

INFRA AWS NAVA-LOG

- VPC (com subnets públicas e privadas)
 - Security Groups (para controle de acesso)
 - Load Balancer (para balancear o tráfego)
 - ECS Cluster + Service + Task Definition
 - Amazon RDS (PostgreSQL)
 - MongoDB Atlas
 - WireMock
 - Amazon ECR (para armazenar a imagem da aplicação)
-

Criando a VPC

1. Acesse o **AWS Console**
 2. Vá para **VPC > Criar VPC**
 3. Selecione **VPC e mais**
 4. Configure:
 - **Nome:** nava - log - vpc
 - **Bloco CIDR:** 10.0.0.0/16
 - **Subnets:**
 - **2 subnets públicas** (exemplo: 10.0.1.0/24 e 10.0.2.0/24)
 - **2 subnets privadas** (exemplo: 10.0.3.0/24 e 10.0.4.0/24)
-
- **Public Subnet 1:** 10.0.1.0/24 (us-east-1a)
 - **Public Subnet 2:** 10.0.2.0/24 (us-east-1b)
 - **Private Subnet 1:** 10.0.3.0/24 (us-east-1a)
 - **Private Subnet 2:** 10.0.4.0/24 (us-east-1b)
-
- **Criar NAT Gateway** (para o ECS acessar a internet) e atualize a Route Table
 - Clique em **Criar VPC**

Criando os Security Groups

Criando o Security Group do Load Balancer

1. Vá para **EC2 > Security Groups > Criar grupo de segurança**
2. Nome: **nava-log-alb-sg**
3. Regras de entrada:
 - **HTTP** (80) – Origem: 0.0.0.0/0 (internet)
 - **HTTPS** (443) – Origem: 0.0.0.0/0 (internet)
4. Clique em **Criar**

Criando o Security Group do ECS

1. **Criar novo grupo de segurança**
2. Nome: **nava-log-ecs-sg**
3. Regras de entrada:
 - **HTTP** (8081) – Origem: **nava-log-alb-sg** (Load Balancer)

Saída: Permitir todo o tráfego.

4. Clique em **Criar**

Criando o Security Group do RDS

1. **Criar novo grupo de segurança**
2. Nome: **nava-log-rds-sg**
3. Regras de entrada:
 - **PostgreSQL** (5432) – Origem: **nava-log-rds-sg** (ECS Cluster)

Saída: Permitir todo o tráfego.

4. Clique em **Criar**
-

3. Criar a Base de Dados RDS

1. Vá para **RDS > Create Database**.
2. Escolha **PostgreSQL** e selecione **Standard Create**.
3. Escolha **Single Instance**.

4. Escolha a **VPC** criada (nava - log - vpc) e selecione as subnets *privadas*.
 5. Em **Authentication**, crie um usuário e senha.
 6. Em **Security Group**, escolha nava - log - rds - sg.
 7. Finalize a criação.
-

4. Criar o MongoDB Atlas

1. Acesse o [MongoDB Atlas](#) e crie um cluster gratuito.
 2. Escolha a região **us-east-1** (para manter na mesma região da AWS).
 3. Defina um usuário e senha.
 4. Adicione o IP da VPC da AWS como permitido para conexões.
-

5. Criando o Load Balancer

1. Vá para **EC2 > Load Balancers > Criar Load Balancer**
 2. Escolha **Application Load Balancer (ALB)**
 3. Configure:
 - **Nome:** nava - log - alb
 - **Tipo:** Público
 - **VPC:** nava - log - vpc
 - **Subnets:** Escolha as subnets públicas
 - **Security Group:** nava - log - alb - sg
 4. **Criar Target Group**
 - **Tipo:** IP
 - **Porta:** 8081
 5. **Criar Listeners**
 - **Porta 80** -> Encaminhar para o Target Group criado
 6. Clique em **Criar Load Balancer**
-

6. Criando o ECR (Armazém de Imagens Docker)

1. Vá para **ECR > Criar Repositório**
2. Nome: nava-log-app
3. Escolha **private**.
4. Clique em **Criar**
5. Após criado, copie o **URI do repositório**.

Associe à VPC nava-log-vpc.

7. Criando o Cluster ECS

1. Vá para **ECS > Clusters > Criar Cluster**
 2. Nome: nava-log-cluster
 3. Tipo: **Fargate**
 4. Escolha a VPC nava-log-vpc e as **subnets privadas**
 5. Clique em **Criar**
-

8. Criar a Tarefa ECS

1. Vá para **ECS > Task Definitions > Create New Task Definition**.
2. Escolha **Fargate**.
3. Adicione um container:
 - **Image:** <URI do ECR>
 - **Port Mapping:** 8081 (container) → 80 (host)
 - **Environment Variables:**
 - **DB_URL:**
jdbc:postgresql://<RDS-ENDPOINT>:5432/nava_log
 - **DB_USERNAME:** <RDS USER>
 - **DB_PASSWORD:** <RDS PASSWORD>
 - **MONGO_URI:** <MongoDB Atlas URI>

4. Escolha **Execution Role** e selecione uma IAM Role com permissões de ECS.

9. Criar o Service ECS

1. Vá para **ECS > Services > Create Service**.
2. Escolha **Fargate**, o Cluster `nava-log-cluster`, e a Task Definition criada.
3. Escolha `nava-log-ecs-sg` no Security Group.
4. Associe ao Load Balancer.

10. Criar o WireMock no ECS

1. Crie uma nova **Task Definition** com:
 - **Image:** `wiremock/wiremock`
 - **Port Mapping:** 8080 (container) → 9090 (host)
 - **Environment Variables:**
 - `WIEMOCK_PORT: 8080`
 - **Execution Role:** mesma usada no ECS.
2. Crie um **novo serviço no ECS**, associando ao mesmo Cluster e Load Balancer.

11. Criar as IAM Roles

1. Vá para **IAM > Roles > Create Role**.
2. Escolha **EC2 e ECS Tasks**.
3. Anexe permissões:
 - `AmazonEC2ContainerRegistryFullAccess`
 - `AmazonRDSFullAccess`
 - `AmazonECS_FullAccess`
 - `SecretsManagerReadWrite`
4. Associe essa Role às Task Definitions.

12. Deploy da Aplicação

Agora, execute os seguintes comandos no terminal:

```
docker build -t strobson/nava-log-app:latest .  
docker tag strobson/nava-log-app:latest <ECR_URI>/nava-log-app:latest  
docker push <ECR_URI>/nava-log-app:latest
```

```
aws ecs update-service --cluster nava-log-cluster --service nava-log-service --force-new-deployment
```