|  |  |  |
| --- | --- | --- |
| Nome do Serviço | Resumo do Serviço | Link Documentação |
| Amazon Macie | O Amazon Macie é um serviço de segurança e privacidade de dados totalmente gerenciado que usa machine learning e correspondência de padrões para descobrir e proteger seus dados confidenciais na AWS. À medida que as organizações gerenciam volumes crescentes de dados, identificar e proteger seus dados confidenciais em escala pode se tornar cada vez mais complexo, caro e demorado.  O Amazon Macie é um serviço de segurança e privacidade de dados totalmente gerenciado que usa machine learning e correspondência de padrões para descobrir e proteger seus dados confidenciais.  AWS Macie é um serviço de segurança e privacidade de dados totalmente gerenciado que usa machine learning e correspondência de padrões para descobrir e proteger seus dados confidenciais na AWS | <https://aws.amazon.com/macie/?nc2=h_ql_prod_se_ma> |
| AWS GuardDuty | Amazon GuardDuty é um serviço de detecção de ameaças.  O Amazon GuardDuty é um serviço de detecção de ameaças que monitora continuamente atividades mal-intencionadas e comportamentos não autorizados para proteger suas contas, cargas de trabalho e dados da AWS armazenados no Amazon S3.  O Amazon GuardDuty é um serviço de detecção de ameaças(não causa raiz) que monitora continuamente suas contas e workloads da para detectar atividade maliciosa e fornece resultados de segurança detalhados para visibilidade e correção. | <https://aws.amazon.com/guardduty/?nc2=h_ql_prod_se_gd>  <https://aws.amazon.com/pt/guardduty/?nc2=type_a>  <https://aws.amazon.com/pt/guardduty/> |
| Amazon LightSail | O Amazon Lightsail oferece instâncias de servidor privado virtual (VPS) fáceis de usar, contêineres, armazenamento, bancos de dados e muito mais a um preço mensal econômico. | <https://aws.amazon.com/pt/lightsail/?nc2=h_ql_prod_fs_ls> |
| Amazon DynamoDB | O DynamoDB é um banco de dados não relacional, sem servidor, totalmente gerenciado pela AWS, não possui esquema e é baseado em tabelas compostas por colunas de chave x valor. Pode ser utilizador para guardar dados não estruturados no formato Json e documentos.  **O Amazon DynamoDB** é um banco de dados de valores-chave e documentos que oferece desempenho em milissegundos de um dígito em qualquer escala.   * É um serviço que roda por default em multi AZ para garantir alta disponibilidade * E integrado ao CloudTRail onde todas as chamadas de apis são registradas como eventos e podem ser monitoradas e utilizadas em eventuais necessidades de auditoria. * Possui uma feature DynamoDB Stream que permite capturar eventos de acordo com a necessidade e pode ser integrado a outros serviços como uma função lambda, SNS entre outros para que em determinado tipo de evento ou registro uma ação seja disparada para algum tipo de processamento. * Possui a feature DAZ (DynamoDB Accelerator) Cache em memória para aumentar a performance de processamento em microssegundos. * Permite alocação de recurso com base nos requisitos de capacidade de taxa de transferência de leitura e gravação.   **Unidade de capacidade de leitura (1 item de até 4kb de tamanho)**   * 1 leitura fortemente consistente por segundo (RCU) * 2 leituras eventualmente consistentes por segundo.   **Unidade de capacidade de gravação (para 1 item de até 1kb de tamanho)**   * 1 gravação por segundo (WCU)     **Formatos de leitura e escrita suportado pelo DynamoDB:**   * **Eventually Consistent Reads (Consistência Eventual)**   + Se você ler um dado de uma tabela e receber um status 200, significa que a operação foi concluída com sucesso mas os dados retornados ainda não foram 100% atualizados, ou seja, se você consultar novamente alguns segundos depois vai receber os dados mais atualizados daquele registro. * **Strongly Consistent Reads (Consistência Fortemente)**   + Quando você faz uma consulta de um registro e recebe um response isso garante que os dados que você recebeu são os dados mais atualizados daquele registro. Porém esse modelo tem algumas desvantagens que precisam ser avaliadas antes de utilizar, como por exemplo.     - Só vai retornar os dados após todos estiverem atualizados o que pode gerar um retorno de erro 500 casos a operação ainda não tenha sido concluída.     - Pode ter um aumento na latência de leitura dos dados     - Não suporta leitura em SGI (Global Secundary Indixes).     - Utiliza em sua leitura de dados maior capacidade de throughput.   **O DynamoDB possui 2 tipos de classes de tabelas, sendo elas:**   * DynamoDB Standard - Pode ser utilizada para a maioria de workloads ou tipo de aplicações * DynamoDB Standard IA -  Pode ser utilizado para determinar custo e tem como exemplo alguns tipos de aplicações com log, site de mídia post.   **Exemplo de questões sobre banco de dados NoSql (DynamoDB)** | <https://aws.amazon.com/pt/dynamodb/?nc2=h_ql_prod_db_ddb>  <https://aws.amazon.com/pt/dynamodb/pricing/on-demand/>  <https://aws.amazon.com/pt/dynamodb/?nc2=type_a>  <https://aws.amazon.com/pt/search/?searchQuery=AWS+DynamoDb>  <https://docs.aws.amazon.com/pt_br/amazondynamodb/latest/developerguide/HowItWorks.ReadConsistency.html>  <https://docs.aws.amazon.com/pt_br/documentdb/latest/developerguide/db-cluster-fault-tolerance.html>  <https://www.youtube.com/watch?v=bTLoK2eHwi4&t=28s> |
| Amazon ElastiCache | **O Amazon ElastiCache** é um serviço que oferece uma solução de armazenamento de informações em cache de forma escalável.  Com o Amazon ElastiCache, você pode configurar, executar e escalar sem problemas conhecidos repositórios de dados na memória compatíveis com código aberto na nuvem. Crie aplicações com uso intenso de dados ou aumente a performance de aplicativos existentes recuperando dados de armazenamentos de dados na memória de alta taxa de transferência e baixa latência.  O Amazon ElastiCache é uma opção popular para casos de uso em tempo real, como armazenamento em cache, armazenamentos de sessão, jogos, serviços geoespaciais, análises em tempo real e filas  **O Amazon ElastiCache possui 2 distribuições de motores de cache, sendo elas:**  **Memcached**   * Trabalha com processamento em multithreading * Possui baixa manutenão e configuração * Facil escalabilidade horizontal com auto discovery   **Redis**   * Possui suporte para estrutura de dados * Trabalha com persistencia de dados com alta durabilidade * Possui sistemas de operações atomicas * Possui sistemas de mensagens de publicação e assinatura (publishing e subscriber) * Possui replicas de leitura e failover (para recuperação caso algum nó pare de funcionar). * Trabalha com modo de cluster e clusters em shard   Exemplo de questoes sobre o serviço de cache AWS ElastiCache:    2 | <https://aws.amazon.com/elasticache/?nc2=h_ql_prod_db_elc>  <https://aws.amazon.com/pt/elasticache/redis/> |
| Route53 | * É um serviço de DNS gerenciado pela AWS. * Route53 é um serviço global da AWS * É responsável por endereçamento de resolução de nomes em endereço IP. * Possui 100% de SLA em todas a zonas de disponibilidades * Possui redirecionamento por localização * Permite ativar recurso de acesso por geolocalização com failover baseado em regras definidas por você. * DNS = Domain Name Server, servidor geral de domínio (nele é identificado o próximo servidor que saberá responder com o endereço do domínio especificado que chama (Name Server) qual sabe o endereço do servidor onde está hospedado o site que se chama NSSERVER. * DNS Records   + Host (A ou AAAA) -> A = IPV4 e AAAA = IPV6 * ALIAS CNAME - são nomes que um endereço de domínio pode possuir por exemplo [blog.gisiona.com.br](http://blog.gisiona.com.br/) * Mail Exchenge (MX) * Service Records (SRV) - Serviço de registro que a empresa pode ter na sua topologia.   + Serviço -> LDAP   + Target -> IP   + Port   + Start Of Authority (SDA) -> Primary e NS * **Possui politicas de roteamentos e métricas sendo elas:**   + **Simple Routing Policy**     - Redireciona uma requisição para cada servidor de forma randômica e possui armazenamento em cache.     - Não permite configurar healthcheck, ou seja se o servidor cair vai continuar recebendo requisição e dando erro pois o servidor não está respondendo, para resolver este problema recomentada utilizar a configuração de **Multivalue Answer Routing Policy.**   + **Failover Routing Policy**     - Possui recurso para configurar um servidor como primário e secundário e assim que o servidor primário cair o trafego será redirecionado para o servidor secundário.     - Nesse tipo de configuração todo o trafego será direcionado para o servidor primário até esse servidor fique indisponível e pare de responder, assim todo o trafego será redirecionado para o servidor secundário.   + **Geolocation Routing Policy**     - Permite utilizar e distribuir aplicação em vários formatos de linguagem diferentes de acordo com a localização do usuário, por exemplo os usuários de paris acessaria meu site em Frances e os usuários do Brasil acessa o conteúdo do meu site em Português.   + **Geoproximity Routing (Traffic Flow Only)**     - Permite aplicar várias configurações para mudar o roteamento do trafego de acordo com a necessidade, por exemplo pode ser configurado um roteamento para que os usuários de um determinado país seja direcionado para o servidor mais próximo da região e também uma outra configuração que permite em caso de falha nessa região o trafego seja direcionado para outra região especificada.     - Permite a utilização do Traffic Flow = que significa condicionais de configurações combinadas que podem ser configurado de acordo com as regras especificadas da necessidade.     - Nesse tipo de politica é permitido utilizar todos os recursos de politicas para ser configurados através do Geoproximity   + **Latency-based Routing Policy**     - Essa politica utiliza de recurso de calculo de menor latência entre os servidores para poder direcionar o trafego de requisição do usuário para o servidor mais próximo em latência de resposta em milissegundos.   + **Multivalue Answer Routing Policy**     - Essa politica é bem parecida com a Simple Routing Policy, ou seja, utiliza do recurso de envio de trafego para todos os servidores ativos, porém se um servidor cair com a configuração do Healthcheck ativo o trafego será redirecionado para o próximo servidor mais próximo, Já a politica do Simple Routing Policy se um servidor cair ela não tem a inteligência de redirecionar a requisição para o próximo servidor sempre vai continuar recebendo a requisição mesmo com o servidor indisponível.   + **Weighted Routing Policy**     - pode ser configurado uma porcentagem de peso de de trafego de redirecionamento de requisição para um servidor ou para outro, ou seja, tipo 20% vai para Paris e 80% vai para São Paulo.     - Permite configurar no Route53 um recurso de Health Check para ficar checando se o servidor está respondendo e caso não responda é redirecionado o trafego para outro servidor configurado. | <https://aws.amazon.com/pt/route53/>  <https://aws.amazon.com/pt/route53/features/?refid=c623d581-46f6-43a2-b227-cabbee9cd673>  <https://disaster-recovery.workshop.aws/pt/services/networking/route53.html>  <https://docs.aws.amazon.com/pt_br/Route53/latest/DeveloperGuide/Welcome.html> |
| S3 |  | <https://aws.amazon.com/pt/s3/storage-classes/> |
| AWS Organizations |  | <https://docs.aws.amazon.com/pt_br/organizations/latest/userguide/orgs_introduction.html> |
| AWS Well-Architected | A AWS Well-Architected ajuda arquitetos de nuvem a construir infraestruturas seguras, resilientes, eficientes e de alta performance para aplicações e workloads. Baseado em seis pilares (excelência operacional, segurança, confiabilidade, eficiência de performance, otimização de custos e sustentabilidade), o AWS Well-Architected fornece uma abordagem consistente para que clientes e parceiros avaliem arquiteturas e implementem designs que podem se expandir com o tempo.  **AWS Well-Architected possui 6 pilares**   * Pilar Excelência operacional * Pilar Segurança * Pilar Confiabilidade * Pilar Eficiência de performance * Pilar Otimização de custos * Pilar Sustentabilidade | <https://aws.amazon.com/pt/architecture/well-architected/?nc1=h_ls>  <https://docs.aws.amazon.com/pt_br/wellarchitected/latest/framework/introduction.html>  <https://aws.amazon.com/pt/architecture/well-architected/> |
| RPO e RTO | RPO significa Recovery Point Objective ou seja está relacionado ao contextodo último ponto de backup que a empresa faz, ou seja se algo de errado acontecer de errado em sua infra estrutura qual é o último ponto de backup a ser recuperado e com isso é preciso conhecer bem o negócio para saber qual é o tempo limite do ponto de recuperação para ajustar sua estrutura e assim garantir que será cumprido a meta em caso de desastre recovery.  RTO significa Recovery Time Objective ou seja está relacionado ao contexto e tempo necessário para que sua infra estrutura e aplicação seja reestabelecida, ou seja se usarmos como exemplo um backup de banco de dados em quanto tempo será possível restaurar esse backup para que sua estrutura seja normalizada. Com isso é preciso conhecer bem o negócio para saber qual é o tempo necessário para restaurar  o backup para ajustar sua estrutura e assim garantir que será cumprido a meta em caso de desastre recovery. | <https://docs.aws.amazon.com/pt_br/wellarchitected/latest/reliability-pillar/recovery-time-objective-rto-and-recovery-point-objective-rpo.html> |
| Criptografia S3 | Existem 4 tipos de criptografia disponíveis para o Bucket, dentre elas podemos citar:  1 - SSE-S3: É a criptografia padrão, na qual é utilizado a chave "AWS/S3" que é uma chave padrão, gerenciada e mantida pela AWS. No entanto, não é possível utilizar o CloudTrail e gerenciar a chave, nem ver seu valor.  2 - SSE-KMS: É a criptografia no qual é informado a chave KMS para criptografia.  3 - SSE-C: É o tipo de criptografia no qual é necessário informar a chave que será utilizada pelo header da requisição, tanto no download quanto no upload.  4 - Client-side: É feita toda a criptografia e descriptografia no lado do cliente, desse modo a AWS apenas recebe o documento.  Por fim, vale lembrar também sobre a segurança, no qual é o conjunto do USER-BASED e RESOURCE-BASED.  O User-based é a segurança baseada nas permissões do usuário, e o resource-based é um documento JSON que representa uma policy de quem pode acessar o bucket e quais ações pode ser feitas pelo "principal". | <https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies.html>  <https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-encryption.html> |
| Storage Gateway | Oferece soluções de armazenamento on-premises com acesso a armazenamento virtual na nuvem praticamente ilimitado.  O Storage Gateway oferece um conjunto de protocolos padrão, como iSCSI, SMB e NFS, que permite o uso de armazenamento da AWS sem necessidade de alterar aplicativos existentes. O serviço também disponibiliza performance de baixa latência armazenando em cache dados acessados com frequência no local e, ao mesmo tempo, armazenando dados de modo seguro e resiliente nos serviços de armazenamento na nuvem da Amazon. O Storage Gateway otimiza a transferência de dados para a AWS enviando apenas dados alterados e compactando os dados. Além disso, o Storage Gateway é integrado nativamente ao armazenamento na nuvem do Amazon S3 e Amazon FSx for Windows File Server, disponibilizando os dados para processamento na nuvem; ao AWS Identity and Access Management (AWS IAM), protegendo o gerenciamento de acesso a serviços e recursos; ao AWS Key Management Service (AWS KMS), criptografando dados em repouso na nuvem; ao Amazon CloudWatch para monitoramento; e ao AWS CloudTrail para registro em log de atividades da conta.  O aws storage gateway possui 3 tipos de interfaces, sendo elas:  **Storage Volume Gateway**  É utilizado como um armazenamento de arquivo  no formato block starage utilizando o ISCSI com o S3 / EBS.  Suporta backup de volume EBS e permite criação de snapshots podendo ser restaurado em maquina virtual on-primeses ou em EC2 na aws.   * Permite também o recurso de Cached Volume onde todos os dados armazenados na aws são mantidos localmente para os arquivos mais acessados. * Permite também o recurso de Stored Volume mantém todos os dados em local para acesso rapido e com baixa latencia.   **Storage File Gateway**  Pode ser utilizado como um ponto de montagem de servidor de arquivo ou seja um file server compartilhado utilizando os protocolos de servidor de arquivos SMS (windows) e NFS (linux). Esse tipo de storage utiliza por debaixo dos pano o armazenamento em seu ambiente local e os dados são salvos no S3 da infra estrutura da AWS, vale lembrar que este tipo de montagem é um serviço regional inclusivo o bucket S3 precisa ser salvo na região que vc determinar.   * Esse modelo de armazenamento é suportado diversa classe de armazenamento do S3 como por exemplo S3 Standard, S3 IA, S3 Inteligent-Tiering. * Os dados mais acessados podem ser armazenando em cache local para facilitar e tornar o acesso mais rápido ao conteúdo. * Neste modelo de armazenamento os dados são armazenado em formatos de objetos visto que utiliza o S3 como repositório.   **Storage Tape Gateway**  È utilizado para substituir os serviços de backup em fitas que é geralmente utilizado pela empresas em seus data center local e trabalha com VTL (Virtual Tape Library). | <https://aws.amazon.com/pt/storagegateway/>  <https://aws.amazon.com/pt/storagegateway/file/>  <https://aws.amazon.com/pt/storagegateway/features/>  <https://aws.amazon.com/pt/storagegateway/file/?nc=sn&loc=2&dn=2>  <https://aws.amazon.com/pt/storagegateway/vtl/?nc=sn&loc=2&dn=3>  <https://aws.amazon.com/pt/storagegateway/volume/?nc=sn&loc=2&dn=4>  <https://aws.amazon.com/pt/storagegateway/hardware-appliance/?nc=sn&loc=2&dn=5>  <https://www.youtube.com/watch?v=ASMqf1t01bk&t=6s> |
| AWS Lambda e Lambda Edge | O AWS Lambda é um serviço de computação sem servidor e orientado a eventos que permite executar código para praticamente qualquer tipo de aplicação ou serviço de backend sem provisionar ou gerenciar servidores. Você pode acionar o Lambda a partir de mais de 200 serviços da AWS e aplicações de software como serviço (SaaS) e pagar apenas pelo que usar.  **AWS Lambda é um serviço de computação totalmente gerenciado pela aws.**   * Com o aws lambda seu código ou sua aplicação pode ser executado sem precisar se preocupar em gerenciar a infraestrutura de instancias EC2. * As funções lambda são iniciadas sempre através de um evento que triga o start da função lambda, um bom exemplo de integração desse tipo de serviço é o SNS, SQS, Api gateway, S3 events entre outros. * O serviço aws Lambda não permite acesso direto a instancia, caso precise analisar os logs que uma aplicação que foi executado em uma função lambda, pode ser integrado ao CloudWatch logs para que assim você possa ver os logs com detalhes.   **O que é o Lambda@Edge?**   * O [Lambda@Edge](https://aws.amazon.com/pt/lambda/edge/) permite que você execute códigos nas localizações da AWS globalmente sem a necessidade de provisionar ou gerenciar servidores, respondendo aos usuários finais com a mais baixa latência de rede. Basta fazer upload do código Node.js ou Python no AWS Lambda e configurar a função para que seja acionada em resposta a solicitações do [Amazon CloudFront](https://aws.amazon.com/pt/cloudfront/) (ou seja, quando uma solicitação do visualizador for recebida, uma solicitação for encaminhada para a origem, ou recebida de volta dela, e logo antes de a resposta ser enviada de volta para o usuário final). Depois disso, o código estará pronto para ser executado nas localizações da AWS globalmente quando uma solicitação de conteúdo for recebida, além de fazer o ajuste de escala de acordo com o volume das solicitações do CloudFront recebidas do mundo todo.   **Quando o Lambda@Edge deve ser usado?**   * O Lambda@Edge é otimizado para casos de uso que dependem da latência e cujos visualizadores estejam distribuídos globalmente. Todas as informações necessárias para tomar uma decisão devem estar disponíveis na borda CloudFront, dentro da função e da solicitação. Isso significa que os casos de uso de tomada de decisões sobre como distribuir conteúdo com base nas características do usuário (por exemplo, localização, dispositivo do cliente etc.) já podem ser executados e distribuídos perto dos usuários, sem que seja necessário acessar um servidor centralizado.   **Exemplo de questões referente ao serviço AWS Lambda.** | <https://aws.amazon.com/pt/lambda/>  <https://aws.amazon.com/pt/lambda/features/>  <https://aws.amazon.com/pt/lambda/faqs/> |
| Amazon Inspector | É um agente que faz varredura em instancia EC2 para encontrar vulnerabilidade de segurança. |  |
| VPC | O Amazon Virtual Private Cloud (Amazon VPC) oferece controle total sobre seu ambiente de redes virtual, incluindo posicionamento de recursos, conectividade e segurança. Comece a usar configurando sua VPC no console de serviço AWS. Em seguida, adicione recursos a ela, como instâncias do Amazon Elastic Compute Cloud (EC2) e Amazon Relational Database Service (RDS). Por fim, defina como suas VPCs se comunicam entre si, entre contas, zonas de disponibilidade (AZs) ou Regiões da AWS. No exemplo abaixo, o tráfego de rede está sendo compartilhado entre duas VPCs dentro de cada região.    **Conexão STATEFUL x STATELESS**  **Statefull**significa que se tiver regra de entrada habilitada não precisa de regra definida de saída, ou seja por default tudo que entra está liberado para sair... exemplo de serviços statefull é o **security group** onde é somente definido e liberado regras de entradas.  **Stateless**significa que se tiver uma regra de entrada e for necessário que precise de acesso para saída, ou seja precisa ter 2 regras de firewall definindo e especificando uma regra para entrada e outra saída permitindo assim o trafego (entrada e saída), exemplo desse tipo de serviço são as politicas de firewall **NACL**.  **Network Acess Control List -**são configurações que gerencia regras de entrada e saída dentro da VPC, um ponto muito importante é que a ordem das regras de ACL importam (por exemplo se for criado uma regra ACL dentro da VPC permitindo conexão SSH para sua subrede de numero da regra 90 e outra regra de numero 100 negando o acesso de entrada via protocolo SSH para subrede, o acesso vai funcionar, pois as regras de ACLs são lidas de forma sequencial, ou seja, se foi encontrado uma regra liberando o acesso as demais regras não serão validades)  **VPC Peering -**Uma conexão de emparelhamento de VPC é uma conexão de rede entre duas VPCs que permite rotear o tráfego entre elas usando endereços IPv4 privados ou endereços IPv6. As instâncias em qualquer VPC podem se comunicar umas com as outras como se estivessem na mesma rede. Você pode criar uma conexão de emparelhamento de VPC entre suas próprias VPCs ou com uma VPC em outra conta da AWS.  As VPCs podem estar em regiões diferentes (também conhecidas como conexão de emparelhamento de VPC entre regiões). Um ponto que precisa ter cuidado é que os endereçamentos das subrede não podem está na mesma camada de rede exemplo.. VPC 1 (subrede com endereço [10.0.0.0/16](http://10.0.0.0/16)) e VPC 2 (subrede com endereço de rede [10.0.0.0/16](http://10.0.0.0/16)) quando isso acontece o roteador que fazer fazer a conexão entre as VPCs não vai saber para onde direcionar o trafego de rede e com isso o trafego não será redirecionado para nenhuma VPC.  O emparelhamento de VPC podem ser feitas entre VPC dentro da mesma conta e região ou entre VPC em conta da aws diferentes  **VPC Endpoints -** Um endpoint da VPC permite conexões privadas entre uma virtual private cloud (VPC – Nuvem Privada Virtual) e os serviços compatíveis, sem exigir um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect. Portanto, você controla os sites, serviços e endpoints de API específicos que podem ser acessados a partir da sua VPC.  Os VPC endpoints são dispositivos virtuais. Eles são componentes de VPC escalados horizontalmente, redundantes e altamente disponíveis. A seguir estão os diferentes tipos de endpoints da VPC. Crie o tipo de VPC endpoint necessário para o serviço compatível.  Os VPC Endpoint são recomendados para conectar serviços que são executados dentro da rede privada como por exemplo EC2, CloudFormation e etc....  **Gateway Endpoint**- Um gateway endpoint serve para que um serviço da rede privada se comunique com um serviço da rede publica como por exemplo uma instancia do EC2 que é executada dentro da rede privada se comunique com um bucket do S3 ou um banco de dados DynamoDB que são executados na rede public, com o uso do gateway endpoint a conexão não passa pede internet e sim por uma conexão de rede privada/publica. Essa conexão da rede privada para um serviço na rede publica não tem a necessidade de usa um NAT, pois os endereço de conexão são registrados na tabela de roteamento (Routing Table).  **VPN Cliente**- As soluções do AWS Virtual Private Network estabelecem conexões seguras entre redes locais, escritórios remotos, dispositivos de clientes e a rede global da AWS. O AWS VPN é composto por dois serviços: AWS Site-to-Site VPN e AWS Client VPN. Juntos, eles entregam uma solução de VPN na nuvem gerenciada, altamente disponível e elástica para proteger o tráfego da sua rede.  O AWS Site-to-Site VPN cria túneis criptografados entre a sua rede e as Amazon Virtual Private Clouds ou os AWS Transit Gateways. Para gerenciar o acesso remoto, o AWS Client VPN conecta seus usuários à AWS ou a recursos locais por meio de um cliente de software VPN.  Conecta de forma segura uma rede local on-premisses com uma VPC privada na cloud AWS de uma VPN onde o protocolo de conexão é https através da porta tcp 443 e tudo é encriptado através dos protocolos de segurança SSL/TLS.  **VPN Site-To-Site -**Por padrão, as instâncias executadas na Amazon VPC não podem comunicar-se com sua própria rede (remota). Você pode habilitar o acesso à sua rede remota a partir da sua VPC criando uma Conexão VPN Site-to-Site AWS e configurando o roteamento para transmitir o tráfego pela conexão.  O termo *conexão VPN* possui sentido amplo, mas, nesta documentação, ele se refere especificamente à conexão entre a VPC e sua rede local. A VPN de local para local oferece suporte para as conexões VPN da Internet Protocol security (IPsec). | <https://aws.amazon.com/pt/vpc/features/>  <https://docs.aws.amazon.com/pt_br/AWSEC2/latest/WindowsGuide/enhanced-networking-ena.html>  <https://docs.aws.amazon.com/pt_br/vpc/latest/tgw/tgw-nacls.html>  <https://docs.aws.amazon.com/pt_br/vpc/latest/tgw/how-transit-gateways-work.html#tgw-attachments-overview>  <https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html>  <https://docs.aws.amazon.com/pt_br/vpc/latest/peering/create-vpc-peering-connection.html>  <https://docs.aws.amazon.com/pt_br/vpc/latest/privatelink/vpc-endpoints.html>  <https://aws.amazon.com/pt/vpn/>  <https://docs.aws.amazon.com/pt_br/vpn/latest/s2svpn/VPC_VPN.html> |
| Armazenamento EBS | É um tipo de armazenamento em bloco diferente do S3 que trabalha com armazenamento de objeto ou seja não incremental.   * Permite ser anexado em instancias EC2 na mesma zona de disponibilidade dentro da mesma região. * Persiste os dados permanente independente da instancia EC2 ser terminada * Fornece acesso aos dados rapidamente e trabalha com persistência de longo prazo. * Permite criação de snapshots * Os dados armazenados no EBS são replicados automaticamente dentro da mesma zona de disponibilidade (AZ). * Os volumes do EBS podem ser criptografados. * Os volumes do EBS não podem ser anexo em mais de uma instancia. * Existe uma feature nova do EBS que permite ataxar o volume em mais de uma instancia, mas depende do tipo de volume e esse recurso não está disponível ainda em todas as regiões.   Possui diversos tipos de volumes para atender todos os tipos de requisitos de negócios.  **SSD de uso geral**   * Fornece volume de inicialização de de aplicações e recomendado para inicializar sistema operacional entre outros tipos de necessidades. * Possui tamanhos de 1 giga até 16 terabytes. * Trabalha com máximo de IOPS de até 10.000 por segundos. * Possui taxa de transferência por volume de até 250 megabytes por segundo.   **SSD de IOPS provisionado (io1)**   * Fornece volume recomendado para sistemas de banco de dados NoSQL e aplicações que exige muita leitura e escrita. * Possui tamanhos de 4 giga até 16 terabytes. * Trabalha com máximo de OPS de até 64.000 por segundos. * Possui taxa de transferência por volume de até 1000 megabytes por segundo.   **HDD otimizada de taxa de transferência (st1)**   * Fornece volume recomendado para Big Data, Data Warehouses e processamento de logs. * Possui tamanhos de 125 giga até 16 terabytes. * Trabalha com máximo de IOPS de até 500 por segundos. * Possui taxa de transferência por volume de até 500 megabytes por segundo. * Possui taxa de transferência otimizada em seu processamento   **HDD a frio (sc1)**   * Fornece volume recomendado para grandes volumes de dados como por exemplo backup que não exige uma grande capacidade de processamento. * Possui tamanhos de 125 giga até 16 terabytes. * Trabalha com máximo de IOPS de até 250 por segundos. * Possui taxa de transferência por volume de até 250 megabytes por segundo.     **Exemplo de questões referente ao EBS** | <https://aws.amazon.com/pt/ebs/>  <https://aws.amazon.com/pt/ebs/volume-types>  <https://aws.amazon.com/pt/ebs/provisioned-iops/>  <https://aws.amazon.com/pt/ebs/throughput-optimized/>  <https://aws.amazon.com/pt/ebs/cold-hdd/>  <https://aws.amazon.com/pt/ebs/general-purpose/> |
| Amazon Elastic Load Balancing (ELB) | Serviço utilizado para distribuir requisições web entre varias instancias EC2 ou serviço de task de um cluster.  É muito recomendado também sua utilização para desacoplar aplicações.  O ELB possui alguns tipos e classes como por exemplo:  N**etwork Load Balancing**   * Trabalha na camada 4 do modelo OSI, ou seja, na camada de Transporte * Tem como principal função fazer o balanceamento de carga de pacote TCP * Muito utilizado para aplicações de alto desempenho * Possui pouca inteligência a nível de aplicação e trabalha muito bem para receber requisições e distribuir para um outro destino. * Trabalha com os protocolos de comunicação TCP, UDP, TLS   **Application Load Balancing**   * Trabalha na camada 7 do modelo OSI, ou seja, na camada de aplicação * Tem como principal função fazer o balanceamento de carga entre instancia. * Por atuar na camada de aplicação consegue fazer o roteamento baseado em cabeçalho da requisição ou baseado no conteúdo entre outros filtros que podem ser configurados * Usa cifras TLS para criptografar e descriptografar os dados da requisição. * Trabalha com os protocolos de comunicação HTTP, HTTPS, gRPC   **Classic Load Balancing**   * Não mais recomendado o seu uso na AWS, mas atua na camada 7 do modelo OSI camada de aplicação e foi substituido pelo ALB.     **Características adicionais do serviços Elastic Load Balancing:**  **Segurança**  Ao utilizar uma Amazon Virtual Private Cloud (VPC), você pode criar e gerenciar grupos de segurança associados ao Elastic Load Balancing para disponibilizar opções adicionais de rede e segurança para Application Load Balancer e Classic Load Balancer. Você pode configurar qualquer um dos balanceadores de carga para que fiquem voltados para a Internet ou criem um balanceador de carga sem endereços IP públicos para atuar como balanceador de carga interno (não voltado para a Internet).  **Alta disponibilidade**  Um Elastic Load Balancer é altamente disponível. Você pode distribuir o tráfego de entrada entre instâncias do Amazon EC2 em uma única zona de disponibilidade ou em várias zonas de disponibilidade. Um Elastic Load Balancer escala automaticamente sua capacidade de gerenciamento de solicitações em resposta ao tráfego de aplicações recebido. Para garantir que seus destinos estejam disponíveis e íntegros, o Elastic Load Balancer executa verificações de integridade em destinos a uma cadência configurável.  **Alta taxa de transferência**  O Elastic Load Balancer foi criado para assimilar o crescimento de tráfego e pode fazer balanceamento de carga de milhões de solicitações por segundo. Além disso, também absorve padrões de tráfego súbitos e voláteis.  **Verificações de integridade**  Um balanceador de carga elástico apenas roteia o tráfego para destinos íntegros, como instâncias do EC2, contêineres, endereços IP, microsserviços, funções Lambda e dispositivos. Com o Elastic Load Balancing, você obtém melhores insights sobre a integridade das aplicações de duas maneiras: (1) aprimoramentos na verificação de integridade que permitem configurar códigos de erro detalhados. As verificações de integridade permitem monitorar a integridade de cada um dos serviços por trás do load balancer; e (2) novas métricas que fornecem insights sobre o tráfego de cada um dos serviços em execução em uma instância do EC2.  **Sticky sessions**  As sticky sessions são um mecanismo usado para rotear solicitações do mesmo cliente para o mesmo destino. Os Elastic Load Balancers são compatíveis com sticky sessions. Esse recurso de sticky sessions, ou stickiness, é definido por grupo de destinos.  **Monitoramento operacional e registro em log**  O Amazon CloudWatch relata métricas do balanceador de carga da aplicação e clássico como número de solicitações, número de erros, tipos de erro, latência de solicitações e muito mais. O Amazon CloudWatch também monitora métricas de balanceadores de carga da rede e do gateway, como contagem de fluxos ativos, contagem de novos fluxos, bytes processados e muito mais. Os balanceadores de carga elástico também estão integrados ao AWS CloudTrail, que monitora chamadas de API para o ELB.  **Proteção contra exclusão**  Você pode habilitar o recurso de proteção contra exclusão em um Elastic Load Balancer para impedir que ele seja excluído acidentalmente.  **Exemplo de pergunta vinculado aos serviço Elastic Load Balancing.** | <https://aws.amazon.com/pt/elasticloadbalancing/>  <https://aws.amazon.com/pt/elasticloadbalancing/gateway-load-balancer/#Features>  <https://aws.amazon.com/pt/elasticloadbalancing/features/?nc=sn&loc=2&dn=1>  <https://aws.amazon.com/pt/elasticloadbalancing/application-load-balancer/?nc=sn&loc=2&dn=2>  <https://aws.amazon.com/pt/elasticloadbalancing/network-load-balancer/?nc=sn&loc=2&dn=3>  <https://aws.amazon.com/pt/elasticloadbalancing/classic-load-balancer/?nc=sn&loc=2&dn=5> |
| Amazon Elastic File System (EFS) | O Amazon Elastic File System (Amazon EFS) é um sistema de arquivos elástico de definição única, simples e sem servidor que permite compartilhar dados de arquivos sem provisionar nem gerenciar o armazenamento. Ele pode ser usado com produtos da AWS e recursos on-premises e foi criado para escalar sob demanda até a escala de petabytes sem interromper as aplicações.  O Amazon EFS é adequado para oferecer suporte a um amplo espectro de casos de uso, de diretórios pessoais a aplicações essenciais aos negócios. Casos de uso incluem armazenamento para aplicações em contêineres e sem servidor, análises de big data, gerenciamento de conteúdo e serviços web, desenvolvimento e testes de aplicações, fluxos de trabalho de mídia e entretenimento e backups de banco de dados.   * O Amazon EFS oferece classes de armazenamento Standard e One Zone para arquivos acessados com frequência e pouca frequência. * As classes de armazenamento Standard e One Zone têm a performance otimizada para oferecer latências baixas e consistentes. * As classes de armazenamento [Amazon EFS Standard-Infrequent Access](https://aws.amazon.com/pt/efs/features/infrequent-access/) (EFS Standard – IA) e [Amazon EFS One Zone-Infrequent Access](https://aws.amazon.com/pt/efs/features/infrequent-access/) (EFS One Zone – IA) têm custos otimizados para arquivos acessados com menos frequência.   Serviço de armazenamento de arquivo na nuvem AWS.   * Pode ser utilizado como armazenamento compartilhado como um serviço de rede. * Possui um sistema de arquivo em escala de petabytes de dados e pode trabalhar com os protocolos (NFS v4.00  e 4.1 NFFSv4) * Possui uma capacidade de crescimento elastica * Pode trabalhar com AMIs baseadas em linux para compatibilidade com amazon EC2. * É um serviço gerenciado pela AWS. * É compatível com sistema operacional Linux. * Trabalha com ponto de montagem de volume através de ENI (Elastic Network Interface) ou seja uma placa de rede virtual da AWS. * O EFS pode ser utilizada em varias instancias dentro da mesma zona de disponibilidade através do ponto de montagem com o ENI. | <https://aws.amazon.com/pt/efs/features/> |
| Amazon S3 | **Armazenamento de objetos construído para armazenar e recuperar qualquer volume de dados de qualquer local**  Serviço de armazenamento de objetos e totalmente gerenciado pela aws com capacidade de armazenamento ilimitado.   * Os buckets de armazenamento são do tipo regional, ou seja, o cliente escolhe em qual região quer armazenar seus dados * Possui recursos de garantia altamente durável de 11 noves (99,999999999%) ou seja a chance de perder um arquivo é muito baixo. * Permite trabalhar com politicas de ciclo de vida (lifecycle) * Possui diversas classes de armazenamento para atender todos os requisitos de negócios como por exemplo:   + S3 Standard (Classe default)   + S3 Glacier (utilizado para dados com pouco acesso)   + S3 Inteligent Tiering * Possui sistema de armazenamento seguro tanto para dados em transito (TLS) como dados em repouso (SSE-KMS, SSE-KMS, SSE-C). * Possui sistema de versionamento de arquivo, ou seja o cliente pode ter varias verões do mesmo arquivos, essa feature não é habilitada por default, o cliente precisa habilitar caso deseje ter esse recurso disponível. * Os dados armazenados no S3 são imutáveis, pois a única forma de alterar um objetos e substituir o objeto, pois o formato de armazenamento em objetos não permite edição do conteudo de um objeto. * O S3 permite armazenar um numero ilimitado de objetos. * Cada objeto pode ter até 5 terabytes de tamanho por objeto armazenado.   **Todos arquivo adicionado dentro do S# ele é bloqueado por default e caso o cliente deseje disponibilizar publicamente precisa trabalhar e habilitar os seguintes recursos disponíveis para controlar o acesso ao S3:**   * ACLs de objetos * ACLs de bucket * Politicas de bucket * Politicas do IAM   **Amazon S3 Glacier é utilizado para armazenar dados de backup de arquivos.**   * Permite trabalhar com politicas de ciclo de vida (lifecycle) * Permite ter criptografia nos dados * É um serviço global com visão regional * Altamente durável 11 noves * Permite configuração de recuperação (rápida, padrão ou em massa)   **Amazon S3 Standard Infrequent Access**   * Recomendado para armazenamento de objetos com pouca frequência de acesso. * Essa classe de armazenamento possui um menor custo por gigas armazenados, porém possui um custo maior caso você precise acessar os dados com uma frequência maior. * Essa classe possui um alto custo por solicitações de acesso aos objetos através de api com os methods (PUT, POST, COPY e GET) * Essa classe recomenda que seja no minimo armazenado por 30 dias para ser classificado na classe IA.   **Amazon SE One Zone:**   * Essa classe de armazenamento utiliza apenas uma zoza de disponibilidade para armazenar seus objetos. * Tem um custo menor por gigas armazenados, mas não possui alta disponibilidade, pois se essa zona de disponibilidade ficar indisponiel automaticamente não será possivel acessar seus objetos.   **Modelo de pagamento do uso do Amazon S3**   * O serviço é pago sob demanda de gigas utilizado por mes. * Transferência de dados para fora da região na qual os objetos estão armazenados. * Solicitações de PUT, COPY, POST, LIST e GET.   **Qual ações não é cobrado no uso do Amazon S3?**   * Upload de objetos para dentro do S3. * Transferência de objetos do S3 para outros serviços dentro da mesma região. * O recurso de delete de objetos do S3 não é cobrado.   **Questões de exemplos referente ao serviço S3:**    2 | <https://aws.amazon.com/pt/s3/features/?nc=sn&loc=2>  <https://aws.amazon.com/pt/s3/storage-classes/?nc=sn&loc=3>  <https://aws.amazon.com/pt/s3/pricing/?nc=sn&loc=4> |
| Arquitetura Tolerante a Falha | **Exemplo de estrutura de desenho de aplicações com arquitetura fracamente e fortemente acoplada**    **Pergunta de exemplos referente ao contexto de tolerancia a falha** | <https://docs.aws.amazon.com/pt_br/awssupport/latest/user/fault-tolerance-checks.html> |
| AWS CloudFormation | Serviço utilizado para implantar sua infraestrutura como código (IaaC) e gerenciar os recursos da AWS de uma maneira consistente e reproduzível.  **Exemplo de questão referente ao CloudFormation** | https://aws.amazon.com/pt/cloudformation/features/ |
| Amazon RDS  Banco de Dados Relacional) | Serviço de banco de dados gerenciados pela AWS, os bancos de dados disponíveis nesse serviços são os bancos relacionais como por exemplo (SQL Server, MySql, Aurora, Oracle, MariaDB, Postgree).  O Amazon RDS é um serviço de banco de dados relacional gerenciado que fornece seis opções de mecanismos de banco de dados conhecidos: [Amazon Aurora](https://aws.amazon.com/pt/rds/aurora/), [MySQL](https://aws.amazon.com/pt/rds/mysql/), [MariaDB](https://aws.amazon.com/pt/rds/mariadb/), [PostgreSQL](https://aws.amazon.com/pt/rds/postgresql/), [Oracle](https://aws.amazon.com/pt/rds/oracle/) e [Microsoft SQL Server](https://aws.amazon.com/pt/rds/sqlserver/). Isso significa que o código, as aplicações e as ferramentas que você já utiliza com os seus bancos de dados existentes podem ser usados com o Amazon RDS. O Amazon RDS executa tarefas rotineiras de banco de dados, como provisionamento, aplicação de patches, backup, recuperação, detecção de falhas e reparo.  O Amazon RDS facilita o uso da replicação para aumentar a disponibilidade e a confiabilidade de workloads de produção. Com a opção de implantação [multi-AZ](https://aws.amazon.com/pt/rds/details/multi-az/), você pode executar workloads essenciais à missão com alta disponibilidade e failover automatizado integrado de seu banco de dados principal para um banco de dados secundário replicado de forma síncrona. Usando as [réplicas de leitura](https://aws.amazon.com/pt/rds/details/read-replicas/), é possível aumentar a escala horizontalmente além da capacidade de uma única implantação de banco de dados para workloads de banco de dados com uso elevado de leitura.  O RDS pode ser utilizado em multi AZ no conceito de um banco de dados primário com replicação para demais zona de disponibilidade, nesse modelo o NÓ primário é executado e em caso de falha é acionado o NÓ secundário na outra zona de disponibilidade.  O RDS também permite trabalhar com replicas de leituras em múltiplas zonas de disponibilidade. As replicas de leituras são sincronizadas de forma síncrona.  O RDS replicas de leitura permite que seja escalado de forma onde somente as read replics são escalada de forma elástica para para garantir alta disponibilidade e performance.  **Caso de uso recomendado para utilizar o RDS**   * Aplicações que precisar utilizar consultas (query) complexas. * Consulta ou gravação de média a alta taxa de transferência. * Alta durabilidade dos dados, pois o RDS utiliza o armazenamento em EBS e permite que seja criado um snapshots para ser utilizado na criação de uma nova instancia em outra região por exemplo, ou até mesmo na recuperação de disastre recovery por exemplo.   **Caso de uso que não e recomendado utilizar o RDS.**   * Taxa de leitura e gravação massiva por exemplo (150 mil gravações por segundo). * Solicitações de consultas (GET e PUT) simples. * Necessidade de personalização ou gerenciamento do RDMS.   **Exemplo de questões sobre o RDS**  **1**    2 | <https://aws.amazon.com/pt/rds/features/>  <https://aws.amazon.com/pt/rds/faqs/> |
| Amazon CloudFront | Serviço de CDN Content Delivery Network, ou seja atua na camada de distribuição de conteúdo estático nas bordas (Edge Location) ou pontos de presença da AWS para agilizar a entrega de conteúdo ao usuário de acordo com sua localização para ter a menor latência na entrega e performance de sua aplicação.  **Entregue conteúdo com segurança com baixa latência e altas velocidades de transferência**   * Atua integrado com diversos serviços gerenciados da aws como por exemplo S3, EC2, ELB entre outros. * Possui uma camada de segurança integrado ao serviço AWS Sheild Stardard e WAF por default para atuar na defesa de ataques DDOS. | <https://aws.amazon.com/pt/cloudfront/features/?nc=sn&loc=2&whats-new-cloudfront.sort-by=item.additionalFields.postDateTime&whats-new-cloudfront.sort-order=desc>  <https://aws.amazon.com/pt/cloudfront/getting-started/?nc=sn&loc=4>  <https://aws.amazon.com/pt/cloudfront/faqs/?nc=sn&loc=5&dn=2> |
| Scaling Vertical x Horizontal | **O Scaling vertical** está relacionado ao aumento de recurso de processamento de uma máquina EC2 como por exemplo aumentar a memoria e cpu da maquina existente para que assim possa receber e ter maior capacidade de processamento.  **O Scaling Horizontal**está relacionado ao aumento de instancia EC2 como por exemplo tenho1 maquina processamento em determinado período aumento para 10 instancia EC2 para que assim possa receber e ter maior capacidade de processamento para atender uma carga de trabalho, esse processo de aumentar a quantidade de instancia se aplica também em remover a quantidade caso não seja mais utilizadas. |  |
| Amazon EC2 Auto Scaling  e AWS Auto Scaling | **AWS Auto Scaling**  O AWS Auto Scaling monitora os aplicativos e ajusta automaticamente a capacidade para manter um desempenho constante e previsível pelo menor custo possível. Com o AWS Auto Scaling, é fácil configurar a escalabilidade de aplicativos para vários recursos em diversos serviços em questão de minutos. O serviço oferece uma interface de usuário simples e eficiente que permite criar planos de escalabilidade para recursos, como instâncias e frotas spot do [Amazon EC2](https://aws.amazon.com/ec2/), tarefas do [Amazon ECS](https://aws.amazon.com/ecs/), tabelas e índices do [Amazon DynamoDB](https://aws.amazon.com/dynamodb/) e réplicas do [Amazon Aurora](https://aws.amazon.com/aurora/). O AWS Auto Scaling ajuda a simplificar a escalabilidade por meio de recomendações que permitem que você otimize o desempenho, os custos ou o equilíbrio entre eles. Se você já está usando o [Amazon EC2 Auto Scaling](https://aws.amazon.com/ec2/autoscaling/) para escalar dinamicamente instâncias do Amazon EC2, pode combiná-lo com o AWS Auto Scaling para escalar outros recursos de outros Serviços da AWS. Com o AWS Auto Scaling, seus aplicativos sempre dispõem dos recursos certos, na hora certa.  É fácil começar a usar o AWS Auto Scaling utilizando o [Console de gerenciamento da AWS](https://console.aws.amazon.com/console/home), a Interface da linha de comando (ILC) ou o SDK. O AWS Auto Scaling está disponível sem custos adicionais. Você paga apenas pelos recursos da AWS necessários para executar os aplicativos e pelas taxas de monitoramento do [Amazon CloudWatch](https://aws.amazon.com/cloudwatch/).   * Gerencia o scaling de múltiplos recusros e serviços gerenciados da aws * Atua com estratégica de scaling predefinidas disponíveis.   **Amazon EC2 Auto Scaling**  O Amazon EC2 Auto Scaling ajuda a manter a disponibilidade das aplicações e permite adicionar ou remover instâncias do EC2 automaticamente de acordo com as condições que você definir. É possível usar os recursos de gerenciamento de frota do EC2 Auto Scaling para manter a integridade e a disponibilidade da sua frota. Você também pode usar os recursos de escalabilidade dinâmica e preditiva do EC2 Auto Scaling para adicionar ou remover instâncias do EC2. A escalabilidade dinâmica responde às mudanças na demanda e a escalabilidade preditiva agenda automaticamente o número certo de instâncias do EC2 com base na demanda prevista. A escalabilidade dinâmica e a escalabilidade preditiva podem ser usadas juntas para escalas mais rápidas.   * É um serviço focando no escalonamento de instancia EC2 (atua com scale up e scale dow). * Executa e termina instancia de acordo com as regra definidas para aumentar ou diminuir. * Registra as instancia automaticamente nos balanceadores de carga (ELB) * Pode executar e escalar instancia em zonas de disponibilidades diferentes. * Integra-se ao Elastic Load Balancing * O Amazon EC2 Auto Scaling pode detectar quando uma instância não está íntegra, encerrá-la e substituí-la por uma nova. * O Amazon EC2 Auto Scaling garante que seu aplicativo sempre tenha a quantidade certa de recursos de computação e também provisiona capacidade proativamente com a escalabilidade preditiva. * O Amazon EC2 Auto Scaling adiciona instâncias apenas quando necessário e pode [escalar entre as opções de compra](https://aws.amazon.com/blogs/aws/new-ec2-auto-scaling-groups-with-multiple-instance-types-purchase-options) para otimizar o desempenho e o custo. * Permite trabalhar com escalonamento agendando     **Questão de exemplo referente ao auto scaling**    2    3 | <https://aws.amazon.com/pt/autoscaling/>  <https://aws.amazon.com/pt/autoscaling/faqs/>  <https://aws.amazon.com/pt/ec2/autoscaling/>  <https://aws.amazon.com/pt/ec2/autoscaling/features/>  <https://aws.amazon.com/pt/ec2/autoscaling/faqs/> |
| Amazon CloudWhatch | O Amazon CloudWatch é um serviço de monitoramento e observabilidade criado para engenheiros de DevOps, desenvolvedores, engenheiros de confiabilidade de sites (SREs), gerentes de TI e proprietários de produtos. O CloudWatch fornece dados e insights úteis para monitorar suas aplicações, responder às mudanças de performance de todo o sistema e otimizar a utilização de recursos. O CloudWatch coleta dados operacionais e de monitoramento na forma de logs, métricas e eventos. Você obtém uma visão unificada da integridade operacional e visibilidade completa de seus recursos, aplicações e serviços da AWS em execução na AWS e on-premises. Você pode usar o CloudWatch para detectar comportamentos anômalos nos seus ambientes, definir alarmes, visualizar logs e métricas lado a lado, executar ações automatizadas, solucionar problemas e descobrir insights para manter suas aplicações em execução contínua.  **O cloudwhatch pode monitorar :**   * CPU * Memória * Tamanho da fila SQS   Atua na coleta e armazenamento de métricas personalizadas para gerar alertas ou startar algum processo de acordo com a configuração definida.    **Questão de exemplo referente ao CloudWhatch** | <https://aws.amazon.com/pt/cloudwatch/>  <https://aws.amazon.com/pt/cloudwatch/features/>  <https://aws.amazon.com/pt/cloudwatch/getting-started/>  <https://aws.amazon.com/pt/cloudwatch/faqs/> |
| Serviços Excelência Operacional | O requisito de excelência operacional tem a capacidade de executar e monitorar os sistemas para agregar valor comercial e melhorar continuamente os processos e os procedimentos de suporte para (Preparar, Operar, Evoluir). Para isso alguns requesitos faz necessários:   * Execute as operações e montagem de workloads com IaaC (infra estrtutura como codigo). * Faça anotações na documentação * Faça alterações frequentes, pequenas e reversiveis * Refine frequentimente os procedimentos de operações * Antecipe a falhas * Aprenda com todas as falhas operacionais. * AWS Config   + Ajuda com a base de configuração para gerar alerta da conta caso alguma coisa sai do definido e experad, por exemplo a conta precisa que todas as imagens do EC2 utilize uma especifica. * AWS CloudFormation   + Ajuda na criação e manutenção da infra estrutura como código * AWS Trusted Advisor   + Gera relatórios da sua conta baseado no framework well architect (como recomendação de melhorar o uso de determinados serviços baseado em questões de segurança, custos e performance. * Amazon Inspector   + Atua como varredura das instancias EC2 para identificar vulnerabilidades * Amazon VPC Flow Logs]   + Loga e armazena tudo que acontece na VPC   + Captura logs a nivel de IP da camada 3 e 4 do modelo OSI] * AWS CloudTrail   + Monitora e gera logs de todas as execuções de chamadas de apis dentro da aws que os usuários executam, por exemplo caso um usuário desligue uma instancia EC2 será registrado que o usuário XPTO fez determinada operação. | <https://wa.aws.amazon.com/wat.pillar.operationalExcellence.pt_BR.html> |
| Modelo de responsabilidade compartilhada na AWS | * O cliente é responsável pela segurança na NUVEN * A AWS é responsável pela segurança da NUVEM. | [https://a](https://a/)ws.amazon.com/pt/compliance/shared-responsibility-model/ |
| AWS IAM | Gerencie com segurança as identidades e o acesso aos produtos e recursos da AWS   * Gerencia de forma centralizada os usuários e permissões de usuários dentro da aws. * Controla quais recursos os usuários e ou aplicações podem acessar * Cria politicas de usuários, grupo e funções * Permite integração dom Microsoft Active Directory e AWS Directory Service. * Trabalha com o principio do menor privilegio de acesso dentro da aws, ou seja tudo é negado até que seja liberado.   O AWS IAM funciona de forma integrada com alguns recursos de autenticação como por exemplo   * Grupos de usuários de IAM * Funções do IAM * Usuários federados * Usuários federados de identidades da web   **Questão de exemplo referente ao IAM**    **2** | <https://aws.amazon.com/pt/iam/features/?nc=sn&loc=2>  <https://aws.amazon.com/pt/iam/faqs/?nc=sn&loc=5>  <https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/best-practices.html>  <https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/IAMBestPracticesAndUseCases.html> |
| Segurança Dados em Transito | A transferência de dados de entrada e saída da sua infra estrutura AWS. A seguranção de dados em transito é através dos protocolos de segurança.   * SSL/TLS pela web * IPSec para conexões on-premises   + VPN   + AWS Direct Connect * Importar e exportar dados com Snowball   Os dados transferia via apis da AWS   * As chamadas de apis da aws usam o protocolo HTTPS * Todas as chamadas são assinadas com Sigv4 | https://docs.aws.amazon.com/pt\_br/wellarchitected/latest/security-pillar/protecting-data-in-transit.html |
| Segurança Dados em Repouso | Os dados armazenados (em repouso) dentro da infra estrutura da AWS são seguros pois utilizam criptografias e chave de acesso privadas.  Os dados armazenados no S3 são privados por padrão e exige credenciais da aws para serem acessados.   * OS acesos aos objetos do S3 são realizados via protocolo HTTP ou HTTPS. * Possui auditoria de acesso de todos os objetos (Configurados) * È permitido criar politicas baseadas em recursos como por exemplo:   + Buckets   + Prefixos (diretório/pasta) dentro do bucket   + Objetos | <https://docs.aws.amazon.com/pt_br/amazondynamodb/latest/developerguide/encryption.howitworks.html> |
| Criptografia de dados | OS dados podem ser criptografa de 2 modos, do lado do servidor e do lado do cliente.   * **Criptografia no lado do servidor**   + Chaves gerenciadas pelo Amazon SE (SSE-S3)   + Chaves gerenciadas pela AWS KMS (SSE-KMS)   + Chaves providas pelo cliente (SSE-C) * **Criptografia no lado do cliente**   + Chave primária do cliente gerenciada pela AWS KMS (CSE-KMS)   + Chave primária do lado do cliente (CSE-C) | <https://docs.aws.amazon.com/pt_br/AmazonS3/latest/userguide/serv-side-encryption.html> |
| AWS KMS (Key Management Service) | O AWS Key Management Service (KMS) facilita a criação e o gerenciamento de chaves criptográficas e o controle do seu uso em uma ampla variedade de serviços da AWS e nas suas aplicações. O AWS KMS é um serviço seguro e resiliente que usa módulos de segurança de hardware validados ou em processo de validação pelo FIPS 140-2 para proteger suas chaves. O AWS KMS é integrado ao AWS CloudTrail para fornecer logs contendo toda a utilização das chaves para ajudar a cumprir requisitos normativos e de compatibilidade   * Serviço de gerenciamento de chaves de criptografia * Possibilita que o conteúdo da chave do cliente seja importado * Aceita chamadas diretas ds aplicações para criptografia/descriptografia de chave * Alterna as chaves automaticamente e gerencia o uso de chaves antigas. * Trabalha somente com criptografia simétrica * Possui um modelo de cobrança por chamada de apis dentro da aws e por chaves gerenciadas. * E totalmente gerenciado pela AWS   O KMS possui integração com diversos serviços e recursos da AWS   * Amazon EBS * Amazon S3 * Amazon DynamoDB * Amazon RDS * Amazon EMR * Amazon SES * Amazon Redshift * Amazon CloudTrail * Entre outros....-   **Exemplo de integração do KMS com o EBS**    **Questão de exemplo referente ao KMS** | <https://aws.amazon.com/pt/kms/>  <https://aws.amazon.com/pt/kms/features/>  <https://aws.amazon.com/pt/kms/faqs/> |
| Amazon CloudHMS | O AWS CloudHSM é um Hardware Security Module (HSM – Módulo de segurança de hardware) baseado na nuvem que permite gerar e usar facilmente suas próprias chaves de criptografia na Nuvem AWS. Com o CloudHSM, você pode gerenciar suas próprias chaves de criptografia usando HSMs validados pelo FIPS 140-2 nível 3. O CloudHSM oferece a flexibilidade de integrar-se aos seus aplicativos usando APIs padrão do setor, como bibliotecas Microsoft CryptoNG (CNG), PKCS#11 e Java Cryptography Extensions (JCE).  O CloudHSM está em conformidade com as normas do setor e permite exportar todas as chaves para a maioria dos outros HSMs disponíveis no mercado, dependendo das suas configurações. Ele é um serviço gerenciado que automatiza para você tarefas administrativas demoradas, como provisionamento de hardware, aplicação de patches de software, alta disponibilidade e backups. O CloudHSM também permite que você ajuste a escala rapidamente ao adicionar e remover capacidade HSM sob demanda, sem custos antecipados.   * Gerenciamento de chave de criptografia baseado em hardware * O CloudHSM gera e armazena com segurança as chaves criptografada * Pode ser usada dentro do próprio aplicativo do cliente * As chaves são gerenciadas somente pelo cliente * CloudHSM ainda não está disponível em todas as regiões * Possui um modelo de cobrança por hora e o KMS possui o modelo de cobrança por chamadas de api com uso de chaves. * Trabalha com um modelo de criptografia simétrica ou assimétrica   **Diferença entre CloudHSM X KMS** | <https://aws.amazon.com/pt/cloudhsm/>  <https://aws.amazon.com/pt/cloudhsm/features/>  <https://aws.amazon.com/pt/cloudhsm/faqs/> |
| Amazon Macie | O Amazon Macie é um serviço de segurança e privacidade de dados totalmente gerenciado que usa machine learning e correspondência de padrões para descobrir e proteger seus dados confidenciais na AWS. À medida que as organizações gerenciam volumes crescentes de dados, identificar e proteger seus dados confidenciais em escala pode se tornar cada vez mais complexo, caro e demorado.  O Amazon Macie é um serviço de segurança e privacidade de dados totalmente gerenciado que usa machine learning e correspondência de padrões para descobrir e proteger seus dados confidenciais.  AWS Macie é um serviço de segurança e privacidade de dados totalmente gerenciado que usa machine learning e correspondência de padrões para descobrir e proteger seus dados confidenciais na AWS | <https://aws.amazon.com/macie/?nc2=h_ql_prod_se_ma> |
| AWS GuardDuty | Amazon GuardDuty é um serviço de detecção de ameaças.  O Amazon GuardDuty é um serviço de detecção de ameaças que monitora continuamente atividades mal-intencionadas e comportamentos não autorizados para proteger suas contas, cargas de trabalho e dados da AWS armazenados no Amazon S3.  O Amazon GuardDuty é um serviço de detecção de ameaças(não causa raiz) que monitora continuamente suas contas e workloads da para detectar atividade maliciosa e fornece resultados de segurança detalhados para visibilidade e correção. | <https://aws.amazon.com/guardduty/?nc2=h_ql_prod_se_gd>  <https://aws.amazon.com/pt/guardduty/?nc2=type_a>  <https://aws.amazon.com/pt/guardduty/> |
| Amazon LightSail | O Amazon Lightsail oferece instâncias de servidor privado virtual (VPS) fáceis de usar, contêineres, armazenamento, bancos de dados e muito mais a um preço mensal econômico. |  |