

# Create a CloudFormation stack

1. Open the CloudFormation Sample Templates page with your region
2. Scroll down, choose Single EC2 Instance with local MySQL database and download the template

The screenshot shows the AWS CloudFormation Sample Templates page for the US West (N. California) region. At the top, there's a navigation bar with links for Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Explore More, Contact Sales, Support, English, My Account, and Sign In to the Console.

The main content area has a heading "AWS CloudFormation Sample Templates - US West (N. California) region". Below it, a paragraph explains that CloudFormation provides an easy way to create collections of related AWS resources. It mentions the AWS Management Console, AWS CLI, AWS CloudFormation APIs, or AWS Toolkits. It notes that templates can be used as-is or as starting points for creating own templates, with sample templates available for download.

Below this, a section titled "Microsoft Windows Server samples" lists four templates:

1	Template URL	Description	Launch in US West (N. California) Region
	<a href="#">Windows_Single_Server_SharePoint_Foundation.template</a>	SharePoint® Foundation 2010 running on Microsoft Windows Server® 2008 R2	<a href="#">Launch Stack</a>
	<a href="#">Windows_Single_Server_Active_Directory.template</a>	Create a single server installation of Active Directory running on Microsoft Windows Server® 2008 R2.	<a href="#">Launch Stack</a>
	<a href="#">Windows_Roles_And_Features.template</a>	Create a single server specifying server roles running on Microsoft Windows Server® 2008 R2.	<a href="#">Launch Stack</a>

A small note at the bottom states: "Microsoft, Windows Server, and SharePoint are trademarks of the Microsoft group of companies."

Further down, a section titled "Open Source Applications" lists two templates:

Description	Template URL	Launch in US West (N. California) Region
Drupal is an open source content management platform powering millions of websites and applications. For more details click <a href="#">here</a> .	<a href="#">Single EC2 Instance with local MySQL database</a>	<a href="#">Launch Stack</a>
Single EC2 Instance web server with Amazon RDS database instance		<a href="#">Launch Stack</a>

The screenshot shows the "Application Framework Examples" section of the AWS CloudFormation Sample Templates page. It includes a heading and a note about provisioning application frameworks for "Hello World" applications.

Below this, a table lists several sample templates:

2	Description	Template URL	Launch in US West (N. California) Region
	A simple LAMP stack running a PHP "Hello World" application.	<a href="#">Single EC2 Instance with local MySQL database</a>	<a href="#">Launch Stack</a>
		<a href="#">Single EC2 Instance web server with Amazon RDS database instance</a>	<a href="#">Launch Stack</a>
	A simple Ruby on Rails "Hello World" application.	<a href="#">Highly Available Web Server with Multi-AZ Amazon RDS Instance</a>	<a href="#">Launch Stack</a>
		<a href="#">Single EC2 Instance with local MySQL database</a>	<a href="#">Launch Stack</a>
		<a href="#">Single EC2 Instance web server with Amazon RDS database instance</a>	<a href="#">Launch Stack</a>
		<a href="#">Highly Available Web Server with Multi-AZ Amazon RDS Instance</a>	<a href="#">Launch Stack</a>

In the first row of the table, the "Template URL" column for the "Single EC2 Instance with local MySQL database" template is circled in red.

Screenshot of the AWS CloudFormation Stacks page showing a single stack named "stack-2058619" with a status of "CREATE\_COMPLETE". The "Create stack" button is highlighted with a red box.

Stacks (1)

Stack name	Status	Created time	Description
stack-2058619	CREATE_COMPLETE	2026-02-09 13:43:59 UTC+0100	-

Filter status: Active

Stack actions: Delete, Update stack, Stack actions, Create stack

Create stack: With new resources (standard) (highlighted), With existing resources (import resources)

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3. Open the CloudFormation page and click on Create stack with new resources

4. Select Upload a template file
5. Choose your template file
6. Click on Next

Screenshot of the AWS CloudFormation 'Create stack' wizard - Step 1: Prerequisites - Prepare template.

The page shows the 'Create stack' wizard with Step 1 selected. It provides options to choose an existing template from S3 or Infrastructure Composer, or to sync from Git. The 'Upload a template file' option is highlighted with a red circle and labeled '4'. Below it, a file input field is also highlighted with a red circle and labeled '5'. The 'Next' button at the bottom right is highlighted with a yellow circle and labeled '6'.

**Create stack**

**Prerequisites - Prepare template**

You can also create a template by scanning your existing resources in the [IaC generator](#).

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Choose an existing template  
Upload or choose an existing template.

Build from Infrastructure Composer  
Create a template using a visual builder.

**Specify template**

This GitHub repository [contains sample CloudFormation templates that can help you get started on new infrastructure projects. Learn more](#)

Template source

Selecting a template generates an Amazon S3 URL where it will be stored. A template is a JSON or YAML file that describes your stack's resources and properties.

Amazon S3 URL  
Provide an Amazon S3 URL to your template.

Upload a template file  
Upload your template directly to the console.

Sync from Git  
Sync a template from your Git repository.

**Upload a template file**

5

ec2wordpress.json

JSON or YAML formatted file

S3 URL: <https://s3.us-east-1.amazonaws.com/cf-templates-1aqijtza2s-us-east-1/2026-02-09T181634.938Z/ec2wordpress.json>

[View in Infrastructure Composer](#)

[Cancel](#) [Next](#) 6

Provide the following:

7. Stack name
8. Database name
9. Database password
10. Database root password
11. Database user
12. Instance type
13. Select KeyName
14. Click on next

The screenshot shows the 'Specify stack details' step of the AWS CloudFormation 'Create stack' wizard. The stack name is set to 'wordpress-stack-demo'. Parameters include DBName (MyDatabase), DBPassword (redacted), DBRootPassword (redacted), DBUser (redacted), and InstanceType (t1.micro). The KeyName field is empty. The SSHLocation field contains '0.0.0.0/0'. The 'Next Step' button is highlighted with a red circle.

Provide a stack name  
Stack name: wordpress-stack-demo

Parameters

DBName: MyDatabase

DBPassword: \*\*\*\*\*

DBRootPassword: \*\*\*\*\*

DBUser: \*\*\*\*\*

InstanceType: t1.micro

KeyName: Select AWS::EC2::KeyPair::KeyName

SSHLocation: 0.0.0.0/0

Cancel Previous Next 14

14. Select the Preserve successfully provisioned resources

15. Click next

**Stack failure options**

**Behavior on provisioning failure**

Specify the roll back behavior for a stack failure. [Learn more](#)

Roll back all stack resources  
Rolls back the stack to the last known stable state.

Preserve successfully provisioned resources  
Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

**Delete newly created resources during a rollback**

Specify whether resources that were created during a failed operation should be deleted regardless of their deletion policy. [Learn more](#)

Use deletion policy  
Retains or deletes created resources according to their attached deletion policy.

Delete all newly created resources  
Deletes created resources during a rollback regardless of their attached deletion policy.

**Additional settings**

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

▶ **Stack policy - optional**  
Defines the resources that you want to protect from unintentional updates during a stack update.

▶ **Rollback configuration - optional**  
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back.

▶ **Notification options - optional**  
Specify a new or existing Amazon Simple Notification Service topic where notifications about stack events are sent.

▶ **Stack creation options - optional**  
Specify the timeout and termination protection options for stack creation.

14

15

[Cancel](#) [Previous](#) **Next**

aws Search [Alt+S] Ask Amazon Q United States (N. Virginia) SimpleText-Account-704 (1111-2222-3333) odi\_user\_2058619

**CloudFormation > Stacks > Create stack**

**Step 1: Specify template**

**Prerequisite - Prepare template**  
Template URL: https://s3.us-east-1.amazonaws.com/cf-templates-1lqjijcra2s-us-east-1/2026-02-09T182527.8892twl-ec2wordpress.json  
Template is ready

**Review and create**

**Step 2: Specify stack details**

**Provide a stack name**  
Stack name: wordpress-stack-demo

**Parameters (7)**  
Search parameters  
Key Value  
DBName MyDatabase  
DBPassword \*\*\*\*\*  
DBRootPassword \*\*\*\*\*  
DBUser \*\*\*\*\*

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**CloudFormation > Stacks > wordpress-stack-demo**

**Stacks (2)**  
Filter status: Active  
View nested

Stack	Created	Status
wordpress-stack-demo	2026-02-09 19:45:59 UTC+0100	<span>CREATE_COMPLETE</span>
stack-2058619	2026-02-09 13:43:59 UTC+0100	<span>CREATE_COMPLETE</span>

**Events (9)**  
Search events

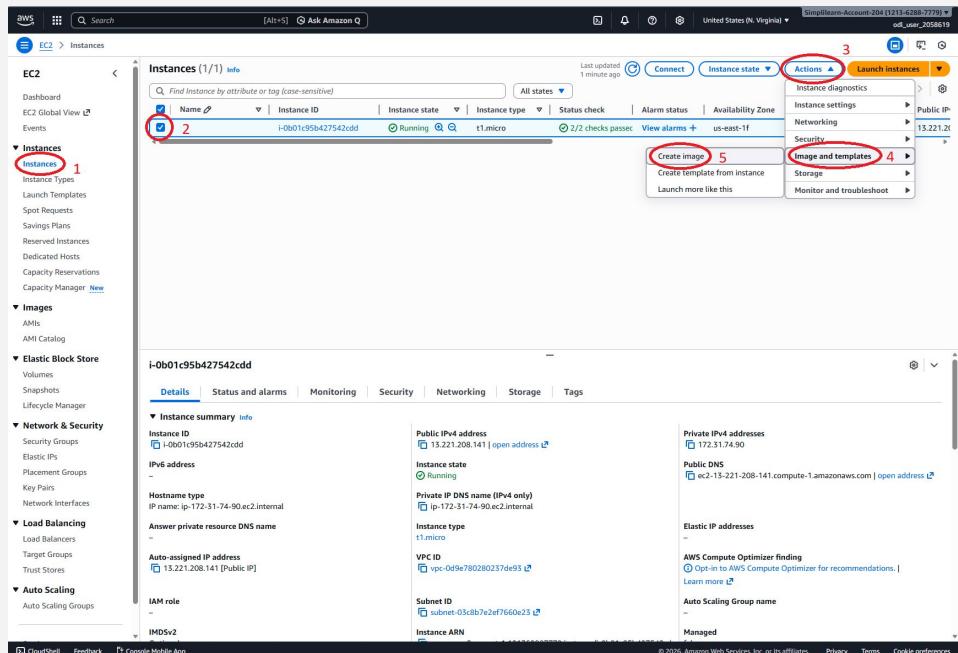
Operation ID	Timestamp	Logical ID	Status	Detailed status	Status reason
72773c-45e7-499-8646-7b915ac4d5fe	2026-02-09 19:50:09 UTC+0100	wordpress-stack-demo	<span>CREATE_COMPLETE</span>	-	-
72773c-45e7-499-8646-7b915ac4d5fe	2026-02-09 19:50:07 UTC+0100	WebServerInstance	<span>CREATE_COMPLETE</span>	-	-

16. Review all the stack configuration details and click Submit

After a few seconds, the stack will be created.

# Create an AMI of the WordPress instance

1. Open the EC2 Instances page
2. Select the instance
3. Click on Actions
4. Select Image and templates
5. Click on Create image



6. Type image name

7. Click on Create Image

The screenshot shows the 'Create image' page in the AWS EC2 console. The 'Image details' section is active, displaying the following fields:

- Instance ID:** i-0b01c95b427542cdd
- Image name:** wordpress-demo-ami (highlighted with a red circle and labeled '6')
- Image description - optional:** (empty)
- Reboot instance:** (checkbox checked)

The 'Instance volumes' section shows one volume configuration:

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/xv...	Create new snapshot from ...	8	EBS General Purpose SSD - ...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/>

The 'Tags - optional' section contains two options:

- Tag image and snapshots together (highlighted with a red circle and labeled '7')
- Tag image and snapshots separately

The 'Create image' button at the bottom right is highlighted with a red circle and labeled '7'.

The screenshot shows the AWS EC2 Instances page. A message at the top states: "Currently creating AMI ami-01c2f6021e4149c2b from instance i-0b01c95b427542dd. Check that the AMI status is Available' before deleting the instance or carrying out other actions related to this AMI." Below this, there is a table for "Instances (1/1) Info". The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. One row is shown: "i-0b01c95b427542dd" (Running, t1.micro, 2/2 checks passed, View alarms, us-east-1f, ec2-13-221-208-141.co..., 13.221.20...). The "Actions" dropdown menu is open.

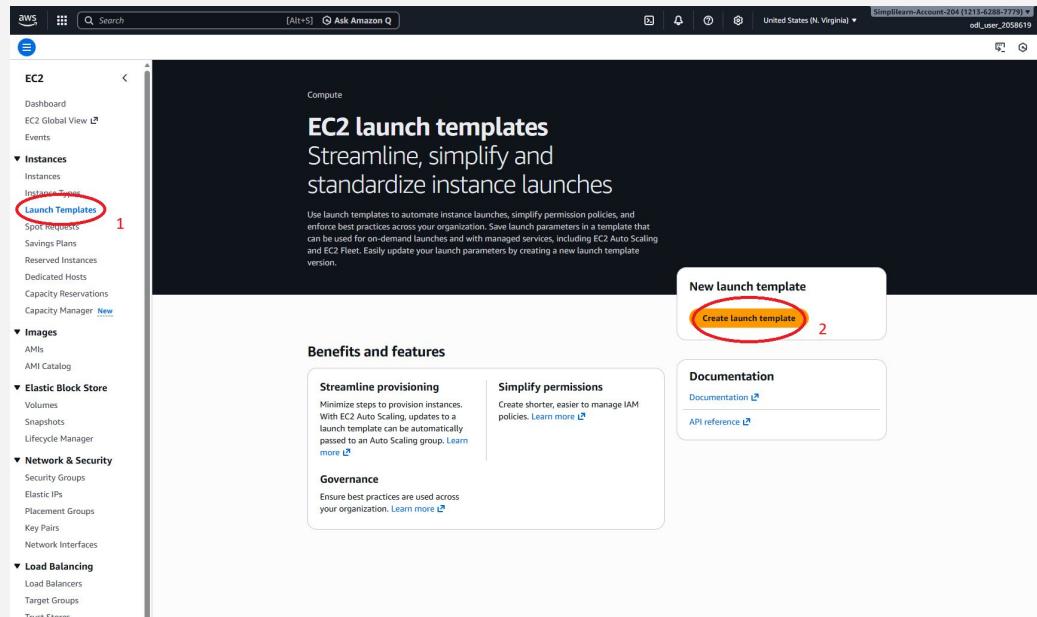
By following steps the AMI will be successfully created

The screenshot shows the AWS EC2 AMIs page. The table header is "Amazon Machine Images (AMIs) (1) Info". The columns are: Name, AMI name, Source, Owner, Visibility, and Status. One row is listed: "wordpress-demo-ami" (ami-01c2f6021e4149c2b, 121362887779/wordpress-demo-ami, 121362887779, Private, Available). The "Actions" dropdown menu is open.

# Configure Auto Scaling to launch a new WordPress instance

1. Select Launch Templates

2. Click on Create Launch template button



3. Name the Launch template

4. Provide a description

5. Select My AMIs

6. Select the previously created AMI

The screenshot shows the 'Create launch template' wizard in the AWS Management Console. The first two steps (Name and Description) are completed. In step 3, the 'Launch template name - required' field contains 'MyWordpressTemp' (circled in red as step 3). In step 4, the 'Template version description' field contains 'My wordpress template' (circled in red as step 4). Step 5 shows the 'Application and OS Images (Amazon Machine Image)' section where the 'My AMIs' tab is selected (circled in red as step 5). Step 6 shows the list of AMIs with 'wordpress-demo-ami' selected (circled in red as step 6). The right side of the screen displays the summary and configuration details for the launch template.

Launch template name and description

Launch template name - required  
MyWordpressTemp **3**  
More info  
Max 128 chars. No spaces or special characters like '!', "!", @!.

Template version description

Template version description  
My wordpress template **4**  
255 chars

Auto Scaling guidance [Info](#)  
Select this if you intend to use this template with EC2 Auto Scaling  
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags  
► Source template

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images **5**

Recents **5** My AMIs **5** Quick Start

Don't include in launch template  Owned by me **6**  Shared with me

Browse more AMIs **6**  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)  
wordpress-demo-ami  
ami-0c920c5ee61a7d001  
2026-02-11T11:56:53.000Z Virtualization: hvm ENA enabled: true Root device type: ebs **6**

Description

Architecture x86\_64 AMI ID ami-0c920c5ee61a7d001

▼ Summary

Software Image (AMI)  
wordpress-demo-ami  
ami-0c920c5ee61a7d001

Virtual server type (instance type)

Firewall (security group)

Storage (volumes)  
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage for t2.micro where t2.micro isn't available when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance. Charges may apply depending on your account's free tier status.

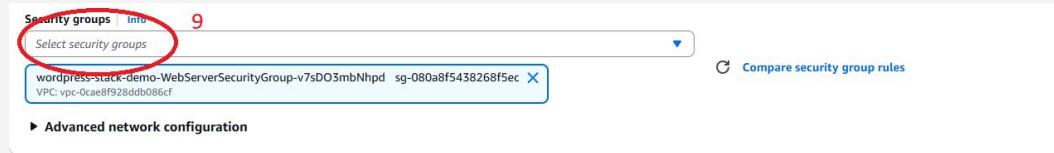
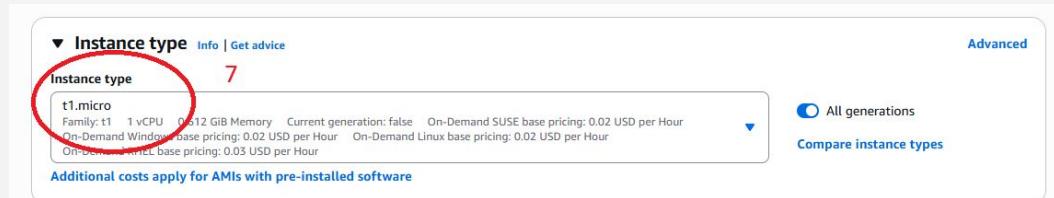
[Cancel](#) [Create launch template](#)

7. Select Instance type

8. Select Key pair name

9. The security group needs to be the same as the first instance security group

10. Click on Create launch template



## 10. Select Auto Scaling Groups

The screenshot shows the AWS EC2 Auto Scaling Groups page. On the left, there's a sidebar with various navigation links. Under the 'Auto Scaling' section, the 'Auto Scaling Groups' link is circled in red and has the number '10' next to it. The main content area displays the 'Amazon EC2 Auto Scaling' landing page. At the top right of this page, there's a 'Create Auto Scaling group' button, which is also circled in red and has the number '11' next to it. The landing page text says: 'Amazon EC2 Auto Scaling helps maintain the availability of your applications'. It also includes a brief description of what Auto Scaling groups are and how they work.

## 11. Click on Create Auto Scaling group button

12. Name the Auto Scaling group

13. Select Launch template

14. Click on Next button

Simplilearn-Account-098 (5327-5275-9110) United States (N. Virginia) od\_User\_2065272

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1: Choose launch template or configuration  
Step 2: Choose instance launch options  
Step 3: optional  
Step 4: optional  
Step 5: optional  
Step 6: optional  
Step 7: Review

**Name** **12**  
Auto Scaling group name  
Enter a name to identify the group.  
**AutoScalingWordpressDemo**  
Must be unique to your account in the current Region and no more than 255 characters.

**Launch template** **Info** **Switch to launch configuration**  
Launch template  
Select a launch template, and contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.  
**MyWordpressTemp**  
Create a launch template **13**  
Version  
Default (1) **13** **Cancel** **Next**

Description  
Autoscaling demo

Launch template  
NewWordpressTemp **14**  
lt-046c41a4219816a59

AMI ID  
ami-0c1fe732b5494dc14

Key pair name  
MyDemoKey

Security groups  
-

Instance type  
t1.micro

Request Spot Instances  
No

Additional details

Storage (volumes)  
-

Date created  
Wed Feb 11 2026 13:18:22 GMT+0100 (Central European Standard Time)

Select Availability Zones and click on Next button

**Network Info**

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

**VPC**

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0c2a74bc545fb3b28  
172.31.0.0/16 Default

Create a VPC

**Availability Zones and subnets**

Define which Availability zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

use1-az1 (us-east-1c) | subnet-0a6c6051576d0e681 X  
172.31.0.0/20 Default

use1-az2 (us-east-1d) | subnet-01adf0571c288c5ce X  
172.31.80.0/20 Default

use1-az3 (us-east-1e) | subnet-04fcf353af118208e X  
172.31.48.0/20 Default

use1-az4 (us-east-1a) | subnet-03901e2a2b4218969 X  
172.31.16.0/20 Default

use1-az5 (us-east-1f) | subnet-00b014264077a13a2 X  
172.31.64.0/20 Default

use1-az6 (us-east-1b) | subnet-00bc0f3b1bf131d7 X  
172.31.32.0/20 Default

Create a subnet

**Availability Zone distribution - new**

Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

**Balanced best effort**  
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability zone.

**Balanced only**  
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Your requested instance type (t1.micro) is not available in 1 Availability Zone. You may need to change the instance type or choose other Availability Zones for better resiliency. [Learn more](#)

[Cancel](#) [Skip to review](#) [Previous](#) **Next**

Click on Attach to a new load balancer and create a target group

Click Next and skip the other steps. In step 7, click the Create Auto Scaling Group button.

Integrate with other services

Step 4 - optional  
Configure group size and scaling  
Step 5 - optional  
Add notifications  
Step 6 - optional  
Add tags  
Step 7  
Review

Select Load balancing options

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to a new load balancer

Load balancer type  
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the Load Balancing console.

Application Load Balancer  
HTTP, HTTPS

Network Load Balancer  
TCP, UDP, TLS

Load balancer name  
Name cannot be changed after the load balancer is created.  
AutoScalingWordpressDemo-1

Load balancer scheme  
Scheme cannot be changed after the load balancer is created.

Internal

Internet-facing

Network mapping  
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC  
vpc-0c2a74bc545fb5b28 [?]

Availability Zones and subnets  
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

use1-az1 (us-east-1c)  
Select a subnet  
subnet-0a6c6051576d0e681

use1-az2 (us-east-1d)  
Select a subnet  
subnet-01adff0571c288c5ce

use1-az3 (us-east-1f)  
Select a subnet  
subnet-009014264077a15a2

use1-az4 (us-east-1a)  
Select a subnet  
subnet-03901e2a2b4218969

use1-az5 (us-east-1b)  
Select a subnet  
subnet-008cf3b1fb131d7

use1-az6 (us-east-1e)  
Select a subnet  
subnet-04fcf553af118208e

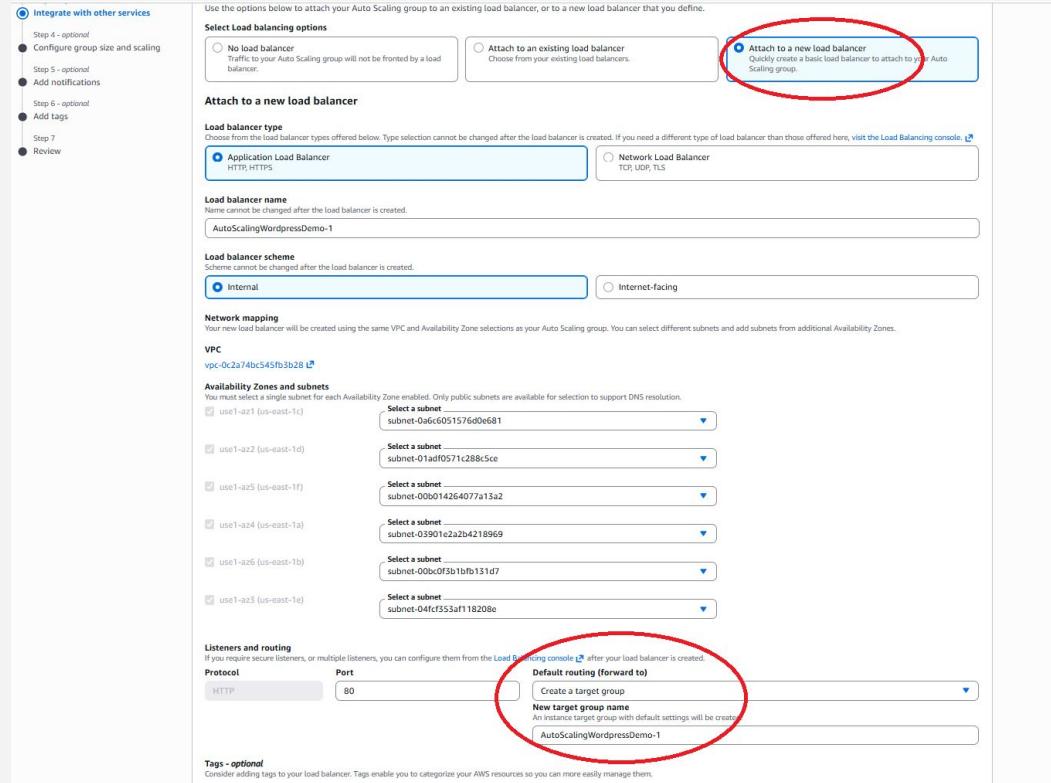
Listeners and routing  
If you require secure listeners, or multiple listeners, you can configure them from the Load Balancing console [?] after your load balancer is created.

Protocol  
HTTP

Port  
80

Default routing (forward to)  
 Create a target group  
 New target group name  
An instance target group with default settings will be created  
AutoScalingWordpressDemo-1

Tags - optional  
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.



# Configure the new WordPress instance to shut down automatically

1. Select the Auto Scaling Group
2. Select Automatic scaling
3. Click on Create scheduled action

The screenshot shows three stacked screenshots of the AWS Management Console for Auto Scaling.

- Screenshot 1 (Top): Auto Scaling groups**
  - The left sidebar shows the EC2 navigation path: EC2 > Auto Scaling groups.
  - The main area displays "Auto Scaling groups (1/1) Info".
  - A single group is listed: "AutoScalingWordpressDemo" (MyWordpressTemp | Version Default).
  - A red circle labeled "1" highlights the checkbox next to the group name.
- Screenshot 2 (Middle): Auto Scaling group details**
  - The left sidebar shows the EC2 navigation path: EC2 > Auto Scaling groups > AutoScalingWordpressDemo.
  - The main area shows the "Auto Scaling group: AutoScalingWordpressDemo" page.
  - The "Automatic scaling" tab is selected, highlighted by a red circle labeled "2".
  - A callout box explains scaling policies: "Scaling policies resize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 instances require more time to initialize."
- Screenshot 3 (Bottom): Scheduled actions**
  - The left sidebar shows the EC2 navigation path: EC2 > Auto Scaling > Auto Scaling Groups.
  - The main area displays "Scheduled actions (0) Info".
  - A red circle labeled "3" highlights the "Create scheduled action" button.

- Type the action name
- Type the desired instance capacity
- Choose Cron and the action time (in this case 09:00 Monday-Friday)
- Select the start date and time
- Create action

**Create scheduled action** X

**Name** 4  
start-wordpress-dev

① Provide at least one value for Desired, Min, or Max Capacity

**Desired capacity** 5  
1      Min: 1      Max: 1

**Recurrence**  
Schedule a specific date and time for the first scheduled action to run. It's interpreted in recurrence time zone: Etc/UTC  
Cron: 0 9 \* \* MON-FRI 5

**Time zone** 6  
Etc/UTC  
Current time in selected time zone is 2026-02-11/17:18 UTC

**Start date** 7  
2026/02/12 (calendar icon)  
Format: DD/MM/YY

**Start time** 7  
08:00 (clock icon)  
Format: hh:mm

**Set End Time**

[Learn more about scheduled scaling](#) ↗

8 Cancel Create

9. Type the action name

10. Type the desired instance capacity

11. Choose Cron and the action time (in this case  
17:00 Monday-Friday)

12. Select the start date and time

13. Create action

Create scheduled action

9

Name  
stop-wordpress-dev

10

Desired capacity      Min      Max  
0      0      0

Recurrence  
Schedule a specific date and time for the first scheduled action to run. Interpreted in recurrence time zone: Etc/UTC

Cron      0 17 \* \* MON-FRI

11

Time zone  
Etc/UTC

Current time in selected time zone is 2026-02-11/17:20 UTC

Start date      Start time  
2026/02/12      16:00

Format: YYYY/MM/DD      Format: hh:mm

12

Set End Time

Learn more about scheduled scaling ↗

13

Cancel      Create

Two scheduled actions are created: **start-wordpress-dev**, which starts the development instance, and **stop-wordpress-dev**, which shuts down the development instance.

The screenshot shows the AWS CloudWatch Metrics interface. On the left, a sidebar menu includes Network Interfaces, Load Balancing (Load Balancers, Target Groups, Trust Stores), Auto Scaling (Auto Scaling Groups), and Settings. The main area displays a table titled "Scheduled actions (2) Info". The table has columns for Name, Start time, End time, Recurrence, and Time zone. Two entries are listed:

Name	Start time	End time	Recurrence	Time zone
start-wordpress-dev	2026 February 12, 09:00:00 AM +...		0 9 * * MON-FRI	Etc/UTC
stop-wordpress-dev	2026 February 12, 05:00:00 PM +...		0 17 * * MON-FRI	Etc/UTC

At the top right of the table, there are buttons for "Actions" and "Create scheduled action". Navigation controls (back, forward, search) are also present at the top right.

# Output Screenshots

## Instances

Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Monitor...	Security group name
i-0113204912c0e9e40	Running	t1.micro	2/2 checks passed <a href="#">View alarms +</a>	us-east-1b	ec2-54-174-118-91.co...	54.174.118.91	-	-	-	disabled	wordpress-stack-demo...
i-0f786ec2f03795f1e	Running	t1.micro	2/2 checks passed <a href="#">View alarms +</a>	us-east-1f	ec2-44-192-69-206.co...	44.192.69.206	-	-	-	disabled	wordpress-stack-demo...

## Health checks

ID	Name	Details	Status in last 24 hours	Current s...	Alarm	State	Actions
<a href="#">087f7f6e-067d-4687-b83d-9cd4101bbcca</a>	wordpress-dev	<a href="http://44.192.69.206:80/">http://44.192.69.206:80/</a>		Healthy	None, Create alarm	Enabled	
<a href="#">96ed5e70-26ca-41e7-a9ca-e1230b6d228e</a>	wordpress-prod	<a href="http://54.174.118.91:80/">http://54.174.118.91:80/</a>		Healthy	None, Create alarm	Enabled	