Setting up OPNsense with MetalLB on Kubernetes

May 8, 2021 · 4 min

▼ Table of Contents

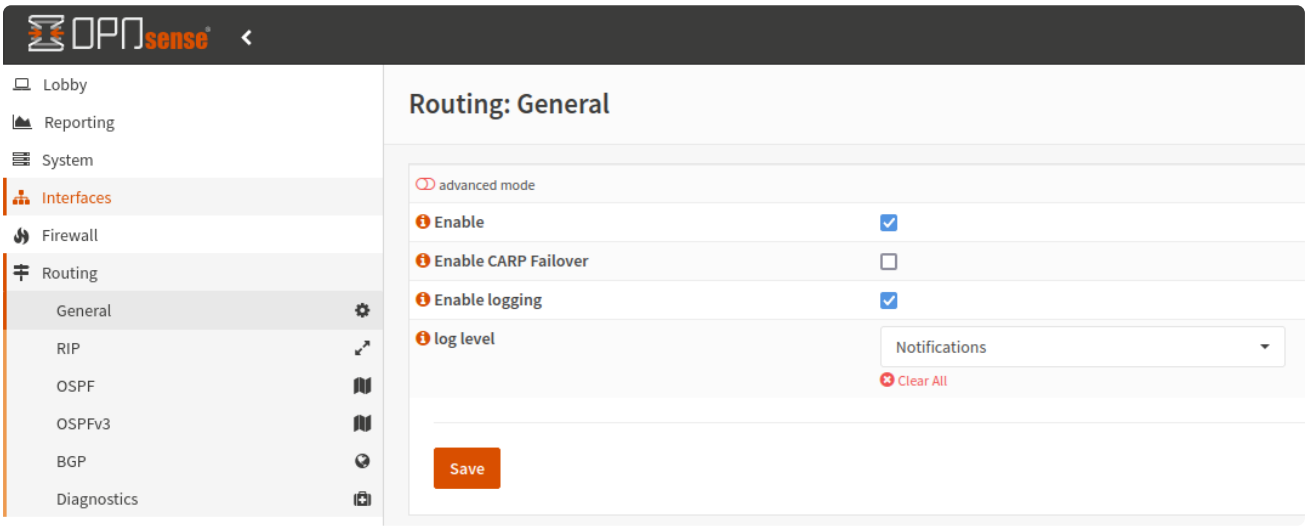
Kubernetes doesn't offer an implemention for Services of type Loadbalancer out of the box for baremetal clusters [1]. This is where MetalLB comes into the picture, it adds what kubernetes is missing here. MetalLB has two modes of operating, in layer 2 mode using ARP requests or with BGP. The BGP mode is needed for 'true' load balancing [2].

So this is why one might want to use MetalLB, but for proper BGP functionality you of course also need a BGP router to do the actual balancing. For this part of the setup I am using the free opensource router software OPNsense together with its FRR (Free Range Routing) package to provide the BGP functionality.

OPNsense Setup

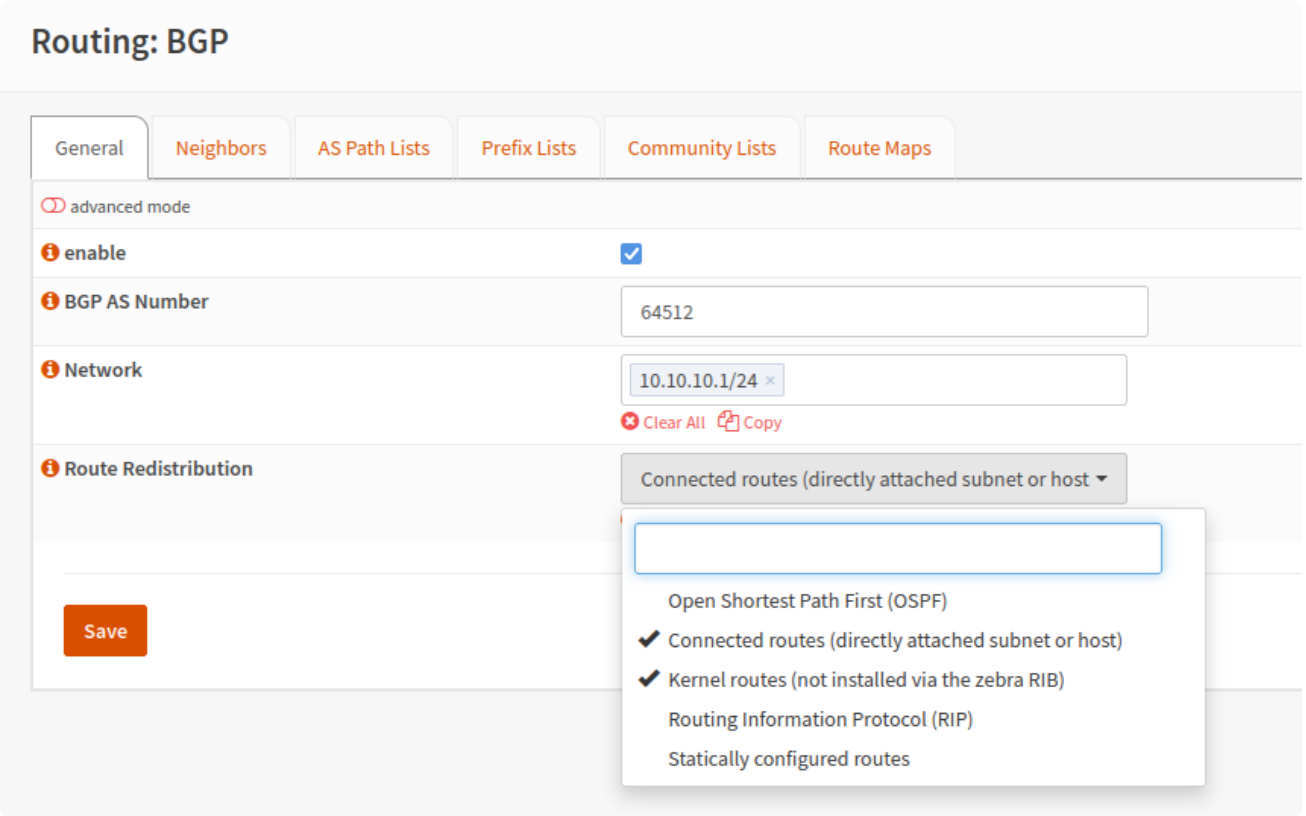
Assuming you already have OPNsense set up just add the FRR Package as described in the OPNsense docs.

After this is done you can go to `Routing > General` and enable the routing deamon.



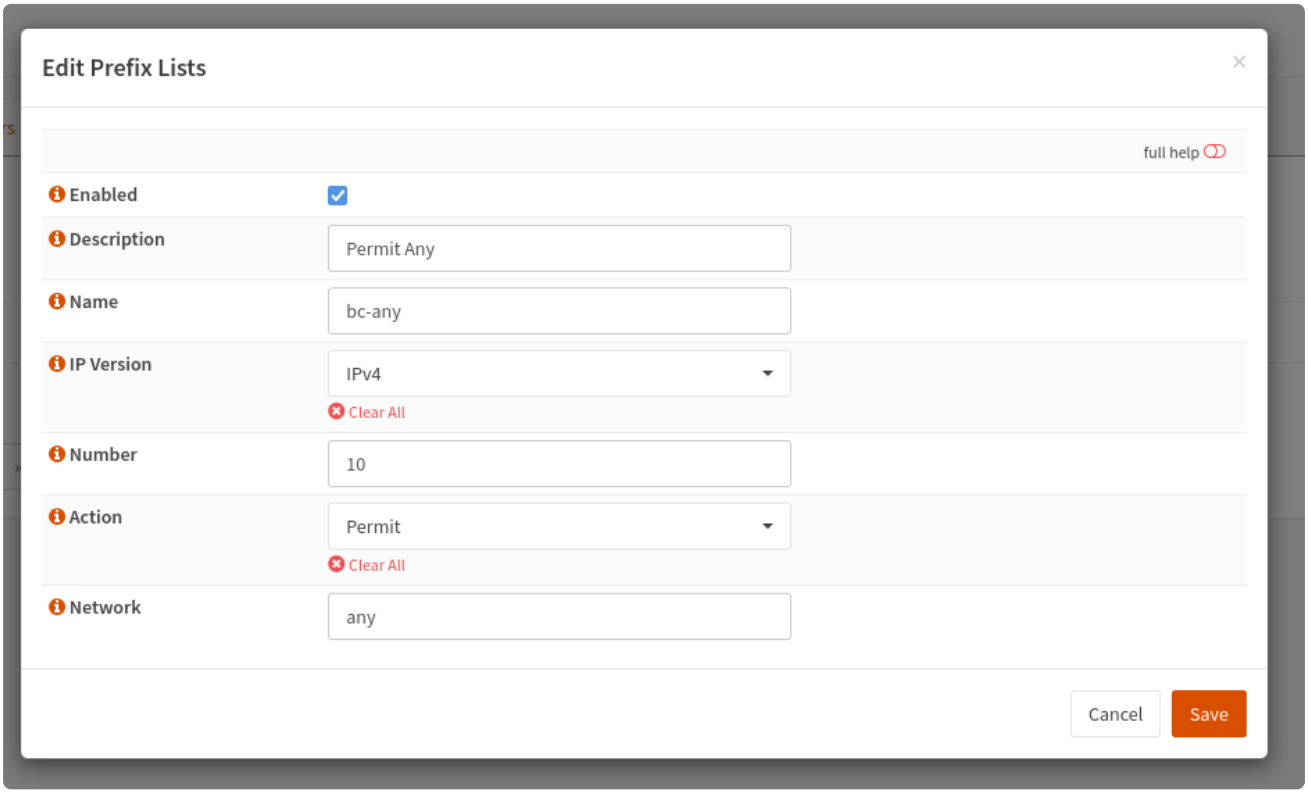
BGP General Settings

Now navigate to the BGP general settings (`Routing > BGP`) and enable this as well. You should also choose an AS number and enter the network mask here. I'd recommend choosing an ASN that falls into the private range as described by RFC6996, that is a number between 64512 and 65534. You also need to set the routing distribution here, I've set it connected and kernel routes.



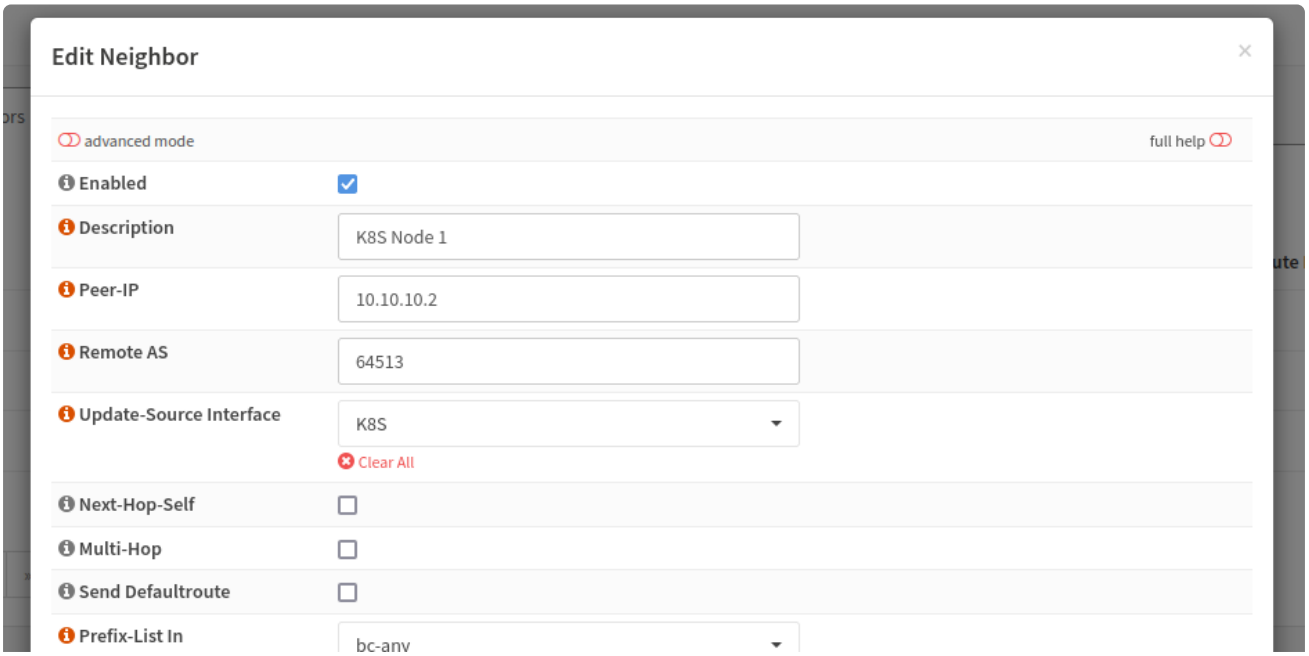
BGP Prefix List config

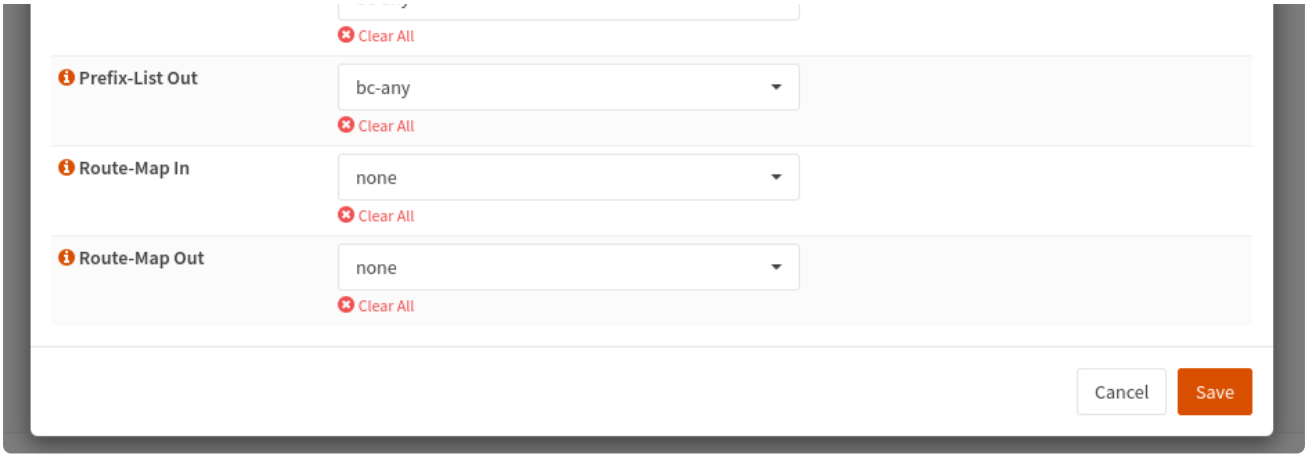
By default FRR won't permit any route announcements so we need to add a rule allowing it from the peers we'll define later. To do so head to the "Prefix List" tab and add a rule with action permit to network any.



BGP Neighbour config

Now that we have configured a prefix list we can add some neigbours that will use that rule. Add neigbours with as `Peer-IP` the IP of that kubernetes worker node, a Remote ASN that is different from OPNsense's and the previously made prefix list for in and output.





Repeat this for as many kubernetes nodes you have keeping the ASN the same and changing the Peer-IP. Also make sure all neighbours show up as enabled.

This is all we need to setup on the router's side.

Kubernetes Setup

For the MetalLB mainly just follow the official documentation.

For the ConfigMap you'll want something like this

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    peers:
    - peer-address: 10.10.10.1
      peer-asn: 64512
      my-asn: 64513
    address-pools:
    - name: default
      protocol: bgp
      addresses:
      - 10.10.10.11-10.10.10.250
```

- `peer-address` being the IP of the router
- `peer-asn` being the ASN of the router you configured
- `my-asn` the ASN you configured for the neighbours.

- `addresses` being the range(s) MetalLB is allowed to use to allocate IPs from.

Finishing Up

Now that everything should be configured we can test this out by making a test deployment like so:

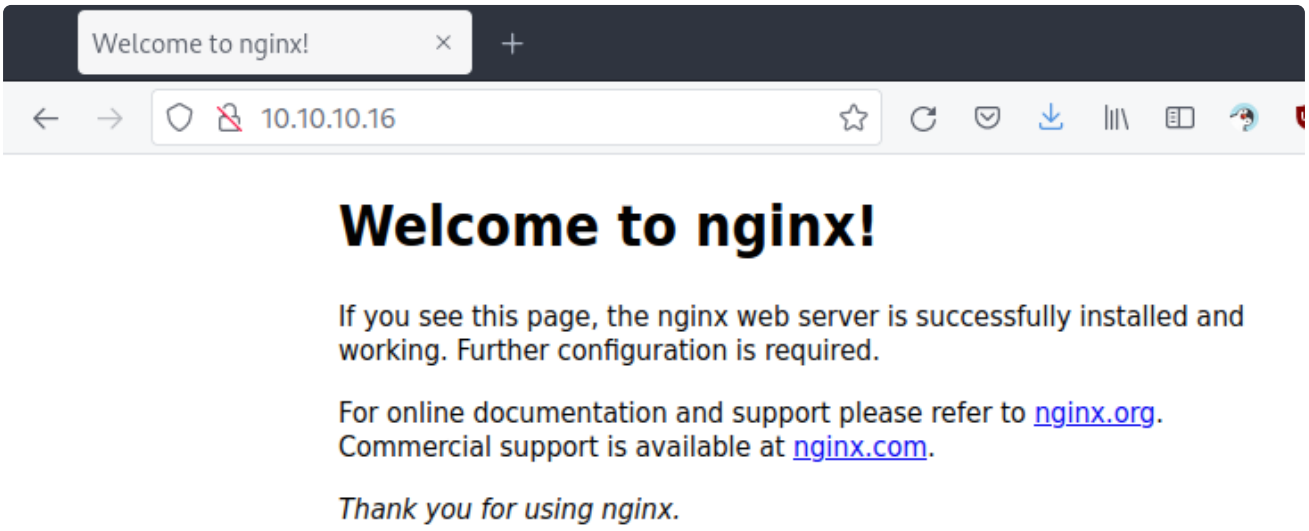
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-nginx
spec:
  selector:
    matchLabels:
      run: test-nginx
  replicas: 3
  template:
    metadata:
      labels:
        run: test-nginx
    spec:
      containers:
      - name: test-nginx
        image: nginx
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: test-nginx
  labels:
    run: test-nginx
spec:
  type: LoadBalancer
  ports:
  - port: 80
    protocol: TCP
  selector:
    run: test-nginx
```

After a few moments you should be able to get the provisioned IP like so:

```
; kubectl describe service test-nginx | grep "LoadBalancer"
```

```
Type:                     LoadBalancer
LoadBalancer Ingress:     10.10.10.16
```

And when navigating to this IP you should see the familiar nginx welcome screen



Congratulations! You now have OPNsense running FRR connected to MetalLB running in Kubernetes to dynamically provision IP addreses and loadbalance services 🎉.

References

Some resources I've used to compile this post which may be useful to you

- https://metallb.universe.tf/
- https://blog.matrixpost.net/set-up-dynamic-routing-with-frr-free-range-routing-in-pfsense-openbgpd-now-depricated/
- https://www.danmanners.com/posts/2019-02-pfsense-bgp-kubernetes/
- https://docs.opnsense.org/manual/dynamic_routing.html

1. These are usually only available in cloud environments like GCP, AWS, Azure, etc. ↩
2. For more information about the different modes please look at the MetalLB docs. ↩

Kubernetes OPNsense Networking