Awesome! You're ready to tackle a more advanced AWS project. Here's a comprehensive **practice task** that incorporates all the services you listed: **EC2, S3, Lambda, VPC, RDS & Aurora, CloudFront, CloudFormation, IAM, and Route 53**.

**🚀 AWS Practice Project: "Scalable Blog Platform"**

**🎯 Objective:**

Build a scalable blog platform hosted on EC2, with images stored in S3, dynamic content served via Lambda, data stored in RDS/Aurora, accelerated via CloudFront, managed with CloudFormation, secured with IAM, and domain-mapped using Route 53.

**🧩 Services Breakdown**

| **Service** | **Role in Project** |
| --- | --- |
| EC2 | Hosts the blog frontend (e.g., WordPress or custom app) |
| S3 | Stores static assets like images, CSS, JS |
| Lambda | Handles dynamic tasks like image optimization or comment moderation |
| VPC | Isolates resources with public/private subnets |
| RDS/Aurora | Stores blog posts, user data, comments |
| CloudFront | Speeds up delivery of static content from S3 |
| CloudFormation | Automates infrastructure setup |
| IAM | Manages access control across services |
| Route 53 | Maps domain name to your blog |

**🛠️ Step-by-Step Tasks**

**1. CloudFormation Template**

* Write a CloudFormation template to provision:
  + VPC with public/private subnets
  + EC2 instance
  + S3 buckets
  + RDS (Aurora MySQL or PostgreSQL)
  + IAM roles
  + Lambda function
  + CloudFront distribution
  + Route 53 hosted zone and record set

**2. VPC Setup**

* Create a VPC with:
  + 2 public subnets (for EC2, NAT Gateway)
  + 2 private subnets (for RDS, Lambda)
  + Internet Gateway, NAT Gateway, route tables

**3. EC2 Instance**

* Launch EC2 in public subnet
* Install a web server (Apache/Nginx) and host a simple blog app
* Connect to RDS for dynamic content
* Use IAM role to allow EC2 to access S3

**4. S3 Buckets**

* Create:
  + blog-assets-bucket for static files
  + blog-uploads-bucket for user-uploaded images
* Enable versioning and encryption
* Set up lifecycle rules to archive old content

**5. Lambda Function**

* Create a Lambda function to:
  + Optimize uploaded images
  + Moderate comments using basic NLP
* Trigger on S3 ObjectCreated events or via API Gateway

**6. RDS / Aurora**

* Launch Aurora MySQL or PostgreSQL in private subnet
* Create tables for posts, users, comments
* Enable backups and Multi-AZ for high availability

**7. CloudFront**

* Create a distribution for blog-assets-bucket
* Enable caching, compression, and custom error pages
* Restrict access using signed URLs or OAI

**8. IAM Roles & Policies**

* Create roles for:
  + EC2 to access S3 and RDS
  + Lambda to access S3 and CloudWatch
  + CloudFormation to provision resources
* Use least privilege principle

**9. Route 53**

* Register or use an existing domain
* Create a hosted zone
* Add A record pointing to EC2 public IP or CloudFront domain

**✅ Bonus Challenges**

* Add HTTPS using ACM and CloudFront
* Use Auto Scaling for EC2
* Add WAF for security
* Monitor with CloudWatch dashboards
* Use Secrets Manager for DB credentials

Would you like help writing the CloudFormation template or setting up any specific part of this project?