

Introducing Cloud Architecting

CLOUD ARCHITECTING · WELL-ARCHITECTED FRAMEWORK · BEST PRACTICES · AWS GLOBAL INFRASTRUCTURE

STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

- 01 Architecture is applied best practice:** Cloud architecture is the practice of applying cloud characteristics to a solution that uses cloud services and features to meet an organization's **technical needs and business use cases**. Like building a house, the decision maker sets the requirements → the **architect** designs the blueprint → the **delivery team** builds it. Always **work backward from the business need**.
- 02 Six pillars, one framework:** The **AWS Well-Architected Framework** gives a consistent way to *evaluate* architectures and guidance to *implement* them, built from foundational questions AWS distilled from thousands of customer designs. Its six pillars: **Operational Excellence · Security · Reliability · Performance Efficiency · Cost Optimization · Sustainability**.
- 03 Ten best practices:** Implement **scalability**, **automate** the environment (including **IaC**), treat resources as **disposable**, use **loosely coupled** components, design **services, not servers**, choose the **right database**, avoid **single points of failure**, **optimize for cost**, use **caching**, and **secure your entire infrastructure** — basing each decision on **empirical data**, not assumptions.
- 04 Global hierarchy:** The AWS Global Infrastructure nests **Region** → **Availability Zone** → **data center**, plus **edge locations** and **regional edge caches** (PoPs) at the network edge. A **Region** is a geographic area of **two or more AZs**; an **AZ** is one or more data centers designed as an **independent failure zone**. You never pick a data center — only a Region and, for some services, an AZ.
- 05 Regions are isolated:** Regions are **isolated from one another** for fault tolerance — resources are **not automatically replicated** across Regions, so cross-Region replication is **your responsibility**. Choose a Region by **compliance and latency**, deploy across **multiple AZs** for resiliency, and note that Regions added after **March 20, 2019** are **disabled by default**.

02 KEY TERMS

TERM	DEFINITION
Cloud architecture	The practice of applying cloud characteristics to a solution that uses cloud services and features to meet an organization's technical needs and business use cases
Cloud architect	Owner of an organization's cloud computing architecture — plans strategy, researches services and workloads, and designs and builds solutions that align technology to business goals
AWS Well-Architected Framework	A guide of foundational questions and best practices, organized in six pillars, that provides a consistent approach to evaluate and improve cloud architectures
AWS WA Tool	Self-service tool in the AWS Management Console that reviews a workload against current best practices and delivers a step-by-step action plan
Design trade-off	A deliberate exchange of one quality for another (for example, consistency and durability for lower latency); should be based on empirical data
Anti-pattern	A bad solution design to avoid — such as configuration drift, hardcoded IP addresses, or a single database serving as a single point of failure
Infrastructure as code (IaC)	Provisioning infrastructure with code instead of manual steps — deploy duplicate environments, cut configuration errors, and propagate changes consistently
Loose coupling	Placing a managed intermediary (a load balancer or a message queue) between layers so failures and scaling are handled automatically
Single point of failure	A component whose failure takes the whole system down; mitigated with standby replicas, on-demand launches, or managed services

TERM	DEFINITION
AWS Region	A physical geographic area containing two or more Availability Zones, isolated from other Regions and connected by the AWS backbone network
Availability Zone (AZ)	One or more data centers in a Region, designed as an independent failure zone with discrete power, networking, and connectivity
Point of presence (PoP)	An edge location or regional edge cache at the network edge that caches content closer to users to lower latency

03 THE SIX PILLARS OF THE WELL-ARCHITECTED FRAMEWORK each pillar has its own whitepaper

PILLAR	WHAT IT ADDRESSES	KEY DESIGN PRINCIPLES
Operational Excellence	Run and monitor systems to deliver business value; continually improve processes	Deliver business value; continually improve; view the entire workload as code
Security	Protect information, systems, and assets through risk assessment and mitigation	Strong identity foundation; maintain traceability; security at all layers; risk assessment and mitigation
Reliability	Recover from disruptions and dynamically acquire resources to meet demand	Recover quickly; dynamically meet compute demand; mitigate disruptions
Performance Efficiency	Use computing resources efficiently and keep that efficiency as demand changes	Choose and maintain efficient resources; democratize advanced technologies; employ mechanical sympathy
Cost Optimization	Deliver value at the lowest price point; iterative and ongoing	Measure efficiency; eliminate unneeded expense; adopt the right consumption model; use managed services
Sustainability	Maximize efficiency and reduce waste; minimize the downstream environmental impact	Establish sustainability goals; maximize utilization; choose efficient hardware and software; reduce downstream impact

04 TEN BEST PRACTICES FOR BUILDING ON AWS and the anti-pattern each one avoids

BEST PRACTICE	CORE IDEA	ON AWS
Implement scalability	Handle changes in demand automatically at every layer; be elastic	CloudWatch alarms + EC2 Auto Scaling
Automate your environment	Auto-provision, terminate, and configure resources; respond to failures	CloudWatch + EC2 Auto Scaling; IaC
Treat resources as disposable	Think of infrastructure as software; replace rather than repair	Identical configs; stop idle resources
Use loosely coupled components	Put a managed intermediary between layers	Load balancers (ELB) and message queues
Design services, not servers	Don't limit the design to EC2 servers	Lambda, SQS, DynamoDB, ELB, SES, Cognito, S3
Choose the right database	Match the technology to the workload, not the reverse	Purpose-built AWS database services
Avoid single points of failure	Assume everything fails, then design backward	Standby replica; managed services
Optimize for cost	Trade fixed expense for variable; provision only what you need	Right-size; stop idle; managed services
Use caching	Store data in an intermediary to cut redundant retrieval	Amazon CloudFront in front of Amazon S3
Secure your entire infrastructure	Build security into every layer	Least privilege, MFA, encryption, security groups

01 Application server crashes → 02 Amazon CloudWatch detects the unhealthy instance → 03 EC2 Auto Scaling launches and configures an identical server → 04 An alarm notifies the administrator → 05 CloudWatch logs the action to a change management solution

06 EXAM TRAPS

where the knowledge check gets you

- × “The Well-Architected Framework has five pillars.” — It has **six**: Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization, and **Sustainability**. (In the WA Tool you answer questions across the first five areas.)
- × “Resources in one Region are automatically replicated to other Regions.” — **False**. Regions are isolated; data you store in a Region is **not** replicated outside it. Cross-Region replication is **your** responsibility.
- × “Every AWS Region is available to your account by default.” — Regions introduced **after March 20, 2019** (such as Hong Kong and Bahrain) are **disabled by default** and must be enabled before use.
- × “An Availability Zone is a single data center.” — An AZ is **one or more** data centers (some have as many as six), and **no data center belongs to two AZs**. You choose a **Region**, and for some services an **AZ** — never a specific data center.
- × “Avoiding single points of failure means duplicating every component.” — Not necessarily. Depending on your SLAs, use **automated on-demand launches** or **managed services** that AWS repairs for you.
- × “Amazon EC2 is the first and only building block for any solution.” — **Design services, not servers**: containers, serverless (Lambda), queues (SQS), managed data (DynamoDB), and off-server assets (S3) are often cheaper and more performant.
- × “The AWS WA Tool fixes your workload for you.” — It **reviews** your workload against best practices and delivers a **step-by-step action plan** — you implement the improvements.

07 SELF-QUIZ

answers are upside-down below

Q1 How many pillars does the AWS Well-Architected Framework have, and what are they?

Q2 Which pillar addresses recovering from disruptions and dynamically acquiring resources to meet demand?

Q3 “View the entire workload as code” is a design principle of which pillar?

Q4 “Democratize advanced technologies” and “employ mechanical sympathy” belong to which pillar?

Q5 Which self-service tool reviews a workload against AWS best practices and returns an action plan?

Q6 On what should design decisions and trade-offs be based, rather than on assumptions?

Q7 What are the two primary AWS solutions for decoupling the layers of a system?

Q8 List the AWS global infrastructure hierarchy from largest to smallest.

Q9 When you store data in a Region, whose job is it to replicate that data to another Region?

Q10 What are the two types of AWS points of presence (PoPs)?

ANSWER KEY (rotate the page)
 01 Six — Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization, and Sustainability. 02 Reliability. 03 “View the entire workload as code” is a design principle of the AWS Well-Architected Tool (WA Tool). 04 Empirical data (for example, load testing and benchmarking). 05 Load balancers and message queues. 06 Region — Availability Zone — data center. 07 Yours — Regions are isolated and do not replicate automatically. 08 Edge locations and regional edge caches.