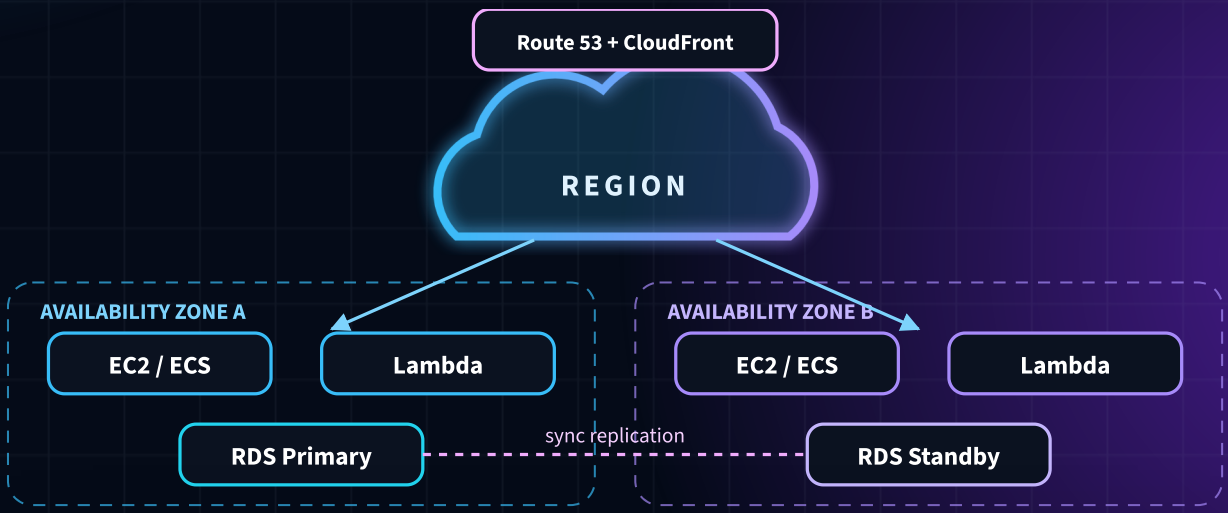
JOB-READY SERIES · JOB-SWITCH EDITION

AWS · AZURE · GCP

Cloud

The Job-Switch Guide · AWS-First, Multi-Cloud

Every cloud service interviews ask about, in simple language: IAM to VPCs, EC2 to Lambda, S3 to DynamoDB, security, cost and architecture, with free-tier labs, real scenarios, projects and 120+ questions with answers.




```
shreyash@cloudshell: ~

$ aws sts get-caller-identity --query Arn  "...:role/CloudEngineer"
$ aws cloudformation deploy --stack-name career  Successfully created
$ aws ce get-cost-forecast  {"Offer": "Accepted", "Bill": "-40%"}
```

IAM · VPC · EC2 · S3 · RDS · DynamoDB · Lambda · ECS / EKS · SQS / SNS · CloudFront · KMS · CloudWatch · CloudFormation · Azure · GCP

WRITTEN BY

Shreyash Tripathi

17 Chapters · Free-Tier Labs · Architecture · Interview Questions

Edition 2026

JOB-READY SERIES · JOB-SWITCH EDITION



Cloud Computing

The Job-Switch Guide, AWS-first and multi-cloud ready

Learn the service | Run the free-tier lab | Design the architecture | Build the project | Ace the interview

WRITTEN BY

Shreyash Tripathi

Job-Switch Edition, 2026

© Shreyash Tripathi. All rights reserved.

Preface

Almost every company now runs on the cloud, and almost every backend, DevOps and platform job description asks for it. AWS alone has more than two hundred services, but interviews and real jobs keep coming back to the same few dozen, and they fit together like building blocks. Identity decides who can do what, the network decides what can talk to what, compute runs the code, storage and databases keep the data, and monitoring and cost control keep it all healthy.

This book teaches those building blocks the way a good school textbook would: numbered chapters, simple language, a check with answers after every topic, and exercises with a full answer key. Because the cloud is learned by doing, every chapter has free-tier hands-on labs with clean-up steps so you never get a surprise bill, Interview Corners, At Work stories, troubleshooting sections, a resume line, and a table mapping each AWS service to Azure and GCP.

Do the labs, build the four portfolio projects in Chapter 16, and you will walk into interviews with real stories instead of memorised definitions.

Shreyash Tripathi

Features of Every Chapter

🎯 **INTERVIEW CORNER**

Questions interviewers really ask, with model answers.

💼 **AT WORK**

What really happens on the job and on call.

🧪 **HANDS-ON LAB**

Do it yourself on the AWS free tier, with clean-up steps.

⚠️ **COMMON MISTAKE**

Traps that break production and fail interviews.

💡 **DO YOU KNOW?**

Real facts from the industry.

😄 **FUNNY SIDE**

Because surprise cloud bills need jokes.

✅ **CHECK YOUR PROGRESS**

Short questions with answers after every topic, plus exercises, scenario questions, a mini project and a full answer key at the end of every chapter.

Contents

PART I: CLOUD FOUNDATIONS

1 Cloud Computing and the Cloud Career Roadmap 6

2 Cloud Foundations: Regions, Pricing and the Well-Architected Framework 24

3 Identity and Access Management 43

4 Cloud Networking: VPC in Depth 65

PART II: CORE SERVICES

5 Compute: EC2, Auto Scaling and Load Balancing 85

6 Storage: S3, EBS, EFS and Backup 103

7 Databases in the Cloud 120

8 Containers and Serverless 137

PART III: CONNECT, SECURE AND OPERATE

9 Messaging, Events and Integration 156

10 DNS, CDN and the Edge 176

11 Cloud Security and Compliance 195

12 Monitoring, Logging and Cost Optimization 213

PART IV: ARCHITECT LIKE A PRO

13 Infrastructure as Code and Automation 232

14 High Availability, Disaster Recovery and Architecture Patterns 254

15 Azure and GCP for AWS Engineers 273

PART V: GET THE JOB

16 Cloud Projects and Real Scenarios 292

17 Cloud Interview Question Bank 309

APPENDICES

A AWS CLI and Service Cheat Sheet 327

B Architecture and Policy Templates 333

C Glossary 339

D Job Switch Toolkit 343

PART I

Cloud Foundations

What the cloud is, how it is priced, who can do what, and how networks are built.

CHAPTER 1

Cloud Computing and the Cloud Career Roadmap

"The cloud is just someone else's computer. The skill is knowing how to rent it without getting a shock on the bill."

After studying this chapter, you will be able to

- Define cloud computing using the five NIST essential characteristics.
- Tell IaaS, PaaS, SaaS and FaaS apart with real service examples.
- Compare public, private, hybrid and multi-cloud deployment models.
- Compare AWS, Azure and GCP by market position and strengths.
- Pick a target cloud role and know what its interviews test.
- Follow a 90-day plan with certifications, portfolio projects and a cloud resume.

Think about how your family gets electricity. Nobody in your colony runs a coal plant on the terrace. You plug into the grid, use what you need, and pay the bill at the end of the month based on the meter reading. If you buy an AC in May, the grid does not ask you to build a new power station first. Cloud computing is the same idea applied to servers, storage, databases and networks. Instead of buying machines, waiting six weeks for delivery and hiring someone to keep them cool, you rent them by the second from a provider who already has millions of them.

This chapter is your map. We first learn what "cloud" really means (interviewers love the NIST definition), then the service and deployment models, then the big three providers. The second half is about you: which cloud role suits you, what each interview checks, which certifications to take in which order, and a concrete 90-day plan to switch jobs.

1.1 What Is Cloud Computing?

Cloud computing is on-demand access to computing resources (servers, storage, databases, networking, software) over the internet, with pay-as-you-go pricing. You do not own the hardware. You call an API or click in a console, and a few seconds later a virtual server is running in a data centre you will never visit.

1.1.1 The NIST definition: five essential characteristics

The US National Institute of Standards and Technology (NIST) published a definition in Special Publication 800-145. It lists five essential characteristics. If a service does not have these, it is just

"hosting", not cloud.

Characteristic	Meaning in plain words	AWS example
On-demand self-service	You create resources yourself, no ticket to a human.	`aws ec2 run-instances` starts a server in under a minute.
Broad network access	Reachable over the network from laptops, phones, scripts.	Console, CLI, SDKs, all over HTTPS.
Resource pooling	Provider shares a big pool of hardware among many customers (multi-tenancy).	Your EC2 instance shares a physical host with others (unless you pay for Dedicated Hosts).
Rapid elasticity	Scale up and down quickly, often automatically.	Auto Scaling group adds 10 instances during a sale.
Measured service	Usage is metered and you pay for what you use.	EC2 billed per second (Linux), S3 per GB-month.

> 🔑 **KEY POINT**
>
> Memorise the five NIST characteristics as "SNP-EM": Self-service, Network access, Pooling, Elasticity, Measured. It is a common first question in Cloud Practitioner and support interviews.

1.1.2 Why companies move to cloud

- **CapEx to OpEx:** no big upfront hardware purchase; you pay monthly operating expense.
- **Speed:** new environment in minutes, not weeks.
- **Global reach:** deploy near users in Mumbai, Frankfurt or Virginia with the same template.
- **Elasticity:** handle a Diwali sale spike without buying servers that sit idle the rest of the year.
- **Managed services:** let the provider patch databases and run queues so your team builds features.

Cloud is not automatically cheaper. A steady, always-on workload that is badly sized can cost more than owned hardware. That is why FinOps exists as a job, which we meet in Section 1.5.

> 💡 **DO YOU KNOW?**
>
> AWS launched its first broadly available service, Amazon S3, in March 2006, followed by EC2 later that year. Microsoft Azure became generally available in 2010, and Google Compute Engine in 2013. AWS's head start is one reason it still leads the market.

CHECK YOUR PROGRESS 1.1

1. **Name the five NIST essential characteristics of cloud computing.**
   Answer: On-demand self-service, broad network access, resource pooling, rapid elasticity and measured service.
2. **A company rents a fixed rack in a data centre on a 3-year contract and raises a ticket to get a new server. Is this cloud? Why?**
   Answer: No. There is no on-demand self-service, no rapid elasticity and no metered pay-per-use. It is traditional colocation or hosting.
3. **What does "CapEx to OpEx" mean?**
   Answer: Moving from large upfront capital purchases (buying servers) to ongoing operating expense (paying monthly for usage).

1.2 Service Models: IaaS, PaaS, SaaS and FaaS

Service models answer one question: **how much does the provider manage, and how much do you manage?** The classic analogy is food.

- **On-premises** = cooking at home. You buy the gas, vegetables, utensils and you wash up.
- **IaaS** = renting a kitchen. Stove and gas are provided; you bring ingredients and cook.
- **PaaS** = a tiffin service where you choose the menu; they cook, you just eat and give feedback.
- **SaaS** = eating at a restaurant. You only order.
- **FaaS** = ordering one samosa only when you are hungry, and paying for exactly that samosa.

```
Layer            On-prem   IaaS     PaaS     SaaS     FaaS
---------------  -------   ------   ------   ------   ------
Application      YOU       YOU      YOU      vendor   YOU(code)
Data             YOU       YOU      YOU      YOU*     YOU
Runtime          YOU       YOU      vendor   vendor   vendor
OS / patching    YOU       YOU      vendor   vendor   vendor
Virtualisation   YOU       vendor   vendor   vendor   vendor
Servers/Storage  YOU       vendor   vendor   vendor   vendor
Network/DC       YOU       vendor   vendor   vendor   vendor
* you still own your data and who can access it
```

Figure 1.1: Who manages what in each service model

Model	AWS	Azure	GCP	Others
IaaS	EC2, EBS, VPC	Virtual Machines, Managed Disks, VNet	Compute Engine, Persistent Disk, VPC	DigitalOcean Droplets
PaaS	Elastic Beanstalk, RDS, App Runner	App Service, Azure SQL Database	App Engine, Cloud SQL, Cloud Run	Heroku, Render
SaaS	Amazon WorkMail, Chime	Microsoft 365	Google Workspace	Salesforce, Zoho, Slack

Model	AWS	Azure	GCP	Others
FaaS	AWS Lambda	Azure Functions	Cloud Run functions (formerly Cloud Functions)	Cloudflare Workers

1.2.1 FaaS and "serverless"

FaaS runs a single function when an event happens (an HTTP request, a file upload, a message in a queue). You pay per request and per millisecond of run time. "Serverless" is a broader word: any service where you do not manage servers and pay for usage, such as Lambda, DynamoDB on-demand, S3 and SQS. There are still servers, of course. You just never SSH into them.

```
# A tiny Lambda handler in Python
def lambda_handler(event, context):
    name = event.get("name", "world")
    return {"statusCode": 200, "body": f"Namaste, {name}!"}
```

🎯 **INTERVIEW CORNER**

Q: Is RDS IaaS or PaaS? **A:** PaaS. AWS manages the OS, database engine install, patching windows, backups and failover. You manage schema, queries, parameter choices and access.

Q: When would you choose EC2 over Lambda? **A:** When the job runs longer than Lambda's 15-minute maximum, needs a special OS or GPU, keeps long-lived connections, or runs steadily at high load where a reserved instance is cheaper than per-request billing.

Q: In SaaS, who is responsible for data? **A:** The customer still owns the data and its access control (who has accounts, sharing settings). The vendor secures the platform.

✅ **CHECK YOUR PROGRESS 1.2**

1. **Classify: Google Workspace, EC2, Cloud SQL, Azure Functions.**
   Answer: SaaS, IaaS, PaaS, FaaS.
2. **In IaaS, who patches the guest operating system?**
   Answer: The customer. The provider manages the hypervisor and hardware below it.
3. **Give one limit of Lambda that pushes a team to containers or EC2.**
   Answer: The maximum timeout of 15 minutes per invocation (others: memory ceiling of 10,240 MB, no persistent local disk beyond ephemeral /tmp).

1.3 Deployment Models

Service models tell you *what* you rent. Deployment models tell you *where* it lives and *who else* shares it.

Model	What it is	Good for	Watch out for
Public cloud	Shared provider infrastructure (AWS, Azure, GCP).	Startups, most new apps, spiky load.	Bill shock, data residency rules.
Private cloud	Cloud-style self-service on hardware used by one organisation (VMware, OpenStack, or provider-hosted dedicated).	Banks, government, strict compliance.	High cost, you still run hardware.
Hybrid cloud	On-prem plus public cloud, connected by VPN or Direct Connect, managed together.	Gradual migration, legacy mainframes, burst to cloud.	Network complexity, two skill sets.
Multi-cloud	Two or more public clouds in use (for example AWS for apps, GCP for BigQuery analytics).	Best-of-breed services, acquisitions, negotiation leverage.	Duplicate tooling, egress costs, thin expertise.

⚠️ **COMMON MISTAKE**

Saying "we are multi-cloud to avoid vendor lock-in" in an interview without trade-offs. A good answer says: true portability is expensive. Most companies are multi-cloud by accident (acquisitions, one team liked BigQuery) and only design for portability where it matters, using Kubernetes and Terraform.

😄 **FUNNY SIDE**

Hybrid cloud is like a joint family. Some members live in the old house (on-prem), some moved to the new flat (cloud), and everyone still argues on the same WhatsApp group (the VPN tunnel). When the group goes silent, the network engineer gets the call.

✅ **CHECK YOUR PROGRESS 1.3**

1. **A bank keeps core banking on-prem but runs its mobile app backend on AWS, connected by Direct Connect. Which model?**
   Answer: Hybrid cloud.
2. **Give one real cost of multi-cloud.**
   Answer: Data egress charges when moving data between clouds, and the need to train and staff two platforms.

1.4 AWS vs Azure vs GCP

Industry trackers such as Synergy Research Group have for years put the market shares of cloud infrastructure roughly at AWS about 30 percent, Microsoft Azure about 20-25 percent and Google Cloud about 10-13 percent, with the rest shared by Alibaba, Oracle, IBM and others. Numbers move every quarter, so check a recent report before quoting them. The order has stayed the same for a long time.

	AWS	Azure	GCP
Strength	Widest and deepest service catalogue, biggest community, most job posts.	Enterprise and Microsoft shops (Active Directory, Windows, .NET, Office 365), strong hybrid (Azure Arc).	Data and analytics (BigQuery), Kubernetes (GKE, Google created Kubernetes), AI/ML tooling, strong global network.
Typical customer in India	Startups, product companies, e-commerce, media.	IT services firms serving enterprise clients, BFSI on Microsoft stack.	Data-heavy startups, analytics teams, retail.
Entry cert	Cloud Practitioner (CLF-C02)	AZ-900 Fundamentals	Cloud Digital Leader
Core associate cert	Solutions Architect Associate (SAA-C03)	AZ-104 Administrator	Associate Cloud Engineer
Indian regions (examples)	Mumbai (ap-south-1), Hyderabad (ap-south-2)	Central India (Pune), South India (Chennai), West India (Mumbai)	Mumbai (asia-south1), Delhi (asia-south2)

1.4.1 Which cloud should you learn first?

Learn AWS first unless your target employer clearly uses another cloud. Reasons: most job listings mention AWS, its concepts (VPC, IAM, EC2, S3) map cleanly to others, and its certification is widely recognised. Once you know one cloud well, the second takes weeks, not months, because the ideas are the same with different names. This book uses that approach: AWS first, then a mapping table in each chapter.

Concept	AWS	Azure	GCP
Virtual machine	EC2	Virtual Machines	Compute Engine
Object storage	S3	Blob Storage	Cloud Storage
Managed SQL	RDS / Aurora	Azure SQL / Database for PostgreSQL	Cloud SQL / AlloyDB
Functions	Lambda	Functions	Cloud Run functions
Managed Kubernetes	EKS	AKS	GKE
Identity	IAM	Entra ID + Azure RBAC	Cloud IAM
IaC (native)	CloudFormation / CDK	ARM / Bicep	Infrastructure Manager (Terraform-based)

Table 1.1: Core service mapping (cert: Cloud Practitioner tests most of these AWS names)

🎯 **INTERVIEW CORNER**

Q: Why AWS over Azure for a new startup? **A:** Depends on context. AWS has the broadest services, mature startup credits programmes and a large hiring pool. If the team is .NET plus Microsoft 365 with Entra ID already, Azure may reduce friction. I would decide on team skills, needed services, credits and region availability, not brand.

Q: Name a GCP service teams pick even when they run on AWS. **A:** BigQuery, for serverless analytics on large datasets.

✅ **CHECK YOUR PROGRESS 1.4**

1. **What is the Azure equivalent of S3?**
   Answer: Azure Blob Storage.
2. **Which provider created Kubernetes originally?**
   Answer: Google, based on its internal system Borg; it was open-sourced in 2014 and donated to the CNCF.

1.5 Cloud Roles and What Each Interview Tests

"Cloud job" is not one job. Titles overlap and vary by company, but the table below shows the common shape. Read it, then pick **one** primary target. Applying to everything with one resume is the most common reason switchers get no calls.

Role	Day-to-day work	Interview focus	Good fit if you are
Cloud Support Engineer / Associate	Solve customer tickets: "cannot SSH", "S3 403", "RDS slow". Reproduce, explain, document.	Linux, networking basics (DNS, TCP, HTTP), troubleshooting scenarios, communication, AWS fundamentals.	Fresher, sysadmin, tech support, good communicator.
Cloud Engineer	Build and run infra: VPCs, EC2/EKS, IAM, monitoring, IaC with Terraform.	VPC design, IAM policies, Terraform, Linux, scripting (Bash/Python), one live debugging task.	Developer or sysadmin with 1-4 years.
DevOps Engineer	CI/CD pipelines, containers, Kubernetes, IaC, release automation.	Git, Docker, Kubernetes, Jenkins/GitHub Actions, Terraform, Linux, scripting, a pipeline design.	Developer who enjoys automation.
Site Reliability Engineer (SRE)	Reliability of production: SLOs, on-call, incident response, capacity, automation to reduce toil.	Systems design, Linux internals, coding (often a real coding round), incident stories, observability.	Strong coder who likes production systems.
Solutions Architect	Design solutions for customers or internal teams, cost and trade-off discussions, pre-sales (at vendors).	Whiteboard architecture, Well-Architected trade-offs, migration strategy, presentation skills.	4+ years, broad knowledge, enjoys talking to people.

Role	Day-to-day work	Interview focus	Good fit if you are
FinOps Analyst / Engineer	Track and reduce cloud spend, tagging, budgets, Savings Plans coverage, showback to teams.	Pricing models, Cost Explorer, CUR analysis, SQL/Excel, rightsizing, communication with finance.	Analytical person from finance, ops or engineering.

> 💡 **DO YOU KNOW?**
>
> The FinOps Foundation, part of the Linux Foundation, runs the FinOps Certified Practitioner exam. FinOps grew because many companies discovered that their cloud bill had become one of their largest costs after payroll.

1.5.1 A typical interview loop



Figure 1.2: A common interview pipeline for cloud roles in India

- **Recruiter screen (15-30 min):** notice period, current CTC, which cloud, certifications. Keep a two-line pitch ready.
- **Online test:** MCQs on AWS/Linux/networking, sometimes a Bash or Python task.
- **Technical round 1:** fundamentals. "Security group vs NACL", "what happens when you type a URL", "how does an IAM role work on EC2".
- **Technical round 2:** design or live task. "Design a highly available web app for 1 lakh users", or "here is a broken Terraform file, fix it", or a screen-share troubleshooting session.
- **Managerial:** STAR stories: an outage you handled, a cost you cut, a disagreement.
- **HR:** offer, joining date, negotiation.

Amazon itself (for roles such as Cloud Support Associate or Solutions Architect at AWS) adds behavioural questions mapped to its Leadership Principles, such as "Customer Obsession" and "Dive Deep". Prepare 6-8 STAR stories that you can bend to fit them.

> 🎯 **INTERVIEW CORNER**
>
> **Q:** Tell me about a time you reduced cost. **A (STAR):** Situation: our dev environment cost about Rs 2.4 lakh a month. Task: cut it without slowing developers. Action: tagged resources by owner, scheduled dev EC2 and RDS to stop at night via EventBridge and Lambda, moved old logs to S3 Glacier. Result: about 45 percent lower monthly bill in 6 weeks, and developers did not notice.
>
> **Q:** What is the difference between DevOps and SRE? **A:** DevOps is a culture and set of practices to make dev and ops work together and ship faster. SRE is Google's specific way of doing operations with software engineering, using SLOs, error budgets and toil limits. A common line: "SRE implements DevOps."

CHECK YOUR PROGRESS 1.5

1. **Which role is most focused on reducing cloud spend?**
   Answer: FinOps analyst or engineer.
2. **Which round usually checks STAR stories?**
   Answer: The managerial or behavioural round.
3. **Name two technical skills that appear in almost every cloud role interview.**
   Answer: Linux and networking fundamentals (also IAM basics).

1.6 Certification Path

Certifications do not replace skill, but in India they strongly help with recruiter filters, especially for switchers. The AWS path below is the most common.



Figure 1.3: Recommended AWS certification order

Cert	Who should take it	What it tests	Prep time (rough)
Cloud Practitioner	Non-technical or total beginners; optional for developers.	Concepts, billing, shared responsibility, service names.	2-3 weeks
Solutions Architect Associate	Almost everyone targeting cloud roles. The most valuable first cert.	VPC, IAM, EC2, S3, RDS, HA design, cost-aware architecture.	6-8 weeks
SysOps Administrator Associate (CloudOps)	Support, operations, cloud engineer roles.	Monitoring, automation, troubleshooting, deployment. AWS has been renaming this toward "CloudOps Engineer"; check the current exam code.	4-6 weeks after SAA
Developer Associate	Developers building on Lambda, DynamoDB, API Gateway.	SDKs, serverless, CI/CD basics.	4-6 weeks
DevOps Engineer Professional	DevOps/SRE with 2+ years.	CI/CD, IaC at scale, incident response, multi-account.	8-12 weeks
Security Specialty	Cloud security engineers.	IAM deep dive, KMS, logging, incident response, data protection.	8-10 weeks

Exam codes and names change every few years. Always check the AWS Certification website for the current version before buying study material. Exam fees are roughly USD 100 for Cloud Practitioner, USD 150 for Associate and USD 300 for Professional and Specialty; check the current price, and look for AWS's 50 percent discount voucher you get after passing any exam.

Non-AWS add-ons that recruiters value: **Terraform Associate** (HashiCorp), **CKA** (Certified Kubernetes Administrator, a hands-on exam), and for Azure targets **AZ-104**.

> ⚠ **COMMON MISTAKE**
>
> Collecting five certificates from exam dumps with no projects. Interviewers ask one follow-up ("How did you debug that?") and the gap shows. One cert plus two real projects beats four certs and zero projects.

> ✅ **CHECK YOUR PROGRESS 1.6**
>
> 1. **Which AWS cert gives the most value as a first serious cert for a developer?**
>
>   **Answer:** Solutions Architect Associate (SAA-C03).
>
> 2. **Name one hands-on, performance-based cert useful for DevOps roles.**
>
>   **Answer:** CKA (Certified Kubernetes Administrator).

1.7 The 90-Day Switch Plan

This plan assumes you have a full-time job and can give about 2 hours on weekdays and 5-6 hours on weekends (roughly 20 hours a week). Adjust if you have more time.

Weeks	Focus	Output by end
1-2	Linux (files, permissions, systemd, logs), networking (IP, CIDR, DNS, HTTP, TLS), Git.	Notes repo on GitHub; can explain "what happens when you open a URL".
3-4	AWS account setup safely, IAM, VPC, EC2, S3 (Chapters 2-4 of this book).	Hand-built 2-tier VPC with a web server; screenshots and README.
5-6	RDS, Load Balancer, Auto Scaling, Route 53, CloudWatch.	Highly available web app behind ALB; Project 1 on GitHub.
7-8	Terraform; rebuild Project 1 as code. SAA-C03 practice tests.	Terraform repo; book the SAA exam.
9-10	Docker, CI/CD with GitHub Actions, intro to ECS or EKS. Pass SAA.	Project 2: container app with pipeline; certificate on LinkedIn.
11-12	Serverless (Lambda, API Gateway, DynamoDB), cost and monitoring. Resume, LinkedIn, mock interviews, applications.	Project 3; 30+ targeted applications; 3 mock interviews done.
13	Interview loop, revise weak areas from mocks.	Offers or clear feedback to fix.

🔑 **KEY POINT**

Start applying around week 9, not week 13. Interview loops take 3-6 weeks in India, and early interviews show you what to study. Notice periods of 60-90 days also mean you should start early.

1.7.1 Portfolio projects that impress

1. **Three-tier web app on AWS with Terraform:** VPC with public/private subnets across 2 AZs, ALB, Auto Scaling group, RDS in private subnets, CloudWatch alarms. Include an architecture diagram and monthly cost estimate.
2. **Serverless URL shortener or image resizer:** API Gateway + Lambda + DynamoDB (or S3 trigger + Lambda). Show IAM least-privilege policies.
3. **CI/CD to containers:** GitHub Actions builds a Docker image, pushes to ECR, deploys to ECS Fargate, using OIDC federation instead of long-lived keys.
4. **Cost and security automation:** a Lambda that finds unattached EBS volumes and untagged instances and posts a daily report to Slack or email.
5. **Static site with CDN:** S3 + CloudFront + ACM certificate + Route 53, deployed by pipeline. Small, but shows DNS and TLS skills.

Every repo needs: a README with architecture diagram, "how to deploy", "how to destroy", estimated cost, screenshots, and a "what I learned / what broke" section. The "what broke" section is gold in interviews.

🧪 **HANDS-ON LAB 1.1: BUILD YOUR CLOUD LEARNING WORKSPACE**

1. Create a GitHub repo called `cloud-journey` with folders `notes/` , `labs/` , `projects/` .
2. Install AWS CLI v2, Terraform and Git on your laptop (setup details in Chapter 2). Verify each: `aws --version` gives output like `aws-cli/2.x.x Python/3.x Windows/10 exe/AMD64` ; `terraform -version` and `git --version` print versions.
3. Create `notes/nist.md` with the five NIST characteristics and one AWS example each, in your own words.
4. Create `notes/roles.md` with your chosen target role and 10 skills it needs, marked green/amber/red for your current level.
5. Commit and push. This repo becomes your study log; recruiters do look.
6. Clean up to avoid charges: nothing was created in AWS in this lab, so there is nothing to delete. Make it a habit to write this line in every lab note anyway.

✅ **CHECK YOUR PROGRESS 1.7**

1. **Why should applications start around week 9?**

   Answer: Interview loops take weeks, early interviews reveal gaps, and notice periods are long.

2. **List four sections every portfolio README should have.**

   Answer: Architecture diagram, deploy steps, destroy steps, cost estimate (also screenshots and lessons learned).

1.8 Resume and Interview Tips for Cloud Roles

1.8.1 Resume rules

- **One page** if under 5 years experience; two at most.
- **Headline matches the target:** "Cloud Engineer | AWS SAA | Terraform | Kubernetes", not "Software Engineer seeking opportunities".
- **Bullets use action + tech + number:** "Migrated 14 VMs to EC2 with Terraform, cutting provisioning time from 3 days to 20 minutes."
- **Skills section grouped:** Cloud (AWS: EC2, VPC, IAM, S3, RDS, Lambda), IaC (Terraform, CloudFormation), Containers (Docker, EKS), CI/CD, Scripting, Monitoring.
- **Projects section** with GitHub links, if your job did not involve cloud.
- **ATS friendly:** plain layout, no tables or images for text, standard headings.

1.8.2 Turning current work into cloud experience

Switchers often already touched cloud without calling it that. Did you deploy to a server, write a Dockerfile, fix a Jenkins job, read CloudWatch logs, or change an S3 bucket? Ask your current team for a small cloud task (automate a report, add an alarm). One real production bullet is worth more than any side project.

> 💼 **AT WORK**
>
> Priya, a Java developer at a services company, wanted to move to a cloud role. She volunteered to fix the team's slow Jenkins deploys to EC2. She moved the build to GitHub Actions, used an IAM role via OIDC instead of stored access keys, and added a CloudWatch alarm on 5xx errors. It took four weekends. In interviews she told this story with numbers (deploy time 25 minutes to 7, zero stored keys). She got two Cloud Engineer offers within two months. Lesson: find cloud work inside your current job first.

1.8.3 Troubleshooting your job search

Symptom	Likely cause	Fix
No recruiter calls after 40 applications	Resume not matching keywords; generic headline.	Tailor headline and skills to the job description; update Naukri/LinkedIn profile with the same keywords; get referrals.
Calls come, but you fail round 1	Weak fundamentals (networking, Linux, IAM).	Revise Chapters 2-4; practise explaining aloud; do mock interviews.
Pass tech rounds, fail design	No structured approach to design.	Use a fixed template: requirements, diagram, HA, security, cost, trade-offs.
Offers are low	Positioning as fresher in cloud.	Count your total engineering years; show production bullets; have a competing offer.

Symptom	Likely cause	Fix
Stuck in "tutorial hell"	Watching courses without building.	Rule: for every hour of video, two hours of hands-on labs.

🎯 **INTERVIEW CORNER**

Q: You have no production cloud experience. Why should we hire you? **A:** Be honest and specific: "My production work is in Java services, which I deployed and debugged on Linux for 3 years. On cloud, I built a Terraform-managed 3-tier app across two AZs, passed SAA, and documented failures I fixed, like a NAT route mistake. I learn fast and can show it in my repo."

Q: What happens when you type a URL in the browser? **A:** DNS resolution (browser cache, OS, resolver, root, TLD, authoritative), TCP handshake, TLS handshake, HTTP request, server or load balancer processing, response, rendering. In AWS terms, Route 53 answers DNS, CloudFront or an ALB terminates TLS, and EC2 or Lambda handles the request.

✅ **CHECK YOUR PROGRESS 1.8**

1. **Rewrite this bullet with numbers: "Worked on AWS infrastructure."**
   Answer: Example: "Built Terraform modules for VPC and EC2 used by 4 teams, reducing new environment setup from 2 days to 30 minutes."
2. **What should you do if you pass technical rounds but fail design?**
   Answer: Practise with a structured design template covering requirements, diagram, availability, security, cost and trade-offs.

What Have We Learnt?

- Cloud computing is on-demand, pay-as-you-go access to IT resources over a network.
- NIST lists five characteristics: self-service, broad network access, pooling, rapid elasticity, measured service.
- IaaS, PaaS, SaaS and FaaS differ in how much the provider manages.
- Public, private, hybrid and multi-cloud are deployment models, each with trade-offs.
- AWS leads the market, Azure is strong in enterprise and Microsoft shops, GCP in data, analytics and Kubernetes.
- Cloud roles (support, cloud engineer, DevOps, SRE, solutions architect, FinOps) have different interview focuses.
- SAA-C03 is the most valuable first AWS cert for most switchers; pair it with projects.
- A 90-day plan with weekly outputs and early applications beats endless course watching.
- Resumes need a targeted headline and bullets with action, technology and numbers.

Key terms: NIST SP 800-145, on-demand self-service, resource pooling, rapid elasticity, measured service, IaaS, PaaS, SaaS, FaaS, serverless, public cloud, private cloud, hybrid cloud, multi-cloud, CapEx, OpEx, SRE, FinOps, SAA-C03, STAR method.

🔑 **RESUME LINE**

"Designed and deployed a 3-tier web application on AWS (VPC, ALB, Auto Scaling, RDS Multi-AZ) using Terraform, with a documented monthly cost of about USD 45 and full teardown in under 5 minutes."
"AWS Certified Solutions Architect Associate; automated nightly shutdown of dev resources, cutting non-prod spend by about 40 percent."

Exercises

A. Multiple Choice Questions

1. Which is NOT one of the NIST essential characteristics?
   a. Rapid elasticity
   b. Measured service
   c. Vendor lock-in
   d. Resource pooling

2. AWS Lambda is an example of:
   a. IaaS
   b. FaaS
   c. SaaS
   d. Private cloud

3. A company uses AWS for apps and GCP BigQuery for analytics. This is:
   a. Hybrid cloud
   b. Private cloud
   c. Multi-cloud
   d. Community cloud

4. In IaaS, who patches the guest OS?
   a. The provider
   b. The customer
   c. Nobody
   d. The hypervisor

5. Which role focuses mainly on SLOs, error budgets and reducing toil?
   a. FinOps
   b. Solutions Architect
   c. SRE
   d. Cloud Support

6. The Azure equivalent of EC2 is:
   a. Azure Functions
   b. Azure Virtual Machines
   c. App Service
   d. AKS

B. Short Answer Questions

1. Explain the difference between PaaS and FaaS with one example each.
2. Give two reasons a bank might choose hybrid cloud.
3. Why is SAA-C03 often recommended over Cloud Practitioner for developers?
4. What are three things that should be in every portfolio project README?

C. Scenario Questions

1. Your manager says, "Our AWS bill doubled this month and finance is asking why." You are the only cloud person. What do you do?
2. A startup founder asks you whether to run their new API on EC2 or Lambda. Traffic is low now but may spike after a marketing campaign. What do you recommend and why?
3. You applied to 50 cloud jobs in 3 weeks and got 1 call. What do you change?

D. Mini Project

Create a public GitHub repo `cloud-career-plan` containing: your target role, a skills gap table, your 90-day calendar as a Markdown table, a certification plan with dates, and three project ideas with one architecture sketch each (ASCII is fine). Pin it on your GitHub profile.

Answer Key: Chapter 1

A. MCQ

1. (c) Vendor lock-in is a risk, not a NIST characteristic.
2. (b) Lambda runs functions on events, billed per request and duration.
3. (c) Two public clouds together is multi-cloud; hybrid means on-prem plus cloud.
4. (b) In IaaS the customer manages the OS and above.
5. (c) SLOs and error budgets are core SRE practices.
6. (b) Azure Virtual Machines is the IaaS VM service.

B. Short Answers

1. PaaS runs your whole application on a managed platform that stays up (Elastic Beanstalk, App Service); you deploy an app and the platform handles OS and runtime. FaaS runs individual functions only when triggered, scales to zero and bills per invocation (Lambda, Azure Functions).
2. Regulations or risk policy keep core banking data on-prem; legacy mainframe systems are hard to move; while newer customer-facing apps benefit from cloud elasticity.
3. Cloud Practitioner tests concepts and billing; SAA tests real architecture (VPC, IAM, HA, storage choices) which matches what cloud jobs need, and recruiters treat it as the baseline technical cert.
4. Architecture diagram, deploy and destroy steps, cost estimate (plus screenshots and lessons learned).

C. Scenario Questions

1. Step 1: open Cost Explorer, group by service, then by usage type and region, compare with last month. Step 2: find the top change (common culprits: NAT Gateway data processing, forgotten large EC2 or RDS, data transfer out, a dev environment left running, logs ingestion). Step 3: check tags or CloudTrail to find who created it. Step 4: fix (stop, rightsize, delete with owner approval). Step 5: prevent: AWS Budgets alerts, mandatory tags, Cost Anomaly Detection. Step 6: send finance a short written explanation with numbers.
2. Recommend Lambda behind API Gateway (or a Lambda function URL) to start: no idle cost, automatic scaling for spikes, less ops work. Mention limits: 15-minute timeout, cold starts, concurrency quota (default account limit is often 1,000 concurrent executions per region; check Service Quotas). If traffic becomes steady and high, revisit containers on ECS Fargate or EC2 with Savings Plans.
3. Pick one target role; rewrite the headline and skills to match job descriptions; add numbers to bullets; link 2-3 strong GitHub projects; update Naukri and LinkedIn with the same keywords and mark "open to work" for recruiters; ask for referrals from ex-colleagues; apply to fewer, better-matched jobs.

D. Mini Project: Model Solution

Repo layout: `README.md` (target role: Cloud Engineer; summary), `skills-gap.md` (table with columns Skill, Current level, Target, Resource), `90-day-plan.md` (weeks 1-13 with outputs from Section 1.7), `certs.md` (SAA-C03 booked for week 9, Terraform Associate for week 14), `projects.md` (three-tier

Terraform app, serverless URL shortener, GitHub Actions to ECS pipeline, each with an ASCII diagram and expected monthly cost). Update weekly with a short progress log; the commit history itself proves consistency.

CHAPTER 2

Cloud Foundations: Regions, Pricing and the Well-Architected Framework

"The first AWS lesson most people learn is not about servers. It is about the billing alarm they forgot to set."

After studying this chapter, you will be able to

- Explain regions, Availability Zones, edge locations and Local Zones, and choose a region.
- Apply the shared responsibility model to IaaS, PaaS and serverless services.
- Use the six Well-Architected pillars to review a design.
- Compare On-Demand, Reserved Instances, Savings Plans and Spot pricing.
- Set up an AWS account safely with root MFA, a billing alarm and an admin identity.
- Install and configure AWS CLI v2 with profiles, and check service quotas.

Imagine a big supermarket chain like a national retailer. It has huge warehouses in a few cities (regions), each warehouse is really several buildings with separate power supply so one fire does not stop deliveries (Availability Zones), and small corner pickup points in hundreds of towns (edge locations). It also has a price list with discounts for bulk monthly buying and cheap clearance-sale items that can be taken back any time. AWS is built and priced exactly like that.

This chapter covers the foundations every interview assumes you know: where AWS runs, who is responsible for what, how AWS judges a "good" design, how pricing works, and how to set up your own account without waking up to a Rs 40,000 bill.

2.1 AWS Global Infrastructure

2.1.1 Regions

A **Region** is a separate geographic area, such as `ap-south-1` (Mumbai) or `us-east-1` (N. Virginia). Regions are isolated from each other. Most services are regional: an EC2 instance created in Mumbai does not appear in Singapore. AWS has more than 30 regions and keeps adding more; check the AWS Global Infrastructure page for the current count.

How to choose a region:

1. **Compliance and data residency:** some data must stay in India (for example, RBI guidelines on payment data storage).
2. **Latency:** pick a region close to your users.
3. **Service availability:** new services often launch in `us-east-1` first.
4. **Price:** the same instance type costs different amounts in different regions.

2.1.2 Availability Zones

An **Availability Zone (AZ)** is one or more discrete data centres with redundant power, networking and cooling, inside a region. Most regions have three or more AZs, connected by low-latency private fibre. AZs are far enough apart that a flood or power failure in one usually does not affect the other, but close enough for synchronous replication (single-digit milliseconds).

AZ names like `ap-south-1a` are mapped randomly per account, so your "1a" may be a different building than your friend's "1a". To compare across accounts, use the **AZ ID** such as `aps1-az1`.

```
$ aws ec2 describe-availability-zones --region ap-south-1 \
    --query "AvailabilityZones[].{Name:ZoneName,Id:ZoneId,State:State}" --output table
-------------------------------------------
|       DescribeAvailabilityZones         |
+-----------+-------------+-------------+
|    Id     |    Name     |    State    |
+-----------+-------------+-------------+
|  aps1-az1 |  ap-south-1a |  available  |
|  aps1-az3 |  ap-south-1b |  available  |
|  aps1-az2 |  ap-south-1c |  available  |
+-----------+-------------+-------------+
```

2.1.3 Edge locations, Local Zones and others

Type	What it is	Used by
Edge location / Point of Presence	Hundreds of small sites in many cities for caching and DNS.	CloudFront (CDN), Route 53, AWS Global Accelerator, Shield.
Regional edge cache	Larger cache between edge and origin.	CloudFront.
Local Zone	AWS compute and storage placed in a big city, attached to a parent region, for single-digit ms latency. Example: Delhi and Kolkata Local Zones attached to Mumbai.	Gaming, media editing, real-time apps.
Wavelength Zone	AWS inside telecom 5G networks.	Ultra-low latency mobile apps.
Outposts	AWS hardware racks installed in your own data centre.	Hybrid with strict on-prem needs.

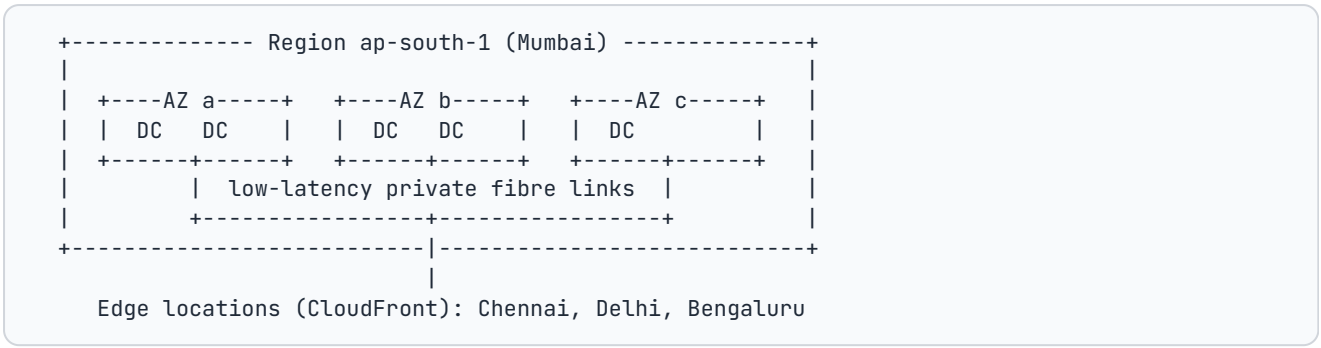


Figure 2.1: A region contains multiple AZs; edge locations sit closer to users

> 💡 **DO YOU KNOW?**
>
> Some AWS services are global, not regional. IAM, Route 53, CloudFront and AWS Organizations are global. Their console page shows "Global" in the region selector. Interviewers like asking "Is IAM regional?" (answer: no, global).

> 🎯 **INTERVIEW CORNER**
>
> **Q:** What is the difference between a region and an AZ? **A:** A region is a geographic area with multiple isolated AZs. An AZ is one or more data centres with independent power and networking. You deploy across AZs for high availability and across regions for disaster recovery or global latency.
>
> **Q:** How do you make an app highly available within a region? **A:** Run instances in at least two AZs behind a load balancer, use Multi-AZ databases, and store state in regional services like S3 or DynamoDB.
>
> **Q:** Why might two accounts both see "ap-south-1a" but be in different data centres? **A:** AZ names are mapped per account to spread load. Use AZ IDs (aps1-az1) to match.

> ✅ **CHECK YOUR PROGRESS 2.1**
>
> 1. **Name four factors for choosing a region.**
>   **Answer:** Compliance/data residency, latency to users, service availability, price.
> 2. **Which AWS service uses edge locations to cache content?**
>   **Answer:** Amazon CloudFront.
> 3. **Is IAM a regional service?**
>   **Answer:** No, IAM is global.

2.2 The Shared Responsibility Model

AWS says it is responsible for security **"of"** the cloud, and the customer is responsible for security **"in"** the cloud. Think of renting a flat: the builder is responsible for the building structure, lift and main gate. You are responsible for locking your own door and not giving the key to strangers.

- **AWS:** physical data centres, hardware, global network, hypervisor, and the software of managed services.
- **Customer:** data, IAM (who can access what), encryption choices, network rules (security groups), and whatever AWS does not manage for that service type.

Task	EC2 (IaaS)	RDS (managed)	Lambda / S3 (serverless)
Physical security, hardware	AWS	AWS	AWS
Hypervisor and host OS	AWS	AWS	AWS
Guest OS patching	Customer	AWS	AWS
Database engine patching	Customer (if self-installed)	AWS applies, customer picks maintenance window / version	n/a
Application code and dependencies	Customer	Customer (queries, schema)	Customer (function code, libraries)
Network firewall rules	Customer (SG, NACL)	Customer (SG, public access setting)	Customer (bucket policies, Block Public Access)
IAM and data encryption settings	Customer	Customer	Customer

🔑 **KEY POINT**

The more managed the service, the less you patch. But you **always** own IAM, your data and its configuration. Almost every famous cloud breach (public S3 buckets, leaked access keys) was on the customer side of the line.

✅ **CHECK YOUR PROGRESS 2.2**

1. **Who patches the OS of an RDS instance?**
   Answer: AWS.
2. **Who is responsible for an S3 bucket accidentally made public?**
   Answer: The customer.

2.3 The AWS Well-Architected Framework

The Well-Architected Framework is AWS's checklist for good design. It has **six pillars**. The sixth, Sustainability, was added in December 2021, so older material shows only five.

Pillar	Core question	Example practices
Operational Excellence	Can we run and improve it smoothly?	IaC, small reversible changes, runbooks, post-incident reviews.

Pillar	Core question	Example practices
Security	Is data and access protected?	Least privilege, MFA, encryption at rest and in transit, logging with CloudTrail.
Reliability	Does it recover from failure?	Multi-AZ, Auto Scaling, backups tested, health checks.
Performance Efficiency	Are we using the right resources?	Right instance family, caching, serverless where it fits, CDN.
Cost Optimization	Are we paying only for value?	Rightsizing, Savings Plans, Spot, stop idle resources, tagging.
Sustainability	Are we minimising environmental impact?	Graviton (ARM) instances, high utilisation, delete unused data.

Mnemonic: **"O S R P C S"**, or "Our Servers Rarely Panic, Code Sleeps". The free **AWS Well-Architected Tool** in the console lets you answer pillar questions about a workload and get a list of high and medium risks.

🎯 **INTERVIEW CORNER**

Q: Name the six pillars. **A:** Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization, Sustainability.

Q: Give a trade-off between two pillars. **A:** Reliability vs Cost: Multi-AZ RDS roughly doubles database cost but gives automatic failover. For a dev environment you accept Single-AZ; for production payments you pay for Multi-AZ.

Q: How would you use the framework in a design round? **A:** After drawing the architecture, walk through each pillar briefly: how it is deployed (IaC), secured (IAM, encryption), made reliable (multi-AZ), sized, priced, and kept efficient.

✅ **CHECK YOUR PROGRESS 2.3**

1. **Which pillar was added last?**
   Answer: Sustainability (2021).
2. **Testing backups belongs mostly to which pillar?**
   Answer: Reliability.

2.4 Pricing Models

AWS pricing follows three basic drivers: **compute** (time running), **storage** (GB-month) and **data transfer out** (data leaving AWS to the internet or crossing regions). Data transfer *into* AWS from the internet is generally free. Data between AZs is charged a small amount per GB in each direction. All numbers below are "roughly"; always check the pricing page and the AWS Pricing Calculator.

2.4.1 Compute purchase options

Option	Commitment	Discount vs On-Demand (roughly)	Best for
On-Demand	None; pay per second (Linux, 60-second minimum) or per hour.	0 percent	Spiky, short-term, unknown workloads, dev/test.
Reserved Instances (Standard / Convertible)	1 or 3 years for a specific instance family in a region.	Up to about 72 percent	Steady databases (RDS RIs are common), legacy.
Savings Plans (Compute / EC2 Instance)	1 or 3 years of a fixed USD per hour spend.	Up to about 72 percent (EC2 Instance SP), up to about 66 percent (Compute SP)	Steady compute that may change family, region, or move to Fargate/Lambda (Compute SP).
Spot Instances	None; AWS can reclaim with a 2-minute warning.	Up to about 90 percent	Fault-tolerant jobs: batch, CI runners, data processing, stateless containers.
Dedicated Hosts	Physical server for you.	Costs more	Licensing tied to physical cores, compliance.

Payment options for RIs and Savings Plans: All Upfront (biggest discount), Partial Upfront, No Upfront.

> 😄 **FUNNY SIDE**
>
> On-Demand is the auto-rickshaw: flexible, metered, a little expensive. Savings Plans are the monthly metro pass: cheap if you actually travel every day, sad if you start working from home. Spot is the shared jeep at the station: very cheap, but the driver may ask you to get down two minutes before your stop.

2.4.2 Free Tier

The AWS Free Tier has changed over time. Historically it had three kinds of offer: **12 months free** (for example 750 hours a month of a t2.micro or t3.micro EC2 instance, 5 GB of S3 Standard), **always free** (for example 1 million Lambda requests a month, 25 GB DynamoDB storage) and **short trials**. In July 2025 AWS introduced a new model for new accounts: a free plan with credits (around USD 100 at sign-up plus up to about USD 100 more for completing activities), valid for a limited period (about 6 months), with a choice of a "free account plan" that cannot be charged until you upgrade, or a paid plan. Accounts created before that keep the older terms. Rules differ by account age, so read the current Free Tier page and the Billing console's "Free Tier" panel for your account.

> ⚠️ **COMMON MISTAKE**
>
> Thinking "Free Tier" means "cannot be charged". NAT Gateways, Elastic IPs not attached to running instances (and since February 2024 all public IPv4 addresses, roughly USD 0.005 per hour each), large instance types, and data transfer are common sources of surprise bills. Always set a budget alert first.

2.4.3 Tools for cost visibility

- **AWS Pricing Calculator:** estimate before you build.
- **Cost Explorer:** charts of actual spend by service, region, tag.
- **AWS Budgets:** alerts when actual or forecast spend crosses a threshold.
- **Cost Anomaly Detection:** machine-learning alerts on unusual spend.
- **Cost and Usage Report (CUR / Data Exports):** line-item data to S3 for Athena queries; FinOps teams live in it.
- **Cost allocation tags:** tags like `Project`, `Owner` activated in Billing so costs can be grouped.

> ✅ **CHECK YOUR PROGRESS 2.4**
>
> 1. **Which option gives the biggest discount but can be interrupted?**
>
>   **Answer:** Spot Instances, with a 2-minute interruption notice.
> 2. **Which Savings Plan also covers Fargate and Lambda?**
>
>   **Answer:** Compute Savings Plans.
> 3. **Is inbound data transfer from the internet charged?**
>
>   **Answer:** Generally no; outbound data transfer is charged.

2.5 Setting Up an AWS Account Safely

A new AWS account has one super-user: the **root user** (the email you signed up with). Root can do everything, including closing the account and changing payment. Treat it like the original property papers kept in a bank locker. Use it only for the few tasks that require it.

2.5.1 The first-hour checklist

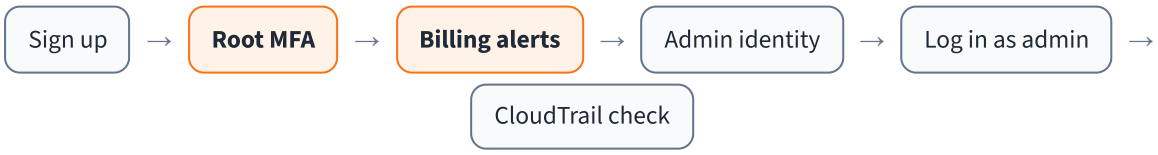


Figure 2.2: Safe account setup order

1. **Enable MFA on root:** Console, top right account name, Security credentials, Assign MFA device. Use an authenticator app or a hardware/passkey. AWS now enforces MFA for root users in most account types; enable it yourself anyway.
2. **Do not create root access keys.** If any exist, delete them.

3. **Turn on billing alerts:** Billing and Cost Management, Budgets, create a monthly cost budget (for example USD 5) with email alerts at 50, 80 and 100 percent of actual and at 100 percent forecasted. Also allow IAM users access to billing ("IAM user and role access to Billing information") so your admin can see costs.
4. **Create an admin identity:** preferred is **IAM Identity Center** with a user and the `AdministratorAccess` permission set (covered in Chapter 3). For a simple personal lab account, an IAM user in an `Admins` group with MFA is acceptable.
5. **Sign out of root** and use the admin identity daily.
6. **Verify CloudTrail:** Event history is on by default for 90 days of management events. For production, create a trail that stores logs in S3.

```
# Create a USD 5 monthly budget with an email alert using the CLI
$ cat budget.json
{
  "BudgetName": "monthly-5-usd",
  "BudgetLimit": { "Amount": "5", "Unit": "USD" },
  "TimeUnit": "MONTHLY",
  "BudgetType": "COST"
}
$ cat notify.json
[
  {
    "Notification": {
      "NotificationType": "ACTUAL",
      "ComparisonOperator": "GREATER_THAN",
      "Threshold": 80,
      "ThresholdType": "PERCENTAGE"
    },
    "Subscribers": [ { "SubscriptionType": "EMAIL", "Address": "you@example.com" } ]
  }
]
$ aws budgets create-budget \
    --account-id 111122223333 \
    --budget file://budget.json \
    --notifications-with-subscribers file://notify.json
# (no output on success)
```

> 💼 **AT WORK**
>
> On a Monday morning, a startup's finance head forwards an AWS invoice that is 6 times higher than usual. The cloud engineer opens Cost Explorer, groups by region, and sees large EC2 charges in `us-east-2` and `sa-east-1`, regions the company never uses. CloudTrail shows `RunInstances` calls from an access key belonging to a former intern, leaked in a public GitHub repo. Fix: deactivate and delete the key, terminate the crypto-mining instances in every region, rotate all other keys, open a case with AWS Support (they sometimes give billing credits for first-time compromise), add an SCP that denies unused regions, and enable GuardDuty. Lesson: budgets plus no long-lived keys would have caught or prevented it in hours, not days.

CHECK YOUR PROGRESS 2.5

1. **Why should you not use the root user daily?**
   Answer: Root has unrestricted access that cannot be limited by IAM policies; if compromised, the attacker controls everything including billing and account closure.
2. **Name two tasks that need root.**
   Answer: Changing account settings such as root email or closing the account, changing the support plan, restoring IAM admin access if locked out.

2.6 AWS CLI v2: Setup and Profiles

2.6.1 Installing

```
# Windows: download and run the MSI
msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi

# macOS
curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
sudo installer -pkg AWSCLIV2.pkg -target /

# Linux x86_64
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install

$ aws --version
aws-cli/2.x.x Python/3.x.x Linux/6.x exe/x86_64
```

Do not install the old v1 with `pip install awscli` for new work. v2 has SSO support, auto-prompt and better defaults.

2.6.2 Configuring credentials

Best practice is short-lived credentials via IAM Identity Center:

```
$ aws configure sso
SSO session name (Recommended): my-sso
SSO start URL [None]: https://d-1234567890.awsapps.com/start
SSO region [None]: ap-south-1
SSO registration scopes [sso:account:access]:
# browser opens, you approve, then choose account and role
CLI default client Region [None]: ap-south-1
CLI default output format [None]: json
CLI profile name [AdministratorAccess-111122223333]: lab-admin

$ aws sso login --profile lab-admin
$ aws sts get-caller-identity --profile lab-admin
{
    "UserId": "AROAEXAMPLEID:priya",
    "Account": "111122223333",
    "Arn": "arn:aws:sts::111122223333:assumed-
role/AWSReservedSSO_AdministratorAccess_abc123/priya"
}
```

For a simple lab with an IAM user and access keys (less safe; rotate and never commit):

```
$ aws configure --profile lab-user
AWS Access Key ID [None]: AKIAEXAMPLEKEY
AWS Secret Access Key [None]: ****************************************
Default region name [None]: ap-south-1
Default output format [None]: json
```

2.6.3 Where settings live and precedence

```
# ~/.aws/config
[default]
region = ap-south-1
output = json

[profile lab-admin]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = AdministratorAccess
region = ap-south-1

[sso-session my-sso]
sso_start_url = https://d-1234567890.awsapps.com/start
sso_region = ap-south-1
sso_registration_scopes = sso:account:access

# ~/.aws/credentials  (only for static keys)
[lab-user]
aws_access_key_id = AKIAEXAMPLEKEY
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Credential precedence (simplified): command-line options, then environment variables (`AWS_ACCESS_KEY_ID`, `AWS_PROFILE`), then the credentials and config files, then container credentials, then the EC2 instance profile. When something uses the "wrong" account, check environment variables first.

```
$ export AWS_PROFILE=lab-admin        # Linux/macOS
$ setx AWS_PROFILE lab-admin          # Windows (new terminals)
$ aws s3 ls --profile lab-user        # override per command
$ aws ec2 describe-instances --query "Reservations[].Instances[].[InstanceId,State.Name]" --
output table
```

🪛 **HANDS-ON LAB 2.1: SAFE ACCOUNT AND CLI IN 45 MINUTES**

1. Sign in as root. Enable MFA on root. Confirm no root access keys exist.
2. Create a USD 5 monthly cost budget with alerts at 80 percent actual and 100 percent forecasted (console or the CLI above).
3. Enable IAM Identity Center in your home region, create a user for yourself, assign the `AdministratorAccess` permission set to your account. Accept the email invite and set MFA.
4. Install AWS CLI v2. Run `aws configure sso` with profile name `lab-admin`, then `aws sts get-caller-identity --profile lab-admin`. Confirm the ARN shows an `assumed-role/AWSReservedSSO_...`.
5. Run `aws ec2 describe-regions --query "Regions[].RegionName" --output text --profile lab-admin` to see enabled regions.
6. Open Service Quotas, search "Running On-Demand Standard", and note your vCPU limit.
7. Clean up to avoid charges: Identity Center, budgets (the first two action-enabled budgets are free; check the current Budgets pricing) and IAM are free. Keep them. Make sure you did not launch any instance; run `aws ec2 describe-instances --profile lab-admin` and confirm it is empty in your region.

✅ **CHECK YOUR PROGRESS 2.6**

1. **Which command shows which identity your CLI is using?**
   Answer: `aws sts get-caller-identity`.
2. **Which wins: `AWS_PROFILE` env var or the `[default]` profile?**
   Answer: The environment variable; `--profile` on the command line wins over both.

2.7 Console vs CLI vs SDK vs IaC

Tool	Good for	Weakness
Management Console	Learning, exploring, one-off checks, dashboards.	Not repeatable; "click-ops" drifts and is hard to review.
CLI (and CloudShell)	Scripts, quick queries, automation in shell.	Imperative; scripts get messy for full environments.
SDK (boto3, AWS SDK for JS, Java)	Applications and automation logic (Lambda tools, custom checks).	You write and maintain code, error handling, retries.
IaC (CloudFormation, CDK, Terraform)	Whole environments, repeatable, reviewed in pull requests, drift detection.	Learning curve; state management (Terraform).

```
# Same task four ways: create an S3 bucket
# CLI
aws s3api create-bucket --bucket priya-lab-2026-logs --region ap-south-1 \
    --create-bucket-configuration LocationConstraint=ap-south-1

# Python SDK (boto3)
import boto3
s3 = boto3.client("s3", region_name="ap-south-1")
s3.create_bucket(Bucket="priya-lab-2026-logs",
    CreateBucketConfiguration={"LocationConstraint": "ap-south-1"})

# CloudFormation (YAML)
Resources:
  LogsBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: priya-lab-2026-logs

# Terraform (HCL)
resource "aws_s3_bucket" "logs" {
  bucket = "priya-lab-2026-logs"
}
```

Note the S3 quirk: in `us-east-1` you must **not** pass `LocationConstraint`; in every other region you must. It is a classic beginner error (`IllegalLocationConstraintException`).

AWS CloudShell is a browser terminal in the console with the CLI pre-installed and authenticated as your console identity. Great when you are on a locked-down office laptop.

✅ **CHECK YOUR PROGRESS 2.7**

1. **Why is IaC preferred over console clicks in production?**

   Answer: It is repeatable, version-controlled, reviewable in pull requests and can recreate environments or detect drift.

2. **What is AWS CloudShell?**

   Answer: A browser-based shell in the console with AWS CLI pre-installed, using your console credentials.

2.8 Service Limits, Quotas and Troubleshooting

Every account has **service quotas** (formerly "limits") to protect you and AWS. Some are adjustable (request an increase), some are hard. Examples (defaults vary; check Service Quotas for your account):

- EC2 On-Demand Standard instances: measured in **vCPUs** per region (new accounts can start low).
- VPCs per region: 5 (adjustable).
- Elastic IP addresses per region: 5 (adjustable).

- Lambda concurrent executions per region: commonly 1,000 (adjustable; new accounts may be lower).
- S3 buckets per account: 10,000 by default since 2024 (was 100), adjustable further.

```
$ aws service-quotas get-service-quota --service-code ec2 \
    --quota-code L-1216C47A --region ap-south-1 \
    --query "Quota.{Name:QuotaName,Value:Value,Adjustable:Adjustable}"
{
    "Name": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances",
    "Value": 32.0,
    "Adjustable": true
}

$ aws service-quotas request-service-quota-increase --service-code ec2 \
    --quota-code L-1216C47A --desired-value 64 --region ap-south-1
```

Separately, APIs have **rate limits** (throttling). Bursts of API calls get `ThrottlingException` or `RequestLimitExceeded`. SDKs retry with exponential backoff; tune retry mode with `AWS_RETRY_MODE=adaptive` or `standard`.

2.8.1 Troubleshooting

Symptom / error	Cause	Fix
`Unable to locate credentials. You can configure credentials by running "aws configure".`	No profile set, wrong profile name, or env var missing.	Run `aws configure list`; set `--profile` or `AWS_PROFILE`.
`Error when retrieving token from sso: Token has expired and refresh failed`	SSO session expired.	`aws sso login --profile lab-admin`.
`An error occurred (VcpuLimitExceeded) when calling the RunInstances operation`	vCPU quota reached in the region.	Terminate unused instances or request quota increase.
`InsufficientInstanceCapacity`	AWS has no capacity for that type in that AZ right now.	Try another AZ or instance type; use multiple types in Auto Scaling.
`SignatureDoesNotMatch` or `InvalidClientTokenId`	Wrong secret, deleted key, or laptop clock skew.	Re-enter keys; check key is active; sync system time.
"My resources disappeared!"	Console set to a different region.	Check region selector; use `--region`; Tag Editor or Resource Explorer to search all regions.
Bill higher than expected	NAT Gateway, public IPv4, idle resources in other regions.	Cost Explorer by service and region; delete; set budgets.

🎯 **INTERVIEW CORNER**

Q: A RunInstances call fails with VcpuLimitExceeded. What do you do? **A:** Check current usage and quota in Service Quotas, terminate unused instances if possible, request an increase for the right quota code and region, and in the meantime use another region or smaller instances if urgent.

Q: How do you avoid "surprise bills" in a team account? **A:** Budgets with alerts, Cost Anomaly Detection, mandatory tags, SCPs denying unused regions and expensive instance families in sandboxes, scheduled shutdown of dev resources.

2.8.2 Mapping and certification

AWS	Azure	GCP
Region / AZ	Region / Availability Zone	Region / Zone
Well-Architected Framework	Azure Well-Architected Framework (5 pillars)	Google Cloud Architecture Framework
Savings Plans / RIs	Azure Savings Plan / Reservations	Committed Use Discounts
Spot Instances	Spot VMs	Spot VMs (formerly Preemptible)
Budgets / Cost Explorer	Cost Management + Budgets	Billing budgets / Reports
AWS CLI / CloudShell	Azure CLI (`az`) / Cloud Shell	gcloud CLI / Cloud Shell

Certification: Cloud Practitioner tests global infrastructure, shared responsibility, pricing models and Well-Architected pillars heavily. SAA-C03 tests cost-optimised and resilient design choices. SysOps tests quotas, CLI and billing alarms.

✅ **CHECK YOUR PROGRESS 2.8**

1. **What is the difference between a service quota and API throttling?**
   Answer: A quota limits how many resources you can have (for example vCPUs); throttling limits how fast you can call APIs.
2. **GCP equivalent of Savings Plans?**
   Answer: Committed Use Discounts.

What Have We Learnt?

- Regions are isolated geographic areas; AZs are isolated data centre groups inside a region.
- Edge locations serve CloudFront and Route 53; Local Zones bring compute closer to big cities.
- Choose a region by compliance, latency, service availability and price.
- AWS secures "of" the cloud; you secure "in" the cloud. You always own IAM and data.
- Six Well-Architected pillars: Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization, Sustainability.
- On-Demand for flexibility, Savings Plans/RIs for steady use, Spot for interruptible work.
- Secure a new account: root MFA, no root keys, budget alerts, admin via Identity Center.
- AWS CLI v2 with SSO profiles gives short-lived credentials; `aws sts get-caller-identity` confirms who you are.
- IaC beats console clicks for anything that must be repeated or reviewed.
- Service quotas and API throttling are common causes of failures; check Service Quotas.

Key terms: Region, Availability Zone, AZ ID, edge location, Local Zone, Outposts, shared responsibility model, Well-Architected Framework, On-Demand, Reserved Instance, Savings Plan, Spot Instance, Free Tier, AWS Budgets, Cost Explorer, root user, MFA, IAM Identity Center, AWS CLI v2, named profile, CloudShell, service quota, throttling.

🔑 **RESUME LINE**

"Moved 3 steady production workloads to Compute Savings Plans and CI runners to Spot, cutting monthly EC2 spend by about 38 percent with no availability impact."
"Set up 12 AWS accounts with root MFA, Identity Center SSO, budget alerts and region-deny guardrails; eliminated all long-lived IAM user keys."

Exercises

A. Multiple Choice Questions

1. Which is a global AWS service?
   a. EC2 b. RDS
   c. IAM d. EBS
2. Who patches the guest OS on EC2?
   a. AWS b. Customer
   c. Both equally d. Nobody, it is automatic
3. Which pillar was added to Well-Architected in 2021?
   a. Reliability b. Cost Optimization
   c. Sustainability d. Security
4. A nightly batch job can be restarted if interrupted. Cheapest option?
   a. On-Demand b. Dedicated Host
   c. Spot d. Standard RI
5. Which command confirms the identity used by the CLI?
   a. aws iam whoami b. aws sts get-caller-identity
   c. aws configure show d. aws sso status
6. VcpuLimitExceeded means:
   a. Wrong IAM permissions b. Instance quota for vCPUs reached in that region
   c. AWS has no capacity anywhere d. Your laptop has too few cores

B. Short Answer Questions

1. Why should you deploy across at least two AZs?
2. Explain the difference between Compute Savings Plans and EC2 Instance Savings Plans.
3. List five steps to secure a new AWS account.
4. Why should you prefer SSO profiles over access keys for the CLI?

C. Scenario Questions

1. Your colleague says "all my EC2 instances are gone" after logging in this morning. What do you check?
2. A sandbox account bill jumped from USD 20 to USD 900. What do you do in the first hour?
3. A deployment script fails with `ThrottlingException` on `DescribeInstances` during a big rollout. How do you fix it?

D. Mini Project

Build a "safe account baseline" repo: a Bash or Python script that (1) prints the caller identity, (2) checks root MFA status using `aws iam get-account-summary` (`AccountMFAEnabled`), (3) lists budgets, (4) lists running instances across all enabled regions, and (5) prints the EC2 vCPU quota per region. Output a short report.

Answer Key: Chapter 2

A. MCQ

1. (c) IAM is global; the others are regional or zonal.
2. (b) EC2 is IaaS; the customer owns the guest OS.
3. (c) Sustainability was added in December 2021.
4. (c) Spot is cheapest for interruptible work.
5. (b) STS GetCallerIdentity returns account, user ID and ARN.
6. (b) It is a regional vCPU quota error.

B. Short Answers

1. AZs fail independently. If one AZ has a power or network failure, instances in the other AZ keep serving traffic behind the load balancer.
2. Compute Savings Plans apply across any EC2 family, size, region, OS, and also Fargate and Lambda, with a smaller discount. EC2 Instance Savings Plans lock to one instance family in one region for a bigger discount, but still flexible on size and OS within that family.
3. Enable root MFA; remove root access keys; create budgets and alerts; create admin identity via Identity Center (or IAM with MFA); stop using root; confirm CloudTrail.
4. SSO gives short-lived credentials that expire automatically, uses central identity with MFA, and leaves no long-lived secrets on disk to leak.

C. Scenario Questions

1. 1) Check the console region selector and the CLI default region. 2) Run `aws sts get-caller-identity` to confirm the right account. 3) Search with Resource Explorer or Tag Editor across regions. 4) If truly gone, check CloudTrail for `TerminateInstances` events and who made them. 5) Check Auto Scaling activity history in case the group replaced instances.
2. 1) Cost Explorer grouped by service and region, daily granularity. 2) Identify unknown resources (often EC2 GPU instances in unused regions). 3) CloudTrail to find the principal and access key. 4) Deactivate the key, remove the attacker's resources in all regions, rotate credentials. 5) Contact AWS Support. 6) Add an SCP to deny unused regions, enable GuardDuty, and stronger budget alerts.
3. Add retries with exponential backoff and jitter (set `AWS_RETRY_MODE=adaptive` and `AWS_MAX_ATTEMPTS`), batch calls using filters instead of per-instance calls, cache results, and slow the rollout concurrency.

D. Mini Project: Model Solution

Files: `baseline.sh` loops `for r in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text)` and runs `aws ec2 describe-instances --region $r --filters Name=instance-state-name,Values=running --query "Reservations[].Instances[].InstanceId" --output text`; uses `aws iam get-account-summary --query "SummaryMap.AccountMFAEnabled"`; `aws budgets describe-budgets --account-id $(aws sts get-caller-identity --query Account --output text)`; and `aws service-quotas get-service-quota --service-code ec2 --quota-code L-1216C47A --region $r`. `README.md` explains

required read-only permissions (for example the `ReadOnlyAccess` managed policy plus `budgets:ViewBudget`) and sample output.