



Data Engineering Cohort 1

Module 4

Assignment 4.5

Deploy Redis Service

Member #1

Name: Muhammad Humza

Roll Number: 2211-018-KHI-DEG

Member #2

Name: Muhammad Ovais

Roll Number: 2211-023-KHI-DEG

Task

Based on the solution from day 1

(/tasks/5_microservices_development/day_1_microservices/integrating_flask_redis/) add Redis as another ECS service and connect it with existing application.

Incorporate results from function *get_and_increase_hit_count()* into the application and show the results on the main page

Solution

Step #1 Creating a Cluster

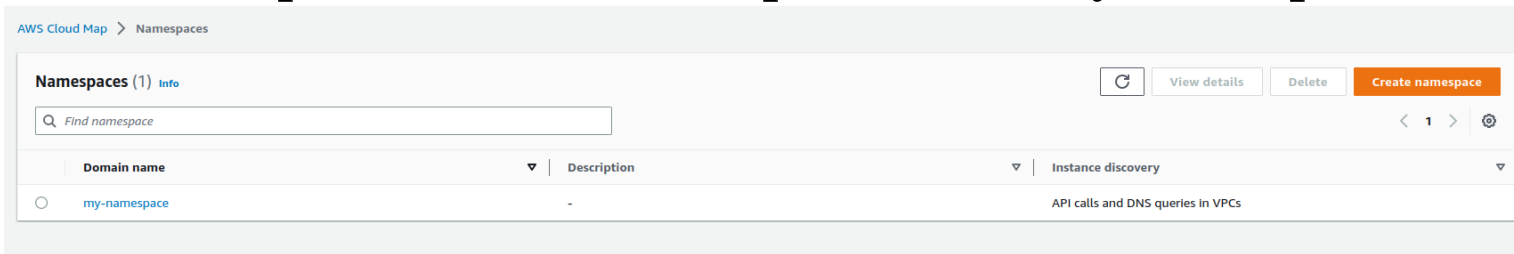
In this step we created a cluster name assignment4-5

The screenshot displays the AWS Management Console for the Amazon ECS cluster 'assignment4-5'. The cluster is in an 'ACTIVE' state. The 'Tasks' tab is selected, showing a list of tasks with columns for Task ID, Task definition, Container instance, Last status, Desired status, Started at, Started By, Group, Launch type, and Platform version. The tasks listed are 'frontend-3' (RUNNING), 'redis-5' (PENDING), and 'inference-3' (RUNNING).

Task	Task definition	Container instance	Last status	Desired status	Started at	Started By	Group	Launch type	Platform version
230e31abb4064b3...	frontend-3	--	RUNNING	RUNNING	2023-01-11 17:14:...	ecs-svc/95486741...	service:frontend	FARGATE	1.4.0
b92ae6bc886c490...	redis-5	--	PENDING	RUNNING		ecs-svc/62320182...	service:redis	FARGATE	1.4.0
ef830aafce524046...	inference-3	--	RUNNING	RUNNING	2023-01-11 17:18:...	ecs-svc/17987542...	service:inference	FARGATE	1.4.0

Step #2 Creating a namespace

In this step we created a namespace named my_namespace



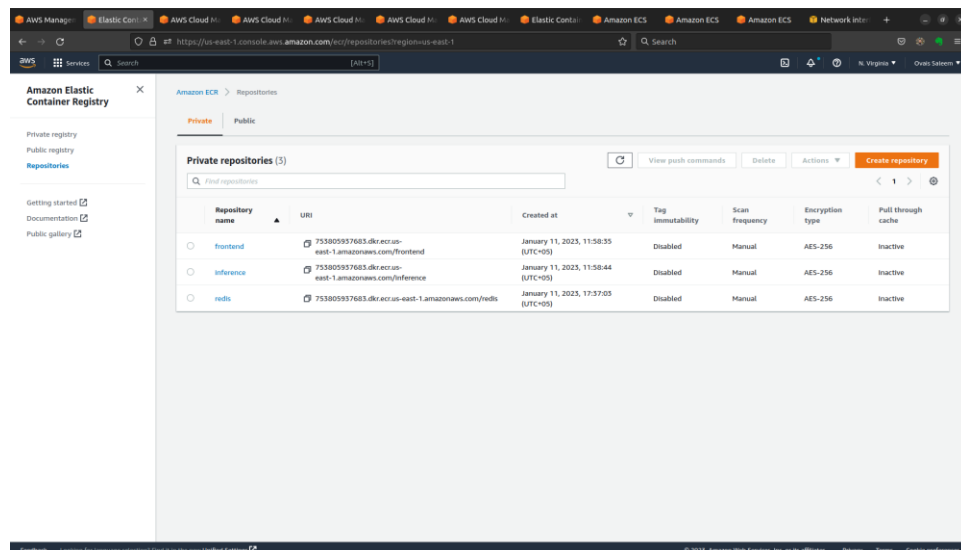
Step #3 Creating namespace services

After creating the namespace, we created 3 services.

You can see them in the namespace.pdf document.

Step #4 Creating Repositories

After creating the namespace services, we pushed the images.



Step #5 Pushing the Images

After creating the repositories, we pushed the images.

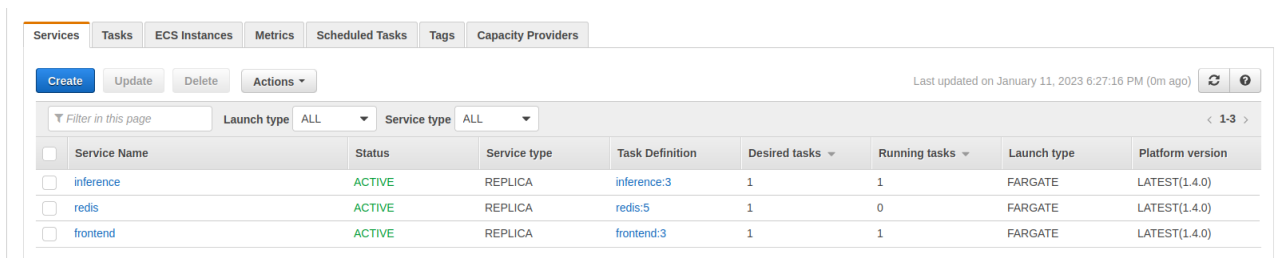
Step #6 Creating Task Definition

After pushing the images, we created task definitions for frontend, inference and redis.

You can find them in their respected pdf files located in the .zip file

Step #7 Creating ECS Service in Cluster Menu

After creating task definition, we created 3 ECS service in Cluster menu using Fargate launch type.



The screenshot shows the AWS ECS console interface. At the top, there are tabs for Services, Tasks, ECS Instances, Metrics, Scheduled Tasks, Tags, and Capacity Providers. Below the tabs, there are buttons for Create, Update, Delete, and an Actions dropdown. A status bar indicates the last update on January 11, 2023, at 6:27:16 PM (0m ago). Below this, there are filters for Launch type (ALL) and Service type (ALL). A table lists three services: inference, redis, and frontend. Each service is in an ACTIVE state, uses the REPLICATION service type, and is launched on FARGATE. The inference service has 1 desired task and 1 running task. The redis service has 1 desired task and 0 running tasks. The frontend service has 1 desired task and 1 running task. All services are using the LATEST(1.4.0) platform version.

Service Name	Status	Service type	Task Definition	Desired tasks	Running tasks	Launch type	Platform version
<input type="checkbox"/> inference	ACTIVE	REPLICATION	inference:3	1	1	FARGATE	LATEST(1.4.0)
<input type="checkbox"/> redis	ACTIVE	REPLICATION	redis:5	1	0	FARGATE	LATEST(1.4.0)
<input type="checkbox"/> frontend	ACTIVE	REPLICATION	frontend:3	1	1	FARGATE	LATEST(1.4.0)

Step #8 Navigating To Task

After creating 3 ECS service in Cluster menu we navigated to task tab.

Step #9 Selecting Existing Task

In the task tab we selected the existing task tab.

The screenshot displays the AWS Management Console for the Amazon ECS console. The left sidebar shows the navigation menu with 'Clusters' selected. The main content area shows the details for cluster 'assignment4-5'. The 'Tasks' tab is active, displaying a table of tasks. The table has columns for Task, Task definition, Container instance, Last status, Desired status, Started at, Started By, Group, Launch type, and Platform version. Three tasks are listed: 'frontend:3' (RUNNING), 'redis:5' (PENDING), and 'inference:3' (RUNNING).

Task	Task definition	Container instance	Last status	Desired status	Started at	Started By	Group	Launch type	Platform version
230e31abb4064b3...	frontend:3	--	RUNNING	RUNNING	2023-01-11 17:14:...	ecs-svc/95486741...	service:frontend	FARGATE	1.4.0
b92ae6bc886c49b0...	redis:5	--	PENDING	RUNNING		ecs-svc/62320182...	service:redis	FARGATE	1.4.0
ef830aa6ce524046...	inference:3	--	RUNNING	RUNNING	2023-01-11 17:18:...	ecs-svc/17987542...	service:inference	FARGATE	1.4.0

Step #10 Selecting ENI ID

After entering the existing task, we selected the ENI ID

The screenshot displays the AWS Management Console interface for an Amazon ECS task. The breadcrumb navigation shows 'Clusters > assignment4-5 > Task: b92ae6bc886c49f083b93727509e4f35'. The task name is 'Task : b92ae6bc886c49f083b93727509e4f35'. The 'Details' tab is selected, showing the following information:

- Cluster: assignment4-5
- Launch type: FARGATE
- Platform version: 1.4.0
- Task definition: redis:5
- Group: service.redis
- Task role: None
- Last status: ACTIVATING
- Desired status: RUNNING
- Created at: 2023-01-11 17:40:20 +0500

The 'Network' section shows the following details:

- Network mode: awsvpc
- ENI id: [eni-034bb3277d5b6fa93](#)
- Subnet id: subnet-00816d49c360cbf66
- Private IP: 10.0.0.227
- Public IP: 44.200.241.27
- Mac address: 02:17:ba:63:f9:b7

The 'Containers' section shows a table with one container:

Name	Container Runtime ID	Status	Image	Image Digest	CPU Units	Hard/Soft memor...	Essential	Resource ID
redis	b92ae6bc886c49f083...	RUNNING	753805937683.dkr.ecr.us-east-1.a...	sha256:d02ca7bffc11138063cb3ec...	--	--/--	true	7bac8dac-23d0-4...

At the bottom of the console, there is a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and '© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Step #11 Selecting Public IPV4 DNS

After selecting the ENI ID we copied the IPV4 DNS and pasted it in browser amd adding the port number 5000 in the end.

The screenshot shows the AWS Management Console interface for a Network Interface (eni-034bb327d5b6fa93). The console is organized into a sidebar on the left with navigation links like 'EC2 Dashboard', 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main content area displays the details of the selected Network Interface, including its status (In-use), VPC ID (vpc-070391ea5b9bf6fc4), Subnet ID (subnet-00816d49c36bcbf66), and IP addresses (Private: 10.0.0.227, Public: 44.200.241.27). The 'Public IPv4 DNS' is highlighted as 'ec2-44-200-241-27.compute-1.amazonaws.com'. Other details include the Elastic Fabric Adapter (False), IPv6 addresses (None), Elastic IP address owner (amazon), and IPv6 Prefix Delegation (None). The 'Instance details' section shows the Instance ID (i-602131350968) and Allocation ID (None). The 'Network interface attachment' section shows the Attachment ID (eni-attach-0385a8ba0a0f5781c) and Attachment time (Wed Jan 11 2023 17:40:25 GMT+0500 (Pakistan Standard Time)).

Attribute	Value
Network interface ID	eni-034bb327d5b6fa93
Network interface status	In-use
VPC ID	vpc-070391ea5b9bf6fc4
Subnet ID	subnet-00816d49c36bcbf66
Owner	753805937683
Source/dest. check	True
IP addresses	Private IPv4 address: 10.0.0.227, Public IPv4 address: 44.200.241.27
Secondary private IPv4 addresses	-
MAC address	02:17:ba:63:f9:b7
Instance details	Instance ID: i-602131350968, Allocation ID: -
Network interface attachment	Attachment ID: eni-attach-0385a8ba0a0f5781c, Attachment time: Wed Jan 11 2023 17:40:25 GMT+0500 (Pakistan Standard Time)

Step #12 Testing the Service

After running the page, we checked if the page is responding by uploading an image and seeing if it's shown in the browser.

