



**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

Coaching - ECS

Cloud Infrastructure Engineering

**Nanyang Technological University
& Skills Union - 2022/2023**

ECS Overview

Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service that simplifies your deployment, management, and scaling of containerized applications.

Use Cases

Modernize applications

Empower developers to build and deploy applications with enhanced security features in a fast, standardized, compliant, and cost-efficient manner with Amazon ECS.

Automatically scale web applications

Automatically scale and run web applications in multiple Availability Zones with the performance, scale, reliability, and availability of AWS.

Support batch processing

Plan, schedule, and run batch computing workloads across AWS services, including Amazon Elastic Compute Cloud (EC2), AWS Fargate, and Amazon EC2 Spot Instances.

Train NLP and AI/ML models

Train natural language processing (NLP) and other artificial intelligence (AI) / machine learning (ML) models without managing the infrastructure by using Amazon ECS with AWS Fargate.

Amazon Elastic Container Service Layers

Provisioning

Amazon Web Services
Command Line Interface

Copilot

Management console

Amazon Web Services
Cloud Developer Kit

Amazon Web Services
Software Developer Kit



Amazon ECS scheduler

Controller



Amazon EC2 instances

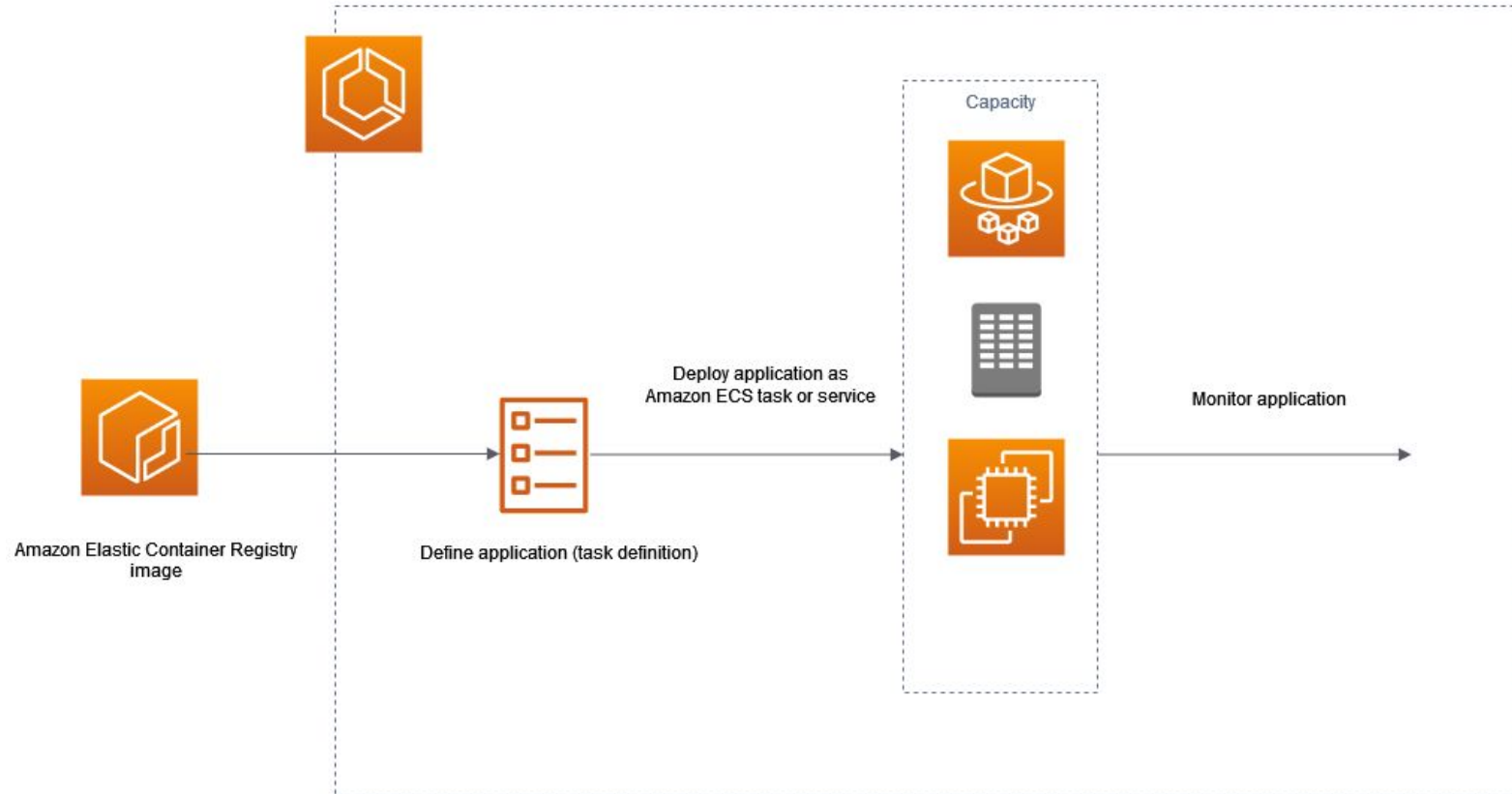
Amazon Web Services Fargate

On-premises compute

Capacity options



Amazon ECS Application Lifecycle



ECS Terminologies



ECS Cluster

An Amazon ECS cluster is a logical grouping of tasks or services. In addition to tasks and services, a cluster consists of the following resources:

- The infrastructure capacity which can be any of the following:
 - Amazon EC2 instances in the AWS cloud
 - Serverless (AWS Fargate (Fargate)) in the AWS cloud
 - On-premises virtual machines (VM) or servers

ECS Capacity Provider

Fargate: With Amazon ECS on AWS Fargate capacity providers, you can use both Fargate and Fargate Spot capacity with your Amazon ECS tasks. With Fargate Spot, you can run interruption tolerant Amazon ECS tasks at a rate that's discounted compared to the Fargate price. Fargate Spot runs tasks on spare compute capacity. When AWS needs the capacity back, your tasks are interrupted with a two-minute warning.

EC2: With AWS ECS on EC2 capacity providers, you can specify the auto scaling group and the setting required for the autoscaling group.

AWS ECS Task Definition

A *task definition* is a blueprint for your application. It is a text file in JSON format that describes the parameters and one or more containers that form your application.

The following are some of the parameters that you can specify in a task definition:

- The Docker image to use with each container in your task
- How much CPU and memory to use with each task or each container within a task
- The launch type to use, which determines the infrastructure that your tasks are hosted on
- The Docker networking mode to use for the containers in your task
- The logging configuration to use for your tasks
- Whether the task continues to run if the container finishes or fails
- The command that the container runs when it's started
- Any data volumes that are used with the containers in the task
- The IAM role that your tasks use

ECS Service vs Tasks

Service: Typically used for long-running tasks / containers that can be stopped and restarted.
E.g. A web application

Task: A standalone task that is used to run a short running task. For e.g. A batch job, or a scheduled cron job to run at a specific time everyday and then exit

Clusters > ecs-tf > new

Create scheduled task

Run Amazon ECS tasks on a cron-like schedule using CloudWatch Events rules and targets.

Schedule rule name*

Schedule rule enabled* ☒

Schedule rule description

Schedule rule type ☒ Run at fixed interval ☐ Cron expression

Run at fixed interval*

Schedule targets

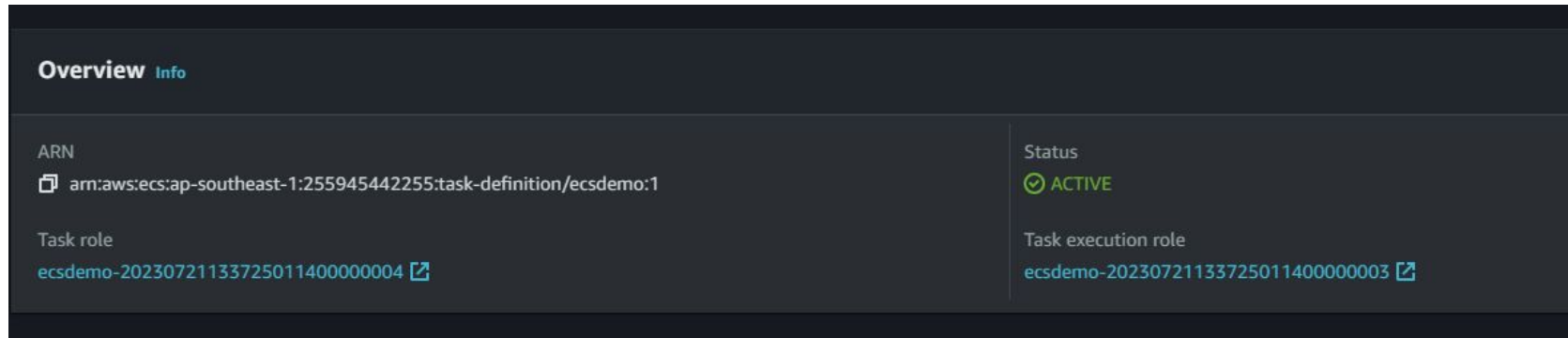
Create one or more task definition targets that run Amazon ECS tasks when the schedule rule is triggered.

Target name
▼ (NEW)
<div>Target id* <input type="text"/></div> <div>Launch type <input type="radio"/> FARGATE <input checked="" type="radio"/> EC2</div> <div>Task Definition <div>Family <input type="text" value="ecsdemo"/> <input type="button" value="Enter a value"/></div><div>Revision <input type="text" value="1 (latest)"/></div></div> <div>Cluster* <input type="text" value="ecs-tf"/></div> <div>Number of tasks* <input type="text" value="1"/></div>





ECS TaskExecutionRole vs TaskRole

TaskExecutionRole: IAM role that executes ECS actions such as pulling the image and storing the application logs in cloudwatch.

TaskRole: IAM role used by the task itself. For example, if your container wants to call other AWS services like S3, SQS, etc then those permissions would need to be covered by the TaskRole



The screenshot displays the AWS IAM console interface for a specific TaskExecutionRole. The 'Overview' tab is selected, showing the role's ARN and its status as 'ACTIVE'. The 'Task role' and 'Task execution role' are both listed as 'ecsdemo-20230721133725011400000004'.

Overview Info	
ARN	Status
 arn:aws:ecs:ap-southeast-1:255945442255:task-definition/ecsdemo:1	 ACTIVE
Task role	Task execution role
ecsdemo-20230721133725011400000004 	ecsdemo-20230721133725011400000003 

Creating a ECS Cluster

Amazon Elastic Container Service > Clusters

Clusters (0) [Info](#)

Create cluster

< 1 >

Cluster	Services	Tasks	Registered container instances	CloudWatch monitoring	Capacity provider strategy
No clusters No clusters to display					

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

jazeel-cluster

There can be a maximum of 255 characters. The valid characters are letters (uppercase and lowercase), numbers, hyphens, and underscores.

▼ Networking [Info](#)

By default tasks and services run in the default subnets for your default VPC. To use the non-default VPC, specify the VPC and subnets.

VPC

Use a VPC with public and private subnets. By default, VPCs are created for your AWS account. To create a new VPC, go to the [VPC Console](#).

vpc-01eb8331fac862b2e
sandbox-vpc

Subnets

Select the subnets where your tasks run. We recommend that you use three subnets for production.

Choose subnets

subnet-0623d78431b777e3e public ✕
public-subnet-01
ap-southeast-1a 10.0.1.0/24

subnet-02a6bf9a87a5dec14 public ✕
public-subnet-02
ap-southeast-1b 10.0.2.0/24

subnet-07d728c6db3bd830b public ✕
public-subnet-03
ap-southeast-1c 10.0.3.0/24

Default namespace - optional

Select the namespace to specify a group of services that make up your application. You can overwrite this value at the service level.

Task Definition Creation

Amazon Elastic Container Service > Task definitions

Task definitions (0) [Info](#)



Deploy ▼

Create new revision ▼

Create new task definition ▼

🔍 Filter task definitions by property or value

< 1 > ⚙️

Task definition ▼

Status of last revision ▼

No task definitions

No task definitions to display.

Create new task definition

Configure task definition and containers

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.

Container - 1 [Info](#)

[Essential container](#)[Remove](#)

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**

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. Any changes to port mappings configuration impacts the associated service connect settings.

Container port**Protocol****Port name****App protocol**[Remove](#)[Add more port mappings](#)[▼ Environment variables - optional \[Info\]\(#\)](#)

Task Definition Example

JSON

task_definition.json

```
1 {  
2   "taskDefinitionArn": "arn:aws:ecs:ap-southeast-1:255945442255:task-definition/ecsdemo:1",  
3   "containerDefinitions": [  
4     {  
5       "name": "ecs-sample",  
6       "image": "public.ecr.aws/docker/library/httpd:latest",  
7       "cpu": 0,  
8       "portMappings": [  
9         {  
10          "name": "ecs-sample",  
11          "containerPort": 80,  
12          "hostPort": 80,  
13          "protocol": "tcp"  
14        }  
15      ],  
16      "essential": true,  
17      "environment": [],  
18      "mountPoints": [],  
19      "volumesFrom": [],  
20      "startTimeout": 30,  
21      "stopTimeout": 120,  
22      "privileged": false,  
23      "readonlyRootFilesystem": false,  
24      "interactive": false,  
25      "pseudoTerminal": false,  
26      "logConfiguration": {  
27        "logDriver": "awslogs",  
28        "options": {  
29          "awslogs-group": "/aws/ecs/ecsdemo/ecs-sample",  
30          "awslogs-region": "ap-southeast-1",
```


Service Creation

Amazon Elastic Container Service > Clusters > jazeel-cluster > Services

jazeel-cluster ↻ Update cluster Delete cluster

Cluster overview

ARN jazeel-cluster	Status Active	CloudWatch monitoring Default	Registered container instances -
Services Draining -	Active -	Tasks Pending -	Running -

[Services](#) | [Tasks](#) | [Infrastructure](#) | [Metrics](#) | [Scheduled tasks](#) | [Tags](#)

Services (0) [Info](#)

↻ Manage tags Update Delete service Create

All launch types All service types < 1 > ⚙️

Service name	Status	ARN	Service type
No services No services to display. Create			

Deployment configuration

Application type [Info](#)

Specify what type of application you want to run.

☒ **Service**

Launch a group of tasks handling a long-running computing work that can be stopped and restarted. For example, a web application.

☐ **Task**

Launch a standalone task that runs and terminates. For example, a batch job.

Task definition

Select an existing task definition. To create a new task definition, go to [Task definitions](#).

☐ **Specify the revision manually**

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

Family

httpd

Revision

1 (LATEST)

Service name

Assign a unique name for this service.

httpd

Service type [Info](#)

Specify the service type that the service scheduler will follow.

☒ **Replica**

Place and maintain a desired number of tasks across your cluster.

☐ **Daemon**

Place and maintain one copy of your task on each container instance.

Desired tasks

Specify the number of tasks to launch.

1

► Deployment options



▼ Networking

VPC [Info](#)

Choose the Virtual Private Cloud to use.

vpc-01eb8331fac862b2e
sandbox-vpc

Subnets

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

Choose subnets

subnet-0623d78431b777e3e public ✕
public-subnet-01
ap-southeast-1a 10.0.1.0/24

subnet-02a6bf9a87a5dec14 public ✕
public-subnet-02
ap-southeast-1b 10.0.2.0/24

subnet-07d728c6db3bd830b public ✕
public-subnet-03
ap-southeast-1c 10.0.3.0/24

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.

Choose security groups

sg-01adb0fa94b766534 ✕
ssh-only

Public IP [Info](#)

Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).

☒ Turned on



Getting IP of Task

Amazon Elastic Container Service > Clusters > jazeel-cluster > Tasks

jazeel-cluster

Cluster overview

ARN: jazeel-cluster Status: Active

Services: Draining Active

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

Tasks (1)

Filter tasks by property or value Running tasks All launch types

<input type="checkbox"/>	Task	Last status	Desired st...	Task ...	Revision	Health sta...	Sta
<input type="checkbox"/>	9e77e5bbbe26472e8a...	Provisioning	Running	httpd	1	Unknown	-

Getting IP of Task




Amazon Elastic Container Service > Clusters > jazeel-cluster > Tasks > 9e77e5bbbe26472e8ad8af7025c04193 > Configuration

9e77e5bbbe26472e8ad8af7025c04193




[Refresh](#) [Stop](#)

[Configuration](#) [Logs](#) [Networking](#) [Tags](#)

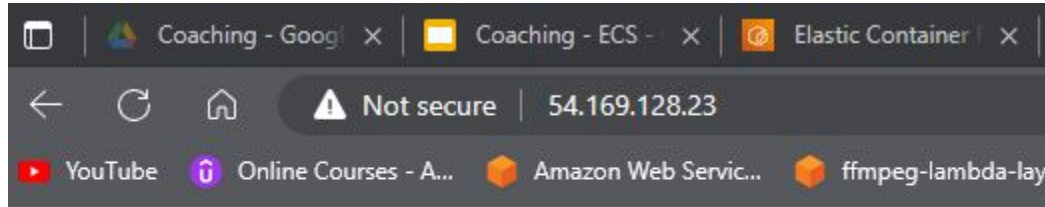
Task overview

ARN  9e77e5bbbe26472e8ad8af7025c04193	Last status  Running	Desired status  Running	Started/Created at 20/07/2023, 15:40:27 UTC 20/07/2023, 15:40:06 UTC
---	---	--	--

Configuration

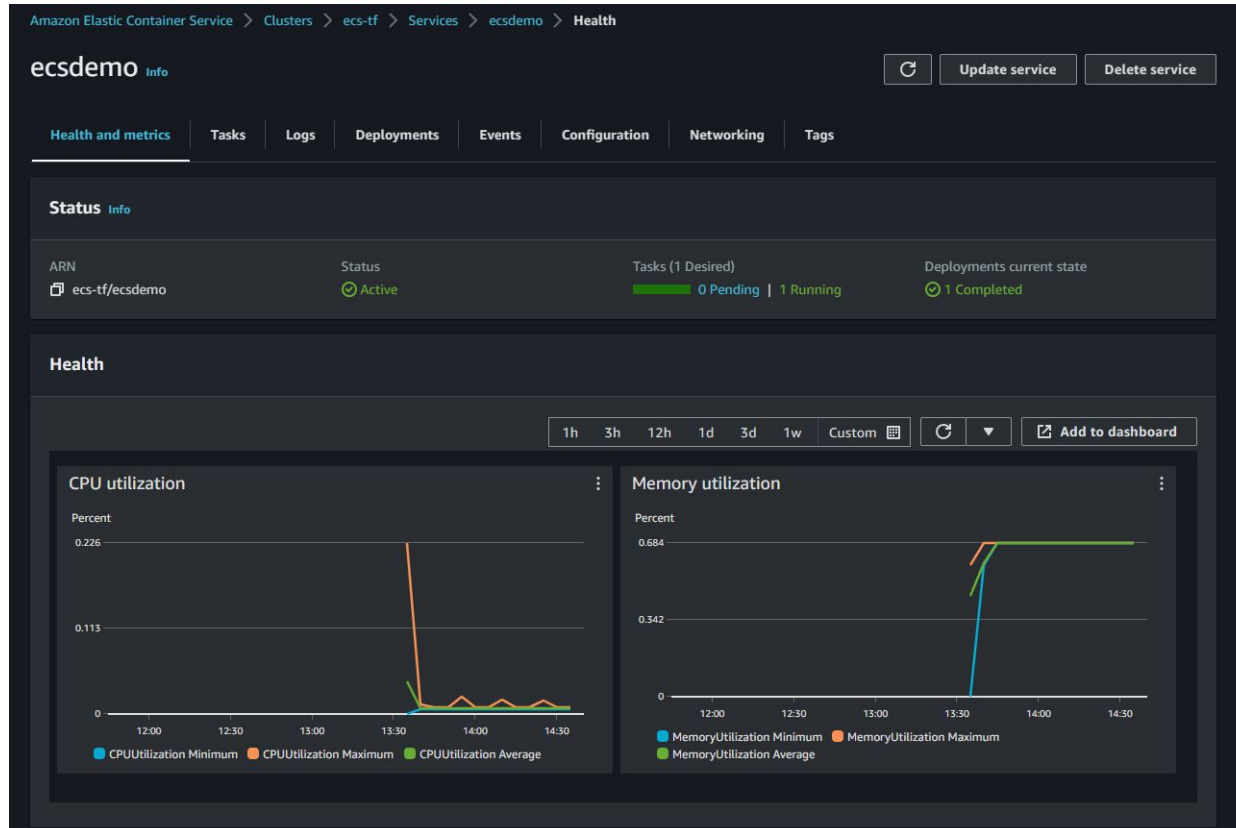
Operating system/Architecture Linux/X86_64	Capacity provider FARGATE	ENI ID eni-0b7b2524866b3d4e3	Public IP  54.169.128.23 open address
CPU Memory 1 vCPU 3 GB	Launch type FARGATE	Network mode awsvpc	Private IP  10.0.2.37
Platform version 1.4.0	Task definition httpd:1	Subnet ID subnet-02a6bf9a87a5dec14	MAC address  02:98:65:df:2d:a0
	Task group service:httpd		

Getting IP of Task



It works!

Service In-depth - Health and metrics



Service In-depth - Logs

Amazon Elastic Container Service > Clusters > ecs-tf > Services > ecsdemo > Logs

ecsdemo Info

Update service Delete service

Health and metrics Tasks **Logs** Deployments Events Configuration Networking Tags

Logs (5+) Info

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Since 1 hour ago View in CloudWatch

Search log events with filter patterns ecs-sample

Timestamp (Local)	Message	Task	Container
7/21/2023, 10:29:15 PM GMT+8	205.210.31.211 - - [21/Jul/2023:14:29:15 +0000] "GET / HTTP/1.1" 200 45	8251e28f6c2d42119 4c405a874bd260b	ecs-sample
7/21/2023, 10:14:25 PM GMT+8	162.216.149.227 - - [21/Jul/2023:14:14:25 +0000] "GET / HTTP/1.1" 200 45	8251e28f6c2d42119 4c405a874bd260b	ecs-sample
7/21/2023, 10:01:59 PM GMT+8	162.243.138.5 - - [21/Jul/2023:14:01:59 +0000] "GET / HTTP/1.1" 200 45	8251e28f6c2d42119 4c405a874bd260b	ecs-sample
7/21/2023, 9:59:39 PM GMT+8	40.77.167.177 - - [21/Jul/2023:13:59:39 +0000] "GET / HTTP/1.1" 200 45	8251e28f6c2d42119 4c405a874bd260b	ecs-sample
7/21/2023, 9:59:33 PM GMT+8	52.167.144.133 - - [21/Jul/2023:13:59:33 +0000] "GET /robots.txt HTTP/1.1" 404 196	8251e28f6c2d42119 4c405a874bd260b	ecs-sample

Service In-depth - Deployment

ecsdemo Info

↻

Update service

Delete service

Health and metrics

Tasks

Logs

Deployments

Events

Configuration

Networking

Tags

Deployment configuration Info

View pipelines ↗

Deployment status
✔ **Completed**

Deployment type
ECS

Platform version
LATEST

Min and max running tasks
100% min and 200% max

▶ Deployment failure detection

▶ Task placement strategy and constraints

Deployments (1) Info

↻

🔍 Filter deployments

< 1 > ⚙️

Start date	Status	Failed tasks	Tasks	Version	Task definition	Revision	Last d
7/21/2023, 9:37:37 PM GMT+8	✔ Primary <div>100%</div>	0	1 Running 0 Pending 1 Desired	1.4.0	ecsdemo	1	✔ Cor

Events (3)

↻

🔍 Filter events by value

< 1 > ⚙️

Started at	Message	Event ID
7/21/2023, 9:38:12 PM GMT+8	service ecsdemo has reached a steady state.	4d4603d8-3ca2-4864-a517-ac5b180a7871
7/21/2023, 9:38:12 PM GMT+8	service ecsdemo deployment ecs-svc/2388272834080545988 deployment completed.	f3812b19-6614-4c53-85e2-98a07f0783a8
7/21/2023, 9:37:42 PM GMT+8	service ecsdemo has started 1 tasks; task 8251e28f6c2d421194c405a874bd260b.	320ecc9c-5bea-4006-8882-b4bb6e2ce8f6

Service In-depth - Service Auto-Scaling

ecsdemo [Info](#)

[Health and metrics](#) [Tasks](#) [Logs](#) [Deployments](#) [Events](#) [Configuration](#) [Networking](#) [Tags](#)

Service configuration [Info](#)

Service ARN
ecs-tf/ecsdemo

Task definition: revision
ecsdemo:1

► Service discovery

Service Connect [Info](#)

Configure

Launch type
FARGATE

Service Connect namespace
-

Service type
REPLICA

Service Connect services
-

Created by
arn:aws:iam::255945442255:user/jazeel

Auto Scaling

Desired tasks
1

Min tasks
1

Max tasks
10

Policies (2)

< 1 >

Policy name	Policy type	Scale-in	Alarm
memory: Tracking ECSServiceAverageMemoryUtilization at 75	Target tracking	On	TargetTracking-service/ecs-tf/ecsdemo-AlarmHigh-f9486426-9ad9-4565-b6d5-5b415be5dd65
cpu: Tracking ECSServiceAverageCPUUtilization at 75	Target tracking	On	TargetTracking-service/ecs-tf/ecsdemo-AlarmHigh-9519d8a8-c3e9-445b-9687-f2b63a141518

Service In-depth - Networking

Amazon Elastic Container Service > Clusters > ecs-tf > Services > ecsdemo > Networking

ecsdemo [Info](#)

[Refresh](#) [Update service](#) [Delete service](#)

[Health and metrics](#) [Tasks](#) [Logs](#) [Deployments](#) [Events](#) [Configuration](#) [Networking](#) [Tags](#)

Network configuration

Network vpc-01eb8331fac862b2e	Security groups sg-0b82d4bfb26b50646 sg-01adb0fa94b766534	Service role AWSServiceRoleForECS	Load balancers -
Subnets subnet-0623d78431b777e3e subnet-02a6bf9a87a5dec14 subnet-07d728c6db3bd830b	Auto-assign public IP ✔ Turned on		DNS names -
	Health check grace period -		Target groups -

Create ECS resources using Terraform



Terraform ECS Module

[terraform-aws-modules/terraform-aws-ecs](https://github.com/terraform-aws-modules/terraform-aws-ecs): Terraform module which creates AWS ECS resources  (github.com)

Activity

Create a new repo in Github and clone it to your local computer

.gitignore template -> Terraform

Add README.md



Activity

Create a main.tf file with below content:

Change the variables in the lines which has # Change in the comment. (line 18 and 29)

<https://github.com/jaezeu/hello-node/blob/main/terraform/main.tf>



Activity

Create a backend.tf file with below content:

```
terraform {  
  
  backend "s3" {  
  
    bucket = "sctp-ce2-tfstate-bkt"  
  
    key   = "jazeel-ecs-1.tfstate" #Change the value of this to yourname-ecs-1.tfstate for example  
  
    region = "ap-southeast-1"  
  
  }  
  
}
```


Activity

Create a provider.tf file with below content:

```
provider "aws" {  
    region = "ap-southeast-1"  
}
```

Activity

Once the files above have been created, Run the following commands:

terraform init

terraform plan

terraform apply

Activity

- 1) Access the cluster you have created
- 2) Go to the tasks(Slide 13)
- 3) Get public IP and display in browser(Slide 14)

CI/CD Workflow

