

Eficiência e escalabilidade em infraestrutura de TI



O mundo corporativo usa computadores há algumas décadas e seu uso passou de um modelo centrado apenas em custos para se tornar essencial na estratégia das empresas.

Ter uma estrutura informatizada (com computadores para as pessoas e servidores com as aplicações usadas na empresa) e com a maior automação possível (tarefas efetuadas por robôs de forma automática, ao invés de manualmente por uma pessoa) já é o mínimo para que as empresas tenham maior eficiência e deleguem aos funcionários as tarefas mais “nobres” como tomada de decisões, por exemplo.

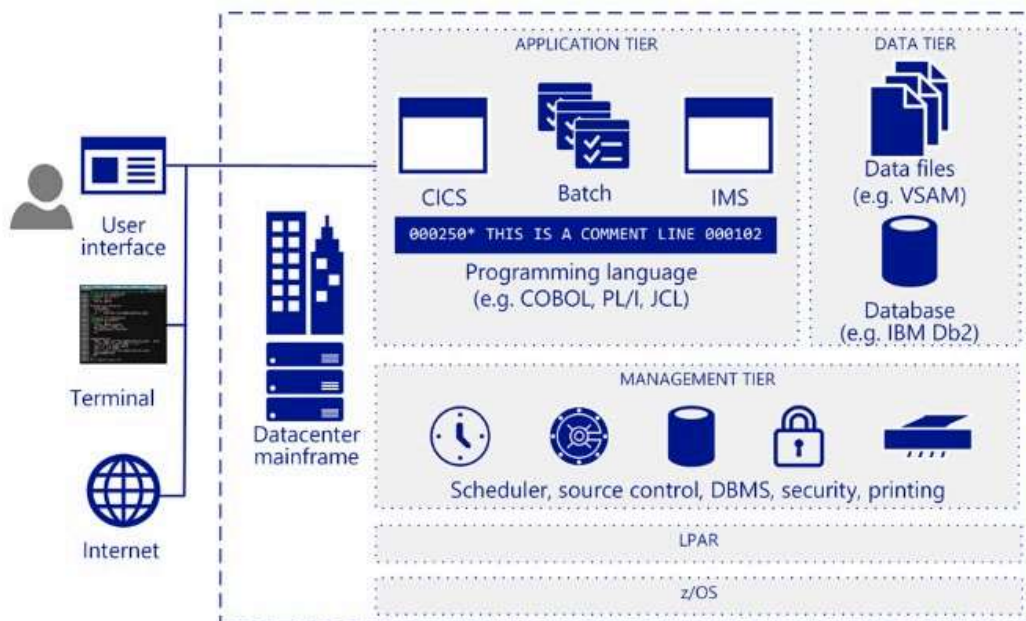
Mas muito do que acontece nos bastidores de uma área de TI (Tecnologia da Informação) nas empresas não fica evidente logo de cara e isso faz com que a área de TI seja muitas vezes vista apenas como um custo ao invés de ser vista como um diferencial.

É aqui que entram alguns detalhes destes bastidores quando falamos da infraestrutura de TI necessária para comportar as aplicações que os desenvolvedores criam para suprir tanto as necessidades de negócios (produtos e serviços baseados em TI para que a empresa atenda os clientes) quanto as necessidades internas (sistemas internos como por exemplo um sistema contábil ou de folha de pagamentos de funcionários).

Na história da área de TI passamos por alguns modelos de disponibilização de infraestrutura, muito relacionado às tecnologias disponíveis nas respectivas épocas e no conhecimento dos modelos de gestão de infraestrutura que foram evoluindo com o passar do tempo.

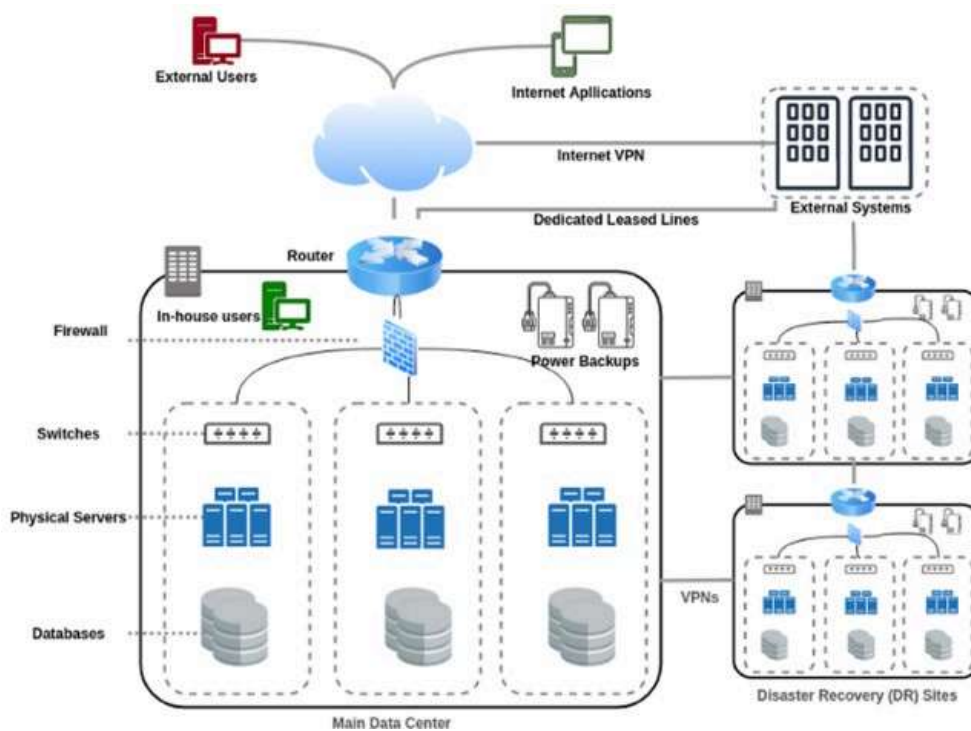
Abaixo estão descritos alguns desses modelos:

Mainframe: esse tipo de infraestrutura é antigo e bastante caro, dado a necessidade dos equipamentos precisarem de um local amplo e seguro, além do licenciamento da IBM! Toda infraestrutura necessária para “rodar” uma aplicação estaria no mainframe, o risco aqui é se o mainframe para, tudo que está lá é impactado junto. Um exemplo mais visual é demonstrado abaixo, onde todas as camadas de uma aplicação completa ficam em uma única infraestrutura:



Plataforma distribuída – On-Premise: A plataforma distribuída “nasceu” como uma opção mais “barata” se comparado ao mainframe, pois você pode segregar suas necessidades de infra em servidores distintos que usem um sistema operacional versátil e que não dependesse de uma estrutura específica (como o mainframe). Daí vieram servidores Windows e servidores com distribuições Linux (Red Hat, Suse, Fedora, Ubuntu, etc) onde você pode distribuir as aplicações necessárias entre vários servidores (aplicações, bancos de dados, etc). A característica do modelo On-Premise é ter a infra toda gerenciada pela empresa, portanto a empresa adquire/aluga locais para manter os servidores e todo o resto de infraestrutura (cabos de energia, de rede, switches, etc), estes locais são geralmente conhecidos como data centers. Nestes data centers compravam-se servidores muito “parrudos”

com meses de antecedência para que a aplicação pudesse “subir” no tempo certo, o problema era que esses servidores enormes ficavam com recursos ociosos, dado que eram comprados com a finalidade de se hospedar muitas aplicações, porém as aplicações eram implantadas aos poucos. Até que o servidor ficasse “cheio”, o servidor ainda estaria consumindo energia elétrica e pessoas dedicadas para mantê-lo disponível, causando desperdício.



- Plataforma distribuída – Cloud: Esta é a opção mais moderna atualmente e consiste em prover infraestrutura como serviço, então nesta opção as empresas podem “alugar” recursos sem a necessidade de manter um local físico (comprado/alugado), uma vez que os servidores, cabos de rede, energia, switches e etc podem ficar com o provedor do serviço chamado “nuvem”. A maior vantagem é que a empresa paga por aquilo que for usado, ao invés de eventualmente pagar para ter e manter um servidor físico como um recurso ocioso. Grandes empresas fornecem este tipo de serviço como a AWS (Amazon Web Services), Microsoft Azure e GCP (Google Cloud Platform), todas muito

similares mudando apenas o nome dos serviços e eventualmente alguma ou outra característica.

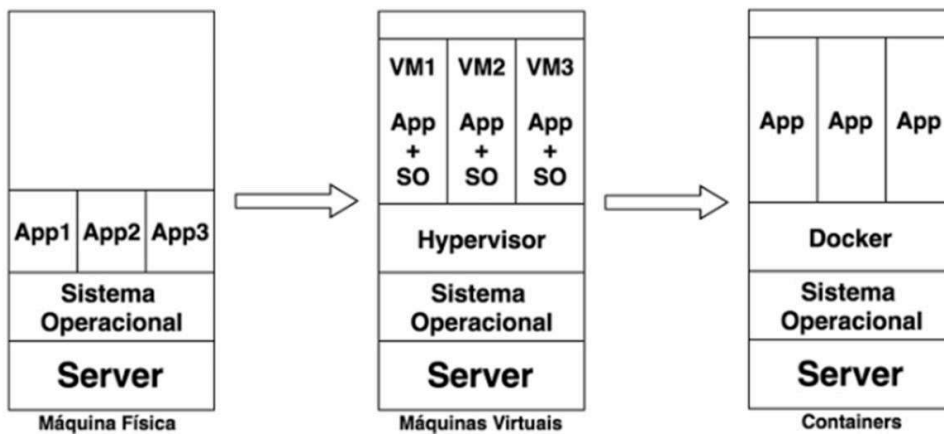
- Cloud privada: existe a opção da empresa manter sua infraestrutura física e prover o que chamamos de “nuvem privada”, que nada mais é do que usar um mesmo servidor como “serviço” para dentro da própria empresa, fornecendo recursos com o clique de um botão. Neste modelo todas as aplicações são mantidas como serviços da web (mesmo que não expostos para a internet, para uso dentro da rede interna da empresa), que é um passo importante para migrar estas mesmas aplicações para um serviço de nuvem pública (AWS, Azure, GCP, etc), uma vez que a aplicação já estará adaptada para isso. Este modelo existe para empresas que querem migrar para a nuvem pública, mas precisam “preparar o terreno” e ganhar experiência com o modelo de nuvem ou para empresas que precisam manter certos recursos de forma privada (em seus datacenters), só que num modelo que siga os preceitos de aplicação web. Aqui ainda é necessário manter uma infraestrutura própria como um data center por exemplo, onde além do prédio em si, há a necessidade de pagar imposto sobre o terreno, água, luz, seguranças, pessoas da limpeza, funcionários, equipamentos (servidores, extintores de incêndio, etc). A diferença para o modelo On-Premise é que aqui mesmo havendo servidores “parrudos” comprados e mantidos, o mesmo era subdividido em servidores menores através de tecnologias de virtualização, desta forma era como se um servidor enorme virasse uns 30 servidores menores (30 é um número fictício) com a vantagem de não precisar de espaço físico para tudo isso, afinal fisicamente era 1 único servidor, e esses eram distribuídos para várias equipes a fim de usar esse pedacinho do servidor. Só que ainda mantém-se o problema de haver ociosidade enquanto todos os “mini servidores” não forem utilizados.

- Cloud pública: se você utiliza algum provedor de serviço em nuvem como a AWS, Azure, GCP ou outros, então pode-se dizer que você está utilizando um serviço de “nuvem pública”. O uso da nuvem pública

requer que suas aplicações estejam na infraestrutura do fornecedor que escolheu, ao invés de ter as aplicações em infraestrutura própria. Os serviços são pagos conforme o uso e pode haver contratos de longa data que barateiam ainda mais estes custos. Neste modelo você não precisa ter uma infraestrutura própria como no modelo de “nuvem privada”, então você “barateia seus custos” por não precisar ter o local físico, pessoas para cuidar do local físico, servidores físicos, etc. O modelo é muito parecido com o de “nuvem privada” em que há um servidor físico “parrudo” dividido em “mini servidores”, porém o custo de manutenção do servidor físico (incluindo o local onde ele fica fisicamente) fica por conta do provedor de “nuvem pública”, é vantajoso pois a possibilidade de utilização de todos os “mini servidores” por vários clientes é mais rápido do que ter apenas 1 empresa para utilizar todos os “mini servidores”, então o provedor tem pouca ociosidade que se traduz em eficiência financeira ao passo que para os clientes também há a vantagem de pagar apenas pelo “pedacinho” que se está utilizando, ao invés de manter 1 servidor ligado 24/7.

Vamos agora dar um foco maior nas estruturas de plataforma distribuídas em que as aplicações mais modernas estão inseridas, com o uso de tecnologia de containers, mas antes de mais nada há uma pergunta a ser respondida: porque surgiu essa necessidade?

Conforme você já deva ter notado até aqui, cada vez mais utilizamos infraestrutura de forma a baratear custos mantendo ou aumentando a eficiência para atender os negócios de uma empresa, num modelo distribuído onde a gestão de cada recurso é apartada garantindo menor desacoplamento entre recursos, maior manutenibilidade e segregação de responsabilidades, entre o uso de um modelo On-Premise e de nuvem privada ou pública, há a diferenciação na forma como implantamos nossas aplicações à depender de como a infraestrutura é constituída. Descomplicando tudo isso vamos dar uma olhada na imagem à seguir:



Na figura acima, a representação de Máquina Física (à esquerda) mostra o modelo On-Premise mais tradicional em que você mantém um servidor físico “parrudo” com um sistema operacional (ex: Windows Server) e várias aplicações “rodando” no mesmo sistema operacional. É sensato pensar que em algum momento o servidor estará com muito recurso ocioso (quando houver apenas 1 pequena aplicação) até que nós consigamos encher o servidor com várias aplicações. Outro ponto a se considerar é que como o sistema operacional é o mesmo para todas as aplicações, há um acoplamento de tecnologia que faz a aplicação ter que ser escrita para aquele sistema operacional específico. Outro ponto negativo é que se precisar adquirir mais um servidor, além da espera e do custo o mesmo iniciará seu ciclo de vida com poucas aplicações, causando ineficiência pelo uso (desperdício).

Já a representação de Máquinas Virtuais (ao meio) mostra o modelo On-Premise já mais evoluído ou um modelo de “nuvem privada”, onde também temos um servidor “parrudo” com um sistema operacional só que a diferença é que neste sistema operacional instalamos uma aplicação muito específica chamada com a tecnologia Hypervisor (ex: Hyper-V ou VMWare). Esta tecnologia permite usar os recursos físicos do servidor “parrudo” como CPU, memória RAM e disco para criar máquinas virtuais ou “mini servidores” gerenciados pelo Hypervisor. Isso possibilita que tenhamos vários “mini servidores” distintos dentro de uma mesma máquina física, cada um desses “mini servidores” pode abrigar um sistema operacional que pode ser diferente do HOST (o HOST é o sistema operacional do servidor físico que abriga o Hypervisor). Assim uma aplicação pode ter um “mini servidor” para ser chamado de seu, então eu poderia na prática ter 1 aplicação para 1 “mini servidor”,

removendo a dependência da aplicação precisar sempre estar acoplada e dependente do sistema operacional do HOST, onde agora podemos ter outros sistemas operacionais nos “mini servidores”. Ainda assim temos o problema de compra de um servidor “parrudo” e do tempo que levará até ter uma quantidade de “mini servidores” criados para usar todo o potencial do servidor físico (o HOST), gerando ainda assim ineficiência (desperdício). Outro ponto negativo é que como estamos falando de “mini servidores” com seus sistemas operacionais próprios e recursos separados para cada máquina virtual (CPU, memória RAM, disco, etc), ao ligar a máquina virtual é como se estivéssemos ligando um computador, ou seja, é necessário esperar o sistema operacional terminar de “carregar” para que a aplicação nela instalada possa ficar disponível, o que leva tempo. Outro ponto importante é que de tempos em tempos há a necessidade de aplicação de patches de segurança no sistema operacional, só que agora não temos apenas 1 sistema operacional (o HOST), temos além do HOST, todos os “minis servidores” com seus sistemas operacionais individuais para aplicarmos esses patches de segurança, o que demanda tempo e recursos para gestão de tudo isso.

Já a representação de containers (à direita) é o que temos de mais moderno atualmente pensando em infraestrutura. Da mesma forma que o Hypervisor, há uma aplicação muito específica que usa tecnologia de containers (ex: Docker) que diferentemente do Hypervisor em que você cria máquinas virtuais ou “mini servidores” com seu sistema operacional próprio, você cria containers que usam as capacidades do sistema operacional do HOST (do servidor físico, em que o Docker está rodando). Significa que teremos o ponto negativo das aplicações terem que estar preparadas para uso do mesmo sistema operacional do HOST, ou seja, se o HOST for um Windows Server, então só poderão existir aplicações para Windows, se for Linux só poderão existir aplicações para Linux, porém há mais pontos positivos do que negativos, vamos explorá-los:

1. A própria Microsoft disponibilizou uma tecnologia chamada WSL (Windows Subsystem for Linux ou Subsistema do Windows para Linux) que permite a execução de aplicações Linux em máquinas com Windows (ex: Windows Server),

se você estiver usando um sistema operacional da Apple (ex: MacOS), também fará proveito do Linux se estiver usando Docker

2. Isso significa que não importa o sistema operacional do HOST, você sempre poderá usar containers sobre o Linux, o que permite a portabilidade para qualquer tipo de servidor

3. A grande maioria das ferramentas, aplicações, etc baseadas em containers já são feitas baseadas no Linux, sendo este sistema operacional a melhor opção

Com o uso de tecnologia de containers, você não precisa se preocupar em manutenção de sistemas operacionais de cada container, uma vez que os containers usam os recursos do sistema operacional HOST (CPU, memória RAM e eventualmente disco), é diferente do modelo com Hypervisor em que você tem várias máquinas virtuais apartadas, cada um “subindo” seu sistema operacional.

Por esta razão uma aplicação em container “sobe” quase que instantaneamente, pois não tem o tempo de inicialização de um sistema operacional, já que usa os recursos do sistema operacional HOST (do servidor físico) que já está ativo! Assim sendo a aplicação de patches poderia ficar restrito apenas ao sistema operacional HOST.

A tecnologia de containers é amplamente utilizada e pode existir em modelos On-Premise, em “nuvem privada” e em “nuvem pública”, assim sendo portar uma aplicação em container entre modelos torna-se muito fácil pois o Docker é instalável em praticamente qualquer sistema operacional.

Atividade Extra

Indico a leitura de alguns artigos complementares sobre os assuntos discorridos nesta aula:

O que é Subsistema do Windows para Linux?, da Microsoft Docs: <https://docs.microsoft.com/pt-br/windows/wsl/about>

The Twelve-Factor App (os famosos 12 fatores para uma aplicação moderna): https://12factor.net/pt_br/

Highlights das tecnologias, conceitos e desafios da computação em nuvem, de Bruno Oliveira: <https://medium.com/internet-das-coisas/cloud-02-highlights-das-tecnologias-conceitos-e-desafios-da-computa%C3%A7%C3%A3o-nas-nuvens-22ceff63b6c4>

Referência Bibliográfica

ARIYARATHNE, Jagath. On-premise, Cloud or Hybrid?, 19 de outubro de 2019. Seção On-premise deployments. Disponível em: <https://medium.com/@jagathsisira/on-premise-cloud-or-hybrid-5708f5e1fd1a>. Acesso em: 21 de maio de 2022.

Visão geral da migração de mainframe. Disponível em <https://docs.microsoft.com/pt-br/azure/cloud-adoption-framework/infrastructure/mainframe-migration/>. Acesso em 21 de maio de 2022.

VITALINO, J. F. N.; CASTRO, M. A. N. Descomplicando o docker. 2.ed. Brasport: 2018.

Ir para exercício