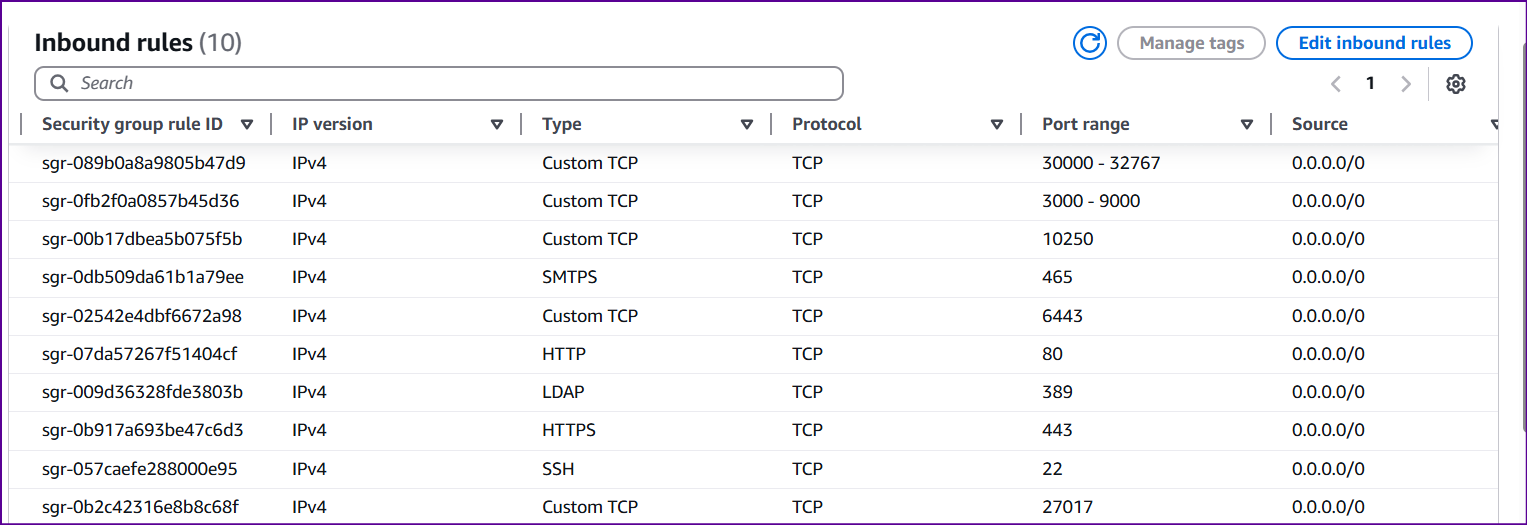
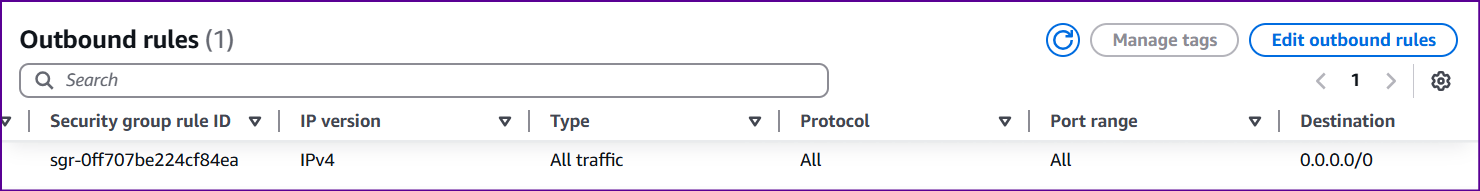
**LOCAL K8 Setup Using Kubeadm**

**Create 2 VMs with Port as below.  
**

****

**🧱 Step 1: Run on All Nodes (Master + Workers)**

# 1. Update

sudo apt update && sudo apt upgrade -y

# 2. Disable Swap (required by kubelet)

sudo swapoff -a

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

# 3. Load required kernel modules

cat <<EOF | sudo tee /etc/modules-load.d/containerd.conf

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

# 4. Set sysctl parameters

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

**🐳 Step 2: Install containerd**

# 1. Install containerd dependencies

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

# 2. Add Docker GPG key and repo

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \

https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# 3. Install containerd

sudo apt update && sudo apt install -y containerd.io

# 4. Generate default config

sudo mkdir -p /etc/containerd

containerd config default | sudo tee /etc/containerd/config.toml > /dev/null

# 5. Enable systemd cgroup driver

sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/' /etc/containerd/config.toml

# 6. Restart containerd

sudo systemctl restart containerd

sudo systemctl enable containerd

**☸️ Step 3: Install Kubernetes**

# 1. Add Kubernetes GPG key and repo

sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | \

sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /" | \

sudo tee /etc/apt/sources.list.d/kubernetes.list

# 2. Install kubeadm, kubelet, kubectl

sudo apt update

sudo apt install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

**🧠 Step 4: Initialize the Control Plane (Master)**

sudo kubeadm init --pod-network-cidr=192.168.0.0/16

📌 Save the kubeadm join ... command shown in the output.

**⚙️ Step 5: Configure kubectl on Master**

mkdir -p $HOME/.kube

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

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

**🌐 Step 6: Install Calico CNI Plugin**

kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.3/manifests/calico.yaml

**🧩 Step 7: Join Worker Nodes**

On each worker node (from the join command you copied earlier):

sudo kubeadm join <MASTER-IP>:6443 --token <TOKEN> --discovery-token-ca-cert-hash sha256:<HASH>

**✅ Fix for CoreDNS Corefile**

1. Run:

kubectl -n kube-system edit configmap coredns

1. Replace **this section**:

forward . /etc/resolv.conf {

max\_concurrent 1000

}

1. **With this block** (correctly indented):

forward . 8.8.8.8 1.1.1.1 {

max\_concurrent 1000

}

The full working Corefile should look like this:

Corefile: |

.:53 {

errors

health {

lameduck 5s

}

ready

kubernetes cluster.local in-addr.arpa ip6.arpa {

pods insecure

fallthrough in-addr.arpa ip6.arpa

ttl 30

}

prometheus :9153

forward . 8.8.8.8 1.1.1.1 {

max\_concurrent 1000

}

cache 30

loop

reload

loadbalance

}

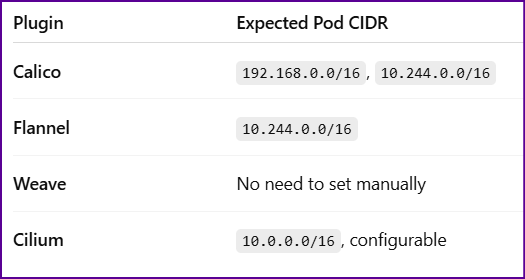
**🚨 Tips:**

* Make sure you **don't touch the indentation** of other blocks.
* The forward block should have 2 spaces more than the .:53 line.
* Avoid using Tab; only use spaces.

**🔄 After saving the file:**

Restart CoreDNS:

kubectl rollout restart deployment coredns -n kube-system



Projects Used:

<https://github.com/jaiswaladi246/Boardgame.git>

<https://github.com/jaiswaladi246/Multi-Tier-BankApp-CD.git>

kubectl get nodes 🡪 To see the worker Nodes  
kubectl create ns webapps 🡪 To create namespace named webapps

kubectl apply -f manifest.yml -n webapps🡪 To deploy in namespace named manifest named manifest.yaml

kubectl describe pod podname -n webapps 🡪 check pod events

kubectl logs podname -n webapps 🡪 check logs of the pod