

# Implementing Monitoring, Elasticity, and High Availability

CLOUDWATCH & EVENTBRIDGE · EC2 AUTO SCALING · DATABASE SCALING · ELASTIC LOAD BALANCING · ROUTE 53  
FAILOVER · WELL-ARCHITECTED HA

STUDY & REVIEW

## 01 MUST KNOW

if you remember five things, remember these

- 01 Monitor, then react:** **CloudWatch** collects metrics and logs from AWS services across Regions into one repository; a **CloudWatch alarm** watches a single metric over a period and — only on a *sustained* state change — sends its action to an **SNS topic** or an **Auto Scaling policy**. **EventBridge** then routes the resulting events to targets.
- 02 Elasticity = scaling:** *Elasticity* lets infrastructure expand and contract as demand changes; *scaling* is how you achieve it. **Vertical** scaling resizes one resource (hardware-bound, usually needs downtime); **horizontal** scaling adds or removes resources — *scale out* / *scale in* — and is the path to internet-scale high availability.
- 03 EC2 Auto Scaling group:** a logical collection of EC2 instances spanning Availability Zones, sized by **minimum** / **maximum** / **desired** capacity and built from a **launch template**. It integrates with ELB, replaces unhealthy instances, balances across AZs, and is *free* — you pay only for the instances.
- 04 Pick the scaling mechanism:** **scheduled** actions for predictable date-and-time changes, **dynamic** policies (target tracking, step, simple) for spiky metric-driven load, and a **predictive** policy that forecasts from historical patterns with machine learning. Dynamic and predictive can be combined.
- 05 Design for HA:** **ELB** distributes traffic to healthy targets across multiple AZs (ALB = layer 7, NLB = layer 4, GWLB = layer 3); **Route 53** adds DNS-level *failover* across Regions. Well-Architected HA = multiple locations, fail over to healthy resources, notify, and scale dynamically.

## 02 KEY TERMS

| TERM                                   | DEFINITION                                                                                                                                                           |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Amazon CloudWatch</b>               | Metrics and logs repository; collects built-in and custom metrics across Regions, calculates statistics, draws dashboards, and raises alarms                         |
| <b>CloudWatch alarm</b>                | Watches a single metric over a set period and invokes an action on a sustained state change — a notification to an SNS topic or an Auto Scaling policy               |
| <b>Namespace</b>                       | Container that isolates metrics so different applications are not aggregated together; AWS metrics use the AWS/service form (for example, AWS/EC2)                   |
| <b>Metric / dimension</b>              | A metric is a time-ordered set of data points with a unit; a dimension is a name-value pair (such as InstanceId) that characterizes and filters it                   |
| <b>Amazon EventBridge</b>              | Serverless event router; event buses fan events out to many targets and pipes handle point-to-point, using rules that match an event pattern or a schedule           |
| <b>Elasticity</b>                      | The ability of infrastructure to expand and contract as capacity requirements change                                                                                 |
| <b>Vertical vs. horizontal scaling</b> | Vertical replaces a resource with a different-sized one; horizontal adds or removes resources (scale out / scale in)                                                 |
| <b>EC2 Auto Scaling group</b>          | Managed collection of EC2 instances across Availability Zones governed by minimum, maximum, and desired capacity settings                                            |
| <b>Launch template</b>                 | Versioned definition of the instance configuration (AMI, instance type, purchase mix) an Auto Scaling group uses to launch instances; replaces launch configurations |
| <b>Target tracking policy</b>          | Dynamic scaling policy that keeps a chosen metric at or near a target value, managing its own CloudWatch alarms — like a thermostat                                  |

| TERM                | DEFINITION                                                                                                                                                        |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Read replica</b> | Read-only, asynchronously updated copy of a database for horizontal read scaling; can be promoted, but is not a substitute for Multi-AZ failover                  |
| <b>Hosted zone</b>  | Route 53 container of records defining how to route traffic for a domain and its subdomains; public zones route internet traffic, private zones route VPC traffic |

### 03 VERTICAL VS. HORIZONTAL SCALING

two ways to change capacity

| VERTICAL (SCALE UP / DOWN)                                                                                                                                                                                                                                                                                                                                    | HORIZONTAL (SCALE OUT / IN)                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>resize one resource</i> <ul style="list-style-type: none"> <li>— Replace a resource with a larger or smaller one (more RAM, CPU, I/O, networking)</li> <li>— App and data transfer to the new resource → usually <b>downtime</b></li> <li>— Hardware-bound: eventually reaches a limit</li> <li>— Not always cost-efficient or highly available</li> </ul> | <i>add or remove resources</i> <ul style="list-style-type: none"> <li>— Adding resources = <b>scale out</b>; ending them = <b>scale in</b></li> <li>— Apps and data are automatically moved to added resources</li> <li>— Builds internet-scale, elastic applications</li> <li>— Foundation for high availability across AZs</li> </ul> |

### 04 EC2 AUTO SCALING MECHANISMS

match the mechanism to the workload

| MECHANISM                        | SCALES ON                                | BEST FOR                                                                            |
|----------------------------------|------------------------------------------|-------------------------------------------------------------------------------------|
| <b>Scheduled action</b>          | A specific date and time                 | Predictable, known changes in load                                                  |
| <b>Dynamic — target tracking</b> | Keeps a metric at or near a target       | Steady state (for example, CPU near 50%)                                            |
| <b>Dynamic — step / simple</b>   | CloudWatch alarm thresholds              | Step scales to the size of the breach; simple ignores alarms until scaling finishes |
| <b>Predictive</b>                | Historical patterns via machine learning | Cyclical or recurring traffic; slow-to-initialize apps                              |

### 05 TYPES OF AWS LOAD BALANCERS

know the OSI layer and the use case

| LOAD BALANCER            | OSI LAYER       | USE IT FOR                                                                        |
|--------------------------|-----------------|-----------------------------------------------------------------------------------|
| <b>Application (ALB)</b> | 7 — application | HTTP/HTTPS and content-based routing for application architectures                |
| <b>Network (NLB)</b>     | 4 — transport   | TCP/UDP/TLS, static IPs, millions of requests per second at ultra-low latency     |
| <b>Gateway (GWLB)</b>    | 3 — network     | Third-party virtual appliance fleets (firewalls, IDS/IPS) via the GENEVE protocol |
| <b>Classic (CLB)</b>     | 3 & 7           | Previous-generation EC2-Classic; only when upgrading is not feasible              |

### 06 HOW A DYNAMIC SCALE-OUT HAPPENS

metric breach to a new healthy instance

**01** Metric crosses the threshold (for example, **CPU > 70%**) → **02 CloudWatch alarm** changes state and stays breached → **03** Alarm invokes the **scaling policy** → **04** Group launches an instance from the **launch template** → **05 ELB** registers it and routes traffic — never exceeding maximum

- ✗ “A CloudWatch alarm fires the instant a metric crosses its threshold.” — It acts only on a **sustained** state change held for the specified number of **evaluation periods**, not on a momentary spike.
- ✗ “Vertical scaling makes a workload highly available.” — Vertical scaling is hardware-bound and usually needs **downtime** to move data; high availability comes from **horizontal** scaling across multiple Availability Zones.
- ✗ “During a big spike, Auto Scaling keeps adding instances past the maximum.” — The group never exceeds **maximum** capacity (or drops below **minimum**), even if more scale-out notifications arrive.
- ✗ “A read replica is an automatic failover standby like Multi-AZ.” — A read replica is **read-only** and **asynchronous**; it can be manually **promoted**, but it is *not* a replacement for Multi-AZ automatic failover.
- ✗ “An Application Load Balancer works at layer 4.” — **ALB is layer 7** (application); **NLB is layer 4** (transport), and **GWLB is layer 3** (network).
- ✗ “Multivalue answer routing is a load balancer.” — It returns multiple **healthy** IP addresses over DNS to improve availability, but it is **not** a substitute for an actual load balancer.
- ✗ “Launch configurations are the current way to define Auto Scaling instances.” — **Launch templates** are the preferred and required mechanism; launch configurations are unsupported for instance types released after December 2022.
- ✗ “EC2 Auto Scaling is an extra paid service you must run.” — Amazon EC2 Auto Scaling is available **free of charge**; you pay only for the EC2 instances it launches.

## 08 SELF-QUIZ

answers are upside-down below

- Q1** A CloudWatch alarm's action can target which two kinds of destination?
- Q2** Before invoking its action, what must a CloudWatch alarm observe about the metric state?
- Q3** Which EventBridge construct is a point-to-point integration between exactly one source and one target?
- Q4** Name the two things an EventBridge rule can match on to run.
- Q5** What are the three capacity settings that govern an EC2 Auto Scaling group?
- Q6** Which dynamic scaling policy keeps a metric at or near a chosen value, managing its own alarms?
- Q7** Which Auto Scaling mechanism uses machine learning on historical traffic to forecast capacity?
- Q8** At which OSI layer does a Network Load Balancer operate?
- Q9** What HTTP status code must a target return to pass an ELB health check?
- Q10** Which Route 53 routing policy serves a primary resource and switches to a secondary when the primary is unhealthy?

ANSWER KEY (rotate the page)

**Q1** An Amazon SNS topic and an Auto Scaling policy **Q2** A sustained change held for the specified number of evaluation periods **Q3** A pipe (Amazon EventBridge Pipes) **Q4** An event pattern, or a schedule **Q5** Minimum, maximum, and desired capacity **Q6** Target tracking **Q7** Predictive scaling **Q8** Layer 4, the transport layer **Q9** 200 **Q10** Failover routing