

Host your Application in the Amazon Cloud with XAMPP and Bitnami

Vikram Vaswani

v1.0

Table of Contents

Introduction	1
What You Will Need	1
Step 1: Register with Amazon Web Services.....	2
Step 2: Register with Bitnami	4
Step 3: Connect your AWS and Bitnami Accounts	5
Step 4: Provision an AWS Cloud Server	8
Step 5: Test PHP and MySQL	11
Step 6: Deploy the XAMPP Application to the Cloud Server	17
Understand the Bitnami Cloud Hosting Dashboard and Monitoring Tools	21
Improve Application Performance	23
Useful Links.....	24
About the author.....	24

Introduction

If you're a PHP developer building a public-facing Web application, there are a number of good reasons why the cloud should be on your radar. It's highly scalable, allowing you to quickly scale up if your application turns out to be a hit. It's cost-efficient, because you only pay for the resources - bandwidth, CPU cycles, memory - you use. And it's secure, because cloud providers have invested a great deal of time and thought into ring-fencing applications and user data.

However, if you're new to the cloud or do most of your development locally, getting your PHP application from your local [XAMPP](#) box to the cloud can be a bit challenging. That's where this tutorial comes in. Over the next few pages, I'll walk you, step by step, through the process of deploying a PHP/MySQL application running on your local XAMPP server, to a cloud server running [Bitnami's LAMP Stack](#). Keep reading!

What You Will Need

Before we begin, a few quick assumptions. This tutorial assumes that you have a XAMPP installation with a working PHP/MySQL application. It also assumes that you're familiar with the [MySQL command-line client](#) and that you have a working knowledge of transferring files between servers using FTP.

If you don't have a custom PHP/MySQL application at hand, use the example application included with this tutorial: it's a simple to-do list, created with [Twitter Bootstrap](#) and PHP. You can download it [from here](#).

Now, if you're new to the cloud, you might be wondering what Amazon Web Services and Bitnami are. Very briefly, [Amazon Web Services](#) is a cloud platform, which allows you to easily create Windows and Linux virtual servers online. Bitnami provides pre-packaged server images for these cloud servers, so that you can become productive with them the moment they come online. In short, Amazon provides the cloud infrastructure, and Bitnami provides the server images and software. And since both Amazon and Bitnami have a free tier, you can run and manage a full-featured PHP server for free for 1 year.

For this tutorial, I'll be using the [Bitnami LAMP Stack](#), which is Linux-based and bundles PHP, MySQL and Apache, together with key applications and components like phpMyAdmin, SQLite, Memcache, OpenSSL, APC and cURL. The LAMP stack also includes a number of common PHP frameworks, including the [Zend Framework](#), [Symfony](#), [CodeIgniter](#), [CakePHP](#), [Smarty](#) and [Laravel](#).

To deploy your application to the Amazon cloud with the Bitnami LAMP Stack, here are the steps you'll follow:

- Register with Amazon Web Services (AWS)
- Register with Bitnami
- Connect your AWS and Bitnami accounts
- Provision an AWS cloud server with the Bitnami LAMP Stack
- Validate the cloud server

- Deploy and test your application on the cloud server

The next sections will walk you through these steps in detail.

Step 1: Register with Amazon Web Services

At the end of this step, you will have signed up for the Amazon Web Services free tier.

Begin by creating an AWS account, by browsing to <http://aws.amazon.com> and clicking the "Sign Up" button at the top of the page. You will need an existing Amazon account to log in and sign up; if you don't have one, create one to proceed.

Once you've signed in to Amazon, sign up for AWS by providing some basic contact information and your mobile phone number.

Amazon Web Services Sign Up

Contact Information

* Required Fields

Full Name*

Company Name

Country* United States

Address* Street, P.O. Box, Company Name, c/o
Apartment, suite, unit, building, floor, etc.

City*

State / Province or Region*

Postal Code*

Phone Number*

Security Check (Optional)

P3V3B3

Once that's done, proceed to the next stage by entering your credit card information.

Amazon Web Services Sign Up

Contact Information Payment Information Identity Verification Support Plan Confirmation

Payment Information

Please enter your payment information below. You will be able to try a broad set of AWS products for free via the Free Usage Tier. We will only bill your credit card for usage that is not covered by our Free Usage Tier.

AWS Free Usage Tier	Compute Amazon EC2	Storage Amazon S3	Database Amazon RDS
free for 1 year	750hrs/month*	5GB	750hrs/month*

*View full offer details >

Credit Card Number

Expiration Date

Cardholder's Name

Choose Your Billing Address
Select the address associated with your credit card.

If you're worried about how much you'll be billed for services, relax. When you first sign up for AWS, you get automatic access to the [AWS free tier](#), which entitles you to 12 months of free usage up to certain limits. This

includes 750 hours per month of free usage of **Amazon EC2** micro servers, which come with 2.5-3.3 GHz Intel Xeon processors, 1 GB of memory and 1 virtual core - just right for development or low-traffic website hosting. So long as your usage falls within the limits of the free tier, your credit card will never be billed. However, Amazon still needs your credit card information for security purposes, to avoid service misuse and to confirm your identity.

IMPORTANT

You should fully understand the limits of the AWS free tier to avoid being unduly charged for service usage.

Amazon will now verify your identity, by making an automated call to your mobile phone number and prompting you to enter the PIN number displayed on the screen.

The screenshot shows the 'Identity Verification' step of the AWS sign-up process. At the top, there's a progress bar with five steps: Contact Information, Payment Information, Identity Verification (which has a red checkmark), Support Plan, and Confirmation. The main area is titled 'Identity Verification' and contains the following text:
You will be called immediately by an automated system and prompted to enter the PIN number provided.
1. Provide a telephone number ✓
2. Call in progress
Please follow the instructions on the telephone and key in the following Personal Identification Number (PIN) on your telephone when prompted.
PIN: 5685
If you have not yet received a call at the number indicated above please wait. This page will automatically update with what you need to do next.
3. Identity verification complete

Once your identity is verified, choose the "Basic" support plan (also free) and confirm your account.

The screenshot shows the 'Support Plan' step of the AWS sign-up process. At the top, there's a progress bar with five steps: Contact Information, Payment Information, Identity Verification (with a red checkmark), Support Plan (which has a red checkmark), and Confirmation. The main area is titled 'Support Plan' and contains the following text:
All customers receive free support. Choosing a paid support plan will allow you to receive one-on-one technical assistance from experienced engineers and access many other support features. Please see below.
Please Select One
 Basic (Free)
Contact Customer Service for account and billing questions, receive help for resources that don't pass system health checks, and access the AWS Community Forums.
 Developer (\$8/month)
Get started on AWS - ask technical questions and get a response to your web case within 12 hours during local business hours.
 Business (Starting at \$100/month - Pricing Example) - Recommended
24/7/365 real-time assistance by phone and chat, a 1 hour response to web cases, and help with 3rd party software. Access AWS Trusted Advisor to increase performance, fault tolerance, security, and potentially save money.
 Enterprise Support
15 minute response to web cases, an assigned technical account manager (TAM) who is an expert in your use case, and white-glove case handling that notifies your TAM and the service engineering team of a critical issue.
If you select this option, you will not be charged immediately. We will contact you to discuss your needs and finalize the signup.

The AWS account registration machine will churn away for a minute or so, and you will then be redirected to a welcome page, which includes a link to the AWS management console. You should also receive an account confirmation email, which tells you that your account is good to go.

Step 2: Register with Bitnami

At the end of this step, you will have created a Bitnami account.

The next step is to create a Bitnami account, so that you can launch a cloud server with Bitnami's LAMP Stack image. If you have a Google, Facebook, Yahoo! or Github account, you can use your credentials from those services with OpenID to create your Bitnami account.

If you don't have accounts with those services (or you don't want to use them), you can use your email address and password to create a Bitnami account, as described below:

- Head to [the Bitnami sign-up page](#).
- Enter your name and email address.
- Choose a password.
- Review and agree to the Bitnami terms of service.

Then, use the "Sign up" button to create your account.

Bitnami will send you an email with a verification link which you'll need to click or browse to, to activate your account. This will also sign you in to your Bitnami account.

Bitnami account registration confirmation

From: hello@bitnami.com, To: _____, Date: _____

Confirm Your Account

Please confirm your account by clicking on the following link:

https://bitnami.com/confirmation?confirmation_token=_____

If you did not sign up for this account, you can disregard this email and the account will not be created.

Regards,

The Bitnami Team

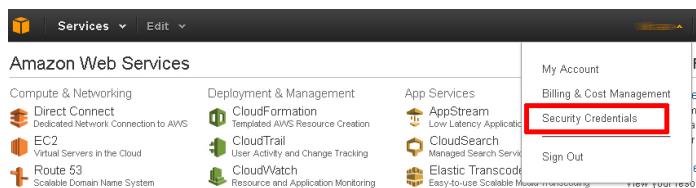
Step 3: Connect your AWS and Bitnami Accounts

At the end of this step, your Bitnami Cloud Hosting account will be configured and you will be ready to provision a cloud server.

The easiest way to set up your AWS cloud server with Bitnami's LAMP Stack is via [Bitnami Cloud Hosting](#), which gives you a simple control panel to provision, start, stop and check status of your AWS cloud servers. Bitnami Cloud Hosting also has a free tier that allows you to deploy, monitor and backup your AWS servers for free. However, to use it, you must first connect your AWS and Bitnami accounts, by obtaining security credentials for your AWS account and saving those credentials in your Bitnami Cloud Hosting account.

To obtain security credentials for your AWS account:

- Log in to your AWS account if you're not already logged in.
- Launch the AWS management console, by browsing to <https://console.aws.amazon.com/>
- Click your user name in the top menu bar and from the resulting menu, click "Security Credentials".



- On the resulting page, select the "Access Keys" sub-menu and the "Create New Access Key" command.

The screenshot shows the AWS IAM Access Keys page. On the left, there's a sidebar with links like Dashboard, Details, Groups, Users, Roles, Identity Providers, Password Policy, and Credential Report. The main area has a title 'Access Keys (Access Key ID and Secret Access Key)'. Below it, a note says: 'You use access keys to sign programmatic requests to AWS services. To learn how to use access keys, see the [signing documentation](#). For your protection, store your access keys securely. In addition, AWS recommends that you rotate your access keys every 90 days.' A note below says: 'Note: You can have a maximum of two access keys (active or inactive) at a time.' At the bottom, there are three tabs: 'Created' (highlighted), 'Deleted', and 'Access Key ID'. A blue button labeled 'Create New Access Key' is visible, with a red box drawn around it. Below the button, a yellow box contains the text: 'Important Change - Managing Your AWS Secret Access Keys'. It states: 'As described in a [previous announcement](#), you cannot retrieve the existing secret access key for your AWS root account, though you can still create a new root access key at any time.' There are also 'Download Key File' and 'Close' buttons.

- AWS will generate a new key pair for your account, which you can see by selecting the "Show Access Key" command in the resulting pop-up window.

The screenshot shows a 'Create Access Key' pop-up window. It displays a success message: 'Your access key (access key ID and secret access key) has been created successfully.' Below it, it says: 'Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.' A note at the bottom says: 'To help protect your security, store your secret access key securely and do not share it.' Under the heading 'Hide Access Key', the 'Access Key ID' and 'Secret Access Key' fields are shown, with a red box highlighting the entire row. At the bottom are 'Download Key File' and 'Close' buttons.

- Note the Access Key ID and Secret Access Key.

You're now ready to connect AWS with Bitnami. To do this:

- Log in to your Bitnami account if you're not already logged in
- Browse to <https://app.bitnamihosting.com>
- Click the "Access Console" button.

The screenshot shows the Bitnami Hosting landing page. The header says 'Bitnami Hosting'. Below it, a text box says: 'Bitnami Cloud Hosting simplifies the process of deploying and managing open source and other applications on the Amazon Cloud. It provides one-click deployment for a vast library of tested, ready-to-run applications, such as Alfresco, Drupal, SugarCRM, Wordpress, Joomla!, Redmine and many others. [Learn more](#)'.

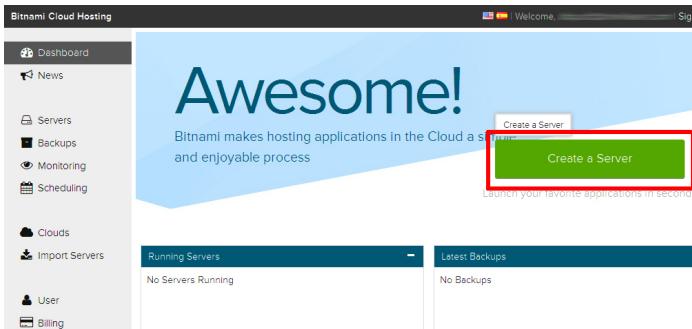
At the bottom, there are two buttons: 'Access Console' (highlighted with a red box) and 'Visit Bitnami'.



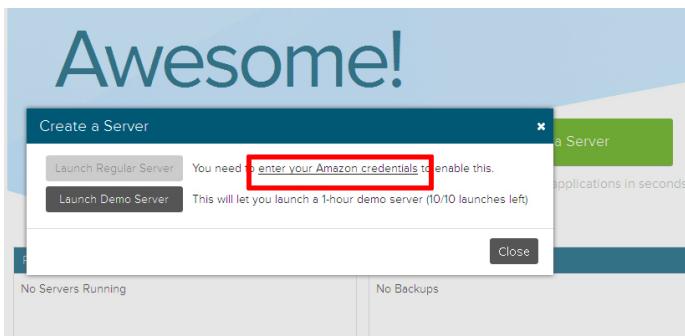
Bitnami Cloud Hosting will recognize your Bitnami credentials and automatically sign you in.

The next step is to set up an administrative password for your Bitnami Cloud Hosting account and connect your AWS account with your Bitnami account.

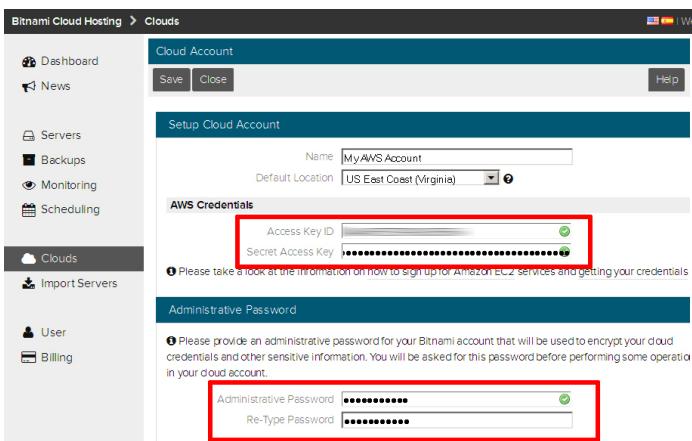
- Click the "Dashboard" menu item in the left sidebar menu, then the big green "Create a Server" button on the resulting page.



- Since this is your first time, you'll be prompted to enter your AWS credentials and an administrative password. You can also later access this screen from the "Clouds" menu item.



- Enter the credentials from the previous step, as well as a hard-to-guess administrative password.



The administrative password offers an additional level of protection against misuse: you'll need to enter it when performing certain operations, such as creating new AWS server instances. Again, make sure you note it down for future reference.

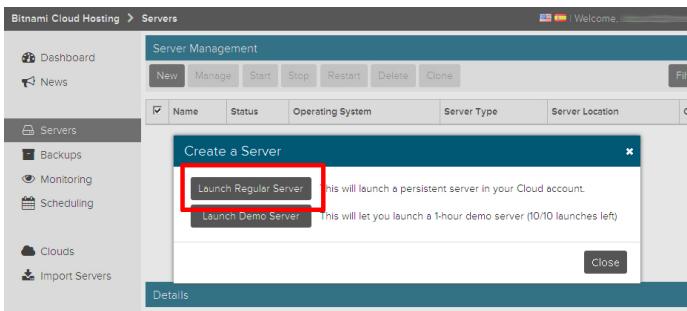
Your AWS and Bitnami accounts will now be connected.

Step 4: Provision an AWS Cloud Server

At the end of this step, your AWS cloud server will be running and you will be able to access it through your Web browser.

To provision your AWS cloud server:

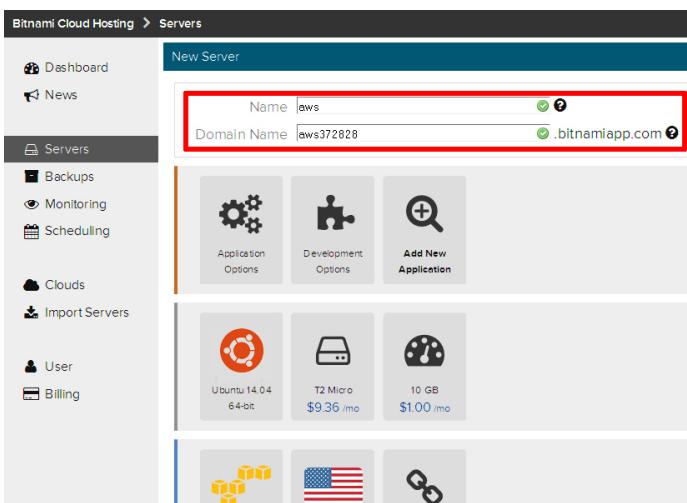
- Select "Servers" in the left sidebar menu.
- On the Server Management screen, click "New" and then "Launch Regular Server" to launch a new server.



- Define a name and domain name for your AWS server. The default server configuration is a "Micro" server with Ubuntu, 1000 MB RAM and 10 GB EBS storage, which is eligible for the AWS free tier.

NOTE

The Bitnami Cloud Hosting dashboard does not yet have the ability to detect whether your AWS servers are in the free tier or not, as Amazon does not provide an API to retrieve this information from AWS. Therefore, it calculates and displays estimated charges for your AWS cloud server as per normal conventions. So long as your usage does not exceed the limits of the AWS free tier, nothing will actually be charged to your credit card and you can safely ignore the estimated charges shown in the Bitnami Cloud Hosting dashboard.



Should you wish to pay for a more sophisticated server, you can choose from a "Small" server (1 dedicated virtual core) all the way up to a "8xLarge" server (32 dedicated virtual cores) depending on the needs of your

application. For more information, refer to [the AWS pricing sheet](#).

TIP

"Micro" servers work just fine for most PHP application development tasks.

- Confirm your selection by hitting the "Build and Launch" button.

Bitnami Cloud Hosting will now begin spinning up the server. The process usually takes a few minutes: a status indicator on the page provides a progress update.

The screenshot shows the Bitnami Cloud Hosting interface. On the left, there's a sidebar with links like Dashboard, News, Servers (which is selected), Backups, Monitoring, Scheduling, Clouds, Import Servers, User, and Billing. The main area is titled 'Server Management' with buttons for New, Manage, Start, Stop, Restart, Delete, and Clone. Below these is a table with columns: Name, Status, Operating System, and Server Type. A single row is selected, showing 'aws' as the name, 'Pending' as the status (highlighted with a red box), 'Ubuntu' as the operating system, and 'T2 Micro' as the server type. There are checkboxes next to each column header.

<input checked="" type="checkbox"/>	Name	Status	Operating System	Server Type
<input checked="" type="checkbox"/>	aws	Pending	Ubuntu	T2 Micro

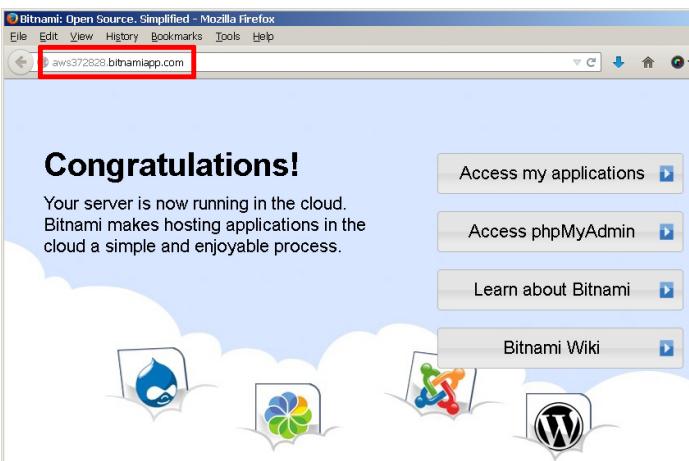
Once the cloud server has been provisioned, the status indicator will show that it's "running", and the "Go to Application" button in the lower panel of the dashboard will become an active link.

This screenshot shows the same interface after the server has been provisioned. The 'Status' column for the 'aws' server now displays a green circle with a white dot, indicating it is 'Running' (highlighted with a red box). The rest of the table and sidebar are identical to the previous screenshot.

<input checked="" type="checkbox"/>	Name	Status	Operating System	Server Type
<input checked="" type="checkbox"/>	aws	Running	Ubuntu	T2 Micro

Below the table, a 'Details' section shows the server's configuration: Applications (None), Latest Backup (Never), and Estimated Cost (\$0.01 hour / \$10.36 month). At the bottom, there are two buttons: 'Go to Application' (highlighted with a red box) and 'Manage Server'.

At this point, you should be able to browse to the cloud server, either by clicking the "Go to Application" button or entering the cloud server host name directly into your browser's address bar. You should see a welcome page like the one below (just so you know, it's served up by Apache, which is part of the Bitnami LAMP Stack).



Once the server is provisioned, you need to gather the security credentials you will need to begin using it. To do this:

- Go back to your Bitnami Cloud Hosting dashboard and in the "Servers" section, select the running server and then click the "Manage" button in the top menu. This will launch the Manage Server screen. Click the "Connect" button.

- In the resulting pop-up window, download the .ppk file which contains the SSH access credentials you will need to connect to the server. Typically, this file is named *bitnami-hosting.ppk*.

- By default, Bitnami Cloud Hosting creates a user account named 'user' and an auto-generated password when a new server is provisioned. You will need this password when accessing Bitnami-supplied applications (including MySQL). Go back to the "Manage Server" screen, look in the "Properties" tab in the lower panel, and display and make a note of the application password.

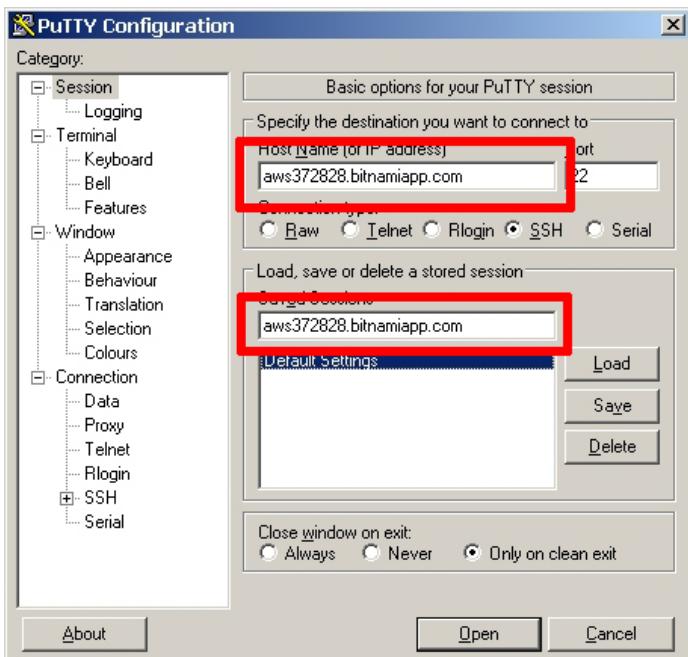
The screenshot shows the Bitnami Cloud Hosting interface with the 'Servers' tab selected. In the center, a 'Manage Server' window for 'aws' is open. The 'Properties' tab is active. Under 'Properties', the 'Status' is listed as 'Running'. Below it, the 'Application Login' field contains the value 'user', which is highlighted with a red rectangular box. The 'Application Password' field is also present but its content is obscured by a red box.

Step 5: Test PHP and MySQL

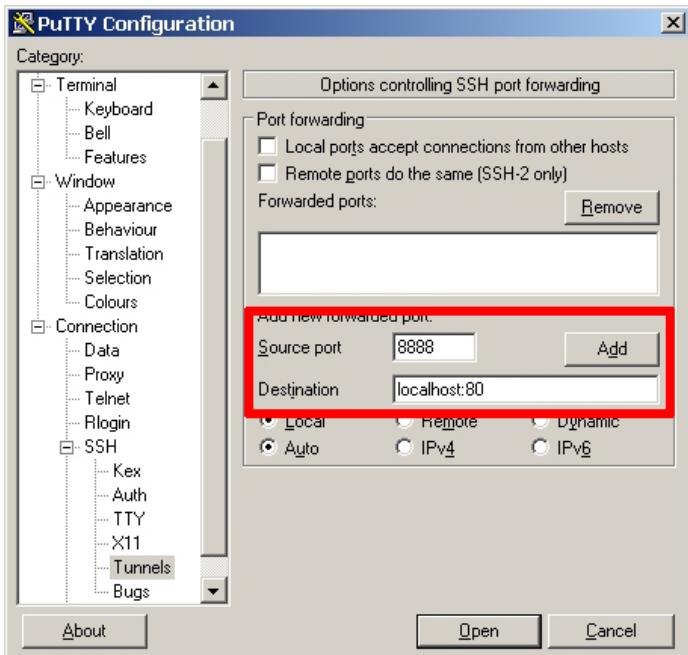
At the end of this step, you will have logged in to your cloud server and verified that PHP, MySQL and phpMyAdmin are working correctly.

You can now connect to the cloud server and test PHP to make sure it's working correctly and has all the extensions you need. The easiest way to do this is with [PuTTY](#), a free SSH client for Windows and UNIX platforms.

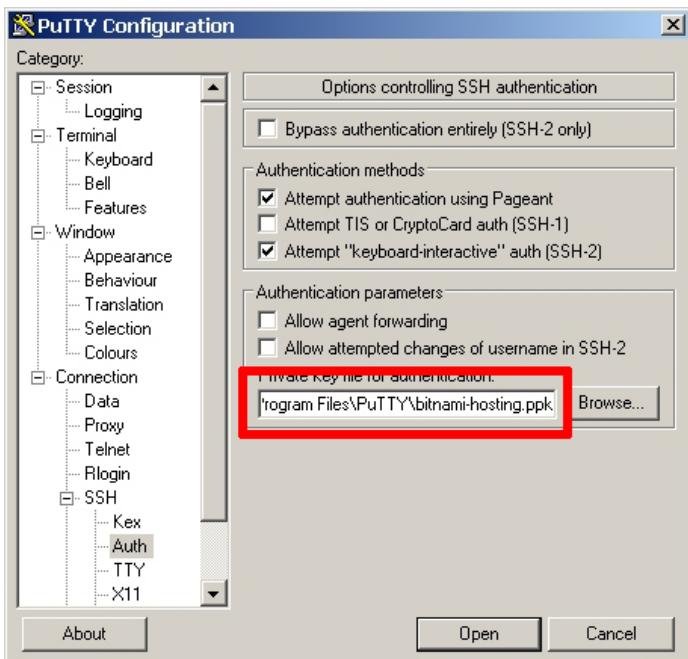
- Download the PuTTY ZIP archive from [its website](#).
- Extract the contents to a folder on your desktop.
- Double-click the *putty.exe* file to bring up the PuTTY configuration window.
- Enter the host name of your cloud server into the "Host Name (or IP address)" field, as well as into the "Saved Sessions" field.
- Click "Save" to save the new session so you can reuse it later.



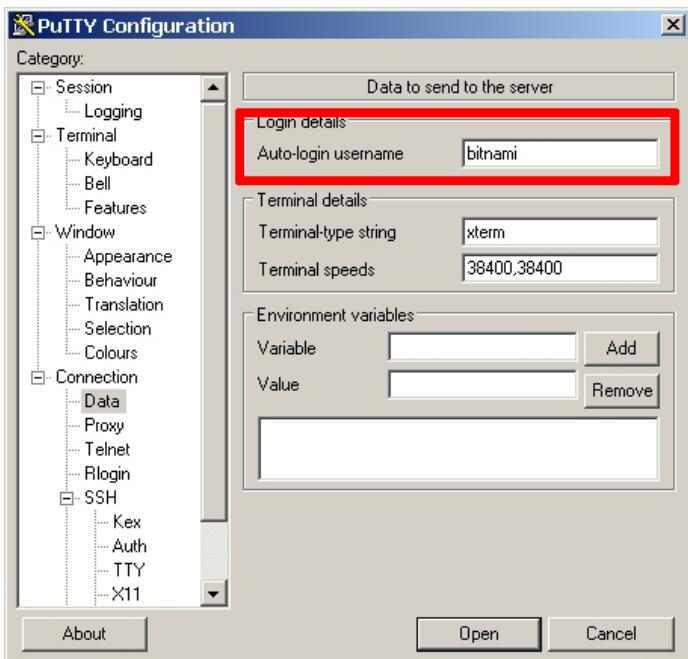
- In the "Connection _ SSH _ Tunnels" section, create a secure tunnel for the phpMyAdmin application by forwarding source port "8888" to destination port "localhost:80".
- Click the "Add" button to add the secure tunnel configuration to the session.



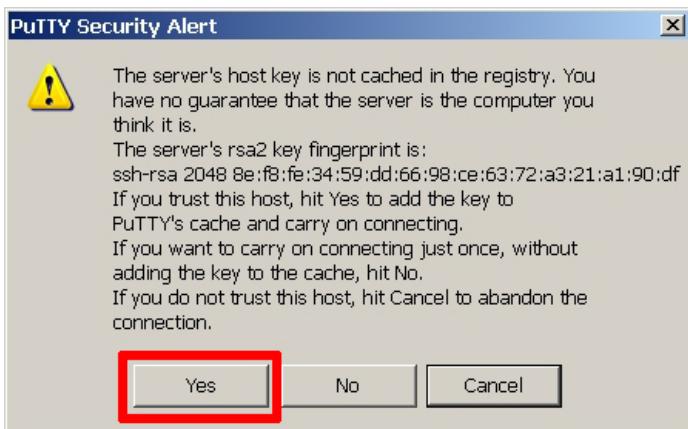
- In the "Connection _ SSH _ Auth" section, select the private key file (*.ppk) you saved in the previous step.



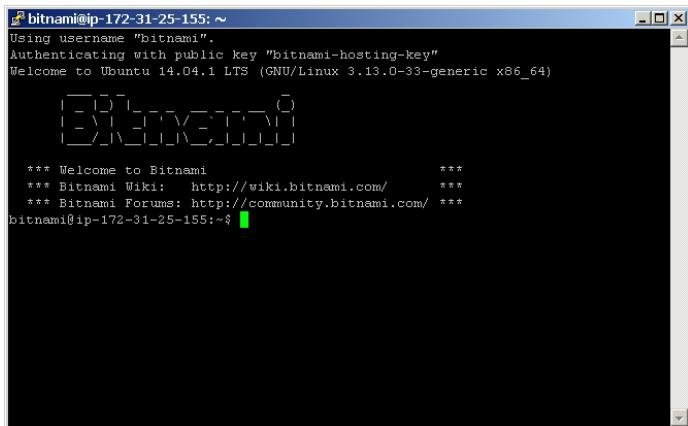
- In the "Connection _ Data" section, enter the username 'bitnami' into the "Auto-login username" field.



- Go back to the "Session" section and save your changes by clicking the "Save" button.
- Click the "Open" button to open an SSH session to the server.
- PuTTY will first ask you to confirm the server's host key and add it to the cache. Go ahead and click "Yes" to this request.



You should now be logged in to your cloud server.



By default, the Bitnami LAMP stack includes running Apache and MySQL servers, and all the packages that come with the stack are located in the `/opt/bitnami` directory. Your first step should be to create a `phpinfo.php` file in the Apache web server root at `/opt/bitnami/apache2/htdocs` directory to verify PHP's capabilities.

```
shell> cd /opt/bitnami/apache2/htdocs  
shell> echo "<?php phpinfo(); ?>" > phpinfo.php
```

Once the file has been copied, browse to `http://[your-cloud-server-hostname]/phpinfo.php` and you should see the output of the `phpinfo()` command.

PHP Version 5.4.31	
System	Linux ip-172-31-25-155 3.13.0-33-generic #58-Ubuntu SMP Tue Jul 29 16:45:05 UTC 2014 x86_64
Build Date	Jul 29 2014 13:34:27
Configure Command	'/configure' '--prefix=/bitnami/megastack-linux-x64/output/php' '--enable-fpm' '--with-fpm-user=daemon' '--with-fpm-group=daemon' '--with-apxs2=/bitnami/megastack-linux-x64/output/apache2/bin/apxs' '--with-com=/bitnami/megastack-linux-x64/output/common' '--with-expat-dir=/bitnami/megastack-linux-x64/output/common' '--with-libxml-dir=/bitnami/megastack-linux-x64/output/common' '--with-zlib-dir=/bitnami/megastack-linux-x64/output/common' '--enable-mbstring=all' '--enable-soap' '--enable-bcmath' '--enable-fp' '--with-xmlrpc' '--enable-fastcgi' '--enable-force-cgi-redirect' '--enable-cgi' '--with-ini-p=/bitnami/megastack-linux-x64/src/imap-2007' '--with-imap-ssl=/bitnami/megastack-linux-x64/output/common' '--with-dom=/bitnami/megastack-linux-x64/output/common' '--with-png-dir=/bitnami/megastack-linux-x64/output/common' '--with-jpeg-dir=/bitnami/megastack-linux-x64/output/common' '--with-openpfd=/bitnami/megastack-linux-x64/output/common' '--with-openssl=/bitnami/megastack-linux-x64/output/common' '--with-ldap=/bitnami/megastack-linux-x64/output/common' '--with-freetype-dir=/bitnami/megastack-linux-x64/output/common' '--enable-calendar' '--enable-ctype' '--enable-pcre' '--enable-session' '--with-regex=php' '--enable-spl' '--enable-zip' '--with-bz2=/bitnami/megastack-linux-x64/output/common' '--enable-sockets' '--with-gmp=/bitnami/megastack-linux-x64/output/common' '--with-xsl=/bitnami/megastack-linux-x64/output/common' '--with-mcrypt=/bitnami/megastack-linux-x64/output/common' '--with-icu-dir=/bitnami/megastack-linux-x64/output/common' '--with-tidy=/bitnami/megastack-linux-x64/output/common' '--with-mysqli=mysqlnd' '--with-mysqlnd=mysqlnd' '--with-pdo-mysql=mysqlnd' '--with-pdo_sqlite=/bitnami/megastack-linux-x64/output/sqlite' '--with-sqlite3=/bitnami/megastack-linux-x64/output/sqlite' '--with-gettext' '--enable-intl' '--with-readline=/bitnami/megastack-linux-x64/output/common'

With this, you know that your PHP installation is configured and working correctly.

You can also check that MySQL is working by launching the MySQL command-line client at the shell prompt.

```
shell> mysql -u root -p
```

When prompted, enter the application password retrieved in the previous step. The client should start up and connect to the local MySQL server, displaying a welcome message as shown below.

```
bitnami@ip-172-31-25-155:~ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.5.38 MySQL Community Server (GPL)

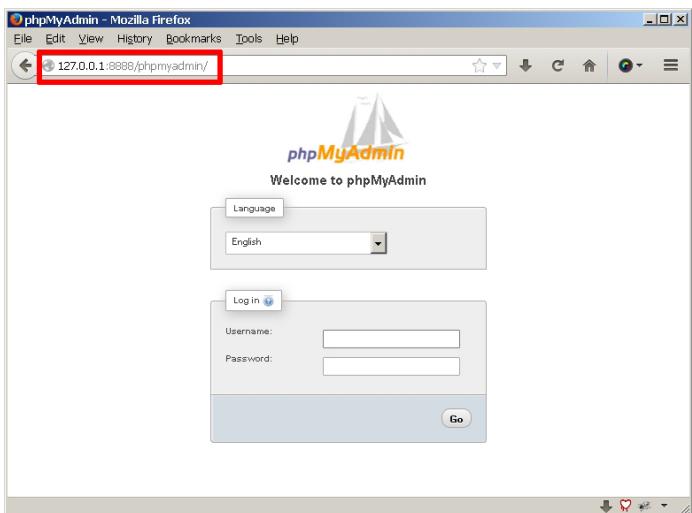
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

You should also be able to access phpMyAdmin through the secure SSH tunnel you created, by browsing to <http://127.0.0.1:8888/phpmyadmin>.



To log in, use username 'root' with the application password from the previous step.

In case you'd like to troubleshoot errors or modify the configuration for Apache, PHP or MySQL - for example, adjusting the maximum upload file size in PHP or changing the path to the MySQL data directory - here are the locations for key configuration and log files in the Bitnami LAMP Stack:

	Configuration file(s)	Log file(s)
Apache	/opt/bitnami/apache2/conf/httpd.conf	/opt/bitnami/apache2/logs/error.log
PHP	/opt/bitnami/php/etc/php.ini	-
MySQL	/opt/bitnami/mysql/my.cnf	/opt/bitnami/mysql/data/mysqld.log

Usually, you'll need to restart your server(s) for your changes to take effect. The Bitnami LAMP Stack includes a control script that lets you easily stop, start and restart Apache, MySQL and PHP. The script is located at `/opt/bitnami/ctlscript.sh`. Call it without any arguments to restart all services:

```
shell> sudo /opt/bitnami/ctlscript.sh restart
```

Or use it to restart a specific service only by passing the service name as argument - for example 'mysql':

```
shell> sudo /opt/bitnami/ctlscript.sh restart mysql
```

```

bitnamix@ip-172-31-25-155:~$ sudo /opt/bitnami/ctlscript.sh restart
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
140924 08:00:02 mysqld_safe mysqld from pid file /opt/bitnami/mysql/data/mysqld.pid ended
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
140924 08:00:06 mysqld_safe Logging to '/opt/bitnami/mysql/data/mysqld.log'.
140924 08:00:06 mysqld_safe Starting mysqld.bin daemon with databases from /opt/bitnami/mysql/data
/opt/bitnami/mysql/scripts/ctl.sh : mysql started at port 3306
[24-Sep-2014 08:00:13] NOTICE: configuration file /opt/bitnami/php/etc/php-fpm.conf test is successful

/opt/bitnami/php/scripts/ctl.sh : php-fpm started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started
bitnamix@ip-172-31-25-155:~$ sudo /opt/bitnami/ctlscript.sh restart mysql
140924 08:03:43 mysqld_safe mysqld from pid file /opt/bitnami/mysql/data/mysqld.pid ended
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
140924 08:03:47 mysqld_safe Logging to '/opt/bitnami/mysql/data/mysqld.log'.
140924 08:03:47 mysqld_safe Starting mysqld.bin daemon with databases from /opt/bitnami/mysql/data

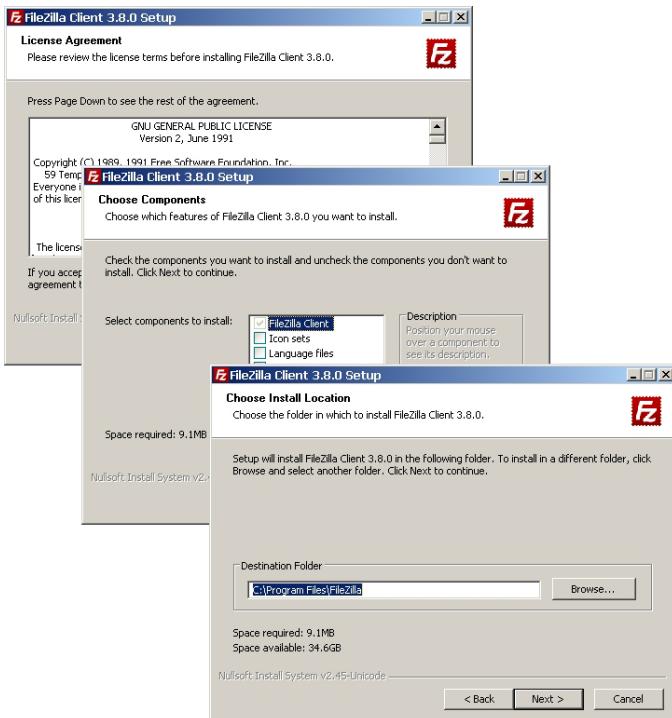
```

Step 6: Deploy the XAMPP Application to the Cloud Server

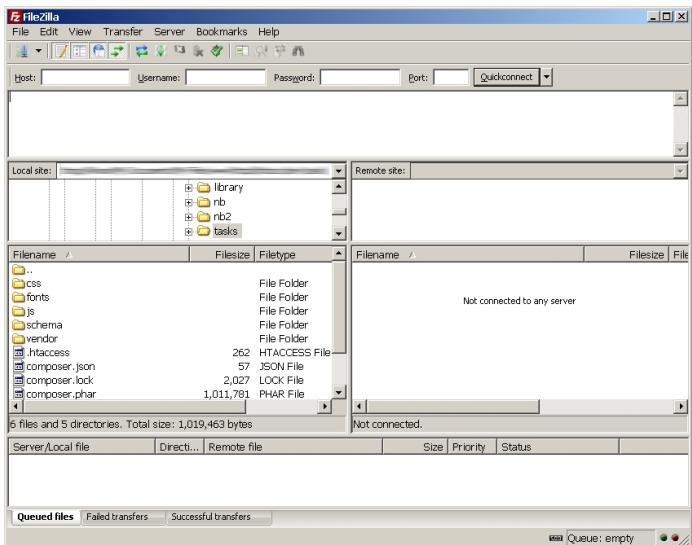
At the end of this step, your PHP/MySQL application will be running in the cloud.

Your cloud server is now provisioned, secured and has a functional PHP/MySQL environment. All that's left is for you to transfer your application code from your local XAMPP environment to your cloud server and set up the database.

The easiest way to transfer files to the server is with FTP or SFTP. Although you can use any FTP/SFTP client, I like [FileZilla](#), a cross-platform, open source and feature-rich client. Download it from [the FileZilla website](#) and install it using the automated installer - it's a quick process, only requiring you to agree to the license, choose the components (the default selection is usually fine) and specify the installation directory.

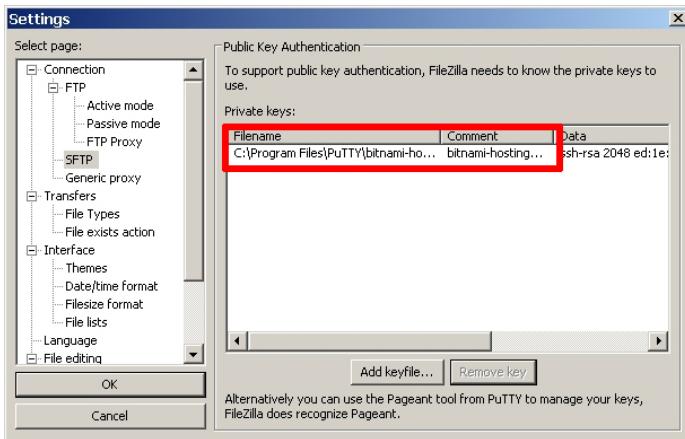


Once FileZilla is installed, launch it and you'll arrive at the main split-screen interface, one side for your local directories and the other for remote directories.

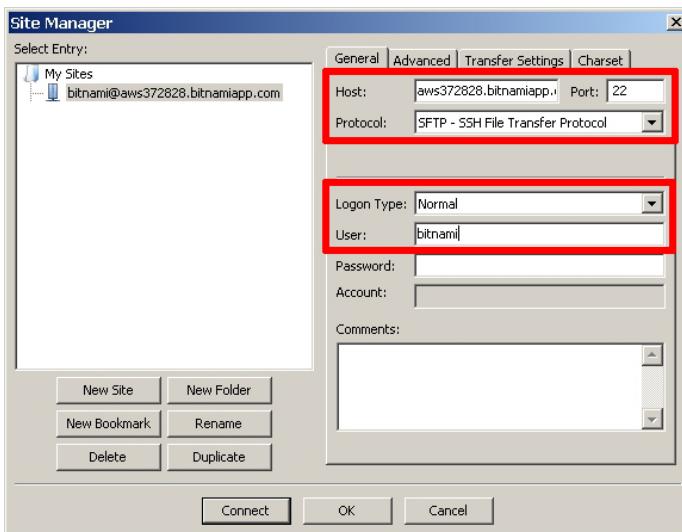


To connect to the cloud server and deploy your application, follow these steps:

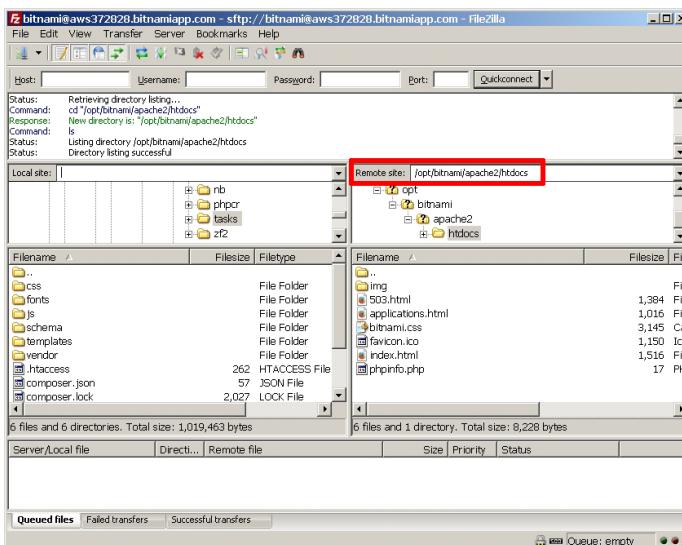
- Use the "Edit _ Settings" command to bring up FileZilla's configuration settings.
- Within the "Connection _ SFTP" section, use the "Add keyfile" command to select the private key file for your server. FileZilla will use this private key to log in to the cloud server.



- Use the "File _ Site Manager _ New Site" command to bring up the FileZilla Site Manager, where you can set up a connection to your cloud server.
- Enter your server host name or IP address and user name.
- Select "SFTP" as the protocol and "Normal" as the logon type.



- Use the "Connect" button to connect to the cloud server and begin an SFTP session.
- On the remote server side of the window, change to the `/opt/bitnami/apache2/htdocs` directory
- On the local server side of the window, change to the directory containing your application code.
- Upload your XAMPP application code to the remote directory by dragging and dropping the files from the local server to the cloud server (you can back up the original contents of the directory if you wish, by downloading them first).

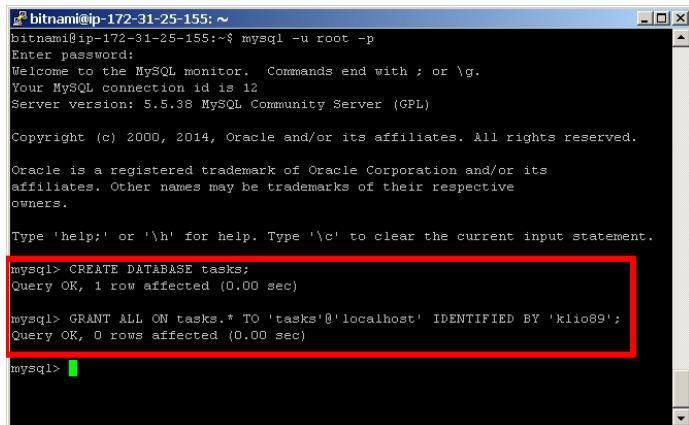


- Once the files are transferred, log in to the server console using PuTTY.
- Create a database for the application using the MySQL command-line client (you can use phpMyAdmin if you prefer a graphical interface). For example, since the application is a to-do list, let's call the database 'tasks'.

```
mysql> CREATE DATABASE tasks;
```

- Follow best practices and create a separate MySQL user with privileges to access only this database.

```
mysql> GRANT ALL ON tasks.* TO 'tasks'@'localhost' IDENTIFIED BY 'klio89';
```



```
bitnami@ip-172-31-25-155:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.5.38 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

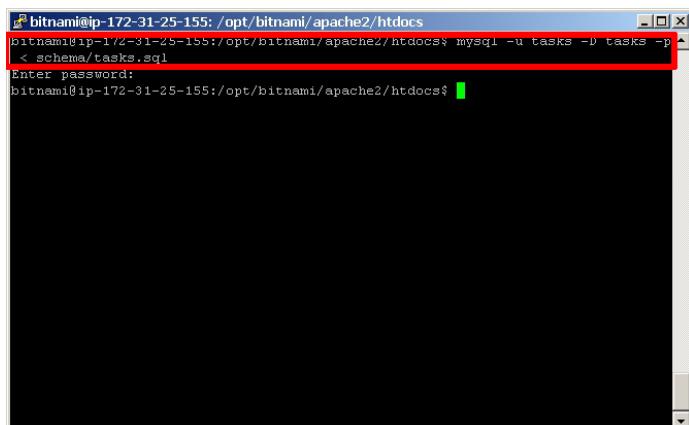
mysql> CREATE DATABASE tasks;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL ON tasks.* TO 'tasks'@'localhost' IDENTIFIED BY 'klio89';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

- If required, update database credentials in your application. Then, install the application schema in the new database (assuming you already uploaded it with the application code). For example, you can use the following command with the MySQL command-line client:

```
shell> mysql -u tasks -D tasks -p < schema/tasks.sql
```



```
bitnami@ip-172-31-25-155:/opt/bitnami/apache2/htdocs$ mysql -u tasks -D tasks -p < schema/tasks.sql
Enter password:
bitnami@ip-172-31-25-155:/opt/bitnami/apache2/htdocs$
```

If you're logged in to phpMyAdmin, you can also import the database schema from your local XAMPP system. To do this, select the "Import" tab of the phpMyAdmin dashboard, select the file containing the schema, and click "Go" to have the tables created in your selected database.



Importing into the current server

File to Import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **[format] [compression]**. Example: **.sql.zip**

Browse your computer: No file chosen (Max: 80MiB)

Character set of the file:

Partial Import:

Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This rarely happens with large files, however it can break transactions.)

Skip this number of queries (for SQL) or lines (for other formats), starting from the first one:

Format:

You can also [learn more about using phpMyAdmin to back up and restore databases](#).

Browse to your cloud server's host name and your application should be active. Here are a few screenshots of the example to-do list application running on the cloud server.

My Tasks

My Tasks

Congratulations! You've successfully deployed your XAMPP application in the cloud.

Understand the Bitnami Cloud Hosting Dashboard and Monitoring Tools

To help you get the most out of your cloud server, Bitnami makes a number of administrative and monitoring tools available. These help you keep track of your application's performance in the cloud and optimize your server and software configuration as needed.

To see these tools in action:

- Browse to the Bitnami Cloud Hosting dashboard and sign in if required using your Bitnami account.
- Select the "Servers" menu item.
- Select your cloud server from the resulting list and click the "Manage" button.

- Click the "Monitoring" tab in the lower pane.

The screenshot shows the 'Monitoring' tab of the Bitnami Cloud Hosting interface. It displays system performance metrics: CPU at 0%, MEM at 1.9%, and DISK at 48%. Below these are service status indicators for Apache, Fpm, and MySQL, all marked as 'Running'. The sidebar on the left includes links for Dashboard, News, Servers (selected), Backups, Monitoring, Scheduling, Clouds, Import Servers, User, and Billing.

This tab provides an overview of CPU, memory and disk usage over the last 3 hours or 24 hours. You can also use the toolbar in the upper pane to restart, shut down, delete or clone the cloud server.

You can also use the Bitnami Cloud Hosting dashboard to resize your virtual server. To do this:

- Select the "Resize" button in the toolbar in the upper pane.

The screenshot shows the 'Servers' page for the 'aws' server. The 'Resize' button in the toolbar is highlighted with a red box. Other buttons visible include 'Assign IP', 'Connect', and 'System Log'.

- Select a new "Server Type" and "Disk Size" on the resulting page.

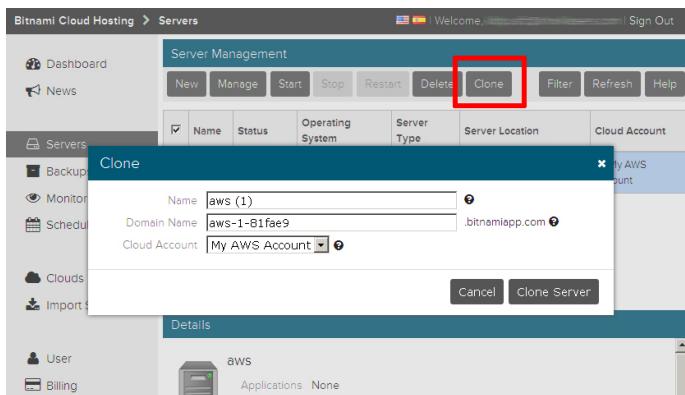
The screenshot shows the 'Resize Server' dialog box. The 'Server Type' dropdown is set to 'M3 Large' and the 'Disk Size' dropdown is set to '270 GB'. Both dropdowns are highlighted with red boxes. At the bottom of the dialog are 'Cancel' and 'Resize' buttons.

- Click the "Resize" button at the bottom of the page.

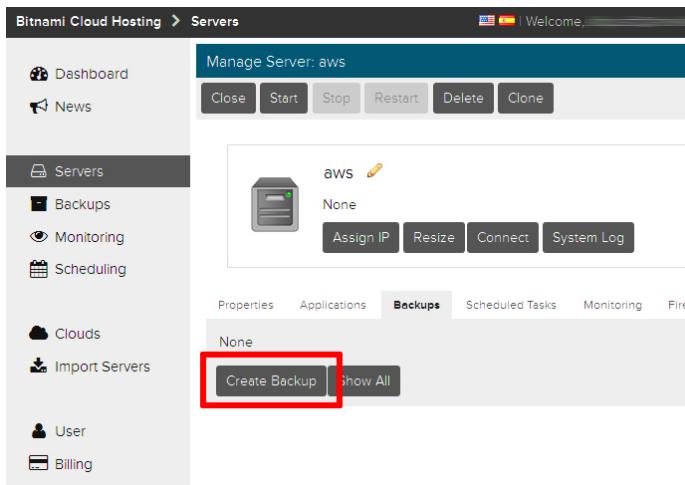
Bitnami Cloud Hosting will resize and restart your cloud server with the new configuration.

Finally, Bitnami Cloud Hosting also includes tools to clone and back up your servers.

- Cloning allows you to make an exact copy of your server so that you can experiment with different settings without affecting the original server. You can clone the server by clicking the "Clone" button on the server management screen.



- Backups allows you to recover from accidental or malicious server failures. Backups can be performed ad-hoc or scheduled. You can backup the server by clicking the "Create Backup" button on the "Backups" tab of the server management screen.



Improve Application Performance

Web application performance problems are hard to debug at the best of times, and more so when your server is in the cloud and running a pre-packaged stack. The responsiveness of your application at any given moment depends on numerous factors: server type, network bandwidth, cloud provider load, database load, caching system in use, application code structure, query structure and various other variables.

IMPORTANT

The Bitnami LAMP Stack already uses the [Apache Event MPM](#) and [PHP-FPM](#) for reduced memory usage and an increase in the number of simultaneous requests that the server can handle (more information). It also comes with the [mod_pagespeed Apache module](#) activated to rewrite pages on the fly and improve latency.

If you're finding that your PHP/MySQL application's performance is not up to scratch, here are a few general

tips you can consider:

- The Bitnami LAMP Stack includes [APC](#), a popular PHP bytecode cache. Usually, when a PHP script is executed, the PHP compiler converts the script to opcodes and then executes the opcodes. APC provides a framework for opcode caching, thereby speeding up PHP applications without needing any code changes. Make sure your APC cache has enough memory and a long TTL. [Read more about APC](#) and [how to use APC with PHP and Bitnami](#).
- The Bitnami LAMP Stack also includes the [PHP memcache extension](#). Memcache is a high-performance, distributed memory object caching system. Consider using memcache to store frequently-accessed fragments of data in memory as arrays, thereby reducing the load on your MySQL database server. Read more about [memcache in PHP](#) and [how to use memcache with PHP and Bitnami](#).
- Turn on MySQL's [slow query log](#) and set MySQL's 'long_query_time' variable to a low number. This lets you track which of your queries are performing inefficiently and adjust them, either structurally or by applying table indexes as needed, to improve performance. You can use tools like [mysqldumpslow](#) or [mysql-slow-query-log-visualizer](#) to parse and analyze the slow query logs generated.
- If your application is database-heavy, you'll gain performance by giving the MySQL server more memory. Use a tool like [MySQLTuner](#) to identify which server parameters need tuning, and incrementally make changes to your server's cache and buffers to improve performance. If your tables are all MyISAM, disable InnoDB in your *my.cnf* file to save further memory.
- Unload Apache modules which you don't need to save memory, and adjust the log level to errors only.
- Minify your JavaScript code, and consider using a CDN for static content like images.

Good luck, and happy coding!

Useful Links

- [Amazon Web Services](#)
- [Bitnami Cloud Hosting](#)
- [Bitnami LAMP Stack](#)
- [Bitnami AMP Stack documentation](#)
- [PuTTY](#)
- [FileZilla](#)
- [Example Project \(.zip\)](#)

About the author

[Vikram Vaswani](#) is the founder of Melonfire, an open source software consultancy firm, and the author of seven books on PHP, MySQL and XML development. Read more about him at <http://vikram-vaswani.in/>.