Semana 26 - Aula 1

Tópico Principal da Aula: Arquitetura de Aplicações na Nuvem: Conceitos Fundamentais

Subtítulo/Tema Específico: Escalabilidade e Disponibilidade

Código da aula: [SIS]ANO1C2B4S26A1

Objetivos da Aula:

- Compreender os conceitos de arquitetura de aplicações em nuvem, focando em **escalabilidade** e **disponibilidade**.
- Conhecer técnicas de computação e gerenciar dados para soluções em nuvem, dimensionando-as de acordo com as necessidades do negócio.

Recursos Adicionais (Sugestão, pode ser adaptado):

- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: Slide 08 - Arquitetura de Aplicações em Nuvem

- **Definição:** A **Arquitetura de Aplicações em Nuvem** é um conjunto de princípios e práticas que guiam o design e a implementação de sistemas de software utilizando a infraestrutura de computação em nuvem. Os dois conceitos cruciais que regem essa arquitetura são a **escalabilidade** e a **disponibilidade**.
- **Aprofundamento/Complemento (se necessário):** O objetivo principal é aproveitar a elasticidade da nuvem (recursos sob demanda) para criar sistemas que sejam resilientes a falhas, seguros e otimizados em custo. Arquiteturas modernas frequentemente utilizam conceitos como **microsserviços** e **contêineres** para atingir esses objetivos.
- **Exemplo Prático:** A arquitetura de uma plataforma de streaming de vídeo como a Netflix deve ser capaz de garantir um desempenho consistente e baixa latência, mesmo com a distribuição global e o acesso simultâneo de milhões de usuários.
    - **Vídeos Sugeridos:**
        - O que é Scalability and High Availability in the CLOUD Computing?
        - Horizontal Vs Vertical Scaling Under 1 Minute

Referência do Slide: Slide 09 - Arquitetura de Aplicações em Nuvem: Escalabilidade

- **Definição:** A **Escalabilidade** é a capacidade de um sistema de se adaptar à demanda, aumentando ou diminuindo seus recursos conforme necessário para lidar com variações na carga de trabalho. O seu objetivo central é a garantia de **desempenho consistente**.
- **Aprofundamento/Complemento (se necessário):** A escalabilidade é uma das grandes vantagens da computação em nuvem (*cloud computing*). Em um ambiente

de nuvem, a escalabilidade é frequentemente automatizada (Auto-Scaling), permitindo que o sistema responda dinamicamente a picos de tráfego, como um lançamento de produto ou uma campanha promocional.

- **Exemplo Prático:** Durante o lançamento de um novo jogo online, o número de usuários ativos salta de 10 mil para 1 milhão. A arquitetura de nuvem escalável deve automaticamente adicionar novos servidores em minutos para suportar o tráfego extra, e removê-los quando a demanda diminuir, controlando os custos.
    - **Vídeos Sugeridos:**
        - Horizontal and Vertical Scaling (How to Scale Your Application) - System Design

Referência do Slide: Slides 10 e 11 - Escalabilidade Horizontal e Vertical

- **Definição:** A escalabilidade é classificada em dois tipos principais:
    - **Escalabilidade Horizontal (Scale Out/In):** Envolve **adicionar ou remover instâncias** de servidores ou máquinas virtuais (VMs) para lidar com a carga de trabalho. Isso é alcançado por meio de técnicas como **balanceamento de carga** e **auto-escalabilidade**.
    - **Escalabilidade Vertical (Scale Up/Down):** Envolve **aumentar ou diminuir os recursos de uma única instância**, como a adição de mais CPU, memória (RAM) ou armazenamento.
- **Aprofundamento/Complemento (se necessário):** A Escalabilidade Horizontal é a preferida na nuvem por oferecer **resiliência** e virtualmente **escalabilidade ilimitada**, pois a falha de uma das instâncias não derruba o sistema. A Escalabilidade Vertical possui um limite físico (teto) de recursos de uma única máquina.
- **Exemplo Prático:**
    - **Horizontal:** Um grupo de 5 servidores de aplicação (VMs) se torna 10 em um pico de tráfego.
    - **Vertical:** Um servidor que originalmente tinha 8 GB de RAM é atualizado para 16 GB de RAM.
    - **Vídeos Sugeridos:**
        - Vertical Vs Horizontal Scaling: Key Differences You Should Know

Referência do Slide: Slide 14 - Arquitetura de Aplicações em Nuvem: Disponibilidade

- **Definição:** A **Disponibilidade** é a capacidade de um sistema de estar **acessível e operacional** para os usuários quando for necessário. O objetivo é **minimizar o tempo de inatividade** não planejado (downtime).
- **Aprofundamento/Complemento (se necessário):** A disponibilidade é frequentemente medida em termos de "noves" (ex: 99.99% de disponibilidade). Para atingir níveis muito altos, como 99.999% ("cinco noves"), a aplicação deve ser distribuída por múltiplas zonas ou regiões geográficas, garantindo que a falha de um *datacenter* não interrompa o serviço.
- **Exemplo Prático:** Um sistema de agendamento hospitalar deve ter alta disponibilidade, pois a inatividade pode afetar diretamente a saúde dos pacientes.

Ele é arquitetado com componentes duplicados e automação para garantir que esteja sempre no ar.
    - **Vídeos Sugeridos:**
        - Confiabilidade e redundância da CDN | Cloudflare

Referência do Slide: Slides 15, 16 e 17 - Práticas para Alta Disponibilidade

- **Definição:** As práticas essenciais para alta disponibilidade incluem:
    - **Redundância:** Duplicação de componentes críticos (servidores, discos, *load balancers*) para que, se um falhar, o outro assuma imediatamente.
    - **Balanceamento de Carga (*Load Balancing*):** Distribui o tráfego de forma eficiente entre os recursos disponíveis, evitando a sobrecarga de um único ponto e desviando automaticamente o tráfego de instâncias não saudáveis.
    - **Monitoramento e Recuperação Automatizada:** Sistemas que detectam falhas (saúde da instância) e acionam mecanismos automáticos de substituição de componentes ou redirecionamento de tráfego.
    - **Georreplicação:** Distribuição de dados e serviços em várias regiões geográficas. É a defesa final contra falhas regionais amplas.
- **Aprofundamento/Complemento (se necessário):** O Balanceamento de Carga é um ponto chave que atende tanto à escalabilidade horizontal quanto à alta disponibilidade, pois ele é a ponte que distribui a carga e gerencia o *failover* (troca para o componente de backup).
- **Exemplo Prático:** Uma empresa de serviços financeiros armazena dados em *buckets* replicados em três regiões. Caso uma região fique inativa devido a um desastre natural, o Balanceador de Carga redireciona todo o tráfego para as outras regiões de forma transparente para o usuário.
    - **Vídeos Sugeridos:**
        - Balanceadores de carga de rede de passagem interna e redes conectadas | Load Balancing | Google Cloud

---

Conteúdo Detalhado da Aula 2

Semana 26 - Aula 2

Tópico Principal da Aula: Provisionamento de Aplicações na Nuvem (Google Cloud)

Subtítulo/Tema Específico: Provisionamento de Máquina Virtual e Diagrama da Solução

Código da aula: [SIS]ANO1C2B4S26A2

Objetivos da Aula:

- Configurar uma aplicação na nuvem do Google (**Google Cloud Platform - GCP**) que seja escalável e com alta disponibilidade.

- Conhecer técnicas de computação e gerenciar dados para soluções em nuvem, parametrizando aplicações e dimensionando de acordo com as necessidades do negócio.

Recursos Adicionais (Sugestão, pode ser adaptado):

- Computador com internet e acesso ao console do Google Cloud (com conta de teste ou projeto ativo);
- Caderno para anotações.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - Diagrama da Solução (GCP)

- **Definição:** O **Diagrama da Solução** é o projeto visual da infraestrutura de *cloud*. Ele ilustra todos os componentes necessários para hospedar a aplicação na nuvem e como eles se comunicam, incluindo Máquinas Virtuais (**Compute Engine**), **Load Balancers**, e serviços de armazenamento ou banco de dados.
- **Aprofundamento/Complemento (se necessário):** O diagrama é fundamental para a comunicação entre arquitetos, desenvolvedores e operações (DevOps), garantindo que todos entendam a lógica de distribuição de tráfego e redundância da aplicação. Ele deve ser a primeira etapa no provisionamento de uma solução.
- **Exemplo Prático:** Um diagrama de solução para um **API Gateway** pode mostrar o tráfego entrando por um Balanceador de Carga, passando por um grupo de instâncias de VMs (Compute Engine) que rodam a API, e estas, por sua vez, se conectando a um **Cloud SQL** (banco de dados).
    - **Vídeos Sugeridos:**
        - Google Compute Engine in Google Cloud: Create first GCP VM Instance
        - How to create a GCE VM

Referência do Slide: Slide 07 - Provisionando a Máquina Virtual (VM)

- **Definição:** O **Provisionamento da Máquina Virtual (VM)** é o processo de criação e configuração do servidor (Instância) no **Google Compute Engine (GCE)**. Para arquiteturas escaláveis, as VMs são provisionadas dentro de um **Grupo de Instâncias (Managed Instance Group)**. Um elemento crucial nesse processo é o **Script de Inicialização (*Startup Script*)**.
- **Aprofundamento/Complemento (se necessário):** O *Startup Script* é um conjunto de comandos (shell script) que é executado automaticamente na primeira vez que uma VM é iniciada. Ele é usado para automatizar a instalação de softwares, dependências e a aplicação propriamente dita. Isso garante que qualquer nova VM adicionada pelo Auto-Scaling esteja imediatamente pronta para receber tráfego, sem intervenção manual.
- **Exemplo Prático:** Um *Startup Script* pode conter comandos para instalar o servidor web Nginx e copiar o código-fonte da aplicação de um repositório git ou de um

bucket de armazenamento, garantindo que o servidor esteja pronto para produção em segundos.

- **Vídeos Sugeridos:**
  - Google Cloud Compute Engine: Provisioning options
  - How to Create a Virtual Machine (VM) on Google Cloud Platform [GCP]

Conteúdo Detalhado da Aula 3

Semana 26 - Aula 3

Tópico Principal da Aula: Provisionamento de Aplicações na Nuvem (GCP) II

Subtítulo/Tema Específico: Acesso SSH e Armazenamento na Nuvem (Cloud Storage)

Código da aula: [SIS]ANO1C2B4S26A3

Objetivos da Aula:

- Como realizar o **acesso remoto (SSH)** à máquina virtual provisionada.
- Conhecer e preparar o serviço de **armazenamento de objetos (Bucket)** na nuvem.
- Conhecer técnicas de computação e gerenciar dados para soluções em nuvem.

Recursos Adicionais (Sugestão, pode ser adaptado):

- Computador com internet e acesso ao console do Google Cloud;
- Caderno para anotações.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - Acesso remoto pelo SSH

- **Definição:** O **SSH (Secure Shell)** é um protocolo seguro utilizado para estabelecer uma conexão remota e criptografada com a máquina virtual. Este acesso é essencial para o gerenciamento, diagnóstico de problemas (*troubleshooting*) e a execução de tarefas manuais de configuração na VM.
- **Aprofundamento/Complemento (se necessário):** No Google Cloud, a maneira mais simples de acesso é pelo **SSH-in-browser** diretamente pelo console, que gerencia automaticamente as chaves de segurança. Alternativamente, pode-se usar clientes SSH externos ou o **gcloud CLI** local para uma experiência de terminal completa.
- **Exemplo Prático:** Após provisionar a VM, o técnico precisa verificar se o servidor de aplicação subiu corretamente. Ele clica no botão "SSH" no console do Compute Engine e, no terminal que se abre, executa um comando como `tail -f /var/log/syslog` para monitorar os logs do sistema em tempo real.
  - **Vídeos Sugeridos:**
    - How To SSH into your VM - Google Cloud Platform (GCP)

        - How to connect to Google Cloud Virtual Machine using SSH from Terminal

Referência do Slide: Slide 07 - Preparando o Bucket e o SDK

- **Definição:** O **Cloud Storage** é o serviço de armazenamento de objetos da GCP, utilizado para guardar dados não estruturados com alta durabilidade e disponibilidade, como imagens, vídeos, ou backups. O **Bucket** é o contêiner fundamental onde os dados são armazenados. O **SDK (Software Development Kit)** é o conjunto de ferramentas para interagir programaticamente com os serviços da nuvem.
- **Aprofundamento/Complemento (se necessário):** O Cloud Storage é ideal para separar o armazenamento de dados estáticos do disco persistente da VM, o que aumenta a resiliência e a escalabilidade (o disco da VM é acoplado a ela, enquanto o Storage é acessível por todas as VMs). A preparação do **SDK** na VM é necessária para que o código da aplicação possa interagir com o *bucket* de forma autenticada.
- **Exemplo Prático:** Um sistema de gestão de documentos armazena todos os arquivos em um **Bucket** do Cloud Storage com política de retenção e redundância em várias regiões. A VM (Compute Engine) executa apenas o código que gerencia o acesso a esses arquivos, sem armazená-los localmente.
    - **Vídeos Sugeridos:**
        - Transferência de arquivos para VMs de Linux | Compute Engine Documentation - Google Cloud
        - Google Cloud Storage: O que é e como criar um Bucket (Exemplo de vídeo sobre Cloud Storage - Adapte o link para um de sua preferência)