

Homework — Software Architecture

Estimated time: ~45 min · No code required · Just a browser and a notebook

Three short tool-tour exercises. The goal: turn the four words from the theory page (*monolith*, *microservices*, *serverless*, *event-driven*) into something you can see in real reference architectures, real engineering blog posts, and real cloud pricing. By the end you will have classified a diagram, read a war story, and felt — in dollars — what "pay-per-request" actually means.

Before you start

Keep a one-page notebook open (paper or a text file). For each task, write down the short answers under (a)/(b)/(c). You are not turning this in — you are training the muscle of *reading* architecture diagrams instead of just nodding at them.

The three tasks

Task 1 — Read a real architecture diagram

Open the AWS Architecture Center at aws.amazon.com/architecture. Pick any reference architecture — e.g., *Serverless E-Commerce*, *Web Application Hosting*, or anything from the *Featured* carousel. Look at the diagram for a couple of minutes before reading the description.

Self-check — answer all three:

- **(a)** Classify the architecture into one of the four shapes from this lesson — *monolith*, *microservices*, *serverless*, *event-driven* — or a hybrid (most real ones are). One sentence of reasoning.
- **(b)** Trace one request path with your finger (or cursor) from the user box on the left through to the database / storage on the right. Note the order of

services it passes through.

- **(c)** Name three AWS services in the diagram you already recognise from Lesson 03 (compute / storage / networking basics), and one you have *not* seen before. Look that one up in a single sentence.

✓ **Self-check: you can name the shape, the path, and at least one new service from this single diagram.**

Task 2 — Find a "moved off" or "moved to" microservices story

Search Google or Hacker News for `"monolith to microservices"` OR `"microservices to monolith"`. Pick one engineering-blog post from a real company. The famous Amazon Prime Video *"monolith move"* post is one option, but anything counts as long as it is a first-party story from the team that lived through it.

Self-check — answer all three, two sentences each is enough:

- **(a)** What was the company's *original* architecture? Use the vocabulary from this lesson.
- **(b)** What did they move *to*?
- **(c)** What was the headline reason for the move? Cost, team velocity, reliability, latency, on-call pain — pick the one the blog post actually leads with.

✓ **Self-check: you can explain, in one paragraph, why a real team changed shape — without using the word "scalability" as a hand-wave.**

Task 3 — Feel serverless pricing in dollars

Open `aws.amazon.com/lambda/pricing`. Using either the pricing page itself or the AWS Pricing Calculator linked from it, compute the monthly cost of a function that is invoked **1,000 times per day** with **512 MB** of memory and runs for **200 ms** each. Then re-compute the same function at **1,000,000 invocations per day**.

Self-check — answer all three:

- **(a)** Approximate monthly cost at the low volume (1,000/day). Hint: it will likely be \$0 or a few cents, because of the free tier.
- **(b)** Approximate monthly cost at the high volume (1,000,000/day — roughly 30M invocations/month).
- **(c)** One sentence on what happens to the *cost per call* as volume grows. (Hint: it stays roughly linear — that linearity is the whole point of serverless pricing, and also the thing that bites you when traffic spikes.)

✓ **Self-check: you can quote two numbers (low and high volume) and explain in one sentence why serverless is "cheap until it isn't".**

Wrap-up

If you completed all three tasks, you have done three things that practising engineers do every week: *read* an architecture diagram without panicking, *read* someone else's hard-won lesson about changing shape, and *price out* a cloud workload before committing to it. Those three habits, repeated, are most of what "thinking architecturally" actually looks like in 2026.

Going further (optional)

Repeat Task 1 with a non-AWS diagram — Google Cloud's *Architecture Center* and Azure's *Architecture Center* publish similar reference diagrams. Notice how the shapes (monolith / microservices / serverless / event-driven) are the same even when the service names change. That is the lesson: the vocabulary travels.