tripleten

AI Systems Engineering

For tech professionals with production experience

22-week accelerator | Online | 10–15 hours a week

How this program works

Learn to build senior-level AI systems, and make your next career step

5 projects → 1 production AI platform

10 h a week on projects · about 15 with all the theory

22 weeks

A year of career support

1

One production system, five projects, five live defenses

Each project ends with a pre-defense and a live defense in front of a working engineer. Theory is optional: skip what you know. 10 hours a week on projects.

2

Work with a senior engineer from day one

Instructors from AMD, Microsoft and Amazon. A 30-minute 1:1 every week, written feedback on every project, your project space always on.

3

Career prep for every round, and for the search itself

Up to 10 live mocks: screening, technical, recruiter and behavioral, with a plan after each. A coach rewrites your resume and LinkedIn, the platform matches roles and tailors your resume to each posting. Support for up to 12 months.

Program summary

Five production projects, defended live

You join the platform team at TripleTen Medical, a fictional company that runs Coldline, a cold-chain logistics platform. It’s already running when you arrive. Each project picks up where the last one left off, and you start every project from a reference solution of the previous one. Not a sandbox: each project is a task platform teams solve in production, with acceptance criteria known upfront.

~220 h of project work · ~110 h of theory

22 weeks total

Project 1	AI System Diagnostics and Capacity Review	~45 h
Project 2	AI Data, Retrieval, and Service Architecture	~60 h
Project 3	AI Platform Operations and Reliability	~60 h
Project 4	AI Security, Guardrails, and Governance	~60 h
Project 5	Agentic AI, Live Evaluation, and End-to-End Delivery	~105 h

Every project ends with a live defense

At about 70% you book a 30-minute pre-defense with your instructor, who has read your repository in advance and flags structural problems while there is still time to fix them. When the project is done, you defend it in a 60-minute live defense with a working engineer. If it doesn’t pass, you get written feedback and book again.

How long will it take you?

Project hours are fixed and set the timeline. Theory is optional and flexes with your experience, so two people can finish the same program on different schedules.

5 h a week	**10 h a week**	15 h a week	20 h a week
44 weeks on projects only	22 weeks on projects only	15 weeks on projects only	11 weeks on projects only
66 weeks with all the theory	33 weeks with all the theory	22 weeks with all the theory	17 weeks with all the theory

Based on 220 hours of project work and 330 hours with all the theory. Plan for 10 hours a week on projects. From there, how long it takes depends on the hours you invest each week, your starting level and how engaged you stay. The program is flexible and fits around your current schedule.

AI System Diagnostics and Capacity Review

You inherit Coldline and diagnose its AI-assisted exception path. The system is already running locally: an API, a worker, a model provider, PostgreSQL, Redis, dashboards, traces, logs and a synthetic load generator. You don’t redesign the product. You trace one important request path, repair the observability signals that are missing or wrong, run a reproducible load test, correct an AI-generated capacity analysis, and write a scaling recommendation backed by evidence.

Project work	~30 h
Theory (optional)	~15 h
Total	~45 h

Theory topics

- AI request paths, synchronous and asynchronous work, provider failure modes
- Golden signals, context propagation, traces, metrics, logs, cardinality
- Latency percentiles, throughput, concurrency, queues, saturation, capacity math
- Cache and database evidence, bottleneck attribution, alternative-cause elimination
- AI-output verification, ADRs, evidence-based engineering decisions

Skills you’ll add

Observability · OpenTelemetry & Jaeger · Prometheus & Grafana · Load testing with Locust · Capacity math · PostgreSQL · Redis · Docker Compose

Learning objectives

Explain runtime behavior

Explain where state, latency and failure accumulate in an AI-assisted request.

Attribute bottlenecks

Distinguish application, provider, database, cache and queue bottlenecks using evidence.

Repair instrumentation

Repair incomplete telemetry without creating unsafe metric cardinality.

Estimate growth

Estimate 10x growth with named assumptions and a bounded confidence statement.

Project 2

AI Data, Retrieval, and Service Architecture

You extend Coldline with authorization-aware retrieval over shipping procedures, handling rules and incident playbooks. The parser, the corpus, the embedding path and the hybrid search are supplied; the design choices behind them are yours. You implement one retrieval extension behind a stable port, evolve one REST path and protect it with idempotency, apply one migration, tune one parameter, attribute one retrieval miss to a named stage, and compare PostgreSQL with pgvector against a supplied Qdrant profile.

Project work	~40 h
Theory (optional)	~20 h
Total	~60 h

Theory topics

- Retrieval architecture, chunking, embeddings, provenance, data-quality boundaries
- pgvector, PostgreSQL full-text search, fusion, filters, precision, recall, latency
- Ports and adapters, REST versioning, idempotency, migrations, transactions
- Authorization-aware retrieval and metadata propagation
- Golden sets, binary relevance, controlled experiments, failure attribution

Skills you'll add

Retrieval & pgvector · Hybrid search · REST versioning · Idempotency · Migrations · Ports & adapters

Project 2

Learning objectives

Explain retrieval

Explain how chunking, embeddings, keyword search, fusion, filters and metadata affect results.

Extend behind a port

Build one bounded retrieval change behind a stable application port.

Guard against duplicates

Prevent cross-scope retrieval and duplicate write side effects.

Evolve contracts

Evolve one relational or API contract without breaking existing consumers.

AI Platform Operations and Reliability

You operate Coldline under provider, queue and dependency failure. You perform one controlled rollout and rollback, bound one model-provider path with timeouts and retries, build a second model adapter and decide when to switch to it, configure one asynchronous workflow on an SQS-semantic queue, recover one dead-lettered message without duplicate effects, define one user-impact SLO and alert, complete one CI gate, and run one failure lab. The project assesses reliable operation of an AI system behind stable ports, not Kubernetes administration.

Project work	~40 h
Theory (optional)	~20 h
Total	~60 h

Theory topics

- Container health checks, immutable tags, rollout, rollback, resource bounds
- Provider timeouts, retry budgets, fallback, load shedding, circuit-breaker decisions
- Visibility timeouts, delivery semantics, dead-letter queues, redrive, idempotency
- SLOs, error budgets, alert design, backpressure, telemetry correlation
- CI gates, image provenance, release verification, recovery runbooks
- Kubernetes, Kafka and managed streaming as decision-level comparisons

Skills you'll add

Rollout & rollback

Timeouts, retries & provider adapters

Queues & DLQ

SLOs & alerting

CI gates

Failure labs & runbooks

Learning objectives

Operate under failure

Operate an AI-assisted workflow under latency, timeout, queue and dependency failures.

Recover without duplicates

Recover one failed asynchronous job without duplicate effects.

Tie signals to users

Tie reliability signals to user impact rather than infrastructure noise.

Prove it

Prove rollout, rollback, alerting, recovery and CI claims with repeatable evidence.

AI Security, Guardrails, and Governance

You secure one high-value exception-resolution workflow. You threat-model it with STRIDE, protect one endpoint with OIDC and RBAC, add output validation and audit logging, integrate and test a supplied PII redactor, move one secret through the secret provider, complete one CI security gate, correct one flawed AI-generated governance artifact, and issue one technical risk decision.
The outcome is inspectable security evidence, not legal certification.

Project work	~40 h
Theory (optional)	~20 h
Total	~60 h

Theory topics

- STRIDE, trust boundaries, abuse cases, prompt injection, output handling, tool authority
- OIDC, JWKS, RBAC, scopes, least privilege, negative authorization testing
- Output schemas, fail-safe behavior, PII redaction, audit trails, residual risk
- Secret custody, runtime injection, rotation concepts, stable provider boundaries
- Static analysis, image and dependency scanning, secret scanning, security CI
- EU AI Act and HIPAA-style control mapping, without certification claims

Skills you'll add

Threat modeling (STRIDE)

OIDC & RBAC **Output guardrails**

PII redaction & audit logging

Secrets management **CI security gates**

Learning objectives

Identify realistic threats

Identify AI-system threats at concrete trust boundaries.

Apply controls proportionately

Authentication, authorization, validation, redaction, audit and secret custody.

Prove controls

Prove controls with positive, negative and held-out evidence.

Reject overclaims

Reject unsupported claims in AI-generated security and governance analysis.

Agentic AI, Live Evaluation, and End-to-End Delivery

The capstone, in two phases.
In Build and Acceptance you extend a supplied two-node LangGraph graph into an autonomous resolution workflow: typed state, one constrained MCP tool, one human-in-the-loop checkpoint, step and token budgets, and deterministic evaluation, integrating all five platform ports. After instructor acceptance you freeze the release and, in End-to-End Delivery, connect that exact release to real AWS services in your own account, S3, SQS and Secrets Manager, call a hosted LLM API, prove it with a cloud verifier, defend it on Final Demo Day and tear it down.
The application itself keeps running on your laptop.

Project work	~70 h
Theory (optional)	~35 h
Total	~105 h

Theory topics

- State graphs, typed state, tool contracts, authority, interrupts, resume, human oversight
- Step and token budgets, auditability, deterministic evaluation, refusal, recovery
- Release provenance, image immutability, schema and configuration versioning, acceptance gates
- S3, SQS and DLQ, Secrets Manager, constrained IAM; what LocalStack can’t prove about managed AWS
- Local-to-cloud fidelity, failure evidence, cost budgets, teardown
- Hosted-model reliability and cost controls; fine-tuning and GPU serving at decision level

Skills you’ll add

LangGraph · MCP tools · Human-in-the-loop · LLM evaluation · Release provenance · AWS services (S3, SQS, Secrets Manager) · Hosted LLM APIs · Cost & teardown

Project 5

What the final system does

Coldline proposes resolutions for shipment exceptions on its own, grounds each proposal in procedures and incident history, asks a human before it acts, refuses what’s outside its authority, stays within its budgets, and runs with real AWS storage, queues and secrets and a hosted model behind it, with evaluation, telemetry and an auditable release.

Learning objectives

Bound the agent

Extend an autonomous workflow without expanding its authority beyond a written boundary.

Freeze a release

Freeze a release whose code, image, schema, configuration and evaluation set are auditable.

Connect to the cloud

Move the accepted system’s storage, queue and secrets to real AWS through adapters and configuration only.

Defend it

Defend architecture, safety, quality, reliability, cost, operations and remaining risk.

Working engineers, not career lecturers

Your instructors build production systems for a living. They meet you one-on-one every week, run your project defenses and your technical mock interviews. More experts join against the same bar.



Instructor

Mohamed Cherif

AMD · Microsoft · Amazon · Skype

GitHub · LinkedIn

Principal-level engineer with 20+ years building GPU-accelerated media pipelines, SDKs and computer vision systems.



Instructor

Ahmed Abouzeid

Amazon · Morgan Stanley · Ericsson

GitHub · LinkedIn

Engineering leader with 15+ years shipping distributed systems, platform infrastructure and GenAI-powered experiences at massive scale.



Instructor

Tejashree Khot

Amazon

GitHub · LinkedIn

Senior engineer with 8+ years building backend and distributed systems at Amazon; recently led LLM-powered API orchestration for Alexa+ as an engineering manager.



Program author

Gonen Eren

LinkedIn

Computer-vision PhD with 20+ years engineering fault-tolerant distributed systems and high-throughput data pipelines on AWS and Azure. Designed this program and the review bar your projects are held to.

Coaches who have been on the hiring side

Your career coach joins after your first project and stays with you for up to 12 months: target role, resume and LinkedIn, recruiter and behavioral mock interviews, and a plan after every real interview. Technical mocks are run by the program’s instructors.



Career coach

Monica Bencze

Google

Flatiron School

LinkedIn

Recruited for Google. Knows what a top tech company screens for.



Career coach

Francesca Ogilvie

Deloitte UK

DLA Piper

LinkedIn

Ran 40+ assessment centers at Deloitte UK. Knows how hiring decisions get made from the inside.



Career coach

Peter Spellman

Berklee College of Music

Boston University

LinkedIn

30+ years in career development. Led Berklee’s career center for 24 years.



Career coach

Daniela Ruiz

Microverse

LinkedIn

Coached 800+ software development bootcamp students toward jobs.

Your career track starts after Project 1 and runs for up to 12 months

We reposition the career capital you already have and get you ready for every interview round. Your career coach joins after your first project and stays with you for up to 12 months.

1

Positioning & target role

Your coach joins after your first project: career platform access, target role and a transition plan.

2

Portfolio & materials

Every defended project becomes a case study. Resume and LinkedIn rewritten for the target role, and tailored to each posting on the career platform.

3

Mock interviews & debriefs

Up to 10 live mocks for every round with instructors and your coach, unlimited AI practice, a plan after each mock, and debriefs of real interviews.

4

Job search & a year of support

Partner job openings and role matching on the career platform. About 4 months of active career work, then up to 8 months while you go through your transition.

A cohort of engineers at your level

You study next to working engineers from different companies and backgrounds. The median student has 5 years in the profession, and some bring 20 years or more.

How the community works

- **Community hub from day one**
  Announcements, events and a member directory with LinkedIn and GitHub. Your access stays after the program.
- **Your project space**
  You're always next to people working on the same problem. Instructors answer there between sessions.
- **Moderated networking events**
  Regular events and guest speakers from the industry.
- **Random coffee**
  We pair you with another student for a one-on-one chat.
- **TripleTen alumni**
  After your capstone you join a network of 7,500+ graduates.

Who you'll study with

Our students have worked at

Chase · Wells Fargo · PNC · KeyBank · Meta · Amazon · Apple · Intel · HP Enterprise · NASA · Northrop Grumman

Backgrounds: software engineering, DevOps and cloud, data engineering and BI, ML and AI engineering, security, QA.

Names are kept private to protect our students' confidentiality. All live activity runs in the evenings and on weekends.

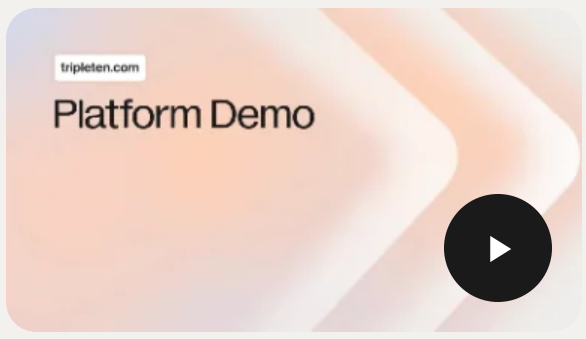
Schedule example

A typical 22 weeks

At 10 hours a week on projects. The career track runs alongside your projects: its milestones are the highlighted rows. Theory is optional and flexes with your experience; week numbers are guides, not deadlines.

Before start	Start	A call with your onboarding manager to build your personal study plan
Day 1	Start	Orientation and your first project brief. You join the community hub and meet your cohort
Weeks 1–3	Project 1	AI System Diagnostics and Capacity Review
~Week 3	Career	Your career coach joins: career platform access, target role and transition plan
Weeks 4–7	Project 2	AI Data, Retrieval, and Service Architecture
Weeks 8–11	Project 3	AI Platform Operations and Reliability
~Week 11	Career	Resume and LinkedIn rewritten for the target role; defended projects become portfolio cases
Weeks 12–15	Project 4	AI Security, Guardrails, and Governance
Weeks 12–22	Career	Mock interviews for every round: up to 10 live sessions with instructors and your coach, unlimited AI practice
Weeks 16–22	Project 5	Agentic AI, Live Evaluation, and End-to-End Delivery
Graduation	Career	Capstone defended on Final Demo Day. Career support continues for up to 12 months from the start of the track

Take a look inside the program before you commit



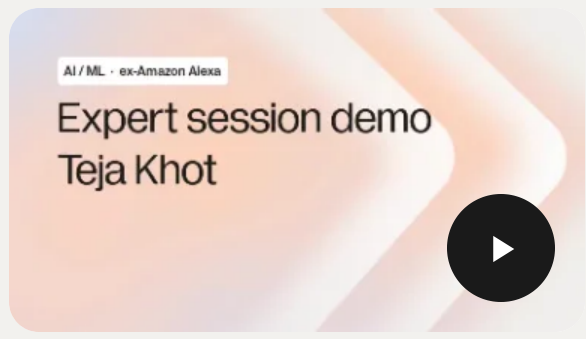
Platform demo

Three minutes inside the platform: your project map, the skill-gap check, theory you can skip, booking a 1:1, pre-defense and defense, events and the community hub.



Program walkthrough

The program's expert walks through Coldline, the five projects, how defenses work and what you finish with.



A live session

A five-minute fragment of a group office hours session with an instructor: how they work with students.

Comparing your options

Most options give you either learning or career support. This program gives you both, at a senior level.

	This program	Self-study	Online courses	Master's and university programs	Interview prep services
Duration	22 weeks	Flexible	Hours to weeks	1–3 years	3–8 months
Production projects, defended live	✓	✗	✗	✗	✗
Weekly 1:1 with a senior engineer	✓	✗	✗	✗	✗
Mock interviews	✓	✗	✗	✗	✓
Resume and LinkedIn repositioning	✓	✗	✗	✓	✓
Fits around a full-time job	✓	✓	✓	✓	✓

Answered straight

Is this live classes or self-paced?

Neither, and that’s the point. You study on your own schedule, with no mandatory class times. But it isn’t a self-paced course library: every week there’s a 30-minute 1:1 with an instructor and live group sessions, and every project ends with a live defense in front of a working engineer.

Do I have to quit my job?

No. The program is designed for people working 40 hours a week or more. There are no fixed class times, and all live sessions run in the evenings and on weekends. Plan for 10 hours a week on projects, and up to 15 if you take all the theory. At fewer hours a week, plan for longer.

Am I qualified? My Python is not strong.

The prerequisite is solid programming experience in a modern language, but not Python specifically. There’s an optional Python theory block inside the program for professionals with experience in other languages.

I’m already advanced in my career. Will this teach me anything new?

That depends entirely on where your gaps are. It is why the call starts with the syllabus and how it compares to your current skills and experience. If you partially know the material, you’ll be able to skip the theory lessons and focus on building.

Who actually teaches, a person or AI?

People. Our instructors run the one-on-one sessions and the live project reviews, and someone reviews your pull request and decides whether you are ready to defend it. AI tools are part of what you learn to build, but you’ll learn from experienced professionals.

How much instructor time do I actually get?

A weekly 30-minute 1:1, written feedback on every project, a pre-defense and a live defense per project, and your project space where instructors answer between sessions.

Will I study alone, or is there a real cohort?

You move in project cohorts grouped by the project you're on, with a project space, live events, a member directory and coffee pairings. Working engineers around you, not a beginners room.

Is TripleTen accredited?

TripleTen is licensed by the Wyoming Department of Education and is registered under California law as an out-of-state institution providing private postsecondary education. You finish with a completion certificate and a defended portfolio of production work.

The program is brand new. How do I know it works?

Yes, the program is new, and you shouldn't take our word for it. The curriculum is built and approved by engineers and hiring managers from AMD, Microsoft and Amazon. We don't accept everyone: an advisor checks your production experience on a call. Progress is measured in defended work, not watched lessons, in front of people who build these systems for a living. Check it yourself: the sample projects, the program walkthrough and this syllabus. And the first 14 days are fully refundable.

Is there a job guarantee?

The hiring decision is always the employer's, so no program can guarantee you a job, and this one has no money-back guarantee. What we commit to is everything in our control: the skills the market is hiring for right now, production projects defended in front of working engineers that show the quality of your work, and career support throughout the program, ensuring that you are up to date with the market trends.

Can I try it before I commit?

Yes. The first 14 days are fully refundable. You can start the program, see the platform and meet your instructor, and still leave at no cost if it's not what you expected.

Will I have to start my career over?

No. This program builds on the experience you already have and prepares you for your next step, not a reset. The curriculum prepares you for senior-level work, and our career team leverages your existing experience as a key qualification. While we ensure hiring managers clearly see the value of your background, final placement and title decisions ultimately rest with the individual employer.

Who this program is for

A quick self-check before you commit. Honest in both directions.

You're a good fit

- You have 2+ years of hands-on production work: backend, full-stack, frontend, mobile, DevOps, SRE, cloud or platform.
- You already use AI at work but haven’t built AI systems yet, and you feel that gap in interviews or promotion conversations.
- You have a strong tech base but under 2 years in production: a junior engineer, analyst, QC or IT specialist, or a CS degree, master’s or PhD.

This is probably not for you

- You’re starting from zero, with no coding experience yet. TripleTen’s core bootcamps are the right entry point; this program assumes a working base.
- You’re after a certificate rather than a portfolio of defended work.

Two or more years is a guide, not a cut-off. Someone hands-on in the last six months is in a stronger position than someone whose last production work was eight years ago.

What you need on day one, and what’s included

Prerequisites

Skill	What we expect
Programming	You work in a general-purpose language and can build a small application on your own. Any language works; the project code you edit is Python 3.12 with FastAPI and pytest
Command line	Find files, run scripts, read the output
Git	Clone, branch, commit, open a pull request
SQL	Basic: SELECT and JOIN
Config files	Read and edit YAML, JSON and .env files
HTTP and REST	Requests, responses, endpoints, status codes
Docker	Start an app with Docker Compose and read its container logs
AI assistant	You use ChatGPT, Claude or Copilot for engineering work

Your computer

macOS, Linux or Windows with WSL2 · 16 GB RAM minimum, 32 GB recommended · 8 CPU cores · 80 GB free disk · Docker with Compose v2 · no GPU. For the capstone you open your own AWS account for S3, SQS and Secrets Manager.

The whole platform runs on your laptop in Docker. Use a personal machine: work laptops often block installing Docker.

Included in your tuition

- A Claude Pro subscription for the length of the program, on request
- Model API usage for the projects: a program-issued allowance, no personal API account needed
- A LocalStack license for the local AWS emulation in Projects 1 to 4
- GitHub Codespaces as a fallback if your laptop can’t run the stack
- Access to your track’s course material after the program, kept up to date

Ready to engineer production systems?

TripleTen is a career learning platform
for a world being reshaped by AI.

Book a call to talk to a career mentor

tripleten