

Resumo – Modelo de Rede Docker

Este trabalho consiste em ler todos os conceitos e fazer um resumo dos itens 5 a 8 do link: <https://github.com/docker/labs/tree/master/networking/concepts>

Item 5: Bridge Networks.

Este item explica como funciona a rede bridge padrão do Docker e as que são implementadas pelos usuários.

A rede bridge padrão roda em qualquer host Docker, através de um driver de rede Linux. Esta rede padrão é responsável receber conexões dos containers caso nenhuma outra rede seja especificada. Dentro desta rede são criados atributos de rede de forma automática, como endereço MAC, iptables, DNS e o nome da rede bridge.

Em resumo, esta rede padrão habilita a comunicação entre os containers conectados nesta rede, impedindo também que containers externos, conectados em outra rede bridge, não se comuniquem com a rede em comum inicial.

Já as redes bridge definidas manualmente pelos usuários seguem o mesmo propósito de comunicação entre os containers, porém oferecem mais opções de customização, como a seleção de um driver diferente, definição de IP e subrede de forma manual, gerando um melhor isolamento dos containers.

A conectividade externa e interna nas duas redes possui um Firewall, fundamental para a segurança. Toda a comunicação de saída é permitida por padrão, mas toda a comunicação de entrada é bloqueada, com exceção de ser a mesma porta de saída, ou seja, a comunicação que saiu por uma porta, só é aceita pela mesma porta para a entrada.

Item 6: Overlay Networks.

O driver de rede de sobreposição ou Overlay, cria uma rede distribuída entre vários hosts do Docker. Esta rede sobrepõe (overlay) as redes internas do host, permitindo que os contêineres conectados à ela se comuniquem com segurança.

Assim como a rede bridge, a rede overlay também pode ser criada de forma manual pelos usuários. Os serviços ou contêineres podem ser conectados a várias redes ao mesmo tempo, porém só se comunicam entre si nas redes conectadas.

Estas redes de sobreposição simplificam as dificuldades encontradas em redes onde existem vários hosts. Utilizam um protocolo de tunelamento da camada 2 sobrepondo a camada 3, a VXLAN. Estas redes costumam ser usadas em uma conexão entre serviços Swarm.

Item 7: MACVLAN.

Em poucas palavras, MACVLAN serve para atribuir um endereço MAC para cada contêiner, ao invés de realizar o NAT, parecendo ser um dispositivo físico na rede, como uma máquina virtual. Isso se traduz em acesso direto aos

contêineres via rede física, permitindo que os contêineres recebam endereços IP da rede física.

É utilizado geralmente em aplicações antigas ou aplicações que monitoram o tráfego de rede, onde não aceitam a virtualização de rede ou é esperado uma conexão de rede física, então o MACVLAN entra para atender estas situações.

Item 8: Host (Native) Network Driver.

O driver nativo de Host conecta diretamente o contêiner à rede do host onde o daemon(emulador) do Docker está rodando, fornecendo performance de rede igual ao do host, para o container. Elimina a necessidade de roteamento, porém caso a porta já esteja sendo utilizada no host, o contêiner não será iniciado.

Isso significa que qualquer container usando o driver de rede do host, poderá se comunicar com qualquer container emulado no host.

Utiliza-se em aplicações de monitoramento, resolução de problemas ou onde se espera um nível elevado de performance.