

SWQ IT ACADEMY LEAD MAGNET

DevOps / DevSecOps Roadmap for Working Professionals

*A practical roadmap to move into DevOps or DevSecOps
without trying to learn everything at once.*

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. This roadmap follows that same standard, one stage at a time.

Written for working IT professionals only, developers, admins, support engineers, QA engineers, cloud engineers, and security professionals who already have an IT background. If you are a college fresher with no prior IT experience, this is not the roadmap for you yet, and we will tell you why in the final section.



A part of the ENGITRIX group

You don't have a motivation problem. You have a sequencing problem.

Here's what we see most often. A professional with real IT experience decides to move into DevOps. They open five browser tabs, start three courses, and within a month they know a little about Docker, a little about Kubernetes, and a little about Terraform, in no particular order and with no idea which one to actually get good at first.

That's not a discipline problem. It's what happens when a genuinely broad field gets handed to you without a sequence attached.

DevOps and DevSecOps keep expanding as fields, and most content treats every tool as equally urgent. It isn't. Some skills are load bearing, everything else depends on them. Others are situational, useful once you're already working in the role, not before. Nobody tells you which is which, so you end up spreading a limited number of evening hours across everything at once, and finishing nothing.

A STRAIGHT ANSWER

If you take one thing from this roadmap, take this: depth in the right order beats breadth in the wrong order, every time we've seen it play out in an interview.

This roadmap gives you that order. Seven stages, each one building on the last, each one telling you plainly what to focus on now and what to leave for later. Every stage ends in something you build, not just something you understand, because a concept you can only describe is not the same as a concept you can demonstrate under interview pressure.

Who this roadmap is for

Built for people who already work in IT and want to move sideways or upward into DevOps or DevSecOps. Not a starting point for someone new to the industry.

Developers

You ship code but want ownership of how it gets built, tested, and deployed, not just the writing of it.

Linux, system, and infrastructure professionals

You already manage servers. You want to move from manual operations to automation and modern infrastructure practice.

Support and operations engineers

You understand production and incidents. You want to build and automate the systems you currently maintain by hand.

QA, release, and build engineers

You're already close to the delivery pipeline. You want to formalise that exposure into real CI/CD skill.

Security professionals moving toward DevSecOps

You understand risk. You want to apply it inside the pipeline instead of reviewing systems after the fact.

Cloud professionals wanting more depth

You know your way around a console. You want the automation and delivery layers built on top of that.

How to use this roadmap

Five working principles. Read them once now, then come back to this page any time you feel the pull to skip ahead.

01 Follow the order, even when a later stage looks more exciting

Kubernetes and cloud security are the two stages professionals want to rush to. Both become genuinely hard without what comes before them, and that difficulty is avoidable, not inherent.

02 One layer at a time

Studying containers, infrastructure, and pipelines in the same week feels productive and produces shallow knowledge in all three. Finish a stage before starting the next one.

03 One tool per stage, not five

Every stage has a recommended tool. Get properly comfortable with it before you go looking at alternatives. You can always add a second tool once the first one is second nature.

04 Every stage ends with something built, not just something understood

The mini project at the end of each stage is not optional homework, it's the actual proof of the stage. Treat it that way.

05 Ask three questions of every topic

What problem does this solve, how do I build it myself, and how does this show up in a real job. If you can only answer the first one, you're not finished with the topic yet.

The roadmap at a glance

Seven stages, in the order they're actually used together on the job. Full detail for each stage follows in the next section, this page is your map.

STAGE	WHAT IT'S REALLY ABOUT	LEARN NOW	CAN WAIT	EFFORT
1. Foundations	The operating layer everything else sits on	Linux, Git, basic networking, shell scripting	Kernel internals, protocol theory	2 to 3 weeks
2. Docker and Containers	Packaging an application so it runs the same everywhere	Dockerfiles, Docker Compose, image basics	Kubernetes, orchestration at scale	2 weeks
3. Terraform and IaC	Defining infrastructure in code instead of clicking it together	Terraform syntax, state, one cloud provider	Multi cloud design, advanced modules	3 weeks
4. CI/CD with GitHub Actions	Automating the path from commit to deployment	Workflows, build and test automation, secrets	Jenkins, GitLab, enterprise CI/CD	2 to 3 weeks
5. Kubernetes	Where containers, infrastructure, and pipelines meet	Pods, deployments, services, kubectl	Custom controllers, service mesh	4 weeks
6. Cloud Security	The security baseline every DevOps role now expects	IAM, security groups, secrets management	Penetration testing, compliance depth	2 to 3 weeks
7. DevSecOps	Putting security checks inside the pipeline itself	Image scanning, dependency scanning	Building custom security tooling	3 weeks

TOTAL REALISTIC TIMELINE

Around eighteen to twenty weeks of consistent, part time evening and weekend study, roughly the length of the ninety day plan later in this document, stretched slightly to account for a real life happening around it.

Stage by stage, in detail

This is where the actual work happens. Each stage below tells you what to build, what mistake to avoid, and what "ready for the next stage" genuinely looks like.

Foundations

Everything past this point assumes you're already comfortable on the command line. Skip this stage and you'll spend the next six months quietly struggling with things that have nothing to do with the stage you're actually on.

LEARN NOW	CAN WAIT
Linux file systems, permissions, and process management	Kernel internals and deep protocol theory
Basic networking: DNS, HTTP, ports, firewalls	Advanced routing and network engineering
Git branching, merging, and a real GitHub workflow	Git internals and advanced rebase workflows
Enough Bash or Python to automate a repetitive task	Any orchestration tool

MINI PROJECT

Find a task you currently do manually at work or at home, write a script that automates it, and push it to GitHub with a README that explains what it does and why you built it.

COMMON MISTAKE

Using Git only to upload finished code instead of practising the branch and merge workflow that actually gets tested in interviews.

READY FOR STAGE 2 WHEN

You can solve a basic Linux task without searching for the exact command, and explain your Git workflow out loud without hesitating.

Outcome: comfortable in Linux, fluent in Git, and no longer doing by hand what a script could do for you.

Docker and Containers

Containers are the packaging format that Terraform, CI/CD, and Kubernetes all assume you already understand. This is a short stage, but skipping it makes Kubernetes needlessly painful later.

LEARN NOW	CAN WAIT
The difference between an image and a container	Kubernetes and container orchestration
Writing a Dockerfile from scratch	Custom container runtimes
Container networking, volumes, and port mapping	Advanced multi stage build optimisation
Docker Compose for multi service setups	Container security scanning in depth (Stage 7)

MINI PROJECT

Containerise an application you already have, and get a two service setup running with Docker Compose, for example the app plus its database.

COMMON MISTAKE

Copying a Dockerfile from a tutorial without understanding each instruction, then being unable to explain why your image is the size it is.

READY FOR STAGE 3 WHEN

You can write a Dockerfile for a new application without a reference open beside you.

Outcome: any application you touch, you can package and run consistently, anywhere.

Terraform and Infrastructure as Code

This is where you stop clicking through a cloud console and start defining infrastructure as something repeatable and version controlled, a genuine mindset shift, not just a new tool.

LEARN NOW	CAN WAIT
Terraform syntax: providers, resources, variables	Multi cloud abstraction patterns
What Terraform state actually is and why it matters	Advanced module design for large teams
Provisioning basic resources on one cloud provider	Enterprise remote state and locking strategies
Basic IAM concepts for that provider	Deep IAM policy design (Stage 6)

MINI PROJECT

Provision a small, real environment with Terraform, for example a virtual machine, a storage bucket, and basic networking, then destroy it cleanly using the same code.

COMMON MISTAKE

Manually editing infrastructure in the console after provisioning it with Terraform. This causes state drift and is one of the fastest ways to look inexperienced in an interview.

READY FOR STAGE 4 WHEN

You can explain what Terraform state does, in your own words, and provision or tear down infrastructure without leaving anything behind manually.

Outcome: you can stand up and tear down real infrastructure through code, not clicks.

CI/CD with GitHub Actions

This is the stage that connects everything before it. It's also, plainly, one of the most commonly tested skills in DevOps interviews, so give it the attention that reflects that.

LEARN NOW	CAN WAIT
Pipeline stages, triggers, jobs, and steps	Jenkins, GitLab CI, or other platforms
Writing GitHub Actions workflow files	Multi environment pipeline orchestration
Automated testing inside a pipeline	Advanced artifact management
Secure secrets handling in a workflow	Pipeline level security scanning (Stage 7)

MINI PROJECT

Build a pipeline that automatically tests, builds, and deploys the containerised application from Stage 2, triggered by every push to your main branch.

COMMON MISTAKE

Hardcoding credentials directly into a workflow file instead of using GitHub Secrets. It works until it very publicly doesn't.

READY FOR STAGE 5 WHEN

You can build a working pipeline for a new project from memory, and debug a failing step using the logs alone.

Outcome: a working CI/CD pipeline you built yourself, and can explain stage by stage.

Kubernetes

Kubernetes is genuinely hard without Stages 1 through 4 in place, and genuinely learnable with them. This is the longest stage in the roadmap on purpose. Give it the time.

LEARN NOW	CAN WAIT
Pods, deployments, services, namespaces	Custom controllers and operators
ConfigMaps and secrets, managed properly	Service mesh (Istio, Linkerd)
kubectl for inspecting and troubleshooting workloads	Multi cluster administration
Basic Helm chart structure	Writing your own Helm charts from scratch

MINI PROJECT

Deploy the application from Stage 4 onto a local cluster (Minikube or Kind), with a proper service definition and configuration managed through ConfigMaps and secrets.

COMMON MISTAKE

Memorising kubectl commands without understanding the objects behind them. It looks fine until a real cluster does something the tutorial never covered.

READY FOR STAGE 6 WHEN

You can deploy an application and troubleshoot a failing pod using logs and describe output, without external help.

Outcome: you can deploy, scale, and debug an application on Kubernetes on your own.

Cloud Security

Whichever path you're on, DevOps or DevSecOps, security literacy is no longer optional. This stage builds the baseline that Stage 7 then puts into practice inside a pipeline.

LEARN NOW	CAN WAIT
Least privilege and IAM roles done properly	Formal penetration testing
Security groups and network segmentation	Offensive security certifications
Proper secrets management, not plain text	Compliance frameworks in full depth
Spotting common misconfigurations	Building your own scanning tooling

MINI PROJECT

Audit the infrastructure you built in Stage 3, find at least three real security gaps, and fix them with proper IAM roles and secrets handling. Document what you found.

COMMON MISTAKE

Granting broad permissions while learning, and never going back to tighten them once the project "works."

READY FOR STAGE 7 WHEN

You can look at an IAM policy or a security group and immediately spot what's overly permissive.

Outcome: you can find and fix real cloud misconfigurations, not just describe security concepts.

This is the stage that actually separates DevSecOps from DevOps with security bolted on afterward. Security checks move inside the pipeline, catching problems before deployment, not after.

LEARN NOW	CAN WAIT
Shift left security as a working practice	Building custom security tools
Container image scanning in a pipeline	Advanced compliance automation
Dependency and SAST scanning basics	Enterprise security orchestration platforms
Failing a build safely on a real risk	

MINI PROJECT

Add a scanning step to the pipeline from Stage 4, so every build is checked for vulnerable dependencies or insecure container images before it can deploy.

COMMON MISTAKE

Adding a scanner to a pipeline and then ignoring what it reports. The scanner isn't the point, acting on its output is.

YOU'RE READY TO SAY YOU KNOW DEVSECOPS WHEN

You can walk someone through your own pipeline and point to exactly where and why a security check happens before code reaches production.

Outcome: a working DevSecOps pipeline you built, not a set of concepts you can only describe.

CHOOSE YOUR PATH

Recommended sequence by goal

The seven stage roadmap is the full picture. Where you're headed changes the order slightly. Here's how.

GOAL	HOW THE SEQUENCE SHIFTS	WHERE TO SPEND EXTRA TIME
Move into DevOps	Follow the default order exactly: Foundations, Docker, Terraform, CI/CD, Kubernetes, then Cloud Security at baseline depth.	Kubernetes and CI/CD. These two get tested the hardest in DevOps interviews.
Move into DevSecOps	Pull Cloud Security and DevSecOps forward, right after CI/CD. Kubernetes moves to the end, and you can go deeper into it once your security foundation is solid.	Cloud Security and DevSecOps. This is the actual core of the role you're targeting.
Still deciding, want cloud and automation depth first	Move Terraform ahead of Docker, since infrastructure and cloud fundamentals are useful regardless of which specialisation you eventually choose.	Terraform and CI/CD. Delay Kubernetes and DevSecOps until your direction is clearer.

STILL NOT SURE WHICH PATH FITS

That's a completely normal place to be, and not something to resolve by guessing. Reach out to SWQ IT Academy for a role based roadmap conversation before you commit three months of evenings to one direction.

Ninety day action plan

Built around eight to ten hours a week. If you have more time, you'll move faster. If you have less, extend the timeline rather than cutting the projects.

Days 1 to 30

Building the foundation

PRIMARY FOCUS

Stage 1 in full, and the start of Stage 2.

WEEKLY GOALS

Weeks 1 and 2: Linux and networking. Week 3: Git and GitHub in depth. Week 4: scripting fundamentals, plus your first Docker image.

HANDS ON TARGET

Daily terminal practice, three scripts pushed to GitHub, one working Docker image by day 30.

BY DAY 30 YOU SHOULD HAVE

A GitHub repository with real automation in it, and enough Docker comfort to containerise a simple app without guidance.

Days 31 to 60

Containers, infrastructure, and delivery

PRIMARY FOCUS

Finishing Stage 2, working through Stage 3, and starting Stage 4.

WEEKLY GOALS

Week 5: Docker Compose. Weeks 6 and 7: Terraform, provisioning real resources. Week 8: your first GitHub Actions pipeline.

HANDS ON TARGET

Two different infrastructure setups provisioned and destroyed with Terraform, plus a basic build pipeline running.

BY DAY 60 YOU SHOULD HAVE

A working Terraform project, and an early CI/CD pipeline connected to your repository.

Days 61 to 90

Orchestration and security

PRIMARY FOCUS

Finishing Stage 4, moving through Kubernetes, and beginning Cloud Security based on your chosen goal path.

WEEKLY GOALS

Week 9: finishing your CI/CD pipeline end to end. Weeks 10 and 11: Kubernetes fundamentals and a real deployment. Week 12: cloud security basics and a review of your own environment.

HANDS ON TARGET

Two successful Kubernetes deployments, one from scratch, plus a documented security review of your infrastructure.

BY DAY 90 YOU SHOULD HAVE

A complete pipeline running from commit to a live Kubernetes deployment, and a written record of the security gaps you found and fixed.

Practical project ideas

Five projects that map directly onto the roadmap and hold up well in an interview.

Containerised application with Docker Compose

01

TEACHES

Application packaging and multi service local environments.

BELONGS TO

Stage 2

PORTFOLIO VALUE

Solid, concrete proof of real containerisation ability, not just familiarity.

Infrastructure as Code environment on one cloud provider

02

TEACHES

Declarative infrastructure design and safe provisioning and teardown.

BELONGS TO

Stage 3

PORTFOLIO VALUE

One of the most frequently requested pieces of evidence in DevOps interviews.

End to end CI/CD pipeline with GitHub Actions

03

TEACHES

Build automation, testing, and deployment triggered by a commit.

BELONGS TO

Stage 4

PORTFOLIO VALUE

A visible pipeline you can walk an interviewer through, stage by stage.

Application deployment on Kubernetes

04

TEACHES

Orchestration, configuration management, and troubleshooting on a real cluster.

BELONGS TO

Stage 5

PORTFOLIO VALUE

Proof you can operate beyond a single container.

Secure pipeline with integrated scanning

05

TEACHES

Embedding security checks into delivery and acting on scan results.

BELONGS TO

Stage 7

PORTFOLIO VALUE

An upgraded version of Project 3, and genuinely strong evidence for a DevSecOps application.

Common roadblocks, answered plainly

“ I don't know where to start

Start at Stage 1, even if parts of it feel familiar. It moves fast if you're already strong there, and if it doesn't move fast, that tells you something useful.

“ I work full time and barely have evenings free

This roadmap assumes that. The ninety day plan is built around eight to ten hours a week, not full days. Showing up consistently for forty five minutes beats an occasional weekend marathon.

“ There are too many tools out there

Every stage in this roadmap recommends exactly one. Learn that one properly. Alternatives can wait until your job or a specific interview actually asks for them.

“ I'm coming from support, security, or pure development, can I still transition

Yes, and you're closer than you think. Support engineers already understand production. Security professionals already understand risk. Developers already understand code and Git. You're reorganising existing knowledge, not starting from nothing.

“ I keep learning but never seem to build anything

Go back to the last stage you "completed" and check honestly whether you actually built its mini project. If not, that's the gap, not a new course.

A weekly rhythm that actually holds up

Built around eight to ten hours a week for someone with a full time job. Adjust the hours, keep the shape.

WHEN	FOCUS	TYPICAL LENGTH
Monday to Thursday evenings	Concept learning: documentation and short, focused explanations rather than long generic tutorials	45 to 60 minutes
Saturday	Hands on practice, working directly inside the current stage's tool	2 to 3 hours
Sunday	Project work, applying the week's concepts to your mini project	2 to 3 hours
One short weekday session	Notes and revision, written in your own words, plus updating your GitHub READMEs	30 minutes

WHY THIS PACE, AND NOT MORE

Eight sustained hours a week for three months consistently beats twenty hours in a burst that fades by week three. This roadmap is built for the pace you can actually keep.

What not to do

Five mistakes responsible for more stalled transitions than any missing technical skill.

- ✗ **Learning every tool at once.** Depth in the tool each stage recommends beats shallow exposure to five alternatives.
- ✗ **Jumping to Kubernetes before the basics are solid.** Without Docker and Linux underneath it, Kubernetes turns into memorisation instead of understanding.
- ✗ **Stitching together random playlists with no structure.** Individual videos are fine for one concept. A pile of unrelated ones is the scattered approach this roadmap replaces.
- ✗ **Skipping the mini projects.** A concept you've only read about rarely survives contact with an actual interview question.
- ✗ **Chasing certifications ahead of fundamentals.** A certificate without practical ability behind it tends to fall apart the moment an interviewer asks you to actually do the thing.

Who should choose this roadmap vs. another one

Honest fit matters more than a polished pitch. Here's how to tell if this is genuinely the right roadmap for where you are.

THIS ROADMAP FITS YOU IF	LOOK FOR A DIFFERENT RESOURCE IF
You already work in IT and have hands on experience in some part of the stack, development, support, QA, security, or infrastructure.	You're a college fresher with no professional IT experience yet. DevOps and DevSecOps roles generally expect prior experience, and this roadmap assumes that base.
You want a staged, sequential path with a clear "learn now versus can wait" line at every step.	You're looking for an exhaustive tour of every tool in the DevOps ecosystem. That exists elsewhere, and it's a reasonable thing to want, just not what this roadmap is built for.
You have limited weekly hours and need a plan that respects that, instead of assuming unlimited free time.	You want a certification focused path with exam preparation as the primary goal. This roadmap treats certifications as optional, well behind hands on ability.
You'd rather build seven real, demonstrable projects than collect a longer list of half finished tutorials.	You already have strong, demonstrable experience in five or six of these seven stages. You likely need a targeted gap filling plan, not a full roadmap from Stage 1.

IF YOU'RE NOT SURE WHICH SIDE YOU FALL ON

That's exactly the kind of question worth a short conversation instead of a guess. Tell us your background and we'll tell you honestly whether this roadmap fits, or point you somewhere better if it doesn't.

You have the map. Let's make sure it fits your terrain.

A general roadmap can only go so far. It doesn't know your current role, how much time you actually have this month, or which of these seven stages you've secretly already half learned. A short conversation usually clears that up faster than another week of second guessing.

Tell us where you're starting from

Message us with your current role and background. We'll tell you honestly which stages you can move through faster, and which ones deserve your full attention, no generic answer, just a straight read on your situation.

Study alongside people doing the same thing

The SWQ IT Academy community is working professionals moving through this exact transition, comparing notes and staying accountable to their own ninety day plan instead of doing it alone.

Still weighing DevOps against DevSecOps

That's worth ten minutes on a call before it costs you three months of evenings. No pitch, just a clear read on which direction actually fits your background.



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.