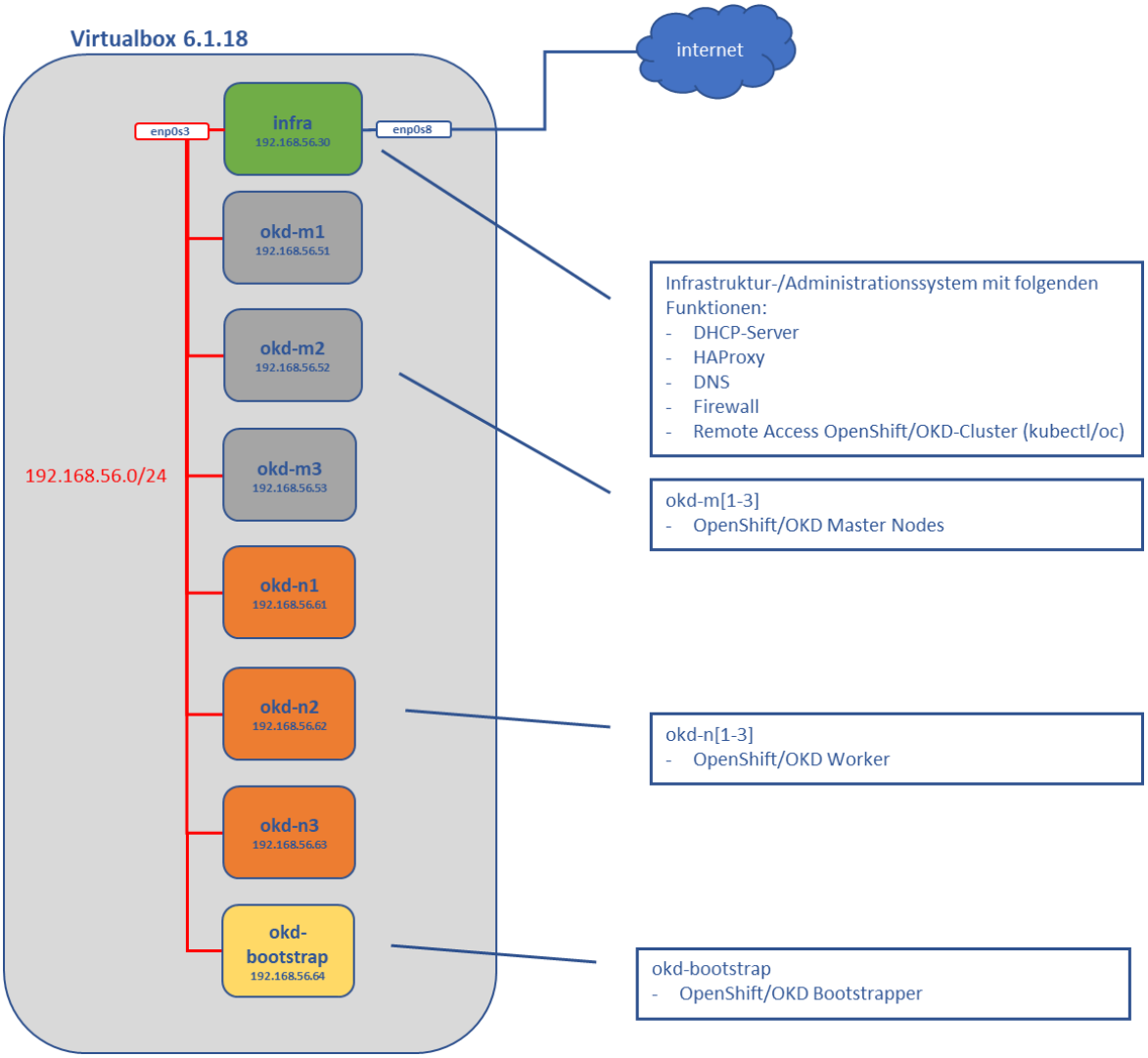
Installation OKD 4.7.x

okd:

- https://www.okd.io/
- Installation ist beispielhaft für 8 Nodes
    - 1 x Infra
    - 1 x Bootstrap
    - 3 x Master
    - 3 x Worker
- Anzahl der Master/Worker kann individuell angepasst werden

Architektur für UPI / OKD



Dokumentation

Releaseinfo:
- https://origin-release.apps.ci.l2s4.p1.openshiftapps.com/

Installation:
- https://docs.okd.io/latest/installing/installing_bare_metal/installing-bare-metal.html

Requirements:
- https://docs.okd.io/latest/installing/installing_bare_metal/installing-bare-metal.html#minimum-resource-requirements_installing-bare-metal

Network Infrastruktur:
- https://docs.okd.io/latest/installing/installing_bare_metal/installing-bare-metal.html#installation-network-user-infra_installing-bare-metal

Virtualisierung

Virtualbox 6.1.x:
- https://www.virtualbox.org/wiki/Downloads

Nodes

Hostname / FQDN	uname -a	CPU/MEM	Nutzung
infra / okd-infra.hs.local	CentOS 7.8	2/8	Administration
			DNS
			DHCP
			Firewall
			HaProxy
okd-m1 / okd-m1.lab.hs.local	CoreOS Fedora release 33	4/16	okd-Master
okd-m2 / okd-m2.lab.hs.local			
okd-m3 / okd-m3.lab.hs.local			
okd-n1 / okd-n1.lab.hs.local	CoreOS Fedora release 33	2/8	okd-Worker
okd-n2 / okd-n2.lab.hs.local			
okd-n3 / okd-n3.lab.hs.local			
okd-bootstrap / okd-bootstrap.lab.hs.local	CoreOS Fedora release 33	4/16	okd-Bootstrapper

Download

Stable Fedora CoreOS für PXE der Master/Worker/Bootstrapper
- https://getfedora.org/en/coreos?stream=stable

CentOS 7.x/8.x für das Infrastruktursystem
- https://www.centos.org/download/

Installtools OKD:
- https://github.com/openshift/okd/releases

Setup Infrastruktur-Node

ToDo‘s:

- Centos 7/8 VM als Infrastruktursystem
  - 2 Netzwerke
    - 192.168.56.0/24 -> Virtualbox HostOnly/Private
    - Public mit Zugang zum Internet -> Bridged
  - Setup Firewalld/IP-Masquerading
  - Setup Apache -> für Installation FCOS
  - Setup DHCP -> für Installation FCOS
    - Konfig im Anhang
  - Setup DNS -> für OKD4/OpenShift
    - Konfig im Anhang
  - Setup HAProxy -> für das LB zu den API-Endpoints
    - Konfig im Anhang
  - Setup NFS -> für PV’s etc.
    - Konfig im Anhang
  - Setup PXE/TFTP -> für Installation FCOS
    - Konfig im Anhang
  - Setup chrony

- Installationvorbereitung

Im Weiteren als root@CentOS 7 (ginge auch als jeder andere User via sudo usw.)

- Konfig-Files für dhcp/haproxy usw. sind exemplarisch im Git Repo enthalten

```
# mkdir -p /data/okd4

# cd /data/okd4

# git clone https://github.com/git67/okd4.git ./local
```

Setup KVM2:

- Wird von oc Cli benötigt

```
# yum install qemu-kvm libvirt libvirt-python libguestfs-tools virt-install

# systemctl enable libvirtd

# systemctl start libvirtd
```

Setup Firewall/IP-Forwarding/Masquerading:

- Kann ggf. feiner granuliert werden
    - Further use …
- enp0s3 -> 192.168.56.0/24
- enp0s8 -> www

```
# sysctl -w net.ipv4.ip_forward=1

# cd /data/okd4/local

#./mk_fw.sh
```

Setup Apache:

```
# yum install httpd -y

# sudo sed -i 's/Listen 80/Listen 8080/' /etc/httpd/conf/httpd.conf

# mkdir -p /var/www/html/okd4/
# chown -R apache: /var/www/html/
# chmod -R 755 /var/www/html/
# setsebool -P httpd_read_user_content 1

# systemctl enable httpd
# systemctl start --now httpd
```

Setup DNS:

- IP/Domain anpassen falls gewünscht
    - Achtung, taucht auch in der OpenShift Installations Konfig auf !!
        - install-config.yaml

```
# yum install bind bind-utils -y

# vi /etc/named.conf (siehe Anhang)
…

# vi /var/named/hs.local.db (siehe Anhang)
…

# vi /var/named/192.168.56 (siehe Anhang)
…

# vi /etc/resolv.conf
…

# systemctl enable named
# systemctl start --now named
```

Setup DHCP:

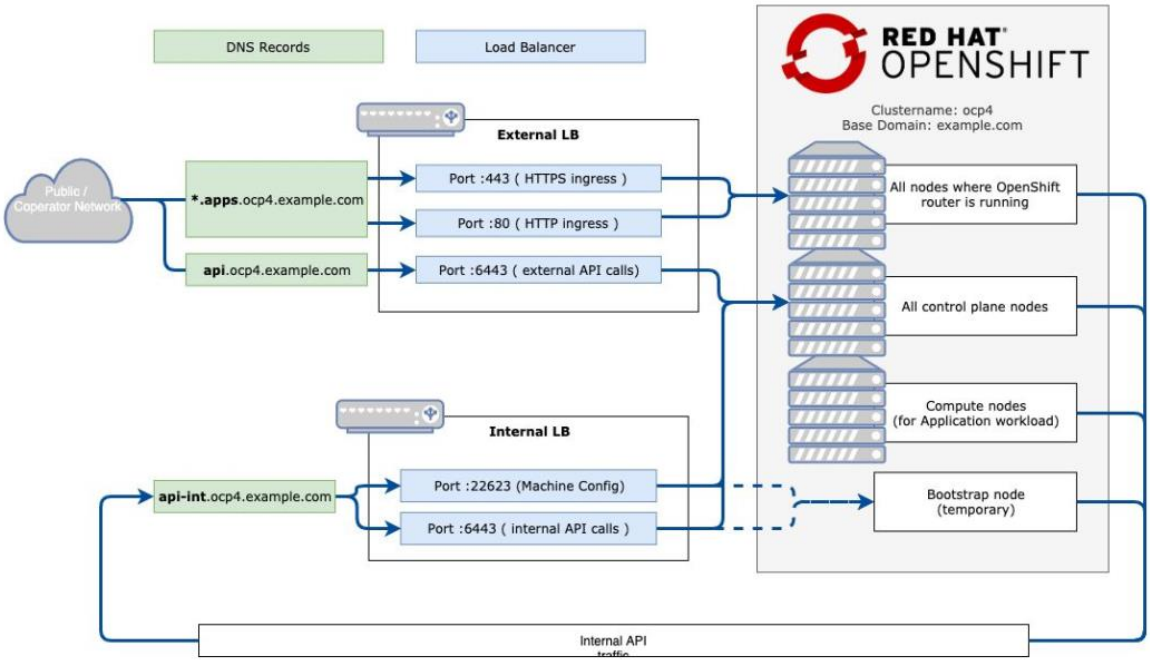
- MAC-Adressen usw. anpassen …
- Falls DNS angepasst worden ist, hier auch nacharbeiten …

```
# yum install dhcp -y

# vi /etc/dhcp/dhcpd.conf (siehe Anhang)
…

# systemctl enable dhcpd
# systemctl start --now dhcpd
```

Setup HAProxy:

- Falls DNS angepasst worden ist, hier auch nacharbeiten …

```
# yum install haproxy -y

# vi /etc/haproxy/haproxy.cfg (siehe Anhang)
…

# setsebool -P haproxy_connect_any 1
# systemctl enable haproxy
# systemctl start --now haproxy
```



Setup NFS-Server:

- /nfs/data
    - OpenShift erwartet für das Registry-PV 100GB, das wird nicht(!) überprüft, also ginge auch weniger 😊

```
# yum install -y nfs-utils
# systemctl enable nfs-server rpcbind

# mkdir -p /nfs/data1/vol /nfs/data1/registry
# chmod -R 777 /nfs
# chown -R nobody:nobody /nfs

# vi /etc/exports
/nfs/data1/vol
192.168.56.0/24(rw,sync,no_root_squash,no_all_squash,no_wdelay)
/nfs/data1/registry
192.168.56.0/24(rw,sync,no_root_squash,no_all_squash,no_wdelay)

# setsebool -P nfs_export_all_rw 1
# systemctl start --now nfs-server rpcbind
```

Setup TFTP/PXE:

```
# yum install -y syslinux tftp-server

# mkdir /var/lib/tftpboot/pxelinux.cfg /var/lib/tftpboot/okd4

# setsebool -P tftp_anon_write 1

# systemctl enable tftp
# systemctl start --now tftp
```

Setup chrony:

- Standard ...

Vorbereitung der Installation:

- Falls noch nicht passiert …

```
# mkdir -p /data/okd4

# cd /data/okd4

# git clone https://github.com/git67/okd4.git ./local

# cd ./local
```

Installations-Konfig:

- https://docs.okd.io/latest/installing/installing_bare_metal/installing-bare-metal.html#installation-bare-metal-config-yaml_installing-bare-metal
- Benötigt:
  - Public SSH-Key für die Nodes
    - ~/.ssh/id_rsa.pub (z.b.)
  - Pull Secret
    - https://cloud.redhat.com/openshift/install/pull-secret
    - Oder der „Fake“
      - {“auths”:{“fake”:{“auth”: “bar”}}}

```
# cd /data/okd4/local

# cp install-config.yaml2edit install-config.yaml

# vi install-config.yaml
apiVersion: v1
baseDomain: hs.local
metadata:
  name: lab
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 0
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OVNKubernetes
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
fips: false
pullSecret: 'HIER das PULL Secret eintragen'
sshKey: 'HIER den PUBLIC SSH-Key eintragen'
```

Download und Installation OpenShift-Installer/-Client und Download und Vorbereitung FCOS:

- Download und Installation OpenShift-Installer/-Client
- Download und Vorbereitung FCOS
- Erstellung Installations-Manifests
- Anpassung Konfig, das App-Pods nicht auf die Master deployed werden
- Erstellung Ignition-Files
- Sicherung intiale Auth's
    - Initiales Passwort kubeadmin
    - Initiales CA
        - export KUBECONFIG=/data/okd4/install/auth/kubeconfig
- Vorbereitung Webserver für FCOS-Installation:

```
# cd /data/okd4/local

# ./mk_inst.sh
```

Vorbereitung FCOS-Installation:

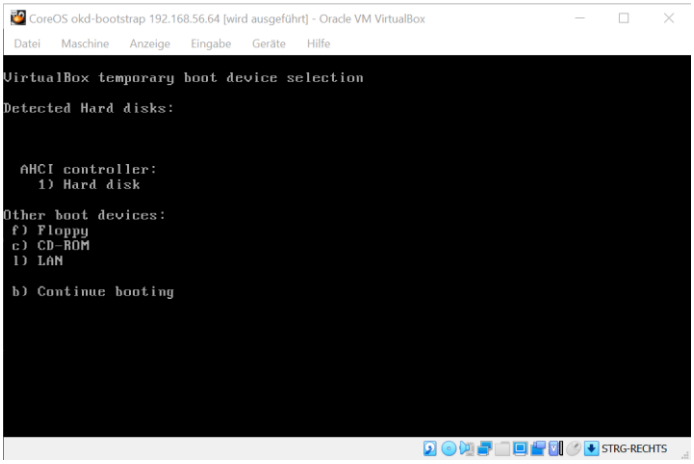
- 7 Files
    - 1 Bootstrap
    - 3 Master
    - 3 Worker
- Pro Node 1 File
    - Siehe Files in /data/okd4/local
        - 01-bootstrap
        - 01-master
        - 01-worker
    - 01-<mac adresse dash separiert, **lower cases** …>
        - Z.b. 01-08-00-27-0e-8b-0d
- Pro Node jeweils das „richtige" Ignition File" eintragen
    - Bootstrap -> bootstrap.ign
    - Master -> master.ign
    - Worker -> worker.ign
- Im Folgenden ein Beispiel für den Bootstrap-Node

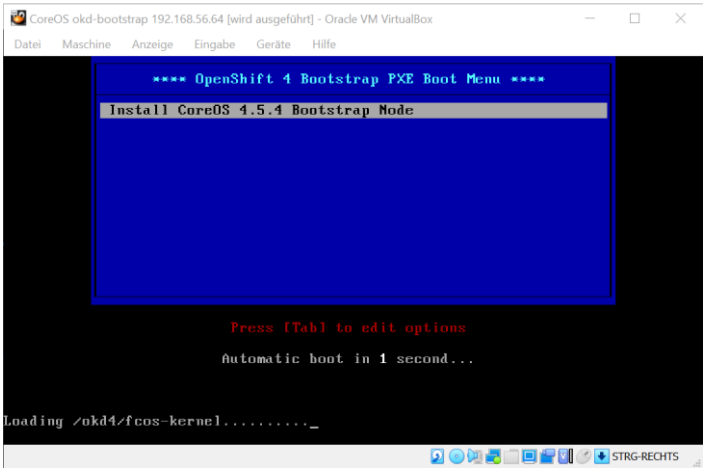
```
# cd /var/lib/tftpboot/pxelinux.cfg/

# vi 01-<mac bootstrap>
default menu.c32
prompt 0
timeout 50
menu title **** OpenShift 4 Bootstrap PXE Boot Menu ****

label Install CoreOS 4.5.4 Bootstrap Node
 kernel /okd4/cos-kernel
 append ip=dhcp rd.neednet=1 coreos.inst.install_dev=sda coreos.inst=yes
coreos.inst.image_url=http://192.168.56.30:8080/okd4/cos.raw.xz
coreos.inst.ignition_url=http://192.168.56.30:8080/okd4/bootstrap.ign
initrd=/okd4/cos-initramfs,/okd4/cos-live-rootfs
```

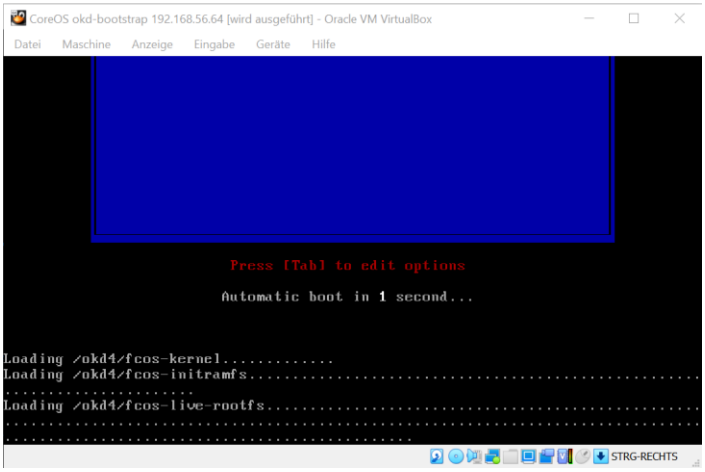
Setup OKD

Virtualbox:

- Einschalten Bootstrap-Node
    - F12
        - Netzwerkboot
- System installiert automatisch und startet neu







Installation tracken:

```
# cd /data/okd4

# openshift-install --dir=install/ wait-for bootstrap-complete --log-level=debug
```

Warten bis Setup des Bootstrap-Node abschlossen ist:

```
…
DEBUG OpenShift Installer 4.7.0-0.okd-2021-02-25-144700
DEBUG Built from commit a005bb9eddcbc97e4cac2cdf4436fe2d524cc75e
INFO Waiting up to 20m0s for the Kubernetes API at
https://api.lab.hs.local:6443...
INFO API v1.20.0-1046+5fbfd197c16d3c-dirty up
INFO Waiting up to 30m0s for bootstrapping to complete...


DEBUG OpenShift Installer 4.7.0-0.okd-2021-02-25-144700
DEBUG Built from commit a005bb9eddcbc97e4cac2cdf4436fe2d524cc75e
INFO Waiting up to 20m0s for the Kubernetes API at
https://api.lab.hs.local:6443...
INFO API v1.20.0-1046+5fbfd197c16d3c-dirty up
…
```

Jetzt die restlichen Nodes (Master/Worker) via Netboot anstarten

Warten bis Setup der restlichen Nodes abschlossen ist:

```
…
INFO Waiting up to 30m0s for bootstrapping to complete...
DEBUG Bootstrap status: complete
INFO It is now safe to remove the bootstrap resources
INFO Time elapsed: 0s
…
```

Jetzt den Bootstrap-Node in /etc/haproxy auskommentieren:

```
# sed -i '/okd-bootstrap/s/^/#/' /etc/haproxy/haproxy.cfg

# systemctl reload haproxy
```

Erster Login:

```
# export KUBECONFIG=/data/okd4/local/auth/kubeconfig

# oc get nodes
```

CSR's freigeben:

```
# oc get csr -ojson | jq -r '.items[] | select(.status == {} ) |
.metadata.name' | xargs oc adm certificate approve

# oc get nodes
```

PV für Registry anlegen:

```
# exportfs -av
exporting 192.168.56.0/24:/nfs/data1/registry
exporting 192.168.56.0/24:/nfs/data1/vol

# vi pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: registry-pv
spec:
  capacity:
    storage: 100Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  nfs:
    path: /nfs/data1/registry
    server: 192.168.56.30

# oc apply -f pv.yaml

# oc edit configs.imageregistry.operator.openshift.io
…
spec:
…
  managementState: Managed
…
  storage:
    managementState:
    pvc:
      claim:
…

# oc get pv
```

Authentifications Provider (htpasswd, ldap, …) einrichten:

- User anlegen
- RBAC -> admin

```
# htpasswd -c -B -b users.htpasswd admin admin

# htpasswd -B -b users.htpasswd devops devops

# oc create secret generic htpasswd-secret \
--from-file htpasswd=users.htpasswd -n openshift-config

# vi htpasswd.yaml
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  name: cluster
spec:
  identityProviders:
  - name: htpasswd_provider
    mappingMethod: claim
    type: HTPasswd
    htpasswd:
      fileData:
        name: htpasswd-secret

# oc apply -f htpasswd.yaml

# oc get identities
NAME                         IDP NAME            IDP USER NAME   USER NAME
USER UID
htpasswd_provider:admin    htpasswd_provider   admin           admin
52c56a52-4a3b-4d7a-a000-b15e2bfb3691
htpasswd_provider:devops   htpasswd_provider   devops          devops
eb22b16a-7a38-4b28-ba9f-d8ecd3f2a69a

# oc adm policy add-cluster-role-to-user cluster-admin admin
```

- User update

```
# htpasswd -B -b users.htpasswd admin admin12

# oc set data secret/htpasswd-secret \
--from-file htpasswd=users.htpasswd -n openshift-config
```

Weitere Schritte:

- Authentifications Provider (htpasswd, ldap, …) einrichten
  - User anlegen
- Project Templates anlegen
  - Quotas etc.
- PV für Registry anlegen
  - Siehe NFS-Server …

/etc/dhcp/dhcpd.conf:

```
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
deny unknown-clients;
option domain-search "hs.local, lab.hs.local";
option domain-name-servers 192.168.56.30;
authorative;
log-facility local7;

subnet 192.168.56.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
    option domain-search "hs.local, lab.hs.local";
    option domain-name-servers 192.168.56.30;
    option routers 192.168.56.30;
    get-lease-hostnames true;
    use-host-decl-names true;
    default-lease-time 600;
    max-lease-time 7200;
    filename "pxelinux.0";
    next-server 192.168.56.30;
 }

 host okd-bootstrap {
    option host-name "okd-bootstrap.lab.hs.local";
    fixed-address 192.168.56.64;
    hardware ethernet 08:00:27:36:5B:BE;
 }

 host okd-m1 {
    option host-name "okd-m1.lab.hs.local";
    fixed-address 192.168.56.51;
    hardware ethernet 08:00:27:BE:3D:8B;
 }

 host okd-m2 {
    option host-name "okd-m2.lab.hs.local";
    fixed-address 192.168.56.52;
    hardware ethernet 08:00:27:C4:91:C6;
 }

 host okd-m3 {
    option host-name "okd-m3.lab.hs.local";
    fixed-address 192.168.56.53;
    hardware ethernet 08:00:27:0E:8B:0D;
 }

 host okd-n1 {
    option host-name "okd-n1.lab.hs.local";
    fixed-address 192.168.56.61;
    hardware ethernet 08:00:27:B8:3B:48;
 }

 host okd-n2 {
    option host-name "okd-n2.lab.hs.local";
    fixed-address 192.168.56.62;
    hardware ethernet 08:00:27:D0:C9:B3;
 }

 host okd-n3 {
    option host-name "okd-n3.lab.hs.local";
    fixed-address 192.168.56.63;
    hardware ethernet 08:00:27:FB:0F:54;
 }
```

/etc/named.conf:

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// See the BIND Administrator's Reference Manual (ARM) for details about the
// configuration located in /usr/share/doc/bind-{version}/Bv9ARM.html

options {
        listen-on port 53 { 127.0.0.1; 192.168.56.30; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        recursing-file  "/var/named/data/named.recursing";
        secroots-file   "/var/named/data/named.secroots";
        allow-query     { localhost; 192.168.56.0/24; };

        /*
         - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
         - If you are building a RECURSIVE (caching) DNS server, you need to enable
           recursion.
         - If your recursive DNS server has a public IP address, you MUST enable access
           control to limit queries to your legitimate users. Failing to do so will
           cause your server to become part of large scale DNS amplification
           attacks. Implementing BCP38 within your network would greatly
           reduce such attack surface
        */
        recursion yes;

        dnssec-enable yes;
        dnssec-validation yes;

        /* Path to ISC DLV key */
        bindkeys-file "/etc/named.root.key";

        managed-keys-directory "/var/named/dynamic";

        pid-file "/run/named/named.pid";
        session-keyfile "/run/named/session.key";
    forwarders {
        8.8.8.8;
        8.8.4.4;
    };
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
        type hint;
        file "named.ca";
};

zone "hs.local" IN {

         type master;

         file "/var/named/hs.local.db";

         allow-update { none; };
};

zone "56.168.192.in-addr.arpa" IN {

          type master;

          file "/var/named/192.168.56.db";

          allow-update { none; };
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";;
```

/var/named/hs.local.db:

```
$TTL    604800
@       IN      SOA     okd-infra.hs.local. admin.hs.local. (
                  1     ; Serial
             604800     ; Refresh
              86400     ; Retry
             241964     ; Expire
              604800     ; Negative Cache TTL
)

; name servers - NS records
    IN      NS      okd-infra

; name servers - A records
okd-infra.hs.local.          IN      A       192.168.56.30

; OpenShift Container Platform Cluster - A records
okd-bootstrap.lab.hs.local.        IN      A      192.168.56.64
okd-m1.lab.hs.local.        IN      A      192.168.56.51
okd-m2.lab.hs.local.         IN      A      192.168.56.52
okd-m3.lab.hs.local.         IN      A      192.168.56.53
okd-n1.lab.hs.local.        IN      A      192.168.56.61
okd-n2.lab.hs.local.        IN      A      192.168.56.62
okd-n3.lab.hs.local.        IN      A      192.168.56.63

; OpenShift internal cluster IPs - A records
api.lab.hs.local.    IN    A    192.168.56.30
api-int.lab.hs.local.    IN    A    192.168.56.30
*.apps.lab.hs.local.    IN    A    192.168.56.30
etcd-0.lab.hs.local.    IN    A     192.168.56.61
etcd-1.lab.hs.local.    IN    A     192.168.56.62
etcd-2.lab.hs.local.    IN    A    192.168.56.63
console-openshift-console.apps.lab.hs.local.     IN     A     192.168.56.30
oauth-openshift.apps.lab.hs.local.     IN     A     192.168.56.30

; OpenShift internal cluster IPs - SRV records
_etcd-server-ssl._tcp.lab.hs.local.    86400     IN    SRV     0    10    2380    etcd-0.lab
_etcd-server-ssl._tcp.lab.hs.local.    86400     IN    SRV     0    10    2380    etcd-1.lab
_etcd-server-ssl._tcp.lab.hs.local.    86400     IN    SRV     0    10    2380    etcd-2.lab
```

/var/named/192.168.56.db:

```
$TTL    604800
@       IN      SOA     okd-infra.hs.local. admin.hs.local. (
                  6     ; Serial
             604800     ; Refresh
              86400     ; Retry
             2419200     ; Expire
              604800     ; Negative Cache TTL
)

; name servers - NS records
    IN      NS      okd-infra.hs.local.

; name servers - PTR records
30    IN    PTR    okd-infra.hs.local.

; OpenShift Container Platform Cluster - PTR records
64    IN    PTR    okd-bootstrap.lab.hs.local.
51    IN    PTR    okd-m1.lab.hs.local.
52    IN    PTR    okd-m2.lab.hs.local.
53    IN    PTR    okd-m3.lab.hs.local.
61    IN    PTR    okd-n1.lab.hs.local.
62    IN    PTR    okd-n2.lab.hs.local.
63    IN    PTR    okd-n3.lab.hs.local.
30    IN    PTR    api.lab.hs.local.
30    IN    PTR    api-int.lab.hs.local.
30    IN    PTR    console-openshift-console.apps.lab.hs.local.
```

/etc/haproxy/haproxy.cfg:

```
# Global settings
#---------------------------------------------------------------------
global
    maxconn     20000
    log         /dev/log local0 info
    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    user        haproxy
    group       haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option http-server-close
    option forwardfor       except 127.0.0.0/8
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          300s
    timeout server          300s
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 20000

listen stats
    bind :9000
    mode http
    stats enable
    stats uri /

frontend okd4_k8s_api_fe
    bind :6443
    default_backend okd4_k8s_api_be
    mode tcp
    option tcplog

backend okd4_k8s_api_be
    balance source
    mode tcp
    server      okd-bootstrap.lab.hs.local 192.168.56.64:6443 check
    server      okd-m1.lab.hs.local 192.168.56.51:6443 check
    server      okd-m2.lab.hs.local 192.168.56.52:6443 check
    server      okd-m3.lab.hs.local 192.168.56.53:6443 check

frontend okd4_machine_config_server_fe
    bind :22623
    default_backend okd4_machine_config_server_be
    mode tcp
    option tcplog

backend okd4_machine_config_server_be
    balance source
    mode tcp
    server      okd-bootstrap.lab.hs.local 192.168.56.64:22623 check
    server      okd-m1.lab.hs.local 192.168.56.51:22623 check
    server      okd-m2.lab.hs.local 192.168.56.52:22623 check
    server      okd-m3.lab.hs.local 192.168.56.53:22623 check

frontend okd4_http_ingress_traffic_fe
    bind :80
    default_backend okd4_http_ingress_traffic_be
    mode tcp
    option tcplog

backend okd4_http_ingress_traffic_be
    balance source
    mode tcp
    server      okd-n1.lab.hs.local 192.168.56.61:80 check
    server      okd-n2.lab.hs.local 192.168.56.62:80 check
    server      okd-n3.lab.hs.local 192.168.56.63:80 check

frontend okd4_https_ingress_traffic_fe
    bind *:443
    default_backend okd4_https_ingress_traffic_be
    mode tcp
    option tcplog

backend okd4_https_ingress_traffic_be
    balance source
    mode tcp
    server      okd-n1.lab.hs.local 192.168.56.61:443 check
    server      okd-n2.lab.hs.local 192.168.56.62:443 check
    server      okd-n3.lab.hs.local 192.168.56.63:443 check
```