

LAB: Scale a Fargate Service with an Application Load Balancer

You need:

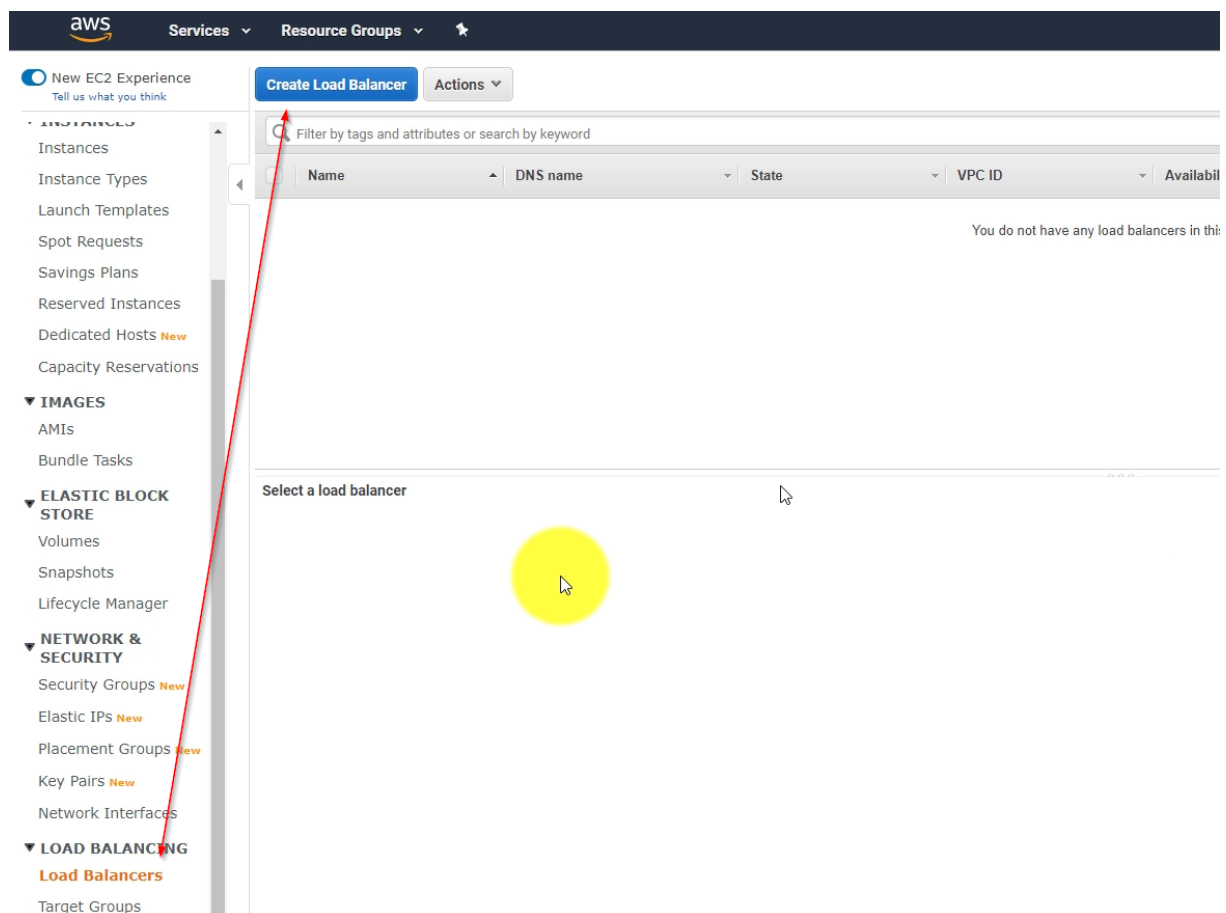
- An AWS Account

Duration of the Lab: 30 Minutes.

Difficulty: medium

Create a new Application Load Balancer

Open the EC2 Dashboard and scroll down to Load Balancers:



1. Create an Application Load Balancer
2. Name: choose any name you like
3. Subnet placement: Place it in all three subnets of the default VPC
4. Security Group: Create a new security group and allow traffic on port 80 to flow in

For the Target group select IP as a Target Type, because Fargate creates an elastic Network Interface for each Task, which gets its own IP:

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and pe

Target group

Target group	<input type="text" value="New target group"/>
Name	<input type="text" value="targetgroup-fargate-cluster"/>
Target type	<p><input type="radio"/> Instance</p> <p><input checked="" type="radio"/> IP</p> <p><input type="radio"/> Lambda function</p>
Protocol	<input type="text" value="HTTP"/>
Port	<input type="text" value="80"/>

Health checks

Protocol	<input type="text" value="HTTP"/>
Path	<input type="text" value="/"/>

► [Advanced health check settings](#)

Then there are no targets to register and create the Application Load Balancer!

Create a Service

Head over to the Amazon ECS Dashboard and open (or create and then open) a Fargate Cluster and create a new Service:

Complete AWS ECS DevOps Masterclass for Beginners

The screenshot shows the AWS Management Console interface for the 'Cluster : myfargate'. The left sidebar contains navigation links for Amazon ECS, Clusters, Task Definitions, Account Settings, Amazon EKS, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main content area displays the cluster details, including the Cluster ARN, Status (ACTIVE), and various task counts. A yellow circle highlights the 'Active service count' and 'Draining service count' metrics, both showing 0. Below the metrics, there are tabs for Services, Tasks, ECS Instances, Metrics, Scheduled Tasks, Tags, and Capacity Providers. The 'Services' tab is selected, showing a table with columns for Service Name, Status, Service type, and Task Definition. The table is currently empty, displaying 'No results'.

Cluster : myfargate

Get a detailed view of the resources on your cluster.

Cluster ARN: arn:aws:ecs:eu-central-1:161952721022:cluster/myfargate

Status: ACTIVE

Registered container instances: 0

Pending tasks count: 0 Fargate, 0 EC2

Running tasks count: 0 Fargate, 0 EC2

Active service count: 0 Fargate, 0 EC2

Draining service count: 0 Fargate, 0 EC2

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

Create | Update | Delete | Actions


Filter in this page | Launch type: ALL | Service type: ALL

	Service Name	Status	Service type	Task Definition
No results				


Let's select the TaskDefinition for our Apache Container we created in previous labs, but this time we connect the services using a loadbalancer:

Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.


Launch type ☒ FARGATE ☐ EC2 

Task Definition Family


simple-apache-server 


Enter a value

Revision


1 (latest) 


Platform version

LATEST 




Cluster


myfargate 




Service name

apache-service 


Service type*

REPLICA 


Number of tasks

3 

Minimum healthy percent

100 

Maximum percent

200 

Fill out the form accordingly and as number of Tasks enter “3”. Then go to the next step.

Select your default VPC and select to place the task in all three availability zones (subnets).

Change the Security group, so that HTTP Traffic is only allowed from the Security Group of your Load Balancer:

Complete AWS ECS DevOps Masterclass for Beginners

Subnets*

Security groups*

Auto-assign public IP

Health check grace period

Configure security groups

A security group is a set of firewall rules that control the traffic for your task. On this page, you can add rules to allow specific traffic to reach your task, or you can choose to use an existing security group. [Learn more.](#)

Assigned security groups

☒ Create new security group

☐ Select existing security group

Security group name* apache-3301

Description Tue Apr 07 2020 11:19:33 GMT+0200 (Central European S...

Inbound rules for security group

Type	Protocol	Port range	Source
HTTP	TCP	80	Source group sg-063e83d1c9a5

[Add rule](#)

Select an application load balancer:

Load balancing

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service. Choose an existing load balancer, or create a new one in the [Amazon EC2 console](#).

Load balancer type*

☐ None

Your service will not use a load balancer.



☒ Application Load Balancer

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

☐ Network Load Balancer

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it selects a target from the target group for the default rule using a flow hash routing algorithm.

☐ Classic Load Balancer

Requires static host port mappings (only one task allowed per container instance); rule-based routing and paths are not supported.

Service IAM role

Task definitions that use the awsvpc network mode use the AWSServiceRoleForECS service-linked role, which is created for you automatically. [Learn more.](#)

Load balancer name

fargate-lb



Add the apachecontainer to the already existing Target Group:

Complete AWS ECS DevOps Masterclass for Beginners

Container to load balance

apachecontainer : 80

Remove ✕

Production listener port* 80:HTTP ⓘ

Production listener protocol* HTTP

Target group name targetgroup-fargate-cl... ⓘ

Target group protocol HTTP ⓘ

Target type ip ⓘ

Path pattern / Evaluation order default

Health check path / ⓘ

Additional health check options can be configured in the ELB console after you create your service.

For this lab we do not need any service discovery, you can de-activate it.

In the next step enable the Auto Scaling, set the minimum number of tasks to 1 and the maximum to 10, the desired count to 3:

Minimum number of tasks 1 ⓘ

Automatic task scaling policies you set cannot reduce the number of tasks below this number.

Desired number of tasks 3 ⓘ

Maximum number of tasks 10 ⓘ

Automatic task scaling policies you set cannot increase the number of tasks above this number.

IAM role for Service Auto Scaling ecsAutoscaleRole ⓘ

Then add a Target Tracking Scaling Policy, where Fargate automatically scales out if the average CPU Utilization is above 50% over the past 5 minutes:

Complete AWS ECS DevOps Masterclass for Beginners

Automatic task scaling policies

Scaling policy type ☒ Target tracking ☐ Step scaling ⓘ

Policy name* ⓘ

ECS service metric* ⓘ

Target value* ⓘ

Scale-out cooldown period seconds between scaling actions ⓘ

Scale-in cooldown period seconds between scaling actions ⓘ

Disable scale-in ☐ ⓘ

Review and Create the Service. It will take little bit until everything is registered and generated.

Testing the Service

Now it's time to test the load-balanced service:

Clusters > myfargate > Service: apache-service

Service : apache-service Update Delete

Cluster: myfargate
Status: ACTIVE
Task definition: simple-apache-server:1
Service type: REPLICAS
Launch type: FARGATE
Platform version: LATEST(1.3.0)
Service role: AWSServiceRoleForECS

Desired count: 3
Pending count: 3
Running count: 0

Details | **Tasks** | Events | Auto Scaling | Deployments | Metrics | Tags | Logs

Last updated on April 7, 2020 11:21:48 AM (0m ago) ⓘ

Task status: Running Stopped

Filter in this page

Task	Task Definition	Last status	Desired status	Group	Launch type	Platform version
551ac9c1-e229-4f8b-9472-e6a7...	simple-apache-server:1	PROVISIONING	RUNNING	service:apache-service	FARGATE	1.3.0
94ad5690-c256-4051-a44b-9357...	simple-apache-server:1	PROVISIONING	RUNNING	service:apache-service	FARGATE	1.3.0
a960e390-5223-46cb-89cf-22cf9...	simple-apache-server:1	PENDING	RUNNING	service:apache-service	FARGATE	1.3.0

Wait until the tasks go from pending to running. You can observe the logs in the meantime:

Complete AWS ECS DevOps Masterclass for Beginners

Clusters > myfargate > Service: apache-service

Service: apache-service Update Delete

Cluster: myfargate
Status: ACTIVE
Task definition: simple-apache-server:1
Service type: REPLICA
Launch type: FARGATE
Platform version: LATEST(1.3.0)
Service role: AWSServiceRoleForECS

Desired count: 3
Pending count: 2
Running count: 1

Details Tasks Events Auto Scaling Deployments Metrics Tags Logs

Task status: RUNNING STOPPED

Last updated on April 7, 2020 11:22:22 AM (0m ago)

Filter logs

Timestamp (UTC+00:00)	Message	Task
2020-04-07 11:22:20	172.31.34.178 -- [07/Apr/2020:09:22:20 +0000] "GET / HTTP/1.1" 200 45	94ad5690-c256-4051-a44b-935769b46d39
2020-04-07 11:22:20	172.31.34.178 -- [07/Apr/2020:09:22:20 +0000] "GET / HTTP/1.1" 200 45	a960e390-5223-46cb-89cf-22cf9089cdf
2020-04-07 11:22:06	[Tue Apr 07 09:22:06.791849 2020] [mpm_event.notice] [pid 1:tid 140278519280768] AH00489: Apache/2...	94ad5690-c256-4051-a44b-935769b46d39
2020-04-07 11:22:06	[Tue Apr 07 09:22:06.791979 2020] [core.notice] [pid 1:tid 140278519280768] AH00094: Command line: 'h...	94ad5690-c256-4051-a44b-935769b46d39
2020-04-07 11:22:03	[Tue Apr 07 09:22:03.207676 2020] [mpm_event.notice] [pid 1:tid 140533628273792] AH00489: Apache/2...	a960e390-5223-46cb-89cf-22cf9089cdf
2020-04-07 11:22:03	[Tue Apr 07 09:22:03.207857 2020] [core.notice] [pid 1:tid 140533628273792] AH00094: Command line: 'h...	a960e390-5223-46cb-89cf-22cf9089cdf

Then copy the URL from your load balancer and open a new tab:

Amazon ECS x Load Balancers | EC2 Management x +

eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#LoadBalancers:sort=loadBalancerName

aws Services Resource Groups

New EC2 Experience Tell us what you think

Create Load Balancer Actions

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones
fargate-lb	fargate-lb-265609707.eu-central-1.elb.amazonaws.com	provisioning	vpc-6570b40f	eu-central-1c, eu-central-1d

Basic configuration

Name	fargate-lb
ARN	arn:aws:elasticloadbalancing:eu-central-1:161952721022:loadbalancer/app/fargate-lb/5912295ef6ed1a14
DNS name	fargate-lb-265609707.eu-central-1.elb.amazonaws.com (A Record)

And open a new tab:

Amazon ECS x Load Balancers | EC2 Management x fargate-lb-265609707.eu-central-1 x +

Not secure | fargate-lb-265609707.eu-central-1.elb.amazonaws.com

It works!

Then have a look at where in the containers your requests are actually being routed:

Complete AWS ECS DevOps Masterclass for Beginners

Details	Tasks	Events	Auto Scaling	Deployments	Metrics	Tags	Logs
Task status RUNNING STOPPED							
Last updated on Ap							
Filter logs							
All 30s 5m 1h 6h 1d 1w							
Timestamp (UTC+00:00)	Message	Task					
2020-04-07 11:22:39	172.31.34.178 -- [07/Apr/2020:09:22:39 +0000] "GET / HTTP/1.1" 304 -	94ad5690-c256-4051-a44b-935769b46d39					
2020-04-07 11:22:38	172.31.34.178 -- [07/Apr/2020:09:22:38 +0000] "GET / HTTP/1.1" 304 -	94ad5690-c256-4051-a44b-935769b46d39					
2020-04-07 11:22:35	172.31.34.178 -- [07/Apr/2020:09:22:35 +0000] "GET / HTTP/1.1" 200 45	94ad5690-c256-4051-a44b-935769b46d39					
2020-04-07 11:22:20	172.31.34.178 -- [07/Apr/2020:09:22:20 +0000] "GET / HTTP/1.1" 200 45	94ad5690-c256-4051-a44b-935769b46d39					
2020-04-07 11:22:20	172.31.34.178 -- [07/Apr/2020:09:22:20 +0000] "GET / HTTP/1.1" 200 45	a960e390-5223-46cb-89cf-22cf9089cdaf					

Clean Up

To clean up

1. Delete the Application load balancer
2. Delete the Service in the cluster
3. Delete then also the Security groups, you will not need them anymore

Lab End
