## ✅ **Step 1 — Define the architecture**

Before you touch AWS, answer:

* Where will the Node.js API run? (EC2? ECS? Lambda?)
* Where will MongoDB run? (Managed or self-hosted?)
* Will you need file storage? (e.g., S3 for images)
* Will you need authentication? (Cognito)
* Will you need domain + HTTPS? (Route 53 + Certificate Manager)

## **✅ Step 2 — Get an AWS account**

* [Create an AWS Free Tier account](https://aws.amazon.com/free/) if you haven’t.
* Set up IAM users: never use the root account for daily tasks.
* [Learn IAM basics](https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html).

## **✅ Step 3 — Pick your backend hosting**

🔹 Option A: Launch an EC2 instance

* Spin up a Linux server (Ubuntu 22.04).
* Install Node.js, PM2, NGINX.
* Deploy your API code.
* Point a domain to it.

[EC2 Getting Started](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html)

🔹 Option B: Use ECS + Docker

* If you know Docker, package your API in a container.
* Run it on ECS Fargate (serverless containers).

[ECS Getting Started](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_GetStarted.html)

✅ For beginners, EC2 is easiest.

## **✅ Step 4 — Connect your database**

* Use [MongoDB Atlas](https://www.mongodb.com/cloud/atlas) (free tier) → easy, secure, managed backups.
* Or self-host MongoDB on another EC2 → more work, not recommended for beginners.

## **✅ Step 5 — Store media files**

If your app uses property images:

* Use [Amazon S3](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html) to store images.
* Generate signed URLs from your API for secure uploads.

## **✅ Step 6 — Add security**

* Set up a domain (Route 53).
* Add HTTPS (Certificate Manager + NGINX reverse proxy).
* Use IAM roles carefully.
* Add environment variables with secrets (DB URI, JWT secret).

## **✅ Step 7 — Monitoring & scaling**

* Use [CloudWatch](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html) for logs & metrics.
* Set up auto-recovery for EC2 if it fails.
* Use Elastic Load Balancer if you need multiple instances.

## **✅ In short: your first AWS step**

👉 Provision an EC2 instance, secure it, deploy your Node.js API, point your mobile app to your public API URL.

## **🎯 Practical Getting Started**

1️⃣ Launch EC2 (Ubuntu)  
2️⃣ SSH in, install Node.js, PM2, NGINX  
3️⃣ Clone your backend repo  
4️⃣ Install dependencies  
5️⃣ Start the server with PM2  
6️⃣ Configure NGINX as a reverse proxy  
7️⃣ Test your REST API from Postman  
8️⃣ Connect your mobile app to it

If you want, I can write you:

* A step-by-step EC2 deploy guide
* Nginx config for Node.js
* MongoDB Atlas connection setup
* S3 bucket + signed URL code

Just say: “Make the EC2 deploy guide.”