Configuration Manual

MSc Research Project

Cloud Computing

Sagar Padhi

Student ID: x21245673

School of Computing

National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sagar Padhi		
Student ID:	x21245673		
Programme:	Cloud Computing	**Year:**	2024
Module:	MSc Research Project		
Lecturer:	Rejwanul Haque		
Submission Due Date:	20/12/2024		
Project Title:	PID control based Cluster Autoscaling for Kubernetes		
Word Count:	329	**Page Count: 7**	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sagar Swarajya Padhi

Date: 12 December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

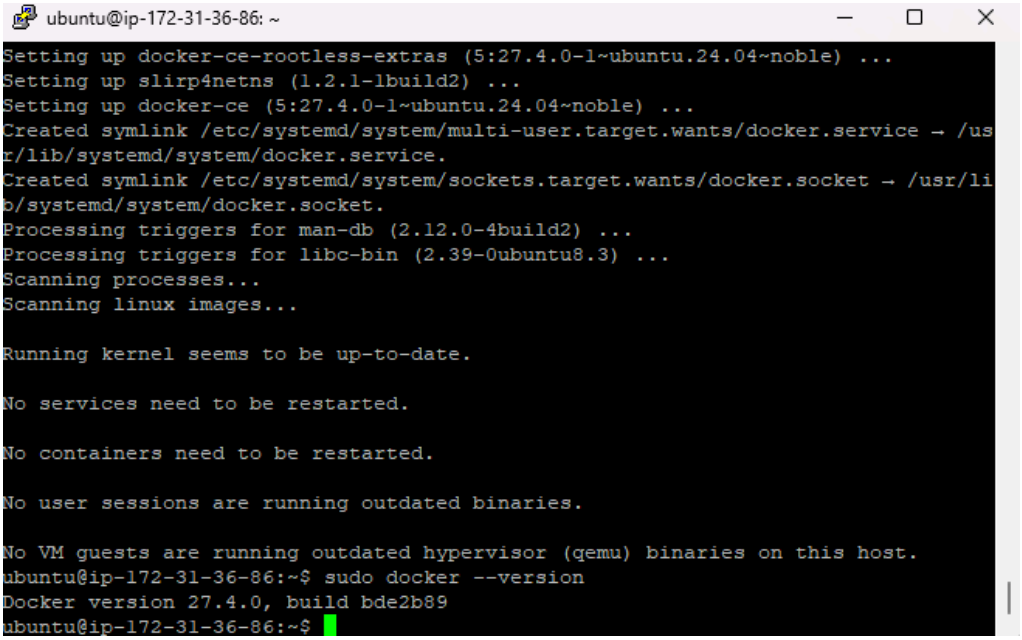
Configuration Manual

Sagar Padhi
Student ID: x21245673

1 Systems Basic Requirements

Any cloud provider with a cloud instance with CPU more than 4 and 100gig of storage memory.AWS cloud for this case.
Deploy 2 instances , 1 for master and 1 for wroker node
For Ubuntu based servers , run the basic commands to update the kernel and install basic tools like docker and aws cli.

sudo apt update -y
Install docker by following command.
sudo apt install -y docker



Start the docker services.
sudo systemctl start docker
sudo systemctl enable docker
check the docker is started and working smoothly.
docker –version

Kubernetas setup

sudo apt-get update

apt-transport-https may be a dummy package; if so, you can skip that package

sudo apt-get install -y apt-transport-https ca-certificates curl gpg

sudo mkdir -p -m 755 /etc/apt/keyrings

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.32/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

This overwrites any existing configuration in /etc/apt/sources.list.d/kubernetes.list

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.32/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

Enable the kubelet service before running kubeadm

sudo systemctl enable --now kubelet
sudo apt-get update && sudo apt-get upgrade -y

Kubernetes requires swap to be disabled

sudo swapoff -a
sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab

sudo apt-get install -y apt-transport-https curl
Add packages
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg **| sudo apt-key add** .
Enable Kernel Modules: Ensure the required kernel modules are loaded:

```
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
```

Set Sysctl Parameters: Apply sysctl parameters required by Kubernetes:

```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables  = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward                 = 1
EOF
sudo sysctl –system
```

.

2 Configuration of kubernetes cluster.

On the master node (control plane), initialize the cluster
sudo kubeadm init

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

once this is initialized in the control plane a token will be generated

```
root@controlplane: ~

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

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

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/

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

kubeadm join 172.31.46.40:6443 --token oon6t5.11f75e6xuuqo3scd \
        --discovery-token-ca-cert-hash sha256:0c53f2f4e2952ff48f79658105cefb2468
60dcbce901cleea693c6f183168d98
root@controlplane:~#
```

as shown in the Fig above the join command and hash key is provided.

Once the worker node is connected to the Control plane, the firs step is to create a application image for testing . in applications folder a file named app.go is used.

```
root@controlplane: ~/application
 ---> Running in be4ad2b55513
 ---> Removed intermediate container be4ad2b55513
 ---> 235ed772af19
Step 4/8 : COPY app.go /src
 ---> e8d69b832c67
Step 5/8 : RUN go build app.go
 ---> Running in 1f4ecc490584
 ---> Removed intermediate container 1f4ecc490584
 ---> a55f2459e282
Step 6/8 : FROM alpine as runtime
latest: Pulling from library/alpine
38a8310d387e: Pull complete
Digest: sha256:21dc6063fd678b478f57c0e13f47560d0ea4eeba26dfc947b2a4f81f686b9f45
Status: Downloaded newer image for alpine:latest
 ---> 4048db5d3672
Step 7/8 : COPY --from=build /src/app /app/app
 ---> 75c3cc0c645f
Step 8/8 : CMD [ "/app/app" ]
 ---> Running in 5c5d52c99594
 ---> Removed intermediate container 5c5d52c99594
 ---> 47308a3d9098
Successfully built 47308a3d9098
Successfully tagged aimvector/application-cpu:v1.0.0
root@controlplane:~/application#
```

CPU-intensive web application designed to simulate real-world microservice workloads. This application is central to the evaluation and testing of the PID-based autoscaling framework within the Kubernetes cluster.This application image file needs to be created.

docker build . -t aimvector/application-cpu:v1.0.0

Apply deployment.yaml

Kubectl apply –f application/deployment.yaml --validate=false.
in this step the daemonset feature and other services are passed in the deployment feature .

Once the service is deployed
Deploy metric server.

Kubectl apply if metrics/metricserver.yaml
Kubectl apply if metrics/traffic-simulator.yaml

3 Testing the cluster

To simulate different rates of workloads the replica of the dockerfile can be deployed and scaled as required.
Kubectl scale deploy/application-cpu --replicas 2

the number of load and me varied by changing the number of replicas at the end of the command. to study the metrics we can acces the metric server installed and used .

by using following commands different metrics are obtained as shown in the results below
kubectl top nodes
kubectl get nodes
kubectl get pods -o wide
kubectl get pods -n kube-system

results obtailed from scaling from 12 - > 2 replicas

```
root@controlplane: ~
        AGE
default       test-nginx                          0/1     ContainerCreating   0
        21m
kube-system   coredns-668d6bf9bc-ccjmp            0/1     ContainerCreating   0
        32m
kube-system   coredns-668d6bf9bc-vx79q            0/1     ContainerCreating   0
        32m
kube-system   etcd-controlplane                   1/1     Running             47 (3m37
s ago)   30m
kube-system   kube-apiserver-controlplane         1/1     Running             47 (3m9s
 ago)    32m
kube-system   kube-controller-manager-controlplane   0/1     CrashLoopBackOff    23 (86s
ago)     33m
kube-system   kube-proxy-5qrjb                    1/1     Running             16 (100s
 ago)    32m
kube-system   kube-proxy-c257b                    1/1     Running             9 (5m11s
 ago)    32m
kube-system   kube-scheduler-controlplane         1/1     Running             49 (3m1s
 ago)    21m
root@controlplane:~# mkdir metrics
root@controlplane:~# cd metrics
root@controlplane:~/metrics# nano metricserver.yaml
root@controlplane:~/metrics# cd
root@controlplane:~# kubectl apply -f metrics/metricserver.yaml
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
serviceaccount/metrics-server created
deployment.apps/metrics-server created
service/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
error: resource mapping not found for name: "v1beta1.metrics.k8s.io" namespace: "" from "
metrics/metricserver.yaml": no matches for kind "APIService" in version "apiregistration.
k8s.io/v1beta1"
ensure CRDs are installed first
root@controlplane:~# kubectl get pods -owide
NAME                                READY   STATUS              RESTARTS   AGE     IP
  NODE          NOMINATED NODE   READINESS GATES
application-cpu-557dd6bf86-jklwf    0/1     ContainerCreating   0          4m32s   <none>
  workernode1   <none>           <none>
test-nginx                          0/1     ContainerCreating   0          26m     <none>
  workernode1   <none>           <none>
root@controlplane:~#
```

results from continiuousscaling down

```
root@controlplane: ~
The connection to the server 172.31.46.40:6443 was refused - did you specify the right host or port?
root@controlplane:~# kubectl get nodes
The connection to the server 172.31.46.40:6443 was refused - did you specify the right host or port?
root@controlplane:~# kubectl get pods
The connection to the server 172.31.46.40:6443 was refused - did you specify the right host or port?
root@controlplane:~# kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
application-cpu-557dd6bf86-bfr7h    0/1     Pending             0          25s
application-cpu-557dd6bf86-jklwf    0/1     ContainerCreating   0          16m
test-nginx                          0/1     ContainerCreating   0          38m
root@controlplane:~# kubectl get pods -o wide
NAME                                READY   STATUS              RESTARTS   AGE   IP       NODE          NOMINATED NODE   READ
INESS GATES
application-cpu-557dd6bf86-bfr7h    0/1     Pending             0          41s   <none>   <none>        <none>           <non
e>
application-cpu-557dd6bf86-jklwf    0/1     ContainerCreating   0          17m   <none>   workernode1   <none>           <non
e>
test-nginx                          0/1     ContainerCreating   0          38m   <none>   workernode1   <none>           <non
e>
root@controlplane:~# kubectl get pods -o wide
NAME                                READY   STATUS              RESTARTS   AGE   IP       NODE          NOMINATED NODE   READ
INESS GATES
application-cpu-557dd6bf86-bfr7h    0/1     Pending             0          95s   <none>   <none>        <none>           <non
e>
application-cpu-557dd6bf86-jklwf    0/1     ContainerCreating   0          17m   <none>   workernode1   <none>           <non
e>
test-nginx                          0/1     ContainerCreating   0          39m   <none>   workernode1   <none>           <non
e>
root@controlplane:~# kubectl scale deploy/application-cpu --replicas 4
deployment.apps/application-cpu scaled
root@controlplane:~# kubectl get pods -o wide
NAME                                READY   STATUS              RESTARTS   AGE     IP       NODE          NOMINATED NODE   RE
ADINESS GATES
application-cpu-557dd6bf86-bfr7h    0/1     Pending             0          2m14s   <none>   <none>        <none>           <n
one>
application-cpu-557dd6bf86-hkr45    0/1     Pending             0          10s     <none>   <none>        <none>           <n
one>
application-cpu-557dd6bf86-jklwf    0/1     ContainerCreating   0          18m     <none>   workernode1   <none>           <n
one>
application-cpu-557dd6bf86-tzz67    0/1     Pending             0          10s     <none>   <none>        <none>           <n
one>
test-nginx                          0/1     ContainerCreating   0          40m     <none>   workernode1   <none>           <n
one>
root@controlplane:~#
```

results from constant spike in traffic

```
root@controlplane: ~
root@controlplane:~# kubectl get pods -o wide
The connection to the server 172.31.46.40:6443 was refused - did you specify the right host or port?
root@controlplane:~# sudo systemctl restart kubelet
root@controlplane:~# kubectl get nodes
NAME           STATUS   ROLES           AGE    VERSION
controlplane   Ready    control-plane   167m   v1.32.0
workernode1    Ready    <none>          165m   v1.32.0
root@controlplane:~# kubectl get pods -o wide
NAME                                READY   STATUS              RESTARTS   AGE    IP       NODE          NOMINATED NODE   READINESS GATES
application-cpu-557dd6bf86-41f1d    0/1     Pending             0          26m    <none>   <none>        <none>           <none>
application-cpu-557dd6bf86-bfr7h    0/1     ContainerCreating   0          117m   <none>   workernode1   <none>           <none>
application-cpu-557dd6bf86-bgngl    0/1     Pending             0          26m    <none>   <none>        <none>           <none>
application-cpu-557dd6bf86-hkr45    0/1     ContainerCreating   0          115m   <none>   workernode1   <none>           <none>
application-cpu-557dd6bf86-jklwf    0/1     ContainerCreating   0          133m   <none>   workernode1   <none>           <none>
application-cpu-557dd6bf86-qmwvb    0/1     Pending             0          26m    <none>   <none>        <none>           <none>
application-cpu-557dd6bf86-tzz67    0/1     Pending             0          115m   <none>   <none>        <none>           <none>
application-cpu-557dd6bf86-v9d7f    0/1     Pending             0          26m    <none>   <none>        <none>           <none>
test-nginx                          0/1     ContainerCreating   0          155m   <none>   workernode1   <none>           <none>
root@controlplane:~# kubectl get pods -o wide
```

```
root@controlplane:~# kubectl get nodes
NAME           STATUS     ROLES           AGE     VERSION
controlplane   NotReady   control-plane   7m52s   v1.32.0
workernode1    NotReady   <none>          23s     v1.32.0
root@controlplane:~#
```

```
root@controlplane: ~
root@controlplane:~# ^C
root@controlplane:~#   mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
cp: overwrite '/root/.kube/config'? y
root@controlplane:~# ^C
root@controlplane:~# kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml --validate=false
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
root@controlplane:~# kubectl get pods -n kube-system
NAME                                       READY   STATUS             RESTARTS         AGE
coredns-668d6bf9bc-lms6x                   0/1     Pending            0                2m2s
coredns-668d6bf9bc-nl2ct                   0/1     Pending            0                2m2s
etcd-controlplane                          1/1     Running            30 (91s ago)     20s
kube-apiserver-controlplane                1/1     Running            30 (2m25s ago)   2m22s
kube-controller-manager-controlplane       0/1     Running            3 (45s ago)      20s
kube-proxy-m9fj6                           1/1     Running            2 (34s ago)      2m2s
kube-scheduler-controlplane                0/1     CrashLoopBackOff   35 (25s ago)     2m56s
root@controlplane:~#
```

References

Deshpande, N. (2021). Autoscaling cloud-native applications using custom controller of kubernetes, Master's thesis, Dublin, National College of Ireland. Submitted. URL: https://norma.ncirl.ie/5089/

Kubernetes Authors, "Kubernetes Documentation," Kubernetes Project, 2024. [Online]. Available: https://kubernetes.io/docs/. [Accessed: Dec. 12, 2024]

Docker, Inc., "Docker Documentation," Docker Official Website, 2024. [Online]. Available: https://docs.docker.com/. [Accessed: Dec. 12, 2024]

Amazon Web Services, Inc., "Auto Scaling for Amazon EC2," AWS Documentation, 2024. [Online]. Available: https://docs.aws.amazon.com/autoscaling/. [Accessed: Dec. 12, 2024]