PREPKLOUD FIELD GUIDE SERIES

CCSK v5 Cloud Security Field Guide

A vendor-neutral, scenario-first guide to the twelve domains of the Certificate of Cloud Security Knowledge.

First edition · 2026 · prepkloud.com

About this guide

The Certificate of Cloud Security Knowledge (CCSK), from the Cloud Security Alliance (CSA), tests whether you understand how security responsibilities, controls, and risks change when workloads move to cloud. It is vendor-neutral. You will not be asked which button to press in one provider's console. You will be asked who is responsible, which control fits, and what to do first.

This guide follows the twelve CCSK v5 domains. Each chapter explains the concepts, gives a decision rule, works through a realistic scenario, and ends with recall prompts. Chapter 14 has 30 original practice questions, and Chapter 15 has a six-week plan, a decision cheat sheet, and a glossary.

Exam facts, as published on CSA's CCSK page when this guide was written: online and open book; 60 multiple-choice questions drawn from a larger pool; 120 minutes; 80% minimum passing score; the exam purchase includes two attempts. Formats, prices, and policies can change. Always confirm them on the official CSA CCSK page before you buy.

Independent publication: CCSK, CSA, Cloud Controls Matrix, CAIQ, and STAR are marks of the Cloud Security Alliance. PrepKloud is not affiliated with or endorsed by CSA. The primary study source is CSA's *Security Guidance for Critical Areas of Focus in Cloud Computing* v5 and the free CCSK v5 Prep Kit. This guide adds structure, not a replacement. All questions are original, not recalled or leaked exam items.

Contents

1. How the CCSK works and how to prepare

2. Cloud computing concepts and architectures

3. Cloud governance

4. Risk, audit, and compliance

5. Organization management

6. Identity and access management

7. Security monitoring

8. Infrastructure and networking

9. Cloud workload security

10. Data security

11. Application security

12. Incident response and resilience

13. Related technologies and strategies: AI and Zero Trust

14. Original practice questions and explanations

15. Six-week plan, cheat sheet, and glossary

CHAPTER 1 · EXAM METHOD

How the CCSK works and how to prepare

The CCSK is a *knowledge* certificate. There is no experience requirement, and CSA recommends only basic familiarity with security fundamentals: firewalls, encryption, secure development, and identity and access management. That makes it a common first cloud security credential for security staff, auditors, architects, developers, and operations engineers.

Open book does not mean easy. With 60 questions in 120 minutes you have about two minutes per question. Candidates who plan to search for every answer run out of time. Study as if the exam were closed book, and treat your references as a safety net for a small number of marked questions.

CCSK v5 domain	Chapter
1. Cloud Computing Concepts & Architectures	2
2. Cloud Governance	3
3. Risk, Audit & Compliance	4
4. Organization Management	5
5. Identity & Access Management	6
6. Security Monitoring	7

7. Infrastructure & Networking	8
8. Cloud Workload Security	9
9. Data Security	10
10. Application Security	11
11. Incident Response & Resilience	12
12. Related Technologies & Strategies	13

A four-step answer method

1. **Find the model:** is it IaaS, PaaS, or SaaS, and public, private, hybrid, or community cloud? Responsibility follows the model.
2. **Find the layer:** management plane, network, workload, application, data, or identity.
3. **Find the owner:** provider, customer, or shared. Most traps misplace this.
4. **Find the qualifier:** words such as *first*, *best*, *most important*, or *always* decide between two otherwise true answers.

What changed in v5

CSA consolidated 14 domains into 12. It expanded workload security (including serverless), application security, DevSecOps, and automation, and added AI and generative AI, Zero Trust, data lakes, and cloud telemetry. Detailed legal and regulatory coverage was reduced, and Security as a Service is no longer a separate topic. If you study from v4 material, check it against v5.

Recall

- Can I name all twelve v5 domains?
- Can I explain why open book does not remove the need to know the concepts?

Cloud computing concepts and architectures

CSA builds on the NIST definition of cloud computing. The five essential characteristics are **on-demand self-service**, **broad network access**, **resource pooling**, **rapid elasticity**, and **measured service**. Resource pooling (multi-tenancy) and self-service through APIs are what make cloud security different: tenants share infrastructure, and anyone with the right credentials can reconfigure the environment in seconds.

Service model	Provider typically manages	Customer always keeps
IaaS	Facilities, hardware, hypervisor, core network	OS, middleware, applications, network configuration, identities, data
PaaS	Everything in IaaS plus runtime and platform	Application code, configuration, identities, data
SaaS	Almost the entire stack	Identities and access, configuration settings, data classification and use

The shared responsibility model

Responsibility shifts toward the provider as you move from IaaS to SaaS, but it never reaches zero for the customer. Across every model, the customer remains accountable for its data, its identities and entitlements, and the configuration choices it

controls. Accountability also cannot be outsourced: if a SaaS provider is breached, the customer still answers to its regulators and users.

CSA's logical model

- **Infrastructure:** the physical and virtual compute, network, and storage.
- **Metastructure:** the protocols and interfaces that connect infrastructure to the other layers. This is the *management plane*, the APIs and consoles used to configure everything. It is the defining new security layer in cloud.
- **Applistructure:** applications and the services they are built on.
- **Infostructure:** the data and information itself.

> **Whose control is it?**
>
> A company runs a database on IaaS virtual machines. An attacker exploits an unpatched OS vulnerability. Patching the guest operating system is the customer's responsibility in IaaS. If the same data sat in a managed PaaS database service, patching the database engine would move to the provider, while access configuration would stay with the customer.

Exam trap

"The provider is responsible for security in SaaS" is incomplete. The provider secures the service, and the customer still secures who can access it, how it is configured, and what data goes into it.

Recall

- Can I list the five NIST essential characteristics?
- Can I explain why the metastructure (management plane) is the highest-value target?

Cloud governance

Governance defines *what* the organization expects and *who* is accountable. Security controls are how those expectations are enforced. In cloud, governance must handle a new reality: a significant part of the control environment is operated by a third party you cannot directly inspect.

Governance tools

- **Contracts and service agreements:** the primary tool for defining responsibilities, notification duties, data location, and exit terms. Anything not in the contract should be assumed to be unavailable.
- **Provider assessments:** reviewing the provider's documentation, certifications, and attestations before and during use.
- **Cloud Controls Matrix (CCM):** CSA's control framework for cloud, mapped to other standards, that shows which party typically owns each control.
- **CAIQ (Consensus Assessments Initiative Questionnaire):** a standard set of yes/no questions providers answer about their CCM controls.
- **CSA STAR registry:** a public registry where providers publish self-assessments (Level 1) and third-party audits or certifications (Level 2).

Governance in practice

Define a cloud policy before teams create accounts. Set approved services and regions, a tagging and ownership standard, classification-driven rules for sensitive data, and an exception process. Governance that arrives after hundreds of accounts already exist is expensive to retrofit.

> **Choosing a SaaS vendor**
>
> A business unit wants to adopt a new SaaS HR platform. Governance asks: what data will it hold, where is it stored, what attestations does the vendor publish (for example in STAR), what do the contract terms say about breach notification and data return, and who in the business owns the risk decision?

Recall

- Can I explain the relationship between CCM, CAIQ, and STAR?
- Why are contracts described as the primary governance tool with a provider?

CHAPTER 4 · DOMAIN 3

Risk, audit, and compliance

Cloud changes how you *obtain evidence*, not whether you need it. You usually cannot audit a hyperscaler's data centre yourself. Instead you rely on the provider's independent attestations and add your own evidence for the parts you control.

Key ideas

- **Compliance inheritance is partial.** A provider's ISO/IEC 27001 certificate or SOC 2 report covers the provider's scope. It says nothing about whether *your* storage bucket is public.
- **Pass-through audits:** you use the provider's audit artifacts for their layer and audit your own configuration for yours.
- **Artifacts:** reports, certificates, and control mappings you collect and review periodically, not once.
- **Continuous compliance:** because cloud configuration changes constantly, automated checks against policy (for example, posture management tools) complement periodic audits.
- **Jurisdiction:** data location and the provider's legal jurisdiction affect which laws apply and who can compel access.

Risk treatment

Risk decisions follow the familiar pattern: avoid, mitigate, transfer, or accept. Transferring risk to a provider through a contract transfers the operation of a control, not the accountability. The data owner still owns the residual risk.

> **Exam trap**
>
> "The provider is PCI DSS compliant, so our workload is compliant" is wrong. The provider's compliance is a necessary foundation. Your workload's compliance depends on your configuration, processes, and evidence.

Recall

- What does a provider attestation prove, and what does it not prove?
- Why is continuous compliance more important in cloud than on premises?

Organization management

Every provider offers a hierarchy for organizing resources: an organization or tenant at the top, then groupings such as organizational units, folders, or management groups, then accounts, subscriptions, or projects. How you use that hierarchy determines your blast radius, how policy is inherited, and how billing and ownership are tracked.

Design principles

- **Separate by environment and sensitivity:** production, non-production, and sandbox workloads in different accounts or subscriptions.
- **Use a landing zone:** a pre-built, governed baseline for new accounts with logging, identity integration, network patterns, and guardrails already in place.
- **Apply guardrails at the top:** organization-level policies can prevent actions such as disabling audit logs or deploying to unapproved regions, regardless of permissions granted lower down.
- **Protect the root:** the organization's root or top-level administrative identities are few, use strong MFA, are monitored, and are rarely used.
- **Centralize security services:** dedicated accounts for log archive and security tooling, which workload teams cannot

alter.

> **Limiting blast radius**
>
> A developer's credentials are phished. Because each application runs in its own account with preventive guardrails, the attacker can affect only the developer's sandbox and cannot disable centralized logging. One large shared account would have exposed everything.

Recall

- Why is a multi-account (or multi-subscription) structure a security control?
- What belongs in a landing zone baseline?

Identity and access management

In cloud, identity is the primary perimeter. Every management-plane action is an authenticated API call, so whoever holds valid credentials with enough entitlement controls the environment. Compromised credentials and excessive permissions are among the most common causes of cloud incidents.

Core concepts

Term	Meaning
Identity	A unique representation of a user, service, or workload.
Authentication	Proving the identity, ideally with multiple factors.
Authorization	Deciding what an authenticated identity may do.
Entitlement	A specific permission mapping an identity to an action on a resource.
Federation	Trusting an external identity provider, using standards such as SAML or OpenID Connect, so users keep one identity.

Decision rules

- **Federate** workforce identities from a central identity provider instead of creating separate local users in every cloud account.

- **Require MFA** for all human access, and phishing-resistant methods for privileged users.
- **Least privilege and just-in-time:** grant minimal standing access and elevate temporarily when needed.
- **RBAC vs ABAC:** role-based access is simpler to reason about, while attribute-based access scales better for dynamic conditions such as tags, device state, or location.
- **Workload identities:** let services use provider-issued, short-lived credentials (roles, managed or workload identities) rather than long-lived static keys embedded in code.
- **Review entitlements** regularly and remove unused permissions.

> **The leaked key**
>
> A static access key is committed to a public repository and used within minutes to launch resources. The fixes: revoke and rotate the key, investigate its activity in management-plane logs, replace static keys with short-lived workload identities, and add secret scanning to the pipeline.

Recall

- Why is identity called the new perimeter?
- What replaces static access keys for workloads?

Security monitoring

Cloud offers richer telemetry than most data centres, but only if it is enabled, collected, protected, and actually analysed. The CCSK expects you to know the main telemetry sources and their priority.

Telemetry source	What it shows
Management-plane (API/audit) logs	Who changed what, where, and when across the cloud account. Enable these first.
Service and resource logs	Activity within specific services, such as storage access or database queries.
Network flow logs	Connections between addresses and ports, without payloads.
Workload and application logs	Operating system, container, function, and application events.
Identity provider logs	Sign-ins, MFA events, and risky authentication.

Monitoring principles

- Send logs to a **central, separate, access-restricted** location so a compromised workload account cannot delete evidence.
- Use **posture management** to detect misconfigurations continuously, not just threats.
- Correlate cloud logs in a SIEM or equivalent analytics platform, and tune detections for cloud-specific behaviour

such as new access keys, disabled logging, or public exposure changes.

- Automate the response to well-understood, high-confidence findings.

> **Exam trap**
>
> Network-based intrusion detection is less useful in cloud. You often cannot place a tap on the provider's network, and most attacks target the management plane through legitimate APIs. The audit log is the first place to look.

Recall

- Which log source should be enabled and protected first?
- Why should logs live outside the workload account?

CHAPTER 8 · DOMAIN 7

Infrastructure and networking

Cloud networks are software-defined. Virtual networks, subnets, routes, and firewall rules are API-configured objects, which means they can be version-controlled, reviewed, and deployed as code, and also misconfigured in seconds.

Network controls

- **Security groups:** stateful, instance- or interface-level allow rules. The default building block of cloud segmentation.
- **Network ACLs:** stateless, subnet-level rules, often used as a coarse secondary layer.
- **Microsegmentation:** fine-grained isolation between workloads, which is easier in cloud because each workload can have its own rules.
- **Private connectivity:** private endpoints and private links let workloads reach provider services without traversing the public internet.
- **Software-defined perimeter and ZTNA:** grant access to specific applications after verifying identity and device, instead of placing users on a flat network through a VPN.

Infrastructure as code (IaC)

Defining infrastructure in templates enables peer review, automated policy checks before deployment, repeatable secure baselines, and drift detection. Treat IaC templates as sensitive: whoever can change them can change production.

> **Open management port**
>
> A scan finds SSH open to the internet on a production instance. Short-term: restrict the security group to approved sources. Long-term: remove direct administrative ports and use a managed session or bastion service, and block the pattern with a policy check in the IaC pipeline.

Recall

- Can I contrast stateful security groups with stateless network ACLs?
- How does IaC improve network security?

Cloud workload security

A workload is a unit of processing: a virtual machine, a container, or a serverless function. Each shifts responsibility differently.

Workload	Customer focus
Virtual machines	Hardened images, OS patching, endpoint protection, host configuration, secure metadata access.
Containers	Trusted base images, image scanning, registry access control, orchestrator (e.g. Kubernetes) configuration, runtime protection, least-privilege pods.
Serverless / FaaS	Function code and dependencies, function permissions, event-source trust, secrets handling, configuration. The provider manages the host.

Immutable workloads

Instead of patching running servers, build a new, hardened image, test it, and replace the old instances. Immutable workloads reduce configuration drift, make rollback simple, and let you disable remote login entirely, removing a common attack path.

Workload protection principles

- Scan images and dependencies before deployment and block known-critical vulnerabilities.

- Give each workload the narrowest identity and network access it needs.
- Collect workload telemetry centrally. Short-lived instances and functions disappear, and so does local evidence.
- For serverless, tightly scope each function's permissions. An over-privileged function is a common escalation path.

Recall

- How does responsibility change from VMs to containers to functions?
- What are two security benefits of immutable workloads?

Data security

Data security starts with knowing what you have: classification drives which controls apply. The CCSK then focuses on protecting data at rest, in transit, and in use, and above all on who controls the encryption keys.

Encryption and key management

Key option	Control	Typical trade-off
Provider-managed keys	Provider creates and manages keys	Simplest; protects against media theft but not against misuse of provider-level access
Customer-managed keys in the provider's KMS	Customer controls policy, rotation, and disabling	Good balance of control and operational effort
Bring your own key / hold your own key	Customer generates or keeps key material, sometimes in its own HSM	Most control; most operational complexity and availability risk

The core principle: encryption protects data only if **access to the keys is separated from access to the data**. If the same administrator can read the data and use the key, encryption adds little against that administrator.

Other data controls

- **Access controls** on storage, the most common failure point (for example, publicly readable buckets).
- **Tokenization and masking** to reduce the amount of sensitive data stored or exposed.
- **Data loss prevention** to detect and block sensitive data leaving approved locations.
- **Data lakes:** large shared repositories need consistent classification, fine-grained access, and monitoring, because many teams and tools read from them.
- **Lifecycle and deletion:** retention rules, and crypto-shredding (destroying keys) as a way to render data unrecoverable.

Recall

- Why does key separation matter more than the encryption algorithm?
- What is crypto-shredding?

Application security

Cloud-native applications are assembled from managed services, APIs, containers, and functions, and deployed many times a day. Security must be built into the delivery pipeline instead of added at the end.

Secure software development lifecycle

- **Design:** threat-model the application, including cloud-specific threats such as over-privileged service identities, exposed storage, and management-plane abuse.
- **Build:** static analysis (SAST), software composition analysis (SCA) of dependencies, secret scanning, and IaC scanning.
- **Test:** dynamic testing (DAST) and API testing against running environments.
- **Deploy:** signed artefacts, policy checks, and approvals in the pipeline.
- **Operate:** runtime monitoring, web application firewalls, and feedback into the backlog.

DevSecOps and the pipeline

The CI/CD pipeline is now part of your attack surface. It holds credentials that can deploy to production. Protect it with least-privilege pipeline identities, protected branches and required

reviews, isolated build environments, and audit logging. Store secrets in a secrets manager, never in code or pipeline variables visible to everyone.

> **Vulnerable dependency**
>
> A widely used open-source library has a critical vulnerability. Teams with SCA in their pipeline find every affected application within minutes, rebuild images with the patched version, and redeploy through the same pipeline. Teams without SCA have to search manually.

Recall

- Where do SAST, SCA, and DAST fit in the pipeline?
- Why is the pipeline itself a high-value target?

Incident response and resilience

The incident response lifecycle stays the same: preparation; detection and analysis; containment, eradication, and recovery; and post-incident activity. What changes is how you perform each step when you cannot touch physical hardware and resources can be created and destroyed through APIs.

Cloud-specific preparation

- Know the provider's support and incident-reporting channels and what the contract says about notification.
- Make sure management-plane and workload logs are enabled, centralized, and retained long enough.
- Pre-create an isolated forensic account and responder identities with the permissions needed to investigate.
- Script containment actions in advance so they can run quickly and consistently.

Containment and forensics in cloud

Cloud gives responders powerful options: snapshot a disk for forensic analysis, isolate an instance by changing its security group, revoke sessions and credentials, or redeploy a clean version from a known-good image. The trade-off is that evidence can disappear quickly when auto-scaling terminates instances, so capture evidence before destroying anything.

Resilience

Design for failure. Use multiple availability zones for high availability and multiple regions where the business case requires it. Define recovery time and recovery point objectives, back up data to a separate, access-restricted location (ideally with immutability), and test restores. Consider provider concentration risk for critical services.

> **Compromised instance**
>
> An instance begins mining cryptocurrency. The responder snapshots its volumes, moves it to an isolation security group, revokes the attached role's sessions, reviews the audit log for how it was compromised, and replaces it with a freshly built image.

Recall

- Which preparation step is most often missed in cloud IR?
- Why snapshot before terminating?

Related technologies and strategies: AI and Zero Trust

AI and generative AI

Organizations increasingly consume AI as a cloud service or build on hosted models. Apply the same questions you use for any service: what data goes in, where it is processed and retained, who can access the model and its outputs, and who is responsible for which layer. AI adds new risks to consider:

- **Prompt injection:** crafted input that makes a model ignore its instructions or disclose data.
- **Data leakage:** sensitive data sent in prompts, or exposed through model outputs and logs.
- **Over-privileged AI agents:** AI systems that can call tools or APIs need least privilege like any other workload identity.
- **Training data and model integrity:** poisoned data or tampered models.
- **Governance:** approved use cases, human oversight for high-impact decisions, and monitoring of AI usage.

Zero Trust

Zero Trust replaces implicit trust based on network location with explicit, continuous verification. Every request is authenticated

and authorized using identity, device posture, and context; access is least privilege; and systems are designed assuming a breach has already occurred. In cloud, Zero Trust is practical because identity-centric access, microsegmentation, and rich telemetry are available as services.

> **Exam trap**
>
> Zero Trust is a strategy, not a single product. A question that asks for a Zero Trust approach is looking for continuous verification and least privilege, not simply "buy a new firewall."

Recall

- Can I name three AI-specific cloud security risks?
- What are the core principles of Zero Trust?

Practice questions and explanations

Answer each question before reading the explanation. These are original scenarios, not recalled CCSK questions.

Concepts, governance, and compliance (1–8)

1. In which service model does the customer typically remain responsible for patching the guest operating system?

IaaS.

In PaaS and SaaS the provider manages the operating system layer.

2. Which layer of CSA's logical model represents the management plane?

Metastructure.

It covers the APIs and consoles that configure all other layers, which is why it is the highest-value target.

3. A company uses a SaaS CRM. Which responsibility remains with the company in every case?

Managing user access and the data it puts into the service.
Identity, entitlements, and data use never move fully to the provider.

4. What is the primary tool for defining security responsibilities between a customer and a provider?

The contract and service agreement.
Assume anything not in the contract is unavailable.

5. Which CSA resource is a standard questionnaire providers answer about their control implementation?

The CAIQ.
It is based on the Cloud Controls Matrix; responses can be published in the STAR registry.

6. A provider holds an ISO/IEC 27001 certificate. What does this prove about your deployment?

Nothing directly about your configuration; it covers the provider's certified scope.
Your own controls still need your own evidence.

7. Transferring a risk to a provider through a contract transfers what?

The operation of the control, not the accountability.
The data owner still owns the residual risk.

8. Why is continuous compliance monitoring particularly valuable in cloud?

Configuration changes constantly through APIs, so point-in-time audits go stale quickly.
Automated posture checks catch drift between audits.

Organization, identity, and monitoring (9–16)

9. What is the main security benefit of separating workloads into multiple accounts or subscriptions?

Reduced blast radius.
A compromise in one account does not automatically expose the others.

10. Which control can prevent any account in an organization from disabling audit logging, even for local administrators?

An organization-level preventive guardrail or policy.
Guardrails inherited from the top of the hierarchy override lower-level permissions.

11. Why is identity described as the primary perimeter in cloud?

Every management action is an authenticated API call, so valid credentials with enough permission control the environment.
Network location no longer decides access.

12. What should replace static access keys embedded in application code?

Short-lived credentials from a provider workload identity (role or managed identity).
They rotate automatically and cannot be leaked as a long-lived secret.

13. What is federation?

Trusting an external identity provider (for example via SAML or OIDC) so users keep one central identity.
It avoids separate local users in every cloud account.

14. Which log source should be enabled and protected first in a new cloud account?

Management-plane (API/audit) logs.
They record who changed what across the account.

15. Why should logs be stored in a separate, restricted account?

So an attacker who compromises a workload account cannot delete or alter evidence.
Separation protects log integrity.

16. Why is traditional network intrusion detection less effective in cloud?

Customers often cannot tap provider networks, and many attacks use legitimate management APIs.
Audit logs and posture monitoring become primary.

Infrastructure, workloads, and data (17–24)

17. Which is stateful: security groups or network ACLs?

Security groups.
Network ACLs are typically stateless and evaluated at the subnet level.

18. What is one security advantage of infrastructure as code?

Changes can be peer-reviewed and policy-checked before deployment.
It also enables repeatable baselines and drift detection.

19. What replaces patching running servers in an immutable workload model?

Building, testing, and deploying a new hardened image, then retiring the old instances.
This reduces drift and allows remote login to be disabled.

20. In serverless computing, which risk is the customer's primary concern?

Function code, dependencies, and over-privileged function permissions.
The provider manages the underlying host.

21. Why must container images be scanned before deployment?

To find known vulnerabilities and unapproved components before they run.
Fixing an image is far cheaper than responding to an exploited container.

22. What principle makes encryption effective against a malicious administrator?

Separating access to encryption keys from access to the data.
If one person holds both, encryption adds little against that person.

23. Which key-management option gives the customer the most control and the most operational burden?

Bring your own key / hold your own key.
Loss or unavailability of the key can make data inaccessible.

24. What is crypto-shredding?

Rendering data unrecoverable by destroying the keys that encrypt it.
It is useful when you cannot physically destroy provider media.

Applications, response, AI, and Zero Trust (25–30)

25. Which testing technique analyses dependencies for known vulnerabilities?

Software composition analysis (SCA).
SAST analyses your own source code; DAST tests a running application.

26. Why is the CI/CD pipeline a high-value target?

It holds credentials and authority to deploy to production.
Protect it with least-privilege identities, reviews, and audit logging.

27. An instance is compromised. What should the responder do before terminating it?

Capture evidence, such as a volume snapshot and relevant logs.
Termination can destroy evidence permanently.

28. Which preparation step enables fast, consistent cloud containment?

Pre-scripted containment actions and pre-provisioned responder access.
Improvising permissions during an incident wastes critical time.

29. An AI assistant can call internal APIs. Which control is most important?

Least-privilege permissions for the assistant's identity, with monitoring.
Treat AI agents as workload identities that can be manipulated by prompt injection.

30. Which statement best describes Zero Trust?

Continuously verify every request using identity and context, grant least privilege, and assume breach.
It is a strategy, not a single product.

Six-week plan, cheat sheet, and glossary

Week	Focus	Output
1	Domains 1–3: concepts, governance, risk and compliance	A shared responsibility table for IaaS, PaaS, and SaaS.
2	Domains 4–6: organization, IAM, monitoring	An account hierarchy sketch with guardrails and a logging plan.
3	Domains 7–8: networking and workloads	A control checklist for VM, container, and function workloads.
4	Domains 9–10: data and application security	A key-management decision tree and pipeline security checklist.
5	Domains 11–12: incident response, AI, Zero Trust	A cloud IR runbook outline; first full timed practice set.
6	Mixed timed practice and weak-domain review	Book the exam when timed practice scores are consistently above 85%.

Decision cheat sheet

- **IaaS / PaaS / SaaS:** customer owns OS and up / code and config / identities, config, and data.
- **CCM / CAIQ / STAR:** control framework / provider questionnaire / public registry.

- **Attestation:** covers the provider's scope, not your configuration.
- **Management-plane logs:** enable first, store separately.
- **Security group / network ACL:** stateful per-interface / stateless per-subnet.
- **Immutable workloads:** replace, don't patch in place.
- **Encryption:** strength depends on key separation.
- **SAST / SCA / DAST:** your code / your dependencies / running application.
- **IR in cloud:** prepare access and scripts, snapshot before destroying.
- **Zero Trust:** verify explicitly, least privilege, assume breach.

Open-book exam-day strategy

1. First pass: answer everything you know; mark uncertain questions.
2. Second pass: use your one-page domain index to look up only the marked set.
3. Watch qualifiers: *first*, *best*, *customer*, *provider*.
4. Keep about ten minutes to review marked answers.

Glossary

CAIQ

Consensus Assessments Initiative Questionnaire; provider answers about CCM controls.

CCM
Cloud Controls Matrix; CSA's cloud security control framework.

Crypto-shredding
Making data unrecoverable by destroying its encryption keys.

Federation
Trusting an external identity provider for authentication.

Immutable workload
A workload replaced from a new image instead of modified in place.

Landing zone
A governed baseline configuration for new cloud accounts.

Metastructure
The management plane: APIs and interfaces that configure cloud resources.

Microsegmentation
Fine-grained network isolation between individual workloads.

Posture management
Continuous detection of cloud misconfigurations against policy.

Shared responsibility
Division of security duties between provider and customer by service model.

STAR
CSA's Security, Trust, Assurance and Risk registry of provider assessments.

Zero Trust

A strategy of explicit, continuous verification and least privilege.

Before test day, read the official CSA Security Guidance v5 alongside this guide, rework every practice question you missed, and prepare your one-page domain index. The best answer places responsibility with the right party and applies the control at the right layer.