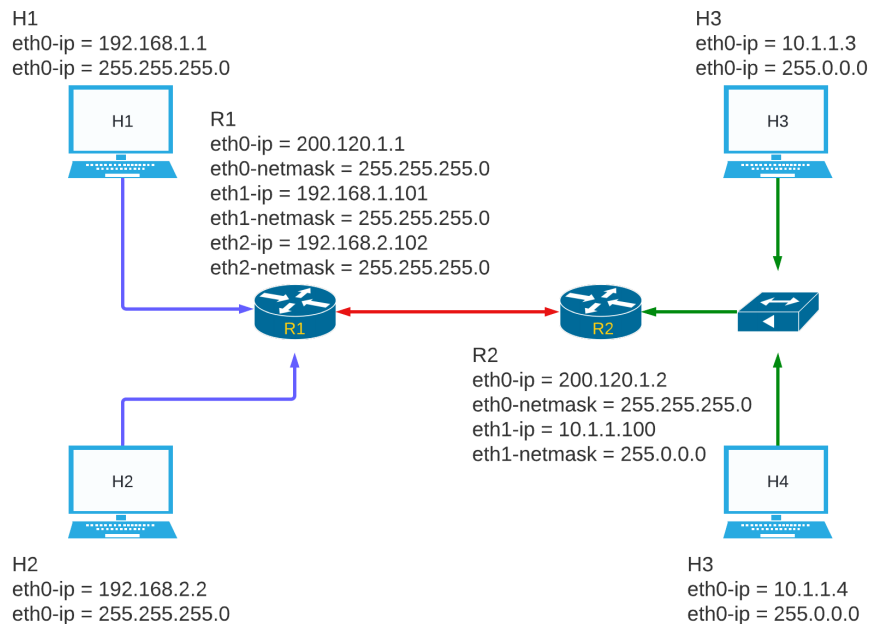


Aula Prática 02 - Conectando Redes utilizando roteadores

Objetivo: Interligar redes distintas, a partir da configuração de roteadores.

Figura 15 – Lab. 02 - Conectando redes através de roteadores.



Para executar os laboratórios do NetEnsina Docker você necessitará adquirir os arquivos previamente, para isso você pode baixá-los através do repositório do Github, procurando pelo diretório da aula que deseja praticar.

Sugerimos que mantenha a organização durante seus estudos, então você poderá, se assim desejar, criar uma pasta para armazenar os laboratórios. Apenas lembre-se de que as pastas do NetEnsina possuem a palavra **lab** em sua identificação.

Esta é uma aula direcionada ao aprendizado de roteadores, nela aprenderemos como configurar dois roteadores para interligarmos duas redes para viabilizarmos a comunicação entre 04 computadores.

Obs.: Para este laboratório destacamos o uso de duas máquinas virtuais destinadas a emulação dos roteadores presentes na topologia, desta maneira teremos resultados semelhantes ao uso de roteadores virtuais.

- **Iniciando o laboratório:**

Se você baixou os arquivos do laboratório através do Github Web deve possuir um arquivo compactado, se estiver utilizando Windows extraia o arquivo em sua pasta de laboratórios, em seguida clique com o botão direito em algum lugar vazio dentro da pasta deste laboratório e abra um terminal, após isso siga para o passo 3 deste documento. Para o Linux siga os passos abaixo:

1. Abra um novo terminal e acesse sua pasta de laboratórios: **cd /home/seunome/lab**
2. Em seguida utilize o comando de descompactação de arquivo: **tar -xf lab1.tar.gz**
3. Antes de avançar se certifique de estar dentro da pasta principal do laboratório, e localize o arquivo **docker-compose**. Use este comando: **ls**
4. Agora digite no terminal o comando a seguir para iniciar as máquinas virtuais do laboratório: **docker-compose up -d**

Serão inicializadas 6 máquinas virtuais, onde **R1** e **R2** representam os dois roteadores dentro topologia apresentada na imagem no início do roteiro. Consequentemente **H1**, **H2**, **H3**, **H4** representam os hosts conectados, onde **H1** e **H2** se conectam à mesma rede do primeiro roteador (**R1**), e **H3**, **H4** conectam-se ao segundo roteador (**R2**).

5. Abra novos terminais de maneira que possua quantidade suficiente para manipular todas as máquinas virtuais. Em seguida organize as janelas de maneira que possa visualizar todas corretamente.

- **Vamos Praticar:**

6. Em cada um dos terminais execute o seguinte comando: **docker container exec -it nomedamaquina /bin/bash**, modificando "**nomedamaquina**" pelos nomes citados anteriormente. Assim você ganhará acesso à elas.
7. Neste momento você pode executar o comando **ifconfig** em todas as máquinas e também nos roteadores, desta maneira você poderá observar as configurações de rede dos dispositivos.

Você poderá observar que todas as máquinas têm interfaces de rede **eth0** ativas, mas apenas os roteadores possuem interfaces adicionais, **eth1** em ambos, **eth2** apenas no **R1**. Isso ocorre pois o roteador **R1** se conecta às redes da máquina **H1** e **H2**.

8. Utilize o comando **ping ipdamaquina** da máquina **H3** para H4.

Lembre-se de substituir *ipdamaquina* pelos endereços IPs das máquinas.

9. Agora repita o comando **ping ipdamaquina**, só que agora da máquina **H1** para **H2**.

*Percebe que não há resposta? Isso ocorre pois **H1** e **H2** pertencem a redes diferentes e ambas as máquinas não conhecem o caminho que leva a informação a outra. Para contornar o problema utilizaremos os roteadores.*

10. Em cada máquina cheque os caminhos conhecidos com o caminho **route**. Você irá perceber que existe rota apenas para a própria interface de rede da máquina.

Sabendo disso, o próximo passo será demarcar as rotas por onde a informação deve ser direcionada.

11. Iniciando pelo host **H1**, use o comando: **route add default gw 192.168.1.101 dev eth0**

*Isto adiciona um **gateway default** em **H1**, assim ele poderá redirecionar as requisições que não conhece para um determinada rota.*

12. Em **H2** utilize, **route add default gw 192.168.2.102 dev eth0**

13. Em **H3** utilize, **route add default gw 10.1.1.100 dev eth0**

14. Em **H4** utilize, **route add default gw 10.1.1.100 dev eth0**

Com os hosts configurados podemos voltar nossa atenção para os roteadores.

15. Para **R1**, execute o comando de adição de rotas da seguinte maneira: **route add -net 10.0.0.0 netmask 255.0.0.0 gw 200.120.1.2 dev eth0**

*Esta rota define que tudo que **R1** receba destinado a rede 10.0.0.0 seja repassado para **R2**, dessa forma passa a ser tarefa do segundo roteador lidar com essas requisições.*

16. Em **R2**, utilize o comando: **route add -net 192.168.0.0 netmask 255.255.252.0 gw 200.120.1.1 dev eth0**

Com esta rota tudo está pronto, agora o **R2** pode enviar as requisições direcionadas às redes que **H1** e **H2** pertencem.

17. para fazer o monitoramento dos pacotes ative o **tcpdump** nos dois roteadores, use os seguintes comandos:

Em **R1**: **tcpdump -i eth0 -w /netensina/lab3-R1.pcap**

Em **R2**: **tcpdump -i eth1 -w /netensina/lab3-R2.pcap**

18. A partir do host **H4** faça um **ping** para H1. Observe que agora é possível a comunicação entre hosts de redes diferentes. Faça isso também entre **H1** e **H2** e observe os resultados. Lembre-se de encerrar o comando com **Ctrl+C** após algum tempo.
19. Encerre agora o **tcpdump** nos dois roteadores utilizando **Ctrl+C**.
20. Vá até a pasta **netensina** dentro da pasta do laboratório, nela você irá encontrar o arquivo **lab3-R1.pcap** e **lab3-R2.pcap**, use-o para estudar no software **Wireshark**.
21. Finalize o laboratório, utilize um terminal da sua máquina REAL e execute o comando: **docker-compose stop -d**