

2020年度新技術  
ソリューション検討会  
Aチーム技術概要報告書

VMware Tanzu

**KEL**

# VMware Tanzu Webinar目次

## ■ Kubernetes導入

- ・ コンテナ背景/コンテナ概要/Kubernetes導入
- ・ Kubernetesの課題/技術的制限
- ・ Kubernetesまとめ

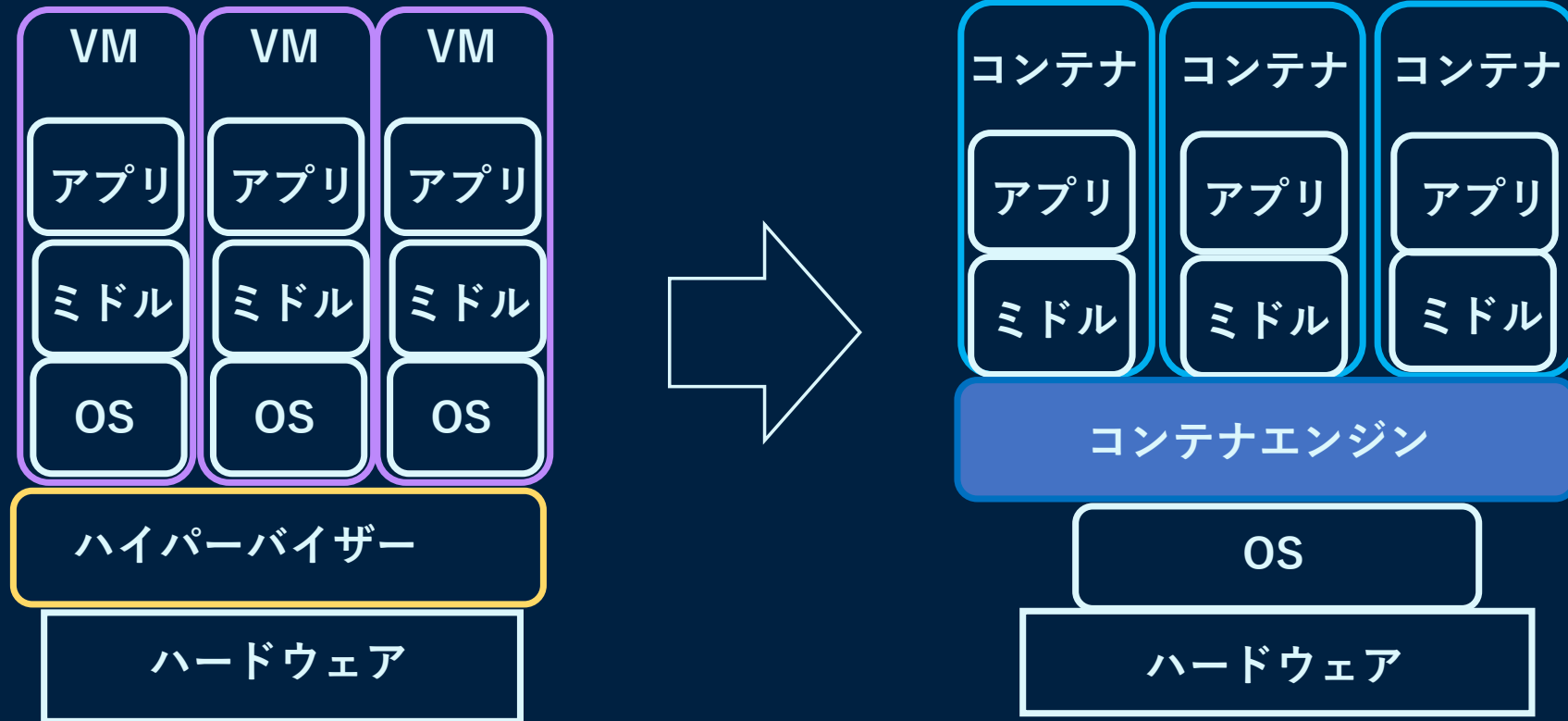
## ■ VMware Tanzu (vSphere with Tanzu)

- ・ 概要/登場背景
- ・ TKG(kubeadm/ClusterAPI)
- ・ vSphere with Tanzuの役割/操作画面/まとめ

## ■ VMware Tanzu vs 標準Kubernetes (検証結果)

- ・ 検証結果
- ・ 今後のアクション

# Kubernetes導入：コンテナ背景



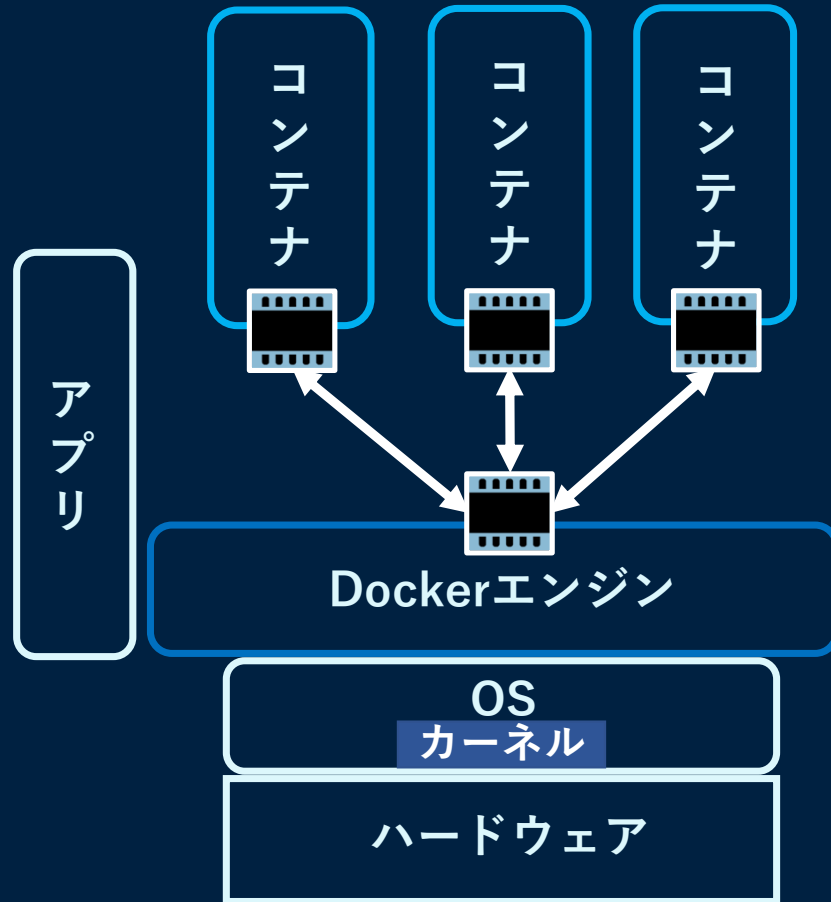
# Kubernetes導入：コンテナ概要

コンテナとは

⇒アプリケーションを動かす為に必要なライブラリやミドルウェアを1つにまとめたもの。

メリット

- ・アプリケーション実行環境構築が超早い
- ・可搬性がある  
⇒OSをまたいで動作する



# Kubernetes導入：Kubernetes概要

## Kubernetes(k8s)とは？

↓  
複数のコンテナを統合的に管理するための  
技術**コンテナオーケストレーションツール**



指示！

指示！



大量のコンテナ



大量のコンテナ



## Kubernetes導入：なぜk8sなのか

### Googleが中心

- ・長年の実績を用いてk8sを開発
- ・CNCFというコミュニティを立ち上げ、ファーストプロジェクトにk8sを献上

### 多機能だが柔軟性アリ

- ・宣言的設定やループなど障害対策も完璧
- ・多数の要求に素早く対応し3か月に1回のアップデートを実施

### 移植性の高さ

- ・クラウドでもオンプレでも同じように使える
- ⇒マニフェストファイルを変えずに移行できる。

### コストの低さ

- ・時間やお金など開発環境構築に割くリソースを削減できる
- ・クラウド事業者はk8sのマネジメントサービスも行っている

デファクトスタンダードに！

## Kubernetesの課題/技術的制限①

### ① LINUXに対する知識

- **kubectl**での操作
- マニフェストファイルの作成方法

```
root root 221 12月 4 12:39 .
root root 224 11月 19 16:19 ..
1 root root 176 12月 4 12:39 .bash_history
1 root root 18 12月 29 2013 .bash_logout
-r--r--r-- 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r-- 1 root root 176 12月 29 2013 .bashrc
-rwxr-xr-x 4 root root 31 11月 19 16:32 .cache
-rwxr-xr-x 3 root root 18 11月 19 16:32 .config
-rw-r--r-- 1 root root 100 12月 29 2013 .cshrc
-rwxr-xr-x 3 root root 25 11月 19 16:21 .dbus
-rw-r--r-- 1 root root 129 12月 29 2013 .tcshrc
-rw-r--r-- 1 root root 1748 11月 19 16:20 anaconda-ks.cfg
-rw-r--r-- 1 root root 1779 11月 19 16:22 initial-setup-ks.cfg
drwxr-xr-x 2 root root 6 11月 19 16:36 sample-a
root@localhost ~]# vi hello.c
root@localhost ~]# gcc hello.c
root@localhost ~]# ./a.out > log.txt
root@localhost ~]# ls
-rw-r--r-- 1 root root 1748 11月 19 16:20 anaconda-ks.cfg
-rw-r--r-- 1 root root 1779 11月 19 16:22 initial-setup-ks.cfg
-rw-r--r-- 1 root root 176 12月 4 12:39 .bash_history
-rw-r--r-- 1 root root 18 12月 29 2013 .bash_logout
-r--r--r-- 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r-- 1 root root 176 12月 29 2013 .bashrc
-rwxr-xr-x 4 root root 31 11月 19 16:32 .cache
-rwxr-xr-x 3 root root 18 11月 19 16:32 .config
-rw-r--r-- 1 root root 100 12月 29 2013 .cshrc
-rwxr-xr-x 3 root root 25 11月 19 16:21 .dbus
-rw-r--r-- 1 root root 129 12月 29 2013 .tcshrc
-rw-r--r-- 1 root root 1748 11月 19 16:20 anaconda-ks.cfg
-rw-r--r-- 1 root root 1779 11月 19 16:22 initial-setup-ks.cfg
-rw-r--r-- 1 root root 176 12月 4 12:39 .bash_history
-rw-r--r-- 1 root root 18 12月 29 2013 .bash_logout
-r--r--r-- 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r-- 1 root root 176 12月 29 2013 .bashrc
-rwxr-xr-x 4 root root 31 11月 19 16:32 .cache
-rwxr-xr-x 3 root root 18 11月 19 16:32 .config
-rw-r--r-- 1 root root 100 12月 29 2013 .cshrc
-rwxr-xr-x 3 root root 25 11月 19 16:21 .dbus
-rw-r--r-- 1 root root 129 12月 29 2013 .tcshrc
-rw-r--r-- 1 root root 1748 11月 19 16:20 anaconda-ks.cfg
-rw-r--r-- 1 root root 1779 11月 19 16:22 initial-setup-ks.cfg
-rw-r--r-- 1 root root 176 12月 4 12:39 .bash_history
-rw-r--r-- 1 root root 18 12月 29 2013 .bash_logout
-r--r--r-- 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r-- 1 root root 176 12月 29 2013 .bashrc
-rwxr-xr-x 4 root root 31 11月 19 16:32 .cache
-rwxr-xr-x 3 root root 18 11月 19 16:32 .config
-rw-r--r-- 1 root root 100 12月 29 2013 .cshrc
-rwxr-xr-x 3 root root 25 11月 19 16:21 .dbus
-rw-r--r-- 1 root root 129 12月 29 2013 .tcshrc
-rw-r--r-- 1 root root 1748 11月 19 16:20 anaconda-ks.cfg
-rw-r--r-- 1 root root 1779 11月 19 16:22 initial-setup-ks.cfg
-rw-r--r-- 1 root root 176 12月 4 12:39 .bash_history
-rw-r--r-- 1 root root 18 12月 29 2013 .bash_logout
-r--r--r-- 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r-- 1 root root 176 12月 29 2013 .bashrc
-rwxr-xr-x 4 root root 31 11月 19 16:32 .cache
-rwxr-xr-x 3 root root 18 11月 19 16:32 .config
-rw-r--r-- 1 root root 100 12月 29 2013 .cshrc
-rwxr-xr-x 3 root root 25 11月 19 16:21 .dbus
-rw-r--r-- 1 root root 129 12月 29 2013 .tcshrc
-rw-r--r-- 1 root root 1748 11月 19 16:20 anaconda-ks.cfg
-rw-r--r-- 1 root root 1779 11月 19 16:22 initial-setup-ks.cfg
root@localhost ~]# cat log.txt
root@localhost ~]# echo $SHELLLL
root@localhost ~]# echo $SHELL
```

## Kubernetesの課題/技術的制限②

### ②運用

⇒Kubernetesは1つのバージョンの保持が**9カ月**  
早いサイクルでアップデートを行う必要がある  
※AWS等では自動アップデートサービス有

### ③学習コスト

⇒リリースサイクルが早く頻繁に機能が追加・削除される  
学習を継続しなければ、  
**安定した稼働が困難**になる





## Kubernetesまとめ

### Kubernetesとは

- ・ コンテナオーケストレーションツール
- ・ 障害に強く、移植性に長けている
- ・ Googleが中心、デファクトスタンダード

### 課題や技術的制限

- ・ 一定以上の技術知識が必要
- ・ 早いサイクルでアップデートが必要
- ・ 開発者の継続した学習が必要

# VMware Tanzu概要



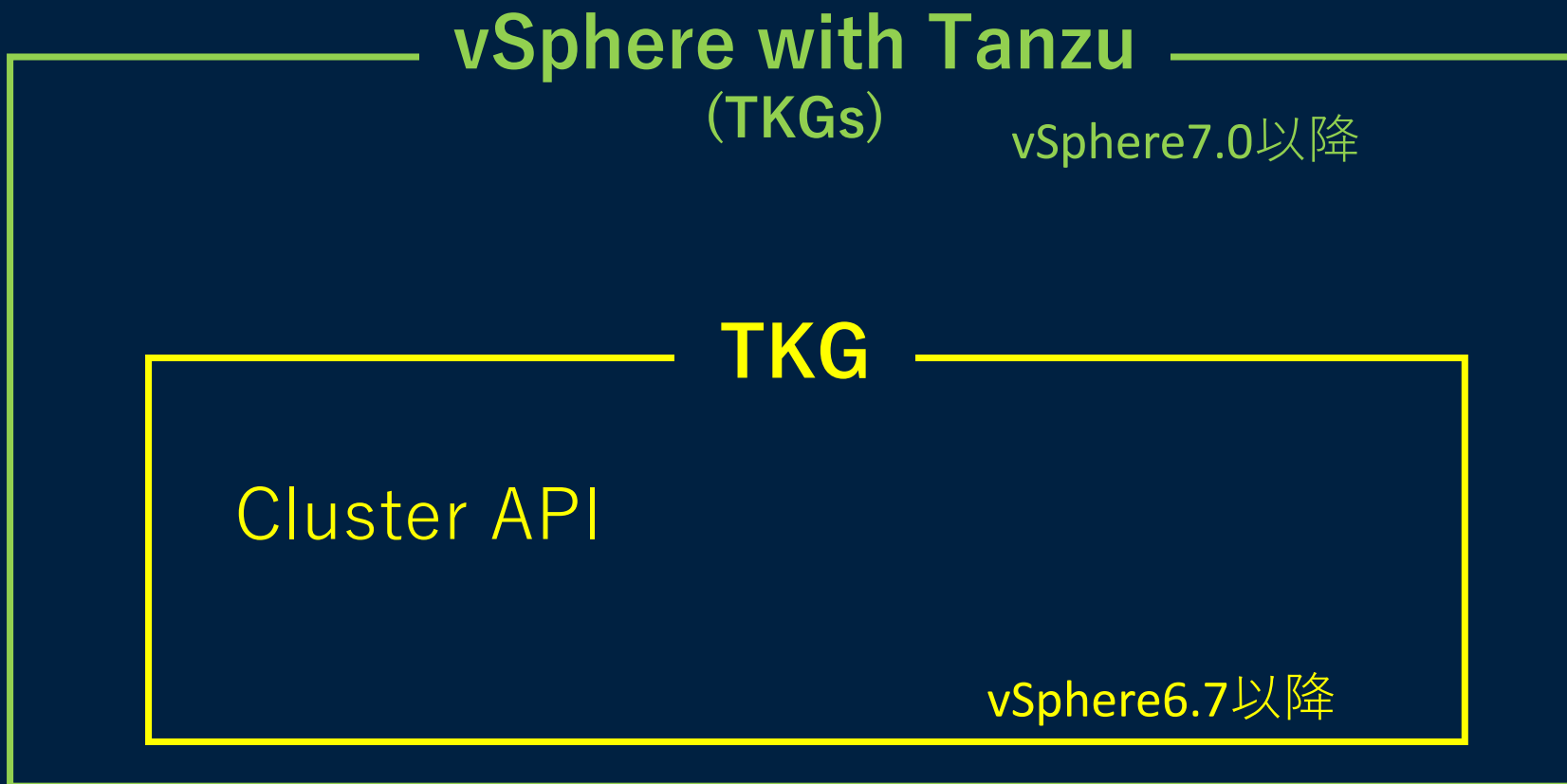
VMware Tanzu



Kubernetesの実行・管理をvSphereの環境下で可能にした構築ツール  
Kubernetes上で稼働するアプリケーションのデプロイや運用など  
ライフサイクルを支援する新サービス

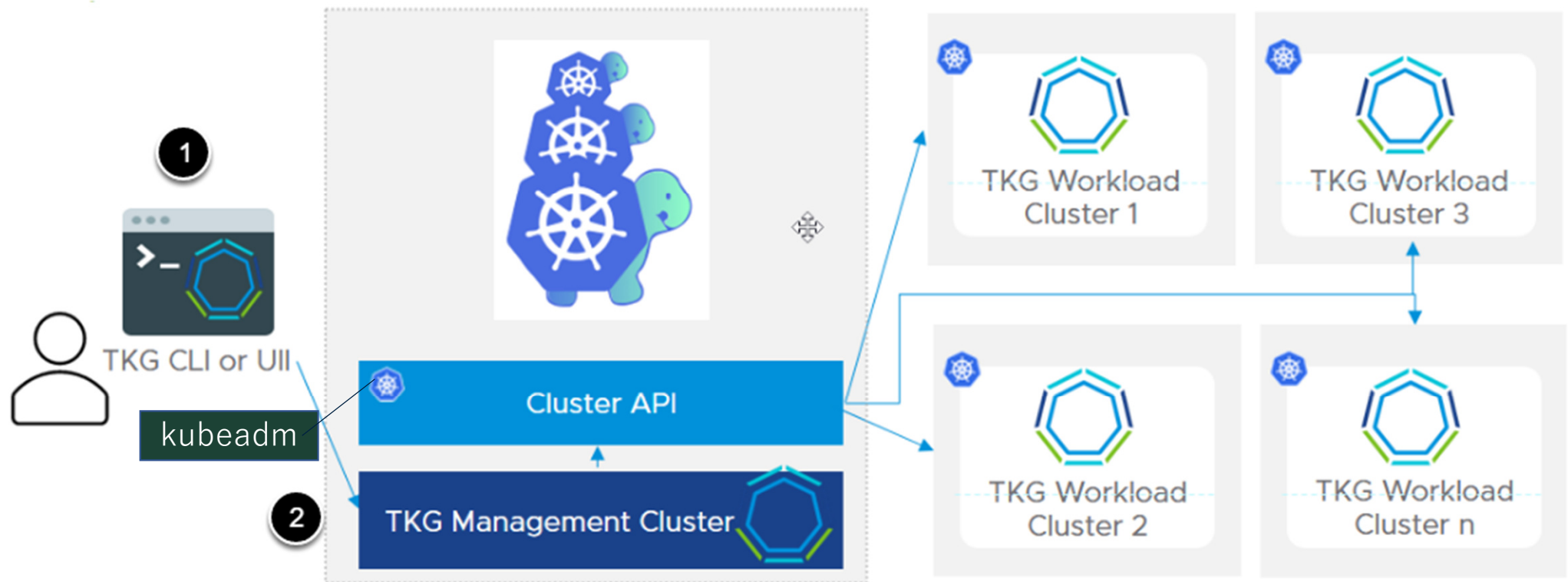
# vSphere with Tanzu

Run  
実行




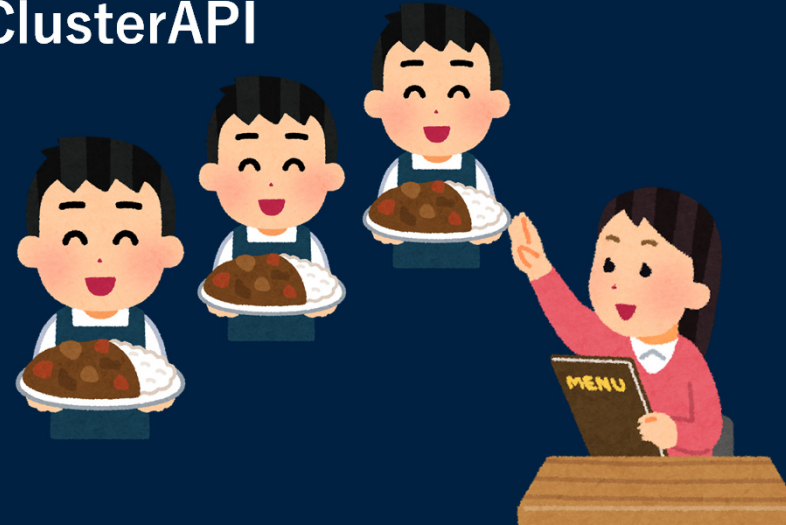
TKG

# TKG(Tanzu Kubernetes Grid)



# TKG(kubeadm/ClusterAPI)

## kubeadmとClusterAPIの違い

|      | kubeadm  | ClusterAPI  |
|------|--|---|
|      |  |  |
| API  | 命令型 (手続き型)   | 宣言型   |
| 使用場面 | 1つのクラスタのみ作成する際に使用  | 複数のクラスタを作成する際に使用  |

TKG

TKGが使用できるベンダー



**KEL**

# TKGがサポートするOSSの例



監視



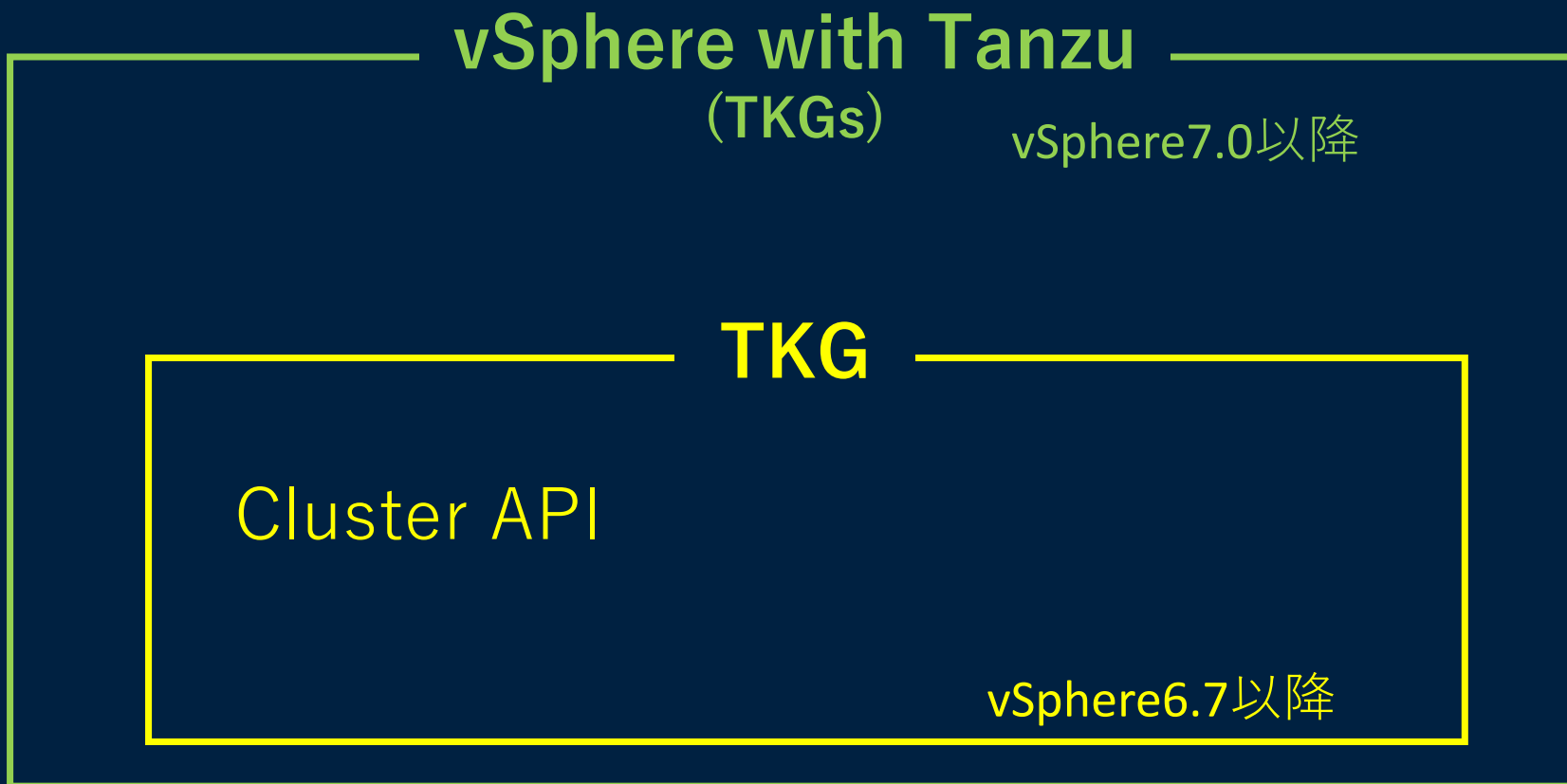
ログ管理



バックアップ

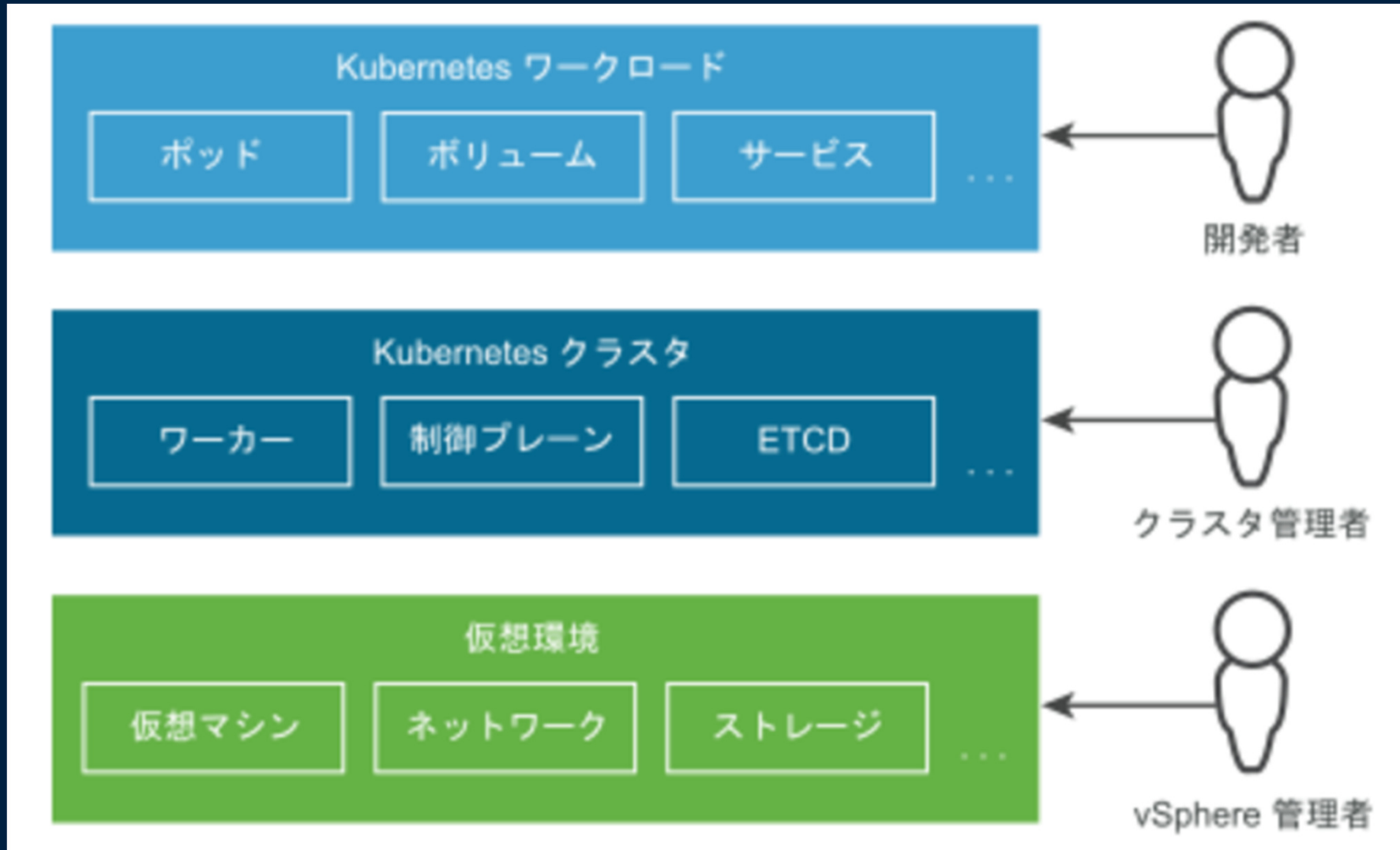
# vSphere with Tanzu

Run  
実行

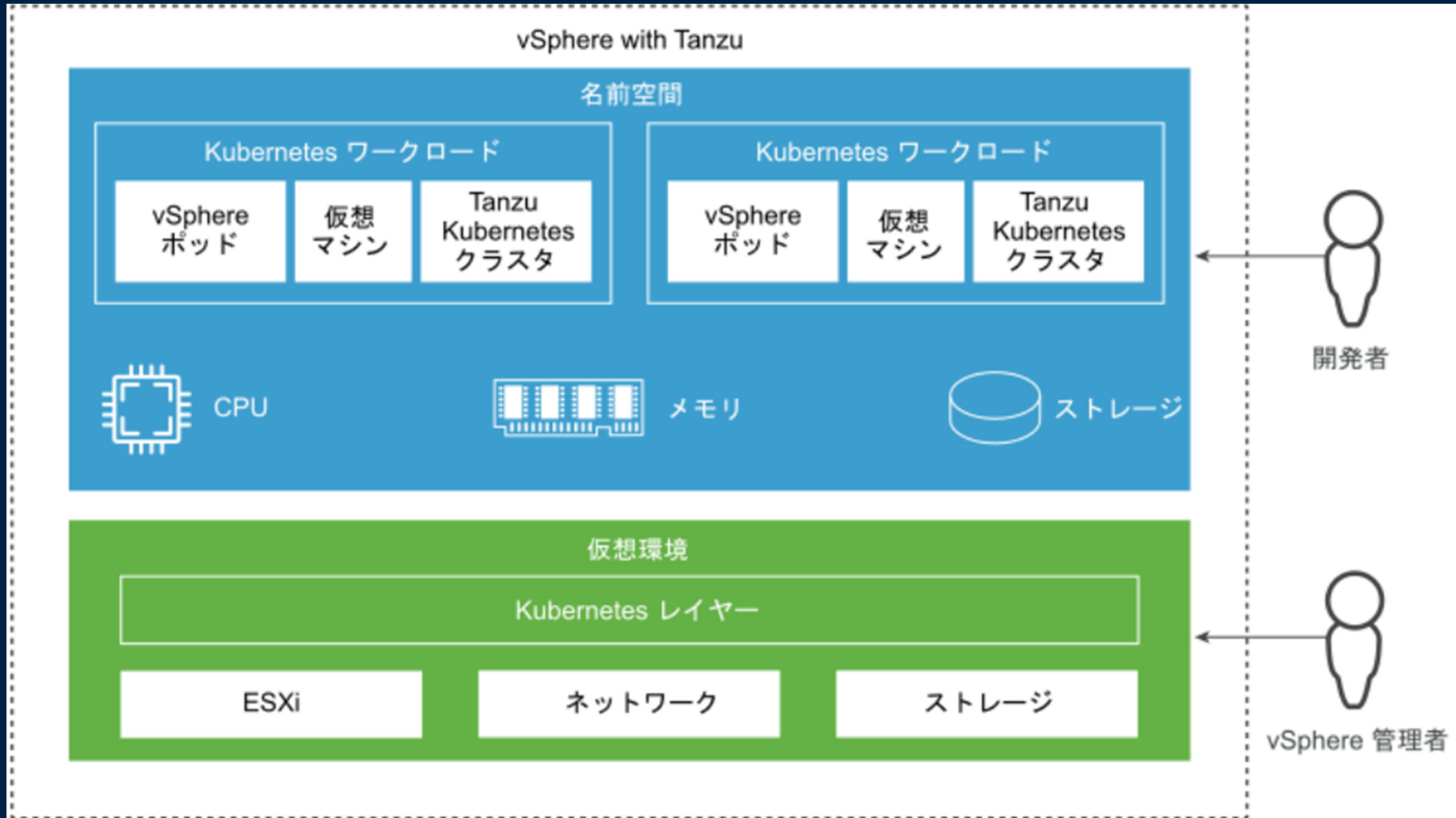




# vSphere with Tanzuを使用していないk8s環境の場合



# vSphere with Tanzuを使用した場合



# vSphere with Tanzu (TKGs) 操作画面

管理者

vSphere Client (GUI)

Namespace  
(リソースプール、アクセスコントロール)

vSphere with Tanzu

ログイン

開発者

vSphere Client (CLI)

クラスタにアプリをデプロイ、削除

The screenshot shows the vSphere Client interface for a Tanzu Kubernetes Grid (TKG) cluster. It is divided into three main sections: Status, Permissions, and Storage. The Status section shows the cluster is 'Running' and 'Active', with location information 'RegionA01-COMP01' and 'vcsa-01a.corp.local'. The Permissions section shows 'Can view' (No users have permission to only view namespaces) and 'Can edit' (Administrator). The Storage section shows 'Persistent Volume Claims' and 'Storage Policies' (kubernetes | No limit). At the bottom, there are buttons for 'MANAGE PERMISSIONS' and 'EDIT STORAGE'.

```
root@linux-01a:~# kubectl vsphere login --server=https://192.168.130.129
Password:
Logged in successfully.

You have access to the following contexts:
192.168.130.129
demo-app-01
tkg-cluster-01

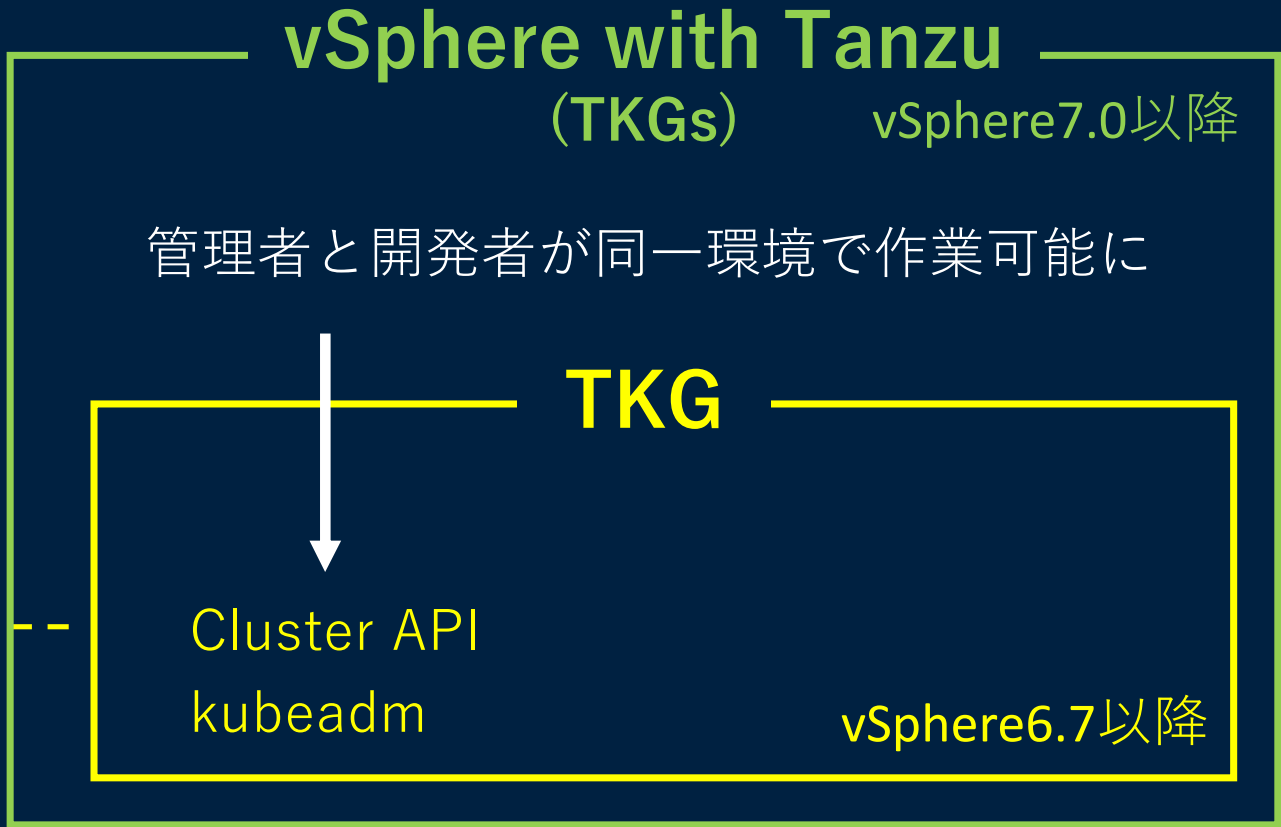
If the context you wish to use is not in this list, you may need to try
logging in again later, or contact your cluster administrator.

To change context, use `kubectl config use-context <workload name>`
```

# vSphere with Tanzu概要

**GUI**でnamespaceの作成が可能  
TKGにのみ使用可能な技術

**CLI**での操作  
AWSやAzure等でも使用可能なK8s構築ツール



## vSphere with Tanzu(NSX-Tあり/なしの場合での違い)

|          | VCF with Tanzu<br>(NSX-T有り)   | vSphere with Tanzu<br>(NSX-T無し)  |
|----------|---|--|
| バージョン    | vSphere7.0以降  | vSphere7.0 U1以降  |
| 最小限サーバ台数 | 3台  | 1台   |
| 機能       | <ul style="list-style-type: none"><li>・ VCF上での利用が前提</li><li>・ vSphere Pod機能が利用可</li><li>・ GUIで確認可</li></ul> | <ul style="list-style-type: none"><li>・ vSphere Pod機能は利用不可<br/>(TKG必須)</li><li>・ GUIで確認不可能</li></ul> |

ポットデプロイ方法

1. vSphere Pod
2. TKG

# 1. 検証案

- 目的

vSphere with Tanzuの知見を深め、将来的なビジネスへ活用する。  
また、vSphere with Tanzu検証環境を社員に提供することで、機能理解の一助を行う。

- 目標

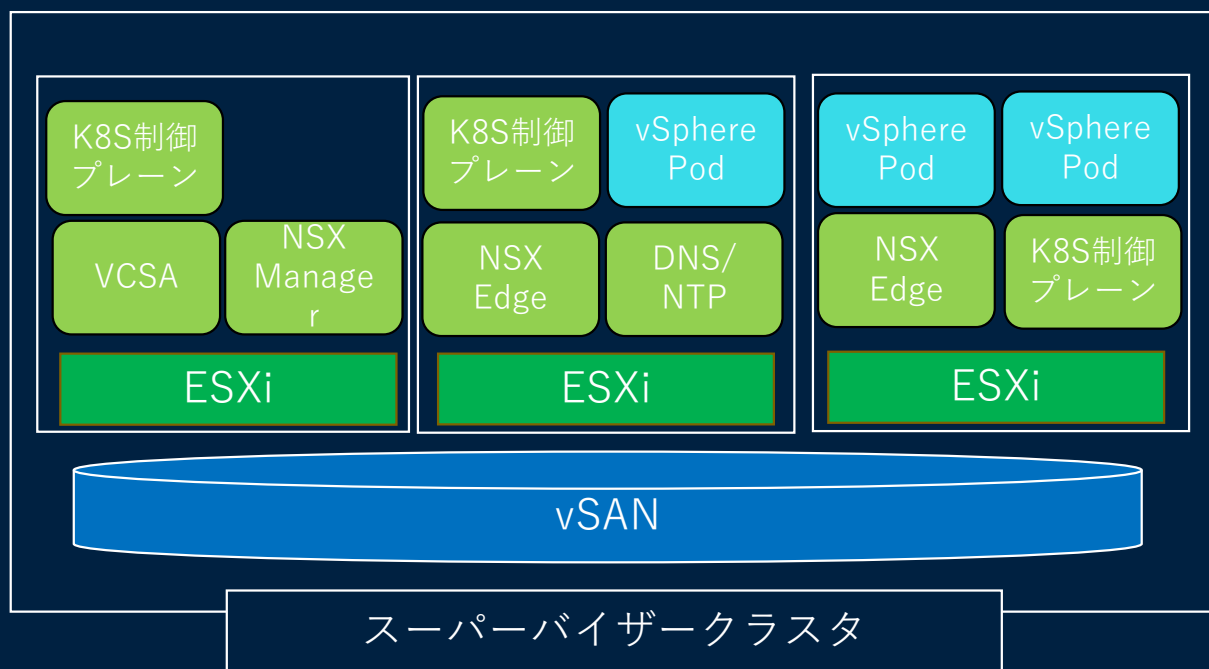
- ①vSphere with Tanzu環境を構築する。
- ②vSphere with Tanzuの基本機能と操作性を確認し、標準Kubernetesとの運用方法の比較を行う。

- 期間

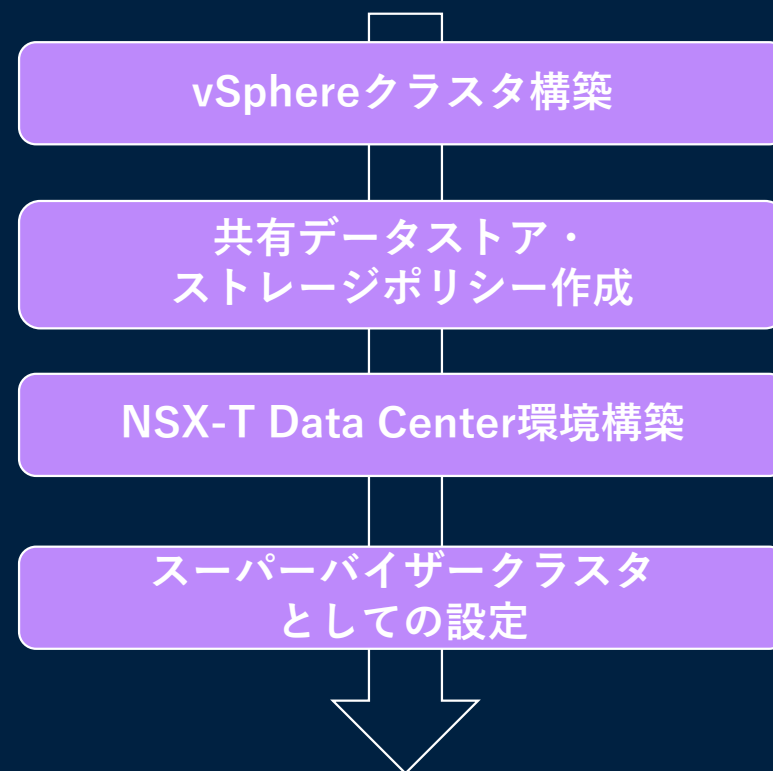
2020/10/7 ~ 2021/2/26

## 2. 検証目標①

スーパーバイザークラスタ構成イメージ



ワークフロー概要



### 3. 検証結果①

検証機納期の影響で現在構築中

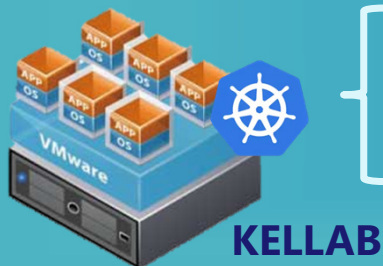


来期もチームの一部メンバーで  
Tanzu環境構築を引き続き実施



## 4. 検証目標②

### 標準k8s環境



```
[root@linux-test1 ~]# minikube status
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
[root@linux-test1 ~]# minikube stop
Stopping "minikube" in node ...
Unable to stop VM: Temporary Error: stop: Maximum number of retries (60) exceeded
minikube is exiting due to an error. If the shown message is not useful, open an issue:
https://github.com/kubernetes/minikube/issues/new/choose
[root@linux-test1 ~]#
[root@linux-test1 ~]#
[root@linux-test1 ~]# minikube status
host: Error
kubelet: Error
apiserver: Error
kubeconfig: Error
```

1. KELLAB上に標準k8sを構築
2. 標準k8s環境(CLI)でのPodデプロイを実施

VS

### vSphere with Tanzu環境



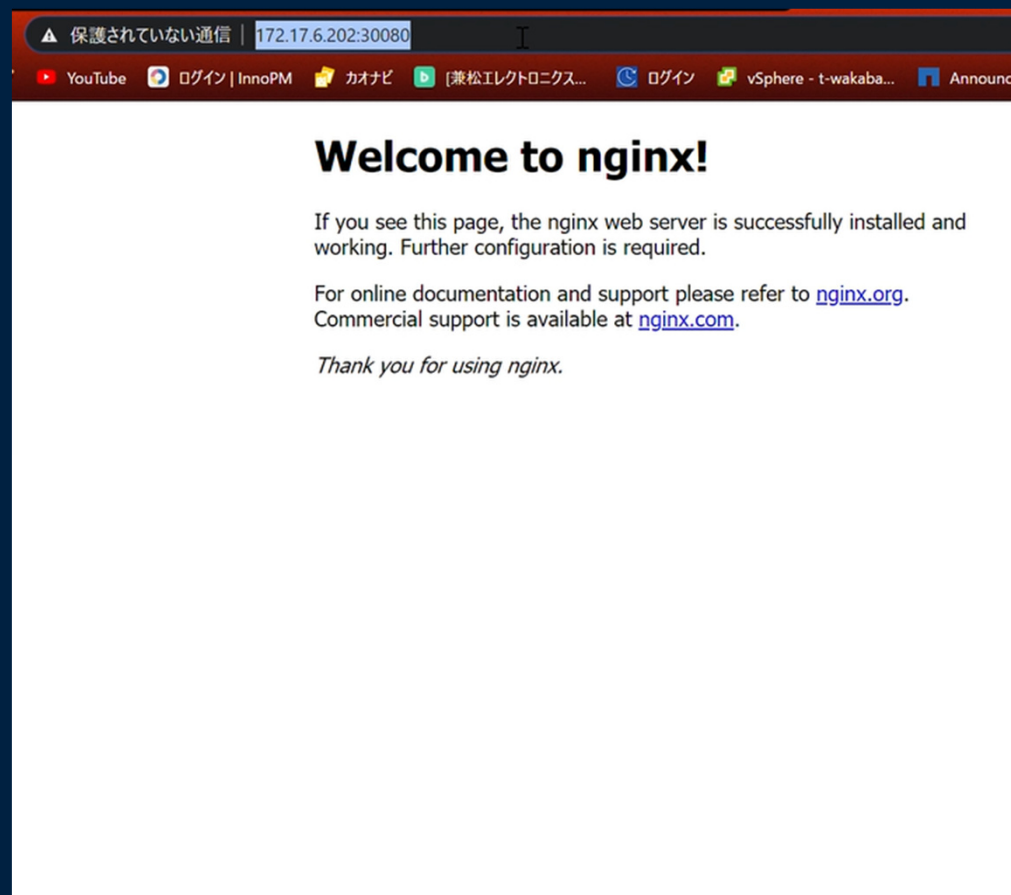
1. サーバ上にvSphere with Tanzuを構築(検証目標①)
2. vSphere with Tanzu(GUI)でのPodデプロイを実施

## 5. 検証結果②(k8s環境)

### 設定実施画面

```
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
selector:
  matchLabels:
    app: nginx-deployment
strategy: {}
template:
  metadata:
    creationTimestamp: null
    labels:
      app: nginx-deployment
  spec:
    containers:
      - image: nginx
        name: nginx
        resources: {}
status: {}
[root@k8s-master ~]# cat service.yaml
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    app: nginx-deployment
  name: node-port
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
    nodePort: 30080
  selector:
    app: nginx-deployment
  type: NodePort
status:
  loadBalancer: {}
[root@k8s-master ~]# kubectl apply -f nginx.yaml
deployment.apps/nginx-deployment created
```

### nginxと通信できることを確認



## 6. 検証結果② (Tanzu環境)

Tanzu環境のPodデプロイ

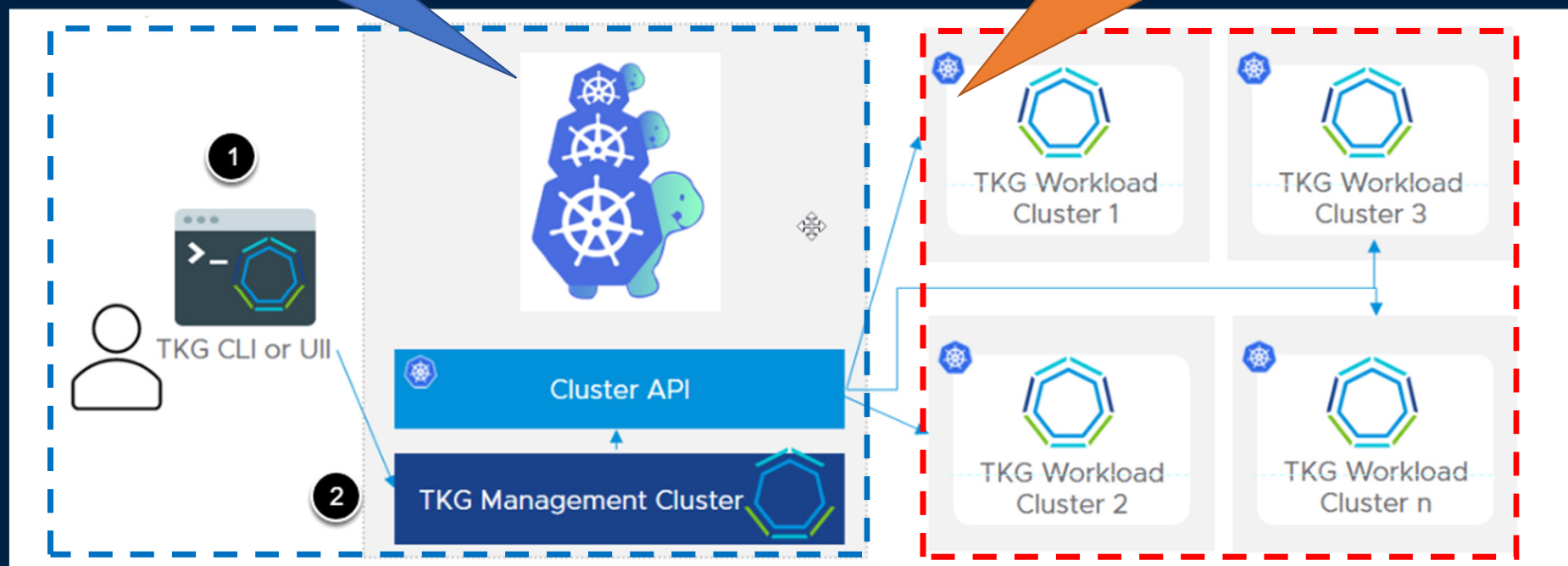


基本的にk8s環境と同じくCLIで実施

## 6. 検証結果② (Tanzu環境)

Management Clusterの  
リソース閲覧や構築は  
GUIで可能

Workload Clusterはリソース閲覧のみ  
GUIで可能、構築やPodデプロイは従  
来通りCLIで実施する必要あり



# 7. 検証結果を基にした今後のアクション

- 今後のアクション

vSphere with Tanzu環境を社員に向けた検証用環境として提供する。

→**今期構築ができなかったため、来期実施予定**

- 具体的なプラン

①KELLABからvSphere with Tanzu環境への接続を可能にする。

②検証で実施した操作手順書や、陥りやすいトラブルとその解決策を資料にまとめて展開する。

# KEL

ありがとうございました