Assignment_4.5_Cloud_Deployment

Peer Members:

- **Syed Muhammad Raqim Ali Shah (2303.KHI.DEG.008)**
- **Maaz Javaid Siddique (2303.KHI.DEG.004)**
- **Qadeer Hussain (2303.KHI.DEG.006)**

Question:

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:

We sets up a connection to Redis using the SERVICE_DISCOVERY environment
variable in app.py.

```
app.py > ...
  1    import time
  2    import os
  3    import redis
  4    from flask import Flask
  5
  6    app = Flask(__name__)
  7    service=os.environ.get("SERVICE_DISCOVERY")
  8    cache = redis.Redis(host=service, port=6379)
  9
 10
 11    def get_and_increase_hit_count():
 12        retries = 5
 13        while True:
 14            try:
 15                return cache.incr("hits")
 16            except redis.exceptions.ConnectionError as exc:
 17                if retries == 0:
 18                    raise exc
 19                retries -= 1
 20                time.sleep(1)
 21
 22
 23    @app.route("/")
 24    def hello():
 25        count = get_and_increase_hit_count()
 26        return "Hello World! I have been seen {} times.\n".format(count)
 27    |
```

We sets environment using ENV SERVICE_DISCOVERY="redis": variable to specify the hostname of the Redis instance in Dockerfile.

```dockerfile
Dockerfile > ...
1    FROM python:3.7-alpine
2
3    WORKDIR /code
4
5    ENV FLASK_APP=app.py
6    ENV FLASK_RUN_HOST=0.0.0.0
7    ENV SERVICE_DISCOVERY="redis"
8
9    RUN apk add --no-cache gcc musl-dev linux-headers
10   COPY requirements.txt requirements.txt
11
12   RUN pip install -r requirements.txt
13
14   EXPOSE 5000
15
16   COPY . .
17
18   CMD ["flask", "run"]
```

Building Cluster

Clusters (1) Info

Create cluster

Search clusters

< 1 >

Cluster	Services	Tasks	Registered container instances	CloudWatch monitoring	Capacity provider strategy
redisflask	2	0 Pending | 4 Running	0	⊘ Default	No default found

Building services in cluster

redisflask

Cluster overview

ARN	Status	CloudWatch monitoring	Registered container instances
☐ redisflask	⊘ Active	🕐 Default	-

Services

Draining	Active
-	2

Tasks

Pending	Running
-	4

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

Services (2) Info

| ⟳ | Manage tags | Update | Delete service | **Create** |

☐	Service name ▽	Status ▽	ARN	Service t... ▽	Deployments and tasks ▽	Last deploy... ▽	Task defi... ▽	Revision
☐	flask	⊘ Active	☐ arn:aws:ecs...	REPLICA	▬▬▬▬▬ 2/2 Tasks ru...	⊘ Completed	flask	1
☐	redis	⊘ Active	☐ arn:aws:ecs...	REPLICA	▬▬▬▬▬ 2/2 Tasks ru...	⊘ Completed	redis	1

flask Info

Health and metrics | Logs | Configuration and tasks | Deployments and events | Networking | Tags

Status Info

ARN	Status	Tasks	Deployments current state
☐ redisflask/flask	⊘ Active	▬▬▬▬ 0 Pending , 2 Running / 2 Desired	⊘ 2 Completed

Health

1h 3h 12h 1d 3d 1w Custom 📅 ⟳ ▽ Add to dashboard

CPU utilization — Percent (CPUUtilization Minimum, CPUUtilization Maximum, CPUUtilization Average)

Memory utilization — Percent (MemoryUtilization Minimum, MemoryUtilization Maximum, MemoryUtilization Average)

flask Info

Update service   Delete service
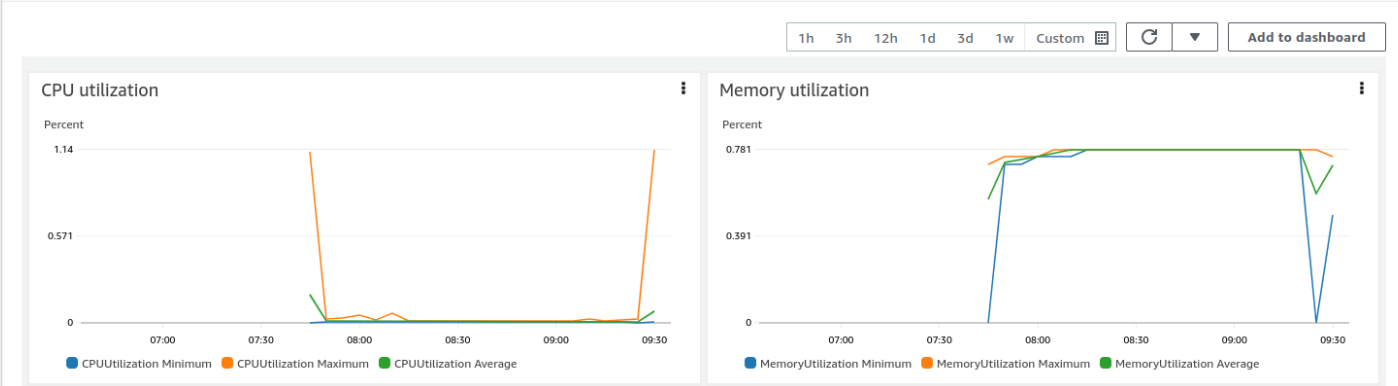
Health and metrics | **Logs** | Configuration and tasks | Deployments and events | Networking | Tags

Logs (53+) Info

Since 1 hour ago   View in CloudWatch ⧉

Filter logs by message   flask

Timestamp (Local)	Message	Task ⧉	Container
5/16/2023, 2:29:51 PM GMT+5	▯[31m▯[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.▯[0m	53f4ecf269bd43678e 1a3517b1c40a9c	flask
5/16/2023, 2:29:51 PM GMT+5	* Running on all addresses (0.0.0.0)	53f4ecf269bd43678e 1a3517b1c40a9c	flask
5/16/2023, 2:29:51 PM GMT+5	* Running on http://127.0.0.1:5000	53f4ecf269bd43678e 1a3517b1c40a9c	flask
5/16/2023, 2:29:51 PM GMT+5	* Running on http://10.0.0.114:5000	53f4ecf269bd43678e 1a3517b1c40a9c	flask
5/16/2023, 2:29:51 PM GMT+5	▯[33mPress CTRL+C to quit▯[0m	53f4ecf269bd43678e 1a3517b1c40a9c	flask
5/16/2023, 2:29:51 PM GMT+5	* Serving Flask app 'app.py'	53f4ecf269bd43678e 1a3517b1c40a9c	flask
5/16/2023, 2:29:51 PM GMT+5	* Debug mode: off	53f4ecf269bd43678e 1a3517b1c40a9c	flask
5/16/2023, 2:29:49 PM GMT+5	▯[31m▯[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.▯[0m	9bb1d4582c1f4a999 040f6bbaf8db947	flask
5/16/2023, 2:29:49 PM GMT+5	* Running on all addresses (0.0.0.0)	9bb1d4582c1f4a999 040f6bbaf8db947	flask

redis Info

Update service   Delete service

Health and metrics | Logs | Configuration and tasks | Deployments and events | Networking | Tags

Status Info

ARN	Status	Tasks	Deployments current state
▢ redisflask/redis	⊘ Active	0 Pending , 2 Running / 2 Desired	⊘ 2 Completed

Health

1h 3h 12h 1d 3d 1w Custom ⊞   Add to dashboard

CPU utilization — Percent

Memory utilization — Percent
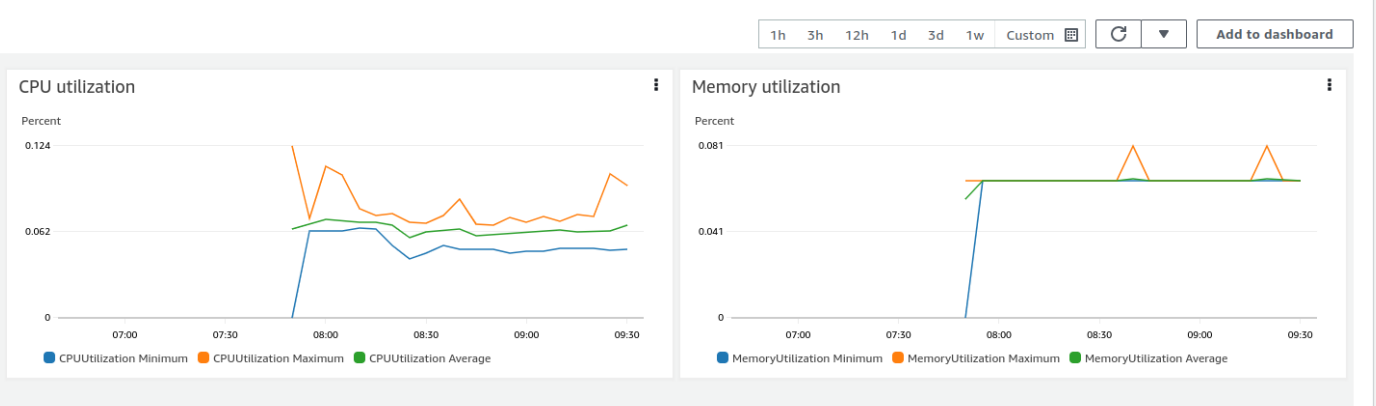
redis Info

Update service   Delete service

Health and metrics | **Logs** | Configuration and tasks | Deployments and events | Networking | Tags

Logs (35+) Info

Since 1 hour ago   View in CloudWatch

Filter logs by message

redis

Timestamp (Local)	Message	Task	Container
5/16/2023, 2:32:02 PM GMT+5	1:M 16 May 2023 09:32:02.107 * DB saved on disk	2b384826df2f4a5caad0078e87d4968e	redis
5/16/2023, 2:32:01 PM GMT+5	1:M 16 May 2023 09:32:01.372 * DB saved on disk	2b384826df2f4a5caad0078e87d4968e	redis
5/16/2023, 2:31:59 PM GMT+5	1:M 16 May 2023 09:31:59.683 * DB saved on disk	2b384826df2f4a5caad0078e87d4968e	redis
5/16/2023, 2:31:59 PM GMT+5	1:M 16 May 2023 09:31:59.423 * DB saved on disk	2b384826df2f4a5caad0078e87d4968e	redis
5/16/2023, 2:31:58 PM GMT+5	1:M 16 May 2023 09:31:58.943 * DB saved on disk	2b384826df2f4a5caad0078e87d4968e	redis
5/16/2023, 2:31:57 PM GMT+5	1:M 16 May 2023 09:31:57.213 * DB saved on disk	2b384826df2f4a5caad0078e87d4968e	redis
5/16/2023, 2:31:56 PM GMT+5	1:M 16 May 2023 09:31:56.712 * DB saved on disk	2b384826df2f4a5caad0078e87d4968e	redis
5/16/2023, 2:28:02 PM GMT+5	1:M 16 May 2023 09:28:02.782 * DB saved on disk	d7b4281d38f242b19e6ffa11efcf37f7	redis
5/16/2023, 2:28:02 PM GMT+5	1:M 16 May 2023 09:28:02.135 * DB saved on disk	d7b4281d38f242b19e6ffa11efcf37f7	redis

AWS Cloud Map > Namespaces

Namespaces (1) Info

View details   Delete   Create namespace

Find namespace

Domain name	Description	Instance discovery	Namespace ID
redis.com	-	API calls and DNS queries in VPCs	ns-dohlwxwcvt3t5ie5

Building Namespace of name redis.com

Namespace: redis.com Info

Edit | Delete

Namespace information

Name	**Instance discovery**
redis.com	API calls and DNS queries in VPCs
Namespace ID	**Date created**
ns-dohlwxwcvt3t5ie5	May 16, 2023, 4:53 AM UTC
Description	**SOA TTL**
-	60
HTTP name	
redis.com	

Route 53 hosted zone

A hosted zone is a container that holds information about how you want to route traffic. AWS Cloud Map automatically creates a Route 53 hosted zone.

Hosted zone ID	**Associated VPCs**
Z05062913JP3QWE2K6ORZ ↗	vpc-00257d507ac7614c3 ↗ us-east-1
Description	
Created by AWS Cloud Map namespace with ARN arn:aws:servicediscovery:us-east-1:664748749082:namespace/ns-dohlwxwcvt3t5ie5	
Type	
Private hosted zone	
Record count	
6	

Service: redis Info

Edit | Delete

Service information

Name	**Service ID**
redis	srv-4ocwdism7kpe4brn
Namespace name	**Description**
redis.com	-
Namespace ID	**Discoverable By**
ns-dohlwxwcvt3t5ie5	DNS_HTTP
Namespace HTTP name	
redis.com	

DNS configuration

DNS routing policy		**Record type**	**TTL**
Weighted routing		A	300

Health check configuration

Health check type
No health check configured

Route 53 hosted zone

A hosted zone is a container that holds information about how you want to route traffic. AWS Cloud Map automatically creates a Route 53 hosted zone.

Hosted zone ID

Z05062913JP3QWE2K6ORZ [↗]

Description

Created by AWS Cloud Map namespace with ARN arn:aws:servicediscovery:us-east-1:664748749082:namespace/ns-dohlwxwcvt3t5ie5

Type

Private hosted zone

Record count

6

Associated VPCs

vpc-00257d507ac7614c3 [↗] us-east-1

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value.

Edit tags

Key ▽	Value

No tags

There are no tags for this resource.

Services (1)

View details   Delete   Create service

🔍 Find service

< 1 >   ⚙

	Name ▽	Description ▽	Discoverable By ▽	Date created ▽	Service ID ▽
○	redis	-	API Calls and DNS Queries	May 16, 2023, 4:54 AM UTC	srv-4ocwdism7kpe4brn

Push commands for flask   ✕

macOS / Linux | Windows

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting 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 us-east-1 | docker login --username AWS --password-stdin
   >   664748749082.dkr.ecr.us-east-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 flask .

3. After the build completes, tag your image so you can push the image to this repository:

   > ⧉   docker tag flask:latest 664748749082.dkr.ecr.us-east-1.amazonaws.com/flask:latest

4. Run the following command to push this image to your newly created AWS repository:

   > ⧉   docker push 664748749082.dkr.ecr.us-east-1.amazonaws.com/flask:latest

[Close]

Steps to authenticate and push our Docker image to an AWS ECR repository:

```
● syedmuhammadraqimali@all-MS-7D35:~/Documents/data_engineering_bootcamp_2303/tasks/4_microservices_development/day_1_microservices/integ
  rating_flask_redis$ aws ecr get-login-password --region us-eas
  t-1 | docker login --username AWS --password-stdin 664748749082.dkr.ecr.us-east-1.amazonaws.com
  WARNING! Your password will be stored unencrypted in /home/syedmuhammadraqimali/.docker/config.json.
  Configure a credential helper to remove this warning. See
  https://docs.docker.com/engine/reference/commandline/login/#credentials-store

  Login Succeeded
● syedmuhammadraqimali@all-MS-7D35:~/Documents/data_engineering_bootcamp_2303/tasks/4_microservices_development/day_1_microservices/integ
  rating_flask_redis$ docker build -t flask .
  [+] Building 1.6s (11/11) FINISHED

   => [internal] load build definition from Dockerfile
                                                                      0.0s
   => => transferring dockerfile: 327B
                                                                      0.0s
   => [internal] load .dockerignore
                                                                      0.0s
   => => transferring context: 2B
                                                                      0.0s
   => [internal] load metadata for docker.io/library/python:3.7-alpine
                                                                      1.5s
   => [1/6] FROM docker.io/library/python:3.7-alpine@sha256:f48c5f6a8a22a73558ea93eb26d2c7928d23f2acb2bb9270be9a08adc2bfa63d
                                                                      0.0s
   => [internal] load build context
                                                                      0.0s
   => => transferring context: 793B
                                                                      0.0s
   => CACHED [2/6] WORKDIR /code
                                                                      0.0s
   => CACHED [3/6] RUN apk add --no-cache gcc musl-dev linux-headers
                                                                      0.0s
   => CACHED [4/6] COPY requirements.txt requirements.txt
                                                                      0.0s
   => CACHED [5/6] RUN pip install -r requirements.txt
                                                                      0.0s
   => [6/6] COPY . .
                                                                      0.0s
   => exporting to image
                                                                      0.0s
   => => exporting layers
                                                                      0.0s
   => => writing image sha256:a08e0a2d23bf4305abcd2374e12c8db74db7c485bf906c1325538a0fb8dd8517
                                                                      0.0s
   => => naming to docker.io/library/flask
```

```
● syedmuhammadraqimali@all-MS-7D35:~/Documents/data_engineering_bootcamp_2303/tasks/4_microservices_development/day_1_microservices/integ
  rating_flask_redis$ docker tag flask:latest 664748749082.dkr.ecr.us-east-1.amazonaws.com/flask:latest
● syedmuhammadraqimali@all-MS-7D35:~/Documents/data_engineering_bootcamp_2303/tasks/4_microservices_development/day_1_microservices/integ
  rating_flask_redis$ docker push 664748749082.dkr.ecr.us-east-1.amazonaws.com/flask:latest
  The push refers to repository [664748749082.dkr.ecr.us-east-1.amazonaws.com/flask]
  fbb1311383cb: Pushed
  6cc793937304: Layer already exists
  3466b493674e: Layer already exists
  f928a7f5ed7c: Layer already exists
  ac4a25f4be5e: Layer already exists
  dc6a6a6fc818: Layer already exists
  faf320a00dfc: Layer already exists
  37411a7a419e: Layer already exists
  208977ac81d7: Layer already exists
  bb01bd7e32b5: Layer already exists
  latest: digest: sha256:e9993f62f94281db7f3d74c140931e0db6211ab9baf0a6e2e3092ab25ddb85d8 size: 2412
```

Our ECR Repository:

Private   Public

Private repositories (2)

[Find repositories]

View push commands   Delete   Actions ▼   **Create repository**

< 1 >

	Repository name ▲	URI	Created at ▽	Tag immutability	Scan frequency	Encryption type	Pull through cache
☐	flask	664748749082.dkr.ecr.us-east-1.amazonaws.com/flask	May 16, 2023, 09:40:35 (UTC+05)	Disabled	Manual	AES-256	Inactive
☐	redis	664748749082.dkr.ecr.us-east-1.amazonaws.com/redis	May 16, 2023, 09:39:16 (UTC+05)	Disabled	Manual	AES-256	Inactive

flask Info

[↻]   Update service   Delete service

Health and metrics   Logs   **Configuration and tasks**   Deployments and events   Networking   Tags

Service configuration Info

Service ARN
redisflask/flask

Service Connect Info
Configure

Service Connect namespace
-

Service Connect services
-

Task definition: revision
flask:1

Launch type
FARGATE

Service type
REPLICA

Created by
arn:aws:iam::664748749082:root

▶ Service discovery

Auto Scaling

Desired tasks
2

Min tasks
-

Max tasks
-

redis Info

C | Update service

Health and metrics | Logs | **Configuration and tasks** | Deployments and events | Networking | Tags

Service configuration Info

Service ARN	Service Connect Info	Service Connect namespace	Service Connect services
⬚ redisflask/redis	Configure	-	-
Task definition: revision	**Launch type**	**Service type**	**Created by**
redis:1	FARGATE	REPLICA	arn:aws:iam::664748749082:root

▶ Service discovery

Auto Scaling

Desired tasks	Min tasks	Max tasks
2	-	-

Task definition in ECR

Task definitions (2) Info

C | Deploy ▼ | Create new revision ▼ | **Create new task definition ▼**

🔍 Filter task definitions by property or value   2 matches   ‹ 1 › ⚙

Status of last revision = ACTIVE ✕ | Clear filters

	Task definition ▽	Status of last revision ▽
○	flask	⊘ ACTIVE
○	redis	⊘ ACTIVE

flask:1

Deploy ▼ | Actions ▼ | Create new revision ▼

Overview Info

ARN	Status	Time created	App environment
arn:aws:ecs:us-east-1:664748749082:task-definition/flask:1	⊘ ACTIVE	5/16/2023, 06:00:29 UTC	FARGATE

Task role	Task execution role	Operating system/Architecture	Network mode
-	ecsTaskExecutionRole ↗	Linux/X86_64	awsvpc

Containers | JSON | Storage | Tags

Task size

Task CPU	Task memory
1 vCPU	3 GB

Containers Info

Container name	Image	Private registry	Essential	CPU	Memory	GPU
flask	664748749082.dkr....	-	Yes	0	-	-

redis:1

Deploy ▼ | Actions ▼ | Create new revision ▼

Overview Info

ARN	Status	Time created	App environment
arn:aws:ecs:us-east-1:664748749082:task-definition/redis:1	⊘ ACTIVE	5/16/2023, 06:01:08 UTC	FARGATE

Task role	Task execution role	Operating system/Architecture	Network mode
-	ecsTaskExecutionRole ↗	Linux/X86_64	awsvpc

Containers | JSON | Storage | Tags

Task size

Task CPU	Task memory
1 vCPU	3 GB

Containers Info

Container name	Image	Private registry	Essential	CPU	Memory	GPU
redis	664748749082.dkr....	-	Yes	0	-	-

Final result shows the interface representing how many times the site is being hit

ec2-3-234-249-202.compute-1.amazonaws.com:5000

Hello World! I have been seen 18 times.

Increments in number of hits on every hit

ec2-3-234-249-202.compute-1.amazonaws.com:5000

Hello World! I have been seen 19 times.