2019-01-25

オンプレミスKubernetesのネットワーク

株式会社Jストリーム　城田 雅水

もっと素敵な伝え方を。
Jstream

目次

もっと素敵な伝え方を。
Jstream

- 自己紹介
- Kubernetesについて
  - Kubernetesとは
  - オンプレミスにしたい理由
- ノード間のネットワーク (CNI)
- ノード外のネットワーク (外部ロードバランサー)

自己紹介

もっと素敵な伝え方を。
Jstream

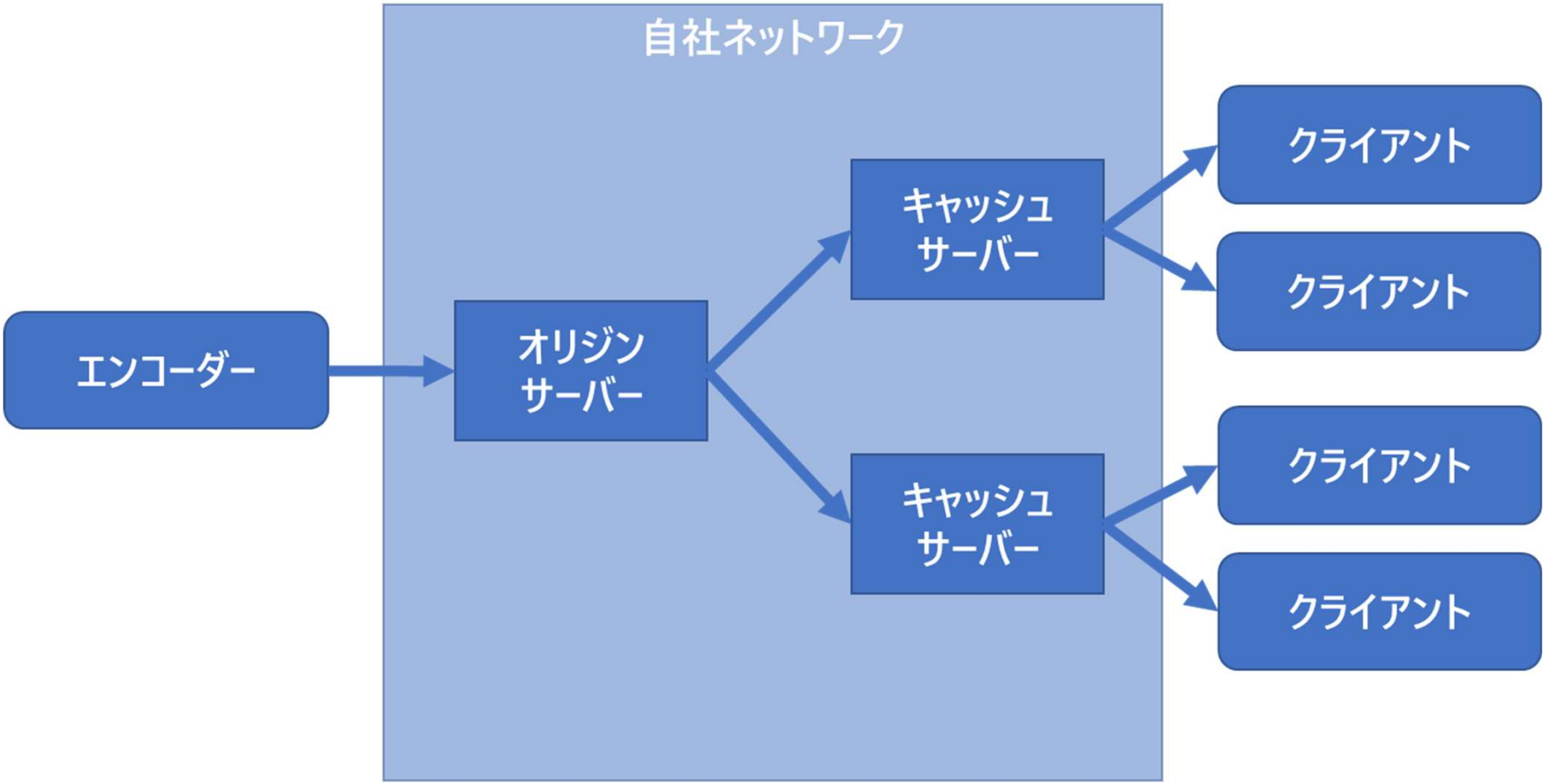
- 城田 雅水 (しろた まさみ)
- 高専では寮の情シス
- 新卒から今までJストリーム
  - 2013/04 – 2014/09: ライブ配信
  - 2014/10 – 2018/03: ネットワーク・サーバー運用
  - 2018/04 - : 開発 (ruby, go)

会社について

もっと素敵な伝え方を。
Jstream

- 動画に関するいろいろなこと
  - CDNによる配信
    - 自社設備だけでなくマルチCDNも
  - ライブストリーミング
    - この中継もやってます！
  - 動画配信プラットフォーム
    - ブラウザで配信する動画を管理
  - 映像制作

CDNの仕組み



もっと素敵な伝え方を。
Jstream

Kubernetesについて

Kubernetesとは

もっと素敵な伝え方を。
Jstream

- コンテナを便利に扱うためのシステム
- Dockerホストをノードとしてクラスタリング
  - コンテナを適当に分散配置
  - 死んだノードにいたコンテナを立ち上げ直す
- リソースという名前でいろいろ抽象化

Kubernetesのリソース

もっと素敵な伝え方を。
Jstream

- Pod (Pod of whales, クジラの群れ)
  - コンテナの集まり (1〜数個)
  - この単位でIPアドレスがつく
- Service
  - Podを束ねるロードバランサー
  - 内部用のVIPとFQDNがつく

Kubernetesを使う動機

もっと素敵な伝え方を。 Jstream

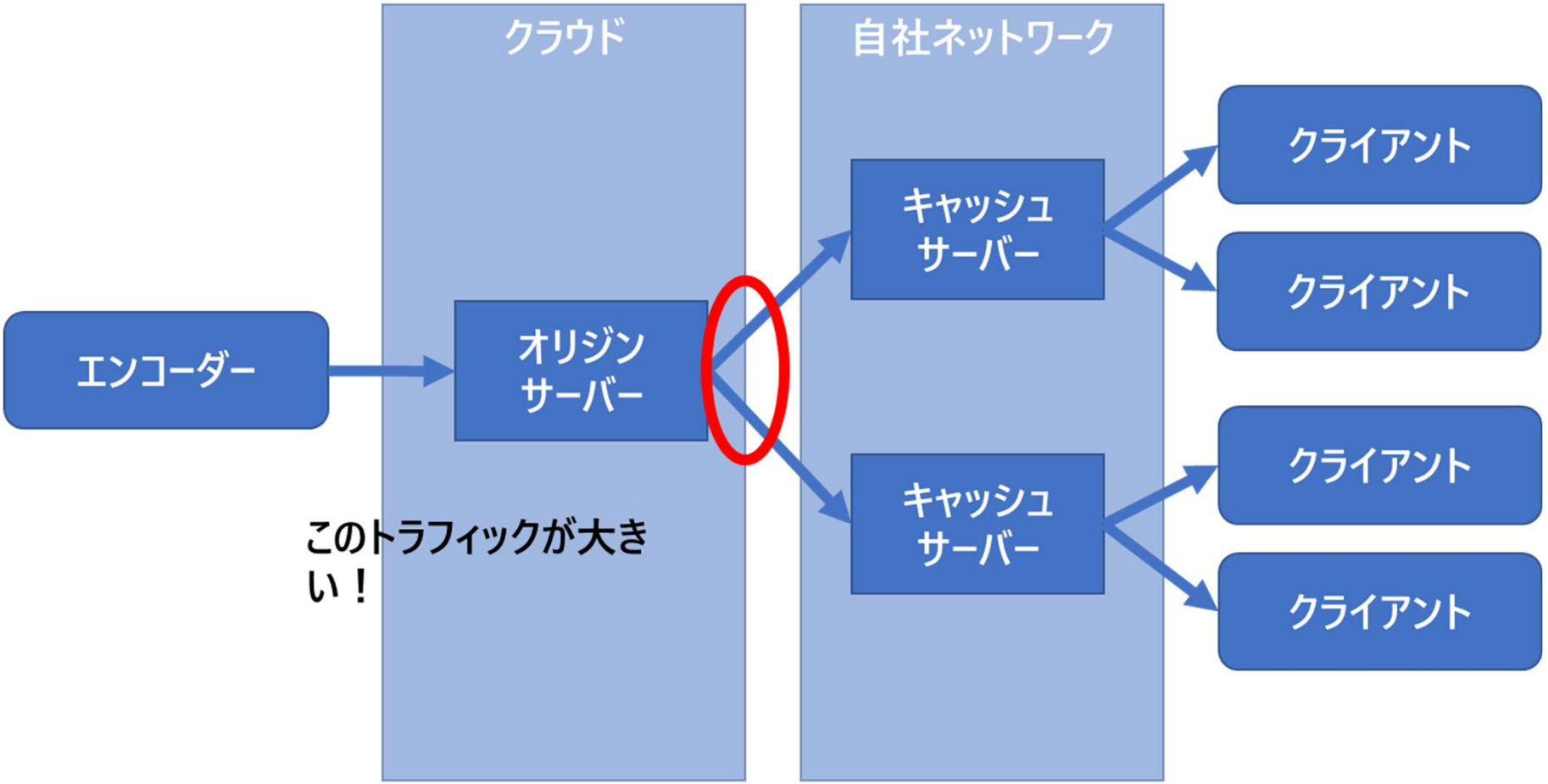
- Podのデプロイが簡単
  - サーバーを増やすのは面倒
    - IPアドレスの空きを確認
    - 仮想ならホストリソースの空きを確認
    - ロードバランサーを設定
    - これらの自動化も大変
  - これらを全部やってくれる
  - アプリケーションからコンテナをデプロイすることも

オンプレミスにしたい理由

もっと素敵な伝え方を。
Jstream

- トラフィックが高い！
- コンテナをオリジンサーバーとして使うとしても、結構な量が流れる。
- CDNを使えない非HTTPなプロトコルも

オンプレミスにしたい理由



オンプレミスにしたい理由

もっと素敵な伝え方を。 Jstream

- トラフィックが高い！

- 月平均:1.5Gbpsのとき
  - 30日で総量は452,622GiB
  - Googleだと4,042,282円
- 95%ile:2Gbpsのとき
  - 400円/Mbpsなら800,000円

ノード間のネットワーク

Container Network Interface

ノード間のネットワーク

もっと素敵な伝え方を。
Jstream

- ノード同士を結ぶネットワークが必要
  - Pod同士の通信
  - Service(ノード)から他ノードPodへ転送
- Kubernetes本体ではネットワークを提供しない

ノード間のネットワーク

もっと素敵な伝え方を。 Jstream

- Container Network Interface
  - Podに仮想NICとIPアドレスを提供
  - Kubernetes以外でも使われている

- Flannel
- Project Calico
- kube-router

Flannel

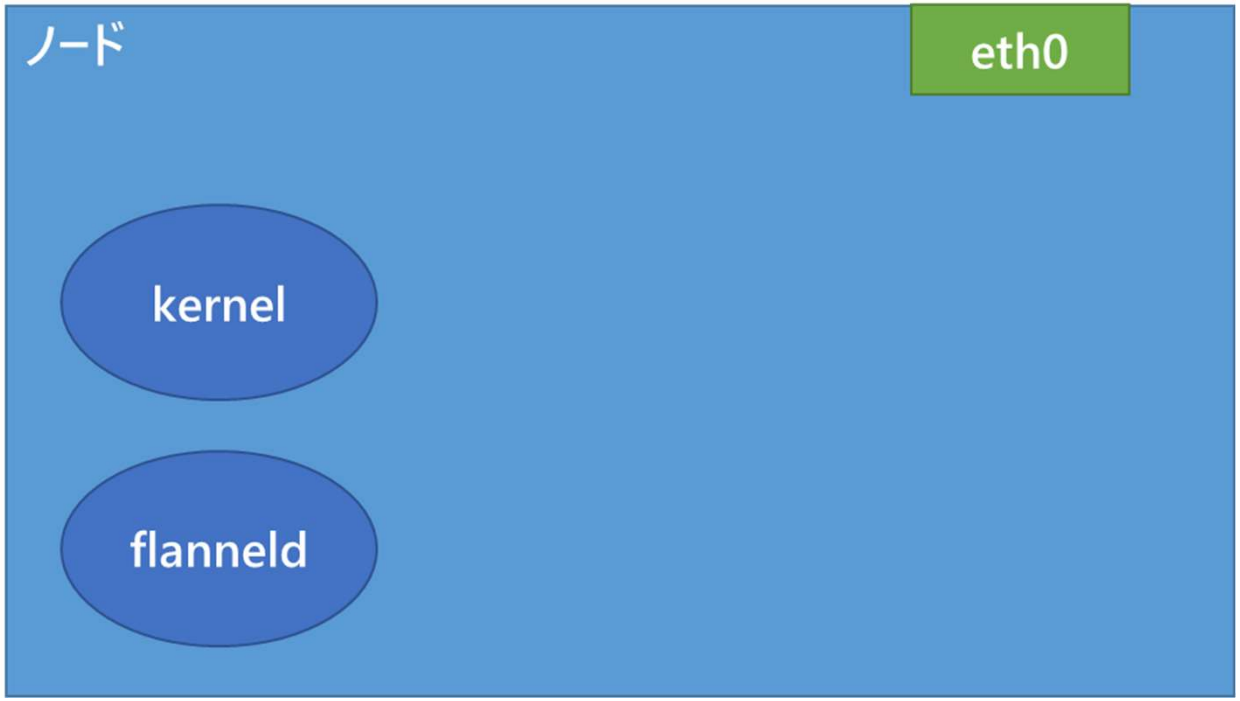
Flannel

もっと素敵な伝え方を。
Jstream

- ユニキャストなVXLANを構築
  - ノード間のルーティングに使うだけ
  - 全Podが1つのブリッジにのるわけではない
- 日本語の記事が多い
  - 4年前から最近まで

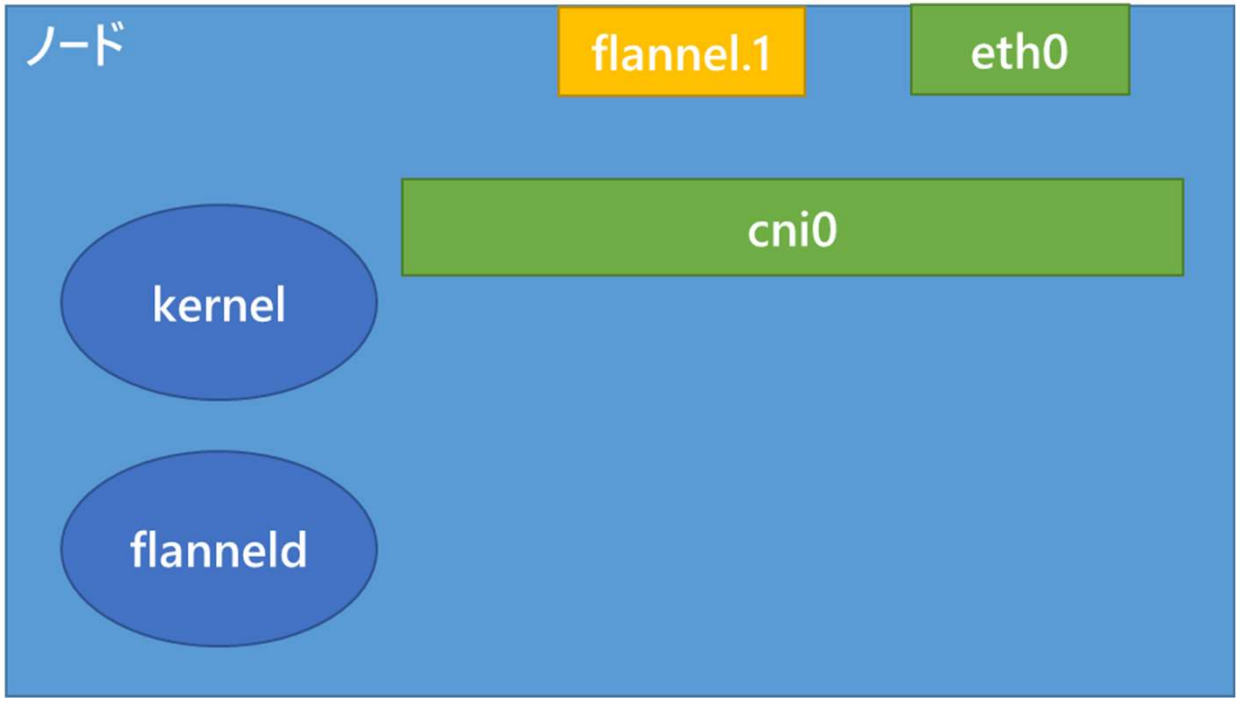
Flannelの仕組み

- 各ノードにflanneldプロセスが常駐



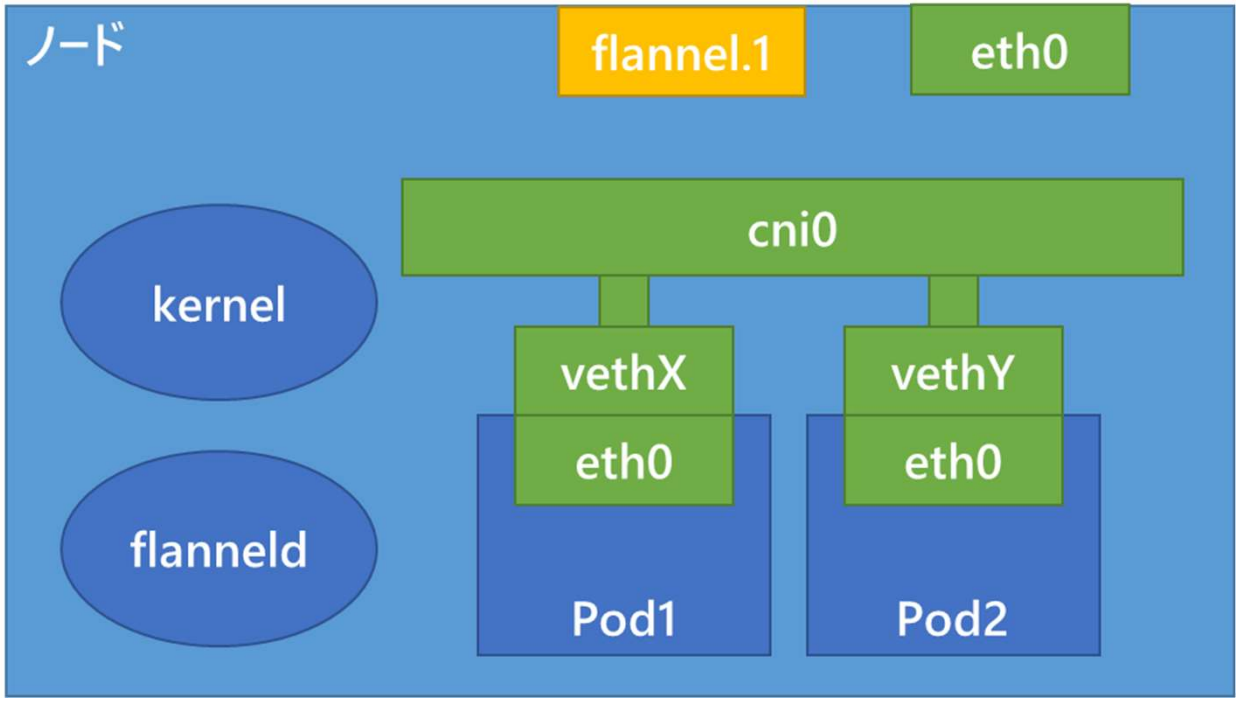
Flannelの仕組み

- インターフェース作成



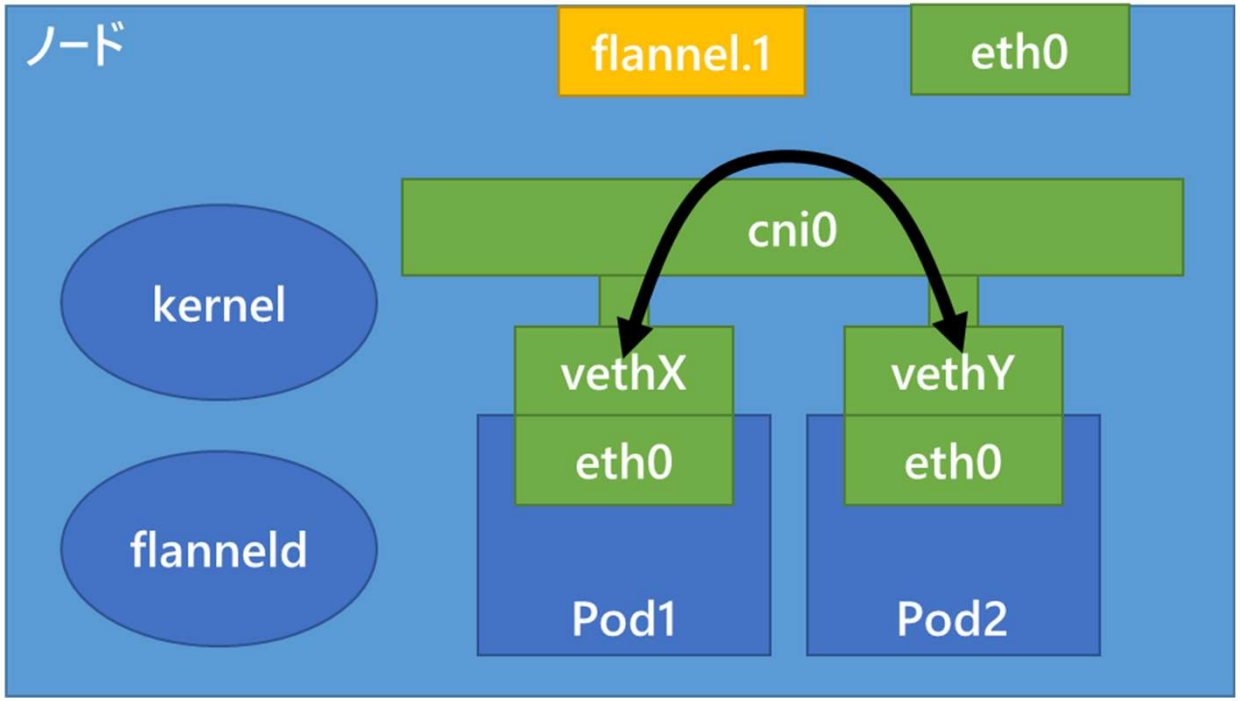
Flannelの仕組み

- Podはcni0にぶらさがる。



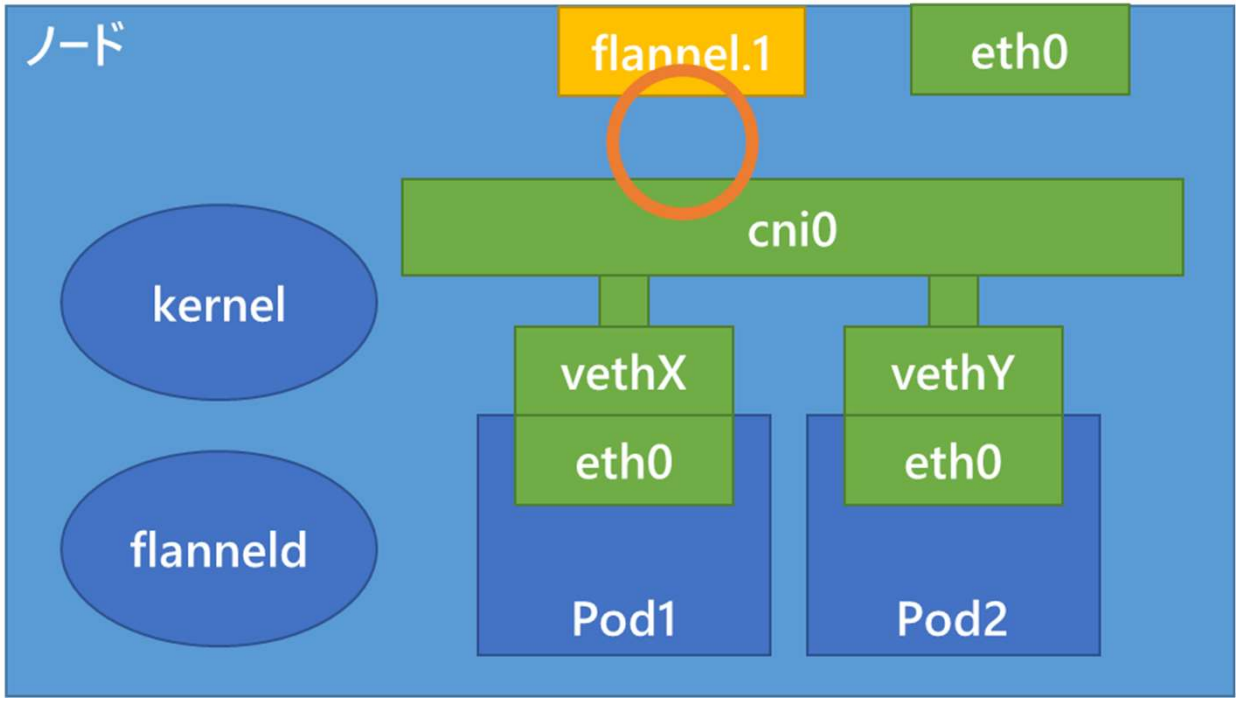
Flannelの仕組み

- Pod同士の通信はブリッジを使う



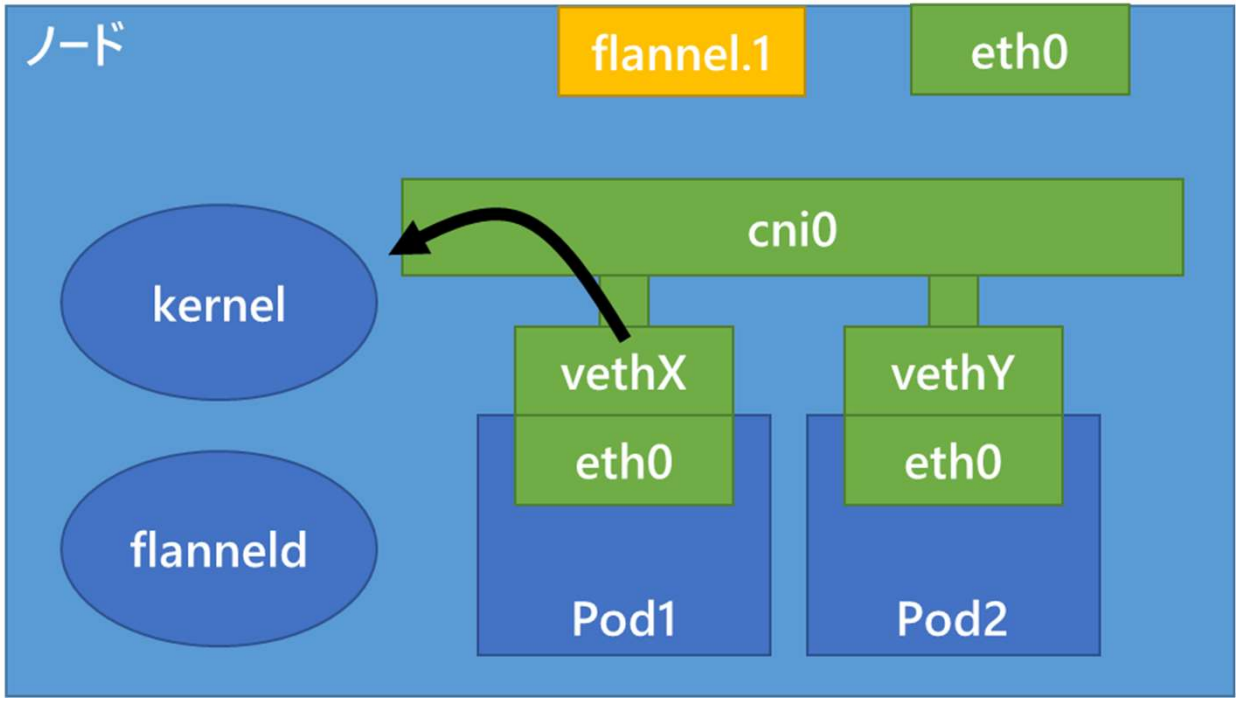
Flannelの仕組み

- 他ノードへの通信はL2延伸ではない



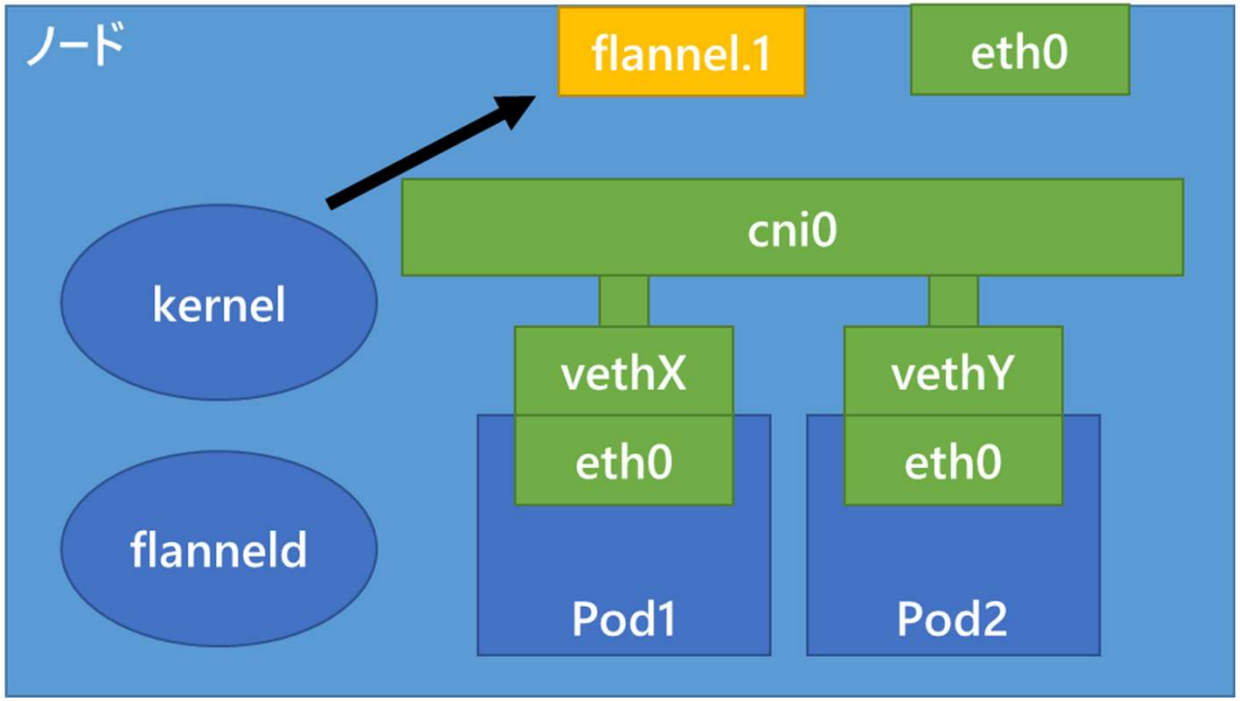
Flannelの仕組み

- まずはPodからノードへルーティング



Flannelの仕組み

- flannel.1から他ノードへ



Flannelの仕組み

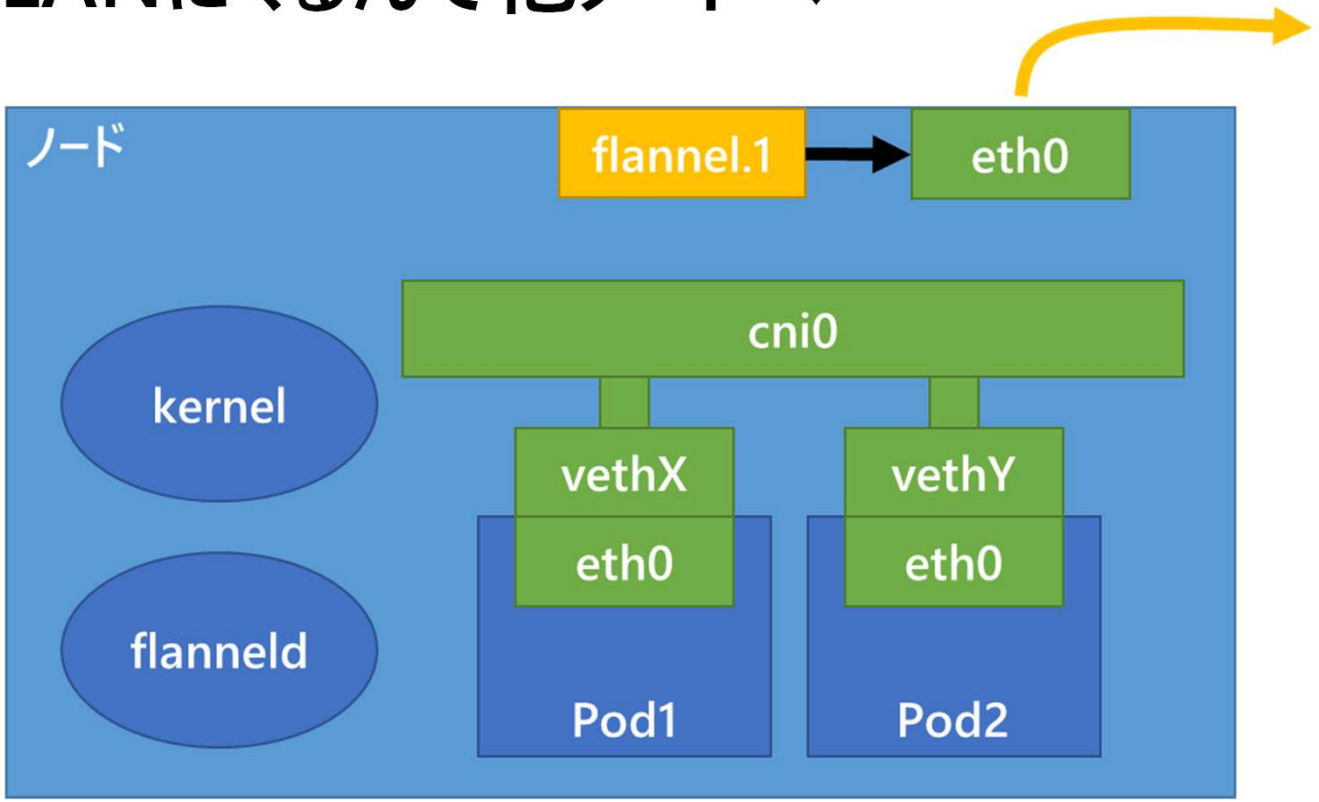
もっと素敵な伝え方を。 Jstream

- Flanneldが転送に必要な情報を教える
  - 宛先ノードの(flannel.1)MACアドレス
  - 宛先ノードの(eth0)IPアドレス

```
$ kubectl describe node master01.example.com
...
Annotations:        flannel.alpha.coreos.com/backend-data={"VtepMAC":"0a:29:40:c3:ca:32"}
                    flannel.alpha.coreos.com/backend-type=vxlan
                    flannel.alpha.coreos.com/kube-subnet-manager=true
                    flannel.alpha.coreos.com/public-ip=192.168.1.101
                    node.alpha.kubernetes.io/ttl=0
                    volumes.kubernetes.io/controller-managed-attach-detach=true
...
```

Flannelの仕組み

- VXLANにくるんで他ノードへ



Flannelの特徴

もっと素敵な伝え方を。 Jstream

- ノード同士がUDPで通信できればクラスタリングできる
- VXLAN以外も扱える
  - ipip, ipsec (experimental)
  - 全ノードが同一セグメントにいればトンネルなしも可能 (host-gw)

Flannelの特徴

- 更新が鈍ってきた？
  - 2017/1/11 : 0.7.0
  - 2017/4/19 : 0.7.1
  - 2017/7/12 : 0.8.0
  - 2017/9/23 : 0.9.0
  - 2017/11/17 : 0.9.1
  - 2018/1/24 : 0.10.0
  - 2019/1/25 : 1年リリースなし

Project Calico

Project Calico

もっと素敵な伝え方を。 Jstream

- すべてL3で疎通させる
- NetworkPolicyも実装
  - Podに対してファイアウォールを設定できる
- Calico = 三毛猫

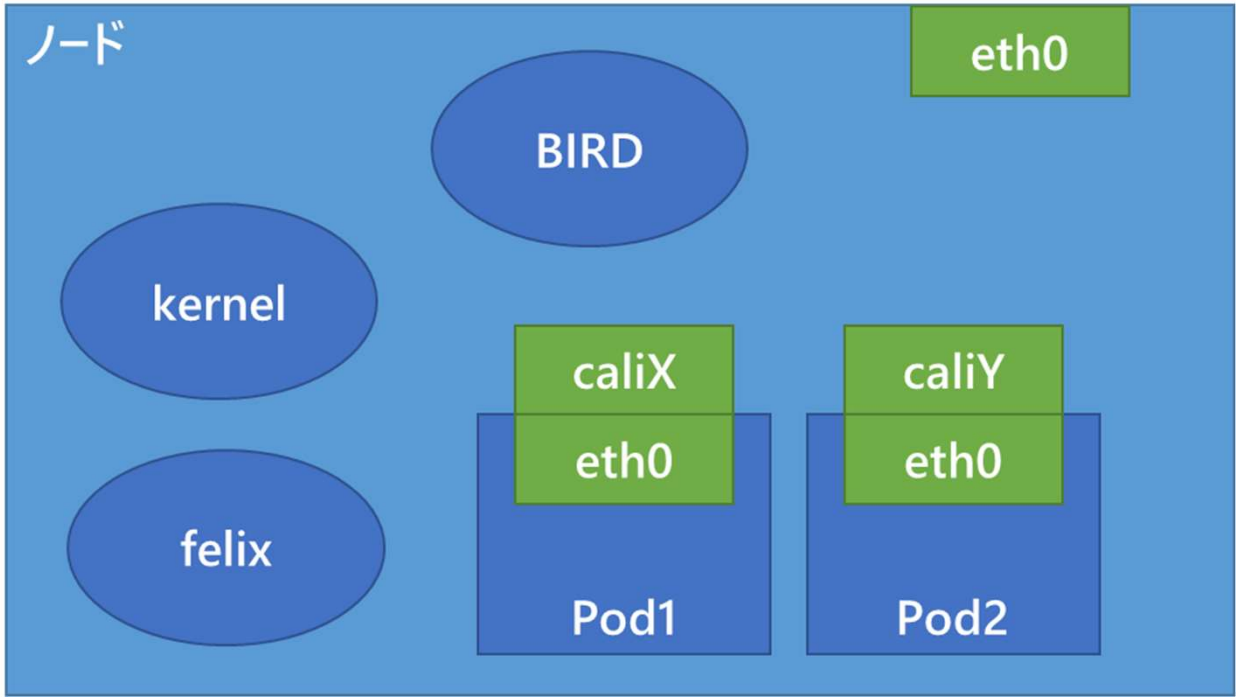
Project Calicoの仕組み

もっと素敵な伝え方を。 Jstream

- いっぱいプロセスが上がる
  - etcd (設定保持)
  - felix (インターフェースの管理など)
  - BIRD (BGPの処理)
  - confd (etcdからBIRDのコンフィグを生成)

Project Calicoの仕組み

- ブリッジは作成しない



Project Calicoの仕組み

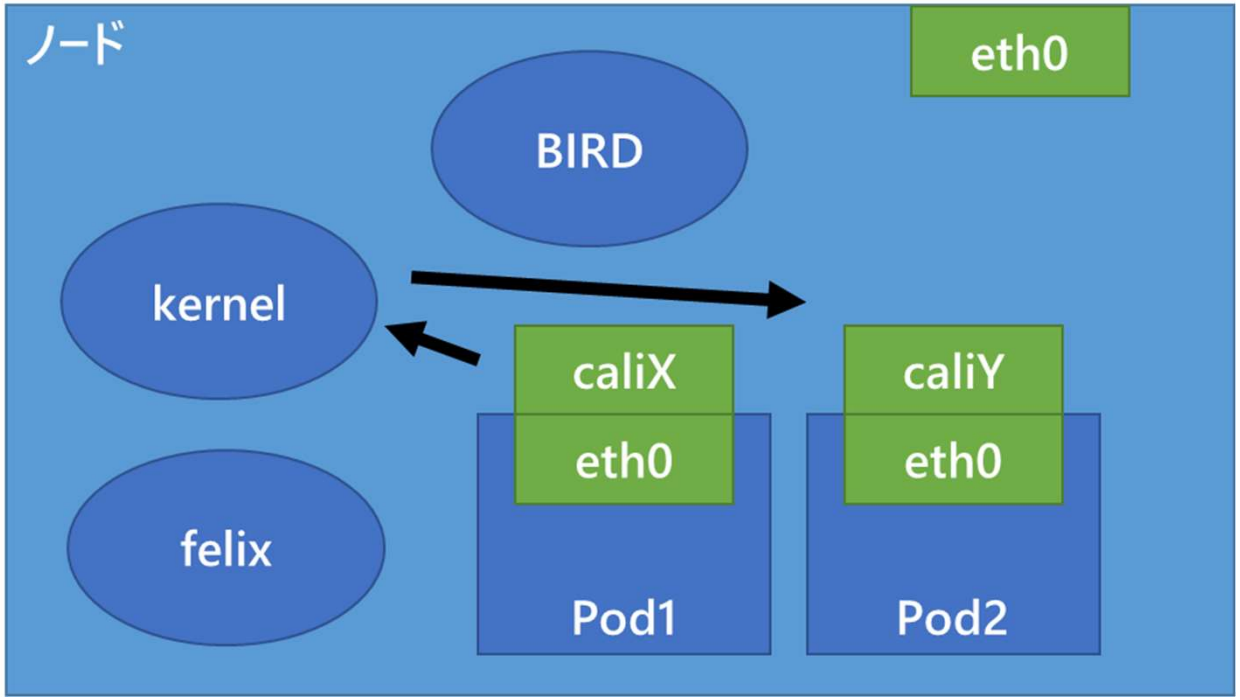
もっと素敵な伝え方を。 Jstream

- BIRDでノード同時をBGPピアリング
  - デフォルトはフルメッシュ
- 各ノードで使うPod用のサブネットを広報

Project Calicoの仕組み

もっと素敵な伝え方を。 Jstream

- Pod間の通信はノードがルーティングする



Project Calicoの仕組み

もっと素敵な伝え方を。 Jstream

- Pod間の通信はノードがルーティングする

```
$ ip route show | grep cali
10.244.140.64 dev cali1419e80a359 scope link
10.244.140.65 dev cali54ab31c27c8 scope link
10.244.140.66 dev cali3616cad90a5 scope link
10.244.140.67 dev cali2c4419dca3e scope link
10.244.140.68 dev calif01c338f54e scope link
...
```

Project Calicoの仕組み

もっと素敵な伝え方を。 Jstream

- ノード間はBGPで得た経路でルーティング
- ノード間にルーターがある場合、そのルーターとともBGPピアリングすることで疎通させる。

```
$ ip route show proto bird
blackhole 10.244.115.192/26
10.244.140.64/26 via 192.168.1.242 dev ens192
10.244.149.64/26 via 192.168.1.241 dev ens192
```

Project Calicoの特徴

もっと素敵な伝え方を。 Jstream

- 設定にcalicoctlというコマンドを使う
  - kubectl風の操作感
  - 設定内容はYAMLで表現

```
$ cat << EOF | calicoctl create -f -
apiVersion: v1
kind: bgpPeer
metadata:
  peerIP: 192.168.1.1
  scope: global
spec:
  asNumber: 65001
EOF
```

Project Calicoの特徴

もっと素敵な伝え方を。
Jstream

- IPIPトンネリングも対応
  - 最近はデフォルトでトンネルに
  - 別ネットワークのノード宛てのみIPIPとすることも

もっと素敵な伝え方を。
Jstream

kube-router

kube-router

もっと素敵な伝え方を。
Jstream

- 1つのgoバイナリでKubernetesのネットワークまわりを全部処理する
- NetworkPolicyの実装に加えてServiceの実装置き換えもできる

kube-routerの仕組み

もっと素敵な伝え方を。
Jstream

- ノード内はFlannelのようにブリッジ
- ノード間はBGPを使ったルーティング
  - GoBGPのライブラリを利用

kube-routerの特徴

- BGPの設定はノードのアノテーションとして Kubernetesクラスタに書いていく
  - Calico用etcdのように冗長性を考えなくていい

```
$ kubectl describe node master01.example.com
...
Annotations:        kube-router.io/peer.asns=65001
                    kube-router.io/peer.ips=192.168.1.1
                    kube-router.io/rr.server=42
                    kubeadm.alpha.kubernetes.io/cri-socket=/var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl=0
                    volumes.kubernetes.io/controller-managed-attach-detach=true
...
```

kube-routerの特徴

もっと素敵な伝え方を。
Jstream

- クラスタ内からルートリフレクタを選出できる

ノード外のネットワーク

外部ロードバランサーとの連携

ノード外のネットワーク

もっと素敵な伝え方を。
Jstream

- Kubernetesクラスタ外からPodへはどうやってアクセスする？
  - PodもServiceもクラスタ内でしか通用しない
- クラウドにはLoadBalancerタイプのServiceがある
  - クラウドのロードバランサーをServiceに自動連携

ノード外のネットワーク

もっと素敵な伝え方を。
Jstream

- NodePortタイプのService
  - いわゆるポートフォワード
  - 全ノードが対象ポートをリッスン
  - ポート443をリッスンできるServiceは1クラスタにつき1つだけ

ノード外のネットワーク

もっと素敵な伝え方を。
Jstream

- Ingress
  - HTTPロードバランサー
  - VirtualHost/SNIで複数サイトに対応
  - 実装はnginxがメイン
  - NodePortタイプのServiceに加えて使う
  - 非HTTPだと結局ポートの取り合いに

ノード外のネットワーク

もっと素敵な伝え方を。
Jstream

- Kubernetesと連携するロードバランサーを用意しないといけない

- MetalLB
- F5 Container Connector

MetalLB

MetalLB

もっと素敵な伝え方を。
Jstream

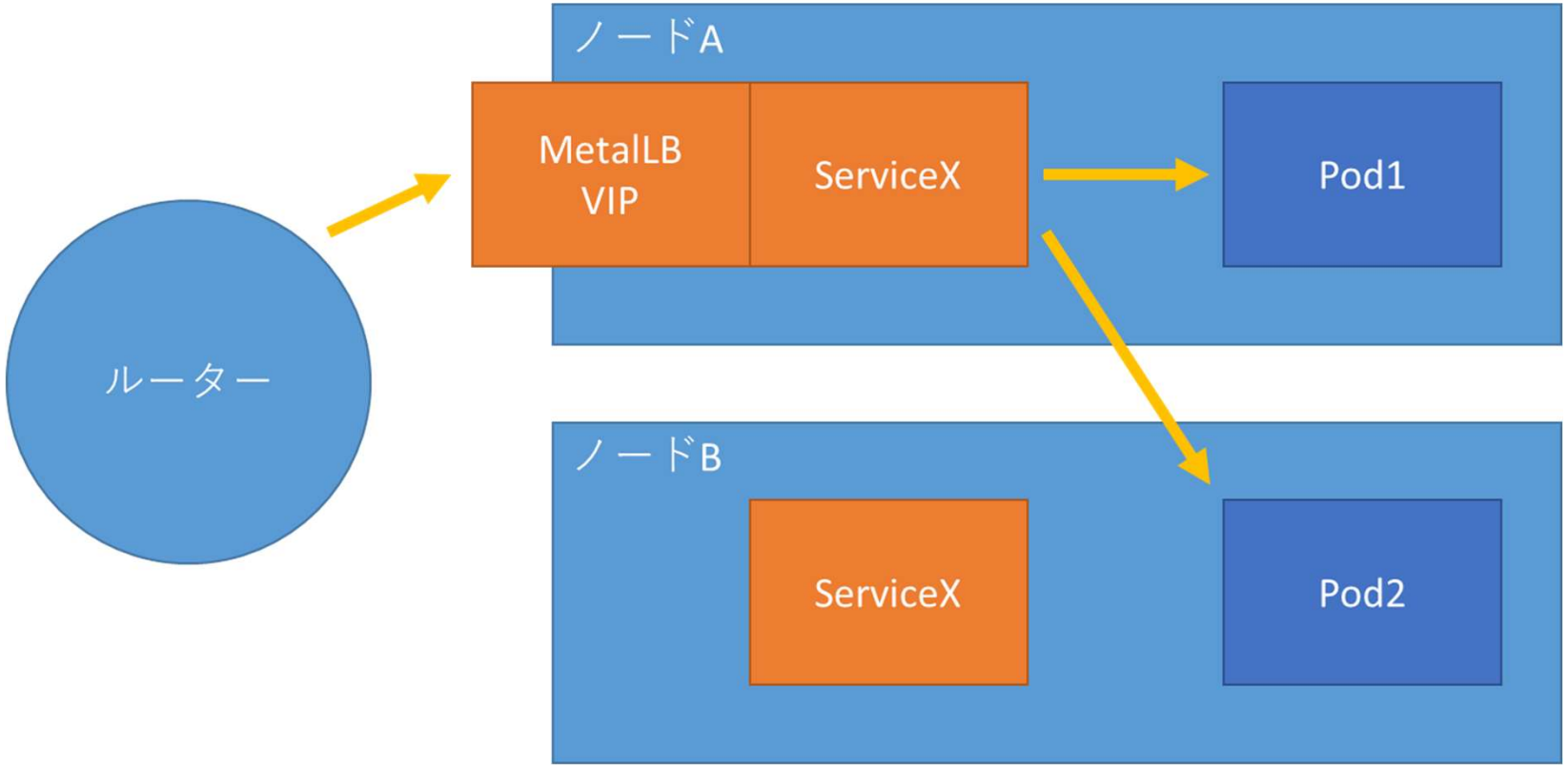
- Kubernetesノードで動くロードバランサー
- 外部からの接続方法によって2つモードがある
- 公開に使うIPアドレス帯を与えると適宜払い出してくれる。

MetalLBの仕組み

もっと素敵な伝え方を。
Jstream

- Layer2モード
- Proxy ARPで1台のノードが代表して受ける
- 代表ノードからPodへロードバランス

MetalLBの仕組み

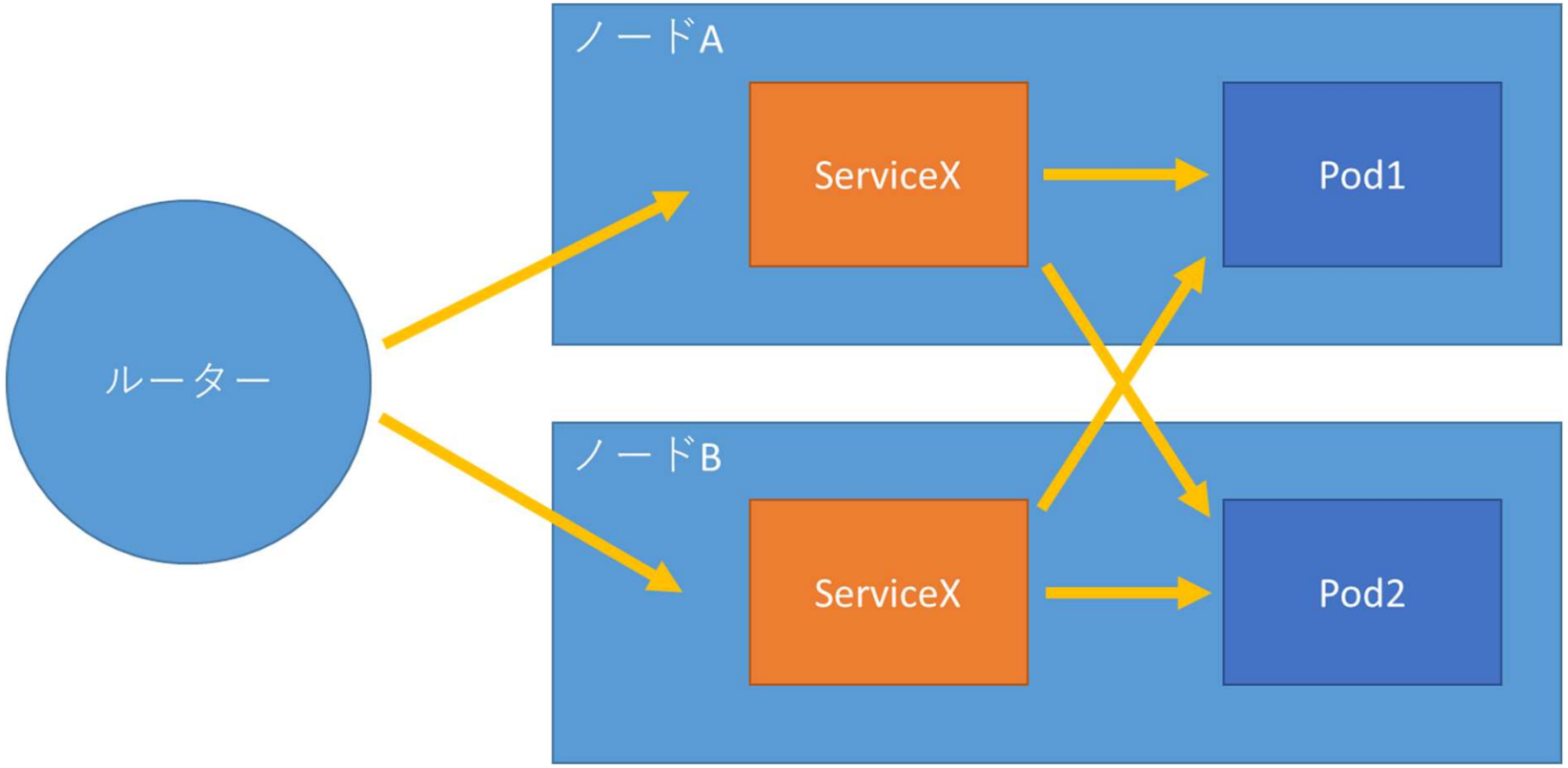


MetalLBの仕組み

もっと素敵な伝え方を。
Jstream

- BGPモード
- ルーターに各ノードがホストルートを広報
- ルーターがECMPでロードバランス

MetalLBの仕組み

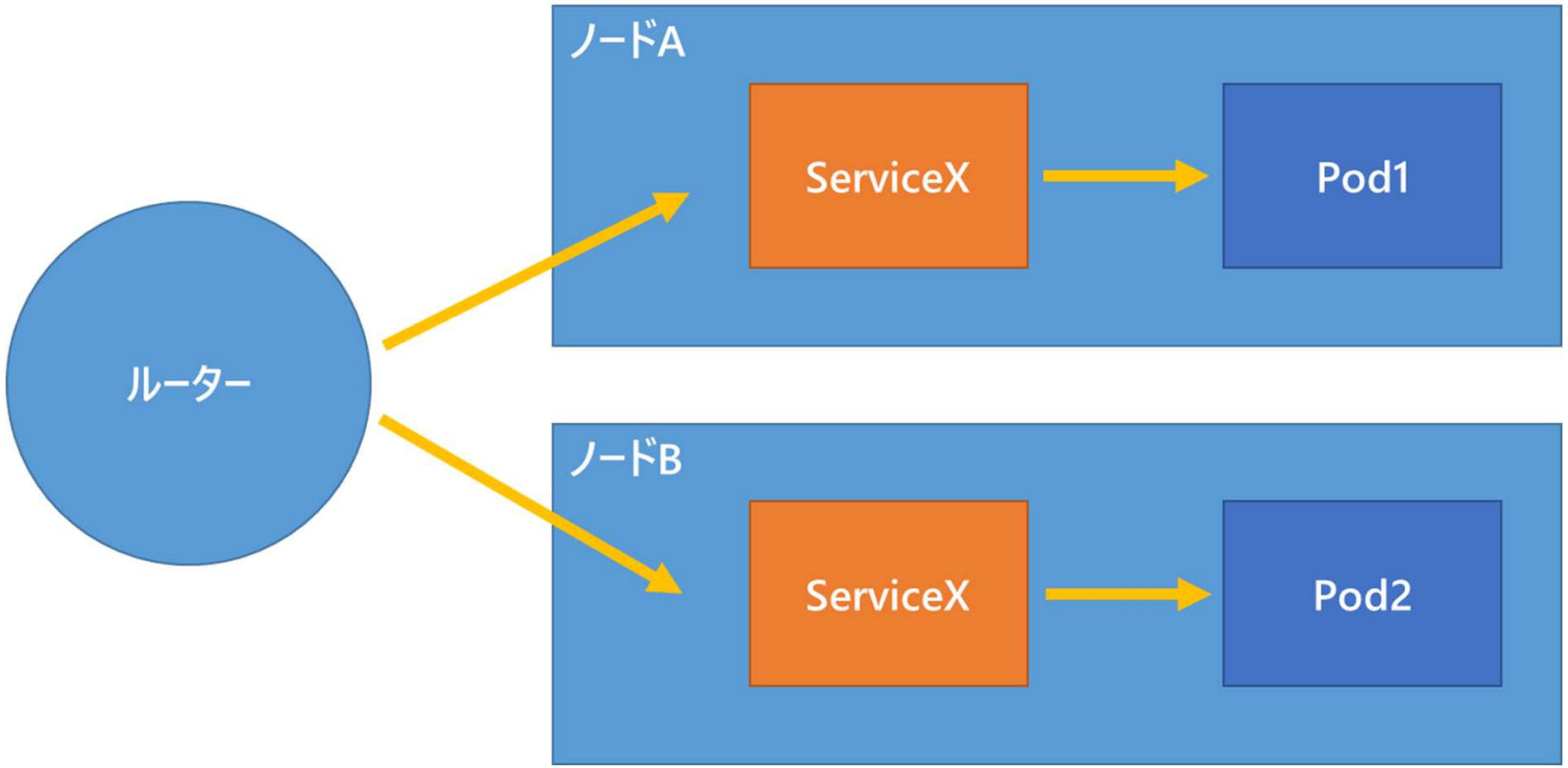


MetalLBの仕組み

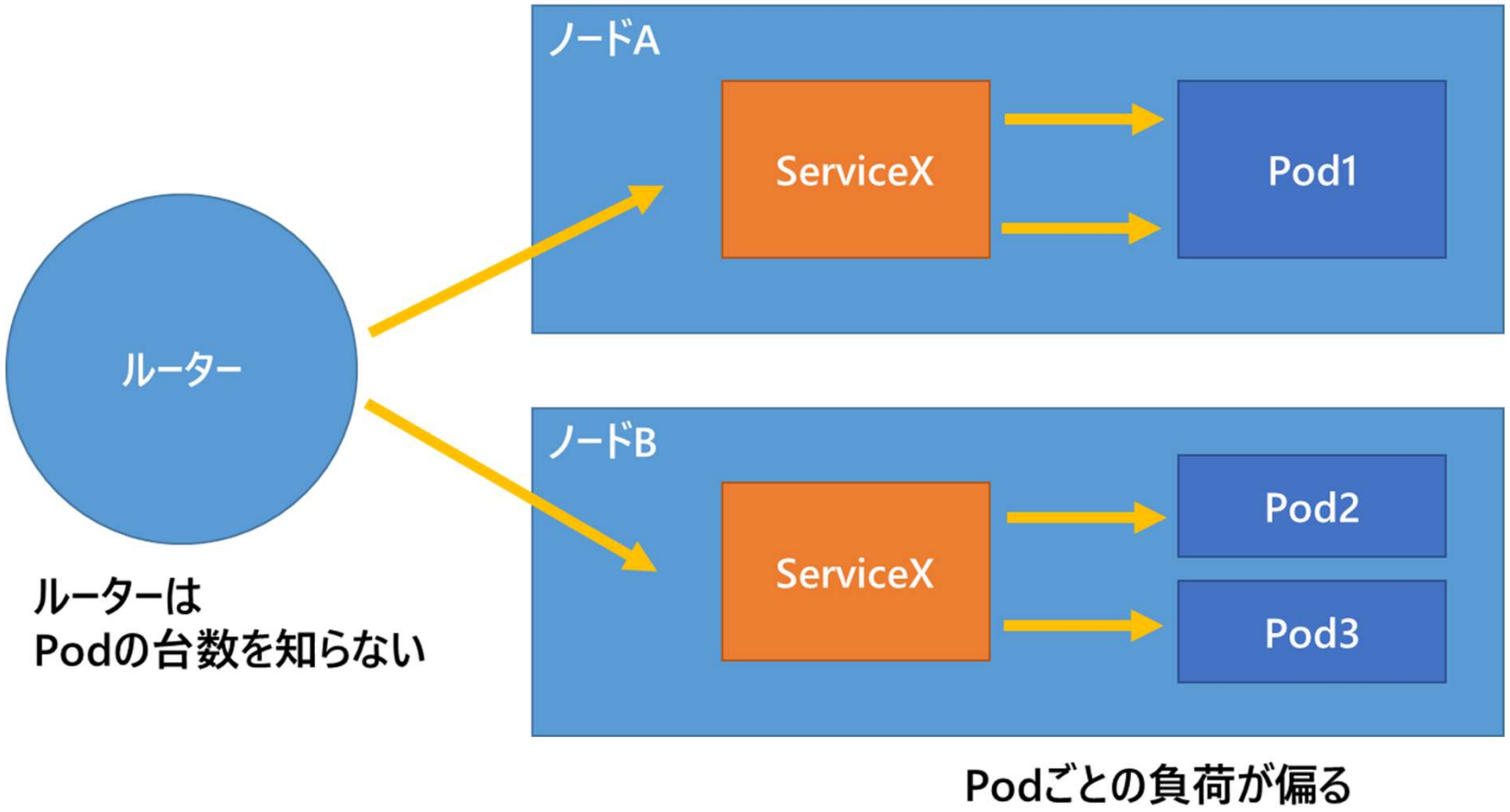
もっと素敵な伝え方を。
Jstream

- ルーターとServiceで二重にロードバランス
- 設定でやめさせることができるが

MetalLBの仕組み



MetalLBの仕組み



MetalLBの特徴

もっと素敵な伝え方を。 Jstream

- Layer2モードは手軽に使える
  - PC内でVM-ホスト間を繋ぐ時など
  - トラフィックが1つのノードに集まってしまう
- BGPを使うCNIと競合することがある
  - 重複してBGPピアを確立できない
    - ルーターを複数用意？
    - VRFで分ける？

F5 Container Connector

F5 Container Connector

もっと素敵な伝え方を。
Jstream

- KubernetesからBIG-IPを自動設定する
- 既存サーバー群といっしょに扱える

F5 Container Connectorの仕組み

もっと素敵な伝え方を。
Jstream

- Kubernetes APIを叩いて情報収集
  - ノード
  - Pod
  - Service
  - VIPの設定

F5 Container Connectorの仕組み

もっと素敵な伝え方を。 Jstream

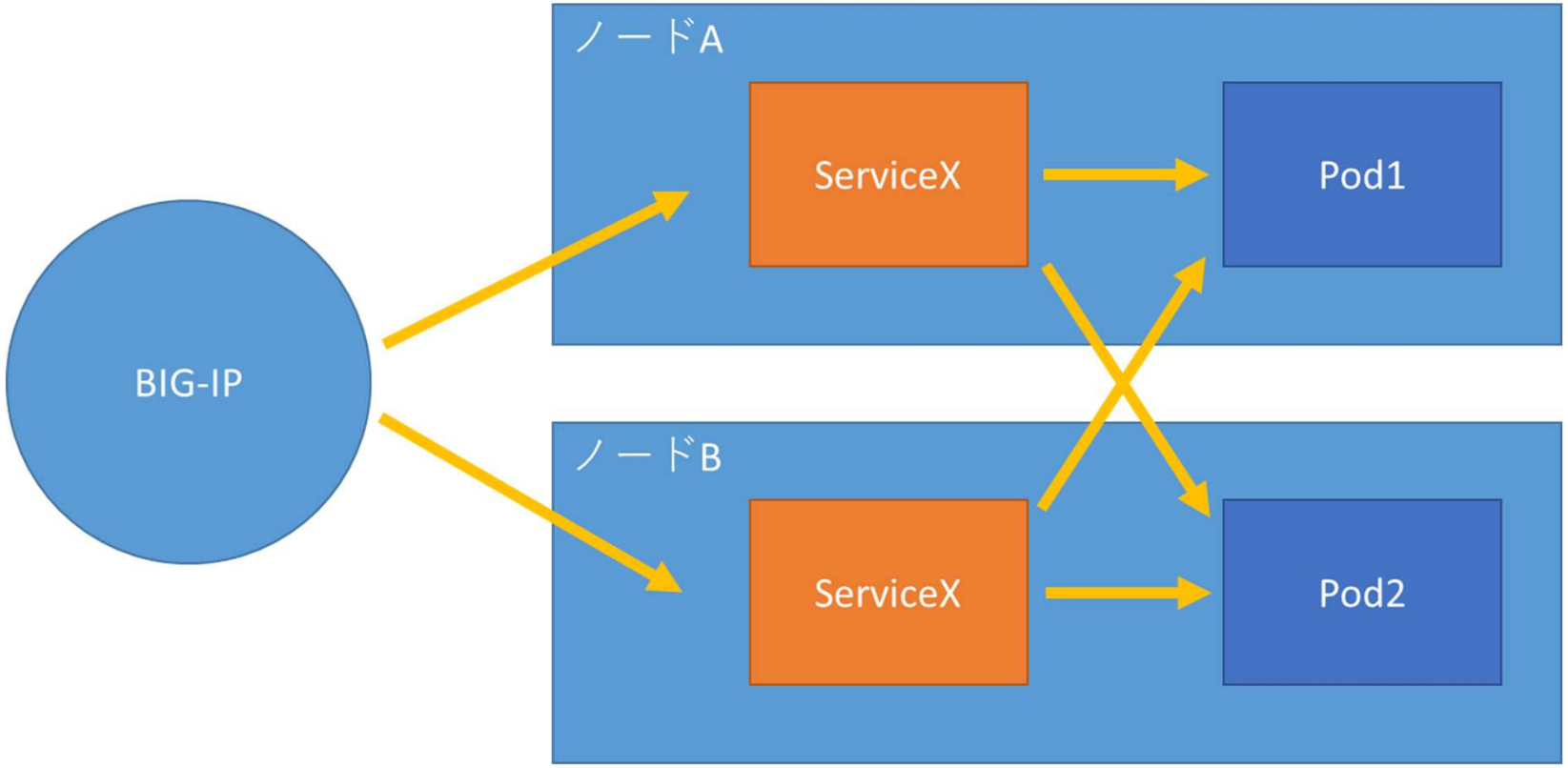
- BIG-IPのAPIを叩いて転送ルールを書き込み
- Podの増減があればBIG-IPの設定をアップデート
- BIG-IPからKubernetesへの転送方法が2種類ある

F5 Container Connectorの仕組み

もっと素敵な伝え方を。 Jstream

- NodePortモード
  - NodePortタイプのServiceに向けてBIG-IPが転送する
  - F5 CCの設定だけで動く
  - BIG-IPとノードで2回ロードバランスする

F5 Container Connectorの仕組み

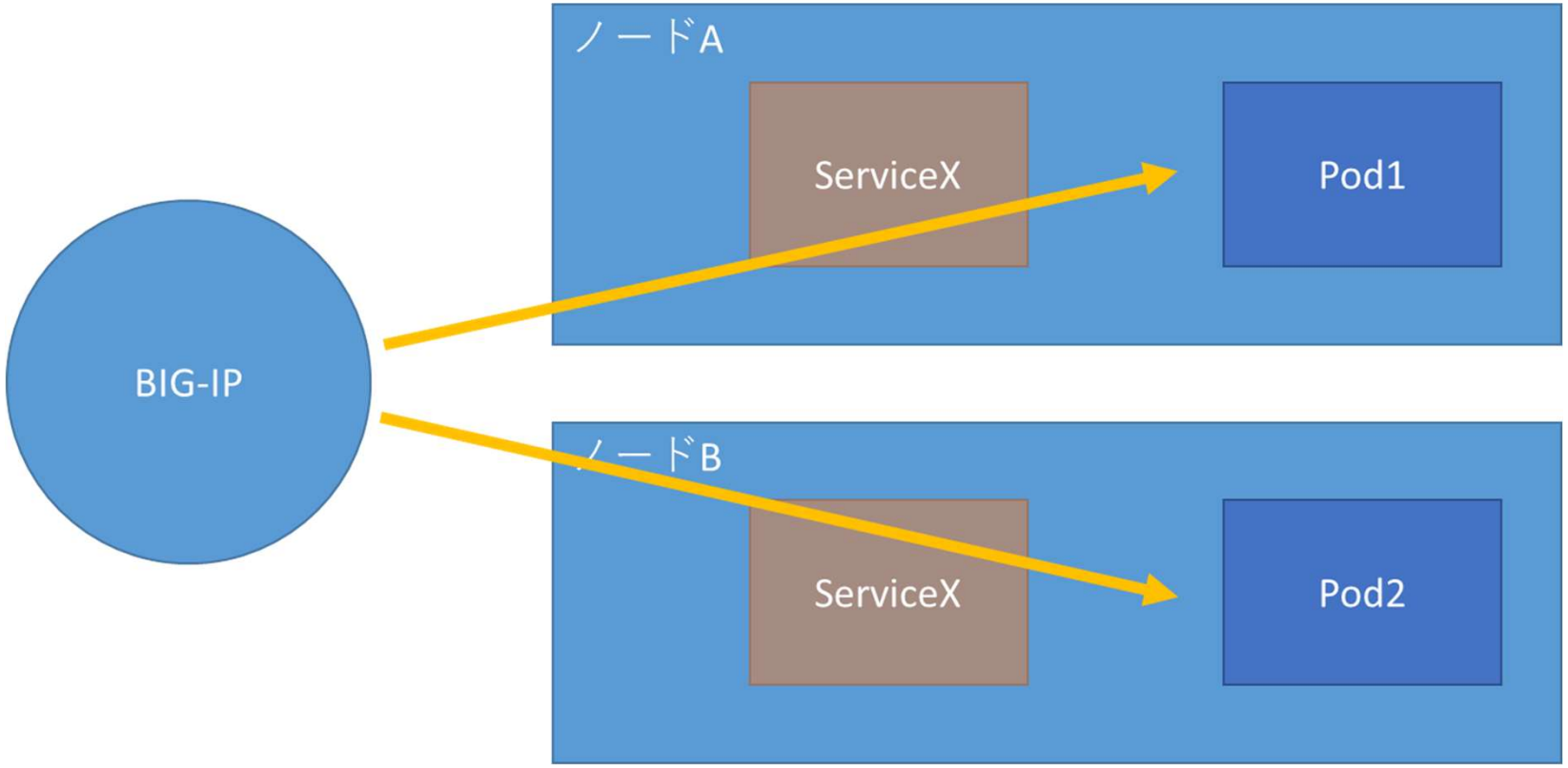


F5 Container Connectorの仕組み

もっと素敵な伝え方を。
Jstream

- Clusterモード
  - BIG-IPをKubernetesのネットワークに組込む
    - 公式にはFlannelとCalicoについて触れられている
  - Podに向けて直接ロードバランスする

F5 Container Connectorの仕組み



F5 Container Connectorの特徴

もっと素敵な伝え方を。 Jstream

- LoadBalancerタイプのServiceとしては動かない
  - IPアドレスの払い出しは手動
  - JSONでBIG-IPのコンフィグを書く
- やろうと思えばIPv6->IPv4の変換も

自社の構成

もっと素敵な伝え方を。
Jstream

- kube-router+F5 CC(Cluster)で構築中
- FlannelとBIG-IPの組合せが面倒
  - VXLANの設定を全部手で移す
  - host-gwならスタティックルートをノード数ぶん

自社の構成

もっと素敵な伝え方を。
Jstream

- kube-router+F5 CC(Cluster)で構築中
- FlannelにはNetworkPolicyがない
  - CalicoなどのNetworkPolicyだけ借りることもできるが、それならFlannelでなくても。

自社の構成

もっと素敵な伝え方を。 Jstream

- kube-router+F5 CC(Cluster)で構築中
- BGPフルメッシュだとノード追加の度にBIG-IPにピアリング設定が必要
  - Calicoはルートリフレクタが別途必要
  - kube-routerはクラスタ内からルートリフレクタを用意できる

議論したいこと

もっと素敵な伝え方を。 Jstream

- どんな構成で使っているか
  - 流行りはCalico？
  - ロードバランサーを自前で実装？
- IPv6対応するには
  - IPv4を捨てると今でも動くらしいが……
  - もうすぐデュアルスタックサポートがくる？
- 運用するときに考えること
  - iptables地獄のデバッグ？

ありがとうございました！

IPv6 onlyの夢

もっと素敵な伝え方を。 Jstream

- クラスタで使用するIPアドレスを全てグローバルに
- BGP対応のCNIでルーターに広報
- PodへのアクセスNetworkPolicyで