

MeltwaterEngineering

Swapping the wheels of a running car migrate from Amazon VPC CNI to Cilium in Kubernetes

Platform Engineers Stockholms

Federico Hernandez, Simone Sciarriati

 Hallo! Ciao! Hello! Hej! Hola!



Simone Sciarrati

Engineering Team Lead

 @dezmodue

 @dezmodue



Federico Hernandez

Principal Engineer

 @recollier

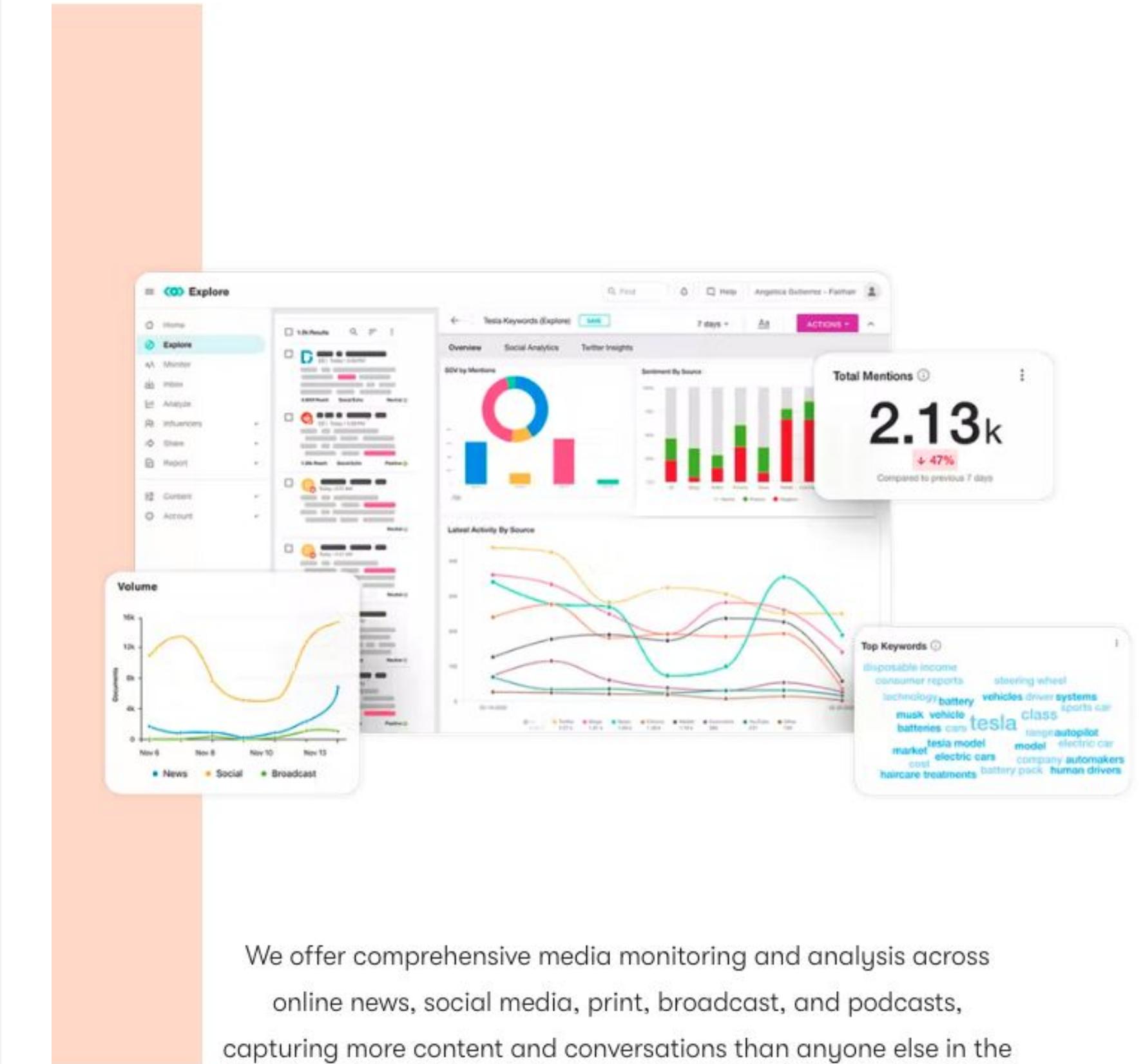
 @recollier

 Meltwater

Meltwater Services

- Media monitoring
- Media outreach
- Social listening
- Social publishing and engagement
- Social influencer management

meltwater.com



We offer comprehensive media monitoring and analysis across online news, social media, print, broadcast, and podcasts, capturing more content and conversations than anyone else in the industry.





70



3000



12000



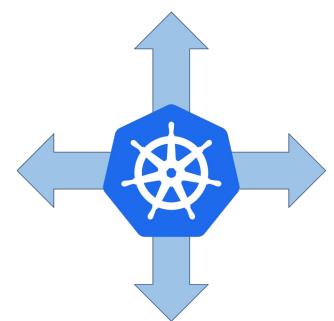
9500



24TB



9 instance types



CA



AutoScaling



DNS



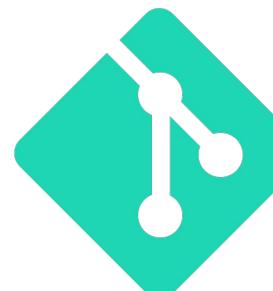
CNI



Observability



Ingress



CICD



kops



Why changing the CNI plugin?

- Additional functionality
- Encryption
- Why Cilium?



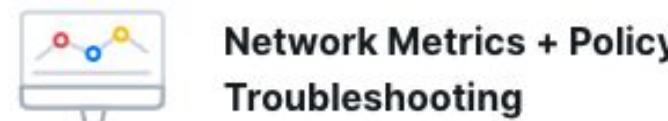
Transparent Encryption



Identity-aware Visibility



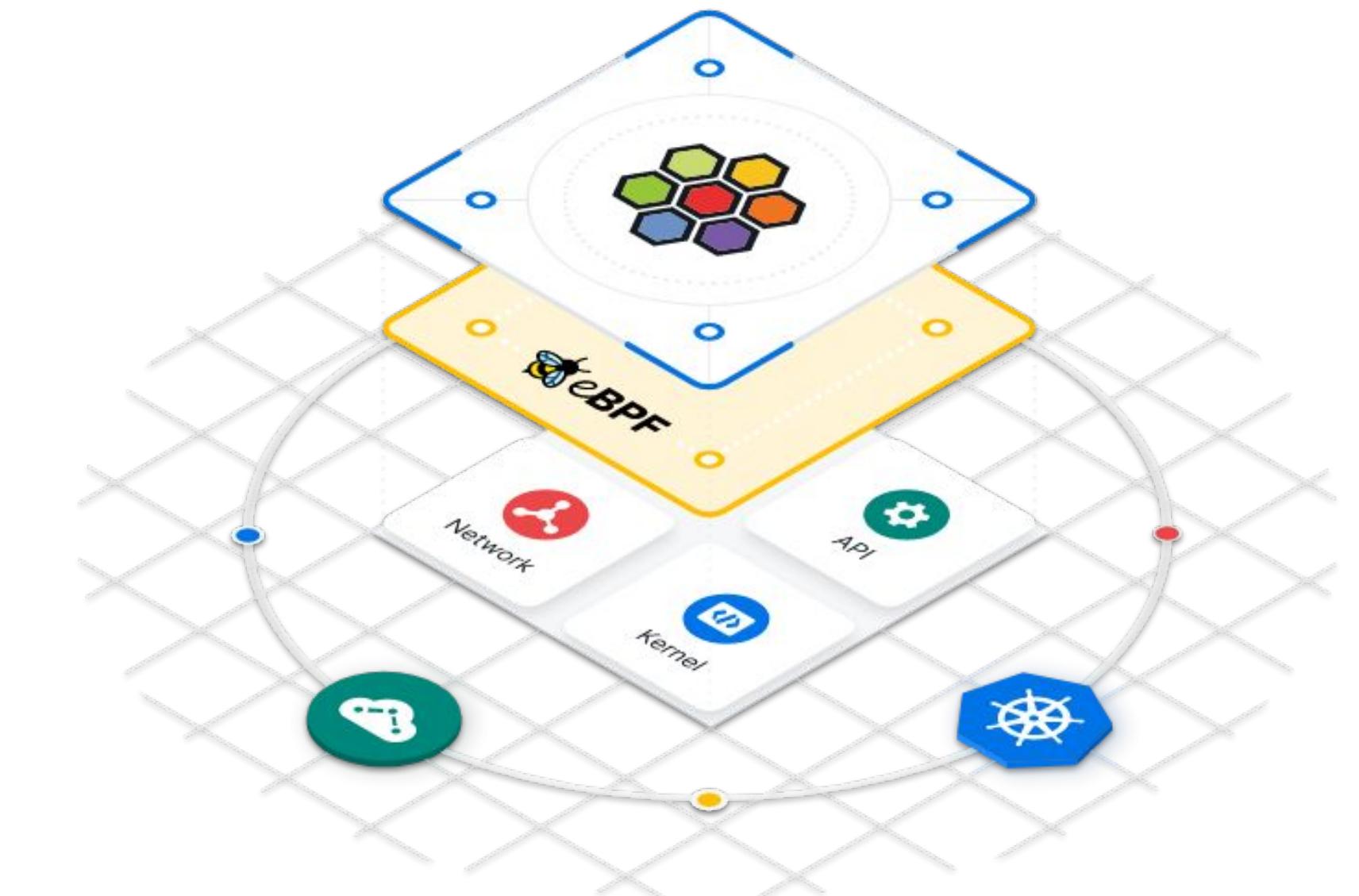
Advanced Network Policy



Network Metrics + Policy
Troubleshooting



Multi-cluster Connectivity



- Seamless
- In place
- Same tooling, and native routing
- No components downtime
- No Kubernetes upgrade



Start small and learn the ropes

 cilium GitHub Stars 11.7k 

Enterprise Learn News and media Documentation

May 11, 2021

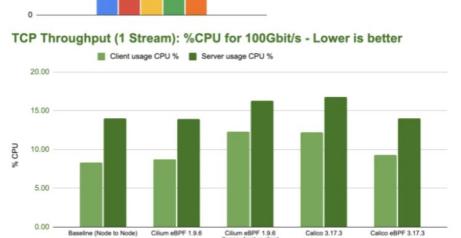
CNI Benchmark: Understanding Cilium Network Performance



TCP Throughput (1 Stream) - Higher is better



TCP Throughput (1 Stream): %CPU for 100Gbit/s - Lower is better



Understanding Cilium Network Performance

Hello 

As more crucial workloads are being migrated to Kubernetes, network selection criteria when deciding what network layer to use are becoming increasingly important. In this blog post, we'll explore the performance characteristics of Cilium based on early benchmarks. Upon popular request, we are also including measurements for Calico.

However, instead of just presenting the numbers, we are going to benchmarking a bit deeper and look at:

- The Throughput Benchmark

[Search or jump to...](#) Pull requests Issues Marketplace Explore

cilium / cilium-perf-networking Public

Code Issues Pull requests Actions Projects Wiki Security Insights

master 2 branches 0 tags Go to file Add file Code

jibi and kkourt cilium: add support for v1.10 and -rc version ... 66796a5 on 12 May 2021 30 commits

playbooks calico: optimize BPF datapath conf 12 months ago

scripts cilium: add support for v1.10 and -rc version 12 months ago

terraform cilium performance evaluation scripts 2 years ago

.gitignore Add zh-lab34-hosts.ini to .gitignore 12 months ago

LICENSE cilium performance evaluation scripts 2 years ago

README.md README: notes 2 years ago

README.md

Introduction

<https://docs.cilium.io/en/latest/operations/performance/>

Additional Details

Installing Cilium

This playbook installs cilium using `kubadm` and `helm`.

Which cilium versions are supported?

We use `cilium-install-with-helm.sh` script to install cilium. You can pass a version available in the cilium helm repo (e.g., `1.8.3`). You can also pass a version such as `1.8` and the script will try and pick the latest one. Finally, you can also use custom docker images as the example below:

 cilium  11721

Search docs

GETTING STARTED

- Introduction to Cilium & Hubble
- Getting Started Guides
 - Installation
 - Network Policy Security Tutorials
- Advanced Networking
 - Setting up Cilium in AWS ENI mode
 - Create an AWS cluster
 - Disable VPC CNI (`aws-node` DaemonSet) (EKS only)
 - Deploy Cilium
 - Create a node group
 - Validate the Installation
 - Specify Environment Variables
 - Enable Hubble for Cluster-Wide Visibility
 - Limitations
 - Troubleshooting
 - Using kube-router to run BGP
 - Using BIRD to run BGP
 - Setting up Cluster Mesh
 - Cilium integration with Flannel (beta)
 - IPVLAN based Networking (beta)

Setting up Cilium in AWS ENI mode

Note

The AWS ENI integration is still subject to some limitations. See [Limitations](#) for details.

Create an AWS cluster

Setup a Kubernetes on AWS. You can use any method you prefer, but for the simplicity of this tutorial, we are going to use `eksctl`. For more details on how to set up an EKS cluster using `eksctl`, see the section [Installation on AWS EKS](#).

```
Copy Line eksctl create cluster --name test-cluster --without-nodegroup
```

Disable VPC CNI (`aws-node` DaemonSet) (EKS only)

If you are running an EKS cluster, you should delete the `aws-node` DaemonSet. Cilium will manage ENIs instead of VPC CNI, so the `aws-node` DaemonSet has to be deleted to prevent conflict behavior.

Note

Once `aws-node` DaemonSet is deleted, EKS will not try to restore it.

```
Copy Line kubectl -n kube-system delete daemonset aws-node
```

Deploy Cilium

Note

First, make sure you have Helm 3 [installed](#). Helm 2 is [no longer supported](#).

Setup Helm repository:

```
Copy Line
```



Adapt tooling

Cilium + kops ?

The screenshot shows a GitHub pull request page for issue #12207. The title of the PR is "Do not set ClusterCIDR for KubeProxy when using CNI networking and kubeProxy.clusterCIDR is not set". The status of the PR is "Merged". The PR has 9 conversations, 1 commit, 8 checks, and 5 files changed. The PR history shows the following interactions:

- dezmodule commented on 27 Aug 2021:** No description provided.
- k8s-ci-robot added labels on 27 Aug 2021:** do-not-merge/contains-merge-commits, needs-ok-to-test
- k8s-ci-robot commented on 27 Aug 2021:** Hi @dezmodule. Thanks for your PR.
I'm waiting for a **kubernetes** member to verify that this patch is reasonable to test. If it is, they should reply with `/ok-to-test` on its own line. Until that is done, I will not automatically test new commits in this PR, but the usual testing commands by org members will still work. Regular contributors should [join the org](#) to skip this step.
Once the patch is verified, the new status will be reflected by the `ok-to-test` label.
I understand the commands that are listed [here](#).
- k8s-ci-robot added labels on 27 Aug 2021:** size/S, cncf-cla: yes
- k8s-ci-robot requested review from joshbranham and zetaab on 27 Aug 2021**
- dezmodule mentioned this pull request on 27 Aug 2021:** Do not set ClusterCIDR when using CNI networking #12208

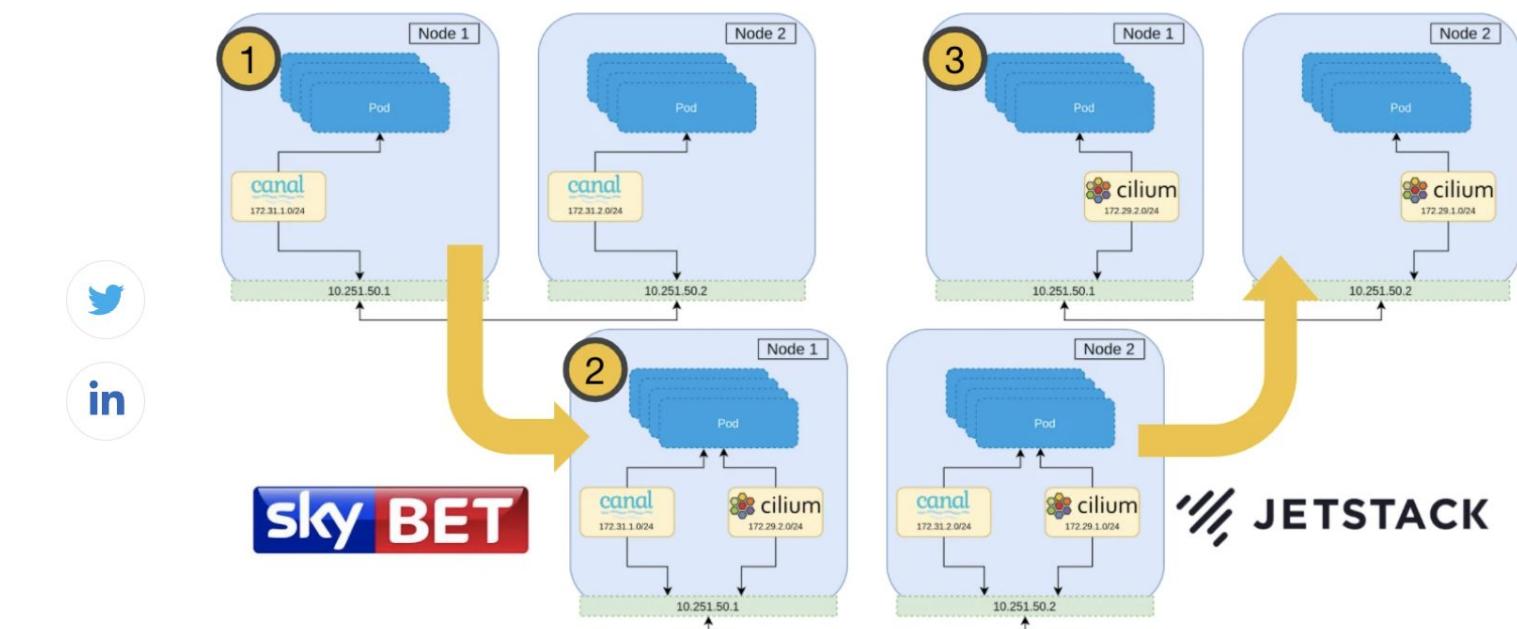
On the right side of the screen, there are several sidebar panels:

- Review:** joh, jos, zet
- Assign:** joh
- Labels:** appro, ok-to
- Project:** None yet
- Milestones:** v1.23
- Development Status:** Success, these issues are ready for review
- Notifications:** You're in 10 notifications

Oct 06, 2020

How to perform a CNI Live Migration from Flannel+Calico to Cilium

- Inspiration from blog post
- Easier with native routing
- Intermediate state with both CNI plugins
- Multiple days rollout



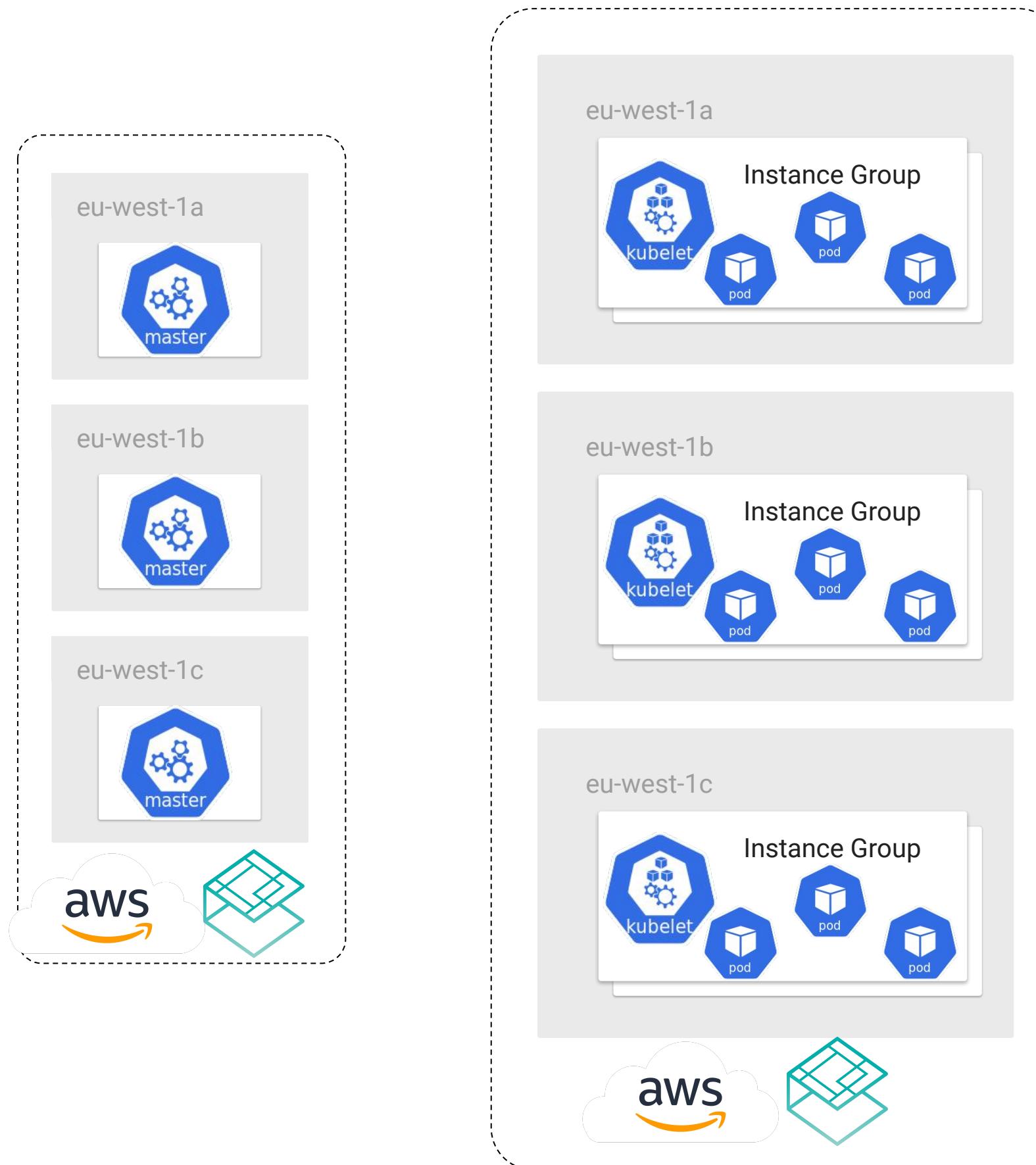
This is a guest blog by Josh Van Leeuwen and covers how Josh implemented a CNI live migration for a customer, [Sky Betting and Gaming](#), to live migrate a cluster running the Canal CNI plugin (Flannel+Calico) to Cilium. Josh is working as a Customer Reliability Engineer for Jetstack. Read on to hear what Josh has to say...

In this post we'll discuss why one might want to change CNIs, what I have learnt developing a solution for live migration, and how it all works.

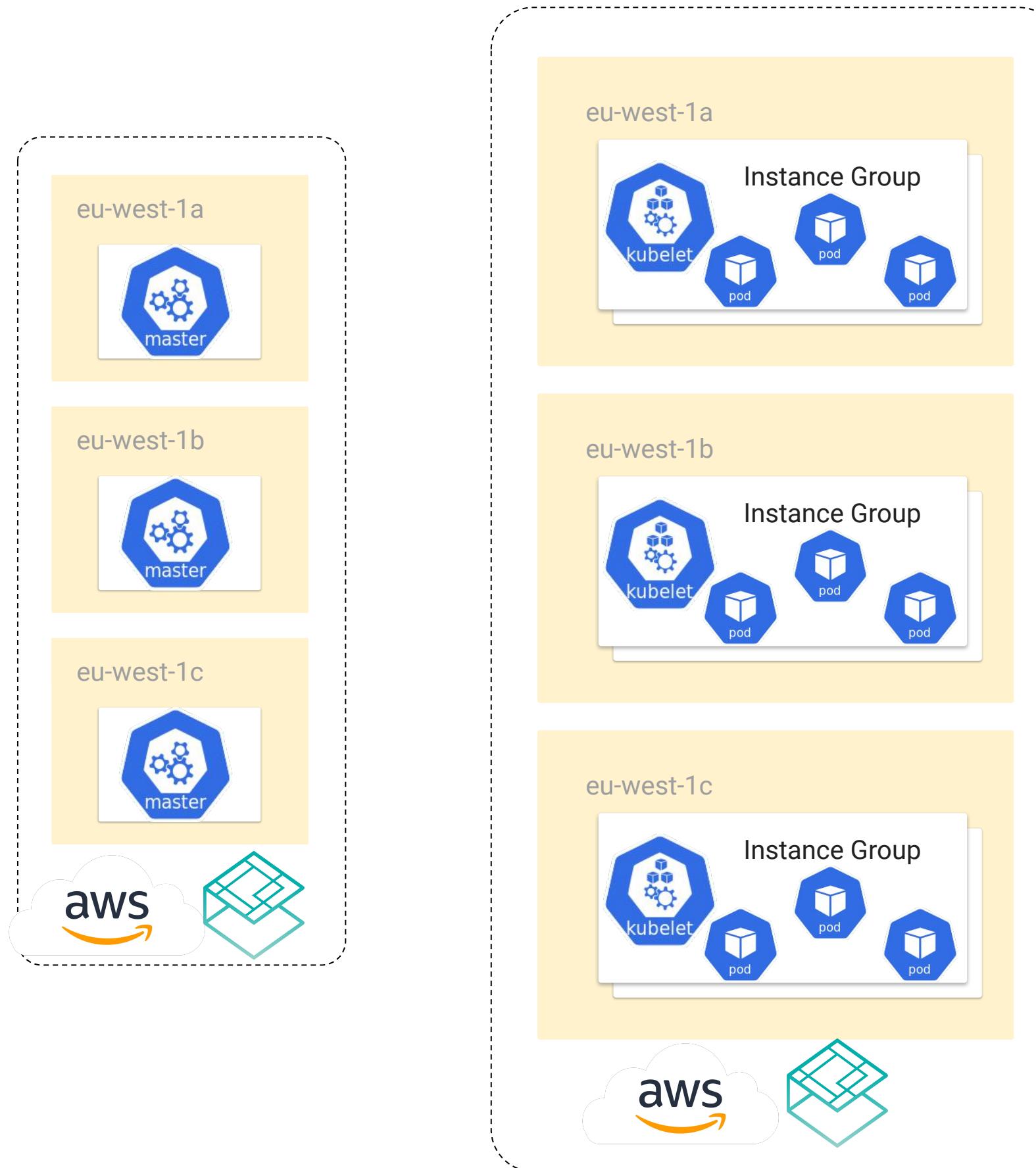
What is CNI, and why change it?

Container Network Interface (CNI) is a big topic, but in short, CNI is a set of specifications that define an interface used by container orchestrators to set up networking between containers. In the [Kubernetes space](#), the Kubelet is responsible for calling the CNI installed on the cluster so Pods are attached to the Kubernetes cluster network during creation, and its resources are properly released during deletion. CNIs can also be responsible for more advanced features than just setting up routes in the cluster, such as network policy enforcement, encryption, load balancing, etc.

Initial state

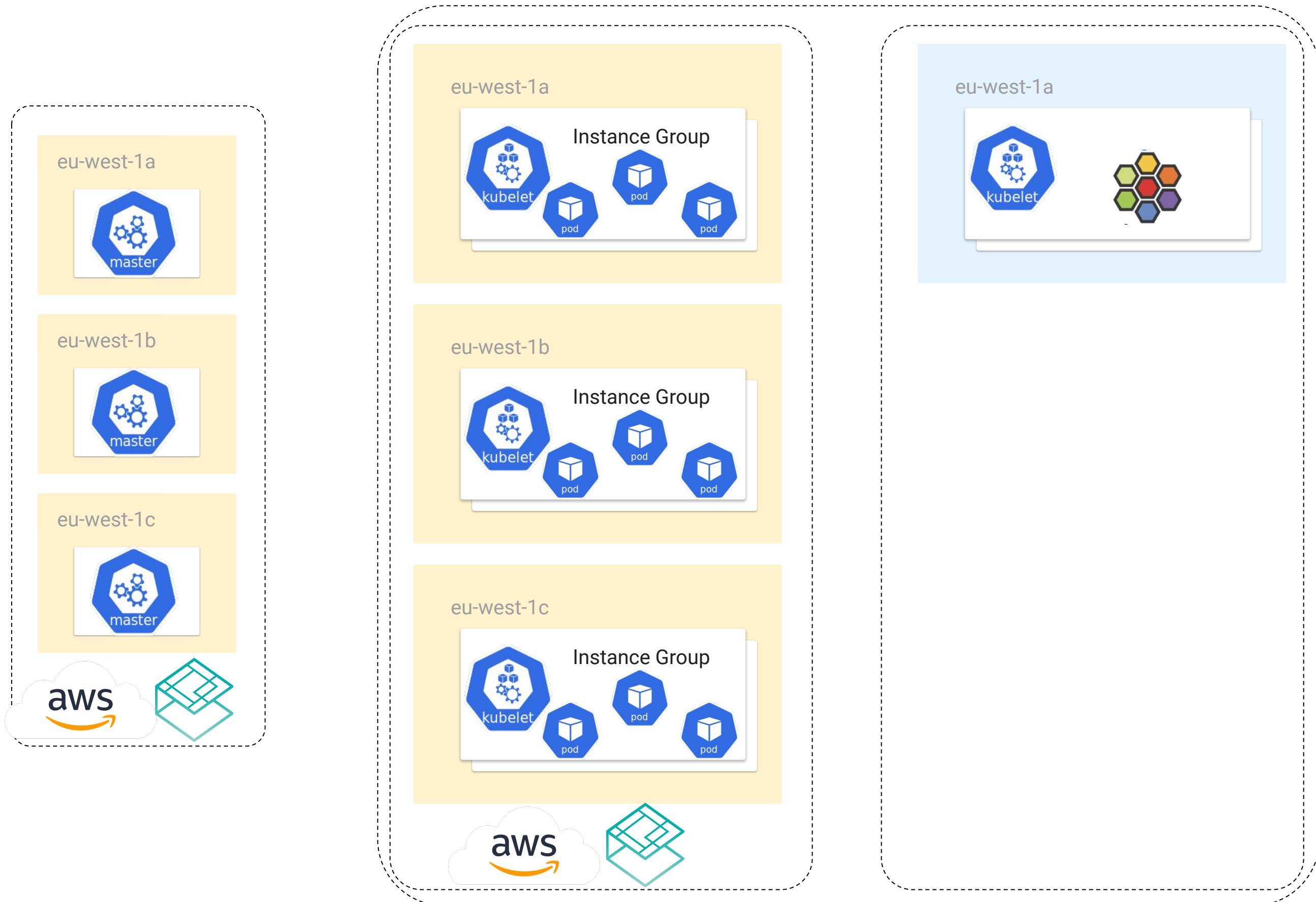


Labelling



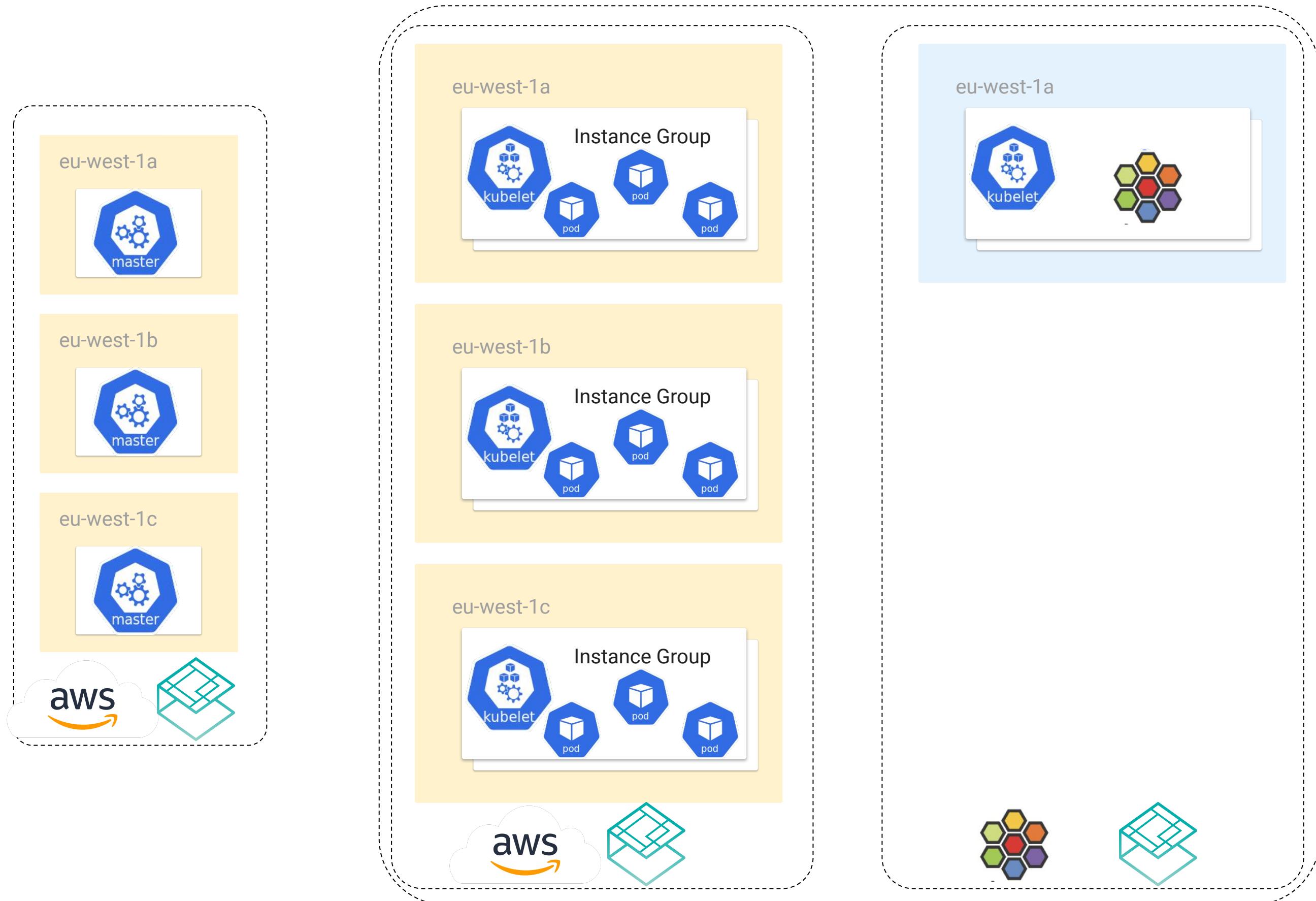
label cniplugin=aws

Cilium operator



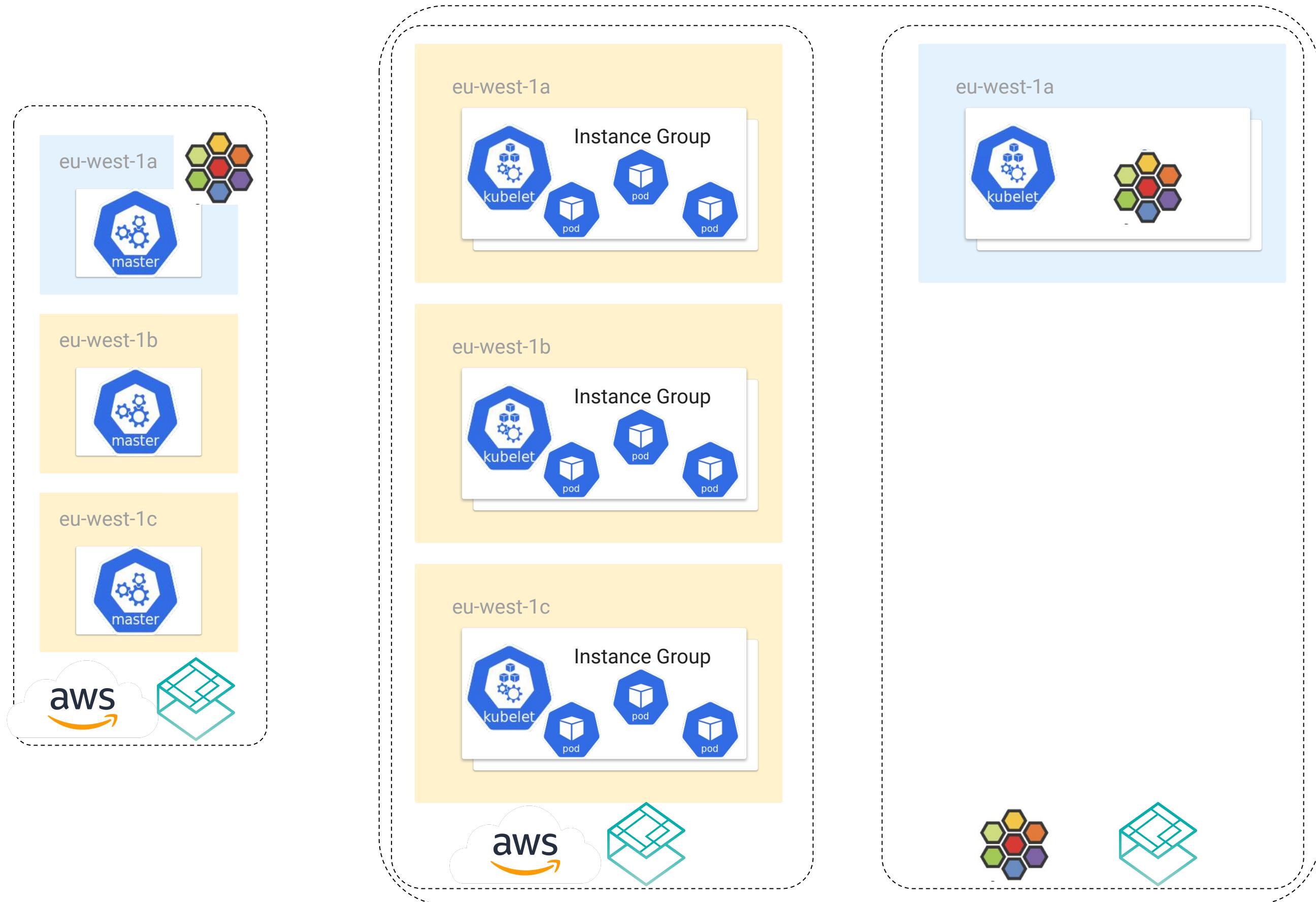
Cilium Operator

Cilium daemonset

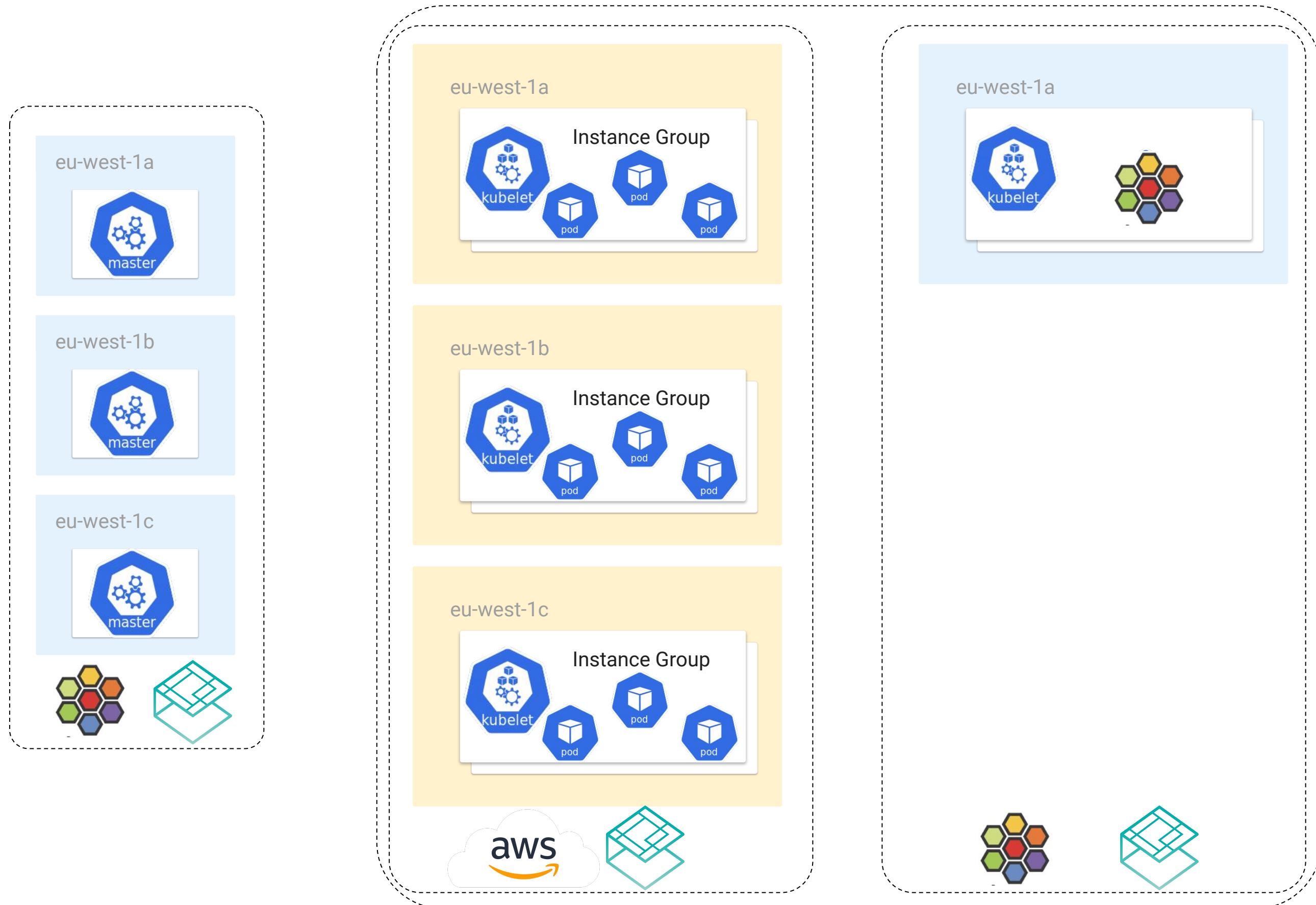


Cilium Agent

Control plane migration

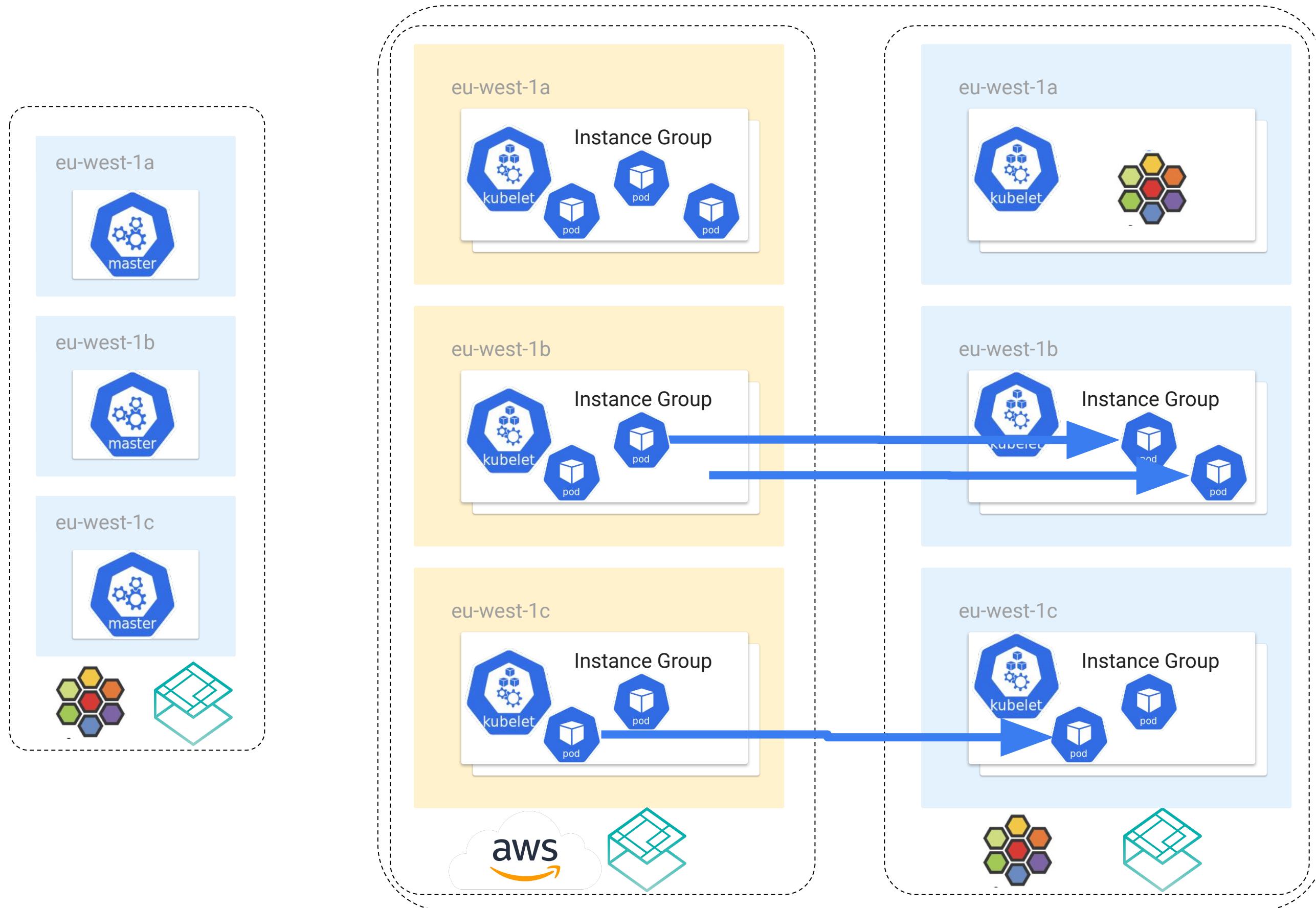


Control plane migration





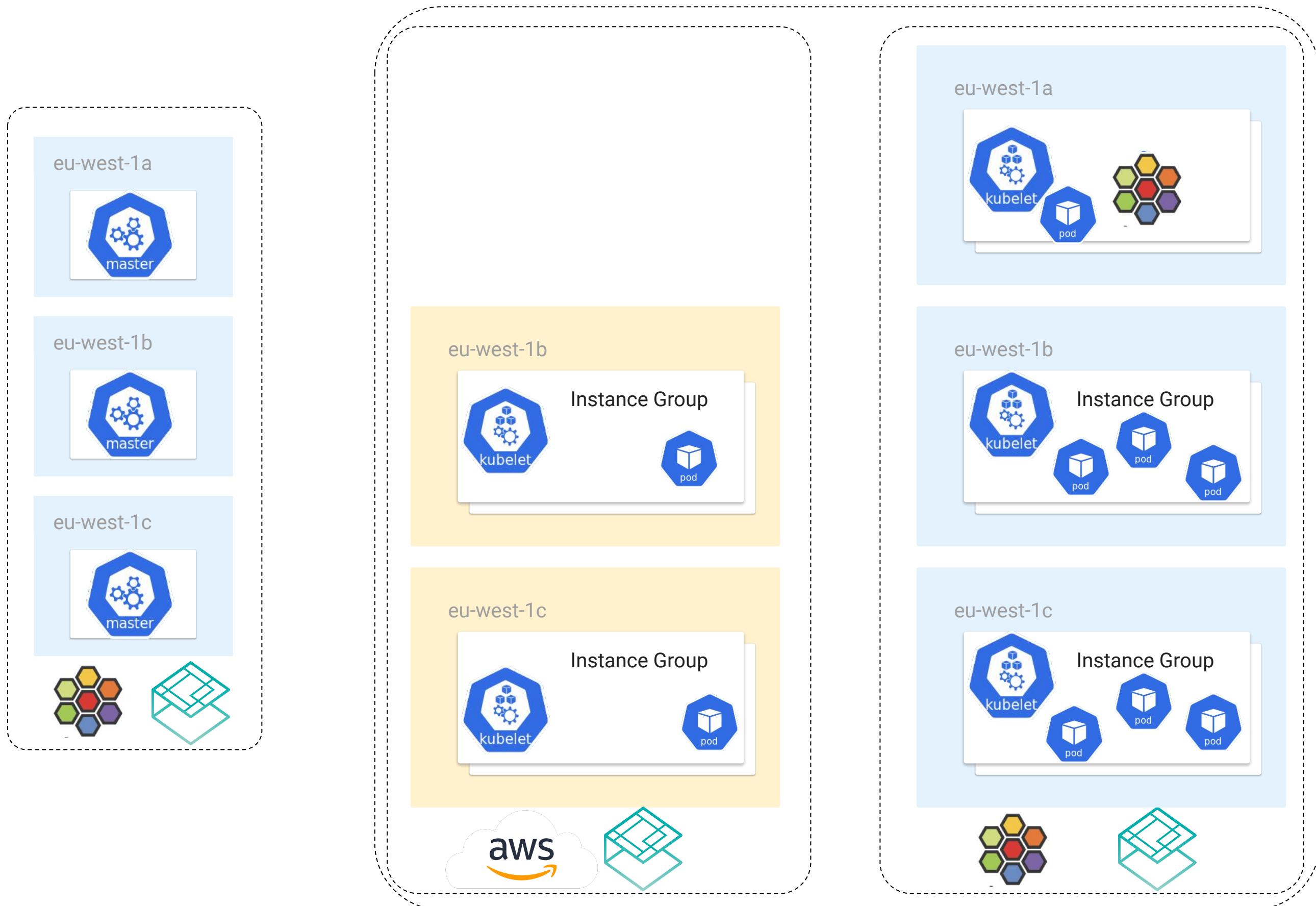
Pods migration



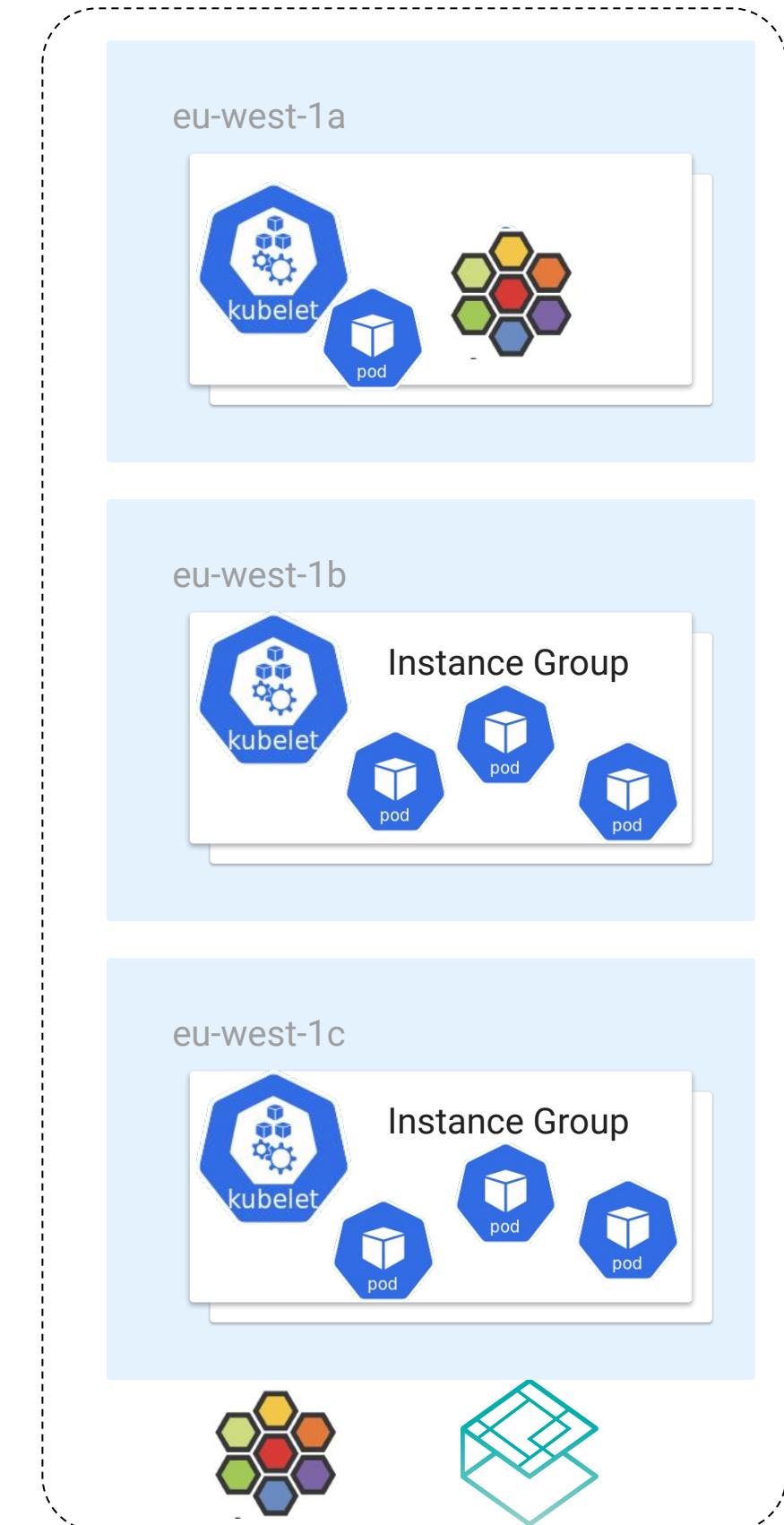
Migrate workloads



Decommissioning



Final state



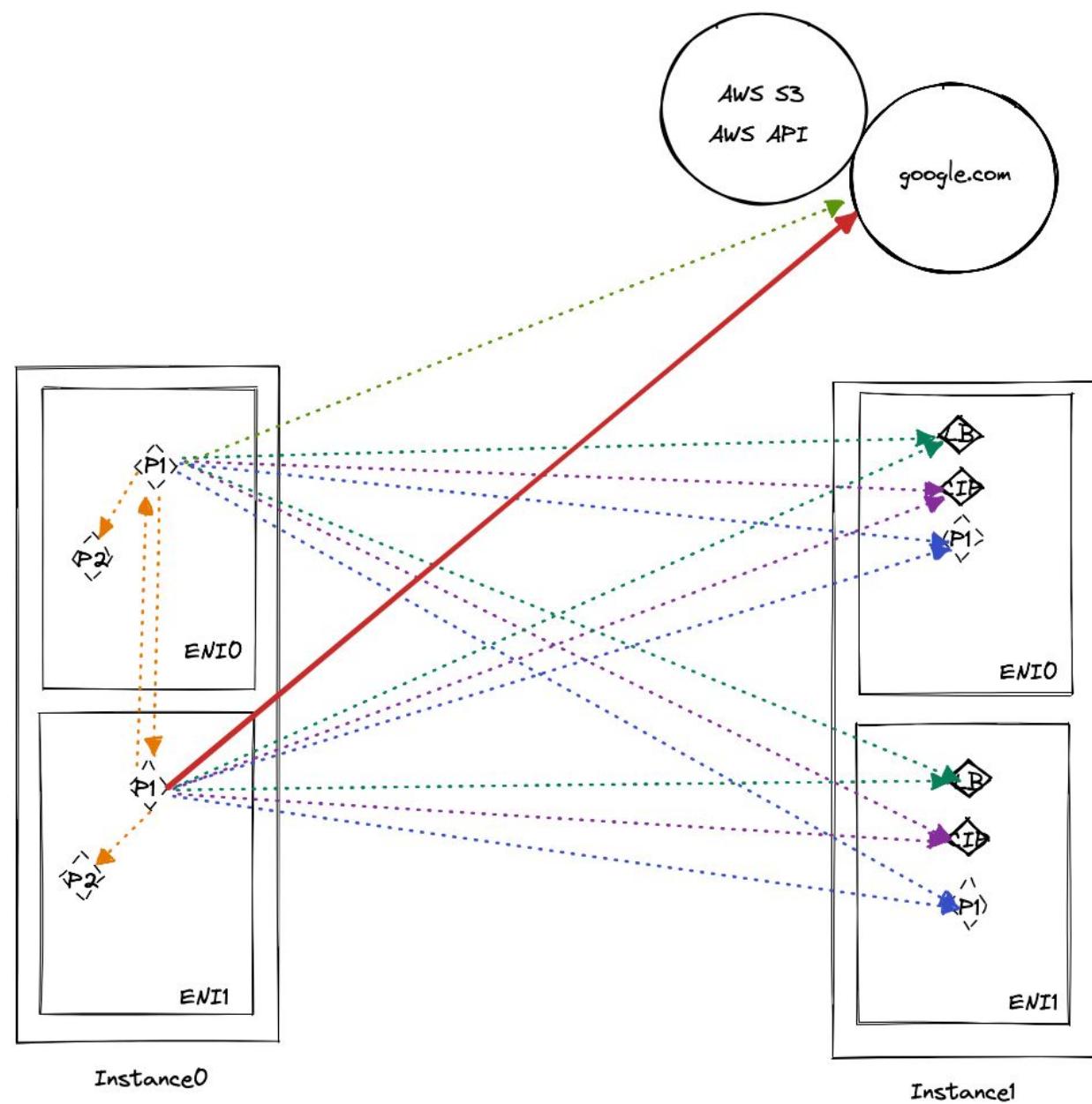
- Codified the procedure
- Testing
- Communication





- The value of dogfooding
- More controlled environment

More tooling - Bravo pinger



egressMasqueradeInterfaces=ens+



>Show time



- Increased resource requests from daemonsets
- Instances with lower IP limits



Success!!

```
> date;cilium status; date  
Thu Feb 10 18:24:25 CET 2022
```



Cilium:	OK
Operator:	OK
Hubble:	OK
ClusterMesh:	disabled

DaemonSet	cilium	Desired: 404, Ready: 404/404, Available: 404/404
Deployment	cilium-operator	Desired: 2, Ready: 2/2, Available: 2/2
Deployment	hubble-relay	Desired: 1, Ready: 1/1, Available: 1/1
Deployment	hubble-ui	Desired: 1, Ready: 1/1, Available: 1/1
Containers:	cilium	Running: 404
	cilium-operator	Running: 2
	hubble-relay	Running: 1
	hubble-ui	Running: 1
Cluster Pods:	7560/7928 managed by Cilium	
Image versions	hubble-relay	quay.io/cilium/hubble-relay:v1.10.4@sha256:be17169d2b68a974e9e27bc1
	hubble-ui	quay.io/cilium/hubble-ui:v0.7.9@sha256:e0e461c680ccd083ac24fe4f9e19
	hubble-ui	quay.io/cilium/hubble-ui-backend:v0.7.9@sha256:632c938ef6ff30e3a080
	hubble-ui	docker.io/envoyproxy/envoy:v1.18.2@sha256:e8b37c1d75787dd1e712ff389
	cilium	quay.io/cilium/cilium:v1.10.4@sha256:7d354052ccf2a7445101d78cebd144
	cilium-operator	quay.io/cilium/operator-aws:v1.10.4@sha256:45df7a09f8278a9c2313fa7d

```
Thu Feb 10 18:28:48 CET 2022
```



It is DNS!

It is always DNS!

It is not DNS...

getaddrinfo

ENOTFOUND

It is AWS Limits!

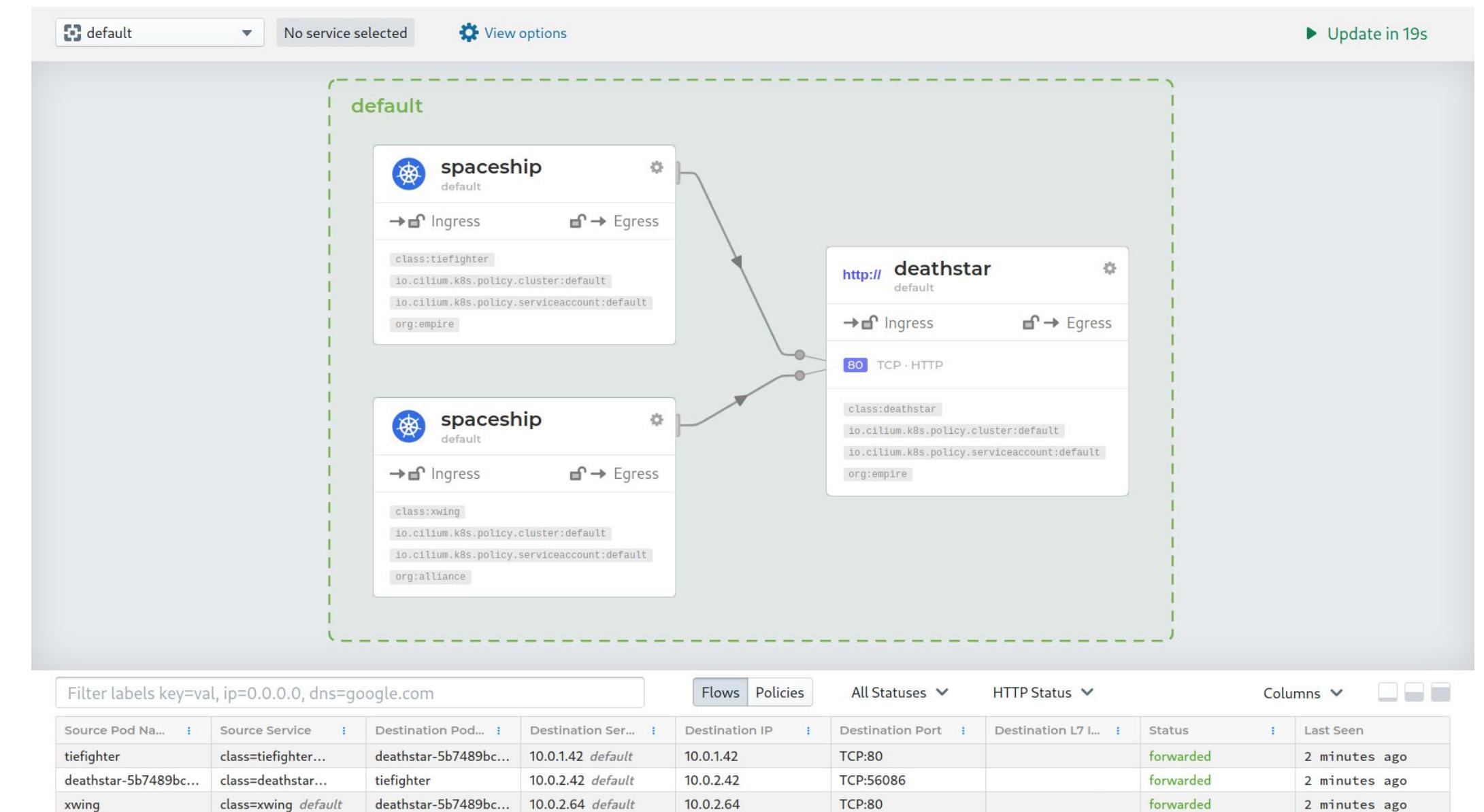
It is not AWS limits...

Network Performance Metrics

Additional instance level network performance metrics are available on instances using ENA both for Windows and Linux operating systems. These include:

- *bw_in_allowance_exceeded* and *bw_out_allowance_exceeded* indicates the number of packets that are shaped and dropped because of instance aggregate bandwidth exceeded BW allowance for the instance.
- *pps_allowance_exceeded* indicates number of packets shaped and dropped due to Packet Per Second (PPS) exceeding the allowance for the instance. PPS allowance is enforced separately to the overall bandwidth allowance and while the instance may still be under overall bandwidth allowance the PPS allowance may exceed if the mean packet size is small.
- *conntrack_allowance_exceeded* indicates the number of packets shaped and dropped due to [exhaustion of tracked session allowance for the instance](#), new sessions will fail to establish once this allowance is exceeded. Sessions are restored once instance session count drops below the allowance.
- *linklocal_allowance_exceeded* indicates the number of packets shaped and dropped due to PPS rate allowance exceeded for local services such as Route 53 DNS Resolver, Instance Metadata Service, Amazon Time Sync Service. This often points to suboptimal design choices or misconfiguration. This allowance is same across instances.

Could it be Cilium?



```
cilium sysdump --help
```

Routes

```
root@ip-10-103-220-24:/home/cilium# ip rule | fgrep 10.103.205.47
20:      from all to 10.103.205.47 lookup main
111:     from 10.103.205.47 to 10.103.0.0/16 lookup 10
111:     from 10.103.205.47 to 10.0.0.0/8 lookup 11
```

```
root@ip-10-103-220-24:/home/cilium# ip route show table 10
default via 10.103.192.1 dev ens5
10.103.192.1 dev ens5 scope link
```

```
root@ip-10-103-220-24:/home/cilium# ip route show table 11
default via 10.103.192.1 dev ens6
10.103.192.1 dev ens6 scope link
```

Contribute

Showing 2 changed files with 42 additions and 1 deletion.

daemon/cmd/ipam.go

```

27 27 @@ -27,6 +27,7 @@ import (
28 28     "github.com/cilium/cilium/pkg/datapath"
29 29     "github.com/cilium/cilium/pkg/linuxrouting"
30 30     "github.com/cilium/cilium/pkg/defaults"
31 31     "github.com/cilium/cilium/pkg/ip"
32 32     "github.com/cilium/cilium/pkg/ipam"
33 33     "github.com/cilium/cilium/pkg/ipam/option"
34 34     "github.com/cilium/cilium/pkg/logging/logfields"
35 35 )
36 36 @@ -250,7 +251,20 @@ func (d *Daemon) allocateDatapathIPs(family datapath.NodeAddressingFamily) (route
37 37     node.SetRouterInfo(routingInfo)
38 38 }
39 39 -
40 40 +    cidrs := make([]net.IPNet, 0, len(result.CIDRs))
41 41 +    for _, k := range result.CIDRs {
42 42 +        var s net.IPNet
43 43 +        ip, mask, _ := net.ParseCIDR(k)
44 44 +        s.IP = ip
45 45 +        s.Mask = mask.Mask
46 46 +        cidrs = append(cidrs, &s)
47 47 }
48 48     resultcidr, _ := iputil.CoalesceCIDRs(cidrs)
49 49     newresult := make([]string, len(resultcidr))
50 50     for i, k := range resultcidr {
51 51         newresult[i] = k.String()
52 52     }
53 53     result.CIDRs = newresult
54 54 }
55 55 -
56 56 +    if err != nil {
57 57     return fmt.Errorf("unable to allocate health IPs: %s, see https://cilium.link/ipam-range-full", err)
58 58 }
59 59 }
```

pkg/datapath/loader/base.go

```

37 37 @@ -37,6 +37,7 @@ import (
38 38     datapathoption "github.com/cilium/cilium/pkg/datapath/option"
39 39     "github.com/cilium/cilium/pkg/datapath/prefilter"
40 40     "github.com/cilium/cilium/pkg/defaults"
41 41     "github.com/cilium/cilium/pkg/ip"
42 42     "github.com/cilium/cilium/pkg/ipam"
43 43     "github.com/cilium/cilium/pkg/logging/logfields"
44 44     "github.com/cilium/cilium/pkg/node"
45 45 )
46 46 @@ -167,6 +168,18 @@ func addENIRules(sysSettings []sysctl.Setting, nodeAddressing datapath.NodeAddressing, IP: nodeAddressing.IPV4(), Router, Mask: net.CIDRMask32, 32),
47 47     cidrs2 := make([]net.IPNet, 0, 8)
48 48     for _, cidr := range cidrs {
49 49         cidrs2 = append(cidrs2, &cidr)
50 50     }
51 51     resultcidr, _ := iputil.CoalesceCIDRs(cidrs2)
52 52     cidrs3 := make([]net.IPNet, 0, 8)
53 53     for _, cidr := range resultcidr {
54 54         cidrs3 = append(cidrs3, &cidr)
55 55     }
56 56     cidrs = cidrs3
57 57     for _, cidr := range cidrs {
58 58         if err = linuxrouting.SetupRules(routerIP, &cidr, info.GetMac().String(), info.GetInterfaceNumber()); err != nil {
59 59             return nil, fmt.Errorf("unable to install ip rule for cilium_host: %w", err)
59 59 }
```



THANK YOU