

TERRAFORM

Terraform Certified Associate with GCP Preparation Training

Delivered from Hacky Security's own curriculum, by instructors who also work our security operations centre. Classroom, online live, or at your office.

ADVANCED

32 HOURS

4 DAYS

DEVOPS LINUX

At a glance

COURSE	Terraform Certified Associate with GCP
TRACK	Terraform
DOMAIN	Cloud & Infrastructure
LEVEL	Advanced
DURATION	32 hours (about 4 days)
DELIVERY	Classroom · Online live · On-site at your office
LANGUAGE	English
CLASS SIZE	Small cohorts — everyone gets lab time

WHAT IS INCLUDED

- ✓ Instructor-led delivery by practitioners who also work our SOC
- ✓ Hands-on lab access for the duration of the course
- ✓ Course materials and lab guides
- ✓ Hacky Security certificate of completion
- ✓ Post-course support by email

Who this is for

This is an advanced course. It suits people already working in or moving into Cloud & Infrastructure who want structured, hands-on preparation rather than a video library. If you are unsure whether the level fits your team, send us the job titles and we will tell you honestly — including if a different course would serve you better.

Course outline

About this course map

Eight objective areas, mapped one-for-one to HashiCorp's published exam content list for the Terraform Associate and taught in the same order, so nothing on the exam is left uncovered.

Check the exam code before you study anything. The current version is Terraform Associate 004, and HashiCorp publishes a separate content list, learning path and sample question set for it. The older 003 objectives are still on the site and still findable by search, which is the trap — the two blueprints are not identical, and the differences are exactly where recent exam questions live. Everything below is mapped to 004. If you already bought preparation material for 003, tell us at enrolment and we will show you what moved.

The visible change in 004 is HashiCorp Cloud Platform Terraform, which is now a full objective area of its own covering workspaces, projects, collaboration and governance rather than a passing mention. Custom conditions for configuration validation and secrets management with Vault also appear explicitly in the configuration objectives. HashiCorp does not pin the objectives to a specific Terraform release; the content list points into the current documentation, so the assumption is that you are working on a recent version of the CLI.

Associate is a breadth exam, not a depth exam. It rewards knowing what each command does, what each block is for, and which of two plausible answers matches the way Terraform actually behaves. The teaching follows that: every objective area has hands-on time, with a terminal open and real state files to look at, because a state question is much easier to answer once you have read one.

Domain 01 — Infrastructure as Code (IaC) with Terraform

The framing objectives. Short, conceptual, and reliably present on the exam.

- What infrastructure as code means, and what distinguishes it from scripted provisioning
- Declarative configuration against imperative scripting, and what Terraform does with the difference
- The advantages of infrastructure as code patterns: repeatability, version control, review, and a plan you can read before anything changes
- Idempotency, and why running the same configuration twice should not build twice
- How Terraform supports multi-cloud, hybrid cloud and service-agnostic workflows through a single language and workflow
- The provider model as the thing that makes service-agnostic workflows possible
- Where infrastructure as code fits alongside configuration management, and what it does not replace

Domain 02 — Terraform fundamentals

Providers, plugins and the first real look at state.

- Installing Terraform and confirming the version you are working with
- How Terraform uses providers to talk to an API, and where provider plugins are installed
- Declaring required providers, and specifying a source and a version constraint
- Version constraint syntax, and the difference between pinning and allowing an upgrade
- Writing configuration that uses more than one provider, including multiple configurations of the same provider through aliases
- Provider authentication approaches, and keeping credentials out of the configuration
- How Terraform uses and manages state, and what state is actually for
- The dependency lock file and the role it plays in reproducible runs

Domain 03 — Core Terraform workflow

Write, plan, apply — and every command in the objectives that surrounds it.

- The Terraform workflow and the point of each stage
- Initialising a working directory, and what init downloads and writes
- Validating a configuration, and the class of error validate will and will not catch
- Generating an execution plan, reading it line by line, and saving a plan file
- Interpreting plan symbols: create, destroy, update in place, and replacement
- Applying changes to infrastructure, including applying a saved plan
- Destroying Terraform-managed infrastructure, and targeting a destroy
- Applying formatting and style adjustments with fmt, and the conventions it enforces
- Recognising which command to reach for from a described symptom, which is how the exam usually asks

Domain 04 — Terraform configuration

The language itself, and the largest objective area on the blueprint.

- Resource blocks and data blocks, and the difference between managing a thing and reading one
- Referring to resource attributes, and building cross-resource references
- Input variables: types, defaults, descriptions, validation and how values are supplied at run time
- Output values, and how they surface data from a configuration or a module
- Local values, and when a local is clearer than repeating an expression
- Complex types — lists, sets, maps, objects and tuples — and choosing between them
- Dynamic configuration with expressions, built-in functions, conditionals, for expressions and dynamic blocks
- Implicit dependencies from references, and explicit dependencies with depends_on
- Validating configuration with custom conditions, including preconditions, postconditions and check blocks
- Handling sensitive data: marking values sensitive, keeping secrets out of state and source control, and sourcing them from Vault

Domain 05 — Terraform modules

Packaging configuration so it can be reused without being copied.

- What a module is, and the distinction between a root module and a child module
- How Terraform sources modules: local paths, the registry, version control and other supported sources
- Calling a module, passing input variables to it and consuming its outputs
- Variable scope within modules, and why a child module cannot see the caller's variables
- Managing module versions, and applying version constraints to registry modules
- The public Terraform Registry, and publishing requirements for a module in it
- Structuring a module so it is genuinely reusable rather than one configuration with parameters
- Using count and for_each with module blocks

Domain 06 — Terraform state management

State is the subject most Associate candidates underestimate, and it is a full objective area.

- The local backend, where state is written, and the risks of leaving it there
- State locking, what it prevents, and what happens when a lock is not released
- Configuring remote state with the backend block

- Backend initialisation and migrating existing state to a new backend
- Sensitive data in state files, and why state must be treated as a secret
- Resource drift: what causes it, how Terraform detects it during refresh, and how the plan reports it
- Reconciling drift, and the difference between changing the infrastructure and changing the state
- State commands that modify state, and treating them as the last resort rather than the first

Domain 07 — Maintain infrastructure with Terraform

Working with infrastructure that already exists, and diagnosing a run that has gone wrong.

- Importing existing infrastructure into a Terraform workspace
- The import block as configuration, and generating configuration for imported resources
- Inspecting state from the command line: listing resources and showing a single resource
- Reading the output of a state inspection to confirm what Terraform believes it manages
- Replacing a resource deliberately, and how that is requested in current Terraform
- When and how to use verbose logging, and the log levels available
- Directing log output to a file and reading it to find the failing API call
- Working out from a failed apply whether the problem is the configuration, the credentials or the provider

Domain 08 — HCP Terraform

The objective area that expanded in 004, and the one most self-taught candidates have never touched.

- Creating infrastructure with HCP Terraform, and how a remote run differs from a local one
- Remote state storage and the state management HCP Terraform provides by default
- Workspaces and projects, and organising them for a real estate rather than a demonstration
- Variables and variable sets at workspace and project level
- Collaboration features: remote runs, run history and the approval step before an apply
- Governance features, including policy enforcement and cost estimation
- Version control integration and the run that is triggered by a commit
- The private registry for sharing modules and providers inside an organisation
- Choosing between CLI-driven, version-control-driven and API-driven workflows

Who should take this

Cloud and platform engineers, DevOps and site reliability engineers, system administrators moving into infrastructure automation, and developers who have to provision their own environments. It also suits architects who are specifying Terraform for a team and want to know what they are committing to.

HashiCorp names two prerequisites for the exam: basic terminal skills, and an understanding of on-premises and cloud architecture. That is an honest summary. You do not need to be a programmer, but you do need a cloud account you understand well enough to recognise what Terraform is creating in it, and you should be comfortable reading a configuration file without being frightened of the braces.

The exam

The HashiCorp Certified: Terraform Associate is set and awarded by HashiCorp.

- Exam version 004 is the current one; the older 003 objectives are still published and are not the same blueprint
- Multiple choice

- One hour
- Delivered online with proctoring
- Available in English
- 70.50 US dollars plus local taxes and fees, with no free retake included
- The credential is valid for two years

HashiCorp does not publish a question count or a passing score for the Associate exam on its certification pages, so we do not quote one. Pace yourself against the clock rather than against a question total.

The exam is purchased directly from HashiCorp. We send you the registration details and you choose your own exam date.

How we deliver it

Two days, classroom or online live, or at your office for a private group. Roughly half the time is at the terminal. Every delegate gets their own sandbox cloud account and an HCP Terraform organisation, because the eighth objective area cannot be learned from slides and it is the one that catches people out.

You will build a configuration from nothing, break it deliberately, read the state file, import a resource that was created outside Terraform, migrate state to a remote backend, and run the same configuration through an HCP Terraform workspace to see what changes. Objective checkpoints at the end of each day, and a full-length timed mock on the final afternoon.

Mapped to HashiCorp's published exam content list for Terraform Associate 004. Taught from Hacky Security's own curriculum and lab estate by instructors who use Terraform in production. These are our own preparation courses; they are not official HashiCorp courseware and we are not a HashiCorp partner.

Sitting the official exam

Exam voucher sold separately. We buy it from the certification body on your behalf and send you the code — you book your own exam date. This course is Hacky Security's own preparation training. The certification itself is awarded by the certification body, not by us.

Investment

FROM

₹65,000

per person

INR	₹65,000 per person
-----	--------------------

USD	\$779 per person
-----	------------------

GBP	£609 per person
-----	-----------------

AED	AED 2,899 per person
-----	----------------------

Group, corporate and repeat-cohort pricing on request. Prices are reviewed quarterly and exclude the exam voucher.

Book this course

Tell us how many people, where, and when — we will come back with dates and a quote within one working day.

+91-8091900310 · info@hackysecurity.com

<https://hackysecurity.com/courses/hacky-hashicorp-terraform-associate/>

CHANDIGARH · SHIMLA · DELHI · DUBAI · LONDON

Independent preparation training delivered from Hacky Security's own curriculum. Hacky Security Private Limited is not affiliated with or endorsed by the certification body named in this document. Certification names and codes are the trademarks of their respective owners and are used only to describe what this course prepares you for. Prices, durations and outlines are indicative and may change; the quotation we send you is the binding one. Issued 12 September 2026.