



Hands-on Creating an OpsWorks Stack

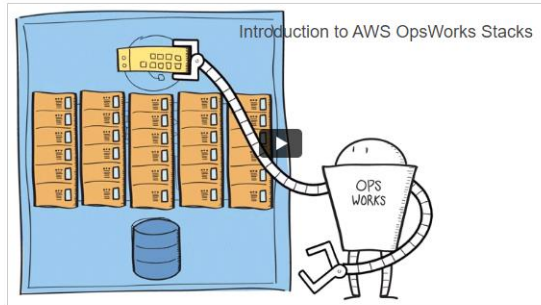
UNIX/Linux Training

support@intellipaat.com
+91-7022374614
US: 1-800-216-8930(Toll Free)

Creating an OpsWorks Stack

Step 1: Go to the OpsWorks console and choose Add your first stack

AWS OpsWorks Stacks is a configuration management service that helps you build and operate highly dynamic applications and propagate changes instantly. [Learn more.](#)



Start fresh

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

[Add your first stack](#)

Register existing instances

Use AWS OpsWorks Stacks with existing **Amazon EC2 instances**, as well as **on-premises servers**, for an easy way to install packages, run scripts, manage operating system users, and more.

[Register instances](#)

Step 2: Go ahead with the Sample stack and hit Create stack

Add stack

Which type of stack do you want to create?



Sample stack

Explore AWS OpsWorks Stacks 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

Create a Chef 12 sample stack with a Node.js app

A Node.js app will be set up to help you explore the features and configuration options of AWS OpsWorks Stacks, for example: layers and lifecycle events. [Learn more.](#)

Operating system type

☒ Linux ☐ Windows

[Cancel](#)

[Create stack](#)

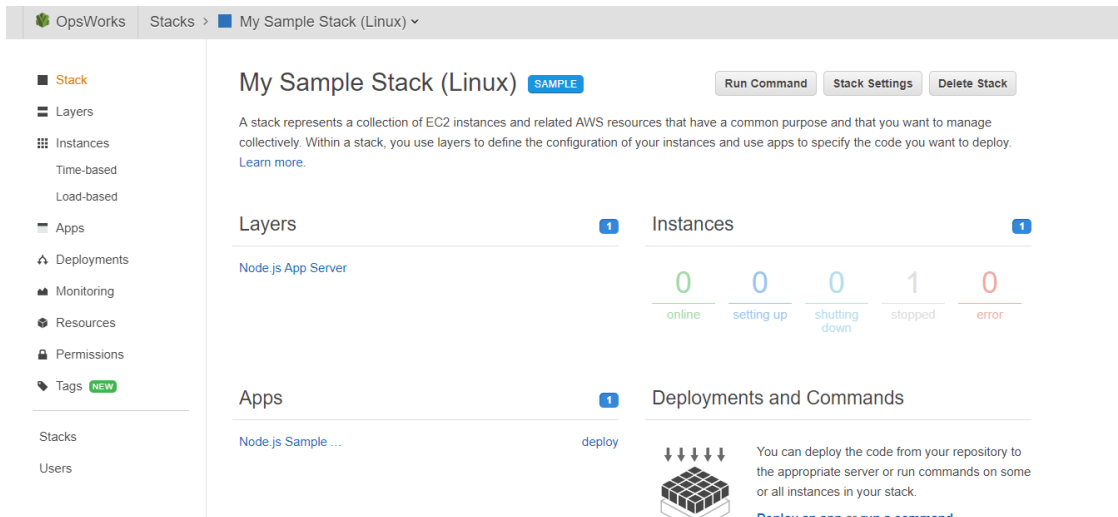
Step 3: Check if all the steps are successful. If all the steps have a green tick, then click Explore the sample stack

Setting up a sample stack

- ✓ 1. Creating a stack named "My Sample Stack (Linux)"
- ✓ 2. Setting the Chef cookbook repository of the stack
- ✓ 3. Creating a layer named "Node.js App Server" in the stack
- ✓ 4. Assigning a recipe to the deploy lifecycle event in the layer
- ✓ 5. Adding an instance to the layer

Explore the sample stack

Step 4: This is how your Dashboard will look



The screenshot shows the AWS OpsWorks console interface for a stack named "My Sample Stack (Linux)". The left sidebar contains a navigation menu with options: Stack, Layers, Instances (Time-based, Load-based), Apps, Deployments, Monitoring, Resources, Permissions, Tags (NEW), Stacks, and Users. The main content area has a header with "OpsWorks" and "Stacks > My Sample Stack (Linux)". Below this, the stack name "My Sample Stack (Linux)" is displayed with a "SAMPLE" badge and buttons for "Run Command", "Stack Settings", and "Delete Stack". A descriptive text explains that a stack represents a collection of EC2 instances and related AWS resources. The dashboard is divided into four sections: "Layers" (showing "Node.js App Server"), "Instances" (showing counts for online, setting up, shutting down, stopped, and error states), "Apps" (showing "Node.js Sample ..."), and "Deployments and Commands" (showing a "deploy" button and a description of deployment options).

Step 5: Now, we must configure the instances. Before starting the instance, change the instance type from t2.medium to t2.micro so that you won't get billed. Click instances then click on the name of the Node.js server.

- Stack
- Layers
- Instances**
 - Time-based
 - Load-based
- Apps
- Deployments
- Monitoring
- Resources
- Permissions
- Tags NEW

- Stacks
- Users

Instances

1 total
0 online
0 setting up
0 shutting down
1 stopped
0 errors

Start All Instances

An instance represents a server. It can belong to one or more layers, that define the instance's settings, resources, installed packages, profiles and security groups. When you start the instance, OpsWorks uses the associated layer's blueprint to create and configure a corresponding EC2 instance. [Learn more.](#)

Node.js App Server

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

Hostname	Status	Size	Type	AZ	Public IP	Actions
nodejs-server1	stopped	t2.medium	24/7	us-east-1a	-	start delete

+ Instance

nodejs-server1

Start Edit Delete

Details

Hostname	nodejs-server1
Status	stopped
Layers	Node.js App Server
EC2 instance ID	-
OpsWorks ID	98e95b04-1c72-46c1-bf4c-818aa45bc602
Instance type	24/7
Size	t2.medium
Subnet	subnet-544d1933 172.31.0.0/20 - us-east-1a
Operating system	Amazon Linux 2
OW Agent version	Inherited from stack
Tenancy	default
Architecture	64bit
Virtualization type	hvm
EBS Optimized	no
Root device type	EBS backed
Root device ID	-

Monitoring

OpsWorks uses CloudWatch metrics to provide detailed [monitoring](#) for your instance.

Volumes

No volumes. [Manage in resources.](#)

Elastic Load Balancing

This instance does not belong to any layers with an ELB attached. [Change layer settings.](#)

Elastic IP

No Elastic IP. [Manage in resources.](#)

Step 6: Click on the edit button on the top-right of the page. Under size, choose t2.micro and hit Save.

Instance nodejs-server1

Hostname	nodejs-server1
Size	t2.micro
SSH key	
Operating system	
Architecture	
Virtualization type	
Root device type	
OpsWorks Agent version	
Layers	

Select the compute and memory size for your instance.

[m5dn.large](#)
[m5dn.xlarge](#)
[m5dn.2xlarge](#)
[m5dn.4xlarge](#)
[m5dn.8xlarge](#)
[m5dn.12xlarge](#)
[m5dn.16xlarge](#)
[m5dn.24xlarge](#)
[m5n.large](#)
[m5n.xlarge](#)
[m5n.2xlarge](#)
[m5n.4xlarge](#)
[m5n.8xlarge](#)
[m5n.12xlarge](#)
[m5n.16xlarge](#)
[m5n.24xlarge](#)
[t2.nano](#)
[t2.micro](#)
[t2.small](#)
[t2.medium](#)

Cancel **Save**

Step 7: Now click the start button. The Status will change from stopped to requested.

nodejs-server1 ●

Start Edit Delete

Details

Hostname	nodejs-server1
Status	stopped
Layers	Node.js App Server
EC2 instance ID	—
OpsWorks ID	98e95b04-1c72-46c1-bf4c-818aa45bc602
Instance type	24/7
Size	t2.micro
Subnet	subnet-544d1933 172.31.0.0/20 - us-east-1a
Operating system	Amazon Linux 2
OW Agent version	Inherited from stack
Tenancy	default

Monitoring

OpsWorks uses CloudWatch metrics to provide detailed [monitoring](#) for your instance.

Volumes

No volumes. [Manage in resources.](#)

Elastic Load Balancing

This instance does not belong to any layers with an ELB attached. [Change layer settings.](#)

nodejs-server1 ●

Stop Edit

Details

Hostname	nodejs-server1
Status	requested
Layers	Node.js App Server
EC2 instance ID	—
OpsWorks ID	98e95b04-1c72-46c1-bf4c-818aa45bc602
Instance type	24/7
Size	t2.micro
Subnet	subnet-544d1933 172.31.0.0/20 - us-east-1a

Monitoring

OpsWorks uses CloudWatch metrics to provide detailed [monitoring](#) for your instance.

Volumes

No volumes. [Manage in resources.](#)

Elastic Load Balancing

Step 8: Wait until the Status changes to Online. Now you can see there is an EC2 instance launched by clicking on the EC2 instance ID. If you go further down, you will get the public IP address of the instance. By running it on your browser, you will be able to view the Node.js application. Use the Public DNS or Public IP.

nodejs-server1

SSH
Run Command
Reboot
Stop
Edit

Details

Hostname	nodejs-server1
Status	online
Layers	Node.js App Server
EC2 instance ID	i-04aa1469d376cb80e
OpsWorks ID	98e95b04-1c72-46c1-bf4c-818aa45bc602
Instance type	24/7
Size	t2.micro
Subnet	subnet-544d1933 172.31.0.0/20 - us-east-1a
Operating system	Amazon Linux 2
Reported OS	Amazon Linux 2
OW Agent version	Inherited from stack
Reported OW Agent	4037 (Jun 4th 2019) Changelog
Tenancy	default
Architecture	64bit
Virtualization type	hvm

Monitoring

OpsWorks uses CloudWatch metrics to provide detailed [monitoring](#) for your instance.

Volumes

No volumes. [Manage in resources.](#)

Elastic Load Balancing

This instance does not belong to any layers with an ELB attached. [Change layer settings.](#)

Elastic IP

No Elastic IP. [Manage in resources.](#)

Network and Security

Public DNS	ec2-34-200-213-241.compute-1.amazonaws.com
Public IP	34.200.213.241
Private DNS	ip-172-31-9-0.ec2.internal
Private IP	172.31.9.0
SSH key	—
SSH RSA fingerprint	61:d5:d0:80:4b:5c:71:33:bb:eb:e2:44:41:d0:1a:f3
SSH DSA fingerprint	7d:2d:67:ca:86:f7:ca:0c:bd:da:5a:ce:cb:3b:bb:7e
Security groups	AWS-OpsWorks-Default-Server AWS-OpsWorks-WebApp

Step 9: This would be the output.

