

Compute

COMPUTE OVERVIEW · AMAZON EC2 · COST OPTIMIZATION · CONTAINERS · AWS LAMBDA · ELASTIC BEANSTALK · STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

- 01 Four compute categories: virtual machines / IaaS** (Amazon EC2), **serverless** (AWS Lambda), **container-based** (ECS, EKS, Fargate, ECR), and **PaaS** (Elastic Beanstalk). The optimal service depends on your use case — app design, usage patterns, and how much configuration you want to manage.
- 02 EC2 launch = nine decisions: AMI → instance type → network settings → IAM role → user data → storage → tags → security group → key pair.** An AMI is the template (OS + software); the instance type fixes CPU, RAM, storage, and network performance.
- 03 Storage rule: Amazon EBS** volumes are durable — data survives stop/start. **Instance store** is ephemeral — data is lost on stop or terminate (it survives a reboot). Only EBS-backed instances can be **stopped** or **hibernated**.
- 04 Pricing models: On-Demand** (no commitment), **Reserved** (1- or 3-year, discounted), **Scheduled Reserved** (recurring schedule, 1-year), **Spot** (spare capacity, 2-minute interruption notice), **Dedicated Hosts** (whole physical server — BYO licenses), **Dedicated Instances** (single-customer hardware).
- 05 Lambda & Beanstalk facts:** Lambda runs code only when triggered — max **10,240 MB** memory and **15-minute** timeout (hard limits), billed in 100-ms increments. Elastic Beanstalk deploys uploaded code on 8 platforms at **no additional charge** — you pay only for the underlying resources.

02 KEY TERMS

TERM	DEFINITION
AMI	Amazon Machine Image — template for launching instances: root-volume template (OS + software), launch permissions, block device mapping
Instance type	Combination of CPU, memory, storage, network — named family.generation.size, e.g. t3.large
Security group	Virtual firewall on the instance: rules allow traffic by protocol, port, and source
Key pair	Public key (AWS stores) + private key file (you store) for secure connection — SSH on Linux, password decryption on Windows
User data	Script that runs with root privileges the first time an instance starts — automates setup
Amazon EBS	Durable block-level storage volumes; data persists independently of the instance
Instance store	Ephemeral block storage on disks physically attached to the host; erased on stop or terminate
Elastic IP address	Persistent public IPv4 address allocated to your account until released; re-associable within a Region
Instance metadata	Data about the running instance at http://169.254.169.254/latest/meta-data/ (only reachable from the instance)
Container	OS-virtualization unit packaging code, configuration, and dependencies; shares the host kernel — starts in hundreds of milliseconds
Lambda function	The AWS resource containing your uploaded code; runs when triggered by an event source or schedule
Event source	AWS service or custom application that produces events that trigger a Lambda function

03 CHOOSING A COMPUTE SERVICE

the classic exam question: match the use case

SERVICE	CATEGORY	USE IT WHEN
Amazon EC2	VMs (IaaS)	You want full control over the OS and configuration — web, app, database, game, mail, media, file, proxy servers
AWS Lambda	Serverless	Event-driven code with zero server management; pay only while code runs; jobs finish within 15 minutes
Amazon ECS	Containers	Orchestrate Docker containers on a managed cluster; integrates ELB, security groups, EBS, IAM roles
Amazon EKS	Containers	Run certified-conformant Kubernetes on AWS with community tools; same tooling on premises and in the cloud
AWS Fargate	Containers	Run ECS containers with no cluster to manage — package the app, set CPU/memory, launch
Amazon ECR	Containers	Store, manage, and deploy Docker images; works with Docker CLI; integrated with ECS and EKS
Elastic Beanstalk	PaaS	Upload code and let AWS handle provisioning, deployment, load balancing, scaling, health monitoring, logging

04 EC2 PRICING MODELS

per-second billing: On-Demand, Reserved & Spot on Amazon Linux / Ubuntu only

MODEL	HOW IT WORKS	BEST FOR
On-Demand	Pay by the hour, no long-term commitment; Free Tier eligible; lowest upfront cost, most flexibility	Spiky, short-term, or unpredictable workloads; dev & test
Reserved	1- or 3-year term; full, partial, or no upfront; discounted hourly rate fixed for the term	Steady-state, predictable usage over months or years
Scheduled Reserved	Capacity reservation recurring daily/weekly/monthly; 1-year term; pay for scheduled time even if unused	Predictable recurring workloads (e.g. nightly batch)
Spot	Bid on unused capacity; runs while bid $\geq$ Spot price; 2-minute interruption notice (terminate / stop / hibernate)	Fault-tolerant, flexible-timing apps; big data; large discounts vs On-Demand
Dedicated Hosts	Physical server fully dedicated to you	Per-socket / per-core / per-VM licenses (Windows, SQL Server); compliance
Dedicated Instances	Instances in a VPC on hardware dedicated to one customer — physically isolated at host level	Regulatory isolation without managing a whole host

05 LAUNCH INSTANCE WIZARD — THE NINE DECISIONS

DECISION	WHAT YOU'RE CHOOSING
1 • AMI	Template: OS + pre-installed software. Quick Start (AWS), My AMIs, AWS Marketplace, Community (unchecked — use at your own risk)
2 • Instance type	CPU, RAM, storage, network performance; categories: general purpose, compute / memory / storage optimized, accelerated computing
3 • Network	VPC + subnet; whether to auto-assign a public IP (default VPC assigns one by default)
4 • IAM role	Optional — grants apps on the instance permission to call AWS APIs (held in an <i>instance profile</i>); never store credentials on an instance
5 • User data	Optional script that runs at first start — patch, install software, configure; fewer custom AMIs to maintain
6 • Storage	Root volume + extra volumes: size, type (SSD/HDD), delete-on-termination, encryption
7 • Tags	Key + optional value (case-sensitive) — filtering, automation, cost allocation, access control

8 • Security group	Firewall rules: protocol (TCP/UDP/ICMP), port, source (IP, range, another security group)
9 • Key pair	Public key AWS keeps, private .pem you keep — the only chance to download it is at creation

06 FOUR PILLARS OF COST OPTIMIZATION

01 **Right size** — match instance to need via CloudWatch metrics, then reserve → 02 **Increase elasticity** — stop idle non-prod instances; automatic scaling → 03 **Optimal pricing model** — mix On-Demand, Spot, Reserved; consider Lambda → 04 **Optimize storage** — resize/re-type EBS, delete old snapshots, S3 lifecycle

07 AWS LAMBDA FACTS & LIMITS

soft limits: support ticket can raise • hard limits: fixed

ITEM	VALUE	TYPE
Memory allocation	128 MB – 10,240 MB	Hard
Function timeout	15 minutes max	Hard
Deployment package	250 MB unzipped (incl. layers); container image up to 10 GB	Hard
Concurrent executions	1,000 per Region	Soft
Function + layer storage	75 GB per Region	Soft
Billing	Requests + compute time in 100-ms increments	—

08 EXAM TRAPS

where the knowledge check gets you

- × “Stopping any instance preserves its data.” — Only **EBS-backed** instances can be stopped at all; an instance-store-backed instance can only be **terminated**, and instance-store data is erased on stop or terminate (it survives only a reboot).
- × “The public IP stays with the instance.” — Stop + start assigns a **new public IPv4 and external DNS name** (the private IP persists). A reboot changes nothing. Persistent public address = **Elastic IP**.
- × “Spot Instances vanish without warning.” — Spot gives a **2-minute interruption notification**; the default action is terminate, but you can configure stop or hibernate.
- × “Elastic Beanstalk costs extra and hides your resources.” — There is **no additional charge**, you pay only for the underlying resources, and you retain **full control** over them.
- × “Lambda can run as long as it needs.” — Hard cap of **15 minutes** per execution and **10,240 MB** memory. Long or stateful workflows belong on EC2, containers, or Step Functions orchestration.
- × “Containers are just lightweight VMs.” — VMs each run a **full OS on a hypervisor**; containers are resource-isolated **processes sharing one OS kernel**, which is why they start in hundreds of milliseconds.
- × “Put access keys on the instance so the app can call AWS.” — Never. Attach an **IAM role** (via instance profile) — at launch or to a running instance; changes propagate to every instance with the role.
- × “User data runs on every boot.” — By default it runs **only the first time** the instance starts (root privileges); running at every boot requires an uncommon MIME multipart setup.

- Q1** Name the four broad categories of AWS compute services.
- Q2** In the instance type name t3.large, what do the three parts mean?
- Q3** Which EC2 storage type survives a stop/start cycle, and which is erased?
- Q4** How much warning does AWS give before interrupting a Spot Instance?
- Q5** Which pricing model lets you use existing per-socket or per-core software licenses?
- Q6** List the four pillars of EC2 cost optimization.
- Q7** What are the maximum memory allocation and maximum run time of a Lambda function?
- Q8** Which ECS cluster option runs containers without any EC2 instances for you to manage?
- Q9** At which URL can a running instance read its own metadata?
- Q10** Sample CCP exam: which AWS service helps developers quickly deploy resources that can use different programming languages, such as .NET and Java — CloudFormation, SQS, Elastic Beanstalk, or EC2?

ANSWER KEY (rotate the page)

Q1 virtual machines (IaaS), serverless, container-based, PaaS **Q2** t = family, 3 = generation, large = size **Q3** EBS survives; instance store is erased **Q4** a 2-minute notification **Q5** Dedicated Hosts **Q6** right size, increase elasticity, optimal pricing model, optimize storage choices **Q7** 10,240 MB and 15 minutes **Q8** AWS Fargate **Q9** <http://169.254.169.254/latest/meta-data/> **Q10** AWS Elastic Beanstalk