

# Open Infrastructure - Task #8447

## Deploy POC IPv6 cluster on DCL (v202009)

09/14/2020 08:23 PM - Nico Schottelius

<b>Status:</b>	Rejected	<b>Start date:</b>	09/14/2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Nico Schottelius	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>PM Check date:</b>			

### Description

#### Setup

- Create 3 Alpine nodes upgraded to edge, set hostnames
  - Resize to at least 2 cores per node
- Reserve networks
  - 2a0a:e5c0:2:12::/64 = node1
  - 2a0a:e5c0:2:13::/64 = services
  - 2a0a:e5c0:2:14::/64 = node3
- Configure routers to accept BGP session (done: in edist)
- Deploy kubernetes on first node
- Deploy kube-router: fail, not IPv6 ready
- Deploy calico: fail
- Deploy cilium: testing
- Create BGP peering
- Verify BGP peering
- Setup access to CEPH for persistent storage

#### OS commands

```
echo node2 > /etc/hostname

cat > /etc/resolv.conf << EOF
nameserver 2a0a:e5c0:2:12:0:f0ff:fea9:c451
nameserver 2a0a:e5c0:2:12:0:f0ff:fea9:c45d
search k8s.ungleich.ch
EOF
chattr +i /etc/resolv.conf

cat > /etc/apk/repositories << EOF
https://mirror.ungleich.ch/mirror/packages/alpine/edge/main
https://mirror.ungleich.ch/mirror/packages/alpine/edge/community
https://mirror.ungleich.ch/mirror/packages/alpine/edge/testing
EOF

apk upgrade
apk add kubeadm kubelet docker
rc-update add kubelet default
rc-update add docker default

echo 'net.ipv6.conf.default.forwarding=1' > /etc/sysctl.d/k8s.conf

kubeadm.conf:

localAPIEndpoint:
  advertiseAddress: 2a0a:e5c0:2:2:0:84ff:fe41:f263
---
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
networking:
  serviceSubnet: 2a0a:e5c0:2:13::/110
```

```
podSubnet: 2a0a:e5c0:2:12::/64
```

## Init cluster:

```
kubeadm init --config kubeadm.conf  
  
useradd -m k8s -s /bin/bash  
mkdir ~k8s/.kube  
cp /etc/kubernetes/admin.conf ~k8s/.kube/config  
chown -R k8s ~k8s
```

## Take aways

- docker sets ipv4 forwarding, but not ipv6 (needs manual sysctl entry)
- Reachability by name (node1) w/o fqdn seems to be important

## Current results

- kube-router does not work out-of-the-box "too many colons"
- calico does not work out of the box "calico-kube-controllers stays in pending" / no network provided

## History

### #1 - 09/14/2020 08:27 PM - Nico Schottelius

- Project changed from 45 to Open Infrastructure
- Description updated

### #2 - 09/14/2020 08:30 PM - Nico Schottelius

- Description updated

### #3 - 09/14/2020 08:54 PM - Nico Schottelius

- Description updated

try1:

```
node1:~# kubeadm init --config kubeadm.conf  
W0914 19:04:31.398219    3022 kubelet.go:200] cannot automatically set CgroupDriver when starting the Kubelet:  
  cannot execute 'docker info -f {{.CgroupDriver}}': executable file not found in $PATH  
W0914 19:04:31.439357    3022 version.go:102] could not fetch a Kubernetes version from the internet: unable to  
  get URL "https://dl.k8s.io/release/stable-1.txt": Get "https://dl.k8s.io/release/stable-1.txt": dial tcp 34.  
  107.204.206:443: connect: network is unreachable  
W0914 19:04:31.439492    3022 version.go:103] falling back to the local client version: v1.19.1  
W0914 19:04:31.439653    3022 configset.go:348] WARNING: kubeadm cannot validate component configs for API groups  
  [kubelet.config.k8s.io kubeproxy.config.k8s.io]  
[init] Using Kubernetes version: v1.19.1  
[preflight] Running pre-flight checks  
[preflight] WARNING: Couldn't create the interface used for talking to the container runtime: docker is required  
  for container runtime: exec: "docker": executable file not found in $PATH  
    [WARNING Hostname]: hostname "node1" could not be reached  
    [WARNING Hostname]: hostname "node1": lookup node1 on [2a0a:e5c0:2:a::a]:53: no such host  
    [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'rc-update add kubelet default'  
error execution phase preflight: [preflight] Some fatal errors occurred:  
  [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2  
  [ERROR FileContent--proc-sys-net-bridge-bridge-nf-call-iptables]: /proc/sys/net/bridge/bridge-nf-call-iptables  
  does not exist  
  [ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1  
  [ERROR FileContent--proc-sys-net-bridge-bridge-nf-call-ip6tables]: /proc/sys/net/bridge/bridge-nf-call-ip6  
  tables does not exist  
  [ERROR FileContent--proc-sys-net-ipv6-conf-default-forwarding]: /proc/sys/net/ipv6/conf/default/forwarding  
  contents are not set to 1  
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`  
  
To see the stack trace of this error execute with --v=5 or higher  
node1:~#
```

### #4 - 09/14/2020 08:56 PM - Nico Schottelius

- Description updated

- Description updated

try 3:

```
node1:~# echo "search k8s.ungleich.ch" >> /etc/resolv.conf
node1:~# kubeadm init --config kubeadm.conf
W0914 19:08:05.185304    2958 kubelet.go:200] cannot automatically set CgroupDriver when starting the Kubelet:
  cannot execute 'docker info -f {{.CgroupDriver}}': executable file not found in $PATH
W0914 19:08:05.186504    2958 version.go:102] could not fetch a Kubernetes version from the internet: unable to get URL "https://dl.k8s.io/release/stable-1.txt": Get "https://dl.k8s.io/release/stable-1.txt": dial tcp 34.107.204.206:443: connect: network is unreachable
W0914 19:08:05.186520    2958 version.go:103] falling back to the local client version: v1.19.1
W0914 19:08:05.186605    2958 configset.go:348] WARNING: kubeadm cannot validate component configs for API groups [kubelet.config.k8s.io kubeproxy.config.k8s.io]
[init] Using Kubernetes version: v1.19.1
[preflight] Running pre-flight checks
[preflight] WARNING: Couldn't create the interface used for talking to the container runtime: docker is required for container runtime: exec: "docker": executable file not found in $PATH
  [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'rc-update add kubelet default'
error execution phase preflight: [preflight] Some fatal errors occurred:
  [ERROR FileContent--proc-sys-net-bridge-bridge-nf-call-iptables]: /proc/sys/net/bridge/bridge-nf-call-iptables does not exist
  [ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1
  [ERROR FileContent--proc-sys-net-bridge-bridge-nf-call-ip6tables]: /proc/sys/net/bridge/bridge-nf-call-ip6tables does not exist
  [ERROR FileContent--proc-sys-net-ipv6-conf-default-forwarding]: /proc/sys/net/ipv6/conf/default/forwarding contents are not set to 1
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...
`
```

To see the stack trace of this error execute with --v=5 or higher

node1:~#

try5:

```
node1:~# kubeadm init --config kubeadm.conf
W0914 19:16:53.747589    3306 version.go:102] could not fetch a Kubernetes version from the internet: unable to get URL "https://dl.k8s.io/release/stable-1.txt": Get "https://dl.k8s.io/release/stable-1.txt": dial tcp 34.107.204.206:443: connect: network is unreachable
W0914 19:16:53.747631    3306 version.go:103] falling back to the local client version: v1.19.1
W0914 19:16:53.747768    3306 configset.go:348] WARNING: kubeadm cannot validate component configs for API groups [kubelet.config.k8s.io kubeproxy.config.k8s.io]
[init] Using Kubernetes version: v1.19.1
[preflight] Running pre-flight checks
  [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Please follow the guide at https://kubernetes.io/docs/setup/cri/
  [WARNING Hostname]: hostname "node1" could not be reached
  [WARNING Hostname]: hostname "node1": lookup node1 on [2a0a:e5c0:2:a::a]:53: no such host
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local node1] and IPs [2a0a:e5c0:2:13::1 2a0a:e5c0:2:2:0:84ff:fe41:f263]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [localhost node1] and IPs [2a0a:e5c0:2:2:0:84ff:fe41:f263 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [localhost node1] and IPs [2a0a:e5c0:2:2:0:84ff:fe41:f263 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
```

```
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[kubelet-check] Initial timeout of 40s passed.
^C
```

## #6 - 09/14/2020 09:11 PM - Nico Schottelius

- Description updated

## #7 - 09/14/2020 09:23 PM - Nico Schottelius

- Description updated

try 6:

```
...
[kubelet-start] Starting the kubelet
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[kubelet-check] Initial timeout of 40s passed.
```

Unfortunately, an error has occurred:  
timed out waiting for the condition

This error is likely caused by:  
- The kubelet is not running  
- The kubelet is unhealthy due to a misconfiguration of the node in some way (required cgroups disable d)

If you are on a systemd-powered system, you can try to troubleshoot the error with the following commands:  
- 'systemctl status kubelet'  
- 'journalctl -xeu kubelet'

Additionally, a control plane component may have crashed or exited when started by the container runtime.  
To troubleshoot, list all containers using your preferred container runtimes CLI.

Here is one example how you may list all Kubernetes containers running in docker:  
- 'docker ps -a | grep kube | grep -v pause'  
Once you have found the failing container, you can inspect its logs with:  
- 'docker logs CONTAINERID'

```
error execution phase wait-control-plane: couldn't initialize a Kubernetes cluster
To see the stack trace of this error execute with --v=5 or higher
node1:~#
```

checks:

- etcd seems to be ok
- scheduler tries to connect do :6443 and fails

network is too big:

```
node1:~# docker logs de26972b1722
Flag --insecure-port has been deprecated, This flag will be removed in a future version.
I0914 19:25:32.735390      1 server.go:625] external host was not specified, using 2a0a:e5c0:2:2:0:84ff:fe41:f263
Error: specified --service-cluster-ip-range is too large; for 128-bit addresses, the mask must be >= 108
node1:~#
```

Old:

```

node1:~# cat kubeadm.conf
apiVersion: kubeadm.k8s.io/v1beta2
kind: InitConfiguration
localAPIEndpoint:
  advertiseAddress: 2a0a:e5c0:2:2:0:84ff:fe41:f263
---
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
networking:
  serviceSubnet: 2a0a:e5c0:2:13::/64
  podSubnet: 2a0a:e5c0:2:12::/64
node1:~#

```

new:

```

node1:~# cat kubeadm.conf
apiVersion: kubeadm.k8s.io/v1beta2
kind: InitConfiguration
localAPIEndpoint:
  advertiseAddress: 2a0a:e5c0:2:2:0:84ff:fe41:f263
---
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
networking:
  serviceSubnet: 2a0a:e5c0:2:13::/110
  podSubnet: 2a0a:e5c0:2:12::/64
node1:~#

```

Success afterwards!

```

node1:~$ kubectl get pods -n kube-system
NAME          READY   STATUS    RESTARTS   AGE
coredns-f9fd979d6-24t7g   0/1     Pending   0          2m22s
coredns-f9fd979d6-jt6hw   0/1     Pending   0          2m22s
etcd-node1      1/1     Running   0          2m39s
kube-apiserver-node1    1/1     Running   0          2m39s
kube-controller-manager-node1  1/1     Running   0          2m39s
kube-proxy-6lpbs        1/1     Running   0          2m22s
kube-scheduler-node1    1/1     Running   0          2m39s
node1:~$ 

```

## #8 - 09/14/2020 09:28 PM - Nico Schottelius

- Description updated

- <https://github.com/cloudnative-labs/kube-router/blob/master/docs/generic.md>
  - kube-apiserver and kubelet must be run with --allow-privileged=true

```

node1:~# cat /etc/conf.d/kubelet
command_args="--cni-bin-dir=/usr/libexec/cni --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml"
node1:~# vi /etc/conf.d/kubelet
node1:~# cat /etc/conf.d/kubelet
command_args="--cni-bin-dir=/usr/libexec/cni --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml --allow-privileged=true"
node1:~# /etc/init.d/kubelet restart
 * Caching service dependencies ...
               [ ok ]
 * Stopping kubelet ...
               [ ok ]
 * Starting kubelet ...
               [ ok ]
node1:~#

```

kube-router/peering:

- <https://github.com/cloudnative-labs/kube-router/blob/master/docs/bgp.md>

Example:

```
--peer-router-ips="192.168.1.99,192.168.1.100"
--peer-router-asns=65000,65000
```

github fail:

```
node1:~$ kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-kuberouter.yaml
Unable to connect to the server: dial tcp 151.101.128.133:443: connect: network is unreachable
node1:~$
```

Untaint, tries to deploy:

```
node1:~$ kubectl taint nodes node1 node-role.kubernetes.io/master-
node1:~$ kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-kuberouter.yaml
```

```
^Cnode1:~$ kubectl get pods -n kube-system
NAME                      READY   STATUS        RESTARTS   AGE
coredns-f9fd979d6-jngp6    0/1     ContainerCreating   0          2m30s
coredns-f9fd979d6-kqjcl    0/1     ContainerCreating   0          2m30s
etcd-node1                 1/1     Running       0          2m45s
kube-apiserver-node1      1/1     Running       0          2m45s
kube-controller-manager-node1 1/1     Running       0          2m45s
kube-proxy-ft7t7           1/1     Running       0          2m30s
kube-router-cxfc2          0/1     CrashLoopBackOff  3          85s
kube-scheduler-node1       1/1     Running       0          2m45s
node1:~$
```

Config is created:

```
node1:~# cat /etc/cni/net.d/10-kuberouter.conflist
{
  "cniVersion": "0.3.0",
  "name": "mynet",
  "plugins": [
    {
      "name": "kubernetes",
      "type": "bridge",
      "bridge": "kube-bridge",
      "isDefaultGateway": true,
      "ipam": {
        "type": "host-local"
      }
    }
  ]
}
```

However kubelet does not have the --allow-privileged=true flag at all. (does not support this parameter)

kube-router crashes without a router-id:

```
node1:~# docker logs e457ef7a933d
I0914 20:11:45.995122      1 kube-router.go:231] Running /usr/local/bin/kube-router version v1.1.0-rcl-dirty,
built on 2020-09-07T21:42:44+0000, go1.13.13
Failed to run kube-router: Failed to create network routing controller: Router-id must be specified in ipv6 operation
node1:~#
```

Patching it in:

```
node1:~$ diff -u kubeadm-kuberouter.yaml.orig kubeadm-kuberouter.yaml
--- kubeadm-kuberouter.yaml.orig
+++ kubeadm-kuberouter.yaml
@@ -55,6 +55,8 @@
      - --run-firewall=true
      - --run-service-proxy=false
      - --bgp-graceful-restart=true
+
+     - --router-id
+     - ${NODE_NAME}
+
 env:
   - name: NODE_NAME
     valueFrom:
```

node1:~\$

Results in a new error:

```
node1:~# docker logs e29309159a4a
I0914 20:28:03.510920      1 kube-router.go:231] Running /usr/local/bin/kube-router version v1.1.0-rcl-dirty,
```

```

built on 2020-09-07T21:42:44+0000, go1.13.13
I0914 20:28:03.638165      1 network_routes_controller.go:1075] Could not find annotation `kube-router.io/bgp
-local-addresses` on node object so BGP will listen on node IP: 2a0a:e5c0:2:2:0:84ff:fe41:f263 address.
I0914 20:28:03.784909      1 network_policy_controller.go:148] Starting network policy controller
I0914 20:28:03.799619      1 network_policy_controller.go:156] Starting network policy controller full sync g
oroutine
E0914 20:28:03.950130      1 network_routes_controller.go:157] Failed to enable required policy based routing
: Failed to add ip rule due to: exit status 2
I0914 20:28:03.953857      1 network_routes_controller.go:228] Starting network route controller
time="2020-09-14T20:28:03Z" level=warning msg="listen failed" Error="listen tcp: address 2a0a:e5c0:2:2:0:84ff:
fe41:f263:50051: too many colons in address" Key="2a0a:e5c0:2:2:0:84ff:fe41:f263:50051" Topic=grpc
time="2020-09-14T20:28:03Z" level=fatal msg="failed to listen grpc port: listen tcp: address 2a0a:e5c0:2:2:0:8
4ff:fe41:f263:50051: too many colons in address"
node1:~#

```

## #9 - 09/14/2020 09:34 PM - Nico Schottelius

- Description updated

Switching to DNS64 servers:

```

node1:~# cat > /etc/resolv.conf
nameserver 2a0a:e5c0:2:12:0:f0ff:fea9:c451
nameserver 2a0a:e5c0:2:12:0:f0ff:fea9:c45d
search k8s.ungleich.ch
node1:~# chattr +i /etc/resolv.conf
node1:~#

```

## #10 - 09/14/2020 09:35 PM - Nico Schottelius

- Description updated

```

node1:~$ kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubea
dm-kuberouter.yaml
configmap/kube-router-cfg created
daemonset.apps/kube-router created
serviceaccount/kube-router created
Warning: rbac.authorization.k8s.io/v1beta1 ClusterRole is deprecated in v1.17+, unavailable in v1.22+; use rba
c.authorization.k8s.io/v1 ClusterRole
clusterrole.rbac.authorization.k8s.io/kube-router created
Warning: rbac.authorization.k8s.io/v1beta1 ClusterRoleBinding is deprecated in v1.17+, unavailable in v1.22+;
use rbac.authorization.k8s.io/v1 ClusterRoleBinding
clusterrolebinding.rbac.authorization.k8s.io/kube-router created
node1:~$ kubectl get pods -n kube-system
NAME                  READY   STATUS    RESTARTS   AGE
coredns-f9fd979d6-24t7g 0/1     Pending   0          14m
coredns-f9fd979d6-jt6hw 0/1     Pending   0          14m
etcd-node1            1/1     Running   0          14m
kube-apiserver-node1  1/1     Running   0          14m
kube-controller-manager-node1 1/1     Running   0          14m
kube-proxy-6lpbs       1/1     Running   0          14m
kube-scheduler-node1  1/1     Running   0          14m
node1:~$ 

```

## #11 - 09/14/2020 09:38 PM - Nico Schottelius

untaint master node for testing:

```

node1:~$ kubectl taint nodes node1 node-role.kubernetes.io/master-
node/node1 untainted

```

## #12 - 09/14/2020 10:25 PM - Nico Schottelius

- Seems like kube-router in v6 only does not accept IPv6 addresses. Bug report created at <https://github.com/cloudnativelabs/kube-router/issues/988>
- According to <https://github.com/cloudnativelabs/kube-router/blob/master/docs/ipv6.md> the state of ipv6 in kube-router is quite limited

## #13 - 09/14/2020 10:51 PM - Nico Schottelius

Resetting and trying with calico:

- <https://docs.projectcalico.org/getting-started/kubernetes/quickstart>

In theory:

```
kubectl create -f https://docs.projectcalico.org/manifests/tigera-operator.yaml
kubectl create -f https://docs.projectcalico.org/manifests/custom-resources.yaml
```

In reality:

```
wget https://docs.projectcalico.org/manifests/calico.yaml
cp calico.yaml calico.yaml.orig
```

```
vi calico.yaml
node1:~$ diff -u calico.yaml.orig calico.yaml
--- calico.yaml.orig
+++ calico.yaml
@@ -3634,6 +3634,9 @@
     - name: DATASTORE_TYPE
       value: "kubernetes"
       # Wait for the datastore.
+     - name: CALICO_ROUTER_ID
+       value: "hash"
+       # Wait for the datastore.
-     - name: WAIT_FOR_DATASTORE
+       value: "true"
+       # Set based on the k8s node name.
@@ -3652,7 +3655,7 @@
       value: "k8s,bgp"
       # Auto-detect the BGP IP address.
-     - name: IP
+       value: "autodetect"
-       value: "none"
+       # Enable IPIP
-     - name: CALICO_IPV4POOL_IPIP
+       value: "Always"
```

Applying:

```
node1:~$ kubectl apply -f calico.yaml
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
serviceaccount/calico-node created
deployment.apps/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
node1:~$
```

calico-node crashes, no error in docker:

calico-node-qfnr9	0/1	PodInitializing	0	16s
calico-node-qfnr9	0/1	RunContainerError	0	28s
calico-node-qfnr9	0/1	RunContainerError	1	36s
calico-node-qfnr9	0/1	CrashLoopBackOff	1	46s
calico-node-qfnr9	0/1	RunContainerError	2	59s

```
node1:~# docker logs d523a007c34b
node1:~# docker logs 577714a283cc
node1:~#
```

Error in the pod description:

```
CriticalAddonsOnly op=Exists
node.kubernetes.io/disk-pressure:NoSchedule op=Exists
node.kubernetes.io/memory-pressure:NoSchedule op=Exists
node.kubernetes.io/network-unavailable:NoSchedule op=Exists
node.kubernetes.io/not-ready:NoExecute op=Exists
node.kubernetes.io/pid-pressure:NoSchedule op=Exists
node.kubernetes.io/unreachable:NoExecute op=Exists
node.kubernetes.io/unschedulable:NoSchedule op=Exists

Events:
Type Reason Age From Message
---- ---- -- ----
Normal Scheduled 119s default-scheduler Successfully assigned kube-system/calico-node-qfnr
9 to node1
Normal Pulling 118s kubelet, node1 Pulling image "calico/cni:v3.16.1"
Normal Pulled 113s kubelet, node1 Successfully pulled image "calico/cni:v3.16.1" in
5.453532909s
Normal Created 112s kubelet, node1 Created container upgrade-ipam
Normal Started 111s kubelet, node1 Started container upgrade-ipam
Normal Pulled 110s kubelet, node1 Container image "calico/cni:v3.16.1" already prese
nt on machine
Normal Created 110s kubelet, node1 Created container install-cni
Normal Started 110s kubelet, node1 Started container install-cni
Normal Pulling 108s kubelet, node1 Pulling image "calico/pod2daemon-flexvol:v3.16.1"
Normal Pulled 105s kubelet, node1 Successfully pulled image "calico/pod2daemon-flexv
ol:v3.16.1" in 3.802406662s
Normal Created 104s kubelet, node1 Created container flexvol-driver
Normal Started 104s kubelet, node1 Started container flexvol-driver
Normal Pulling 103s kubelet, node1 Pulling image "calico/node:v3.16.1"
Normal Pulled 92s kubelet, node1 Successfully pulled image "calico/node:v3.16.1" in
11.364803287s
Normal Created 72s (x3 over 91s) kubelet, node1 Created container calico-node
Warning Failed 72s (x3 over 91s) kubelet, node1 Error: failed to start container "calico-node": Er
ror response from daemon: path /sys/fs is mounted on /sys but it is not a shared mount
Warning BackOff 53s (x3 over 83s) kubelet, node1 Back-off restarting failed container
Normal Pulled 40s (x3 over 91s) kubelet, node1 Container image "calico/node:v3.16.1" already pres
ent on machine
node1:~$
```

trying to fix with:

```
node1:~# mount --make-rshared /
```

(from: <https://github.com/kubernetes/kubernetes/issues/61058>)

Fails to start, because of missing autodetection, which was disabled according to <https://docs.projectcalico.org/networking/ipv6>

```
node1:~# docker logs 1d28c0b279f9
2020-09-14 21:03:07.982 [INFO][9] startup/startup.go 361: Early log level set to info
2020-09-14 21:03:07.982 [INFO][9] startup/startup.go 377: Using NODENAME environment for node name
2020-09-14 21:03:07.982 [INFO][9] startup/startup.go 389: Determined node name: node1
2020-09-14 21:03:07.984 [INFO][9] startup/startup.go 421: Checking datastore connection
2020-09-14 21:03:07.990 [INFO][9] startup/startup.go 445: Datastore connection verified
2020-09-14 21:03:07.990 [INFO][9] startup/startup.go 109: Datastore is ready
2020-09-14 21:03:07.992 [INFO][9] startup/customresource.go 101: Error getting resource Key=GlobalFelixConfig(
name=CalicoVersion) Name="calico
version" Resource="GlobalFelixConfigs" error=the server could not find the requested resource (get GlobalFelix
Configs.crd.projectcalico.org ca
licoversion)
2020-09-14 21:03:07.998 [INFO][9] startup/startup.go 487: Initialize BGP data
2020-09-14 21:03:07.998 [WARNING][9] startup/startup.go 557: No IP Addresses configured, and autodetection is
not enabled
2020-09-14 21:03:07.998 [WARNING][9] startup/startup.go 1310: Terminating
```

Resetting the autodetection fails as follows:

```
node1:~# docker logs 0efcaf6f9021
2020-09-14 21:08:03.993 [INFO][8] startup/startup.go 361: Early log level set to info
2020-09-14 21:08:03.993 [INFO][8] startup/startup.go 377: Using NODENAME environment for node name
2020-09-14 21:08:03.993 [INFO][8] startup/startup.go 389: Determined node name: node1
2020-09-14 21:08:03.995 [INFO][8] startup/startup.go 421: Checking datastore connection
2020-09-14 21:08:04.000 [INFO][8] startup/startup.go 445: Datastore connection verified
2020-09-14 21:08:04.000 [INFO][8] startup/startup.go 109: Datastore is ready
2020-09-14 21:08:04.004 [INFO][8] startup/customresource.go 101: Error getting resource Key=GlobalFelixConfig(
```

```

name=CalicoVersion) Name="calicoversion" Resource="GlobalFelixConfigs" error=the server could not find the requested resource (get GlobalFelixConfigs.crd.projectcalico.org calicoversion)
2020-09-14 21:08:04.015 [INFO][8] startup/startup.go 487: Initialize BGP data
2020-09-14 21:08:04.015 [WARNING][8] startup/startup.go 742: Unable to auto-detect an IPv4 address: no valid IPv4 addresses found on the host interfaces
2020-09-14 21:08:04.015 [WARNING][8] startup/startup.go 509: Couldn't autodetect an IPv4 address. If auto-detecting, choose a different autodetection method. Otherwise provide an explicit address.
2020-09-14 21:08:04.015 [INFO][8] startup/startup.go 325: Clearing out-of-date IPv4 address from this node IP=""
2020-09-14 21:08:04.021 [WARNING][8] startup/startup.go 1310: Terminating
Calico node failed to start
node1:~#

```

Next try: set IP6 to autodetect / replace IP with IP6

```

node1:~$ diff -u calico.yaml.orig calico.yaml
--- calico.yaml.orig
+++ calico.yaml
@@ -3634,6 +3634,9 @@
      - name: DATASTORE_TYPE
        value: "kubernetes"
        # Wait for the datastore.
+      - name: CALICO_ROUTER_ID
+        value: "hash"
+        # Wait for the datastore.
-      - name: WAIT_FOR_DATASTORE
+        value: "true"
+        # Set based on the k8s node name.
@@ -3651,7 +3654,7 @@
      - name: CLUSTER_TYPE
        value: "k8s,bgp"
        # Auto-detect the BGP IP address.
-      - name: IP
+      - name: IP6
        value: "autodetect"
        # Enable IPIP
-      - name: CALICO_IPV4POOL_IPIP
node1:~$ 

```

crashes as well with:

```

node1:~# docker logs edb8a6ca84d0
2020-09-14 21:12:58.952 [INFO][9] startup/startup.go 361: Early log level set to info
2020-09-14 21:12:58.952 [INFO][9] startup/startup.go 377: Using NODENAME environment for node name
2020-09-14 21:12:58.952 [INFO][9] startup/startup.go 389: Determined node name: node1
2020-09-14 21:12:58.954 [INFO][9] startup/startup.go 421: Checking datastore connection
2020-09-14 21:12:58.960 [INFO][9] startup/startup.go 445: Datastore connection verified
2020-09-14 21:12:58.960 [INFO][9] startup/startup.go 109: Datastore is ready
2020-09-14 21:12:58.963 [INFO][9] startup/customresource.go 101: Error getting resource Key=GlobalFelixConfig(name=CalicoVersion) Name="calicoversion" Resource="GlobalFelixConfigs" error=the server could not find the requested resource (get GlobalFelixConfigs.crd.projectcalico.org calicoversion)
2020-09-14 21:12:58.998 [INFO][9] startup/startup.go 487: Initialize BGP data
2020-09-14 21:12:58.998 [WARNING][9] startup/startup.go 742: Unable to auto-detect an IPv4 address: no valid IPv4 addresses found on the host interfaces
2020-09-14 21:12:58.998 [WARNING][9] startup/startup.go 509: Couldn't autodetect an IPv4 address. If auto-detecting, choose a different autodetection method. Otherwise provide an explicit address.
2020-09-14 21:12:58.998 [INFO][9] startup/startup.go 329: Clearing out-of-date IPv6 address from this node IP=""
2020-09-14 21:12:59.005 [WARNING][9] startup/startup.go 1310: Terminating
Calico node failed to start
node1:~#

```

Setting IP to none and IP6 to autodetect:

```

node1:~$ diff -u calico.yaml.orig calico.yaml
--- calico.yaml.orig
+++ calico.yaml
@@ -3634,6 +3634,9 @@
      - name: DATASTORE_TYPE
        value: "kubernetes"
        # Wait for the datastore.
+      - name: CALICO_ROUTER_ID
+        value: "hash"
+        # Wait for the datastore.
-      - name: WAIT_FOR_DATASTORE
+        value: "true"
+        # Set based on the k8s node name.
node1:~$ 

```

```

        value: "true"
    # Set based on the k8s node name.
@@ -3652,6 +3655,8 @@
        value: "k8s,bgp"
    # Auto-detect the BGP IP address.
- name: IP
+     value: "none"
+
- name: IP6
    value: "autodetect"
# Enable IPIP
- name: CALICO_IPV4POOL_IPIP

```

And calico-node is running!

However calico-kube-controllers stays pending:

```

node1:~$ kubectl get pods -n kube-system
NAME                               READY   STATUS    RESTARTS   AGE
calico-kube-controllers-c9784d67d-w46kf   0/1    Pending   0          32s
calico-node-51hhv                   1/1    Running   0          32s
coredns-f9fd979d6-kptlk            0/1    Pending   0          44s
coredns-f9fd979d6-shvw6            0/1    Pending   0          44s
etcd-node1                        0/1    Running   0          59s
kube-apiserver-node1              1/1    Running   0          59s
kube-controller-manager-node1      0/1    Running   0          59s
kube-proxy-hrxmz                  1/1    Running   0          44s
kube-scheduler-node1              0/1    Running   0          58s
node1:~$

```

... describe:

Type	Reason	Age	From	Message
----	-----	----	----	-----
Warning	FailedScheduling	75s (x3 over 81s)	default-scheduler	0/1 nodes are available: 1 node(s) had taint {node.kubernetes.io/not-ready: }, that the pod didn't tolerate.

the node is not ready: <https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/>

Not ready, because the network is not ready. Hmm.....

Conditions:

Type	Status	LastHeartbeatTime	LastTransitionTime	Reason
----	-----	-----	-----	-----
NetworkUnavailable	False	Mon, 14 Sep 2020 21:17:49 +0000	Mon, 14 Sep 2020 21:17:49 +0000	CalicoIsUp
		Calico is running on this node		
MemoryPressure	False	Mon, 14 Sep 2020 21:22:18 +0000	Mon, 14 Sep 2020 21:17:10 +0000	KubeletHasS
sufficientMemory	kubelet has sufficient memory available			
DiskPressure	False	Mon, 14 Sep 2020 21:22:18 +0000	Mon, 14 Sep 2020 21:17:10 +0000	KubeletHasN
oDiskPressure	kubelet has no disk pressure			
PIDPressure	False	Mon, 14 Sep 2020 21:22:18 +0000	Mon, 14 Sep 2020 21:17:10 +0000	KubeletHasS
sufficientPID	kubelet has sufficient PID available			
Ready	False	Mon, 14 Sep 2020 21:22:18 +0000	Mon, 14 Sep 2020 21:17:10 +0000	KubeletNotR
eady	runtime network not ready: NetworkReady=false reason:NetworkPluginNotReady message:docker: n	etwork plugin is not ready: cnf config uninitialized		

Removing taint based on hint from

<https://forum.linuxfoundation.org/discussion/855616/calico-plugin-not-working-with-1-12-kubernetes-please-update-the-k8smaster-sh>

```

node1:~$ kubectl taint nodes node1 node.kubernetes.io/not-ready-
node/node1 untainted

```

Containers are being started now:

```

calico-kube-controllers-c9784d67d-w46kf   0/1    ContainerCreating   0          10m
coredns-f9fd979d6-kptlk                 0/1    ContainerCreating   0          10m
coredns-f9fd979d6-shvw6                 0/1    ContainerCreating   0          10m

```

Resetting, trying with typha. Result: same as above.

#14 - 09/14/2020 11:48 PM - Nico Schottelius

- Description updated

#15 - 09/15/2020 12:33 AM - Nico Schottelius

- Description updated

#16 - 09/15/2020 08:18 PM - Nico Schottelius

Trying cilium, one operator fails with "level=fatal msg="Unable to init cluster-pool allocator" error="IPv4CIDR can not be set if IPv4 is not enabled" subsys=cilium-operator-generic"

```
node1:~$ diff -u cilium-quick-install.yaml.orig cilium-quick-install.yaml
--- cilium-quick-install.yaml.orig
+++ cilium-quick-install.yaml
@@ -38,11 +38,11 @@
 
 # Enable IPv4 addressing. If enabled, all endpoints are allocated an IPv4
 # address.
- enable-ipv4: "true"
+ enable-ipv4: "false"
 
 # Enable IPv6 addressing. If enabled, all endpoints are allocated an IPv6
 # address.
- enable-ipv6: "false"
+ enable-ipv6: "true"
 enable-bpf-clock-probe: "true"
 
 # If you want cilium monitor to aggregate tracing for packets, set this level
node1:~$
```

```
node1:~$ diff -u cilium-quick-install.yaml.orig cilium-quick-install.yaml
--- cilium-quick-install.yaml.orig
+++ cilium-quick-install.yaml
@@ -38,11 +38,11 @@
 
 # Enable IPv4 addressing. If enabled, all endpoints are allocated an IPv4
 # address.
- enable-ipv4: "true"
+ enable-ipv4: "false"
 
 # Enable IPv6 addressing. If enabled, all endpoints are allocated an IPv6
 # address.
- enable-ipv6: "false"
+ enable-ipv6: "true"
 enable-bpf-clock-probe: "true"
 
 # If you want cilium monitor to aggregate tracing for packets, set this level
node1:~$ kubectl create -f cilium-quick-install.yaml
serviceaccount/cilium created
serviceaccount/cilium-operator created
configmap/cilium-config created
clusterrole.rbac.authorization.k8s.io/cilium created
clusterrole.rbac.authorization.k8s.io/cilium-operator created
clusterrolebinding.rbac.authorization.k8s.io/cilium created
clusterrolebinding.rbac.authorization.k8s.io/cilium-operator created
daemonset.apps/cilium created
deployment.apps/cilium-operator created
```

```
node1:~$ kubectl get pods -n kube-system
NAME                      READY   STATUS      RESTARTS   AGE
cilium-cm916                0/1    Init:CrashLoopBackOff  1          24s
cilium-operator-7f4dc846b6-7bfkj  0/1    Pending           0          24s
cilium-operator-7f4dc846b6-r4rv5  0/1    CrashLoopBackOff  1          24s
coredns-f9fd979d6-84rft       1/1    Running          0          2m16s
coredns-f9fd979d6-rsvrn       1/1    Running          0          2m16s
etcd-node1                  1/1    Running          2          2m33s
kube-apiserver-node1         1/1    Running          2          2m32s
kube-controller-manager-node1 1/1    Running          2          2m32s
kube-proxy-xtbcv             1/1    Running          0          2m16s
kube-scheduler-node1         1/1    Running          3          2m33s
node1:~$
```

```
node1:~$ kubectl get pods -n kube-system
NAME                      READY   STATUS      RESTARTS   AGE
cilium-cm916                0/1    Init:CrashLoopBackOff  3          76s
cilium-operator-7f4dc846b6-7bfkj  0/1    Pending           0          76s
cilium-operator-7f4dc846b6-r4rv5  0/1    CrashLoopBackOff  2          76s
```

```

coredns-f9fd979d6-84rft      1/1    Running     0      3m8s
coredns-f9fd979d6-rsvrn      1/1    Running     0      3m8s
etcd-node1                   1/1    Running     2      3m25s
kube-apiserver-node1         1/1    Running     2      3m24s
kube-controller-manager-node1 1/1    Running     2      3m24s
kube-proxy-xtbcv             1/1    Running     0      3m8s
kube-scheduler-node1         1/1    Running     3      3m25s
node1:~$
```

```

level=info msg="--version='false'" subsys=cilium-operator-generic
level=info msg="Cilium Operator 1.8.3 54cf3810d 2020-09-04T14:01:53+02:00 go version go1.14.7 linux/amd64" subsys=cilium-operator-generic
level=info msg="Establishing connection to apiserver" host="https://[2a0a:e5c0:2:13::1]:443" subsys=k8s
level=info msg="Starting apiserver on address 127.0.0.1:9234" subsys=cilium-operator-generic
level=info msg="Connected to apiserver" subsys=k8s
level=info msg="attempting to acquire leader lease kube-system/cilium-operator-resource-lock..." subsys=klog
level=info msg="Operator with ID \"node1-vPeGhfrZeV\" elected as new leader" operator-id=node1-SkVuCxxsHW subsys=cilium-operator-generic
level=info msg="successfully acquired lease kube-system/cilium-operator-resource-lock" subsys=klog
level=info msg="Leading the operator HA deployment" subsys=cilium-operator-generic
level=fatal msg="Unable to init cluster-pool allocator" error="IPv4CIDR can not be set if IPv4 is not enabled" subsys=cilium-operator-generic
node1:~#
```

After rebooting a node it does not get out of the nodenotready state, the following problem occurs:

```

Ready           False   Tue, 15 Sep 2020 18:38:24 +0000   Tue, 15 Sep 2020 18:37:59 +0000   KubeletNotReady
               runtime network no
t ready: NetworkReady=false reason:NetworkPluginNotReady message:docker: network plugin is not ready: cni config uninitialized
```

## #17 - 09/15/2020 08:18 PM - Nico Schottelius

- Description updated

## #18 - 09/15/2020 08:59 PM - Nico Schottelius

Trying the statically routed approach:

```
[20:44] router1.place6:~# ip -6 route add 2a0a:e5c0:2:12::/64 via 2a0a:e5c0:2:2:0:84ff:fe41:f268
[20:47] router1.place6:~# ip -6 route add 2a0a:e5c0:2:13::/64 nexthop via 2a0a:e5c0:2:2:0:84ff:fe41:f268 nexthop via 2a0a:e5c0:2:2:0:84ff:fe41:f263 nexthop via 2a0a:e5c0:2:2:0:84ff:fe41:f269
[20:49] router1.place6:~# ip -6 route add 2a0a:e5c0:2:14::/64 via 2a0a:e5c0:2:2:0:84ff:fe41:f269
[20:49] router1.place6:~#
```

cni configuration on worker nodes:

```
node2:~# cat 10-node2-cni.conf
```

```
{
  "cniVersion": "0.3.0",
  "name": "mynet",
  "type": "bridge",
  "bridge": "cbr0",
  "isDefaultGateway": true,
  "ipMasq": false,
  "hairpinMode": true,
  "ipam": {
    "type": "host-local",
    "ranges": [
      [
        {
          "subnet": "2a0a:e5c0:2:12::/64",
          "gateway": "2a0a:e5c0:2:12::1"
        }
      ]
    ]
  }
}
```

```
node2:~# rm /etc/cni/net.d/*
node2:~# cp 10-node2-cni.conf /etc/cni/net.d/
```

```
node3:~# cat 10-node3-cni.conf
```

```
{
  "cniVersion": "0.3.0",
  "name": "mynet",
  "type": "bridge",
  "bridge": "cbr0",
  "isDefaultGateway": true,
  "ipMasq": false,
  "hairpinMode": true,
  "ipam": {
    "type": "host-local",
    "ranges": [
      [
        {
          "subnet": "2a0a:e5c0:2:14::/64",
          "gateway": "2a0a:e5c0:2:14::1"
        }
      ]
    ]
  }
}
```

```
node3:~# mkdir -p /etc/cni/net.d/
node3:~# cp 10-node3-cni.conf /etc/cni/net.d/
node3:~#
```

Created an nginx deployment:

```
node1:~$ cat nginxdeploy.yaml
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
```

```
node1:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-585449566-9vjt8   1/1     Running   0          3m1s
nginx-deployment-585449566-gxbdp   0/1     ErrImagePull   0          3m1s
node1:~$
```

```
[21:06] bridge:~% curl http://[2a0a:e5c0:2:14::6]
```

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>. <br/>
```

```

Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
[21:06] bridge:~%
```

-> 1 pods is reachable from outside.

Testing services, cmp <https://kubernetes.io/docs/concepts/services-networking/connect-applications-service/>

```

node1:~$ kubectl expose deployment/nginx-deployment
service/nginx-deployment exposed

node1:~$ kubectl get svc nginx-deployment
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
nginx-deployment   ClusterIP  2a0a:e5c0:2:13::1acd  <none>        80/TCP       69s
node1:~$
```

```

node1:~$ curl http://[2a0a:e5c0:2:13::1acd]
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
```

```

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Looks good, let's verify it from outside:

```

[21:12] bridge:~% curl http://[2a0a:e5c0:2:13::1acd]

<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
```

```

<p><em>Thank you for using nginx.</em></p>
</body>
```

```
</html>
```

Testing multi path route via bird directly, works:

```
# Router specific networks
protocol static this_router_v6 {
    ipv6 {};

    route 2a0a:e5c0:e::/48 unreachable;                      # Router local test
    route 2a09:2944::/32 unreachable;                         # Router local test

    route 2a0a:e5c0:2:13::/64
        via 2a0a:e5c0:2:2:0:84ff:fe41:f268
        via 2a0a:e5c0:2:2:0:84ff:fe41:f263
        via 2a0a:e5c0:2:2:0:84ff:fe41:f269;
}
```

**#19 - 09/15/2020 09:31 PM - Nico Schottelius**

- *Description updated*

**#20 - 09/15/2020 10:24 PM - Nico Schottelius**

Testing the ceph connection.

- Opened firewall for workers nodes
- created new ceph pool
- See also:
  - <https://docs.ceph.com/docs/master/rbd/rbd-kubernetes/>
  - <https://dev.to/ingoleajinkya/kubernetes-storage-using-ceph-4lbp>
- result: pvc stays in state pending

```
[21:55:54] black2.place6:~# ceph osd pool create kubernetes 128
pool 'kubernetes' created
[21:57:25] black2.place6:~# ceph osd pool set kubernetes crush_rule hdd-big
set pool 19 crush_rule to hdd-big
[21:57:46] black2.place6:~# ceph osd pool application enable kubernetes rbd
enabled application 'rbd' on pool 'kubernetes'
[21:58:16] black2.place6:~#
[22:00:32] black2.place6:~# ceph auth get-or-create client.kubernetes mon "profile rbd" osd "profile rbd pool=kubernetes" mgr "profile rbd pool=kubernetes"
```

on the nodes:

```
apk add ceph-common
```

(adds support for rbd)

**#21 - 01/03/2024 10:59 AM - Nico Schottelius**

- *Status changed from In Progress to Rejected*