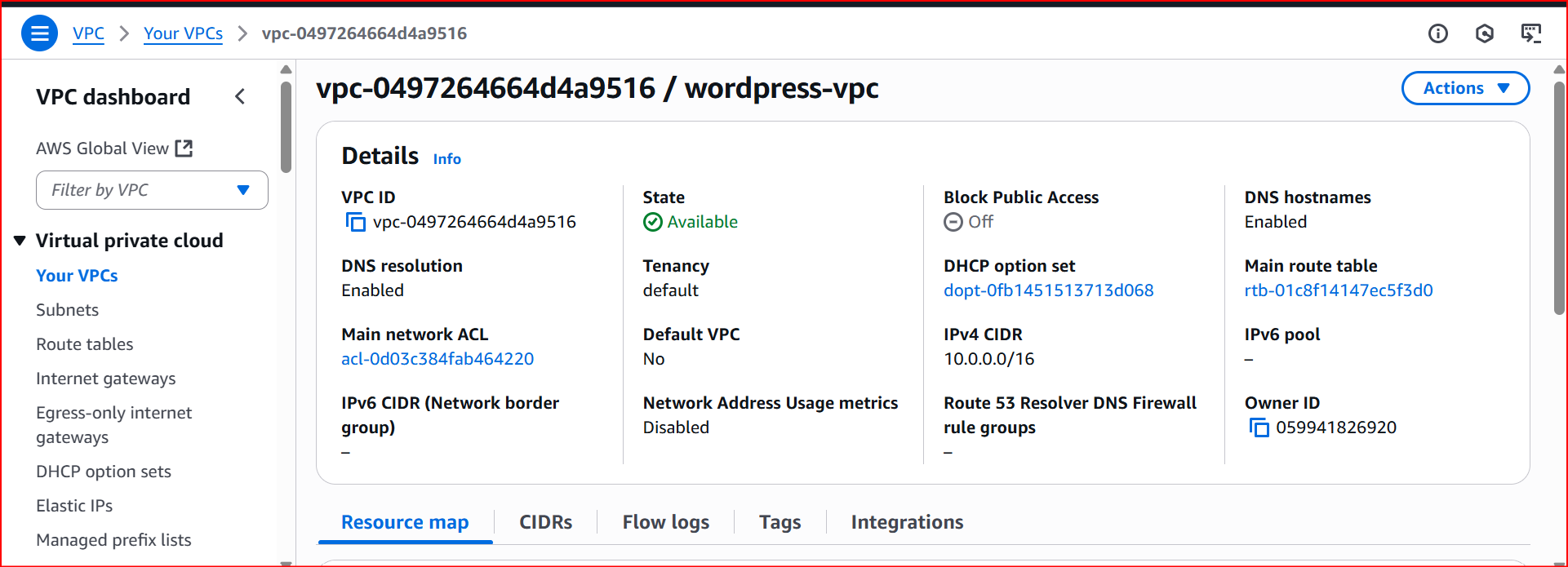
**Architecture Overview**

* **ECS Fargate**: Serverless container hosting for WordPress
* **RDS MySQL**: Managed database service
* **ALB**: Application Load Balancer for traffic distribution
* **VPC**: Virtual Private Cloud for network isolation
* **ECR**: Elastic Container Registry for WordPress image

**Step 1: Create VPC and Networking**

**1.1 Create VPC**

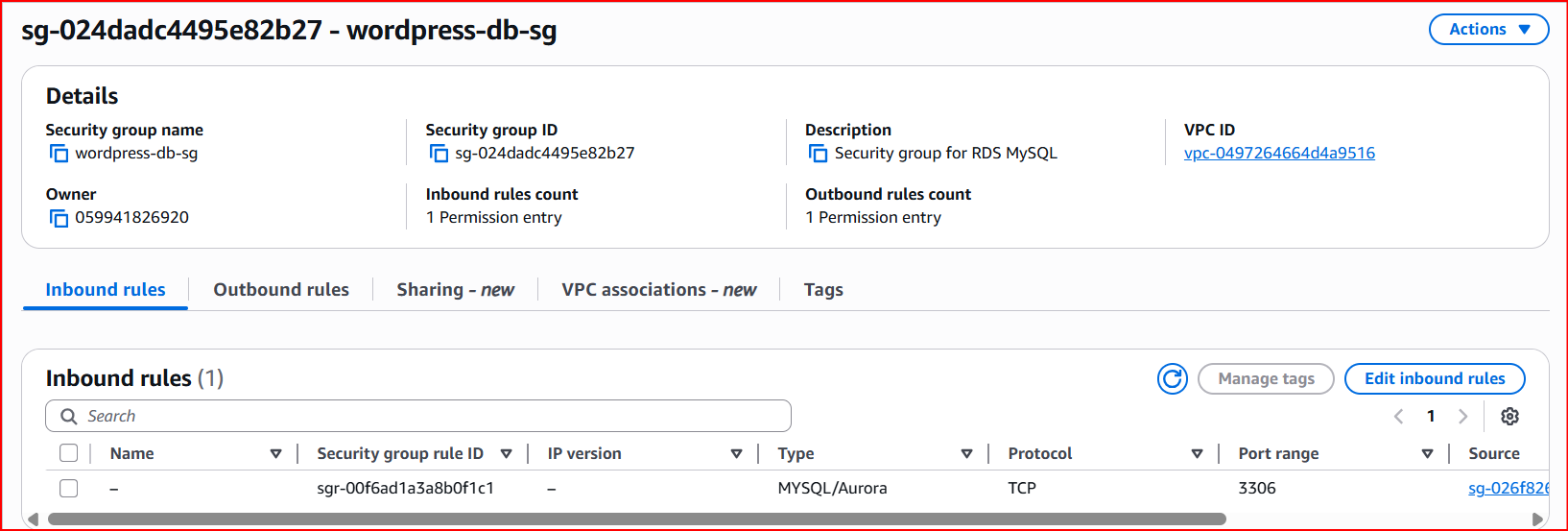
1. Go to **VPC Console** → **Create VPC**
2. Choose **VPC and more**
3. Configure:
   * **Name**: wordpress-vpc
   * **IPv4 CIDR**: 10.0.0.0/16
   * **Availability Zones**: 2
   * **Public subnets**: 2
   * **Private subnets**: 2
   * **NAT gateways**: 1 per AZ
   * **VPC endpoints**: None
4. Click **Create VPC**



**1.2 Create Security Groups**

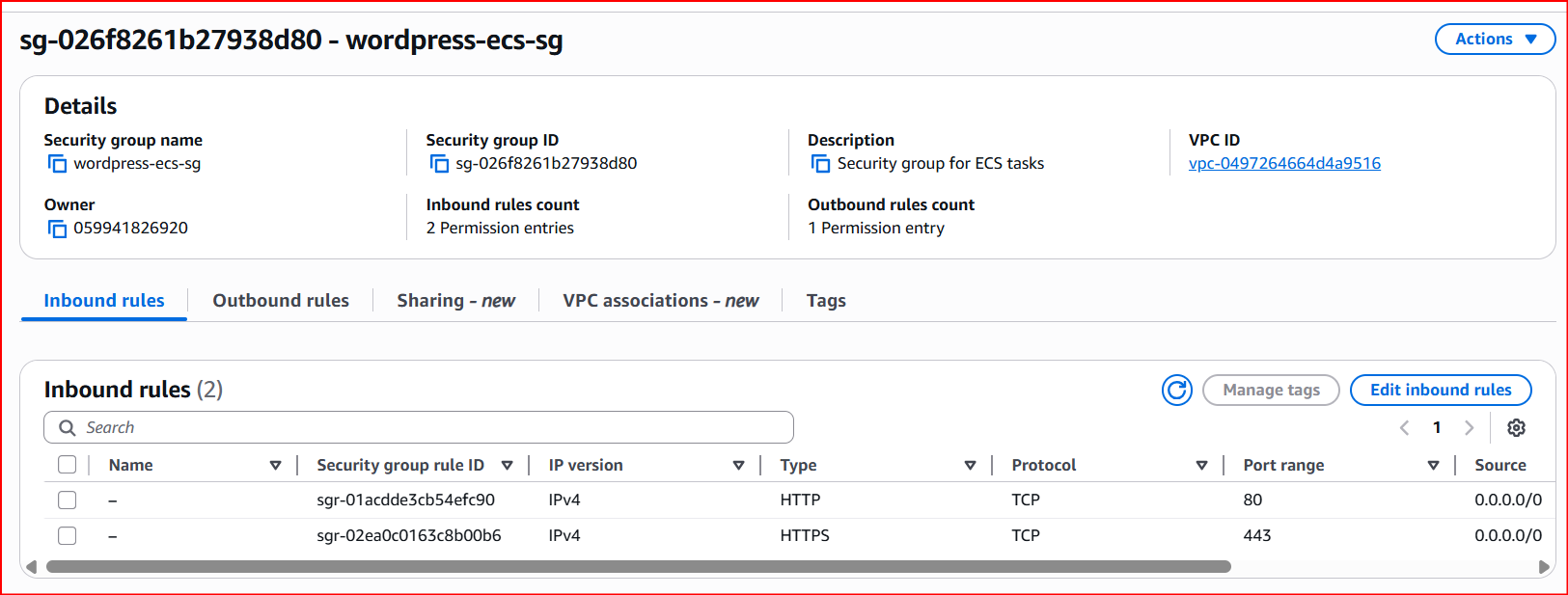
**Database Security Group**

1. Go to **EC2 Console** → **Security Groups** → **Create Security Group**
2. Configure:
   * **Name**: wordpress-db-sg
   * **Description**: Security group for RDS MySQL
   * **VPC**: Select wordpress-vpc
3. **Inbound Rules**:
   * Type: MySQL/Aurora (3306)
   * Source: Custom (select ECS security group - we'll create this next)
4. Click **Create Security Group**



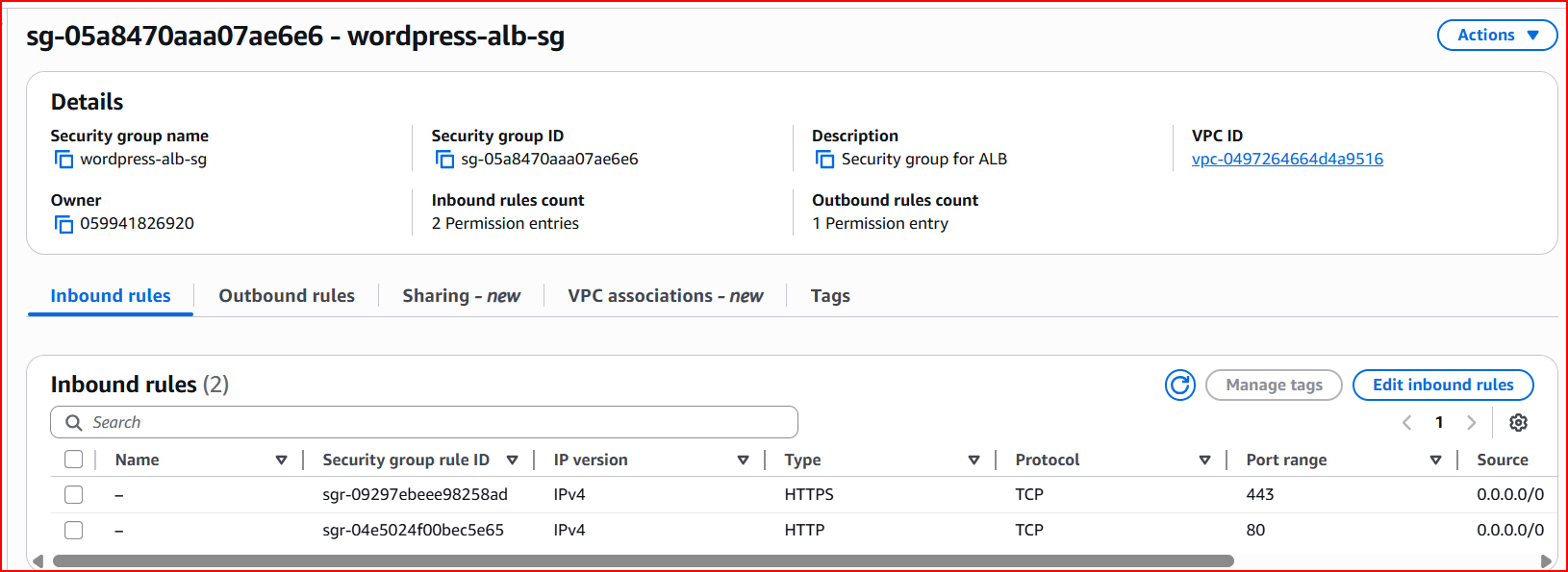
**ECS Security Group**

1. Create another security group:
   * **Name**: wordpress-ecs-sg
   * **Description**: Security group for ECS tasks
   * **VPC**: Select wordpress-vpc
2. **Inbound Rules**:
   * Type: HTTP (80), Source: Custom (select ALB security group - we will create this next)
   * Type: HTTPS (443), Source: Custom (select ALB security group - we will create this next)
3. Click **Create Security Group**



**Load Balancer Security Group**

1. Create another security group:
   * **Name**: wordpress-alb-sg
   * **Description**: Security group for ALB
   * **VPC**: Select wordpress-vpc
2. **Inbound Rules**:
   * Type: HTTP (80), Source: 0.0.0.0/0
   * Type: HTTPS (443), Source: 0.0.0.0/0
3. Click **Create Security Group**



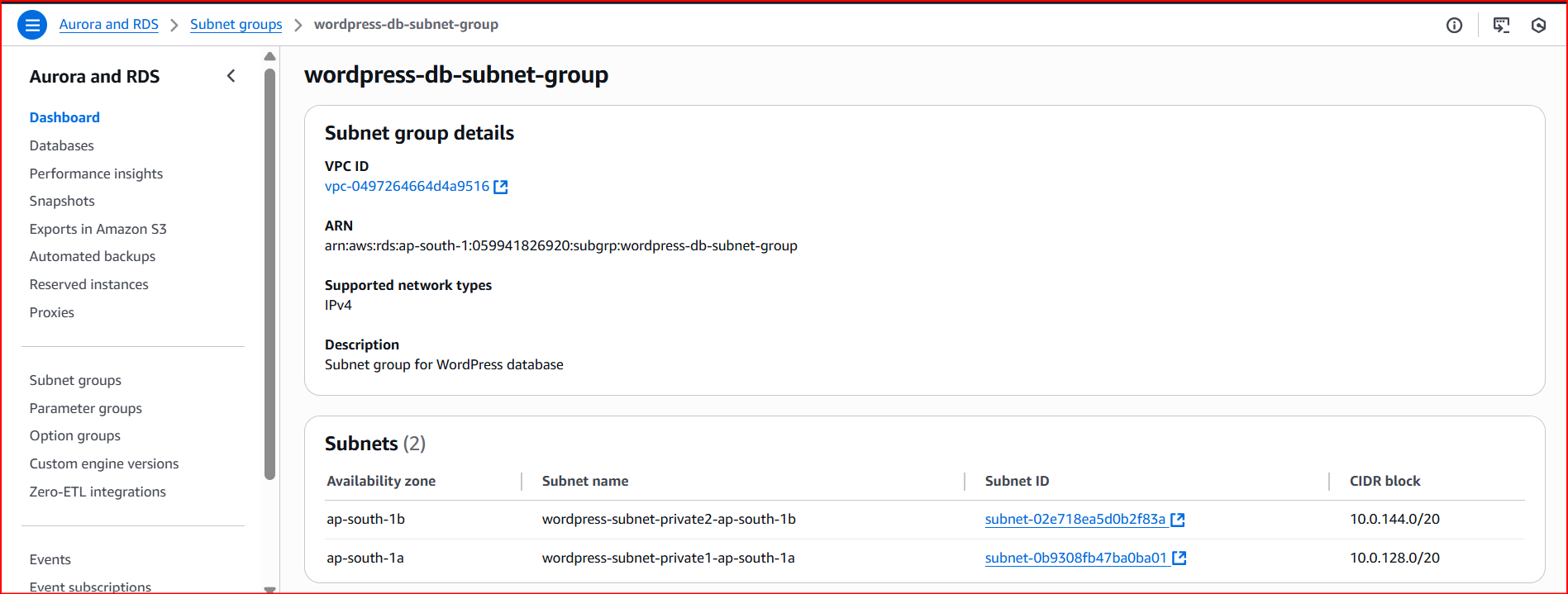
**1.3 Update Database and ECS Security Group**

1. Go back to wordpress-db-sg
2. Edit inbound rules and change the source to wordpress-ecs-sg
3. Go back to wordpress-ecs-sg
4. Edit inbound rules and change the source to wordpress-alb-sg

**Step 2: Create RDS MySQL Database**

**2.1 Create DB Subnet Group**

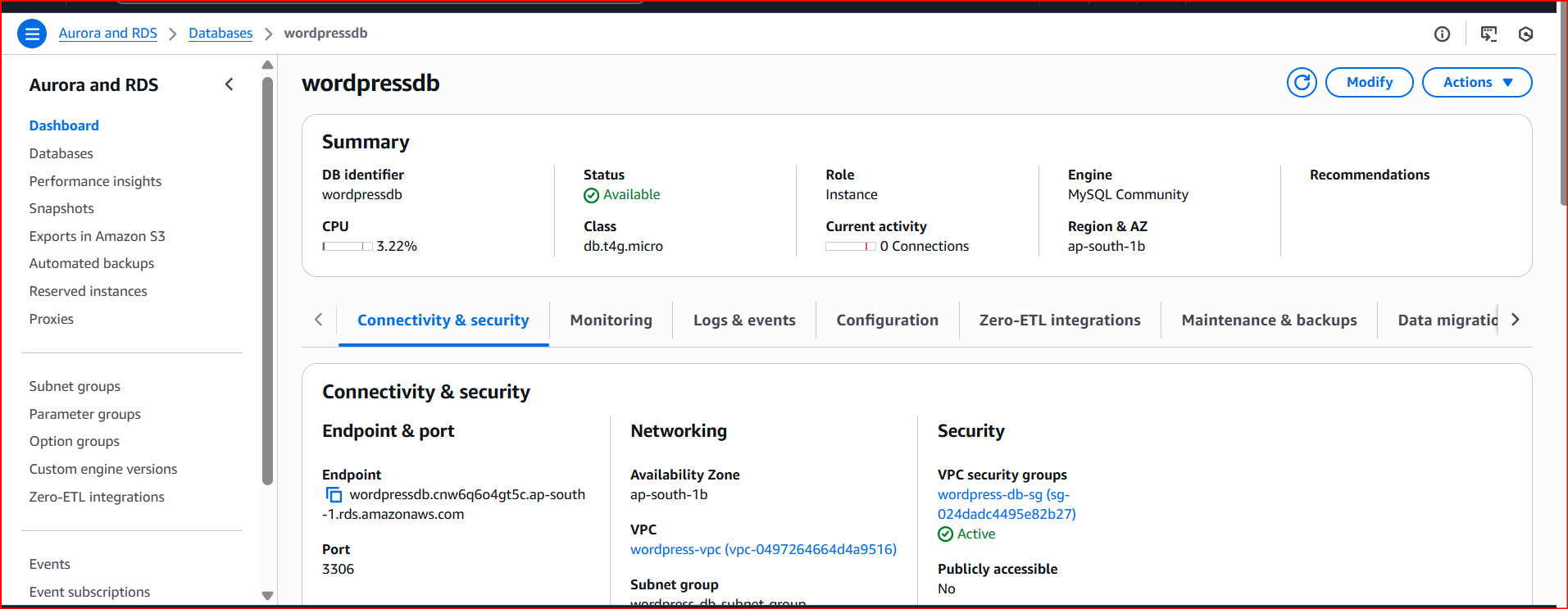
1. Go to **RDS Console** → **Subnet Groups** → **Create DB Subnet Group**
2. Configure:
   * **Name**: wordpress-db-subnet-group
   * **Description**: Subnet group for WordPress database
   * **VPC**: Select wordpress-vpc
   * **Availability Zones**: Select both AZs
   * **Subnets**: Select both private subnets
3. Click **Create**



**2.2 Create RDS Instance**

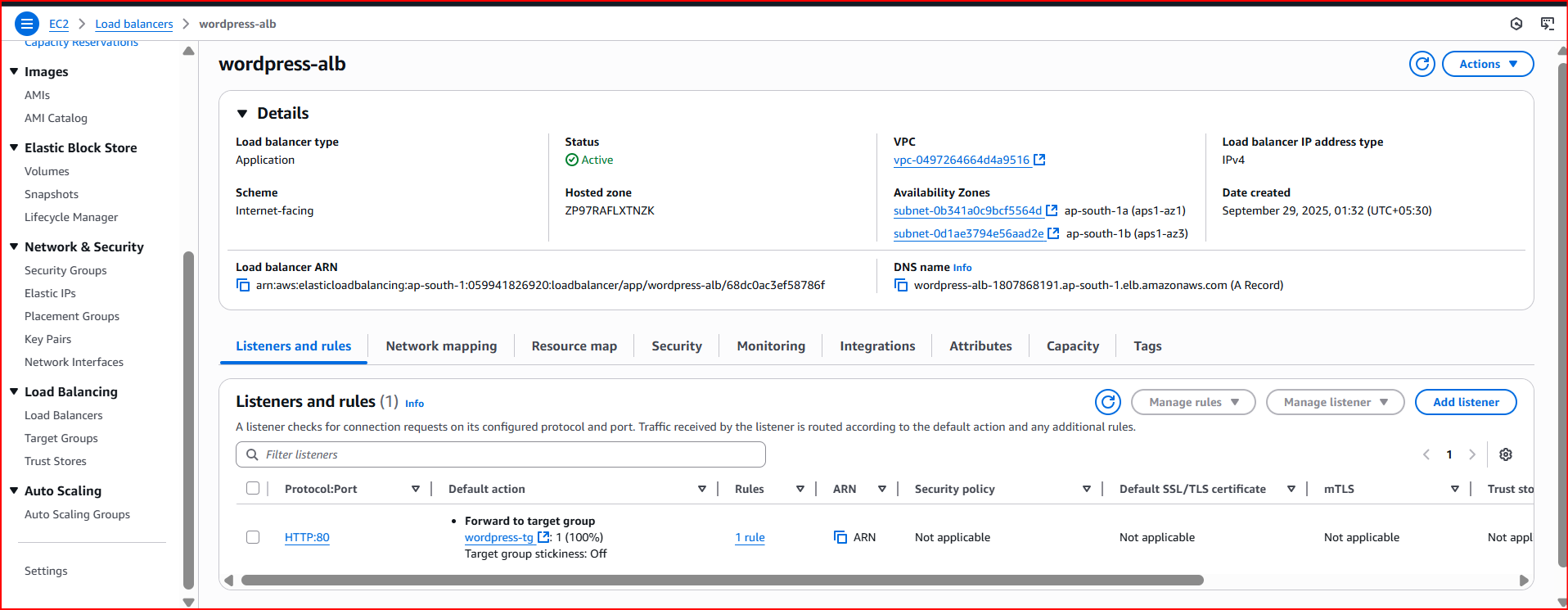
1. Go to **RDS Console** → **Create Database**
2. **Engine Options**:
   * **Engine**: MySQL
   * **Version**: Latest (e.g., 8.0.35)
   * **Template**: Free tier (or Production for production use)
3. **Settings**:
   * **DB Instance Identifier**: wordpress-db
   * **Master Username**: admin
   * **Master Password**: Create a secure password (save this!)
4. **DB Instance Class**:
   * Free tier: db.t3.micro
   * Production: db.t3.small or larger
5. **Storage**:
   * Storage Type: General Purpose SSD (gp2)
   * Allocated Storage: 20 GB (minimum)
6. **Connectivity**:
   * **VPC**: wordpress-vpc
   * **DB Subnet Group**: wordpress-db-subnet-group
   * **VPC Security Groups**: wordpress-db-sg
   * **Availability Zone**: No preference
   * **Public Access**: No
7. **Additional Configuration**:
   * **Initial Database Name**: wordpress
   * **Backup Retention**: 7 days
   * **Monitoring**: Enable Enhanced Monitoring (optional)
8. Click **Create Database**

**Note**: Database creation takes 10-15 minutes. Note down the endpoint when ready.



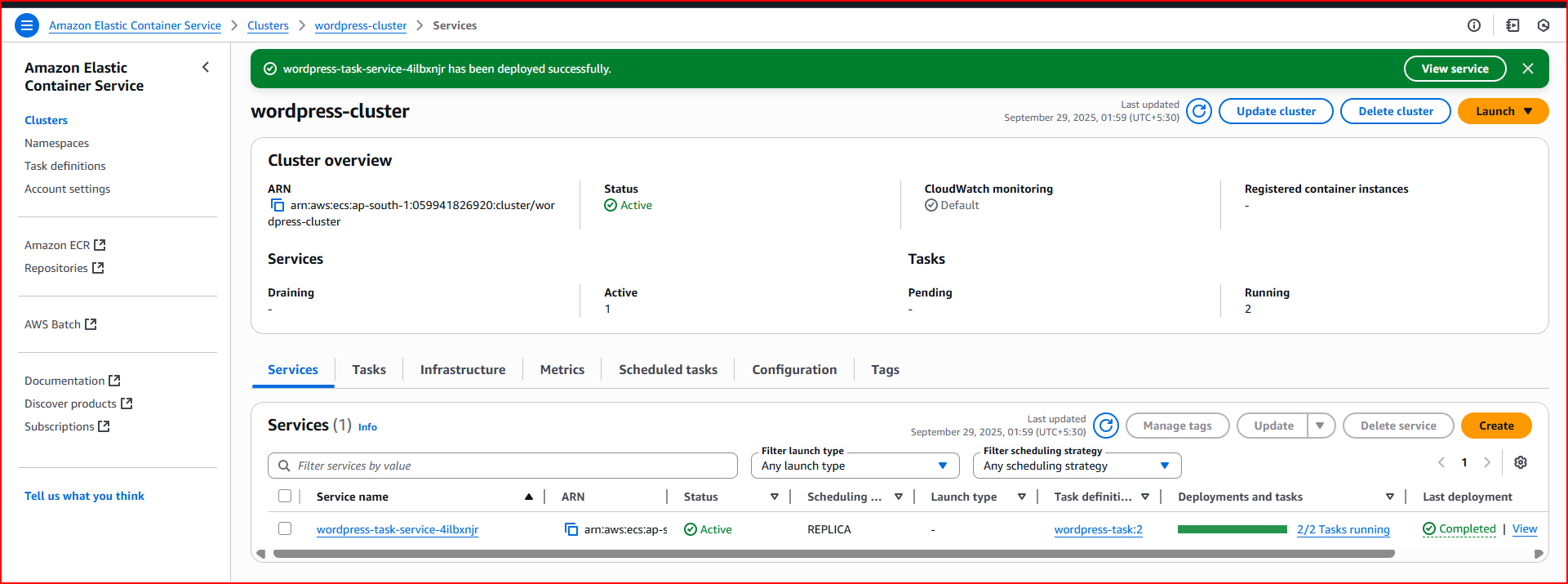
**Step 3: Create Application Load Balancer**

1. Go to **EC2 Console** → **Load Balancers** → **Create Load Balancer**
2. Choose **Application Load Balancer**
3. **Basic Configuration**:
   * **Name**: wordpress-alb
   * **Scheme**: Internet-facing
   * **IP Address Type**: IPv4
4. **Network Mapping**:
   * **VPC**: wordpress-vpc
   * **Mappings**: Select both public subnets
5. **Security Groups**: Select wordpress-alb-sg
6. **Listeners and Routing**:
   * **Protocol**: HTTP, **Port**: 80
   * **Default Action**: Create new target group
     + **Target Group Name**: wordpress-tg
     + **Target Type**: IP
     + **Protocol**: HTTP, **Port**: 80
     + **VPC**: wordpress-vpc
     + **Health Check Path**: /
7. Click **Create Load Balancer**



**Step 4: Create ECS Cluster**

1. Go to **ECS Console** → **Clusters** → **Create Cluster**
2. **Cluster Configuration**:
   * **Cluster Name**: wordpress-cluster
   * **Infrastructure**: AWS Fargate (serverless)
3. **Monitoring**: Enable Container Insights (optional)
4. Click **Create**



**Step 5: Create Task Definition**

**5.1 Create Task Definition**

1. Go to **ECS Console** → **Task Definitions** → **Create New Task Definition**
2. **Task Definition Configuration**:
   * **Task Definition Family**: wordpress-task
   * **Launch Type**: AWS Fargate
   * **Operating System**: Linux/X86\_64
   * **CPU**: 0.5 vCPU (512)
   * **Memory**: 1 GB (1024)
   * **Task Role**: ecsTaskExecutionRole (create if doesn't exist)
   * **Task Execution Role**: ecsTaskExecutionRole

**5.2 Configure Container**

1. **Container Details**:
   * **Name**: wordpress
   * **Image URI**: wordpress:latest
   * **Port Mappings**:
     + Container Port: 80
     + Protocol: TCP
     + Port Name: wordpress-80-tcp
2. **Environment Variables**:

WORDPRESS\_DB\_HOST=<RDS\_ENDPOINT>:3306

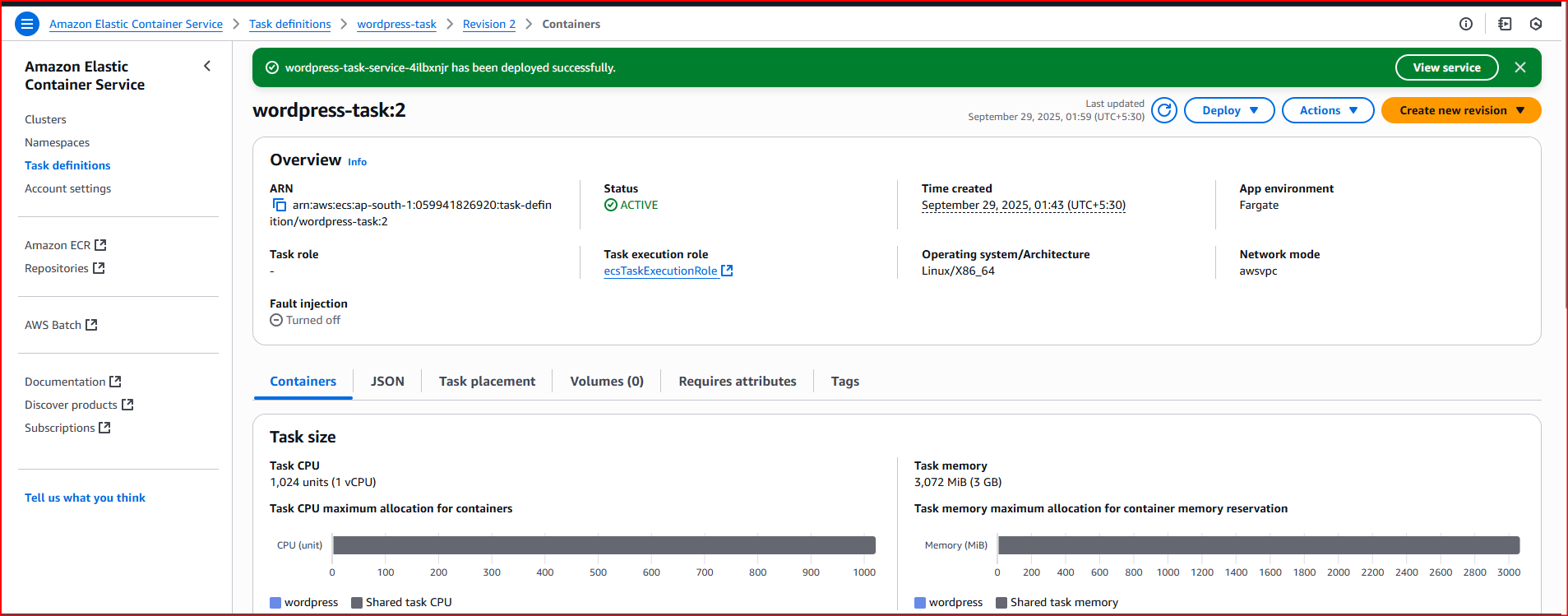
WORDPRESS\_DB\_USER=admin

WORDPRESS\_DB\_PASSWORD=<YOUR\_DB\_PASSWORD>

WORDPRESS\_DB\_NAME=wordpress

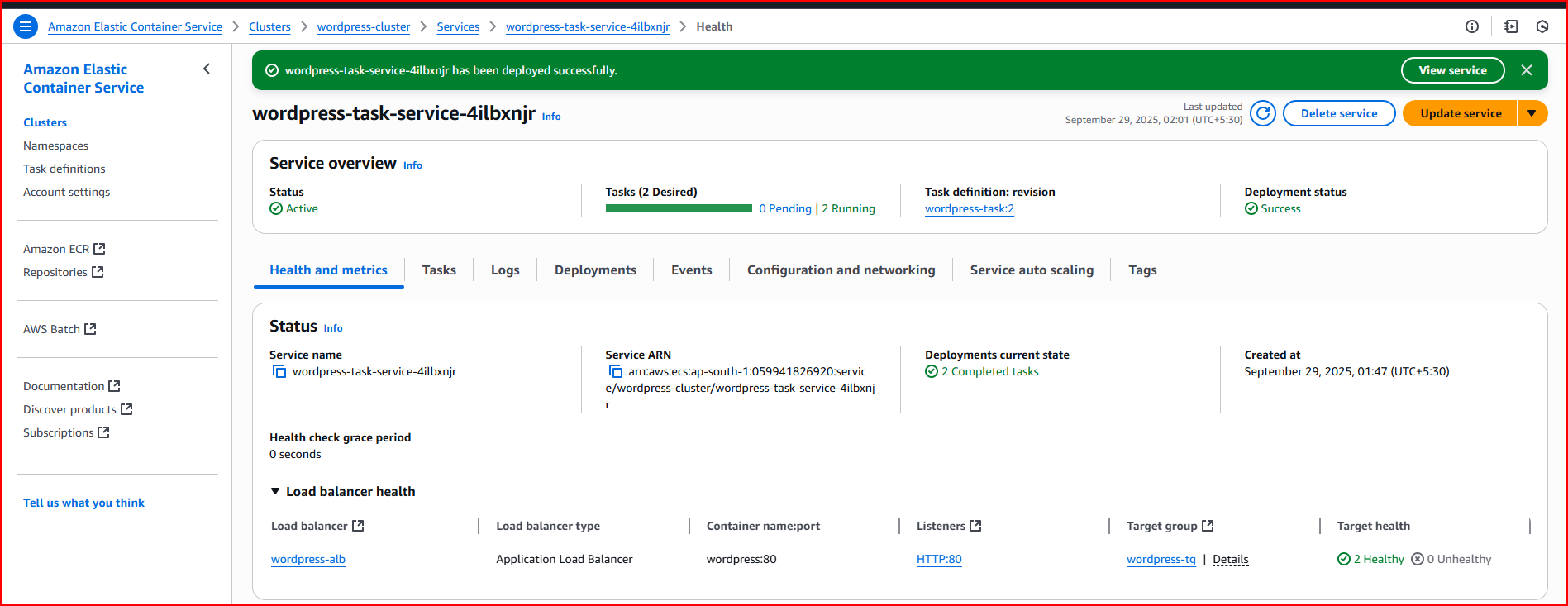
Replace <RDS\_ENDPOINT> with your actual RDS endpoint and <YOUR\_DB\_PASSWORD> with your database password.

1. **Logging** (Optional):
   * Enable **Use log collection**
   * **Log Driver**: awslogs
   * **Log Group**: Create new: /ecs/wordpress-task
2. Click **Create**

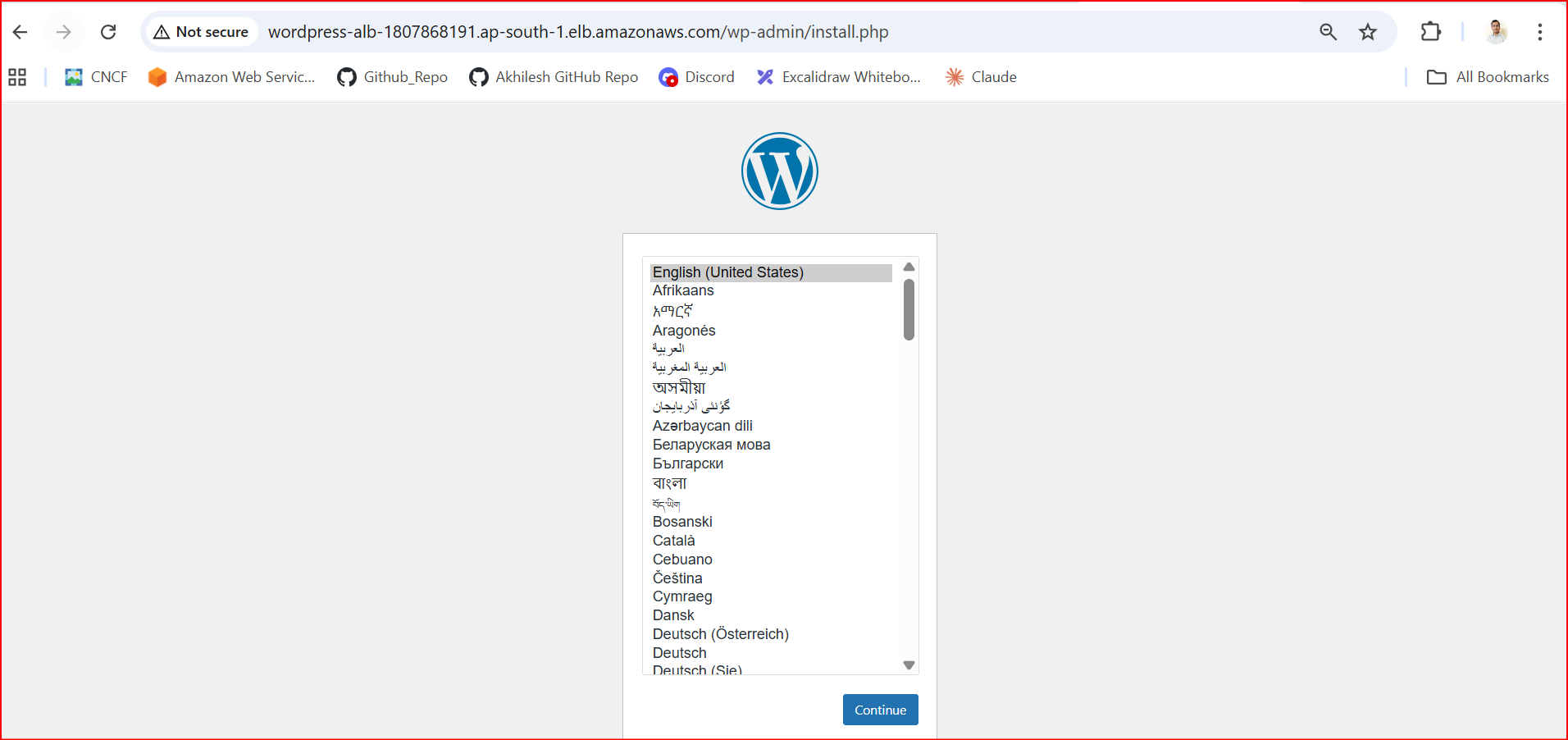


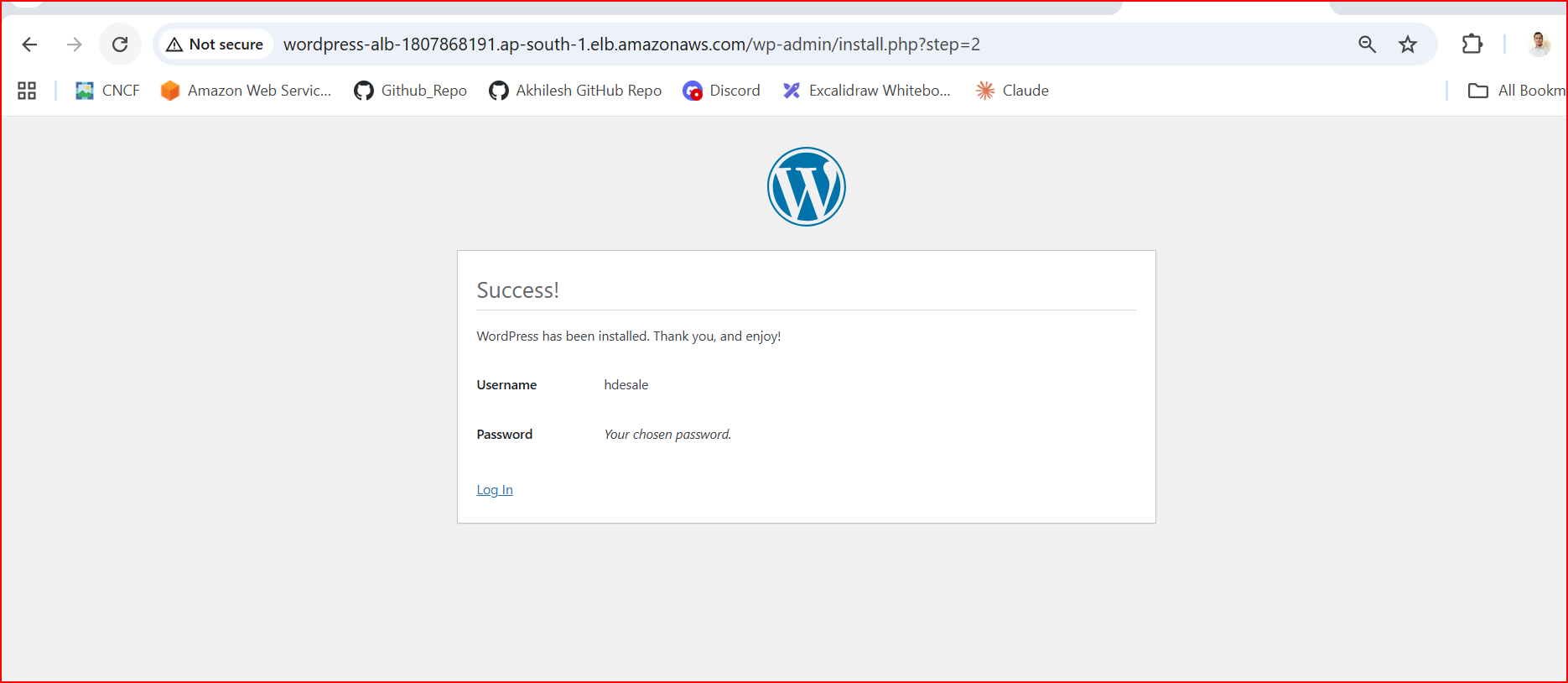
**Step 6: Create ECS Service**

1. Go to **ECS Console** → **Clusters** → Select wordpress-cluster
2. **Services** tab → **Create Service**
3. **Deployment Configuration**:
   * **Application Type**: Service
   * **Task Definition**: Select wordpress-task
   * **Service Name**: wordpress-service
   * **Desired Tasks**: 2
4. **Networking**:
   * **VPC**: wordpress-vpc
   * **Subnets**: Select private subnets
   * **Security Groups**: Select wordpress-ecs-sg
   * **Public IP**: Enabled (required for Fargate in private subnets with internet access)
5. **Load Balancing**:
   * **Load Balancer Type**: Application Load Balancer
   * **Load Balancer**: Select wordpress-alb
   * **Target Group**: Select wordpress-tg
   * **Container**: wordpress 80:80
6. Click **Create Service**



**Step 7: Access WordPress using ALB DNS**





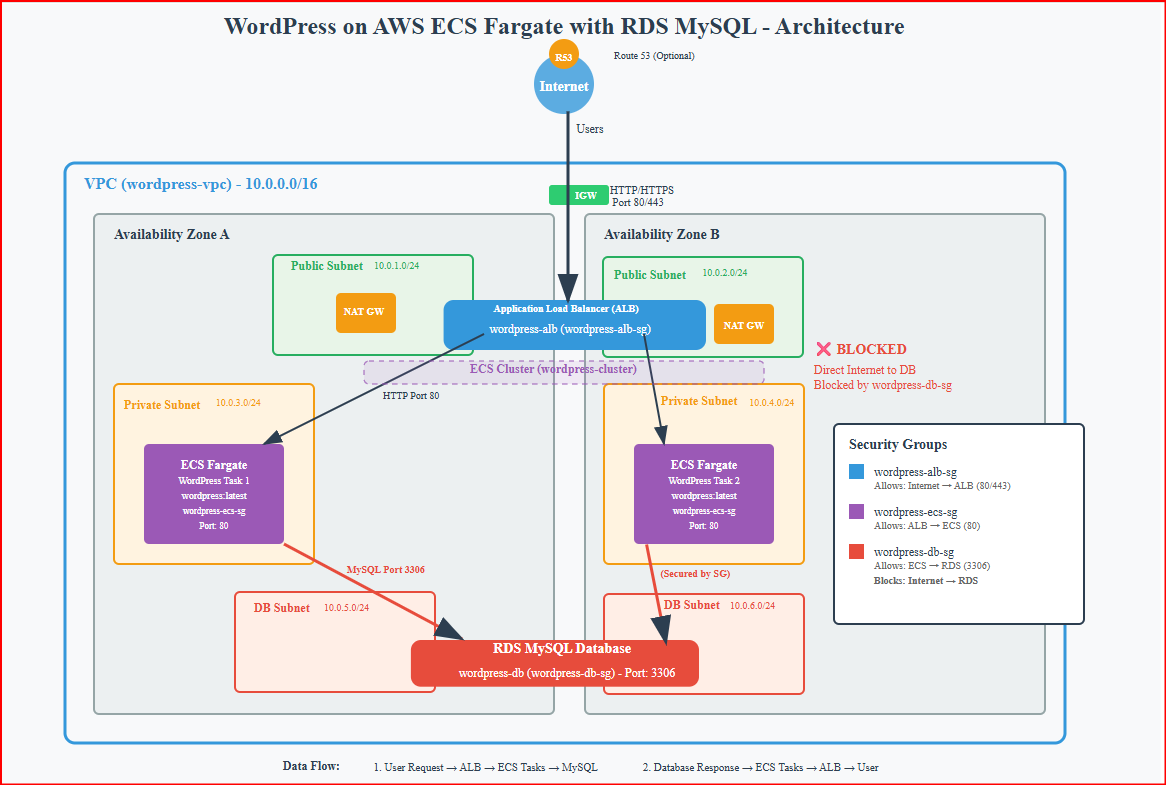
**The Security Flow :**

**Internet → Load Balancer → WordPress Containers → Database**

**✓ Allowed ✓ Allowed ✓ Allowed**

**Internet → Database (Direct)**

**❌ BLOCKED by wordpress-db-sg**

****