Site-to-site networking

Use site-to-site layer 3 (L3) networking to connect two subnets on your Tailscale network with each other. The two subnets are each required to provide a subnet router but their devices do not need to install Tailscale. This scenario applies to Linux subnet routers only.

> This scenario will not work on subnets with overlapping CIDR ranges, nor with 4via6 subnet routing.

The subnet routers in this example are running Ubuntu 22.04 x64.

Step 1: Run Tailscale and specify network configuration

For this scenario, let's say you have two subnets with no connectivity between each other, and the subnet routes are 10.0.0.0/20 and 10.118.48.0/20.

1. For both subnets, choose a node to serve as a subnet router. For the 10.0.0.0/20 subnet, we'll use 10.0.0.2 as the subnet router. For the 10.118.48.0/20 subnet, we'll use 10.118.48.2 as the subnet router. On both subnet routers, install Tailscale, enable IP forwarding, and start the Tailscale client with the appropriate flags to serve as site-to-site networking subnet routers:

```
curl -sSL https://tailscale.com/install.sh | sh
echo 'net.ipv4.ip_forward = 1' | sudo tee -a /etc/sysctl.conf
echo 'net.ipv6.conf.all.forwarding = 1' | sudo tee -a /etc/sysctl.conf
sudo sysctl -p /etc/sysctl.conf
```

2. On the 10.0.0.2 device, advertise routes for 10.0.0.0/20:

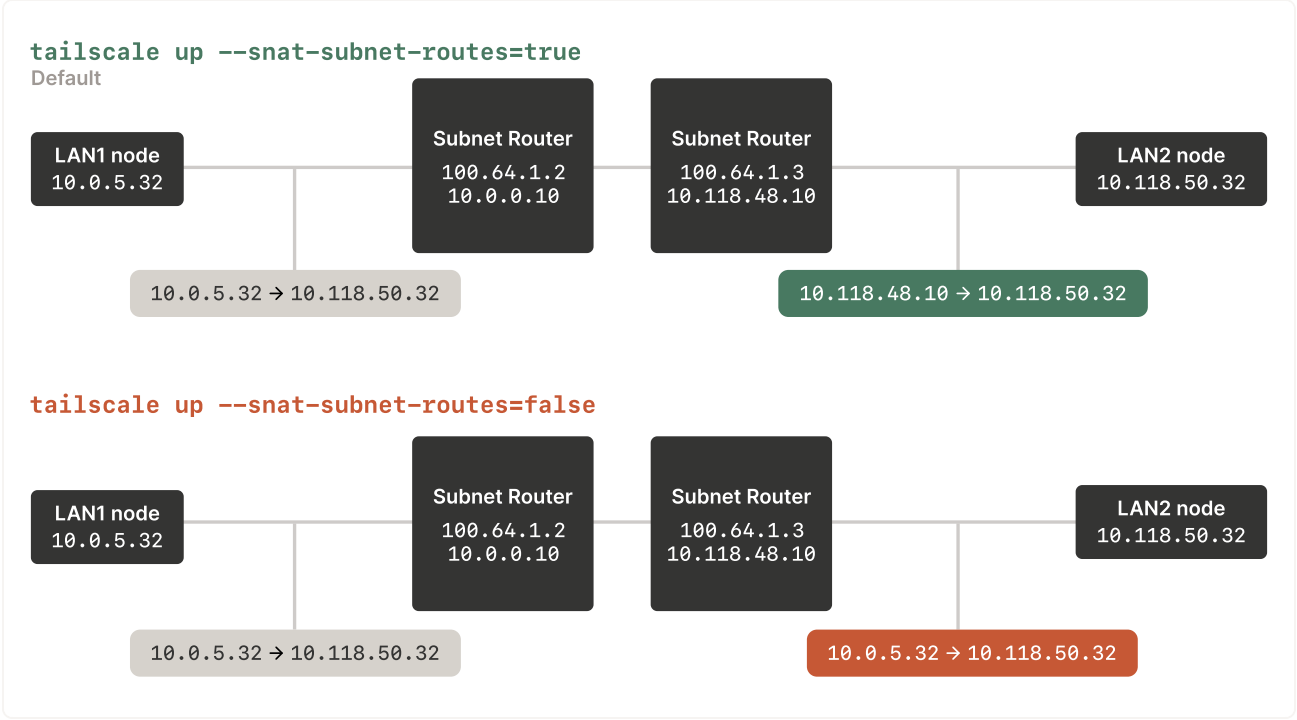
```
tailscale up --advertise-routes=10.0.0.0/20 --snat-subnet-routes=false --accept-routes
```

   The `tailscale up` command flags used are:

   `--advertise-routes`: Exposes the physical subnet routes to your entire Tailscale network.

   `--snat-subnet-routes=false`: Disables source NAT. In normal operations, a subnet device will see the traffic originating from the subnet router. This simplifies routing, but does not allow traversing multiple networks. By disabling source NAT, the end machine sees the LAN IP address of the originating machine as the source.

   `--accept-routes`: Accepts the advertised route of the other subnet router, as well as any other nodes that are subnet routers.



3 Likewise on the 10.118.48.2 device, advertise routes for 10.118.48.0/20:

```
tailscale up --advertise-routes=10.118.48.0/20 --snat-subnet-routes=false --accept-route
```

4 Configure both subnet routers to clamp the maximum segment size (MSS) to the maximum transmission unit (MTU):

```
iptables -t mangle -A FORWARD -i tailscale0 -o eth0 -p tcp -m tcp \
--tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Step 2: Enable subnet routes from the admin console

This step is not required if using `autoApprovers`.

Open the machines page in the admin console, and locate the devices that you configured as subnet routers. You can look for the **Subnets** badge in the machines list, or use the `has:subnet` filter in the search bar to see all devices advertising subnet routes. For each device that you need to approve, click the icon at the end of the table, and select **Edit route settings**. In the **Edit route settings** panel, approve the device.

The Tailscale side of the routing is complete.

> You may prefer to disable key expiry on your subnet nodes to avoid having to periodically reauthenticate. See key expiry for more information about machine keys and how to disable their expiry. If you are using ACL tags, key expiry is disabled by default.

Step 3: Configure the non-Tailscale devices

1 On each non-Tailscale device on the 10.0.0.0/20 subnet that you want to connect, you need to

1 On each non-Tailscale device on the 10.0.0.0/20 subnet that you want to connect, you need to add a static route to the tailnet and to the remote 10.118.48.0/20 LAN:

```
ip route add 100.64.0.0/10 via 10.0.0.2
ip route add 10.118.48.0/20 via 10.0.0.2
```

2 Likewise on each device on the 10.118.48.0/20 subnet that you want to connect, add a static route to the tailnet and to the remote 10.0.0.0/20 LAN:

```
ip route add 100.64.0.0/10 via 10.118.48.2
ip route add 10.0.0.0/20 via 10.118.48.2
```

Alternatively, the settings in this step can be set in your cloud environment routing, or on most routers by adding a "next hop" static route. For any of these techniques, you are selecting the remote network subnet as the target, and the LAN IP address of the local Tailscale subnet router as the router.

> The `ip route` commands on the client are not persistent—they need to be run again after rebooting To make the IP route settings persistent, you could add them to your network manager config or netplan config, depending on your setup. Alternatively, they can be managed by the DHCP server on your network.

You can now reach the LAN machines on either subnet, without requiring the LAN machines to have Tailscale installed or running. For example, on a device that is on the 10.0.0.0/20 subnet, you could ping a device on the 10.118.48.0 subnet (assuming both of these devices have added routes as described above):

```
# run this ping command from a device on the 10.0.0.0/20 subnet
ping 10.118.48.3

PING 10.118.48.3 (10.118.48.3) 56(84) bytes of data.
64 bytes from 10.118.48.3: icmp_seq=1 ttl=64 time=9.34 ms
64 bytes from 10.118.48.3: icmp_seq=2 ttl=64 time=3.85 ms
```

Last updated Sep 17, 2022

ON THIS PAGE

RELATED PAGES

4via6 subnet routers

Subnet router failover

Subnet routers and traffic relay nodes

Access AWS RDS privately using Tailscale

Access AWS RDS privately using Tailscale

Access Azure Linux VMs privately using Tailscale

tailscale