

vFORUM

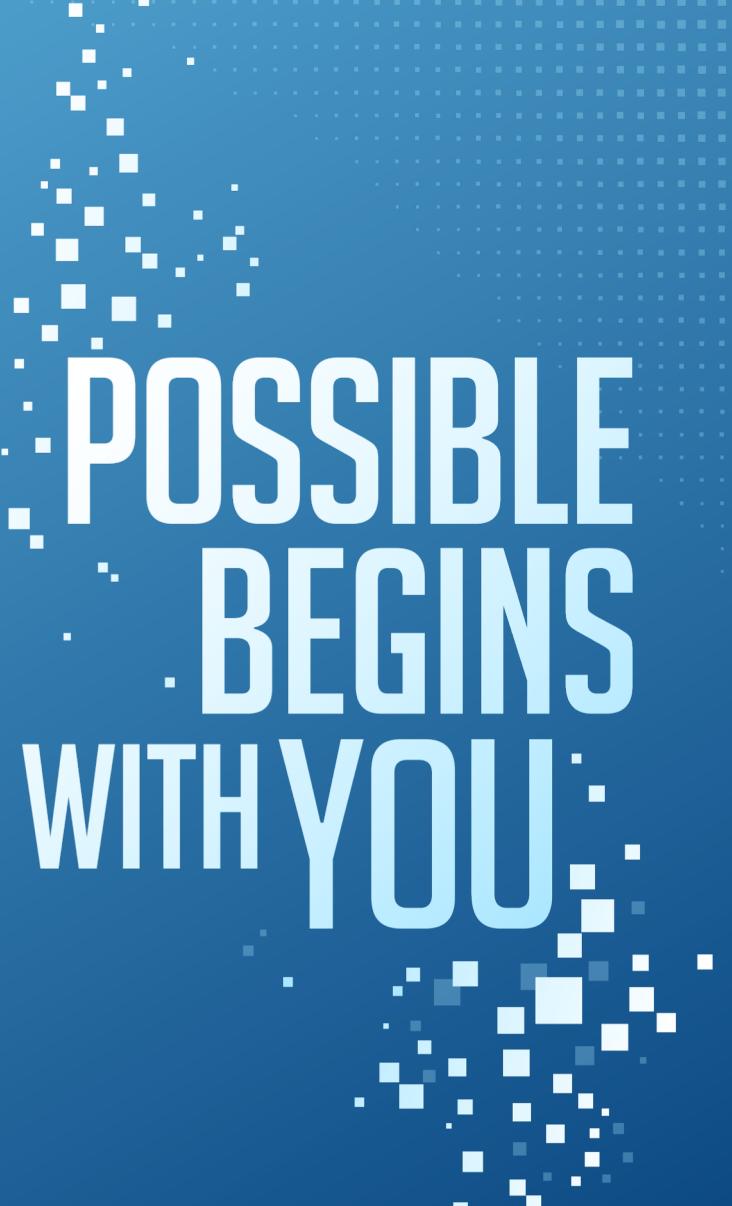
DC138

VMware PKS の提供する
次世代アプリケーションプラットフォーム
インフラ視点での Kubernetes との付き合い方

ヴィエムウェア株式会社
ソリューションビジネス本部 クラウド技術部
クラウドスペシャリスト 岩渕 友裕

#vforumjp

vmware®



POSSIBLE
BEGINS
WITH YOU

免責事項

- このセッションには、現在開発中の製品/サービスの機能が含まれている場合があります。
- 新しいテクノロジーに関するこのセッションおよび概要は、VMware が市販の製品/サービスにこれらの機能を搭載することを約束するものではありません。
- 機能は変更される場合があるため、いかなる種類の契約書、受注書、または販売契約書に記述してはなりません。
- 技術的な問題および市場の需要により、最終的に出荷される製品/サービスでは機能が変わることあります。
- ここで検討されているまたは提示されている新しいテクノロジーまたは機能の価格およびパッケージは、決定されたものではありません。

Agenda



コンテナの市場動向

コンテナ基盤導入に伴う課題とアプローチ

コンテナに関連する複数のテクノロジー

異なるスキルセットと運用手法

分散環境のインテグレーション

導入事例

コンテナの市場動向

新しいワーク LOADへの対応

次世代アプリケーション基盤の準備はできていますか？

既存アプリケーション

主に仮想マシンとして実装



堅牢性



コストの抑制



オンプレミス環境

次世代アプリケーション

主にコンテナとして実装



俊敏性



変化への対応



パブリッククラウド環境



IT 部門は限られたリソースで
両方のニーズに対応していく必要がある

コンテナが採用される背景

次世代アプリケーションの要件を満たすコンテナ

ポートアビリティ(可搬性)

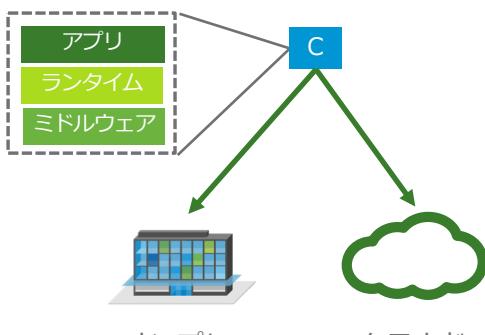
- オンプレ/クラウドをまたいだポートアビリティ
- ビジネス要件に合わせ最適な環境を選択可能

スピード(俊敏性)

- 迅速なアップデートによるサービスの継続的な改善
- ユーザの期待値を超える経験の提供

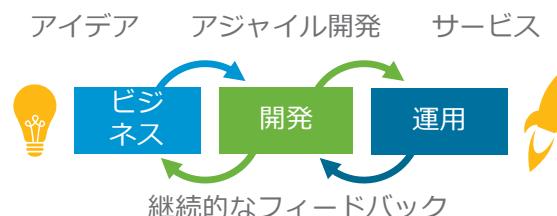
デジタルネイティブなテクノロジー

- 新しいテクノロジーをサービスに適用
- デジタルビジネスにおける競争優位性



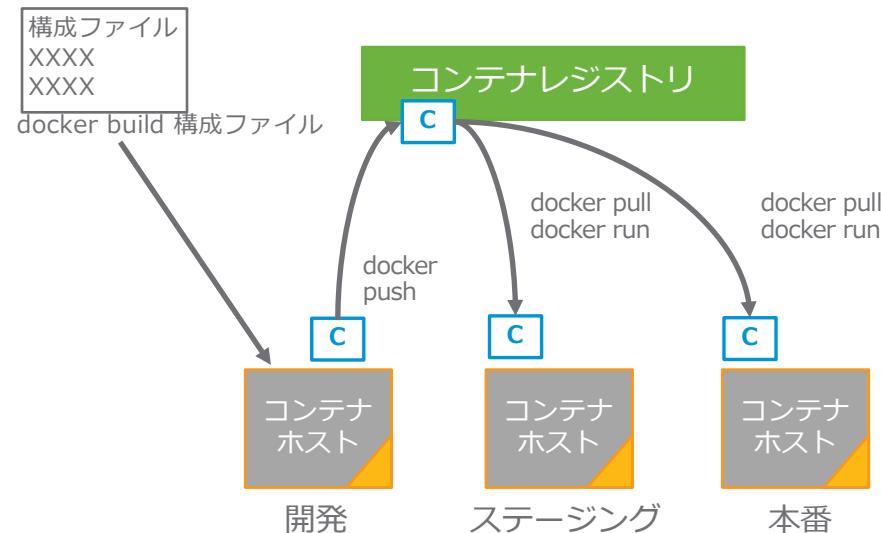
vmware®

©2018 VMware, Inc.



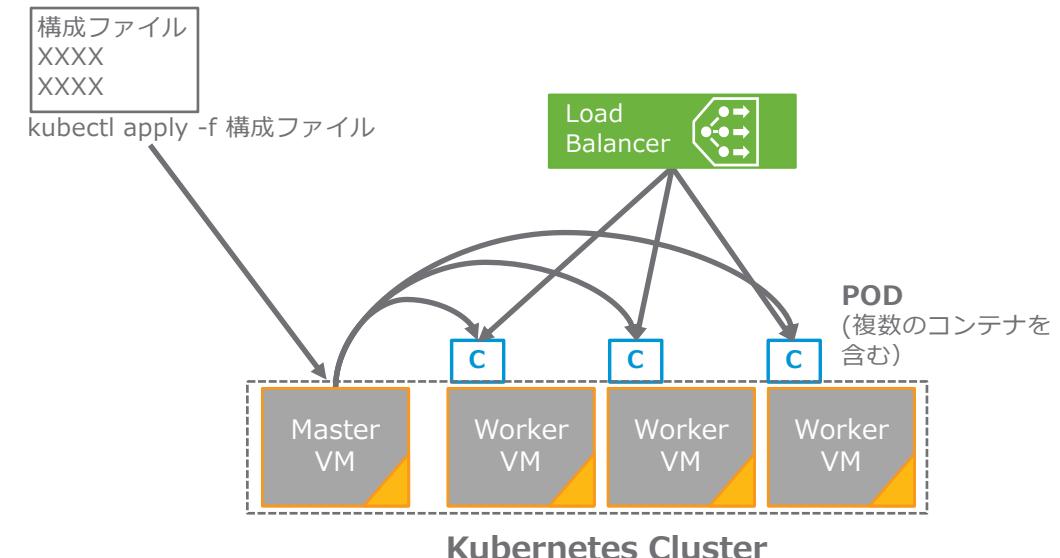
「コンテナ」に含まれるもの

コンテナエンジンとオーケストレーション



コンテナエンジン (Docker)

- **単一のホスト上**でコンテナを動かす
- コンテナのレプリケーション、ネットワーク及びストレージの設定は含まない

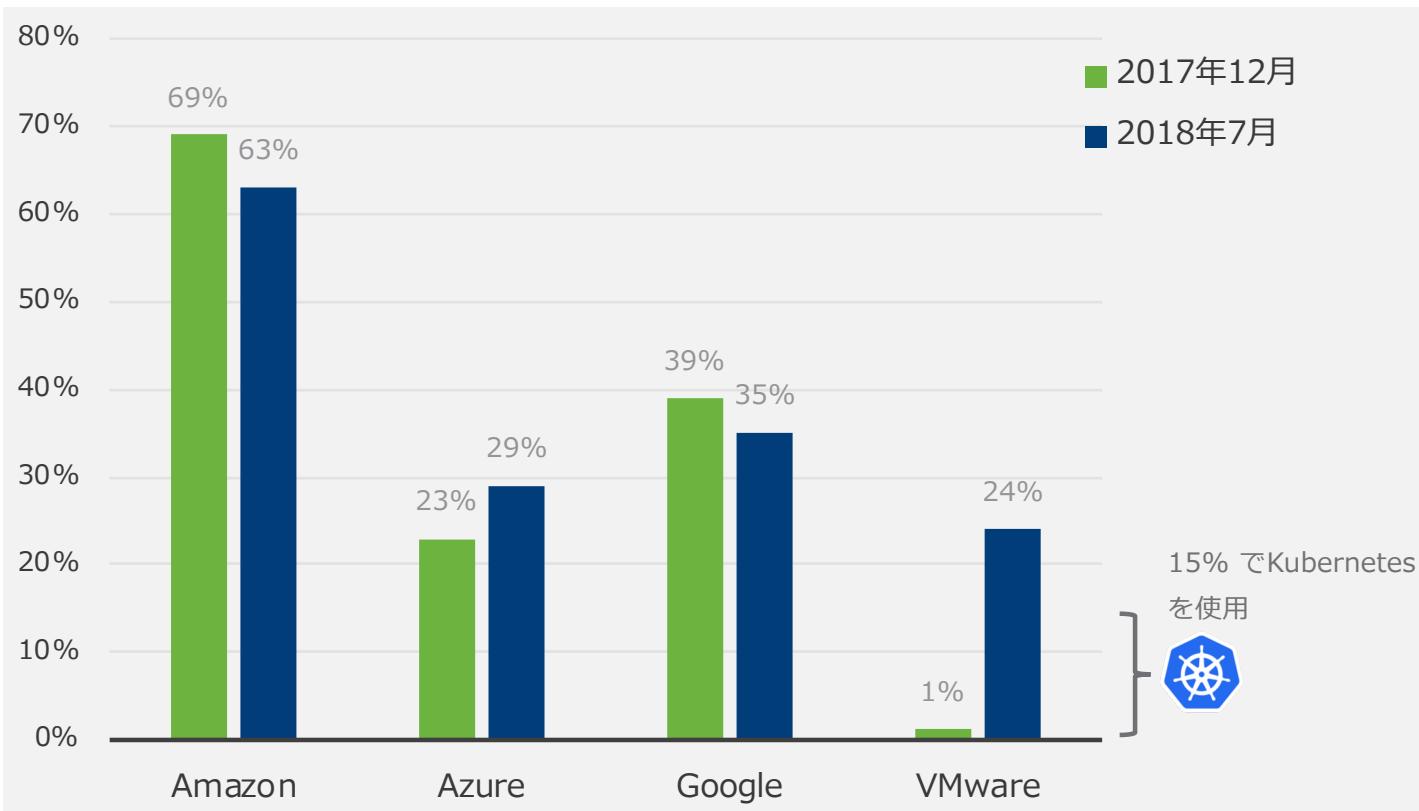


オーケストレーション (Kubernetes)

- **クラスタ化されたホスト上**でコンテナを動かす
- Master ノードが Worker ノードを含むクラスターを管理
- コンテナのレプリケーション、**ネットワークセキュリティ**及び**ストレージの仕組み**を内包

コンテナ基盤として使用されるプラットフォーム

2018年7月のCNCF サーベイからの抜粋



VMware 環境をコンテナ基盤として活用するケースの増加

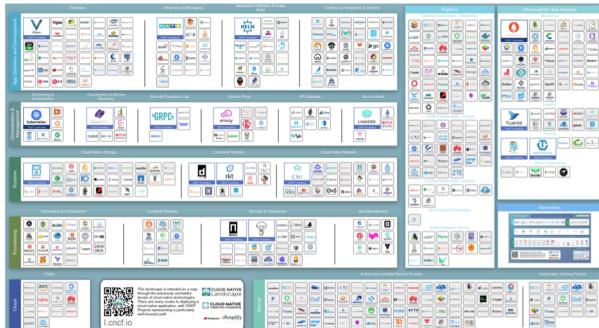
- 2018 年7 月の調査で24% に (2017 年12 月は1.2%)
- ハイブリッドクラウド環境でのコンテナの運用
- VMware (仮想マシン) 上でコンテナ基盤を構成

コンテナ基盤導入に伴う課題

主要な検討項目

コンテナに関連する複数のテクノロジー

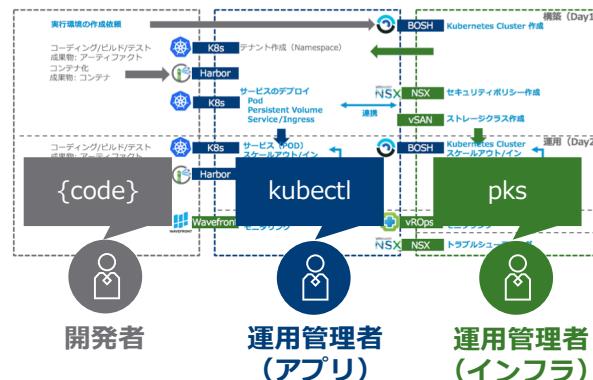
- 最適なテクノロジーの選択と運用
- OSS を運用する人員の確保



<https://landscape.cncf.io/format=landscape>

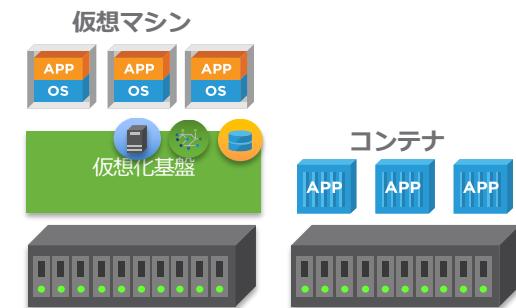
スキルセットと運用手法

- 新しいスキルセットの習得
- 運用フローの見直し



分散環境のインテグレーション

- コンテナ - 仮想マシン
- オンプレ - クラウド

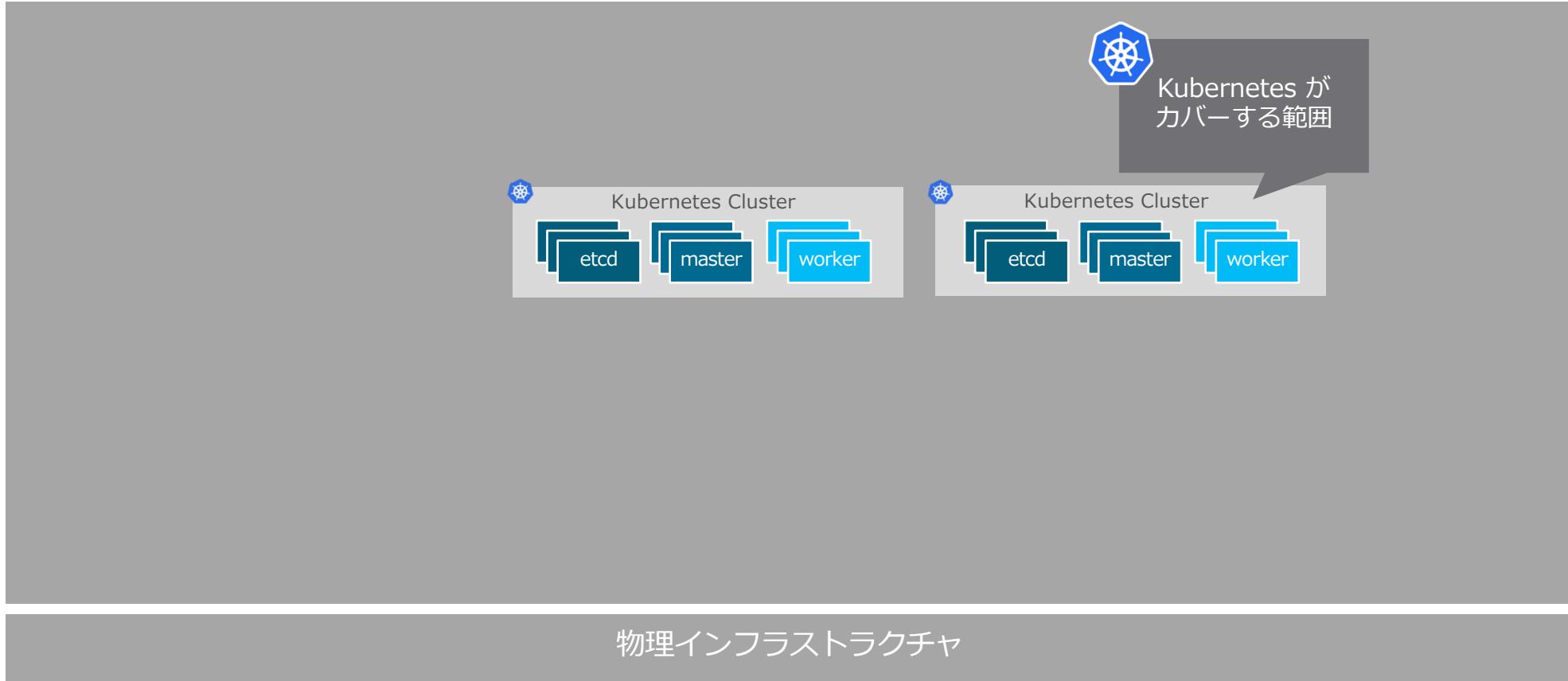


コンテナ基盤導入に伴う 課題とアプローチ

コンテナに関連する複数のテクノロジー

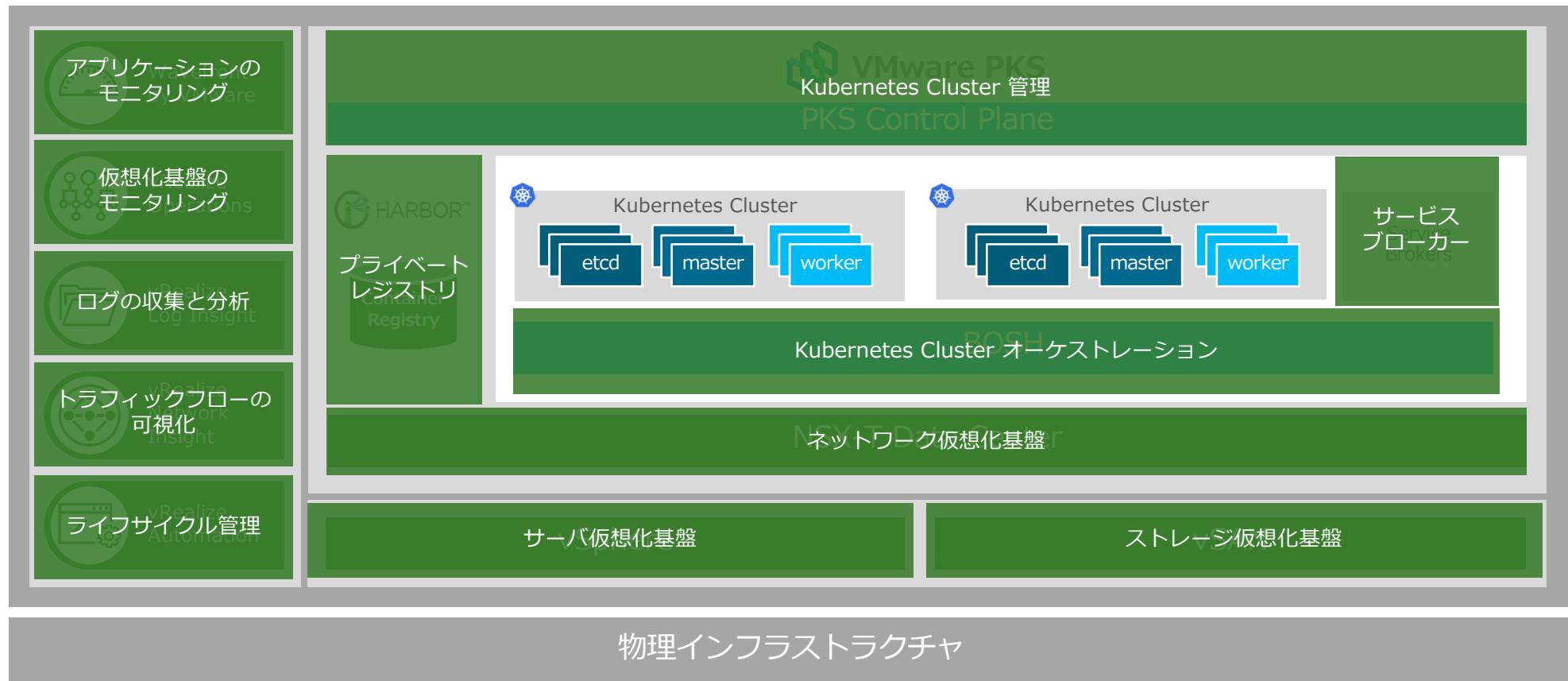
Kubernetes がカバーする範囲

コンテナ基盤の中での Kubernetes



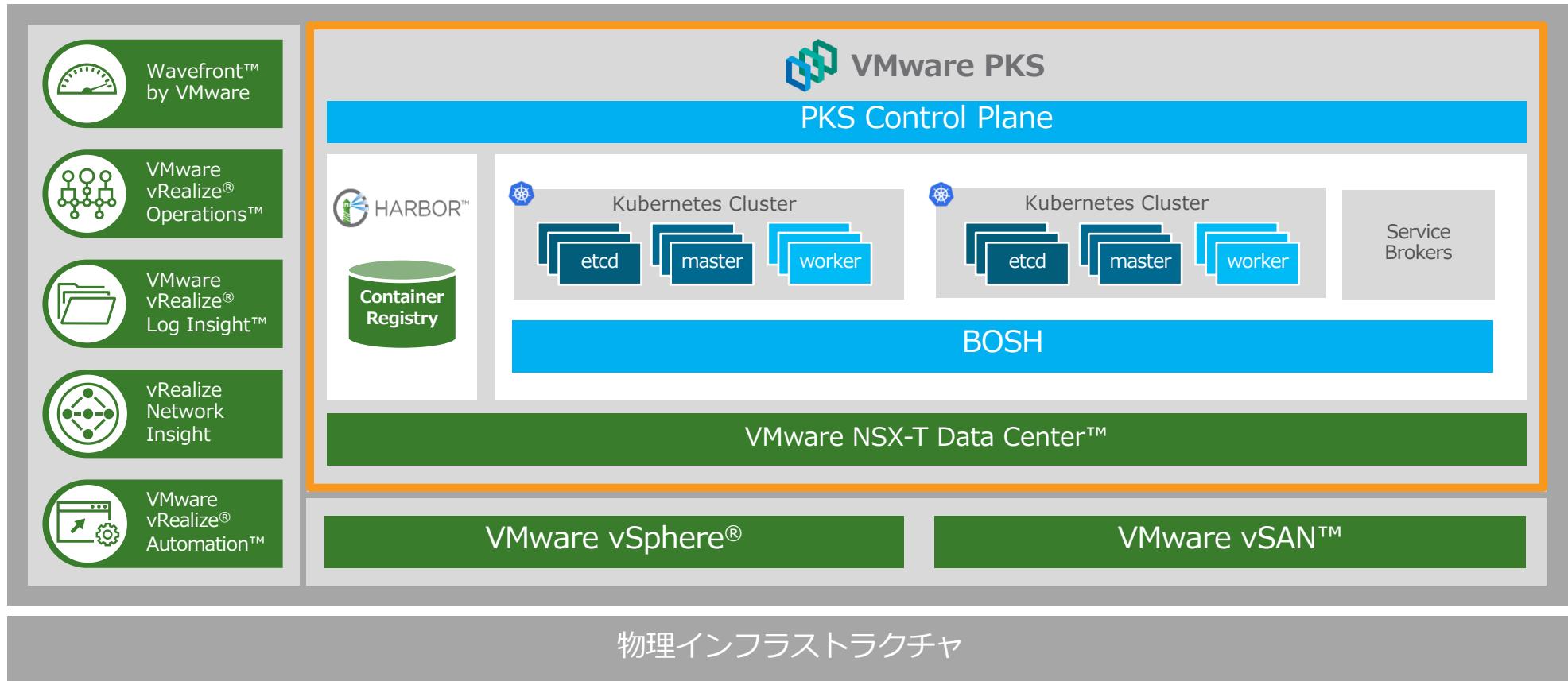
Kubernetes がカバーする範囲

コンテナ基盤の中での Kubernetes



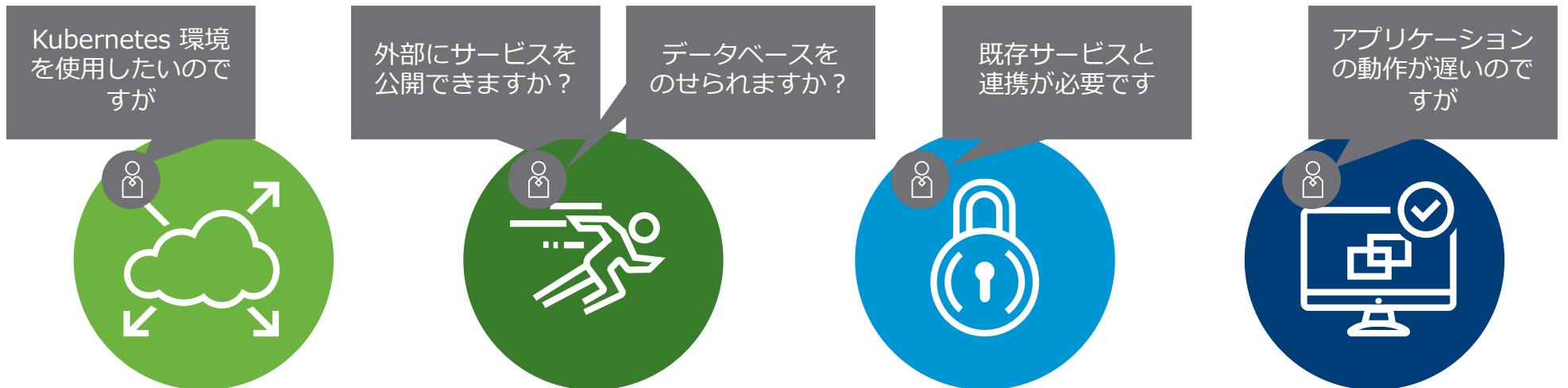
VMware PKS + SDDC

コンテナ基盤に必要なコンポーネントを包括的に提供



テクノロジー視点での課題

サービス提供部門からの要望



コンテナ環境の構築と運用

クラスターの拡出し
パッチ適用
障害の検知と復旧

インフラ構成の自動化

コンテナへの
ネットワーク/ストレージ
サービスの提供

セキュリティ

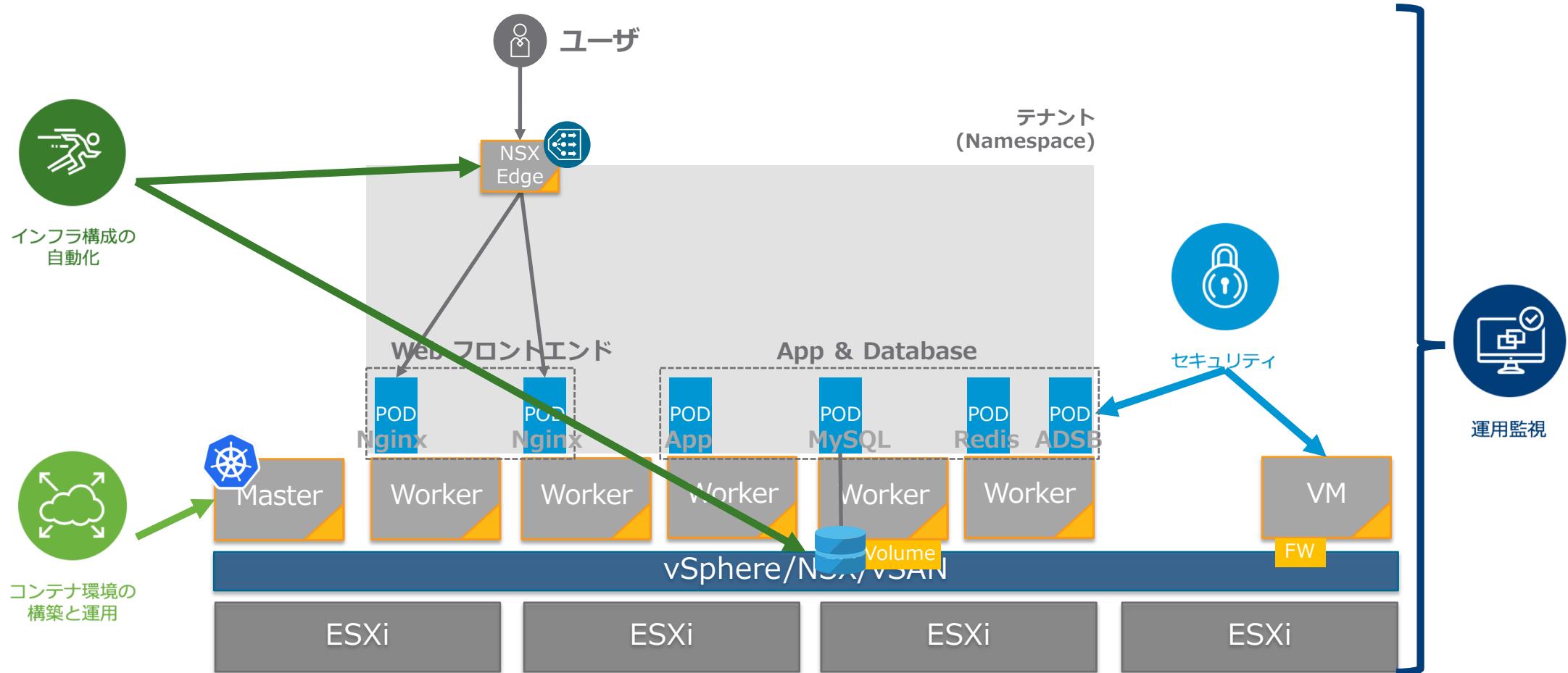
コンテナ間や既存サーバ
とのネットワーク通信におけるセキュリティ

運用監視

コンテナ環境と
コンテナ内アプリの
運用監視

テクノロジー視点での課題

サービス提供部門からの要望



テクノロジー視点での課題

サービス提供部門からの要望

Kubernetes 環境
を使用したいので
すが



コンテナ環境の 構築と運用

クラスターの払い出し
パッチ適用
障害の検知と復旧



インフラ構成の 自動化

コンテナへの
ネットワーク/ストレージ
サービスの提供



セキュリティ

コンテナ間や既存サーバ
とのネットワーク通信に
おけるセキュリティ



運用監視

コンテナ環境と
コンテナ内アプリの
運用監視

コンテナホストの構築と運用

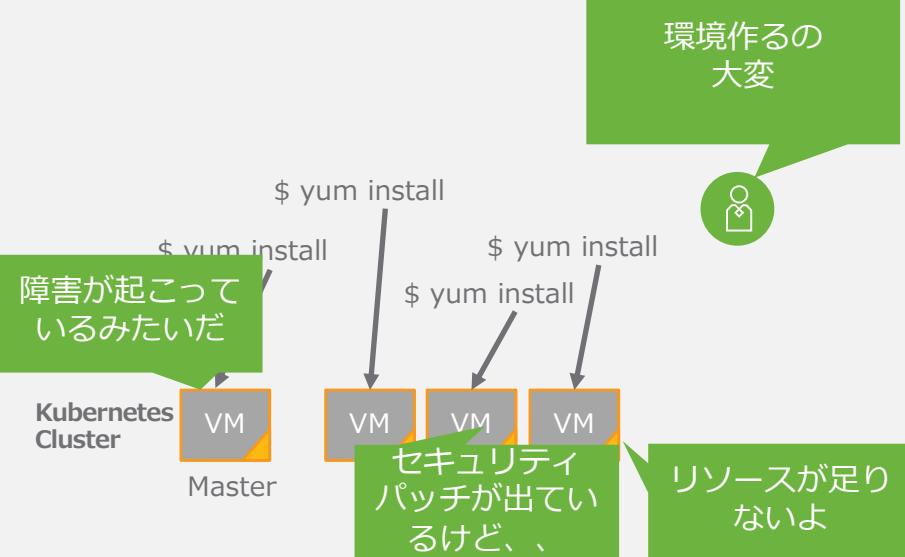
構築だけではなく運用が必要



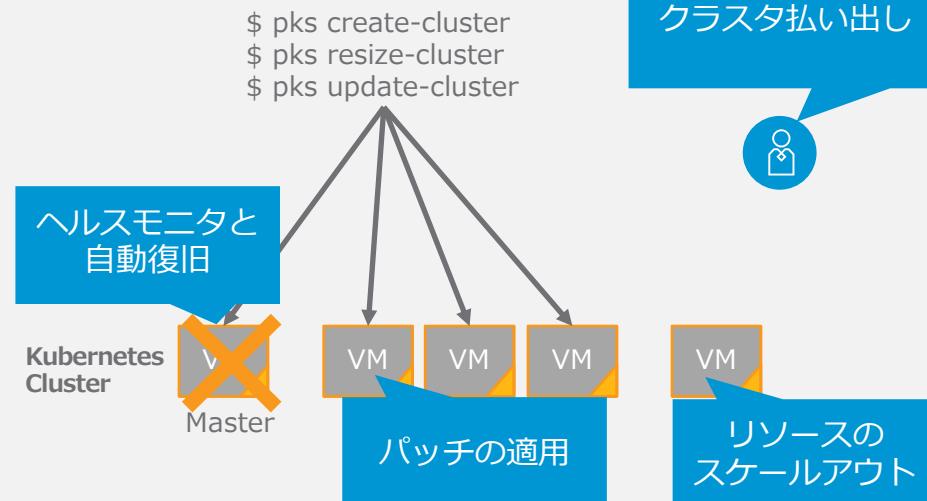
コンテナ環境の構築と運用

手作業による構築・運用

Kubernetes 環境
を使用したいので
すが



PKS による自律的なシステム運用

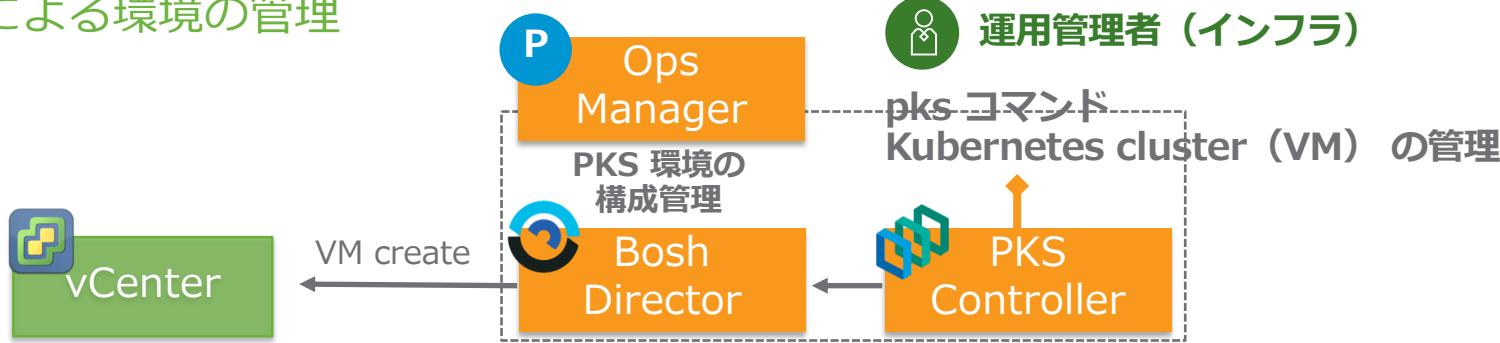


PKS Controllerへのログイン

BOSHによる環境の管理



コンテナ環境の構築と運用

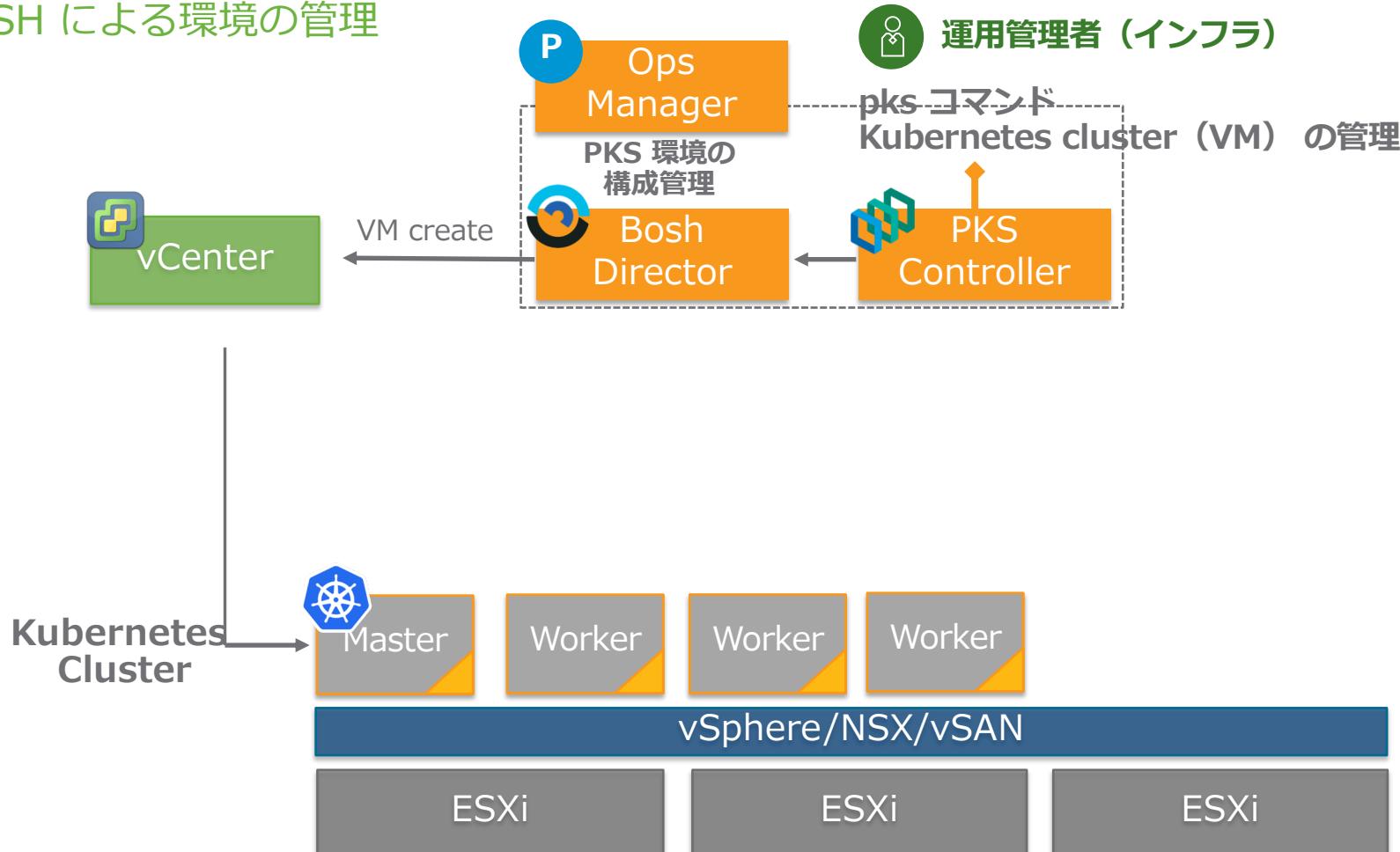


Kubernetes Cluster の構築

BOSH による環境の管理



コンテナ環境の
構築と運用



2. vmware@pks01-term01: ~ (ssh)

```
vmware@pks01-term01:~$ pks create-cluster pks-cluster-100 --external-hostname pks-cluster-100 --plan medium --num-nodes 2
```

K8s クラスタ テンプレートを選択

例) 開発用

 シングルマスタテンプレート

例) 本番用

 マルチマスタテンプレート

K8s ワーカノード数
を選択

2. vmware@pks01-term01: ~ (ssh)

```
vmware@pks01-term01:~$ pks create-cluster pks-cluster-100 --external-hostname pks-cluster-100 --plan medium --num-nodes 2
```

Name: pks-cluster-100
Plan Name: medium
UUID: 0a1ef756-762b-44e5-9e9a-2b37ad87374c
Last Action: CREATE
Last Action State: in progress
Last Action Description: Creating cluster
Kubernetes Master Host: pks-cluster-100
Kubernetes Master Port: 8443
Worker Nodes: 2
Kubernetes Master IP(s): In Progress
Network Profile Name:

K8s クラスタ テンプレートを選択
例) 開発用
 シングルマスタテンプレート
例) 本番用
 マルチマスタテンプレート

K8s ワーカノード数
を選択

Use 'pks cluster pks-cluster-100' to monitor the state of your cluster

```
vmware@pks01-term01:~$
```

vSphere Client | メニュー | 検索 | Administrator@VSPHERE.LOCAL | +

保護されていない通信 | https://172.18.176.11/ui/#?extensionId=vsphere.core.inventory.serverObjectViewsExtension&objectId=urn:vmomi:Folder:10000000000000000000000000000000

vSphere Client | サマリ | 監視 | 設定 | データセンター | ホストおよびクラスタ | 仮想マシン | データストア | ネットワーク | リンクされた vCenter Server システム

pks01-vc01.vbc.local | アクション | カスタム属性 | タグ

仮想マシン: 38 ホスト: 3

CPU 空き容量: 113.72 GHz
使用済み: 18.1 GHz キャパシティ: 131.82 GHz

メモリ 空き容量: 380.44 GB
使用済み: 387.19 GB キャパシティ: 767.63 GB

ストレージ 空き容量: 8.53 TB
使用済み: 2.39 TB キャパシティ: 10.92 TB

最近のタスク | ターゲット | ステータス | 開始者 | 待ち時間 | 開始時刻 | 完了時刻 | サーバ

すべて | 他のタスク

vSphere Client | メニュー | 検索 | Administrator@VSPHERE.LOCAL | +

保護されていない通信 | https://172.18.176.11/ui/#?extensionId=vsphere.core.inventory.serverObjectViewsExtension&objectId=urn:vmomi:Fo...

vm vSphere Client メニュー 検索 Administrator@VSPHERE.LOCAL

pks01-vc01.vbc.local | アクション

サマリ 監視 設定 データセンター ホストおよびクラスタ 仮想マシン データストア ネットワーク リンクされた vCenter Server システム

仮想マシン: 38 ホスト: 3

CPU 空き容量: 113.72 GHz
使用済み: 18.1 GHz キャパシティ: 131.82 GHz
メモリ 空き容量: 380.44 GB
使用済み: 387.19 GB キャパシティ: 767.63 GB
ストレージ 空き容量: 8.53 TB
使用済み: 2.39 TB キャパシティ: 10.92 TB

カスタム属性

属性	値

タグ

K8s クラスタノードの展開テンプレート (Stemcell) からのクローン

最近のタスク アラーム

タスク名	ターゲット	ステータス	開始者	待ち時間	開始時刻	完了時刻	サーバ
仮想マシンのクローン作成	sc-8835ee81-1d4c-4...	12%	VSPHERE.LOCAL\Admini...	5 ミリ秒	2018/11/10 14:01:00		pks01-vc01.vbc.local
仮想マシンのクローン作成	sc-8835ee81-1d4c-4...	12%	VSPHERE.LOCAL\Admini...	8 ミリ秒	2018/11/10 14:00:59		pks01-vc01.vbc.local
仮想マシンのクローン作成	sc-8835ee81-1d4c-4...	12%	VSPHERE.LOCAL\Admini...	5 ミリ秒	2018/11/10 14:00:59		pks01-vc01.vbc.local
仮想マシンのクローン作成	sc-8835ee81-1d4c-4...	12%	VSPHERE.LOCAL\Admini...	4 ミリ秒	2018/11/10 14:00:59		pks01-vc01.vbc.local

すべて 他のタスク

vSphere Client | メニュー | 検索 | Administrator@VSPHERE.LOCAL | +

保護されていない通信 | https://172.18.176.11/ui/#?extensionId=vsphere.core.inventory.serverObjectViewsExtension&objectId=urn:vmomi:Folder:... | ? | : |

vm vSphere Client メニュー 検索 Administrator@VSPHERE.LOCAL

pks01-vc01.vbc.local | アクション

サマリ 監視 設定 データセンター ホストおよびクラスタ 仮想マシン データストア ネットワーク リンクされた vCenter Server システム

仮想マシン: 38 ホスト: 3

K8s クラスタ
マスタノード
AZ(Availability Zone)
を分散して展開

RP-COMPO1

- vm-091cedd8-ec5d...
- vm-1d0cbcba6-1a25-4...
- vm-652d9233-45bc...
- vm-675c7b55-cd35...
- vm-9075bfbe-dfb2...
- vm-9475dfbd-f4ad...
- vm-a142c107-50d7...
- vm-ac37b7c4-8b37...
- vm-ccf6128a-e16a-4...

RP-COMPO2

- vm-0aca1a5bc-eed2...
- vm-0b4c8660-6e4f...
- vm-0e5229a1-61b6...
- vm-1e0cf39b-b304...
- vm-34457060-1d2a...
- vm-acda09f1-7cba-4...
- vm-c4268496-dc99...

RP-COMPO3

- sc-8835ee81-1d4c-4...
- sc-c026191c-2d9b-4...
- vm-000842b1-e850...
- vm-525da7ee-9057...
- vm-535980d0-e79f...
- vm-764a824d-dc9a...
- vm-7c002450-83ec...

カスタム属性

属性	値
No items to display	

編集...

K8s クラスタ
ワーカノード
AZ(Availability Zone)
を分散して展開

タグ

割り当てられたタグ	カテゴリ	説明
No items to display		

割り当てる 削除...

バージョン情報

バージョン	6.5.0
ビルド	9451637

最近のタスク アラーム

Kubernetes Cluster のスケールアウト

BOSH による環境の管理



コンテナ環境の
構築と運用



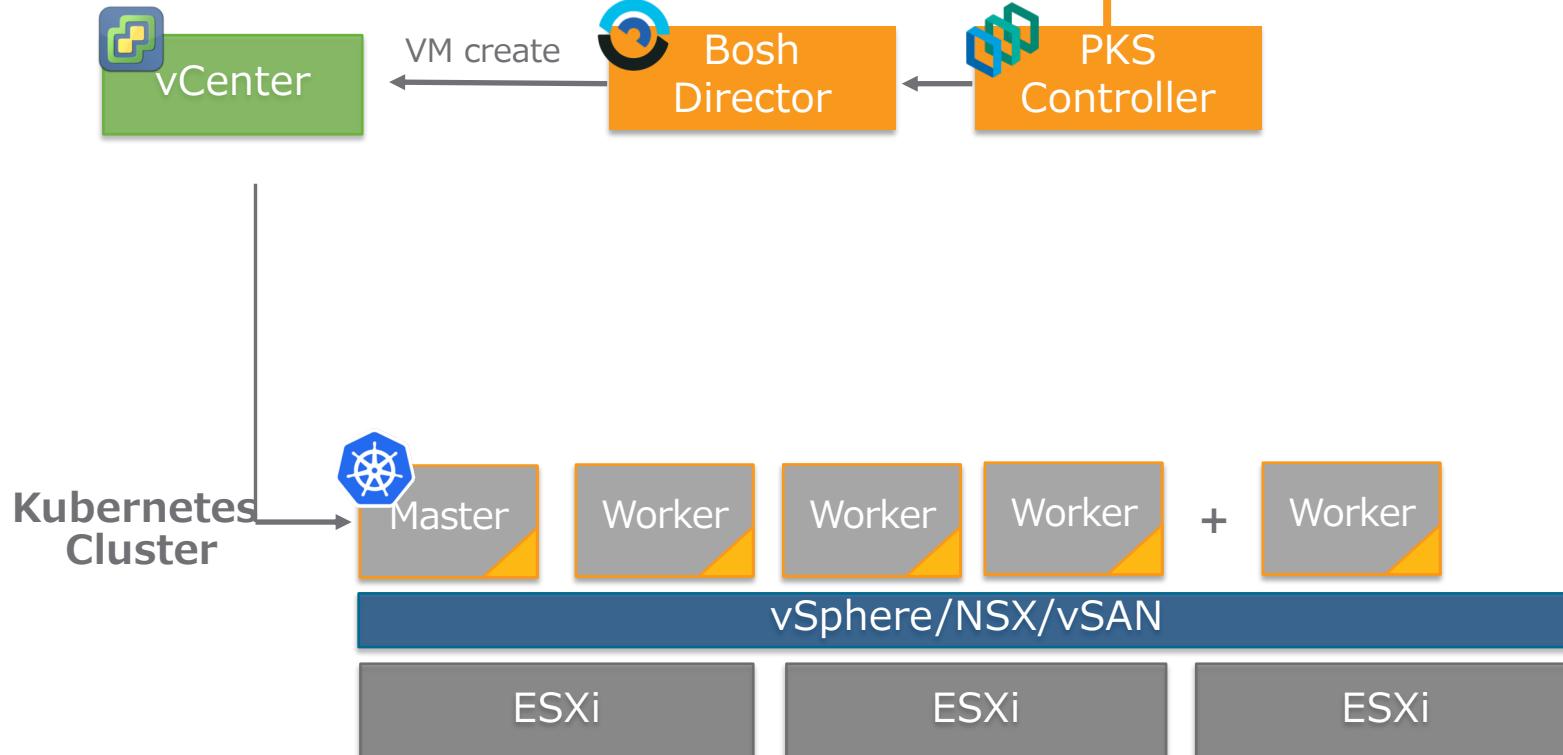
運用管理者 (インフラ)

pks コマンド

Kubernetes cluster (VM) の管理



PKS
Controller



2. vmware@pks01-term01: ~ (ssh)

```
vmware@pks01-term01:~$ pks cluster pks-cluster-100
```

```
Name: pks-cluster-100
Plan Name: medium
UUID: 0a1ef756-762b-44e5-9e9a-2b37ad8737dc
Last Action: CREATE
Last Action State: succeeded
Last Action Description: Instance provisioning completed
Kubernetes Master Host: pks-cluster-100
Kubernetes Master Port: 8443
Worker Nodes: 2
Kubernetes Master IP(s): 172.18.182.1
Network Profile Name: [REDACTED]
```

```
vmware@pks01-term01:~$ pks resize pks-cluster-100 --num-nodes 3
```

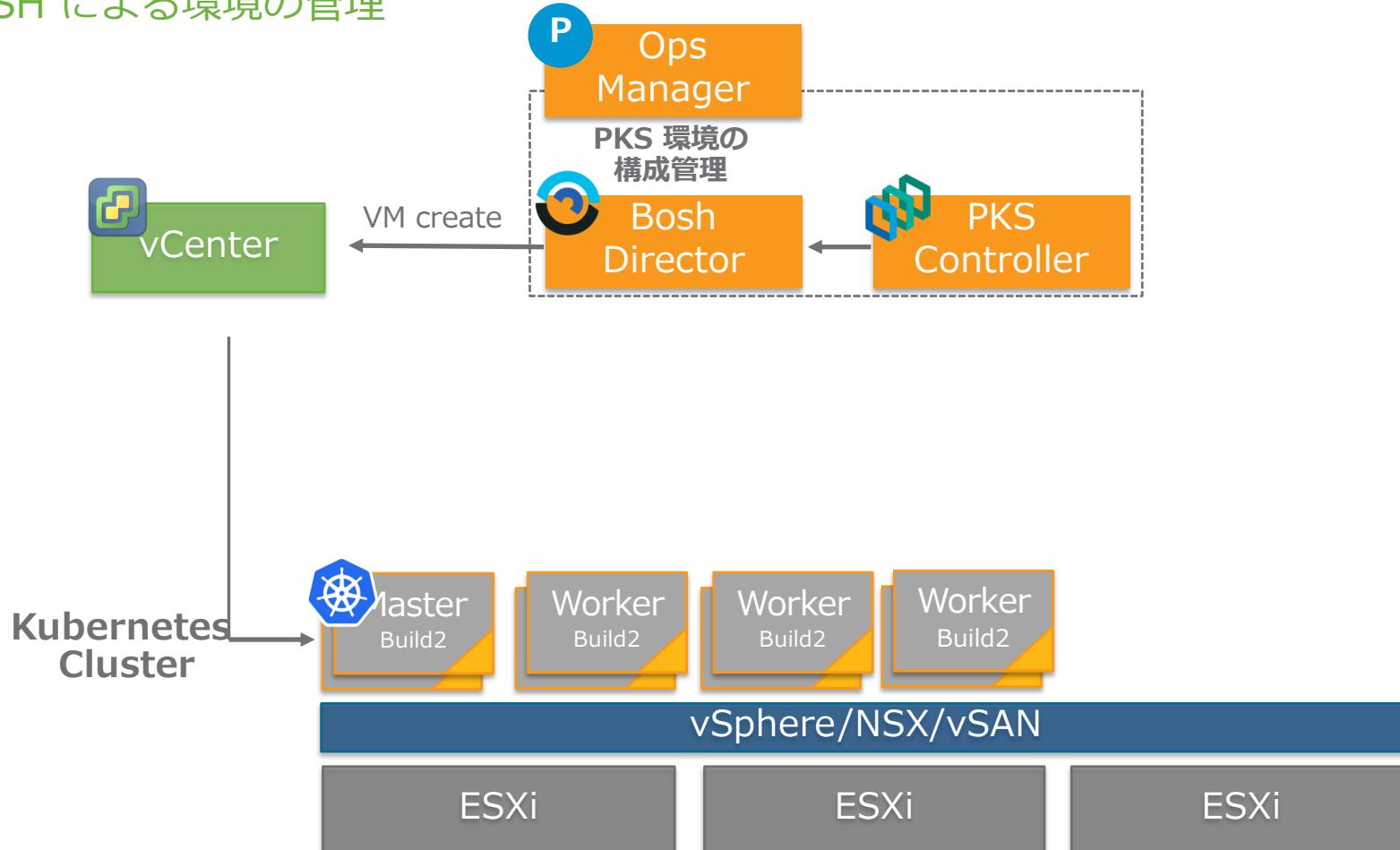
K8s ワーカーノード数の
選択

Kubernetes Cluster のパッチ適用

BOSH による環境の管理



コンテナ環境の
構築と運用



Kubernetes Cluster のパッチ適用

BOSH による環境の管理



コンテナ環境の構築と運用

P

Ops
Manager

PCF Ops Manager

INSTALLATION

テンプレート (Stemcell) のアップロード

vmware ▾

Stemcell Library

Product	Required	Deployed	Staged
Pivotal Container Service Version 1.2.0-build.47	ubuntu-xenial 97.17	97.19	97.19 <input checked="" type="checkbox"/> Latest stemcell.
VMware Harbor Registry Version 1.6.0-build.35	ubuntu-xenial 97	97.19	97.19 <input checked="" type="checkbox"/> Latest stemcell.

IMPORT STEMCELL SAVE

PCF Ops Manager 2.3-build.170; ©2013-2018 Pivotal Software, Inc; All Rights Reserved.

API Docs | End User License Agreement

変更の適用

Service Version 1.2.0-build.47

Depends on
No Dependencies

ERRANDS

VMware Harbor Registry Version 1.6.0-build.35

Depends on
No Dependencies

ERRANDS

APPLY CHANGES

PCF Ops Manager 2.3-build.170; ©2013-2018 Pivotal Software, Inc; All Rights Reserved.

API Docs | End User License Agreement

vmware®

©2018 VMware, Inc.

28

Kubernetes Cluster のパッチ適用

BOSH による環境の管理



コンテナ環境の構築と運用

P

Ops
Manager

P

PCF Ops Manager

INSTALLATION DASHBOARD

STEMCELL LIBRARY

CHANGELOG

Applying Changes

100%

- Installing BOSH
- Uploading runtime config releases to the director
- Updating BOSH director with 2.0 cloud config
- Updating CPI configs
- Updating internal UAA Configuration
- Putting Tile Credentials into CredHub
- Updating runtime config for pivotal-container-service
- Updating runtime config for harbor-container-registry
- Uploading stemcell for Pivotal Container Service
- Uploading releases for Pivotal Container Service
- Migrating credentials to director CredHub
- Installing Pivotal Container Service
- Running errand NSX-T Validation errand for Pivotal Container Service
- Running errand Upgrade all clusters errand for Pivotal Container Service
- Running errand Create pre-defined Wavefront alerts errand for Pivotal Container Service
- Uploading stemcell for VMware Harbor Registry
- Uploading releases for VMware Harbor Registry
- Migrating credentials to director CredHub
- Installing VMware Harbor Registry
- Running errand smoke-testing for VMware Harbor Registry
- Cleaning up BOSH director

Changes Applied

Your changes were successfully applied.
We recommend that you export a backup of this installation from the actions menu.

CLOSE

RETURN TO DASHBOARD

全自動でクラスタの
アップデートが完了

vmware®

©2018 VMware, Inc.

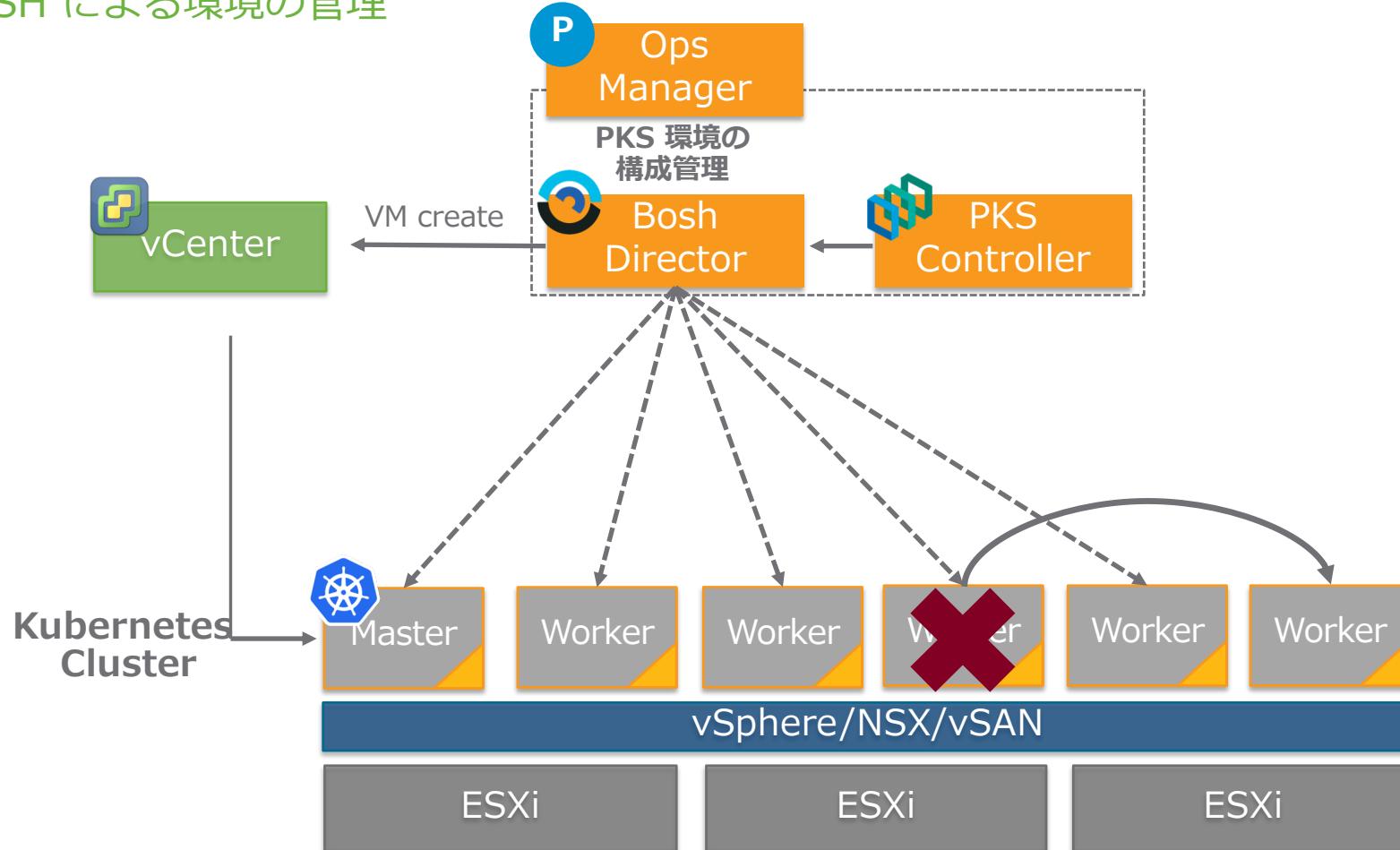
29

Kubernetes Cluster のヘルスモニタ

BOSH による環境の管理



コンテナ環境の
構築と運用



vSphere Client | メニュー | 検索: 7c00 | Administrator@VSPHERE.LOCAL | +

保護されていない通信 | <https://172.18.176.11/ui/#?extensionId=vsphere.core.inventory.serverObjectViewsExtension&objectId=urn:vmomi:Vir...>

vm-7c002450-83ec-4c1c-999a-1f7e17c5af8c | アクション

サマリ 監視 設定 データストア ネットワーク

ゲスト OS: Ubuntu Linux (64-bit)
互換性: ESXi 5.1 以降 (仮想マシン バージョン 9)
VMware Tools: 実行中、バージョン:10304 (管理対象ゲスト)
DNS 名: 5bc35f4c-8a51-46b8-aebc-c52f4abefbce
IP アドレス: 172.18.0.8
ホスト: pks01-esx01.vbc.local

CPU 使用量: 153 MHz
メモリ使用量: 778 MB
ストレージ使用量: 14.45 GB

パワーオン

Web コンソールの起動
Remote Console を起動

仮想マシンのハードウェア

関連オブジェクト

クラスタ CL-PKS

ホスト pks01-esx01.vbc.local

RP-COMP03

pks01-esx01.vbc.local

K8s クラスタ
ワークノードの停止

タグ

注

メモの編集...

カスタム属性

属性	値
compiling	
created_at	2018-11-10T05:35:54Z
deployment	service-instance_Oa1ef756-762b-44e5-9e9a-2b37ad8...
director	p-bosh
id	8b051705-d137-47b4-8cab-a9830d5fb6...

9 items

編集...

最近のタスク アラーム

2. vmware@pks01-term01: ~ (ssh)

```
vmware@pks01-term01:~$ bosh vms -d service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8737dc
Using environment '172.18.176.52' as client 'ops_manager'
```

Task 1092. Done

Deployment 'service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8737dc'

Instance Type	Active	Process	State	AZ	IPs	VM CID	VM
master/9fdbb3a1-f5d2-4e8f-b9dd-85ac58a4a830	true		running	AZ-COMP03	172.18.0.4	vm-525da7ee-9057-4bce-8bc7-4b836e10a0dc	lar
master/ad435b1e-3c4d-4b9c-b2fb-b6b6f1c4436c	true		running	AZ-COMP02	172.18.0.3	vm-1e0cf39b-b30a-45b3-b820-76016251e0f6	lar
master/dba4578a-9c82-4f34-b7d8-7ae2b35dd7a2	true		running	AZ-COMP01	172.18.0.2	vm-1d0cbca6-1a25-4c24-9bf8-9cf317f6a04f	lar
worker/8b051705-d137-47b4-8cab-a9830d5fb630	true		running	AZ-COMP03	172.18.0.8	vm-7c002450-83ec-4c1c-999a-1f7e17c5af8c	med
worker/b3a47409-1e9c-4943-8c61-e6a945aee1f2	true		running	AZ-COMP01	172.18.0.5	vm-9075bfbe-dfb2-433b-9c91-e285b0db522d	med
worker/c0c4f3f8-0ade-4bb3-8e08-ecb89c77d39d	true		running	AZ-COMP02	172.18.0.6	vm-0b4c8660-6e4f-4f00-b87f-d66fecef327	med

6 vms

Succeeded

```
vmware@pks01-term01:~$ bosh vms -d service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8737dc
Using environment '172.18.176.52' as client 'ops_manager'
```

Task 1093

2. vmware@pks01-term01: ~ (ssh)

```
worker/c0c4f3f8-0ade-4bb3-8e08-ecb89c77d39d running AZ-COMP02 172.18.0.6 vm-0b4c8660-6e4f-4f00-b87f-d66fecefc327 medium.disk true
```

6 vms

Succeeded

```
vmware@pks01-term01:~$ bosh vms -d service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8737dc  
Using environment '172.18.176.52' as client 'ops_manager'
```

Task 1093. Done

Deployment 'service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8737dc'

K8s クラスタ
ワーカノードの停止
の検知

Instance	Process State	AZ	IPs	VM CID
master/9fdbb3a1-f5d2-4e8f-b9dd-85ac58a4a830	running	AZ-COMP03	172.18.0.4	vm-525da7ee-9057-4bce-8bc7-4b836e10a0dc
large true				
master/ad435b1e-3c4d-4b9c-b2fb-b6b6f1c4436c	running	AZ-COMP02	172.18.0.3	vm-1e0cf39b-b30a-45b3-b820-76016251e0f6
large true				
master/dba4578a-9c82-4f34-b7d8-7ae2b35dd7a2	running	AZ-COMP01	172.18.0.2	vm-1d0cbca6-1a25-4c24-9bf8-9cf317f6a04f
large true				
worker/8b051705-d137-47b4-8cab-a9830d5fb630	unresponsive agent	AZ-COMP03	172.18.0.8	vm-7c002450-83ec-4c1c-999a-1f7e17c5af8c
medium.disk true				
worker/b3a47409-1e9c-4943-8c61-e6a945aee1f2	running	AZ-COMP01	172.18.0.5	vm-9075bfbe-dfb2-433b-9c91-e285b0db522d
medium.disk true				
worker/c0c4f3f8-0ade-4bb3-8e08-ecb89c77d39d	running	AZ-COMP02	172.18.0.6	vm-0b4c8660-6e4f-4f00-b87f-d66fecefc327
medium.disk true				

6 vms

Succeeded

```
vmware@pks01-term01:~$
```

2. vmware@pks01-term01: ~ (ssh)

```
vmware@pks01-term01:~$ bosh tasks  
Using environment '172.18.176.52' as client 'ops_manager'
```

ID	State	Started At	Last Activity At		User	Deployment
			Description	Result		
1094	processing	Sat Nov 10 05:55:42 UTC 2018	Sat Nov 10 05:55:42 UTC 2018	scan and fix -	health_monitor	service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8737dc

1 tasks

Succeeded

```
vmware@pks01-term01:~$ bosh task 1094
```

K8s クラスタ
へするモニタによる
ワーカノードの停止
の検知と
復旧プロセスの実行

2. vmware@pks01-term01: ~ (ssh)

```
vmware@pks01-term01:~$ bosh vms -d service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8737dc
Using environment '172.18.176.52' as client 'ops_manager'
```

Task 1104. Done

Deployment 'service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8737dc'

Instance Type	Active	Process	State	AZ	IPs	VM CID	VM
master/9fdbb3a1-f5d2-4e8f-b9dd-85ac58a4a830	true		running	AZ-COMP03	172.18.0.4	vm-525da7ee-9057-4bce-8bc7-4b836e10a0dc	large
master/ad435b1e-3c4d-4b9c-b2fb-b6b6f1c4436c	true		running	AZ-COMP02	172.18.0.3	vm-1e0cf39b-b30a-45b3-b820-76016251e0f6	large
master/dba4578a-9c82-4f34-b7d8-7ae2b35dd7a2	true		running	AZ-COMP01	172.18.0.2	vm-1d0cbca6-1a25-4c24-9bf8-9cf317f6a04f	large
worker/8b051705-d137-47b4-8cab-a9830d5fb630	ium.disk true		running	AZ-COMP03	172.18.0.8	vm-432b7a01-6a3e-4162-b44c-f6bb8a2ca43c	medium
worker/b3a47409-1e9c-4943-8c61-e6a945aee1f2	ium.disk true		running	AZ-COMP01	172.18.0.5	vm-9075bfbe-dfb2-433b-9c91-e285b0db522d	medium
worker/c0c4f3f8-0ade-4bb3-8e08-ecb89c77d39d	ium.disk true		running	AZ-COMP02	172.18.0.6	vm-0b4c8660-6e4f-4f00-b87f-d66fecef327	medium

6 vms

Succeeded

```
vmware@pks01-term01:~$
```

K8s クラスタ
ワーカノードの復旧

保護されていない通信 | <https://172.18.176.11/ui/#?extensionId=vsphere.core.inventory.serverObjectViewsExtension&objectId=urn:vmomi:Vir...>

vSphere Client メニュー ▾ メニュー ▾ 7c00

Administrator@VSPHERE.LOCAL ▾

vm-432b7a01-6a3e-4162-b44c-f6bb8a2ca43c アクション ▾

サマリ 監視 設定 権限 データストア ネットワーク

ゲスト OS: Ubuntu Linux (64-bit)
互換性: ESXi 5.1 以降 (仮想マシン バージョン 9)
VMware Tools: 実行中、バージョン:10304 (管理対象ゲスト)
DNS 名: bc055a25-b1bc-4e1a-9e3d-457c097cb4fd
IP アドレス: 172.18.0.8
ホスト: pks01-esx01.vbc.local

CPU 使用量 1.19 GHz
メモリ使用量 2.88 GB
ストレージ使用量 14.46 GB

K8s クラスタ
ワークノードの復旧

属性	値
compiling	
created_at	2018-11-10T05:57:30Z
deployment	service-instance_0a1ef756-762b-44e5-9e9a-2b37ad8...
director	p-bosh
id	8b051705-d137-47b4-8cab-a9830d5fb6...

タグ

最近のタスク アラーム

(参考) マルチテナント環境の考え方

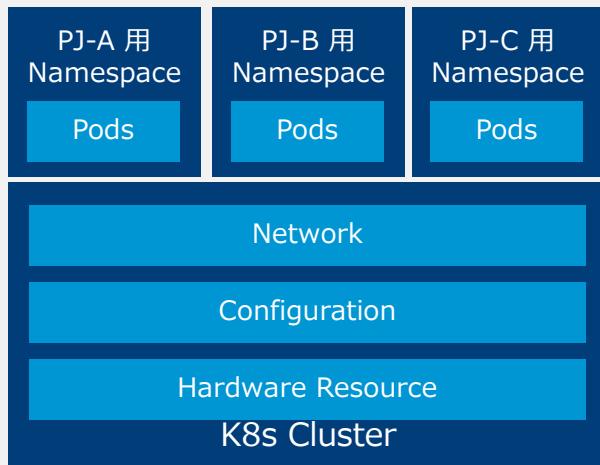
ソフトマルチテナントに加えハードマルチテナント構成が容易



コンテナ環境の
構築と運用

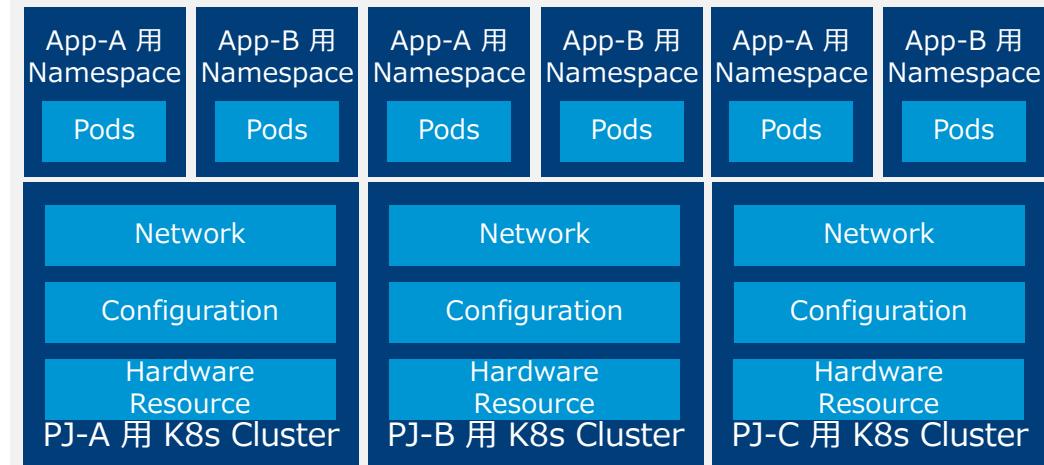
ソフトマルチテナント

K8s クラスタを
テナント間で共有するアプローチ



ハードマルチテナント

K8s クラスタを
テナント単位で分離するアプローチ



テクノロジー視点での課題

サービス提供部門からの要望



コンテナ環境の構築と運用

クラスターの払い出し
パッチ適用
障害の検知と復旧



インフラ構成の自動化

コンテナへのネットワーク/ストレージサービスの提供



セキュリティ

コンテナ間や既存サーバとのネットワーク通信におけるセキュリティ



運用監視

コンテナ環境とコンテナ内アプリの運用監視



インフラ構成の
自動化

インフラリソースの提供

ネットワーク/ストレージインフラの考慮点

物理機器ベースのインフラ

外部にサービスを
公開できますか？

ネットワーク
ストレージを
どう設計すれば
いいのだろう、、

ネットワーク
ロードバランサの
設定をして、、



Linux Linux Linux



、、



ストレージの
ボリュームを切
り出して、、



vmware®

©2018 VMware, Inc.

ソフトウェアベースのインフラ

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

、、

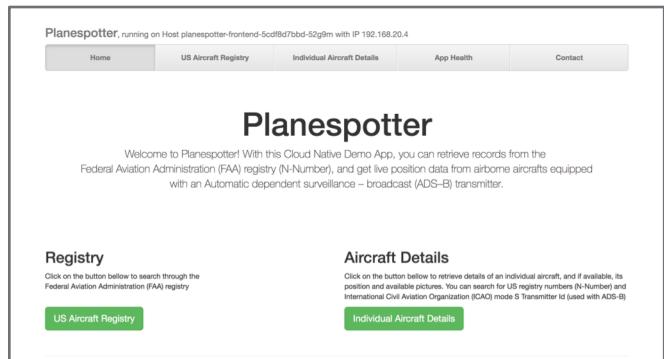
、、

、、

、、

、、</

サンプルアプリケーション構成



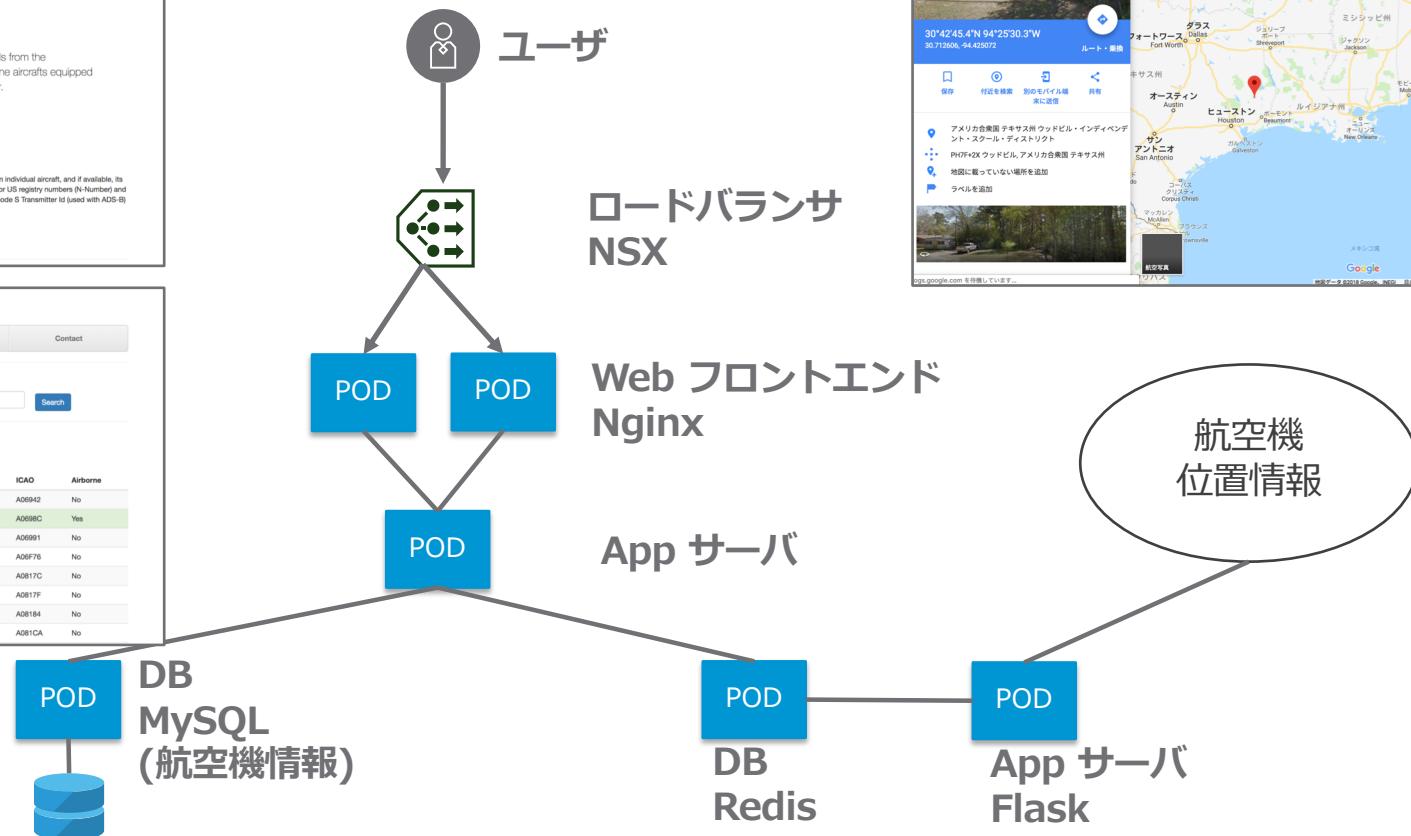
Planespotter, running on Host planespotter-frontend-5cdff8d7bbd-52gfm with IP 192.168.20.4

Home **US Aircraft Registry** **Individual Aircraft Details** **App Health** **Contact**

Registration Id Owner Name Manufacturer Aircraft Model ICAO Search

displaying 21 - 30 Aircrafts in total 754

N-Number	Owner	City	State	Manufacturer	Model	ICAO	Airborne
N-12540	UNITED AIRLINES INC	CHICAGO	IL	EMBRAER	E175LR	A06942	No
N-12564	UNITED AIRLINES INC	CHICAGO	IL	EMBRAER	E175LR	A0698C	Yes
N-12569	UNITED AIRLINES INC	CHICAGO	IL	EMBRAER	E175LR	A06991	No
N-1270A	UNITED AIRLINES INC	CHICAGO	IL	BOEING	737-800	A06F76	No
N-13110	UNITED AIRLINES INC	CHICAGO	IL	BOEING	737-800	A0617C	No
N-13113	UNITED AIRLINES INC	CHICAGO	IL	BOEING	737-800	A0617F	No
N-13118	UNITED AIRLINES INC	CHICAGO	IL	EMBRAER	E175XR	A06184	No
N-13138	UNITED AIRLINES INC	CHICAGO	IL	BOEING	737-800	A061CA	No



コンテナの展開 -POD の展開

Kubernetes による環境の管理

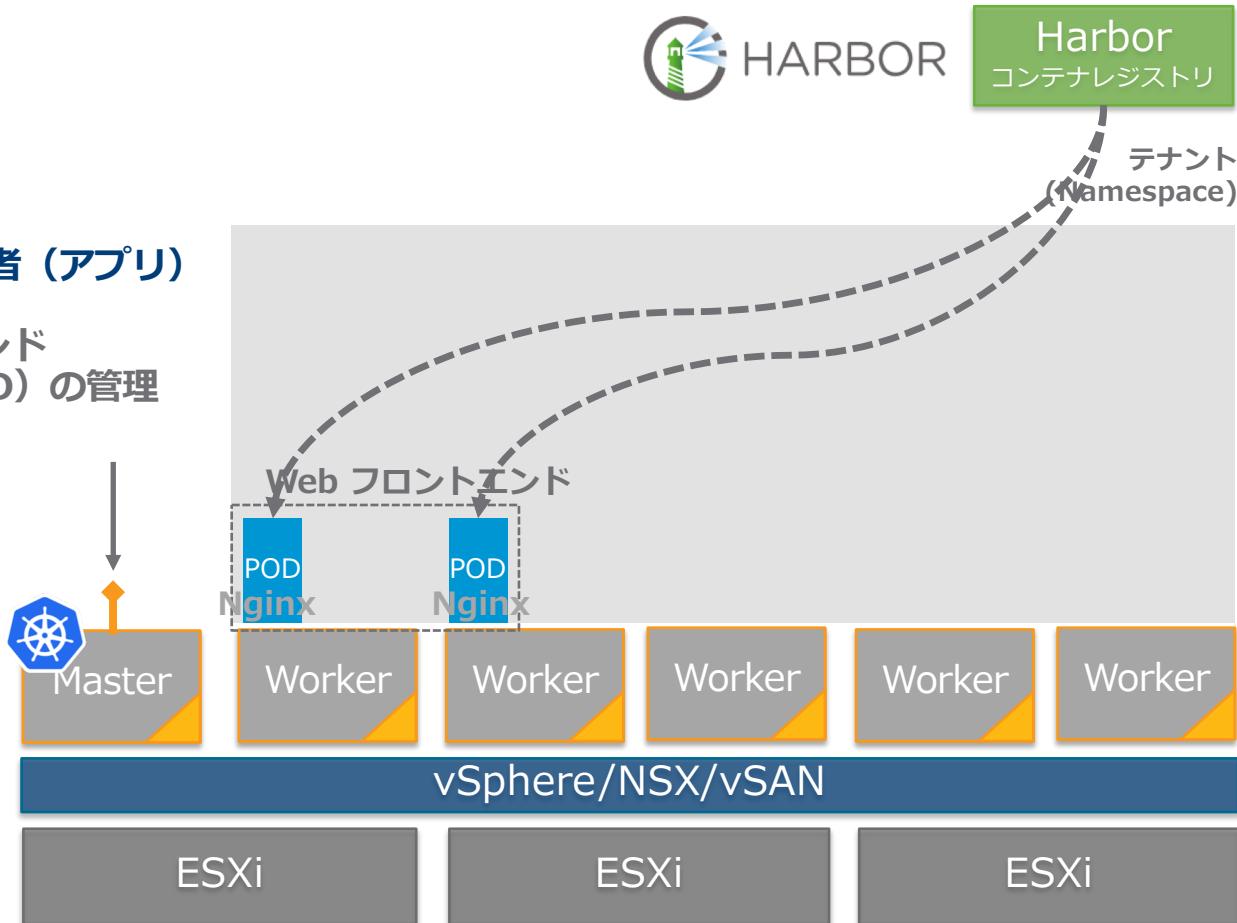


インフラ構成の
自動化



運用管理者（アプリ）

kubectl コマンド
コンテナ（POD）の管理



```
2. vmware@pks01-term01:~/plane (ssh)
vmware@pks01-term01:~/plane$ cat frontend.yaml
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: planespotter-frontend
  namespace: planespotter
  labels:
    app: planespotter-frontend
    tier: frontend
spec:
  replicas: 2
  selector:
    matchLabels:
      app: planespotter-frontend
  template:
    metadata:
      labels:
        app: planespotter-frontend
        tier: frontend
    spec:
      containers:
        - name: planespotter-frontend
          image: pks01-pv-harb01.vbc.local/plane/planespotter-frontend:v1
          imagePullPolicy: IfNotPresent
          env:
            - name: PLANESPOTTER_API_ENDPOINT
              value: planespotter-svc
vmware@pks01-term01:~/plane$ kubectl apply -f frontend.yaml
```

オブジェクトのタイプ

展開するコンテナ
(POD) の数

使用するコンテナの
イメージ

← → ⌘ ⌘ ⌘ 保護されていない通信 | <https://172.18.176.54/harbor/sign-in?signout=true>



Harbor

Search Harbor...

English ▾ About

Harbor

Username

Password

Remember me

[Forgot password](#)

LOG IN

[Sign up for an account](#)

Popular Repositories

Name	Tags	Pulls
library/redis	1	1
plane/planespotter-app-server	1	3
library/gb-redisslave	1	2
plane/redis	1	3
plane/mysql	1	3

1 - 5 of 8 items [1](#) [2](#) [>](#)

保護されていない通信 | <https://172.18.176.54/harbor/projects/3/repositories/plane%2Fplanespotter-frontend>

Harbor English admin

Projects Logs Administration Users Registries Replications Configuration

Projects < Repositories plane/planespotter-frontend

Info Images

SCAN COPY DIGEST + ADD LABELS X DE

Tag Size Pull Command Vulnerability

v1 271.79MB

No vulnerability
0 of 199 packages has known vulnerability.
199 None
Scan completed time: 11/10/2018 09:00:54

コンテナイメージの脆弱性スキャン

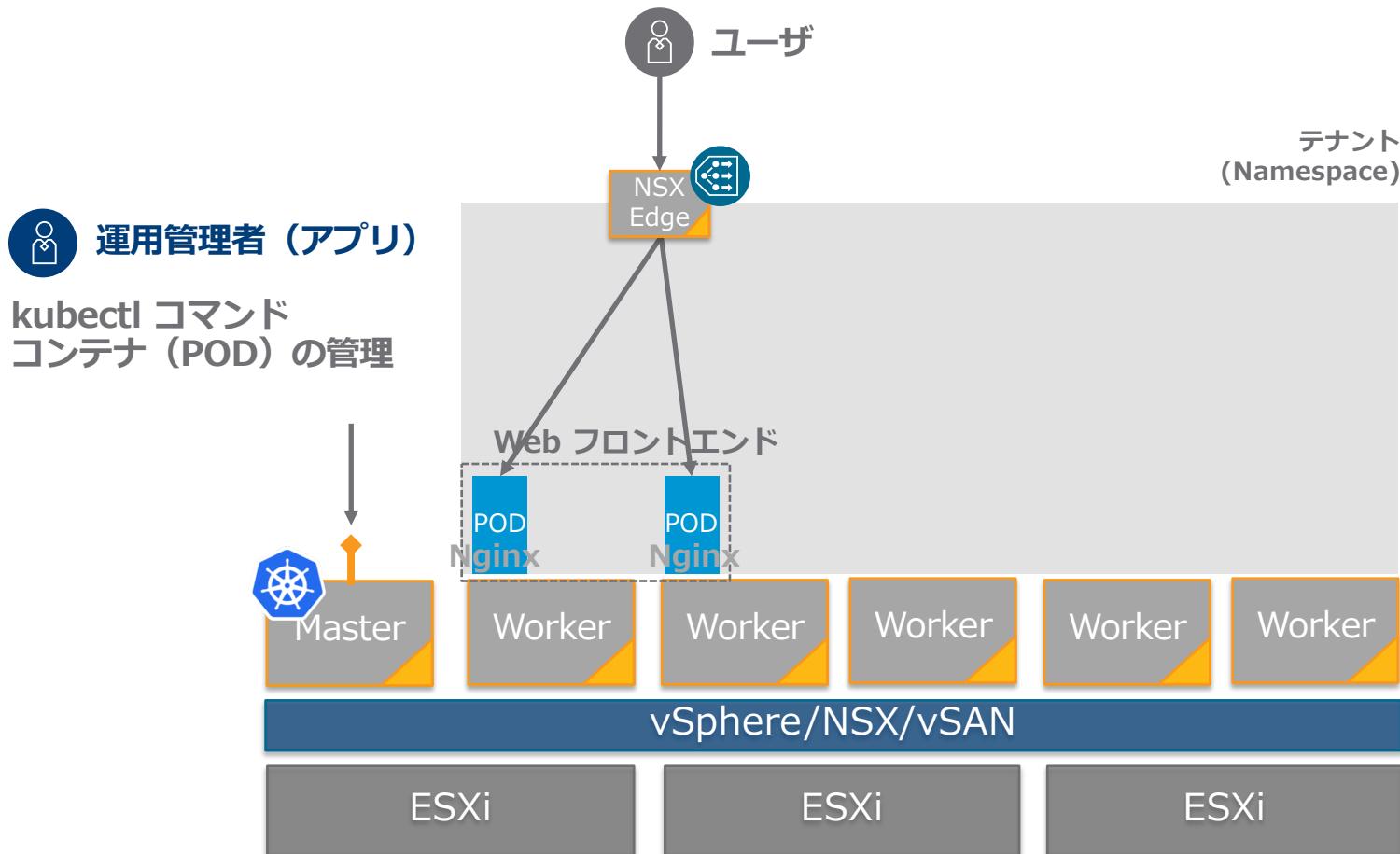
EVENT

コンテナの展開 - ロードバランサの提供

Kubernetes による環境の管理



インフラ構成の
自動化



2. vmware@pks01-term01: ~/plane (ssh)

```
vmware@pks01-term01:~/plane$ cat frontend-service.yaml
apiVersion: v1
kind: Service
metadata:
  namespace: planespotter
  name: planespotter-frontend
  labels:
    app: planespotter-frontend
spec:
  ports:
    # the port that this service should serve
    - port: 80
  selector:
    app: planespotter-frontend
  type: LoadBalancer
vmware@pks01-term01:~/plane$
```

オブジェクトのタイプ

サービスのタイプ

2. vmware@pks01-term01: ~/plane (ssh)

```
vmware@pks01-term01:~/plane$ cat frontend-service.yaml
```

```
apiVersion: v1
kind: Service
metadata:
  namespace: planespotter
  name: planespotter-frontend
  labels:
    app: planespotter-frontend
spec:
```

```
  ports:
    # the port that this service should serve on
    - port: 80
```

```
  selector:
    app: planespotter-frontend
  type: LoadBalancer
```

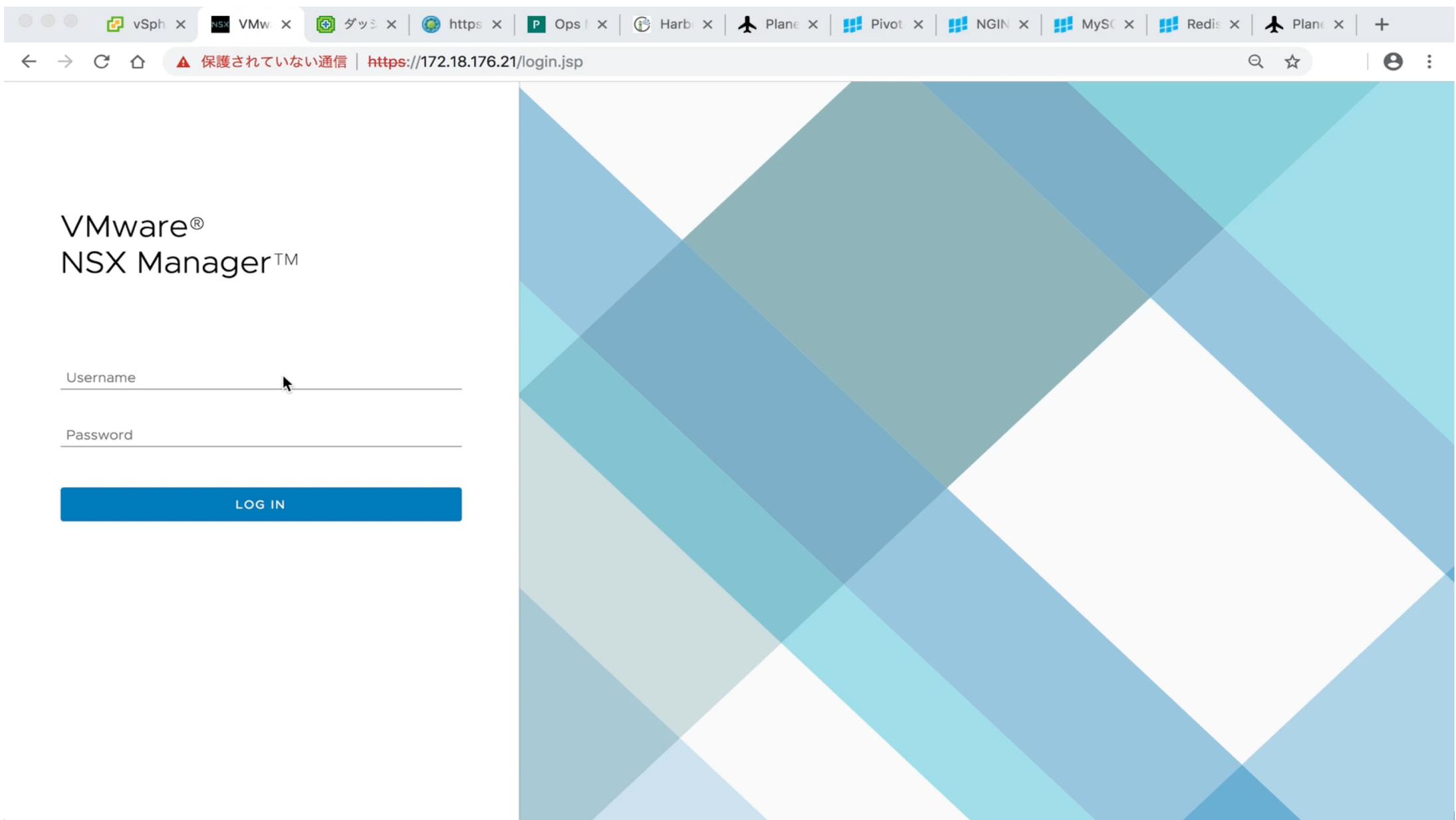
```
vmware@pks01-term01:~/plane$ kubectl apply -f frontend-service.yaml
```

```
service/planespotter-frontend created
```

```
vmware@pks01-term01:~/plane$ kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
planespotter-frontend	LoadBalancer	10.100.200.78	100.64.0.47,172.18.182.69	80:30573/TCP	6s

```
vmware@pks01-term01:~/plane$ █
```



保護されていない通信 | <https://172.18.176.21/nsxmanager/index.html#/nsx/lb/virtualservers>

NSX

ロード バランシング

ロード バランサ 仮想サーバ Server Pools Profiles Monitors

+ ADD EDIT CLONE DELETE ACTIONS

名前	プロトコル	IP アドレス: ポート	デフォルトのサーバ プール	ロード バランサ
lb-pks-9d905734-286f-44d5-82ba-b508ef09da85-virtual-se...	TCP	172.18.182.28:84...	lb-pks-9d905734-286f-44d5-82ba-b508ef09da85-lb-pool	lb-pks-
pks-0a1ef756-762b-44e5-9e9a-2b37ad8737dc-http	TCP	172.18.182.2:80		lb-pks-
pks-0a1ef756-762b-44e5-9e9a-2b37ad8737dc-https_termin...	TCP	172.18.182.2:443		lb-pks-
pks-54027c5c-aa20-470d-be3f-68d9d74985bb-http	TCP	172.18.182.22:80		lb-pks-
pks-54027c5c-aa20-470d-be3f-68d9d74985bb-https_termin...	TCP	172.18.182.22:443		lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-default-nginx...	TCP	172.18.182.27:80	pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-default-nginx-wavefront-80	lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-https_termin...	TCP	172.18.182.8:80		lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-planespotter-...	TCP	172.18.182.8:443		lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-planespotter-frontend-80	TCP	172.18.182.20:80	pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-planespotter-planespotter-frontend-80	lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-sock-shop-front-end-80	TCP	172.18.182.18:80	pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-sock-shop-front-end-80	lb-pks-
pks-9d905734-286f-44d5-82ba-b508ef09da85-https_termin...	TCP	172.18.182.29:80		lb-pks-
pks-9d905734-286f-44d5-82ba-b508ef09da85-https_termin...	TCP	172.18.182.29:443		lb-pks-
pks-9d905734-286f-44d5-82ba-b508ef09da85-planespotter...	TCP	172.18.182.69:80	pks-9d905734-286f-44d5-82ba-b508ef09da85-planespotter-planespotter-frontend-80	lb-pks-

ロード バランサ VIP の構成

COLUMNS REFRESH Last Updated: Just Now

NSX

ロード バランシング

ロード バランサ 仮想サーバ Server Pools Profiles Monitors

+ ADD EDIT CLONE DELETE ACTIONS

<input type="checkbox"/> 名前	プロトコル	IP アドレス: ポート	デフォルトのサーバ プール	ロード バランサ
lb-pks-9d905734-286f-44d5-82ba-b508ef09da85-virtual-se...	TCP	172.18.182.28:84...	lb-pks-9d905734-286f-44d5-82ba-b508ef09da85-lb-pool	lb-pks-
pks-0a1ef756-762b-44e5-9e9a-2b37ad8737dc-http	TCP	172.18.182.2:80		lb-pks-
pks-0a1ef756-762b-44e5-9e9a-2b37ad8737dc-https_termin...	TCP	172.18.182.2:443		lb-pks-
pks-54027c5c-aa20-470d-be3f-68d9d74985bb-http	TCP	172.18.182.22:80		lb-pks-
pks-54027c5c-aa20-470d-be3f-68d9d74985bb-https_termin...	TCP	172.18.182.22:443		lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-default-nginx...	TCP	172.18.182.27:80	pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-default-nginx-wavefront-80	lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-http	TCP	172.18.182.8:80		lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-https_termin...	TCP	172.18.182.8:443		lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-planespotter-...	TCP	172.18.182.20:80	pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-planespotter-planespotter-frontend-80	lb-pks-
pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-sock-shop-fr...	TCP	172.18.182.18:80	pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-sock-shop-front-end-80	lb-pks-
pks-9d905734-286f-44d5-82ba-b508ef09da85-http	TCP	172.18.182.29:80		lb-pks-
pks-9d905734-286f-44d5-82ba-b508ef09da85-https_termin...	TCP	172.18.182.29:443		lb-pks-
pks-9d905734-286f-44d5-82ba-b508ef09da85-planespotte...	TCP	172.18.182.69:80	pks-9d905734-286f-44d5-82ba-b508ef09da85-planespotter-planespotter-frontend-80	lb-pks-

COLUMNS REFRESH Last Updated: Just Now

保護されていない通信 | <https://172.18.176.21/nsxmanager/index.html#/nsx/lb/serverpools/f54fcc32-501f-450f-9f46-1ae3b377ca16/pool-m...>

NSX

ロード バランシング

ロード バランサ 仮想サーバ Server Pools Profiles Monitors

+ ⌂ ⌂ ⌂ ⌂

Name
<input type="checkbox"/> lb-pks-0a1ef756-7
<input type="checkbox"/> lb-pks-54027c5c-
<input type="checkbox"/> lb-pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-lb-pool
<input type="checkbox"/> lb-pks-9d905734-286f-44d5-82ba-b508ef09da85-lb-pool
<input type="checkbox"/> pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-default-nginx...
<input type="checkbox"/> pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-planespotter-...
<input type="checkbox"/> pks-6ad8b9d6-868b-435a-b467-cc783c1fc36c-planespotter-...
<input checked="" type="checkbox"/> pks-9d905734-286f-44d5-82ba-b508ef09da85-planespot...

ロードバランサ
Pool Member の構成

pks-9d905734-286f-44d5-82ba-b508ef09da85-planespotter-planespotter-frontend-80

Overview Virtual Servers Pool Members **Pool Member Statistics**

Display Statistics from Load Balancer [lb-pks-9d905734-286f-44d5-82ba-b508ef09da85](#)

IP:Port	Status	Current Sessions	Max Sessions	Bytes in	Bytes out	Http Request Rate
192.168.20.3:80	↑ UP	0	7	301027	18195	0
192.168.20.4:80	↑ UP	0	6	565736	32946	0

COLUMNS REFRESH Last Updated: Just Now 2 Pool Members

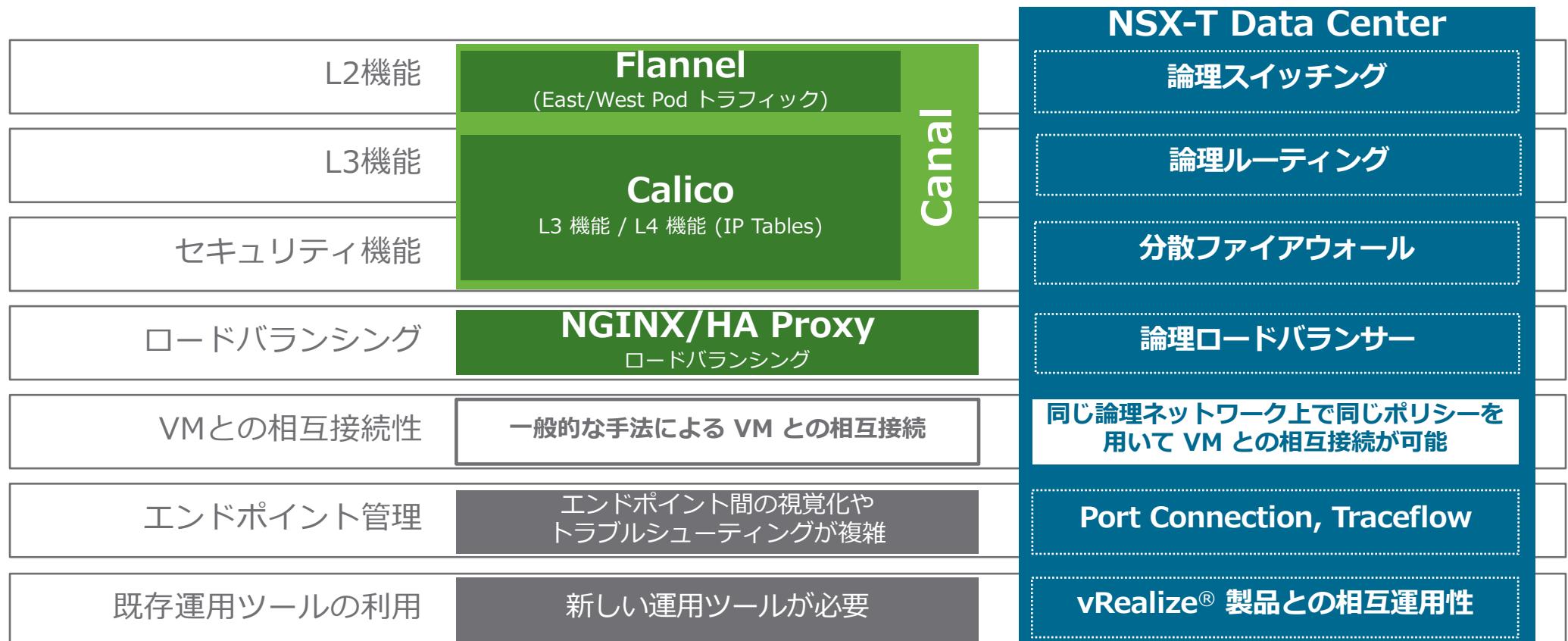
Pools used in Layer4 VIP do not have HTTP Requests counters. Pools used in Layer7 VIP do not have Packets counters

(参考) Kubernetes 環境のネットワーク

NSX-T Data Center が提供する包括的なネットワークスタック

オープンソーススタックの例

VMware

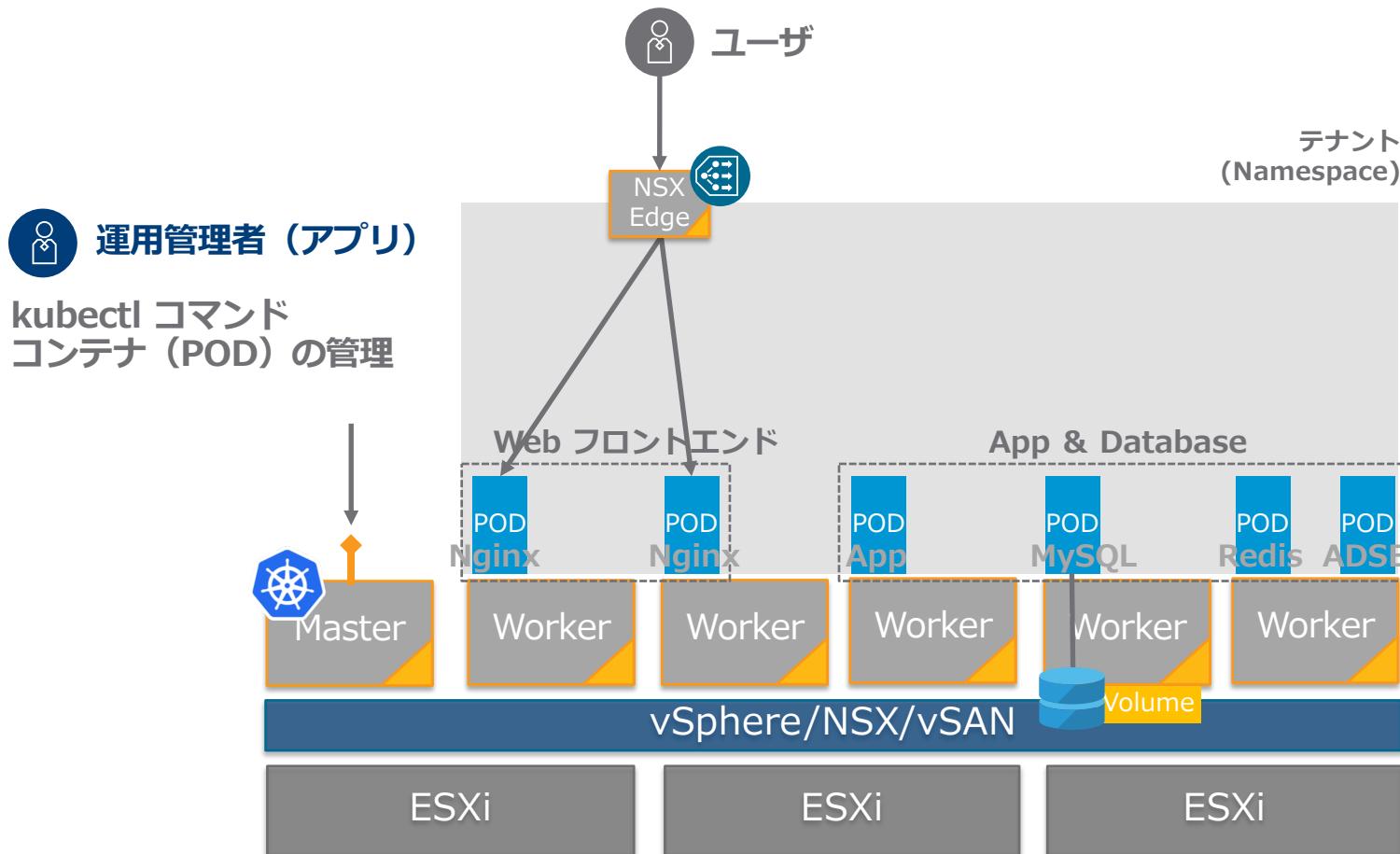


コンテナの展開 - 永続性ストレージの提供

Kubernetes による環境の管理



インフラ構成の
自動化



2. vmware@pks01-term01: ~/plane (ssh)

```
vmware@pks01-term01:~/plane$ cat vsan-sc.yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: thin-disk
provisioner: kubernetes.io/vsphere-volume
parameters:
  diskformat: thin
vmware@pks01-term01:~/plane$
```

オブジェクトのタイプ

ストレージの特性

```
2. vmware@pks01-term01: ~/plane (ssh)
chmod +x create-planespotter-db.sh
./create-planespotter-db.sh
fi
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-start-map
  namespace: planespotter
data:
  mysql-start.sh: |
    #!/bin/sh
    echo "starting planespotter-installer in background"
    /bin/planespotter-install.sh &
    echo "starting mysqld.."
    /entrypoint.sh mysqld
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mysql-vol
  namespace: planespotter
  annotations:
    volume.beta.kubernetes.io/storage-class: thin-disk
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
vmware@pks01-term01:~/plane$ kubectl apply -f mysql-single.yaml
```

オブジェクトのタイプ

ストレージクラスの指定

ストレージの容量の指定

2. vmware@pks01-term01: ~/plane (ssh)

```
chmod +x create-planespotter-db.sh
./create-planespotter-db.sh
fi
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-start-map
  namespace: planespotter
data:
  mysql-start.sh: |
    #!/bin/sh
    echo "starting planespotter-installer in background"
    /bin/planespotter-install.sh &
    echo "starting mysqld.."
    /entrypoint.sh mysqld
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mysql-vol
  namespace: planespotter
  annotations:
    volume.beta.kubernetes.io/storage-class: thin-disk
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

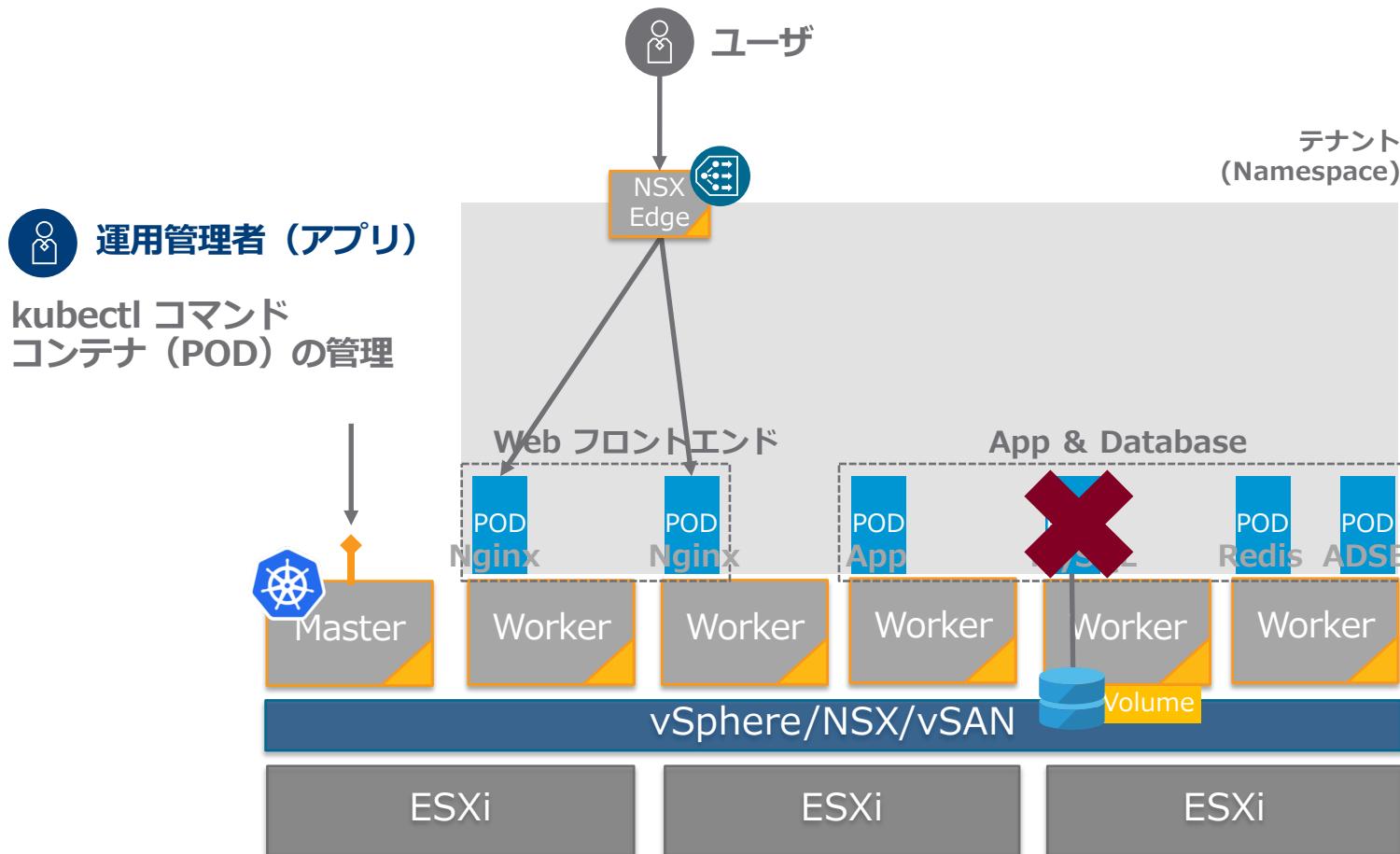
vmware@pks01-term01:~/plane\$ kubectl apply -f mysql-single.yaml

コンテナの展開 - 永続性ストレージの提供

Kubernetes による環境の管理



インフラ構成の
自動化



2. vmware@pks01-term01: ~/plane (ssh)

```
vmware@pks01-term01:~/plane$ kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
adsb-sync-547b7f6db8-lw2mm   1/1     Running   0          7m
mysql-0               1/1     Running   0          1m
planespotter-app-7794c7db85-rscjx   1/1     Running   0          7m
planespotter-frontend-5cdf8d7bbd-4sgfr   1/1     Running   0          31m
planespotter-frontend-5cdf8d7bbd-52g9m   1/1     Running   0          31m
planespotter-frontend-5cdf8d7bbd-pwgjt   1/1     Running   0          14m
redis-server-7bcfbcdf44-n282b      1/1     Running   0          7m
vmware@pks01-term01:~/plane$ kubectl delete pod mysql-0
pod "mysql-0" deleted
```

Pod(Mysql) の削除

2. vmware@pks01-term01: ~/plane (ssh)

```
vmware@pks01-term01:~/plane$ kubectl get pod
NAME                               READY   STATUS    RESTARTS   AGE
adsb-sync-547b7f6db8-lw2mm        1/1     Running   0          7m
mysql-0                            1/1     Running   0          1m
planespotter-app-7794c7db85-rscjx  1/1     Running   0          7m
planespotter-frontend-5cdf8d7bbd-4sgfr 1/1     Running   0          31m
planespotter-frontend-5cdf8d7bbd-52g9m 1/1     Running   0          31m
planespotter-frontend-5cdf8d7bbd-pwgjt 1/1     Running   0          14m
redis-server-7bcfbcdf44-n282b      1/1     Running   0          7m
```

```
vmware@pks01-term01:~/plane$ kubectl delete pod mysql-0
```

```
pod "mysql-0" deleted
```

```
vmware@pks01-term01:~/plane$ kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
adsb-sync-547b7f6db8-lw2mm	1/1	Running	0	7m
mysql-0	0/1	ContainerCreating	0	2s
planespotter-app-7794c7db85-rscjx	1/1	Running	0	8m
planespotter-frontend-5cdf8d7bbd-4sgfr	1/1	Running	0	31m
planespotter-frontend-5cdf8d7bbd-52g9m	1/1	Running	0	31m
planespotter-frontend-5cdf8d7bbd-pwgjt	1/1	Running	0	15m
redis-server-7bcfbcdf44-n282b	1/1	Running	0	7m

K8s ヘルスモニタによる
Pod(Mysql) の復旧

Planespotter, running on Host planespotter-frontend-5cdf8d7bbd-52g9m with IP 192.168.20.4

Home

US Aircraft Registry

Individual Aircraft Details

App Health

Contact

App Healthcheck

✓ App Server connectivity success

⚠ Database connectivity failure

✓ Position Server connectivity success

✓ Picture Server connectivity success

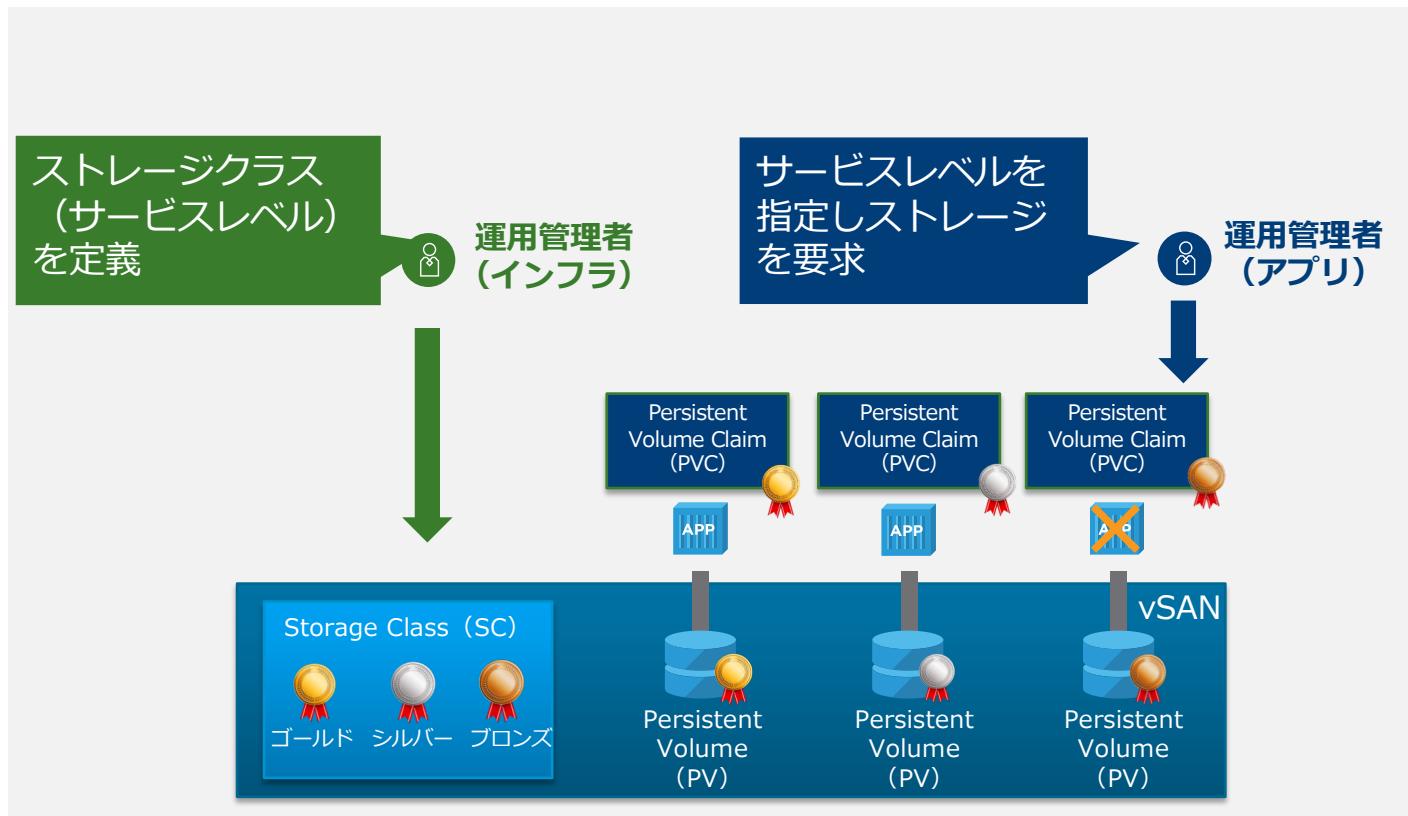
✓ Redis Server connectivity success



インフラ構成の
自動化

(参考) 永続性ストレージ

ストレージ仮想化 (vSAN) による永続性ストレージの提供



コンテナを削除しても消えない
永続性ストレージの提供

- サービスレベルに応じた **ストレージクラス**をサービス展開時に適用
- ストレージ仮想化環境と **同じ運用手法**を踏襲
- 仮想マシン-コンテナ間で **一貫したポリシー**の適用

テクノロジー視点での課題

サービス提供部門からの要望



コンテナ環境の構築と運用

クラスターの払い出し
パッチ適用
障害の検知と復旧



インフラ構成の自動化

コンテナへのネットワーク/ストレージサービスの提供



セキュリティ

コンテナ間や既存サーバとのネットワーク通信におけるセキュリティ



運用監視

コンテナ環境とコンテナ内アプリの運用監視

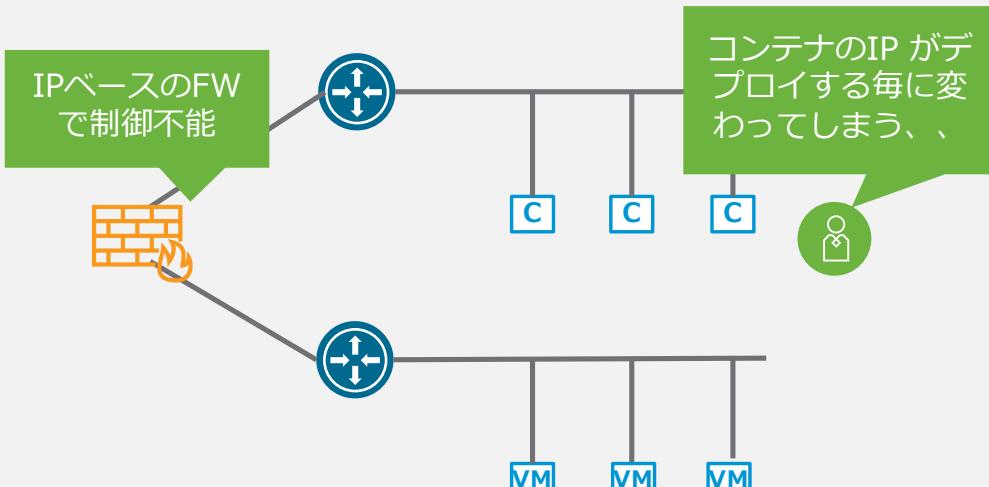
セキュリティ

既存サービスとのセキュアな連携

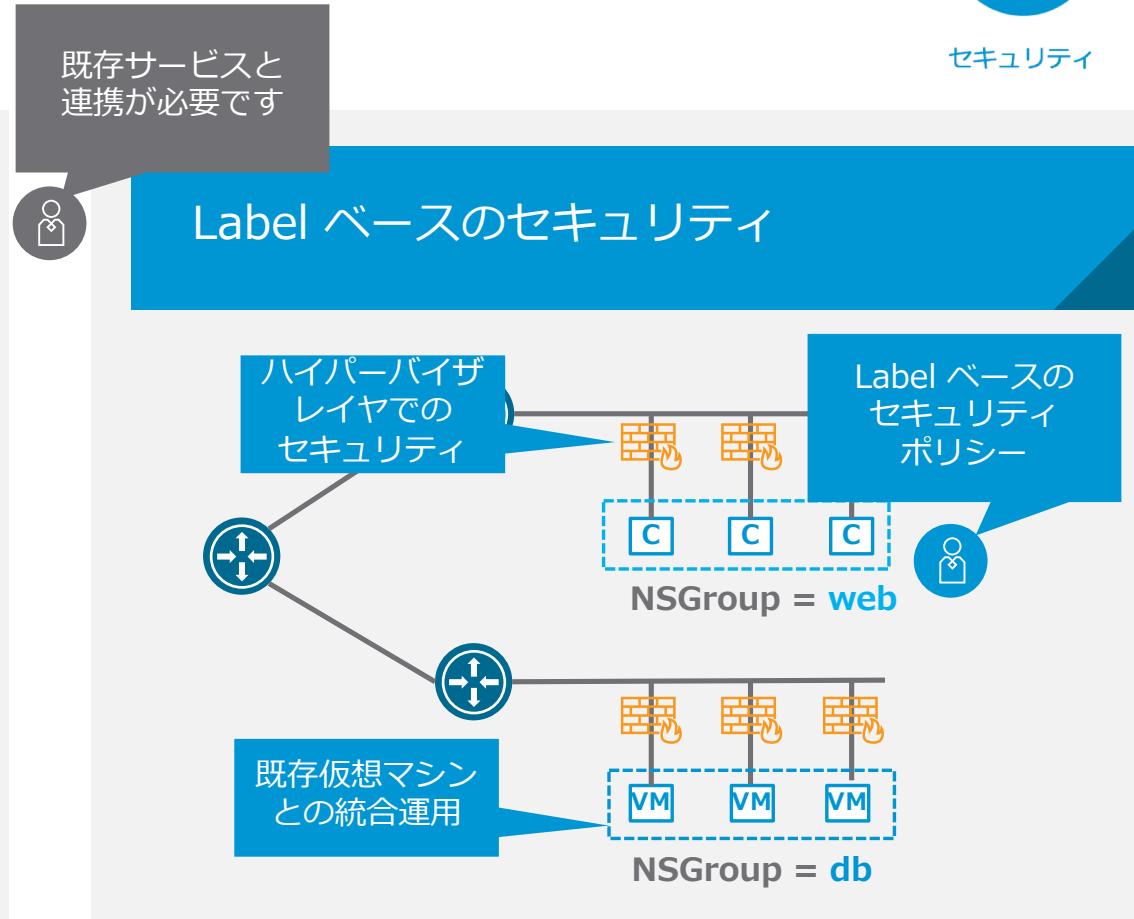


セキュリティ

IP ベースのセキュリティ



Label ベースのセキュリティ



セキュリティ

既存サービスとのセキュアな連携



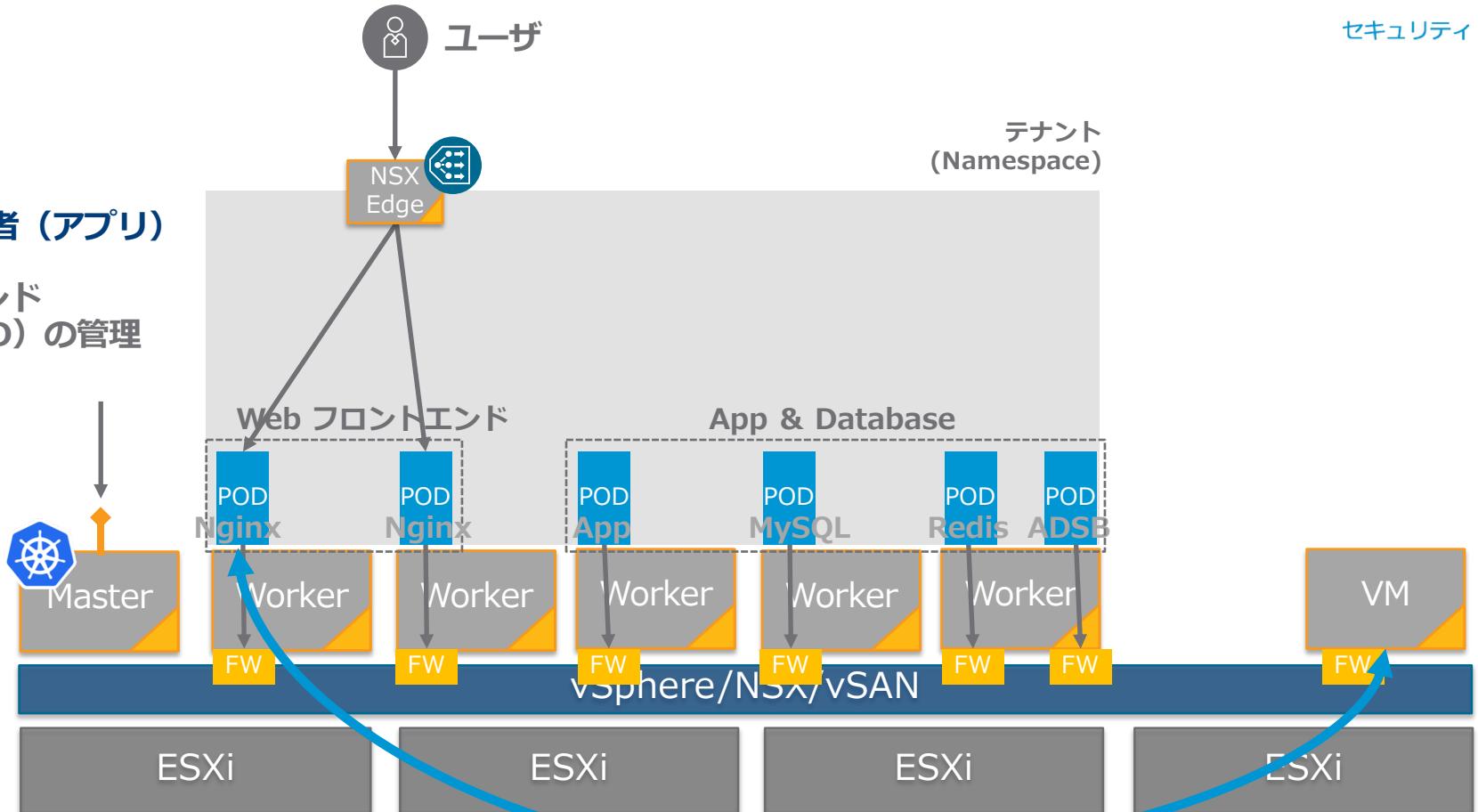
セキュリティ



運用管理者（アプリ）



kubectl コマンド
コンテナ（POD）の管理



2. vmware@pks01-term01: ~/plane (ssh)							
NAME	NOMINATED NODE	READY	STATUS	RESTARTS	AGE	IP	NODE
adsb-sync-547b7f6d1e49e	<none>	1/1	Running	0	17m	192.168.20.7	72c1a859-87ee-4882-989e-dba487
mysql-0	17c6d0	1/1	Running	0	9m	192.168.20.9	cf354719-81ca-4ceb-9496-08d43a
planespotter-app-d1e49e	<none>	1/1	Running	0	17m	192.168.20.6	72c1a859-87ee-4882-989e-dba487
planespotter-frontend-5cdf8d7bbd-4sgfr	d1e49e	1/1	Running	0	41m	192.168.20.3	72c1a859-87ee-4882-989e-dba487
planespotter-frontend-5cdf8d7bbd-52g9m	17c6d0	1/1	Running	0	41m	192.168.20.4	cf354719-81ca-4ceb-9496-08d43a
planespotter-frontend-5cdf8d7bbd-pwgjt54af87	<none>	1/1	Running	0	24m	192.168.20.2	7c622885-8ce7-4723-bb55-aa3868
redis-server-7bcfbcd44-n282b	17c6d0	1/1	Running			192.168.20.8	cf354719-81ca-4ceb-9496-08d43a
vmware@pks01-term01: ~/plane\$ kubectl get pod --show-labels							
NAME		READY	STATUS	RESTARTS	AGE	LABELS	
adsb-sync-547b7f6d1e49e	Web フロントエンド	1/1	Running	0	17m	app=adsb-sync,pod-template-hash=1036392864	
mysql-0	75584-statefulset	1/1	Running	0	9m	app=mysql,controller-revision-hash=mysql-77b5	
planespotter-app-7754c7ab85-1scjx	planespott	1/1	Running	0	17m	app=planespotter,pod-template-hash=3350738641	
planespotter-frontend-5cdf8d7bbd-4sgfr	789483668,tier=frontend	1/1	Running	0	41m	app=planespotter-frontend,pod-template-hash=1	
planespotter-frontend-5cdf8d7bbd-52g9m	789483668,tier=frontend	1/1	Running	0	41m	app=planespotter-frontend,pod-template-hash=1	
planespotter-frontend-5cdf8d7bbd-pwgjt789483668,tier=frontend		1/1	Running	0	25m	app=planespotter-frontend,pod-template-hash=1	
redis-server-7bcfbcd44-n282b,tier=backend		1/1	Running			app=redis-server,pod-template-hash=3679678900	
vmware@pks01-term01: ~/plane\$							

Pod に付与された
IP アドレス

Pod に付与された
ラベル

```

2. vmware@pks01-term01: ~/plane (ssh)

vmware@pks01-term01:~/plane$ kubectl get pod -o wide
NAME                                READY   STATUS    RESTARTS   AGE      IP           NODE
NOMINATED NODE
adsb-sync-547b7f6db8-lw2mm          1/1     Running   0          17m     192.168.20.7   72c1a859-87ee-4882-989e-dba487
d1e49e <none>
mysql-0                             1/1     Running   0          9m      192.168.20.9   cf354719-81ca-4ceb-9496-08d43a
17c6d0 <none>
planespotter-app-7794c7db85-rscjx   1/1     Running   0          17m     192.168.20.6   72c1a859-87ee-4882-989e-dba487
d1e49e <none>
planespotter-frontend-5cdf8d7bbd-4sgfr 1/1     Running   0          41m     192.168.20.3   72c1a859-87ee-4882-989e-dba487
d1e49e <none>
planespotter-frontend-5cdf8d7bbd-52g9m 1/1     Running   0          41m     192.168.20.4   cf354719-81ca-4ceb-9496-08d43a
17c6d0 <none>
planespotter-frontend-5cdf8d7bbd-pwgjt 1/1     Running   0          24m     192.168.20.2   7c622885-8ce7-4723-bb55-aa3868
54af87 <none>
redis-server-7bcfbcd44-n282b        1/1     Running   0          17m     192.168.20.8   cf354719-81ca-4ceb-9496-08d43a
17c6d0 <none>
vmware@pks01-term01:~/plane$ kubectl get pod --show-labels
NAME                                READY   STATUS    RESTARTS   AGE      LABELS
adsb-sync-547b7f6db8-lw2mm          1/1     Running   0          17m     app=adsb-sync,pod-template-hash=1036392864
mysql-0                             1/1     Running   0          9m      app=mysql,controller-revision-hash=mysql-77b5
75584,statefulset.kubernetes.io/pod-name=mysql-0
planespotter-app-7794c7db85-rscjx   1/1     Running   0          17m     app=planespotter,pod-template-hash=3350738641
planespotter-frontend-5cdf8d7bbd-4sgfr 1/1     Running   0          41m     app=planespotter-frontend,pod-template-hash=1
789483668,tier=frontend
planespotter-frontend-5cdf8d7bbd-52g9m 1/1     Running   0          41m     app=planespotter-frontend,pod-template-hash=1
789483668,tier=frontend
planespotter-frontend-5cdf8d7bbd-pwgjt 1/1     Running   0          25m     app=planespotter-frontend,pod-template-hash=1
789483668,tier=frontend
redis-server-7bcfbcd44-n282b        1/1     Running   0          17m     app=redis-server,pod-template-hash=3679678900
,tier=backend
vmware@pks01-term01:~/plane$ 

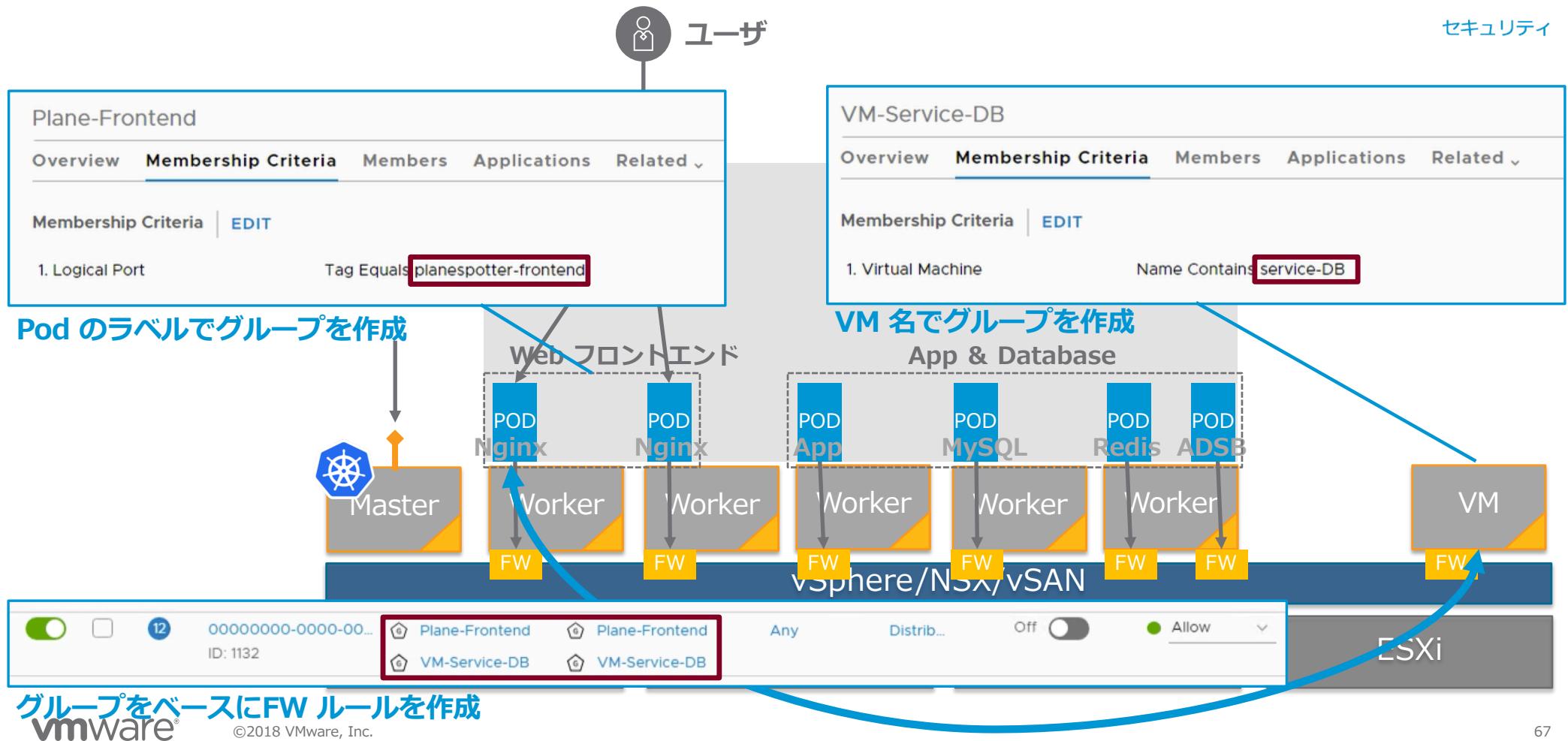
```

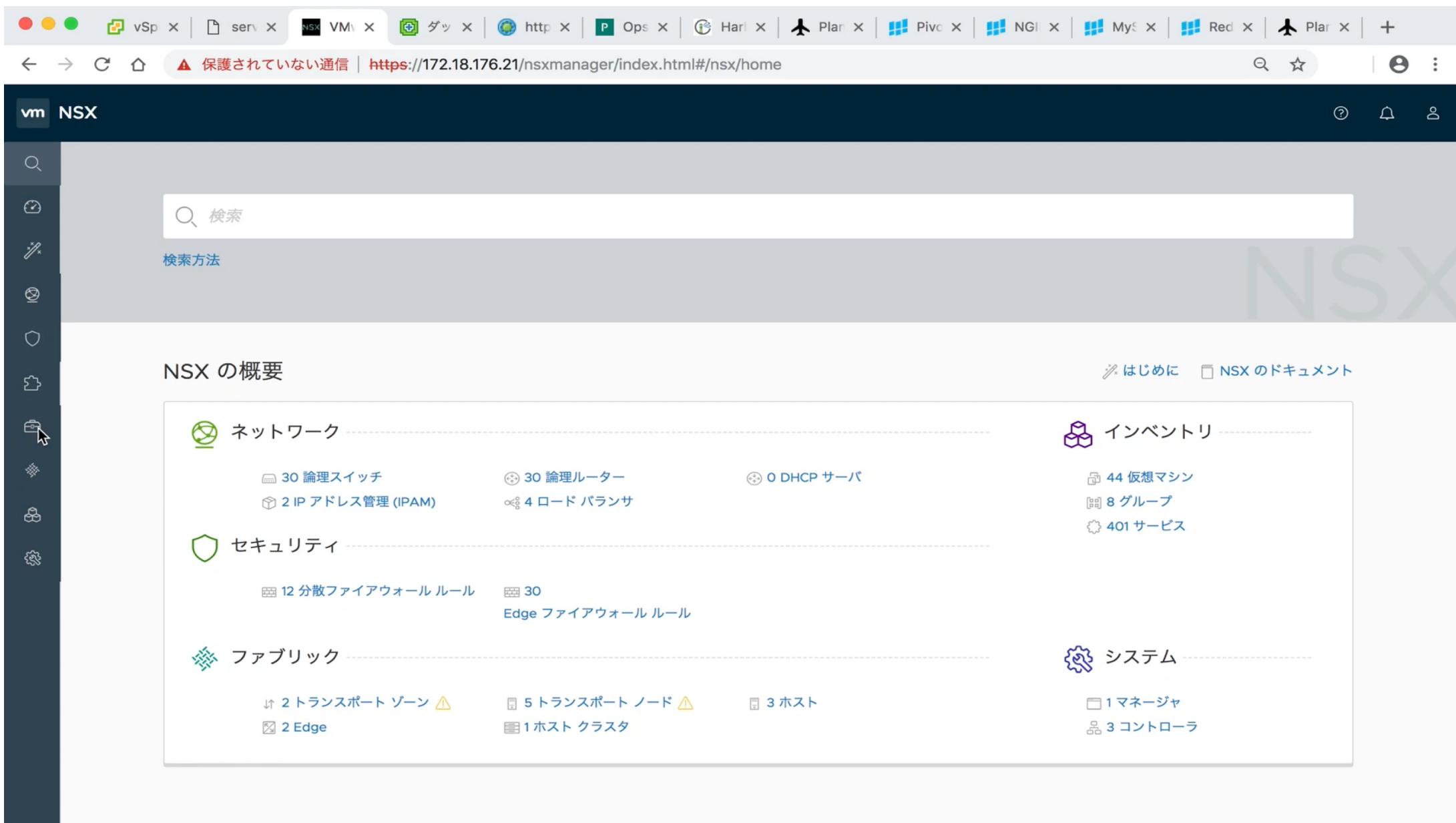
セキュリティ

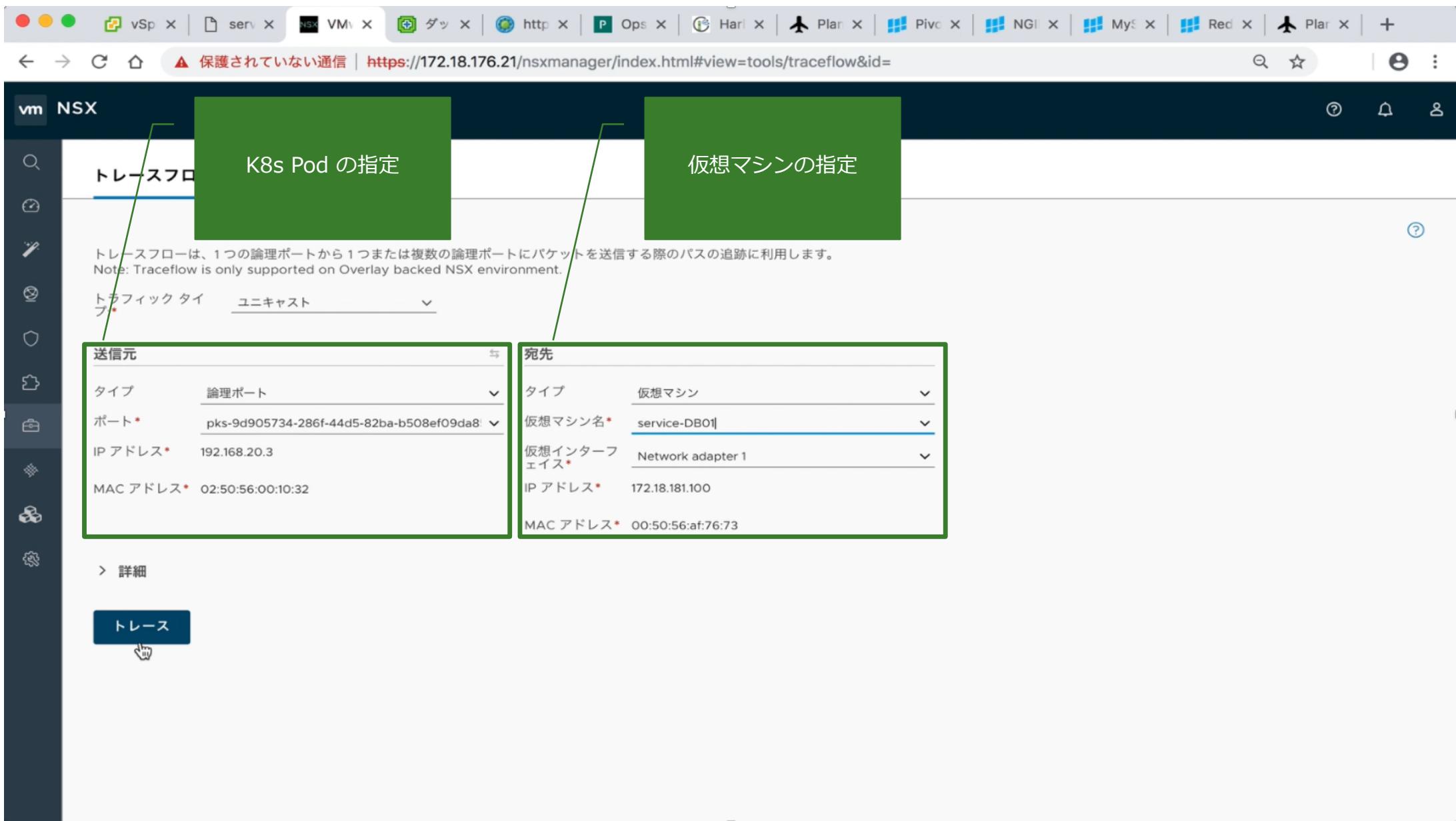
既存サービスとのセキュアな連携



セキュリティ







NSX VMs | ダッシュ | http | Ops | Harlan | Planar | Pivotal | NGI | MySQL | Red Hat | Planar | +

保護されていない通信 | https://172.18.176.21/nsxmanager/index.html#view=tools/traceflow&id=

vm NSX

ト雷斯フロー

送信元 完先

ポート pks-9d905734-286f-44d5-82ba-b508ef09da85-planespotter-frontend-5cdf8d7bbd-4sgfr 仮想マシン service-DB01

IP /MAC アドレス 192.168.20.3/02:50:56:00:10:32 IP /MAC アドレス 172.18.181.100/00:50:56:af:76:73

トレス結果 表示: すべて 1 個を配信済み 0 個がドロップ

物理ホップ 観測タイプ トランスポートノード コンポーネント

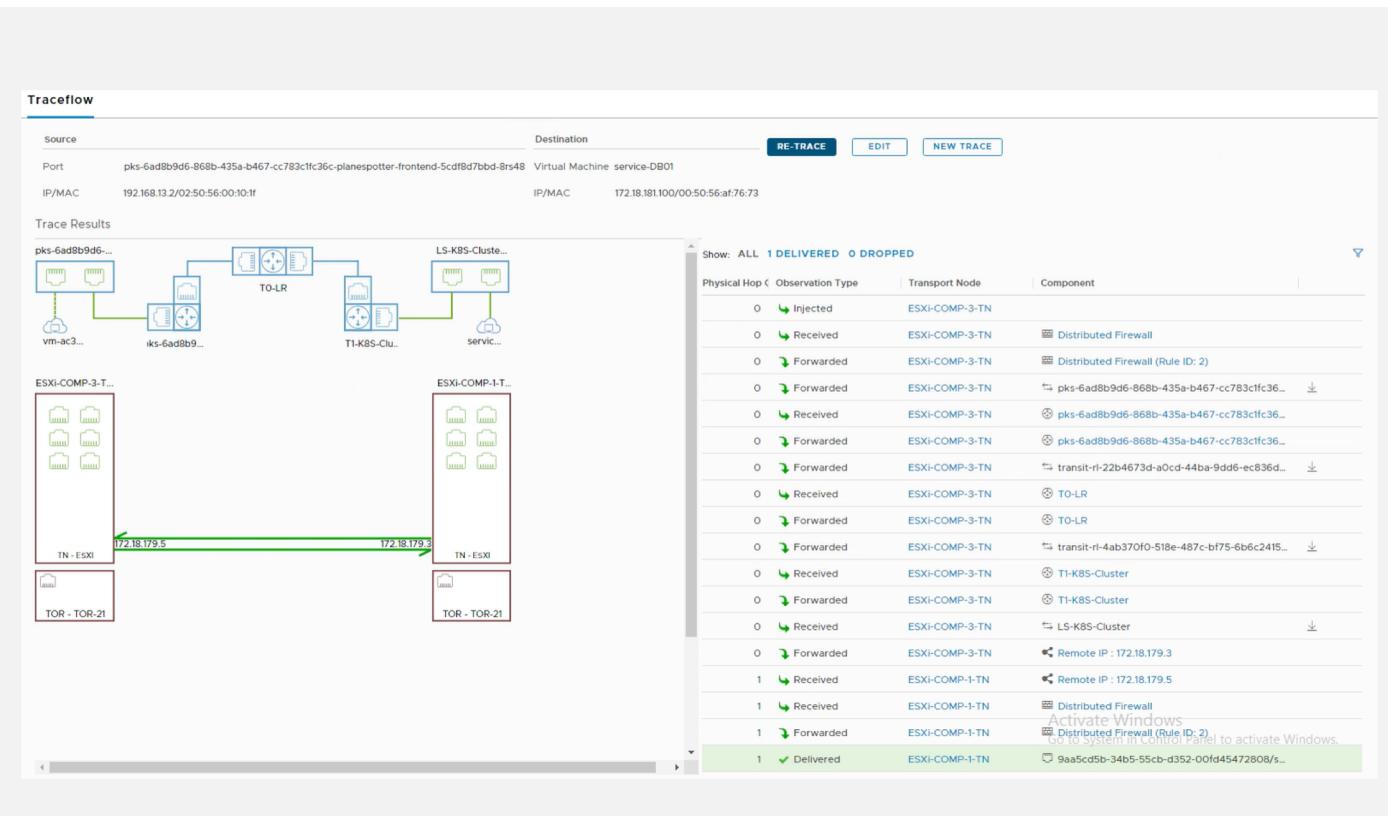
観測が見つかりません

(参考) 仮想マシン - コンテナ間の通信

NSX-T (Traceflow) による通信経路の確認



セキュリティ



コンテナ環境の通信経路を可視化

- 仮想マシン - コンテナ間の通信経路を可視化
- 経路上の障害ポイントを特定

テクノロジー視点での課題

サービス提供部門からの要望



コンテナ環境の構築と運用

クラスターの払い出し
パッチ適用
障害の検知と復旧



インフラ構成の自動化

コンテナへのネットワーク/ストレージサービスの提供



セキュリティ

コンテナ間や既存サーバとのネットワーク通信におけるセキュリティ



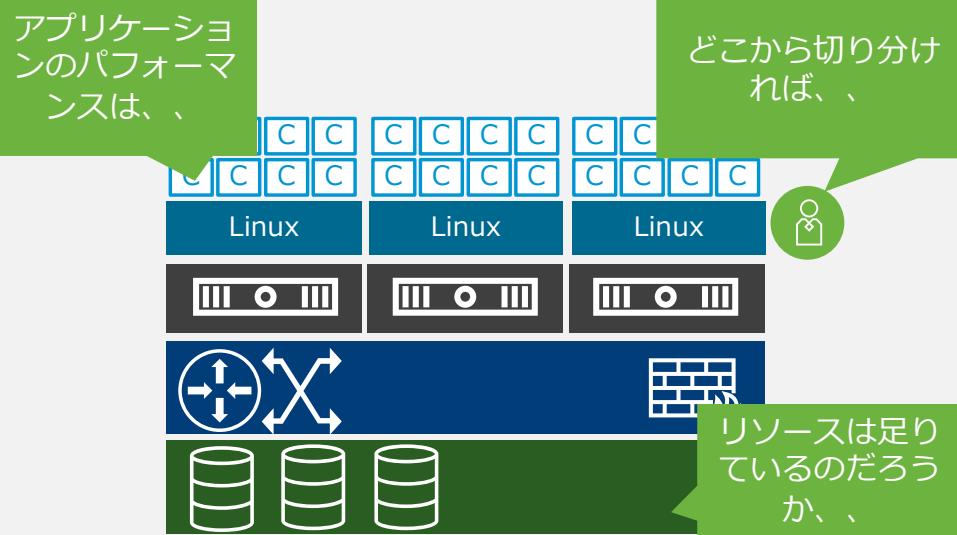
運用監視

コンテナ環境とコンテナ内アプリの運用監視

運用監視

インフラ基盤からコンテナまで包括的な運用

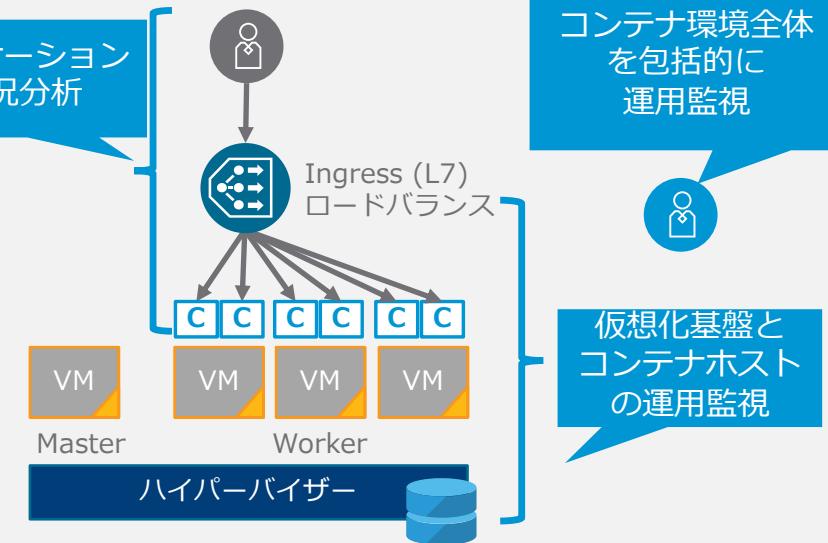
インフラとコンテナ個々の運用監視



アプリケーション
の動作が“遅いので
すが

包括的な運用監視

アプリケーション
の状況分析



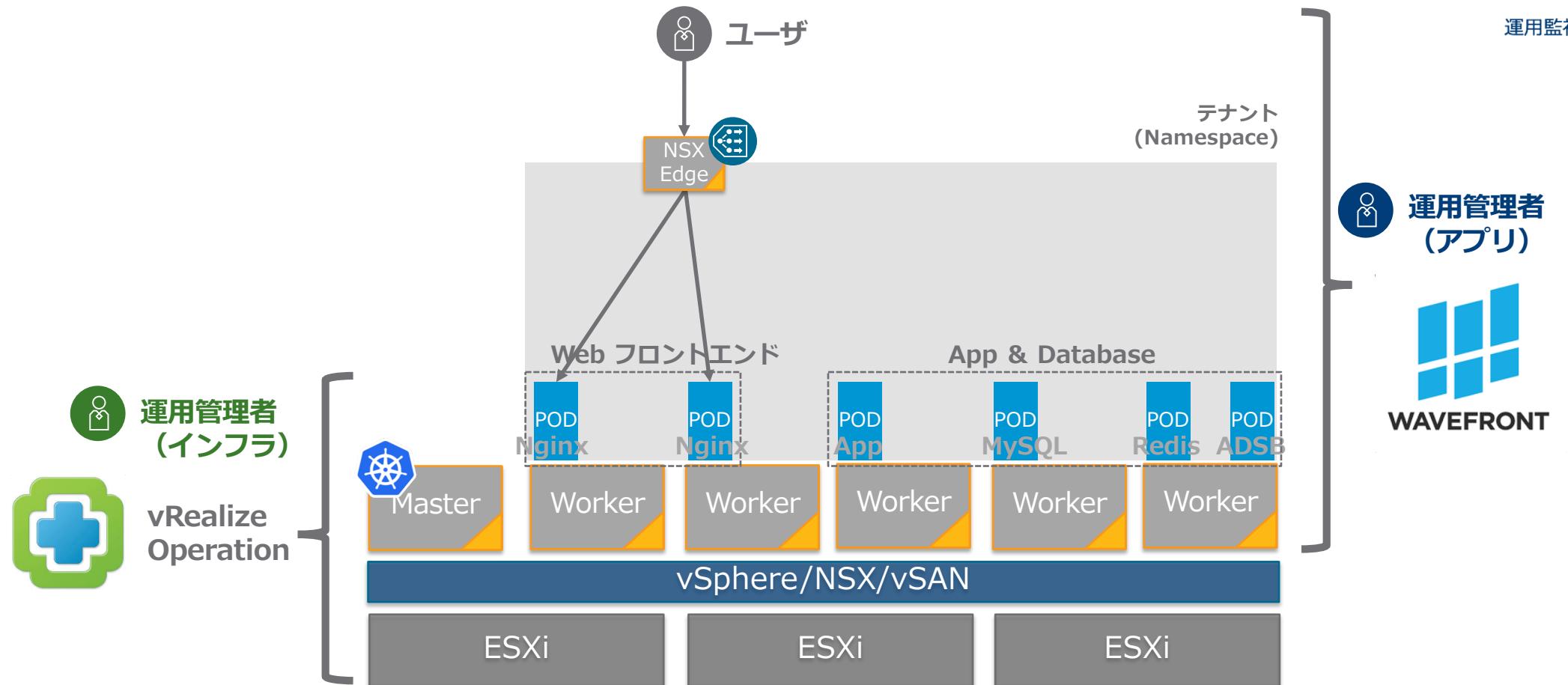
運用監視

運用監視

Kubernetes 環境の運用監視

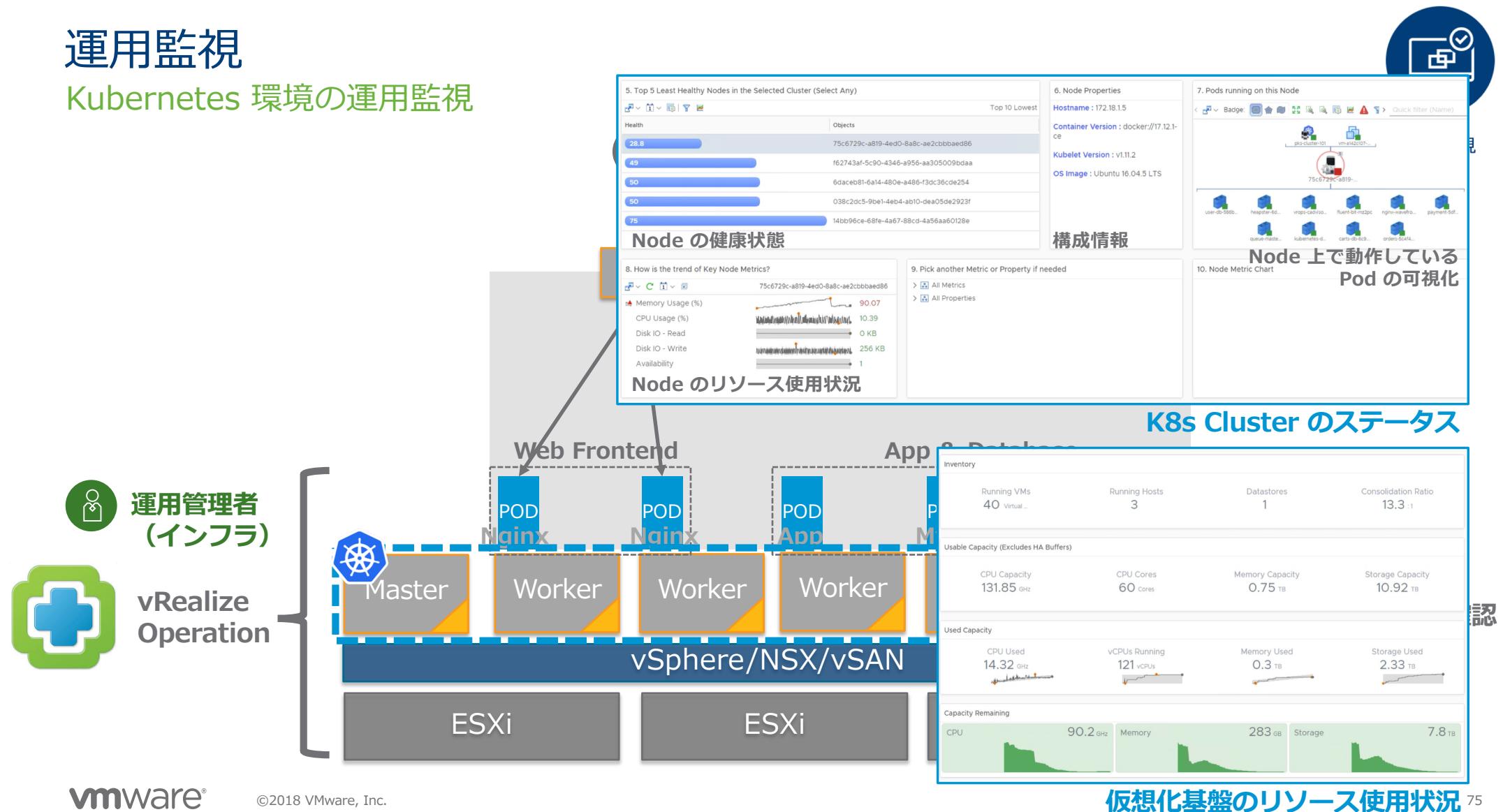


運用監視



運用監視

Kubernetes 環境の運用監視





← → C ⌂ 保護されていない通信 | <https://172.18.176.42/ui/login.action?skipSSO=true>

VMware™ vRealize Operations Manager®

ローカル ユーザー

用户名

密码

ログイン

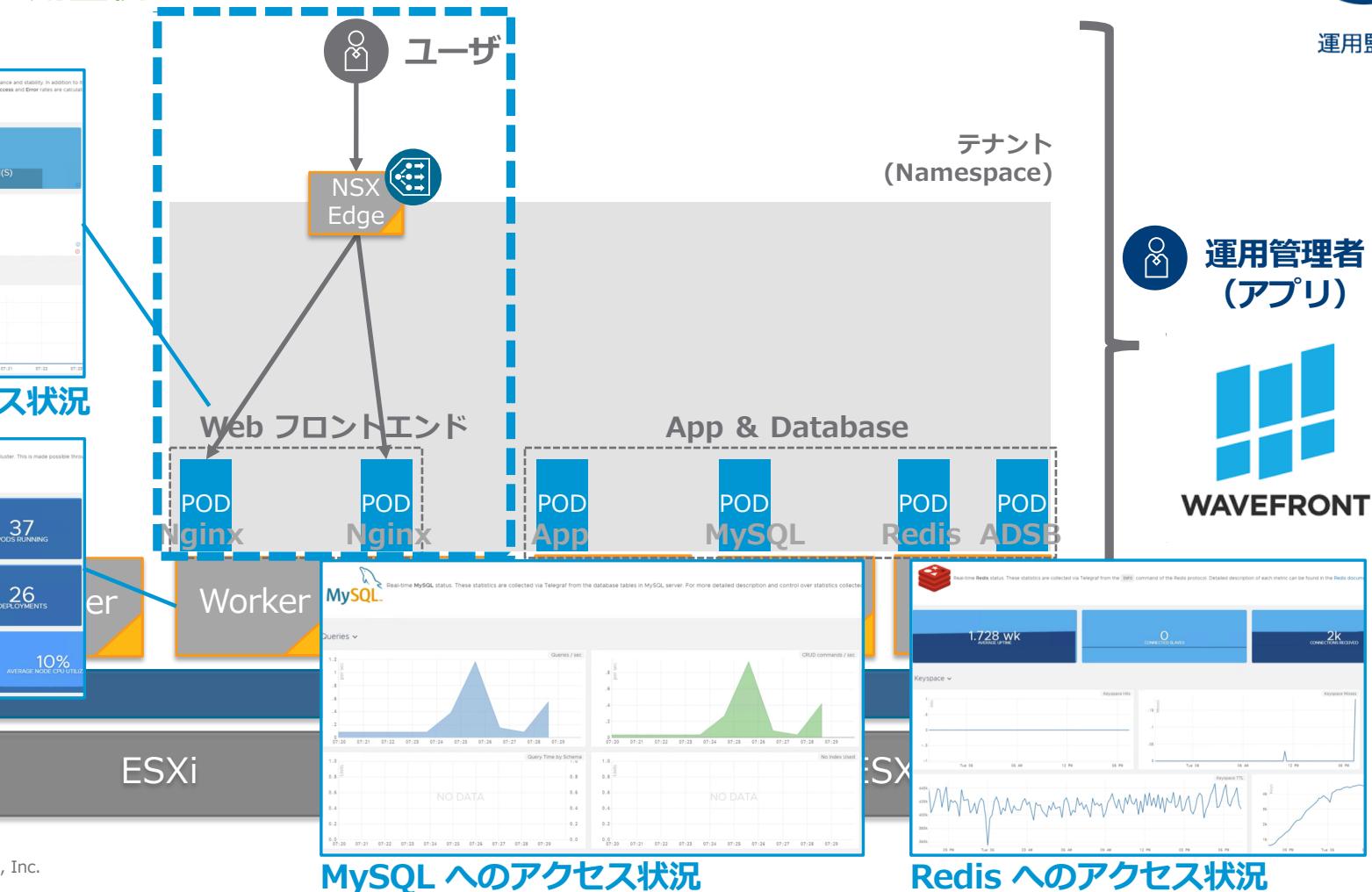


運用監視

Kubernetes 環境の運用監視



運用監視



vSp x | serv x | NSX VMW x | ダッシュ x | http x | Ops x | Harl x | Plan x | Pivo x | NGI x | MyS x | Red x | Plan x | +

← → C ⌂ https://longboard.wavefront.com/dashboard/integration-pks#_v01(g:(c:off,d:691200,ls:!f,s:1541133025,w:'8d'),p:('1_cluster':(d:defaultQuery,f:T...))

We use cookies for advertising, social media and analytics purposes. Read about how we use cookies and how you can control them here: [COOKIE POLICY](#)

WAVEFRONT by VMware Dashboards 2 Alerts Integrations Browse Getting Started 100% Search Dashboards

LIVE DATA CUSTOM DATE Start 11/02/18 1:30 PM End 11/10/18 1:30 PM 10M 2H 6H 12H 1D 8D Compare OFF Timezone BROWSER DEFAULT Show Events FROM CHART

OVERVIEW NODES NAMESPACES PODS POD CONTAINERS DEPLOYMENTS SERVICES REPLICASET

Cluster service-instance_6ad8b9d6-868b-435a-b467-cc783c1fc36c Select Namespace All Namespace default

Overview

 **Pivotal Container Service™** This dashboard provides complete visibility into each level of a Pivotal Container Service (PKS) cluster. This is made possible through Wavefront's easy out-of-the-box integration with **Heapster** and **kube-state-metrics**. Use the dropdown menu at the top to select a different Cluster and Namespace.

5 NODES	6 NAMESPACE ACTIVE	0 NAMESPACE TERMINATING	37 PODS RUNNING	0 PODS FAILED	0 PODS PENDING
---------	--------------------	-------------------------	-----------------	---------------	----------------

VMware PKS + SDDC

コンテナ基盤に必要なコンポーネントを包括的に提供

コンテナをホストする高度なインフラ環境を
容易に導入/運用することができる**運用管理者**のためのソリューション

Kubernetes 環境の構成と運用

- Kubernetes クラスタ構成と運用の自動化
- クラスタの自動復旧による自律的なインフラの運用

柔軟なソフトウェアベースのインフラストラクチャ

- ロードバランサまで含むネットワークの提供
- 仮想化/コンテナ環境の一貫したセキュリティ
- 永続性ストレージ

包括的な運用監視

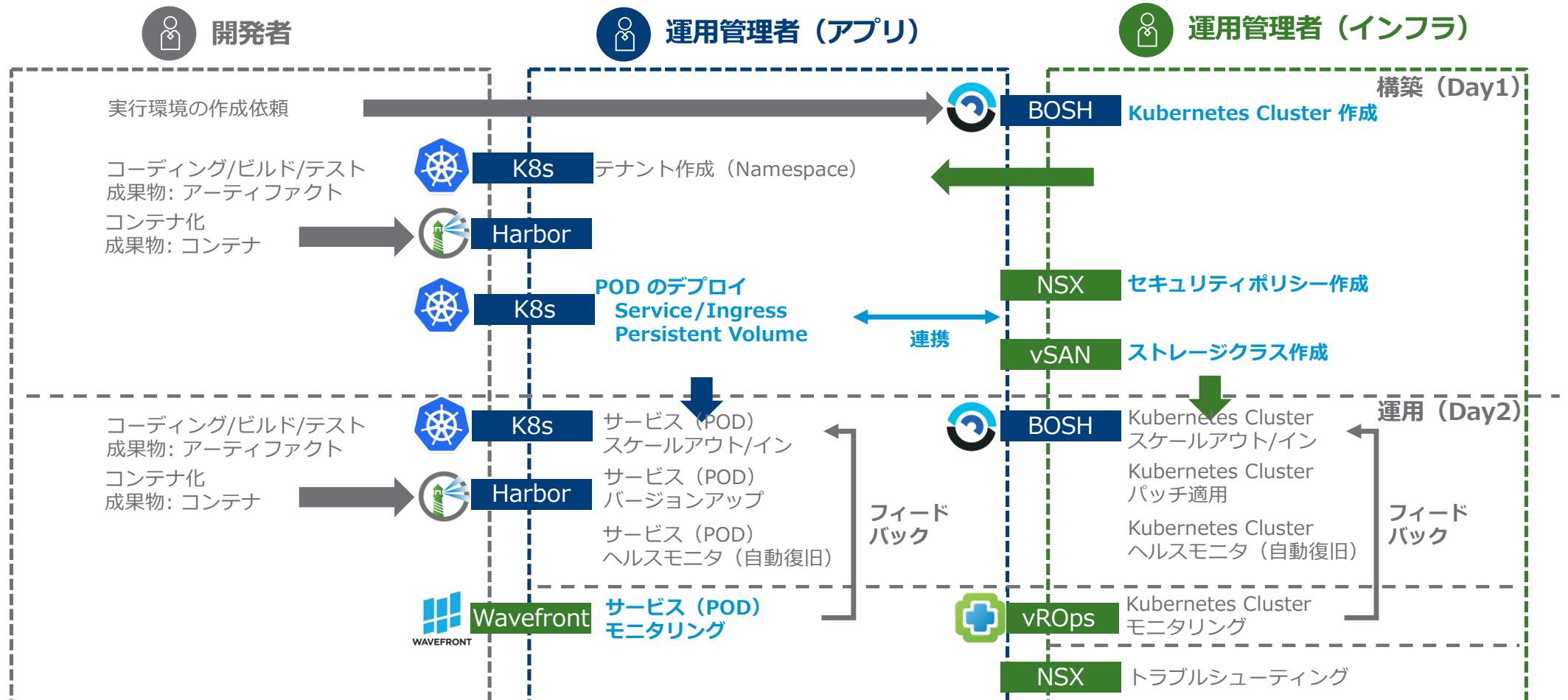
- SDDC 環境全体の運用監視
- 仮想化基盤からコンテナ基盤まで包括的に可視化

コンテナ基盤導入に伴う 課題とアプローチ

スキルセットと運用手法

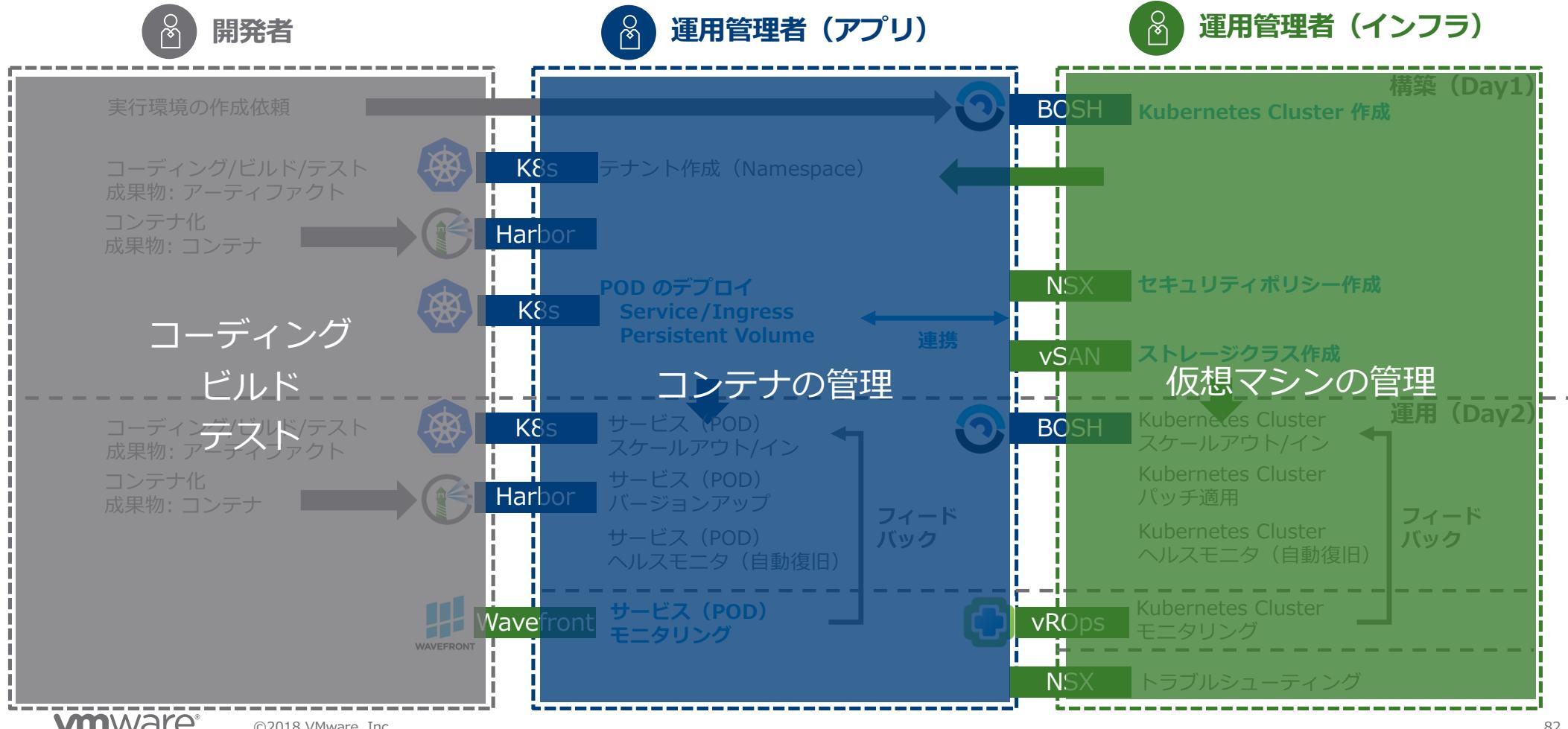
VMware PKS + SDDC による課題へのアプローチ

オペレーションフロー



VMware PKS + SDDC による課題へのアプローチ

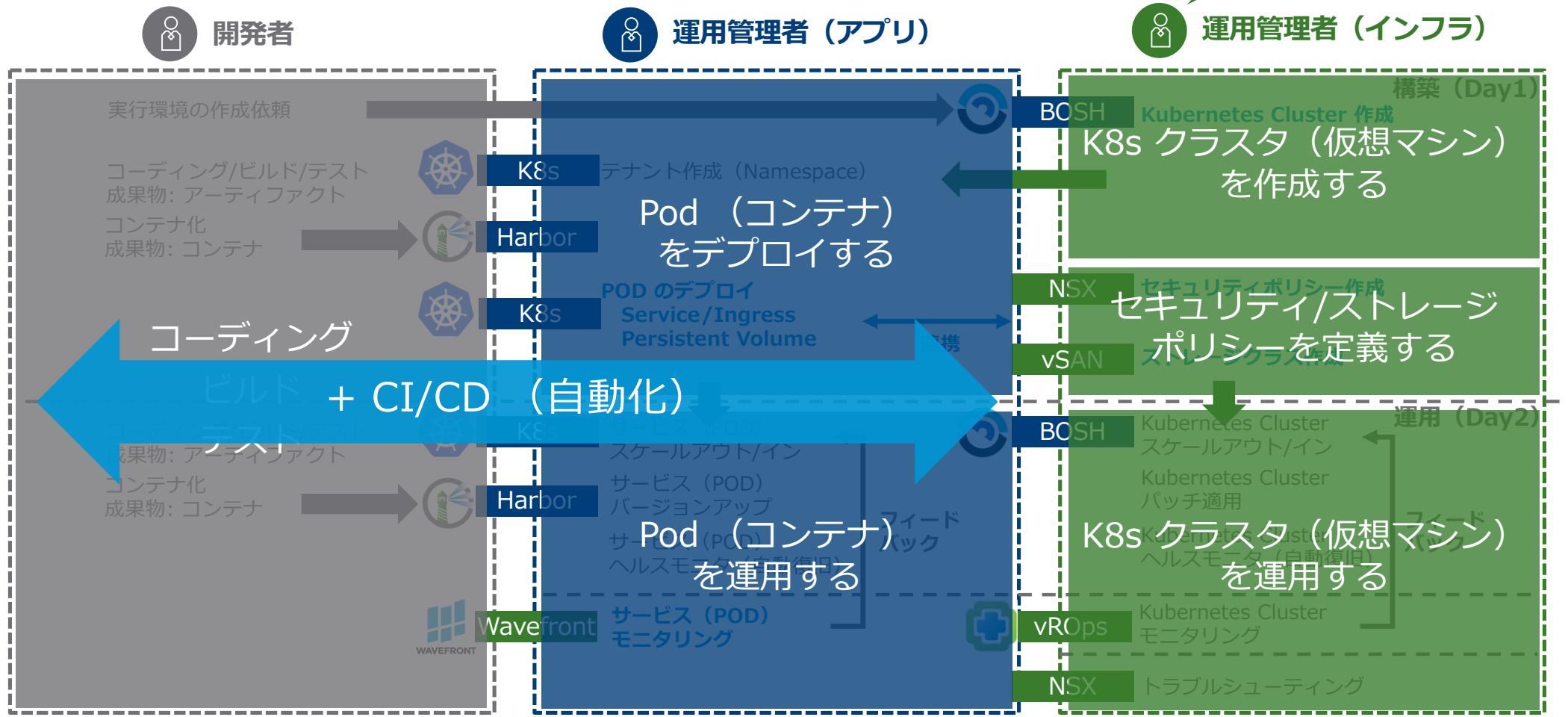
オペレーションフロー



VMware PKS + SDDC による課題へのアプローチ

オペレーションフロー

仮想化環境のスキルセットでコンテナ基盤を運用できる

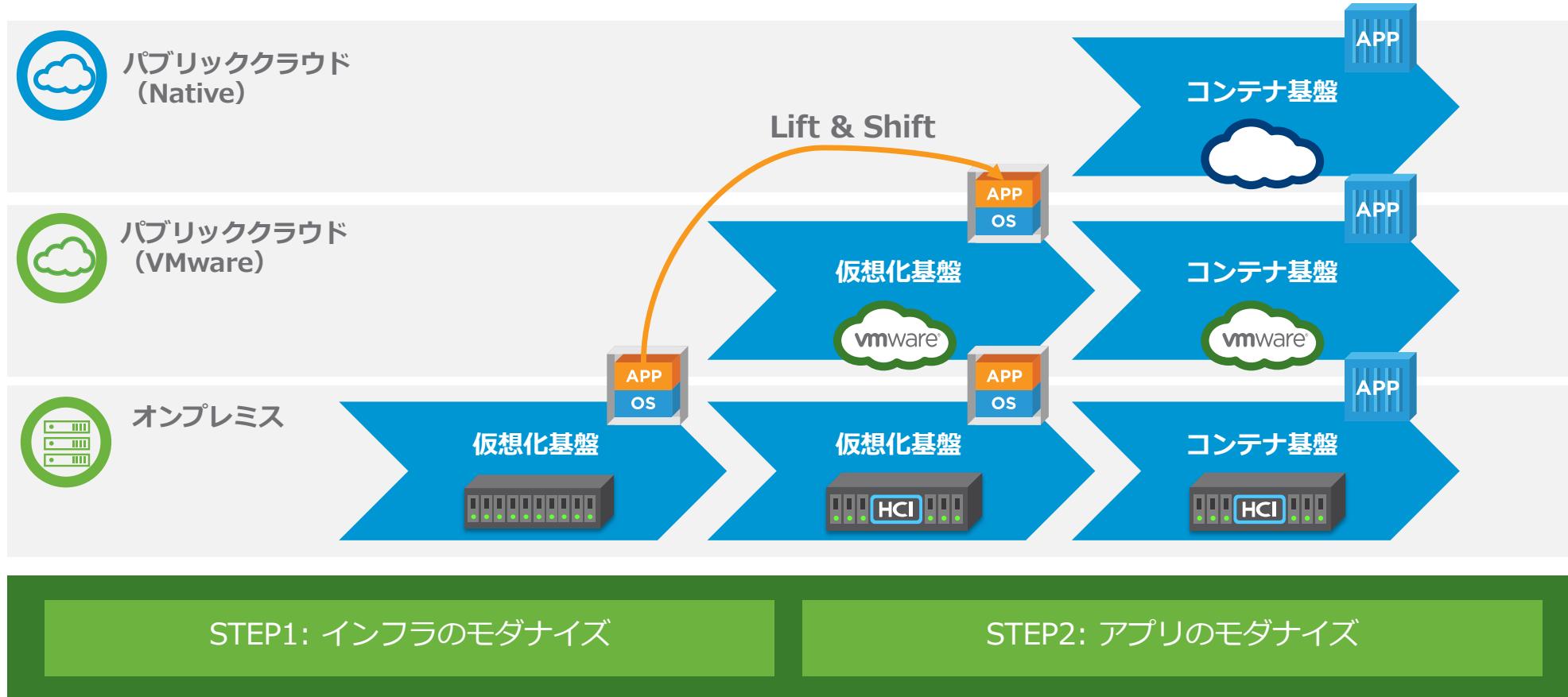


コンテナ基盤導入に伴う 課題とアプローチ

分散環境のインテグレーション

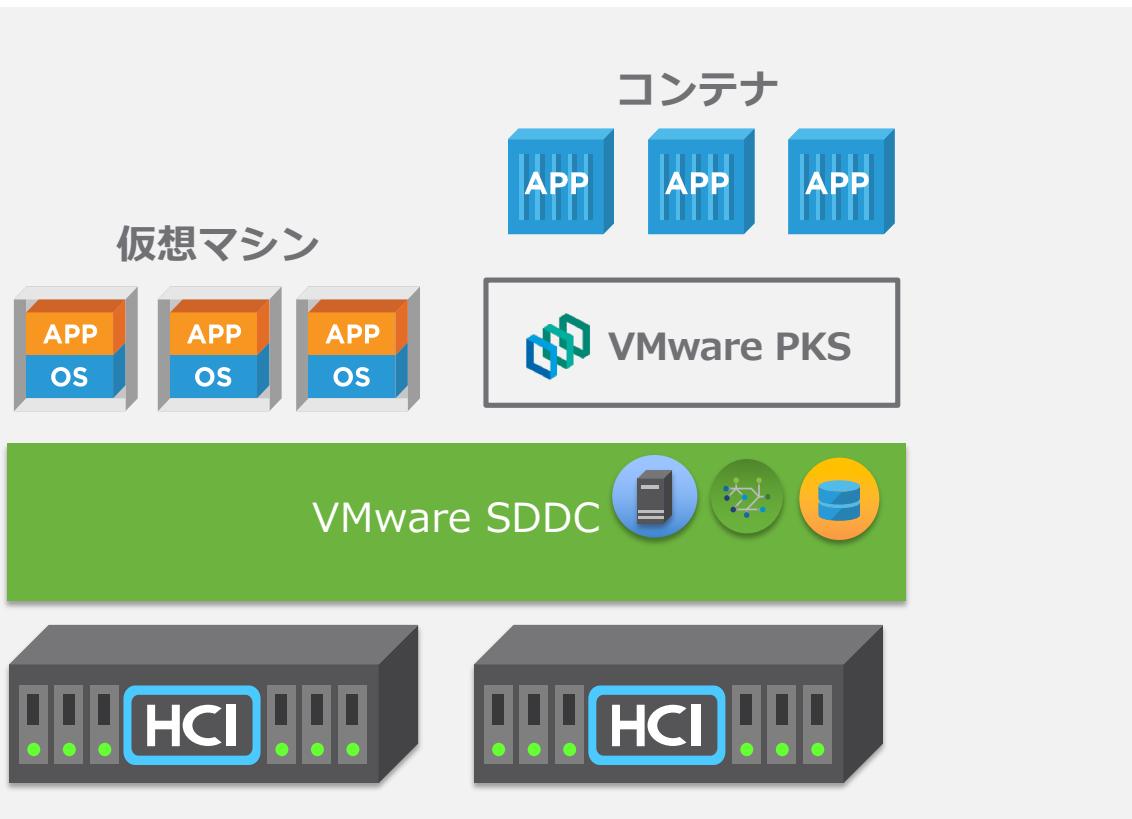
分散環境のインテグレーション

コンテナ化のアプローチ



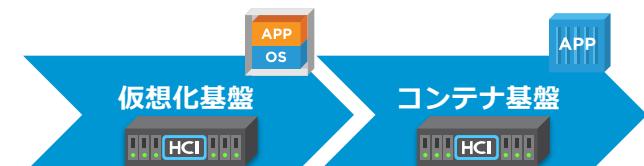
仮想マシンとコンテナ

仮想マシンとコンテナサービスを同一のプラットフォーム上で提供



コンテナと既存ワークLOADとの連携

- 新旧のワークLOADを同一
プラットフォーム上で管理
- 既存インフラと同じ運用手法
を踏襲
- 仮想マシン-コンテナ間で
一貫したポリシーの適用



オンプレミスとパブリッククラウド

Kubernetes をベースにしたハイブリッドクラウド環境の提供

Kubernetes



VMware PKS

Self-Managed

オンプレミス上に仮想化基盤と PKS を構築



パブリッククラウド
(VMware)



PKS on VMC

Managed (IaaS)

VMC on AWS (ハードウェア + 仮想化ソフトウェア) 上に PKS を構築



パブリッククラウド
(Native)

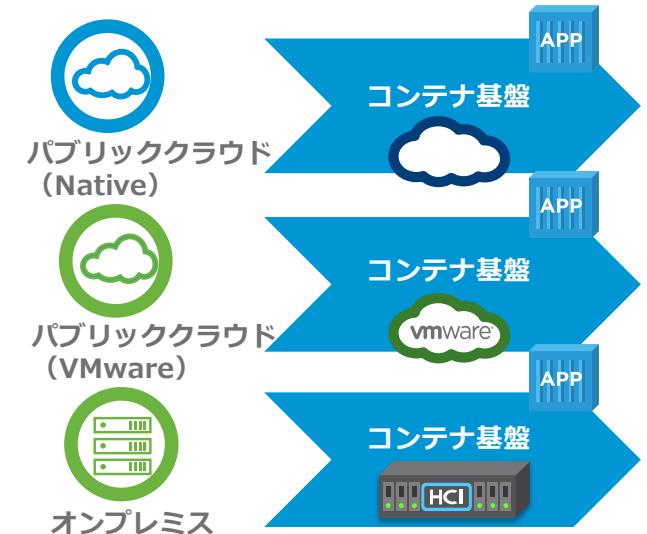


VMware Cloud PKS

Fully-Managed
(SaaS)

Kubernetes サービスを SaaS として提供

プライベートクラウドからパブリッククラウドまで包括的なサービスの提供



導入事例

オンラインゲーム会社

自社製の Kubernetes 環境を PKS でリプレース



背景と課題

自社で構築した Kubernetes 環境を2年間運用

複雑な環境の維持・管理が負担になり
結果として運用コストと人件費が増大

Kubernetes 環境と既存のインフラ環境
が統合されておらず新しい IT サイロに

解決策

サポートのあるエンタープライズ
グレード Kubernetes ソリューション

既存 VMware SDDC 環境とシームレスな
運用

PKS を選んだ理由

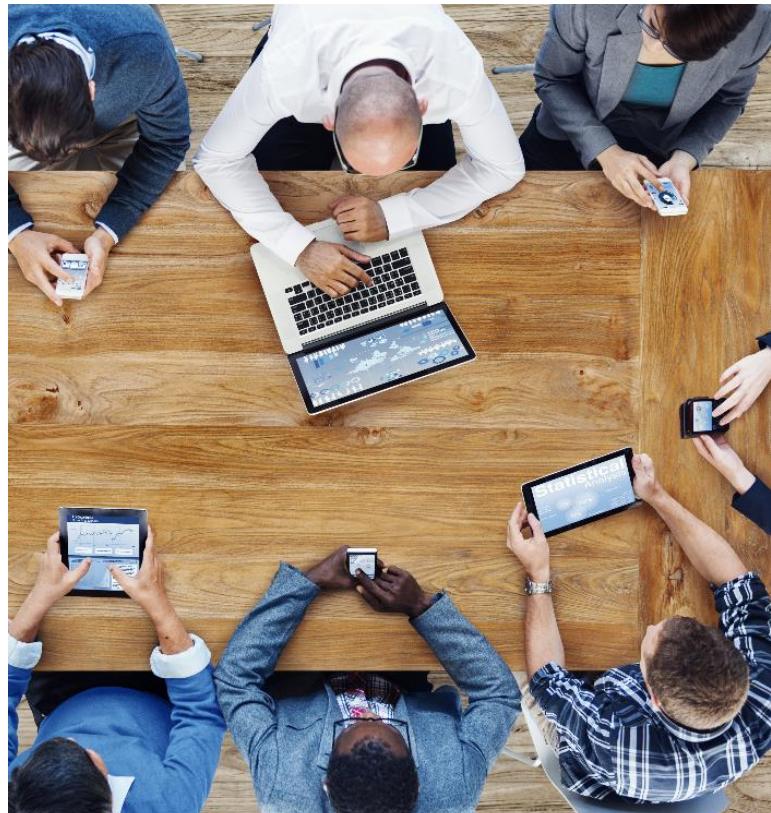
VMware インフラと統合されているため、既存のスキルセットの延長で運用、IT サイロ化を防ぐ

ワークフローが自動化されており、
エンタープライズ向けの機能を備えることにより、day1 (構築) や day2 (運用) の複雑なタスクを単純化

プラットフォームの管理を開発
チームからインフラチームに移管できるので、開発チームはコーディングに集中できるようになる

通信事業者

レガシーなアプリケーションを PKS に移行



背景と課題

WebSphere や WebLogic 等の環境で動作するアプリケーションの維持・管理が負担に

大規模環境でのアドホックで細かな変更が、運用コストとインフラコストを拡大

アプリケーションのコンテナ移行により、アドホックな変更を削減したい

解決策

コンテナ化したワークロードを実行するための Kubernetes 環境の導入

PKS を選んだ理由

包括的な Kubernetes ソリューションでコンテナ・アプリを実行することで、アプリの展開と管理を単純化できる

VMware SDDC との統合により、VMware 製品の利用範囲を拡大できることも重要だった

大手銀行

コンテナ技術による事業変革



背景と課題

新しい製品やサービスの市場投入を加速するため、デジタルトランスフォーメーションを検討

新しいアプリを開発において、開発チームは“コンテナファースト”戦略を採用

既存の IT インフラではこうした新しいコンテナアプリをサポートできない

解決策

コンテナアプリを構築/実行するための現代的なプラットフォーム

金融業界の厳しいセキュリティ要件に適合するコンテナソリューション

PKS を選んだ理由

ネイティブな Kubernetes によるオープンなアーキテクチャにより、ベンダーロックインを排除し、スキルを持つ人材の確保も容易になる

NSX-T Data Center のマイクロセグメンテーション機能と PCI 互換性により、厳しいセキュリティ要件に対応

自動化されたワークフローにより day1 (構築) や day2 (運用) の複雑なタスクを単純化でき、開発者はコード開発に集中できるようになる

まとめ

VMware の提供する次世代アプリケーションプラットフォーム

コンテナに関連する複数のテクノロジー

- コンテナ基盤に必要なコンポーネントを包括的に提供
- ソフトウェアベースの柔軟なインフラ環境との統合

スキルセットと運用手法

- 既存仮想化環境のスキルセットで運用可能
- 運用時のオペレーション自動化

分散環境のインテグレーション

- 仮想マシンからコンテナまでをシームレスに接続
- オンプレからクラウド環境まで包括的なサービスの提供

本セッションに関連する展示・ハンズオンラボのご紹介





ご清聴、ありがとうございました。