
ESCOLA LINUX
TREINAMENTOS
Guia de
ESTUDOS
amazon
web services™
João Batista Correa Jr
ECNOAPP

Sobre o autor

João Batista Correa Jr. é natural de Porto Alegre - RS, trabalha desde 2002 com informática, formado em ciências humanas - Licenciatura - com pós graduação em história. Entre as certificações em TI se destacam LPI-2, LPI-304 (Virtualização e alta disponibilidade), Amazon Arquitetura de soluções - Associate. Sócio da empresa de consultoria em TI; Tecnoapp Soluções de Informática (www.tecnoapp.com.br), instrutor na Escola Linux com cursos de, Virtualização Proxmox, Amazon Web Services entre outros.

Sobre a Escola Linux

Observamos que a tecnologia se transforma em alta velocidade, o mercado tem exigido cada vez mais dos seus profissionais e o tempo tem se tornado escasso para adquirir novas informações e conhecimentos para acompanhar essas transformações. Baseado nisto, a Linux Solutions tem investido em cursos com duas perspectivas diferentes:

CURSOS IN COMPANY: São cursos direcionados para necessidades das empresas. O instrutor responsável pelo treinamento planeja suas aulas de acordo com a disponibilidade de tempo, grau de conhecimento e assuntos de interesse da empresa e dos participantes do treinamento. O horário e dias são definidos junto ao cliente de acordo sua necessidade.

CURSOS ONLINE GRAVADOS: Cursos chamados de Online, com uma duração um pouco maior, no qual o aluno recebe um treinamento objetivo e dinâmico através da internet. São cursos ministrados por profissionais conceituados e atuantes no mercado. As aulas acontecem pela internet e já se encontram com todas as aulas gravadas.

Esses treinamentos são planejados para estudantes e profissionais que desejam se aprofundar no assunto específico ligado ao Ambiente Linux Open Source.

Treinamentos com Foco em:

- Segurança de Rede
- E-mail
- Arquivos
- Virtualização
- Alta Disponibilidade
- Inteligência Competitiva

Prepare-se para os novos desafios do mercado tecnológico, aprendendo os principais conceitos e práticas, garantindo assim as melhores oportunidades!

Sumário

CAPÍTULO 1 - O que é AWS?

***A**mazon Web Services*, também conhecido como **AWS**, é uma plataforma de serviços de computação em nuvem, que formam uma plataforma de computação na nuvem oferecida pela Amazon.com. Os serviços são oferecidos em 16 diferentes regiões distribuídas no mundo. Lançado oficialmente em 2006, Amazon Web Services é um provedor de serviços online para websites ou aplicações cliente servidor baseado na nuven. A maioria destes serviços não é acessível pela internet, mas oferecem funcionalidades que outros desenvolvedores podem usar em suas aplicações. Amazon Web Services pode ser acessado usando HTTP, protocolo REST, estilo de arquitetura ou pelo protocolo SOAP. O modelo de cobrança pelos serviços é de acordo com o uso.

Com este material você terá de forma rápida e pratica acesso aos principais tópicos relacionados a essa poderosa plataforma.

"Este **GUIA DE ESTUDOS** foi desenvolvido como uma ferramenta complementar de estudos para a certificação Amazon Arquiteto de soluções - Associate. Baseado no livro oficial *AWS Certified Solutions Architect Official Study Guide: Associate Exam 1st Edition.* Você terá de forma rápida e pratica acesso aos principais tópicos relacionados a essa poderosa plataforma, bons estudos!"

CAPÍTULO 2 - S3 e Glacier

Serviço central para armazenamento de objetos na AWS

- Armazenamento ilimitado e dados com alta durabilidade.
- Usado para backup, arquivamento, conteúdo web, analise de bigdata, website estático, disaster recovery, etc.
- Integrado a vários serviços aws como IAM, KMS, EC2, EBS, EMR, DynamoDB, Redshift, SQS, Lambda e Cloudfront
- Diferença do object storage (s3) para um tradicional storage: Storage de bloco tradicional acessa endereços de bloco dos dispositivos e armazenamento de arquivos são operados pelo sistema operacional através de níveis com arquivos e pastas. Object Storage contém tanto metadata como data, manipulado por API.
- Bucket são containers para armazenamento de objetos. Os nomes dos buckets devem ser globalmente únicos.
- Cada Bucket é criado em uma região específica, os dados não são replicados para outras regiões a não ser que seja configurado pelo usuário.
- Cada objeto armazenado no bucket pode ter de 0 até 5TB de tamanho.
- Objetos são identificados por chaves (nome do arquivo).
- Não existem diretórios (ou pastas) no S3, para simular isso, a chave do objeto pode conter barras, contra-barras, etc, para simular e melhor organizar o conteúdo. (/ , \).
- Cada objeto pode ter um endereço único de URL composto por web service endpoint + bucket name e a key do objeto(nome do arquivo).
- S3 usa uma API mínima para operações básicas e interface REST baseada em padrões HTTPS como GET, PUT, POST e DELETE.
- S3 tem durabilidade de 11 noves (99,999999999%) e 4 noves para disponibilidade (99,99%).
- S3 tem consistência eventual, ou seja arquivos deletados podem talvez serem lidos. Arquivos alterados podem ser lidos como antes da alteração, porém novos objetos 'PUT' somente são lidos após a confirmação de gravação.
- Objetos S3 são privados por padrão e podem ser alterados para público, para exibição de páginas de internet estáticas por exemplo.

- Controles de Acessos podem ser usados através de ACL , AWS IAM, bucket Policies e URLs de conjunto de caracteres com expiração

- S3 lificycle permite configurar os objetos para tipos diferentes de armazenamento conforme o tempo de vida.

- S3 lifecycle: Standard (uso geral) > Standard-IA (acessos menos frequentes) > RRS (baixa redundância com 99,99% de durabilidade) > Glacier (baixo custo com tempo de recuperação maior, entre 3 a 5 horas)

- Limite máximo de provisionamento de 100 buckets, mais que isso deve ser requisitado

- S3 pode usar criptografia pelo lado do servidor ou cliente. Chaves podem ser gerenciadas pelo AWS KMS

- Deleção segura de objetos ativando o MFA Delete.
- Possibilidade de replicação de buckets entre regiões para novos objetos após ativada a configuração.

- URL pré-assinada com permissão de limite de tempo via SDK

- Upload Multipart para objetos grandes e Range GETS para download de porções de objetos do S3 ou Glacier

- Logs podem ser ativados nos buckets para rastreio de ação nos objetos

- Tipos de de criptografia suportadas: SSE-S3 (Aws gerencia as chaves), SSE-C (cliente gerencia as chaves), SSE-KMS (AWS KMS gerencia) ou uma biblioteca do cliente

- Notificação do S3 para SNS, SQS ou função Lambda.
- S3 Transfer acceleration pode ser usado para habilitar upload e download a partir dos endpoints da AWS
- Glacier pode ser usado separadamente ou dentro da configuração do ciclo de vida do S3

- Usando o ciclo de vida do S3, somente podemos usar as APIs do S3 para obter acesso ao arquivo. Ao solicitar uma recuperação de arquivos, uma cópia é armazenada no S3 para recuperação.

- Glacier armazena dados em arquivos. Os container são chamados Vaults, limite de 1000 vaults e sem limite de arquivos.
- Quanto aos tempo de recuperação do Glacier. A *expressa* é disponível para arquivos menores de 250MB e estão disponíveis de 1 a 5 minutos, a *padrão* é de 3 a 5 horas e a *recuperação em massa* de 5 a 12 horas
- Cada arquivo pode ter até 40Tb de tamanho limite
- Glacier pode ter permissões bloqueadas.

Exercícios

1 Criar um bucket em uma determinada região pelo Aws Console

Upload de arquivo no bucket (imagem)
Testar acesso ao objeto pelo link URL
Tornar objeto publico
Testar acesso ao objeto pelo link URL
Renomear
Testar acesso ao objeto pelo link URL
Deletar objeto
Testar acesso ao objeto pelo link URL

2 Habilitar versionamento do bucket

Criar um arquivo de texto com uma palavra e upar o objeto
Alterar a palavra no objeto e upá-lo
Analisar as diferenças de versões nos arquivos upados
Observar o ID de versão de arquivos na URL disponibilizada (no final do endereço colocar ?versionId=ESCREVER_ID_DA_VERSÃO)

3 Excluir objetos e restaurar

Excluir versões
Excluir arquivos
Recuperar arquivo excluído

4 LifeCycle

Ativar LifeCycle em Settings > Management
Analisar as possibilidades

5 Habilitar Web Static hosting

Ativar
Enviar arquivos index.txt e error.txt
Transformar em público
Acessar o link correto e forçar o acesso a um link com problemas
Excluir conteúdo

CAPÍTULO 3 - AWS EC2 e EBS

- Descrição Instance type mostra o tipo de hardware alocado
- AMI define o software inicial da instância, tanto o OS como a aplicação
- AMIs podem ser:
  Genérica Publicada pela AWS
  Market Place, publicada por partners
  Criada por Usuários em geral
  Criada através de uploads de virtual servers (importadas) Import/Export.
  Somente podemos exportar VMs importadas
- Instâncias podem ser acessadas por Public DNS, Public IP ou Elastic IP
- No Linux o acesso é por SSH usando um par de chaves
- No Windows a senha do administrador é gerada randomicamente e criptografada, exigindo ser descriptografada antes do acesso
- Firewall por host se chama Security Group que permite regras que podem bloquear tráfego baseado em direção, porta, protocolo e source/destination address
- Bootstrapping permite rodar scripts na primeira vez que as instâncias são inicializadas.
- Com a instância criada, é possível alterar o tipo da instância e Security groups associados
- Opções de instâncias
  Reserved Instance (menor preço para servers q devem ficar sempre ligados. Com possibilidade de escolha de tempo de reserva e % de pagamento no Upfront)
  Spot (redução de preços e instância pode parar a qualquer momento)
- Até 10 Tags por instância
- Os meta-data da instância pode ser obtido através de um acesso http ao endereço http://169.254.169.254/latest/meta-data/ à partir da própria instância
- Enhanced networking: São IOs mais diretos a placas de rede com menor Jitter, alta performance (10gb) e baixa latência
- EBS magnetico de 1Gb a 1 TB (Pago por quantidade alocada)
- EBS General-Purpose SSD 1Gb até 16 TB. Até 10.000 IOPS. Pequenos Volumes até 3.000 IOPS
  Fornece IOPS conforme quantidade de Gb provisionados
- EBS Provisioned IOPS SSD 4Gb até 16 TB
  Pago por quantitade alocada e IOPS selecionados. Até 20.000 IOPS
- Instance Store tem os dados apagados quando a instância é desligada ou terminada. Quando reinicia não apaga os dados.
- Snapshots são backups incrementais feitos no Aws S3
- Limite de execução de 20 instâncias EC2. Para alguns tipos maiores esse número diminui. Se precisa mais do que isso, pode preencher o formulário de solicitações da AWS
- Limite de 5 Elastic IP por conta e por região. Precisando mais que isso, deve ser preenchido um relatório da AWS
- A garantia de SLA de tempo de funcionamento mensal para os serviços de EC2 e EBS para cada região é de 99,95%

CAPÍTULO 4 – VPC

- VPC é uma rede lógica e isolada na AWS Cloud
- Por padrão, podemos ter até 5 VPCs configuradas para cada conta da AWS em uma região
- Componentes chaves da VPC:
  Subnets, Route Tables, DHCP Options sets, Security groups, Network ACL
- Componentes opcionais:
  IGWs (Internet Gateway), EIP (elastic IP Address), (limite de 5 EIP por conta em cada região), Endpoints, Peering, Nat instance e Nat Gateway, VPG (Virtual Private Gateway) (Limite de 5 VPG por conta e 10 VPN Ipsec por VPG), CGW (Customer gateway) e VPN.
- Intervalo de IP em uma VPC padrão: 172.31.0.0/16, as subnets recebem a rede /20
- Subnets permitem segmentar a rede e alocá-las em diferentes zonas de disponibilidade
- Menor subnet reservada: /28 sendo 5 reservados para aws sobrando 11 ips para utilização
- 3 tipos de VPC
  Pública: Roteia o acesso para o IGW (internet Gateway), Privada: Não roteia o acesso para o IGW, Vpn-only: Roteia o acesso para VPG (Vpn Gateway).
- O IP interno da instancia é sempre privado e não roteável para internet
- IGW utiliza NAT para acesso das instancias a internet. Quando uma Subnet esta associada a um IGW então é pública
- EIP: Permite alocar ips públicos a instâncias de maneira que não se percam ao terminal ou parar a instância
- VPC Endpoint permite criar conexões privadas entre a VPC e serviços da AWS em uma mesma região sem usar a internet
- VPC Peering: Permite conectar uma VPC a outra em uma mesma região, do mesmo ou de outra AWS account, des de que não exista conflito de IPS.
  Também não é possível rotear entre as VPCs
- Até 500 Security groups pode ser criados por VPC
  Cada grupo pode ter 50 regras de entrada e 50 regras de saída; Cada interface de rede pode ter até 5 security groups associadas; Security groups são stateful; SG (Security group) default permite saída total e conexão entre as instâncias do SG default.
- Network ACL é uma outra camada de segurança
  Firewall a nível de subnet de rede; Stateless; Análise por ordem de regras.
- Nat instance e NAT gateway permitem uma subnet privada acessar a internet
  NAT gateway é gerenciada pela AWS como um serviço
- VPG é o 'gateway' que receberá a conexão VPN (serviço da AWS)
- CGW (customer gateway) é o hardware ou aplicação para estabelecer VPN com a Aws
- Protocolo de VPN Ipsec
- A VPN consiste em 2 túneis para alta disponibilidade.

Exercícios

1 Criar uma VPC

Range 192.168.0.0/16 nome VPC Teste
Default tenacy

2 Criar duas subnets em VPC Teste

192.168.1.0/24 (Subnet publica) em AZ-A
192.168.2.0/24 (Subnet Privada) em AZ-B

3 Conectar a VPC criada com a internet

Criar um Internet Gateway com o nome 'IGW Teste'
Atachar a VPC 'VPC Teste'
Adicionar uma nova rota 0.0.0.0/0 associada ao IGW TEste na VPC Teste
Criar uma NAT Gateway associada a Subnet Publica e vincular um EIP
Criar uma nova tabela de rotas chamada 'Rotas Subnet privada' com a regra 0.0.0.0/0 associada ao nat criado.
Associar a nova tabela de rotas a subnet privada

4 Fazer testes com Instancias na subnet pública (com e sem vinculação de IP público) e subnet privada

Sem IP público ele não navega mesmo na subnet pública
NA subnete privada ele navega pelo NAT sem precisar de associação de iP

CAPÍTULO 5 - ELB, Amazon CloudWatch e Auto Scaling

- **ELB** funciona nas seguintes camadas OSI
  Camada 4 para TCP
  Camada 7 para Http e Https
- Tipos de ELB:
  Internet-Facing: Padão onde a requisição vem da internet para uma instância EC2
  Internal-LoadBalancer: Usado para rotear o tráfico em uma Subnet Interna
  HTTPS-LoadBalancer: Criptografia SSL entre cliente e ELB assim como ELB e EC2 (Não suporte SNI'Server Name Indication' suporta SAN'Subject Alternative Name')
- Opções de configuração do ELB
  Idle Connection Timeout: Tempo de espera por dados, até cancelar as conexões ativas (padrao 60 segundos). Para Http sugere usar opção keep-alive (reaproveita conexão) no S.O.
  Cross-Zone: Habilita balanceamento de pacotes entre as zonas de disponibilidades
  Connection Draining: o ELB para de enviar novas conexões para servidores problemáticos, enquanto estes permanecem com algumas conexões ainda abertas. pode configurar o tempo máximo p/ q as conexões abertas no server problemático permaneçam ativas. Após isso serão forçadas a desregistrar.
  Proxy Protocol: Insere no cabeçalho do pacote, informações de ip de origem e destino e portas. Essa informação é repassada para os servidores de back-end
  Sticky Sessions: Por padrão as requisições são enviadas para o Back-end de menor carga. Habilitando o Stick session, ele define que as conexões de um usuário sempre serão direcionadas para o mesmo back-end. ELB cria um cookie chamado AWSELB para mapear as sessões.
  Health Check: Verifica a saúde da instância EC2. OK é InService, fora é OutOfService. Pode ser um ping, uma tentativa de conexão ou uma pagina q será checada periodicamente

- **CloudWhatch**
  Tipo básico, sem cobrança adicional, envia informações a cada 5 minutos
  Tipo detalhado, com cobrança adicional, envia dados a cada 1 minuto, permitindo agregação de dados
  Não agrega dados de diferentes regiões
  Scripts CloudWhatch Permite enviar dados de metricas (PUT), usado para consumo de memória e disco de EC2 por exemplo
  CloudWhatch Logs permite monitorar logs de instâncias EC2, AWS CloudTrail e outros recursos.
    Pode monitorar erros de acordo com o limite estabelecido e enviar email
    Existe um agent para enviar os logs de SO direto para o CloudWhatch Logs.

Limites:

5.0 armes por conta

Dados de metricas: 15 meses

Envio de email SNS: 1000 por email para free

Outros limites no guia de desenvolvimento (http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch_limits.html)

- **Auto Scaling**
- Limites:
  20 instâncias por região
  100 launch configuration por região
  Comando para verificar os limites do Auto Scaling: 'aws autoscaling describe-account-limits'
- Planos de uso:
  Manter as instancias: Se uma cai, outra é iniciada automaticamente
  Manual: Definir quantidade mínima e máxima, assim o Auto Scaling group inicia ou finaliza as instâncias conforme definido
  Escalonamento agendado: Com data fixa
  Dinamicamente escalonado: Definir parâmetros de controle para o escalonamento automático das instâncias
- Parâmetros de configuração:
  Launch configuration: Template para criação de novas instâncias, como AMI e tipo de instância
  Auto Scaling group: Informa número de instâncias máximas, mínimas e designadas para rodar. Pode conter ec2 do tipo on-demand ou spot, mas não ambas
  Scaling policy: Politica de escalonamento, ligada aos alarmes do Cloudwatch.

  Boas práticas para escalar rapidamente para cima e lentamente para baixo.

  Questão do preço do Ec2 por hora cheia

CAPÍTULO 6 - AWS Identity and Access Management; IAM

- IAM é autenticação para a infraestrutura aws, não é um serviço de armazenamento de autorização para aplicações. Nas aplicações é necessário usar o Active Directory (Aws Directory no aws) ou mesmo o Cognito (principalmente para Mobile apps)
- 'Princial': Nome da entidade que permite acessar os recursos da AWS. São de 3 tipos:
  'root user': Acesso criado quando se ativa o AWS na primeira vez. Tem acesso completo e deve ser usado somente para criar o primeiro IAM User, após isso guardar o acesso em um local seguro
  'IAM user': Usuários para acesso aos recursos da AWS, também podem ser contas de aplicações. Boas práticas indicam usar a configuração de menor privilégio possível para as contas
  'Roles/temporary security Tokens': Roles são usadas para garantir privilégios específicos para atores específicos com uma duração específica. Estes atores podem ser autenticados pelo AWS ou algum sistema confiável externo. Quando um destes atores assumem uma função, o AWS provê um token temporário através do AWS Security Token Services (STS), assim o ator pode usar os recursos da AWS. O range de tempo de vida do token temporário varia de 15 minutos a 36 horas. Estes roles são divididos de acordo com os casos de uso:
    'Amazon EC2 Roles': Provê permissões as aplicações rodando em uma instância Aws EC2 para acesso a recursos da AWS. Uma aplicação instalada em um EC2 pode usar uma IAM User para acessar AWS Resources, porém para isso ela teria que armazenar as informações de usuário e senha do IAM User, para evitar isso, pode ser criada uma role e vinculada a uma instância EC2. Ao ser executada, ela 'pega' um token temporário.
    'Cross-Account Access': Provê permissões para usuários de outras contas AWS (root accounts)
    'Federation': Provê permissões para usuários autenticados por um sistema externo confiável. Subdividido em dois tipos de federações externas:
      Como Facebook ou Google, através do OpenID Connect (OIDC)
      Para modelos como Active Directory ou LDAP, usando o SAML 2.0
- 'Autentichation': São as formas de autenticação, divididas em 3 tipos:
  'Username/password': É o modo que autenticação usado na console, interface humana. Permite políticas de complexidade de senha e expiração
  'Access key': Combina Key ID e Access secret key. Usado para conexões de programas via API, como o Command Line e SDKs
  'Access key/Session Token': Usado nas interações através das roles, usando as duas partes do Access Key fornecidos de forma temporária e um 'session token'
- Uma IAM User criada, por padrão não tem password definido e nem Access Key, isto deve ser definido pelo administrador.

- . 'Authorization': Após autenticado, o 'principal' se depara com o que pode ou não acessar. Nesta parte entra a authorization, através das policies que devem ser associadas aos 'principals', diretamente no menu User Policy ou Group policy, ou diretamente em 'Managed Policies' na tela de configuração do IAM.

'Policies': São documentos JSON que definem permissões para manipular e acessar recursos da AWS. Pode conter uma ou mais permissões, em cada permissão. Deve conter:

'Effect': Se permite ou nega (allow or denny)

'Service': Para quais serviços a permissão se aplica.

'Resource': O valor do recurso específico da infraestrutura aws específica para a qual se aplica a permissão. Usado através do Amazon Resource Name (ARN) Como formato básico: "arn:aws:service:region:account-id:[resourcetype:]resource" Permitido wildcards para alguns serviços.

'Action': Especifica a ação em um recurso, por exemplo leitura (Read*) ou escrita (Write*)

'Condition': Opcional. Mais uma etapa de possível restrição para obter acesso ao recurso, como um range de tempo ou mesmo um range específico de IP

----- Exemplo de uma policy que será associada a um usuário ---

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Sid": "Colocar_o_prefixo_específico_do_usuário",
       "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::examplebucket/Alice/*"
      }
   ]
}
```

- Uma Role pode ter uma policy associada.

- Outras características do IAM:

MFA: Multi autenticação

Exercícios

1 Criar um grupo, associar a política de Administrador Full nele

2 Criar um usuário e colocar no grupo Administrador, habilitar MFA nele

3 Criar uma Role para pemitir uma EC2 acessar um bucket s3

Criar a role do tipo EC2
Associar a policy AmazonS3ReadOnlyAccess a Role criada
Associar a Role na instância EC2
Fazer o teste de acesso

4 Criar uma policy de negação de acesso ao S3, associar ao grupo administrator para identificar erros de conflitos de policy

Criação de policy: Effect: denny; Aws Service: Amazon S3; Actions:*; ARN:*

CAPÍTULO 7 - Databases na AWS: RDS, RedShift e DynamoDB

- Bancos de dados relacionais são chamados de RDS 'Amazon Relational Database Service' na Aws, divididos em 2 tipos
  OLTP: Tradicionais, acessados a todo o momento com SQLs simples
  OLAP: Para grande volume de dados, acessado de forma bem menos frequente com SQLs complexos. (Redshift)
- Data Warehousing: Repositório central de dados com uma ou mais origens
- NoSql é mais simples e permite escalar horizontalmente

RDS

- RDS: Suporta MySQL, PostgreSQL, MariaDB, Oracle, SQL Server e Amazon Aurora. Define as configurações em DBParameterGroup: Definições de opções do banco de dados ou de um grupo de banco de dados
  DBOptionGroup: Definições de features (características) de um banco de dados, por exemplo configuração de Memcached para o mysql
- AWS Database Migration pode ser usado para migração de bancos de dados de diferentes engines
- Limite de 40 instâncias de banco por conta(10 para Sql Server). Nos SGBD gratuitos não a limite de banco de dados por engine, nos pagos, 1 bd para Oracle e 30 para SQL Server.
- Escala armazenamento, sem precisar ficar com sessão indisponível. No SQL Server não é possível aumentar armazenamento por limitação do Windows Server.
- Bancos de dados que precisam de licenças podem usá-las de 2 tipos:
  Licença incluída: incluída no preço do produto na AWS
  Usando sua própria licença (BYOL): Você inclui sua licença do produto
- Sobre o Amazon Aurora DB:
  Compatível com Mysql
  Incrementa velocidade até 5x maior q o Mysql
  Ao criar a instância, deve ser criado um Cluster
  Dois tipos de instâncias:
  'Primary Instance': Instância principal, suporta tanto leitura quanto escrita, dados são sempre modificados na Primary instance
  'Amazon Aurora Replica': Instância secundária, suporta operações de leitura. Pode haver até 15 replicas adicionais a instância primária.
- Opções de armazenamento (Storages):
  Magnetic: Baixo nível de IO, discos magneticos, baixo custo
  General Purpose (SSD): Também chamado de gp2, boa performance e preço para pequenos e médios databases
  Provisioned IOPS (SSD): Para cargas de trabalho intensivas, ajustando o IOPS necessário.

- Backup e restore usando RDS:
  RPO - Recovery Point Objective: É o máximo período em que os dados podem ser perdidos em caso de falha ou incidente. Tempo entre um backup e outro por exemplo.
  RTO - Recovery Time Objective: É o tempo que leva a restauração dos dados, até a reativação e disponibilização do serviço.
- Backup automático: É feito um snapshot diário da instância. Utiliza IO das instâncias e pode apresentar lentidão. Configurações possíveis:
  'backup retention period': Manter os backups até x dias
  'Maintenance window': Janela de manutenção em que o backup é feito
  'Manual DB Snapshot': Faz o snapshot a qualquer momento, não é deletado após o período de retenção.
- Recovery:
  Cria uma nova instância na restauração com 'parameter group' e 'security group' padrão, deve alterar se necessário após a restauração.
  Ao restaurar o banco utilizando um backup automático, o RDS utiliza o Snapshot e os logs de transações para restaurar o banco até os últimos 5 minutos
- Multi AZ: Usada para replicação dos dados entre dias instâncias RDS. O cliente acessa por um nome, em caso de falha no Database Master, a AWS automaticamente inicia as conexões para o secundário que se encontra em outra zona de disponibilidade, colocando o master em standby. Isto também pode ser feito de maneira manual.
- Escalando conforme a necessidade:
- Scaling Vertical: No RDS podemos modificar a instância escolhendo outra configuração de equipamento que melhor atenda a demanda. As novas configurações entram em vigor na próxima janela de manutenção. O Storage também pode ser aumentado ou ter o seu tipo alterado com exceção do Microsoft SQL Server.
- Scaling horizontal: Este escalonamento é mais complicado no RDS. Pode-se utilizar mais de uma instância, porém necessita de uma nova camada lógica da aplicação. NoSql escala horizontalmente sem problemas.
  Outra técnica é o uso de replicas de leitura (suportada pelos SGBDs libres + Aurora), o RDS suporta replicas de leitura que são criadas a partir do principal que escreve os dados de forma assíncrona. Usado por exemplo quando a requisição de leitura é bem maior que a escrita e pode ser pulverizada em várias instâncias. Precisa-se ativar backups automáticos na instância de banco de dados antes de criar réplicas de leitura.
  Limite de 15 replicas de leitura, para uma instância de banco de dados
  Pode ser criadas replicas de banco de dados de leitura inclusive em outras regiões.
- Segurança no RDS:
- Proteger o acesso ao recurso através do IAM
- Usar Private Subnet
- Criar um DBSubnetGroup e configurar quais subnets são disponíveis para o deploy do RDS

- Restringir o acesso usando Network ACL e SG permitindo somente o acesso a partir de um range de IPs de origem.
- Usar usuários de acesso ao banco com senhas fortes
- Usar criptografia

Amazon Redshift:

- Banco de dados para OLAP
- Componente principal é o Cluster que é composto pelo 'leader node' e um ou mais 'computer nodes'. O acesso se dá para o leader node.
- Dividido em 6 tipos de nós, subdivididos em duas categorias, Dense compute(326TB) e Dense Storage(2Pb de capacidade)
- O processamento é dividido entre os nós do cluster que podem ser adicionados posteriormente
- Estratégias de distribuição de dados entre os nós. Existem 3 tipos que são definidos no momento da criação da tabela, eles influenciam diretamente a performance.
  'EVEN distribution': Padrão, Distribui de forma uniforme independente do tipo de dados
  'KEY distribution': As linhas são distribuídas conforme os valores em uma coluna. incrementando performance para joins
  'ALL distribution': Uma cópia completa das entradas da tabela é distribuída em todos os nós.
- Carregando dados: Usar preferencialmente o comando COPY, que carrega dados em uma tabela de maneira muito mais eficiente. Utilizar arquivos de texto planos no S3 para carregar estes dados é o mais rápido e eficiente
- Comandos importantes:
  'VACUUM': Reorganiza os dados e libera o espaço dos arquivos deletados
  'ANALYZE': Atualiza tabela de estatísticas
  'UNLOAD': Exporta os dados para fora do Redshift em arquivos de texto no S3
- Comando SELECT é suportado, pode-se analisar a performance de uma query ou do cluster no cloudwatch para otimizá-la
- Usar o WLM (Workload Management) para gerenciar uma fila e priorizar quando usar o Redshift com várias queries
- Suporta Snapshot do cluster, tanto manual como automático
- Para Segurança de acesso aos dados, utiliza o mesmo plano do RDS
- Possibilidade de criação de usuários para "partes" específicas dentro do Redshift, como schemas, tables, independente do IAM.

DynamoDB

- Ferramenta NoSql da AWS
- Vantagem em relação ao RDS(sql)
  Flexibilidade em relação ao schema do banco
  Necessidade apenas de uma chave primária
  Novos atributos podem ser criados para itens conforme a necessidade
  Deve ser escolhido o tipo do 'primary key', a partir de 2 tipos possíveis:
  'partition key': formato por um atributo
  'partition and sort key' feito pelo partition key + sort key, assim pode haver mais de um item com o mesmo partition key.

- Selecionar a capacidade estimada na criação da tabela. Pode ser alterado depois com o comando UpdateTable
- Tipos de index secundários:
  Global Secondary Index (multiplos)
  Local Secondary Index (apenas 1 local index)
- Comandos para manipulação de Itens no DynamoDB através de API
  De Escrita: PutItem, UPdateItem e DeleteItem
  Batch(escrita e leitura em lotes): BatchGetItem, BatchWriteItem (Até 25 itens em uma única operação)
  Leitura: GetItem
- Tipos de consistência: Como o DynamoDB é um sistema distribuído, algumas vezes a gravação de uma atualização ainda não foi atualizada em todos os pontos, para isso, existem dois tipos de consistências:
  Eventually Consistents reads: Pode não retornar o ultimo valor atualizado, após uma nova pesquisa o último valor é retornado
  Strongly Consistent reads: Verifica, compara e exibe a última modificação armazenada em todas as pontas da tabela distribuída. O custo de computação e o tempo de resposta é maior. Parâmetro deve ser especificado na requisição.
- Pesquisa de itens: Query e Scan
  Query: precisa da chave primária e exibe de maneira mais rápida e direta o item
  scan: Scaneia toda a tabela pesquisando pelo item, é bem mais onerosa.
- Partitions: Uma tabela pode ser dividida em diferentes partições. Cada partição suporta 10gb de dados, 3 mil capacidade de leitura e 1 mil capacidade de escrita. Para melhorar o desempenho, deve-se dividir a tabela em diferentes partições
- Segurança: Integrada com o IAM e com policies para usar condições de acesso restrito a itens ou atributos individuais. Todas as operações precisam ser autenticadas com um usuário válido.
- DynamoDB Streams: Quando ativado exibe todas as modificações que as tabelas sofreram em um período de até 24horas. Pode servir para outro programa capturar estas informações, ou mesmo fazer uma replicação dos dados. É exibida em forma sequencial e dividida em grupos chamados de shards.

Exercícios

1 Criar um RDS Mysql com Multi-AZ habilitado

2 Testar Multi-AZ reiniciando a instancia e observando o local de disponibilidade

3 Criar replica de leitura do RDS Mysql

4 Criar uma tabela DynamoDB

Tipo de partition key: String para userID
Criar novos itens na tabela, em userID colocar U01 e outro atributo chamado name com o valor Joao
Usar o Scan para pesquisar os itens da tabela

5 Criar um cluster de um nó do RedShift, criar uma tabela usando o SQL Workbench e carregar dados usando o comando COPY.

CAPÍTULO 8 - SQS, SWF e SNS

- SQS é um buffer entre os componentes da aplicação que recebem dados e outros componentes que processam os dados no sistema. SQS tradicional não garante FIFO, SQS do tipo FIFO sim.
- Suporta até 120.000 mensagens (in flight) ao mesmo tempo
- As filas do tipo FIFO suportam até 300 TPS (Transações por segundo)
- Trabalha com duas configurações para visibilidade da mensagem na fila: Delay Queues e Visibility Timeouts.
  Delay Queues: A mensagem fica invisível logo após ser colocada em fila até 'x' segundos definidos neste parâmetro. Padrão é 0
  Visibility Timeouts: A mensagem fica 'x'segundos invisível a partir da primeira requisição. Suporta no máximo 12 horas de visibility Timeouts. Padrão 30 segundos
- Identificadores: Queue URLs, mesage IDs e receipt handles
  Queue URLs: Gerado na criação da fila, o nome dado a fila + alguns atributos definidos pela aws
  Message IDs: Reportado pelo SQS no momento do envio da mensagem para a fila, usado para controle da aplicação de envio
  Receipt handles: Quando uma mensagem é recebida da fila, o SQS gera e envia este identificador. É necessário para excluir a mensagem do SQS
- Suporta até 10 atributos nas mensagens, que podem ser processados sem precisar ler a mensagem em si, economizando processamento.
- 'Suporta dead Letter queues' que é uma fila na qual se destinam as mensagem com erro de processamento, para identificação posterior do motivo do erro.
- IAM controla interação/permissão das identidades de uma mesma conta Amazon nas filas, mas Amazon SQS Access Control, permite de forma efetiva a interação entre diferentes contas da AWS com as filas. A utilização de IAM Roles não é efetiva nestes casos.
- SQS somente retorna OK no envio da mensagem (SendMessage) quando ela é gravada, garantindo a segurança.
- A ‘sondagem longa’ geralmente é o melhor modo de verificar as mensagem na fila. Ela somente retorna quando existem dados na fila.
- 'Long polling timeout' (Timeout de sondagem longa) deve ser configurado se a aplicação usa loop para verificar novas mensagnes, isto diminui a requisição de processamento do serviço. Período máximo de timeout: 20 Segundos
- Período máximo para retenção da mensagem: 14 dias. Padrão é: 4 dias
- **SWF** É um coordenador de ações durante o fluxo de trabalho utilizando componentes distribuídos. Abaixo algumas definições:
- Workers, starters e deciders; workflow; workflow history;
  actors; Podem ser trabalhadores ativos, iniciadores de workflow ou deciders
  tasks;
  domains; Recurso de um workflow. Workflow em diferentes domínios não interagem
  object identifiers; Identificadores dos objetos. Podem ser: Workflow types, activity type e activity tasks.

task list; workflow execution closure; long pooling

- **Amazon Simple Notification Service (SNS)**
- Serve para operar e enviar mensagens
- Dois tipos de clientes: publisher e subscribers.
- O Publisher envia a mensagem para o 'sns topic' que se encarrega de entregar para os subscribers, sejam eles Lambda, SQS, Email, SMS, etc.
- Desenho: publisher → Topic → (envia para todos os subscribers configurados no tópico)
- Alguns casos de uso: Mensagem de alerta para email, envio de mensagem para mais de uma fila simultaneamente, envio de notificações para aplicação mobile.
- Protocolos usados: http/s, SMS, email, email-JSON, SQS e Lambda

Exercícios

1 Criar um Tópico SNS, adicionando um subscription do tipo E-mail e publicando uma mensagem no tópico.

2 Criar uma fila em SQS

60 segundos de default visibility
5 minutos de periodo de retenção da mensagem

3 Associar a fila SQS a um Tópico SNS e enviar uma mensagem

Consultar no SQS se a mensagem está disponível lá

CAPÍTULO 9 - Route53

- **Funcionamento básico do DNS, nomes para IPs**
- TLD são os domínios Top Level como .com ou .org
- Registrar é uma autoridade para incluir nomes de domínios diretamente nos TLDs
- Cada domínio tem que ser único e deve ser registrado no NIC de cada região
- Tipos de registros no DNS:
  A: Vinculado a IPv4
  AAAA: Vinculado a IPV6
  CNAME: Vinculado a nome de dominio ou Alias Name
  MX: Servidor de EMail
  NS: Direcionam a pesquisa de um subdomino para outro DNS, tambem usados no TLD
  PTR: É o A reverso, de IP para nome
  SOA: Informações do DNS sobre o domínio
  SPF: Para evitar o uso de SPAM, verifica se o dominio do enviador de Email pertence ao dono
  TXT: Descritivo
  SRV: Usado para encaminhar a resolução de nomes por serviços específicos
- **Amazon Route 53**
- Limite de 50 domínios, 500 zonas hospedadas e 10,000 conjunto de registros de recursos por zona hospedada
- Conecta requisições de nomes com serviços rodando na infra estrutura da Amazon, como EC2, ELB
- Pode direcionar requisições para fora da AWS
- Organizado por Hosted zones (Zonas hospedadas), similar aos hosted files do DNS, divididos por domínios.
- Uma zona hospedada é um conceito do Amazon Route 53. Uma zona hospedada é análoga a um arquivo tradicional da zona do DNS; ela representa um conjunto de registros que podem ser gerenciados juntos, pertencendo a um único nome de domínio pai
- Os Hosted Zones podem ser do tipo pública e privada, onde o público resolve as consultas externas e as privadas resolvem das VPC associadas a ela.
- Ao usar o Route53, podemos configurar diferentes políticas de roteamento, que segue:
  Simple: Classico, encaminha direto
  Weighted(peso): Para direcionar um percentual de trafego para um determinado recurso. Deve se atribuir mais de um recurso e configurar um ID único e o peso de cada um. O sistema vai então encaminhar o tráfego conforme o referido peso do recurso
  Latency-basead: Encaminha o tráfego para a menor latência, a partir do usuário. O Route53 testa qual a menor latencia da origem da query até o recurso de destino. Usado para encaminhar para a rede mais rápida. Mais de um ELB em diferentes regiões.
  Failover: Caso um recurso fique indisponível ele encaminha o trafego para outro recurso. Faz verificação por HTTP, HTTPS e TCP.
  Geolocation: Faz o roteamento de acordo com a localização do usuário final.

Exercícios

1 Criar uma nova zona relacionada ao domínio

*Atentar para SOA record e NS (Name Servers) q será usado para registrar o domínio no registars

2 Apontar o registro de DNS para um ec2 WebServer

3 Preparar 2 ec2 webservers em 2 zonas diferentes. Associar estes webservers a um LB cada e criar um apontamento DNS do tipo Weighted para os 2 LBS com o mesmo peso

4 Criar um HOstedZone do tipo interno e configurar a VPC para resolver por este dominio

CAPÍTULO 10 - Amazon ElastiCache

- Solução para utilização de cache em aplicações, principalmente WEB para armazenar dados frequentemente utilizados em memória.
- Funciona como uma cache entre as pesquisas do banco de dados e a aplicação
- Não suporta acesso direto por IP de internet (Ip público)
- Suporta Security Groups e Network ACL, assim como IAM Policies
- Compatível com Redis e Memcached
  MemcacheD: Mais simples, suporta múltiplos nodes, no máximo 20 NODES, inclusive para suportar falhas. A biblioteca ElasticCache suporta "auto discovery" (no cliente) permitindo encontrar novos nodes conforme são incluídos.
  Redis: Mais complexo, suporta um node de gravação/leitura e mais 5 de leitura chamado "Replication Group". Suporte MultiAZ.
- O Redis quando aplicado o uso com cluster, pode ser configurado com 'shards'(fragmentos), cada shard pode tem um número máximo de 5 nodes, mas somente um deles será de gravação/leitura, os demais de leitura. podem haver até 15 shards em um cluster Redis.
- Ambos os tipos de engine suportadas permitem a mudança do tipo de host conforme o aumento da necessidade de hardware. (Vertical Scaling)
- A replicação entre os nós é assíncrona e pode demorar algum tempo para que o outro node tenha acesso a informação no cache
- Backups: O Memcached não suporta backup ou snapshot, já o Redis suporta snapshots nos mesmos moldes do RDS.

Exercícios

1 Criar um Amazon ElasticCache usando a engine Memcached

2 Expandir o cluster MemcacheD adicionando mais um host

*Clicar no nome do memcached criado antes e ir no botão 'em add node'

3 Criar um Amazon ElasticCache usando a engine Redis e configurar o 'Replication Group'

CAPÍTULO 11 - Outros serviços chave

- Podem ser subdivididos em: *Storage and content Delivery; Security; Analytics e DevOps*
- **Storage and Content Delivery**
- Amazon CloudFront: É um CDN (Content Delivery Network) Global. Serve como uma espécie de cache para servidores Web e outros conteúdos. Para ser efetivo pode ser usado com a configuração de Geolocalização do Route53. Desenvolvido para trabalhar em conjunto com os seguintes serviços como origem: S3, Ec2, ELB.
  Também funciona como Webserver, além dos protocolos HTTP(s) e Streaming de media.
  Para ativá-lo atentar para 3 configurações:
    Distribuitons: O nome para acesso do conteúdo, que será redirecionado ao CloudFront
    Origins: O serviço e servidor de origem (pode ser o S3, Ec2, ELB ou endereço de website)
    Cache control: Opções de período de expiração do objeto no CloudFront, o padrão é 24horas
  Pode usar a chamada de API "invalidation" para expirar os objetos e fazer com que o cloudfront tenha que buscar a nova versão do site.
  Uso avançado do Cloudfront:
    As opções de "cache behavior" ou comportamento do cache habilitam o uso avançado do Cloudfront, como por exemplo, conforme a requisição de extenção (.jpg ou .php) o Cloudfront encaminha para um serviço específico (s3 ou ec2 por exemplo). Pode ser configurada uma listagem com as regras de encaminhamento e cache, a prioridade de execução é executada de cima para baixo, a ultima geralmente é * para definir qualquer arquivo ou regra.
  Alguns casos em que não é aconselhado o uso do CDN: Muitas requisições a partir de um mesmo lugar; Requisições através de uma VPN
- AWS Storage Gateway: É um serviço que conecta o datacenter local com os serviços de armazenamento S3 e Glacier da Amazon. Ele é na verdade uma maquina virtual para ser instalado no host de infra-estrutura local e expõe uma interface iSCSI, permitindo três configurações distintas:
  "Gateway-cached Volumes": Expande o storage local para o S3, onde os dados lidos recentemente são mantidos no storage local para prover baixa latência. Cada volume é limitado a 32TB, um único gateway pode suportar 32 volumes, totalizando 1Pb de dados. Todos os dados são criptografados e é permitido o uso de snapshots, "Gateway-stored Volumes": Armazena todos os dados no Storage Gateway local e de forma assíncrona envia os dados para o S3. Cada volume suporta 16TB com o máximo de 32 volumes totalizando 512TB, permite snapshots, Gateway Virtual Type libraries (VTL)": Uma espécie de fita dat de backup virtual, para ser compatível com softwares de backups. Após armazenar o backup no Glacier ele arquiva os dados em uma VTS (Especie de prateleira de fitas)

- **Security**
- 'Aws Directory Service': Serviço de diretório da Amazon. Pode escolher Directory Service dos Microsoft Active Directory (o AD), AD simple e AD Connector. Por definição já resilientes a falha, utilizando também snapshots e multi zonas.
  Microsoft Active Directory : Mesmas funcionalidades do AD da Microsoft e mais facilidade na integração com serviços da AWS
  AD simple: Compatível com Microsoft AD, utiliza o Samba 4, ou seja as funcionalidades básicas de um Active Directory da Microsoft. Não é permitido fazer as relações de segurança entre o Simple AD e o Microsoft Active Directory
  AD Connector: É uma maneira de conectar o Microsoft Active Directory on-premises a nuvem AWS sem precisar utilizar a replicação do AD, funciona como um serviço de proxy, onde as requisições de usuário/senha são encaminhadas para o AD on-premisses

- 'AWS Key Management Service': O (KMS) é um serviço gerenciado que facilita a criação e o controle das chaves de criptografia usadas para criptografar dados criptográficos simétricos. Fornecendo a gerencia sobre o uso das chaves criptográficas.
  Customer Managed Keys: Usa a chave 'Customer Master Key (CMK)'. Pode ser usado para encriptar 'data keys' gerados
  Data Keys: Usado para encriptar grande quantidade de dados usando sua própria aplicação fora do AWS KMS. Utiliza plaintext e uma das melhores práticas é excluir o conteúdo da memória após a utilização.
  Envelope Encryption: AWS KMS usa isto para proteger os dados.
  Encryption Context: É uma opção não obrigatória nas operações de criptografia. Esta opção é exibida no log e pode ser auditada.
  'AWS CloudHSM': Faz uso do Hardware Security Modules no Aws Cloud. Usado para aplicações ou corporações que precisam, seja por regulamento ou requerimentos de complience, de um hardware appliance para criptografia. É designado para armazenar de forma segura chaves criptográficas. Usado por exemplo por empresas de cartão de crédito.

- 'AWS CloudTrail': Ele grava todas as chamadas de API feitas para uma conta AWS. Ele armazena informações importantes como nome da API, identidade de quem chama, tempo de chamada da API, parâmetros de requisição e o que o AWS Service retornou.
  Ele armazena este log de chamadas em um arquivo de texto no Amazon S3, também pode ser configurado para entregar estes logs no CloudWatch Logs.
  Podem ser criados Trails que são configurações de atividades que serão relatadas como eventos na conta AWS. Eles são de 2 tipos:
    Trail aplicada a todas as regiões: Opção default
    Trail aplicada a uma região:
  Por padrão o CloudTrail entrega arquivos de log em torno de 15 minutos de uma chamada API.

Usado para auditoria de conformidade externa, Analisar acessos não autorizados a contas AWS já que analisa praticamente a conexão a todos os meios possíveis. Console, SDK, AWS CLi, etc.

- **Analytics**: Soluções analíticas e de BigData requerem um agrande quantidade de processamento e local para armazenar um grande volume de dados, mas sua característica básica é que utilizam este conjunto de requisitos de hardware por um pequeno espaço de tempo.
- Aws Kinesis: Solução para trabalhar com streaming de dados, ou seja análise de dados 'entrantes' em tempo real. Subdividido em três tipos:
  Amazon Kinesis Firehose: Habilitado para carregar um volume massivo de streaming de dados na AWS salvando do S3. Não é necessário escrever código, apenas configurar o destino dos dados. A origem pode ser gerada a partir de API da AWS, usando por exemplo o agente kinesis.
  Amazon Kinesis Streams: construir aplicações customizáveis para analises mais complexas de streaming de dados em tempo real. Para grande volume de dados, estes podem ser divididos em 'shards'(Fragmentos) para processar em instâncias EC2 separadas.
  Amazon Kinesis Analytics: Servico para analisar facilmente um streaming de dados real time utilizando um padrão SQL, além de escalar de modo automático quando necessário.
  Casos de uso incluem ingestão de dados; processamento em tempo real de uma grande massa de dados.
  Não confundir Kinesis com o AWS Data Pipeline que é mais indicado para trabalhos em lotes.
- Amazon Elastic MapReduce (Amazon EMR): Provê acesso ao Hadoop framework, facilitando a criação de clusters hadoop. Algumas opções no momento do cluster: Tipo de instância; Número de nós do cluster; Versão do hadoop, ferramentas adicionais como Hive, Pig, , Spark ou Presto
  2 Tipos de storage podem ser usador:
  Hadoop Distributed FileSystem (HDFS): Padrão, todos os dados são replicados por múltiplas instâncias. Os dados são persistentes quando a instancia é finalizada neste tipo de storage; Usado para 'persistent cluster'
  EMR File System (EMRFS): Utiliza o S3 para armazenar os dados. É ideal para clusters que são iniciado e finalizados mas que não podem ter seus dados perdidos; Usado para 'clusters transients'.
- Casos de uso incluem: Análise de logs semi estruturados com petabytes de informações, tarefas cientificas como o projeto genoma.
- AWS Data Pipeline: Para transferir informações entre dois sistemas diferentes ou o sistema on-premisses para nuvem AWS de maneira confiável, armazenando os dados em um S3, RDS, DynamoDB ou EMR. Você pode acessar seus dados com frequência onde eles estiverem armazenados, transformá-los e processá-los em escala
  Em resumo cria agendamentos para execução de tarefas para execuções em lote. Entre os casos de uso, se destacam tarefas executadas em partes onde por exemplo precisam verificar se os dados estão no S3, criar um cluster EMR, extrair estes dados no cluster, gerar os dados em um banco Redshift por exemplo.

- AWS Import/Export: Serve para transferir um grande volume de dados para a nuvem através de dispositivos de armazenamento convencionais, enviados para a Amazon por entrega comum. Compreende dois tipos:
  AWS Import/Export snowball: Utiliza um equipamento de armazenamento que é enviado pela Amazon para guardar os dados que serão remetidos de volta para a AWS. É protegido pela AWS KMS. Disponível em 2 tamanhos, 50TB e 80 TB. Ele oferece um console para gerenciamento de tarefas. Também é desenvolvido para não danificar os discos em caso de queda.
  AWS Import/Export disk: Utiliza o próprio equipamento do cliente para fazer a transferência dos dados através da rede interna de alta-velocidade da Amazon Web Service. Pode utilizar como destino o S3, EBS, Glacier

DevOps

- AWS OpsWorks: Serviço de gerenciamento de configuração par auxiliar na configuração e aplicação do Chef. Através dele pode ser estipulado quais pacotes serão instalados na instância, qual configuração abrangendo inclusive o próprio data center interno, usando apenas uma configuração para a nuvem ou para instâncias locais.
  Trabalha com 'stacks' que são um grupo de servidores ou serviços que devem ser inicializados ou mesmo criados em conjunto. Como o EC2 + RDS para uma aplicação por exemplo.
  'Layers' são as camadas de configuração de serviços com uma função específica. ELB ou database por exemplo.
  'app' São configurações relacionadas ao aplicativo e indicam por exemplo o repositório entre outros detalhes.
  OpsWorks tem integração com o CloudWhatch, criando as respectivas métricas.

- AWS CloudFormation: Permite controlar o ambiente através do versionamento de infra estrutura, semelhante a um software.
  Trabalha utilizando templates e stacks.
    Template é um arquivo de texto formato JSON, o Cloudformation usa estes templates para construir os AWS resources
  Pode-se utilizar o mesmo arquivo template mas com opções diferentes para criação diferenciada de infraestrutura.
  Pode-se criar um template para editar a infra atual, apenas modificando algum recurso, ele então trabalha com versões de templates.
  O CloudFormation sempre trabalha com templates salvos no S3
  Um dos casos de uso é replicar de maneira simples uma estrutura de uma região para outra, criar uma nova estrutura de testes semelhante a de operação entre outros.

- 'AWS Elastic Beanstalk': Uma maneira rápida de rodar as aplicações na AWS, apenas upando o código para a nuvem.
  O código da aplicação é colocado no S3, é criado então o 'environment' que pode rodar uma versão da aplicação. O Elastic Beanstalk provisiona automaticamente os recursos para a aplicação.
  'environment configuration' é a coleção de parâmetros e configurações que determinam como o recurso alocado vai operar. Em caso de update destas

configurações o Beanstalk aplica as modificações, dependendo ele destrói os recursos atuais e cria novos.
Provê suporte para várias linguagens de programas como java, Node.js, PHP, Python, Ruby e suporte a Webcontainers como Tomcat, Docker, etc.
Mesmo provendo a infraestrutura de maneira automática o consumidor do serviço tem acesso a inúmeras métricas de desempenho da aplicação, assim como também pode escolher várias opções de infra estrutura como tipo de instância, tipo de banco de dados, acesso a instâncias Ec2 para debug de erros, configurações de variáveis de ambiente, ajuste de configurações de auto scaling e de configurações do servidor de aplicação, entre outras.

- AWS TrustAdvisor: Utiliza o conhecimento na aplicação das melhores práticas de nuvem para analisar a infraestrutura e apontar recomendações para economizar dinheiro, melhorar a performance e aplicar alta disponibilidade.
  Provê as melhores práticas em 4 categorias: Otimização de custo, segurança, tolerância a falhas e performance. Apontando para cada caso ou que esta tudo OK, que existe uma recomendação de análise ou uma ação recomendada.
  Dentro das opção sem cobrança deste serviço, ele exibe a seguintes checagens:
    Limites do serviço: Se esta usando mais que 80% do limite do serviço
    Security Group com portas irrestritas
    IAM Use: Se estamos usando usuários IAM
    MFA na conta Root: Se usamos o MFA na conta de root
  Se o usuário conta com o suporte Business ou Enterprise, pode usar todos os recursos do Trusted Advisor, com mais de 50 checagens.

- AWS Config: Provê um inventário do uso de recursos na AWS, incluindo histórico de configurações, alterações. Serve para auditoria, análise de segurança, rastreamento de alteração de recursos, compliance e analise de erros.
  Por padrão ao ativar o serviço ele faz uma análise dos recursos em uso atualmente e começa a armazenar os históricos das modificações em todas as regiões compatíveis.
  Podem ser criados avisos para que se um determinado recurso ao ser criado, não obedecer as especificações de compliance configurados no AWS Config, uma notificação SNS é feita.

CAPÍTULO 12 - Segurança na AWS

- Modelo de segurança compartilhado. A AWS é responsável pela segurança da nuvem e o consumidor é responsável pela segurança na nuvem
- AWS tem inúmeras certificações de segurança em sua estrutura e isso pode ser estendido para seus consumidores a fim de também alcançarem certificações de segurança.
- Acesso a área física: Total controle do acesso as pessoas, detecção de incêndio, energia, climatização e destruição de dispositivos de armazenamento quando do descarte
- Gerenciamento de continuidade do negócio:
  Configuração de cluster sem 'cluster frio'. Serviços principais na configuração N+1
  Regiões separadas com zonas de disponibilidade conectadas entre si mas separadas
  Resposta a incidentes 24x7
  Comunicação e aprimoramento interno. Alerta externo como o Health Dashboard. Inscrição no suporte para comunicação direta
- Segurança de rede:
  Pontos de acesso a estrutura restritos a API e conexão HTTPs
  Mais de um provedor de internet com hardwares específicos
  Possibilidade de uso de VPC e VPN-Ipsec para acesso aos recursos no Cloud
- Monitoramento e proteção da rede para: DDoS, Man in the Middle, IP Spoofing, Packet Sniffing by Other Tenants.
  Port Scanning são proibidos de serem executados na estrutura da Amazon para fora. Scaneamentos de fora para a Amazon dependem da liberação do firewall pelo usuário. Para testes de scaneamento, deve ser informado a amazon através de uma submissão via Site
  Ataques ARP e trambém Snifers não funcionam no EC2 e VPC
- Características da segurança das contas
- AWS usa vários tipos de credenciais para autenticação como:
  Password; Para acesso pela console do Root e usuários IAM. Permite até 128 caracteres entre eles os especiais. Permite política de tipo de password
  Opção de MFA; Para acesso pela console do Root e usuários IAM e também ao API. Ao logar, deve fornecer um numero de 6 digitos através de um token ou aplicativo compatível; Usa o esquema 'Time-Based One-Time Password (TOTP)'
  Chaves de acesso; Digitalmente assinada, requeridas para AWS APIs (SDK e AwsCli)
  Par de chaves; Login SSH ao Amazon EC2, decodificar o password em maquinas Windows e também criar URLs privadas no CloudFront
  Certificados X.509; Digitalmente assinado para requisições SOAP ao AWS API, SSL ou HTTPs (Usado no S3)

- Permite o uso múltiplo de chaves para ocorrer o rotacionamento
- Segurança nos serviços em especial:
- EC2:
  Utiliza Xen modificado, host roda no Dom-0, instâncias no Dom-1
  O isolamento das instâncias em um host conta com firewall entre a placa física e as placas virtuais, Memória RAM sendo gravada com zeros após a liberação da VM e parcelas de storages isolados por usuário.
  Segurança de atualizações e firewall do Guest-OS estão por conta do usuário
  Pode ser usado o recurso de instance profile, para fazer um EC2 acessar recursos sem precisar informar ou usar chaves de acessos. Basta associar este profile a uma Iam Role.
- EBS: Tem os dados replicados dentro da mesma zona. Para máxima durabilidade o ideal é fazer um snapshot q será armazenado no S3
  Criptografia AES-256 no host do EC2

- Rede: ELB, nos pontos de segurança se destacam:
  Criptografia de dados fim-a-fim, ponto único de contato
  Nos logs do ELB são armazenados as informações de origem da requisição com IP e porta e o destino
- Rede VPC; Isolamento de rede. Necessário o atachamento de um Internet Gateway para o acesso a internet
  Com múltiplas subnets alguns ataques como MAC Spoofing e ARP Spoofing podem ser bloqueados. A tabela de roteamento define a conectividade das subnets.
  Security group define por padrão acesso a entrada somente de outas instancias do mesmo grupo e saída completa. Firewall a nível de Host.
  Network ACL: Firewall a nível de subnet, Statless, baseado no filtro de IPS
- Rede CloudFront: Não garante a durabilidade dos dados de cahce na ponta; Para fazer o acesso a recursos como S3 de maneira segura sem tornar o conteúdo público, pode ser usado o 'Origin Access Identity' e criada uma ACL para ele no S3.
  Para controlar quem pode fazer o download do coteúdo, pode ser usada URL assinada (signed-URL).
  Se configurado, pode encaminhar requisição HTTP para HTTPs, mas também aceita normalmente http.

- Storage: S3; Possibilidades de controle de acesso:
  IAM Policies: Controla acesso a usuários específicos IAM
  ACL: Controla acessos a grupo de usuários ou de outras AWS accounts (não IAM)
  Bucket Policies; Controla as permissões de maneira geral dentro de um bucket. Pode ser vinculada a usuários, grupos ou buckets. Usado para granular o acesso dos usuários a recursos do S3
  Query String Authentication: Permite que as informações de acesso sejam passadas por URL que deve ser assinada. Pode-se criar várias condicionantes, como data, ip de origem, entre outros.
  S3 pode usar criptografia pelo servidor (SSE), usando o AES-256. Ele criptografa os dados no momento do Upload e decriptografa no Download.

Suporta a ativação de LOGS contendo todas as informações que são feitas em um bucket

- Bandos de Dados
  DynamoDB: Acesso por IAM User; Permissões específicas internas no banco de dados e permissões detalhadas até de itens específicos através das IAM Policy's
  RDS: Utiliza Security Groups; Suporta VPC; DB Subnet Groups
  RDS utiliza as opções de multi-az, atualização de path na janela de manutenção, suporte a criptografia
  Redshift: Mesmo molde do RDS; usa Cluster e tenta manter a integridade dos dados com cópia nos HDs dos nodes. Usando o KMS ele usa a hierarquia Four-tier para encriptar os dados (master, cluster, database e data encryption keys).
  ElasticCache: Suporta Security Group

Aplicações

- SQS: Por padrão, somente quem cria a fila tem acesso a ela, mas pode ser definidas novas politicas de acesso com o 'SQS-generate Policy'. Os dados não são armazenados criptografados.
- SNS: Usuário autenticado tem acesso completo ao SNS porém para operações em tópicos precisam de acesso garantido por policy
- AWS EMR(hadoop): Permite conexão por SSH no master, por par de chaves
- Kinesis: Acesso controlado por IAM User ou IAM Role para acessos a partir do EC2. Endpoint para criptografia disponível apenas no endereço 'kinesis.us-east-1.amazonaws.com'

Serviços de Deploy e de gerenciamento

- IAM:
  Prática do menor privilégio possível
  Role: Como prática de segurança; permissão para executar um recurso específico.
  Usuários federados, ou non-aws users: São usuário como por exemplo do AD externo que podem ter permissões conforme as configurações de Roles
  Utilização do SAML2 para autenticação, como logins do Face, Amazon.
  Utilização para permitir que outro usuário de outra conta Amazon tenha acesso temporário a algum recurso da minha nuvem aws
  Recurso necessário a partir de um EC2 para outro recurso específico.
  Utilização de credenciais temporárias que fazem o rotacionamento são bem mais seguros.
  Cognito: Utiliza a autenticação do Google, Face, etc, para criar uma permissão de autenticação. Através das Roles pode-se criar permissões específicas.

Applications

AWS Workspaces: Usa o protocolo PC-over-IP (PCoIP) com suporte a compressão e criptografia.
Pode fazer a autenticação no AD, assim como é uma boa prática utilizar as polices de modificação pelo AD
Suporte MFA

CAPÍTULO 13 - Risco e conformidade

- O cliente que tem a responsabilidade de manter o controle de seu ambiente
- O modelo de responsabilidade compartilhada não se limita a segurança, mas também a instrumentos de controle de TI.
- AWS utiliza algumas formas para passar as informações relevantes a respeito do seu ambiente.
  Através das certificações homologadas por partes terceiras
  Através de material disponibilizado no site, blog
  Em alguns casos a partir do 'acordo de não divulgação' Non-Disclosure Agreements (NDAs)
- Para alcançar um modelo de conformidade forte (Strong Comliance) o cliente pode:
  Analisar todas a documentação fornecida pela AWS sobre suas certificações de ambiente
  Analisar e implantar controles objetivos para sua organização
  Fazer a analise por terceiros do seu ambiente
  Integrar com os controles da AWS
- AWS Riski and Compliance Program:
  É um programa da Amazon para auxiliar clientes a adorarem o programa de governança da AWS em sua organização.
  A AWS fornece com isso informações detalhadas em três áreas principais:
    Risk Management; Análise dos riscos e testes seguidos a vulnerabilidades no ambiente da AWS exposto. SOMENTE AWS
    Control Environment; Consiste em: Policies, processes e Controle de atividades
    Information Security; Publicação do material de segurança, como whitepaper por exemplo.

CAPÍTULO 14 - Melhores práticas na arquitetura

- Destaques para:
  Desenhar para falhar e não para se vai falhar;
  Tudo falha, todo o tempo
  Não coloque todos os ovos no mesmo sexto
  Redundâncias são múltiplos recursos para a mesma tarefa. Pode ser standby ou modo ativo.
- No modo Standby; a tarefa é assumida pelo recusro em standby, este processo se chama failover e geralmente demora algum tempo até o recurso secundário começar a receber as tarefas. Neste intervalo de tempo os recursos ficam indisponíveis. Comumente usado como nos bancos de dados.
- No modo de redundância ativa, as requisições são distribuídas a múltiplos recursos, quando um deles falha, o restante dos recursos saudáveis absorve o workload. Este modelo traz uma melhor ocupação do recurso e menor tempo de indisponibilidade.
  Componentes importantes para aplicar Failover são as zonas de disponibilidade e para implementar redundância é o ELB (Elastic Load Balancer)

- Implementar elasticidade:
  Elasticidade para acrescentar capacidade computacional ou para reduzir capacidade computacional de dois tipos:.
  Tipo vertical: Incrementa recurso em um recurso individual. Upgrade de um EC2 por exemplo. Necessita reiniciar o EC2, geralmente limitada, nem sempre tem um bom custo beneficio ou abordagem de alta disponibilidade. É muito fácil de ser implementada, principalmente em curto prazo.
  Tipo horizontal: Incrementa o número de recursos, por exemplo, alocando várias EC2, uma ótima opção para aplicações escaláveis de internet. Importante entender a característica da aplicação para adotar este modelo. Uma das características chaves é a arquitetura stateless e stateful
    stateless: Não necessita de informações da conexão anterior e não guarda informação de sessão. Ela pode escalar horizontalmente porque as requisições podem ser servidas por qualquer recurso alocado para ela. Neste caso pode ser utilizado o serviço de Auto Scaling. Nas aplicações que mesmo do tipo stateless precisam guardar alguma informação do usuário, como HTTP nos cookies por exemplo, elas nunca devem ser armazenadas no recurso local da nuvem; um dos recursos é utilizar o NoSQL(DynamoDB) para guardar estas informações.
    stateful: Aplicações Stateful não suportam escalar de forma horizontal. Bancos de dados são um exemplo.
  Possibilidade de criação de recursos através de automação de APIs

Possibilidade de uso de Bootstraps que são inicalizações de novas instâncias com AMIs pré configuradas e pré instaladas

- Levantar diferentes opções de armazenamento: Levar em consideração custo, performance e aspectos funcionais.
  Analisar as características de cada storage para adotar no melhor cenário.
  De maneira geral:
  S3: Alta capacidade e alta durabilidade dos objetos
  Glacier: Armazenamento por um tempo muito grande
  CloudFront: Entrege via rede global
  EBS: Block-storage, para bancos de dados internos no Ec2 por exemplo
  EFS: Para file systems, como compartilhamentos locais
  Demais bancos de dados como RDS, DynamoBD, Redshift e ElasticCache, cada um com sua função
  O tipo de workload vai de forma decisiva ajudar na escolha do tipo de armazenamento a ser escolhido.

- Construir segurança em todas as camadas
  Utilizar as ferramentas de controle que a AWS fornece para monitoramento constante
  Criptografar todos os dados em transito e localmente
  Utilizar as ferramentas que já conhece e que utiliza na estrutura tradicional para em conjunto implantar segurança na nuvem
  Utilizar em profundidade as ferramentas de segurança da AWS, como VPC, os Security groups, controle de rotinas, AWS Web Application Firewall, IAM com políticas granulares.
  Pensando no modelo de responsabilidade compartilhada e que o consumidor é responsável por seu ambiente, reduzir a sua própria administração reduz riscos e diminui o tempo de envolvimento, como a utilização de RDS ao invés de instalar um banco de dados em instância EC2
  Reduzir ou eliminar a necessidade de credenciais salvas em arquivos de configuração, utilizando por exemplo IAM roles. Também pode ser usado o Cognito para autenticação e também utilização de tokens temporários
  Garantir sempre o menor privilégio
  Segurança como código já que na AWS o ambiente pode ser replicado e analisado em código de programação, inclusive com controles de versões. Usando ferramentas como CloudFormation e AWS Config.
  Utilizar ferramentas de monitoramento de risco constante, como AWS Config Rules, Amazon Inspector e AWS Trust Advisor. Habilitando também as ferramentas de log como CloudTrail e CloudWatch logs
  Pode ser usado o AWS Lambda, Amazon ElasticSearch ou ferramentas de terceiros no marketplace para analisar estes logs e identificar alguma tentativa de acesso não autorizado, ou fora da conformidade.

- Pense paralelo

  Internalizar o conceito de paralelização e de repetição de processos de forma automática no cloud é simples.
  Utilizar operações de gravação ou de upload de dados de forma paralela faz com que se consiga muito mais velocidade de throughput, tornando muito mais rápido que uma leitura ou gravação sequencial.
  Utilizando a elasticidade e a paralelização, permite levantar um cluster com várias instâncias de computação, executando processos paralelos, gravando dados de forma paralela e finalizá-lo em minutos.

- Acoplamentos soltos

  Reduzir interdependência entre componentes de uma aplicação para não permitir que a falha de um comprometa o restante.
  Desenvolver componentes independentes e com a menor requisição de acoplamentos estreitos é a melhor prática e permite escalar da melhor maneira. Uma das maneiras é que os componentes se interconectem através de REST APIs (Integração assincrona)
  Utilizar a Amazon APIGateway é um recurso que pode facilitar o uso de APIs, com controle de acesso e monitoramento
  Neste modelo em que dois componentes de uma aplicação não interagem diretamente, pode ser usado o serviço de SQS por exemplo, como uma camada escalável e segura para transição de informações.

- Não tema restrições

  Se a nuvem não disponibiliza os recursos iguais ao que o consumidor conta no data center on premisse, existem métodos ou tecnologias para resolver estas questões.
  Por exemplo uma requisição de RDS, pode der minimizada através de réplicas de leitura ou adoção de cache de memória.
  A utilização de serviços gerenciados pela AWS (SQS, SNS, SES, Etc) pode ser uma das saídas para além de driblar restrições, utilizar os recursos de maneira mais racional e com o melhor custo benefício.

Exercícios

Criar o seguinte ambiente elástico e totalmente disponível.

1 Criar uma VPC 192.168.0.0/16

2 Anexar gateway de internet a VPC

3 Atualizar Main Route para 0.0.0.0 gateway de internet

4 Criar duas subnets publicas (192.168.1.0/24 e 192.168.3.0/24) em AZ diferentes

5 Criar NAT e vincular ao VPC

6 Criar Private Route Table - vincular 0.0.0.0/0 a NAT adicionada anteriormente

7 Criar duas subnets privadas (192.168.2.0/24 e 192.168.4.0/24) em AZ diferentes coincidindo com as azs das subnetes publicas, vincular a Private Route table as novas subnets privadas

8 Criar os security groups para cada aplicação, onde:

ELB-SG, para o load balancer de qualquer origrm a porta 80 liberada
Web-SG, para os servidores web, portas 80 do ELB e 22 de qualquer internet
RDS-SG, porta 3306 comente dos servidores Web

9 Criar RDS multi-az colocando nas subnets privadas criadas

10 Criar ELB encaminhando para porta 80 e configurando checagem de ping e site na 80

11 Criar autoScaling group, no minimo 2 instances, no userdata configuração web e adicionando o site .html iniciando nas subnets publicas

12 Adicionar o Load Balancer ao Auto Scalling group criado

13 Configurar Route53 para apontar como Alias para as requisições do domínio para o ELB