

Install Docker on Linux -

As I'm using Amazon Linux2, I can install docker with a
`yum install docker -y`

```
[root@ip-172-31-9-147 ~]# yum install docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.7-5.amzn2 will be installed
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.7-5.amzn2.x86_64
--> Processing Dependency: libcgrouper >= 0.40.rc1-5.15 for package: docker-20.10.7-5.amzn2.x86_64
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.7-5.amzn2.x86_64
--> Processing Dependency: pigz for package: docker-20.10.7-5.amzn2.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.4.6-7.amzn2 will be installed
--> Package libcgrouper.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.0.0-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
docker x86_64 20.10.7-5.amzn2 amzn2extra-docker 42 M
Installing for dependencies:
containerd x86_64 1.4.6-7.amzn2 amzn2extra-docker 24 M
libcgrouper x86_64 0.41-21.amzn2 amzn2-core 66 k
pigz x86_64 2.3.4-1.amzn2.0.1 amzn2-core 81 k
=====
```

Start docker service - **`systemctl start docker`**

For Docker Installations - <https://docs.docker.com/engine/install/>

Install Go -

Go to <https://go.dev/dl/>

Copy the link of go tarfile for linux platform

Install using wget

```
[root@ip-172-31-9-147 ~]# wget https://go.dev/dl/go1.17.7.linux-amd64.tar.gz
--2022-02-21 07:30:44-- https://go.dev/dl/go1.17.7.linux-amd64.tar.gz
Resolving go.dev (go.dev)... 216.239.36.21, 216.239.34.21, 216.239.38.21, ...
Connecting to go.dev (go.dev)|216.239.36.21|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://dl.google.com/go/go1.17.7.linux-amd64.tar.gz [following]
--2022-02-21 07:30:45-- https://dl.google.com/go/go1.17.7.linux-amd64.tar.gz
Resolving dl.google.com (dl.google.com)... 142.250.183.174, 2404:6800:4009:821:200e
Connecting to dl.google.com (dl.google.com)|142.250.183.174|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 134835336 (129M) [application/x-gzip]
Saving to: 'go1.17.7.linux-amd64.tar.gz'

100%[=====>] 134,835,336 146MB/s in 0.9s

2022-02-21 07:30:46 (146 MB/s) - 'go1.17.7.linux-amd64.tar.gz' saved [134835336/134835336]

[root@ip-172-31-9-147 ~]#
```

Extract it using - **`tar xzf go1.17.7.linux-amd64.tar.gz -C /usr/local`**

Add **`/usr/local/go/bin`** to the PATH variable.

export **PATH=\$PATH:/usr/local/go/bin** and add it to .bashrc file to make it permanent and make that file executable using **chmod +x ~/.bashrc**

Install KinD

go install sigs.k8s.io/kind@v0.11.1

You can replace v0.11.1 with the latest stable kind version

Move the KinD Binary to /usr/local/bin -

- You can find the kind binary inside the directory /usr/local/go/bin
- Move it to /usr/local/bin - **mv /usr/local/go/bin/kind /usr/local/bin**
- Make sure you have a path setup for /usr/local/bin

export **PATH=\$PATH:/usr/local/bin** and add it to .bashrc file to make it permanent and make that file executable using **chmod +x ~/.bashrc**

Install Latest Version of Kubectl:

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s  
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"  
chmod +x kubectl  
mv kubectl /usr/local/bin
```

Create a cluster with kind

kind create cluster --name labcluster

```
[root@ip-172-31-9-147 ~]# kind create cluster --name labcluster
Creating cluster "labcluster" ...
 ✓ Ensuring node image (kindest/node:v1.21.1) 📦
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📄
 ✓ Starting control-plane 💻
 ✓ Installing CNI 🌿
 ✓ Installing StorageClass 🗄️
Set kubectl context to "kind-labcluster"
You can now use your cluster with:

kubectl cluster-info --context kind-labcluster

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 😊
```

kubectl get nodes -o wide

```
[root@ip-172-31-9-147 ~]# kubectl get node -o wide
NAME                                STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
labcluster-control-plane           Ready     control-plane,master   8m3s   v1.21.1   172.18.0.2    <none>         Ubuntu 21.04         5.10.96-90.460.amzn2.x86_64   containerd://1.5.2
```

You are now running KinD successfully in Linux

Once done you can delete the cluster using

kind delete cluster

Multi-Node Cluster -

Create a config file kind-example-config.yaml

kind: Cluster

apiVersion: kind.x-k8s.io/v1alpha4

nodes:

- role: control-plane
- role: worker
- role: worker

Create KinD cluster with the config

kind create cluster --config kind-cluster-config.yaml

```
[root@ip-172-31-9-147 ~]# kind create cluster --config kind-cluster-config.yaml
Creating cluster "kind" ...
 ✓ Ensuring node image (kindest/node:v1.21.1) 📦
 ✓ Preparing nodes 📦📦📦
 ✓ Writing configuration 📄
 ✓ Starting control-plane 🎮
 ✓ Installing CNI 🛠️
 ✓ Installing StorageClass 💾
 ✓ Joining worker nodes 🚀
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Have a nice day! 🍀
```

You now have a running KinD Cluster with one master and two worker nodes.

kubectl get nodes -o wide

```
[root@ip-172-31-9-147 ~]# kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
kind-control-plane	Ready	control-plane,master	3m47s	v1.21.1	172.18.0.4	<none>	Ubuntu 21.04	5.10.96-90.460.amzn2.x86_64	containerd://1.5.2
kind-worker	Ready	<none>	3m18s	v1.21.1	172.18.0.5	<none>	Ubuntu 21.04	5.10.96-90.460.amzn2.x86_64	containerd://1.5.2
kind-worker2	Ready	<none>	3m12s	v1.21.1	172.18.0.3	<none>	Ubuntu 21.04	5.10.96-90.460.amzn2.x86_64	containerd://1.5.2