

CGE-GCP EXAM BLUEPRINT

Certified GRC Engineer, GCP Specialty

Official Examination Guide

GRC Engineering Club Training Academy

Instructor: AJ Yawn

Version 1.0 | November 2026

Built by the community, for the community

www.grcengclub.com

TABLE OF CONTENTS

- About This Document
- Exam Structure & Format
- Domain Overview
- Course to Domain Crosswalk
- Detailed Domain Specifications
- The Capstone: Ship Your Pipeline
- Certification Maintenance
- Bloom's Taxonomy Reference
- Candidate Preparation Recommendations

ABOUT THIS DOCUMENT

Purpose: This exam blueprint is the official guide for the Certified GRC Engineer, GCP Specialty (CGE-GCP) certification. It outlines the knowledge, skills, and competencies required to pass the examination and the graded capstone, and provides detailed specifications for all seven exam domains.

How to Use: Candidates should use this blueprint to understand exam scope, identify knowledge gaps, focus study on high-weight domains, and practice with question types aligned to Bloom's Taxonomy levels. Each domain description includes task statements, knowledge requirements, and skill statements to guide preparation.

Certification Value: The CGE-GCP validates that a GRC engineer can build and operate a complete, automated GRC engineering pipeline on Google Cloud, in their own project: a system that discovers what is running, activates what is missing, stores evidence in a store they own, generates auditor deliverables, narrates them weekly, and remediates non-compliant resources through a least-privilege identity. The one-liner: don't map controls to Google Cloud. Build the pipeline that proves them.

Prerequisites: CGE-P is the only prerequisite, enforced in the platform. The course assumes Terraform, Git and pull requests, and control objectives, because CGE-P taught all three. No Google Cloud certification and no other CGE specialty is required.

Tenancy: The course is project-scoped. Every lab runs in one Google Cloud project on a personal account with no Organization above it, and was validated on exactly that. Security Command Center and Org Policy both need an Organization, so the pipeline does not depend on either; they are appendix material.

Pricing: $350, or free for active GRC Engineering Club members.

EXAM STRUCTURE & FORMAT

Attribute	Details
Format	Computer-based, online proctored
Time Limit	75 minutes
Total Questions	50

Attribute	Details
Multiple-Choice	40 questions (4 options, single correct answer)
Scenario-Based	10 questions built on real artifacts (read a short artifact, answer 1 or 2 follow-up questions)
Passing Score	70% (35/50 correct)
Scoring	All questions weighted equally; scenarios count as individual questions
Negative Marking	None
Capstone	REQUIRED. The pipeline, deployed as code in your own project, submitted as a public repo and graded pass/fail against a published rubric (see The Capstone section)
Retake Policy	14-day waiting period between attempts; maximum 3 attempts per 12 months
Breaks	No section breaks; 75 minutes total

Question Distribution: The exam covers all seven domains with questions distributed by domain weight. The Evidence Plane (Domain 4) is the heaviest domain by design: it is where the pipeline gets built.

Difficulty Levels: Questions span all five Bloom's Taxonomy levels, weighted toward Apply and Analyze. CGE-GCP holders build and fix controls, so memorized definitions carry only a small share of the exam.

Scenario Questions: Each scenario presents a short artifact of the kind you produce in the labs (an IAM policy binding, a Terraform plan excerpt, a Cloud Audit Log entry, a Cloud Asset Inventory record, a failing check result, a pipeline run log) plus one or two questions asking you to identify the gap, choose the right control, or fix it.

DOMAIN OVERVIEW

#	Domain	Weight	Questions	Focus
1	GCP Foundations	15%	8	Organization, folders, projects, resources; IAM bindings, basic versus predefined roles, service accounts, Workload Identity Federation; shared responsibility
2	The GCP GRC Toolkit	15%	7	Cloud Asset Inventory and checks as code, preventive and corrective controls, Cloud Audit Logs, Cloud Run functions, Firestore, Cloud Storage Bucket Lock, Vertex AI
3	Terraform on GCP	15%	8	The google provider, keyless auth locally and in CI, APIs as code, the Cloud Storage backend, additive IAM, secure bucket defaults, the policy gate
4	The Evidence Plane	20%	10	Pipeline stages 1 to 3: read-only discovery and the gap map, conditional activation, the Firestore evidence store, the locked reports bucket, the keyless collector
5	Reporting & Narrative	15%	7	Pipeline stages 4 and 5: POA&M, SAR, and OSCAL 1.1.2 SSP from Firestore only; the separate reporter identity; the optional Gemini weekly digest
6	Enforcement & Operations	10%	5	Pipeline stage 6: the remediation ladder (report-only, approve, auto), the remediator identity, the OPA gate, drift
7	Capstone	10%	5	Requirements, rubric, reference walkthrough, submission

#	Domain	Weight	Questions	Focus
TOTAL		100%	50	

Domain Weight Explanation: Domains 1 through 3 make you fluent: the language, the services, and Terraform on the google provider. Domains 4 through 6 build the pipeline stage by stage, and the Evidence Plane carries the most weight because every report and every enforcement action traces back to it. Domain 7 covers the capstone itself.

COURSE TO DOMAIN CROSSWALK

Every lesson is mapped to the domain it is tested under. There is about an hour of video: 25 short lessons plus 6 lab walkthroughs, with most lessons running 2 to 3 minutes. Two text articles (00_02 How to Use This Course, 07_04 Certification Exam Guide) and six hands-on labs complete the course. Plan on 10 to 14 hours of study time, most of it at the keyboard in the labs.

#	Domain	Lessons covered
	Course Introduction	00_01 · 00_02 (article)
1	GCP Foundations	01_01 · 01_02 · 01_03 · 01_04 (Lab 1)
2	The GCP GRC Toolkit	02_01 · 02_02 · 02_03 · 02_04 · 02_05 (Lab 2)
3	Terraform on GCP	03_01 · 03_02 · 03_03 (Lab 3)
4	The Evidence Plane	04_01 · 04_02 · 04_03 · 04_04 (Lab 4)
5	Reporting & Narrative	05_01 · 05_02 (Lab 5) · 05_03
6	Enforcement & Operations	06_01 · 06_02 (Lab 6)
7	Capstone	07_01 · 07_02 · 07_03 · 07_04 (article)

DOMAIN 1: GCP FOUNDATIONS

Weight: 15% (8 questions)

TASK STATEMENTS

1.1 Read the Google Cloud resource hierarchy (Organization, folders, projects, resources) as a map of control inheritance

1.2 Explain why the project is the unit of billing, APIs, IAM, quotas, and audit logs, and why this course scopes every control to one project

1.3 Read any IAM allow policy binding as principal plus role plus resource

1.4 Distinguish basic, predefined, and custom roles, and recognize a basic-role grant or a user-managed service account key as a finding

1.5 Explain how workloads authenticate without keys: attached service accounts, and Workload Identity Federation for CI

1.6 Apply the shared responsibility model and map each customer-side responsibility to the pipeline stage that produces evidence for it

KNOWLEDGE STATEMENTS

- The four hierarchy levels and downward policy inheritance: a control set at the top covers everything below it, a control set on one project covers one project
- Billing accounts are a separate object: one billing account can pay for many projects, and access to billing is its own permission set
- A personal @gmail.com account gives you projects with no Organization; accounts on a company or custom domain put every new project under that domain's Organization
- The three principal types (users, groups, service accounts) and the three role kinds: basic (Owner, Editor, Viewer), predefined, custom
- IAM conditions narrow a grant; deny policies sit above allow policies, and a deny on a permission wins over any grant
- Shared responsibility and Google's shared-fate framing: Google secures the data centers, hardware, hypervisor, and managed services; you secure everything you configure, and the evidence is yours too
- The sandbox from Lab 1: one project, a $25 budget with alerts at 50%, 90%, and 100% (a budget notifies, it does not stop spending), two logins, a pinned project, and the probe script

SKILLS STATEMENTS

- Scope an audit boundary by naming the level at which each control is set
- Read a binding and state who can do what, and where
- Diagnose the two most common setup failures: a missing Application Default Credentials login, and USER_PROJECT_DENIED from a gcloud config pointing at another project
- Place any control on the correct side of the shared responsibility line

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	2
Understand	3
Apply	2
Analyze	1
Evaluate	0

DOMAIN 2: THE GCP GRC TOOLKIT

Weight: 15% (7 questions)

TASK STATEMENTS

2.1 Name the eight services that run the pipeline and the one job each does

2.2 Query Cloud Asset Inventory for resource and IAM state, and read a check result end to end: check ID, resource, status, evidence, controls

2.3 Classify a control as preventive, detective, or corrective, and choose the project-scoped mechanism for each

2.4 State which Cloud Audit Logs exist by default, enable Data Access logs deliberately, and query logs for evidence

2.5 Explain why Security Command Center and Org Policy are appendix material and how the course builds the same controls itself

KNOWLEDGE STATEMENTS

- The toolkit: Cloud Asset Inventory (what exists), Cloud Audit Logs (who did what), IAM (who can act), the policy gate (what can deploy), Cloud Run functions on Cloud Scheduler, Firestore, Cloud Storage with a locked retention policy, and Vertex AI
- gcloud asset search-all-resources and search-all-iam-policies; a query for allUsers finds every public resource in the project
- The five checks as code: CHK-STG-01 no public buckets, CHK-STG-02 uniform bucket-level access on, CHK-IAM-01 no basic roles on service accounts, CHK-IAM-02 no user-managed service account keys, CHK-LOG-01 Data Access logs on for Cloud Storage; each maps to NIST CSF 2.0 and SP 800-53
- Inventory freshness: the inventory updates shortly after a change, not instantly (57 seconds on the validation project)
- The four audit log types: Admin Activity (always on), System Event and Policy Denied (written by default), and Data Access (mostly off by default, your decision per service)
- Log bucket retention: _Required keeps 400 days and cannot be changed; _Default keeps 30 days unless you change it; a log sink to longer storage closes the gap
- Org Policy enforces constraints at request time and has a native dry-run mode, but on a project with no Organization even the Owner is denied, and the Org Policy Admin role cannot be granted on a project
- The three layers you own instead: secure defaults in Terraform and the policy gate (preventive), and the remediation function (corrective) with report-only, approval, and automatic modes

SKILLS STATEMENTS

- Name the service that owns an auditor's question
- Find every public binding in a project with one Asset Inventory query, and re-run it after the measured delay if it comes back empty
- Trace a bucket IAM change to a person in Logs Explorer (storage.setIamPermissions; project IAM changes are SetIamPolicy)
- Choose between a secure default, a gate rule, and a remediation for a given control requirement

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	2
Understand	3
Apply	1
Analyze	1
Evaluate	0

DOMAIN 3: TERRAFORM ON GCP

Weight: 15% (8 questions)

TASK STATEMENTS

3.1 Configure and pin the google provider, and authenticate with Application Default Credentials locally and Workload Identity Federation in CI, with no key file in either case

3.2 Manage APIs as code with google_project_service and disable_on_destroy = false

3.3 Keep state in a Cloud Storage backend with native locking, and give each pipeline stage its own root module and state

3.4 Write IAM additively with google_project_iam_member, and set secure defaults on every bucket

3.5 Deploy the foundation from code, prove each identity starts with nothing, connect your fork's CI without keys, and watch the gate fail

KNOWLEDGE STATEMENTS

• The google and google-beta providers; provider pinning

• Why disable_on_destroy = false: destroying one stage must not switch off an API another stage depends on

• The gcs backend locks state natively; the bootstrap creates the state bucket with versioning on and public access prevention enforced

• The three project IAM resources: _iam_member adds one member to a role and touches nothing else, _iam_binding owns every member of one role, _iam_policy owns the entire policy and removes anything not in code

• Secure bucket defaults: public_access_prevention = "enforced" and uniform_bucket_level_access = true

• Stage 00 creates six service accounts with zero permissions: grc-builder, grc-collector, grc-reporter, grc-narrator, grc-remediator, grc-scheduler; each later stage grants its own identity only what that stage needs

• arm-your-fork.sh takes your fork as a required argument and creates a Workload Identity Federation pool that trusts tokens from that fork only; CI holds read-only roles and never applies

• The policy gate: conftest checks every pull request's plan against the Rego rules in policy/

SKILLS STATEMENTS

• Write a google provider and backend configuration that needs no stored credential

• Read a plan for IAM members being removed and guardrail settings changing before you apply

• Prove a new service account holds no roles from the project IAM policy

• Add a deliberately bad change (roles/editor on a pipeline identity) and read the gate failure that names the rule

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	1
Understand	1
Apply	4
Analyze	2
Evaluate	0

DOMAIN 4: THE EVIDENCE PLANE

Weight: 20% (10 questions)

TASK STATEMENTS

4.1 Build the discovery stage: measure what is running, change nothing, and output a gap map the next stage consumes

4.2 Build the activation stage: act only on what discovery measured as absent, adopt what is present, and leave different settings to a human

4.3 Deploy the evidence store: Firestore collections with deterministic document IDs, the daily collector running five checks over Cloud Asset Inventory, and the reports bucket with a locked retention policy

4.4 Wire the store's identities: the collector can read Asset Inventory and write Firestore and nothing else, a dedicated builder builds the function, a dedicated scheduler identity invokes it, and no key exists anywhere

4.5 Prove the two properties an assessor asks about: re-collection does not duplicate, and a stored report cannot be deleted

KNOWLEDGE STATEMENTS

- Discovery's four questions: Data Access logs for Cloud Storage, the audit log sink, buckets missing a guardrail, user-managed service account keys
- The four gap map states: present (leave it), absent (activate it), different (a human decides), pipeline (the pipeline created it, keep managing)
- Activation in code: a count driven by the gap map; it creates the audit config, the grc-audit-logs log bucket (365-day retention), and the grc-audit-sink, and marks what it owns with managed-by: cgegcp
- Firestore collections: assessments (one document per check per resource), frameworks, and mappings (the crosswalk from each check to NIST CSF 2.0 and on to SP 800-53)
- Lineage: every collector run stamps run_id and collected_at; the document ID is built from the check and the resource, so a re-run updates one document
- Bucket Lock makes a retention policy permanent: nobody, including the Owner, can delete or replace an object until it ages out, and the lock can never be removed; the labs use a one-day window
- On a new project with no Organization, the first collector run fails CHK-IAM-01: Google grants two default service accounts the Editor role

SKILLS STATEMENTS

- Read a gap map and predict what activation will and will not change
- Deploy stages 1 to 3 in order, seed the frameworks and mappings, and force-run the collector from Cloud Scheduler
- Prove the upsert: same document count after a second run, with a newer run_id on every document
- Capture the WORM proof: the failed delete whose error names the retention policy

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	0
Understand	1
Apply	4
Analyze	3
Evaluate	2

DOMAIN 5: REPORTING & NARRATIVE

Weight: 15% (7 questions)

TASK STATEMENTS

5.1 Name the pipeline's reports (POA&M daily, SAR weekly, OSCAL 1.1.2 SSP weekly) and explain what each gives an assessor

5.2 Defend the Firestore-only rule: report functions read the evidence store and never call live platform APIs

5.3 Explain separation of duties between the collector and the reporter, and prove the reporter's limits with a captured permission-denied error

5.4 Trace one POA&M line to its Firestore document and the inventory record behind it, and validate the SSP

5.5 Explain the narrative stage's guardrails: it describes, it never decides, and the pipeline still works if the stage is deleted

KNOWLEDGE STATEMENTS

• The POA&M lists each open finding with its CSF control, the 800-53 crosswalk, severity, and days open; the SAR summarizes assessment results by control; the SSP is machine-readable OSCAL 1.1.2

• grc-reporter can read Firestore and create objects in the reports bucket; it cannot write Firestore, overwrite or delete a report, or read Asset Inventory

• Every report file name carries a timestamp; because the bucket is locked, every version is retained for the retention period

• Even the project Owner cannot act as a pipeline identity without an explicit grant on that one service account

• Stage 5 uses a Gemini model on Vertex AI to write a weekly digest from the SAR and POA&M, never from raw findings; the prompt requires a file and line citation for every factual sentence, and the output is labeled AI-generated

• grc-narrator can read the reports bucket, add the digest to it, and call Vertex AI, nothing more; Stage 5 is optional and never graded

SKILLS STATEMENTS

• Force-run the report generators against a populated evidence store and find the outputs in the locked bucket

• Complete the trace from a POA&M line to its assessments document: same resource, same run_id

• Validate a generated SSP against the official NIST OSCAL 1.1.2 SSP schema

• Evaluate whether a proposed AI use in the pipeline describes or decides

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	0
Understand	1
Apply	3
Analyze	2

Level	Questions
Evaluate	1

DOMAIN 6: ENFORCEMENT & OPERATIONS

Weight: 10% (5 questions)

TASK STATEMENTS

6.1 Design enforcement as a ladder: report_only, then approve, then auto, moving up only when the data from the rung below says it is safe

6.2 Scope the remediator: one dedicated identity on a custom role that can fix a public bucket and do nothing else

6.3 Run the complete loop: a violation made by hand, detected, approved, fixed, re-collected, and dropped from the next POA&M

6.4 Operate the pipeline: the policy gate on every pull request, and nightly drift detection

KNOWLEDGE STATEMENTS

• The reference control is CHK-STG-01: no Cloud Storage bucket may be public

• The remediator runs after each collection and reads failing results from Firestore; in report_only mode it records would_fix and changes nothing

• In approve mode it writes an approval request, a human approves, and the next run removes the public grant and enforces public access prevention on that one bucket

• The mode is a Terraform variable: raising it is a reviewed pull request, so the decision to automate is on the record

• grc-remediator holds the custom role grcBucketRemediator (storage.buckets.get, getIamPolicy, setIamPolicy, update); it cannot read or delete objects or delete a bucket

• The four gate rules: no basic roles, no service account keys, bucket guardrails on every bucket, and the remediation function must run as its own dedicated service account

• drift.yml plans nightly and opens an issue if reality and code disagree

SKILLS STATEMENTS

• Deploy Stage 6 in report_only mode and read the record of what it would have fixed

• Raise the mode through a pull request, approve a pending fix, and confirm the result in Firestore

• Show the gate blocking a bucket without guardrails and a service account key, and save both failures as evidence

• Tear down with a locked bucket: destroy in reverse order, then delete the reports bucket after its retention expires

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	0
Understand	1
Apply	2

Level	Questions
Analyze	1
Evaluate	1

DOMAIN 7: CAPSTONE

Weight: 10% (5 questions)

TASK STATEMENTS

7.1 State the capstone requirements: the foundation, stages 1 to 3, at least two reports from Firestore only, Stage 6 remediation in report_only mode or higher, the gate and drift workflows, and the evidence folder

7.2 Evaluate a complete pipeline the way the grader does, using the five published rubric dimensions

7.3 Identify the auto-fail triggers and the common failure modes before submitting

7.4 Plan submission timing: exam first, capstone after, while the scheduled collector builds run history

KNOWLEDGE STATEMENTS

• The five rubric dimensions, each weighted 20%: Infrastructure-as-Code Quality, Control Implementation & Identity Design, Evidence Integrity & Traceability, Pipeline Operations, Documentation & Control Mapping

• Pass is a weighted average of 70 or higher with no auto-fail triggers; grading is automated and pass/fail, with 2 attempts and a 24-hour cooldown; appeals go to hello@grcengclub.com

• Submission format: a public repo link, built in your fork of github.com/GRCEngClub/cgegcp, with a README that opens with your tenancy statement

• The evidence folder: the WORM proof (failed delete), the separation-of-duties proof (reporter denied), and a gate failure from a closed test pull request

• Stage 5 (the AI narrative) is optional extra credit and never graded

SKILLS STATEMENTS

• Calibrate your own build against the reference walkthrough in 07_02

• Run ./self-check.sh, the mechanical half of the rubric, before you submit

• Read the published rubric at cert.grcengclub.com/rubric/cge-gcp and map each dimension to your repo

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	0
Understand	0
Apply	2
Analyze	2
Evaluate	1

THE CAPSTONE: SHIP YOUR PIPELINE

Why There Is a Capstone

CGE-GCP certifies builders. The exam proves you know the platform; the capstone proves the pipeline runs. The capstone is REQUIRED: the pipeline, deployed as code in your own Google Cloud project, submitted as a public repo link, and graded pass/fail against a published rubric you can read from day one. Six labs build it stage by stage in one repo, so if you do the labs, the capstone is that system, cleaned up and submitted.

What You Submit

A public GitHub repository containing, at minimum: a README that opens with your tenancy statement and explains how to deploy from an empty project; Terraform for the foundation, stages 1 to 3 (discovery, activation, the evidence store with a locked reports bucket), and Stage 6 remediation in report_only mode or higher; at least two reports generated from Firestore only, with real run history; the gate and drift workflows, armed on your fork; docs/CONTROLS.md current with your code; and an evidence/ folder with your WORM proof, your separation-of-duties proof, and a gate failure from a closed test pull request. Stage 5, the AI narrative, is optional extra credit and never graded.

How It Is Graded

Dimension	Weight	What It Measures
Infrastructure-as-Code Quality	20%	One root module per stage with separate state, pinned provider, discovery-first activation, APIs as code, additive IAM only
Control Implementation & Identity Design	20%	Every job on its own identity with only its job's roles, no basic roles, no keys, remediator on a custom role, one check of your own
Evidence Integrity & Traceability	20%	Locked reports bucket with the failed-delete proof, lineage-stamped collection, deterministic IDs, Firestore-only reporting
Pipeline Operations	20%	Run history that accumulates, a gate proven to block, remediation behind an approval or a written justification, drift detection
Documentation & Control Mapping	20%	Cold-readable README with deploy order, CONTROLS.md current with the code, blast-radius notes on the remediator

Pass is a weighted average of 70 or higher AND no auto-fail triggers. Grading is automated: 2 attempts, with a 24-hour cooldown between them, and written feedback either way. Appeals go to hello@grcengclub.com. The full rubric, including auto-fail triggers, is published at cert.grcengclub.com/rubric/cge-gcp and detailed in the CGE-GCP Capstone Overview PDF.

CERTIFICATION MAINTENANCE

Validity Period

The CGE-GCP is valid for 365 days from the date of issuance and is renewed annually, following the CGE family renewal model.

Renewal Options

Both options are accepted; only one is required to keep the certification active.

Option 1: Active Club Membership. Maintain an active GRC Engineering Club membership. The certification auto-renews while membership is in good standing.

Option 2: CPE Hours. Complete 30 Continuing Professional Education (CPE) hours within the annual cycle.

Content is reviewed annually against Google Cloud changes: Asset Inventory types, google provider major versions, and CSF crosswalk updates.

BLOOM'S TAXONOMY REFERENCE

Taxonomy Levels

Level	Description	Example Question Stem
Remember	Recall facts and basic concepts	"Which audit log type is always on and cannot be turned off?"
Understand	Explain ideas or concepts	"Why does the assessment document ID combine the check and the resource?"
Apply	Use information in new situations	"Given this Terraform plan excerpt, which gate rule fails the pull request?"
Analyze	Draw connections among ideas	"This report function imports the Asset Inventory client. Which pipeline rule does that break, and what does it cost you in an audit?"
Evaluate	Justify a decision or course of action	"Is this remediator identity least-privilege? Defend your answer from its role."

Distribution Across Exam

The blueprint targets are L1 Remember 10%, L2 Understand 20%, L3 Apply 35%, L4 Analyze 25%, L5 Evaluate 10%. Seventy percent of the exam is Apply, Analyze, and Evaluate. Over a 50-question draw the targets round to the counts below.

Level	Total Questions
Remember	5
Understand	10
Apply	18
Analyze	12
Evaluate	5

CANDIDATE PREPARATION RECOMMENDATIONS

Study Strategy

1. **Do the labs.** Six labs build the pipeline stage by stage in one repo, and the capstone is that repo. About an hour of video against 10 to 14 hours of total study time: the learning happens in the labs.

2. Set up once, carefully. Use a personal @gmail.com account, create one project for the whole course, set the $25 budget before anything else, log in twice, and pin the project. The Google Cloud free trial gives new customers $300 in credit for 90 days with no automatic charge, and the labs stay far inside it. Start it when you are ready to work, so the 90-day window covers Labs 1 through 6 and the capstone. The article 00_02 walks through it.

3. Weight your study toward the Evidence Plane. Domain 4 carries 20% of the exam and is the heart of the pipeline. Domains 1, 2, 3, and 5 carry 15% each.

4. Practice reading real artifacts. Scenario questions hand you IAM bindings, Terraform plan excerpts, audit log entries, Asset Inventory records, failing check results, and pipeline run logs. Read the artifact first and the question second.

5. Use the Study Guide. Work through all seven domains in order using the CGE-GCP Study Guide as the companion to this blueprint.

6. Take the practice exam. The free practice exam at cert.grcengclub.com/practice/cge-gcp has 20 questions with an explanation for every answer. Use it to find the domains to review.

7. Read the rubric before you build. The capstone rubric is published from day one. Calibrate against the reference walkthrough in 07_02 and run ./self-check.sh before you submit.

Recommended Resources

- GRC Engineering Club Training Academy (CGE-GCP video course and labs)
- CGE-GCP Study Guide (companion to this blueprint)
- CGE-GCP Capstone Overview (the rubric, submission requirements, and lab-to-capstone map)
- The lab and capstone starter repo: github.com/GRCEngClub/cgegcp
- The published capstone rubric: cert.grcengclub.com/rubric/cge-gcp
- Google Cloud documentation for Cloud Asset Inventory, IAM, Cloud Audit Logs, and Cloud Storage Bucket Lock
- Terraform google provider documentation (registry.terraform.io)
- NIST CSF 2.0 and SP 800-53 references; OSCAL 1.1.2 documentation
- #studygroup_cge_certs in the Club Slack

Contact Information

Website: www.grcengclub.com

Email: academy@grcengclub.com

Community: GRC Engineering Club (Patreon)