>>> >>> Kubernetes

Name: Henrique Tsuyoshi Yara (OPUS-software)
Date: January 17, 2023

Figure: AWS logo

>>> Índice

>>> Recursos

* Cada recurso vai ter um **Amazon Resource name** (Identificador único)

>>> Free Tier

* São recursos que podem ser usadas de graça na Amazon

>>> Calculadora

* É utilizada para calcular o custo total de algum recurso
  * Calculadora antiga
  * Calculadora nova

>>> Regiões

* Cada região tem um preço diferente
* Uma região é composta de zonas de disponibilidade
* Algumas regiões podem ter mais serviços que outras
* **OBS:** É bom saber se juridicamente a gente pode armazenar os dados fora do Brasil
  * Regiões e zonas de disponibilidade
  * Serviços regionais
* **OBS:** Tráfegos entre zonas de disponibilidade ou regiões podem acabar sendo cobrados

>>> Zonas de disponibilidade

* Compõem as regiões
  * Serviços regionais
* **OBS:** Tráfegos entre zonas de disponibilidade ou regiões podem acabar sendo cobrados

>>> Status AWS

* Para verificar o status das zonas de disponibilidade/regiões ou recursos
  * Status AWS
* **OBS:** Tráfegos entre zonas de disponibilidade ou regiões podem acabar sendo cobrados

>>> Pontos de presença

* Pontos de cache utilizado pela AWS (É possível usar *CNDs*)

>>> IAM

* Identity and Acess Management
* Boas práticas:
  * Habilitar MFA
  * Criar um usuário padrão para cada pessoa do time e dar permissões (Não usar o **root**)
  * Usar grupos para atribuir permissões
  * Aplicar uma política de senhas do IAM

>>> IAM - Users

* **Programmatic access**
* Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.
  * Instalar o CLI para ter acesso ao AWS
  * Dar acesso de um bucket para uma aplicação
* *AWS Management Console access*
* Enables a password that allows users to sign-in to the AWS Management Console.
  * Precisa dar permissão para esse usuário

>>> IAM - Tags

* Servem para a gente identificar serviços
* É possível fazer um relatório de faturamento baseado em *Tags*
* **OBS:** É possível ter até 50 tags por serviço

>>> IAM - Políticas Pt.1

* É uma boa prática criar grupos com permissões para os usuários. E não colocar permissões diretamente no usuário
* Permissões mais específicas são mais fortes (Permissão de usuário prevalece contra permissão de grupo)
* Políticas de senha
    * Exigir que o usuário use senhas fortes
    * Expiração de senhas
    * Impedir reutilização de senhas
    * Etc...

>>> IAM - Políticas Pt.2

* Políticas de acesso
  * As políticas podem ser definidas por um arquivo Json
  * Pode ser usado políticas prontas ou criar políticas específicas
  * Então as políticas podem ser atribuídas em usuários/grupos

>>> Funções/Roles

* Dar permissões para:
    * Recursos
        * Ex: Dar permissão para uma instância acessar um bucket
    * Outras contas AWS
    * Federações do SAML 2.0
    * Identidade web (Login Google, amazon, etc...)

>>> Relatórios de acesso

* Relatórios de credenciais
  * Lista de todas as credenciais geradas
* Access Analyzer: Gera um relatório de políticas pra a gente ver o que precisa ser modificado. é possível arquivar, resolver, etc...

>>> Virtual Private Cloud (VPC)

* VPCs são isoladas entre si, mas podem ser configuradas para se comunicarem
* Cada região tem uma VPC padrão, mas é recomendada criar sua própria VPC para o ambiente de produção
* Dentro de uma VPC é possível criar uma subnet
* As subnets são aplicadas em AZs (Zonas de disponibilidade)
* Subnet:
  * Pública: Pode ser acessada remotamente por qualquer lugar
  * Privada: Só vai ser acessível por dentro da AWS
* **VPC wizard** tem algumas configurações pré-definidas de VPC
* Lembrar de verificar e configurar:
  * DHCP options set
  * DNS resolution
  * DNS hostname

>>> Internet Gateway

* Libera a entrada e a saída de determinado **Route Table**
* Não tem custo

>>> Route table

* Associa as **subnets**
* Se a **Route table** não tiver uma rota default ela não está pública

>>> Security Groups X NetworkACL

* **Securty Groups**
  * Opera no nível de instância (Primeira camada de defesa)
  * Apenas regras de liberação
  * Stateful: o tráfego de retorno é automaticamente permitido, independentemente de quaisquer regras
  * Aplica-se a uma instância somente quando especificado o grupo de segurança
* **NetworkACL**
  * Regra de segurança da rede (Como se fosse um *firewall*)
  * Regras de liberação e negação
  * Stateless: o tráfego de retorno deve ser explicitamente permitido pelas regras
  * Aplica a todas as instâncias nas sub-redes

>>> NetworkACL

* Cada regra vai ter uma prioridade
* É bom deixar um espaço entre cada regra para possíveis regras futuras (Ex: deixar 10 espaços entre cada regra)
* **OBS**: É bom liberar portas efêmeras (1024-65535). São usadas para comunicações de saída através do protocolo de rede TCP/IP

>>> Virtualização

* Criação de infraestruturas virtuais a partir de uma estrutura física

>>> Cloud

* Conceito que reúne vários *softwares* e utiliza de virtualização
* Possui algumas características específicas:
  * Autoserviço sob demanda
  * Amplo acesso a rede
  * Pool de recursos
  * Rápida elaticidade
  * Serviços mensuráveis
* **Colocation:** 1000 **VMs** na rede, **VPS** por um provedor, servidor físico em um provedor, etc...

>>> NIST

* Um modelo para habilitar o acesso por rede a um conjunto compartilhado de recursos de computação e precisa ser:
  * Ubíquo (Pode ser encontrado em todos os lugares)
  * Conveniente
  * Sob demanda
* **Recursos de computação:** Redes, servidores, armazenamento, aplicações e serviços
* Esses recursos devem ser provisionados e liberados com o mínimo de esforço de gerenciamento ou interação com o provedor de serviços.

>>> Características

* Auto-serviço sob demanda
* Amplo acesso por rede
* Agrupamento de recursos
* Elasticidade rápida
* Serviço mensurado

>>> Auto-serviço sob demanda

* O consumidor pode providionar por conta própra **Recursos de computação**
* Não necessita da intervenção humana dos provedores de serviço

>>> Amplo acesso por rede

* Os **Recursos de computação** estão disponíveis através da rede
* São acessados através de mecanismos padronizados que promovem o uso por dispositivos, clientes leves ou ricos de diversas plataformas (Smartphones, tablets, laptops ou desktops)

>>> Agrupamento de recursos

* Os **Recusos de computação** do provedor são agrupados para atender a múltiplos consumidores em modalidade multi-inquilinos (Recursos físicos e virtuais diferentes dinamicamentes atribuídos e reatribuídos conforme a demanda dos consumidores)
* Há uma certa independência de localização geográfica, uma vez que o consumidor em geral não controla ou conhece a localização exata dos recursos fornecidos
* Mas pode ser capaz de especificar a localização em um nível de abstração mais alto (país, estado, datacenter)

>>> Elasticidade rápida

* Os recursos podem ser provisionados e liberados elasticamente, em alguns casos automaticamentes, para rapidamente aumentar ou diminuir de acordo com a demanda
* Para o consumidor, os recursos disponíveis para provisionamento muitas vezes parecem ser ilimitados e podem ser alocados em qualquer quantidade e a qualquer tempo

>>> Serviços mensurado

* Os sistemas na nuvem automaticamente controlam e otimizam o uso dos recursos através de medições em um nível de abstração apropriado para o tipo de serviço (como armazenamento, processamento, comunicação de ree e contas de usuário ativas)
* A utilização de recursos pode ser monitorada, controlada e informada, gerando transparência tanto para o fornecedor como para o consumidor do serviço utilizado

>>> Papéis e atividades do profissional cloud

* Consumidor de nuvem
* Provedor de nuvem
* Broker de nuvem
* Auditor
* Operadora de nuvem

>>> Consumidor de nuvem

* Uma pessoa/organização que mantém relação comercial com o fornecedor da nuvem, e usa o serviço
* Uso:
  * Um consumidor de nuvem procura o catálogo de serviços de um provedor de nuvem
  * Solicita o serviço apropriado
  * Configura contratos de serviço com o provedor da nuvem
  * Usa o serviço
* O consumidor pode ser cobrado pelo serviço fornecido e precisa organizar os pagamentos de acordo
* **OBS:** Dependendo dos serviços solicitados, as atividades e os cenários de uso podem ser diferentes entre os consumidores da nuvem

>>> Provedor de nuvem

* Um provedor de nuvem pode ser uma pessoa, uma organização ou uma entidade responsável por disponibilizar um serviço aos consumidores de nuvem
* Um provedor de nuvem:
  * Cria o software/plataforma/serviços de infraestrutura solicitados
  * Gerencia a infraestrutura técnica necessária para fornecr os serviços
  * Providencia os acordos de níveis de serviço (SLA) e protege a segurança e a privacidade dos serviços

>>> Broker de nuvem

* Uma entidade que
  * Gerencia o uso
  * Desempenho
  * Entrega de serviços na nuvem
  * Negocia relaões entre o **Provedor de nuvem** e o **Consumidor de nuvem**

>>> Auditor

* Pode avaliar os serviços fornecidos por um **Provedor de nuvem:**
  * Controles de segurança
  * Impacto de privacidade
  * Desempenho
  * Aderência aos parâmetros do acordo de nível de serviço (SLA)

>>> Operadora de nuvem

* Um intermediário que fornece conectividade e transporte de serviços na nuvem entre **Consumidores de nuvem** e **Provedores de nuvem**
* As operadoras de nuvem fornecem acesso aos consumidores através de redes, telecomunicações e outros dispositivos de acesso (computadores, laptops, telefones celulares, etc...)
* A distribuição de serviços na nuvem é normalmente fornecida por operadoras de rede e telecomunicações ou por um agente de transporte

>>> Tipos de nuvem

* Infraestrutura como Serviço (Iaas)
* Software como Serviço (SaaS)
* Plataforma como Serviço (PaaS)

>>> IaaS - Infrastructure as a Service

* O recurso fornecido ao consumidor é provisionar:
  * Processamento
  * Armazenamento
  * Comunicação de rede
  * Outros recursos de computação funcamentais nos quais o consumidor pode instalar e executar softwares em geral, incluindo sistemas operaionais e aplicativos
  * Possivelmente um controle limitado de alguns componentes de rede (*firewall*)

>>> PaaS - Plataform as a Service

* O recurso fornecido ao consumidor é instalar na infraestrutura na nuvem aplicativos criados ou adiquiridos pelo consumidor,
* O consumidor tem controle sobre as aplicações instaladas e possívelmente configuraçoes de hospedagem de aplicações
* O consumidor não gerencia nem controla a infraestrutura na nuvem subjacente (Rede, servidores, sistemas operacionais, armazenamento ou mesmo recursos individuais da aplicação, com a possível exeção de configurações limitadas por usuário)

>>> SaaS - Software as a Service

* O recurso fornecido ao consumidor é o uso de aplicções do fornecedor executando em uma infraestrutura na nuvem
* As aplicações podem ser acessadas por vários dispositivos clientes através de interfaces leves ou ricas
* O consumidor não gerencia nem controla a infraestrutura na nuvem subjacente (Rede, servidores, sistemas operacionais, armazenamento ou mesmo recursos individuais da aplicação, com a possível exeção de configurações limitadas por usuário)

>>> Exemplos

Tipo	Serviço	Exemplos
IaaS	Rede virtualizada Armazenamento de dados Servidores Virtuais	AWS VPC, Azura Virtual Network AWS S3, Google cloud storage AWS EC2, Azure Virtual Machines
PaaS	Infraestrutura para desenvolvimento, implantação e execução de aplicativos Plataforma testes e gerenciamento de aplicações	Heroku, Google App Engine AWS Elastic Beanstalk
SaaS	Armazenamento Dados Editor de textos e planilha SIstema de Gestão de banco de dados	DropBox, Google Drive Gsuite e Office 365 AWS RDS, Google Cloud SQL

Table: Exemplos de serviços

>>> Categorias de Serviços de nuvem

* Comunicação como serviço (CaaS)
* Computação como serviço (CompaaS)
* Armazenamento de dados como serviço (DSaaS)
* Rede como serviço (NaaS)
* Banco de dados como serviço (DBaaS)

>>> Comunicação como serviço (CaaS)

* As capacidades oferecidas ao cliente do serviço de nuvem são a interação e a colaboração em tempo real

>>> Computação como serviço (CompaaS)

* As capacidades oferecidas ao cliente do serviço de nuvem são a provisão e o uso de recursos de processamento necessários à implantação e execucão de softwares

>>> Armazenamento de dados como serviço (DSaaS)

* As capacidades oferecidas ao cliente do serviço de nuvem são a provisão e o uso de armazenamento de dados e suas capacidades relacionadas

>>> Rede como serviço (NaaS)

* As capacidades oferecidas ao cliente do serviço de nuvem são a conectividade para o transporte e as capacidades relacionadas à rede

>>> Banco de dados como serviço (DBaaS)

* Oferece a funcionalidade d eum banco de dados semelhante ao que é encontrado em SGBDs

>>> Modelos de implementação de nuvem

* Nuvem pública
* Nuvem privada
* Nuvem híbrida
* Nuvem comunitária

>>> Nuvem pública

* Provisionada para uso aberto ao público em geral
* Sua propriedade, gerenciamento e operação podem ser de:
  * Uma empresa
  * Uma instituição acadêmica
  * Uma organização do governo
  * Ou uma combinação mista
* Fica nas instalações do fornecedor
* **OBS**: Criar uma estrutura na Amazon e configurar usando VPN ou conexão direta continua sendo uma nuvem pública

>>> Nuvem privada

* Provisionada para uso exclusivo por uma únic organização composta de diversos consumidores
* A sua propriedade, gerenciamento de operação podem ser de:
  * A própria organização
  * Terceiros
  * Combinação mista
* Pode estar dentro ou fora das instalações da organização
* Nuvem privada não é organização e não precisa estar instalada localmente

>>> Nuvem comunitária

* Provisionada para uso exclusivo por uma determinada comunidade de consumidores de organizações que têm interesses em comum (missão, requisitos de segurança, políticas, observância de regulamentações)
* A sua propriedade, gerenciamento e operação podem ser de:
  * Uma organização
  * Mais de uma organizações da comunidade
  * Terceiros
  * Combinação mista
* Pode estar dentro ou fora das instalações das organizações participantes

>>> Nuvem híbrida

* Composição de duas ou mais infraestruturas na nuvem (**privadas**, **comunitárias** ou **públicas**) que permanecem entidades distintas
* São interligadas por tecnoogia padronizada ou proprietária que permite a comunicação de dados e portabilidade de aplicações (Transferencia de processamento para a nuvem para balanceamento de carga entre nuvens)

>>> Single-tenant X Multi-tetant

* Single-tenant

* Várias empresas compartilham a mesma instância para armazenamento
* Instância é dividida/particionada para que as empresas não acessem informações de outra

* Benefícios
  * Máxima privacidade: 1 instância por usuário
  * Sem prioridades
  * Pode usar os recursos como quiser

* Desvantagens
  * Custear todo sistema sozinho
  * O uso do sistema não é o mais eficiente

* Multi-tenant

* Cada empresa possui sua própria instância do aplicativo e infra-estrutura

* Benefícios
  * Economia de Hardware e energia (custo)
  * Esforço maior para atualizar
  * Backup e Redundância mais fáceis em relação ao Single-tenant

* Desvantagens
  * Menos customização específica
  * Menos autorização e Atraso de tempo (Recursos ou funcionalidades podem ser adiadas, empresas maiores ganham preferencia)

>>> Inquilino isolado

* Cada inquilito tem seu próprio stack de tecnologia, não havendo compartilhamento de recursos
* Para uma oferta SaaS, este modelo carece de agilidade e de elasticidade, porque adicionar um novo inquilino requer o provisionamento de sua própria instância de hardware e de software
* Embora não seja verdadeiramente Computação em Nuvem, é um passo nessa direção, oferecendo como atrativo a facilidade de uma rápida oferta para SaaS

>>> Multi-inquilino (Virtualização)

* Cada inquilino tem seu próprio stack de tecnologia, mas o hardware é alocado dinamicamente a partir de um pool de recursos, via mecanismos de virtualização
* Bastante similar ao modelo anterior, mas permitindo elasticidade na camada do hardware
* Entretanto, apresenta limitações, pois a unidade de alocação e liberação de recursos é a máquina virtual onde aplicação vai operar.

>>> Multi-inquilino via container

* Vários inquilinos são executados na mesma instância de um container de aplicação (um servidor de aplicações), mas cada inquilino está associado a uma instância separada do software de banco de dados
* O ambiente de execução é compartilhado entre vários inquilinos, mas a plataforma de dados é a mesma
* Premissa do modelo é que o isolamento do banco de dados garante integridade dos dados dos inquilinos, ao mesmo tempo em que o container de execução, por ser compartilhado, oferece as vantagens de elasticidade e de customização

>>> Multi-inquilino via todo o stack de software compartilhado

* É uma evolução do modelo anterior, agora com todo o stack de software sendo compartilhado
* Neste modelo, além do container da aplicação, também uma única instância do banco de dados é compartilhada por todos os inquilinos
* Vídeo explicativo

>>> Devo migrar?

* **Custo real:** Verificar se o modelo atual usado pela empresa tem um custo mais alto do que o modelo de computação em nuvem
* **Confiabilidade:** É muito importante avaliar a reputação do provedor de nuvem, e também as políticas de segurança desse provedor
* **Legalidade:** Nem todas as empresas podem mover suas aplicações para nuvens públicas, e um dos motivos são os fatores legais, regulamentações do tipo de negócio ou país que a empresa opera, que não permitem que os dados estejam localizados fora do país.

>>> Custo real

* Deve ser levado em conta:
  * Quanto de armazenamento será necessário
  * Qual o poder computacional vai precisar como processamento e outros
  * Quanto de tráfego vai utilizar
  * O valor de licença de software
  * Contratar pessas para desenvolver aplicações para nuvem? Capacitar a equipe?
  * Investir dinheiro em certificações e para se adaptar às regulamentações da empresa
  * Custos inesperados como customização de aplicações
  * Transferência de dados
  * Custos de validação
  * Outros
* Após somar tudo isso certifique que o ROI (retorno sobre o investimento) seja favorável para a migração.

>>> Custo real

* Se você quer reduzir custos operacionais de atualização, manutenção e licenciamento de software ou se você tem uma empresa pequena ou de médio porte mas não tem pessoal suficiente para manter a TI mas precisa de tecnologia de ponta, ou se a empresa não dispõe de recursos para investir em infraestrutura e precisa de tecnologia de ponta o modelo ideal é a nuvem pública
* Mas se a empresa quer ter o controle de todo o datacenter, servidores, softwares, segurança ou por questões legais não pode hospedar seus serviços fora da empresa aí você deve utilizar nuvem privada
* Mas tem um outro caso que é a empresa que gosta de manter o controle dos dados locais mas também gostaria de oferecer alguns serviços que estão disponíveis em nuvem pública, neste caso você utiliza uma estrutura com nuvem híbrida, e isso é o que vem acontecendo com a maioria das empresas

>>> Custo real

* Responsabilidade do provedor: Você precisa ler o contrato que o provedor disponibiliza
* Recuperação contra desastre: Saber se o provedor tem um plano de contingência em caso de falha do serviço principal, isso vale mais para SaaS.
* Modelo de adoção suportados pelo provedor: Verificar, e se o provedor suporta a integração da nuvem pública com a sua nuvem privada para poder criar uma nuvem híbrida.
* Segurança dos dados: O que é responsabilidade do provedor e o que é sua responsabilidade, na maioria das vezes a segurança é compartilhado, o provedor disponibiliza as ferramentas, mas você precisa utilizá-las, conhecer as certificações que o provedor tem na área de segurança também é muito importante.
* Modelo de controle de identidade: pesquisar os tipos de controle de acesso fornecidos pela nuvem, saber se é possível fazer a integração de seus usuário locais com os usuários na nuvem, utilizando o mesmo modelo de autenticação.
* Manutenção dos serviços: Saber como são os procedimentos de manutenção, e isso serve para qualquer modelo de nuvem.
* Visão futura: É muito importante você saber quais são os projetos do provedor para o futuro, saber se tem algo que eles não ofereçam hoje mas vão oferecer no futuro, pois essa é uma parceria de longo prazo, não pense só no presente.
* Desempenho: Muitos provedores disponibilizam um período para você fazer testes e validar se o desempenho satisfaz as suas necessidades.
* Flexibilidade: Você deve saber se o seu provedor tem flexibilidade de customização, isso é muito importante principalmente para o modelo SaaS e também flexibilidade nos termos contratuais, isso pode tornar a negociação menos complicada.
* Segurança física: É muito importante procurar documentações e conhecer as certificações que provem a segurança física dos datacenters dos provedores.

>>> Service Level Agreement (SLA)

* Alguns exemplos de SLA da AWS e da Microsoft.
  * https://aws.amazon.com/pt/rds/sla/
  * https://aws.amazon.com/pt/ec2/sla/
  * https://aws.amazon.com/pt/s3/sla/
  * https://azure.microsoft.com/pt-br/support/legal/sla/virtual-machines/v1_6/
  * https://azure.microsoft.com/pt-br/support/legal/sla/storage/v1_2/
  * https://contaazul.com/termos/