

- Install 3 VM - ubuntu-20.04.3-live-server-amd64.iso (2 proc, 4 Gb RAM, 100Gb, NAT, Install OpenSSH)
- On each VM:
 - `sudo apt update`
 - `sudo apt upgrade`
 - `sudo vi /etc/hosts`
 - `"ip_m" "master_name"`
 - `"ip_node1" "node1_name"`
 - `"ip_node2" "node2_name"`
 - Ping each VM by hostname (hostname must be resolved)
 - `sudo apt-get install curl apt-transport-https git iptables-persistent`



Lab #1. Install Kubernetes



Configuring iptables-persistent

Current iptables rules can be saved to the configuration file /etc/iptables/rules.v4. These rules will then be loaded automatically during system startup.

Rules are only saved automatically during package installation. See the manual page of iptables-save(8) for instructions on keeping the rules file up-to-date.

Save current IPv4 rules?

<Yes>

<No>

- `sudo swapoff -a`
- `sudo vi /etc/fstab`
 - Comment string `#/swap.img none swap sw 0 0`
- `sudo vi /etc/modules-load.d/k8s.conf`
 - `br_netfilter`
 - `overlay`
- `sudo modprobe br_netfilter`
- `sudo modprobe overlay`



Lab #1. Install Kubernetes



- `sudo vi /etc/sysctl.d/k8s.conf`
 - `net.bridge.bridge-nf-call-ip6tables = 1`
 - `net.bridge.bridge-nf-call-iptables = 1`
- `sudo sysctl --system`
- `sudo apt-get install docker docker.io`
- `sudo systemctl enable docker`
- `sudo vi /etc/docker/daemon.json`
 - `{`
 - `"exec-opts": ["native.cgroupdriver=systemd"],`
 - `"log-driver": "json-file",`
 - `"log-opts": {`
 - `"max-size": "100m"`
 - `},`
 - `"storage-driver": "overlay2",`
 - `"storage-opts": [`
 - `"overlay2.override_kernel_check=true"`
 - `]`
 - `}`



Lab #1. Install Kubernetes



- `sudo systemctl restart docker`
- `sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -`
- `sudo vi /etc/apt/sources.list.d/kubernetes.list`
 - `deb https://apt.kubernetes.io/ kubernetes-xenial main`
- `sudo apt-get update`
- `sudo apt-get install kubelet kubeadm kubectl`
- `sudo apt-mark hold kubelet kubeadm kubectl`
- `sudo kubectl version --client`

```
loki@ubuntumaster:~$ sudo kubectl version --client
Client Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.2", GitCommit:"9d142434e3af351a628bffee3939e64c681afa4d", GitTreeState:"clean", BuildDate:"2022-01-19T17:35:46Z", GoVersion:"go1.17.5", Compiler:"gc", Platform:"linux/amd64"}
```



Lab #1. Install Kubernetes



Master node:

- `sudo kubeadm init --pod-network-cidr=10.244.0.0/16`
- `export KUBECONFIG=/etc/kubernetes/admin.conf`
- `sudo vi /etc/environment`
 - `export KUBECONFIG=/etc/kubernetes/admin.conf`
- `sudo kubectl get nodes`

```
loki@ubuntumaster:~$ sudo kubectl get nodes
NAME                STATUS    ROLES    AGE     VERSION
ubuntumaster        NotReady  control-plane,master  4m50s   v1.23.2
loki@ubuntumaster:~$
```

- `sudo kubectl apply -f`
<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>
- `sudo kubeadm token create --print-join-command`



Lab #1. Install Kubernetes



Worker nodes:

- `sudo kubeadm join 192.....`

Master nodes:

- `sudo kubectl get nodes`

```
loki@ubuntumaster:~$ sudo kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ubuntumaster	Ready	control-plane,master	12m	v1.23.2
ubuntunode1	Ready	<none>	4m13s	v1.23.2
ubuntunode2	Ready	<none>	46s	v1.23.2

