

What to Learn Now vs What Can Wait, A DevOps / DevSecOps Priority Guide

Ten topics, sorted honestly into what actually needs your attention this month and what can genuinely wait.

SWQ IT Academy trains working IT professionals with a simple standard: skills you can actually demonstrate, not skills you have merely watched someone else use. Before that standard means anything, you need to know what deserves your limited hours first.

Built for working IT professionals who already have a roadmap or a project list, but keep feeling behind because everything seems equally urgent. It usually isn't, and this guide is here to say so plainly.



Not everything on the list deserves this week's hours.

Open ten different DevOps learning threads and you'll find ten different opinions on what to learn first, usually delivered with equal confidence. Docker before Terraform. No, Terraform first. Kubernetes is essential. No, Kubernetes is overrated, learn observability instead. Every voice sounds certain, and none of them agree.

That's the actual source of most of the overwhelm we see. It's rarely that the material is too hard. It's that professionals are trying to give every topic the same weight and the same urgency, when in practice they don't carry either.

Some topics are load bearing, almost nothing else works without them. Some are genuinely valuable but only once the load bearing pieces are solid. And some are worth knowing about in passing, without needing real depth, until a specific job or project actually calls for them.

THE HONEST VERSION

You can't tell the difference between foundational and later stage just by how often a topic gets mentioned online. Kubernetes gets mentioned constantly and still isn't where most people should start. This guide is our attempt to say plainly what we'd tell you if you asked us directly.

What follows is a straightforward triage, ten common DevOps and DevSecOps topics, sorted into what needs your attention now, what's worth knowing is coming, and what can genuinely sit untouched for a few months without costing you anything.

Foundational, important, and later stage

Three tiers, used consistently through the rest of this guide.

LEARN NOW

Load bearing. Almost everything else in this field assumes you already have this, and skipping it makes every later topic harder than it needs to be.

IMPORTANT, NOT URGENT

Genuinely necessary for the role, but it lands better once the foundational layer is solid. Learning it too early usually means relearning it properly later anyway.

CAN WAIT

Worth knowing exists, not worth deep study yet. Most people pick these up naturally once a real job or project actually calls for them.

These tiers aren't a judgment on how interesting or important a topic is in the abstract, cloud security matters enormously in a real DevSecOps role. They're a judgment on sequencing, on what earns your limited hours this month specifically, given everything else you haven't built yet.

Priority table, ten topics

This is the default order for someone without a specific specialisation decided yet. If you already know you're headed toward DevSecOps specifically, cloud security and DevSecOps controls move up, see the background specific guidance right after this table.

TOPIC	TIER	WHY THIS TIER	REVISIT PROPERLY WHEN
Linux, networking, Git	Learn now	Every other topic on this list assumes command line comfort and a real Git habit underneath it	Not applicable, this stays foundational throughout your career
Docker	Learn now	The packaging format almost everything else on this list is built to work with	Deepen once you're building multi service, production style setups
CI/CD	Learn now	The automation mindset this entire field is organised around, and one of the most interview tested skills	Deepen with multi environment and approval gated pipelines once employed
Terraform	Important, not urgent	Genuinely core, but lands far better once Docker and basic CI/CD are already solid	Right after your first working CI/CD pipeline exists
Kubernetes	Important, not urgent	Where containers, infrastructure, and pipelines meet, genuinely hard without those three first	Once you have a container and a pipeline you understand end to end
Cloud basics	Important, not urgent	Necessary for Terraform and Kubernetes to mean anything real, but doesn't need to come first	Alongside Terraform, learned through using it rather than studying it separately
Cloud security	Can wait	Matters a great deal eventually, but reviewing security on infrastructure you haven't built yet teaches very little	Once you have real infrastructure of your own to audit

DevSecOps controls	Can wait	Needs a working pipeline to attach to, learning it in isolation produces theory without practice	Once you have a CI/CD pipeline that's actually yours
Monitoring and observability	Can wait	There's nothing meaningful to monitor until you have a real deployment running somewhere	Once something you built is actually live and could actually break
Helm, ArgoCD, advanced tooling	Can wait	These are refinements on top of Kubernetes fundamentals, not a substitute for having them	Once plain Kubernetes manifests genuinely feel repetitive to you

ADJUST FOR YOUR SITUATION

Learn now, learn later, by background

The default table above shifts depending on what you already bring in. Here's the adjustment for three common starting points.

DEVELOPERS

LEARN NOW	LEARN LATER
Docker, and a basic CI/CD pipeline. Both build directly on skills you already have.	Terraform and Kubernetes. Genuinely important, but they'll land better once Docker and CI/CD are second nature.

INFRA AND SUPPORT PROFESSIONALS

LEARN NOW	LEARN LATER
Git and scripting, then Docker, then Terraform. Your infrastructure instincts make Terraform land faster than it does for most people.	Kubernetes and cloud security depth. Valuable, but CI/CD is the more urgent gap for this background and deserves the earlier slot.

SECURITY PROFESSIONALS

LEARN NOW	LEARN LATER
Linux, Git, scripting, and Docker. The hands on engineering base that's usually the real gap for this background.	Deep Kubernetes administration, monitoring, and advanced tooling. Cloud security and DevSecOps controls move up sooner for you than for other backgrounds, right after a working CI/CD pipeline exists.

LOOKING FOR THE FULL SEQUENCE, NOT JUST NOW VERSUS LATER

SWQ IT Academy's Role Based DevOps / DevSecOps Learning Paths resource covers the complete order, projects, and 60 to 90 day plan for each of these backgrounds in detail. This guide is the fast triage, that one is the full path.

Common mistakes in topic prioritisation

Five ways a reasonable person ends up with the wrong topic open on a Tuesday evening.

- ✗ **Trying to learn all ten topics at once.** Splitting eight hours a week across ten topics gives you ten shallow exposures and zero real skills. Depth in three beats breadth across ten every time.

- ✗ **Jumping to Kubernetes or advanced tooling because it's trendy.** Popularity in job postings and readiness to learn something are two different questions, and only one of them is about you right now.

- ✗ **Treating cloud security or DevSecOps as urgent before the fundamentals exist.** Security work needs something real to secure. Studying it in a vacuum produces vocabulary, not a demonstrable skill.

- ✗ **Chasing monitoring and observability tools before there's anything to monitor.** A dashboard watching nothing real teaches you the tool's interface and very little else.

- ✗ **Confusing important with urgent.** Terraform is important. It is not this week's most urgent topic if you haven't containerised anything yet. Both things can be true at once.

A 12 week suggested topic order

One reasonable way to sequence these ten topics across twelve weeks, roughly eight to ten hours a week.

WEEKS	TOPIC	WHAT "ENOUGH" LOOKS LIKE
1 to 2	Linux, networking, Git	Comfortable on the command line, a real branch and pull request habit
3 to 4	Docker	Can containerise an application from scratch without a reference open
5 to 6	CI/CD	A working pipeline that tests and builds automatically on every push
7 to 8	Terraform, and cloud basics alongside it	Can provision and cleanly destroy real infrastructure through code
9 to 10	Kubernetes	Can deploy and troubleshoot a real workload on a local cluster
11	Cloud security, introductory	Can review your own infrastructure and spot at least three real issues
12	DevSecOps controls, introductory, plus a first look at monitoring	A scanning step added to your pipeline, and a basic sense of what you'd want to monitor if this were live

WHERE HELM AND ARGOCD FIT

Deliberately absent from these twelve weeks. They're worth a look once plain Kubernetes manifests start to feel repetitive, which for most people is closer to month four or five than week twelve.

A general priority order gets you most of the way. Your situation gets you the rest.

This guide covers what's true for most working professionals most of the time. It can't know which topics your current job has already half taught you, or which ones a specific job posting you're eyeing actually needs first. That's worth a direct question rather than a guess.

Ask us to sanity check your list

Tell us what you're currently learning and we'll tell you honestly if the order makes sense, or if something further down this list deserves to move up for your situation.

Get the full role based path

This guide is the fast triage. For the complete sequence, projects, and a 60 to 90 day plan built around your specific background, ask for SWQ IT Academy's Role Based Learning Paths resource.

Join the SWQ IT Academy community

Compare notes with other working professionals figuring out the same tradeoffs, and stay honest about what's actually urgent versus what just feels urgent.



A part of the ENGITRIX group

Email: itacademy@swqgroup.com

WhatsApp and calls: +91 8123999170

Web: itacademy.swqgroup.com

We build industry ready professionals. The job is yours to earn, we make sure you are fully equipped to earn it.