

Demo 2: ECS Basics - A region-based service

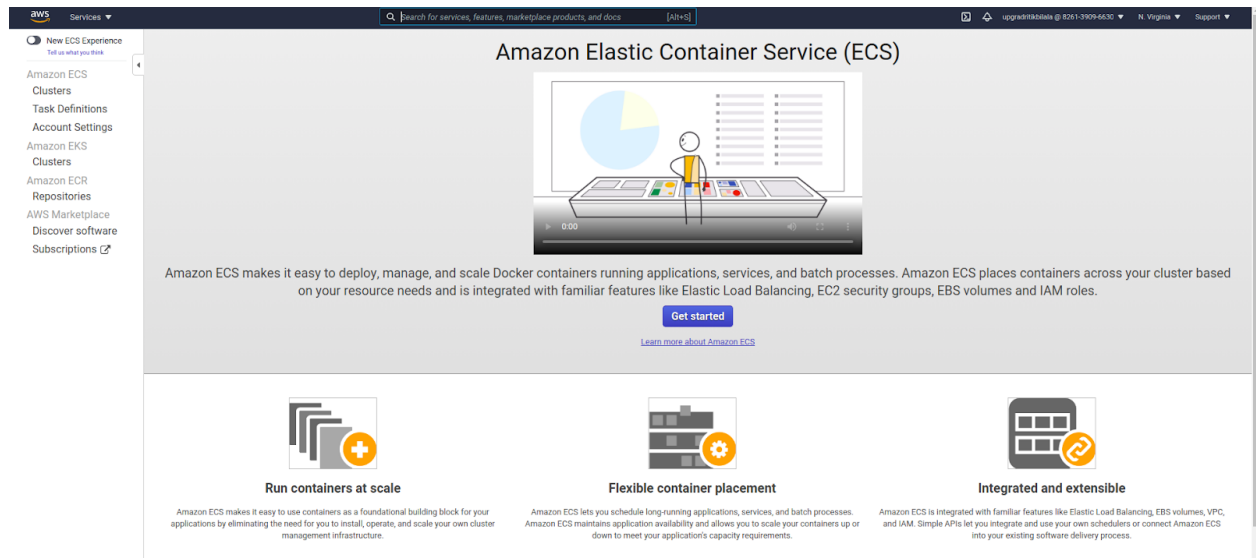
Introduction

Following are the learning objectives of this demonstration:

- EC2 and Fargate mode overview
- Launch 'get started' cluster, a walkthrough by AWS
- ECS task and service overview


ECS Basics:

1. Browse to ECS console



2. Overview of ECS cluster modes:

From the left panel, browse to the clusters page; Clusters >> Create Clusters.

 Services ▼

Search for services, features, marketplace products, and docs

Create Cluster

Step 1: Select cluster template
Step 2: Configure cluster

Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only
Resources to be created:
Cluster
VPC (optional)
Subnets (optional)

Powered by AWS Fargate

EC2 Linux + Networking
Resources to be created:
Cluster
VPC
Subnets
Auto Scaling group with Linux AMI

EC2 Windows + Networking
Resources to be created:
Cluster
VPC
Subnets
Auto Scaling group with Windows AMI

*Required

Cancel

Next step

There are three modes :

1. Fargate
2. EC2- Linux
3. EC2 - Windows

Before creating our cluster, let's look at what amazon already has for us.

Launch a 'getting started' cluster

This is a tutorial by amazon to walk us through the ECS features.

1. Click on the " Get Started" button on the "clusters" page. You will see the following page:

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

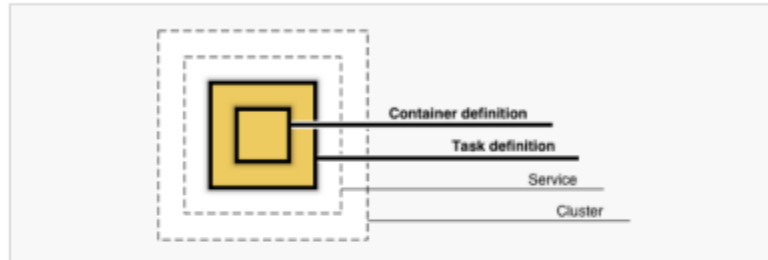
Step 1: Container and Task

Step 2: Service

Step 3: Cluster

Step 4: Review

Diagram of ECS objects and how they relate



Container definition

Edit

Choose an image for your container below to get started quickly or define the container image to use.

sample-app

Image : httpd:2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

nginx

Image : nginx:latest
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

tomcat-webserver

Image : tomcat
memory : 2GB (2048)
cpu : 1 vCPU (1024)

custom

Image : --
memory : --
cpu : --

Configure

Task definition

Edit

A task definition is a blueprint for your application, and describes one or more containers through attributes. Some attributes are configured at the task level but the majority of attributes are configured per container.

2. Choose an **Nginx** image for your container.
3. Click on edit under the task definition.

Configure task definition: first-run-task-definition

Task definition details

Task definition name*

first-run-task-definition

Network mode*

awsvpc

Task execution role

You are giving permission to Elastic Container Service to create and use ecsTaskExecutionRole.

Compatibilities*

FARGATE

[Learn more](#) about compatibilities

Task size

Task size allows you to size at the task level and optionally set container-specific CPU and memory sizes. You are billed for the task memory and task CPU allocated.

Task memory*

0.5GB (512)

Task CPU*

0.25 vCPU (256)

*Required

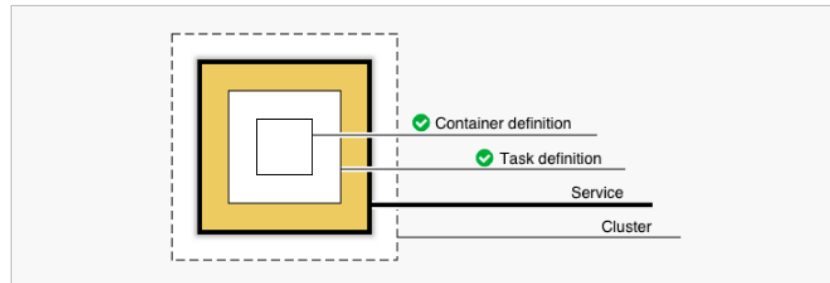
Cancel Save

- On the next page, define your service.

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

[Step 1: Container and Task](#)**Step 2: Service**[Step 3: Cluster](#)[Step 4: Review](#)

Diagram of ECS objects and how they relate



Define your service

[Edit](#)

A service allows you to run and maintain a specified number (the "desired count") of simultaneous instances of a task definition in an ECS cluster.

Service name **sample-app-service**Number of desired tasks **1**Security group **Automatically create new**

A security group is created to allow all public traffic to your service only on the container port specified. You can further configure security groups and network access outside of this wizard.

Load balancer type ☒ None☐ Application Load Balancer

*Required

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

Set up service: sample-app-service

Service name*

sample-app-service

Number of desired tasks*

1

Network access

If you do not use a load balancer, a security group is created to allow all public traffic to your service ONLY on the container port specified. If you use an Application Load Balancer, two security groups are created to secure your service: An Application Load Balancer security group that allows all traffic on the Application Load Balancer port and an Amazon ECS security group that allows all traffic ONLY from the Application Load Balancer security group. You can further configure security groups and network access outside of this wizard.

Security group*

Automatically create new

CIDR block

0.0.0.0/0

Changing this value affects which IP addresses can access your service.

Port range

80

Elastic Load Balancing (optional)

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service.

Load balancer type*

☐ None

☒ Application Load Balancer

Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple strings can use the same listener port on a single load balancer with rule-based routing and paths.

Container to load balance

sample-app : 80

Load balancer listener port*

80

Load balancer listener protocol*

HTTP

Outside the first-run wizard, you can select a certificate to use HTTPS.

*Required

Cancel

Save

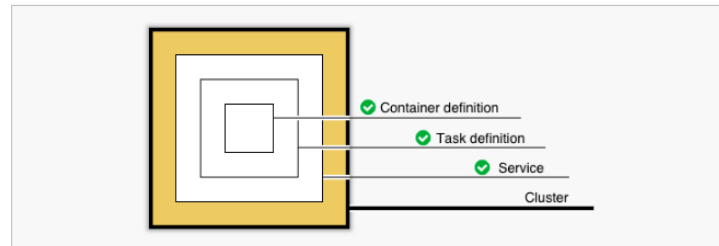
5. Edit the cluster name and click next

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

[Step 1: Container and Task](#)[Step 2: Service](#)**Step 3: Cluster**[Step 4: Review](#)

Diagram of ECS objects and how they relate



Configure your cluster

The infrastructure in a Fargate cluster is fully managed by AWS. Your containers run without you managing and configuring individual Amazon EC2 instances.

To see key differences between Fargate and standard ECS clusters, see the [Amazon ECS documentation](#).

Cluster name

Cluster names are unique per account per region. Up to 255 letters (uppercase and lowercase), numbers, and hyphens are allowed.

VPC ID **Automatically create new** ⓘ

Subnets **Automatically create new** ⓘ

*Required

[Cancel](#)

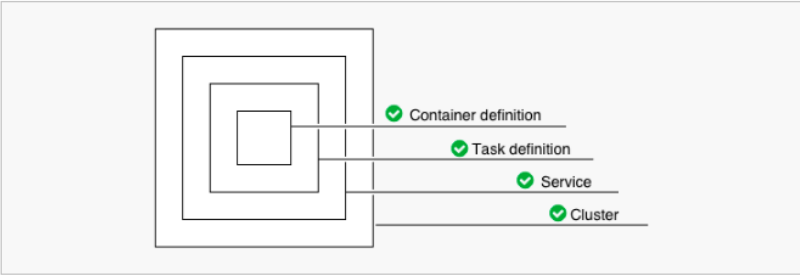
[Previous](#)

[Next](#)

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

- Step 1: Container and Task
- Step 2: Service
- Step 3: Cluster
- Step 4: Review**

Diagram of ECS objects and how they relate



Review

Review the configuration you've set up before creating your task definition, service, and cluster.

Task definition

[Edit](#)

Task definition name	first-run-task-definition
Network mode	awsvpc
Task execution role	Create new
Container name	nginx
Image	nginx:latest
Memory	512
Port	80
Protocol	HTTP

Service

[Edit](#)

6. Click next and create the cluster.

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

Launch Status

We are creating resources for your service. This may take up to 10 minutes. When we're complete, you can view your service.

[Back](#) [View service](#) Enabled after service creation completes successfully

Additional features that you can add to your service after creation

Scale based on metrics

You can configure scaling rules based on CloudWatch metrics

Preparing service : 2 of 10 complete

ECS resource creation	pending
Cluster upgrad	complete
Task definition first-run-task-definition1	complete
Service	pending
Additional AWS service integrations	pending
Log group /ecs/first-run-task-definition	complete
CloudFormation stack	pending
VPC	pending
Subnet 1	pending
Subnet 2	pending
Security group	pending
Load balancer	pending

7. Once the cluster is launched, click on the “view services” button.

Amazon ECS Clusters

Cluster : upgrad

Cluster ARN: arn:aws:ecs:us-east-1:826139096630:cluster/upgrad

Status: ACTIVE

Registered container instances: 0

Pending tasks count: 0 Fargate, 0 EC2

Running tasks count: 1 Fargate, 0 EC2

Active service count: 1 Fargate, 0 EC2

Draining service count: 0 Fargate, 0 EC2

Services | Tasks | ECS Instances | Metrics | Scheduled Tasks | Tags | Capacity Providers

Create | Update | Delete | Actions

Last updated on March 19, 2021 8:38:45 PM (0m ago)

Service Name	Status	Service type	Task Definition	Desired tasks	Running tasks	Launch type	Platform version
nginx-service	ACTIVE	REPLICA	first-run-task-definition:1	1	1	FARGATE	LATEST(1.4.0)

- The above page shows every detail of the cluster
- Click on the task tab

Task status: **Running** Stopped

Filter in this page

< 1-3 > Page size 50

Task	Task Definition ...	Last status	Desired status ...	Group	Launch type	Platform versio...
004ba4fded9f404...	first-run-task-defi...	RUNNING	RUNNING	service:nginx-ser...	FARGATE	1.3.0
2413b40957404c...	first-run-task-defi...	PROVISIONING	RUNNING	service:nginx-ser...	FARGATE	1.3.0
cb5f56925c804b...	first-run-task-defi...	PROVISIONING	RUNNING	service:nginx-ser...	FARGATE	1.3.0

- Hit the public IP of any tasks

Not secure | 52.90.199.69

Apps | Gmail | YouTube | POC | AWS Console | TIMESHEET | JIRA | H1 | Linux Academy | MS | Program Cale... | EKS-Setup-up...

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

- Try terminating the tasks, and service will respawn the tasks to maintain availability.

Note:

- Carefully look at all the pages and options of the “getting started” tutorial, followed in the demonstration above.
- Don't forget to **delete** the cluster after you have gone through all the details.

- logging, monitoring

Additional features that you can add to your service after creation

Scale based on metrics

You can configure scaling rules based on CloudWatch metrics

Preparing service : 7 of 9 complete

ECS resource creation		pending
Cluster	test-cluster	complete
Task definition	first-run-task-definition:1	complete
Service		pending
Additional AWS service integrations		pending
Log group	/ecs/first-run-task-definition	complete
CloudFormation stack		pending
VPC	vpc-04cd21518d1b418d8	complete
Subnet 1	subnet-0036bdcac13f1fcad	complete
Subnet 2	subnet-0bd2690a77fed084a	complete
Security group	sg-00f70c60e796ccea6	complete

