ECS Task Role vs. ECS Task Execution Role: What's the Difference?

1. **ECS Task Role**
   - The **ECS Task Role** allows your **task** (which is the actual containerized application) to **make API calls** to other AWS services. For example, if your app running inside an ECS task needs to access an S3 bucket, DynamoDB, or an RDS database, the task role is what gives it permission to do so.
   - This role is **specific to the application** running inside your task. It ensures that each task has the necessary permissions to interact with other AWS resources securely and in isolation from other tasks.

Example use case:

- Your containerized app needs to upload files to S3 or query DynamoDB.

Documentation:
ECS Task Role - AWS Documentation

---

2. **ECS Task Execution Role**
   - The **ECS Task Execution Role**, on the other hand, is used **by ECS itself** to carry out operations needed to launch and run your task. This role allows ECS to pull container images from Amazon ECR or Docker Hub, and to write logs to CloudWatch on behalf of your task.
   - This role is required for managing the infrastructure aspects of your ECS tasks. The tasks themselves don't use it directly, but ECS does.

Example use case:

- ECS needs to download your container image from ECR or Docker Hub, or it needs to write logs to CloudWatch.

Documentation:
ECS Task Execution Role - AWS Documentation

Summary of Differences

- **ECS Task Role**: Permissions **for the application running inside the ECS task** to interact with other AWS services.
- **ECS Task Execution Role**: Permissions **for ECS itself** to manage the infrastructure aspects of the task (e.g., pulling images, writing logs).