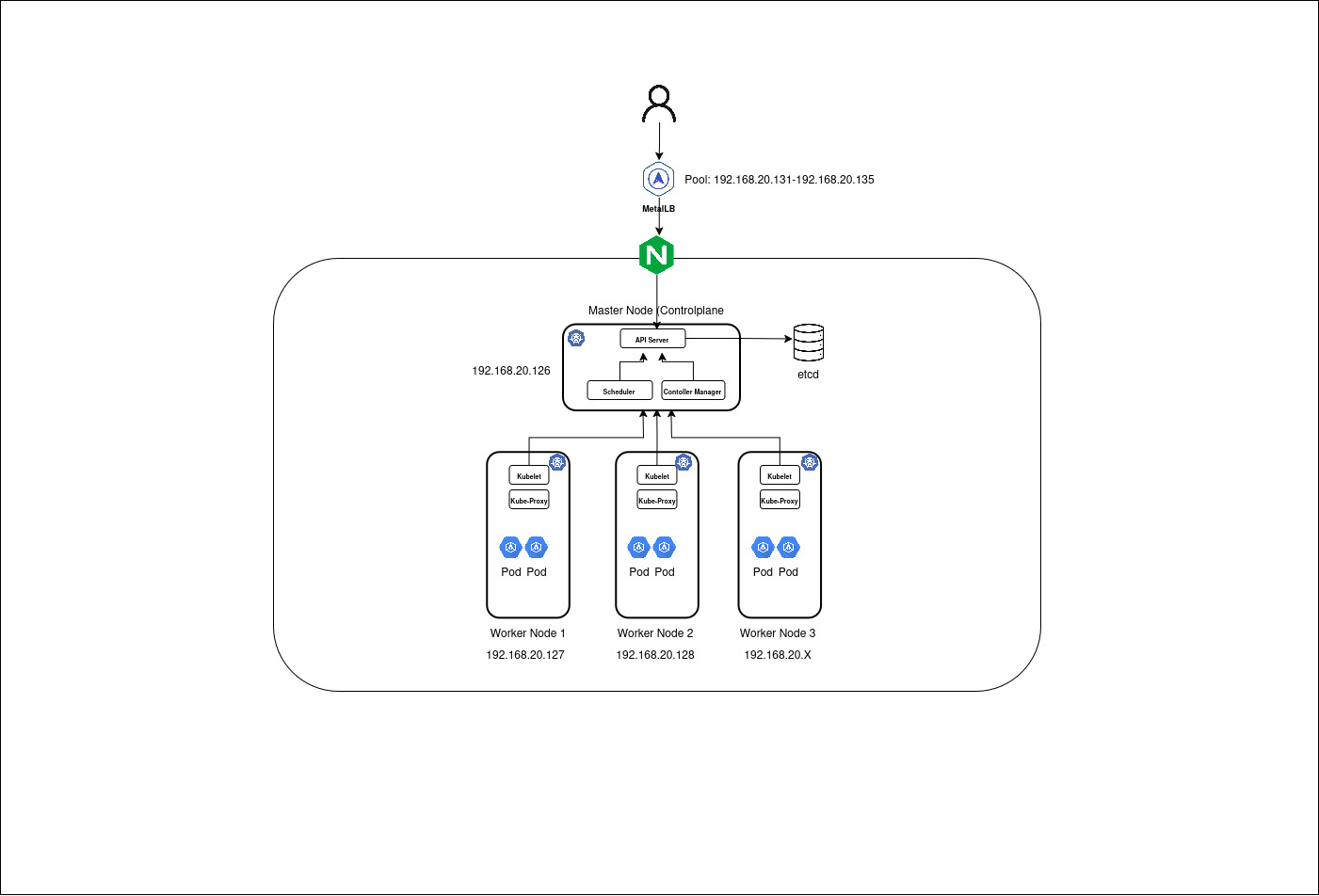
MetalLB Configuraiton



Create Kubernetes Cluster

Master Node Deployment Script: <https://tinyurl.com/k8s-master-node>

Worker Node Deployment Script: <https://tinyurl.com/k8s-worker-node>

Configure MetalLB

kubectl get ns

kubectl get configmap -n kube-system

kubectl edit configmap kube-proxy -n kube-system

<output omitted>

ipvs:

excludeCIDRs: null

minSyncPeriod: 0s

scheduler: ""

strictARP: true

syncPeriod: 0s

tcpFinTimeout: 0s

tcpTimeout: 0s

udpTimeout: 0s

<output omitted>

Source: <https://metallb.io/installation/>

kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.14.9/config/manifests/metallb-native.yaml

kubectl get pods -n metallb-system

Configure IP Address Pool

cat > metallb-config.yaml << EOF

apiVersion: metallb.io/v1beta1

kind: IPAddressPool

metadata:

name: default-pool

namespace: metallb-system

spec:

addresses:

- 192.168.20.131-192.168.20.135

---

apiVersion: metallb.io/v1beta1

kind: L2Advertisement

metadata:

name: default

namespace: metallb-system

spec:

ipAddressPools:

- default-pool

EOF

kubectl apply -f metallb-config.yaml

kubectl get ipaddresspools -n metallb-system

kubectl get l2advertisements -n metallb-system

kubectl get pods -n metallb-system

Install Helm

Source: https://helm.sh/docs/intro/install/

curl -fsSL -o get\_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3

chmod 700 get\_helm.sh

./get\_helm.sh

Install Nginx Ingress Controller

Source: https://kubernetes.github.io/ingress-nginx/deploy/

helm upgrade --install ingress-nginx ingress-nginx \

--repo https://kubernetes.github.io/ingress-nginx \

--namespace ingress-nginx --create-namespace

kubectl get all -n ingress-nginx

Run Workloads:

1. Nginx Deployment Manifest

cat > nginx-deployment.yaml << EOF

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

namespace: default

spec:

replicas: 2

strategy:

type: RollingUpdate

rollingUpdate:

maxUnavailable: 20%

maxSurge: 30%

minReadySeconds: 10

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.20

ports:

- containerPort: 80

resources:

requests:

memory: "256Mi" # 25% of 1 GiB memory.

cpu: "250m" # 25% of 1 core.

limits:

memory: "512Mi"

cpu: "500m"

EOF

2. Nginx Service Manifest

cat > nginx-service.yaml << EOF

apiVersion: v1

kind: Service

metadata:

name: nginx-service

namespace: default

spec:

type: ClusterIP

selector:

app: nginx

ports:

- protocol: TCP

port: 80

targetPort: 80

EOF

3. Nginx Horizontal Pod Autoscale Manifest

cat > nginx-hpa.yaml << EOF

apiVersion: autoscaling/v2

kind: HorizontalPodAutoscaler

metadata:

name: nginx-hpa

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: nginx-deployment

minReplicas: 5

maxReplicas: 20

metrics:

- type: Resource

resource:

name: cpu

target:

type: Utilization

averageUtilization: 70

EOF

4. Nginx Secret

kubectl create secret tls apsissolutions.com \

--cert=/home/jamal/Documents/SSL-CERTS/apsissolutions.com/2025/CA\_chain.crt \

--key=/home/jamal/Documents/SSL-CERTS/apsissolutions.com/2025/apsissolutions.com.key -n default

5. Ingress Resource:

cat > nginx-ingress.yaml << EOF

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: nginx-ingress

namespace: default

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /

nginx.ingress.kubernetes.io/ssl-redirect: "true"

spec:

ingressClassName: nginx

tls:

- hosts:

- mynginx.apsissolutions.com

secretName: apsissolutions.com

rules:

- host: mynginx.apsissolutions.com

http:

paths:

- path: /

pathType: Prefix

backend:

service:

name: nginx-service

port:

number: 80

EOF