



**INSTITUTO  
FEDERAL**

Rio Grande do Norte

Campus  
Natal-Central

## ATIVIDADE 2

INFRAESTRUTURA COM DOCKER

Alunos: Ana Julia, Laura Batista e Silvio Alexsander

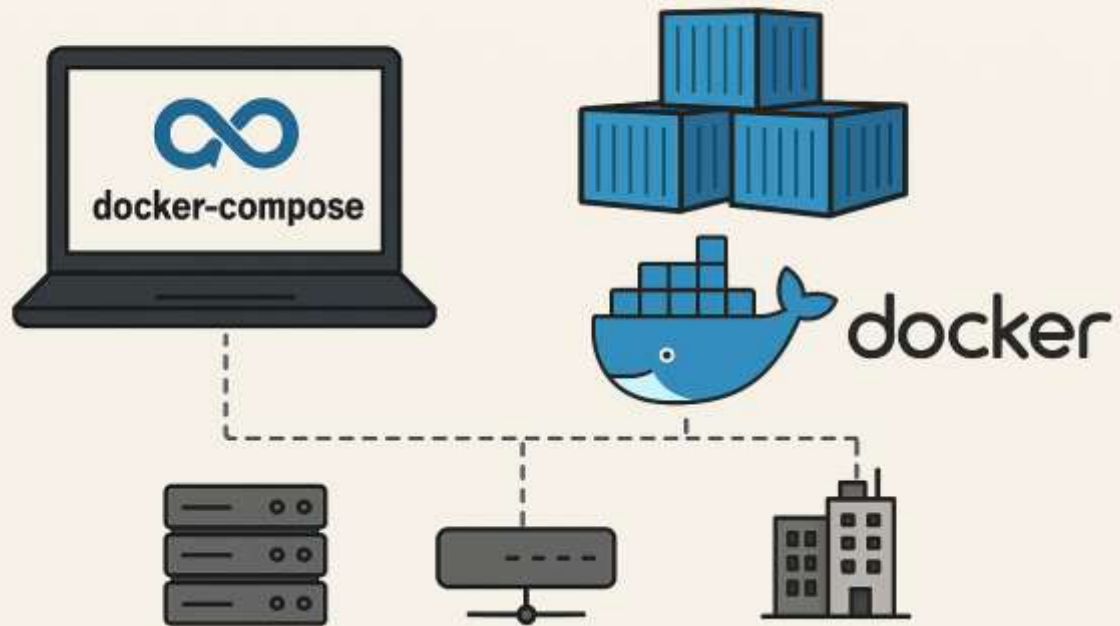
Professor: Sales Filho

Disciplina: Administração de Sistemas Abertos



# INTRODUÇÃO

## CONTAINER-BASED INFRASTRUCTURE



- Apresentar a construção de uma infraestrutura baseada em contêineres utilizando Docker e Docker Compose.
- **Componentes principais:**
  - Servidor DNS com zona primária
  - Servidor proxy reverso HTTP
  - Dois servidores web com páginas personalizadas



# CONCEITOS TEÓRICOS

## Servidor DNS com zona primária

- **DNS (Domain Name System):** Responsável por traduzir nomes de domínios para endereços IP.
- **Zona primária:** Base de dados principal de um domínio, armazenada no servidor DNS autoritativo.
- **Software utilizado (Bind9):** Servidor DNS amplamente utilizado para criar zonas personalizadas.

## Proxy reverso HTTP

- **Proxy reverso:** Intermediário que recebe requisições do cliente e as encaminha para servidores internos.
- **Vantagens:**
  - Balanceamento de carga
  - Cache de conteúdo
  - Segurança e anonimato
- **Software utilizado (Nginx):** Usado para direcionar as requisições HTTP para os servidores web.

## Servidores web com páginas personalizadas

- **Servidores Web:** Responsáveis por hospedar e entregar páginas HTML ao navegador.
- **Exemplos utilizados:** Apache HTTP Server, Nginx.
- **Páginas personalizadas:** Conteúdo único em cada servidor para demonstrar o roteamento feito pelo proxy.



## OBJETIVO DA ATIVIDADE



Criação de infraestrutura com  
Docker/Compose



Servidor DNS com zona primária



Servidor Proxy reverso



Servidor Web pagina personalizada

# CRIAÇÃO DE INFRAESTRUTURA COM DOCKER/COMPOSE

```
services:
  dns:
    build: ./dns
    container_name: dns_server
    ports:
      - "53:53/udp"
    networks:
      dns_net:
        ipv4_address: 10.5.0.10

  web01:
    build: ./web01
    container_name: web_server1
    networks:
      web_net:
        ipv4_address: 10.6.0.11

  web02:
    build: ./web02
    container_name: web_server2
    networks:
      web_net:
        ipv4_address: 10.6.0.12

  proxy:
    build: ./proxy
    container_name: proxy_server
    ports:
      - "80:80"
    depends_on:
      - web01
      - web02
    networks:
      web_net:
        ipv4_address: 10.6.0.20

networks:
  dns_net:
    driver: bridge
    ipam:
      config:
        - subnet: 10.5.0.0/24

  web_net:
    driver: bridge
    ipam:
      config:
        - subnet: 10.6.0.0/24
```

# SERVIDOR DNS COM ZONA PRIMÁRIA

## Arquivo Dockerfile:

```
1 # 1) Base: usamos uma imagem oficial do Ubuntu
2 FROM ubuntu:latest
3
4 # 2) Instalamos o serviço BIND9 (servidor DNS) e utilitários
5 RUN apt-get update && \
6     apt-get install -y bind9 dnsutils && \
7     rm -rf /var/lib/apt/lists/*
8
9 # 3) Copiamos a configuração de zona para dentro do container
10 COPY named.conf.local /etc/bind/named.conf.local
11 COPY db.asa.br /etc/bind/db.asa.br
12
13 # 4) Expomos a porta 53/UDP (porta padrão de DNS)
14 EXPOSE 53/udp
15
16 # 5) Comando para iniciar o BIND em primeiro plano
17 CMD ["named", "-g", "-c", "/etc/bind/named.conf.local"]
```

## named.conf.local:

```
1 // Definição da zona "asa.br"
2 zone "asa.br" {
3     type master; // é zona primária, onde se faz alterações
4     file "/etc/bind/db.asa.br"; // aponta para o arquivo de registros
5 };
```

# SERVIDOR DNS COM ZONA PRIMÁRIA

db.asa.br:

```
1      $TTL      604800      ; tempo de vida padrão dos registros
2      @          IN  SOA ns.asa.br. admin.asa.br. (
3                                  3      ; serial (a cada alteração, incremente este número)
4                                  604800  ; refresh
5                                  86400   ; retry
6                                  2419200 ; expire
7                                  604800 ) ; negative caching
8      ;
9      @          IN  NS  ns.asa.br.    ; servidor de nomes
10     @          IN  A   10.6.0.20     ; faz asa.br apontar para o proxy
11     ns         IN  A   10.5.0.10     ; IP do container DNS
12     www        IN  A   10.6.0.20     ; IP do proxy
```

# SERVIDOR PROXY REVERSO

dockerfile:

```
1  # 1) Base: usamos o Nginx oficial
2  FROM nginx:alpine
3
4  # 2) Copiamos nossa configuração personalizada
5  COPY default.conf /etc/nginx/conf.d/default.conf
6
7  # 3) Expomos porta HTTP
8  EXPOSE 80
9
10 # 4) O próprio Nginx, na imagem, já está preparado para iniciar
```

default.conf:

```
1  server {
2      listen 80;
3      server_name _;
4
5      # Se alguém acessar /web01 exatamente, redireciona para /web01/
6      location = /web01 {
7          return 301 /web01/;
8      }
9
10     # Tudo que vier em /web01/ é proxypass para o web01, removendo o prefixo
11     location /web01/ {
12         proxy_pass http://web01:80/;
13         proxy_set_header Host $host;
14     }
15
16     # Mesma lógica para web02
17     location = /web02 {
18         return 301 /web02/;
19     }
20
21     location /web02/ {
22         proxy_pass http://web02:80/;
23         proxy_set_header Host $host;
24     }
25 }
```



# SERVIDOR WEB 01

dockerfile:

```
1 FROM ubuntu:latest
2
3 RUN apt-get update && \
4     apt-get install -y nginx && \
5     rm -rf /var/lib/apt/lists/*
6 COPY index.html /var/www/html/index.html
7 EXPOSE 80
8 CMD ["nginx", "-g", "daemon off;"]
```

index.html:

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5     <meta charset="UTF-8">
6     <title>Web01 - Ubuntu</title>
7 </head>
8
9 <body>
10     <h1>Servidor Web 01</h1>
11     <p>Este é o container <strong>web01</strong> rodando em Ubuntu.</p>
12 </body>
13
14 </html>
```

# SERVIDOR WEB 02

dockerfile:

```
10 FROM debian:11
11 RUN apt-get update && \
12     apt-get install -y nginx && \
13     rm -rf /var/lib/apt/lists/*
14 COPY index.html /var/www/html/index.html
15 EXPOSE 80
16 CMD ["nginx", "-g", "daemon off;"]
```

index.html:

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4     <meta charset="UTF-8">
5     <title>Web02 - Debian</title>
6 </head>
7 <body>
8     <h1>Servidor Web 02</h1>
9     <p>Este é o container <strong>web02</strong> rodando em Debian.</p>
10 </body>
11 </html>
```

[LINK PARA REPOSITÓRIO](#)

## ASA-Entrega-02

O objetivo desta atividade é preparar uma infraestrutura com Docker Compose contendo, no mínimo:

1. Um servidor de DNS com uma zona primária
2. Um servidor proxy reverso HTTP
3. Dois servidores web com páginas personalizadas



### Estrutura do Projeto

```
ASA-Entrega-02/
├── compose.yaml          # Arquivo de definição do Docker Compose
├── dns/                  # Contêiner de DNS (BIND)
│   ├── Dockerfile        # Dockerfile para o servidor de DNS
│   ├── named.conf.local  # Configuração da zona no BIND
│   └── db.asa.br         # Arquivo de zona primária
├── proxy/                # Contêiner de proxy reverso (Nginx)
│   ├── Dockerfile        # Dockerfile para o proxy reverso
│   └── default.conf      # Configuração do Nginx
├── web01/                # Primeiro servidor web (Ubuntu)
│   ├── Dockerfile        # Dockerfile para o servidor web 01
│   └── index.html        # Página HTML personalizada
├── web02/                # Segundo servidor web (Debian)
│   ├── Dockerfile        # Dockerfile para o servidor web 02
│   └── index.html        # Página HTML personalizada
└── README.md             # Documentação deste projeto
```

# REFERÊNCIAS

- <https://www.docker.com/>
- <https://chatgpt.com/>
- <https://github.com/salesfilho>
- <https://github.com/AJCMelo>
- <https://github.com/laubatistaa>
- <https://github.com/SilvioAlexsander>
- **KUROSE, James F.; ROSS, Keith W.** Redes de computadores: uma abordagem top-down. 6. ed. São Paulo: Pearson, 2013.

OBRIGADO