



Build Roles: Cloud Engineer, Architect, Developer

Careers chapter | Daily work, skills, tools, certifications, portfolio projects and interview questions with model answers for cloud engineers, solutions architects and cloud developers

Careers Cloud engineer Solutions architect Developer SAA-C03 DVA-C02

The three roles

Cloud engineer

ENTRY → SENIOR

Builds and keeps the AWS foundation running: accounts, VPCs, compute, IAM, backups, as code

A DAY IN THE ROLE

- Write Terraform or CloudFormation for a new environment
- Review a pull request that opens a security group
- Debug why an instance cannot reach S3
- Patch AMIs and rotate secrets

KEY SKILLS

Linux

VPC and DNS

IAM

IaC

scripting (Bash, Python)

TYPICAL TOOLS

Terraform

CloudFormation/CDK

AWS CLI

GitHub Actions

CloudWatch

FROM sysadmin, network admin, cloud support

NEXT Senior / platform engineer, Solutions architect, DevOps engineer

Solutions architect

MID → PRINCIPAL

Turns requirements into designs that are secure, reliable, performant and affordable, and gets them adopted

A DAY IN THE ROLE

- Workshop with a product team on requirements
- Draw a target architecture and cost estimate
- Write an ADR comparing Aurora and DynamoDB
- Run a Well-Architected review

KEY SKILLS

broad AWS service knowledge

networking and security

cost modelling

writing and whiteboarding

stakeholder management

TYPICAL TOOLS

draw.io / Lucid

Pricing Calculator

Well-Architected Tool

ADRs

FROM senior cloud engineer or developer

NEXT Principal / enterprise architect, Head of cloud, Pre-sales SA at a vendor or partner

Cloud developer

ENTRY → SENIOR

Ships features as cloud-native code: APIs, functions, events, data access, with tests and pipelines

A DAY IN THE ROLE

- Build a Lambda + API Gateway endpoint
- Tune a DynamoDB access pattern
- Add tracing to find a slow call
- Fix a failing pipeline stage

KEY SKILLS

Python, TypeScript, Java or Go

AWS SDKs

serverless and containers

testing

IAM for apps

TYPICAL TOOLS

SAM / CDK

boto3, AWS SDK v3

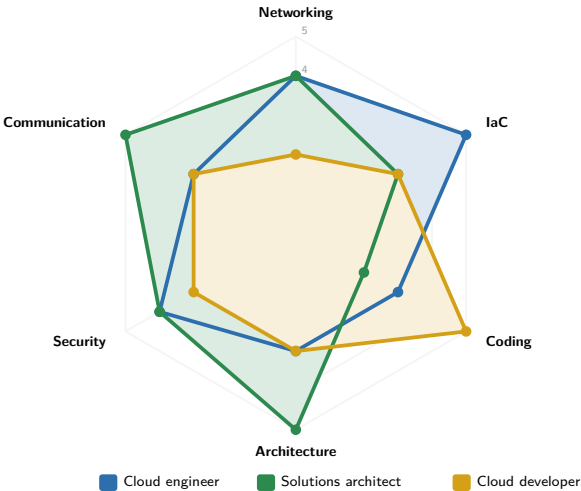
X-Ray / OpenTelemetry

CodeBuild or GitHub Actions

FROM software developer

NEXT Senior developer, DevOps / platform, GenAI engineer

Skill profile (1 = aware, 5 = expert, typical mid-level)



Certifications and portfolio

Role	Certifications	Portfolio project that proves it
Cloud engineer	CLF → SAA → SOA → SAP or DOP; Terraform Associate (HashiCorp)	Two-AZ VPC + ALB + Auto Scaling + RDS in Terraform, remote state in S3, CI plan/apply, README with diagram and monthly cost
Solutions architect	SAA → SAP; optional ANS or SCS	A design pack: diagram, ADRs (3), Well-Architected self-review, cost estimate, DR plan with RTO/RPO for a fictional company
Cloud developer	DVA → DOP or AIP	Serverless URL shortener or image service: API Gateway, Lambda, DynamoDB, S3, Cognito, SAM/CDK, tests, X-Ray traces

- What hiring managers check in a repo it deploys from scratch with one command; no secrets committed; IAM least privilege; a diagram; a teardown command; cost noted
- What SA candidates must show trade-offs written down (why not the alternative), numbers (latency, cost, RTO), a clear one-page summary

- What developers must show tests in CI, idempotent handlers, retries with backoff, structured logs, a README a stranger can follow

Interview classic: size a web tier

PROBLEM Peak load is $R = 2,000$ requests/s. One instance serves $c = 250$ req/s at 100 % CPU and you run at $u = 60$ % target utilisation. How many instances, across 3 AZs, so the site survives the loss of one AZ?

GIVEN

- $R = 2,000$ req/s peak
- $c = 250$ req/s per instance at full load
- $u = 0.6$ target utilisation
- $z = 3$ AZs; must survive 1 AZ down

1 Capacity per instance at target

$$c \times u = 250 \times 0.6 = 150 \text{ req/s}$$

Why: headroom for spikes and slow requests

2 Instances needed at peak

$$N = \text{ceil}(R / 150) = \text{ceil}(13.3) = 14$$

Why: round up, never down

3 Survive one AZ down

14 must fit in 2 AZs $\rightarrow$ 7 per AZ

Why: remaining AZs carry the full load

4 Total across 3 AZs

$$3 \times 7 = 21 \text{ instances}$$

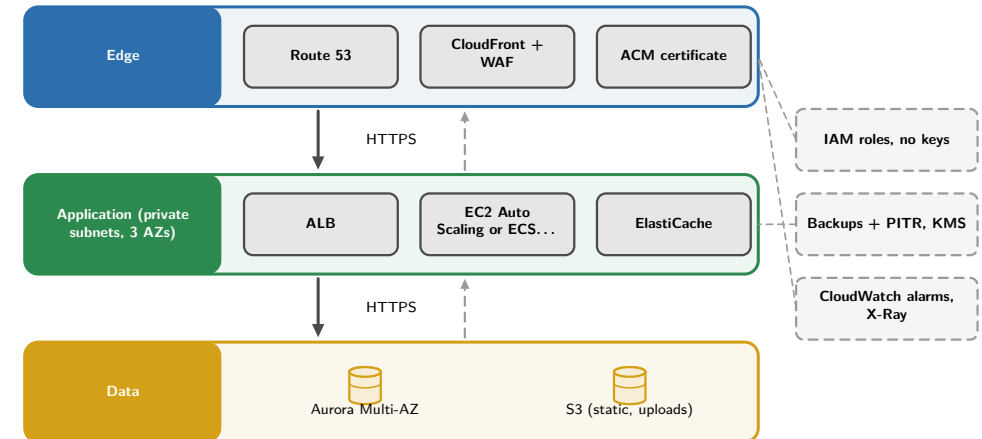
Why: 50 % over the bare minimum, the price of AZ resilience

ANSWER 21 instances, 7 per AZ, in an Auto Scaling group behind an ALB (minimum 21 at peak hours)

CHECK With 2 AZs you would need 14 per AZ = 28: three AZs are cheaper for N+1 resilience.

Interview questions with model answers

The whiteboard answer to 'design a web app on AWS'



Q1 Security group vs network ACL?

A SG: stateful, instance/ENI level, allow rules only, all rules evaluated. NACL: stateless, subnet level, allow and deny, rules in number order; return traffic needs its own rule (ephemeral ports 1024-65535).

Q2 An instance in a private subnet cannot download packages. Why?

A No route to the internet: add a NAT gateway in a public subnet and a 0.0.0.0/0 route to it; or use VPC endpoints / a package mirror. Also check SG egress and NACL return ports.

Q3 How do you give an app on EC2 access to S3 without keys?

A Attach an IAM role through an instance profile with a least-privilege policy on the bucket ARN; the SDK picks credentials from IMDSv2.

Q4 SQS standard vs FIFO?

A Standard: near-unlimited throughput, at-least-once, best-effort order. FIFO: exactly-once processing and order per message group, 300 msg/s per API action (3,000 with batching; higher in high-throughput mode).

Q5 RDS Multi-AZ vs read replica?

A Multi-AZ: synchronous standby for availability, automatic failover, not readable (classic). Read replica: asynchronous copy for read scaling or DR, promotable manually.

Q6 How would you cut this architecture's cost by 30 %?

A Measure first (Cost Explorer by service and tag), then rightsize (Compute Optimizer), Savings Plans for steady compute, Graviton, S3 lifecycle to cheaper classes, remove idle resources (NAT, unattached EBS, old snapshots), cache with CloudFront.

Q7 Lambda times out calling a database. Causes?

A Function in a VPC without route to the DB, SG not allowing the port, connection exhaustion (use RDS Proxy), timeout lower than query time, cold start plus slow DNS. Check CloudWatch Logs and X-Ray.

Q8 Behavioural: tell me about a design you got wrong.

A STAR: the situation and constraint, your decision, what failed (with a metric), what you changed, what you now do differently (an ADR, a load test, a review).

How SA interviews are scored

Interviewers look for: clarifying questions first (users, traffic, data size, RTO/RPO, budget), a simple baseline before optimising, explicit trade-offs, numbers, failure modes (what if an AZ dies, the DB is slow, a key leaks), and cost. Naming 20 services without reasons scores lower than 6 services well justified.

Recap and self-test

MYTH OR FACT?

“Architects no longer touch code or the console”

MYTH: Myth: credible architects prototype and read IaC
Most SA interviews include hands-on or deep technical rounds.

MYTH OR FACT?

“A developer does not need to know IAM or networking”

MYTH: Myth: most production bugs in cloud apps are permissions, timeouts and networking
AccessDenied and VPC routing are a developer's daily debugging.

MYTH OR FACT?

“Cloud engineers just click in the console”

MYTH: Myth: the job is infrastructure as code with reviews and pipelines
Console changes create drift; interviewers ask how you prevent it.

MYTH OR FACT?

“More services in the answer means a better design”

MYTH: Myth: the simplest design meeting the requirements wins
Every extra service is cost, failure modes and on-call load.

Tick a box only when you could explain the point to someone else



Engineer builds the foundation as code

SAA, SOA, Terraform



Architect decides and explains trade-offs

SAA → SAP, ADRs, cost



Developer ships cloud-native code

DVA, SDKs, serverless, CI/CD



Answer with numbers

capacity, latency, RTO, monthly cost



One deployable project per role

README, diagram, teardown

Q1 First thing to do in a system design interview?

A Ask clarifying questions: users, traffic, data size, latency, availability (RTO/RPO), budget, compliance.

Q2 Which certification pair is the classic architect path?

A Solutions Architect Associate SAA-C03 then Professional SAP (SAP-C02 until 2026-11-16, then SAP-C03).

Q3 Peak 900 req/s, 100 req/s per instance at target, survive one of 3 AZs down?

A 9 needed at peak → must fit in 2 AZs: 5 per AZ (ceil 4.5) → 15 instances.

Q4 Why idempotent message handlers?

A Queues and retries deliver at least once; reprocessing must not double-charge or double-write.

Q5 What is an ADR?

A Architecture decision record: context, options, decision, consequences, in one short document.

Q6 Name three things that make a portfolio repo credible.

A One-command deploy and teardown, a diagram with a cost note, no secrets and least-privilege IAM.

Q7 Developer's tool to find which downstream call is slow?

A Distributed tracing: AWS X-Ray or OpenTelemetry (ADOT) traces.

Q8 Typical next roles after cloud engineer?

A Senior or platform engineer, DevOps engineer, solutions architect.