Плагіни мережевої взаємодії для стеку Docker/Kubernetes

Виконав: студент групи ДА-62, Соловей В. О.

Науковий керівник: доцент, к.т.н., Гіоргізова-Гай В. Ш.

Актуальність проблеми

- Кластери Kubernetes мають багаторівневу структуру;
- Kubernetes використовує модель призначення власних IP- адрес подам* для маршрутизації трафіку;
- Існує велика кількість сторонніх рішень для побудови мережі Kubernetes зі своїми недоліками та перевагами;

- *Під – основна структурна одиниця Kubernetes, об’єднує контейнери що мають працювати за однаковою логікою.

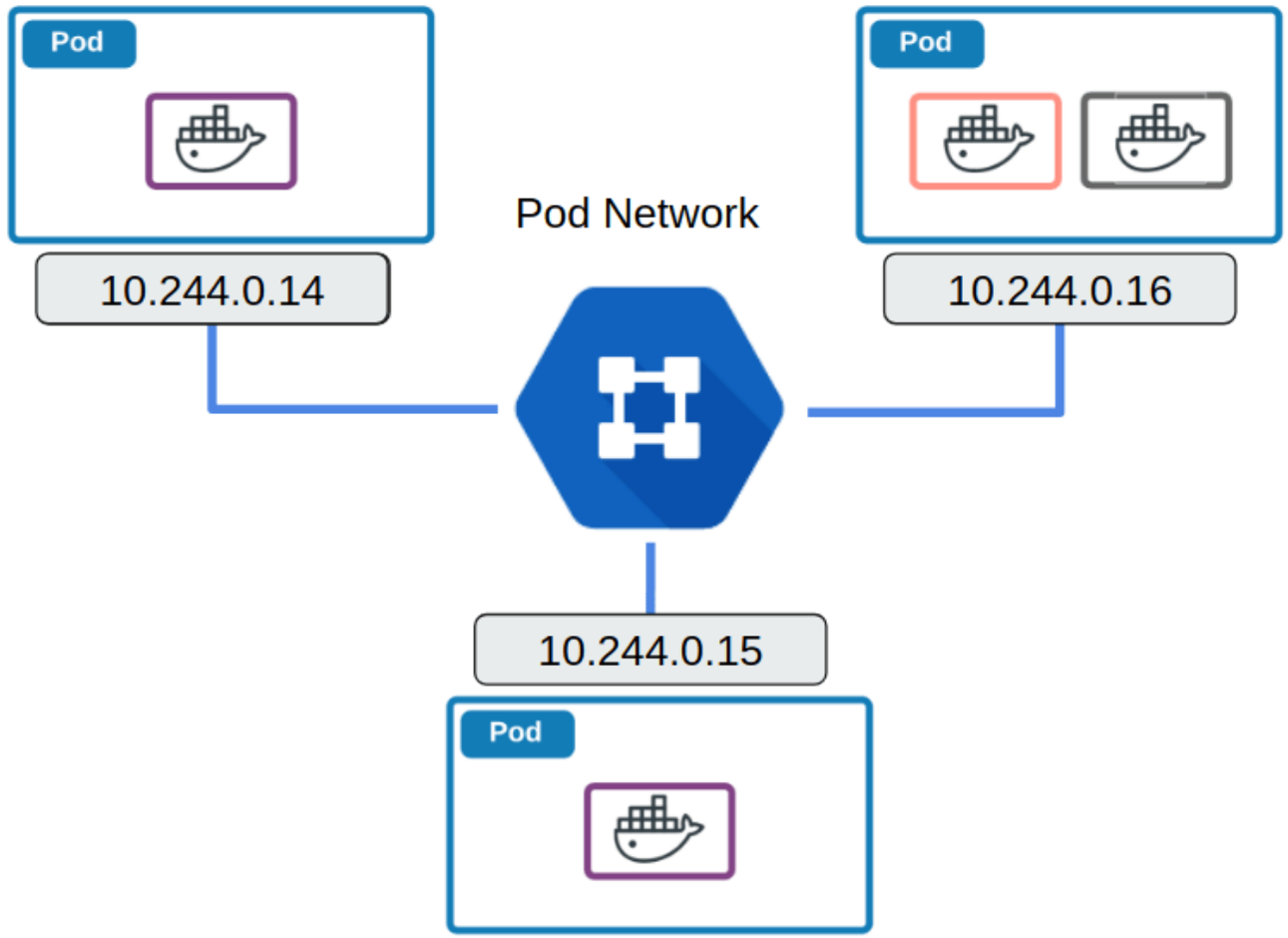
Мета і завдання дипломної роботи

- Провести аналіз мережевої моделі Kubernetes;
- Провести аналіз концепції та архітектури CNI;
- Дослідити існуючі підходи до організації мережевої взаємодії та їх реалізації у популярних мережевих засобах;
- Зробити порівняльну характеристику обраних мережевих плагінів;
- Визначити умови найкращого застосування кожного з рішень, в яких проявляються їх переваги.

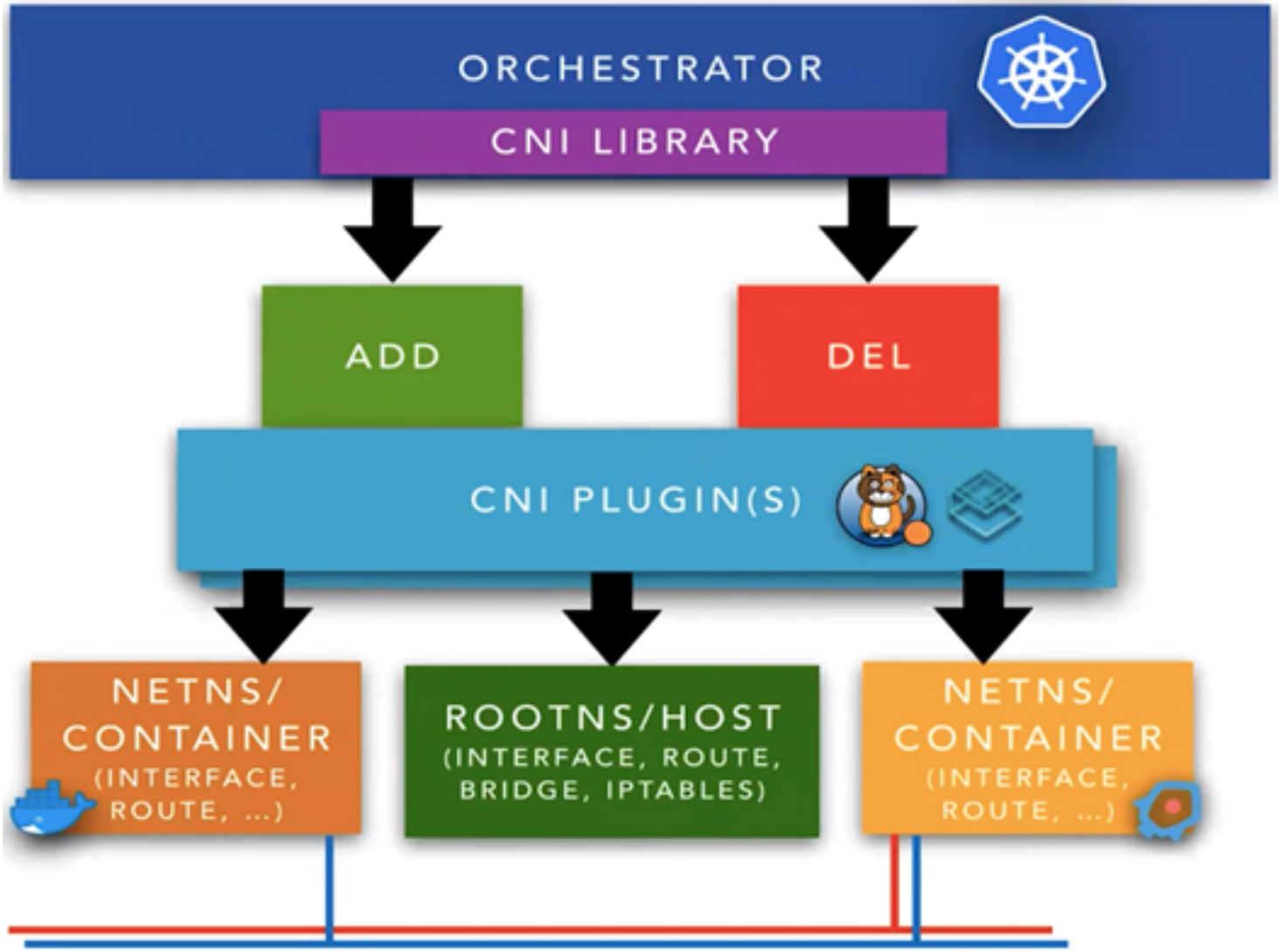
Мережеві задачі у кластері Kubernetes

- Високозв'язні комунікації між контейнерами у поді
- Комунікації між подами;
- Зв'язок між службами;
- Мережева взаємодія між вузлами;
- Зв’язок служб із зовнішніми ресурсами.

Ці проблеми вирішуються за допомоги специфікації CNI.



Специфікація CNI



Способи організації мережі у Kubernetes

- Вбудовані інструменти Linux
  - BPF та XDP – інструменти фільтрації та швидкої передачі трафіку у Linux
- Накладні мережі
  - VXLAN – протокол інкапсуляції кадрів Ethernet у UDP пакети
  - BGP – протокол маршрутизації між автономними мережами

```
73  kind: ConfigMap
74  apiVersion: v1
75  metadata:
76    name: kube-flannel-cfg
77    namespace: kube-system
78    labels:
79      tier: node
80      app: flannel
81  data:
82    cni-conf.json: |
83      {
84        "name": "cbr0",
85        "cniVersion": "0.3.1",
86        "plugins": [
87          {
88            "type": "flannel",
89            "delegate": {
90              "hairpinMode": true,
91              "isDefaultGateway": true
92            }
93          },
94          {
95            "type": "portmap",
96            "capabilities": {
97              "portMappings": true
98            }
99          }
100       ]
101     }
102   net-conf.json: |
103     {
104       "Network": "10.244.0.0/16",
105       "Backend": {
106         "Type": "vxlan"
107       }
108     }
```

Розгортання кластеру із Flannel

На рисунку зображено частину файлу конфігурації Flannel із налаштуваннями плагіну.

У низу зображення є секція налаштування діапазону IP адреси мережі та модель її організації

Перевірка роботи Flannel

```
MacBook-Air-Dice:~ dice$ kubectl get pods --all-namespaces
NAMESPACE     NAME                                    READY   STATUS    RESTARTS   AGE
kube-system   coredns-66bff467f8-bmj55                1/1     Running   0          2m34s
kube-system   coredns-66bff467f8-mrzjs                1/1     Running   0          2m34s
kube-system   etcd-minikube                           1/1     Running   0          2m38s
kube-system   kube-apiserver-minikube                 1/1     Running   0          2m38s
kube-system   kube-controller-manager-minikube        1/1     Running   0          2m38s
kube-system   kube-flannel-ds-amd64-kt7qv             1/1     Running   2          31s
kube-system   kube-proxy-vsgks                        1/1     Running   0          2m34s
kube-system   kube-scheduler-minikube                 1/1     Running   0          2m38s
kube-system   storage-provisioner                     1/1     Running   0          2m39s
```

```
MacBook-Air-Dice:~ dice$ kubectl run --namespace=flannel-test access --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ # wget -q --timeout=1 nginx -O -
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
/ # wget -q --timeout=5 google.com -O -
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="uk"><head><meta content="text/html;
```

Тестування кластеру із Calico

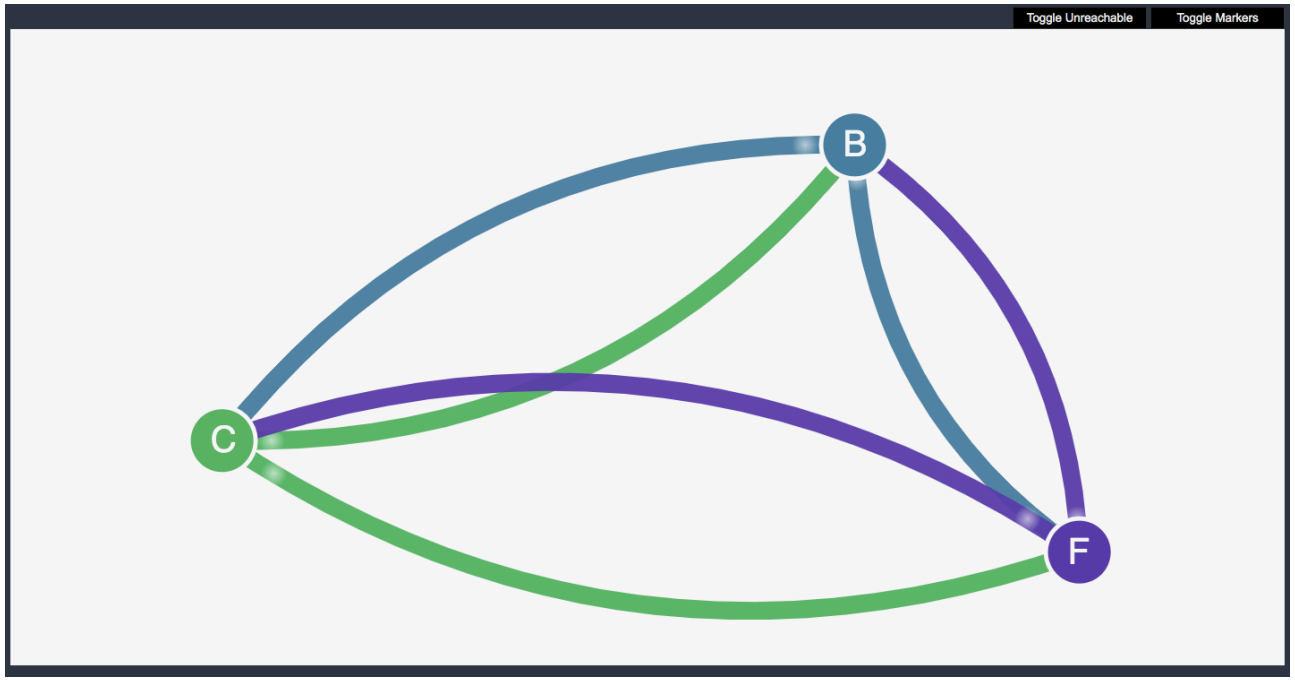
Візуалізація трафіку між трьома тестовими подами у кластері з Calico без налаштованих мережевих політик .

B - позначає бекенд сайту;

C – позначає користувача;

F – фронтенд.

Як бачимо, усі поди вільно обмінюються трафіком, що не є правильним.



Застосування мережевих політик у кластері із Calico

На цьому слайді приведено мережеві політик, що мають дозволити трафік лише між клієнтом та фронтендом, та фронтендом і бекендом.

```yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny
spec:
  podSelector:
    matchLabels: {}

```

```yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
  name: allow-ui
spec:
  podSelector:
    matchLabels: {}
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              role: management-ui
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: client
  name: allow-ui
spec:
  podSelector:
    matchLabels: {}
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              role: management-ui
```

```yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
  name: frontend-policy
spec:
  podSelector:
    matchLabels:
      role: frontend
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              role: client
      ports:
        - protocol: TCP
          port: 80
```

```yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
  name: backend-policy
spec:
  podSelector:
    matchLabels:
      role: backend
  ingress:
    - from:
        - podSelector:
            matchLabels:
              role: frontend
      ports:
        - protocol: TCP
          port: 6379
```

Тестування кластеру із Calico

Візуалізація трафіку між трьома тестовими подами у кластері з Calico з правильно налаштованими мережевими політиками.

B - позначає бекенд сайту;

C – позначає користувача;

F – фронтенд.

Як бачимо, ми обмежили небажаний трафік.

Тестування кластеру із Weave Net

```
MacBook-Air-Dice:~ dice$ kubectl run busybox --rm -ti --image=busybox -- /bin/sh
wIf you don't see a command prompt, try pressing enter.
/ # wget --spider --timeout=1 nginx
Connecting to nginx (10.110.215.253:80)
remote file exists
/ #
```

Пояснення до рисунків (нумерація згори до низу).

Рисунок 1 – перевірка зв’язку у кластері за допомогою wget запиту до nginx поду.

```
1  apiVersion: networking.k8s.io/v1
2  kind: NetworkPolicy
3  metadata:
4    name: access-nginx
5  spec:
6    podSelector:
7      matchLabels:
8        app: nginx
9    ingress:
10   - from:
11     - podSelector:
12         matchLabels:
13           access: "true"
14
```

Рисунок 2 – мережева політика, що обмежує доступ до nginx усім подам без мітки «access: true».

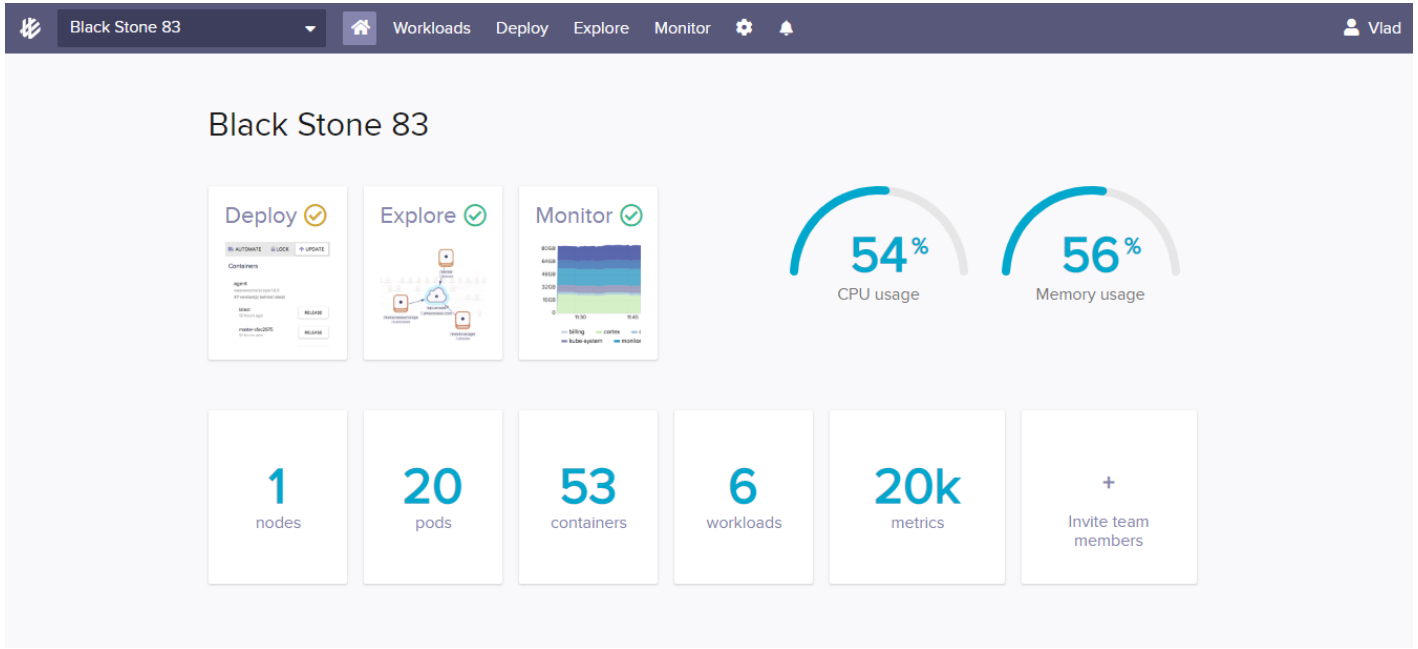
```
MacBook-Air-Dice:~ dice$ kubectl run busybox --rm -ti --image=busybox -- /bin/sh
If you don't see a command prompt, try pressing enter.
/ # wget --spider --timeout=1 nginx
Connecting to nginx (10.110.215.253:80)
wget: download timed out
/ #
```

Рисунок 3 – перевірка зв’язку після застосування мережевої політик із поду без потрібної мітки.

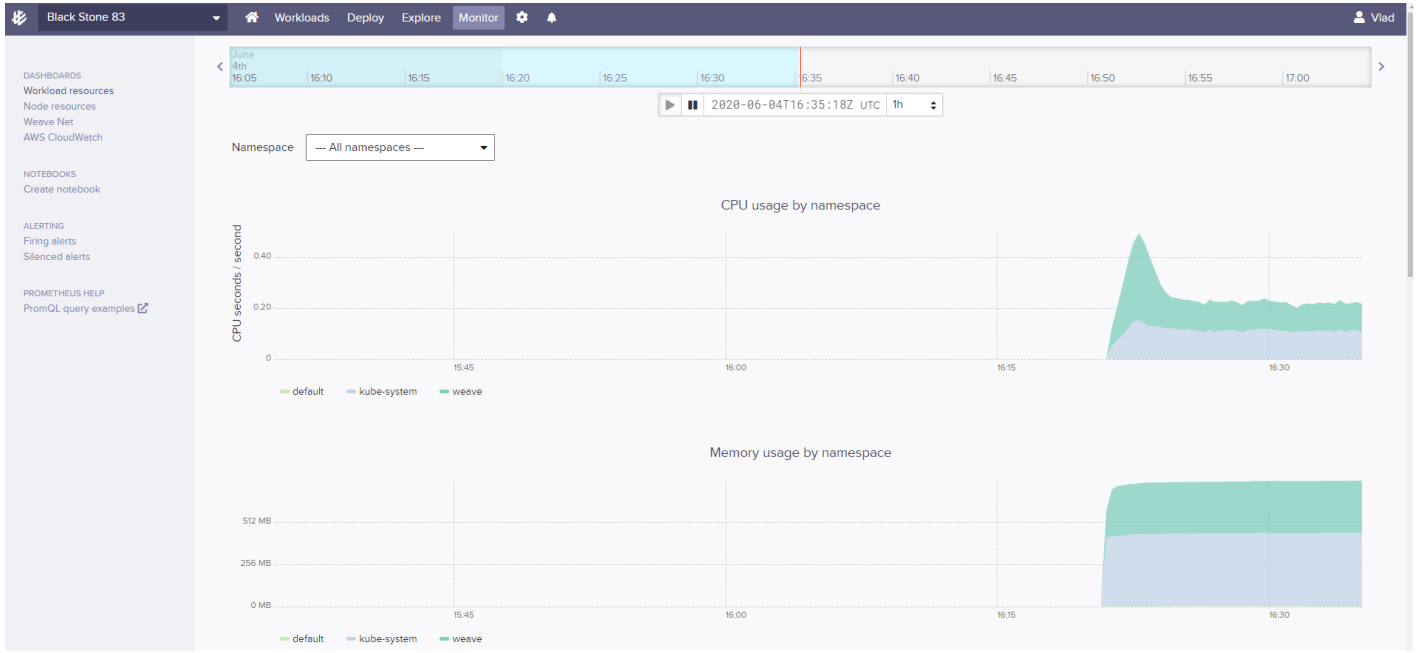
```
MacBook-Air-Dice:~ dice$ kubectl run busybox --rm -ti --labels="access=true" --image=busybox -- /bin/sh

If you don't see a command prompt, try pressing enter.
/ #
/ # wget --spider --timeout=1 nginx
Connecting to nginx (10.110.215.253:80)
remote file exists
/ #
```

Рисунок 4 – под із міткою «access=true» може зв’язатися із nginx.

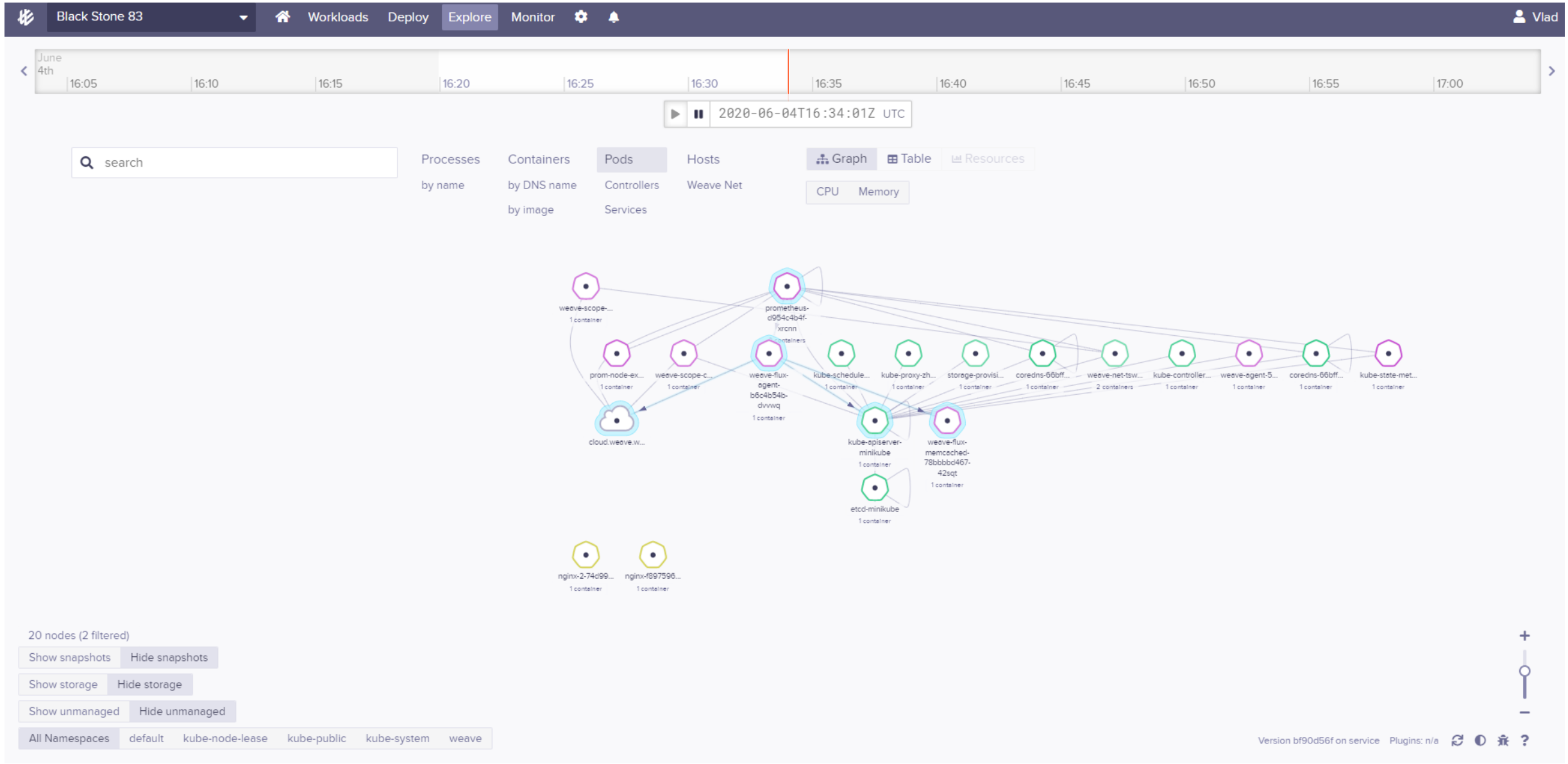
Підключення сервісу Weave Cloud до тестового кластеру



Вікно моніторингу стану навантаженості кластеру та вікно із хронологічними метриками стану кластеру



Структура тестового кластеру із Weave Net



Тестування кластеру із Cilium

Рисунок 1 – перевірка мережевих зв’язків у кластері. Колонка Ready вказує на результат тесту.

Рисунок 2 – збірка Hubble у файл маніфесту за допомогою Helm

```
MacBook-Air-Dice:~ dice$ kubectl get pods --all-namespaces
NAMESPACE     NAME                                                READY   STATUS    RESTARTS   AGE
default       echo-a-5f555bbc8b-46bq5                             1/1     Running   0          3m26s
default       echo-b-659766fb56-xcwck                             1/1     Running   0          3m26s
default       echo-b-host-65d7db76d8-656qm                        1/1     Running   0          3m25s
default       host-to-b-multi-node-clusterip-c7557d4f8-p2rmd      0/1     Pending   0          3m25s
default       host-to-b-multi-node-headless-5dfcdf9b76-895r4      0/1     Pending   0          3m25s
default       pod-to-a-6cf58894b7-ws47q                           1/1     Running   0          3m24s
default       pod-to-a-allowed-cnp-5898f7d8c9-7w5xj               1/1     Running   0          3m25s
default       pod-to-a-external-1111-5779fb7cb9-q9bjj             1/1     Running   0          3m23s
default       pod-to-a-l3-denied-cnp-74b9566cc7-rm4x2             0/1     Running   2          3m24s
default       pod-to-b-intra-node-77b485d996-dbt2r                1/1     Running   0          3m24s
default       pod-to-b-intra-node-nodeport-56975db6c7-rh5r4       0/1     Running   2          3m24s
default       pod-to-b-multi-node-clusterip-75f5c78f68-tp94j      0/1     Pending   0          3m24s
default       pod-to-b-multi-node-headless-5df88f9bd4-7t2nj       0/1     Pending   0          3m23s
default       pod-to-b-multi-node-nodeport-55b9769455-xtgph       0/1     Pending   0          3m23s
default       pod-to-external-fqdn-allow-google-cnp-74466b4c6f-87bw6   1/1     Running   0          3m23s
kube-system   cilium-operator-69f548c879-fzb8x                    1/1     Running   0          15m
kube-system   cilium-sk2nq                                        1/1     Running   0          15m
```

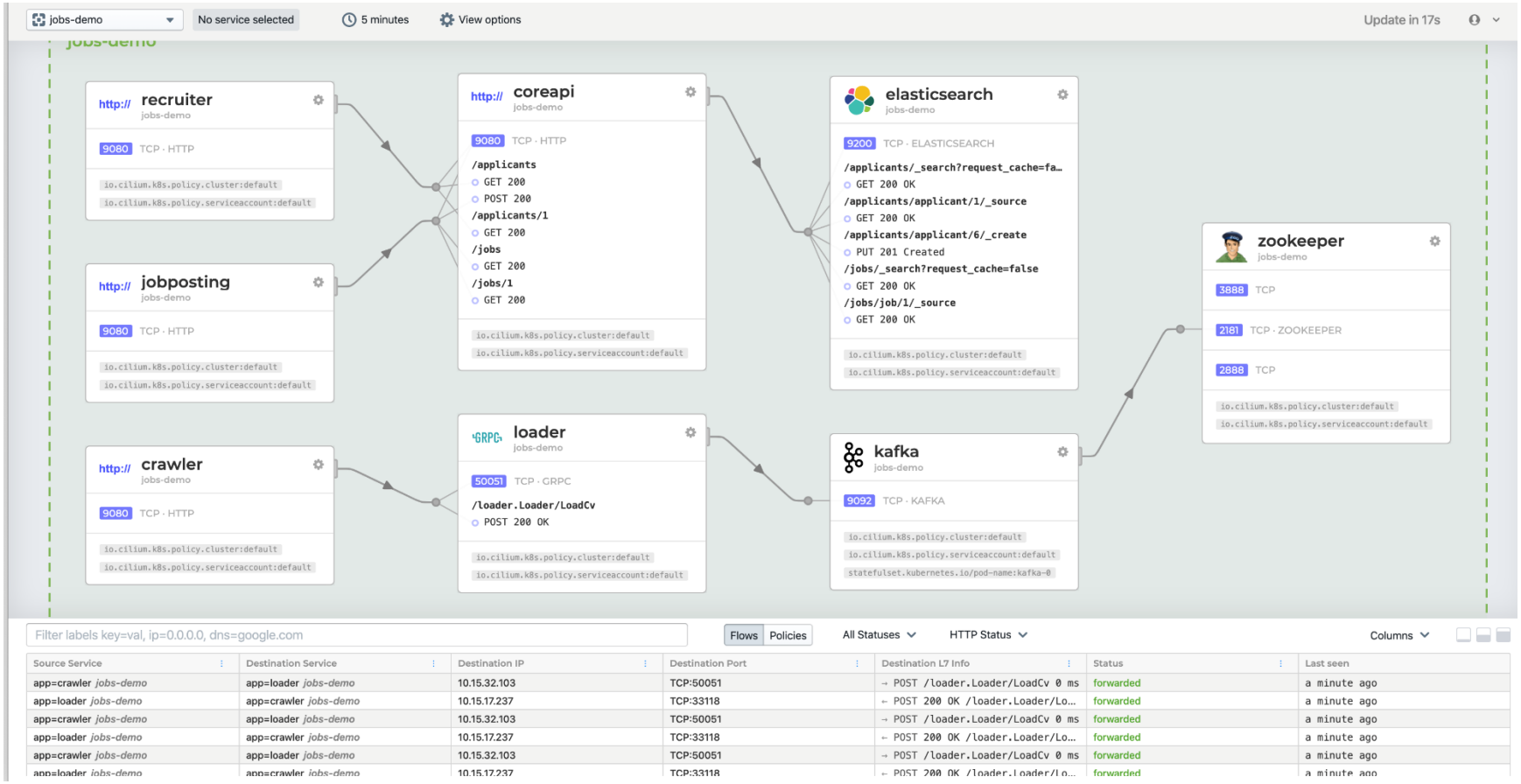
```
MacBook-Air-Dice:~ dice$ git clone https://github.com/cilium/hubble.git --branch v0.5
Клонирование в «hubble»…
remote: Enumerating objects: 7044, done.
remote: Total 7044 (delta 0), reused 0 (delta 0), pack-reused 7044
Получение объектов: 100% (7044/7044), 11.77 MiB | 2.32 MiB/s, готово.
Определение изменений: 100% (3345/3345), готово.
Распаковка файлов: 100% (2561/2561), готово.
MacBook-Air-Dice:~ dice$ helm template hubble \
>     --namespace kube-system \
>     --set metrics.enabled="{dns,drop,tcp,flow,port-distribution,icmp,http}" \
>     --set ui.enabled=true \
> hubble.yaml
```

Підключення Hubble та візуалізація тестового навантаження

```
MacBook-Air-Dice:kubernetes dice$ kubectl get pods --all-namespaces
NAMESPACE     NAME                              READY   STATUS    RESTARTS   AGE
jobs-demo     coreapi-6df4d98c75-b82l7          1/1     Running   1          4m
jobs-demo     crawler-787fcdd6cd-cbs98          1/1     Running   0          3m59s
jobs-demo     elasticsearch-66b87656d7-vdps5    1/1     Running   0          4m
jobs-demo     jobposting-6b97f5b774-hq97p       1/1     Running   0          4m1s
jobs-demo     kafka-0                           1/1     Running   1          4m
jobs-demo     loader-78cd676f95-4bfvj           1/1     Running   1          4m
jobs-demo     recruiter-86cc559f57-jdzzq        1/1     Running   0          4m1s
jobs-demo     zookeeper-bd9cfc487-jqdfm         1/1     Running   0          4m
```

Рисунок 1 – працюючи поди із тестовими сервісами у просторі імен jobs-demo.

Рисунок 2 – графічний інтерфейс Hubble з візуалізацією трафіку між тестовими подами



Використання Grafana для візуалізації метрик

Рисунок 1 – візуалізація загальних метрик кластеру за допомогою Grafana.

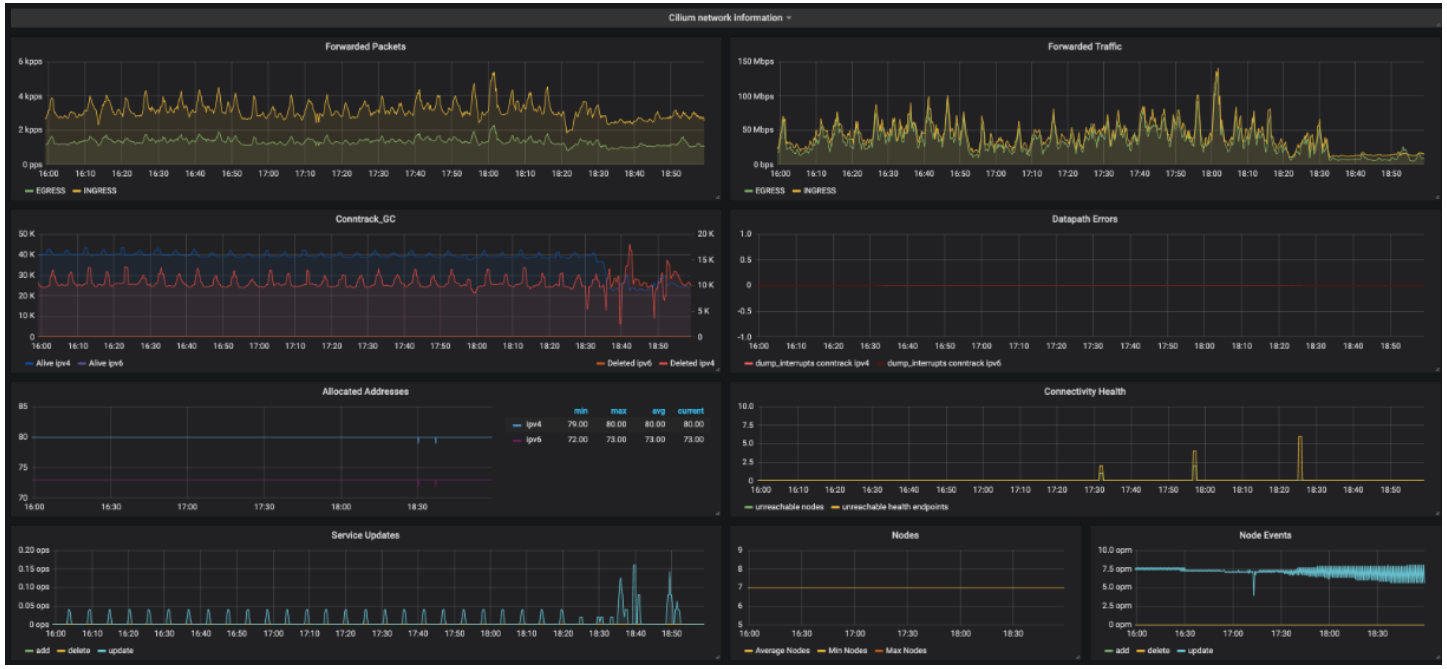


Рисунок 2 – візуалізація мережевих метрик кластеру за допомогою Grafana.

Порівняльна характеристика CNI плагінів

	Flannel	Calico	Weave Net	Cilium
Версія IP	IPv4	IPv4, IPv6	IPv4	IPv4, IPv6
Шифрування трафіку	-	-	NaCl	IPSec
Мережеві політики	-	Ingress, Egress	Ingress, Egress	Ingress, Egress
Перевірка мережевих політик	-	Платна	-	Так, через Cilium Hubble
Рекомендована max кількість вузлів	-	5000	500	5000+
Модель мережі за замовченням	Рівень 3 моделі OSI, VXLAN	Рівень 3 моделі OSI, BGP	Рівень 2 моделі OSI, VXLAN	Рівень 3 моделі OSI, BPF
Підтримка багатоадресної передачі	-	-	Так	-
Балансування навантаження	-	Так	Так	Так, може замінити kube-proxy
Системи моніторингу стану кластеру	-	-	Weave Cloud, платна	Hubble, Grafana та Prometheus

Висновки

- Проведено аналіз мережевої моделі Kubernetes;
- Проведено аналіз концепції та архітектури CNI;
- Теоретично і практично проведено аналіз чотирьох плагінів мережевої взаємодії Kubernetes у локальному тестовому середовищі;
- Створено порівняльну характеристику мережевих плагінів за визначеними критеріями;
- Було визначено умови найкращого застосування в залежності від вимог конкретного кластеру.

Дякую за увагу!