Deploy our First POD

Step 0: the basic YAML

myFist-ksPod

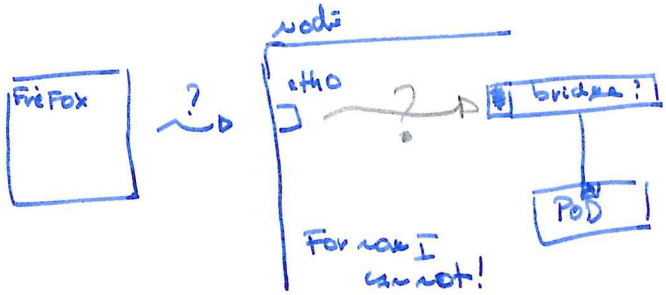
obs:
- Labels, are needed in order to allow for selection later on
- Ports, Port 8080 is dictated by the application the only thing is that we are exposing it

[this operation is more "informative" than useful, as a matter of fact we don't need it!]

Deploying a POD

→ kaf main.yaml
k get pods
k delete pod /podname
kdf main.yaml ⚠ Delete object with a given name.

Looking @ what is inside our pod



Trick for debugging kubectl port-forward PodName XX:XX --address if on remote host

→ Later we will try to open a better channel!

Connecting (Attaching) to a pod

k exec -it podname -- bash

other parameters

my seconDks pod!!

resources

- request: if not met ⇒ not scheduled in pod node
- Limits: Max resources usable by container! → Malloc will trigger OOM killer!

$\sum$ Pods_requested resources ≤ node resources!

⇒ use the kernel interface to trigger OOM killer!

Probes

we saw them before, they are regular Http request!

curl ip:port/healthy

→ if we fail healthprobe for a series of time the Podget's automatically rebooted

→ if we fail the ready probe.... nothig happens :-<

> our traffic still works because we used a trick!

↳ the POD stays up & runnig!!

> Typical Selector:
>
> some obs consume other obs. they choose which obs to use by the mean of SELECTOR.
>
> EXAMPLE: a pod can use the nodeSelector: (part of SPEC)
> nodePartition: prod
> to decide where to get scheduled
> arbitrary key — arbitrary value.

LABELS

key = value pairs

attached to obsects

used by the user to

- Select
- Organize obsect

key structure

prefix/name

if no prefix: the key is Private

Private keys:
- release: < release canary
- environment: < dev, prod

You can select them by using complex structures obviously like

keyA in Values & keyB not in Values

OLD API

```
selector:
  a:b
  c:d
```

→ new api looks like

```
selector:
  matchLabels:
    a: b
    c: d
  matchExpressions:
    complex expression
```

ANNOTATIONS

Similar to Labels. But for external tools & Libraries.