Creating a Stack in OpsWorks & Deploying an App in a Stack

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

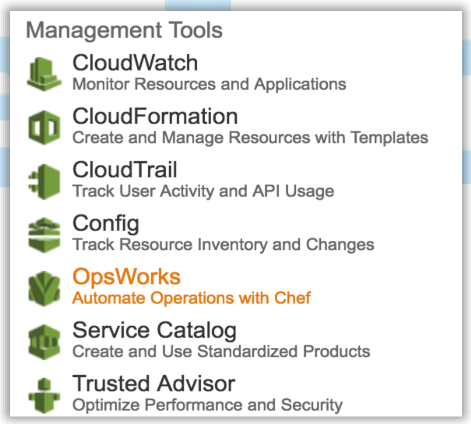
By end of this lab, you will be able to:

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

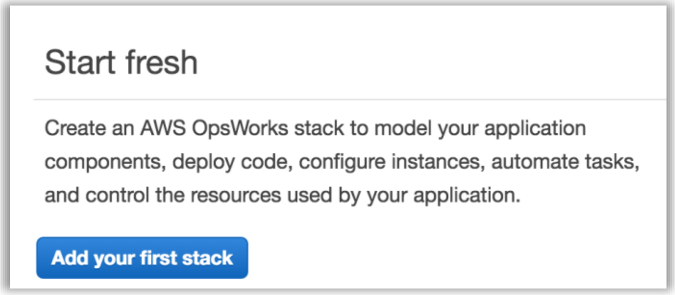
Add Your First Stack

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.

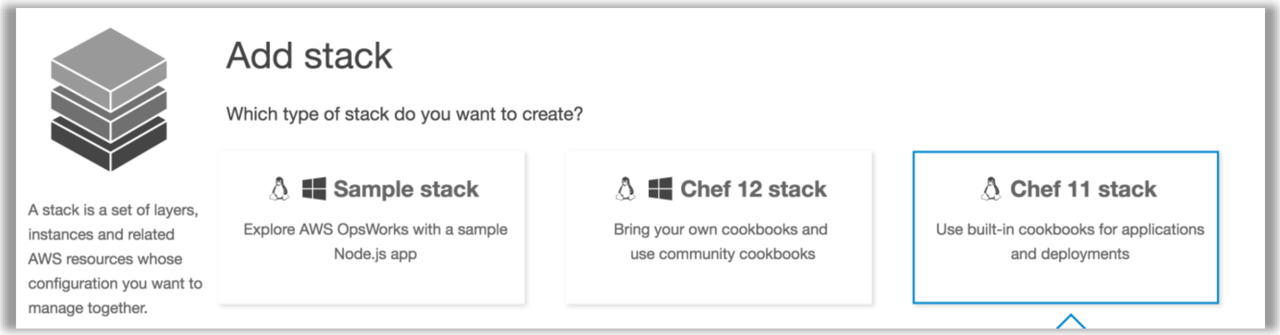

Management Tools
CloudWatch
Monitor Resources and Applications
CloudFormation
Create and Manage Resources with Templates
CloudTrail
Track User Activity and API Usage
Config
Track Resource Inventory and Changes
OpsWorks
Automate Operations with Chef
Service Catalog
Create and Use Standardized Products
Trusted Advisor
Optimize Performance and Security

2. Click Add your first stack.


Start fresh

Create an AWS OpsWorks stack to model your application components, deploy code, configure instances, automate tasks, and control the resources used by your application.

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

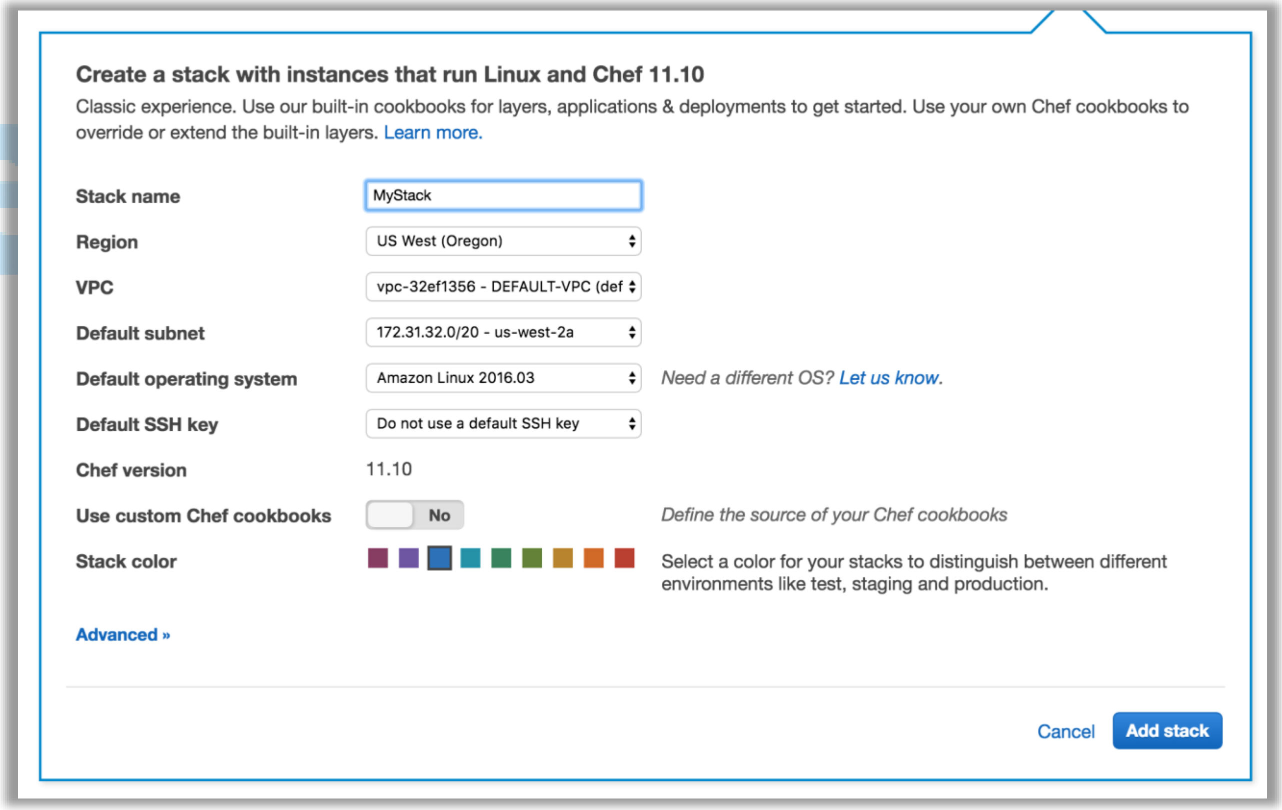
Chef 12 stack
Bring your own cookbooks and use community cookbooks

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.



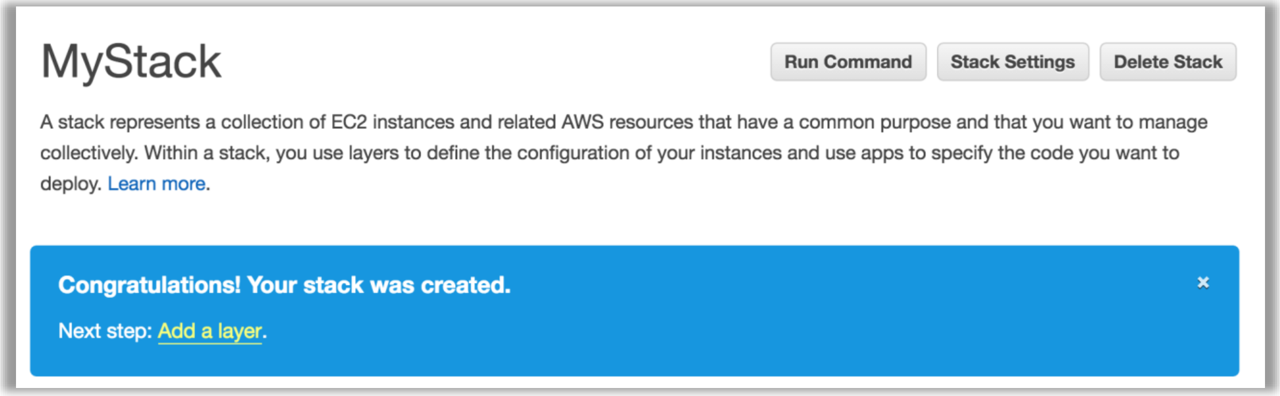
Create a stack with instances that run Linux and Chef 11.10

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	MyStack	
Region	US West (Oregon)	
VPC	vpc-32ef1356 - DEFAULT-VPC (def	
Default subnet	172.31.32.0/20 - us-west-2a	
Default operating system	Amazon Linux 2016.03	*Need a different OS? Let us know.*
Default SSH key	Do not use a default SSH key	
Chef version	11.10	
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.

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.



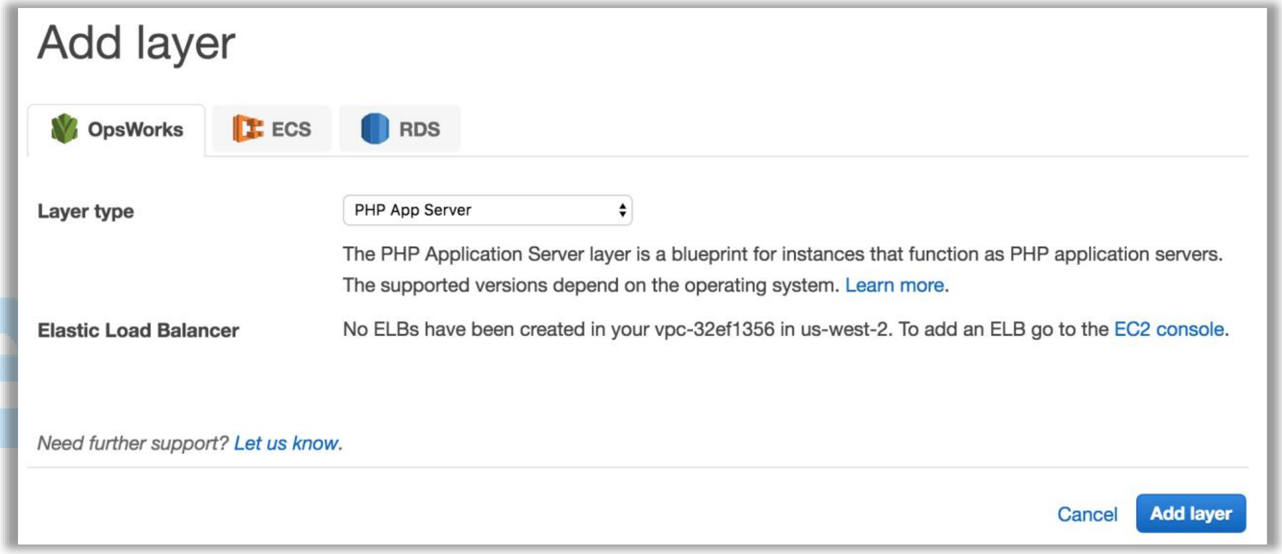
MyStack

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

Layer type: PHP App Server

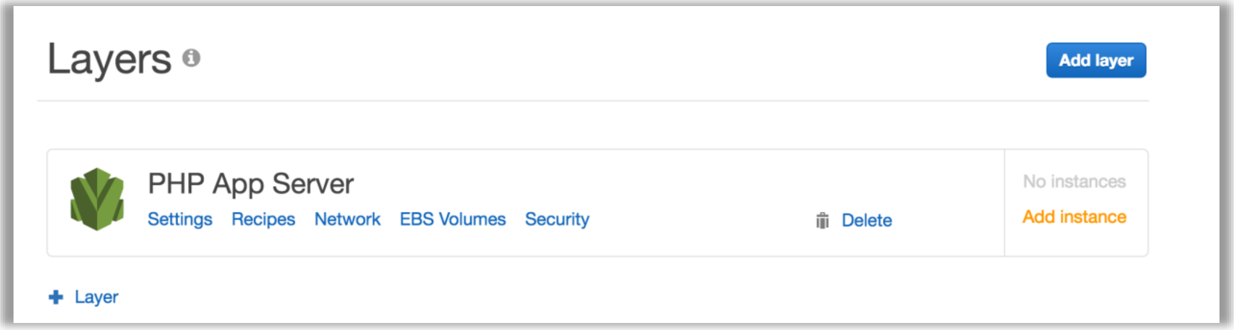
The PHP Application Server layer is a blueprint for instances that function as PHP application servers. The supported versions depend on the operating system. Learn more.

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.

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

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

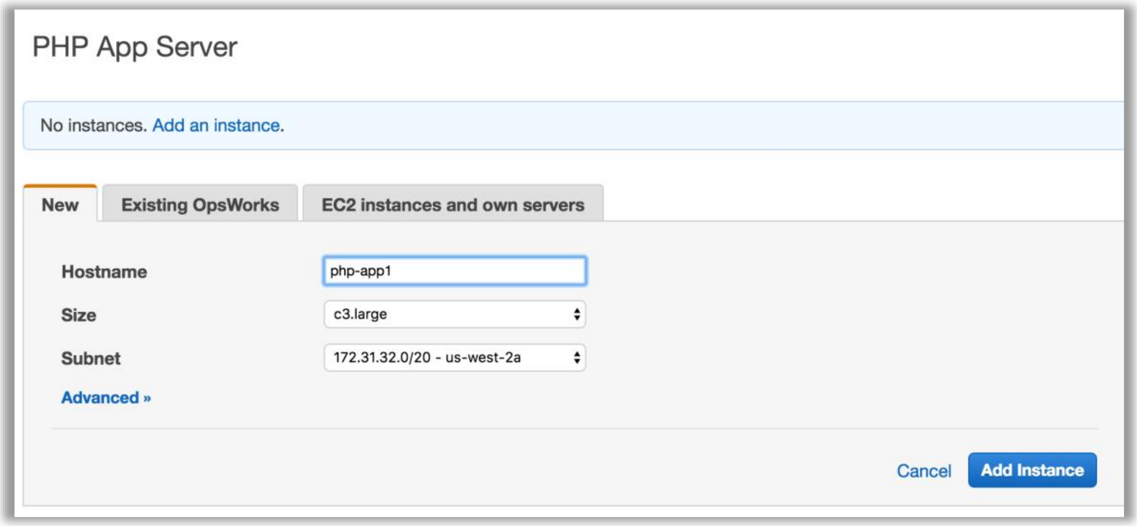


Layers

PHP App Server

No instances

11. For this lab, just accept the default settings and click Add Instance to add the instance to the layer.

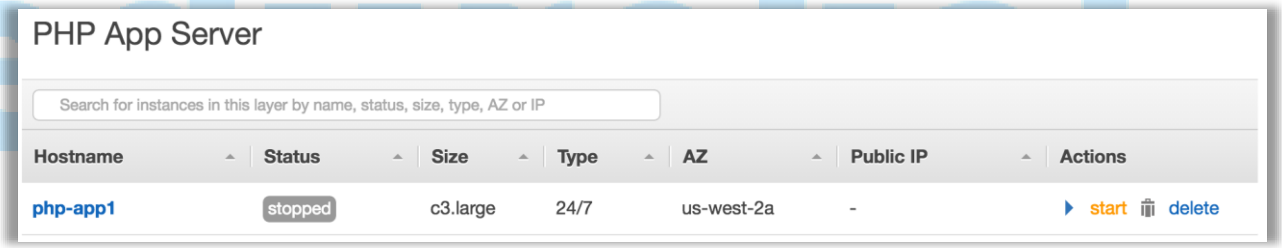


PHP App Server

No instances. Add an instance.

Hostname	php-app1
Size	c3.large
Subnet	172.31.32.0/20 - us-west-2a

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

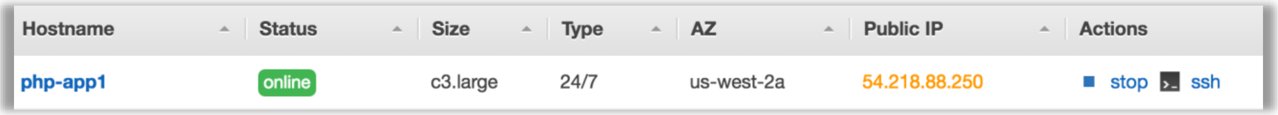


PHP App Server

Search for instances in this layer by name, status, size, type, AZ or IP

Hostname	Status	Size	Type	AZ	Public IP	Actions
php-app1	stopped	c3.large	24/7	us-west-2a	-	start delete

→ 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.

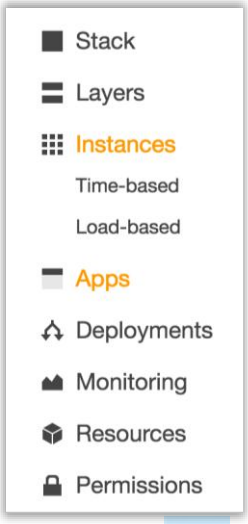


Hostname	Status	Size	Type	AZ	Public IP	Actions
php-app1	online	c3.large	24/7	us-west-2a	54.218.88.250	stop ssh

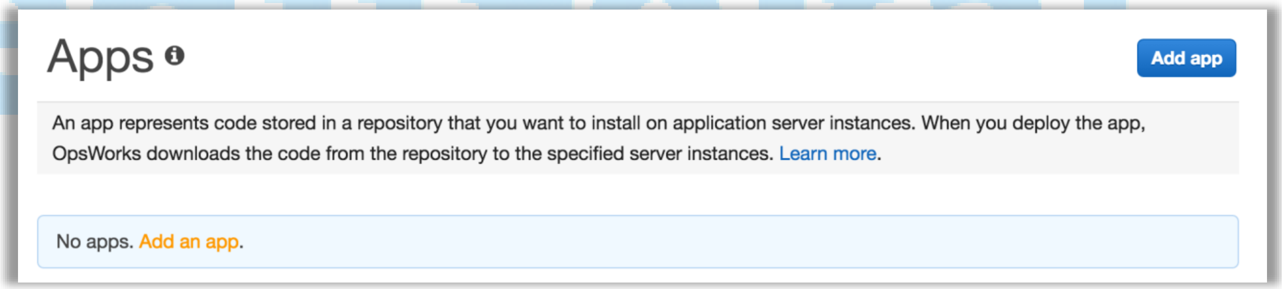
Deploying an app to you OpsWorks stack

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.



Apps

Add app

An app represents code stored in a repository that you want to install on application server instances. When you deploy the app, OpsWorks downloads the code from the repository to the specified server instances. Learn more.

No apps. 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.

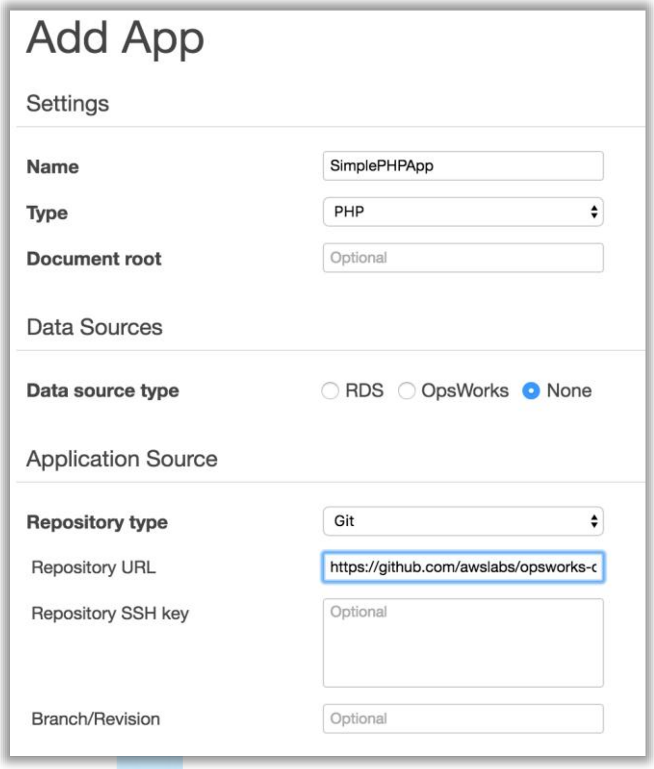
→ Data Source Type, an associated database server, if any. Select None.

→ 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/awslabs/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.



Add App

Settings

Name	SimplePHPApp
Type	PHP
Document root	Optional

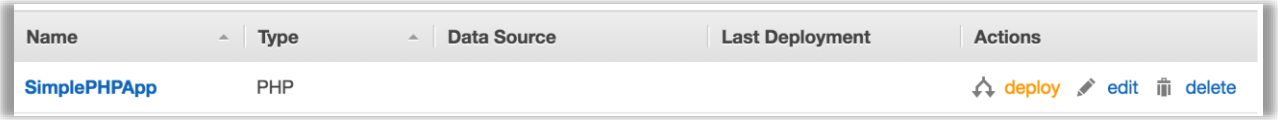
Data Sources

Data source type ○ RDS ○ OpsWorks ● None

Application Source

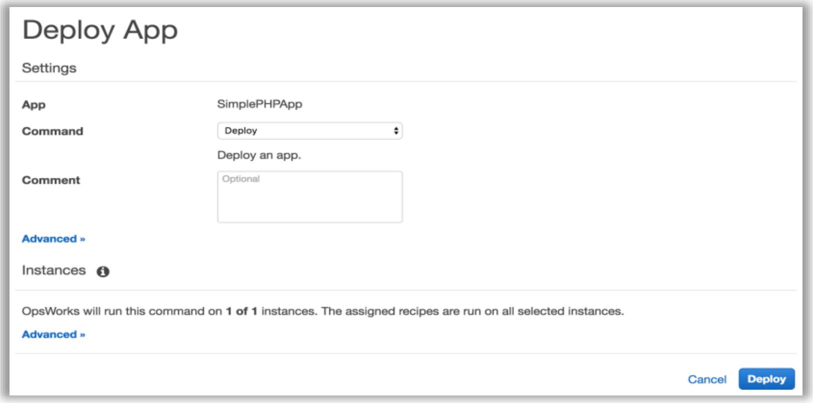
Repository type	Git
Repository URL	https://github.com/awslabs/opsworks-c
Repository SSH key	Optional
Branch/Revision	Optional

16. To install the code on the server, you must *deploy* the app. To do so, click deploy in the SimplePHPApp Actions column.



Name	Type	Data Source	Last Deployment	Actions
SimplePHPApp	PHP			deploy edit delete

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



Deploy App

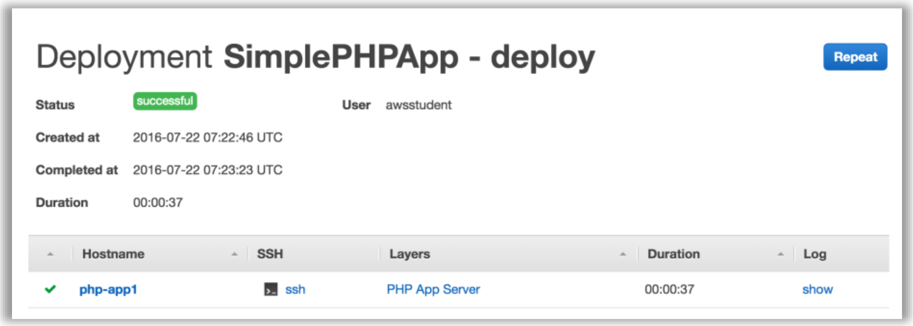
Settings

App	SimplePHPApp
Command	Deploy Deploy an app.
Comment	Optional

Instances

OpsWorks will run this command on **1** of **1** instances. The assigned recipes are run on all selected instances.

→ 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.



Deployment **SimplePHPApp - deploy**

Status successful **User** awsstudent

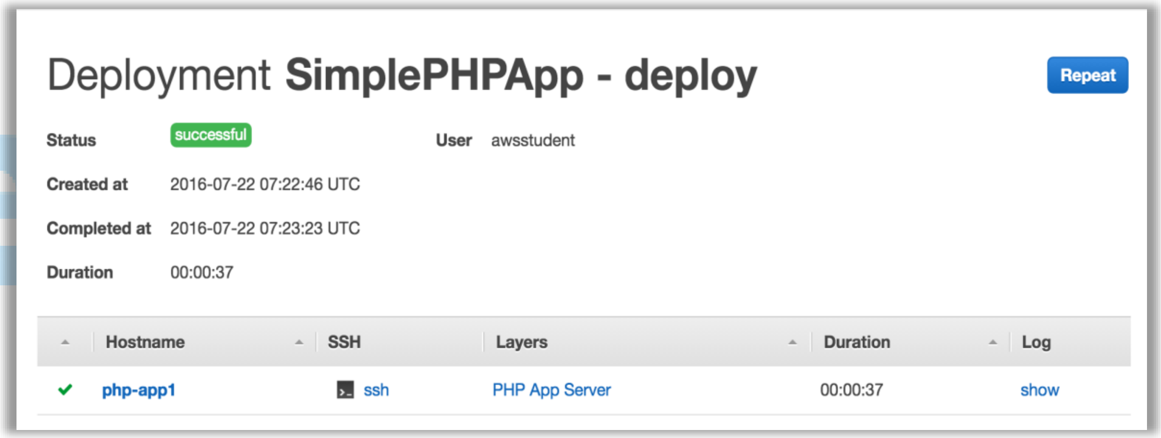
Created at 2016-07-22 07:22:46 UTC

Completed at 2016-07-22 07:23:23 UTC

Duration 00:00:37

	Hostname	SSH	Layers	Duration	Log
✓	php-app1	ssh	PHP App Server	00:00:37	show

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.



Deployment **SimplePHPApp - deploy**

Status successful **User** awsstudent

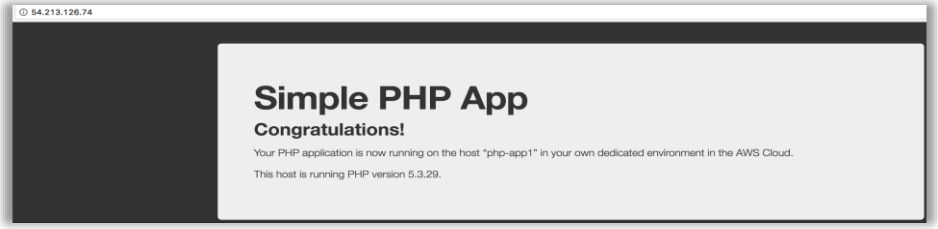
Created at 2016-07-22 07:22:46 UTC

Completed at 2016-07-22 07:23:23 UTC

Duration 00:00:37

	Hostname	SSH	Layers	Duration	Log
✓	php-app1	ssh	PHP App Server	00:00:37	show

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



Simple PHP App

Congratulations!

Your PHP application is now running on the host "php-app1" in your own dedicated environment in the AWS Cloud.

This host is running PHP version 5.3.29.

Conclusion

Congratulations! You have now successfully:

→ Created a new AWS OpsWorks stack.

→ Deployed an app to an AWS OpsWorks stack file.