

Forward Deployed Engineer Skills Checklist: Assess Your Readiness

*A self-assessment for aspiring forward deployed engineers across
programming, cloud, AI, and communication.*

Introduction: Why This Checklist Exists

A forward deployed engineer (FDE) is a technical professional who works directly with customers or business teams to understand their problems, build software solutions, and make those solutions work in real environments. That means the job asks for two things at once: strong technical ability and strong customer-facing ability.

Most people preparing for this career focus on one side. A developer might spend months on Python and cloud tools but never practise explaining a technical decision to a non-technical manager. Someone with excellent communication skills might never have deployed an application. This checklist helps you see both sides clearly, so you can spot gaps before an employer does.

The checklist follows the same skill areas covered in the blog *7 Steps to Become a Forward Deployed Engineer*, so you can use the two together: read a step in the blog, then rate yourself on the matching section here.

What you will get from this checklist:

- A clear picture of your current strengths across seven skill areas
- A score for each area and an overall readiness level
- A shortlist of your three weakest areas to work on first
- A simple 30-60-90 day plan to close those gaps

How to Use This Checklist

Work through the document in order. It takes about 20 to 30 minutes if you answer honestly.

1. **Read each skill and its description.** The description tells you what “having this skill” actually looks like, so you are not guessing.
2. **Rate yourself from 0 to 3** using the scale below.
3. **Add a note** if you want to remember an example, a project, or something you need to revisit.
4. **Total your score** at the end of each section.
5. **Use the Score Summary** to find your readiness level and your three biggest gaps.
6. **Fill in the Action Plan** so the results turn into next steps.

The Rating Scale

0	Not started	You have not studied or used this skill yet.
1	Learning	You understand the concept but have only followed tutorials.
2	Can do with guidance	You can complete tasks but need documentation, examples, or help.
3	Can do independently	You can do it on your own, fix problems, and explain what you did.

Example: Take the skill “Working with APIs.” If you have watched a video on REST APIs but never called one yourself, you score a 1. If you have built a small script that pulls

data from a public API, but you needed a tutorial open the whole time, you score a 2. If you have connected two services using an API, handled errors, and could explain it to a teammate, you score a 3.

A note on honesty: Rating yourself a 3 on everything will not help you. A lower score is not a failure. It is simply a map of where to spend your time.

Section 1: Programming

Programming is the starting point for this career. Forward deployed engineers write code to build integrations, automate tasks, and adapt software to a customer's needs. You do not need to know many languages. You need to know one well enough to solve real problems, with Python and JavaScript being the most common choices.

Why it matters in the role: Imagine a logistics company that exports shipment data as messy spreadsheets every day. As an FDE, you might write a short Python script that cleans the data, removes duplicates, and sends a summary to the operations team. That one script saves hours of manual work, and it only exists because you can program.

Python or JavaScript fundamentals	You can write programs in at least one language without constantly looking up basic syntax.	—	
Variables, loops, functions, and data structures	You can use lists, dictionaries, and functions to organise logic and handle data.	—	

Object-oriented programming	You can create classes and objects and know when this approach makes code easier to maintain.	—	
Error handling and debugging	You can read an error message, find the cause, and fix it using logs or a debugger.	—	
Writing clean, readable code	Others can read your code, with clear names, simple structure, and helpful comments.	—	

Section 1 score: — / 15

If your score is low: Pick one language and build three small programs, such as a to-do list tool, a file organiser, and a script that reads a CSV file and prints a summary. Practise a little every day rather than in long, rare sessions.

Section 2: Software Development

Fundamentals

A forward deployed software engineer needs more than coding ability. You must understand how applications are built, connected, tested, and shared with other developers. Customers rarely want a standalone program. They want something that fits into the systems they already use.

Why it matters in the role: Suppose a company wants its customer relationship management (CRM) system connected to an AI assistant. You would use an API to

exchange information, a database query to fetch customer records, backend logic to process requests, and Git to track your changes as the project evolves.

Frontend and backend basics	You understand how the user-facing side and server side of an application work together.	—	
Working with APIs	You can send requests, read responses, handle authentication, and deal with errors.	—	
SQL and databases	You can write queries to retrieve, filter, join, and update data.	—	
Git and GitHub	You can commit, branch, merge, and share code, and you can recover from mistakes.	—	
Basic software testing	You can write simple tests to confirm that your code behaves as expected.	—	

Section 2 score: — / 15

If your score is low: Build a small project that has a backend, a database, and one external API. For example, a weather dashboard that stores search history in a database. Keep the code on GitHub and write a short README.

Section 3: Cloud Computing and Deployment

Forward deployed engineers often work inside a customer’s existing technical environment. Your solution has to run somewhere, and that “somewhere” is frequently a cloud platform such as AWS, Microsoft Azure, or Google Cloud. You do not need to be a cloud architect, but you do need to understand how an application moves from your laptop to production.

Why it matters in the role: A customer’s security team might require that your application runs in a container, reads secrets from environment variables, and logs errors to a monitoring tool. If you have never done any of this, a deployment that should take an afternoon can take weeks.

Cloud basics (AWS, Azure, or Google Cloud)	You understand virtual machines, storage, networking, permissions, and application hosting.	—	
Docker and containerisation	You can package an application in a container and run it consistently.	—	
Linux command-line basics	You can navigate files, manage processes, and read logs from the terminal.	—	
Environment variables and configuration	You can keep settings and secrets out of your code and change them per environment.	—	
CI/CD fundamentals	You understand how code is automatically tested and deployed after changes.	—	

Application monitoring and troubleshooting	You can use logs and basic metrics to find why an application is failing.	—	
--	---	---	--

Section 3 score: __ / 18

If your score is low: Deploy one of your Section 2 projects to a cloud platform using Docker. Then break it on purpose, such as by using a wrong environment variable, and practise finding the problem from the logs.

Section 4: Data and AI

Businesses increasingly use AI to automate tasks, analyse information, and support decisions. An AI forward deployed engineer may help an organisation connect its data to a language model, build a retrieval-augmented generation (RAG) application, or add an AI agent to an existing workflow. You do not need to be a researcher. Practical skill in integrating and deploying AI is a strong starting point.

Why it matters in the role: Imagine a company that wants an assistant that answers employee questions from internal policy documents. You would connect the document repository, retrieve the relevant passages, send them to a large language model (LLM), and test whether the answers are accurate and useful. You would also need to make sure the assistant does not reveal documents that a particular employee is not allowed to see.

Data handling with Python and SQL	You can clean, transform, and analyse data from files and databases.	—	
Machine learning fundamentals	You understand training data, models, evaluation, and common limitations.	—	
LLMs and prompt engineering	You can write clear prompts and understand how model behaviour changes with instructions.	—	
AI APIs and integrations	You can call an AI service from code and handle its responses.	—	
RAG applications	You understand how documents are indexed, retrieved, and passed to a model as context.	—	
Data privacy, access controls, and responsible AI	You know how to protect sensitive data and are aware of model limits such as incorrect answers.	—	

Section 4 score: — / 18

If your score is low: Build a small chatbot that answers questions from a set of PDFs or text files. Test it with ten questions, record which answers were wrong, and write down why you think they failed.

Section 5: Problem-Solving

Technical knowledge only helps if you can apply it to messy, real situations. Customers rarely describe problems in neat technical terms. They say things like “our reports take too long” or “the team keeps entering the same data twice.” Your job is to turn that into something you can solve.

Why it matters in the role: If a customer says “the dashboard is slow,” a strong problem-solver does not immediately start rewriting code. They check whether the slowness comes from the database query, the network, the amount of data, or the dashboard itself. They test one possibility at a time and confirm the cause before making changes.

Debugging and root-cause analysis	You look for the underlying cause instead of fixing only the visible symptom.	—	
Breaking complex problems into smaller tasks	You can split a large goal into steps you can finish and test.	—	
Designing practical solutions	You choose the simplest approach that meets the need, not the most impressive one.	—	
Adapting to changing requirements	You stay productive when the customer changes priorities or adds new constraints.	—	

Section 5 score: ___ / 12

If your score is low: Whenever something breaks, write down three possible causes before you touch anything. Then test them one by one and record what you found. This habit alone improves problem-solving quickly.

Section 6: Communication and Customer Skills

This is the section that separates a forward deployed engineer from many other engineering roles. A technically impressive solution can still fail if it does not address what the customer actually needs. You will spend time in meetings, workshops, and calls, and you will often be the person translating between technical and non-technical people.

Why it matters in the role: Suppose a customer asks for “an AI tool for support.” A good FDE asks follow-up questions: What kinds of tickets does the team handle? Which ones take the most time? What would a good result look like? After a few questions, the real need may turn out to be “automatically suggest answers for the five most common ticket types,” which is a much smaller and more achievable project.

Asking clear requirement-gathering questions	You can uncover the real problem, constraints, and success criteria.	—	

Explaining technical ideas to non-technical people	You can describe your approach in plain language without losing accuracy.	—	
Documentation	You can write instructions, notes, and handover documents others can follow.	—	
Presenting solutions and collecting feedback	You can demo your work, listen to reactions, and turn feedback into changes.	—	
Cross-functional collaboration	You can work with product managers, data scientists, developers, and customer teams.	—	

Section 6 score: __ / 15

If your score is low: Practise explaining one of your projects to a friend who is not technical, in under two minutes. If they cannot repeat back what it does and why it matters, simplify your explanation and try again.

Section 7: Business Understanding

Forward deployed engineers are judged by results, not just by working code.

Understanding how a customer's business operates helps you decide what to build, what to skip, and how to show that your solution was worth the effort.

Why it matters in the role: If a finance team spends 20 hours a month preparing a report, and your automated tool brings that down to two hours, you have a clear, measurable result you can share. Without that understanding, you may build something clever that nobody uses.

Understanding customer workflows	You can map how work currently flows through a team and where it slows down.	—	
Linking solutions to business outcomes and ROI	You can explain how your solution saves time, reduces cost, or improves quality.	—	
Measuring whether a solution worked	You can define success measures before building and check them afterwards.	—	

Section 7 score: — / 9

If your score is low: For each project in your portfolio, add a short “impact” statement with a number, even if it is an estimate. For example: “Reduced manual report preparation from about 3 hours to 20 minutes.”

Score Summary and Interpretation

Transfer your section scores into the table below and add them up.

1. Programming	—	15
2. Software Development Fundamentals	—	15

3. Cloud Computing and Deployment	—	18
4. Data and AI	—	18
5. Problem-Solving	—	12
6. Communication and Customer Skills	—	15
7. Business Understanding	—	9
Total	** — **	102

Your Readiness Level

- **0 to 34 (Beginner):** You are early in your journey. Focus on programming and software development fundamentals first (Sections 1 and 2). Build small projects and practise regularly. Do not rush into cloud or AI until you are comfortable writing and debugging code.
- **35 to 68 (Developing):** You have a solid base and some gaps. Strengthen cloud, AI, and communication skills (Sections 3, 4, and 6), and start building portfolio projects that solve a business problem from beginning to end.
- **69 to 102 (Job-ready):** You are in a good position to start applying. Polish your portfolio, practise interview scenarios, and review job descriptions for forward deployed engineer, solutions engineer, AI engineer, and implementation engineer roles.

Remember: These ranges are a guide, not a verdict. A person who scores 60 overall but has strong projects and good communication may be more ready than someone who scores 75 on paper but has never built anything real.

Skill Gap Analysis

Look at your section scores and find the three areas where you scored lowest as a share of the maximum. Write them here:

1			
2			
3			

Example: A learner scores 11 out of 18 in Cloud, 7 out of 18 in Data and AI, and 14 out of 15 in Programming. Their top priority is Data and AI, so their action might be “Build a document question-answering chatbot using a public dataset.”

Action Plan and Next Steps

A score on its own changes nothing. Turn it into a plan.

30-60-90 Day Learning Plan

Use your three priority areas to fill in the template below. Keep goals small and specific.

First 30 days				☐
Days 31 to 60				☐
Days 61 to 90				☐

Example plan for someone strong in programming but weak in cloud and AI:

- **First 30 days:** Learn Docker basics and containerise an existing project.
- **Days 31 to 60:** Deploy the container on a cloud platform and set up basic logging.
- **Days 61 to 90:** Build a RAG chatbot over a set of documents and document the results.

Portfolio Checklist

Employers want proof that you can solve real problems. Before you apply, check that you have:

- ☐ At least one project that solves a specific business problem, not just a coding exercise
- ☐ A GitHub repository with clean code and a clear README

- ☐ A short write-up for each project covering the problem, your approach, the tools used, the challenges, and the result
- ☐ At least one project that connects two systems or uses an API
- ☐ At least one project that uses data or AI

Example project ideas:

- An AI chatbot that answers questions from company documents
- A dashboard that combines data from multiple APIs
- An automated report-generation tool
- A customer support workflow that uses AI and business data

Interview Readiness Checklist

- ☐ I can solve basic coding problems and explain my reasoning aloud
- ☐ I can describe how I would integrate with an API and handle failures
- ☐ I understand system design fundamentals at a basic level
- ☐ I can walk through how I would troubleshoot a failing deployment
- ☐ I can explain how I would handle an unclear customer requirement
- ☐ I can describe how I would measure whether a solution was successful
- ☐ I have researched the salary range for my location, experience level, and specialisation

Recommended Next Steps

1. Revisit the blog *7 Steps to Become a Forward Deployed Engineer* and read the steps that match your lowest-scoring sections.
2. Retake this checklist after 90 days to measure your progress.
3. Start applying for roles once your total score and portfolio show you are ready.

Closing: Support Your Learning

Self-assessment shows you where you stand. Structured learning can help you close the gaps faster, especially when it includes hands-on practice and a clear understanding of how forward deployed engineering connects technology with business needs.

The **GSDC Forward Deployed Engineer Certification** can support your preparation by helping you understand the role, the importance of customer collaboration, and the practical skills needed to connect technology with business requirements. It works best when combined with hands-on projects, programming practice, and cloud or AI implementation experience.

Explore the GSDC Forward Deployed Engineer Certification: [add certification page link here]

Read the full blog: [add blog link here]

Website and contact details: [add website and contact details here]

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.



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.



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.gsdccouncil.org