

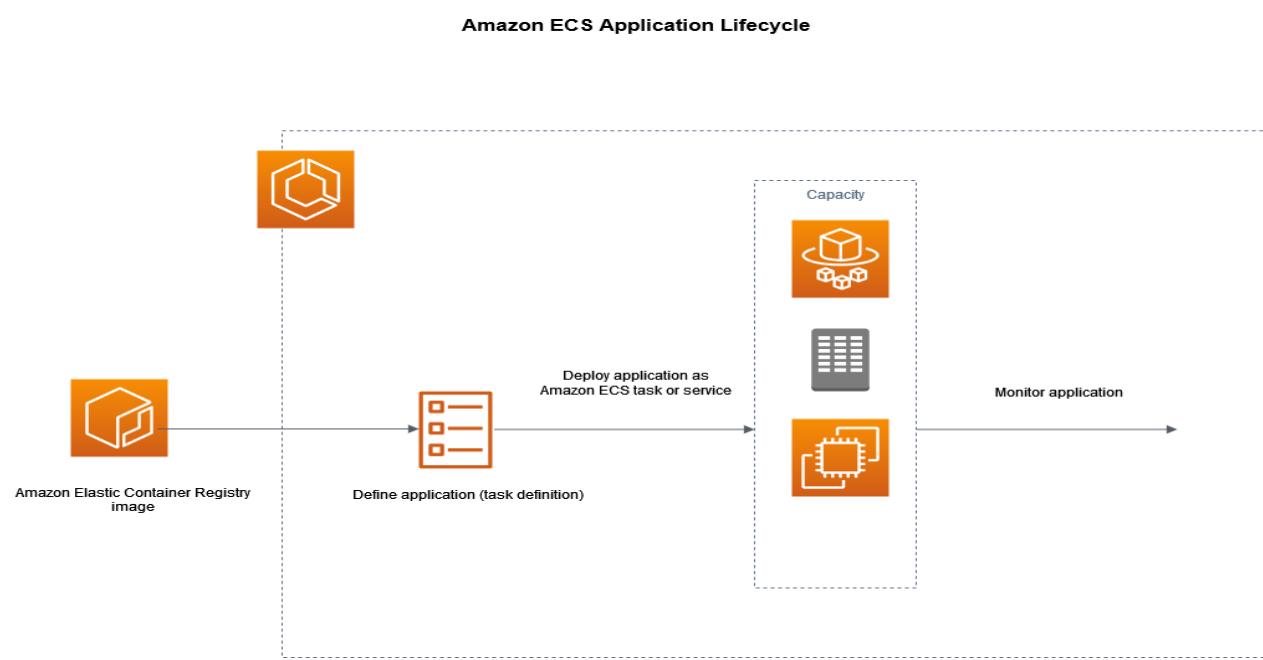
Deploy Serverless Applications

ECS-[Elastic Container Service]

Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service that helps you easily deploy, manage, and scale containerized applications.

Application lifecycle

The following diagram shows the application lifecycle and how it works with the Amazon ECS components.



Steps:

- Launch a Linux instance

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. In the 'Name and tags' section, the name 'myLinux' is entered. In the 'Application and OS Images (Amazon Machine Image)' section, the search bar contains 'Search our full catalog including 1000s of application and OS images'. The 'Quick Start' tab is selected. On the right, the 'Summary' section shows 1 instance selected, the software image is 'Amazon Linux 2023 AMI 2023.6.2...', the virtual server type is 't2.micro', and the firewall is set to 'New security group'. Buttons for 'Cancel', 'Launch instance', and 'Preview code' are visible at the bottom.

- Connect with instance

The screenshot shows the AWS EC2 Connect to instance page for an instance with ID i-063fcc5e4bb8c12a4. The 'SSH client' tab is selected. It provides instructions for connecting via SSH:

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is LinuxKeyPair.pem
- Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "LinuxKeyPair.pem"
- Connect to your instance using its Public DNS:
ec2-3-108-51-27.ap-south-1.compute.amazonaws.com

Example command:
ssh -i "LinuxKeyPair.pem" ec2-user@ec2-3-108-51-27.ap-south-1.compute.amazonaws.com

A note at the bottom states: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username."

The browser address bar shows: ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-063fcc5e4bb8c12a4

The operating system taskbar at the bottom includes icons for CloudShell, Feedback, Search, and various system status indicators like battery level and network.

- --> Install Docker

```
root@ip-172-31-1-38:/home/ec2-user
[ec2-user@ip-172-31-1-38 ~]$ sudo su
[root@ip-172-31-1-38 ec2-user]# yum update -y
Last metadata expiration check: 0:02:46 ago on Tue Dec 31 06:10:10 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-1-38 ec2-user]# yum install docker -y
Last metadata expiration check: 0:02:56 ago on Tue Dec 31 06:10:10 2024.
Dependencies resolved.

Transaction Summary
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
  docker.x86_64 25.0.6-1.amzn2023.0.2 amazonlinux 44 M
  containerd.x86_64 1.7.23-1.amzn2023.0.1 amazonlinux 36 M
  iptables-libx86_64 1.8.8-3.amzn2023.0.2 amazonlinux 401 k
  iptables-nft.x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 183 k
  libcgroup.x86_64 3.0-1.amzn2023.0.1 amazonlinux 75 k
  libnetfilter_conntrack.x86_64 1.0.8-2.amzn2023.0.2 amazonlinux 58 k
  libnftnl.x86_64 1.0.1-19.amzn2023.0.2 amazonlinux 30 k
  libnftnl.x86_64 1.2.2-2.amzn2023.0.2 amazonlinux 84 k
  pigz.x86_64 2.5-1.amzn2023.0.3 amazonlinux 83 k
  runc.x86_64 1.1.14-1.amzn2023.0.1 amazonlinux 3.2 M

Transaction Summary
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
  docker.x86_64 25.0.6-1.amzn2023.0.2 amazonlinux 44 M
  containerd.x86_64 1.7.23-1.amzn2023.0.1 amazonlinux 36 M
  iptables-libx86_64 1.8.8-3.amzn2023.0.2 amazonlinux 401 k
  iptables-nft.x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 183 k
  libcgroup.x86_64 3.0-1.amzn2023.0.1 amazonlinux 75 k
  libnetfilter_conntrack.x86_64 1.0.8-2.amzn2023.0.2 amazonlinux 58 k
  libnftnl.x86_64 1.0.1-19.amzn2023.0.2 amazonlinux 30 k
  libnftnl.x86_64 1.2.2-2.amzn2023.0.2 amazonlinux 84 k
  pigz.x86_64 2.5-1.amzn2023.0.3 amazonlinux 83 k
  runc.x86_64 1.1.14-1.amzn2023.0.1 amazonlinux 3.2 M
```

The terminal output shows the user logging in as root, updating the package repository, installing Docker, and listing the installed packages. The packages listed are containerd, docker, iptables-libx86_64, iptables-nft, libcgroup, libnetfilter_conntrack, libnftnl, pigz, and runc. The transaction summary indicates 10 packages were installed, totaling 317 MB.

--> And create Image

```
[root@ip-172-31-3-247 ~]# service docker start
[root@ip-172-31-3-247 ~]# vi Dockerfile
[root@ip-172-31-3-247 ~]# vi index.html
[root@ip-172-31-3-247 ~]# cat index.html
<html>
    <h1>Welcome to ECR</h1>
</html>

[root@ip-172-31-3-247 ~]# cat Dockerfile
FROM ubuntu
RUN apt update -y
RUN apt-get install apache2 -y
COPY index.html /var/www/html
EXPOSE 80
CMD ["apachectl","-D",""FOREGROUND"]

[root@ip-172-31-3-247 ~]# docker build -t img1:v1 .
[+] Building 26.8s (9/9) FINISHED
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 236B
--> [internal] load metadata for docker.io/library/ubuntu:latest
--> [internal] load .dockerignore
--> => transferring context: 2B
--> [1/4] FROM docker.io/library/ubuntu:latest@sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab
--> => resolve docker.io/library/ubuntu@latest@sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab
--> => sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab 0.0s
--> => sha256:6e75a10070bfcb0bead763c5118a369bc7cc30dfc1b749c49lbb2lf15c3c7 424B / 424B 0.0s
--> => sha256:bd1d9f0ab8155947974e522b380878cf9ba82d4cfb6729bcf931c3aa69ae4 2.30kB / 2.30kB 0.0s
--> => sha256:de44b265507ae44b212defcb50694d666f136b35c1090d9709068bc861bb2d64 29.75MB / 29.75MB 0.6s
--> => extracting sha256:de44b265507ae44b212defcb50694d666f136b35c1090d9709068bc861bb2d64 1.5s
--> [internal] load build context
--> => transferring context: 139B 0.0s
[2/4] RUN apt update -y 6.7s
[3/4] RUN apt-get install apache2 -y 13.7s
[4/4] COPY index.html /var/www/html 0.1s
--> exporting image 1.8s
--> exporting layers 1.8s
--> => writing image sha256:715975a90057cd8b1040f288ad37a3f06dbcf946fd260dd5e3f77e47aa2a70c 0.0s
--> => naming to docker.io/library/img1:v1 0.0s
[root@ip-172-31-3-247 ~]# docker images
REPOSITORY        TAG          IMAGE ID         CREATED          SIZE
img1              v1           715975a90057   13 seconds ago   229MB
[root@ip-172-31-3-247 ~]#
```

--> create repository in ECR--> provide name

The screenshot shows the AWS ECR Private registry Repositories page. A green success message at the top states "Successfully created myrepo11". Below it, a table lists one repository:

Repository name	URI	Created at	Tag immutability	Encryption type
myrepo11	637423422597.dkr.ecr.ap-south-1.amazonaws.com/myrepo11	December 29, 2024, 17:01:44 (UTC+05:5)	Mutable	AES-256

The left sidebar includes sections for Private registry (Repositories, Features & Settings) and Public registry (Repositories, Settings). External links for ECR public gallery, Amazon ECS, and Amazon EKS are also present.

--> goin repo name --> select viewpush command nd copy paste the commands inside putty after configuring AWS

The screenshot shows the AWS ECR Private registry Images page for the "myrepo11" repository. A blue info message at the top says "Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag." Below it, a table lists images:

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
No images No images to display					

The left sidebar includes sections for Private registry (Images, Permissions, Lifecycle Policy, Repository tags) and Public registry (Repositories, Settings). External links for ECR public gallery are also present.

Started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

```
aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin
637423422597.dkr.ecr.ap-south-1.amazonaws.com
```

Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
docker build -t myrepo11 .
```
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag myrepo11:latest 637423422597.dkr.ecr.ap-south-1.amazonaws.com/myrepo11:latest
```
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 637423422597.dkr.ecr.ap-south-1.amazonaws.com/myrepo11:latest
```

Copied

```
root@ip-172-31-3-247:/home/ec2-user
-> [internal] load metadata for docker.io/library/ubuntu:latest
-> [internal] load .dockerrignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/ubuntu:latest@sha256:80dd3c09c6cecb9f167e9290b3bc61b70c2678c02bdः
-> => resolving docker.io/library/ubuntu:latest@sha256:80dd3c09c6cecb9f167e9290b3bc61b70c2678c02bdः
-> => sha256:80dd3c1b9c6cecb9f167e9290b3bc61b70c2678c02bdः
-> => sha256:ee75a10070bfc0:head@763c5118a369bc7ec30dfc1b0743c491bb21f15c3c7 424B / 424B
-> => sha256:b1d9df8ab81559494794e522b38087cf9ba82d4c1fb67293cf931c3a69ae4 2.30kB / 2.30kB
-> => sha256:de44b265507ae44b212defcb50694d666f136b35c1090d9709068bc861bb2d64 29.75MB / 29.75MB
-> => extracting sha256:de44b265507ae44b212defcb50694d666f136b35c1090d9709068bc861bb2d64
-> [internal] load build context
-> => transferring context: 139B
-> [2/4] RUN apt update -y
-> [3/4] RUN apt-get install apache2 -y
-> [4/4] COPY index.html /var/www/html
-> exporting to image
-> => exporting layers
-> => writing image sha256:15975a90057cd8b1040f288ad37a3f06dbcfa946fd260dd5e3f77e47aa2a70c
-> => naming to docker.io/library/img1:v1
[root@ip-172-31-3-247 ec2-user]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
img1 v1 715975a90057 13 seconds ago 229MB
[root@ip-172-31-3-247 ec2-user]# aws configure
AWS Access Key ID [None]: AKIAZ12LFRSCYROXUVKV
AWS Secret Access Key [None]: gsiV5qd4sbyWpaz6VfU43pOBWRKtYGFvTbBh/ECW
Default region name [None]:
Default output format [None]:
[root@ip-172-31-3-247 ec2-user]#
[root@ip-172-31-3-247 ec2-user]# aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 637423422597.dkr.ecr.ap-south-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@ip-172-31-3-247 ec2-user]# docker tag img1:v1 637423422597.dkr.ecr.ap-south-1.amazonaws.com/myrepo11:v1
[root@ip-172-31-3-247 ec2-user]# docker push 637423422597.dkr.ecr.ap-south-1.amazonaws.com/myrepo11:v1
The push refers to repository [637423422597.dkr.ecr.ap-south-1.amazonaws.com/myrepo11]
681e1057e708: Pushed
aaccd5a27c1c: Pushed
8f391455f10a: Pushed
687d50f2f6a6: Pushed
v1: digest: sha256:0ba05f82bc0ba8a5c30c541474bbce2ea588b66c5afalacbb3e407636781c9c1 size: 1160
[root@ip-172-31-3-247 ec2-user]#
```

Amazon Elastic Container Registry

Private registry

Repositories

Images

Permissions

Lifecycle Policy

Repository tags

Features & Settings

Public registry

Repositories

Settings

ECR public gallery

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

close putty -->
--> Search ECS

Amazon Elastic Container Service

Clusters Updated

Namespaces

Task definitions

Account settings Updated

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

CloudShell Feedback

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

--> Create Cluster

The screenshot shows the AWS Elastic Container Service (ECS) Clusters page. The URL in the browser is `ap-south-1.console.aws.amazon.com/ecs/v2/clusters?region=ap-south-1`. The page title is "Clusters (0)". A search bar at the top says "Search clusters". On the left, there's a sidebar with navigation links: "Amazon Elastic Container Service" (selected), "Clusters Updated", "Namespaces", "Task definitions", "Account settings Updated", "Install AWS Copilot", "Amazon ECR", and "Repositories". The main content area has a header with "Cluster", "Services", "Tasks", "Container instances", and "CloudWatch m". Below this, it says "No clusters" and "No clusters to display". The bottom of the screen shows the AWS navigation bar with links for "Privacy", "Terms", and "Cookie preferences". The status bar at the bottom right shows the date "01-01-2025" and time "19:37".

-->provide name

-->under Infrastructure select AWS fargate which is serverless

The screenshot shows the 'Create cluster' configuration page for Amazon Elastic Container Service (ECS). The left sidebar lists navigation options like Clusters (Updated), Namespaces, Task definitions, Account settings (Updated), and Install AWS Copilot. The main content area is titled 'Cluster configuration' and contains fields for 'Cluster name' (set to 'mycluster13') and 'Default namespace - optional' (set to 'mycluster13'). Below this, a section titled 'Infrastructure' includes a checked checkbox for 'AWS Fargate (serverless)' and an unchecked checkbox for 'Amazon EC2 instances'. A 'Serverless' button is also present in this section. The top right corner shows the AWS logo, the region 'Asia Pacific (Mumbai)', and a user named 'Pratiksha'. The bottom navigation bar includes CloudShell, Feedback, and links for Privacy, Terms, and Cookie preferences.

--> Select Create--> Go in Task definition--> create new task defn 2--> give family name--> choose launch type as AWS fargate

Create task definition | Elastic C X +

ap-south-1.console.aws.amazon.com/ecs/v2/create-task-definition?region=ap-south-1

aws Search [Alt+S] Mumbai Pratiksha

Amazon Elastic Container Service > Create new task definition

Clusters Updated Namespaces Task definitions Account settings Updated

Install AWS Copilot

Amazon ECR Repositories

AWS Batch

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 19:47 01-01-2025

View cluster

Cluster mycluster13 has been created successfully.

Create new task definition Info

Task definition configuration

Task definition family Info
Specify a unique 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.

--> Inside Task size select 0.5CPU and 1 gb memory-->

Create task definition | Elastic C X +

ap-south-1.console.aws.amazon.com/ecs/v2/create-task-definition?region=ap-south-1

aws Search [Alt+S] Mumbai Pratiksha

Amazon Elastic Container Service > Create new task definition

Clusters Updated Namespaces Task definitions Account settings Updated

Install AWS Copilot

Amazon ECR Repositories

AWS Batch

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 19:50 01-01-2025

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

Task size Info
Specify the amount of CPU and memory to reserve for your task.

CPU **Memory**

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.

--> Inside Container-1 provide Container name

Amazon Elastic Container Service

Clusters Updated

Namespaces

Task definitions

Account settings Updated

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

CloudShell Feedback

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

ENG IN 19:54 01-01-2025

--> Search AWS ECR public Gallery

AWS ECR Public gallery

All Images Videos Shopping News Web Maps More Tools

Sponsored

Amazon Web Services https://aws.amazon.com › account › sign-up

AWS - Amazon Web Services - Get Started For Free Today ✓

Build, Deploy, and Manage Websites, Apps or Processes On AWS Secure, Reliable Network.

ECR Public Gallery https://gallery.ecr.aws

Amazon ECR Public Gallery ✓

Amazon ECR Public Gallery is a website that allows anyone to browse and search for public container images, view developer-provided details, ...

Nginx/nginx ✓

Official NGINX docker image. NGINX is an HTTP and reverse ...

Docker ✓

Amazon ECR Public Gallery is a website that allows anyone to ...

CloudShell Feedback

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

ENG IN 19:51 01-01-2025

--> Select NGiNX--> select the first link--> COpy the URI--> and paste it in Image URI

The screenshot shows the AWS ECR Public Gallery interface. The URL in the address bar is `gallery.ecr.aws/nginx/nginx`. The page displays the **nginx/nginx** image by NGINX, Inc., which has over 1.4B+ downloads. A green button labeled **Launch with App Runner** is visible. Below it, a message says **public.ecr.aws/nginx/nginx:stable-perl copied**. The page also includes sections for **About**, **Usage**, and **Image tags**.

What is nginx?

NGINX (pronounced "engine-x") is an open source reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols, as well as a load balancer, HTTP cache, and a web server (origin server). The nginx project started with a strong focus on high concurrency, high performance and low memory usage. It is licensed under the 2-clause BSD-like license and it runs on Linux, BSD variants, Mac OS X, Solaris, AIX, HP-UX, as well as on other *nix flavors. It also has a proof of concept port for Microsoft Windows.

wikipedia.org/wiki/Nginx

19:52 ENG IN 01-01-2025

--> Under the logging untick use log collection

The screenshot shows the Amazon Elastic Container Service (ECS) task definition creation page. The URL is `ap-south-1.console.aws.amazon.com/ecs/v2/create-task-definition?region=ap-south-1`. The left sidebar shows the navigation menu for the ECS service, with **Task definitions** selected. The main content area is titled **Create new task definition**. It includes a section for **Add environment file** and a **Logging - optional** section. In the **Logging - optional** section, there is a note about **CPU and memory allocation for a sidecar** and a recommendation to use **log collection**. There is also a checkbox for **Use log collection** which is currently unchecked. Other optional sections include **Restart policy - optional** and **HealthCheck - optional**.

--> Create.

Create task definition | Elastic Container Service

Amazon ECR Public Gallery - N

ap-south-1.console.aws.amazon.com/ecs/v2/create-task-definition?region=ap-south-1

aws | Search [Alt+S]

Mumbai | Pratiksha

Amazon Elastic Container Service > Create new task definition

Clusters Updated

Namespaces

Task definitions

Account settings Updated

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Add volume

Volumes from | Info

Mount data volumes from another container.

Add volume from

▶ Monitoring - optional

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

--> Go inside the Cluster name

Cluster services | Elastic Container Service

Amazon ECR Public Gallery - N

ap-south-1.console.aws.amazon.com/ecs/v2/clusters/mycluster13/services?region=ap-south-1

aws | Search [Alt+S]

Mumbai | Pratiksha

Amazon Elastic Container Service > Clusters > mycluster13 > Services

Clusters Updated

Namespaces

Task definitions

Account settings Updated

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Task definition successfully created

TD13:1 has been successfully created. You can use this task definition to deploy a service or run a task.

View task definition

mycluster13

Last updated January 01, 2025 at 19:55 (UTC+5:30)

ARN arn:aws:ecs:ap-south-1:637423422597:cluster/mycluster13

Status Active

CloudWatch monitoring Default

Registered container instances

Services

Draining

Pending

Active

Running

Encryption

CloudShell Feedback

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

ENG IN 19:55 01-01-2025

-> scroll below and select Create service

The screenshot shows the AWS Elastic Container Service (ECS) console. On the left, there's a sidebar with options like Clusters (Updated), Namespaces, Task definitions, Account settings (Updated), Install AWS Copilot, Amazon ECR, Repositories, and AWS Batch. The main area is titled 'Services' and shows 'Fargate ephemeral storage' under 'mycluster13'. Below this, there are tabs for Services, Tasks, Infrastructure, Metrics, Scheduled tasks, and Tags. The Services tab is active, showing 'Services (0)'. It includes a search bar, filters for launch type and service type, and a table header with columns for Service name, ARN, Status, and Deployments and tasks. A message says 'No services' and 'No services to display.' At the bottom right of this section is a large orange 'Create' button. The status bar at the bottom indicates the URL as 'ap-south-1.console.aws.amazon.com/ecs/v2/clusters/mycluster13/services?region=ap-south-1'.

--> under deployment configuration select our task family

The screenshot shows the 'Create service' page in the AWS Elastic Container Service (ECS) console. The left sidebar is identical to the previous screenshot. The main area is titled 'Deployment configuration'. It has two radio button options: 'Service' (which is unselected) and 'Task' (which is selected). A callout box highlights the 'Task' option. Below this, there's a 'Task definition' section with a note about selecting an existing task definition or creating a new one. There's also a checkbox for 'Specify the revision manually' which is unchecked. Further down are dropdown menus for 'Family' (set to 'TD13') and 'Revision' (set to '1 (LATEST)'). At the bottom is a 'Desired tasks' section with a text input field containing the number '1'. The status bar at the bottom indicates the URL as 'ap-south-1.console.aws.amazon.com/ecs/v2/clusters/mycluster13/create-service?region=ap-south-1'.

--> Provide Service Name--> Under Networking Select Linux Security Group

The screenshot shows the 'Create service' wizard in the AWS Elastic Container Service (ECS) console. The left sidebar lists 'Clusters Updated' and other services like Amazon ECR and AWS Batch. The main pane shows network configuration:

- Subnets:** Three subnets are selected: subnet-0c8dde09bdd7c8eac (ap-south-1c, 172.31.16.0/20), subnet-0bed7e95e010d2720 (ap-south-1b, 172.31.0.0/20), and subnet-018030706c398b32a (ap-south-1a, 172.31.32.0/20).
- Security group:** A dropdown menu shows 'Choose security groups' with one item: sg-070b5824b9aedc64b (LinuxSecurityGroup).
- Public IP:** A note says 'Choose whether to auto-assign a public IP to the task's elastic network interface (ENI)'.

At the bottom, the status bar shows CloudShell, Feedback, and various system icons.

--> Expand Load balancer--> select Application Load balancer--> Provide Load balancer Name

The screenshot shows the 'Create service' wizard in the AWS Elastic Container Service (ECS) console, focusing on load balancing:

- Load balancing - optional:** A note says 'Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.'
- Load balancer type:** Set to 'Application Load Balancer'.
- Container:** Set to 'Cont-1 80:80'.
- Application Load Balancer:** A dropdown menu shows 'Create a new load balancer' selected.
- Load balancer name:** A field where the user can assign a unique name for the load balancer.

At the bottom, the status bar shows CloudShell, Feedback, and various system icons.

--> Provide Target Group name

Amazon Elastic Container Service

Clusters Updated

Namespaces

Task definitions

Account settings Updated

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Create service | Elastic Container Service | Amazon ECR Public Gallery - N | public.ecr.aws/nginx/nginx:stal | +

ap-south-1.console.aws.amazon.com/ecs/v2/clusters/mycluster13/create-service?region=ap-south-1

Search [Alt+S]

Mumbai | Pratiksha

Manually input the revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family TD13 Revision 1 (LATEST)

Service name myService13

Service type Replica

Desired tasks 1

Availability Zone rebalancing | Info

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 20:10 01-01-2025

--> and create

Amazon Elastic Container Service

Clusters Updated

Namespaces

Task definitions

Account settings Updated

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Create service | Elastic Container Service | Amazon ECR Public Gallery - N | public.ecr.aws/nginx/nginx:stal | +

ap-south-1.console.aws.amazon.com/ecs/v2/clusters/mycluster13/create-service?region=ap-south-1

Search [Alt+S]

Mumbai | Pratiksha

VPC Lattice - optional, new 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.

Cancel Create

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 20:11 01-01-2025

--> After Service is Created go in load balancer Name Inside EC2 Service

The screenshot shows the AWS EC2 Load balancers page. On the left, there's a sidebar with navigation links for Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), and Images (with sub-links for AMIs and AMI Catalog). The main content area is titled "Load balancers (1)". It displays a table with one row for "LB13". The columns in the table are Name, DNS name, State, VPC ID, and Availability Zones. The "DNS name" column shows "LB13-1610048338.ap-sout...". The "State" column shows "Active". The "VPC ID" column shows "vpc-09867797ad23c076c". The "Availability Zones" column shows "3 Availability Zones". Below the table, a message says "0 load balancers selected" and "Select a load balancer above." At the top right, there are "Actions" and "Create load balancer" buttons.

--> Copy the DNS name and paste it in New Tab

The screenshot shows the AWS Load balancer details page for "LB13". The left sidebar is identical to the previous screenshot. The main content area shows the "Listeners and rules" tab selected. It displays the "Load balancer ARN" as "arn:aws:elasticloadbalancing:ap-south-1:637423422597:loadbalancer/app/LB13/83abf19f94577e2a" and the "DNS name" as "LB13-1610048338.ap-south-1.elb.amazonaws.com (A Record)". A green callout bubble indicates that the DNS name has been copied. At the bottom, there are tabs for "Listeners and rules (1)", "Manage rules", "Manage listener", and "Add listener". The status bar at the bottom shows "CloudShell Feedback Search ENG IN 2015 01-01-2025".

--> Here u will see the Welcome Page of nginx

