

Overview

This lab introduces you to AWS OpsWorks. In this lab, you will create a PHP web application server stack and deploy an PHP application to it.

Topics Covered

By end of this lab, you will be able to:

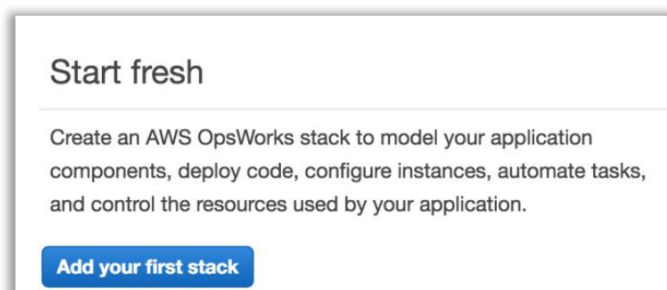
- Create a new AWS OpsWorks stack.
- Deploy an app on this stack.

Add Your First Stack

A stack orchestrates all EC2 instances and other AWS resources you need to run your apps. You usually have a stack per production/app and stage.

1. In the AWS Management Console, click [OpsWorks](#) under [Management Tools](#).

2. Click [Add your first stack](#).



3. Click the [Chef 11 stack](#) box.

Add stack

Which type of stack do you want to create?

A stack is a set of layers, instances and related AWS resources whose configuration you want to manage together.

- Sample stack**
Explore AWS OpsWorks with a sample Node.js app
- Chef 12 stack**
Bring your own cookbooks and use community cookbooks
- Chef 11 stack**
Use built-in cookbooks for applications and deployments

4. In the [Stack name](#) box, type a name, such as [MyStack](#).

→ The other settings deal with the location your EC2 instances, which operating system is used, instance specifics, security settings, and the configuration management. All these settings can be left at their defaults.

5. To create the stack, click [Add Stack](#).

Create a stack with instances that run Linux and Chef 11.10

Classic experience. Use our built-in cookbooks for layers, applications & deployments to get started. Use your own Chef cookbooks to override or extend the built-in layers. [Learn more](#).

Stack name

Region

VPC

Default subnet

Default operating system [Need a different OS? Let us know.](#)

Default SSH key

Chef version

Use custom Chef cookbooks ☐ No [Define the source of your Chef cookbooks](#)

Stack color ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ [Select a color for your stacks to distinguish between different environments like test, staging and production.](#)

[Advanced »](#)

[Cancel](#) [Add stack](#)

6. When the process is completed, you will be directed to the [Stack](#) dashboard, with you new stack displayed.

7. Click [Add a layer](#) to add your first layer.

MyStack

Run CommandStack SettingsDelete Stack

A stack represents a collection of EC2 instances and related AWS resources that have a common purpose and that you want to manage collectively. Within a stack, you use layers to define the configuration of your instances and use apps to specify the code you want to deploy. [Learn more.](#)

Congratulations! Your stack was created.

Next step: [Add a layer.](#)

8. In the Layer type drop-down box, select **PHP App Server**.
9. Click **Add Layer**.

Add layer

OpsWorksECSRDS

Layer type

PHP App Server

The PHP Application Server layer is a blueprint for instances that function as PHP application servers. The supported versions depend on the operating system. [Learn more.](#)

Elastic Load Balancer

No ELBs have been created in your vpc-32ef1356 in us-west-2. To add an ELB go to the [EC2 console](#).

Need further support? [Let us know.](#)


CancelAdd layer

After you finish adding the layer, OpsWorks displays the **Layers** page


10. In the **PHP App Server** row, click **Add Instance**.

Layers

Add layer

 **PHP App Server**

[Settings](#) [Recipes](#) [Network](#) [EBS Volumes](#) [Security](#)

 Delete

No instances

Add instance

+ Layer

11. For this lab, just accept the default settings and click [Add Instance](#) to add the instance to the layer.

12. Find [php-1](#) under [PHP App Server](#) and click [start](#) in the row's [Actions](#) column to start the instance.

→ It typically takes several minutes to boot the EC2 instance and install the packages. Once you get the status as [online](#), this means that your instance is ready for use.

Hostname	Status	Size	Type	AZ	Public IP	Actions
php-app1	online	c3.large	24/7	us-west-2a	54.218.88.250	stop ssh

Deploying an app to your OpsWorks stack

In this lab, you will manually deploy an example app from a public Git repository to an application server.

13. In the left navigation menu, click [Apps](#).
14. On the [Apps](#) page, click [Add an app](#).
15. On the [Add App](#) page, specify the following values:
 - The app's [Name](#), which OpsWorks uses for display purposes.
Type [SimplePHPApp](#).
 - The app's [Type](#), which determines where to deploy the app. Leave it to PHP, which deploys the app to PHP App Server instances.
 - [Data Source Type](#), an associated database server, if any. Select [None](#).
 - The app's [Repository Type](#). The example app is stored in a [Git](#) repository.
 - The app's [Repository URL](#). Enter the example app's repository URL as shown here: <https://github.com/aws-labs/opsworks-demo-php-simple-app>
 - The app's [Branch/Revision](#). Enter [version 1](#) for this revision of the app.

→ Keep the default values for the remaining settings and click [App](#).

16. To install the code on the server, you must [deploy](#) the app. To do so, click [deploy](#) in the SimplePHPApp [Actions](#) column.

17. On the [Deploy App](#) page, [Command](#) should already be set to [Deploy](#). Keep the defaults for the other settings and click [Deploy](#).

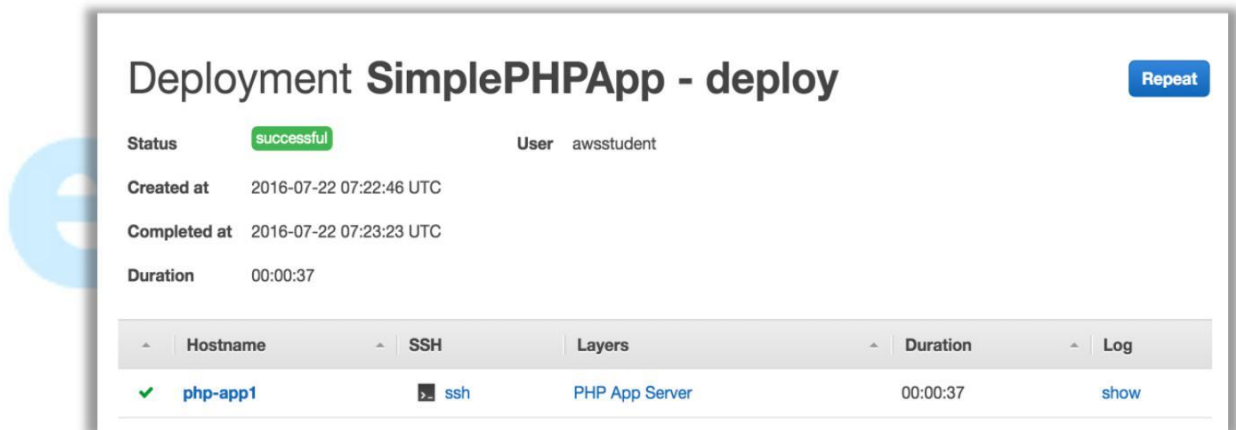
The screenshot shows the 'Deploy App' dialog box. The title is 'Deploy App'. Below the title is a 'Settings' section. In the 'Settings' section, there are three rows: 'App' with the value 'SimplePHPApp', 'Command' with a dropdown menu showing 'Deploy', and 'Comment' with a text box containing 'Optional'. Below the 'Settings' section is an 'Advanced' section with a blue arrow icon. Below the 'Advanced' section is an 'Instances' section with a blue arrow icon. Below the 'Instances' section is a message: 'OpsWorks will run this command on 1 of 1 instances. The assigned recipes are run on all selected instances.' Below the message is an 'Advanced' section with a blue arrow icon. At the bottom right of the dialog box are two buttons: 'Cancel' and 'Deploy'.

→ When the deployment is complete, the [Deployment](#) page displays a [Status](#) of [successful](#), and php-app1 will have a green check mark next to it.

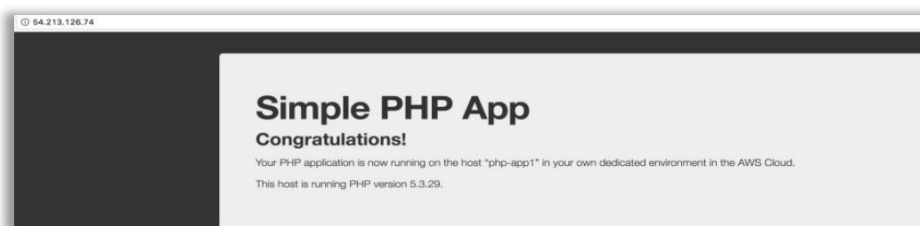


18. SimplePHPApp is now installed and ready to go. To run it, click [Instances](#) in the left Navigation menu.

19. Then click the php-app1 instance's [Public IP](#) address.



→ The web page will display a simple message of congratulations.



Conclusion

Congratulations! You have now successfully:

- Created a new AWS OpsWorks stack.
- Deployed an app to an AWS OpsWorks stack file.