LEARNITGUIDE.NET

Install & Configure Kubernetes Cluster with Docker on Linux

📅 August 11, 2018

This tutorial provides you the step by step procedure for installing and configuring kubernetes multinode cluster with Docker on Linux (RHEL7 / CentOS7) using Kubeadm and Kubectl.

But there is also a another way of setting up Kubernetes cluster with Minikube. This post will not cover the minikube installation for now. Because, Minikube is a tool that makes it easy to run Kubernetes locally. Minikube runs a single-node Kubernetes cluster inside a VM on your laptop for beginners who wants to practice Kubernetes, that will not really help on production environment.

In the previous posts, already we have explained the below topics. Refer those links to understand this topic from basics.

What is Kubernetes - Learn Kubernetes from Basics
Create Kubernetes Deployment, Services & Pods Using Kubectl
Create Kubernetes YAML for Deployment, Service & Pods
What is Docker - Get Started from Basics - Docker Tutorial
What is Container, What is Docker on Container - Get Started
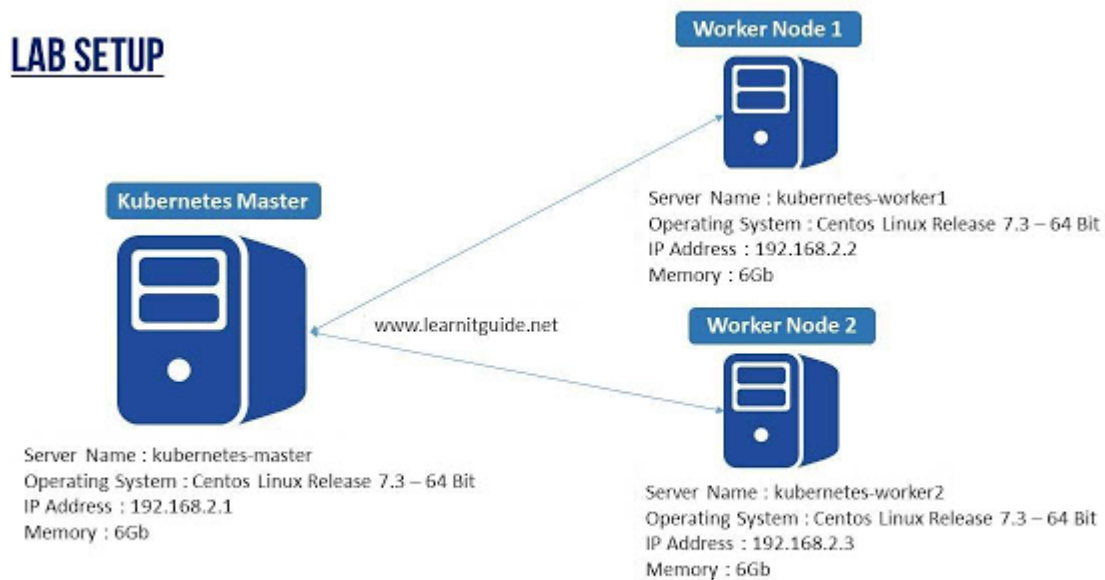How to Install Docker on CentOS 7 / RHEL 7
Docker Images Explained with Examples - Docker Tutorial
How to Run Docker Containers - Explained with Examples

Let's get started.

Our Lab Setup:

Also You can Watch this Tutorial video on our YouTube Channel.

Configure Kubernetes Cluster with Docker on Linux using K...

Prerequisites:

1. Make an entry of each host in /etc/hosts file for name resolution on all kubernetes nodes as below or configure it on DNS if you have DNS server.

> cat /etc/hosts
>
> 192.168.2.1 kubernetes-master.learnitguide.net kubernetes-master
>
> 192.168.2.2 kubernetes-worker1.learnitguide.net kubernetes-worker1
>
> 192.168.2.3 kubernetes-worker2.learnitguide.net kubernetes-worker2

2. Make sure kubernetes master and worker nodes are reachable between each other.
3. Kubernetes doesn't support "Swap". Disable Swap on all nodes using below command and also to make it permanent comment out the swap entry in /etc/fstab file.

> swapoff -a

4. Internet must be enabled on all nodes, because required packages for kubernetes cluster will be downloaded from official repository.

Steps involved to create kubernetes cluster,

On All Nodes:

1. Enable Kubernetes repository on master and all worker nodes
2. Install the required packages on master and all worker nodes
3. Start and Enable docker and kubelet services on master and all worker nodes
4. Allow Network Ports in firewall on master and all worker nodes

On Master Node:

5. Initializing and setting up the kubernetes cluster on Master node
6. Copy /etc/kubernetes/admin.conf and Change Ownership only on Master node

7. Install Network add-on to enable the communication between the pods only on Master node

On Worker Nodes:
8. Join all worker nodes with kubernetes master node

1. Enable Kubernetes repository on master and all worker nodes
Create a repo file for kubernetes and append the lines given below.

```
cat /etc/yum.repos.d/kubernetes.repo
```

Output:

```
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
       https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

2. Install the required packages on master and all worker nodes
Install "docker" and "kubeadm" packages using yum command.

```
yum -y install docker kubeadm
```

3. Start and Enable docker and kubelet services on master and all worker nodes

```
systemctl start docker && systemctl enable docker
systemctl start kubelet && systemctl enable kubelet
```

4. Allow Network Ports in firewall on master and all worker nodes

```
firewall-cmd --permanent --add-port=6443/tcp
firewall-cmd --permanent --add-port=10250/tcp
firewall-cmd --reload
```

Kubernetes service's endpoints are being set with port 6443 and Kubelet listens on port 10250

Also Set "1" to bridge firewall rules,

```
echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
```

If you get an error as "No such file or directory", then load the module "br_netfilter" and try the command again.

```
modprobe br_netfilter
echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
```

5. Initializing and setting up the kubernetes cluster on Master node

Use "kubeadm" command to initialize the kubernetes cluster along with "apiserver-advertise-address" and "--pod-network-cidr" options. It is used to specify the IP address for kubernetes cluster communication and range of networks for the pods.


```
[root@kubernetes-master ~]# kubeadm init --apiserver-advertise-address 192.168.2.1 --pod-network-cidr=172.16.0.0/16
```

Output:

```
[init] using Kubernetes version: v1.11.2
[preflight] running pre-flight checks
      [WARNING Firewalld]: firewalld is active, please ensure ports [6443 10250] are open or
your cluster may not function correctly
I0811 21:10:04.905996   12195 kernel_validator.go:81] Validating kernel version
I0811 21:10:04.906058   12195 kernel_validator.go:96] Validating kernel config
[preflight/images] Pulling images required for setting up a Kubernetes cluster
[preflight/images] This might take a minute or two, depending on the speed of your
internet connection
[preflight/images] You can also perform this action in beforehand using 'kubeadm config
images pull'
[kubelet] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-
```

```
flags.env"
[kubelet] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[preflight] Activating the kubelet service
[certificates] Generated ca certificate and key.

...........................................................

suppressed few messages

---------------------------------------------------

[bootstraptoken] creating the "cluster-info" ConfigMap in the "kube-public" namespace
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
Your Kubernetes master has initialized successfully!
To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/
You can now join any number of machines by running the following on each node
as root:

  kubeadm join 192.168.2.1:6443 --token pxavv6.zwqgdlivwfgbaaud --discovery-token-ca-
cert-hash sha256:0cd1e77fd1514a6ec60e3c67c678c0d88ac80b18ff8184271ecef1ccdc01ee55
[root@kubernetes-master ~]#
```

Kubernetes cluster initialization is completed, Copy the join token highlighted in yellow color from the "kubeadm init" command output and store it somewhere, it is required while joining the worker nodes.

6. Copy /etc/kubernetes/admin.conf and Change Ownership only on Master node

Once kubernetes cluster is initialized, copy "/etc/kubernetes/admin.conf" and change ownership. You could see this same instructions in the output of "kubeadm init" command.

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

7. Install Network add-on to enable the communication between the pods only on Master node

We have lot of network add-on available to enable the network communication with different functionality, Here I have used flannel network provider. Flannel is an overlay network provider that can be used with Kubernetes. You can refer more add-on from <u>here</u>.

```
[root@kubernetes-master ~]# kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-
flannel.yml
```

Output:

```
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.extensions/kube-flannel-ds-amd64 created
daemonset.extensions/kube-flannel-ds-arm64 created
daemonset.extensions/kube-flannel-ds-arm created
daemonset.extensions/kube-flannel-ds-ppc64le created
daemonset.extensions/kube-flannel-ds-s390x created
```

Use "kubectl get nodes" command to ensure the kubernetes master node status is ready. Wait for few minutes until the status of the kubernetes master turn ready state.

```
[root@kubernetes-master ~]# kubectl get nodes
NAME              STATUS  ROLES   AGE      VERSION
kubernetes-master Ready   master  14m      v1.11.1
[root@kubernetes-master ~]#
```

8. Join all worker nodes with kubernetes master node

Now, Login into all worker nodes and use the join token what you have copied earlier to join all the worker nodes with kubernetes master node as below.

```
kubeadm join 192.168.2.1:6443 --token pxavv6.zwqgdlivwfgbaaud --discovery-token-ca-
cert-hash sha256:0cd1e77fd1514a6ec60e3c67c678c0d88ac80b18ff8184271ecef1ccdc01ee55
```

Output:

```
[preflight] running pre-flight checks
    [WARNING RequiredIPVSKernelModulesAvailable]: the IPVS proxier will not be used,
because the following required kernel modules are not loaded: [ip_vs_wrr ip_vs_sh ip_vs
ip_vs_rr] or no builtin kernel ipvs support: map[nf_conntrack_ipv4:{} ip_vs:{} ip_vs_rr:{}
ip_vs_wrr:{} ip_vs_sh:{}]
you can solve this problem with following methods:
 1. Run 'modprobe -- ' to load missing kernel modules;
2. Provide the missing builtin kernel ipvs support
I0811 21:22:23.219089    2523 kernel_validator.go:81] Validating kernel version
```

```
I0811 21:22:23.219199   2523 kernel_validator.go:96] Validating kernel config
[discovery] Trying to connect to API Server "192.168.2.1:6443"
[discovery] Created cluster-info discovery client, requesting info from
"https://192.168.2.1:6443"
[discovery] Requesting info from "https://192.168.2.1:6443" again to validate TLS against the
pinned public key
[discovery] Cluster info signature and contents are valid and TLS certificate validates
against pinned roots, will use API Server "192.168.2.1:6443"
[discovery] Successfully established connection with API Server "192.168.2.1:6443"
[kubelet] Downloading configuration for the kubelet from the "kubelet-config-1.11"
ConfigMap in the kube-system namespace
[kubelet] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-
flags.env"
[preflight] Activating the kubelet service
[tlsbootstrap] Waiting for the kubelet to perform the TLS Bootstrap...
[patchnode] Uploading the CRI Socket information "/var/run/dockershim.sock" to the Node
API object "kubernetes-worker1" as an annotation
This node has joined the cluster:
* Certificate signing request was sent to master and a response
  was received.
* The Kubelet was informed of the new secure connection details.
Run 'kubectl get nodes' on the master to see this node join the cluster.
```

Once worker nodes are joined with kubernetes master, then verify the list of nodes within the kubernetes cluster. Wait for few minutes until the status of the kubernetes nodes turn ready state.

```
[root@kubernetes-master ~]# kubectl get nodes
NAME              STATUS  ROLES   AGE     VERSION
kubernetes-worker1  Ready   <none>  56s     v1.11.2
kubernetes-worker2  Ready   <none>  48s     v1.11.2
kubernetes-master   Ready   master  50m     v1.11.1
[root@kubernetes-master ~]#
```

Thats it, We have successfully configured the kubernetes cluster. Our Kubernetes master and worker nodes are ready to deploy the application.

Going forward we will play more with kubernetes tool. Keep practicing and have fun. Leave your comments if any.

Also refer other articles,

[What is Kubernetes - Learn Kubernetes from Basics](#)
[Create Kubernetes Deployment, Services & Pods Using Kubectl](#)
[Create Kubernetes YAML for Deployment, Service & Pods](#)
[What is Docker - Get Started from Basics - Docker Tutorial](#)
[What is Container, What is Docker on Container - Get Started](#)
[How to Install Docker on CentOS 7 / RHEL 7](#)
[Docker Images Explained with Examples - Docker Tutorial](#)
[How to Run Docker Containers - Explained with Examples](#)

Support Us: Share with your friends and groups.

Stay connected with us on social networking sites, Thank you.
[YouTube](#) | [Facebook](#) | [Twitter](#) | [Pinterest](#) | [Rss](#)

Incoming searches: Kubernetes, Configure Kubernetes Cluster, install Kubernetes, configure Kubernetes, install Kubernetes cluster, create Kubernetes cluster, setup Kubernetes, Kubernetes cluster install, Kubernetes setup, Kubernetes cluster creation, Kubernetes docker, Kubernetes tutorial, Kubernetes wiki, create Kubernetes using minikube, Kubernetes cluster manually, Kubernetes cluster without minikube, full Kubernetes cluster, Kubernetes cluster architecture, Kubernetes cluster tutorial, Kubernetes cluster wiki, install Kubernetes cluster on linux, install Kubernetes cluster on centos, install Kubernetes cluster on rhel, install Kubernetes cluster on rhel 7, install Kubernetes cluster kubeadm, setup kubernetes cluster, multinode kubernetes cluster setup
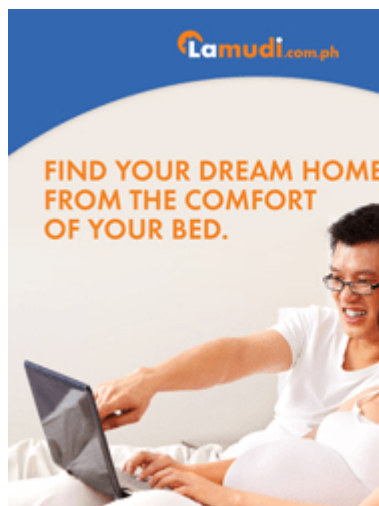
🏷 **Tags**   Containerization   21   Devops   45   Devops Videos   26   Kubernetes   6   Kubernetes Videos   5   Trending   61   Videos   39

POST A COMMENT

Disqus   Facebook   Blogger

3 Comments   **learnitguide**   1   **Login**

♡ **Recommend**   🐦 **Tweet**   f **Share**   Sort by Newest

Join the discussion…

LOG IN WITH   OR SIGN UP WITH DISQUS ?

Name

BALA NAGIREDDY • 12 days ago

kubelet service is not starting after installing docker and kublet
∧ | ∨ • Reply • Share ›

Manikanth Kommoju • 5 months ago

kubelet service is not starting after installing docker and kublet

ᐱ | ᐯ • Reply • Share ›

Admin Mod ➔ Manikanth Kommoju • 5 months ago

What is the error in logs? also what is the OS version?

ᐱ | ᐯ • Reply • Share ›

✉ Subscribe   ⓓ Add Disqus to your siteAdd DisqusAdd   🔒 Disqus' Privacy PolicyPrivacy PolicyPrivacy