



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# Intro: Kubernetes VMware User Group

Best Practices for Running on VMware

Myles Gray

Senior Technical Marketing  
Architect, VMware

Steven Wong

Software Engineer, VMware

August 18, 2020 – 21:30 Central European Summer Time

# Agenda

Intro to the vSphere cloud provider and related storage plugins

Recent features/changes: What are they?  
What are they? How to use them?

Running Kubernetes on "desktop" hypervisors  
Kubernetes on Fusion and Workstation

How to get involved in the User Group



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# The vSphere Cloud Provider

## What does it do?

Cloud providers are what makes Kubernetes “cloud native”

- plug-in abstraction layer that links to underlying infrastructure in a public cloud or on-prem

The vSphere Cloud Provider:

- Supports reporting the availability zones of underlying infrastructure tying this in with the CSI storage plugin
- Does not support a specific load balancer, routes, or interface to return a cluster list

For more details see: [cloud-provider-vsphere.sigs.k8s.io/](https://cloud-provider-vsphere.sigs.k8s.io/)



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# Intro to storage on vSphere

## The VCP, CPI and CSI



KubeCon



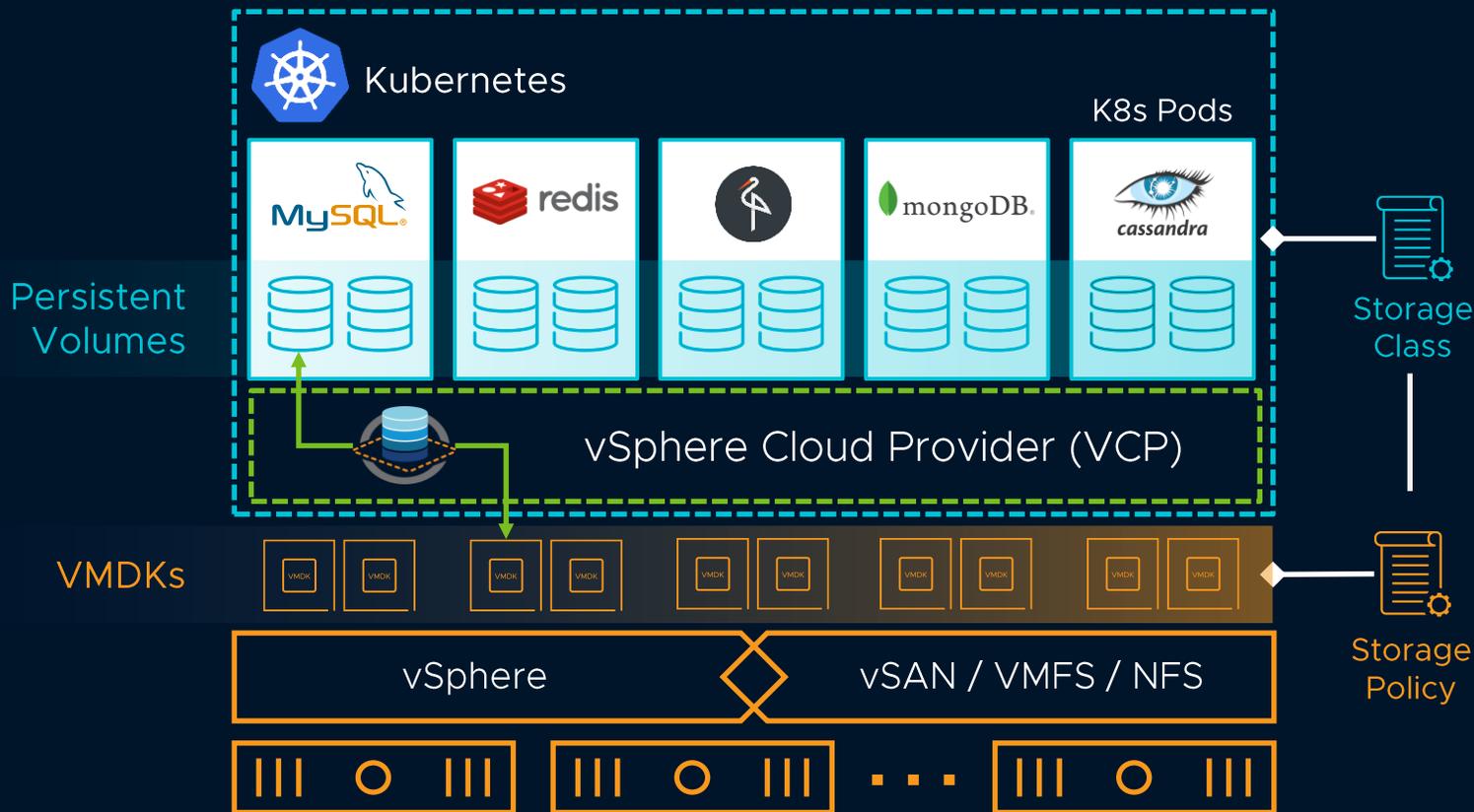
CloudNativeCon

Europe 2020

*Virtual*

# vSphere Cloud Provider (VCP) for Kubernetes

The past



Natively built into Kubernetes

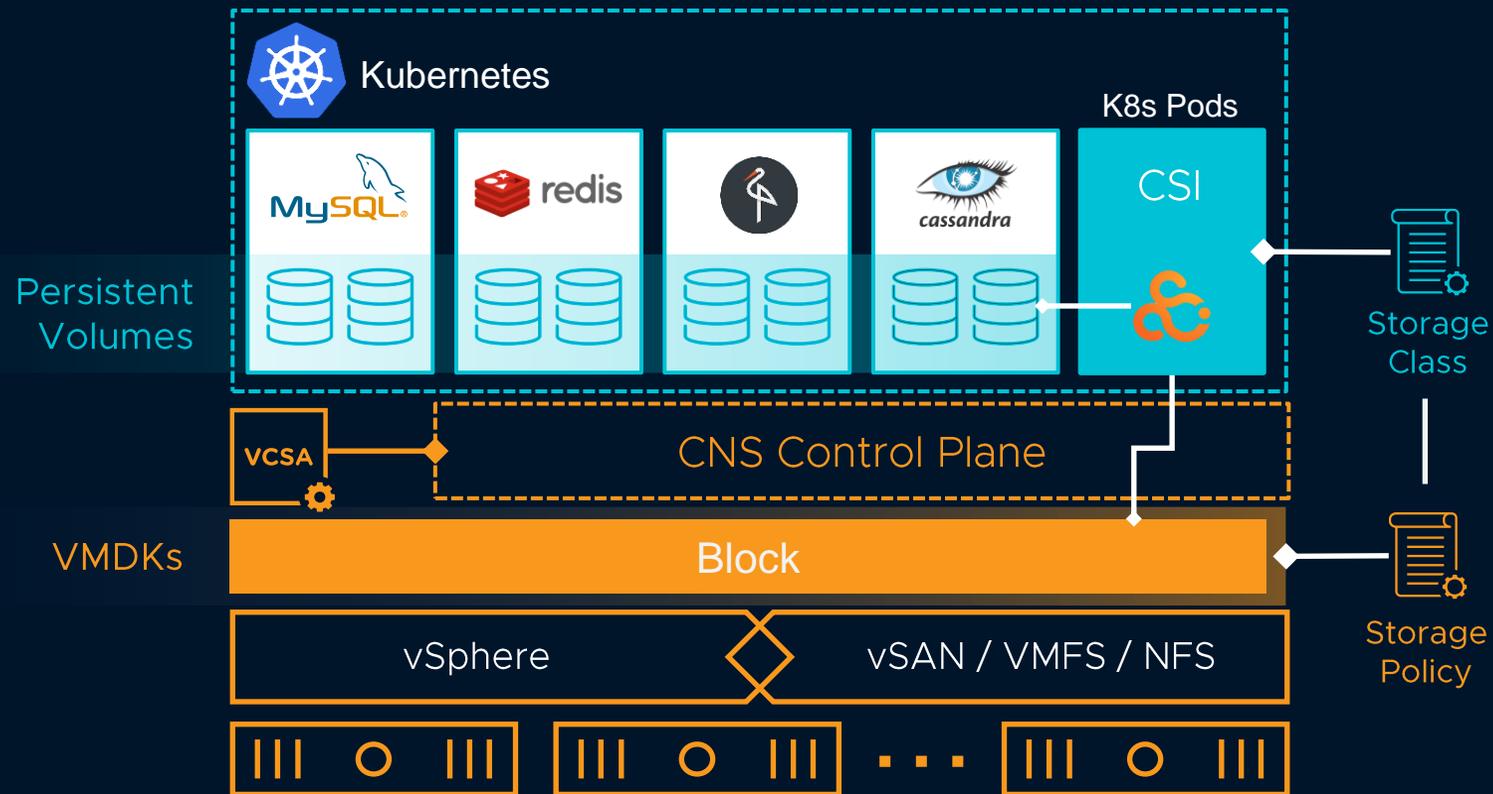
Policy driven dynamic provisioning of Kubernetes persistent volumes

Data services at a granularity of a container volume via SPBM

Not without its drawbacks

# vSphere 6.7 Update 3 – CNS Platform Introduction

The present



Built on the **CSI standard** for container storage

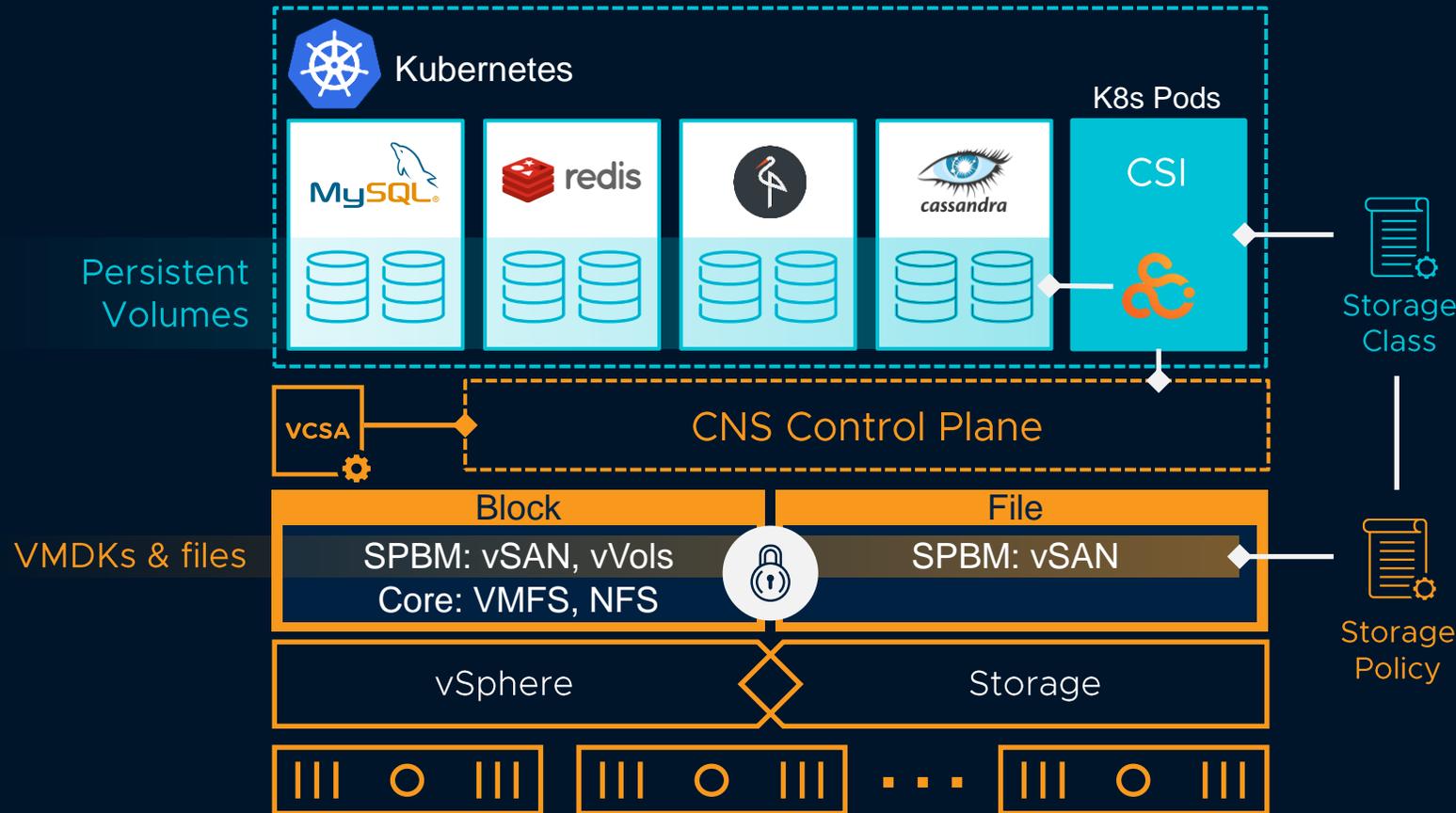
**Policy driven dynamic provisioning** of Kubernetes persistent volumes

Enabling **operational consistency** between VM and container infrastructure management

Abstracts the storage **infrastructure for developers**

# Continued Integration of Cloud Native Storage in vSphere and vSAN

## The present



Offer **file-based** persistent volumes on vSAN

Supports basic **vVol** primitives

Enable persistent volume **encryption and snapshots**

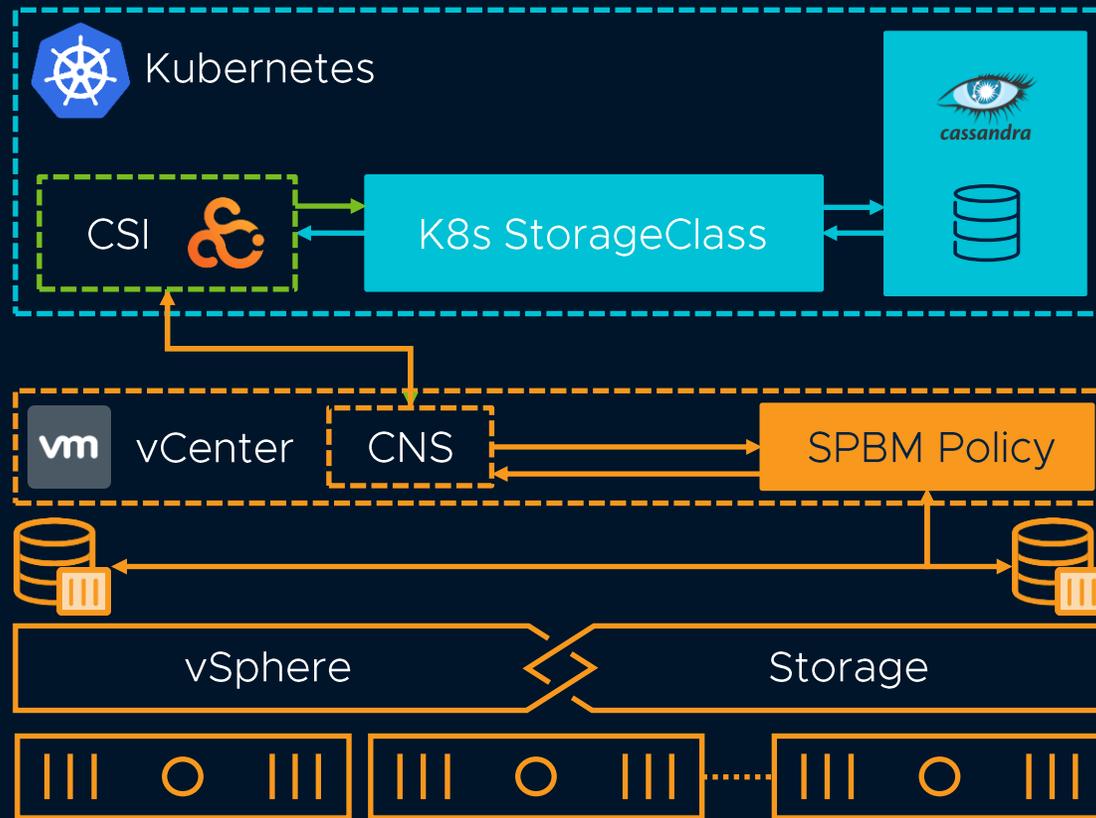
Supports volume resizing

Supports a mix of tooling

- Wavefront
- **Prometheus**
- vR Ops

# Policy Based Management for Kubernetes Workloads

## Dynamic Provisioning Workflow for Block Container Volumes



**Dynamically create volumes**  
on tiers of storage

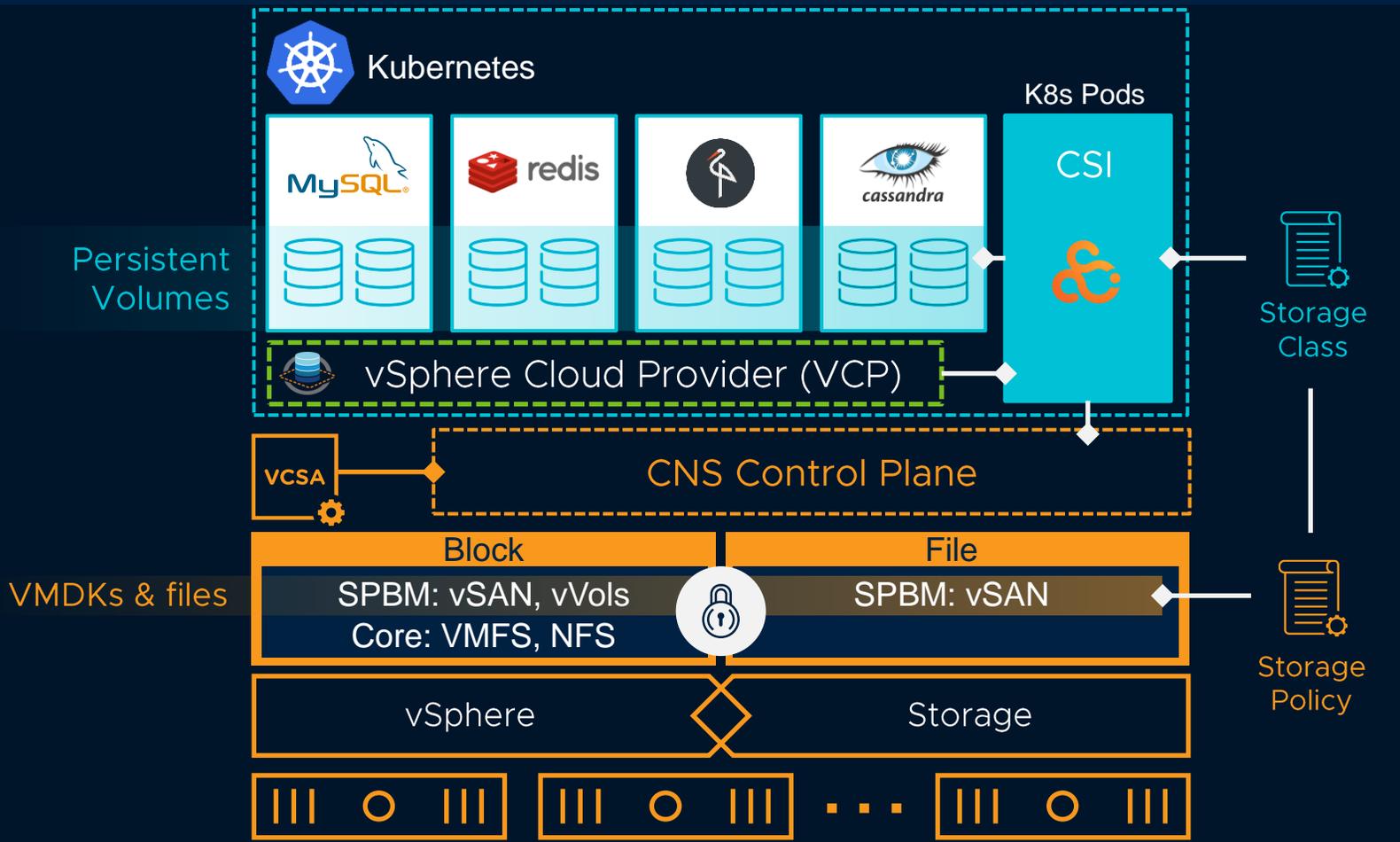
**Preferred method** of storage  
provisioning

Admin **intervention not**  
**required**

Completely **automated**  
**volume LCM**

# VCP to CSI Migration

Beta in K8s 1.19 and vSphere 7.0 U1



Offers **migration** from legacy VCP **to supported CSI driver**

Transparent to the application

**Volumes** converted to FCDs and **included in CNS UI**

Requires new CSI driver and vSphere version

# Recently added Features + Changes

What are they? How to Use Them



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# Recent / Planned Changes

Warning this session was pre-recorded so this is based on plans for a release in the KubeCon Europe timeframe - we will update actual status during Q&A

[NEW] Support for NSX-T based Load Balancers PR: [#292](#)

[NEW] Initial implementation for YAML based config. PR: [#305](#)

[NEW] Update CPI image to use non-root account. PR: [#297](#)

[NEW] Add support for Resource Pool and Folder traversal for Zones/Regions. PR: [#362](#)

[ENHANCE] InstanceExistsByProviderID Signal Deletion to K8s. PR: [#359](#)

[ENAHNCE] Update docs for VMTools exclude-nics filtering. PR: [#349](#)

[ENHANCE] Added guide on how to install CSI on an already-existing K8s cluster. PR: [#293](#)

[ENAHNCE] Documentation updates. PR: [#310](#),

[BUG] Don't cache instances if addresses are not found. PR: [#336](#)

[BUG] vSphere cloud-controller-manager should tolerate not-ready taint. PR: [#339](#)

[BUG] Prevent guest from getting cached when unable to get properties. PR: [#343](#)

[BUG] Remove ToLower when using FQDN. PR: [#352](#)

[BUG] Log does not print node initialization success when zone labels are not configured. PR: [#361](#)

# Kubernetes on Desktop hypervisors

## Unique aspects of running on Fusion and Workstation



KubeCon



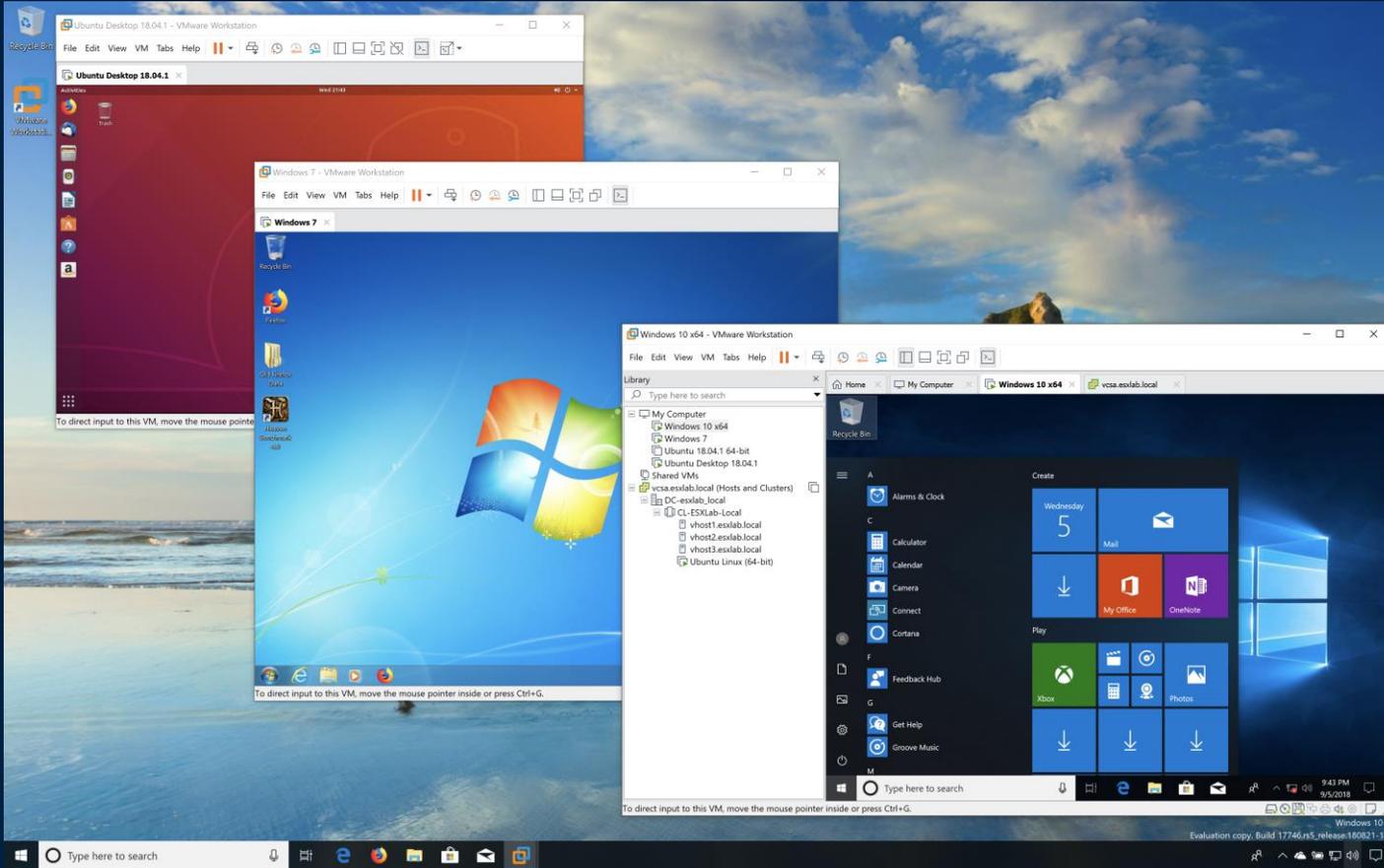
CloudNativeCon

Europe 2020

*Virtual*

# VMware Desktop Hypervisors: Fusion + Workstation

## Industry changing local virtualization tools



## VMware Workstation

- Type-2 Hypervisor
- Originally “VMware 1.0 for Linux”
  - Introduced in 1999
  - ESX was based on this
- Windows + Linux
  - Host and Guest
- Virtual Networking
- Used by *millions* of Developers and IT professionals worldwide



KubeCon



CloudNativeCon

Virtual

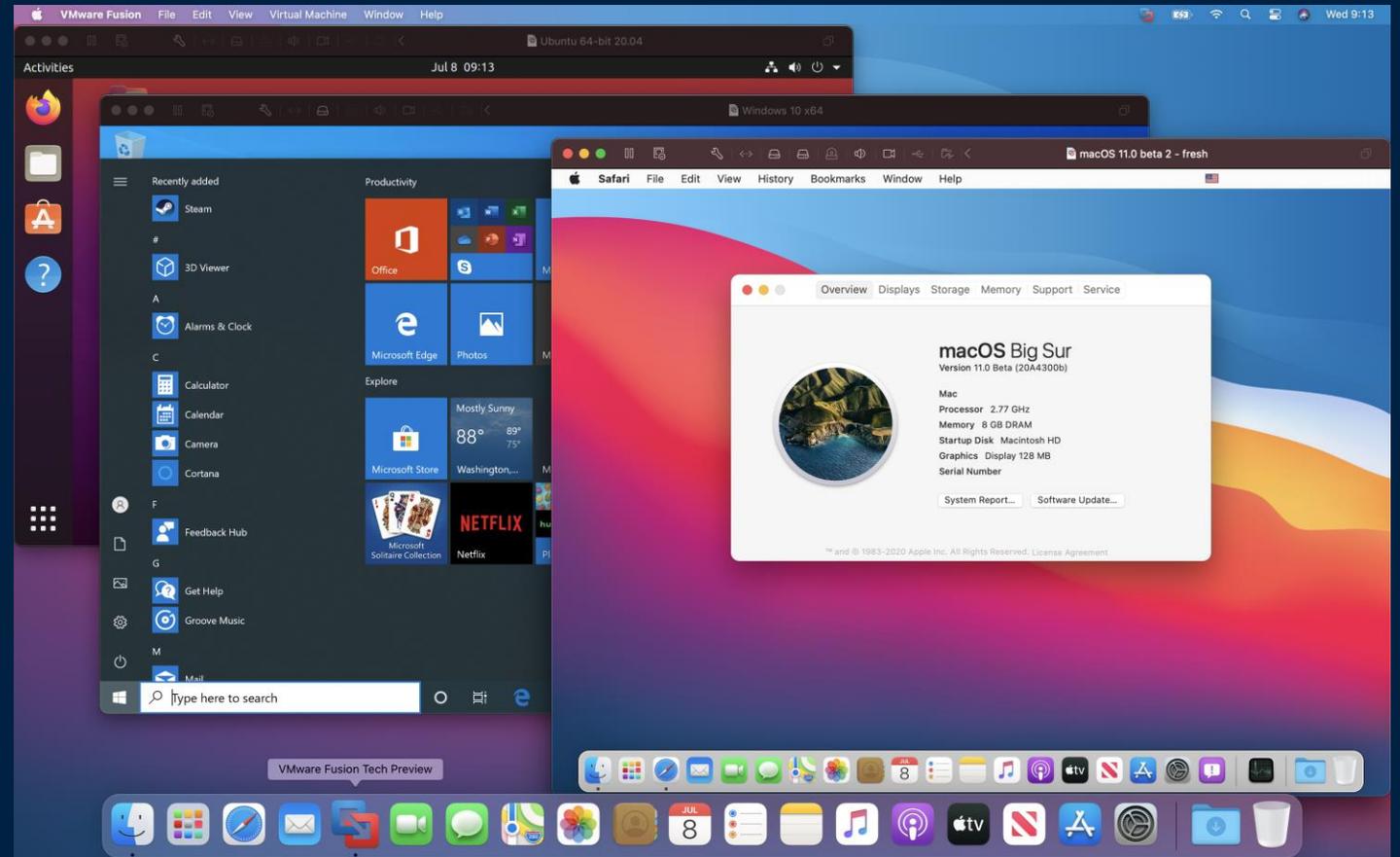
Europe 2020

# VMware Desktop Hypervisors: Fusion + Workstation

Industry changing local virtualization tools

## VMware Fusion

- Type-2 Hypervisor
- Originally introduced in 2007
  - Possible because Apple transitioned to Intel x86
  - Based on Workstation code
- Mac only
- Windows, Linux + macOS Guests
- Virtual Networking
- Used by *millions* of Developers and IT professionals worldwide



# Project Nautilus: OCI Containers in Fusion and Workstation

```
mike@OctoBook: ~  
vctl - A CLI tool for the Nautilus Container Engine powered by VMware Fusion  
vctl Highlights:  
• Build and run OCI containers.  
• Push and pull container images between remote registries & local storage.  
• Use a lightweight virtual machine (CRX VM) based on VMware Photon OS to host a container  
• Use 'vctl system config -h' to learn more.  
• Easy shell access into virtual machine that hosts container. See 'vctl execvm'.  
  
USAGE:  
vctl COMMAND [OPTIONS]  
  
COMMANDS:  
build      Build a container image from a Dockerfile.  
create     Create a new container from a container image.  
describe   Show details of a container.  
exec       Execute a command within a running container.  
execvm     Execute a command within a running virtual machine that hosts container.  
help       Help about any command.  
images     List container images.  
ps         List containers.  
pull       Pull a container image from a registry.  
push       Push a container image to a registry.  
rm         Remove one or more containers.  
rmi        Remove one or more container images.  
run        Run a new container from a container image.  
start      Start an existing container.  
stop       Stop a container.  
system     Manage the Nautilus Container Engine.  
tag        Tag container images.  
version    Print the version of vctl.  
  
Run 'vctl COMMAND --help' for more information on a command.  
  
OPTIONS:  
-h, --help  Help for vctl  
  
mike@OctoBook
```

## Containers without Docker

- vctl: New but familiar CLI
- Based on containerd
- Facilitated by 'CRX' runtime from vSphere (Project Pacific)
- Simple 'build', 'run', 'pull', 'push' workflows
  - **'kind' support coming soon!**
- containerd socket exposed: hack away!
  - `~/.vctl/storage/containerd/run/containerd/containerd.sock`

See

<https://github.com/VMwareFusion/nautilus>  
for latest (20H2 as of now)

# Desktop Hypervisor resources

15.5.6 Download Links: [Linux](#) [Windows](#) MAC: [Fusion](#) [11.5.5](#)

## Workstation Tech Preview 20H2

- Download Links [Linux](#) [Windows](#) [MAC](#)
- TP getting Started Guide [link](#)

# Kubernetes on a Desktop Hypervisor

demo



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# Minikube on Workstation for Windows

Linux and MAC (Fusion) are similar

Step 1 – download the parts you will need

1. Windows Installer for Workstation hypervisor

<https://bit.ly/get-workstation-tp>

2. Go language – we are going to install driver by building it from source

<https://golang.org/dl/go1.14.6.windows-amd64.msi>

3. Git for command line – needed by go

[github.com/git-for-windows/git/releases/download/v2.28.0.windows.1/Git-2.28.0-64-bit.exe](https://github.com/git-for-windows/git/releases/download/v2.28.0.windows.1/Git-2.28.0-64-bit.exe)

4. Kubectl Kubernetes CLI

<https://storage.googleapis.com/kubernetes-release/release/v1.18.6/bin/windows/amd64/kubectl.exe>

5. Minikube for Windows

[github.com/kubernetes/minikube/releases/download/v1.12.1/minikube-installer.exe](https://github.com/kubernetes/minikube/releases/download/v1.12.1/minikube-installer.exe)



KubeCon



CloudNativeCon

Europe 2020

Virtual

# Minikube on Workstation for Windows

## Step 2 Install what you downloaded

1. Install the Workstation hypervisor
2. Install Go language tools
3. Install Git
4. Put Workstation and Go directories in your search path  
Example `C:\Program Files (x86)\VMware, C:\Go\bin`
5. Install Docker machine driver used by minikube, at a command line  
`go get -u github.com/machine-drivers/docker-machine-driver-vmware`
6. Put kubectl in a location within your search path
7. Install minikube



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# Minikube on Workstation for Windows

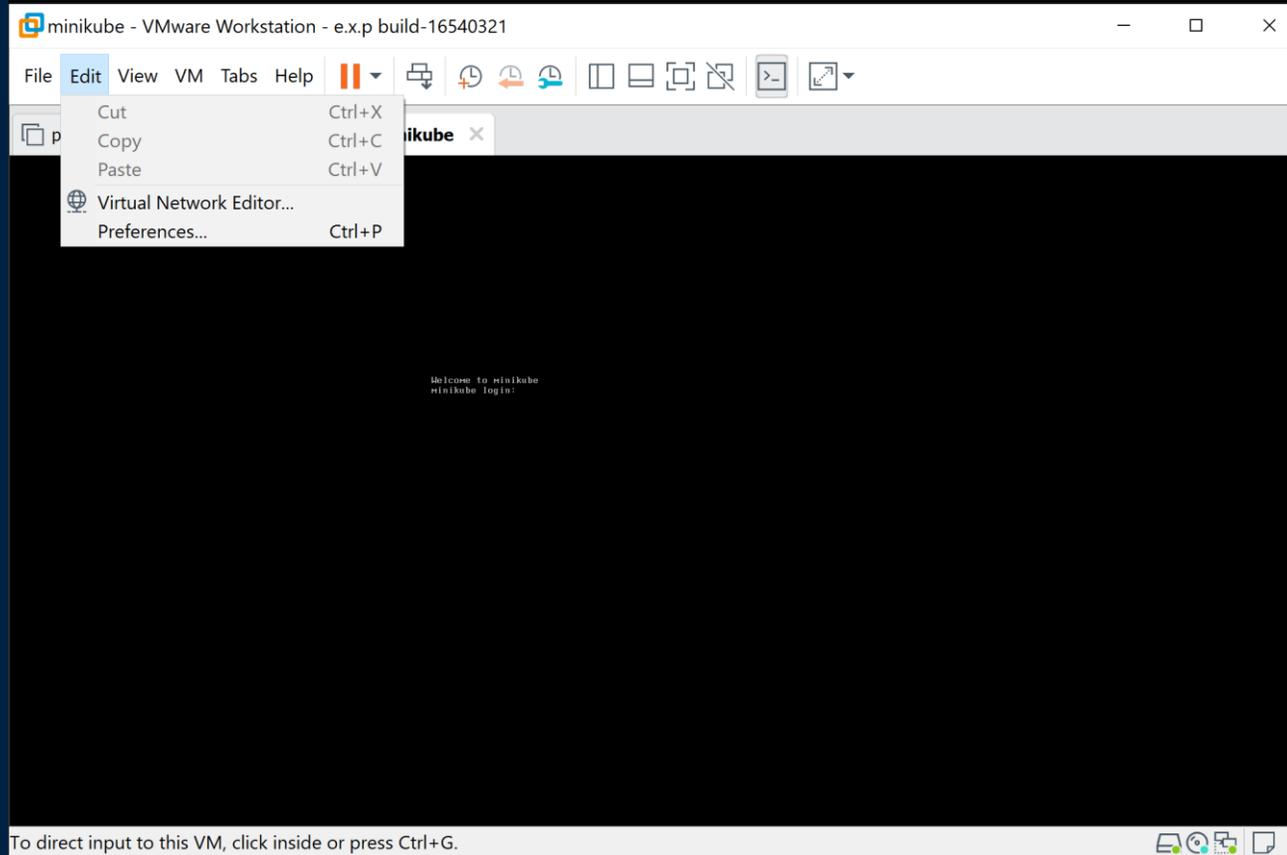
Step 3 - Configure minikube : do these at a command line

```
minikube config set vm-driver vmware  
minikube config set cpus 4  
minikube config set memory 16384  
minikube config set disk-size 50g  
minikube config set host-only-cidr 192.168.99.1/24  
minikube config set kubernetes-version v1.18.6
```

```
minikube config view
```

# Invoke network editor

Optional Step 4: Set network CIDR using Network editor from VMware Workstation Menu

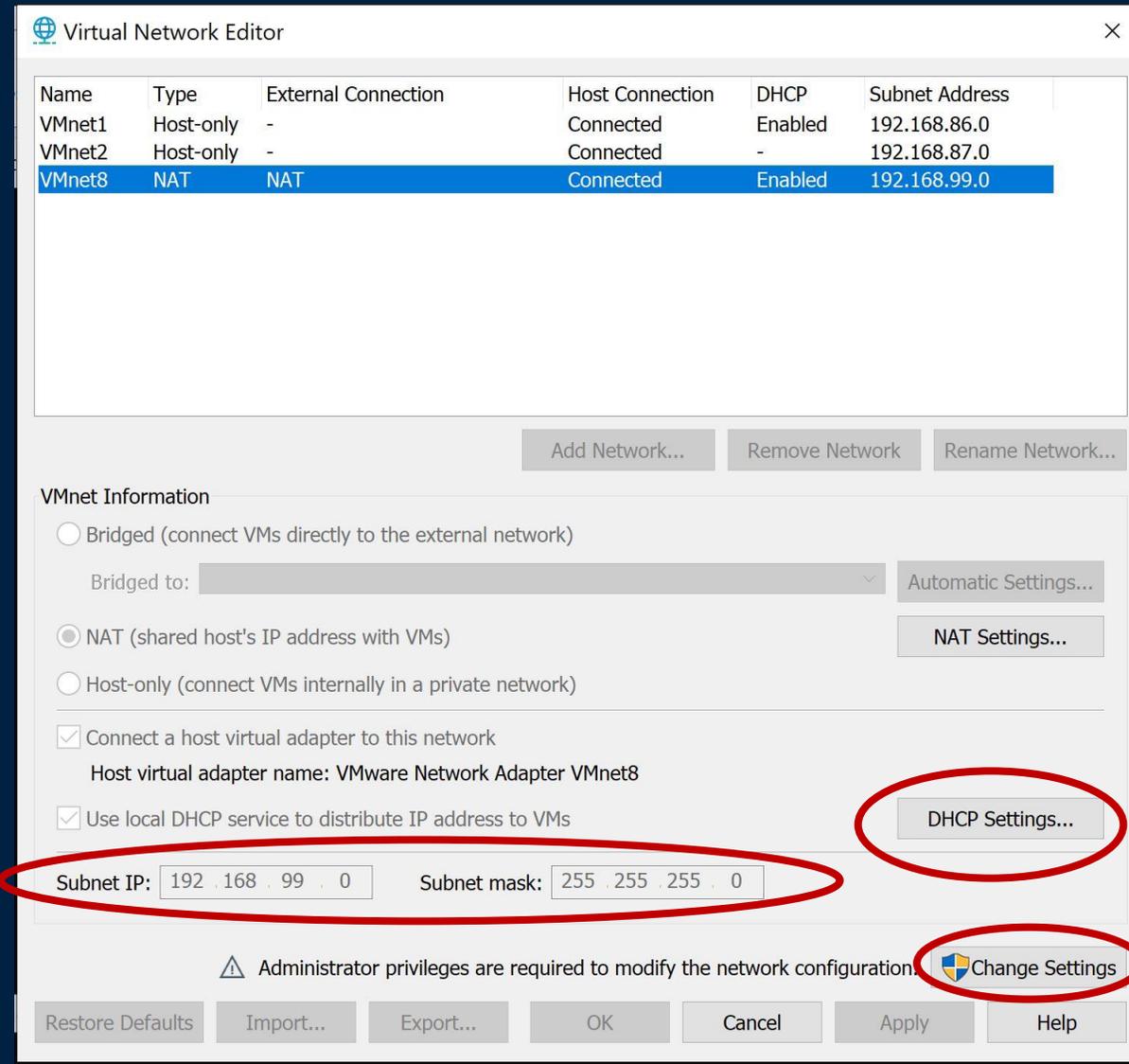


# Invoke network editor

We are going to edit VMnet8

We are changing the CIDR of this network to 192.168.99.0/24

1. Hit Change Settings button
2. Change the Subnet IP
3. Hit the DHCP Settings button



# Invoke network editor

We are going to edit VMnet8

Change prefix of DHCP range to 192.168.99.x

The screenshot shows the 'Virtual Network Editor' window. A 'DHCP Settings' dialog box is open, displaying the following information:

- Network: vmnet8
- Subnet IP: 192.168.99.0
- Subnet mask: 255.255.255.0
- Starting IP address: 192 . 168 . 99 . 128
- Ending IP address: 192 . 168 . 99 . 254
- Broadcast address: 192.168.99.255

The 'Starting IP address' and 'Ending IP address' fields are circled in red. Below these fields are spinners for 'Default lease time' (0 days, 0 hours, 30 minutes) and 'Max lease time' (0 days, 2 hours, 0 minutes). At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

In the background, the main window shows network configuration options:

- NAT (shared host's IP address with VMs)
- Host-only (connect VMs internally in a private network)
- Connect a host virtual adapter to this network  
Host virtual adapter name: VMware Network Adapter VMnet8
- Use local DHCP service to distribute IP address to VMs

At the bottom of the main window, there is a warning: 'Administrator privileges are required to modify the network configuration.' and a 'Change Settings' button. At the very bottom are buttons for 'Restore Defaults', 'Import...', 'Export...', 'OK', 'Cancel', 'Apply', and 'Help'.

# Minikube on Workstation for Windows

## Step 5 – Start minikube – build and run a VM with Kubernetes installed

```
C:\>minikube start --alsologtostderr -v=8
```

```
...  
*Done! kubectl is now configured to use minikube ...
```

```
C:\>minikube status
```

```
minikube status  
minikube  
type: Control Plane  
host: Running  
kubelet: Running  
apiserver: Running  
kubeconfig: Configured
```

```
C:\>kubectl version
```

```
Client Version: version.Info{Major:"1", Minor:"18", GitVersion:"v1.18.6", GitCommit:"dff82dc0de47299ab66c83c626e08b245ab19037", GitTreeState:"clean",  
BuildDate:"2020-07-15T16:58:53Z", GoVersion:"go1.13.9", Compiler:"gc", Platform:"windows/amd64"}  
Server Version: version.Info{Major:"1", Minor:"18", GitVersion:"v1.18.6", GitCommit:"dff82dc0de47299ab66c83c626e08b245ab19037", GitTreeState:"clean",  
BuildDate:"2020-07-15T16:51:04Z", GoVersion:"go1.13.9", Compiler:"gc", Platform:"linux/amd64"}
```

# Minikube on Workstation for Windows

## Step 6 – use Kubernetes

```
C:\>kubectl version
```

```
Client Version: version.Info{Major:"1", Minor:"18", GitVersion:"v1.18.6", GitCommit:"dff82dc0de47299ab66c83c626e08b245ab19037", GitTreeState:"clean", BuildDate:"2020-07-15T16:58:53Z",  
GoVersion:"go1.13.9", Compiler:"gc", Platform:"windows/amd64"}  
Server Version: version.Info{Major:"1", Minor:"18", GitVersion:"v1.18.6", GitCommit:"dff82dc0de47299ab66c83c626e08b245ab19037", GitTreeState:"clean", BuildDate:"2020-07-15T16:51:04Z",  
GoVersion:"go1.13.9", Compiler:"gc", Platform:"linux/amd64"}
```

```
C:\>minikube addons enable metallb
```

```
C:\>minikube addons enable dashboard
```

```
C:\>minikube addons list
```

ADDON NAME	PROFILE	STATUS
ambassador	minikube	disabled
dashboard	minikube	enabled <input checked="" type="checkbox"/>
default-storageclass	minikube	enabled <input checked="" type="checkbox"/>
efk	minikube	disabled
freshpod	minikube	disabled
gvisor	minikube	disabled
helm-tiller	minikube	disabled
ingress	minikube	disabled
ingress-dns	minikube	disabled
istio	minikube	disabled
istio-provisioner	minikube	disabled
kubevirt	minikube	disabled
logviewer	minikube	disabled
metallb	minikube	enabled <input checked="" type="checkbox"/>



KubeCon



CloudNativeCon

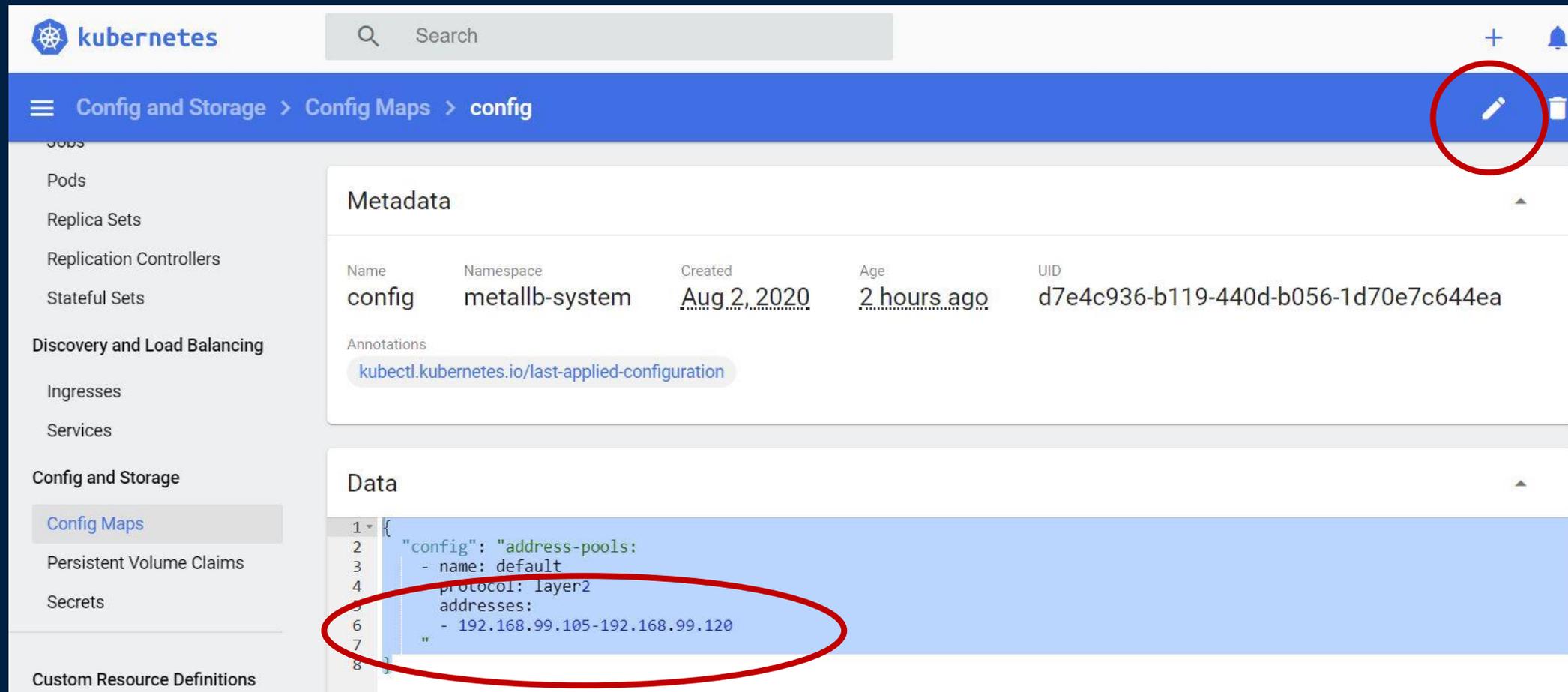
Europe 2020

Virtual

# Minikube on Workstation for Windows

Configure load balancer using dashboard

C:\>minikube dashboard



The screenshot shows the Kubernetes dashboard interface. The breadcrumb navigation is "Config and Storage > Config Maps > config". The "Edit" icon (pencil) in the top right is circled in red. The "Metadata" section shows the ConfigMap details:

Name	Namespace	Created	Age	UID
config	metallb-system	Aug 2, 2020	2 hours ago	d7e4c936-b119-440d-b056-1d70e7c644ea

The "Annotations" section contains the link [kubectrl.kubernetes.io/last-applied-configuration](https://kubectrl.kubernetes.io/last-applied-configuration).

The "Data" section is highlighted with a red oval and contains the following JSON configuration:

```
1 - {  
2   "config": "address-pools:  
3     - name: default  
4     protocol: layer2  
5     addresses:  
6     - 192.168.99.105-192.168.99.120  
7   }  
8 }
```

# Minikube on Workstation for Windows

## Deploy a service

Open a new command prompt session since the dashboard has the first one tied up

```
C:\>kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.10
```

```
C:\>kubectl expose deployment hello-minikube --type=LoadBalancer --port=80 --target-port=8080
```

Find the service in the dashboard with an `exposed` URL link or get the load balancer hosted URL from the command line



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# Kubernetes VMware User Group

## What is it?

Similar to SIGs and Working Groups - intended to serve the needs of users running Kubernetes on particular platforms.

The VMware User group is the first (and currently only) K8s UG for a platform - covers running K8s on all VMware hypervisors.

## Why is this important?

Create community culture among our users

- Users can help each other
- Users can help us make Kubernetes better – and strengthen user experience on our platforms:
  - Feature requests
  - Feedback + issue resolution

## Who is involved?

### Co-chairs

- Steven Wong, MAPBU CET
- Myles Gray, VMware Storage Tech Marketing, UK

### User Co-leads

- Bryson Shepherd, Walmart
- Joe Searcy, T-Mobile

125+ Slack channel participants as of July 2020



KubeCon



CloudNativeCon

Europe 2020

Virtual

# Kubernetes VMware User Group

User Group Meeting:

First Thursday each month 11am PT  
calendar [link](#)



Link to join the group

- [groups.google.com/forum/#!forum/kubernetes-ug-vmware](https://groups.google.com/forum/#!forum/kubernetes-ug-vmware)

Link to join Slack channel

- <https://kubernetes.slack.com/messages/ug-vmware>



# Speaker contact info

Deck link: <https://sched.co/ZewL>



Myles Gray  
VMware

@mylesagray

Some other related sessions:

Cloud Provider out of tree (next): [sched.co/ZeuY](https://sched.co/ZeuY)

K8s User experience (Thursday): [sched.co/Zeue](https://sched.co/Zeue)

vSphere Cloud Provider (Thursday): [sched.co/ZevZ](https://sched.co/ZevZ)



Steve Wong  
VMware

@cantbewong

KubeCon

CloudN

Europe 2021

Virtual



# KEEP CLOUD NATIVE CONNECTED

Thank You

