



Atlas: Compute and Containers

AWS Compute and Containers categories, as of 2026 | EC2 families and naming, Auto Scaling, Lightsail, Lambda, Batch, Elastic Beanstalk, App Runner, ECS, EKS, Fargate, ECR, Outposts and edge compute, HPC; when to choose each and the look-alikes

EC2 **Lambda** **ECS** **EKS** **Fargate** **CLF-C02 D3**

Virtual machines: EC2 and friends

Family	Letters	Good for
General purpose	M, T (burstable), Mac	web and app servers, small databases; T for spiky low CPU (CPU credits)
Compute optimised	C	batch, media encoding, gaming servers, HPC, high-throughput APIs
Memory optimised	R, X, z, U (high memory)	in-memory caches, large relational DBs, SAP HANA
Storage optimised	I, D, H	NoSQL, data warehousing, high local NVMe IOPS or HDD throughput
Accelerated computing	P, G (GPU), Trn (Trainium), Inf (Inferentia), F (FPGA)	ML training and inference, graphics, video
HPC optimised	Hpc	tightly coupled simulations with EFA networking

EC2 families (as of 2026)

Part	Meaning
m	family: general purpose
7	generation (higher = newer, usually cheaper per performance)
g	processor: g = Graviton (Arm), i = Intel, a = AMD
d	option: local NVMe instance store (n = network, e = extra memory/storage, z = high frequency)
2xlarge	size: 8 vCPU, 32 GiB for m-family (each size doubles: large 2 vCPU, xlarge 4, 2xlarge 8)

Read m7gd.2xlarge

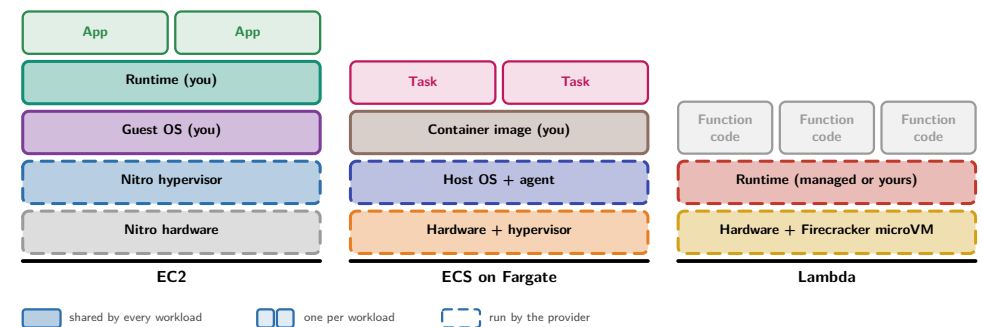
- **Amazon Lightsail** VPS bundles (instance + SSD + transfer + static IP) at a fixed monthly price; for simple sites and beginners, peers with a VPC
- **EC2 Auto Scaling** keeps a group healthy and sized; target tracking (keep CPU at 50 %), step, scheduled, predictive scaling
- **Elastic Load Balancing** spreads traffic over instances (sheet 03)
- **EC2 Image Builder** pipelines that build, test and distribute patched AMIs and container images
- **AWS Outposts, Local Zones, Wavelength** EC2 on premises, in metros, in 5G networks (welcome chapter)

- **AWS ParallelCluster, AWS PCS** HPC clusters with Slurm; Elastic Fabric Adapter (EFA) for low-latency MPI
- **VMware Cloud on AWS** VMware SDDC on dedicated hosts; relocate vSphere VMs without changes (sold through Broadcom, check availability)

⚡ Serverless and platform compute

Service	What it is	Choose it when	Hard limits that bite
Lambda	functions on events, per request + GB-s	event-driven, spiky, short tasks; glue code	15 min max; 128 MB-10,240 MB memory; 10 GB /tmp; 6 MB sync payload; 1,000 concurrent per region by default
Fargate	containers without servers	long-running services, any language, no host management	per vCPU/GB-hour; no GPUs; no privileged containers
AWS Batch	managed job queues on EC2, Spot or Fargate	many batch or HPC jobs, array jobs	you pay underlying compute; job definitions per image
Elastic Beanstalk	PaaS: upload code, it builds EC2, ALB, ASG	classic web apps with little ops knowledge	less control; platform versions to update
App Runner	container or source → HTTPS service, autoscaled	simple web APIs, no VPC design	fewer knobs than ECS; check regional availability
Lightsail containers	fixed-price container service	tiny apps	small sizes only

Who manages which layer



What does a Lambda function cost per month?

Y

An API runs $R = 3$ million invocations a month, each lasting $d = 200$ ms with $m = 512$ MB of memory. Estimate the bill before the free tier (x86, us-east-1 example prices).

Given: $p_{req} = 0.20$ USD per million requests; $p_{gbs} = \text{approx. } 0.0000166667$ USD per GB-second (check the Lambda pricing page); $GB\text{-}s = R \times d(s) \times m(GB)$

Y

Step 1: How many GB-seconds?

$3,000,000 \times 0.2 \text{ s} \times 0.5 \text{ GB} = 300,000 \text{ GB-s}$
duration is billed per ms, scaled by memory

T

Y

Step 2: Compute charge?

$300,000 \times 0.0000166667 = 5.00$ USD
price per GB-s

T

Y

Step 3: Request charge?

$3 \times 0.20 = 0.60$ USD
per million requests

T

Y

Step 4: Total?

$5.00 + 0.60 = 5.60$ USD per month
the always-free tier (1 M requests, 400,000 GB-s) could cover most of it

T

About 5.60 USD a month; the same load on an always-on t3.small would cost more and need patching

Check: Arm (Graviton) Lambda is about 20 % cheaper per GB-s; doubling memory can shorten duration so cost may not double.

T

Feature check

Item	Serverless option	Kubernetes API	GPU support	Runs > 15 min	Scales to zero	Total
Lambda	☒	☐	☐	☐	☒	2
ECS	☒	☐	☒	☒	☐	3
EKS	☒	☒	☒	☒	☐	4
App Runner	☒	☐	☐	☒	☐	2
AWS Batch	☒	☐	☒	☒	☒	4
EC2	☐	☐	☒	☒	☐	2
Total	5	1	4	5	2	

- **Amazon ECR** private registry per region; image scanning (basic or Inspector enhanced), cross-region replication, lifecycle rules to expire old tags; pull-through cache for Docker Hub, Quay, GitHub
- **EKS extras** EKS Auto Mode (AWS manages nodes, networking, storage add-ons), Pod Identity for IAM, EKS Anywhere and Hybrid Nodes for on-premises
- **ECS extras** Service Connect for service-to-service traffic, capacity providers mixing Fargate and Fargate Spot, task IAM roles
- **Depth** containers-and-eks chapter; Lambda depth: serverless chapter

Containers

Which container service?

Conditions	K8s	Hosts	Default	Simplest
Need Kubernetes APIs, Helm, CNCF tools	Y	N	N	N
Want to manage hosts / GPUs / daemons	--	Y	N	N
Simple HTTP service, minimal config	--	N	N	Y
Actions				
EKS (managed nodes, Karpenter or Auto Mode)	X			
ECS on EC2		X		
ECS on Fargate			X	
App Runner				X

Look-alikes, recap and self-test

ECS AWS-native orchestrator, simpler Tell: Does the team need Kubernetes APIs and tooling?	vs	EKS managed Kubernetes, portable
Fargate containers, any duration Tell: Long-running or > 15 min, or custom container runtime?	vs	Lambda functions, 15 min max, per request
Elastic Beanstalk PaaS on EC2 you can inspect Tell: Do you want EC2 instances you can tune?	vs	App Runner container service, no instances visible
Lightsail fixed monthly bundles Tell: Simple and predictable price vs full control?	vs	EC2 pay per second, all options
AWS Batch queues of jobs Tell: Many independent compute jobs vs a workflow of calls?	vs	Step Functions orchestrates steps and services

Statement	True	False
1. A Lambda function can run for 30 minutes → 15 minutes maximum; use Fargate, Batch or Step Functions	☐	☒
2. The g in c7g means Graviton → Arm processor designed by AWS	☒	☐
3. Fargate works with both ECS and EKS → serverless compute for tasks and pods	☒	☐
4. T instances give unlimited sustained full CPU at base price → they burst on CPU credits; unlimited mode bills surplus credits	☐	☒
5. ECR scanning can use Amazon Inspector → enhanced scanning, continuous	☒	☐

EC2 = full control
family by bottleneck: CPU, memory, storage, GPU

Lambda = events, 15 min
per request + GB-s

Fargate = containers, no hosts
ECS or EKS

ECS simple, EKS Kubernetes
both use ECR

Graviton for price-performance
check Arm compatibility

Q1 Which EC2 family for an in-memory cache of 500 GiB?

A Memory optimised (R or X family).

Q2 Run a nightly 2-hour container job without managing servers?

A ECS task on Fargate (scheduled with EventBridge Scheduler) or AWS Batch on Fargate.

Q3 Maximum Lambda memory and timeout?

A 10,240 MB and 15 minutes.

Q4 What does the 'd' in m7gd mean?

A Local NVMe instance store disks.

Q5 Simplest way to deploy a web container with HTTPS and autoscaling, no VPC design?

A AWS App Runner (check regional availability); otherwise ECS on Fargate behind an ALB.

Q6 Which service builds and patches AMIs on a schedule?

A EC2 Image Builder.

Q7 Kubernetes workers fully managed by AWS, including scaling and patching?

A EKS Auto Mode (or Fargate profiles for pods).

Q8 Fixed-price VPS for a WordPress blog?

A Amazon Lightsail.

Q9 Cost of 1 million Lambda invocations of 100 ms at 1 GB (x86, example price)?

A $100,000 \text{ GB-s} \times 0.0000166667 \approx 1.67 \text{ USD} + 0.20 \text{ USD requests} \approx 1.87 \text{ USD}$ before the free tier.