

# Ops Track

Red Hat K.K.  
Specialist Solutions Architect  
Mayumi Koshimizu  
2024/02/08

# Modern Application Development Roadshow

# ハンズオンコース 概要説明

# Modern Application Development Roadshow Ops Track

- Module 1
  - OpenShift管理者の基礎知識
- Module 2
  - Red Hat Advanced Cluster Security for Kubernetesを用いたクラウド環境のセキュリティ強化
- Module 3
  - Red Hat Advanced Cluster Management for Kubernetesを用いたマルチクラスター管理

# Module 1

OpenShift管理者の基礎知識

14:10 - 15:30

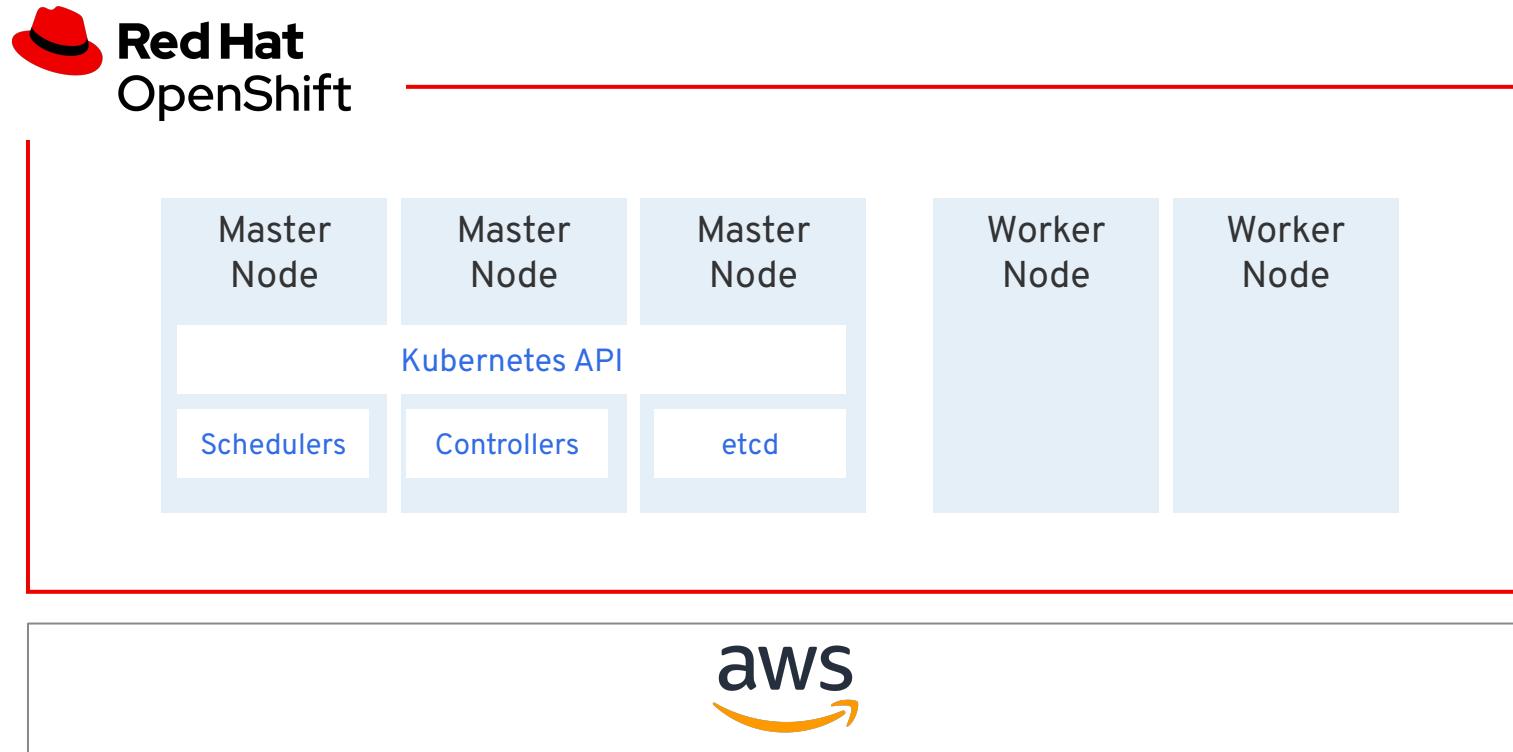
# OpenShift管理者の基礎知識

## 演習環境の概要

1. OpenShiftクラスタのインストールの検証
2. アプリケーション管理の基礎
3. アプリケーションストレージの基礎
4. MachineSets, Machines, and Nodes
5. インフラストラクチャノードとOperator
6. 外部認証プロバイダ(LDAP)の設定
7. OpenShift Monitoring
8. プロジェクト・リクエスト・テンプレートとクォータ/制限
9. OpenShift の Network Policy ベースの SDN
10. Projectのセルフプロビジョニングの無効化
11. クラスタリソースのクォータ
12. Taint と Toleration

# OpenShiftクラスタのインストールの検証

ワークショップの環境は、AWS上にIPI(Installer-provisioned infrastructure)インストールされています



## Master Node (3 nodes)

Master Nodesには、コンテナを制御するコンポーネントと、クラスタの状態を構成管理するデータ(etcd)があります。

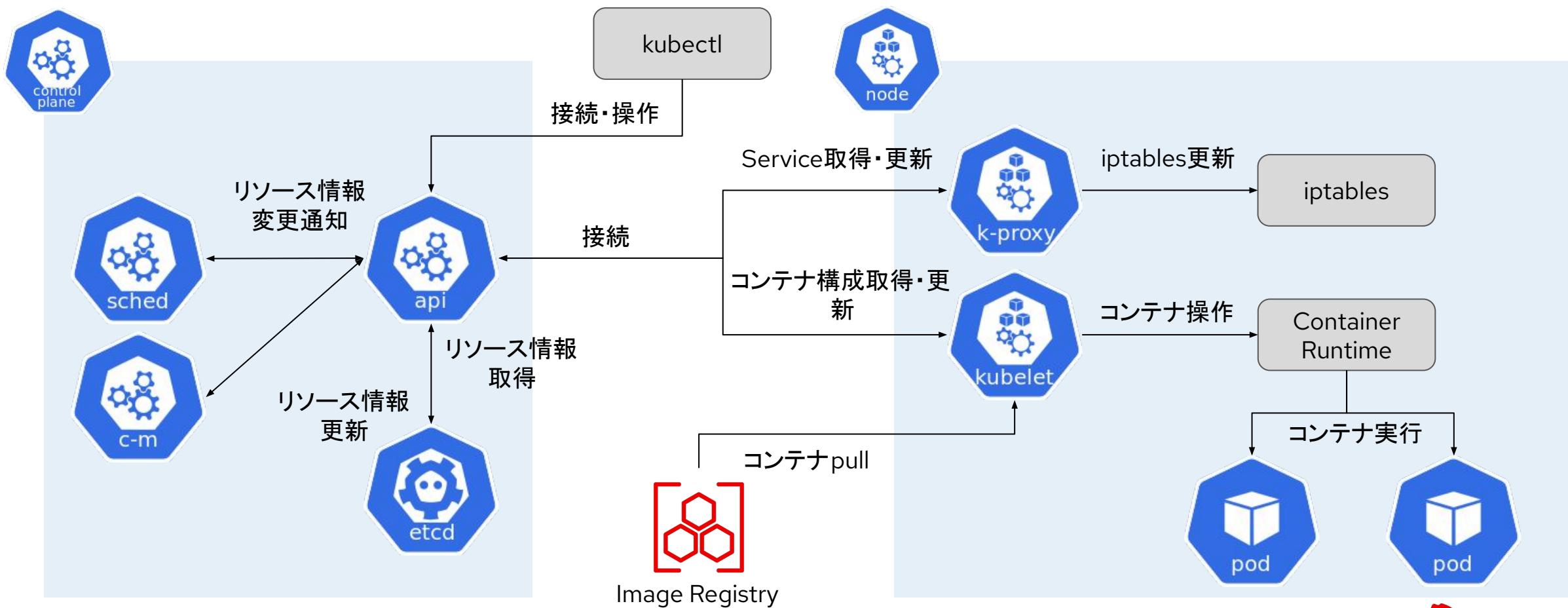
**Controllers:** リソースのデプロイを行う機能

**Schedulers:** リソースの空き状況を管理する機能

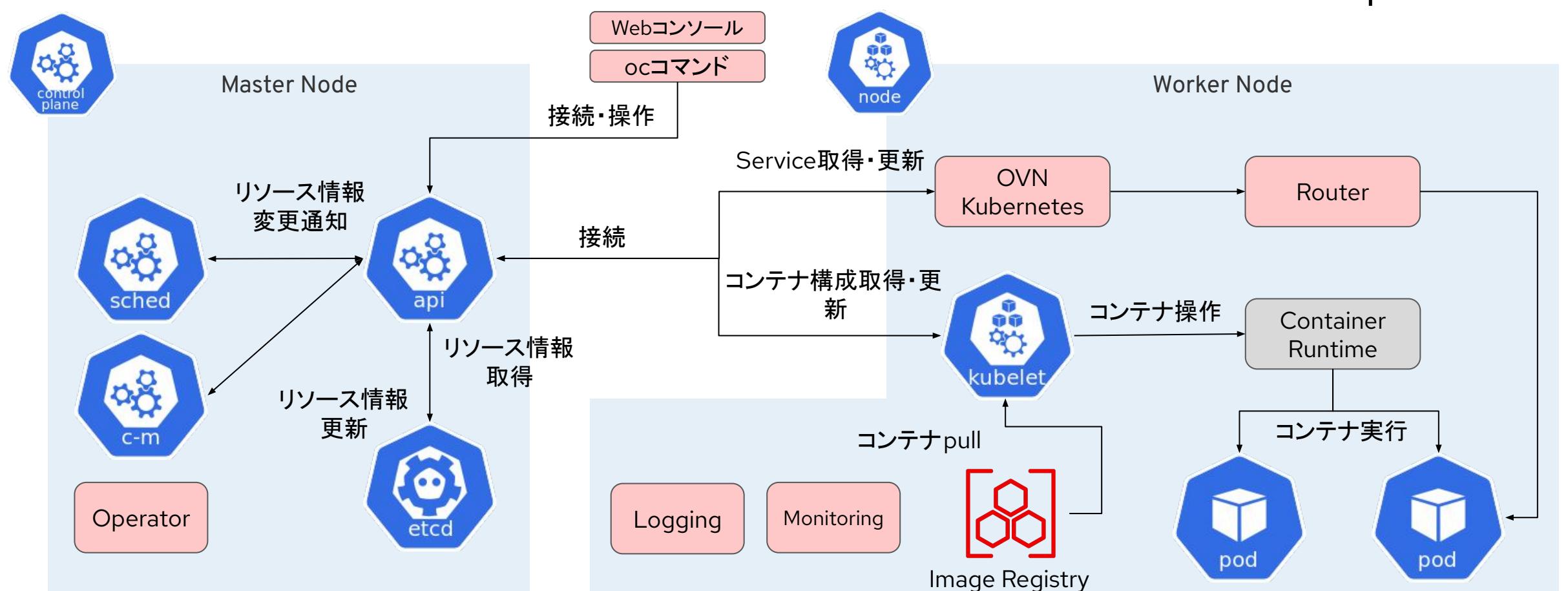
## Worker Node (2~N nodes)

Worker Nodesでは、スケジュールされたコンテナがデプロイされ、コンテナを死活監視します。

# Kubernetesのアーキテクチャ

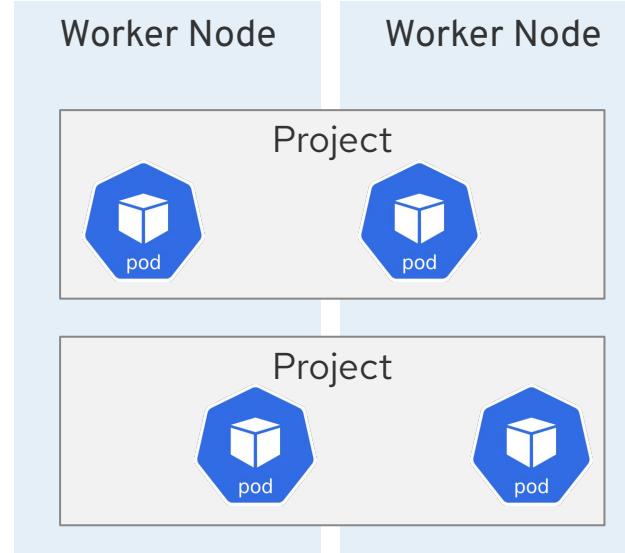


# OpenShiftのアーキテクチャ



# アプリケーション管理の基礎

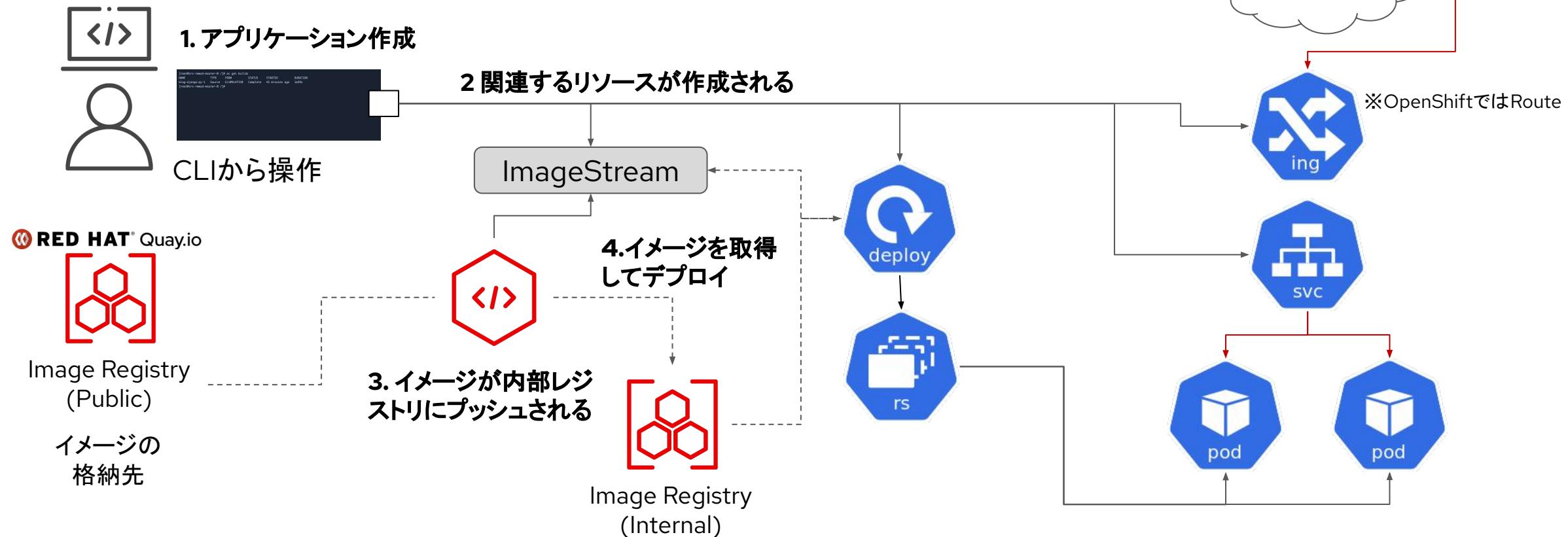
OpenShiftのコアとなる概念について学びます



リソース	概要
Project	Project単位でリソースを分離。管理の観点からは、各Projectはテナントのように考えることができます
Pod	ホスト上に一緒にデプロイされた1つまたは複数のコンテナ。PodはOpenShiftで定義、デプロイ、管理できるコンピュートリソースの最小の単位です
Service	OpenShift環境内でPodのようなグループを見つけるための抽象化レイヤーを提供。OpenShift環境内からPodにアクセスする必要があるものとの間の内部プロキシ/ロードバランサーとしても機能します
Deployment	OpenShift内に何をどのようにデプロイするかを定義します
ReplicaSet	必要な数のPodを確実に存在させるために使用されます。ReplicaSetはOpenShiftが自己修復する方法を提供します
Route	OpenShift外のクライアントがOpenShift内で実行されているアプリケーションにアクセスする方法を提供します

# アプリケーション管理の基礎

イメージを利用したアプリケーションのデプロイを行い作成されたリソースを確認します



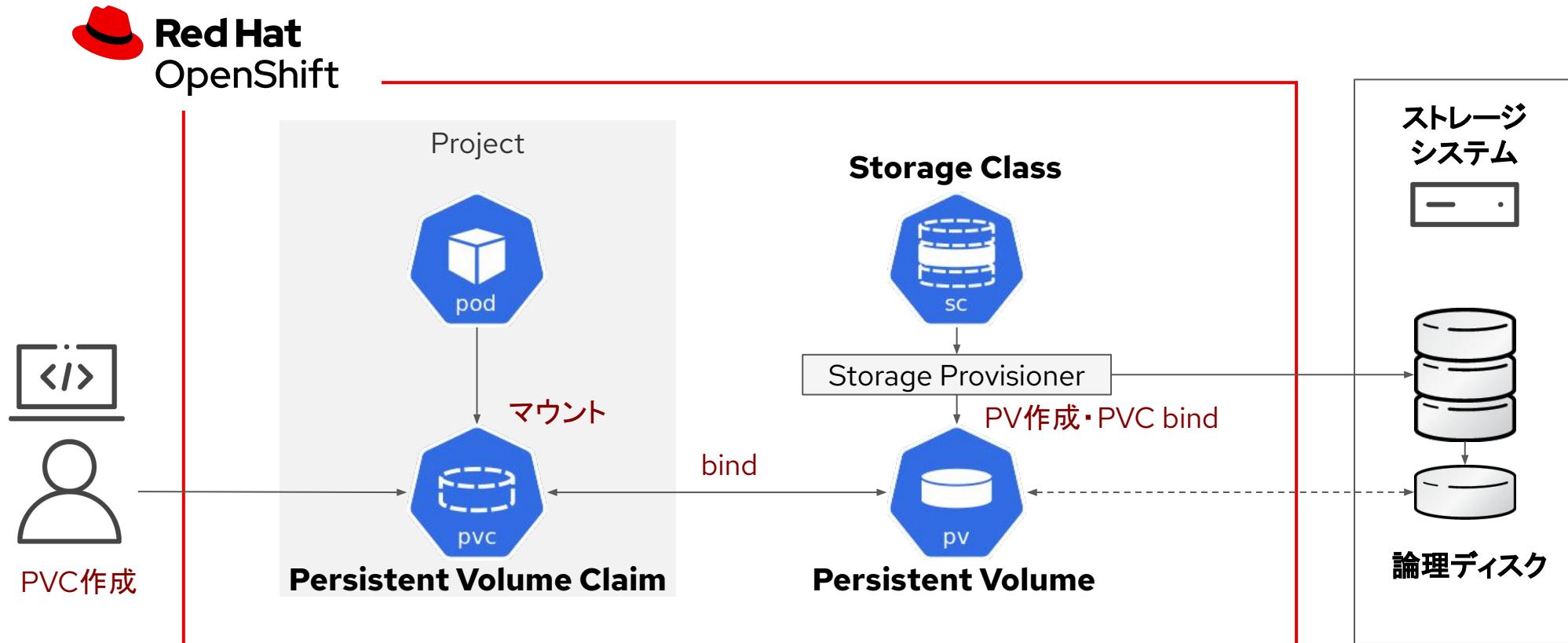
# アプリケーション管理の基礎

イメージを利用したアプリケーションのデプロイを行い作成されたリソースを確認します

1. プロジェクト作成: `oc new-project app-management`
2. アプリケーション作成: `oc new-app quay.io/openshiftroadshow/mapit`
3. アプリケーションに関連するリソースが作成されます  
各リソースの内容を確認します
4. routeを作成し、外部からアクセスできることを確認します

# アプリケーションストレージの基礎

OpenShiftにおける永続ボリューム(Persistence Volume)の動作を確認します



# アプリケーションストレージの基礎

OpenShiftにおける永続ボリューム(Persistence Volume)の動作を確認します

1. アプリケーション mapit で下記の内容の永続ボリュームを使用します

PVC名 : mapit-storage

ReadWriteOnce モード

サイズ : 1Gi

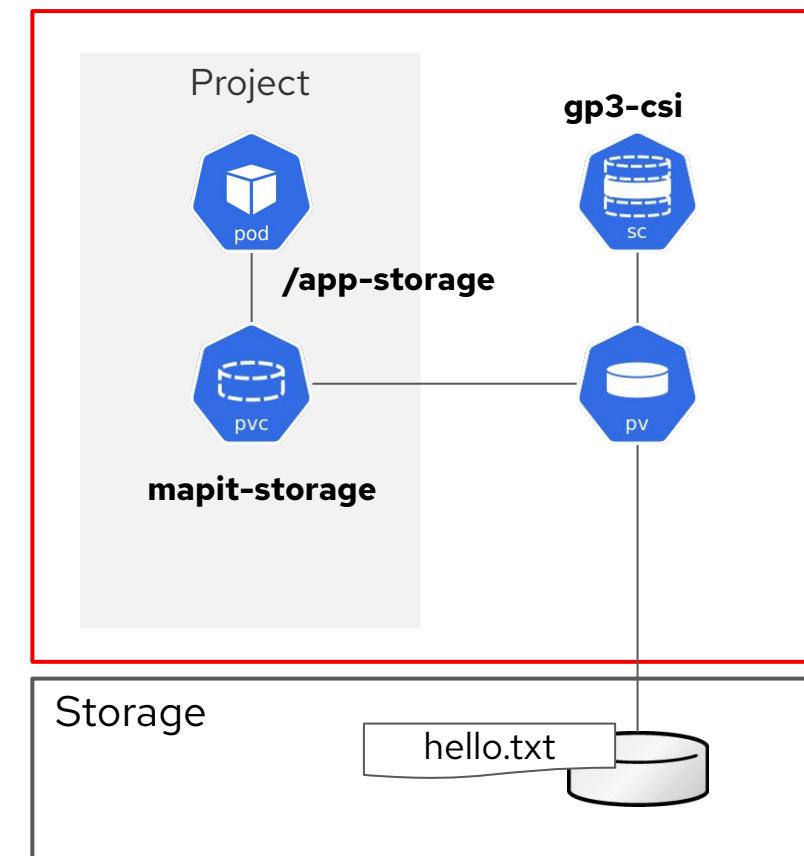
mount-path : /app-storage

2. 永続ボリュームにファイルを作成します

3. 稼働している Pod を削除します

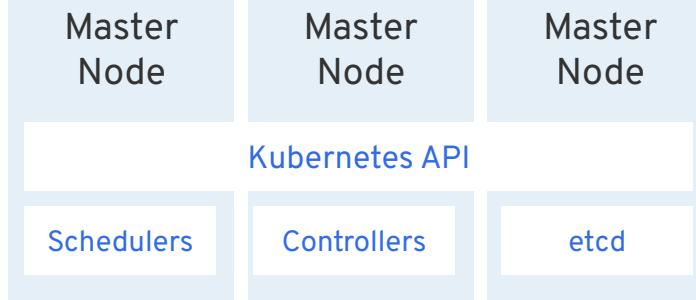
4. 新しい Pod がすぐに稼働します

5. 永続ボリュームに作成したファイルを継続して利用できることを確認します

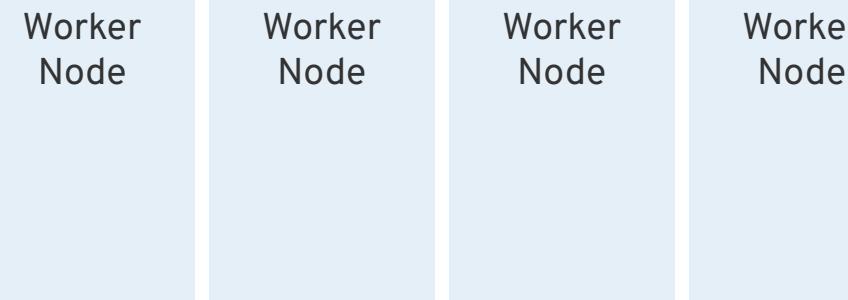


# MachineSet, Machine, and Node

MachineSet/Machineを利用して、ノードを環境に追加(削除)します



cluster-9fddq-vk46x-worker-ap-southeast-1c  
cluster-9fddq-vk46x-worker-ap-southeast-1b  
cluster-9fddq-vk46x-worker-ap-southeast-1a



ReplicaSetがPodの数を  
コントロール

**ReplicaSet**

**Pod**

MachineSetがMachine(node)の  
数をコントロール

**MachineSet**

**Machine**

## MachineSet

cluster-9fddq-vk46x-worker-ap-southeast-1a	1	1	1	31h
cluster-9fddq-vk46x-worker-ap-southeast-1b	1	1	1	31h
cluster-9fddq-vk46x-worker-ap-southeast-1c	0	0	0	31h

## Machine

cluster-9fddq-vk46x-worker-ap-southeast-1a-hpwng	Running
cluster-9fddq-vk46x-worker-ap-southeast-1b-d7dll	Running

cluster-9fddq-vk46x-worker-ap-southeast-1a	1	1	1	32h
cluster-9fddq-vk46x-worker-ap-southeast-1b	1	1	1	32h
cluster-9fddq-vk46x-worker-ap-southeast-1c	2	2	2	32h

cluster-9fddq-vk46x-worker-ap-southeast-1a-hpwng	Running
cluster-9fddq-vk46x-worker-ap-southeast-1b-d7dll	Running
cluster-9fddq-vk46x-worker-ap-southeast-1c-l5jjw	Provisioned
cluster-9fddq-vk46x-worker-ap-southeast-1c-zm2dt	Provisioned

## Node

ip-10-0-198-39.ap-southeast-1.compute.internal
ip-10-0-209-58.ap-southeast-1.compute.internal



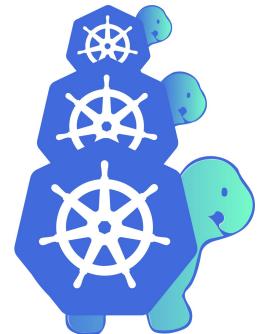
# Machine-API

Machine-APIはアップストリームのKubernetes Cluster APIをOpenShiftに機能実装したもの

## Cluster API

kubernetesクラスタのプロビジョニング、アップグレード、運用を簡素化するための宣言型 APIとツールの提供にフォーカス  
クラスタのライフサイクル管理を自動化する  
多種多様なインフラストラクチャ環境において、一貫性のある再現可能なクラスタのデプロイを可能にする

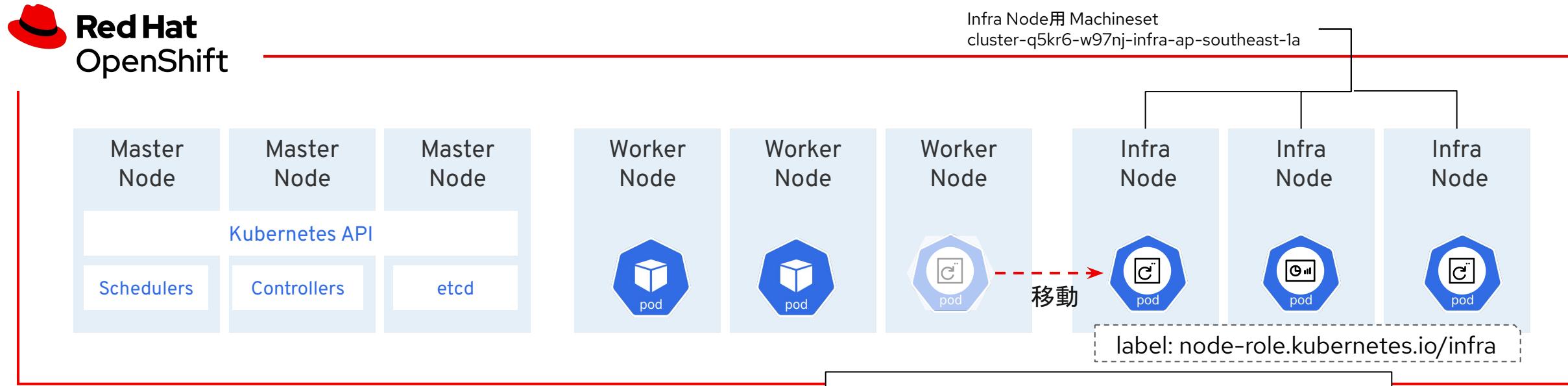
- ・ノードのスケールイン・スケールアウトがコマンドラインから可能
- ・ノードのスケール調整そのものを自動化( autoscale)
- ・ノードのローリングアップデート



参考資料 : <https://www.slideshare.net/ssuser804f1a/machine-configoperator>

# インフラストラクチャノードと Operator

Infra Nodeを作成し、Worker Node上で稼働しているPod(Router, Registry等)をInfra Nodeに移動させます



Infra Nodeには、OpenShiftの追加サービスを管理するためのコンポーネントが配置されます

Infra Nodeに配置できる機能

- ・Router
- ・OpenShift-included Registry
- ・OpenShift cluster monitoring
- ・OpenShift log aggregation
- など

```
apiVersion: operator.openshift.io/v1
kind: IngressController
metadata:
  finalizers:
  - ingresscontroller.operator.openshift.io/finalize
  name: default
  namespace: openshift-ingress-operator
spec:
  nodePlacement:
    nodeSelector:
      matchLabels:
        node-role.kubernetes.io/infra: ""
```

例) Router PodがInfra Nodeで稼働するようnodeSelectorを指定



# インフラストラクチャノードと Operator

例) cluster-5gqm-pkmgl-worker-ap-southeast-1a

```
spec:  
replicas: 1  
selector:  
matchLabels:  
    machine.openshift.io/cluster-api-cluster: cluster-5gqm-pkmgl  
    machine.openshift.io/cluster-api-machineset: cluster-5gqm-pkmgl-worker-ap-southeast-1a  
template:  
metadata:  
labels:  
    machine.openshift.io/cluster-api-cluster: cluster-5gqm-pkmgl  
    machine.openshift.io/cluster-api-machine-role: worker  
    machine.openshift.io/cluster-api-machine-type: worker  
    machine.openshift.io/cluster-api-machineset: cluster-5gqm-pkmgl-worker-ap-southeast-1a  
spec:  
lifecycleHooks: {}  
metadata: {}  
providerSpec:  
value:  
ami:  
    id: ami-0f827a1be73b9de83 AmazonマシンイメージID
```

レプリカ数  
クラスタID  
machineset名  
クラスタID  
machineset名

# インフラストラクチャノードと Operator

例) cluster-5gqmq-pkmg1-worker-ap-southeast-1aをもとにInfra Node用の新しいmachinesetを作成しスケールアウト

## machineset-generator.shの内容

```
#!/bin/bash
export AMI=$(oc get machineset -n openshift-machine-api -l 'machine.openshift.io/os-id!=Windows' -o
jsonpath='{.items[0].spec.template.spec.providerSpec.value.ami.id}')
AMI情報
export CLUSTERID=$(oc get infrastructures.config.openshift.io cluster -o jsonpath='{.status.infrastructureName}')
クラスタID
export REGION=$(oc get infrastructures.config.openshift.io cluster -o jsonpath='{.status.platformStatus.aws.region}')
リージョン情報
export COUNT=${1:-3}
export NAME=${2:-workerocs}
export SCALE=${3:-1}

$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null 2>&1 && pwd )/machineset-cli -scale $SCALE -name $NAME -count $COUNT -ami
$AMI -clusterID $CLUSTERID -region $REGION
```

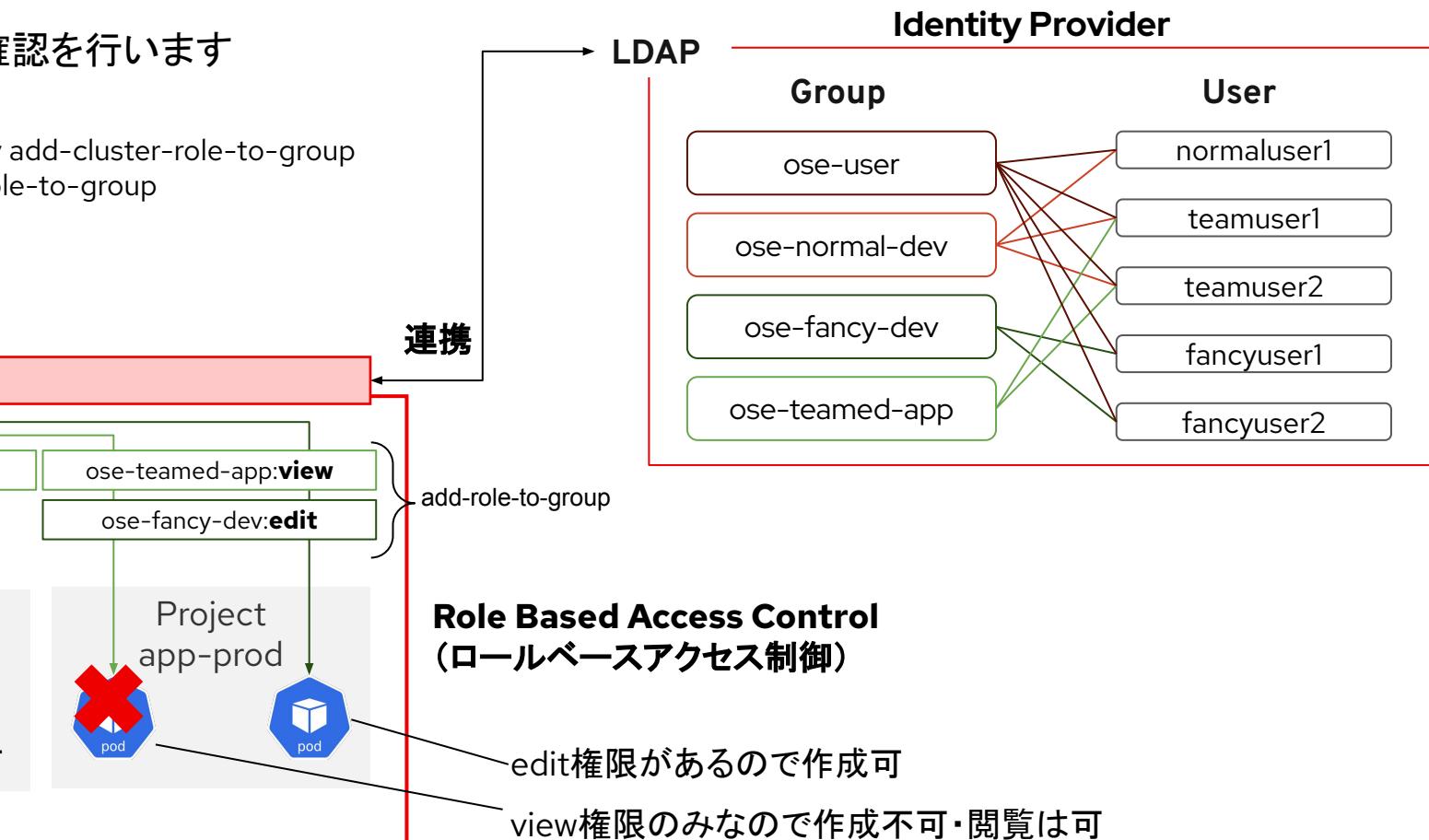
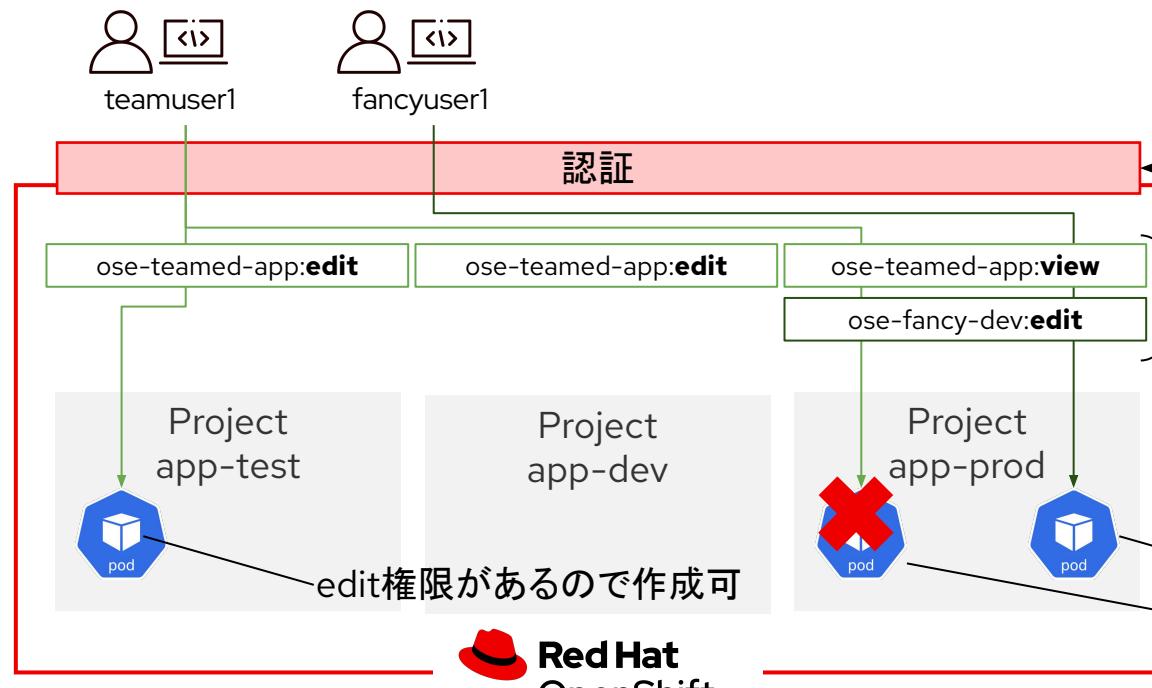
## Infra Node用Machineset作成とスケールアウトの実行

```
bash {{ HOME_PATH }}/support/machineset-generator.sh 1 infra 0 | oc create -f -
export MACHINESET=$(oc get machineset -n openshift-machine-api -l machine.openshift.io/cluster-api-machine-role=infra -o
jsonpath='{.items[0].metadata.name}')
oc patch machineset $MACHINESET -n openshift-machine-api --type='json' -p='[{"op": "add", "path":
"/spec/template/spec/metadata/labels", "value": {"node-role.kubernetes.io/worker": "", "node-role.kubernetes.io/infra": ""}} ]'
oc scale machineset $MACHINESET -n openshift-machine-api --replicas=3
```

# 外部認証プロバイダ(LDAP)の設定

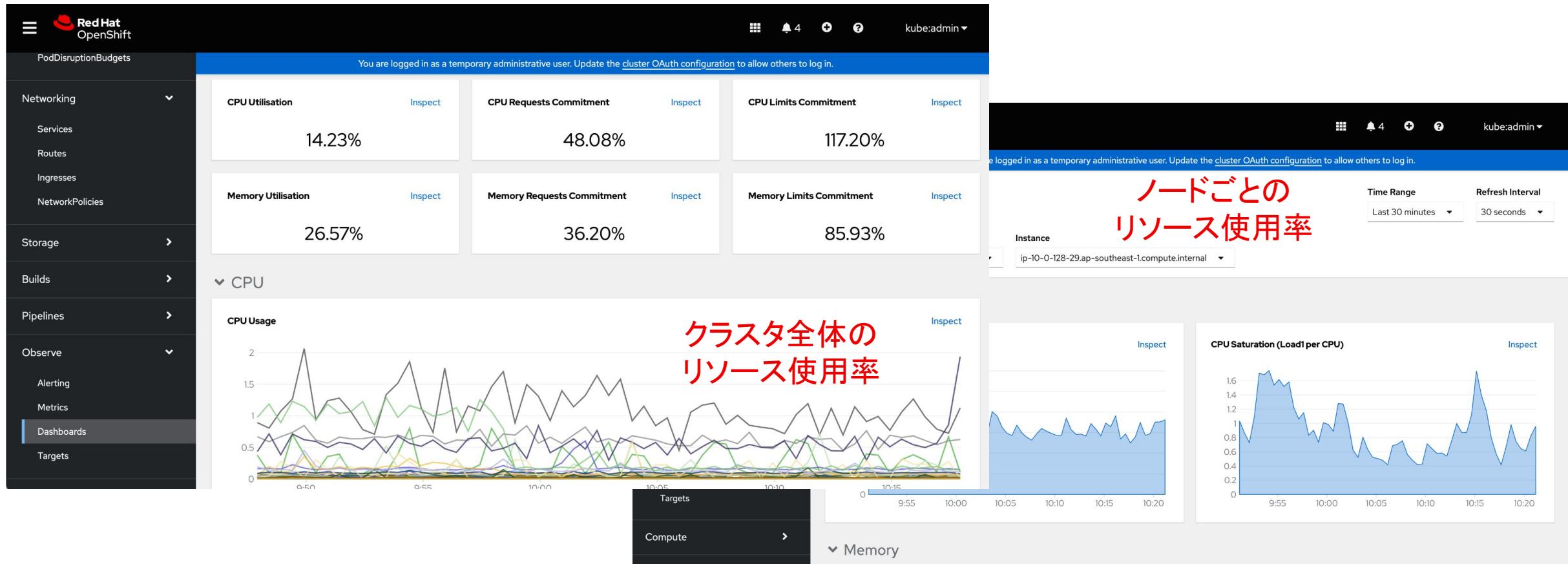
- LDAP連携の設定を行います
- Role Based Access Control設定と動作確認を行います

グループへのクラスタロールの追加: oc adm policy add-cluster-role-to-group  
 グループへのロールの追加: oc adm policy add-role-to-group



# OpenShift Monitoring

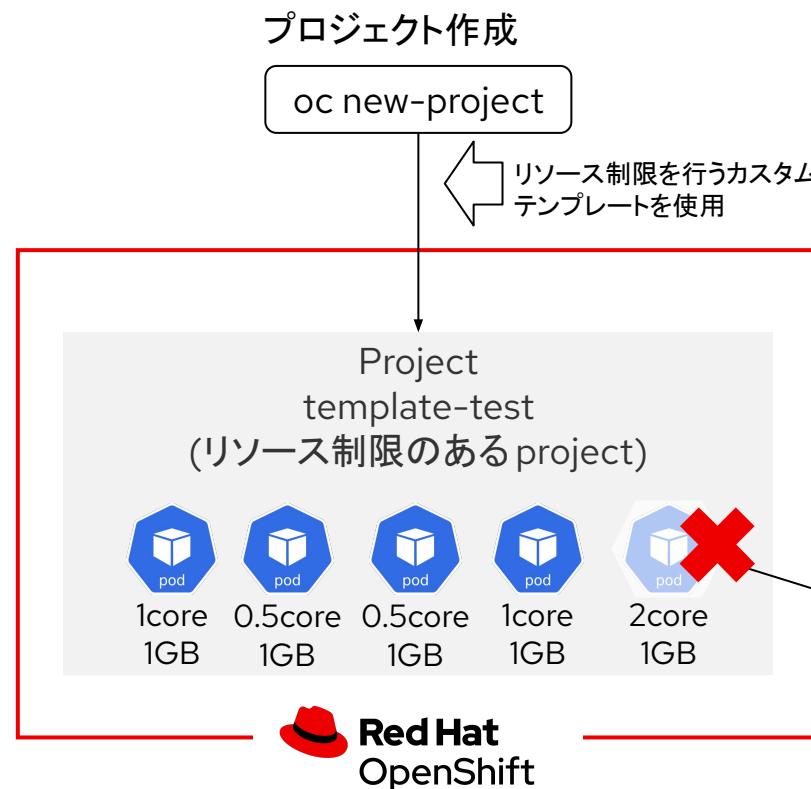
OpenShiftをインストールした時点で、クラスタに関する監視はCluster Monitoring(Prometheus)によって設定済みです。



# プロジェクト・リクエスト・テンプレートとクオータ/制限

プロジェクト内で使用するオブジェクト数やリソース(CPU,メモリ等)を制限するテンプレートを設定します

oc new-projectを利用するとデフォルトのTemplateを使用してプロジェクトを作成する(デフォルトではリソース制限なし)



## ResourceQuota

プロジェクトごとの総リソース消費量を制限

```

- apiVersion: v1
kind: ResourceQuota
metadata:
  name: ${PROJECT_NAME}-quota
spec:
  hard:
    pods: 10
    requests.cpu: 4000m
    requests.memory: 8Gi
    resourcequotas: 1
    requests.storage: 50Gi
    persistentvolumeclaims: 5
  
```

作成可能なPodの数

CPU総量

メモリ総量

ストレージ総量

PVC数

ResourceQuotaの制約で  
CPUは合計4coreまでなので追加で  
2coreのPodはデプロイ不可

## LimitRange

プロジェクト内のリソース毎の消費量を指定

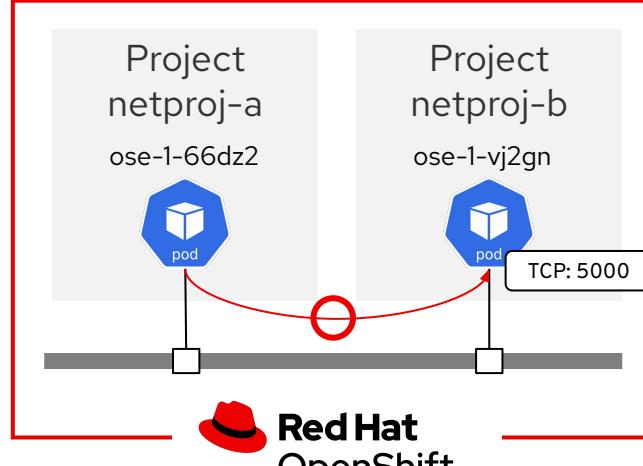
```

- apiVersion: v1
kind: LimitRange
metadata:
  name: ${PROJECT_NAME}-limits
  creationTimestamp: null
spec:
  limits:
  - type: Container
    max: limitやrequestに指定できる最大値
    cpu: 4000m
    memory: 1024Mi
    min: limitやrequestに指定できる最小値
    cpu: 10m
    memory: 5Mi
    default: 指定がない場合の最大値
    cpu: 4000m
    memory: 1024Mi
    defaultRequest: 指定がない場合の最小値
    cpu: 100m
    memory: 512Mi
  
```

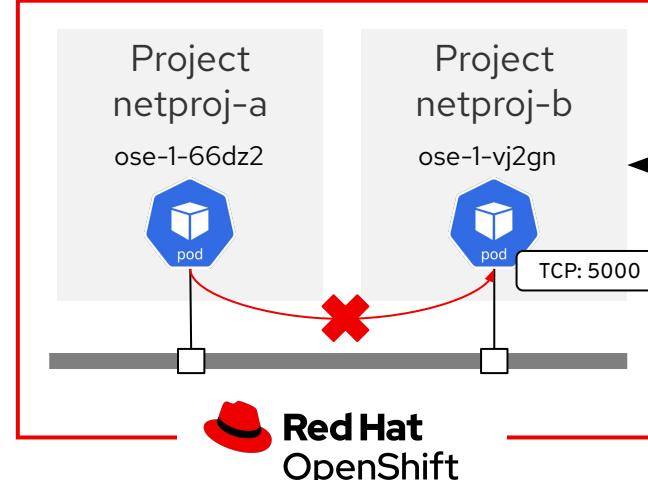
# OpenShift の Network Policy ベースの SDN

NetworkPolicyカスタムリソースを利用してプロジェクト間のネットワークを分離することができます

NetworkPolicyカスタムリソースなし  
デフォルトの状態ではすべての Pod間通信が可能



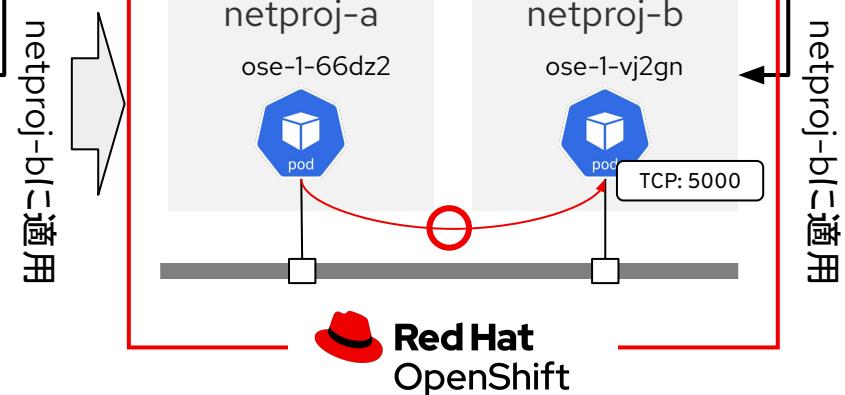
NetworkPolicyカスタムリソース  
上記の設定は ALL Deny に相当



```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: deny-by-default
spec:
  podSelector: すべてのPodが対象
  ingress: []
```

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: allow-tcp-5000-from-netproj-a-namespace
spec:
  podSelector:
    matchLabels:
      run: ose
      run: oseというlabelにマッチするPod
  ingress:
    - ports:
        - protocol: TCP
        - port: 5000
        - name: netproj-a
      from:
        - namespaceSelector:
            matchLabels:
              name: netproj-a
```

**project-aからのTCP5000を許可**



netproj-bに適用

# Projectのセルフプロビジョニングの無効化

- ・デフォルトでは認証済みのユーザはプロジェクトを作成することが可能です
- ・プロジェクトのセルフプロビジョニングを無効化することができます

## ClusterRoleBinding

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  creationTimestamp: "2022-06-07T02:52:33Z"
  name: self-provisioners
  resourceVersion: "9988"
  uid: 24f23fdd-4a10-40f2-8ca2-0068d4b51a8a
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: self-provisioner
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated:oauth
```

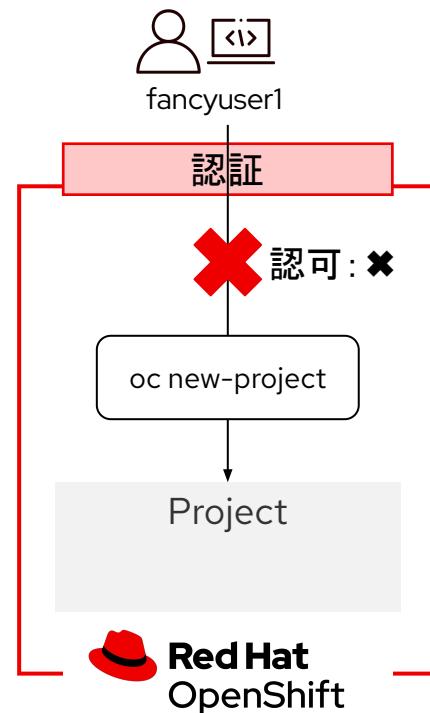
認証済みのユーザが属するグループは、  
self-provisioners roleにバインドされているた  
めセルフプロビジョニングが可能

## ClusterRoleBinding

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "false"
  creationTimestamp: "2022-06-07T02:52:33Z"
  name: self-provisioners
  resourceVersion: "66495"
  uid: 24f23fdd-4a10-40f2-8ca2-0068d4b51a8a
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: self-provisioner
```

oc patchコマンドで設定変更

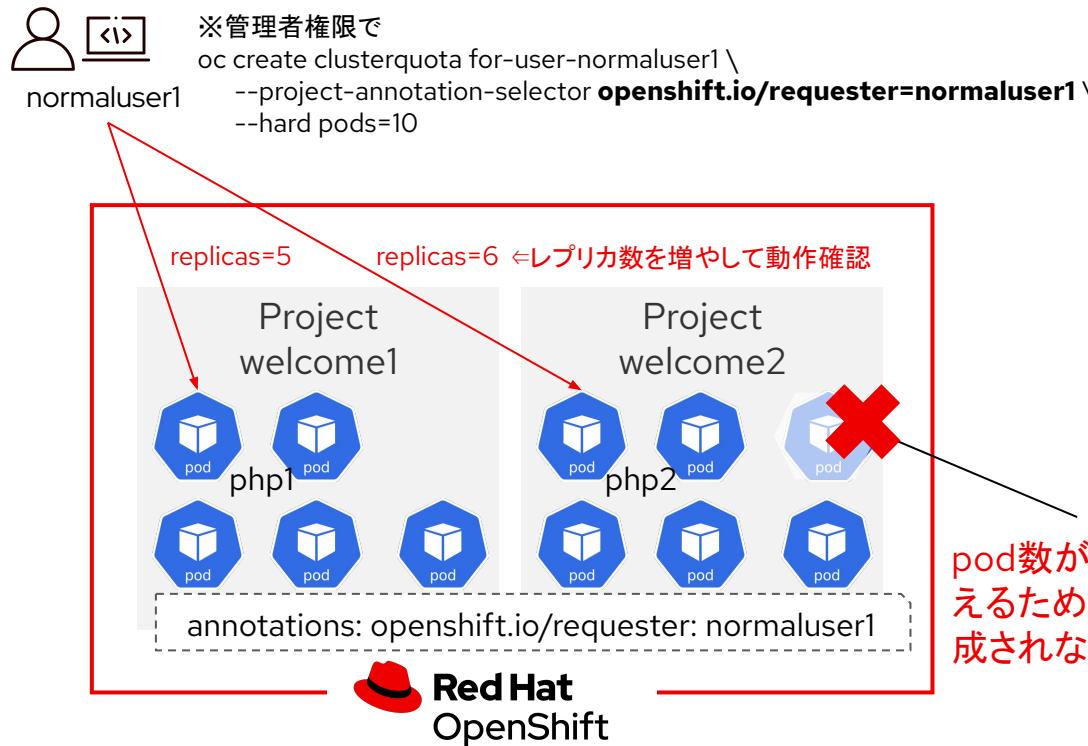
```
oc patch clusterrolebinding.rbac self-provisioners -p '{"subjects": null}'
oc patch clusterrolebinding.rbac self-provisioners -p '{"metadata": {"annotations": {"rbac.authorization.kubernetes.io/autoupdate": "false" }}}'
```



# クラスタリソースのクオータ

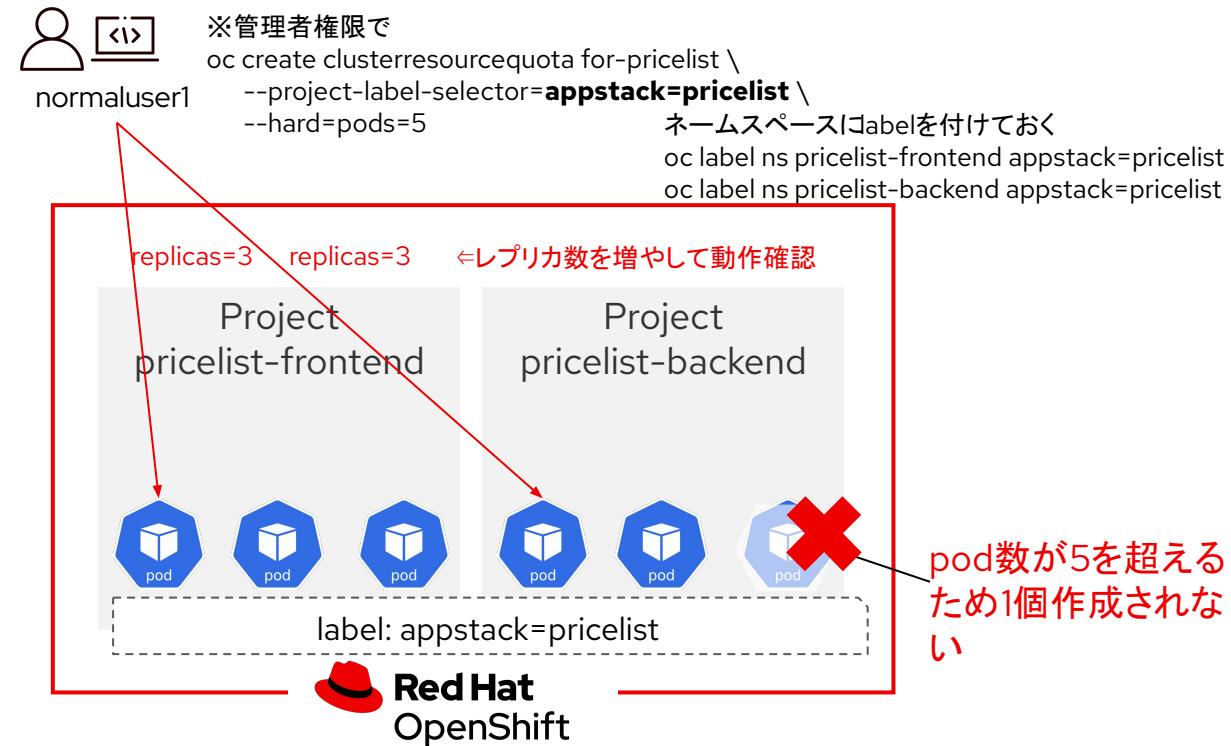
- 特定のユーザにクオータを設定します

例) 特定のユーザに対して Pod数を10以下に制限する  
 clusterresourcequotaを作成します  
[openshift.io/requester](https://openshift.io/requester)のannotationキーを利用します



- labelを利用してアプリケーションにクオータを設定します

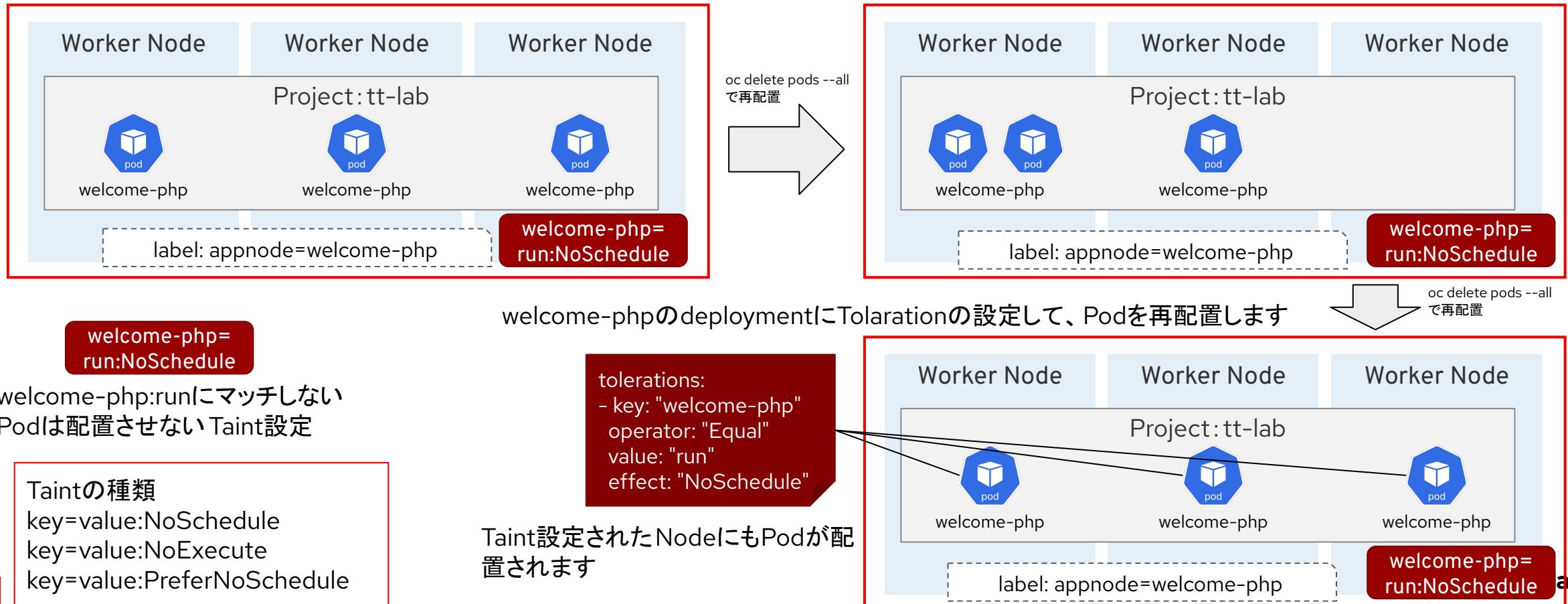
例) 特定のアプリに対して Pod数を5以下に制限する  
 clusterresourcequotaを作成します  
 labelを利用してします



# Taint と Toleration

Taint と Toleration は、協調して動作し、ワークロードが他のノードにスケジュールされないようにします

1つのNodeにTaint設定を行い、Podを再配置すると、Tolerationの設定がないため、TaintされたNodeではPodが起動しません



# Module 2

Red Hat Advanced Cluster Security for Kubernetes

15:30 - 16:30

16:30 - 17:15

# Red Hat Advanced Cluster Security for Kubernetes

クラウド・ワークロード・プロテクション・プラットフォームとクラウド・セキュリティ・ポスマネジメントにより"シフトレフト"を可能にします

## シフトレフト

### サプライチェーンのセキュリティ

スキャンとコンプライアンスを開発に拡張する(DevSecOps)

The screenshot shows the 'Vulnerability Management' section of the dashboard. It lists several CVE entries, each with a severity rating (e.g., Critical, High, Medium, Low), affected nodes, and a detailed description of the issue. For example, one entry for CVE-2020-2241 states: "The Linux kernel through 5.2.2, rglgt011\_mrgd\_wnd\_gwod in net/wireless/wireless.c does not reject a long GID-B, leading to a Buffer Overflow." Another entry for CVE-2019-17133 states: "In the Linux kernel through 5.2.2, rglgt011\_mrgd\_wnd\_gwod in net/wireless/wireless.c does not reject a long GID-B, leading to a Buffer Overflow." The interface includes filters for type (Header, Image CVE), CVSS score, and impact.

## Kubernetesセキュリティ 状態管理(KSPM)

### セキュアなインフラ

組み込みのKubernetes CSPMを活用して、リスクのある構成を特定し、修正する

The screenshot shows the 'Compliance' section of the dashboard. It displays a summary of system violations: 0 Critical, 5 High, 1 Medium, and 5 Low. Below this, there are two charts: 'VIOLATIONS BY CLUSTER' showing counts for critical, high, medium, and low violations across different clusters, and 'TOP BUSY DEPLOYMENTS' showing active violations over time for various deployments like 'calico-node', 'etcd', 'nginx', 'sensor', 'adminer-control', and 'scanner-db'. The interface includes filters for cluster and deployment.

## クラウドワークロード保護 (CWPP)

### セキュアなワークロード

ワークロード保護に対する  
"zero-trust execution"アプローチの維持と実施

The screenshot shows the 'Network Graph' section of the dashboard. It displays a complex network graph with nodes representing various cluster components like 'calico-node', 'etcd', 'nginx', 'sensor', 'adminer-control', and 'scanner-db', and external entities like 'NAT-1.GATEWAY'. Edges represent connections between these nodes. The interface includes filters for node type (Host, Container, All) and resource filters.

# Red Hat Advanced Cluster Security: ユースケース

アプリケーションのライフサイクル全体にわたるセキュリティ



## 脆弱性管理

イメージと実行中のコンテナにおける既知の脆弱性から身を守る



## ネットワークセグメンテーション

各アプリケーションのネットワーク分離とアクセス制御の適用と管理



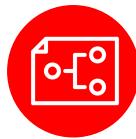
## セキュリティ構成管理

デプロイがセキュリティのベストプラクティスに従って構成されていることを確認する



## コンプライアンス

契約上および規制上の要件を満たし、それに対する監査を容易に行う



## リスクプロファイリング

OpenShift クラスターと Kubernetes クラスター全体のセキュリティ問題の優先順位を決めるためのコンテキストを得る



## 検知と対応

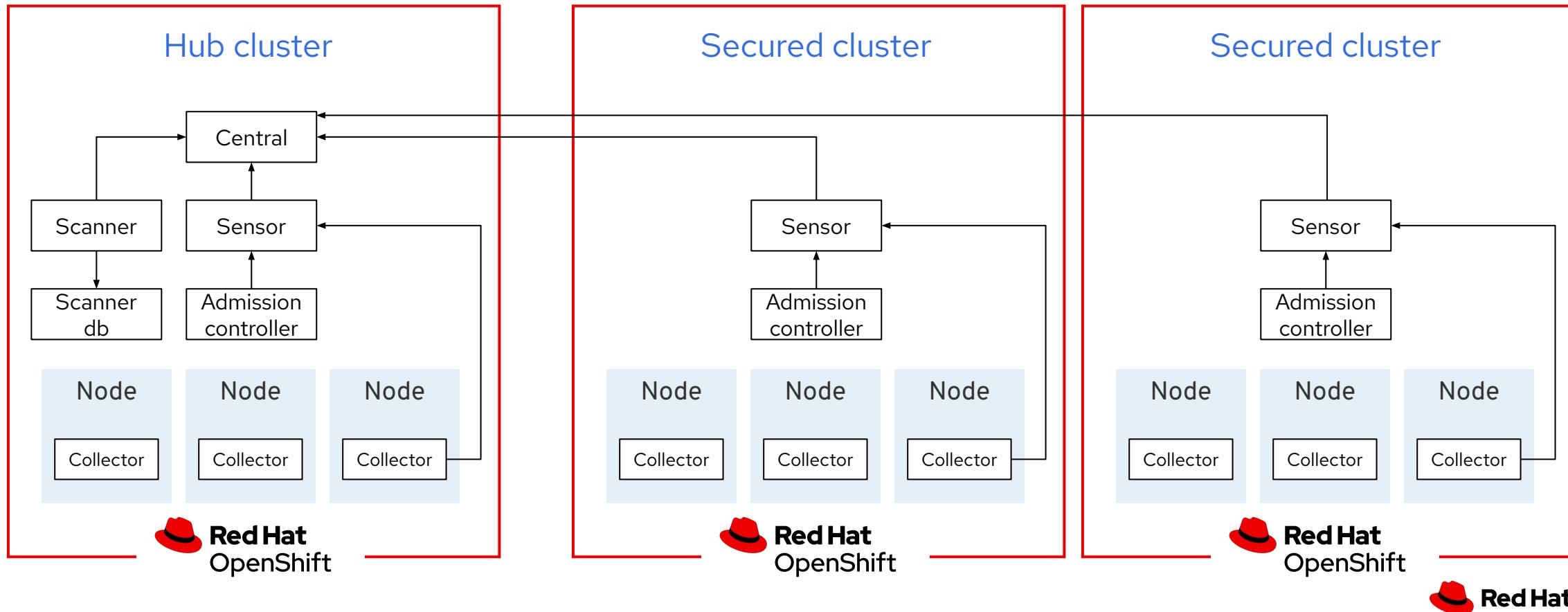
環境内のアクティブな脅威に対処するためのインシデントレスポンスの実施

# Red Hat Advanced Cluster Security for Kubernetes

Hub cluster : Centralが稼働しているクラスタ

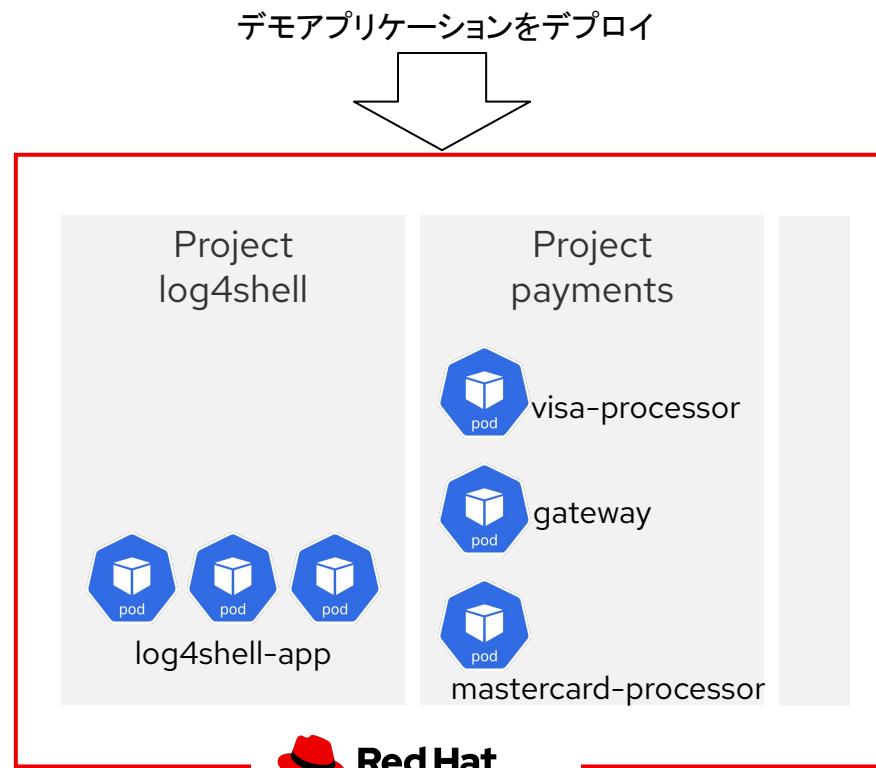
Secured cluster: 保護対象のクラスタ

※本ワークショップではhub clusterとSecured clusterが同一クラスタです



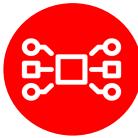
# ACSによる脆弱性のスキャン

Advanced Cluster Security (ACS) ダッシュボードで修正可能な脆弱性(log4shellなど)を正確に特定し、トリアージする方法を学びます



- ・デモアプリケーションをデプロイします
- ・ACS ダッシュボードへアクセスします
- ・ユーザーインターフェースの操作方法
- ・脆弱性管理ダッシュボードを使用して、クラスタ内的一般的な脆弱性と問題点を分析します
- ・得られた知識を使用して、クラスタ内のlog4shellエクスプロイトを検出します
- ・log4shellの脆弱性を含む脆弱なアプリケーションのデプロイをブロックするポリシーを作成します

## log4shellの脆弱性の発見と解析



### log4shellとは

Log4Shellは、アプリケーションのエラーメッセージを記録するための一般的なJavaライブラリであるApache Log4j 2のソフトウェア脆弱性です。CVE-2021-44228として公開されているこの脆弱性は、デバイスが特定のバージョンのLog4j 2を実行している場合、リモート攻撃者がインターネット上のデバイスを制御することを可能にします。



### なぜ、脆弱性の発見が重要なのか？

セキュリティチームや運用チームには、深刻度の異なる脆弱性に対応するためのプロセスや手順を用意しておくことが不可欠です。これらは、あらゆる脆弱性の正確な評価とともに、深刻度の高い脆弱性には迅速な対応に値する緊急性をもって対処し、深刻度の低い脆弱性にはより慎重な態度で対処することを可能にします。



### 検出と対応

コンテナの脆弱性を迅速に検出し、対応することができれば、チームはより早くサービスの開発に戻ることができます。

重大な脆弱性が発生した場合、脆弱性の範囲にアクセスし、トリアージする間、ほとんどのデプロイは保留されます。

## visa-prosessorの脆弱性の発見と解析

**脆弱性ダッシュボード**

Top Riskiest Images → View All

**Top Riskiest Images → visa-prosessorを検索**

**CVE情報**

image:visa-processorに関する情報

Red Hat logo

# log4shellの脆弱性の発見と解析

The screenshot shows the Red Hat Advanced Cluster Security for Kubernetes dashboard. In the top right corner, there is a search bar with the placeholder "Search CLI". Below it, a red box highlights the "Top Riskiest Images" section. A red arrow points from this section to the main content area. The main content area displays a table titled "Images Entity list" with columns: Image, Image CVEs, Top CVSS, Created, Scan Time, Image OS, Image Status, Entities, and Risk Priority. One row in the table is highlighted with a red box. Another red arrow points from this row to a detailed view of the image entry. The detailed view shows the image entry "quay.io/mfoster/log4shell-demo:0.10" with its details, including a pie chart of CVSS scores and a list of related entities.

Top Riskiest Images → View All

Top Riskiest Images → CVE-2021-44228を検索

image:log4shell-demoに関する情報

## デプロイメントvisa-prosessorとlog4shellのリスク指標

The screenshot shows the Red Hat Advanced Cluster Security for Kubernetes dashboard. The left sidebar includes links for Dashboard, Network Graph, Violations, Compliance, Vulnerability Management (2.0), Vulnerability Management (1.0), Configuration Management, Risk, and Platform Configuration. The main area displays a table of 91 deployments under a 'Risk' view. The table columns are Name, Created, Cluster, Namespace, and Priority. Two specific rows are highlighted with red boxes: 'visa-processor' (Priority 2) and 'log4shell-app' (Priority 6). A vertical red bar on the right side contains the word 'Feedback'.

Name	Created	Cluster	Namespace	Priority
visa-processor	11/12/2023   11:11:36AM	production	payments	2
log4shell-app	11/12/2023   11:11:20AM	production	log4shell	6
backend-atlas	11/12/2023   11:10:59AM	production	backend	3
asset-cache	11/12/2023   11:11:09AM	production	frontend	4
yelb-ui	11/12/2023   11:11:44AM	production	yelb	5
spring4shell-app	11/12/2023   11:11:41AM	production	spring4shell	6
reporting	11/12/2023   11:11:23AM	production	medical	9
yelb-appserver	11/12/2023   11:11:53AM	production	yelb	10
monitor	11/12/2023   11:11:17AM	production	frontend	11
yelb-db	11/12/2023   11:11:50AM	production	yelb	12
redis-server	11/12/2023   11:11:47AM	production	yelb	12
search-postgres	11/12/2023   9:38:40AM	production	open-cluster-management	14
tekton-operator-webhook	11/12/2023   9:25:09AM	production	openshift-operators	15
wordpress	11/12/2023   11:11:14AM	production	frontend	17

## ACSポリシー

Policies > Log4Shell: log4j Remote Code Execution vulnerability

Log4Shell: log4j Remote Code Execution vulnerability Enabled ポリシーが有効になっている

Actions ▾

Policy behavior

### Policy details

	どの段階で評価するか？
Severity	Critical
Categories	Vulnerability Management
Type	System default
Description	Alert on deployments with images containing the Log4Shell vulnerabilities (CVE-2021-44228 and CVE-2021-45046). There are flaws in the Java logging library Apache Log4j in versions from 2.0-beta9 to 2.15.0, excluding 2.12.2.
Rationale	These vulnerabilities allows a remote attacker to execute code on the server if the system logs an attacker-controlled string value with the attacker's JNDI LDAP server lookup.
Guidance	Update the log4j library to version 2.16.0 (for Java 8 or later), 2.12.2 (for Java 7) or later. If not possible to upgrade, then remove the JndiLookup class from the classpath: zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
MITRE ATT&CK	
Policy has no MITRE ATT&CK vectors	

検出した場合の振る舞い

Lifecycle stages	Build, Deploy
Event source	N/A
Response	Inform

### Policy criteria 判断基準

1	CVEがCVE-2021-44228または CVE-2021-45046に該当した場合
CVE	CVE identifier is:
CVE-2021-44228	— OR —
CVE-2021-45046	

## ACSポリシーの複製と編集

Policies > Log4Shell: log4j Remote Code Execution vulnerability (COPY)

### Log4Shell: log4j Remote Code Execution vulnerability (COPY) Enabled

Actions ▾

#### Policy details

Severity	<span>Critical</span>
Categories	Vulnerability Management
Type	User generated
Description	Alert on deployments with images containing the Log4Shell vulnerabilities (CVE-2021-44228 and CVE-2021-45046). There are flaws in the Java logging library Apache Log4j in versions from 2.0-beta9 to 2.15.0, excluding 2.12.2.
Rationale	These vulnerabilities allows a remote attacker to execute code on the server if the system logs an attacker-controlled string value with the attacker's JNDI LDAP server lookup.
Guidance	Update the log4j library to version 2.16.0 (for Java 8 or later), 2.12.2 (for Java 7) or later. If not possible to upgrade, then remove the JndiLookup class from the classpath: zip -q -d log4j-core-*jar org/apache/logging/log4j/core/lookup/JndiLookup.class
<b>MITRE ATT&amp;CK</b>	
Policy has no MITRE ATT&CK vectors	

#### Policy behavior

Lifecycle stages	Build, Deploy	BuildとDeployの段階
Event source	N/A	
Response	Enforce	強制的にブロック
Enforcement	Deploy	Deploy時

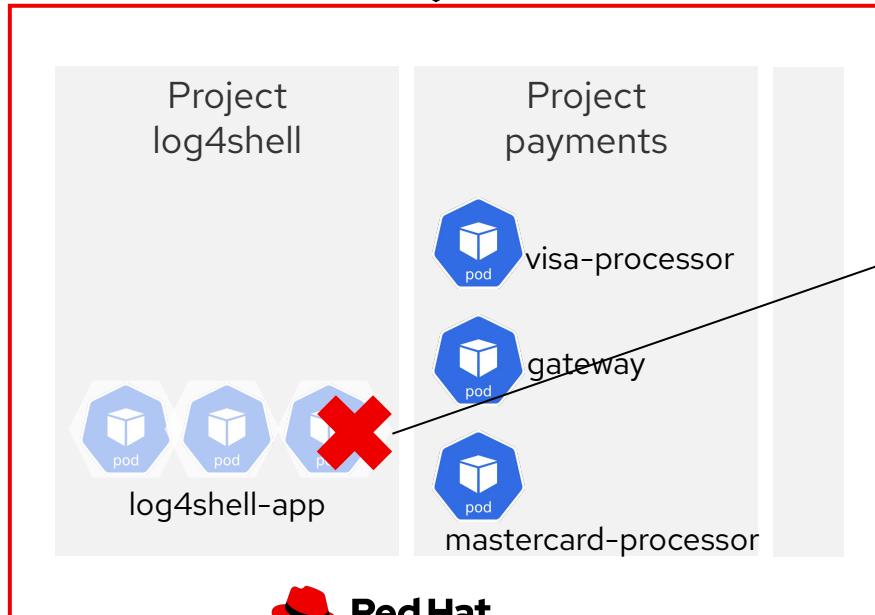
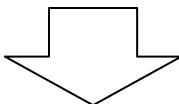
#### Policy criteria

1	CVEがCVE-2021-44228または CVE-2021-45046に該当した場合
CVE	CVE identifier is:
CVE-2021-44228	— or —
CVE-2021-45046	

## ACSポリシーの動作確認

Log4Shell: log4j Remote Code Execution vulnerability (COPY)が有効になっているため、該当する脆弱性を含むデプロイメントはブロックされて Podは起動しません

log4shell-appを新規にデプロイ



**ACSコンソールの[Violations]**

**Log4Shell: log4j Remote Code Execution vulnerability (COPY)  
in "log4shell-app" deployment**

Violation Enforcement Deployment Policy

Enforcement on this deployment is enabled for this policy

Deployment data was evaluated against this security policy. Deployment scaled to 0 replicas in response to this policy violation.

If the enforcement action is being a

**OpenShift Webコンソールの[Topology]**

local-cluster

Project: log4shell Application: All applications

- Developer
- +Add
- Topology**
- Observe
- Search
- Builds
- Pipelines
- Helm
- Project
- ConfigMaps
- Secrets

log4shell-app

Health checks  
Container log4shell-app does not have health checks to ensure your application is running correctly. Add health checks

Details Resources Observe

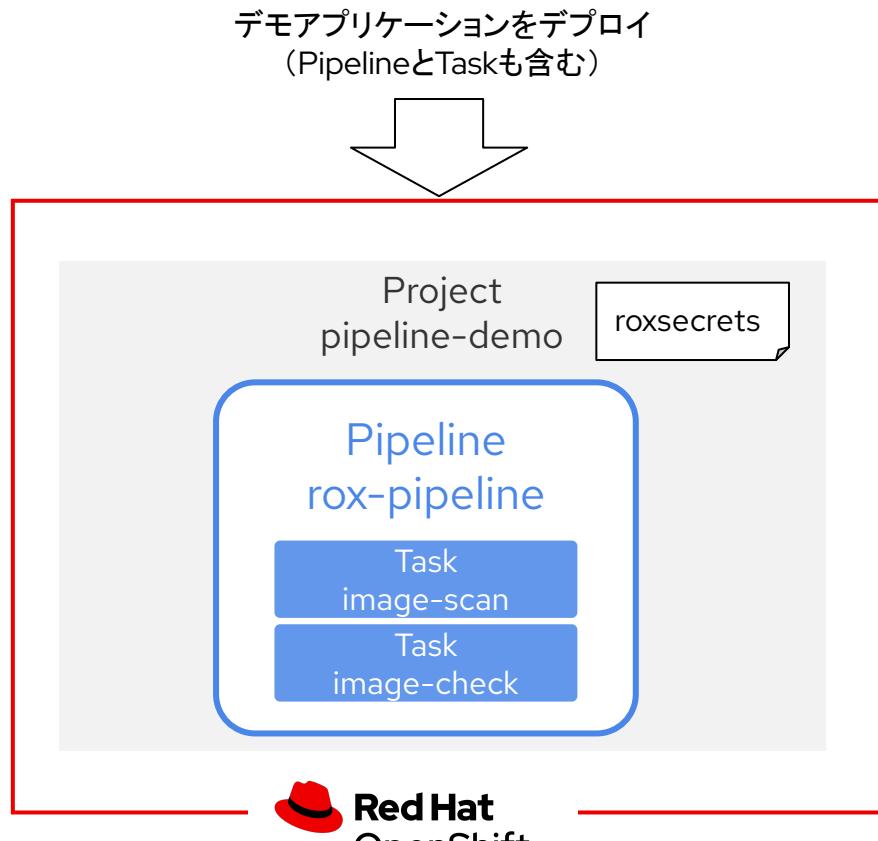
Pods  
No Pods found for this resource.

Services  
No Services found for this resource.

Routes  
No Routes found for this resource.

# DevSecOps

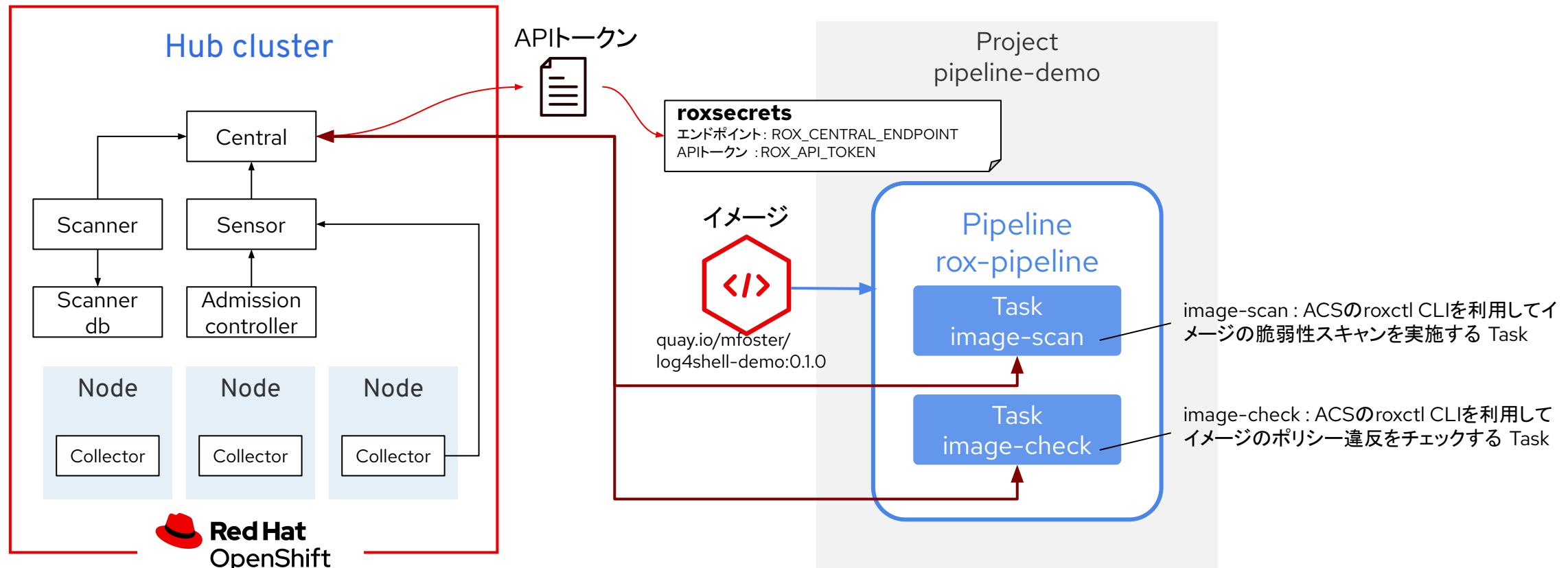
OpenShift PipelinesのワークフローにACS脆弱性ポリシーを実装します



1. デモアプリケーションをデプロイします  
(PipelineとTask、Secretが作成されます)
2. OpenShiftのパイプラインがACSセントラルと通信するために  
必要な認証を設定します
3. パイプラインを実行し、脆弱性検知によりパイプラインが  
中断されることを確認します

# DevSecOps

ACSのroxctl CLIを利用するTaskをPipelineに組み込むことにより、イメージスキャンやポリシーチェックを自動的に行うことができ、脆弱性のあるアプリケーションやイメージのビルトやデプロイを中断させることができます



# DevSecOps

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: pipeline-demo ▾

Administrator ▾

Home ▶

Operators ▶

Workloads ▶

Networking ▶

Storage ▶

Builds ▶

Pipelines ▾

- Pipelines
- Tasks
- Triggers

Observe ▶

Compute ▶

User Management ▶

Administration ▶

PLR rox-pipeline-odah4n Failed

Actions ▾

Details YAML TaskRuns Parameters Logs Events

### PipelineRun details

✓ image-scan 1/1

✖ image-check 0/1

→ Pipelineが中断される

Name: rox-pipeline-odah4n

Status: Failed

Namespace: pipeline-demo

Message: Failure on task - check logs for details.

Labels:  Edit

Annotations: 2 annotations

Created at:

Log snippet:

```
-----+-----+
WARN: A total of 6 policies have been violated
ERROR: failed policies found: 1 policies violated that are failing the check
ERROR: Policy "Fixable Severity at least Important" - Possible remediation: "Use your
package manager to update to a fixed version in future builds or speak with your
security team to mitigate the vulnerabilities."
ERROR: checking image failed after 3 retries: failed policies found: 1 policies
violated that are failing the check
```

# Module 3

**Red Hat Advanced Cluster Management for Kubernetes**

**15:30 - 16:30**

**16:30 - 17:15**



# Red Hat

## Advanced Cluster Management for Kubernetes

### 操作とメンテナンスの簡素化

単一のコンソールで、表示、管理、操作、問題解決のすべてを行うことができる

### OpenShift上で動作

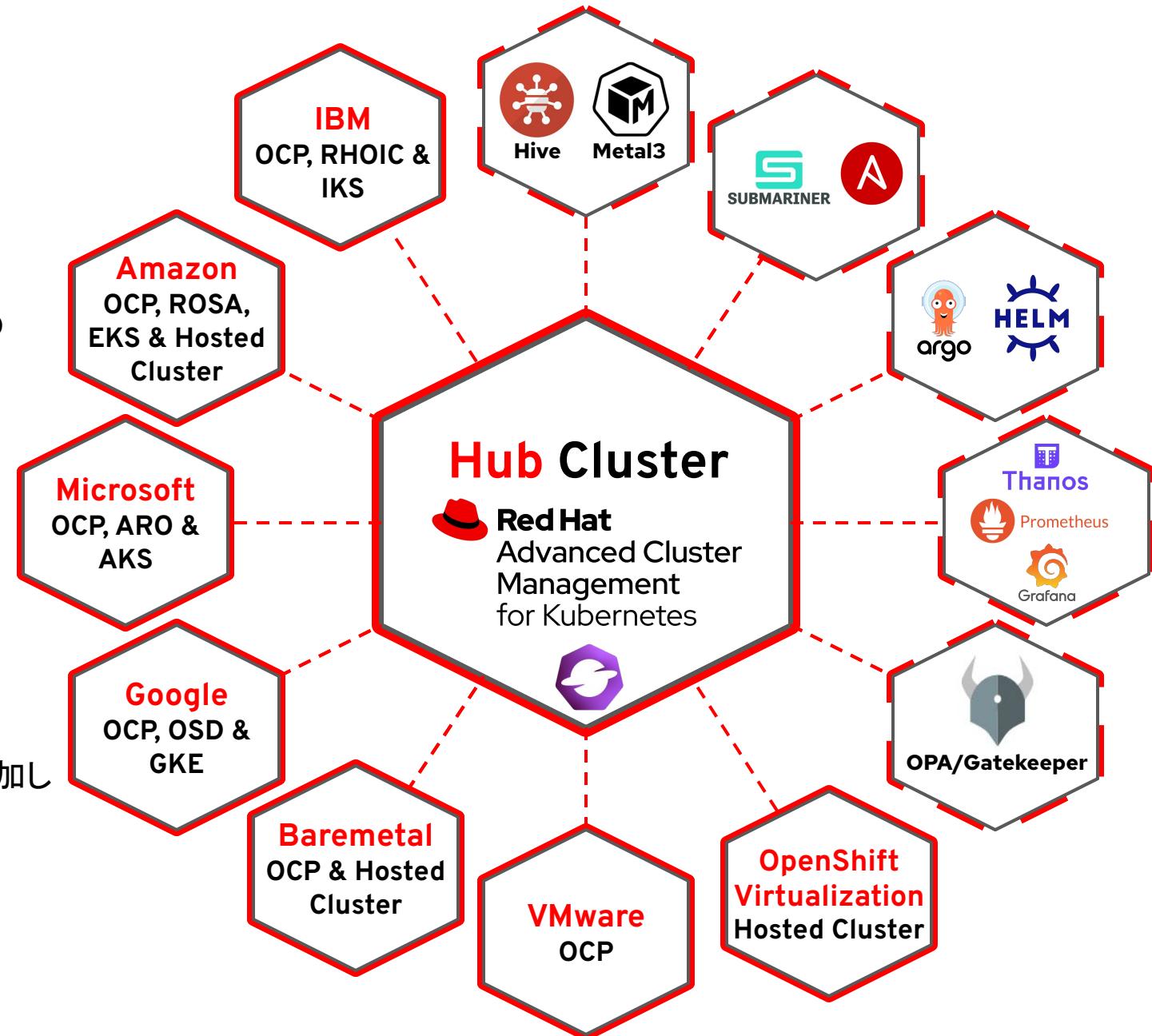
他のKubernetesアプリと同様に、OpenShiftクラスタ上で簡単に実行・管理できる

### ハブ・スポークアーキテクチャ

すべての設定をHubクラスタコンポーネントで管理し、Spoke Kubernetesクラスタをハブにシームレスに追加します。

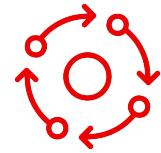
### タイトな統合

RHACMには豊富なAPI、アドオンが付属しており、他の主要な企業ツールと統合することができる。



# Red Hat Advanced Cluster Management for Kubernetes

## End-to-end automation with Red Hat Ansible Automation Platform integration



マルチクラスター・ライフサイクル管理



ポリシードリブンガバナンス、リスト、コンプライアンス



高度なアプリケーション・ライフサイクル管理



健全性と最適化のためのマルチクラスター・オブザーバビリティとサーチ



相互接続のためのマルチクラスター・ネットワーキング

The screenshot displays the Red Hat Advanced Cluster Management for Kubernetes interface. The top navigation bar includes the Red Hat logo, a search bar, and user information. The main area has two tabs: 'Overview' and 'Governance'.  
**Overview:** This tab provides a high-level summary of the environment. It shows the number of clusters (22 Applications, 13 Clusters, 6 Kubernetes type, 2 Regions, 49 Notes), and four circular dashboards: Cluster violations (8%), Pods (99% Running), Cluster status (92% Ready), and Cluster issues (1).  
**Governance:** This tab shows policy violations across various standards and categories. It includes sections for Policy set violations (2) and Policy violations (35). A table lists clusters with violations, such as 'rosacluster' (1), 'local-cluster' (22), and 'migration' (10).  
On the right side of the interface, there is a detailed 'Deployment' timeline diagram showing the flow from Application pacman-appl to Ansiblejob to Subscription pacman-appl to Placement pacman-appl. Below this is a 'Metrics' section with a table of memory usage and a graph of Top 5 Utilized Clusters (% CPU Usage).

# Red Hat Advanced Cluster Management for Kubernetes

マルチクラウド環境におけるKubernetesクラスタのフリートを効率的に管理する方法を学びます。RHACMダッシュボードから以下の方法を学びます

## 1. クラスター・ライフサイクルの操作

- a. クラウドインフラへのアクセスを提供するクラウドプロバイダーの認証情報を設定します
- b. AWS上にOpenShiftクラスタを作成します
- c. AWS上にSNO(Single-node OpenShift) クラスタを作成します

## 2. アプリケーションの作成と管理

- a. マネージドクラスタにアプリケーションをデプロイします
- b. application placementを理解します

## 3. ガバナンス(ACMでポリシーを作成・適用)

- a. ETCD暗号化を有効にするK8sポリシーを作成します
- b. ポリシーがinform modeまたはenforce modeに設定できることを理解します

# クラスターライフサイクルの操作

- local-clusterのコンソールをAll Clustersに切り替えます
- 本環境はAWS上に構築しているため、AWSの認証情報を設定します
- 認証情報を設定後、ウィザードに従ってクラスタを作成します

Here is some important information about your environment:

OpenShift Console: <https://console-openshift-console.apps.cluster-9fddq.9fddq.sandbox1562.opentlc.com> OpenShift API for command line 'oc' client: <https://api.cluster-9fddq.9fddq.sandbox1562.opentlc.com:6443> Download oc client from <http://mirror.openshift.com/pub/openshift-v4/clients/ocp/stable-4.12/openshift-client-linux.tar.gz>

Access the workshop at <https://dashboard-lab-ocp-cns.apps.cluster-9fddq.9fddq.sandbox1562.opentlc.com>

Login with 'kubeadmin' and 'druMK-SAcLK-8At2l-6qb7k'

Workshop may not be accessible until rollout finishes shortly.

Your RHACS console is available at:

<https://central-stackrox.apps.cluster-9fddq.9fddq.sandbox1562.opentlc.com>

RHACS portal username: admin

RHACS portal password: MjcwNzc2

Use the following credentials to deploy in the AWS sandbox account where your Hub is running.

AWS\_ACCESS\_KEY\_ID: AKIAUVI7DCRELC64VV2E +  
AWS\_SECRET\_ACCESS\_KEY: 0RXUvE6WV14IYnmq5N8oytLsNCaVBqUlBYjJVcLm +  
Top level domain: <.sandbox1562.opentlc.com> +

AWSの認証情報

You can access your bastion via SSH: ssh [demo-user@bastion.9fddq.sandbox1562.opentlc.com](ssh://demo-user@bastion.9fddq.sandbox1562.opentlc.com)

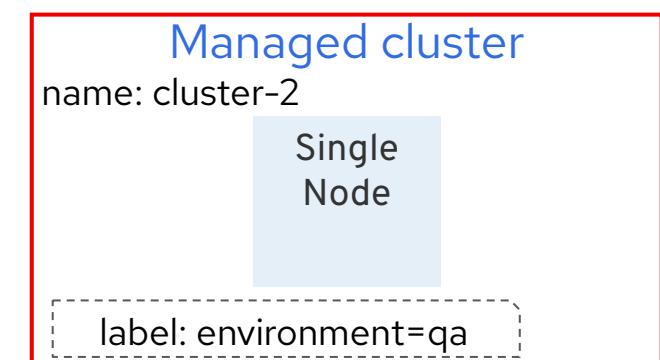
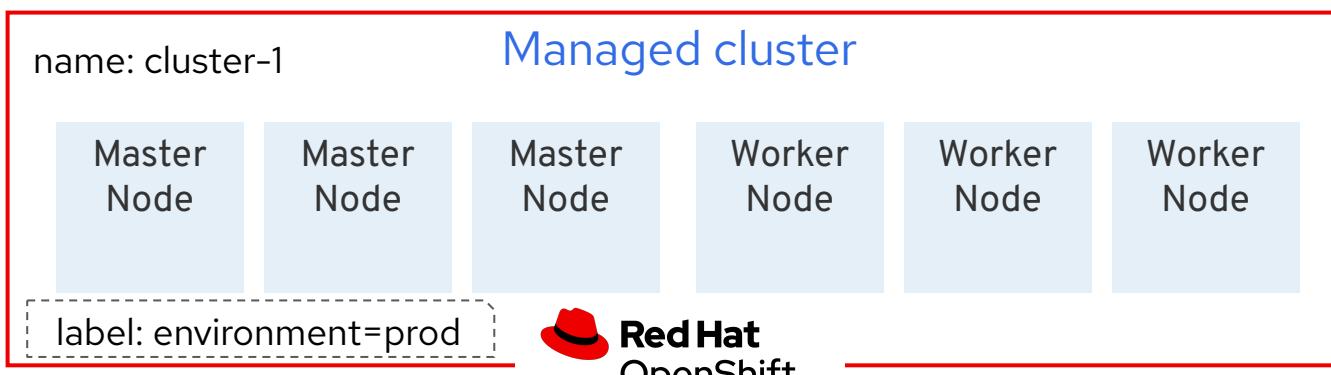
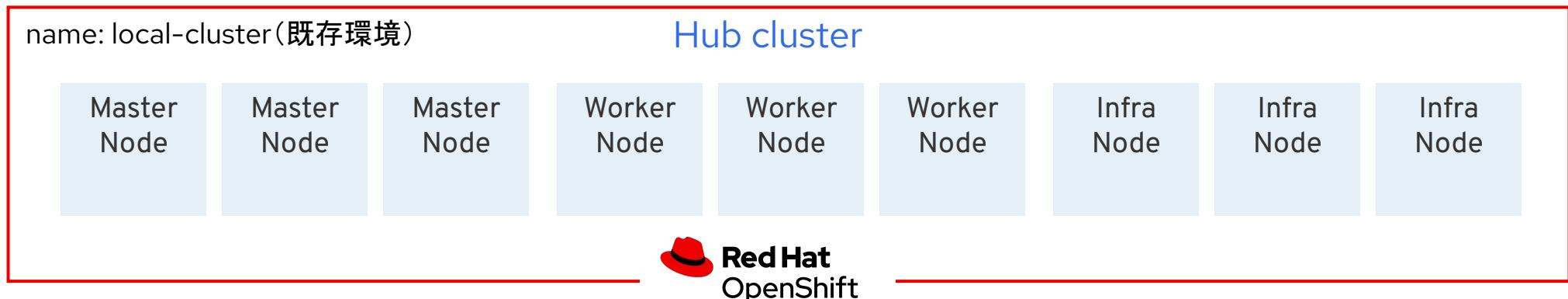
Make sure you use the username 'demo-user' and the password 'QuXuR4HNTqGn' when prompted.

You may manage your RHDP service at:

<https://demo.redhat.com/services/user-mkoshimi-redhat-com/sandboxes-gpte.ocp4-acm-acs-ops-wksp.prod>

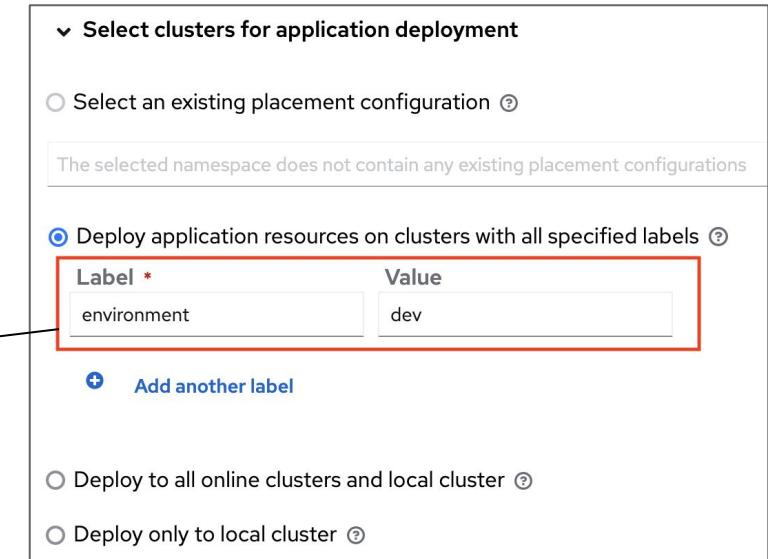
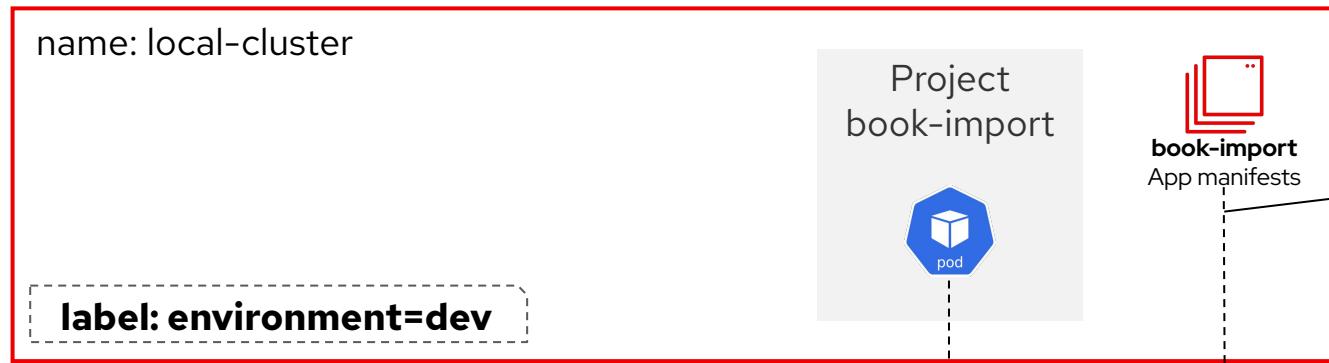
# クラスターライフサイクルの操作

- ・クラスタの構築が完了すると以下の構成になります



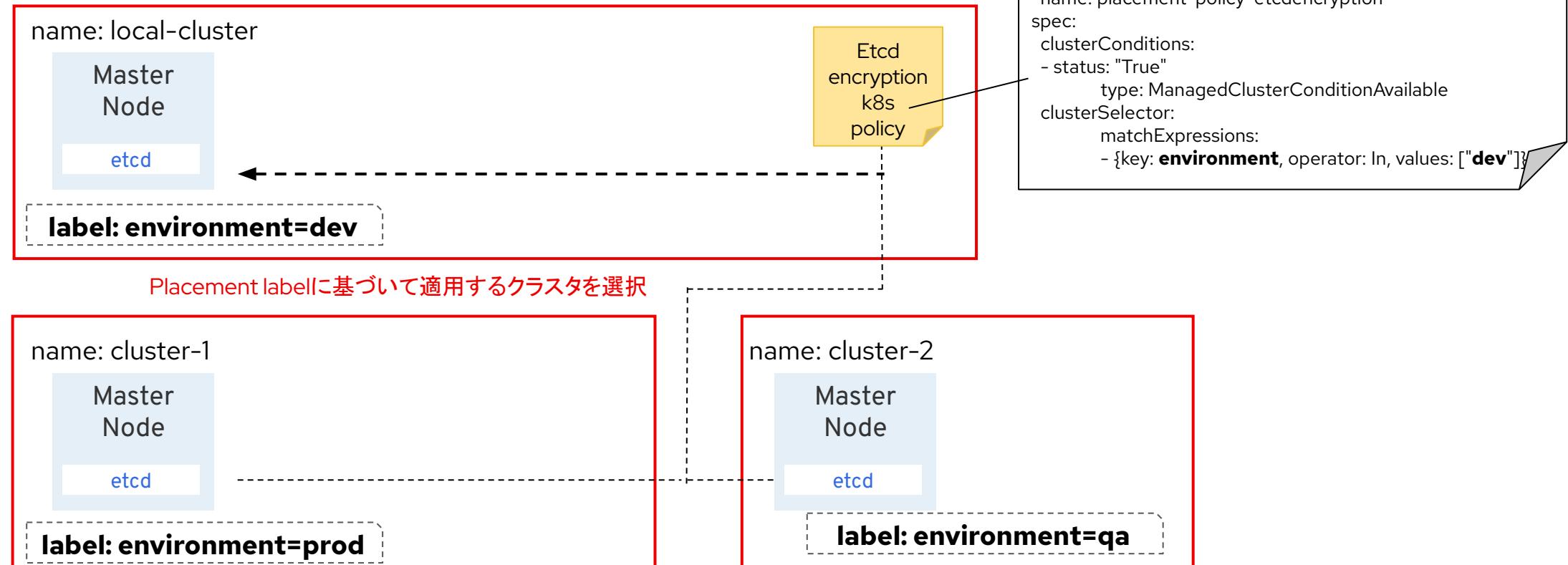
# アプリケーションの作成と管理

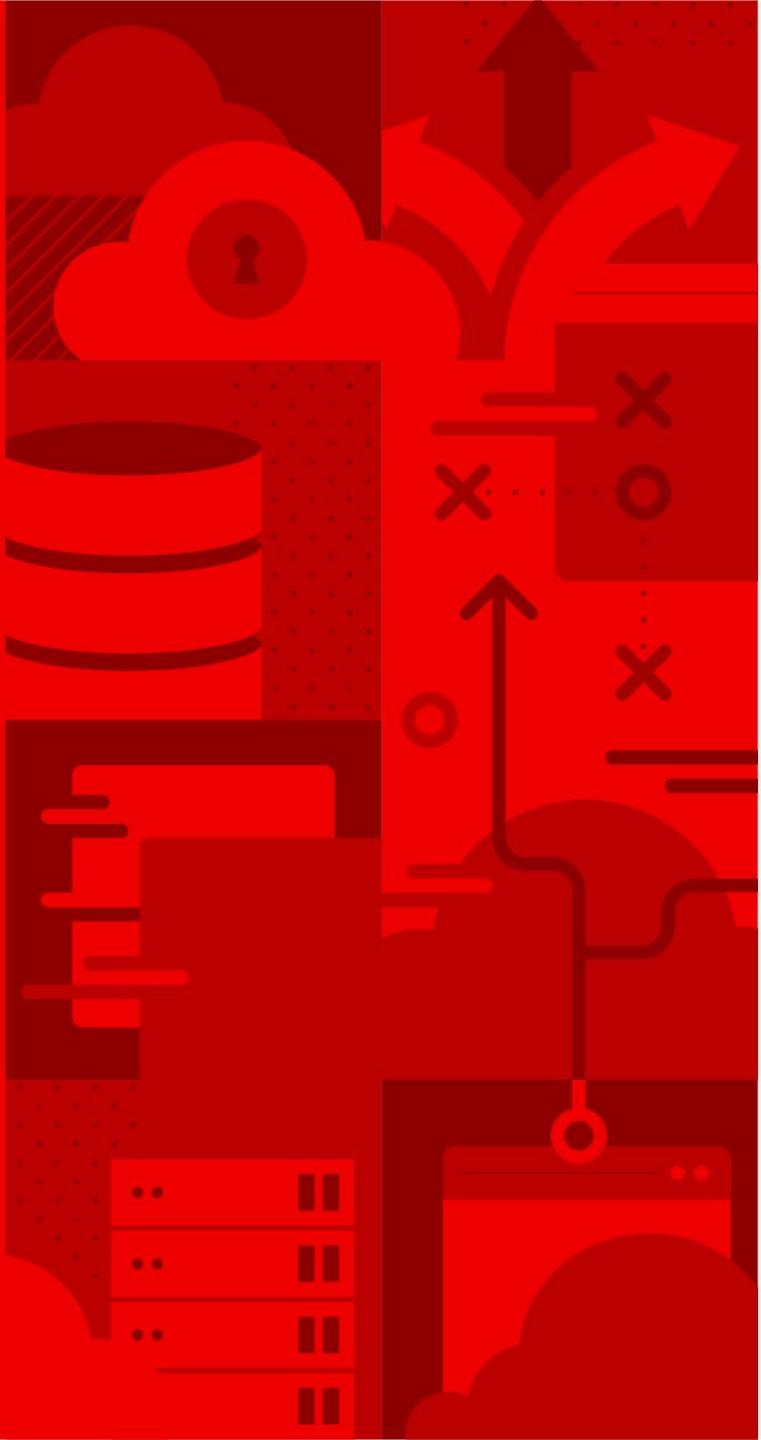
アプリケーションはlabelに基づいてデプロイされます



# ACMでポリシーを作成・適用

ポリシーの中でlabelに基づいてクラスタを選択し、適用します





# Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[twitter.com/RedHat](https://twitter.com/RedHat)