# Photo Album

by N, Santosha

Unit testing

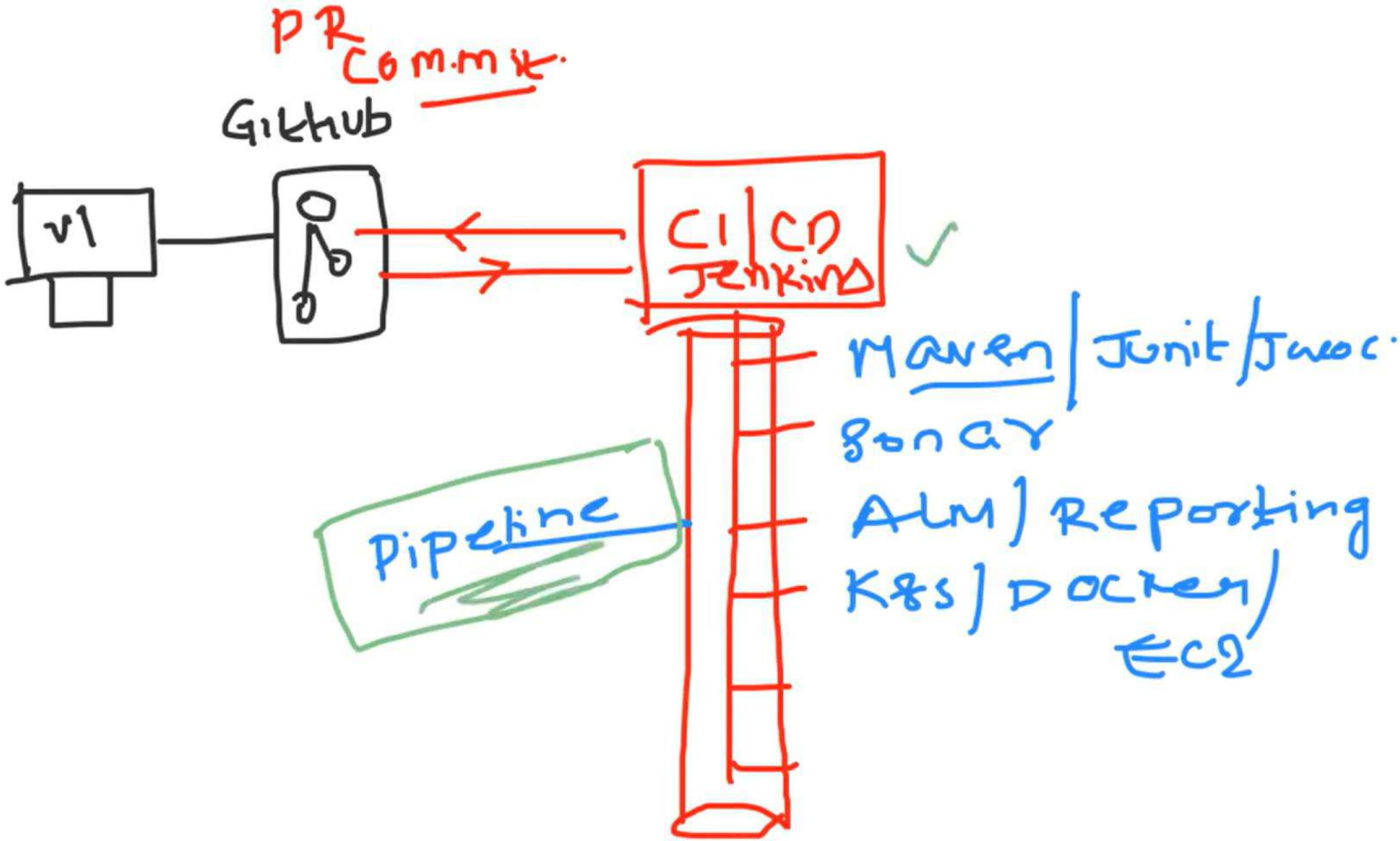
Static Code Analysis

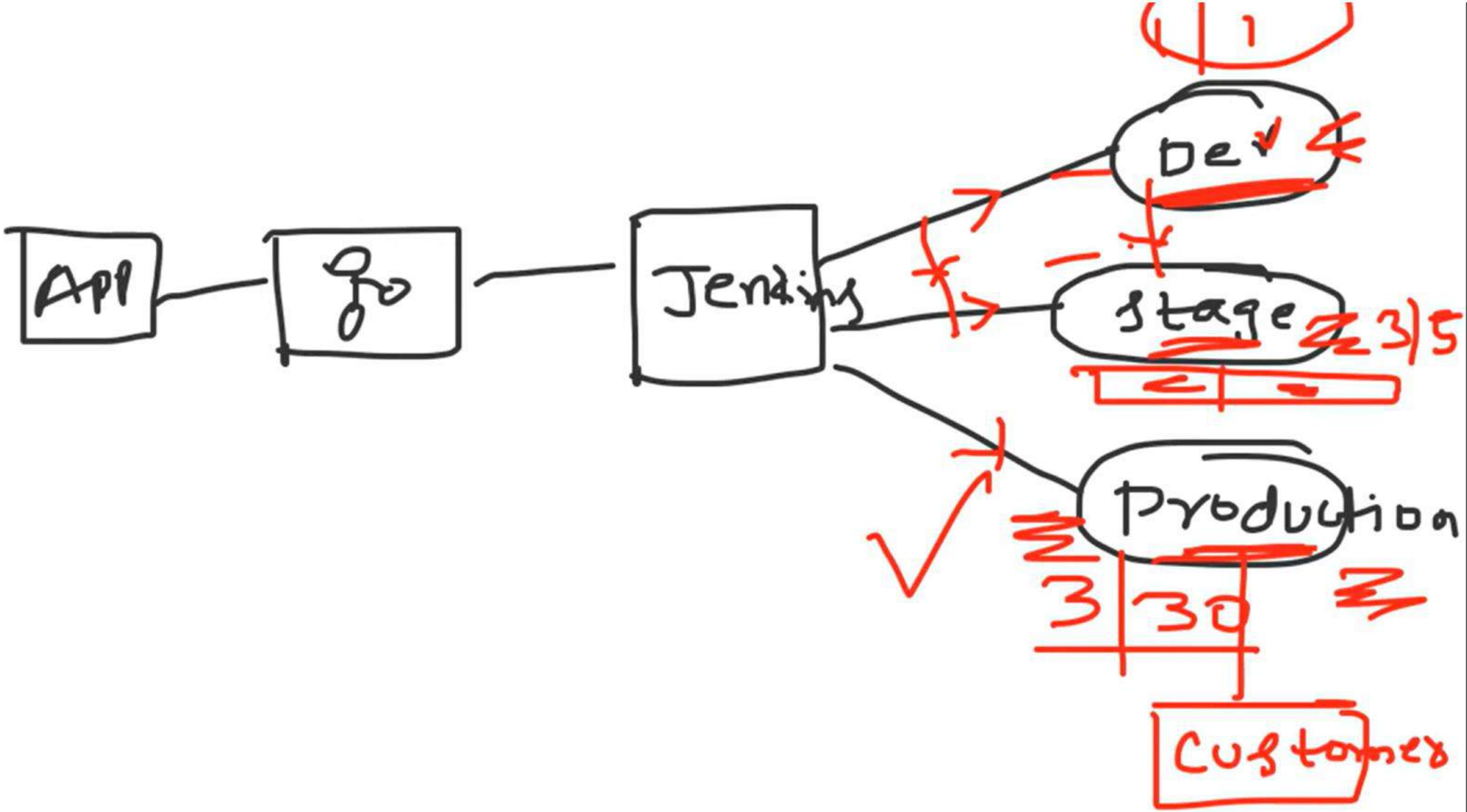
Code Quality /vulnerability

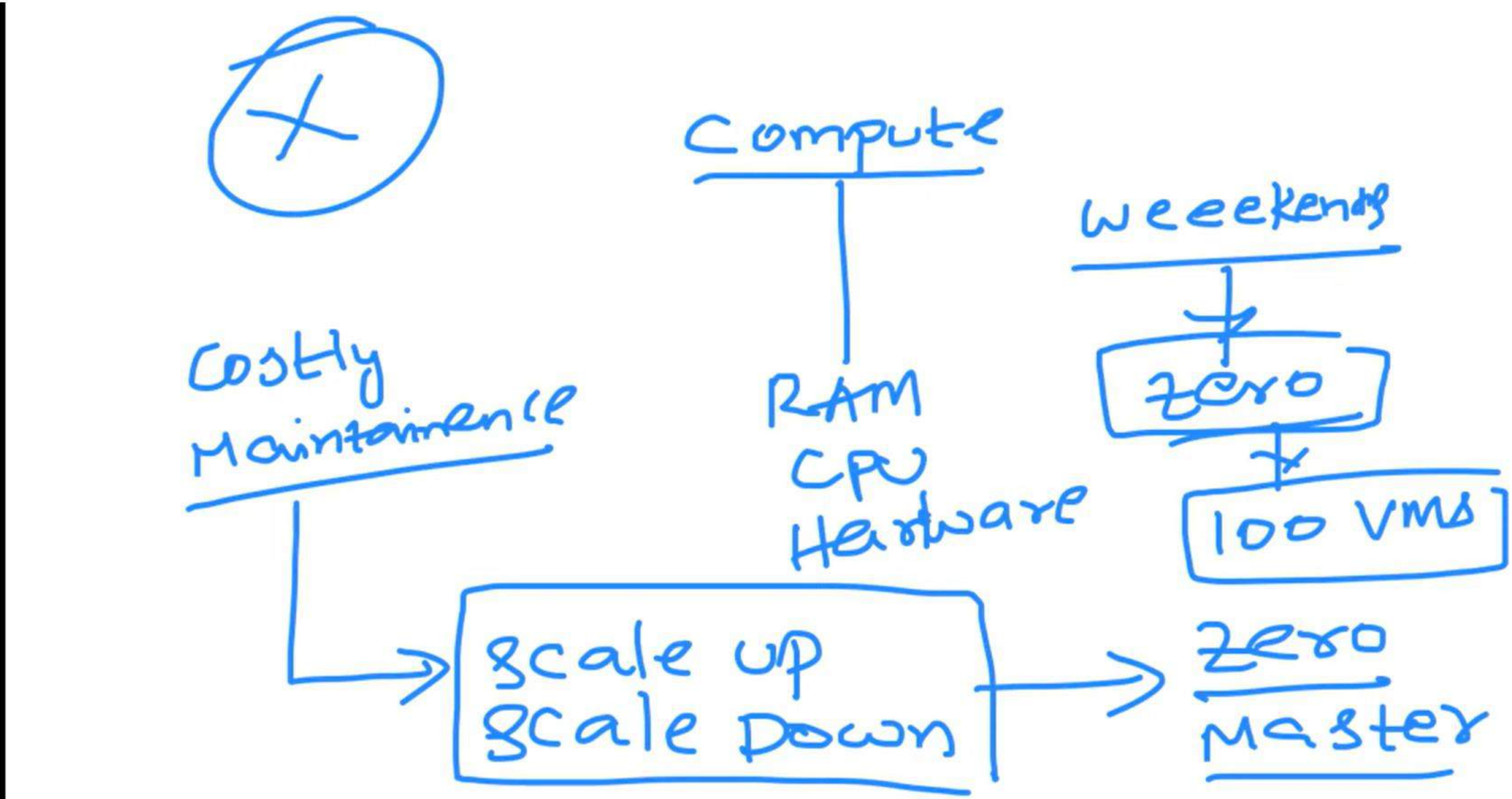
Automation

Reports

Deployment







## Question 7

## What are the networking types in Docker and what is the default ?

The default networking in Docker is Bridge.

However, you can change the default type and configure one of the

1. Bridge
2. Overlay
3. Host
4. MacVlan

## Question 8

## Can you explain how to isolate networking between the containers ?

## Question 7

## What are the networking types in Docker and what is the default ?

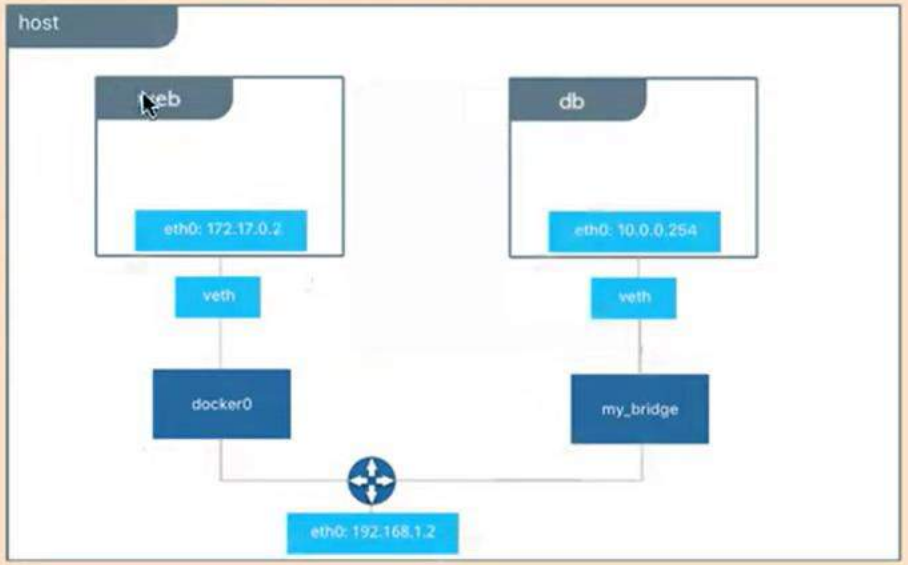
The default networking in Docker is Bridge.

However, you can change the default type and configure one of the

1. Bridge
2. Overlay
3. Host
4. MacVlan

## Question 8

## Can you explain how to isolate networking between the containers ?



## Question 9

## What is a multi stage build in Docker ?

Multi stage build allows you to build your docker container in multiple stages allowing you to copy artifacts from one stage to other. The major advantage of this is to build light weight containers.

## Question 9

## What is a multi stage build in Docker ?

Multi stage build allows you to build your docker container in multiple stages allowing you to copy artifacts from one stage to other. The major advantage of this is to build light weight containers.

## Question 10

## What are distro less images in Docker ?

Distroless images contain only your application and its runtime dependencies with a very minimum operating system libraries. They do not contain package managers, shells or any other programs you would expect to find in a standard Linux distribution. They are very small and lightweight images.

# 11. Real Time Challenges with Docker ?

- Docker is a single daemon process. Which can cause a single point of failure, If the Docker Daemon goes down for some reason all the applications are down.

- Docker Daemon runs as a root user. Which is a security threat. Any process running as a root can have adverse effects. When it is comprised for security reasons, it can impact other applications or containers on the host.

- Resource Constraints: If you're running too many containers on a single host, you may experience issues with resource constraints. This can result in slow performance or crashes.

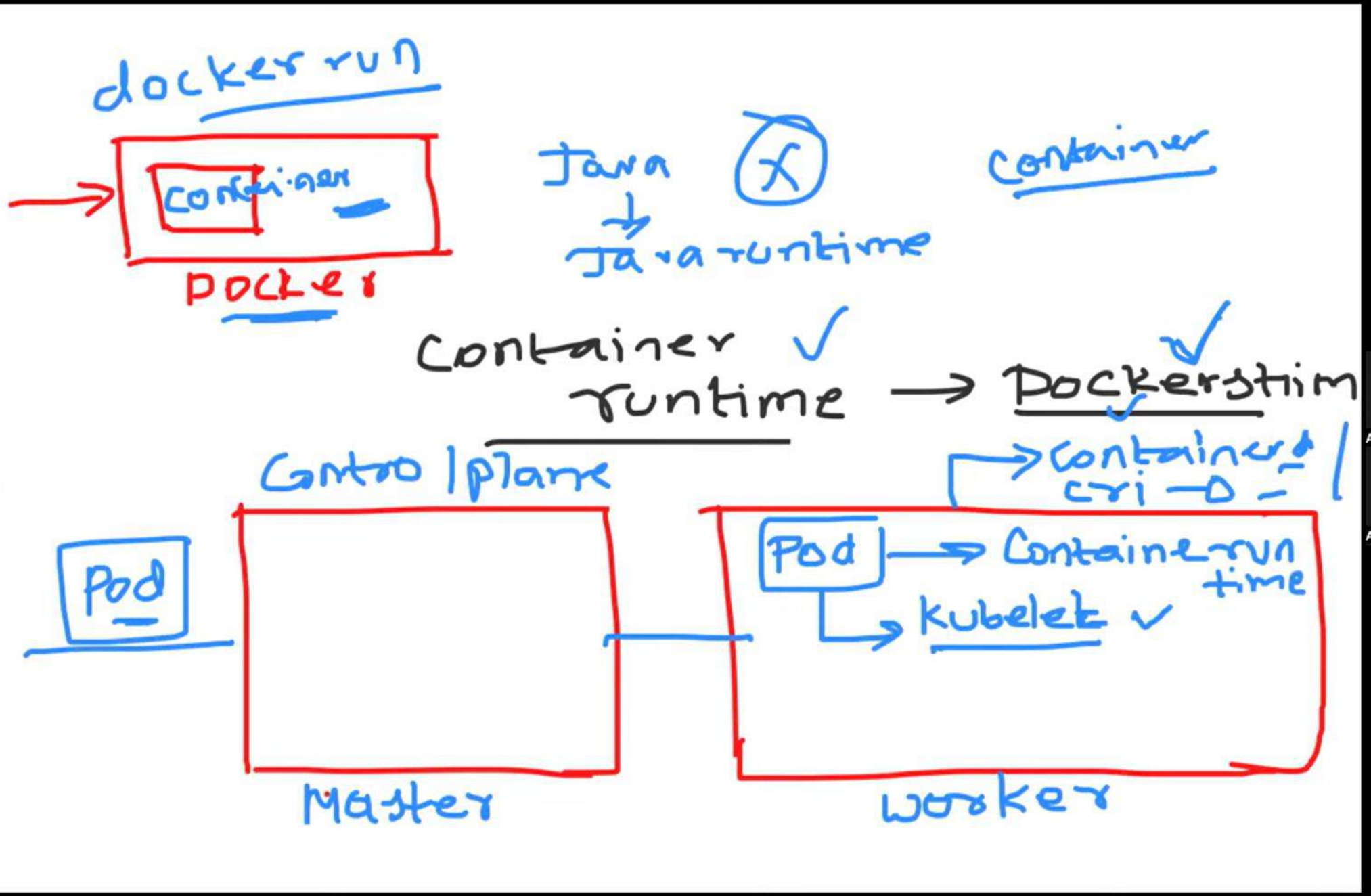
# 12. What steps would you take to secure containers ?

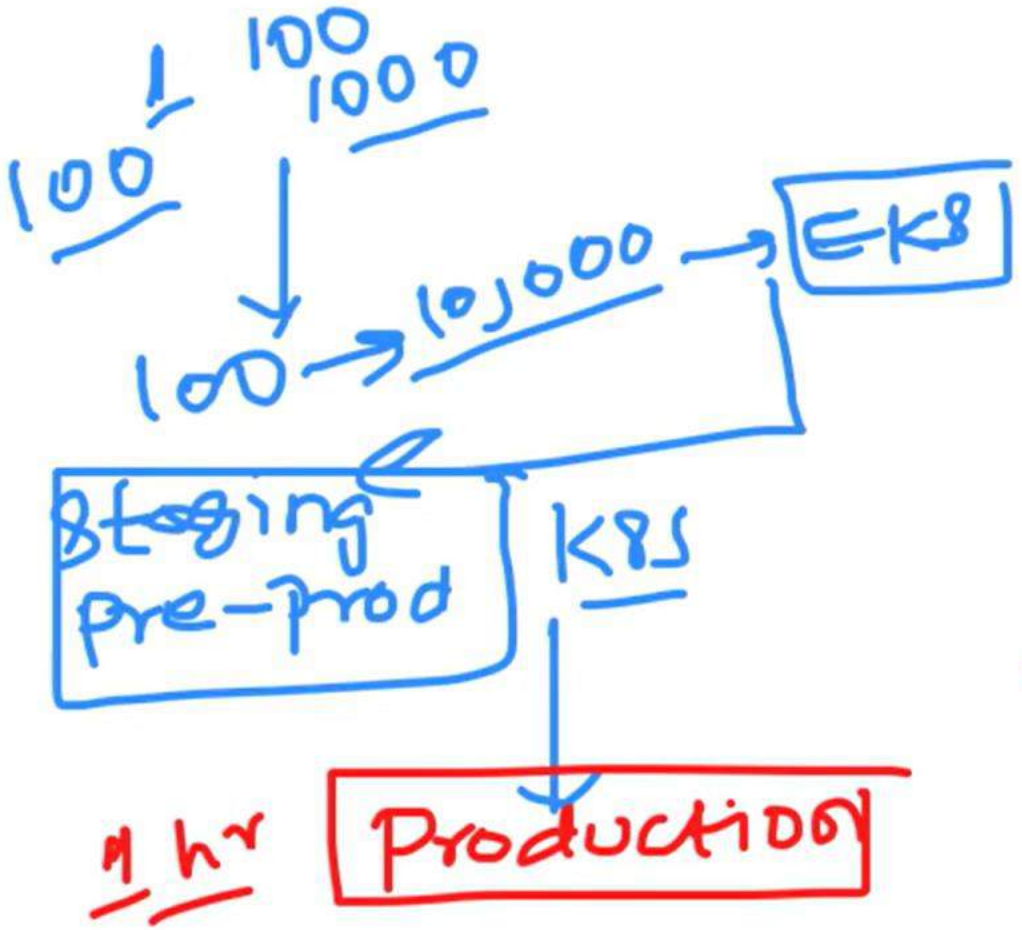
Some of the steps,

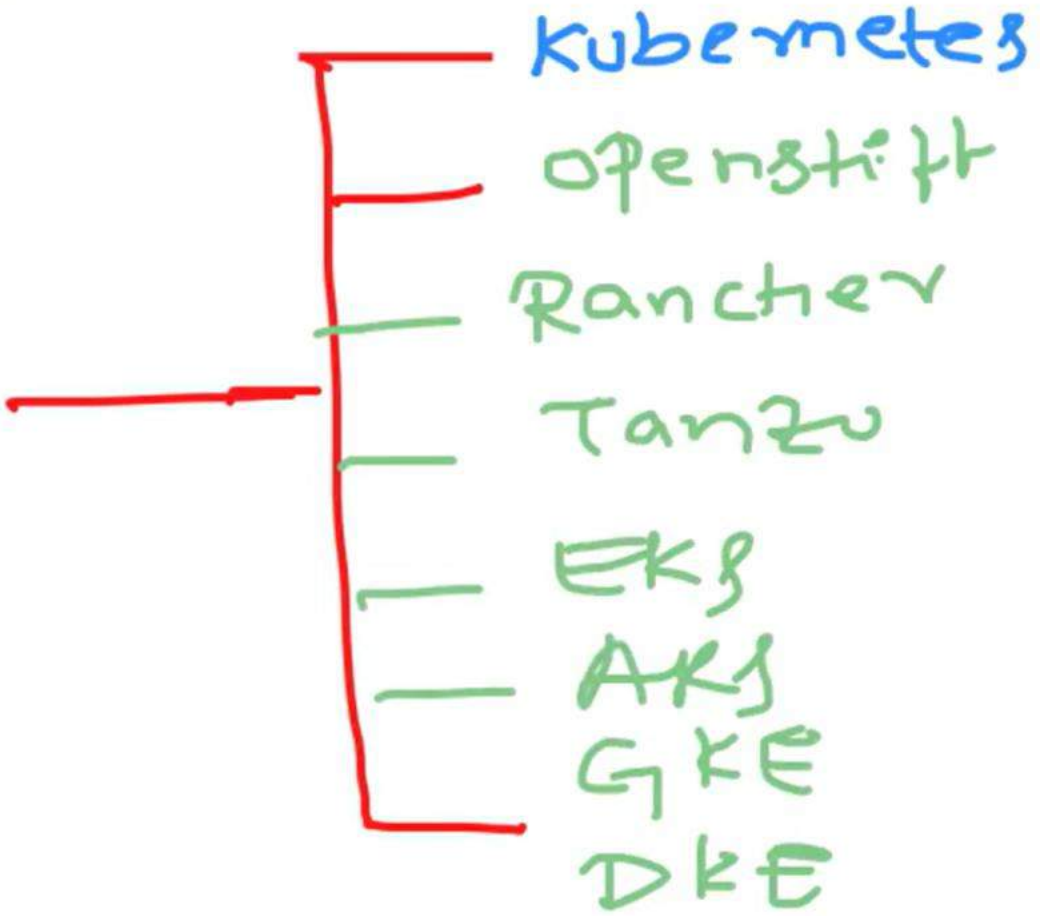
1. Use Distroless or Images with not too many packages as your final image in multi stage build, so that there is less chance of CVE or security issues.

2. Ensure that the networking is configured properly. This is one of the most common reasons for security issues. If required configure custom bridge networks and assign them to isolate containers.

3. Use utilities like Sync to scan your container images.



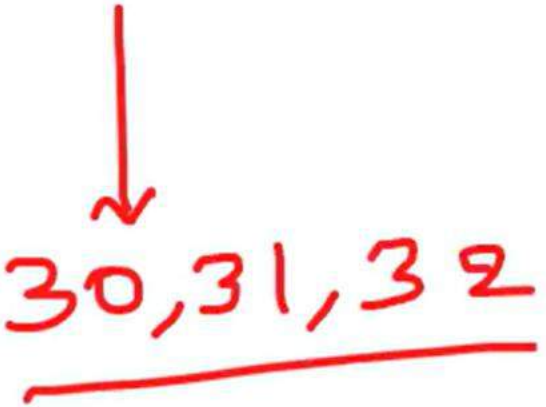

100
1000
100
100
10,000
EKS
Staging
pre-prod
K8S
1 hr
Production


Kubernetes
openshift
Rancher
Tanzu
EKS
AKS
GKE
DKE

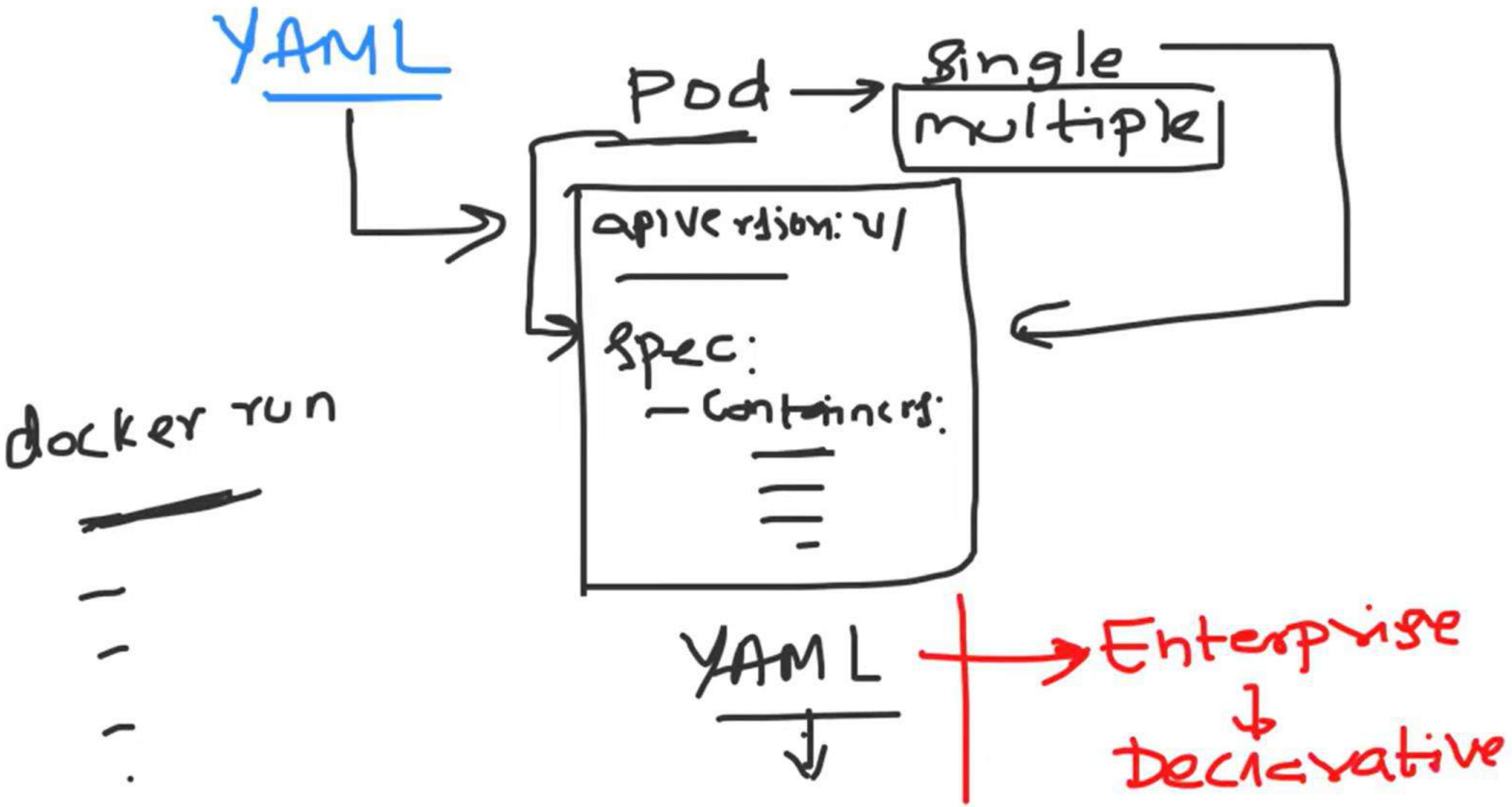

Abhishek Veeramalla
Abhishek Veeramalla
Abhishek Veeramalla

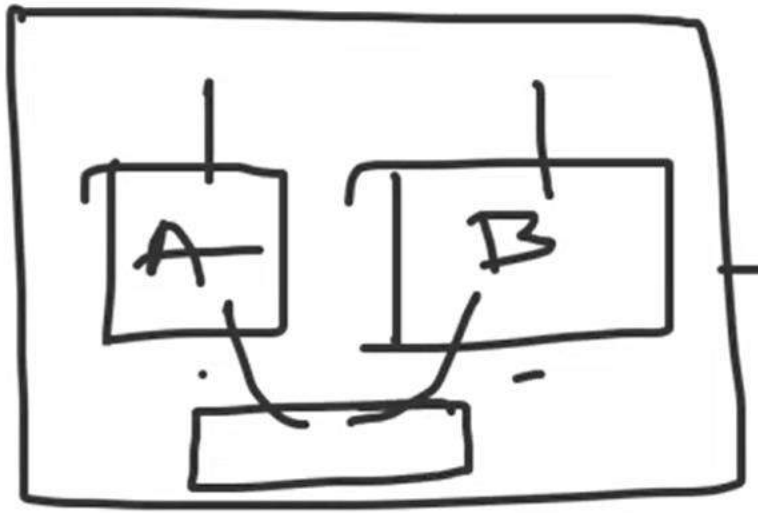
```
➜  Downloads aws route53 create-hosted-zone --name dev.example.com --caller-refer
ence 1
```

# Deploy your first app




Day 33


YAML
Pod
Single
multiple
apiVersion: v1
spec:
- containers:
docker run
YAML
Enterprise
Declarative




Pod

↓

Shared Network

Shared Storage

localhost: 3000

↓

3000 → Sidecar

# Pod

1. Kubectl

↓

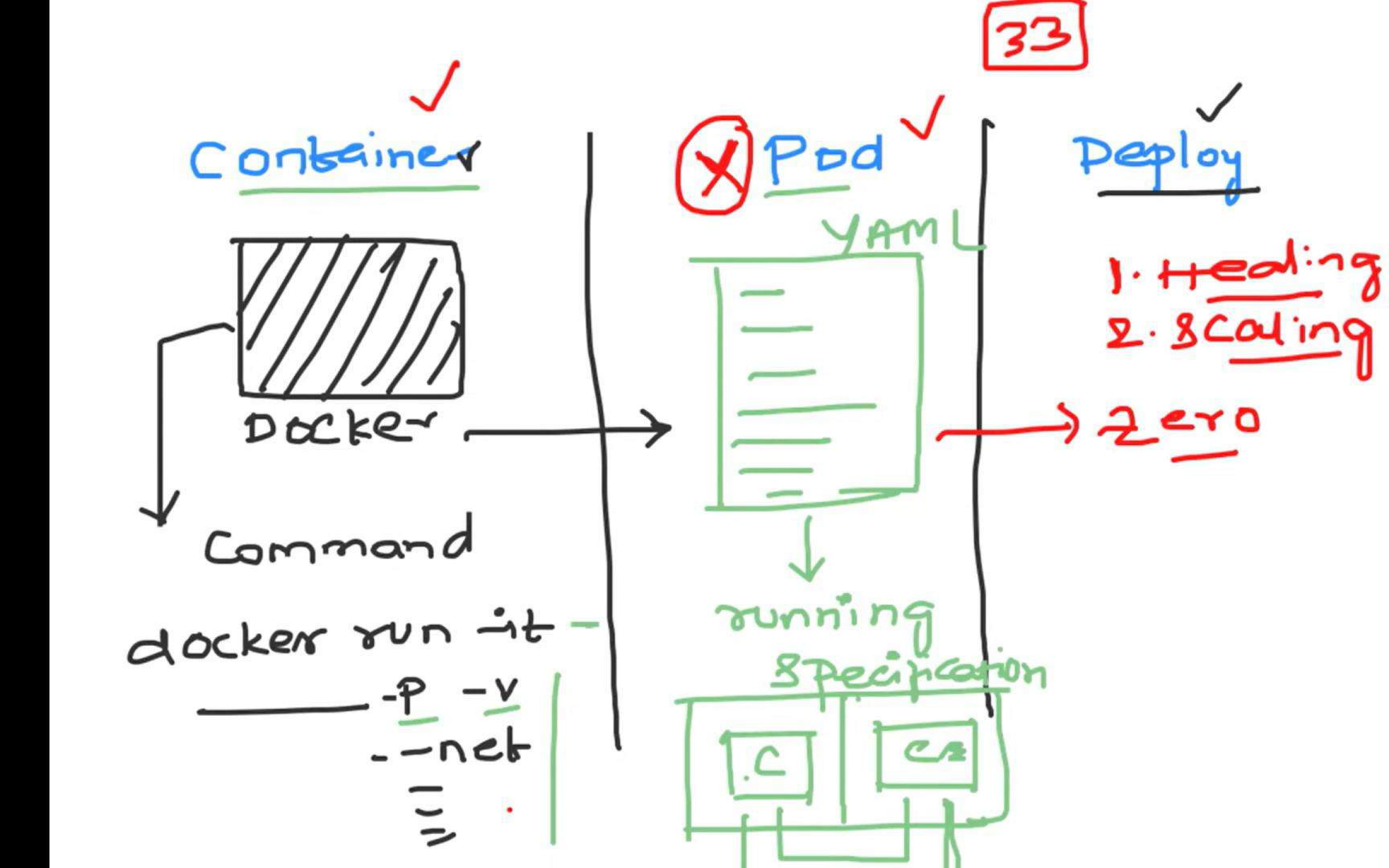
What ?

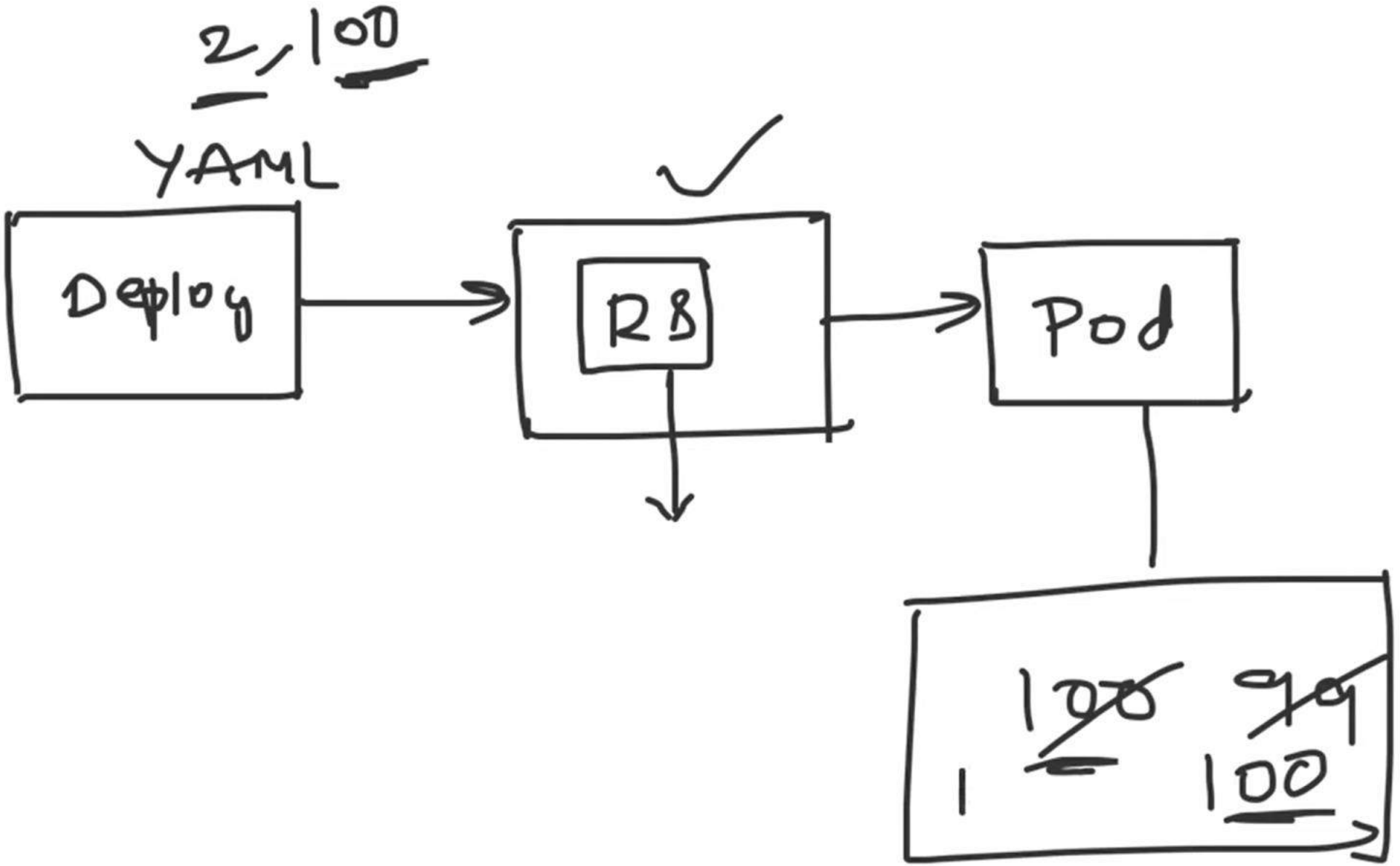
→ docker CLI

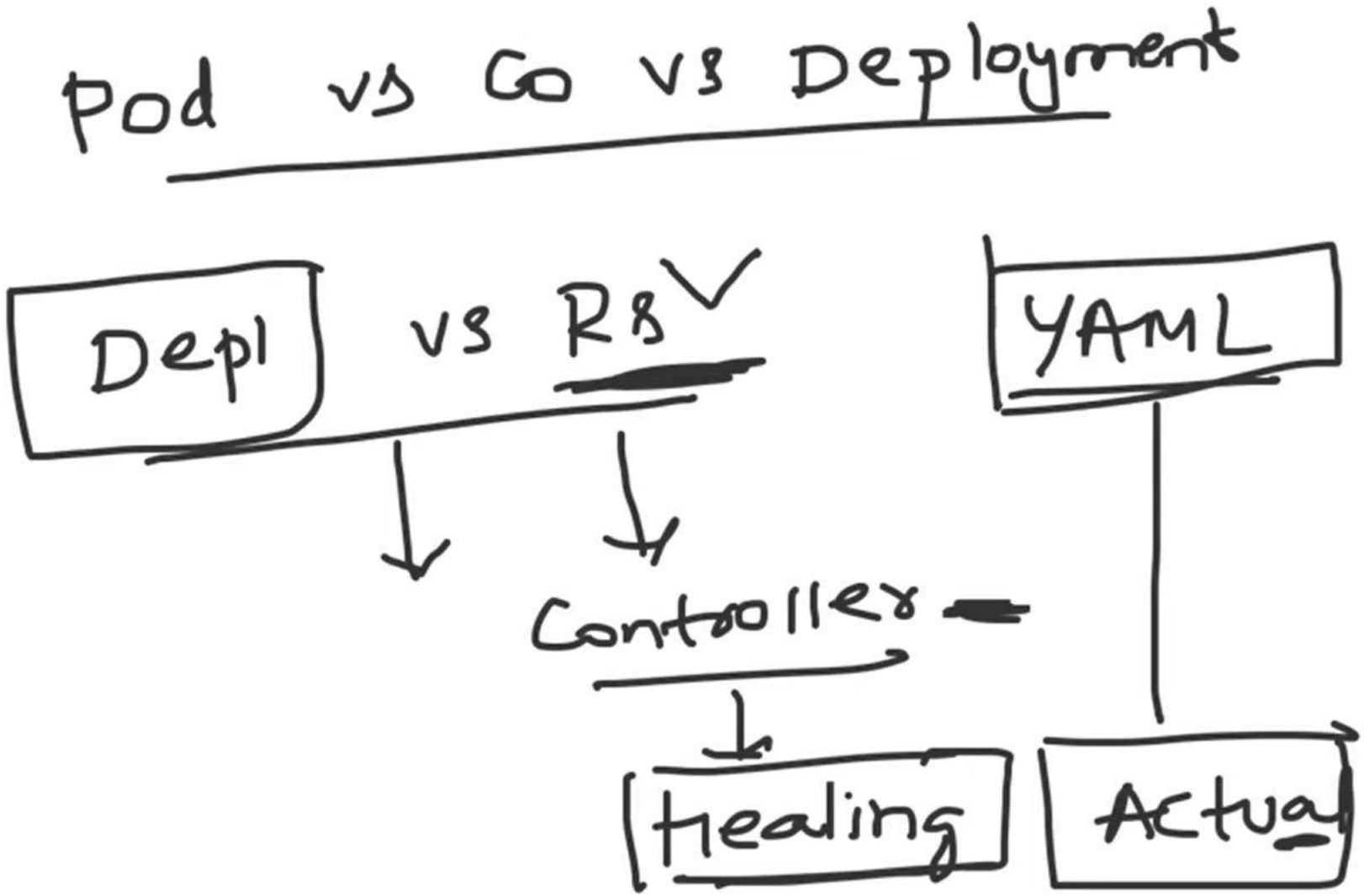
Command line

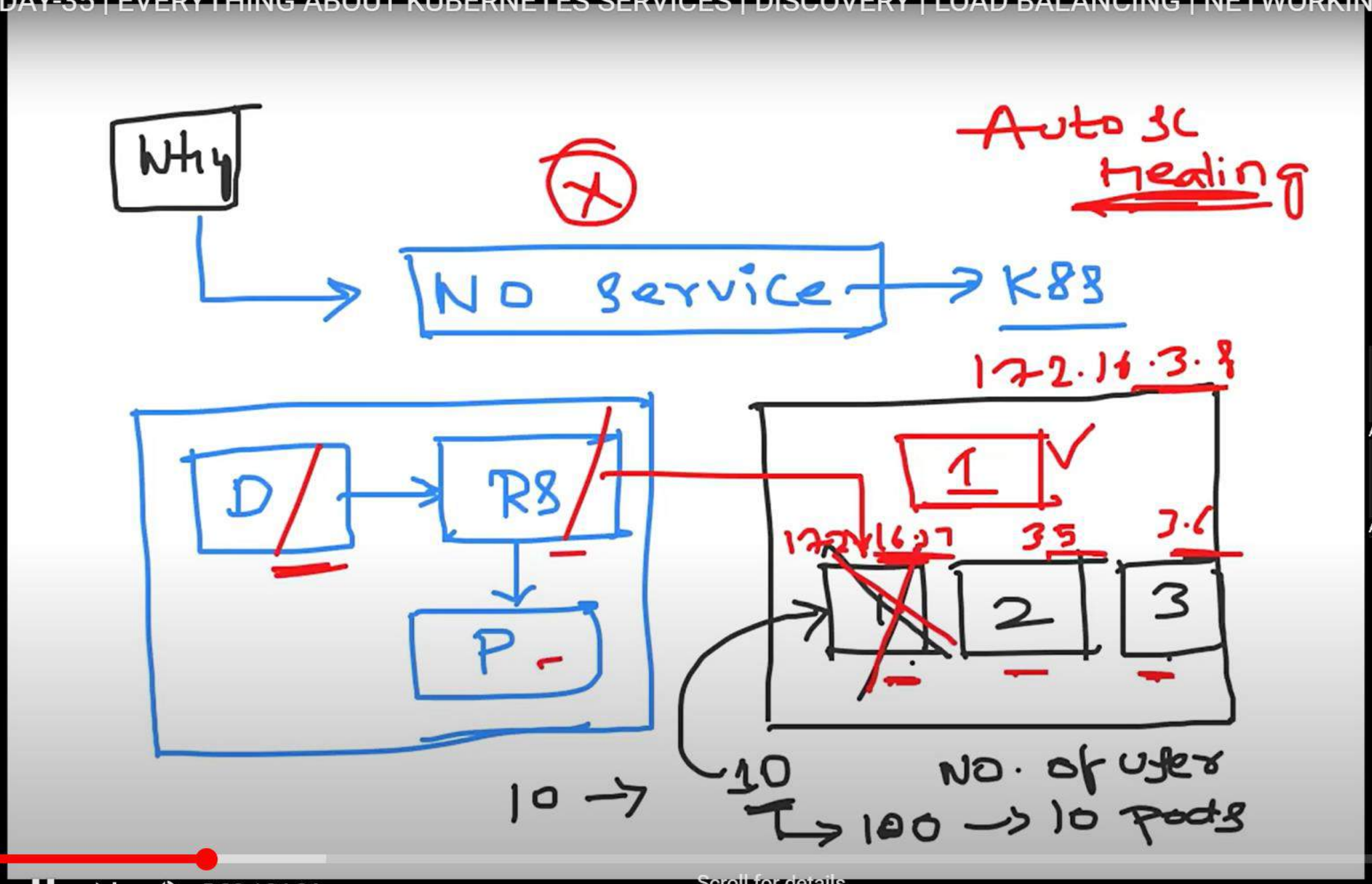
↓

10 Nodes → Kubecl get nodes


33
Container
Pod
Deploy
YAML
1. Healing
2. Scaling
Docker
Zero
Command
docker run -it
-p -v
--net
running
Specification
C
C2

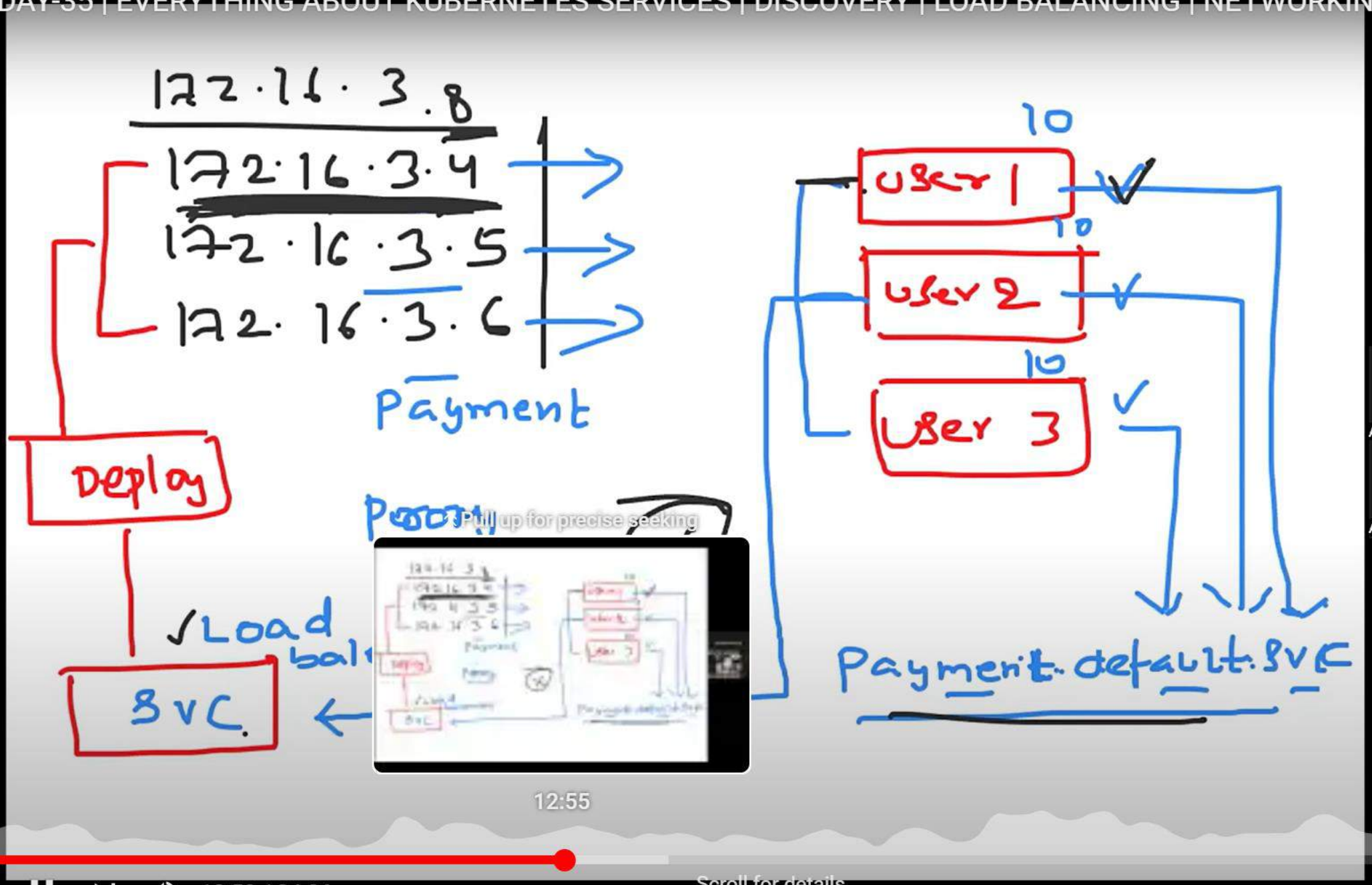

2, 100
YAML
Deploy
RS
Pod
100
99
1
100






Abhishek Veeramalla
Abhishek Veeramalla
Abhishek Veeramalla

DAY-35 | EVERYTHING ABOUT KUBERNETES SERVICES | DISCOVERY | LOAD BALANCING | NETWORKING | #de...




Abhishek Veeramalla

Abhishek Veeramalla



Abhishek Veeramalla

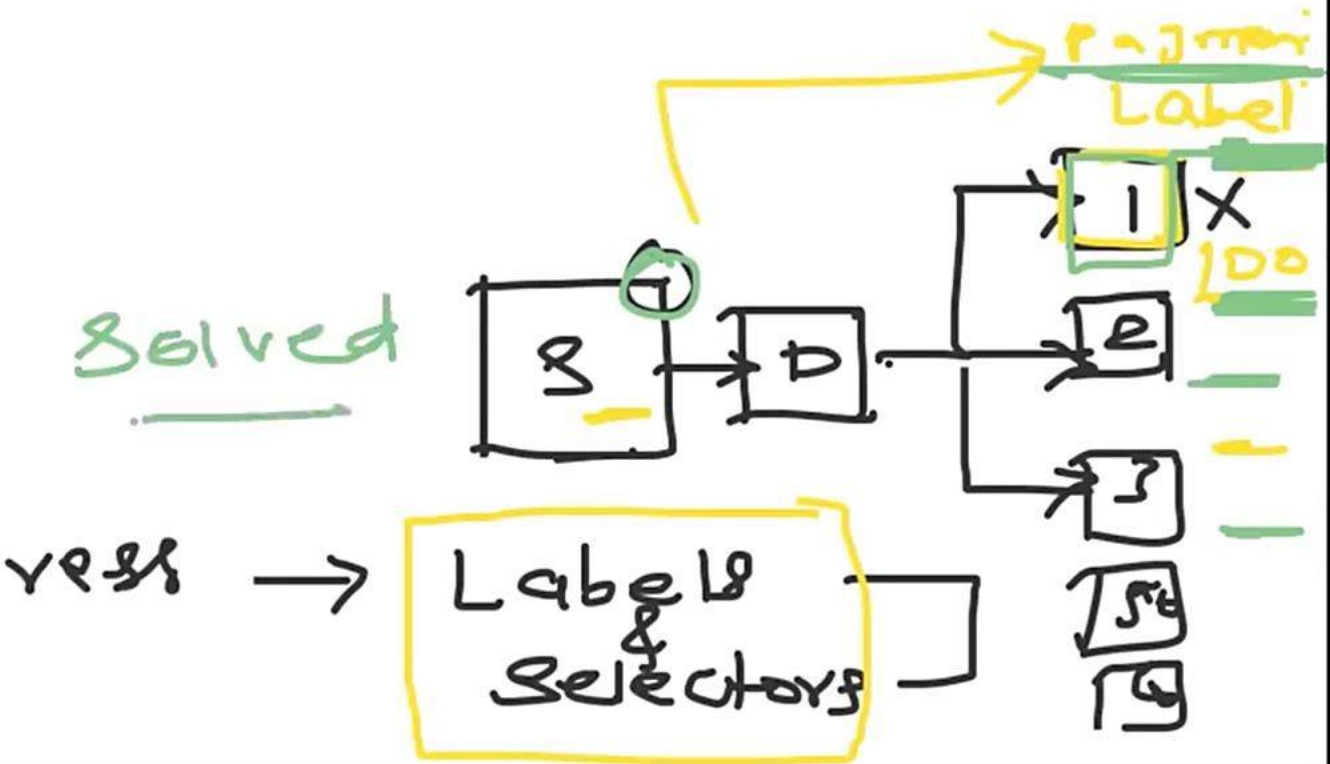
12:55

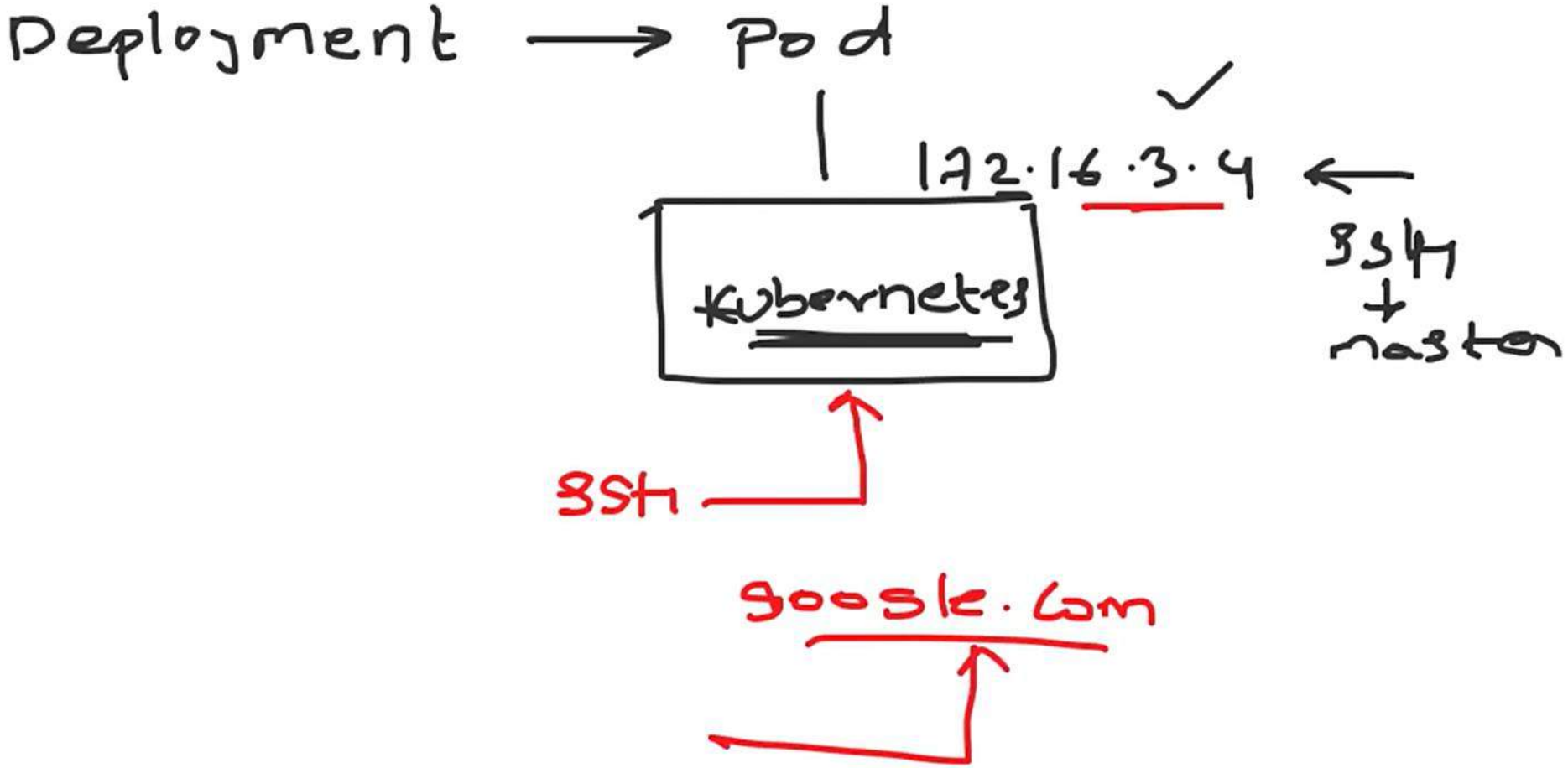
12:58 / 34:36

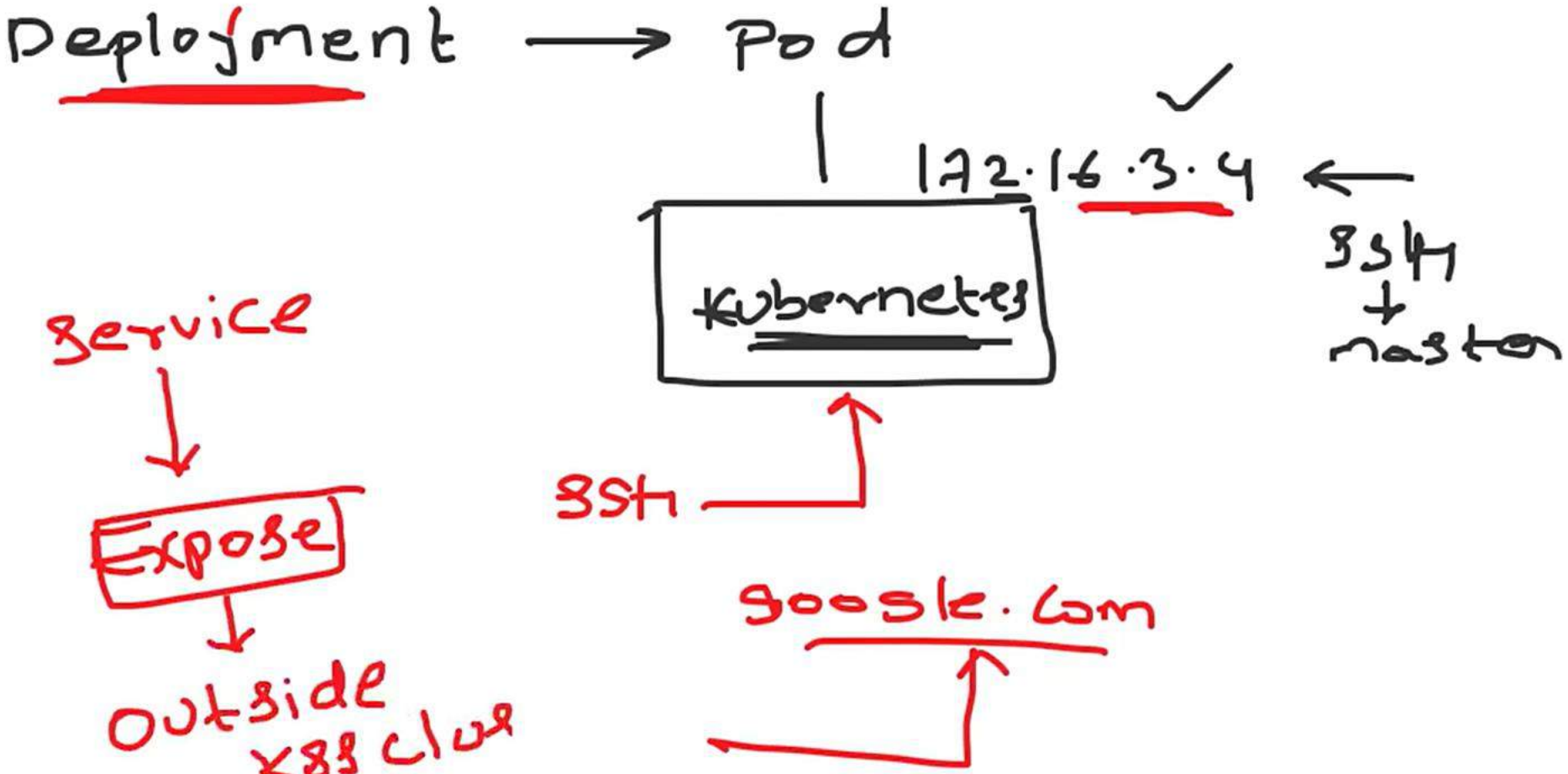
Scroll for details

Svc → 1. Load balancing

2. Service Discovery






Deployment
Pod
172.16.3.4
Kubernetes
ssh
master
Service
Expose
Outside
K8s cluster
ssh
google.com


Abhishek Veeramalla
Abhishek Veeramalla
Abhishek Veeramalla

YAML

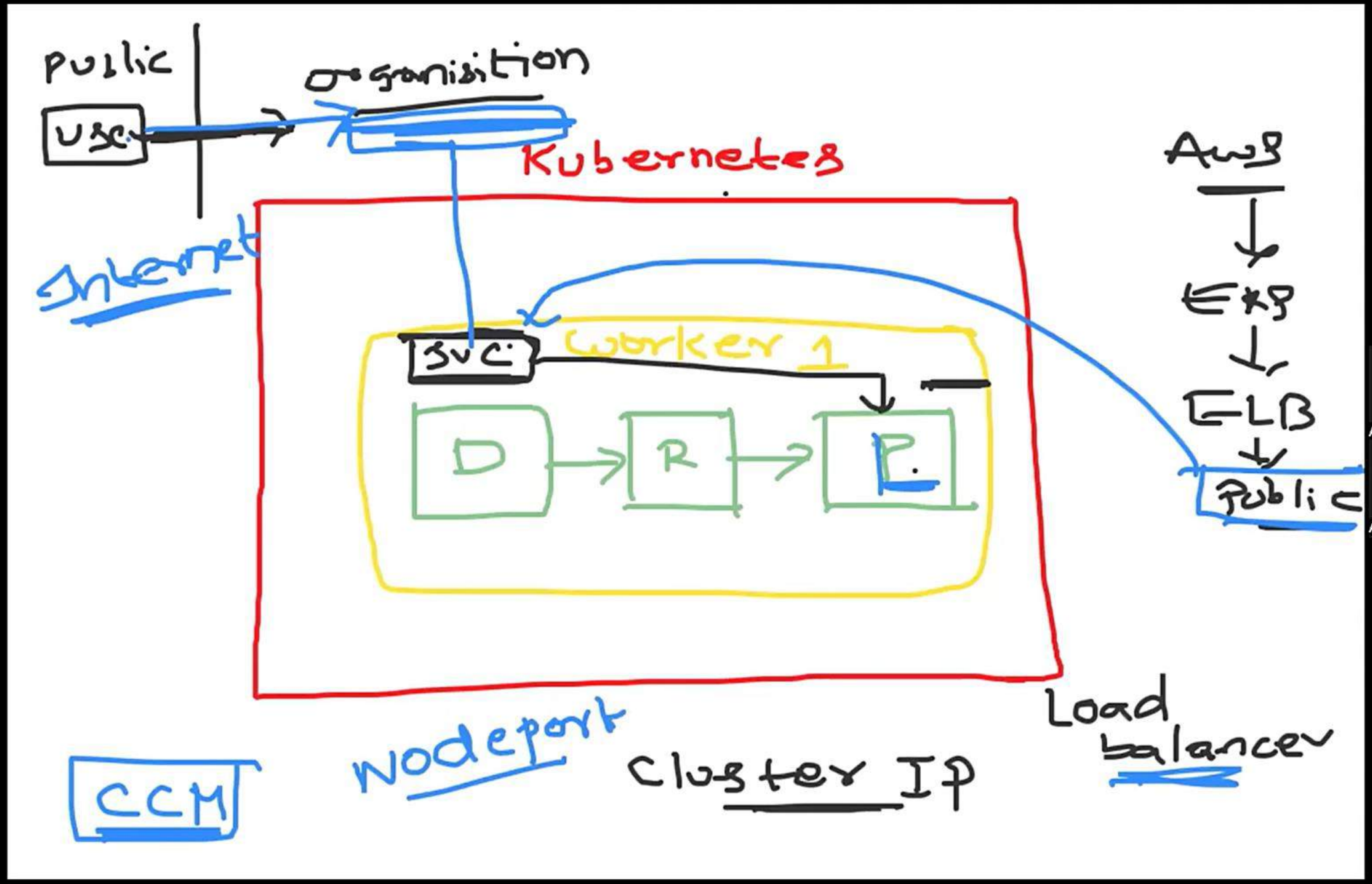
Discovery
Load balancing

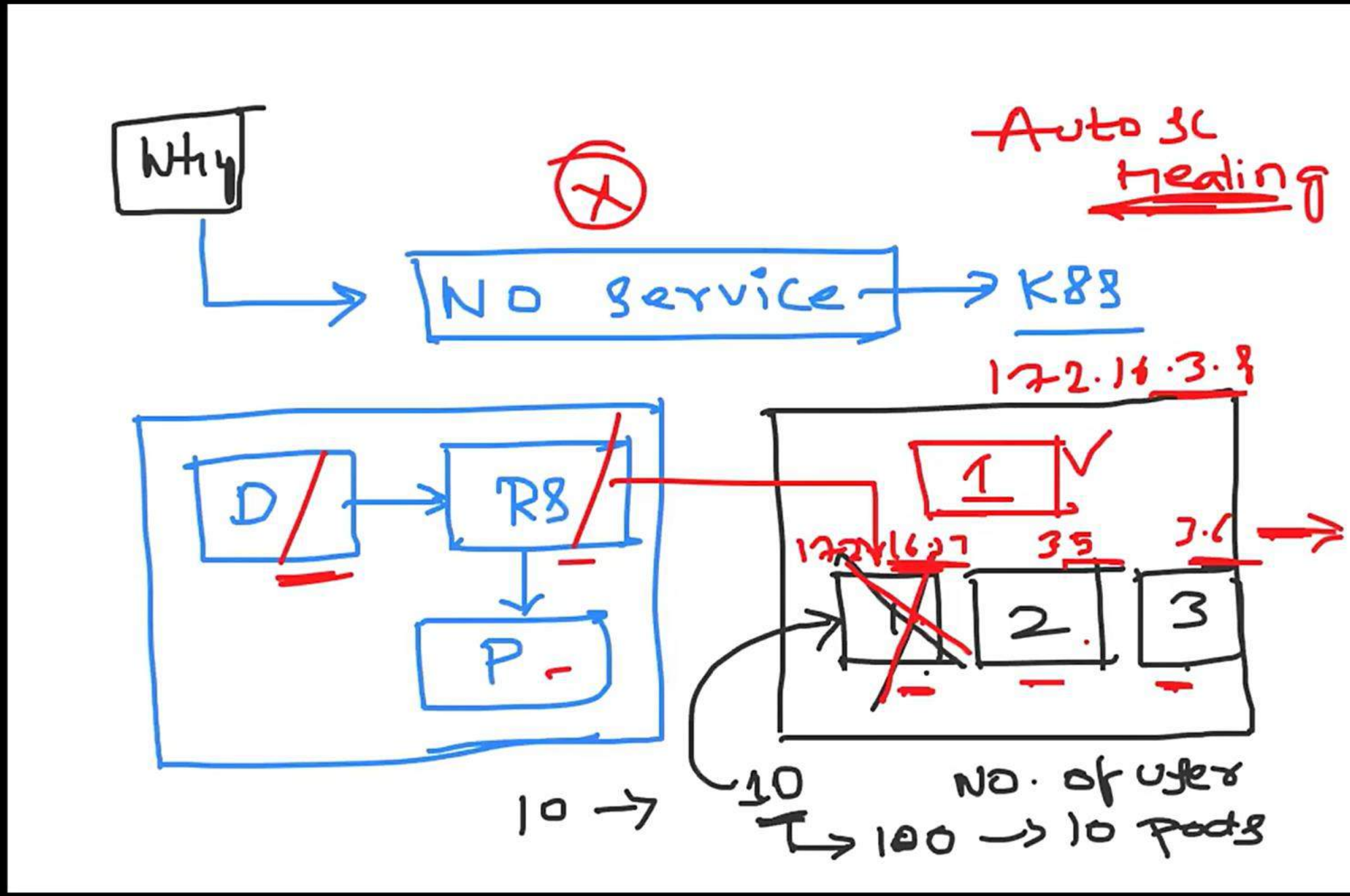
1. Clusterip → inside
2. Nodeport → inside org.
3. Loadbalancer → enter no1 world

worker node

EKS

payments-defaultsvc → ELB → Public address.





Load balancer

Amazon.com ✓

Nodepool VPC

Nodes

Cluster IP Cluster network

## Question 1

## What is the difference Docker and Kubernetes ?

Docker is a container platform where as Kubernetes is a container orchestration environment that offers capabilities like Auto healing, Auto Scaling, Clustering and Enterprise level support like Load balancing.

## Question 1

## What is the difference Docker and Kubernetes ?

Docker is a container platform where as Kubernetes is a container orchestration environment that offers capabilities like Auto healing, Auto Scaling, Clustering and Enterprise level support like Load balancing.

## Question 2

## What are the main components of Kubernetes architecture?

On a broad level, you can divide the kubernetes components in two parts

1. Control Plane (API SERVER, SCHEDULER, Controller Manager, C-CM, ETCD)
2. Data Plane (Kubelet, Kube-proxy, Container Runtime)

## Question 3

**What are the main differences between the Docker Swarm and Kubernetes?**

**Kubernetes is better suited for large organisations as it offers more scalability, networking capabilities like policies and huge third party ecosystem support.**

## Question 4

**What is the difference between Docker container and a Kubernetes pod ?**

**A pod in kubernetes is a runtime specification of a container in docker. A pod provides more declarative way of defining using YAML and you can run more than one container in a pod.**

# Question 5

## What is a namespace in Kubernetes ?

In Kubernetes namespace is a logical isolation of resources, network policies, rbac and everything. For example, there are two projects using same k8s cluster. One project can use ns1 and other project can use ns2 without any overlap and authentication problems.

## Question 7

## What are the different types of services within Kubernetes?

**There are three different types of services that a user can create.**

1. **Cluster IP Mode**
2. **Node Port Mode**
3. **Load Balancer Mode**

# Question 8

## What is the difference between NodePort and LoadBalancer type service ?

When a service is created a NodePort type, The kube-proxy updates the IPTables with Node IP address and port that is chosen in the service configuration to access the pods.

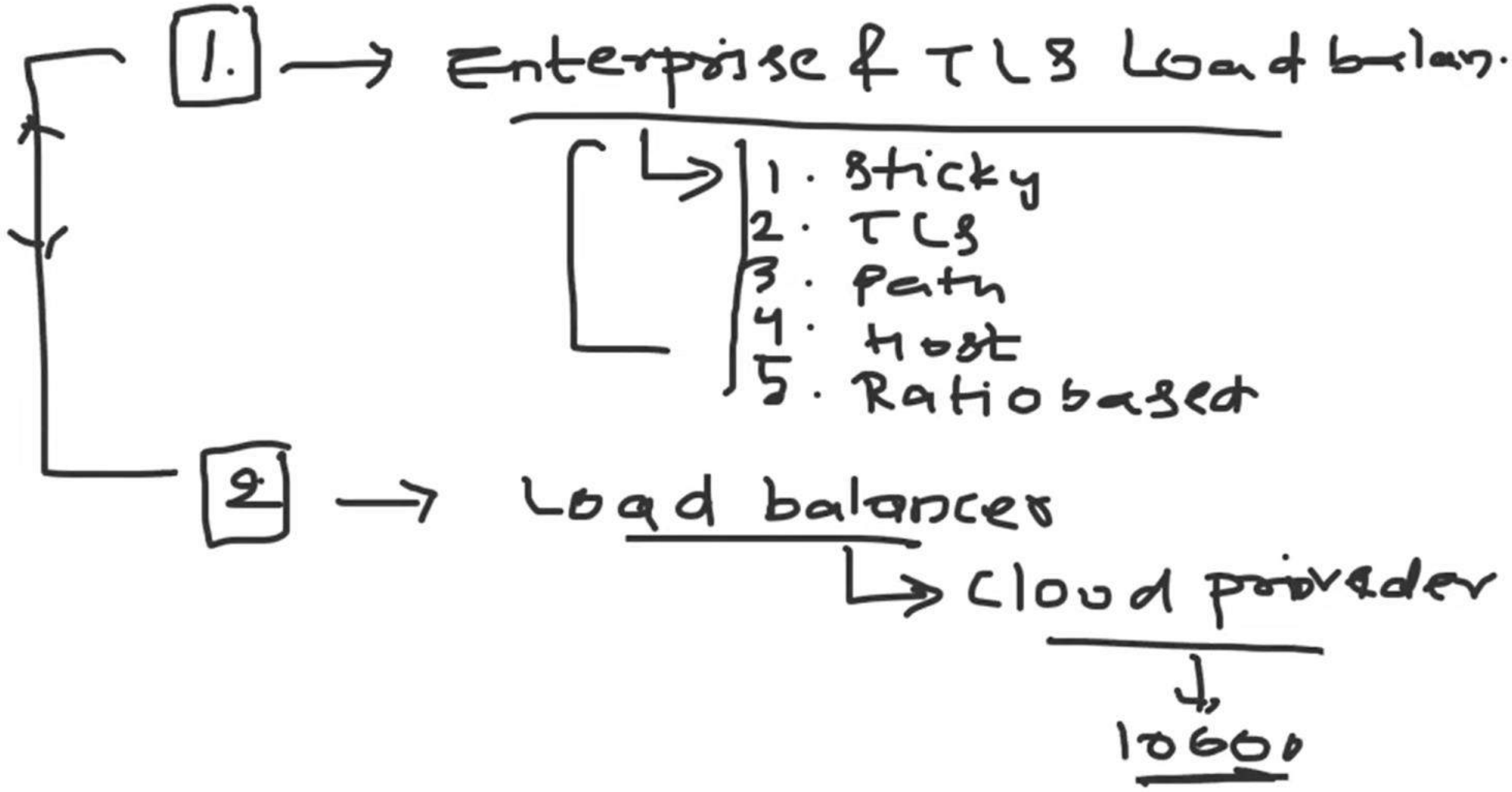
Where as if you create a Service as type LoadBalancer, the cloud control manager creates a external load balancer IP using the underlying cloud provider logic in the C-CM. Users can access services using the external IP

# Question 9

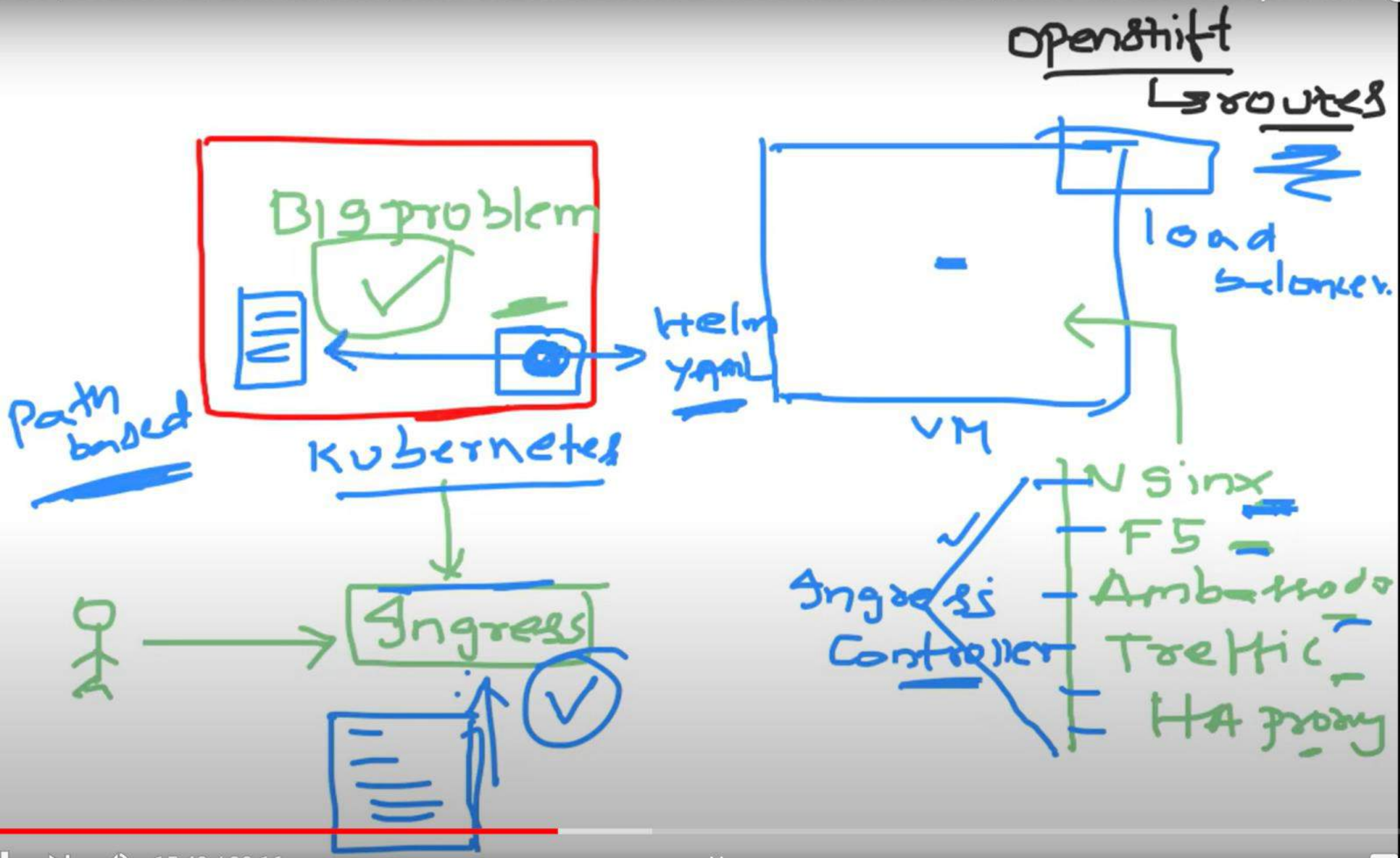
## What is the role of Kubelet ?

Kubelet manages the containers that are scheduled to run on that node. It ensures that the containers are running and healthy, and that the resources they need are available.

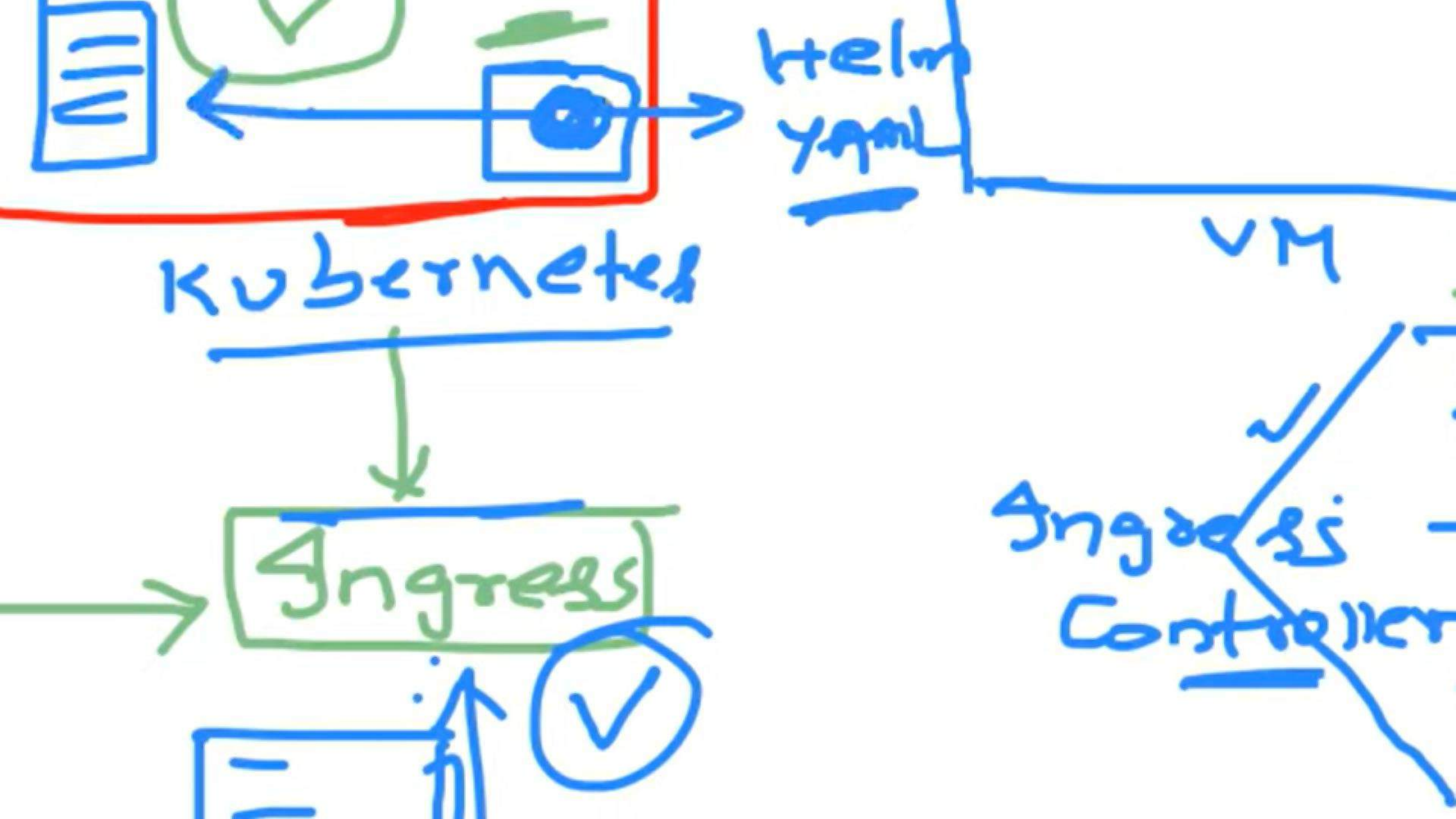
Kubelet communicates with the Kubernetes API server to get information about the containers that should be running on the node, and then starts and stops the containers as needed to maintain the desired state. It also monitors the containers to ensure that they are running correctly, and restarts them if necessary.


1.
Enterprise & TLS Load balan.
1. Sticky
2. TLS
3. Path
4. Host
5. Ratio based
2.
Load balancer
Cloud provider
10600

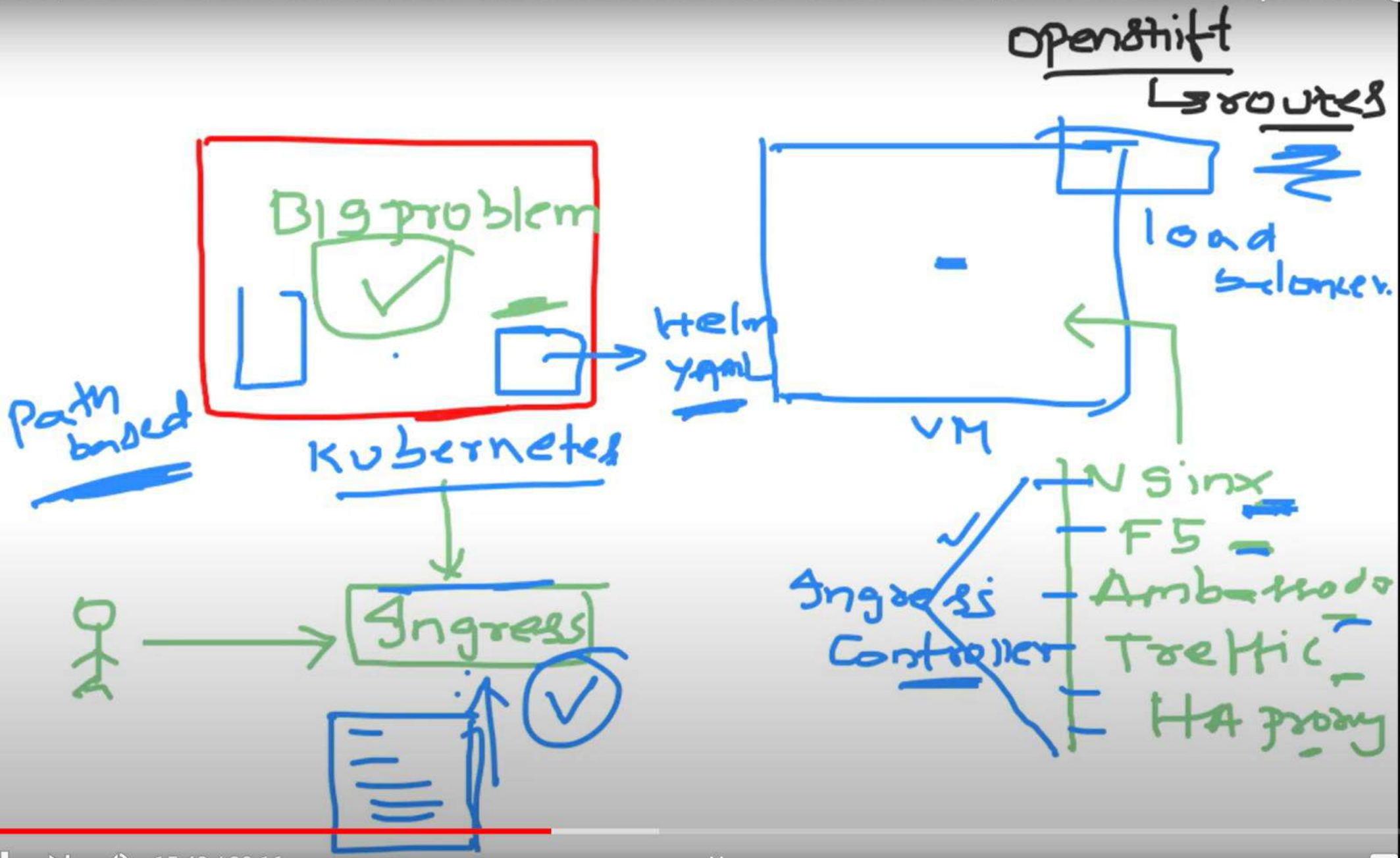
DAY-38 | KUBERNETES INGRESS | WATCH DEMO USING LINK IN DESCRIPTION | #kubernetes #devops #fr...



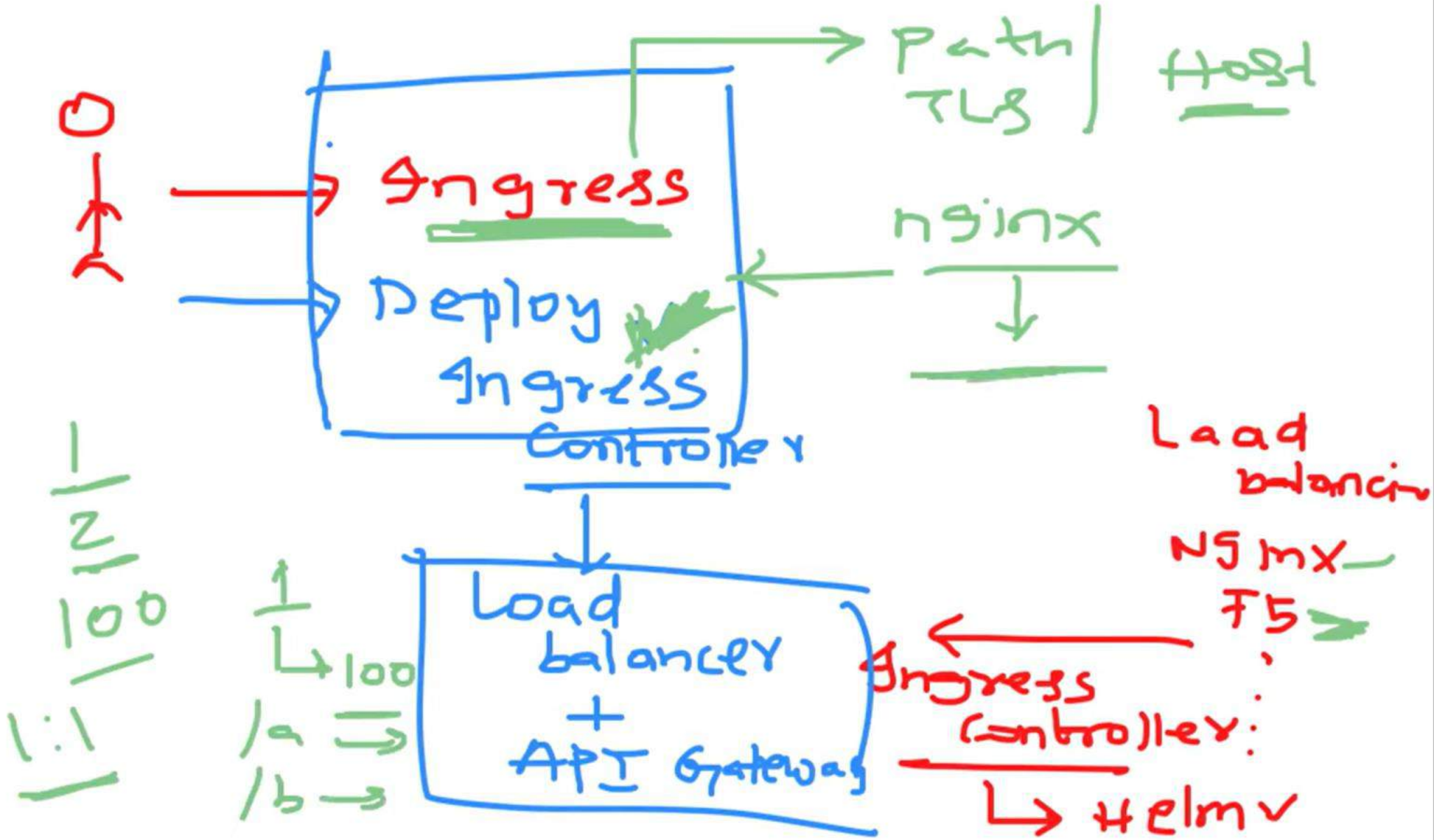

15:49 / 39:16


helm
yaml
VM
Kubernetes
Ingress
Ingress
Controller

DAY-38 | KUBERNETES INGRESS | WATCH DEMO USING LINK IN DESCRIPTION | #kubernetes #devops #fr...




15:40 / 39:16


Path
TLS
Host
Ingress
Deploy
Ingress
Controller
nginx
1
2
100
1:1
1
100
/a
/b
Load
balancer
+
API Gateway
Laad
balancin
NGINX
F5
Ingress
Controller:
helmv

1. Enterprise

↳ Security

load balancing

2. Service → IP


nginx

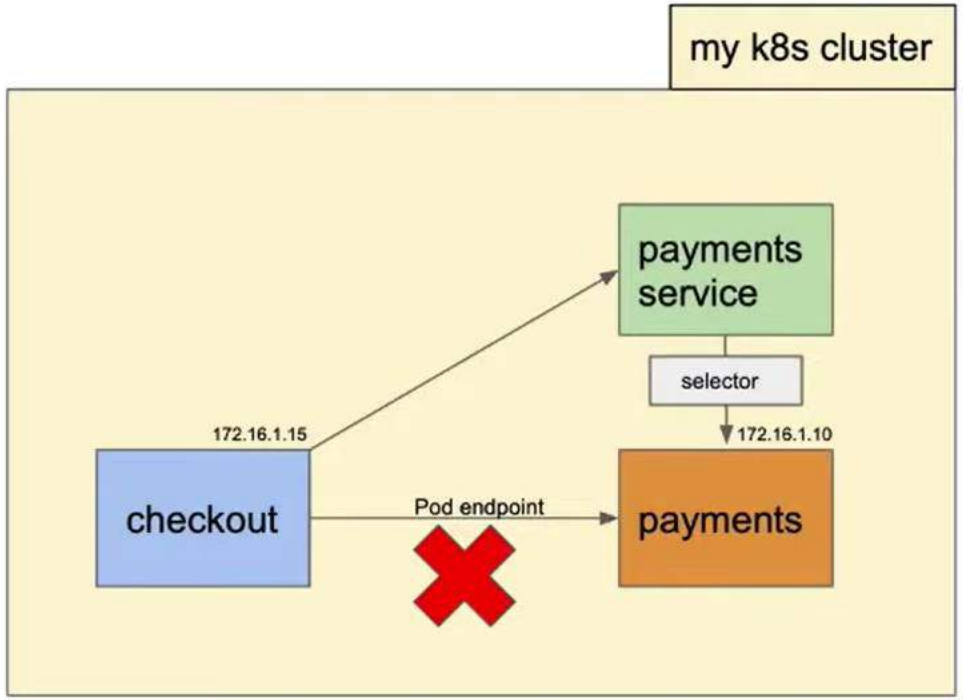
F5

Ingress

Controller

# Kubernetes Service

An abstract way to expose an application running on a set of Pods as a network service.




Cluster IP

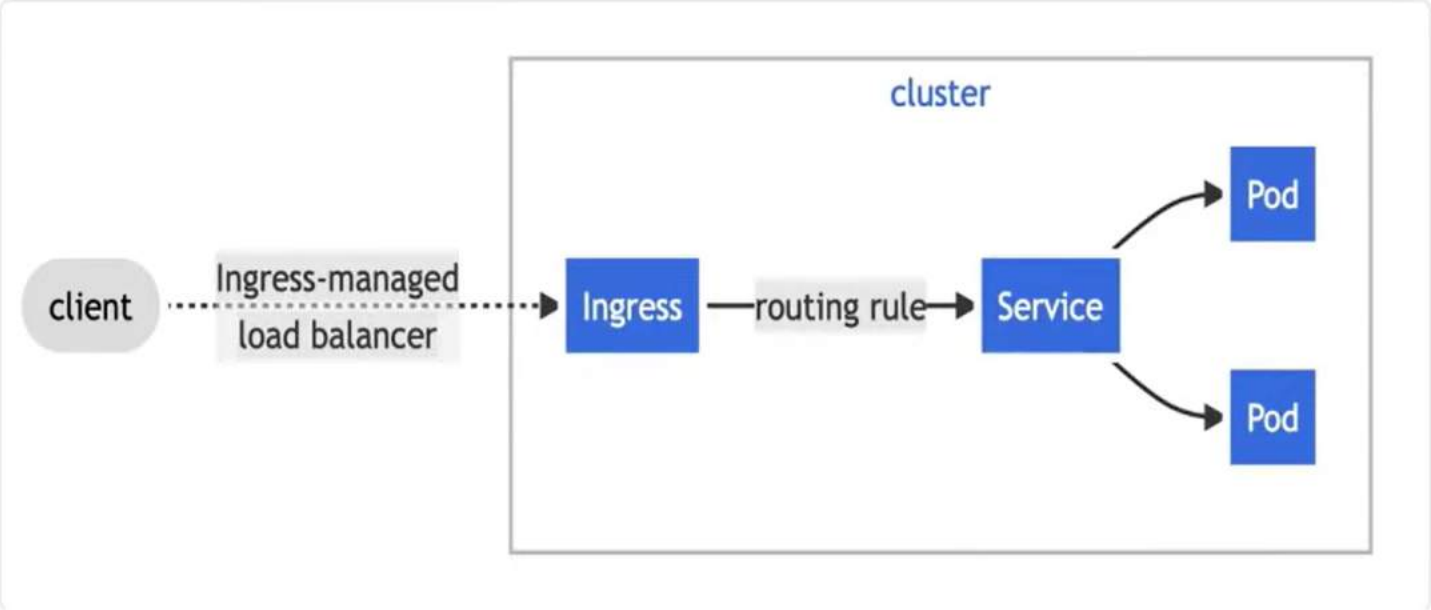
NodePort

Load Balancer

Abhishek Veeramalla

# What is a Kubernetes Ingress ?

Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.




source: k8s docs

A
Abhishek Veeramalla

# Sample Ingress

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-ingress
spec:
  defaultBackend:
    service:
      name: test
      port:
        number: 80
```

Reference:
https://raw.githubusercontent.com/kubernetes/website/main/content/en/examples/service/networking/test-ingress.yaml


Abhishek Veeramalla

## Host Based Routing

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-with-auth
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: http-svc
            port:
              number: 80
  - host: example.bar.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: meow-svc
            port:
              number: 80
```

## Path Based Routing

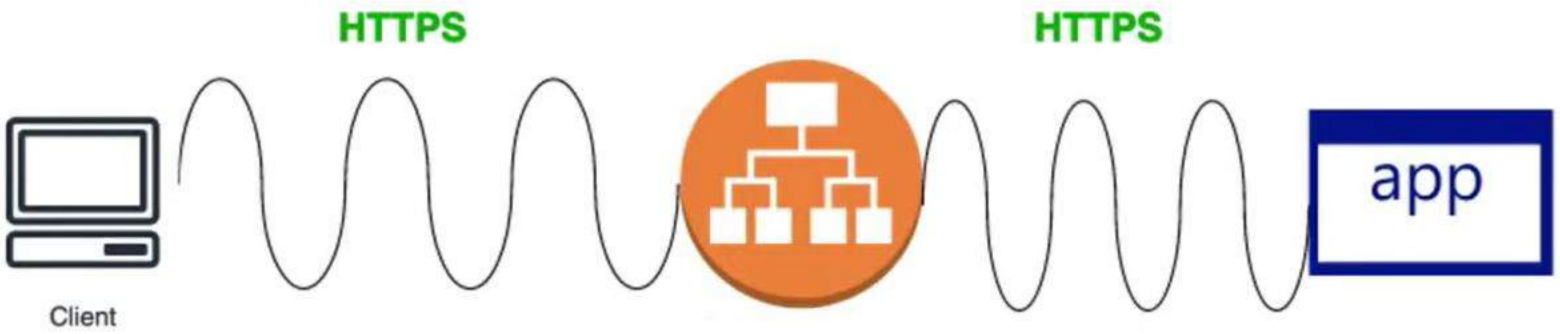
```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-with-auth
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /first
        pathType: Prefix
        backend:
          service:
            name: http-svc
            port:
              number: 80
      paths:
      - path: /second
        pathType: Prefix
        backend:
          service:
            name: meow-svc
            port:
              number: 80
```


Abhishek Veeramalla

# TLS

## SSL Passthrough

SSL passthrough passes encrypted HTTPS traffic directly to the backend servers without decrypting the traffics at the load balancer.




Load Balancer capabilities are merely used.

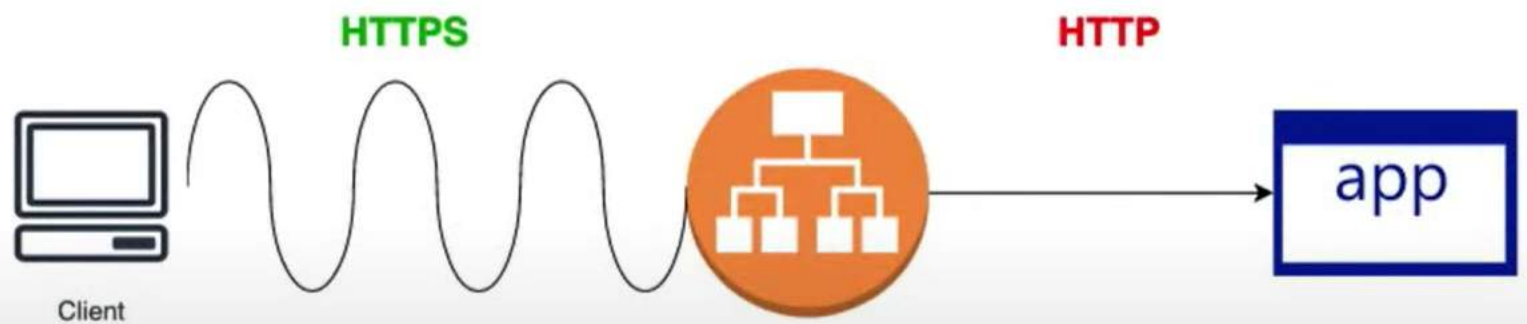
Attacker can pass hacking codes in the traffic and will be directly passed to the backend server.

SSL Passthrough is also a costly process. Might require more CPU.

# TLS

## SSL Offloading

SSL Offloading (SSL Termination) decrypts all HTTPS traffics when it arrives at the load balancer and the data is sent to the destination server as plain HTTP traffic.




Vulnerable to data theft, man-in-the-middle attacks.

Faster

A
Abhishek Veeramalla

# TLS

## SSL Bridging

SSL Bridging decrypts all **HTTPS** traffics when it arrives at the load balancer and the data is sent to the destination server as **HTTPS** traffic by **re-encrypting**.


E2E encrypted and validated for malware attacks by Load Balancer.

More secure more costly in processing as server has to decrypt the traffic.

# Pros and Cons !!

| SSL Passthrough | SSL-Offloading/Termination | SSL-Bridge/Re-encrypt |
|---|---|---|
| Costly in processing for Server | Fast in processing | Costly in processing for Server |
| Secure in many cases | Insecure | Secure |
| L4 (TCP) Load Balancing | L7 Load Balancing | L7 Balancing |
| Choose when you don't bother about access rules, blocking, cookie e.t.c.,. | When you want less latency and can compromise on security. | When you need security and advanced load balancer capabilities. |
| No Load Balancer Inspection. | Load Balancer Inspects the packets. | Load Balancer Inspects the packets. |
| Recommended* | Highly unrecommended. | Recommended |


Abhishek Veeramalla

# Secure Routes

*Routes does not support storing the TLS certs in secrets - Issue

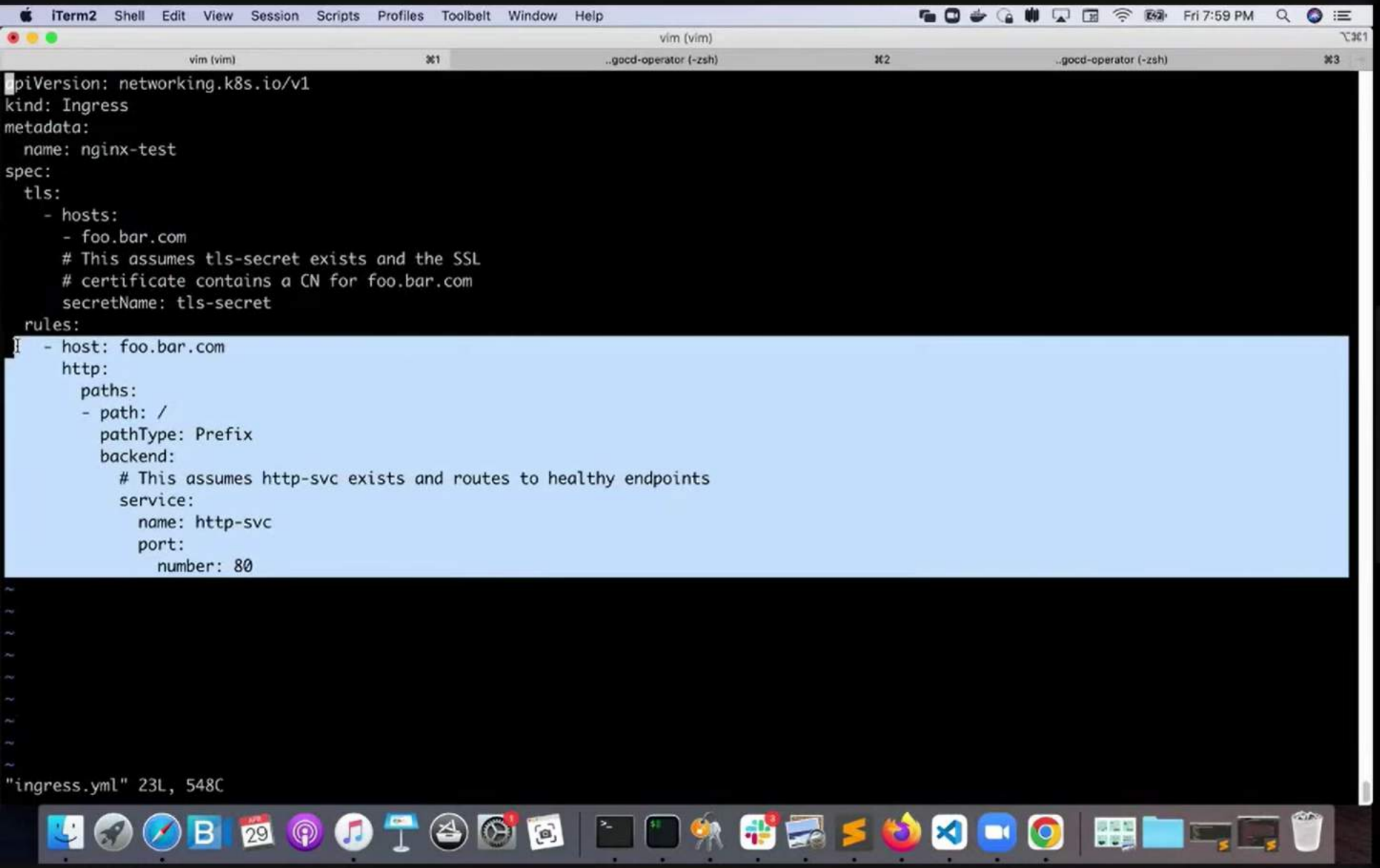
Routes are simple, you cannot add multiple services, paths or hosts in a single route.

```yaml
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: route-passthrough-secured (1)
spec:
  host: www.example.com
  port:
    targetPort: 8080
  tls:
    termination: passthrough (2)
    insecureEdgeTerminationPolicy: None
  to:
    kind: Service
    name: frontend
```

```yaml
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: frontend
spec:
  host: www.example.com
  to:
    kind: Service
    name: frontend
  tls:
    termination: edge
    key: |-
      -----BEGIN PRIVATE KEY-----
      [...]
      -----END PRIVATE KEY-----
    certificate: |-
      -----BEGIN CERTIFICATE-----
      [...]
      -----END CERTIFICATE-----
    caCertificate: |-
      -----BEGIN CERTIFICATE-----
      [...]
      -----END CERTIFICATE-----
```

```yaml
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: frontend
spec:
  host: www.example.com
  to:
    kind: Service
    name: frontend
  tls:
    termination: reencrypt
    key: |-
      -----BEGIN PRIVATE KEY-----
      [...]
      -----END PRIVATE KEY-----
    certificate: |-
      -----BEGIN CERTIFICATE-----
      [...]
      -----END CERTIFICATE-----
    caCertificate: |-
      -----BEGIN CERTIFICATE-----
      [...]
      -----END CERTIFICATE-----
    destinationCACertificate: |-
      -----BEGIN CERTIFICATE-----
      [...]
      -----END CERTIFICATE-----
```


Abhishek Veeramalla



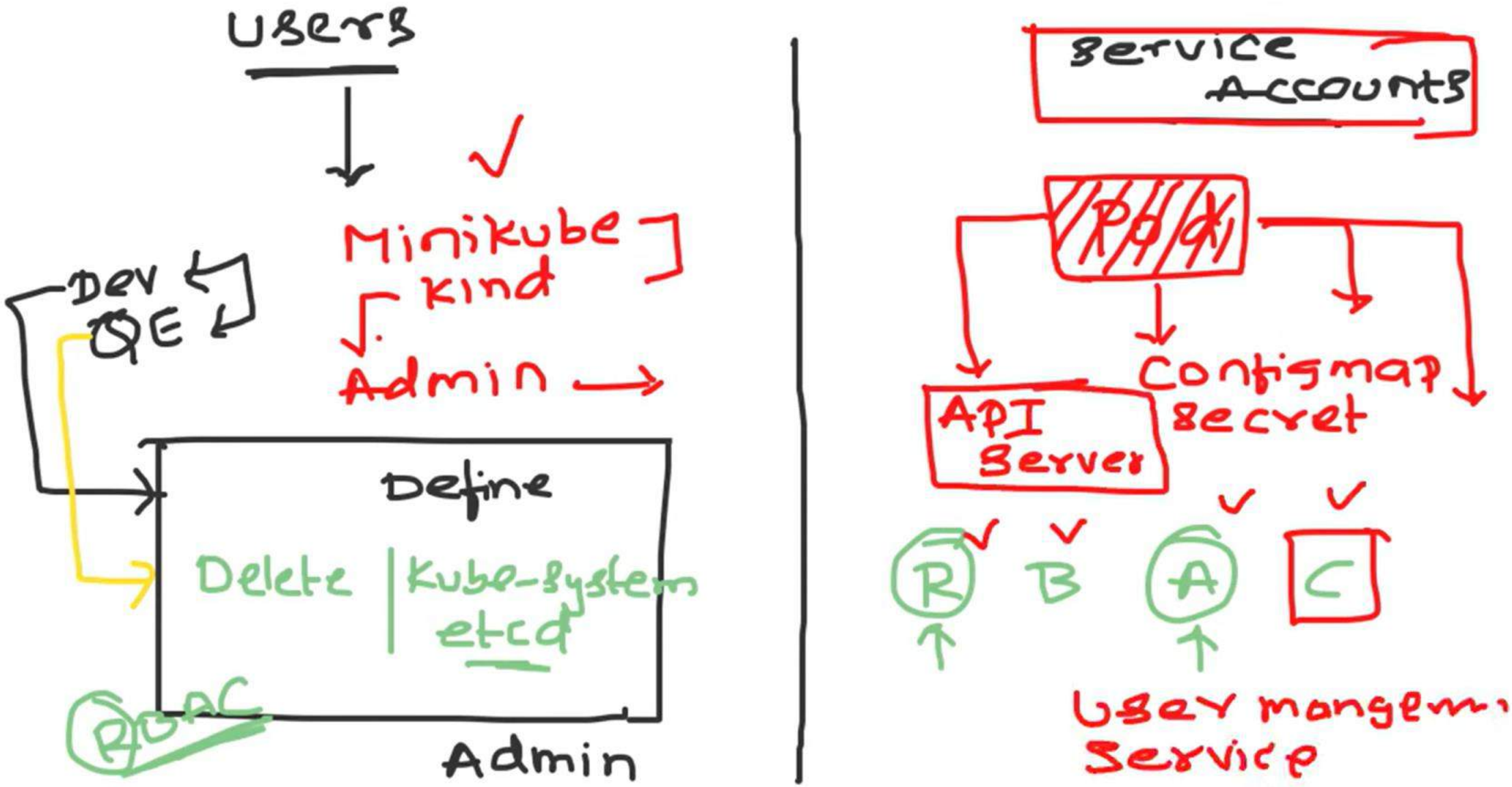


Abhishek Veeramalla

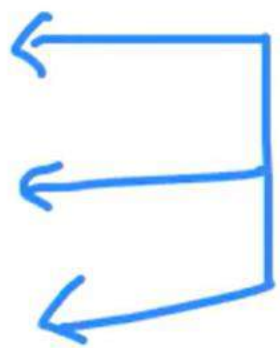
youtube.com - To exit full screen, press Esc

# RBAC

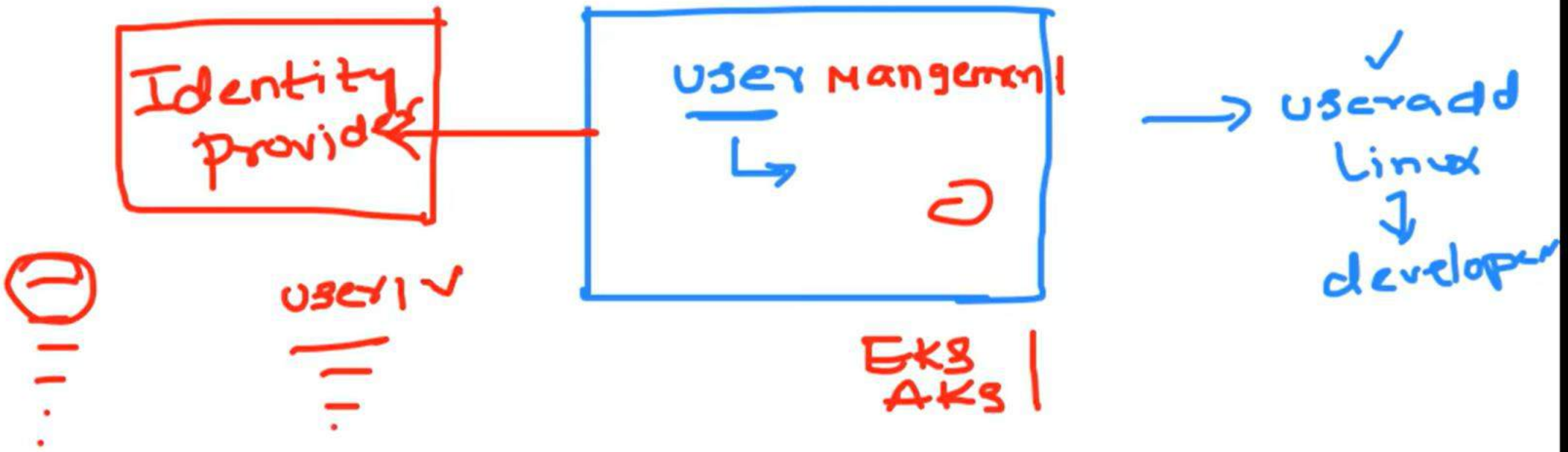
Day

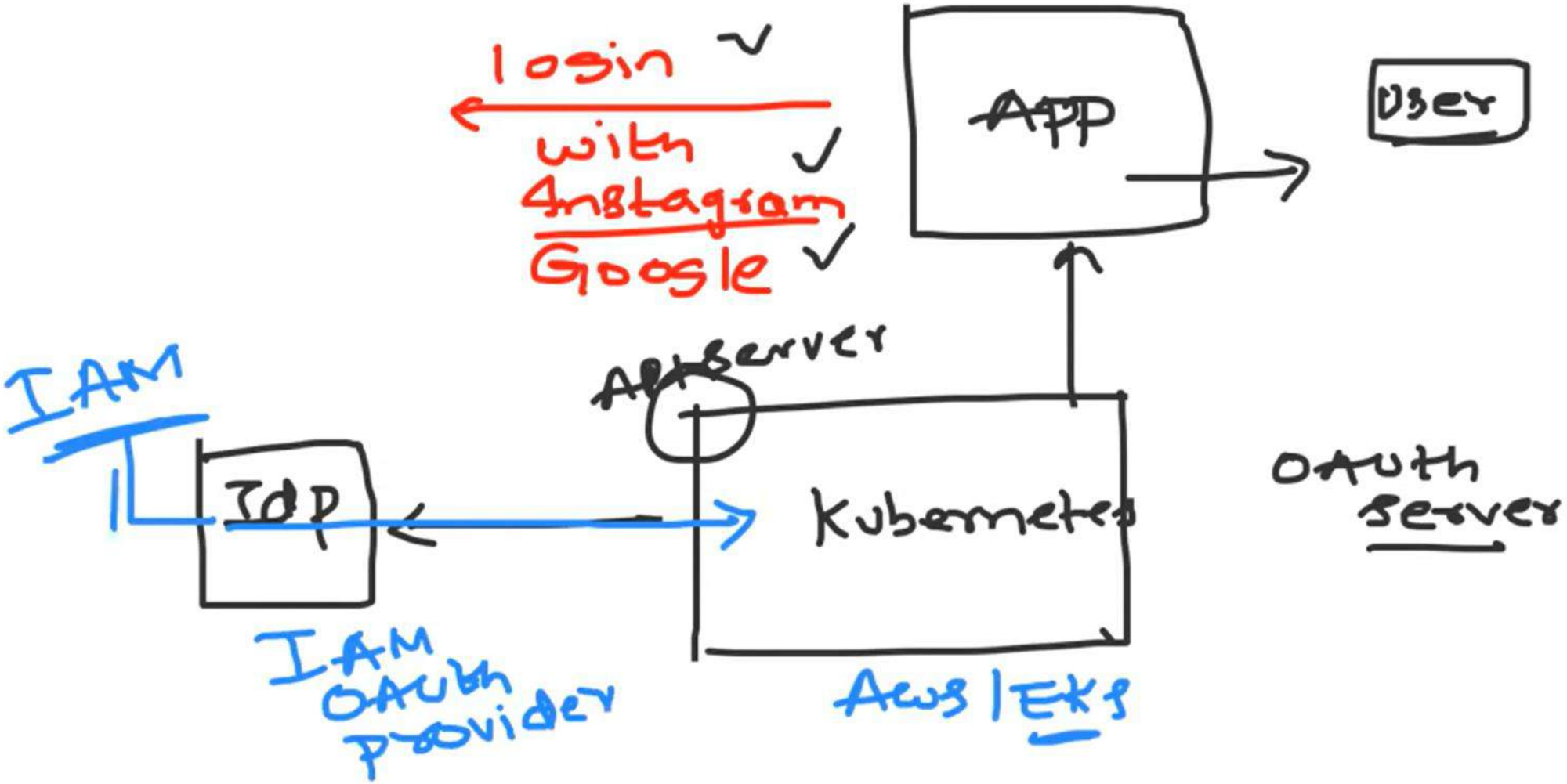

Users
Minikube
kind
Dev
QE
Admin
Define
Delete
Kube-system
etcd
RBAC
Admin
Service
Accounts
RBAC
API
Server
Configmap
secret
R
B
A
C
User mangem
Service

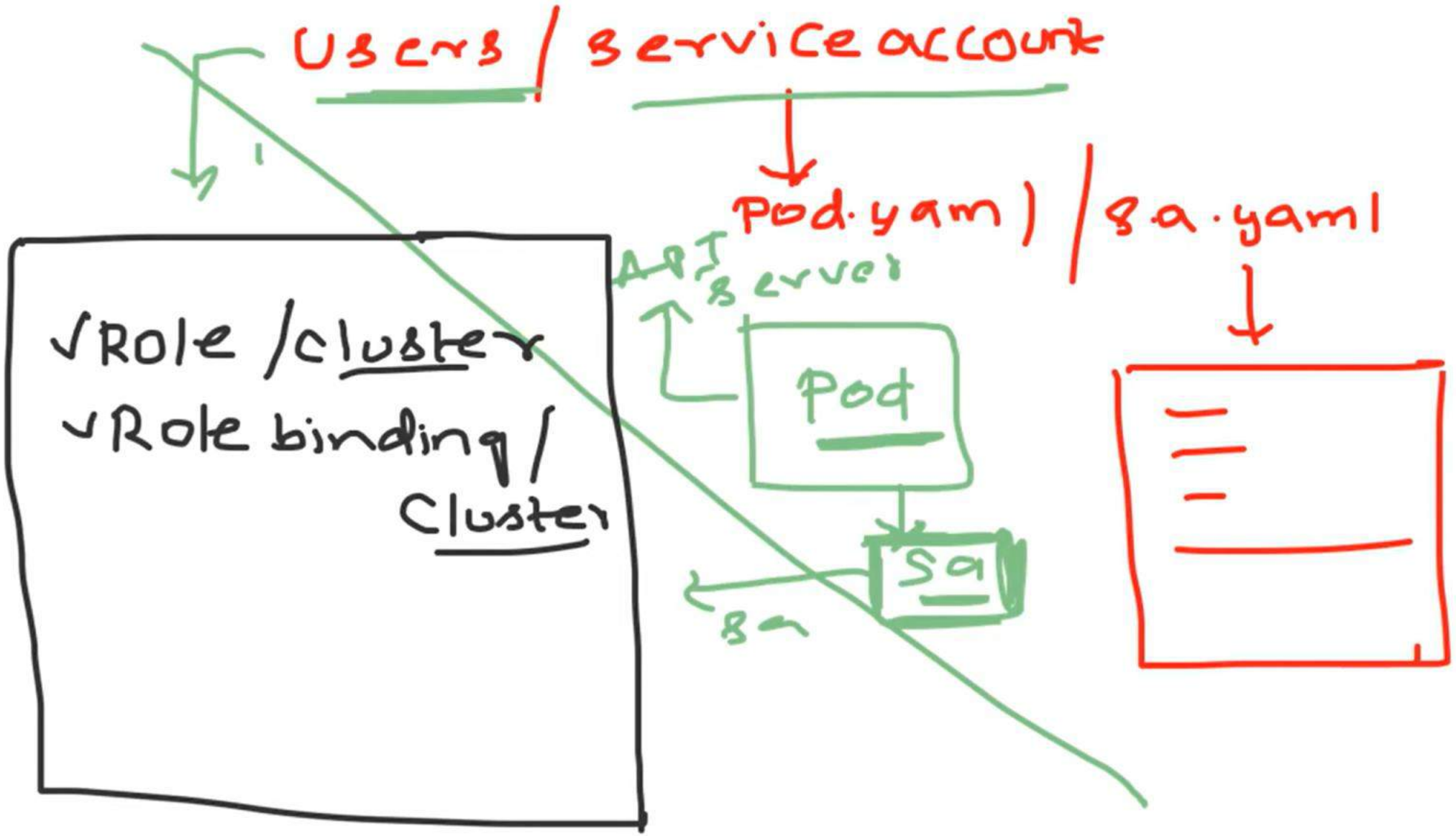
1. Service Account / Users
2. Roles / Cluster Role
3. Role binding / CRB

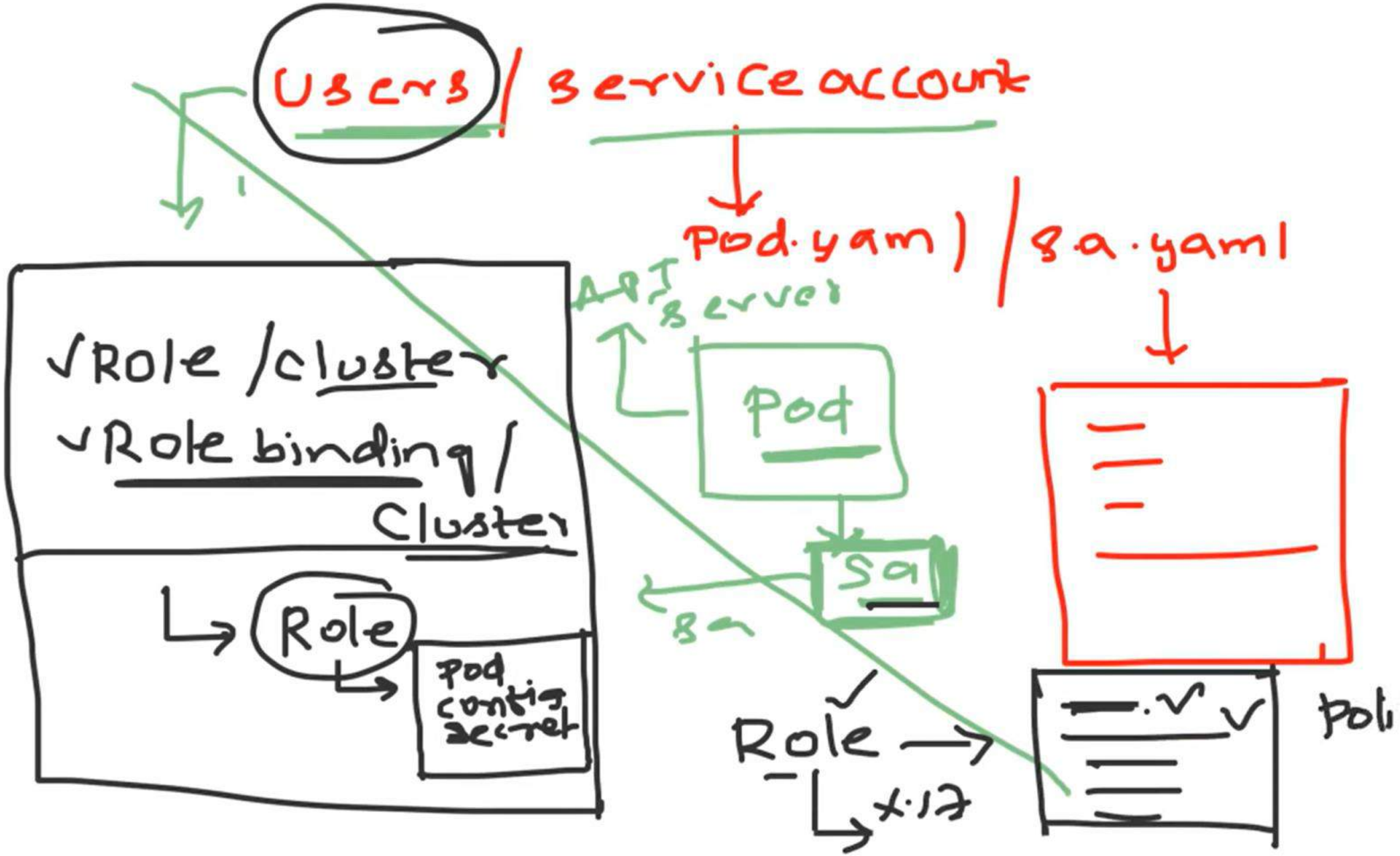


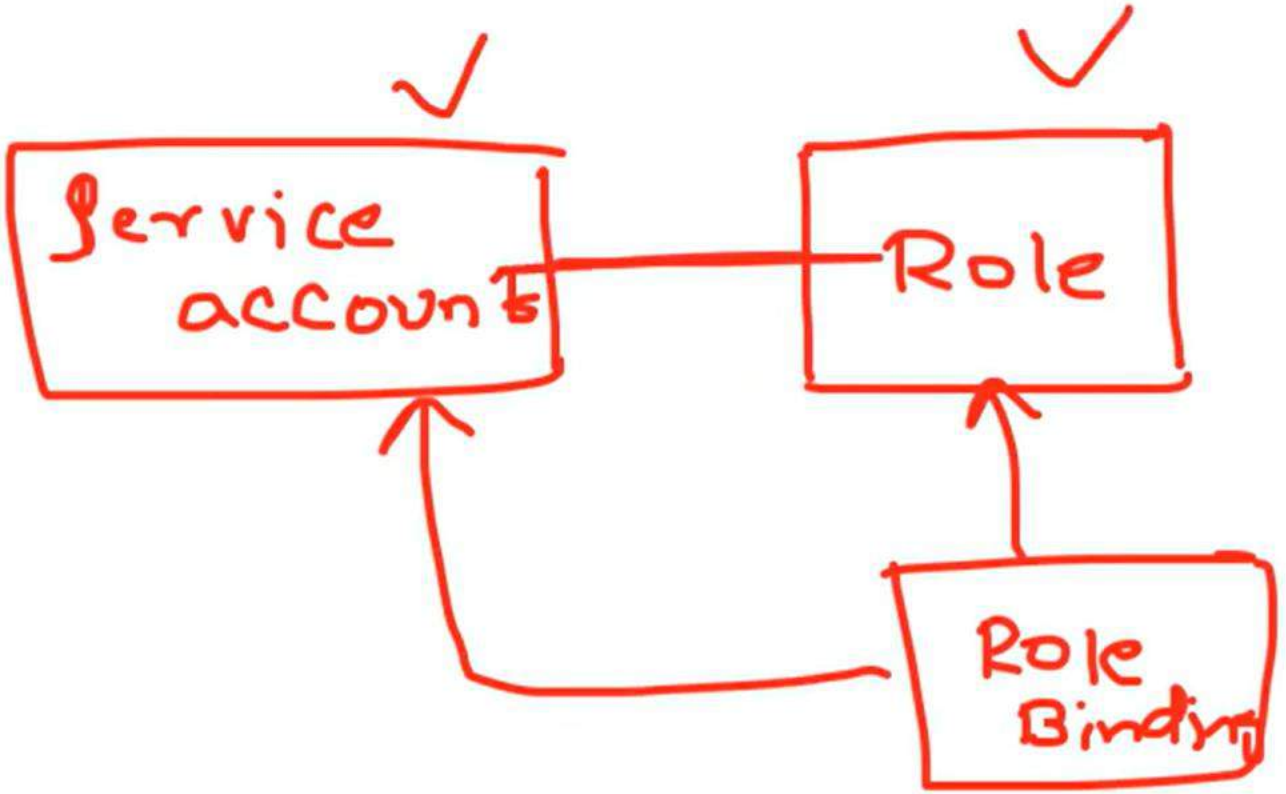
1. Service Account / Users
2. Roles / Cluster Role
3. Role binding / CRB

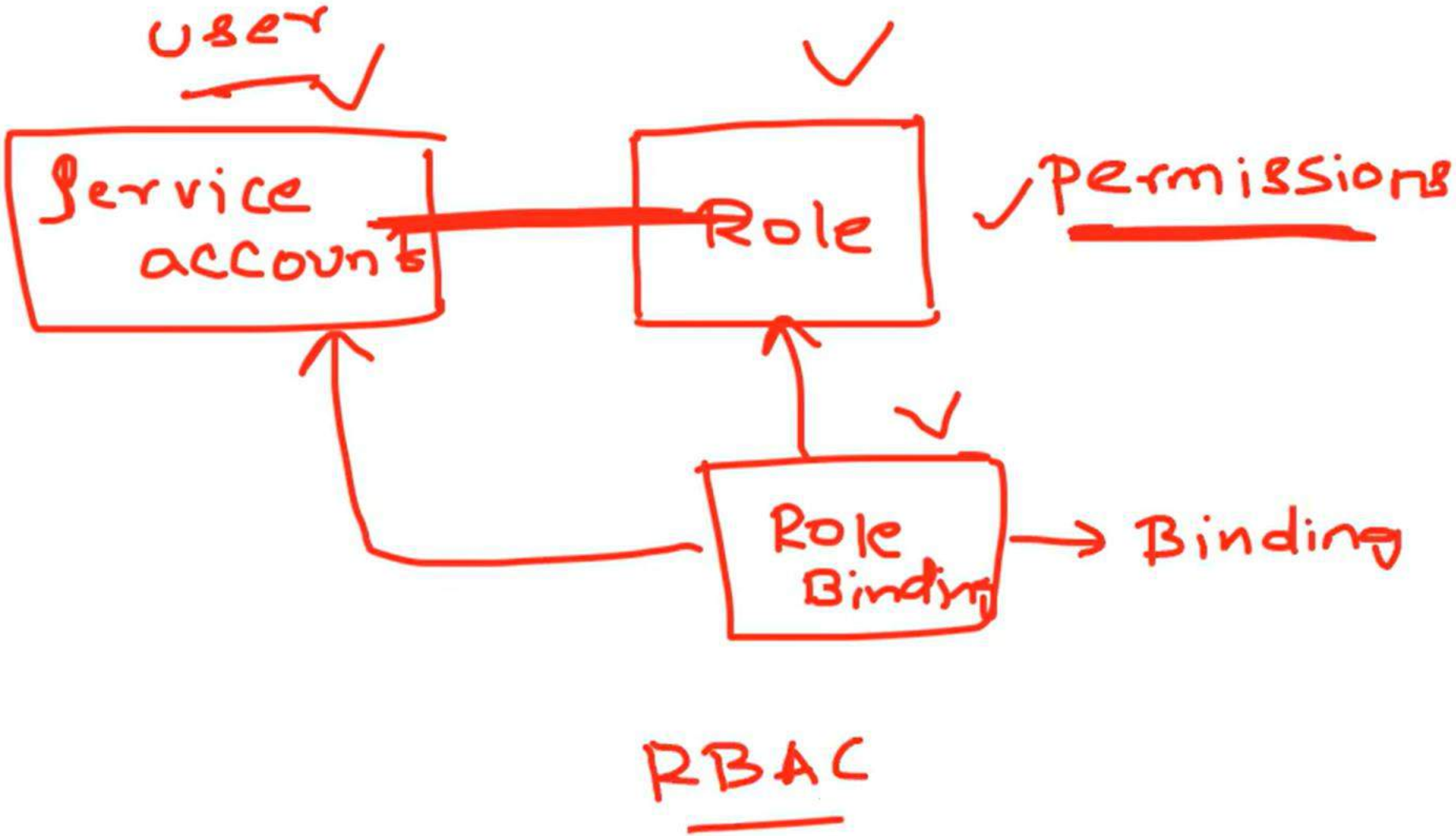



login
with
Instagram
Google
App
User
IAM
IdP
Kubernetes
OAuth
Server
IAM
OAuth
provider
AWS | EKS


Users / Service account
Pod.yaml / sa.yaml
API Server
Pod
sa
sa
Role /cluster
Role binding /
Cluster


Users / Service account
Pod.yaml / sa.yaml
API server
Pod
sa
Role /cluster
Role binding /
Cluster
Role
Pod config secret
Role
x.17
Poli






RBAC

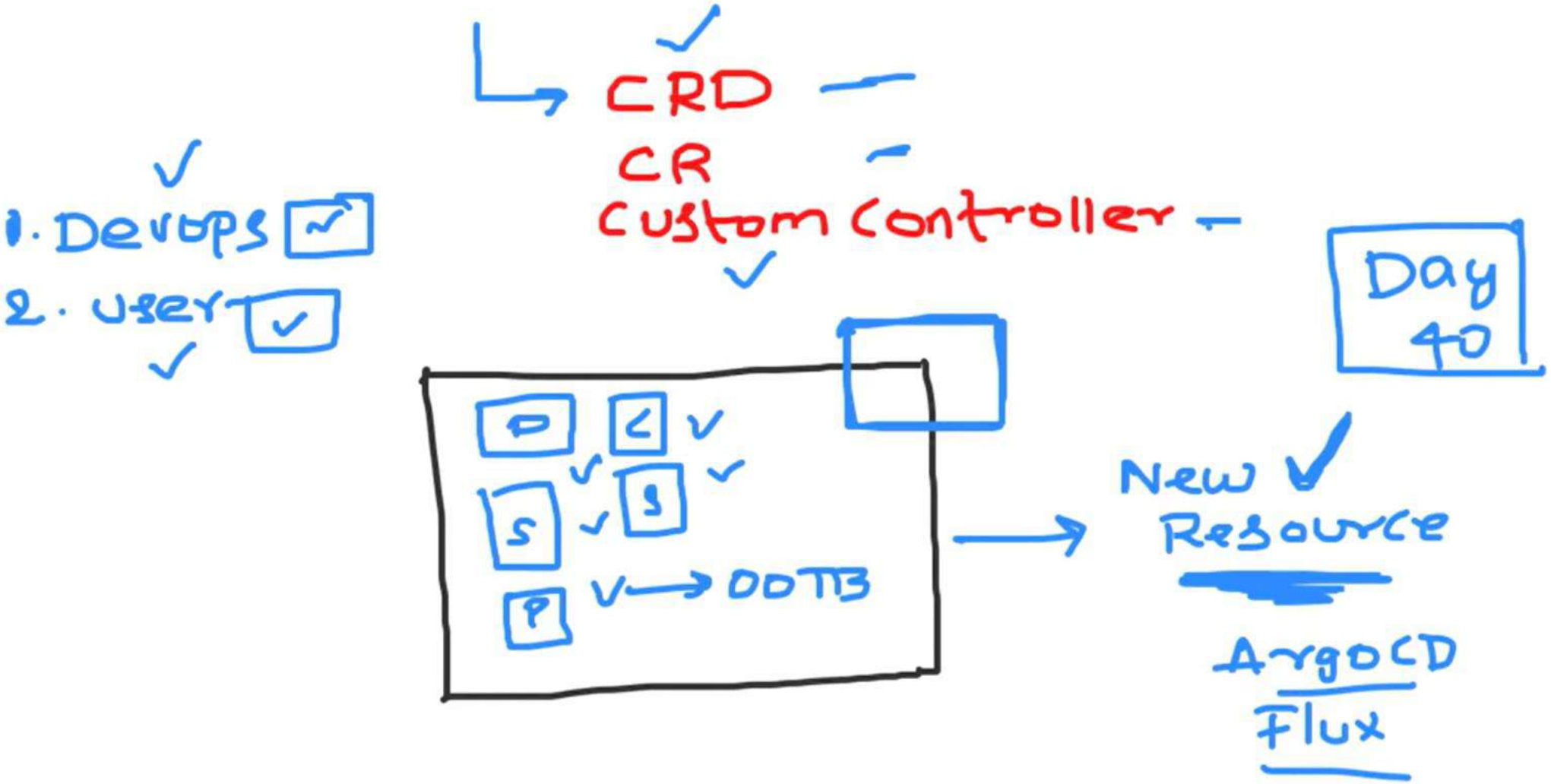
youtube.com - To exit full screen, press Esc

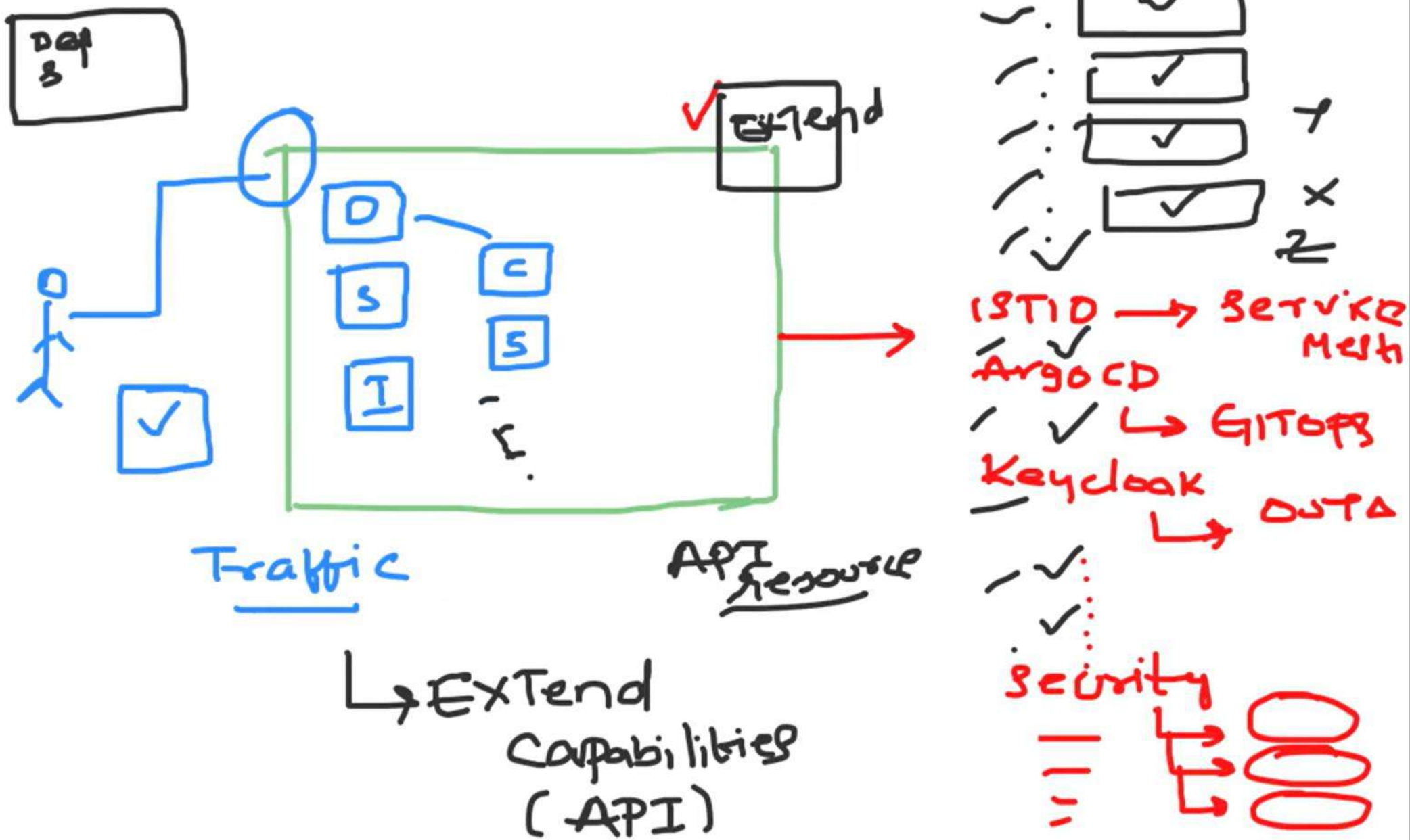
# Custom Resource

Day 40

youtube.com - To exit full screen, press Esc

# Custom Resource

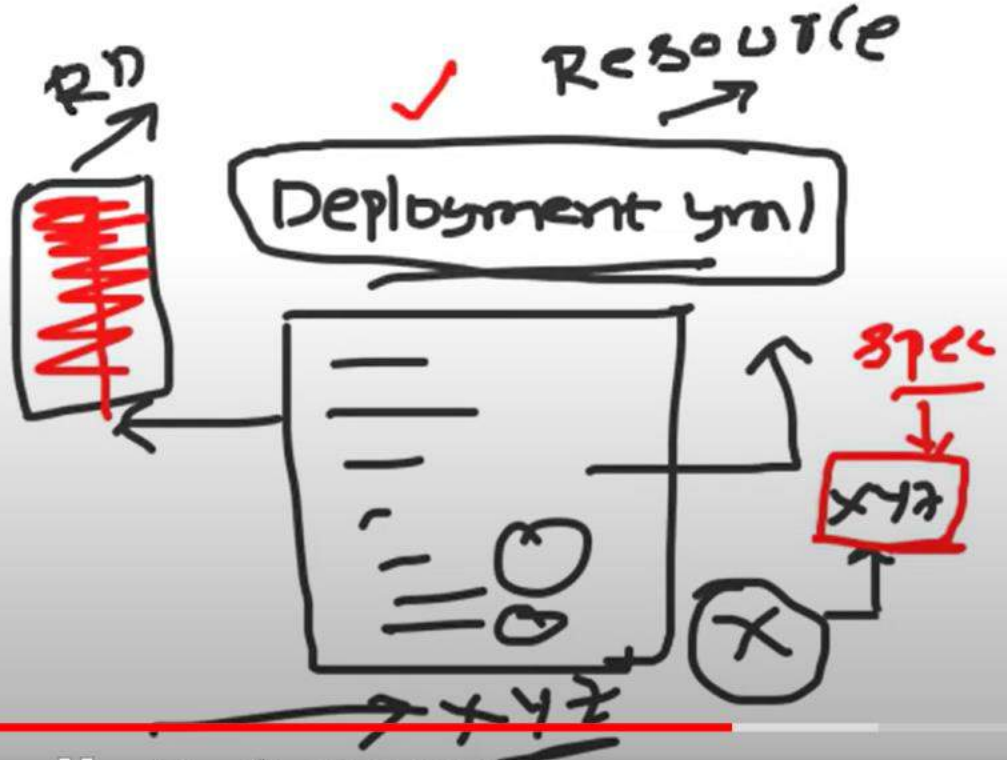



DB
S
Extend
D
S
I
C
S
Traffic
API Resource
Istio → Service Mesh
ArgoCD → GitOps
Keycloak → OIDC
Security
↳ Extend
Capabilities
(API)

1. CRD
2. CR
3. Custom Controller

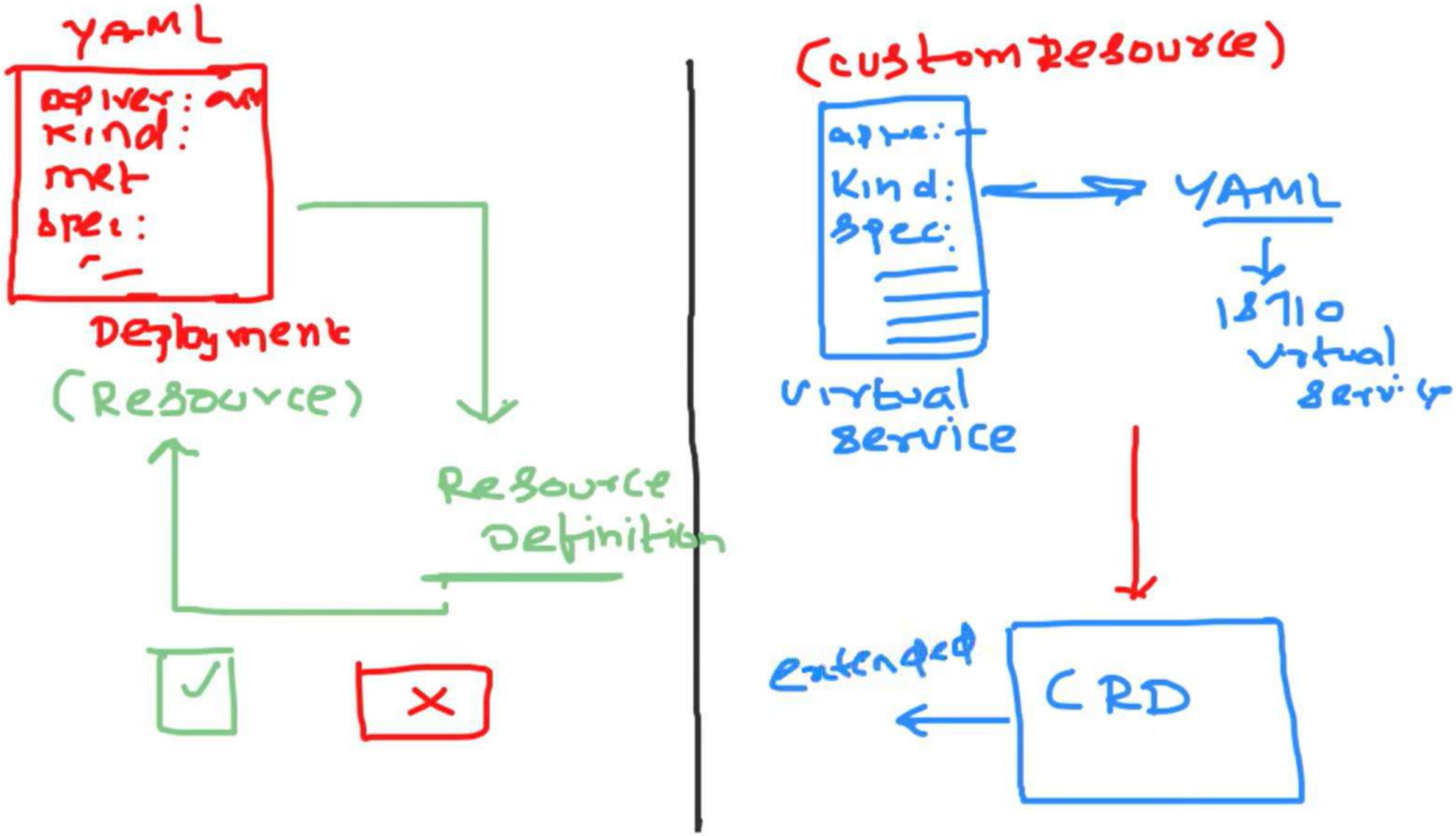
Defining a new type of API to k8s

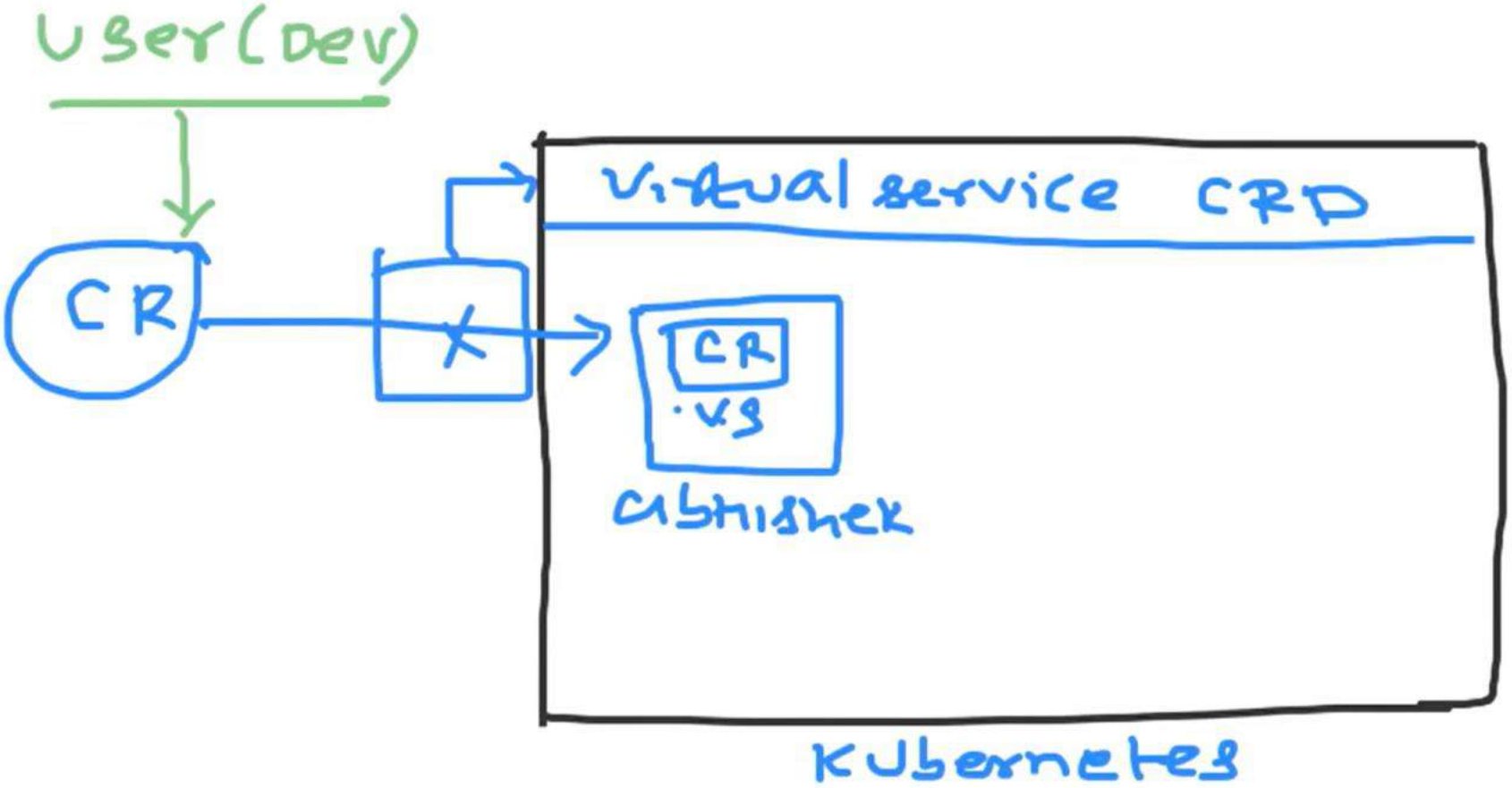
CRD



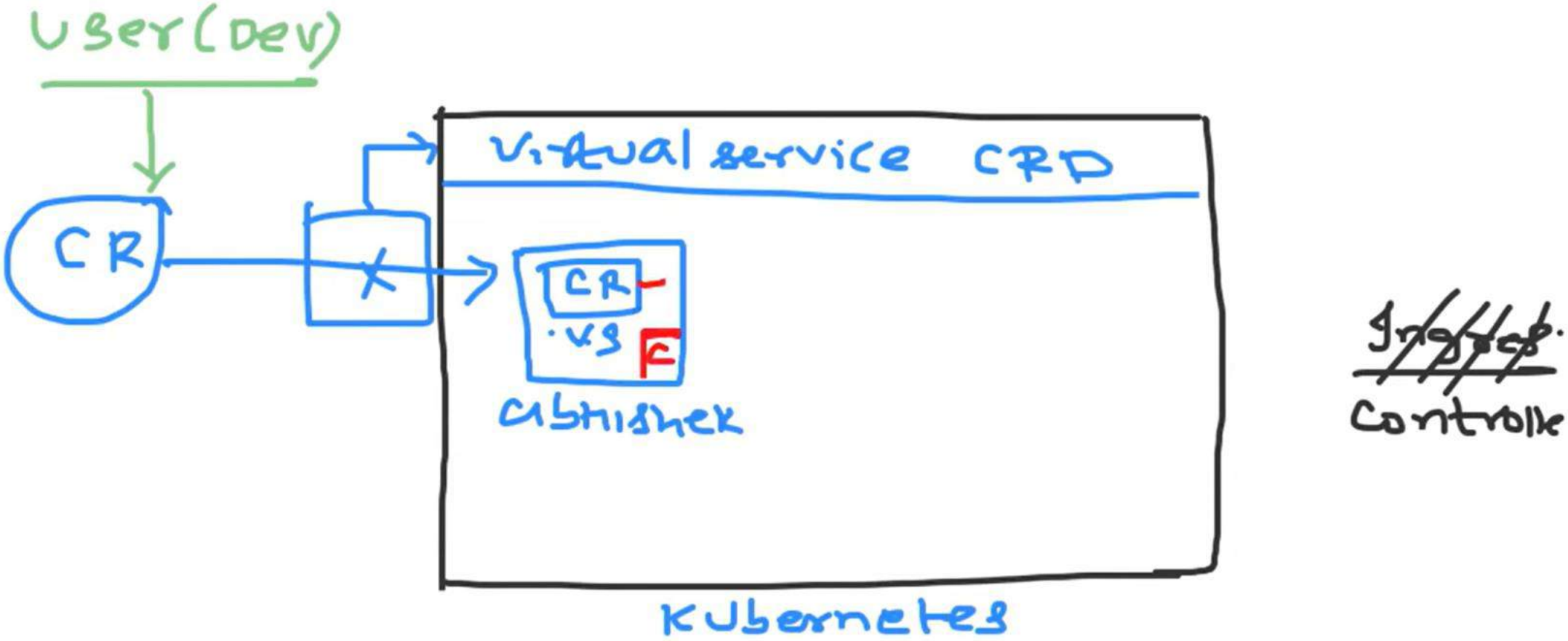

YAML

Options


YAML
apiVer:
kind:
met
spec:
Deployment
(Resource)
Resource definition
(CustomResource)
apiVe:
Kind:
Spec:
YAML
Istio virtual servi
Virtual service
extended
CRD




DevOps ① CRD → DOC → Manifests/Helm/Operator

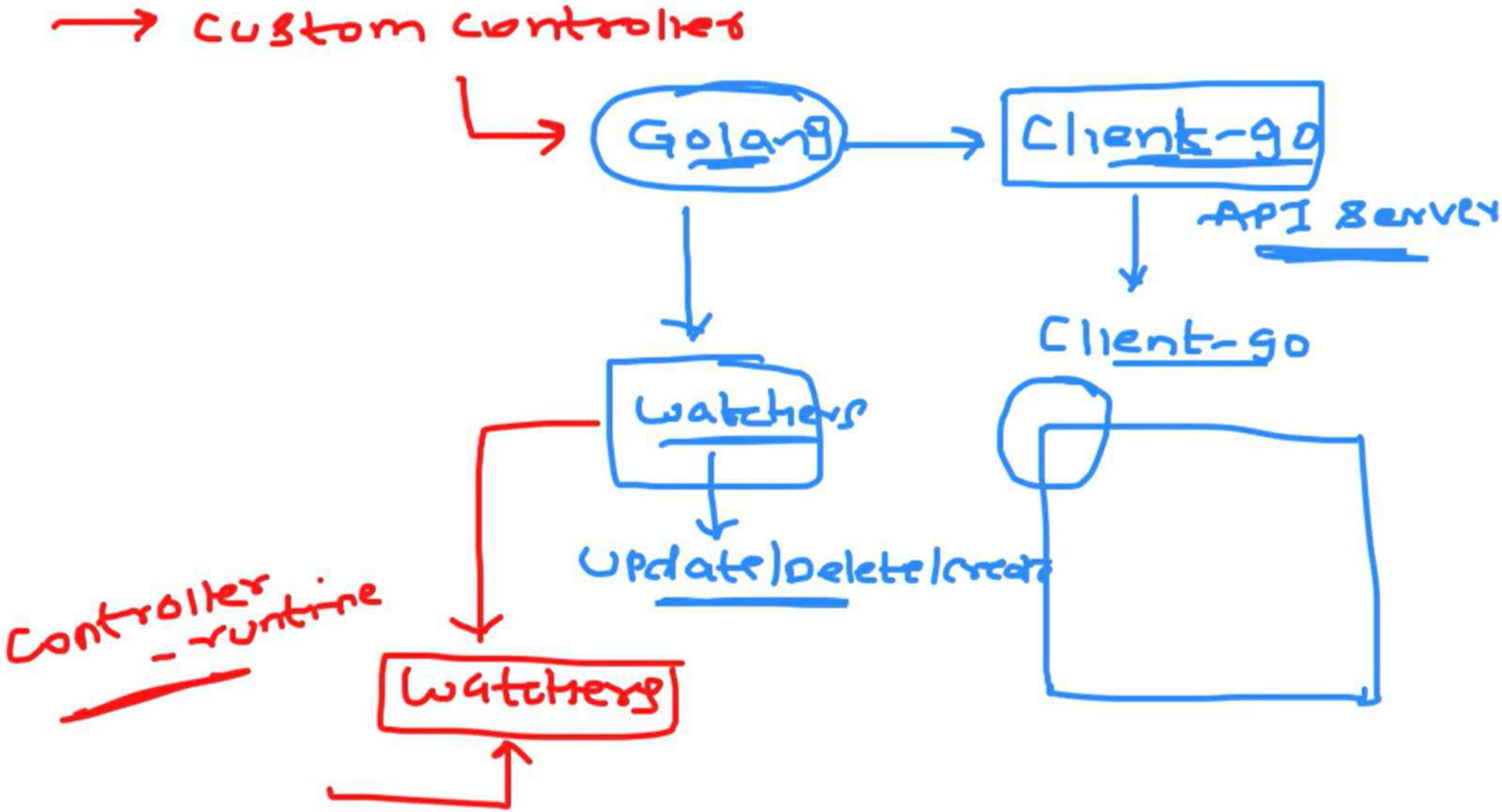


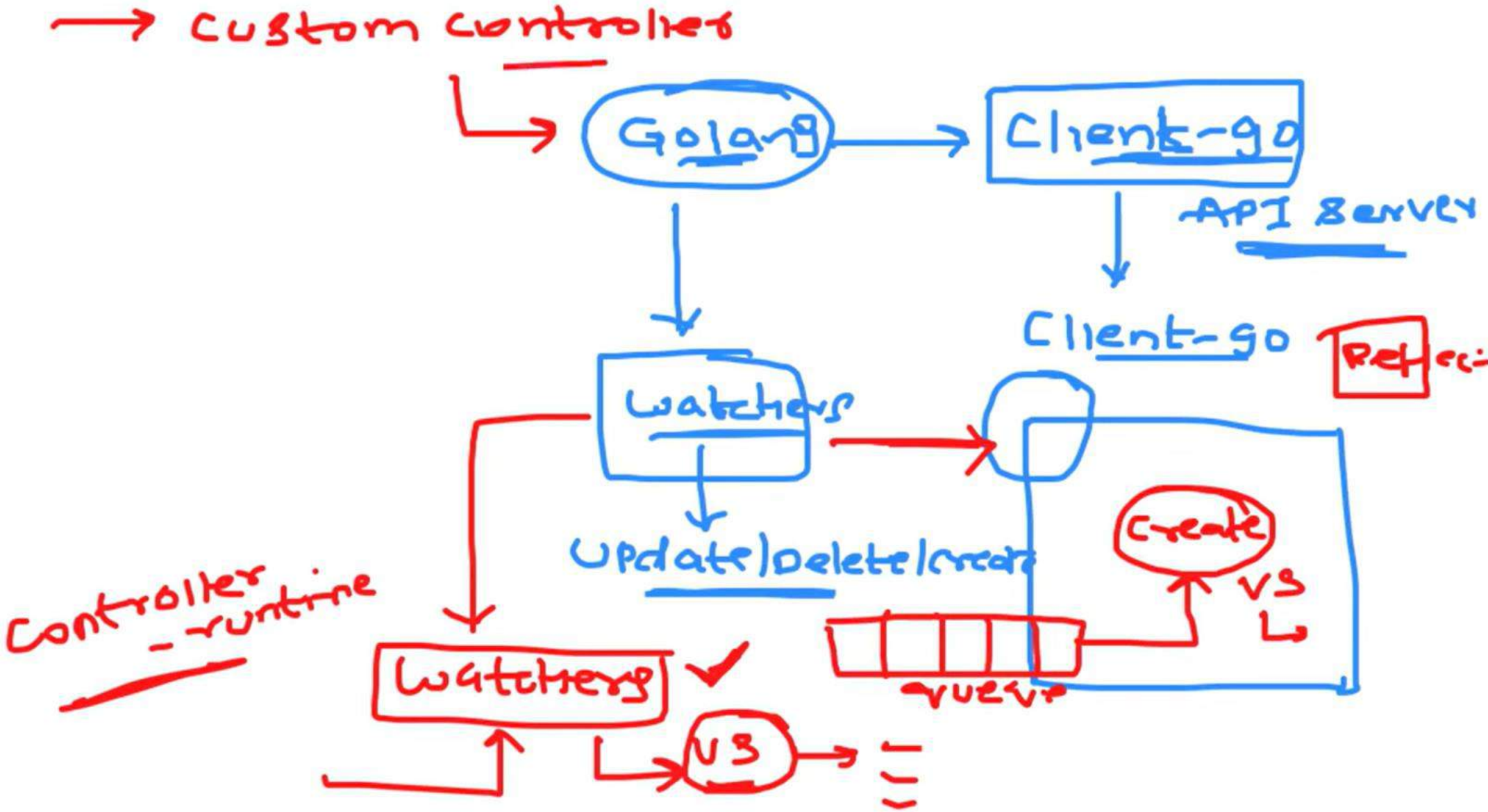

Controller

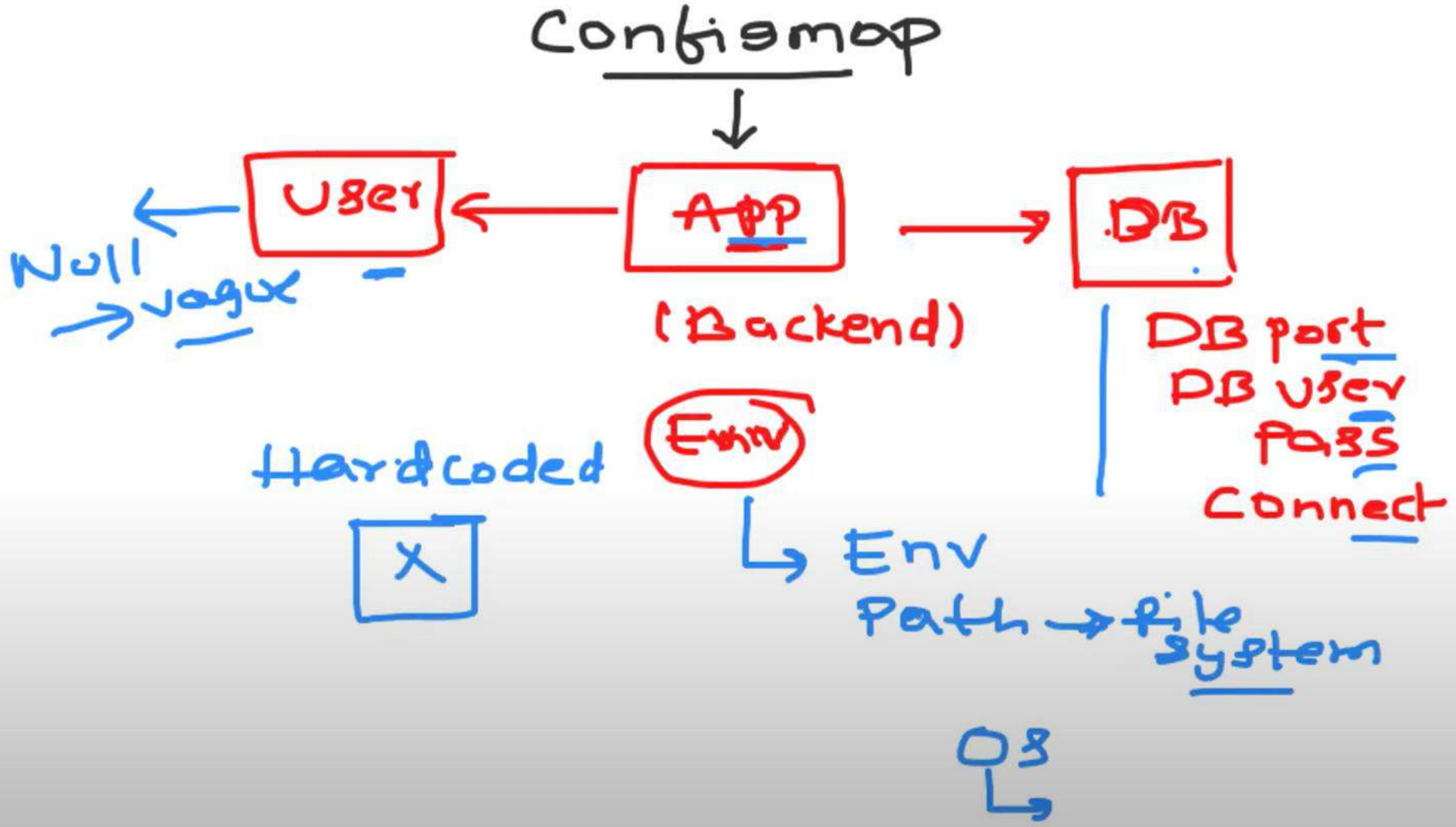
DevOps

1. CRD → Doc → Manifests/Helm/Operator

2. Custom Controller → Helm/Operator



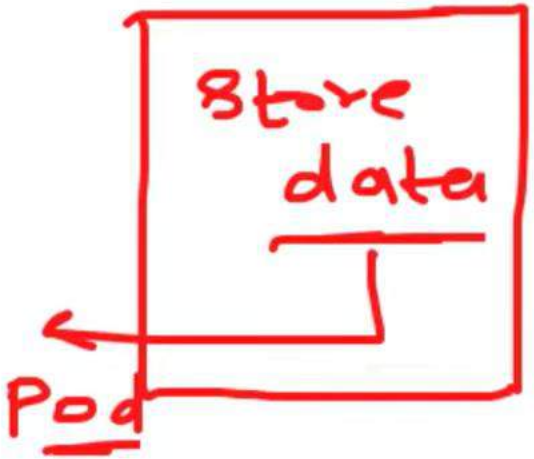

Custom controllers
Golang
Client-go
API Server
Client-go
Watchers
Update/Delete/Create
Create
VS
Queue
Controller
-runtime
Watchers
VS



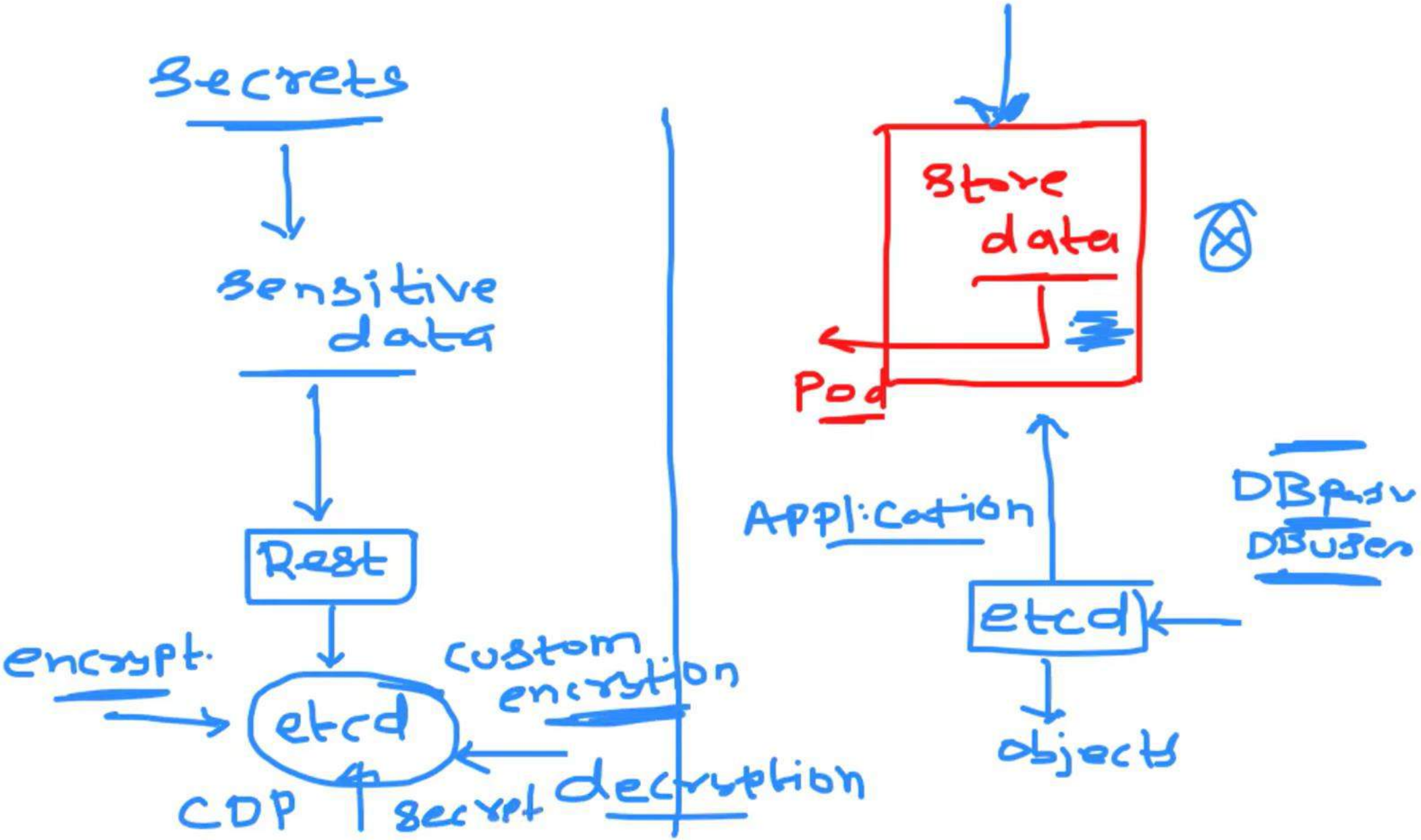
Secrets

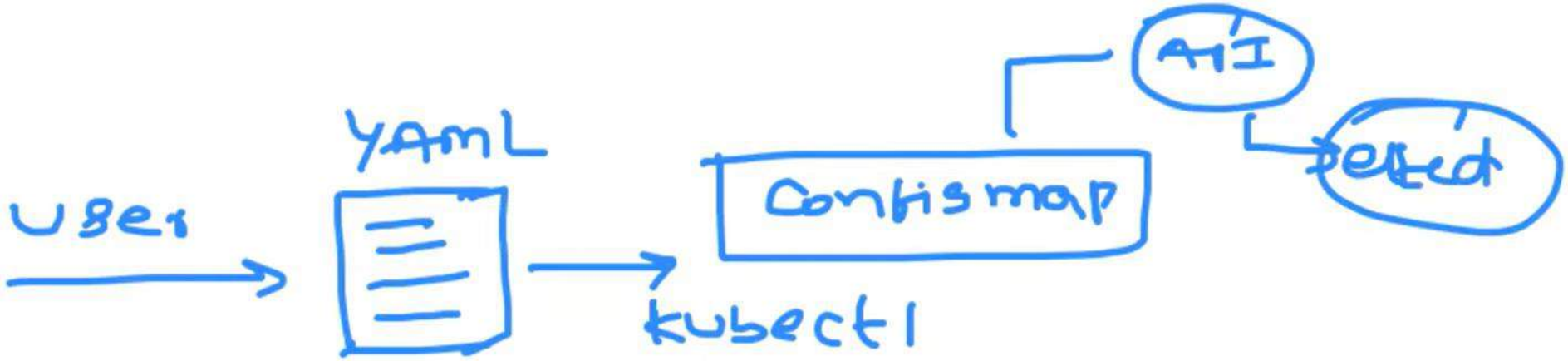
↓

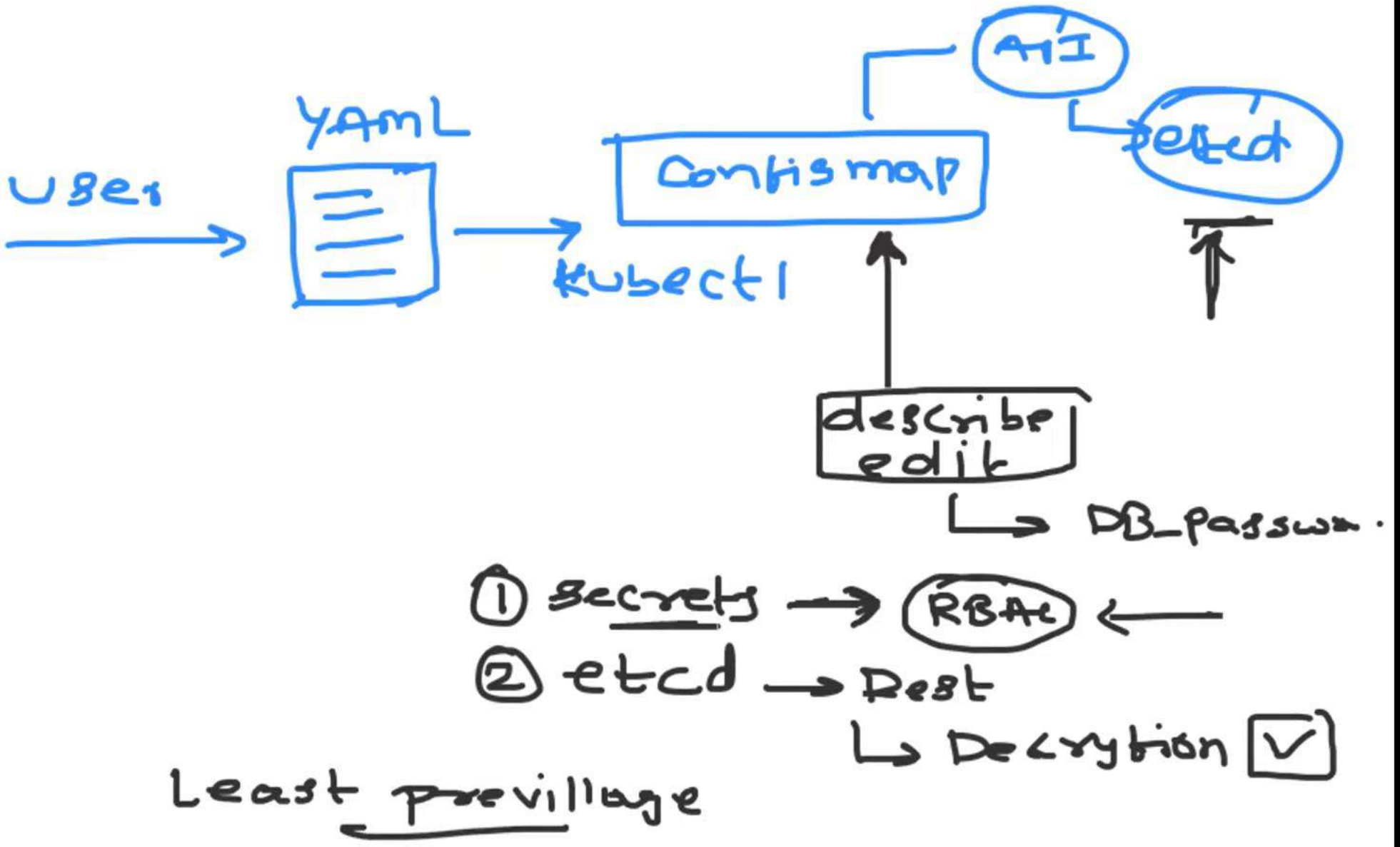
Sensitive data

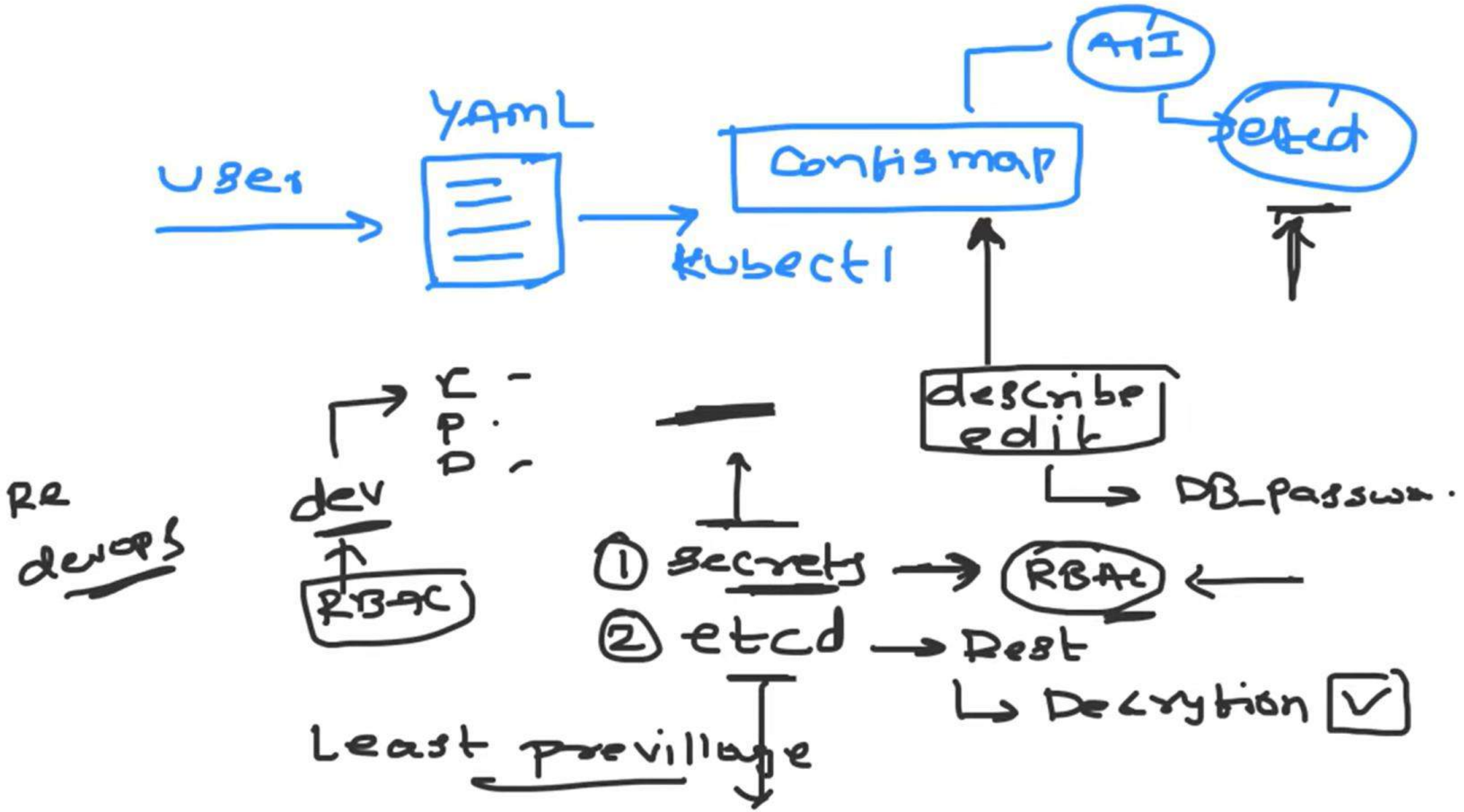



DP



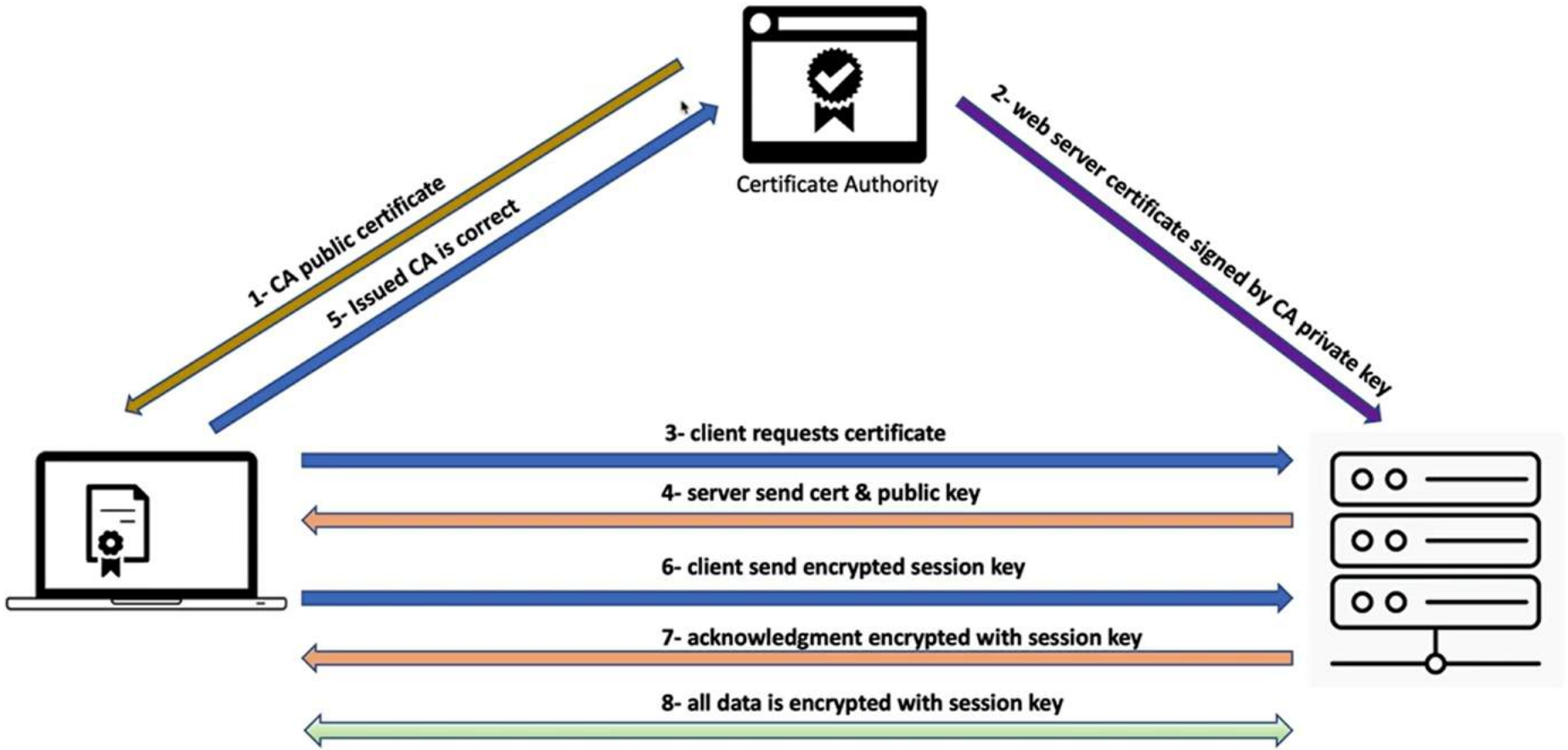





API
etcd
YAML
User
Configmap
kubectl
describe
edit
DB_passwo
RR
devops
dev
RBAC
① secrets
RBAC
② etcd
Rest
Decrytion
Least previllage

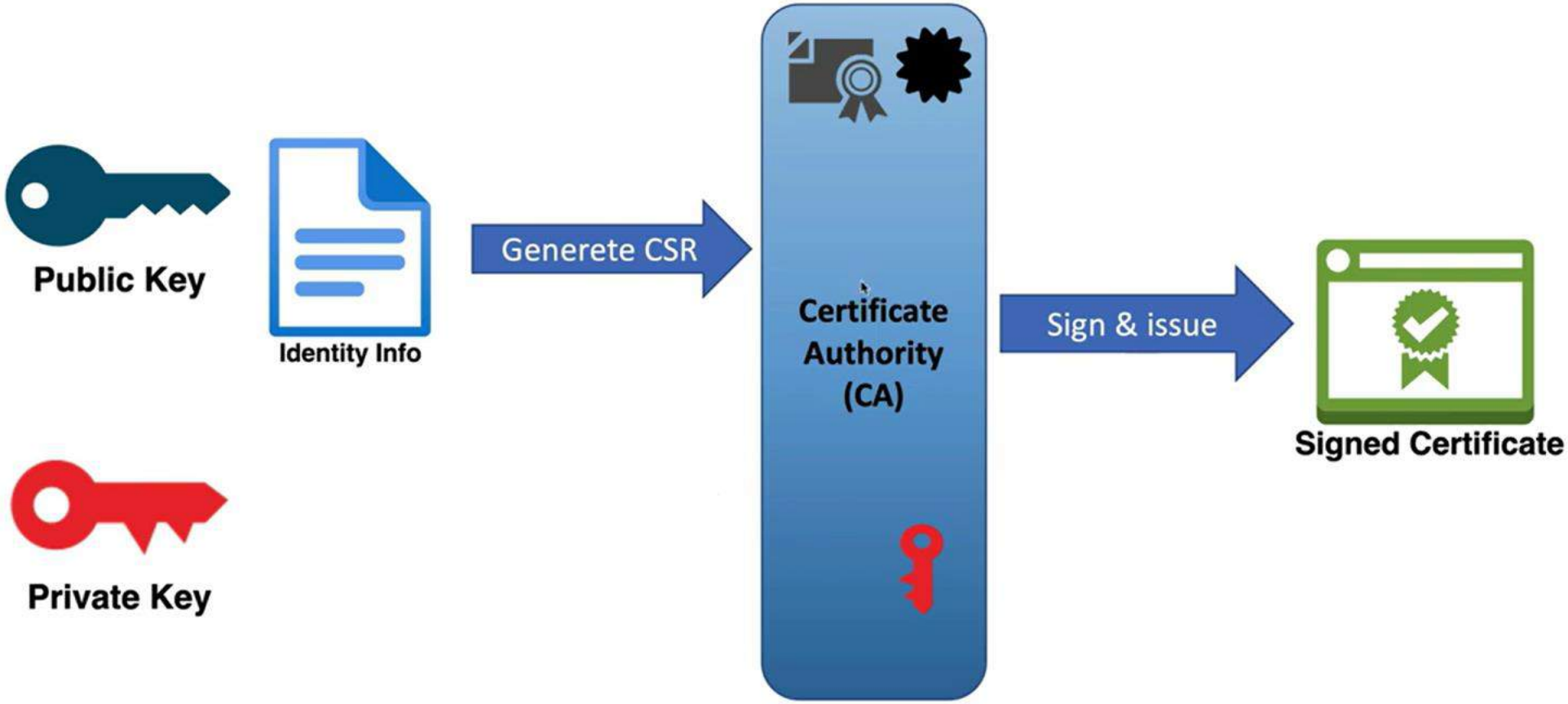
# HTTPS -TLS

Learn with GVR



# Certificate Signing

Learn with GVR




**Public Key**: Who you are
**Private Key**: Prove who you are, decryption

# Kubernetes CA

Learn with GVR

- **Public CA == used for websites**
- **Private CA == used with in Organization**

# Kubernetes CA

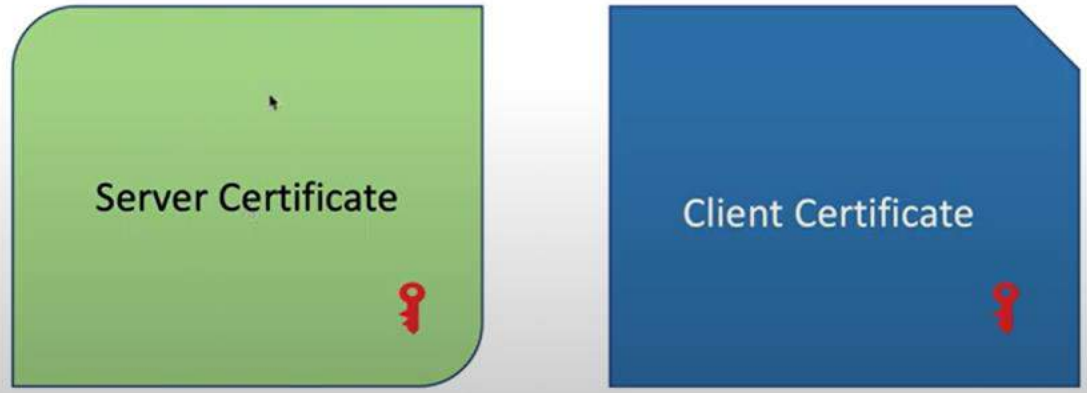
Learn with GVR

- **Public CA == used for websites**
- **Private CA == used with in Organization**
- Kubernetes uses **Private CA** for Components
- Kubernetes CA creates Certs for all components

# Kubernetes CA

Learn with GVR

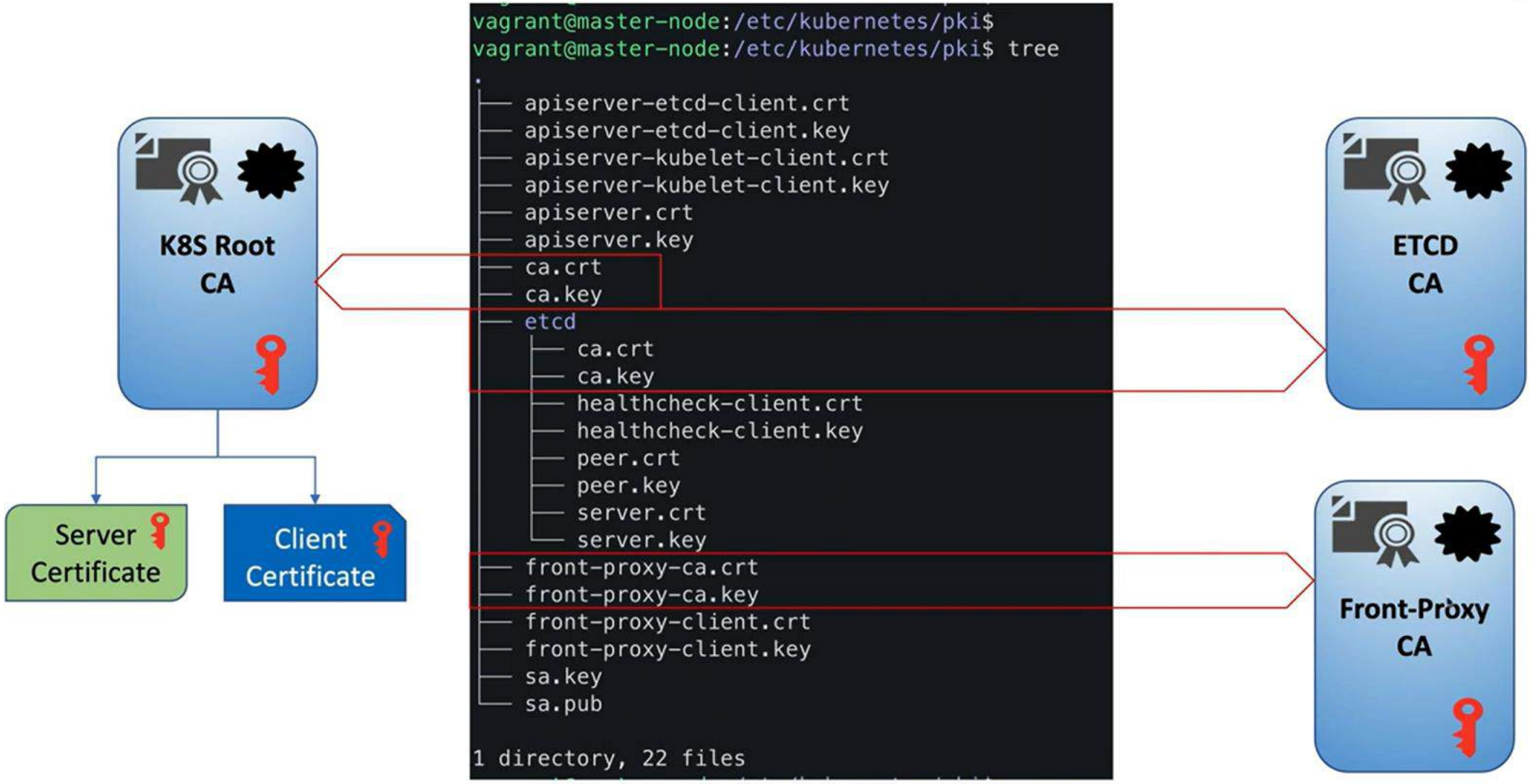
- **Public CA == used for websites**
- **Private CA == used with in Organization**
- Kubernetes uses **Private CA** for Components
- Kubernetes CA creates Certs for all components



```
vagrant@master-node:~$
vagrant@master-node:~$
vagrant@master-node:~$ cd /etc/kubernetes/pki/
vagrant@master-node:/etc/kubernetes/pki$ ll_
```

# Kubernetes CAs

Learn with GVR




```
vagrant@master-node:/etc/kubernetes/pki$
vagrant@master-node:/etc/kubernetes/pki$ tree
.
├── apiserver-etcd-client.crt
├── apiserver-etcd-client.key
├── apiserver-kubelet-client.crt
├── apiserver-kubelet-client.key
├── apiserver.crt
├── apiserver.key
├── ca.crt
├── ca.key
├── etcd
│   ├── ca.crt
│   ├── ca.key
│   ├── healthcheck-client.crt
│   ├── healthcheck-client.key
│   ├── peer.crt
│   ├── peer.key
│   ├── server.crt
│   └── server.key
├── front-proxy-ca.crt
├── front-proxy-ca.key
├── front-proxy-client.crt
├── front-proxy-client.key
├── sa.key
└── sa.pub

1 directory, 22 files
```

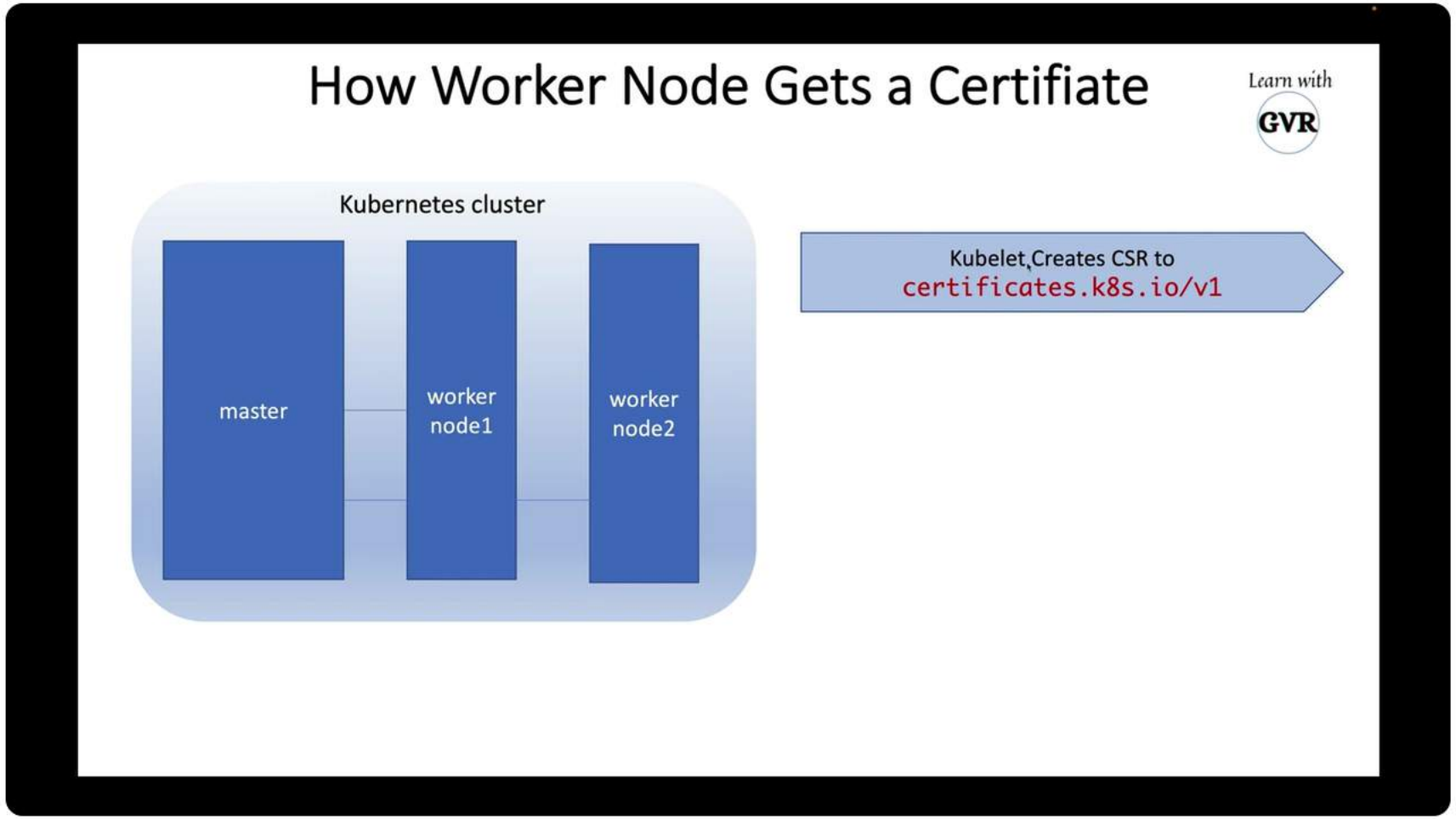
```
root@master-node:/etc/kubernetes/pki# kubeadm certs check-expiration
[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
W0320 02:13:45.902035   38434 utils.go:69] The recommended value for "resolvConf" in "KubeletConfiguration" is: /run/systemd/resolve/resolv.conf; the provid
ed value is: /run/systemd/resolve/resolv.conf

CERTIFICATE                EXPIRES                  RESIDUAL TIME   CERTIFICATE AUTHORITY   EXTERNALLY MANAGED
admin.conf                 Mar 14, 2023 11:17 UTC   359d            ca                      no
apiserver                  Mar 14, 2023 11:17 UTC   359d            ca                      no
apiserver-etcd-client      Mar 14, 2023 11:17 UTC   359d            etcd-ca                 no
apiserver-kubelet-client   Mar 14, 2023 11:17 UTC   359d            ca                      no
controller-manager.conf    Mar 14, 2023 11:17 UTC   359d            ca                      no
etcd-healthcheck-client    Mar 14, 2023 11:17 UTC   359d            etcd-ca                 no
etcd-peer                  Mar 14, 2023 11:17 UTC   359d            etcd-ca                 no
etcd-server                Mar 14, 2023 11:17 UTC   359d            etcd-ca                 no
front-proxy-client         Mar 14, 2023 11:17 UTC   359d            front-proxy-ca          no
scheduler.conf             Mar 14, 2023 11:17 UTC   359d            ca                      no

CERTIFICATE AUTHORITY   EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
ca                      Mar 11, 2032 11:17 UTC   9y              no
etcd-ca                 Mar 11, 2032 11:17 UTC   9y              no
front-proxy-ca          Mar 11, 2032 11:17 UTC   9y              no
root@master-node:/etc/kubernetes/pki#
```

https://www.youtube.com/watch?v=OHmgb7h-2-g

cerificate management in kubernetes telugu




# Kubernetes - Manage TLS Certificates, CA, Certificate Signing Request CSR, Signers, Usage

Learn with GVR
5.17K subscribers
Subscribe

178 Share Download

9K views 2 years ago #k8s #cks #kubenetes




Kubernetes HTTPS with cert-manager and Let's Encrypt
Kubesimplify
7.5K views · 1 year ago

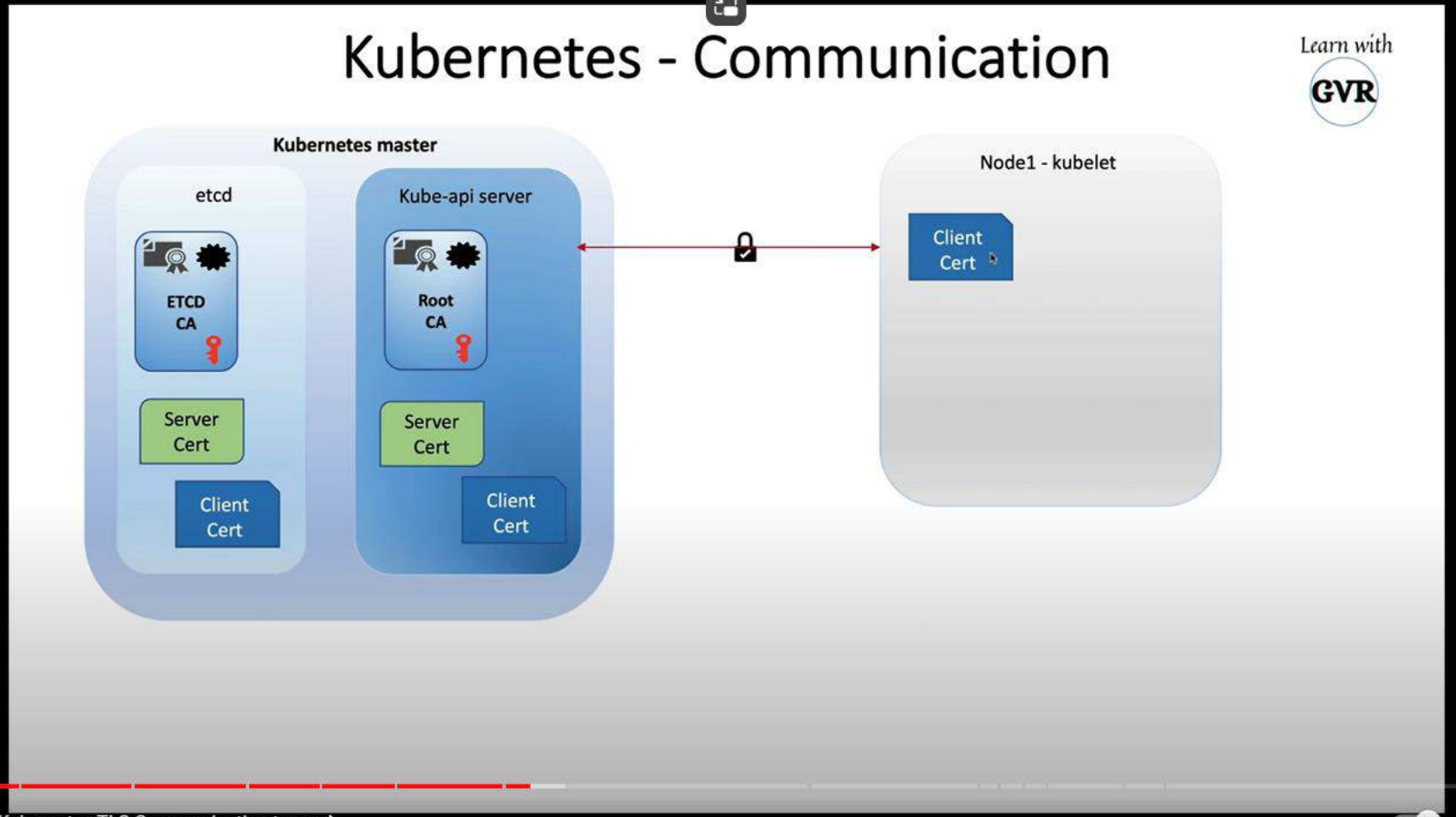

Nginx Ingress Controller & Cert Manager Setup in 2024
david hwang
838 views · 2 months ago


Certifik8s: All You Need to Know About Certificates in...
CNCF [Cloud Native Computing Fou...
43K views · 6 years ago

https://www.youtube.com/watch?v=OHmgb7h-2-g

YouTube

cerificate management in kubernetes telugu




23:33 / 58:14 • Kubernetes TLS Communication types >

# Kubernetes - Manage TLS Certificates, CA, Certificate Signing Request CSR, Signers, Usage

Learn with GVR
5.17K subscribers

Subscribe

178 Share Download

https://www.youtube.com/watch?v=OHmgb7h-2-g

YouTube

cerificate management in kubernetes telugu

```
spec:
  containers:
  - image: ramanagali/hello-world-js:2.0.0
    imagePullPolicy: IfNotPresent
    name: hello-world-js
    ports:
    - containerPort: 3000
      protocol: TCP
    resources: {}
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
    volumeMounts:
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
      name: kube-api-access-9md7f
      readOnly: true
  dnsPolicy: ClusterFirst
  enableServiceLinks: true
  nodeName: worker-node01
  preemptionPolicy: PreemptLowerPriority
  priority: 0
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  serviceAccount: default
  serviceAccountName: default
  terminationGracePeriodSeconds: 30
  tolerations:
  - effect: NoExecute
    key: node.kubernetes.io/not-ready
    operator: Exists
    tolerationSeconds: 300
  - effect: NoExecute
    key: node.kubernetes.io/unreachable
    operator: Exists
    tolerationSeconds: 300
  volumes:
  - name: kube-api-access-9md7f
    projected:
      defaultMode: 420
      sources:
      - serviceAccountToken:
          expirationSeconds: 3607
          path: token
      - configMap:
          items:
```

# Kubernetes - Manage TLS Certificates, CA, Certificate Signing Request CSR, Signers, Usage

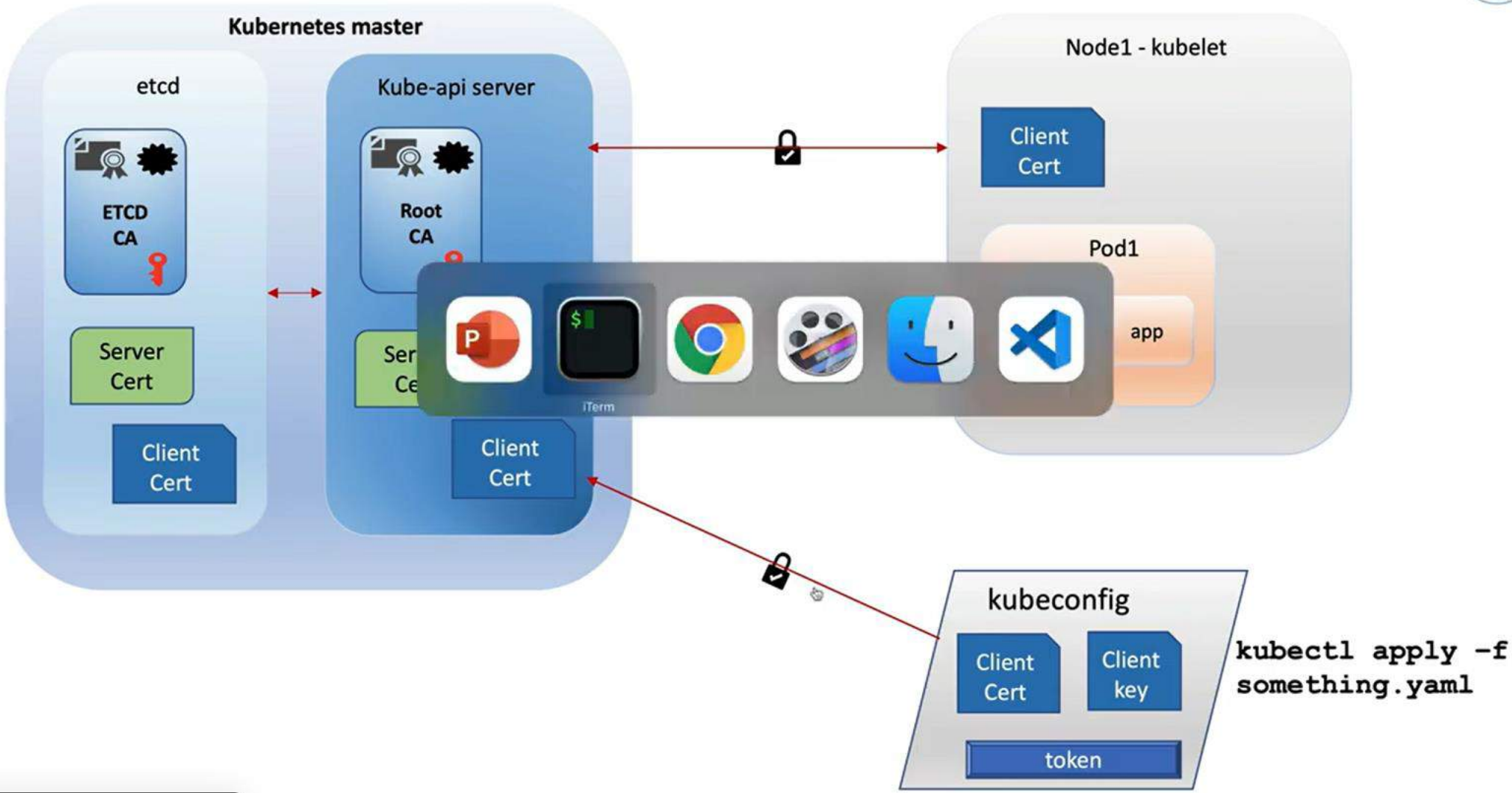
178 | Share | Download

Experiencing interruptions? Find out why

https://www.youtube.com/watch?v=gXz4cq3PKdg

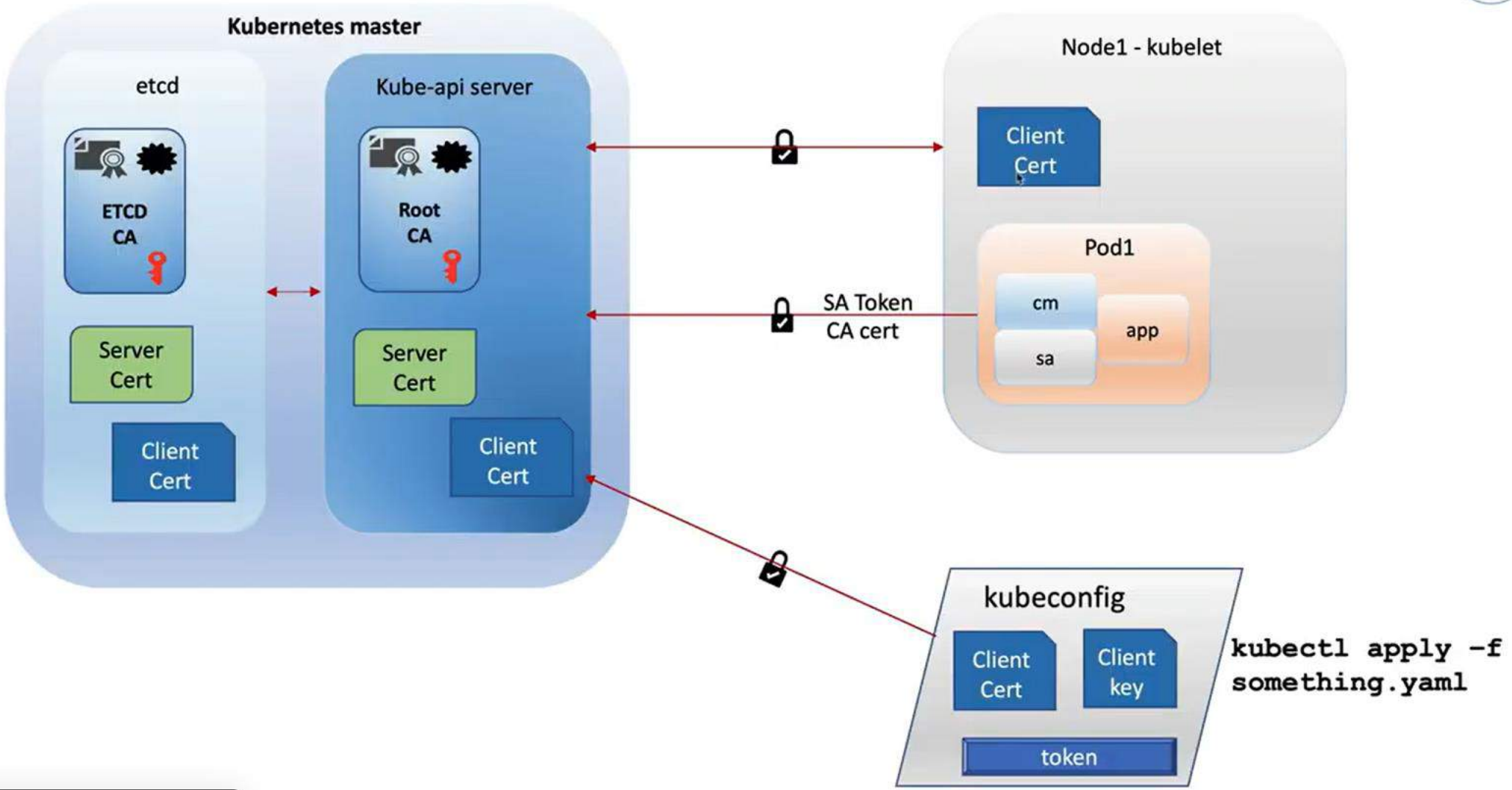
# Kubernetes - Communication

Learn with GVR



# Kubernetes - Communication

Learn with GVR



# Kubernetes CSR Manifest

Learn with GVR

```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: dev-csr
spec:
  groups
    - sy
  request: $(cat venkat.csr | base64 | tr -d '\n')
  signerName: kubernetes.io/kube-apiserver-client
  expirationSeconds: 86400 # one day
  usages:
    - client auth
```

# Kubernetes CSR Manifest

Learn with GVR

```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: hello-world-csr
spec:
  groups:
    - system:authenticated
  request: $(cat server.csr | base64 | tr -d '\n')
  signerName: learnwithgvr.io/serving
  usages:
    - digital signature
    - key encipherment
    - server auth
```

# Kubernetes CSR Manifest

Learn with GVR

```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: hello-world-csr
spec:
  groups:
    - system:authenticated
  request: $(cat server.csr | base64 | tr -d '\n')
  signerName: learnwithgvr.io/serving
  usages:
    - digital signature
    - key encipherment
    - server auth
```

# CSR SIGNERS

Learn with GVR

- 1. "**kubernetes.io/kube-apiserver-client**"
- 2. "**kubernetes.io/kube-apiserver-client-kubelet**"
- 3. "**kubernetes.io/kubelet-serving**"
- 4. **Custom signerNames** (ex: learnwithgvr.io/serving )

# CSR KEY USAGE

Learn with GVR

- Indicates Key Usage

```
Valid values are:
"signing", "digital signature", "content commitment", "key encipherment",
"key agreement", "data encipherment", "cert sign", "crl sign", "encipher
only", "decipher only", "any", "server auth", "client auth", "code
signing", "email protection", "s/mime", "ipsec end system", "ipsec tunnel",
"ipsec user", "timestamping", "ocsp signing", "microsoft sgc", "netscape
sgc"
```

# CSR KEY USAGE

Learn with GVR

- Indicates Key Usage

```
Valid values are:
"signing", "digital signature", "content commitment", "key encipherment",
"key agreement", "data encipherment", "cert sign", "crl sign", "encipher
only", "decipher only", "any", "server auth", "client auth", "code
signing", "email protection", "s/mime", "ipsec end system", "ipsec tunnel",
"ipsec user", "timestamping", "ocsp signing", "microsoft sgc", "netscape
sgc"
```

Requests for **TLS client certificates** typically request:

**"digital signature", "key encipherment", "client auth"**

# csr commands

Learn with GVR

```
kubectl get csr
kubectl describe csr my-csr
```

# csr commands

Learn with GVR

```
kubectl get csr
kubectl describe csr my-csr
kubectl certificate approve my-csr
```

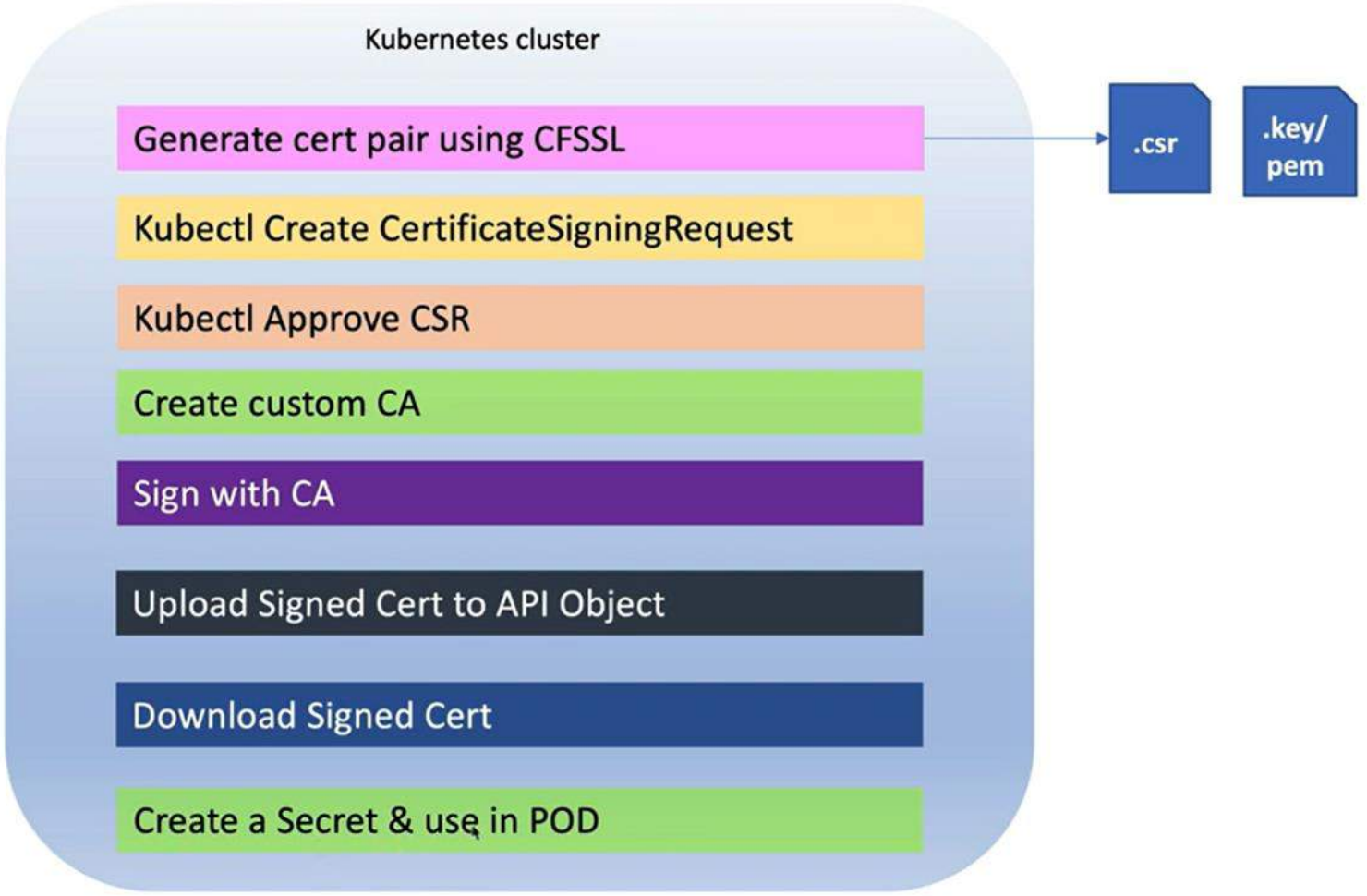
# csr commands

Learn with GVR

```
kubectl get csr
kubectl describe csr my-csr
kubectl certificate approve my-csr
```

# CSR – Custom CA Steps

Learn with GVR



# USE CASE

Learn with GVR

Create a Secret & use in POD

To create Kubeconfig User – talk to Kube API Server

xyz service - Admission controller webhook -– Kube API Server will call

# USE CASE

Learn with GVR

Create a Secret & use in POD

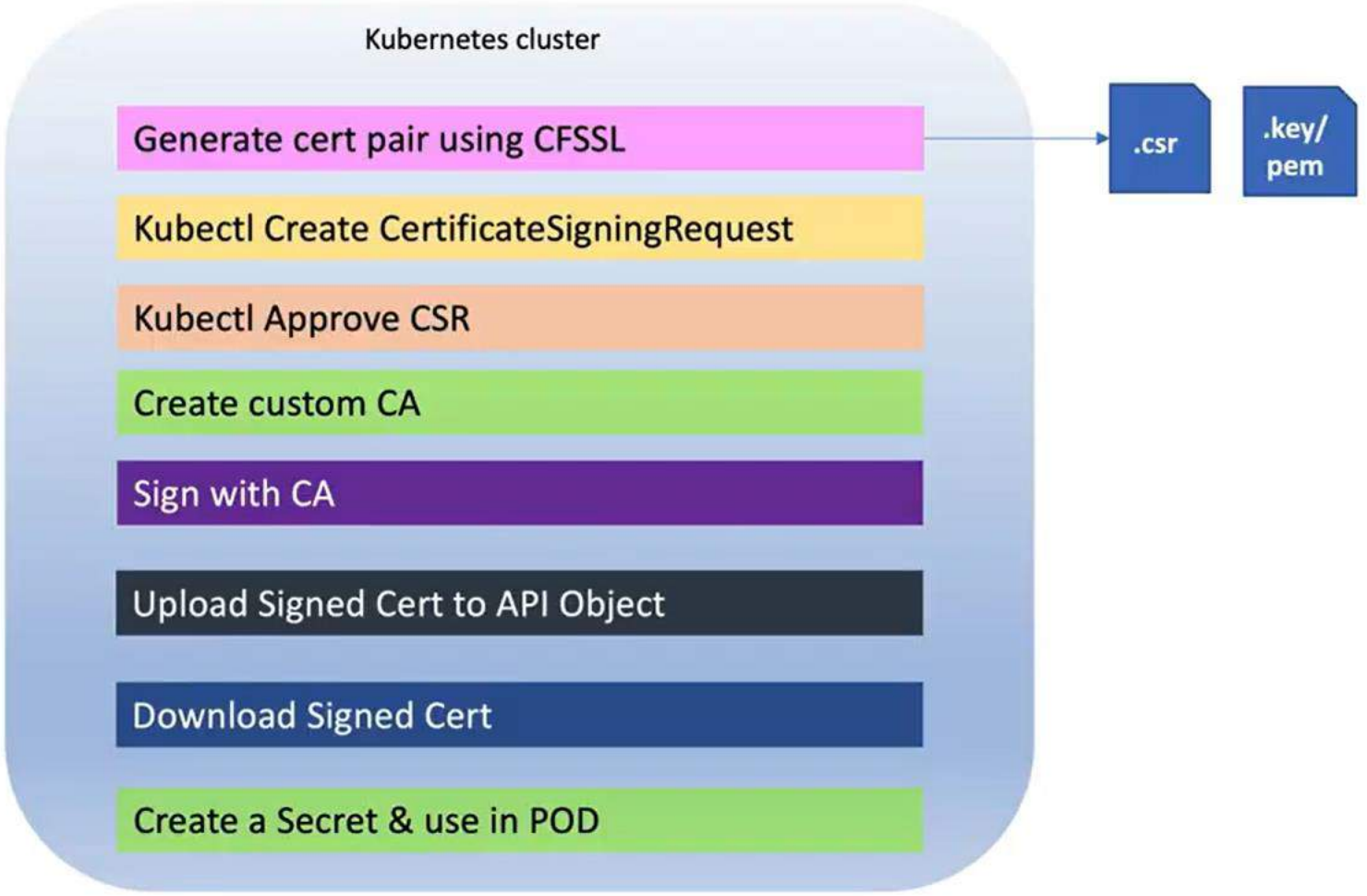
To create Kubeconfig User – talk to Kube API Server

xyz service - Admission controller webhook -– Kube API Server will call

xyz service from ABC org – dual communication with Kubernetes

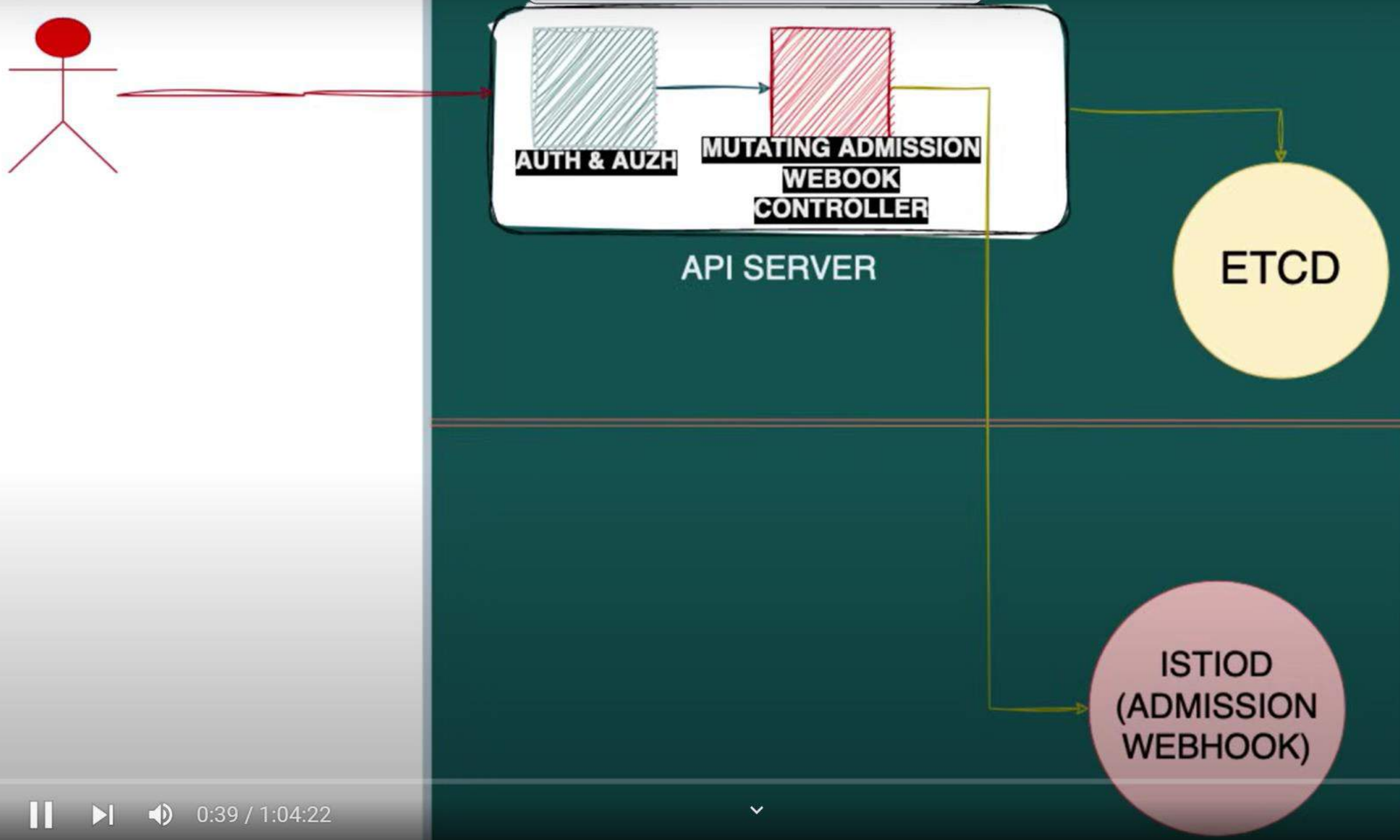
# CSR – Custom CA Steps

Learn with GVR

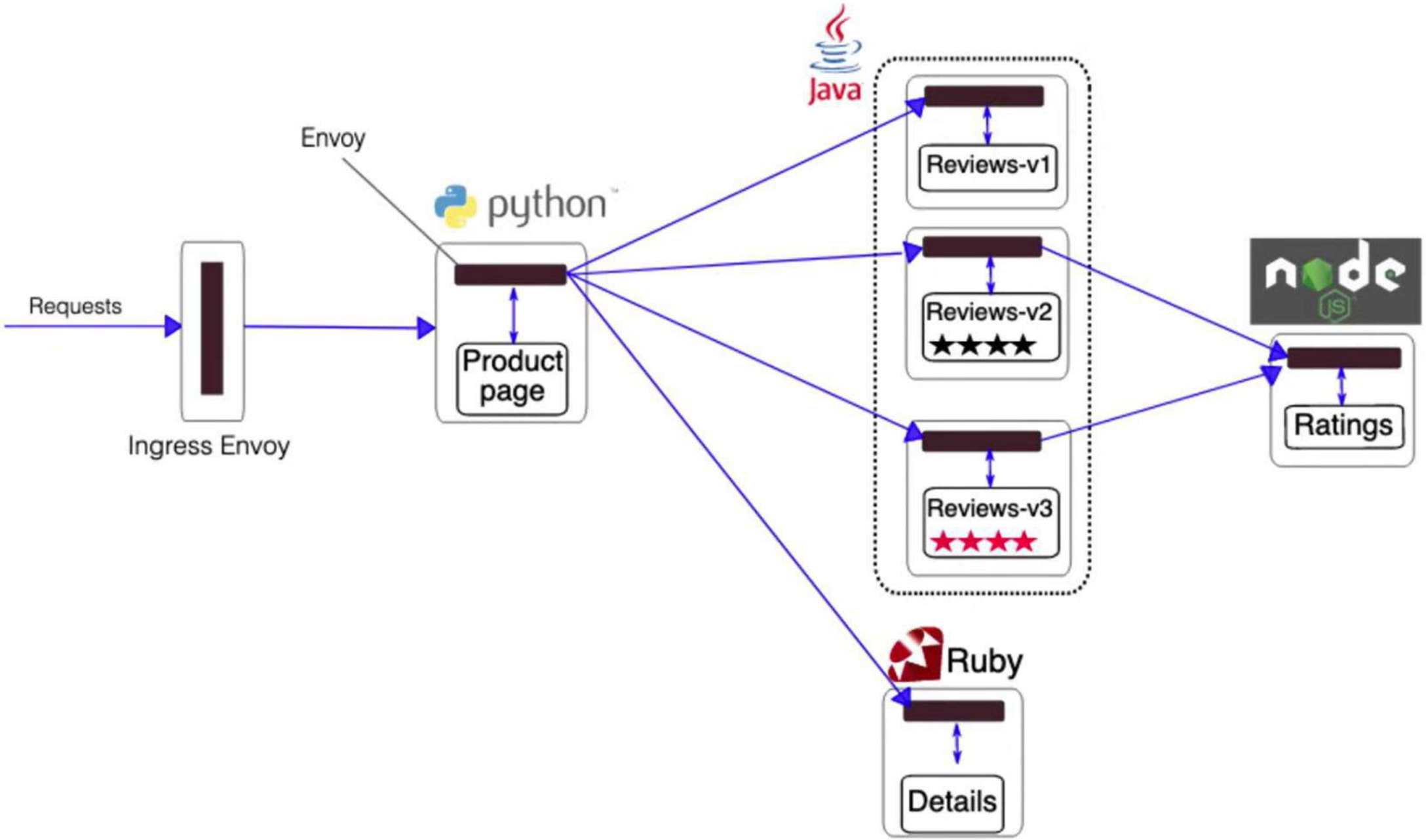


Service Mesh explained in 60 minutes | Istio mTLS and Canary Demo | Complete beginner level guide

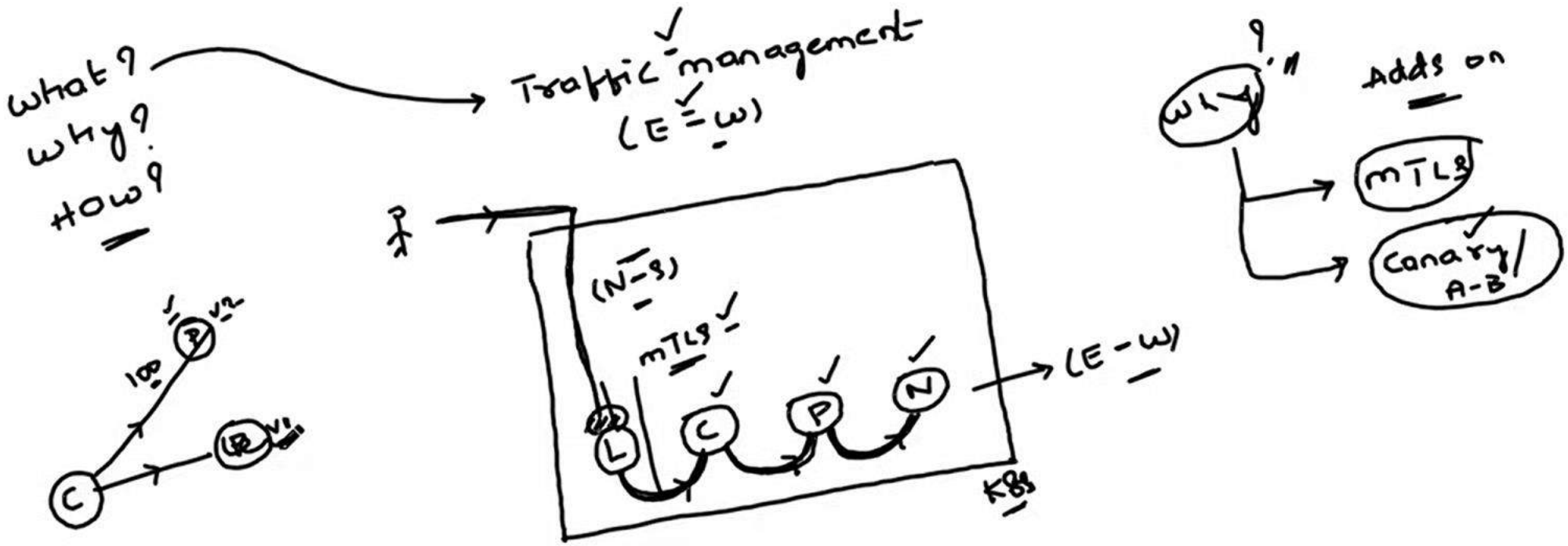
youtube.com - To exit full screen, press Esc




0:39 / 1:04:22




Bookinfo Application



# Traffic Management

Tasks that demonstrate Istio's traffic routing features.

## Request Routing

This task shows you how to configure dynamic request routing to multiple versions of a microservice.

## Fault Injection

This task shows you how to inject faults to test the resiliency of your application.

## Traffic

Shows
new ve

## TCP Traffic Shifting

Shows you how to migrate TCP traffic from an old to new version of a TCP service.

## Request Timeouts

This task shows you how to set up request timeouts in Envoy using Istio.

## Circui

This ta
breakin
detecti

## Mirroring

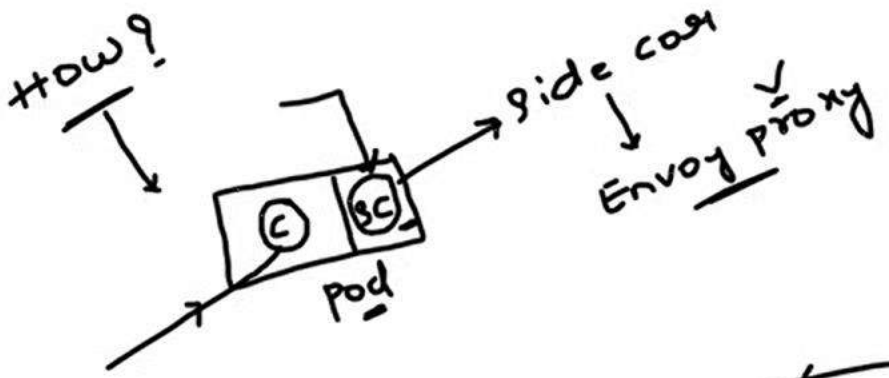
This task demonstrates the traffic mirroring/shadowing capabilities of Istio.

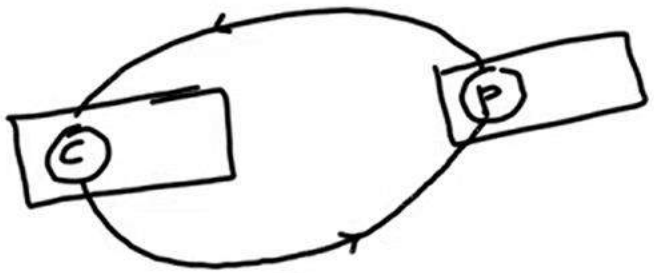
## Locality Load Balancing

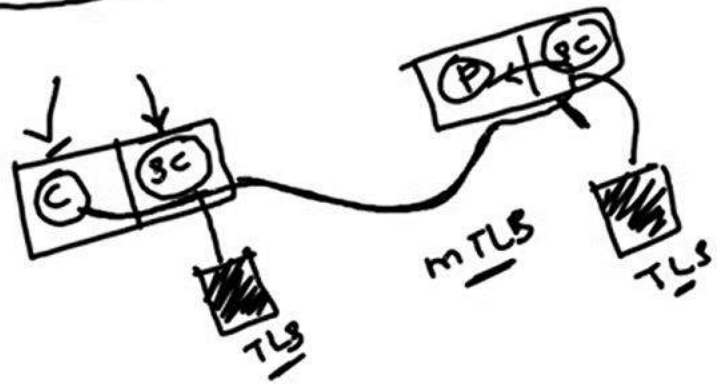
This series of tasks demonstrate how to configure locality load balancing in Istio.

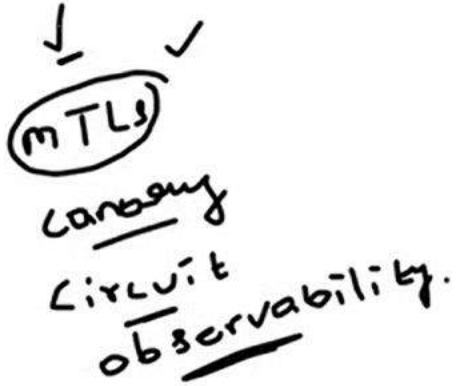
## Ingres

Contro
mesh.




Search this site

- Documentation
- Getting started
- Concepts
- Tasks
- Tutorials
- Reference
  - Glossary
  - API Overview
  - API Access Control
    - Authenticating
    - Authenticating with Bootstrap Tokens
    - Certificates and Certificate Signing Requests
    - **Admission Controllers**
    - Dynamic Admission Control
    - Managing Service Accounts

Kubernetes Documentation / Reference / API Access Control / Admission Controllers

# Admission Controllers Reference

This page provides an overview of Admission Controllers.

## What are they?

An *admission controller* is a piece of code that intercepts requests to the Kubernetes API server prior to persistence of the object, but after the request is authenticated and authorized.

Admission controllers may be *validating*, *mutating*, or both. Mutating controllers may modify objects related to the requests they admit; validating controllers may not.

Admission controllers limit requests to create, delete, modify objects. Admission controllers can also block custom verbs, such as a request connect to a Pod via an API server proxy. Admission controllers do *not* (and cannot) block requests to read (**get**, **watch** or **list**) objects.

The admission controllers in Kubernetes 1.30 consist of the list below, are compiled into the `kube-apiserver` binary, and may only be configured by the cluster administrator. In that list, there are two special controllers: MutatingAdmissionWebhook and ValidatingAdmissionWebhook. These execute the mutating and validating (respectively) admission control webhooks which are

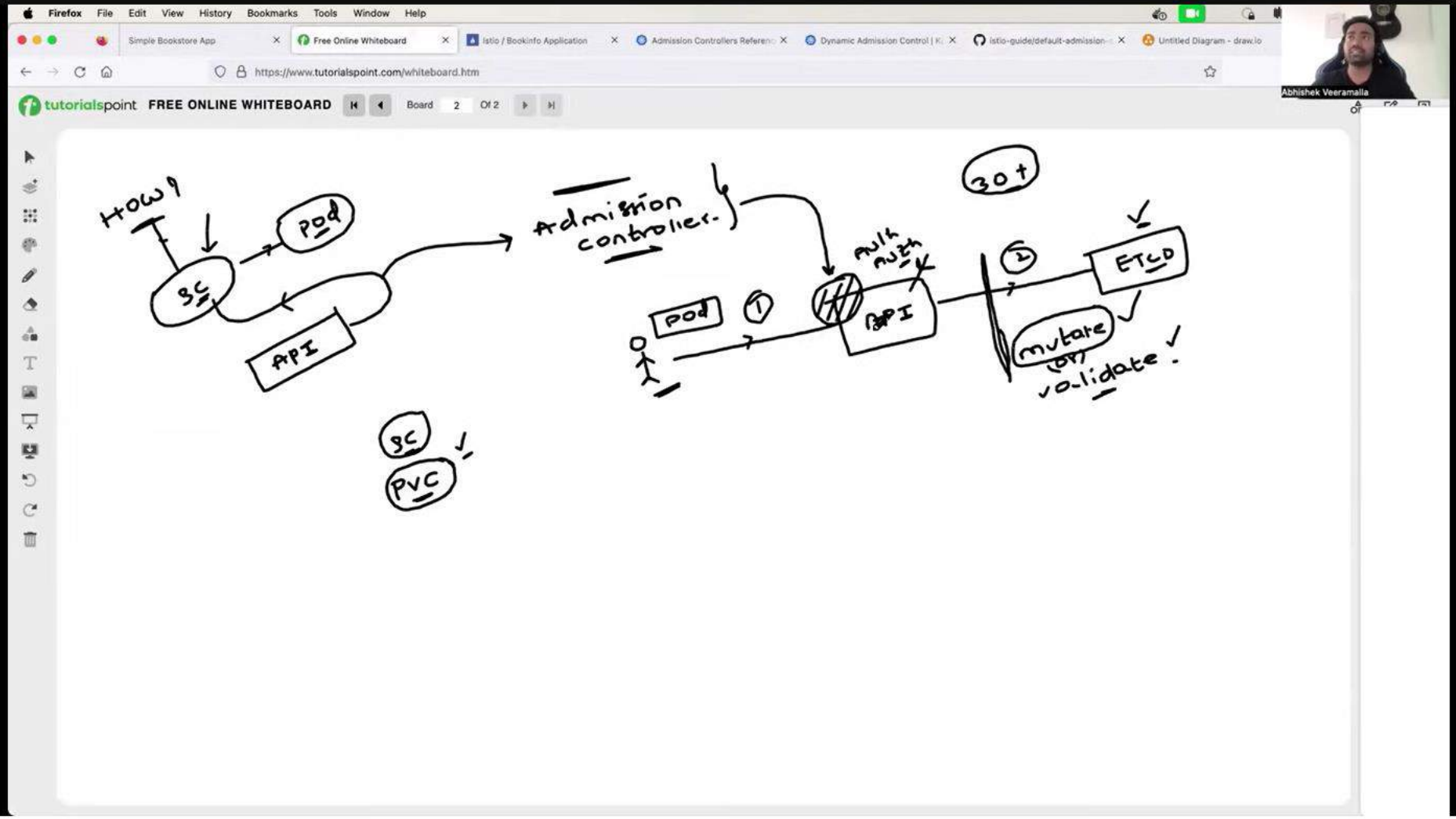
Which plugins are enabled by default?
What does each admission controller do?
AlwaysAdmit
AlwaysDeny
AlwaysPullImages
CertificateApproval
CertificateSigning
CertificateSubjectRestriction
DefaultIngressClass
DefaultStorageClass
DefaultTolerationSeconds
DenyServiceExternalIPs
EventRateLimit
ExtendedResourceToleration
ImagePolicyWebhook
LimitPodHardAntiAffinityTopology
LimitRanger
MutatingAdmissionWebhook
NamespaceAutoProvision
NamespaceExists
NamespaceLifecycle
NodeRestriction
OwnerReferencesPermissionEnforcement

prometheus-Grafan | Jenkins erros.pdf | Santhosha Nagendr | Confirmed - Calend | Service Mesh e | istio-guide/default- | can f2 depandant b

https://www.youtube.com/watch?v=eSNetKBe7Z8&list=PLdpzxOOAlwvJdsW6A0jCz_3VaANuFMLpc&index=22

YouTube

Search




# Service Mesh explained in 60 minutes | Istio mTLS and Canary Demo | Complete beginner level guide

Abhishek.Veeramalla
294K subscribers

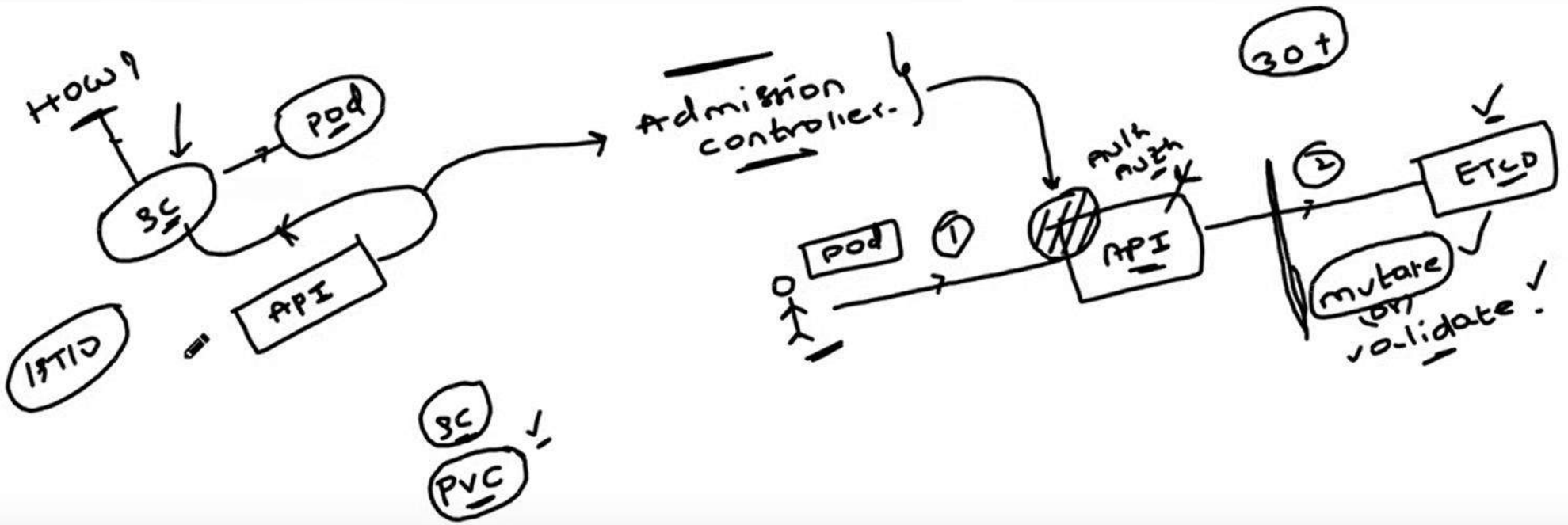
Subscribe

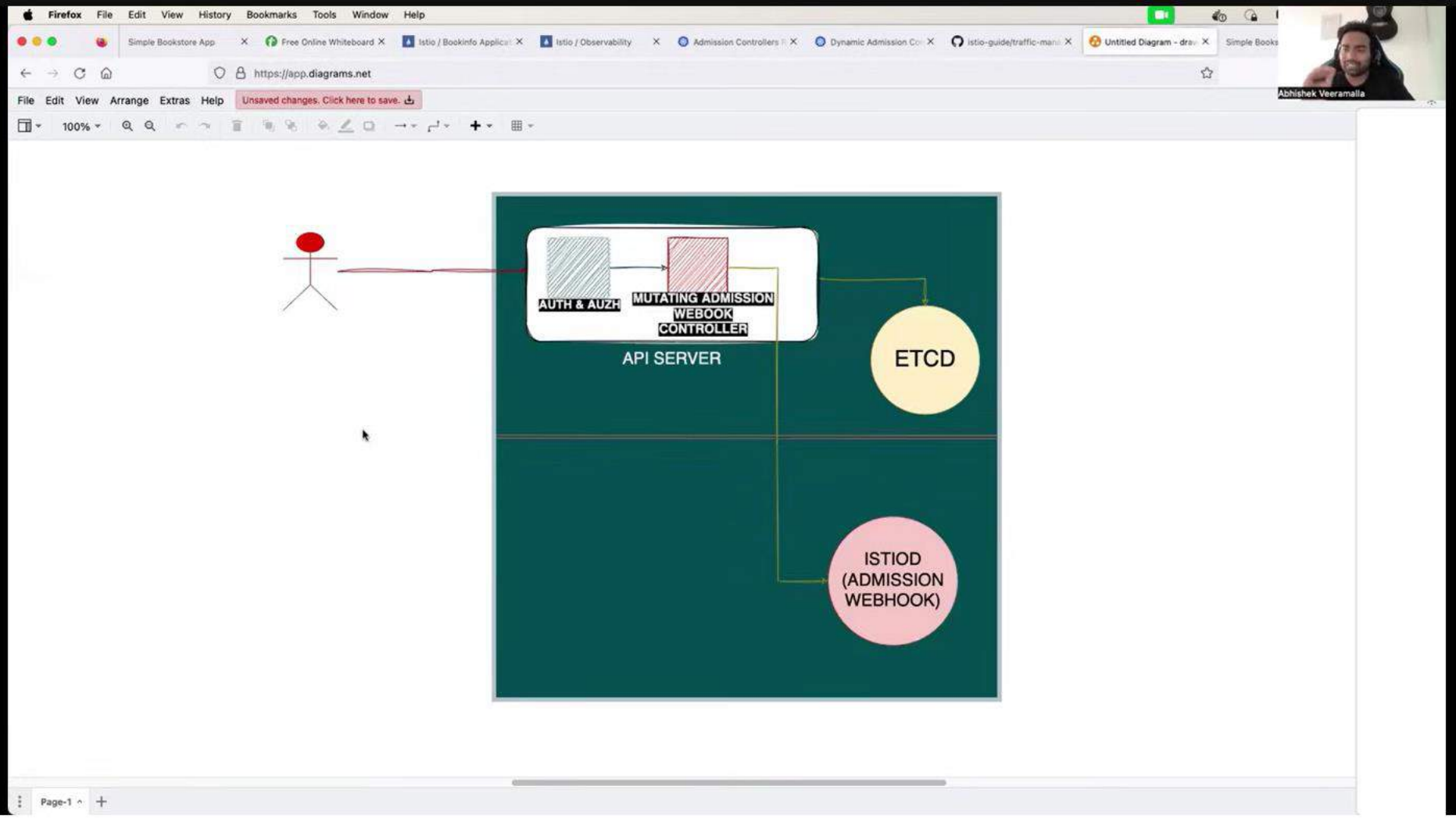
616 | Share | Download

**Kubernetes**

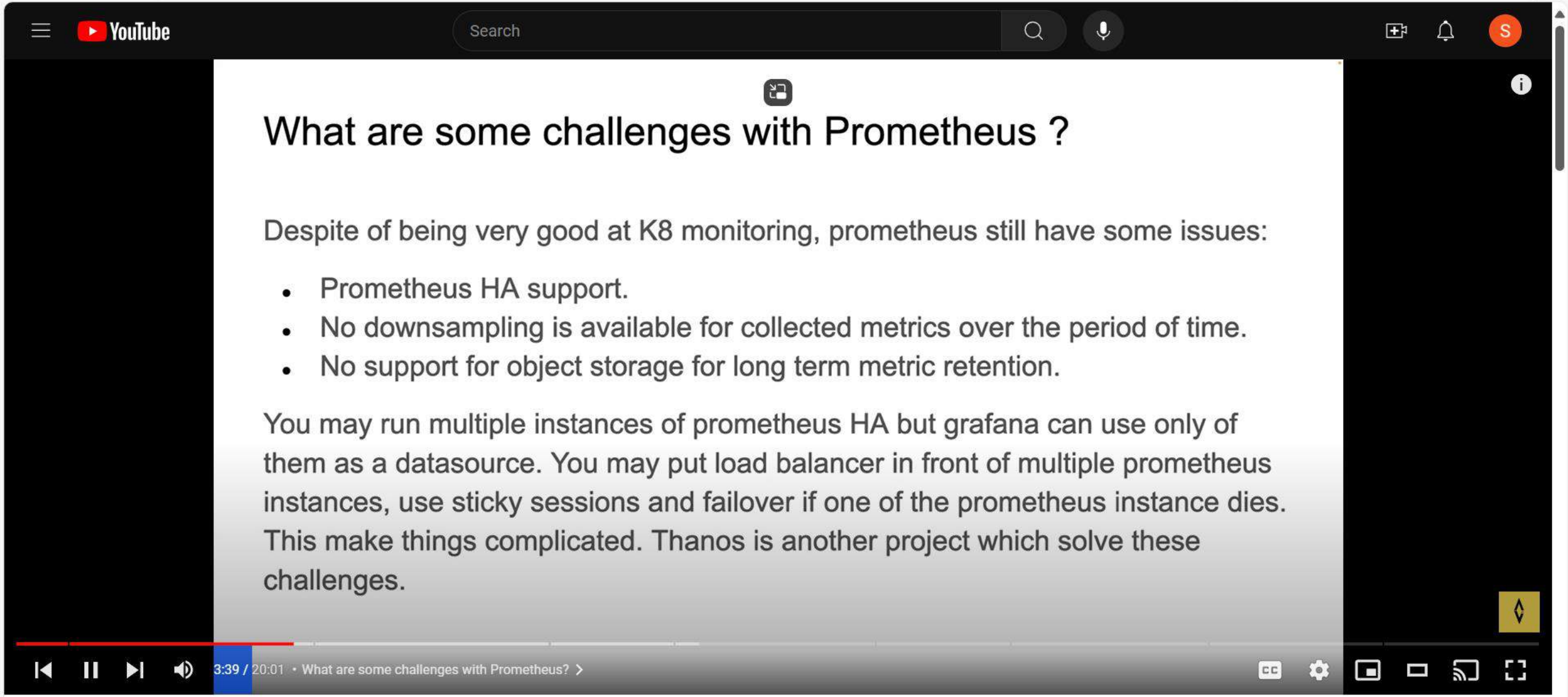
Abhishek.Veeramalla - 22 / 31

# Service Mesh explained in 60 minutes | Istio mTLS and Canary Demo | Complete beginner level guide






# Service Mesh explained in 60 minutes | Istio mTLS and Canary Demo | Complete beginner level guide



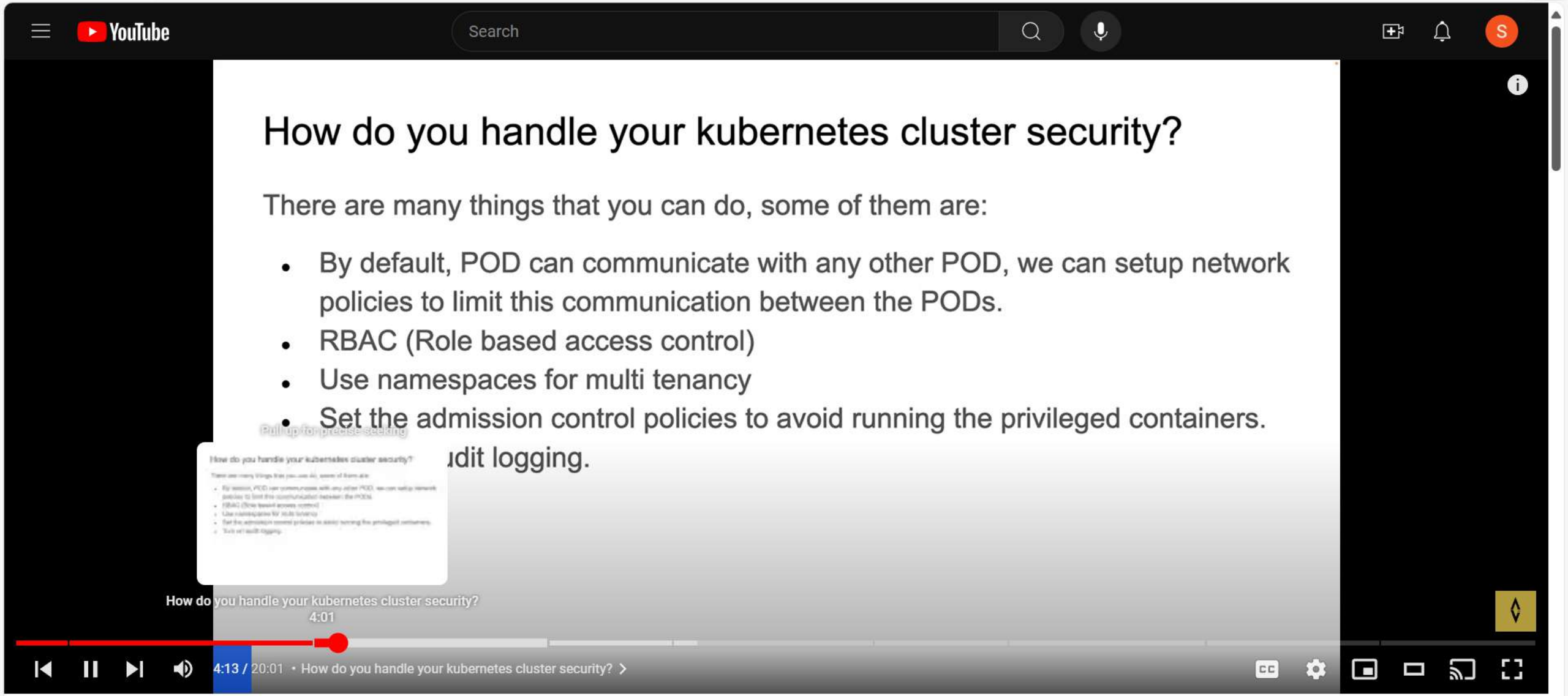

**Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview**

Abhishek.Veeramalla
294K subscribers



Chapters

What are some




# Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview

Abhishek.Veeramalla
294K subscribers

Subscribe

993 Share Download

Chapters

How do you handle your

# How two containers running in a single POD have single IP address?

Kubernetes makes use of Pause containers for sharing networking.

Kubernetes implements this by creating a special container for each pod whose only purpose is to provide a network interface for the other containers. These is one pause container which is responsible for namespace sharing in the POD. Generally, people ignore the existence of this pause container but actually this container is the heart of network and other functionalities of POD. It provides a single virtual interface which is used by all containers running in a POD.

**Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview**

Abhishek.Veeramalla
294K subscribers

# How two containers running in a single POD have single IP address?

Kubernetes makes use of Pause containers for sharing networking.

Kubernetes implements this by creating a special container for each pod whose only purpose is to provide a network interface for the other containers. These is one pause container which is responsible for namespace sharing in the POD. Generally, people ignore the existence of this pause container but actually this container is the heart of network and other functionalities of POD. It provides a single virtual interface which is used by all containers running in a POD.

**Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview**

# What is Service Mesh and why do we need it ?

A service mesh ensures that communication among containerized and often ephemeral application infrastructure services is fast, reliable, and secure. The mesh provides critical capabilities including service discovery, load balancing, encryption, observability, traceability, authentication and authorization, and support for the circuit breaker pattern.

## Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview

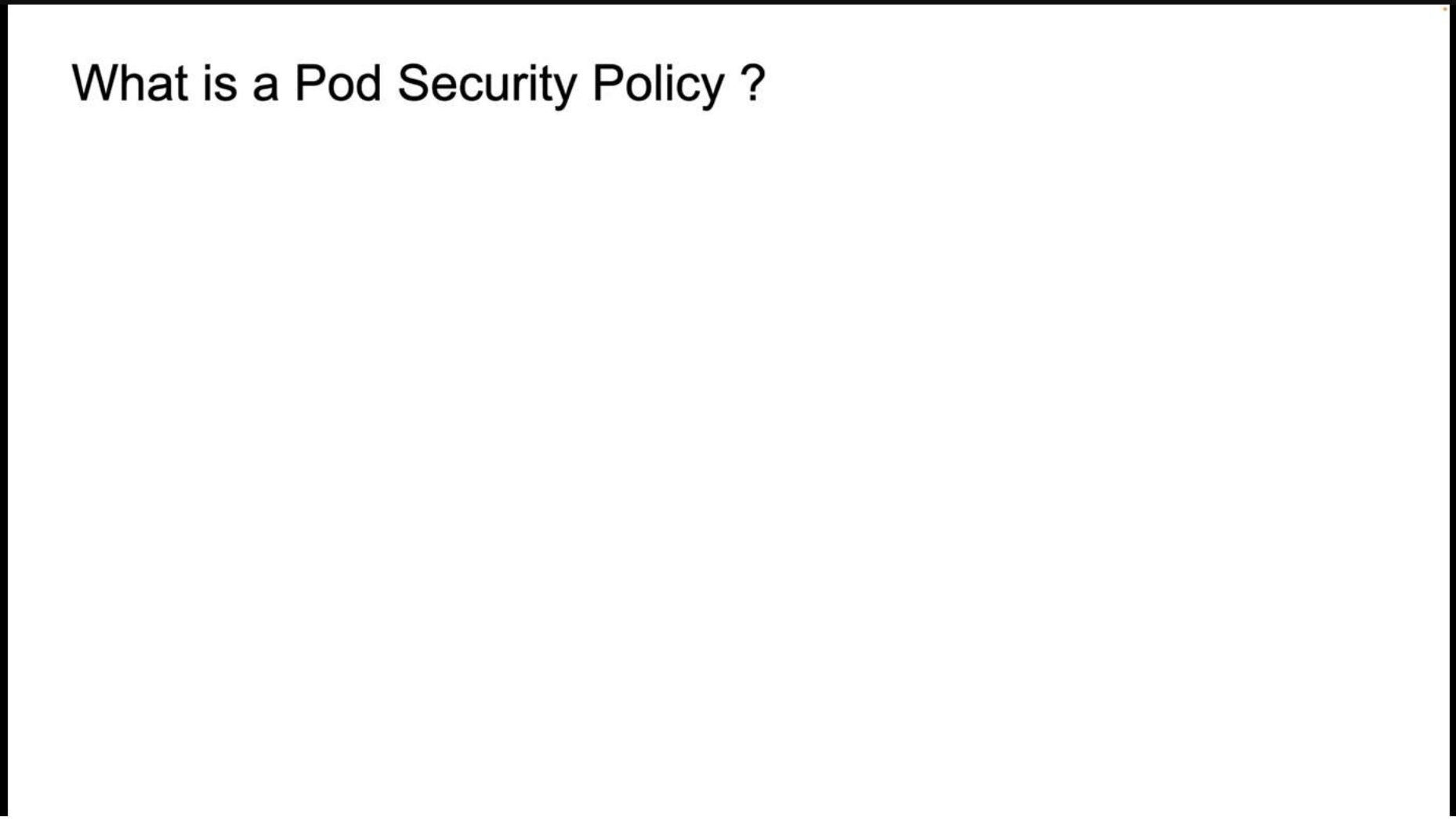
# What is init container and why do we need it ?

Init Containers are the containers that should run and complete before the startup of the main container in the pod. It provides a separate lifecycle for the initialization so that it enables separation of concerns in the applications.

All the init Containers will be executed sequentially and if there is an error in the Init container the pod will be restarted which means all the Init containers are executed again. So, it's better to design your Init container as simple, quick, and Idempotent.

source:
https://medium.com/bb-tutorials-and-thoughts/kubernetes-interview-questions-part-1-eb88a9df785f

**Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview**



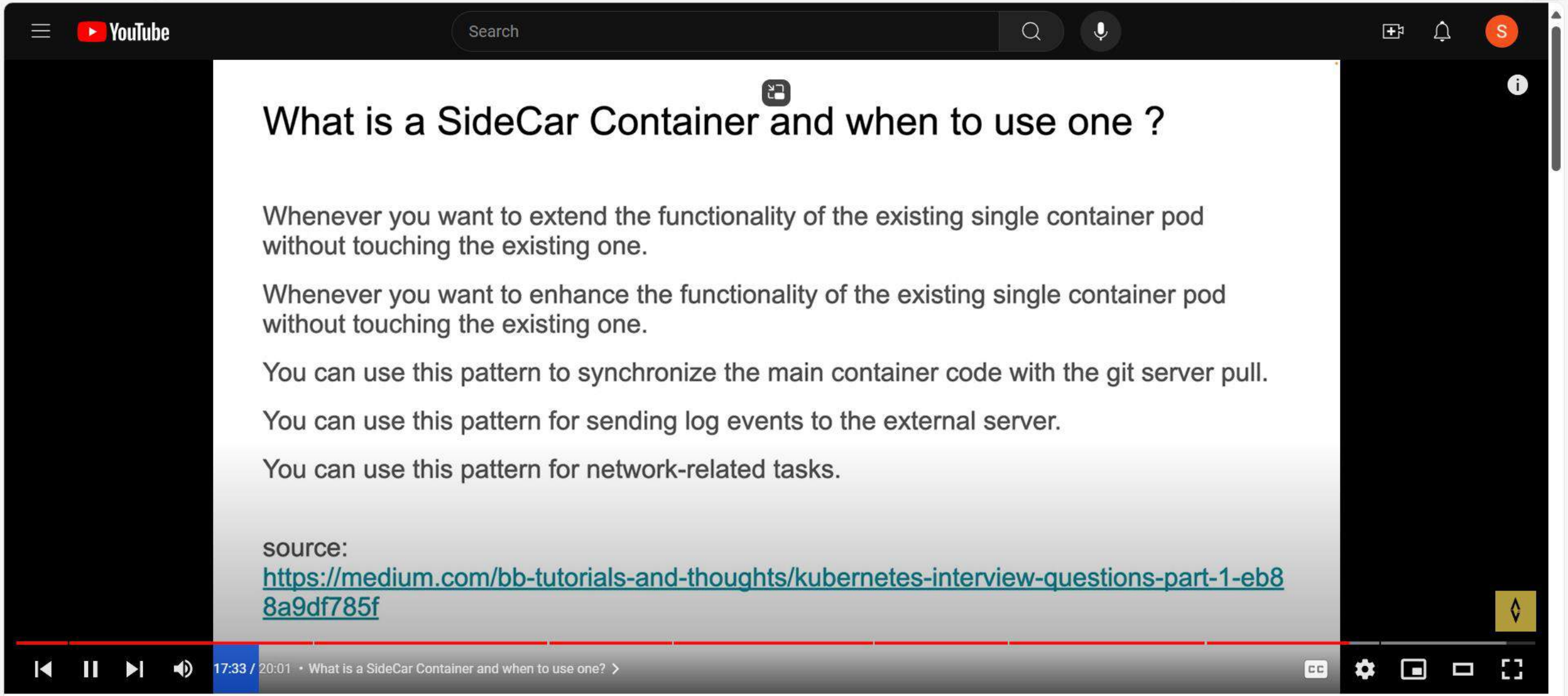

# Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview

Abhishek.Veeramalla
294K subscribers



## Chapters

What is Service Mesh and why do we




# Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview

Abhishek.Veeramalla
294K subscribers

Subscribe

993 Share Download

Chapters

What is Service Mesh and why do we

# What is a SideCar Container and when to use one ?

Whenever you want to extend the functionality of the existing single container pod without touching the existing one.

Whenever you want to enhance the functionality of the existing single container pod without touching the existing one.

You can use this pattern to synchronize the main container code with the git server pull.

You can use this pattern for sending log events to the external server.

You can use this pattern for network-related tasks.

source:
https://medium.com/bb-tutorials-and-thoughts/kubernetes-interview-questions-part-1-eb88a9df785f

**Kubernetes Toughest Interview Scenarios & Questions | How many can you answer ? |#devops #interview**