built on RHEL 5.4 64bit. Use this image to design, prototype, develop, and test database applications (non-production use). Build 3.2.sby

Complete the fields below to configure your instance selection. Required fields are indicated with an asterisk (\*).

**Request Name:** dW DB2 data server

**Quantity:** 1

**Server Size:** Bronze 64 bit

**Expires on:** 10/9/10

**Key:** dW key (default) Add Key

**Select IP:** system generated How do I add an IP?

**Mount Storage:** Choose one How do I add Storage?

**Image ID:** 10010275

Previous Next Cancel

Figure 3: Step 2 of 4, configure the DB2 image

4. Set the passwords for the three new user accounts that DB2 requires.

**IBM** Close [x]

### Add instance

Step 2b of 4: configuration additional parameters

Complete the fields below to configure your instance selection. Required fields are indicated with an asterisk (\*).

**Instance owner (db2inst1) password:** [password field]  
 Virtually everything you do with DB2 will be done as the instance owner.

**Re-enter Password:** [password field]

**Fenced user (db2fenc1) password:** [password field]  
 The fenced user runs certain stored procedures.

**Re-enter Password:** [password field]

**DAS user (dasusr1) password:** [password field]  
 The DAS administrator starts and stops the DAS service.

**Re-enter Password:** [password field]

Previous Next Cancel

Figure 4: Step 2b of 4, configure the DB2 passwords



5. Verify the configuration of the instance.

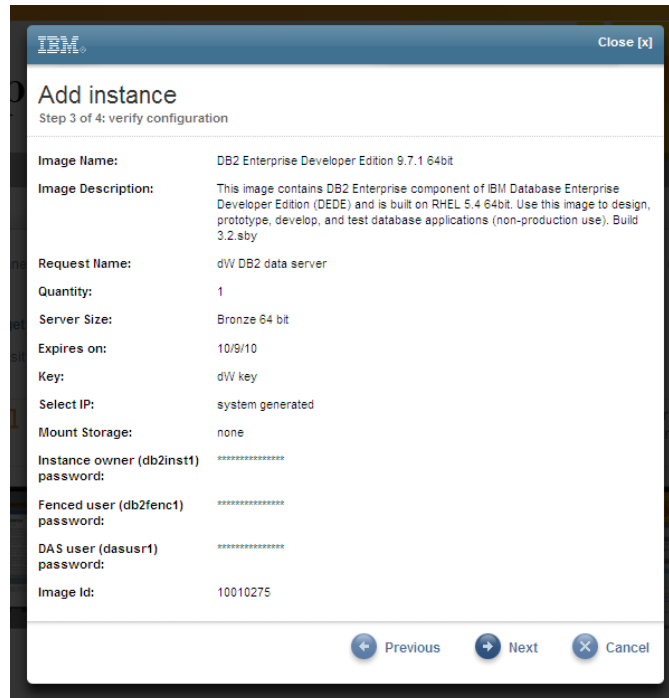


Figure 5: Step 3 of 4, verify instance configuration

6. Review the license agreement and choose Next.

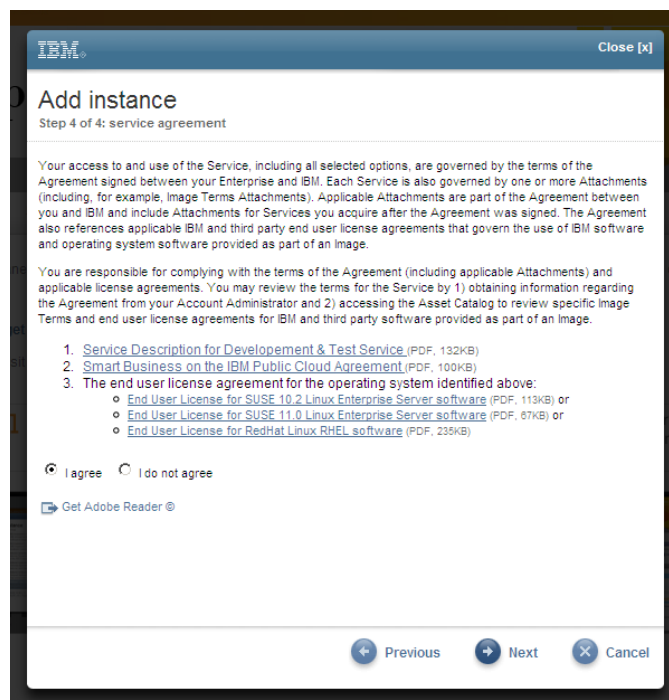


Figure 6: Step 4 of 4, read the license agreement

7. Confirm that the instance request has been submitted.

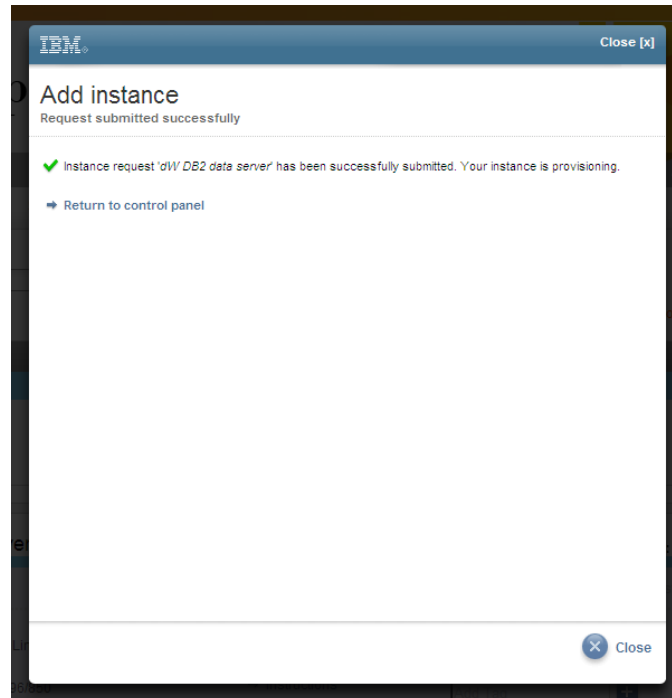


Figure 7: Instance request submitted successfully

That's all we need to do to spin up an instance to be used as our data server. You can monitor the control panel, which will show the status of the instance as it is provisioned and starts up. When the system boots, DB2 will already be running.

### Provision a Linux Web application server

We will create another instance from the same image and install Zend Server Community Edition. This is the simplest way to ensure that the DB2 client that the PHP driver needs is already installed and available.

1. Log into the Development & Test application and select the Control Panel tab. Select "Add Instance" as you did in the previous section.
2. Choose the same "DB2 Enterprise Developer Edition 9.7.1 64bit" image from the catalog.
3. Give it a name, and generate a key if necessary, as above. I've called my instance "dW Zend Web server" and chosen my "dW" public key.
4. Set the DB2 passwords.
5. Confirm the configuration.
6. Review the license agreement.
7. Click Next and wait for confirmation.

**Tip: Reusing a PuTTY session configuration**

You can reuse the PuTTY session you set up below for future instances by loading it, changing the “Host name” field to your new host, renaming it, and clicking Save.

You can now monitor the status of the instance request on the control panel. The DB2 client will be installed and running when the instance starts up.

**Convert your private key and connect to your server using SSH**

Once the Web application server instance has provisioned, you can log into it and install Zend Server Community Edition.

The following steps assume you are using Windows for SSH and X forwarding. If you're using a different platform, you'll need to install the equivalent tools for that platform.

1. When you created the instances above, you had an option to generate a public/private key pair in order to connect to your servers without a user name and password.

You'll need to convert the private key that you downloaded to your workstation from OpenSSH format to PuTTY's private key format using the PuTTY Key Generator, PuTTYgen. Open PuTTYgen and choose File... Load private key.

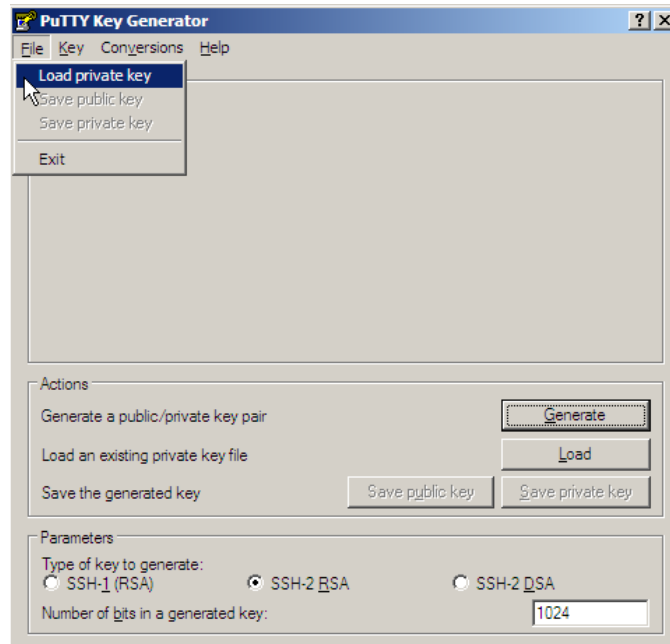


Figure 8: Convert private key

2. You'll see a notice that this key is in the OpenSSH format. Click OK.

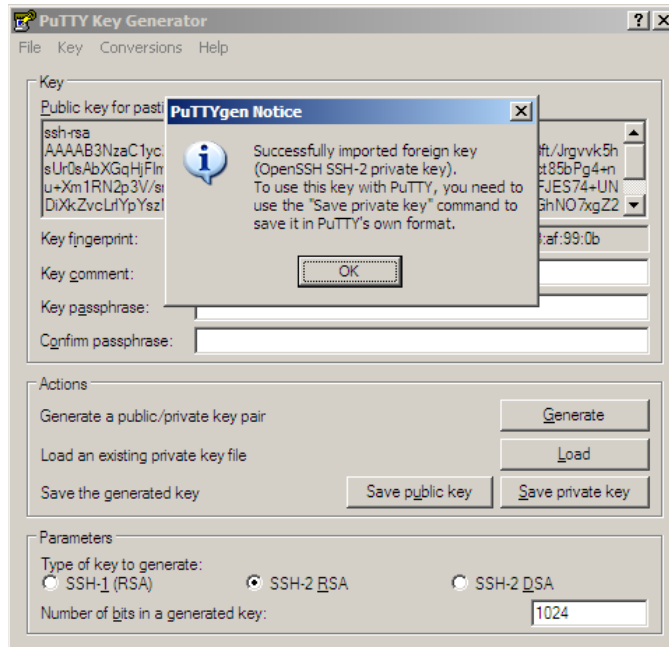


Figure 9: Loaded OpenSSH private key

3. Select the Save private key button and pick a new location for the converted file. You can choose Yes to confirm that you want to save the converted key without a passphrase.

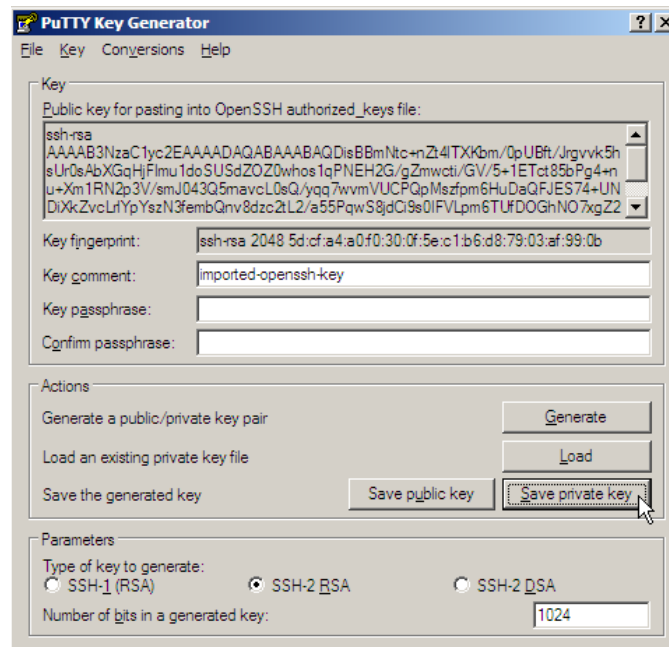


Figure 10: Save converted key

4. Install and configure Xming. It should start automatically. If not, click the Xming shortcut.
5. Configure PuTTY to log into the instance using SSH. Add the hostname you see in the control panel after your instance has finished provisioning. Click save, then Load.

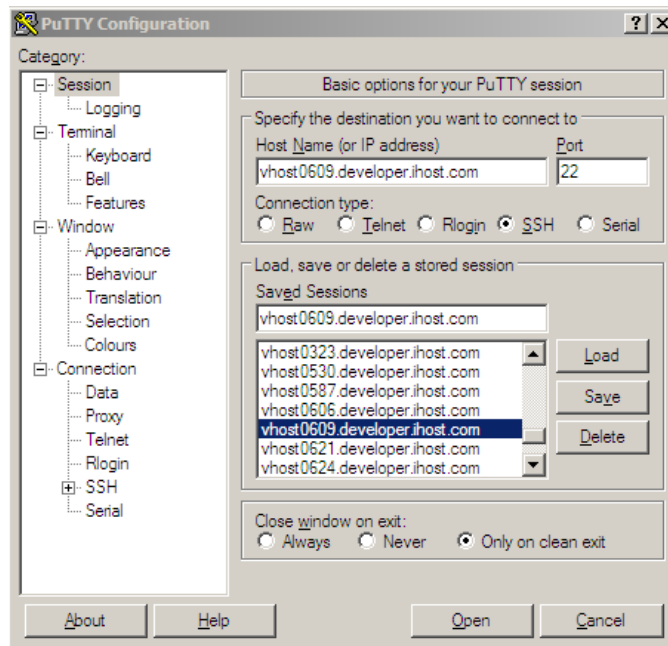


Figure 11: PuTTY configuration

6. Set the user to “idcuser” under Connection...Data in the Auto-login username field.

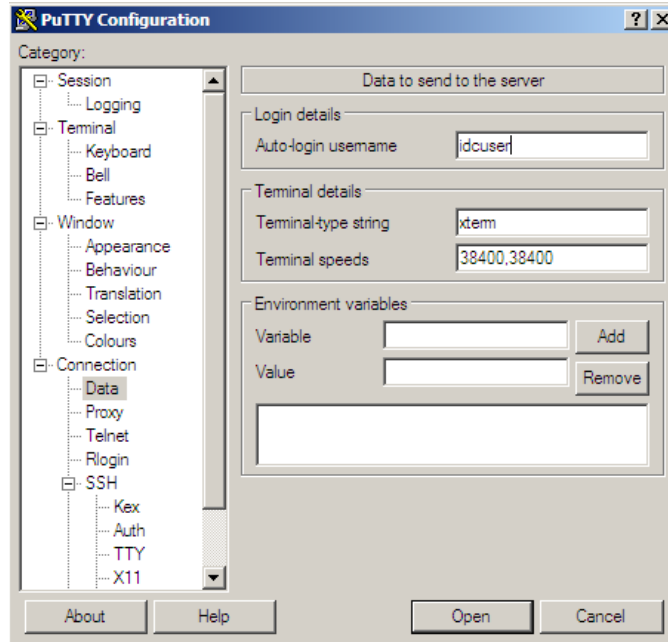


Figure 12: Set user name

7. Import your private key on the SSH...Auth screen and Browse for the private key file you converted earlier.

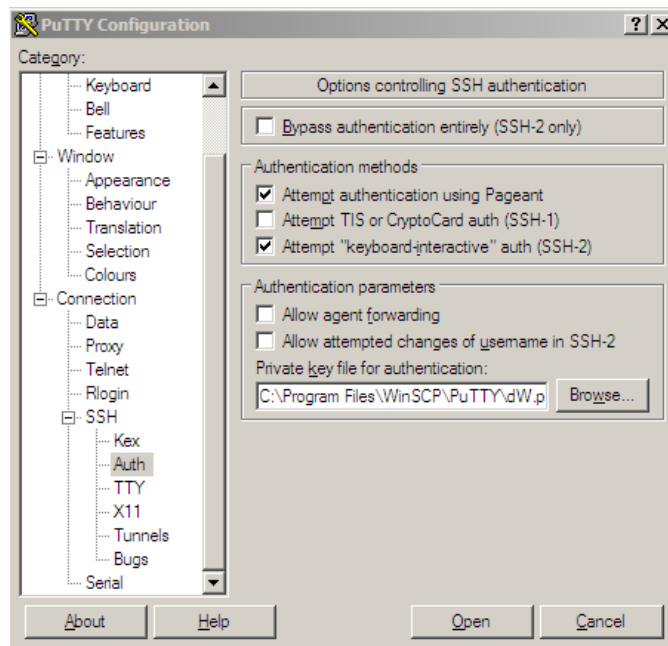


Figure 13: Import converted key

8. Finally, set up X-forwarding in order to view X Windows from the Linux server on your Windows desktop. On the SSH...X11 screen, check the X11 forwarding box and enter "localhost:0" in the X display location field.

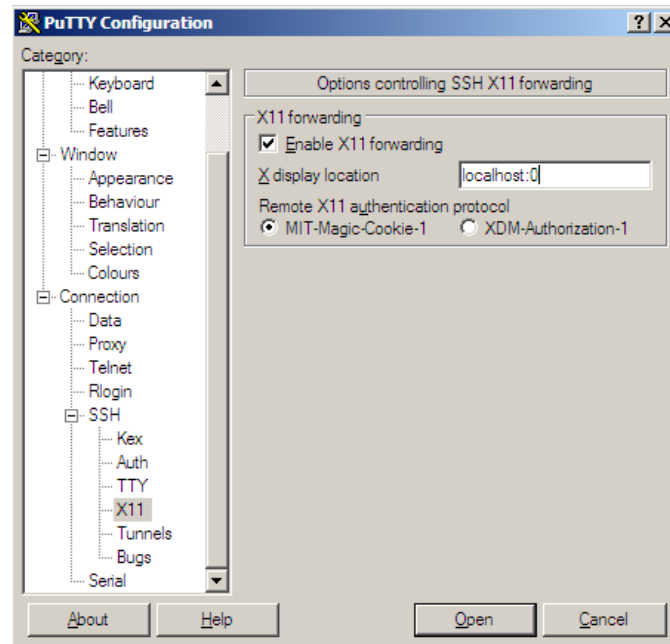


Figure 14: Configure X forwarding

9. Now, go back to Session and click Save, then Open. Click Yes to the PuTTY Security Alert and accept the instance host key, as this is the first time you are connecting to this server.
10. Disable the firewall in order to allow X forwarding. Then test X Windows using a simple test with xclock:

*Listing 1:* Shut down the firewall and run xclock

```
$ sudo /etc/init.d/iptables stop
$ xclock
```

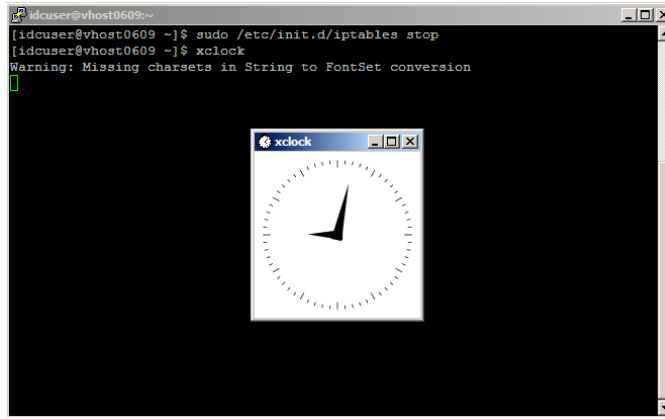


Figure 15: Disable iptables and test X forwarding.

### Install a pre-configured PHP distribution

1. Now that you are logged in and set up for a GUI installation, download Zend Server Community Edition (Zend Server CE (PHP 5.3) 5.0.3 (64 bit) (TAR.GZ) 55.95 MB). We'll use the direct download link below, but you can find out more about Zend Server at on the Zend product page .

#### Listing 2: Downloading and installing Zend Server

```
$ mkdir src
$ cd src
$ wget http://www.zend.com/download/353?start=true
$ tar -xvzf ZendServer-CE-php-5.3.3-5.0.3-linux-glibc23-
    x86_64.tar.gz
$ cd ZendServer-5.0.3_Tarball_B3-php5.3.3-linux-glibc23-
    x86_64/
$ sudo su -
# cd /home/idcuser/src/ZendServer-5.0.3_Tarball_B3-
    php5.3.3-linux-glibc23-x86_64/
# ./install.sh
```



Allow time for the system to complete the installation. After a few minutes, you should see the following screen.

```
lighttpd: Could not reliably determine the server's fully qualified domain name, us
/usr/local/zend/bin/apachectl start [OK]
spawn-fcgi: child spawned successfully: PID: 10728
Starting Zend Server GUI [Lighttpd] [OK]
[ 18.09.2010 21:18:53 SYSTEM] watchdog for lighttpd is running.
[ 18.09.2010 21:18:53 SYSTEM] lighttpd is running.

Zend Server started...

----- INSTALLATION SUMMARY -----

Zend Server was installed to /usr/local/zend
The End User License Agreement [EULA] can be viewed under /usr/local/zend/d
See the README in /usr/local/zend/doc/README more information

To change the GUI password run /usr/local/zend/bin/gui_passwd.sh
Apache is up and running on port 10088!
To control Zend Server, please use /usr/local/zend/bin/zendctl.sh
To enable the Java bridge, please run /usr/local/zend/bin/setup_jb.sh
Web interface is accessible from https://localhost:10082/ZendServer

----- ENJOY Zend Server -----
idcuser@vhost0609 ZendServer-5.0.3_Tarball_B3-php5.3.3-linux-glibc23-x86_64$
```

Figure 16: Install Zend Server

2. Type `exit` to escape the root shell and resume the `idcuser` session, then launch the Firefox Web browser. You won't be able to launch any windows as root, as you configured X forwarding in PuTTY only for `idcuser`.

### Listing 3: Launch Firefox as `idcuser`

```
# exit
$ firefox https://localhost:10082/ZendServer
```

Since Zend Server is running over HTTPS with a self-signed certificate, you'll need to add an exception when prompted to view the page. Once Firefox has launched, the first page you will see is the End User License Agreement. Read it and click Next.

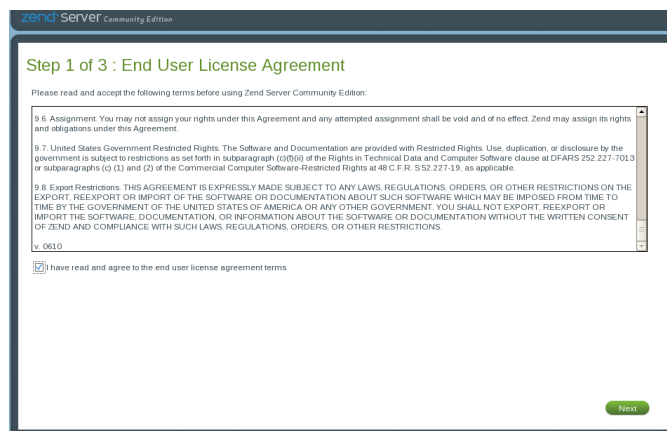


Figure 17: Zend Server EULA

3. On the second page, set a console password, retype it and click Next.
4. On the third page you can optionally provide your email address if you would like to join the Zend Server Community Edition mailing list. Then click Finish.
5. You'll then see the Zend Server Dashboard. The most common extensions are included and enabled out of the box. The DB2 driver extension is included, but we need to activate it.
6. Choose the Server Setup tab, then the Extensions link below it and scroll down to `ibm_db2`. Click Turn On.



Figure 18: `ibm_db2` extension

7. In a new PuTTY session, we'll source the DB2 environment for the Apache Web server and change the default port to 80, then restart all Zend Server daemons.

First, edit the Apache configuration file and change the line "Listen 10088" to "Listen 80"

*Listing 4:* Edit httpd.conf to change the port.

```
$ sudo su -  
# vi /usr/local/zend/apache2/conf/httpd.conf
```

Second, source the DB2 environment for Apache each time it starts up. This is necessary in order to set the path for Apache to find DB2 for the PHP driver. Once you open the file, add the line “. /home/db2inst1/sqllib/db2profile” anywhere after the first line.

*Listing 5:* Edit apachectl to source the DB2 environment.

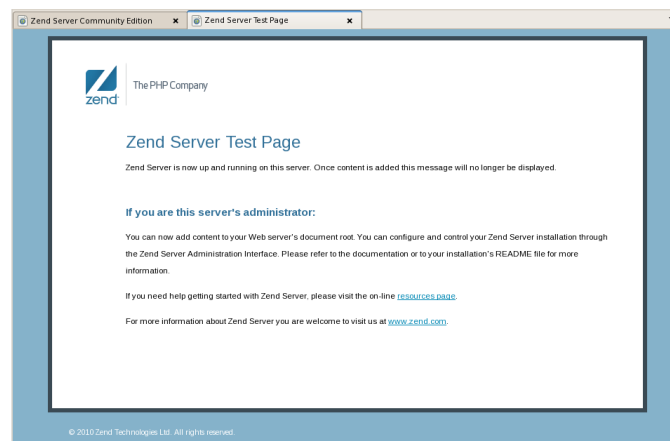
```
# vi /usr/local/zend/apache2/bin/apachectl
```

Third, restart all of the Zend Server processes from the command line. Clicking the “Restart PHP” button in the GUI will not be sufficient.

*Listing 6:* Restart the Zend Server processes.

```
# /usr/local/zend/bin/zendctl.sh restart
```

Open a new tab in the existing Firefox window and enter `http://localhost/`. You should see a screen that says “Zend Server Test Page.”



*Figure 19: Zend Server Test Page*

## Test the connection between the Zend Web server and the DB2 data server

1. Finally, we'll create a simple PHP script to test that we have configured the DB2 connection correctly. I'll create the following PHP script in the /var/www/html directory. It simply connects to the data server to verify the connection.

*Listing 7:* Edit apachectl to source the DB2 environment.

```
# vi /usr/local/zend/apache2/htdocs/db2info.php
```

*Listing 8.* PHP script to connect to DB2

```
<?php
$username = 'db2inst1';
$password = 'passw0rd';
$database = 'sample';
$hostname = 'vhost0323.developer.ihost.com';
$port      = '50001';

$conn_string =
    "DRIVER={IBM DB2 ODBC DRIVER};DATABASE=$database;" .
    "HOSTNAME=$hostname;PORT=$port;PROTOCOL=TCPIP;" .
    "UID=$username;PWD=$password;";

$conn_resource = db2_connect($conn_string, '', '');

if ($conn_resource) {
    echo 'Connection to database succeeded.';
    db2_close($conn_resource);
} else {
    echo 'Connection to database failed.<br />';
    echo 'SQLSTATE: ' . db2_conn_error() . '<br />';
    echo 'Message: ' . db2_conn_errormsg() . '<br />';
}
?>
```

2. After saving this file, we should see the following screen when we load <http://localhost/db2info.php> in the browser.

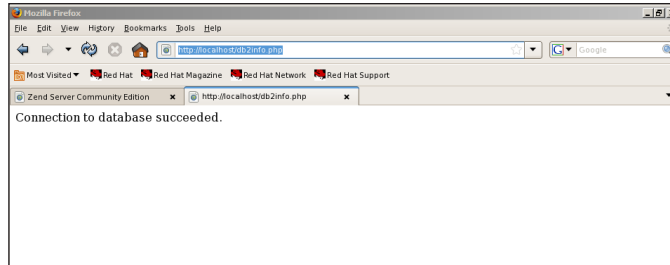


Figure 20: Successful connection from PHP to DB2

Congratulations, you now have two virtual servers configured to support your PHP and DB2 powered Web application! The steps you followed in to create this system will be similar for any other virtual servers you decide to instantiate in the future.

## 5 Next steps

With a typical test configuration up and running, you can now tackle any one of the scenarios I described earlier in the article.

- Verify your deployment environment.
- Speed innovation and encourage experimentation.
- Resolve performance problems and pre-empt issues.
- Expand the market for your software.
- Improve your open source project.
- Build your skills and broaden your knowledge of IBM software.

Now that you've seen how to provision a PHP development environment, connect to it, and install a simple script, you can start to experiment to your heart's content. If you discover any other compelling use cases, we invite you to tell us about them in the the Support forum.

If you don't yet have an academic or commercial account and you'd like to find out more about the Development and Test Cloud, visit the [offering page](#).

## 6 Conclusion

In this article, you learned how the IBM Smart Business Development & Test on the IBM Cloud can be used to improve your PHP applications and expand their potential market. You then set up a virtual development server configuration that can serve as a starting point to experiment with other development and test scenarios.

This article showed you:

- The unique opportunities that cloud computing and the Development & Test Cloud can offer PHP developers.
- How to quickly spin up a PHP development environment on the Development & Test Cloud.

Sometimes a paradigm shift in computing can make vague promises about time and money saved. I hope this article gave you concrete examples of what the Cloud can help developers and testers do. And hope it helps you start thinking about other scenarios where the Cloud helps improve your apps and bring them to market.

## 7 Resources

- [Offering: IBM Smart Business Development & Test on the IBM Cloud](http://www.ibm.com/services/us/igs/cloud-development/)  
(<http://www.ibm.com/services/us/igs/cloud-development/>)
- [Download: Zend Server](http://www.zend.com/products/server/)  
(<http://www.zend.com/products/server/>)
- [Download: PuTTY and PuTTYgen](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)  
(<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>)
- [Download: Xming X Server for Windows](http://sourceforge.net/projects/xming/)  
(<http://sourceforge.net/projects/xming/>)
- [IBM developerWorks: Recommended PHP reading list](http://www.ibm.com/developerworks/library/os-php-read/index.html)  
(<http://www.ibm.com/developerworks/library/os-php-read/index.html>)
- [IBM Redbooks: Developing PHP Applications for IBM Data Servers](http://www.redbooks.ibm.com/abstracts/sg246899.html)  
(<http://www.redbooks.ibm.com/abstracts/sg246899.html>)
- [IBM Redbooks: Up and Running with DB2 on Linux](http://www.redbooks.ibm.com/abstracts/sg246899.html)  
(<http://www.redbooks.ibm.com/abstracts/sg246899.html>)



©Copyright IBM Corporation 2010

Cloud Computing Enablement, Enterprise Initiatives  
Route 100, Somers, NY 10589 U.S.A.

Printed in the United States of America 2009. All Rights Reserved.

IBM, the IBM logo, the e-business logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.