**Standard Operating Process**

**(SOP) for**

**How to Deploy Kubernetes Cluster on CentOS**

**Prepared By**

****

**Manish Shahi, Wipro Technologies**

# 

# 1. Contents

[**1.** **Contents 2**](file:///C:/l)

[**2. Document Details 3**](file:///C:/l)**-4**

[**3. Revision History**](file:///C:/l) **4-6**

[**4. Statement of Confidentiality**](file:///C:/l) **6-7**

[**5. Copyright Acknowledgment**](file:///C:/l) **8**

[**6. Disclaimer**](file:///C:/l) **9**

[**7. Warranty**](file:///C:/l) **10**

[**8. Purpose**](file:///C:/l) **11**

[**9. Pre-requisite**](file:///C:/l) **12**

[**10. Introduction**](file:///C:/l) **13**

[**11. Keywords**](file:///C:/l) **14**

[**12. References**](file:///C:/l) **15**

# 2.Document Details

|  |  |
| --- | --- |
| Project Name | How to deploy Kubernetes Cluster on CentOS |
| Account |  |
| Current Version | 1.0.0 |
| List of Contributors | Unix |
| Document Name/ID | How to deploy Kubernetes Cluster on CentOS |

# 3.Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version | Date of Revision | Next Review Date | Description | Author  (Wipro) | Reviewed By (Wipro) | Approved By  (XXXXXX) |
| 1.0.0 | 21-11-2023 |  | How to deploy Kubernetes Cluster on CentOS e Guard Cluster HP-UX | Manish Shahi |  |  |

# 4.Statement of Confidentiality

The information contained in this document is confidential to Wipro. It shall not be disclosed, duplicated, or used for any purpose other than that stated herein, in whole or in part

# 5.Copyright Acknowledgment

Any registered names, trademarks and logos used in this manual remain the property of the owning respective companies or organizations concerned. This documentation may in whole or in part, be copied or replicated to any electronic medium for use within XXXXXX organization only. All copies that are made shall retain all copyright notices.

# 6.Disclaimer

The documentation is prepared based on the specifications and requirements specified or disclosed to Wipro Technologies by XXXXXX and is tailored for the requirements and environment of XXXXXX.

This document provides information and user instructions for XXXXXX. These functional descriptions have an underlying assumption that XXXXXX internal procedures will have defined and implemented levels of user authorization appropriate to the various functions involved in order to minimize the risk of accidental or intentional misuse of those systems.

The contents of the documentation could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein: these changes will be incorporated in subsequent version(s) of the documentation. XXXXXX may make improvements and / or changes in the documentation.

# 7.Warranty

This documentation is provided “as is“, without warranty of any kind, either expressed, implied or statutory, including but not limited to the implied warranties of merchantability or fitness for a particular purpose.

# 8.Purpose

The purpose of this document is to provide the work instructions for How to deploy Kubernetes Cluster on CentOS

**9.Pre-requisite**

At least 2 CentOS servers (in this example we will use 3 servers: 1 master node and 2 worker nodes).

User with sudo or root privileges on each server.

# 10.Introduction

This document is prepared for Wipro sustenance team and provides the concepts of to deploy Kubernetes Cluster on CentOS

.

#### **How to deploy Kubernetes Cluster on CentOS**

We will explore the steps needed to deploy Kubernetes on CentOS. We will install container runtime, install Kubernetes on CentOS, create Kubernetes Cluster, and connect Worker nodes to the cluster.

Kubernetes cluster is a set of nodes that execute applications within containers. Clusters consist of a master node and several worker nodes. These nodes could be physical computers or virtual machines. It depends on the configuration of the cluster. The master node manages and coordinates the worker node. The worker nodes are responsible for executing tasks and running containerized applications by the master node’s directions.

**Step 1: Install Containerd on Master Node and each Worker Node.**

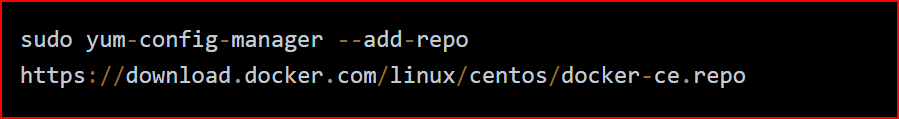
First of all, for installing Kubernetes we need to have one of the container runtime services. In this SOP,

will use Containerd.

Containerd is a container runtime that manages the lifecycle of a container virtual machine. It is a process, which creates, starts, stops, and destroys containers. It is also can download container images from container registries, mount storage, and enable networking for a container.

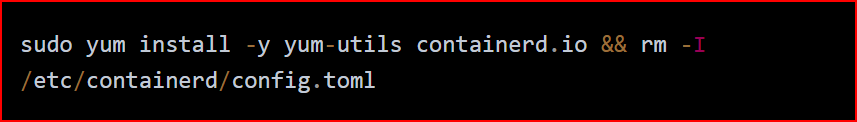
There are no containerd package in the default CentOs repository and because of that, we need to add this package.

To add it we need to execute the following command.



After adding the containerd package to the repository we can install a containerd service using the “yum” package management tools.

For that, we will run the following command.

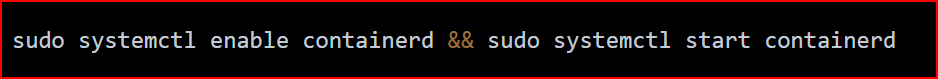


where “yum-utils” — dependency for correct work containerd.

And after successful installation, we need to remove the auto-generated config file “config.toml”. This is necessary to avoid errors when starting Kubernetes in the feature.

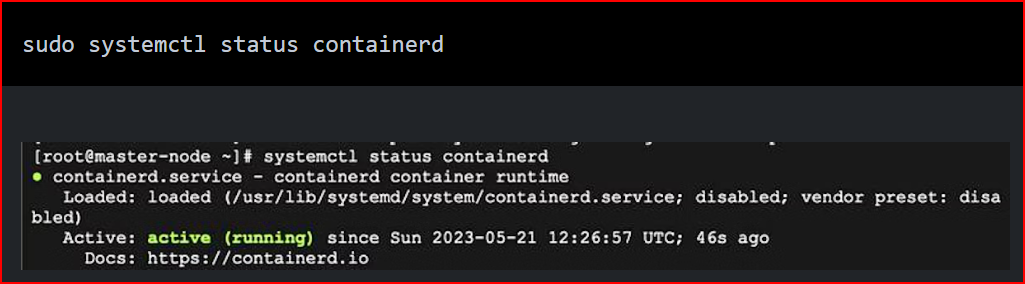
For now, the containerd was successfully installed but it has not yet been launched. Also, we need to enable our service to autostart after a system reboot.

For that, we need to run the following command.



After executing the previous command we will check that service is active and successfully running.

This command will show you the status of your service.

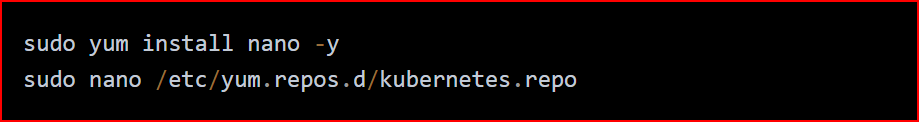


**Step 2: Install and configure Kubernetes on Master Node and each Worker Node.**

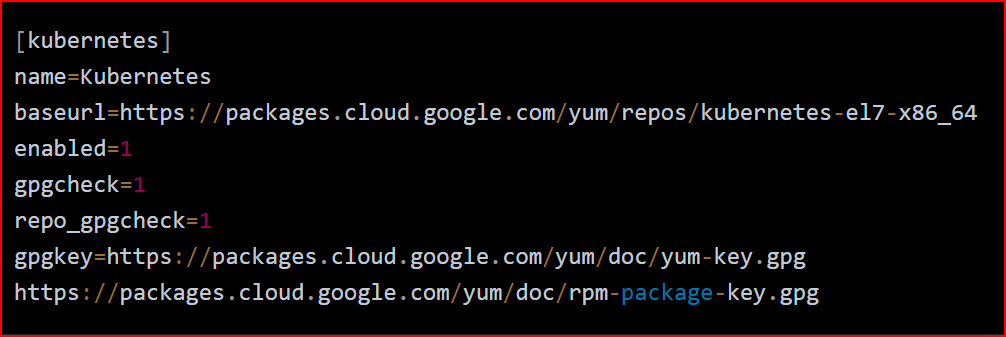
First of all, we need to add the Kubernetes package to the CentOs directory, because it is also absent in the default CentOS package.

For this, we create a new file (use Vim or Nano text editor) and add the following content to the file. In this example, we will use a Nanotext editor.

First of all install nano editor if it is not already installed and then create the file.



Paste the following content to the file.



Where:

baseurl — URL from where the package manager pulls the Kubernetes packages.

enabled — Indicate that repository is enabled and can be used for package installations and updates.

gpgcheck — Indicate that the package manager will verify the GPG signatures of the packages.

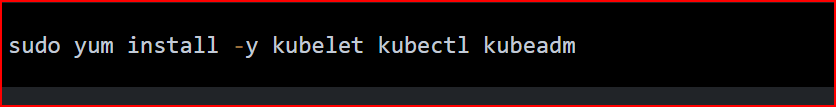
repo\_gpgcheck — Ensures that the repository is trusted and gpgkey should be checked.

gpgkey — URLs where the GPG keys are located.

Once pasted, press Ctrl+O, then Enter to save it. Then press Ctrl+X to exit.

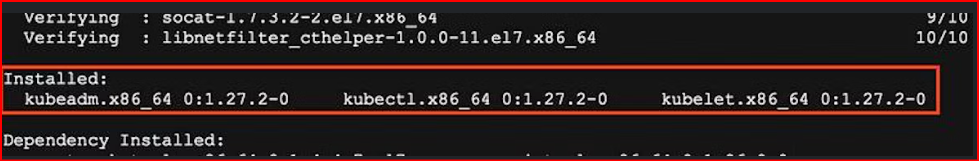
After adding the Kubernetes package to the CentOs directory we can now install Kubernetes services.

For that, we need to execute the following command.



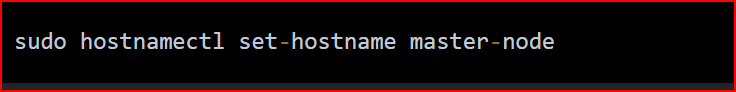
Where kubelet is needed for running and managing containers in the Kubernetes cluster. Kubectl needed to cooperate with a Kubernetes cluster. Kubeadm is needed for bootstrapping the new cluster.

After the execution command, you will see that all three services are successfully installed.

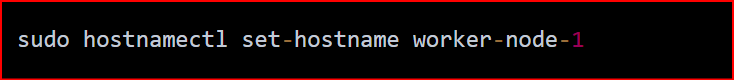


In the next step, we will change our VM's names. This is necessary for a convenient understanding of our hostname.

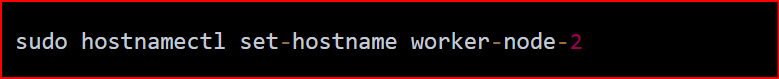
On Master Node we need to execute the following command:



On Worker Node-1 we need to execute the following command:

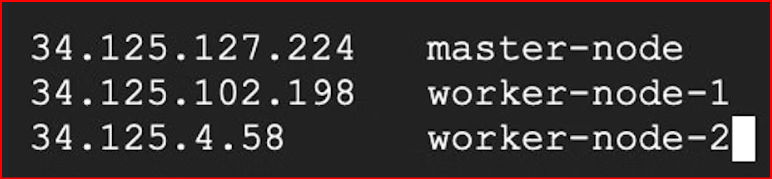


On Worker Node-2 we need to execute the following command:

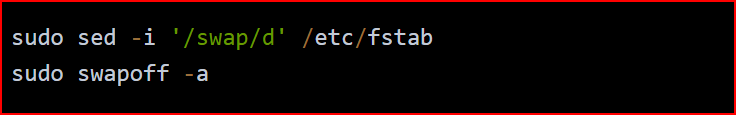


Change your “hosts” file and add the IP address and your new hostname from previous commands to the end of the file.



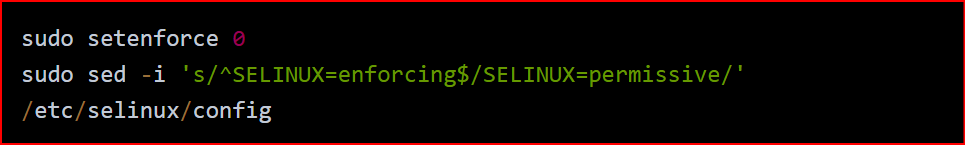


To be sure that nothing will interfere with the Kubernetes work, we need to disable the swap. Because Kubernetes will be managing a memory when running containers.



Also, we need to disable SELinux to avoid any potential conflicts or issues that may arise during the installation process.

For that execute the following commands.

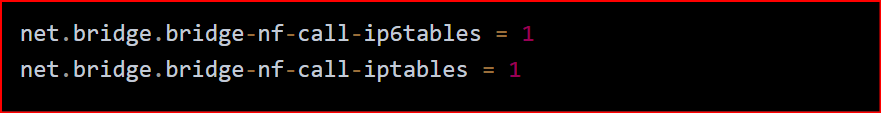


Update iptables settings. To ensure proper network communication and routing for the Kubernetes cluster we need to update iptables settings.

For this, we need to create a “k8s.conf” file.



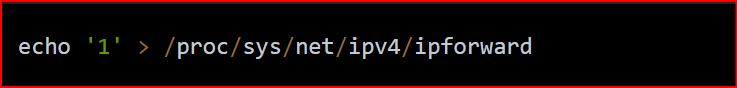
Add to the file the following rows and save the file.



To apply this configuration run the following command.

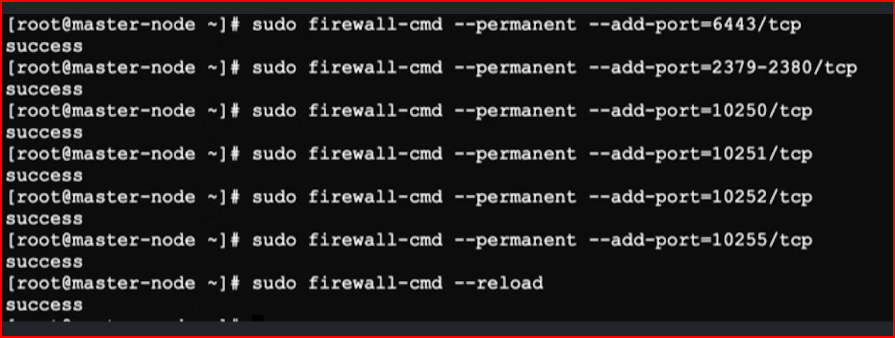


After the previous command, we need to enable the br\_netfilter kernel module so that the packets, that pass through the bridge, will be processed by iptables for filtering and port forwarding, and the Kubernetes pods across the cluster could communicate with each other.

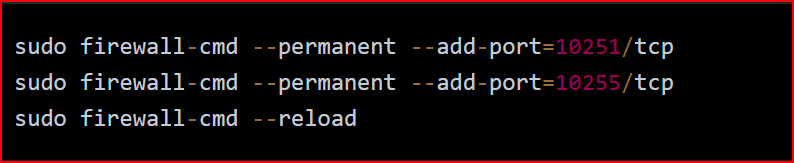


For persistent communication between virtual machines, pods, and containers, we need to add new firewall rules.

On Master Node:

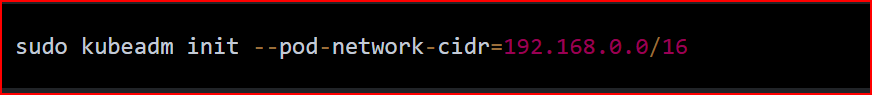


On each Worker Node:



**Step 3: Deploy Kubernetes Cluster.**

The first thing we have to do is initialize a cluster. On Master Node, execute the following command.

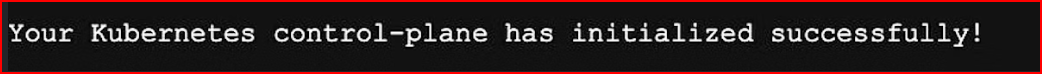


Where “-- pod-network-cidr” defines the IP address range that is assigned to individual pods within the cluster.

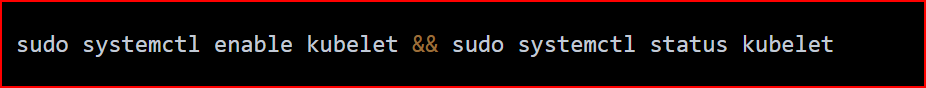
**Note:-**

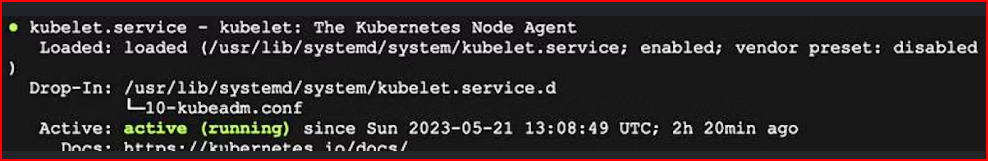
**we used the Calico virtual network. If 192.168.0.0/16 is already in use within your network or you want to use a custom virtual network, you must select a different pod network CIDR, replacing 192.168.0.0/16 in the above command You can choose it on the official Kubernetes web site.**

After executing the command we will see that our Kubernetes control plane has initialized successfully.



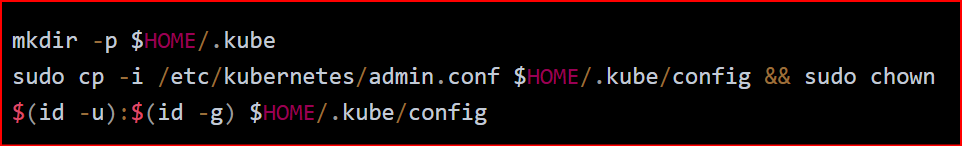
Also, we need to enable auto start service after rebooting and check the kubelet status to make sure that the service status is Active.





For managing the Kubernetes cluster we need to create a folder to store Kubernetes configuration files, copy already generated (from the previous command) config file, and then change permissions.

For that, we need to execute the following commands.



Where:

mkdir — create new folder

cp — copy configuration file to the folder from the previous command.

chown $(id -u):$(id -g) — set the user and group ownership to the current user.

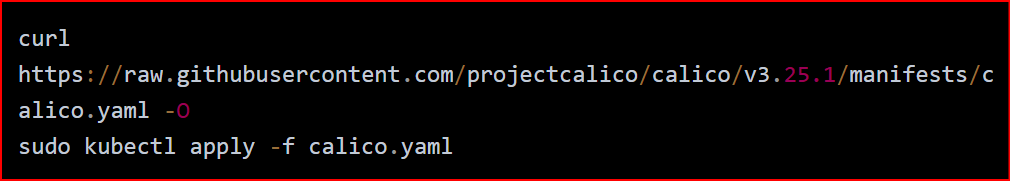
After adding the configuration file we need to set up the network pod. Pods in the cluster are connected via Pod Network.

For this scenario, we use the Calico pod network.

Use the following commands to download and install Calico Pod Network.

**Note:**

**If you use a different virtual network, please change “https://raw.githubusercontent.com/projectcalico/calico/v3.25.1/manifests/calico.yaml -O” and “calico. yaml” in the second command to yours.**

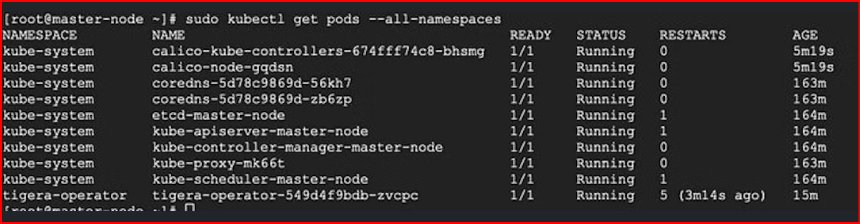
****

After applying the configuration file from the previous command we can check the status of our node.

To verify the status of our node, we will run the following command on Master Node.



Also, let’s confirm that each service has a running status.

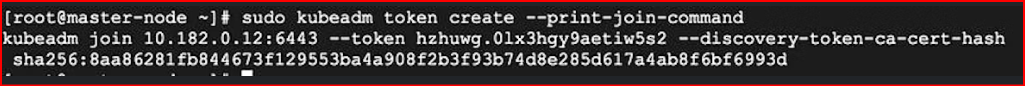


As we can see the status of our Master Node is Ready and all our services have running status.

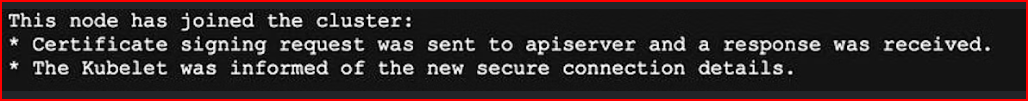
**Step 4: Join Worker Node to the Cluster.**

For this, we need to generate a join command.

To get it, execute the following command on Master Node.



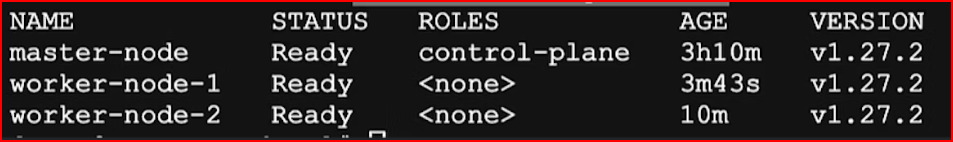
After that, run generated result command from the previous step on each Worker Node.



As we can see our Worker node is successfully connected. And we can check the status of our Worker node on the Master node.

For that, we need to execute the following command on the Master node.





That’s it, all Worker nodes are successfully connected and have Ready status.