



KubeCon



CloudNativeCon

North America 2021

Beyond Namespaces: Virtual Clusters are the Future of Multi-Tenancy

KubeCon NA 2021, Los Angeles

Lukas Gentele
CEO @ Loft Labs, Inc.
lg@loft.sh



Hi! I'm Lukas.



- ▷ CEO @ Loft Labs, Inc.
- ▷ Based in SF but made in Germany

Product



Open Source



DevSpace



vcluster



kiosk



jsolicy



@LukasGentele

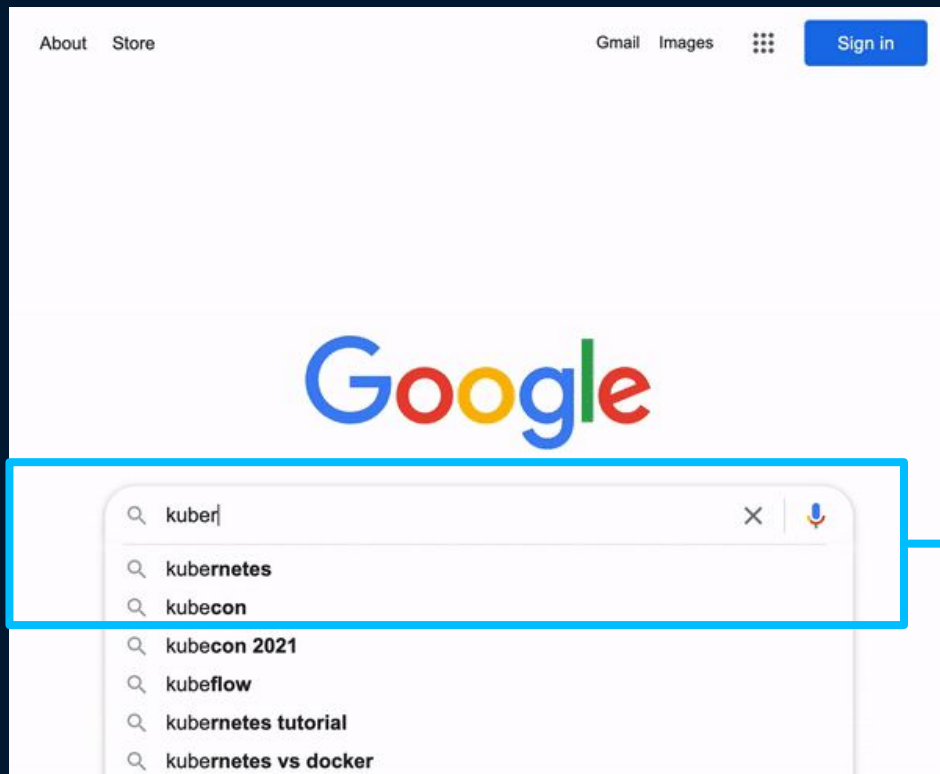
I know this.

?????

Beyond Namespaces: Virtual Clusters
are the Future of Multi-Tenancy



What Are **Kubernetes Namespaces** ?



Namespaces = Virtual Clusters????

Namespaces \neq (Virtual) Clusters



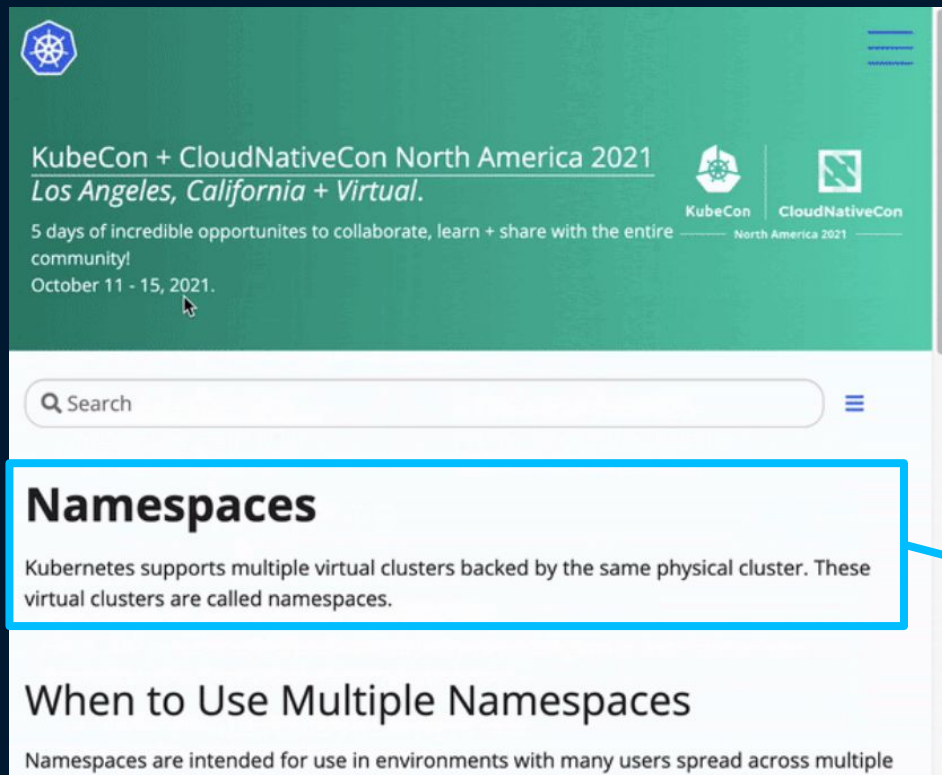
“Namespaces provide a **scope for names**.
Names of resources need to be unique within
a namespace, but not across namespaces.”

```
$ kubectl api-resources --namespaced=true
```

“Not all objects are in a namespace”
= **Cluster-wide resources** such as nodes,
persistent volumes, and namespaces

```
$ kubectl api-resources --namespaced=false
```

Hands up if you are a Kubernetes contributor!



KubeCon + CloudNativeCon North America 2021
Los Angeles, California + Virtual.

5 days of incredible opportunities to collaborate, learn + share with the entire community!
October 11 - 15, 2021.

Search

Namespaces

Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called namespaces.

When to Use Multiple Namespaces

Namespaces are intended for use in environments with many users spread across multiple



The person who fixes this docs page gets a surprise present from me!

Multi-Tenancy? Why Should I Care?

▷ Spinning up thousands of single-tenant clusters is a terrible idea:

- ▶ Too costly
- ▶ Too hard to manage

▷ Use Namespaces instead?


▷ Multi-Tenant Clusters?

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

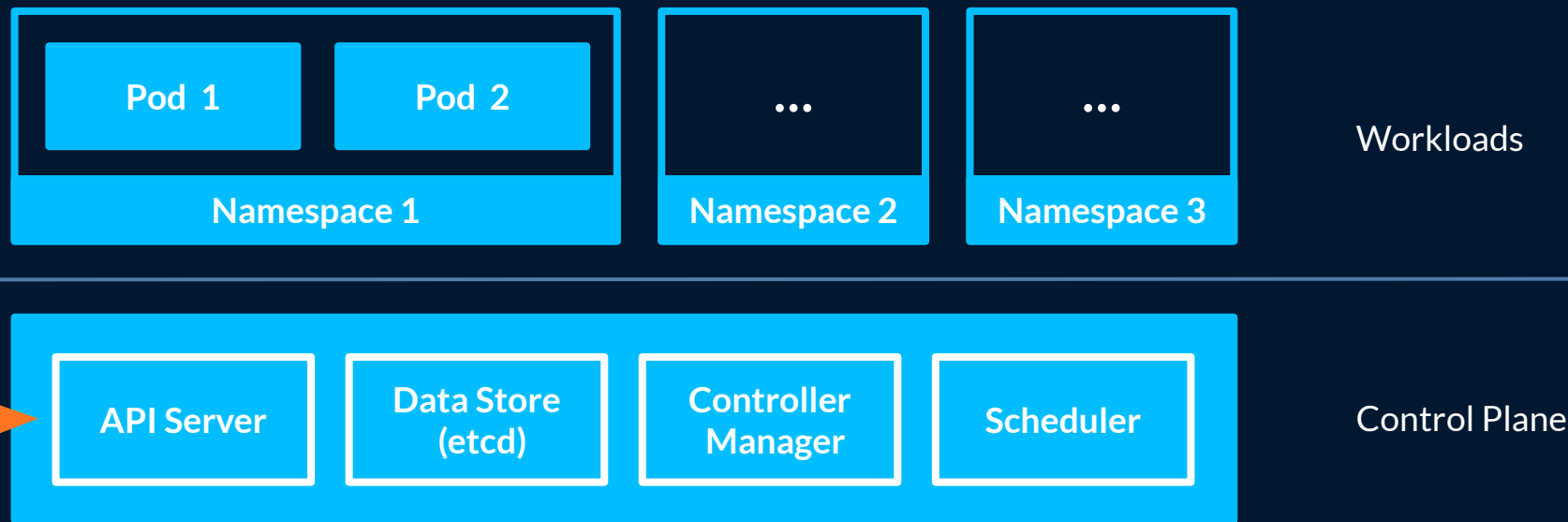
Can't we get the best of both worlds?

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

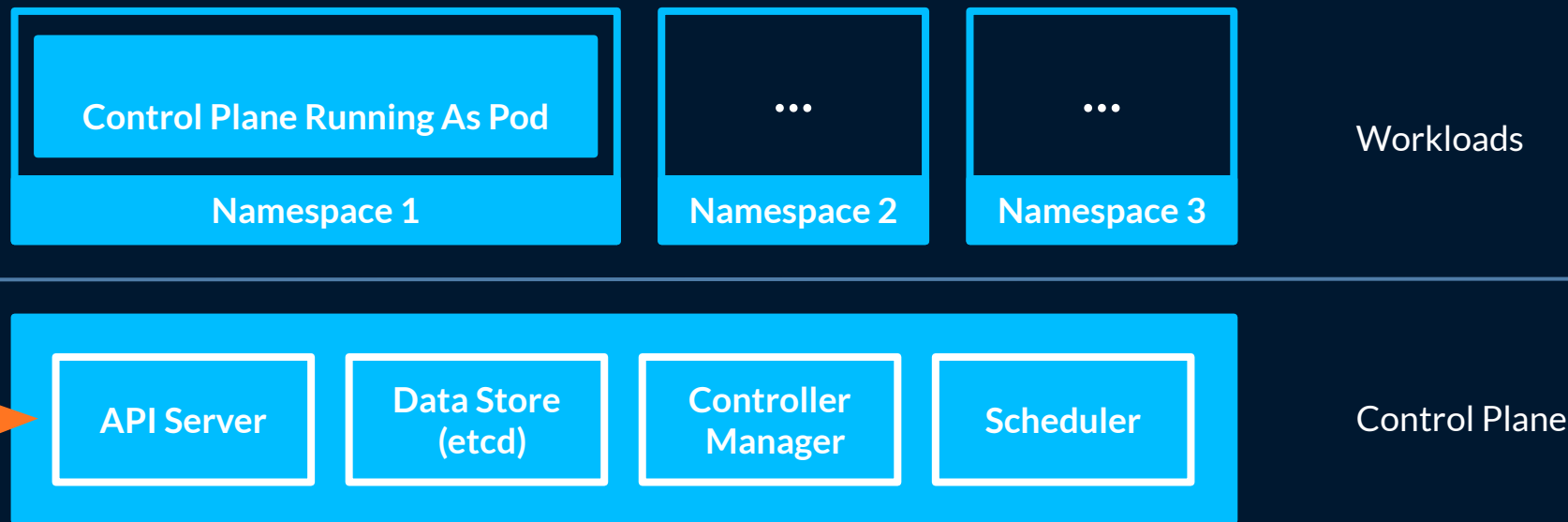
Virtual Clusters For Kubernetes Multi-Tenancy

	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

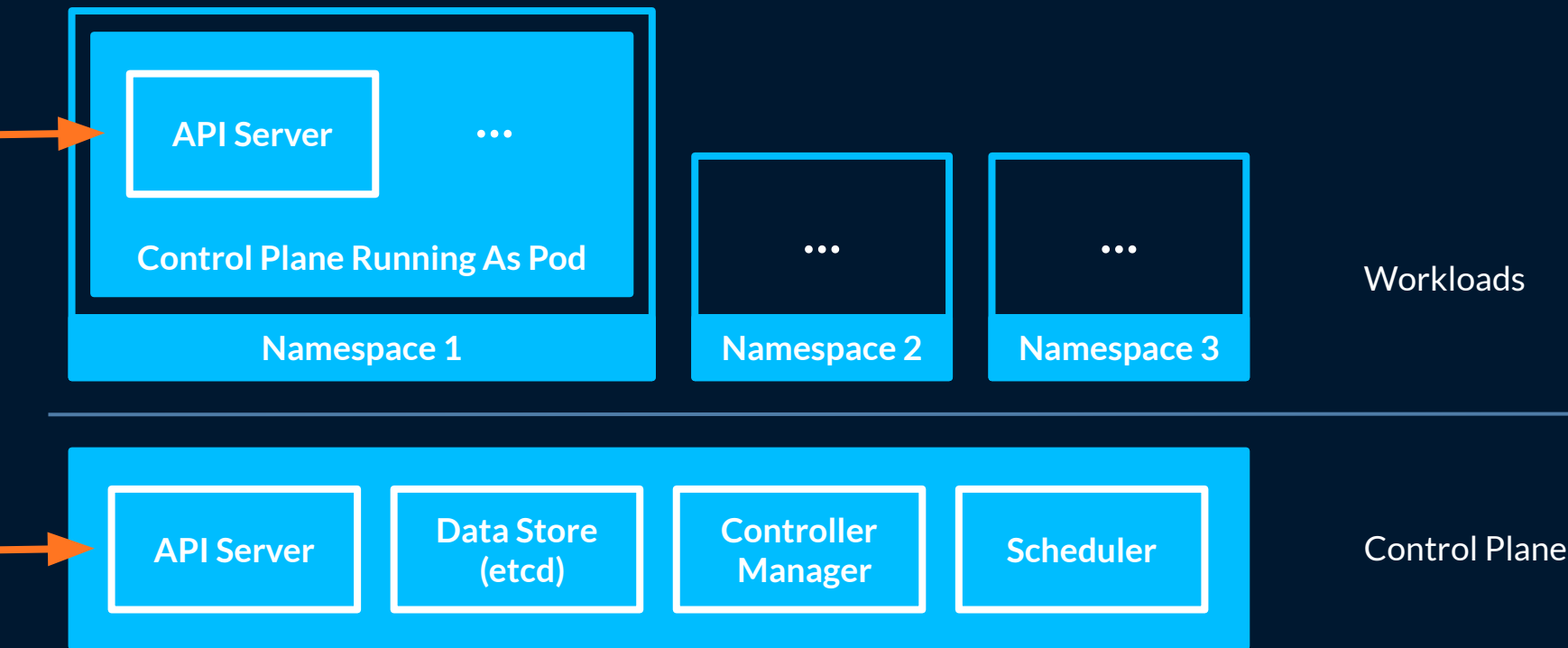
Virtual Clusters = Control Planes Running In Another Cluster



Virtual Clusters = Control Planes Running In Another Cluster



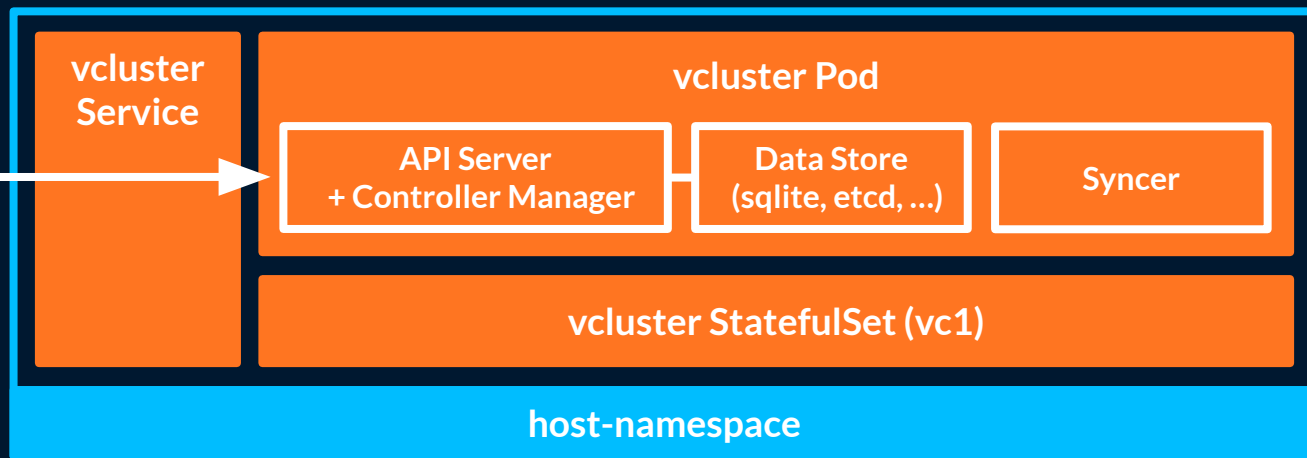
Virtual Clusters = Control Planes Running In Another Cluster



Demo Virtual Clusters with vcluster

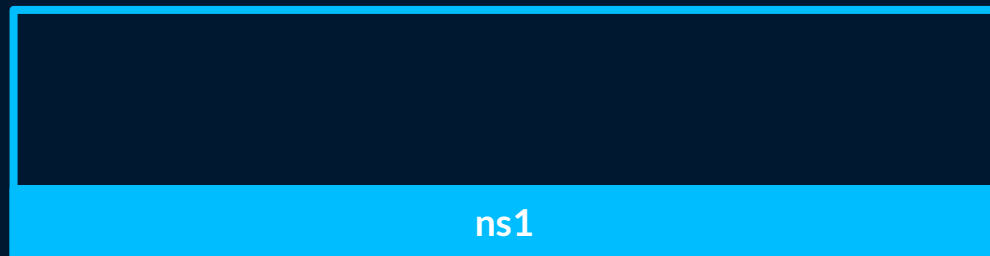


Architecture Virtual Clusters with vcluster

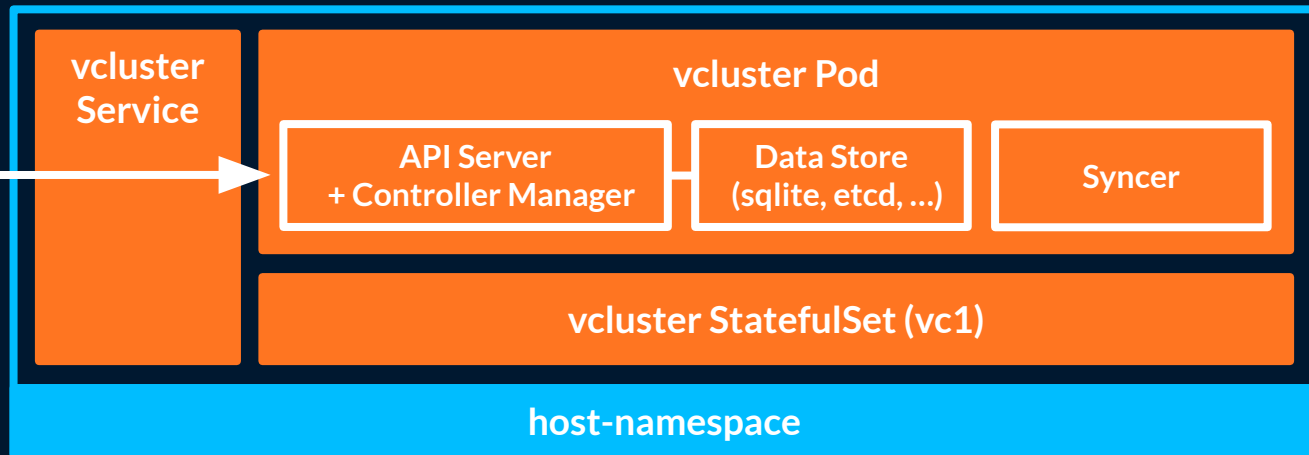


Control Plane
Of Host Cluster

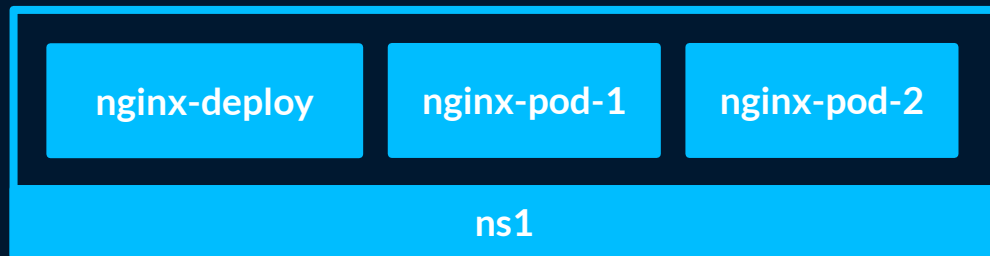
vcluster Creating Namespaces



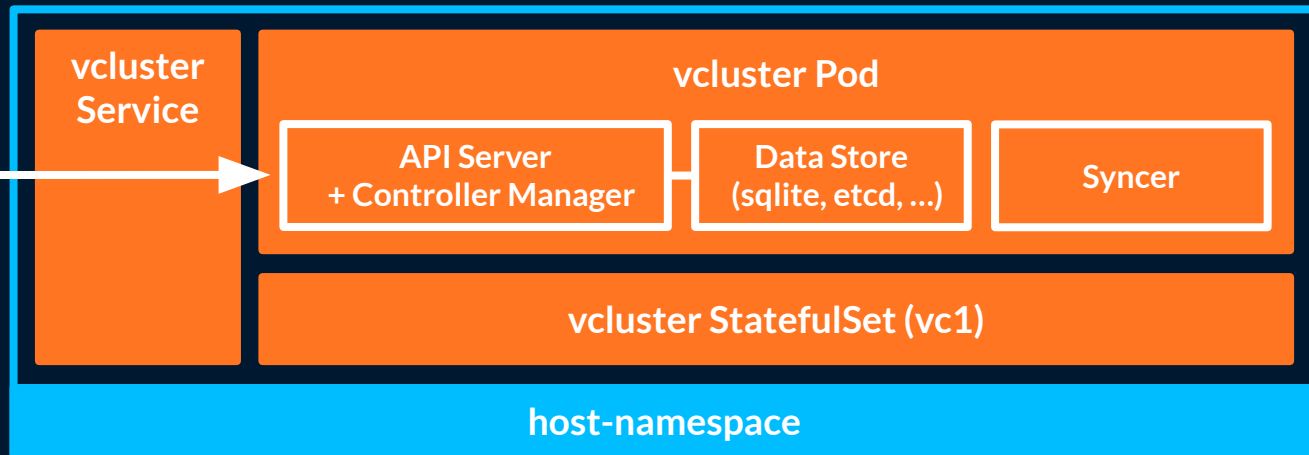
```
$ kubectl create namespace ns1
```



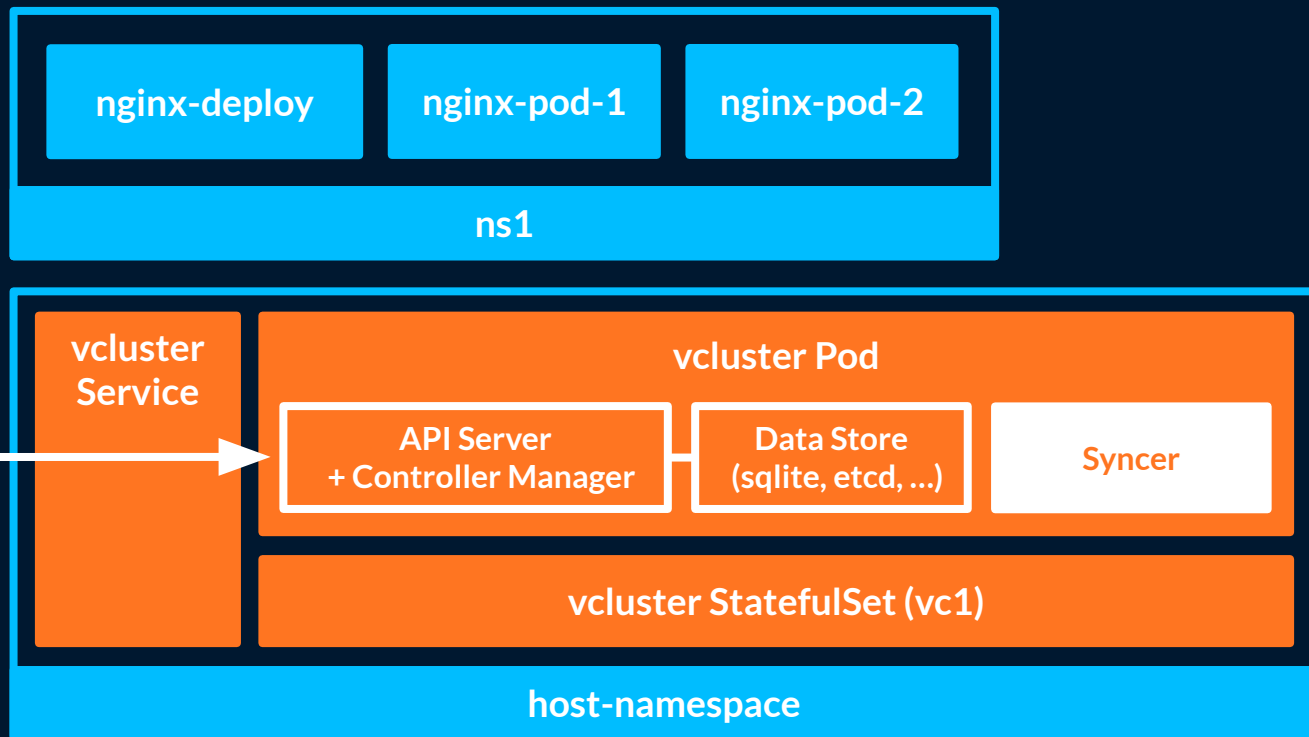
vcluster Creating Deployments



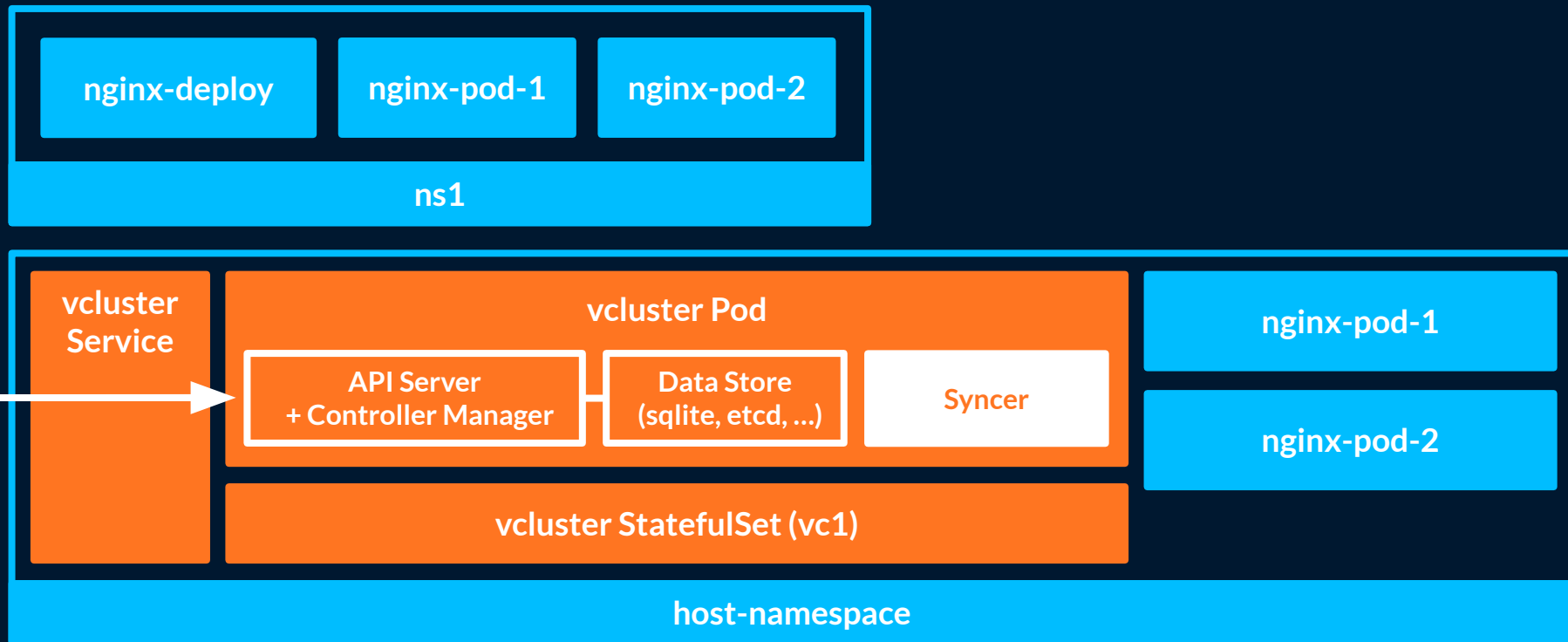
```
$ kubectl create deployment nginx-deploy  
-n ns1 --image=nginx --replicas=2
```



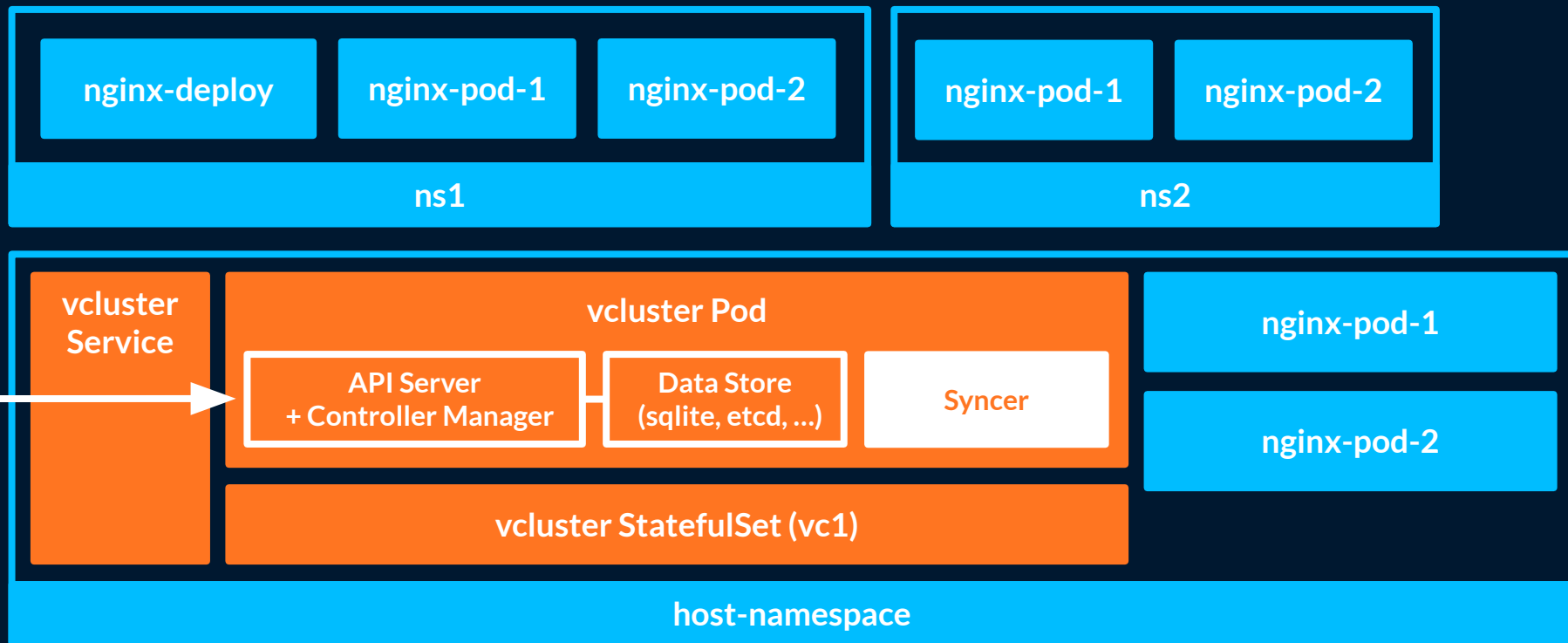
vcluster Syncer syncs Pods To Host Namespace / “Real” Cluster



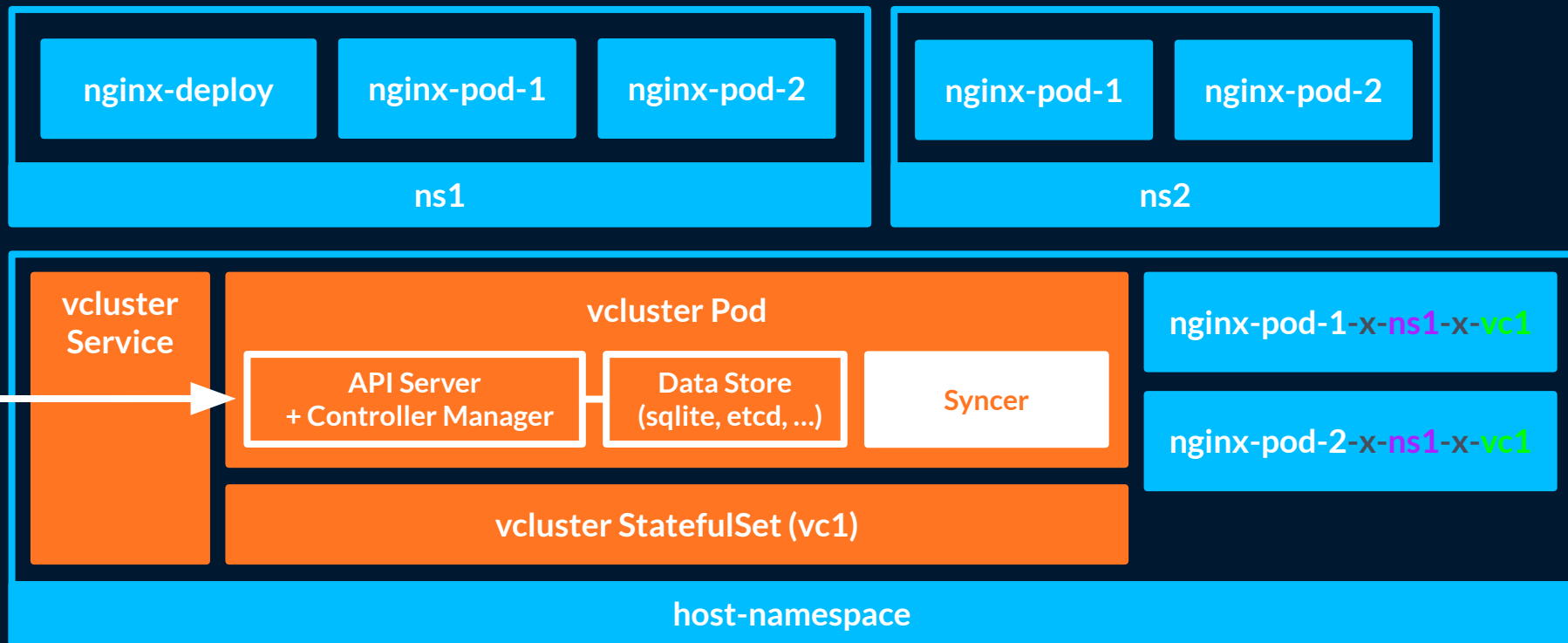
vcluster Syncer syncs Pods To Host Namespace / “Real” Cluster



vcluster What About Naming Conflicts?



vcluster Syncer Rewrites Names To Prevent Collisions



vcluster Syncer Only Syncs What Pods Need To Run

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 Secrets, ConfigMaps
- ▷ Other: Service Accounts, Jobs, etc.
- ▷ Custom Resources (+ CRDs)

The vast majority of objects will only exist in the vcluster

vcluster Pod Networking & DNS

- ▷ Pod-To-Pod Communication
 - ▶ Works out-of-the-box since all pods actually run in the underlying cluster
- ▷ Pod-To-Service Communication
 - ▶ Services are synchronized down to the host cluster
- ▷ (v)Cluster-Internal DNS
 - ▶ Syncer adds `dnsConfig` to pod spec and makes sure pods connect to the separate CoreDNS service started in the vcluster
- ▷ Ingresses sync is enabled by default to allow to use of the underlying cluster's ingress controller. This can also be disabled using a flag for the syncer container.

- ▷ It depends.
 - ▶ Fake Nodes (default, dynamic)
 - ▶ Real Nodes (dynamic)
 - ▶ All Real Nodes
 - ▶ Real Nodes with Label Selector (also enforceable for pod scheduling)
- ▷ Configurable via flags for the syncer pod

vcluster What Else Is Possible / Will Be Possible?

- ▶ Running `$ kubectl` inside a vcluster pod
 - ▶ Syncer sets env var KUBERNETES_SERVICE_HOST for each pod
 - ▶ In-cluster kube-config will point to the API server of the respective vcluster
- ▶ Exposing vcluster API server via ingress, LB, etc.
- ▶ Sync for Network Policies (upcoming)
- ▶ Pod Disruption Budgets (upcoming)
- ▶ Non-Root vclusters (upcoming)
- ▶ Nested virtual clusters



Rich Burroughs @ KubeCon
@richburroughs

I made a YouTube video about how to create a virtual Kubernetes cluster inside of a virtual Kubernetes cluster with vcluster. It's some pretty fun Kubernetes inception :)



youtube.com

vcluster Inception

vcluster is open source software that allows you to create virtual Kubernetes clusters quickly and easily. In this video ...



@LukasGentele

Use Cases For Virtual Clusters

- ▷ Ephemeral CI/CD Environments
 - ▶ Integration, End-To-End, Acceptance Testing
 - ▶ Instant Preview Environments Per Pull-Request
- ▷ Remote Development Environments
- ▷ Experimentation (ML/AI Model Creation)
- ▷ Cluster Simulations
- ▷ More Resilient Multi-Tenant Clusters In Production
- ▷ Sales Demos

vcluster How To Get Started?

- ▷ Download CLI && `$ vcluster create my-vc1 && vcluster connect my-vc1`
- ▷ Getting Started: <https://www.vcluster.com/docs/quickstart>
- ▷ GitHub: <https://github.com/loft-sh/vcluster>
- ▷ Website: www.vcluster.com
- ▷ Twitter: https://twitter.com/loft_sh

Questions? Reach out!

- ▷ Questions? Ask now!
- ▷ Join us on Slack: <https://slack.loft.sh>
- ▷ Reach out via Twitter: @LukasGentele || @loft_sh
- ▷ >>> **Booth SU49** <<< (near CNCF project maintainer area)
 - ▶ Swag: T-Shirts, Stickers, Fridge Magnets etc.
 - ▶ KubeCon Postcards for your loved ones who cannot be here in person



We are hiring!
www.loft.sh/careers



Fabian Kramm
CTO @ Loft Labs