

Cloud Economics and Billing

AWS pricing fundamentals, total cost of ownership, AWS Organizations, billing tools, and support plans

Cloud economics is the business argument for everything else in this course. AWS prices like a utility — you pay for compute, storage, and outbound data transfer as you consume them, you pay less when you commit or consume at volume, and prices fall as AWS's economies of scale grow. This module builds that pricing model from the ground up, then shows how to *prove* the economics with a Total Cost of Ownership comparison (including a worked 3-year example and a real customer case study), how AWS Organizations turns many accounts into one manageable, one-bill entity, which tools in AWS Billing and Cost Management track and forecast spend, and which of the four support plans a workload actually needs.

LEARNING OBJECTIVES

- Explain the AWS pricing philosophy
- Recognize fundamental pricing characteristics
- Indicate the elements of total cost of ownership
- Discuss the results of the AWS Pricing Calculator
- Identify how to set up an organizational structure that simplifies billing and account visibility to review cost data
- Identify the functionality in the AWS Billing Dashboard
- Describe how to use AWS Bills, AWS Cost Explorer, AWS Budgets, and AWS Cost and Usage Reports
- Identify the various AWS technical support plans and features

2.1 Fundamentals of pricing

Three characteristics drive almost every AWS bill: **compute**, **storage**, and **outbound data transfer**. Details vary by product and pricing model, but the pattern holds: compute is charged per hour or per second (per-second applies to Linux) at a rate that varies by instance type; storage is typically charged per GB; and data transfer is typically charged per GB — but only *outbound*. Outbound transfer is **aggregated across services** and appears on the monthly statement as **AWS Data Transfer Out**. In most cases there is **no charge for inbound data transfer or for transfer between AWS services within the same Region** — with some exceptions, so verify rates before adopting a service.

TABLE 2.1 The three fundamental cost drivers

DRIVER	HOW IT'S CHARGED	DETAIL WORTH KEEPING
Compute	Per hour or per second (per-second: Linux only)	Rate varies by instance type
Storage	Typically per GB	Tiered for S3 / EBS / EFS — more use, lower unit price
Data transfer out	Per GB, aggregated across services	Billed as “AWS Data Transfer Out”; rates are tiered
Data transfer in	No charge (some exceptions)	Same for service-to-service transfer within one Region

The pricing *philosophy* has four tenets. **Pay for what you use** — at the end of each month you pay only for what you consumed; start or stop anytime, with **no large upfront expenses, long-term contracts, or complex licensing**. **Pay less when you reserve** — for services like Amazon EC2 and Amazon RDS, reserved capacity saves **up to 75 percent** over equivalent on-demand while helping organizations minimize risk, budget predictably, and satisfy commitment policies. **Pay less by using more** — tiered, volume-based pricing drops the per-GB price of S3, EBS, or EFS as usage grows, and the storage portfolio lets you match cost to access frequency and retrieval performance. **Pay even less as AWS grows** — economies of scale come back as lower prices: **75 price reductions between 2006 and September 2019**, with future higher-performing resources replacing current ones at no extra charge. If none of these fit, **custom pricing** serves high-volume projects with unique requirements.

TABLE 2.2 Reserved Instance payment options

OPTION	UPFRONT PAYMENT	DISCOUNT
AURI — All Upfront Reserved Instance	Everything upfront	Largest discount
PURI — Partial Upfront Reserved Instance	Part upfront	Lower discount, less capital tied up
NURI — No Upfront Payments Reserved Instance	Nothing upfront	Smallest discount; frees capital

FREE TIER AND NO-CHARGE SERVICES

- **AWS Free Tier:** free hands-on experience for new customers for **1 year** — e.g. an EC2 T2 micro instance, plus free usage tiers for S3, EBS, Elastic Load Balancing, and data transfer.
- Services with **no charge:** Amazon VPC · IAM · Consolidated Billing · Elastic Beanstalk · CloudFormation · automatic scaling · OpsWorks.
- To estimate any cost: identify each service's fundamental characteristics, estimate usage of each, and map that usage to the posted prices.

COMMON PITFALL

A free *service* is not a free *bill*. Elastic Beanstalk, CloudFormation, and automatic scaling cost nothing themselves, but the resources they provision are billed normally — when scaling adds EC2 instances, you pay for those instances.

2.2 Total Cost of Ownership

On-premises and cloud infrastructures differ in *how they are deployed*, and therefore in how they cost. An **on-premises** infrastructure runs on a company's own servers and carries **fixed capital expenses** — facilities, hardware, licenses, maintenance staff. Scaling up is expensive and slow; scaling down **does not reduce the fixed costs**. A **cloud** infrastructure is purchased from a provider who builds and maintains the facilities, hardware, and staff; the customer pays for what is used, scaling either direction is simple, and costs track service use. The two are hard to compare directly — capex and long planning cycles versus flexibility, agility, and consumption-based cost.

Total Cost of Ownership (TCO) is the bridge: a financial estimate of the **direct and indirect costs** of a product or system — the cost of the service *plus* everything associated with owning it. Use TCO to compare an environment or specific workload on-premises (or in colocation) against the same workload on AWS, for **budgeting** and to **build the business case** for moving to the cloud. Cloud costs are transparent — published per usage metric (RAM, storage, bandwidth), fixed per unit of time. In-house costs are harder: an honest tally includes the **direct** costs of running a server (power, floor space, storage, IT operations) and the **indirect** costs (network and storage infrastructure).

TABLE 2.3 TCO cost categories for a data center

CATEGORY	WHAT IT INCLUDES
Server costs	Hardware and software, plus facilities to house the equipment
Storage costs	Storage hardware, administration, and facilities
Network costs	Network hardware, administration, and facilities
IT labor costs	Administering the whole solution — including security and application administration

WORKED EXAMPLE: 3-YEAR ON-PREMISES VS. ALL-IN-CLOUD

Two similar environments: on-premises, **1 VM with 4 CPUs, 16 GB RAM, Linux**, at 100% average utilization (optimized by RAM); on AWS, **1 m4.xlarge (4 CPUs, 16 GB RAM)** as a **3-year Partial Upfront Reserved Instance**, including business-level support. Three-year totals: on-premises **\$167,422** vs. AWS **\$7,509** — **96% savings**, or **\$159,913** (and the on-prem figure *excludes* additional direct and indirect costs). Why the gap? On-premises keeps incurring cost whether or not capacity is used; the AWS solution is commissioned when needed and decommissioned when idle.

To build estimates like this, use the **AWS Pricing Calculator**: it estimates monthly costs, identifies opportunities to reduce them, **models solutions before you build them**, exposes the price points and calculations behind an estimate, and finds the instance types and contract terms that fit. You can name an estimate and organize services into named **groups** — containers arranged by cost center, department, or product architecture. (The class activity has teams price a provided specification with the calculator and report back.)

READING A PRICING CALCULATOR ESTIMATE

Every estimate has three numbers: **first 12 months total** (upfront + monthly combined for the current group and everything in it), **total upfront** (paid while setting up the stack), and **total monthly** (run cost). Deck sample: first 12 months **\$886.92** = total upfront **\$0** + total monthly **\$73.91** × 12. Each service's cost is visible within its group — to price alternative designs, build one group per variation and compare.

A TCO delta alone is a narrow view of the financial impact. A **return on investment (ROI)** analysis starts with **hard benefits** — direct, visible cost reductions — then adds **soft benefits**: value that is hard to quantify but can be worth *more* than the hard savings.

HARD BENEFITS	SOFT BENEFITS
<i>direct, visible, easy to measure</i> ▪ Reduced spend on compute, storage, networking, security ▪ Fewer hardware/software purchases (capex) ▪ Lower operational, backup, and disaster-recovery costs ▪ Reduction in operations personnel	<i>harder to quantify — often worth more</i> ▪ Reuse of services and applications to define and redefine solutions ▪ Increased developer productivity ▪ Improved customer satisfaction ▪ Agile processes; quick response to new opportunities ▪ Increased global reach

Case study — Delaware North. A 1915 peanut-and-popcorn vendor turned global food-and-hospitality leader: **200+ locations, 500 million customers a year, \$3 billion revenue.** Its on-premises data center had grown too expensive and inefficient — rising demand to deploy new solutions plus constantly aging equipment. After migrating ~50 websites to AWS in 2013, it evaluated a full move on three criteria: a **broad solution** for all enterprise workloads, flexibility to **modify core IT processes** and eliminate busy work (like patching software), and a demonstrable **positive ROI**.

DELAWARE NORTH: WHAT THE MOVE DELIVERED

- Estimated **\$3.5 million saved over a 5-year run rate** using 3-year EC2 Reserved Instances and renewals.
- **Eliminated 205 servers — over 90%** of its hardware; kept ~20 on-premises for communications and file-and-print.
- New-business provisioning fell from **2–3 weeks to 1 day**; DevOps pushes out a service in **minutes**.
- Bonuses: cost-effective security compliance, enhanced disaster recovery with seamless failovers, continuous cost optimization.

2.3 AWS Organizations

AWS Organizations is a **free** account-management service that consolidates multiple AWS accounts into one organization you create and centrally manage, meeting budgetary, security, and compliance needs. Main benefits: **centrally managed access policies** across accounts, **controlled access to AWS services**, **automated account creation and management** via APIs, and **consolidated billing** — one payment method, a combined view of charges, and pricing benefits from aggregated usage.

The structure is an upside-down tree: the **root** at top, **organizational units (OUs)** branching beneath, **accounts** as the leaves. An OU is a container for accounts and can contain other OUs; an OU has exactly **one parent**, and each account belongs to exactly **one OU**. Attach a policy to any node and it **flows down** to every branch and leaf beneath; attach it to a single account to control only that account.

Organizations does not replace IAM. IAM policies allow or deny access to AWS services (like S3), individual resources (a specific bucket), or individual API actions (`s3:CreateBucket`) — but apply only to users, groups, and roles, and can **never restrict the account root user**. **Service control policies (SCPs)** allow or deny AWS services for whole accounts or OUs — and bind **every** identity in the account, **including the root user**.

IAM POLICIES	SERVICE CONTROL POLICIES
<i>inside one account</i>	<i>across the organization</i>
▪ Attach to users, groups, and roles ▪ Allow/deny services, individual resources, individual API actions ▪ Can never restrict the account root user	▪ Attach to accounts or OUs ▪ Allow/deny access to AWS services for those accounts ▪ Affect all users, groups, roles — including member-account root

SETTING UP AN ORGANIZATION, IN ORDER

- **1 — Create the organization** with your current account as primary; invite one account, create another as a member.
- **2 — Create OUs** and place the member accounts in them.
- **3 — Create SCPs** restricting what can be delegated to users and roles in member accounts.
- **4 — Test restrictions:** sign in per role and observe, or use the IAM policy simulator.

TABLE 2.4 AWS Organizations limits

ENTITY	LIMIT
Names (accounts, OUs, roots, policies)	Unicode, ≤250 characters
Roots · OUs · policies	1 · 1,000 · 1,000
Maximum SCP document size	5,120 bytes
OU nesting depth	5 levels under a root
Invitations/day · concurrent account creations	20 · 5
Entities a policy can attach to	Unlimited

Manage an organization through four interfaces: the **AWS Management Console** (browser; every task), the **AWS CLI** (often faster), **SDKs** in many languages (they handle request signing, errors, retries), and the **HTTPS Query API** — direct programmatic requests, which you must digitally sign with your credentials.

COMMON PITFALL

Watch the direction of control: an IAM policy inside a member account cannot touch that account's root user — but an SCP applied from the organization restricts *everyone* in the member account, root included.

2.4 AWS Billing and Cost Management

AWS Billing and Cost Management is the service you use to **pay your bill, monitor usage, and budget costs** — and to forecast future costs and usage so you can plan ahead. You can set a custom time period, view data at **monthly or daily** granularity, and filter and group across many dimensions. The **Billing Dashboard** (shown in the module's recorded demo) gives the at-a-glance view: month-to-date spend, the services driving most of it, and the cost trend. Two graphs matter: **Spend Summary** — last month's spend, month-to-date cost, and a forecast for the current month — and **Month-to-Date Spend by Service** — top services and each one's share of cost. From the dashboard you reach four deeper tools.

TABLE 2.5 The four cost-management tools

TOOL	WHAT IT DOES	KEY FACTS
AWS Bills	Lists the past month's costs per service	Breakdown by Region and linked account; the most up-to-date view of costs and usage
Cost Explorer	Visualize and manage costs and usage over time as graphs	Free · past 13 months of data · forecasts the next 3 months with a confidence interval · default report of top cost-incurring services · surfaces patterns, problem areas, busiest AZs and linked accounts
AWS Budgets	Notify when actual or forecasted costs exceed budget	Built on Cost Explorer's visualization · monthly, quarterly, or yearly with custom start/end · alerts via email or Amazon SNS
Cost and Usage Report	Single location for the most comprehensive cost/usage data	Line items hourly or daily per service category (plus activated taxes) · published to an S3 bucket , updated once a day

COMMON PITFALL

Keep the tools straight: the **Pricing Calculator** models a solution *before* you build; **Cost Explorer** graphs what you *actually* spent; **Budgets** raises alerts against thresholds; the **Cost and Usage Report** is the raw, comprehensive record.

2.5 Technical support

AWS Support pairs tools with expertise, tiered by how you use AWS: **experimenting**, running **production** workloads, or depending on it as a **business-critical** resource. Three named resources recur on the exam. A **Technical Account Manager (TAM)** provides *proactive guidance* — a designated primary point of contact for architectural review and ongoing communication as you plan, deploy, and optimize. **AWS Trusted Advisor** covers *best practices* — an online, customized-cloud-expert tool that checks for opportunities to reduce monthly spend and increase performance and fault tolerance. The **Support Concierge** handles *account assistance* — a billing and account expert for all **non-technical** billing and account-level questions.

TABLE 2.6 The four AWS Support plans

PLAN	DESIGNED FOR	WHAT YOU GET
Basic	Every customer, automatically	24/7 customer service, documentation, whitepapers, forums · six core Trusted Advisor checks · Personal Health Dashboard · no technical case support
Developer	Testing / early development	Guidance and technical support for customers exploring AWS or running non-production workloads
Business	Production workloads	For applications in production, multiple services activated or key services used extensively; solutions must stay available, scalable, secure
Enterprise	Business/mission-critical workloads	Proactive management · best-practice operation · launch and migration support · a TAM with detailed knowledge of your use case and architecture

Cost is only half of choosing a plan — the other half is **service levels**: response times depend on the plan *and* the **case severity**. The **Basic plan includes no case support at all**. (The scavenger-hunt activity has teams defend a support-plan recommendation for a business case — good practice for exam scenarios.)

TABLE 2.7 Case severity levels

SEVERITY	MEANING
Critical	Business at risk; critical functions of the application unavailable
Urgent	Business significantly impacted; important functions unavailable
High	Important functions impaired or degraded
Normal	Non-critical functions abnormal, or a time-sensitive development question
Low	General development question or feature request

COMMON PITFALL

Don't confuse the humans with the tool: a **TAM** is a person (Enterprise plan) giving proactive guidance; **Trusted Advisor** is an automated tool (six core checks even on Basic) making cost, performance, fault-tolerance, and security *recommendations* — the sample exam question hinges on that keyword.

Module summary

- Three cost drivers: compute (per hour/second), storage (per GB), and aggregated data transfer **out** — inbound and same-Region service-to-service transfer are generally free.
- Philosophy: pay for what you use, pay less when you reserve (RIs up to 75% off; AURI > PURI > NURI), pay less by using more (tiered pricing), pay even less as AWS grows (75 price cuts 2006–2019).
- The Free Tier gives new customers a free year; VPC, IAM, Elastic Beanstalk, CloudFormation, automatic scaling, OpsWorks, and Consolidated Billing are free — the resources they provision are not.
- TCO totals direct **and** indirect costs (server, storage, network, IT labor, facilities) to compare on-premises with AWS; deck example: \$167,422 vs. \$7,509 over 3 years — 96% savings.
- The Pricing Calculator models solutions before you build; estimates split into first-12-months, upfront, and monthly, organized in groups. ROI = hard benefits (visible cost cuts) + soft benefits (productivity, agility, reach).
- Organizations (free) arranges accounts in a root → OU → account tree where policies flow down; SCPs cap whole accounts — including member root users; consolidated billing yields one bill plus volume discounts.
- Billing and Cost Management: Billing Dashboard plus AWS Bills, Cost Explorer (free; 13 months back, 3-month forecast), AWS Budgets (email/SNS alerts), and the Cost and Usage Report (hourly/daily line items to S3).
- Support plans: Basic (automatic; no cases), Developer (early development), Business (production), Enterprise (mission-critical, with a TAM); Trusted Advisor makes recommendations; severity (critical → low) drives response times.

Review questions

1. Why does an AWS architect care about the *direction* of data transfer?

Answer: Only outbound transfer is billed: it is aggregated across services and charged per GB as AWS Data Transfer Out, while inbound transfer and same-Region service-to-service transfer generally cost nothing — so designs that keep traffic inside a Region avoid the metered direction.

2. One company wants maximum Reserved Instance savings; another wants to preserve capital. Which options fit, and what's the rule?

Answer: The first buys All Upfront RIs (AURI) — the largest discount; the second buys No Upfront (NURI) — smallest discount, zero capital outlay; Partial Upfront (PURI) sits between. Rule: more upfront = greater discount, up to 75% versus on-demand.

3. Which cost categories must an honest on-premises TCO include beyond the server purchase?

Answer: Server hardware/software plus facilities; storage and network hardware with their administration and facilities; IT labor (including security and application administration); plus direct costs like power and floor space and indirect costs like network/storage infrastructure.

4. Explain how the module's 3-year comparison reached 96% savings.

Answer: An on-prem VM (4 CPUs, 16 GB RAM, Linux, 100% utilization) cost \$167,422 over 3 years because fixed costs accrue whether or not capacity is used; the equivalent m4.xlarge as a 3-year Partial Upfront RI with business-level support cost \$7,509 — a \$159,913 difference.

5. How do SCPs and IAM policies divide the work of securing an organization?

Answer: IAM policies act inside an account on users, groups, and roles — never the root user — down to individual resources and API actions. SCPs act from the organization on accounts or OUs and bind every identity in a member account, including root. Organizations complements IAM; it doesn't replace it.

6. Beyond the projected \$3.5 million savings, what did Delaware North gain from migrating?

Answer: It eliminated 205 servers (over 90% of its hardware), gained cost-effective security compliance, stronger disaster recovery with seamless failovers, cut new-business provisioning from 2–3 weeks to 1 day, and let DevOps push out services in minutes instead of weeks.

7. Pick the right tool: (a) estimate a solution before building it; (b) graph 13 months of actual spend; (c) get an email when forecasted costs exceed a limit; (d) hourly line items delivered to S3.

Answer: (a) AWS Pricing Calculator, (b) Cost Explorer (free, with a ~3-month forecast), (c) AWS Budgets (alerts by email or Amazon SNS), (d) the AWS Cost and Usage Report (updated once a day).

8. A startup runs production workloads; an enterprise wants a designated technical contact for mission-critical systems. Which support plans, and why not the cheaper tiers?

Answer: The startup needs Business Support — built for production workloads that must stay available, scalable, and secure. The enterprise needs Enterprise Support — the only plan with a Technical Account Manager. Developer targets non-production work, and Basic includes no technical case support at all.
