KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Setup your Laptop

for Kubernetes and get produktive

Archy (Ayrat Khayretdinov)
Prune (Sebastien THOMAS)


KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM

Our speakers

KubeCon | CloudNativeCon
North America 2022

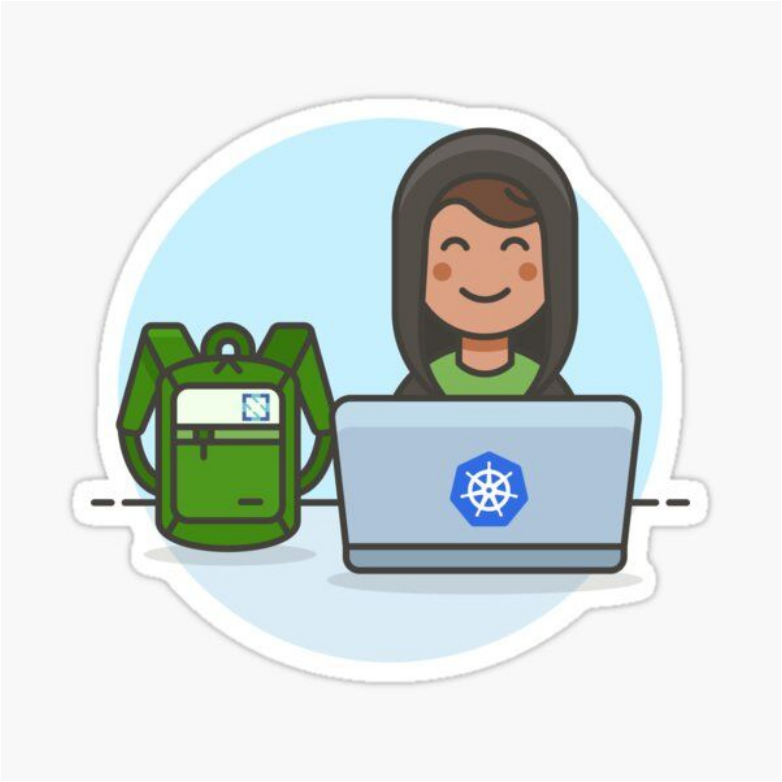
BUILDING FOR THE ROAD AHEAD

DETROIT 2022

October 24-28, 2021

Archy
CNCF Ambassador
Google

Prune
Software Engineer
Wunderkind



Who I am

Archy Khayretdinov

CNCF Ambassador

Google Cloud

CLOUD NATIVE
EASTERN CANADA

CLOUD NATIVE
COMPUTING FOUNDATION
AMBASSADOR

SUPERFROGS
DEPLOY: HIGHER. FURTHER. FASTER.

Kubernetes
Community Days

TORONTO
2020

MONTREAL
Kubernetes
Community Days

KUBERNETES
QUÉBEC
2020
COMMUNITY DAY

Kubernetes Community Day
Ottawa

Who I am

I’m Sebastien / Prune

Lead Software Engineer,
part of the Infrastructure Team at **Wunderkind**
System Engineer, K8s fan, casual Go developer

Co-organiser of *CNCF/Kubernetes Eastern-Canada* meetups

Living in Quebec, CANADA

CLOUD NATIVE
EASTERN CANADA

https://www.linkedin.com/in/prune/

https://github.com/prune998

Scenarios

- **Scenario 1 -** Setup Laptop with Kubectl and Kubernetes
- **Scenario 2** - Authenticating to Kubernetes Cluster (kubeconfig)
- **Scenario 3 -** Create K8s resources using kubectl commands ()
- **Scenario 4 -** Write Declarative YAML Manifests and kubectl apply
  - Deploy apps to multiple clusters, namespaces
  - Deploy apps to multiple environments (kustomize)
  - Deploy 3d party apps (helm)
- **Scenario 5**

Agenda

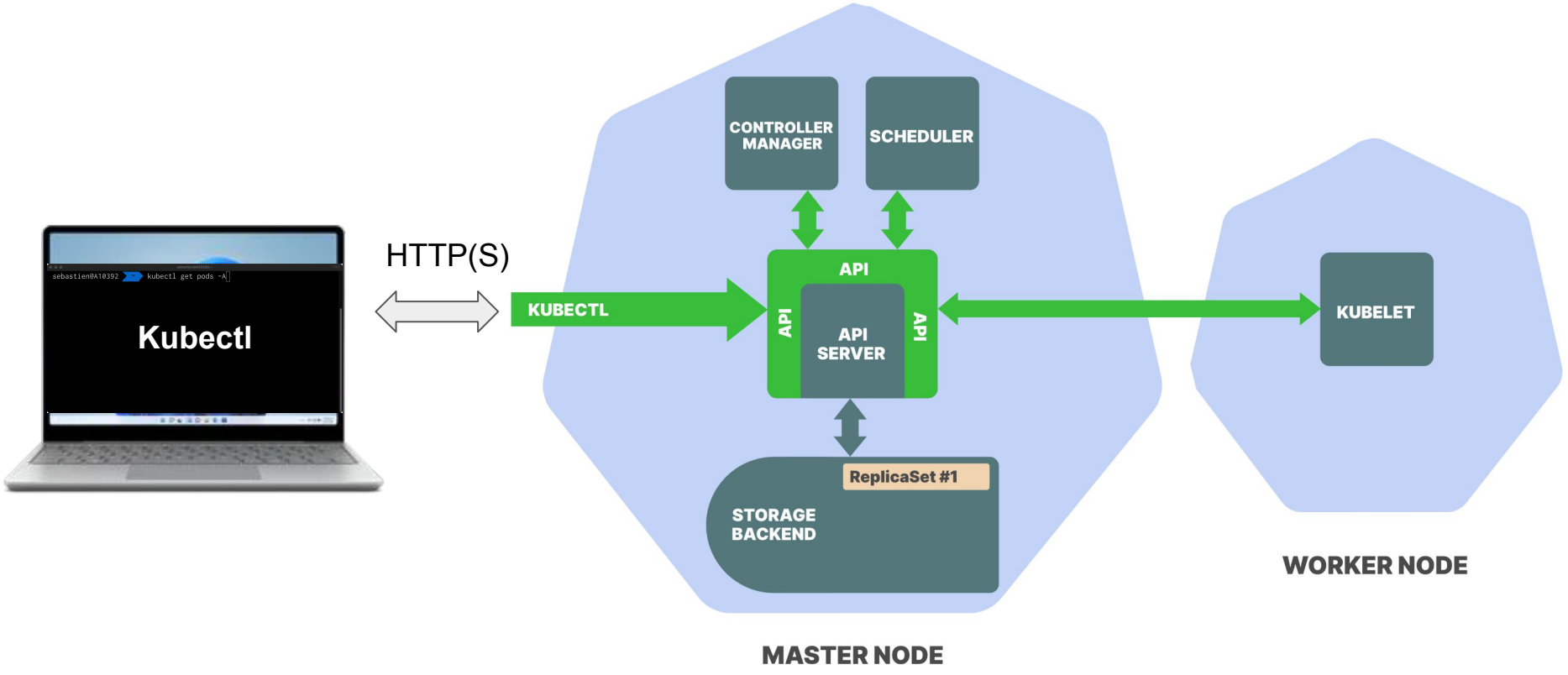
- No more SSH, I need Kubectl
- Replacing Docker-For-Desktop
- Use Containers without Docker
- Create a local K8s cluster
- I’m typing too much kubectl commands
- Kubectl arguments are too long
- My terminal is clogged, I can’t read it
- Too many outputs from k8s commands

Agenda

- I want more from kubectl
- Help me switch my k8s context
- Reading logs is a pain
- I’m fed up with the Shell
- YAML is so hard to manage
- I’m too advanced for this talk, bring it on !
- References

What we need for this tutorial?

Kubectl: Internals



Kubectl: Internals

HTTP(S)

KUBECTL

CONTROLLER MANAGER

SCHEDULER

API

API

API SERVER

API

ReplicaSet #1

STORAGE BACKEND

MASTER NODE

KUBELET

WORKER NODE

What we're talking about today

Requirements for this Tutorial

Mac OS:

- Iterm2 (https://iterm2.com/)
- ZSH (default OsX shell for quite some time now)
- Working **Homebrew** installer
- Any **Kubernetes Cluster** (Docker Desktop or K8s on Cloud)

Linux:

- Any terminal
- ZSH preferably or Bash
- Any **Kubernetes Cluster** (Docker Desktop or K8s on Cloud)

Requirements for this Tutorial: K8s Cluster

KubeCon CloudNativeCon North America 2022

Pick ANY option!

Run it yourself !

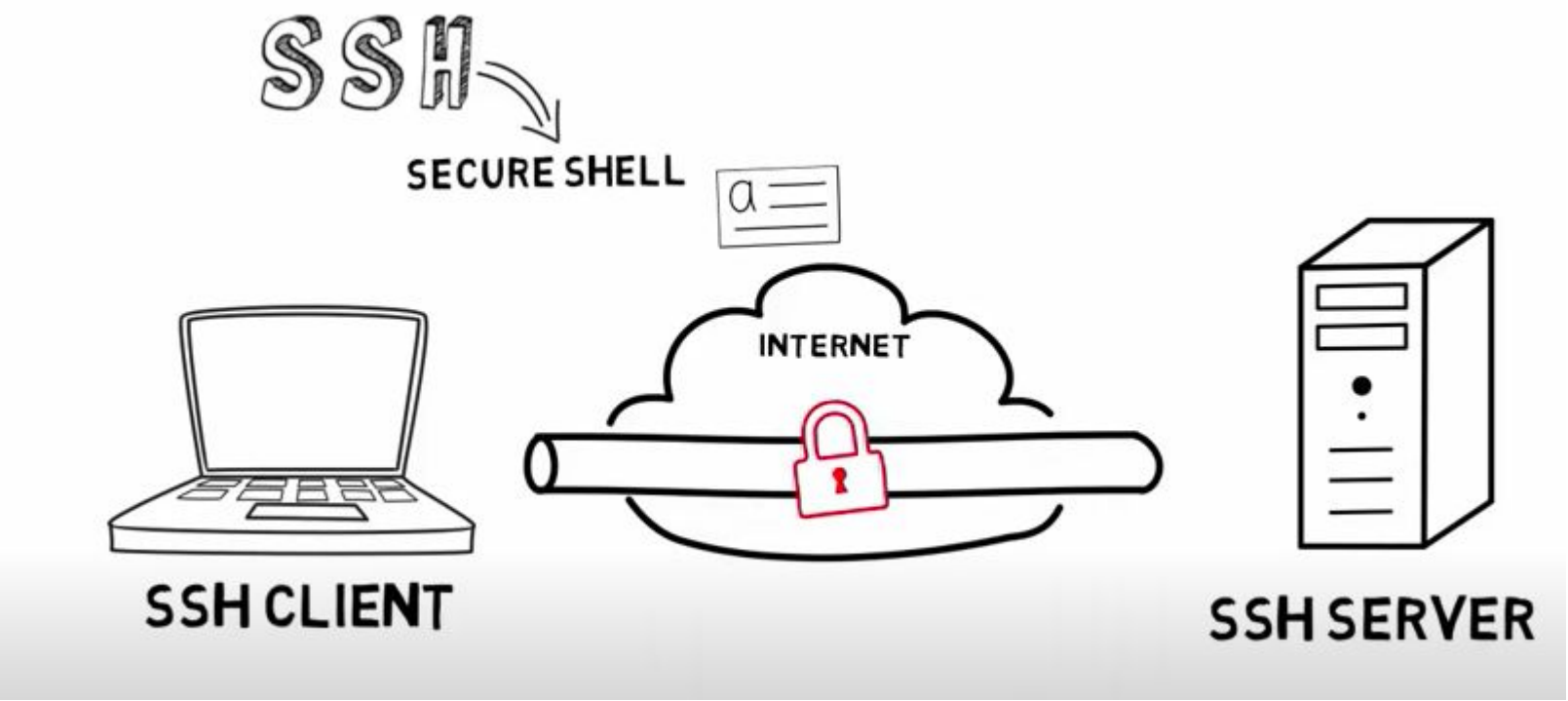
https://rebrand.ly/k8s_tools

Part 1

New SSH

KubeCTL

Using ssh to connect to VM



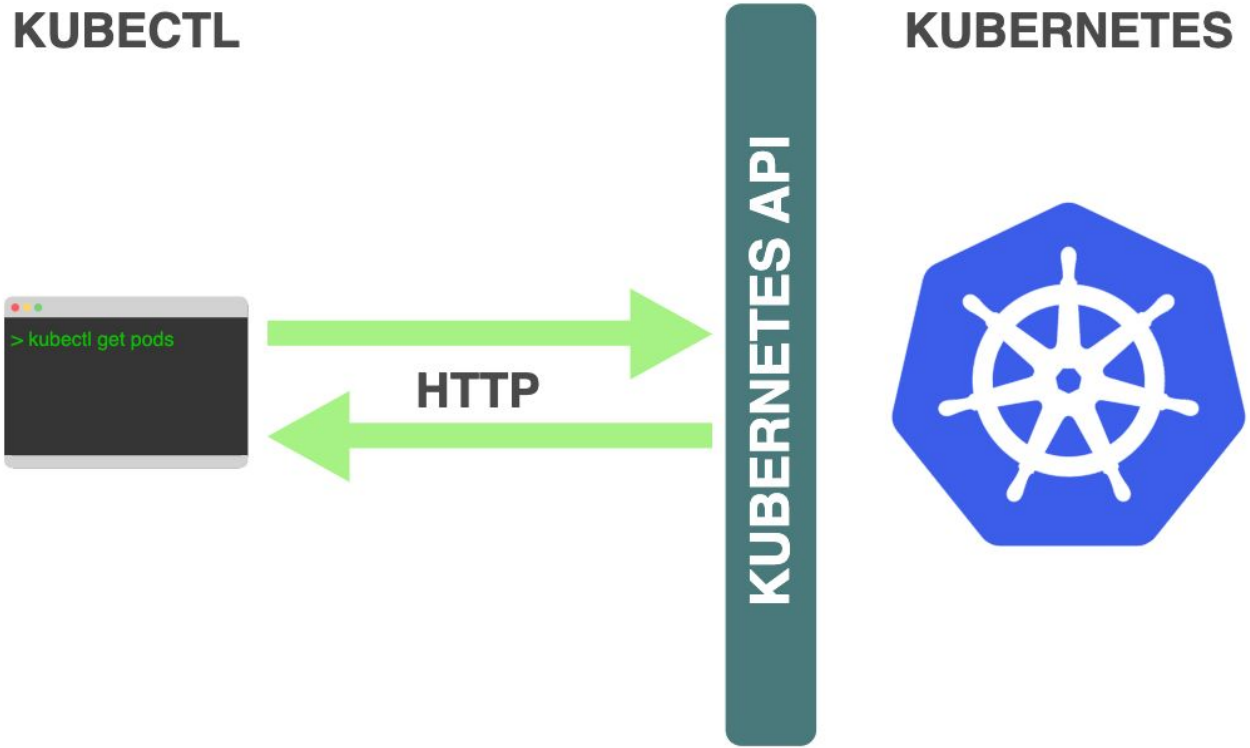

```
$ ssh -i ~/.ssh/id_rsa admin@192.168.24.25
```

Kubectl is a new SSH

← Tweet

Kelsey Hightower
@kelseyhightower

kubectl is the new SSH. I can do this all day.



Demo: Installing Kubectl

Install kubectl

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/darwin/amd64/kubectl"
chmod 755 kubectl
```

(or `brew install kubectl` or install through gcloud CLI…)

For Linux:

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
chmod 755 kubectl
```

Kubectl - Know before it’s too late :)

Always try to use the same `kubectl` version as the server you are targeting

```
kubectl version

Client Version: version.Info{Major:"1", Minor:"22",
GitVersion:"v1.22.2"}
Server Version: version.Info{Major:"1", Minor:"20+",
GitVersion:"v1.20.9-gke.1001"}

WARNING: version difference between client (1.22) and server (1.20)
exceeds the supported minor version skew of +/-1
```

Local K8s Dev Environment

Mac/Windows/Linux

What is the requirements?

Must have:

- **Build Docker Images (Docker Engine)**

  # docker build -t image_name .

- **Run Docker Containers**

  # docker run image_name

- **Deploy to Kubernetes**

  # kubectl create deployment –image=image_name

[Optional] Nice to have:

- **Able to run Docker Compose**

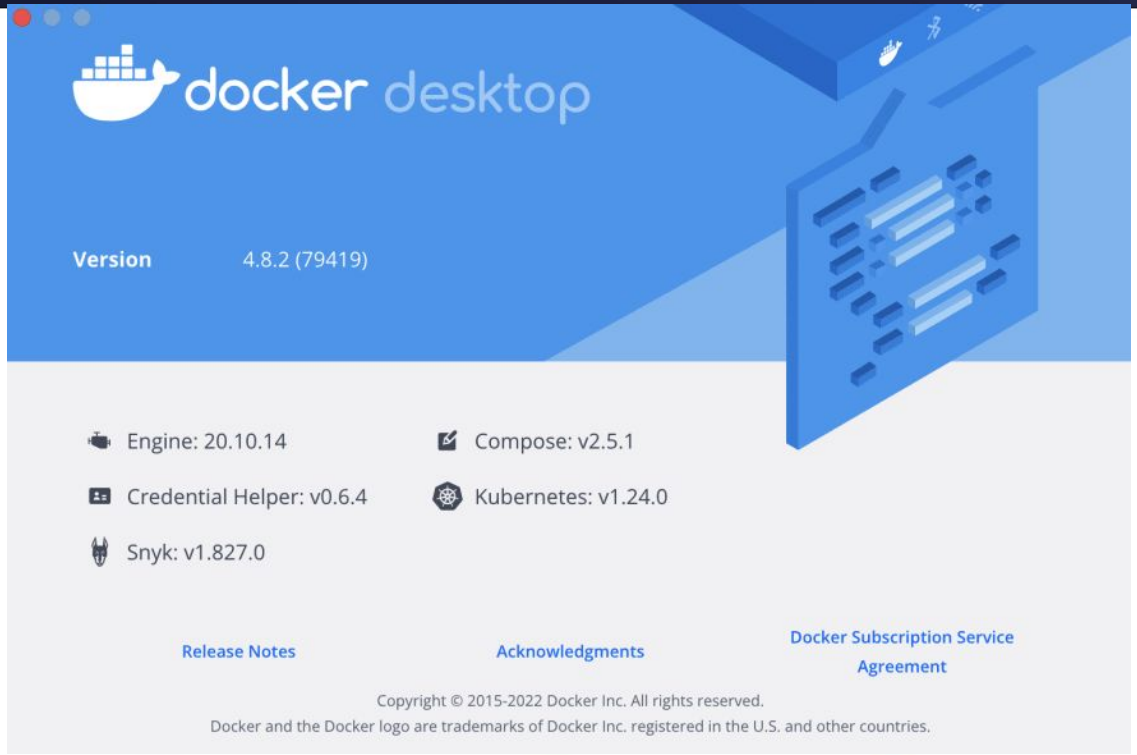
  # docker-compose up .

- **Scan Images**

  # docker scan image

- **Nice UI**

Docker for Desktop



Apple Windows Linux

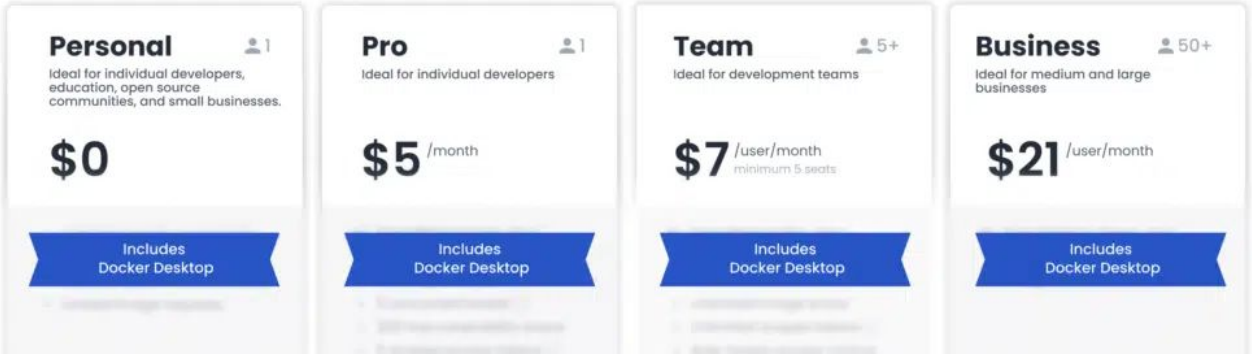
[MAC] Docker for Desktop

What's problem with Docker for Desktop?

Docker Desktop is no longer free for enterprise users

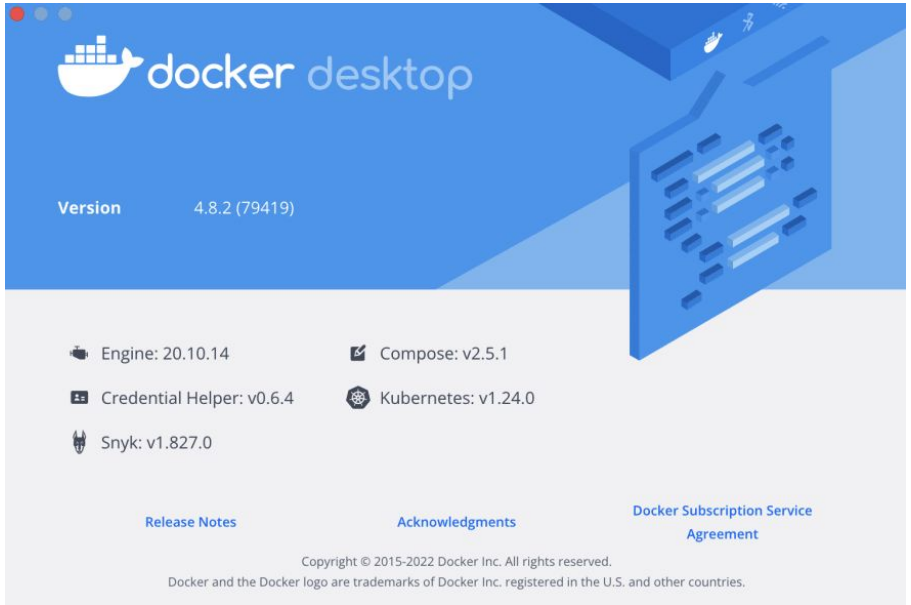
Docker is changing its pricing plans, ending free Docker Desktop use for larger business customers and replacing its Free plan with a Personal plan.

By Scott Carey
Managing Editor, News, InfoWorld | AUG 31, 2021 8:00 AM PDT



Docker for Desktop: Pros and Cons

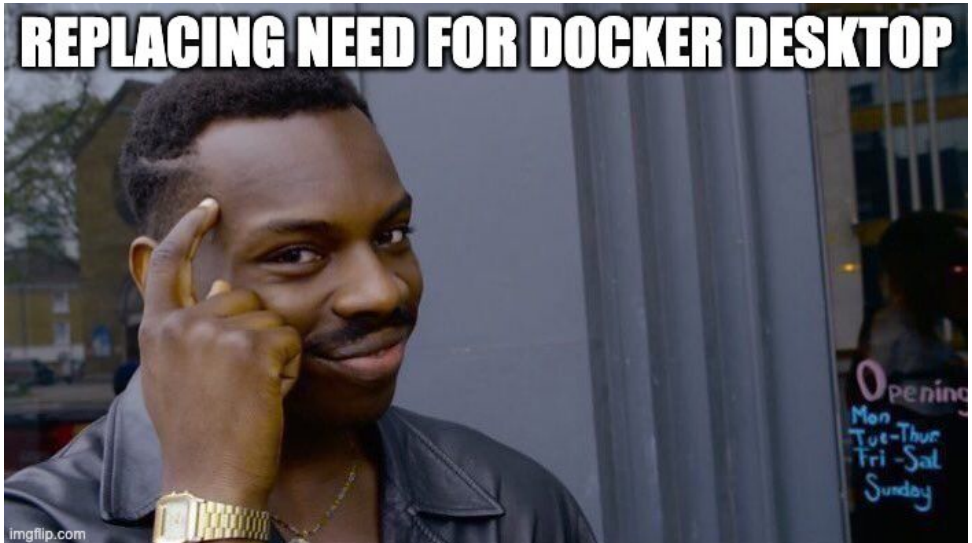
- Multi platform
- UI
- Great Dev Experience
- Paid product for enterprise
- Frequent updates
- High CPU usage
- Little k8s control



Replacing Docker-For-Desktop

For Mac and Windows

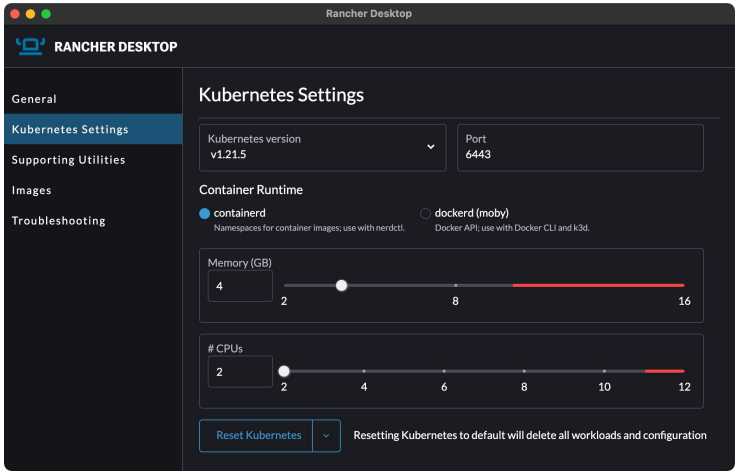
Docker for Desktop Alternatives

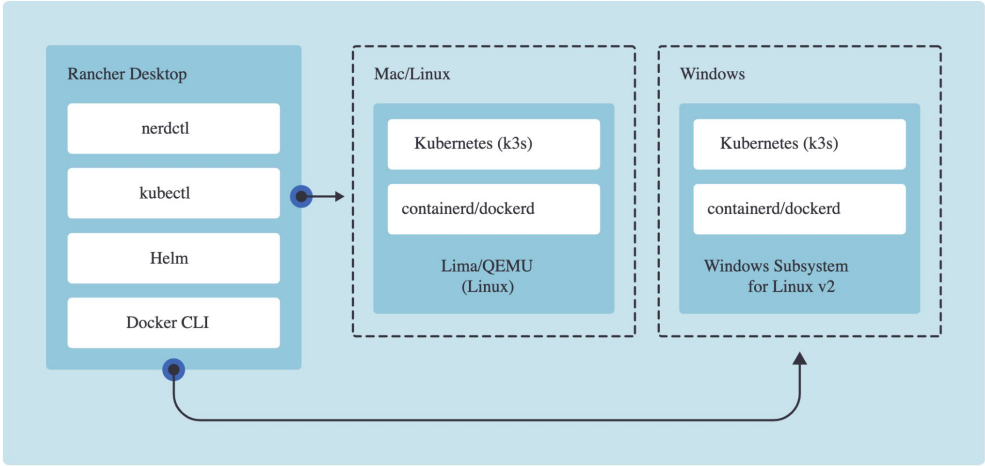


Using Rancher Desktop as Docker Desktop Replacement

Download Mac (Apple Silicon) Download Mac (Intel) Download Windows Install on Linux

RANCHER DESKTOP





[Mac] Containers without Docker: Colima

- Colima Container runtimes on macOS (and Linux) with minimal setup
  - Intel and M1 Macs support
  - Simple CLI interface
  - Docker and **Containerd** support
  - Port Forwarding
  - Volume mounts
  - Kubernetes (hidden K3s)
  - Fully replace Docker-for-Desktop (**but no UI**)

[Mac] Containers without Docker: Colima

Colima

```
brew install colima

# optional, not needed for containerd
brew install docker

# default using Docker runtime
colima start

# start kubernetes local cluster
colima start --with-kubernetes

# remove docker completely
colima start --runtime containerd --with-kubernetes
```

Using minikube as Docker Desktop Replacement

- Minikube: local Kubernetes focusing on making it easy to learn and develop for Kubernetes.

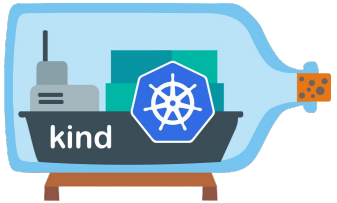
  - Official Kubernetes tool maintained by K8s SIGs
  - Supports the latest Kubernetes release
  - Cross-platform (Linux, macOS, Windows)
  - Multiple container runtimes (CRI-O, containerd, docker)
  - Can use minikube as a Docker Desktop Replacement

```
# Install Docker CLI (https://download.docker.com/ or `brew install
docker`)
# Install Minikube (https://minikube.sigs.k8s.io/docs/start/)
# Install HyperKit (https://minikube.sigs.k8s.io/docs/drivers/hyperkit/)
minikube start --driver hyperkit  # will be auto-detected
Or
minikube start --container-runtime=docker --no-kubernetes
minikube docker-env
eval $(minikube docker-env)
```

Local Kubernetes

Kind

Local Cluster: Kind

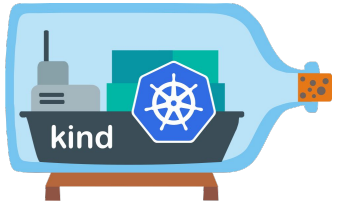


Kind stands for Kubernetes in Docker.
It's a tool to create **lightweight** Kubernetes clusters using **Docker** container "nodes".

- Official Kubernetes tool maintained by K8s SIGs
- It can be used to deploy Local K8s cluster or for CI
- Support ingress / LB (with some tuning)
- Support deployment of multiple clusters / versions
- Supports deployment of single or multi node clusters

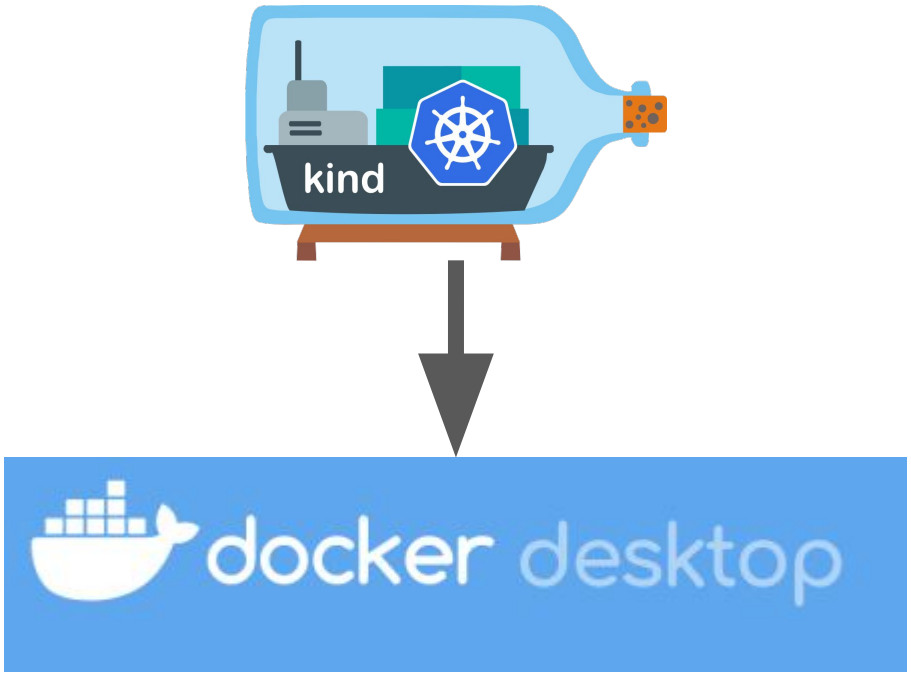
https://kind.sigs.k8s.io/

Local Cluster: Kind

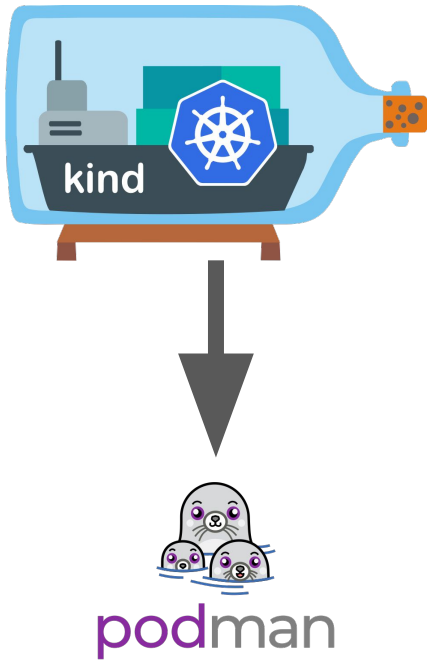



```
$ brew install kind

$ kind create cluster

$ kind get clusters
```

Local Cluster: Kind



Local Cluster: Kind



Replacing Docker Engine

podman=rootless containers

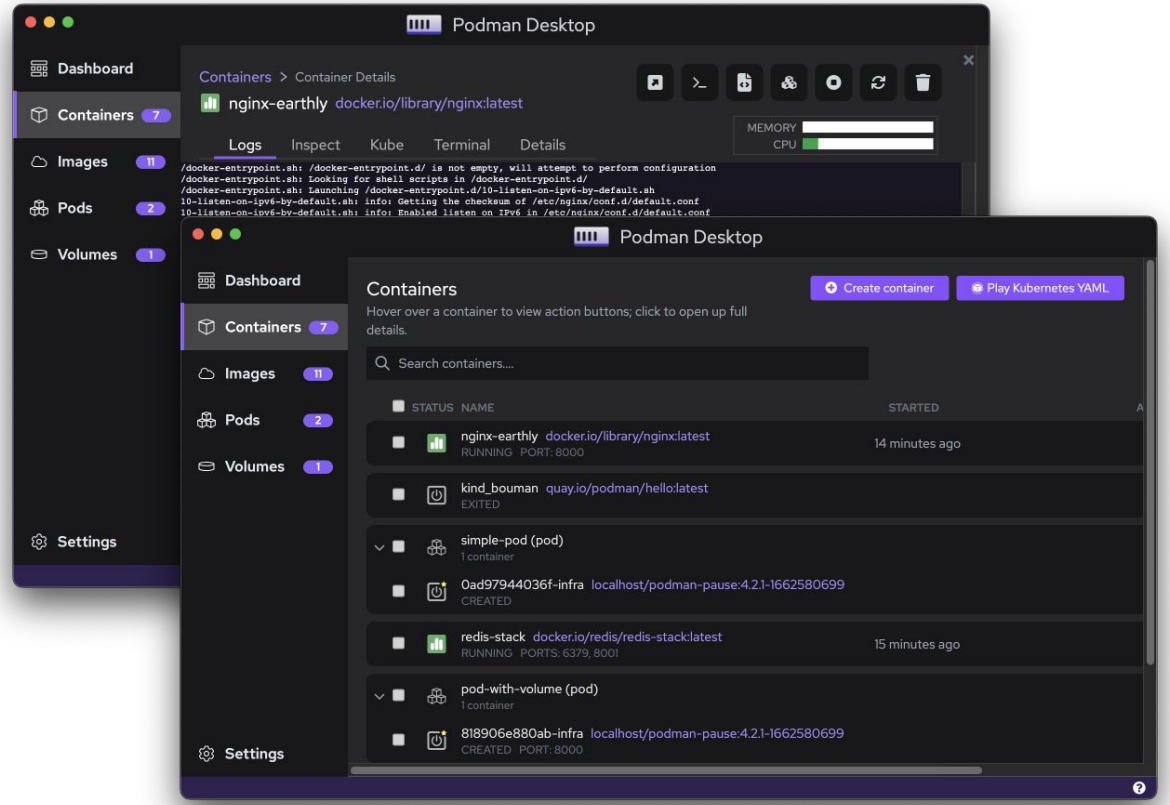
Containers without Docker Engine: PodMan

podman

- Podman is the container Swiss-Army knife maintained by RedHat
  - Replaces **Docker-For-Desktop, Docker Engine, Docker CLI**
  - Works for Windows, Mac, Linux
  - Podman version 3.4+ now support **M1 Apple Macs**
  - **Rootless** mode provide more security
  - Multiple image formats including the OCI and Docker image
  - Container image management

    (managing image layers, overlay filesystems, etc)
  - **However doesn't provide Kubernetes cluster!!! So use Kind**

(Optional) Podman Desktop



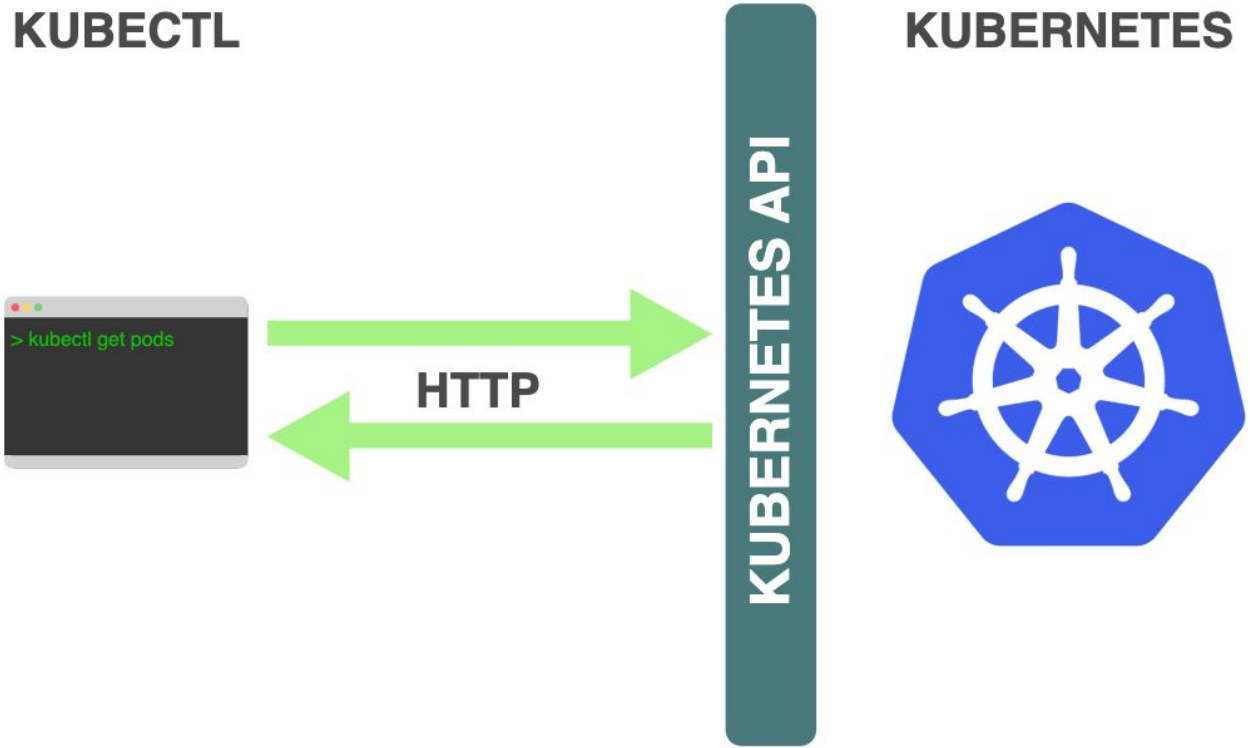
Demo: Installing Podman and Kind

podman

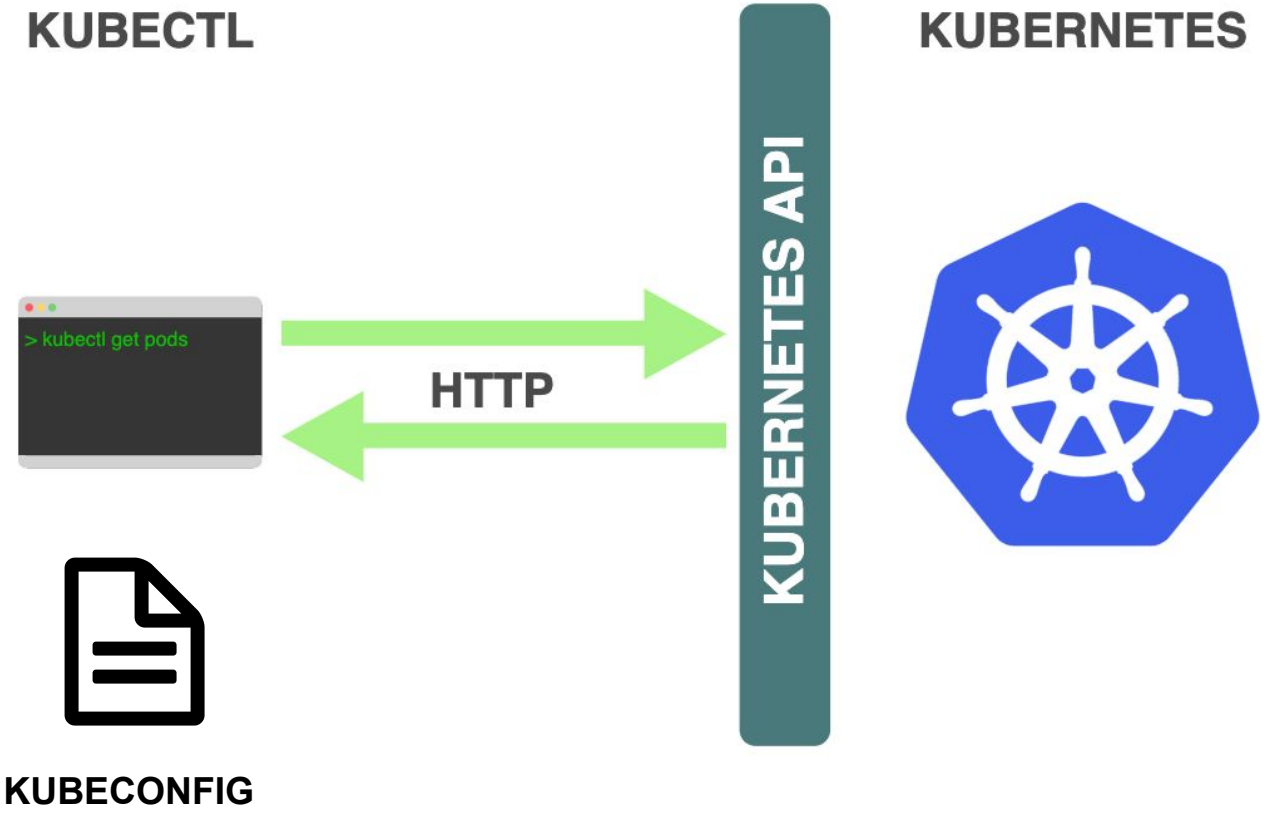
Authenticating to Kubernetes

kubeconfig

Authenticating to Kubernetes Cluster



KUBECONFIG


KUBECTL
KUBERNETES
KUBERNETES API
> kubectl get pods
HTTP
KUBECONFIG

Demo: Authenticating to Kubernetes

KUBECONFIG

Deploying application on k8s

kubectl create
kubectl apply

Creating kubernetes resources

- Creating k8s resources with command line

  # kubectl create deployment webserver --image=nginx --port=80
  # kubectl create service clusterip webserver --tcp 80:80
  # kubectl create configmap webserver-config --from-file web.config

- Creating k8s resources **Declaratively** with **YAML**:

  # kubectl apply -f nginx_pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.7.9
    ports:
     - containerPort: 80
```

Creating Pods and Deployments

Kubectl can be used to generate YAML manifests and help you build your own:

- `kubectl create <resource>` to create a resource (`kubectl run` for pods)
- `-output yaml` (`-o yaml`) to dump the yaml
- `-dry-run=client` to simulate the resource
- `kubectl explain <resource>` to get explanation of the resource

Demo: Deploying applications on K8s with YAML

kubectl apply

Part 2

I’m Typing Too Much Kubectl

Aliasing

I’m Typing Too Much Kubectl

Just create some Alias: use **`k`** instead of **`kubectl`**

```
alias k=kubectl
```

You can add this command to your *.zshrc* so it is added to all your (new) shells

Spoiler alert: it’s the same for Bash (in your *.bashrc*)

BONUS

- Use singular name of the resource
- Use short name of the resource

pods -> pod -> po

```
k api-resources

NAME          SHORTNAMES  APIVERSION              NAMESPACED   KIND
pods          po          v1                      true         Pod
deployments   deploy      apps/v1                 true         Deployment
ingresses     ing         networking.k8s.io/v1    true         Ingress
services      svc         v1                      true         Service
namespaces    ns          v1                      false        Namespace
```

Kubectl arguments too long

Shell Completion

Shell Setup: Completion

Completions allows to use <tab> to list options or some list of resources

```
source <(kubectl completion zsh)
```

For Bash:

```
source <(kubectl completion bash)
complete -o default -F __start_kubectl k
```

Shell Setup: BASH

Alias, Completion… in your .zshrc: use **`k`** instead of **`kubectl`**

```
alias k=kubectl
source <(kubectl completion bash)
complete -o default -F __start_kubectl k
ulimit -n 2048          # kubectl opens one cnx (file) per resource
```

Shell Setup: Side note

Kubectl, or the Kube GO client, sometimes open multiple connexions to the server, sometimes too much. This is much experiences when using `helm diff` on a large setup

Change your Open-File limit (file descriptors):

```
ulimit -n 2048   # kubectl opens one cnx (file) per resource
```

Terminal for Produktivity

Oh My ZSH !

Shell Setup: ZSH

ZSH, OhMyZSH and themes:

- Oh-My-ZSH : lots of features in your shell
  - Use plugins

```
plugins=(brew kubectl git python tmux vault terraform)
```

- Themes
  - Agnoster ZSH theme: better prompt using Powerline Fonts
  - PowerLevel10k: emphasizes speed, flexibility and out-of-the-box experience
- Fonts
  - Powerline Font: recommend NerdFonts Inconsolata or Firacode

Shell Setup: ZSH

Alias `k` instead of `kubectl` is not needed anymore

```
# alias k=kubectl # replaced by OhMyZSH kubectl plugin
source <(kubectl completion zsh)
ulimit -n 2048          # kubectl opens one cnx (file) per resource
```

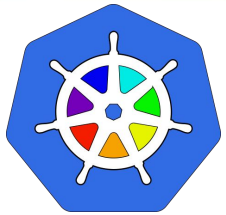
Kubernetes Shell Tooling

There's too many output from kubectl, I can't read it

Kubecolor

Kubecolor (https://github.com/kubecolor/kubecolor)

Colorize the output of kubectl command, highlight important features



```
  brew install kubecolor/tap/kubecolor
Or
  go install -ldflags="-X main.Version=latest"
github.com/kubecolor/kubecolor/cmd/kubecolor@latest

 # in your .zshrc or .bashrc
 alias kubectl=kubecolor
 compdef kubecolor=kubectl # only needed for zsh
```

Kubecolor

(https://github.com/kubecolor/kubecolor)

KubeCon CloudNativeCon North America 2022

prune@arcadia:~

```
❯ kubectl get pods -A
NAMESPACE            NAME                                         READY   STATUS    RESTARTS          AGE
kube-system          coredns-6d4b75cb6d-j2774                     1/1     Running   0                 20h
kube-system          coredns-6d4b75cb6d-jvmg9                     1/1     Running   0                 20h
kube-system          etcd-demo-control-plane                      1/1     Running   0                 20h
kube-system          kindnet-wx2sx                                1/1     Running   0                 20h
kube-system          kindnet-wx9jg                                1/1     Running   0                 20h
kube-system          kube-apiserver-demo-control-plane            1/1     Running   0                 20h
kube-system          kube-controller-manager-demo-control-plane   1/1     Running   43 (3h11m ago)    20h
kube-system          kube-proxy-6wh5v                             1/1     Running   0                 20h
kube-system          kube-proxy-nknzg                             1/1     Running   0                 20h
kube-system          kube-scheduler-demo-control-plane            1/1     Running   43 (3h11m ago)    20h
local-path-storage   local-path-provisioner-6b84c5c67f-6fhrr      1/1     Running   0                 20h
```

prune@arcadia:~

```
❯ kubecolor get pods -A
NAMESPACE            NAME                                         READY   STATUS    RESTARTS          AGE
kube-system          coredns-6d4b75cb6d-j2774                     1/1     Running   0                 20h
kube-system          coredns-6d4b75cb6d-jvmg9                     1/1     Running   0                 20h
kube-system          etcd-demo-control-plane                      1/1     Running   0                 20h
kube-system          kindnet-wx2sx                                1/1     Running   0                 20h
kube-system          kindnet-wx9jg                                1/1     Running   0                 20h
kube-system          kube-apiserver-demo-control-plane            1/1     Running   0                 20h
kube-system          kube-controller-manager-demo-control-plane   1/1     Running   43 (3h19m ago)    20h
kube-system          kube-proxy-6wh5v                             1/1     Running   0                 20h
kube-system          kube-proxy-nknzg                             1/1     Running   0                 20h
kube-system          kube-scheduler-demo-control-plane            1/1     Running   43 (3h19m ago)    20h
local-path-storage   local-path-provisioner-6b84c5c67f-6fhrr      1/1     Running   0                 20h
```

Kubecolor

(https://github.com/kubecolor/kubecolor)

KubeCon CloudNativeCon North America 2022

prune@arcadia:~

```
❯ export KUBECOLOR_OBJ_FRESH=12h # highlight resources newer than 12h
k get pods -A
k run another-test-pod --image=alpine:latest sleep 30
k get pods
sleep 10
k get pods
NAMESPACE           NAME                                       READY   STATUS             RESTARTS          AGE
default             test-pod                                   0/1     CrashLoopBackOff   5 (18s ago)       6m24s
kube-system         coredns-6d4b75cb6d-j2774                   1/1     Running            0                 20h
kube-system         coredns-6d4b75cb6d-jvmg9                   1/1     Running            0                 20h
kube-system         etcd-demo-control-plane                    1/1     Running            0                 20h
kube-system         kindnet-wx2sx                              1/1     Running            0                 20h
kube-system         kindnet-wx9jg                              1/1     Running            0                 20h
kube-system         kube-apiserver-demo-control-plane          1/1     Running            0                 20h
kube-system         kube-controller-manager-demo-control-plane 1/1     Running            43 (3h58m ago)    20h
kube-system         kube-proxy-6wh5v                           1/1     Running            0                 20h
kube-system         kube-proxy-nknzg                           1/1     Running            0                 20h
kube-system         kube-scheduler-demo-control-plane          1/1     Running            43 (3h57m ago)    20h
local-path-storage  local-path-provisioner-6b84c5c67f-6fhrr     1/1     Running            0                 20h
pod/another-test-pod created
NAME               READY   STATUS              RESTARTS      AGE
another-test-pod   0/1     ContainerCreating   0             0s
test-pod           0/1     CrashLoopBackOff    5 (18s ago)   6m24s
NAME               READY   STATUS              RESTARTS      AGE
another-test-pod   1/1     Running             0             11s
test-pod           0/1     CrashLoopBackOff    5 (29s ago)   6m35s
```

✔ 10s ⌛ 11:41:11 ⊙

Shell Setup: ZSH

Alias, Completion… in your .zshrc: revisited

```
export PATH="${KREW_ROOT:-$HOME/.krew}/bin:$PATH"
alias kubectl=kubecolor
source <(kubectl completion zsh)
complete -F __start_kubectl k
compdef kubecolor=kubectl
source <(stern --completion=zsh)
ulimit -n 2048          # kubectl opens one cnx (file) per resource
```

I need to read my logs

stern

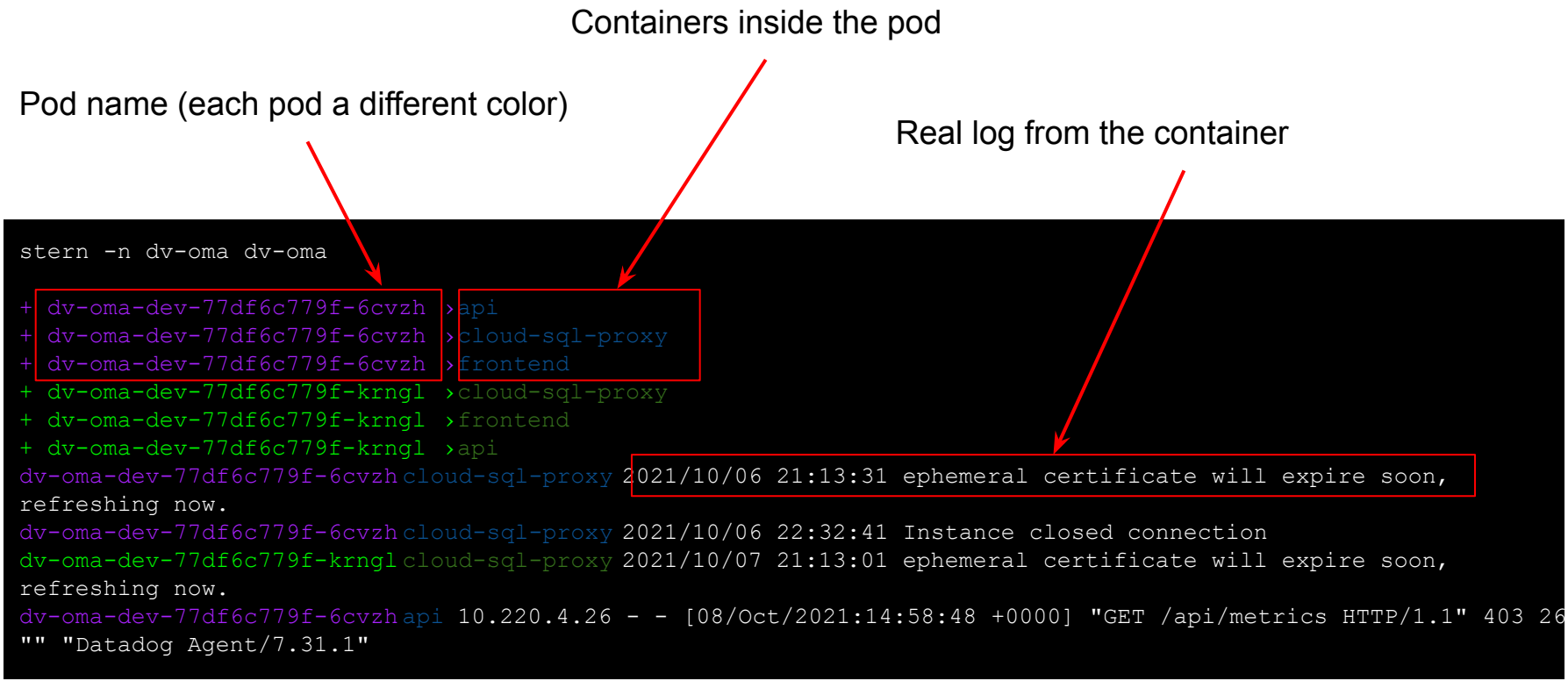
Stern (https://github.com/stern/stern)

Stern allows you to **tail multiple pods** on Kubernetes and **multiple containers** within the pod. Each result is color coded for quicker debugging.
Install:

```
brew install stern
```

```
stern -n dv-oma dv-oma

+ dv-oma-dev-77df6c779f-6cvzh ›api
+ dv-oma-dev-77df6c779f-6cvzh ›cloud-sql-proxy
+ dv-oma-dev-77df6c779f-6cvzh ›frontend
+ dv-oma-dev-77df6c779f-krngl ›cloud-sql-proxy
+ dv-oma-dev-77df6c779f-krngl ›frontend
+ dv-oma-dev-77df6c779f-krngl ›api
dv-oma-dev-77df6c779f-6cvzh cloud-sql-proxy 2021/10/06 21:13:31 ephemeral certificate will expire soon,
refreshing now.
dv-oma-dev-77df6c779f-6cvzh cloud-sql-proxy 2021/10/06 22:32:41 Instance closed connection
dv-oma-dev-77df6c779f-krngl cloud-sql-proxy 2021/10/07 21:13:01 ephemeral certificate will expire soon,
refreshing now.
dv-oma-dev-77df6c779f-6cvzh api 10.220.4.26 - - [08/Oct/2021:14:58:48 +0000] "GET /api/metrics HTTP/1.1" 403 26
"" "Datadog Agent/7.31.1"
```

Stern (https://github.com/stern/stern)



Stern (https://github.com/stern/stern)

Logs in JSON ?

```
--output json | jq '.'
```

Tail only the last 10 logs ?

```
--tail 10
```

Search pods in all namespaces ?

```
--all-namespaces
```

Search pods per labels (selectors) ?

```
-l app=prometheus
```

Stern (https://github.com/stern/stern)

New: Logs in JSON with namespace/pod/container ?

```
--all-namespaces --output extjson | jq '.'
```

New: Logs in JSON with pretty-print with colors ?

```
--output ppextjson
```

I want to extend Kubectl with new commands

Kubectl Plugins

Kubectl Plugin Manager

kubectl krew list

Krew - kubectl plugin manager

Krew is kubectl plugin manager.
Install: https://krew.sigs.k8s.io/docs/user-guide/setup/install
Adding Plugins: https://github.com/kubernetes-sigs/krew-index


```
kubectl krew list

PLUGIN  VERSION
ctx     v0.9.4
krew    v0.4.1
ns      v0.9.4
whoami  v0.0.36

kubectl krew search
NAME                            DESCRIPTION
INSTALLED
access-matrix                   Show an RBAC access matrix for server resources     no
blame                           Show who edited resource fields.                    no
cert-manager                    Manage cert-manager resources inside your cluster   no
ctx                             Switch between contexts in your kubeconfig          yes
...
```

How to create YAML manifest for my deployment?

kubectl neat

Switching Namespaces is a pain

kubectl ns

Switching namespaces with Kubeconfig context

NS

List / change Namespaces (ns)

```
# List all namespaces, current NS in yellow
kubectl ns

datadog-agents
default
kube-public
kube-system

# Set default NS by hand
kubectl config set-context --current --namespace=kube-system

# Set default Namespace
kubectl ns kube-system
```

Switching Cluster Context is a pain

kubectl ctx

Switching clusters with Kubeconfig context

CTX

List / change Cluster (context)

```
# list all the existing context, current one in yellow
kubectl ctx

arn:aws:eks:us-east-1:111111111111:cluster/eks-example
gke-dv-st-cluster-1
gke-dev_us-central1_test-gke-cluster

# change context “manually”
kubectl config use-context gke-dev_us-central1_test-gke-cluster

# change the context using ctx
k ctx gke-dev_us-central1_test-gke-cluster
```

Local Cluster: Bonus

Export the current Kube Config into another file:

```
kubectl config view --minify --raw > ~/.kube/demo.config

export KUBECONFIG=~/.kube/demo.config

k ctx
k get pods -A
```

Our favorite kubectl plugins

- whoami: who the cluster thinks you are from your authentication
  https://github.com/rajatjindal/kubectl-whoami
- who-can: RBAC rules introspection
  https://github.com/aquasecurity/kubectl-who-can
- View-secret: directly view secret content without having to decode

```
kubectl krew install ctx ns stern whoami who-can view-secret
```

I want multiple context at the same time

kubie

Kubie (https://github.com/sbstp/kubie)

Kubie offers **context** switching, **namespace** switching and **prompt** modification in a way that makes each shell independent from others.

```
# install
brew install kubie

# display a selectable menu of contexts
kubie ctx

# display a selectable menu of namespaces
kubie ns

# execute a command in all contexts matched by the wildcard and in the given namespace
kubie exec  'kind-*' kube-system kubectl get pods
```

Krew

Other cool plugins recommended:

- https://twitter.com/ahmetb/status/1563263534397419522
- get-all, images, access-matrix, resource-capacity, rbac-tool, rbac-view, rbac-lookupm, view-allocations, view-secret, view-utilization, whoami, who-can,eksporter/neat,

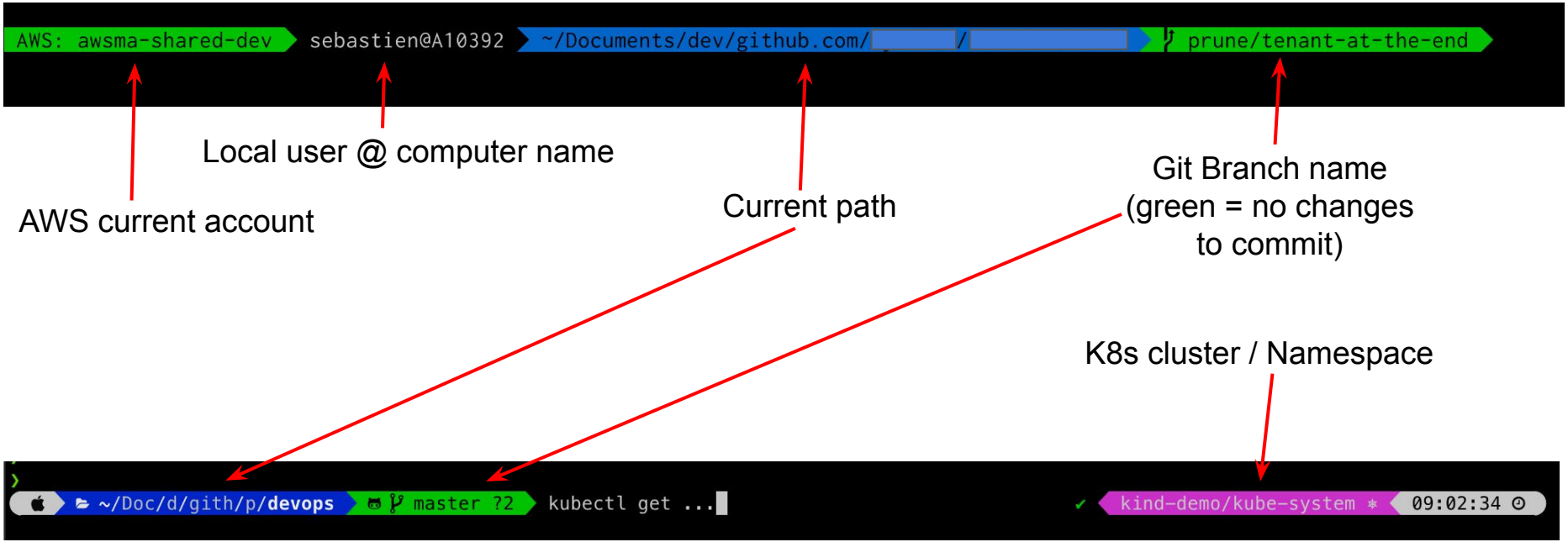
Use `kubectl krew search` to list them all !

I have too many context !

prompt

PowerLevel10k Prompt customization

Shell customization examples:



What if I’m using Cloud K8s

Cloud CLIs

Cloud Provider: Google

Install / Setup:

```
brew install --cask google-cloud-sdk
gcloud components install kubectl # Optional

gcloud init
gcloud auth login
gcloud components install gke-gcloud-auth-plugin

gcloud config set compute/region us-east1
```

Configure GKE cluster in kube config:

```
gcloud container clusters get-credentials <cluster> --project <project>
```

Cloud Provider: AWS


```
curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
sudo installer -pkg AWSCLIV2.pkg -target /
```

Setup SSO for AWS:

```
export AWS_DEFAULT_REGION=us-east-1
export AWS_PAGER="" # prevent aws cli to auto-page = display inline
export BROWSER=echo # Do not open a browser, let you choose which browser to open
complete -C '/usr/local/bin/aws_completer' aws # add that to .zshrc for completion
unset AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY
aws configure sso
aws sso login --profile profile_xxxxxx
export AWS_PROFILE=profile_xxxxxx
```

Cloud Provider: AWS

Configure EKS cluster in kube config:

```
aws eks update-kubeconfig \
    --region us-east-1    \
    --name <cluster_name> \
    --alias <friendly_name>
```

Cloud Provider: completion

Add completion for Google and AWS CLIs

```
# gcloud
source
"/usr/local/Caskroom/google-cloud-sdk/latest/google-cloud-sd
k/path.zsh.inc"
source
"/usr/local/Caskroom/google-cloud-sdk/latest/google-cloud-sd
k/completion.zsh.inc"

# AWS
complete -C '/usr/local/bin/aws_completer' aws
```

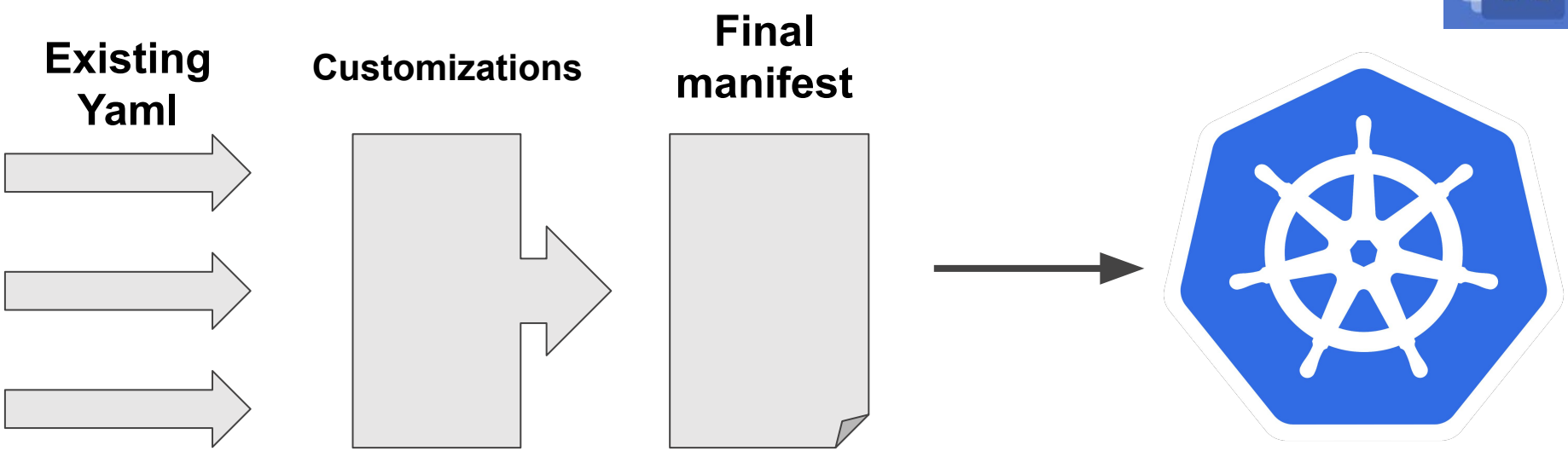
Part 3

How can I deploy my app to Dev/Stg/Prod?

Kustomize

Kustomize

Kustomize is a Kubernetes native, **template-free** way to customize application configuration



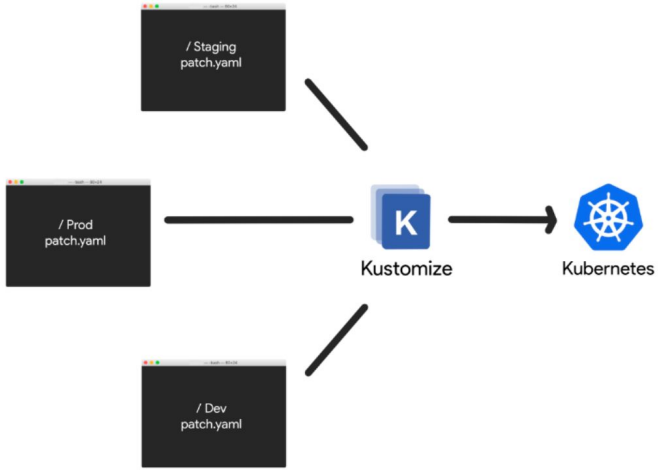
kustomize

kubectl apply

```
kustomize build .
```

How it works?

- [base] folder represents **existing Kubernetes YAML**
- [overlay] folder [dev/stg/prod]
  - **`kustomization.yaml`** `defines files that needs to be customized/patched`
  - Customizations files that needs to be overlaid on top of existing YAML files
- The output can be applied directly to k8s with kubectl



How it works?

base.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
spec:
  containers:
  - name: myapp
    image: busybox
    command: ['app']
```

patch.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
spec:
  containers:
  - name: myapp
    command: ['app', '--dev']
```

output.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
spec:
  containers:
  - name: myapp
    image: busybox
    command: ['app', '--dev']
```

```
kustomize build .
```

```
kustomize build dev/ | kubectl apply -f -
```

```
~/myapp
├── base/
│   ├── deployment.yaml
│   ├── service.yaml
│   └── kustomization.yaml
├── dev/
│   ├── deployment.yaml
│   └── kustomization.yaml
```

dev/kustomization.yaml

```
apiVersion: kustomize.config.k8s.io/v1beta1

kind: Kustomization

resources

 - ../base/

namespace: frontend

namePrefix: dev-

patchesStrategicMerge:

 - deployment.yaml
```

Kustomize Overview

System for managing and applying patches on top of existing Kubernetes YAML

Additional Features

- Add a prefix to all resource names
- Apply common:
  - Labels
  - Annotations
- Easily manage ConfigMaps and Secrets
- Apply patches (new resources)

Kustomize

Kustomize is a Kubernetes native, **template-free** way to customize application configuration

- Bundled with kubectl, but not all the features are available
- Better install the full version if you use specific features
- Only output rendered YAML, you have to apply manually

```
kubectl   kustomize --enable-alpha-plugins /path/to/kustomize/folder

kustomize build     --enable-alpha-plugins /path/to/kustomize/folder
```

Apply the resulting yaml

```
kustomize build --enable-alpha-plugins \
 /path/to/kustomize/folder | kubectl apply -f -
```

Kustomize

Kustomize requires a `kustomization.yaml` file in the target folder

```
cat /path/to/kustomize/folder/kustomization.yaml

apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

resources:
  - my_resources_file.yaml
  - ../base

patches:
  - ./manifests/my_patch.yaml

generators:
  - my_generator.yaml
```

Kustomize

Can be used to quickly template app deployment in multiple environments

```
cat /path/to/kustomize/folder/kustomization.yaml

apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

resources:
- ./../../base

namespace: dev
namePrefix: dev-
```

How can I deploy 3rd party applications?

Helm

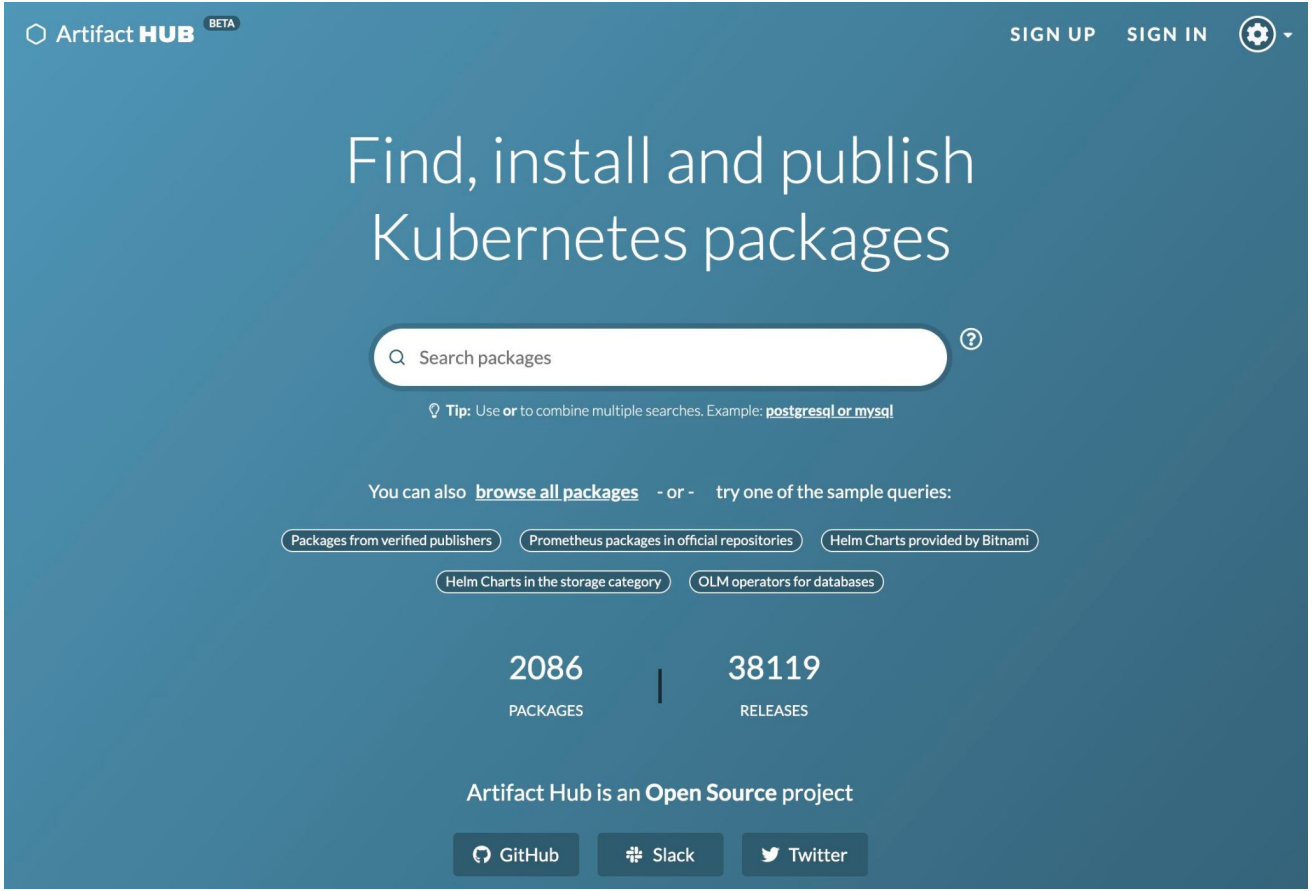
Artifact Hub - is a new home for Helm Charts

Helm is the package manager for Kubernetes.

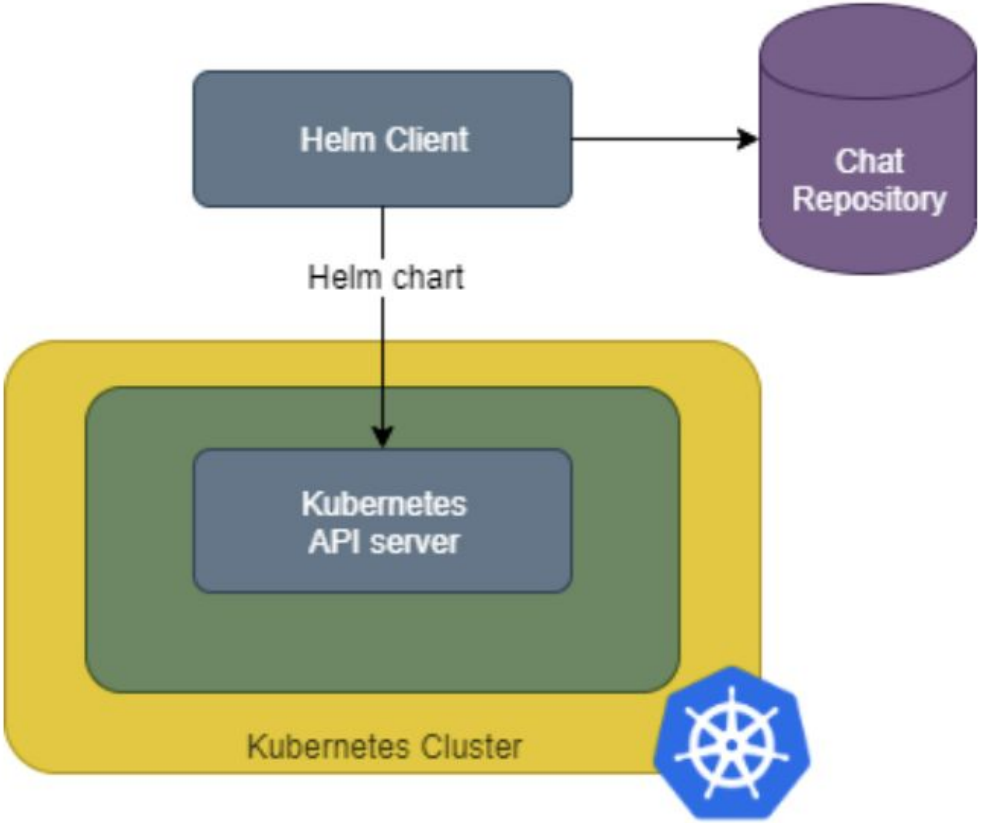
Artifact Hub - is a new home for Helm Charts

KubeCon CloudNativeCon North America 2022

https://artifacthub.io/



How it works?


Helm Client
Chat Repository
Helm chart
Kubernetes API server
Kubernetes Cluster

Helm


```
brew install helm
```

Then you can install a chart or manage remote repositories

```
helm repo add mysql-operator https://mysql.github.io/mysql-operator/
helm repo update
helm install my-mysql-operator mysql-operator/mysql-operator \
   --namespace mysql-operator --create-namespace
```

Helm 3 uses many strategies to manage deployments and rollbacks

Part 4
UIs

How can I observe what's deployed in my cluster?

K9s

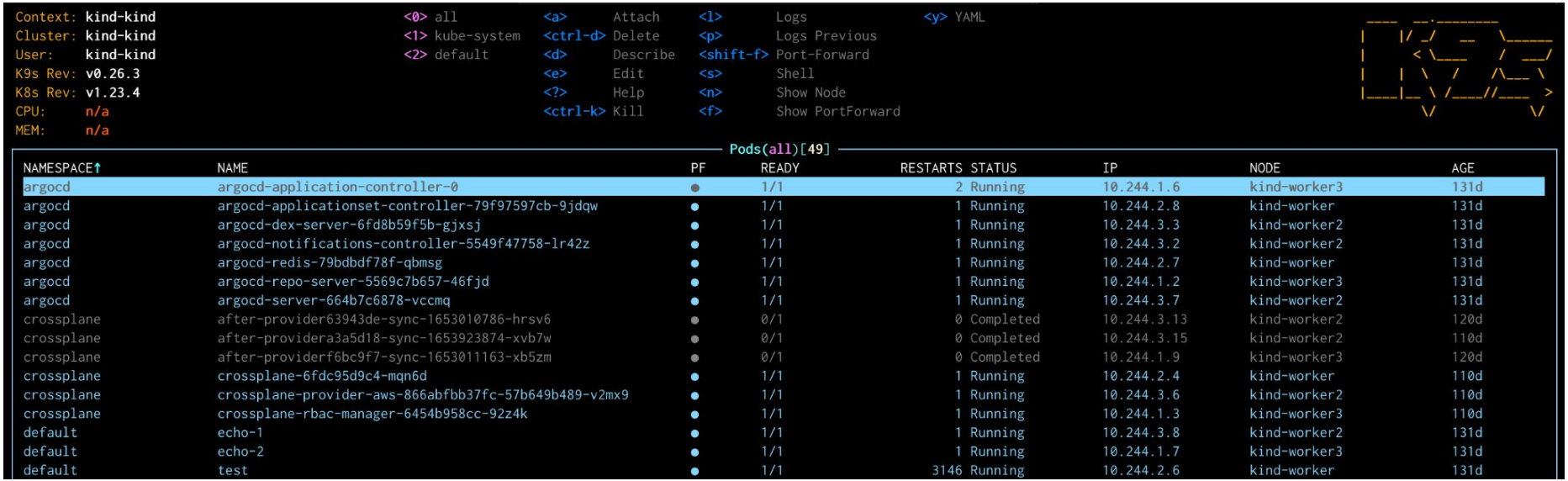
Interfaces: K9s

- K9s - Kubernetes CLI To Manage Your Clusters In Style!
- Open-Source
- In your terminal, like `top`
- CRUD operations on K8s resources

https://github.com/derailed/k9s

```
brew install k9s

k9s -n <namespace>       # To run K9s in a given namespace

k9s --context <context> # Start K9s in an existing KubeConfig context

k9s --readonly # Start K9s in readonly mode - no editing
```

Interfaces: K9s



I love UIs

Interfaces/UI/IDE

Kubernetes IDE: Lens

LENS

https://k8slens.dev/

- Lens is the IDE for Kubernetes clusters
- Lets you inspect Kubernetes cluster and its applications.
- Standalone desktop app; no in-cluster components required
- Supports:
  - Connection to Remote clusters
  - Port-Forward
- Manages:
  - CustomResourceDefinitions (CRDs)
  - Helm chart deployment inspection

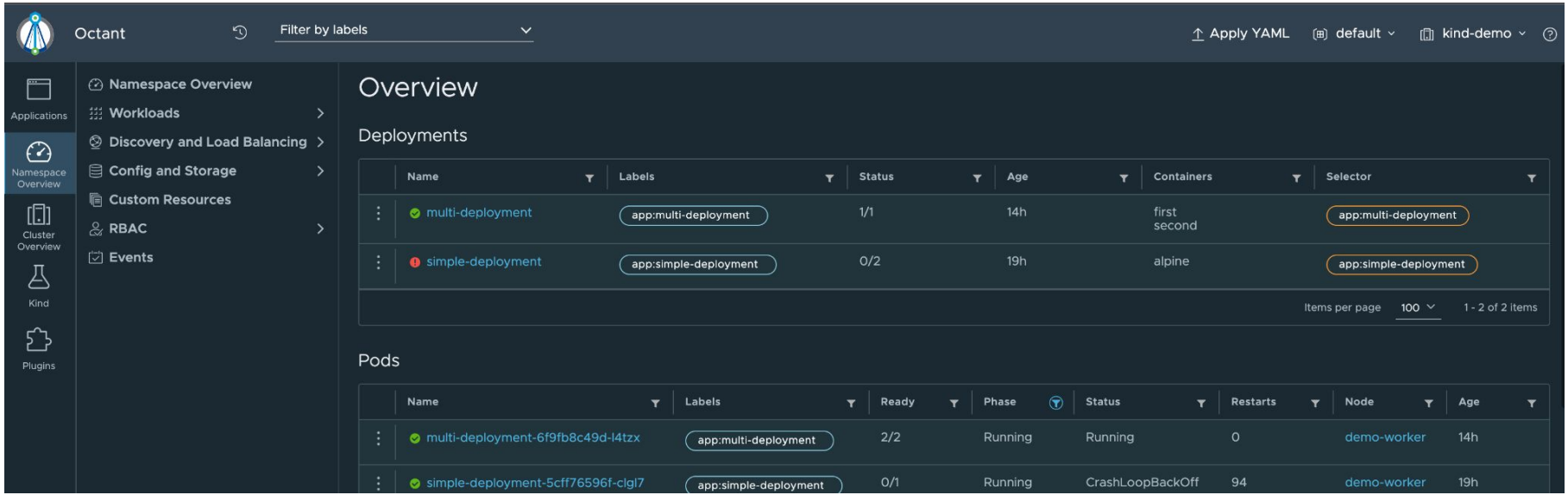
Interfaces: Octant

- Do-it-all UI
- Local “server” app that expose a local webserver for your browser
- Includes plugins to extend capabilities
- Not under active development anymore…

https://reference.octant.dev/

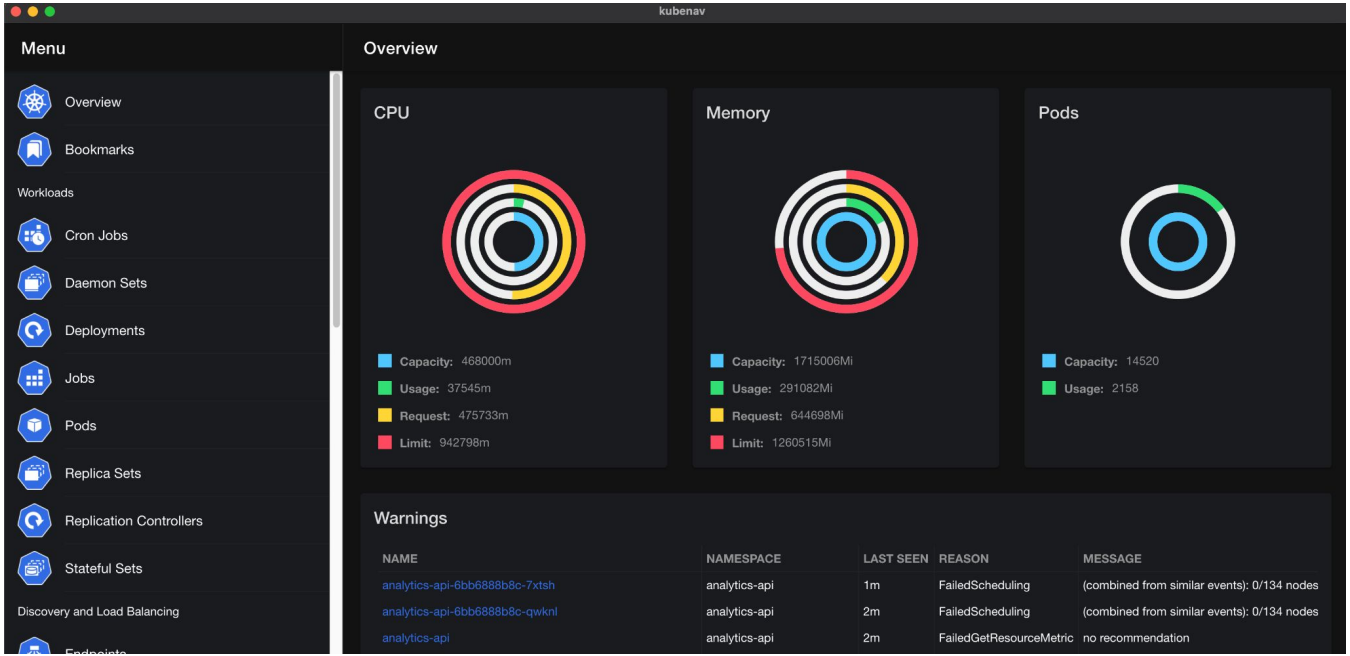
```
brew install octant
mkdir -p  $HOME/.config/octant/plugins

octant
```

Interfaces: Octant



Interfaces: KubeNav

- Iphone and Android app
- Desktop app



Working with YAML is so hard

VsCode Extensions

VsCode extensions

- Kubernetes: Develop, deploy and debug Kubernetes applications
- YAML: Language Support, with built-in Kubernetes syntax support
- Indent-Rainbow: helper to better see Yaml indentations

```yaml
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: external-dns
rules:
  - apiGroups:
      - ""
    resources:
      - endpoints
      - pods
      - services
    verbs:
      - get
      - watch
      - list
  - apiGroups:
      - extensions
```

Day 2 Ops

Working with K8s Resources

- KubePug to ensure your cluster is not using deprecated resources

  Verifies the current kubernetes cluster or input files, checking whether exists objects in this deprecated API Versions, allowing the user to check before migrating

\

I need for tooling for my advanced usage

Other tooling

Working with K8s Resources

- Dive to inspect the Docker Images
  Ensure your Docker (container) images are not too big and does not contain unnecessary data

- Dasel to query and modify data structures using selector strings
  Comparable to jq / yq, but supports JSON, YAML, TOML, XML and CSV with zero runtime dependencies

Dasel to replace JQ


```
# Pretty Print JSON
echo '[{"name": "Tom"},{"name": "Paul"}]' | dasel -p json
[
  {
    "name": "Tom"
  },
  {
    "name": "Paul"
  }
]

# JSON to YAML
echo '[{"name": "Tom"},{"name": "Paul"}]' | dasel -r json -w yaml
- name: Tom
- name: Paul
```

Dasel for Kubernetes


```
# Select the image for a container named auth
dasel -p yaml select -f deployment.yaml -s
"spec.template.spec.containers.(name=manager).image"
alpine:latest

# Change the image for a container named auth
dasel put string -f deployment.yaml -s
"spec.template.spec.containers.(name=manager).image" "ubuntu:latest"

# Update replicas to 3
dasel put int -f deployment.yaml -s "spec.replicas" 3
```

Dasel for Kubernetes


```
# Add a new env var
dasel put object -f deployment.yaml -s
"spec.template.spec.containers.(name=manager).env.[]" -t string -t
string name=MY_NEW_ENV_VAR value=MY_NEW_VALUE

# Update an existing env var
dasel put string -f deployment.yaml -s
"spec.template.spec.containers.(name=manager).env.(name=MY_NEW_ENV_VAR)
.value" NEW_VALUE
```

References

- Install Kubectl: https://kubernetes.io/docs/tasks/tools/
- **K8s Shell**
  - **Kubectl tuning**: https://suraj.io/post/being-productive-with-kubectl/
  - **Kubie**: switch k8s context per shell https://github.com/sbstp/kubie
  - Krew: https://github.com/kubernetes-sigs/krew
  - Kubecolor: https://github.com/kubecolor/kubecolor
  - Stern: https://github.com/stern/stern
  - NerdFonts: https://www.nerdfonts.com/
  - OhMyZsh: https://ohmyz.sh/
  - Agnoster: https://github.com/agnoster/agnoster-zsh-theme
  - PowerLevel10k: https://github.com/romkatv/powerlevel10k
  - Node-shell: https://github.com/kvaps/kubectl-node-shell
- **K8s app Configuration Editing**
  - Kustomize: https://kustomize.io/
  - Kpt: https://kpt.dev/

References

- **Local K8s Clusters:**
  - Kind: https://kind.sigs.k8s.io
  - K3s: https://k3s.io/
  - Minikube: https://minikube.sigs.k8s.io/docs/start/
- **Kubernetes UIs:**
  - K9s: https://github.com/derailed/k9s
  - Mirantis Lens: https://k8slens.dev/
  - Infra App: https://infra.app/
  - Kubenav: https://kubenav.io/
  - Octant: https://github.com/vmware-tanzu/octant
- **Cool tools:**
  - KubePug: https://github.com/rikatz/kubepug
  - Dive: https://github.com/wagoodman/dive
  - Dasel: https://github.com/TomWright/dasel
- **Containerd ecosystem (Build, Run, Scan):**
  - Colima: https://github.com/abiosoft/colima
  - Podman: https://podman.io/

Epilogue

KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Please scan the QR Code above
to leave feedback on this session


KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM