Certified Kubernetes Application Developer (CKAD)

D2IQ

MESOSPHERE

Ken Sipe

Distribute Application Engineer

Apache Mesos Contributor
Apache Committer Myriad, Open DCOS
Developer: Embedded, C++, Java, Groovy, Grails, C#, GoLang

@KenSipe

Agenda

- CKAD - Expectation
- Setting Up
- Core Concepts
- Configuration
- Pod Design
- Services / Networking
- Persistence

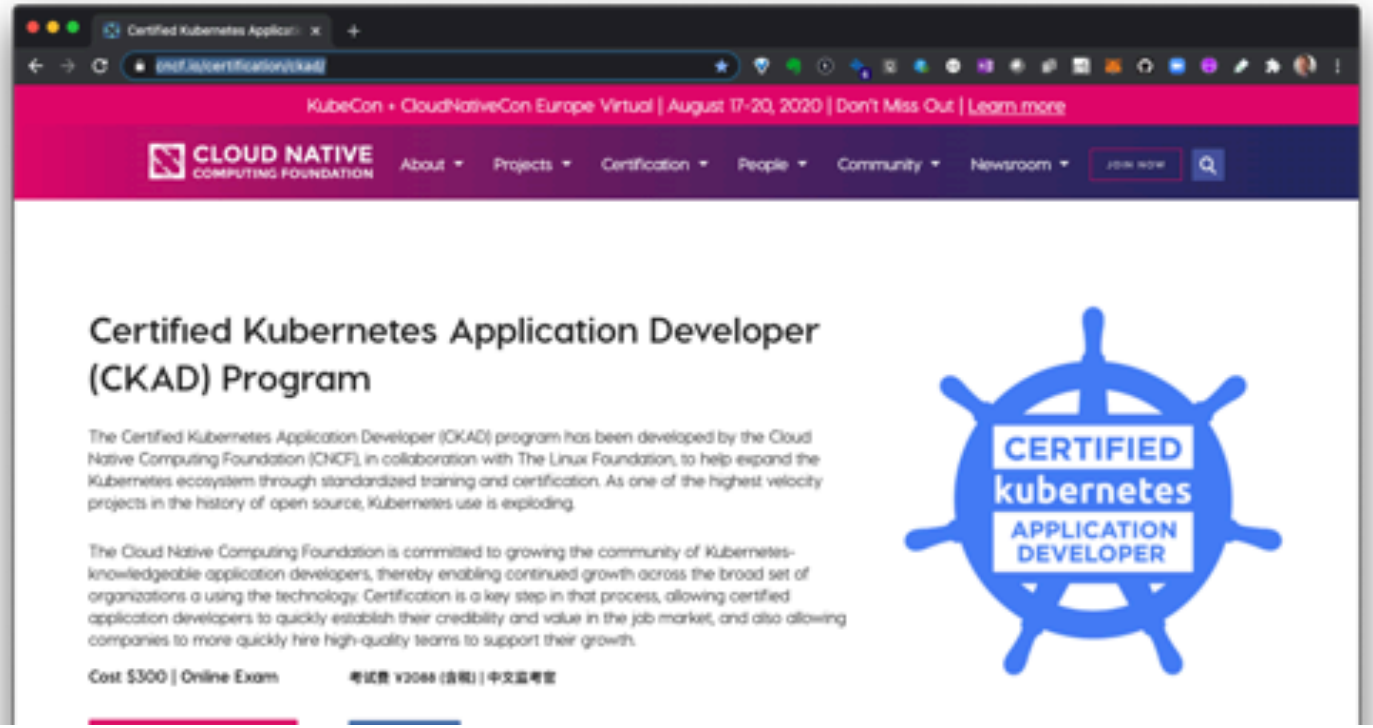
CKAD Expectations

Expectations

- Number of Questions
- Time per Question
- Desk / Proxy
- Cost / 2x
- Topics / percents
- Signing up (links)

Sign up

https://www.cncf.io/certification/ckad/



Cost

- $300
  - Includes 1 Retest
- 2 hours

Time

- 2 hours
- 20 questions (weighted)

- **6 mins / Question**

Taking Test

- Remote Proxy
  - Video scan desk area
  - Closed uninterruptible space
  - No-one in the area
  - Microphone, Video and full view of all screens
  - Clear Desktop
- Reliable Internet

Browser

- Current version of Chrome or Chromium
- Limited Access
  - Test
  - kubernetes.io
  - No Google!
- Search is on kubernetes.io

Browser

- Know your weakness

Examples using `kubectl` and JSONPath expressions:

```
ctl get pods -o json
ctl get pods -o=jsonpath='{@}'
ctl get pods -o=jsonpath='{.items[0]}'
ctl get pods -o=jsonpath='{.items[0].metadata.name}'
ctl get pods -o=jsonpath="{.items[*]['metadata.name', 'status.capacity']}"
ctl get pods -o=jsonpath='{range .items[*]}{.metadata.name}{"\t"}{.status.s
```

https://kubernetes.io/docs/reference/kubectl/jsonpath/

https://kubernetes.io/docs/reference/kubectl/cheatsheet/

13% - Core Concepts

- Understand Kubernetes API primitives
- Create and configure basic Pods

8% - State Persistence

- Understand PersistentVolumeClaims for storage

20% - Pod Design

- Understand Deployments and how to perform rolling updates
- Understand Deployments and how to perform rollbacks
- Understand Jobs and CronJobs
- Understand how to use Labels, Selectors, and Annotations

18% - Observability

- Understand LivenessProbes and ReadinessProbes
- Understand container logging
- Understand how to monitor applications in Kubernetes

18% - Configuration

- Understand ConfigMaps
- Understand SecurityContexts
- Define an application's resource requirements
- Create & consume Secrets
- Understand ServiceAccounts

13% - Services & Networking

- Understand Services
- Demonstrate basic understanding of NetworkPolicies

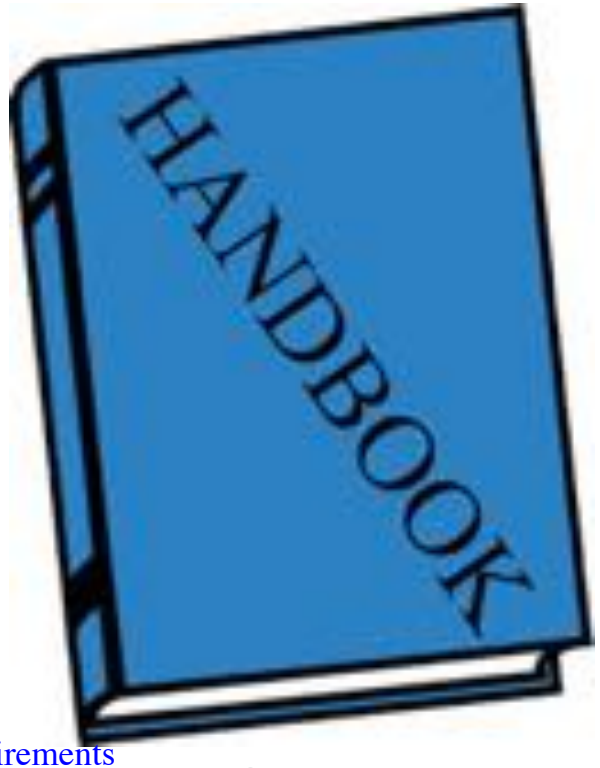
13% - Services & Networking

- Understand Services
- Demonstrate basic understanding of NetworkPolicies

10% Multi-Container Pods

- Understand Multi-Container Pod design patterns (e .g. ambassador, adapter, sidecar)

Primary ID (non-expired and including photograph and signature)	Secondary ID (non-expired and including signature with Candidate name in Latin characters)
Passport	
Government-issued driver's license/permit	Debit (ATM) Card
Government-Issued local language ID (with photo and signature)	Credit Card
	Health Insurance Card
National Identity card	U.S. Social Security Card
State or province-issued identity card	

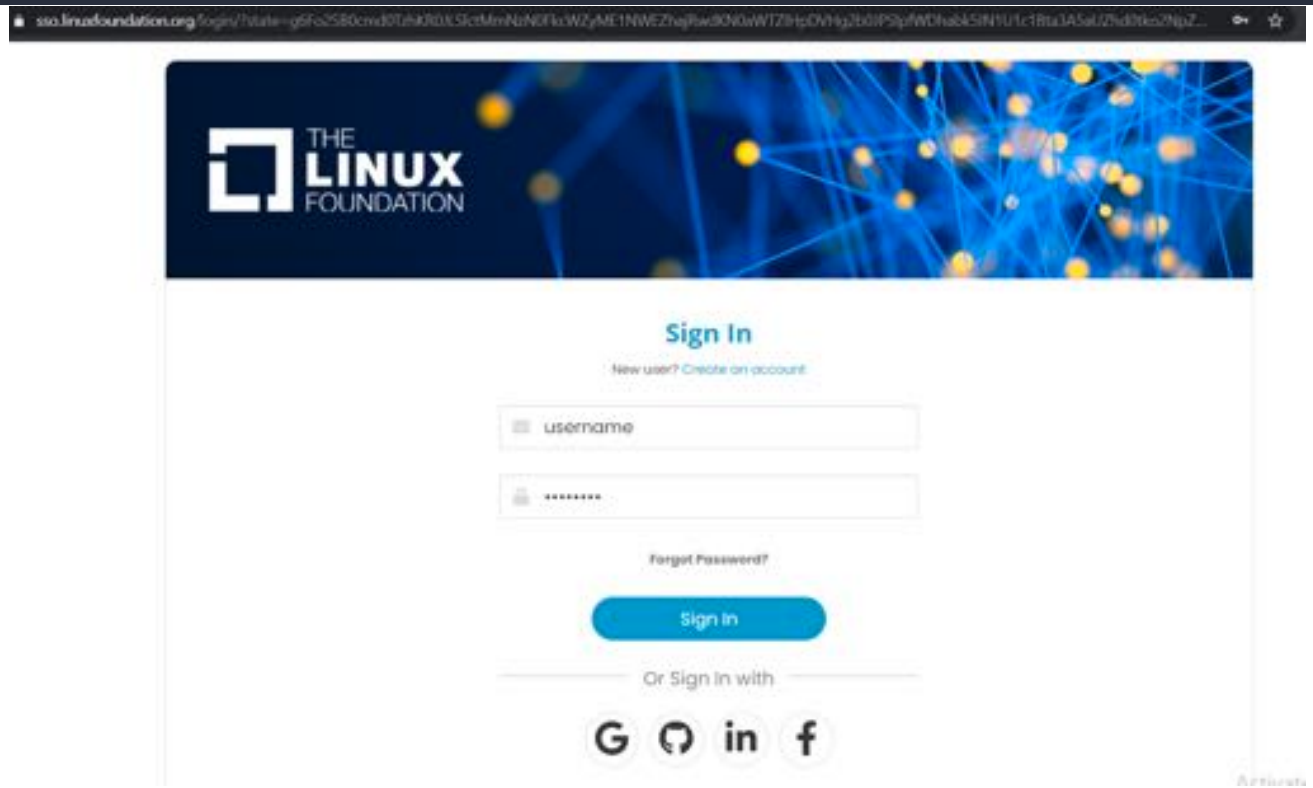


https://docs.linuxfoundation.org/tc-docs/certification/lf-candidate-handbook/candidate-requirements

Passing Test

66 / 100

Sign up and Schedule



https://trainingportal.linuxfoundation.org/

CKAD
Setting up

Expectations

- VI skills and setup
- Shell in browser
- Browser setup and bookmarks
- Time savers

VI

- Shell inside Browser
- alias vi=vim

```
cat ~/.vimrc
:set number
:set et
:set sw=2 ts=2 sts=2
```

- “number” - sets left col line nums
- “et” - expand tab
- “sw” - shiftwidth
- “ts” - tab stop
- “sts” - softtabstop

** configuration for 2 spaces for a tab

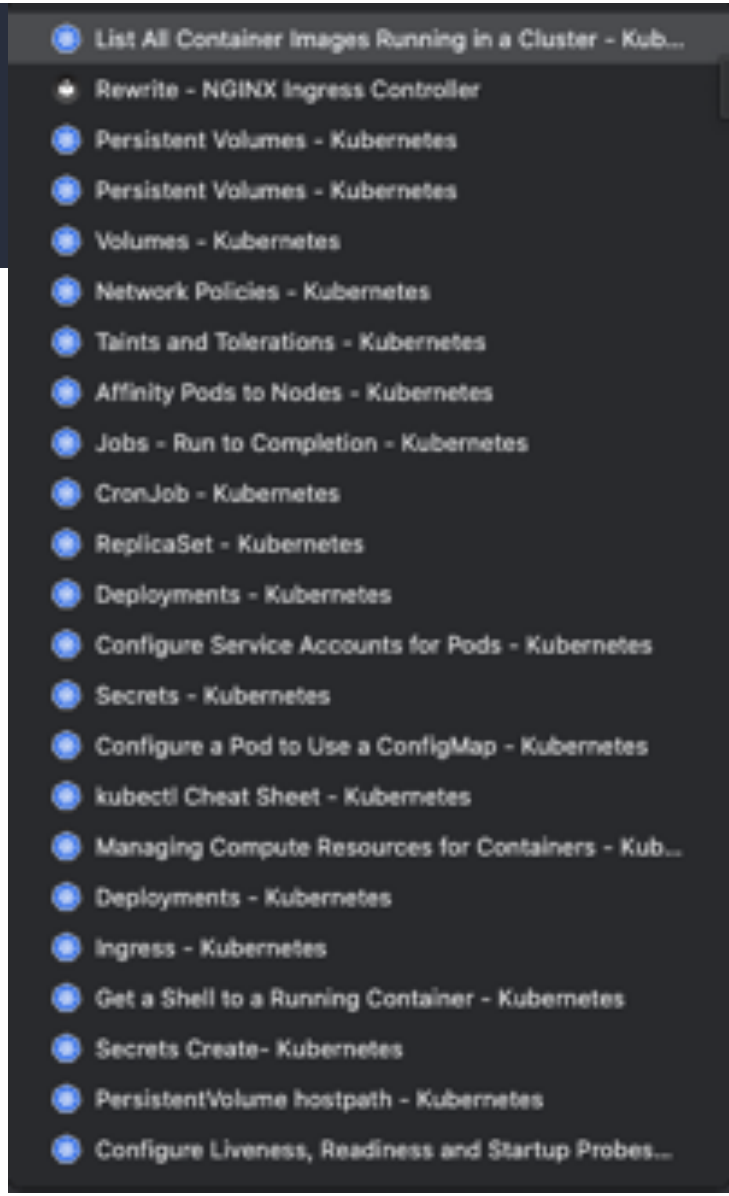
VI keys

- Useful keys:
- `/` - search
- i - insert / edit mode
- <esc> exit edit mode
- `:` command prompt
- :q! - quit without writing
- :wq - write and quit
- `A` - append to end of line
- `:undo` - undo
- `dd` - delete line
- `cw` - change word
- `dw` - delete word

https://medium.com/free-code-camp/how-not-to-be-afraid-of-vim-anymore-ec0b7264b0ae

Browser

- Browser bookmarks
- 2 browser windows open
  - Test
  - Docs



Aliases

- Required

```
alias k=kubectl
```

- Nice to have

```
alias kdr="kubectl --dry-run -o=yaml "
alias kget="kubectl -o=yaml get "
alias desc="kubectl describe "
```

Unix 1 Liners

```
while true; do date >> /var/log/app.txt; sleep 5; done
```

```
i=0; while true; do echo "$i: $(date)";i=$((i+1));sleep 1; done
```

```
mkdir -p collect; while true; do cat /var/data/* > /collect/data.text; sleep 10; done
```

```
a=10;b=5; if [$a -le $b]; then echo "a is small"; else echo "b is small";fi
```

Unix Skills

- ctrl+r search for previous command
- <esc> . - last argument used
    - (useful when you just edited yaml, now want to apply)

Lab: Setup Kind

- Install [Kind]

```
kensipe@kens-mbp-2: ~/presentations/2020/ckad/labs
❯ kind create cluster
Creating cluster "kind" ...
 ✓ Ensuring node image (kindest/node:v1.17.0) 🖼
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Have a nice day! 👋
❯ ~/pre/2/c/labs  kind create cluster --config=./kind.yaml
```

K8S Manifest Shortcuts

```
k run nginx --image=nginx  (deploy)
```

```
k run nginx --image=nginx --restart=Never (pod)
```

```
k run nginx --image=nginx --restart=OnFailure (job)
```

```
k run nginx --image=nginx --restart=OnFailure --schedule="* * * * * " (cronjob)
```

**** pre- 1.18**

K8S Manifest Shortcuts

```
k create deployment nginx --image=nginx  (deploy)
```

```
k run nginx --image=nginx --restart=Never (pod)
```

```
k create job nginx --image=nginx  (job)
```

```
k create cronjob nginx --image=nginx --schedule="* * * * *” (cronjob)
```

**** post- 1.18**

Rapid Deployment Creation

k create deployment nginx --image=nginx --dry-run -o yaml > test-1.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        name: nginx
```

Know Explain

- k explain pod.spec
- k explain pod —recursive
- k explain cronjob.spec.jobTemplate --recursive

- need details on nodeselector

```
k explain pod.spec | grep -i nodeselector
```

- Know grep

```
k explain pod.spec | grep -i -C 4 nodeselector
```

Know Manifest Layout

- Can’t remember?
  - Dry-run output a pod
  - Or explain pod

```
apiVersion:
kind:
metadata:
spec:
```

Know Describe

- Describe pod, deployment, etc.
- Looking for annotations or events… reduce output

```
k describe pods nginx | grep --context=10 annotations:
k describe pods | grep -C 10 Events:
```

Know Shortcuts

- deploy - deployment
- svc - service
- ns - namespace
- netpol - networkPolicy
- pv, pvc, sa

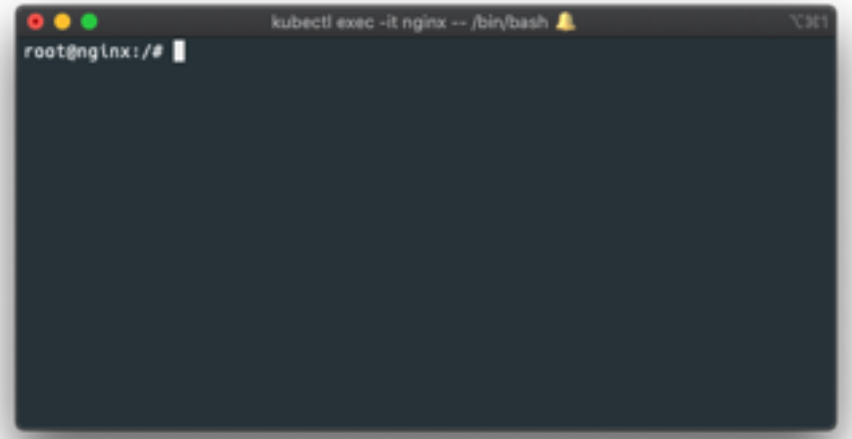
kensipe@kens-mbp-2: ~/presentations/2020/ckad/res

```
❯ k api-resources
NAME                              SHORTNAMES   APIGROUP                       NAMESPACED   KIND
bindings                                                                      true         Bind
componentstatuses                 cs                                          false        Compo
configmaps                        cm                                          true         Conf
endpoints                         ep                                          true         Endpo
events                            ev                                          true         Event
limitranges                       limits                                      true         Limit
namespaces                        ns                                          false        Names
nodes                             no                                          false        Node
persistentvolumeclaims            pvc                                         true         Persi
persistentvolumes                 pv                                          false        Persi
pods                              po                                          true         Pod
podtemplates                                                                  true         PodTe
replicationcontrollers            rc                                          true         Repli
resourcequotas                    quota                                       true         Resou
secrets                                                                       true         Secre
serviceaccounts                   sa                                          true         Servi
services                          svc                                         true         Servi
mutatingwebhookconfigurations                  admissionregistration.k8s.io   false        Mutat
ion
validatingwebhookconfigurations                admissionregistration.k8s.io   false        Valid
ation
customresourcedefinitions         crd,crds     apiextensions.k8s.io           false        Custo
apiservices                                    apiregistration.k8s.io         false        APISe
controllerrevisions                            apps                           true         Contr
daemonsets                        ds           apps                           true         Daemo
deployments                       deploy       apps                           true         Deplo
replicasets                       rs           apps                           true         Repli
statefulsets                      sts          apps                           true         State
tokenreviews                                   authentication.k8s.io          false        Token
localsubjectaccessreviews                      authorization.k8s.io           true         Local
selfsubjectaccessreviews                       authorization.k8s.io           false        Selfs
selfsubjectrulesreviews                        authorization.k8s.io           false        Selfs
subjectaccessreviews                           authorization.k8s.io           false        Subje
horizontalpodautoscalers          hpa          autoscaling                    true         Horiz
cronjobs                          cj           batch                          true         CronJ
jobs                                           batch                          true         Job
```

Know how to enter a running Pod

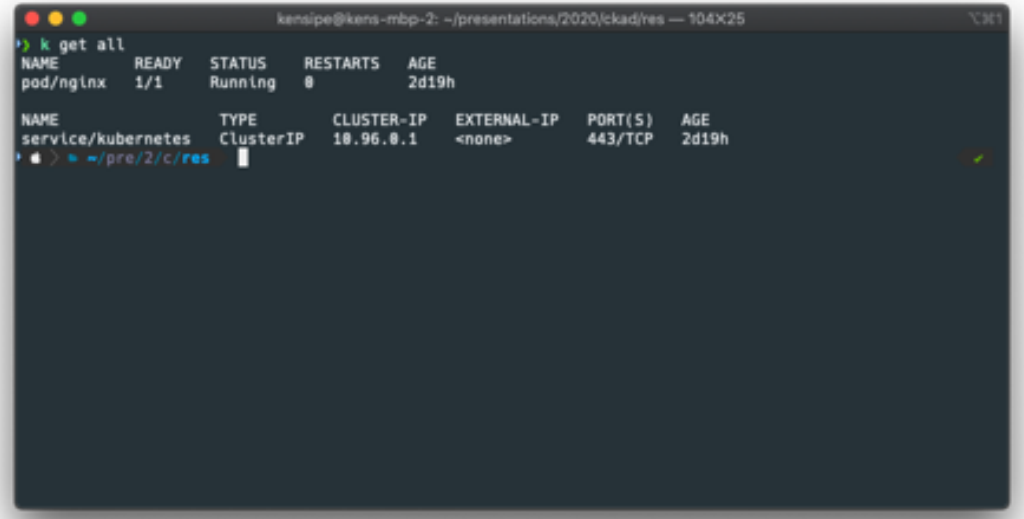
- Assuming a pod named **nginx**

```
kubectl exec -it nginx -- /bin/bash
```



Quick Namespace View

```
k get all
```



Setting Context for Namespace

```
k config set-context $(k config current-context) --namespace=dev
```

```
k config set-context $(k config current-context) --namespace=default
```

Quick View of Pod Env

```
k run busybox --image=busybox --command --restart=Never -- env
```

```
k logs busybox
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=busybox
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT=443
HOME=/root
```

Edits

- k edit (k edit pod nginx)
- Prefer:
  - K get pod nginx -o yaml > 1.yaml
  - vi 1.yaml
  - k apply -f 1.yaml

Speedy Deletes

- k delete pod nginx —grace-period=1
- k delete pod nginx —grace-period=0 —force

Linux: Grep

- Grep
  - -i case-insensitive
  - -C # (content around match by # lines)

Linux: One Liners

- Loops with output

```
while true; do date >> /var/log/app.txt; sleep 5; done
```

```
i=0; while true; do echo "$i: $(date)";i=$((i+1));sleep 1; done
```

```
mkdir -p collect; while true; do cat /var/data/* > /collect/data.text; sleep 10; done
```

Sample Apps

https://github.com/kubernetes/examples

README.md

Kubernetes Examples

This directory contains a number of examples of how to run real applications with Kubernetes.

Refer to the **Kubernetes documentation** for how to execute the tutorials.

Maintained Examples

Maintained Examples are expected to be updated with every Kubernetes release, to use the latest and greatest features, current guidelines and best practices, and to refresh command syntax, output, changed prerequisites, as needed.

Name	Description	Notable Features Used	Complexity Level
Guestbook	PHP app with Redis	Deployment, Service	Beginner
Guestbook-Go	Go app with Redis	Deployment, Service	Beginner
WordPress	WordPress with MySQL	Deployment, Persistent Volume with Claim	Beginner
Cassandra	Cloud Native Cassandra	Daemon Set, Stateful Set, Replication Controller	Intermediate

> Note: Please add examples that are maintained to the list above.

See **Example Guidelines** for a description of what goes in this directory, and what examples should contain.

Starting

Before Time Starts

- Clean Desktop
- 2 Browser (half screen each or 2 displays)
- Bookmarks for [k8s.io](k8s.io)

Immediately when time starts

```
alias k=kubectl
alias vi=vim
```

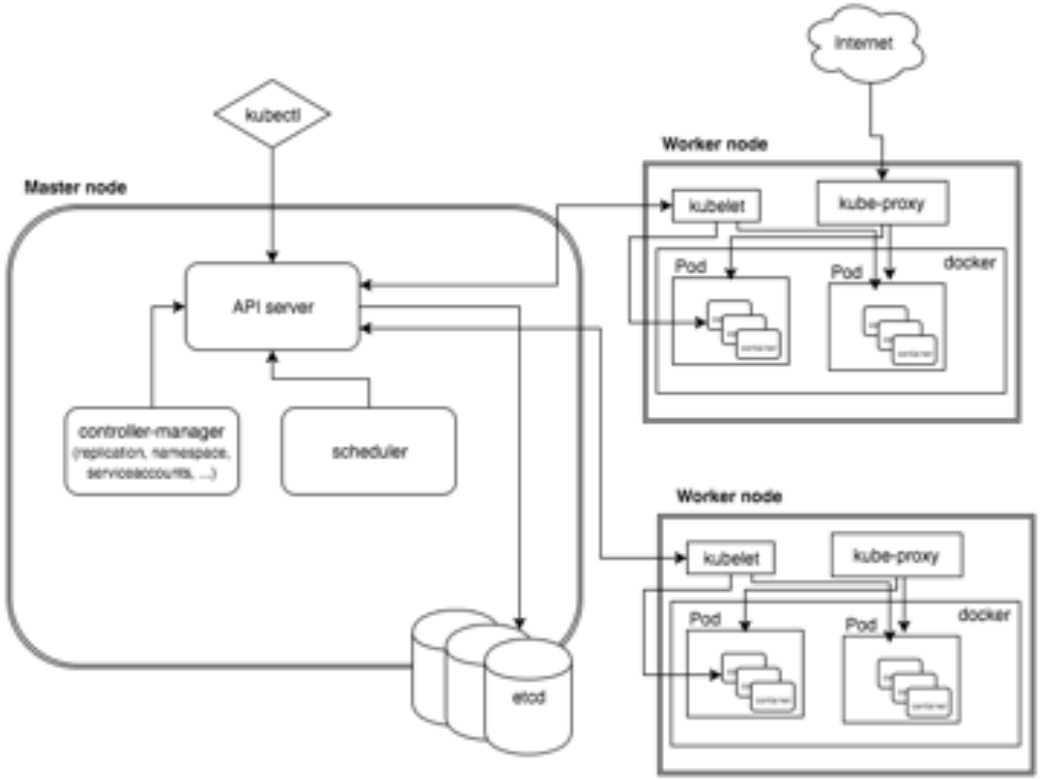
```
vi ~/.vimrc
:set number
:set et
:set sw=2 ts=2 sts=2
```

Core Concepts 13%

Core

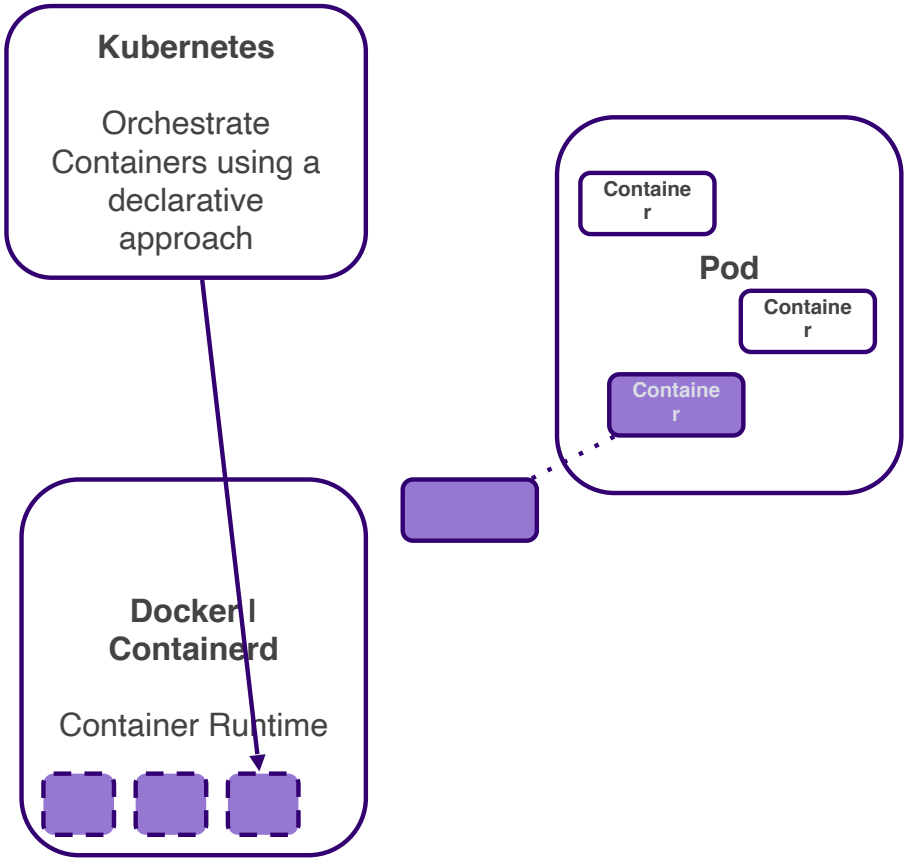
- Kube Architecture
- Pod
- YAML
- ReplicaSets
- Deployments
- Namespace

KUBERNETES ARCHITECTURE



Pods

- Pods provide an abstraction to
  - Automate scaling of containers
  - Adding resources such as networking and storage (shared resources)
  - Ability to use different container runtimes - we are using containerd
  - A Pod can contain one or more containers
- Kubernetes leverages Pods as the key abstraction to orchestrate containers
- All containers in a Pod live within a logical unit that has a single IP, but can communicate with each other over localhost.



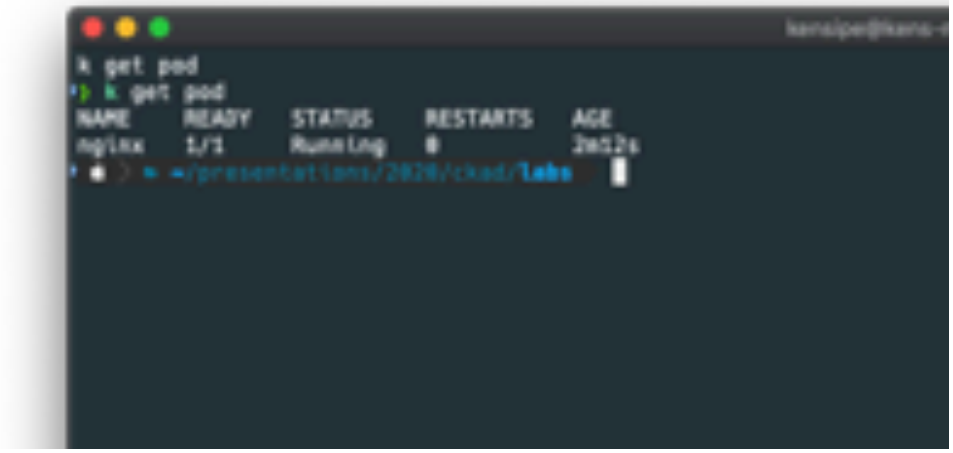
Pods

- Fundamental Unit of execution
- 1+ containers
- Metadata

```
k run nginx --image nginx -lapp=front-end
```

Pods

```
❯ k run nginx --image nginx -lapp=front-end --dry-run -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    app: front-end
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```



ReplicaSet

- ReplicaSetController - Provides high availability
- Ensures specified number of pods are running across the cluster

ReplicationController

```
k create -f lab9-1.yaml
replicationcontroller/webapp-rc created
k get rc
NAME        DESIRED   CURRENT   READY   AGE
webapp-rc   3         3         3       5s
k get pods
NAME              READY   STATUS    RESTARTS   AGE
nginx             1/1     Running   0          20h
webapp-rc-dwb4g   1/1     Running   0          9s
webapp-rc-xgxpd   1/1     Running   0          9s
```

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: webapp-rc
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
```

ReplicaSet

```
k create -f labp-1.yaml
replicaset.apps/webapp-rs created
k get rs
NAME        DESIRED   CURRENT   READY   AGE
webapp-rs   3         3         3       6s
k get pods
NAME              READY   STATUS    RESTARTS   AGE
webapp-rs-6qdlk   1/1     Running   0          11s
webapp-rs-fcfbq   1/1     Running   0          11s
webapp-rs-grvgl   1/1     Running   0          11s
```

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: webapp-rs
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      app: front-end
```

Scaling

- Update “replicas” in manifest yaml, then `k replace -f rs-def.yaml`
- `k scale –replicas= 6 -f rs-def.yaml`
- `k scale –replicas=6 rs webapp-rs`

Kubectl commands

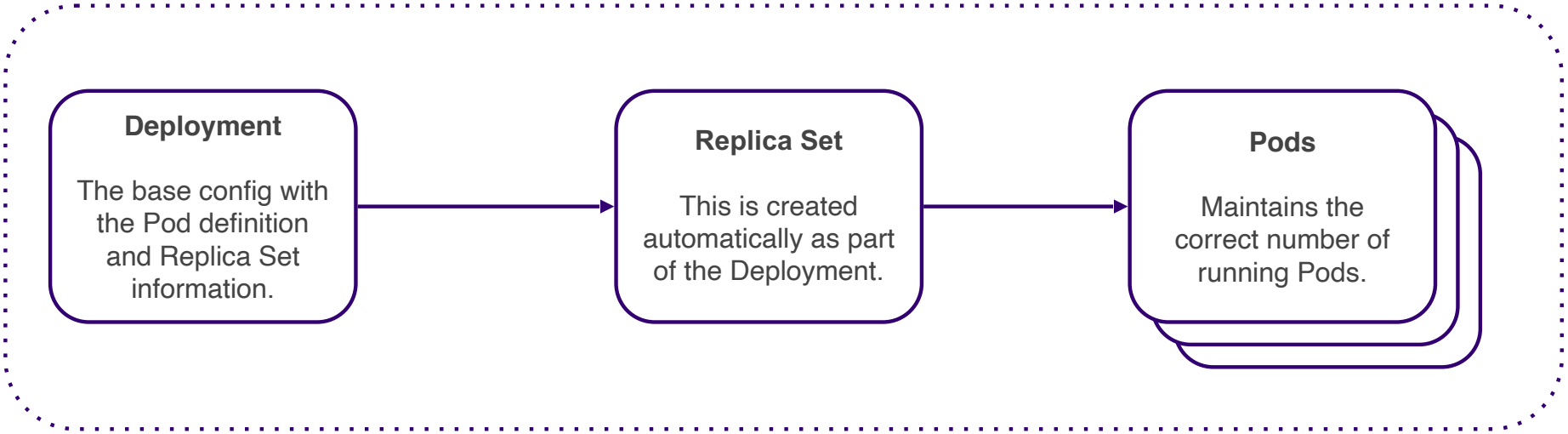
```
k create -f rs-def.yaml
k get rs
k delete -f rs-def.yaml
k replace -f rs-def.yaml
k scale –replicas=6 -f rs.def.yaml
```

Deployments

- A versioned set of replicaset, which is a collection of pods

Deployments

- A versioned set of replicaset, which is a collection of pods
- They should be used instead of Replica Sets in most cases.
- Additional functionality is added in the form of rollouts and rollbacks.
- Allows developers and operations staff to scale up and scale down applications.
- Unlike a Replica Set it contains the Pod definition as part of its configuration.

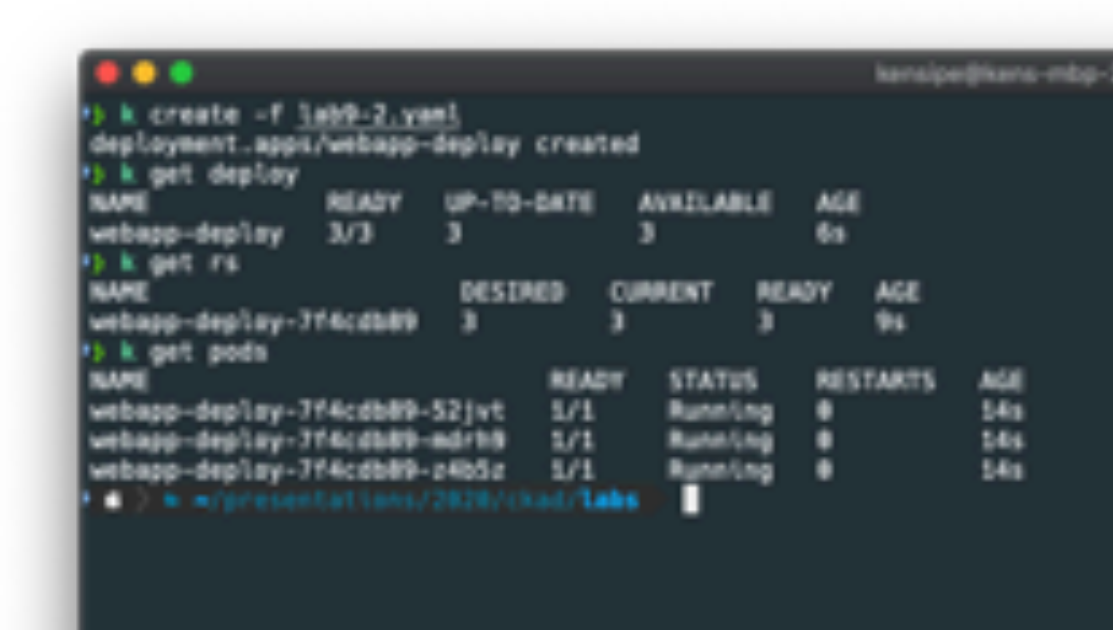


Deployment Manifest

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: webapp-rs
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      app: front-end
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp-deploy
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-contai
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      app: front-end
```

Deployments




```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp-deploy
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-contai
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      app: front-end
```

Deployments Status and History

```
k create -f lab9-2.yaml
deployment.apps/webapp-deplay created
k get deplay
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
webapp-deplay   3/3     3            3           7s
k get rs
NAME                      DESIRED   CURRENT   READY   AGE
webapp-deplay-7f4cdb89    3         3         3       14s
k get pods
NAME                              READY   STATUS    RESTARTS   AGE
webapp-deplay-7f4cdb89-pf2zw      1/1     Running   0          18s
webapp-deplay-7f4cdb89-snjpz      1/1     Running   0          18s
webapp-deplay-7f4cdb89-wtx9s      1/1     Running   0          18s
k rollout status deplay webapp-deplay
deployment "webapp-deplay" successfully rolled out
k rollout history deplay webapp-deplay
deployment.apps/webapp-deplay
REVISION  CHANGE-CAUSE
1         <none>

~/presentations/2020/ckad/labs
```

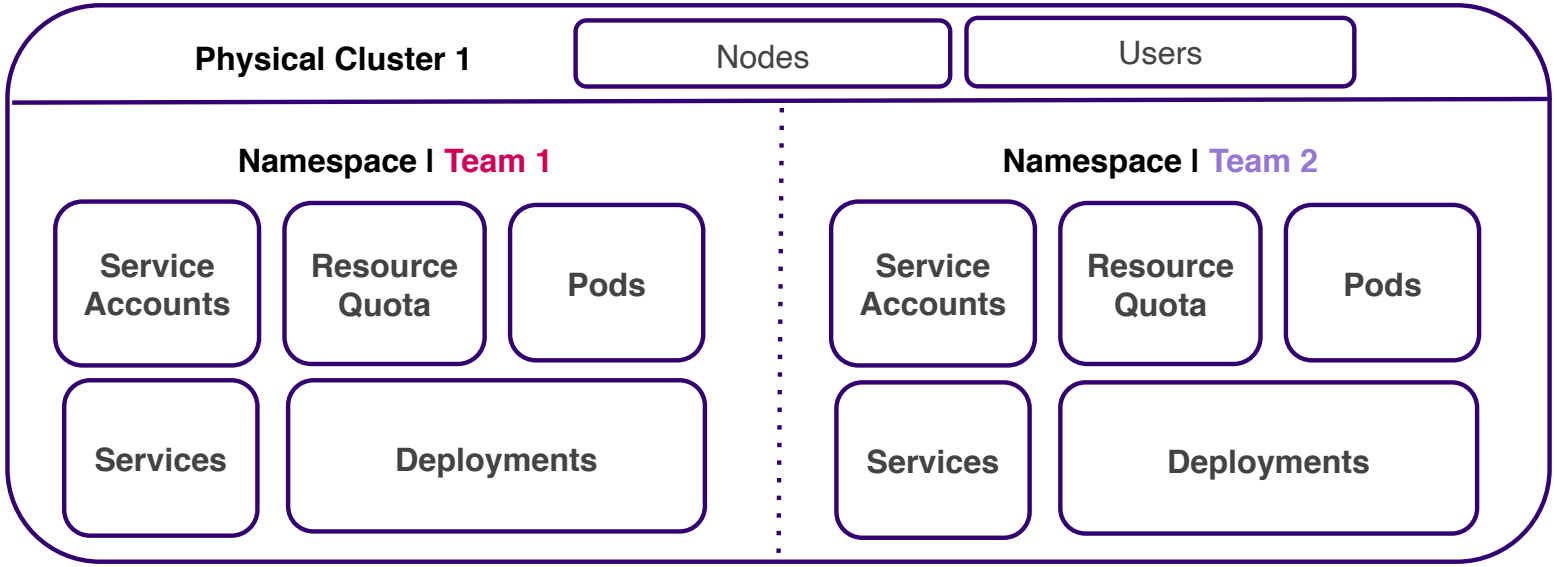
Namespace

- Provides logical name separation

- default is the default ns
- kube-system is the system namespace

Namespaces

- Each namespace can have it's own set of policies which define:
- Resource limits
- Quota
- Security



Naming in Namespace and DNS

- Within a namespace, a service can “reference” a service by it’s name:

`service.connect(“db-service”)`

- References outside namespace, it is necessary to provide namespace details

`service.connect(“db-service.dev.svc.cluster.local”)` for a “db-service” in the “dev” namespace.

Naming in Namespace and DNS

Switching Namespaces

```
k config set-context $(k config current-context) —namespace=dev
```

Always know your context / namespace during the test!

Configuration
18%

Configuration

- Commands and Arguments
- ENV
- ConfigMaps
- Secrets
- Security Context
- Service Accounts
- Resource Limits
- Taints / Tolerations
- Affinity

Commands

From Ubuntu

ENTRYPOINT [“sleep”]

CMD [“5”]

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: sleeper
spec:
  containers:
  - image: ubuntu-sleeper
    name: ubuntu
    args: ["10"]
```

docker run ubuntu-sleeper 10

Commands

From Ubuntu

ENTRYPOINT [“sleep”]

CMD [“5”]

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: sleeper
spec:
  containers:
  - image: ubuntu-sleeper
    name: ubuntu
    args: [“10"]
    command: ["sleep2.0"]
```

docker run —entrypoint sleep2.0 ubuntu-sleeper 10

ENV

- Key / Value pairs
- Array under “env”

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: sleeper
spec:
  containers:
  - image: ubuntu-sleeper
    name: ubuntu
    env:
      - name: SLEEP_TIME
        value: 1200
```

ConfigMaps

Create ConfigMap

- Imperative

```
k create configmap app-config -from-literal=SLEEP_TIME=1200
```

```
k create configmap app-config --from-file=app_config.properties
```

- Declarative

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  SLEEP_TIME: "1200"
```

Inject ConfigMap

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper-2
spec:
  containers:
  - image: ubuntu
    envFrom:
      - configMapRef:
          name: app-config
```

Secrets

Create Secret

- Imperative

```
k create secret generic app-secret -from-literal=DB_Pass=mysecret
```

```
k create secret generic app-secret --from-file=app_secret.properties
```

- Declarative

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: app-secret
data:
  DB_Password: bXlzZWNyZXQ=
```

linux: echo -n ‘mysecret’ | base64

Inject Secret

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper-2
spec:
  containers:
  - image: ubuntu
    envFrom:
      - secretRef:
          name: app-secret
```

Secrets

View Secrets

```
k create secret generic app-secret –
from-literal=DB_Pass=mysecret
```

```
❯ k describe secrets app-secret
Name:         app-secret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
DB_Pass:  8 bytes
```

view value: k get secrets app-secret -o yaml

linux: echo -n ‘bXlzZWNyZXQ=’ | base64 —decode

File Based Secrets

- In container

ls /opt/app-secret-volume

DB_Pass

```yaml
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      volumes:
      - name: app-secret-volume
        secret:
          secretName: app-secret
```

Security Context

Pod level security

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
spec:
  securityContext:
    runAsUser: 1000
    capabilities:
      add: ["MAC_ADMIN"]
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
```

Container level security

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
      securityContext:
        runAsUser: 1000
        capabilities:
          add: ["MAC_ADMIN"]
```

Security Accounts

- User
  - Admin
  - Developer
- Service
  - Prometheus
  - Jenkins

Working with Service Accounts (sa)

- k create sa foo

- k get sa

```
NAME      SECRETS   AGE
default    1         2d23h
foo        1         6s
```

```
>k describe sa foo
Name:                foo
Namespace:           default
Labels:              <none>
Annotations:         <none>
Image pull secrets:  <none>
Mountable secrets:   foo-token-8prwq
Tokens:              foo-token-8prwq
Events:              <none>
```

sa creates token, token is stored in a secret

Working with Service Accounts (sa)

```
k describe secret foo-token-8prwq
Name:        foo-token-8prwq
Namespace:    default
….

Data
====
ca.crt:     1025 bytes
namespace:  7 bytes
token:
eyJhbGciOiJSUzI1NiIsImtpZCI6IlhmWVh5RldSTEdwSkRtZWFWakdoU0hUZk5tNWxqTzhqcmRJNkQ
1M0RidFEifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ
2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3
VudC9zZWNyZXQubmFtZSI6ImZvby10b2tlbi04cHJ3cSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY
291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOiJmb28iLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2Nv
dW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJjOWE2M2MxMS04YzFiLTRlYTUtODY1My1hZjRkZTZm
OWUzM2QiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6ZGVmYXVsdDpmb28ifQ.egg4hKgZoe
YadrqpcZXZAsf3ZXMEQScydHzA4lez7939p2FOQ79vfB2D7RND_8CIabuB5CstY70YIfKMI4ktBVdEY
getHIb4g3FbAMeoSLqVKAnTJKGboMaTXoJ_BBIO-
```

Pods, Namespaces, SA

```
> k describe pod ubuntu-sleeper
Name:         ubuntu-sleeper
Namespace:    default
Node:         kind-control-plane/172.17.0.2
Labels:       <none>
Annotations:  Status:  Running
Containers:
  ubuntu:
    Image:         ubuntu
    Command:
      sleep
      4800
    State:         Running
      Started:     Mon, 17 Aug 2020 15:12:43 -0500
    Ready:         True
    Restart Count:  0
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-jk9mk (ro)
```

```
) k exec -it ubuntu-sleeper -- /bin/bash
root@ubuntu-sleeper:/# ll /var/run/secrets/kubernetes.io/serviceaccount/
..2020_08_17_20_12_37.030576906/ namespace
..data/                          token
ca.crt
root@ubuntu-sleeper:/# ll /var/run/secrets/kubernetes.io/serviceaccount/
total 4
drwxrwxrwt 3 root root  140 Aug 17 20:12 ./
drwxr-xr-x 3 root root 4096 Aug 17 21:32 ../
drwxr-xr-x 2 root root  100 Aug 17 20:12 ..2020_08_17_20_12_37.030576906/
lrwxrwxrwx 1 root root   31 Aug 17 20:12 ..data -> ..2020_08_17_20_12_37.030576906/
lrwxrwxrwx 1 root root   13 Aug 17 20:12 ca.crt -> ..data/ca.crt
lrwxrwxrwx 1 root root   16 Aug 17 20:12 namespace -> ..data/namespace
lrwxrwxrwx 1 root root   12 Aug 17 20:12 token -> ..data/token
root@ubuntu-sleeper:/#
```

Resource

- Default Requirements for a Pod
- CPU: 0.5
- MeM: 256Mi

CPU 0.1 is min
(also known as 100m)

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
      resources:
        requests:
          memory: "1Gi"
          cpu: 1
```

Resource Limits

- Limits on resources

```
        limits:
          memory: "2Gi"
          cpu: 2
```

CPU is throttled
Mem limits will OOM Killed

```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
      resources:
        requests:
          memory: "1Gi"
          cpu: 1
        limits:
          memory: "2Gi"
          cpu: 2
```

Node Taints

- Nodes can be tainted
- When a node is tainted, only tolerant pods can be scheduled.
- Anti-Affinity

```
k taint nodes node-name key=value:taint-effect
```

```
k taint nodes node1 app=red:NoSchedule
```

- Taint-Effect
  - NoSchedule
  - PreferNoSchedule
  - NoExecute

Pod Tolerations

- Pods can have tolerations
- Used to tolerate a taint

```
kubectl describe node kubemaster | grep Taint
Taints:             node-role.kubernetes.io/master:NoSchedule
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
  tolerations:
    - key: "app"
      operator: "Equal"
      value: "red"
      effect: "NoSchedule"
```

Node Selectors

- Label Nodes

```
k label nodes <node-name> key=value
```

```
k label nodes node-1 size=Large
```

- Limited:

- ~~NOT Small~~

~~- Medium or Large~~

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
  nodeSelector:
    size: Large
```

Node Affinity

- Affinity
  - requiredDuringSchedulingIgnoredDuringExecution
  - preferredDuringSchedulingIgnoredDuringExecution
- Operators
  - In
  - NotIn
  - Exists
  - DoesNotExist
  - Gt
  - Lt

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
            - key: size
              operator: In
              values:
              - Large
```

Pod Design
20%

Pods Design

- Labels, Selectors and Annotations
- Deployments
  - Rolling Updates
  - Rollbacks
- Jobs and CronJobs

Labels and Selectors

- Labels provide a way to describe an object in a way that it can be selected

- app=front-end

- app=auth

- app=db

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
  labels:
    app: front-end
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
```

Labels and Selectors

- Learn to create with generator

```
❯ k run nginx --image nginx -lapp=front-end
```

- Use Selector

```
❯ k get pods --selector app=front-end
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper
  labels:
    app: front-end
spec:
  containers:
    - image: ubuntu
      name: ubuntu
      command: ["sleep", "2000"]
```

ReplicaSets

- ReplicaSet selector will match existing matching apps that
- Labels in template is for pod
- Labels for replicaset are separate

- Similar for service

```yaml
apiVersion: v1
kind: ReplicaSet
metadata:
  name: ubuntu-sleeper
  labels:
    app: front-end
spec:
  replicas: 3
  selector:
    matchLabels:
      app: front-end
    template:
      metadata:
        labels:
          app: front-end
      spec:
        containers:
          - name: ubuntu-sleeper
            image:
```

Annotations

- Annotations are not selectable

```yaml
apiVersion: v1
kind: ReplicaSet
metadata:
  name: ubuntu-sleeper
  labels:
    app: front-end
  annotations:
    build: 1.0
spec:
  replicas: 3
  selector:
    matchLabels:
      app: front-end
    template:
```

Deployments

- A new deployment, triggers a rollout
- Rollout creates a new revision
- Commands to know:

  k rollout status deployment foo

  k rollout history deployment foo

- Deployment Strategies

- Recreate
- Rolling Update (default)

Deployments

```
❯ k rollout history deploy web-app
deployment.apps/web-app
REVISION  CHANGE-CAUSE
1         <none>

❯ ~/presentations/2020/ckad/labs
```

```
❯ k set image deploy web-app nginx-container=nginx:1.9.1
deployment.apps/web-app image updated
❯ ~
```

```
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  5m7s  deployment-controller  Scaled up replica set web-app-597c9dfcfb to 3
  Normal  ScalingReplicaSet  36s   deployment-controller  Scaled up replica set web-app-5bf48794c7 to 1
  Normal  ScalingReplicaSet  17s   deployment-controller  Scaled down replica set web-app-597c9dfcfb to 2
  Normal  ScalingReplicaSet  17s   deployment-controller  Scaled up replica set web-app-5bf48794c7 to 2
❯ ~/presentations/2020/ckad/labs  k describe deploy web-app
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-app
  labels:
    app: front-end
spec:
  replicas: 3
  selector:
    matchLabels:
      app: front-end
  template:
    metadata:
      name: web-app-pod
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
```

Deployments Imperatively

```
k create deployment nginx --image=nginx
```

```
❯ k set image deploy nginx nginx=nginx:1.9.1
deployment.apps/nginx image updated
```

```
❯ k describe deploy nginx
Name:                   nginx
Namespace:              default
CreationTimestamp:      Tue, 18 Aug 2020 16:34:56 -0500
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=nginx
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.9.1
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-7864b4477b (1/1 replicas created)
Events:
  Type    Reason             Age    From                   Message
  ----    ------             ----   ----                   -------
  Normal  ScalingReplicaSet  3m12s  deployment-controller  Scaled up replica set nginx-86c57db685 to 1
  Normal  ScalingReplicaSet  112s   deployment-controller  Scaled up replica set nginx-7864b4477b to 1
  Normal  ScalingReplicaSet  111s   deployment-controller  Scaled down replica set nginx-86c57db685 to 0
```

Rollback

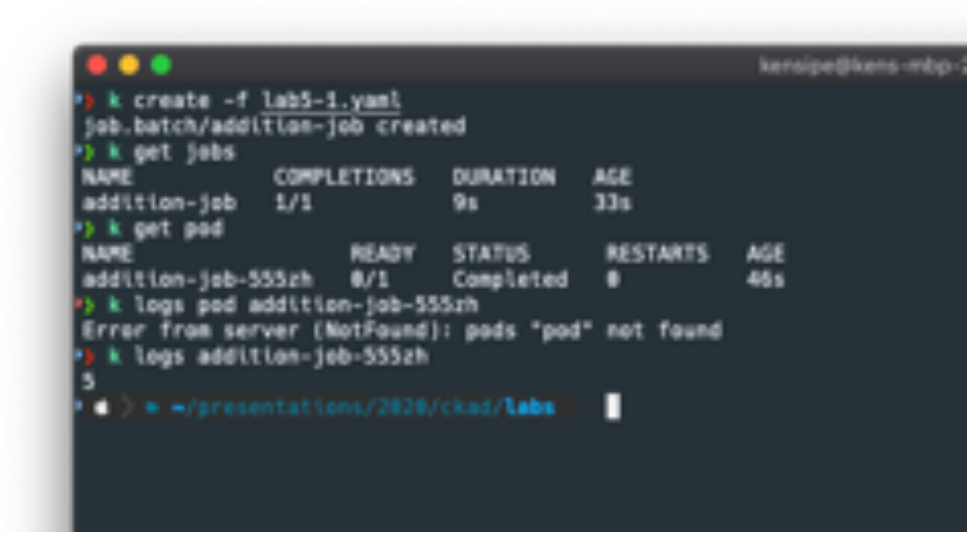
- Undo a deployment

```
k rollout undo deploy web-app
```

-

```
❯ k get rs
NAME                 DESIRED   CURRENT   READY   AGE
web-app-597c9dfcfb   3         3         3       9m46s
web-app-5bf48794c7   0         0         0       5m15s
```

Jobs




```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: addition-job
spec:
  template:
    spec:
      containers:
        - name: addition
          image: ubuntu
          command: ["expr", "3", "+", "2"]
      restartPolicy: OnFailure
```

Jobs

- Completions - The number of successful completions before job is done
- Parallelism - The number of concurrent jobs at any given point in time.

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: addition-job
spec:
  completions: 3
  parallelism: 2
  template:
    spec:
      containers:
        - name: addition
          image: ubuntu
          command: ["expr", "3", "+", "2"]
      restartPolicy: OnFailure
```

CronJob

- Spec inside a template, inside a spec, inside a jobTemplate, inside a spec!

- Schedule == cron

```yaml
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: addition-cron
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: addition
              image: ubuntu
              command: ["expr", "3", "+", "2"]
          restartPolicy: Never
```

https://en.wikipedia.org/wiki/Cron

```
k create cronjob nginx --image=nginx --schedule="* * * * *" --dry-run -o yaml
```

CronJob on Test

```
k create cronjob nginx --image=nginx --schedule="* * * * *" --dry-run -o yaml
```

- You can’t google
- Know a few cron examples at least 5 positions

Observability
18%

Observability

- Container Logs
- LivenessProbes
- ReadinessProbes

Container Logs

```
kubectl run box --image=busybox -- /bin/sh -c  'i=0; while true; do echo "$i: $(date)"; i=$((i+1)); sleep 1; done'
❯ kubectl run box --image=busybox -- /bin/sh -c  'i=0; while true; do echo "$i: $(date)"; i=$((i+1)); sleep 1; done'
pod/box created
❯ k get pod
NAME   READY   STATUS    RESTARTS   AGE
box    1/1     Running   0          3s
❯ k logs box
0: Wed Aug 19 20:11:30 UTC 2020
1: Wed Aug 19 20:11:31 UTC 2020
2: Wed Aug 19 20:11:32 UTC 2020
3: Wed Aug 19 20:11:33 UTC 2020
4: Wed Aug 19 20:11:34 UTC 2020
5: Wed Aug 19 20:11:35 UTC 2020
6: Wed Aug 19 20:11:36 UTC 2020
7: Wed Aug 19 20:11:37 UTC 2020
❯ k logs box
0: Wed Aug 19 20:11:30 UTC 2020
1: Wed Aug 19 20:11:31 UTC 2020
2: Wed Aug 19 20:11:32 UTC 2020
3: Wed Aug 19 20:11:33 UTC 2020
4: Wed Aug 19 20:11:34 UTC 2020
5: Wed Aug 19 20:11:35 UTC 2020
6: Wed Aug 19 20:11:36 UTC 2020
7: Wed Aug 19 20:11:37 UTC 2020
8: Wed Aug 19 20:11:38 UTC 2020
9: Wed Aug 19 20:11:39 UTC 2020
```

Pod Status / Conditions

- Pod Status
  - Pending - Trying to be scheduled
  - ContainerCreating - Pull Images
  - Run

- Pod Conditions (T/F)
  - PodScheduled
  - Initialized
  - ContainerReady
  - Ready

Pod Status / Conditions

Condition Ready

Status

```
❯ k get pods
NAME    READY   STATUS    RESTARTS   AGE
box     1/1     Running   0          19m
```

Meaning of Ready

- When a Container is “Ready” - Service Traffic will be routed to it!

Readiness Probes

- HTTP
- TCP
- Exec Command (exit 0)

HTTP Readiness Probe

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  labels:
    name: webapp
spec:
  containers:
  - name: webapp
    image: nginx
    ports:
      - containerPort: 80
    readinessProbe:
      httpGet:
        path: /
        port: 80
```

Readiness Probes

```yaml
readinessProbe:
  httpGet:
    path: /
    port: 80
```

```yaml
readinessProbe:
  tcpSocket:
    port: 3306
```

```yaml
readinessProbe:
  exec:
    command:
      - cat
      - /var/log/app.log
```

Readiness Probes

```yaml
readinessProbe:
  httpGet:
    path: /
    port: 80
  initialDelaySeconds: 10
  periodSeconds: 5
  failureThreshold: 8
```

```yaml
readinessProbe:
  tcpSocket:
    port: 3306
```

```yaml
readinessProbe:
  exec:
    command:
      - cat
      - /var/log/app.log
```

Liveness Probes

Health Check!

- HTTP
- TCP
- Exec Command (exit 0)

HTTP Liveness Probe

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  labels:
    name: webapp
spec:
  containers:
  - name: webapp
    image: nginx
    ports:
      - containerPort: 80
    livenessProbe:
      httpGet:
        path: /
        port: 80
```

Liveness Probes

```yaml
livenessProbe:
  httpGet:
    path: /
    port: 80
```

```yaml
livenessProbe:
  tcpSocket:
    port: 3306
```

```yaml
livenessProbe:
  exec:
    command:
      - cat
      - /var/log/app.log
```

Multi-Container 10%

Multi-Container

- Ambassador
- Adapter
- Sidecar

Pods

- Designed to be Multi-Container
- Share:
  - Lifecycle
  - Network
  - Storage

Multi-Container

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  labels:
    name: webapp
spec:
  containers:
  - name: webapp
    image: nginx
    ports:
      - containerPort: 80
  - name: log-agent
    image: log-agent
```

Architectural Patterns

- **Sidecar**
  - 2nd non-primary container for logging, analytics, etc.
- **Adapter**
  - 2nd non-primary container that processes to common format or summary format which forwards to another service
- **Ambassador**
  - 2nd container that switches env your are connecting to… like dev, test, prod

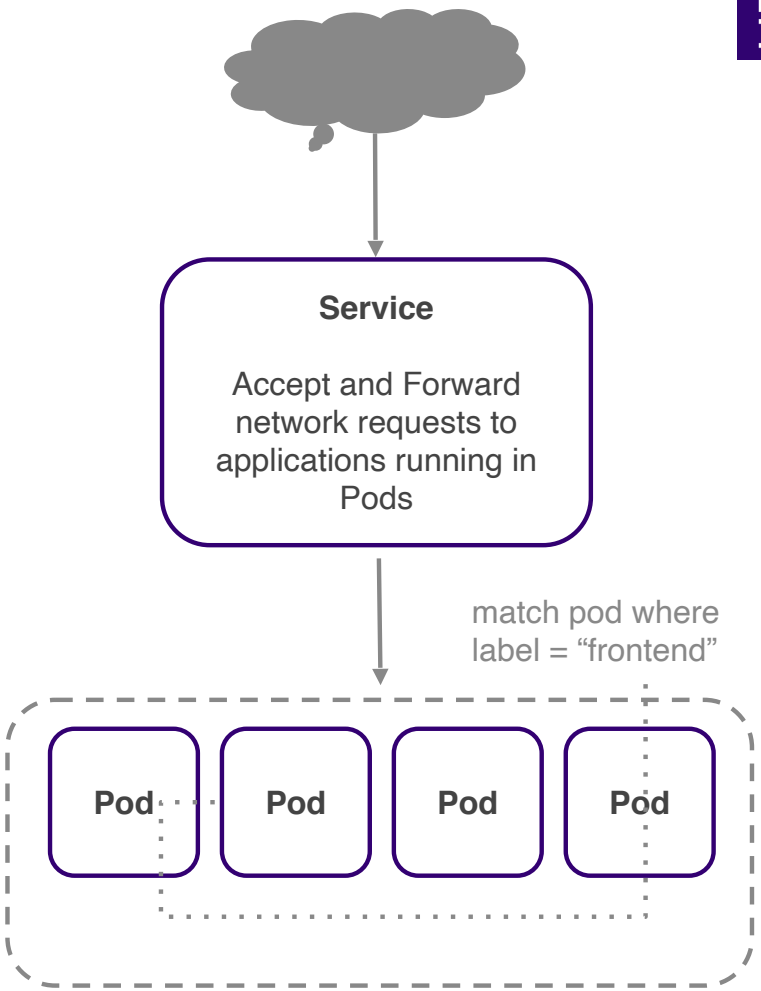
Services / Networking 13%

Services / Networking

- Services
- NetworkPolicies

Services

- Networking rules based on labels to control traffic to applications
- Service type determines how the application is exposed
  - Within the cluster (**ClusterIP**)
  - Through a static port on each node which allows external access (**NodePort**)
  - Through load balancer - physical or cloud provider (**LoadBalancer**)

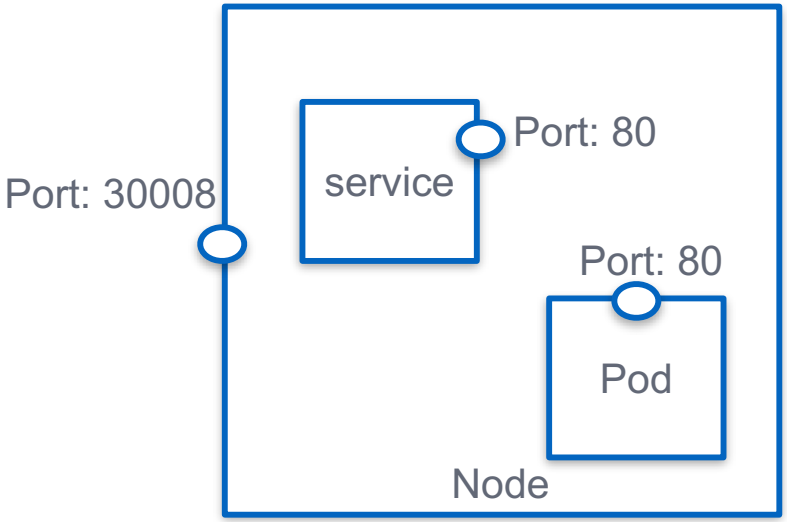


Service Types

- NodePort - service an internal pod port available on the node
- ClusterIP - creates a VIP in the cluster to enable access to a set of pods for a service (think backend or mid-tier service like a database)
- LoadBalancer - provisions a LB from cloud providers (think front-end service like web)

Service - NodePort

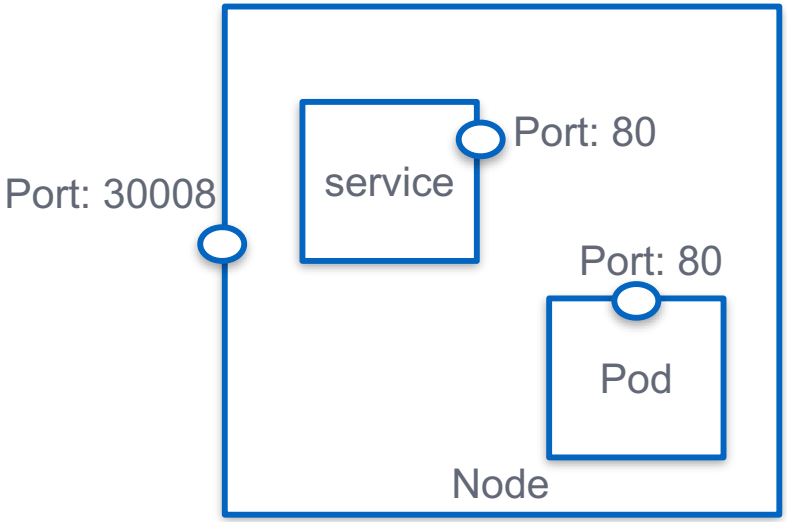
- Target Port - on pod: 80
- Port - on service
- NodePort - on node: 30008
  - default range: 30,000 - 32,767



* terms are from the perspective of the service

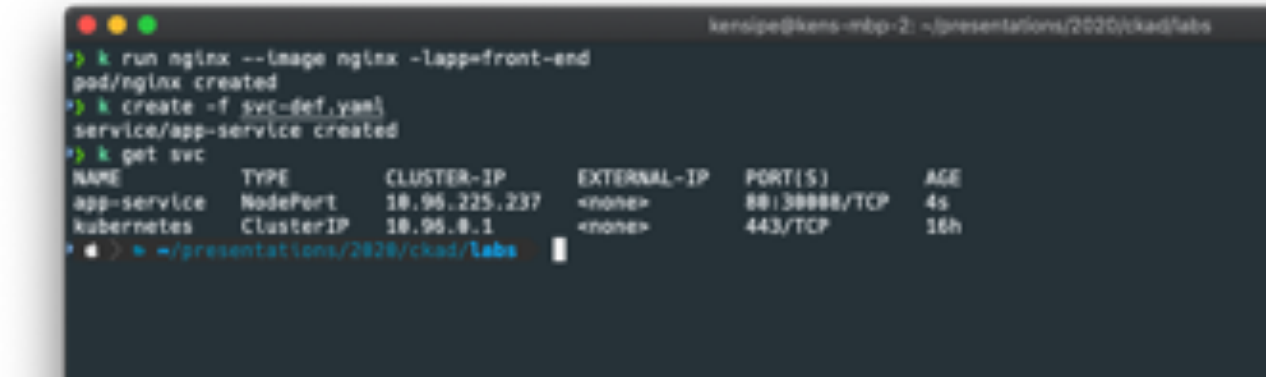
Service - NodePort

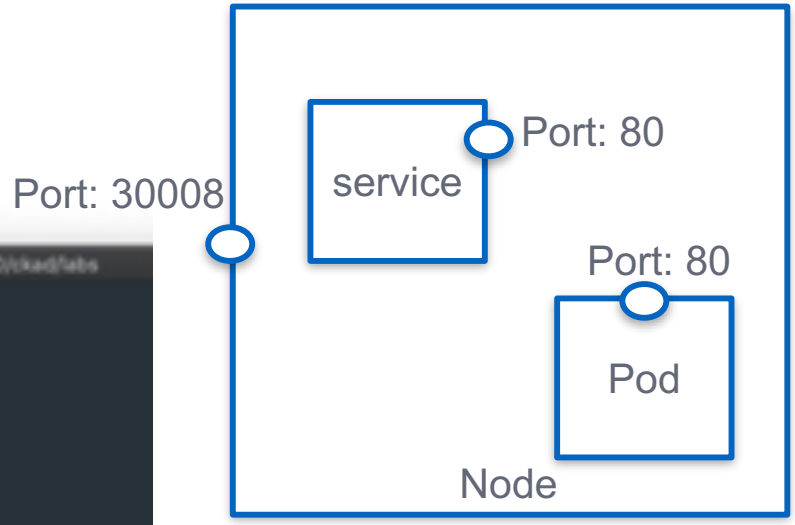
```yaml
apiVersion: v1
kind: Service
metadata:
  name: app-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
    app: front-end
```



- If you don't provide a targetPort, it is assumed to be the same as port
- If you don't provide a nodePort, it will be provided within the range

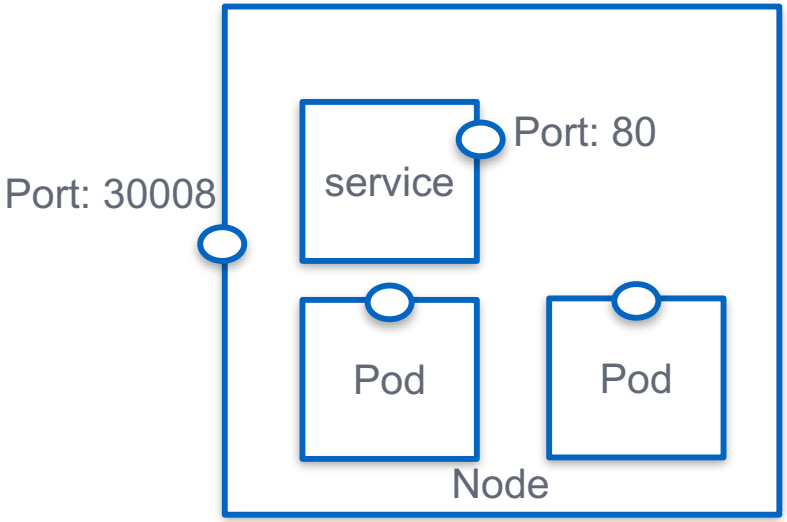
Service - NodePort





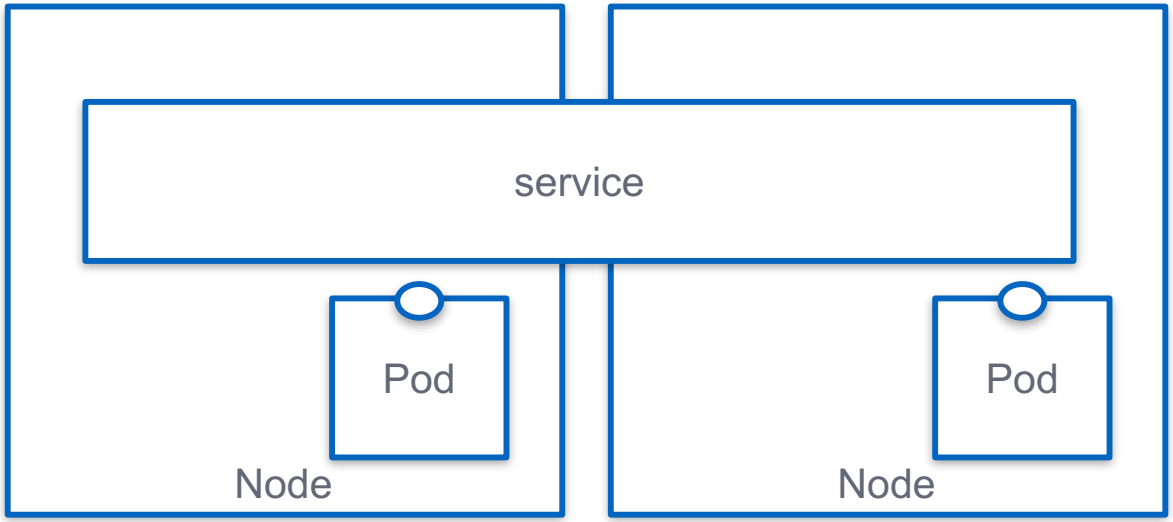
Service - NodePort

- Multiple pods on Node are:
- Randomly load balanced
- With Session Affinity



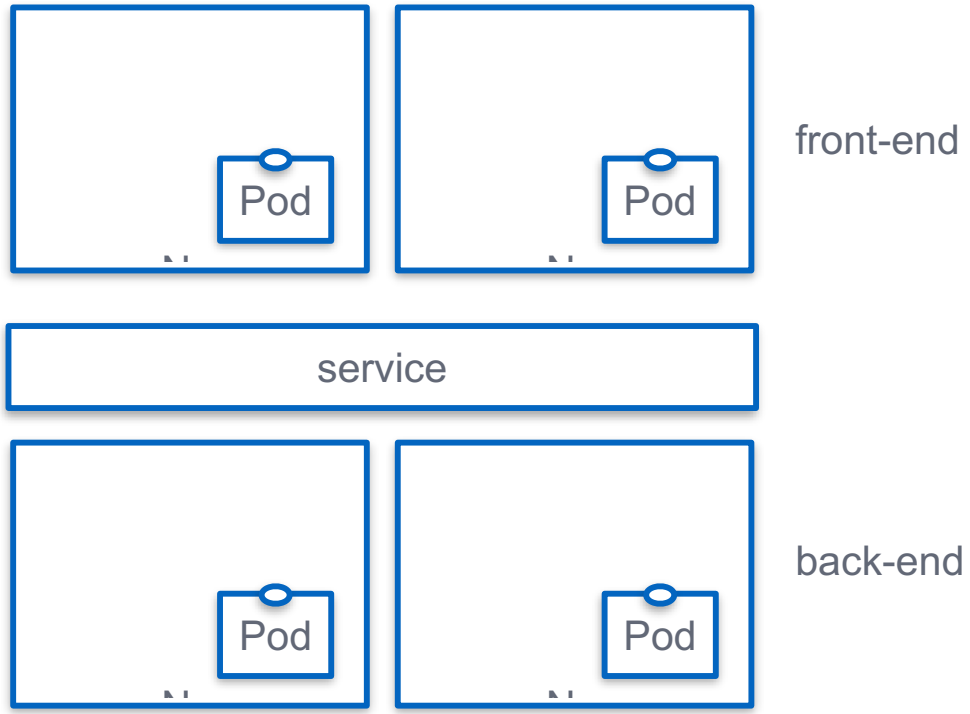
Service - NodePort

- Logic Service across Nodes
- All Node IPs have same NodePort port to provide access to service



Service - ClusterIP

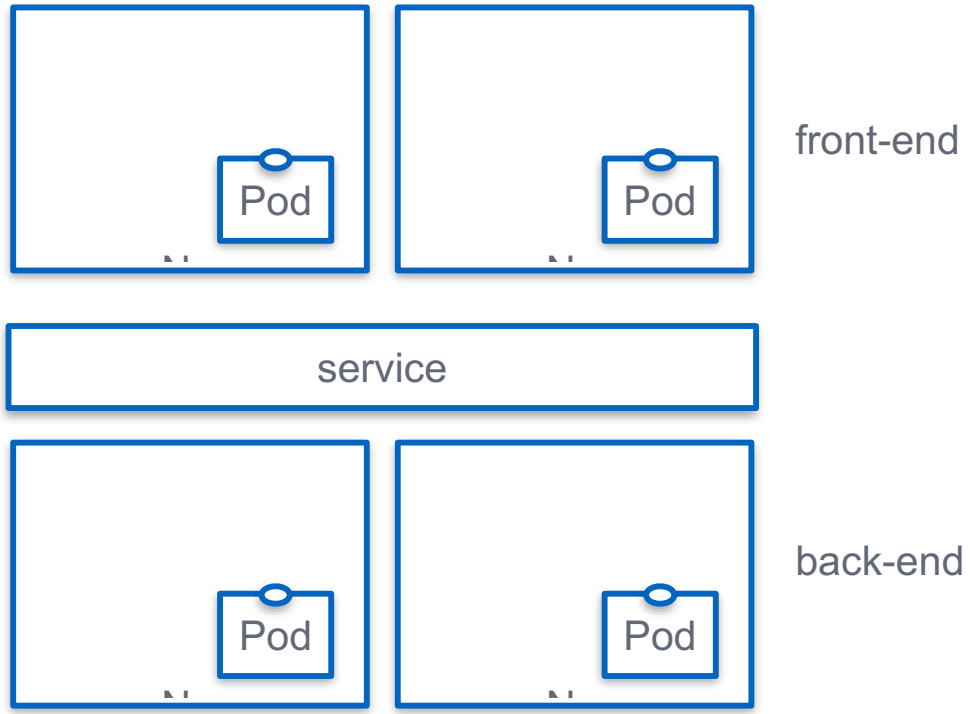
- Pods have IP… but they are elastic
-



Service - ClusterIP

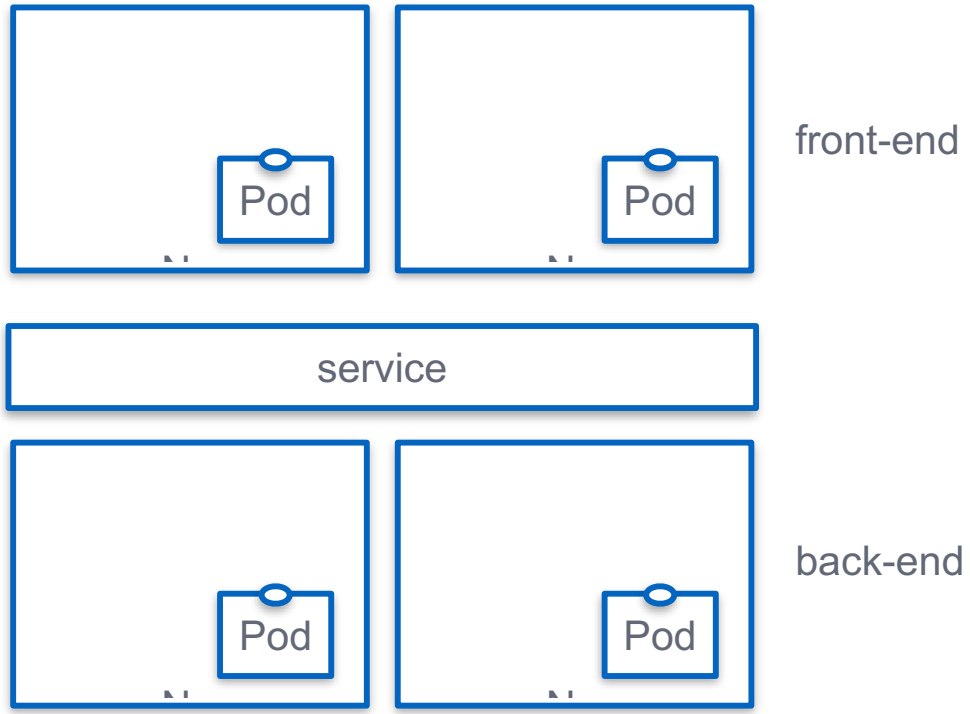
```yaml
apiVersion: v1
kind: Service
metadata:
  name: back-end
spec:
  type: ClusterIP
  ports:
    - targetPort: 80
      port: 80
  selector:
    app: back-end
```

- type: ClusterIP is the default



Service - ClusterIP

```
kensipe@kens-mbp-2: ~/presen
➜ k run nginx --image nginx -lapp=back-end
pod/nginx created
➜ k create -f svc-def.yaml
service/back-end created
➜ k get svc
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE
back-end     ClusterIP   10.96.42.25   <none>        80/TCP    3s
kubernetes   ClusterIP   10.96.0.1     <none>        443/TCP   103s
➜ ~/presentations/2020/ckad/labs
```



Service - ClusterIP / NodePort

- ClusterIP

k expose pod nginx —port

k expose deploy nginx —port

- NodePort

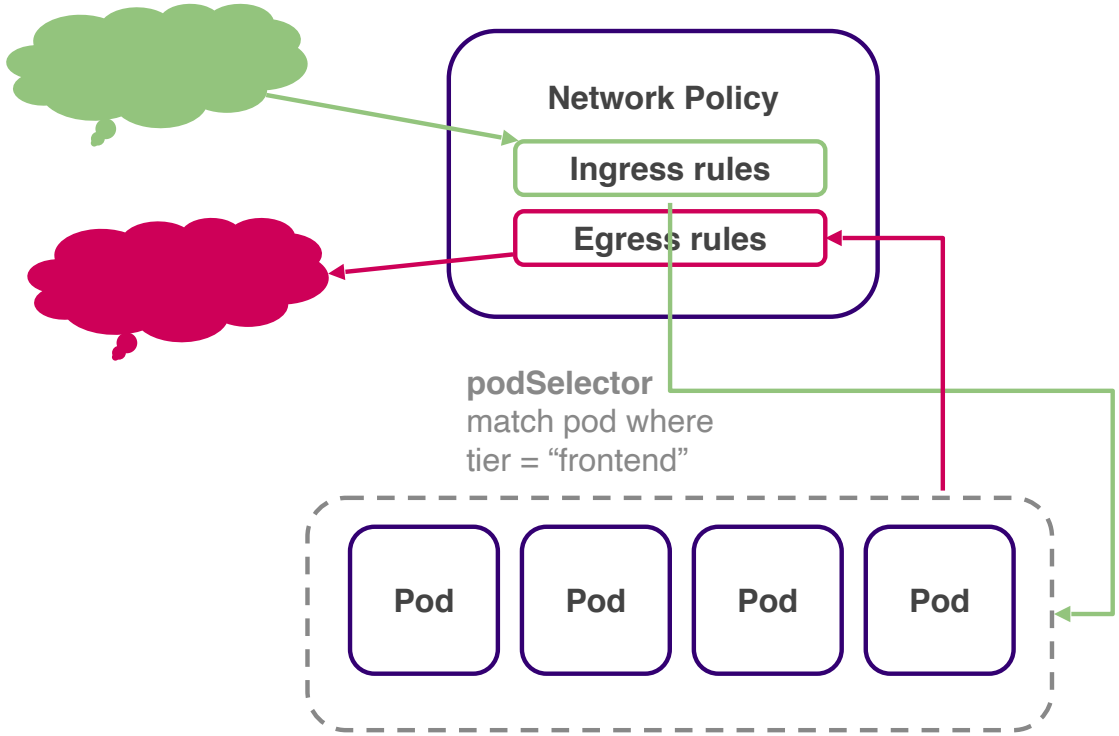
k expose pod nginx --port 8080 --type="NodePort"

Network Policies

Kubernetes has a flat network where by default any pod can communicate with every other pod through an IP address.

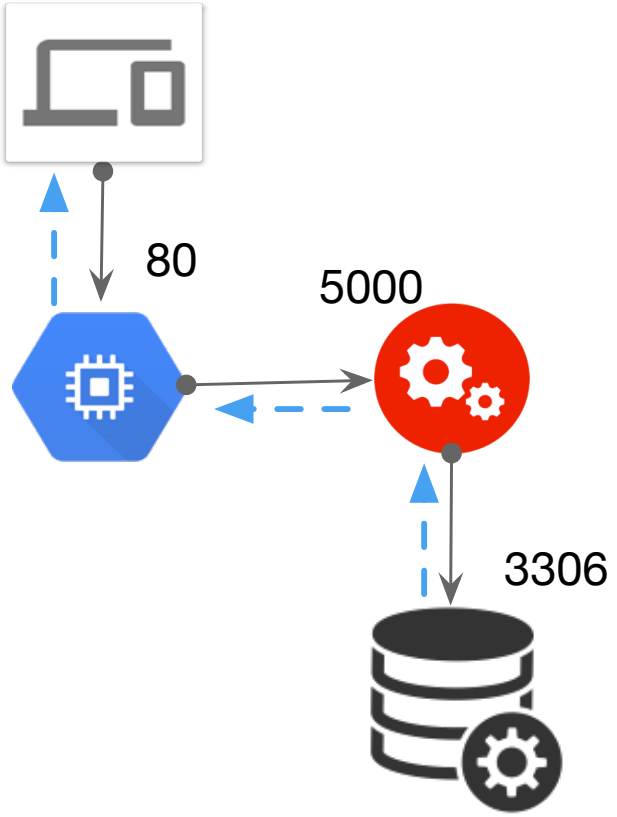
Network policies are implemented using labels much like Services.

Ingress and egress rules are generally defined by the Cluster administrator. Developers can consume the policies defined by the administrator and solely focus on the application.



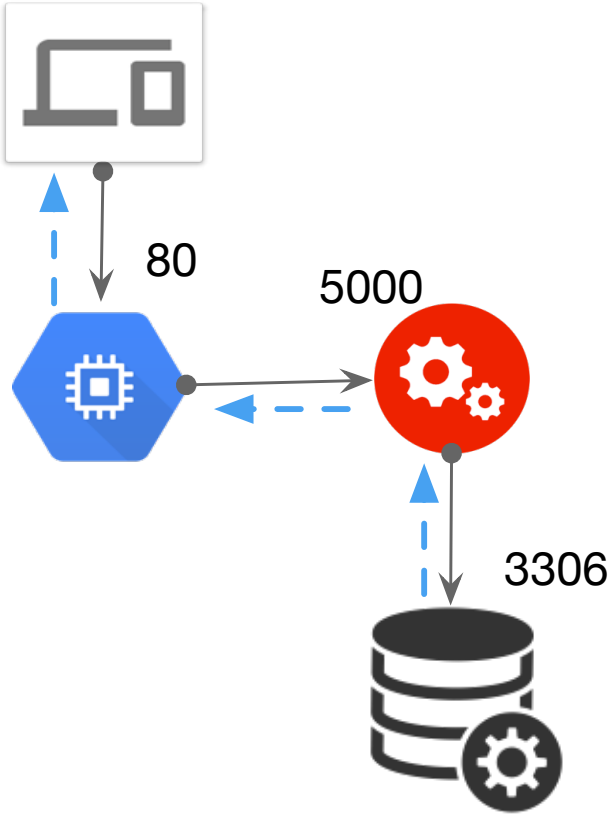
Network Policies (netpol)

- Ingress / Egress is from components perspective
- Ingress
  - Web: 80
  - Mid: 5000
- Egress
  - Web: 5000
  - Mid: 3306



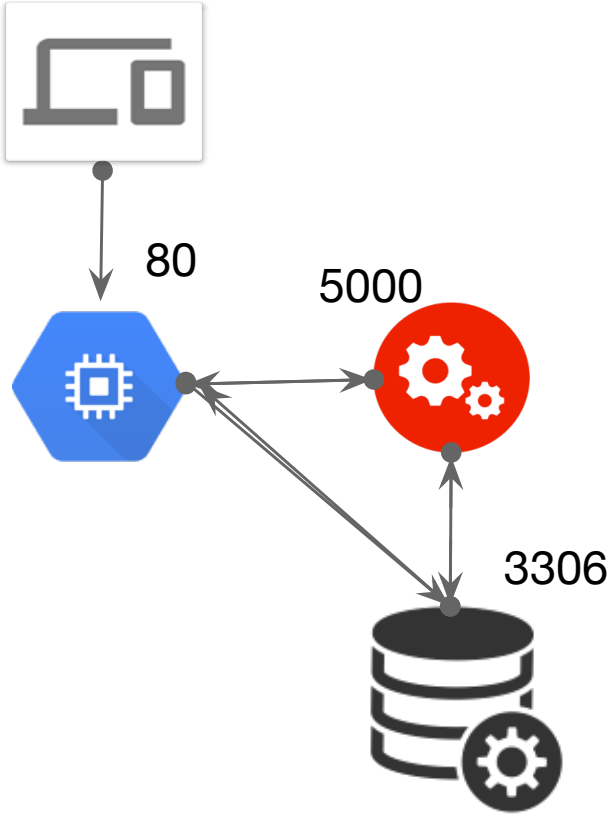
Network Policies: Rules

- Web
  - Ingress: 80
  - Egress: 5000
- Mid
  - Ingress: 5000
  - Egress: 3306
- DB
  - Ingress: 3306

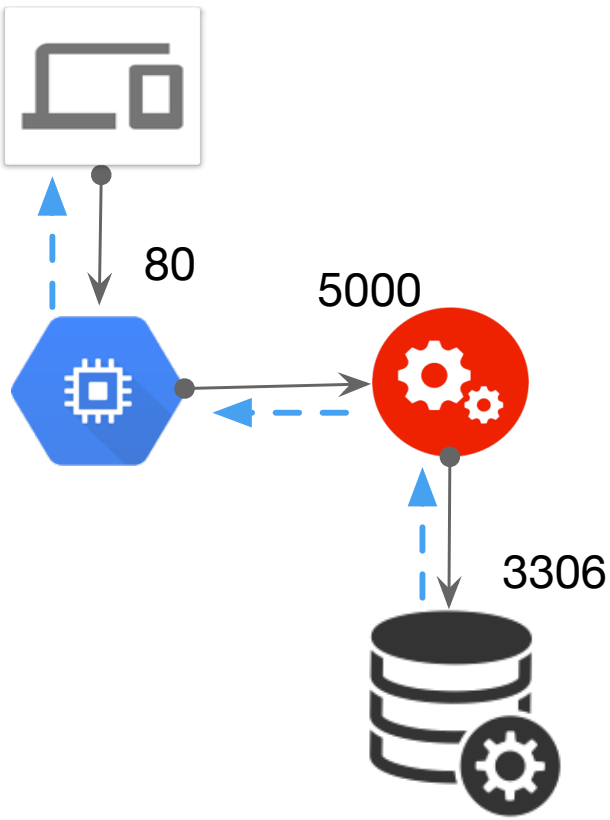


K8S Routing Rules

- All Pods can talk to all pods



NetworkPolicy




```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
    ports:
    - protocol: TCP
      port: 3306
```

NetPol Note!

Solutions support netpol

- Kube-router
- Calico
- weave-net

Do NOT support netpol

- Flannel

State Persistence 8%

State Persistence

- PV / PVC
- File Based Secrets

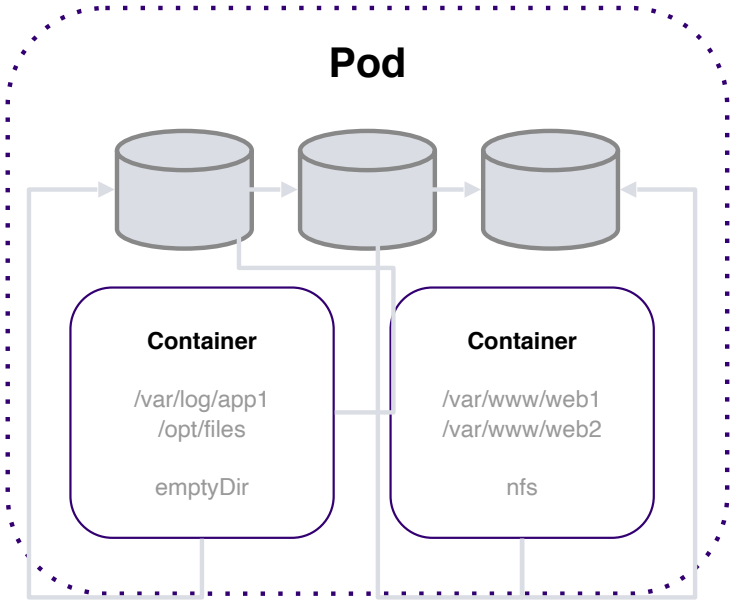
* Misleading 8% - PV needed for File Based Secrets

Kubernetes Pods

- Designed to be transient
- Need a way to manage data
  - Volume mount
  - Needs various options

Volumes

- Storage by default is ephemeral within containers
- Pods shared resources such as storage and networking
- Storage is defined as volumes and those volumes can be mounted within containers
- The underlying storage can be abstracted away through Volume types
- There are several Volume types:
  - emptyDir
  - nfs
  - hostPath
  - configMap
  - secret



Volumes definition
pod.spec.volumes

Volume mounts definition
pod.spec.containers.volumeMounts

Local Volume

```yaml
volumes:
  - name: data-volume
    hostPath:
        path: /data
        type: Directory
```

- Mounts local directory on the node to a path in the pod
- Needs to be mapped to a directory in a container…

Local Volume

- The node path “/data” will be mapped into the container “webapp” to a mount point “/opt”
- The mapping for the mount point is via the name “data-volume”
- Data written the the container to /opt will be on the host /data

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  labels:
    name: webapp
spec:
  containers:
  - name: webapp
    image: nginx
    ports:
      - containerPort: 80
    volumeMounts:
      - mountPath: /opt
        name: data-volume
  volumes:
    - name: data-volume
      hostPath:
          path: /data
          type: Directory
```

Volume Storage Options

- Kubernetes supports lots of underlying solutions
  - NFS
  - GlusterFS
  - Ceph
  - Flocker
  - Various Cloud solutions (AWS, GCP, Azure)

Volume Storage Options

- Kubernetes supports lots of underlying solutions
  - NFS
  - GlusterFS
  - Ceph
  - Flocker
  - Various Cloud solutions (AWS, GCP, Azure)

AWS EBS

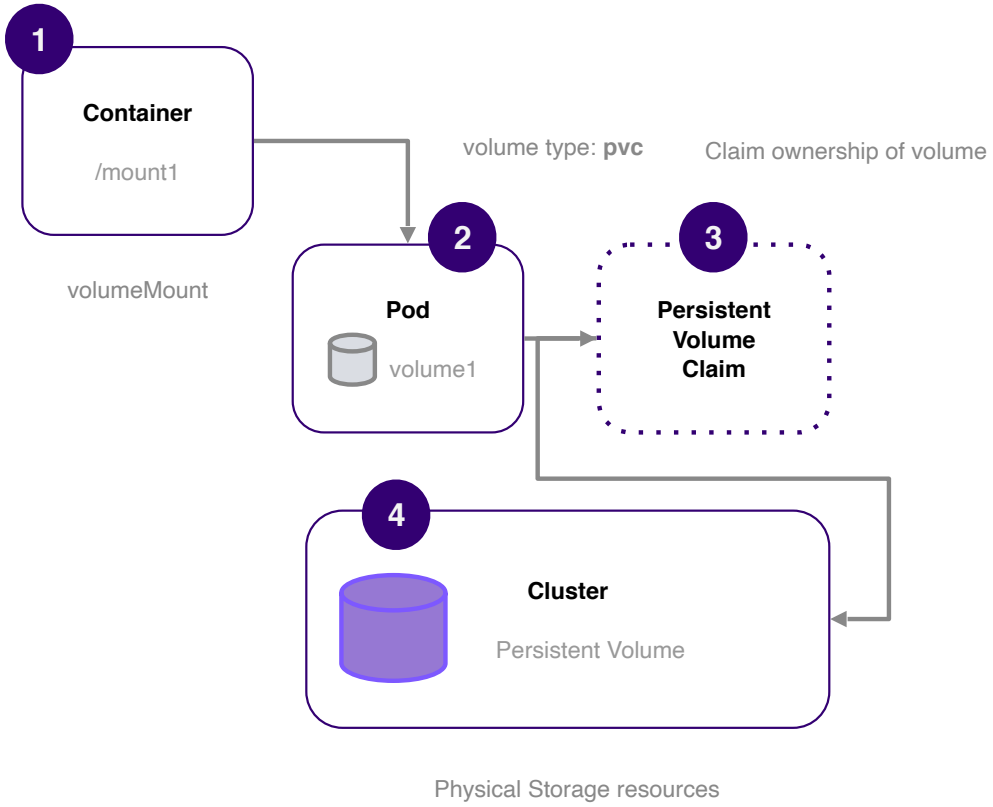
```yaml
  volumes:
    - name: data-volume
      awsElasticBlockStore:
          volumeID: <vol-id>
          fsType: ext4
```

Persistent Volumes

- Provides ability to centrally manage a “pool” of data storage volumes
- Pods can place a claim (called Persistent Volume Claims (PVC)) on a volume

Persistent Volumes

- Provides ability to centrally manage a “pool” of data storage volumes
- Pods can place a claim
  - Persistent Volume Claims (PVC)



Persistent Volume

- Access Modes:
  - ReadOnlyMany
  - ReadWriteOnce
  - ReadWriteMany

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-vol1
spec:
  accessModes:
    - ReadWriteOnce
```

Persistent Volume

```
k create -f lab10-2.yaml
persistentvolume/pv-vol1 created
k get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
pv-vol1   1Gi        RWO            Retain           Available                                   16s
```

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-vol1
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 1Gi
  hostPath:
    path: /tmp/data
```

PV vs PVC

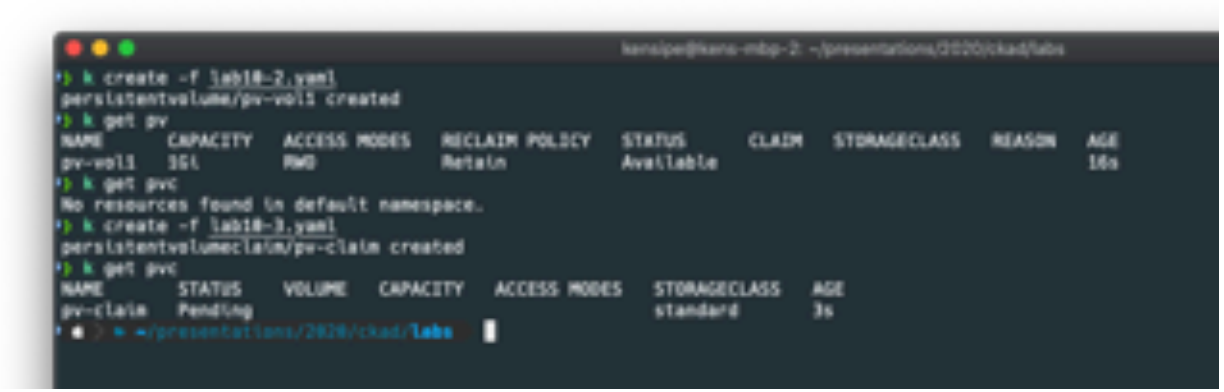
- Admin creates PV and manages storage
- User creates a claim on that PV or something that closes matches the requirements for the pods storage needs.

- PVC are Bound to a PV based on request properties

- It is possible to bind to a volume by selectors

- Once a PV is claimed.. it can’t be bound by another claim
- If a PVC can’t match a PV… it remains in a Pending State.

PVC




```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Mi
```

PVC Retention

- persistentVolumeReclaimPolicy:
  - Retain
  - Delete
  - Recycle

Pod using PVC

```
kensiper@kens-mbp-2: ~/presentations/2020/ckad/labs
k create -f lab10-2.yaml
persistentvolume/pv-vol1 created
k get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
pv-vol1   500Mi      RWO            Retain           Available                                   4s
k create -f lab10-3.yaml
persistentvolumeclaim/pv-claim created
k get pvc
NAME       STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
pv-claim   Pending                                      standard       3s
k get pvc
NAME       STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
pv-claim   Pending                                      standard       27s
k event
Error: unknown command "event" for "kubectl"
Run 'kubectl --help' for usage.
k get events
LAST SEEN   TYPE     REASON                 OBJECT                          MESSAGE
119s        Normal   WaitForFirstConsumer   persistentvolumeclaim/pv-claim  waiting for first consumer to be created before binding
9s          Normal   WaitForFirstConsumer   persistentvolumeclaim/pv-claim  waiting for first consumer to be created before binding
11m         Normal   Killing                pod/webapp-deploy-7f4cdb89-pf2zv Stopping container nginx-container
11m         Normal   Killing                pod/webapp-deploy-7f4cdb89-snjpz Stopping container nginx-container
11m         Normal   Killing                pod/webapp-deploy-7f4cdb89-wtx9s Stopping container nginx-container
11m         Normal   Scheduled              pod/webapp                      Successfully assigned default/webapp to kind-worker
11m         Normal   Pulling                pod/webapp                      Pulling image "nginx"
11m         Normal   Pulled                 pod/webapp                      Successfully pulled image "nginx"
11m         Normal   Created                pod/webapp                      Created container webapp
11m         Normal   Started                pod/webapp                      Started container webapp
k create -f lab10-4.yaml
Error from server (AlreadyExists): error when creating "lab10-4.yaml": pods "webapp" already exists
k delete pod webapp
pod "webapp" deleted
k create -f lab10-4.yaml
pod/webapp created
k get pvc
NAME       STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
pv-claim   Bound    pvc-a20f8a5e-37da-4324-8d16-d0a804bf1389   500Mi      RWO            standard       5m19s
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  labels:
    name: webapp
spec:
  containers:
  - name: webapp
    image: nginx
    ports:
      - containerPort: 80
    volumeMounts:
      - mountPath: /var/www/html
        name: mystate
  volumes:
    - name: mystate
      persistentVolumeClaim:
          claimName: pv-claim
```

Pod using PVC

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-vol1
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 1Gi
  hostPath:
    path: /tmp/data
```

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Mi
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  labels:
    name: webapp
spec:
  containers:
  - name: webapp
    image: nginx
    ports:
      - containerPort: 80
    volumeMounts:
      - mountPath: /var/www/html
        name: mystate
  volumes:
    - name: mystate
      persistentVolumeClaim:
        claimName: pv-claim
```

Storage Classes

- Mechanism for auto-creating storage (PV) on demand

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: google-storage
provisioner: kubernetes.io/gce-pd
```

PVC with Storage Class

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: google-storage
provisioner: kubernetes.io/gce-pd
```

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: google-storage
  resources:
    requests:
      storage: 500Mi
```

StatefulSet

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp-rs
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      app: front-end
```

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: webapp-deploy
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-conta
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      app: front-end
   serviceName: webapp-h
```

StatefulSets

- When scaling up (or down)
  - Ordered
  - Graceful Deployment
  - Stable, unique DNS record

- Pod will only start if the previous pod is "ready"

mysql-0 mysql-1 mysql-2

Headless Services

- Standard “service” load-balances across pods
- A common data source trait, is that reads can originate from all nodes, but writes need to be coordinated or be written by a “master”
- When need a DNS entry for access to an instance of a service without load balancing across the service nodes.

Service Part of Headless Service

```yaml
apiVersion: v1
kind: Service
metadata:
  name: webapp-h
spec:
  ports:
    - port: 80
  selector:
    app: front-end
  clusterIP: None
```

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: webapp-deploy
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      app: front-end
  serviceName: webapp-h
```

Headless DNS

webapp-0.webapp-h.dev.svc.cluster.local

stateful pod name | service name | namespace | service | domain

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: webapp-deploy
  labels:
    name: webapp
spec:
  template:
    metadata:
      name: webapp
      labels:
        app: front-end
    spec:
      containers:
        - name: nginx-cont
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      app: front-end
  serviceName: webapp-h
```

Thank You!

@kensipe
kensipe@gmail.com