DevOpsSchool

Course curriculum · November 2026

LEVEL 04 OF 04 · ADVANCED

Makefiles Advanced Training

Design fast, portable and maintainable make-based build systems for large codebases, with GNU conventions, reproducibility and CI scale.

LIVE ONLINE **Max 8-10 hours**	CORPORATE (DAY-WISE) **5 days · Days 1-5**	SELF-PACED VIDEO **1-year LMS access**
BATCH SIZE **Max 10 learners**	NEXT BATCHES **1st of every month · weekends: first Saturday**	CERTIFICATE **Verifiable certificate of completion**

Foundation	Essential	Intermediate	Advanced
1 day · Included	2 days · Included	3 days · Included	5 days · This curriculum

INSIDE THIS CURRICULUM

What you will achieve	Outcomes and level focus
Who should attend	Audience and prerequisites
Agenda	46 topics · Foundation to Advanced
Hands-on labs	4 labs
Formats and fees	Live, 1:1, self-paced, corporate
Trainer and reviews	Rajesh Kumar
Enrolment	Batches, payment, contacts

WHY DEVOPSSCHOOL

- Rated 4.6 on Google
- 1,500+ verifiable certificates issued in the public registry
- Trainer Rajesh Kumar: 20 years in industry
- Small batches: max 10 learners, every session recorded

Course page: www.devopsschool.com/courses/makefile/makefiles-advance-training.html
Talk to us: WhatsApp +91 99057 40781 · +1 (469) 756-6329 · contact@DevOpsSchool.com · reply within 1 hour

OUTCOMES

What you will achieve

By the end of the Makefiles course path, you will be able to:

1

Read and write Makefiles

Understand targets, prerequisites, recipes and variables well enough to maintain any Makefile you inherit.

2

Automate project workflows

Use make to drive builds, tests, containers and CI jobs through one consistent interface.

3

Build at scale

Design portable, parallel and reproducible make-based build systems for large codebases.

WHAT THE ADVANCED LEVEL ADDS

Large builds, portability, performance and standards

Hands-on: Scalable build system capstone. Build a non-recursive, parallel-safe build for a multi-module codebase with GNU-standard targets, packaging and a CI pipeline.

WHO SHOULD ATTEND

- Experienced Makefiles practitioners responsible for quality, security and scale.
- Senior engineers, architects and leads who design and troubleshoot Makefiles solutions.
- Teams making Makefiles an organisation-wide standard.

BEFORE YOU START

- The Intermediate level, or equivalent hands-on experience.
- You can already: Automate builds, tests and container workflows across languages using make functions, parallel jobs and reliable debugging techniques.
- Basic computer skills and a laptop with a stable internet connection for the live sessions and labs.
- This curriculum includes every earlier level, so it is complete on its own.

AGENDA · FOUNDATION TO ADVANCED

Makefiles Advanced Training agenda

46 topics. Each level builds on the one before it, so this curriculum lists every earlier level in full. Days are for corporate (day-wise) delivery; live online batches cover the same agenda in max 8-10 hours.

DAY 1 — Foundation: Makefile anatomy, setup and first targets

Included from Foundation

1. Why Makefiles: build automation, dependency tracking and make as a portable project task runner
2. Setup: GNU make on Linux, macOS (Homebrew gmake) and Windows via WSL or MSYS2, checking make --version
3. Makefile anatomy: target, prerequisites and recipe lines, comments and line continuation
4. Tabs and shells: why recipes need a tab, one shell per line and the @ echo prefix
5. Timestamps: how make compares file modification times to decide what to rebuild
6. Default goal: the first target, .DEFAULT_GOAL and running make with named goals
7. Phony targets: .PHONY for build, test, clean and run targets
8. Simple variables: defining and referencing variables to avoid repeated compiler flags and paths
9. Dry runs and silent runs: make -n, make -s and reading the commands make would execute
10. First project: a Makefile that compiles a small C program and cleans its build output

DAY 2 — Essential: Pattern rules, variables and everyday workflows

Included from Essential

11. Variable flavours: recursive =, simple :=, conditional ?= and append +=
12. Automatic variables: $@, $<, $^, $? and $* inside generic rules
13. Pattern rules: %.o from %.c, build output directories and order-only prerequisites
14. Built-in rules and variables: CC, CXX, CFLAGS, CPPFLAGS, LDFLAGS and LDLIBS
15. Wildcards and file lists: $(wildcard), $(patsubst) and substitution references
16. Conditionals: ifeq, ifneq, ifdef and ifndef for OS detection and debug or release builds
17. Command-line overrides: make VAR=value, environment variables and the override directive
18. Recipe control: ignoring errors with -, .DELETE_ON_ERROR and stopping on the first failure
19. Including files: include and -include for shared settings and generated dependency files
20. Header dependencies: -MMD -MP generated .d files so header changes trigger rebuilds
21. Help target: self-documenting Makefiles that list targets with descriptions
22. Team workflow: a consistent make build, make test and make lint interface across repositories

DAY 3 — Intermediate: Functions, multi-language automation and debugging

Included from Intermediate

23. Text functions: subst, patsubst, filter, filter-out, sort, words and strip
24. File name functions: dir, notdir, basename, suffix, addprefix and addsuffix
25. Control functions: foreach, if, call, eval and define for generated rules
26. Shell integration: $(shell) for versions and git commit IDs, and avoiding slow parse-time calls
27. Static libraries: building archives with ar and linking them into applications
28. Multi-directory projects: recursive make with $(MAKE) -C versus include-based builds
29. Parallel builds: make -j, the jobserver and writing rules that are safe to run concurrently
30. Makefiles for Go and Python: build, test, lint, virtualenv and dependency targets

31. Container workflows: docker build, tag and push targets with version variables
32. Makefiles in CI: GitHub Actions, GitLab CI and Jenkins steps that call make targets
33. Guard checks: $(error), $(warning) and $(info) to validate tools and required variables
34. Debugging: make -p, --debug, --trace and fixing missing separator and circular dependency errors

DAYS 4-5 **Advanced: Large builds, portability, performance and standards** — New at this level

35. Non-recursive build design: a single dependency graph across a large source tree
36. Generated rules at scale: eval and call macros for per-module build definitions
37. Build performance: profiling slow builds, reducing re-evaluation and minimising $(shell) usage
38. Portability: POSIX make versus GNU make versus BSD make and writing portable recipes
39. GNU Makefile conventions: all, install, uninstall, check, dist and distclean targets
40. Installation variables: prefix, bindir, libdir, datadir and DESTDIR for packaging
41. Reproducible builds: pinned toolchains, SOURCE_DATE_EPOCH and deterministic outputs
42. Cross-compilation: CROSS_COMPILE prefixes, sysroots and per-target toolchains
43. Extending GNU make: Guile scripting and loadable objects with the load directive
44. When to move on: CMake, Meson, Bazel or Ninja versus hand-written Makefiles
45. Governance: shared build includes, versioned conventions and reviewing build changes like code
46. Capstone: a portable, parallel, non-recursive build with install, test and package targets in CI

PRACTICE

Hands-on labs

You build these yourself during the course, guided by the trainer.

FOUNDATION

First Makefile for a C program

Write a Makefile with build, run and clean targets for a small C program and verify incremental rebuilds after editing one file.

ESSENTIAL

Reusable build with debug mode

Turn the project into a pattern-rule build with a separate build directory, header dependency tracking and a DEBUG=1 switch.

INTERMEDIATE

Polyglot project automation with make

Automate a repository with a C library, a Go service and a Python tool, including tests, Docker images and a CI job that runs make.

ADVANCED

Scalable build system capstone

Build a non-recursive, parallel-safe build for a multi-module codebase with GNU-standard targets, packaging and a CI pipeline.

HOW YOU CAN LEARN

Formats and fees

The same agenda in four formats. Fees are in Indian rupees as of November 2026; the course page shows them in your currency.

FORMAT	DURATION	WHAT YOU GET	FEE
Live & interactive (online)	Max 8-10 hours	Instructor-led public batch of max 10 learners with demos, labs, Q&A and assignment review. Weekday batches start on the 1st of every month, weekend batches on the first Saturday of every month.	**INR 34,999**
1:1 live training (online)	Max 8-10 hours, one-to-one	The same agenda, taught one-to-one by the trainer. Starts immediately, without waiting for the next batch; sessions planned around your working hours, at your pace.	**INR 99,999**
Self-paced video	1-year LMS access	All four Makefiles levels, plus 20+ more courses and 50+ tools, at your own pace.	**INR 833/month** billed yearly at INR 9,996
Corporate (day-wise)	5 days for this level	A private batch for your team, on demand. Foundation 1 day, Essential 2, Intermediate 3, Advanced 5 (cumulative).	**Quote on request**

ASSESSMENT AND CERTIFICATE

Your progress is checked through hands-on labs and assignments at every level. On completion you receive a DevOpsSchool certificate of completion with a unique credential ID that anyone can verify at devopsschool.com/certificates.

INCLUDED WITH EVERY LIVE BATCH

- Every live session is recorded and stays in your LMS for a year.
- Can't attend your batch? Move to next month's batch.
- Free demo class before you pay.
- Free 10-15 minute call with an advisor or the trainer to check the course fits you.

YOUR TRAINER

Learn from Rajesh Kumar

Principal Architect, trainer and mentor in DevOps, SRE, DevSecOps and platform engineering, with 20 years across 8+ software MNCs, including Staff DevOps/SRE Engineer for a leading Japanese mobile-payment platform and Principal DevOps Architect and Manager in Europe.

Profile: www.rajeshkumar.xyz · Videos: youtube.com/@TheDevOpsSchool

20 years in industry	**200+** corporate trainings
10,000+ engineers mentored	**70+** organisations coached

REVIEWS

What learners say

Rated 4.6 on Google · more reviews at devopsschool.com/testimonials and in our video testimonials.

> “Rajesh is very good trainer. Rajesh was able to resolve our queries and question effectively. We really liked the hands-on examples covered during this training program.”
>
> Indrayani, India

> “The training was very useful and interactive. Rajesh helped develop the confidence of all.”
>
> Abhinav Gupta, Pune

START YOUR MAKEFILES TRAINING

Join the next batch

Weekday batches start on the 1st of every month and weekend batches on the first Saturday of every month, with max 10 learners each. 1:1 training can start immediately; corporate batches run on demand. We reply within 1 hour.

Pay by UPI, credit and debit cards, net banking or PayPal. Before you pay, read our refund policy. All levels of this course: www.devopsschool.com/courses/makefile/

ENROL OR RESERVE A SEAT
Makefiles Advanced Training

WHATSAPP · INDIA
+91 99057 40781

WHATSAPP · USA
+1 (469) 756-6329

EMAIL
contact@DevOpsSchool.com