

OpenWRT NetBird

version 21

Latest version:

<https://raw.githubusercontent.com/egc112/OpenWRT-egc-add-on/main/notes/OpenWRT%20Netbird.pdf>

This is a W.I.P., I am working to make a real install guide, comments are welcome.

Introduction

NetBird is a modern, fully open-source Zero Trust VPN that provides a simple and secure way to connect users, devices, and services. It is built on WireGuard®, offering fast, encrypted peer-to-peer networking without the complexity of traditional VPN setups.

Unlike some alternatives, NetBird is fully open source and allows the entire control plane to be self-hosted. This can give full control over your networking infrastructure and data, and avoids dependency.

You can use it for remote access to your home network, to connect multiple routers and other clients (phone/PC/Mac etc.) and when setup as exit node also as a remote VPN.

Usually you can do the same by setting up your own WireGuard server and clients.

[WireGuard Server Setup Guide](#)

[WireGuard Client Setup Guide](#)

But this only works if you have at least a public IP address on one side of the connection.

If you are behind CGNAT, so do not have a public IPv4 address and also do not have a public IPv6 address (check with: *ifstatus wan6*) or using IPv6 is not applicable then you have to involve a third party as man-in-the-middle.

This can be a VPN provider which supports port forwarding (e.g. ProtonVPN), or you can rent a Virtual Private Server (I have an Oracle VPS which can be had for free, see at the bottom of this guide), or use things like [NetBird](#), [Zerotier](#) (also layer 2), [Cloudflared](#), [Tailscale](#) or [ngrok](#) and there are more.

I favor NetBird because it is fully open source and has some [advantages](#) over Tailscale, but it still is a commercial third party however a free tier is available which is sufficient for small to medium sized networks with up to 5 users (admins) and 100 peers (routers, VPS, client PC, phone, etc.).

Big advantage of NetBird is that it is fully open source and you can self host the control plane (Dashboard) in which case you are your own third party, self hosting the control plane is outside the scope of this paper.

Start with viewing: <https://docs.netbird.io/how-to/getting-started>

All the docs can be found at: <https://docs.netbird.io/>

A [NetBird wiki](#) is in the works

A very useful OpenWRT thread for discussion and support: <https://forum.openwrt.org/t/netbird-support-discussion-thread/237831>

Table of Contents

Introduction.....	1
Make a free account on NetBird.....	3
MFA (Multi Factor Authentication).....	3
Install NetBird on OpenWRT router.....	5
Setup on Router with SSO login.....	6
Setup on router with manual made key.....	7
Create a setup key for your OpenWRT router: NetBird Dashboard > Setup Keys:.....	7
Post installation.....	8
Network setup.....	9
Firewall setup.....	9
Check and Troubleshoot.....	11
Interface.....	11
Routing.....	11
Status.....	11
Info.....	11
WireGuard.....	11
Log.....	11
OpenWRT Forum.....	11
Online.....	11
Allow SSH access from Dashboard.....	12
Changes Starting with version 0.60.....	13
Create Routes.....	14
Create Networks.....	17
DNS settings.....	21
Create Exit node.....	22
Support.....	25
Known Problems.....	25
SSH-Access from Dashboard.....	25
Policy Based Routing (PBR) together with NetBird.....	25
Install on Oracle VPS with Ubuntu (24.04).....	25
Throughput improvements via transport layer offloading.....	26
Setup Oracle free OpenVPN cloud server.....	28
References.....	28
NetBird Releases.....	28
Configuration.....	28
Addendum 1.....	29

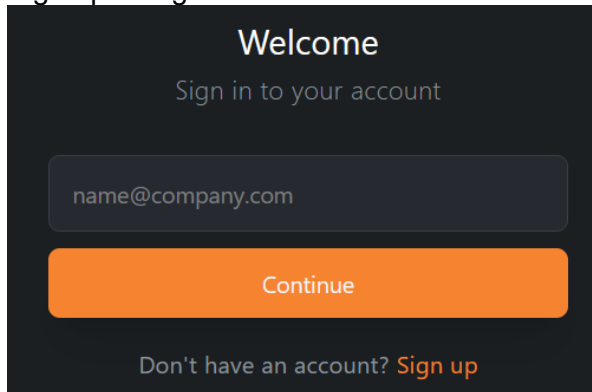
Make a free account on NetBird

go to: <http://netbird.io>

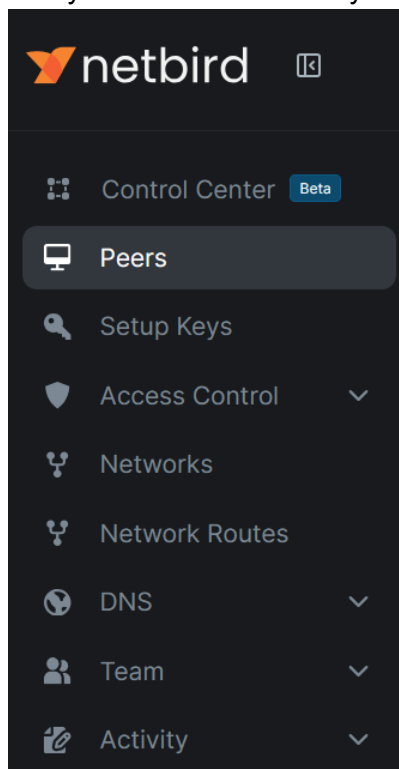
Click:



Sign up or login:



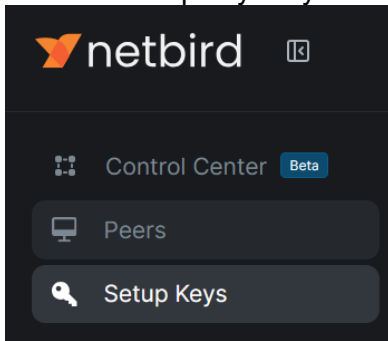
Now you are connected to your NetBird Dashboard the central administration (<https://app.netbird.io>):




MFA (Multi Factor Authentication)

If you click on your Icon in the upper right hand corner > Profile Settings: you can enable NetBird MFA for added security.


Create a setup key for your OpenWRT router: **NetBird Dashboard > Setup Keys:**



Fill in the name of your router and change the other items, shown are my settings, when done Click *Create Setup Key*.

**Create New Setup Key**
Use this key to register new machines in your network


Name
Set an easily identifiable name for your key


 **Make this key reusable**

Use this type to enroll multiple peers

☒


Usage limit
For example, set to 30 if you want to enroll 30 peers




Unlimited 


Peer(s)

Expires in
Days until the key expires.
Leave empty for no expiration.




Unlimited 

Day(s)

 **Ephemeral Peers**

Peers that are offline for over 10 minutes will be removed automatically



☐


 **Allow Extra DNS Labels**


Enable multiple subdomain labels when enrolling peers (e.g., host.dev.example.com).

☒


Auto-assigned groups
These groups will be automatically assigned to peers enrolled with this key

 Routing Peers 



Learn more about [Setup Keys](#) 

Cancel

 Create Setup Key

Copy and store the setup key

Install NetBird on OpenWRT router

NetBird is a rather large package around 12 MB written in Go so make sure your storage is sufficient, if you install it as a package it will take up double this storage, if your storage is not sufficient then make a new build using e.g. the [firmware selector](#) and *customize installed packages* adding *netbird*.

If you already have a running build then a very handy tool to make a list for the firmware selector is [owut](#) (*owut list*).

You can even use it for upgrading and adding a package like netbird (*owut upgrade --add netbird*).

Installing package

opkg (24.10):

opkg update

opkg install netbird

apk (25.12 and higher):

apk update

apk add netbird

NetBird is added as a service to OpenWRT which can be called from */etc/init.d/netbird*

See **service netbird help** for commands but the regular commands are available

Make the NetBird service start at boot up with **service netbird enable** but this should be done automatically.

Having the service start is imperative for the right config path as that is set in the service profile.

The NetBird executable is stored in */usr/bin/netbird*.

You can use **netbird help** to see the available commands e.g.:

netbird up/down/status etc.

The config file (>0.55) is stored in */root/.config/netbird*, you might add this path to */etc/sysupgrade.conf*, so that it is included in the backup!

Setup on Router with SSO login

After installing NetBird check that the OpenWRT service is running with `service netbird status`, if this is the case you can Run UP command to log in with SSO (interactive login): `netbird up`

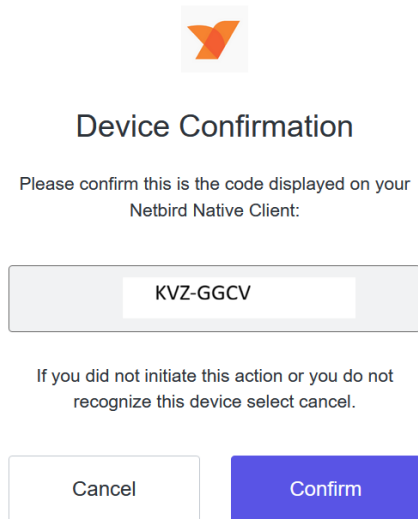
You will see the following text on the console:

Please do the SSO login in your browser.

If your browser didn't open automatically, use this URL to log in:

`https://login.netbird.io/activate?user_code=ZKZX-AACCV`

On your PC where you have opened the NetBird dashboard use your browser to login with the link (user_code) from the routers console and you should see:



Confirm the device and go to your NetBird Dashboard where you should see the new Peer been added

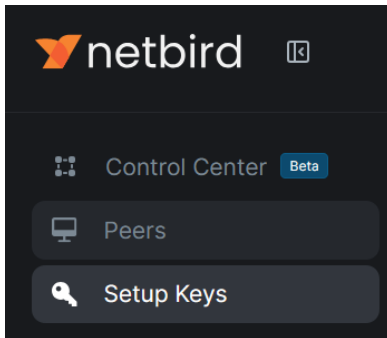
If this is successful proceed to [Post installation](#), [Network setup](#) and [Firewall setup](#)

|


If this does **not** work go to the next section, [Setup on router with key](#) and manually make a peer key on the Dashboard and use that peer key to setup NetBird.

Setup on router with manual made key


Create a setup key for your OpenWRT router: **NetBird Dashboard > Setup Keys:**



Fill in the name of your router and change the other items, shown are my settings, when done Click **Create Setup Key**.


 **Create New Setup Key**
Use this key to register new machines in your network

Name
Set an easily identifiable name for your key

 **Make this key reusable**
Use this type to enroll multiple peers


☒

Usage limit
For example, set to 30 if you want to enroll 30 peers




Peer(s)


Expires in
Days until the key expires.
Leave empty for no expiration.



Day(s)


 **Ephemeral Peers**
Peers that are offline for over 10 minutes will be removed automatically


☐

 **Allow Extra DNS Labels**
Enable multiple subdomain labels when enrolling peers
(e.g., host.dev.example.com).

☒


Auto-assigned groups
These groups will be automatically assigned to peers enrolled with this key

 Routing Peers ×



Learn more about [Setup Keys](#)

Cancel

 Create Setup Key

Copy and store the setup key

Head back to the console of the router and execute:

```
netbird up --setup-key <key from previous step>
```

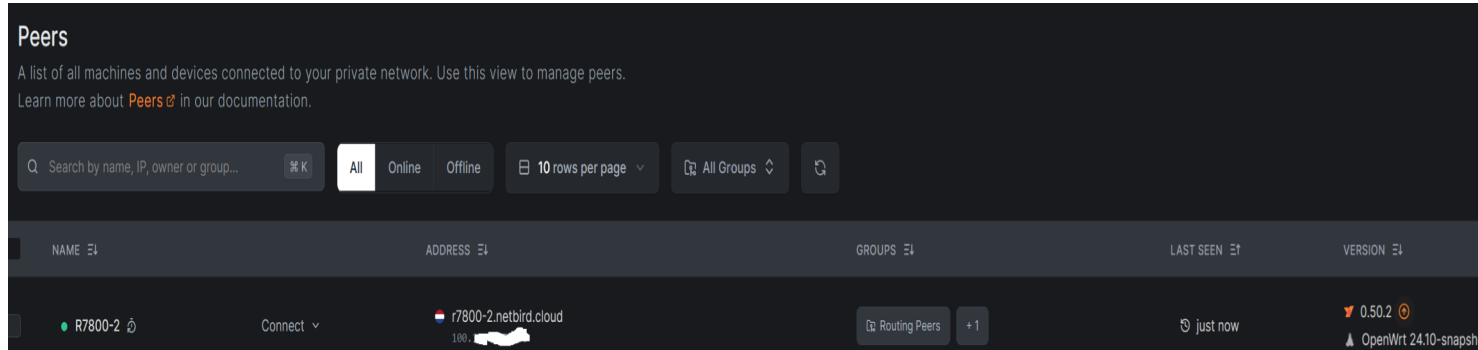
After some time you will see:

```
root@R7800-2:~# netbird up --setup-key E20033F4-0XXXXXXXXXXXXXXXXX
```

Connected

```
root@R7800-2:~#
```

In your Dashboard you can now see the installed peer



with ifconfig or ip address show on the router, you should see the new interface (device) **wt0**

|

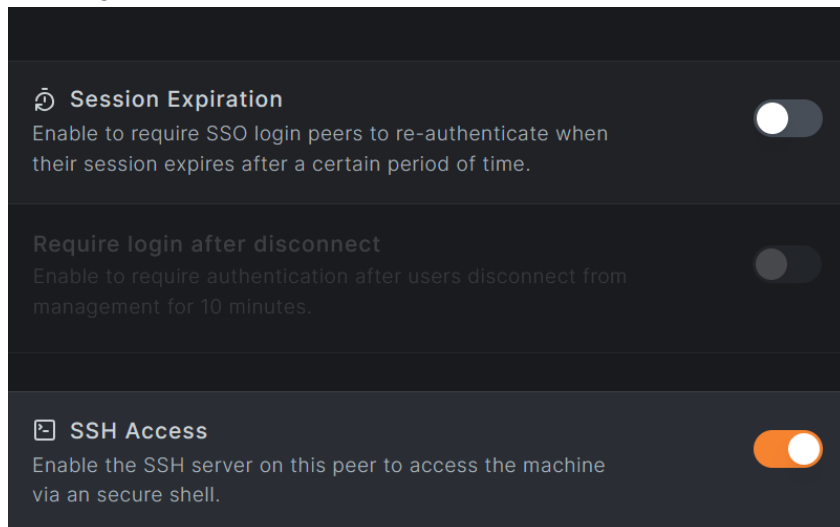
Check NetBird status with: *netbird status* and *service netbird status* both should output: *running*.

If not reboot and check again

Post installation

Open the Peer in the Dashboard by clicking on it in and change the settings.

You might want to disable Session Expiration and Enable SSH Access:



Network setup

Create a new unmanaged interface via LuCi: **Network > Interfaces > Add new interface**

- Name: **netbird1**
- Protocol: **Unmanaged**
- Device: **wt0** #For compatibility e.g. with e.g. PBR always name your interface e.g. **wtX**

Interfaces » netbird1

General Settings Advanced Settings Firewall Settings DHCP Server

Status

Device: wt0

Uptime: 0h 0m 9s

RX: 0 B (0 Pkts.)

TX: 0 B (0 Pkts.)

Protocol

Unmanaged

Device

wt0

Disable this interface

☐

Bring up on boot

☒

/etc/config/network:

```
config interface 'netbird1'
    option proto 'none'
    option device 'wt0'
```

Firewall setup

Create a new firewall zone via LuCi: **Network → Firewall → Zones → Add**

- Name: **netbird**
- Input: **ACCEPT** (default)
- Output: **ACCEPT** (default)
- Forward: **ACCEPT**
- Masquerading: **on** (might not be necessary)
- MSS Clamping: **on** (might not be necessary when using [0.59.12](#))
- Covered networks: **netbird1**
- Allow forward to destination zones: Select your **LAN** (and/or other internal zones or WAN if you plan on using this device as an exit node), as this is an exit node **WAN** is selected, if you are going to use a VPN and traffic from the exit node should go via this VPN then also add the VPN interface!
- Allow forward from source zones: Select your **LAN** (and/or other internal zones or leave it blank if you do not want to route LAN traffic to other NetBird hosts)

Click **Save & Apply**

Firewall - Zone Settings

General Settings Advanced Settings Conntrack Settings

This section defines common properties of "netbird". The *input* and *output* options set the default policies for traffic entering or leaving the zone. The *covered networks* option describes the policy for forwarded traffic between different networks within the zone. *Covered networks* specifies which networks are covered by this zone.

Name	<input type="text" value="netbird"/>
Input	<input type="text" value="accept"/>
Output	<input type="text" value="accept"/>
Intra zone forward	<input type="text" value="accept"/>
IPv4 Masquerading	<input checked="" type="checkbox"/> Enable network address and port translation IPv4 (NAT4 or NAT6) typically enabled on the wan zone.
MSS clamping	<input checked="" type="checkbox"/>
Covered networks	<input type="text" value="netbird1:"/>

The options below control the forwarding policies between this zone (netbird) and other zones. *Destination zones* cover forwarded traffic from other zones **targeted at netbird**. *Source zones* match forwarded traffic from other zones **targeted at netbird**. The forwarding rule is *unidirectional*, e.g. a forwarding rule to forward from wan to lan as well.

Allow forward to *destination zones*:

lan lan:	wg_stos_6: (empty)	ovpn_client tun1: (empty)	wg_mullv_se:	wg_proton_nl: (empty)	mullvad_ro: (empty)	wan wan:	wan6:
-----------------	--------------------	----------------------------------	--------------	-----------------------	---------------------	-----------------	-------

Allow forward from *source zones*:

lan lan:	wg_stos_6: (empty)
-----------------	--------------------

/etc/config/firewall:

```
config zone
    option name 'netbird'
    option input 'ACCEPT'
    option output 'ACCEPT'
    option forward 'ACCEPT'
    option masq '1'
    option mtu_fix '1'
    list network 'netbird1'
```

```
config forwarding
    option src 'netbird'
    option dest 'lan'
```

```
config forwarding
    option src 'lan'
    option dest 'netbird'
```

As this is an exit node traffic from netbird to wan is allowed

```
config forwarding
    option src 'netbird'
    option dest 'wan'
```

In the end **reboot** the router or do *service network restart*, *service firewall restart* and *service netbird restart*.

Check and Troubleshoot

Interface

`ip address show wt0`

```
31: wt0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1280 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/none
    inet 100.105.224.116/16 brd 100.105.255.255 scope global wt0
        valid_lft forever preferred_lft forever
```

Routing

`ip route`

```
default via 192.168.0.1 dev wan proto static src 192.168.0.9
100.105.0.0/16 dev wt0 proto kernel scope link src 100.105.224.116
```

Status

`netbird status --detail` to get a detailed status report which also will show if you have a fast P2P connection or a relayed connection.

Info

`service netbird info` to get the info from the OpenWRT ubus service e.g. the NB_STATE_DIR, `service netbird status` will show if the service is running. Use `service netbird help` for more commands

WireGuard

`wg show` this shows the connections to peers and the Allowed IPs which should also show the subnet routes if you made any.

Log

`cat /var/log/netbird/client.log`

You can increase the log level by starting NetBird with `--log-level notice|debug` (see: <https://docs.netbird.io/get-started/cli>)

OpenWRT Forum

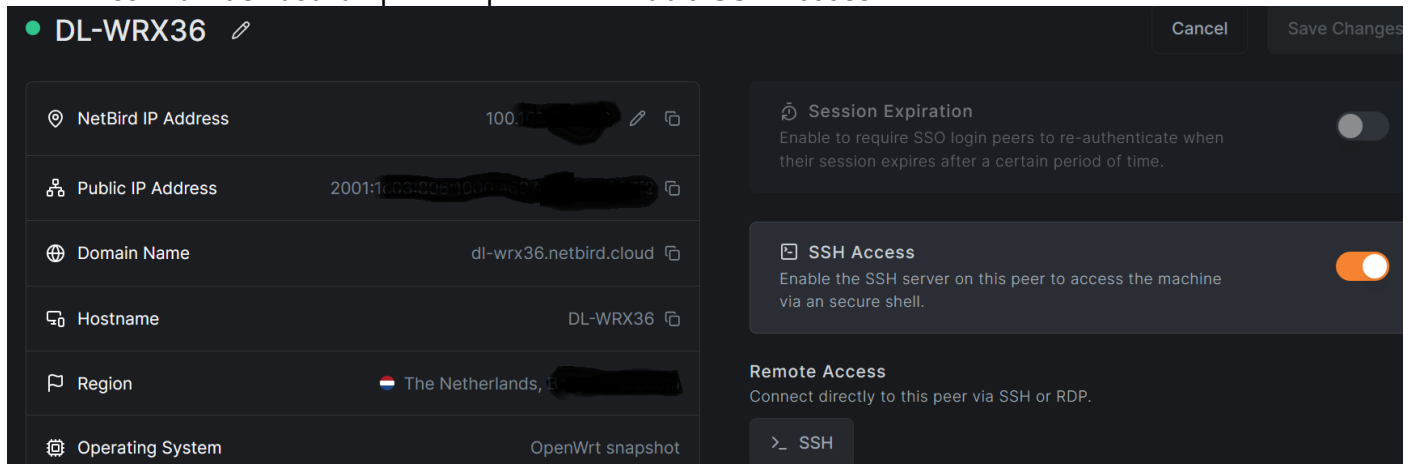
<https://forum.openwrt.org/t/netbird-support-discussion-thread/237831/1>

Online

See: <https://docs.netbird.io/help/troubleshooting-client>

Allow SSH access from Dashboard

In the **NetBird Dashboard** open the peer and *Enable SSH Access*:



On the router

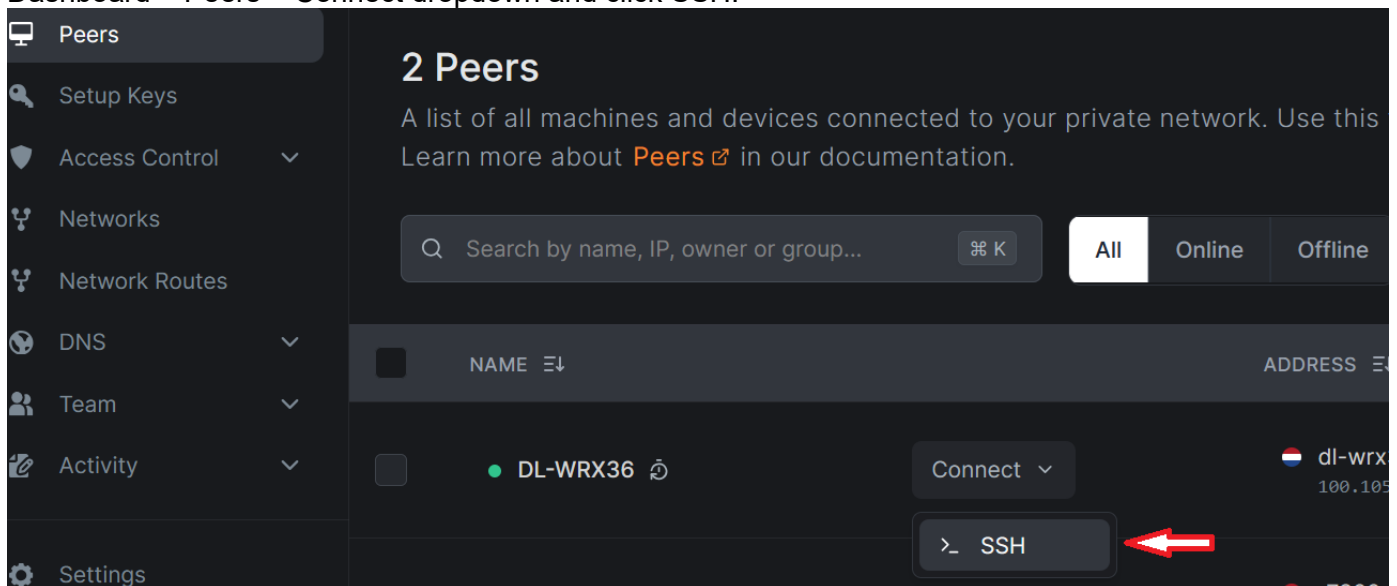
Make sure SSH is allowed (<https://github.com/netbirdio/netbird/issues/2632>):

netbird down

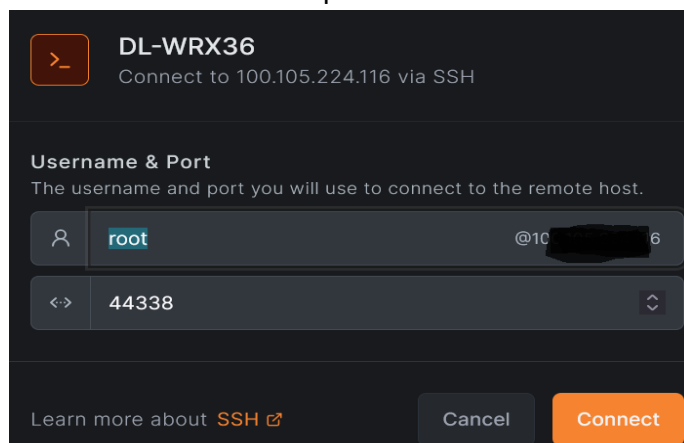
netbird up --allow-server-ssh

On your NetBird dashboard you should now be able to SSH into your router:

Dashboard > Peers > Connect dropdown and click SSH:



Connect with the default port 44338 to the in NetBird included SSH server:



Changes Starting with version 0.60

<https://docs.netbird.io/manage/peers/ssh>

<https://forum.netbird.io/t/upcoming-breaking-change-to-netbird-ssh/292>

You need to start NetBird with:

```
netbird up --allow-server-ssh --disable-ssh-auth --enable-ssh-root
```

Furthermore you need to add an Access policy for port 22 and port 22022 (when using the Standard Dropbear SSH) in the NetBird Dashboard:

Access Control > Policies > Add Policy:

Policy | Posture Checks | Name & Description

Protocol
Allow only specified network protocols. To change traffic direction and ports, select **TCP** or **UDP** protocol. TCP

Source **Destination**

All All

Ports
Allow network traffic and access only to specified ports. Select ports or port ranges between 1 and 65535.

22 22022

Enable Policy
Use this switch to enable or disable the policy. ☒

Start the SSH with user root and port 22.

If you have Dropbear running (which is the default in OpenWRT) instead of OpenSSH (sshd) then you need to use port 22022 and also make an Access Policy for port 22022

Username & Port
The username and port you will use to connect to the remote host.

root @ 152.5

22

Username & Port
The username and port you will use to connect to the remote host.

root @

22022

Create Routes

See: <https://docs.netbird.io/how-to/routing-traffic-to-private-networks>

Note for routing between your peers it is imperative that all involved subnets are unique!

My DL-WRX36 has subnet 192.168.9.0/24.

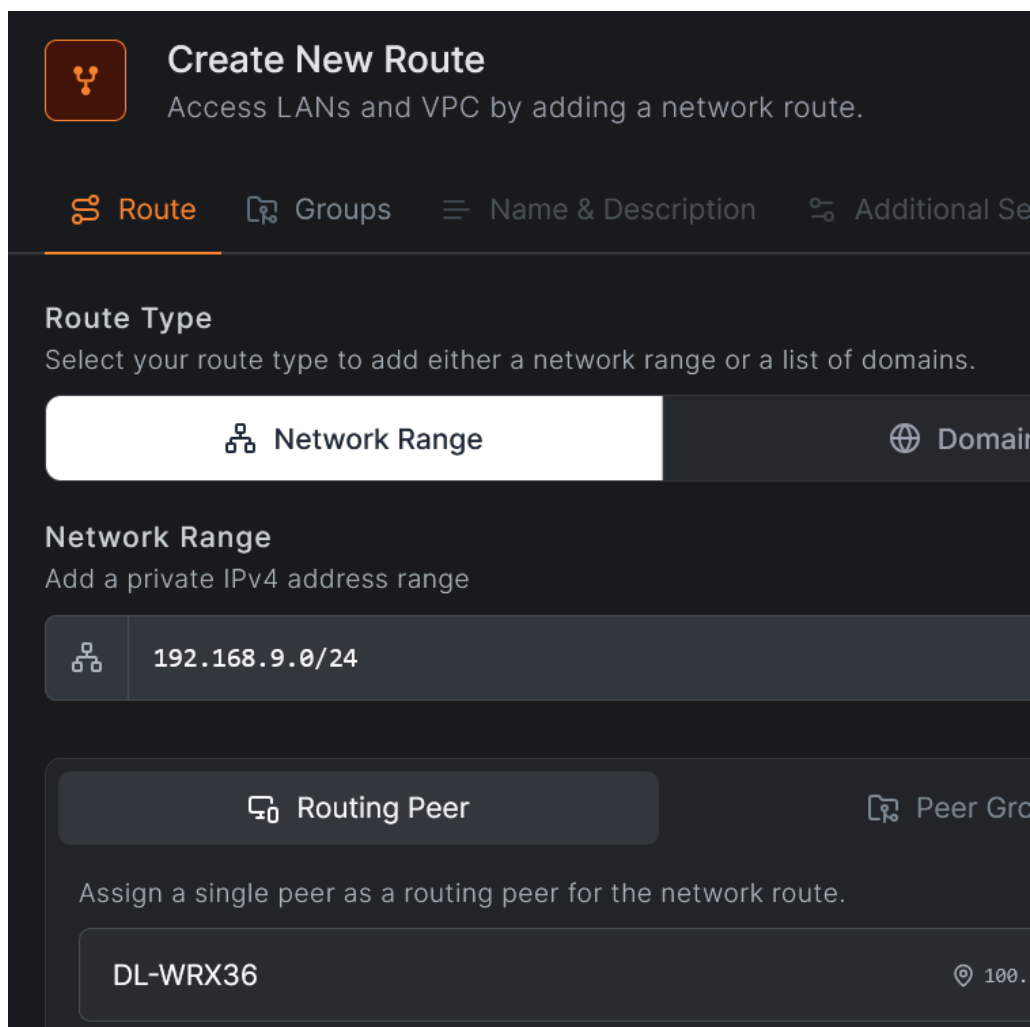
I will create a routing rule to create a route for this 192.168.9.0/24 subnet to my DL-WRX36 and push that route to all peers.

Those pushed routes are pushed to an alternate routing table on all peers, this table is usually called netbird.

Lets go:


NetBird Dashboard> Network Routes > Add Route

Add the network range to my DL-WRX36:







The screenshot shows the 'Create New Route' interface in the NetBird dashboard. At the top, there's a title 'Create New Route' with a subtitle 'Access LANs and VPC by adding a network route.' Below this is a navigation bar with tabs: 'Route' (selected), 'Groups', 'Name & Description', and 'Additional Settings'. The main section is titled 'Route Type' with the instruction 'Select your route type to add either a network range or a list of domains.' There are two buttons: 'Network Range' (selected) and 'Domain'. Under 'Network Range', there's a sub-instruction 'Add a private IPv4 address range' and a text input field containing '192.168.9.0/24'. Below this is a section for 'Routing Peer' with the instruction 'Assign a single peer as a routing peer for the network route.' There's a button 'Routing Peer' and a 'Peer Group' option. A dropdown menu is open showing 'DL-WRX36' as the selected peer, with a location pin icon and '100%' next to it.

Advertise this route to all my peers:




Create New Route

Access LANs and VPC by adding a network route.

 Route  **Groups**  Name & Description  Additional Settings

Distribution Groups


Advertise this route to peers that belong to the following groups

 Routing Peers ×

Access Control Groups (optional)


These groups allow you to limit access to this route. Simply use these groups as a destination when creating access policies.

Add or select group(s)...

[Learn more about Network Routes](#) 





BackContinue

Name and description:



Create New Route

Access LANs and VPC by adding a network route.

 Route  Groups  **Name & Description**  Additional Settings

Network Identifier

Add a unique network identifier that is assigned to each device.


DL-WRX36





Description (optional)



Write a short description to add more context to this route.



Route to DL-WRX36 192.168.9.0/24 subnet|



Additional settings:

 **Create New Route**
Access LANs and VPC by adding a network route.

 Route  Groups  Name & Description  **Additional Settings**

 **Enable Route**
Use this switch to enable or disable the route. 

 **Masquerade**
Allow access to your private networks without configuring routes on your local routers or other devices. 

Metric
A lower metric indicates higher priority.  9999 

You might need to restart NetBird on all peers

On my Oracle VPS I can now see the rules and the alternate routing table created by NetBird:

```
ubuntu@vps-egc:~$ ip rule show
0:    from all lookup local
105:  from all lookup main suppress_prefixlength 0
110:  not from all fwmark 0x1bd00 lookup netbird
32766: from all lookup main
32767: from all lookup default
ubuntu@vps-egc:~$
```

```
ubuntu@vps-egc:~$ ip route show table netbird
192.168.9.0/24 dev wt0
ubuntu@vps-egc:~$
```

wg show should also show the 192.168.9.0/24 added to the correct peer:

```
ubuntu@vps-egc:~$ sudo wg show
peer: < peer key >
  endpoint: [XXXXX:fedf]:33423
  allowed ips: 100.211.224.116/32, 192.168.9.0/24
  latest handshake: 42 seconds ago
  transfer: 1.41 KiB received, 2.00 KiB sent
  persistent keepalive: every 25 seconds

peer: < peer key >
  endpoint: [XXXX:1000:bea5:11ff:fe3e]:51555
  allowed ips: 192.168.5.0/24, 100.211.152.75/32
  latest handshake: 1 minute, 39 seconds ago
  transfer: 156 B received, 392 B sent
  persistent keepalive: every 25 seconds
```

So from my oracle VPS there now is a route to my DL-WRX36 subnet

Create Networks

See: <https://docs.netbird.io/how-to/networks>

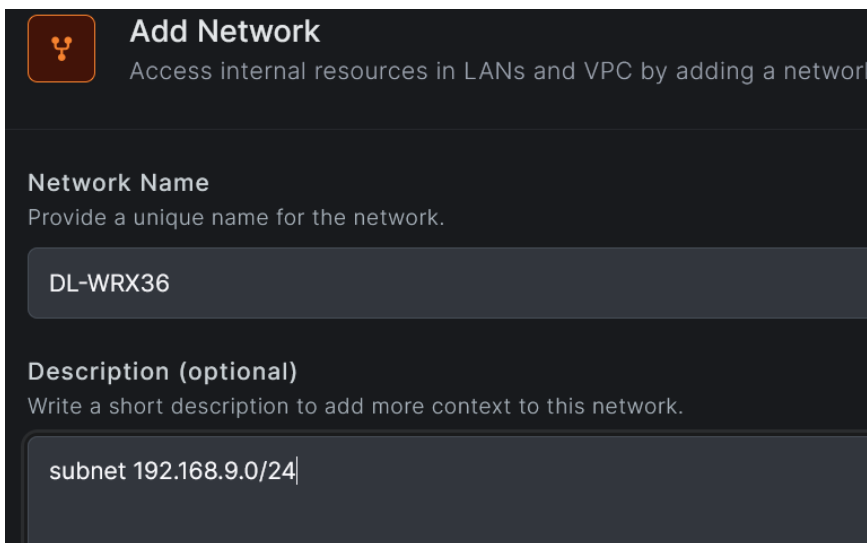
As I have just a few routers and a VPS to connect I use `Network Routes` which is simpler then using Networks.

Networks is new and has finer grained control but needs more work to setup and lacks support for exit nodes so that has still to be done with [Network Routes](#).

For OpenWRT the networks are often simple, so basically a Network has one routing peer which is the router or appliance in that network which holds the NetBird connection. This connects NetBird with the resources of the routing peer so basically the subnet or an IP address of a server which is running on this subnet. In this case I have my subnet as resource.

The access is controlled by Access Policies more on that later.

Start with creating a new Network e.g. the Network of my DL-WRX36 router which has subnet 192.168.9.0/24 and I want everyone to have access to this subnet:



Add Network
Access internal resources in LANs and VPC by adding a network

Network Name
Provide a unique name for the network.


DL-WRX36

Description (optional)
Write a short description to add more context to this network.

subnet 192.168.9.0/24

Proceed with making a **new Resource**:

Fill in name and address, under **Destination Groups** (these are the Access Control List > Groups) make a new group which later will hold our routing peer (added later to this the Access Control Destination group). Just enter the text for the new group e.g. DL-WRX36-group



Add Resource

Add new resource to "DL-WRX36"

Name



Provide a name for your resource

Description (optional)

Write a short description to add more context to this resource.


Address


Enter a single IP address, CIDR block or domain name


 

Destination Groups

Add this resource to groups and use them as destinations when creating policies





 DL-WRX36-group

Add this group by pressing **'Enter'**

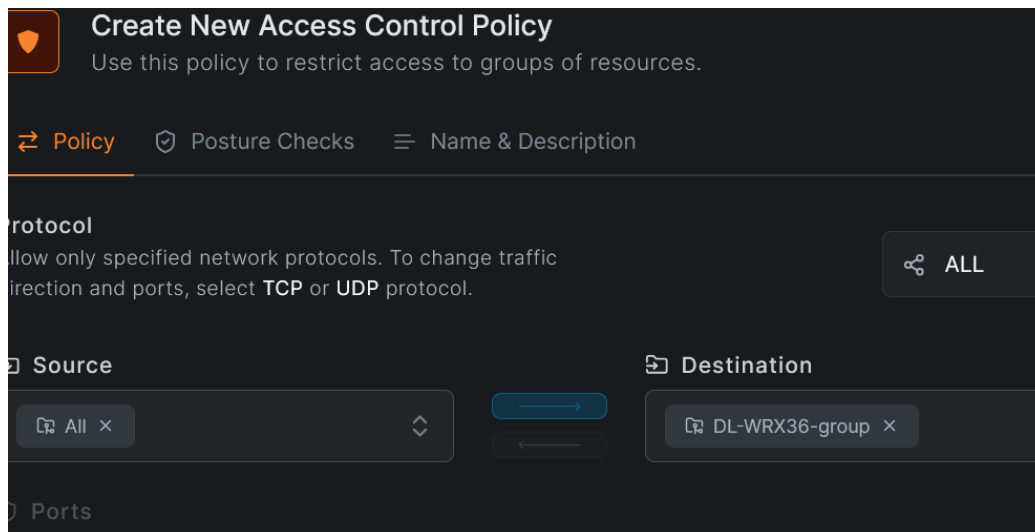
Proceed with making an Access Policy.

The destination is automatically your newly created Destination group (we will later add our peer to this Access group).

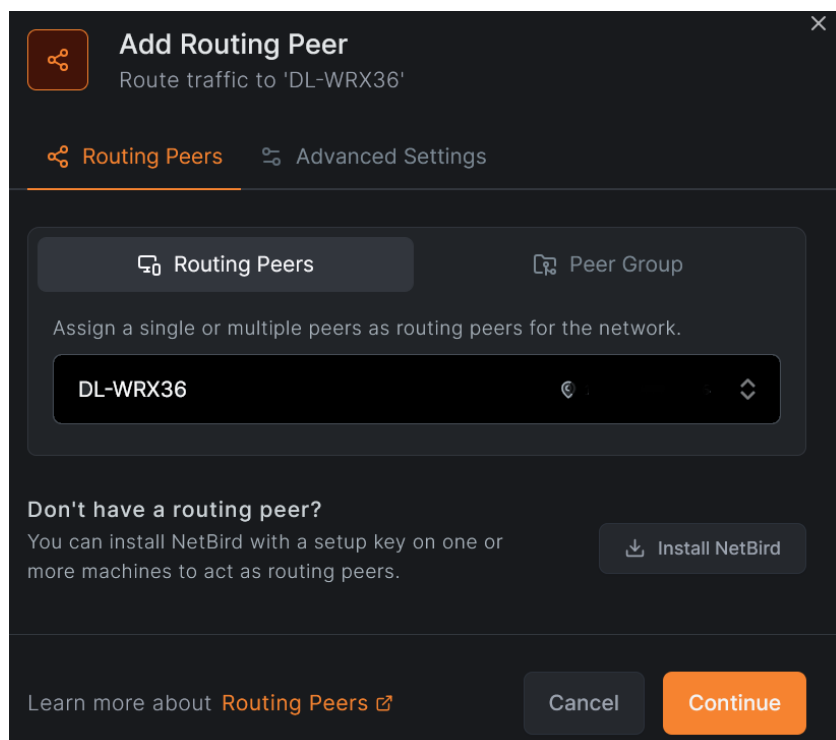
As Source I use `All` as everything in my network can have access but you can restrict it to your liking.

Note:

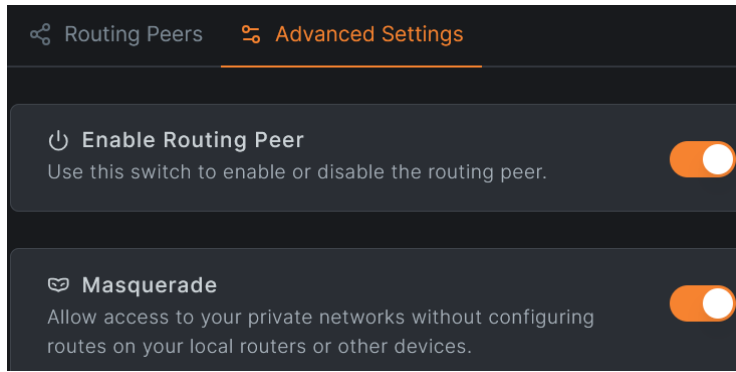
For every resource you have to make an Access Policy as the Destination Group you added is not automatically added to the `All` group



Proceed with adding a Routing Peer which is of course my DL-WRX36 peer:

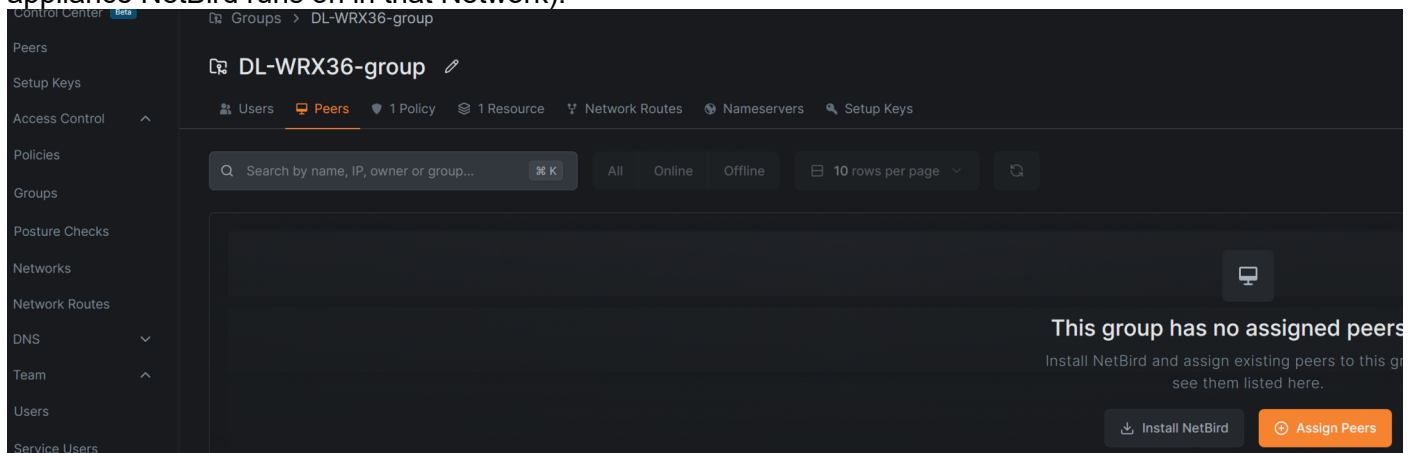


On Advanced settings enable Masquerading:



The last step is to add the DL-WRX36 peer to the Access Control Group DL-WRX36-group.

Goto Access Control > Groups and open the DL-WRX36 -group > Peers and assign you peer (the router or appliance NetBird runs on in that Network):



What I found confusing is that when making a resource you have to add the destination group. The Destination group is an Access List group to control access and it is only possible to control access by group and not by individual peer so you have to make a new Access control group and add your peer later to that group.

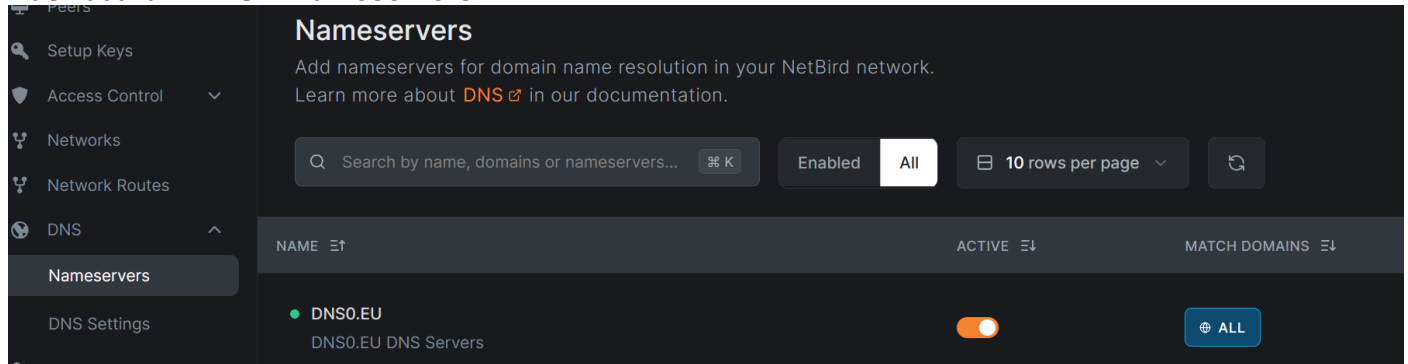
DNS settings

See: <https://docs.netbird.io/how-to/manage-dns-in-your-network>

NetBird by default runs its own DNS server on the peers, this is included in the NetBird executable and is used by default.

So make sure you set a Nameserver under DNS settings:

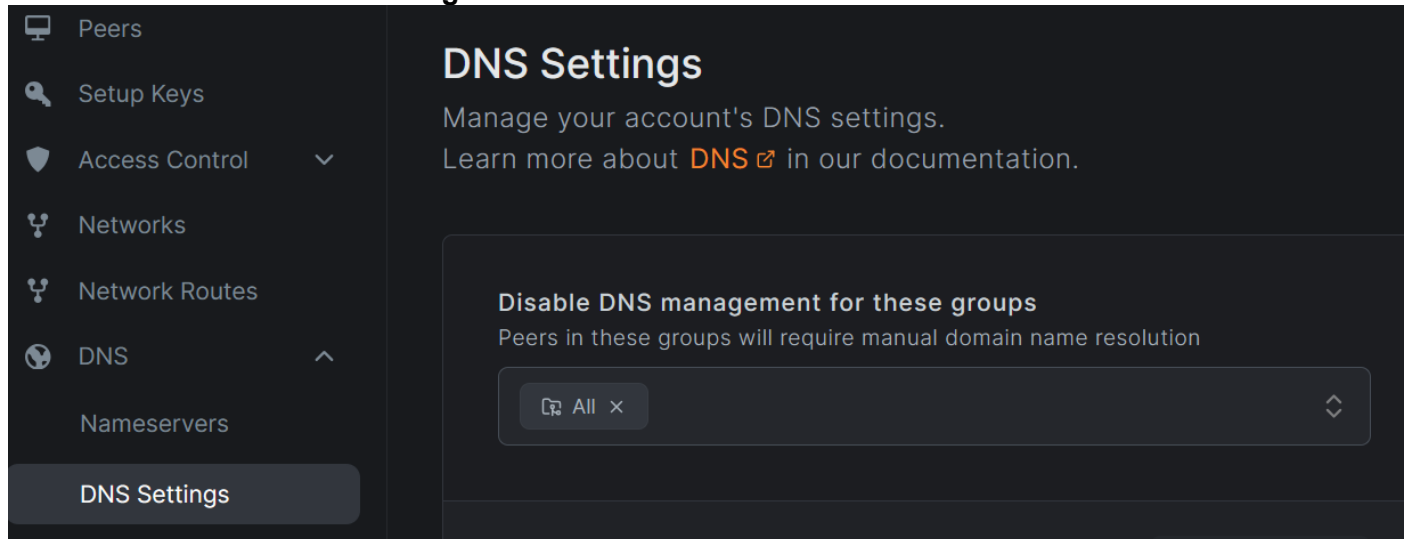
Dashboard > DNS > Nameservers:



I have chosen the DNS0.EU nameserver but you can choose others from a list or add your own.

If you do not want to use the NetBird DNS on your peers than you can disable it:

Dashboard > DNS > DNS Settings:



In this example I have disabled it for all peers, so all peers are using their own DNS settings and servers.

If you have one central DNS server you can set the NetBird name (e.g. mywrx36.netbird.cloud) of that server as Nameserver

Create Exit node

An exit node is a peer which acts as a VPN server other designated peers route all their traffic via the exit node.

On the exit node it is important that the firewall allows forwarding from **netbird** to **wan**, see paragraph about [firewall](#).

NetBird documentation: <https://netbird.io/knowledge-hub/netbird-network-routes>, scroll down to the bottom.

Log in in the NetBird dashboard

Peers > Click on the peer you want to be the exit node > On the overview page scroll to the bottom and click **Setup Exit node**

The screenshot shows the NetBird dashboard for a peer named DL-WRX36. The interface is dark-themed. At the top, there's a header with the peer name and two buttons: 'Cancel' and 'Save Changes'. Below this, the configuration is divided into two main sections. The left section contains a list of peer details: NetBird IP Address (100...), Public IP Address (2001...), Domain Name (dl-wrx36.netbird.cloud), Hostname (DL-WRX36), Region (The Netherlands), Operating System (OpenWrt snapshot), Registered on (8 October, 2025 at 6:53 PM (2 days ago)), Last seen (just now), and Agent Version (0.58.2). The right section contains settings for Session Expiration (disabled), SSH Access (enabled), Remote Access (SSH button), and Assigned Groups (All, Routing Peers). Below these sections, there are tabs for 'Network Routes', 'Accessible Peers', and 'Traffic Events'. The 'Network Routes' tab is active, showing a table with columns: NAME, NETWORK, DISTRIBUTION GROUPS, and ACTIVE. The table has one entry for DL-WRX36 with network 192.168.9.0/24, distribution group 'Routing Peers', and an active status. At the bottom right of the 'Network Routes' section, there are two buttons: 'Set Up Exit Node' and 'Add Route'.

DL-WRX36

NetBird IP Address 100... Public IP Address 2001... Domain Name dl-wrx36.netbird.cloud Hostname DL-WRX36 Region The Netherlands Operating System OpenWrt snapshot Registered on 8 October, 2025 at 6:53 PM (2 days ago) Last seen just now Agent Version 0.58.2

Session Expiration Enable to require SSO login peers to re-authenticate when their session expires after a certain period of time.

SSH Access Enable the SSH server on this peer to access the machine via an secure shell.

Remote Access Connect directly to this peer via SSH or RDP.

SSH

Assigned Groups Use groups to control what this peer can access.

All Routing Peers

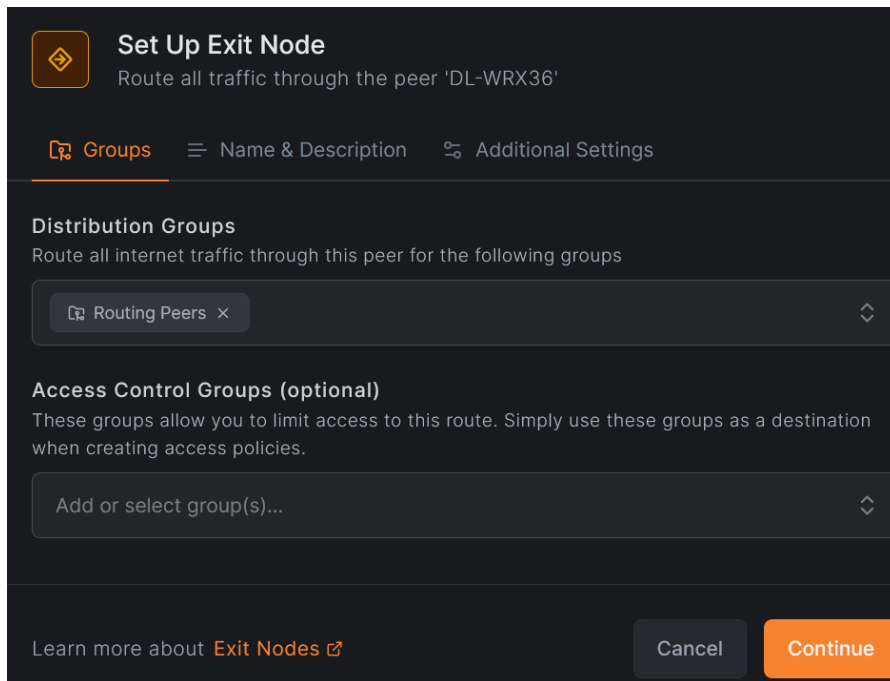
Network Routes Access other networks without installing NetBird on every resource.

Set Up Exit Node Add Route

NAME	NETWORK	DISTRIBUTION GROUPS	ACTIVE
DL-WRX36	192.168.9.0/24	Routing Peers	Active

Delete

Under **Groups** add the peers you want to use the exit node, I had created a group **Routing Peers** and I want all those peers to use this router as exit node



Set Up Exit Node
Route all traffic through the peer 'DL-WRX36'

Groups | Name & Description | Additional Settings

Distribution Groups
Route all internet traffic through this peer for the following groups

Routing Peers x

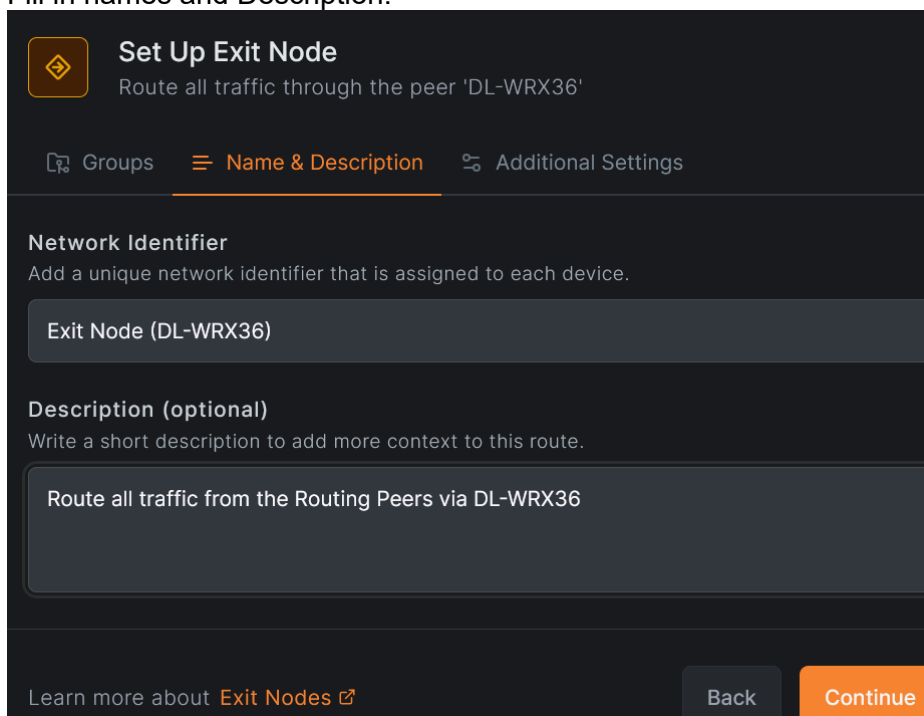
Access Control Groups (optional)
These groups allow you to limit access to this route. Simply use these groups as a destination when creating access policies.

Add or select group(s)...

Learn more about [Exit Nodes](#) | Cancel Continue

Continue

Fill in names and Description:



Set Up Exit Node
Route all traffic through the peer 'DL-WRX36'

Groups | **Name & Description** | Additional Settings

Network Identifier
Add a unique network identifier that is assigned to each device.

Exit Node (DL-WRX36)


Description (optional)
Write a short description to add more context to this route.

Route all traffic from the Routing Peers via DL-WRX36

Learn more about [Exit Nodes](#) | Back Continue

Continue


Enable Route and Auto Apply Route




Set Up Exit Node

Route all traffic through the peer 'DL-WRX36'

[Groups](#)
[Name & Description](#)
[Additional Settings](#)


Enable Route


Use this switch to enable or disable the route.


Auto Apply Route


Automatically apply this exit node to your distribution groups. This requires NetBird client v0.55.0 or higher.

Metric

A lower metric indicates higher priority routes.



9999



[Learn more about Exit Nodes](#)
[Back](#)
[+ Add Exit Node](#)

Add Exit Node






My DL-WRX36 is running Snapshot with NetBird 0.58 (you can see it on the overview page if you click on the Peer) so all routes are applied automatically.

My DL-WRX36 now has set a route to its own subnet (which is 192.168.9.0/24), pushed to all the Routing peers en an Exit node which pushes a default route to all the routing peers.

[Network Routes](#)
[Accessible Peers](#)
[Traffic Events](#)

2 Network Routes

Access other networks without installing NetBird on every resource.

NAME	NETWORK	DISTRIBUTION GROUPS	ACTIVE
 Exit Node (DL-WRX36)	 Exit Node	 Routing Peers	<div></div>
 DL-WRX36	192.168.9.0/24	 Routing Peers	<div></div>

You can check on one of the other routing peers e.g. my R7800-2 where you can see the pushed default route and the pushed route to reach the DL-WRX36:

```
root@R7800-2:~# ip route show table netbird
default dev wt0
192.168.9.0/24 dev wt0
```

Now all traffic from the R7800-2 (and all its LAN clients are routed) via NetBird interface, which is actually your WireGuard interface and `wg show` will show you how the routing is taking place via the exit node

Support

For support and questions see the NetBird support thread:

<https://forum.openwrt.org/t/netbird-support-discussion-thread/237831/8>

Known Problems

SSH-Access from Dashboard

Enabling [Lazy Connections](#) might stop SSH access from the Dashboard (Settings > Clients):

<https://netbird.io/knowledge-hub/lazy-connections>

<https://forum.openwrt.org/t/netbird-support-discussion-thread/237831/44?u=egc>

Policy Based Routing (PBR) together with NetBird

Policy based routing can be used to route traffic via different routes according to source and/or destination. It can be useful if you have set your peer (router) to use an exit node which means all traffic is routed via the specified exit node and you want to route some traffic via the WAN instead of the exit node.

You can setup this up [manually](#) but it is often easier to install the [PBR app](#).

To work together with the PBR app, add to the PBR config (**version 1.2.X** and higher):

/etc/config/pbr:

```
option uplink_ip_rules_priority '99'
```

or from command line:

```
uci set pbr.config.uplink_ip_rules_priority="99"
```

```
uci commit pbr
```

```
service pbr restart
```

This will make sure the PBR ip rules will come before the NetBird rules (>100).

For checking rules and routes see [Check and Troubleshoot](#) section.

If everything runs you should see something like this:

```
root@R7800-2:~# ip rule show
```

```
0:      from all lookup local
```

```
97:      from all sport 52199 lookup pbr_wan
```

```
97:      from all lookup main suppress_prefixlength 1
```

```
98:      from all fwmark 0x20000/0xff0000 lookup pbr_netbird1
```

```
99:      from all fwmark 0x10000/0xff0000 lookup pbr_wan
```

```
105:     from all lookup main suppress_prefixlength 0
```

```
110:     not from all fwmark 0x1bd00 lookup netbird
```

```
32766:   from all lookup main
```

```
32767:   from all lookup default
```

Note the still existing NetBird rules and the PBR rules **before** the Netbird rules, so these take precedence.

Netbird starts rather late so you might need to restart PBR (*service pbr restart*) after setting up or after a reboot

Install on Oracle VPS with Ubuntu (24.04)

```
sudo apt-get update
```

```
sudo apt install ca-certificates curl gnupg -y
```

```
curl -sSL https://pkgs.netbird.io/debian/public.key | sudo gpg --dearmor --output /usr/share/keyrings/netbird-archive-keyring.gpg
```

```
echo 'deb [signed-by=/usr/share/keyrings/netbird-archive-keyring.gpg] https://pkgs.netbird.io/debian stable main' | sudo tee /etc/apt/sources.list.d/netbird.list
```

```
sudo apt-get update
sudo apt-get install netbird
# only for the GUI
#sudo apt-get install netbird-ui
```

```
netbird up --setup-key <setup-key made on dashboard> --allow-server-ssh
```

Log on Ubuntu: `cat /var/log/netbird/client.log`

SSH access note that the user name is usually: *ubuntu*

For (SSH) Access add thes firewall rules

```
sudo iptables -I INPUT 3 -p udp --dport 3478 -j ACCEPT # NetBird TURN
sudo iptables -I INPUT 4 -p tcp --dport 44338 -j ACCEPT # SSH service port from NetBird

sudo iptables -I INPUT 5 -p udp --dport 51820 -j ACCEPT # NetBird WireGuard
#sudo iptables -t nat -I POSTROUTING -o wt0 -j MASQUERADE #To Masquerade traffic ?
#sudo iptables -t nat -I POSTROUTING -o ens3 -j MASQUERADE #To Masquerade traffic ?
```

Make persistent:
`sudo netfilter-persistent save`

vcn-XXX > Security > Default Security List for vcn-XXX > Security rules:

<input type="checkbox"/>	No	0.0.0.0/0	UDP	All	3478
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	44338
<input type="checkbox"/>	No	0.0.0.0/0	UDP	All	51820

Throughput improvements via transport layer offloading

Tuning two features may show improved throughput:

- `rx-udp-gro-forwarding`: Enables UDP Generic Receive Offload (GRO) forwarding, which aggregates incoming UDP packets to reduce CPU overhead on receive.
- `rx-gro-list`: If disabled (off), it prevents multiple flows from being aggregated simultaneously which simplifies flow handling and performance on some workloads.

From command line:

1. Install "ethtool":

```
apk update
apk add ethtool
```

2. Apply the changes:

Substitute "wan" below for your WAN interface.</WRAP>

```
ethtool -K wan rx-gro-list off
ethtool -K wan rx-udp-gro-forwarding on
```

3. Test the changes before and after before committing them permanently with something similar to the following commands.

You want to verify:

- Packet aggregation is working as measured by reduced packets/sec on the wire with GRO enabled (verify with tools like: `ethtool -S <interface> | grep udp` or `netstat -su`)

- CPU usage is reduced. Lower CPU usage on the receiver compared to same test with rx-udp-gro-forwarding turned off
- High throughput is achieved near line rate (e.g., 1 Gbps, 10Gbps, etc) without packetloss. You need iperf3 for proper measurement

Setup Oracle free OpenVPN cloud server

<https://www.youtube.com/watch?v=E-CLtExRzX8>

<https://mateo.cogeanu.com/2020/wireguard-vpn-pihole-on-free-oracle-cloud/>

References

NetBird support thread: <https://forum.openwrt.org/t/netbird-support-discussion-thread/237831/8>

Upgrade from 0.50 to 0.58: <https://github.com/netbirdio/netbird/issues/4322>

NetBird Releases

<https://github.com/netbirdio/netbird/releases>

Configuration

See: <https://docs.netbird.io/get-started/cli>

Flags are for using on the command line e.g. **netbird up --flag** if these are also config items then it will be stored in the config in `/root/.config/netbird/default.json` so there is no need to add the flag again. Most flags can also be set as Environment variable with syntax `NB_FLAG` and using underscores instead of hyphens e.g.: `export NB_FLAG=1` or set in the environment in the init script: `procd_append_param env NB_DISABLE_SSH_CONFIG="1"`

Disable SSH Authorization per user

flag (command line): `--disable-ssh-auth`
env param: `NB_DISABLE_SSH_AUTH=1`
config: `"DisableSSHAAuth": true,`

Enable SSH root user

flag: `--enable-ssh-root`
env param: `NB_ENABLE_SSH_ROOT`
config: `EnableSSHRoot": true,`

Disable Netbird DNS, this disables the writing to `resolv.conf`

config: `"DisabledDNS": true,`

Disable SSH integration with OpenSSH, this disables the writing of `/etc/ssh/ssh_config.d/99-netbird.conf`:

flag: `-` `--disable-ssh-config`
env param: `NB_DISABLE_SSH_CONFIG=1`
config: `"DisableSSHConfig": true`

Addendum 1

/root/.config/netbird/default.json:

```
{
  "PrivateKey": "kDJR2ykm0=",
  "PreSharedKey": "",
  "ManagementURL": {
    "Scheme": "https",
    "Opaque": "",
    "User": null,
    "Host": "api.netbird.io:443",
    "Path": "",
    "RawPath": "",
    "OmitHost": false,
    "ForceQuery": false,
    "RawQuery": "",
    "Fragment": "",
    "RawFragment": ""
  },
  "AdminURL": {
    "Scheme": "https",
    "Opaque": "",
    "User": null,
    "Host": "app.netbird.io:443",
    "Path": "",
    "RawPath": "",
    "OmitHost": false,
    "ForceQuery": false,
    "RawQuery": "",
    "Fragment": "",
    "RawFragment": ""
  },
  "WgIface": "wt0",
  "WgPort": 51820,
  "NetworkMonitor": null,
  "IFaceBlackList": [
    "wt0",
    "wt",
    "utun",
    "tun0",
    "zt",
    "ZeroTier",
    "wg",
    "ts",
    "Tailscale",
    "tailscale",
    "docker",
    "veth",
    "br-",
    "lo"
  ],
  "DisableIPv6Discovery": false,
  "RosenpassEnabled": false,
  "RosenpassPermissive": false,
  "ServerSSHAllowed": true,
  "EnableSSHRoot": true,
  "EnableSSHSFTP": null,
  "EnableSSHLocalPortForwarding": null,
  "EnableSSHRemotePortForwarding": null,
}
```

```
"DisableSSHAAuth": true,  
"SSHJWTCacheTTL": null,  
"DisableClientRoutes": false,  
"DisableServerRoutes": false,  
"DisableDNS": true,  
"DisableFirewall": false,  
"BlockLANAccess": false,  
"BlockInbound": false,  
"DisableNotifications": true,  
"DNSLabels": null,  
"SSHKey": "-----BEGIN PRIVATE KEY-----\nMC4CA\n-----END PRIVATE KEY-----\n",  
"NATExternalIPs": null,  
"CustomDNSAddress": "",  
"DisableAutoConnect": false,  
"DNSRouteInterval": 60000000000,  
"ClientCertPath": "",  
"ClientCertKeyPath": "",  
"LazyConnectionEnabled": false,  
"MTU": 1280  
}
```