**Api server - to communicate with kubectl command line it is front end**

**Controller manger - kubelet to creating the pod how much pod in running and creating**

**Scheduler - scheduler the pods activity**

**Etcd - key as store in cluster details**

**Kube-proxy - the logical bridge app user and the pods**

**Kubelet - kubectl command to communicate to the pods using kubelet agent for workernodes**

**Pod - group of container inside application running**

**port redirects the traffic to the container from the service. NodePort : is the port that enables the service to access the externally**

**Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.**

Kops, kubeadm, kubespray, miniqube Version 1.20.0

----------------------Kops Binary setup-----------------------

curl -Lo kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4)/kops-linux-amd64

chmod +x kops

sudo mv kops /usr/local/bin/kops

chmod +x ./kops

mv ./kops /usr/local/bin/

-----------------------Kubectl binary setup-------------------

curl -Lo kubectl https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

chmod +x ./kubectl

mv ./kubectl /usr/local/bin/

aws configure

access id AKIAWLMGZOS525YPZPOO

access key 7jnmH8rEnN/MkLJkKo5v8Y+chKjB9DaGoJVhjTCL

-----------------enviroment vaiable----------------------------

ex export age=1 any path will access echo suresh $age

export AWS\_ACCESS\_KEY\_ID=AKIAWLMGZOS525YPZPOO

export AWS\_SECRET\_ACCESS\_KEY=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

-------------s3-------------- CREATE BUCKETS

aws s3 ls

aws s3 mb s3://newkcops --region ap-south-1

delete----------------aws s3 rb s3://bucket-name --force

-------------version enable------------

aws s3api put-bucket-versioning --bucket newkcops --versioning-configuration Status=Enabled

-------------ssh client machine connetions enviroment to master node and worker node-----------

ssh-keygen

Gossip based clusters use a peer to peer network instead of externally hosted DNS for proagation the k8s api address

---------DNS domain------aws or other environment persuring

export NAME=suresh.k8s.local

export KOPS\_STATE\_STORE=s3://newkcops

------------Cluster create a command-----------

kops create cluster --zones ap-south-1a ${NAME}

**Finally configure your cluster with:**

kops update cluster --name suresh.k8s.local --yes --admin

**validate cluster:**

kops validate cluster --wait 10m

----------login master node---------

ssh to the master: ssh -i ~/.ssh/id\_rsa ubuntu@api.suresh.k8s.local

Suggestions: more master node

\* list clusters with: kops get cluster

kubectl get nodes

\* edit this cluster with: kops edit cluster suresh.k8s.local

\* edit your node instance group: kops edit ig --name=suresh.k8s.local nodes-ap-south-1a

kops update cluster

kops update cluster --yes

kops rolling-update cluster

\* edit your master instance group: kops edit ig --name=suresh.k8s.local master-ap-south-1a

ssh -i ~/.ssh/id\_rsa ubuntu@ip master connection

kubectl get nodes

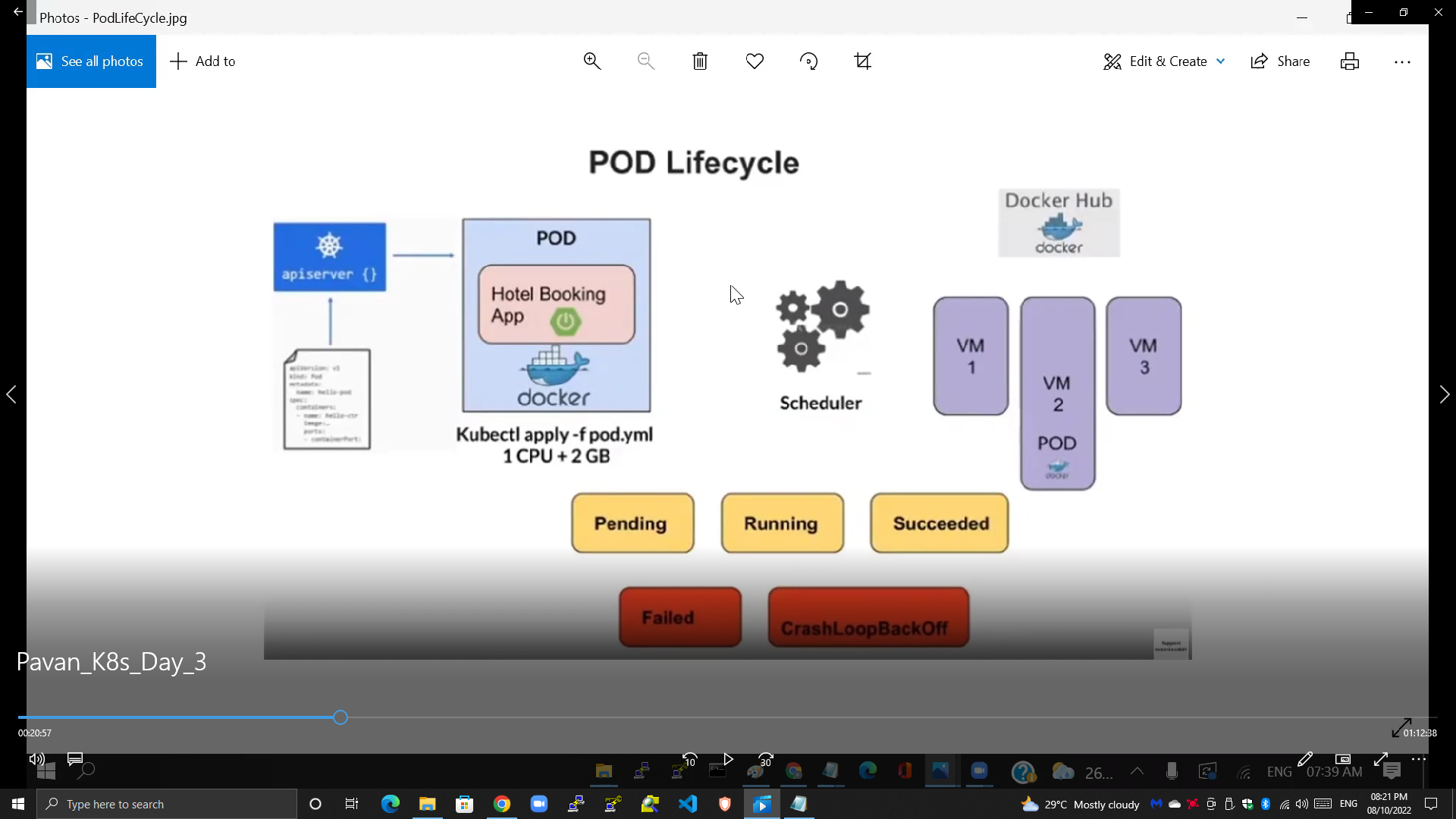
—--cops delete—----

kops delete cluster suresh.k8s.local

Interview

Launch configurations to update and lauch template - editable and version can be done

Pod lifecycle



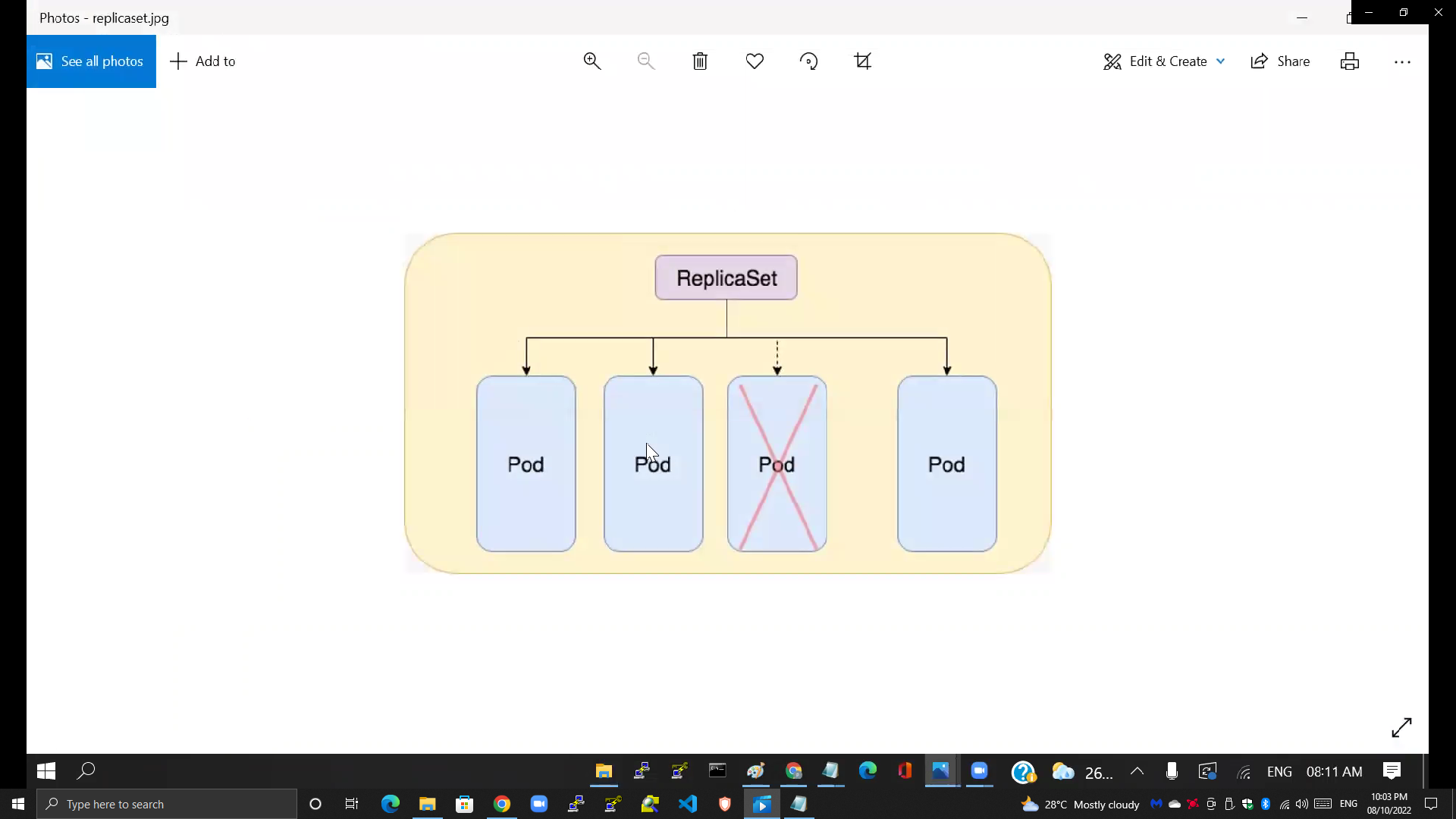
Interview networking

Container to container communication -> network namespace

Container from one pod to another pod from same node -> network bridge using ethernet

Container from one node to another node -> network routing table

Replicatset -> autoscaleing concepts



Kubectl get nodes

Node list

**“Suggestions: more master node” nodes should me edit add**

\* edit your node instance group: kops edit ig --name=suresh.k8s.local nodes-ap-south-1a

kops update cluster

kops update cluster --yes

kops rolling-update cluster

Kubectl get nodes – view nodes list

Vi pod.txt rename ymal file

apiVersion: v1

kind: Pod

metadata:

name: memory-demo-2

spec:

containers:

- name: pod1

image: nginx:1.14.2

ports:

- containerPort: 80

Mv pod.txt pod.yaml

Kubectl apply -f pod.yaml – pod creation

**“Namespace”**

Kubectl get ns

**Default - yaml file, kube-node-lease (This namespace holds** [**Lease**](https://kubernetes.io/docs/reference/kubernetes-api/cluster-resources/lease-v1/) **objects associated with each node.) , kube-public - pod create using intranet in move internet using kube-public, kube-system - all master node compounded(scheduler, proxy, api server, etcd)**

Kubectl get pods

Kubectl get pod -n default

Kubectl get pod -n kube-system —----list master node component

Kubectl create ns createdirectory

Kubectl get ns **(ns -namespace)**

Createdirectory

Vi pod.txt rename ymal file

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: memory-demo-2**

**Namespace: creatediretory enter**

**spec:**

**containers:**

**- name: pod1**

**image: nginx:1.14.2**

**ports:**

**- containerPort: 80**

Kubectl get pods

Deployment

Kubectl apply -f pod.yaml

Kubectl get pods

Kubectl get pod -n createdirectory

Replicaset.yaml file replocaset simply pots

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: replicaset-example

spec:

# modify replicas according to your case

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.114

Kubectl apply -f replicaset.yaml

Kubectl get pods

Kubectl delete pod name of the pod name (fraction of second create new pods)

Kubectl get rs **(rs - Replicaset) “replicaset recreation deployment rollback”**

Deployment

apiVersion: apps/v1

kind: Deployment

metadata:

name: deployment-example

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: httpdpod

image: nginx:1.114 **(ecr image name)**

# ports:

# - containerPort: 80

Kubectl apply -f deployment.yaml

Kubectl get pods

**Inside pods**

kubectl exec -it pod name /bin/bash

cd /usr/local/apche2#

Ls

Cd htdocs cat index.html

Uname -a

Apt-get install curl -y

Curl http//localhost: 80

**Interview question docker image scanner “anchore”**

**Deployment**

The rolling deployment is the standard default deployment to kubernetes, it works

By slowly, one by one, replacing pods of the previous version of your application with

Pods of the new version without any cluster downtime

Kubectl get deployment

Kubectl get all **interview question**

Kubectl describe pods name id completed statements

Kubectl get pods -o wide **(deployment list)**

Daemonset.yaml **daemonset simply nodes**

apiVersion: apps/v1

kind: DaemonSet

metadata:

name: mynode-exporter

labels:

app: mynode-exporter

spec:

selector:

matchLabels:

app: mynode-exporter

template:

metadata:

labels:

name: mynode-exporter

spec:

containers:

- name: fluentd-elasticsearch

image: quay.io/fluentd\_elasticsearch/fluentd:v2.5.2

Ports:

- containerport : 9100

hostPort: 9100

Protocol: TCP

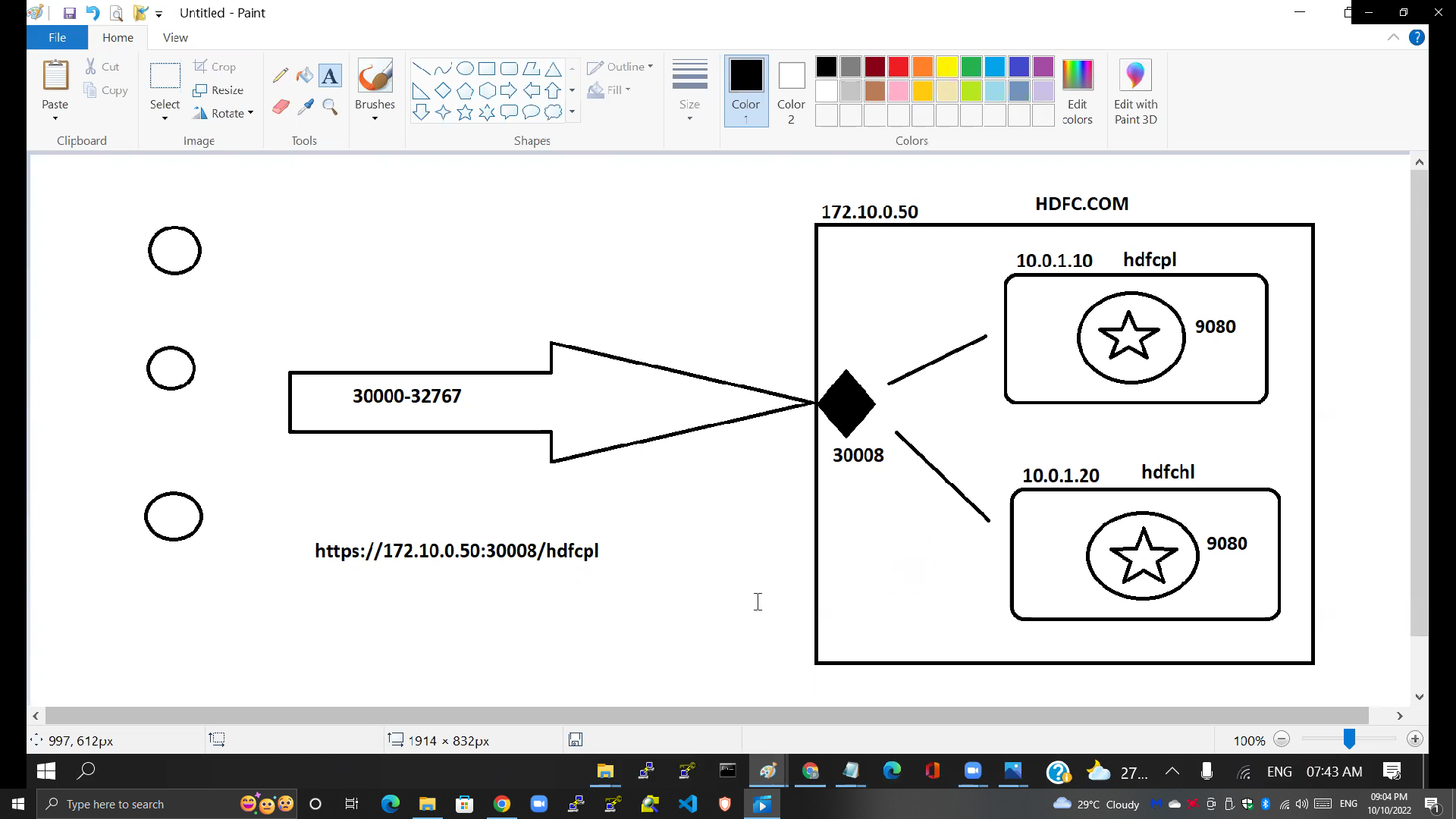
Kubectl get pods

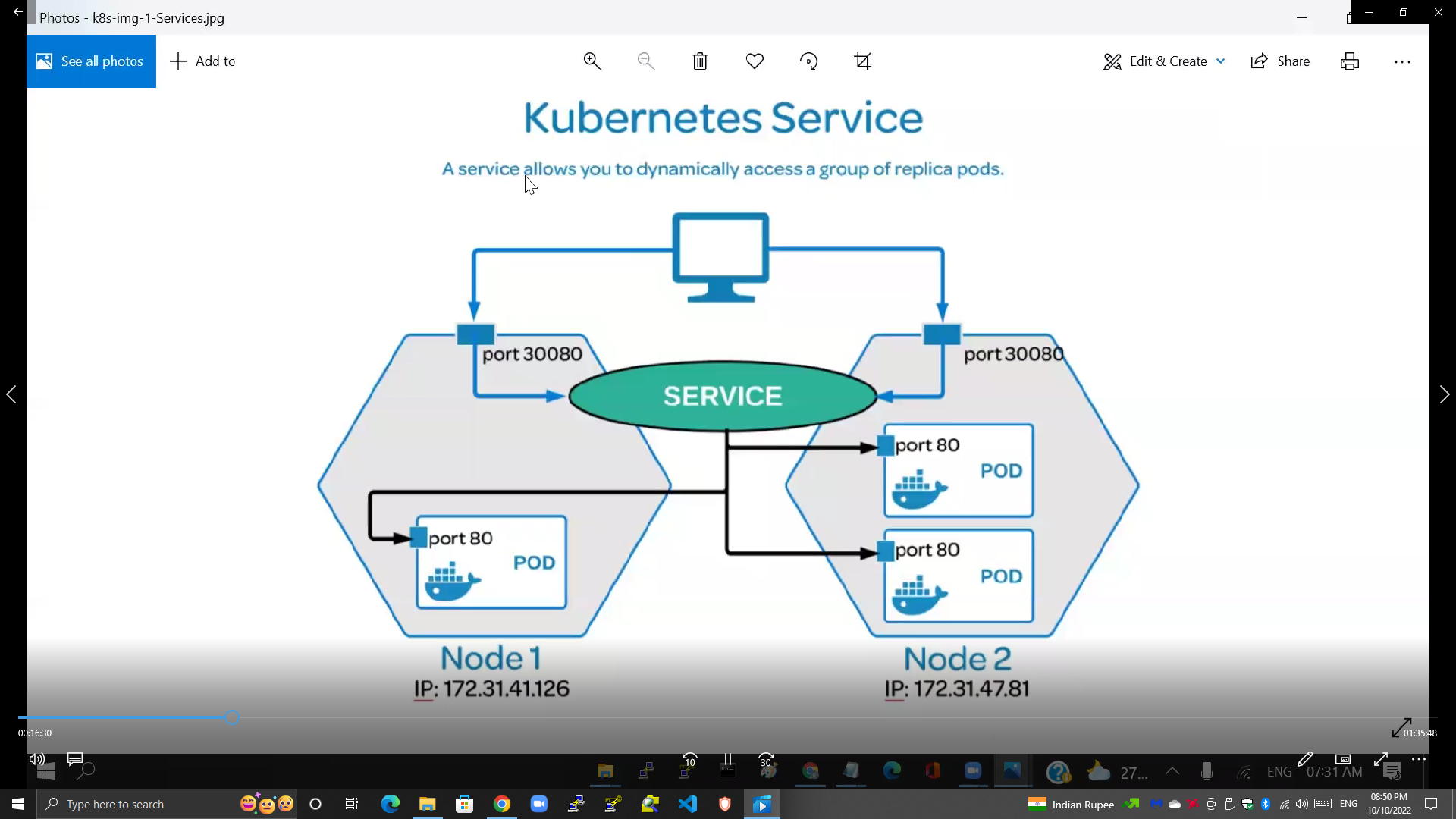
=================================================================

DAy 5

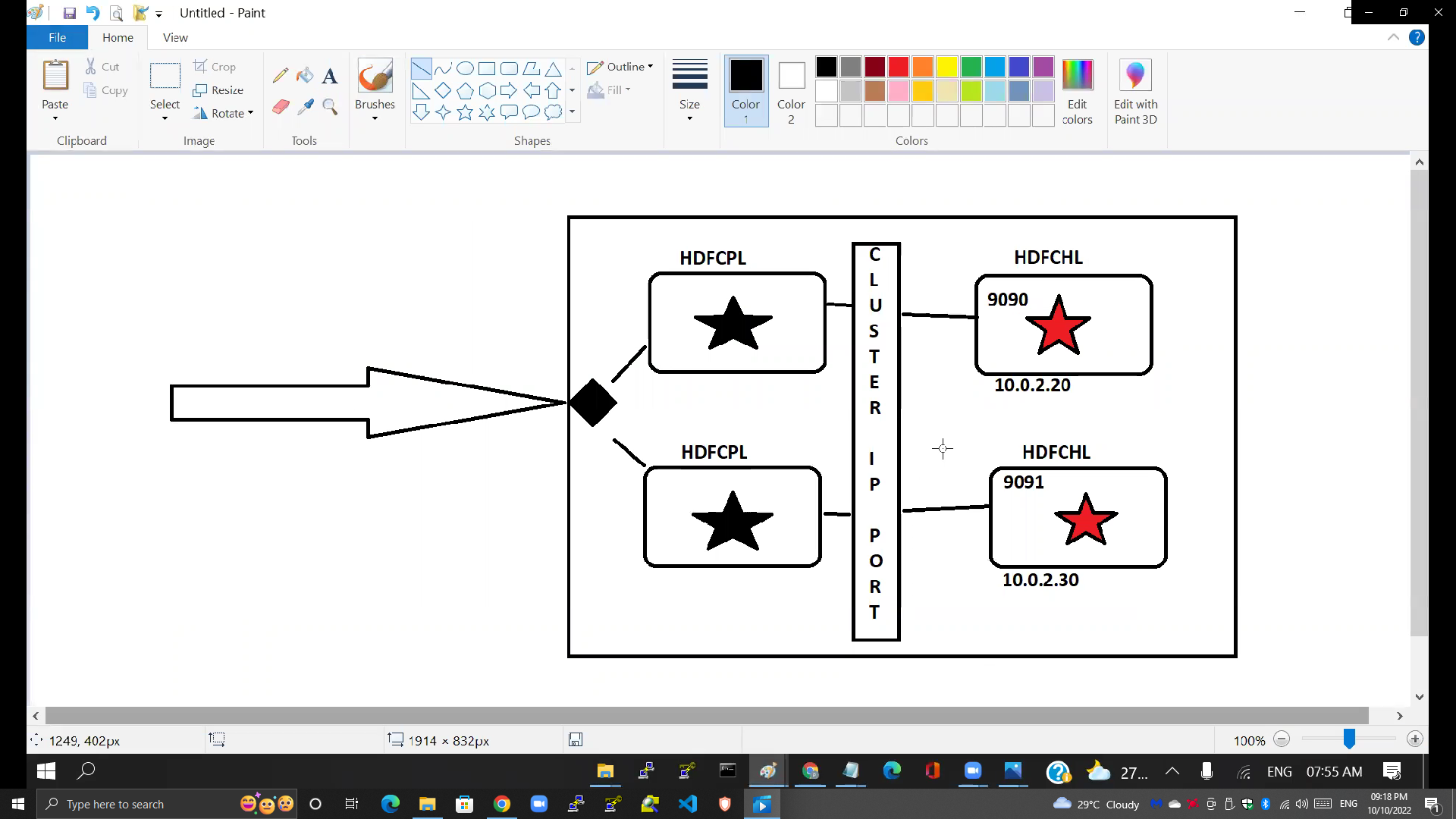
How much service in K8n

**Nodeport ip —>** external traffic efficient manage compounds





**Cluster ip —->** internal communication search app order app communication

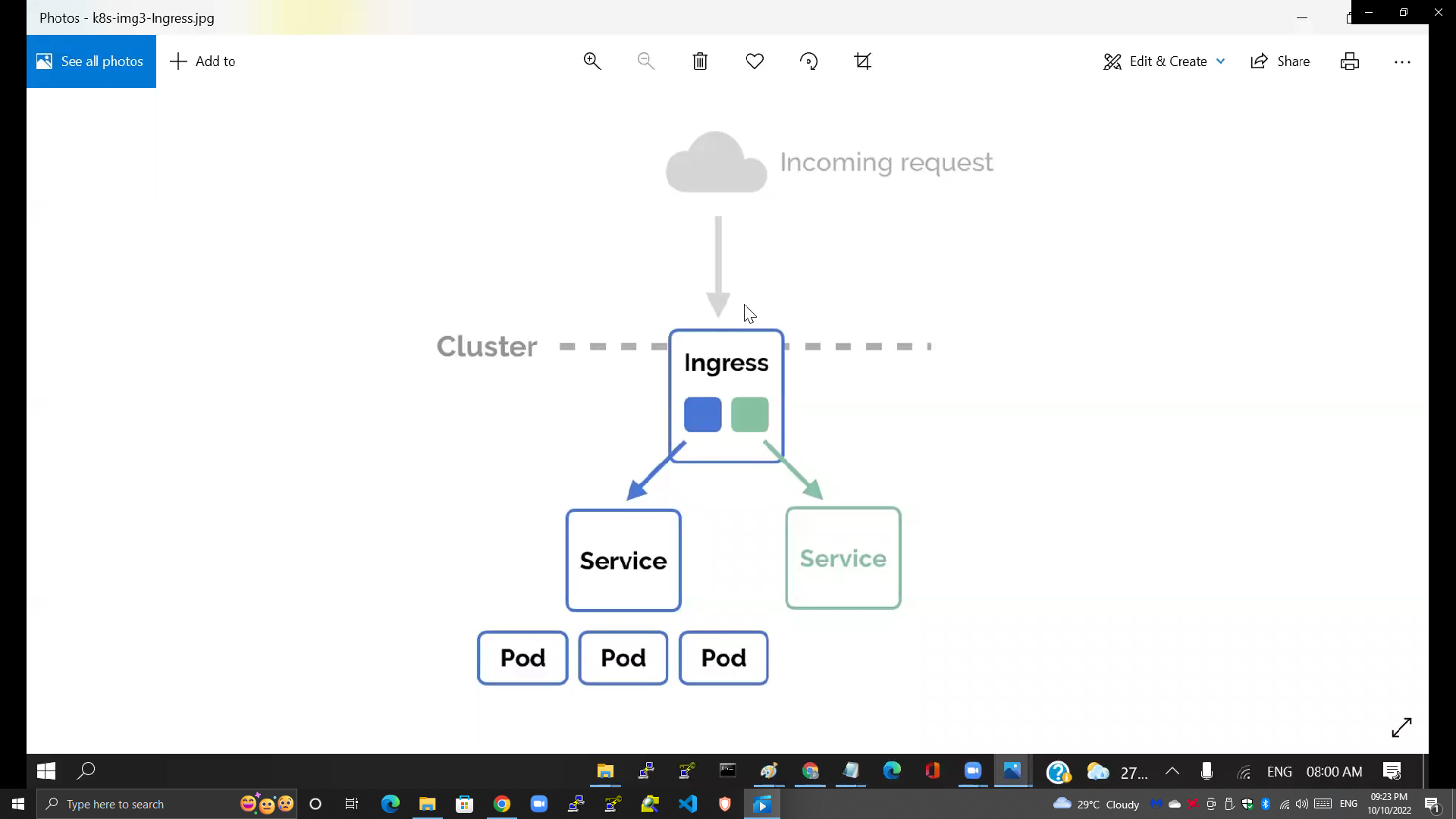


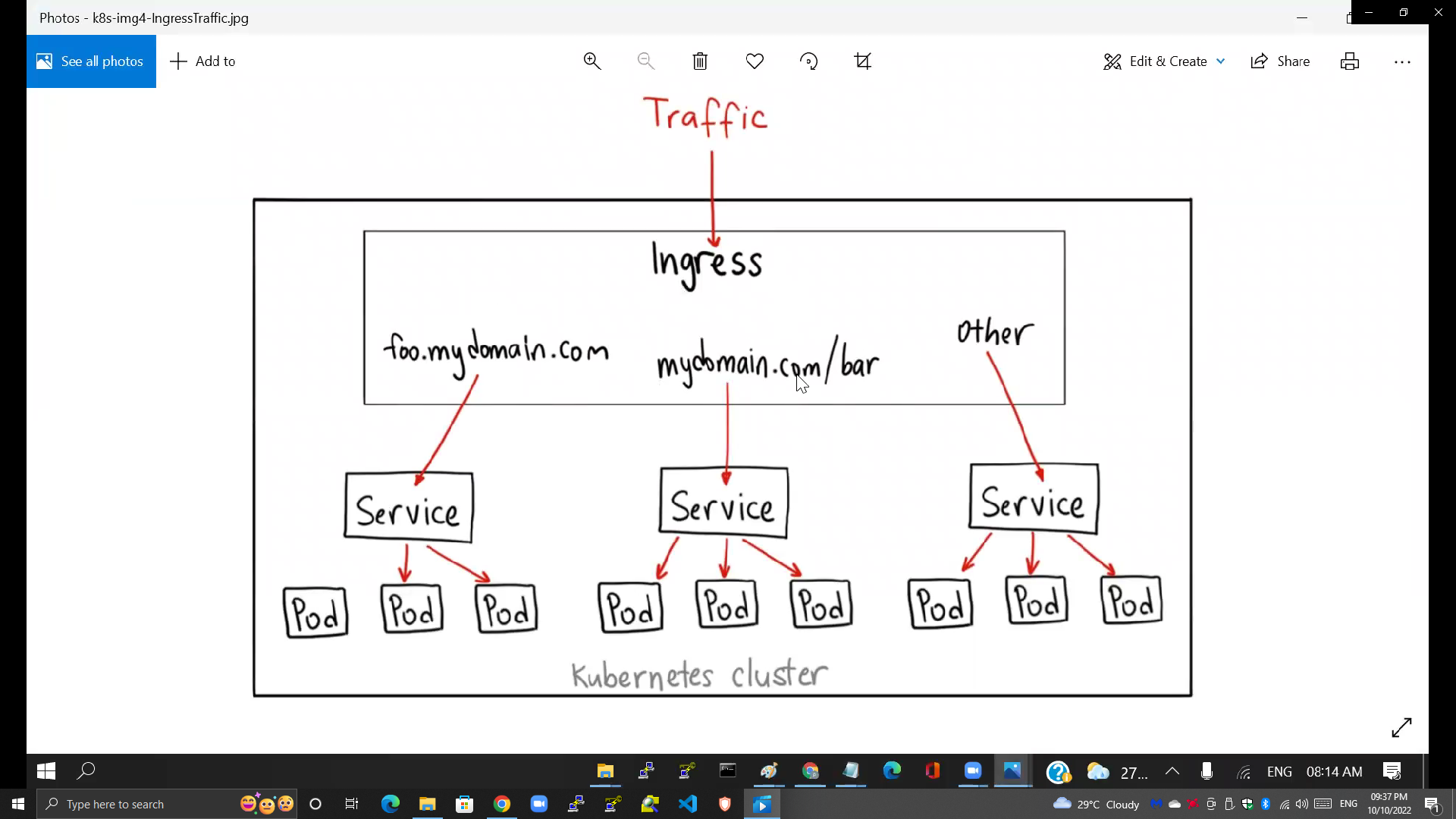
**Load balancer —>** cloud provider cost

**Now using**

**Ingress controller —-->** one of load balancer inbuilt load balancer in kubernetes cheaper

Search app zomato





**type ingress best production**

**Nginix ingress controller using currently**

=================================================================

**ConfigMap**

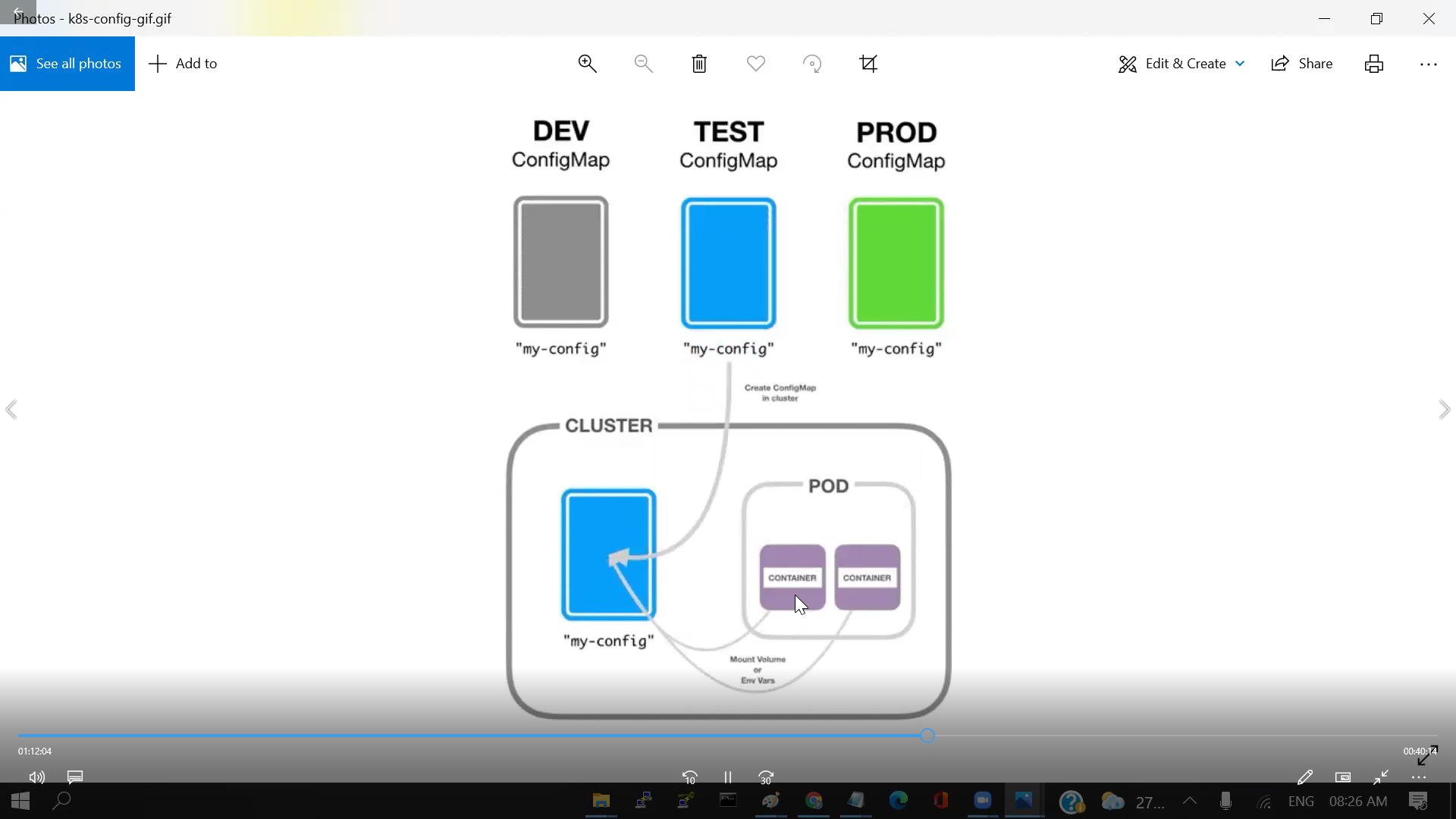
{Ear -> war+jar

Propertyfiles

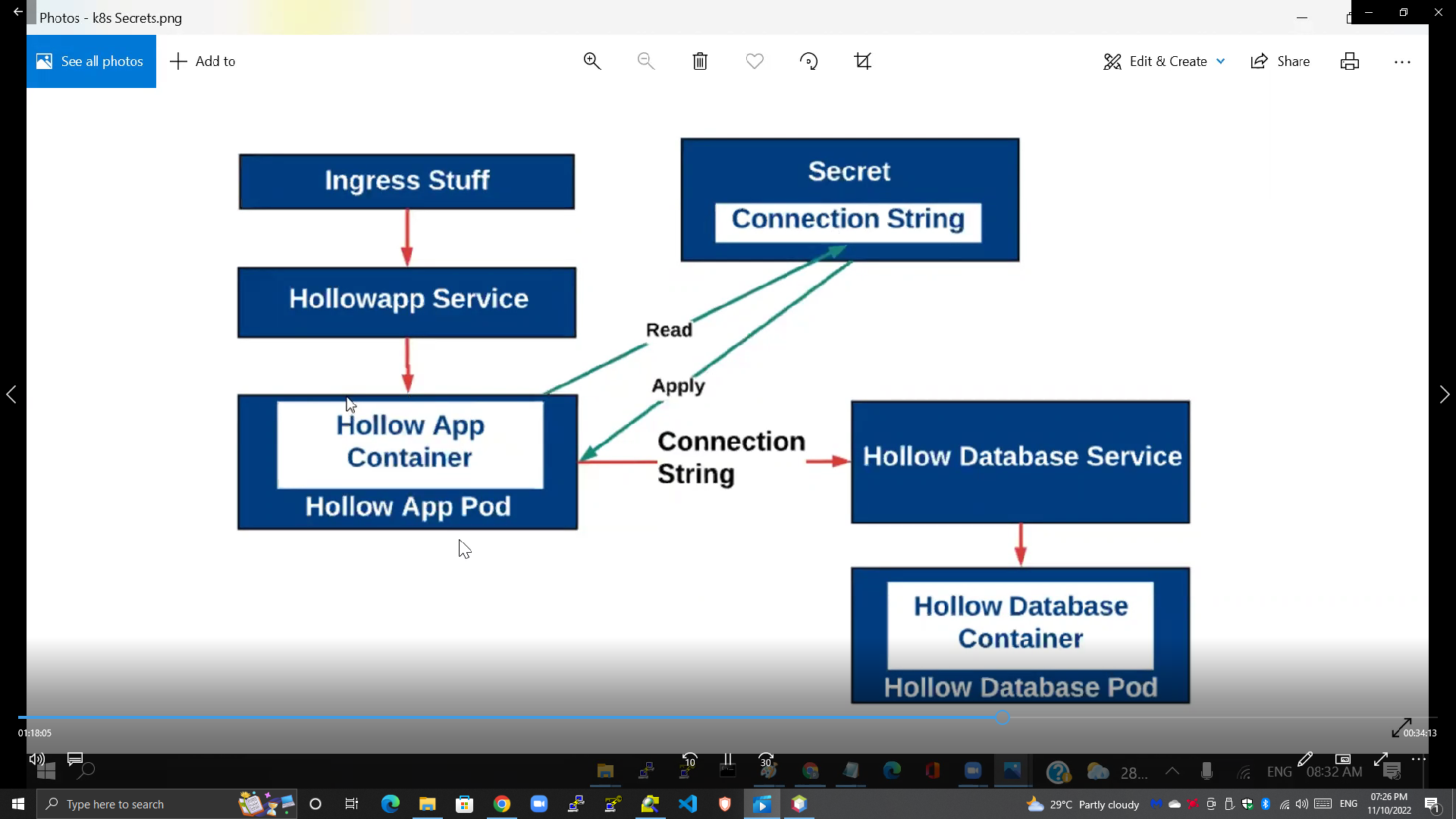
Xml files

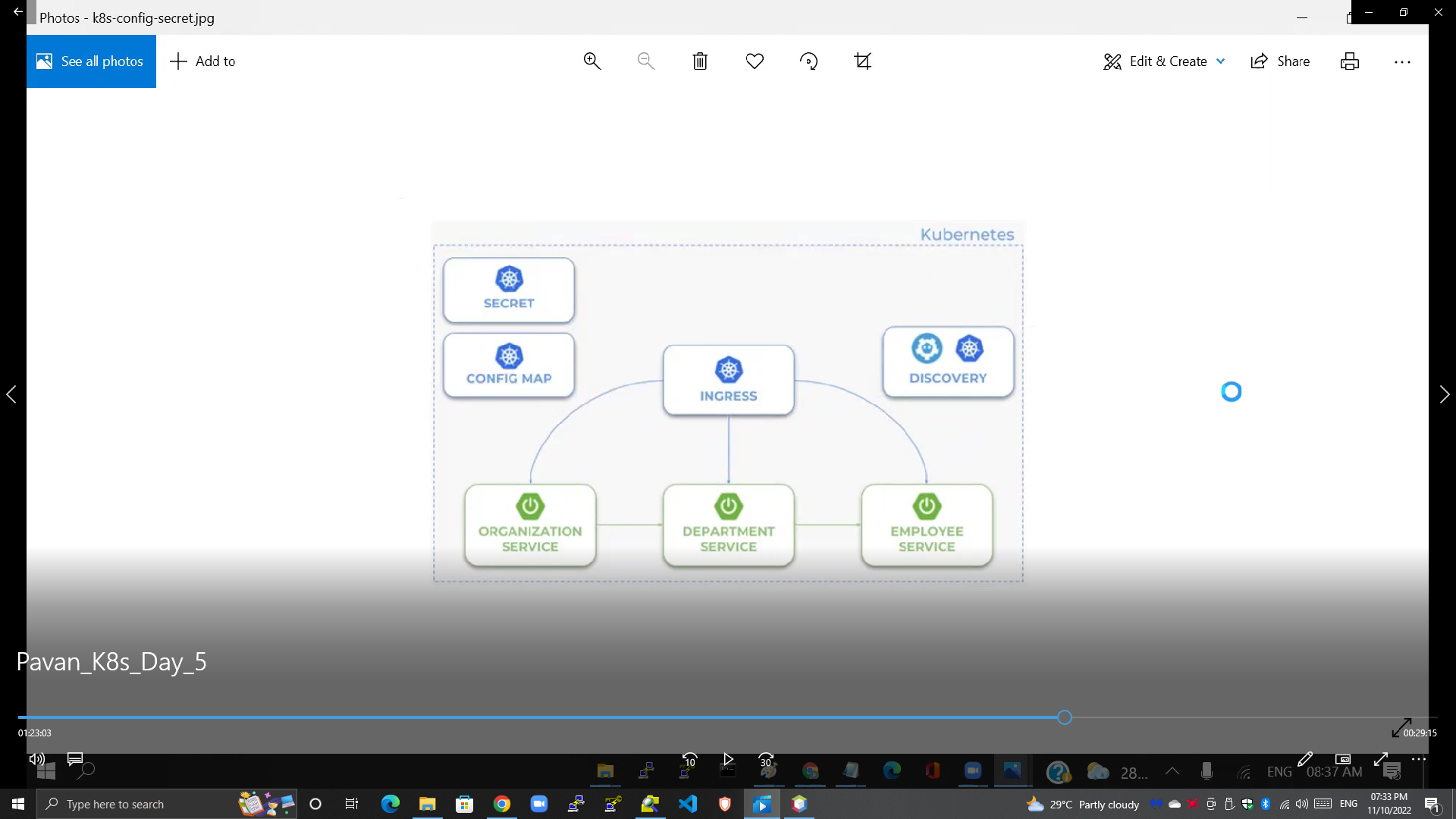
Db files

Jdbc} old



**Secrets**





Kubernetes service discovery is **an abstraction that allows an application running on a set of Pods to be exposed as a network service**. This enables a set of Pods to run using a single DNS name, and allows Kubernetes load balancing across them all.

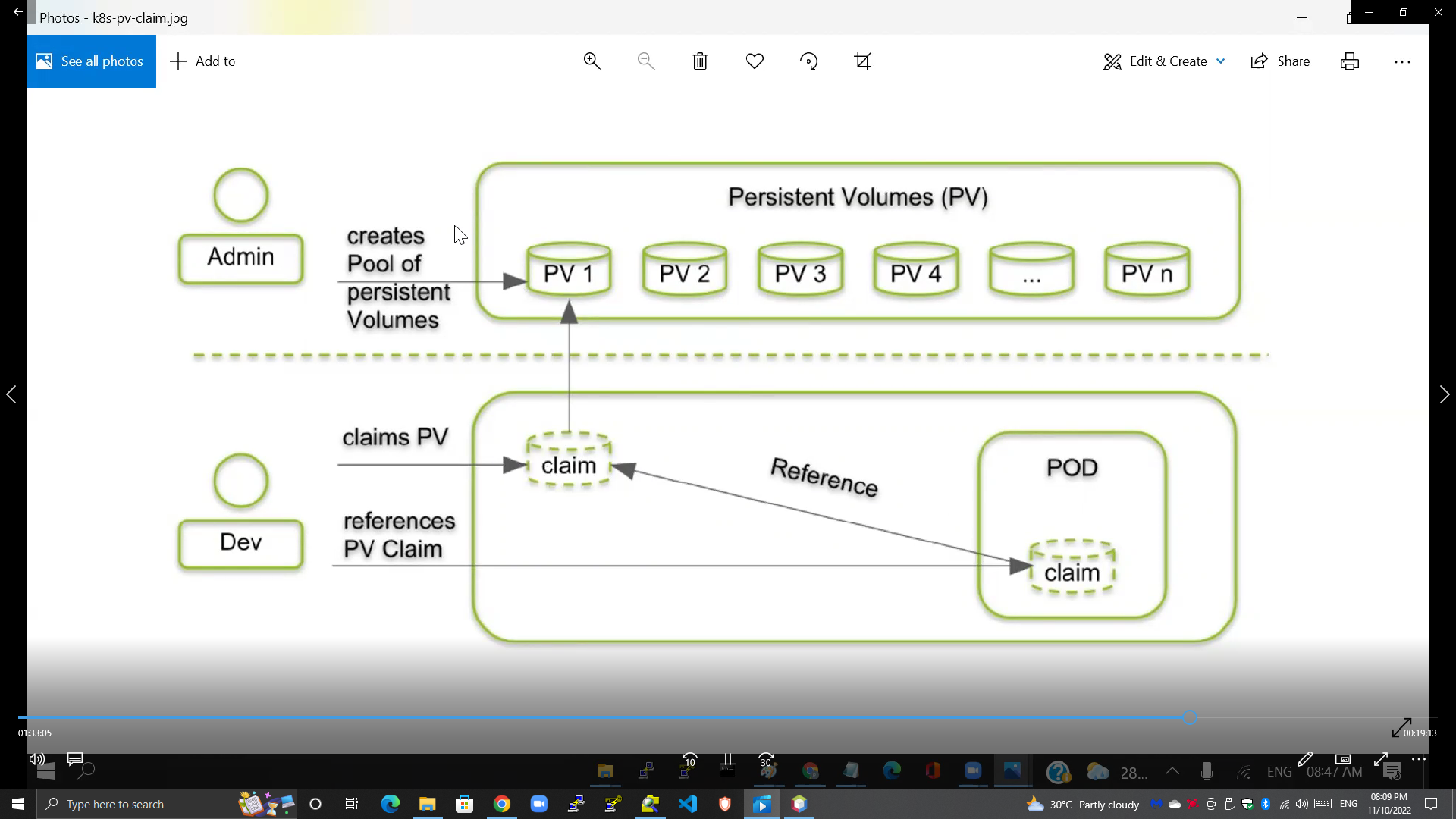
DNs sub domain cluspeer name to pass the network

**Interview**

**Secret vs configmap in kuberneted**

**Secrets store data in base64 (encrypted) format meanwhile configmaps store data in a plain text**

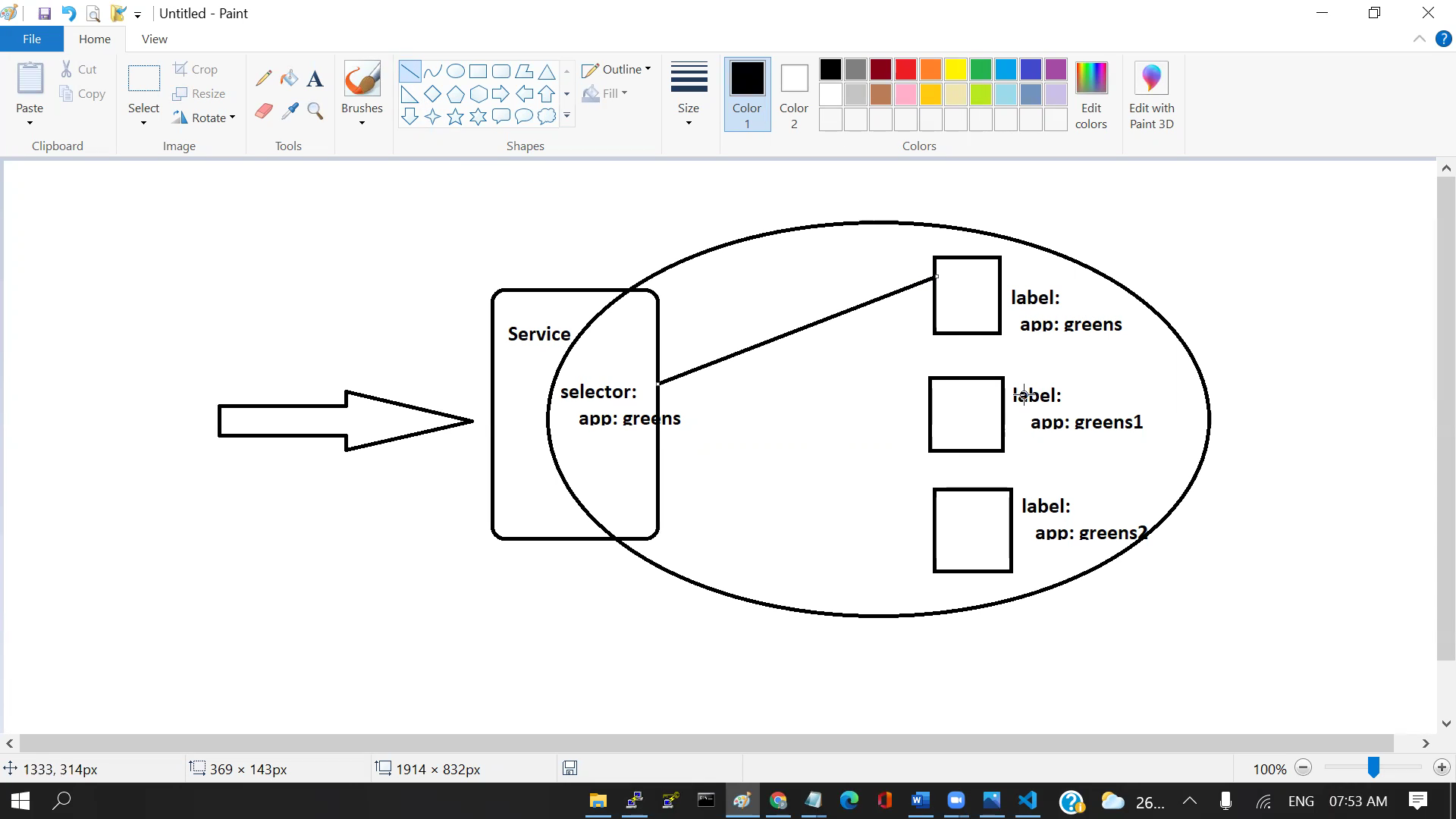
**Volume (pv) claim store volume data store crase the pv1 it will recover it volume to share**

****

**Day 6**

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.22/>

Nodeport service.yaml



apiVersion: v1

kind: Service

metadata:

name: nodeport-svc

spec:

selector:

app: nginx

ports:

- protocol: http

port: 80

targetPort: 80

nodePort: 30008

type: Nodeport

**Another yaml file to communicate service to label to pair**

**Interview question node traffic based upon selector and label traffic pods rooted**

**Nodeport—>30000-32767—>30008 external**

apiVersion: v1

kind: Service

metadata:

name: nodeport-svc

spec:

**selector:**

**app: webapps // value pair key**

ports:

- name: http

port: 80

nodePort: 30008

type: Nodeport

* - -

apiVersion: v1

kind: Pod

metadata:

name: nodeport-svc

**labels:**

**app: webapps**

spec:

containers:

- name: pod1

Image: nginx

ports:

- containerPort: 80

mv nodeport.txt nodeport.yaml

**View clusterip**

Kubectl get svc

kubectl apply -f nodeport.yaml

kubectl get pods

kubectl get svc

**Interview How to find pods which node command -** kubectl get pods -o wide

=================================================================

**Clusterip.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

selector:

matchLabels:

app: nginx-app

replicas: 1

template:

metadata:

labels:

app: nginx-app

spec:

containers:

- name: nginx

image: nginx:1.13.12

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: clusterip-svc

spec:

type: ClusterIP

ports:

- port: 80

selector:

app: nginx-app

=================================================================

loadbalaner.yaml

apiVersion: v1

kind: Service

metadata:

name: loadbalancer-svc

spec:

type: LoadBalancer

selector:

app: hello

ports:

- name: http

protocol: TCP

port: 80

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: loadbalancer-pod

spec:

replicas: 1

selector:

matchLabels:

app: hello

template:

metadata:

labels:

app: hello

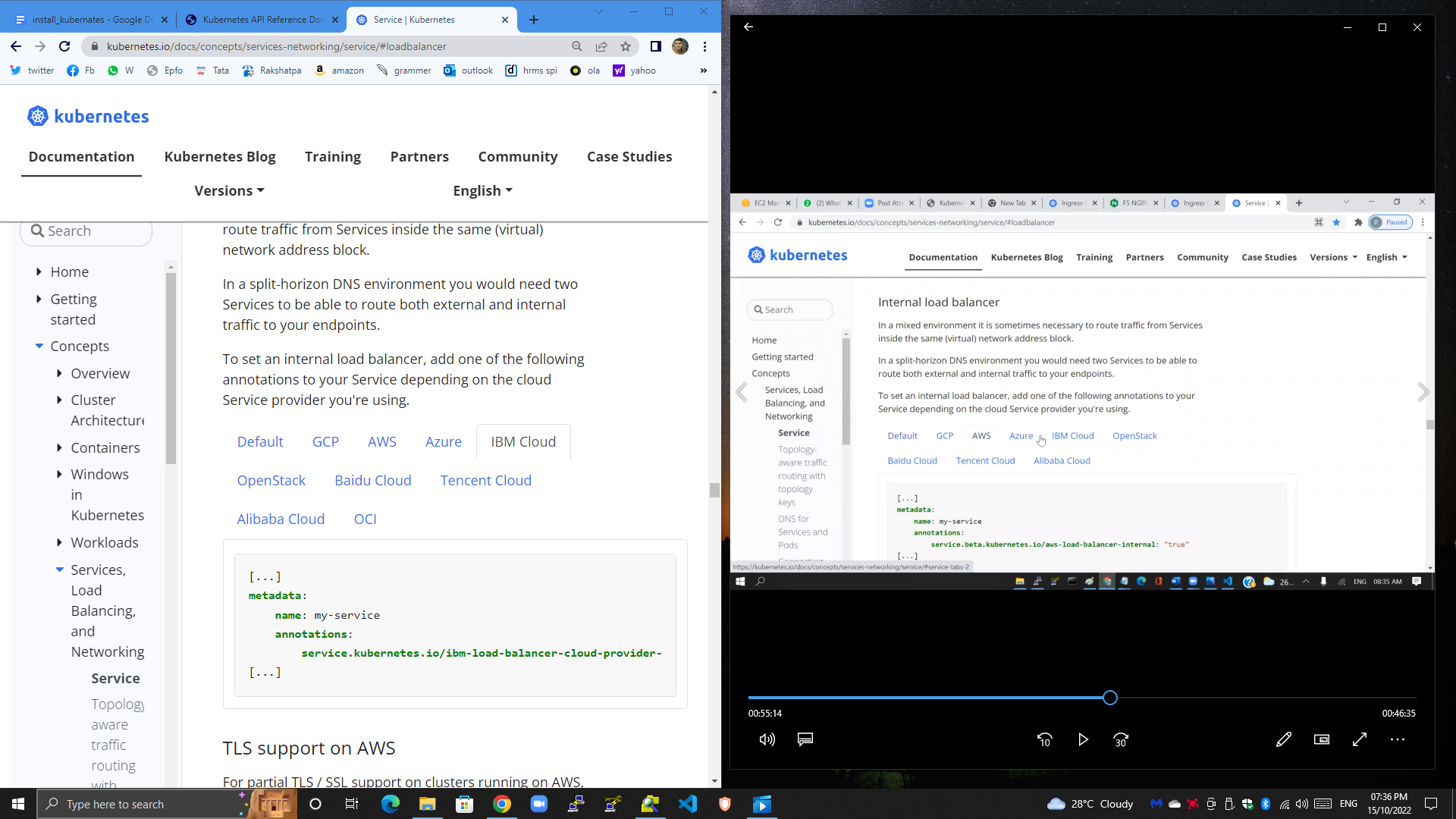
spec:

containers:

- name: hello

image: httpd

kubernetes.io/docs/concepts/services-networking/service/#loadbalancer



=================================================================

Ingress controller

Ingress.yaml -git hub code link- https://github.com/nginxinc/kubernetes-ingress/

# Refer from Official websites nginxinc

# https://docs.nginx.com/nginx-ingress-controller/installation/installation-with-manifests/

---

apiVersion: v1

kind: Namespace

metadata:

name: nginx-ingress

labels:

app.kubernetes.io/customer: test

---

apiVersion: v1

kind: ServiceAccount

metadata:

name: nginx-ingress

namespace: nginx-ingress

labels:

app.kubernetes.io/customer: test

---

kind: ClusterRole

apiVersion: rbac.authorization.k8s.io/v1beta1

metadata:

name: nginx-ingress

labels:

app.kubernetes.io/customer: test

rules:

- apiGroups:

- ""

resources:

- services

- endpoints

verbs:

- get

- list

- watch

- apiGroups:

- ""

resources:

- secrets

verbs:

- get

- list

- watch

- apiGroups:

- ""

resources:

- configmaps

verbs:

- get

- list

- watch

- update

- create

- apiGroups:

- ""

resources:

- pods

verbs:

- list

- watch

- apiGroups:

- ""

resources:

- events

verbs:

- create

- patch

- apiGroups:

- extensions

resources:

- ingresses

verbs:

- list

- watch

- get

- apiGroups:

- "extensions"

resources:

- ingresses/status

verbs:

- update

- apiGroups:

- k8s.nginx.org

resources:

- virtualservers

- virtualserverroutes

verbs:

- list

- watch

- get

---

kind: ClusterRoleBinding

apiVersion: rbac.authorization.k8s.io/v1beta1

metadata:

name: nginx-ingress

labels:

app.kubernetes.io/customer: test

subjects:

- kind: ServiceAccount

name: nginx-ingress

namespace: nginx-ingress

roleRef:

kind: ClusterRole

name: nginx-ingress

apiGroup: rbac.authorization.k8s.io

---

apiVersion: v1

kind: Secret

metadata:

name: default-server-secret

namespace: nginx-ingress

labels:

app.kubernetes.io/customer: test

type: Opaque

data:

tls.crt: 

tls.key: 

---

kind: ConfigMap

apiVersion: v1

metadata:

name: nginx-config

namespace: nginx-ingress

labels:

app.kubernetes.io/customer: test

data:

server-tokens: "false"

server-snippets: |

add\_header X-Frame-Options "SAMEORIGIN";

add\_header X-XSS-Protection "1; mode=block";

add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains always;";

---

apiVersion: apiextensions.k8s.io/v1beta1

kind: CustomResourceDefinition

metadata:

name: virtualservers.k8s.nginx.org

labels:

app.kubernetes.io/customer: test

spec:

group: k8s.nginx.org

versions:

- name: v1

served: true

storage: true

scope: Namespaced

names:

plural: virtualservers

singular: virtualserver

kind: VirtualServer

shortNames:

- vs

---

apiVersion: apiextensions.k8s.io/v1beta1

kind: CustomResourceDefinition

metadata:

name: virtualserverroutes.k8s.nginx.org

labels:

app.kubernetes.io/customer: test

spec:

group: k8s.nginx.org

versions:

- name: v1

served: true

storage: true

scope: Namespaced

names:

plural: virtualserverroutes

singular: virtualserverroute

kind: VirtualServerRoute

shortNames:

- vsr

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-ingress

namespace: nginx-ingress

labels:

app.kubernetes.io/customer: test

spec:

replicas: 1

selector:

matchLabels:

app.kubernetes.io/name: ingressController

app.kubernetes.io/part-of: requestHandler

template:

metadata:

labels:

app.kubernetes.io/name: ingressController

app.kubernetes.io/part-of: requestHandler

app.kubernetes.io/customer: test

#annotations:

#prometheus.io/scrape: "true"

#prometheus.io/port: "9113"

spec:

serviceAccountName: nginx-ingress

containers:

- image: nginx/nginx-ingress:latest # 1.6.2

name: nginx-ingress

ports:

- name: http

containerPort: 80

- name: https

containerPort: 443

#- name: prometheus

#containerPort: 9113

resources:

limits:

cpu: 210m

memory: 210Mi

requests:

cpu: 200m

memory: 200Mi

securityContext:

allowPrivilegeEscalation: true

runAsUser: 101 # nginx

capabilities:

drop:

- ALL

add:

- NET\_BIND\_SERVICE

env:

- name: POD\_NAMESPACE

valueFrom:

fieldRef:

fieldPath: metadata.namespace

- name: POD\_NAME

valueFrom:

fieldRef:

fieldPath: metadata.name

args:

- -nginx-configmaps=$(POD\_NAMESPACE)/nginx-config

- -default-server-tls-secret=$(POD\_NAMESPACE)/default-server-secret

#- -v=3 # Enables extensive logging. Useful for troubleshooting.

#- -report-ingress-status

#- -external-service=nginx-ingress

#- -enable-leader-election

#- -enable-prometheus-metrics

---

apiVersion: v1

kind: Service

metadata:

name: nginx-ingress

namespace: nginx-ingress

labels:

app.kubernetes.io/customer: test

spec:

externalTrafficPolicy: Local

type: LoadBalancer

ports:

- port: 80

targetPort: 80

protocol: TCP

name: http

- port: 443

targetPort: 443

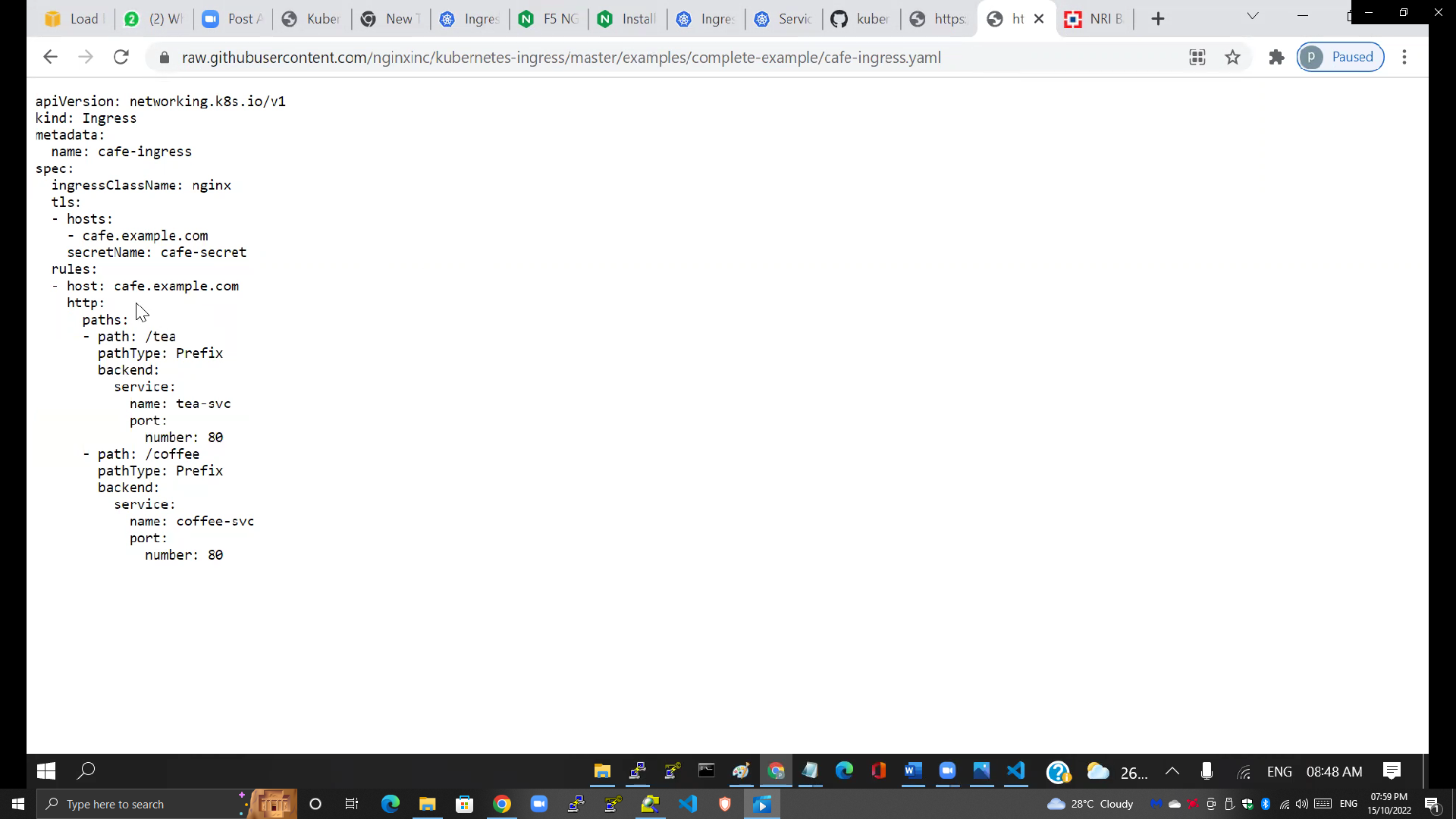
protocol: TCP

name: https

selector:

app.kubernetes.io/name: ingressController

app.kubernetes.io/part-of: requestHandler



**raw.gihubusercontent.com/nginxinc/kubernetes-ingress/master/examples/complete-example/cafe-ingress.yaml**

=================================================================

Day 7

Ev, configmap, secrets

Volume, persistent volume, persistent volume claim

Hpa, node selector, node affinity

**Env.yaml**

apiVersion: v1

kind: Pod

metadata:

name: envar-demo

labels:

**purpose: demonstrate-envvars**

spec:

containers:

- name: envar-demo-container

image: nginx

ports:

- containerPort: 8080

env:

- name: DEMO\_GREETINGS

value: "Hello from the Environment"

**kubectl get pods -l purpose=demonstrate-envvars -> to find the purpose running as pods**

=================================================================

configmap.yaml **to check all detail external and internal pods application port number, username, ip, endpoints**

apiVersion: v1

kind: ConfigMap

metadata:

name: test-config1

data:

name: DEMO\_GREETINGS\_SESSION

value: "Hello from the DEMO GREETINGS environment"

—-

pod1.yaml

apiVersion: v1

kind: Pod

metadata:

name: envar-demo1

labels:

purpose: demonstrate-envars1

spec:

containers:

- name: envar-demo-container

image: nginx

ports:

- containerPort: 8081

envFrom:

- configMapRef:

name: test-config1

======================================================

**Secret.yaml file**

**Grep encrypted value -> echo -n ‘root’ | base64**

apiVersion: v1

kind: Secret

metadata:

name: **test-secret1**

data:

DB\_HOST: TWl0aHJhbg==

DB\_USER: QWRtaW4xMjM=

Pod2.yaml

apiVersion: v1

kind: Pod

metadata:

name: envar-demo2

labels:

purpose: demonstrate-envars2

spec:

containers:

- name: envar-demo-container

image: nginx

ports:

- containerPort: 8082

envFrom:

- secretRef:

name: **test-secret1**

**=======================================================**

Volume just a storage type dir, file, dirCreate, FileorCreate

volume.yaml

apiVersion: v1

kind: Pod

metadata:

name: test-pod

spec:

containers:

- name: test-pod

image: nginx

volumeMounts:

- mountPath: /test-pd

name: test-volume

volumes:

**- name: test-volume**

**hostPath:**

**path: /data**

**type: Directory**

echo “text” > 1.txt → data enter

Advance **Node Selector** one node same node to create pods to declare the data sink

=================================================================

persistent volume

persistentvolume.yaml

apiVersion: v1

kind: PersistentVolume

metadata:

name: test-volume

spec:

accessModes:

- ReadWriteOnce

capacity:

storage: 1Gi

hostPath:

path: /tmp/data

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: **testclaim**

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 500Mi

---

apiVersion: v1

kind: Pod

metadata:

name: mypod

spec:

containers:

- name: testpod

image: nginx

volumeMounts:

- mountPath: "/var/www/html"

name: mypd

volumes:

- name: mypd

persistentVolumeClaim:

claimName: **testclaim**

**Interview question kubernetes access modes**

**Readwriteonce**, Readonlymany, readwritemany, **readwriteoncepod**

**Configmap and secrets\**

**Persistent volume vs persistent volume claim**

**Autoscaling horizontal and vertical**

**kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale-walkthrough/**

**Node selector– > label link official documentation**

**and node affinity → node affinity official doc**