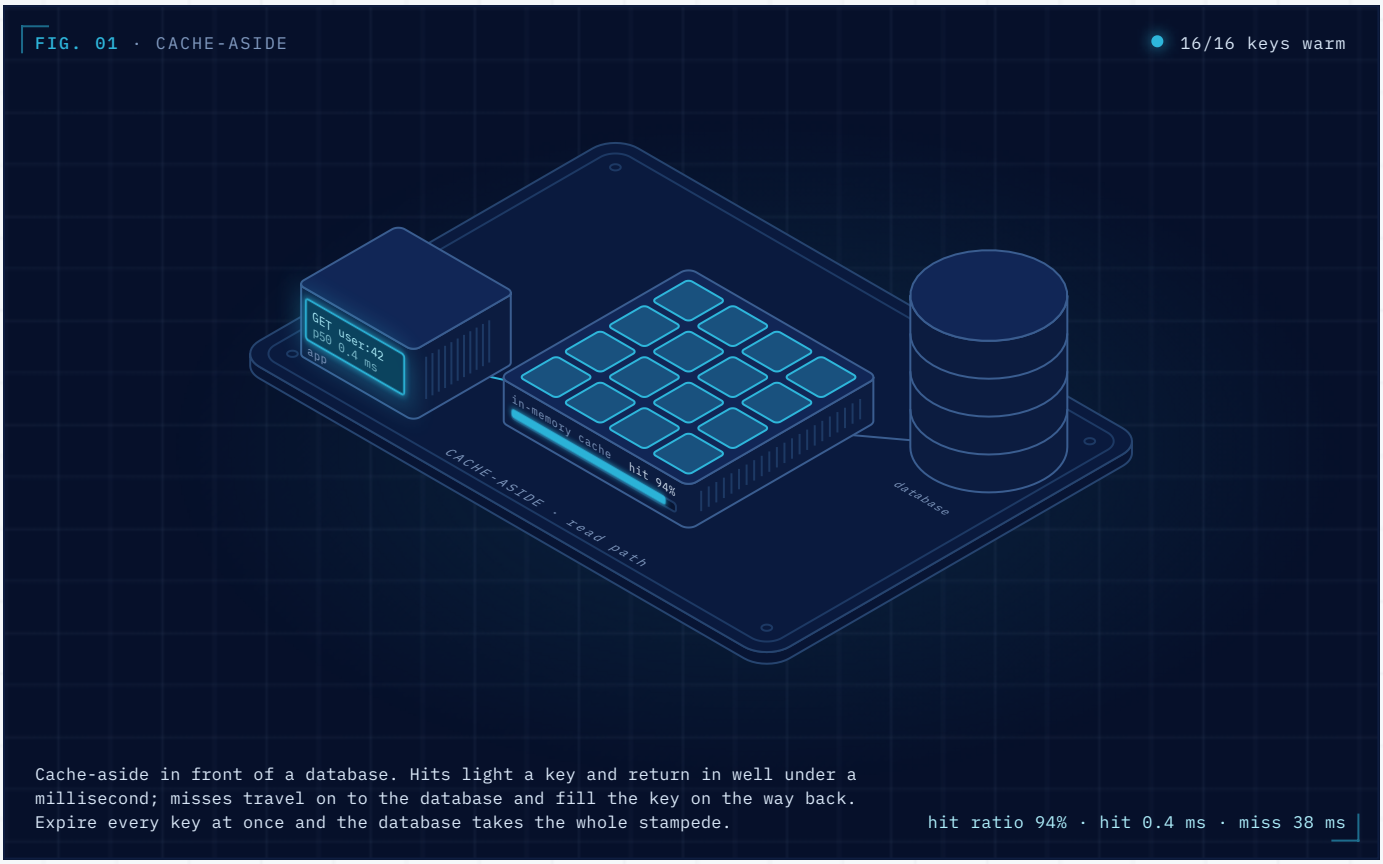
TRAINING & CONSULTING · CACHING

# Caching that makes systems *faster*, not wrong.

**Consulting and training on Redis, Valkey and caching architecture.** What to cache, for how long, and what happens when the cache is cold, full or down — we design cache layers that cut latency and database load without serving stale data.




**< 1 ms**
Typical cache read — the latency we design for

**3 days**
From data structures to clusters and failover

**≤ 15**
People per class, with a lab environment each

**EN · TR**
Delivered in English or Turkish, remote or on-site

### CONSULTING

**Cache the right things.** Hot paths, keys, TTLs and invalidation designed together — and measured under real load.

### TRAINING

**Know the failure modes.** Stampedes, stale reads, hot keys and failovers, rehearsed in labs before they happen in production.

### SUPPORT

**Stay ahead of growth.** Capacity reviews, upgrades and incident help, including the Redis ↔ Valkey decision.

02 · CONSULTING

# Fast reads, *correct* data, a bill that makes sense.

A cache is a second copy of your data with its own rules for going stale. We map where it sits in your request paths, decide what each key is allowed to be wrong about and for how long, and make sure the system survives the cache being cold, full or gone.

01

### Cache architecture review

Which reads to cache, key design, TTLs and invalidation paths across API, application and CDN layers.

YOU GET

A cache map of your hot paths with hit-ratio targets.

02

### Latency & performance

Hot keys, big keys, slow commands, pipelining, connection pools and client-side caching.

YOU GET

p99 fixes measured under load, before and after.

03

### Consistency & invalidation

Stale reads, write races, stampedes and thundering herds; event-driven invalidation from CDC or Kafka.

YOU GET

An invalidation design with stampede protection built in.

04

### High availability & scaling

Replication, Sentinel, Redis Cluster and hash slots, persistence choices, failover drills.

YOU GET

An HA topology and a rehearsed failover runbook.

05

### Memory & cost

Eviction policies, data-structure choice, compression, fragmentation and right-sizing.

YOU GET

A memory-per-key model and a smaller bill.

06

### Beyond caching

Sessions, rate limiting, Streams as queues, leaderboards, distributed locks — and when not to use Redis.

YOU GET

Reviewed patterns with their failure modes covered.

## SIGNALS IT IS TIME TO CALL

- Database CPU spikes when many keys expire together
- Users see old prices, stock or permissions
- p99 latency jumps during traffic peaks
- Memory is full and evictions keep climbing
- A failover takes minutes — or loses writes
- One key takes most of the traffic

## HOW AN ENGAGEMENT RUNS

01 **Scoping call** — 45 min · free
We hear the problem, agree goals and the numbers success is judged by.

02 **Assessment** — 1 week
Read-only access, diagnostics, logs and short interviews. Findings in writing.

03 **Fix & tune** — 1–4 weeks
Changes made with your team in small steps, each measured before and after.

04 **Hand-over** — ongoing
Runbooks, a focused workshop for your engineers and optional retained support.

FIELD EXPERIENCE: Retail & e-commerce · Banking & finance · Media · Gaming · SaaS · Public sector

03 · TRAINING

# From data structures to *failover* drills.

Three days that move from how Redis stores data, through the caching patterns that keep it correct, to running it highly available. Every pattern is tried in a lab — including the ways it breaks.

DAY 1 · FOUNDATIONS

## Redis Essentials

6 hours

LEVEL Beginner – intermediate

PREREQ. Backend development experience

- Strings, hashes, lists, sets, sorted sets
- Streams, bitmaps and HyperLogLog
- Keys, expiry and key-naming conventions
- Single-threaded model and pipelining
- Persistence: RDB snapshots and AOF
- Memory model and encodings
- Redis, Valkey and managed services compared

DAY 2 · PATTERNS

## Caching Patterns

6 hours

LEVEL Intermediate

PREREQ. Day 1 or equivalent

- Cache-aside, read-through, write-through, write-behind
- TTL design, jitter and soft expiry
- Invalidation: explicit, versioned keys, events
- Stampedes: locks, early refresh, request coalescing
- Eviction policies: LRU, LFU, volatile vs. all-keys
- Client-side caching and tracking
- HTTP and CDN caching layers

DAY 3 · PRODUCTION

## Running Redis

6 hours

LEVEL Intermediate – advanced

PREREQ. Days 1–2

- Replication and Sentinel failover
- Redis Cluster: hash slots, resharding, hot spots
- Monitoring: hit ratio, latency, evictions, fragmentation
- ACLs, TLS and safe commands
- Redis on Kubernetes: operators and storage
- Benchmarking and capacity planning
- Failover drill: break it, then recover it

### WHO IT IS FOR

- Backend developers adding or fixing a cache
- Architects designing read-heavy systems
- SRE and platform teams running Redis or Valkey
- Tech leads chasing latency or database load

### WHAT YOUR TEAM LEAVES WITH

- The right caching pattern for each read path
- TTLs and invalidation that never serve stale data for long
- A database protected from stampedes and hot keys
- Redis run highly available, with the right things monitored

**DELIVERY**
Remote or on-site; daytime, evening or weekend sessions

**LABS**
A cloud lab for every participant, on realistic data

**CLASS SIZE**
Up to 15 people, so everyone gets hands-on time

**LANGUAGE**
English or Turkish, with all materials provided

**TAILORING**
Content, length and examples fitted to your stack

NEXT STEP

## Start with a 45-minute scoping call.

Tell us what is slow, wrong or about to grow. We reply within one working day with a suggested next step — consulting, training or both.

info@scaleyazilim.com

scaleyazilim.com/services/redis

Ankara · Türkiye — remote or on-site, worldwide