COMPUTAÇÃO EM NUVEM

AULA 3

Profª Ana Paula Costacurta

CONVERSA INICIAL

No Tema 1, falaremos sobre diferença entre máquina virtual e *container*, apresentando as principais características e quando utilizar cada um desses tipos. Conheceremos o conceito de *Docker* e microsserviços e como é a forma de implantação de um *container* no AWS, bem como quais ferramentas estão disponíveis da AWS.

No Tema 2, estudaremos o AWS *Identity and Access Management* (IAM) e principais termos. Veremos a infraestrutura do IAM com seus componentes. Aprenderemos como é realizado o Gerenciamento de Identidade e Acesso, como é o funcionamento das políticas com contas, usuários e grupos. Será apresentado a diferença entre usuários federados e funções.

No Tema 3, veremos a Identidade do IAM, conhecendo em detalhes como é a identificação de usuários, características importantes e ferramentas para gerenciamento dos grupos e como funciona e quais os tipos de funções.

No Tema 4, estudaremos o gerenciamento de acesso, aprendendo como criar, anexar as políticas à identidade e recursos e quais os tipos de políticas. Veremos em detalhes as políticas baseadas em identidade e em recursos. Conheceremos as noções básicas e como são utilizadas as políticas para permissões ou negações para cada serviço.

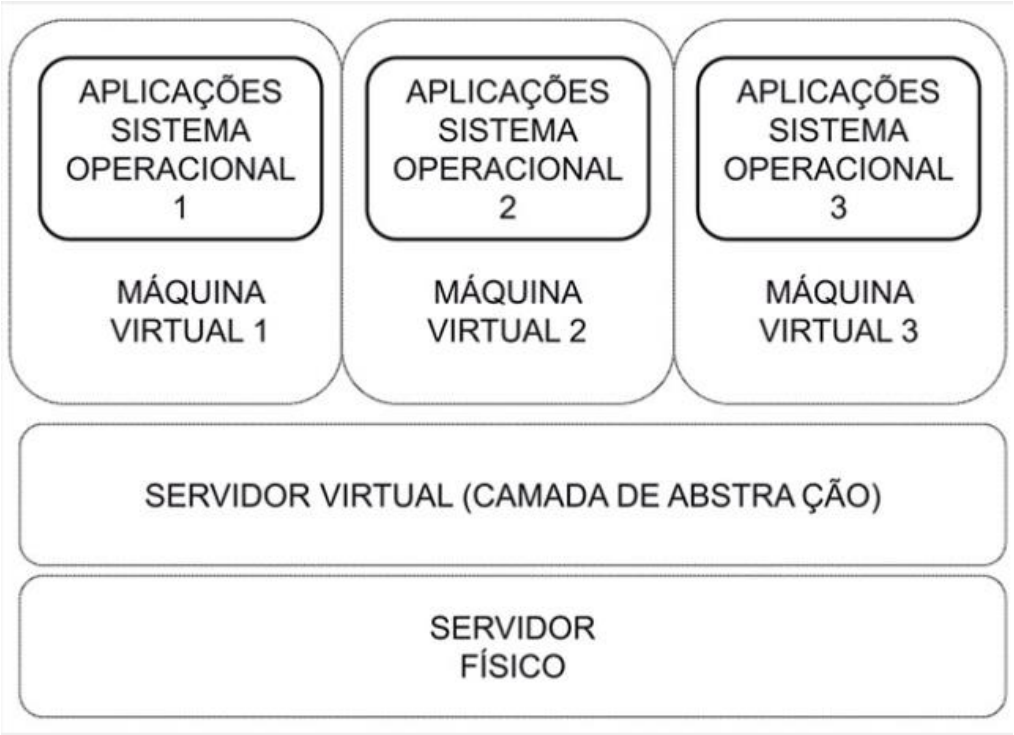
No Tema 5, conheceremos sobre o serviço *web Amazon Elastic Compute Cloud* (Amazon EC2) e os conceitos básicos. Estudaremos os tipos de instâncias de uso geral e instâncias otimizadas, sendo de quatro tipos: para computação, para memória, para computação acelerada e para armazenamento. Conheceremos conceitos básicos e diferenças dos tipos de imagens de Máquina da Amazon (AMIs).

TEMA 1 – MÁQUINA VIRTUAL E *CONTAINER*

As máquinas virtuais emulam o sistema operacional e todo o *hardware,* com isso acaba utilizando mais recursos e utilizando *container* os recursos são compartilhados. Com essa redução na utilização de recursos, torna possível executar mais *containers* comparado à execução com máquina virtual (Vitalino; Castro, 2018).

1.1 Máquina virtual

Figura 1 – Arquitetura da virtualização



Fonte: Veras, 2016.

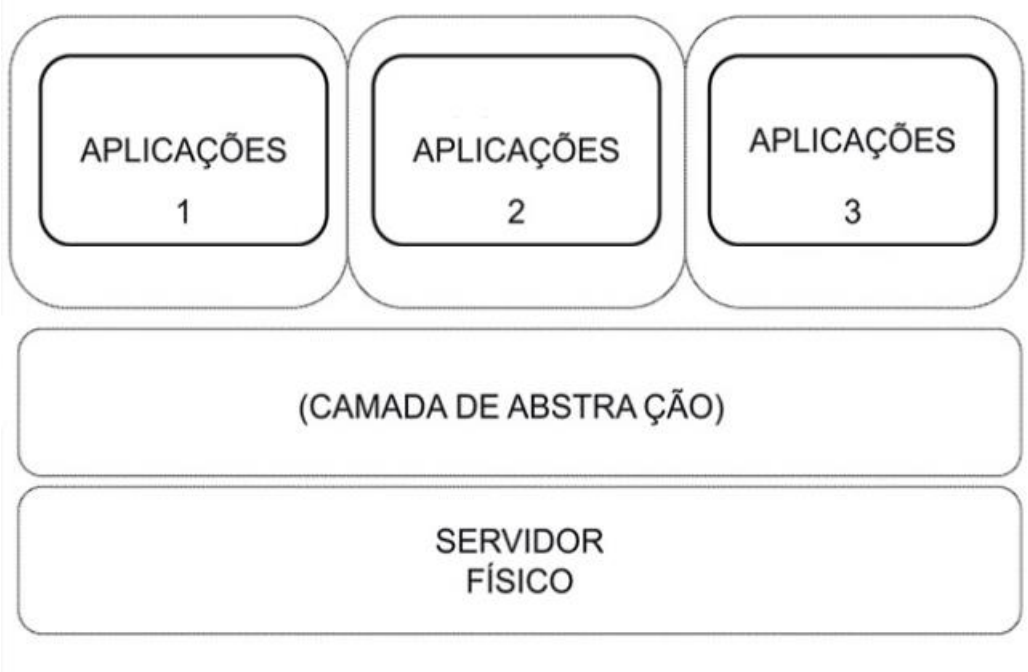
Segundo Veras (2016), a virtualização é uma camada entre o *hardware* e o *software* para proteger os recursos físicos do *hardware* diretamente pelo *software*, isolando a camada da aplicação e sistema operacional da camada do *hardware*. A virtualização mais comum é implementada com a utilização de um *software*. Na Figura 1, podemos ver a representação a arquitetura de virtualização.

1.2 *Container*

É o empacotamento da aplicação combinado às suas dependências, são similares à máquina virtual com a vantagem de serem mais leves e integradas com o sistema operacional do servidor ou máquina local que está rodando (Vitalino; Castro, 2018).

O propósito do *container* é emular apenas as aplicações e suas dependências para torna-la portátil, já que a máquina virtual realiza a emulação do sistema operacional dentro de outro sistema operacional. Na Figura 2, podemos visualizar a arquitetura do *container* (Vitalino; Castro, 2018).

Figura 2 – Arquitetura do *container*



Fonte: Vitalino; Castro, 2018.

Quando utilizamos *containers,* não ficamos presos ao ambiente que criou, poderá rodar em outro *container* com o mesmo *Docker* instalado independente do sistema operacional, pois as dependências estão todas dentro do *container* (Vitalino; Castro, 2018).

Com a utilização de *containers,* podemos citar três principais benefícios (AWS Brasil, 2020):

a) Execução em qualquer lugar;
b) Melhoria na utilização de recursos;
c) Alta escalabilidade.

1.3 *Docker*

É um PaaS que não é atrelado a nenhuma linguagem de programação, foi criado em 2008 e é um projeto *open source*. A AWS oferece suporte a *Docker* na nuvem. Suporta atualmente máquinas Linux 64 bits e outras plataformas, como Windows e MacOS (Vitalino; Castro, 2018).

1.4 Microsserviços

As aplicações quando criadas em um único *container* como um único serviço chamamos de aplicações monolíticas. Essas aplicações se tornam

problemáticas, pois são complicadas para manter e atualizar (Kim; Humble; Debois, 2018).

Quando cada componente do aplicativo roda independente e realiza comunicação com outros serviços por meio de uma API, utilizamos a arquitetura de microsserviços. Os microsserviços são independentes e podem ser escritos com estruturas e linguagens de programações diferentes.

1.5 *Containers* na AWS

Para ajudar a implantação do *container,* a AWS oferece vários serviços. As ferramentas para gerenciamento no AWS podem ser divididas em três categorias: registro, orquestração e computação (AWS Brasil, 2020).

1) *Amazon Elastic Container Registry* (ECR) é um registro de *container* que realiza o gerenciamento, armazenamento e implementação da imagem do *container*, sendo integrado ao Amazon ECS. Simplifica o fluxo de trabalho de produção, sem a necessidade de se preocupar com operação dos repositórios próprios ou dimensionar infraestrutura (AWS Brasil, 2020).
2) *Amazon Elastic Container Service* (ECS) é um serviço de orquestração que realiza a execução de aplicativos em *container* ou construção de microsserviços.
3) *Amazon Elastic Kubernetes Services* (EKS) é um serviço de orquestração de kubernetes que os utiliza na realização da execução de aplicativos.
4) AWS *Fargate* realiza a computação sem gerenciamento de servidor para *containers* facilitando o foco de seus aplicativos
5) *Amazon* EC2 é serviço de computação que realiza a execução de *containers* em infraestrutura de máquina virtual, possibilitando o controle total em nível de servidor sobre configuração e escalabilidade.

TEMA 2 – AWS *IDENTITY AND ACCESS MANAGEMENT* (IAM)

O IAM possibilita o gerenciamento seguro de quem fez o *login* e tem permissões para os serviços e recursos da AWS. É um recurso disponibilizado gratuitamente, somente os demais serviços do AWS são cobrados. O IAM possui uma infraestrutura completa para controlar a autenticação e autorização da conta da AWS.

Para entender o funcionamento do IAM, precisamos conhecer o significado de alguns termos:

1) Recursos: usuários, grupo, função, provedor de identidade e objetos que serão armazenados no AWS.
2) Identidades: usuários, grupos e funções. São alguns objetos de recursos que podem identificar e agrupar, podendo anexar a Identidade do IAM uma política.
3) Entidades: usuários IAM, usuários federados e funções assumidas do IAM. São alguns objetos de recursos que realizam a autenticação.
4) Principais: pessoa ou aplicativo para fazer *login* e solicitações para AWS.

Com o AWS IAM, é possível realizar gerenciamento dos usuários e seus acessos, gerenciamento de funções e suas permissões e gerenciamento de usuários federados e suas permissões.

2.1 Infraestrutura do IAM

Os principais elementos da infraestrutura do IAM são:

1) Principal: é uma pessoa ou aplicativo que estão aptos a realizar uma solicitação de ação ou operação para um recurso da AWS. Ao criar a conta na AWS é criada uma entidade Principal chamada usuário raiz, neste caso, é criada uma única identidade de *login* do tipo usuário raiz. Como melhor prática recomendada, deve ser utilizado o usuário raiz apenas para tarefa de gestão de contas e serviços, em que para melhor proteção da conta deve ser criadas entidades do IAM para realizar as tarefas diárias, ou seja, criar usuários e funções.
2) Solicitação: quando é utilizado o *console* de gerenciamento, API ou CLI do AWS por uma entidade principal, é preciso enviar uma solicitação para a AWS. São enviadas informações da solicitação em um contexto de solicitação: ações ou operações que deseja executar, recurso na qual serão realizadas ações ou operações, principal que enviou a solicitação, os dados do ambiente (IP, agente de usuário, status SSL ou hora do dia) e dados do recurso que está sendo solicitado (nome da tabela do DynamoDB ou *tag* da instância do Amazon EC2). O contexto de solicitação é utilizado para avaliar a solicitação realizada.

3) Autenticação: para enviar uma solicitação deve ser realizada uma autenticação na conta AWS. A autenticação é a ação de realizar o *login* e conectar na AWS. Para autenticação do usuário raiz, utiliza-se o e-mail e senha; o usuário IAM utiliza usuário e senha; para autenticação em um API ou AWS CLI, devem ser fornecidas chave de acesso e chave secreta.
4) Autorização: para concluir uma solicitação deve ser autorizado. São utilizados os valores do contexto da solicitação para verificação das políticas e, caso existam, se podem ser aplicadas aquela solicitação. Para realizar a permissão de acesso, é utilizada a política baseada em identidade, mas existem vários tipos de políticas que podem afetar a autorização da solicitação.
5) Ações ou operações: após autenticação e autorização, será realizada a aprovação das ações ou operações pelo AWS. Os serviços são operações que incluem o que você pode fazer nos recursos: visualizar, criar, editar e excluir. O IAM atualmente oferece em torno de 40 operações para o recurso de usuário; podemos citar quatro principais: *CreateUser, DeleteUser, GeUser e UpdateUser*.
6) Recursos: a execução nos recursos é realizada apenas após aprovação das operações em sua solicitação. Dentro de um serviço, objetos que são chamados de recursos são, por exemplo, instâncias do *Amazon EC2*, usuário do IAM e *Bucket do Amazon* S3.

2.2 Gerenciamento de identidade

É possível criar usuários com permissões personalizadas para ter maior segurança e organização. Também é possível simplificar o acesso com a utilização de federação de identidade existente.

Alguns usuários podem ser aplicativos e também podem ser criados vários usuários em uma conta do AWS, em que cada usuário tem uma senha individual para acesso ao console de gerenciamento do AWS.

Quando uma organização já tem uma forma de autenticação, não é necessário criar usuários no IAM, basta criar a federação dessas identidades de usuários no AWS. Também é possível utilizar o *OpenId Connect* (OIDC) para aplicativos móveis ou aplicativo baseado na web, por exemplo, Amazon, Facebook, Google.

2.3 Gerenciamento de acesso

Definição do que a entidade principal poderá fazer na conta, chamado de autorização. Para realizar o gerenciamento de acesso criamos políticas para anexar as identidades ou recursos do AWS. Um objeto que define suas permissões é uma política; quando um principal usa uma identidade para fazer solicitação são avaliadas as políticas. É nas políticas que são definidas as permissões nas quais é consentida ou negada a solicitação.

2.3 Políticas e contas

Para gerenciar uma única conta, podem ser definidas permissões, porém, quando se realiza gerenciamento de várias contas, torna-se necessário para facilitar o gerenciamento a criação de funções do IAM. As funções são feitas com base em lista de controles de acesso (ACLs).

2.4 Políticas e usuários

Quando é criado um usuário, ele não terá acesso a nada, será necessário conceder permissão criando politicas baseadas em identidade, que serão anexadas ao usuário ou ao grupo. Por padrão, as permissões que não são concedidas explicitamente são negadas. O IAM possui três tabelas de políticas no console de gerenciamento da AWS: resumo de políticas, resumo de serviços e resumo de ações.

2.5 Políticas e grupos

Os usuários podem ser organizados em grupos do IAM para anexar as políticas mantendo as credenciais individuais, mas todos terão as permissões anexadas no grupo. Podem existir várias políticas anexadas aos usuários ou aos grupos, concedendo permissões diferentes. Para conceder permissão, neste caso, será realizado um cálculo com base nas combinações de políticas.

2.6 Usuários, federados e funções

Diferente dos usuários do IAM, o usuário federado não possui identidade permanente em uma conta na AWS. Para atribuir permissão, neste caso, é necessário criar função e definir permissões para essa função. No momento do

login, será verificado quem está associado a uma função e as permissões são concedidas de acordo com as definições da função.

TEMA 3 – IDENTIDADES DO IAM

Para fornecer autenticação a pessoas e processos, as contas da AWS, criamos identidades do IAM. Para gerenciamento de usuários com uma unidade, criamos grupos. As políticas são associadas às identidades determinarão se um usuário, uma função ou um grupo podem executar uma ação e os recursos e condições necessários para tal.

3.1 Usuários

São criados para a representação de uma pessoa ou aplicativo que interagirá com a AWS, possui nome e credenciais. O usuário raiz tem acesso total aos serviços e recursos da conta da AWS. Para cada pessoa que necessitar de acesso de administrador, é recomendado que seja criado um usuário IAM incluindo-o no grupo “Administradores” e anexar política gerenciada *AdministratorAccess*.

Após a criação do usuário, são criadas formas de identificar esse usuário no IAM:

1) Nome amigável: é visto no Console da AWS; nome que foi especificado ao criar o usuário. Exemplos: Bob, TestApp etc.
2) Nome do recurso da Amazon (ARN): usado para identificar exclusivamente em toda a AWS. É utilizado para política de permissões e possui o seguinte formato: *arn:partition:service:region:account:resource*. (*partition* é onde o recurso se encontra, por padrão é *aws*; *service* é o produto, para o IAM o produto é *iam*; *region* é a região do recurso, para *iam* é deixada em branco; *account* é o ID da conta sem hífens; *resource* é a identificação por nome do recurso). Exemplo: arn:aws:iam::123456789012:root
3) Identificador exclusivo (ID): retornado apenas quando é utilizado a API, não é possível visualizar o ID no *console*. São utilizados prefixos para indicar o tipo de entidade o ID exclusivo se aplica: AAGA – Grupo de ações, ACCA – Credencial específica de contexto, AGPA – Grupo, AIDA – Usuário IAM, AIPA – Perfil da instância Amazon EC2, AKIA – Chave de

acesso, ANPA – Política gerenciada, ANVA-Versão em uma política gerenciada, APKA - Chave pública, AROA – Função, ASCA – Certificado e ASIA – Chaves temporárias (AWS STS). Exemplo: AIDAJQABLZS4A3QDU576Q.

3.2 Grupos

É um conjunto de usuários em que é possível realizar a especificação de permissões para vários usuários. O grupo não é uma identidade e possui algumas características importantes:

1) Pode conter vários usuários; um usuário pode estar em vários grupos.
2) Não podem conter outros grupos, apenas usuários.
3) Não há um grupo padrão já criado e que usuários quando criados são automaticamente incluídos. É necessário criar o grupo e incluir o novo usuário.
4) Existe um limite de grupos e um limite de quantos grupos o usuário pode participar.

O AWS possui várias ferramentas para o gerenciamento dos grupos no IAM em que podem ser: listagem de grupo, inclusão e renomeação de usuário nos grupos, anexar política no grupo, renomeação de um grupo e exclusão de um grupo.

3.3 Funções

É uma identidade criada com permissões específicas, sua função é similar à de usuários. No entanto, não está associada a uma pessoa e pode ser assumida por qualquer um que precisar. Função não possui credenciais de longo prazo, apenas são geradas credenciais temporárias para a pessoa que assumir a função.

É possível delegar função a usuários, aplicativos ou serviços sem incorporação da chave e também conceder acesso às pessoas externas que realizarão auditoria nos recursos.

As funções podem ser utilizadas pelas seguintes entidades:

1) Usuários da mesma conta da AWS;
2) Usuários em contas da AWS diferentes;

3) Usuários web, como usuários Amazon EC2;
4) Usuário externo, com autenticação por provedor de serviço *Open ID* ou SAML 2.0 ou *Identidy broker.*

Podemos ter três tipos de funções:

1) Função de serviço: um serviço assume uma função para realizar ações na conta da AWS. É necessário realizar uma configuração no ambiente e definir a função que assumirá o serviço, em que inclui todas as permissões necessárias para acessar os recursos da AWS. Os acessos só podem ser concedidos dentro da mesma conta.
2) Função de serviço para uma instância do EC2: a função é atribuída à instância quando for executada, quando um aplicativo estiver em execução pode recuperar as credenciais de segurança temporárias para realizar ações que forem permitidas pela função.
3) Função vinculada ao serviço: um serviço da AWS é vinculado diretamente à função. As permissões necessárias para chamada de outros serviços da AWS são incluídas na função que são predefinidas para aquele serviço.

TEMA 4 – GERENCIAMENTO DE ACESSO

Criando políticas e anexando a identidade e recursos é possível realizar a gestão de acessos no AWS. Quando anexamos uma política a uma identidade ou recursos, as permissões são definidas. São as permissões que possibilitam a avaliação da solicitação e a AWS avalia decidindo se permite ou nega.

4.1 Tipos de políticas

Existem seis tipos de políticas disponíveis para utilização no AWS:

1) Baseada em identidade: concede permissões a uma identidade (usuário, grupos e funções).
2) Baseada em recursos: concede permissões a entidade principal.
3) Lista de permissões: define número máximo de permissões, podendo conceder a uma entidade (usuário ou função), porém, não concede permissão.

4) SCPs de organizações: define o número máximo de permissões, para políticas baseadas em identidade ou recursos, que podem ser concedidas dentro da conta, porém, não concede permissão.
5) Lista de controle de acessos (ACLs): são políticas para conceder permissão entre contas.
6) Políticas de sessões: limitam as permissões concedidas às políticas baseadas em identidade ou função, concedendo a uma sessão criada, porém não concedem permissão.

4.2 Políticas baseadas em identidades

São documentos JSON[1] anexados a uma identidade. As políticas baseadas em identidade são as regras que controlam as ações que uma entidade (usuário ou função) pode executar, quais recursos e quais condições. Existem duas categorias de política de identidade:

1) Política gerenciadas: são independentes e podem ser anexadas em vários usuários, grupos e funções. Podem ser de dois tipos: criadas e gerenciadas pela AWS ou criadas e gerenciadas pelo cliente.
2) Política em linha: são incorporadas direto em um usuário, grupo ou função.

4.3 Políticas baseada em recursos

São documentos JSON anexados a um recurso, são também políticas embutidas e não possuem políticas gerenciadas. A política de confiança de uma função é a única suportada pelo serviço IAM, que definem quais entidades principais podem assumir a função. O principal trabalhará sem ser preciso renunciar às permissões e recebendo mais as permissões da função, trabalhando em uma conta confiável tento acesso a recursos de outra conta.

Os principais que podem realizar a especificação de uma política de confiança são: contas, usuários do IAM, usuários federados, funções, sessões da função ou serviços da AWS.

---

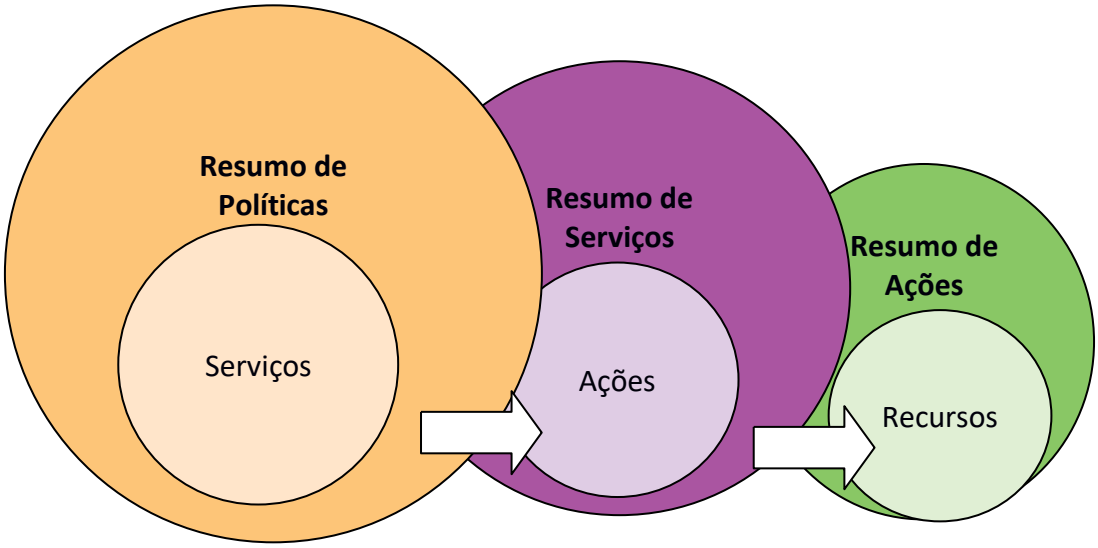
1 *JavaScript Object Notation* (JSON) é um modelo para realizar o armazenamento das políticas e a transmissão no formato texto, a estrutura do documento é bem mais compacta que o modelo XML, tornando mais rápida a transmissão de grandes volumes de dados.

Podemos citar três serviços da AWS que oferecem suporte a políticas baseadas em recurso: *buckets* da Amazon S3, tópicos da Amazon SNS e filas Amazon SQS.

4.4 Noções básicas sobre permissões

As tabelas do resumo de políticas que são incluídas no console do IAM são a descrição do nível de acesso, os recursos e permissões ou negações para cada serviço.

Figura 3 – Tabelas do resumo de políticas



Fonte: AWS Brasil, 2020.

Na Figura 3, podemos ver a representação das três tabelas de políticas. Para entender melhor o funcionamento, vamos resumi-lo: uma lista de serviços é encontrada na tabela de resumo de políticas; quando selecionado um serviço, poderemos ver uma lista de ações da tabela de resumo de serviços. A lista de serviço pode ser visualizada quando escolhemos uma ação da tabela de resumo de serviços; será apresentada, então, uma lista de recursos e condições para ação na tabela de resumo de ações. Os resumos das políticas podem ser utilizados para a compreensão do que é permitido e negado por uma política.

TEMA 5 – *AMAZON ELASTIC COMPUTE CLOUD*: EC2

A *Amazon* EC2 é um serviço web que possibilita as organizações em um ambiente computacional virtual, com interface de serviço para criação de instâncias com possibilidade de escolha de vários sistemas operacionais.

5.1 Conceitos básicos

Existem várias formas de utilizar o *Amazon* EC2: *console* para gerenciamento da AWS, linha de comando (ILC) ou os SDKs, fornecendo, assim, um controle total dos recursos computacionais e permitindo trabalhar em um ambiente da Amazon.

Com os recursos da *Amazon* EC2 você pode:

1) Criar instâncias sem servidores;
2) Otimizar performance e custos da computação com diversos tipos de instâncias;
3) Hibernar e retomar uma instância no estado posterior;
4) Instâncias de Unidade de Processamento Gráfico (GPU) Computacional de uso geral;
5) Instâncias de Unidade de Processamento Gráfico (GPU) de alta capacidade gráfica;
6) Instâncias com taxas de E/S acima de 3 milhões de *Input/Output per Second* (IOPS);
7) Instâncias de armazenamento denso;
8) Configurações de número de CPUs e desabilitar *multithread*;
9) Opção de armazenamentos flexíveis (*Amazon* EBS e *Amazon* EFS);
10) Pagamento de acordo com utilizado;
11) Regiões e zonas de disponibilidade para colocar as instâncias;
12) Endereços IP elásticos para computação em nuvem dinâmica;
13) *Amazon* EC2 *AutoScaling* para ampliação e redução automática da capacidade do *Amazon* EC2;
14) *Cluster* de computação de alta performance (HPC) com capacidade de rede de alta performance;
15) Redes aperfeiçoadas obtendo performance de pacotes por segundo (PPS);

16) Adaptador de malha elástica (*Elastic Frabic Adapter* – EFA) é a interface de rede que possibilita execução de aplicativos HPC;
17) AWS *PrivateLink* foi projetado para acessar os serviços de forma altamente disponível e com alta performance, mantendo o tráfego de rede dentro da rede da AWS;
18) *Amazon Time SyncService* fornece fonte de horário altamente precisa.

5.2 Instâncias de uso geral

Consistem em várias combinações de CPU, memória, armazenamento e capacidade de rede. Os tipos de instâncias incluem um ou mais tamanhos, possibilitando escalabilidade de recursos. São nove tipos de instância para uso geral:

1) A1: com processador AWS *Graviton Custom*, oferecem economia e são ideais para cargas de trabalhos com escala horizontal baseada em *Arm*, como servidores *web*, *conteiners* de microsserviços, frotas de armazenamento em cache, *datastores* distribuídos e ambientes de desenvolvimentos.
2) T3: com processadores Intel Xeon, possui capacidade de intermitência e fornece equilíbrio entre recursos de computação, memória e rede para aplicativos com uso moderado de CPU, como microsserviços, aplicativos interativos de baixa latência, banco de dados de pequeno e médio porte, área de trabalhos virtuais, ambientes de desenvolvimentos, repositórios de código e aplicativos essenciais ao negócio.
3) T3a: com processadores AMD EPYC Série 700, é semelhante àT3, porém, oferece economia de 10% em relação aos outros tipos de instâncias que podem ser comparadas.
4) T2: com processadores Intel Xeon, são instâncias de performance com capacidade de intermitência acima da referência como *site* e aplicativos *web*, ambientes de desenvolvimento, servidores de compilação repositórios de código, microsserviços, ambientes de testes e preparação e aplicativos da linha de negócios.
5) M6g: com processadores AWS Graviton2, oferecem preço e performance até 40% melhor em relação as instâncias M5 e também equilíbrio de recursos de computação, memória e rede. Utilizada para aplicativos em

software de código aberto, como servidores de aplicativos, microsserviços, servidores de jogos, repositórios de dados com tamanho médio e armazenamento cache.

6) M5: com processadores Intel Xeon Platinum 8175M, oferece equilíbrio entre recursos de computação, memória e rede. Utilizada por banco de dados de pequeno e médio porte, tarefas que exigem memória adicional no processamento de dados, armazenamento de cache, servidores *backend* para SAP, *SharePoint*, da Microsoft, computação em *cluster* e demais aplicativos de empresas.
7) M5a: com processadores AMD EPYC série 7000, é semelhante à M5, porém, oferece economia de 10% em relação a outras instâncias que podem ser comparadas.
8) M5n: com processadores Intel Xeon de 2ª Geração (*Cascade Lake*), é ideal para aplicativos que necessitam de altas taxas de transferência de rede e performance de taxa de pacotes. Utilizada em servidores de aplicativos *web*, banco de dados pequeno e médio, processamento de cluster, servidores de jogos, armazenamento em cache e outros aplicativos da empresa.
9) M4: com processador Intel Xeon E5-2686 v4 (*Broadwell*), é boa opção para muitos aplicativos. Utilizada por bancos de dados de pequeno e médio porte, tarefas de processamento de dados que exigem memória adicional, armazenamento em cache e para executar servidores de *backend* para SAP, Microsoft *SharePoint*, computação em *cluster* e outros aplicativos empresariais.

5.3 Instâncias otimizadas para computação

Para aplicativos que precisam de processadores de alto desempenho, são adequadas para cargas de trabalho em lote e aplicativos com uso intensivo de computação. São três os tipos de instâncias otimizadas para computação:

1) C5: com dois tipos de processador Intel Xeon de 2ª Geração ou Intel Xeon *Scalable* de 2ª Geração, oferece performance alta com um preço baixo por taxa de computação.
2) C5n: com processador Intel Xeon Platinum, oferece taxa de transferência de rede melhorada e desempenho do taxa de pacotes.

3) C4: com processador Intel Xeon E5-2666 v3 (Haswell), oferece alto desempenho e economia com baixo preço por desempenho de computação.

5.4 Instâncias otimizadas para memória

São projetadas para oferecer rápido desempenho para trabalhos com carga na memória de grande conjunto de dados, como para análise de *big data* em tempo real e outros aplicativos com uso intensivo de memória. São oito tipos de instâncias otimizadas para memória:

1) R5: com processadores Intel Xeon Platinum 8175, fornece 5% mais de memória por vCPU que R4 e no máximo 768 GB de instância.
2) R5a: com processadores AMD EPYC série 7000, oferece custo até 5% menor em comparação com outras que são comparáveis.
3) R5n: com processador Intel Xeon de 2ª Geração, oferece altas taxas de transferências de rede e performance de taxa de pacotes.
4) R4: com processador Intel Xeon E5-2686 v4 (*Broadwell*), oferece um preço melhor por GiB de RAM que instâncias R3.
5) X1e: com processador Intel Xeon E7-8800 v3 (*Haswell*), oferece menores preços por GiB de RAM entre os demais tipos de instâncias da Amazon EC2.
6) X1: com processador Intel Xeon E7-8800 v3 (*Haswell*), para aplicativos em memória, de escala grande e nível empresarial.
7) Mais memória: com dois tipos de processadores Intel Xeon Platinum 8176M (*Skylake*) de 6 TB até 12 TB ou 18 TB e 24 TB Intel Xeon Scalable, para execução de grandes bancos de dados em memória, incluindo implantação do SAP HANA.
8) z1d: com processador Intel Xeon Scalable, possui alta frequência, sendo a mais rápida de todas as instâncias de nuvem.

5.5 Instâncias para computação acelerada

Possuem aceleradores de *hardware* para funções como cálculo de ponto flutuante, processamento gráfico de forma mais eficiente nas CPUs. São seis os tipos de instâncias para computação acelerada, conforme vemos a seguir.

1) P3: com dois tipos de processadores Intel Xeon E5-2686 v4 (*Broadwell*) e Intel Xeon P-8175M, com até 8 GPUs NVIDIA Tesla V100 GPUs, para aplicativos de uso geral.
2) P2: com processadores Intel Xeon E5-2686 v4 (*Broadwell*) com GPUs NVIDIA K80 de alta performance, para aplicativos de uso geral.
3) Inf1: com processadores Intel Xeon Scalable de 2ª geração e até 16 chips AWS Inferentia, criados para inferência de *machine learning*.
4) G4: com processadores Intel Xeon Scalable (*Cascade Lake*) com GPUs NVIDIA T4 Tensor Core, ajuda a acelerar a inferências de *machine learning* e cargas com muitos gráficos.
5) G3: com processadores Intel Xeon E5-2686 v4 (*Broadwell*) com GPUs NVIDIA Tesla M60, para aplicativos com consumo gráficos altos.
6) F1: com processadores Intel Xeon E5-2686 v4 (*Broadwell*) com FPGAs (*Field Programmable Gate Arrays* – arranjos programáveis de portas em campo).

5.6 Instâncias otimizadas para armazenamento

São para alto acesso de leitura de gravação de conjunto de dados muito grandes para armazenamento local. Podem oferecer taxa de E/S de dezenas de milhares de IOPS. São quatro os tipos de instâncias otimizadas para armazenamento:

1) I3: com processadores Intel Xeon E5-2686 v4 (*Broadwell*) e baseado em SSD *Non-Volatile Memory Express* (NVMe) com alto IOPS com baixo custo. Também oferecem instâncias sem SO para cargas não virtualizadas.
2) I3zen: com processador Intel Xeon Scalable (*Skylake*) com até 60 TB de SSD NVMe, oferece o menor preço por GB de armazenamento no Amazon EC2.
3) D2: com processador Intel Xeon E5-2676 v3 (*Haswell*), oferece até 48 TB de armazenamento local baseado em HDD, oferece menor preço de *Throughtput* de disco da Amazon EC2.
4) H1: com processador Intel Xeon E5 2686 v4, oferece até 16TB de armazenamento local baseado em HDD.

5.7 Imagens de Máquina da Amazon (AMIs)

As imagens fornecem as informações que são necessárias para iniciar uma instância. É necessário realizar a especificação de uma AMI ao iniciar uma instância. É possível a execução de várias instâncias em uma única AMI quando são necessárias várias instâncias com as mesmas configurações. São incluídos dentro de uma AMI:

1) Um ou mais *Snapshots* do EBS ou modelo para volume raiz da instância;
2) Permissões de execução, em que são indicadas quais contas podem utilizar;
3) Volumes que serão anexados à instância quando for executada.

Com base nas características, pode ser selecionada uma AMI para uso:

1) Região.
2) Sistema Operacional.
3) Arquitetura (32 bits ou 64 bits).
4) Permissões de execução podem ser públicas, quando é concedida permissão a todas as contas; explícita, quando é concedida permissão a contas indicadas; ou implícita, quando há permissão de forma implícita para uma AMI. A *Amazon* EC2 já oferece várias AMIs públicas e também existem AMI pagas criadas por desenvolvedores.
5) Armazenamento para o dispositivo raiz: pode ser um volume do *Amazon* EBS criado por meio de um *snapshot* ou um volume de armazenamento de instância criado no *Amazon* S3. Na Tabela 1, são resumidas as diferenças entre os dois tipos de armazenamento.

Tabela 1 – Diferenças dos tipos de armazenamento do AMI

Característica	**AMI com *Amazon* EBS**	**AMI com armazenamento de instâncias da *Amazon* S3**
Tempo de inicialização para uma instância	Geralmente menos de um minuto	Geralmente menos de cinco minutos
Limite de tamanho para um dispositivo raiz	16 TiB	10 GiB

Volume do dispositivo raiz	Volume do *Amazon* EBS	Volumes de armazenamento de instâncias
Persistência de dados	Por padrão, o volume raiz é excluído quando a instância é encerrada. Os dados em todos os outros volumes do *Amazon* EBS persistem após o encerramento da instância, por padrão.	Os dados em qualquer volume do armazenamento de instâncias persistem apenas durante sua vida útil.
Modificações	O tipo de instância, o kernel, o disco da RAM e os dados do usuário podem ser alterados enquanto a instância está parada.	Os atributos de instância são fixos durante a vida útil de uma instância.
Cobranças	Você é cobrado pelo uso de instância, uso de volume do *Amazon* EBS e pelo armazenamento da AMI como um *snapshot* do *Amazon* EBS.	Você é cobrado pelo uso da instância e pelo armazenamento da AMI no *Amazon* S3.
Criação/empacotamento da AMI	Usa um único comando/chamada	Requer instalação e uso de ferramentas de AMI
Estado parado	Pode ser colocada em estado parado quando a instância não está em execução, mas o volume raiz é mantido no *Amazon* EBS	Não pode estar no estado parado. As instâncias estão em execução ou encerradas

Fonte: AWS Brasil, 2020.

FINALIZANDO

No Tema 1, estudamos máquina virtual e *container* e aprendemos que:

- A Máquina Virtual é uma camada entre HW e SW implementada por SW.
- O *container* é o empacotamento da aplicação com as dependências mais leves e integradas com sistema operacional do servidor. Os três benefícios de utilizar container são: execução em qualquer lugar, melhoria na utilização de recursos e alta escalabilidade.
- *Docker* é um PaaS sem linguagem de programação atrelada e os microsserviços quando componentes rodam independentes e realizam comunicação com outros serviços por API.

- As ferramentas oferecidas para gerenciamento de *containers* na AWS são: *Amazon Elastic Container Registry* (ECR), *Amazon Elastic Container Service* (ECS), *Amazon Elastic Kubernetes Services* (EKS) e AWS *Fargate*, *Amazon* EC2.

No Tema 2, estudamos o AWS IAM, que realiza o gerenciamento seguro de acesso dos serviços e recursos da AWS. Termos importantes: **recursos**, **identidades**, **entidades** e **principais**. Conhecemos os principais elementos da infraestrutura do IAM:

1) **Principal** é a pessoa ou aplicativo;
2) **Solicitação** é a ação ou operação que deseja executar em determinado contexto;
3) **Autenticação** é a ação de realizar *login* e conexão na AWS;
4) **Ação** ou **operação** é o que pode ser feito nos recursos: criar, visualizar, editar e excluir;
5) **Recurso** é o serviço oferecido dentro da AWS: instâncias *Amazon* EC2, por exemplo.

Conhecemos em detalhes que o gerenciamento de identidade é a forma de identificação de pessoas ou aplicativos e o gerenciamento de acesso é a autorização que a entidade poderá realizara na conta na AWS.

Vimos as políticas e contas em que é realizado o gerenciamento da permissão individual ou por funções com ACLs; as políticas e usuários em que são criadas políticas anexas ao usuário ou ao grupo, armazenadas em três tabelas que contêm as políticas (resumo de políticas, resumo de serviços e resumo de ações); as políticas e grupos em que são criadas políticas comuns para vários usuários e a organização podem ser realizadas por grupos do IAM, mantendo as credenciais individuais; os usuários federados e funções que são a forma de atribuir permissão a usuários que não possuem conta na AWS.

No Tema 3, estudamos sobre a identidade do IAM e a maneira de realizar a gestão de identidade e formas de associação. Os usuários são pessoas ou aplicativos, existindo três formas de identificar usuários: nome amigável, ARN e ID. Os grupos são conjuntos de vários usuários e as ferramentas para sua gestão são: listagem de grupo, inclusão e renomeação de usuário nos grupos, anexar política no grupo, renomeação de um grupo e exclusão de um grupo; as funções

são similares a usuários, porém, não estão associadas a uma única pessoa e pode ser assumida por qualquer pessoa que necessitar. Existem três tipos de funções: de serviço, serviço para uma instância do EC2 e vinculada ao serviço.

No Tema 4, estudamos o gerenciamento de acesso em que realizamos a criação e associação das políticas a identidades e recursos. Os seis tipos de políticas são: baseadas em identidade, baseadas em recursos, lista de permissão, SCPs de organização, ACLs e políticas de sessões. As políticas baseadas em identidades são documentos JSON anexados à identidade, sendo categorizadas de duas formas: gerenciadas (pela AWS ou pelo cliente) ou em linha que são incorporadas. As políticas baseadas em recursos são documentos JSOn anexados a um recurso, e as políticas de confiança de uma função pode ser especificadas pelos seguintes principais: contas, usuários do IAM, usuários federados, funções, sessões da função ou serviços da AWS.

Conhecemos as noções básicas sobre permissões e estudamos as três tabelas de resumo das políticas, que descrevem os níveis de acesso, os recursos e permissões e negações para os serviços.

No Tema 5, conhecemos o *Amazon* EC2, que é o serviço web para criação de instâncias com possibilidade de utilização de vários SO. Estudamos os conceitos básicos e aprendemos que, com os recursos da *Amazon* EC2, é possível realizar criação de instâncias sem servidores, otimização de *performance* e custos, hibernar instâncias, instâncias com GPU, armazenamento denso, armazenamento flexível com *Amazon* EBS e EFS, pagamento pelo que foi utilizado, *AutoScaling*, HPC, redes aperfeiçoadas, EFA e *Amazon Time SyncService*.

Existem vários tipos de instâncias da *Amazon* EC2, sendo elas:

1) Instâncias de uso geral: A1, T3, T3a, T2, M6g, M5, M5a, M5n e M4;
2) Instâncias otimizadas para computação: C5, C5n e C4;
3) Instâncias otimizadas para memória: R5, R5a, R5n, R4, X1e, X1, mais memória e Z1d;
4) Instâncias para computação acelerada: P3, P2, Inf1, G4, G3 e F1;
5) Instâncias otimizadas para armazenamento: I3, I3Zen, D2 e H1.

Aprendemos sobre as Imagens de Máquina da Amazon (AMIs), que fornecem o que é necessário para iniciar uma instância. E, por fim, vimos as

principais diferenças das AMI criadas pelo *Amazon* EBS ou instâncias da *Amazon* S3.

REFERÊNCIAS

AWS BRASIL. Disponível em: <https://aws.amazon.com/pt>. Acesso em: 28 out. 2020.

KIM, G.; HUMBLE, J.; DEBOIS, P. **Manual de DevOps**. Rio de Janeiro: Alta Book, 2018.

VERAS, M. **Virtualização**. 2. ed. Rio de Janeiro: Brasport, 2016.

VITALINO, J. F. N.; CASTRO, M. A. N. **Descomplicando o Docker**. 2. ed. Rio de Janeiro: Brasport, 2018