Commands used to create assignment 1:

Firewall

sudo apt-get -y update

sudo apt-get -y install firewalld

sudo systemctl enable firewalld

sudo firewall-cmd --state

sudo firewall-cmd --permanent --zone=public --add-port=6443/tcp

sudo firewall-cmd --permanent --zone=public --add-port=5000/tcp

sudo firewall-cmd --permanent --zone=public --add-port=8080/tcp

sudo firewall-cmd --permanent --zone=public --add-port=80/tcp

sudo firewall-cmd --permanent --zone=public --add-port=10250/tcp

sudo firewall-cmd --permanent --zone=public --add-port=6784/udp

sudo firewall-cmd --permanent --zone=public --add-port=6783/udp

sudo firewall-cmd --permanent --zone=public --add-port=6783/tcp

sudo firewall-cmd --reload

Docker

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl gnupg2 software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get update

sudo apt-get install -y docker.io

sudo usermod -aG docker ${USER}

Put all the files from winSC

sudo apt install python3-pip

pip install flask numpy opencv-python-headless

Commands to run locally:

python3 iWebLens\_server.py

python3 iWebLens\_client.py inputfolder/ http://10.0.0.192:5000/api/object\_detection 4

Build Docker:

docker build -t assignmentdocker .

Commands to run remotely:

docker run --rm -p 5000:5000 assignmentdocker:latest

python3 iWebLens\_client.py inputfolder/ http://168.138.12.102:5000/api/object\_detection 4

Kubernetes

sudo apt-get update  
sudo apt-get install [docker.io](http://docker.io/)  
docker version  
sudo systemctl enable docker  
sudo systemctl status docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker  
curl -s <https://packages.cloud.google.com/apt/doc/apt-key.gpg> | sudo apt-key add  
sudo apt-add-repository "deb <http://apt.kubernetes.io/> kubernetes-xenial main"  
sudo apt-get install kubeadm kubelet kubectl  
sudo apt-mark hold kubeadm kubelet kubectl  
kubeadm version

(Optional, If you have problems about get the token, try:

sudo nano /etc/docker/daemon.json

and added below:

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

Then

sudo systemctl daemon-reload

sudo systemctl restart docker

sudo systemctl restart kubelet

)

sudo hostnamectl set-hostname w1 (for worker 1)

sudo hostnamectl set-hostname w2 (for worker 2)

sudo hostnamectl set-hostname master (for master)

sudo swapoff -a

sudo sed -i '/ swap / s/^/#/' /etc/fstab

Run on master:

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

https://kubernetes.io/docs/concepts/cluster-administration/addons/

Weave on master:

kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 10.0.0.192:6443 --token 4ddu7w.rlt377v91gnmush6 --discovery-token-ca-cert-hash sha256:3df99cd7d1477b2029ee1139b66fd1a2f8335cbf0745ed0c0bcd51ed470a428a

to check the nodes on master:

I/P:

kubectl get nodes

O/P:

master Ready control-plane,master 8m40s v1.23.6

w1 Ready <none> 73s v1.23.6

w2 Ready <none> 83s v1.23.6