

Logs

kubectl logs -f <pod-name>

- * If there are two or more containers inside pod,
kubectl logs -f <pod-name> <container-name>

Labels & Selectors

- * Mention labels under metadata

- * We can get pods based on labels

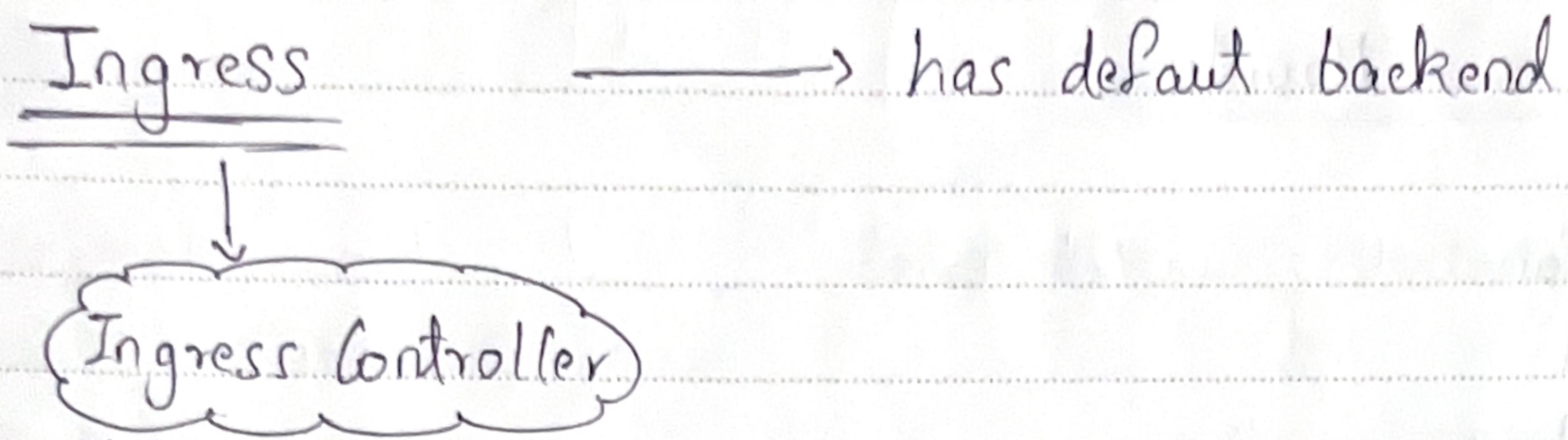
kubectl get pods --selector <labels>

Annotations are used to record other details for
informative purpose

Ex: annotations:

buildVersion: 1.24

Placed under metadata
in yaml file



- evaluates all rules
 - manages redirections
 - entrypoint to cluster
 - many third-party implementation

Ex: K8s Nginx Ingress Controller

YAML file

apiversion: networking.k8s.io/v1

Kend: Ingress

metadata:

name: simple-app

Spec.:

rules :

- host : myapp.com

http://

paths.:

- path: IAnalytics → pathType: Prefix

backend:

serviceName: analytics-service

ServicePort - 3000

- path: /shopping → PathType: Prefix

backend:

Service Name: shopping-service

servicePort: 8080

We can have multiple subdomains
& multiple hosts

Ex: Host: analytics.myapp.com

→ changed

Service:

Name:

Part:

number:

Volumes & Mounts

```
apiVersion: v1
kind: PersistentVolume
metadata:
spec:
  containers:
    - name: myapp
      image: nginx
      volumeMounts:
        - mountPath: /opt
          name: idata-volume
  volumes:
    - name: idata-volume
      hostPath:
        path: /data
        type: Directory
      } awsElasticBlockstore:
        volumeID: <volume-ID>
        fsType: ext4
```

Persistent Volume

It is a cluster wide pool of storage volumes configured by an administrator to be used by users deploying applications on cluster.

- » Users can now select storage from this pool using persistent volume claim (PVC)

pv-def.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-voll
```

Spec:

accessModes:

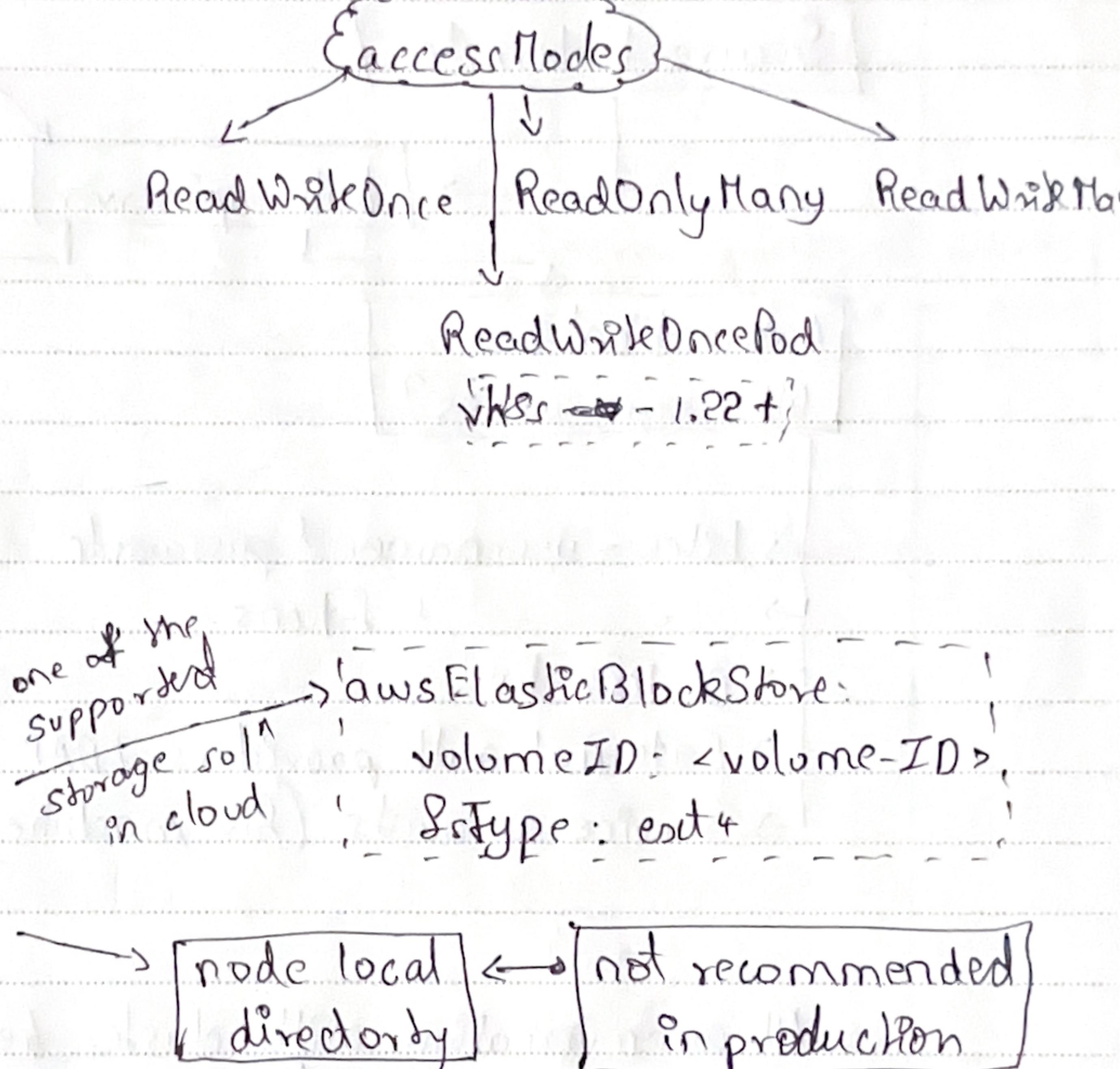
- ReadWriteOnce

capacity:

storage: 1Gi

hostPath:

path: /tmp/data



Persistent Volume Claim

Same for deployment / Replicaset

pvc-def.yaml

```
apiVersion: v1
```

kind: PersistentVolumeClaim

metadata:

name: myclaim

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 500Mi

volumeName: <pv-volume-name>

Spec:

containers:

- name:

volumeMounts:

- mountPath: "/var/www/html"

name: mypd

volumes:

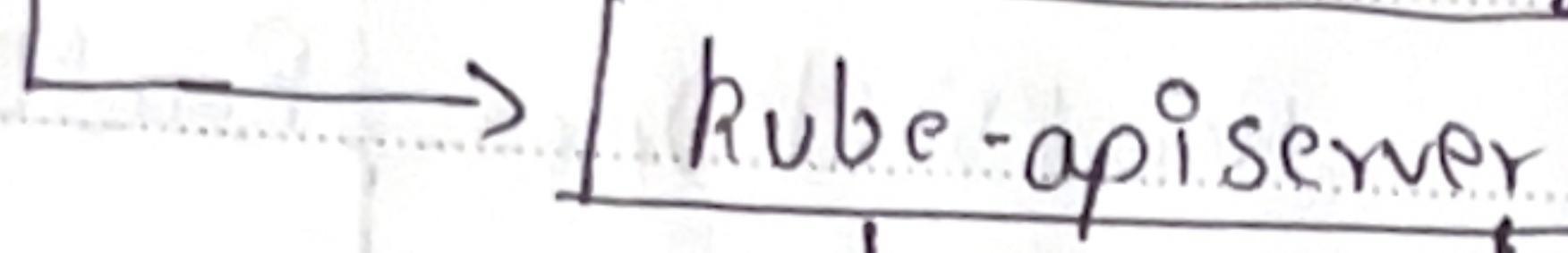
- name: mypd

persistenVolumeClaim:

claimName: myclaim

Secure Kubernetes

center of all operations



Authentication

Who can access?

- Files - usernames & passwords
- " " & tokens
- Certificates
- External auth providers - LDAP
- Service accounts (for machines)

Authorization

What can they do?

- RBAC authorization
- ABAC (attributes based)
- Node authorization
- Webhooks

All communication with cluster between components secured using

