**Setting up Kubernetes using Kubeadm on AWS EC2**

# **EC2 Prerequisites**

1. **At least 2 CPU cores. (t2.medium)**
2. **At least 4 GB of memory. (t2.medium)**
3. **Ubuntu 22.04 operating system.**
4. **Security Group 6443 allowed.**

# **Run Following Commands on All nodes(Master + Worker)**

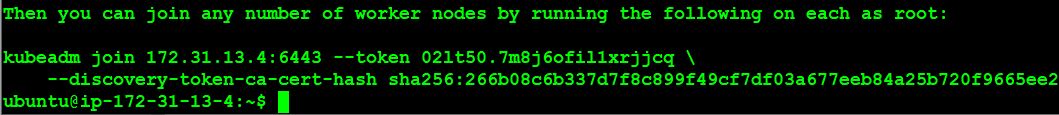
|  |  |
| --- | --- |
| **Update all** | sudo apt-get update |
| **Download the packages** | sudo apt install apt-transport-https curl |
| **Make Directory keyrings** | sudo mkdir -p /etc/apt/keyrings |
| **Get the Docker gpg key** | curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg |
| **Download packages from docker.com** | 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 |
| **Update all** | sudo apt-get update |
| **Install runtime contained** | sudo apt-get install containerd.io |
| **Make Directory containerd** | sudo mkdir -p /etc/containerd |
| **Set configuration** | sudo containerd config default | sudo tee /etc/containerd/config.toml |
| **Open File config.toml** | sudo vim /etc/containerd/config.toml |
| **SystemdCgroup = true** | Set SystemdCgroup = true |
| **Restart Containerd** | sudo systemctl restart containerd |
| **Get the Kubernetes gpg key** | curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add |
| **Add repository** | sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main" |
| **Install Kubernetes packages** | sudo apt install kubeadm kubelet kubectl kubernetes-cni |
| **Disable swap** | sudo swapoff -a |
| **communication between Kubernetes pods** | sudo modprobe br\_netfilter |
| **Enable iptables immediately** | sudo sysctl -w net.ipv4.ip\_forward=1 |

# **Run Following Commands on Master Only**

1. **Initialize the cluster**

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

Note: By Running this command in the output the token will be shown at the last copy that and paste on every Node to join those nodes in cluster.



1. **Set up local kubeconfig**

mkdir -p $HOME/.kube

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

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

1. **Apply Flannel CNI network overlay**

kubectl apply -f https://raw.githubusercontent.com/flannel-io/flannel/v0.20.2/Documentation/kube-flannel.yml

# **Run Following Commands on Node Only**

1. **Switch user to root**

sudo su -

1. **Join the worker nodes to the cluster**

Paste the command which is copied in step Initialize the cluster.

# **Done with the setup let’s check (Login to Master Node)**

1. **Verify the worker nodes have joined the cluster successfully**

kubectl get nodes

# 

**Integrating Docker Hub Private Registry with K8s**

1. **Create a secret by command**

kubectl create secret docker-registry regcred --docker-server=<your-registry-server> --docker-username=<your-name> --docker-password=<your-password> --docker-email=<your-email>

where:

* **<your-registry-server>**is your Private Docker Registry FQDN.
* **<your-name>** is your Docker username.
* **<your-password>** is your Docker password.
* **<your-email>** is your Docker email.

1. **Create Pod using Following Manifest**

apiVersion: v1

kind: Pod

metadata:

name: private-reg

spec:

containers:

- name: private-reg-container

**image: <your-private-image>**

**imagePullSecrets:**

- name: **regcred**