■ Amazon EC2

Elastic Compute Cloud

Complete Enterprise Guide

Interview Ready • Production SOP • Architecture • Costing

GUI Steps • POC • GitHub README • Benefits & Features

■ INTERVIEW	■ PRODUCTION	■ COSTING	■ ARCHITECTURE

■ Table of Contents

1. What is Amazon EC2?

■ Interview Explanation

Amazon EC2 (Elastic Compute Cloud) is a web service that provides resizable compute capacity in the cloud. Think of it as renting a virtual computer in Amazon's data center where you have full root/admin access, can install any OS, and pay only for what you use.

In simple terms: Instead of buying a physical server (which costs $5,000–$50,000+), you launch a virtual machine on AWS in under 60 seconds, use it, and pay per second.

Key Interview Points:

- EC2 = Infrastructure as a Service (IaaS) — you manage the OS and above.
- Launched in 2006, EC2 was one of AWS's first and most foundational services.
- Supports Linux, Windows, macOS operating systems.
- Integrates with 200+ AWS services: S3, RDS, VPC, IAM, CloudWatch, etc.
- Instances are backed by physical hardware in AWS Availability Zones.
- You control: instance type, AMI, storage, networking, security groups.
- Auto Scaling allows horizontal scaling based on demand automatically.

The Analogy That Always Works in Interviews:

"EC2 is like renting an apartment instead of buying a house. You get a fully functional living space, you can customize the interior (OS, software), you pay monthly rent (per-hour/second), and you can upgrade to a bigger apartment or move out anytime without long-term commitment."

2. Core Concepts & Architecture

■ AMI — Amazon Machine Image

A template containing the OS, pre-installed software, and configuration. Like a 'snapshot blueprint' for your instance. You can use AWS-provided AMIs, Marketplace AMIs, or create your own custom AMIs.

■ Instance Type — Hardware Profile

Defines the CPU, RAM, network, and storage performance. Examples: t3.micro (1 vCPU, 1 GB RAM), m5.xlarge (4 vCPU, 16 GB RAM), c5.2xlarge (8 vCPU, 16 GB RAM).

■ EBS — Elastic Block Store

Persistent block storage that attaches to EC2 like a hard drive. Survives instance stop/start. Types: gp3 (general purpose SSD), io2 (high IOPS), st1 (throughput HDD).

■ Security Group — Virtual Firewall

Controls inbound/outbound traffic at the instance level using rules. Stateful — return traffic is automatically allowed.

■ Key Pair — SSH Authentication

RSA or ED25519 key pair for secure SSH access to Linux instances or RDP password decryption for Windows instances.

■ VPC — Virtual Private Cloud

Your isolated network within AWS. EC2 instances run inside subnets within a VPC. You control IP ranges, routing tables, and internet gateways.

■ Elastic IP — Static Public IP

A static IPv4 address you can attach/detach from instances. Unlike regular public IPs, Elastic IPs persist through instance stop/start.

■ User Data — Bootstrap Script

A script that runs automatically when the instance first launches. Used to install software, configure settings, pull code from S3/Git.

■ IAM Role — EC2 Permissions

Attach an IAM role to grant EC2 permissions to AWS services (S3, DynamoDB, etc.) without hardcoding credentials.

■ Placement Group — Hardware Placement Strategy

Controls how instances are placed on hardware. Cluster (low latency), Spread (high availability), Partition (large distributed systems).

3. EC2 Instance Type Families

AWS groups EC2 instances into families based on workload optimization. Understanding these is critical for both the AWS exam and real-world architecture decisions.

Family	Prefix	Best For	Example
General Purpose	t, m	Web servers, dev/test, small DBs	t3.micro, m6i.large
Compute Optimized	c	HPC, batch processing, gaming servers	c5.2xlarge, c6g.4xlarge
Memory Optimized	r, x, z	In-memory DBs, SAP HANA, Redis	r6i.4xlarge, x1e.32xlarge
Storage Optimized	i, d, h	Data warehouses, Hadoop, NoSQL	i3.8xlarge, d3en.xlarge
Accelerated	p, g, f	ML training, GPU, FPGA, inference	p4d.24xlarge, g5.xlarge
Network Optimized	c5n, r5n	100 Gbps networking, HPC clusters	c5n.18xlarge

How to Read Instance Names:

Format: **[Family][Generation][Attribute].[Size]** → Example: **c6gn.xlarge**

Part	Value	Meaning
Family	c	Compute Optimized
Generation	6	6th generation hardware
Attribute	g	Graviton (ARM) processor
Attribute	n	Network optimized
Size	xlarge	4 vCPU, 8 GB RAM

4. EC2 Pricing & Costing

On-Demand	No commitment
Pay per second (Linux) or per hour (Windows). Most expensive. No upfront cost.	
■ *Use when: Dev/test, unpredictable traffic, short-term projects.*	

Reserved (RI)	1 or 3 year term
Up to 72% savings vs On-Demand. Pay all-upfront, partial, or no upfront.	
■ *Use when: Steady-state production workloads, baseline capacity.*	

Savings Plans	1 or 3 year term
Up to 66% savings. More flexible than RIs — applies across instance types/regions.	
■ *Use when: Flexible workloads, mixed instance families.*	

Spot Instances	Market pricing
Up to 90% savings. AWS can reclaim with 2-min notice. Fault-tolerant only.	
■ *Use when: Batch jobs, data analysis, CI/CD, ML training.*	

Dedicated Hosts	Per host billing
Physical server for your exclusive use. Most expensive. For compliance/licensing.	
■ *Use when: BYOL software licenses, regulatory compliance.*	

Dedicated Instances	Per instance
Runs on hardware dedicated to you but shared with other instances in your account.	
■ *Use when: Compliance requirements without full host billing.*	

■ Real-World Cost Example (us-east-1, 2024):

Instance	vCPU	RAM	On-Demand/hr	Monthly (On-Dem)	1-yr RI/hr	Monthly (RI)	Spot/hr (est)
t3.micro	2	1 GB	$0.0104	$7.49	$0.0066	$4.75	$0.003

t3.medium	2	4 GB	$0.0416	$29.95	$0.0264	$19.01	$0.013
m5.large	2	8 GB	$0.096	$69.12	$0.063	$45.36	$0.029
m5.xlarge	4	16 GB	$0.192	$138.24	$0.126	$90.72	$0.058
c5.2xlarge	8	16 GB	$0.34	$244.80	$0.224	$161.28	$0.102
r5.4xlarge	16	128 GB	$1.008	$725.76	$0.672	$483.84	$0.302

Note: Prices vary by region and change over time. Always check the AWS Pricing Calculator at calculator.aws for the most accurate estimates. Data transfer costs are additional.

5. How to Create an EC2 Instance — GUI (AWS Console)

Complete step-by-step walkthrough of launching an EC2 instance using the AWS Management Console. Follow these steps exactly for a successful deployment.

STEP 1

Sign in to AWS Console

Navigate to https://console.aws.amazon.com → Sign in with your AWS account credentials → Select your preferred Region from the top-right dropdown (e.g., us-east-1 N. Virginia).

STEP 2

Open EC2 Dashboard

In the top search bar, type 'EC2' → Click 'EC2' under Services → You land on the EC2 Dashboard → Click the orange button 'Launch instance'.

STEP 3

Name Your Instance

Under 'Name and tags', enter a descriptive name such as 'prod-web-server-01' → Optionally add tags: Key=Environment, Value=Production | Key=Team, Value=Backend.

STEP 4

Choose an AMI (OS)

Under 'Application and OS Images (Amazon Machine Image)', browse or search → Common choices: Amazon Linux 2023 (free tier), Ubuntu 22.04 LTS, Windows Server 2022 → Click 'Select' on your chosen AMI → Architecture: x86_64 or arm64 (Graviton, cheaper).

STEP 5

Choose Instance Type

Under 'Instance type', click the dropdown → Type to filter e.g. 't3.micro' → Free tier: t2.micro or t3.micro → For production: m5.large or c5.xlarge → Review the vCPU and memory shown on screen.

STEP 6

Create or Select Key Pair

Under 'Key pair (login)', click 'Create new key pair' → Name it (e.g., 'my-ec2-key') → Type: RSA | Format: .pem (Linux/Mac) or .ppk (PuTTY/Windows) → Click 'Create key pair' → Browser downloads the private key — SAVE THIS SECURELY, it cannot be re-downloaded!

STEP 7

Configure Network Settings (VPC/Subnet/SG)

VPC: Select your VPC (or default VPC for testing) → Subnet: Select a subnet in your preferred AZ → Auto-assign Public IP: Enable (for internet access) → Firewall (Security Groups): 'Create security group' → Add rules: SSH (port 22) from My IP, HTTP (port 80) from Anywhere, HTTPS (port 443) from Anywhere.

STEP 8

Configure Storage (EBS)

Root volume: Default 8 GB gp3 (Amazon Linux) or 30 GB (Windows) → Click 'Add new volume' to add extra EBS volumes → Volume type: gp3 (recommended, 3000 IOPS baseline, cheaper than gp2) → Enable 'Delete on termination' checkbox based on your data retention needs → Enable EBS encryption for production workloads.

STEP 9

Advanced Details (Optional but Important)

IAM Instance Profile: Attach a role (e.g., S3ReadOnlyAccess) → User data: Paste bootstrap script (see below) → Tenancy: Shared (default) or Dedicated → Monitoring: Enable detailed CloudWatch monitoring (1-min intervals, extra cost).

STEP 10

Review & Launch

Review the Summary panel on the right showing all your selections → Number of instances: 1 (or more for batch launch) → Click 'Launch instance' (orange button) → AWS shows 'Successfully initiated launch' → Click 'View all instances'.

STEP 11

Connect to Your Instance

In EC2 Dashboard → Instances → Select your instance → Wait for 'Instance state: running' and 'Status check: 2/2 checks passed' → Click 'Connect' button → SSH Client tab: copy the ssh command → In your terminal: chmod 400 my-ec2-key.pem → ssh -i my-ec2-key.pem ec2-user@

User Data Bootstrap Script Example:

```
#!/bin/bash yum update -y yum install -y httpd systemctl start httpd systemctl enable
httpd echo '<h1>Hello from EC2!</h1>' > /var/www/html/index.html
```

6. How EC2 Works — Under the Hood

The Nitro Hypervisor

AWS uses its custom-built Nitro System — a combination of dedicated hardware and lightweight hypervisor that offloads storage and networking virtualization to dedicated hardware. This means nearly all server resources go to your instance with near bare-metal performance.

Layer	Component	What Happens
Physical	AWS Data Center	Your instance runs on a physical server in an AWS AZ. Hardware is maintained 24/
Hypervisor	Nitro / Xen	Nitro System virtualizes CPU, memory, storage. KVM-based for C5, M5+ instances.
Instance	Your EC2 VM	Gets dedicated vCPUs (hyperthread cores), memory pages, and network bandwidth
Networking	ENA / EFA	Elastic Network Adapter provides up to 100 Gbps. EFA for HPC/ML workloads.
Storage	EBS / Instance Store	EBS = network-attached persistent SSD. Instance Store = local NVMe SSD (ephem
Security	Nitro Security Chip	Hardware root of trust. Validates firmware. Prevents side-channel attacks.

EC2 Lifecycle States:

State	Description
Pending	Instance is being provisioned. Not billed.
Running	Instance is active. Billing starts. Ready for use.
Stopping	Instance is preparing to stop. Not billed (transitional).
Stopped	Instance is off. EBS charges apply. No compute charges.
Rebooting	OS-level restart. Stays on same host. Billing continues.
Shutting-down	Termination in progress. Not billed.
Terminated	Instance is permanently deleted. Cannot be recovered.

7. Benefits & Features

■ **Elasticity** Scale from 1 to 1000+ instances in minutes. Auto Scaling adjusts capacity automatically based on CPU, memory, custom metrics, or schedules.	■ **Global Reach** Deploy in 33+ AWS Regions and 105+ Availability Zones worldwide. Put compute close to your users for low latency.
■ **Security** VPC isolation, Security Groups, NACLs, IAM roles, KMS encryption, AWS Shield DDoS protection, and Nitro hardware security.	■ **Cost Efficiency** Multiple pricing models (On-Demand, Reserved, Spot, Savings Plans). Spot instances save up to 90%. Pay-per-second billing.
■ **Flexibility** 600+ instance types. Any OS. Full root/admin access. Custom AMIs. Bring your own licenses (BYOL).	■ **Monitoring** CloudWatch integration: CPU, network, disk metrics. Detailed monitoring (1-min). CloudTrail for audit logging.
■ **High Availability** Multi-AZ deployment. Elastic Load Balancing across instances. Auto Scaling replaces unhealthy instances automatically.	■■ **Storage Options** EBS (persistent SSD/HDD), Instance Store (local NVMe), EFS (shared NFS), S3 (object storage via API), FSx (managed file systems).
■ **Networking** Up to 100 Gbps network bandwidth. ENA for enhanced networking. EFA for HPC. VPC peering, Transit Gateway, PrivateLink.	■ **Automation** Launch templates, CloudFormation stacks, AWS Systems Manager, EC2 Image Builder for automated AMI pipelines.

8. Enterprise Architecture Diagram

Production 3-Tier Web Architecture on AWS EC2:

```
INTERNET
|
[Route 53 DNS]
|
[CloudFront CDN] ← S3 Static Assets
|
[AWS WAF + Shield]
|
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ PUBLIC SUBNET (Multi-AZ) ■
■ [Application Load Balancer] ■
■ AZ-1: 443/80 → EC2 Web | AZ-2: 443/80→EC2 ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
|
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ PRIVATE SUBNET - App Tier ■
■ Auto Scaling Group (min:2 max:10) ■
■ [EC2 App Server AZ-1] [EC2 App Server AZ-2] ■
■ IAM Role: S3Read, DynamoDB, Secrets Manager ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
|
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ PRIVATE SUBNET - Data Tier ■
■ [RDS Aurora Multi-AZ] [ElastiCache Redis] ■
■ Primary (AZ-1) ←→ Replica (AZ-2) ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
|
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ MANAGEMENT PLANE ■
■ CloudWatch Logs | CloudTrail | Config ■
■ Systems Manager | Secrets Manager | KMS ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
|
[NAT Gateway] → Internet (for outbound only)
[VPC Flow Logs] → S3 → Athena (security analysis)
```

Architecture Components Explained:

Component	Purpose
Route 53	DNS routing with health checks and failover. Latency-based routing to nearest region.
CloudFront	CDN for caching static content globally. Reduces load on EC2 by 60-80%.

WAF + Shield	Web Application Firewall blocks SQLi, XSS. Shield Standard = free DDoS protection.
Application Load Balancer	Layer 7 load balancing. SSL termination. Path-based and host-based routing.
Auto Scaling Group	Maintains desired instance count. Replaces unhealthy instances. Scales on metrics.
EC2 in Private Subnet	No direct internet access. Accessed via ALB. Uses NAT Gateway for outbound.
RDS Aurora Multi-AZ	Managed database with automatic failover. Read replicas for performance.
ElastiCache Redis	In-memory cache. Reduces DB load. Session storage. Sub-millisecond latency.
NAT Gateway	Allows private subnet instances to reach internet (updates, APIs) without public IP.
Systems Manager	Patch management, Run Command, Session Manager SSH (no port 22 needed).

9. Proof of Concept (POC) — 3-Tier Web App

This POC deploys a complete 3-tier web application on EC2 in under 30 minutes. It demonstrates ALB, Auto Scaling, and RDS integration.

Phase 1: VPC & Network Setup

```
# Create VPC aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications \
'ResourceType=vpc,Tags=[{Key=Name,Value=poc-vpc}]' # Create public subnets (2 AZs) aws
ec2 create-subnet --vpc-id vpc-xxxx --cidr-block 10.0.1.0/24 \ --availability-zone
us-east-1a --tag-specifications \
'ResourceType=subnet,Tags=[{Key=Name,Value=public-1a}]' aws ec2 create-subnet --vpc-id
vpc-xxxx --cidr-block 10.0.2.0/24 \ --availability-zone us-east-1b # Create private
subnets aws ec2 create-subnet --vpc-id vpc-xxxx --cidr-block 10.0.10.0/24 \
--availability-zone us-east-1a # Internet Gateway aws ec2 create-internet-gateway aws
ec2 attach-internet-gateway --vpc-id vpc-xxxx --internet-gateway-id igw-xxxx
```

Phase 2: Launch EC2 App Servers

```
# Create Launch Template aws ec2 create-launch-template \ --launch-template-name
poc-web-lt \ --version-description 'v1' \ --launch-template-data '{ "ImageId":
"ami-0c02fb55956c7d316", "InstanceType": "t3.small", "SecurityGroupIds": ["sg-xxxx"],
"IamInstanceProfile": {"Name": "EC2-S3-Role"}, "UserData": "base64encodedscript" }' #
Create Auto Scaling Group aws autoscaling create-auto-scaling-group \
--auto-scaling-group-name poc-asg \ --launch-template 'LaunchTemplateName=poc-web-lt' \
--min-size 2 --max-size 6 --desired-capacity 2 \ --vpc-zone-identifier
'subnet-xxxx,subnet-yyyy' \ --target-group-arns arn:aws:elasticloadbalancing:...
```

Phase 3: User Data Script (runs on instance launch)

```
#!/bin/bash set -e exec > /var/log/userdata.log 2>&1 # Update system yum update -y #
Install Node.js and app dependencies curl -sL https://rpm.nodesource.com/setup_18.x |
bash - yum install -y nodejs git # Pull application code git clone
https://github.com/your-org/your-app.git /app cd /app && npm install --production # Set
environment variables from SSM Parameter Store export DB_HOST=$(aws ssm get-parameter
--name /poc/db/host --query 'Parameter.Value' --output text) export DB_PASS=$(aws ssm
get-parameter --name /poc/db/pass --with-decryption --query 'Parameter.Value' --output
text) # Start application with PM2 npm install -g pm2 pm2 start app.js --name 'web-app'
pm2 startup && pm2 save # Health check endpoint must return 200 curl -f
http://localhost:3000/health
```

Phase 4: Validation Checklist

- ■ ALB DNS resolves and returns HTTP 200
- ■ Auto Scaling Group shows 2 healthy instances
- ■ Target Group health checks passing
- ■ EC2 instances can reach RDS on port 3306
- ■ CloudWatch logs showing application output
- ■ IAM role permissions validated (no hardcoded credentials)

- ■ Security Groups restrict traffic correctly
- ■ EBS volumes encrypted

10. Standard Operating Procedures (SOP)

■ SOP-EC2-001: Launch New Production Instance

```
1. Request approval via JIRA ticket
2. Use approved AMI from EC2 Image Builder
3. Apply mandatory tags: Env, Team, CostCenter, Owner
4. Use Launch Template (never manual config)
5. Attach IAM role (principle of least privilege)
6. Enable EBS encryption with KMS CMK
7. Add to appropriate Auto Scaling Group
8. Verify CloudWatch alarms configured
9. Run compliance scan (AWS Config rules)
10. Document in CMDB
```

■ SOP-EC2-002: Stop/Start Instance Safely

```
STOP:
1. Notify stakeholders via Slack/PagerDuty
2. Drain connections from Load Balancer target group
3. Wait for in-flight requests to complete (60s)
4. Take EBS snapshot before stopping
5. Stop via Console or: aws ec2 stop-instances --instance-ids i-xxxx
6. Verify state is 'stopped'

START:
1. Verify EBS volumes attached
2. Start instance: aws ec2 start-instances --instance-ids i-xxxx
3. Wait for status checks (2/2)
4. Verify application health check
5. Re-register with Load Balancer
```

■ SOP-EC2-003: Emergency Patching Procedure

```
1. Identify affected instances via AWS Systems Manager Patch Manager
2. Create snapshot of all EBS volumes
3. Apply patches via SSM Run Command:
aws ssm send-command --document-name AWS-RunPatchBaseline
4. Monitor patch compliance in SSM Compliance dashboard
5. Reboot if required (kernel patches)
6. Verify application functionality post-patch
7. Update patch baseline record
8. Close security ticket with evidence
```

■ SOP-EC2-004: Instance Right-Sizing Review

```
Monthly cadence:
1. Run AWS Compute Optimizer recommendations
2. Review CloudWatch CPU/Memory/Network metrics (last 30 days)
3. Identify instances <20% CPU for 7+ days (over-provisioned)
4. Identify instances >80% CPU regularly (under-provisioned)
5. Schedule maintenance window for resizing
6. Stop instance → Change instance type → Start
7. Validate performance post-change
8. Update cost tracking documentation
```

■ SOP-EC2-005: Terminate Instance

```
1. Confirm no active sessions (SSM Session Manager)
2. Verify data backed up (EBS snapshot or AMI)
3. Remove from load balancer target group
4. Remove from Auto Scaling Group if applicable
5. Release Elastic IP if attached
6. Terminate: aws ec2 terminate-instances --instance-ids i-xxxx
7. Verify EBS volumes deleted (if delete-on-termination enabled)
8. Remove DNS records
9. Archive logs to S3 Glacier
10. Update CMDB and cost allocation tags
```

11. GitHub README (Professional Format)

Copy this as your project's README.md:

~~~
# ■ AWS EC2 Production Deployment

![AWS](https://img.shields.io/badge/AWS-EC2-orange?logo=amazonaws)
![License](https://img.shields.io/badge/license-MIT-blue)
![Terraform](https://img.shields.io/badge/IaC-Terraform-purple)

Enterprise-grade EC2 deployment with Auto Scaling, ALB, and RDS.

## ■ Prerequisites

- AWS CLI configured (`aws configure`)
- Terraform >= 1.5.0
- IAM permissions: EC2FullAccess, VPCFullAccess, IAMReadOnly

## ■■ Architecture

Internet → Route53 → CloudFront → WAF → ALB → EC2 ASG → RDS Aurora

## ■ Quick Start

```bash
git clone https://github.com/your-org/aws-ec2-prod.git
cd aws-ec2-prod
cp terraform.tfvars.example terraform.tfvars
# Edit tfvars with your values
terraform init
terraform plan -out=tfplan
terraform apply tfplan
```

## ■■ Configuration

| Variable         | Default        | Description              |
|-----------------|---------------|-------------------------|
| instance_type    | t3.small       | EC2 instance type        |
| min_size         | 2              | ASG minimum instances    |
| max_size         | 10             | ASG maximum instances    |
| ami_id           | ami-xxxxxxx    | Amazon Linux 2023 AMI    |
| environment      | production     | Deployment environment   |

## ■ Security

- EBS volumes encrypted with KMS CMK
- Security Groups with least-privilege rules
- IAM roles (no hardcoded credentials)
~~~

```
- SSM Session Manager (no port 22)
- AWS Config rules for compliance

## ■ Monitoring

CloudWatch dashboards, alarms at 80% CPU, and PagerDuty integration.

## ■ Estimated Cost (us-east-1)

- 2x t3.small On-Demand: ~$30/month
- ALB: ~$18/month
- RDS db.t3.medium Multi-AZ: ~$60/month
- **Total baseline: ~$110/month**

## ■ Contributing

1. Fork → Feature branch → PR → Code review → Merge

## ■ License

MIT License — see LICENSE file.
```

12. Key Takeaways & Interview Cheat Sheet

Top 20 EC2 Interview Q&A;:

Q1: What is EC2?	Elastic Compute Cloud — IaaS service providing resizable virtual machines (instances) in the cloud. Pay per second.
Q2: What is an AMI?	Amazon Machine Image — template with OS + pre-installed software. Used to launch identical instances. Region-specific.
Q3: Difference: Stop vs Terminate?	Stop: OS shutdown, EBS preserved, can restart. Terminate: permanent deletion, root EBS deleted by default.
Q4: What is a Security Group?	Virtual stateful firewall. Controls traffic by port/protocol/source. Up to 5 SGs per instance. Deny by default.
Q5: What is a Placement Group?	Cluster (same rack, low latency), Spread (different racks, HA), Partition (isolated groups for Cassandra/HDFS).
Q6: Spot vs Reserved?	Spot: 90% cheaper, can be interrupted with 2-min notice. Reserved: 72% cheaper, 1-3yr commitment, guaranteed capacity.
Q7: What is EBS?	Elastic Block Store — persistent network-attached block storage. Types: gp3, io2, st1, sc1. Snapshot to S3.
Q8: Instance Store vs EBS?	Instance Store: local NVMe, ephemeral (lost on stop), highest IOPS. EBS: persistent, network-attached, snapshots.
Q9: What is User Data?	Bootstrap script that runs once at first launch. Used to install software, configure system, start services.
Q10: What is an Elastic IP?	Static public IPv4 address. Persists across stop/start. Free when attached to running instance. Charged when idle.
Q11: How to scale EC2?	Vertical: resize instance type (stop required). Horizontal: add more instances via Auto Scaling Groups (preferred).
Q12: What is EC2 Auto Scaling?	Automatically adjusts instance count based on metrics (CPU, custom). Min/Max/Desired capacity. Multi-AZ support.

Q13: What is a Launch Template?	Versioned configuration for launching EC2 (AMI, instance type, SG, key pair). Best practice over Launch Configurations.
Q14: EC2 vs Lambda?	EC2: persistent, full control, long-running, stateful. Lambda: serverless, event-driven, max 15 min, stateless, auto-scale.
Q15: What is EC2 Hibernate?	Saves RAM to EBS, faster boot than cold start. Instance state preserved. Must enable at launch. Limited instance types.
Q16: How does billing work?	Per second (min 60s) for Linux. Per hour for Windows/commercial. EBS charged separately even when stopped.
Q17: What is Nitro?	AWS's custom hypervisor system using dedicated hardware chips for networking and storage. Provides near bare-metal performance.
Q18: What is IMDSv2?	Instance Metadata Service v2 — session-oriented, token-based. Protects against SSRF attacks. Always use v2 in production.
Q19: How to connect without SSH port 22?	AWS Systems Manager Session Manager — browser or CLI-based shell. No inbound port needed. IAM-controlled access.
Q20: Best practices?	Use IAM roles (never hardcode keys), encrypt EBS, use private subnets, enable IMDSv2, tag all resources, use SSM not SSH.

■ *PRO TIP: In interviews, always mention: Multi-AZ for HA, Auto Scaling for elasticity, IAM roles for security, EBS gp3 for cost optimization, and CloudWatch for observability. These 5 pillars demonstrate production-level thinking.*

■ YOU ARE NOW EC2 READY ■