

# ネットワーク屋がコンテナスキルを 身につけたいと考えたとき

NEC 金海好彦

### はじめに

- ◆ これまで箱庭で遊んでいたコンテナ技術(k8sやdocker)を、2023年4月から仕事でも 担当することになり、楽しみながら四苦八苦しています。その状況(半分弱音?) を共有します。
- ◆ ここでの話は私見であり、会社としての見解ではありません。
- ◆ そもそもコラム的な内容ですので、時間ともに忘れてください、僕も忘れます。
- ◆コンテナ技術やk8sの一般論は話しません。各自ChatGPT等に聞いてください。

### 自己紹介

- ◆ 現在のスペック
  - ■ずっとNEC。NICTの特別研究員やCKPの幹事等々もしてます。
  - ネットワーク業界をうろうろ(ShowNetやJGN、CKP、WIDE、JANOGとかとか)
  - ■4月にOSSやKubernetes を担当する部門に異動(技術開発)
  - ■ソースコードを読めないし、見えません!
  - ■最近、同じ設定をたくさんのマシンに施すので、設定したことをシェルスクリプト化し、github にあげる程度はできるようになりました
  - YAML はなんとか見えるようになってきました(読んで理解する手前)



### 異動してから感じていること

- ◆ コンテナススキルを手に入れるためには(私見)
  - ソースコードを読み、理解、実装 → 実装状況や変化を探索等々
  - カーネル/OS → 権限制御等
  - ストレージ → PV(Persistent Volume)の管理等
  - ■データベース(RDBだけでなく、NoSQLとかも) →情報は非同期なので、焦ってはだめです
  - ネットワークも大切 → 複雑になる一方、LBも大切
  - ■障害の再現力(でもムズイ) → ここが一番大切
- ◆1人では無理なので、巨人たちの肩に乗っています
  - ■巨人たち=エスパーと魔術師



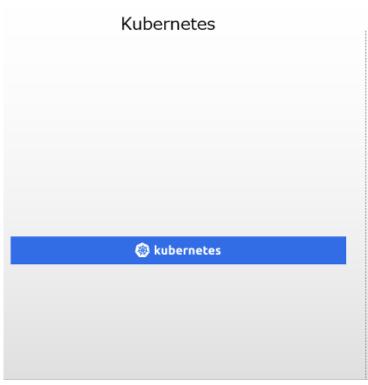
### 所属チーム

- ◆担当プロダクト: k8s/OpenShift
- ◆ カーネル屋の集団
  - 過去に自分たちでOSを開発してきた巨人たち(エスパーと魔術師)
  - ■なになに口グを取れとか、とってない場合は解析できないとか、お客様やSEに要求(上から目線でw)
  - ■でも分析結果には満足してもらえているので、ステークホルダーからのチームの評価は高い
- ◆顧客先で起きた障害を自社内の環境で再現させちゃう
- ◆ レファレンスモデルやTIPS を社内に公開し、サポート範囲を限定しちゃう
  - ■k8sって、広範囲で多機能、さらにブランチが多すぎ、かつ短い間隔でアップデートされる
  - ■しかも、アップグレードが袋小路になる場合もある(そうなってしまったら、クラスタ再構築)
  - k8s/OpenShift の公開マニュアルに誤記(嘘?)が多い

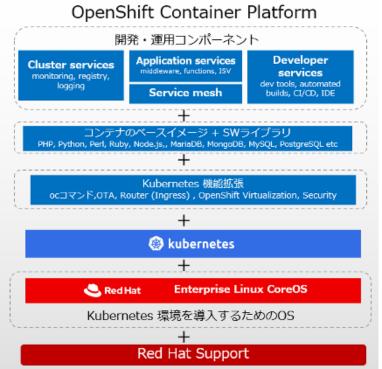


### 担当する部品

- ◆ サービスメッシュ
  - ■実機を使った動作確認
  - ■ユースケースや利用シーンを創作(妄 想)



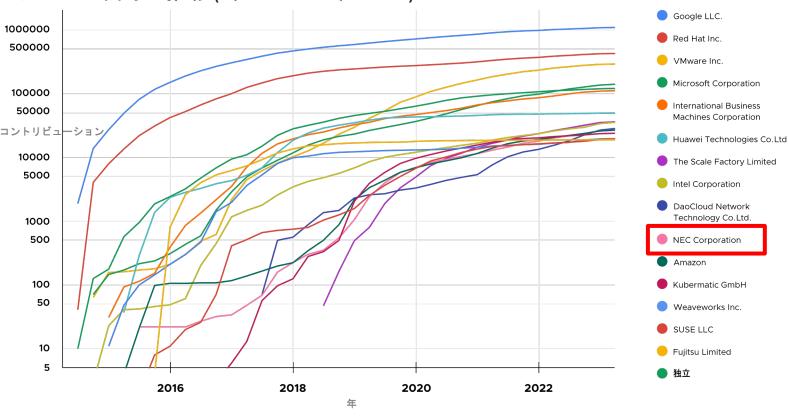
- ◆TIPS作成
  - クラスタを作って、使って、見つける(自作自演)
  - ■チーム各自が持っている知見を共有知に
  - ■もちろん公開は社内のみ





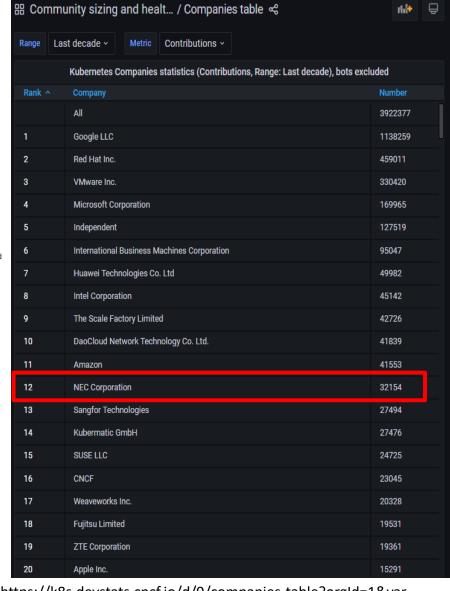
### NEC と Kubernetes の関係

#### 過去10年間の推移(Q2 2014 - Q2 2023)



https://www.cncf.io/reports/kubernetes-project-journey-report-jp/

#### 過去10年間の累計



https://k8s.devstats.cncf.io/d/9/companies-table?orgId=1&var-period\_name=Last%20decade&var-metric=contributions



### NEC のKubernetes への貢献

### k8s Node障害復旧制御機構の導入に貢献 (デザイン策定/k8sコミュニティへのpush)

Q Search

#### 2023

Kubernetes Legacy Package Repositories Will Be Frozen On September 13, 2023

Gateway API v0.8.0: Introducing Service Mesh Support

Kubernetes 1.28: A New (alpha) Mechanism For Safer Cluster Upgrades

Kubernetes v1.28: Introducing native sidecar

#### Kubernetes 1.28: Non-Graceful Node Shutdown Moves to GA

Wednesday, August 16, 2023

Authors: Xing Yang (VMware) and Ashutosh Kumar (Elastic)

The Kubernetes Non-Graceful Node Shutdown feature is now GA in Kubernetes v1.28. It was introduced as alpha in Kubernetes v1.24, and promoted to beta in Kubernetes v1.26. This feature allows stateful workloads to restart on a different node if the original node is shutdown unexpectedly or ends up in a non-recoverable state such as the hardware failure or unresponsive OS.

What is a Non-Graceful Node Shutdown

**Self Node Remediation Operator 強化、** 貢献者としてベースOSSの**Medik8s**にて NEC口ゴ掲載



#### Medik8s - Kubernetes Node Remediation

Medik8s is a project consists of several kubernetes operators that provide automatic node remediation and high availability for singleton workloads

Collaborators



Orchestrating a brighter world



https://www.medik8s.io/

https://kubernetes.io/blog/2023/08/16/kubernetes-1-28-non-graceful-node-shutdown-ga/



### 4.12.0から4.12.37へのアップグレードパス

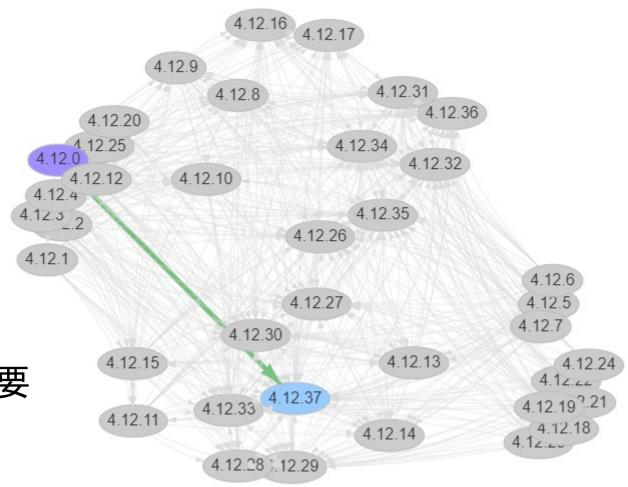
#### OpenShiftの場合

#### (2023年10月20日現在)

- 4.12.0は、4.12台で初めにStable になったバージョン
- 4.12.37は、4.12台の最後のStableバージョン



袋小路の場合もあるので注意が必要



### 4.12.0から4.13.15へのアップグレードパス

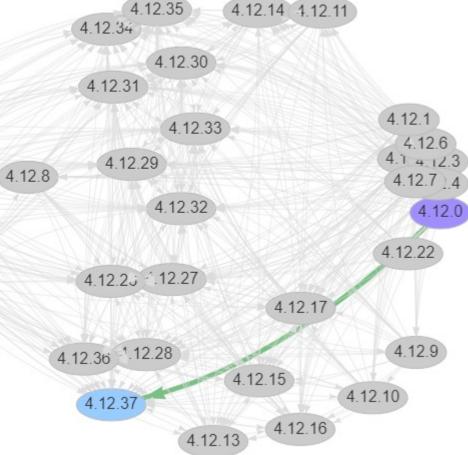
#### OpenShiftの場合

#### (2023年10月20日現在)

- 4.12.0は、4.12台で初めにStable になったバージョン
- 4.13.15は、4.12台の最新のStableバージョン
- 4.12.0から一気に4.13.\*にはアップグレードできない



4.12.25 4.12.20 4.12.20 4.12.21 4.12.24 4.12.24 4.12.22



Orchestrating a brighter world

### 袋小路の場合もあるので注意が必要

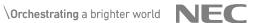
### これまで遭遇したハマリポイント

- ◆ 証明書の管理
  - これまで「インターネット、こんにちは」の世界で生きてきて、いきなり。。。
  - ここを通らないと先に進めないので、頑張る。証明書を無視するIn-secure の設定は御法度
- ◆ ネット上に落ちている情報は古い・片手落ち・無責任
  - とは言っても、Qiitaは結構参考になる
  - ChatGPTや社内のLLMを使って、整理
  - 最終的に、自分なりにまとめて、ドキュメントを作る
- ◆ (k8s とは関係ないですが)評価環境の構築が大変。単純なルーティングだけじゃだめ
  - ルータがないので、Linuxマシンでルーティング
  - IP masquerade、redirect やfirewalld を駆使
  - これまでのSDNを含んだネットワークを"グリグリ"する経験が生きた



### 読書感想文

- ◆ K8s/OpenShift 環境を作るだけなら、まぁ簡単。でも、運用も考えよう
  - ■プロダクションなシステムでは、テへると偉い人が謝り侍化
  - テストベッドでSDN で作ることの大変さは経験したが、運用はフリーダムだった
  - ■OSPFやTCP等のプロトコルのタイムアウトまでに復旧すればいいので、UTPやファイバーを一瞬、 抜き差ししていました。もうそんなことはできません。。。
- ◆ところで、そもそもカーネルって?
  - ■用語を知らなさすぎ(例:InboxとOutboxのドライバー、Non-Maskable Interrupt等々)
- ◆ ほとんどのお客様はインターネットから切り離してほしい(つないでもええやん)
  - ■セキュリティ
  - ■要求に従うと構築手順がとても複雑



### 読書感想文

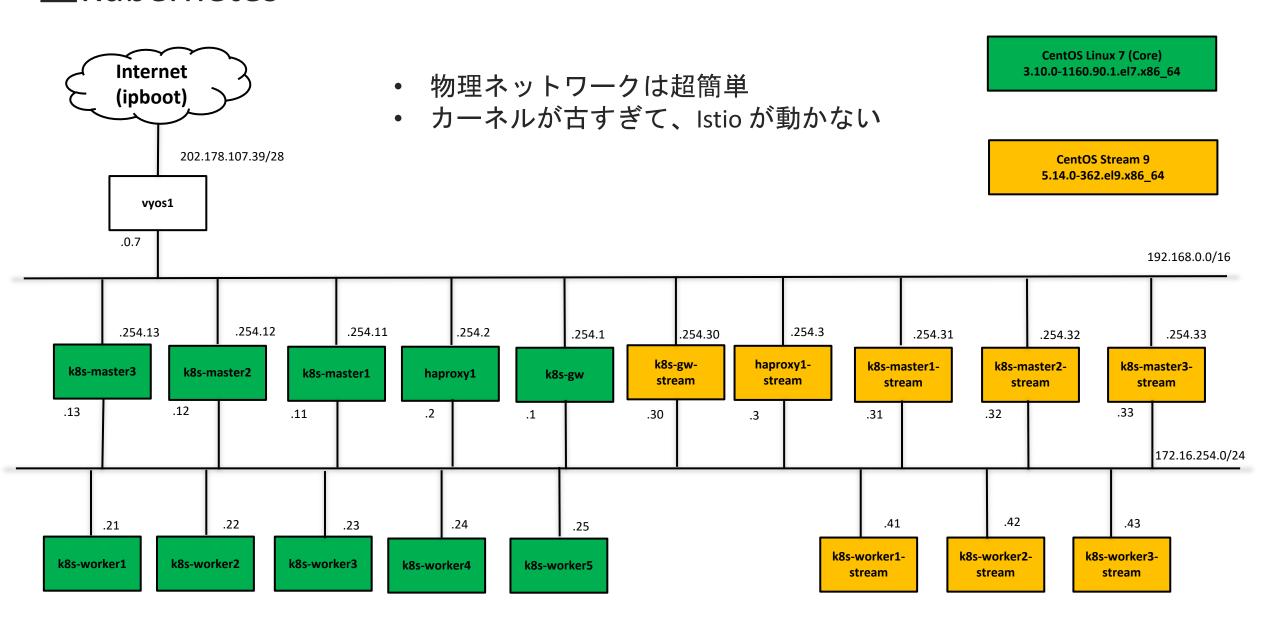
- ◆ やっぱり、証明書、ムズイ(キライ)
- ◆ カスタムイメージのビルド
  - Dockerhub から持ってくればいいんじゃないの?セキュリティどうのどうのでNG
  - k8s 上で開発する人の気持ちになってみるが、アプリ作ったことないので、限界が。。。
- ◆ ネットワークは複雑怪奇
  - iptablesとOpenFlow 1.3の混在
  - OpenShift では、そろそろOVNとなりOpenFlowのみでNicra 拡張(メガップレマス)



### OVNのフローテーブル

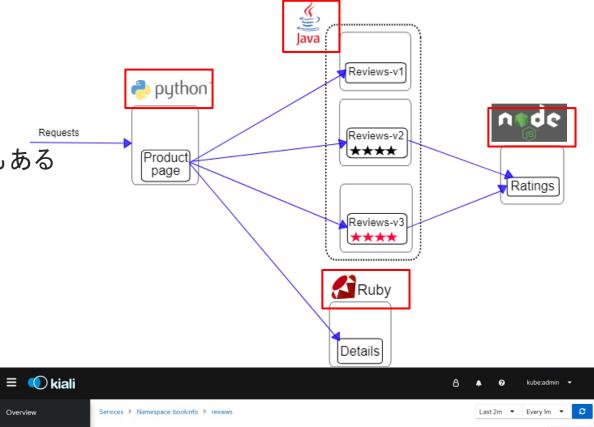
ovnkube-db かovnkube-master にログインして"ovn-nbctl show" ovn-Kubernetes 論理ネットワーク 00:00:a9:fe:21:01 169.254.33.1/24 00:00:a9:fe:21:01 169.254.33.1/24 00:00:a9:fe:21:01 169.254.33.1/24 br-local master-ovn br-local worker2-oyr br-local worker1-ovn ext master-ovn ext worker2-ovn ext worker1-ovn etor-GR worker2-ovn etor-GR master-ovn etor-GR worker1-ovn addresses: ["26:a4:6b:16:8d:4c"] rtoe-GR master-ovn addresses: ["de:1e:e4:7d:a2:48"] mac: "de:1e:e4:7d:a2:48" rtoe-GR worker1-ovn rtoe-GR worker2-ovn ["7a:84:61:7e:02:48"] networks: [\*169,254,33,2/24"] mac: "7a:84:61:7e:02:48" mac: "26:a4:6b:16:8d:4c" nat networks [\*169.254.33 2/24\*\* networks nat nat ["169.254.33.2/24"] GR worker2-own external ip: GR master-own external ip: GR worker1-ovn external ip 169.254.33.2 "169.254.33.2" rtoj-GR worker2-ovn 169.254.33.2 logical ip: logical ip: rtoi-GR master-or rtoj-GR worker1-own mac: \*0a:58:64:40:02:0 logical io: 10.244.0.0/16 \*10.244.0.0/16 mac: "0a:58:64:40:01:01 networks: mac: "0a:58:64:40:00:01" \*10.244.0.0/16 type: "snat" type: "snat" ["100,64,2,1/29"] networks: type: "snat" ["100.64.1.1/29"] itor-GR worker2-ovn ["100.64.0.1/29"] itor-GR worker1-ovn A STATE OF itor-GR master-ovn join worker2-ovn join worker1-ovn join\_master-ovn jtod-worker2-ovn itod-worker1-ovn itod-master-ovn dtoj-worker1-ovn mac: "0a:58:64:40:01:02" dtoj-worker2-ovn mac: "0a:58:64:40:02:02" mac: "0a:58:64:40:00:02" networks networks: networks: ["100.64.2.2/29"] ovn cluster router ["100.64.0.2/29"] ["100.64.1.2/29"] rtos-master-ovn rtos-worker1-ovn rtos-worker2-ovn mac: "0a:58:0a:f4:02:01" mac: "0a:58:0a:f4:00:01" mac: "0a:58:0a:f4:01:01" networks: networks: networks: ["10.244.0.1/24"] ["10.244.2.1/24"] ["10.244.1.1/24"] switch 4826b70a-8e46-4d7a-9835-5f54da34deea switch 2658d826-2237-4706-ada9-111e99cf1ebo switch 8dfd967e-cd89-487e-9ece-9766563bef6b ext master-own)
port br-local master-own
type: localnet
addresses: ["unknown"]
port etor-GR master-oyn ext worker1-ovn) port etor-GR worker1-ovn (ext worker2-ovn) port br-local worker2-ovn stor-worker2-ovn stor-master-ovn port etor-usk workert-ovn type: router addresses: ["78:84:61:7e:02:48"] router-port: rice-GR worker1-ovn port br-local worker1-ovn type: localnet addresses: ["unknown"] stor-worker1-avn port br-local workerz-ovn type: localinet addresses: ["unknown"] port etor-GR workerz-ovn type: router addresses: ["26:a4:6b16:8d:4c"] addresses: f4:01:01"] addresses: 1 0a:58:0a:f4:02: addresses: ["0a:58:0a:f4:00:01"] コンテナ master-ovn type: router addresses: ["de:1e:e4:7d:a2:48"] 10.244.2.0/2 worker1-ovn 10.244.0.0/24 10.244.1.0/2 router do of 1: doc 68 master own router bot 611 5-2015-468 - 500b - 450b 1260 3cb8 (58 master own port not of 50 master own port not of 50 master own port not 60 master own not 50 master own not 50 master own port not 60 master own port not 60 master own not 60 router 676afb70-f026-43c1-b7cf-54d3e5d582cb router 03d862c4-8625-4707-afcd-122750a5379d router 034862c4-8625-4107-sfcd-122750s5379d (GR wirker2-own port ftd: GR worker2-own mec: '0x-85664.840.02.01' mec: '0x-85664.840.02.01' port ftde: GR worker2-own mec: '262.856.01 fc866.01' met worker [1169.258.33.2/24'] net worker [1169.258.33.2/24'] net s230.260-002-3-239-9404.88283992109 external or; '109.258.33.2' external or; '109.258.33.2' met s230.264.00.016' k8s-worker1-ovn k8s-master-ovn k8s-worker2-ovn addresses: ["0axc7:28xxd:55:ce addresses: ["82:ff:fe:1b:a6:22 10.244.0.2"] mac: "0a:58:64:40:01:01 addresses: ["8a:35 10,244,2,2"] mac: va:38:64:90.01:01 notworks: [100.64.1):79\*] port rice GR worker1-ovn mac: 7a:28:61:7e0.24-3. notworks: [169.25-33.2:24\*] notworks: [169.25-33.2:24\*] notworks: [169.25-33.2:24\*] logical ip: 10.244.0.016\* 10.244.1.2"] ovn-k8s-mp0 ovn-k8s-mp0 82:ff:fe:1b:a6:22 10.244.0.2/24 8a:39:89:14:4c:55 10.244.2.2/24 0ax:7:28:cd:55:ce 10.244.1.2/2 switch 3a5860c0-e210-45a6-9f02-95a165d7bb82 switch 812e5e4b-850e-461e-86b4-cb61788896a1 ijoin worker1-ovn) port jtod-worker1-ovn type: router (jain worker2-own) port jtod-worker2-own port itod-master-ovn type: router router-port: dtoj-worker2-ovn port jtor-GR worker2-ovn router-port: dtoj-worker1-ovn port.jtor-GR worker1-ovn type: router router-port: dtoi-master-ovn port itor-GR master-ovn type: router router-port: rtoi-GR master-ov router port: rtoj CR worker2 ov router 85cd5c94-4237-4dcb-80e8-3d84dc9f0d8c router 85cd5c94-4237-4dcb-80e (own cluster router) port rtos-worker2-ovn msc: "0x58x0af43/201" networks: [\*10,244,2.1/24\*] port rtos-master-ovn msc: "0x58x0af43/0x01" networks: [\*10,244,2.1/24\*] port dtoi-worker2-ovn msc: "0x58x0af43/0x01" opt rtos-master-ovn msc: "0x58x0af43/0x01" networks: [\*10,244,0/24\*] port dtoi-worker2-ovn msc: "0x58x64x40/0x30" networks: [\*100,64,2.2/29\*] switch 608522a7-frlor-4o57-9284-21ffo7cffh1h switch b4f65a5f-ef68-4c2e-9dcb-1fee0b31007c iswitch 60l53/247-1d66-4e57-9284-21Tfe7cf workier2-own port default ngine-dev-579d499b5-zs9wz addresses: ["dynamic"] port default centes-tool2 addresses: ["dynamic"] port default centes-tool3 addresses: ["dynamic"] master-ovn) port stor-master-ovn type: router addresses: ["0a:58:0a:f4:00:01"] router-port: rtos-master-ovn port kube-system coredns-66bff467f8-rokz8 addresses: ["e6:cd:34:f4:00:0510.244.0.4"] addresses: 1 eo.co.2====0 port k8s-master-ovn addresses: 1"82-fffe:1ba6:22.10.244.0.2"1 port kube-system coredns-66bff467f8-scal6 addresses: 1"e6icd:34rf4:00:04.10.244.0.3" accoresses: [\*oynamic\*]
port centos-ssh-ostos-ssh-95b48bd5-gzwt
addresses: [\*dynamic\*]
port stor-worker2-own
type: router
addresses: [\*0ac58:0arf4:02:01\*]
put/forestricts-propriess\* networks: [\*100,64.2.2/29\*]
port dtoi-wwikerl-own
msc: "0e:58:64:40:01:02\*
networks: [\*100,64.1.2/29\*]
port rtos-workerl-own
msc: "0e:58:06:4601:01\*
networks: [\*10,244.1.1/24\*]
port dtoi-master-own
msc: "0e:58:06:440:00:02\*
networks: [\*100,44.0.2/29\*] router-port: rtos-worker2-ovn port default rejine: 7768f77d5c-55x4d addresses: ["dynamic"] port k8s-worker2-ovn addressec ["8s-30x8914rkci55-10.244.2.2"] ovn-k8s-mp0 avn-k8s-mo0 switch bad6-128e-4c6e-452e-94ac-3439b806d934 (worker1-vvv) (worker1-vvv) (worker1-vvv) addresses ("thyramic") port (88s-worker1-vvv) addresses ("thyramic") port (88s-worker1-vvv) addresses ("thyramic") port default righter ("thyramic") addresses ("thyramic") addresses ("thyramic") addresses ("thyramic") addresses ("thyramic") addresses ("thyramic") addresses ("thyramic") ovn-k8s-mp0 \Orchestrating a brighter world 14 29 © NEC Corporation 2023

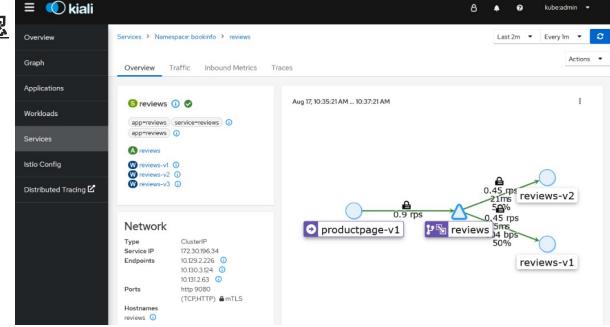
### 生Kubernetes

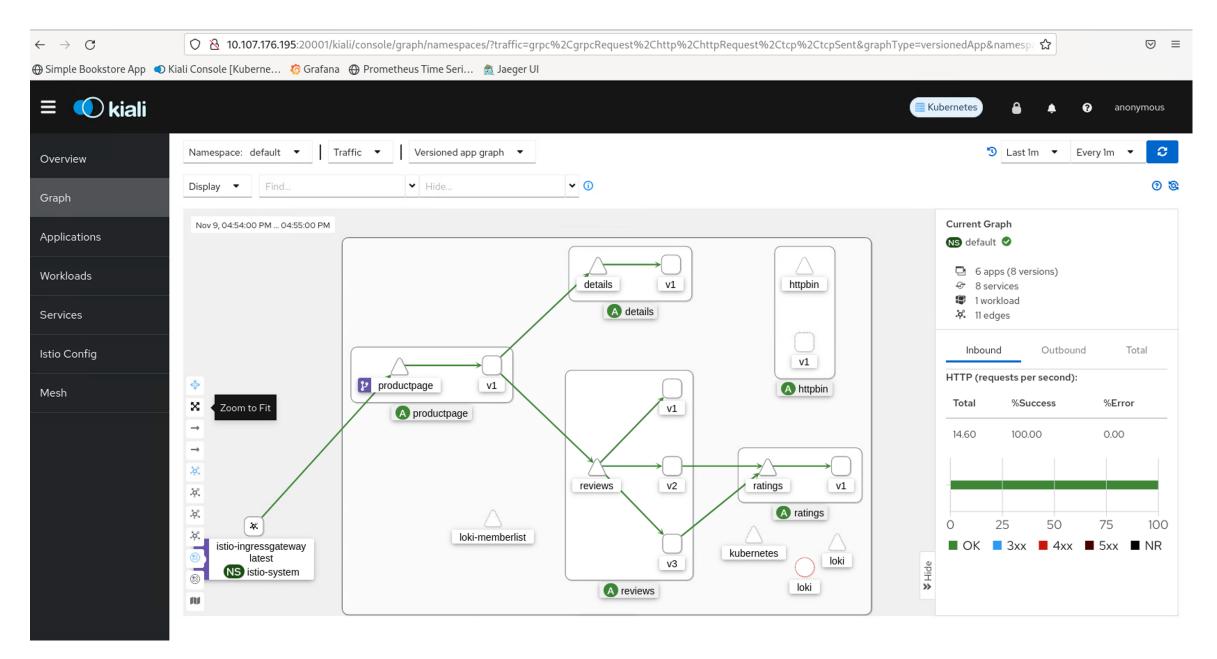


## サービスメッシュ(Istio) のデモ

- ◆ サービスメッシュの利点
  - 確かにネットワーク的な要素はあるが、それ以外にもある
  - 多言語で実装された差分を吸収(gRPC と同じ?)
  - ボトルネックを発見・可視化
- ◆ ユースケース
  - バックエンドへのトラフィックの制御
  - 遅延シミュレーションで遅延箇所の特定
  - 障害シミュレーションで<u>サービスの稼働状況の確認</u>
  - バックエンドサービスの過負荷防止
  - マイクロサービス間通信の制御
  - Service Mesh 内から外部サービスへの通信の制御
  - トラフィックミラーリング
  - Path ベース ルーティング
  - Rate limit







### うちの部署に来たい人は声かけて

- ◆ カーネル/OS分かる人
- ◆ I/O分かる人(ネットワークを含む)
- ◆コード書ける人
- ◆ OSS開発に興味がある人
- ◆ 手を動かすのが好きな人
- ◆エスパーや魔術師と一緒に仕事したい人

ただし、大きな組織なので、それなりに面倒なことはあります



### 最後に

- ◆ ご清聴ありがとうございました。
  - これからも頑張りますので、励ましのお言葉をお願いします(爆)
- ◆ サービスメッシュって流行るの?使っている人って限定的よね?
  - 使っているなら、どんな使い方してる?
- ◆ これまでやってきたことは裏切らないはず
  - 体系的に理解しようするスタンスや、検証環境構築、サービスメッシュに生かせた(と思う)
- ◆ K8s のLTS のアンケートに答えて欲しい
  - 頻繁にアップデートされると運用の現場は評価で時間が潰れる
  - http://bit.ly/k8s-upgrade-survey

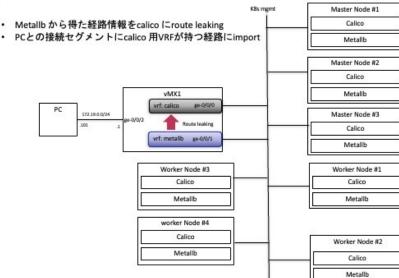






	ASN
vMX1	64512
Calico	64513
Metallb	64514





19