

## Weekly Tasks

	Description	Deliverable
<b>Week 6</b>	<p>In this task, you will deploy Metabase on Amazon ECS using the Fargate launch type and connect it to a PostgreSQL database hosted on Amazon RDS. You'll use the official Metabase Docker image from Docker Hub and configure the environment variables needed for database connectivity.</p> <p>Ensure both ECS and RDS are in the same VPC to allow communication. The RDS security group should allow inbound traffic on port 5432 from the ECS task.</p>	<p>Provide clear screenshots showing the RDS PostgreSQL instance details, the ECS task definition and running service, the security group rules allowing traffic from ECS to RDS on port 5432, and the Metabase setup screen confirming a successful connection to the database.</p>

## WEEK 6

### STEP 1: Set Up the Database (PostgreSQL on RDS)

Go to: RDS > Databases > Create database

Choose:

Engine: PostgreSQL

Template: Free Tier (if eligible)

DB instance identifier: metabase-db

Username: metabase\_user

Password: Choose and remember it

Aurora and RDS > Databases > metabase

**metabase**

**Summary**

- DB identifier: metabase
- Status: Available
- Role: Instance
- Engine: PostgreSQL
- Region & AZ: us-east-1f

**Connectivity & security**

Endpoint & port	Networking	Security
Endpoint: metabase.co5qakycehh.us-east-1.rds.amazonaws.com Port: 5432	Availability Zone: us-east-1f VPC: vpc-0ebb3f57ccb0485 Subnet group: default-vpc-0ebb3f57ccb0485 Subnets:	VPC security groups: default (sg-086310df28ac72ec6) (Active) Publicly accessible: No Certificate authority: rds-ca-rsa2048-g1

## STEP 2: Create ECS Cluster

Go to: ECS > Clusters > Create Cluster  
 Choose: “Networking only (Fargate)”  
 Name it: metabase-cluster

Amazon Elastic Container Service > Clusters > metabase-clus > Services

**metabase-clus**

**Cluster overview**

ARN	Status	CloudWatch monitoring	Registered container instances
arn:aws:ecs:us-east-1:48887424811:cluster/metabase-clus	Active	Default	-

**Services**

Draining	Active	Pending	Running
-	1	-	1

**Tasks**

Service	Count
metabase-tsk-service	1

**Services (1) Info**

Service name	ARN	Status	Scheduled tasks	Task definitions	Deployments and tasks
metabase-tsk-service	arn:aws:ecs:us-east-1:48887424811:task-definition/metabase-tsk-service:1	Active	REPLICA	FARGATE	metabase... 1/1 Tasks

## STEP 3: Create Security Groups

ECS Task SG

Name: ecs-task-sg

Inbound Rule: allow TCP 3000 from the Load Balancer SG

Load Balancer SG

Name: alb-sg

Inbound Rule: allow HTTP 80 from 0.0.0.0/0

RDS SG

Name: rds-sg

Inbound Rule: allow TCP 5432 from ECS Task SG

The screenshot shows the AWS EC2 Security Groups interface. On the left, a sidebar lists various AWS services like Instances, Images, and Network & Security. The main panel displays the details for a security group named 'sg-086310dfc8ac72ec6 - default'. It shows the security group name is 'default', the owner is '488874248115', and it has a VPC ID of 'vpc-0ebb33f37ccb0485'. The 'Inbound rules' tab is selected, showing four rules:

Port range	Source	Description
5432	sg-0f8cf44350257490d...	-
80	0.0.0.0/0	-
All	sg-086310dfc8ac72ec6 ...	-
443	0.0.0.0/0	-

This screenshot shows the details for a security group named 'sg-0f8cf44350257490d - ecs-sg-tasks'. The details pane includes the security group name 'ecs-sg-tasks', owner '488874248115', and a description 'allows communication with RDS'. The VPC ID is 'vpc-0ebb33f37ccb0485'. The 'Inbound rules' tab is selected, showing three rules:

Type	Protocol	Port range	Source	Description
HTTP	TCP	80	0.0.0.0/0	-
Custom TCP	TCP	3000	sg-086310dfc8ac72ec6 ...	-
HTTPS	TCP	443	0.0.0.0/0	-

## STEP 4: Create Target Group

Go to: EC2 > Target Groups > Create

Name: metabase-tg

Target type: IP

Protocol/Port: HTTP / 3000

Health check settings:

Protocol: HTTP

Path: /

Port: traffic port

Healthy threshold: 3

Unhealthy threshold: 5

Timeout: 10s  
Interval: 30s

The screenshot shows the AWS EC2 Target Groups console. The left sidebar navigation includes: AMI Catalog, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Settings. Under Load Balancing, 'Target groups' is selected. The main panel displays the 'metabase-tg' target group. The 'Details' section shows the ARN: arn:aws:elasticloadbalancing:us-east-1:488874248115:targetgroup/metabase-tg/f806d670249060fe. It lists the Target type as IP, Protocol as HTTP: 3000, and Protocol version as HTTP1. The VPC is set to vpc-0ebb33f37ccbf0485. The 'Health checks' tab is active, showing 0 Total targets, 0 Healthy, 0 Unhealthy, 0 Unused, 0 Initial, and 0 Draining. Below this, the 'Health check settings' section is shown, with Protocol set to HTTP, Path to /, Port set to Traffic port, and Healthy threshold set to 3 consecutive health check successes. The unhealthy threshold is 5 consecutive health check failures. The timeout is 10 seconds, and the interval is 30 seconds. Success codes are listed as 200.

## STEP 5: Create Load Balancer

Go to: EC2 > Load Balancers > Create Application Load Balancer

Name: metabase-lb

Scheme: internet-facing

Listeners: HTTP (port 80)

Select 2 public subnets

Security group: alb-sg

Forwarding rule:

Port 80 → Target group: metabase-tg

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar navigation includes: AMI Catalog, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Settings. Under Load Balancing, 'Load balancers' is selected. The main panel displays the 'metabase-lb' application load balancer. The 'Details' section shows the Load balancer type as Application, Status as Active, and Hosted zone as Z355XDQTRQ7X7K. The VPC is vpc-0ebb33f37ccbf0485. The Availability Zones listed are subnet-05a49f56ea62db8, subnet-0a27e044943442d6, subnet-0c1b9fa38e9d82a8c, subnet-036b563c66d09ecb8, subnet-0d2f1cd7f62a20d22, and subnet-057dd3f46ca48011c. The Load balancer IP address type is IPv4. The DNS name is metabase-lb-143720316.us-east-1.elb.amazonaws.com (A Record). The 'Listeners and rules' tab is active, showing a single listener for port 80. The 'Network mapping' tab shows the target group metabase-tg. The 'Monitoring' tab shows CloudWatch Metrics and CloudWatch Logs. The 'Integrations' tab shows AWS Lambda and AWS Step Functions. The 'Attributes' tab shows the load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:488874248115:loadbalancer/app/metabase-lb/6848c6b2fc4e4b46. The 'Canary' and 'Tags' tabs are also present.

## STEP 6: Create Task Definition (Fargate)

Go to ECS > Task Definitions > Create new task definition

Launch type: Fargate

Task name: metabase-tsk

CPU/Memory: 1 vCPU, 3 GB

Add container:

Name: metabase

Image: metabase/metabase:latest

Port mapping: 3000

Environment Variables:

MB\_DB\_TYPE=postgres

MB\_DB\_DBNAME=metabase

MB\_DB\_PORT=5432

MB\_DB\_USER=metabase\_user

MB\_DB\_PASS=your\_password

MB\_DB\_HOST=your-rds-endpoint.rds.amazonaws.com

```
"healthCheck": {  
    "command": ["CMD-SHELL", "curl -f http://localhost:3000 || exit 1"],  
    "interval": 30,  
    "timeout": 5,  
    "retries": 3,  
    "startPeriod": 60  
}
```

The screenshot shows the AWS Elastic Container Service (ECS) Task Definition Overview page for a task named 'metabase-tsk:1'. The task is currently active and was created on July 30, 2025. It runs on Fargate and uses the 'awslogs' network mode. The task role is 'ecsTaskExecutionRole'. The task size is set to 1 vCPU and 3 GB of memory. The task memory usage is shown as 3,072 MiB (3 GB). The page also includes tabs for JSON, Task placement, Volumes (0), Requires attributes, and Tags.

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there's a search bar and a navigation bar with tabs for 'Metrics' and 'Logs'. Below the search bar, there are two log streams: 'lambda-1' and 'lambda-2'. The 'lambda-1' stream is selected and has a preview pane showing log entries. The first entry is timestamped at 2023-09-07T10:45:00+00:00 and contains the message 'Hello from Lambda!'. The second entry is timestamped at 2023-09-07T10:45:00+00:00 and contains the message 'Hello from Lambda!'. The bottom of the screen shows the AWS footer with links to 'AWS Home', 'AWS Support', 'AWS Marketplace', 'AWS Documentation', and 'AWS API Reference'.

```
18     ],
19     "essential": true,
20     "environment": [
21       {
22         "name": "MB_DB_DBNAME",
23         "value": "metabase"
24       },
25       {
26         "name": "MB_DB_HOST",
27         "value": "metabse.c0qakyicch.us-east-1.rds.amazonaws.com"
28       },
29       {
30         "name": "MB_DB_PASS",
31         "value": "Security123!"
32       },
33       {
34         "name": "MB_DB_PORT",
35         "value": "5432"
36       },
37       {
38         "name": "MB_DB_TYPE",
39         "value": "postgres"
40       },
41       {
42         "name": "MB_DB_USER",
43         "value": "metabase"
44       }
45     ],
46     "environmentFiles": [],
47     "mountPoints": []
48   ],
```

## STEP 7: Create ECS Service

Go to ECS > Clusters > metabase-cluster > Create service

Launch type: Fargate

Task Definition: metabase-tsk

Service name: metabase-service

Number of tasks: 1

VPC: Same as your RDS and LB

Subnets: Select 2 public subnets

Security Group: ecs-task-sg

Enable Load Balancer integration:

Type: Application Load Balancer

Listener: HTTP 80

Target Group: metabase-tg

Click Deploy Service

The screenshot shows the AWS Elastic Container Service (ECS) console. On the left, there's a sidebar with links to Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main area displays the 'metabase-tsk-service' info page. At the top, it shows 'Status: Active', 'Tasks (1 Desired)', 'Task definition: revision metabase-tsk-1', and 'Deployment status: Success'. Below this, tabs for Health and metrics, Tasks, Logs, Deployments, Events, Configuration and networking, Service auto scaling, and Tags are visible. Under 'Health and metrics', there's a 'Status' section with details like Service name (metabase-tsk-service), Service ARN (arn:aws:ecs:us-east-1:488874248115:service/metabase-clus/metabase-tsk-service), Deployments current state (1 Completed Task), and Created at (July 30, 2025 at 13:20 (UTC+1:00)). It also shows Load balancer health for 'metabase-lb' with an Application Load Balancer port 3000 mapping to Target group 'metabase-tg' (1 Healthy, 0 Unhealthy). The bottom of the page includes CloudShell, Feedback, and standard AWS footer links.

Go to the Load Balancer, copy the DNS name and paste it in a new tab

The screenshot shows the AWS Elastic Load Balancing (ELB) console. The left sidebar includes links for AMI Catalog, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), Auto Scaling (Auto Scaling Groups), and Settings. The main area shows the 'metabase-lb' load balancer details. It lists subnets (subnet-056b565c6609ec0b, subnet-0d2f1cd7f62a20d22, subnet-057d3f46ca48011c), the Load balancer ARN (arn:aws:elasticloadbalancing:us-east-1:488874248115:loadbalancer/app/metabase-lb/6848c6b2fc4e4b46), and the DNS name (metabase-lb-143720316.us-east-1.elb.amazonaws.com [A Record]). Below this, the 'Listeners and rules' tab is selected, showing one rule for port HTTP:80 forwarding to target group 'metabase-tg' (100% weight) with target group stickiness off. Other tabs include Network mapping, Resource map, Security, Monitoring, Integrations, Attributes, Capacity, and Tags.

Not secure metabase-lb-143720316.us-east-1.elb.amazonaws.com

Home

How to use Metabase

COLLECTIONS

- Our analytics
- Your personal collection
- Examples
- Other users' personal collections

BROWSE

- Models
- Databases
- Metrics

Trash

+ Add database

Hey there, Ayomitope

Try out these sample x-rays to see what Metabase can do.

- A summary of Accounts
- A glance at People
- A look at Orders
- A summary of Analytic Events

- A glance at Products
- Some Insights about Feedback
- Some insights about Reviews

A look at Invoices

Metabase tips

metabase-lb-143720316.us-east-1.elb.amazonaws.com/auto/dashboard/table/5