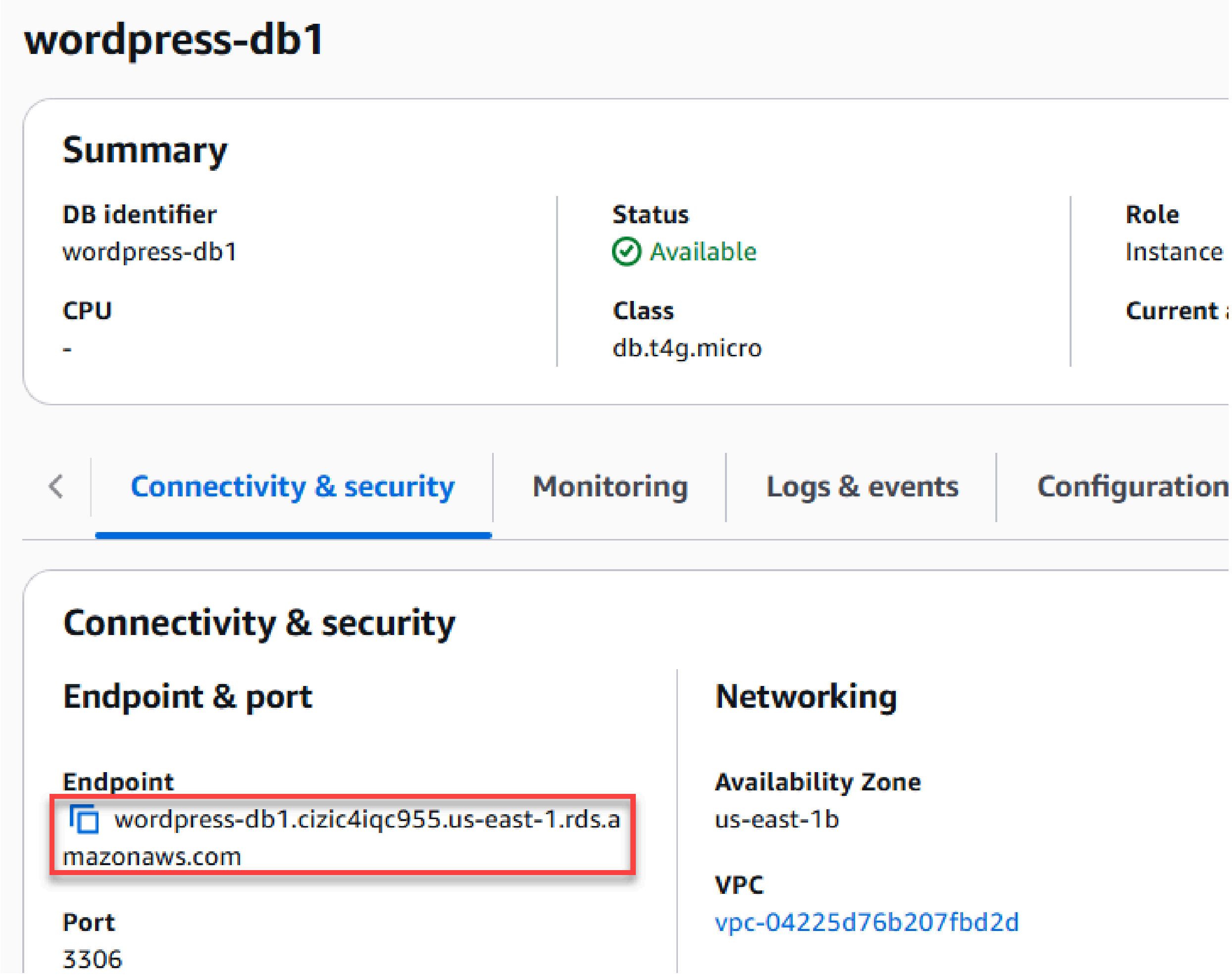
WordPress Deployment using ECS and RDS - Assignment

RDS Creation



wordpress-db1

Summary

DB identifier
wordpress-db1

CPU
-

Status
✓ Available

Class
db.t4g.micro

Role
Instance

Current

Connectivity & security

Endpoint & port

Endpoint
wordpress-db1.cizic4iqc955.us-east-1.rds.amazonaws.com

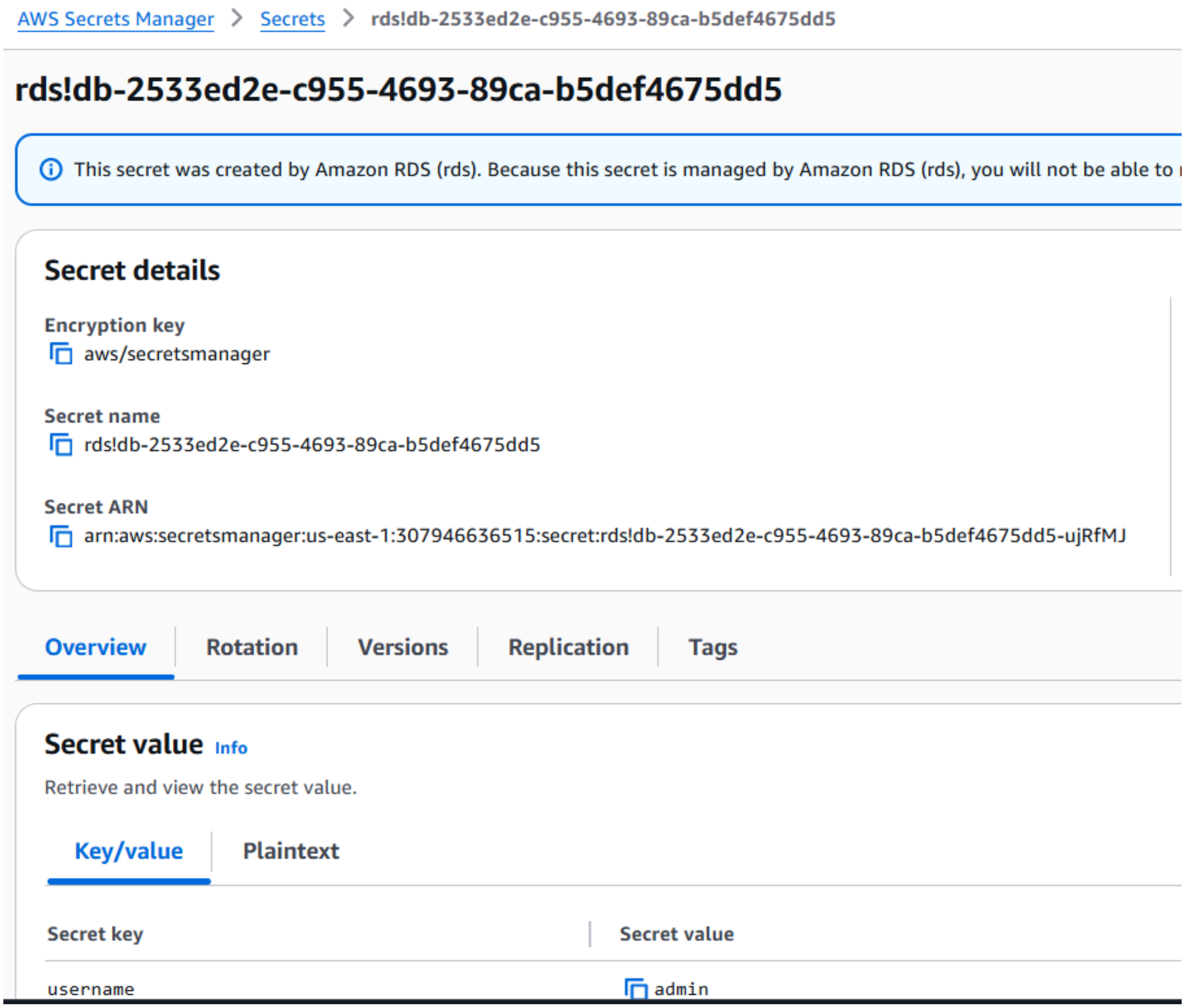
Port
3306

Networking

Availability Zone
us-east-1b

VPC
vpc-04225d76b207fbd2d

AWS Secret Manager



AWS Secrets Manager > Secrets > rds!db-2533ed2e-c955-4693-89ca-b5def4675dd5

rds!db-2533ed2e-c955-4693-89ca-b5def4675dd5

ⓘ This secret was created by Amazon RDS (rds). Because this secret is managed by Amazon RDS (rds), you will not be able to

Secret details

Encryption key
aws/secretsmanager

Secret name
rds!db-2533ed2e-c955-4693-89ca-b5def4675dd5

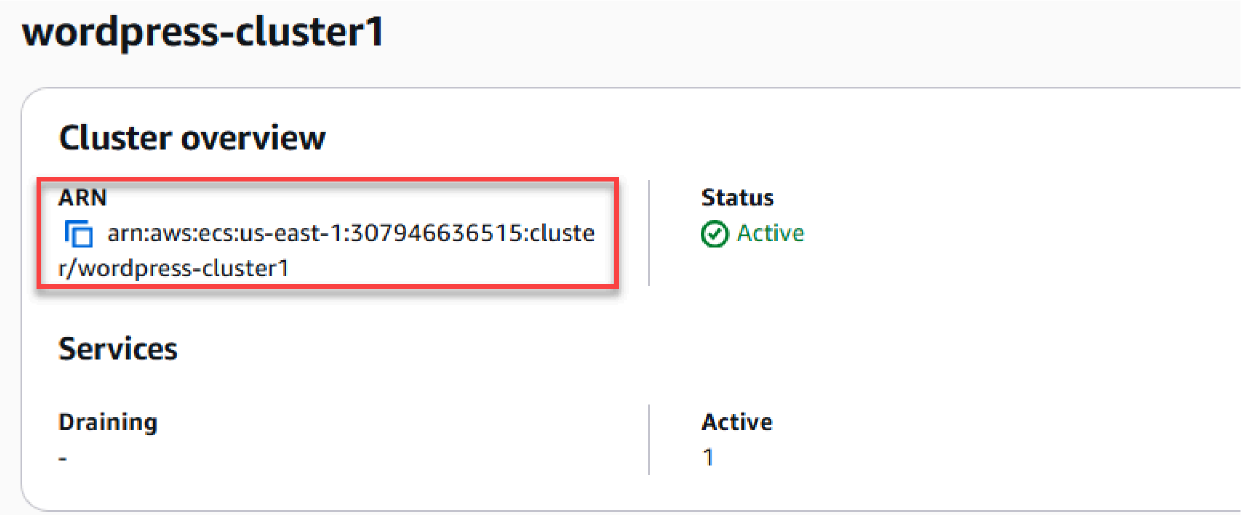
Secret ARN
arn:aws:secretsmanager:us-east-1:307946636515:secret:rds!db-2533ed2e-c955-4693-89ca-b5def4675dd5-ujRfMJ

Secret value

Retrieve and view the secret value.

Secret key	Secret value
username	admin

ECS Cluster Creation



wordpress-cluster1

Cluster overview

ARN
arn:aws:ecs:us-east-1:307946636515:cluster/wordpress-cluster1

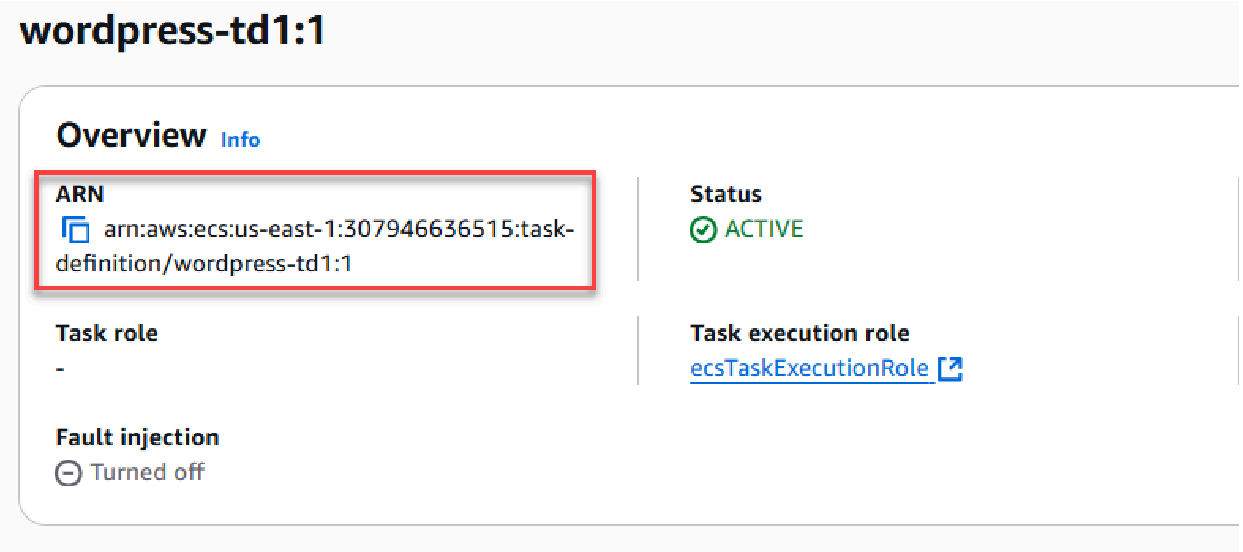
Status
Active

Services

Draining
-

Active
1

ECS Task Definition



wordpress-td1:1

Overview Info

ARN
arn:aws:ecs:us-east-1:307946636515:task-definition/wordpress-td1:1

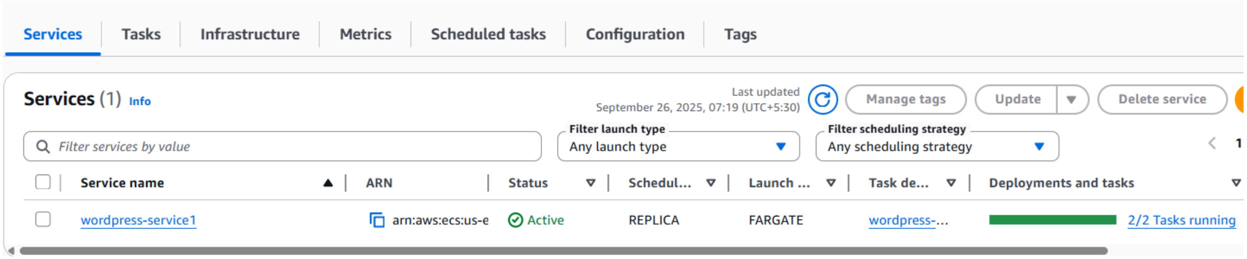
Status
ACTIVE

Task role
-

Task execution role
ecsTaskExecutionRole

Fault injection
Turned off

ECS Service Creation



Services (1) Info

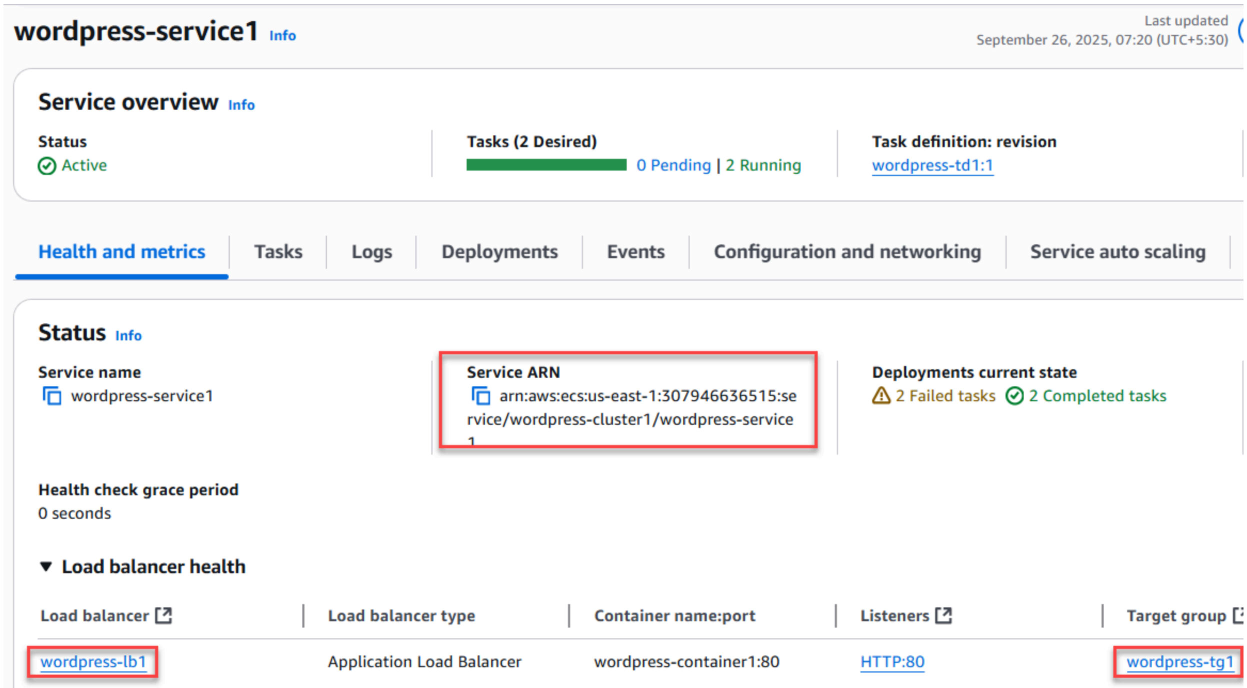
Last updated
September 26, 2025, 07:19 (UTC+5:30)

Filter services by value

Filter launch type: Any launch type

Filter scheduling strategy: Any scheduling strategy

□	Service name	ARN	Status	Schedul...	Launch ...	Task de...	Deployments and tasks
□	wordpress-service1	arn:aws:ecs:us-e	Active	REPLICA	FARGATE	wordpress-...	2/2 Tasks running



wordpress-service1 Info

Last updated
September 26, 2025, 07:20 (UTC+5:30)

Service overview Info

Status	Tasks (2 Desired)	Task definition: revision
✓ Active	0 Pending | 2 Running	wordpress-td1:1

Status Info

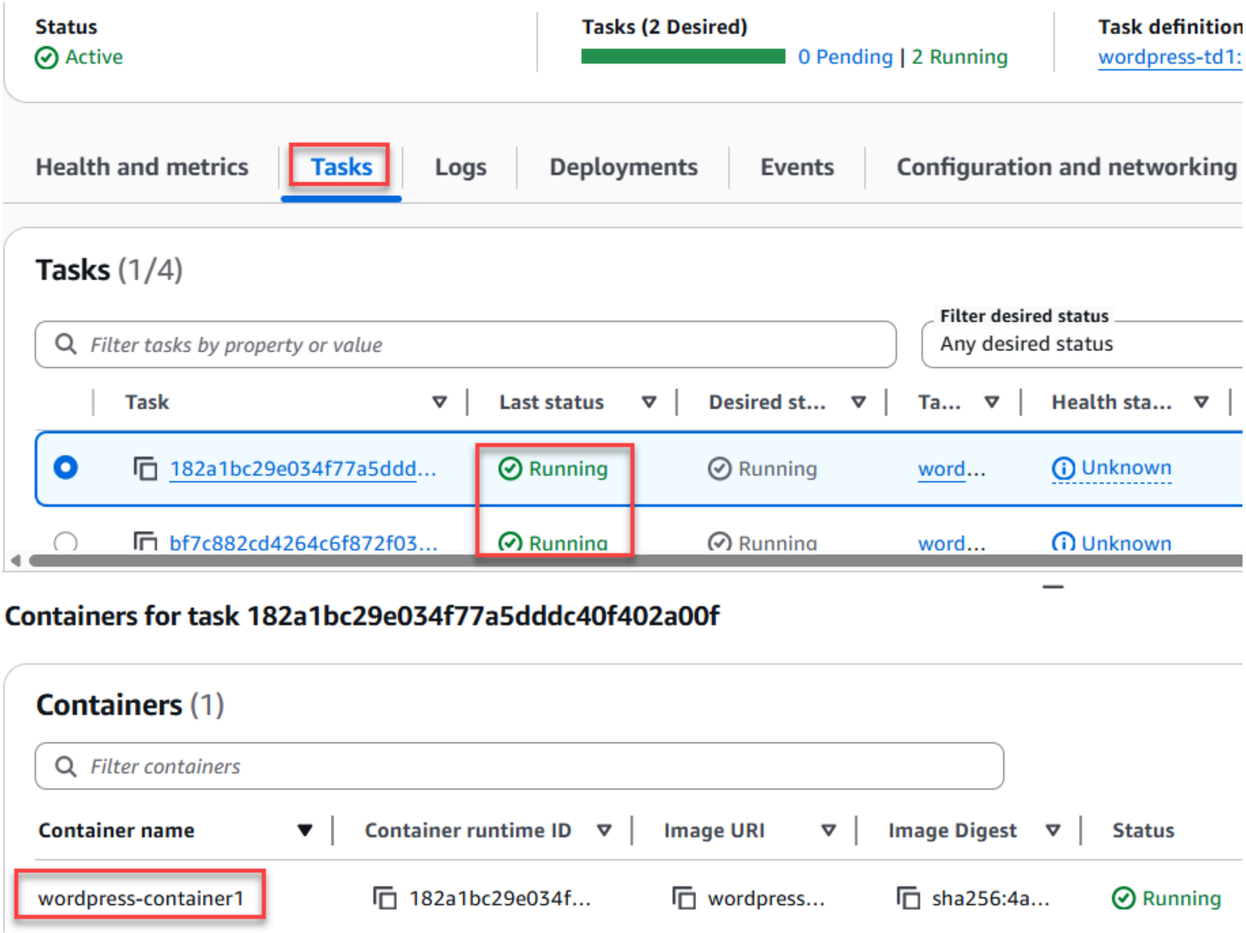
Service name	Service ARN	Deployments current state
wordpress-service1	arn:aws:ecs:us-east-1:307946636515:service/wordpress-cluster1/wordpress-service1	⚠ 2 Failed tasks ✓ 2 Completed tasks

Health check grace period
0 seconds

▼ **Load balancer health**

Load balancer	Load balancer type	Container name:port	Listeners	Target group
wordpress-lb1	Application Load Balancer	wordpress-container1:80	HTTP:80	wordpress-tg1

Task Details



Status

✓ Active

Tasks (2 Desired)

🟩 0 Pending | 2 Running

Task definition

wordpress-td1:

Tasks (1/4)

Filter tasks by property or value

Filter desired status: Any desired status

	Task	Last status	Desired st...	Ta...	Health sta...
●	182a1bc29e034f77a5ddd...	✓ Running	✓ Running	word...	ⓘ Unknown
○	bf7c882cd4264c6f872f03...	✓ Running	✓ Running	word...	ⓘ Unknown

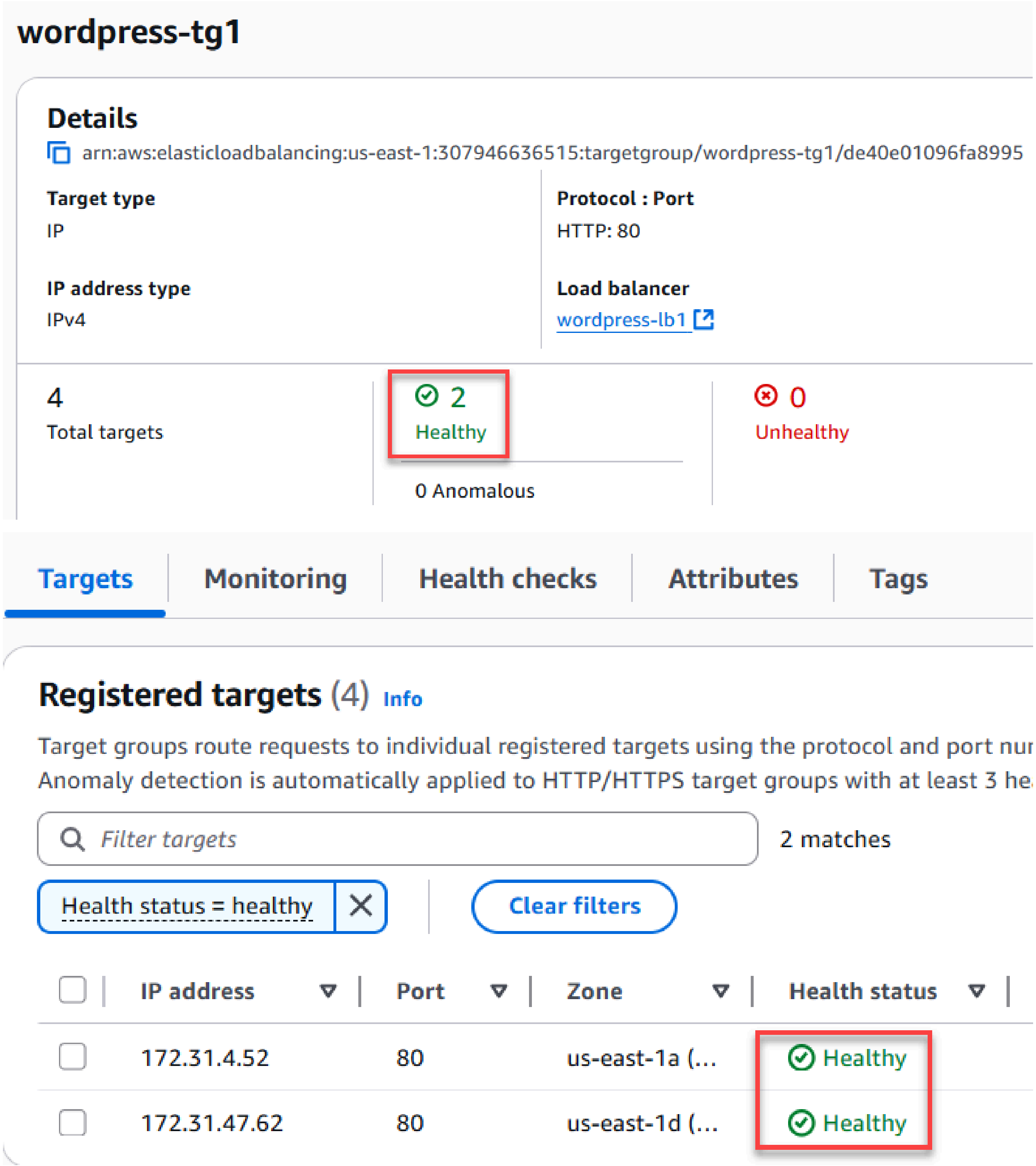
Containers for task 182a1bc29e034f77a5dddc40f402a00f

Containers (1)

Filter containers

Container name	Container runtime ID	Image URI	Image Digest	Status
wordpress-container1	182a1bc29e034f...	wordpress...	sha256:4a...	✓ Running

Target Group



wordpress-tg1

Details

arn:aws:elasticloadbalancing:us-east-1:307946636515:targetgroup/wordpress-tg1/de40e01096fa8995

Target type IP	**Protocol : Port** HTTP: 80
IP address type IPv4	**Load balancer** wordpress-lb1

4 Total targets

✓ 2 Healthy

0 Anomalous

⊗ 0 Unhealthy

Registered targets (4) Info

Target groups route requests to individual registered targets using the protocol and port nur
Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 he

Filter targets

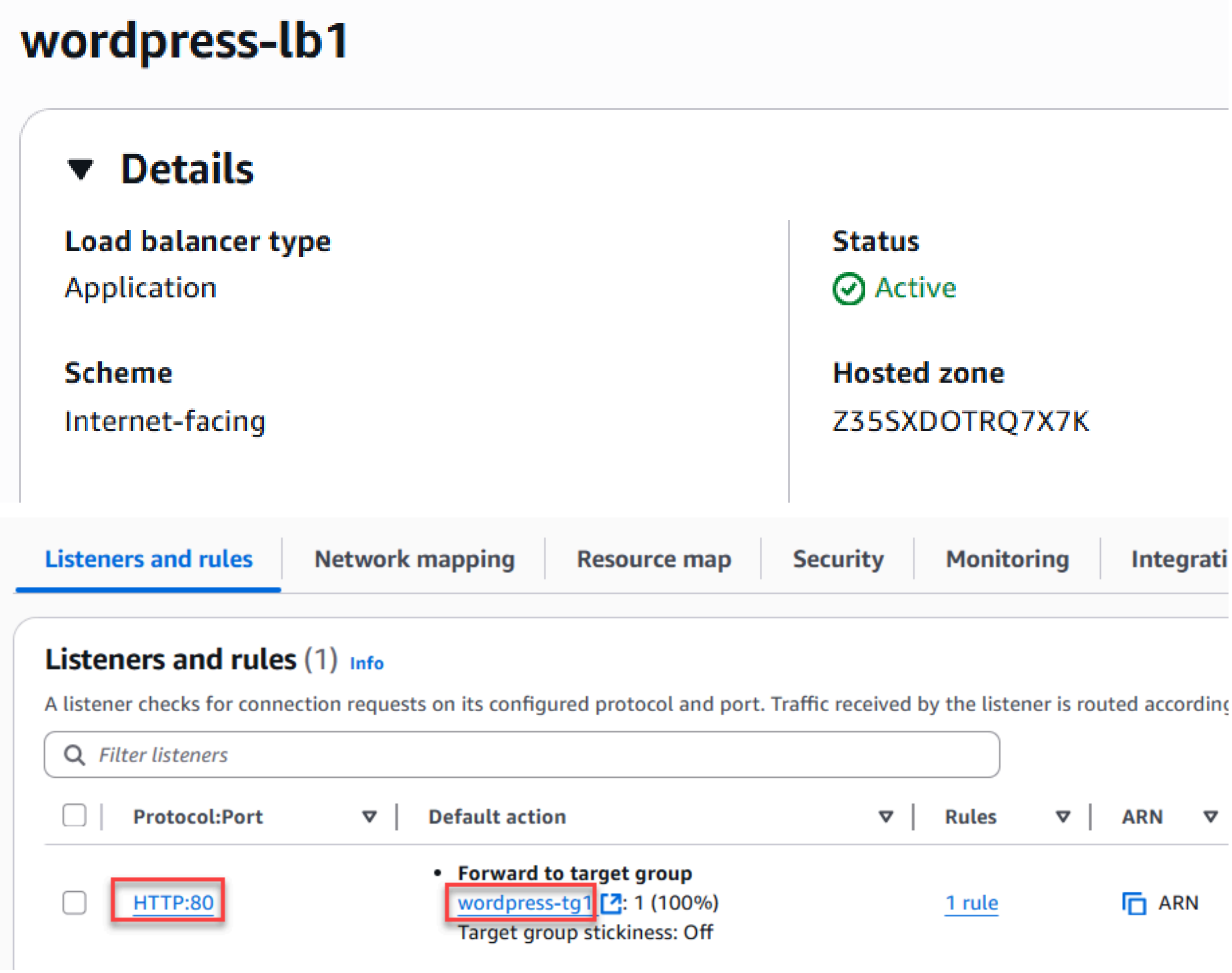
2 matches

Health status = healthy ✕

Clear filters

□	IP address	Port	Zone	Health status
□	172.31.4.52	80	us-east-1a (...	✓ Healthy
□	172.31.47.62	80	us-east-1d (...	✓ Healthy

Load Balancer



wordpress-lb1

Details

Load balancer type
Application

Scheme
Internet-facing

Status
Active

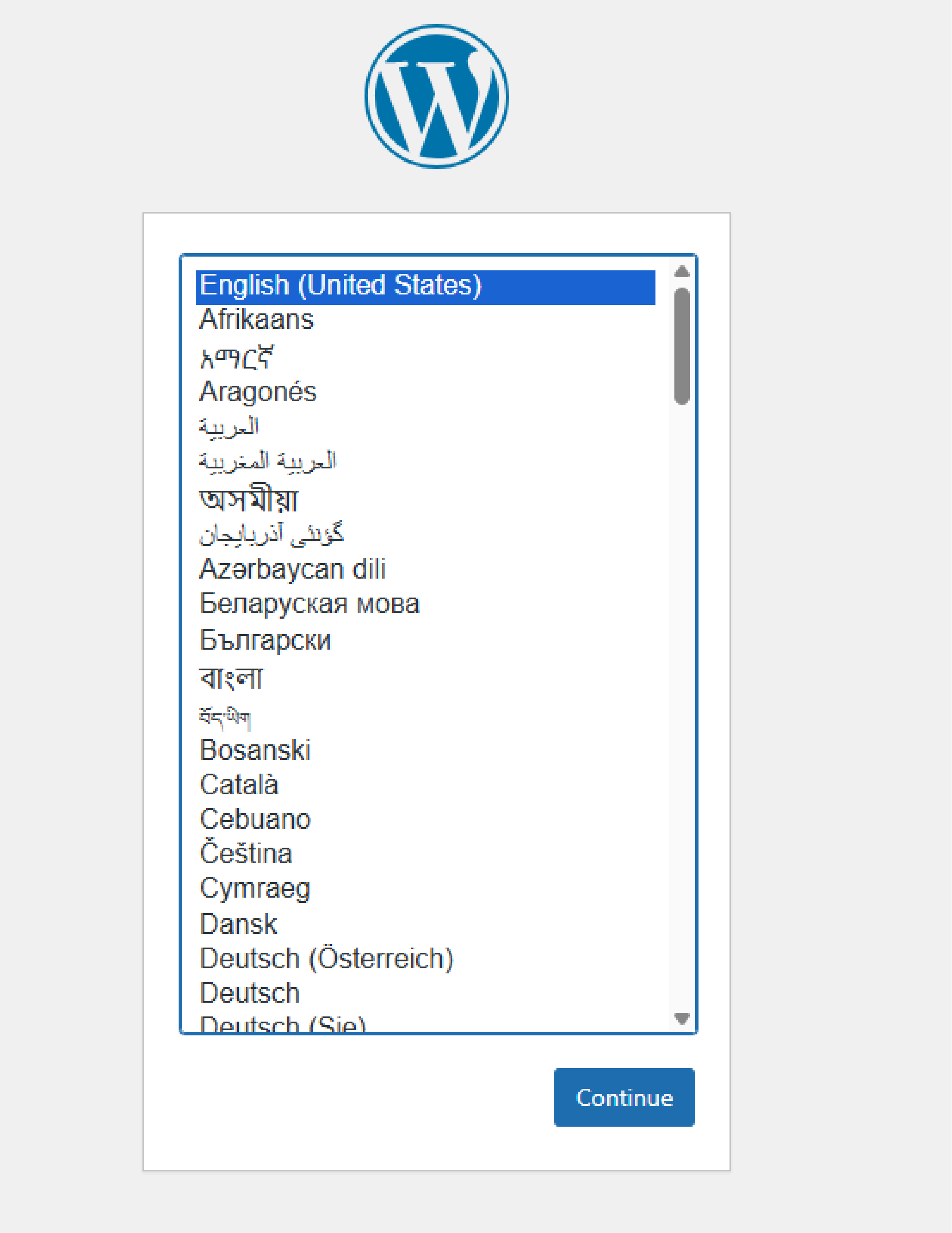
Hosted zone
Z35SXDOTRQ7X7K

Listeners and rules (1) Info

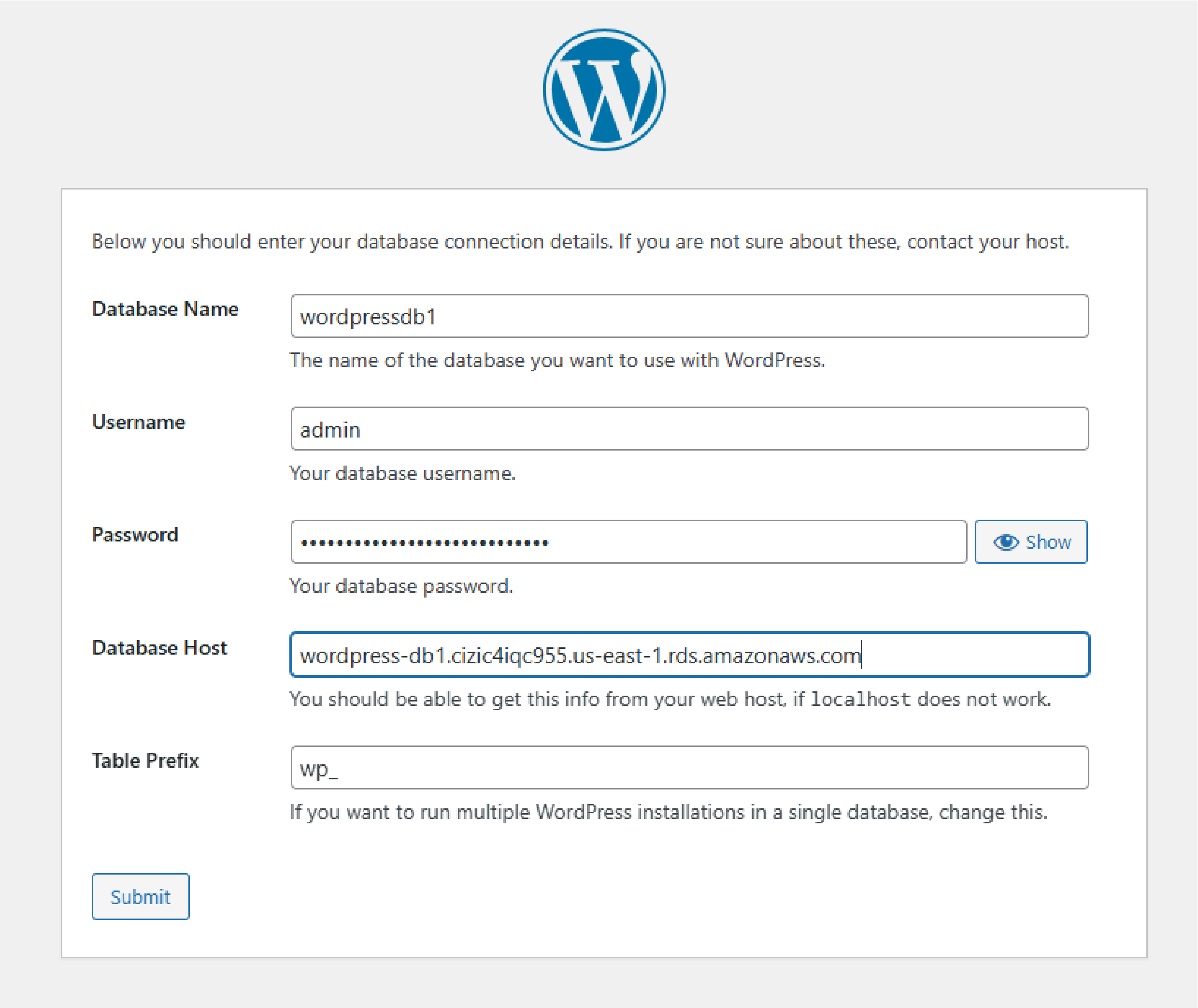
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed accordinç

□	Protocol:Port	Default action	Rules	ARN
□	HTTP:80	- **Forward to target group** wordpress-tg1: 1 (100%) Target group stickiness: Off	1 rule	ARN

wordpress-lb1-603091662.us-east-1.elb.amazonaws.com

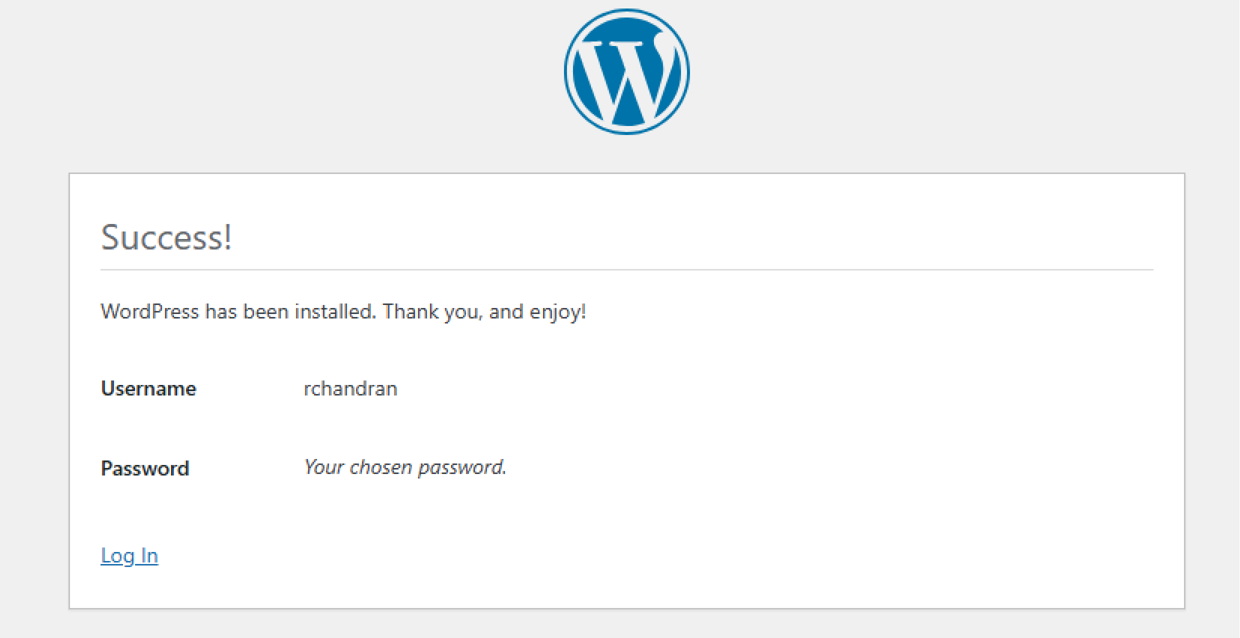


Wordpress Conection Details



Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name	wordpressdb1 The name of the database you want to use with WordPress.
Username	admin Your database username.
Password	••••••••••••••••••••••••••• Your database password.
Database Host	wordpress-db1.cizic4iqc955.us-east-1.rds.amazonaws.com You should be able to get this info from your web host, if `localhost` does not work.
Table Prefix	wp_ If you want to run multiple WordPress installations in a single database, change this.



Success!

WordPress has been installed. Thank you, and enjoy!

Username rchandran

Password *Your chosen password.*

Log In

Brief Report

1. What I learned

This assignment gave me hands-on experience with deploying a containerized application (WordPress) on **Amazon ECS (Elastic Container Service)** with **Fargate launch type**. I learned how different AWS services integrate to host a scalable and fault-tolerant WordPress site:

- **ECS Cluster and Service** – I understood how ECS tasks and services work together to run and manage containers.
- **RDS MySQL Database** – I deployed a managed database service and connected it to WordPress using secure credentials.
- **Application Load Balancer (ALB)** – I configured an ALB with a target group to distribute incoming traffic across ECS tasks.
- **Security Groups and Networking** – I gained practical knowledge about controlling traffic between ECS tasks, RDS, and the load balancer using inbound and outbound rules.

- **Custom Domain Setup** – I mapped my domain to the ALB DNS name using Route 53, which provided a professional endpoint for accessing WordPress.

- **Health Checks** – I learned how ECS deployment circuit breakers and ALB health checks work, and why proper health check configuration is critical for container deployments.

Overall, this assignment strengthened my understanding of **container orchestration, service discovery, and AWS networking**.

2. Any problems I faced

During the deployment, I encountered a few challenges:

1. **ECS Deployment Circuit Breaker Triggered**

    - Initially, the ECS service kept failing because the ALB marked my tasks as unhealthy. The issue was that the ALB health check path (`/`) returned a **302 redirect** instead of 200.

    - **Solution**: I updated the Target Group health check path to `/wp-login.php` and allowed success codes `200–399`. After this, the tasks registered as healthy.

2. **Database Connection Issues**

    - WordPress containers failed to connect to RDS due to missing or incorrect environment variables.

    - **Solution**: I provided the correct `WORDPRESS_DB_HOST`, `WORDPRESS_DB_USER`, `WORDPRESS_DB_PASSWORD`, and `WORDPRESS_DB_NAME` values in the ECS task definition. I also ensured that the RDS security group allowed inbound traffic on port 3306 from the ECS task security group.

3. **Networking and Access**

    - At first, my RDS was not publicly accessible, so ECS tasks couldn't connect.

    - **Solution**: I confirmed that both ECS and RDS were deployed in the same VPC and subnet configuration.

These troubleshooting steps helped me understand how different AWS components depend on each other and how to debug real-world deployment issues.

3. My custom domain name

For this assignment, I mapped my **custom domain**:

```
https://wordpress.rajeshapps.site
```

(This was configured in Route 53 to point to the ALB DNS name. I bought this domain name “rajeshapps.site” from godaddy.com)

4. ALB DNS name for comparison

The Application Load Balancer also provides a default AWS DNS name, which can be used to access the WordPress site without a custom domain:

```
http://wordpress-lb1-603091662.us-east-1.elb.amazonaws.com
```

By comparing both, I was able to verify that the custom domain correctly resolves to the same ECS-hosted WordPress site.

Conclusion

This assignment gave me valuable experience in deploying a production-style WordPress application on ECS. I not only practiced container orchestration but also learned how to integrate ECS with RDS, ALB, and Route 53 while solving real troubleshooting issues. These skills will directly help me in designing and managing cloud-native applications in AWS.