**שלבים שלי:**

**במידה ומשוכפל שינוי שם ושחרור IP וחידוש:**

sudo hostnamectl set-hostname <$HOSTNAME>

sudo dhclient -r # release the cloned IP

**קביעה של IP סטטי:**

sudo nano /etc/sysconfig/network-scripts/ifcfg-<interface\_name>  
  
change the BOOTPROTO=”dhcp” to static

and add this to the end of file:

IPADDR=<your\_static\_ip>  
NETMASK=<your\_netmask>  
GATEWAY=<your\_gateway>  
DNS1=<your\_dns\_server1>  
DNS2=<your\_dns\_server2>

sudo systemctl restart network # if you loged with ssh reboot

התקנת kubernetes cluster with kubeadm

1.

sudo yum -y update && sudo systemctl reboot

2.עדכון ה repo’s:

עריכת הקובץ /etc/yum.repos.d/kubernetes.repo

בצורה הבאה:

[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>

\*\*אם בהתקנה של kubectl עולה שגיאה של gpg key אפשר לקבוע את הערכים של בדיקת gpg ל 0

3. התקנות:

sudo yum clean all && sudo yum -y makecache # update the new repo

sudo yum -y install epel-release vim git curl wget kubelet kubeadm kubectl --disableexcludes=kubernetes

4. ביטול אבטחה מוגברת:

sudo setenforce 0

sudo sed -i 's/^SELINUX=.\*/SELINUX=permissive/g' /etc/selinux/config

5. כיבוי ה SWAP מייד ובעליה מחדש :

comment the swap line on /etc/fstab

sudo swapoff -a

free -h # this will show me if I use swap  
cat /proc/swaps # this will show me if I use swap

6. טעינת מודולים שנחוצים לקרנל על מנת שירוצו קונטיינרים:

sudo modprobe overlay

sudo modprobe br\_netfilter

7. איפשור תקשורת בין ה PODS וב cluster   
יצירת קובץ /etc/sysctl.d/kubernetes.conf ועריכה בצורה הבאה:

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

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

net.ipv4.ip\_forward = 1

טעינה מחדש:

sudo sysctl --system # double –

8. התקנת docker :

curl -fsSL [https://get.docker.com](https://get.docker.com/) -o get-docker.sh

sh get-docker.sh

שינוי של ה cgorup ל systemd :

nano /etc/docker/daemon.json

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

אפשר לראות את זה ב docker info | grep Cgroup # start with space

[root@kubernetes docker]# docker info | grep Cgroup

Cgroup Driver: systemd

Cgroup Version: 1

הפעלה של docker

systemctl enable docker

systemctl start docker

**kubernetes לא יודע לעבוד באופן טבעי עם דוקר אלא אם מתקינים לו את הכלי  
cri-dockerd (או כלים דומים) מצאתי התקנה פשוטה במדריך הבא, אם היינו עובדים עם מנוע קונטיינרים אחר הכל היה רץ חלק:**

[**https://computingforgeeks.com/install-mirantis-cri-dockerd-as-docker-engine-shim-for-kubernetes/?expand\_article=1**](https://computingforgeeks.com/install-mirantis-cri-dockerd-as-docker-engine-shim-for-kubernetes/?expand_article=1)

**בסיום כל השלבים במדריך שם הservice שמפעיל את ה cri-docker :**

cri-docker.socket

9. נטרול fire-wall (אני חייב ללמוד איך עושים את זה גם איתו):

sudo systemctl disable --now firewalld

10. הפעלת ה kubeadm

systemctl enable kubelet && systemctl start kubelet

**כאן לכבות את המכונה וליצור שכפולים!!!!**

**רק במאסטר:**

11. משיכה של ה images :

sudo kubeadm config images pull --cri-socket /var/run/cri-dockerd.sock --v=5

אני מציין להשתמש בcri-dockerd.sock שהתקנתי אחרת זה לא ירוץ (אם יש רק דוקר) או יתן שגיאה שיש לי שני cri זמינים..

12. יצירת רשומת DNS ב /etc/host

[root@master-node run]# cat /etc/host

192.168.30.3 master-node  
  
# create later worker-node1,2 etc.

13. kubeadm init – מציין את הרשת הפנימית, את ה CRI ואת שם המאסטר

sudo kubeadm init \

--pod-network-cidr=192.168.0.0/16 \

--cri-socket /var/run/cri-dockerd.sock \

--control-plane-endpoint=master-node