>>> >>> AWS

Name: Henrique Tsuyoshi Yara (OPUS-software)
Date: January 20, 2023

Figure: AWS logo

>>> Índice

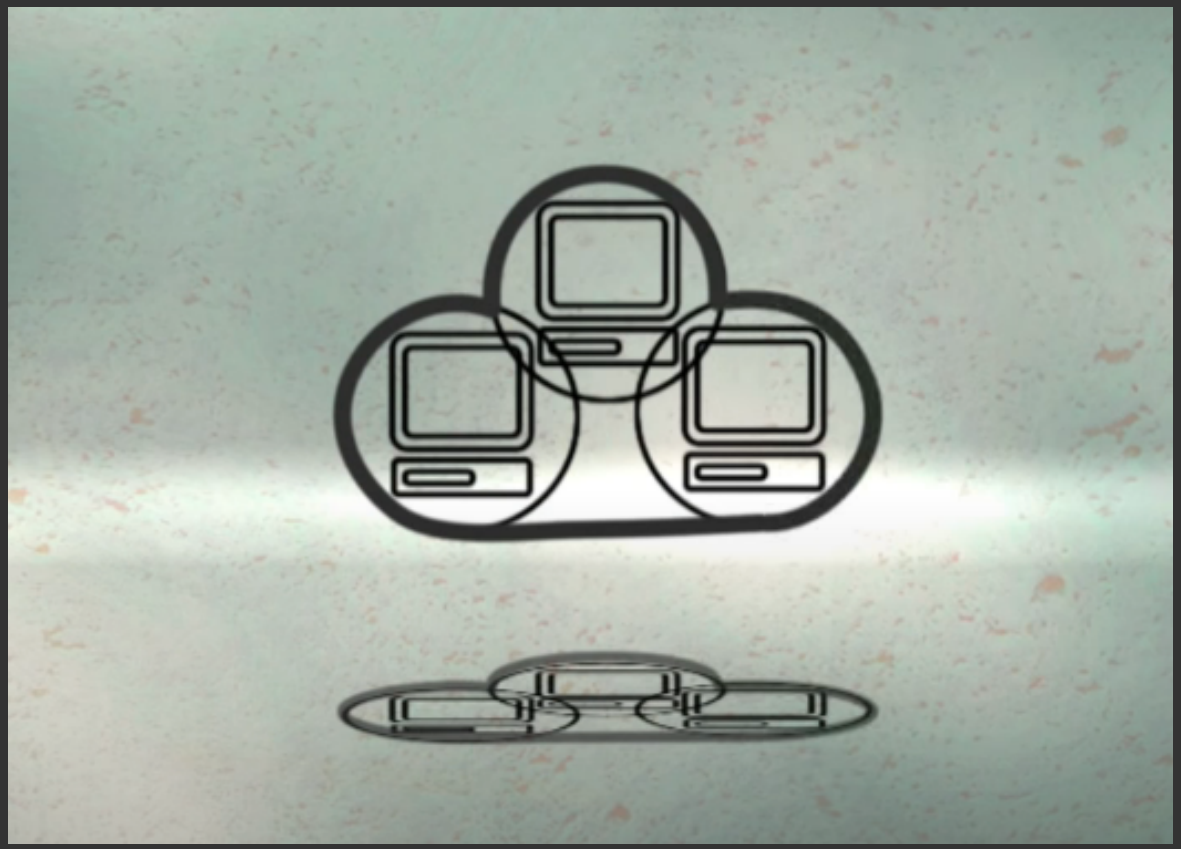
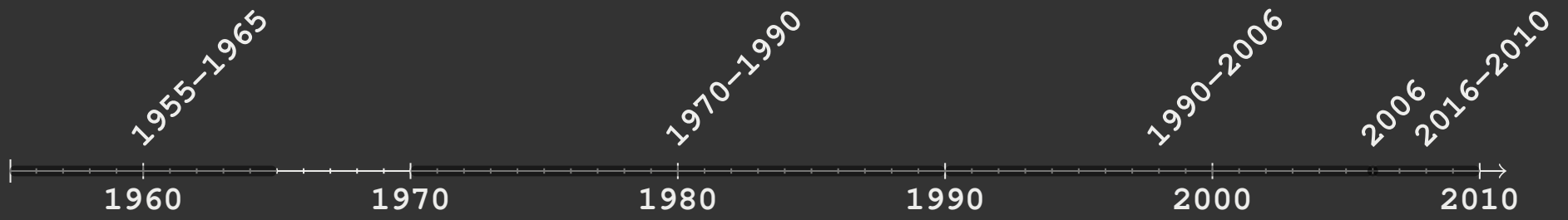


Figure: A cluster of servers drawn in a system diagram [1]

[1] Image source link

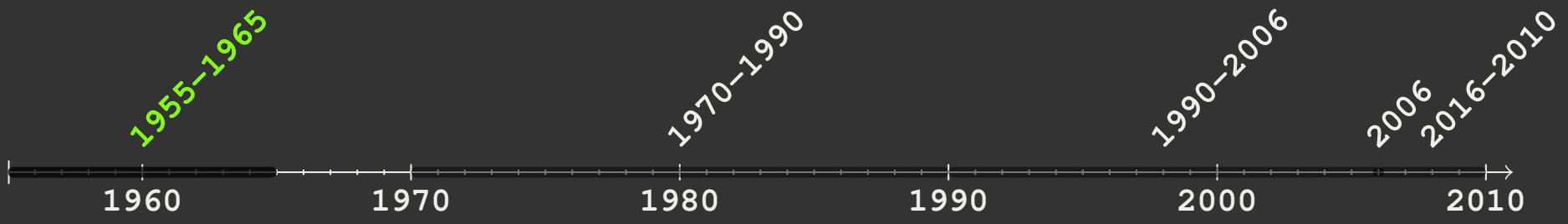
>>> Linha do tempo

História da computação em nuvem[2]



* (1955–1965) Problemas na infraestrutura de TI
* (1970–1990) Hypervisors e a internet
* (1990–2006) Internet para todos
* (2006–2006) Precipitation
* (2006–2010) Primeiros dias da computação em nuvem

>>> Linha do tempo



(1955–1965)

Problemas na infraestrutura de TI

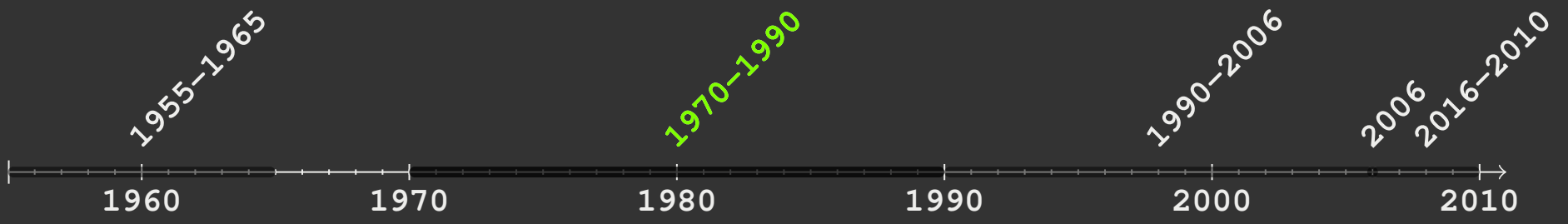
>>> (1955-1965) Problemas na infraestrutura de TI

* 1942 - John Vincent Atanasoff construiu o computador ABC
* Server/Mainframe (Componentes principais):
  * Uma CPU
  * Um dispositivo de armazenamento de dados
* 1960 - Muito caro para aderir os computadores
  * Sala inteira para o servidor (Manter temperatura ideal, espaço, etc...)
  * Computador
  * Funcionários especializados
  * Problemas para adaptar o software

>>> (1955–1965) Problemas na infraestrutura de TI

* Apenas empresas que com poder aquisitivo conseguiram aderir os computadores
* 1961 – John MacCharty fez uma palestra no MIT
  * Computação poderia ser vendida como água ou eletricidade [1]
  * Mas seria necessário de uma grande evolução tecnológica

>>> Linha do tempo



(1970–1990) Hypervisors e a internet

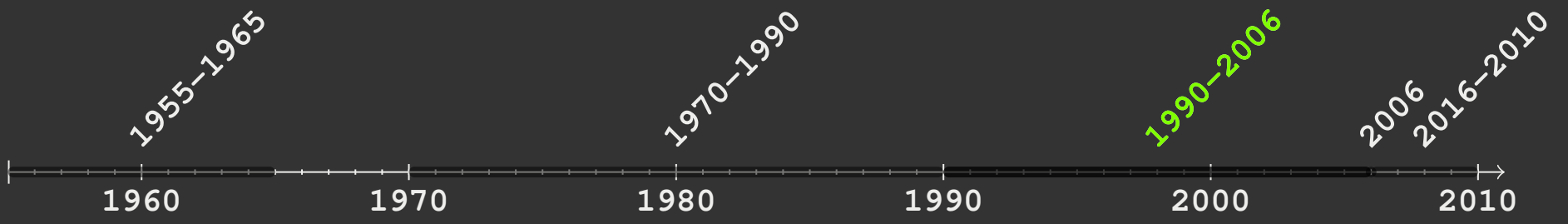
>>> (1970–1990) Hypervisors e a internet

* Diminiur os custos da adoção do computador
  * Múltiplos usuários podem compartilhar o mesmo armazenamento e o poder de processamento da CPU
* 1970 – Nasceu a tecnologia da virtualização
  * Um computador pode ser particionado em vários partes
  * Cada parte pode rodar um código independente
  * Cada parte é chamada de Máquina Virtual (VM)
  * O server principal é chamado de *host*
* O server tem um software que cria roda essas VMs virtualmente compartilhando seus recursos, dessa forma a máquina física se torna um hypervisor

>>> (1970-1990) Hypervisors e a internet

* 1983 - A internet nasceu
  * Começou pelo projeto ARPANet para comunicação de professores de universidades
* Foi introduzida a arquitetura cliente-servidor, o cliente e o server eram os mainframes interagindo com código e informação conectados por cabos (Internet)
* Conforme os números de páginas foram crescendo o número de servers cresceram
  * Os servers se moveram para datacenters

>>> Linha do tempo

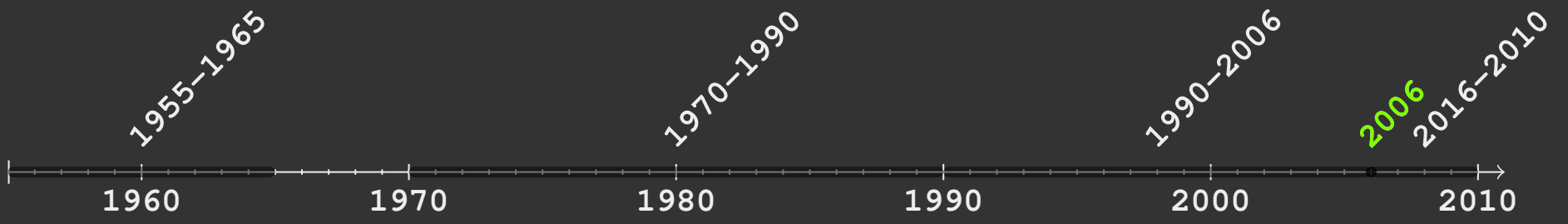


(1990-2006) Internet para todos

>>> (1990-2006) Internet para todos

* A empresas maiores precisavam de datacenters maiores
* Problemas:
  * If your website was not accessed much, all the hardware hosting your code and everything that comes with it would be unused. This would lead to software providers bleeding capital by keeping the servers on.
  * "Scaling" business for a software company was difficult as it required managing physical devices and real estate space. Consider facebook.com, a website with an incredible growth rate. Keeping up with that growth would be impossible, causing the website to crash (the case of Facebook's predecessor – Facesmash, the local intranet of Harvard university).
  * The speed of shipping updates was a problem since developers had to manually upload the code on the host server each time, to maintain its health and performance. Some days it would even involve marching into the data center with a screwdriver to

>>> Linha do tempo

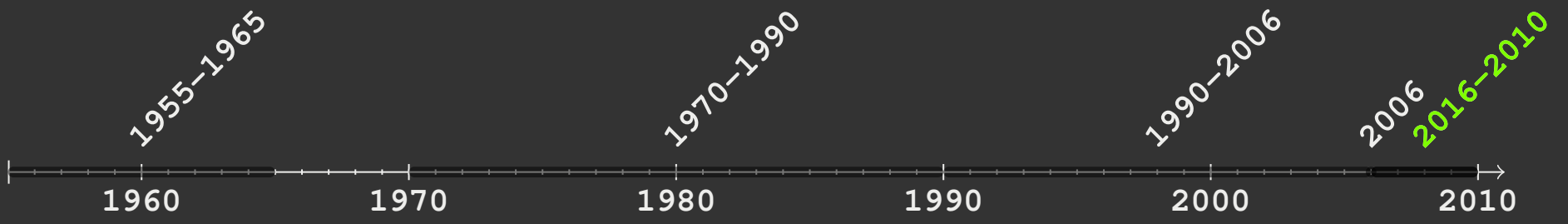


(2006) Precipitation

>>> (2006) Precipitação

* Grandes empresas (*Google, Amazon, eBay, etc...*)
  * *Também tiveram problemas, mas eles tinham dinheiro*
  * *Criaram data centers com centenas/milhares de servers de alta qualidade no mundo inteiro*
  * *Até mesmo eles tinham máquinas ociosas fora da temporada que só traziam prejuízos*
  * *Decidiram começar a alugar essas máquinas*
* *03/03/2006 - "Cloud Computing" foi introduzino no mundo como forma de aluguel de poder computacional*
* *A Amazon chamou essa difisão de Amazon Web Services (AWS)*
* *A AWS foi a primeira de muitos provedores de cloud*
* *Dessa forma empresas não precisam mais gerenciar sua própria infraestrutura de TI em data centers*
* *Depois de 50 anos o sonho de John McCharty foi realizado*

>>> Linha do tempo



(2006-2010) Primeiros dias da computação em nuvem

>>> (2006-2010) Primeiros dias da computação em nuvem

* Por 6 anos a AWS não teve competição e consegui estabelecer seu monopólio na área de nuvem
* A computação em nuvem permitiu que a empresa possa focar no desenvolvimento e deixar a parte da infra estrutura para a computação em nuvem
  * Os dados são criptografados e seguros
  * Dados são armazenados com redundância em nuvem
  * Escalabilidade
  * De fácil deploy
  * Alta disponibilidade

>>> (2006-2010) Primeiros dias da computação em nuvem

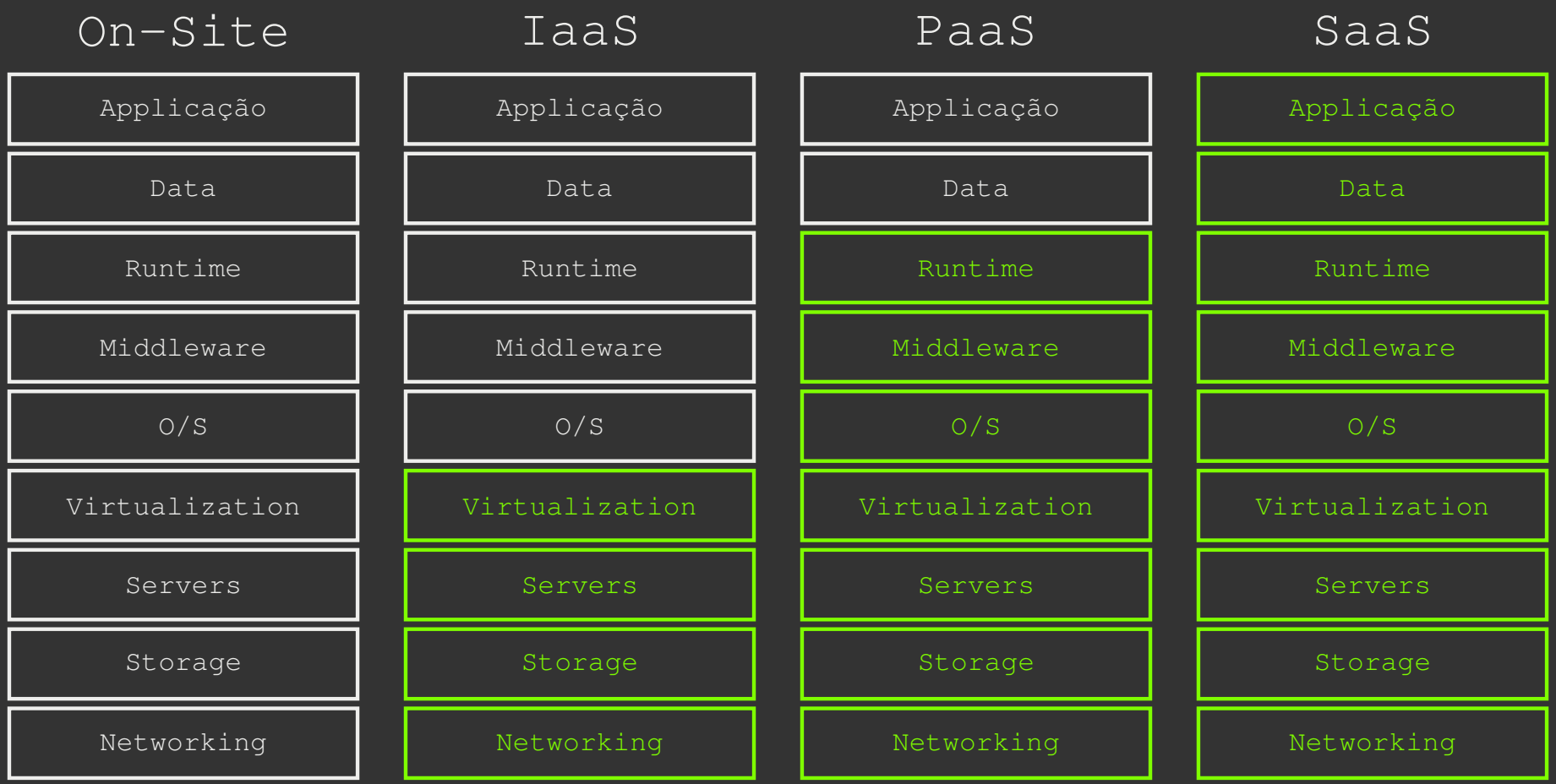


Figure: Imagem retirado do site da redhat[2]

[2] RedHat IaaS, PaaS e SaaS

>>> O que é cloud para o 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.

>>> 5 Características

* Conceito que reúne vários *softwares* e utiliza de virtualização
* Possui algumas características específicas (NIST):
    * Autoserviço sob demanda
    * Amplo acesso a rede
    * Pool de recursos
    * Rápida elaticidade
    * Serviços mensuráveis

>>> 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

>>> 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

>>> Referencias

[1] Simson Garfinkel. *The Cloud Imperative*. URL: https://www.technologyreview.com/2011/10/03/190237/the-cloud-imperative/ (visited on 01/20/2023).

[2] Ankit R Sanghvi. *History of Cloud Computing*. URL: https://www.cohesive.so/blog/the-history-of-cloud-computing (visited on 01/20/2023).