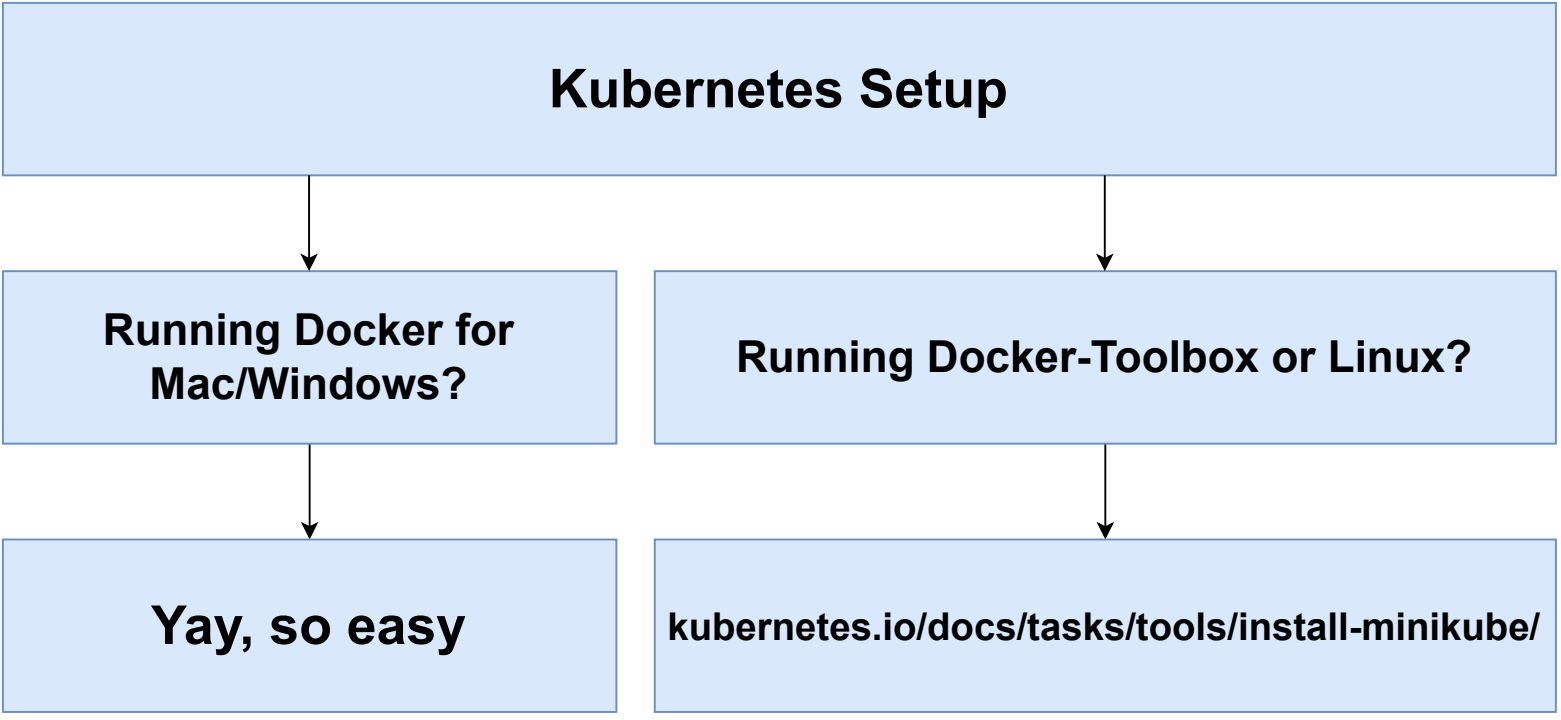
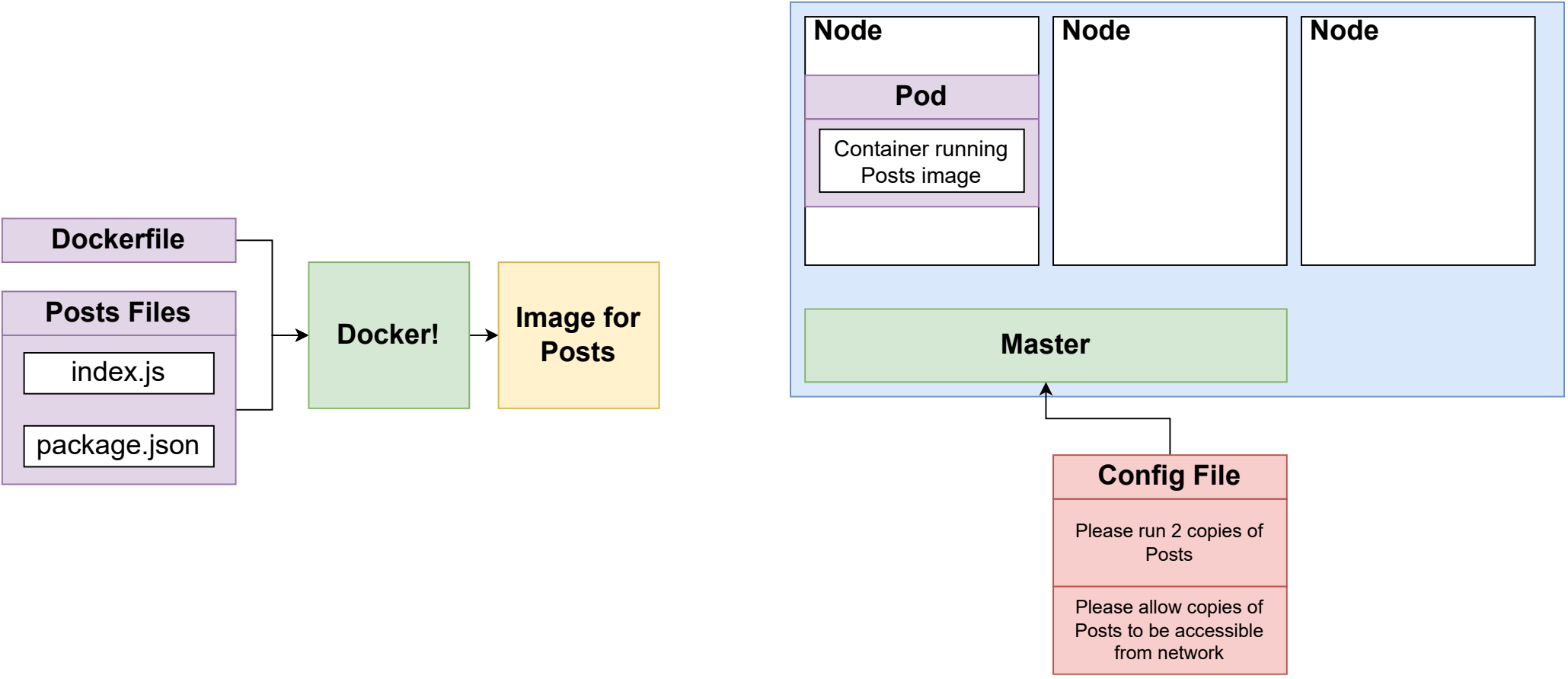
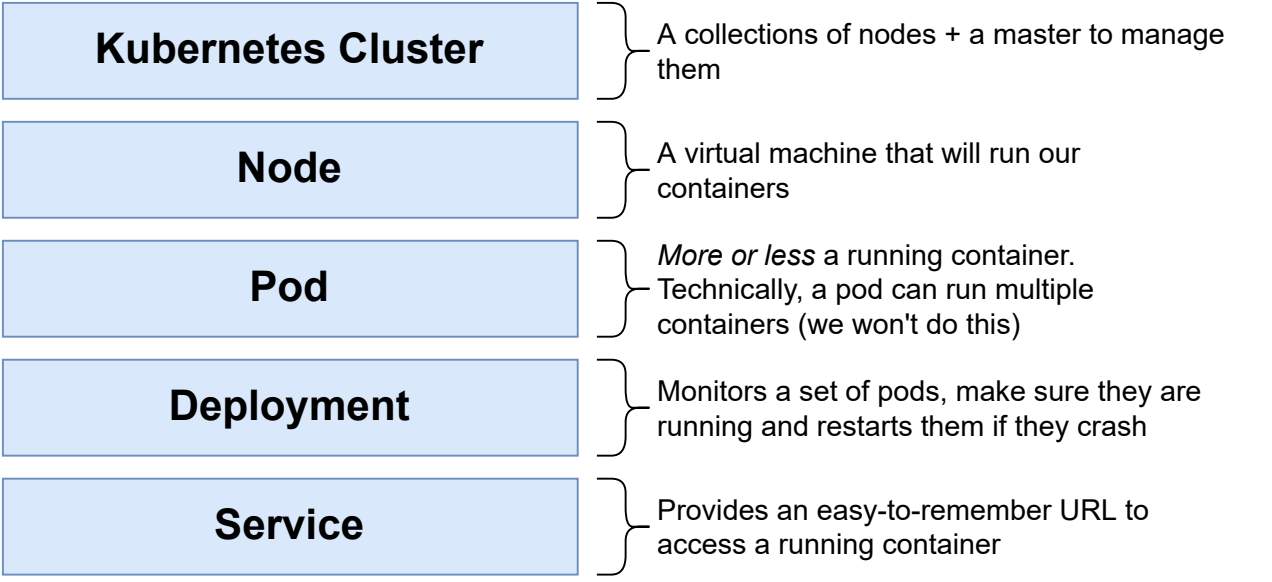

Kubernetes Setup
Running Docker for Mac/Windows?
Running Docker-Toolbox or Linux?
Yay, so easy
kubernetes.io/docs/tasks/tools/install-minikube/

Whirlwind Tour of Kubernetes

I do not expect you to
memorize all/any of this


Dockerfile
Posts Files
index.js
package.json
Docker!
Image for Posts
Node
Pod
Container running Posts image
Node
Node
Master
Config File
Please run 2 copies of Posts
Please allow copies of Posts to be accessible from network


Kubernetes Cluster
A collections of nodes + a master to manage them
Node
A virtual machine that will run our containers
Pod
More or less a running container. Technically, a pod can run multiple containers (we won't do this)
Deployment
Monitors a set of pods, make sure they are running and restarts them if they crash
Service
Provides an easy-to-remember URL to access a running container

Kubernetes Config Files

Tells Kubernetes about the different Deployments, Pods, and Services (referred to as 'Objects') that we want to create

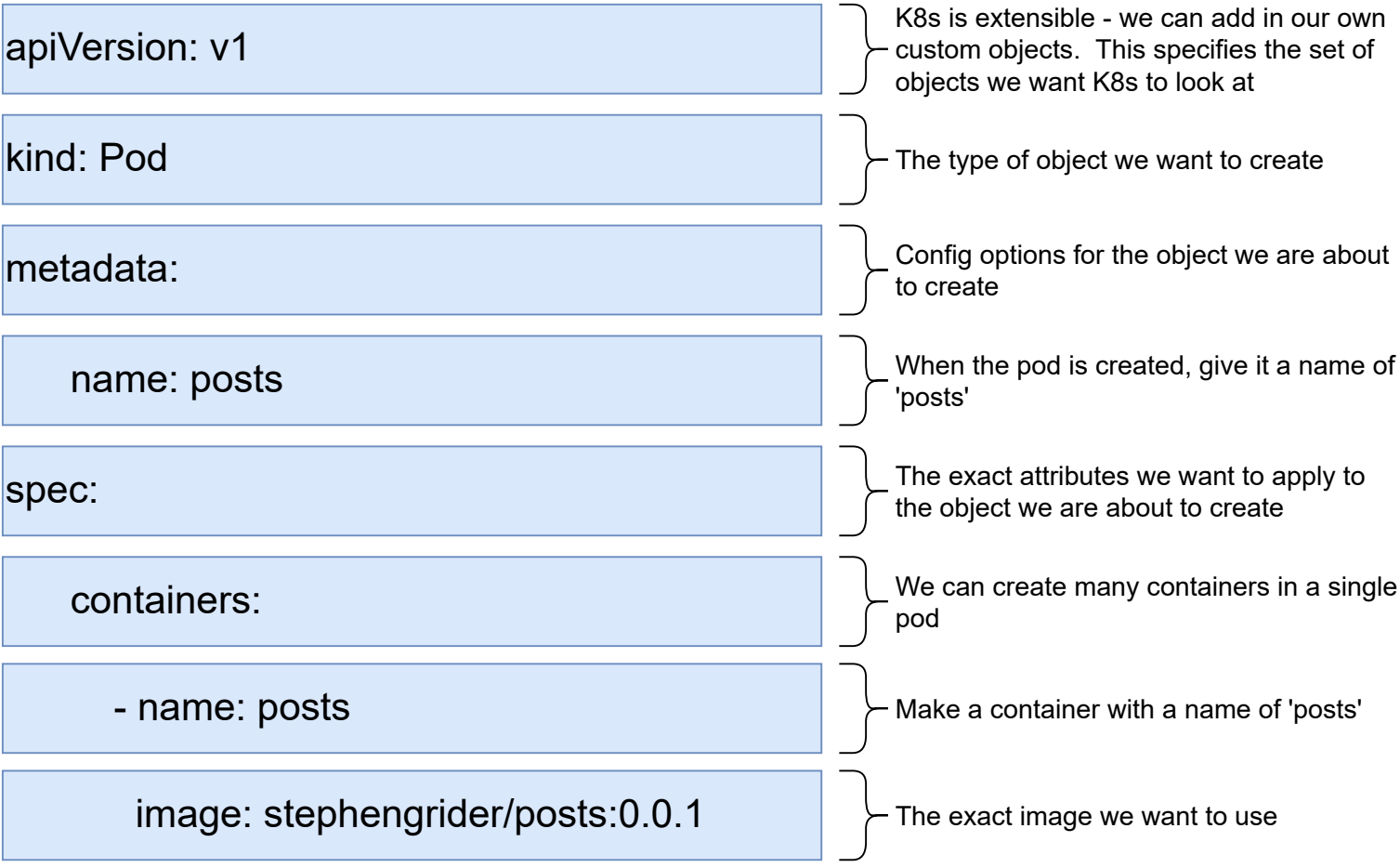
Written in YAML syntax

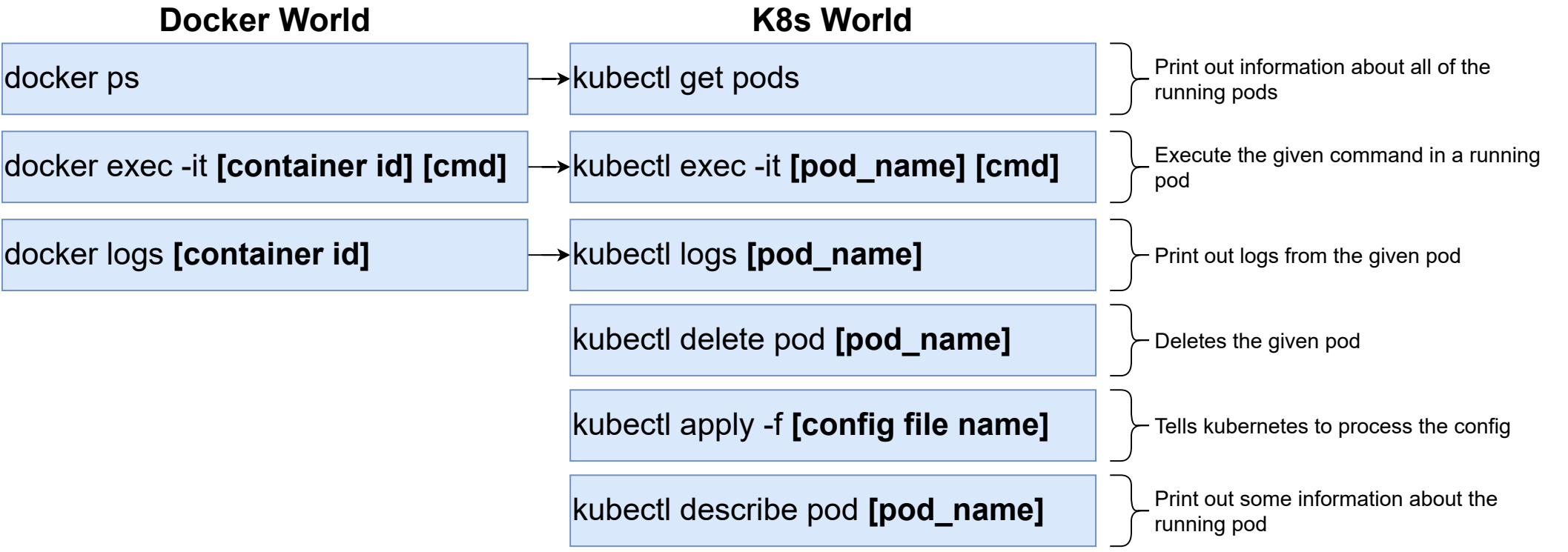
Always store these files with our project source code - they are documentation!

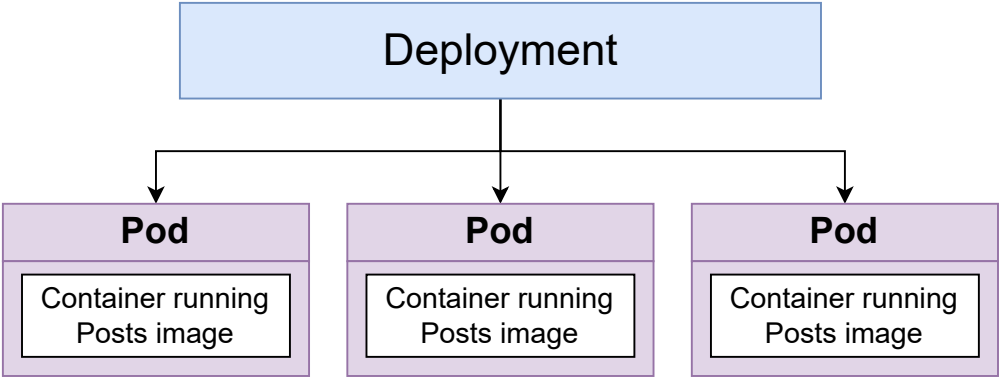
We can create Objects *without* config files - ***do not do this***. Config files provide a precise definition of what your cluster is running.

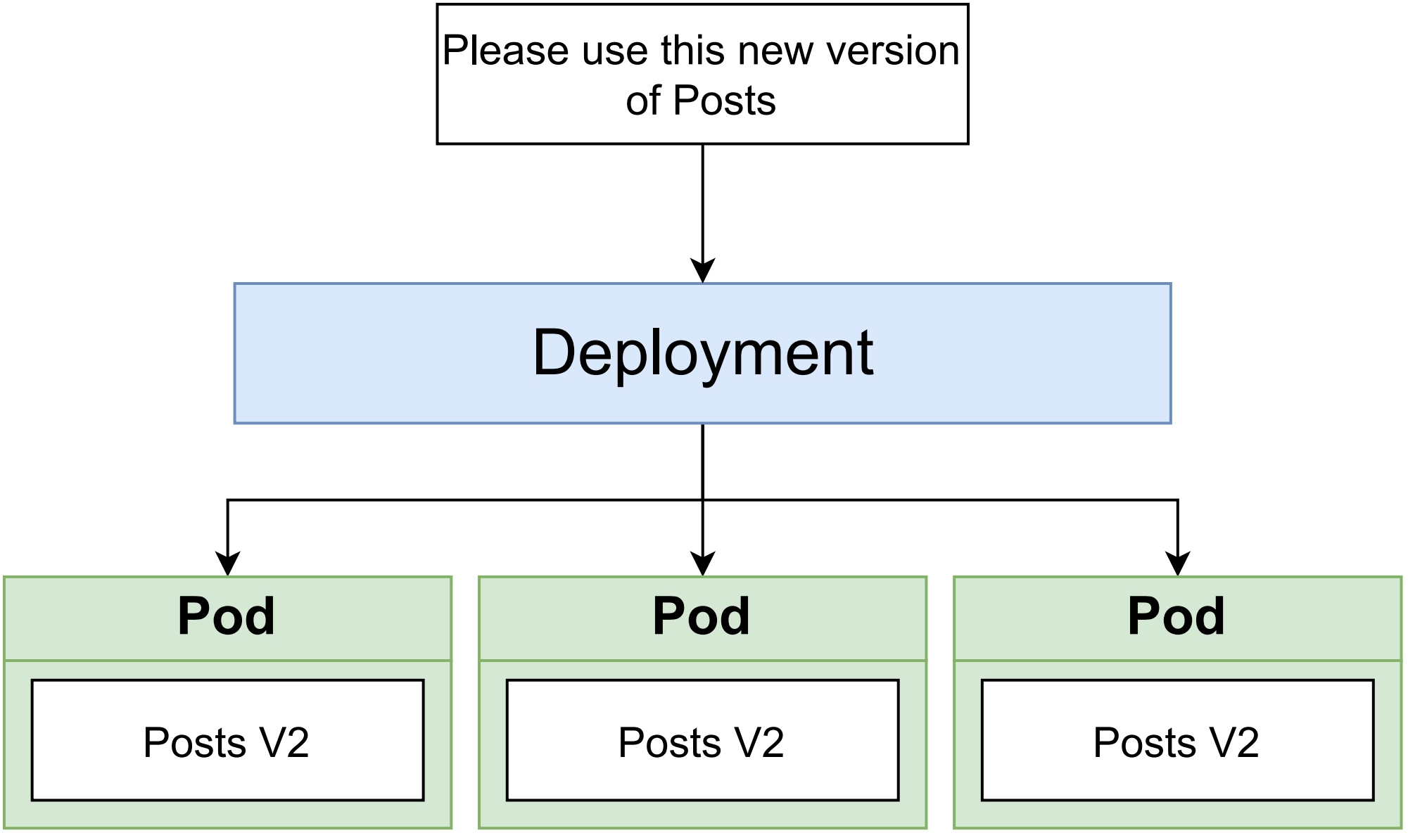
Kubernetes docs will tell you to run direct commands to create objects - *only do this for testing purposes*

Blog posts will tell you to run direct commands to create objects - *close the blog post!*


apiVersion: v1
K8s is extensible - we can add in our own custom objects. This specifies the set of objects we want K8s to look at
kind: Pod
The type of object we want to create
metadata:
Config options for the object we are about to create
name: posts
When the pod is created, give it a name of 'posts'
spec:
The exact attributes we want to apply to the object we are about to create
containers:
We can create many containers in a single pod
- name: posts
Make a container with a name of 'posts'
image: stephengrider/posts:0.0.1
The exact image we want to use


Docker World
K8s World
docker ps
kubectl get pods
Print out information about all of the running pods
docker exec -it [container id] [cmd]
kubectl exec -it [pod_name] [cmd]
Execute the given command in a running pod
docker logs [container id]
kubectl logs [pod_name]
Print out logs from the given pod
kubectl delete pod [pod_name]
Deletes the given pod
kubectl apply -f [config file name]
Tells kubernetes to process the config
kubectl describe pod [pod_name]
Print out some information about the running pod


Deployment
Pod
Container running Posts image
Pod
Container running Posts image
Pod
Container running Posts image


Please use this new version of Posts
Deployment
Pod
Posts V2
Pod
Posts V2
Pod
Posts V2

Deployment Commands

`kubectl get deployments`	List all the running deployments
`kubectl describe deployment` **[depl name]**	Print out details about a specific deployment
`kubectl apply -f` **[config file name]**	Create a deployment out of a config file
`kubectl delete deployment` **[depl_name]**	Delete a deployment
`kubectl rollout restart deployment` **[depl_name]**	Get a deployment to restart all pods. Will use latest version of an image *if* the pod spec has a tag of 'latest'

Updating the Image Used By a Deployment - Method #1

Steps

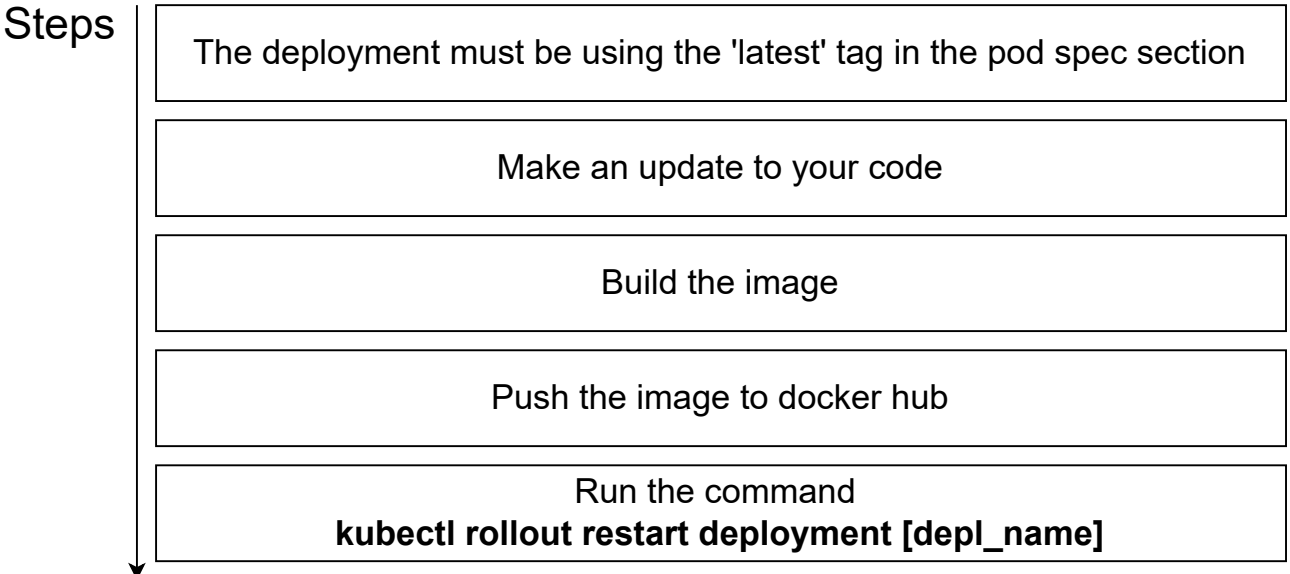
Make a change to your project code

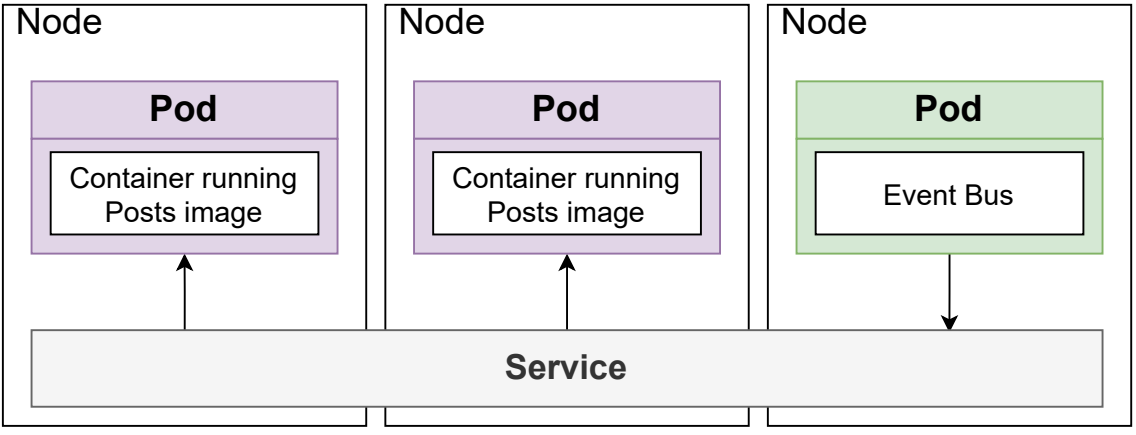
Rebuild the image, specifying a new image version

In the deployment config file, update the version of the image

Run the command
kubectl apply -f [depl file name]

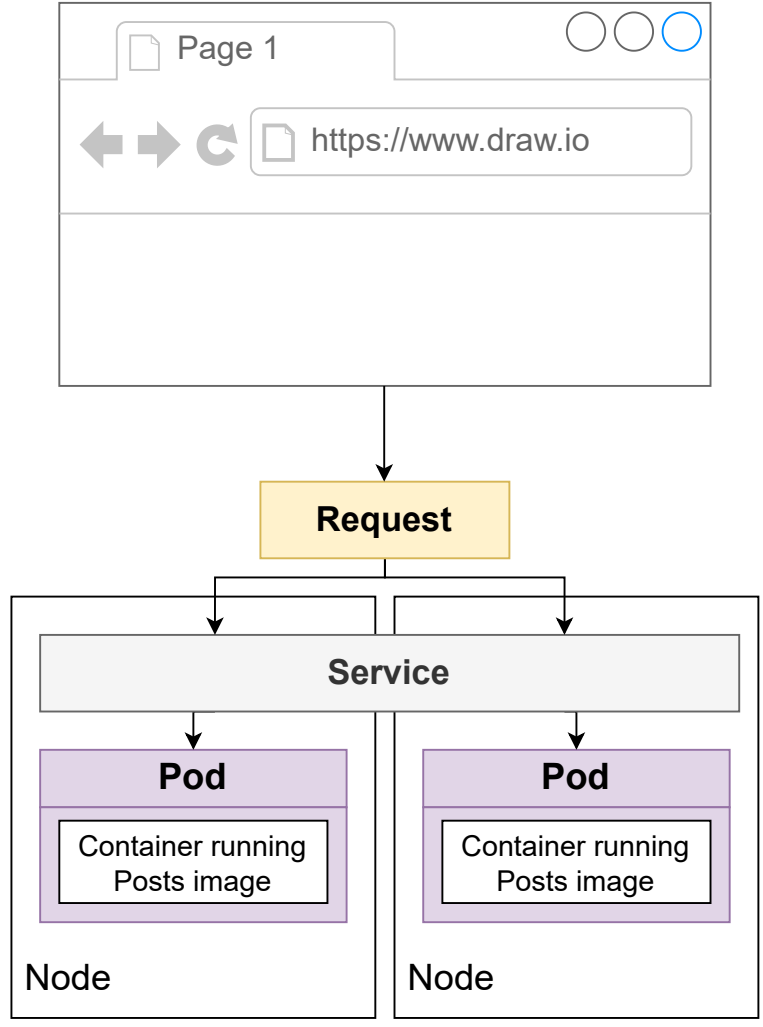
Updating the Image Used By a Deployment - Method #2



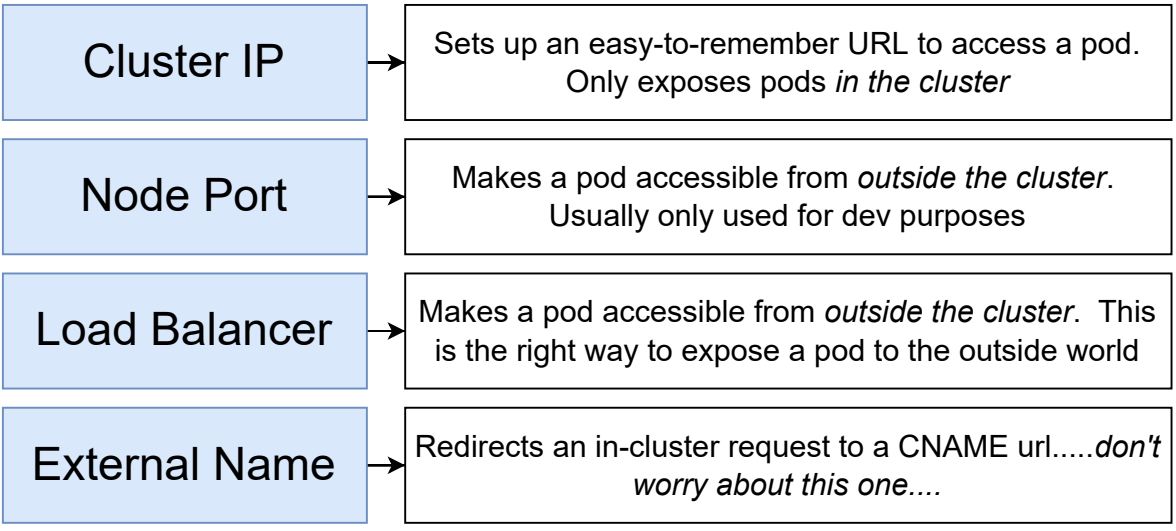


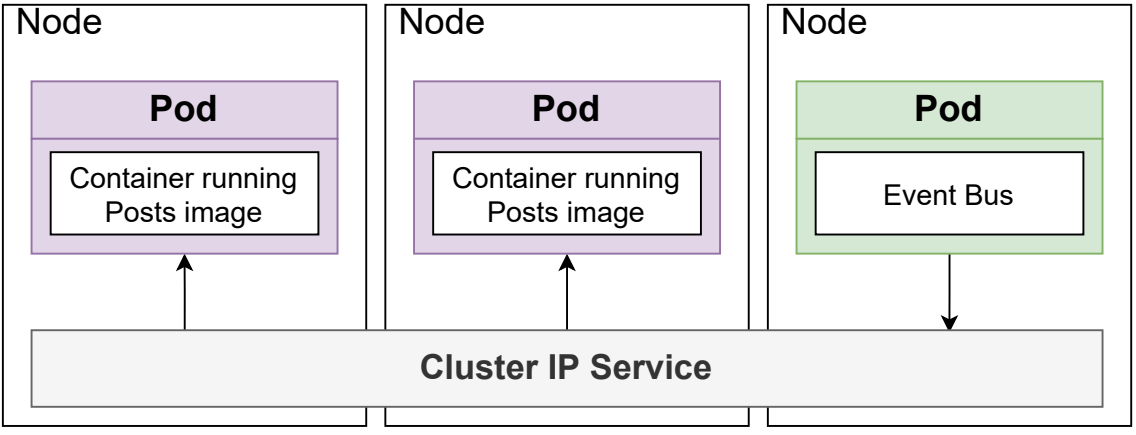
Services provide networking *between* pods....

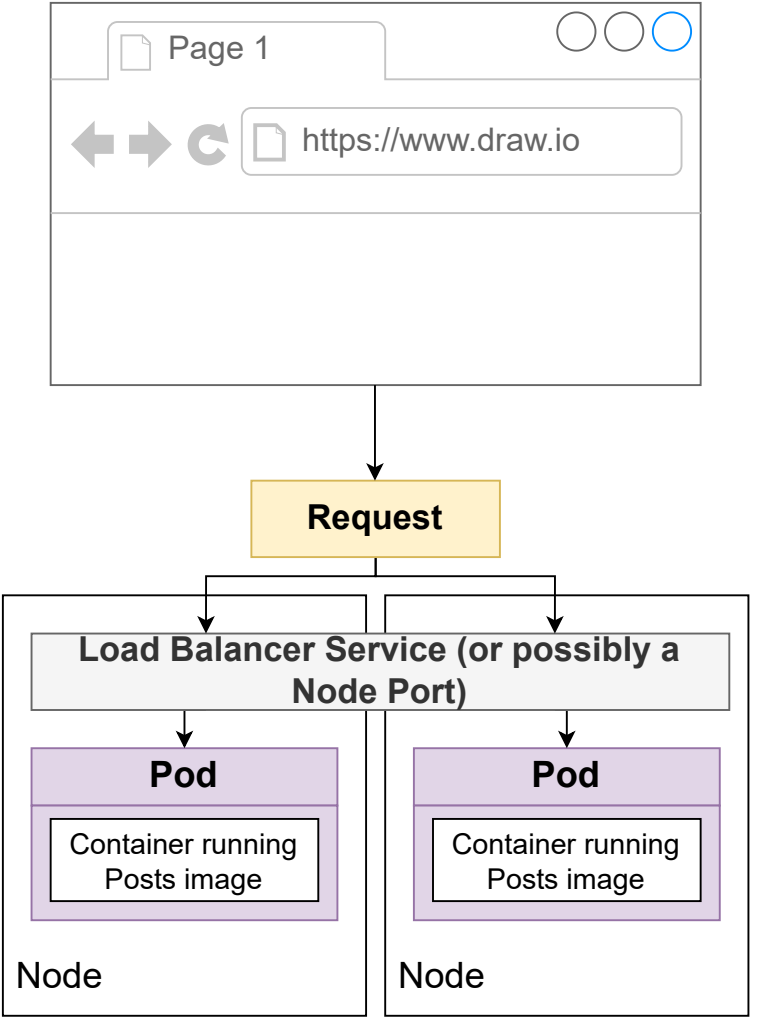
....and from the outside world to a pod

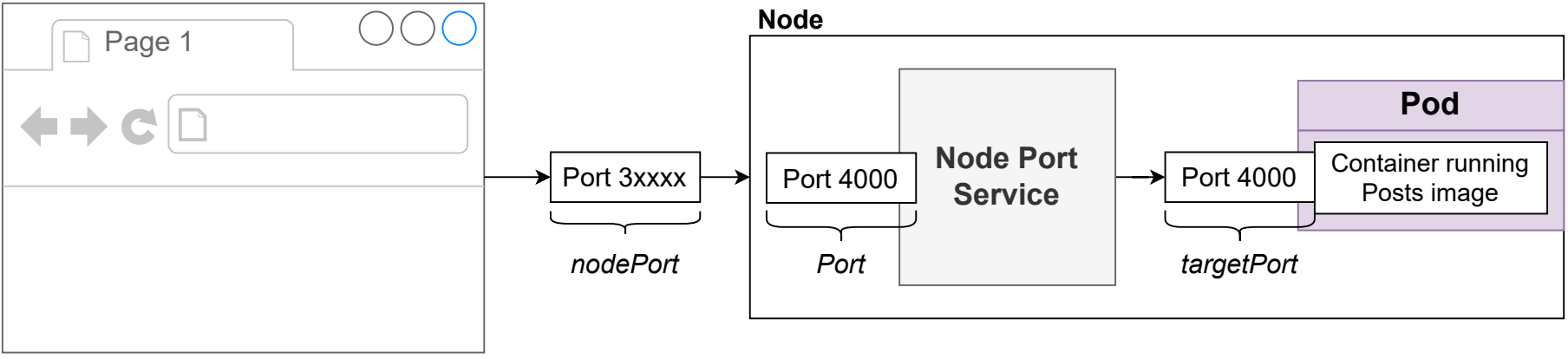


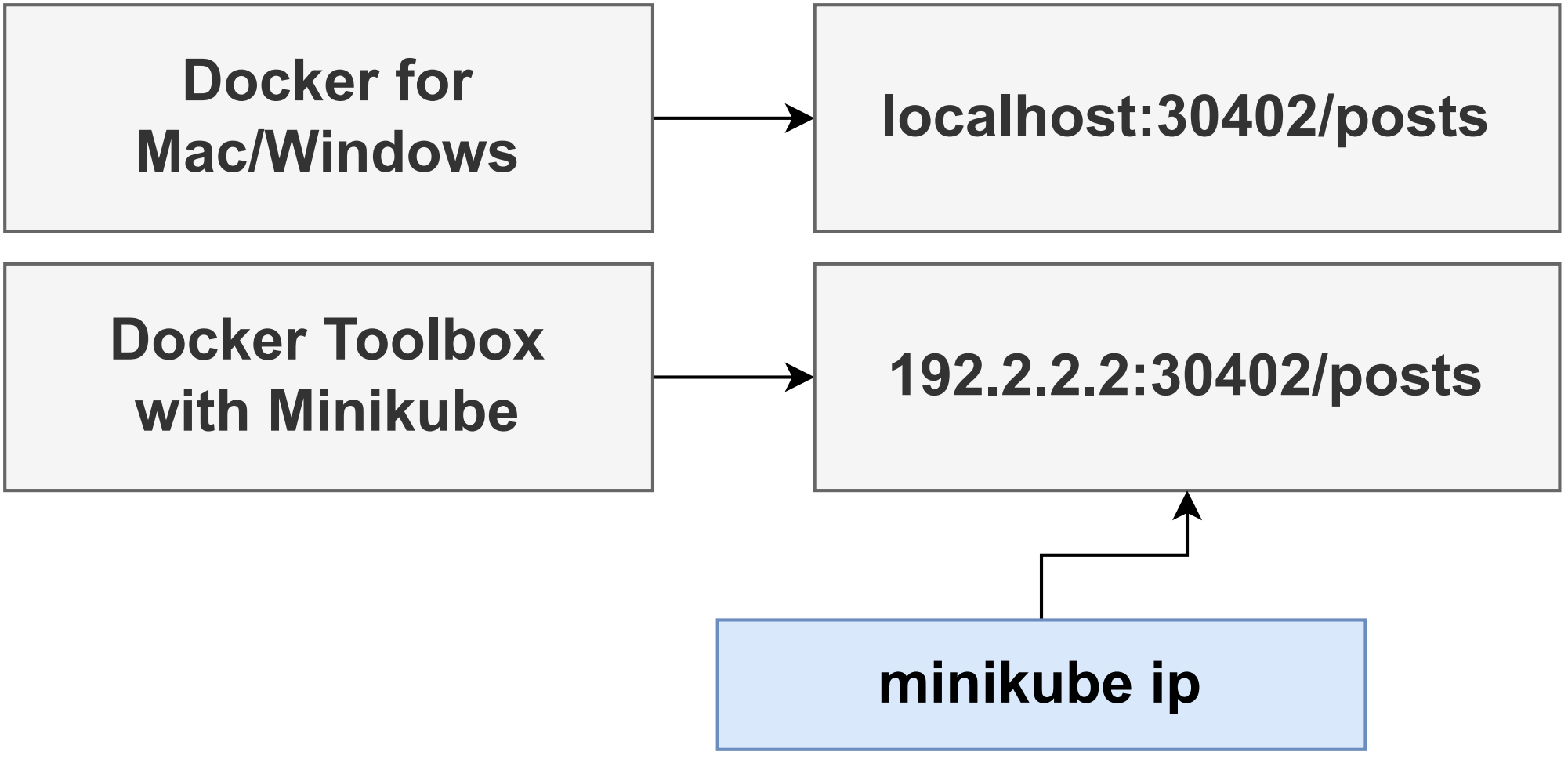
Types of Services




Node
Node
Node
Pod
Container running
Posts image
Pod
Container running
Posts image
Pod
Event Bus
Cluster IP Service


Page 1
https://www.draw.io
Request
Load Balancer Service (or possibly a Node Port)
Pod
Container running Posts image
Pod
Container running Posts image
Node
Node


Page 1
Port 3xxxx
nodePort
Node
Port 4000
Port
Node Port Service
Pod
Port 4000
targetPort
Container running Posts image


Docker for
Mac/Windows
localhost:30402/posts
Docker Toolbox
with Minikube
192.2.2.2:30402/posts
minikube ip

Goals Moving Forward

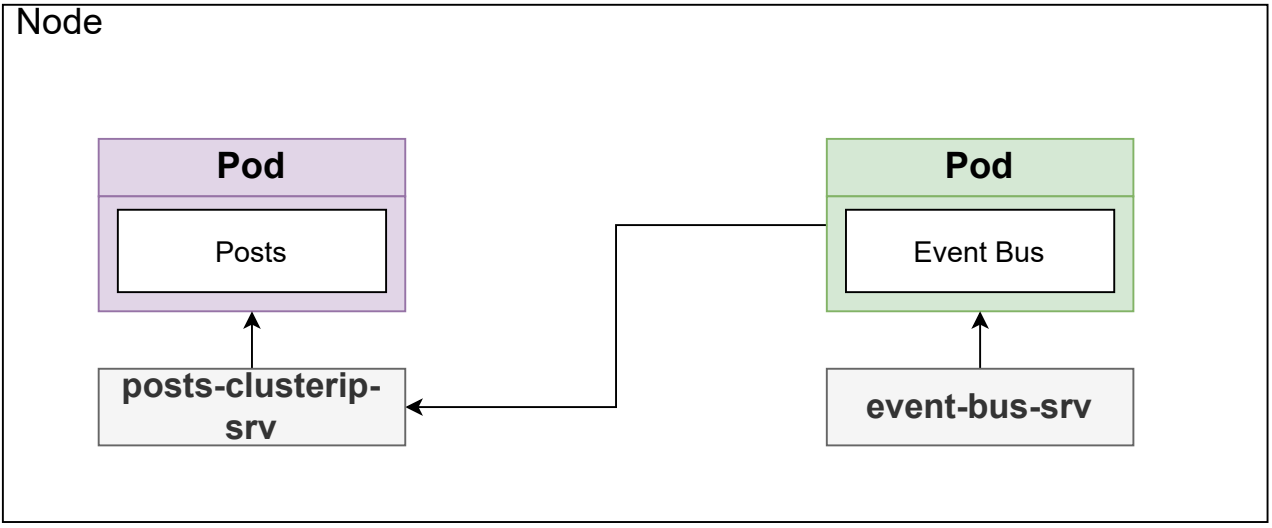
Build an **image** for the Event Bus

Push the image to Docker Hub

Create a **deployment** for Event Bus

Create a **Cluster IP service** for Event Bus and Posts

Wire it all up!


Node
Pod
Posts
Pod
Event Bus
posts-clusterip-srv
event-bus-srv

Adding More Services

For 'comments', 'query', 'moderation'....

Update the URL's in each to reach out to the 'event-bus-srv'

Build images + push them to docker hub

Create a deployment + clusterip service for each

Update the event-bus to once again send events to 'comments', 'query', and 'moderation'