

AI Assisted Courseware Development and Student Learning support

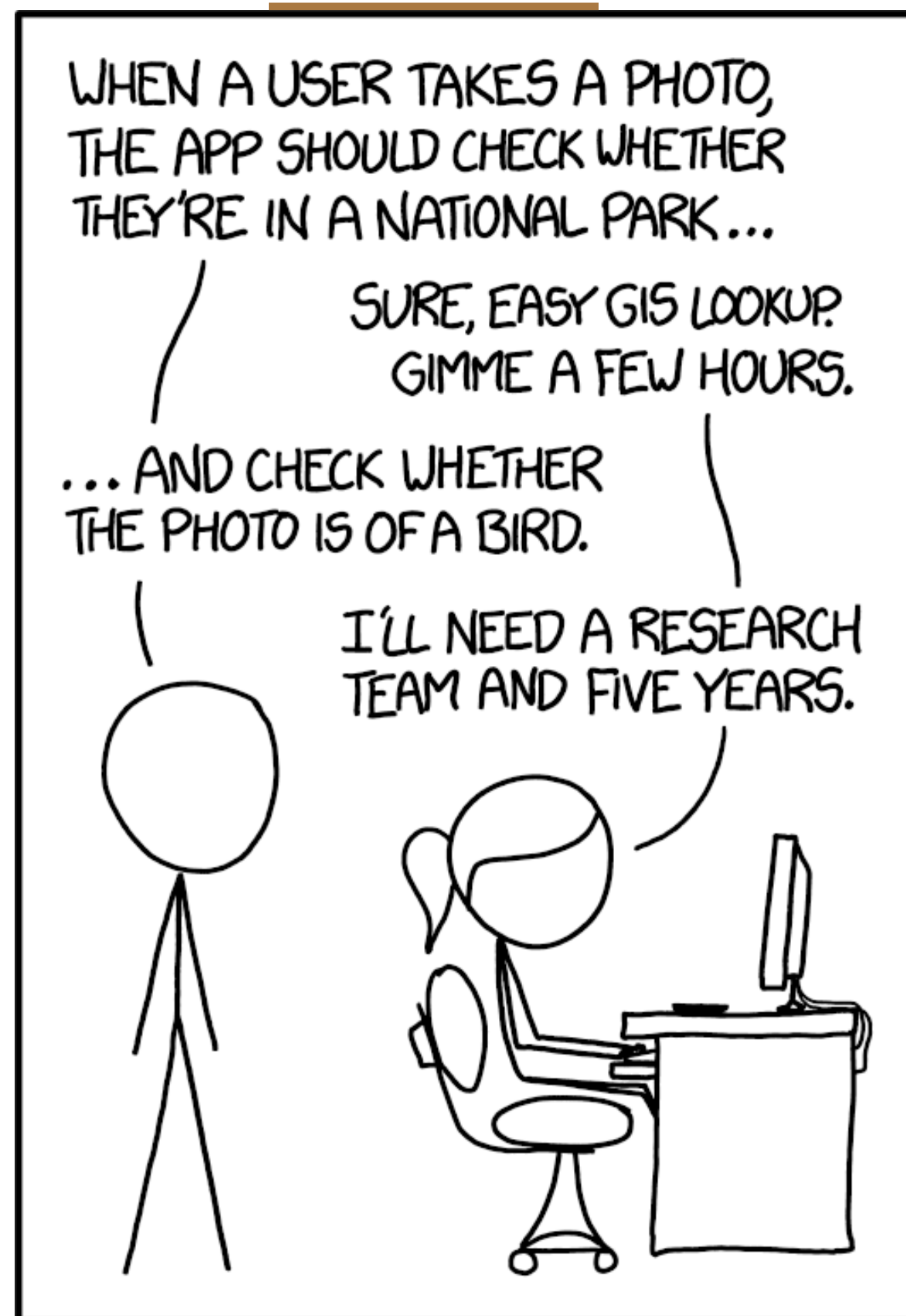
“Automating CS 35100 Cloud Computing”

Connor Couetil, Senior Project Spring 2026

<https://www.cs351.cloud>

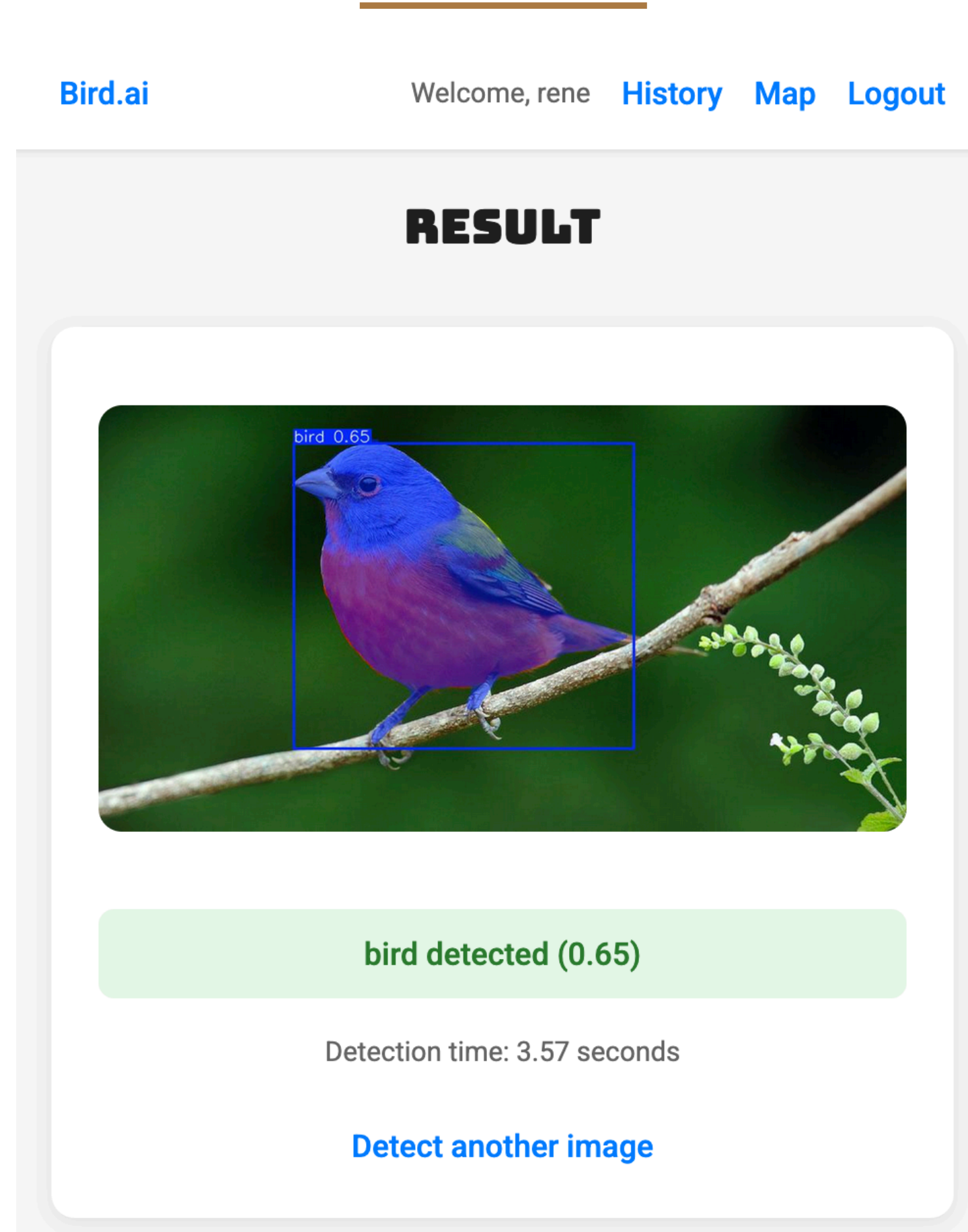
<https://senior-project.couetil.com>

2014



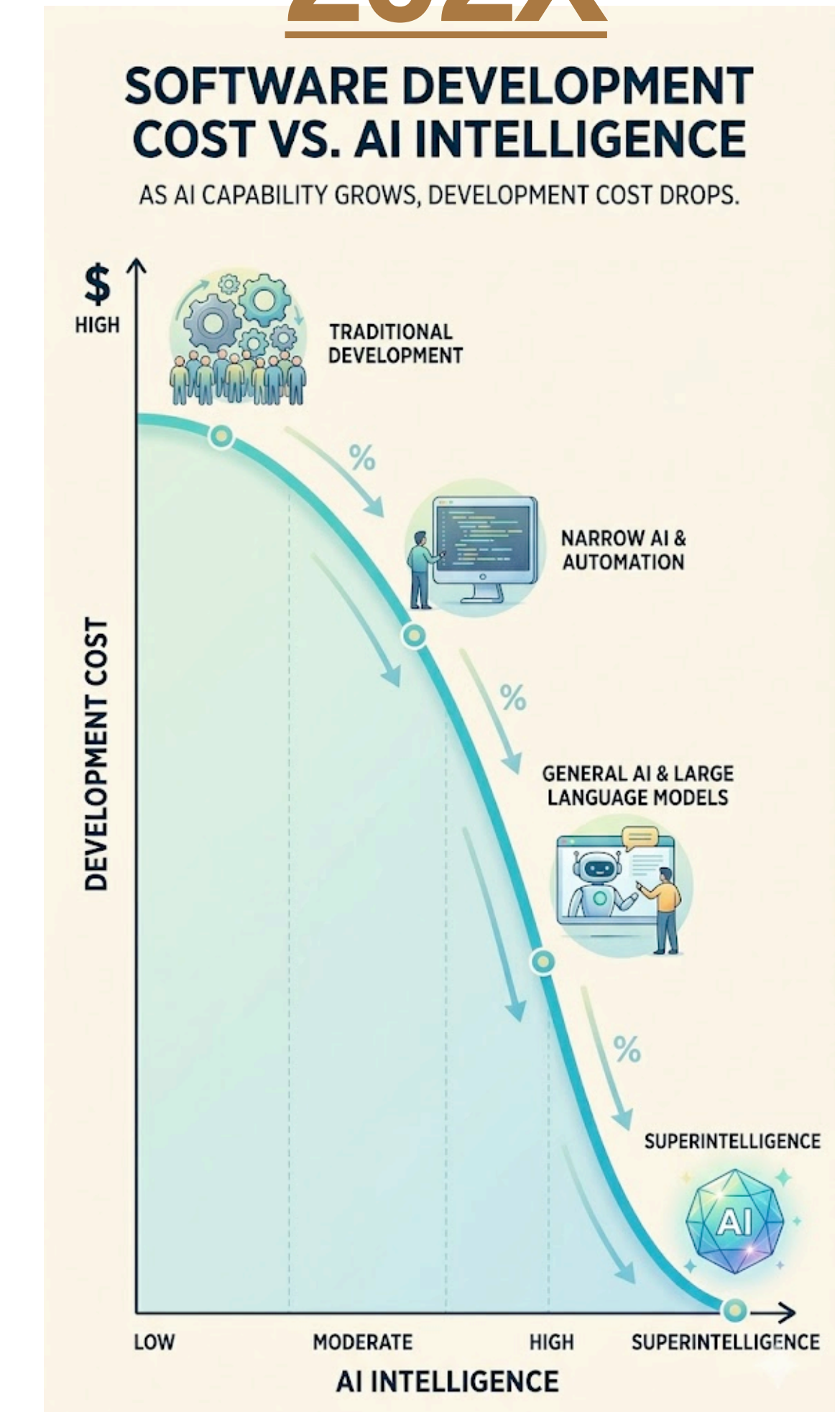
IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

2026

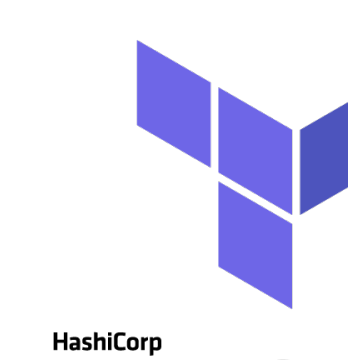
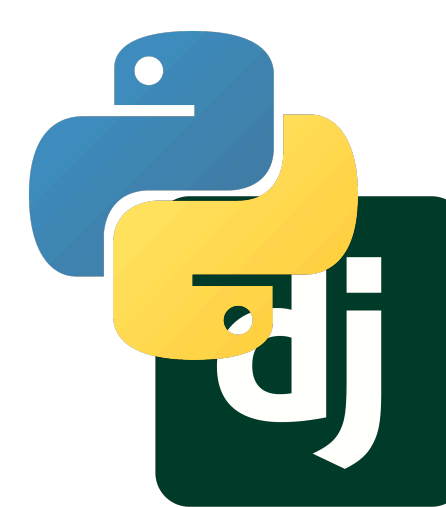


(On a t4g.small!)

202X



Cloud Assignments



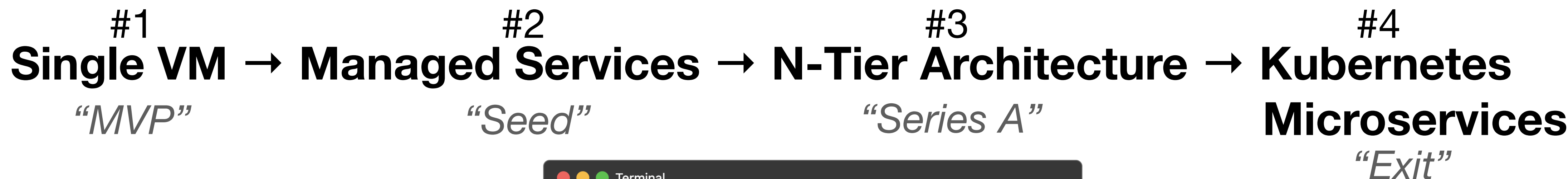
ANSIBLE

HashiCorp

Terraform

kubernetes

- System + Application
- Polished Design
- Story-driven
- Multifaceted



Instructions

Complete the following "Part 1" and "Part 2" sections to accumulate assignment points. Each section contains tasks for you to perform. Some will tell you exactly what to do, while others will give you hints or ask you to refer to a previous Cloud Assignment. Take your time and learn from the process. When you see



in a task section, submit your credentials to the assignment on Gradescope so the autograder can give you feedback on your progress.

```
Terminal
$ cd ansible
$ ansible-playbook playbook.yml
```

Voila! Visit your EC2 instance's public IP and you'll see "Welcome to nginx!"

Autograder
+10 (20/100)

Food for thought

What steps does the Ansible playbook take to prepare the instance? Do you recognize any of them? What similarities and differences do you see in the approach?

The More You Know

Why a c7i-flex.large? The [c7i](#) family is **compute-optimized**, meaning it has a higher ratio of CPU to memory compared to general-purpose instances. That's a good fit for a reverse proxy. Nginx spends most of its time doing CPU-bound work like handling TLS termination, compressing responses, and managing thousands of concurrent connections, rather than holding large amounts of data in memory. The "flex" variant gives you shared baseline CPU at a lower cost with the ability to burst when traffic spikes.

Vibe Check

Was this deploy of bird.ai harder or easier than in Cloud Assignment 2? What stood out to you most about the differences between the deployment methods between assignments? Let us know on [Ed](#)!

Hint

Refer to Compose documentation showing the syntax for [build](#) and [ports](#), and [make sure Docker Compose is installed](#).

The More You Know

A Docker Compose file has three important [top-level elements](#): [services](#), [networks](#), and [volumes](#). Services define the containers that make up your application, networks control how they communicate, and volumes manage persistent storage.

The `build` key specifies the [build context](#), a directory that contains a Dockerfile. Compose will build the image from the Dockerfile automatically when you run `docker compose up`.

Compose also has useful features for mimicking production constraints. For example, if your EC2 instance has 0.5 vCPU and 2 GB RAM, you can set matching [resource limits](#) on your service:

```
compose.yaml
services:
  proxy:
    build: ./nginx
    deploy:
      resources:
        limits:
          cpus: "0.5"
          memory: 2G
```

This helps you catch performance issues locally before they surprise you in production.

Part 1: N-Tier Architecture

You're going to be implementing a [N-Tier Web Architecture](#) to accommodate the increasing load on the bird.ai application. This is a versatile system architecture that lets you scale individual components according to how your application's usage patterns evolve. Our initial minimum-viable product (MVP) had a strong benefit, simplicity, which allowed us to move quickly and develop a proof-of-concept on a single virtual machine. However, this virtual machine becomes a [single point of failure](#).

[Mark Zuckerberg](#) gave a talk to a Harvard CS class in 2005, where he says ["The first decision we had to make, was how to expand the architecture to go from a single school type setup that we had just at Harvard, to something that supported multiple schools"](#), and that ["one really important decision that's helped us scale really well, is how we decided to distribute the data."](#) They were able to scale to 300,000 users and 400 million page views a day in 2005 using a N-tier web architecture!

This architecture continues to be relevant today, and we will enhance it by implementing automation that lets us manage it effectively with a small team.

“Auto-grading”

- Automatic feedback to develop student mastery
- Data + analysis for course instructional team

Cloud Assignment 3 [Edit](#) [Analysis](#)

Released ⓘ Mar 24, 3:00 PM EDT
 Due ⓘ Apr 3, 11:00 PM EDT
 Timezone America/Indiana/Indianapolis
 Draft No
 Course [CS 351 Spring 2026](#)
 Has API Key Yes
 Created Mar 1, 3:45 PM

-
-
-
-
-
-
-
-
-
-

Autograder Results

Autograder Output (hidden from students)

```
[delegate] Job 154: polling https://www.cs351.cloud/api/autograde/154/status/
[delegate] Job 154: running (10s)
[delegate] Job 154: completed (20s)
[delegate] Downloaded results to /autograder/results/results.json
```

0) Submitted credentials file (0/0)

- Credentials file has [default] section
- Credentials file does not contain "aws_access_key_id"
- Credentials file does not contain "aws_secret_access_key"
- Credentials file has "aws_account_id"
- Valid AWS Account ID (897729129333)
- Credentials file has "external_id"
- external_id contains only valid characters

1) Initial Setup (0/0)

- Found AWS Account (ID: 897729129333)
- Found budget: My Zero-Spend Budget
- Found role "CS351-autograder"
- No user named "CS351-autograder" exists

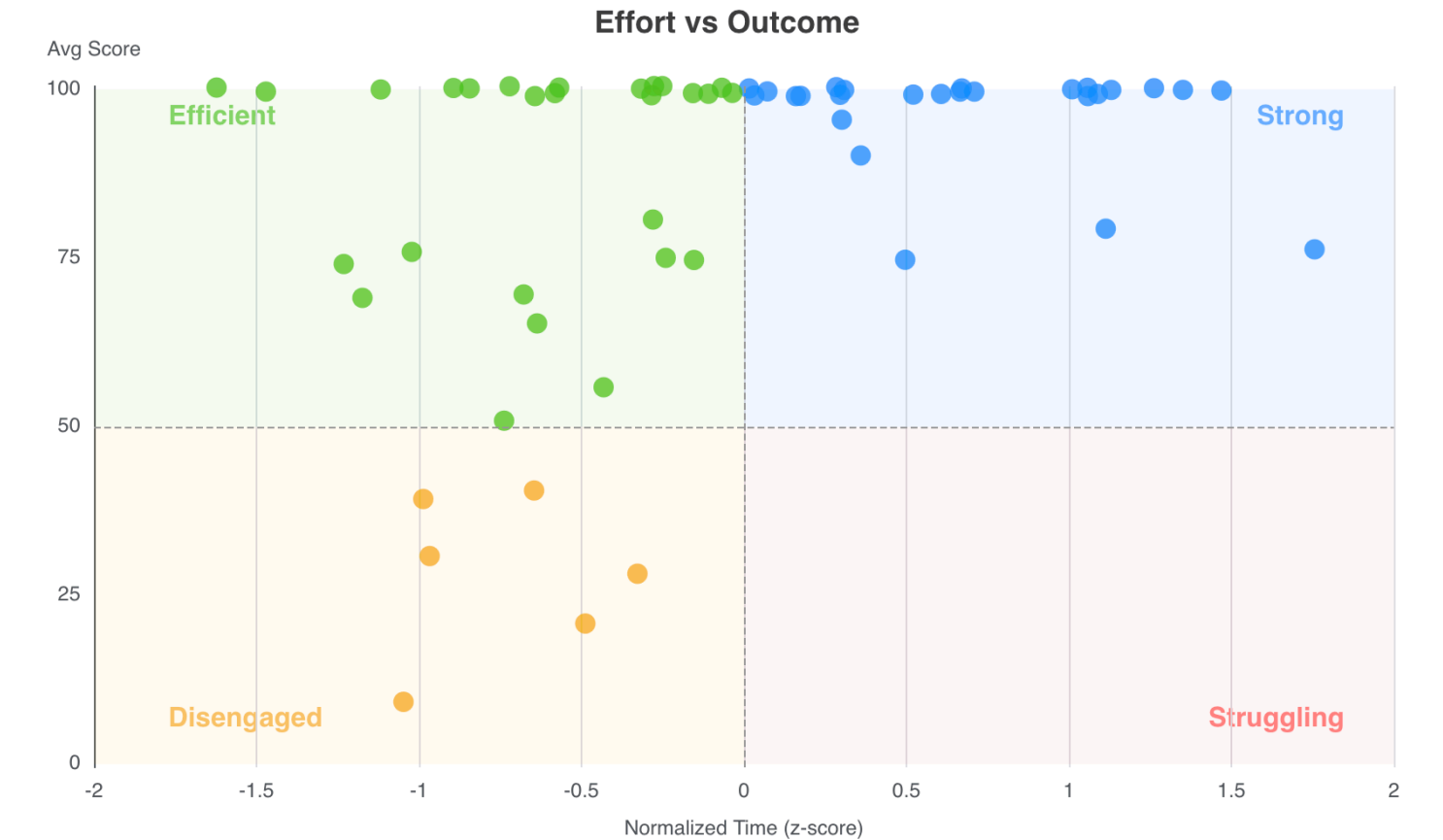
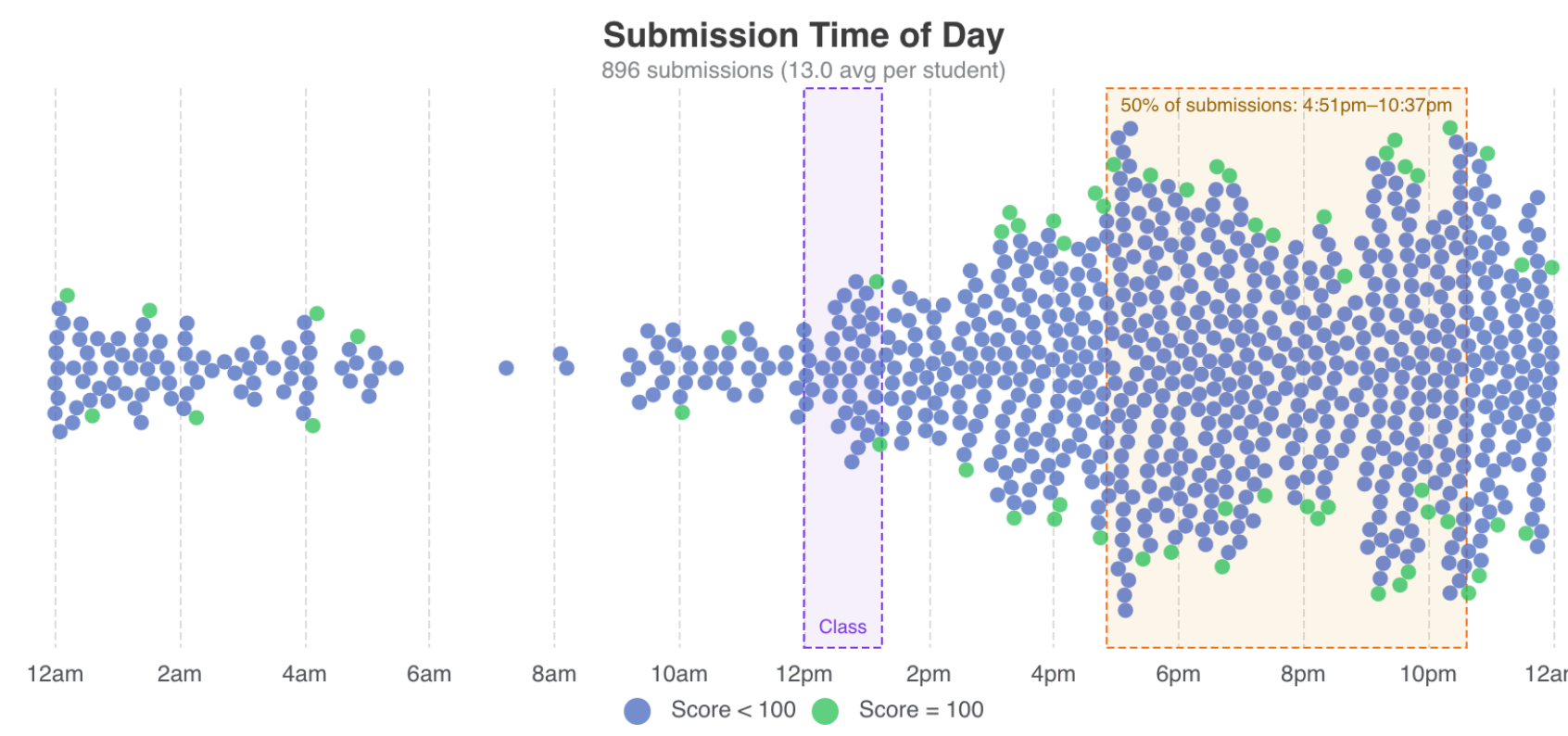
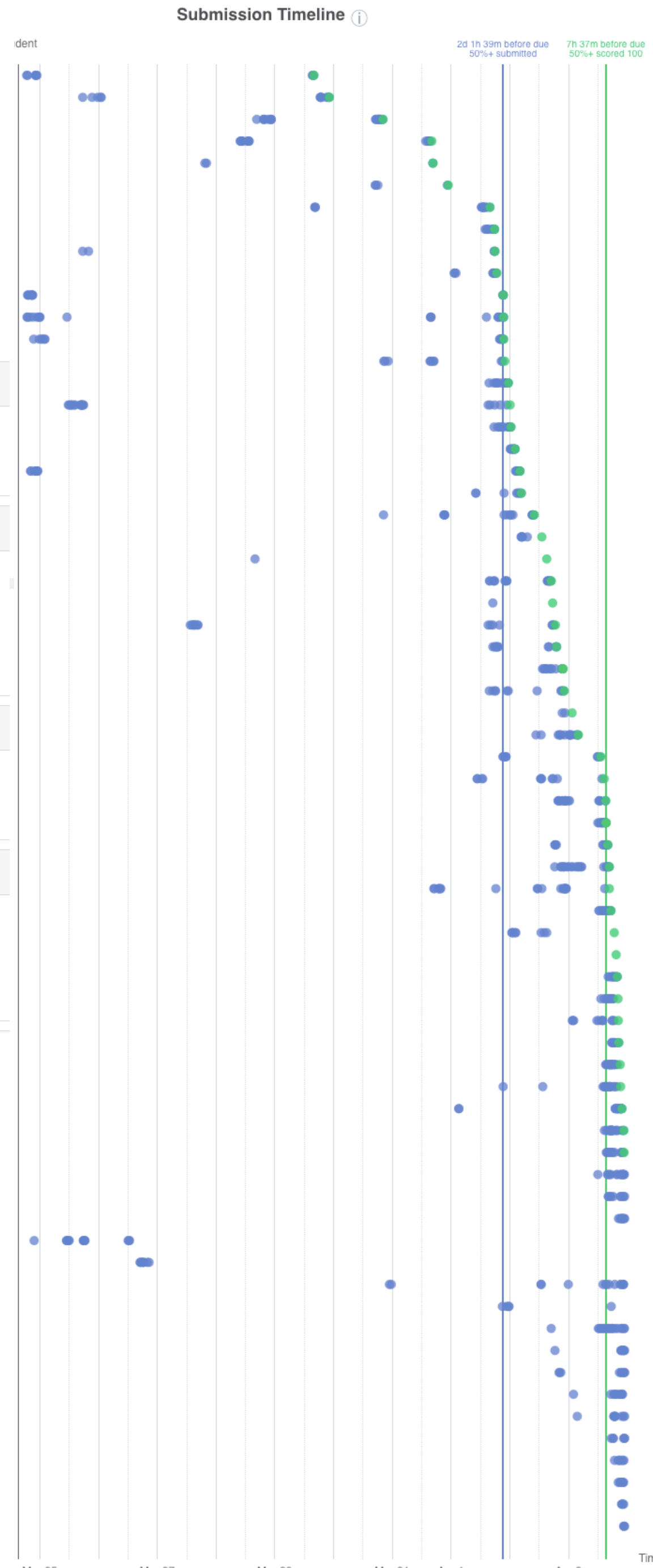
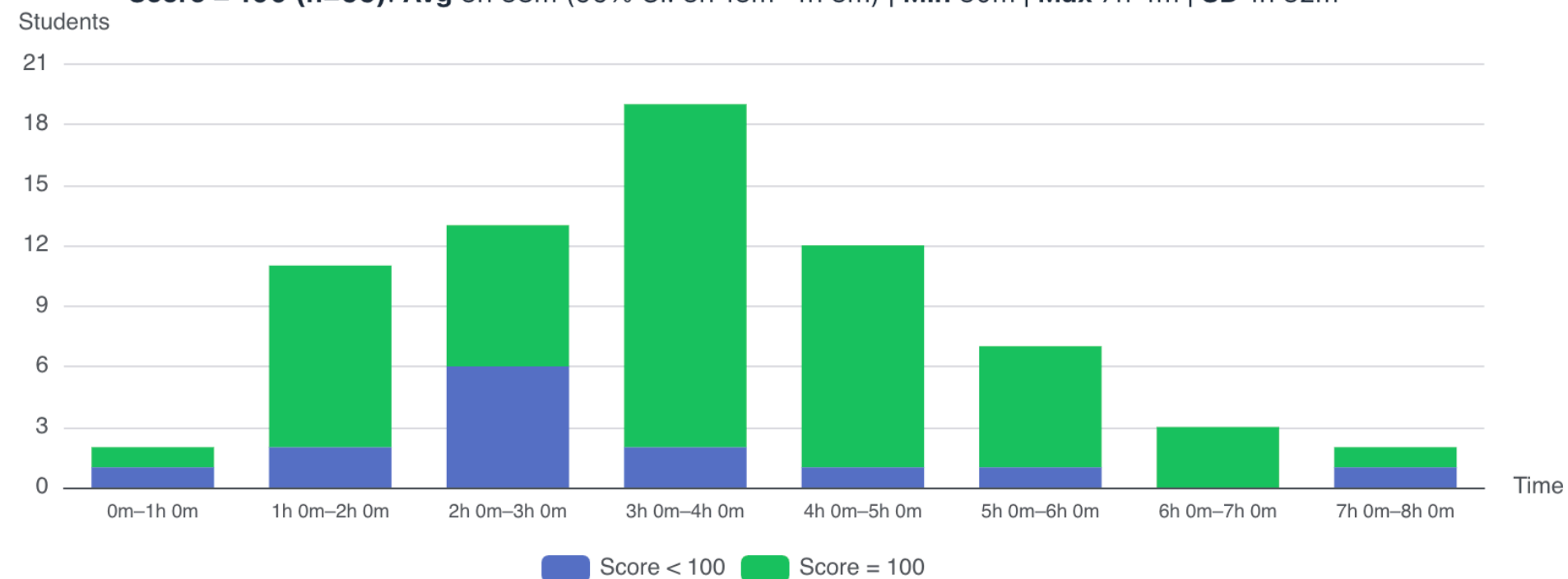
2) Pushing to a container repository (10/10)

- Found public ECR repository "ca3/proxy"
- Repository contains an image tagged "latest"
- Found IAM user "ca3"
- User "ca3" has "PowerUserAccess" policy attached
- User "ca3" has "IAMFullAccess" policy attached
- User "ca3" has an access key

Charts

Time Spent Distribution ⓘ

Score = 100 (n=55): Avg 3h 38m (95% CI: 3h 13m–4h 3m) | Min 30m | Max 7h 4m | SD 1h 32m

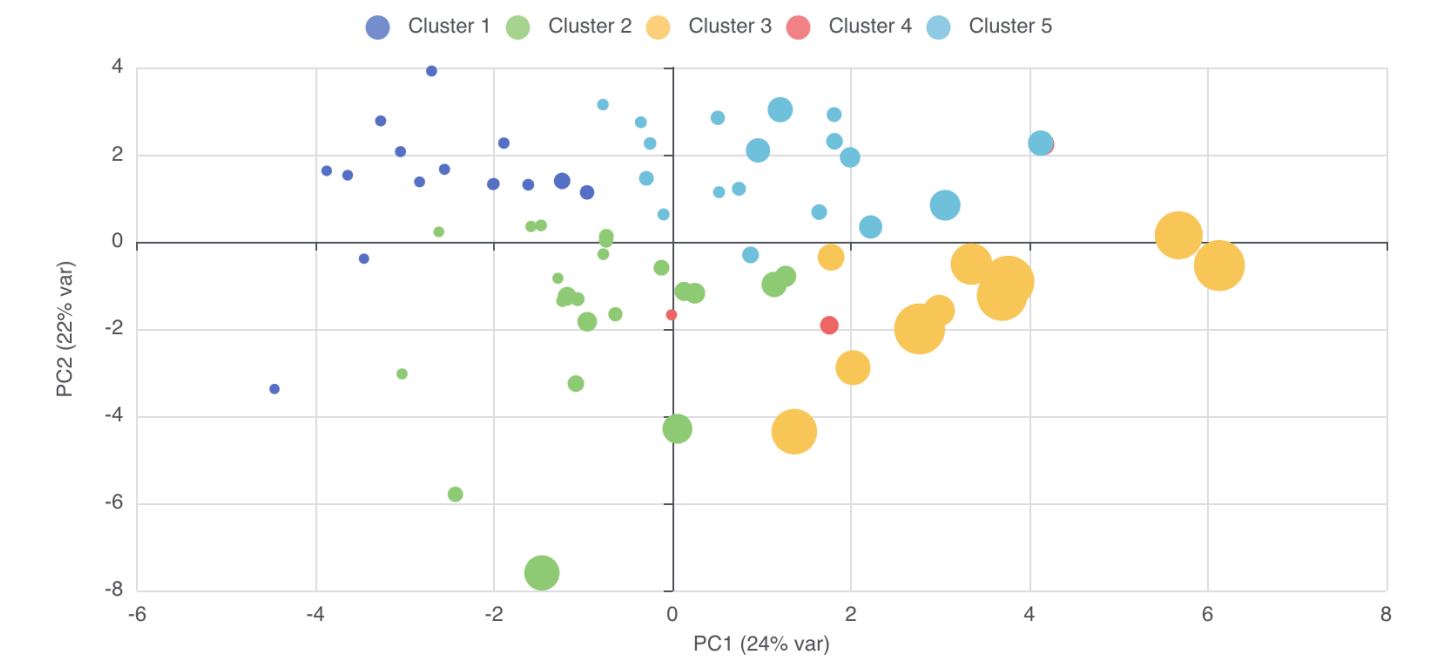


None | Days Before Due Last Submission | **Working Span Days** | Longest Break Hours | Days Before Due First Submission

Weekend Submission Pct | Last 24H Attempt Pct | Peak Submission Hour | Session Count | Night Submission Pct | Submission Density

Num Score Decreases | Avg Session Hours | Total Attempts | Attempts Before First Pass | Total Session Hours | Active Days

Longest Plateau | First Score | Score Improvement Rate | Score Change Ratio | Avg Minutes Between Attempts | Attempts After 100



Class Discussion Board Reply Bot

Question 6 Clarification #141



 Yesterday in **Assignments - HW5** PRIVATE PIN STAR WATCH 9 VIEWS

Hi everyone,

While Netflix uses AWS Lambda to handle event driven media transcoding, wouldn't a startup also be a good candidate to use a serverless contract for its operation since a startup is less likely to need to have a server running 24/7 if their customer count is small allowing them to leverage scale to zero?

I would love a second opinion on this!

Thanks,

 **Connor Couetil** STAFF 1d
 **Be careful** — this response was generated by AI. It can be wrong, so think critically about its suggestions — but try to learn from the reasoning it presents. An instructor will follow up if needed.

Hi 

That's a great question and you're thinking about this very well! You're absolutely right to point out that a startup *could* also benefit from serverless computing, especially with the "scale to zero" advantage you mentioned. However, the question asks which company is *most* likely to contract for serverless, and in this case, Netflix is the stronger answer.

The key difference lies in the *predictability* and *volume* of their workloads. Netflix has consistent, high-volume transcoding needs. They are constantly uploading and preparing new video content. This makes it a perfect fit for serverless functions like Lambda, which can handle these ongoing, event-driven tasks efficiently and at scale. While a startup might have periods of low activity, they also likely have more unpredictable spikes in demand as they grow, and potentially a wider variety of tasks that aren't as easily broken down into discrete, event-driven functions.

Think about it this way: serverless is excellent for well-defined, repeatable tasks. While a startup *could* use it, they're also more likely to need more control over their infrastructure as they experiment and iterate, and might benefit more from the flexibility of traditional servers — especially if they have complex, stateful applications. Netflix's transcoding workload is highly standardized, making it ideal for the serverless model. You're right to identify the potential benefits for a startup, but Netflix's established, high-volume needs make serverless a more natural and impactful choice for them.

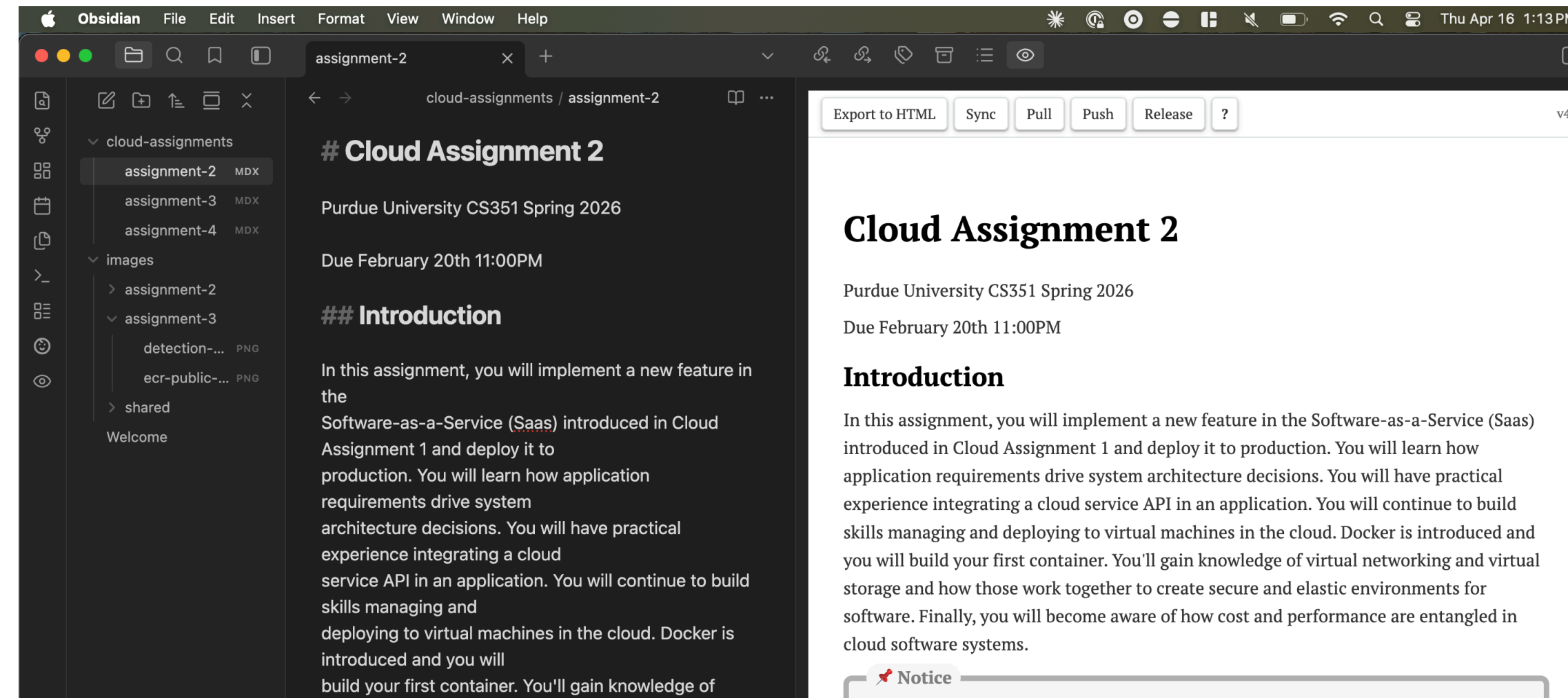
Model: *gemma3:27b* | Generated: Apr 15 3:55 PM UTC | Tokens: 1,551 in → 319 out

1 Reply Edit Delete ...

Tools

For students and instructors

Cloud-synced Obsidian Plugin



Instructional Site > Ed Discussion > AI Summaries Search... 🔍K admin Log out

AI Discussion Board Summary

AI Summaries

Categories

- [Assignments \(7\)](#)
- [Assignments > CA1 \(11\)](#)
- [Assignments > CA2 \(37\)](#)
- [Assignments > CA3 \(19\)](#)
- [Assignments > CA4 \(5\)](#)
- [Assignments > HW2 \(1\)](#)
- [Assignments > HW3 \(2\)](#)
- [Assignments > HW4 \(1\)](#)
- [Assignments > HW5 \(1\)](#)
- [General \(46\)](#)
- [Lectures \(3\)](#)
- [Social \(1\)](#)

Generate Summary: Assignments > CA3

System Prompt
You are analyzing Ed Discussion posts from a cloud computing course (CS 351) for the instructor.

Output Guidance (appended to end of context)
Analyze the discussion and organize your response into these sections:

1. Student Experience: How did students experience this assignment? What was difficult, confusing, or straightforward? Note any patterns in timing or progression.

Models

- codellama:latest
- deepseek-r1:14b
- deepseek-r1:1.5b
- deepseek-r1:32b
- deepseek-r1:70b
- deepseek-r1:7b
- devstral-small-2:latest
- gemma3:12b

Generate

Run History

Run #1 · Apr 04 17:59 · 18 threads · 73 comments · Cloud Assignment 3 [View details](#)
deepseek-r1:70b ✓ gemma3:27b ✓ gpt-oss:120b ✓ llama3.3:70b ✓ qwen3:32b ✓ qwq:32b-fp16 ✓

Per-Assignment Dashboards

CA4 Kubernetes Dashboard

Last snapshot: Apr 16, 3:55 PM

▶ Snapshot History (10)

68/70

Provisioned

1

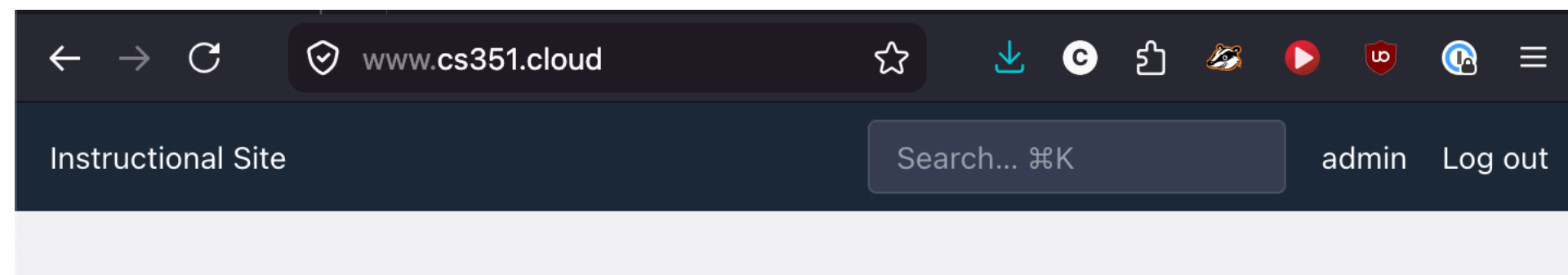
Deployed

Cluster Capacity

CPU requests

0.9 / 7.84 cores (11.5%)

Powered by Instructional Site



Welcome, admin!

ACTIVE ASSIGNMENT

Cloud Assignment 4

Due Apr 24, 11:00 PM

PAST DUE (PRACTICE)

Cloud Assignment 3

Due Apr 3, 11:00 PM

PAST DUE (PRACTICE)

Cloud Assignment 2

Due Feb 20, 11:59 PM

[CS 351 Spring 2026](#)

[Submissions](#) — View and review student assignment submissions

[Assignments](#) — Manage cloud assignments and their API keys

[Students](#) — Enrolled students and their submission history

[Billing](#) — Monitor estimated AWS charges across student accounts

[Ed Discussion](#) — Browse and analyze Ed Stem threads and comments

[Enrollments](#) — Manage enrollments for CS 351 Spring 2026

Content

[Assignment Documents](#) — Manage MDX assignment documents, versions, and releases

Looking to the Future

- Interactive assignments powered by AI and course knowledge graph
- Integrate with LMS
- AI TA in discussion boards hosted privately at Purdue (using RCAC GenAI Studio)
- Pre-built cloud architectures for new assignment development
- Agent harness for developing assignments and learning materials
- Use case specific cross evaluations of LLMs (“front-end” vs “back-end”)
- Student work session scheduling based on assignment time data and personal calendar
- More data! More analysis!