The Forward Deployed Engineer

Skills Roadmap

Skills, self-assessment and a 90-day plan for connecting AI with real business systems

A free guide from the Global Skill Development Council (GSDC)

AI models are easy to demo and hard to deploy. Most enterprise value sits inside CRMs, ERPs, billing platforms and decades-old databases, not inside a chatbot window. The engineers who close that gap are called Forward Deployed Engineers (FDEs). This roadmap shows you which skills the role needs, how to measure where you stand today, and how to build field-ready capability in 90 days.

Inside this guide:

- What an FDE does, and how the role differs from related engineering roles
- The FDE Skill Map: three layers and seven skill domains
- A self-assessment scorecard to find your gaps
- A week-by-week 90-day learning plan
- Three portfolio projects that prove your skills to employers
- How to validate your skills with the GSDC Certified Forward Deployed Engineer credential

1. What a Forward Deployed Engineer does

A Forward Deployed Engineer is the engineer who sits between the customer, the business problem and the AI solution. An FDE works inside the customer's real environment, with its real data, real systems and real constraints, rather than building in isolation. Success is measured by business outcomes, not by lines of code shipped.

Consider a logistics company that wants an AI assistant to answer "Where is my shipment?" queries. The model itself is the easy part. The FDE has to find where shipment data actually lives, maybe in a 15-year-old Oracle database, and get read-only access. They map cryptic fields like SHP_STAT_CD to plain statuses, and decide what happens when the database is down. Finally, they convince the operations team to trust the output.

How the FDE role compares

Role	Primary focus	Works closest with	Typical output

Forward Deployed Engineer	Making AI work inside a specific customer's systems and workflows	Customer business and technical teams	Deployed, integrated solution tied to a business outcome
Software Engineer	Building and maintaining product features	Product managers, other engineers	Product code, features, releases
ML Engineer	Training, tuning and serving models	Data scientists, platform teams	Models, training pipelines, inference services
Solutions Architect	Designing solutions and advising before the sale	Sales teams, customer architects	Architecture diagrams, proposals, proofs of concept

The key difference is ownership. A Solutions Architect may design the integration, and an ML Engineer may build the model. The FDE is the one

who makes it run in production at the customer site and stays until it delivers value.

A typical FDE week

An FDE's week moves through four kinds of work, often across more than one customer:

- **Discovery (Monday):** Interview the inventory team about how they currently spot low stock. Learn that they export a CSV from the ERP every morning and review it in Excel.
- **Integration (Tuesday–Wednesday):** Build a service that reads that daily export, validates product codes and loads clean records into a staging database.
- **Deployment (Thursday):** Connect an LLM-based recommender to the staging data. Ship it to five pilot users behind an approval step.
- **Iteration (Friday):** Review pilot feedback and error logs. Discover that two warehouses use a different date format, then fix the parser and redeploy.

This cycle of listening, building, shipping and fixing is the core of the role. The skills in the rest of this guide are the ones that make each step faster and safer.

2. The FDE Skill Map

FDE skills stack in three layers, and each layer depends on the one below it. A brilliant RAG pipeline is useless if the API feeding it returns bad data. A working integration fails if the business doesn't trust it.

Layer	Purpose	Skill domains
Field Skills (the multiplier)	Make it trusted and adopted	Business communication, Security awareness, Testing and monitoring
AI Implementation (the engine)	Turn data into answers	LLM APIs, RAG, Evaluation, Prompt design
Core Engineering (the foundation)	Move data reliably	API integration, Database knowledge, System architecture

- **Core Engineering (the foundation):** APIs, databases and system architecture. These skills move data reliably between

legacy systems and new applications. *Example:* pulling clean order data from a 20-year-old ERP without slowing down production.

- **AI Implementation (the engine):** LLM APIs, retrieval-augmented generation (RAG), evaluation and prompt design. These skills turn business data into answers, summaries and recommendations. *Example:* an assistant that reads that order data and explains delivery delays in plain language.
- **Field Skills (the multiplier):** Business communication, security awareness, and testing and monitoring. These skills make a solution safe, reliable and adopted. *Example:* agreeing success metrics with the operations head, masking customer data and alerting the team when the ERP feed breaks.

Most engineers arrive strong in one layer. Backend developers usually start strong in Core Engineering, data scientists in AI Implementation, and consultants in Field Skills. The FDE's advantage is being competent across all three.

3. The 7 skill domains

Each domain below follows the same structure: why it matters, what each skill level looks like, a worked example, and how to prove the skill to an employer or client.

Domain 1: API development and integration

APIs are the safest, most structured way for an AI application to talk to an existing system. Most FDE integrations start here.

- **Beginner:** Make GET and POST requests with Postman or curl. Read and write JSON. Understand HTTP status codes such as 200, 401, 404 and 500.
- **Intermediate:** Build a REST API in Python (FastAPI) or Node.js (Express). Handle API keys and OAuth 2.0. Add retries, timeouts and pagination.
- **Advanced:** Design an integration layer that exposes only the data an AI needs. Handle rate limits, webhooks and API versioning across several systems.

Example: A CRM exposes /customers/{id} with 40 fields. The FDE builds a thin wrapper endpoint that returns only name, plan and open tickets. The AI never sees payment details it doesn't need.

Proof of skill: A GitHub repo with a documented API wrapper, including error handling and an OpenAPI spec.

Domain 2: Database knowledge

Legacy business data usually lives in relational databases, and AI is only as good as the data it receives.

- **Beginner:** Write SELECT queries with JOIN, WHERE and GROUP BY. Understand tables, keys and data types.
- **Intermediate:** Use read-only database users and parameterized queries. Validate data for nulls, duplicates and invalid formats. Build field mappings between systems.
- **Advanced:** Design replica-based or change-data-capture pipelines that never load production. Optimize queries on large tables.

Example: A legacy table stores dates as 28-10-2026 while the AI service expects ISO format, 2026-10-28. The FDE adds a mapping layer that converts CUST_ID to customer_id, normalizes dates, and rejects rows with missing product codes.

Proof of skill: A SQL notebook showing data profiling, cleaning rules and a field-mapping table for a sample dataset.

Domain 3: AI application development

This is the layer that turns clean business data into answers, summaries and recommendations.

- **Beginner:** Call an LLM API, such as Claude, GPT or Gemini, with a structured prompt. Understand tokens, context windows and temperature.
- **Intermediate:** Build a retrieval-augmented generation (RAG) pipeline with embeddings and a vector store. Use tool or function calling to query live systems. Return structured JSON outputs.
- **Advanced:** Build evaluation sets to measure accuracy and hallucination rates. Compare models on cost and latency. Design fallbacks when the model is uncertain.

Example: An HR assistant answers policy questions. Instead of fine-tuning, the FDE indexes 200 policy PDFs into a vector store, retrieves the top 5 passages per question, and requires the model to cite them. An evaluation set of 50 real employee questions measures how often answers are correct.

Proof of skill: A RAG demo with a written evaluation report: questions tested, accuracy score and failure analysis.

Domain 4: System architecture

FDEs must choose the right way to connect systems that were never designed to work together.

- **Beginner:** Draw how two applications communicate. Understand client–server design, synchronous vs asynchronous calls, and batch vs real-time processing.
- **Intermediate:** Use middleware and message queues such as RabbitMQ, Kafka or AWS SQS. Build file-based pipelines for CSV and XML exports. Choose between the five integration methods: API, middleware, database, file-based and event-driven.
- **Advanced:** Design for failure with circuit breakers, dead-letter queues and idempotent retries. Plan gradual modernization without replacing the legacy core.

Example: An old billing system can't expose an API but writes an XML file every hour. The FDE builds a watcher that picks up each file, validates it,

and publishes events to a queue. The AI anomaly detector consumes events from the queue, and the billing system is never touched.

Proof of skill: An architecture diagram and design document explaining why you chose one integration method over the alternatives.

Domain 5: Security awareness

Connecting AI to business systems creates new paths to sensitive data. One careless integration can expose customer records.

- **Beginner:** Never hard-code credentials. Use environment variables. Understand the basics of authentication vs authorization.
- **Intermediate:** Apply least-privilege access. Use a secrets manager such as AWS Secrets Manager, Azure Key Vault or HashiCorp Vault. Mask personal data before it reaches the model.
- **Advanced:** Design role-aware AI access, so the assistant only shows what the asking user is allowed to see. Add audit logs and prompt-injection defenses, and align with GDPR or HIPAA where relevant.

Example: A support agent asks the AI about a customer's order. The integration checks the agent's role, redacts card numbers and logs the query. It returns only order status, not billing history.

Proof of skill: A security checklist applied to one of your projects, showing access roles, data masking rules and audit logging.

Domain 6: Business communication

FDEs succeed or fail on whether the business adopts what they build. Technical skill alone is not enough.

- **Beginner:** Ask open questions in discovery sessions. Summarize meetings into clear action items.
- **Intermediate:** Translate business pain into technical requirements. Explain trade-offs in plain language, for example "real-time costs more and adds risk; hourly refresh meets your need."
- **Advanced:** Define success metrics with stakeholders before building. Run pilots, manage expectations and present results to leadership.

Example: A finance head asks for "AI that catches fraud." Through discovery, the FDE learns the real pain is 6 hours a week spent manually reviewing duplicate invoices. They scope a focused duplicate-detection tool and agree on the target upfront: cut review time by half.

Proof of skill: A one-page discovery summary and success-metric agreement for a sample client scenario.

Domain 7: Testing and monitoring

An integration that works in a demo can fail silently in production. FDEs make failure visible.

- **Beginner:** Write unit tests for data transformations. Log errors with useful context.
- **Intermediate:** Write integration tests against mock legacy systems. Track response time, error rate and success rate on a dashboard such as Grafana or CloudWatch.
- **Advanced:** Monitor AI output quality over time. Detect schema drift when a legacy system changes its format. Set alerts and runbooks for on-call response.

Example: A legacy ERP silently adds a new column to its daily export. The FDE's schema check flags the change, pauses the pipeline and alerts the team. The AI never runs on misaligned data.

Proof of skill: A monitoring dashboard screenshot and a short runbook describing what to do when each alert fires.

4. Self-assessment scorecard

Rate yourself honestly from 1 to 5 in each domain, then add up your total. Base your rating on what you have actually built or done, not what you have read about.

Rating guide:

- **1:** No hands-on experience
- **2:** Followed tutorials or courses
- **3:** Built something working on my own (Beginner level complete)
- **4:** Used it in a real project at work (Intermediate level complete)
- **5:** Designed or led it in production and can teach others (Advanced level)

Skill domain	Your rating (1–5)	Evidence (what you built or did)
1. API development and integration		
2. Database knowledge		
3. AI application development		
4. System architecture		
5. Security awareness		
6. Business communication		
7. Testing and monitoring		
Total (out of 35)		

What your score means

Score	Stage	What it means	Focus next

7–15	Foundation stage	You have early exposure but limited hands-on work	Start at Day 1 of the 90-day plan. Prioritize APIs and SQL.
16–27	Developing FDE	You can build parts of an integration but not yet own one end to end	Target your two lowest domains. Complete one full portfolio project.
28–35	Field-ready	You can own an AI integration from discovery to production	Validate your skills with certification and lead a real deployment.

Example: Priya is a DevOps engineer. She scores 4 in APIs, 3 in databases, 2 in AI development, 5 in architecture, 4 in security, 2 in communication and 5 in monitoring, for a total of 25. That places her in the Developing FDE stage. Her focus areas are AI application development and business communication. She should spend Days 31–60 on RAG and evaluation, and practice running discovery sessions.

Your focus areas

- ☐My lowest-scoring domain: _____
- ☐My second-lowest domain: _____

- ☐One project I will build to close these gaps: _____
- ☐My target score in 90 days: ____ / 35

5. 90-day learning plan

The plan assumes 8–10 hours a week. Each phase ends with a deliverable you can add to your portfolio. If you scored above 27, you can skip ahead to the phase covering your weakest domains.

Days 1–30: Foundations

The goal is to move data reliably between systems. Every later skill depends on this.

- **Week 1 (APIs):** Learn HTTP methods, status codes and JSON. Call 3 public APIs with Postman, for example a weather API, GitHub and a currency API.
- **Week 2 (Building APIs):** Build a small FastAPI or Express service with 3 endpoints, API-key authentication and error responses.

- **Week 3 (SQL):** Practice SELECT, JOIN and GROUP BY on a sample dataset such as the Northwind database. Create a read-only database user.
- **Week 4 (Data quality):** Profile a messy CSV for nulls, duplicates and mixed date formats. Write Python cleaning and mapping rules.

Deliverable: An API that serves cleaned customer data from a SQL database, with a README explaining your field mappings.

Days 31–60: The AI layer

The goal is to connect a language model to real data safely and measure how well it performs.

- **Week 5 (LLM APIs):** Call an LLM API with system prompts and structured JSON output. Learn how tokens and context limits affect cost.
- **Week 6 (RAG):** Build a retrieval pipeline using embeddings and a vector store such as Chroma, pgvector or Pinecone over 20–50 documents.

- **Week 7 (Tool calling):** Let the model call your Phase 1 API to fetch live data, for example “What is the status of order 1042?”
- **Week 8 (Evaluation):** Write 30 test questions with expected answers. Measure accuracy and log every failure with its cause.

Deliverable: An AI assistant that answers questions from your sample database, with an evaluation report showing its accuracy score.

Days 61–90: Field readiness

The goal is to make your solution safe, observable and ready for real users.

- **Week 9 (Security):** Move secrets to a vault or environment manager. Add role-based access and mask personal data before it reaches the model.
- **Week 10 (Guardrails):** Add a human approval step for sensitive actions. Make sure the AI reports “data unavailable” instead of guessing when the source system fails.
- **Week 11 (Monitoring):** Add logging, a metrics dashboard and alerts for errors, latency and schema changes.

- **Week 12 (Communication):** Write a one-page stakeholder summary of the problem, solution, metrics and risks. Record a 5-minute demo walkthrough.

Deliverable: An end-to-end capstone that is secured, monitored and documented, with a stakeholder-ready demo.

Day 90 checkpoint: Retake the self-assessment scorecard. Compare your new total with your Day 1 score to see which domains improved and where to focus next.

6. Portfolio projects

Employers hiring FDEs want proof that you can connect AI to messy, real-world systems. These three projects mirror real field work. Each one is small enough to finish in 2–4 weeks.

Project	Difficulty	Time	Main domains
AI order-status assistant	Beginner	2 weeks	APIs, databases, AI development

CSV-to-AI pipeline	Intermediate	3 weeks	Databases, architecture, testing
Inventory replenishment recommender	Advanced	4 weeks	All 7 domains

Project 1: AI order-status assistant

A customer asks “Where is my order?” and the assistant answers from a mock legacy database instead of guessing.

- **Scope:** Create a SQLite or PostgreSQL database with 500 sample orders, using legacy-style field names such as ORD_NO and STAT_CD. Wrap it in a read-only API. Connect an LLM through tool calling so it looks up the order before replying.
- **Must include:** A clear “I couldn’t retrieve that order” response when the lookup fails, and status-code mapping, for example STAT_CD = 3 becomes “Shipped.”
- **Skills shown:** API integration, SQL, field mapping, tool calling, safe failure handling.

Project 2: CSV-to-AI pipeline

Many legacy systems only export daily files. This project shows you can turn those files into something AI can use reliably.

- **Scope:** Simulate a nightly CSV export with deliberate problems: mixed date formats, duplicate rows and missing IDs. Build a pipeline that validates, cleans and loads the data, then have an LLM generate a daily summary of key changes.
- **Must include:** A rejected-rows report, a schema check that stops the pipeline if columns change, and unit tests for every cleaning rule.
- **Skills shown:** File-based integration, data validation, automation (cron, n8n or Airflow), testing, monitoring.

Project 3: Inventory replenishment recommender

This is a full-scale version of the case study from the blog post and makes a strong capstone.

- **Scope:** Use mock ERP data covering stock levels, 90 days of sales and supplier lead times. Build an integration service and an AI layer that flags products likely to run out within 7 days. Show recommendations on a simple dashboard.

- **Must include:** A human approval step, with no automatic purchase orders. Add role-based access so only managers see supplier costs, plus an error and latency dashboard and an evaluation of recommendation accuracy against past stockouts.
- **Skills shown:** All seven domains, including a one-page stakeholder summary presenting the business case.

Tip: For each project, write a short README covering the problem, architecture diagram, integration method chosen and why, how failures are handled, and results. Hiring managers read the README before the code.

7. Validate your skills

A portfolio shows what you have built. A certification shows employers and clients that your skills meet an independent standard. The GSDC Certified Forward Deployed Engineer credential is designed for professionals who connect AI solutions with real business systems and workflows, the same work this roadmap prepares you for.

How the certification fits this roadmap

- **If you scored 7–15:** Use the 90-day plan to build your foundations first. Then use the certification as your structured next step.
- **If you scored 16–27:** The certification helps you close gaps in your weakest domains and adds formal recognition to your portfolio.
- **If you scored 28–35:** You are ready to validate your experience. Certification strengthens your profile for FDE, AI integration and solutions engineering roles.

Who it's for

- Software and backend developers moving into AI implementation
- DevOps, cloud and automation engineers who already integrate systems
- Solutions and sales engineers who want to own delivery, not just design
- Technical consultants leading AI adoption for clients

Take the next step

Explore the certification: GSDC Certified Forward Deployed Engineer

Download the certification brochure to see the full curriculum, exam format and eligibility.

GSDC
Global Skill Development Council

CERTIFIED FORWARD DEPLOYED ENGINEER

LEARN DIRECTLY FROM INDUSTRY EXPERTS, AI PRACTITIONERS, AND ENGINEERING LEADERS WHO ARE BUILDING AND DEPLOYING SCALABLE AI DRIVEN SOLUTIONS FOR ENTERPRISES WORLDWIDE.



ABOUT GSDC CERTIFICATION

EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.

CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.

EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.

LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

LEARNING OBJECTIVE

- Validate your Forward Deployed Engineering expertise.
- Increase your earning potential with an in-demand credential.
- Get complimentary career assistance after certification.

Enroll now with the code **LEARN20** To avail **20%** discount

Enroll Now

www.gsdcouncil.org