

# Cloud Computing Lab - CAT-2

## Team members:

Devasena K (1934006)

Lekha sree J(1934022)

Sarika M(1934040)

**Goal :** To dockerize a web application and configure, deploy in kubernetes Cluster

**Application :** Health insurance cross sell predictor

Web application to predict whether health insurance owner will be interested in vehicle insurance using **Machine Learning Techniques**

**Web app is deployed over Heroku :** <https://ml-cross-sell-prediction.herokuapp.com/>

## Tools/ plug-ins used :

- VScode
- Jupyter notebook
- Flask
- Docker desktop
- Azure - kubernetes deployment
- Git bash

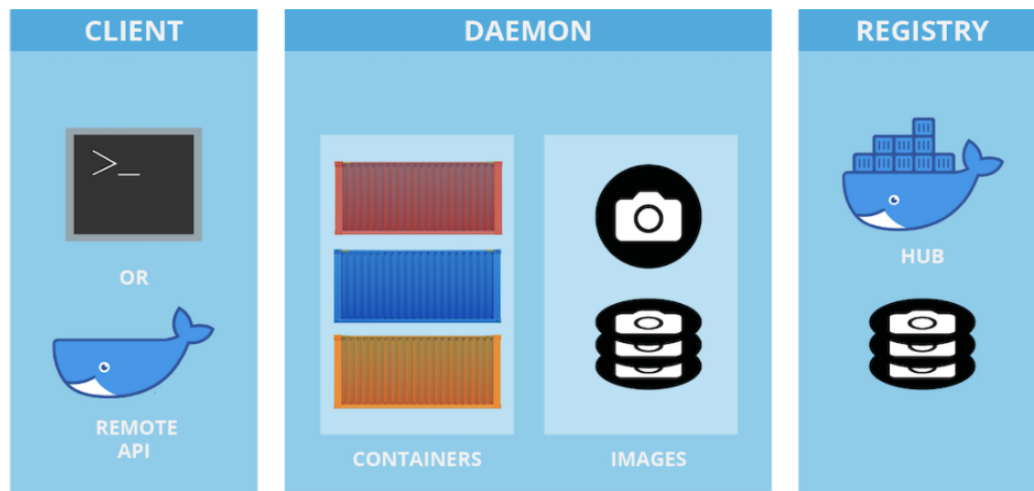
## Docker

Docker is a tool that promises to easily encapsulate the process of creating a distributable artifact for any application, deploying it at scale into any environment, and streamlining the workflow and responsiveness of agile software organizations.

**docker is need for following purposes,**

- Compatibility
- Dependency
- Long Setup Time
- Different Environment

## Docker Architecture



Some advantages of docker are ,

- Packaging software
- Bundling application software and required OS filesystems together in a single standardized image format
- Using packaged artifacts to test and deliver the exact same artifact to all systems in all environments
- Abstracting software applications from the hardware without sacrificing resources

### Containers:

A container is the same idea as a physical container think of it like a box with an application in it. Inside the box, the application seems to have its own machine name and IP address, and it also has its own disk drive. It uses the same OS Kernel

### Container Vs VM

Containers	Virtual Machines
Fast boot time	Slow boot time
Uses execution engine	Uses hypervisor
Less resource utilization	More resource utilization
Less disk space	More disk space
Easy deployment	Lengthy deployments

**Docker images :**

A Docker image is an immutable (unchangeable) file that contains the source code, libraries, dependencies, tools, and other files needed for an application to run.

**Docker container :**

A Docker container is a virtualized run-time environment where users can isolate applications from the underlying system.

**Dockerfile of our web application :**

```
FROM ubuntu

RUN apt-get update
RUN apt-get install -y python3 python3-pip
COPY requirements.txt requirements.txt
RUN pip3 install -r requirements.txt
RUN mkdir app

WORKDIR /opt/app
COPY . /opt/app
ENTRYPOINT FLASK_APP=/opt/app/app.py flask run --host 0.0.0.0
```

**Docker features:****Image Features**

- Running the app in container
- Writing the dockerfile
- Building the image and push it to Dockerhub

**Docker commands used :**

Building image :

docker build -t cross .

Check whether our image is running or not :

docker run -t cross

Fetch log for our image(cross):

docker logs cross

Check which images are running currently:

docker ps

Check which images are ran locally :  
docker ps -a

List docker images:  
Docker images

```
Lenono@DESKTOP-MQF4H56 MINGW64 /d/SEM 5/assessment_project (main)
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ml_1	latest	d3715fa2de92	4 days ago	1.92GB
sarika088/ml_w	latest	d3715fa2de92	4 days ago	1.92GB
cross	latest	5904002ae03e	7 days ago	997MB
sarika088/cross	latest	5904002ae03e	7 days ago	997MB
sarika088/myimage1	latest	f6c426698fa6	2 weeks ago	418MB
sarika088/myimage	latest	0f31c96423e4	2 weeks ago	418MB
cal	latest	f44f1d87a89a	2 weeks ago	418MB
sarika088/calcap	latest	f44f1d87a89a	2 weeks ago	418MB
<none>	<none>	b1c57212f4c1	2 weeks ago	418MB
sarika088/sleepy1	latest	9926fa3bfa2b	3 weeks ago	72.8MB
sarika088/image3	latest	fd2c6b1b1bef	3 weeks ago	72.8MB
sarika088/sleepy2	latest	7f5bcff4fe6c	3 weeks ago	72.8MB
ubuntu	latest	d13c942271d6	3 weeks ago	72.8MB
sarika088/sleepy3	latest	a6620c60e82a	3 weeks ago	72.8MB
sarika088/image2	latest	bed101333cf5	3 weeks ago	72.8MB
sarika088/image1	latest	2f5de477e6bb	3 weeks ago	72.8MB
nginx	latest	605c77e624dd	4 weeks ago	141MB
mysql	latest	3218b38490ce	5 weeks ago	516MB
alpine	latest	c059bfaa849c	2 months ago	5.59MB
docker/desktop-kubernetes	kubernetes-v1.22.4-cni-v0.8.5-critools-v1.17.0-debian	493a106d3678	2 months ago	294MB

Image caching :

sarika088/cross	<span>IN USE</span>	latest	5904002ae03e	8 days ago	997.28 MB	<span>⋮</span> <span>RUN ▶</span>
-----------------	---------------------	--------	--------------	------------	-----------	-----------------------------------

Layer caching :

Containers / Apps

Images

Volumes

Dev Environments PREVIEW

🐳

🔍

📁

🔧

hardcore\_robinson sarika088/cross

EXITED (255)

LOGS

INSPECT

STATS

🔄

⏪

⏩

🔄

🛑

Environment

PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

📄

Port

5000/tcp

0.0.0.0:8000

📄

Login to Docker Hub :  
docker login

Push dockerized image to DockerHub:

```
docker push -it sarika088/cross
```

Pull image from Docker image from Docker Hub:

```
docker pull sarika088/cross:latest
```

Running the pulled image locally:

```
docker run -t sarika088/cross
```

Delete a image :

```
docker rmi cross-dummy
```

Docker limitations:

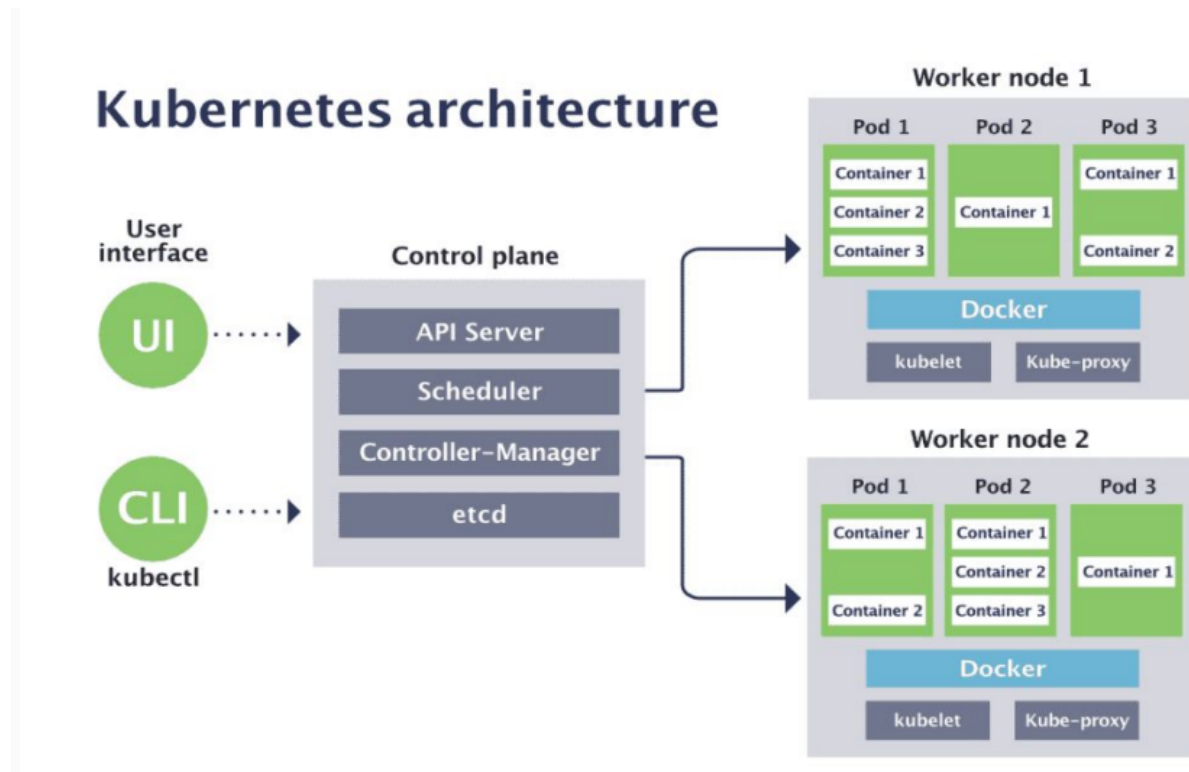
- Enterprise virtualization platform
- Cloud platform
- Configuration management
- Deployment framework
- Workload management tool

Docker hub link : <https://hub.docker.com/repository/docker/sarika088/cross>

## Kubernetes:

Kubernetes is a system that manages containers (containerized applications) where a container could be explained as a lightweight virtual machine. To build an application you need to build a bunch of containers and then use Kubernetes to manage those containers.

## Architecture :



## Steps for Creating Kubernetes Cluster in Azure :

- Create kubernetes cluster
- Managing Resource Groups and Deploying Kubernetes Cluster
- Communicating with Kubernetes Cluster using Cloud Shell
- Hosting an Application on Kubernetes Cluster
- Running Application on Kubernetes Cluster

## Concepts used:

### Namespace:

In Kubernetes, *namespaces* provide a mechanism for isolating groups of resources within a single cluster. Names of resources need to be unique within a namespace, but not across namespaces. Namespace-based scoping is applicable only

for namespaced objects (e.g. *Deployments, Services, etc*) and not for cluster-wide objects (e.g. *StorageClass, Nodes, PersistentVolumes, etc*).

### **Pods :**

A pod is the smallest deployable unit that can be managed by Kubernetes. A pod is a logical group of one or more containers that share the same IP address and port space. The main purpose of a pod is to support co-located processes, such as an application server and its local cache

### **Nodes :**

A node is a machine either physical or virtual machine on which Kubernetes is installed. A node is a worker machine and this is where containers inside the pods will be launched by Kubernetes

### **Replication factor:**

Responsible for maintaining the correct number of pods for every replication controller object in the system

### **Deployment :**

- Deployment is the easiest and most used resource for deploying your application. It is a Kubernetes controller that matches the current state of your cluster to the desired state mentioned in the Deployment manifest.
- Deployments are usually used for stateless applications. However, you can save the state of deployment by attaching a Persistent Volume to it and make it stateful, but all the pods of a deployment will be sharing the same Volume and data across all of them will be same.

### **Yaml file :**

#### **Yaml- Yet Another Markup Language**

The following are the building blocks of a YAML file:

- **Key Value Pair** — The basic type of entry in a YAML file is of a key value pair. After the Key and colon there is a space and then the value.
- **Arrays/Lists** — Lists would have a number of items listed under the name of the list. The elements of the list would start with a -. There can be a n of lists, however the indentation of various elements of the array matters a lot.
- **Dictionary/Map** — A more complex type of YAML file would be a Dictionary and Map

Yaml file for our project:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: cross-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: cross-app
  template:
    metadata:
      labels:
        app: cross-app
    spec:
      nodeSelector:
        "kubernetes.io/os": linux
      containers:
        - name: cross-app
          image: sarika088/cross
          env:
            - name: ALLOW_EMPTY_PASSWORD
              value: "yes"
          resources:
            requests:
              cpu: 100m
              memory: 128Mi
            limits:
              cpu: 250m
              memory: 256Mi
          ports:
            - containerPort: 5000
              name: cross1
---
apiVersion: v1
kind: Service
metadata:
  name: cross-app
spec:
  type: LoadBalancer
```



```

ports:
- port: 5000
  targetPort: 5000
  nodePort: 30000
selector:
  app: cross-app

```

Kubernetes deployment steps:

To create namespace:

Kubectl create namespace

To create pod:

Kubectl create pod

To run config file for deployment and service:

Kubectl apply -f deployment.yaml

To get nodes

Kubectl get nodes

```

Lenono@DESKTOP-MQF4H56 MINGW64 /d/SEM 5/assessment_project (main)
$ kubectl get nodes
NAME                STATUS    ROLES                  AGE      VERSION
docker-desktop     Ready    control-plane,master   3d12h    v1.22.4
(base)

```

To get Pods

Kubectl get pods

```

Lenono@DESKTOP-MQF4H56 MINGW64 /d/SEM 5/assessment_project (main)
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
cross-app-5595c56bb-x8blz          1/1     Running   2 (4h43m ago)    2d11h
hello                               1/1     Running   3 (4h43m ago)    3d4h
(base)

```

```
sarika@Azure:~$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
cross-app-6b69948dd7-8pfr2	1/1	Running	0	8m46s
cross-app-6b69948dd7-dxp2l	1/1	Running	0	3m1s
cross-app-6b69948dd7-t7gtb	1/1	Running	0	8m46s

```
sarika@Azure:~$
```

To get services link

Kubectl get service (30000)

```
sarika@Azure:~$ kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
cross-app	LoadBalancer	10.0.8.41	20.121.98.187	5000:30000/TCP	2d11h
kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP	2d12h

List all pods in all namespaces

kubectl get pods --all-namespaces

```
sarika@Azure:~$ kubectl get pods --all-namespaces
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
default	cross-app-6b69948dd7-8pfr2	1/1	Running	0	10m
default	cross-app-6b69948dd7-dxp2l	1/1	Running	0	4m30s
default	cross-app-6b69948dd7-t7gtb	1/1	Running	0	10m
kube-system	azure-ip-masq-agent-9268x	1/1	Running	0	5m9s
kube-system	azure-ip-masq-agent-sszx4	1/1	Running	0	5m7s
kube-system	coredns-845757d86-8qnls	1/1	Running	0	10m
kube-system	coredns-845757d86-m5ngz	1/1	Running	0	10m
kube-system	coredns-autoscaler-5f85dc856b-khvwz	1/1	Running	0	10m
kube-system	csi-azuredisk-node-2z8lm	3/3	Running	0	5m9s
kube-system	csi-azuredisk-node-n769z	3/3	Running	0	5m7s
kube-system	csi-azurefile-node-5xwld	3/3	Running	0	5m9s
kube-system	csi-azurefile-node-wvc7l	3/3	Running	0	5m7s
kube-system	kube-proxy-bxdc7	1/1	Running	0	5m9s
kube-system	kube-proxy-wlkk5	1/1	Running	0	5m7s
kube-system	metrics-server-6bc97b47f7-9wgqn	1/1	Running	0	10m
kube-system	tunnelfront-58bf44f648-z452n	1/1	Running	0	10m

List all pods in the current namespace, with more

kubectl get pods -o wide

```
sarika@Azure:~$ kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
cross-app-6b69948dd7-8pfr2	1/1	Running	0	10m	10.244.0.7	aks-agentpool-33418364-vmss000004	<none>	<none>
cross-app-6b69948dd7-dxp2l	1/1	Running	0	5m14s	10.244.1.2	aks-agentpool-33418364-vmss000005	<none>	<none>
cross-app-6b69948dd7-t7gtb	1/1	Running	0	10m	10.244.0.3	aks-agentpool-33418364-vmss000004	<none>	<none>

```
sarika@Azure:~$
```

To fetch the pod logs

kubectl logs my-pod

```
sarika@Azure:~$ kubectl logs cross-app-6b69948dd7-8pfr2
* Serving Flask app "/opt/app/app.py"
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
sarika@Azure:~$
```

Stop cluster

```
az aks stop --name mykubecuster --resource-group AzureCluster
```

Deployed image :

Prediction for cross sell

Gender  
Female

Age  
20

Previously insured ?  
No

Vehicle Age  
1-2 years

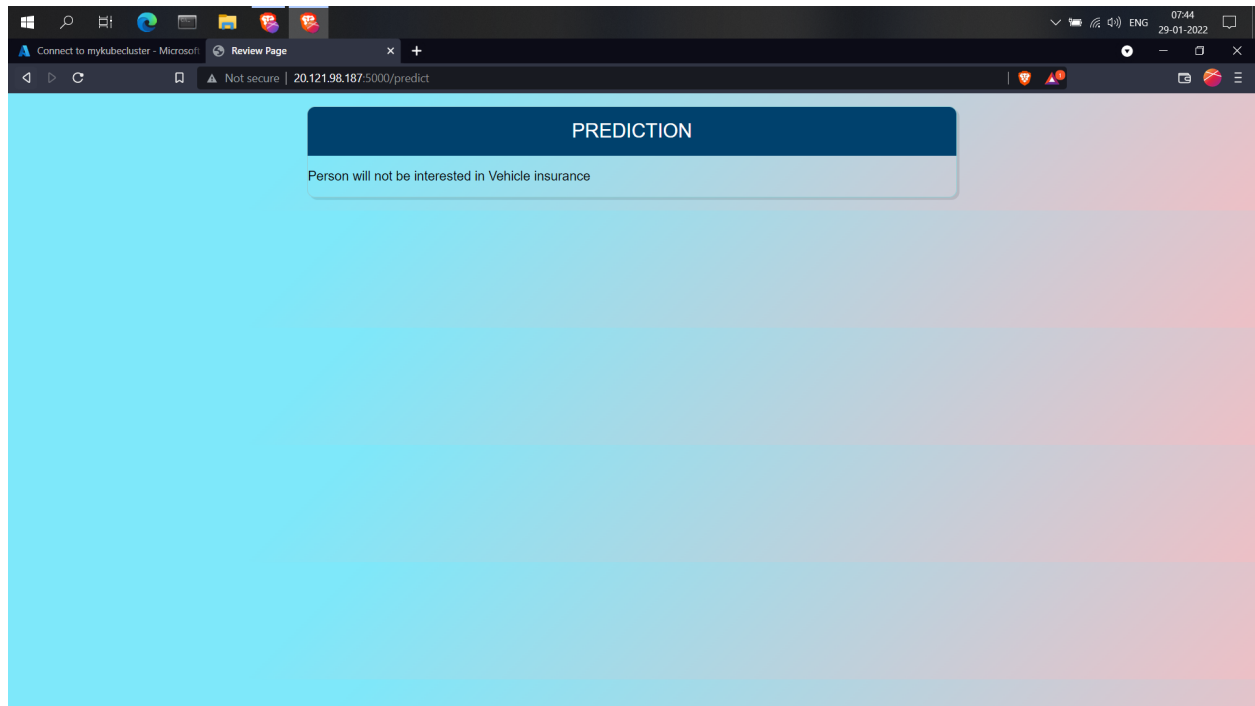
Is Vehicle Damaged ?  
No

Annual Premium  
12324332

Vintage  
234

Please select your Algorithm  
☐ logistic ☒ Decision

SUBMIT



**Conclusion :**

Successfully dockerized and deployed a web application in Azure Kubernetes cluster