Here's a structured learning approach that builds incrementally:

**Phase 1: Foundation Setup (Day 1)**

**Morning: MongoDB Basics**

1. Install MongoDB locally OR set up MongoDB Atlas (cloud)
2. Use MongoDB Compass (GUI) to create database, collections
3. Practice basic operations: insert, find, update, delete
4. Compare to SQL: Collections vs Tables, Documents vs Rows

**Afternoon: .NET Integration**

1. Create dotnet new webapi -n TaskAPI
2. Install MongoDB.Driver NuGet package
3. Build simple CRUD operations with hardcoded connection
4. Test endpoints with Postman/curl

**Phase 2: Containerization (Day 2)**

**Morning: Dockerize API**

1. Create Dockerfile for .NET app
2. Build and run container locally
3. Fix any connection/environment issues

**Afternoon: Docker Compose**

1. Add MongoDB container to docker-compose.yml
2. Connect API to containerized MongoDB
3. Use environment variables for connection strings
4. Verify full stack works locally

**Phase 3: Kubernetes Basics (Day 3)**

**Morning: K8s Fundamentals**

1. Install kubectl, set up local cluster (Docker Desktop K8s)
2. Deploy simple nginx pod to understand concepts
3. Learn kubectl commands: get, describe, logs, exec

**Afternoon: Deploy Your App**

1. Create basic deployment.yaml for your API
2. Create service.yaml for networking
3. Deploy and troubleshoot connectivity

**Phase 4: Production Setup (Day 4)**

**Morning: Persistent Storage**

1. Add MongoDB StatefulSet with persistent volumes
2. Configure proper networking between API and DB

**Afternoon: Azure Migration**

1. Create AKS cluster
2. Push images to Azure Container Registry
3. Deploy to cloud and test

**Learning Tips:**

* **Document everything** - save commands, configs, troubleshooting steps
* **Break when stuck** - move to next concept, come back later
* **Test incrementally** - verify each step works before moving on
* **Use official docs** - MongoDB docs, Docker docs, Kubernetes docs are excellent

Want me to detail any specific phase?