



OpenShift ACM





Open Cluster Manager

TRAINING

CHOI GOOK HYUN

Freelancer Instructor

DAY 1

- ▶ About the OpenShift Advanced Cluster

THE FEDERATION

THE FEDERATION

with kubernetes

"The Federation" API는 올해 2022년도 10월에 더 이상 개발을 하지 않기로 결정하였음. 다만, API사양은 계속 개발하며, 프로그램 부분은 OCM, KARMADA와 같은 커뮤니티에 개발을 맡기기로 결정.

공식적으로 Federation API는 v1, v2기반으로 되어 있으며, 쿠버네티스는 API정의 애플리케이션 개발은 외부 그룹에서 맡기로 하였음.

현재는 다음과 같이 두 그룹이 주요 리더로 끌고 가고 있음.

- ▶ Open Cluster Manager
- ▶ Karmada

THE FEDERATION

with kubernetes

Follow-up: discussion on archiving Kubefed 조회수 177회



pmo...@gmail.com

받는사람 kubernetes-sig-multiclustar

Everyone-

In recent Kubecon SIG updates we have previously communicated that SIG leadership has been considering archiving the Kubefed[1] repository. This topic was given discussion offline. This communication is a follow-up on that action item.

TLDR: Our current favored option is to archive the kubefed project but we are soliciting community input before making a final decision. Continue reading for rationale.

In our discussions, Jeremy and I considered the following questions:

KARMADA

KARMADA

with kubernetes

KARMADA는 쿠버네티스 federation v1, v2프로토콜 기반으로 작성이 되어 있음. 쿠버네티스 기반으로 동작하며, 동작 방식은 OCM과 매우 흡사하나, 중간에 "Karmada API Server"중심으로 "etcd", "scheduler"와 통신하며 중간에 "Karmada controller"를 통해서 정책 배포 및 관리를 한다.

현재 KARMADA는 "CLOUD NATIVE"의 샌드박스(Sandbox) 프로젝트로 선언이 되어 있다. CNCF에서는 좀 더 KARMADA를 지원하고 있는 분위기.

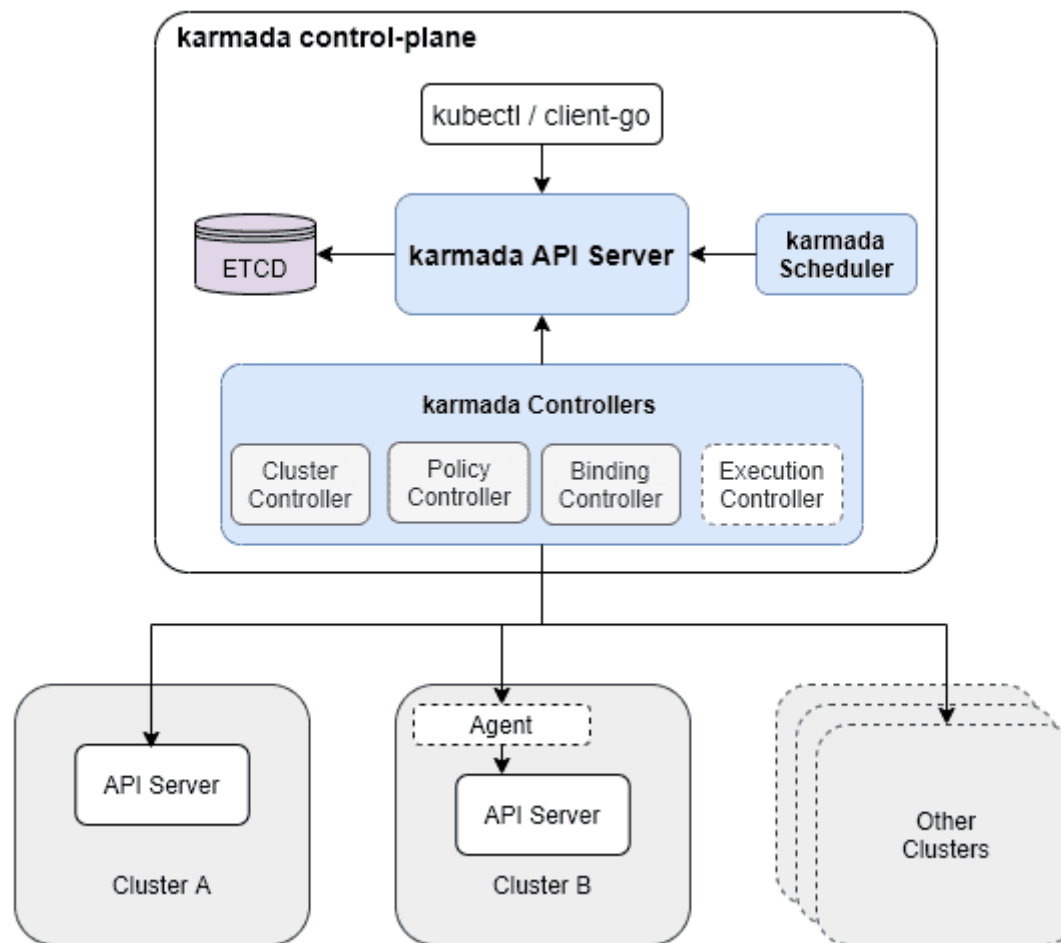
KARMADA

with kubernetes

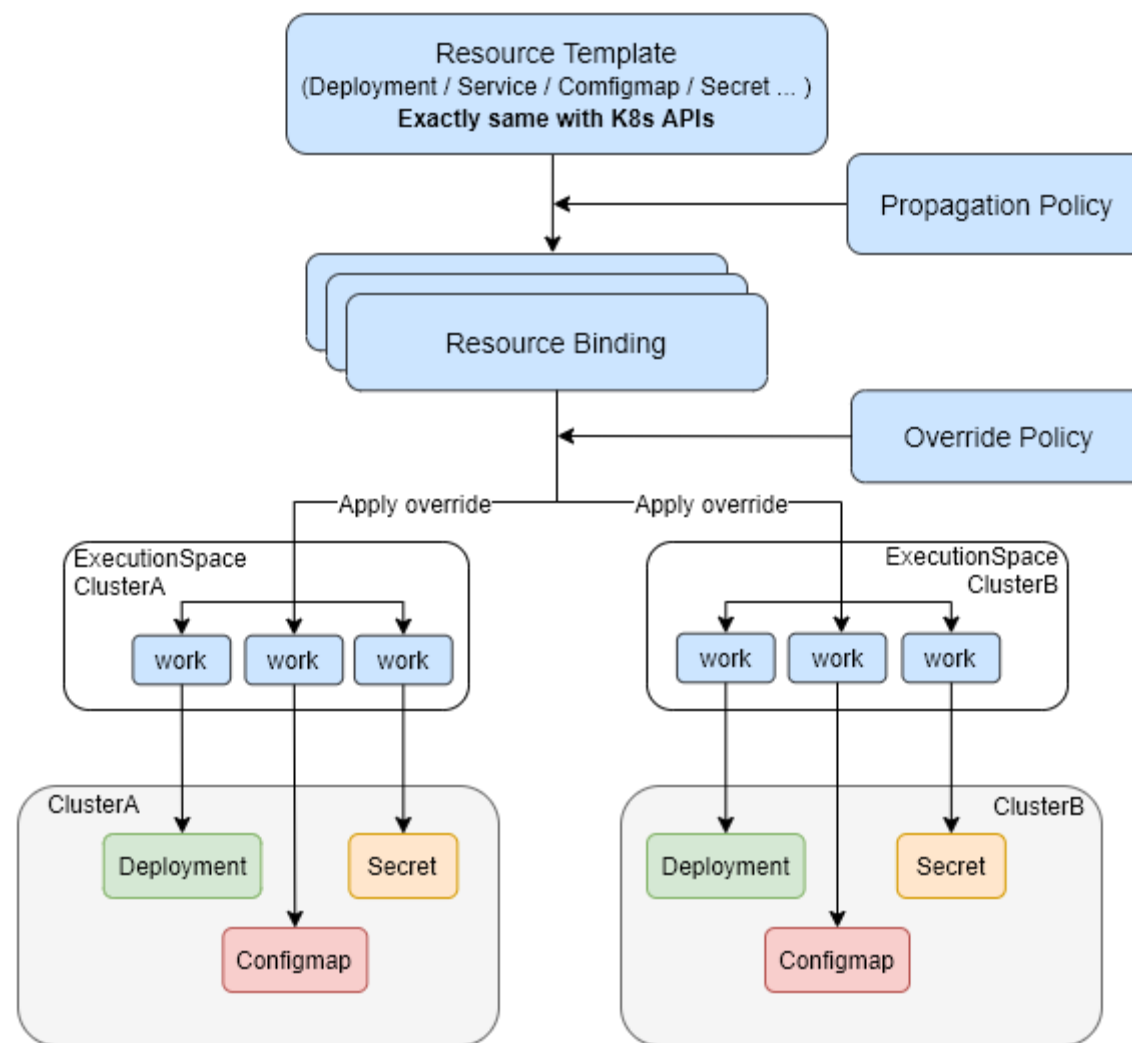
주요 목적은 다음과 같다.

- ▶ KIS native API compatible
- ▶ Out of the Box
- ▶ Avoid Vendor Lock-in
- ▶ Centralized Management
- ▶ Fruitful Multi-Cluster Scheduling Policies
- ▶ Open and Neutral

KARMADA



Karmada Concepts



OCM

OCM

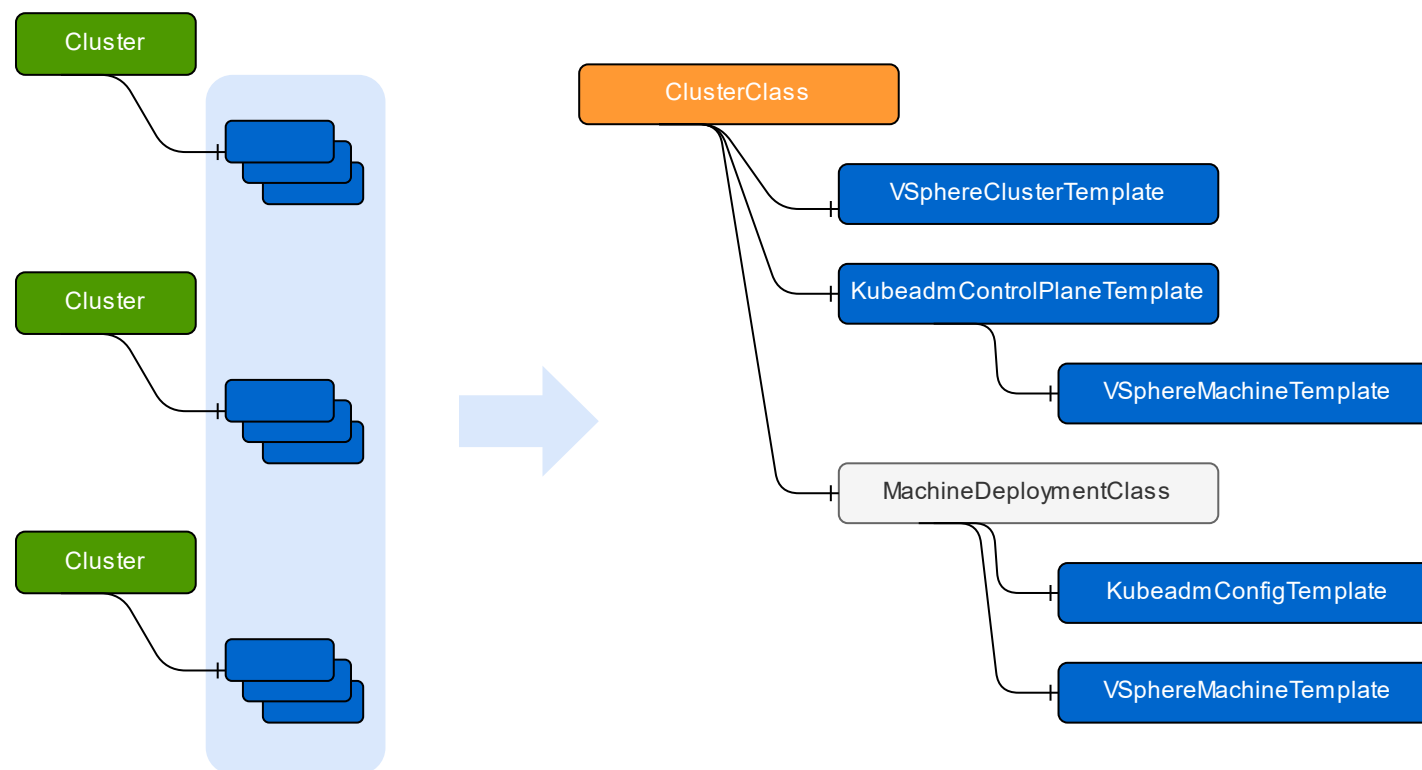
with OpenShift

OpenShift는 Kubernetes기반으로 작성이 되어 있음. 이는 두 가지 API기반으로 구성이 되어 있음.

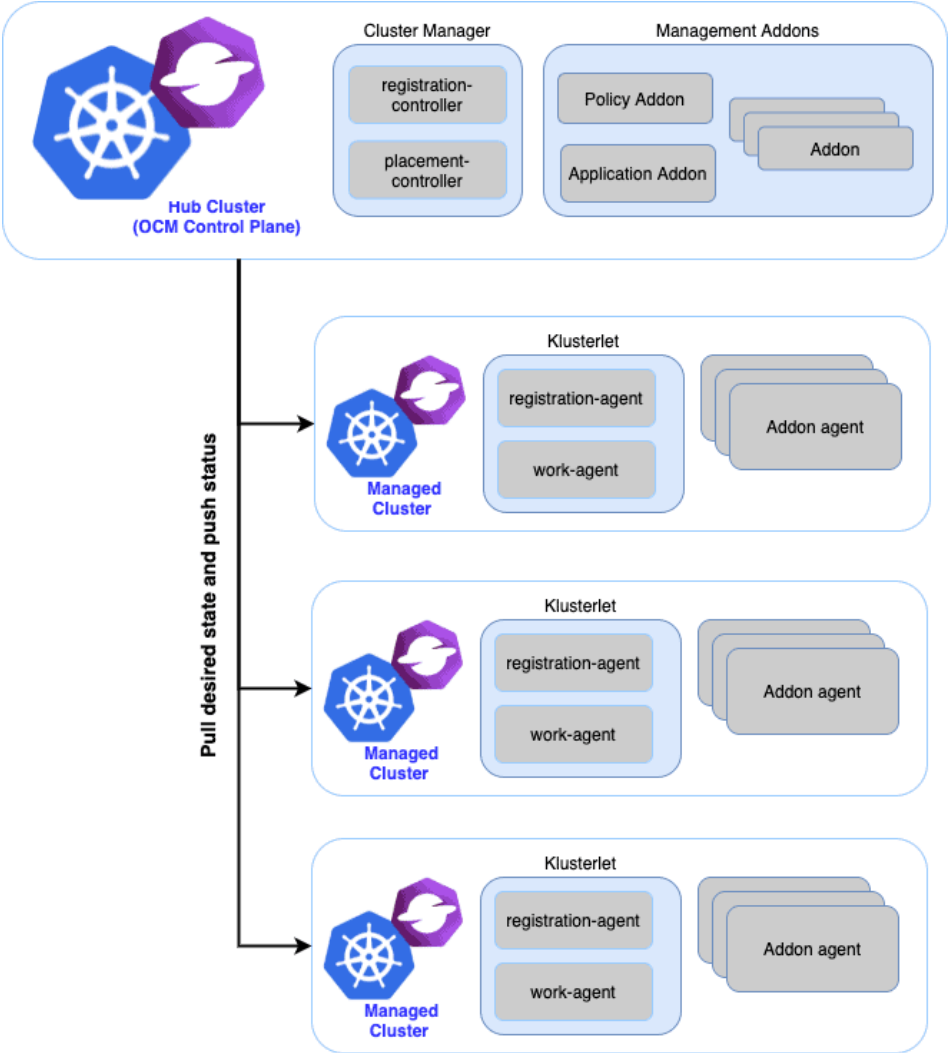
- ▶ Cluster API
- ▶ Federation API

Cluster API는 Kubernetes SIG에서 맨 처음 릴리즈 한 API. Federation API는 다중 클러스터 운영을 위해서 SIG에서 개발을 시작. OCM에는 현재 레드햇이 개발을 참여하고 있으며, 이 프로젝트도 "CLOUD NATIVE COMPUTING FOUNDATION"의 일부분.

Federation



OCM



OCM

ClusterAPI

클러스터 API는 다음과 같은 기능을 기본적으로 다루고 있다.

- ▶ 자원을 라이프 사이클(LifeCycle) 관리
- ▶ 설정 및 정책(Deployment, Policy) 적용 및 배포
- ▶ Pod 및 컨테이너 복제
- ▶ 네임스페이스 관리

OCM

Federation V1

Fede v1는 SIG에서 멀티 클러스터를 관리하기 위해서 프로젝트를 시작하였다. 쿠버네티스 SIG에서 제일 큰 프로젝트 중 하나이었으며, 앞서 말한 것처럼 멀티 클러스터 관리를 위해서 v1를 설계를 하였다. 주요 목적은 다음과 같다.

- Difficulties in re-implementing the Kubernetes API at the cluster level, as federation-specific extensions were stored in annotations.
- Limited flexibility in federated types, placement and reconciliation, due to 1:1 emulation of the Kubernetes API.
- No settled path to GA, and general confusion on API maturity; for example, Deployments are GA in Kubernetes but not even Beta in Federation v1.

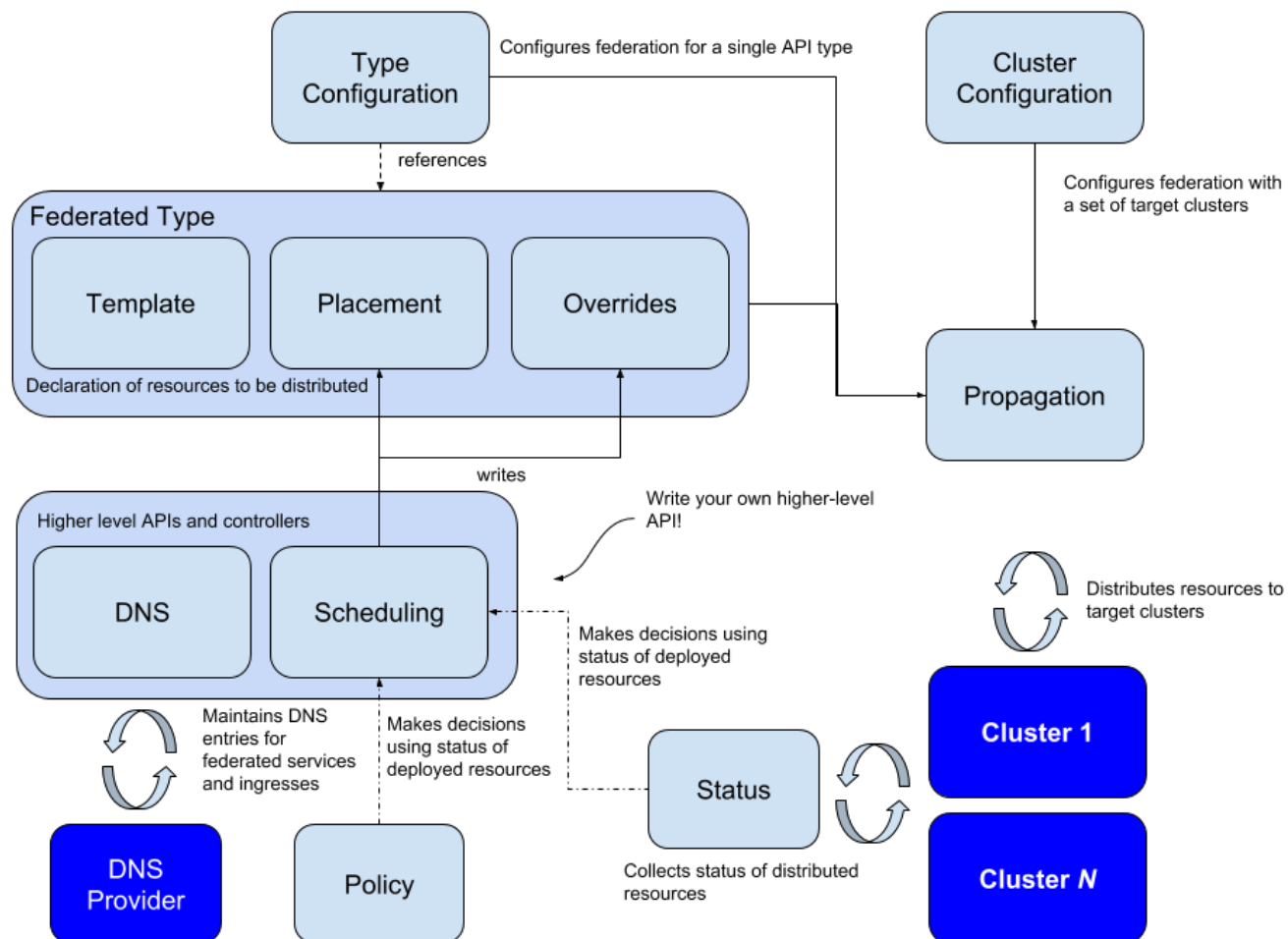
OCM

Federation V1

Federation은 v1기반으로 v2로 진화가 되었으며, 기본적인 컨셉은 v1에서 가져와서 v2에서 기능 추가를 시작하였다.

Federation은 "Hub and Spoke(pull and push)"구조를 가지고 있고 다음과 같이 동작한다.

Federation



OCM

Federation V1

연합의 주요 목표는 쿠버네티스 자원을 연합 그리고, "API to API"를 그룹으로 정의 하는게 주요 목표이자 목적이다.

쿠버네티스의 기능을 확장하기 위해서 CRD(CustomResourceDefinitions)를 통해서 구현이 됩니다.

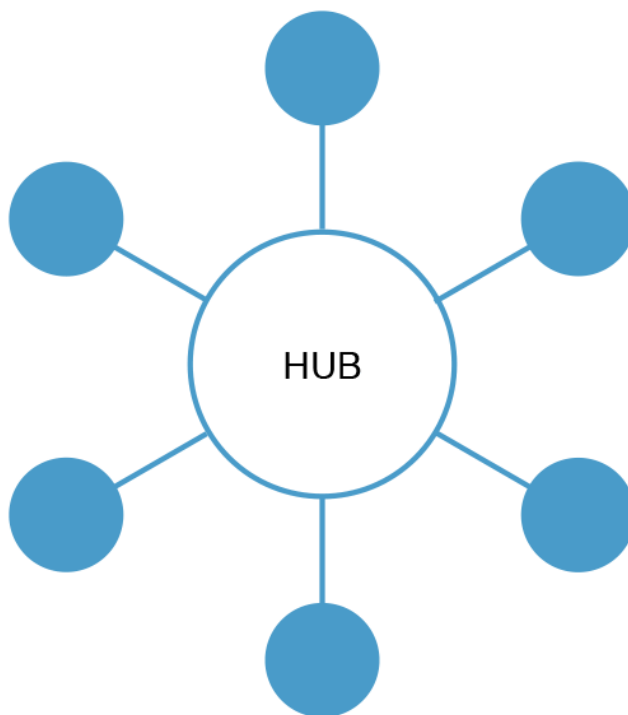
이는 문제 없이 동작하는 API를 다른 쿠버네티스 클러스터에 API를 배포하는 메커니즘이다.

Kubernetes

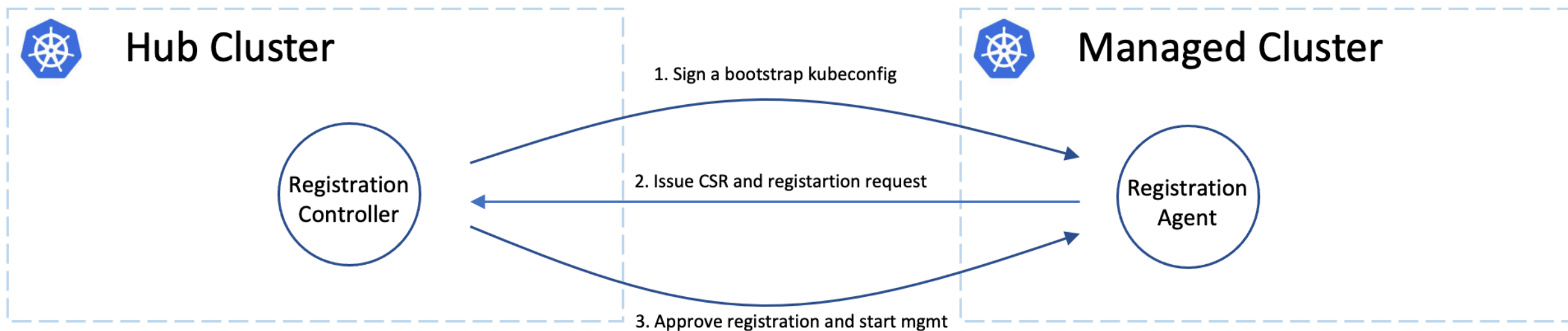
Federation

During the journey of defining building blocks of the federation APIs, one of the near term goals also evolved as 'to be able to create a simple federation a.k.a. simple propagation of any Kubernetes resource or a CRD, writing almost zero code'. What ensued further was a core API group defining the building blocks as a Template resource, a Placement resource and an Override resource per given Kubernetes resource, a TypeConfig to specify sync or no sync for the given resource and associated controller(s) to carry out the sync.

HUB and SPOKE



HUB and SPOKE



Kubernetes

Federation

Federation에서 주요 자원 지시자는 다음과 같다.

- ▶ Template
 - A Template type holds the base specification of the resource – for example, a type called FederatedReplicaSet holds the base specification of a ReplicaSet that should be distributed to the targeted clusters

Kubernetes

Federation

- ▶ Placement
 - A Placement type holds the specification of the clusters the resource should be distributed to – for example, a type called FederatedReplicaSetPlacement holds information about which clusters FederatedReplicaSets should be distributed to

Kubernetes

Federation

- ▶ Override
 - An optional Overrides type holds the specification of how the Template resource should be varied in some clusters - for example, a type called FederatedReplicaSetOverrides holds information about how a FederatedReplicaSet should be varied in certain clusters.

Kubernetes

Federation

- ▶ TypeConfig
 - specify sync or no sync for the given resource and associated controller(s) to carry out the sync.

Federation v2

Higher-level behaviour

API v2에서 핵심 기능은 API에서 "템플릿", "배치" 및 "오버라이드"를 통해서 컨트롤러(Controller)통해서 상위 API를 재구성(CRD) 합니다.

클러스터를 가로지르면서 자원을 생성 시 "ReplicaSchedulingPreference"를 통해서 Deployment 및 ReplicaSet를 통해서 연합 클러스터에 자원을 생성 및 구성이 가능하다. 역시 기존 클러스터에서 사용하였던 기능 "limits", "quote"같은 자원 관리도 API를 통해서 배포가 가능하다.

OpenShift vs OCM

Components

오픈 시프트의 ACM은 쿠버네티스의 Federation기반으로 되어 있으며, 레드햇은 OCM기반으로 ACM를 개발하기 시작 하였다. 레드햇 ACM은 다음과 같은 부분에 OCM과 다르다.

- ▶ 손쉬운 설치 및 오픈 시프트와 통합된 환경
- ▶ 클라우드 환경에서 손쉬운 확장
- ▶ Quay.io, Ceph, InSight(OpenScape)와 같은 에코 환경 구성
- ▶ 강화된 보안모델 제공

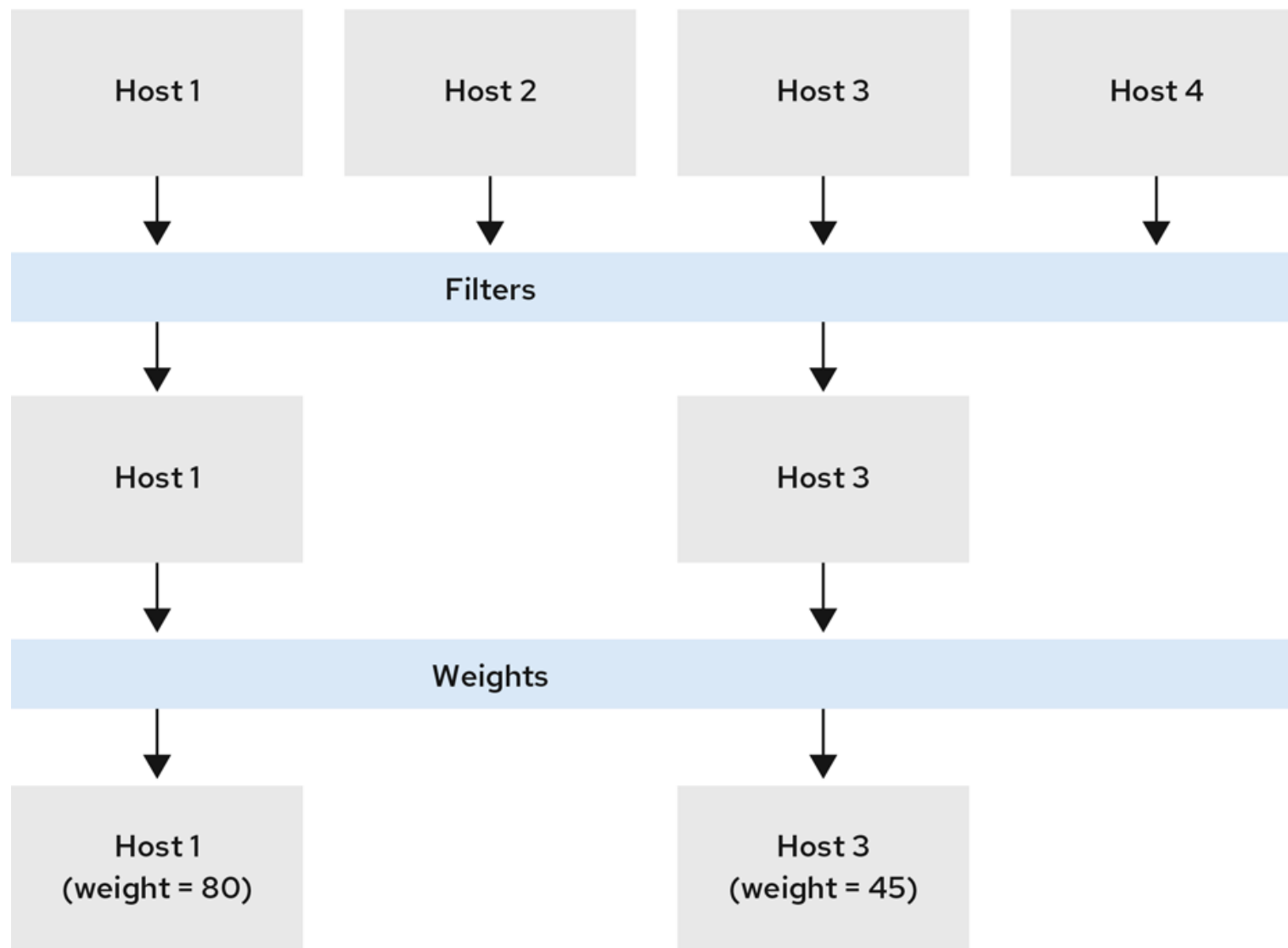
OpenShift ACM

Components

ACM은 오픈시프트와 동일하게 통합된 CI/CD환경을 여러 클러스터에 제공한다. 이때 사용하는 개념은 Hub 및 Subscription기반으로 동작하게 된다. ACM는 Fleets라는 개념기반(앞서 이야기한 Placement)으로 클러스터를 관리한다.

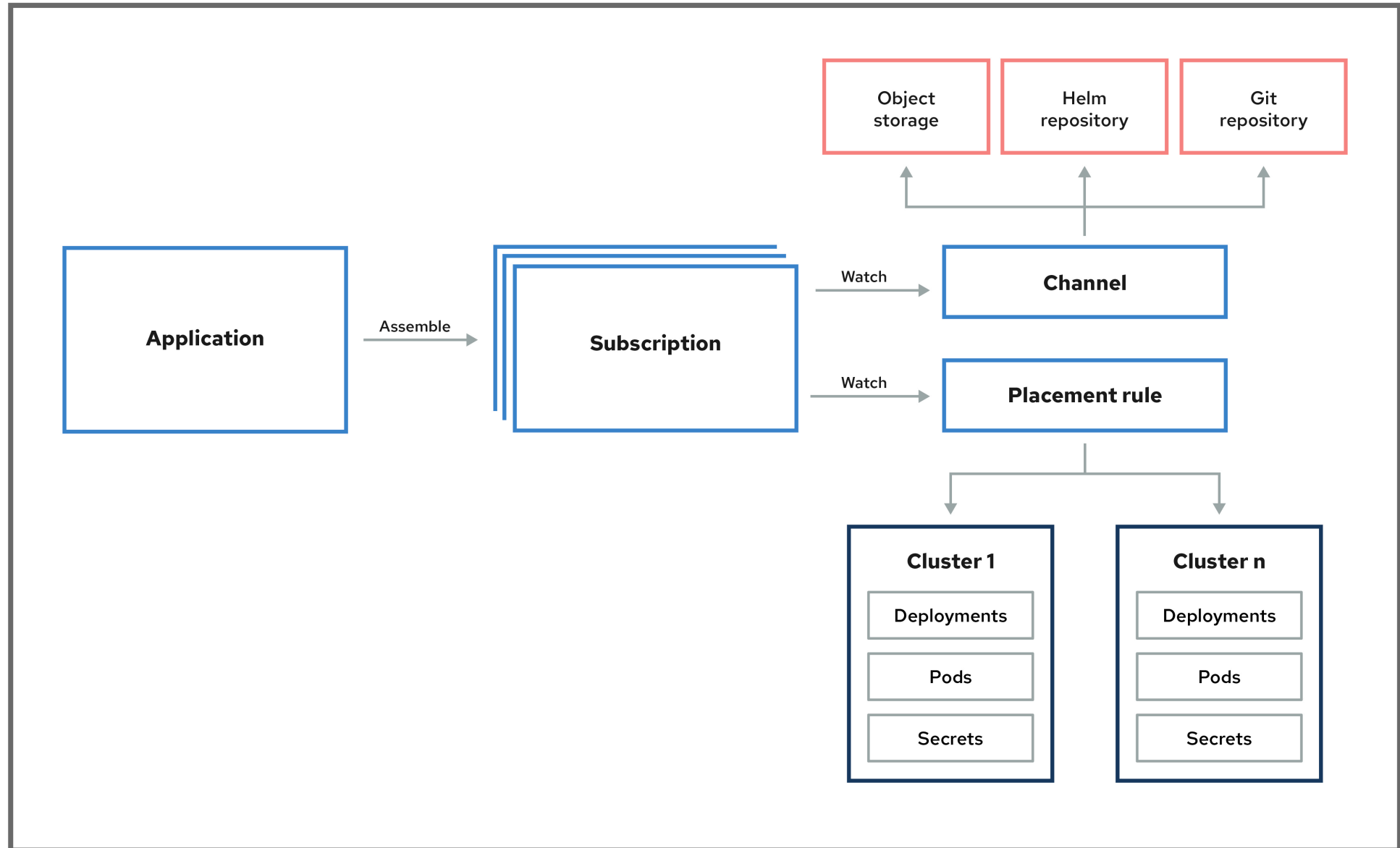
오픈스택의 Placement와 비슷하지만 조금 다른 개념이다.

진행 전, 미리 한번 비교 해본다.



APPLICATION SUBSCRIPTION MODEL

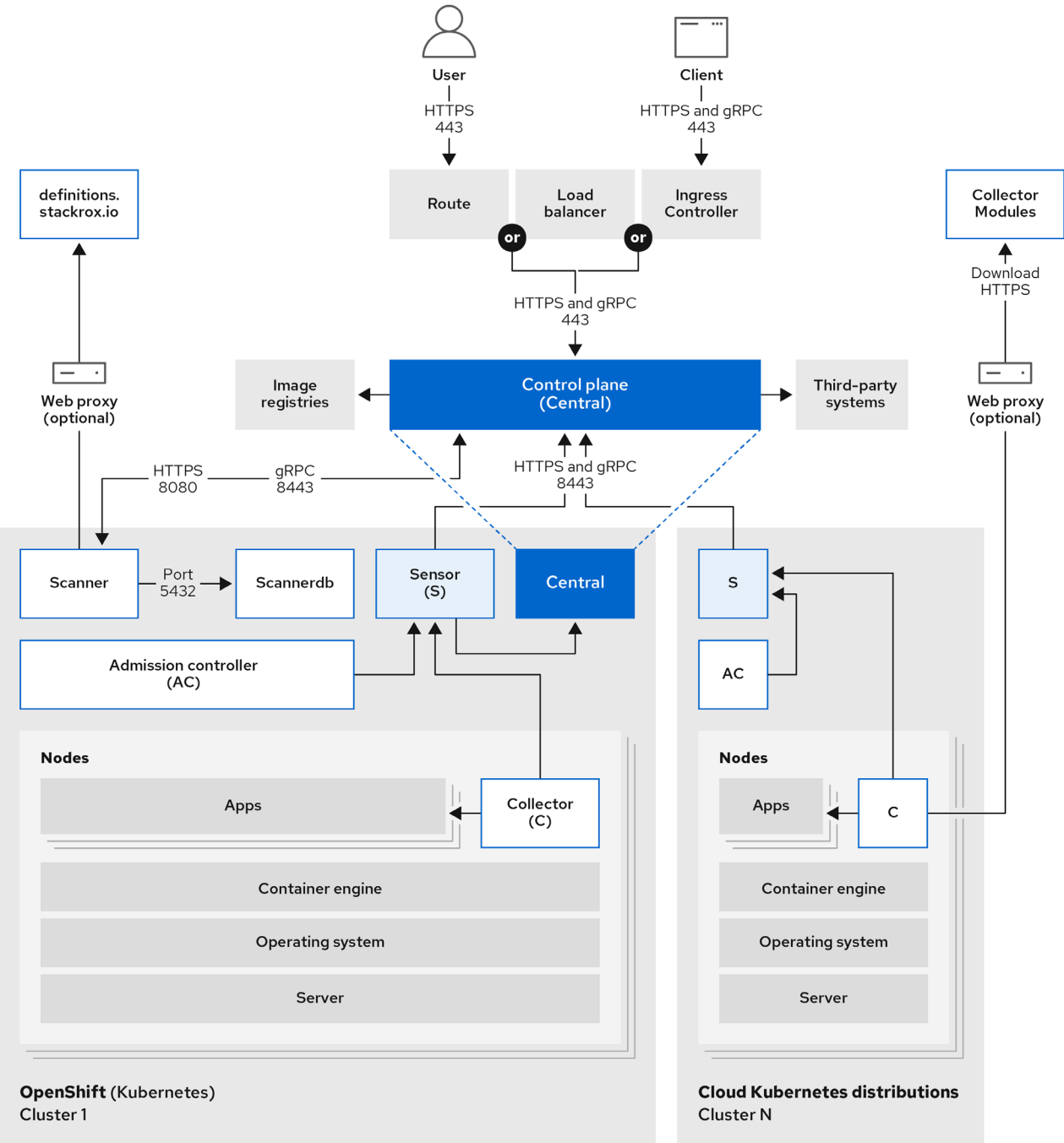
AL designator

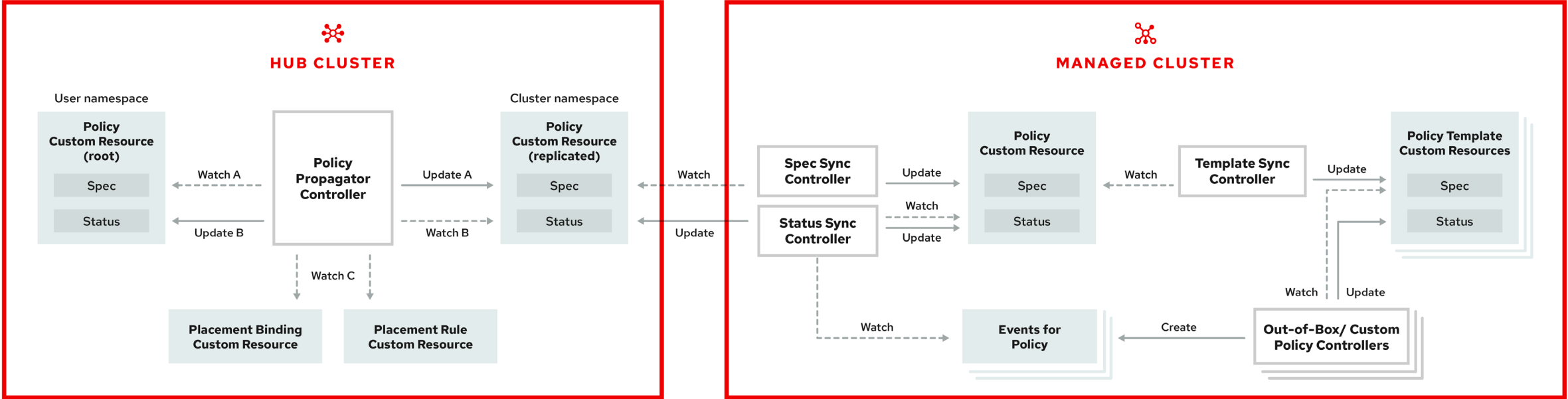


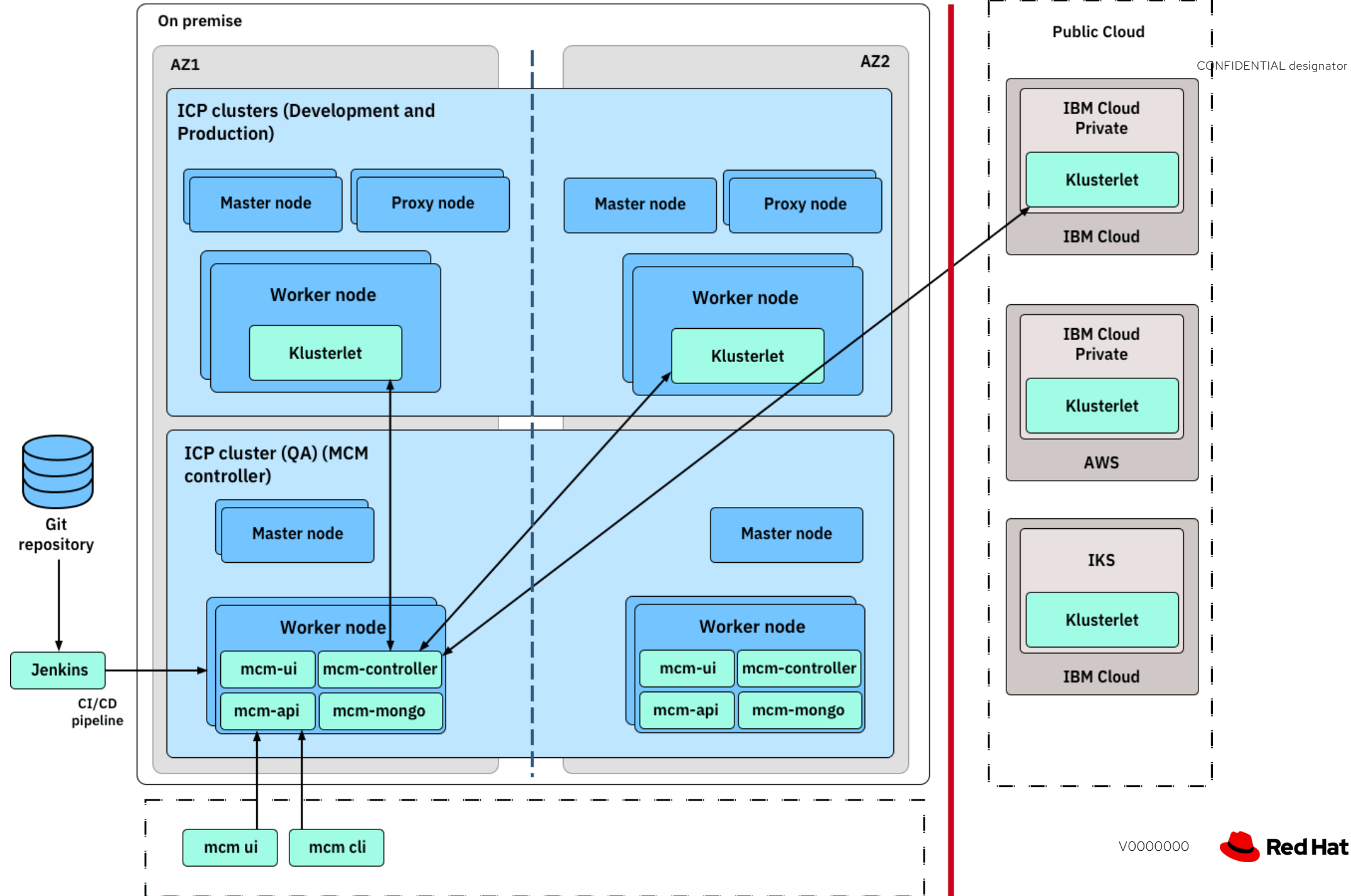
Klusterlet

Components

OCM 그리고 ACM에서는 Klusterlet은 매우 중요하다. Klusterlet은 기존 쿠버네티스나 혹은 오픈시프트에서 사용하는 kubelet하고 비슷한 기능을 한다. 다만, 차이점이라고 하면, "Node to Node"에서 "Cluster to Cluster"로 변경이 되었다.







DAY 2

- ▶ About the OpenShift Advanced Cluster