3 июня 2025 Москва, LOFT HALL#2

БЕКОН '25

Конференция по БЕзопасности
КОНтейнеров и контейнерных сред

Как спрятать Control-Plane Kubernetes от любопытных глаз

Каиржан Аубекеров

MTS Web Services

Кто я?

Каиржан Аубекеров

Software Engineer (Cloud Solutions)

- Специализация: Kubernetes, Cluster API, Practical DevOps, Golang (k8s-development)
- Занимаюсь исследованием и разработкой решений для продукта Managed Kubernetes
- Начинал с поддержки кластеров во внутреннем облаке — Ocean. Наша команда занимается поддержкой 1500+ кластеров для внутренних клиентов

kaubekerov@mts.ru
@kaubekerov

План доклада

01
Предпосылки и цели

02
Постановка задачи

03
Пара слов о Hosted-CP: наша мотивация и тренды

04
kOsmotron

05
Hypershift/OKD

06
Kamaji

07
Итоги и заключение

08
Вопросы и ответы

Предпосылки и цели

А зачем вообще скрывать Control-Plane?

- На CP-нодах хранится всё состояние кластера: манифесты подов, секретов, ключи от SA, корневые сертификаты
- Атаке может быть подвержен любой софт, запущенный на мастерах.
- Один из ключевых векторов атаки — kube-apiserver
- Сами ноды, если они видны, также могут послужить вектором атаки.

Пример уязвимостей, эксплуатируемых при открытых control-plane-нодах

kube-apiserver	CVE-2023-1260	Обход аутентификации: злоумышленник с ограниченными правами может создавать/изменять привилегированные Pod'ы (обход admission-controls)
kube-apiserver	CVE-2023-5408	Обход NodeRestriction plugin: можно подделать label ноды, переместить workload с etcd/CP-нод, получив эскалацию привилегий
kube-apiserver	CVE-2023-45288	Уязвимость HTTP/2: удалённый атакующий может перегрузить API-сервер (DoS). GKE рекомендовал ограничить доступ к API через список доверенных IP-адресов (authorized networks)
kube-apiserver	CVE-2023-2728	Обход admission-политик: позволяет запускать ephemeral-контейнеры с доступом к секретам, даже если это запрещено политикой
etcd	CVE-2023-32082	Утечка ключей lease API: можно узнать имена ключей, даже без прав на чтение их содержимого
secrets-store CSI driver	CVE-2023-2878	Утечка токенов: токены service-account попадали в открытом виде в логи

Какие преимущества дает Hosted-CP

С точки зрения создания и поддержки кластеров Kubernetes

- Более прогнозируемая нагрузка на Control Plane
- Возможность спрятать многие controller-поды
- Упрощается процесс обновления кластера
- Изоляция CP: тренд и good security practice

GKE
Security bulletin

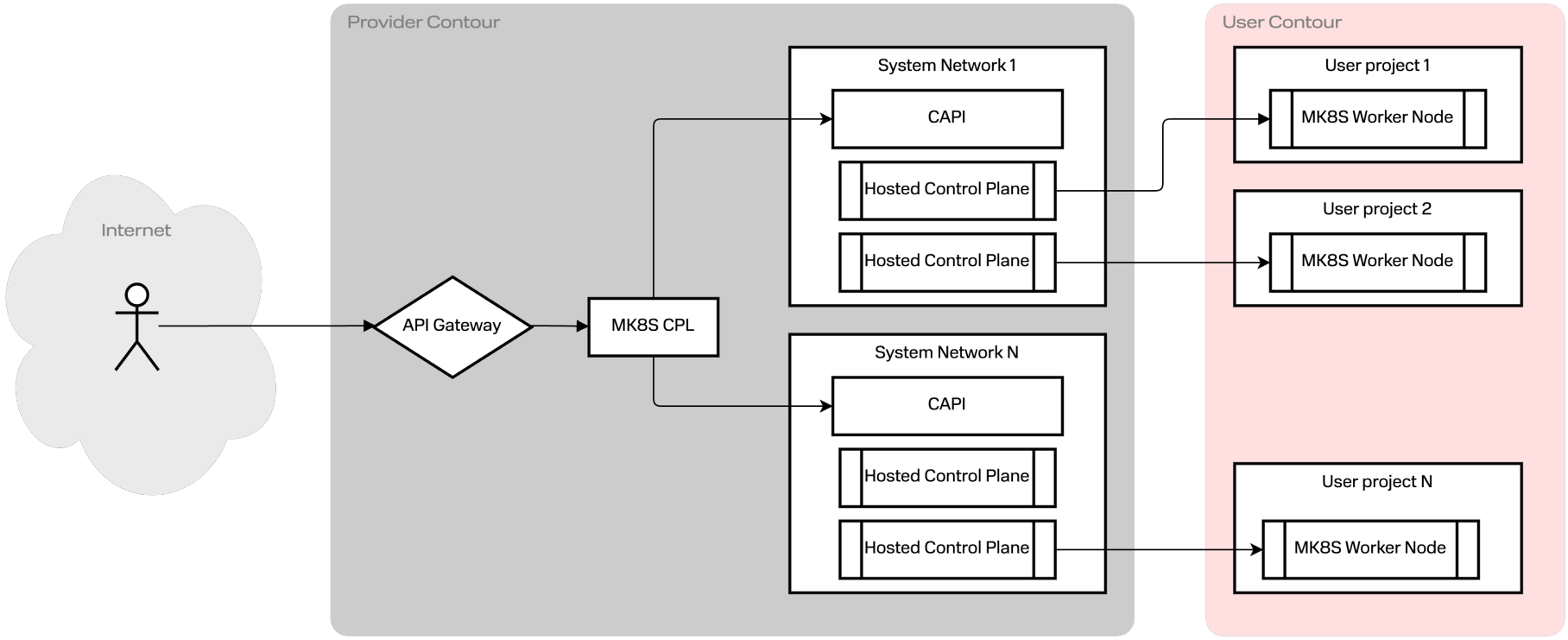
AWS private
subnet

Постановка задачи

Каково наше видение «Идеального Kubernetes» в облаке?

- Используем технологию Cluster-API
- kube-api endpoint as-a-service
- Пользователю видны только Worker-ноды
- Изоляция Control Plane инфраструктурно и по сети
- Ванильный bootstrap Control Plane/Worker nodes

Схема архитектуры Kubernetes в облаке



~~Предпосылки и цели~~

~~Постановка задачи~~

- Пару слов о Hosted-CP: наша мотивация и тренды

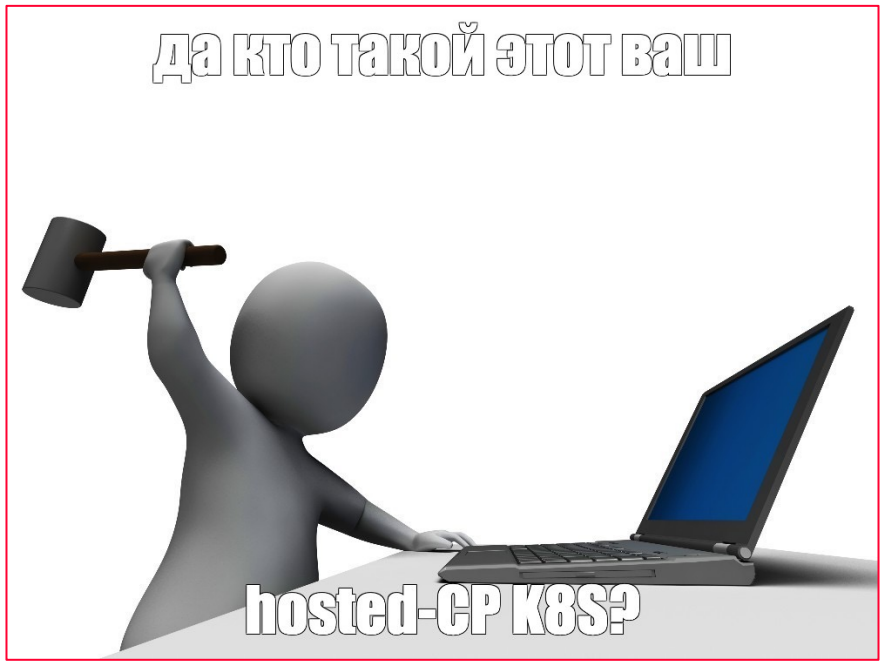
kOsmotron

Hypershift/OKD

Kamaji

Итоги и заключение

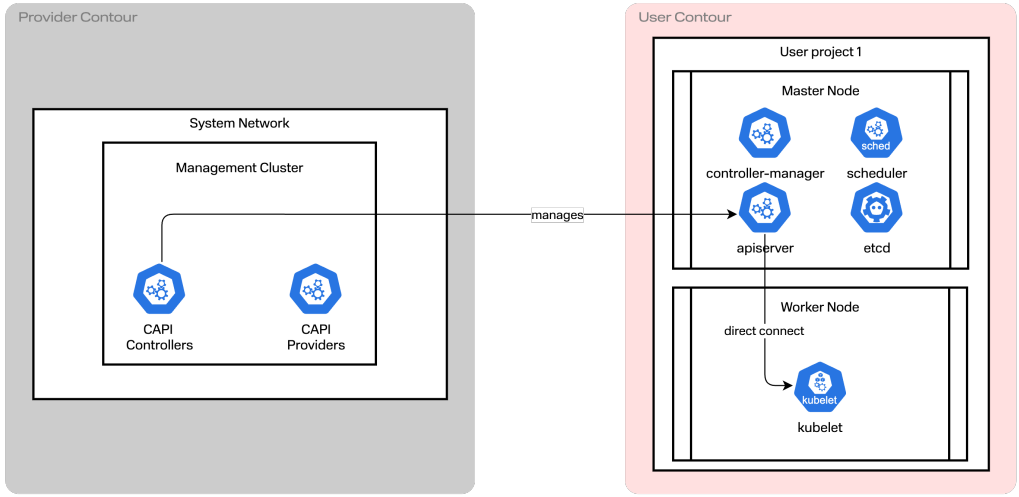
Вопросы и ответы

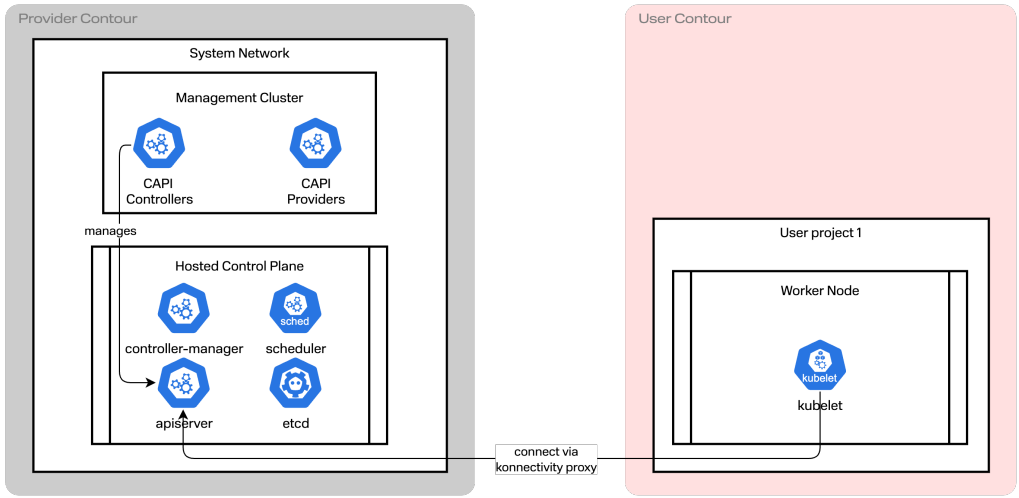


Пару слов о Hosted Control Plane

Что это такое?

- Control Plane в отдельном инфраструктурном сегменте и полностью контролируется провайдером
- Это может быть как workload на отдельно выделенных нодах, так и поды, запущенные в namespace Management-кластера
- CP-ноды не видны пользователю и недоступны для взаимодействия
- Worker-ноды могут находиться где угодно: в публичном облаке, в дата-центре клиента или даже на Edge-устройствах





Что для нас важно?

- Масштабируемость
- Мультитенантность
- Сложность инсталляции и обслуживания
- Требования к инфраструктуре
- Портируемость решения: возможность инсталляции on-prem в приватном облаке

Кто уже использует подобную архитектуру

Из крупных провайдеров на облачном рынке

**Red Hat/OKD — проект [Hypershift](): OpenShift на базе Hosted Control Planes. Под капотом — операторы, управляющие отдельными Control Plane'ами внутри выделенных Namespace. Также есть open-source имплементация в [OKD]()

[Amazon EKS]() и [Amazon Auto Cluster]() — AWS также строит Kubernetes с Hosted Control Plane для упрощённого управления CP

[Google GKE Autopilot]() — концептуально очень близкий подход: полностью управляемый Control Plane без необходимости заботиться о его инфраструктуре

Bonus-пост о трендах на Hosted-CP от Clastix (kamaji):

Рассмотренные нами open-source решения

3 варианта развернуть кластер с Hosted Control-plane

- kOsmotron

Hypershift/OKD

Kamaji

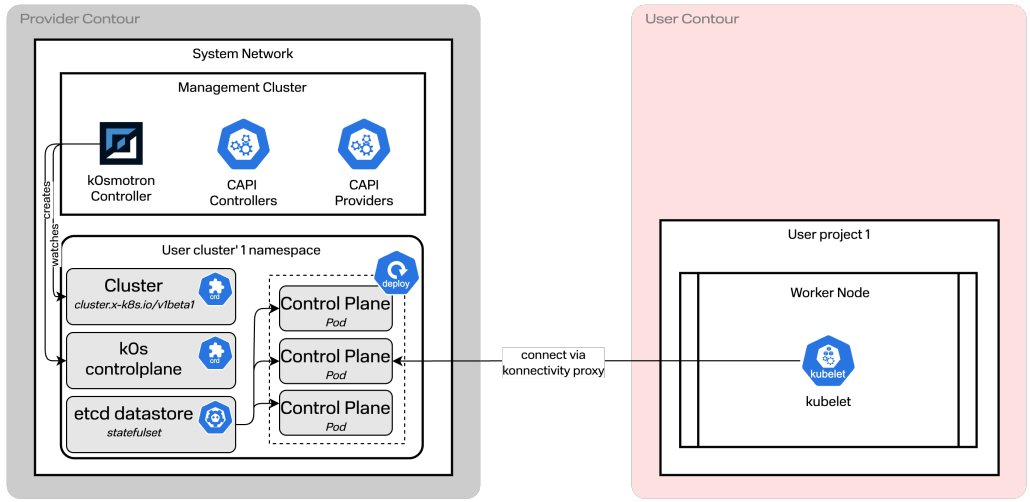
kOsmotron

Провайдер kOs для CAPI

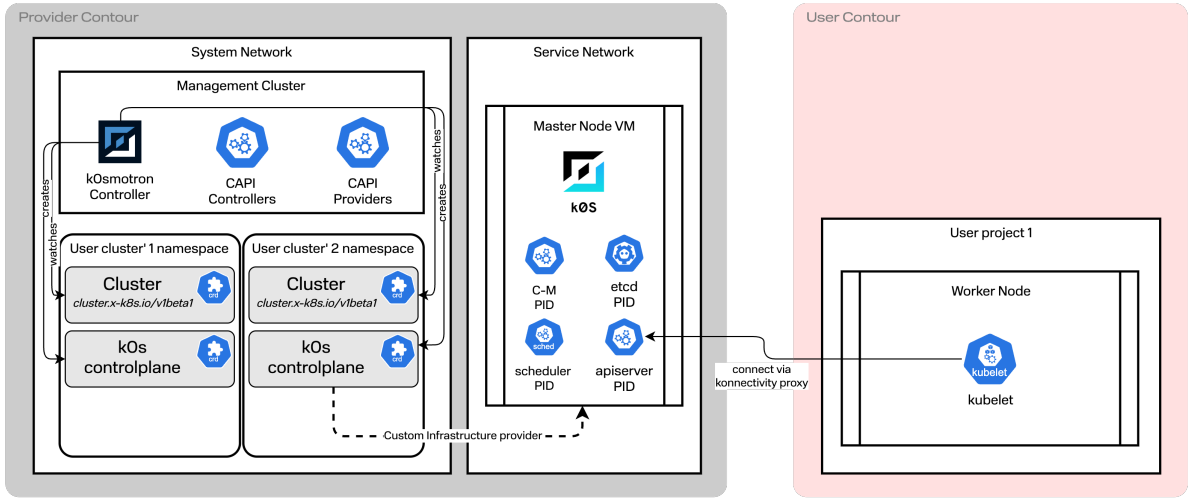
- Под капотом используется kOs: компоненты Kubernetes запускаются как linux-процессы, вне контейнеров.

- Две возможные инсталляции CP:
  - **In-Cluster:** CP в неймспейсе manager-кластера в виде подов
  - **Out-of-Cluster:** kOsmotron выступает в роли бутстрап-провайдера для «традиционных» объектов CAPI Machines

Docs:



kOsmotron In-Cluster



kOsmotron Out-of-Cluster

capi.yaml

```yaml
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: k0s-cp-test
spec:
  clusterNetwork:
    pods:
      cidrBlocks:
        - 10.244.0.0/16
    services:
      cidrBlocks:
        - 10.96.0.0/12
  controlPlaneRef:
    apiVersion: controlplane.cluster.x-k8s.io/v1beta1
    kind: K0sControlPlane
    name: k0s-cp-test
  infrastructureRef:
    apiVersion: infrastructure.ocean.mts.ru/v1alpha1
    kind: VsphereCluster
    name: k0s-cp-test
---
apiVersion: infrastructure.ocean.mts.ru/v1alpha1
kind: VsphereCluster
metadata:
  name: k0s-cp-test
spec:
  * * *

```

k0smotron.yaml

```yaml
apiVersion: controlplane.cluster.x-k8s.io/v1beta1
kind: K0sControlPlane
metadata:
  name: k0s-cp-test
spec:
  replicas: 3
  k0sConfigSpec:
    k0s:
      apiVersion: k0s.k0sproject.io/v1beta1
      kind: ClusterConfig
      metadata:
        name: k0s-cp-test
      spec:
        * * *
  machineTemplate:
    infrastructureRef:
      apiVersion: infrastructure.ocean.mts.ru/v1alpha1
      kind: VSphereMachineTemplate
      name: k0s-cp-test-machine-template
      namespace: k0s-cp-test
---
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: VSphereMachineTemplate
metadata:
  name: k0s-cp-test-machine-template
spec:
  template:
    spec:
      * * *
```

Pros & Cons kOsmotron

✓ Наличие провайдера ClusterAPI	✗ Отсутствие менеджмента над etcd в HA-режиме
✓ Достаточная изоляция CP-k8s, мультитенантность	! Сложности со скейлом в режиме CP-bootstrap
✓ Простота установки	✗ Неванильный процесс бутстрапа CP/Worker'ов
✓ Независимость от инфраструктуры	✗ Неполная совместимость с clusterAPI flow

```
$ kubectl get cluster,machine,kmc

NAME                                  CLUSTERCLASS   PHASE         AGE   VERSION
cluster.cluster.x-k8s.io/my-cluster                  Provisioned   13h

NAME                                                  CLUSTER      NODENAME                      PROVIDERID                                       PHASE         AGE   VERSION
machine.cluster.x-k8s.io/my-cluster-0                 my-cluster                                 vsphere://4215e794-c281-5cde-8193-95df9521cf08   Provisioned   13h   v1.29.1
machine.cluster.x-k8s.io/my-cluster-1                 my-cluster                                 vsphere://4215efc3-303e-938d-3787-c4cc6143f722   Provisioned   13h   v1.29.
machine.cluster.x-k8s.io/my-cluster-2                 my-cluster                                 vsphere://4215b207-1052-e130-54fe-fd124594ed2b   Provisioned   13h   v1.29.1
machine.cluster.x-k8s.io/my-cluster-md-0-vr6s6-ndl64  my-cluster   my-cluster-md-0-vr6s6-ndl64   vsphere://4215eae5-3456-9bc5-4b56-b00dda520b7d   Running       13h   v1.29.1+k0s.1
```

~~KOsmotron~~

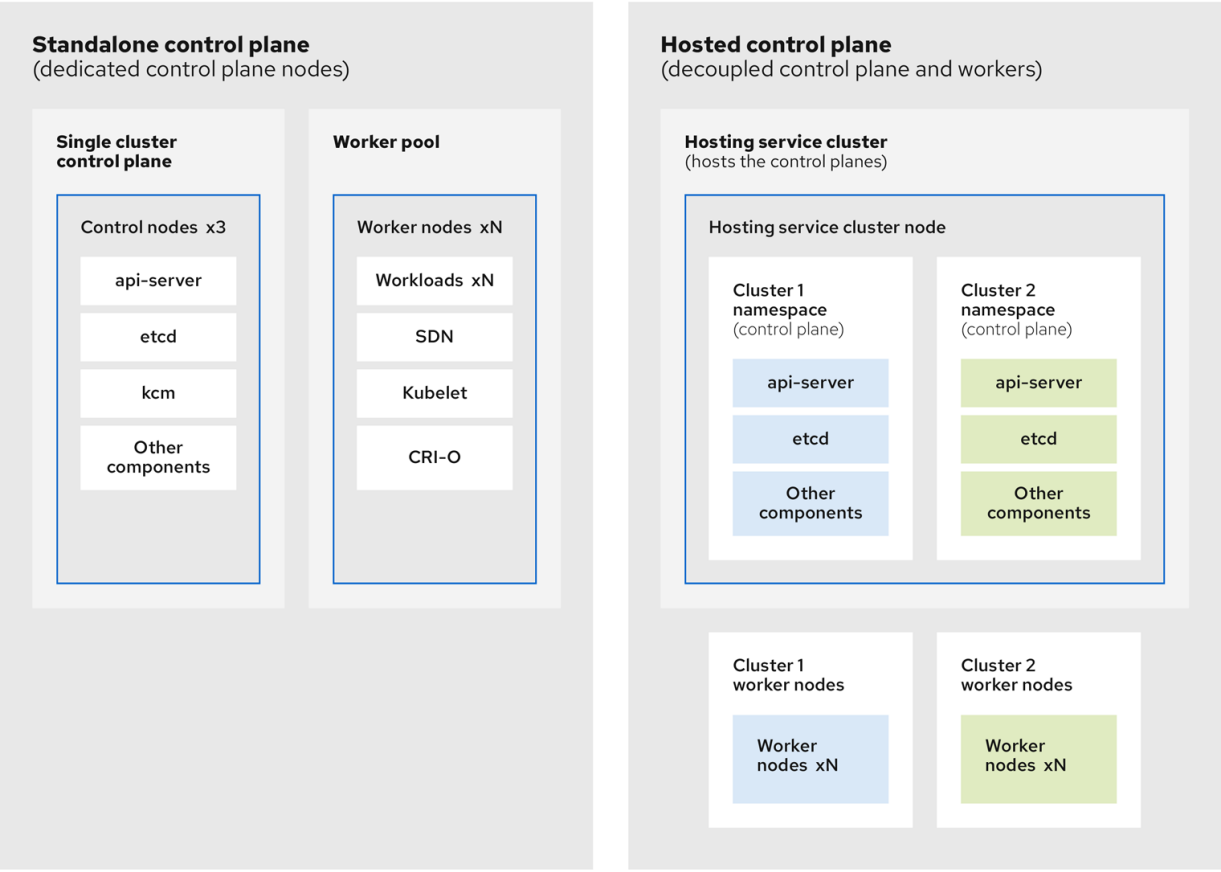
- Hypershift/OKD

Kamaji

Hypershift

Openshift/OKD with Hosted-CP mode

- Набор операторов Openshift для развертки CP в виде подов
- etcd также предоставляется оператором и представляет собой StatefulSet
- CNI, CSI, Ingress, ... - контроллеры живут рядом с CP
- ClusterAPI-like алгоритм: CRD **`hypershift.openshift.io/HostedCluster`** и **`hypershift.openshift.io/NodePool`**



Openshift

Hypershift

OKD

Hosted-CP

Pros & Cons OKD Hosted-CP

✓ Достаточная изоляция CP-k8s, мультитенантность	✗ Неванильный Kubernetes; сильная привязка к API Openshift
✓ Полный менеджмент над всеми сущностями кластера	✗ Нетривиальный процесс инсталляции и поддержки
✓ Централизованное управление над lifecycle k8s	! Ограничения и специфические требования к инфрастрктуре
✓ Зрелый проект, большое сообщество Openshift/OKD	! Не поддерживает ванильный ClusterAPI flow (полностью свой API)

~~KOsmotron~~

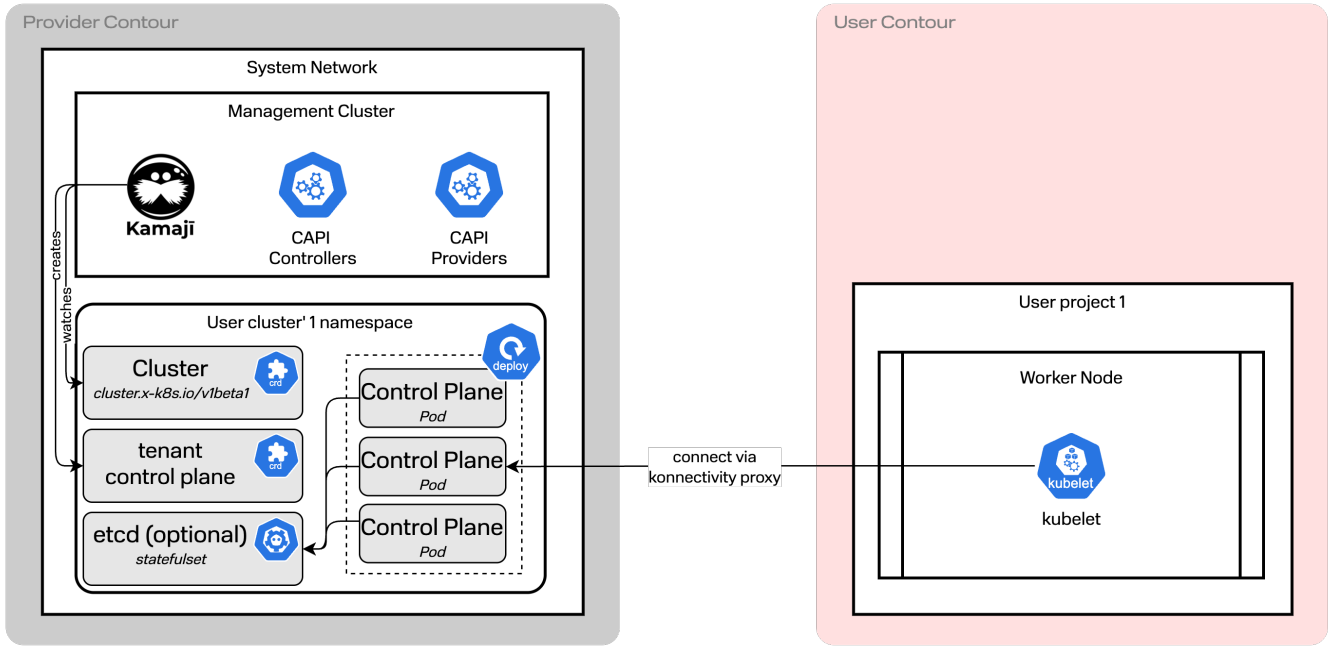
~~Hypershift/OKD~~

- Kamaji

Kamaji

Yet another Control Plane provider

- **Kamaji Controller.** Оператор внутри управляющего кластера
- **Tenant Control Plane.** CRD, порождающий CP в виде деплоймента
- **Data Store.** etcd в виде StatefulSet, либо внешний кластер
- **ClusterAPI CP provider Kamaji.** Провайдер для ClusterAPI



Kamaji docs

CAPI-provider docs

Kamaji

Как это работает?

- Поддерживаются нативные CRD ClusterAPI
- Kamaji Controller порождает Deployment и запускает поды k8s CP
- etcd в виде helm-чарта, либо как external cluster
- Ванильный bootstrap на kubeadm
- Клиентский kube-api endpoint в виде k8s service

kamaji.yaml

```yaml
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: cp-test
spec:
  clusterNetwork:
    * * *
    services:
      * * *
  controlPlaneRef:
    apiVersion: controlplane.cluster.x-k8s.io/v1alpha1
    kind: KamajiControlPlane
    name: cp-test
  infrastructureRef:
    apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
    kind: . . . # Your infrastructure kind may vary
    name: cp-test
---
apiVersion: controlplane.cluster.x-k8s.io/v1alpha1
kind: KamajiControlPlane
metadata:
  name: cp-test
spec:
  * * *

```

```
$ kubectl get kamajicontrolplane,pod,svc,machine

NAME                                                                VERSION   INITIALIZED   READY   AGE
kamajicontrolplane.controlplane.cluster.x-k8s.io/kaubekerov-test    1.28.6    true          true    214d

NAME                                    READY   STATUS    RESTARTS   AGE
pod/kaubekerov-test-5764f9b5bb-jcxln    4/4     Running   0          26d
pod/kaubekerov-test-5764f9b5bb-kwc9k    4/4     Running   0          26d
pod/kaubekerov-test-5764f9b5bb-zjzjg    4/4     Running   0          26d

NAME                        TYPE           CLUSTER-IP       EXTERNAL-IP     PORT(S)                         AGE
service/kaubekerov-test     LoadBalancer   10.222.100.152   10.31.201.142   6443:32122/TCP,8132:31467/TCP   214d

NAME                                                                CLUSTER          NODENAME                       PROVIDERID                                 PHASE     AGE    VERSION
machine.cluster.x-k8s.io/kubemark-external-md-0-l5zl8-926fn         kaubekerov-test  kubemark-external-md-0-wvrvr   kubemark://kubemark-external-md-0-wvrvr    Running   214d   v1.28.6
machine.cluster.x-k8s.io/kubemark-external-md-0-l5zl8-f67bp         kaubekerov-test  kubemark-external-md-0-c66kr   kubemark://kubemark-external-md-0-c66kr    Running   214d   v1.28.6
```

Control Plane в виде Pods kamaji, Worker Nodes от внешнего провайдера

Pros & Cons Kamaji

✓ Наличие провайдера ClusterAPI, неплохая интеграция	! Необходимость в менеджменте etcd, либо external etcd
✓ Достаточная изоляция CP-k8s, мультитенантность	! DIY-процесс бутстрапа воркер-нод
✓ Простота установки	! Относительно небольшое сообщество, на момент тестирования встречаются баги
✓ Полная независимость от инфраструктуры	! Проект выводится из open-source, стабильные билды выходят по подписке
✓ Ванильный бутстрап, построенный на kubedm	

Feature	kOsmotron	HyperShift	Kamaji
Ванильный bootstrap	✗ (k0s-based)	✗ (только OpenShift)	✓ Да (kubeadm)
Изоляция Control Plane	✓ На уровне pod/ns или cluster/infra	✓ На уровне OS Management cluster	✓ На уровне pod/namespace
Infra-Agnostic	✓ Да (in-cluster)	✗ Только OpenShift	✓ Да
Bootstrap worker-нод	✓ Токены из CRD/k0s b-strap	! Требует Ignition/OpenShift	✗ Вручную / внешние ноды
HA-инсталляция	✗ Проблемы с etcd	✓ Да	✓ Да
Совместимость с CAPI	✓ Нативные CRD	✗ OS Hosted-Cluster CRD	✓ Нативные CRD
Сложность в эксплуатации	✓ Низкая	✗ Высокая	✓ Низкая
etcd-менеджмент	✗ Вручную/k0s (может упасть в HA-режиме)	! Операторы OS (StS, external)	! Helm (StS), либо external

Заключение

А что в итоге взяли мы?

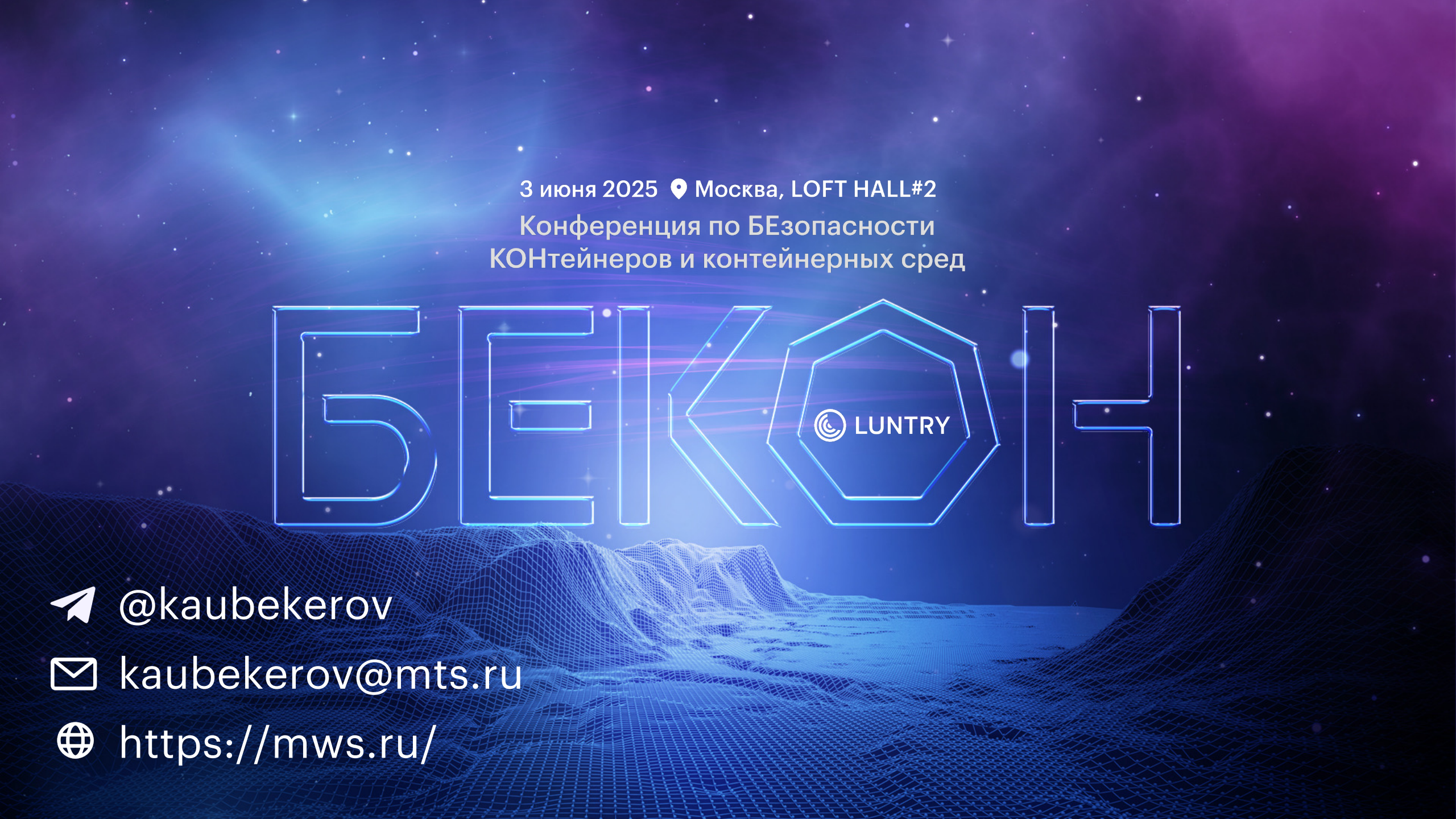
- Коротко – ничего из перечисленного
- Берем все лучшее из провайдеров
- На этой основе пишем свой kubeadm-провайдер
- kubeadm-stages + kubelet: registerNode: false

kubeadm.yaml

```yaml
apiVersion: kubeadm.k8s.io/v1beta4
kind: InitConfiguration
localAPIEndpoint: {}
skipPhases:    # Skip certain kubeadm phases we do not need
- addon
- certs/ca
- certs/front-proxy-ca
- certs/etcd-ca
- certs/sa
- upload-config/kubelet
- upload-certs
- mark-control-plane
- kubelet-finalize
- show-join-command
---
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
authentication:
  * * *
  x509:
    clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
  * * *
address: 127.0.0.1
readOnlyPort: 10255
cgroupDriver: systemd
serializeImagePulls: false

registerNode: false # Do NOT register node!

staticPodPath: /etc/kubernetes/manifests
containerRuntimeEndpoint: unix:///run/containerd/containerd.sock
```


3 июня 2025 Москва, LOFT HALL#2
Конференция по БЕзопасности
КОНтейнеров и контейнерных сред
БЕКОН
LUNTRY
@kaubekerov
kaubekerov@mts.ru
https://mws.ru/