CGE-AUD EXAM BLUEPRINT

Certified GRC Engineer, Auditor Specialty

Official Examination Guide

GRC Engineering Club Training Academy

Instructor: AJ Yawn, former SOC 2 auditor

Version 1.0 (Beta) | June 2026 | Launches July 15, 2026

Built by the community, for the community

www.grcengclub.com

TABLE OF CONTENTS

ABOUT THIS DOCUMENT

Purpose: This exam blueprint is the official guide for the Certified GRC Engineer, Auditor Specialty (CGE-AUD) certification. It outlines the knowledge, skills, and competencies required to pass the examination 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-AUD validates that an auditor can evaluate a modern, GRC-engineered organization: reading cloud configurations, infrastructure as code, and pipelines as evidence, testing automated and continuous controls, and documenting it all in defensible workpapers. It is the auditor side of GRC Engineering. CGE-P certifies the engineers who build automated compliance; CGE-AUD certifies the auditors who evaluate it.

Prerequisites: None. The CGE-AUD assumes near-zero prior technical knowledge and is standalone, with no CGE-P prerequisite. We skip a generic "frameworks 101" chapter on purpose: commercial auditors already live in SOC 2, ISO 27001, PCI DSS, and HIPAA. The exam opens straight into the technical literacy they are missing.

EXAM STRUCTURE & FORMAT

Attribute	Details
Format	Computer-based, online proctored
Time Limit	75 minutes
Total Questions	50
Multiple-Choice	40 questions (4 options, single correct answer)
Scenario-Based	10 questions (read a short artifact, answer 1–2 follow-up questions)
Passing Score	70% (35/50 correct)

Attribute	Details
Scoring	All questions weighted equally; scenarios count as individual questions
Negative Marking	None
Portfolio	None (knowledge exam only)
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. Higher-weighted domains have more questions, allowing comprehensive assessment of the most critical auditor competencies.

Difficulty Levels: Questions span multiple Bloom's Taxonomy levels, weighted toward Remember, Understand, and Apply. Auditors need to recognize, interpret, and evaluate engineered evidence, not build it, so there are no Create-level items.

Scenario Questions: Each scenario presents a short artifact (a snippet of Terraform, a pull request description, a CloudTrail event, a build log) plus one or two questions asking the auditor to evaluate it, identify the finding, or choose the right test procedure.

DOMAIN OVERVIEW

#	Domain	Weight	Questions	Focus
1	Cloud & Code Literacy for Auditors	20%	10	Cloud, shared responsibility, JSON/YAML, Git, reading code
2	Auditing Infrastructure as Code	15%	7	Reading Terraform, sampling resources, state & drift as evidence
3	Auditing CI/CD Pipelines	15%	8	Pipeline anatomy, SoD, build logs, approval gates
4	Cloud-Native Monitoring & Continuous Controls	15%	7	Testing automated controls, drift, audit logs, KSIs
5	Claude Code & AI Tooling for Auditors	15%	8	AI-assisted evidence pull, test procedures, guardrails
6	Evidence Evaluation in a GRC-Engineered Org	10%	5	Evidence quality, sampling, AWS Audit Playbook, workpapers
7	Applied Commercial Audit Scenarios	10%	5	SOC 2 Type II, ISO 27001, PCI DSS in the cloud
TOTAL		100%	50	

Domain Weight Explanation: Weights reflect how much of a modern auditor's real work each area represents. Cloud & Code Literacy is weighted highest because every other domain depends on it: an auditor who cannot read JSON, navigate a repository, or interpret a Terraform file cannot evaluate the evidence the other domains produce.

COURSE → DOMAIN CROSSWALK

Every video lesson is mapped to the domain it is tested under. Use this table to make sure your study time is distributed by exam weight. There are no hands-on labs: the CGE-AUD is a knowledge certification.

#	Domain	Video lessons covered
1	Cloud & Code Literacy for Auditors	01_01 · 01_02 · 01_03 · 01_04 · 01_05
2	Auditing Infrastructure as Code	02_01 · 02_02 · 02_03 · 02_04
3	Auditing CI/CD Pipelines	03_01 · 03_02 · 03_03 · 03_04
4	Cloud-Native Monitoring & Continuous Controls	04_01 · 04_02 · 04_03 · 04_04
5	Claude Code & AI Tooling for Auditors	05_01 · 05_02 · 05_03 · 05_04 · 05_05 · 05_06
6	Evidence Evaluation in a GRC-Engineered Org	06_01 · 06_02 · 06_03 · 06_04
7	Applied Commercial Audit Scenarios	07_01 · 07_02 · 07_03 · 07_04

DOMAIN 1: CLOUD & CODE LITERACY FOR AUDITORS

Weight: 20% (10 questions)

TASK STATEMENTS

1.1 Articulate why a modern auditor must read code and configurations directly rather than request screenshots

1.2 Apply the shared responsibility model to scope a SOC 2 or ISO 27001 audit of a cloud-native organization

1.3 Read JSON and YAML well enough to evaluate them as audit evidence

1.4 Navigate a GitHub repository and pull commit history as audit evidence

1.5 Read a Terraform configuration well enough to know what infrastructure it deploys

KNOWLEDGE STATEMENTS

- Cloud accounts, subscriptions, and projects; regions and availability zones; core service categories across AWS, Azure, and GCP
- The shared responsibility model and what it means for control scoping
- JSON anatomy (keys, values, nesting, arrays) and YAML basics
- Why every cloud artifact is JSON or YAML: reading a CloudTrail event and an IAM policy
- Git fundamentals: repositories, commits, branches, pull requests, and the audit trail buried in history
- HCL syntax at a glance: resource blocks, variables, and modules

SKILLS STATEMENTS

- Use the shared responsibility model to set audit scope boundaries
- Read a CloudTrail event and an IAM policy expressed in JSON
- Navigate a GitHub repository and pull commit history without using the command line
- Read a Terraform file and identify the infrastructure it provisions

BLOOM'S TAXONOMY DISTRIBUTION

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

DOMAIN 2: AUDITING INFRASTRUCTURE AS CODE

Weight: 15% (7 questions)

TASK STATEMENTS

2.1 Explain infrastructure as code and identify how it changes the audit approach for cloud-native organizations

2.2 Read a Terraform module and evaluate whether key SOC 2 / ISO 27001 controls are implemented

2.3 Design a sampling approach for an IaC-defined population of resources

2.4 Use IaC state and drift detection output as audit evidence and document it in workpapers

KNOWLEDGE STATEMENTS

- Declarative infrastructure and why IaC changes sampling and evidence
- What to look for first in a module: encryption-at-rest, logging, public-access blocks, required tags
- Traditional sampling versus populations defined in code; complete-population testing
- The Terraform state file, what drift means, and how state and drift reports serve as point-in-time evidence

SKILLS STATEMENTS

- Evaluate a Terraform module for encryption, logging, and public-access controls
- Spot red flags and control gaps in an IaC configuration
- Scope a defensible sample from a module-generated population
- Document IaC evidence (state, drift reports) properly in a workpaper

BLOOM'S TAXONOMY DISTRIBUTION

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

DOMAIN 3: AUDITING CI/CD PIPELINES

Weight: 15% (8 questions)

TASK STATEMENTS

3.1 Recognize the components of a CI/CD pipeline and read a pipeline definition file

3.2 Evaluate segregation of duties controls in a modern CI/CD environment

3.3 Use build logs and artifacts as audit evidence for change management and security testing controls

3.4 Identify a well-designed compliance-focused pipeline and recognize common control failures

KNOWLEDGE STATEMENTS

• Pipeline anatomy and stages (build, test, scan, deploy) across GitHub Actions, GitLab CI, and Jenkins

• Segregation of duties in the pipeline era: who can push code, approve pull requests, and deploy; what branch protections enforce

• What every pipeline run produces and how to map outputs to specific SOC 2 / ISO controls

• Manual versus automated approvals; SAST, DAST, and SCA; vulnerability scanning in the pipeline

SKILLS STATEMENTS

• Read a GitHub Actions workflow file

• Test segregation of duties using branch protections and approval settings

• Use build logs as control evidence in place of screenshots

• Recognize a mature compliant pipeline and flag the gaps in a weak one

BLOOM'S TAXONOMY DISTRIBUTION

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

DOMAIN 4: CLOUD-NATIVE MONITORING & CONTINUOUS CONTROLS

Weight: 15% (7 questions)

TASK STATEMENTS

4.1 Adapt audit test procedures for continuously-operating automated controls

4.2 Interpret drift detection output and use it to identify control failures

4.3 Request, read, and sample cloud-native audit logs as evidence for access and change controls

4.4 Evaluate management's key security indicators (KSIs) and determine whether they provide sufficient audit comfort

KNOWLEDGE STATEMENTS

- The shift from point-in-time sampling to evaluating continuous controls, and what evidence of operating effectiveness looks like when controls are automated
- What drift is and the tools that detect it (Terraform, AWS Config, cloud-native drift detection)
- Cloud-native audit logging: what CloudTrail, Azure Monitor, and GCP Audit Logs record, and common log gaps
- KSIs and KPIs in continuous monitoring; what makes a KSI trustworthy and how to validate the calculation itself

SKILLS STATEMENTS

- Design test procedures for automated, continuously-operating controls
- Read a drift report and turn it into an audit finding
- Request, read, and sample cloud-native audit logs
- Validate a KSI calculation and judge whether it provides sufficient audit comfort

BLOOM'S TAXONOMY DISTRIBUTION

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

DOMAIN 5: CLAUDE CODE & AI TOOLING FOR AUDITORS

Weight: 15% (8 questions)

TASK STATEMENTS

5.1 Explain why AI-assisted audit is becoming the baseline for evidence work, not a nice-to-have

5.2 Set up and run Claude Code as an auditor with no coding background

5.3 Use the GRC Engineering Club Claude Code project and the Auditor Skills Library to pull configurations and draft test procedures

5.4 Apply AI tooling to a worked control test, such as validating a SOC 2 CC6.1 control across an AWS account

5.5 Apply guardrails: review obligations, audit independence, and documentation requirements when AI is used in the workflow

KNOWLEDGE STATEMENTS

- Where AI-assisted audit fits in the modern evidence workflow
- Installing and running Claude Code without writing code

• The Club's Claude Code project and Auditor Skills Library: evidence-pull, control-mapping, and workpaper-drafting skills

• When NOT to trust AI output: review obligations, independence implications, and documentation requirements

SKILLS STATEMENTS

• Use Claude Code to pull configurations and analyze JSON evidence without writing code

• Invoke auditor skills to draft test procedures and workpaper language

• Document AI-assisted procedures so audit independence and the review trail are preserved

• Recognize when AI output requires human verification before it enters a workpaper

BLOOM'S TAXONOMY DISTRIBUTION

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

Featured tools: the GRC Engineering Club Claude Code project and the Auditor Skills Library, published with the course.

DOMAIN 6: EVIDENCE EVALUATION IN A GRC-ENGINEERED ORG

Weight: 10% (5 questions)

TASK STATEMENTS

6.1 Evaluate evidence quality in an automated environment and articulate criteria for sufficient evidence

6.2 Design a sampling strategy appropriate for continuously-generated evidence

6.3 Run and interpret the AWS Audit Playbook as a working example of evidence collection without screenshots

6.4 Write a workpaper that properly documents the testing of an automated control

KNOWLEDGE STATEMENTS

• Evidence quality criteria for a GRC-engineered org: authenticity, integrity, completeness; system-generated versus human-generated evidence

• Statistical versus judgmental sampling when populations are continuous; anomaly-based sampling and complete-population testing where feasible

• The AWS Audit Playbook (AJ Dehn, AuditOps.io): an open-source boto3 project that collects AWS evidence as JSON and generates machine-readable and PDF reports

• Documenting an automated control test: referencing IaC files, build logs, and JSON outputs; common workpaper anti-patterns

SKILLS STATEMENTS

- Apply evidence-quality criteria to system-generated artifacts
- Choose a sampling strategy appropriate for continuously-generated evidence
- Interpret AWS Audit Playbook output and integrate it into a real audit
- Write a defensible workpaper for an automated control test

BLOOM'S TAXONOMY DISTRIBUTION

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

Featured tool: the AWS Audit Playbook by AJ Dehn (AuditOps.io). "AWS audits, without screenshots."

DOMAIN 7: APPLIED COMMERCIAL AUDIT SCENARIOS

Weight: 10% (5 questions)

TASK STATEMENTS

7.1 Apply the full CGE-AUD toolkit to a SOC 2 Type II audit of a cloud-native SaaS

7.2 Apply ISO 27001 audit procedures to a cloud-native environment

7.3 Scope a PCI DSS assessment in the cloud and identify segmentation evidence requirements

KNOWLEDGE STATEMENTS

- SOC 2 Type II of a cloud-native SaaS: scoping, control selection, and test procedures built on IaC and CI/CD evidence
- How ISO 27001 Annex A controls map to cloud-native evidence; ISMS scoping in the cloud
- PCI DSS scoping in the cloud: the cardholder data environment, segmentation, shared responsibility, and common cloud PCI pitfalls

SKILLS STATEMENTS

- Scope and execute a SOC 2 Type II using engineered evidence end to end
- Map ISO 27001 Annex A controls to cloud-native artifacts
- Scope a PCI DSS assessment and identify the segmentation evidence required

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	1

Level	Questions
Understand	2
Apply	1
Analyze	0
Evaluate	1

SCENARIO-BASED ASSESSMENT (NO PORTFOLIO)

Why There Is No Portfolio

Auditors evaluate, they do not build. The CGE-AUD is a knowledge certification: there is no capstone, no lab setup, and no code to write. Instead, the exam proves you can read engineered artifacts and reach the right audit conclusion. Ten of the fifty questions are scenario-based for exactly this reason.

Scenario Question Format

Each scenario presents a short artifact and asks one or two questions. The artifact is the kind of evidence an auditor actually receives in a GRC-engineered engagement:

- **Terraform snippet:** evaluate whether encryption-at-rest, logging, or public-access controls are configured.
- **Pull request description:** assess change management and segregation of duties.
- **CloudTrail event (JSON):** identify the access or change activity and whether it is in scope.
- **Build log:** decide whether it is sufficient evidence for a security testing or change control.

Questions ask the auditor to evaluate the artifact, identify the finding, or choose the right test procedure. They are scored as individual questions, weighted equally with multiple-choice items.

CERTIFICATION MAINTENANCE

Validity Period

The CGE-AUD is valid for three years from the date of issuance. To keep the credential active across the cycle, candidates maintain it through one of the two paths below. After three years, candidates re-take the exam to continue holding the certification.

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: CEU Hours. Complete 15 Continuing Education Unit (CEU) hours within the three-year cycle.

Approved CEU Activities

Activity	CEU Hours
Attending GRC Engineering Club events and workshops	1 hour per event hour
Published article or blog post on audit or GRC engineering	2 hours
Open-source contribution to audit tooling	2–4 hours
Completing advanced training courses	1 hour per course hour
Mentoring new CGE-AUD candidates	1 hour per session (max 10)

Annual content review keeps the curriculum aligned with commercial framework updates (SOC 2 / SSAE revisions, ISO 27001 editions, PCI DSS versions, HITRUST CSF releases).

BLOOM'S TAXONOMY REFERENCE

Taxonomy Levels

Level	Description	Example Question Stem
Remember	Recall facts and basic concepts	"What does the shared responsibility model cover?"
Understand	Explain ideas or concepts	"Which best describes why build logs can replace screenshots?"
Apply	Use information in new situations	"Given this Terraform snippet, is encryption-at-rest configured?"
Analyze	Draw connections among ideas	"Which combination of evidence best supports this control?"
Evaluate	Justify a decision or course of action	"Is this KSI sufficient audit comfort? Defend your answer."

Distribution Across Exam

Auditors must recognize, interpret, and evaluate engineered evidence, not build it. The exam has no Create-level items.

Level	Total Questions
Remember	10
Understand	18
Apply	15
Analyze	5
Evaluate	2

CANDIDATE PREPARATION RECOMMENDATIONS

Study Strategy

1. **Start with Cloud & Code Literacy.** Domain 1 is the foundation for everything else. If you cannot read JSON, navigate a repo, or interpret a Terraform file, the other domains will not land.

2. Focus on High-Weight Domains. Cloud & Code Literacy (20%) deserves the most study time, followed by the four 15% domains.

3. Practice Reading Real Artifacts. Scenario questions put real Terraform, pull requests, CloudTrail events, and build logs in front of you. Practice on real repositories and logs, not summaries.

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

5. Watch the Video Course. About four hours of video, taught by AJ Yawn, a former SOC 2 auditor, pairs a technical primer with auditor-specific guidance in every domain.

6. Try Claude Code Early. Set up Claude Code and run the worked examples so Domain 5 is hands-on familiar, not theoretical.

Recommended Resources

- GRC Engineering Club Training Academy (CGE-AUD video course)
- CGE-AUD Study Guide (companion to this blueprint)
- The AWS Audit Playbook by AJ Dehn (AuditOps.io)
- AWS, Azure, and GCP audit logging documentation (CloudTrail, Azure Monitor, Cloud Audit Logs)
- Terraform documentation (terraform.io)
- SOC 2, ISO 27001, and PCI DSS commercial framework references

Contact Information

Website: www.grcengclub.com

Email: academy@grcengclub.com

Community: GRC Engineering Club (Patreon)