CGE-AZ EXAM BLUEPRINT

Certified GRC Engineer, Azure Specialty

Official Examination Guide

GRC Engineering Club Training Academy

Instructor: AJ Yawn, author of GRC Engineering for AWS

Version 1.0 | September 2026 | Launches September 18, 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, Azure Specialty (CGE-AZ) 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-AZ validates that a GRC engineer can build and operate a complete, automated GRC engineering pipeline on Azure: a system that discovers what is running, activates what is missing, stores immutable evidence, generates auditor deliverables, narrates them weekly, and remediates non-compliant resources through a least-privilege identity. It is the first cloud specialty in the CGE family. The one-liner: do not map controls to Azure. Build the pipeline that proves them.

Prerequisites: CGE-P is a hard prerequisite, enforced in the platform. The course re-teaches nothing CGE-P covered: no Terraform basics, no Git, no control theory, no OSCAL introduction. No Microsoft certification is required.

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
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
Capstone	REQUIRED. The full pipeline, deployed as code in your own subscription, graded 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-AZ holders build and fix controls, not just recognize them.

Scenario Questions: Each scenario presents a short artifact (a policy definition, a KQL query, a Terraform plan excerpt, a pipeline run log, a Defender assessment) plus one or two questions asking the engineer to identify the gap, choose the right effect, or fix the control.

DOMAIN OVERVIEW

#	Domain	Weight	Questions	Focus
1	Azure Foundations	15%	8	Tenant → management group → subscription → resource group; Entra ID, RBAC, managed identities; shared responsibility
2	The Azure GRC Toolkit	15%	7	Defender for Cloud, Azure Policy, Log Analytics + KQL, Functions, Cosmos DB, Blob Storage, each mapped to its AWS counterpart
3	Terraform on Azure	15%	8	The azurerm provider, OIDC and managed-identity auth, remote state, governance as code
4	The Evidence Plane	20%	10	Pipeline stages 1–3: discovery, conditional activation, and the immutable evidence store
5	Reporting & Narrative	15%	7	Pipeline stages 4–5: SAR, POA&M, framework report, OSCAL SSP from Cosmos only; the AI weekly digest
6	Enforcement & Operations	10%	5	Pipeline stage 6: auto-remediation through a least-privilege identity, the OPA gate, drift, the loop
7	Capstone	10%	5	Requirements, rubric, reference walkthrough, submission
TOTAL		100%	50	

Domain Weight Explanation: Domains 1 through 3 make you fluent: the language, the services, and Terraform on the azurerm 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 → DOMAIN CROSSWALK

Every lesson is mapped to the domain it is tested under. The course is 25 bite-sized video lessons totaling about 110 minutes, plus two text articles (00_02 How to Use This Course, 07_04 Certification Exam Guide) and six hands-on labs. The learning happens in the labs, not in the lectures.

#	Domain	Lessons covered
–	Course Introduction	00_01 · 00_02 (article)
1	Azure Foundations	01_01 · 01_02 · 01_03 · 01_04 (Lab 1)
2	The Azure GRC Toolkit	02_01 · 02_02 · 02_03 · 02_04 · 02_05 (Lab 2)
3	Terraform on Azure	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: AZURE FOUNDATIONS

Weight: 15% (8 questions)

TASK STATEMENTS

1.1 Map the Azure resource hierarchy (tenant, management group, subscription, resource group) and translate it from AWS Organizations vocabulary

1.2 Explain why assignment scope determines control inheritance and why the hierarchy is the control architecture

1.3 Read and evaluate any Azure role assignment as role plus principal plus scope

1.4 Distinguish Azure RBAC from Entra directory roles, and control-plane from data-plane permissions

1.5 Explain why the pipeline runs entirely on managed identities and select the right principal type for a workload

1.6 Apply the shared responsibility model to scope the customer-owned control categories and connect each to the pipeline stage that proves it

KNOWLEDGE STATEMENTS

• The four hierarchy levels: tenant (Entra ID), management groups (nestable to six levels), subscriptions, resource groups; assignments flow down

• The four principal types: users, groups, service principals, managed identities (system-assigned and user-assigned)

• Two role systems: Azure RBAC for the resource world, Entra directory roles for the directory world

• Control plane versus data plane: managing a resource is a separate permission set from touching its data

• Shared responsibility on Azure: Microsoft attests the platform; the customer engineers identity, configuration state, log routing and retention, policy guardrails, and evidence

• The sandbox pattern from Lab 1: mg-grc → mg-grc-sandbox, a tagged resource group, a scoped Reader assignment, and a budget alert

SKILLS STATEMENTS

- Translate an AWS Organizations structure (OUs, accounts, SCPs) into Azure hierarchy terms
- Read a role assignment listing and state who can do what, where, and whether it was assigned or inherited
- Stand up the sandbox hierarchy with cost guardrails and the teardown habit in place
- Place any Azure 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 AZURE GRC TOOLKIT

Weight: 15% (7 questions)

TASK STATEMENTS

2.1 Map the six pipeline services to their AWS counterparts and to the pipeline stage each one runs

2.2 Navigate Defender for Cloud as the discovery and assessment engine, distinguishing free foundational CSPM from paid Defender plans and the 30-day trial

2.3 Read any Azure Policy definition and choose the right effect from the ladder: audit, deny, modify, deployIfNotExists

2.4 Route the Activity Log and resource logs to Log Analytics and write KQL that answers an auditor's question directly from the data

2.5 Activate the toolkit in a subscription: a Defender plan, the NIST CSF 2.0 standard, log routing, and a first pull from the assessments API

KNOWLEDGE STATEMENTS

- The translation table: Security Hub → Defender for Cloud; Config → Azure Policy + Resource Graph; Lambda → Functions; S3 → Blob Storage (WORM); Audit Manager → Cosmos DB
- Defender assessments as continuous, timestamped control tests; secure score; the regulatory compliance dashboard and the CSF 2.0 standard
- Policy anatomy: if/then blocks, aliases, initiatives, assignments with parameters, exemptions as time-boxed risk acceptances, remediation through an assignment identity
- KQL anatomy (where, project, summarize) and log retention decisions
- Why the evidence store is a database you own (Cosmos DB) rather than a rented compliance service: own the schema, own the program

SKILLS STATEMENTS

- Name the service that runs each pipeline stage and its AWS counterpart

- Pull an assessment from the Defender API and read its rule, resource, status, and timestamp
- Choose audit, deny, modify, or deployIfNotExists for a given control requirement
- Write a KQL query over the Activity Log that answers who did what, when

BLOOM'S TAXONOMY DISTRIBUTION

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

DOMAIN 3: TERRAFORM ON AZURE

Weight: 15% (8 questions)

TASK STATEMENTS

3.1 Configure the azurerm provider (provider and features blocks, subscription targeting, 4.x conventions)

3.2 Choose the correct authentication path per context: az login locally, OIDC in CI, managed identity inside Azure

3.3 Stand up remote state in a storage account with versioning and locking, and treat the state storage itself as a controlled resource

3.4 Deploy a custom policy definition, an initiative, and a management-group assignment from Terraform, including the identity block remediation effects require

3.5 Deploy the pipeline's foundation stage from code and prove enforcement with a denied deployment

KNOWLEDGE STATEMENTS

- The azurerm provider and features block; provider pinning
- Authentication without stored secrets: OIDC federation for CI, managed identities for anything running in Azure
- Remote state in an Azure storage account: versioning, locking, and why state is audit-relevant
- azurerm_policy_definition with jsonencode, azurerm_policy_set_definition for initiatives, management-group-scoped assignments
- Why policy-as-code beats portal clicks: review, history, and repeatability
- The pipeline repo structure: staged root modules with separate state

SKILLS STATEMENTS

- Write a working azurerm provider configuration with correct auth for each context
- Bootstrap remote state and import hand-built resources into Terraform
- Deploy policy as code with an identity block and verify the assignment
- Demonstrate a denied deployment as proof that enforcement works

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: interrogate the environment with Terraform data sources and Azure Resource Graph, and emit a typed inventory plus gap map that downstream stages consume

4.2 Build the activation stage: conditionally enable Defender plans and the CSF 2.0 standard from code, driven entirely by discovery's outputs

4.3 Deploy the evidence store: Cosmos DB containers and document schema, the collector Function (timer → assessments API → Cosmos), and the WORM Blob container for artifacts

4.4 Wire the store's identities: collectors get data-plane write, humans get read, nobody holds account keys

4.5 Trace a single finding from Defender to an immutable stored record and defend the collect-once guarantee

KNOWLEDGE STATEMENTS

- Why discover before you act: idempotency, safe re-runs, and respecting what already exists
- The discovery output contract: inventory plus gaps
- azurerm_security_center_subscription_pricing and workspace settings in Terraform; for_each over the gap map
- The evidence document schema: assessments, frameworks, and mappings as data; runId and collectedAt lineage on every document
- WORM immutability on the reports container and what it proves to an assessor
- Collect once against NIST CSF 2.0, crosswalk to 800-53 as a data change, not a re-collection

SKILLS STATEMENTS

- Read discovery output and predict what activation will and will not change
- Deploy the full evidence plane in a subscription and trigger a collection run
- Verify documents in Cosmos and WORM on the reports container
- Trace one finding from the Defender portal to its stored, immutable record

BLOOM'S TAXONOMY DISTRIBUTION

Level	Questions
Remember	0

Level	Questions
Understand	1
Apply	4
Analyze	3
Evaluate	2

DOMAIN 5: REPORTING & NARRATIVE

Weight: 15% (7 questions)

TASK STATEMENTS

5.1 Name the six pipeline reports (SAR, POA&M, framework report, OSCAL SSP, weekly digest, ATO package), and explain what question each answers and for whom

5.2 Defend the Cosmos-only rule: report generators read from the evidence store, never from live platform APIs

5.3 Generate a POA&M and a SAR from your own evidence store, verify WORM storage and dated paths, and trace a report number back to its Cosmos document

5.4 Explain the narrative layer's architecture and guardrails: AI describes, never decides

KNOWLEDGE STATEMENTS

- The six deliverables and their audiences; OSCAL 1.1.2 as the machine-readable format
- Report schedules and run history: live timers, not manual bursts
- Why store-only reporting makes every report number reproducible by a stored query
- The narrative stage: AI Foundry model deployment, the digest Function (report JSON in, prose out), Communication Services delivery
- The AI guardrails: additive never load-bearing, grounded prompts, human-auditable inputs; no model touches a finding, a score, or a remediation

SKILLS STATEMENTS

- Run the report generators against a populated evidence store
- Complete a full number-to-source trace: the auditor's test, passed on your own pipeline
- Identify which deliverable an assessor is asking for from the question they ask
- 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
Evaluate	1

DOMAIN 6: ENFORCEMENT & OPERATIONS

Weight: 10% (5 questions)

TASK STATEMENTS

6.1 Design the enforcement stage: Azure Policy remediation flowing through one dedicated, least-privilege, user-assigned identity

6.2 Apply the escalation ladder (audit → dry-run → enforce) with a human approving each escalation

6.3 Run the complete loop: a violation introduced, detected, remediated with approval, and self-documented in the next collection's reports

6.4 Operate the pipeline: the OPA/conftest gate on the repo's own Terraform, and drift detection for both questions (does reality match code, and who is touching reality)

KNOWLEDGE STATEMENTS

- Remediation effects at scale and why every modify/deployIfNotExists assignment needs an identity block
- The remediation identity as a whitelist of its job, never Owner or Contributor
- Remediation evidence in the Activity Log: who fixed it is an auditable identity
- The CI gate: conftest/OPA against Terraform plans on every pull request, failing closed
- Drift detection: scheduled plans for code drift, an activity-log tripwire for out-of-band change

SKILLS STATEMENTS

- Deploy enforcement in dry-run and approve a remediation task
- De-escalate and re-escalate an effect through a reviewed parameter change
- Show an auditor the full loop: detected, fixed, re-collected, documented
- Demonstrate the gate blocking a non-compliant plan

BLOOM'S TAXONOMY DISTRIBUTION

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

DOMAIN 7: CAPSTONE

Weight: 10% (5 questions)

TASK STATEMENTS

7.1 State the capstone requirements: foundation, evidence plane, at least two reports generated from Cosmos only, and enforcement in dry-run, all deployed as code

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

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

7.4 Plan submission timing relative to the exam and the Defender 30-day trial window

KNOWLEDGE STATEMENTS

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

• Pass = weighted average of 70 or higher with no auto-fail triggers; grading is automated, 2 attempts with a 24-hour cooldown

• Submission format: a public repo link; the starter repo is github.com/GRCEngClub/cgeaz

• What distinguishes a strong build from a passing one: run history that accumulates, a gate proven to block, a documented trace

SKILLS STATEMENTS

• Calibrate your own build against the reference walkthrough

• Run the rubric checks (plan cleanliness, identity scopes, the trace, gate test, docs) on your own repo before submitting

• Read the published rubric at cert.grcengclub.com/rubric/cge-az 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-AZ certifies builders. The exam proves you know the platform; the capstone proves the pipeline runs. The capstone is REQUIRED: the full pipeline, deployed as code in your own Azure subscription, submitted as a repo link, and graded against a published rubric you can read from day one. Six labs build it stage by stage, so if you do the labs, the capstone is assembly, not invention.

What You Submit

A public GitHub repository containing, at minimum: Terraform (azurerm) for the governed foundation, the evidence store (Cosmos DB plus immutable Blob), and enforcement in dry-run; at least two report generators that read from the evidence store only, with real run history; a CI gate on the repo's own Terraform plus scheduled drift detection; a README that deploys from an empty subscription; and a CONTROLS.md mapping

every policy, collector, and gate rule to NIST CSF 2.0 categories.

How It Is Graded

Dimension	Weight	What It Measures
Infrastructure-as-Code Quality	20%	Staged root modules, pinned providers, clean plans, discovery-first activation
Control Implementation & Identity Design	20%	Deliberate effects, one whitelisted remediation identity, SoD between pipeline identities
Evidence Integrity & Traceability	20%	WORM proven, lineage-stamped collection, store-only reporting, crosswalk as data
Pipeline Operations	20%	Live schedules with real history, a gate proven to block, drift detection, escalation ladder
Documentation & Control Mapping	20%	Cold-readable repo, current CSF 2.0 mapping, blast-radius notes on enforcement

Pass = weighted average of 70 or higher AND no auto-fail triggers. Grading is automated: 2 attempts, with a 24-hour cooldown between them. Every score comes back with per-dimension results, the check outputs that informed them, and quoted file evidence. The full rubric, including auto-fail triggers, is published at cert.grcengclub.com/rubric/cge-az and detailed in the CGE-AZ Capstone Overview PDF.

CERTIFICATION MAINTENANCE

Validity Period

The CGE-AZ 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, including at least 20 Group A hours.

Content is reviewed annually against Azure service changes: Defender plan renames, azurerm major versions, and CSF crosswalk updates.

BLOOM'S TAXONOMY REFERENCE

Taxonomy Levels

Level	Description	Example Question Stem
Remember	Recall facts and basic concepts	"Which service is Azure's counterpart to AWS Security Hub?"
Understand	Explain ideas or concepts	"Why does a modify assignment require an identity block?"
Apply	Use information in new situations	"Given this policy definition, which effect enforces the requirement without breaking production?"
Analyze	Draw connections among ideas	"This Terraform plan activates a Defender plan unconditionally. What did discovery miss?"
Evaluate	Justify a decision or course of action	"Is this remediation identity least-privilege? Defend your answer from its role assignments."

Distribution Across Exam

The blueprint targets are L1 Remember 10%, L2 Understand 20%, L3 Apply 35%, L4 Analyze 25%, L5 Evaluate 10%. CGE-AZ holders build and fix controls, not just recognize them, so the weight sits on Apply and Analyze. 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, and the capstone assembles them. About 110 minutes of video versus 10+ hours of hands-on work: the learning happens in the labs.

2. Time the trial windows. Create the Azure free account and start the Defender 30-day trial together, so the $200 credit window and the trial window both cover Labs 2 through 6 and the capstone.

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 policy definitions, KQL queries, Terraform plan excerpts, pipeline logs, and Defender assessments. Read them the way the labs taught you.

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

6. Read the rubric before you build. The capstone rubric is published from day one. Calibrate against the reference walkthrough in 07_02 before you submit.

Recommended Resources

- GRC Engineering Club Training Academy (CGE-AZ video course and labs)
- CGE-AZ Study Guide (companion to this blueprint)

- CGE-AZ Capstone Overview (the rubric, submission requirements, and lab-to-capstone map)
- The lab and capstone starter repo: github.com/GRCEngClub/cgeaz
- The published capstone rubric: cert.grcengclub.com/rubric/cge-az
- Microsoft documentation for Defender for Cloud, Azure Policy, and Azure Monitor / KQL
- Terraform azurerm provider documentation (registry.terraform.io)
- NIST CSF 2.0 and SP 800-53 references; OSCAL 1.1.2 documentation

Contact Information

Website: www.grcengclub.com

Email: academy@grcengclub.com

Community: GRC Engineering Club (Patreon)