前言: 问题描述

当前方案: 在多租户场景下，交付以裸机 (目前主要指 x86) 为主要算力，KubeVirt VM为弹性算力的 k8s clusters。

问题描述: 由Management k8s cluster 管理和维护所有租户的 k8s clusters 的生命周期。

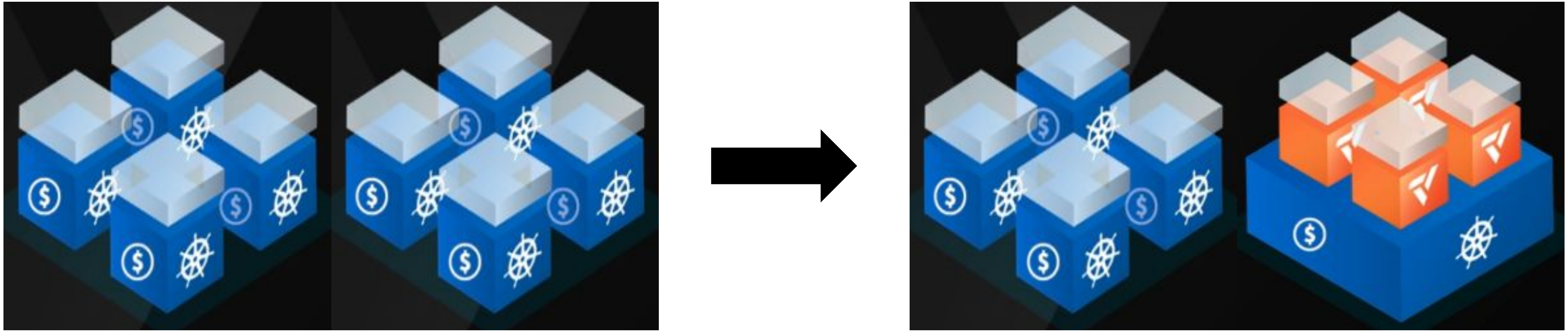
- 优点：租户与租户之间是真实物理 k8s cluster 级别的隔离。
- 缺点：1）每个租户各自维护自己一套或者多套的 k8s clusters，资源开销大；2）当租户数量达到一定数量后，管理其 k8s clusters 会非常复杂。

前言: 解决方案 v1

解决方案: 在隔离和资源开销中寻找新的平衡。

kubernetes

vCluster

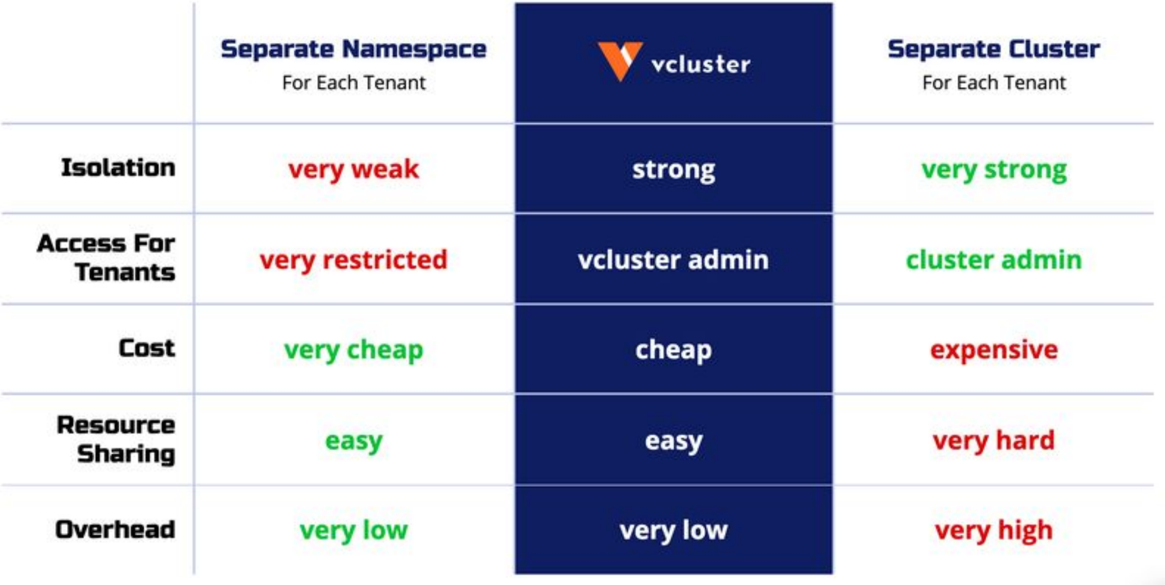


Agenda

- vCluster: Introduction & Benefits
- vCluster: Key Concepts & Demo
  - Control Plane & Demo
  - Pods/Deployments/Services & Demo
  - Networking & Demo
- vCluster: Features
  - Multi-Tenancy
  - Better Isolation
  - Better Performance
  - Compatibility
    - Integrate vCluster with KubeVirt
    - Integrate vCluster with Cluster API
- References

vCluster: Introduction & Benefits

vClusters are fully functional k8s clusters nested inside a physical host cluster providing better isolation and flexibility to support multi-tenancy, With vClusters, multiple teams can operate independently within the same physical infrastructure while minimizing conflicts, maximizing autonomy, and reducing costs [1].



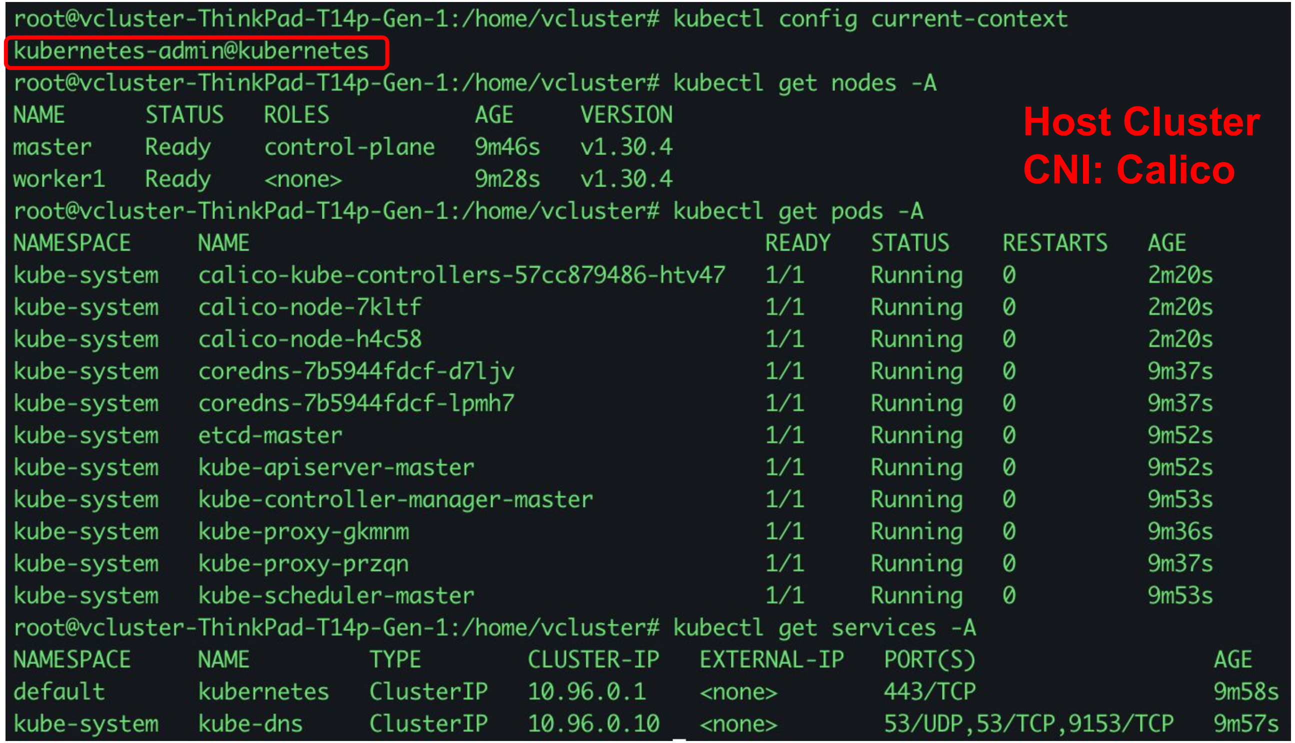
	Separate Namespace For Each Tenant	vcluster	Separate Cluster For Each Tenant
Isolation	very weak	strong	very strong
Access For Tenants	very restricted	vcluster admin	cluster admin
Cost	very cheap	cheap	expensive
Resource Sharing	easy	easy	very hard
Overhead	very low	very low	very high

Figure: Comparison among Namespace, vCluster, Cluster [1]

vCluster: Control Plane

- vCluster’s control plane runs as a pod in host cluster
- vCluster’s control plane contains:
  - api server
  - controller manager
  - data store mount (eg, etcd)
  - by default a syncer (optionally a scheduler)

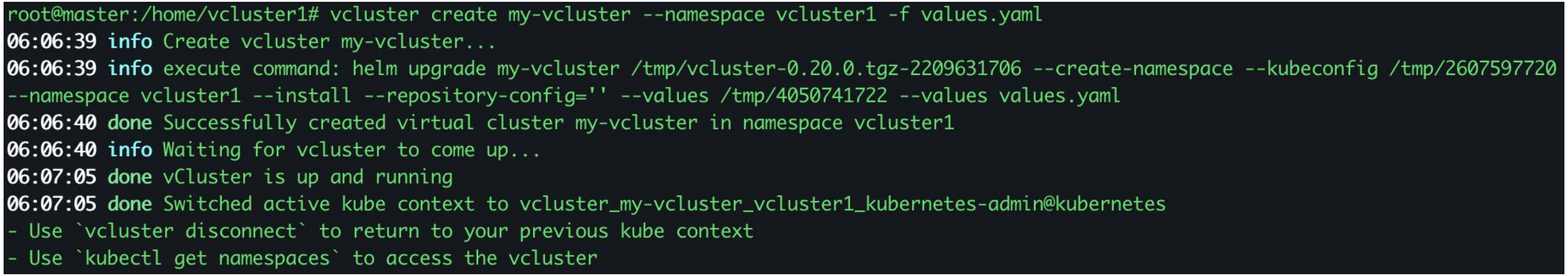
Demo: Env Description



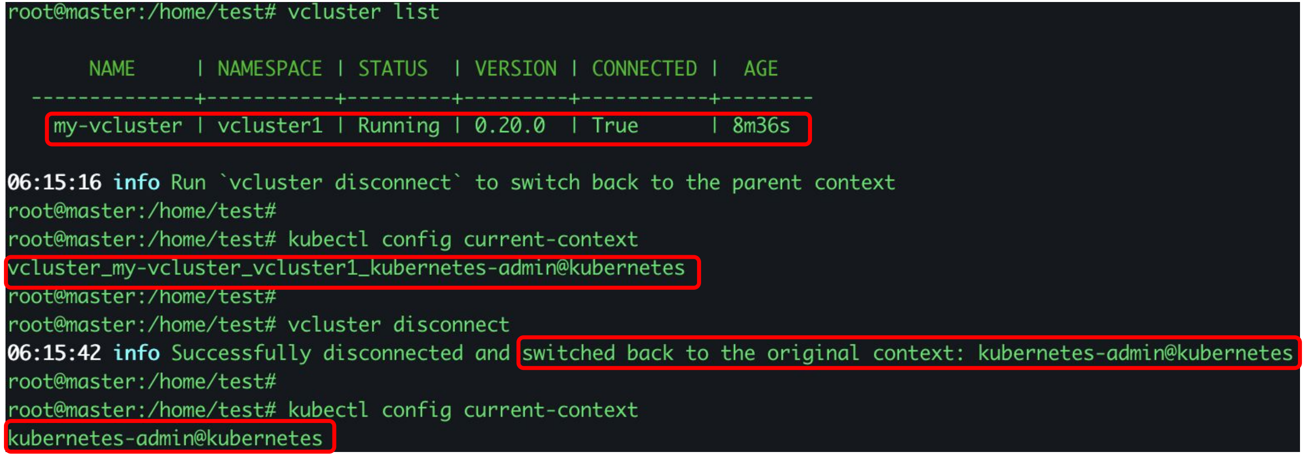
```
root@vcluster-ThinkPad-T14p-Gen-1:/home/vcluster# kubectl config current-context
kubernetes-admin@kubernetes
root@vcluster-ThinkPad-T14p-Gen-1:/home/vcluster# kubectl get nodes -A
NAME      STATUS   ROLES           AGE     VERSION
master    Ready    control-plane   9m46s   v1.30.4
worker1   Ready    <none>          9m28s   v1.30.4
root@vcluster-ThinkPad-T14p-Gen-1:/home/vcluster# kubectl get pods -A
NAMESPACE     NAME                                       READY   STATUS    RESTARTS   AGE
kube-system   calico-kube-controllers-57cc879486-htv47   1/1     Running   0          2m20s
kube-system   calico-node-7kltf                          1/1     Running   0          2m20s
kube-system   calico-node-h4c58                          1/1     Running   0          2m20s
kube-system   coredns-7b5944fdcf-d7ljv                   1/1     Running   0          9m37s
kube-system   coredns-7b5944fdcf-lpmh7                   1/1     Running   0          9m37s
kube-system   etcd-master                                1/1     Running   0          9m52s
kube-system   kube-apiserver-master                      1/1     Running   0          9m52s
kube-system   kube-controller-manager-master             1/1     Running   0          9m53s
kube-system   kube-proxy-gkmnm                           1/1     Running   0          9m36s
kube-system   kube-proxy-przqn                           1/1     Running   0          9m37s
kube-system   kube-scheduler-master                      1/1     Running   0          9m53s
root@vcluster-ThinkPad-T14p-Gen-1:/home/vcluster# kubectl get services -A
NAMESPACE     NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)                  AGE
default       kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP                  9m58s
kube-system   kube-dns     ClusterIP   10.96.0.10   <none>        53/UDP,53/TCP,9153/TCP   9m57s
```

Host Cluster
CNI: Calico

Demo: Create a vCluster (1)

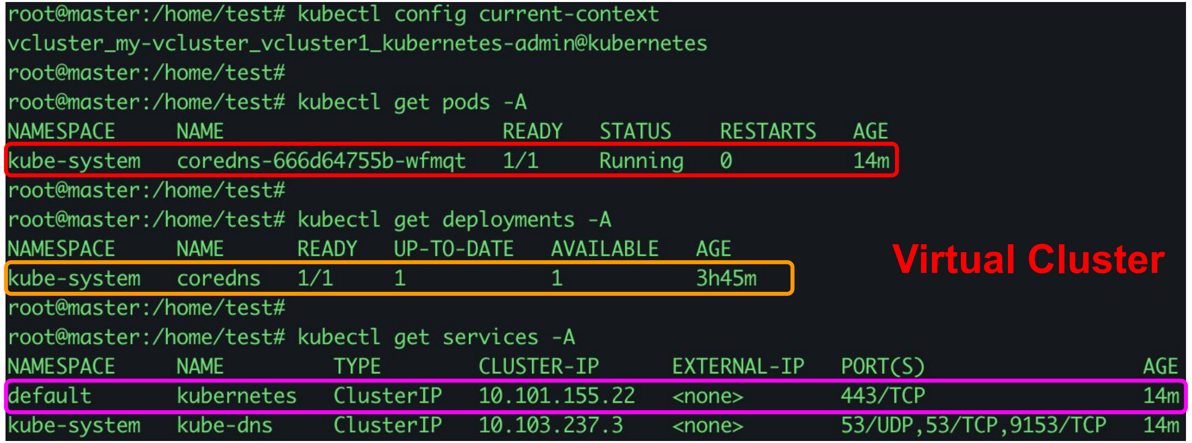


```
root@master:/home/vcluster1# vcluster create my-vcluster --namespace vcluster1 -f values.yaml
06:06:39 info Create vcluster my-vcluster...
06:06:39 info execute command: helm upgrade my-vcluster /tmp/vcluster-0.20.0.tgz-2209631706 --create-namespace --kubeconfig /tmp/2607597720
--namespace vcluster1 --install --repository-config='' --values /tmp/4050741722 --values values.yaml
06:06:40 done Successfully created virtual cluster my-vcluster in namespace vcluster1
06:06:40 info Waiting for vcluster to come up...
06:07:05 done vCluster is up and running
06:07:05 done Switched active kube context to vcluster_my-vcluster_vcluster1_kubernetes-admin@kubernetes
- Use `vcluster disconnect` to return to your previous kube context
- Use `kubectl get namespaces` to access the vcluster
```



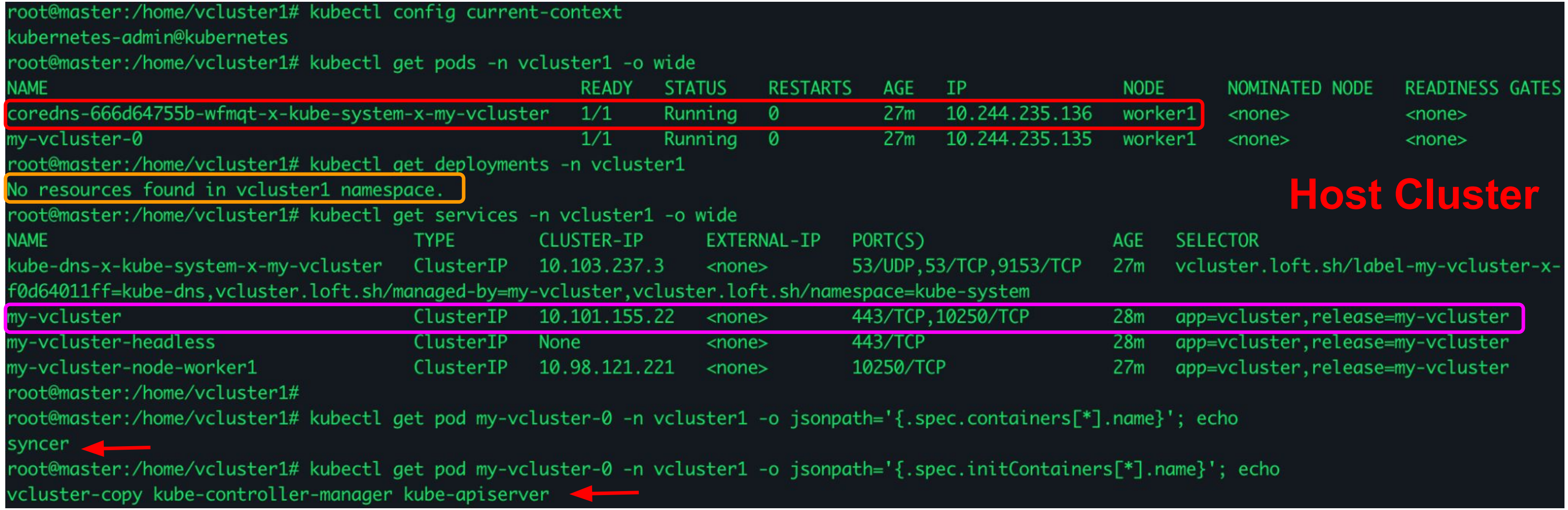
```
root@master:/home/test# vcluster list

      NAME      | NAMESPACE | STATUS  | VERSION | CONNECTED |  AGE
  --------------+-----------+---------+---------+-----------+--------
    my-vcluster | vcluster1 | Running | 0.20.0  | True      | 8m36s

06:15:16 info Run `vcluster disconnect` to switch back to the parent context
root@master:/home/test#
root@master:/home/test# kubectl config current-context
vcluster_my-vcluster_vcluster1_kubernetes-admin@kubernetes
root@master:/home/test#
root@master:/home/test# vcluster disconnect
06:15:42 info Successfully disconnected and switched back to the original context: kubernetes-admin@kubernetes
root@master:/home/test#
root@master:/home/test# kubectl config current-context
kubernetes-admin@kubernetes
```

Demo: Create a vCluster (2)



```
root@master:/home/test# kubectl config current-context
vcluster_my-vcluster_vcluster1_kubernetes-admin@kubernetes
root@master:/home/test#
root@master:/home/test# kubectl get pods -A
NAMESPACE     NAME                       READY   STATUS    RESTARTS   AGE
kube-system   coredns-666d64755b-wfmqt   1/1     Running   0          14m
root@master:/home/test#
root@master:/home/test# kubectl get deployments -A
NAMESPACE     NAME      READY   UP-TO-DATE   AVAILABLE   AGE
kube-system   coredns   1/1     1            1           3h45m
root@master:/home/test#
root@master:/home/test# kubectl get services -A
NAMESPACE     NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                  AGE
default       kubernetes   ClusterIP   10.101.155.22   <none>        443/TCP                  14m
kube-system   kube-dns     ClusterIP   10.103.237.3    <none>        53/UDP,53/TCP,9153/TCP   14m
```

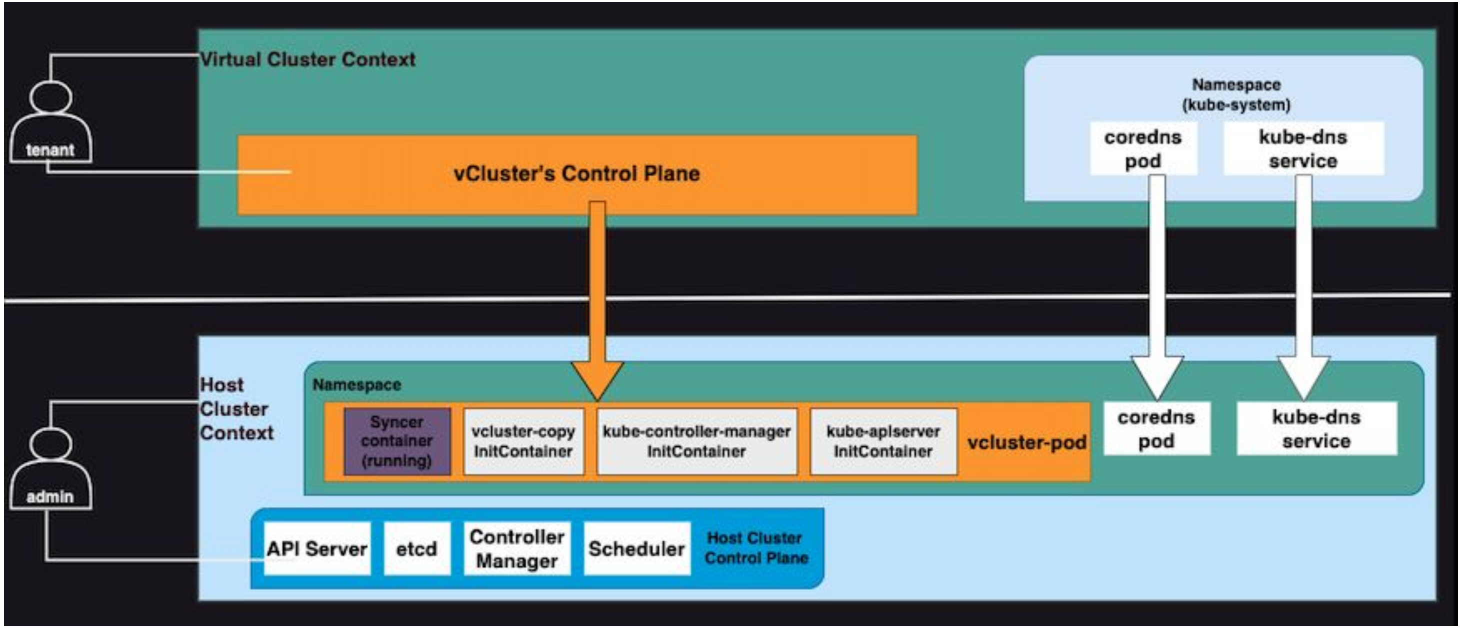
Virtual Cluster



```
root@master:/home/vcluster1# kubectl config current-context
kubernetes-admin@kubernetes
root@master:/home/vcluster1# kubectl get pods -n vcluster1 -o wide
NAME                                           READY   STATUS    RESTARTS   AGE   IP               NODE      NOMINATED NODE   READINESS GATES
coredns-666d64755b-wfmqt-x-kube-system-x-my-vcluster   1/1     Running   0          27m   10.244.235.136   worker1   <none>           <none>
my-vcluster-0                                  1/1     Running   0          27m   10.244.235.135   worker1   <none>           <none>
root@master:/home/vcluster1# kubectl get deployments -n vcluster1
No resources found in vcluster1 namespace.
root@master:/home/vcluster1# kubectl get services -n vcluster1 -o wide
NAME                                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                  AGE   SELECTOR
kube-dns-x-kube-system-x-my-vcluster   ClusterIP   10.103.237.3    <none>        53/UDP,53/TCP,9153/TCP   27m   vcluster.loft.sh/label-my-vcluster-x-
f0d64011ff=kube-dns,vcluster.loft.sh/managed-by=my-vcluster,vcluster.loft.sh/namespace=kube-system
my-vcluster                           ClusterIP   10.101.155.22   <none>        443/TCP,10250/TCP        28m   app=vcluster,release=my-vcluster
my-vcluster-headless                  ClusterIP   None            <none>        443/TCP                  28m   app=vcluster,release=my-vcluster
my-vcluster-node-worker1              ClusterIP   10.98.121.221   <none>        10250/TCP                27m   app=vcluster,release=my-vcluster
root@master:/home/vcluster1#
root@master:/home/vcluster1# kubectl get pod my-vcluster-0 -n vcluster1 -o jsonpath='{.spec.containers[*].name}'; echo
syncer
root@master:/home/vcluster1# kubectl get pod my-vcluster-0 -n vcluster1 -o jsonpath='{.spec.initContainers[*].name}'; echo
vcluster-copy kube-controller-manager kube-apiserver
```

Host Cluster

Summary


Virtual Cluster Context
tenant
vCluster's Control Plane
Namespace
(kube-system)
coredns
pod
kube-dns
service
Host
Cluster
Context
admin
Namespace
Syncer
container
(running)
vcluster-copy
InitContainer
kube-controller-manager
InitContainer
kube-apiserver
InitContainer
vcluster-pod
coredns
pod
kube-dns
service
API Server
etcd
Controller
Manager
Scheduler
Host Cluster
Control Plane

vCluster: Pods/Deployments/Services

A vCluster doesn't have actual worker nodes or a network.

By default, the syncer synchronizes certain **low-level vCluster Pod resources** to the host namespace so that the host cluster scheduler can schedule these pods with access to these resources.

The syncer also propagates certain changes made in the host cluster back into the virtual cluster.

Syncing = Low-Level Resources

- Pods, Plus:
  - Mounted ConfigMaps
  - Mounted Secrets
  - Persistent Volumes & Claims
- Services
- Ingresses (Optional)
- Nodes (Configurable)

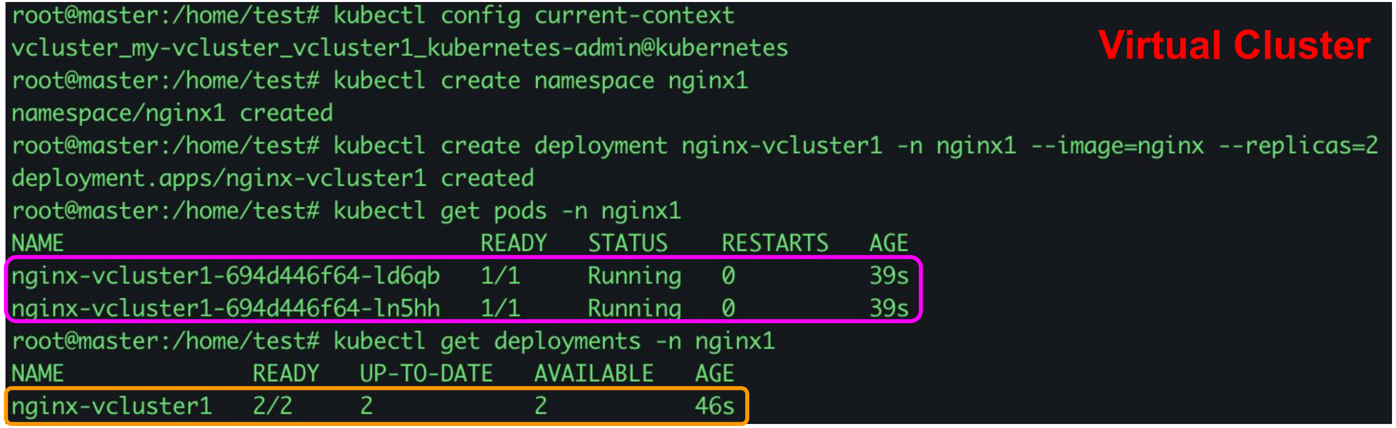
Syncer syncs back the status of each object.

Not Syncing = High-Level Resources

- Replica Controlled Resources
  - Deployments
  - StatefulSets
  - DaemonSets
- Not (yet) Mounted ConfigMaps, Secrets
- Other: Service Accounts, Jobs, etc.
- Custom Resources (+CRDs)

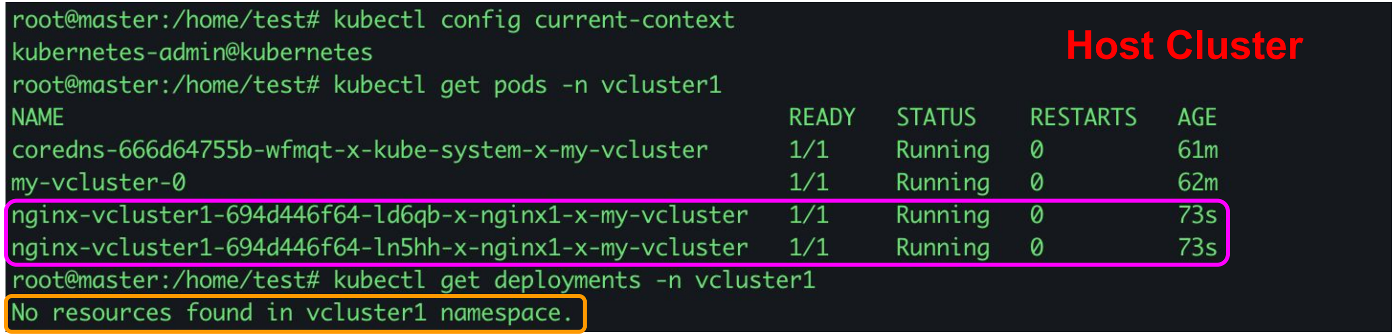
The vast majority of objects will only exist in the vcluster.

Demo: Create a Deployment in vCluster



Virtual Cluster

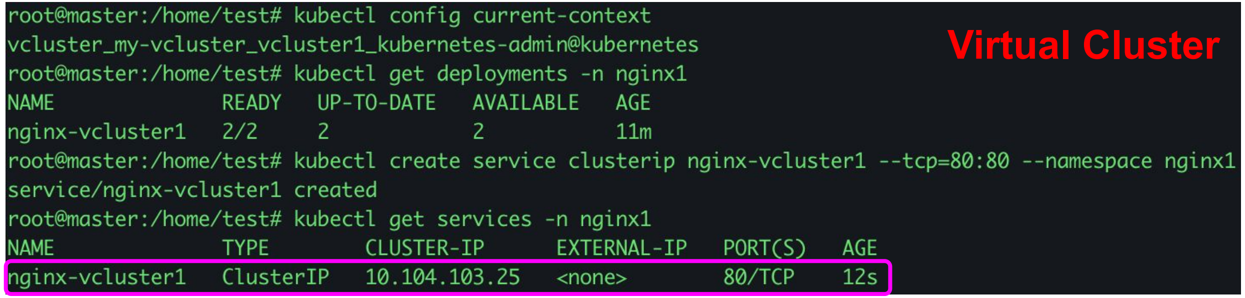
```
root@master:/home/test# kubectl config current-context
vcluster_my-vcluster_vcluster1_kubernetes-admin@kubernetes
root@master:/home/test# kubectl create namespace nginx1
namespace/nginx1 created
root@master:/home/test# kubectl create deployment nginx-vcluster1 -n nginx1 --image=nginx --replicas=2
deployment.apps/nginx-vcluster1 created
root@master:/home/test# kubectl get pods -n nginx1
NAME                               READY   STATUS    RESTARTS   AGE
nginx-vcluster1-694d446f64-ld6qb   1/1     Running   0          39s
nginx-vcluster1-694d446f64-ln5hh   1/1     Running   0          39s
root@master:/home/test# kubectl get deployments -n nginx1
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
nginx-vcluster1   2/2     2            2           46s
```



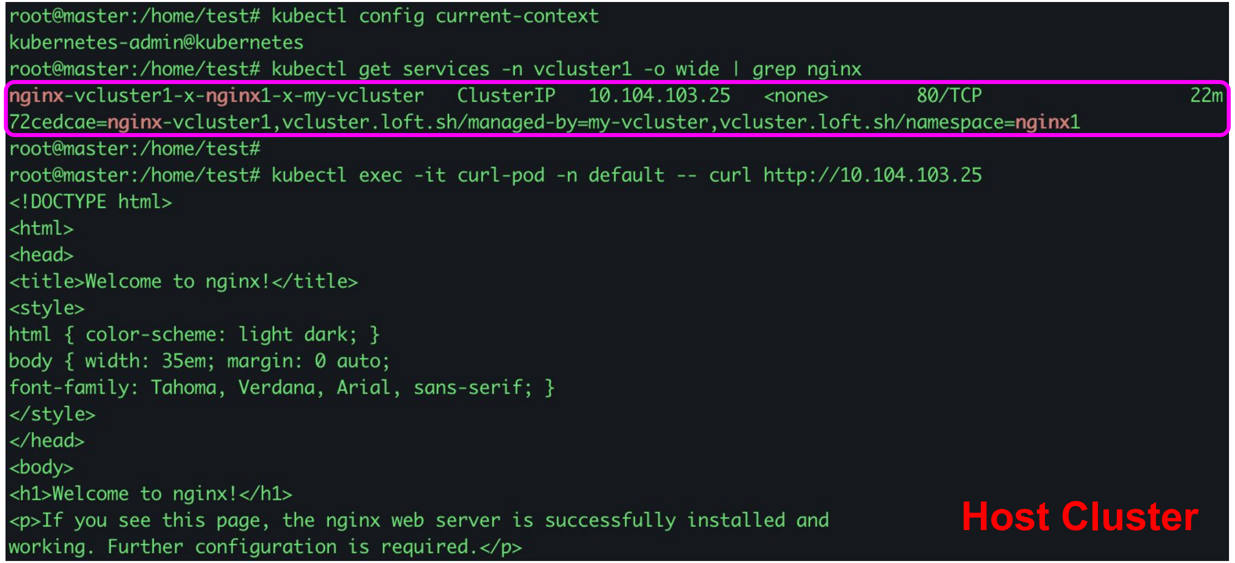
Host Cluster

```
root@master:/home/test# kubectl config current-context
kubernetes-admin@kubernetes
root@master:/home/test# kubectl get pods -n vcluster1
NAME                                                  READY   STATUS    RESTARTS   AGE
coredns-666d64755b-wfmqt-x-kube-system-x-my-vcluster  1/1     Running   0          61m
my-vcluster-0                                         1/1     Running   0          62m
nginx-vcluster1-694d446f64-ld6qb-x-nginx1-x-my-vcluster   1/1     Running   0          73s
nginx-vcluster1-694d446f64-ln5hh-x-nginx1-x-my-vcluster   1/1     Running   0          73s
root@master:/home/test# kubectl get deployments -n vcluster1
No resources found in vcluster1 namespace.
```

Demo: Create a Service in vCluster

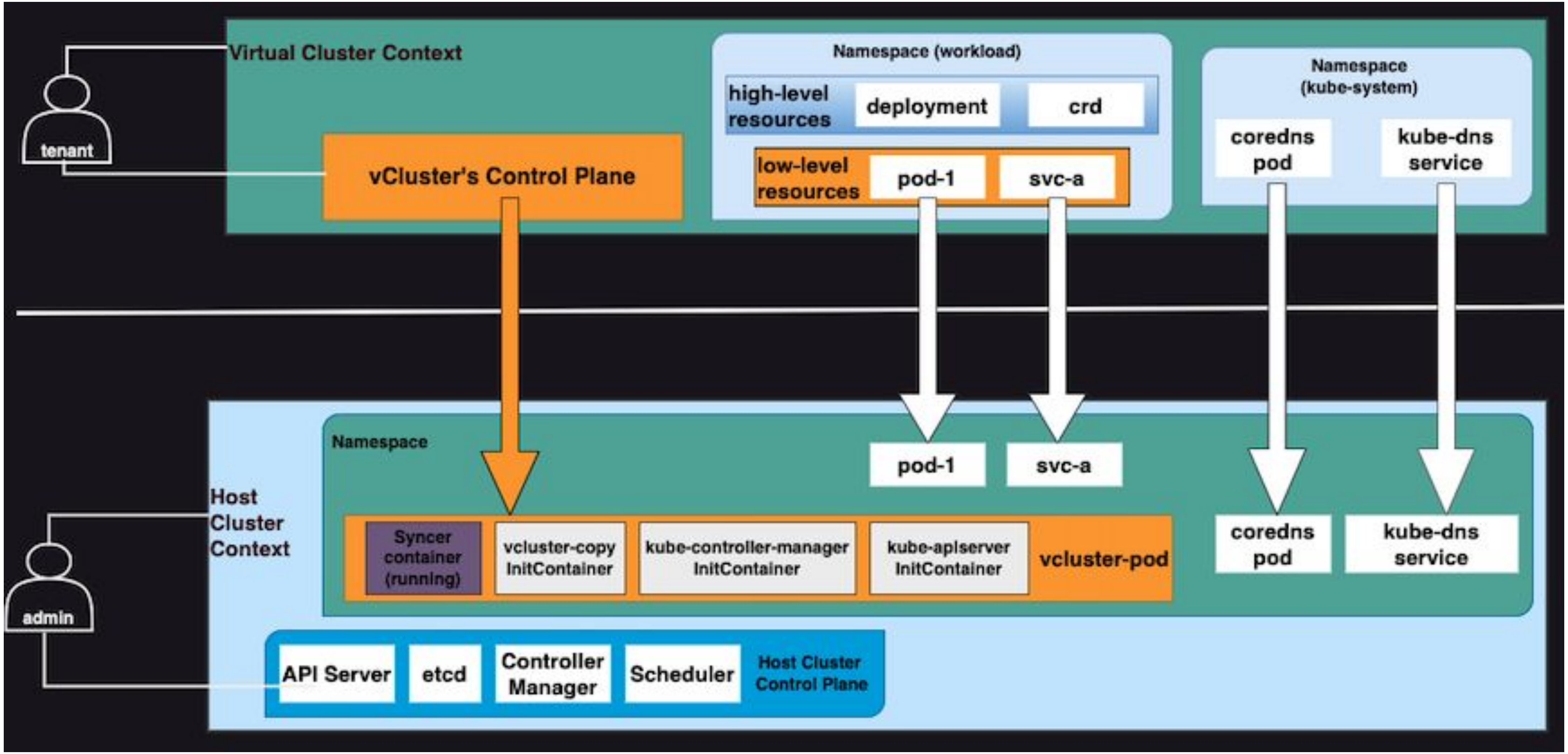


```
root@master:/home/test# kubectl config current-context
vcluster_my-vcluster_vcluster1_kubernetes-admin@kubernetes
root@master:/home/test# kubectl get deployments -n nginx1
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
nginx-vcluster1   2/2     2            2           11m
root@master:/home/test# kubectl create service clusterip nginx-vcluster1 --tcp=80:80 --namespace nginx1
service/nginx-vcluster1 created
root@master:/home/test# kubectl get services -n nginx1
NAME              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
nginx-vcluster1   ClusterIP   10.104.103.25    <none>        80/TCP    12s
```



```
root@master:/home/test# kubectl config current-context
kubernetes-admin@kubernetes
root@master:/home/test# kubectl get services -n vcluster1 -o wide | grep nginx
nginx-vcluster1-x-nginx1-x-my-vcluster   ClusterIP   10.104.103.25   <none>        80/TCP                      22m
72cedcae=nginx-vcluster1,vcluster.loft.sh/managed-by=my-vcluster,vcluster.loft.sh/namespace=nginx1
root@master:/home/test#
root@master:/home/test# kubectl exec -it curl-pod -n default -- curl http://10.104.103.25
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
```

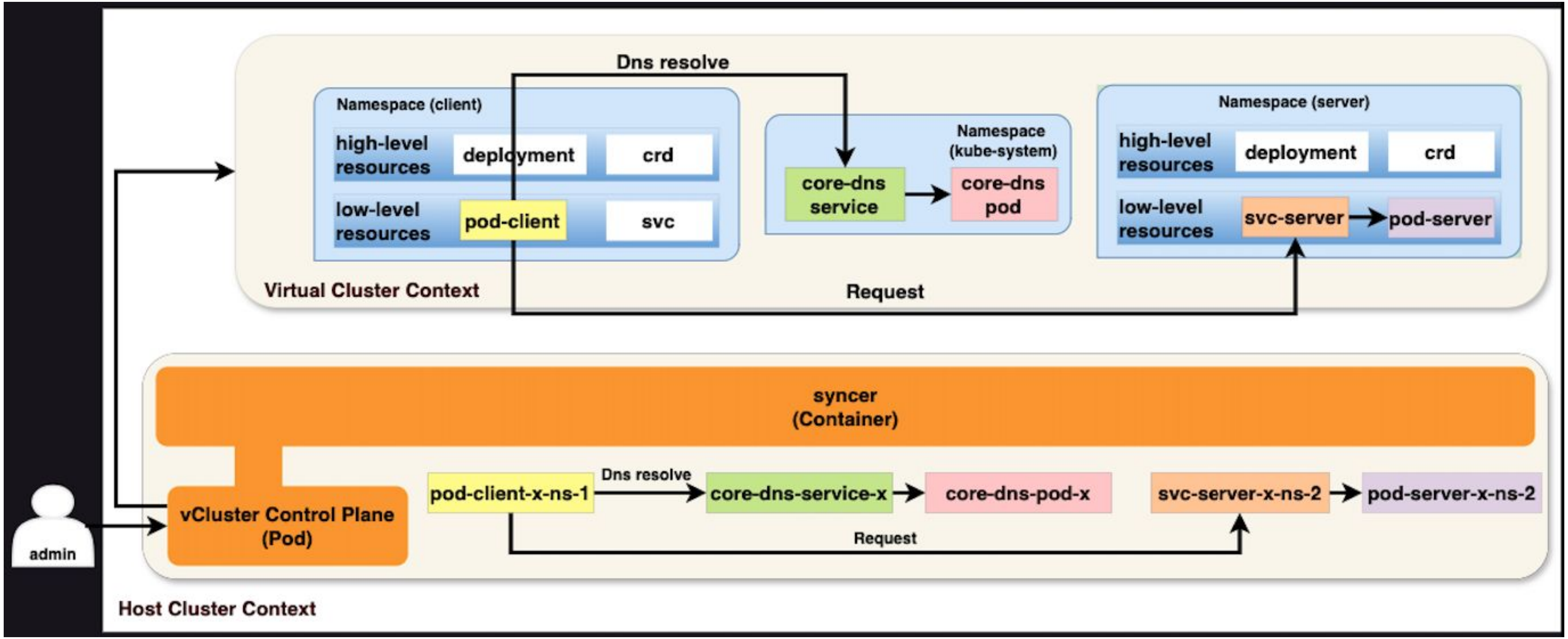
Summary


Virtual Cluster Context
tenant
vCluster's Control Plane
Namespace (workload)
high-level resources
deployment
crd
low-level resources
pod-1
svc-a
Namespace (kube-system)
coredns pod
kube-dns service
Host Cluster Context
admin
Namespace
pod-1
svc-a
Syncer container (running)
vcluster-copy InitContainer
kube-controller-manager InitContainer
kube-apiserver InitContainer
vcluster-pod
coredns pod
kube-dns service
API Server
etcd
Controller Manager
Scheduler
Host Cluster Control Plane

vCluster: Networking

- By default, each vCluster deploys its own individual DNS service, namely CoreDNS.
    - The DNS service lets pods within the virtual cluster resolve the IP addresses of other services running in the same virtual environment.
    - This capability is anchored by the syncer component, which maps service DNS names within the vCluster to their corresponding IP addresses in the host cluster, adhering to k8s’s DNS naming conventions.
- The vCluster will fallback to the host cluster's DNS for resolving domains if fallbackHostDNS is enabled.

Summary


Dns resolve
Namespace (client)
high-level resources
deployment
crd
low-level resources
pod-client
svc
Namespace (kube-system)
core-dns service
core-dns pod
Namespace (server)
high-level resources
deployment
crd
low-level resources
svc-server
pod-server
Virtual Cluster Context
Request
syncer (Container)
vCluster Control Plane (Pod)
admin
pod-client-x-ns-1
Dns resolve
core-dns-service-x
core-dns-pod-x
svc-server-x-ns-2
pod-server-x-ns-2
Request
Host Cluster Context

vCluster: Multi-Tenancy

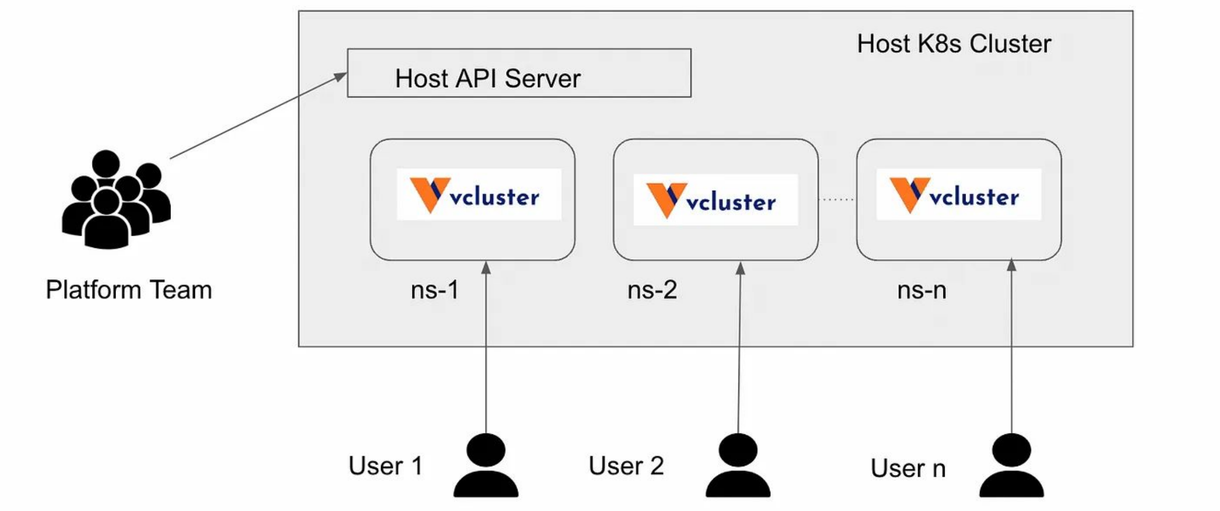
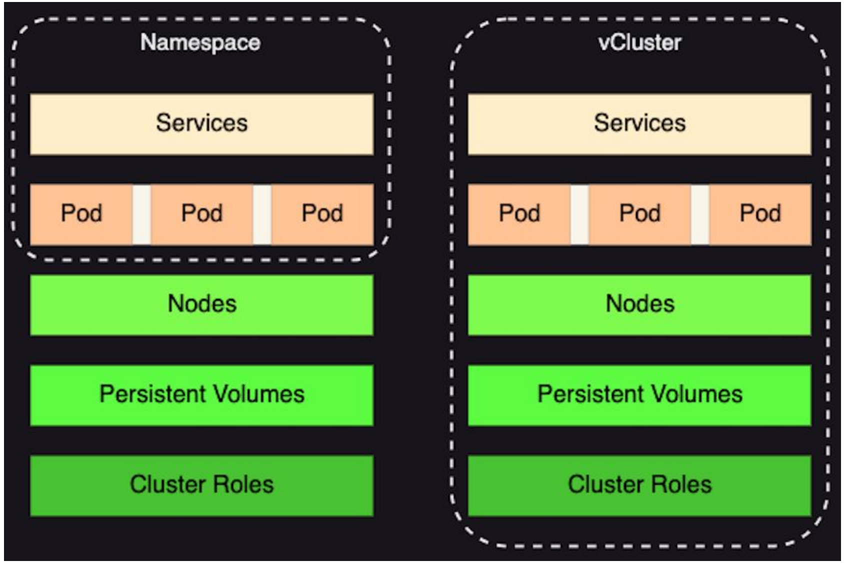


Figure: Multi-Tenancy with vCluster[2]

vCluster: Better Isolation



The **virtual control plane** in a vCluster replicates key Kubernetes components (API server, controller manager, etcd) within a host cluster's namespace.

This setup allows each virtual cluster to operate independently with its own resources (pods, services, deployments), isolated from other vClusters and the host cluster.

vCluster: Better Performance

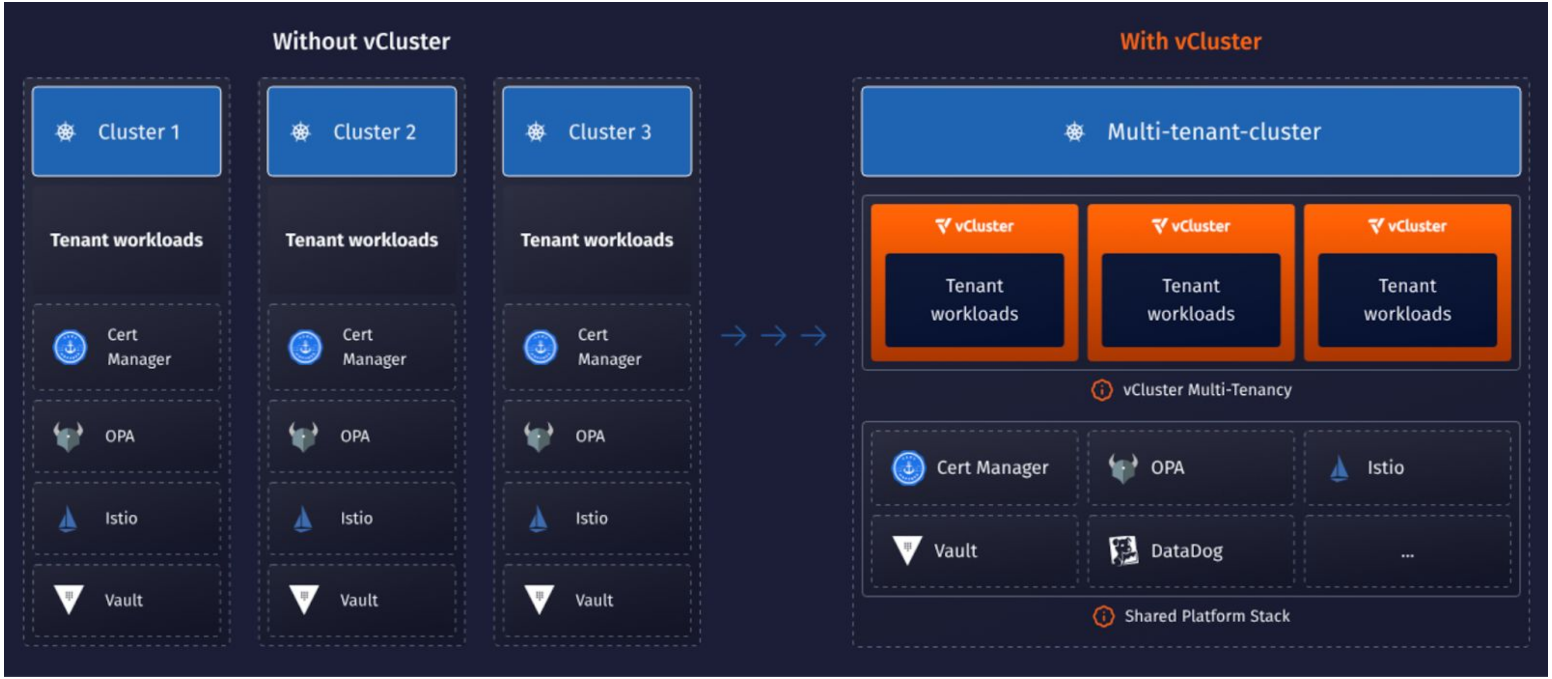
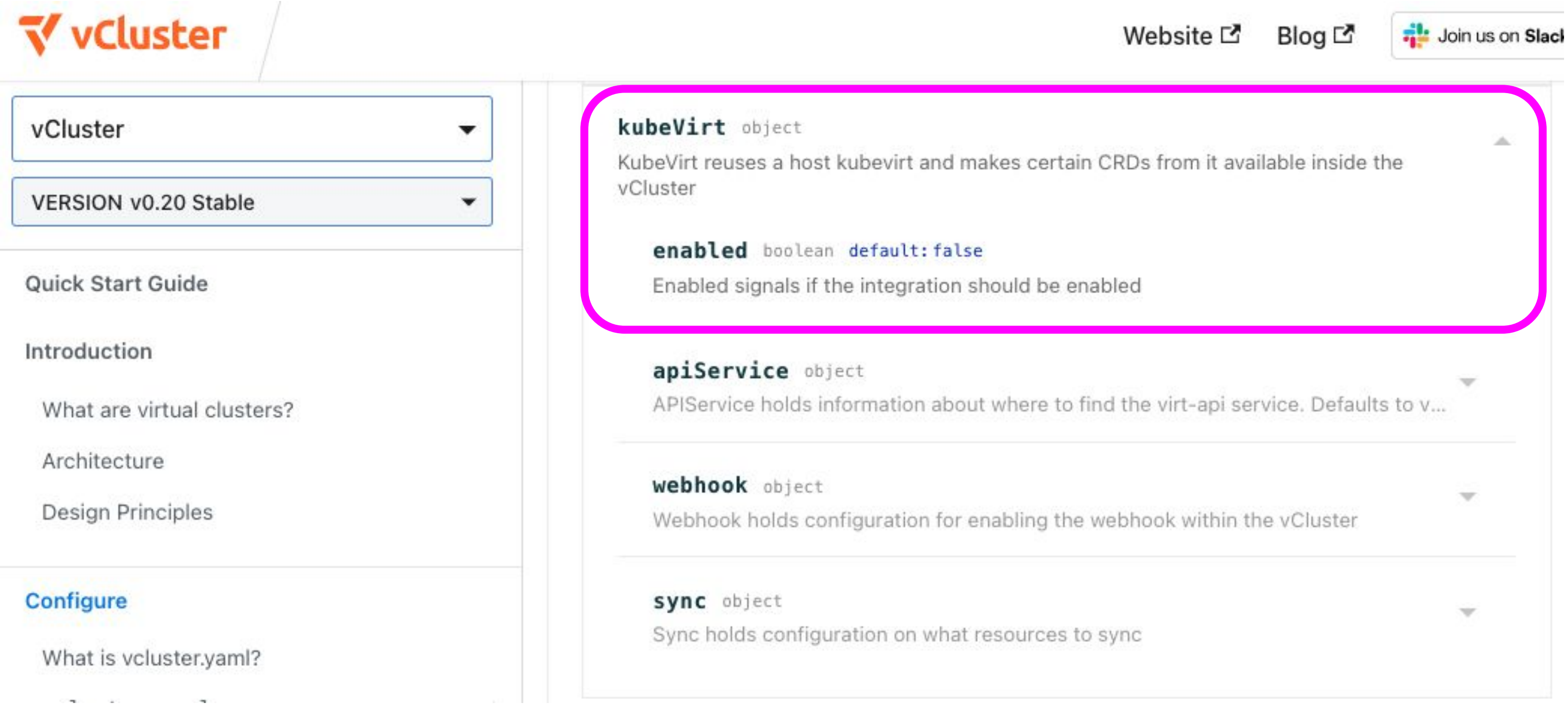


Figure: Without vCluster vs. With vCluster [3]

vCluster + KubeVirt VM (1)

https://github.com/loft-sh/vcluster/issues/2124



kubeVirt `object`
KubeVirt reuses a host kubevirt and makes certain CRDs from it available inside the vCluster

enabled `boolean` `default: false`
Enabled signals if the integration should be enabled

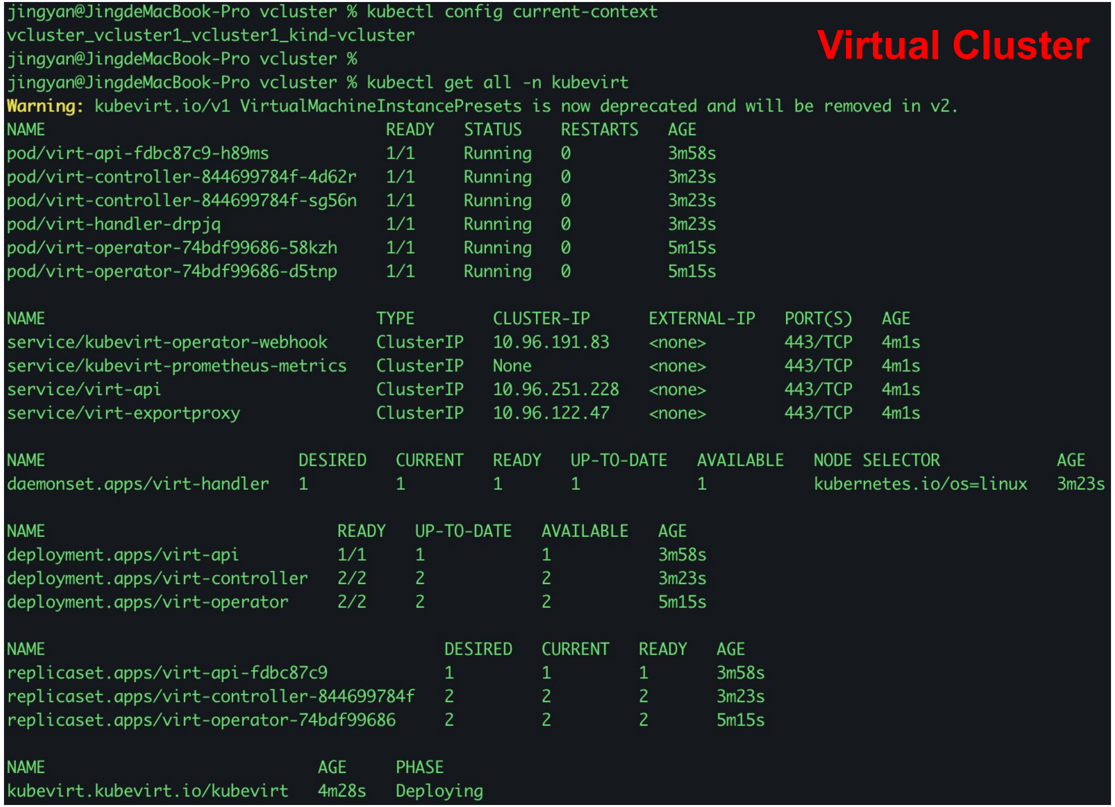
apiService `object`
APIService holds information about where to find the virt-api service. Defaults to v...

webhook `object`
Webhook holds configuration for enabling the webhook within the vCluster

sync `object`
Sync holds configuration on what resources to sync

vCluster + KubeVirt VM (2)

Deploy Kubevirt Operator and CRDs in vCluster

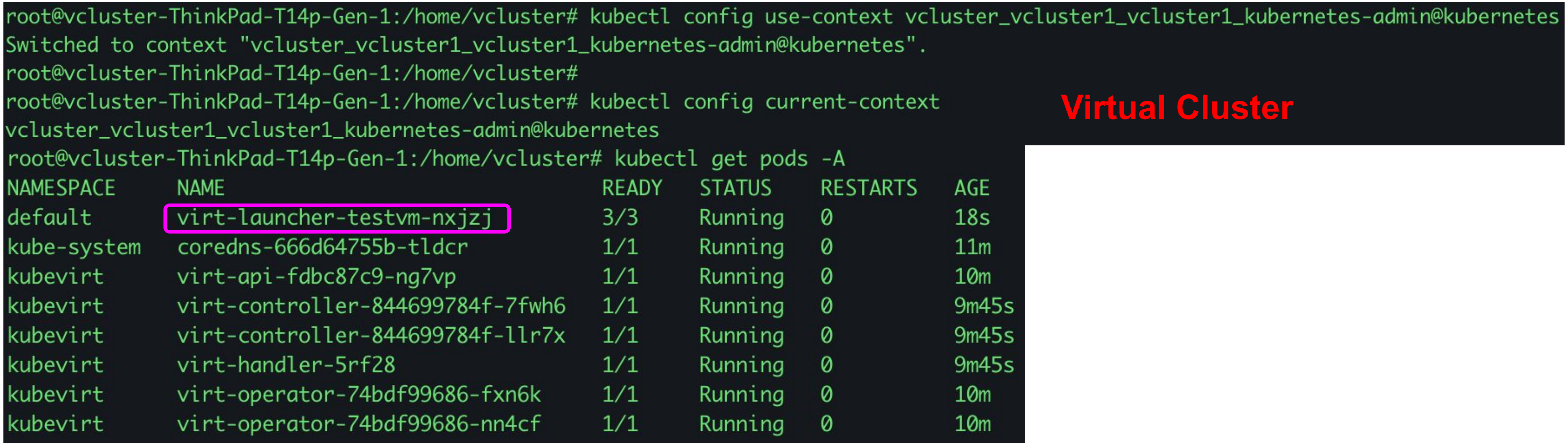


Virtual Cluster

```
jingyan@JingdeMacBook-Pro vcluster % kubectl config current-context
vcluster_vcluster1_vcluster1_kind-vcluster
jingyan@JingdeMacBook-Pro vcluster %
jingyan@JingdeMacBook-Pro vcluster % kubectl get all -n kubevirt
Warning: kubevirt.io/v1 VirtualMachineInstancePresets is now deprecated and will be removed in v2.
NAME                                   READY   STATUS    RESTARTS   AGE
pod/virt-api-fdbc87c9-h89ms            1/1     Running   0          3m58s
pod/virt-controller-844699784f-4d62r   1/1     Running   0          3m23s
pod/virt-controller-844699784f-sg56n   1/1     Running   0          3m23s
pod/virt-handler-drpjq                 1/1     Running   0          3m23s
pod/virt-operator-74bdf99686-58kzh     1/1     Running   0          5m15s
pod/virt-operator-74bdf99686-d5tnp     1/1     Running   0          5m15s

NAME                                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/kubevirt-operator-webhook     ClusterIP   10.96.191.83    <none>        443/TCP   4m1s
service/kubevirt-prometheus-metrics   ClusterIP   None            <none>        443/TCP   4m1s
service/virt-api                      ClusterIP   10.96.251.228   <none>        443/TCP   4m1s
service/virt-exportproxy              ClusterIP   10.96.122.47    <none>        443/TCP   4m1s

NAME                          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/virt-handler   1         1         1       1            1           kubernetes.io/os=linux   3m23s

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/virt-api          1/1     1            1           3m58s
deployment.apps/virt-controller   2/2     2            2           3m23s
deployment.apps/virt-operator     2/2     2            2           5m15s

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/virt-api-fdbc87c9            1         1         1       3m58s
replicaset.apps/virt-controller-844699784f   2         2         2       3m23s
replicaset.apps/virt-operator-74bdf99686     2         2         2       5m15s

NAME                            AGE     PHASE
kubevirt.kubevirt.io/kubevirt   4m28s   Deploying
```

vCluster + KubeVirt VM (3)

Deploy Kubevirt VM Pod in vCluster

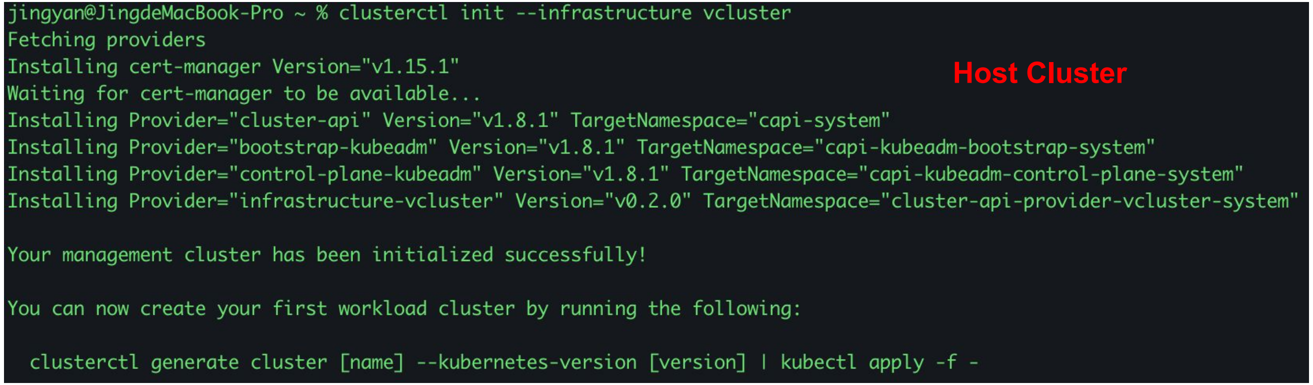


```
root@vcluster-ThinkPad-T14p-Gen-1:/home/vcluster# kubectl config use-context vcluster_vcluster1_vcluster1_kubernetes-admin@kubernetes
Switched to context "vcluster_vcluster1_vcluster1_kubernetes-admin@kubernetes".
root@vcluster-ThinkPad-T14p-Gen-1:/home/vcluster#
root@vcluster-ThinkPad-T14p-Gen-1:/home/vcluster# kubectl config current-context
vcluster_vcluster1_vcluster1_kubernetes-admin@kubernetes
root@vcluster-ThinkPad-T14p-Gen-1:/home/vcluster# kubectl get pods -A
NAMESPACE     NAME                              READY   STATUS    RESTARTS   AGE
default       virt-launcher-testvm-nxjzj        3/3     Running   0          18s
kube-system   coredns-666d64755b-tldcr          1/1     Running   0          11m
kubevirt      virt-api-fdbc87c9-ng7vp           1/1     Running   0          10m
kubevirt      virt-controller-844699784f-7fwh6  1/1     Running   0          9m45s
kubevirt      virt-controller-844699784f-llr7x  1/1     Running   0          9m45s
kubevirt      virt-handler-5rf28                1/1     Running   0          9m45s
kubevirt      virt-operator-74bdf99686-fxn6k    1/1     Running   0          10m
kubevirt      virt-operator-74bdf99686-nn4cf    1/1     Running   0          10m
```

Virtual Cluster

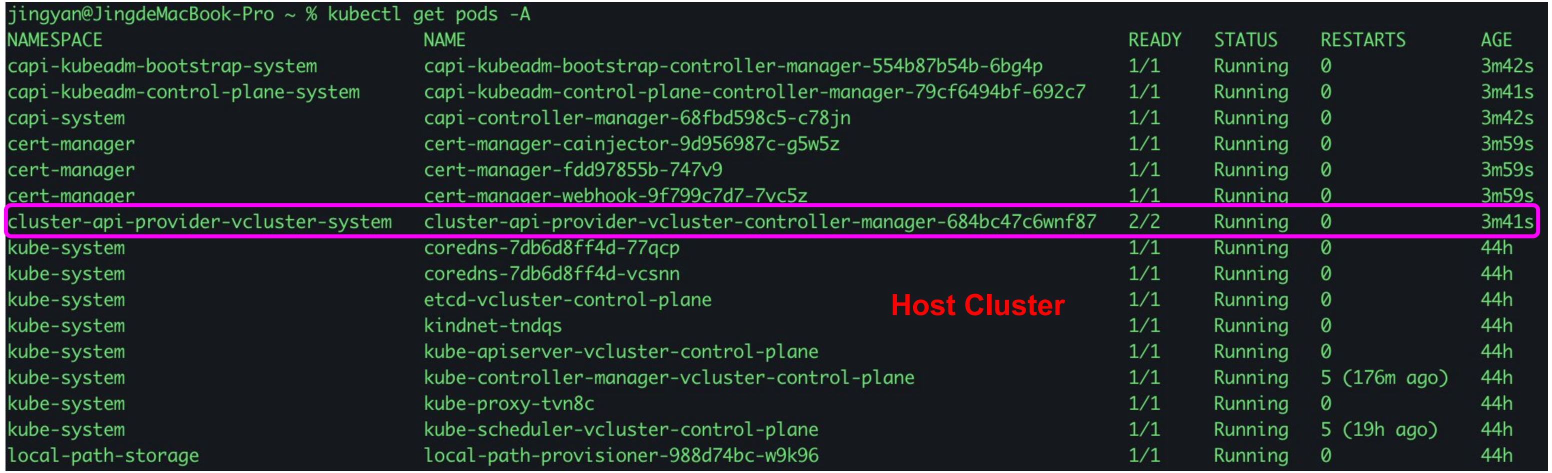
vCluster + Cluster API (1)

cluster-api-provider-vcluster: https://github.com/loft-sh/cluster-api-provider-vcluster



```
jingyan@JingdeMacBook-Pro ~ % clusterctl init --infrastructure vcluster
Fetching providers
Installing cert-manager Version="v1.15.1"
Waiting for cert-manager to be available...
Installing Provider="cluster-api" Version="v1.8.1" TargetNamespace="capi-system"
Installing Provider="bootstrap-kubeadm" Version="v1.8.1" TargetNamespace="capi-kubeadm-bootstrap-system"
Installing Provider="control-plane-kubeadm" Version="v1.8.1" TargetNamespace="capi-kubeadm-control-plane-system"
Installing Provider="infrastructure-vcluster" Version="v0.2.0" TargetNamespace="cluster-api-provider-vcluster-system"

Your management cluster has been initialized successfully!

You can now create your first workload cluster by running the following:

  clusterctl generate cluster [name] --kubernetes-version [version] | kubectl apply -f -
```

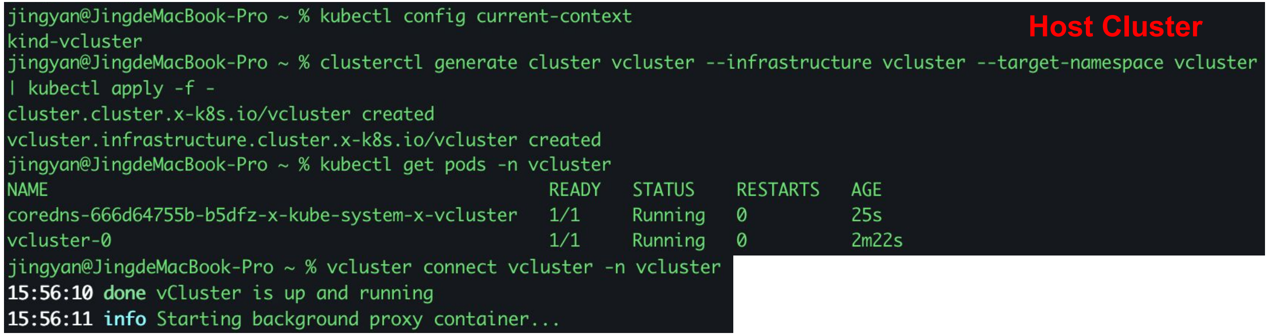
Init management k8s cluster and deploy cluster-api-provider-vcluster



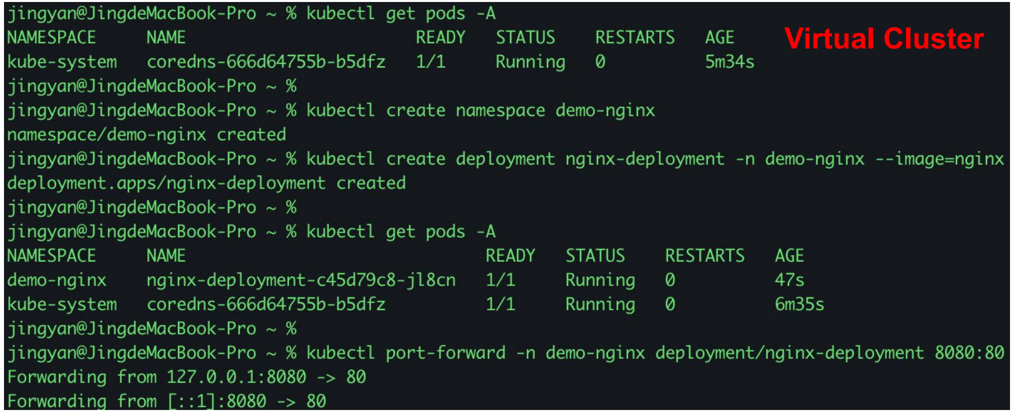
```
jingyan@JingdeMacBook-Pro ~ % kubectl get pods -A
NAMESPACE                              NAME                                                              READY   STATUS    RESTARTS       AGE
capi-kubeadm-bootstrap-system          capi-kubeadm-bootstrap-controller-manager-554b87b54b-6bg4p        1/1     Running   0              3m42s
capi-kubeadm-control-plane-system      capi-kubeadm-control-plane-controller-manager-79cf6494bf-692c7    1/1     Running   0              3m41s
capi-system                            capi-controller-manager-68fbd598c5-c78jn                          1/1     Running   0              3m42s
cert-manager                           cert-manager-cainjector-9d956987c-g5w5z                           1/1     Running   0              3m59s
cert-manager                           cert-manager-fdd97855b-747v9                                      1/1     Running   0              3m59s
cert-manager                           cert-manager-webhook-9f799c7d7-7vc5z                              1/1     Running   0              3m59s
cluster-api-provider-vcluster-system   cluster-api-provider-vcluster-controller-manager-684bc47c6wnf87   2/2     Running   0              3m41s
kube-system                            coredns-7db6d8ff4d-77qcp                                          1/1     Running   0              44h
kube-system                            coredns-7db6d8ff4d-vcsnn                                          1/1     Running   0              44h
kube-system                            etcd-vcluster-control-plane                                       1/1     Running   0              44h
kube-system                            kindnet-tndqs                                                     1/1     Running   0              44h
kube-system                            kube-apiserver-vcluster-control-plane                             1/1     Running   0              44h
kube-system                            kube-controller-manager-vcluster-control-plane                    1/1     Running   5 (176m ago)   44h
kube-system                            kube-proxy-tvn8c                                                  1/1     Running   0              44h
kube-system                            kube-scheduler-vcluster-control-plane                             1/1     Running   5 (19h ago)    44h
local-path-storage                     local-path-provisioner-988d74bc-w9k96                             1/1     Running   0              44h
```

vCluster + Cluster API (2)

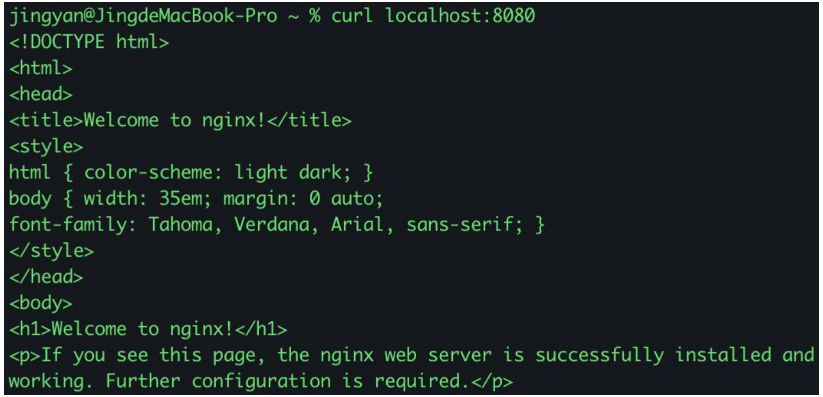
- Create a target vCluster via Cluster API, Connect to the target vCluster
- Deploy service in target vCluster and access it



```
jingyan@JingdeMacBook-Pro ~ % kubectl config current-context
kind-vcluster
jingyan@JingdeMacBook-Pro ~ % clusterctl generate cluster vcluster --infrastructure vcluster --target-namespace vcluster
| kubectl apply -f -
cluster.cluster.x-k8s.io/vcluster created
vcluster.infrastructure.cluster.x-k8s.io/vcluster created
jingyan@JingdeMacBook-Pro ~ % kubectl get pods -n vcluster
NAME                                                READY   STATUS    RESTARTS   AGE
coredns-666d64755b-b5dfz-x-kube-system-x-vcluster   1/1     Running   0          25s
vcluster-0                                          1/1     Running   0          2m22s
jingyan@JingdeMacBook-Pro ~ % vcluster connect vcluster -n vcluster
15:56:10 done vCluster is up and running
15:56:11 info Starting background proxy container...
```



```
jingyan@JingdeMacBook-Pro ~ % kubectl get pods -A
NAMESPACE     NAME                       READY   STATUS    RESTARTS   AGE
kube-system   coredns-666d64755b-b5dfz   1/1     Running   0          5m34s
jingyan@JingdeMacBook-Pro ~ %
jingyan@JingdeMacBook-Pro ~ % kubectl create namespace demo-nginx
namespace/demo-nginx created
jingyan@JingdeMacBook-Pro ~ % kubectl create deployment nginx-deployment -n demo-nginx --image=nginx
deployment.apps/nginx-deployment created
jingyan@JingdeMacBook-Pro ~ %
jingyan@JingdeMacBook-Pro ~ % kubectl get pods -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
demo-nginx    nginx-deployment-c45d79c8-jl8cn    1/1     Running   0          47s
kube-system   coredns-666d64755b-b5dfz           1/1     Running   0          6m35s
jingyan@JingdeMacBook-Pro ~ %
jingyan@JingdeMacBook-Pro ~ % kubectl port-forward -n demo-nginx deployment/nginx-deployment 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```



```
jingyan@JingdeMacBook-Pro ~ % curl localhost:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
```

References

[1] https://www.vcluster.com/docs/v0.19/what-are-virtual-clusters
[2] https://github.com/loft-sh/vcluster
[3] https://www.vcluster.com/