Kubernetes Installation Tutorial

This tutorial will guide you through setting up a Kubernetes (K8s) cluster on your servers using Ansible. We will use containerd as the container runtime instead of Docker and configure the necessary tools for a successful Kubernetes installation.

### **Prerequisites**

1. Ansible: Ensure Ansible is installed on your control machine.
2. Servers: You need access to the servers (referred to as webservers) where Kubernetes will be installed. SSH access with sudo privileges is required.
3. <https://github.com/devopsmay24batch/mydevopsrepo/blob/main/ansible/k8s-install.yaml> - main yaml file is here. This document will explain the steps that are followed to setup the Kuberentes.

### **Step-by-Step Guide**

#### **1. Uninstall Docker**

Docker needs to be removed to avoid conflicts with containerd.

- name: Uninstall docker.io apt: name: docker.io state: absent

* apt: This Ansible module manages apt packages on Debian-based systems.
* name: Specifies the package name to manage.
* state: absent: Ensures the package is removed.

#### **2. Install containerd**

Install containerd, the container runtime we'll use.

- name: Install containerd apt: name: containerd.io state: present

* state: present: Ensures the package is installed.

#### **3. Install Kubernetes tools**

Install kubeadm, kubelet, and kubectl.

- name: Install kubeadm, kubelet, and kubectl apt: name: "{{ item }}  
 state: present

loop:   
 - kubeadm

- kubelet   
 - kubectl

* loop: Allows iterating over a list of items to perform the task multiple times.

#### **4. Initialize containerd directory**

Create the directory for containerd configuration.

- name: Initialize containerd directory   
 file: path: /etc/containerd   
 state: directory

* file: Manages file and directory operations.
* state: directory: Ensures a directory is present.

#### **5. Generate containerd configuration**

Generate the default containerd configuration file.

- name: Initialize containerd config   
 command: containerd config default | sudo tee /etc/containerd/config.toml

* command: Executes the given command on the target host.
* tee: Writes the output of containerd config default to /etc/containerd/config.toml.

#### **6. Configure systemd cgroup driver**

Modify the containerd configuration to use the systemd cgroup driver.

- name: Configure systemd cgroup driver for runC   
command: sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/g' /etc/containerd/config.toml

* sed -i 's/.../.../g': Finds and replaces text in the file in place.

#### **7. Restart containerd**

Restart containerd to apply the new configuration.

- name: Restart containerd   
 systemd:   
 name: containerd state: restarted

* systemd: Manages systemd services.
* state: restarted: Restarts the specified service.

#### **8. Initialize Kubernetes master node**

Initialize the Kubernetes master node with a specific pod network CIDR.

- name: Initialize Kubernetes master   
 command: kubeadm init --pod-network-cidr=10.244.0.0/16   
 register: kubeadm\_output   
 ignore\_errors: yes

* register: Stores the output of the command in a variable (kubeadm\_output).
* ignore\_errors: yes: Allows the playbook to continue even if this step fails (useful for debugging).

#### **9. Set up kubectl for the current user**

Configure kubectl to use the kubeconfig generated by kubeadm.

- name: Set up kubectl for the current user   
 command: "{{ item }}"   
 loop:   
 - mkdir -p $HOME/.kube

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

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

environment:

HOME: /home/{{ ansible\_user\_id }}

* loop: Executes each command for setting up kubectl.
* environment: Ensures the HOME environment variable is correctly set.

#### **10. Install Calico network plugin**

Install the Calico network plugin to manage networking within the Kubernetes cluster.

- name: Install Calico network plugin   
 command: kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

when: "'calico' in kubeadm\_output.stdout"

* kubectl apply -f: Applies a configuration from a file or URL.
* when: Conditional execution based on whether the string 'calico' appears in the output of the kubeadm init command.

### **Full Playbook**

**Here is the complete playbook that is tested on the labs:**

https://github.com/devopsmay24batch/mydevopsrepo/blob/main/ansible/k8s-install.yaml

### **Running the Playbook**

To execute this playbook, save it to a file (e.g., k8s-setup.yml) and run:

ansible-playbook k8s-setup.yml -i your\_inventory\_file

Replace your\_inventory\_file with the path to your inventory file that lists the target servers.

By following this guide, you should have a working Kubernetes cluster with containerd as the container runtime. The playbook ensures that all necessary steps are automated and consistently applied across your servers.

## Important

Kubernetes is hard; even though the above playbook is tested well, there could be issues that need deeper troubleshooting with networking, Linux kernel settings, or issues that are specific to Kubernetes versions. If there are issues during the install, check discuss..io pages or other Kubernetes forum for solutions.

## Common Issues and probable fixes:

Here is a detailed guide for troubleshooting common issues that can occur during Kubernetes master node setup. This includes checking the 6443 port, kubelet configurations and logs, and configuring the firewall (UFW) to enable Kubernetes ports.

### **1. Checking if Port 6443 is Refused**

Symptom: You try to access the Kubernetes API server on port 6443 but receive a "Connection refused" error.

Steps to troubleshoot:

Verify API Server is Running:

1. Ensure that the Kubernetes API server is running. Use the following command:  
   sudo systemctl status kube-apiserver  
   Look for "active (running)" status.

Check Port Binding:

1. Use netstat to check if port 6443 is being used by the API server.  
   sudo netstat -tuln | grep 6443

Inspect API Server Logs:

1. If the API server is not running, check its logs for any errors:  
   sudo journalctl -u kube-apiserver

### **2. Checking Kubelet Configurations**

Symptom: Kubelet is not functioning correctly or failing to start.

**Steps to troubleshoot:**

Verify Kubelet Service:

1. Ensure that the kubelet service is active.  
   sudo systemctl status kubelet

**Review Kubelet Configuration File:**

1. Check the kubelet configuration file, usually located at /var/lib/kubelet/config.yaml or /etc/kubernetes/kubelet.conf.  
   cat /var/lib/kubelet/config.yaml

**Check Kubelet Logs:**

1. Review kubelet logs for any errors or warnings.  
     
   sudo journalctl -u kubelet

**Verify Node Registration:**

1. Ensure the kubelet is properly registered with the API server. Check the nodes in the cluster.  
     
   kubectl get nodes

### **3. Configuring UFW for Kubernetes**

**Symptom: Firewall is blocking necessary Kubernetes ports.**

**Steps to troubleshoot:**

**Enable UFW:**

1. Ensure UFW is enabled.  
   sudo ufw enable

**Allow Kubernetes Ports:**

1. Open the required ports for Kubernetes.  
   sudo ufw allow 6443/tcp # Kubernetes API server   
   sudo ufw allow 2379:2380/tcp # etcd server client API  
   sudo ufw allow 10250/tcp # Kubelet API   
   sudo ufw allow 10251/tcp # kube-scheduler   
   sudo ufw allow 10252/tcp # kube-controller-manager   
   sudo ufw allow 10255/tcp # Read-only Kubelet API (optional)

**Verify UFW Status:**

1. Check the status of UFW to ensure rules are applied.  
   sudo ufw status

### **4. Additional General Troubleshooting Steps**

**Check Cluster Health:**

1. Use kubectl to check the overall health of the cluster.  
   kubectl get componentstatuses  
   Check Pods in kube-system Namespace:
2. Ensure all the essential Kubernetes system pods are running.  
   kubectl get pods -n kube-system
3. Network Plugin Issues - Verify that the network plugin (like Calico, Flannel, Weave, etc.) is correctly installed and configured. Network issues can often cause the cluster to malfunction.

Node Resource Availability:

1. Ensure that the master node has sufficient resources (CPU, memory, disk space).  
   free -h df -h

### **Conclusion**

By following these steps, you should be able to diagnose and resolve most common issues encountered during the setup of a Kubernetes master node. For any other issues, check kubernetes discuss.io website or any other forums. Be specific to check for solutions that match your kubernetes components version (containerd.io, kubeadm, kubelet, kubectl).