

A dark blue vertical bar runs along the left edge of the slide. A blue arrow-shaped banner points to the right from this bar, containing the date. In the bottom-left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

8/28/2023

Kubernetes cluster installation

Amin Ziaei

برای نصب Kubernetes نیازمند اجرا مراحل زیر در سرور های مستر(کنترلر) و Worker می باشیم.

Worker & Master Node Commands

```
sudo apt update && sudo apt upgrade -y
```

```
sudo hostnamectl set-hostname master ---> for Master
```

```
sudo hostnamectl set-hostname worker ---> for Worker
```

اضافه کردن آیدی های Worker و مستر به طور متقابل در `/etc/hosts` هر دو نود مستر و Worker

حال طبق [داکیومنت مربوط به نصب Kubernetes](#) اسکریپت زیر را اجرا میکنیم.

```
#!/bin/bash
```

```
# Install and configure prerequisites
```

```
## load the necessary modules for Containerd
```

```
cat <<EOF | sudo tee /etc/modules-load.d/containerd.conf
```

```
overlay
```

```
br_netfilter
```

```
EOF
```

```
sudo modprobe overlay
```

```
sudo modprobe br_netfilter
```

```
## setup the required kernel/sysctl parameters
```

```
cat <<EOF | sudo tee /etc/sysctl.d/99-kubernetes-cri.conf
```

```
net.bridge.bridge-nf-call-iptables = 1
```

```
net.bridge.bridge-nf-call-ip6tables = 1
```

```
net.ipv4.ip_forward = 1
```

```
EOF
```

```
## Apply sysctl params without reboot
```

```
sudo sysctl -system
```

sudo swapoff -a

حال باتوجه به اینکه container run time مد نظرم ان containerd می باشد نیازمند نصب پکیج آن می باشیم که برای این منظور چندین راه وجود دارد

sudo apt-get update

sudo apt-get -y install containerd

چالشی که با آن مواجه شدم طبق توضیحات داده شده در لینک های زیر پکیج نصب شده با نسخه 22.04 ناسازگار بوده و نیازمند برخی تغییرات در فایل کانفیگ آن می باشد که طی بررسی های انجام شده کلاستر نصب و راه اندازی شد اما پایداری لازم را نداشته و به طور متناوب برخی پاد های روی آن ریستارت میشد لذا برای رفع این مشکل روش دیگر را تست کردم

```
ubuntu@master:~$ kubectl get pods -n kube-system -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE             NOMINATED NODE   READINESS GATES
coredns-5dd5756b68-22tw9            0/1     Pending   0           62m   <none>          <none>            <none>            <none>
coredns-5dd5756b68-kxfz5            0/1     Pending   0           62m   <none>          <none>            <none>            <none>
etcd-master                          1/1     Running   31 (5m27s ago)  61m   188.121.118.185 master            <none>            <none>
kube-apiserver-master               1/1     Running   41 (47m ago)   62m   188.121.118.185 master            <none>            <none>
kube-controller-manager-master      1/1     Running   54 (3m33s ago)  61m   188.121.118.185 master            <none>            <none>
kube-proxy-82s2j                    1/1     Running   10 (65s ago)   62m   188.121.118.185 master            <none>            <none>
kube-scheduler-master               0/1     Running   50 (33s ago)   61m   188.121.118.185 master            <none>            <none>
```

<https://github.com/kubernetes/kubernetes/issues/110177#issuecomment-1161647736>

استفاده از Docker Engine

مطابق با توضیحات داده شده [در این لینک](#) پس از تنظیم dns شکن اقدام به نصب پکیج برنامه میکنیم

Docker Engine depends on containerd and runc. Docker Engine bundles these dependencies as one bundle: containerd.io

sudo apt-get update

sudo apt-get install ca-certificates curl gnupg

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg

```
echo \
```

```
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \
```

```
"$(. /etc/os-release && echo "$VERSION_CODENAME)" stable" | \
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

روش آخر که طبق داکيومنت های مربوط به containerd وجود داشت نصب آخرین ورژن از طریق باینری فایل ها containerd ,runc
cni plugin بود که این روش پایداری لازم را داشته و خطا های مربوطه در کلاستر دریافت نمیشد

<https://github.com/containerd/containerd/blob/main/docs/getting-started.md>

مطابق توضیحات دقیقاً تمامی مراحل عیناً در هر دو نود مستر و Worker اجرا شد.

Step 1: Installing containerd

Download the containerd-<VERSION>-<OS>-<ARCH>.tar.gz archive from
<https://github.com/containerd/containerd/releases> , verify its sha256sum, and extract it under
/usr/local:

```
$ tar Cxvzf /usr/local containerd-1.6.2-linux-amd64.tar.gz
```

```
bin/
```

```
bin/containerd-shim-runc-v2
```

```
bin/containerd-shim
```

```
bin/ctr
```

```
bin/containerd-shim-runc-v1
```

```
bin/containerd
```

```
bin/containerd-stress
```

download the containerd.service unit file from

<https://raw.githubusercontent.com/containerd/containerd/main/containerd.service> into
[/usr/local/lib/systemd/system/containerd.service](https://raw.githubusercontent.com/containerd/containerd/main/containerd.service)

```
systemctl daemon-reload
```

```
systemctl enable --now containerd
```

البته ممکن است در برخی سیستم عامل ها تمامی مسیر ذکر شده وجود نداشته باشد لذا دستی فولدرهایی که وجود نداشت ایجاد شد.

حال باتوجه به توضیحات داده شده در سایت ذکر شده نیازمند اعمال تغییرات در کانفیگ فایل

Containerd

می باشیم لذا مراحل زیر طی شد.

```
sudo mkdir -p /etc/containerd
```

```
containerd config default | sudo tee /etc/containerd/config.toml
```

رفته و مقدار containerd فایل را ادیت کرده و در فایل مربوطه به بخش مربوط به پلاگین

SystemdCgroup = false

را به **true** تغییر میدهیم.

```
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
```

```
BinaryName = ""
```

```
CriImagePath = ""
```

```
CriuPath = ""
```

```
CriuWorkPath = ""
```

```
IoGid = 0
```

```
IoUid = 0
```

```
NoNewKeyring = false
```

```
NoPivotRoot = false
```

```
Root = ""
```

```
ShimCgroup = ""
```

SystemdCgroup = true

حال سرویس مربوطه را ریستارت میکنیم.

```
sudo systemctl restart containerd
```

پس از رفع مشکل بقیه مراحل مربوط به نصب نیازمندی ها را طی کرده که شامل نصب و راه اندازی runc و پلاگین cni می باشد.

Download the runc.<ARCH> binary from <https://github.com/opencontainers/runc/releases> , verify its sha256sum, and install it as /usr/local/sbin/runc.

```
install -m 755 runc.amd64 /usr/local/sbin/runc
```

نصب پلاگین cni

Download the cni-plugins-<OS>-<ARCH>-<VERSION>.tgz archive from <https://github.com/containernetworking/plugins/releases> , verify its sha256sum, and extract it under /opt/cni/bin:

```
mkdir -p /opt/cni/bin
```

```
tar Cxzvf /opt/cni/bin cni-plugins-linux-amd64-v1.3.0.tgz
```

```
./
```

```
./macvlan
```

```
./static
```

```
./vlan
```

```
./portmap
```

```
./host-local
```

```
./vrf
```

```
./bridge
```

```
./tuning
```

```
./firewall
```

```
./host-device
```

```
./sbr
```

```
./loopback
```

```
./dhcp
```

```
./ptp
```

```
./ipvlan
```

```
./bandwidth
```

ورژن هایی که در این داکيومنت نصب شده

Cni=1.3.0

Runc=1.1.9

Containerd=1.7.5

در نهایت آخرین ورژن Kubernetes که هم اکنون ورژن 1.28.1 می باشد طبق توضیحات [داکیومنت](#) مربوطه نصب گردید.

Kubernetes package repositories

These instructions are for Kubernetes 1.28.

1. Update the apt package index and install packages needed to use the Kubernetes apt repository:
2. `sudo apt-get update`
3. *# apt-transport-https may be a dummy package; if so, you can skip that package*
4. `sudo apt-get install -y apt-transport-https ca-certificates curl`
5. Download the public signing key for the Kubernetes package repositories. The same signing key is used for all repositories so you can disregard the version in the URL:
6. `curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg`
7. Add the appropriate Kubernetes apt repository:
8. *# This overwrites any existing configuration in /etc/apt/sources.list.d/kubernetes.list*
9. `echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list`
10. Update the apt package index, install kubelet, kubeadm and kubectl, and pin their version:
11. `sudo apt-get update`
12. `sudo apt-get install -y kubelet kubeadm kubectl`
`sudo apt-mark hold kubelet kubeadm kubectl`

حال با توجه اینکه فرآیند نصب کلاستر باید توسط kubeadm انجام شود.

Master Node

نکته باتوجه به اینکه network باید توسط calico انجام شود. مطابق [داکیومنت کالیکو](#) pod-network-cidr=192.168.0.0/16 در دستور درج شد.

```
sudo kubeadm init --apiserver-advertise-address=37.32.26.241 --pod-network-cidr=192.168.0.0/16
```

پس از اتمام فرآیند نصب دستورات داده شده زده شد.

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
```

همچنین در اینجا توکن جهت راه اندازی نود Worker نیز داده شده که آن را ذخیره کرده و پس از اتمام نصب و راه اندازی شبکه کلاستر در نود Worker میزنیم، البته بعدا نیز امکان دریافت توکن وجود دارد

حال به سراغ نصب و راه اندازی network رفته و مطابق با توضیحات ذکر شده [calico](#) را نصب میکنیم.

```
Every 2.0s: kubectl get pods -n calico-system

NAME                                READY   STATUS              RESTARTS   AGE
calico-kube-controllers-54cc44867c-cw7hg  0/1     ContainerCreating   0          18m
calico-node-7hhd8                      1/1     Running             0          18m
calico-typha-6444f4b497-hw7cj           1/1     Running             0          18m
csi-node-driver-6zmqg                   0/2     ContainerCreating   0          18m
```

در نهایت باید تمامی پاد ها به وضعیت زیر باشد.

```
Every 2.0s: kubectl get pods -A

NAMESPACE   NAME                                READY   STATUS              RESTARTS   AGE
calico-apiserver  calico-apiserver-7446647b69-5c2c9  0/1     Running           0          3m27s
calico-apiserver  calico-apiserver-7446647b69-bll54  0/1     Running           0          3m27s
calico-system    calico-kube-controllers-54cc44867c-cw7hg  1/1     Running           0          23m
calico-system    calico-node-7hhd8                  1/1     Running           0          23m
calico-system    calico-typha-6444f4b497-hw7cj       1/1     Running           0          23m
calico-system    csi-node-driver-6zmqg              2/2     Running           0          23m
kube-system     coredns-5dd5756b68-5c8vz          1/1     Running           0          27m
kube-system     coredns-5dd5756b68-xbjv6          1/1     Running           0          27m
kube-system     etcd-master                        1/1     Running           0          27m
kube-system     kube-apiserver-master              1/1     Running           0          27m
kube-system     kube-controller-manager-master     1/1     Running           1 (2m36s ago)  27m
kube-system     kube-proxy-66nfb                   1/1     Running           0          27m
kube-system     kube-scheduler-master              1/1     Running           1 (2m30s ago)  27m
tigera-operator  tigera-operator-94d7f7696-vptbf    1/1     Running           0          24m
```

پس از نصب و راه اندازی شبکه مستر و همچنین نصب پکیج های ذکر شده کلاستر آماده است تنها کافی است نود Worker را به Master جوین کنیم که دستور مربوطه در نود Worker حتما باید اجرا شود.

طبق توضیحات داده شده در داکيومنت مربوط به Kubernetes در نود های مستر Worker نیازمند باز کردن پورت های قرار داده شده می باشیم که با iptable پورت ها را باز می کنیم

Control plane

Protocol	Direction	Port Range	Purpose	Used By
TCP	Inbound	6443	Kubernetes API server	All
TCP	Inbound	2379-2380	etcd server client API	kube-apiserver, etcd
TCP	Inbound	10250	Kubelet API	Self, Control plane
TCP	Inbound	10259	kube-scheduler	Self
TCP	Inbound	10257	kube-controller-manager	Self

Although etcd ports are included in control plane section, you can also host your own etcd cluster exte

Worker node(s)

Protocol	Direction	Port Range	Purpose	Used By
TCP	Inbound	10250	Kubelet API	Self, Control plane
TCP	Inbound	30000-32767	NodePort Services†	All

کامند مثال

```
sudo iptables -A INPUT -p tcp --dport 6463 -j ACCEPT
```

نکته باتوجه به اینکه تغییرات اعمال شده در iptables دائمی نبوده و چنانچه vm به هر دلیلی ریستارت شود تمامی موارد پاک می شود از پکیج زیر استفاده کرده که به طور دائم ذخیره بماند.

```
sudo apt install iptables-persistent
```

همچنین در صورت ایجاد pod باچند رپلیکلا به جهت جلوگیری از ایجاد pod روی نود مستر کامند زیر را اجرا کرده با اجرا این کامند Scheduler بر روی سایر worker ها pod جدید ایجاد می کند.

```
kubectl taint nodes master node-role.kubernetes.io/master:NoSchedule
```

[/https://kubernetes.io/docs/reference/networking/ports-and-protocols](https://kubernetes.io/docs/reference/networking/ports-and-protocols)

[/https://kubernetes.io/docs/setup/production-environment/container-runtimes](https://kubernetes.io/docs/setup/production-environment/container-runtimes)

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm>

<https://www.tecmint.com/set-permanent-dns-nameservers-in-ubuntu-debian>

<https://deploy.equinix.com/blog/installing-and-deploying-kubernetes-on-ubuntu>

<https://github.com/containerd/containerd/blob/main/docs/getting-started.md>

<https://docs.tigera.io/calico/latest/getting-started/kubernetes/quickstart>

<https://linuxconfig.org/how-to-make-iptables-rules-persistent-after-reboot-on-linux>