

## 1. EC2 vs. Lambda

- **EC2:** Servidor virtual (IaaS). Você controla tudo (sistema operacional, updates, escala).
- **Lambda:** Serverless (FaaS). Executa código sob demanda, sem gerenciar servidor.

### Quando usar:

- **EC2** → Aplicações com necessidades de controle de SO, longos tempos de execução ou software legado.
  - **Lambda** → Tarefas curtas, eventos ocasionais, baixo custo e escalabilidade automática.
- 

## 2. S3 vs. EBS vs. EFS

- **S3:** Armazenamento de objetos. Ideal para backups, imagens, vídeos, logs.
- **EBS:** Armazenamento em bloco. Usado com EC2 como “HD”.
- **EFS:** Armazenamento de arquivos. Compartilhado entre várias instâncias.

### Quando usar:

- **S3** → Arquivos estáticos e acessos via URL (ex: sites estáticos).
  - **EBS** → Sistema de arquivos persistente ligado a uma EC2.
  - **EFS** → Múltiplas EC2s precisam acessar os mesmos dados como em um NFS.
- 

## 3. ELB vs. Auto Scaling

- **ELB (Elastic Load Balancer):** Distribui tráfego entre instâncias EC2.
- **Auto Scaling:** Cria ou remove instâncias EC2 automaticamente com base na demanda.

### Quando usar:

- **ELB** → Balanceamento de carga entre servidores.
  - **Auto Scaling** → Crescer ou reduzir capacidade automaticamente.
- 

## 4. RDS vs. DynamoDB vs. Aurora

- **RDS:** Banco de dados relacional (MySQL, PostgreSQL, etc.).
- **DynamoDB:** NoSQL totalmente gerenciado.
- **Aurora:** Banco relacional compatível com MySQL/PostgreSQL, altamente escalável.

### Quando usar:

- **RDS** → Aplicações tradicionais que exigem SQL.
  - **DynamoDB** → Altíssima escalabilidade, baixa latência, dados semi-estruturados.
  - **Aurora** → Melhor performance e disponibilidade que RDS padrão.
- 

## 5. SNS vs. SQS vs. EventBridge

- **SNS (Simple Notification Service):** Notificações push (pub/sub).
- **SQS (Simple Queue Service):** Fila de mensagens (decoupling entre sistemas).

- **EventBridge**: Orquestração de eventos entre serviços AWS e externos.

#### Quando usar:

- **SNS** → Disparar notificações (ex: e-mail, SMS) em tempo real.
  - **SQS** → Fila entre produtor e consumidor, com mensagens sendo processadas.
  - **EventBridge** → Reagir a eventos (ex: disparar Lambda quando S3 recebe objeto).
- 



## 6. IAM vs. Organizations vs. STS

- **IAM**: Gerencia permissões e usuários dentro da conta.
- **Organizations**: Gerencia múltiplas contas AWS.
- **STS (Security Token Service)**: Permissões temporárias entre contas.

#### Quando usar:

- **IAM** → Controle fino dentro da conta.
  - **Organizations** → Governança multi-conta.
  - **STS** → Acesso temporário e cross-account.
- 



## 7. CloudFront vs. S3 Website vs. ALB

- **CloudFront**: CDN global para cache e desempenho.
- **S3 Website**: Hosting estático simples.
- **ALB (Application Load Balancer)**: Balanceamento para aplicações HTTP/HTTPS.

#### Quando usar:

- **CloudFront** → Melhor performance global e cache.
  - **S3 Website** → Site estático básico, sem SSL nativo.
  - **ALB** → Aplicações web dinâmicas (back-end EC2 ou containers).
- 



## 8. CloudWatch vs. CloudTrail vs. Config

- **CloudWatch**: Monitoramento de performance e alarmes.
- **CloudTrail**: Registro de atividades da conta (auditoria).
- **AWS Config**: Histórico e compliance da configuração de recursos.

#### Quando usar:

- **CloudWatch** → Métricas e logs (ex: uso de CPU).
  - **CloudTrail** → Saber *quem fez o quê*.
  - **Config** → Avaliação de conformidade e rastreamento de alterações.
-

---

## 9. CloudFormation vs. Elastic Beanstalk vs. OpsWorks

- **CloudFormation:** Infraestrutura como código (IaC), 100% declarativo em YAML/JSON.
- **Elastic Beanstalk:** PaaS gerenciado para deploy de apps (Node, Java, Python...).
- **OpsWorks:** Gerenciamento de configuração baseado em Chef/Puppet.

**Quando usar:**

- **CloudFormation** → Controle completo da infraestrutura, altamente reutilizável.
- **Beanstalk** → Deploy rápido sem se preocupar com infraestrutura.
- **OpsWorks** → Ambientes complexos com automação de configuração.

---

## 10. S3 Glacier e Glacier Deep Archive

- Armazenamento **de longo prazo e baixo custo**.
- **Glacier:** Recuperação em minutos ou horas.
- **Glacier Deep Archive:** Mais barato, recuperação em horas.

**Quando usar:** Backup ou arquivamento que raramente será acessado.

---

## 11. KMS vs. Secrets Manager vs. Parameter Store

- **KMS (Key Management Service):** Criação e controle de chaves de criptografia.
- **Secrets Manager:** Armazena e rotaciona segredos (senhas, API keys).
- **Parameter Store:** Armazena parâmetros e segredos simples (parte do Systems Manager).

**Quando usar:**

- **KMS** → Criptografia de dados em S3, EBS, RDS etc.
- **Secrets Manager** → Aplicações que precisam de segredos com rotação automática.
- **Parameter Store** → Configs e segredos simples sem rotação automática.

---

## 12. Cost Management & Billing

- **Cost Explorer:** Visualiza custos e tendências.
- **Budgets:** Alertas para limite de gastos.
- **Savings Plans / Reserved Instances:** Redução de custos com compromissos.

**Quando usar:**

- Monitorar e otimizar custos de forma proativa.

---

## 13. VPC, Subnets, NACLs, SGs

- **VPC:** Sua rede virtual.

- **Subnets:** Divisão da rede (pública/privada).
- **Security Groups (SG):** Firewall de nível de instância (stateful).
- **NACLs:** Firewall de nível de subnet (stateless).

**Cai muito na prova:** saber diferenciar **SG vs. NACL**, e configurar acesso correto via **IGW, NAT Gateway, Route Tables**.

---

## 14. API Gateway vs. ALB (para APIs)

- **API Gateway:** APIs REST ou HTTP com integração ao Lambda e autenticação.
- **ALB:** Pode expor APIs HTTP com roteamento por path.

**Quando usar:**

- **API Gateway** → APIs serverless, com autenticação, throttle e monitoramento nativo.
  - **ALB** → Quando há backend com EC2 ou containers e API simples.
- 

## 15. Cognito vs. IAM Identity Center (SSO)

- **Cognito:** Autenticação de usuários em apps web/mobile.
  - **IAM Identity Center (SSO):** Acesso centralizado para múltiplas contas AWS e apps.
- 

## 16. Well-Architected Framework

São **cinco pilares** cobrados em questões de arquitetura:

1. **Operational Excellence** – Automação e monitoramento.
  2. **Security** – IAM, criptografia, controles.
  3. **Reliability** – Tolerância a falhas, recovery.
  4. **Performance Efficiency** – Escolha do serviço certo e elasticidade.
  5. **Cost Optimization** – Pagar só pelo que usa, economizar com planos.
- 

## 17. Route 53

- DNS gerenciado.
- Suporta **routing policies** como:
  - *Simple* (um único IP),
  - *Weighted* (balanceamento),
  - *Latency-based* (menor latência),
  - *Failover* (alta disponibilidade).

**Quando usar:** Gerenciar domínios, fazer failover geográfico, balancear carga baseado em latência.

---

## 18. WAF vs. Shield vs. GuardDuty vs. Inspector

- **WAF (Web Application Firewall):** Bloqueia ataques em camada 7 (SQLi, XSS).

- **Shield:** Proteção contra DDoS (Shield Standard é gratuito).
- **GuardDuty:** Detecção de ameaças (machine learning + logs).
- **Inspector:** Avaliação de vulnerabilidades em instâncias EC2 e containers.

**Quando usar:**

- **WAF** → Proteção de apps web.
  - **Shield** → Proteção automática contra DDoS.
  - **GuardDuty** → Análise de segurança automatizada.
  - **Inspector** → Scan de vulnerabilidades e compliance.
- 

## 19. Step Functions

- Orquestração de tarefas (ex: sequenciar Lambdas).
- Workflow serverless com controle de falhas, loops e paralelismo.

**Quando usar:** Processos com múltiplas etapas e decisões condicionais.

---

## 20. ECS vs. EKS vs. Fargate

- **ECS (Elastic Container Service):** Gerencia containers (orquestrador AWS nativo).
- **EKS (Elastic Kubernetes Service):** Gerencia clusters Kubernetes.
- **Fargate:** Execução de containers **sem gerenciar servidores** (pode usar com ECS ou EKS).

**Quando usar:**

- **ECS** → Orquestração simples, nativa da AWS.
  - **EKS** → Kubernetes quando há exigência de portabilidade.
  - **Fargate** → Containers sem precisar provisionar EC2.
- 

## 21. Systems Manager (SSM)

- Gerencia instâncias EC2 (ou on-premises) com automação, patching, run commands e inventário.

**Quando usar:** Automatizar operações em escala, acesso remoto seguro sem SSH.

---

## 22. Snowball / Snowcone / Snowmobile

- Transporte físico de dados para a AWS.
- **Snowball:** Até 80 TB.
- **Snowcone:** Até 8 TB.
- **Snowmobile:** Exabytes (contêiner caminhão).

**Quando usar:** Migração de grandes volumes quando upload online é inviável.

---

## 23. Backup e DR (Disaster Recovery)

- **AWS Backup:** Serviço centralizado para backups de RDS, EBS, DynamoDB, etc.
- **Disaster Recovery Strategies:**
  - **Backup & Restore**
  - **Pilot Light**
  - **Warm Standby**
  - **Multi-site / Active-Active**

Cai em perguntas de cenário. Ex: “O cliente quer restaurar em poucas horas e com custo controlado”.

---

## 24. Migration Services

- **AWS Migration Hub:** Centraliza tracking de migração.
  - **Server Migration Service (SMS):** Migração de VMs para EC2.
  - **Database Migration Service (DMS):** Migração de banco (homogênea ou heterogênea).
- 

## 25. Global Accelerator

- Redireciona tráfego para o endpoint mais saudável via **rede da AWS** (não via Internet pública como Route 53).
  - Melhora **latência e disponibilidade global**.
- 

## 26. Application Integration (integrações entre serviços)

- **Amazon MQ:** Broker gerenciado (compatível com ActiveMQ e RabbitMQ).  
→ Quando apps legados usam protocolos como AMQP.
  - **Amazon EventBridge** (e antes o CloudWatch Events):  
→ Para reagir a eventos de serviços AWS ou de aplicações próprias.
- 

## 27. DevOps & Infraestrutura como Código

- **CodeCommit:** Git gerenciado (como GitHub).
- **CodeBuild:** Build contínuo (CI).
- **CodeDeploy:** Deploy automático.
- **CodePipeline:** Pipeline completo CI/CD.

Cai em perguntas sobre automatização e zero-downtime deploy.

---

## 28. Machine Learning

- **Amazon Rekognition, Comprehend, Polly, Lex:**  
Serviços gerenciados para visão computacional, linguagem natural, voz e chatbots.

→ Aparecem raramente, mas pode haver **1 questão simples** tipo “qual serviço usar para identificar objetos em imagens?”.

---

## 29. Direct Connect

- Conexão **dedicada e privada** entre seu data center e a AWS.
- Mais estável e com menor latência que VPN.

→ Importante em **arquiteturas híbridas**.

---

## 30. AWS Certificate Manager (ACM)

- Gerencia certificados SSL/TLS gratuitamente.

→ Usado com **CloudFront, ALB, API Gateway**.

---

## 31. Service Catalog & Control Tower

- **Service Catalog:** Padroniza recursos (ex: instâncias EC2 aprovadas).
  - **Control Tower:** Governa várias contas AWS com boas práticas desde o início.
- 

## 32. IAM Políticas Avançadas

- **Resource-based policies, Service control policies (SCPs), Permission boundaries.**

→ Cai em perguntas sobre limitar o que usuários podem ou não fazer em contextos cross-account ou multi-org.

---

## 33. ElastiCache (Redis/Memcached)

- Cache gerenciado para melhorar performance de apps.

→ Quando a pergunta fala em **reduzir latência de leitura de banco**.

---

## 34. AWS Outposts

- Infraestrutura AWS **on-premises**.  
→ Aparece como solução para **compliance ou latência local**.
- 

## 35. AWS Global Services x Regionais

Saber **quais serviços são globais** (IAM, CloudFront, Route 53) e **quais são regionais** (EC2, S3, Lambda, RDS) é **essencial** para responder perguntas sobre replicação, failover e DR.