



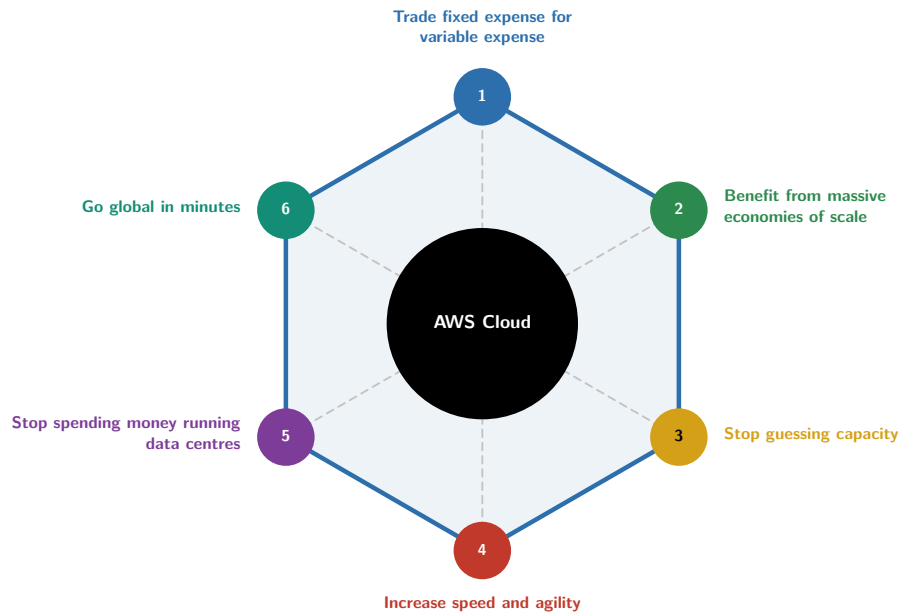
CLF-C02 Domain 1: Cloud Concepts

CLF-C02 exam guide Domain 1 (24 %), Overview of AWS whitepaper, Well-Architected Framework, AWS CAF, as of 2026 | Benefits and six advantages, design principles and the six pillars, migration (CAF, 7 Rs, tools), cloud economics (CapEx/OpEx, TCO, rightsizing, licensing)

CLF-C02 Domain 1 Well-Architected CAF 7 Rs

1.1 Benefits of the AWS Cloud

The six advantages of cloud computing (AWS wording)



Benefit	Means	Exam clue
High availability	keeps working when a component fails (Multi-AZ)	'minimal downtime', 'withstand AZ failure'
Elasticity	capacity grows and shrinks automatically with demand	'unpredictable traffic', 'scale in at night'
Scalability	can handle growth by adding resources	'grow to millions of users'
Agility	fast, cheap experiments; resources in minutes	'reduce time to market', 'experiment'
Global reach	deploy in many regions, edge locations near users	'low latency for users worldwide'
Fault tolerance	no user-visible interruption when a component fails	'zero downtime' (stronger than HA)

EXAM TIP

'Stop guessing capacity' = elasticity / Auto Scaling; 'trade fixed for variable expense' = pay-as-you-go instead of buying servers; 'go global in minutes' = regions + CloudFront. Questions often quote these phrases nearly word for word.

1.2 Design principles: Well-Architected

Pillar	Key design principles	Services that help
Operational excellence	operations as code, small reversible changes, anticipate failure, learn from operational events	CloudFormation, Systems Manager, CloudWatch
Security	strong identity foundation, traceability, security at all layers, automate, protect data in transit and at rest, prepare for events	IAM, KMS, CloudTrail, GuardDuty
Reliability	automatically recover, test recovery, scale horizontally, stop guessing capacity, manage change through automation	Auto Scaling, Multi-AZ, Route 53, Backup
Performance efficiency	democratise advanced tech, go global in minutes, use serverless, experiment more, mechanical sympathy	Lambda, CloudFront, Graviton, managed DBs
Cost optimisation	cloud financial management, consumption model, measure efficiency, stop spending on undifferentiated work, analyse and attribute spend	Cost Explorer, Budgets, Savings Plans
Sustainability	understand your impact, maximise utilisation, use managed services, reduce downstream impact	Graviton, Customer Carbon Footprint Tool

A mini Well-Architected review of a startup's web app

AWS Well-Architected

PILLAR	RISK PER QUESTION	HRI	MRI	ASKED
Security		1	0	2 ◀
Reliability		1	1	2 ◀
Cost		0	1	1
Operations		0	1	1
Performance		0	1	1
Sustainability		0	0	1

■ high risk (HRI) ■ medium risk (MRI) ■ no risk

TOP 4 IMPROVEMENTS

1	HIGH	Enable root MFA, delete root keys Security: Root MFA and no root keys?
2	HIGH	Auto Scaling group across 2 AZs + ALB Reliability: Single EC2 in one AZ?
3	MEDIUM	Restore test each quarter Reliability: Database backups tested?
4	MEDIUM	Schedule or scale in; Savings Plan for baseline Cost: Instances idle at night?

HRI = high-risk issue, MRI = medium-risk issue: a question answered with that risk. Fix HRI's first; the ◀ marks the pillar with the most HRI's.

TIP

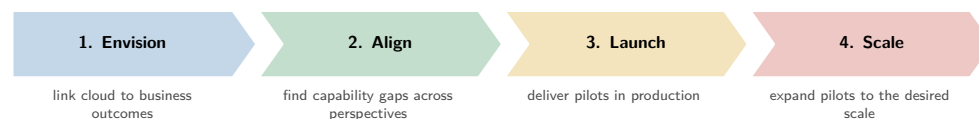
The AWS Well-Architected Tool (free in the console) runs this review on your workload with the official questions and lenses (Serverless, SaaS, Machine Learning, Generative AI...). Taught in depth in the well-architected-framework chapter.

1.3 Migration: CAF, 7 Rs and tools

Perspective	Owners	Focus
Business	CEO, CFO, COO	cloud investments accelerate business outcomes
People	CHRO, HR, leaders	culture, organisational structure, skills, change management
Governance	CFO, CIO, CDO, risk	programme and project management, benefits, risk, cloud financial management
Platform	CTO, architects, engineers	architecture, provisioning, modern apps, data, CI/CD
Security	CISO, CCO	identity, detection, infrastructure and data protection, incident response
Operations	IT operations managers	observability, event and incident management, availability, capacity

AWS CAF: six perspectives

CAF transformation phases



A portfolio placed on the 7 Rs

PORTFOLIO: 7 applications

retire 1	retain 1	rehost 1	relocate 1	replatform 1	repurchase 1	refactor 1
EFFORT BENEFIT		APPLICATIONS (REASON)				
Retire (1) decommission: switch it off: nobody needs it any more		Old intranet --- nobody uses it				
Retain (1) revisit: keep it where it is for now (not ready, recent upgrade, compliance)		Mainframe billing --- not ready; revisit in 2 years				
Rehost (1) lift and shift: same servers moved to cloud VMs, no code change		Web servers --- data centre exit, MGN lift-and-shift				
Relocate (1) hypervisor-level lift and shift: move the virtualised estate (e.g. VMware) as is		VMware farm --- VMware Cloud on AWS, no changes				
Replatform (1) lift, tinker and shift: a few cloud optimisations: managed database, containers		MySQL on VM --- move to RDS, little code change				
Repurchase (1) drop and shop: replace with a SaaS or another product		CRM --- drop and shop: SaaS CRM				
Refactor (1) re-architect: rebuild cloud-native: serverless, microservices, managed services		Order system --- re-architect to serverless for scale				

Dots: effort (top row) and cloud benefit (bottom row), 0 to 4, relative to the other Rs. Rehost and relocate move fastest; refactor costs most and gains most.

- **Benefits of migrating** reduced business risk, higher revenue (faster launches), lower operational costs, more efficiency, better sustainability
- **Migration tools in scope** Migration Evaluator (business case), Application Discovery Service, Migration Hub, MGN (servers), DMS + Schema Conversion (databases), DataSync, Snowball (offline data)
- **Programs** Migration Acceleration Program (MAP: method, tools, credits), AWS Professional Services and partners

1.4 Cloud economics

On premises	AWS
CapEx: buy servers, storage, network up front	OpEx: pay per second, request or GB
pay for peak capacity, idle most of the time	elastic: pay for what runs
facilities, power, cooling, hardware refresh every 3-5 years	AWS runs facilities and hardware (economies of scale)
staff for racking, patching, replacing parts	staff move to higher-value work (automation, products)

Three-year TCO: on premises vs AWS (illustrative figures)

Y

Compare the 3-year cost of keeping 10 servers on premises with migrating them to AWS.
Given: H = 80,000 USD hardware refresh now; P = 20,000 USD/year facilities, power, network; L = 40,000 USD/year staff time for hardware and OS; M = 4,500 USD/month on AWS after rightsizing (compute, storage, transfer, support); C_mig = 20,000 USD one-off migration; y = 3 years

Y

Step 1: On-premises TCO?

$$H + y \times (P + L) = 80,000 + 3 \times 60,000 = 260,000 \text{ USD}$$

include facilities and labour, not only hardware

T

Y

Step 2: AWS TCO?

$$C_{\text{mig}} + 12 \times y \times M = 20,000 + 36 \times 4,500 = 182,000 \text{ USD}$$

migration is a real cost

T

Y

Step 3: Saving?

$$(260,000 - 182,000) / 260,000 = 30 \%$$

relative to the on-prem baseline

T

Y

Step 4: What could improve it?

Savings Plans on the steady baseline, Graviton, turning off dev at night commitments and scheduling cut OpEx further

T

About 182,000 USD on AWS vs 260,000 USD on premises: about 30 % less over 3 years

T

Check: Migration Evaluator builds this case from measured utilisation; licences (BYOL vs licence included) can swing it either way.

Licensing — Licence-included (Windows, SQL Server, RDS for Oracle LI) bills per hour; BYOL reuses your licences, sometimes requiring Dedicated Hosts for per-socket or per-core terms.

Rightsizing — Measure (CloudWatch, Compute Optimizer), then downsize or change family; right-sizing before buying Savings Plans avoids committing to waste.

Automation — IaC and scheduled start/stop remove labour cost and idle hours; managed services remove patching and backups work.

Recap and self-test

Statement	True	False
1. 'Stop guessing capacity' is one of the six advantages → and a Reliability design principle	☒	☐
2. Sustainability is the newest Well-Architected pillar → added in December 2021 as the sixth pillar	☒	☐
3. Rehost means rewriting the application for serverless → rehost = lift and shift; refactor = re-architect	☐	☒
4. The People perspective of CAF covers culture and skills → owned by HR and leadership	☒	☐
5. Moving to the cloud turns OpEx into CapEx → it turns CapEx into OpEx	☐	☒
6. Repurchase means moving to a SaaS product → 'drop and shop'	☒	☐

Key takeaways

01

Six advantages

variable expense, scale economies, no guessing, agility, no data centres, global

02

Six pillars

ops, security, reliability, performance, cost, sustainability

03

CAF: 6 perspectives, 4 phases

business, people, governance / platform, security, operations

04

7 Rs

retire, retain, rehost, relocate, replatform, repurchase, refactor

05

TCO, not server price

facilities, labour, licences, migration

Q1 Which advantage describes paying only when you consume resources?

A Trade fixed expense for variable expense.

Q2 Which pillar includes 'automatically recover from failure'?

A Reliability.

Q3 Which pillar includes 'maximise utilisation' and minimising environmental impact?

A Sustainability.

Q4 Moving a MySQL VM to Amazon RDS with minimal code change: which R?

A Replatform (lift, tinker and shift).

Q5 Which CAF perspective covers identity, detection and incident response?

A Security.

Q6 Which CAF perspective covers architecture, provisioning and CI/CD?

A Platform.

Q7 Which service builds a migration business case from your data?

A Migration Evaluator.

Q8 Which pricing aspect lowers unit prices for every customer over time?

A Economies of scale.

Q9 What should you do before buying Savings Plans?

A Rightsize, so you do not commit to idle capacity.