PET DETECTION USING RASPBERRY PI

SUPERVISED BY:

PROF. DR. CHRISTIAN BAUN,

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Group-3

Shikha Singh (1421659)

Sushil Koirala (1421387)

Naga Venkata Sai Kumar Vadlani (1417605)

Vikrant Shah (1411054)

Zaghman Saghir (1411960)

Content

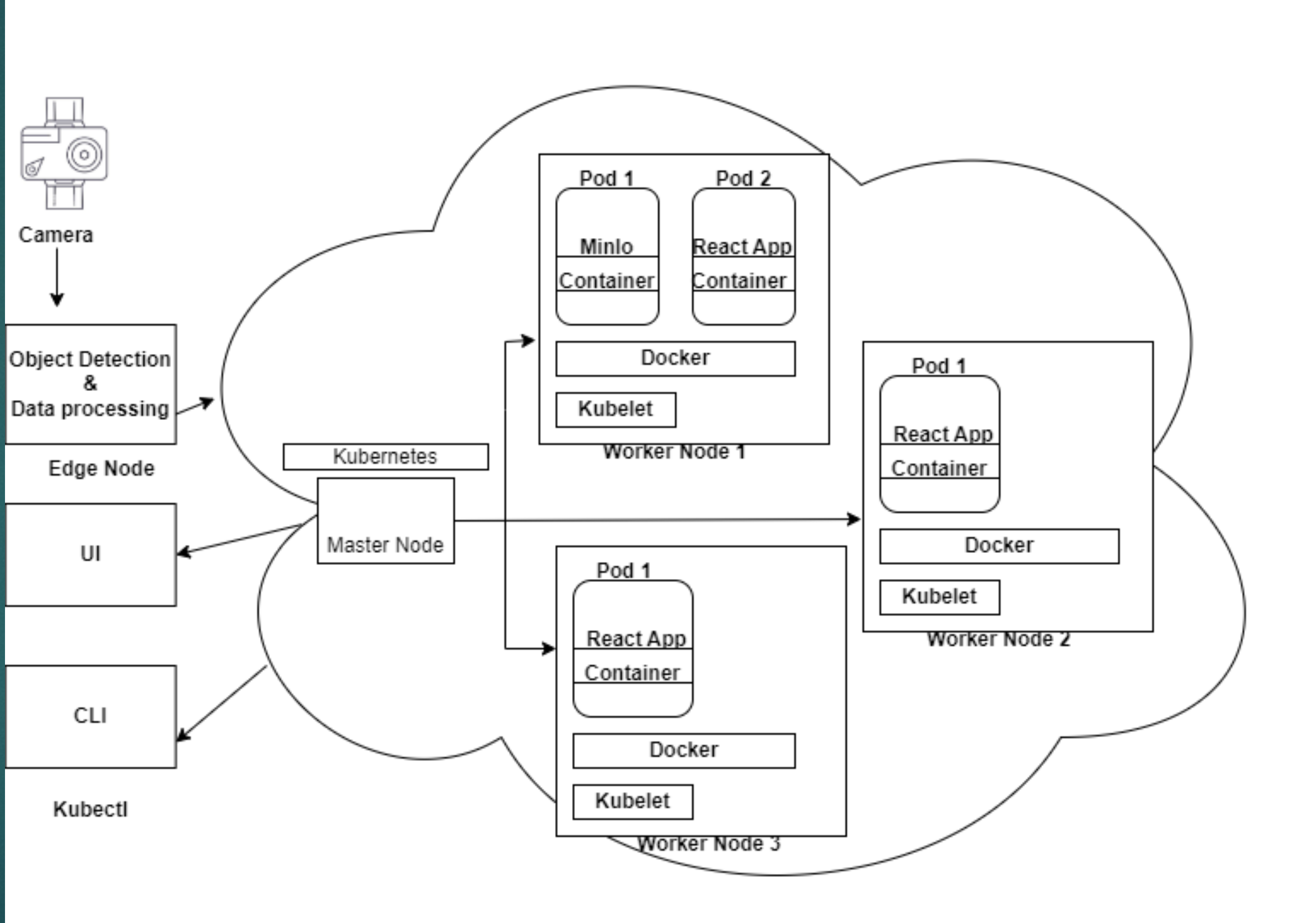
- Introduction
- System Architecture
- K3S Kubernetes Cluster
- Model Training
- Docker and MinIO
- Backend and Frontend UI
- Demonstration

Introduction

- An edge computing solution has been developed to identify the presence of pets at the sensor node.

- This system proposes **Raspberry Pi** for model implementation, **Cameras** for capturing images, and **YOLO** v5s for testing and training of the objections detection model.

System Architecture



K3S Kubernetes Cluster

For the Raspberry pi cluster, a lightweight Kubernetes distribution k3S is used.

1. Lightweight Kubernetes Distribution: K3s is an open-source, lightweight Kubernetes distribution designed for edge computing and resource-constrained environments.

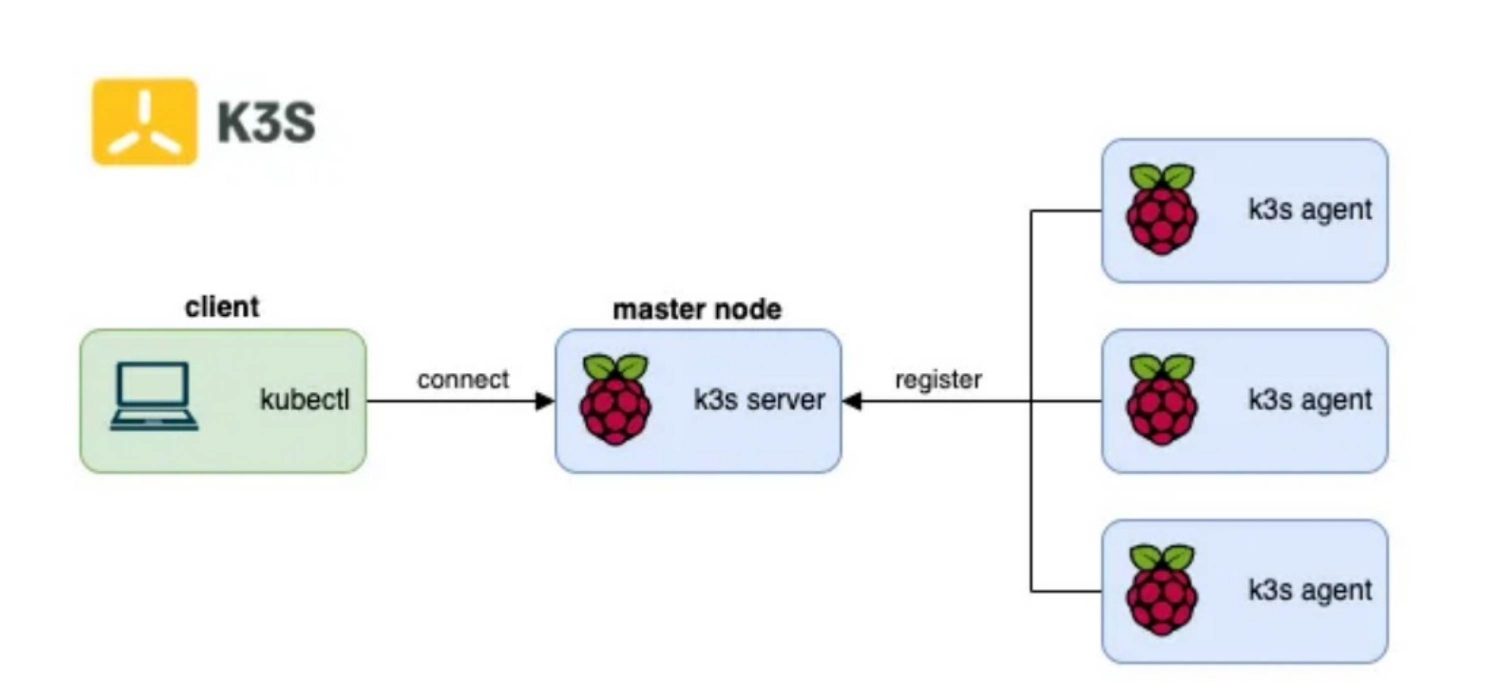
2. Simplified Operations: K3s offers simplified operations with reduced resource requirements, making it easier to deploy and manage Kubernetes clusters.

3. Essential Features: Despite its lightweight nature, K3s retains essential features such as container orchestration, automatic scaling, load balancing, Helm charts support, and a built-in service mesh (Traefik).

4. Use Cases: K3s is well-suited for edge computing, Internet of Things (IoT) deployments, and environments with limited resources, where efficiency and performance are critical.

5. Active Community and Future Developments: K3s benefits from an active community and enjoys ongoing development, ensuring a vibrant ecosystem with regular updates, improvements, and community-driven enhancements.

K3S Kubernetes Cluster



K3S Kubernetes Cluster

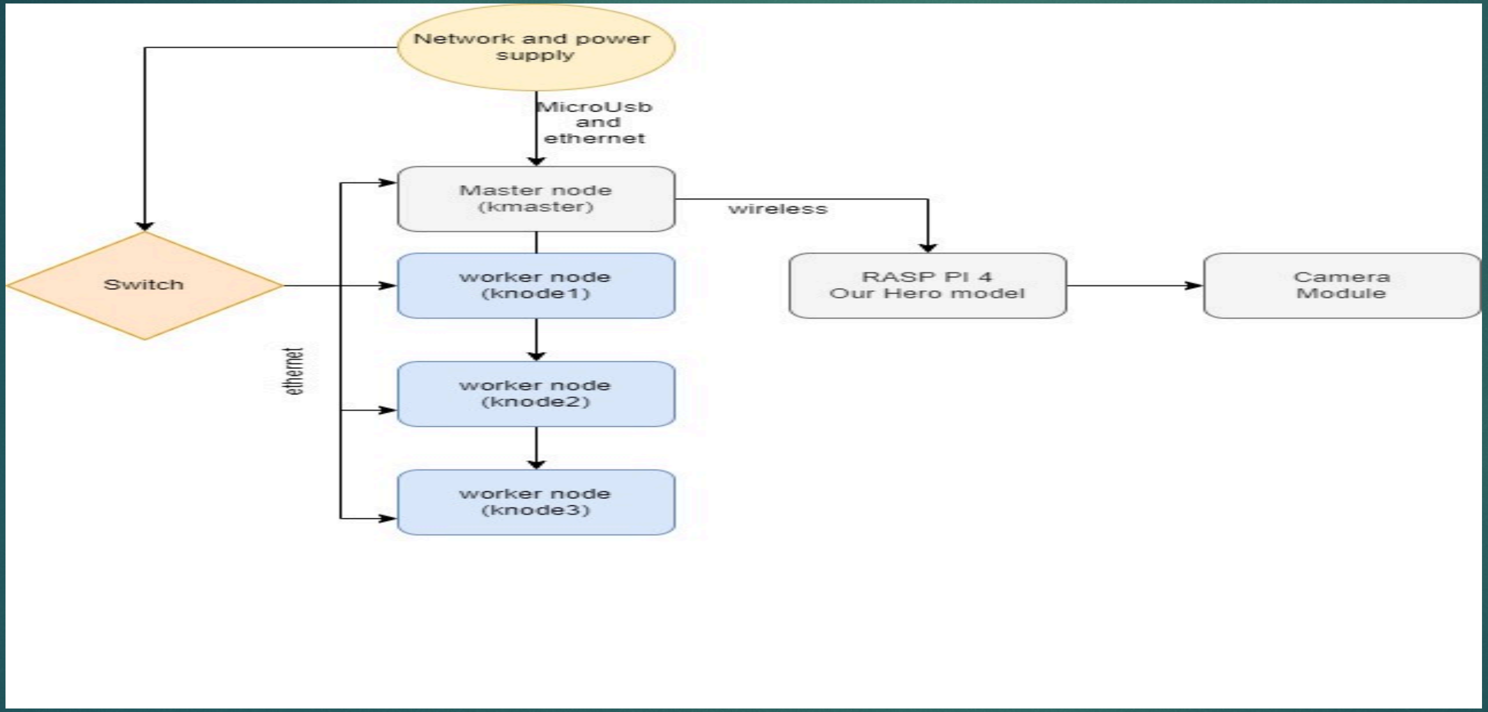
- To install K3S the following command must be executed on the master node:

```
curl sfL https://get.k3s.io | K3S_KUBECONFIG_MODE="644" sh -s –
```

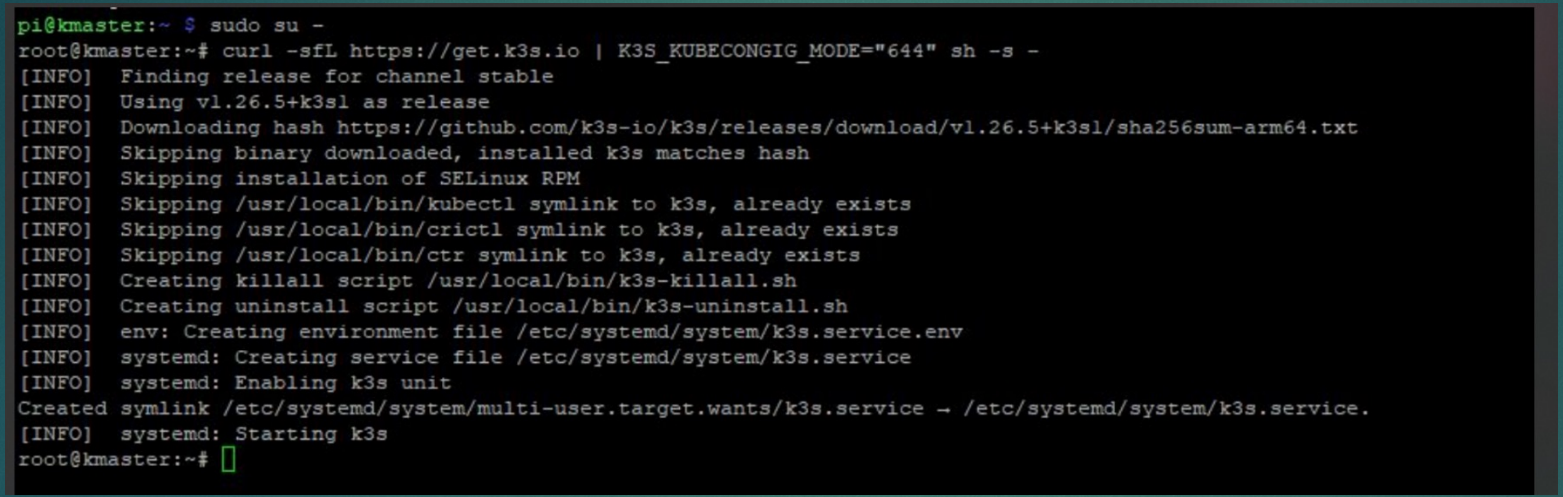
- With this token, k3S can be installed on worker nodes bythe following command:

```
curl sfL https://get.k3s.io |K3S_TOKEN="<TOKEN>"
K3S_URL="https://<master_node_ip>:6443" sh -
```

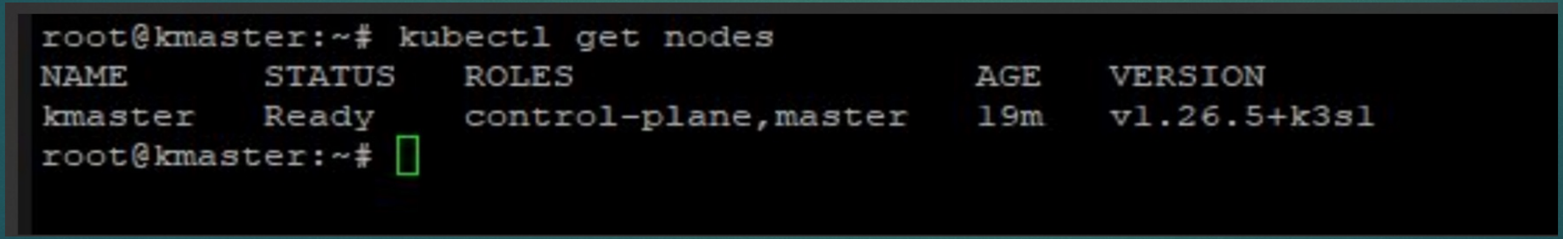
K3S Kubernetes Cluster



K3S Kubernetes Cluster

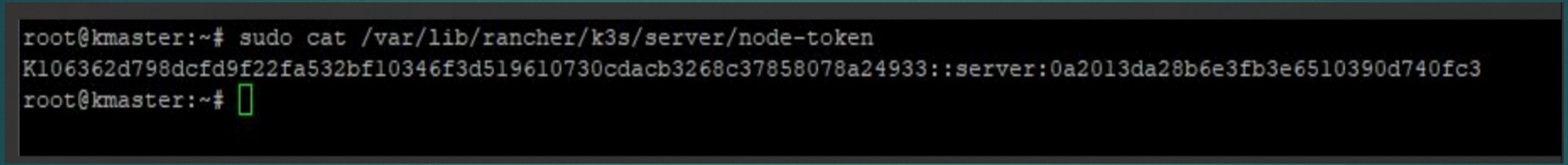


```
pi@kmaster:~ $ sudo su -
root@kmaster:~# curl -sfL https://get.k3s.io | K3S_KUBECONGIG_MODE="644" sh -s -
[INFO]  Finding release for channel stable
[INFO]  Using v1.26.5+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.26.5+k3s1/sha256sum-arm64.txt
[INFO]  Skipping binary downloaded, installed k3s matches hash
[INFO]  Skipping installation of SELinux RPM
[INFO]  Skipping /usr/local/bin/kubectl symlink to k3s, already exists
[INFO]  Skipping /usr/local/bin/crictl symlink to k3s, already exists
[INFO]  Skipping /usr/local/bin/ctr symlink to k3s, already exists
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s.service
[INFO]  systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.
[INFO]  systemd: Starting k3s
root@kmaster:~# 
```

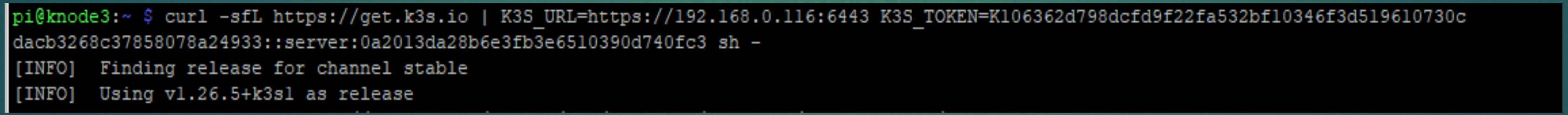


```
root@kmaster:~# kubectl get nodes
NAME      STATUS   ROLES                  AGE   VERSION
kmaster   Ready    control-plane,master   19m   v1.26.5+k3s1
root@kmaster:~# 
```

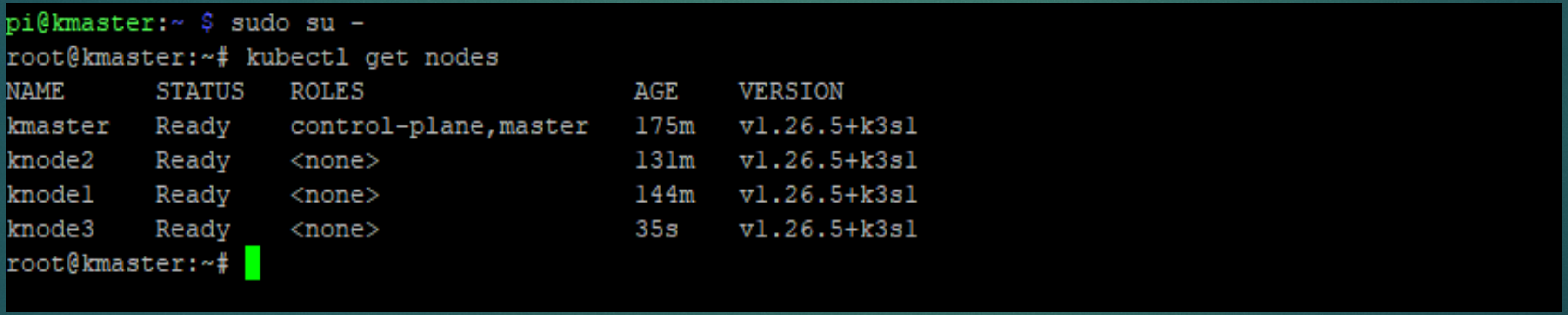
K3S Kubernetes Cluster



```
root@kmaster:~# sudo cat /var/lib/rancher/k3s/server/node-token
K106362d798dcfd9f22fa532bf10346f3d519610730cdacb3268c37858078a24933::server:0a2013da28b6e3fb3e6510390d740fc3
root@kmaster:~# 
```



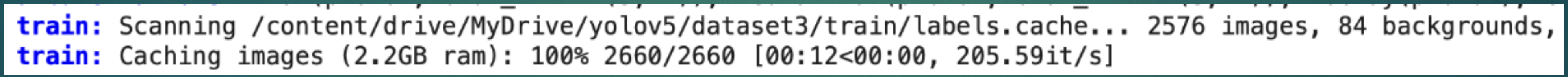
```
pi@knode3:~ $ curl -sfL https://get.k3s.io | K3S_URL=https://192.168.0.116:6443 K3S_TOKEN=K106362d798dcfd9f22fa532bf10346f3d519610730c
dacb3268c37858078a24933::server:0a2013da28b6e3fb3e6510390d740fc3 sh -
[INFO]  Finding release for channel stable
[INFO]  Using v1.26.5+k3s1 as release
```



```
pi@kmaster:~ $ sudo su -
root@kmaster:~# kubectl get nodes
NAME      STATUS   ROLES                  AGE    VERSION
kmaster   Ready    control-plane,master   175m   v1.26.5+k3s1
knode2    Ready    <none>                 131m   v1.26.5+k3s1
knode1    Ready    <none>                 144m   v1.26.5+k3s1
knode3    Ready    <none>                 35s    v1.26.5+k3s1
root@kmaster:~# 
```

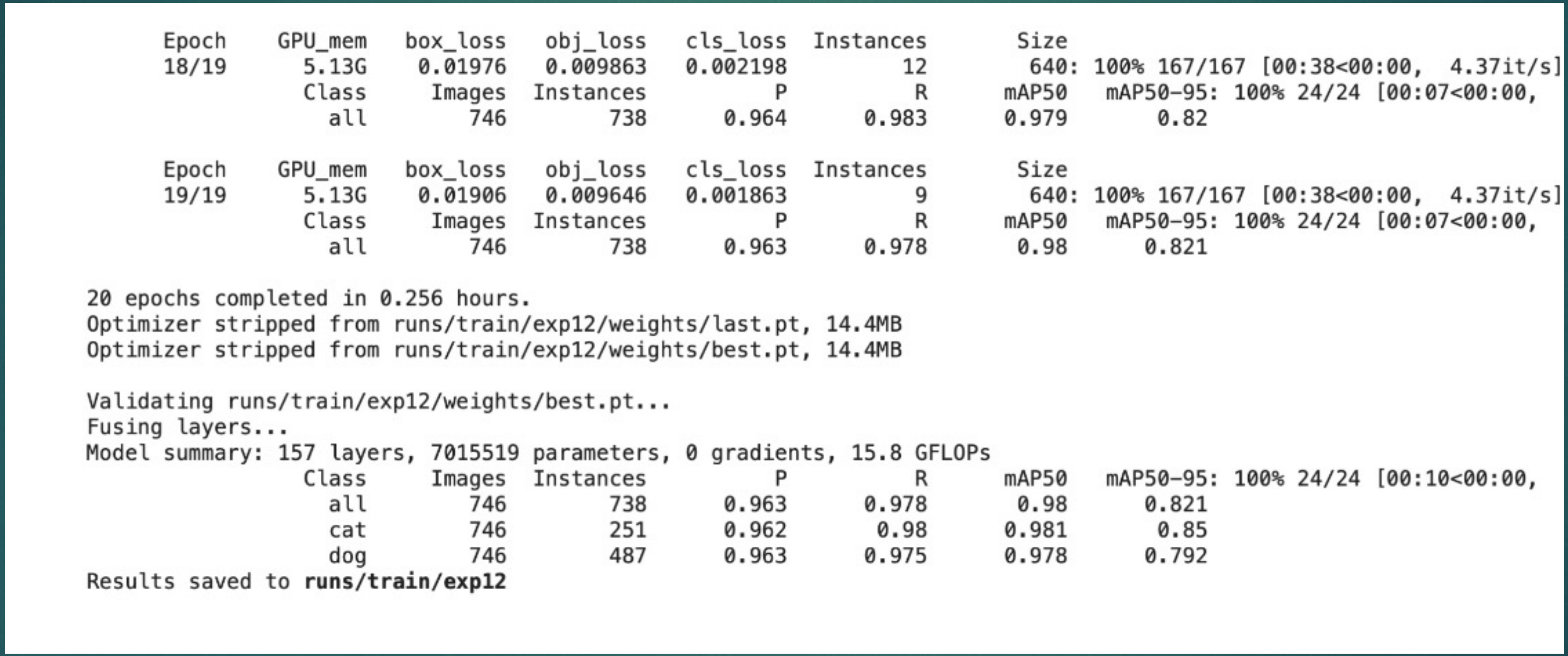
Model Training

- Trained the YOLO v5s model using dataset using google Collab.
- We used the datasets for dogs and cats to transfer train the model.
- 2k+ images were used.
- A total of 20 epochs were run for the training purpose.



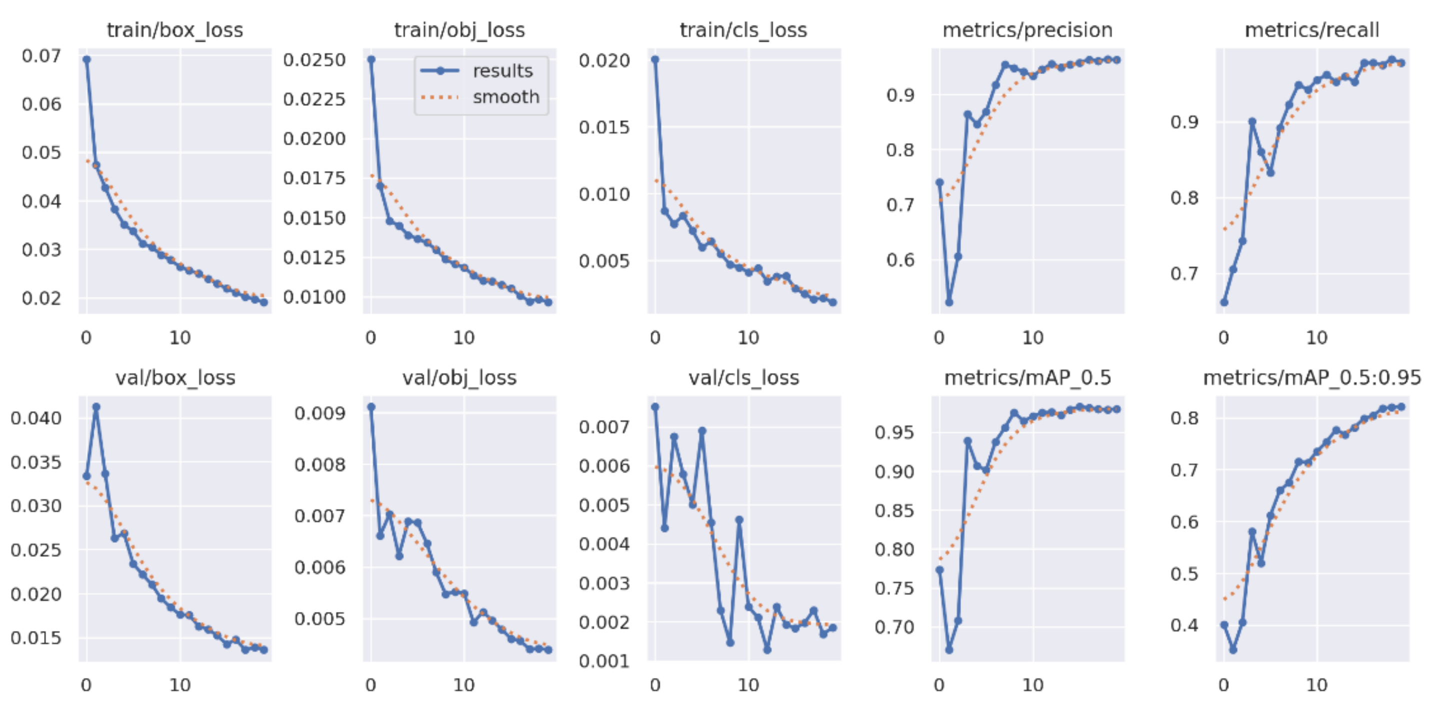
```
train: Scanning /content/drive/MyDrive/yolov5/dataset3/train/labels.cache... 2576 images, 84 backgrounds,
train: Caching images (2.2GB ram): 100% 2660/2660 [00:12<00:00, 205.59it/s]
```

Model Training



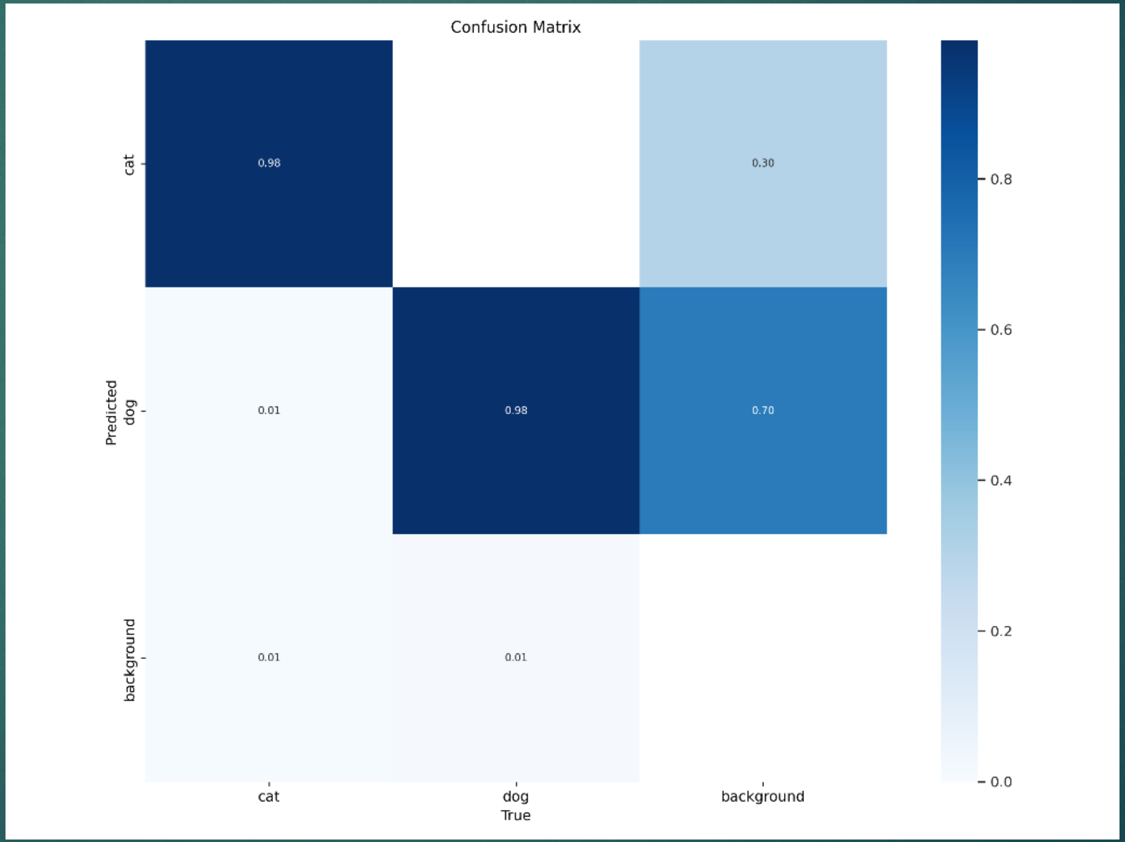
```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      18/19      5.13G    0.01976   0.009863   0.002198         12        640: 100% 167/167 [00:38<00:00,  4.37it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 24/24 [00:07<00:00,
                   all        746        738      0.964      0.983      0.979       0.82

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      19/19      5.13G    0.01906   0.009646   0.001863          9        640: 100% 167/167 [00:38<00:00,  4.37it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 24/24 [00:07<00:00,
                   all        746        738      0.963      0.978       0.98      0.821

20 epochs completed in 0.256 hours.
Optimizer stripped from runs/train/exp12/weights/last.pt, 14.4MB
Optimizer stripped from runs/train/exp12/weights/best.pt, 14.4MB

Validating runs/train/exp12/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 24/24 [00:10<00:00,
                   all        746        738      0.963      0.978       0.98      0.821
                   cat        746        251      0.962       0.98      0.981       0.85
                   dog        746        487      0.963      0.975      0.978      0.792
Results saved to runs/train/exp12
```

Model Training



Model Training

By examining the confusion matrix, we can observe that the model has made accurate predictions in most cases, with only a few minor errors.



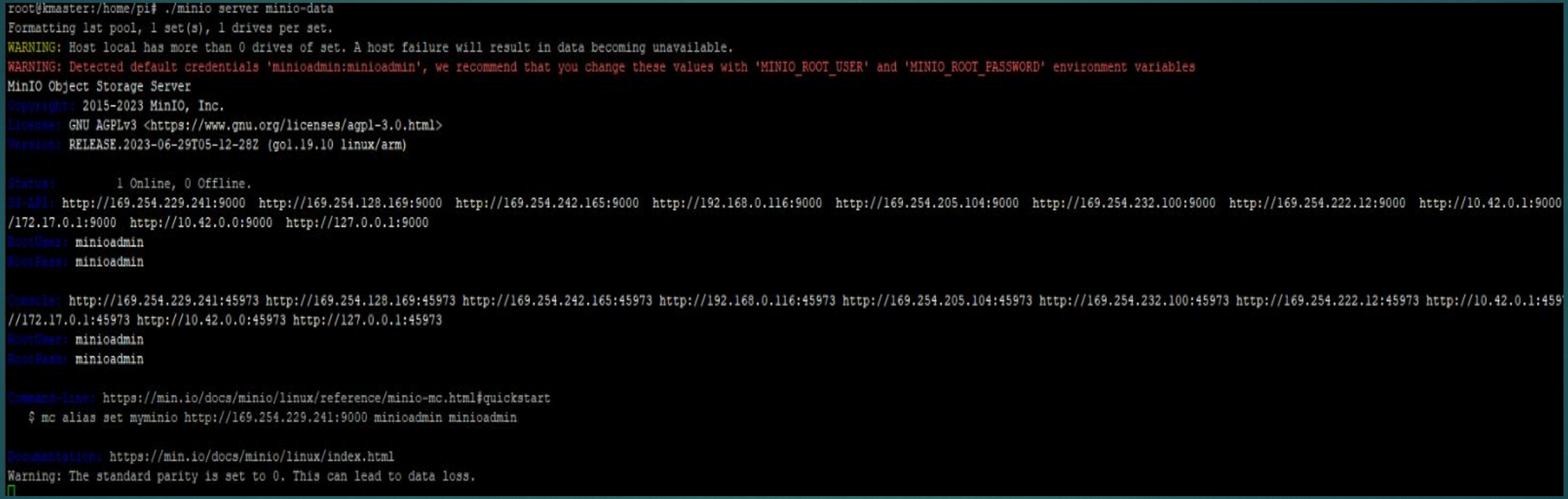
Docker as Runtime

- Installing docker runtime on each node.
- Setting the k3s on the master using docker as runtime instead of contained.
- Setting the worker nodes
- Enabling the kubectl for ease of use

MinIO deployment

- Using MinIO for resilience, scalability features
- Efficient for Object detection software
- Availability of RESTful API suited for Kubernetes clusters
- Single Node Single Drive v/s Multi Node Multi Drive
- Investigated use of Hazelcast for synchronization
- Enabled as a service for UI Edge Node usage
- Use of Persistent Volume claims

MinIO deployment

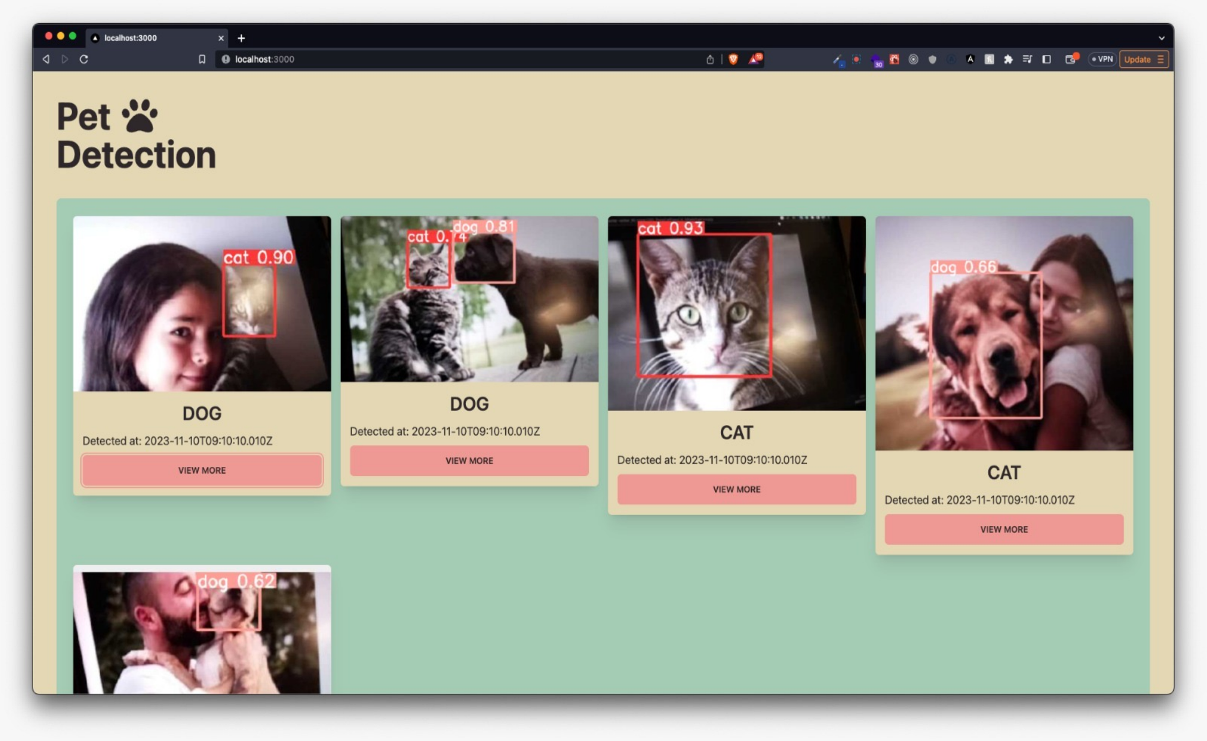


```
root@kmaster:/home/pi# ./minio server minio-data
Formatting 1st pool, 1 set(s), 1 drives per set.
WARNING: Host local has more than 0 drives of set. A host failure will result in data becoming unavailable.
WARNING: Detected default credentials 'minioadmin:minioadmin', we recommend that you change these values with 'MINIO_ROOT_USER' and 'MINIO_ROOT_PASSWORD' environment variables
MinIO Object Storage Server
Copyright: 2015-2023 MinIO, Inc.
License: GNU AGPLv3 <https://www.gnu.org/licenses/agpl-3.0.html>
Version: RELEASE.2023-06-29T05-12-28Z (go1.19.10 linux/arm)

Status:         1 Online, 0 Offline.
S3-API: http://169.254.229.241:9000  http://169.254.128.169:9000  http://169.254.242.165:9000  http://192.168.0.116:9000  http://169.254.205.104:9000  http://169.254.232.100:9000  http://169.254.222.12:9000  http://10.42.0.1:9000
/172.17.0.1:9000  http://10.42.0.0:9000  http://127.0.0.1:9000
RootUser: minioadmin
RootPass: minioadmin

Console: http://169.254.229.241:45973 http://169.254.128.169:45973 http://169.254.242.165:45973 http://192.168.0.116:45973 http://169.254.205.104:45973 http://169.254.232.100:45973 http://169.254.222.12:45973 http://10.42.0.1:459
//172.17.0.1:45973 http://10.42.0.0:45973 http://127.0.0.1:45973
RootUser: minioadmin
RootPass: minioadmin

Command-line: https://min.io/docs/minio/linux/reference/minio-mc.html#quickstart
   $ mc alias set myminio http://169.254.229.241:9000 minioadmin minioadmin

Documentation: https://min.io/docs/minio/linux/index.html
Warning: The standard parity is set to 0. This can lead to data loss.
```

FRONTEND

- HTML, CSS, JavaScript and ReactJS:
- HTML (Hypertext Markup Language) is used to structure the content of web pages.
- CSS (Cascading Style Sheets) is used to define the appearance and layout of web pages.
- JavaScript is a client-side scripting language that can be used to add interactivity and dynamic behavior to web pages.

RESULTS



Pet 🐾
Detection

DOG
Detected at: 2023-11-10T09:10:10.010Z

DOG
Detected at: 2023-11-10T09:10:10.010Z

CAT
Detected at: 2023-11-10T09:10:10.010Z

CAT
Detected at: 2023-11-10T09:10:10.010Z

DEMONSTRATION

Thank you!!

Reference:

[1] [Online]. Available: ”https://medium.com/thinkport/how-to-build-a-raspberry-pi-kubernetes-cluster-with-k3s-76224788576c

[2] [Online].Available:”https://docs.k3s.io/advanced#raspberry-pi

[3] [Online]. Available: ”https://saintcoder.wordpress.com/2017/08/14/ sharing-internet-connection-to-raspberry-pis-wired-to-local-network/

[4] [Online]. Available: https://medium.com/datadriveninvestor/how-do-you-use-yolo-to-detect-objects- 8bb634ec7d2c

[5] [Online]. Available: https://jooskorstanje.com/super_simple_yolo_notebook.html

[6] [Online]. Available: https://github.com/ultralytics/yolov5

[7] [Online]. Available: https://pytorch.org/hub/ultralytics_yolov5/

[8] [Online].Available:”https://www.eclipse.org/paho/index.php?page=clients/python/index.php

[9] [Online] https://min.io/docs/minio/kubernetes/upstream/