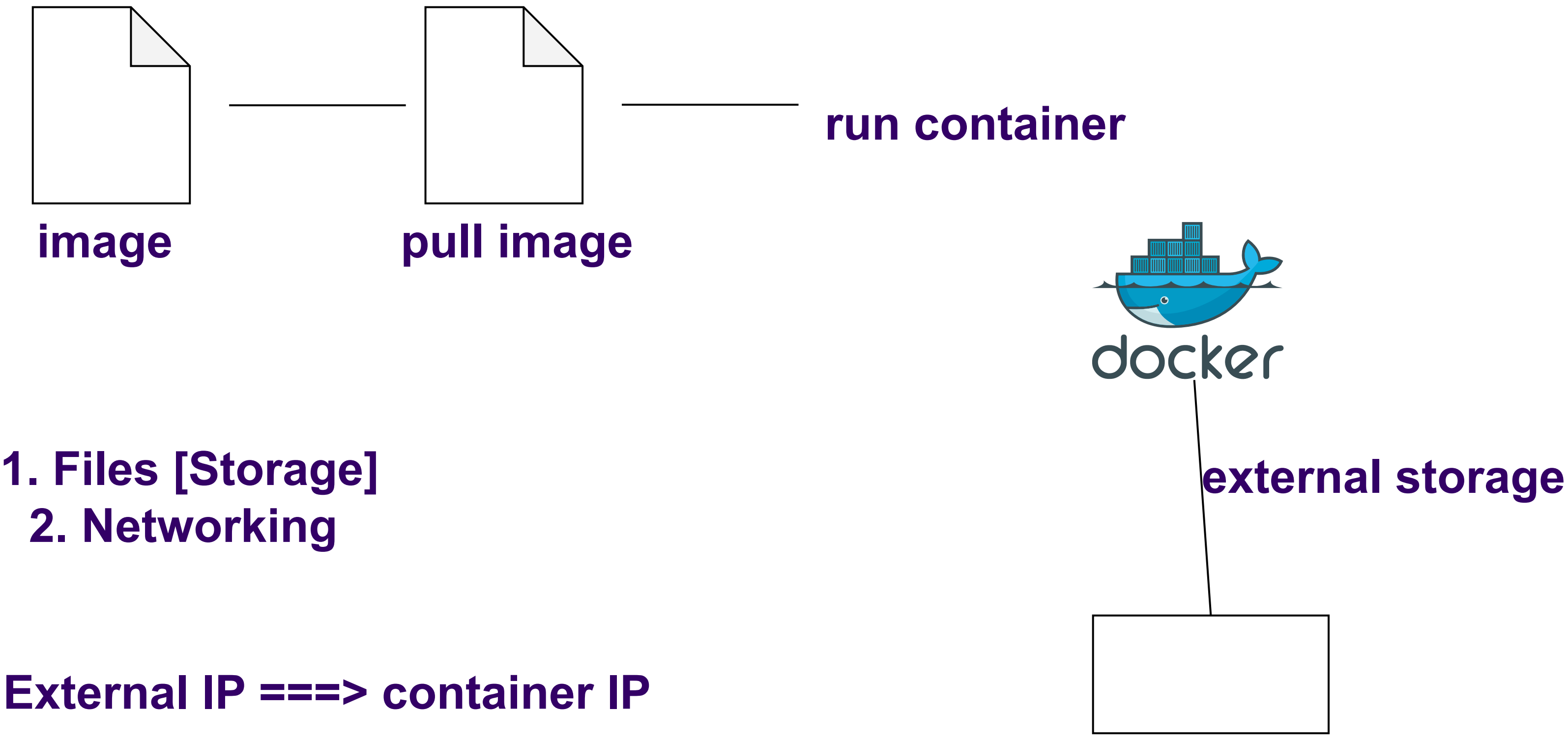
Container - complete software package

Kubernetes - Manages (docker or any) container

Docker and K8s - Microservices
- high vol
- Works great with Stateless

-Auto Scaling
- Health checks
- replacing unhealthy resource

Container
No Data persistence



image

pull image

run container

1. Files [Storage]
2. Networking

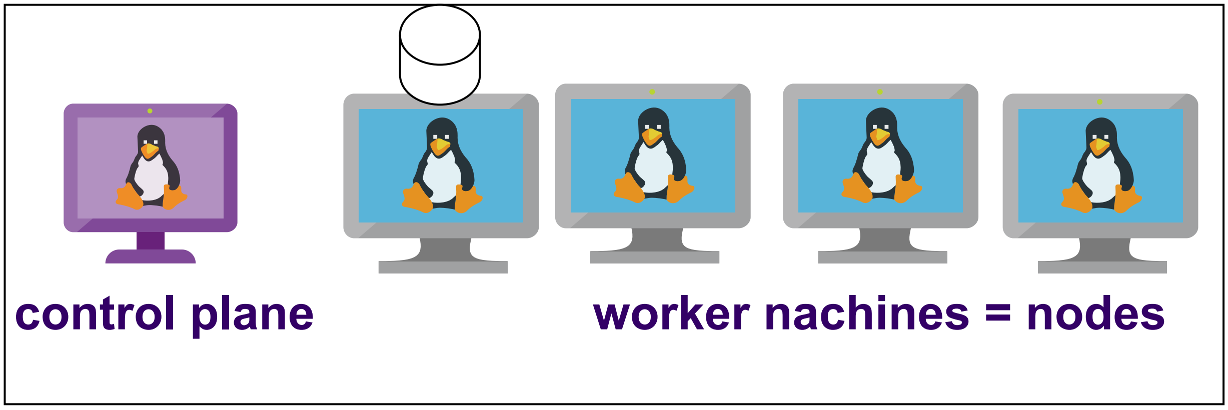
external storage

External IP ===> container IP

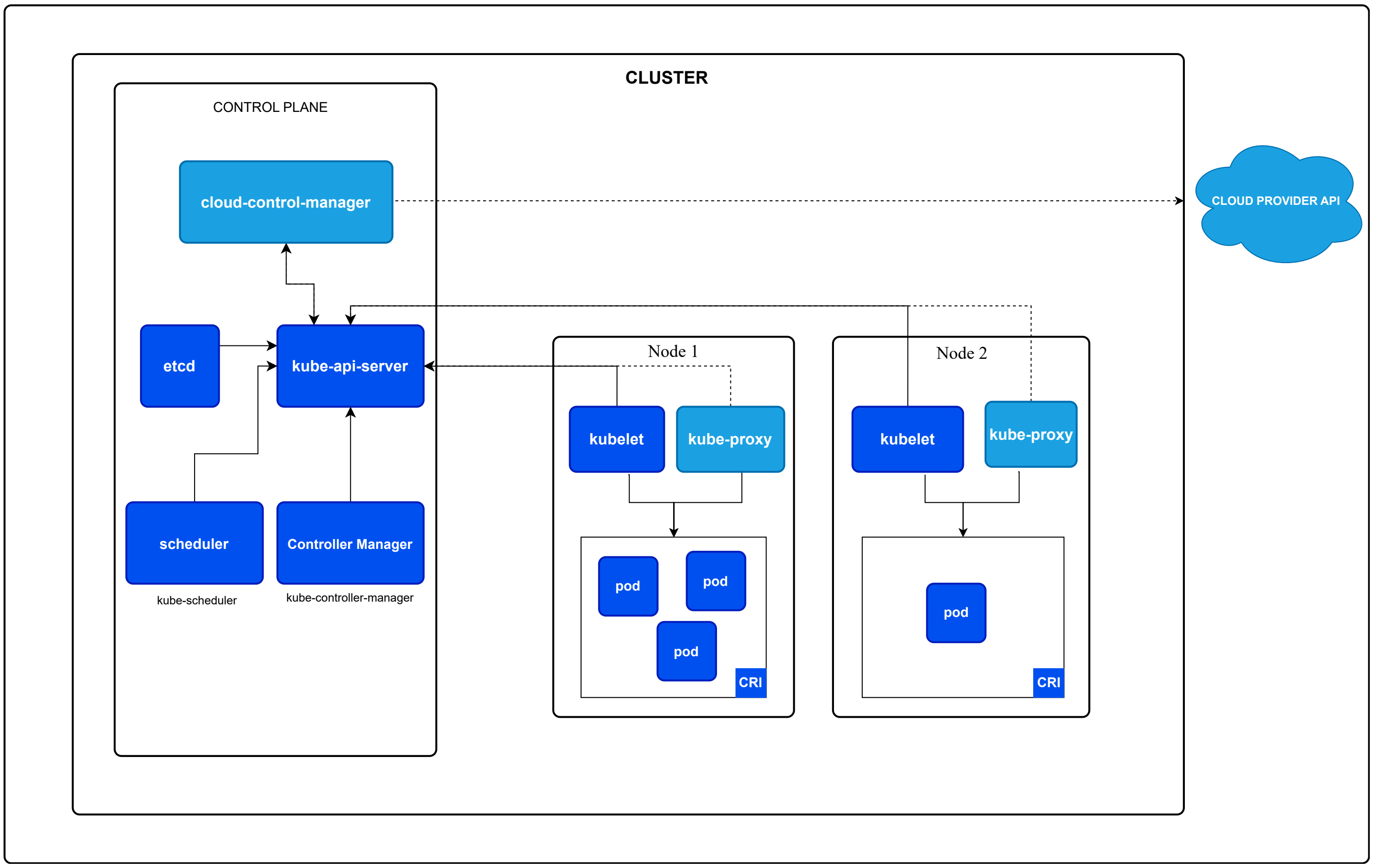
- updates needs to be performed manually

Kubernetes

nodeport = port opened in the
where app is running



Cluster

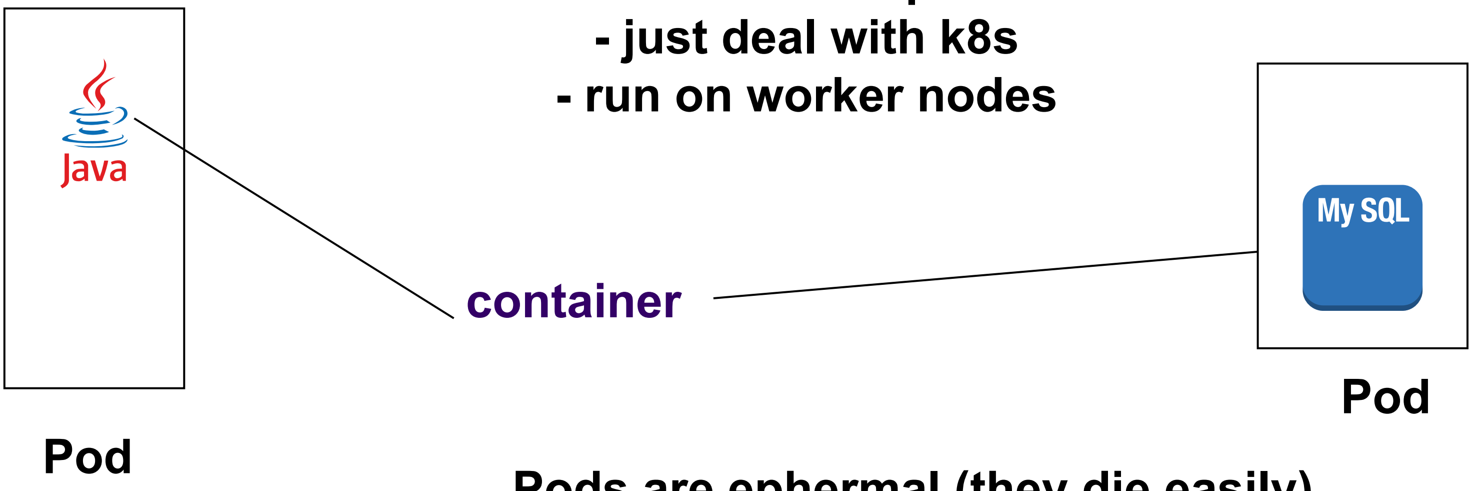


K8s API - UI / Kubectl / API

Master Node -
it has control plane component

FOr HA and Resource Managemnet - use AKS

Pod - Abstraction on top of containers
- just deal with k8s
- run on worker nodes



Pods are ephermal (they die easily)
- get new pvt ip when launched

Service [Permanent IP]: Attached to any pod
- is independent of pod's lifecycle
- External Service [Internet]
- Internal Service [Private]
- Load balancing

http://ip:port

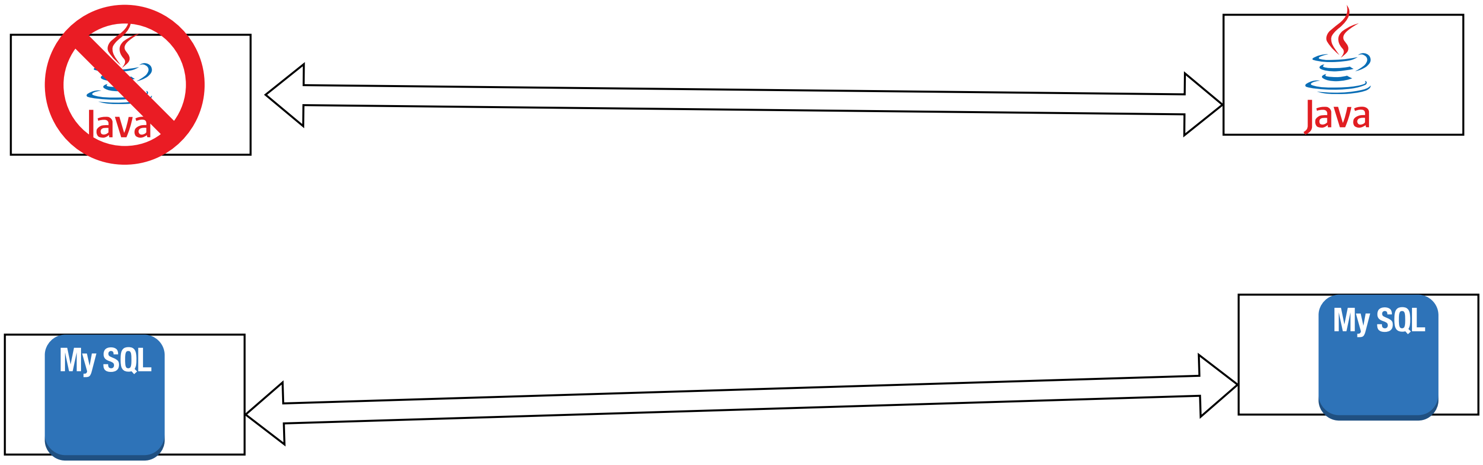
ingress (routing) service Pod
http://ip:port

my.compnay.com

Java connects (MySQL URL)
Authentication (Username
Password)

config map (external configs)
- DB URL
- stores in plain text

secret:
- better for password / username
- stores in base64

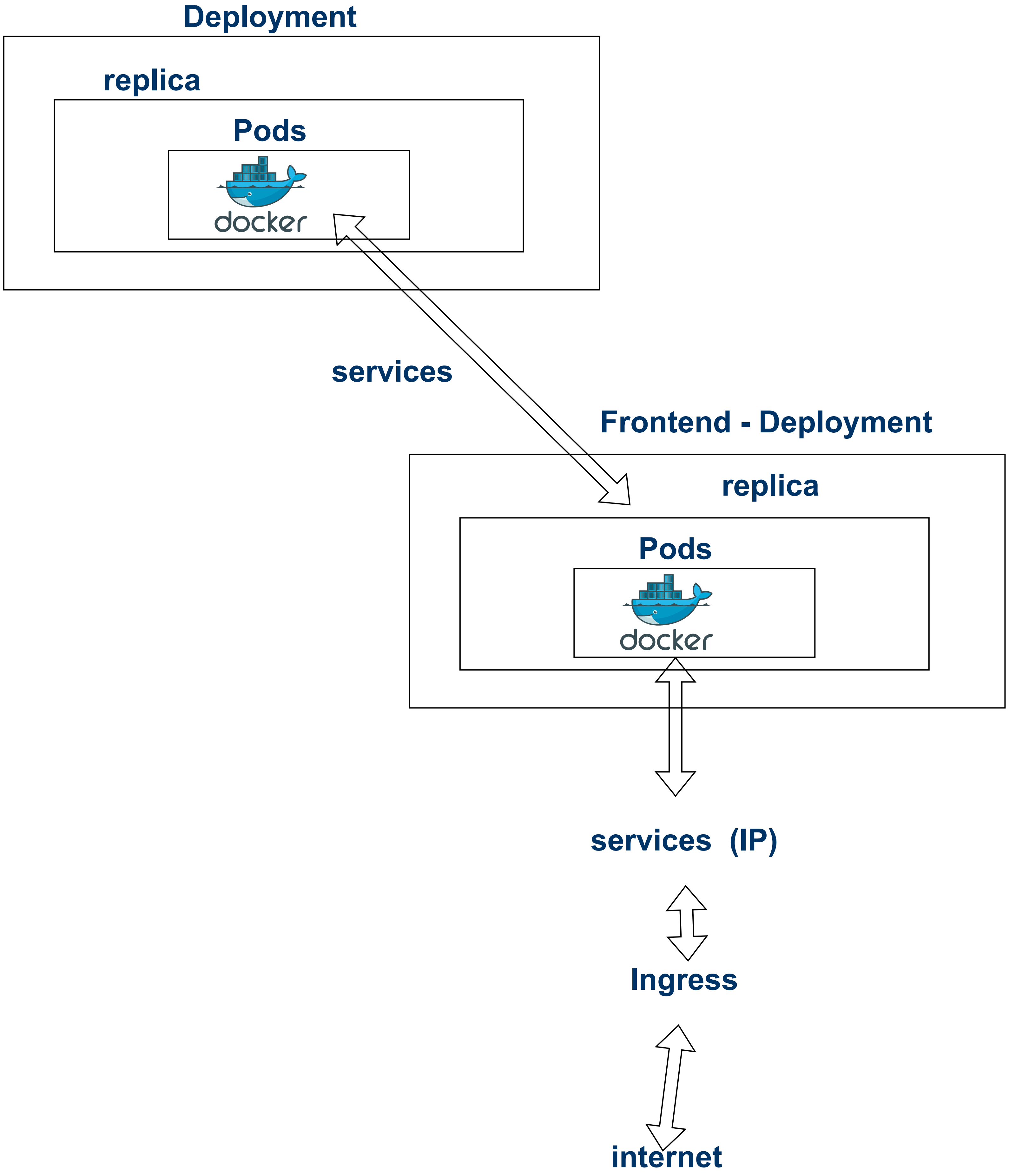


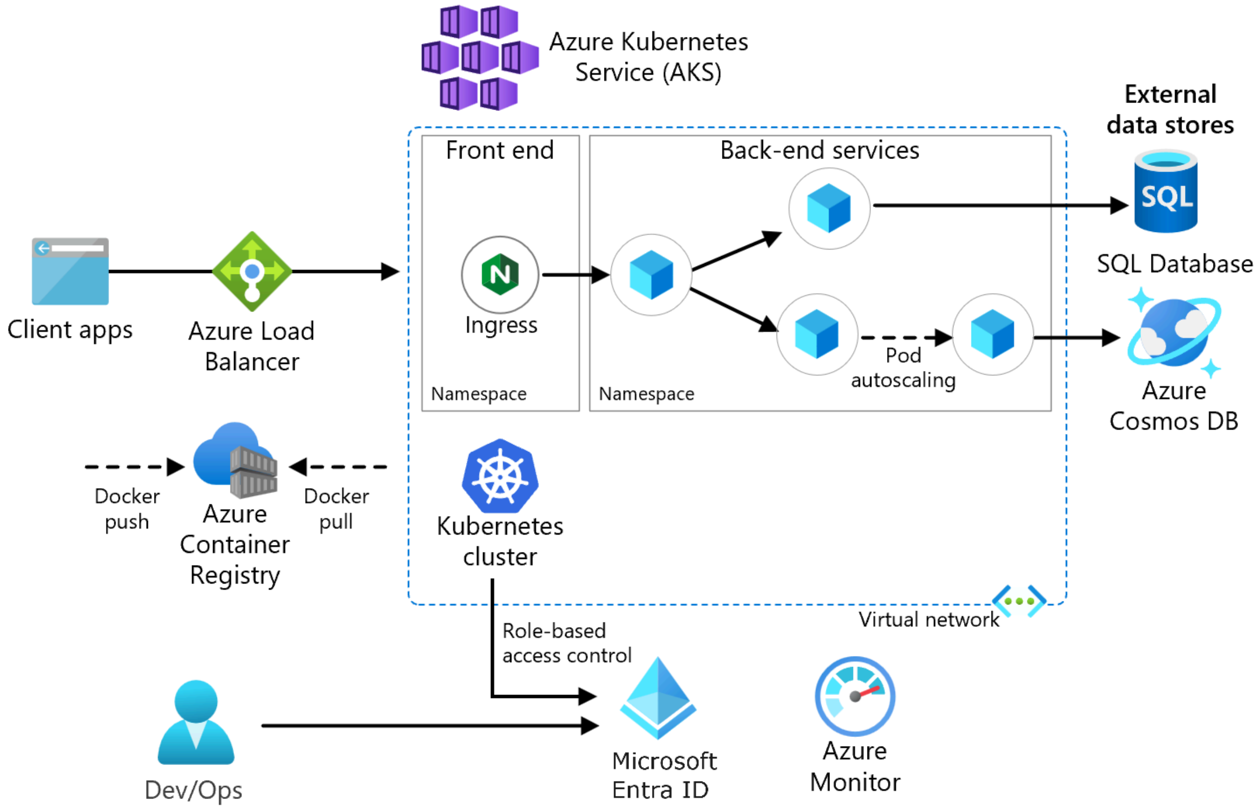
Replica = Copies

* Deployments - You don't create Pods
- we create deployments

deployment - stateless
statefulsets - stateful apps like Mysql

Both specify
how many
repica


Deployment
replica
Pods
docker
services
Frontend - Deployment
replica
Pods
docker
services (IP)
Ingress
internet



Principle of Least Privilege:
- IF you can't see
you can't destroy !

Namespace - organize cluster resources
like Pods, services etc
- most components are part of namespace

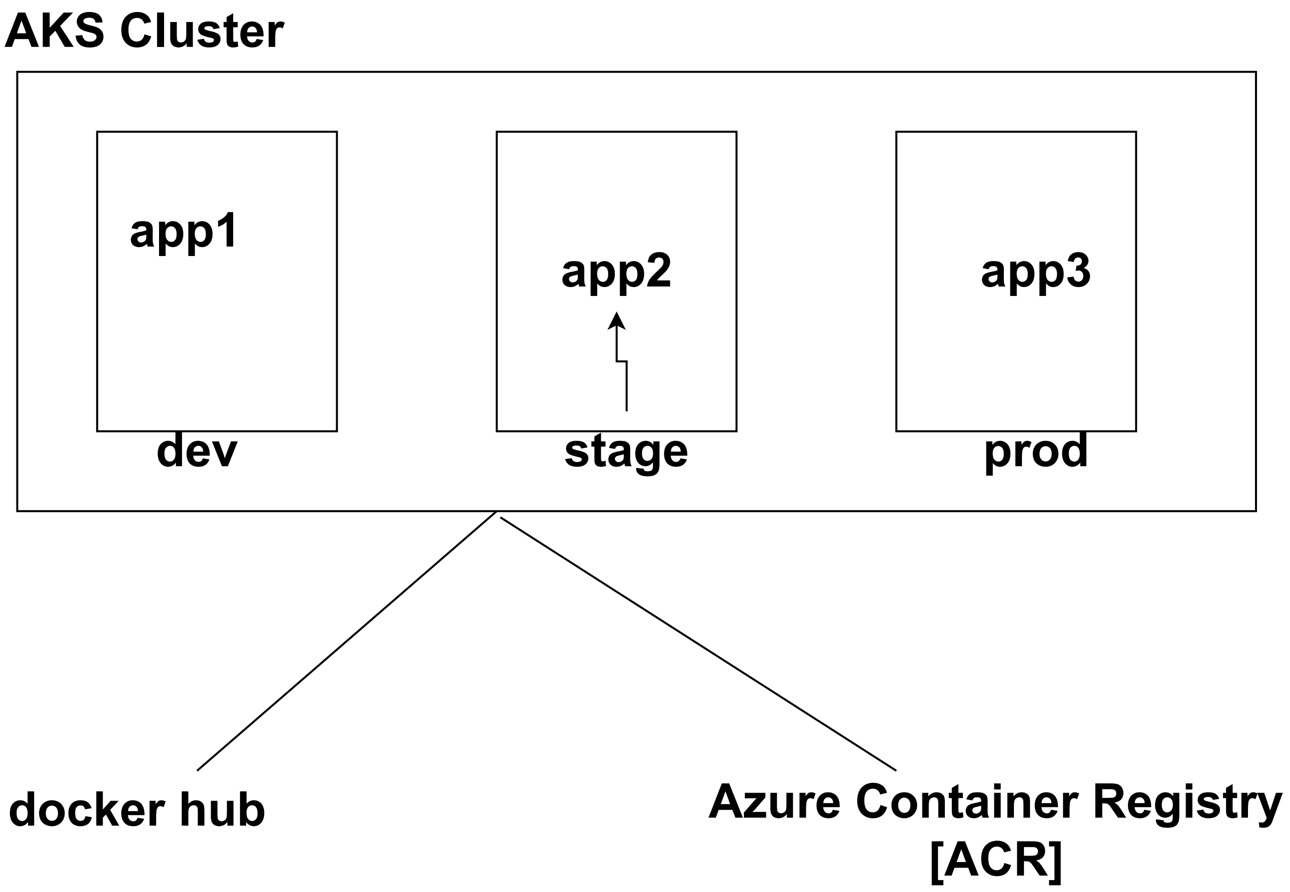
Node, volumes - not namespace

virtual network - helps all component connect

AKS will create and manage control plane (master node)

Customer ==> worker nodes - pods ($$$)

worker Nodes - Azure VM
Hybrid Benefit - License
Reservations - [Commit to use vm for 1 or 3 years
~ 40%-60%]



AKS Cluster

app1

dev

app2

stage

app3

prod

docker hub

Azure Container Registry
[ACR]

kubectl - cmd line tool [Imperative]

YAML file - Github, Azure Repo [VCS]
- recommended

Namespace

default:
Which is used by default to create resourcees

kube-node-lease:
heartbeat of pods

kube-public: visible to all users

kube-system: system [control plane]

When to create new namespace:

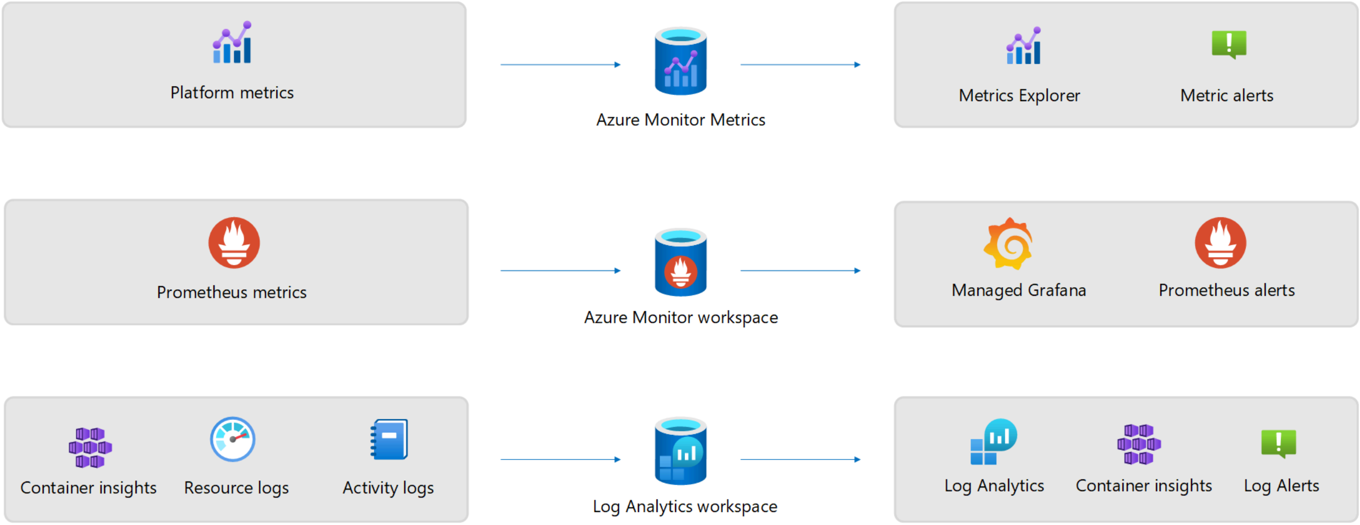
- **app wise**
- **team wise**
- **env wise**

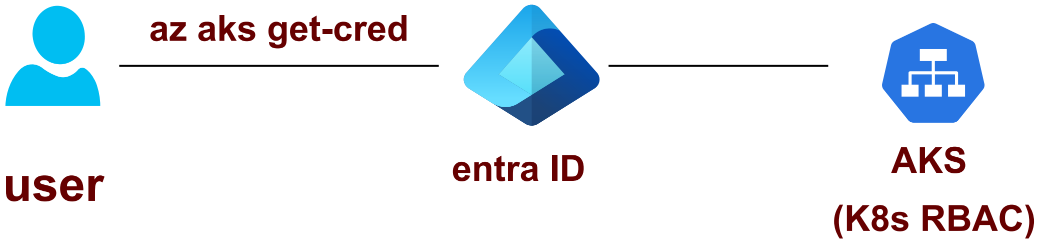
**** user restriction + resource (CPU / RAM) ****

Azure Monitring -

Azure Monitor Metrics = Nodes
COntainer Insights = Pods
[Code] Application Insight =

Frequency : how quick
retention -





- while creation of AKS

Horizontal Pod Auttoscaler - adds / removes pods

Cluster autoscaler: Add / removes nodes in cluster