

Deploy WordPress Docker Container on ECS

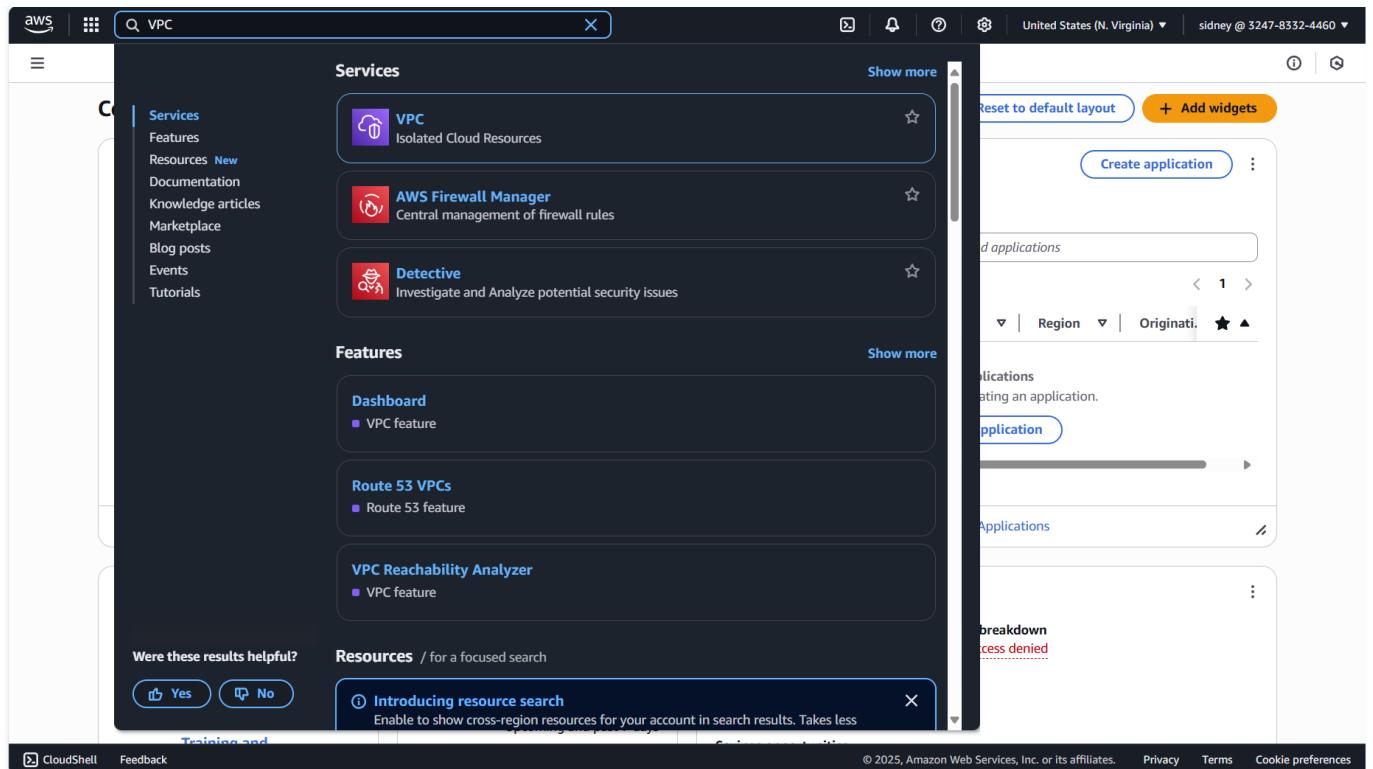
In this tutorial, we will walk through the process of hosting a WordPress site on Amazon ECS. While some people might choose to set up their WordPress site on Virtual machines, we will be taking a more advanced approach using Amazon ECS for a scalable, flexible and modern cloud environment.

By the end of this tutorial, you will understand how to deploy your WordPress site on AWS ECS with ease.

STEP 1: Create a VPC

Our first step is to create our Virtual Private Cloud which is our VPC, which will serve as the dedicated network for our WordPress deployment.

So, we have to login to AWS Management console and search for VPC



Click on “VPC”

Screenshot of the AWS VPC Dashboard. The dashboard shows various resources by region, including VPCs, Subnets, Route Tables, Internet Gateways, NAT Gateways, VPC Peering Connections, Network ACLs, Security Groups, Customer Gateways, and Virtual Private Gateways. A sidebar on the left lists options like Virtual private cloud, Security, and PrivateLink and Lattice. A top bar includes a search bar, navigation icons, and user information.

Click on “Create VPC”

Screenshot of the Create VPC page. The left panel shows the VPC settings, including options for creating only a VPC or both a VPC and other networking resources. The right panel shows a preview of the VPC structure, including four subnets (us-east-1a, us-east-1b) and their associated route tables and IP ranges. An orange arrow points to the “VPC and more” button in the settings panel.

In “VPC Settings”, we will use “VPC and More”

AWS | Search [Alt+S] | United States (N. Virginia) | sidney @ 3247-8332-4460

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation Info
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.
 Auto-generate

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.
 65,536 IPs
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
 No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block

Tenancy Info
 Default

Number of Availability Zones (AZs) Info
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

Preview

VPC Show details Your AWS virtual network

project-vpc

Subnets (4)		Route tabl
us-east-1a		Route network !
A	project-subnet-public1-us-east-1a	project-rtb-p
B	project-subnet-private1-us-east-1a	project-rtb-p
us-east-1b		project-rtb-p
A	project-subnet-public2-us-east-1b	project-rtb-p
B	project-subnet-private2-us-east-1b	project-rtb-p

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

On “Name tag Auto-generation”, we will give a name. Let us call it “**demo-VPC**”

AWS | Search [Alt+S] | United States (N. Virginia) | sidney @ 3247-8332-4460

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation Info
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.
 Auto-generate

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.
 65,536 IPs
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
 No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block

Tenancy Info
 Default

Number of Availability Zones (AZs) Info
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

Preview

VPC Show details Your AWS virtual network

demo-VPC-vpc

Subnets (4)		Route tabl
us-east-1a		Route network !
A	demo-VPC-subnet-public1-us-east-	demo-VPC-rt
B	demo-VPC-subnet-private1-us-east-	demo-VPC-rt
us-east-1b		demo-VPC-rt
A	demo-VPC-subnet-public2-us-east-	demo-VPC-rt
B	demo-VPC-subnet-private2-us-east-	demo-VPC-rt

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then on “**IPv4 CIDR block**”, we will use the default IP Range, that is “**10.0.0.16**”

The screenshot shows the 'Create VPC' wizard in the AWS Management Console. The 'VPC settings' section includes fields for 'Name tag auto-generation', 'IPv4 CIDR block', 'IPv6 CIDR block', 'Tenancy', and 'Number of Availability Zones (AZs)'. The 'Preview' section shows a hierarchical tree of resources: a VPC named 'demo-VPC-vpc' containing four subnets ('us-east-1a' and 'us-east-1b') each with two subnets, and a route table with three routes.

Scroll down to “Availability Zones”

This screenshot shows the 'Customize AZs' section of the VPC creation wizard. It includes fields for 'Number of Availability Zones (AZs)', 'Number of public subnets', 'Number of private subnets', 'NAT gateways (\$)', and 'VPC endpoints'. An orange arrow points from the 'Number of Availability Zones (AZs)' field to the '2' option, indicating it is selected.

On “Number of Availability Zones”, choose “2”. This is already chosen by default.

Number of Availability Zones (AZs) [Info](#)

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 | 2 | 3

► Customize AZs

Number of public subnets [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | 2

Number of private subnets [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 | 2 | 4

► Customize subnets CIDR blocks

NAT gateways (\$) [Info](#)

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None | In 1 AZ | 1 per AZ

VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None | S3 Gateway

For number of Public and Private subnets, we will choose two each.

Number of public subnets [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | 2

Number of private subnets [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 | 2 | 4

► Customize subnets CIDR blocks

NAT gateways (\$) [Info](#)

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None | In 1 AZ | 1 per AZ

VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None | S3 Gateway

DNS options [Info](#)

- Enable DNS hostnames
- Enable DNS resolution

► Additional tags

[Cancel](#)

 [Preview code](#)

[Create VPC](#)

In this tutorial, we are not going to enable “**NAT Gateway**” and the “**S3 Gateway**”. So, both have to be “**None**”

Number of public subnets [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | 2

Number of private subnets [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 | 2 | 4

► Customize subnets CIDR blocks

NAT gateways (\$) [Info](#)

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None | In 1 AZ | 1 per AZ

VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None | S3 Gateway

DNS options [Info](#)

- Enable DNS hostnames
- Enable DNS resolution

► Additional tags

[Cancel](#)

 [Preview code](#)

[Create VPC](#)

Click on “Create VPC”

aws | Search [Alt+S] | United States (N. Virginia) | sidney @ 3247-8332-4460

VPC > Your VPCs > Create VPC > Create VPC resources

Create VPC workflow

>Create internet gateway

42%

▼ Details

- ✓ Create VPC: vpc-0e734dd4ca9c669a3 [?] (Success)
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: vpc-0e734dd4ca9c669a3 [?] (Success)
- ✓ Create subnet: subnet-0ff9c3e4b5b11293f [?] (Success)
- ✓ Create subnet: subnet-08c4b8ca5a6cee0a2 [?] (Success)
- ✓ Create subnet: subnet-00441c0e0bbfe1d73 [?] (Success)
- ✓ Create subnet: subnet-0fd9abe69173e3f9d [?] (Success)
- ✓ Create internet gateway
- ✓ Attach internet gateway to the VPC
- ✓ Create route table
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Create route table
- ✓ Associate route table
- ✓ Create route table
- ✓ Associate route table
- ✓ Create route table
- ✓ Associate route table
- ✓ Verifying route table creation

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The VPC is being created

aws | Search [Alt+S] | United States (N. Virginia) | sidney @ 3247-8332-4460

VPC > Your VPCs > Create VPC > Create VPC resources

Create VPC workflow

Success

▼ Details

- ✓ Create VPC: vpc-0e734dd4ca9c669a3 [?] (Success)
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: vpc-0e734dd4ca9c669a3 [?] (Success)
- ✓ Create subnet: subnet-0ff9c3e4b5b11293f [?] (Success)
- ✓ Create subnet: subnet-08c4b8ca5a6cee0a2 [?] (Success)
- ✓ Create subnet: subnet-00441c0e0bbfe1d73 [?] (Success)
- ✓ Create subnet: subnet-0fd9abe69173e3f9d [?] (Success)
- ✓ Create internet gateway: igw-0a4081a6000ffdaa4 [?] (Success)
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: rtb-0278b8aabb83afeb5 [?] (Success)
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Create route table: rtb-0bb0cc2f33b52b79f [?] (Success)
- ✓ Associate route table
- ✓ Create route table: rtb-0d3d48c44e0bc5a7b [?] (Success)
- ✓ Associate route table
- ✓ Verifying route table creation

View VPC

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The VPC has been created. Click on “View VPC”

The screenshot shows the AWS VPC Details page for a VPC named "demo-VPC-vpc". The VPC ID is "vpc-0e734dd4ca9c669a3". The "Details" tab is selected, showing the following configuration:

VPC ID	State	Block Public Access	DNS hostnames
vpc-0e734dd4ca9c669a3	Available	Off	Enabled
DNS resolution	Tenancy	DHCP option set	Main route table
Enabled	default	dopt-067ba7e48365eae16	rtb-0dbb64bf324b0f2af
Main network ACL	Default VPC	IPv4 CIDR	IPv6 pool
acl-062eb2bb40bdb4ac6	No	10.0.0.0/16	-
IPv6 CIDR (Network border group)	Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID
-	Disabled	-	324783324460

The "Resource map" tab is also visible, showing the VPC's subnets, route tables, and network connections.

Click on “Your VPCs”

The screenshot shows the AWS VPC dashboard. In the left sidebar, under "Virtual private cloud", the "Your VPCs" link is highlighted with an orange arrow. The main pane displays a table titled "Your VPCs (1/2) Info" with one item listed:

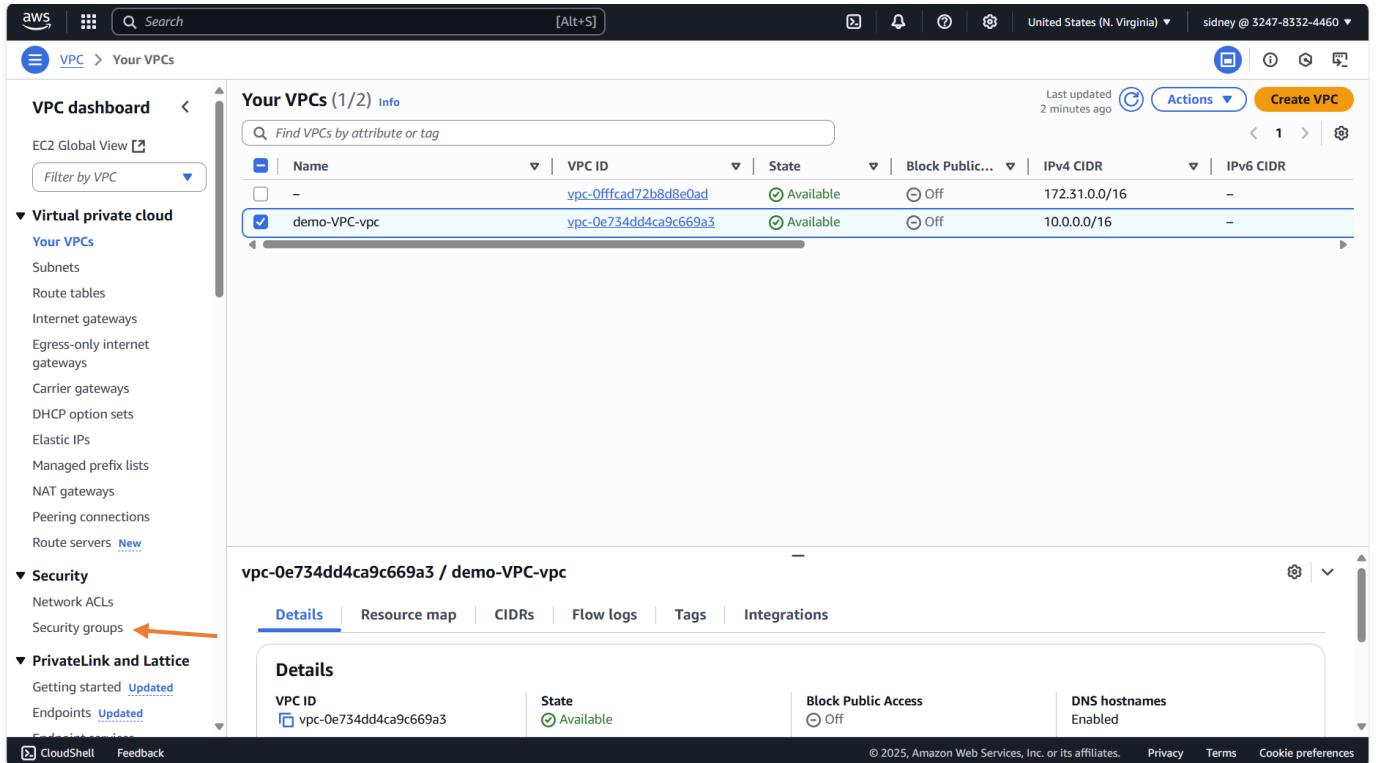
Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
demo-VPC-vpc	vpc-0e734dd4ca9c669a3	Available	Off	10.0.0.0/16	-

The "Details" tab is selected in the bottom navigation bar, showing the same configuration details as the previous screenshot.

You can see the VPC we just created.

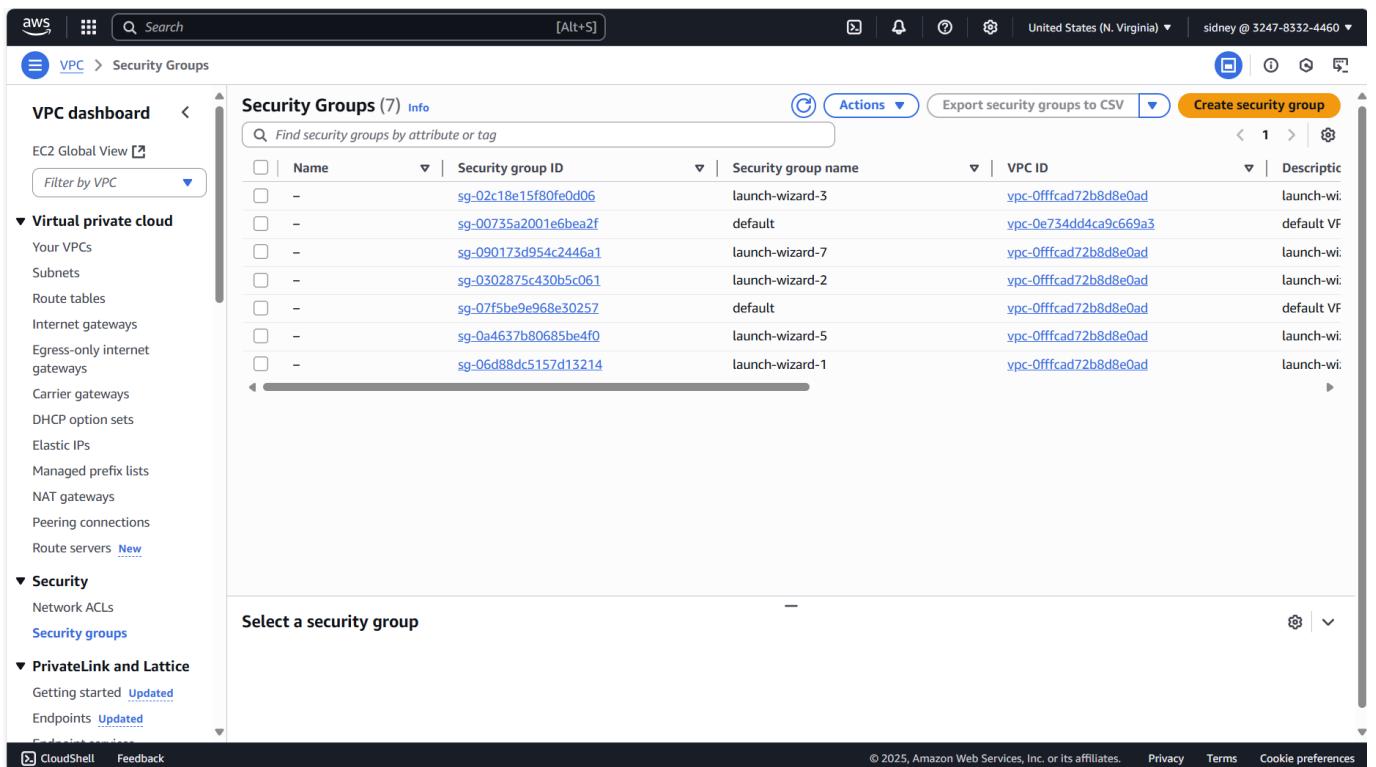
STEP 2: Create Security Group

The next thing to do is to create our security group. Let us navigate to security group



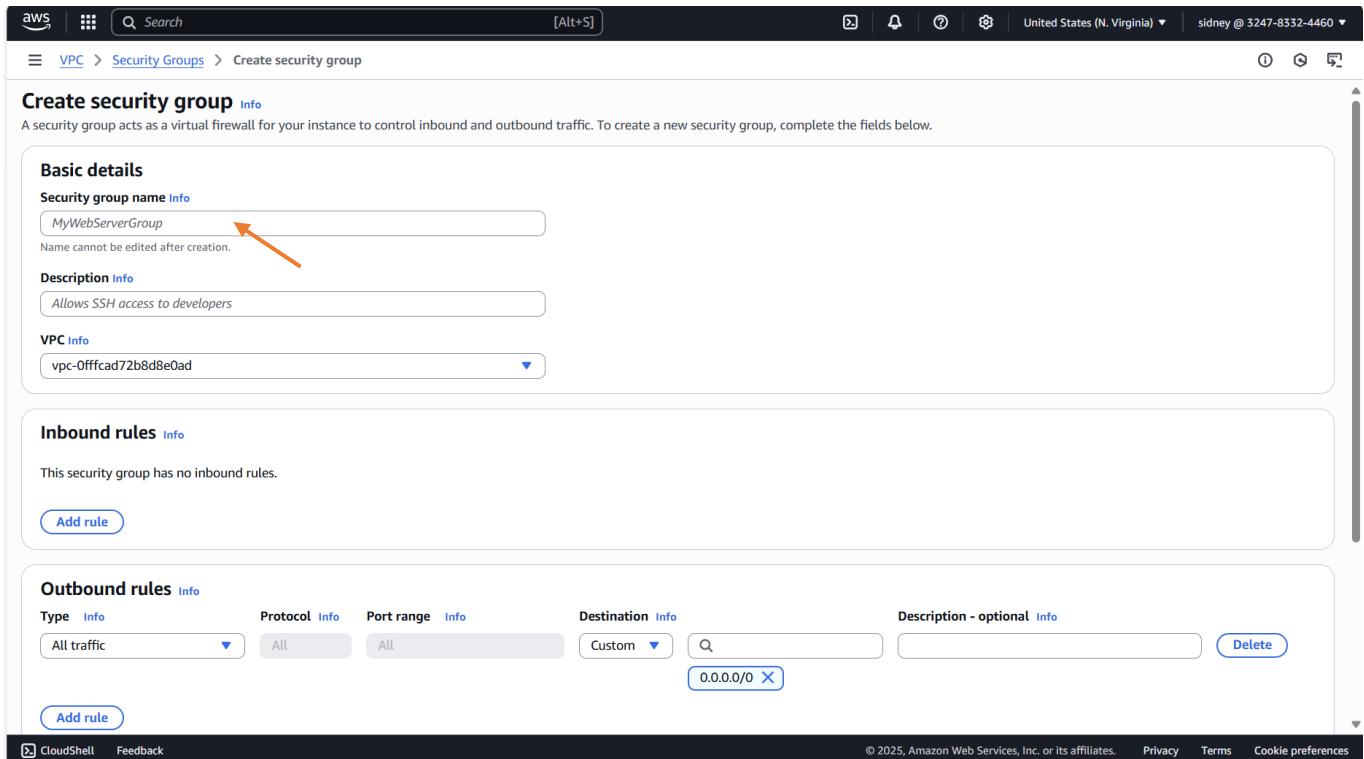
The screenshot shows the AWS VPC dashboard. On the left sidebar, under the 'Virtual private cloud' section, the 'Security groups' link is highlighted with a red arrow. The main content area displays 'Your VPCs (1/2) Info' with two entries: one for a default VPC and one for 'demo-VPC-vpc'. The 'demo-VPC-vpc' entry has its details expanded, showing fields like VPC ID, State, Block Public Access, and DNS hostnames.

Click on “Security Groups”



The screenshot shows the AWS Security Groups page. The left sidebar shows the 'Virtual private cloud' section with the 'Security groups' link selected. The main content area displays 'Security Groups (7) Info' with a list of seven security groups, each with its name, security group ID, and associated VPC ID. At the top right of the table, there is a 'Create security group' button. Below the table, a section titled 'Select a security group' is visible.

Click on “Create Security Group” to create a security group that will be dedicated to our WordPress environment. We are going to set this up to allowing HTTP traffic by opening Port 80 for inbound access to enable user to reach our WordPress site over the web.



The screenshot shows the AWS VPC "Create security group" interface. In the "Basic details" section, the "Security group name" field contains "MyWebServerGroup". A red arrow points to this field. Below it, the "Description" field contains "Allows SSH access to developers". The "VPC" dropdown is set to "vpc-0fffcad72b8d8e0ad". The "Inbound rules" section indicates "This security group has no inbound rules." The "Outbound rules" section shows a single rule for "All traffic" with a destination of "0.0.0.0/0".

Let's call the security group “wp-sg” and put same for description. On “VPC”, click on the drop down and select the VPC we created.

The screenshot shows the 'Create security group' page in the AWS VPC console. In the 'Basic details' section, the security group name is 'wp-sg' and the VPC is 'vpc-0e734dd4ca9c669a3 (demo-VPC-vpc)'. The 'Inbound rules' section is empty, with a blue 'Add rule' button highlighted by a red arrow. The 'Outbound rules' section also has an 'Add rule' button.

On the “Inbound Rules”, we are going to add “HTTP”. Click on “Add Rule”

The screenshot shows the 'Create security group' page again. The 'Type' dropdown in the 'Inbound rules' section is set to 'Custom TCP', which is highlighted by a red arrow. The other fields in the 'Inbound rules' section are 'Protocol' (TCP), 'Port range' (0), and 'Source' (Custom). The 'Outbound rules' section remains empty.

Select “HTTP” and make it accessible from anywhere

The screenshot shows the AWS Security Groups creation wizard. It has three main sections:

- Inbound rules**: Type: HTTP, Protocol: TCP, Port range: 80, Source: Anywhere, Description: optional.
- Outbound rules**: Type: All traffic, Protocol: All, Port range: All, Destination: Custom, Description: optional.
- Tags - optional**: A note explaining tags, followed by an "Add new tag" button and a note about adding up to 50 more tags.

At the bottom right, there are "Cancel" and "Create security group" buttons. An orange arrow points from the "Create security group" button towards the EC2 Security Groups list in the next screenshot.

Click on “Create Security Group”

The screenshot shows the AWS EC2 Security Groups list. On the left is a navigation sidebar with options like EC2, Instances, Images, Elastic Block Store, Network & Security, etc. The main area shows a success message: "Security group (sg-0ca6fe563eb50634b | wp-sg) was created successfully". Below it is the "Details" section for the security group "sg-0ca6fe563eb50634b - wp-sg".

Security group name	Security group ID	Description	VPC ID
wp-sg	sg-0ca6fe563eb50634b	wp-sg	vpc-0e734dd4ca9c669a3

The "Inbound rules" tab is selected, showing one rule:

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-040d2d11a46639ba1	IPv4	HTTP	TCP	80

We have created our own security group.

STEP 3: Create Application Load Balancer

The next thing is to create a Load Balancer and a Target group to efficiently manage the incoming traffic.

So, go to the EC2 dash board

The screenshot shows the AWS EC2 Instances dashboard. On the left, there is a navigation sidebar with the following categories:

- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations
- Images** (selected)
- AMIs
- AMI Catalog
- Elastic Block Store**
- Volumes
- Snapshots
- Lifecycle Manager
- Network & Security**
- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces
- Load Balancing** (selected)
- Load Balancers
- Target Groups
- Trust Stores
- Auto Scaling**
- Auto Scaling Groups

An orange arrow points to the "Load Balancers" link under the "Load Balancing" category. The main content area is titled "Instances Info" and displays a message: "No instances. You do not have any instances in this region". It includes a "Launch instances" button and a search bar with placeholder text "Find Instance by attribute or tag (case-sensitive)".

Click on “Load Balancers”

Screenshot of the AWS EC2 Load Balancers page. The left sidebar shows navigation options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled "Load balancers" and displays a message: "Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic." It includes a search bar, filters for Name, DNS name, State, VPC ID, Availability Zones, Type, and Date, and a "Create load balancer" button. A red arrow points to the "Create load balancer" button.

Click on “Create Load Balancer”

Screenshot of the "Compare and select load balancer type" page. The top navigation bar shows "EC2 > Load balancers > Compare and select load balancer type". The main content is titled "Compare and select load balancer type" and includes a note: "A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)".

The page compares three types of load balancers:

- Application Load Balancer**: Handles HTTP and HTTPS traffic. It can target Lambda functions, API Gateways, and Application Load Balancers. A red arrow points to the "Create" button.
- Network Load Balancer**: Handles TCP, UDP, and TLS traffic. It supports VPCs and Network Load Balancers. A red arrow points to the "Create" button.
- Gateway Load Balancer**: Handles traffic from third-party virtual appliances supporting GENEVE. It includes icons for a lock, document, and fire.

At the bottom, there is a link to "Classic Load Balancer - previous generation" and a "Close" button.

We will use “Application Load Balancer”. Click on “Create”

The screenshot shows the AWS CloudFormation console with the URL [https://console.aws.amazon.com/cloudformation/home?#/stacks/create/edit](#). The page title is "Create Application Load Balancer". The "Basic configuration" section includes fields for "Load balancer name" (set to "wp-alb") and "Scheme" (set to "Internet-facing"). The "Load balancer IP address type" section shows "IPv4" selected. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and a footer with copyright information.

Let us call the load balancer “wp-alb” and for the “Scheme”, we will use “Internet-facing”

The screenshot shows the AWS CloudFormation console with the URL [https://console.aws.amazon.com/cloudformation/home?#/stacks/create/edit](#). The page title is "Create Application Load Balancer". The "Basic configuration" section includes fields for "Load balancer name" (set to "wp-alb") and "Scheme" (set to "Internet-facing"). The "Load balancer IP address type" section shows "IPv4" selected. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and a footer with copyright information.

Scroll down to “Network Mapping”

aws Search [Alt+S] United States (N. Virginia) sidney @ 3247-8332-4460

EC2 > Load balancers > Create Application Load Balancer

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info

The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups [\[?\]](#). For a new VPC, [create a VPC](#) [\[?\]](#).



vpc-0fffcad72b8d8e0ad
IPv4 VPC CIDR: 172.31.0.0/16

IP pools - new Info

You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view Pools in [Amazon VPC IP Address Manager console](#) [\[?\]](#).

Use IPAM pool for public IPv4 addresses

The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets Info

Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

- us-east-1a (use1-az6)
- us-east-1b (use1-az1)
- us-east-1c (use1-az2)
- us-east-1d (use1-az4)
- us-east-1e (use1-az3)
- us-east-1f (use1-az5)

Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#) [\[?\]](#).

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Select the VPC we created

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info

The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups [\[?\]](#). For a new VPC, [create a VPC](#) [\[?\]](#).



demo-VPC-vpc
vpc-0e734dd4ca9c69a3
IPv4 VPC CIDR: 10.0.0.0/16

IP pools - new Info

You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view Pools in [Amazon VPC IP Address Manager console](#) [\[?\]](#).

Use IPAM pool for public IPv4 addresses

The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets Info

Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

- us-east-1a (use1-az6)
- us-east-1b (use1-az1)

Select the two availability zones and ensure they are on public subnet

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#). For a new VPC, [create a VPC](#).

demo-VPC-vpc
vpc-0e734dd4ca9c669a3
IPv4 VPC CIDR: 10.0.0.0/16 (C)

IP pools - new [Info](#)
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view Pools in [Amazon VPC IP Address Manager console](#).

Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets [Info](#)
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

us-east-1a (use1-az6)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-Off9c3e4b5b11293f
IPv4 subnet CIDR: 10.0.0.0/20 demo-VPC-subnet-public1-us-east-1a (C)

us-east-1b (use1-az1)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-08c4b8ca5a6cee0a2
IPv4 subnet CIDR: 10.0.16.0/20 demo-VPC-subnet-public2-us-east-1b (C)

Scroll down to “security Group”

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups
 (C)

default
sg-00735a2001e6bea2f VPC: vpc-0e734dd4ca9c669a3

Select the security group we created, that is “wp-sg”

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups
 (C)

wp-sg
sg-0660e7fc567957e4a VPC: vpc-0e734dd4ca9c669a3

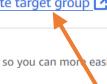
Scroll down to “Listeners and Routing” and create a target group

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 Remove

Protocol	Port	Default action
HTTP	80	Forward to <input type="button" value="Select a target group"/> (C)
1-65535		

Create target group 

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

Add listener tag
You can add up to 50 more tags.

Add listener

Click on “Create target group” and a new window will open

The screenshot shows the AWS Lambda "Create target group" wizard. On the left, a sidebar shows "Step 1: Specify group details" and "Step 2: Register targets". The main area is titled "Specify group details" with the sub-section "Basic configuration". It says "Your load balancer routes requests to the targets in a target group and performs health checks on the targets." Below this, it says "Settings in this section can't be changed after the target group is created." A section titled "Choose a target type" contains several options:

- Instances
 - Supports load balancing to instances within a specific VPC.
 - Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.
- IP addresses
 - Supports load balancing to VPC and on-premises resources.
 - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
 - Offers flexibility with microservice based architectures, simplifying inter-application communication.
 - Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- Lambda function
 - Facilitates routing to a single Lambda function.
 - Accessible to Application Load Balancers only.
- Application Load Balancer
 - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
 - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Select “IP Addresses” for “target type”

This screenshot shows the same "Create target group" wizard as the previous one, but with a different selection in the "Choose a target type" section. The "IP addresses" option is now selected, indicated by a blue border around its radio button. The other options—Instances, Lambda function, and Application Load Balancer—are still available but not selected.

Scroll down to “Target Group”

Target group name



A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol
Protocol for load balancer-to-target communication. Can't be modified after creation.

HTTP

Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.

1-65535

IP address type
Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

demo-VPC-vpc
vpc-0e734dd4ca9c669a3
IPv4 VPC CIDR: 10.0.0.0/16

Protocol version

HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

We will call the target group “**wp-TG**”

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol
Protocol for load balancer-to-target communication. Can't be modified after creation.

HTTP

Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.

1-65535

IP address type
Only targets with the indicated IP address type can be registered to this target group.

IPv4

IPv6

VPC
Select the VPC that hosts the load balancer. Only VPCs that support the IP address type selected above are available in this list. On the [Register targets](#) page, you can register IP addresses from this VPC, or from private IP addresses located outside of this load balancer's VPC (such as a peered VPC, EC2-Classic, or on-premises targets that are reachable over Direct Connect or VPN).

demo-VPC-vpc
vpc-0e734dd4ca9c669a3
IPv4 VPC CIDR: 10.0.0.0/16

Protocol version

HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Scroll down to the end

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

► Advanced health check settings

Attributes

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► Tags - optional

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

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

[CloudShell](#) [Feedback](#) © 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Click on “Next”

[aws](#) | [☰](#) [Search](#) [Alt+S] [✉](#) [🔔](#) [?](#) [⚙️](#) United States (N. Virginia) sidney @ 3247-8332-4460

☰ [EC2](#) > [Target groups](#) > Create target group

Step 1
Specify group details
Step 2
Register targets

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

IP addresses

Step 1: Choose a network
You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.

Network

demo-VPC-vpc
vpc-0e734dd4ca9c669a3
IPv4 VPC CIDR: 10.0.0.0/16

Step 2: Specify IPs and define ports
You can manually enter IP addresses from the selected network.

Enter an IPv4 address from a VPC subnet.
10.0.0. [Remove](#)

[Add IPv4 address](#)

You can add up to 4 more IP addresses.

Ports
Ports for routing to this target.
80
1-65535 (separate multiple ports with commas)

[CloudShell](#) [Feedback](#) © 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Complete our IPv4 address, that is “10.0.0.0/16”

aws | Search [Alt+S] | United States (N. Virginia) | sidney @ 3247-8332-4460

EC2 > Target groups > Create target group

Step 1
Specify group details
Step 2
[Register targets](#)

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

IP addresses

Step 1: Choose a network
You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.

Network
demo-VPC-vpc
vpc-0e734dd4ca9c669a3
IPv4 VPC CIDR: 10.0.0.0/16

Step 2: Specify IPs and define ports
You can manually enter IP addresses from the selected network.

Enter an IPv4 address from a VPC subnet.
10.0.0.0/16 [Remove](#)

[Add IPv4 address](#)
You can add up to 4 more IP addresses.

Ports
Ports for routing to this target.
80
1-65535 (separate multiple ports with commas)
[Include as pending below](#)

Review targets

Step 3: Review IP targets to include in your group
Confirm the IP targets to include in your target group. Add more IP targets by repeating steps 1 and 2 on this page. You can also register additional targets after your target group is created.

Targets (0)
Filter targets [Remove all pending](#)
Show only pending
Remove IPv4 address | Health status | IP address | Port | Zone
No IP addresses included yet
Specify IP addresses above and add to list.

0 pending [Cancel](#) [Previous](#) [Create target group](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Create target group”

Successfully created the target group: wp-TG. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.

wp-TG

Details

Target type IP	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0e734dd4ca9c669a3
IP address type IPv4	Load balancer <small>(None associated)</small>		
0 Total targets	0 Healthy	0 Unhealthy	0 Unused
0 Anomalous		0 Initial	0 Draining

Targets **Monitoring** **Health checks** **Attributes** **Tags**

Registered targets (0) Info Anomaly mitigation: Not applicable Deregister Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Targets Filter targets 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Head back to the application Load balancer tab

Select up to 5 security groups

Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener HTTP:80

Protocol HTTP	Port 80	Default action <small>Info</small> Forward to Select a target group C
------------------	------------	---

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Add listener

Load balancer tags - optional

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Refresh and select the target group we just created

The screenshot shows the AWS CloudShell interface with the following details:

- AWS Logo**
- Search Bar**: Search
- Region**: United States (N. Virginia)
- User**: sidney @ 3247-8332-4460
- Breadcrumbs**: EC2 > Load balancers > Create Application Load Balancer
- Header Actions**: [Alt+S], Refresh, Help, Settings
- Section: Select up to 5 security groups**
 - wp-SG sg-0660e7fc567957e4a VPC: vpc-0e734dd4ca9c669a3
- Listeners and routing**
 - Listener HTTP:80**

Protocol	Port
HTTP	80 1-65535

Default action: Forward to wp-TG Target type: IP, IPv4
 - Create target group**
- Listener tags - optional**
 - Add listener tag
 - You can add up to 50 more tags.
- Add listener**
- Load balancer tags - optional**
 - Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

Scroll down to the end

aws | Search [Alt+S] United States (N. Virginia) sidney @ 3247-8332-4460

[EC2](#) > [Load balancers](#) > Create Application Load Balancer

Review and confirm your configurations. [Estimate cost](#)

Basic configuration Edit	Network mapping Edit VPC: vpc-0e734dd4ca9c669a3 Public IPv4 IPAM pool: - Availability Zones and subnets: <ul style="list-style-type: none">• us-east-1a<ul style="list-style-type: none">subnet-0ff9c3e4b5b11293fdemo-VPC-subnet-public1-us-east-1a• us-east-1b<ul style="list-style-type: none">subnet-08c4b8ca5a6cee0a2demo-VPC-subnet-public2-us-east-1b	Security groups Edit wp-sg sg-0660e7fc567957e4a	Listeners and routing Edit HTTP:80 Target group: wp-TG
Service integrations Edit Amazon CloudFront + AWS Web Application Firewall (WAF): - AWS WAF: - AWS Global Accelerator: -	Tags Edit -		
Attributes	<p> Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.</p>		

Click on “Create Load Balancer”

The screenshot shows the AWS EC2 Load Balancers console. A green success message at the top states: "Successfully created load balancer: wp-alb. It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks." The main page displays the details of the newly created load balancer "wp-alb". Key information includes:

- Load balancer type:** Application
- Status:** Provisioning
- VPC:** vpc-0e734dd4ca9c669a3
- Hosted zone:** Z35SXDOTRQ7XK
- Availability Zones:** subnet-08c4b8ca5a6cee0a2 (us-east-1b (use1-az2)), subnet-Off9c3e4b5b11293f (us-east-1a (use1-az1))
- Load balancer IP address type:** IPv4
- Date created:** July 13, 2025, 13:50 (UTC-04:00)

The "Listeners and rules" tab is selected, showing one listener rule for port 80. The DNS name is wp-alb-481123596.us-east-1.elb.amazonaws.com (A Record). Other tabs include Network mapping, Resource map, Security, Monitoring, Integrations, Attributes, Capacity, and Tags.

The load balancer is being created

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar lists various services under "EC2". The main area displays a table of existing load balancers:

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
wp-alb	wp-alb-481123596.us-east-...	Provisioning...	vpc-0e734dd4ca9c669a3	2 Availability Zones	application	July 13

A message at the bottom indicates: "0 load balancers selected. Select a load balancer above."

The application load balancer is provisioning, wait for it to be active

The screenshot shows the AWS EC2 Load Balancers console. On the left, there's a navigation sidebar with various EC2-related options like Instances, Images, and Network & Security. The main area is titled "Load balancers (1/1)" and displays a single entry for "wp-alb". The table row for "wp-alb" includes columns for Name (wp-alb), DNS name (wp-alb-481123596.us-east-...), State (Active), VPC ID (vpc-0e734dd4ca9c669a3), Availability Zones (2 Availability Zones), Type (application), and Date created (July 13). Below this, a detailed view for "Load balancer: wp-alb" is shown with tabs for Details, Listeners and rules, Network mapping, Resource map, Security, Monitoring, Integrations, and Attributes. The "Details" tab is selected, showing information such as Load balancer type (Application), Status (Active), VPC (vpc-0e734dd4ca9c669a3), Hosted zone (Z35SXDOTRQ7X7K), and Load balancer IP address type (IPv4). Other tabs show Listener rules, Network mapping, and Resource maps.

The application load balancer is now “ACTIVE”.

STEP 4: Create ECS Task Definition

The next step is to create the ECS Task Definition. Go to AWS Management Console and search for “ECS”. Task definition is the blue print of your container.

The screenshot shows the AWS Management Console search results for "ECS". The search bar at the top has "ECS" typed into it. The left sidebar has a collapsed "EC2" section and expanded "Load Balancers" section, which is currently active. The main content area is titled "Services" and lists three services: "Elastic Container Service" (highlighted in orange), "Batch", and "AWS FIS". Below this, there are sections for "Features" (Get started, Clusters, Namespaces) and "Resources" (with a note about introducing resource search). At the bottom, there are buttons for "Were these results helpful?" (Yes or No) and a note about introducing resource search.

Click on “Elastic Container Service”

The screenshot shows the AWS ECS console. On the left, there's a sidebar with links like 'Clusters', 'Namespaces', 'Task definitions' (which has a red arrow pointing to it), and 'Account settings'. The main area shows a table titled 'Clusters (1)'. It has columns for 'Cluster', 'Services', 'Tasks', 'Container instances', and 'CloudWatch monitoring'. One cluster named 'wp-cluster' is listed, showing 1 service, 0 tasks running, 0 EC2 instances, and a 'Default' status. At the top right, there's a 'Create cluster' button.

Click on “Task Definition”

This screenshot shows the 'Task definitions' page within the AWS ECS console. The left sidebar includes links for 'Clusters', 'Namespaces', 'Task definitions' (with a red arrow), and 'Account settings'. The main content area displays a table titled 'Task definitions (0)'. It features a 'Filter task definitions' search bar and a 'Create new task definition' button. A message indicates 'No task definitions' and 'No task definitions to display.' The top right corner includes standard AWS navigation buttons like 'Deploy', 'Create new revision', and another 'Create new task definition' button.

Click on the drop down on “Create new task definition” and select “Create new task definition”

Amazon Elastic Container Service < Create new task definition

Task definition configuration

Task definition family | [Info](#)
Specify a unique task definition family name.

Task definition family name

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Infrastructure requirements

Specify the infrastructure requirements for the task definition.

Launch type | [Info](#)
Selection of the launch type will change task definition parameters.

AWS Fargate
Serverless compute for containers.

Amazon EC2 instances
Self-managed infrastructure using Amazon EC2 instances.

OS, Architecture, Network mode
Network mode is used for tasks and is dependent on the compute type selected.

Operating system/Architecture | [Info](#) **Network mode** | [Info](#)

Linux/X86_64 awsvpc

Task size | [Info](#)
Specify the amount of CPU and memory to reserve for your task.

CPU **Memory**

1 vCPU 3 GB

Task roles - conditional

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Give the Task Definition a name, I will call it “**wordpress-task-def**”. In “Infrastructure Requirements”, choose “**AWS Fargate**”

Amazon Elastic Container Service < Create new task definition

Task definition configuration

Task definition family | [Info](#)
Specify a unique task definition family name.

Task definition family name

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Infrastructure requirements

Specify the infrastructure requirements for the task definition.

Launch type | [Info](#)
Selection of the launch type will change task definition parameters.

AWS Fargate
Serverless compute for containers.

Amazon EC2 instances
Self-managed infrastructure using Amazon EC2 instances.

OS, Architecture, Network mode
Network mode is used for tasks and is dependent on the compute type selected.

Operating system/Architecture | [Info](#) **Network mode** | [Info](#)

Linux/X86_64 awsvpc

Task size | [Info](#)
Specify the amount of CPU and memory to reserve for your task.

CPU **Memory**

1 vCPU 3 GB

Task roles - conditional

Scroll down to “**Task Role**”

Amazon Elastic Container Service

- Clusters
- Namespaces
- Task definitions**
- Account settings

- Amazon ECR
- Repositories

- AWS Batch

Task roles - conditional

Task role | Info
A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role from the [IAM console](#).

Task execution role | Info
A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't already have a task execution IAM role created, we can create one for you.

[Create new role](#)

Task placement - optional

Task placement constraints are not supported for AWS Fargate launch type.

Fault injection - optional

Click on “IAM Console” and a new window will pop up

Identity and Access Management (IAM)

- Dashboard
- Access management**
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
 - Root access management [New](#)
- Access reports**
 - Access Analyzer
 - Resource analysis [New](#)
 - Unused access
 - Analyzer settings
 - Credential report
 - Organization activity
 - Service control policies
 - Resource control policies [New](#)

Roles (28) | Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

<input type="checkbox"/> Role name	▲ Trusted entities	Last activity
AWSRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-Linked Role)	46 days ago
AWSRoleForAPIGateway	AWS Service: ops.apigateway (Service-Linked Role)	-
AWSRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	41 days ago
AWSRoleForBackup	AWS Service: backup (Service-Linked Role)	9 hours ago
AWSRoleForECS	AWS Service: ecs (Service-Linked Role)	10 minutes ago
AWSRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	28 minutes ago
AWSRoleForOrganizations	AWS Service: organizations (Service-Linked Role)	-
AWSRoleForRDS	AWS Service: rds (Service-Linked Role)	18 minutes ago
AWSRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
demo-lambda-function-role-tqq7gcuo	AWS Service: lambda	20 days ago
demo-lambda-role-86v84k2m	AWS Service: lambda	24 days ago
demo-lambda-role-rzlp3cp8	AWS Service: lambda	22 days ago
demo-role-dskwoldz	AWS Service: lambda	11 days ago
demoFunctionRole-5luembo3	AWS Service: lambda	15 days ago

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Create Role”

Step 1
Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Select trusted entity

Trusted entity type

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Choose a service or use case

Cancel **Next**

On “Trusted Entity Type”, select “AWS Service” and on “Use Case”, search for “Elastic Container Service”. Then select “Elastic Container Service -Task”.

Add permissions

Step 3 Name, review, and create

Trusted entity type

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Elastic Container Service

Choose a use case for the specified service.

Use case

- Elastic Container Service Allows ECS to create and manage AWS resources on your behalf.
- Elastic Container Service Autoscale Allows Auto Scaling to access and update ECS services.
- Elastic Container Service Task Allows ECS tasks to call AWS services on your behalf.
- EC2 Role for Elastic Container Service Allows EC2 instances in an ECS cluster to access ECS.
- Elastic Container Service for VPC Lattice Allows access to create and manage AWS service resources required to manage VPC Lattice feature in ECS workloads

Cancel **Next**

Click on “Next”

Permissions policies (1081) [Info](#)

Choose one or more policies to attach to your new role.

Policy name	Type	Description
AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
AdministratorAccess-Amplify	AWS managed	Grants account administrative permisi...
AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permisi...
AIOpsAssistantPolicy	AWS managed	Provides ReadOnly permissions requir...
AIOpsConsoleAdminPolicy	AWS managed	Grants full access to Amazon AI Opera...
AIOpsOperatorAccess	AWS managed	Grants access to the Amazon AI Opera...
AIOpsReadOnlyAccess	AWS managed	Grants ReadOnly permissions to the A...
AlexaForBusinessDeviceSetup	AWS managed	Provide device setup access to AlexaFo...
AlexaForBusinessFullAccess	AWS managed	Grants full access to AlexaForBusiness ...
AlexaForBusinessGatewayExecution	AWS managed	Provide gateway execution access to A...
AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	Provide access to Lifesize AVS devices
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	Provide access to Poly AVS devices
AlexaForBusinessReadOnlyAccess	AWS managed	Provide read only access to AlexaForB...
AmazonAPIGatewayAdministrator	AWS managed	Provides full access to create/edit/delete...
AmazonAPIGatewayInvokeFullAccess	AWS managed	Provides full access to invoke APIs in A...
AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	Allows API Gateway to push logs to us...

The permission we will be giving to this is “**AmazonECSTaskExecutionRolePolicy**”, search for “**AmazonECSTask**”

Add permissions [Info](#)

Permissions policies (1081) [Info](#)

Choose one or more policies to attach to your new role.

Policy name	Type	Description
AmazonECSTaskExecutionRolePolicy	AWS managed	Provides access to other AWS service reso...

▶ Set permissions boundary - optional

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

Select the policy and click on “**Next**”

Sidney's screenshot of the AWS IAM 'Create role' wizard Step 3: Name, review, and create.

Role details

Role name: (highlighted with a red arrow)

```
Maximum 64 characters. Use alphanumeric and '+=_@-' characters.
```

Description:

```
Allows ECS tasks to call AWS services on your behalf.
```

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: '_+=,. @-/[\{\}]#\$%^&^0~`'.

Step 1: Select trusted entities

Trust policy:

```
1 * [ { "Version": "2012-10-17", "Statement": [ { "Sid": "", "Effect": "Allow", "Principal": { "Service": [ "ecs-tasks.amazonaws.com" ] }, "Action": "sts:AssumeRole" } ] }
```

Step 2: Add permissions

Give the role a name. I will call it “**ECSTaskExecutionRole**”

Sidney's screenshot of the AWS IAM 'Create role' wizard Step 3: Name, review, and create.

Role details

Role name: ECSTaskExecutionRole

Maximum 64 characters. Use alphanumeric and '+=_@-' characters.

Description:

```
Allows ECS tasks to call AWS services on your behalf.
```

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: '_+=,. @-/[\{\}]#\$%^&^0~`'.

Step 1: Select trusted entities

Trust policy:

```
1 * [ { "Version": "2012-10-17", "Statement": [ { "Sid": "", "Effect": "Allow", "Principal": { "Service": [ "ecs-tasks.amazonaws.com" ] }, "Action": "sts:AssumeRole" } ] }
```

12 } "Action": "sts:AssumeRole"
 13 }
 14]
 15 }

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonECSTaskExecutionRolePolicy	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional Info
 Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
 No tags associated with the resource.

Add new tag
 You can add up to 50 more tags.

Create role

Click on “Create Role”

Identity and Access Management (IAM)

Roles (29)

Role ECSTaskExecutionRole created.

Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-Linked)	46 days ago
AWSServiceRoleForAPIGateway	AWS Service: ops.apigateway (Service-Linked)	-
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked)	41 days ago
AWSServiceRoleForBackup	AWS Service: backup (Service-Linked)	9 hours ago
AWSServiceRoleForECS	AWS Service: ecs (Service-Linked Role)	30 minutes ago
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked)	48 minutes ago
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked)	-
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	37 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-
demo-lambda-function-role-tqq7gcuo	AWS Service: lambda	20 days ago
demo-lambda-role-86v84k2m	AWS Service: lambda	24 days ago
demo-lambda-role-rzlp3cp8	AWS Service: lambda	22 days ago

The role has been created. I will go back and continue with my Task definition tab

Amazon Elastic Container Service > Create new task definition

Amazon Elastic Container Service

- Clusters
- Namespaces
- Task definitions**
- Account settings

Amazon ECR

AWS Batch

Task roles - conditional

Task role | Info

A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role from the IAM console.

Task execution role | Info

A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't already have a task execution IAM role created, we can create one for you.

Create new role

Task placement - optional

Task placement constraints are not supported for AWS Fargate launch type.

Fault injection - optional

Click on the drop down and select the Task Role we just created

Amazon Elastic Container Service > Create new task definition

Amazon Elastic Container Service

- Clusters
- Namespaces
- Task definitions**
- Account settings

Amazon ECR

AWS Batch

Task roles - conditional

Task role | Info

A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role from the IAM console.

ECSTaskExecutionRole

Task execution role | Info

A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't already have a task execution IAM role created, we can create one for you.

ECSTaskExecutionRole

Task placement - optional

Task placement constraints are not supported for AWS Fargate launch type.

Fault injection - optional

Scroll down to “Container-1”

Amazon Elastic Container Service > Container - 1

Container details

Name **wordpress**

Image URI **repository-uri/image:tag**

Essential container **Yes**

Private registry | Info

Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

Private registry authentication

Port mappings | Info

Add port mappings to allow the container to access ports on the host to send or receive traffic. For port name, a default will be assigned if left blank.

Container port	Protocol	Port name	App protocol
80	TCP	container-port-protocol	HTTP

Add port mapping

Read only root file system | Info

When this parameter is turned on, the container is given read-only access to its root file system.

Read only

On the container name, enter “**wordpress**” and on the “**Image URI**”, also enter “**wordpress**” since we are using the default WordPress image.

Amazon Elastic Container Service

- Clusters
- Namespaces
- Task definitions**
- Account settings

- Amazon ECR
- Repositories

- AWS Batch

- Documentation
- Discover products
- Subscriptions

[Tell us what you think](#)

▼ Container - 1 [Info](#)

Container details
Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name	Image URI	Essential container
wordpress	wordpress	Yes

Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores are allowed.

Private registry [Info](#)
Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

Private registry authentication

Port mappings [Info](#)
Add port mappings to allow the container to access ports on the host to send or receive traffic. For port name, a default will be assigned if left blank.

Container port	Protocol	Port name	App protocol	Remove
80	TCP	container-port-protocol	HTTP	Remove

[Add port mapping](#)

Read only root file system [Info](#)
When this parameter is turned on, the container is given read-only access to its root file system.

Read only

Resource allocation limits - conditional [Info](#)
Container-level CPU, GPU, and memory limits are different from task-level values. They define how much resources are allocated for the container. If container attempts to exceed the memory specified in hard limit, the container is terminated.

CPU	GPU	Memory hard limit	Memory soft limit
1	1	3	1

Scroll down to the end

Amazon Elastic Container Service

- Clusters
- Namespaces
- Task definitions**
- Account settings

- Amazon ECR
- Repositories

- AWS Batch

- Documentation
- Discover products
- Subscriptions

[Tell us what you think](#)

▼ Storage - optional

Ephemeral storage [Info](#)
The amount of ephemeral storage, in GiB, to allocate for the task. By default, your tasks hosted on AWS Fargate receive a minimum of 20 GiB of ephemeral storage.

Amount	<input type="text" value="21"/>
---------------	---------------------------------

To specify a custom amount of ephemeral storage, specify a value between 21 GiB up to a maximum of 200 GiB.

Volumes [Info](#)
Add one or more data volumes for your task to provide additional storage for the containers in the task. For each data volume, you must add a mount point to specify where to mount the data volume in the container.

[Add volume](#)

Volumes from [Info](#)
Mount data volumes from another container.

[Add volume from](#)

► Monitoring - optional [Info](#)
Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.

► Tags - optional [Info](#)
Tags help you to identify and organize your task definitions.

[Cancel](#) [Create](#)

[CloudShell](#) [Feedback](#) © 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Click on “Create”

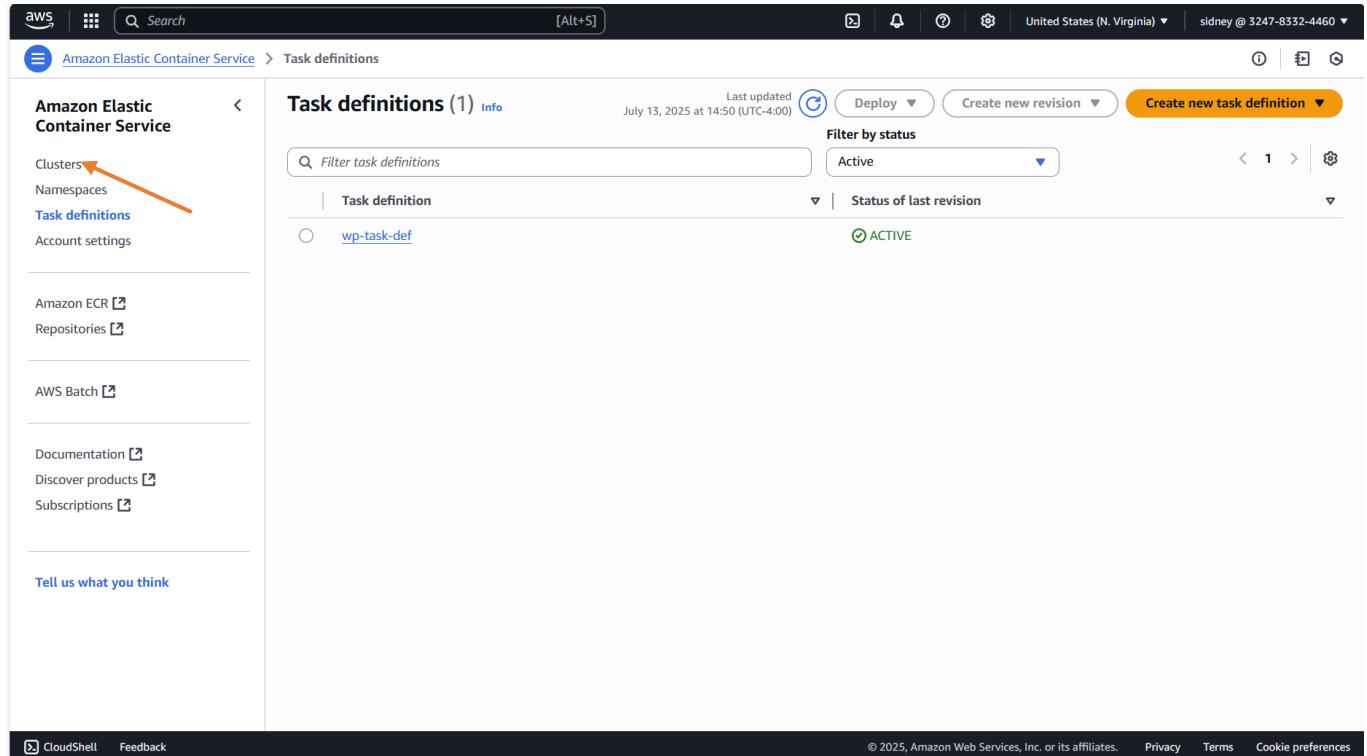
The screenshot shows the AWS ECS Task Definitions page. A green success message at the top states: "Task definition successfully created" and "wordpress-task-def:1 has been successfully created. You can use this task definition to deploy a service or run a task." The main section is titled "wordpress-task-def:1". It includes tabs for Overview, Containers, JSON, Task placement, Volumes (0), Requires attributes, and Tags. Under Overview, it shows the ARN (arn:aws:ecs:us-east-1:324783324460:task-definition/wordpress-task-def:1), Status (ACTIVE), Time created (July 13, 2025 at 18:07 (UTC-4:00)), App environment (Fargate), Task role (ECSTaskExecutionRole), Task execution role (ECSTaskExecutionRole), Operating system/Architecture (Linux/X86_64), and Network mode (awsvpc). It also shows Fault injection (Turned off). Below this, there are sections for Task size (Task CPU: 1,024 units (1 vCPU), Task CPU maximum allocation for containers: 1000 CPU units), Task memory (3,072 MiB (3 GB), Task memory maximum allocation for container memory reservation: 3000 Memory MiB), and a note that the task definition has been updated at 18:08 (UTC-4:00).

Click on “View task Definition”

The screenshot shows the AWS ECS Task Definitions page. On the left, there is a sidebar with links to Clusters, Namespaces, Task definitions (selected), Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main content area shows a list of task definitions under the heading "wordpress-task-def (1)". It includes a search bar ("Filter task definition revisions by value"), a "Last updated" timestamp (July 13, 2025 at 18:08 (UTC-4:00)), and a "Filter status" dropdown set to "Active". The list shows one item: "wordpress-task-def:1" with a status of "ACTIVE". At the bottom right, there are navigation controls for the list.

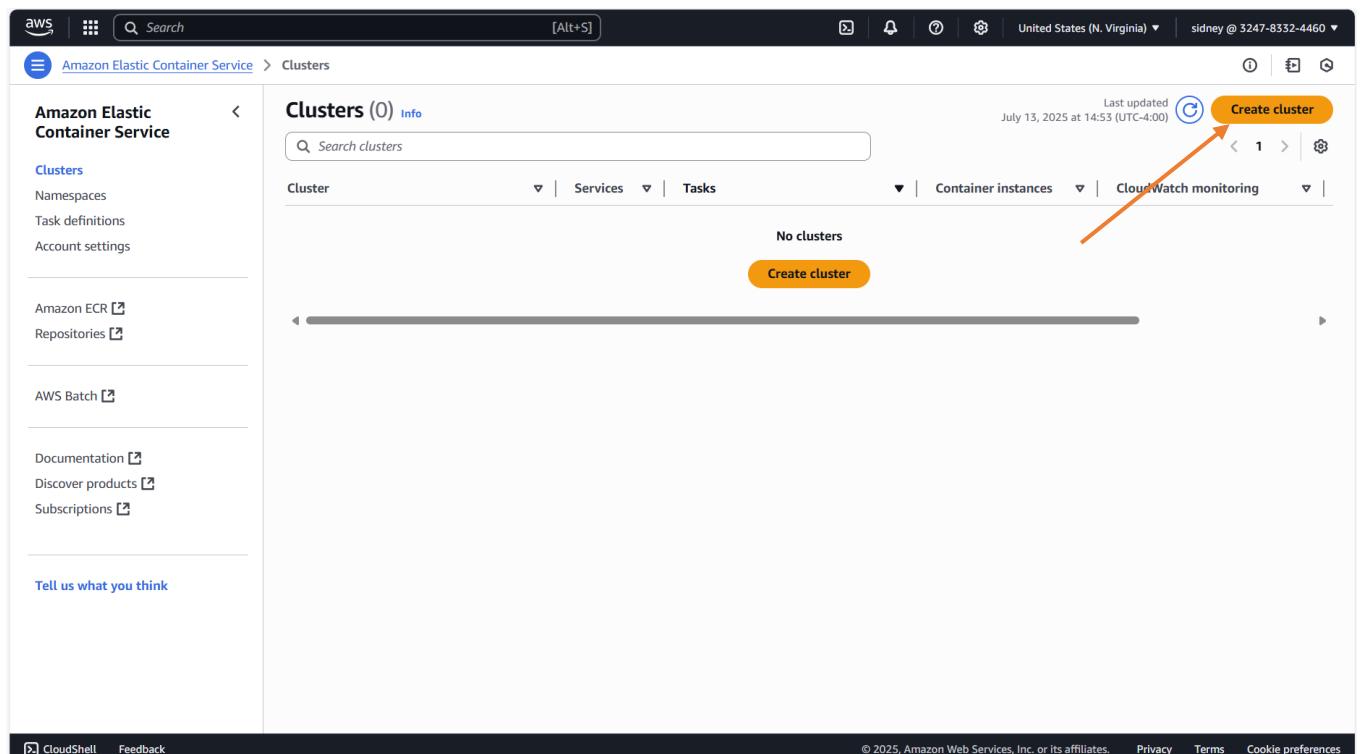
We have created the Task Definition. Let is now create the ECS Cluster.

STEP 4: Create ECS Cluster



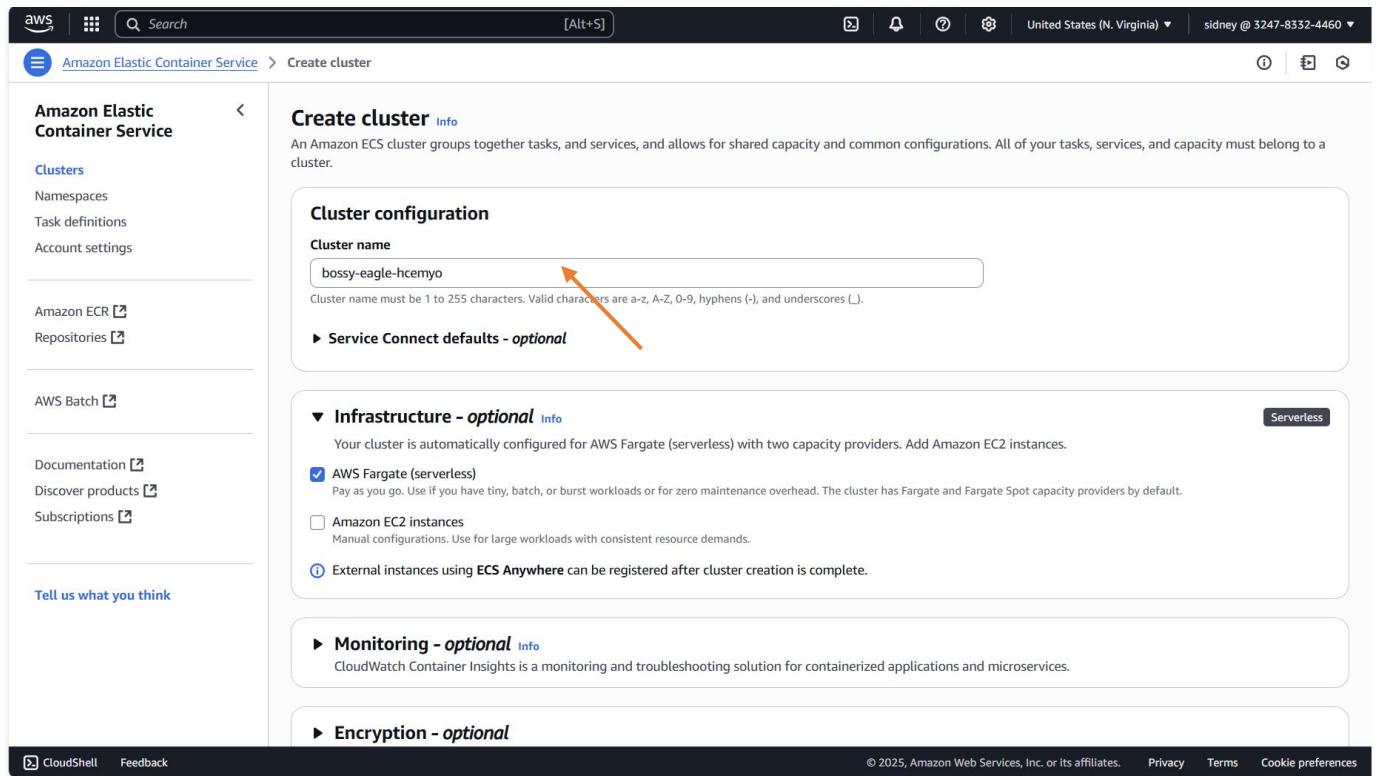
The screenshot shows the AWS Elastic Container Service (ECS) Task definitions page. On the left, there's a sidebar with links for Clusters, Namespaces, Task definitions, and Account settings. A red arrow points to the 'Clusters' link. The main area displays 'Task definitions (1)' with a single entry named 'wp-task-def' which is 'ACTIVE'. There are buttons for Deploy, Create new revision, and Create new task definition.

Click on “Clusters”



The screenshot shows the AWS Elastic Container Service (ECS) Clusters page. On the left, there's a sidebar with links for Clusters, Namespaces, Task definitions, and Account settings. A red arrow points to the 'Clusters' link. The main area displays 'Clusters (0)' with a 'Create cluster' button highlighted by a red box. The status bar at the bottom indicates 'Last updated July 13, 2025 at 14:53 (UTC-4:00)'.

Click on “Create Cluster”



Create cluster Info

An Amazon ECS cluster groups together tasks, and services, and allows for shared capacity and common configurations. All of your tasks, services, and capacity must belong to a cluster.

Cluster configuration

Cluster name

bossy-eagle-hcemyo

Cluster name must be 1 to 255 characters. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

► **Service Connect defaults - optional**

▼ **Infrastructure - optional** Info Serverless

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances.

AWS Fargate (serverless)
Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.

Amazon EC2 instances
Manual configurations. Use for large workloads with consistent resource demands.

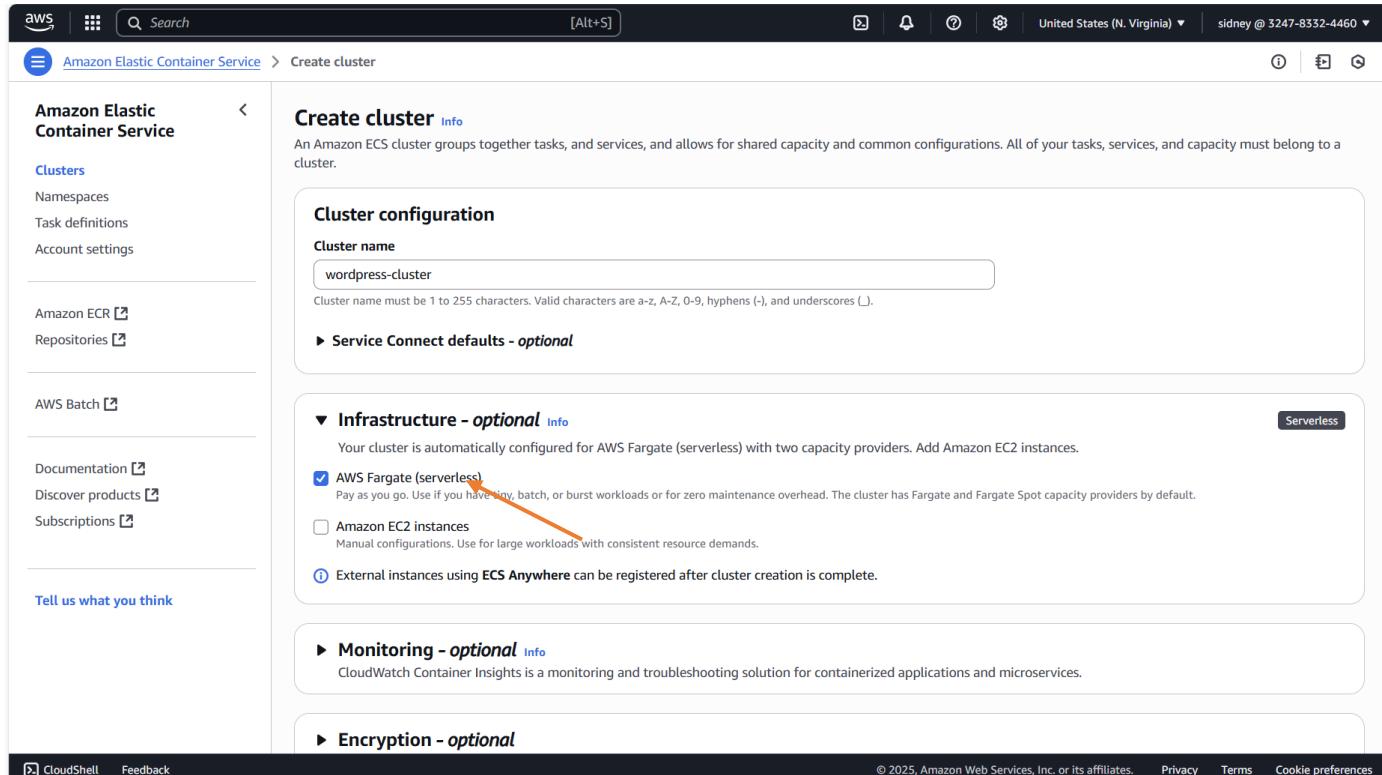
ⓘ External instances using ECS Anywhere can be registered after cluster creation is complete.

► **Monitoring - optional** Info
CloudWatch Container Insights is a monitoring and troubleshooting solution for containerized applications and microservices.

► **Encryption - optional**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Give the cluster a name, I will call it “**wordpress-cluster**”



Create cluster Info

An Amazon ECS cluster groups together tasks, and services, and allows for shared capacity and common configurations. All of your tasks, services, and capacity must belong to a cluster.

Cluster configuration

Cluster name

wordpress-cluster

Cluster name must be 1 to 255 characters. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

► **Service Connect defaults - optional**

▼ **Infrastructure - optional** Info Serverless

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances.

AWS Fargate (serverless)
Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.

Amazon EC2 instances
Manual configurations. Use for large workloads with consistent resource demands.

ⓘ External instances using ECS Anywhere can be registered after cluster creation is complete.

► **Monitoring - optional** Info
CloudWatch Container Insights is a monitoring and troubleshooting solution for containerized applications and microservices.

► **Encryption - optional**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

On “**Infrastructure – Optional**”, we will use “**AWS Fargate**”. Scroll down to the end

Amazon Elastic Container Service > Create cluster

Cluster name must be 1 to 255 characters. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

► Service Connect defaults - *optional*

▼ Infrastructure - *optional* Info Serverless

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances.

AWS Fargate (serverless)
Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.

Amazon EC2 instances
Manual configurations. Use for large workloads with consistent resource demands.

ⓘ External instances using ECS Anywhere can be registered after cluster creation is complete.

► Monitoring - *optional* Info

CloudWatch Container Insights is a monitoring and troubleshooting solution for containerized applications and microservices.

► Encryption - *optional*

Choose the KMS keys used by tasks running in this cluster to encrypt your storage.

► Tags - *optional* Info

Tags help you to identify and organize your clusters.

Cancel Create

Click on “Create”

Amazon Elastic Container Service > Clusters

Cluster wordpress-cluster creation is in progress. [View in CloudFormation](#) X

Clusters (0) Info

Last updated July 13, 2025 at 18:10 (UTC-4:00)

Create cluster

Cluster	Services	Tasks	Container instances	CloudWatch monitoring
No clusters				

Create cluster

The cluster is being created

AWS | Search [Alt+S] | United States (N. Virginia) | sidney @ 3247-8332-4460 | Amazon Elastic Container Service > Clusters

Clusters (1) Last updated July 13, 2025 at 18:11 (UTC-4:00)

Cluster Services Tasks Container instances CloudWatch monitoring

[wordpress-cluster](#) 0 No tasks running 0 EC2 Default

Clusters Namespaces Task definitions Account settings

Amazon ECR Repositories

AWS Batch

Documentation Discover products Subscriptions

Tell us what you think

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

STEP 6: Create ECS Service

The cluster has been created, click on the cluster name

AWS | Search [Alt+S] | United States (N. Virginia) | sidney @ 3247-8332-4460 | Amazon Elastic Container Service > Clusters > [wordpress-cluster](#) > Services

wordpress-cluster Last updated July 13, 2025 at 18:12 (UTC-4:00)

Cluster overview

ARN arn:aws:ecs:us-east-1:32478332446 0:cluster/wordpress-cluster	Status Active	CloudWatch monitoring Default	Registered container instances -
---	------------------	----------------------------------	-------------------------------------

Services

Draining -	Active -	Pending -	Running -
---------------	-------------	--------------	--------------

Tasks

Draining -	Active -	Pending -	Running -
---------------	-------------	--------------	--------------

Services (0) Last updated July 13, 2025 at 18:12 (UTC-4:00)

Manage tags Update Delete service Create

Filter launch type Any launch type Filter service type Any service type

Service name ARN Status Service... Created at Deployments and tasks

No services
No services to display.

Create

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Create”

The screenshot shows the 'Create service' page for the Amazon Elastic Container Service. On the left, there's a sidebar with links like Clusters, Namespaces, Task definitions, and Account settings. The main area has a title 'Create service' with a 'Info' link. Under 'Service details', there's a 'Task definition family' dropdown menu with a blue arrow pointing to it. Below it is a 'Task definition revision' input field with a search icon and a 'Service name' input field containing 'wordpress-service'. In the 'Compute configuration (advanced)' section, 'Capacity provider strategy' is selected. At the bottom right, there's an 'AWS Fargate' button.

Click on the drop down and select the Task definition we created

This screenshot shows the same 'Create service' page as the previous one, but with changes made to the inputs. The 'Task definition family' dropdown now shows 'wordpress-task-def'. The 'Service name' input field now contains 'wordpress-service'. The rest of the page, including the 'Compute configuration (advanced)' section, remains identical to the first screenshot.

Let is give the service a name, I will call it “wordpress-service”

Scroll down to “Networking” and click on it

Select the VPC we created

Amazon Elastic Container Service > Clusters > wp-cluster > Create service

Networking

VPC | **Info**
Select a VPC to use for your Amazon ECS resources.
vpc-0e734dd4ca9c669a3
demo-VPC-vpc

Subnets
Choose the subnets within the VPC that the task scheduler should consider for placement.

subnet-08c4b8ca5a6cee0a2 public
demo-VPC-subnet-public2-us-east-1b
us-east-1b 10.0.16.0/20

subnet-0ff9c3e4b5b11293f public
demo-VPC-subnet-public1-us-east-1a
us-east-1a 10.0.0.0/20

subnet-0fd9abe69173e3f9d
demo-VPC-subnet-private2-us-east-1b
us-east-1b 10.0.144.0/20

subnet-00441c0e0bbfe1d73
demo-VPC-subnet-private1-us-east-1a
us-east-1a 10.0.128.0/20

Security group | **Info**
Choose an existing security group or create a new security group.
 Use an existing security group
 Create a new security group

Security group name
Choose an existing security group.

sg-00735a2001e6bea2f
default

Public IP | **Info**
Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).
 Turned on

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on the drop down and select the security group we created for this project

Amazon Elastic Container Service > Clusters > wp-cluster > Create service

Networking

VPC | **Info**
Select a VPC to use for your Amazon ECS resources.
vpc-0e734dd4ca9c669a3
demo-VPC-vpc

Subnets
Choose the subnets within the VPC that the task scheduler should consider for placement.

subnet-08c4b8ca5a6cee0a2 public
demo-VPC-subnet-public2-us-east-1b
us-east-1b 10.0.16.0/20

subnet-0ff9c3e4b5b11293f public
demo-VPC-subnet-public1-us-east-1a
us-east-1a 10.0.0.0/20

subnet-0fd9abe69173e3f9d
demo-VPC-subnet-private2-us-east-1b
us-east-1b 10.0.144.0/20

subnet-00441c0e0bbfe1d73
demo-VPC-subnet-private1-us-east-1a
us-east-1a 10.0.128.0/20

Security group | **Info**
Choose an existing security group or create a new security group.
 Use an existing security group
 Create a new security group

Security group name
Choose an existing security group.

sg-0660e7fc567957e4a
wp-sg

Public IP | **Info**
Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).
 Turned on

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Scroll down to “Load Balancer”

Amazon Elastic Container Service

- Clusters**
 - Namespaces
 - Task definitions
 - Account settings
- Amazon ECR**
 - Repositories
- AWS Batch**
- Documentation**
 - Discover products
 - Subscriptions

Tell us what you think

► Service Connect - optional Info
Service Connect allows for service-to-service communications with automatic discovery using short names and standard ports.

► Service discovery - optional
Service discovery uses Amazon Route 53 to create a namespace for your service, which allows it to be discoverable via DNS.

▼ Load balancing - optional
Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

Use load balancing

► VPC Lattice - optional Info
Fully managed application networking service to connect, secure, and monitor your services across multiple accounts and virtual private clouds (VPCs). When you use VPC Lattice, there is a cost associated with it.

► Service auto scaling - optional
Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

► Volume - optional Info
Configure a data volume to provide additional storage for the containers in the task.

Check the box on “Use Load Balancing”

Amazon Elastic Container Service

- Clusters**
 - Namespaces
 - Task definitions
 - Account settings
- Amazon ECR**
 - Repositories
- AWS Batch**
- Documentation**
 - Discover products
 - Subscriptions

Tell us what you think

▼ Load balancing - optional
Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

Use load balancing

VPC
The VPC for your load balancing resources must be the same as the VPC for your service with awsvpc.

Load balancer type Info
Specify the load balancer type to distribute incoming traffic across the tasks running in your service.

Application Load Balancer
An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports.

Network Load Balancer
A Network Load Balancer makes routing decisions at the transport layer (TCP/UDP).

Container
The container and port to load balance the incoming traffic to

Application Load Balancer
Specify whether to create a new load balancer or choose an existing one.

Create a new load balancer

Use an existing load balancer

Load balancer name
Assign a unique name for the load balancer.

Listener Info

Select “Use an existing load balancer”

Amazon Elastic Container Service

Clusters

Namespaces

Task definitions

Account settings

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

Tell us what you think

CloudShell Feedback

Load balancing - optional

Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

Use load balancing

VPC

The VPC for your load balancing resources must be the same as the VPC for your service with awsvpc.

vpc-0e734dd4ca9c669a3

Load balancer type | Info

Specify the load balancer type to distribute incoming traffic across the tasks running in your service.

Application Load Balancer
An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports.

Network Load Balancer
A Network Load Balancer makes routing decisions at the transport layer (TCP/UDP).

Container

The container and port to load balance the incoming traffic to

wordpress 80:80

Host port:Container port

Application Load Balancer

Specify whether to create a new load balancer or choose an existing one.

Create a new load balancer

Use an existing load balancer

Load balancer

Choose an existing load balancer to distribute traffic. View existing load balancers and create new one in EC2 Console

Choose an existing load balancer

Listener | Info

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on the drop down and select the Application Load Balancer we created

Amazon Elastic Container Service

Clusters

Namespaces

Task definitions

Account settings

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

Tell us what you think

Load balancing - optional

Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

Use load balancing

VPC

The VPC for your load balancing resources must be the same as the VPC for your service with awsvpc.

vpc-0e734dd4ca9c669a3

Load balancer type | Info

Specify the load balancer type to distribute incoming traffic across the tasks running in your service.

Application Load Balancer
An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports.

Network Load Balancer
A Network Load Balancer makes routing decisions at the transport layer (TCP/UDP).

Container

The container and port to load balance the incoming traffic to

wordpress 80:80

Host port:Container port

Application Load Balancer

Specify whether to create a new load balancer or choose an existing one.

Create a new load balancer

Use an existing load balancer

Load balancer

Choose an existing load balancer to distribute traffic. View existing load balancers and create new one in EC2 Console

wp-alb
wp-alb-481123596.us-east-1.elb.amazonaws.com

internet-facing

Listener | Info

Scroll down to “Listener”

Listener | Info
Specify the port and protocol that the load balancer will listen for connection requests on.

Create new listener
 Use an existing listener

Port
80

Protocol
HTTP

Target group | Info
Specify whether to create a new target group or choose an existing one that the load balancer will use to route requests to the tasks in your service.

Create new target group
 Use an existing target group

Target group name

Protocol
HTTP

Deregistration delay
The amount of time to wait before the state of a deregistering target changes from draining to unused.
300 seconds

Health check protocol | Info
HTTP

Health check path | Info
/

On “Listener”, select “Use an existing listener”, then click on the drop down and select “HTTP:80”.

And on “Target Group”, select “Use an existing Target Group”, then click on the drop down and select the target group we created.

Listener | Info
Specify the port and protocol that the load balancer will listen for connection requests on.

Create new listener
 Use an existing listener

Listener
HTTP:80

Listener rules for 80:HTTP (1)
Traffic received by the listener is routed according to its rules. Rules are evaluated in priority order, from the lowest value to the highest value. The default rule is evaluated last.

Priority	Rule path	Target group
default	/	wp-TG

Target group | Info
Specify whether to create a new target group or choose an existing one that the load balancer will use to route requests to the tasks in your service.

Create new target group
 Use an existing target group

Target group name
wp-TG

Health check path
/

Health check protocol | Info
HTTP

Scroll down to the end

Amazon Elastic Container Service

Clusters

Namespaces
Task definitions
Account settings

Amazon ECR ▾
Repositories ▾

AWS Batch ▾

Documentation ▾
Discover products ▾
Subscriptions ▾

Tell us what you think

Task Definition

Step 3: Task Definition

Task definition name: wp-cluster

Task definition ARN: arn:aws:ecs:us-east-1:32478332446:task-definition/wp-cluster

Task definition revision: 1

Task definition family: wp-cluster

Task definition configuration:

Container definitions:

Container 1: wp

Image: arn:aws:lambda:us-east-1:32478332446:function:wp-lambda

Memory: 512

CPU: 256

Environment variables:

Health check path: /

Health check protocol: HTTP

VPC Lattice - optional Info
Fully managed application networking service to connect, secure, and monitor your services across multiple accounts and virtual private clouds (VPCs). When you use VPC Lattice, there is a cost associated with it.

Service auto scaling - optional
Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

Volume - optional Info
Configure a data volume to provide additional storage for the containers in the task.

Tags - optional Info
Tags help you to identify and organize your resources.

Create **Cancel**

Click on “Create”

Amazon Elastic Container Service

Clusters

Namespaces
Task definitions
Account settings

Amazon ECR ▾
Repositories ▾

AWS Batch ▾

Documentation ▾
Discover products ▾
Subscriptions ▾

Tell us what you think

Cluster overview

wp-cluster

Last updated July 13, 2025 at 15:30 (UTC-4:00)

Services

Draining	Status	Tasks	Registered container instances
-	Active	Pending	-
-	-	-	-

Services **Tasks** **Infrastructure** **Metrics** **Scheduled tasks** **Configuration** **Tags**

Services (0) Info

Create

Filter launch type: Any launch type

Filter service type: Any service type

Service name: ARN Status Service... Created at Deployments and tasks

No services

No services to display.

Create

The service is being created

The screenshot shows the AWS Elastic Container Service (ECS) Cluster Overview page. At the top, a blue banner states: "wordpress-service deployment is in progress. It takes a few minutes." Below this, the cluster name "wordpress-cluster" is displayed, along with a "View in CloudFormation" button and "Update cluster" and "Delete cluster" buttons. The main section is titled "Cluster overview" and includes tabs for ARN, Status, CloudWatch monitoring, and Registered container instances. It also shows "Services" and "Tasks" sections. The "Services" tab is selected, showing one service named "wordpress-service" with ARN arn:aws:ecs:us-east-1:32478332446, status Active, and type REPLICA. The "Tasks" tab shows one pending task. At the bottom, there are tabs for Services, Tasks, Infrastructure, Metrics, Scheduled tasks, Configuration, and Tags.

Our service has been created. Click on the service name

The screenshot shows the AWS Elastic Container Service (ECS) Service Overview page for the "wordpress-service" within the "wordpress-cluster". At the top, a blue banner states: "Last updated July 13, 2025 at 18:21 (UTC-4:00)". Below this, the service name "wordpress-service" is displayed, along with "Delete service" and "Update service" buttons. The main section is titled "Service overview" and includes tabs for Status, Tasks, Task definition, Deployment status, and Configuration and networking. The "Status" tab is selected, showing the service is Active with 1 Desired task (0 Pending | 2 Running). The "Task definition" tab shows "revision wordpress-task-def:1". The "Deployment status" tab shows "In progress". The "Configuration and networking" tab is selected, showing details for the service name (wordpress-service), Service ARN (arn:aws:ecs:us-east-1:32478332446:service/wordpress-cluster/wordpress-service), Deployments current state (2 Completed tasks), and Created at (July 13, 2025 at 18:18 (UTC-4:00)). Below this, the "Health and metrics" tab is selected, showing a table for Load balancer health with rows for wp-alb (Application Load Balancer), wordpress:80 (Container name:port), HTTP:80 (Listener), and wp-TG (Target group). The "Health" tab is also visible at the bottom.

Click on “Configuration and Networking”

Screenshot of the AWS Elastic Container Service (ECS) Configuration page for a WordPress service.

Amazon Elastic Container Service > **Clusters** > **wordpress-cluster** > **Services** > **wordpress-service** > **Configuration**

Capacity provider strategy (1)

Capacity provider	Base	Weight
FARGATE	0	1

▶ Fargate ephemeral storage

Network configuration

- VPC**: [vpc-0e734dd4ca9c669a3](#)
- Subnets**: [subnet-08c4b8ca5a6cee0a2](#), [subnet-0ff9c3e4b5b11293f](#), [subnet-0fd9abef9173e3f9d](#), [subnet-00441c0e0bbfe1d73](#)
- Security groups**: [sg-0ca6fe563eb50634b](#)
- Auto-assign public IP**: Turned on
- Service role**: [AWSserviceRoleForECS](#)
- Health check grace period**: 0 seconds
- DNS names**: [wp-alb-1395455939.us-east-1.elb.amazonaws.com](#) | [open address](#)

https://us-east-1.console.aws.amazon.com/ecs/v2/clusters/wordpress-cluster/services/wordpress-service/configuration?region=us-east-1

Copy the DNS name and paste on your web browser

Screenshot of a web browser showing the WordPress setup configuration page.

The URL in the address bar is: <https://wp-alb-1395455939.us-east-1.elb.amazonaws.com/wp-admin/setup-config.php>

The page displays a language selection dropdown menu:

- English (United States)
- Afrikaans
- አማርኛ
- Aragonés
- العربية
- العربية المغاربية
- অসমীয়া
- گوئىچى ئۇرپاجان
- Azerbaiyan dili
- Беларуская мова
- Български
- বাংলা
- ଓଡ଼ିଆ
- Bosanski
- Català
- Cebuano
- Čeština
- Cymraeg
- Dansk
- Deutsch (Schweiz)
- Deutsch (Schweiz, Du)

A blue "Continue" button is visible at the bottom of the dropdown menu.

We have been able to successfully deploy our WordPress site using ECS. We started by creating our own VPC, Security Group, our load balancer, our target group. We also created our Task definition in ECS, our cluster and our service.