

Arquivo Principal de Orquestração

Título: [docker-compose.yml](#)

Este arquivo é o coração da infraestrutura, pois reúne todos os serviços que formam o ambiente corporativo. Nele, definimos quais imagens Docker cada serviço irá utilizar sejam oficiais, personalizadas via Dockerfile ou de terceiros, quais volumes manterão os dados e configurações, além das redes virtuais responsáveis pela comunicação segura entre contêineres. Também estabelecemos dependências, garantindo que, por exemplo, o serviço web só inicie depois do DNS estar pronto, ou que o Samba aguarde o LDAP para autenticação. Uma configuração bem feita assegura que tudo suba na ordem certa, se fale de forma eficiente e reproduza fielmente um cenário de produção.

1. Serviço DHCP

Arquivos:

- [config/dhcp/dhcpd.conf](#)
- [config/dhcp/Dockerfile](#)
- [config/dhcp/dhcpd.leases](#) (+ backup [dhcpd.leases~](#))

1. dhcpd.conf

Define os detalhes de rede para atribuição dinâmica de IPs: faixas disponíveis, gateway padrão, máscaras de sub-rede e servidores DNS. Também podemos reservar endereços estáticos para equipamentos críticos (como servidores e impressoras), garantindo que sempre mantenham o mesmo IP.

2. Dockerfile (DHCP)

Cria a imagem personalizada do servidor DHCP (por exemplo, ISC DHCP Server). Ele instala o software, copia o [dhcpd.conf](#) e configura scripts de inicialização para que o serviço funcione automaticamente dentro do contêiner.

3. dhcpd.leases (+ dhcpd.leases~)

Armazena o histórico de concessões de IP: endereço liberado, MAC do cliente e período de aluguel. O arquivo com til (~) é um backup automático, útil na recuperação após falhas. Montá-lo em um volume Docker garante que o estado das concessões não seja perdido.

3. Serviço DNS

Arquivos:

- [config/dns/db.redes.local](#)
- [config/dns/named.conf](#) / [named.conf.local](#) / [named.conf.options](#)
- [config/dns/entrypoint.sh](#)
- [config/dns/Dockerfile](#)

1. db.redes.local

Contém os registros DNS que associam nomes internos (como [servidorweb.redes.local](#) ou [intranet.redes.local](#)) aos respectivos IPs, permitindo acesso por nome em vez de número.

2. named.conf & variantes

- [named.conf](#) agrega as configurações principais do BIND;
- [named.conf.local](#) declara as zonas gerenciadas (diretas e reversas);
- [named.conf.options](#) define regras globais, como servidores forwarders, recursão e ACLs para segurança.

3. entrypoint.sh

Roda assim que o contêiner inicia, ajustando permissões, validando arquivos de configuração e, por fim, executando o daemon do BIND ([named](#)).

4. Dockerfile (DNS)

Monta a imagem com BIND9 instalado, inclui todos os arquivos de zona e configuração e adiciona o [entrypoint.sh](#), garantindo consistência e isolamento no serviço de DNS.

4. Serviço FTP

Arquivos:

- [config/ftp/vsftpd.conf](#)

- [config/ftp/entrypoint.sh](#)
- [config/ftp/Dockerfile](#)

1. vsftpd.conf

Controla tudo no servidor FTP: habilita ou não acessos anônimos, define usuários virtuais, especifica diretórios home, gerenciar permissões de upload/download e configurar SSL/TLS para segurança das transferências.

2. entrypoint.sh

Responsável por criar usuários virtuais dinamicamente (via variáveis de ambiente ou arquivos de configuração), ajustar permissões e iniciar o daemon VSFTPD.

3. Dockerfile (FTP)

Instala o VSFTPD, copia [vsftpd.conf](#) e o script de inicialização, e já prepara os diretórios que serão montados via volumes para manter os dados persistentes.

5. Serviço LDAP

Arquivos:

- [config/ldap/entrypoint.sh](#)
- [config/ldap/Dockerfile](#)

1. entrypoint.sh

Prepara o ambiente LDAP: inicializa o diretório de dados, importa esquemas LDIF (definindo atributos de usuários e grupos) e popula entradas iniciais, como unidades organizacionais e administradores.

2. Dockerfile (LDAP)

Instala o servidor LDAP (por exemplo, [slapd](#)), copia os esquemas e o script de inicialização, e define variáveis de ambiente essenciais, como senha do admin e base DN. Os dados do LDAP são montados em volume para garantir persistência.

6. Serviço de Roteador/Firewall

Arquivos:

- [config/router/entrypoint.sh](#)
- [config/router/Dockerfile](#)

1. **entrypoint.sh**

Habilita o encaminhamento de pacotes no kernel (IP forwarding), aplica regras de firewall via iptables (ou nftables) para controlar tráfego entre redes interna, externa, DMZ e cuida do NAT para compartilhamento de internet, além de políticas de bloqueio de portas e IPs suspeitos.

2. **Dockerfile (Router/Firewall)**

Utiliza uma imagem leve (Alpine ou Debian slim), instala iptables e ferramentas de rede, e inclui o [entrypoint.sh](#). Exige privilégios elevados (NET_ADMIN) para manipular as interfaces e as regras de rede.

7. Serviço Samba

Arquivos:

- [config/samba/smb.conf](#)
- [config/samba/entrypoint.sh](#)
- [config/samba/Dockerfile](#)

1. **smb.conf**

Define os compartilhamentos de arquivos e impressoras via SMB/CIFS, mapeando pastas (ex.: [\\servidor\documentos](#)), gerenciando permissões por usuário/grupo e, se integrado ao LDAP, permitindo autenticação centralizada.

2. **entrypoint.sh**

Cria usuários Samba (mapeados a contas do sistema ou LDAP), ajusta permissões em diretórios compartilhados e inicia os serviços [smbd](#) e [nmbd](#).

3. **Dockerfile (Samba)**

Instala o Samba, inclui o [smb.conf](#) e o script de inicialização, e garante que as pastas compartilhadas sejam montadas em volumes para manter os dados fora do ciclo de vida do contêiner.

8. Serviço Web

Arquivos:

- [config/web/index.html](#)
- [config/web/Dockerfile](#)

1. index.html

Página estática simples, usada como placeholder ou para validar rapidamente se o servidor web está servindo conteúdo corretamente. Em um ambiente real, daria lugar ao portal ou aplicação interna.

2. Dockerfile (Web)

Configura um servidor HTTP (Nginx ou Apache), copia o [index.html](#) e outros ativos estáticos (ou o código da aplicação), e define como o servidor deve iniciar dentro do contêiner.