Tailnet Lock
Route Traffic
- Set up a subnet router
  - Subnet router BGP advertisement
  - 4via6 subnet routers
  - Site-to-site networking
- Set up an exit node
- Set up an app connector
- Use DNS
- Set up MagicDNS
- Set up high availability

Set Up Servers
- Set up a server
- Use tags
- Install Tailscale with cloud-init
- Use auth keys
- Automate key expiry
- Use Tailscale SSH
- Set up HTTPS certificates
- Run an ephemeral node
- Run unattended

Access & Share Services
- Share nodes

Search...

Docs › How-to Guides › Route Traffic › Set up a subnet router

Subnet routers

Subnet routers are available for all plans.

Subnet routers let you extend your Tailscale network (known as a tailnet) to include devices that don't or can't run the Tailscale client. They act as gateways between your tailnet and physical subnets, enabling secure access to legacy devices, entire networks, or services without installing Tailscale everywhere. This capability maintains Tailscale's security model while providing flexibility for complex network environments.

Why it matters

When designing a secure network, installing the Tailscale client directly on each device provides the best security and performance through end-to-end encryption. However, network administrators frequently encounter situations where direct installation isn't feasible. Devices like printers often lack the capability to run Tailscale, and in large environments such as AWS VPCs or legacy networks undergoing gradual modernization, installing clients on every endpoint becomes impractical.

Subnet routers bridge this gap by functioning as gateways that relay traffic between your tailnet and conventional subnet-based networks. They maintain Tailscale's security model by respecting access control policies while extending connectivity to non-Tailscale devices. This approach offers a practical balance between security and connectivity requirements.

An important consideration for organizations is that devices behind subnet routers don't count toward your pricing plan's device limit. Nevertheless, when possible, installing Tailscale directly on devices remains preferable for optimal performance, security, and configuration simplicity.

Benefits

The subnet router approach provides several important advantages for network administrators and organizations. Each benefit addresses specific challenges in modern network environments.

- **Connect legacy devices**—include devices that can't run the Tailscale client in your Tailscale network.
- **Integrate entire networks**—connect large networks, such as AWS VPCs, without installing Tailscale on each device.
- **Gradual deployment**—phase in Tailscale adoption by connecting existing network segments through subnet routers.
- **Maintain access control**—subnet routers respect Tailscale's access control policies, maintaining security across your network.

Use cases

Subnet routers solve practical problems in various network environments by extending Tailscale's secure connectivity model. These use cases represent common deployment scenarios where subnet routers provide substantial value.

- **Managed service access**—securely connect to cloud-managed services like Amazon RDS or Google Cloud SQL without exposing them to the public internet.
- **Cloud network integration**—seamlessly connect cloud VPCs or other cloud network segments to your Tailscale network.
- **Device connectivity**—make devices like printers or cameras accessible to remote Tailscale users without needing to install the Tailscale client.

How subnet routers work

Subnet routers function as networking bridges that connect separate network environments under a unified access model. They operate at the network layer to facilitate communication between your Tailscale network and traditional subnet-based networks.

A subnet router connects subnets, which are parts of a larger network. In Tailscale, a subnet router is a device in your tailnet that you use as a gateway to advertise routes to other devices. This lets devices connect to your tailnet without installing the Tailscale client.

Any device that uses the subnet router as a gateway is considered *behind* the subnet router. Subnet routers use Source Network Address Translation (SNAT) by default. When SNAT is enabled, traffic from a device behind a subnet router appears to come from the router itself, not the original device. If preserving the original source IP address is important for your use case, you can disable SNAT to maintain the original device's IP address in the traffic packets.

Subnet routers and exit nodes

Subnet routers and exit nodes serve different purposes in the Tailscale ecosystem, though they both involve routing traffic. Understanding the distinction helps you deploy the right solution for your networking needs.

Exit nodes route outbound internet traffic from your tailnet devices, effectively functioning as VPN servers. When you connect to an exit node, your internet traffic appears to come from the exit node's location. This is useful for accessing geo-restricted content or improving privacy. In contrast, subnet routers provide access to specific private subnets. They enable tailnet devices to reach non-Tailscale devices within those subnets, but don't affect internet traffic routing. If you need to access private networks like office LANs or cloud VPCs, subnet routers are the appropriate solution.

Set up a subnet router

Setting up a subnet router involves installing Tailscale on a device that will act as the gateway, configuring it to advertise routes, and ensuring proper access controls. This process requires administrative access to both the subnet router device and your Tailscale network.

You can use almost any device that runs the Tailscale client as a subnet router. To configure a device to run as a subnet router, use the instructions below or refer to the quickstart guide.

1. Install the Tailscale client.
2. Connect to Tailscale as a subnet router.
3. Enable subnet routes from the admin console.
4. Add access rules for advertised subnet routes.
5. Verify your connection.
6. Use your subnet routes from other devices.

Install the Tailscale client

The first step in creating a subnet router is installing the Tailscale client on the device that will serve as your gateway. Installation procedures vary by platform, but the process is straightforward across supported operating systems.

Linux | macOS | tvOS | Windows | Android

Download and install Tailscale onto the device you plan to use as a subnet router.

Connect to Tailscale as a subnet router

After installing Tailscale, you need to configure the device to function as a subnet router by enabling IP forwarding and advertising the subnet routes you want to make available. These steps transform a standard Tailscale node into a gateway for other networks.

Linux | macOS | tvOS | Windows | Android

To use a Linux device as a subnet router, you need to complete two essential configurations: enabling IP forwarding and advertising subnet routes. Linux devices make particularly good subnet routers due to their stability and networking capabilities.

1. Enable IP forwarding.
2. Advertise subnet routes.

Enable IP forwarding

When enabling IP forwarding, ensure your firewall denies traffic forwarding by default. This is the default setting for standard firewalls like `ufw` and `firewalld`. Blocking traffic forwarding by default prevents unintended routing of traffic.

IP forwarding is required to use a Linux device as a subnet router. This kernel setting lets the system forward network packets between interfaces, essentially functioning as a router. The process for enabling IP forwarding varies between Linux distributions. However, the following instructions work in most cases.

If your Linux system has a `/etc/sysctl.d` directory, use:

```
echo 'net.ipv4.ip_forward = 1' | sudo tee -a /etc/sysctl.d/99-tailscale.conf
echo 'net.ipv6.conf.all.forwarding = 1' | sudo tee -a /etc/sysctl.d/99-tailscale.conf
sudo sysctl -p /etc/sysctl.d/99-tailscale.conf
```

Otherwise, use:

```
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
```

If your Linux node uses `firewalld`, you might need to allow masquerading due to a known issue. As a workaround, you can allow masquerading with this command:

```
firewall-cmd --permanent --add-masquerade
```

Advertise subnet routes

After you enable IP forwarding, run `tailscale set` with the `--advertise-routes` flag. It accepts a comma-separated list of subnet routes.

```
sudo tailscale set --advertise-routes=192.0.2.0/24,198.51.100.0/24
```

Make sure to replace the subnets in the example above with the correct ones for your network. All platforms except Apple TV support both IPv4 and IPv6 subnets. Apple TV only supports IPv4 subnets.

If the device is authenticated by a user who can advertise the specified route in `autoApprovers`, the subnet router's routes will automatically be approved. You can also advertise any subset of the routes allowed by `autoApprovers` in the tailnet policy file. If you'd like to expose default routes (`0.0.0.0/0` and `::/0`), consider using exit nodes instead.

Enable subnet routes from the admin console

The admin console provides a centralized interface for approving and managing subnet routes advertised by your devices. This step ensures that the routes you've configured on your subnet router become active in your tailnet.

You can skip this step if you use `autoApprovers`.

1. Open the **Machines** page of the admin console.
2. Locate the **Subnets** badge in the devices list or use the `property:subnet` filter to list all devices advertising subnet routes.
3. Select a device with the `subnet` property, then go to the **Subnets** section.
4. Select **Edit**. This opens the **Edit route settings**.
5. Under **Subnet routes**, select the routes to approve, then select **Save**.

You can disable key expiry on your server to avoid having to periodically reauthenticate. If you use tags, key expiry is disabled by default.

Add access rules for the advertised subnet routes

Access controls determine which devices and users can access resources through your subnet router. Properly configured access rules are essential for maintaining security while enabling the connectivity you need.

You can skip this step if you already have rules that allow access to your advertised subnet routes.

1. Open the **Access controls** page of the admin console to update your tailnet policy file.
2. Create an access rule that lets access to the advertised subnet.

The following example tailnet policy configuration ensures members of `group:dev` can access devices in the subnets `192.0.2.0/24`, `198.51.100.0/24` and `2001:db8::/32`, and ensures the subnet `192.0.2.0/24` can access the subnet `198.51.100.0/24` and vice versa, *if subnet route masquerading is disabled.*

```
{
  "groups": {
    "group:dev": ["alice@example.com", "bob@example.com"]
  },
  "grants": [
    {
      "src": ["group:dev","192.0.2.0/24", "198.51.100.0/24"],
      "dst": ["192.0.2.0/24", "198.51.100.0/24", "2001:db8::/32"],
      "ip": ["*:*"]
    }
  ]
}
```

Verify your connection

Verification ensures that your subnet router is properly configured and functioning as expected. This step confirms that your tailnet devices can communicate with the subnet router before attempting to access resources behind it.

Check that you can ping the Tailscale IP address of your new subnet routers from a tailnet device (such as a Linux, macOS, or Windows device). You can find the Tailscale IP in the admin console or by running the following command on the subnet router.

```
tailscale ip -4
```

Use your subnet routes from other devices

Once your subnet router is configured and verified, you need to ensure that other devices in your tailnet can discover and use the new routes. This process varies slightly by operating system.

Android, iOS, macOS, tvOS, and Windows automatically pick up your new subnet routes.

By default, Linux devices only discover Tailscale IP addresses. To enable automatic discovery of new subnet routes on Linux devices, use the `--accept-routes` flag:

```
sudo tailscale set --accept-routes
```

Update subnet routes

Network requirements evolve over time, and you may need to modify the subnet routes advertised by your subnet router. This process involves updating the route advertisements and ensuring that the changes are properly approved and accessible.

To update subnet routes:

1. Connect to Tailscale as a subnet router.
2. Enable subnet routes from the admin console.
3. Add access rules for advertised subnet routes.
4. Verify your connection.
5. Use your subnet routes from other devices.

You can exclude any routes to prevent the subnet router from advertising them.

Use advanced subnet routing

After you set up a subnet router, you might consider:

- Route DNS lookups to an internal DNS server.
- Set up high availability for subnet routers.
- Connect two or more subnets using site-to-site networking.
- Disable source NAT (SNAT).

Route DNS lookups to an internal DNS server

DNS configuration lets your tailnet resolve names both for Tailscale devices and for resources on the advertised subnets. This capability enables seamless name resolution across your hybrid network environment.

You can add Tailscale IP addresses to public DNS records because Tailscale IP addresses are only accessible to authenticated users of your network. You can use an internal DNS server on your subnet by configuring split DNS in the **DNS** page of the admin console.

Set up high availability

For critical environments, redundant subnet routers provide reliability by ensuring continued connectivity even if individual subnet router devices fail. This approach is essential for production networks where continuous availability is required.

You can set up high availability to ensure your network is connectable even if one subnet router goes offline. For more information, refer to our topic on high availability failover.

When setting up subnet routers for high availability (HA), be careful with the `--accept-routes` flag. If you turn on `--accept-routes` for subnet routers that share the same routes in the same region, the standby router will accept its own advertised routes from the primary router.

This leads to an inefficient routing path. The standby router will send traffic for its directly connected subnet through the primary router instead. For example, if both subnet routers advertise and accept the same route, such as `192.168.1.0/24`, the standby router will send all `192.168.1.0/24` traffic through the primary router, even though it is directly connected to that network.

For most HA subnet router setups, use the `--advertise-routes` flag alone. Avoid using `--accept-routes` unless you specifically need that routing behavior.

Disable SNAT

Source Network Address Translation (SNAT) affects how source IP addresses appear to devices in different parts of your network. By default, Tailscale performs SNAT on traffic passing through subnet routers, but this behavior can be modified when necessary.

By default, when you advertise subnet routes, Tailscale uses source network address translation (SNAT) (also called masquerading). You can disable SNAT by using the `--snat-subnet-routes=false` flag (Linux only) with the `tailscale up` command. Disabling SNAT preserves the source IP addresses of the hosts behind the subnet router.

```
tailscale up --snat-subnet-routes=false
```

When you disable source NAT on a subnet router, devices behind it can access the Tailscale IP addresses of devices they connect to but don't automatically know how to route traffic back to those Tailscale IP addresses. To fix this, you must add a return route that tells the devices to send all Tailscale traffic through your subnet router. You can configure this route in one of three places:

- On the device's operating system
- In your VPC settings
- Through your DHCP server

The route should include:

- Network: `100.64.0.0/10` (the Tailscale IP address range)
- Next hop or gateway: The LAN IP address of your subnet router

The `--snat-subnet-routes` flag only works with Linux subnet routers.

Last updated Sep 30, 2025

RELATED PAGES
- Connect to an AWS VPC using subnet routes
- Connect to external services with IP block lists
- Exit nodes (route all traffic)
- Connecting without installing Tailscale
- Set up high availability

Ask AI